Engineering Digital Ecosystems

A methodology for model-based and subject-oriented design

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Jakob Bönsch
geb. in Aachen

Tag der mündlichen Prüfung:	02.12.2025
Hauptreferent:	Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova
Korreferent:	Prof. Dr.-Ing. Kai Furmans

Kurzfassung

Digitale Ecosysteme stellen ein neuartiges Paradigma für industrielle Zusammenarbeit dar, das Coopetition, Resilienz und Anpassungsfähigkeit zwischen vielfältigen Akteuren betont. Ihre inhärente Komplexität – bedingt durch Heterogenität, Emergenz und Föderation – erschwert jedoch eine effektive Gestaltung erheblich. Diese Studie begegnet diesen Herausforderungen mit der Einführung der Engineering Digital Ecosystems (EDE) Methodologie, einem modellbasierten und subjektorientierten Gestaltungsrahmen, der auf die Bedürfnisse sozio-technischer Digitaler Ecosysteme zugeschnitten ist, insbesondere im Kontext kleiner und mittelständischer Unternehmen (KMU). Geleitet vom Paradigma der Design Science Research (DSR) untersucht die Arbeit zwei zentrale Forschungsfragen: (1) Wie kann der Gestaltungsprozess Digitaler Ecosysteme durch eine modellbasierte Methodologie unterstützt werden? und (2) Welche Art von subjektorientiertem Modell eignet sich zur Beschreibung solcher Ecosysteme? Die resultierende EDE-Methodologie integriert einen Vorgehensmodell basierend auf dem Double-Diamond mit einem mehrschichtigen Synthesis Model, das strukturelle, funktionale, interaktive und verhaltensbezogene Aspekte Digitaler Ecosysteme abbildet. Der Rahmen wurde iterativ im industriellen Anwendungskontext (IntWertL) entwickelt und evaluiert, wodurch validierte Artefakte und praxisnahe Erkenntnisse gewonnen wurden. Diese Arbeit liefert die erste zufriedenstellende Methodologie zur Gestaltung Digitaler Ecosysteme und bietet einen systematischen, nutzerorientierten Ansatz, der Klarheit, Kommunikation, und Abstimmung entlang des gesamten Designprozesses verbessert.

Abstract

Digital Ecosystems offer a novel paradigm for industrial collaboration, emphasizing coopetition, resilience, and adaptability among diverse stakeholders. However, their inherent complexity—stemming from heterogeneity, emergent behaviors, and distributed governance—poses significant challenges to effective design. This study addresses these challenges by introducing the Engineering Digital Ecosystems (EDE) methodology, a model-based and subject-oriented design framework tailored to the needs of socio-technical Digital Ecosystems, particularly for small and medium-sized enterprises (SMEs). Guided by the Design Science Research (DSR) paradigm, the study investigates two core research questions: (1) how the design process of Digital Ecosystems can be supported by a model-based methodology, and (2) what kind of subject-oriented model is suitable for describing such ecosystems. The resulting EDE methodology integrates a modified Double Diamond design approach with a multi-layered Synthesis Model that captures structural, functional, interactional, and behavioral aspects of Digital Ecosystems. The framework was iteratively developed and evaluated within a real-world industrial context (IntWertL), yielding validated artifacts and practical insights. This work contributes the first satisficing methodology for Engineering Digital Ecosystems, offering a systematic, stakeholder-inclusive approach that enhances clarity, communication, and alignment throughout the ecosystem design process.

Preface

An dieser Stelle möchte ich "danke" sagen. *Danke* an all jene Menschen, die mich auf dem Weg hierher begleitet und unterstützt haben. Einige wenige von euch möchte ich auch persönlich ansprechen:

An erster Stelle steht dabei meine Familie: Liebe Anita, *danke* für alles! Die Bedeutung deiner Unterstützung ist nicht in Worte zu fassen. Ohne dich wäre ich heute ein anderer Mensch, und mein Leben wäre bedeutend leerer. Lieber Jonas, liebe Thea, ihr habt mir das Schreiben dieser Arbeit oft erschwert. Trotzdem *danke*, denn ihr habt diese Zeit so unendlich viel schöner gemacht. *Danke* an meine Eltern, ihr habt mir diesen Lebensweg ermöglicht. Und schließlich *vielen Dank* auch an meine Geschwister: Elisabeth & Matthias, Gereon & Doro, Cordula & Max, Johannes & Anna. Eure Anmerkungen haben diese Arbeit sprachlich, stilistisch und inhaltlich bereichert. Viel Wichtiger ist mir aber noch, welche Stütze ihr darüber hinaus in meinem Leben seid.

Während der vergangenen fünf Jahre habe ich am IMI – dem Institut für Informationsmanagement im Ingenieurwesen – meine wissenschaftliche Heimat gefunden. Vielen *Dank* an alle meine Kolleginnen und Kollegen aus den vergangenen Jahren, die diese Zeit geprägt haben. Zunächst *danke* ich dir, Matthes, dass du mich ans IMI geholt und mich mit der Subjektorientierung vertraut gemacht hast. Dieses Paradigma ist ein Grundstein dieser Dissertation und bleibt zentral für mein Denken. Andreas, *danke* für deinen persönlichen Einsatz, der mir einen hervorragenden Start ermöglicht hat. *Danke* Svenja, für die Zusammenarbeit und die Diskussionen – dein Input sowie dein großer Einsatz haben das Tandem aus Matthes und mir wesentlich bereichert. Zum Abschluss meiner Promotionsphase hatte ich das Glück mit zwei außerordentlich talentierten Menschen im gleichen Boot zu sitzen: Anjela und Lucas – ihr wart Motivatoren, Ratgeberinnen und Freunde: *Danke*!

Kein Projekt hat meine Arbeit so sehr geprägt wie IntWertL. *Danke* an alle Ideengeber, Projektpartner und Förderer. Namentlich zuerst Wolfgang, Gerhard und Michael, die mit der Projektidee auf mich zugekommen sind und mich mit ihrer Begeisterung schnell angesteckt haben. *Danke* an Isabell, Thomas, Vladimir und Oliver für euer Engagement und das Vorantreiben der Vision. *Danke* Sven und Gaston, dass ich mit euch zusammen Konzepte für Datenräume und digitale Ecosysteme erkunden und erschaffen konnte, und danke an Josef, Mike und John, dass wir diese dann gemeinsam realisiert haben.

Danke auch an Prof. Kai Furmans und Prof. Marcus Geimer, die sich intensiv mit dieser Arbeit auseinandergesezt haben, für Ihre Zeit. Ihre kritischen Nachfragen und Verbesserungsvorschläge waren eine Bereicherung.

Sehr geehrte Prof. Dr. Dr.-Ing. Dr. h. c. Ovtcharova, liebe Jivka, abschließend ein besonders großes *DANKE* an dich. Du hast mir sowohl als Institutsleiterin als auch als Doktormutter stets Entfaltungsspielraum gegeben, den ich sehr wertzuschätzen weiß. Schon 2020 durfte ich im Kontext von RegioMORE deine Vision von Digitalen Ecosystemen kennenlernen – eine Vision, die mich seitdem begleitet und die ich auch ein bisschen mein Eigen machen konnte. *Danke*, dass du mir die Chance gegeben hast auf Basis deiner Vision *Engineering Digital Ecosystems* zu entwickeln.

Contents

List of Figures

List of Tables

Abbreviations

ALPS	Abstract Layered PASS
BDD	Block Definition Diagram
BPMN	Business Process Modeling Notation
DBE	Digital Business Ecosystem
DSR	Design Science Research
EDE	Engineering Digital Ecosystems
I1;I2;I3	Iteration 1; Iteration 2; Iteration 3
IDE	Integrated Development Environment
IntWertL	BMWK funded research project: IntWertL - Intelligent value networks for lightweight vehicles in small quantities (from German: Intelligente Wertschöpfungsnetzwerke für Leichtbaufahrzeuge geringer Stückzahl)
IS	Information System
JSON	Java Script Object Notation
JSON – LD	Java Script Object Notation for Linked Data
KIT	Keep It Together (In the context of Catena-X, Tractus-X)
MBSE	Model Based Systems Engineering
MVP	Minimal Viable Product

PASS	Parallel Activity Specification Schema
regex	regular expression
SBD	Subject Behavior Diagram
SDSM	Soft Design Science Methodology
SFIB – Layer	Structure-, Function-, Interaction-, Behavior-Layer
SID	Subject Interaction Diagram
SSM	Soft Systems Methodology
SWOT	Strengths, Weaknesses, Opportunities, and Threats
SysML	Systems Modeling Language
UCD	Use Case Diagram
UML	Unified Modeling Language
UUID	Universally Unique Identifier

1 Introduction

Digital Ecosystems provide a foundation for demand orientation and a resilient economy. In contrast to traditional networks, which are predominantly transaction-oriented and rely solely on economies of scale, Digital Ecosystems exhibit emergent properties by facilitating a novel form of collaboration among participants. This research study focuses on *Engineering Digital Ecosystems*. This should be understood both as Digital Ecosystems in the engineering domain and as "the creative application of scientific principles to design or develop" (Smith 2024) such ecosystems. The objective is to develop a methodology that empowers small and medium-sized enterprises (SMEs) to create and manage efficient, resilient, and adaptable industrial value creation networks. The effective design of a Digital Ecosystem can only be achieved through a thorough understanding of the corresponding problem space before initiating work on the solution space. Therefore, the Engineering Digital Ecosystems (EDE) methodology provides a framework for the design of this problem space.

First, Section 1.1 introduces the concept of Digital Ecosystems and discusses their growing significance. Section 1.2 addresses the engineering challenges related to ecosystem design, while Section 1.3 presents the objectives of this study and outlines its primary research questions. Finally, Section 1.4 concludes the chapter by detailing the structure and organization of the thesis.

1.1 Motivation

The current geopolitical landscape highlights the vulnerabilities of linear supply chains, which are primarily optimized for scale and cost efficiency. Tariffs, mobility restrictions, supply bottlenecks, rising energy prices, and volatile markets collectively signify the increasing uncertainty within the global economy. These factors are prompting companies to reevaluate their established business models regarding global supply chains and outsourced production resources. The concept of collaborative value creation within business ecosystems as a critical component of crisis management is not new and has been discussed for three decades (Moore 1996). However, technological advancements such as the Internet of Things (IoT) and advanced data analytics, including artificial intelligence (AI), have introduced new opportunities alongside their inherent complexities (Ardolino et al. 2025). The challenges faced by modern industries necessitate a shift in how companies perceive and implement value creation networks. In this context, the establishment of *Digital Ecosystems* opens up new possibilities and provides a foundation for sustainable innovation and improvement.

Digital Ecosystems extend beyond the traditional platform economy by emphasizing *coopetition*[1] rather than purely competitive dynamics (Bengtsson and Kock 2000), enabling participants, such as providers and consumers, to interact and collaborate directly to jointly develop customized offerings. Unlike classic platform businesses, where a single platform owner sets the rules for complementors and customers, Digital Ecosystems distribute governance across multiple autonomous actors (Adner 2017). This distribution allows value to emerge from interactions, such as data sharing and service composition, rather than solely from transactions channeled through a dominant player. These interactions are grounded in equal partnership and

[1] Coopetition stands for **coop**eration despite comp**etition**.

trust, reflecting shared values. This federated model is particularly beneficial for SMEs, as it enables them to leverage scale effects without sacrificing strategic control (da Silva and Marques Cardoso 2024). These networks promote flexibility, resilience, and adaptability while fostering healthy competition. By effectively shaping coopetition, Digital Ecosystems act as incubators for innovation and growth. In this way, they foster a community where participants simultaneously cooperate and compete to co-create services that no single actor could effectively build or govern alone.

1.2 Engineering Challenge

Engineering Digital Ecosystems are characterized by inherent complexity. They involve a significant number of diverse participants and stakeholders who require assistance in defining the problem space. Consequently, designing these networks entails more than merely integrating new APIs into a supply chain. It necessitates the coordination of various sovereign entities that must consistently align their data, rules, and motivations while maintaining their sovereignty. This task is particularly challenging due to three key factors:

1. **Heterogeneity:** Participants range from global corporations to two-person start-ups, engaged within various domains, each with its own vocabulary, IT stack, and governance culture.

2. **Emergence:** New interaction patterns and business models surface unpredictably once data and (micro-)services are shared.

3. **Coopetition:** Organizations must reveal just enough to collaborate on shared functionality while still competing on customer value; thus, creating a delicate tension between openness and protection.

This study proposes a model-based and subject-oriented methodology to address these challenges. In alignment with the prevalent concept of Model-Based Systems Engineering (MBSE) for the design of complex products, a model-based approach is adopted. This approach emphasizes the use of models as the primary means of information exchange, replacing traditional documents. The model-based methodology aims to enhance the efficiency, clarity, and comprehensiveness of systems engineering by facilitating the creation, analysis, management, and tracing of requirements to system elements throughout the development lifecycle using formalized modeling languages. As demonstrated by Elstermann (2020) a valuable model of complex systems, such as an Engineering Digital Ecosystem, necessitates specialized modeling methods. In this context, subject-orientation has proven to be an exceptionally effective approach. In contrast to widespread modeling approaches, this paradigm prioritizes active entities and places these so-called subjects first in consideration. This focus on subjects, rather than data (objects) or procedures (verbs), is particularly beneficial for complex, decentralized systems.

1.3 Research Objectives

The primary objective is to manage the complexity of Digital Ecosystems by providing the right information to the right people from the right perspective in an accessible manner. As Simon (1996, p. 144) states: "Solving a problem simply means representing it so as to make the solution transparent." Therefore, this study aims to develop a more effective and efficient framework for designing Digital Ecosystems, ensuring the seamless implementation of solutions for all participants. The approach seeks to foster an environment conducive to problem exploration and unrestricted ideation while facilitating emergence through transformed collaboration dynamics and enhanced knowledge exchange. The EDE methodology explicitly addresses the needs of SMEs, supporting the formation and management of efficient, resilient, and adaptable industrial ecosystems. Importantly, this approach must equally consider human-centered issues, change management, and technological solutions. Ultimately, the EDE methodology should integrate the diverse perspectives of all stakeholders into a cohesive model with shared terminology.

Addressing these objectives by integrating a model-based approach with the subject-orientation paradigm promises several benefits. However, no such methodology currently exists. Consequently, this research study focuses on two primary research questions:

RQ1 How can the design process of Digital Ecosystems be supported by a model-based methodology?

The first research question focuses on the EDE Approach. It examines the practical mechanics of transforming an abstract ecosystem vision into an engineered reality. By analyzing how model-based practices can structure decision-making, reveal interdependencies, and provide stakeholders with a common reference point, this question addresses the current lack of systematic guidance for ecosystem architects. The goal is to improve efficiency and quality. Thus, the EDE methodology can streamline the systems engineering process, reducing the time and cost associated with designing the Digital Ecosystem. Continuous validation with all stakeholders can facilitate the early identification and resolution of issues in the design process, leading to higher quality and reliability.

RQ2 What kind of subject-oriented model can be used to describe a Digital Ecosystem?

This question targets the design artifact, i. e., the model itself. Investigating which modeling constructs most effectively represent the intentions, responsibilities, and interactions within a socio-technical Digital Ecosystem addresses a persistent gap between technical diagrams and real-world collaboration patterns. In accordance with the principle of subject-orientation, special attention is given to the explicit representation of active entities, including humans, services, and organizations. Determining which constructs, abstractions, and notations best capture these facets is crucial. A well-selected model is often more precise than standard textual descriptions and serves as a unified semantic anchor for all stakeholders, including engineers, managers, and customers. Models are instrumental in visualizing complex systems, thereby fostering communication and understanding. Consequently, the EDE methodology enhances the comprehension of interactions and dependencies among participants in various usage scenarios within the Digital Ecosystem.

1.4 Outline and structure

This thesis is organized into eight main chapters. Each chapter systematically contributes to the EDE methodology by progressively developing the theoretical framework, research methodology, and evaluation techniques. Figure 1.1 illustrates the outline of the thesis and the key contributions of each chapter.

Chapter 1 *Introduction* motivates this research and establishes the primary objectives through two fundamental research questions. The chapter concludes in outlining the structure of the thesis.

Chapter 2 *Fundamentals* establishes the theoretical foundation. It begins by introducing Digital Ecosystems, focusing on core values, services, solutions, orchestrators, and participants. Next, a review of MBSE in conceptual design is presented, with an emphasis on modeling languages. This is followed by an examination of the fundamentals of subject-orientation. The chapter concludes with a brief overview of design thinking and design science frameworks.

Chapter 3 *State of the Art in Research and Application* critically surveys academic and industrial approaches to modeling Digital Ecosystems. It reviews

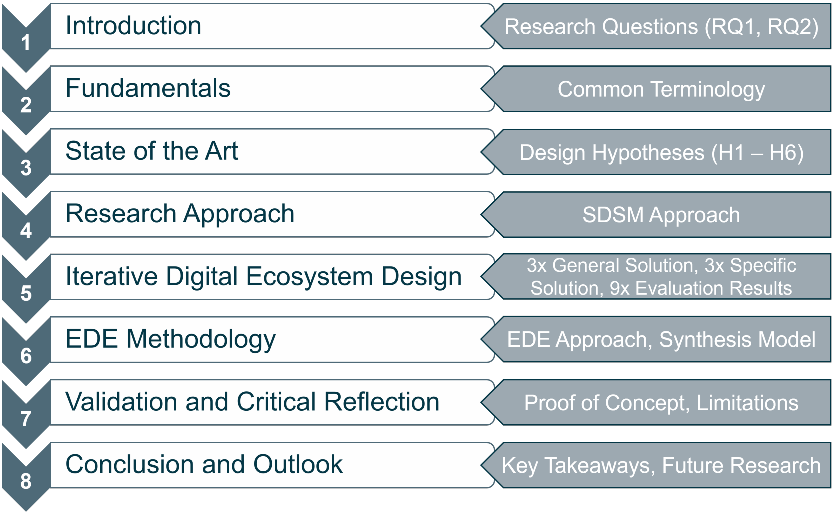

1
Introduction
Research Questions (RQ1, RQ2)
2
Fundamentals
Common Terminology
3
State of the Art
Design Hypotheses (H1 – H6)
4
Research Approach
SDSM Approach
5
Iterative Digital Ecosystem Design
3x General Solution, 3x Specific Solution, 9x Evaluation Results
6
EDE Methodology
EDE Approach, Synthesis Model
7
Validation and Critical Reflection
Proof of Concept, Limitations
8
Conclusion and Outlook
Key Takeaways, Future Research

Figure 1.1: Outline of the thesis and primary contributions of each chapter.

conceptual modeling in the literature, identifies research and industry initiatives (e. g., International Data Spaces, Gaia-X, Catena-X, Manufacturing-X), extracts research gaps, and formulates guiding design hypotheses.

Chapter 4 *Research Approach* outlines the overall research approach, which is grounded in the Design Science Research (DSR) paradigm and operationalized through the Soft Design Science Methodology (SDSM). It refines the research objectives, contextualizes the IntWertL project as a case study, and presents the iterative evaluation and model consolidation strategy.

Chapter 5 *Iterative Digital Ecosystem Design* presents the three cycles of the SDSM, employing a process-centric, structure-centric, and synthesized approach. This chapter details both the respective design artifacts and the overall design process. It encompasses thorough discussions on domain analysis, requirements definition, general solutions, specific solutions, ex ante and ex post evaluations, and the implementation of solutions.

Chapter 6 *Engineering Digital Ecosystems Methodology* consolidates the results into the EDE methodology, which serves as the primary contribution of this study. This chapter provides a comprehensive description of the EDE Approach, detailing best practices and a toolkit of recommended methods. Additionally, the Synthesis Model, along with its presentation and representation, is discussed in detail. Finally, a practical application guide is included.

Chapter 7 *Validation and Critical Reflection* first evaluates the rigorous construction of the EDE methodology and subsequently reflects on the underlying design hypotheses. It concludes with a discussion of the limitations associated with this study and the EDE methodology.

Finally, Chapter 8 *Conclusion* summarizes the research contributions, outlines potential future research directions, and presents an overview of advancing model-based Digital Ecosystem design.

With this structure, readers will first acquire theoretical knowledge of the research domain (Chapters 1–3), followed by an understanding of the research strategy (Chapter 4). They will then engage with artifact development (Chapters 5–6) and, finally, examine its validation and implications (Chapters 7–8).

2 Fundamentals

This chapter introduces the key concepts and terminology relevant to this thesis. Initially, it delineates the terminology used to describe Digital Ecosystems throughout the document. Subsequently, it provides an overview of Model-Based Systems Engineering. Following this, the paradigm of subject-orientation is discussed. Lastly, the fundamentals of Design and Design Thinking are introduced.

2.1 Fundamentals of Digital Ecosystems

Digital Ecosytems are socio-technical systems comprised of digital or technological systems, as well as organizations and people. They are characterized by the relationships and interactions among these entities and their respective compositions of CORE VALUES, CORE SERVICES, and SOLUTIONS.[1]

Based on Burkhalter et al. (2021) four archetypal roles can be distinguished. Figure 2.1 illustrates the alignment of these elements. Digital Ecosystems can be viewed as emergent systems where the interactions and interdependencies among various entities, particularly through coopetition, and their environment generate complex behaviors and properties that exceed the

[1] Small capitals are employed to denote all terms that pertain to the core principles of Digital Ecosystems. If highlighted this way, these terms are intended to be understood as defined within this chapter. This typographic choice is made to emphasize the specialized vocabulary and to assist the reader in recognizing wether a term is describing one of the key concepts central to our discussion on Digital Ecosystems or is to be understood in a broader sense.

mere sum of their individual components. Coopetition refers to a strategic collaboration between competing organizations aimed at mutually benefiting both parties, particularly within a shared ecosystem. Entities cooperate despite competition to achieve shared goals or address common challenges. Consequently, Digital Ecosystems are constructed iteratively and collaboratively, with the value creation structure designed to render the added value for all Core Participants visible as directly and promptly as possible.

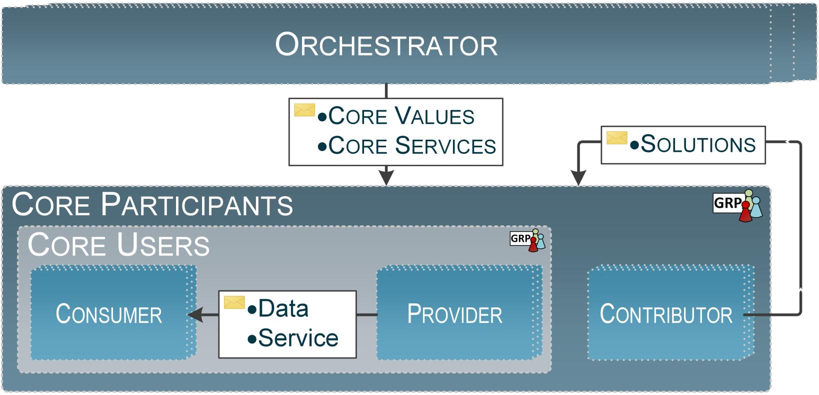


Figure 2.1: Main Elements of a Digital Ecosystem based on Burkhalter et al. (2021).

The following subsections provide definitions for these core elements of Digital Ecosystems.

2.1.1 Core Values

Definition 2.1 (CORE VALUES)

CORE VALUES are the shared strategic principles that explain *why* the ecosystem exists and prescribe *how* actors commit to behave. They codify the collective value propositions, such as interoperability, openness, or data sovereignty, and thereby establish trust, co-creation, and coopetition among all CORE PARTICIPANTS.

CORE VALUES provide the moral and strategic *north-star* of a Digital Ecosystem. Rooted in institutional theory, these elements serve as the "rules of the game", influencing acceptable behavior in a manner more sustainable than reliance on contractual governance alone. In practice, CORE VALUES exert three effects:

- **Boundary-setting:** They declare who is welcome (or excluded) without dictating business models.
- **Design orientation:** They guide architectural choices, such as decentralized identifiers to honor data sovereignty.
- **Incentive alignment:** By institutionalizing reciprocity, they lower the perceived risk of opportunism and thus catalyze network effects.

When defining CORE VALUES, the ORCHESTRATORS should ensure they are few in number, memorable, and technology-agnostic. Values that are "too specific" risk becoming outdated quickly, while those that are overly vague do not effectively limit misconduct. CORE VALUES are jointly developed by all CORE PARTICIPANTS and the ORCHESTRATORS of a Digital Ecosystem. Periodic reaffirmation, through certification schemes, steering committee votes, or community summits, maintains their legitimacy throughout the ecosystem life cycle.

2.1.2 Core Services

Definition 2.2 (Core Services)

> **Core Services** are the set of *standardized,* Orchestrator-governed functionalities and interaction patterns that collectively realize the officially defined use cases of a Digital Ecosystem. They specify the mandatory functionalities that the ecosystem Solutions must provide via uniform, non-differentiating interfaces, while leaving the details of implementation to the Contributors. At least one conformant instance of each Core Service must exist in the ecosystem; the Orchestrators may restrict further deployments.

Core Services operate at a functional rather than infrastructural level. Each service bundles a broad capability—e. g., *"Traceability"* or *"Certification Exchange"*—but its specification can be further decomposed into more detailed mechanisms or optional reference implementations supplied by the Orchestrators when necessary. By mandating shared interface contracts while allowing for diverse back-ends, Core Services achieve a balance between interoperability and implementation freedom, thereby enabling scalability.

Two broad subclasses are recognized:

- **Business Core Services:** Interaction patterns that deliver direct domain value (e. g., transaction or compliance flows) and therefore surface visibly in Core Users' Solutions.
- **Enabling Core Services:** supportive patterns (e. g., onboarding, catalog synchronization) whose presence is essential for the ecosystem's operational integrity but may stay transparent to Core Users.

Contributors package one or more Core Services into deployable Solutions. All Core Users must either consume such Solutions or host their own certified implementation. Governance artifacts—such as interface versioning, certification tests, and deployment quotas—ensure that the set of CORE SERVICES evolves coherently with the ecosystem roadmap while maintaining a level playing field.

2.1.3 Solutions

Definition 2.3 (SOLUTIONS)

> **SOLUTIONS** are deployable artifacts—applications, data services, or composite workflows—that *instantiate* one or more conformant CORE SERVICES and may incorporate additional, distinguishing functionalities. Engineered by CONTRIBUTORS and certified by the ORCHESTRATORS, they expose the uniform interfaces mandated by the referenced CORE SERVICES while delivering domain-specific value to CORE USERS.

Built on one or more standardized CORE SERVICES, a SOLUTION occupies an open solution space. This framework institutionalizes coopetition, thereby fostering innovation without fragmenting the ecosystem: multiple providers leverage identical CORE SERVICES while differentiating through domain expertise, user experience, or total cost of ownership, hence tailoring functionality to each CORE USER while remaining fully interoperable with any other conformant SOLUTIONS. From a service-dominant-logic perspective, value is co-created during the use of SOLUTIONS. Since all SOLUTIONS rely on the same interfaces, onboarding additional CORE PARTICIPANTS or scaling to new user segments requires minimal additional integration effort, which supports ecosystem scalability. Key ORCHESTRATOR levers include solution certification, revenue-sharing models, and data-usage policies that enable secondary innovation while preserving data sovereignty. Over-regulation dampens entrepreneurial activity; under-regulation erodes interoperability benefits.

2.1.4 Orchestrator

Definition 2.4 (ORCHESTRATOR)

ORCHESTRATOR denotes the actor—or consortium of actors—that designs and enforces the governance model of a Digital Ecosystem. It certifies CORE SERVICES and SOLUTIONS, defines (and may narrow) permissible business model boundaries, and administers the onboarding and offboarding of all CORE PARTICIPANTS. Although primarily a neutral steward, the ORCHESTRATOR may itself function as a CONTRIBUTOR or CORE USER when permitted by the ecosystem charter.

In practice, the ORCHESTRATOR may take the form of a single lead firm, a multi-stakeholder association, or a non-profit foundation. Its authority is structural rather than hierarchical; it cannot dictate CORE PARTICIPANTS' strategies but can accept, reject, or limit SOLUTIONS and enforce modifications to interfaces or policies. By governing rule compliance while maintaining transparency in entry pathways, the ORCHESTRATOR balances innovation with coherence, thereby supporting the scalability of the ecosystem.

2.1.5 Core Participants

Definition 2.5 (CORE PARTICIPANTS)

CORE PARTICIPANTS are entities that accept the Digital Ecosystem's CORE VALUES, implement or consume certified CORE SERVICES, and interact through SOLUTIONS. They may fluidly adopt one or several of the following roles:

- **CORE USERS** – organizations that derive direct business benefit from SOLUTIONS.

  - CONSUMER: Consumes data or service through SOLUTIONS from PROVIDERS.
  - PROVIDER: Provides data or service through SOLUTIONS to CONSUMERS.
- **Contributor** – provides SOLUTIONS to the CORE PARTICIPANTS of the Digital Ecosystem.

Role boundaries are intentionally permeable: a single firm can, and often does, switch or combine roles as strategic priorities evolve—this dynamic is essential in coopetition. Entry obligations, voting rights, and incentive schemes are defined according to each ecosystem's charter; the only universal requirement is sustained adherence to the shared CORE VALUES, interface standards, and governance decisions of the ORCHESTRATORS.

Interlinkage Summary

CORE VALUES inspire CORE SERVICES; CORE SERVICES enable competitive yet interoperable SOLUTIONS; the ORCHESTRATOR manages both and arbitrates the latter; CORE PARTICIPANTS, in their various roles, engage in coopetition surrounding these artifacts. Together, these core elements provide an analytically coherent vocabulary for engineering-centric Digital Ecosystems.

2.2 Model-Based Systems Engineering in Conceptual Design

Model-Based Systems Engineering (MBSE) has evolved into the standard paradigm for managing the increasing complexity, interdisciplinarity, and dynamism of contemporary systems—ranging from physical systems to cyber-physical platforms and socio-technical systems like *Digital Ecosystems*. Central to MBSE is the principle that an integrated, formal model serves as the single source of truth for all life-cycle activities, thereby replacing the fragile, error-prone chains of narrative documents that have historically governed systems projects (Friedenthal et al. 2014, INCOSE 2014). Such a model facilitates early validation, rigorous impact analysis, automated consistency checking, and continuous communication across stakeholder communities—capabilities that are crucial when engineering large-scale, heterogeneous ecosystems that are subject to rapid change and stringent regulatory constraints.

Within this modeling paradigm, the *Systems Modeling Language* (SysML)[2] provides a domain-agnostic notation for capturing requirements, structure, behavior, and analytical constraints in a unified repository. SysML extends and generalizes the software-oriented Unified Modeling Language (UML) by introducing the concept of *blocks*—versatile abstractions capable of representing hardware, software, data, personnel, or even intangible organizational entities—thus furnishing the fidelity and flexibility demanded by multidisciplinary engineering (Friedenthal et al. 2014).

2.2.1 Fundamental Principles of MBSE for Early-Phase Modeling

MBSE is defined as “the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” (Friedenthal et al. 2014). This approach represents a fundamental transition from document-centric practices to a model-centric paradigm in systems engineering (Friedenthal et al. 2014).

[2] Unless explicitly marked as “SysML v2”, all references to SysML in this dissertation pertain to the industrially established SysML 1.x family of specifications. (Object Management Group 2019)

Instead of disparate textual specifications, MBSE promotes an integrated system model as the primary artifact, improving consistency and enabling multi-viewpoint analysis (requirements, behavior, structure, etc.) within a single coherent framework (Friedenthal et al. 2014). The intended benefits of MBSE include enhanced specification quality, increased design reuse, and improved communication among stakeholders, as all aspects of the system are represented in an interconnected model (Friedenthal et al. 2014). INCOSE's *Systems Engineering Vision 2025* principles reinforce these concepts, anticipating that MBSE will become standard practice across industries, with models serving as the authoritative source of system knowledge from the earliest design stages (INCOSE 2014).

In the conceptual phase, MBSE techniques enable engineers to capture stakeholder needs and explore alternative solutions through abstract models, facilitating early validation of system concepts against requirements before committing to detailed design (Friedenthal et al. 2014). Notably, the use of models in early phases supports trade-off analyses (e. g., performance, cost, feasibility) by providing a rigorous means to evaluate different ideas while maintaining traceability to high-level objectives (Friedenthal et al. 2014). As the project evolves, the model can be refined in greater detail; however, the foundation established during conceptual modeling ensures that all subsequent development remains aligned with the original stakeholder goals and requirements.

2.2.2 The Role of SysML in Conceptual Design

SysML is a general-purpose modeling language developed to support MBSE throughout the system life cycle, including the conceptual design phase (Friedenthal et al. 2014). As an Object Management Group (OMG) standard (current version 1.6 as of 2019) (Object Management Group 2019), SysML provides a formal notation for representing a system's requirements, behavior, structure, and parametric constraints within a single integrated model (Delligatti 2014).

In the context of early-phase design, SysML's rich semantics enable systems engineers to create high-level representations of the system of interest and its environment without overspecifying implementation details. Key system aspects—such as stakeholder requirements, usage scenarios, and candidate architecture concepts—can be captured in various SysML diagrams and maintained consistently within the model repository (Friedenthal et al. 2014). For instance, stakeholder needs can be modeled as textual requirements, operational scenarios as use cases, and high-level architecture as abstract blocks, all interconnected to ensure traceability (Friedenthal et al. 2014).

SysML's capacity to interconnect various aspects is essential in conceptual design: changes in one view (e. g., a new stakeholder requirement) can be propagated to other views (e. g., prompting modifications of a use case or system block), ensuring consistency between the problem definition and solution concept. A well-defined SysML model effectively bridges the gap between system requirements and realization strategies, serving as a blueprint for discussion and decision-making during early trade studies (Friedenthal et al. 2014).

2.2.3 SysML Diagram Types and Their Categories

SysML defines nine types of diagrams that are illustrated in Figure 2.2, each providing a different perspective on the system model (Friedenthal et al. 2014). These diagram types can be grouped into four categories – *Structure*, *Behavior*, *Requirements*, and *Auxiliary* – according to the aspect of the system they primarily represent. Below, every SysML diagram type is enumerated and classified by category (with brief descriptions) (Friedenthal et al. 2014, Object Management Group 2019):

- **Package Diagram** – *Auxiliary (Structure)*: Organizes the model into packages (hierarchical containers) and sub-packages, showing the structure of the model itself and how model elements are grouped

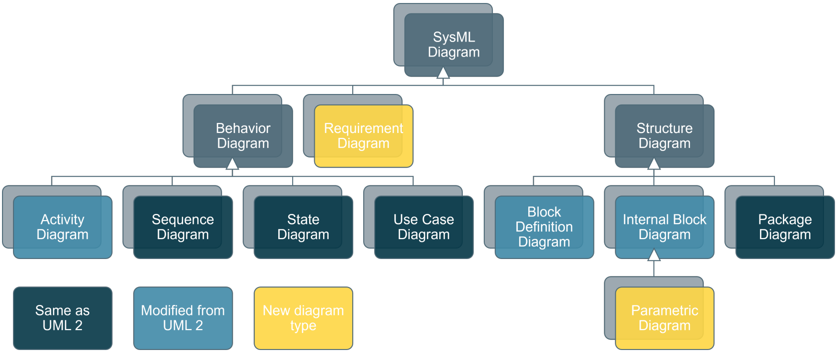


Figure 2.2: SysML Diagram Taxonomy based on (Hause 2006, p. 3).

(adopted from UML Package Diagram) (Friedenthal et al. 2014). Package diagrams do not represent the system's content directly but provide a way to structure the model for manageability.

- **Requirement Diagram** – *Requirements*: Depicts textual requirements (e. g., functional, performance, or constraint requirements) and their relationships to other requirements or to design elements (via «derive», «satisfy», «verify», «refine» dependencies) (Friedenthal et al. 2014). This diagram is unique to SysML (no direct UML counterpart) and is used for requirements capture and traceability management.

- **Use Case Diagram** – *Behavior*: Illustrates the functionality of the system as perceived by external actors (users or other systems) (Friedenthal et al. 2014). It shows actors (stick figures or role icons) and use cases (ovals) along with their associations, giving a high-level view of the system's intended uses or services (equivalent to UML Use Case Diagram).

- **Activity Diagram** – *Behavior*: Represents flow-oriented behavior, specifying sequences of actions, control flows (decision, fork/merge), and object flows (Friedenthal et al. 2014). An activity diagram models the logic of complex operations or scenarios, showing how inputs are transformed to outputs through a controlled sequence (adapted from UML Activity Diagram).

- **Sequence Diagram** – *Behavior*: Describes interactions in terms of the sequence of messages exchanged between parts (or actors) over time (Friedenthal et al. 2014). It is useful for elaborating specific scenarios or use case instances, detailing the temporal order of events (same as UML Sequence Diagram in SysML).

- **State Machine Diagram** – *Behavior*: Describes the lifecycle behavior of an individual block or entity in terms of states and state transitions triggered by events (Friedenthal et al. 2014). A state machine diagram captures how an object responds to stimuli over time (same as UML Statechart/State Machine Diagram).

- **Block Definition Diagram (BDD)** – *Structure*: Shows the system's static structural elements (blocks) and relationships such as composition (whole-part hierarchies), associations, and generalizations (Friedenthal et al. 2014). A BDD is analogous to a UML class diagram and is used to define the types of components in the system and how they relate to one another (e. g., system breakdown structure).

- **Internal Block Diagram (IBD)** – *Structure*: Zooms into a particular block to show its internal structure, specifically how its parts (sub-blocks) are interconnected via ports and connectors (Friedenthal et al. 2014). The IBD is based on UML composite structure diagrams and is typically used after defining blocks on a BDD, to specify the internal configuration of a system or subsystem.

- **Parametric Diagram** – *Auxiliary*: Expresses constraints among properties (parameters) of blocks using constraint equations, supporting engineering analyses such as performance or trade studies (Friedenthal et al. 2014). Parametric diagrams (unique to SysML) do not define new structure or behavior per se; rather, they overlay mathematical relationships (e. g., $F = m \times a$) onto the structural model, enabling calculation and simulation of system metrics.

Each of these diagram types plays a distinct role in a SysML model, enabling a comprehensive specification of system requirements, structure, behavior, and constraints. Notably, all diagram types are consistent with the underlying SysML model: elements depicted in one diagram (e. g., a block or an actor) may also appear in other diagrams with different relationships, ensuring a unified system representation (Friedenthal et al. 2014). The designation of "auxiliary" for Package and Parametric diagrams indicates that these are supportive views (representing model organization and analytical constraints, respectively), complementing the core views of requirements, structure, and behavior.

2.2.4 Block Definition Diagram (BDD): Purpose, Notation, Strengths, Limitations

The Block Definition Diagram (BDD) is the primary structural view in SysML. Its purpose is to define the system's structure in terms of blocks (abstract system components) and to illustrate the relationships between these blocks (Friedenthal et al. 2014). A "block" in SysML is a generalized concept that can represent any entity with structure—such as a system, subsystem, hardware component, software element, actor, or even an abstract conceptual element—that the modeler chooses to regard as a discrete part of the system architecture (Friedenthal et al. 2014). Figure 2.3 illustrates the main elements of a BDD.

In notation, a block is represented as a rectangle with the block's name and optional compartments for features (values, operations, etc.), similar to a UML class. BDDs typically include blocks connected by different types of lines: a simple solid line indicates association (some structural relationship), a solid line with a hollow diamond denotes aggregation (whole-independent part), a solid line with a filled diamond signifies composition

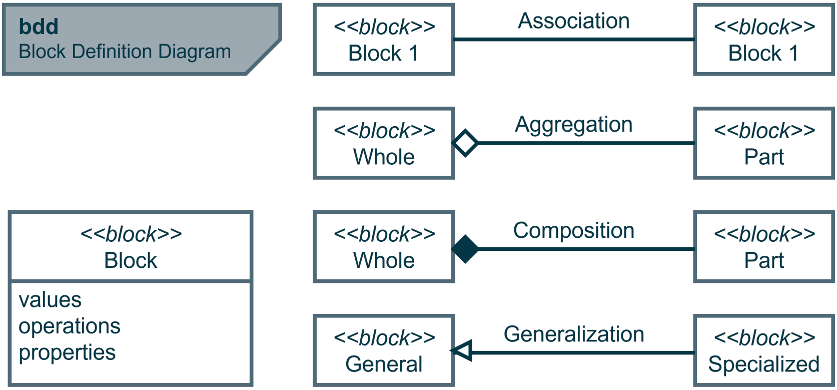


Figure 2.3: SysML notation for Block Definition Diagram (bdd).

(whole–dependent part), and a hollow triangle arrow represents generalization/specialization (inheritance) (Friedenthal et al. 2014). These notational elements enable the modeler to clearly represent system hierarchies and taxonomies. For example, a BDD might illustrate a top-level `System` block composed of several `Subsystem` blocks, or a `Vehicle` block generalizing a more specific `Car` block and `Truck` block, etc. The BDD does not depict how parts connect or interact, but it excels at presenting the "big picture" of static structure: what the components of the system are and how they are conceptually related (Friedenthal et al. 2014).

BDD strengths for conceptual work are threefold. First, they define the *system-of-interest* and its environment within a single contextual view, clarifying scope and external interfaces early (Friedenthal et al. 2014). Second, top-level decomposition illustrates major subsystems and their relationships without committing to low-level design, giving stakeholders an immediate understanding of the candidate architecture and a foundation for later refinement. Third, the diagram's UML-derived notation provides a precise yet widely recognized vocabulary, facilitating cross-disciplinary communication and enabling rapid checks for completeness, naming consistency, and responsibility allocation across the model (Friedenthal et al. 2014).

These benefits are balanced by notable limitations. A BDD is purely static: it conveys neither temporal behavior nor interaction sequences. Consequently, structural elements must be linked to behavioral and requirement views to justify their inclusion and to avoid functionless "orphan" blocks. Overly detailed early models pose a second risk: excessive properties or fine-grained components obscure the architectural picture and reduce readability. Best practice is to model "just enough" structure and defer technical details to later iterations or to IBDs (Delligatti 2014). Moreover, since requirements are not inherent to the notation, explicit «satisfy» traces are crucial; without them, the architecture may deviate from stakeholder intent (Friedenthal et al. 2014). Lastly, since blocks can represent any structured entity, the model element itself cannot be distinctly classified as either an active or passive component.

2.2.5 Use Case Diagram (UCD): Purpose, Notation, Strengths, Limitations

Use Case Diagrams (UCD) in SysML (inherited from UML) provide a high-level visualization of a system's intended functionality and its interactions with external entities (actors). The primary purpose of a UCD is to capture the goals or services that the system offers to external users or stakeholders, in a manner that is easy to understand and grounded in real-world usage scenarios (Friedenthal et al. 2014). Figure 2.4 illustrates the main elements of a UCD.

Notationally, a UCD consists of actors, typically represented as stick figures (for human users) or labeled icons (for external systems or other interacting entities), and use cases, depicted as ellipses (ovals) usually enclosed within the system boundary. The system boundary can be illustrated as a rectangle surrounding the use cases, with the system name included, indicating which use cases belong to the system of interest. Actors lie outside this boundary and are connected to use cases by associations (solid lines), indicating that

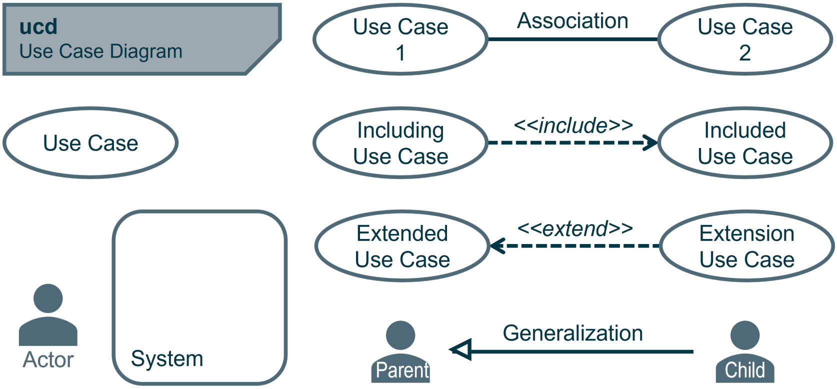


Figure 2.4: SysML notation for Use Case Diagram (ucd).

the actor participates in those use case scenarios (Friedenthal et al. 2014). Each use case represents a set of scenarios or a capability that yields an observable result of value to an actor. Use cases can include or be extended by other use cases. Generalization between actors can also be depicted in UCDs.

UCDs illustrate the system's capabilities and external interactions using an actor-goal vocabulary that is accessible to non-technical stakeholders. Case names expressed in verb-noun form provide an immediate understanding of functionality, making the notation effective for eliciting and validating high-level requirements (Friedenthal et al. 2014). Each identified case serves as a container for detailed behavior and as an anchor for assigning responsibilities to structural elements, thereby maintaining traceability from stakeholder intent to architecture.

1. **Boundary definition:** Actors are placed outside the system context, clarifying external interfaces.
2. **Stakeholder communication:** Minimal notation expresses operational goals without modeling jargon.

3. **Analysis driver:** Every case can be refined into activity, sequence, or state diagrams and linked to blocks via «allocate» and to textual requirements via «refine».

The descriptive nature of diagrams imposes certain constraints. Without elaboration, event sequences and data flows remain unspecified, leading to potential functional gaps. Excessive granularity blurs user goals with technical tasks, while insufficient detail obscures alternative or exceptional paths. Moreover, UCDs exclude requirement objects, making explicit «refine»/«satisfy» relations essential to prevent divergence between user intent and design. Finally, accurately scoping actors and cases is non-trivial in complex systems; ambiguous or overlapping definitions lead to inconsistencies. Therefore, UCDs must remain high-level, be rigorously traced, and be complemented by behavioral and structural views to realize their full value in conceptual MBSE.

2.2.6 Integrating BDDs and UCDs in Conceptual MBSE

UCDs express *external purpose* (actor goals), whereas BDDs encode *internal structure* (blocks and their composition). Joint utilization provides a comprehensive overview during the early phases of development: the use-case boundary delineates the scope, while the BDD indicates the architecture required to fulfill that scope. Consistency is maintained through three traceable mappings (Friedenthal et al. 2014):

1. **Actor–Block Alignment:** Each actor maps to a context block (often "actor") and each system boundary maps to a top-level block, clarifying interfaces.
2. **Behavior Allocation:** Every use case is refined by behavior (activity, sequence, state); actions therein are "allocated" to blocks, proving structural support for all functions.

3. **Requirement Flow:** Textual requirements are "refined" by use cases and "satisfied" by blocks.

Best-practice guidelines: Keep use cases at stakeholder goal level and BDDs at a coarse “black-box” decomposition; premature fine detail in either view creates misalignment. Maintain bidirectional reviews—“Which block realizes this step?” vs. “Which use case justifies this block?”—to expose omissions, over-engineering, or orphan structure (Friedenthal et al. 2014).

Pitfalls to avoid:

- Unallocated use cases or unsatisfied requirements signal functional gaps.
- Excessive granularity obscures architecture and hinders change impact analysis.
- Divergent abstraction levels impede traceability.

By providing a standardized and tool-supported approach to visualize and analyze systems at a high level, SysML plays a critical role in conceptual design. It enables early validation of system concepts, ensures requirements traceability, and supports a smooth transition from abstract ideas to concrete designs (INCOSE 2014). In summary, SysML serves as the lingua franca of MBSE during the conceptual phase, ensuring that diverse stakeholders—customers, engineers from various disciplines, and management—maintain a shared and precise understanding of the emerging system architecture and its intended behavior (Friedenthal et al. 2014).

2.3 Subject-Orientation

The modeling paradigm of subject-orientation originates from the context of distributed systems and distributed information processing, initially introduced by Fleischmann (1994). Given that Digital Ecosystems are classified as distributed systems, the application of subject-orientation is a natural fit. The foundational concepts of subject-orientation were further elaborated by Fleischmann et al. (2011). Based on this and other relevant literature, Elstermann defines subject-orientation as follows:

> **Subject-Orientation** – a modeling or description paradigm for processes that is derived from the structure of natural languages. It requires the explicit and continuous consideration of active entities within the bounds of a process as the conceptual center of description. Active entities (subjects) and passive elements (objects) must always be distinguished, and activities or tasks can only be described in the context of a subject. The interaction between subjects is of particular importance and must explicitly be described as an exchange of information that cannot be omitted. (Elstermann 2020, p. 80)

In other words, subject-oriented models reflect the fundamental subject-verb-object structure of natural language sentences. This approach advocates for modeling processes in terms of *who* (subject) performing *what* (action) on *whom/what* (object), thereby making the communication between the "who"s explicit. Since this thesis is not limited to process models alone, we will utilize an adapted definition:

Definition 2.6 (Subject-Orientation)

Subject-Orientation is a modeling or description paradigm derived from natural language structure. It requires the explicit and continuous treatment of active entities (subjects) as the conceptual center;

passive elements (objects) must always be distinguished. Every activity is described within the context of a subject, and interactions between subjects are expressed as explicit information exchanges that cannot be omitted. (Adapted from Elstermann (2020, p. 80))

The core of subject-orientation is thus a reference to natural language structures, specifically constructs of subject, predicate, and object. Here, the subject stands as the central actor within this descriptive logic. These described subjects are not a novel invention. In other modeling paradigms, such as procedural or object-oriented approaches, they often exist but are only implicitly represented (Fleischmann et al. 2011). As Schmidt et al. (2009) illustrate, the consideration of subjects is possible across various modeling languages. However, in practice, it is common for only certain aspects of subject-orientation to be applied in modeling, often due to the use of modeling languages that do not fully support subject-orientation. In these instances, the approach is referred to as *subject-based modeling* rather than subject-orientation (Elstermann 2020). For this concept, Elstermann (2020) provides the following definition which is adopted for this thesis:

Definition 2.7 (Subject-Based Modeling)

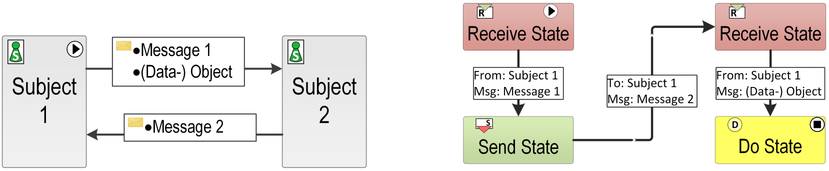
Subject-Based Modeling – A definition for process models that follow the broad principles of subject-orientation, but without embracing a modeling notation that enforces the paradigm and therefore lacks the possibility of automation tool support and requires strict paradigm adherence by a modeler without the corresponding tool support. (Elstermann 2020, p. 83)

The only process modeling language that fully incorporates all fundamental elements of natural language (subject, object, predicate) is the Parallel Activity Specification Scheme (PASS) (Schmidt et al. 2009) which is elaborated upon in the subsequent section.

2.3.1 The Parallel Activity Specification Scheme (PASS)

PASS was originally introduced by Fleischmann (1994) and further detailed in Fleischmann et al. (2011). PASS remains the only fully executable, Turing-complete modeling language specifically designed to comprehensively support subject-orientation. PASS is fully specified by an open-source Web Ontology Language (OWL) description and corresponding documentation.[3] The upcoming section introduces the core structure of PASS and the most important modeling elements. This discussion is primarily grounded in the works of Fleischmann et al. (2011) and Elstermann (2020), with the structure and descriptions closely following Elstermann's contributions.

PASS fundamentally distinguishes between two types of diagrams: the *Subject Interaction Diagram* (SID) and the *Subject Behavior Diagram* (SBD), both of which are illustrated in Figure 2.5. The SID represents the relationships between active entities, referred to as subjects, while the corresponding SBDs model the specific behaviors of the subjects depicted in the SID. Thus, the complete model of a described process consists of one SID and one or more SBDs. Standard PASS is based on five core elements: subjects and messages within SIDs, and send states, receive states, and function states, along with their associated transitions within SBDs.



(a) Subject Interaction Diagram (SID)

(b) Subject Behavior Diagram (SBD)

Figure 2.5: Example of a PASS model.

[3] The Standard-PASS-Ontology is available under `https://github.com/I2PM/Standard-PASS-Ontology` and the corresponding PASS-Standard-Book-Project is available under `https://github.com/I2PM/PASS-Standard-Book-Tex-Project`.

Each subject in PASS can be regarded as an independent agent (human or machine) with its own behavior, responsible for sending and receiving messages to synchronize with other subjects. This design ensures a clear separation of concerns: each participant's perspective is modeled independently, with their only connection being through message flows. This approach significantly contrasts with conventional process modeling techniques such as BPMN (Business Process Model and Notation), which primarily emphasize the sequence of activities and control flow, with "actors" typically represented informally through swimlane partitions.

2.3.1.1 Subject Interaction Diagram (SID)

As the name suggests, an SID illustrates all interactions between subjects within a specified process. Each SID consists of two types of elements: subjects and messages. Comprehensive descriptions of the elements in an SID are presented in Table 2.1.

The type and number of subjects in an SID is a critical design decision. Effective subject delineation requires careful consideration of the intended division of labor within the process. High granularity introduces numerous specialized subjects and messages, increasing complexity and the potential for misunderstandings or delays. Conversely, too few subjects may lead to broad task coverage by each entity, potentially exceeding their functional scope or authority. Achieving the right balance in subject delineation is essential for the model's clarity and utility. (Fleischmann et al. 2011)

2.3.1.2 Subject Behavior Diagram (SBD)

The second type of diagram in PASS is the SBD. For each fully specified subject in a process's SID, an SBD exists. This diagram models the possible

Table 2.1: Elements of an SID based on Elstermann (2020, p. 85)

Element	Description
Subjects	Subjects define the active entities and roles within a process. Each fully described subject has an individual SBD that depicts the subject's internal workflow within the process.
Interface Subjects	These are subjects whose behavior is not described (represented as a "black box"). This lack of description may result from insufficient knowledge about the behavior or because the behavior is considered too insignificant to model. Alternatively, an interface subject can serve as a placeholder for a process described in a different model.
Multi-Subjects	Multi-Subjects can be instantiated multiple times during a process, indicating that various actors within the process exhibit the same behavior. Multi-Subjects can be fully specified or interface subjects.
Messages	Messages exchanged between subjects define the ways in which a sending subject can convey information (information objects) to a receiving subject.

sequence of actions (i. e., the behavior) of a subject. The subject's behavior provides insights into the causal and/or temporal progression of the described process. Generally, PASS distinguishes between *action behavior* and *communication behavior*. Action behavior describes internal functions that can be performed independently of other subjects, in any order and frequency, represented in PASS through function states and function transitions. Communication behavior, on the other hand, involves all actions related to message exchanges and includes send states, send transitions, receive states, and receive transitions in PASS.

As implied, behavior in PASS is characterized by the condition that a subject is always in a specific state, which dictates the tasks the subject performs.

When the exit conditions of a state are met—and only then—the subject may proceed to the next state via the designated transition. According to this logic, each SBD has a singular defined start state where the behavior initiates. It is feasible for receive and function states to possess multiple exit conditions (XOR connections), enabling branching and potentially resulting in multiple end states.

PASS also supports user- or time-dependent exit transitions for communication and function states. A *user-cancel transition* describes situations where a state is exited due to an unspecified user decision, even if the defined exit condition has not been satisfied. There are two types of time transitions: those that specify a duration within a state (similar to a timer) and those that trigger at designated intervals (similar to a reminder). Table 2.2 summarizes the elements of an SBD.

As with subject delineation, a process can be segmented into as many states as necessary to achieve the desired level of detail. For a simpler and more comprehensible representation, multiple states may be consolidated. In an SID, other subjects primarily focus on communication behavior rather than action behavior; therefore, sequences of functional states can be grouped into a single overarching functional state. This approach provides flexibility in describing both communication and action behaviors. To ensure a clear and accurate assignment of tasks to subjects, an appropriate level of detail must be determined. The delineation of subjects ultimately defines the specific set of tasks that each SBD must model. (Fleischmann et al. 2011)

2.4 Design Foundations: From Design Thinking to Engineering

Design is fundamentally a human-centered problem-solving process that integrates creativity, research, and iterative development to provide meaningful solutions. It is applicable across various domains, including product

Table 2.2: Elements of an SBD based on Elstermann (2020, p. 88)

Element	Description
Send State	The state in which information (information objects) is sent to another subject. The send transition specifies the receiving subject and the message to be sent. The state ends once the message has been dispatched. Unlike other states, a send state has only one possible send transition (one exit).
Receive State	This state requires input from another subject before continuing behavior. The receive transition specifies the sending subject and the received message. The state ends upon receiving the message. Receive states can have multiple (alternative) receive transitions.
Function State	This describes a state in which a subject performs a task independently of other subjects. Function transitions specify possible outcomes necessary to exit the state. Function states can have multiple (alternative) function transitions.
Start/End State	Any type of state can be the single start state for the behavior. However, only receive and function states can be one of (possibly) multiple end states.
User-Cancel Transition	This marks the exit from a state (primarily for receive states) due to an arbitrary user decision without meeting any predefined exit condition.
Time Transition	This enables a state to be exited based on a time event, akin to receiving a message from an implicit alarm or calendar subject. Two types of time transitions are distinguished: (1) transitions defining a duration within the state (like a timer) and (2) transitions triggered at a defined frequency (like a reminder).

development, services, systems, and digital experiences, while consistently striving to balance user needs, technical feasibility, and business objectives. At its core, design involves a deep understanding of challenges, thorough exploration of potential solutions, and refinement of the most promising ideas through testing and feedback. This process is non-linear, encompassing continuous cycles of discovery, ideation, and iteration.

Design research focuses on the intentional creation of *artificial* artifacts, services, or socio-technical systems that transition the world from its current state to a preferred one (Simon 1996). Contemporary scholarship perceives design as (i) a human-centered investigation of what ought to be, and (ii) an interventionist practice that transforms insights into sustainable change. This section is structured around three complementary conceptions: *design thinking*, the *Double Diamond*, and Wieringa's *Design Cycle*.

2.4.1 Design Thinking

Originating from the reflective practices of professional designers and popularized by institutions such as Stanford d.school and IDEO, design thinking is an abductive reasoning approach that integrates empathy, ideation, and rapid prototyping to address ill-defined ("wicked") problems (Buchanan 1992, Brown 2008). Its theoretical foundation prioritizes:

- **Desirability:** deep understanding of human motives and emotions,
- **Feasibility:** awareness of technological opportunities and constraints,
- **Viability:** alignment with organizational and societal economics.

Thus, design thinking functions as a mindset, a toolbox, and a socio-cognitive process that integrates divergent exploration with convergent decision-making.

2.4.2 The Double Diamond Framework

To illustrate the oscillation between divergence and convergence, the UK Design Council introduced the **Double Diamond** model in 2005. It consists of four quadrants as illustrated in Figure 2.6:

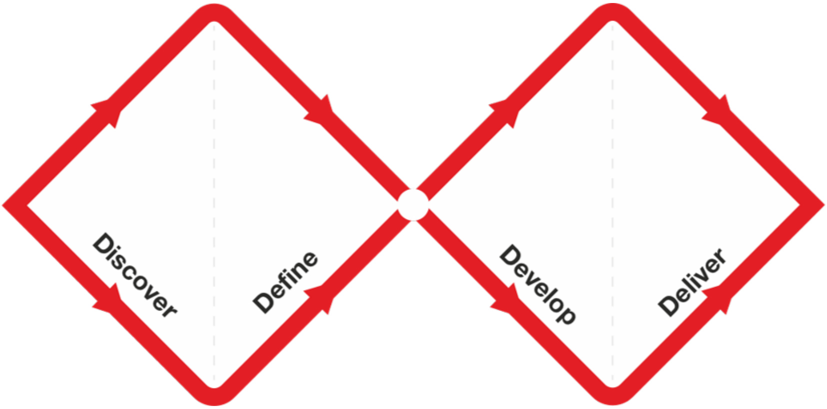


Figure 2.6: The design thinking Double Diamond (Design Council, 2005).

1. **Discover**: This initial phase focuses on researching the context and uncovering insights. Designers explore the problem space broadly, engaging with users and stakeholders to understand their experiences, needs, and pain points.

2. **Define**: After gathering insights, this phase involves synthesizing the findings to clearly frame the core problem. This stage helps teams align on a focused design challenge or problem statement.

3. **Develop**: In the second diamond, designers ideate, prototype, and experiment with a range of possible solutions. This phase encourages creative exploration and collaborative thinking.

4. **Deliver**: The final phase involves testing and refining solutions, eventually implementing the most viable and effective design. This is where final validation and deployment occur.

Its strength resides in offering a minimal yet extensible framework: specific methods (e. g., interviews, journey maps, experiments) can be integrated, while the macro structure safeguards against premature problem fixation or solution lock-in. Consequently, the Double Diamond is widely employed not only in traditional design disciplines but also in service design, policy innovation, and digital transformation initiatives.

2.4.3 The Design Cycle in Design Science Research

While the Double Diamond emerged from practice, Wieringa (2014) formalized a corresponding *Design Cycle* as part of the more comprehensive *Engineering Cycle* for rigorous Design Science Research (DSR). The design cycle corresponds to the first three tasks of the engineering cycle, that is illustrated in Figure 2.7. The engineering cycle comprises:

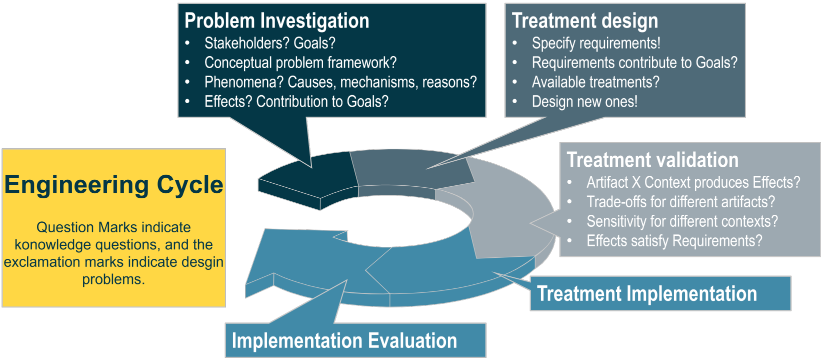

Problem Investigation
• Stakeholders? Goals?
• Conceptual problem framework?
• Phenomena? Causes, mechanisms, reasons?
• Effects? Contribution to Goals?
Treatment design
• Specify requirements!
• Requirements contribute to Goals?
• Available treatments?
• Design new ones!
Engineering Cycle
Question Marks indicate konowledge questions, and the exclamation marks indicate desgin problems.
Treatment validation
• Artifact X Context produces Effects?
• Trade-offs for different artifacts?
• Sensitivity for different contexts?
• Effects satisfy Requirements?
Treatment Implementation
Implementation Evaluation

Figure 2.7: The Engineering Cycle based on Wieringa (2014, p. 28).

1. **Problem Investigation:** What needs to be improved and why?

2. **Treatment Design:** Design as an artifact one or more possible treatments for the problem.
3. **Treatment Validation:** Is the design artifact a solution to the problem?
4. **Treatment Implementation:** Apply the artifact to the problem domain.
5. **Implementation Evaluation:** Is the artifact successful in treating the problem?
   The Evaluation may lead into another iteration through the engineering cycle. (Wieringa 2014)

Iteration and learning loops connect these steps, ensuring that knowledge contributions (design principles, artifact theories) are systematically accumulated.

2.4.4 Design *versus* Engineering

In DSR, *design* and *engineering* are complementary yet epistemically distinct activities. Design addresses the *normative* question "*What should exist?*" - it reframes ambiguous situations, challenges boundary conditions, and values novelty, empathy, and contextual fit. Engineering addresses the *analytic* question "*How can it reliably exist?*" - it operationalizes scientific laws and formal methods to ensure performance, safety, and efficiency within established constraints.

Hence, while engineers prioritize correctness, designers focus on relevance and meaning. The Double Diamond and Wieringa's Design Cycle demonstrate that sophisticated design work integrates both logics: it *creates* new options (design) and *realizes* them with technical rigor (engineering). A dissertation in design science must therefore showcase mastery of this dual competence by articulating clear problem framings, contributions of artifacts, and empirically validated outcomes.

2.5 Summary

This chapter defines all substantive terms in the dissertation title to establish a common terminology for the subsequent sections of this thesis.

First, we established the term *Digital Ecosystem* and its core elements: CORE VALUES drive CORE SERVICES. The ORCHESTRATORS manage both and mediates the creation of competitive and interoperable SOLUTIONS. On the other hand, CORE PARTICIPANTS are role fluid and engage in coopetition while using or contributing to SOLUTIONS. These elements collectively provide a cohesive vocabulary for engineering-focused Digital Ecosystems.

Second, we elucidated the adjective *model-based*. Model-Based Systems Engineering (MBSE) and the Systems Modeling Language (SysML) are introduced, along with a detailed description of the diagram types that will be recurrent throughout the thesis, specifically the Block Definition Diagram (BDD) and the Use Case Diagram (UCD).

Third, we unfolded the qualifier *subject-oriented* by introducing the corresponding paradigm and its concrete realization in the Parallel Activity Specification Scheme (PASS). This allows for the expression of distributed behavior through interacting *subjects* whose semantics closely align with the roles in natural language while remaining formally executable.

Finally, we examined *design* itself, bridging human-centered Design Thinking, the divergence-convergence rhythm of the Double Diamond model, and the rigor of Design Science Research (DSR). This emphasizes that engineering artifacts to conceptualize Digital Ecosystems represent both creative and research-intensive endeavors.

Together, these four lenses establish the theoretical foundation for *model-based and subject-oriented design of Digital Ecosystems* in the subsequent chapters. Next, the State of the Art (SotA) of existing research and industrial applications is presented. This involves an analysis of current methods, identification of gaps, and extraction of design hypotheses.

3 State of the Art in Research and Application

Engineering Digital Ecosystems are complex networks of organizations, platforms, and technologies that collaboratively share data and services. They promise seamless data exchange across company boundaries, enabling new efficiencies and business models in Industry 4.0. However, designing such ecosystems is challenging due to the need for data sovereignty, interoperability, and trust among independent participants (Ishihara and Matsutsuka 2025). Traditional system design methods often fall short in these multi-organizational settings. Nevertheless, Engineering Digital Ecosystems are rapidly becoming a strategic imperative for the manufacturing and related engineering sectors. This chapter introduces the current state of the art in research and industrial applications related to Digital Ecosystems, emphasizing their conceptual design rather than technical implementation.

The academic landscape has witnessed a significant increase in publications related to Engineering Digital Ecosystems in recent years (Shahatha Al-Mashhadani et al. 2021, Kaiser et al. 2023, Coskun-Setirek et al. 2024). However, many of these works pertain to specific SOLUTIONS (Shahatha Al-Mashhadani et al. 2021) or technical reference architectures (Kaiser et al. 2023). Nevertheless, two recent meta-reviews (Tsai et al. 2022, Coskun-Setirek et al. 2024) offer a comprehensive overview of the conceptual modeling of Digital Ecosystems, thereby establishing a solid foundation.

Comparatively, real-world applications demonstrate significant momentum through industrial initiatives such as *Gaia-X*, *Catena-X*, and *Manufacturing-X*. However, they also reveal methodological blind spots that academic research has yet to address. By integrating insights from these two domains, this chapter establishes the foundation for a gap analysis and for formulating design hypotheses related to the core research questions of the dissertation:

RQ1 How can the design process of Digital Ecosystems be supported by a *model-based methodology*?

RQ2 What kind of *subject-oriented model* can be used to describe a Digital Ecosystem?

The remainder of the chapter is structured as follows: Section 3.1 reviews the current academic SotA, while Section 3.2 analyzes industrial progress in engineering and manufacturing. Based on this, Section 3.3 synthesizes the identified research gaps in light of RQ1 and RQ2. Finally, Section 3.4 derives focused design hypotheses from the SotA, and Section 3.5 concludes the chapter.

3.1 Academic State of the Art

Although ecosystem thinking is gaining traction in engineering, the initial scoping review indicates that peer-reviewed studies explicitly addressing manufacturing or production contexts are still sparse and scattered across specialized venues. In this context, Coskun-Setirek et al. (2024) argue that data on Digital Ecosystem architecture is insufficient to infer differences between industries. To establish a sound theoretical basis, this chapter therefore consults the broader research stream on *Digital Business Ecosystems* (DBEs). The DBE concept extends the business-ecosystem metaphor introduced by Moore (1996), who described co-evolving constellations of firms around a keystone organization, to socio-technical configurations orchestrated through digital platforms.

DBE scholars "define DBE as a socio-technical environment of individuals, organisations and digital technologies with collaborative and competitive relationships to co-create value through shared digital platforms." (Senyo et al. 2019, p. 2) Given that engineering and non-engineering firms face numerous analogous challenges—including multi-actor coordination, data sovereignty, and dynamic governance—the DBE body of knowledge remains relevant, even in contexts where primary studies in purely engineering settings are limited.

Senyo et al. (2019) argue that the distinctive characteristics of DBEs render many existing information systems artifacts unsuitable. They advocate for dedicated frameworks, models, and methodologies that can be empirically validated. In their view, more DBE-specific artifacts are needed to provide a solid foundation for future studies and should be tested with real data. The present review therefore turns next to present conceptual modeling approaches for DBE.

3.1.1 Conceptual modeling of DBE

Conceptual models serve as essential analytical tools in ecosystem design, providing an abstract, multi-perspective representation of actors, interactions, data artifacts, and supporting infrastructure. Two recent systematic literature reviews investigate the notations and viewpoints employed in Digital Ecosystem studies. Tsai et al. (2022) review 94 publications up to 2021, while Coskun-Setirek et al. (2024) analyze 64 architecture papers. Table 3.1 summarizes the counts of modeling notations reported by both studies.

The combined evidence indicates three tendencies:

1. **Only a limited range of formal languages is frequently utilized:** UML dominates both corpora (seven plus one (SysML) papers in Tsai et al.

Table 3.1: Use of modeling notations in two systematic reviews

Modeling notation / family	**Tsai et al. (2022)**	**Coskun-Setirek et al. (2024)**
UML (all diagram types)	7	9
System Modeling Language (SysML)	1	0
BPMN	4	4
e^3-value	4	0
Matrix-based representations	4	0
Ontology-based formalisms	4	0
Social-network graphs	4	0
ArchiMate	2	1
Canvas-based templates	3	0
Enterprise-modeling languages	3	0
DBE-specific languages	2	0
Flow charts	0	2
IDEF (e.g. IDEF0)	0	2
Petri nets	0	1
Unspecified / informal (boxes and lines)	19	44
Total studies in review	94	64

(2022); nine in Coskun-Setirek et al. (2024)), followed by BPMN. Specialized ecosystem languages—such as e^3-value, matrix-based formalisms, or DBE-specific notations—collectively appear in fewer than one-tenth of all papers.

2. **Informal box-and-line sketches remain the default choice:** 19 studies in Tsai et al. (2022) and 44 in Coskun-Setirek et al. (2024) rely on unspecified diagramming conventions, which undermines precision and reusability.

3. **Most research addresses a single viewpoint:** only 23% of papers in Coskun-Setirek et al. (2024) integrate multiple viewpoints. Typically this viewpoint encompasses the process or business perspective. In these studies, the notion of actor/role is the most prevalent element, as 87% of studies in Tsai et al. (2022) focus on this perspective.

Tsai et al. (2022) therefore conclude that few contributions provide a holistic modeling method consisting of notation, meta-model, and procedure. Coskun-Setirek et al. (2024) reach a similar conclusion, advocating for additional viewpoints (e. g., data, governance, evolution) and for more formal yet practitioner-friendly notations. In summary, the literature offers little systematic guidance on how to capture the full socio-technical breadth of a DBE—especially in engineering domains where precise specification of data flows, capabilities, and compliance constraints is critical. This gap underpins the objective of the present study, which is to develop a model-based, subject-oriented methodology that unifies actor, interaction, and value perspectives while remaining extensible to additional concerns such as data sovereignty and governance policies.

3.1.2 Implications for the research questions

The academic landscape highlights several research gaps related to the objectives of this study. The most relevant gaps are listed below, with denotations indicating the corresponding research questions (RQs).

Need for a design methodology (RQ1): Only fifteen percent of the studies surveyed by Tsai et al. (2022) include any type of stepwise procedure for modeling a DBE, and even these primarily focus on scoping, and actor identification or interaction capture rather than a fully integrated method. None of these studies combine their procedures with complementary methods such as requirements engineering or enterprise architecture analysis. The review, therefore, advocates for a systematic, multi-perspective methodology that can balance detail with holism and effectively manage ecosystem resilience. Although Coskun-Setirek et al. (2024) concentrate on DBE architecture, they also emphasize the need for means to support DBE design. Addressing this gap is central to RQ1, which seeks a design approach explicitly tailored to the complexity of Digital Ecosystems.

Model-based engineering versus informal sketches (RQ1): Both reviews note a significant reliance on informal box-and-line diagrams and ad-hoc layered illustrations; formal notations remain exceptions (Coskun-Setirek et al. 2024). This prevalence of informal artifacts undermines precision, reusability, and machine processability, leaving practitioners without an executable reference model. Consequently, any methodology addressing RQ1 must emphasize model-based engineering principles—utilizing formal, well-defined languages—and offer guidance on selecting notations that support analysis, transformation, and ultimately, implementation.

Motivation for a subject-oriented perspective (RQ2): Process and business viewpoints dominate the literature. Furthermore, they are seldom integrated with structural or technical perspectives. Only a quarter of the papers combine multiple viewpoints (Coskun-Setirek et al. 2024). Simultaneously, actors and roles are the most frequently modeled elements, appearing in 87% of the studies analyzed by Tsai et al. (2022). The subject-oriented paradigm addresses this imbalance by treating the actor (subject) as the primary modeling construct and binding every action to that subject, thereby aligning the existing actor focus with a structured interaction model. A subject-oriented approach should explicitly address this implicitly existing alignment.

Toward a multi-perspective ecosystem model (RQ2): Both reviews emphasize that current languages and metamodels neglect numerous facets of ecosystems—such as capabilities, goals, policies, and evolution—indicating that no single perspective is adequate. Future work should focus on defining ecosystem-specific design patterns, modular structures, and scalable viewpoints (Coskun-Setirek et al. 2024). Accordingly, RQ2 aims to develop a composite, modular model that integrates actor, interaction, value, governance, and technical layers while remaining adaptable to change and scalable.

Capturing dynamics, capabilities and cohesion (RQ2): Digital Ecosystems co-evolve: capabilities develop, SMEs join or leave, and innovation cycles accelerate. However, current models rarely account for temporal development, capability evolution, or goal alignment among stakeholders (Tsai et al. 2022). A next-generation model should encapsulate these dynamic aspects—development stages, partnership cohesion, and policy conflicts—and enable designers to analyze automation, open innovation, and long-term resilience. This necessity underscores the importance of subject-oriented, modular constructs that can be instantiated and adapted over time.

Need for tool support and visual analytics (RQ1 & RQ2): Only one primary study in either review proposes a dedicated modeling tool, while practitioners primarily rely on generic (drawing) applications. Tsai et al. (2022) therefore advocate new or extended tool chains that integrate formal modeling, visual analytics, and agile collaboration features. The visualization of actor networks has demonstrated its utility in isolated cases; embedding such capabilities within an integrated tool would operationalize the methodology proposed in RQ1 and enable the multi-perspective analysis sought in RQ2.

3.2 Industrial State of the Art

Industrial practice has embraced the concept that *federated* Digital Ecosystems depend on a shared technical and governance substrate: the *data space*.

A data space provides the foundational infrastructure—identity, policy enforcement, and semantic interoperability—that enables legally independent organizations to exchange data while maintaining full sovereignty over their assets. In essence, a data space serves as the framework upon which higher-level ecosystem services (*the “X-initiatives”*) can operate. Europe has emerged as the focal point of this development, particularly due to the European Commission’s 2020 *Data Strategy* directive mandating the establishment of common data spaces for key sectors and funding associated lighthouse projects.

Three programs illustrate the European trajectory:

- **Gaia-X** (2019) defines the reference architecture and federation services—identity, catalogue, compliance—for trustworthy cross-cloud data exchange.
- **Catena-X** (2021) applies the Gaia-X principles to the automotive supply chain, delivering open reference connectors (Eclipse Tractus-X) and shared semantics for various use cases in the automotive value chain, like traceability, quality and sustainability.
- **Manufacturing-X** (2022) seeks to generalize the Catena-X blueprint to the wider industrial landscape, with a strong emphasis on SME onboarding and global interoperability.

This section reviews European initiatives, examining how each builds upon the data space paradigm and identifying remaining open issues, including onboarding, governance, and scalability. Although the analysis focuses on Europe, parallel initiatives are emerging elsewhere—e. g., *Manufacturing Data Hub* in the United States, Japan’s *Trusted Web* project, and China’s *Industrial Internet Platforms*. These initiatives are briefly discussed in relation to internationalization trends and their implications for the convergence or divergence of ecosystem standards.

3.2.1 International Data Spaces

While academia has explored modeling paradigms, the industrial sector, particularly in Europe, has actively developed frameworks and initiatives to enable real-world Digital Ecosystems, emphasizing data sharing and interoperability across organizations. These initiatives are based on the idea of *data spaces*.

In 2020, the European Commission released its "European Strategy for Data," which announced the establishment of sector-specific *European data spaces*-including manufacturing, health, mobility, and others-to dismantle data silos and facilitate sovereign, standards-based data sharing among independent organizations.[1] Data spaces enable federated Digital Ecosystems, allowing multiple parties to share data based on agreed standards and trust mechanisms, without centralizing all data in a single repository.

One foundational effort in this domain is the work of the *International Data Spaces Association (IDSA)*. The IDSA, originating from a German research initiative and now comprising a broad coalition of organizations, developed the *International Data Spaces (IDS)* reference architecture, illustrated in Figure 3.1. IDS enable "sovereign and self-determined exchange of data via a standardized connection across company boundaries," (Pettenpohl et al. 2022, p. 29) with the goal of addressing interoperability, trust, and data security in cross-company data sharing.

The Reference Architecture Model for IDS defines a set of roles (e. g., data provider, data consumer, broker, identity provider) and key components required for a data space. Notably, it introduces the concept of a *data connector*—a software component that each organization deploys at its edge to technically enforce the usage policies associated with the data. In an IDS

[1] European Commission. *A European Strategy for Data* (COM(2020) 66 final). Brussels, 19 February 2020. Available at: `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2020:66:FIN`

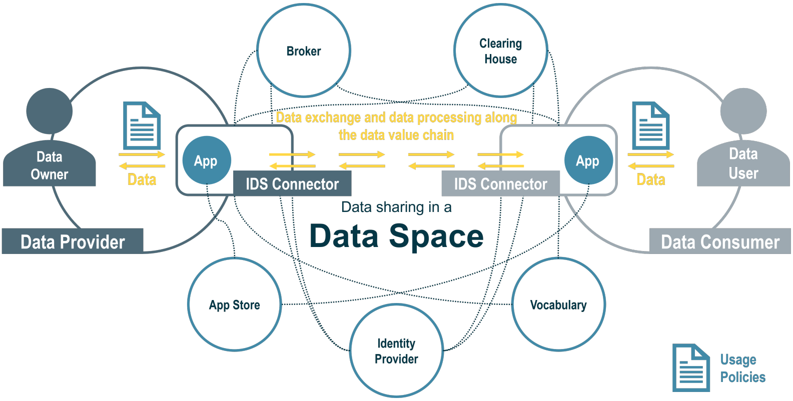


Figure 3.1: Reference Architecture for International Data Spaces, based on IDSA (Pettenpohl et al. 2022).

data space, data remains with the owner until a usage agreement is established. Even then, the recipient's use of that data can be governed by specific conditions (e. g., the data may be made read-only or restricted to one-time use), which are enforced by the connectors (Pettenpohl et al. 2022).

The architecture is inherently decentralized: there is no central database for storing all shared data. Instead, participants connect in a peer-to-peer manner using standardized connectors, supported by services such as catalogs and identity management to facilitate discovery and establish trust. The IDS architecture has undergone multiple revisions, reaching version 3.0 in 2019, and has been formalized as the DIN SPEC 27070 standard. This architecture has established much of the groundwork for subsequent European data space initiatives.

3.2.2 Gaia-X

Building on the principles of the IDS and extending them to cloud and edge infrastructure, the *Gaia-X* initiative was launched in late 2019 by Germany and France to establish a federated, and secure data backbone for Europe.

According to the public mandate on its website, Gaia-X aims to "enable trusted decentralised digital ecosystems" (vision) and to "create the de-facto standard for federated and trusted data and infrastructure ecosystems by developing common specifications, rules, policies and a verification framework"(mission).[2] The overarching goal is to interconnect cloud services, edge devices, and sector-specific data spaces while safeguarding European values of data sovereignty, openness, and privacy (Gaia-X European Association for Data and Cloud AISBL 2020).

Gaia-X established the *X-initiatives* by introducing its X-model, illustrated in Figure 3.2, which aims to connect the *"Split-X-Model"*.

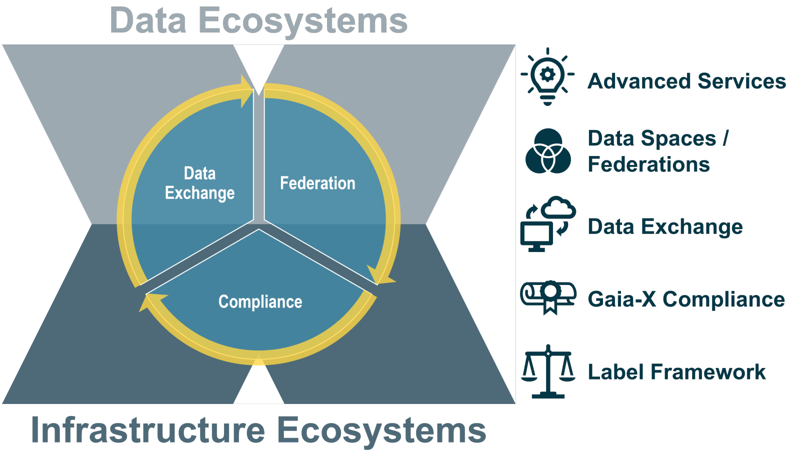


Figure 3.2: Gaia-X Framework. Own representation based on `https://gaia-x.eu/gaia-x-framework/`.

[2] `https://gaia-x.eu/about`

Rather than developing a singular cloud, Gaia-X establishes an architecture of federated services that can be implemented by various providers and industries. Key components of Gaia-X include federated identity and trust services, certification and compliance frameworks, and federated catalogs for discovering services and data offerings. By mid-2020, the Gaia-X Association was established to promote this architecture. By 2021-2022, reference implementations of core services—often referred to as Gaia-X Federation Services (GXFS)—were in development.

A cornerstone of Gaia-X is its use of *Self-Sovereign Identity (SSI)* technologies: organizations—and potentially devices—hold cryptographic wallets containing W3C verifiable credentials that attest to attributes such as company registration or ISO certification (Gaia-X European Association for Data and Cloud AISBL 2022). These digitally signed credentials can be presented and automatically validated by any counterparty, thereby realizing the *identity & trust* pillar of the Gaia-X architecture without a central authority. Aligned with the Trust-over-IP stack and EU eIDAS regulations, SSI provides every participant with a portable, universally verifiable identity that significantly reduces the need for bespoke registries or point-to-point trust integrations; a supplier certified in one data space, can reuse the same credential in another data space. Industrial consortia are now developing interoperable wallets and infrastructures—such as the European Blockchain Services Infrastructure (EBSI)—to operationalize this vision across sectors and borders.

Several domain-specific or national ecosystem initiatives have emerged under the Gaia-X umbrella, often referred to as *lighthouse projects*. One of the most prominent initiatives is *Catena-X*, officially announced in early 2021 as a collaborative network for the automotive industry. Other notable projects include *Gaia-X 4 AI*, *Gaia-X 4 Mobility*, and *Healthcare-X*. However, the results of these projects require consolidation due to the risk of fragmentation if each project develops a siloed solution. Additionally, the translation of prototypes into sustainable operational services is not guaranteed; it necessitates ongoing investment and coordination at the European level.

3.2.3 Catena-X

Catena-X[3] established the first operational data space for the automotive industry. It facilitates secure and efficient data exchange among automotive manufacturers, suppliers, and other stakeholders (e. g., dealerships, equipment providers) throughout the entire value chain. The initiative for Catena-X arose from the demand for enhanced transparency and data-driven collaboration in automotive production and supply chains, addressing use cases such as tracking components for recalls, sharing supply chain information, and enabling digital twins of vehicles. This initiative was further catalyzed by the German government designating "automotive data space" as a priority.

Catena-X leverages the Gaia-X infrastructure by implementing the IDS standard for data connectors and aligns with Gaia-X federation services for identity management, cataloging, and compliance. The initial consortium behind Catena-X comprises leading German automakers (e. g., BMW, Mercedes), suppliers (e. g., Bosch, Schaeffler), IT providers (e. g., SAP, Siemens), and others, with support from the German Federal Ministry for Economic Affairs and Climate Protection (BMWK).

By adopting a common standard, Catena-X aims to enable any participant in the network to share specific data (e. g., parts traceability information, or carbon footprint data of components) with other participants in a controlled manner. The governing association, Catena-X e.V. publishes specifications (e. g., an operating model, and technical standards) and supports a comprehensive open-source development environment in Tractus-X[4]. With the support of the dedicated operating company Cofinity-X, the Catena-X data space went live in January 2023. Thus, Catena-X represents a practical

[3] Catena-X is extensively documented in the Catena-X-Library at `https://catenax-ev.github.io/`.

[4] `https://eclipse-tractusx.github.io/`

realization of a Digital Ecosystem within a specific sector, built on the fundamental principles of data sovereignty and interoperability.

To create a global data space, Catena-X provides standards and developer tools. The standards are available on the Catena-X Association's website (Standard Library). Developer resources, i. e. KITs[5], are accessible for collaboration in the open-source Eclipse Tractus-X project. KITs demonstrate broad applicability beyond the automotive value chain. At least one KIT is available to support companies for every use case. Each KIT is consistently structured and facilitates the onboarding process for companies joining the Catena-X data space. To ensure interoperability and data sovereignty in Catena-X use cases, compliance with a minimum set of elements is mandatory. These elements include semantic models for data integration and comprehension, logic and schema for value calculation, API implementations for intercommunication, and access and usage policies to protect data sovereignty. The contents of the KIT toolbox are presented in Figure 3.3. In Tractus-X, these KITs are organized by domain and maintained independently. This supports significant expressibility but opens up room for inconsistencies between KITs.

3.2.4 Manufacturing-X

Following the success of Catena-X, the concept of X-initiatives spread to data spaces for other sectors. In late 2022 and 2023, Germany proposed and initiated *Manufacturing-X*, a broader initiative aimed at extending the data space approach to the manufacturing industry as a whole (beyond automotive). The goal is to create a cross-industry ecosystem for manufacturing companies, particularly in mechanical engineering, plant engineering,

[5] In context of Tractus-X KIT stands for *Keep It Together*.

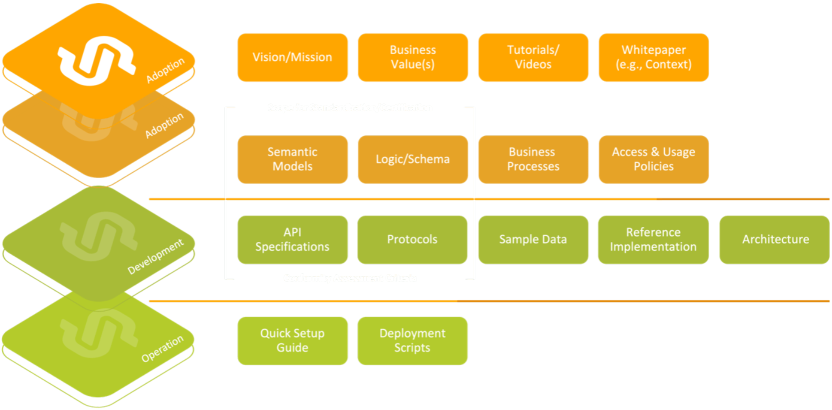


Figure 3.3: KIT Toolbox (as presented by Eclipse Tractus-X).

equipment, electronics, and related fields, to share data across complex supply chains and lifecycles. Manufacturing-X is envisioned as a federated system of data spaces that interoperate, ultimately forming an "International Manufacturing-X Council" with other nations. (Plattform Industrie 4.0 2023)

The focus is on enabling SMEs to participate in data-driven value networks without being locked into proprietary platforms. This is achieved through the use of open standards (e. g., Gaia-X, IDS) and the promotion of trust and data sovereignty. As of 2025, Manufacturing-X is still in its early implementation phase, with consortia being formed and initial projects funded. However, it underscores the commitment at both policy and industry levels to advance the SotA in real industrial data ecosystems.

3.2.5 Industry Conclusion

To summarize the industrial landscape (2019–2025): The concept of data spaces and federated Digital Ecosystems has transitioned from theory to practice, primarily within Europe. The IDS reference architecture serves as a blueprint for ensuring data sovereignty in data sharing. Gaia-X has expanded this vision into a comprehensive infrastructure framework, integrating advanced identity and trust technologies.

Domain-specific ecosystems such as Catena-X (automotive) and emerging initiatives like Manufacturing-X are implementing these concepts, backed by substantial government and industry investment. Many of these initiatives are still in development; for example, Catena-X is integrating new members and refining its standards, while others are in the pilot or conceptual stages. Table 3.2 presents a high-level overview of key initiatives, their focus areas, and identified challenges.

The current industrial SotA has generated promising building blocks and initial ecosystems. However, it also exposes certain gaps. For instance, although technology standards such as IDS connectors and SSI exist, their widespread adoption in the industry is just commencing. Many companies, particularly smaller ones, encounter challenges in integrating into these ecosystems due to technical complexity or an unclear value proposition. For example, Factory-X, one of the Manufacturing-X projects, shares the *Manufacturing as a Service* and *Modular Production* use cases with Catena-X, but include in part different functionality in these use cases. However, their current description means lack the power to clearly express these.

Moreover, each initiative typically focuses on a specific aspect or domain, raising questions about their interconnections and generalizability. For instance, how might a solution developed in Catena-X be applicable to another domain, and how can multiple data spaces interoperate instead of forming new silos? Consequently, this becomes even more complex when IDS are to be designed interoperable (Ishihara and Matsutsuka 2025). These challenges reflect existing academic gaps: a comprehensive methodology is required to systematically guide the design and implementation of interoperable Digital Ecosystems.

Table 3.2: Major industrial initiatives for Digital Ecosystems (2019-2025) and identified challenges/gaps.

Initiative	Scope and Focus	Challenges/Gaps
IDSA	Architecture and standards for sovereign data sharing across organizations; IDS-RAM (v3.0 in 2019) defines roles (provider, consumer, broker, etc.) and connector-based data exchange.	Adoption beyond pilots is slow; requires common standards uptake and trust in connector certification; integration with legacy systems remains an issue.
Gaia-X	Pan-European federated data infrastructure initiative (launched 2019) providing rules and software (Federation Services) for identity, catalogs, compliance to enable cross-cloud and cross-organization data ecosystems.	Complex governance with many stakeholders; ensuring interoperability across many domain-specific implementations; still developing trust and certification frameworks; convincing global cloud providers to adapt to Gaia-X principles.
Catena-X	Industry data space for automotive (announced 2021) enabling OEMs, suppliers, etc. to share data (e. g., parts traceability, supply chain, quality) on a common platform using Gaia-X/IDS standards.	Scaling up onboarding of diverse suppliers (esp. SMEs) into the network; harmonizing data models across companies; data governance and liability concerns; proving value for all participants to motivate data sharing.
Manufacturing-X	Initiative (Germany-led, 2022+) for a federated ecosystem across manufacturing industries (mechanical engineering, electronics, etc.), building on Catena-X concepts to improve data-based collaboration in supply chains.	Early stage – conceptual architecture in progress; challenge to coordinate multiple industry domains; ensuring compatibility with international efforts (International Manufacturing-X Council); significant R&D needed to implement basic services.

3.3 Research Gaps

Bridging the academic and industrial perspectives, we can identify several research gaps that need to be addressed to advance the engineering of Digital Ecosystems.

Gap 1: Lack of holistic modeling methodology for Digital Ecosystems. As noted, current modeling approaches either address only parts of the ecosystem (e. g., business actors or IT architecture in isolation), or they are not widely recognized or adopted in practice. There is a gap in having a *unified, model-based methodology* that encompasses high-level ecosystem design (including the identification of actors, their goals, and interactions) down to the functional realization (such as data sharing mechanisms and applicable standards). Neither the academic frameworks nor the industry architectures fully address this issue. This gap means practitioners lack guidance on *how to systematically design* a Digital Ecosystem—currently, much is done ad hoc or via lengthy consensus processes in consortia.

Gap 2: Insufficient integration of formalized modeling paradigms into Digital Ecosystem design. Subject-orientation provides a framework for modeling distributed systems of autonomous entities, which aligns well with the characteristics of Digital Ecosystems. However, its application has primarily been limited to workflow and process management within individual organizations or, at most, inter-organizational process choreographies. There exists a research gap in the application and potential extension of subject-oriented modeling to the scale of data spaces and ecosystems. For instance, how can subjects, as modeling constructs, be aligned with the actors and roles in data spaces? How can message exchanges in a model correlate with actual data exchange protocols, such as APIs in IDS connectors? There is a conceptual gap in merging the world of subject-orientation with the world of data space architectures. Addressing this gap could yield a modeling approach that is both intuitive (subject-oriented) and aligned with SotA infrastructure.

Gap 3: Socio-technical and governance aspects in modeling. Digital Ecosystems encompass more than mere technology; governance—comprising policies, legal agreements, and trust frameworks—plays a crucial role in their success. However, most modeling techniques, including the use of PASS, primarily concentrate on the "operational process" level, neglecting the explicit modeling of governance rules or incentive structures. Research is needed to extend models to capture these aspects or to link them to organizational models. For instance, can we model not only the message flows but also the rules for data usage or compliance checks as first-class elements, e. g., as part of a message payload? Currently, initiatives like Gaia-X define these elements in textual policies and agreements, separate from any process model. The challenge lies in how to incorporate governance into a model-based design in a formal or semi-formal manner.

Gap 4: Evaluation of ecosystem design choices. When designing a Digital Ecosystem, various choices must be made, such as centralized versus federated approaches, selection of standards, and considerations of openness versus exclusivity. There is limited academic guidance on systematically evaluating these design options. The primary gap lies in decision support: criteria and models that aid in predicting outcomes (e. g., how does the inclusion of more diverse participants impact data quality? or how does stringent enforcement of data sovereignty influence ecosystem growth?). A promising research opportunity exists in developing simulation or analytical tools that can evaluate specific properties (such as robustness, scalability, and participant incentives) based on a model of the ecosystem. This approach transcends current practices, which predominantly rely on expert opinion.

Overall, these gaps indicate that while the foundational elements exist (both conceptual and technical), there is a need for an integrated approach to designing Digital Ecosystems. However, these gaps are creating opportunities for various research pathways. To align with the context of this dissertation, the following subsection will introduce hypotheses that connect the identified gaps with our two main research questions.

3.4 Hypotheses

In light of the identified research gaps, we formulate six hypotheses (H1 - H6): three focus on **RQ1** (model-based methodology) and three concentrate on **RQ2** (subject-oriented model).

H1 CORE VALUES and CORE SERVICES need to be considered in Digital Ecosystem design.
Design starts with investigating the problem space. In Digital Ecosystems this means understanding the CORE VALUES. Only if these are aligned between stakeholders an effective engineering of CORE SERVICES can start. Starting with CORE VALUES is expected to shorten later consensus cycles, reduce design conflicts and provide a stable reference for iterative refinements of functionality.

H2 Digital Ecosystems are complex enough that a model-based approach only makes sense.
Because a Digital Ecosystem already demands extensive documentation, employing a formal, machine-interpretable modeling language should add little overhead while substantially improving cross stakeholder comprehension and reducing rework. The hypothesis predicts measurable gains in clarity and defect avoidance over informal box-and-line sketches.

H3 A Digital Ecosystem is a system of systems and CORE PARTICIPANTS perform a complex choreography.
A Digital Ecosystem is more than a single, monolithic workflow. It is only describable in terms of interconnected use cases in which the CORE PARTICIPANTS are engaged. The entirety of activities in the complex system of systems can not be centrally orchestrated.

H4 Subject-orientation is well suited for socio-technical Digital Ecosystems.
Subject-orientation originates in distributed systems research. DBE

descriptions have process descriptions as most prevalent perspective and actors as most used element. Explicitly engaging a subject-oriented description unlock substantial potential in description expressiveness and comprehension.

H5 A Digital Ecosystem model requires multiple perspectives.
Digital Ecosystems bring together many stakeholders with diverse backgrounds. Furthermore, they often include various use cases. Consequently, it is highly unlikely that one single perspective will sufficiently describe a Digital Ecosystem. This raises the question of how to enforce subject-orientation from different viewpoints.

H6 Digital Ecosystems evolve over time.
Constant growth, iterative evolution, and open innovation are inherent characteristics of Digital Ecosystems, making change its only constant. The representation of the subject-oriented model needs to support easy, and ideally federated, maintainability. This calls for rigorous versioning and release planing of the model artifacts.

These hypotheses aim to guide the development of a model-based, subject-oriented methodology for designing Digital Ecosystems.

3.5 Conclusion and Next Steps

In this chapter, we reviewed the SotA in both research and industrial practices regarding Digital Ecosystems and their modeling. Academically, there is recognition of the need for improved modeling approaches for Digital Business Ecosystems (DBEs). Subject-oriented modeling has emerged as a promising paradigm that aligns well with the distributed nature of ecosystems. Industrially, significant initiatives such as IDS, Gaia-X, and Catena-X are advancing the development of operational ecosystems. Although these initiatives are not explicitly model-driven, they implicitly adhere to certain architectures and principles that could inform a methodology.

The analysis identified several gaps, notably the absence of an integrated model-based design approach and the insufficient application of explicitly subject-oriented thinking in current practices. These findings lay the groundwork for the subsequent sections of this dissertation.

The next steps involve developing and proposing a research method that addresses the identified gaps and aligns with our two research questions (RQ1 & RQ2) as well as the newly amended design hypotheses (H1-H6). Specifically, the following chapter will introduce the research approach adopted to investigate these questions. After the presentation of results in Chapters 5 & 6 the hypotheses are further discussed in Chapter 7.

4 Research Approach

This work is situated within Design Science Research (DSR), which focuses on the iterative development and evaluation of artifacts to address real-world problems (Hevner et al. 2004). DSR is particularly well-suited for this study, as it emphasizes both the creation of practically relevant solutions and the contribution to scientific knowledge. In this work, the primary artifact is a novel methodology for designing Digital Ecosystems.

The foundations of DSR can be traced back to Herbert A. Simon's seminal work *The Science of the Artificial* (Simon 1996), which introduced design as a problem-solving paradigm centered on the purposeful creation of artifacts. Building on this tradition, DSR has evolved into a well-established research approach capable of addressing complex sociotechnical challenges through iterative and rigorously evaluated design processes.

This chapter outlines the research approach employed in this study. First, it introduces the DSR paradigm and design guidelines that offer structure and evaluation criteria for DSR studies. Building on this foundation, a specific DSR method—the Soft Design Science Methodology (SDSM)—is selected and its suitability for this context is discussed.

Following this, section 4.2 outlines the specific design of this research study. It begins with the articulation of the research goal and continues by introducing the IntWertL project as a case study for empirical application. The subsequent section describes the evaluation strategy employed to achieve the research goals, including evaluation objectives, properties, and episode design. Building on this, the iterative process for developing and refining Level 1 artifacts—the Synthesis Models—is presented using an adapted nine-step structure of the SDSM. Finally, the chapter describes how the knowledge generated through these design cycles is consolidated into a Level 2 artifact: the Engineering Digital Ecosystems (EDE) methodology.

The chapter concludes with a summary that integrates the components of the research approach and positions the methodological contribution within the broader context of the thesis.

4.1 Design Science Research

DSR is a research paradigm that emphasizes the creation and evaluation of artifacts to solve real-world problems while simultaneously contributing to the advancement of knowledge. In the context of Information Systems (IS), which are inherently socio-technical systems, DSR integrates technical solutions with the complex social and organizational environments in which these solutions are applied. According to Hevner et al. (2004), DSR operates on the principle that research artifacts, including constructs, models, methods, and instantiations, must not only address identified problems but also be rigorously evaluated for utility, quality, and efficacy within their intended context. Peffers et al. (2007) further highlights that the DSR process is iterative and involves problem identification, objective definition, and research communication, as well as artifact design, demonstration, and evaluation. This structured approach ensures that solutions are both practically relevant and theoretically grounded, making DSR a vital methodology for addressing the dynamic challenges inherent in complex socio-technical systems, such as Digital Ecosystems.

Over the years, various DSR methodologies and frameworks have emerged to guide the systematic development and evaluation of artifacts. These include the widely recognized framework proposed by Hevner et al. (2004), the Design Science Research Methodology (DSRM) by Peffers et al. (2007), and more recent approaches such as the Soft Design Science Methodology

(SDSM) by Baskerville et al. (2009). Each methodology emphasizes different aspects of the design process, from theoretical rigor to stakeholder engagement and contextual relevance. However, regardless of the choice of a specific method, the DSR paradigm necessitates adherence to certain guiding principles for effective application (Hevner et al. 2004, Venable et al. 2017).

4.1.1 Design Science Research Guidelines

Hevner et al. (2004) provide seven foundational guidelines for conducting DSR, emphasizing the creation and evaluation of artifacts to address identified problems within a research context. These guidelines are presented in Table 4.1. This work ensures its validity by adhering to these widely recognized seven DSR guidelines.

The first guideline, *Design as an Artifact*, specifies that DSR must produce viable artifacts, such as constructs, models, methods, or instantiations. The second guideline, *Problem Relevance*, ensures that the research addresses significant and practical problems. The third guideline, *Design Evaluation*, necessitates rigorous evaluation of the utility, quality, and efficacy of the artifact within its application domain. The fourth guideline, *Research Contributions*, emphasizes the necessity for clear contributions in the form of novel artifacts, methodologies, or theoretical insights. The fifth guideline, *Research Rigor*, underscores the importance of employing rigorous methods for both artifact creation and evaluation. The sixth guideline, *Design as a Search Process*, highlights the iterative nature of design, focusing on the refinement of artifacts to achieve the desired results. Finally, the seventh guideline, *Communication of Research*, mandates effective communication of results to both technical and managerial audiences. Collectively, these guidelines provide a robust framework for conducting DSR that is both practically relevant and theoretically rigorous.

Table 4.1: Design Science Guidelines based on Hevner et al. (2004, p. 83).

Guideline	Description
Guideline 1: **Design as an Artifact**	DSR involves the creation of a construct, a model, or a method as an artifact.
Guideline 2: **Problem Relevance**	The objective is to design technology-based solutions to important business problems.
Guideline 3: **Design Evaluation**	The utility, quality, and efficacy of the design must be rigorously demonstrated via well-executed methods.
Guideline 4: **Research Contributions**	Effective DSR must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: **Research Rigor**	DSR relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: **Design as a Search Process**	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: **Communication of Research**	DSR must be presented effectively to both technology-oriented and management-oriented audiences.

4.1.2 The Soft Design Science Methodology (SDSM)

SDSM, developed by Baskerville, Pries-Heje, and Venable (Pries-Heje et al. 2007, Baskerville et al. 2009, Pries-Heje et al. 2014), provides a robust framework for addressing complex socio-technical problems. It integrates principles from both DSR and Soft Systems Methodology (SSM), allowing for a comprehensive approach that bridges the technical and social aspects of problem-solving. This section outlines the rationale for selecting SDSM, as well as its key principles, structure, and contributions, establishing a foundation for its application in this study.

4.1.2.1 Selection of SDSM

SDSM was selected for this study due to its strong methodological alignment with the evaluation criteria outlined in the *DSR Methodology Comparison Framework* by Venable et al. (2017). This selection is informed by the technological rules proposed by Venable et al., which guide the choice of DSR methodologies based on conceptual orientation and research focus.

According to the *First Technological Rule*, SDSM is situated within the subjectivist and interpretivist paradigm. This positioning is particularly suitable for contexts where socio-technical phenomena are not governed by universal laws, and the validity of knowledge relies on contextual relevance and stakeholder interpretation. Such a stance aligns with the ontological complexity of Digital Ecosystems, where heterogeneous actors, evolving structures, and social negotiation play central roles. In these environments, the assumption of a single "best" artifact or purely objective solution is inappropriate. Instead, iterative engagement with stakeholders and systemic reflection are essential for co-constructing context-sensitive and adaptable solutions.

The selection of SDSM is further justified by the *Second Technological Rule* for subjectivist/interpretive methodologies, which emphasizes the prioritization of theoretical insights over direct interventions in organizational practice. SDSM enables the abstraction of specific problem instances into generalized problem classes, thus facilitating the generation of transferable design knowledge that extends beyond the immediate application context. This capability aligns with the objectives of this study, which aims to develop a broader methodology, including a reference model for Digital Ecosystems.

In addition to these methodological considerations, SDSM explicitly incorporates stakeholder participation, iterative refinement, and systemic abstraction (Pries-Heje et al. 2014). These characteristics ensure that the research process remains responsive to real-world complexity while maintaining the rigor necessary for scientific contribution, making it an ideal choice for this study.

4.1.2.2 Principles of the SDSM

SDSM operates within a design-oriented paradigm, emphasizing the iterative creation, evaluation, and refinement of artifacts as mechanisms for addressing specific socio-technical challenges while generating generalizable design knowledge. This methodological orientation directly supports the primary objective of this study: to develop and validate a methodology that addresses practical issues related to the design of Digital Ecosystems and contributes to the advancement of DSR theory.

The problem domain addressed in this research comprises complex socio-technical systems, characterized by dynamic interactions, multiple stakeholder perspectives, and emergent properties. As noted by Pries-Heje et al. (2014), SDSM is particularly well-suited for such environments due to its systematic integration of principles from SSM. SSM's systemic thinking facilitates problem exploration and abstraction beyond surface-level symptoms, enabling the identification of underlying structural and social dynamics that shape problem contexts.

The hybridization of SSM's systemic inquiry with DSR's artifact-centric focus constitutes the core of SDSM's methodological approach. This integration allows for the development of solutions that are both contextually grounded and theoretically informed. Central to SDSM is the emphasis on iterative, participatory design cycles in which artifacts are collaboratively designed, implemented, evaluated, and refined. Stakeholder engagement is embedded at each stage of the process, ensuring that solutions remain responsive to contextual needs while supporting reflective learning and theory-building. Following these principles SDSM pursues two interrelated objectives:

- The construction and deployment of effective solutions tailored to specific socio-technical problems.
- The derivation of transferable theoretical insights through cyclical refinement, wherein each iteration contributes to both problem-solving efficacy and the accumulation of design knowledge.

By systematically balancing practical intervention with the abstraction necessary for knowledge generalization, SDSM enables the dual achievement of pragmatic outcomes and academic contribution (Pries-Heje et al. 2014). This dual focus makes SDSM particularly appropriate for research settings where the complexity of stakeholder relationships and the fluidity of system boundaries demand both methodological rigor and contextual adaptability.

4.1.2.3 Structure of the SDSM

One iteration of SDSM consists of seven activities that provide a structured yet flexible framework for problem-solving. These seven steps are depicted in Figure 4.1.

The SDSM framework is based on the iterative interaction between problem exploration, artifact design, and evaluation. This methodology, as outlined by Baskerville et al. (2009), comprises the following steps:

1. **Problem Identification:** A specific problem is identified collaboratively with stakeholders, focusing on delineating its boundaries and scope.

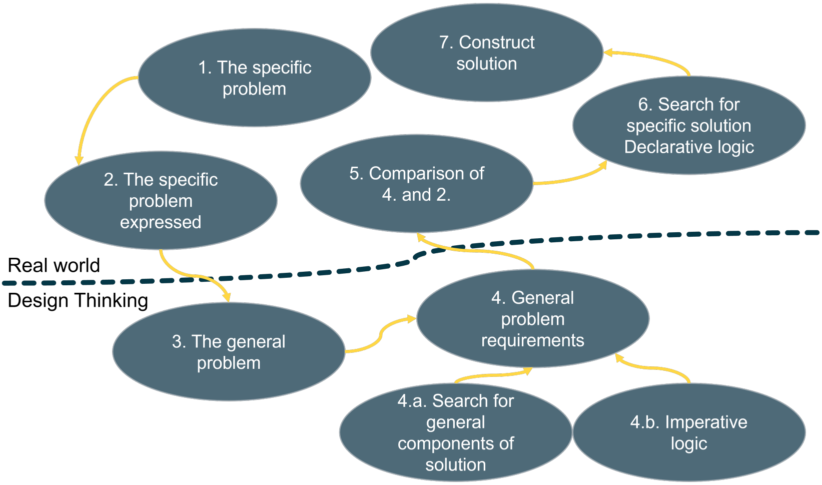

1. The specific problem
7. Construct solution
6. Search for specific solution Declarative logic
2. The specific problem expressed
5. Comparison of 4. and 2.
Real world
Design Thinking
3. The general problem
4. General problem requirements
4.a. Search for general components of solution
4.b. Imperative logic

Figure 4.1: SDSM Framework. Own representation based on Baskerville et al. (2009), Pries-Heje et al. (2014).

2. **Requirements Definition:** The problem is translated into a set of specific requirements that guide solution development.

3. **General Problem Abstraction:** Using systemic and design thinking, the specific problem is abstracted into a general problem that captures both technical and social dimensions. At this stage, design thinking shifts from the specific problem to a class of problems.

4. **General Solution Design:** A general solution or class of solutions is derived for the abstracted problem, employing design science techniques and imperative logic to define general requirements.

5. **Fit Analysis:** The general design is analyzed for its fit with the specific problem, ensuring alignment between the problem and the solution. The specific problem is rearticulated in terms of the general requirements and the imperative logic.

6. **Specific Solution Search:** A search is performed for specific components that instantiate the general solution for the identified problem.

This step addresses both technical feasibility and practical implementation.

7. **Solution Deployment:** A specific solution is constructed and deployed in the social system. This stage includes observing the impact of the solution, deriving lessons, and iterating the process for further refinement.

4.2 Research Method

This section outlines the research approach employed to develop and validate the EDE methodology. The EDE methodology includes a structured, model-based approach and a generalized conceptual reference model, referred to as *Synthesis Model*.

The research is organized into three iterative design cycles, each producing a version of the Synthesis Model as a Level 1 artifact that clarifies and refines the problem space of Digital Ecosystem design. These cycles are conducted within the context of the IntWertL project, which provides a rich and dynamic environment for validation. The process is guided by an adapted version of the SDSM, supplemented by a structured evaluation strategy. Finally, the emerging design knowledge is consolidated into the EDE methodology as a Level 2 design artifact. [1]

[1] The notion of Level 1, Level 2, and Level 3 artifacts is adopted from Gregor and Hevner (2013): "A specific DSR research project can produce artifacts on one or more of these levels ranging from specific instantiations at Level 1 in the form of products and processes, to more general (i.e., abstract) contributions at Level 2 in the form of nascent design theory (e.g., constructs, design principles, models, methods, technological rules), to well-developed design theories about the phenomena under study at Level 3."

4.2.1 Research Goal

The central goal of this research is to design and validate a methodology, along with a corresponding conceptual model, that facilitates the structured development of Digital Ecosystems. This work is guided by two main research questions:

RQ1 How can the design process of Digital Ecosystems be supported by a model-based and subject-oriented methodology?

RQ2 What kind of subject-oriented model can be used to describe a Digital Ecosystem?

To address these questions, the research adopts an DSR approach, grounded in the principle of design as a knowledge-generating activity through the creation of purposeful artifacts. This is operationalized through three iterative design cycles, each following an adapted version of the SDSM. These cycles result in the creation and evaluation of Level 1 design artifacts—specifically, successive versions of a Synthesis Model that describes the Digital Ecosystem, with particular focus on its CORE SERVICES and the problem space of its SOLUTIONS. According to the classification by Gregor and Hevner (2013), this represents a contribution of type *invention*, where both the solution and the application domain exhibit low maturity.

Building on the knowledge accumulated during these iterations, the research consolidates a Level 2 artifact: the EDE methodology. This artifact encompasses a practitioner-oriented design approach and a generalized conceptual model derived from the evaluated Synthesis Models. Drawing on principles from DSR, design thinking, and subject-orientation—and leveraging formal modeling languages such as SysML and PASS—the methodology is adapted for reuse in similar yet evolving application domains. As such, the EDE methodology constitutes an instance of *exaptation* in Gregor and Hevner's framework, where a mature solution is applied to a nascent domain context.

This study contributes to emerging design theory by articulating scientifically grounded and practically applicable design knowledge. It positions the EDE methodology as a novel, subject-oriented framework developed through iterative engagement with real-world challenges, aimed at supporting open innovation in dynamic and federated Digital Ecosystem environments.

4.2.2 Application Context: The IntWertL Project

The application context for this study is the BMWK-funded project *IntWertL*[2]. This interdisciplinary research and development project provides the empirical setting for the iterative design and evaluation of Level 1 artifacts, as well as for the consolidation of the EDE methodology.

IntWertL addresses the challenge of developing digital platforms to support engineering collaboration among SMEs in the automotive sector, as illustrated in Figure 4.2. These enterprises often lack the resources to independently create such platforms and face significant obstacles in coordinating across organizational and technical boundaries. Additional complexity arises from the diversity of stakeholders involved—including engineering service providers, manufacturing service providers, automotive OEMs, and software solution providers—each having differing expectations, terminologies, and interpretations of key concepts.

Beyond these operational and semantic challenges, the project operates within a rapidly transforming mobility sector. Global competition, sustainability mandates, and changing societal expectations regarding mobility are driving the need for alternative, resource-efficient transportation solutions. While use-case-optimized lightweight vehicles exist as prototypes, their commercial adoption is hindered by the high costs associated with

[2] IntWertL stands for *Intelligente Wertschöpfungsnetzwerke für Leichtbaufahrzeuge geringer Stückzahl* (Intelligent Value Networks for Lightweight Vehicles in Small Quantities).

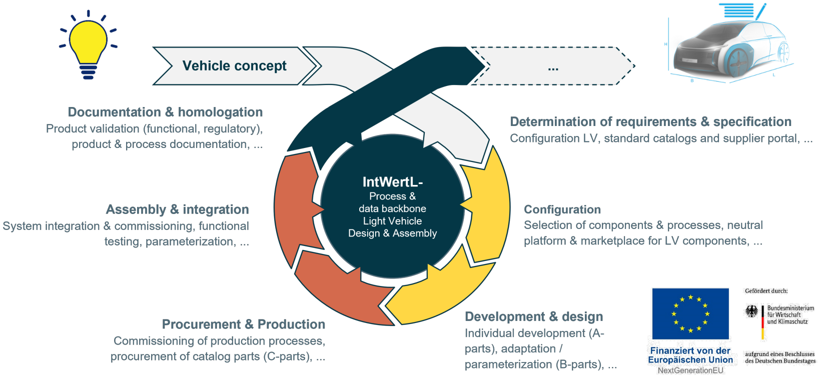

Vehicle concept
...
Documentation & homologation
Product validation (functional, regulatory), product & process documentation, ...
Determination of requirements & specification
Configuration LV, standard catalogs and supplier portal, ...
IntWertL-
Process & data backbone Light Vehicle Design & Assembly
Assembly & integration
System integration & commissioning, functional testing, parameterization, ...
Configuration
Selection of components & processes, neutral platform & marketplace for LV components, ...
Procurement & Production
Commissioning of production processes, procurement of catalog parts (C-parts), ...
Development & design
Individual development (A-parts), adaptation / parameterization (B-parts), ...
Finanziert von der Europäischen Union
NextGenerationEU
Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

Figure 4.2: IntWertL: Fully digitized process for product creation, from concept to car. Own representation based on Bönsch et al. (2024).

small-series production. In response, IntWertL aims to establish a digital development and production platform tailored for SMEs. The proposed platform will facilitate distributed engineering and manufacturing processes, leveraging digitization and AI to enhance automation, reduce development cycles, and decrease costs. Ultimately, the goal is to enable SMEs to act as integrators or OEMs, improving their global competitiveness through digitally supported collaboration.

A central difficulty is the lack of a shared understanding. For instance, what constitutes a *digital platform*? Here, the term *platform* is used inconsistently, referring to digital collaboration tools, physical vehicle architectures, or marketplace-style environments. This terminological ambiguity hinders effective communication and alignment, leading to fragmented development efforts. Furthermore, the integration of pre-existing software components from multiple partners further complicates coordination.

This evolving and multifaceted project context provides an ideal setting for this research study. The absence of a shared reference model, the open-ended nature of the design problem, and the high diversity of stakeholder perspectives render IntWertL particularly suitable for applying and evaluating a model-based, subject-oriented methodology for Digital Ecosystems.

4.2.3 Design Evaluation

As shown, rigorous evaluation of design artifacts is essential for the DSR process (Hevner et al. 2004). This is particularly important when the maturity of the application domain is low, and more abstract, complete, and mature knowledge (levels 2 or 3) must be contributed (Gregor and Hevner 2013). Over the years, numerous methods and evaluation techniques have been proposed (Dresch et al. 2015, chap. 4.4). The Framework for Evaluation in Design Science Research (FEDS) by Venable et al. (2016) is utilized here to select an appropriate evaluation approach. Venable et al. propose four steps utilizing FEDS:

1. **Evaluation Goals**: Explicate the goals of the evaluation.
2. **Evaluation Strategy**: Choose the evaluation strategy or strategies.
3. **Evaluation Properties**: Determine the properties to evaluate.
4. **Evaluation Episodes**: Design the individual evaluation episode(s). (Venable et al. 2016, p. 6)

The following subsections illustrate the application of these four steps in designing the evaluation approach for the Level 1 design artifacts, resulting from the SDSM component of this study.

4.2.3.1 Evaluation Goals

Venable et al. (2016) propose that there are at least four competing general objectives in the evaluation of DSR projects: *Rigor*, *Uncertainty and Risk Reduction*, *Ethics*, and *Efficiency*. These objectives are further detailed in Table 4.2. Based on these broadly described objectives, four main goals for the evaluation in this study are defined:

EG1 **Relevance:** Design artifacts should address significant and relevant topics for intended users.

EG2 **Enrichment:** Design artifacts should provide well-developed insights and information that enhance existing knowledge and understanding contextually.

EG3 **Usability:** Ensure design artifacts are easily understood and communicate their purpose effectively without additional explanation.

EG4 **Timeliness:** Each Design Artifact must be engineered within time and budget constraints.

EG1 and EG3 are based on *uncertainty and risk reduction*. Here, only the "human social/use risks (i. e., risks that the artefact will not fit well into the use or social situation and therefore not work or cause further problems)" (Venable et al. 2016, p. 6) are addressed, as no technical risks are relevant to this research. Intuitive Usability (EG3) is particularly necessary, as the users interacting with the model of the Digital Ecosystem are heterogeneous and diverse. Moreover, most stakeholders do not come from the IT domain, and no prior knowledge of modeling notation can be assumed. Additionally, decisions on the alignment of the Digital Ecosystem are made by mid-to-upper management of the ORCHESTRATORS and CORE PARTICIPANTS who are pressed for time and unlikely to spend it on understanding the notation first. Relevance (EG1) combines the social risk of acceptance of the model

Table 4.2: Summary of general objectives in designing the evaluation component of DSR based on Venable et al. (2016)

Objective	Description
Rigor	Ensures the artifact instantiation causes observed outcomes (efficacy) and works in real situations (effectiveness). Artificial evaluation is ideal for efficacy, while naturalistic evaluation suits effectiveness. Summative evaluation, typically occurring towards the end of the evaluation process, ensures more rigor than formative evaluation. Multiple summative evaluations may be required to cover different artifacts, aspects, or contexts.
Uncertainty & Risk Reduction	Formative evaluation is critical when design uncertainties are significant. It identifies human social/use risks and technical risks early, reducing costs and improving the artifact's quality. Conducted early in the evaluation process, formative evaluation helps mitigate risks through early detection of difficulties.
Ethics	Evaluation must consider risks to stakeholders (people, organizations, public, future generations) and the environment. Formative evaluation reduces immediate risks, while summative evaluation ensures meticulous mitigation of long-term risks. The evaluation process itself should avoid introducing risks.
Efficiency	Balances evaluation goals against resources like time and money. Formative evaluation is cost-effective as it identifies issues early, reducing costs of instantiation and theory specification. Naturalistic evaluation is generally costlier and more time-intensive than artificial evaluation. Non-empirical (artificial) methods offer significant cost savings.

with the imperative for effectiveness. Therefore, the artifact must be efficient in its representation and show what information is of interest to the stakeholders.

EG2, Enrichment, is also a more precise part of the generalized goal of Rigor. The goal is not just to build useful artifacts but to contribute knowledge to the EDE methodology. Thus, "the evaluation approach not only needs to address the quality of the artefact, but also the quality of its knowledge outcomes." (Venable et al. 2016, p. 2)

Lastly, Timeliness (EG4) considers *Efficiency*. The design artifacts for the IntWertL research project are developed within a strict timeline. The initial platform MVP must be operational and demonstrable within one year of commencement, necessitating the completion of the first Synthesis Model iteration within three months. Furthermore, project partners and stakeholders expect to utilize the model without engaging in extensive validation procedures.

It is important to note that there is no immediate risk to stakeholders or the environment from the generation of the Synthesis Models. Therefore, the *Ethics* objective is not relevant to this study. Additionally, unlike many other DSR projects, the specific styling of the design artifact is not a central purpose of this study. This is likely to be influenced more by the Corporate Identities of the ORCHESTRATORS of a Digital Ecosystem than by the employed design methodology.

4.2.3.2 Evaluation Strategy

Venable et al. (2016) propose heuristic selection criteria to choose among four different evaluation strategies. Table 4.3 summarizes these four relevant evaluation strategies, and Figure 4.3 displays them within the FEDS domains. The major design risk being user-oriented indicates the use of a *Human Risk & Effectiveness* strategy. However, the Synthesis Model is a relatively small artifact, which suggests the use of the *Quick & Simple* strategy.

As Venable et al. (2016) acknowledged in their work, combinations of the presented strategies or even new strategies may be suitable for certain DSR projects, leading to the adapted *Quick Human Risk & Effectiveness* strategy that is employed here.

Table 4.3: Circumstances for selecting a relevant DSR evaluation strategy (Venable et al. 2016, p. 6)

DSR evaluation strategy	**Circumstance selection criteria**
Quick & Simple	If small and simple construction of design, with low social and technical risk and uncertainty
Human Risk & Effectiveness	If the major design risk is social or user oriented *and/or* if it is relatively cheap to evaluate with real users in their real context *and/or* if a critical goal of the evaluation is to rigorously establish that the utility/benefit will continue in real situations and over the long run
Technical Risk & Efficacy	If the major design risk is technically oriented *and/or* if it is prohibitively expensive to evaluate with real users and real systems in the real setting *and/or* if a critical goal of the evaluation is to rigorously establish that the utility/benefit is due to the artifact, not something else
Purely Technical Artifact	If artifact is purely technical (no social aspects) or artifact use will be well in future and not today

The *Quick Human Risk & Effectiveness* strategy comprises two main components. Conducting research efficiently while minimizing false starts and revisions necessitates performing formative evaluations on one or more design iterations. This process is crucial for establishing an effective research

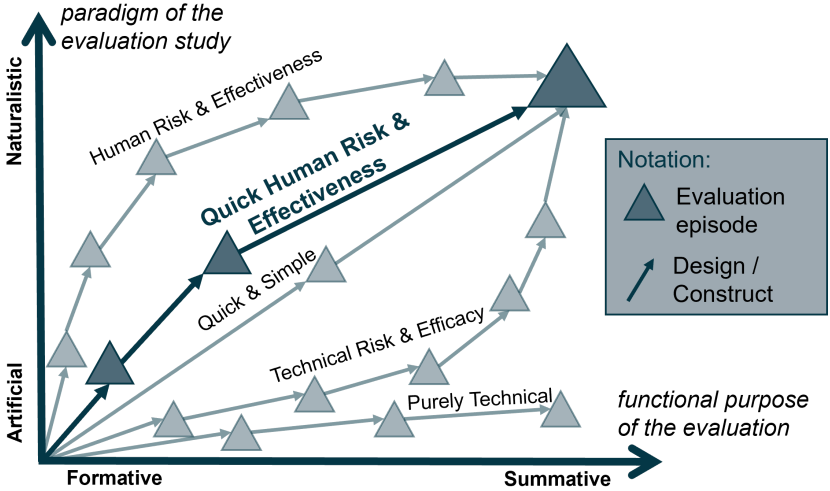

paradigm of the evaluation study
Naturalistic
Artificial
Human Risk & Effectiveness
Quick Human Risk & Effectiveness
Quick & Simple
Technical Risk & Efficacy
Purely Technical
Notation:
Evaluation episode
Design / Construct
functional purpose of the evaluation
Formative
Summative

Figure 4.3: DSR evaluation strategies based on Venable et al. (2016) and the adapted *Quick Human Risk & Effectiveness* strategy that is employed here.

trajectory, particularly in addressing user-oriented uncertainty. Furthermore, to facilitate rapid progress and improve usability, an early artificial evaluation is employed. These aspects are derived from the *Quick & Simple* strategy. However, given the extensive requirements of a more advanced knowledge contribution, an additional phase of summative evaluation is also considered essential. Additionally, the detailed nature of the research methodology and the user-centric design underscore the importance of integrating a naturalistic evaluation as well. Nonetheless, the pure *Human Risk & Effectiveness* strategy conflicts with EG4 Timeliness. Consequently, the *Quick Human Risk & Effectiveness* strategy initiates an early sequence of formative and artificial evaluations to mitigate the risk of failure. It then progresses swiftly towards a comprehensive summative and naturalistic evaluation.

4.2.3.3 Evaluation Properties

Given the formative nature of this research, a limited number of evaluands is appropriate, as recommended by Venable et al. (2016). The selected evaluands must directly reflect the human-centered risks and effectiveness concerns pertinent to the research objectives. To maintain alignment with these goals and reduce complexity, one evaluand is defined for each evaluation goal (EG).

An exception is EG4 (Timeliness), which pertains to the development process rather than the artifact itself. Since adherence to timeliness is a strict prerequisite—no artifact can be produced without it—it is excluded from evaluation.

Accordingly, the following three evaluation properties are defined:

- **Display of Active Entities (EG1 Relevance)**
  While the model-based nature of the artifacts ensures structural relevance, their subject-oriented characteristics require evaluation to confirm alignment with the second central aspect to this research.
- **Effectiveness (EG2 Enrichment)**
  This criterion assesses whether the artifact effectively represents the concerns of a Digital Ecosystem that are of primary interest.
- **Interpretability (EG3 Usability)**
  Usability is operationalized as interpretability. If stakeholders cannot easily understand the model, its practical value is diminished.

4.2.3.4 Evaluation Episodes

The design of evaluation episodes follows the structured approach proposed by Venable et al. (2016). The first step involves identifying and analyzing environmental constraints that may impact the evaluation process. Although the research project IntWertL provides access to all relevant stakeholders, a comprehensive, naturalistic evaluation of multiple alternatives is infeasible due to the limited availability of managing directors in SMEs. Attention spans are scarce, and the collective cannot be presented with numerous or potentially conflicting representations of the Digital Ecosystem, as this could undermine the formation of a shared frame of reference. Moreover, the project is time-constrained: an MVP is expected within one year, necessitating an early version of the Synthesis Model.

The second step involves prioritizing contextual factors. At this stage, it is critical to establish a common frame of reference among stakeholders. Consequently, the Synthesis Model does not require fully specified data structures, API definitions, or detailed architectural blueprints. Instead, evaluation efforts under EG1 (Relevance) focus on the model's overarching conceptual structure rather than its technical details.

The third step involves planning concrete evaluation episodes, i. e. "decide a plan including determination of how many evaluation episodes there will be as well as when particular evaluation episodes will be conducted and in what way. Hence the outcome is: Who? Is doing what? When?" (Venable et al. 2016, p. 8) As shown in Figure 4.3 and the iterative approach outlined in Pries-Heje et al. (2014), each design iteration includes three evaluation episodes, that are categorized according to Hevner et al. (2004):

1. **Ex Ante Evaluation (General)**
   Conducted prior to the creation of specific artifacts, this episode evaluates the general solution concept. It is a formative and artificial evaluation, carried out by researchers using the descriptive method of informed argumentation. The goal is to ensure internal consistency and conceptual soundness of the approach before engaging stakeholders.

2. **Ex Ante Evaluation (Specific)**
   This episode involves the analysis of concrete design artifacts using both static and dynamic analytical methods. It can include both formative and summative components, depending on the maturity of the

artifacts. Evaluation is typically artificial but may also involve expert feedback.

3. **Ex Post Evaluation**
   Performed after artifact implementation, this episode assesses the artifact in its real-world context. In early iterations, this includes observations (field-studies) during stakeholder interaction. In the third iteration, a more descriptive approach is taken, involving a structured questionnaire. The rather naturalistic and summative Ex Post Evaluation episodes provide valuable, formative results for the next iteration and the nascent methodology.

4.2.4 Iterative Design of Synthesis Models – An SDSM Approach

This section outlines the iterative process used in designing the Synthesis Model of the IntWertL Digital Ecosystem. The approach is based on the SDSM, enhanced with a *Quick Human Risk & Effectiveness* evaluation strategy to ensure stakeholder alignment and facilitate early evaluability.

The Synthesis Model is centered around the CORE SERVICES of the Digital Ecosystem. It does not propose concrete SOLUTIONS; rather, it defines the conceptual structure, constraints, and rules within which SOLUTIONS can emerge. This is particularly important in the context of Digital Ecosystems, which involve federated services, diverse stakeholder groups, and continuous evolution. The model must remain open enough to foster innovation while maintaining sufficient structure to provide a shared reference point across ORCHESTRATORS, CORE USERS and CONTRIBUTORS.

The resulting nine-step design process is illustrated in Figure 4.4. [3] Each activity supports iterative development across three distinct design cycles, as presented in Chapter 5, emphasizing the importance of ensuring the model's relevance, effectiveness, and interpretability. Each of the nine steps is briefly described in the following subsections.

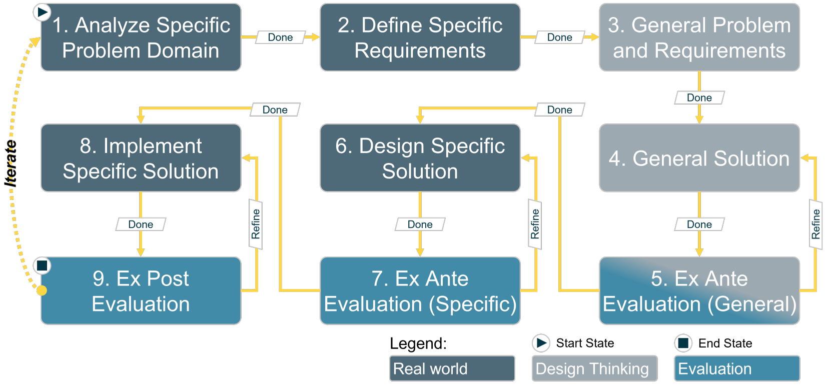

1. Analyze Specific Problem Domain
2. Define Specific Requirements
3. General Problem and Requirements
4. General Solution
5. Ex Ante Evaluation (General)
6. Design Specific Solution
7. Ex Ante Evaluation (Specific)
8. Implement Specific Solution
9. Ex Post Evaluation
Done
Refine
Iterate
Legend:
Start State
End State
Real world
Design Thinking
Evaluation

Figure 4.4: SBD of main activities for the researcher in the iterative SDSM approach to design Synthesis Models (based on Baskerville et al. (2009), Pries-Heje et al. (2014)).

4.2.4.1 Analyze Specific Problem Domain

This step centers on establishing an initial understanding of the specific problem domain in which the Digital Ecosystem will be designed. The objective is to identify key stakeholder concerns, systemic challenges, and socio-technical constraints that inform the context of the Synthesis Model. Instead of pursuing technical completeness, the analysis emphasizes relevance and stakeholder alignment—particularly in relation to perceived risks, collaborative dynamics, and desired outcomes.

[3] Figure 4.4 presents an SBD. The corresponding subject is the SDSM research practitioner, i. e. I. The labeling of states occasionally omits the traditional verb + object format to enhance visual representation and improve alignment with Figure 4.1.

This analysis utilizes problem-oriented knowledge and contextual understanding to inform subsequent steps. It establishes the foundation for defining specific requirements and ensures that the design process remains anchored in real-world conditions.

4.2.4.2 Define Specific Requirements

In this step, the problem analysis is translated into a structured set of specific requirements for the Synthesis Model. These requirements articulate what the model must represent to be deemed relevant, effective, and interpretable within the context of the targeted Digital Ecosystem. The emphasis is placed on stakeholder-relevant content, including roles, interactions, and ecosystem-level concerns, rather than on technical specifications.

The outcome of this step establishes a design boundary: it constrains the aspects that the artifact must capture and informs evaluative judgments in later stages. Requirements are expected to evolve over iterations; however, they must be sufficiently concrete to guide the initial development of a Synthesis Model.

4.2.4.3 General Problem and Requirements

This step abstracts specific requirements and challenges into a generalizable format. The objective is to identify broader patterns or recurring issues that transcend the immediate context of the current Digital Ecosystem. This abstraction facilitates the creation of solutions that are not only contextually appropriate but also adaptable to other environments with similar characteristics.

Outlining the general problem and requirements connects context-specific insights with the conceptual framework necessary for generating design knowledge. Furthermore, it aids in the development of evaluation criteria that reflect both local and general significance.

4.2.4.4 General Solution

Based on the generalized problem and requirements, this step defines a conceptual solution that addresses the fundamental challenges of Digital Ecosystem design. The general solution outlines the principles, structures, and design heuristics that the Synthesis Model should adhere to, independent of specific organizational constraints.

This abstraction allows for the identification of reusable and theoretically grounded solution patterns, serving as a reference for the design of specific artifacts. Additionally, it facilitates Ex Ante evaluation by allowing for informed argumentation and comparison with existing knowledge or design alternatives.

4.2.4.5 Ex Ante Evaluation (General)

This step evaluates the conceptual validity and relevance of the general solution prior to the design of specific artifacts. It is conducted as a formative artificial evaluation through informed argumentation by researchers. The objective is to ensure internal coherence, theoretical plausibility, and alignment with the established general problem and requirements.

Given the limited evaluative capacity of stakeholders at this stage, this evaluation is not intended for stakeholder engagement. Rather, it acts as a quality assurance checkpoint within the design process to confirm that the proposed solution is conceptually viable and offers a defensible basis for specific implementation.

4.2.4.6 Design Specific Solution

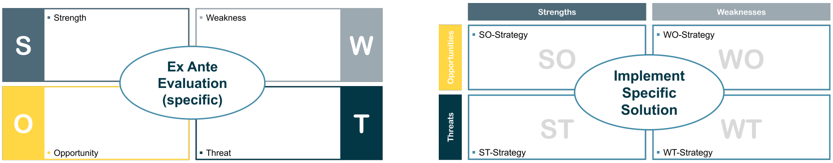
In this step, the general solution is instantiated into a specific artifact: the Synthesis Model of the Digital Ecosystem. The model is tailored to the previously identified requirements, incorporating structures, roles, and dynamics relevant to stakeholders. The focus is on conceptual clarity and communicability, rather than on technical implementation.

This activity transforms abstract design principles into a tangible representation that can be assessed and iteratively refined. It serves as the central artifact around which subsequent evaluations and stakeholder engagements are concentrated.

4.2.4.7 Ex Ante Evaluation (Specific)

This step involves an analytical evaluation of the designed Synthesis Model, focusing on its ability to meet the specified requirements. The evaluation is structured through a SWOT analysis to systematically assess both static and dynamic aspects of the artifact. Static qualities—such as conceptual clarity, completeness, and internal consistency—are categorized as strengths (S) and weaknesses (W). Conversely, opportunities (O) and threats (T) represent dynamic qualities that emerge from the expected application of the model within its context. The results are summarized with the template in Figure 4.5.

The evaluation is primarily formative and artificial, conducted by researchers through a detailed inspection of the model's structure and intended use. Where appropriate, it may also incorporate summative elements to inform design decisions and guide further refinements prior to deployment for stakeholders.



(a) Evaluation Template

(b) Implementation Strategy Template

Figure 4.5: Templates for SWOT Analysis.

4.2.4.8 Implement Specific Solution

The implementation of the Synthesis Model is directly informed by the findings from the preceding SWOT evaluation. Since the most significant design risks are user-centered, the implementation strategy must effectively present the artifact while ensuring that it fosters stakeholder understanding and engagement.

To address these challenges, tailored implementation strategies are developed using established SWOT logic, as presented with the template in Figure 4.5: leveraging strengths to exploit opportunities (SO), utilizing opportunities to mitigate weaknesses (WO), applying strengths to counter threats (ST), and minimizing weaknesses to avoid threats (WT). These strategies inform the presentation, communication, and integration of the model within its context of use. The result is a context-sensitive, strategically positioned artifact, prepared for evaluation under real-world conditions.

4.2.4.9 Ex Post Evaluation

This step evaluates the implementation of the synthesis model within its real-world context. The evaluation is naturalistic and summative, concentrating on both overall relevance and application-specific insights. Observations are conducted across the broader project landscape to assess how the model facilitates activities throughout IntWertL and engages various stakeholder groups.

To complement this general observation, selected applications are examined in greater depth. This layered evaluation approach—combining broad usage monitoring with targeted analysis—enables a nuanced understanding of the model's effectiveness in practice. Furthermore, it provides valuable insights across all three main evaluands: its relevance to stakeholder needs and contexts, its effectiveness in addressing ecosystem-level concerns, and its interpretability as a shared frame of reference.

4.2.5 Approach to Consolidating the Engineering Digital Ecosystems Methodology

This section outlines the process through which the EDE methodology is established as a Level 2 artifact. The methodology is based on the iterative design, implementation, and evaluation of preliminary versions of the Synthesis Model (Level 1 artifacts), which are developed across three iterations of the previously described design cycle. The methodological consolidation is grounded in both the empirical insights generated through this process and a reflective abstraction of key design activities.

The EDE methodology integrates specific elements of the SDSM, focusing on process steps relevant to practitioners. Steps associated with formative evaluation or the general solution space are excluded, as the Synthesis Model inherently encapsulates the general solution within the EDE context. Consequently, the methodology is presented as a ready-to-use guide for practitioners, with the potential for further refinement in future DSR efforts.

The consolidation approach, illustrated in Figure 4.6, is primarily inductive, leveraging insights from nine structured evaluation episodes alongside the iterative implementation of the complete design cycle. Recurring design decisions, representational challenges, and stakeholder interactions are synthesized into the core components of the EDE methodology: a systematic approach (EDE approach) and the Synthesis Model itself, which encompasses both a formal model representation and a practitioner-friendly model presentation.

The EDE methodology is designed for all stakeholders within Digital Ecosystems, including ORCHESTRATORS, CORE USERS (adopters), and CONTRIBUTORS (e. g., developers or service providers). Its consolidation is primarily justified by the performance and evaluative feedback from the Level 1 artifacts, as well as adherence to the design science principles of Hevner et al.

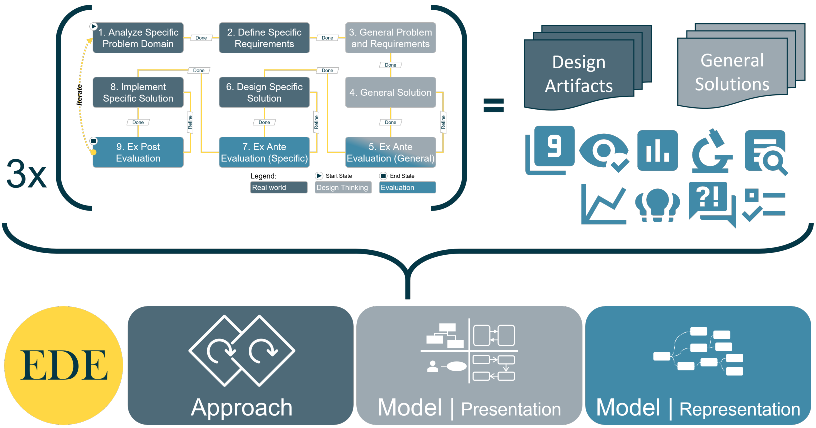

3x
1. Analyze Specific Problem Domain
2. Define Specific Requirements
3. General Problem and Requirements
4. General Solution
5. Ex Ante Evaluation (General)
6. Design Specific Solution
7. Ex Ante Evaluation (Specific)
8. Implement Specific Solution
9. Ex Post Evaluation
Done
Refine
Iterate
Legend:
Real world
Start State
Design Thinking
End State
Evaluation
=
Design Artifacts
General Solutions
EDE
Approach
Model | Presentation
Model | Representation

Figure 4.6: Approach to consolidate the EDE methodology.

(2004). A descriptive validation of the methodology—particularly concerning Hevner's guidelines—and its applicability is presented in Chapter 7.

4.3 Summary

Metaphorically, the research design can be viewed as a building: theory serves as the foundation; the three iterations of the Synthesis Model function as pillars; the EDE methodology spans these pillars, acting as a conceptual bridge that supports the forthcoming discussion chapter, which constitutes the roof, situating the entire structure within its broader scientific context, as illustrated in Figure 4.7.

This chapter outlines the research approach employed in this thesis, focusing on the iterative development and evaluation of Level 1 artifacts—namely, the Synthesis Models. These artifacts serve as the foundation for a broader methodological contribution: the EDE methodology.

The methodological foundation is grounded in the principles of Design Science Research (DSR) and is implemented through a practitioner-oriented

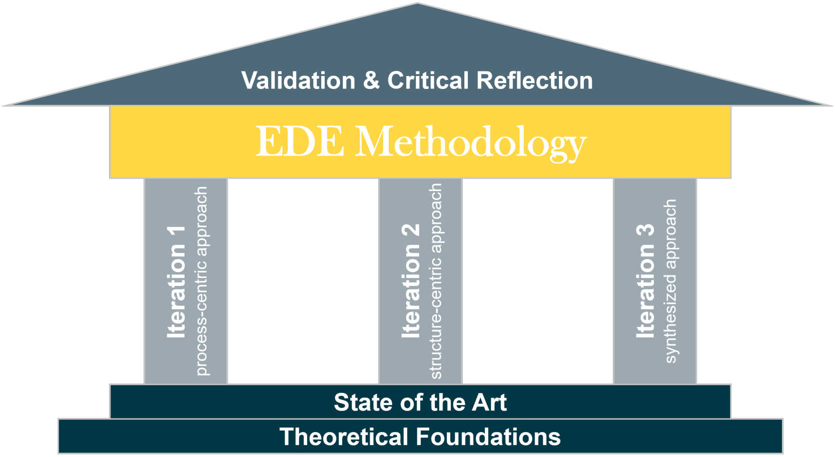


Figure 4.7: Metaphorical structure of the research approach: theory as foundation, Synthesis Model iterations as pillars, EDE methodology as bridging contribution, and the discussion chapter as conceptual roof.

adaptation of the Soft Design Science Methodology (SDSM). The selected process structure emphasizes conceptual clarity, iterative evaluation, and stakeholder integration. During three design iterations, nine structured evaluation episodes are conducted, guided by the Framework for Evaluation in Design Science (FEDS). These evaluations ensure that each version of the artifact contributes not only to addressing the design problem but also to generating transferable design knowledge.

The chapter elucidates how insights from Level 1 artifact design are integrated into the Level 2 artifact—the EDE methodology. This integration selectively incorporates relevant steps from the SDSM process for practitioners, while also integrating inductive insights derived from iterative cycles and evaluation data. The resulting methodology consists of both a generalized EDE approach and a structured representation and presentation of the generalized Synthesis Model itself.

Positioned within the framework of Gregor and Hevner (2013), the Level 1 artifact design is categorized as *invention*—characterized by low solution and application maturity—while the EDE methodology represents an *exaptation*, applying a well-refined solution to an emerging application domain. Consequently, this research contributes a Level 2 artifact validated through iterative Level 1 design work.

The following chapter presents the detailed results of the three design iterations, illustrating how each version of the Synthesis Model refines the problem space and informs the EDE methodology.

5 Iterative Digital Ecosystem Design

At the outset of the research project IntWertL, we aimed to rapidly establish a common frame of reference for the diverse stakeholders. Beginning with a process and requirements analysis, we planned to develop a *Synthesis Process Model* as the foundation for all subsequent tasks, including platform design and implementation, business model development, validation, and preparation for commercialization through network expansion. However, as demonstrated in this chapter, the initial draft did not adequately address all challenges. Consequently, we quickly recognized the necessity for continuous effort and an iterative approach to develop this common frame of reference.

This chapter discusses the three main iteration cycles of the Synthesis Model. First is the process-centric approach (5.1) which was initially deemed sufficient but did not adequately consider or depict certain aspects of the system architecture. These aspects were addressed in the second iteration, the structure-centric approach (5.2). Finally, we designed a synthesized solution (see section 5.3), as neither approach provided sufficient information. The existence of two models side by side resulted in a lack of a single source of truth and, consequently, the absence of a common frame of reference. The presentation of these three iterations aims to yield Level 1 validation of the Level 2 results presented in the following chapter 6. Therefore, the focus is not on in-depth descriptions of the specific solutions of the Synthesis Models, but rather on the formative and summative evaluation results that informed the EDE methodology. This chapter concludes in a brief summary (5.4).

5.1 Iteration 1: Process-Centric Approach

The results of the first iteration of the Synthesis Model have been published in Bönsch et al. (2023). This section integrates our findings into the broader context of this DSR study and demonstrates the alignment within the utilized SDSM.

At the outset of the IntWertL research project, our goal was to capture and model the fundamentals of the *IntWertL Ecosystem*. Given that research on the application domain of cooperative small-scale vehicle production and the design of a Digital Ecosystem are both in their early stages, this endeavor represents a true innovation challenge. Therefore, it is essential to identify the primary stakeholders and define the use cases of the ecosystem prior to considering an operational model, a technical architecture, or a governance structure.

Consequently, we initially concentrated on defining a common vision statement and a definition of CORE VALUES. This was subsequently employed for the identification of CORE PARTICIPANTS and the synthesis of CORE SERVICES. These elements are essential for establishing the strategic foundations of the ecosystem and creating a common frame of reference.

However, the *Synthesis (Process) Model V1* presented here (as a result of the first iteration) was developed within the IntWertL work package *1100 - Process & requirements analysis in platform development*. This work package has a distinct focus on the process view, which is clearly demonstrated by the main tasks and expected results detailed in the grant agreement (see table 5.1). Additionally, the Synthesis Model was directly integrated into the work package *4100 - Process modeling, management, & automation (documentation, validation, homologation & release)*. All of this placed the processes at the center of consideration in this first iteration.

Table 5.1: Tasks and Results from IntWertL work package 1100

Tasks	Results
The tasks involve: • Analyzing requirements and identifying user groups such as customers, engineers, and manufacturers. This includes both platform users and external parties. • Defining use cases and understanding how processes and workflows interact and communicate. • Identifying how workflows can be grouped or consolidated. • Determining roles, views, and interaction mechanisms to enable collaboration.	**The results of the analysis include:** • Identification of key stakeholders. • Documentation of processes and workflows. • Definition of the user environment and their interactions with the platform. • Specification of engineering and production services. • Establishment of functional requirements for the engineering and production platform.

The following subsections will systematically discuss each of the nine steps of the SDSM. To establish a common frame of reference for development tasks, the time allocated for designing the Synthesis Model was limited: work commenced in November 2022, and the finalized model was presented at the end of January 2023. This constraint necessitated a focus on efficiency, as detailed in chapter 4. However, the summative evaluation (Ex Post Evaluation) did not need to be completed by January and was conducted subsequently.

5.1.1 Analyze Specific Problem Domain

A more general introduction to the problem domain, IntWertL, is provided in section 4.2.2. This section focuses on describing the actions taken to analyze this domain. The core of the problem domain arises from the combination of limited resources, diverse stakeholder perspectives, and an unclear understanding of the requirements for the IntWertL platform. Addressing these challenges necessitates a comprehensive approach to gathering and synthesizing stakeholder input, aligning expectations, and developing a unified vision for the platform. Moreover, the lack of a clear and standardized framework for platform development—particularly in ecosystem design—exacerbates the situation.

To address these challenges, a combination of complementary methods was employed, which are discussed in more detail in Bönsch et al. (2023):

- **Process Exploration:** Interviews were conducted with representatives from SMEs in the automotive sector. Information was gathered using text-based documentation and process modeling with PASS. In the second step, the individual results were analyzed and contrasted.
- **Exploratory User Stories:** Over 200 user stories were submitted by stakeholders, capturing their individual needs and expectations in a structured template. The elicited user stories were subsequently completed, refined, unified, and prioritized.
- **Vision Workshops:** Collaborative sessions were organized to establish a knowledge base for a unified vision of the platform's functionality and long-term objectives.
  - **Customer Journey:** A workshop format designed to map the end-to-end experience of stakeholders interacting with the platform. This helps identify pain points, opportunities for improvement, and critical touchpoints where the platform can deliver maximum value.

- **Walt Disney Method:** A creative problem-solving workshop that adopts three distinct perspectives: *Dreamer*, *Realist*, and *Critic*. All IntWertL partners assumed the Dreamer role for their company to explore innovative ideas, evaluate their feasibility, and address potential challenges systematically.

The collected information was subsequently organized into a mind map (see Figure 5.1), where statements from the workshops were matched to the User Stories and clustered. Particular emphasis was placed on ensuring traceability back to the origin of the statements. The outcome of the analysis consists of several PASS process model fragments[1], a spreadsheet containing over 200 User Stories, and a visual representation of approximately 1000 objects that represent the specific problem domain. All of this is then utilized for the definition of specific requirements.

5.1.2 Define Specific Requirements

First, the information from the analysis was utilized to identify and define the most important roles and processes for IntWertL. The definitions are presented in Table 5.2 and Table 5.3. The roles can be divided into two groups. The first group consists of roles that are independent of the ecosystem and its operations, namely *Customers* and *OEMs*. The second group includes the up to this point already identified set of CORE PARTICIPANTS.

The processes can be categorized into three groups based on the role organizations play in the ecosystem. First, if an organization is not yet a participant, it can engage in the *Registration Process*. Second, Platform Participants are involved in the *Request/Order Process*, including the *Requirements Management Process*, and the *Network Configuration Process* to become *Contract*

[1] As Process Exploration was only used to analyze the specific problem domain, most models are incomplete, lack semantical and/or syntactical correctness. However, they still can be useful for improved understanding of the problem domain.

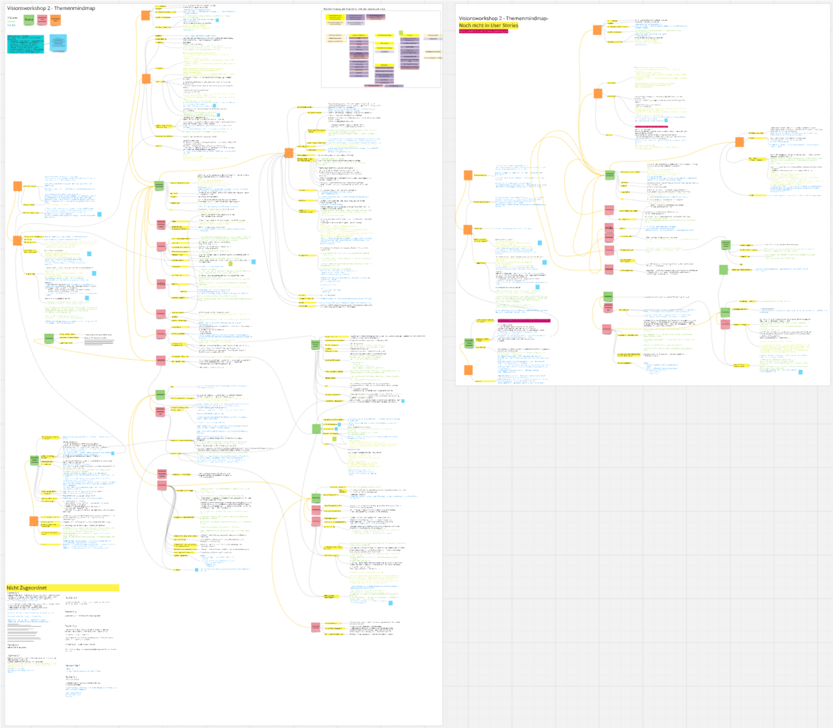

Visionsworkshop 2 - Themenmindmap
Visionsworkshop 2 - Themenmindmap-
Noch nicht in User Stories
Nicht Zugeordnet

Figure 5.1: Visual representation of the collected information on the problem domain. Clustering of statements from the Walt Disney Method vision workshop to overarching topics and elicited User Stories in a mind map.

Participants. These *Contract Participants* can subsequently participate in the *Order Management Processes*, namely *Controlling/Monitoring Processes*, *Engineering Processes*, *Production Processes*, and *Support Processes*. The Synthesis Model aims to illustrate the interconnections between these processes and the interactions among the various roles within the ecosystem.

Additionally, the information obtained from the analysis step is used to formulate the initial iteration of a vision statement. The vision represents the long-term objective of IntWertL. It does not address technical or operational

Table 5.2: I1: Initial set of roles and their definitions in context of IntWertL

Role	Definition
Customer	End User of the vehicle. Not necessarily a platform participant (e. g., municipalities, SMEs, etc.).
OEM	Original Equipment Manufacturer. Brand responsibility. A legal entity, potentially a consortium of multiple organizations (e. g., partnerships).
CORE PARTICIPANTS	
Platform Participant	Legal entity registered on the platform (e. g., sole proprietorships, partnerships, etc.).
Service Provider	Platform participant that offers solutions on the platform.
Integrator	Contract participant. Information process and organization of contract participants. Communication with the customer side.
Contract Participant	Platform participants selected in the network configuration.

implementations. The vision serves as a guiding orientation and a motivating force that informs actions and decisions. Visions are often inspiring and ambitious, conveying a clear meaning and purpose. It differs from specific goals in its general and far-reaching nature.

This first iteration of the vision is generated in various representations for distinct purposes, as depicted in Figure 5.2. The first is a textual description of the vision for IntWertL. Therefore, the mind map is refined, leading to the identification of three main topics. These concepts are then articulated in writing, resulting in a brief text[2] created with input from experts on the identified topics. From this, an even more concise representation is developed in the form of a One Pager. Another key representation combines visualization and storytelling to enhance engagement. Initially, a storyboard is produced.

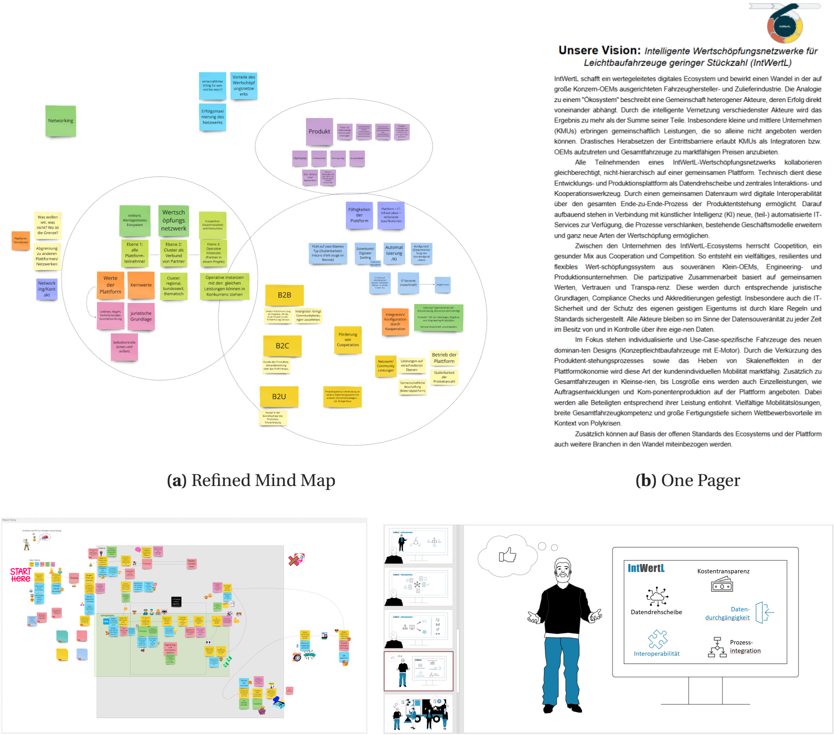
[2] Word Count: 1171 (less than four DIN-A4 pages).

Table 5.3: I1: Processes and their definitions on the platform

Process	Definition
Registration Process	Acceptance of the modal framework. Fulfillment of standards to become a platform participant.
Request/Order Process	Creation of requests or orders on the platform.
• Requirements Management Process	Sub-process of the order process. Coordination of customer requirements with the available services.
Network Configuration Process	Selection of contract participants, considering the coordinated requirements from platform participants.
Order Management Processes	Initiating and orchestrating the subsequent processes of an order.
• Controlling Processes/ Monitoring Processes	Tracking order execution (status, visualization, costs, time, etc.).
• Engineering Processes	Providing engineering services (development, design, simulation, etc.).
• Production Processes	Providing production services (manufacturing, assembly, etc.).
• Support Processes	Non-value-adding services that provide added value only in conjunction with engineering or production processes (e. g., sales, after-sales, certification, quality management, etc.).

This narrative outlines the journey from concept to solution, illustrating how IntWertL facilitates this process, with a particular focus on the benefits experienced by personified stakeholders and their emotions at each stage. To enhance presentation, this Vision Story is formatted as a slideshow.

The specific requirement is to integrate the roles and processes, illustrating their interactions and how they contribute to the vision of IntWertL. The emphasis is on the processes executed on the IntWertL platform. Given the number and diversity of participants in IntWertL, the outcome must be easily understandable and self-explanatory.



Unsere Vision: *Intelligente Wertschöpfungsnetzwerke für Leichtbaufahrzeuge geringer Stückzahl (IntWertL)*

IntWertL schafft ein wertegeleitetes digitales Ecosystem und bewirkt einen Wandel in der auf große Konzern-OEMs ausgerichteten Fahrzeughersteller- und Zulieferindustrie. Die Analogie zu einem "Ökosystem" beschreibt eine Gemeinschaft heterogener Akteure, deren Erfolg direkt voneinander abhängt. Durch die intelligente Vernetzung verschiedenster Akteure wird das Ergebnis zu mehr als der Summe seiner Teile. Insbesondere kleine und mittlere Unternehmen (KMUs) erbringen gemeinschaftlich Leistungen, die so alleine nicht angeboten werden können. Drastisches Herabsetzen der Eintrittsbarriere erlaubt KMUs als Integratoren bzw. OEMs aufzutreten und Gesamtfahrzeuge zu marktfähigen Preisen anzubieten.

Alle Teilnehmenden eines IntWertL-Wertschöpfungsnetzwerks kollaborieren gleichberechtigt, nicht-hierarchisch auf einer gemeinsamen Plattform. Technisch dient diese Entwicklungs- und Produktionsplattform als Datendrehscheibe und zentrales Interaktions- und Kooperationswerkzeug. Durch einen gemeinsamen Datenraum wird digitale Interoperabilität über den gesamten Ende-zu-Ende-Prozess der Produktentstehung ermöglicht. Darauf aufbauend stehen in Verbindung mit künstlicher Intelligenz (KI) neue, (teil-) automatisierte IT-Services zur Verfügung, die Prozesse verschlanken, bestehende Geschäftsmodelle erweitern und ganz neue Arten der Wertschöpfung ermöglichen.

Zwischen den Unternehmen des IntWertL-Ecosystems herrscht Coopetition, ein gesunder Mix aus Cooperation und Competition. So entsteht ein vielfältiges, resilientes und flexibles Wert-schöpfungssystem aus souveränen Klein-OEMs, Engineering- und Produktionsunternehmen. Die partizipative Zusammenarbeit basiert auf gemeinsamen Werten, Vertrauen und Transpa-renz. Diese werden durch entsprechende juristische Grundlagen, Compliance Checks und Akkreditierungen gefestigt. Insbesondere auch die IT-Sicherheit und der Schutz des eigenen geistigen Eigentums ist durch klare Regeln und Standards sichergestellt. Alle Akteure bleiben so im Sinne der Datensouveränität zu jeder Zeit im Besitz von und in Kontrolle über ihre eige-nen Daten.

Im Fokus stehen individualisierte und Use-Case-spezifische Fahrzeuge des neuen dominan-ten Designs (Konzeptleichtbaufahrzeuge mit E-Motor). Durch die Verkürzung des Produktent-stehungsprozesses sowie das Heben von Skaleneffekten in der Plattformökonomie wird diese Art der kundenindividuellen Mobilität marktfähig. Zusätzlich zu Gesamtfahrzeugen in Kleinse-rien, bis Losgröße eins werden auch Einzelleistungen, wie Auftragsentwicklungen und Kom-ponentenproduktion auf der Plattform angeboten. Dabei werden alle Beteiligten entsprechend ihrer Leistung entlohnt. Vielfältige Mobilitätslösungen, breite Gesamtfahrzeugkompetenz und große Fertigungstiefe sichern Wettbewerbsvorteile im Kontext von Polykrisen.

Zusätzlich können auf Basis der offenen Standards des Ecosystems und der Plattform auch weitere Branchen in den Wandel miteinbezogen werden.

(a) Refined Mind Map

(b) One Pager

(c) Storyboard of the Vision Story

(d) Vision Story as slideshow

Figure 5.2: I1: Illustration of the vision of IntWertL.

5.1.3 General Problem and Requirements

The general problem is to establish a unified frame of reference for all stakeholders within a Digital Ecosystem. Crucially, clarity is required regarding CORE USERS and CORE SERVICES as well as their interactions. The Synthesis Model is not designed to directly represent system architecture, business models, or workflows. It is not required to be machine-interpretable and is intended solely to facilitate discussions among human participants.

The general requirements for the Synthesis Model (Iteration 1) are defined as follows:

I1.1 **Display Active Entities:** Identify and represent active entities involved in the Digital Ecosystem, especially CORE USERS.

I1.2 **Support Complex Process Systems:** Precisely describe interaction patterns and processes within the complex environment that is a Digital Ecosystem.

I1.3 **Ensure Interpretability:** Make the model self-explanatory, ensuring stakeholders can easily comprehend the roles, processes, and interactions. The model must be clear and interpretable without requiring technical expertise, enabling its use in discussions among diverse stakeholders.

5.1.4 General Solution

Subject orientation has been demonstrated to be an effective strategy for modeling complex process systems, particularly from a people-centric perspective. Consequently, the initial version of the Synthesis Model is also developed using PASS. An introduction to subject-orientation and PASS is provided in section 2.3. This model employs subject-oriented principles, emphasizing active entities, or subjects, to highlight interactions in processes. It yields a process model that focuses on engagement opportunities within the Digital Ecosystem, rather than on system architecture or behavior. As a result, only SIDs are created, while SBDs are omitted.

This general solution method resembles conventional approaches to subject-oriented modeling and involves three main tasks:

- **Subject Delineation:** Clear delineation between active subjects and passive objects ensures role clarity. First, the CORE USERS are modeled, then any other (external) subjects that are needed. To be able to describe multilateral communication in a simpler way, the concept of subject groups is introduced. Most introduced subjects in the Synthesis Model should represent organizations. However, PASS allows subjects to represent people or technology (IT-systems) as well.

- **Explicate Communication:** Define the interaction that is associated with the main processes. Structure communication flows to improve process transparency. Keep in mind any restrictions that might apply to the communication as to the means of message transmission in the Digital Ecosystem. Especially data sovereignty concerns might lead to initially not intuitive communication patterns.
- **Conceptualize Abstraction:** PASS yields flexible and powerful expression capabilities to meet the dynamic requirements of complex socio-technical systems. It provides the possibility of combining different logical layers into one model. This task includes deciding what level of detail can still be put into one representation and when to use a subject group or system interface to separate concerns (creating a connected SID). Another concern is to identify the linear or cyclical behavior of subjects to properly combine them into one coherent model. Additionally, this task is integral for introducing any (multi-layered) concepts from ALPS.

As is customary in process modeling, these tasks are not executed sequentially, but rather iteratively. Any modeling decision made in one task can and will impact the other tasks. The outcome is a singular PASS model that describes the interactions of CORE USERS within one SID (or across multiple interconnected SIDs).

5.1.5 Ex Ante Evaluation (General)

The Ex Ante Evaluation of the general solution is conducted as an informed argument by modeling experts. The results are presented in Table 5.4.

Table 5.4: I1: Matching of requirements to general solution

Requirement	Solution
I1.1 Display Active Entities	• Subject-orientation necessitates placing active entities (subjects) at the forefront of consideration. • Using SIDs only always displays active entities and never just workflows. • Subject groups allow to model multilateral communication and collaboration as an active entity.
I1.2 Support Complex Process Systems	• PASS is a complete and formal process modeling language that allows the modeling of any process, no matter how complex. • PASS allows for linear and cyclical behavior in one model. • Possibility to describe interaction within the Digital Ecosystem, as well as with external subjects within one model. • Multiple abstraction mechanisms, including ALPS, if needed.
I1.3 Ensure Interpretability	• Easy to understand through alignment with widespread thinking structures and the use of only five basic elements (even less with only SIDs). • Explicit modeling of communication improves comprehensibility. • Natural context separation and separation of concerns, through abstraction mechanisms, allows to focus on the relevant sections for each stakeholder.
Note: The advantages of PASS are based on Elstermann (2020, p. 138f)	

In addition to this descriptive evaluation method, various partners in the research project were introduced to the concepts of subject-orientation and the PASS modeling language to identify any reservations. However, no objections to this approach were noted. Furthermore, this preliminary evaluation resulted in workshops with consortium members, who found the subject-orientation paradigm and/or the PASS modeling language relevant to their own research interests. These workshops focused on the feasibility and usability of the subject-oriented approach. At the end of the workshops, participants were asked to identify the benefits they perceived in PASS and the application of subject-orientation in the context of IntWertL. Although some participants did not see any direct applications for this paradigm, the feedback provided was valuable in facilitating acceptance of the overall solution approach. Subsequently, PASS was defined as the go-to process modeling language for the entire research project.

5.1.6 Design Specific Solution

This section outlines the design of the specific PASS Synthesis Model V1 for IntWertL. Initially, an overview of the design process and the tools employed is presented. Subsequently, the artifact itself is described. To ensure alignment with the general solution and to separate concerns, the Synthesis Model consists of two SIDs: the *Platform SID* and the *Project Instance SID*.

5.1.6.1 Design Process

In the design of the Synthesis Model, we employed a pair modeling approach, leveraging the collaborative efforts of domain experts and modeling experts. Using Microsoft Visio with a custom set of stencils for the PASS modeling language, we iteratively developed and refined the model in real time. This method ensured high accuracy, immediate validation, and seamless integration of domain knowledge into the model. The collaborative nature of pair modeling improved efficiency by minimizing misunderstandings and fostering knowledge exchange. To further enhance intuitive interpretability, we consistently varied the modeling pairs between sessions. This approach ensured that the language used was intuitively understandable to all participants. As an in-depth analysis of the problem domain was conducted beforehand, most modeling pairs consisted of two modeling experts with a deep understanding of the collected information.

Especially since this is the first iteration of the Synthesis Model, the main objective was to unify the various perspectives on the Digital Ecosystem while still maintaining a significant solution space for further detailing CORE SERVICES and developing appropriate SOLUTIONS.

5.1.6.2 Platform SID

Figure 5.3 presents the first SID of the Synthesis Model V1. This diagram provides an overview of the Digital Ecosystem and the communication facilitated by the IntWertL platform. Since the general solution does not intend to use SBDs, most entities are modeled as Interface Subjects. However, the base platform and the Project Instance are represented as subject groups to highlight multilateral communication and collaboration.

The *Platform* subject group includes the following subjects: *Platform Participant*, *Integrator*, the IT systems *Marketplace Portal*, *Onboarding System*, and additional *Services*, as well as the *Project Instance* subject group. The Platform SID of the Synthesis Model illustrates how each project is processed on the IntWertL platform. As the platform is designed to manage multiple projects simultaneously, individual Project Instances are created for each project. One Project Instance includes the subjects *Project Instance Management* and *Project Participants -Engineering*, *-Production*, and *-Others*. No messages are depicted within the Project Instance, as that communication is handled in a separate SID. The *Customer* and *(Platform) External Participant* participate in the overall process but have not completed onboarding

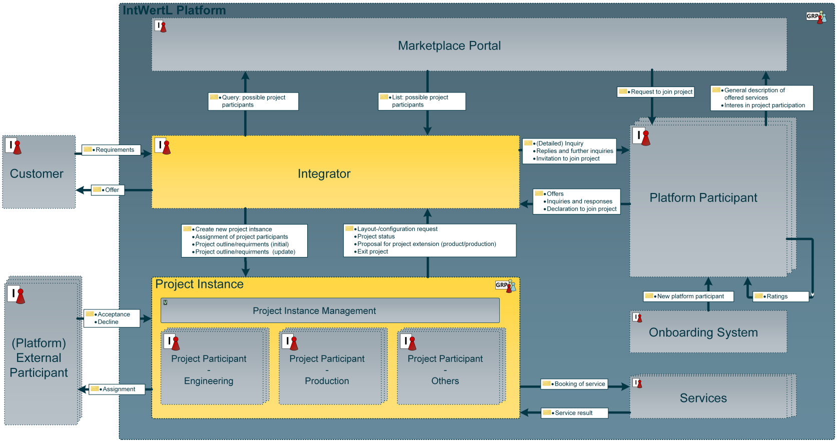

IntWertL Platform
GRP
Marketplace Portal
•Query: possible project participants
•List: possible project participants
•Request to join project
•General description of offered services
•Interes in project participation
Customer
•Requirements
•Offer
Integrator
•(Detailed) Inquiry
•Replies and further inquiries
•Invitation to join project
•Offers
•Inquiries and responses
•Declaration to join project
Platform Participant
•Create new project intsance
•Assignment of project participants
•Project outline/requirments (initial)
•Project outline/requirments (update)
•Layout-/configuration request
•Project status
•Proposal for project extension (product/production)
•Exit project
Project Instance
GRP
Project Instance Management
(Platform) External Participant
•Acceptance
•Decline
•Assignment
Project Participant - Engineering
Project Participant - Production
Project Participant - Others
•New platform participant
•Ratings
Onboarding System
•Booking of service
•Service result
Services

Figure 5.3: Synthesis Model V1 - Platform SID

to the platform. Therefore, communication with these subjects is not directly supported by the IntWertL platform.

All participants on the platform engage in an onboarding process facilitated by the onboarding system. This process encompasses technological, legal, and procedural aspects. At the core of the entire project process is the Integrator. The Integrator is a registered platform participant who acts as a mediator, representing the entire platform's value network to the customer.

The marketplace portal is utilized for the network configuration process. Platform participants can list their products and services within this portal. The integrator can subsequently search for appropriate project participants and initiate the request/order and requirement management processes. The marketplace portal sends participation requests to potential project participants, who respond with their expressions of interest. The integrator then finalizes the participation details and confirms either acceptance or rejection. This interaction is modeled in principle rather than in detail to maintain flexibility in the solution space. For example, subsequent integration of artificial intelligence into these processes is feasible at this level of abstraction but could conflict with the model if it were more detailed.

The integrator establishes a project instance on the platform. Within the project instance, participants are organized into groups according to their roles (e. g., engineering, production). The integrator communicates customer requirements and updates, while participants provide design suggestions, progress reports, and proposed modifications. The platform supports managing inquiries, project extensions, and participant departures. In the future, the integration of AI and automation will streamline these processes.

The platform enables contributors to offer additional services, including instant quoting, 3D viewers, lifecycle assessment tools, and others. External participants can also be incorporated into a project. For instance, an integrator may include a tenured business partner in a project, even if they are not a participant on the platform. However, the platform services are not accessible to that external participant, and another project participant must assume the responsibility of updating the project status for the external participant.

5.1.6.3 Project Instance SID

Figure 5.4 illustrates the interactions within a project instance, showcasing communication between order management and participants from engineering, production, and other domains. Order management conveys conditions such as budget and deadlines, while participants respond with reports, design proposals, and issue alerts. Engineering and production participants exchange design and configuration requests to facilitate seamless collaboration. Production participants coordinate timelines, share production capabilities, and provide quality feedback, thereby creating an essential feedback loop for product refinement. These messages represent a more detailed breakdown of the order management processes.

The interorganizational design of vehicles necessitates close coordination between engineering and production. Engineering submits procurement and production requests while considering manufacturability. Production

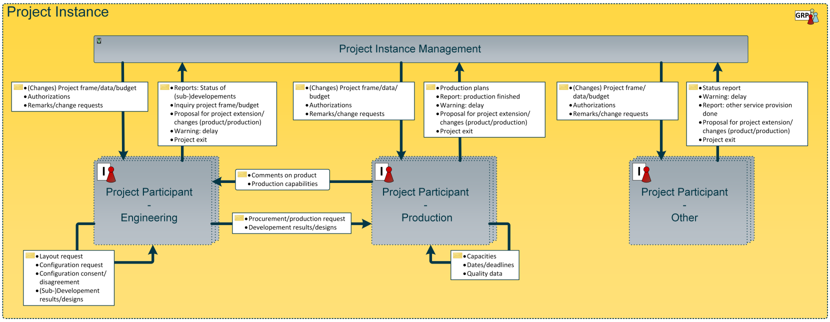

Project Instance
GRP
Project Instance Management
• (Changes) Project frame/data/budget
• Authorizations
• Remarks/change requests
• Reports: Status of (sub-)developements
• Inquiry project frame/budget
• Proposal for project extension/ changes (product/production)
• Warning: delay
• Project exit
• (Changes) Project frame/data/ budget
• Authorizations
• Remarks/change requests
• Production plans
• Report: production finished
• Warning: delay
• Proposal for project extension/ changes (product/production)
• Project exit
• (Changes) Project frame/ data/budget
• Authorizations
• Remarks/change requests
• Status report
• Warning: delay
• Report: other service provision done
• Proposal for project extension/ changes (product/production)
• Project exit
I
Project Participant - Engineering
• Comments on product
• Production capabilities
• Procurement/production request
• Developement results/designs
I
Project Participant - Production
I
Project Participant - Other
• Layout request
• Configuration request
• Configuration consent/ disagreement
• (Sub-)Developement results/designs
• Capacities
• Dates/deadlines
• Quality data

Figure 5.4: Synthesis Model V1 - Project Instance SID

offers feasibility feedback and refines the development process. Production participants also exchange information regarding production capabilities, coordinate manufacturing timelines, and communicate quality assessments.

This synthesis model serves as the foundation for designing IntWertL SOLUTIONS. It illustrates the efficient integration of multiple stakeholders within the Digital Ecosystem, while preserving flexibility for future automation and AI-driven optimizations.

5.1.7 Ex Ante Evaluation (Specific)

The specific Ex Ante Evaluation is analytical in nature. First, Table 5.5 and Table 5.6 illustrate how identified roles and processes are represented in the Synthesis Model V1. Subsequently, further results are categorized as strengths (S), weaknesses (W), opportunities (O), and threats (T). In this context, strengths and weaknesses reflect static qualities, representing inherent outcomes of the model's structure. Conversely, opportunities and threats capture dynamic qualities, representing outcomes derived from the application of the artifact within the relevant domain.

Table 5.5: I1: Alignment of initial roles with representations in Synthesis Model V1

Role	Representation
Customer	Subject: *Customer*
OEM	Not directly shown in the model. However, most Integrators will take on the role of OEM for the outcomes of their projects.
CORE PARTICIPANT	
Platform Participant	Multi-Subject: *Platform Participant*
Service Provider	Multi-Subject: *Platform Participant* and its Message: *General description of offered services.*
Integrator	Subject: *Integrator*
Contract Participant	Subjects: *Project Participant -Engineering, -Production,* and *-Others.*

5.1.7.1 Strengths

- **Simplicity:** The model consists of only two elements. Boxes represent active entities (subjects), and connectors with labels are messages or data objects. Splitting into two SIDs makes each of them comprehensible. Only having two diagrams, which are clearly coupled and visually color-coded, makes keeping the overview easy.

- **Completeness:** All of the identified CORE PARTICIPANTS are clearly represented in the model. Additionally, platform external roles are modeled as well. The main components of interaction for each of the identified platform processes are included, even though they are not directly interconnected and have different levels of abstraction.

- **Open Solution Space:** The model creates room for innovation and original thought, yet remains detailed enough to maintain a shared framework. By avoiding the prescription of exact SOLUTIONS, it allows for adaptability and encourages the integration of distinct benefits and novel ideas.

Table 5.6: I1: Alignment of initial processes with representations in Synthesis Model V1

Process	Representation
Registration Process	Subject: *Onboarding System*
Request/Order Process	Messages between *Integrator* and *Platform Participant*
• Requirements Management Process	Messages: *Project outline/requirements (initial)*, and *Project outline/requirements (update)*
Network Configuration Process	Subject: *Marketplace Portal*, Messages: *Assignment of project participants*, *Proposal for project extension (Product/Production*, and *Exit project*.
Order Management Processes	Subject group: *Project Instance*
• Controlling Processes/ Monitoring Processes	Subject: *Project Instance Management*
• Engineering Processes	Subject: *Project Participant - Engineering*
• Production Processes	Subject: *Project Participant - Production*
• Support Processes	Subject: *Project Participant - Others*

5.1.7.2 Weaknesses

- **Lack of detail:** Without SBDs a definitive sequence of message exchanges is not illustrated. Additionally, significant subject-specific information could be integrated through SBDs. Some supplementary details are provided via comments linked to shapes. However, these are not readily accessible to all users. Moreover, additional informal descriptions are not included in the Synthesis Model V1; these could encompass examples or images. This results in an abstract process model description that on its own does not tell a story (such as a customer journey).

- **No structural information:** The model outlines an overarching process. It is challenging to delineate distinct functional blocks or use cases. Enabling services are represented only by the platform subject group and are not explicitly articulated. At best, roles and functions are conflated; for instance, the marketplace portal and the provider of the marketplace portal are depicted as a single entity.

5.1.7.3 Opportunities

- **Common frame of reference wanted:** From the outset of the research project it became clear to all stakeholders that a common frame of reference is needed for effective collaboration. As stakeholders from all domains are already waiting for this, an initial acceptance is expected.
- **No prevalent modeling language among stakeholders:** Subject-oriented approaches are not widespread and therefore often considered a threat. However, here no (semi) formal modeling language is common among the stakeholders (mostly SMEs). As PASS is also used in other work packages and no alternatives are proposed, no closed mindset in regards to the modeling technique is expected.

5.1.7.4 Threats

- **Innovation:** Design of a Digital Ecosystem is a real innovation. Therefore, a general understanding of the fundamentals of EDE can not be anticipated among stakeholders. Neither a common vision for the ecosystem nor its solutions are well established. Nevertheless, the aim is to disruptively change small-scale vehicle production. This change, as with every major disruption, will produce reservations of some kind.
- **Paywall:** The Synthesis Model V1 is created in Microsoft Visio as a .vsdx-file. This is proprietary and cannot be viewed or changed properly without the appropriate software. By exporting to images, some additional information (such as the comments mentioned above) can be lost.

5.1.7.5 Summary Ex Ante Evaluation (Specific)

These SWOT are summarized in Figure 5.5. The specific solution is true to the general solution. All elements defined as specific requirements have been integrated into a single model. The main weaknesses identified stem from design decisions, including a process-centric approach and the exclusion of SBDs. These same decisions have resulted in strengths for the specific solution, such as simplicity and an open solution space, as well as strengths for the general solution, such as interpretability. Analyzing both static and dynamic qualities facilitated the search process for the specific solution. Furthermore, the SWOT analysis is employed to define the implementation strategy for the artifact.

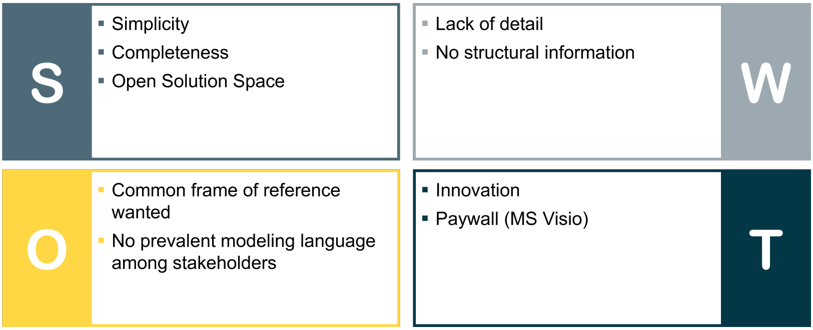
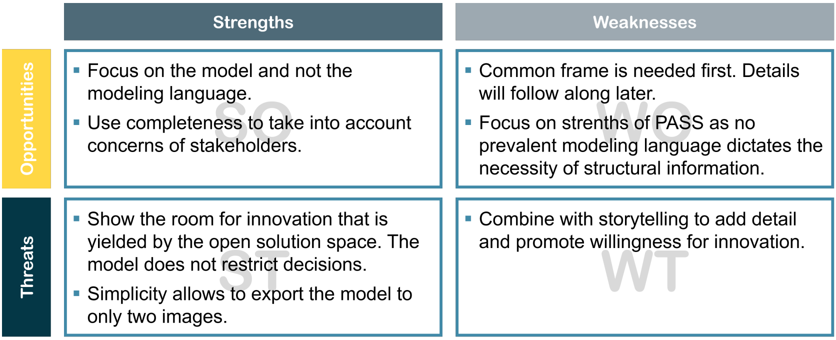

S
Simplicity
Completeness
Open Solution Space
W
Lack of detail
No structural information
O
Common frame of reference wanted
No prevalent modeling language among stakeholders
T
Innovation
Paywall (MS Visio)

Figure 5.5: Synthesis Model V1 - SWOT

5.1.8 Implement Specific Solution

The primary design risk for the Synthesis Model is user-oriented. Consequently, both the artifact and its implementation strategy must be meticulously designed. Based on the SWOT analysis, various implementation strategies [3] for the Synthesis Model are outlined in Figure 5.6.



	Strengths	Weaknesses
Opportunities	SO ▪ Focus on the model and not the modeling language. ▪ Use completeness to take into account concerns of stakeholders.	WO ▪ Common frame is needed first. Details will follow along later. ▪ Focus on strenths of PASS as no prevalent modeling language dictates the necessity of structural information.
Threats	ST ▪ Show the room for innovation that is yielded by the open solution space. The model does not restrict decisions. ▪ Simplicity allows to export the model to only two images.	WT ▪ Combine with storytelling to add detail and promote willingness for innovation.

Figure 5.6: Synthesis Model V1 - Implementation strategies

The model is introduced to the stakeholders, specifically the partners from IntWertL, in three representations:

First, the digital Visio model and its exports to PNG and PDF formats. Second, it was printed as a DIN A0 poster. Third, as a PowerPoint presentation. This presentation is part of the general assembly in January 2023 and serves as the inaugural introduction of the Synthesis Model to all stakeholders. The presentation directly follows an enactment of the vision story. No additional introduction to PASS or subject-orientation is provided to focus solely on the Synthesis Model and its content. As the vision story is personalized for the stakeholders, their buy-in regarding the "how to achieve this

[3] **SO strategies** use strengths to take advantage of opportunities. **WO strategies** overcome weaknesses by using opportunities. **ST strategies** use strengths to mitigate threat. **WT strategies** minimize weaknesses to avoid threats.

vision," as depicted in the model, is facilitated. Constant cross-referencing to the vision and other stakeholders is aimed at further enhancing this buy-in. After establishing this common framework, a lunch break is scheduled. This allows stakeholders to discuss opportunities for innovation within the solution space and further details of the processes presented.

After this introduction, the Synthesis Model should facilitate communication among stakeholders from various domains. It serves as the foundation for discussing the requirements of the first MVP which is scheduled for presentation in September 2023.

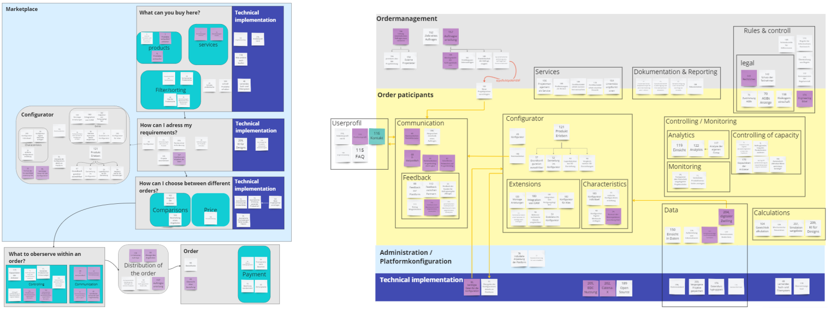
5.1.9 Ex Post Evaluation

As part of the first iteration, an observational field study was conducted to examine the practical application of the Synthesis Model V1. Its general relevance was evident as it was utilized across all active IntWertL work packages. This included various aspects such as engagement with ecosystem requirements, implementation of a software MVP, shaping a business model, and onboarding new participants. This breadth of applications underscores the versatility of Synthesis Model V1. However, particularly because the MVP represented the first significant milestone of the project, its main area of application was the development of SOLUTIONS. Consequently, the primary focus of this Ex Post Evaluation rested on two development-related aspects:

1. **User Story Elaboration:** systematic grouping and generation of user stories to guide SOLUTION development
2. **Scenario Process Model:** creation and refinement of a scenario process model to define MVP functionalities

5.1.9.1 User Story Elaboration

The analysis of the problem domain for this first iteration, in Section 5.1.1, involves the elicitation and prioritization of over 200 user stories. Bönsch et al. (2023) describe in detail how the Synthesis Model V1 is used for the further elaboration of these user stories. This approach facilitated a clearer understanding of how each proposed feature fits into the overall process framework provided by the Synthesis Model, thereby streamlining strategic decision-making and guiding subsequent implementation steps. The approach by Bönsch et al. (2023) yields graphical elaborations of the user stories, clustered by the reference frame introduced by the Synthesis Model. These mappings are illustrated in Figure 5.7. The two clusters, *Project Initiation* and *Project Instance*, were chosen for detailed visualization as they encompass the majority of the highly prioritized user stories.



(a) Project Initiation

(b) Project Instance

Figure 5.7: Graphical Elaboration of User Stories based on Bönsch et al. (2023)

- **Project Initiation** involves presenting the *Marketplace* user stories highlighted in pale blue. This includes topics like platform offerings, customer request entries, a product configurator, and offer decision guidance. Arrows show the possible action sequence, while dark blue boxes summarize technical details. Outside the pale blue section, three gray boxes represent user stories on collaboration steps, informal communication between integrators and platform participants, and order placement and payment features.

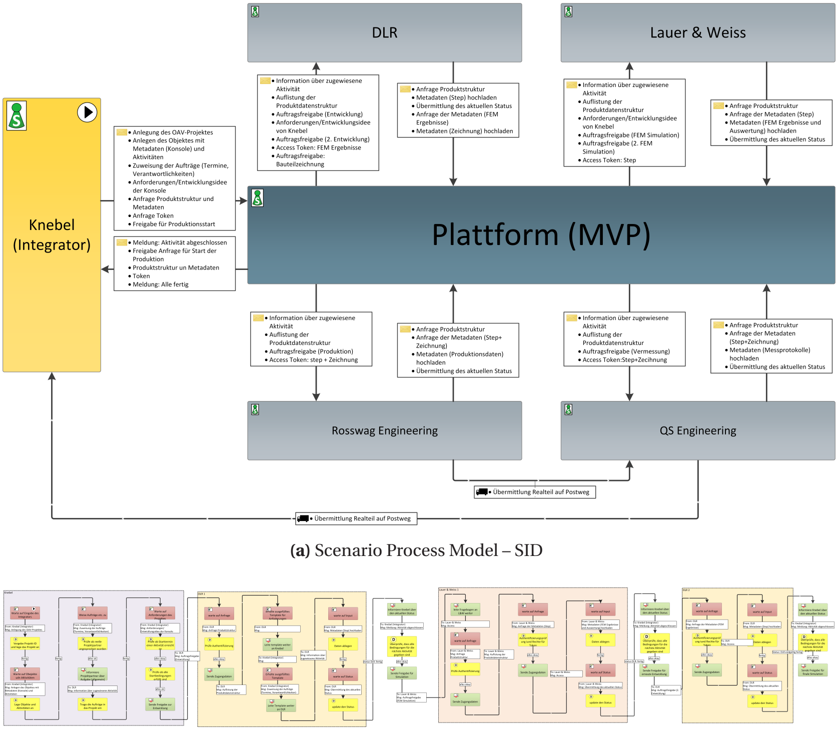
- **Project Instance** builds upon original subjects from the Synthesis Model. Each subject is displayed as a colored box, with additional boxes representing topics from user story elicitation. Smaller boxes show individual user stories. Headed frames group user stories by main themes (e. g., communication), with arrows indicating feature dependencies. The functionalities within the project instance can occur concurrently and are not in a fixed sequence.

These elaborations guided the implementation activities during the initial phases of the IntWertL project. It is evident that this type of graphical representation lacks standardization. The use of these graphical representations suggests a need for structural information.

5.1.9.2 Scenario Process Model

Building on the Synthesis Model V1, a dedicated scenario process model was developed to facilitate the implementation of the MVP. The resulting PASS model is illustrated in Figure 5.8.

The aim of this model is to provide a comprehensive walkthrough of all interactions necessary to fulfill an order, from project initiation on the platform to the shipping of the finished part. This includes both engineering and production tasks, encompassing iterations of the development object as it is optimized through Finite Element Analysis. However, it follows a strictly linear "happy path" and does not account for the possibility of project abandonment or similar deviations. All six subjects participating in the process are fully specified with complete SBDs. The primary focus is on the platform's MVP, which assists in aligning functionality with stakeholder requirements.


DLR
Lauer & Weiss
Knebel (Integrator)
Plattform (MVP)
Rosswag Engineering
QS Engineering
• Anlegung des OAV-Projektes
• Anlegen des Objektes mit Metadaten (Konsole) und Aktivitäten
• Zuweisung der Aufträge (Termine, Verantwortlichkeiten)
• Anforderungen/Entwicklungsidee der Konsole
• Anfrage Produktstruktur und Metadaten
• Anfrage Token
• Freigabe für Produktionsstart
• Meldung: Aktivität abgeschlossen
• Freigabe Anfrage für Start der Produktion
• Produktstruktur un Metadaten
• Token
• Meldung: Alle fertig
• Information über zugewiesene Aktivität
• Auflistung der Produktdatenstruktur
• Auftragsfreigabe (Entwicklung)
• Anforderungen/Entwicklungsidee von Knebel
• Auftragsfreigabe (2. Entwicklung)
• Access Token: FEM Ergebnisse
• Auftragsfreigabe: Bauteilzeichnung
• Anfrage Produktstruktur
• Metadaten (Step) hochladen
• Übermittlung des aktuellen Status
• Anfrage der Metadaten (FEM Ergebnisse)
• Metadaten (Zeichnung) hochladen
• Information über zugewiesene Aktivität
• Auflistung der Produktdatenstruktur
• Anforderungen/Entwicklungsidee von Knebel
• Auftragsfreigabe (FEM Simulation)
• Auftragsfreigabe (2. FEM Simulation)
• Access Token: Step
• Anfrage Produktstruktur
• Anfrage der Metadaten (Step)
• Metadaten (FEM Ergebnisse und Auswertung) hochladen
• Übermittlung des aktuellen Status
• Information über zugewiesene Aktivität
• Auflistung der Produktdatenstruktur
• Auftragsfreigabe (Produktion)
• Access Token: step + Zeichnung
• Anfrage Produktstruktur
• Anfrage der Metadaten (Step+ Zeichnung)
• Metadaten (Produktionsdaten) hochladen
• Übermittlung des aktuellen Status
• Information über zugewiesene Aktivität
• Auflistung der Produktdatenstruktur
• Auftragsfreigabe (Vermessung)
• Access Token:Step+Zecihnung
• Anfrage Produktstruktur
• Anfrage der Metadaten (Step+Zeichnung)
• Metadaten (Messprotokolle) hochladen
• Übermittlung des aktuellen Status
• Übermittlung Realteil auf Postweg
• Übermittlung Realteil auf Postweg

(a) Scenario Process Model – SID

(b) Scenario Process Model – Extract of Platform SBD

Figure 5.8: Illustration of PASS scenario process model, derived directly from Synthesis Model V1

The Scenario Process Model serves as a concrete reference for structuring the MVP roadmap and prioritizing functionalities.

Alongside the execution of the MVP, the Scenario Process Model played a crucial role in validating the MVP by assisting in the design of the validation plan for its components. This plan was organized into distinct sections for each component. However, it was observed that components unrelated to the Scenario Process Model could not be adequately validated.

5.1.9.3 Summary Ex Post Evaluation

Overall, the ex post evaluation demonstrates that Synthesis Model V1 provided a robust framework for coordinating the implementation of SOLUTIONS, particularly the MVP. The graphical representation of user stories offered structure and clarified the prioritization of development efforts from multiple stakeholder perspectives. The Synthesis Process Model facilitated the design of functionalities in the development of the specific MVP for the Project Instance.

5.1.10 Conclusion of Iteration 1

Iteration 1 of the Synthesis Model commenced with the establishment of a common vision and the collection of high-level requirements to create a shared frame of reference for the Digital Ecosystem. Central to this iteration was the development of a *process-centric* model. In this context, the General Solution is an SID adopted from PASS, ensuring interpretability and the capability to represent complex interactions among active entities.

Key Outcomes of Iteration 1:

- **Process-Centric Foundation:** An SID-based modeling approach provided a clearly structured basis, resulting in two interconnected diagrams that visually outlined core ecosystem actors and their interactions.
- **Broad Applicability:** The resulting artifact quickly became the most widely used common reference across all work packages, guiding both *User Story Elaboration* and the creation of a *Scenario Process Model*.
- **Validated Through Evaluations:** General and specific ex ante evaluations confirmed the approach's suitability, while an ex post assessment revealed substantial adoption and effectiveness in moving the project toward its first major milestone: the platform MVP.

- **User-Oriented Design Measures:** Targeted implementation strategies addressed potential design risks by emphasizing usability and stakeholder alignment, promoting high acceptance among diverse participants.

Overall, this first iteration produced a model that not only fostered a shared vision but also served as a practical reference framework, supporting various activities including requirements gathering, collaborative solution ideation, and MVP implementation. Overall, the process-centric Synthesis Model V1 establishes the baseline framework for designing the IntWertL Ecosystem.

5.2 Iteration 2: Structure-centric approach

The initial iteration of the Synthesis Model concentrated on identifying fundamental processes within the IntWertL Ecosystem, thereby establishing a foundational perspective on stakeholder interactions and service definitions. However, as research progressed, it became evident that a strictly process-centric approach inadequately addressed the structural dimensions required for a comprehensive ecosystem model. In particular, the absence of a unified structural representation hindered consensus on critical decisions, such as functional architecture, business model, sales strategy, and unique selling proposition (USP).

To address these shortcomings, the second iteration of the Synthesis Model was initiated at the end of October 2023 and presented at the general assembly of IntWertL in late February 2024. This iteration aimed to enhance the existing process model by introducing a structure-centric perspective. A key driver for this iteration was the necessity to incorporate considerations related to alignment with Catena-X and the homologation process. Furthermore, challenges related to the operating model, integration strategies, and distribution mechanisms underscored the need for a coherent structural foundation.

The subsequent sections will go through the nine SDSM steps taken to design a structure-centric Synthesis Model V2.

5.2.1 Analyze Specific Problem Domain

For the second iteration, the analysis of the problem domain did not focus on the overall vision for IntWertL. Instead, newly identified boundary conditions were further analyzed. The three primary topics of focus were Catena-X, the homologation process, and considerations of functional architecture.

A substantial amount of information about Catena-X is available online, particularly through Tractus-X. Furthermore, several IntWertL participants are members of the Catena-X Association or have contributed to the Catena-X research project, enhancing the analysis of this domain.

A comprehensive understanding of the homologation process was achieved through interviews with experienced IntWertL partners. Additionally, discussions were held with software providers specializing in homologation and certification. [4][5] The primary insight was that rapid homologation is crucial for the IntWertL vision, as it facilitates the accelerated engineering and production of individualized vehicles that meet customer needs. However, homologation is a separate concern and does not constitute a part of the activities in IntWertL. It should only be considered in the modeling phase without further elaboration of details at this time.

[4] Certivity - A Start-Up from Munich that specializes in automotive RegTech. See www.certivity.io

[5] TypeMaster - The software solution from T-Systems that is a leading Enterprise-Content-Management-System for homologation. See www.t-systems.com/de/en/content-and-collaboration/loesungen/enterprise-content-management/typemaster

Following the presentation and initial testing phase of the MVP, a workshop was organized for participants of IntWertL to identify and structure the functions and modules relevant to the Digital Ecosystem. This workshop was divided into two main parts, each connected to one of the two building blocks of the IntWertL SOLUTIONS:

1. **Function:** Functions are specifically designed to execute certain tasks or operations within a software application or system. They vary in complexity, ranging from simple, repeatable tasks to intricate processes. The primary aim of a function is to accomplish a distinct task effectively and consistently, thereby ensuring the smooth operation of the software.

2. **Module:** A module represents an extension or an independent software component that contributes additional features or functionalities to the platform. Modules are often leveraged to tailor the platform according to the user's requirements and to enhance its capabilities.

Initially, a wide range of functions was gathered through eight inquiries, as illustrated in Figure 5.9.

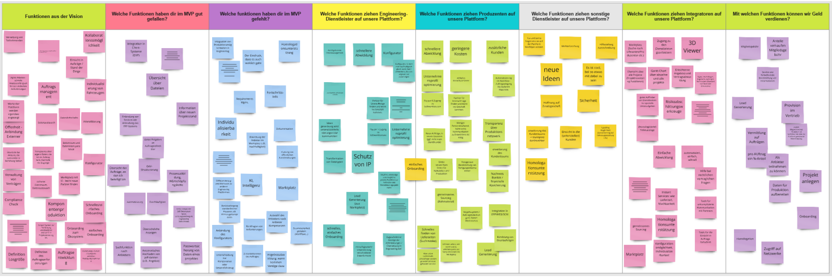

Funktionen aus der Vision
Welche Funktionen haben dir im MVP gut gefallen?
Welche funktionen haben dir im MVP gefehlt?
Welche Funktionen ziehen Engineering-Dienstleister auf unsere Plattform?
Welche Funktionen ziehen Produzenten auf unsere Plattform?
Welche Funktionen ziehen sonstige Dienstleister auf unsere Plattform?
Welche Funktionen ziehen Integratoren auf unsere Plattform?
Mit welchen Funktionen können wir Geld verdienen?

Figure 5.9: I2: Feature Development Workshop - Function Collection

The inquiries utilized in the workshop are as follows:

- Which functions are part of the vision of IntWertL?

- Which functions in the MVP did you find appealing?
- Which functions did you feel were lacking in the MVP?
- Which functions are appealing to engineering service providers on our platform?
- Which functions are appealing to production service providers on our platform?
- Which functions are appealing to other service providers on our platform?
- Which functions are appealing to integrators on our platform?
- Which functions can be monetized?

Subsequently, these functions were categorized into 14 distinct modules. The results of the functional segmentation are displayed in Figure 5.10.

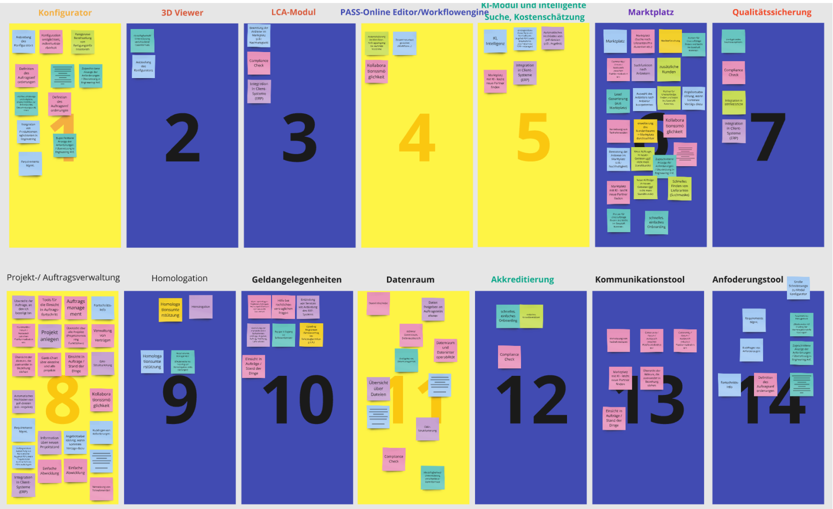


Figure 5.10: I2: Feature Development Workshop - Function Cluster

In the second segment of the workshop, five of these modules were developed in greater detail. First, participants in module development and stakeholders were identified. Second, the functions from phase one were grouped and organized into a schematic module architecture. This was checked for completeness, and missing functions were added. All functions were then sorted by the current progress of development. Third, the functions of each module were ranked in terms of importance and release timeline. Finally, a timeline and important milestones for each module were set.

5.2.2 Define Specific Requirements

In the process of enhancing platform functionality and developing business and operating models, several questions have emerged. These questions pertain to four areas:

- **Functional Architecture**: Refers to the design and structure of the CORE SERVICES, outlining how different components and services interact to fulfill the platform's functions.
- **Business Model**: Describes how the platform generates revenue, who its target users are, and what value it provides.
- **Sales Strategy**: Involves the strategies and processes employed to sell the platform's services or products to its customers.
- **Unique Selling Proposition (USP)**: Identifies what makes the platform stand out from competitors, highlighting its unique features and benefits.

For all four fields, critical questions have emerged that influence one another. Therefore, a unified perspective, specifically a new Synthesis Model, is essential to address these questions, utilizing a common frame of reference. Consequently, specific topics that were previously only vaguely defined or ambiguously understood have been identified for inclusion in the new Synthesis Model to enhance clarity. These topics are presented in Table 5.7.

Table 5.7: I2: Ambigious topics regarding the IntWertL Ecosystem and their descriptions

Topic	Description
CORE SERVICES	Investigation of essential functions beyond marketplace, project management, and homologation requirements.
EcoVity Module Composition	Identification of core SOLUTIONS within the IntWertL Ecosystem and potential for additional integrations.
Clarity of Target Architecture	Analysis of the target platform architecture and its contributions to the overall IntWertL ecosystem.
Modularity and Platform Design	Evaluation of modularization types: a unified platform versus standalone applications.
Module Interaction Dynamics	Examination of inter-module interactions and their implications for the overall system performance.
Catena-X	Alignment of IntWertL with Catena-X. Explanation of Catena-X's objectives and benefits (dismantling of existing barriers).

5.2.3 General Problem and Requirements

The primary challenge remains establishing a shared reference point for all stakeholders in the Digital Ecosystem. In its second iteration, the Synthesis Model shifts its focus away from interaction. While active entities, particularly CORE PARTICIPANTS, continue to be important, there is now an increased emphasis on functional architecture and the delineation of functions, thereby clarifying the role of each function. The Synthesis Model V2 is intended to shape the business and operating model, guide the design of sales strategies, and define the USPs of CORE USERS. This model is designed for human discussions and not for machine interpretation.

The general requirements for the Synthesis Model (Iteration 2) are defined as follows:

I2.1 **Display Active Entities:** Identify and represent active entities involved in the Digital Ecosystem, especially CORE USERS.

I2.2 **Provide Structure:** Present a functional architecture that distinctly separates the CORE SERVICES into individual components.

I2.3 **Allow Functional Breakdown:** Add an abstraction layer to facilitate a more detailed decomposition of functions at a lower level.

I2.4 **Ensure Interpretability:** Make the model self-explanatory, ensuring stakeholders can easily understand it. The model must be clear and interpretable without requiring technical expertise, enabling its use in discussions among diverse stakeholders.

5.2.4 General Solution

The general solution in the second iteration employs a specialized version of the SysML BDD as the Synthesis Model. The BDD facilitates the modeling of socio-technical systems by clearly delineating hierarchical relationships. The Synthesis Model V2 consists of two components: the actors and the services. Both components are represented as blocks. The structure of the BDD is illustrated in Figure 5.11.

In contrast to the structure of conventional BDDs, not all blocks in Synthesis Model V2 are created equal. There are *actor blocks* and *service blocks.* According to the paradigm of subject-orientation, all active entities must be

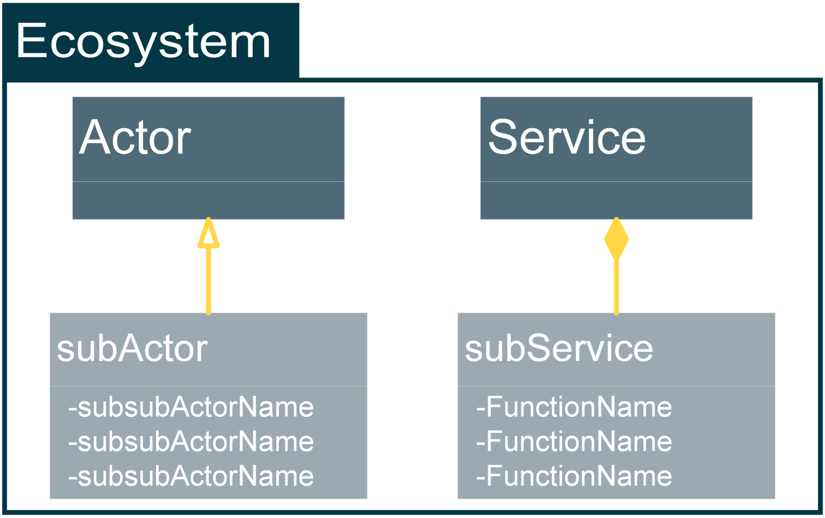


Figure 5.11: I2: General Solution

modeled as the former type, while the latter block type is designated for describing passive structures (objects).[6] All *service blocks* must be executed by an active entity to be utilized in any process.

The first section of the Synthesis Model identifies all relevant actors. These primarily include CORE USERS but may also encompass ORCHESTRATORS or other CORE PARTICIPANTS, specifically CONTRIBUTORS if it aids in clarification. All relationships between actors are hierarchical and represent generalizations. In contrast to SysML specifications, multiple inheritance is prohibited to maintain a clearly defined structure within the actors section.

[6] To conform with the logic of BDDs this is equivalent to having an *abstract actor block* and an *abstract service block* from which all *actor blocks* and *service blocks* inherit the specifications. This is very similar to the definition employed for the general block type in SysML-here a block is just a specialization of a UML class.

The second section encompasses services. These can be CORE SERVICES along with any other supporting or value-adding services within the ecosystem. According to the definition, the CORE SERVICES represent the most relevant components. Only hierarchical relationships between services are modeled. To ensure a clear structure, the strongest relationship, composition, is employed for all these relationships. Service blocks at a lower composition level are also referred to as functions. This approach aligns with the underlying assumption of strict orchestration of services: each function is unique and belongs to only one service.

The overall expressiveness of the BDD is significantly constrained to simplify the Synthesis Model. As a result, no attributes (values), operations (behaviors) or properties are included. This information can be conveyed through a textual glossary. The available relationships are limited exclusively to the strongest hierarchical relationships: generalization for actors and composition for services. To further emphasize the focus on structure, no constraints or interfaces are described either.

5.2.5 Ex Ante Evaluation (General)

The Ex Ante Evaluation of the general solution is conducted as an informed argument by modeling experts. The results are presented in Table 5.8.

5.2.6 Design Specific Solution

The Synthesis Model V2 offers a more streamlined design solution compared to the synthesis process model from the initial iteration. At the core of this model are many of the same elements as in the Platform SID, with a notable modification being the replacement of the Project Instance with IntWertL CORE SERVICES, which is derived from the implemented Platform MVP. Most subjects have been integrated as blocks into the new Synthesis Model, as depicted in Figure 5.12. Note that some of the formerly active subjects have

Table 5.8: I2: Matching of requirements to general solution

Requirement	Solution
I2.1 Display Active Entities	• Clear distinction between *actor blocks* and *service blocks*. • CORE USERS have to be depicted. • Other CORE PARTICIPANTS or ORCHESTRATORS that take on an active role can be included in the model.
I2.2 Provide Structure	• BDD is a diagram of structure. • Restricting multiple inheritance on the generalizations and choosing composition rather than aggregation provides clear structure. • Excluding all weak relationships.
I2.3 Allow Functional Breakdown	• Generalization allows a more specialized breakdown of roles. • Functional Breakdown of the CORE SERVICES is provided as composition.
I2.4 Ensure Interpretability	• Just one modeling element (i. e. blocks) and just one relationship type per component type. • Association between blocks still comprehensible without knowledge of different relationship types (i. e. composition and generalization).

now been transformed to passive objects that can not execute any behavior. Although the diagrams presented here were also created using MS Visio, they are simpler to produce than SIDs and can be easily modeled in other tools such as PowerPoint.

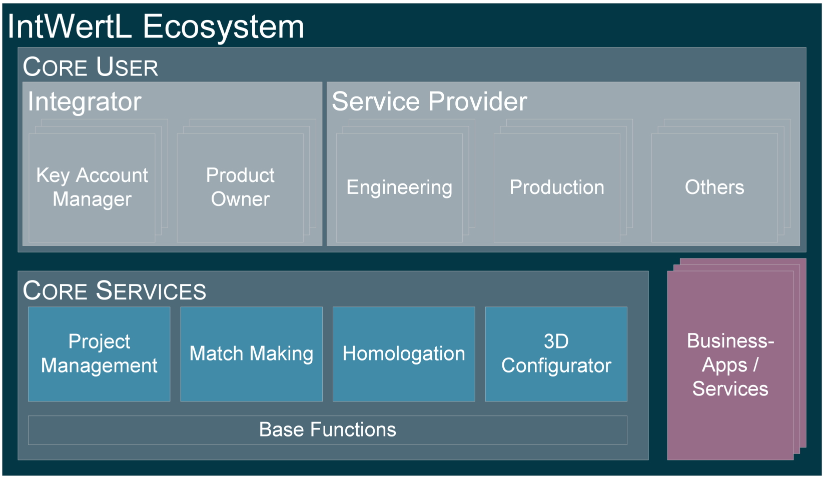


Figure 5.12: Synthesis Model V2 - Ecosystem

In this first diagram, the high-level structure of the IntWertL ecosystem is depicted. This diagram illustrates the key actors within the system and their hierarchical classification as well as the top level breakdown of services.

The actor section includes the Core Users. Similar to Synthesis Model V1 the integrator remains to be integral to the ecosystem. Integrators are platform users who have initiated a project and are responsible for its execution. An integrator represents the interests of the end customer in a project. The integrator takes on the tasks of a project manager. They are responsible for all aspects of project execution, including coordination, communication, and monitoring. They oversee all product aspects, including engineering, production, and homologation. The two main roles of an integrator in any given project are *Key Account Manager* and *Product Owner*. Similar to the

Project Instance from the first iteration, the *Service Providers* are still categorized as *Engineering*, *Production*, and *Others*.

The second section of the diagram enumerates the services within the ecosystem. Here the focus is on the CORE SERVICES, with the *Business-Apps / Services* block accounting for extended functionality. As outlined in the general solution, services are strictly orchestrated through composition. Figure 5.13 details the hierarchical breakdown pf CORE SERVICES. This is presented in a separate diagram to avoid overloading the first one.

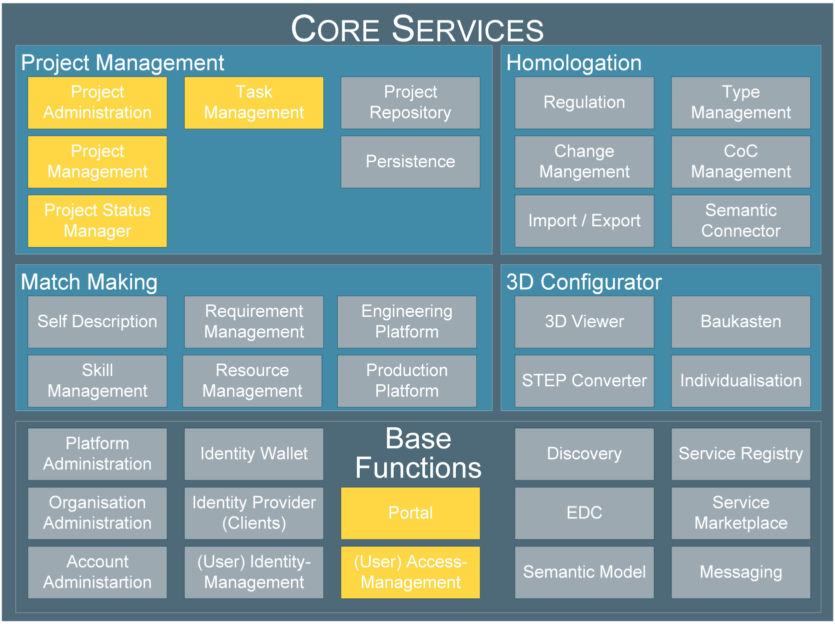


Figure 5.13: Synthesis Model V2 - Core Services Breakdown. Yellow - Function is part of the IntWertL MVP.

The CORE SERVICES breakdown illustrates the functions that compose the higher-level service blocks. The functions already demonstrated in the MVP are highlighted in yellow. The CORE SERVICES can be decomposed as follows:

- **Project Management:** Encompasses all functions related to project coordination and administration. Most of the business functions demonstrated in the MVP are part of this service, specifically *Project Administration, Project Management, Task Management, Project Status Manager.*
- **Match Making:** Facilitates alignment between system requirements and available resources. Key functions include the *Engineering- and Production Platforms* as well as the *Self Description.*
- **Homologation:** Addresses regulatory compliance and certification processes that are not further analyzed or implemented in IntWertL but important to the vision of the ecosystem.
- **3D Configurator:** [7] Provides visualization and customization tools for system components. Two functions, namely *STEP Converter* and *3D Viewer* had already been implemented by IntWertL participants.
- **Base Functions:** Fundamental system operations, including *Portal* and *User Access Management.* Most of the Catena-X enablement services are included in this ervice as they provide the basis for collaboration in a shared data space.

5.2.7 Ex Ante Evaluation (Specific)

The Synthesis Model V2 provides a common frame of reference to eliminate ambiguities related to the topics outlined in section 5.2.2. The representation of these topics in the model is displayed in Table 5.9.

Furthermore, Figure 5.14 maps all the primary areas of concern that have been identified to this iteration of the Synthesis Model.

[7] The 3D Configurator was not part of the initial drafts for the Synthesis Model V2. However, it was added in after discussions in the specific Ex Ante Evaluation.

Table 5.9: I2: Topics and their representation in the Synthesis Model V2

Topic	Representation
Core Services	Core Services are represented as the four blocks *Base Functions, Project Management, Match Making, Homologation*, and *3D Configurator*.
IntWertL Module Composition	Specified through the blocks Core Services and *Business-Apps / Services*.
Clarity of Target Architecture	Provided via the blocks in the Core Services breakdown. Distinction between *Base Functions* and the other modules. Alignment of functions with MVP.
Modularity and Platform Design	Not directly depicted in Synthesis Model (as it is tbd at time of implementation). Derivations of the Synthesis Model V2 illustrated in Figure 5.18 were created.
Module Interaction Dynamics	Not part of the model (modularity still tbd).
Catena-X	Alignment of Core Services breakdown with Catena-X language. Derivations of the Synthesis Model V2 illustrated in Figures 5.20, 5.21.

Subsequent results are categorized into strengths (S), weaknesses (W), opportunities (O), and threats (T). Strengths and weaknesses describe static qualities, i. e. inherent characteristics of the model structure. In contrast, opportunities and threats describe dynamic qualities, i. e. results arising from the application of the artifact in its operational context.

5.2.7.1 Strengths

- **Simplicity:** The model is comprised of a single component: Blocks. By dividing the Synthesis Model into two separate diagrams, it becomes easier to understand both the ecosystem perspective and the functional decomposition. Although the relationships between actor blocks (generalization) and service blocks (composition), differ in nature, it is not essential for most users of this model to concentrate on

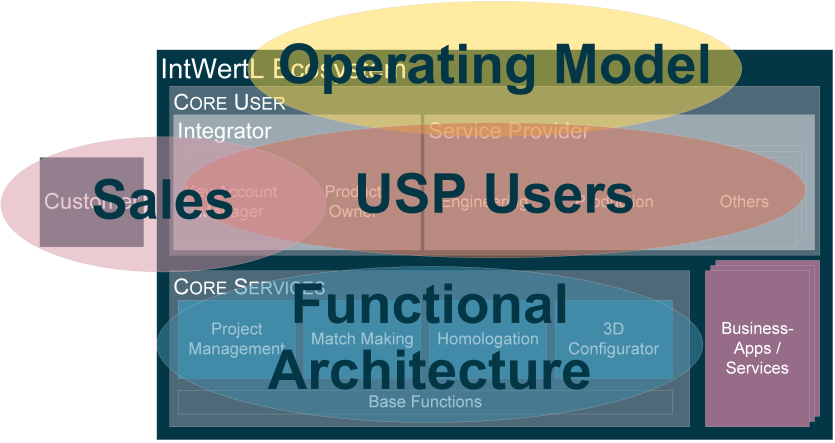


Figure 5.14: Synthesis Model V2 - Positioning of main areas of concern

these differences. This is particularly true since both types of relationships are purely hierarchical, given the limitation on multiple inheritance.

- **Extensibility:** It is easily possible to add new Core Participants or Core Services if the design of the Digital Ecosystem changes. It is also easily possible to add new functions or reallocate them from one service to another.

- **Alignment with software architecture:** The services are easily transformed into implementable modules. The functional decomposition can be understood as micro-service architecture. As a BDD is nothing but a specialized UML Class Diagram, these diagram designs are familiar to system architects and can yield as starting point of their technical system architecture design (where Blocks can become classes and attributes as well as methods are included).

5.2.7.2 Weaknesses

- **No processual information:** The Synthesis Model V2 conveys only structural information and does not encompass the relationships between actors and services. Additionally, it does not include the interfaces utilized among services. Consequently, actual customer journeys or use case scenarios are omitted.
- **Ambiguity of labels:** The model relies on precise labeling. Actors, services, and functions are described solely through brief labels within the model. Inadequately chosen labels can lead to misunderstandings, particularly in contexts involving diverse stakeholders.
- **No detailed descriptions:** Detailed descriptions of the model elements are maintained separately from the model itself. The additional effort required to reference these descriptions may hinder the correct interpretation of the model and lead to erroneous conclusions. Furthermore, no single source of truth is maintained; block labels or connected glossaries can be updated independently, potentially resulting in inconsistent terminology.

5.2.7.3 Opportunities

- **Alignment with MVP:** At the time of model generation, the MVP had just been presented and released for testing and validation. A model that clearly delineates the functionalities present in this MVP and outlines potential extensions fosters interest and aids in comprehending the various levels of abstraction.
- **Decentralized Architecture:** IntWertL emphasizes a fully decentralized architecture. This architecture comprises federated data and federated services. The easily extensible structure of the second iteration effectively supports these concepts of decentralization.

- **Catena-X compatibility:** The structure of Synthesis Model V2 is highly compatible with the descriptions of the Catena-X Data Space. The representation format even resembles the Catena-X Service Map.[8]
- **Reference point for diverse concerns:** As illustrated in Figure 5.14 many different concerns can be allocated within the model. Given that these four domains have been identified as the primary interaction points for discussions extending beyond the work packages, a unifying model to guide these discussions is essential.

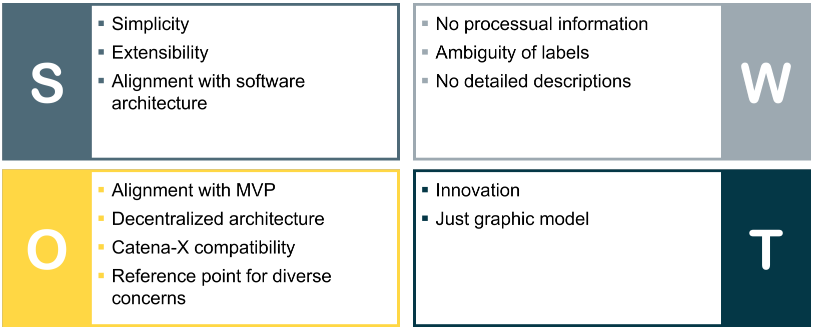
5.2.7.4 Threats

- **Innovation:** Design of a Digital Ecosystem is a real innovation. Therefore, a general understanding of the fundamentals of EDE can not be anticipated among stakeholders. Neither a common vision for the ecosystem nor its solutions are well established. Nevertheless, the aim is to disruptively change small-scale vehicle production. This change, as with every major disruption, will produce reservations of some kind.
- **Just graphic model:** The Synthesis Model V2 is realized just as a graphical model. because of its basic structure and shapes it can easily be mistaken for only a diagram or "just another PowerPoint-engineering feat". The following of BDD structure might be lost in communication and therefore the understanding of the actual hierarchical relationships.

[8] The Catena-X Service Map is part of the Catena-X Operating Model, that is maintained by the Catena-X association and subject to regular updates. Note that the understanding of Core Services in Catena-X differs from the definition of CORE SERVICES used in this thesis.

5.2.7.5 Summary Ex Ante Evaluation (Specific)

The analysis of SWOTs is summarized in Figure 5.15 and the specific solution is true to the general solution. All topics identified in section 5.2.2 are either represented in, or can be discussed using, the Synthesis Model. Furthermore, the four primary areas of concern have been mapped to this model. Similar to the first iteration, the principal weaknesses stem from design choices that prioritize simplicity. Due to significant uncertainty regarding many guiding decisions for the development of the IntWertL ecosystem, which can all be discussed using this artifact, there is considerable potential for opportunities compared to threats. Nevertheless, these SWOTs still need to be translated to a suitable implementation strategy.



S
- Simplicity
- Extensibility
- Alignment with software architecture

W
- No processual information
- Ambiguity of labels
- No detailed descriptions

O
- Alignment with MVP
- Decentralized architecture
- Catena-X compatibility
- Reference point for diverse concerns

T
- Innovation
- Just graphic model

Figure 5.15: Synthesis Model V2 - SWOT

5.2.8 Implement Specific Solution

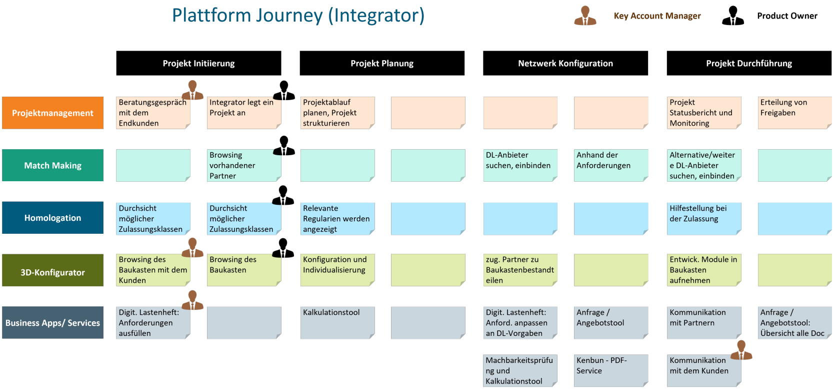
The primary risk for the Synthesis Model remains user-oriented. Figure 5.16 illustrates the translation of the SWOT analysis into implementation strategies for the second iteration.

	Strengths	Weaknesses
Opportunities	SO - Adapt and derivate the model to comply with diverse concerns. - Double down on creating a decentralized software architecture for the Digital Ecosystem.	WO - Use external descriptions to provide more detail (such as processual information, glossary, etc.) and use the Synthesis Model as connecting reference point.
Threats	ST - Adopters: Simplicity reigns supreme over modeling means. - Developers: Focus on similarities to UML Class diagrams.	WT - Combine with storytelling to add detail and promote willingness for innovation.

Figure 5.16: Synthesis Model V2 - Implementation strategies

Similarly to the first iteration, the Synthesis Model V2 is presented in multiple forms: First, it is available digitally in Visio, PowerPoint, and PNG formats; second, as a printed poster; and lastly, as part of a presentation during the general assembly of IntWertL in February 2024. This presentation addresses all stakeholders. Therefore, it focuses on the modeled information and the simplicity of presentation. The modeling choices and design of the general solution are not discussed. Immediately following the presentation, the model elements are used to describe a platform journey for Integrators, as illustrated in Figure 5.17. This approach adds an engaging element while supplementing process information, thereby fostering acceptance. Additionally, a revised glossary is introduced that aligns with the terminology of the Synthesis Model V2.

In subsequent presentations, the need for a decentralized architecture is emphasized to effectively align with Catena-X. Later that day, workshops are conducted with groups focusing on the four primary areas of concern (functional architecture, business model, sales, USPs). During these sessions, model derivations are utilized to address the areas of concern. In the functional architecture group, primarily composed of developers, the similarity to UML Class Diagrams is noted.



Plattform Journey (Integrator)

Key Account Manager

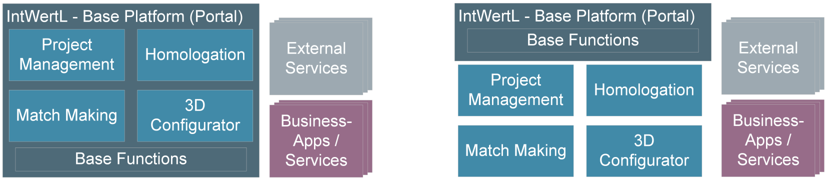
Product Owner

	Projekt Initiierung		Projekt Planung		Netzwerk Konfiguration		Projekt Durchführung	
Projektmanagement	Beratungsgespräch mit dem Endkunden	Integrator legt ein Projekt an	Projektablauf planen, Projekt strukturieren				Projekt Statusbericht und Monitoring	Erteilung von Freigaben
Match Making		Browsing vorhandener Partner			DL-Anbieter suchen, einbinden	Anhand der Anforderungen	Alternative/weitere DL-Anbieter suchen, einbinden	
Homologation	Durchsicht möglicher Zulassungsklassen	Durchsicht möglicher Zulassungsklassen	Relevante Regularien werden angezeigt				Hilfestellung bei der Zulassung	
3D-Konfigurator	Browsing des Baukasten mit dem Kunden	Browsing des Baukasten	Konfiguration und Individualisierung		zug. Partner zu Baukastenbestandteilen		Entwick. Module in Baukasten aufnehmen	
Business Apps/ Services	Digit. Lastenheft: Anforderungen ausfüllen		Kalkulationstool		Digit. Lastenheft: Anford. anpassen an DL-Vorgaben	Anfrage / Angebotstool	Kommunikation mit Partnern	Anfrage / Angebotstool: Übersicht alle Doc
					Machbarkeitsprüfung und Kalkulationstool	Kenbun - PDF-Service	Kommunikation mit dem Kunden	

Figure 5.17: I2: Platform Journey

5.2.9 Ex Post Evaluation

Similar to the first iteration, the Ex Post Evaluation of the second iteration was conducted as an observational field study. Beginning with the workshops during the general assembly—where Synthesis Model V2 was introduced—the model and its derivations were utilized extensively throughout the IntWertL project. For example, the derivations presented in Figure 5.18 were integral to the discussions at these workshops. Until the introduction of Synthesis Model V3, Synthesis Model V2 remained in use, and several derivations of V2 were created for stakeholder discussions.



(a) Centralized Architecture

(b) Decentralized Architecture

Figure 5.18: Derivation of Synthesis Model V2 to discuss (de-)centralization of modules

At the general assembly, four task forces—one for each major area of concern (see Figure 5.14)—were established. The next milestone, following the introduction of Synthesis Model V2, focused on presenting the business and operating models, prompting the respective task force to address an urgent need for action. Additionally, IntWertL partners emphasized the importance of aligning with Catena-X. They utilized the structural overview and functional breakdown from Synthesis Model V2 to guide these discussions. Consequently, the following observations from the Ex Post Evaluation are organized into three sections:

1. **Operating & Business Model:** systematic exploration of service responsibilities and governance, as well as supporting legal and revenue considerations in a broader business context.
2. **Other task forces:** focusing on *Functional Architecture*, *Sales* and CORE USER *USPs*.
3. **Catena-X:** defining how to pursue Catena-X integration and the extent of technical alignment.

5.2.9.1 Operating & Business Model

The operating and business models are outlined in work packages 3100 and 3200, as well as by the dedicated *Operating Model* task force of IntWertL. The operating model primarily focuses on clarifying responsibilities for the management of each service and defining the governance structure, while the business model addresses legal frameworks and revenue considerations. All related models, presented at the subsequent milestone, build upon the services and functions described in Synthesis Model V2, serving as a reference point for stakeholder discussions regarding the delivery and integration of these services within the broader IntWertL ecosystem. Figure 5.19 illustrates one of several derivations of Synthesis Model V2 that outlines potential business structures.

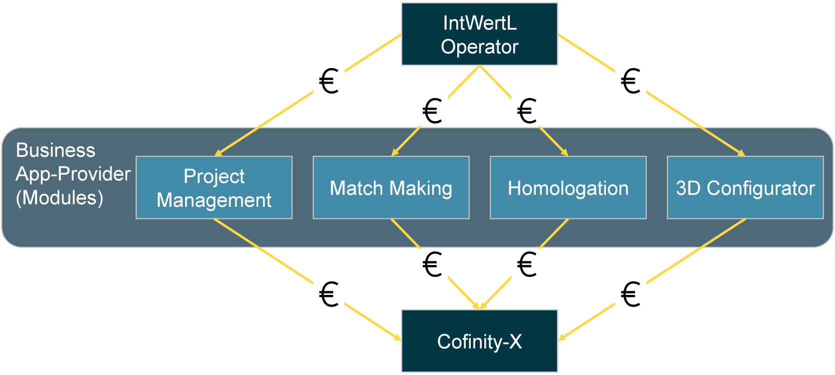


Figure 5.19: Derivation of Synthesis Model V2 to articulate the business model

5.2.9.2 Other task forces

Additionally, the *Functional Architecture*, *Sales* and CORE USER *USPs* task forces used the Synthesis Model V2 as a reference framework. This framework guides their activities and, more importantly, facilitates the regular alignment of all four task forces to coordinate major concerns for the entire IntWertL ecosystem. One of the core architectural questions is whether specific platform elements should be centralized or decentralized (see Figure 5.18). Another significant aspect of the efforts involves further detailing and clarification of existing modules, particularly the 3D configurator. Figure 5.20 presents another derivative of the Synthesis Model V2. It employs color-coding to analyze how Catena-X use cases might be applied in the depicted functions.

However, since Synthesis Model V2 prioritizes structural and functional aspects over process descriptions, some teams independently developed their own workflows and process diagrams. In certain instances, references to the glossary or textual definitions provided with V2 were overlooked, leading to inconsistencies. This issue was particularly evident when multiple task forces, or a task force and a work package, addressed the same topic without

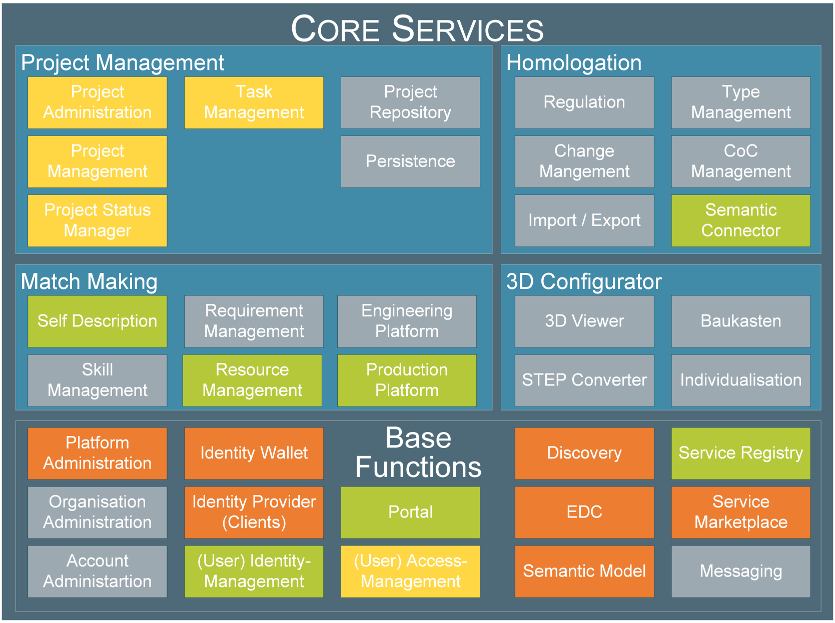


Figure 5.20: Derivation of Synthesis Model V2 to identify Catena-X use cases and functions that could be used as part of the IntWertL CORE SERVICES. Yellow - part of the IntWertL MVP; Orange - Same as Catena-X; Green - Compare to Catena-X.

coordination, resulting in inconsistent models and an inability to rely on a common frame of reference.

IntWertL partners increasingly required process information, resulting in the adoption of various process modeling methods such as PASS, BPMN, flowcharts, UML, and informal approaches like process flows modeled in spreadsheets. Software development partners created process models to clarify information exchange and interfaces between their modules, independently developing sequence diagrams to coordinate efforts. However, the modified definitions of their modules presented challenges in maintaining a common reference framework.

5.2.9.3 Catena-X

A key objective in this iteration was to clarify the potential integration of Catena-X into the IntWertL ecosystem. Figure 5.21 illustrates how Synthesis Model V2 was used to visualize and organize potential Catena-X use cases and functionalities, assisting stakeholders in identifying which services should adhere to Catena-X standards.

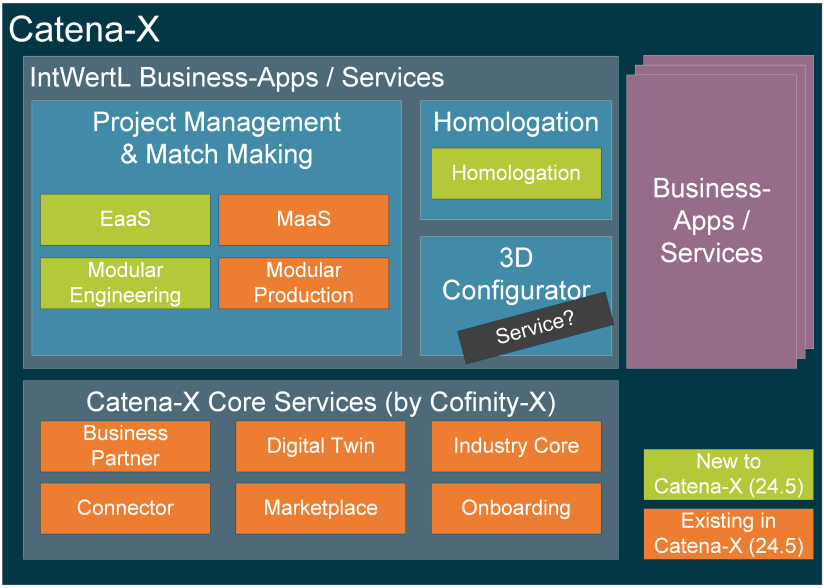


Figure 5.21: Derivation of Synthesis Model V2 to articulate alignment with Catena-X

These diagrams were utilized in consultations with the Catena-X Association, funding agencies, and the Catena-X research consortium. While IntWertL partners opted to pursue compatibility with Catena-X, the degree of technical integration—such as specific interfaces, data formats, or security protocols—remains an open question. Nonetheless, by addressing Catena-X requirements early in the process, the IntWertL partners ensured that certain CORE SERVICES were designed with interoperability in mind.

5.2.9.4 Summary Ex Post Evaluation

Overall, the Ex Post Evaluation of the second iteration demonstrates that Synthesis Model V2 remained integral to various aspects of the IntWertL project. From specifying the operators of each service to managing revenue and governance structures, as well as exploring refinements to existing modules, such as the 3D configurator, V2 served as a robust structural reference. Additionally, alignment with Catena-X has emerged as a key focus, with Synthesis Model V2 acting as a guide for identifying necessary standards and integrations. The increasing demand for process details prompted certain work packages and task forces to develop their own diagrams. While this flexibility allowed for domain-specific explorations, it also introduced the risk of inconsistent documentation when teams failed to coordinate effectively.

5.2.10 Conclusion Iteration 2

The first iteration of the Synthesis Model aimed to capture core processes within the IntWertL Ecosystem, focusing on stakeholder interactions and services. However, it lacked a structural framework, which hindered decision-making regarding functional architecture and commercialization. To address this, a second iteration of the model was developed to introduce a structure-centric perspective, aligning with Catena-X and addressing four core areas of interest: *Operating Model*, *Functional Architecture*, *Sales Strategy*, and CORE USER *USPs*.

The general solution for the second iteration is based on an adaptation of SysML BDDs. The Synthesis Model V2 comprises two primary components: actor blocks and service blocks. The relationships between these components are strictly hierarchical, consisting of generalizations for actors and compositions for services. The formative ex ante evaluation of both the general and specific solutions demonstrated appropriate alignment with the requirements for delivering structural information while remaining self-explanatory.

After its implementation, the Synthesis Model V2 was extensively utilized throughout the IntWertL project, in both regular work packages and newly established *Operating Model*, *Functional Architecture*, *Sales*, and CORE USER *USPs* task forces. The Ex Post Evaluation indicated that the Synthesis Model and its derivatives were employed in all these areas of interest, as well as for proper alignment with Catena-X. However, the evaluations also revealed the need to integrate process information into the common reference framework.

5.3 Iteration 3: Synthesized approach

After the second iteration, further work on the functional architecture of the IntWertL Ecosystem was conducted. It became evident that, although the structure-centric approach from Synthesis Model V2 provided a solid foundation, it remained insufficient. Specifically, there was a need for more detailed process information, relationships among actors and functions, and comprehensive descriptions of all modeling elements. In response, the functional architecture task force addressed these issues. Ultimately, it became clear that the process-centric approach from the first iteration and the structure-centric approach from the second iteration needed to be integrated into a unified model.

The refinement of the functional architecture began in March 2024. The resultant Synthesis Model V3 was presented at the IntWertL General Assembly in February 2025. The general solution in this third iteration closely aligns with the Synthesis Model structure of the EDE methodology presented in section 6.3. As with the previous iterations, this section will guide through the nine-step SDSM design process.

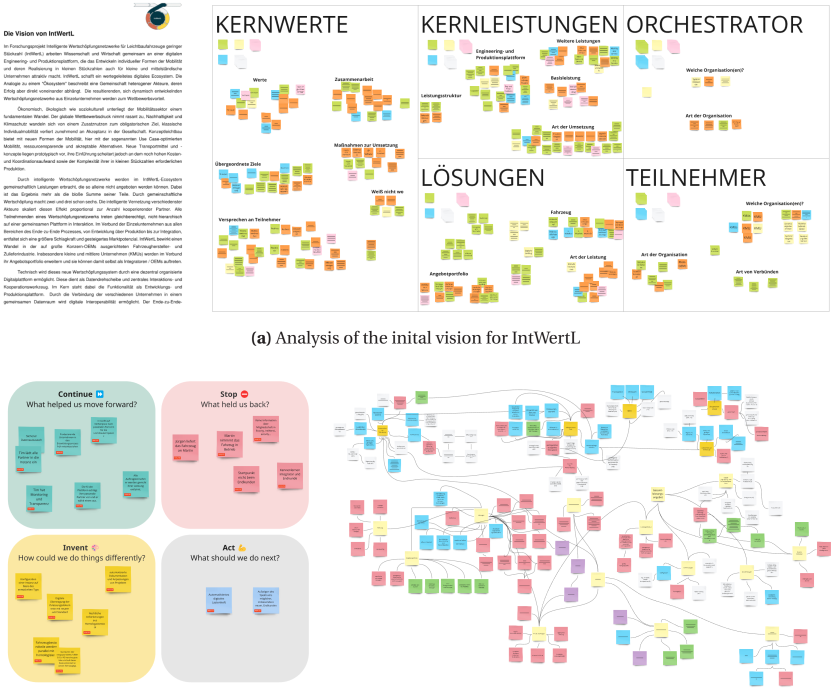
5.3.1 Analyze Specific Problem Domain

The results from the first two iterations form the foundation for the third iteration. Furthermore, the domain is analyzed from two perspectives: establishing a strategic vision through vision workshops and addressing specific topics related to current challenges. This approach distinguishes future strategies from immediate analyses. Both the outcomes of the workshops and the preparatory steps yield significant insights into the problem domain.

Initially, the vision was developed to enhance the understanding of the long-term identity of the IntWertL Ecosystem. Rather than focusing on technical or operational implementation, this vision serves as a guiding and motivational force, directing actions and decisions. It is intended to be inspiring and ambitious, providing a clear sense of meaning and purpose distinct from specific goals due to its broad and aspirational nature. To refine this vision, two workshops were organized. The first was conducted online and concentrated on analyzing existing elements of the vision, such as the vision text and vision stories, as illustrated in Figure 5.22. Following this, the results of the workshop were clustered and mind-mapped to synthesize insights.

The second workshop, conducted in person with all work package leaders, aimed to distill the vision of the IntWertL Ecosystem into three statements addressing the questions: *"Why?"* (Purpose), *"What?"* (Vision), and *"How?"* (Mission). The preparation involved systematically refining previous insights through focused group sessions. This process resulted in the creation of three concise vision statements, developed through team collaboration and mind-mapping activities. These statements served as a foundation for the second vision workshop:

1. **Why are we doing this?** By developing the IntWertL Ecosystem, we enable small and medium-sized companies to successfully secure their future position in the market.



Die Vision von IntWertL

Im Forschungsprojekt Intelligente Wertschöpfungsnetzwerke für Leichtbaufahrzeuge geringer Stückzahl (IntWertL) arbeiten Wissenschaft und Wirtschaft gemeinsam an einer digitalen Engineering- und Produktionsplattform, die das Entwickeln individueller Formen der Mobilität und deren Realisierung in kleinen Stückzahlen auch für kleine und mittelständische Unternehmen attraktiv macht. IntWertL schafft ein wertegeleitetes digitales Ecosystem. Die Analogie zu einem "Ökosystem" beschreibt eine Gemeinschaft heterogener Akteure, deren Erfolg aber direkt voneinander abhängt. Die resultierenden, sich dynamisch entwickelnden Wertschöpfungsnetzwerke aus Einzelunternehmen werden zum Wettbewerbsvorteil.

Ökonomisch, ökologisch wie soziokulturell unterliegt der Mobilitätssektor einem fundamentalen Wandel. Der globale Wettbewerbsdruck nimmt rasant zu, Nachhaltigkeit und Klimaschutz wandeln sich von einem Zusatznutzen zum obligatorischen Ziel, klassische Individualmobilität verliert zunehmend an Akzeptanz in der Gesellschaft. Konzeptleichtbau bietet mit neuen Formen der Mobilität, hier mit der sogenannten Use Case-optimierten Mobilität, ressourcensparende und akzeptable Alternativen. Neue Transportmittel und -konzepte liegen prototypisch vor, ihre Einführung scheitert jedoch an dem noch hohen Kosten- und Koordinationsaufwand sowie der Komplexität ihrer in kleinen Stückzahlen erforderlichen Produktion.

Durch intelligente Wertschöpfungsnetzwerke werden im IntWertL-Ecosystem gemeinschaftlich Leistungen erbracht, die so alleine nicht angeboten werden können. Dabei ist das Ergebnis mehr als die bloße Summe seiner Teile. Durch gemeinschaftliche Wertschöpfung macht zwei und drei schon sechs. Die intelligente Vernetzung verschiedenster Akteure skaliert diesen Effekt proportional zur Anzahl kooperierender Partner. Alle Teilnehmenden eines Wertschöpfungsnetzwerks treten gleichberechtigt, nicht-hierarchisch auf einer gemeinsamen Plattform in Interaktion. Im Verbund der Einzelunternehmen aus allen Bereichen des Ende-zu-Ende Prozesses, von Entwicklung über Produktion bis zur Integration, entfaltet sich eine größere Schlagkraft und gesteigertes Marktpotenzial. IntWertL bewirkt einen Wandel in der auf große Konzern-OEMs ausgerichteten Fahrzeughersteller- und Zulieferindustrie. Insbesondere kleine und mittlere Unternehmen (KMUs) werden im Verbund ihr Angebotsportfolio erweitern und sie können damit selbst als Integratoren / OEMs auftreten.

Technisch wird dieses neue Wertschöpfungssystem durch eine dezentral organisierte Digitalplattform ermöglicht. Diese dient als Datendrehscheibe und zentrales Interaktions- und Kooperationswerkzeug. Im Kern steht dabei die Funktionalität als Entwicklungs- und Produktionsplattform. Durch die Verbindung der verschiedenen Unternehmen in einem gemeinsamen Datenraum wird digitale Interoperabilität ermöglicht. Der Ende-zu-Ende-

KERNWERTE
Werte
Zusammenarbeit
Maßnahmen zur Umsetzung
Übergeordnete Ziele
Weiß nicht wo
Versprechen an Teilnehmer
KERNLEISTUNGEN
Weitere Leistungen
Engineering- und Produktionsplattform
Basisleistung
Leistungsstruktur
Art der Umsetzung
ORCHESTRATOR
Welche Organisation(en)?
Art der Organisation
LÖSUNGEN
Fahrzeug
Art der Leistung
Angebotsportfolio
TEILNEHMER
Welche Organisation(en)?
Art der Organisation
Art von Verbünden
KMU
Continue ⏩
What helped us move forward?
Sicherer Datenaustausch
Tim lädt alle Partner in die Instanz ein
Produzierende Unternehmen in den Entwicklungsprozess früh miteinbeziehen
Er sucht auf Marketplace nach passenden Partnern für die Leichtbaukompetenz
Tim hat Monitoring und Transparenz
Die KI der Plattform schlägt ihm passende Partner vor und er wählt einen aus.
Alle Auftragsteilnehmer werden gerecht ihrer Leistung entlohnt.
Stop ⛔
What held us back?
Jürgen liefert das Fahrzeug an Martin
Martin nimmmt das Fahzeug in Betrieb
Keine Information über Mitgliedschaft in Ecoviy, IntWertL, ecoulty...
Startpunkt nicht beim Endkunden
Kennenlernen Integrator und Endkunde
Invent 🎨
How could we do things differently?
Konfiguration einer Instanz auf Basis des entwickelten Typs
automatisierte Dokumentation und Anpassungen von Projekten
Digitale Übertragung der Zulassungsdokumente mit neuem xml Standard
Rechtliche Anforderungen aus Homologationstool
Fahrzeugbestandteile werden parallel mit homologisiert
Act 💪
What should we do next?
Automatisiertes digitales Lastenheft
Aufzeigen des Spektrums möglicher, insbesondere neuer, Endkunden

(a) Analysis of the inital vision for IntWertL

(b) Analysis of the Vision Story

(c) Mind map of the vision context

Figure 5.22: Analysis of the vision for the IntWertL Ecosystem.

2. **What are we doing?** We are shaping the future of mobility through a value-based, digital ecosystem that supports SMEs to develop innovative mobility solutions as equal players.

3. **How do we do that?** Through intelligent networking and value-based collaboration we achieve radical change in the automotive industry. This change enables resilient and flexible value networks.

The design of the second workshop included an in-person session aimed at analyzing the problem domain from multiple perspectives. Two distinct methods were utilized: the first method focused on subconscious exploration, while the second employed a more analytical approach. This dual-method strategy sought to establish a solid foundation for defining three universal vision statements. The methods employed are:

- **View from the Universe**: This technique is a creative approach for envisioning future scenarios in an immersive, structured way. It integrates visualization, shared drawing, and discussion for a collective depiction of a potential future. The process starts with a short meditation to clear and focus the mind, allowing deeper imagination. Participants then visualize themselves above a future world where the IntWertL Ecosystem or another concept is already realized. In this imagined space, they observe its workings and defining elements, mentally exploring structures and systems without constraints. Following visualization, participants engage in a drawing session, illustrating their visions. This method, focusing on visuals over words, frees ideas from linguistic limits as they collaborate on a shared canvas, adding, refining, and building on each other's input. A collective vision emerges, not as a single coherent image but a rich tapestry of insights. This approach promotes diverse thinking and new ideas. By alternating between abstract and specific details, the technique offers both macro and micro perspectives on future scenarios. Its open, collaborative nature makes it ideal for futures thinking and visionary ideation, offering a robust tool to explore and shape possible futures.
- **Vision Map**: A systematic technique for refining and detailing future visions. Unlike the initial brainstorming phase, which fosters creativity, this method employs a structured framework. At its core is the vision, with key themes branching out to illustrate essential elements needed to achieve the future scenario. These categories act as guides, assisting participants in breaking down the vision into specific elements. This is visualized in an actual vision map as illustrated in Figure 5.23. Engaging with the Vision Map involves exploring each

theme by answering two key questions. First, they determine existing strengths and foundations—"What current resources or initiatives can support building this future?" This may include successful models or existing resources. This step ties the vision to reality, building on existing potential. Second, participants identify gaps and missing components—"What is lacking and needs development to make the vision real?" They address obstacles and areas needing innovation. By identifying strengths and gaps, the method balances achievements and challenges, fostering strategic and solution-focused discussions. The map's clarity aids in visualizing the vision's interconnected aspects and pinpoints areas needing focus. Unlike conventional methods, it maintains overall vision while breaking it into manageable parts. Effective in strategic planning and innovation, the Vision Map aligns forward-thinking with a realistic present assessment, crafting a feasible action plan to turn ambitious ideas into actionable steps toward the future.

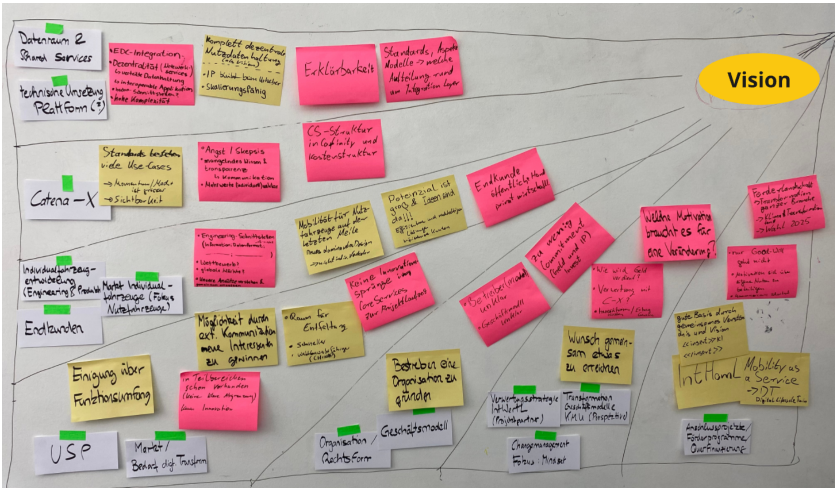

Vision
Datenraum & Shared Services
technische Umsetzung Plattform (?)
•EDC-Integration
•Dezentralität (Netzwerk-services)
↳ verteilte Datenhaltung
↳ interoperable Applikation
•techn. Schnittstellen?
•hohe Komplexität
Komplett dezentrale Nutzdatenhaltung (als Vision)
•IP bleibt beim Urheber
•Skalierungsfähig
Erklärbarkeit
Standards, Aspekte Modelle -> welche Aufteilung rund um Integration Layer
CS-Struktur in Cofinity und Kostenstruktur
Catena-X
Standards bestehen viele Use-Cases
-> Momentum / Markt ist grösser
-> Sichtbarkeit
°Angst / Skepsis
•mangelndes Wissen & transparenz
↳ Kommunikation
•Mehrwerte (individuell) unklar
Potenzial ist groß & Ideen sind da!!!
Endkunde öffentliche Hand privat wirtschaft
Förderlandschaft
-> Transformation ganzer Branche
-> Klima & Transformationsfond
-> Wahl 2025
Mobilität für Nutzfahrzeuge auf der letzten Meile
neues dominantes Design
-> nicht indiv. Verkehr
Individualfahrzeug-entwicklung (Engineering & Produktion)
Markt Individual-fahrzeuge (Fokus Nutzfahrzeuge)
•Engineering-Schnittstellen (Information; Datenformat; ...)
•Wettbewerb?
•globale Märkte?
•Unsere Ansätze verstehen & gemeinsam umsetzen
Keine Innovations-sprünge
Core Services zur Projektlaufzeit
zu wenig Commitment (Geld und IP) Invest
Welche Motivation braucht es für eine Veränderung?
•Wie wird Geld verdient?
•Verwertung mit C-X?
•Investitionen / Kosten / Ertrag Gewinn
•nur Good-Will geht nicht
•Motivation sich über eigene Nutzen zu beteiligen
•Betreiber(modell) unklar
•Geschäftsmodell unklar
Endkunden
Möglichkeit durch ext. Kommunikation neue Interessenten zu gewinnen
°Raum für Entfaltung
•schneller
•wettbewerbsfähiger (CHINA)
Einigung über Funktionsumfang
in Teilbereichen schon vorhanden (keine klare Abgrenzung)
kaum Innovation
Bestreben eine Organisation zu gründen
Wunsch gemeinsam etwas zu erreichen
gute Basis durch gemeinsames Verständnis und Vision
<<insert>> KI
<<insert>>
IntHomL
Mobility as a Service
-> DT
Digital Lifecycle Twin
USP
Markt / Bedarf digt. Transform.
Organisation / Rechtsform
Geschäftsmodell
Verwertungsstrategie IntWertL (Projektpartner)
Transformation Geschäftsmodelle KMU (Perspektive)
Changemanagement Fokus: Mindset
Anschlussprojekte / Förderprogramme / Querfinanzierung

Figure 5.23: Vision Map

In addition to adopting a long-term and strategic perspective, an examination of the current challenges was conducted. The four task forces initiated to facilitate the implementation of Synthesis Model V2 each adopted different approaches to related concepts, resulting in inconsistencies in process modeling. To address or at least identify these disparities, a workshop was held during the October 2024 general assembly. This workshop utilized a World Cafe format, allowing groups to rotate and explore four topics across three dimensions. The topics investigated were: *Business Domain of Individual Vehicle Development*, *Data Space / Shared Services*, *Onboarding and Network Expansion*, and *Vehicle and Configurator*. Each topic was examined through the lens of three dimensions: *Business Model of the Companies*, *Connection to Catena-X*, and *Legal Form of the IntWertL* ORCHESTRATOR. Although no tangible outcomes were achieved, the primary areas of concern were identified. It is crucial to address these issues cohesively within the model of the Digital Ecosystem to adequately define and implement the SOLUTIONS of the IntWertL Ecosystem by the project's conclusion.

5.3.2 Define Specific Requirements

After the conclusion of the vision workshops in this third iteration, three new vision statements were formulated. These statements define CORE VALUES and must be adhered to in the synthesis of CORE SERVICES:

1. **Why are we doing this?** The IntWertL Ecosystem empowers medium-sized companies to collaborate efficiently in the development and production of custom vehicles.
2. **What are we doing?** The IntWertL Ecosystem elevates the potentials of digital transformation by connecting sovereign companies through a common data space.

3. **How do we do that?** The IntWertL Ecosystem provides technical solutions and standardization in the area of core services: product portfolio management, engineering and manufacturing as a service, modular engineering and production, as well as homologation.

The third vision statement explicitly defines the requirements. We elaborate in Bönsch et al. (2024) how these requirements translate into six essential use cases for a Digital Ecosystem focused on individual vehicle development. Table 5.10 presents the functionalities considered in each of these use cases.

Table 5.10: Main topics in individual vehicle development

Topic	Description
Product Portfolio Management	Standardized framework for managing product Portfolios, making product information shareable and accessible across organizations.
Engineering as a Service (EaaS)	Simplifies and accelerates the development of complex systems, providing seamless access to specialized engineering capabilities.
Modular Engineering	Standardized approach to engineering collaboration, splitting development into manageable tasks with clear interfaces.
Manufacturing as a Service (MaaS)	Federated network of networks, enabling digital matching of manufacturing demand and capabilities.
Modular Production	Data models, interfaces and services, supplying information about production status and parameters.
Homologation	Streamling the homologation process, ensuring compliance with regulatory standards while reducing time, costs, and effort.
Note: MaaS and Modular Production are based on Catena-X use cases.	

5.3.3 General Problem and Requirements

The fundamental goal remains to establish a common reference for all stakeholders in the Digital Ecosystem. The third iteration synthesizes the first two approaches, thereby combining the previously established general requirements. Importantly, by integrating both structural and process insights, Version 3 of the Synthesis Model must provide layered perspectives to differentiate various concerns and prevent logical inconsistencies. Furthermore, the second iteration has demonstrated the necessity of focusing on the problem domain. Instead of detailing the SOLUTIONS, only the more abstract CORE SERVICES should be included in the description. The aim is to support strategic discussions and decision-making without confining participants to a rigid interpretation. Ultimately, the model should enable stakeholders to comprehend both static structures and dynamic interactions without requiring technical expertise.

The general requirements for the Synthesis Model (Iteration 3) are defined as follows:

I3.1 Display Active Entities: Identify and represent active entities involved in the Digital Ecosystem, especially CORE USERS.

I3.2 Provide Structure: Present a functional architecture that distinctly separates the CORE SERVICES into individual components.

I3.3 Allow Functional Breakdown: Add an abstraction layer to facilitate a more detailed decomposition of functions at a lower level.

I3.4 Support Complex Process Systems: Precisely describe interaction patterns and processes within the complex environment that is a Digital Ecosystem.

I3.5 Ensure Interpretability: Make the model self-explanatory, ensuring stakeholders can easily understand it. The model must be clear and interpretable without requiring technical expertise, enabling its use in discussions among diverse stakeholders.

I3.6 **Coherent views:** Make sure that structural information and process information are interconnected and not just stored side by side, to ensure coherent representation.

I3.7 **Open Solution Space:** Focus on the problem domain and do not restrict the solution space. Make sure to specify the framework of CORE SERVICES and not the SOLUTIONS themselves.

5.3.4 General Solution

The third iteration of the Synthesis Model builds upon the strengths of previous approaches by integrating both interaction-focused and functionally structured perspectives into a unified framework. Consequently, diagrams from SysML are combined with PASS to create a comprehensive toolkit for this iteration. The resulting model comprises four layers: Structure, Function, Interaction, and Behavior. The top two layers include diagrams adapted from SysML, specifically BDDs and UCDs, while the Interaction and Behavior layers consist of SIDs and SBDs, respectively. Each of the Structure-Function-Interaction-Behavior (SFIB) layers can incorporate diagrams with varying levels of detail or granularity and is defined by its ability to address a specific question:

- **Structure Layer: Who?** – Depiction of involved entities and their hierarchical breakdown. This includes *Actors* as well as *Domains* with their *Subdomains* and *Use Cases.*
- **Function Layer: Which?** – Depiction of *Use Cases* and *Functions* as well as associated *Actors* and *Roles.* Also displays the relationships between these entities.
- **Interaction Layer: What?** – Depiction of the interaction of *Subjects* and the exchange of (information) objects within a specific *Use Case* or *Function.*

- **Behavior Layer: How?** – Definition of the behavior of a *Subject*, that results in the previously described interaction.

Figure 5.24 illustrates the concept of connectivity among the SFIB layers. The F-layer contains one UCD for each use case.[9] This establishes a link between the structural description and enumeration of use cases in the S-layer and the detailed, process-oriented information in the IB-layers. This concept is further elaborated in the description of the resulting EDE methodology in section 6.3.

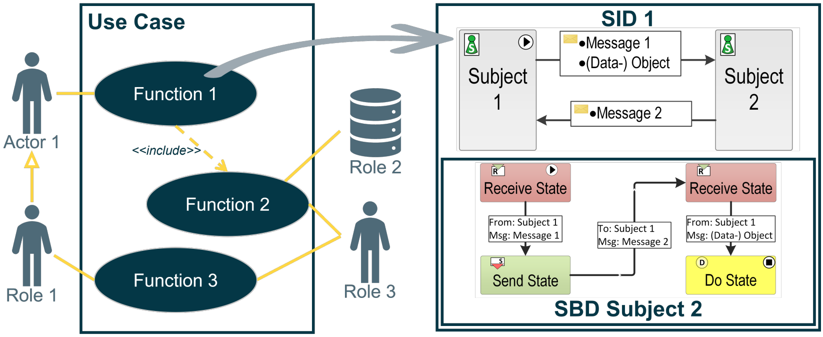


Figure 5.24: I3: Concept of linking FIB-layers.

5.3.5 Ex Ante Evaluation (General)

The Ex Ante Evaluation of the general solution is performed as an informed argument by modeling experts. The results are presented in Table 5.11 and Table 5.12. It is demonstrated that all general requirements can be sufficiently met by the proposed general solution.

[9] Note that the terminology and notation for UCDs in the Sysnthesis Model slightly differs from the terminology employed in SysML. See section 6.3 for a comprehensive explanation.

Table 5.11: I3: Matching of requirements to general solution (part 1)

Requirement	Solution
I3.1 Display Active Entities	• S-Layer: *Actors* depict at least CORE USERS. Other stakeholders such as CORE PARTICIPANTS or ORCHESTRATORS can be included. • F-Layer: *Actors, Roles* • IB-Layer: *Subjects*
I3.2 Provide Structure	• S-Layer: – Based on BDD, which is a diagram of structure. – Excluding all weak relationships. – Restricting multiple inheritance.
I3.3 Allow Functional Breakdown	• S-Layer: Actors can be organized hierarchically with generalization relationships. CORE SERVICES can be broken down in Domains, Sub-Domains, Use Cases and Functions. • F-Layer: Link IB-Layers to Use Cases or Functions. • IB-Layer: ALPS can provide additional abstraction layers.
I3.4 Support Complex Process Systems	• F-Layer: Link PASS diagrams to structure elements. • IB-Layer: PASS supports complex processes. See Table 5.4 from iteration 1.

The general solution of the third iteration primarily synthesizes the two preceding approaches. Consequently, a comprehensive understanding of the various diagram types is expected. The key innovation is the introduction of the F-Layer, which integrates structural and processual information.

Table 5.12: I3: Matching of requirements to general solution (part 2)

Requirement	Solution
I3.5 Ensure Interpretability	• Separating concerns into four different views (SFIB-Layer). • Extensive textual descriptions as part of the model possible. • S-Layer: Only four clearly labeled elements: (Actor, Domain, Subdomain, Use Case) with clear hierarchy. • F-Layer: UCD well known form of representation. Only three types of elements and four labeled relationships. • I-Layer: Only two modeling elements (subjects, messages). See Table 5.4 from iteration 1. • B-Layer: Only three states and corresponding transitions.
I3.6 Coherent Views	• Each diagram in a given layer is connected to an element of the Digital Ecosystem. • S-Layer: BDD of the structure of an Ecosystem. • F-Layer: UCD of a Use Case from the Ecosystem. • I-Layer: SID for a Function from the Use Case. • B-Layer: SBD for a Subject from the SID.
I3.7 Open Solution Space	• Actors, Roles, and Subjects are abstracted and not detailing real-world entities. • Only Domains, Use Cases and Functions of the Digital Ecosystem are specified. The implementation of SOLUTIONS is omitted by design.

5.3.6 Design Specific Solution

The Synthesis Model v3 provides a thorough description of the Digital Ecosystem as detailed by Bönsch et al. (2024). The SFIB model presented herein addresses all six critical Use Cases for inter-organizational collaboration within the *Individual Vehicle Development* (IVD) domain. These are:

- **Product Portfolio Management (PPM):** Standardized framework for managing product Portfolios, making product information shareable and accessible across organizations.
- **Engineering as a Service (EaaS):** Simplifies and accelerates the development of complex systems, providing seamless access to specialized engineering capabilities.
- **Modular Engineering (ME):** Standardized approach to engineering collaboration, splitting development into manageable tasks with clear interfaces.
- **Manufacturing as a Service (MaaS):** Federated network of networks, enabling digital matching of manufacturing demand and capabilities.
- **Modular Production (MP):** Data models, interfaces and services, supplying information about production status and parameters.
- **Homologation:** Streamling the homologation process, complying with regulatory standards while reducing time, costs, and effort.

Enabled by a collaborative, open, and secure data space aligned with Gaia-X and IDSA principles. This ecosystem approach fosters coopetition, thereby enhancing innovation, adaptability, and growth. This section will not detail all entities within the Synthesis Model V3. Instead, an overview of the contents and their mapping to model elements will be provided. This offers a clear understanding of the level of detail available in the model. Furthermore, significant design decisions will also be discussed.

5.3.6.1 Design Process

The design process involves the concurrent development of a JSON schema, a JSON model, and a model-viewer application, which are interrelated to facilitate iterative refinement. This approach emphasizes the necessity of simultaneous development to ensure that changes in one component are reflected in the others, thereby promoting a cohesive system. Additionally, the previously identified common weaknesses of the first two iterations are addressed by incorporating long-format textual descriptions as markdown files. The PASS models are stored in JSON-LD format and linked to the JSON model. The primary objective is to enable the visualization of graphical information directly from the JSON model. This capability aims to enhance discussions with stakeholders by providing visual representations that highlight various aspects of the project, thereby improving communication and feedback loops during the design phase.

An in-depth discussion of the representation of the Synthesis Model in JSON is presented in section 6.4.. In the remainder of this section, the generated SFIB-Layers using this JSON model are described and supplemented with graphical representations.

5.3.6.2 S-Layer

The S-Layer offers a structural overview of the Digital Ecosystem, as illustrated in Figure 5.25. In a manner analogous to the second iteration, Actors are positioned at the top, while Domains are located at the bottom of the diagram.

Many elements from the second iteration remain part of the Synthesis Model V3. The Core Users remain unchanged; however, to complete the representation of Core Participants, the actor section has been expanded to include the Contributors, which were previously only implicitly depicted as operators of Business-Apps / Services. Additionally, the Orchestrators

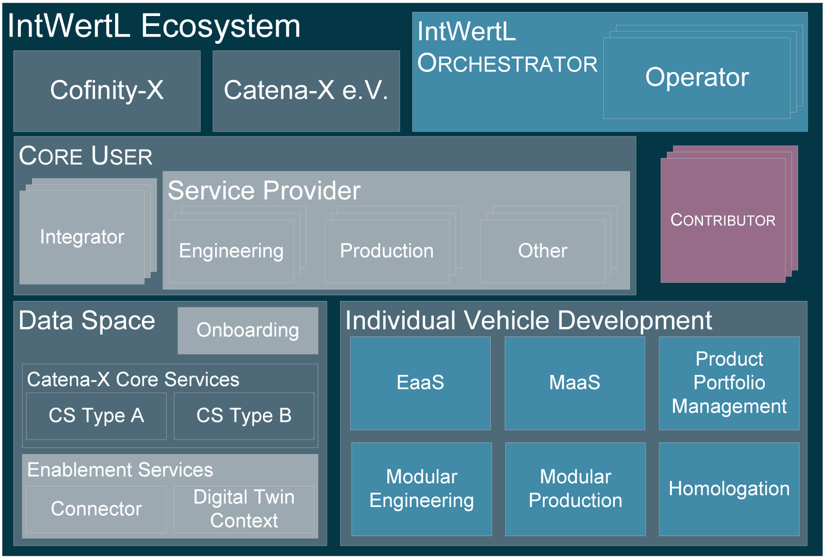


Figure 5.25: Synthesis Model V3 - S-Layer

are explicitly included in the third iteration of the model. In this iteration, the CORE SERVICES are carefully described as Use Cases and their Functions, rather than as SOLUTIONS. Consequently, the IVD Domain and its six Use Cases are shown in the S-Layer.

New in this third iteration is the stronger alignment with the Catena-X ecosystem. It has been decided that SOLUTIONS will be provided within the Catena-X data space. Consequently, it is essential for stakeholders to gain a better understanding of how IVD can be integrated with the existing Catena-X domains. This prompted the decision to include the Catena-X association and the primary Core Service Provider of Catena-X, Cofinity-X, as actors. Furthermore, the service map of the Catena-X operating system has been integrated with the Synthesis Model and is included as *Data Space* Domain.[10]

In IVD, four entirely new Use Cases are introduced, while MaaS and MP are based on existing Catena-X Use Cases. In IntWertL, additional functions for MaaS and MP are proposed to enhance IVD support; however, the main components are those described in Tractus-X. Homologation remains a significant topic for IntWertL, but it is primarily deemed out of scope. Consequently, the design of the three remaining IVD use cases, namely EaaS, ME, and PPM, becomes the primary focus at this stage of the research project.

5.3.6.3 F-Layer

The F-Layer includes UCDs for each of the eleven Use Cases in this third iteration. Furthermore, it illustrates the integration of the 59 Functions of the model into these Use Cases. Thirty of these Functions are assigned to the Data Space Domain. The remaining twenty-nine Functions are linked to the six IVD Use Cases, with some Functions assigned to multiple Use Cases. The resulting UCD for the ME Use Case is depicted in Figure 5.26 as an example of the F-Layer.

The UCD illustrates the five ME Functions designated within the ME Use Case. Additionally, relationships are illustrated: First, associations from the Actor *Engineering Provider* to the Functions *Engineering Forecast* and *Engineering Tracking* are included. These associations are general in nature and do not bear any labels. Both of these Functions have their own extend relationship with the *Activity Management* Function, which is not part of the ME Use Case (allocated in PPM). The *Conformity Assessment* Function from the Homologation Use Case extends the *Bill of Requirements* Function.

[10] Since this design decision was made Catena-X has redesigned its domain delineation. A Synthesis Model V4 of the IntWertL Ecosystem could then divide its use cases among the Catena-X domains: *Engineering, Industry Core, Manufacturing,* and *Network Services*.

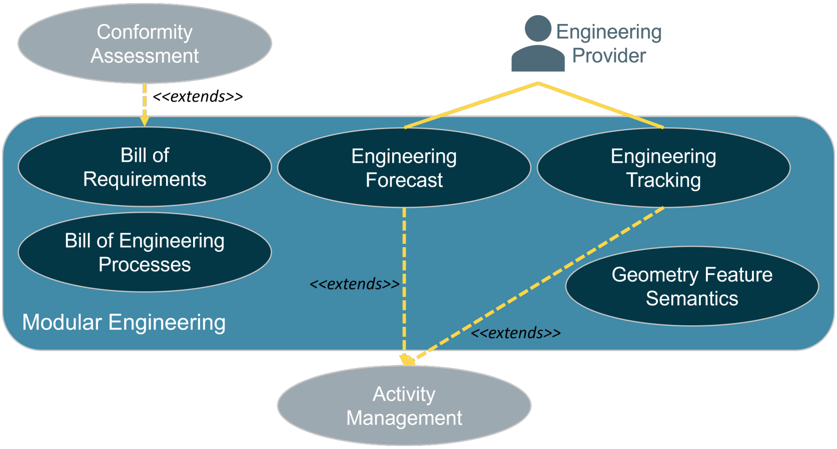


Figure 5.26: Synthesis Model V3 - F-Layer: Modular Engineering

However, the F-Layer also provides additional information. Each function allocated within a use case is described in greater detail. This includes a plain textual description as well as a tabular listing of the preconditions and postconditions that must be considered for the use or implementation of this function.

5.3.6.4 I-Layer

Each function may possess its own PASS model. The I-Layer presents the corresponding SIDs. In Figure 5.27 the interaction within the *Engineering Forecast* process is detailed.

This function details a peer-to-peer data exchange. The *EngineeringForecastData* model adheres to a defined schema, and three API calls for the *Engineering Forecast* function are specified in the data space: *GetEngineeringForecastData*, *ProvideEngineeringForecastData*, and *DELETE: relatedMessageID*. The exchange between the *Client* and the *EDC*, whether on the CONSUMER or PROVIDER side, is not required to rely on the specified API

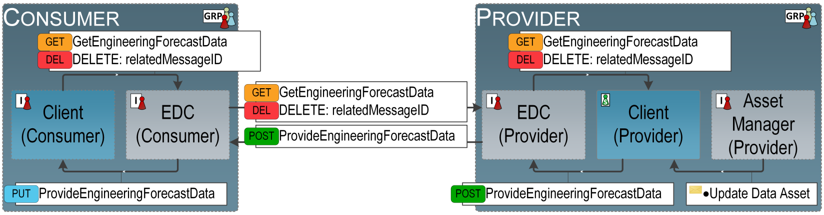

CONSUMER
GRP
GET GetEngineeringForecastData
DEL DELETE: relatedMessageID
Client (Consumer)
EDC (Consumer)
PUT ProvideEngineeringForecastData
GET GetEngineeringForecastData
DEL DELETE: relatedMessageID
POST ProvideEngineeringForecastData
PROVIDER
GRP
GET GetEngineeringForecastData
DEL DELETE: relatedMessageID
EDC (Provider)
Client (Provider)
Asset Manager (Provider)
POST ProvideEngineeringForecastData
•Update Data Asset

Figure 5.27: Synthesis Model V3 - I-Layer: Engineering Forecast

calls. However, the implementation of the relevant business logic must be validated and certified with the appropriate credentials.

5.3.6.5 B-Layer

Every fully specified Subject from the corresponding SID has an SBD. This SBD is represented on the B-Layer, as shown in Figure 5.28 for the *Client (Provider)* from the *Engineering Forecast* function.

Here, each *GetEngineeringForecastData* API call initiates an instance of the depicted behavior. The three possible data exchange modes specified by the *communicationMode* property are distinguished. If this property is *"synchronous"* then the behavior is completely linear. In the cases of *"cyclic"* or *"notification"* a loop is initiated. The loop only terminates after receiving the *DELETE: relatedMessageID* message.

5.3.7 Ex Ante Evaluation (Specific)

An extensive, analytical ex ante evaluation was conducted, involving detailed bilateral discussions to clarify and align the modeling of individual IVD use cases. Emphasis was placed on assessing and aligning the suitability of the General Solution, particularly regarding continuity from the S-Layer through the F-Layer to the IB-Layers. Special attention was dedicated to

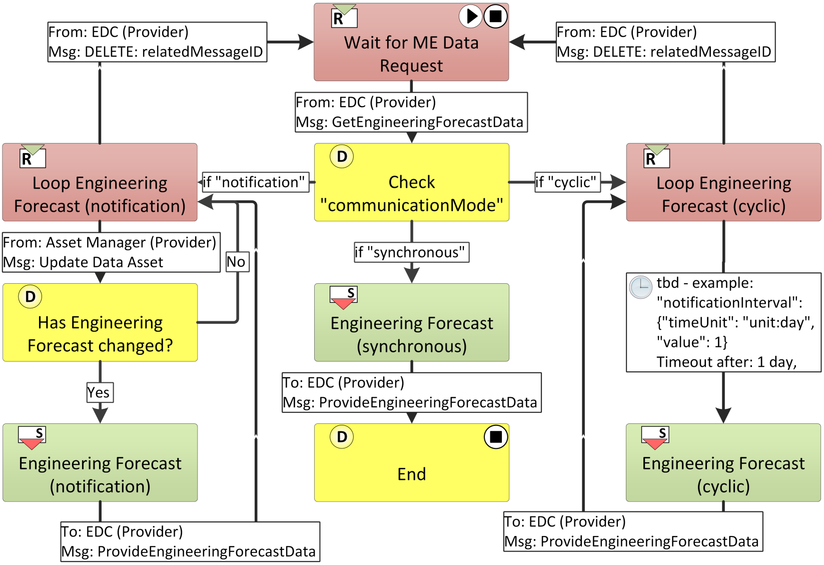

From: EDC (Provider)
Msg: DELETE: relatedMessageID
R
Wait for ME Data Request
From: EDC (Provider)
Msg: DELETE: relatedMessageID
From: EDC (Provider)
Msg: GetEngineeringForecastData
R
Loop Engineering Forecast (notification)
if "notification"
D
Check "communicationMode"
if "cyclic"
R
Loop Engineering Forecast (cyclic)
From: Asset Manager (Provider)
Msg: Update Data Asset
No
if "synchronous"
tbd - example:
"notificationInterval":
{"timeUnit": "unit:day",
"value": 1}
Timeout after: 1 day,
D
Has Engineering Forecast changed?
S
Engineering Forecast (synchronous)
Yes
To: EDC (Provider)
Msg: ProvideEngineeringForecastData
S
Engineering Forecast (notification)
D
End
S
Engineering Forecast (cyclic)
To: EDC (Provider)
Msg: ProvideEngineeringForecastData
To: EDC (Provider)
Msg: ProvideEngineeringForecastData

Figure 5.28: Synthesis Model V3 - B-Layer: Engineering Forecast - Client (Provider)

ensuring that the proposed SFIB-Layers meet the requirements of an inter-organizational Digital Ecosystem. Review sessions of the Synthesis Model V3 were held with modeling experts and domain specialists, representing all relevant stakeholders in engineering, production, and software development. Additionally, a focus group was convened with the IntWertL project management team. The Synthesis Model V3 was presented, and feedback was gathered. However, this feedback primarily targeted the implementation strategy and ways to improve the model's reception within the IntWertL community.

All six required use cases within the IVD domain are comprehensively documented in the Synthesis Model V3. As in previous iterations, the evaluation outcomes are classified into strengths (S), weaknesses (W), opportunities (O), and threats (T). In this context, strengths and weaknesses denote static attributes inherent to the model's structural composition. In contrast, opportunities and threats represent dynamic characteristics that emerge from the practical application of the artifact within its intended domain.

5.3.7.1 Strengths

- **Concise & precise:** Having dedicated Layers to answer different questions allows concise, yet very precise modeling. This focused separation of concerns ensures that stakeholders only see—and work with—the level of detail relevant to their responsibilities. Instead of combining structural and process views into a single diagram, each layer presents precisely what is needed for its specific viewpoint. Consequently, the model avoids redundancy and eliminates unnecessary overlap, enabling both brevity (when a high-level perspective suffices) and precision (when deeper technical detail is needed). This layered approach thereby sustains clarity across complex, inter-organizational environments while retaining the ability to capture fine-grained design considerations. For instance: Through the introduction of a single, streamlined API message for certain types of interactions, the I-Layer is simplified (compared to e. g. sequence diagrams). Behavior, such as different *communicationModes* is then detailed in the B-Layer. Consequently, the complexity is reduced, making the model easier to read and interpret. This precise modeling approach prevents unnecessary details from obscuring the high-level objectives of each Use Case.

- **Expressiveness:** Combining structural and processual information in a synthesized model yields comprehensive expressiveness. This holistic perspective reveals critical interdependencies and provides a multi-faceted view of the ecosystem, thereby enhancing communication and decision-making among diverse teams. If need be, additional textual (or even multi-modal) descriptions can be added through the use of markdown files.

- **Extensibility:** The Synthesis Model V3 can easily be extended. Additional Actors, Domains, Use Cases, and Functions can be included as new entities. The usage of an ID property that is separate from the name property in the JSON file makes the model entities easily maintainable. There is no requirement to immediately provide extensive markdown files, PASS diagrams for all Functions or even a complete breakdown of Actors. All of this can be added subsequently.
- **Open Solution Space:** The design of the Synthesis Model purposefully refrains from imposing strict constraints on technical implementations or vendor-specific platforms. By specifying only the core Functions, Use Cases, and Actors, it invites various CONTRIBUTORS to develop diverse SOLUTIONS, while preserving interoperability and semantic clarity. As a result, stakeholders retain autonomy in developing domain-specific implementations suited to their technological landscapes.
- **Simple Syntax and Schema:** Despite covering a wide range of functionalities, the Synthesis Model V3 employs a limited set of modeling elements (Actors, Domains, Functions, Use Cases, Subjects, States). This minimalistic yet structured schema lowers the entry barrier for new adopters. This is enhanced further as the more complex schema definition for PASS is only regarded in a linked but separate JSON-LD file. Additionally, it helps avoid confusion among stakeholders who may not be familiar with traditional, more complex modeling notations. The simplicity fosters clear communication and reduces the likelihood of misunderstanding or modeling errors.

5.3.7.2 Weaknesses

- **More complicated model:** Although the layered concept enhances expressiveness, it can make the Synthesis Model more complicated and complex. Understanding how the four layers interact (e. g., linking the structural hierarchy in the S-Layer to specific interactions in the I-Layer) may require specialized knowledge or guidance. The sheer amount of entities within the Synthesis Model can be overwhelming when trying to comprehend all at the same time.

5.3.7.3 Opportunities

- **Context Separation (Usage):** First goal remains to preserve a coherent view of the entire Digital Ecosystem. However, most adopters and developers will focus on one Use Case or even one Function at a time. The according FIB-Layers can easily be used, disregarding all other Use Cases for the time being. This separation of concerns paves the way for more focused stakeholder discussions and clearer delineations of responsibility within project teams. All the while being linked back to a coherent framework in the S-Layer.
- **Context Separation (Maintenance):** The underlying JSON model uses entities that are all maintained on the same level. Hierarchy is deducted from properties of the JSON objects in the model-viewer application. In combination with the usage of UUIDs for all main entities, this allows for complete separation of maintenanance tasks with subsequent merging of the model. This modular approach to maintenance reduces the risk of regression issues and fosters more streamlined version control.
- **Integration with Use Case repositories:** An ORCHESTRATOR needs to provide at least some kind of information on how to collaborate among CORE PARTICIPANTS. This typically consists of a combination of adoption, development and operation resources, typically provided through project repositories (such as Tractus-X). Similar repositories to the Tractus-X KITs can be used for the IVD Use Cases and the Synthesis Model information can be part of it. Even this iteration of the

Synthesis Model directly pulled markdown descriptions for MaaS and MP from the according Tractus-X repositories.

- **Support for System of Systems:** Digital Ecosystems can be understood as System of Systems. It is difficult to define the minimum viable Ecosystem. However, using the Synthesis Model allows to easily describe a "Super Ecosystem" composed of two to many Digital Ecosystems. For instance, there is a definite overlap of the IntWertL Ecosystem and the Catena-X Ecosystem. Therefore, aspects from the Catena-X Ecosystem have been used in the here presented IntWertL Ecosystem. Catena-X however has seven domains and over twenty Use Cases (and not just the Data Space domain presented here). A unified Synthesis Model, including all of these could easily be modeled with the presented general solution.

5.3.7.4 Threats

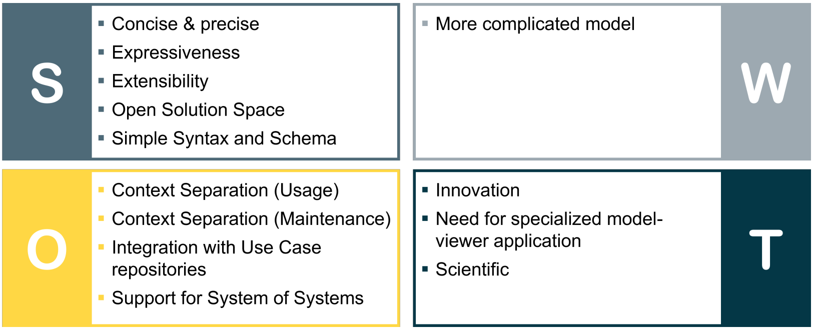
- **Innovation:** Design of a Digital Ecosystem is a real innovation. Therefore, a general understanding of the fundamentals of EDE can not be anticipated among stakeholders. Neither a common vision for the ecosystem nor its solutions are well established. Nevertheless, the aim is to disruptively change small-scale vehicle production. This change, as with every major disruption, will produce reservations of some kind.
- **Model-viewer:** Effective visualization of the Synthesis Model V3 requires a specialized viewer application. While a dedicated model-viewer can significantly improve clarity and user experience, it also introduces dependency on a particular software tool. If the viewer is not actively maintained, any updates to the schema may not be reflected properly. Furthermore, designing a proper UX for the model-viewer can be a complicated task and a topic for DSR research in its

own regard. Relying on this application can lead to rejection of the model itself just because of a bad UX.

- **Scientific:** In academic and industrial research contexts, maintaining rigor and reproducibility is paramount. However, the adopters and developers from the IntWertL community tend to be very pragmatic. Therefore, in-depth modeling following guidelines and a general solution can be interpreted as "too brainy" and not relevant to "the real world".

5.3.7.5 Summary Ex Ante Evaluation (Specific)

The summary of the analyzed SWOTs is illustrated in Figure 5.29. The ex ante evaluation of the Synthesis Model V3 highlights its ability to provide comprehensive clarity and rigorous technical detail in modeling inter-organizational Digital Ecosystems. By separating structure, function, interaction, and behavior into distinct layers, stakeholders can concentrate on their primary concerns while maintaining an understanding of the broader ecosystem context. This modularity simplifies maintenance, encourages incremental development, and fosters a more inclusive solution space where multiple Core Users and Contributors can collaborate.



S
- Concise & precise
- Expressiveness
- Extensibility
- Open Solution Space
- Simple Syntax and Schema

W
- More complicated model

O
- Context Separation (Usage)
- Context Separation (Maintenance)
- Integration with Use Case repositories
- Support for System of Systems

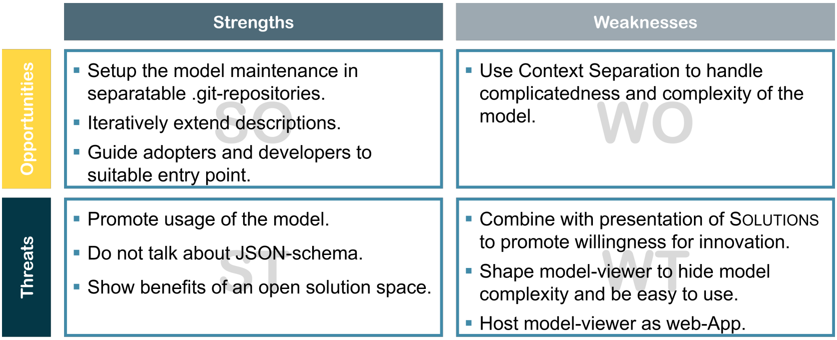
T
- Innovation
- Need for specialized model-viewer application
- Scientific

Figure 5.29: Synthesis Model V3 - SWOT

At the same time, the features that facilitate comprehensive expressiveness—such as layered views and specialized modeling tools—also introduce potential complexity. Full adoption may necessitate stakeholder training, careful tool support, and a collective commitment to modeling discipline. Despite these challenges, the Synthesis Model V3 offers a promising framework: its strengths in expressiveness and extensibility position it favorably to support the evolving needs of stakeholders from both industrial and research communities.

5.3.8 Implement Specific Solution

In the third iteration, the primary design risk for the Synthesis Model remains user-oriented. Consequently, an implementation strategy has been developed based on the specific Ex Ante Evaluation, as illustrated in Figure 5.30.

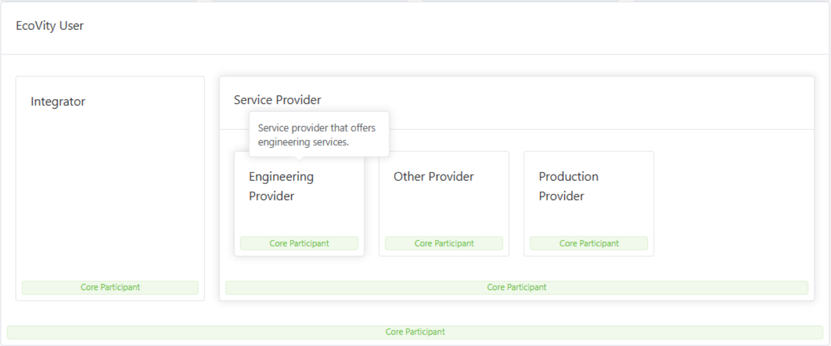


	Strengths	Weaknesses
Opportunities	▪ Setup the model maintenance in separatable .git-repositories. ▪ Iteratively extend descriptions. ▪ Guide adopters and developers to suitable entry point.	▪ Use Context Separation to handle complicatedness and complexity of the model.
Threats	▪ Promote usage of the model. ▪ Do not talk about JSON-schema. ▪ Show benefits of an open solution space.	▪ Combine with presentation of SOLUTIONS to promote willingness for innovation. ▪ Shape model-viewer to hide model complexity and be easy to use. ▪ Host model-viewer as web-App.

Figure 5.30: Synthesis Model V3 - Implementation Strategies

At the IntWertL general assembly in February 2025, all stakeholders were introduced to the Synthesis Model V3. As with previous versions, this edition included the S-Layer and a visual representation of functions linked to the IVD Use Cases, presented through PowerPoint slides. This method was chosen due to the incompatibility of the model-viewer application with projectors in large settings. However, a concise live demonstration of the model-viewer accompanied the presentation. Following the unveiling of the Synthesis Model V3, the audience was shown how the SOLUTIONS planned for implementation by the project's end connected with the displayed CORE SERVICES, thereby illustrating the open solution space. This introduction served as an accessible starting point for stakeholders interested in further exploring the model.

Shortly after the general assembly, the model viewer was made available as a web application. The link and instructions were disseminated via email, along with a brief survey intended to encourage immediate use of the model and to gather feedback for evaluation, as outlined in section 5.3.9. The model-viewer is designed to simplify navigation by masking complexity while ensuring user-friendliness. Entities within the S-Layer are represented as *Actor Cards* or *Domain Cards*. Figure 5.31 illustrates the CORE USERS.



Service provider that offers engineering services.

Figure 5.31: Synthesis Model Viewer - Actor Card

Tags indicate the ecosystem role of each individual actor. Hovering over any component displays a description for the card, while clicking it opens a drawer with more detailed information.

Figure 5.32 illustrates the *Domain Cards*. These cards include collapsible sections for Subdomains and Use Cases.

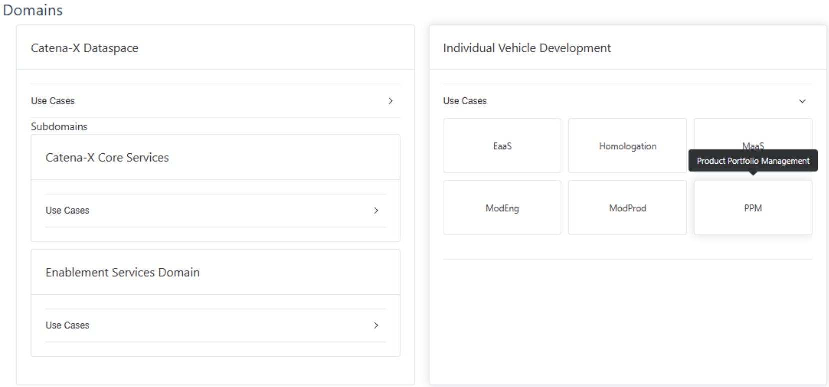


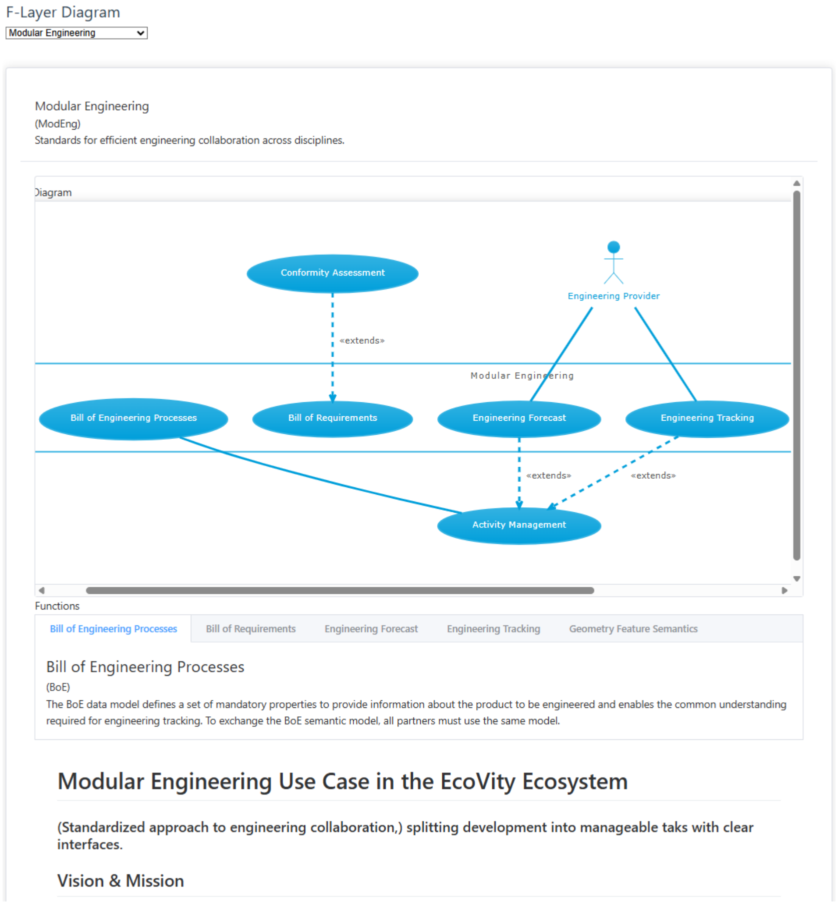
Figure 5.32: Synthesis Model Viewer - Domain Cards

The F-Layer for each Use Case can be accessed by clicking on a Use Case in the S-Layer or by using an additional site with a dropdown selection. The corresponding F-Layer in the model-viewer is illustrated in Figure 5.33.

To enhance model acceptance and mitigate risks associated with scientific complexity, the strategy should focus on presenting its concise and precise nature through direct experience, avoiding extensive discussion of the JSON schema to prevent creating an impression of a complicated theoretical construct.

5.3.9 Ex Post Evaluation

The third iteration of the Ex Post Evaluation has a more summative rather than formative nature. It is divided into two parts: first, the presentation of findings from the observational field study, which is consistent with previous iterations. Second, the results from the questionnaire, distributed to the IntWertL partners along with the link to the model-viewer application, are presented.



F-Layer Diagram

Modular Engineering ▾

Modular Engineering

(ModEng)

Standards for efficient engineering collaboration across disciplines.

Diagram

Functions

Bill of Engineering Processes | Bill of Requirements | Engineering Forecast | Engineering Tracking | Geometry Feature Semantics

Bill of Engineering Processes

(BoE)

The BoE data model defines a set of mandatory properties to provide information about the product to be engineered and enables the common understanding required for engineering tracking. To exchange the BoE semantic model, all partners must use the same model.

Modular Engineering Use Case in the EcoVity Ecosystem

(Standardized approach to engineering collaboration,) splitting development into manageable taks with clear interfaces.

Vision & Mission

Figure 5.33: Synthesis Model Viewer - F-Layer

In this third iteration, the artifact was utilized by multiple IntWertL participants prior to its implementation at the general assembly. Following the second series of vision workshops, it was time to transform the new CORE VALUES into corresponding CORE SERVICES. The primary objective is to establish a Digital Ecosystem that thrives even after the conclusion of IntWertL. This Digital Ecosystem, the product of IntWertL, is called EcoVity. Therefore, the preliminary Synthesis Model and the nascent EDE methodology were used to shape EcoVity. After implementation it was used to guide the breakdown of remaining deliverables until the conclusion of IntWertL, especially the SOLUTIONS that are to be demonstrated.

Therefore, the Ex Post Evaluation of this third evaluation is as follows:

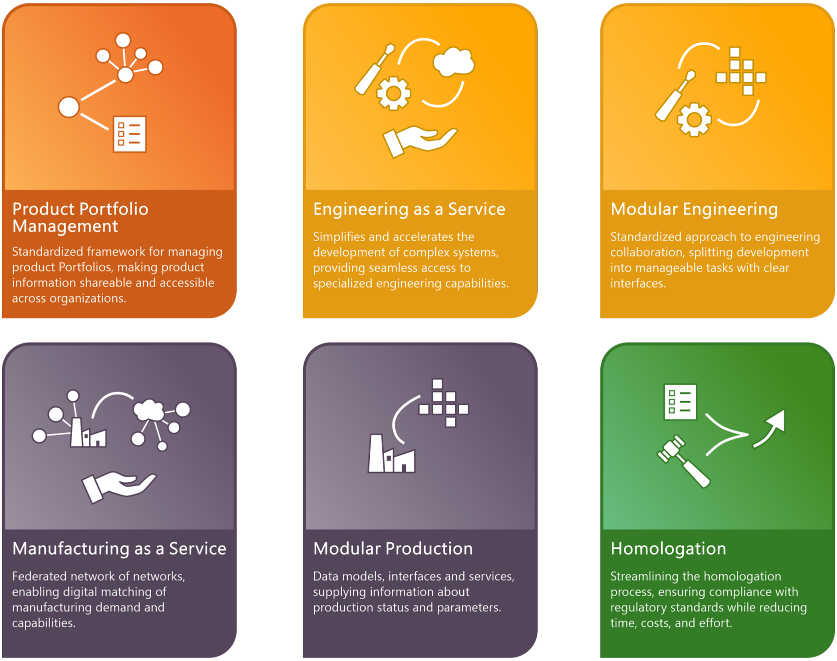
1. **Shaping EcoVity:** Pre implementation application of the Synthesis Model V3 to design the EcoVity Ecoystem.
2. **IntWertL Deliverables breakdown:** Usage throughout various work packages but especially for the definition of presentable SOLUTIONS.
3. **Questionnaire:** Summative evaluation of *model interpretability and consistency*, *model relevance and applicability*, as well as *resistance to actual model use*.

5.3.9.1 Shaping EcoVity

The initial step in further developing EcoVity was to draft a collaborative reference paper on the topic of *intelligent value creation networks for individual vehicle development* (Bönsch et al. 2024), which builds upon findings from the design process in this third iteration.

At the general assembly of IntWertL in December 2024, it became evident that the overall concept for EcoVity was still unclear. With just over one year remaining in the project, the necessity for a coherent concept for the emerging ecosystem became apparent. At that time, the Synthesis Model V3 was undergoing refinement, consolidating the design of the specific solution and the findings from the ex ante evaluation. Nevertheless, it was employed to integrate technical aspects and business model considerations, resulting in the *general concept* for EcoVity. This includes a perspective for SOLUTIONS to be implemented before the conclusion of IntWertL, as well as a vision for the long-term establishment of the Digital Ecosystem. This general concept was put to a vote and was unanimously accepted.

Design decisions from the third iteration and this general concept layed the foundation for the presentation of IntWertL and EcoVity at the Hannover Messe 2025.[11]. Figure 5.34 illustrates how the Core Services designed with the Synthesis Model V3 were presented at the fair.



Product Portfolio Management

Standardized framework for managing product Portfolios, making product information shareable and accessible across organizations.

Engineering as a Service

Simplifies and accelerates the development of complex systems, providing seamless access to specialized engineering capabilities.

Modular Engineering

Standardized approach to engineering collaboration, splitting development into manageable tasks with clear interfaces.

Manufacturing as a Service

Federated network of networks, enabling digital matching of manufacturing demand and capabilities.

Modular Production

Data models, interfaces and services, supplying information about production status and parameters.

Homologation

Streamlining the homologation process, ensuring compliance with regulatory standards while reducing time, costs, and effort.

Figure 5.34: Presentation of the use cases from Synthesis Model V3 at the Hannover Messe 2025

[11] IntWertL was selected to present itself at the joint stand of the Platform Industry 4.0, bridging the gap between the two main themes of *Mobility* and *Data Spaces*.

5.3.9.2 IntWertL Deliverables Breakdown

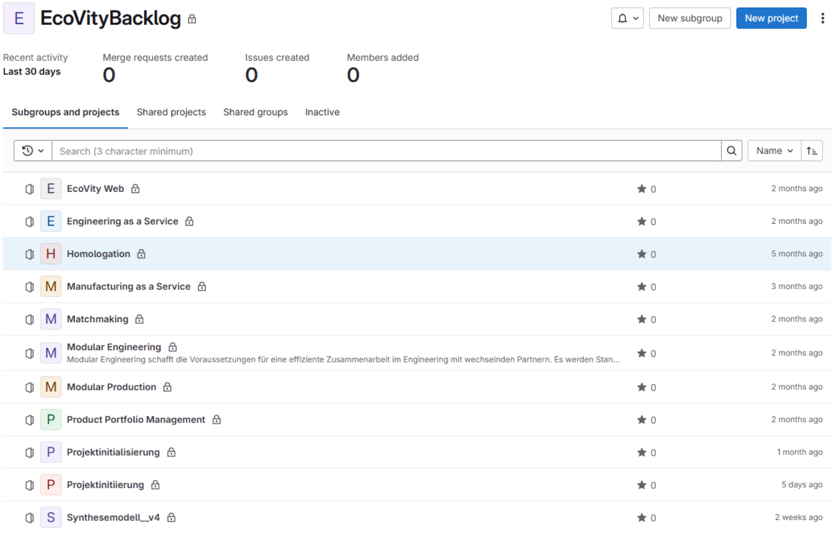
Following its implementation, the third version of the Synthesis Model was rapidly adopted. Beginning at the general assembly in February 2025, monetization strategies based on the CORE SERVICES and actor decomposition were presented to IntWertL stakeholders. The ORCHESTRATORS from the Synthesis Model were also considered in the presentation of an unbundling business model for EcoVity.

Second, the software development partners of IntWertL utilized the Synthesis Model V3 to define and approve the software demonstrators intended to exhibit the realization of CORE SERVICES in the implemented SOLUTIONS. The interfaces between these SOLUTIONS are to be defined in accordance with the Synthesis Model, including process descriptions and data models associated with the GitLab repositories of the use cases. The setup of the corresponding *EcoVityBacklog* repository is depicted in Figure 5.35. The use case-related project repositories adhere to the terminology of the Synthesis Model and are intended to be published as open source at the conclusion of IntWertL to promote the integration of the designed CORE SERVICES.

Finally, not only did software developers change their deliverables to align with the Synthesis Model, but stakeholders from the engineering domain did as well. Instead of document-based platform descriptions, they committed to integrating their results into the corresponding Markdown files.

5.3.9.3 Questionnaire

Lastly, IntWerL participants completed a short questionnaire to qualitatively and summatively evaluate the Synthesis Model. The questionnaire was distributed alongside access to Synthesis Model V3 in the form of the model-viewer application. Out of 87 unique visits to the survey, 16 complete questionnaires were collected. Only these 16 will be evaluated.



E EcoVityBacklog

Recent activity
Last 30 days

Merge requests created 0

Issues created 0

Members added 0

E	EcoVity Web	★ 0	2 months ago
E	Engineering as a Service	★ 0	2 months ago
H	Homologation	★ 0	5 months ago
M	Manufacturing as a Service	★ 0	3 months ago
M	Matchmaking	★ 0	2 months ago
M	Modular Engineering Modular Engineering schafft die Voraussetzungen für eine effiziente Zusammenarbeit im Engineering mit wechselnden Partnern. Es werden Stan...	★ 0	2 months ago
M	Modular Production	★ 0	2 months ago
P	Product Portfolio Management	★ 0	2 months ago
P	Projektinitialisierung	★ 0	1 month ago
P	Projektinitiierung	★ 0	5 days ago
S	Synthesemodell_v4	★ 0	2 weeks ago

Figure 5.35: GitLab repository of the EcoVityBacklog

The survey comprises nine questions rated on a ten-point Likert scale. The final question allows for a free-text response, soliciting general feedback and suggestions for improvement. The first nine questions are organized in three categories of three questions each:

1. Part 1: model interpretability and consistency
2. Part 2: model relevance and applicability
3. Part 3: resistance to actual model use

The following paragraphs present the results for all survey questions. Each section of the questionnaire is illustrated with a radar chart and a histogram. The numbers in parentheses on the radar chart represent the mean values. The mean is utilized as the ten-point Likert scale is considered quasi-metric. Due to the limited number of participants, this survey lacks quantitative robustness; therefore, no additional statistical analyses are conducted.

Part 1: Model Interpretability and Consistency

Figure 5.36 presents the results of the first part of the questionnaire. The statements are evaluated using a ten-point Likert scale ranging from *1|Fully agree* to *10|Do not agree at all.*

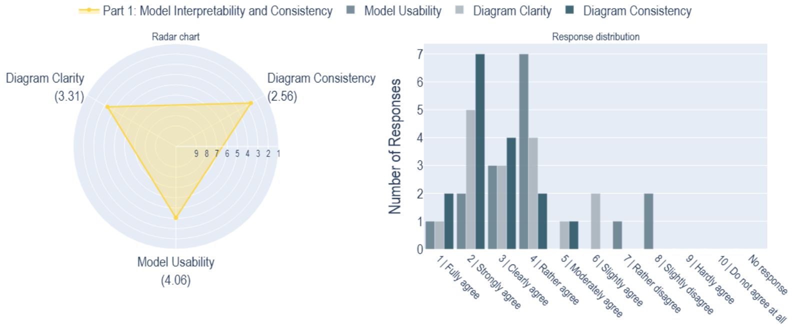


Figure 5.36: Questionnaire part 1: Model Interpretability and Consistency

The three statements are:

1. **Model Usability:** The model is easy to understand and intuitive to use.
2. **Diagram Clarity:** The individual diagrams are clearly structured and easy to interpret.
3. **Diagram Consistency:** The various diagrams are logically coordinated and do not contradict each other.

Part 2: Model Relevance and Applicability

Figure 5.37 presents the results of the second part of the questionnaire. The statements are evaluated using a ten-point Likert scale, ranging from *1|Fully agree* to *10|Do not agree at all.*

The three statements are:

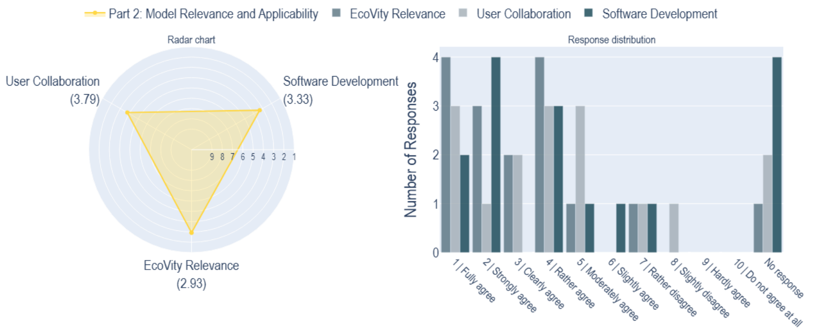


Figure 5.37: Questionnaire part 2: Model Relevance and Applicability

4. **EcoVity Relevance:** The model is relevant for the design of the Digital Ecosystem (EcoVity).

5. **User Collaboration:** The model is relevant for collaboration between platform users (integrators, engineering, production, etc.).

6. **Software Development:** The model is relevant for software development, especially for planning, implementation, maintenance, and documentation of software projects.

Part 3: Resistance to Actual Model Use

Figure 5.38 presents the results of the third part of the questionnaire. The statements are evaluated using a ten-point Likert scale ranging from *1|No resistance at all* to *10|Extreme resistance*.

The three statements are:

7. **Ecosystem Description:** The model explains the most important relationships of the Digital Ecosystem (EcoVity).

8. **Future Usage:** I will use the model in the further course of the IntWertL project.

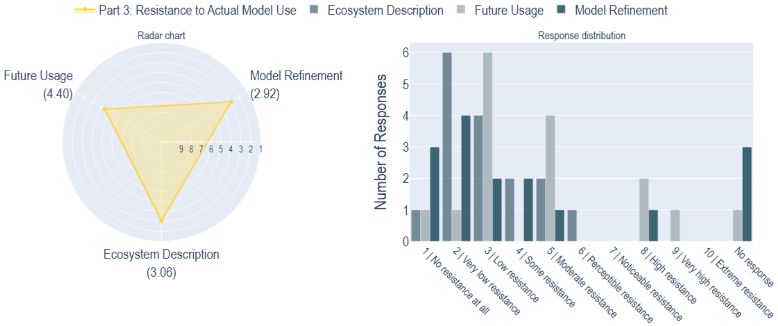


Figure 5.38: Questionnaire part 3: Resistance to Actual Model Use

9. **Model Refinement:** I will support further development of the model's content during the project.

Part 4: Feedback

The questionnaire yielded four feedback statements:

- "The transition from theory/science to practice and demonstrating feasibility are still largely missing. The model itself is brilliant, but implementation is all the more difficult. It should be demonstrated more how this is supposed to work."
- "What I am missing is a variant without using the EDC."
- "The model is very good, but the implementation is somewhat too theoretical and therefore difficult to realize in practice."
- "The site [URL] is probably still under construction. It is somewhat distracting that clicking on the use case diagrams results in a 404 error, but you are probably working on that. For usability, it would be good if one 'learns' that things have changed on the page when you select a synthesis model at the front. Hope this helps. Best of luck with everything and your doctoral dissertation."

Additionally, some feedback was communicated in person. The most notable request came from a project partner who inquired whether they could access the source code of the Synthesis Model and the model-viewer application. The project partner intended to utilize the Synthesis Model to catalog and describe all software services implemented within their department, thereby enhancing the understanding of their collective work.

5.3.9.4 Summary Ex Post Evaluation

Overall, it is evident that the Synthesis Model V3 is extensively utilized across IntWertL. In this third iteration, the concept of Synthesis Models has become firmly established within the research project, being applied even before their official release and implementation with all stakeholders. The Ex Post Evaluation illustrates the application in business model formulation, the development of SOLUTIONS, and the design of CORE SERVICES.

The conducted survey indicates a strong principle endorsement of the Synthesis Model. Moreover, there appears to be low resistance to both the modeling approach and its application. Nonetheless, some reservations persist among stakeholders. In accordance with the specific ex ante evaluation, feedback highlights concerns related to implementation and usability rather than criticisms of the Synthesis Model itself.

5.3.10 Conclusion Iteration 3

The third iteration of the SDSM application produced a synthesized artifact that integrates structural and processual information. This is achieved by distinctly focusing on describing CORE SERVICES of the IntWertL ecosystem. The presentation employs a layered yet interconnected approach, separating concerns into interlinked but standalone diagrams. Consequently, the general solution necessitates a customized model representation. The corresponding JSON models are visualized using a dedicated model-viewer application. Thus, during the summative evaluation, the distinction between evaluating the model and assessing the application becomes less clear. Nevertheless, observations from the conducted field studies indicate the rapid and widespread adoption of the Synthesis Model V3.

The third iteration provides the most comprehensive, albeit complex, description of the IntWertL ecosystem. Consequently, it also generates the most preliminary versions. By embracing the concept of design as a search process and aligning with the model-based approach introduced in section 6.2 these versions were promptly utilized by stakeholders. As a result, the third iteration influenced the ecosystem design even prior to the implementation of models. This illustrates that a model-based approach does not necessitate reliance on static and complete models.

5.4 Summary

This chapter outlines the design process of three versions of Synthesis Models as Level 1 artifacts. It presents three general solutions, three specific instantiations, and nine comprehensive evaluation episodes. Moreover, knowledge contributions can be derived from the rigorous design process itself.

The first iteration employed a process-centric approach, utilizing PASS SIDs as a general solution. Consequently, the Synthesis Model V1 served as the foundation for the MVP implementation and the initial development of the IntWertL ecosystem. Nevertheless, a Digital Ecosystem cannot be comprehensively described as a singular process.

The second iteration focused on structural issues, utilizing an adapted BDD as representation. This approach enabled the development of business and operational models, alongside the breakdown of functional architecture. However, the actual implementation of services was not addressed.

The third iteration synthesized the previous two approaches. It employed four diagram types: BDD, UCD, SID, and SBD, interlinking them to create a coherent representation.

Findings from all three iterations establish the foundation for formulating the EDE methodology as a Level 2 artifact. This emergent methodology is detailed in the following chapter. The critical reflection on the Level 1 artifacts in relation to the research hypotheses is included in Chapter 7.

6 The Engineering Digital Ecosystems Methodology

This chapter introduces the *Engineering Digital Ecosystems (EDE)* methodology, which serves as a generalized foundation for the model-based and subject-oriented design of Digital Ecosystems. It builds on the findings from the three iterations of designing Digital Ecosystem models for individual vehicle development (Synthesis Model v1-3), as described in Chapter 5, and consolidates the derived Level 2 knowledge in accordance with the DSR framework (Gregor and Hevner 2013). Thus, the EDE methodology represents the primary outcome and main artifact of this research study.

The EDE methodology consists of two complementary components: first, the *EDE Approach*, a structured methodological approach that guides the design process; and second, the *Synthesis Model*, a formal metamodel that facilitates the subject-oriented and model-based (re-) presentation of Digital Ecosystems. [1] Together, these components tackle the fundamental challenges of EDE—specifically, the inherent complexity, federated structure, and diversity of stakeholders typical of Digital Ecosystems.

This chapter is organized as follows: Section 6.1 lays out the conceptual framework of the EDE methodology. Section 6.2 elaborates best practices

[1] In Chapter 5, the term *Synthesis Model* often refers to a specific model instantiation. In the present Chapter 6, *Synthesis Model* designates the metamodel, i. e., the set of concepts and construction rules from which such instances are derived.

and methods for designing Digital Ecosystems. Section 6.3 outlines the layered structure of the Synthesis Model, followed by its technical representation in a JSON schema in Section 6.4. Section 6.5 presents an exemplary practical implementation, demonstrating how instantiations of Synthesis Models can be collaboratively maintained through Python-driven GitLab workflows and visualized using a model viewer application. The chapter concludes with a summary.

6.1 Concept of the EDE Methodology

Designing Digital Ecosystems presents unique challenges due to their distributed and federated nature, as well as the involvement of diverse, (semi-) autonomous stakeholders. The EDE methodology addresses these challenges by introducing a structured, iterative, stakeholder-centered approach for designing such ecosystems at both conceptual and architectural levels. This methodology aims to bridge the gap between abstract ecosystem thinking and collaboratively designed functional architectures.

The EDE methodology is both *model-based* and *subject-oriented*. It provides a formal foundation for describing the roles, interactions, and value co-creation mechanisms that characterize Digital Ecosystems. Rather than focusing on software implementation or deployment (SOLUTIONS), the EDE methodology facilitates the structured design of ecosystems, their CORE VALUES and CORE SERVICES.

6.1.1 Methodological Foundations

The conceptual foundation of the EDE methodology integrates insights from three complementary traditions:

- **Design Thinking**, which emphasizes human-centered innovation and stakeholder inclusion in iterative design cycles;

- **Systems Thinking**, which provides a holistic view of the interdependencies and feedback mechanisms in complex, multi-actor systems;
- **Design Science Research (DSR)**, which grounds the methodology in a rigorous research process aiming at the development of prescriptive knowledge.

In particular, the EDE methodology produces what DSR refers to as *Level 2 knowledge* (Gregor and Hevner 2013): prescriptive design knowledge that is generalizable across a class of problems, specifically related to the design of Digital Ecosystems.

6.1.2 Methodological Scope and Boundaries

The EDE methodology is specifically focused on the design phase of Digital Ecosystems. It offers a structured approach to problem framing, stakeholder alignment, and the definition of functional architecture. However, it does not stipulate implementation strategies, development workflows, or deployment architectures. These responsibilities are assumed to be managed by subsequent engineering processes, informed by the validated outputs of the EDE design process.

6.1.3 Structure of the EDE Methodology

The conceptual structure of the EDE methodology is illustrated in Figure 6.1. It comprises two main components: the *EDE Approach* and the *Synthesis Model*. The approach includes a CORE VALUES cycle and a CORE SERVICES cycle, while the Synthesis Model is subdivided into presentation and representation. Each of these three aspects (approach, presentation, representation) is described in greater detail in the subsequent sections.

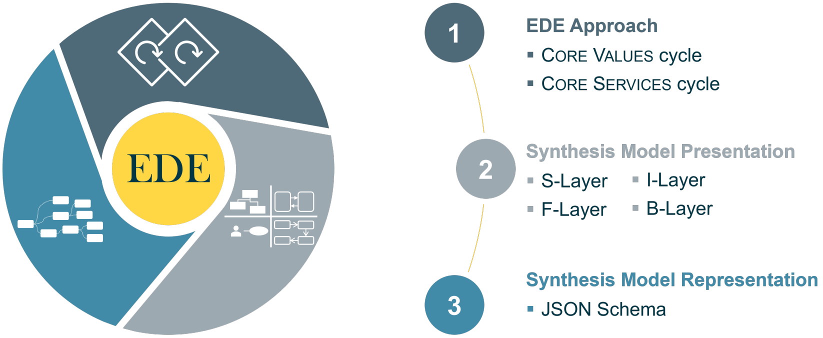


Figure 6.1: Conceptual structure of the EDE methodology.

6.1.4 EDE Framework

The EDE methodology is operationalized through a structured decomposition framework that connects ecosystem-level thinking with specific data usage descriptions. The hierarchy of decomposition is illustrated in Figure 6.2.

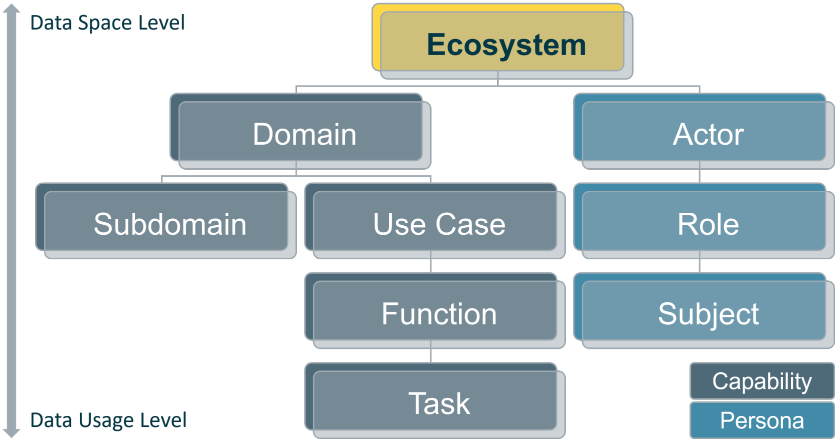


Figure 6.2: EDE methodology - decomposition framework.

This framework supports two primary decomposition pathways:

- **Persona decomposition**: from ecosystem *actors* to *roles* to *subjects*, enabling the assignment of responsibilities and modeling of interactions.
- **Capability decomposition**: from *domains* to *subdomains*, to *use cases*, *functions*, and finally to *tasks*, providing a layered approach to defining ecosystem functionalities.

Table 6.1 describes the breakdown of *What?* the ecosystem is supporting, while Table 6.2 cascades the *Who?* is involved.

Table 6.1: EDE methodology: Capability Decomposition framework

Concept	Description
Domain	Broad areas of business activity within the ecosystem, encompassing related subdomains and use cases.
Subdomain	A more specific category of a domain, focusing on particular aspects of business operations.
Use Case	Specific scenarios that describe interactions between actors and the system, outlining goals and outcomes.
Function	High-level, logically encapsulated operations that are part of a use case, detailing actions that contribute to achieving the use case goals.
Task	Smaller, actionable items or steps that need to be completed as part of a function.

This decomposition logic forms the structural basis for the Synthesis Model and consequently, the EDE methodology.

Table 6.2: EDE methodology: Persona Decomposition framework

Concept	Description
Actor	Active entities on ecosystem level. Mostly ORCHESTRATORS and CORE PARTICIPANTS. (e. g., humans, organizations, or external IT systems).
Role	A persona on use case level; can specify or generalize actors.
Subject	The concrete instantiation of any active entity (organizations, IT systems, people) in a particular function context, including their task execution behavior.

6.2 The EDE Approach

The EDE approach adapts the UK's Design Council's Double Diamond model to meet the specific needs of Digital Ecosystems. Each diamond signifies a complete cycle of divergence and convergence. The resulting EDE Double Diamond is depicted in Figure 6.3.

This structure is based on a simplified version of the SDSM methodology, utilized in the three iterations discussed in Chapter 5. The EDE Approach omits the identification of the general problem, requirements, and solutions, which are provided through the EDE methodology. All evaluation steps are deliberately excluded as well, as they must be specifically designed to fit the design task at hand (see Section 4.2.3. Instead, it focuses exclusively on the design phase, facilitating stakeholder alignment, ecosystem modeling, and the formalization of the system's functional architecture.

- **CORE VALUES Cycle** (Diamond 1) – *Analyze → Define*
  Aligns participants around a shared purpose and a concise set of CORE VALUES.

- **CORE SERVICES Cycle** (Diamond 2) – *Create → Synthesize*
  Translates those values into a coherent portfolio of CORE SERVICES captured in the SynthesisModel.

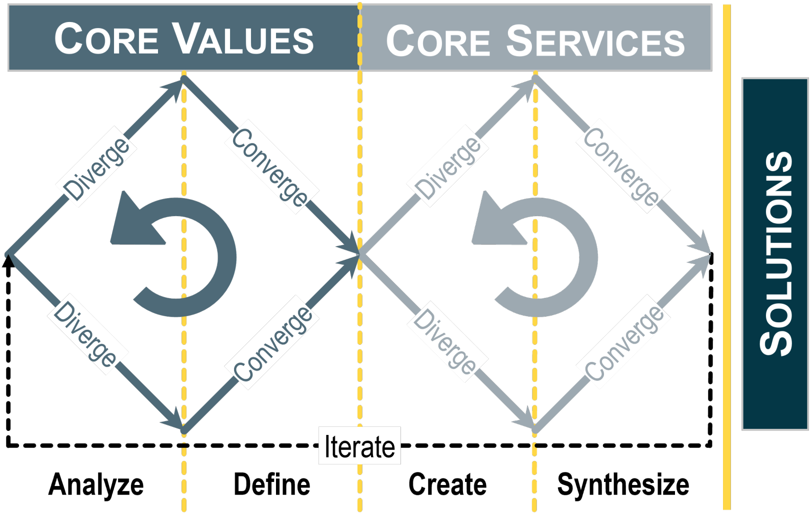


Figure 6.3: The EDE Double Diamond: Defining CORE VALUES and synthesizing CORE SERVICES.

These two cycles are intentionally lightweight while also being highly collaborative, allowing for iteration whenever the ecosystem evolves. By the end of a complete cycle, the Digital Ecosystem possesses:

- a stakeholder-validated set of CORE VALUES, and
- a synthesized, subject-oriented model of CORE SERVICES.

The CORE SERVICES can then be implemented in various SOLUTIONS. However, a Digital Ecosystem is a dynamic entity. It is never complete; therefore, regular revision of CORE VALUES and extension of CORE SERVICES are necessary.

The primary goals, activities, and artifacts for each step of the EDE Double Diamond are summarized in Figure 6.4.

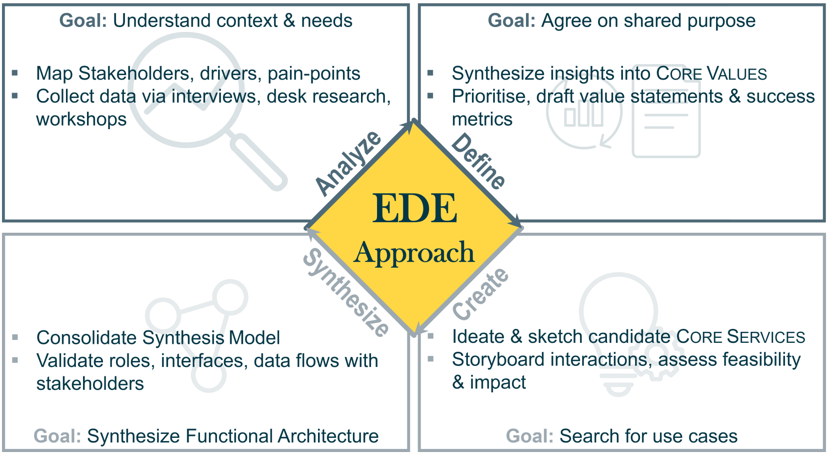

Goal: Understand context & needs
Map Stakeholders, drivers, pain-points
Collect data via interviews, desk research, workshops
Goal: Agree on shared purpose
Synthesize insights into CORE VALUES
Prioritise, draft value statements & success metrics
Analyze
Define
EDE
Approach
Synthesize
Create
Consolidate Synthesis Model
Validate roles, interfaces, data flows with stakeholders
Goal: Synthesize Functional Architecture
Ideate & sketch candidate CORE SERVICES
Storyboard interactions, assess feasibility & impact
Goal: Search for use cases

Figure 6.4: Four step approach of the EDE methodology.

Step 1 | Analyze Map the ecosystem's boundaries, actors, and prevailing incentives through desk research, stakeholder interviews, and contextual inquiry. Special attention is paid to latent needs and potential conflicts that might later hinder adoption. The outcome is a comprehensive problem frame and an evidence-based inventory of stakeholder goals.

Step 2 | Define Translate the collected insights into a non-overlapping, compact set of CORE VALUES that answer the question *"Why should we collaborate?"*. Facilitation techniques, such as value-proposition canvases and prioritization matrices, assist participants in negotiating trade-offs and reaching consensus on success metrics. A brief "value charter" is created to document the alignment and serve as a stable reference for subsequent design work.

Step 3 | Create Ideate candidate CORE SERVICES that operationalize each value, transitioning swiftly from napkin sketches to service blueprints or low-fidelity mock-ups. Feasibility, desirability, and ecosystem-level impact are evaluated in co-creation workshops, often employing rapid prototyping or value-stream mapping. The outcome is a prioritized backlog of service concepts, each linked to the value(s) it supports.

Step 4 | Synthesize Consolidate the shortlisted services into a cohesive functional architecture formalized in the Synthesis Model. This involves defining roles, interaction patterns, data structures, and governance rules, followed by validation with domain experts through walkthroughs or simulations. The model serves as the authoritative blueprint for downstream implementation and as a dynamic artifact to be updated as the ecosystem evolves.

Guiding Design Principles

1. **Manage emergence** instead of enforcing control.
2. **Iterate** rather than follow a linear sequence.
3. Balance **divergence and convergence**.
4. Emphasize **co-creation** with ecosystem stakeholders.
5. Maintain **systemic sensitivity** and context awareness.

The following subsections detail the activities, artifacts, and recommended methods for each step.

6.2.1 Analyze

The *Analyze* step serves as the divergent opening move of the EDE methodology. Its objective is to draw an evidence-based picture of the Digital Ecosystem, examining CORE PARTICIPANTS and their needs, as well as the existing structures within the application domain. Building on lessons learned from the three iterations in Chapter 5, we discovered that merely “collecting data” is insufficient; this step must establish a shared problem frame that can accommodate limited resources, diverse stakeholder perspectives, and initially fuzzy requirements. By explicitly broadening the design space, this initial step safeguards subsequent work from blind spots and premature solution bias. Figure 6.5 illustrates the purpose, task, and application of the *Analyze* step.

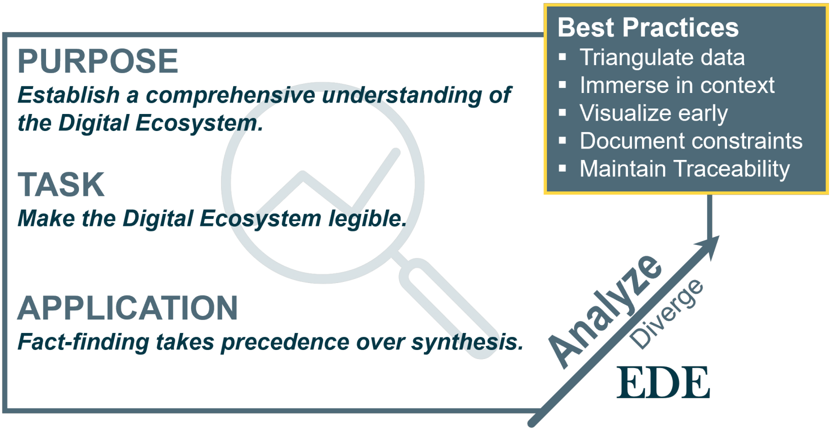


Figure 6.5: Step 1 - Analyze: Purpose, Task, and Application.

6.2.1.1 Purpose

Establish a comprehensive understanding of the Digital Ecosystem before any design decisions are taken.

The primary obligation is visibility, not consensus. The analysis offers a holistic view of the Digital Ecosystem, incorporating commercial, regulatory, and technical perspectives to establish a common factual baseline. A disciplined, data-driven assessment up front creates an explicit problem frame that designers, domain experts, and sponsors can confidently reference.

6.2.1.2 Task

Make the Digital Ecosystem legible: Who is involved, how is value created and exchanged, and where do unmet needs or structural barriers lie?

Turn the fuzzy "idea of an ecosystem" into a clear design brief that all stakeholders can reference. Concretely, this involves:

- **Demarcate the arena:** Outline clear ecosystem boundaries and adjacent domains.
- **Map the cast:** List actors (people, organizations, systems) and chart how value (data, money, reputation) flows among them.
- **Surface intentions & pain points:** Document stakeholder goals, frictions, and success metrics through observation and dialogue.
- **Log external forces:** Record regulatory, technological, and market factors that could enable or constrain the ecosystem.
- **Link evidence:** Store raw quotes, statistics, and field notes alongside every insight for traceability.

6.2.1.3 Application

Deliberately diverge: fact-finding takes precedence over synthesis.

The *Analyze* step operates in a deliberately divergent mode: fact-finding takes precedence over synthesis. Therefore, a cyclical breadth-to-depth research cadence should be employed. First *widen the lens* and then *zoom in* until informational saturation is achieved—typically after two to three iterations:

1. **Widen the lens:** Cast a broad net to surface facts and signals from four complementary views:

  - Process: how work is done (e. g., process exploration, lightweight subject-oriented modeling);
  - Need: what users and stakeholders want or struggle with (e. g., high-volume user stories, interviews);
  - System: how tangible and intangible value circulates (e. g., value-network and ecosystem mapping);
  - Strategy: why each actor participates and how success is framed (e. g., vision-building workshops, horizon scans).

2. **Zoom in:** Probe the most consequential discoveries: domain standards, regulatory pinch points, legacy architectures, developing collaborations. Deep-dive methods such as value-network analysis, targeted expert interviews, and mini vision workshops confirm or refute hypotheses.

Alternating between breadth and depth prevents tunnel vision while maintaining a manageable workload. Each cycle enhances insight, ultimately resulting in an evidence-linked repository that supports subsequent design decisions. When new cycles merely confirm existing patterns, the *Analyze* step is considered complete.

Best Practices

- **Triangulate data:** Combine qualitative and quantitative sources (e. g., interviews, surveys, desk research) to avoid single-method bias.
- **Immerse in context:** Short ethnographic site visits or contextual inquiry expose tacit routines and hidden frictions.
- **Visualize early:** Ecosystem and value-flow diagrams give stakeholders a common reference and reveal gaps.

- **Document constraints:** Log regulatory hurdles, incentive conflicts, or technical bottlenecks while they surface.
- **Maintain traceability:** Capture quotes, user stories, and workshop artifacts in a living repository so later design choices stay evidence-linked.

Suggested Methods

- **Contextual Inquiry:** Observe users in situ, asking clarifying questions while they work to uncover tacit practices and unmet needs.
- **Ecosystem Mapping:** Graphically represent the overall ecosystem structure, value flows, and connections, including dependencies, gaps, and interaction points.
- **Journey Mapping:** Visualize end-to-end experiences for archetypal personas, flagging friction points and opportunity moments.
- **Rapid Ethnography:** Conduct two– to three–day field immersions that add cultural insight without the burden of full ethnography.
- **Stakeholder Mapping:** Systematically position actors by power and interest to reveal allies, blockers, and engagement priorities.
- **Structured Ideation:** Enhance collaboration in brainstorming by employing techniques such as the Six Thinking Hats or the Walt Disney Method.
- **Surveys / Questionnaires:** Validate qualitative findings quantitatively and segment user groups at scale.
- **Trend & Literature Scan:** Inventory external technological, regulatory, and market forces that may enable or constrain the ecosystem.
- **User & Stakeholder Interviews:** Run semi-structured “probe-why” conversations that expose motivations, pain points, and success criteria.

- **Value-Network Analysis:** Map how data, money, and reputation circulate; detect bottlenecks and leverage points.

6.2.2 Define

Following the divergent scan of *Analyze*, the *Define* step constitutes the methodology's first explicit *convergence* step. Its objective is to distill the dispersed findings into a compact, non-overlapping set of CORE VALUES that can serve as a design compass.

The define step builds on the *Purpose*, *Vision*, *Mission*, *Values*, Strategy, Goals, and Tactics framework, which is commonly used to define and guide the direction of an organization, or in this case, the Digital Ecosystem:

Purpose frames the existential "why", *Vision* articulates the desired future state, *Mission* clarifies the ecosystem's enduring role, and *Values* translate these elements into succinct principles that guide design trade-offs. Lower-tier constructs—goals, strategies, tactics—are deliberately excluded here because they are too short-term or context-specific to serve as stable alignment devices throughout the ecosystem's life cycle.

By negotiating these four cascading statements, stakeholders address the overarching question: *"Why should we collaborate?"*. Consequently, we consider all of these four concepts when defining CORE VALUES. Figure 6.6 illustrates the purpose, task, and application of the *Define* step.

6.2.2.1 Purpose

Converge on* CORE VALUES *that articulate why the Digital Ecosystem exists and what principles will govern its design.

Where *Analyze* sought visibility, *Define* seeks alignment. By translating dispersed insights into four interlocking statements—Purpose, Vision, Mission,

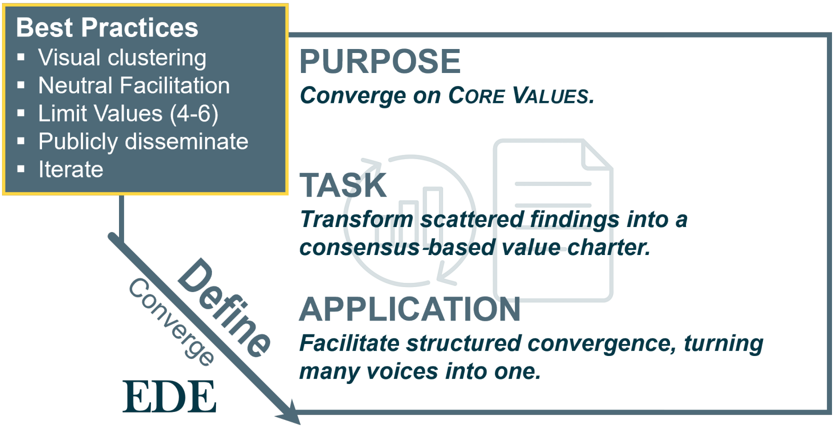


Figure 6.6: Step 2 - Define: Purpose, Task, and Application.

and Values—we establish a stable North Star for all stakeholders to reference during trade-offs.

This cascade addresses the fundamental question, "Why should we collaborate?" While maintaining sufficient abstraction to transcend short-term objectives or tactics, the resulting "value charter" serves as the definitive reference for creating and synthesizing CORE SERVICES.

6.2.2.2 Task

Transform scattered findings into a consensus-based value charter that captures Purpose, Vision, Mission, and Values.

The value charter consolidates qualitative insights into an actionable alignment artifact that serves as a reference for all future design activities. The overall task of the *Define* step can be broken down into four components:

- **Cluster insights:** Group themes from *Analyze* around shared aspirations, constraints, and motives.

- **Draft the cascade:** Formulate a one-sentence *Purpose*, a vivid future-state *Vision*, a durable role-orienting *Mission*, and 4-6 crisp *Values*.
- **Facilitate consensus:** Run cross-stakeholder workshops to negotiate wording and resolve trade-offs.
- **Publish CORE VALUES:** Communicate the agreed upon charter loud and clear to all stakeholders.

6.2.2.3 Application

Facilitate structured convergence, turning many voices into one coherent value charter.

Unlike the deliberately divergent *Analyze* step, *Define* operates in a *converging* mode. We begin by visualizing diverse insights using affinity diagrams, insight clusters, or thematic maps to identify commonalities. Next, we guide stakeholders through a sequence of co-creation moves that progressively reduce variation while allowing for emergence. Negotiate and iterate by facilitating discussions and employing lightweight voting to merge duplicates, resolve conflicts, and refine phrasing. Capture the agreed cascade in a single-page, version-controlled document for easy reference and governance. Disseminate the value charter publicly, frequently, and emphatically. Iterate between *Analyze* and *Define* if coherence or completeness is lacking.

Note that the value charter does not need to only specify the purpose, vision, mission, and values but may also define other tangible ecosystem considerations, such as CORE PARTICIPANTS, or use cases. Furthermore, the vision aspect of the charter should be examined across different time horizons. A short-term vision may differ significantly from the desired long-term future. This value charter serves as a stable benchmark when new stakeholders join or when priorities shift.

Best Practices

- **Start with visual clustering** to make areas of agreement and conflict immediately visible.
- **Use neutral facilitation** so no single stakeholder dominates the wording of purpose, vision, mission or values.
- **Limit values to 4–6 items** to keep the charter memorable and avoid principle overload.
- **Publicly disseminate** and version-control the charter.
- **Iterate** between *Analyze* and *Define* if coherence or completeness is missing.

Suggested Methods

- **Affinity Diagramming:** Cluster raw observations into thematic groups, exposing patterns that feed directly into Purpose-Vision-Mission-Values wording.
- **CATWOE Analysis:** Frame the problem through six stakeholder lenses (Customers, Actors, Transformation, World-view, Owner, Environment) to surface hidden assumptions the charter must reconcile.
- **How-Might-We Questions:** Convert key insights into open, opportunity-oriented challenges that keep the emerging value set actionable.
- **Personas:** Personify archetypal user segments to test whether proposed statements resonate with real stakeholder motivations.
- **Purpose Canvas:** Structured tool to collaboratively define the ecosystem's overarching purpose and long-term vision.
- **Journey Mapping:** Visualize end-to-end stakeholder experiences to reveal value gaps the charter should explicitly address.

- **Quality Function Deployment (QFD):** Link prioritized stakeholder needs to measurable design criteria, anchoring each value in concrete specifications.
- **Root Cause Analysis (5 Whys):** Probe persistent tensions until foundational causes emerge, ensuring the charter tackles root issues rather than symptoms.
- **Value Proposition Design:** Systematic development of stakeholder-specific value propositions that reflect CORE VALUES.

6.2.3 Create

The *Create* step marks the methodology's return to divergence. Building on the value charter, we treat ecosystem design as a *search process* in the Simonian sense—systematically exploring a solution space to locate satisfying concepts rather than a single optimal answer (Simon 1996). Corresponding to the *Develop* stage in Design Thinking, this step promotes breadth: multiple CORE-SERVICE ideas are generated, sketched, and rapidly prototyped, followed by iterative refinement. The three iterations documented in Chapter 5 illustrate how successive cycles of ideation and low-fidelity prototyping resulted in increasingly concrete ecosystem solutions for each use case. Figure 6.7 illustrates the purpose, task, and application of the *Create* step.

6.2.3.1 Purpose

Conceptualize* CORE SERVICES *of the Digital Ecosystem.

Where *Define* aligned intent, *Create* ignites invention. Its objective is to transition from "what must be achieved" to "how it could be achieved" by generating and refining multiple CORE SERVICE ideas. This phase intentionally expands the design space—encouraging bold, even speculative notions—before converging on a select few prototypes or architectural

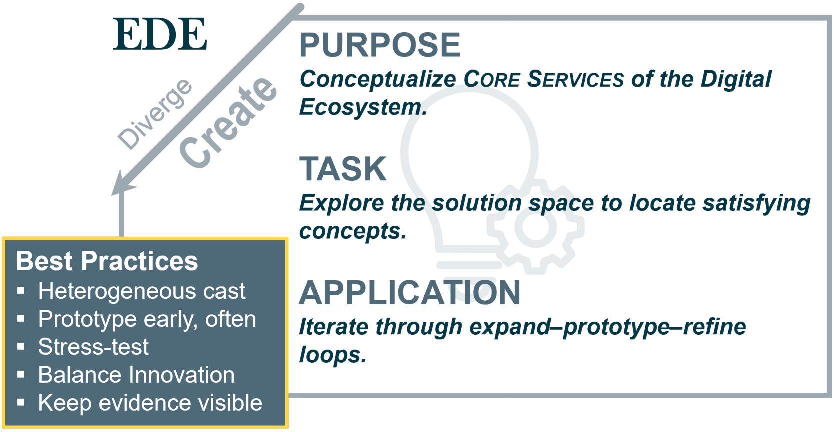

EDE
Diverge
Create
PURPOSE
Conceptualize CORE SERVICES of the Digital Ecosystem.
TASK
Explore the solution space to locate satisfying concepts.
APPLICATION
Iterate through expand–prototype–refine loops.
Best Practices
▪ Heterogeneous cast
▪ Prototype early, often
▪ Stress-test
▪ Balance Innovation
▪ Keep evidence visible

Figure 6.7: Step 3 - Create: Purpose, Task, and Application.

sketches that clearly meet the constraints and opportunities identified in *Analyze* and *Define*. By its conclusion, at least one concept is deemed viable enough to enter formal synthesis.

6.2.3.2 Task

Explore the solution space to locate satisfying concepts.

The responsibility involves transitioning from CORE VALUES and abstract intent to tangible designs through an iterative process of ideation, prototyping, and evaluation. Specifically, the work entails:

- **Generate concepts:** Run divergent ideation sessions (brain-writing, SCAMPER, morphological charts) to identify ecosystem functionalities and create a wide set of CORE SERVICE candidates.
- **Visualize early:** Sketch architectures, storyboards, or process models that make ideas concrete enough for critique.

- **Prototype & test:** Build low-fidelity or Wizard-of-Oz prototypes; collect rapid feedback from stakeholders and potential users.
- **Filter & refine:** Merge complementary ideas, drop weak ones, and enhance promising concepts.
- **Document rationale:** Capture why each design decision was taken, linking back to user insights, domain constraints, and success metrics.

6.2.3.3 Application

Iterate through expand–prototype–refine loops until one concept stands out.

Create operates within a controlled divergence framework: each loop begins with broad ideation, translating ideas into rough artifacts (sketches, paper prototypes, service blueprints), and subsequently narrows the focus through rapid validation against CORE VALUES and feasibility constraints. Typically, two to four such loops are conducted, alternating group creativity, artifact refinement, and lightweight testing, until one or two concepts consistently outperform the others based on desirability, feasibility, and viability metrics. The phase concludes when a shortlist, ideally comprising a single lead concept, has been prototyped and demonstrated to meet the established thresholds of feasibility, desirability, and viability, thus preparing it to enter *Synthesize*.

When designing Digital Ecosystems, it is beneficial to concentrate on a single domain, use case, or function at a time. This approach allows for more manageable design cycles and targeted iterations, facilitating the effective management of complexity and rapid adaptation to specific requirements. However, a previous iteration of the *Create* step must first identify these potential focus areas.

Best Practices

- **Engage a heterogeneous cast:** Invite technical, business, and user-domain experts to maximize idea diversity.
- **Prototype early, prototype often:** Use low-fidelity artifacts so feedback cycles are measured in hours, not weeks.
- **Stress-test with extremes:** Include edge-case scenarios (extreme users, peak loads) to uncover hidden constraints or breakthrough ideas.
- **Balance the innovation triangle:** Weigh desirability (user value), feasibility (technical/operational realism), and viability (economic sustainability) in every down-selection round.
- **Keep evidence visible:** Display user quotes, domain rules, and evaluation scores alongside prototypes so trade-off decisions remain transparent and traceable.

Suggested Methods

- **Brain-writing (Group Ideation):** Fast, rule-based sessions that unleash a broad set of ideas by deferring judgment and encouraging "yes-and" building, while continuously documenting results.
- **Co-Creation Workshops:** Mixed teams of designers, end-users, and stakeholders jointly sketch, storyboard, and critique concepts, boosting buy-in and diversity of insight.
- **Low-Fidelity Prototyping:** Paper sketches, wire-frames, or Wizard-of-Oz facades that make ideas testable within hours, enabling rapid feedback cycles.
- **Morphological Analysis:** A function-by-function matrix that systematically explores all component options and their combinations, ensuring thorough coverage of the design space.

- **SCAMPER:** Seven prompt lenses (Substitute, Combine, Adapt, Modify, Put to other use, Eliminate, Reverse) applied to spark creative modifications or entirely new concepts.
- **TRIZ Inventive Principles:** Leverage a curated set of contradiction-breaking patterns from global patents to inspire technically novel solutions.

6.2.4 Synthesize

The *Synthesize* step is the convergent finale of the EDE methodology. In this step, the various artifacts produced during *Create*—including prototypes, service blueprints, and feedback logs—are consolidated into a single, formalized, and validated functional architecture of the Digital Ecosystem: the Synthesis Model. The focus shifts from creativity to coherence as the CORE SERVICES are iteratively refined and validated. The final design is communicated to stakeholders through the layered presentation of the Synthesis Model. An exemplary implementation of the application of the *Synthesize* step is illustrated in Section 6.5.

Figure 6.8 illustrates the purpose, task, and application of the *Synthesize* step.

6.2.4.1 Purpose

Synthesize the Functional Architecture of the Digital Ecosystem.

Where *Create* provided candidate solutions, *Synthesize* ensures viability. This step confirms that the preferred design meets all practical requirements—technical, organizational, and economic—while integrating individual CORE SERVICES into a cohesive whole. Different artifacts are synthesized into a coherent and formalized Synthesis Model, which facilitates

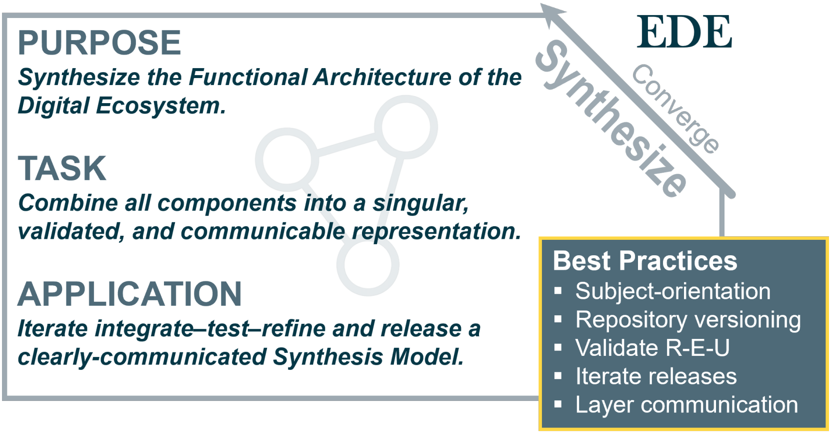

PURPOSE
Synthesize the Functional Architecture of the Digital Ecosystem.
TASK
Combine all components into a singular, validated, and communicable representation.
APPLICATION
Iterate integrate–test–refine and release a clearly-communicated Synthesis Model.
EDE
Synthesize
Converge
Best Practices
▪ Subject-orientation
▪ Repository versioning
▪ Validate R-E-U
▪ Iterate releases
▪ Layer communication

Figure 6.8: Step 4 - Synthesize: Purpose, Task, and Application.

communication, enabling CORE USERS, CONTRIBUTORS, and ORCHESTRATORS to implement the ecosystem with confidence.

6.2.4.2 Task

Combine all components into a singular, validated, and communicable representation: the Synthesis Model.

The work focuses on three interconnected streams:

- **Integration:** Merge the concepts from the *Create* step into one coherent specification of the Synthesis Model.
- **Validation:** Iteratively test and refine relevance, enrichment, and usability of the Synthesis Model.
- **Communication:** Effectively communicate the Digital Ecosystem design to all stakeholders by using the layered presentation of the Synthesis Model.

6.2.4.3 Application

Iterate the Digital Ecosystem design through integrate–test–refine cycles and release a clearly-communicated Synthesis Model.

Integration, validation, and communication function as interleaved micro-loops. Each loop extracts the most recent artifacts from *Create* into a unified source-of-truth repository for the Synthesis Model. The modeling employs a strictly subject-oriented approach, supported by pair modeling. Synthesis can follow either a distinct top-down or bottom-up structure; however, a middle-out rhythm is predominantly utilized. The consolidated model undergoes relevance, enrichment, and usability checks, followed by appropriate refinements, and is subsequently republished as an updated version. The cadence is scalable—from a narrow, domain-specific slice to a comprehensive ecosystem release—but consistently produces a layered, human-readable package: diagrams and markdown narratives combined in a machine-readable schema that can be interpreted and acted upon by ORCHESTRATORS, CONTRIBUTORS, and CORE USERS.

The resulting Synthesis Model is publicly accessible and effectively communicated throughout the Digital Ecosystem. Consequently, ORCHESTRATORS and CORE PARTICIPANTS are consistently informed about the current CORE SERVICES architecture and any upcoming changes.

Best Practices

- **Embrace subject-orientation:** Express every element of the Synthesis Model in clear *Subject–Object–Verb* form. Using the same grammatical backbone in all layers and diagrams prevents role/function mix-ups and keeps cross-domain mappings consistent.
- **Maintain a single version-controlled repository:** Store all diagrams, schemas, and markdown narratives in one canonical repository; tag

releases (e. g., semantic versioning) so every stakeholder can locate the exact model state under discussion.

- **Validate on three axes:** Check *relevance* (still solves the right problem), *enrichment* (adds meaningful detail without bloat), and *usability* (target roles can navigate and apply it) before advancing a model version.
- **Iterate in releasable slices:** When the ecosystem is large, integrate and publish per use case or domain; short cycles keep feedback focused and reduce merge complexity.
- **Layer the communication:** Guide each audience to the domain of interest and right depth of information from the Synthesis Model (SFIB-Layer) so both technical and managerial stakeholders can understand, adopt, and extend the ecosystem design.

Suggested Methods

- **A/B Testing:** Run live experiments comparing design variants to gather quantitative evidence of which configuration best meets the success metrics.
- **Heuristic Evaluation:** Invite usability experts to inspect the model's interface layers against established principles, surfacing problems early and cheaply.
- **Pair Modeling:** Seat a domain expert and a modeling specialist side-by-side to merge artifacts, validate semantics in real time, and capture tacit knowledge without rework.
- **Pilot Testing (Field Trial):** Deploy the integrated solution at limited scale to verify socio-technical fit and expose latent defects before full roll out.

- **Scenario Analysis:** Stress-test the Synthesis Model against alternative future contexts (market shifts, regulatory changes) to ensure robustness and reveal contingency needs.
- **SWOT Analysis:** Assess the strengths, weaknesses, opportunities, and threats inherent in the current model version, guiding focused improvements.
- **Usability Testing:** Observe representative users performing real tasks with the model's interactive views to uncover effectiveness, efficiency, and satisfaction issues.
- **User Acceptance Testing (UAT):** Have end-users execute predefined acceptance scenarios to confirm the final model meets operational and business criteria.

6.2.5 Conclusion

The EDE Double Diamond Approach is inherently iterative. It encourages micro-iterations within individual steps, iterations within the CORE VALUES and CORE SERVICES cycles, and planned iterations to re-evaluate CORE VALUES and CORE SERVICES, while also allowing for the introduction of entirely new ideas.

These steps are highly collaborative and participatory, utilizing qualitative methods such as workshops, interviews, and facilitated modeling sessions to ensure stakeholder inclusion and alignment. By the end of the CORE SERVICES cycle, the EDE methodology generates a comprehensive, stakeholder-aligned, and machine-readable model of the Digital Ecosystem, which serves as the foundation for technical implementation. This Synthesis Model is the guiding framework for the realization of the Digital Ecosystem SOLUTIONS.

6.3 Layers of the EDE Synthesis Model

The EDE Synthesis Model organizes the presentation of Digital Ecosystems into four interconnected and semantically distinct layers: Structure, Function, Interaction, and Behavior (SFIB). This layered approach promotes modularity, separation of concerns, and traceability across different levels of abstraction—from ecosystem-wide architecture to the individual behaviors of participating entities. Figure 6.9 presents a conceptual overview of these layers and their interrelations.

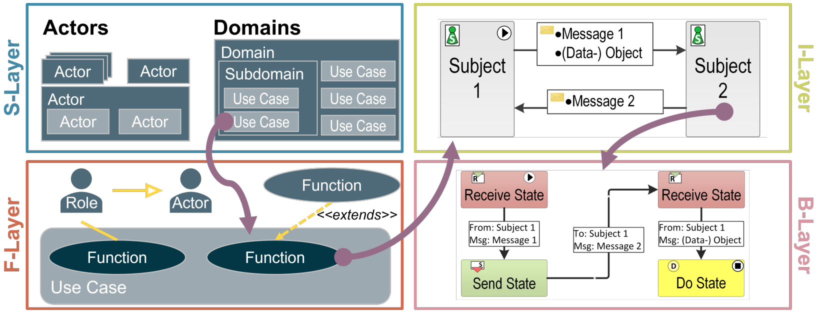


Figure 6.9: Overview of the SFIB layers in the EDE Synthesis Model

Each layer addresses a specific modeling question and focuses on a dedicated aspect of the overall system logic:

- **S-Layer (Structure) – Who is involved?** Defines the structural composition of the digital ecosystem. The primary modeling elements are *Actors*, *Domains*, and *Use Cases*. This architectural view includes only hierarchical relationships of model elements.

- **F-Layer (Function) – Which functions are realized?** Connects the architectural elements of the S-Layer to functional perspectives. For each Use Case, relevant *Functions* are identified, and the *Roles* assumed by Actors are defined. Roles are active entities specific to the Use Case level and are conceptually distinct from Actors.

- **I-Layer (Interaction) – What is exchanged?** Models the exchange of information between *Subjects* within the scope of a Function. These interactions occur within a defined process context and are represented as message flows, abstracting from technical implementation details while maintaining semantic clarity.
- **B-Layer (Behavior) – How does a Subject act?** Specifies the internal logic of a single Subject, including message handling, decisions, and autonomous actions. This view reflects the localized and often partial understanding of a Subject in the context of a larger distributed system.

The four layers are not isolated; rather, they are interconnected through an explicit referencing structure. Elements in each layer are linked to their counterparts in adjacent layers via a directed graph:

- Each Use Case from the S-Layer is linked to its UCD on the F-Layer.
- Each Function from the F-Layer is connected to an SID on the I-Layer.
- Each (fully specified) Subject in the I-Layer owns an SBD on the B-Layer.

This reference-based architecture enables a consistent and clear mapping from high-level ecosystem design to executable subject behavior. The resulting model is modular, scalable, and supports both top-down analysis and bottom-up synthesis of complex socio-technical systems.

6.3.1 Structure Layer

The Structure Layer (S-Layer) delineates the static architecture of a Digital Ecosystem. It serves as the conceptual foundation of the Synthesis Model by identifying and organizing the participating Actors, along with the respective ecosystem Domains and Use Cases. Positioned at the apex of the SFIB hierarchy, the S-Layer addresses the question *"Who is involved?"* and provides structural anchors for the subsequent dynamic and functional layers.

In the general solution of the Synthesis Model, the S-Layer builds upon the principles of SysML BDDs, specifically tailored to the requirements of socio-technical Digital Ecosystems. Instead of modeling technical system components, it presents elements derived from the ecosystem decomposition framework. Each of these elements is distinctly categorized and hierarchically structured to improve clarity and interpretability. The diagrams in the S-Layer fully conform to BDDs but restrict the expressions to enhance understanding and reinforce a subject-oriented approach. The expressibility of the S-Layer is illustrated in Figure 6.10.

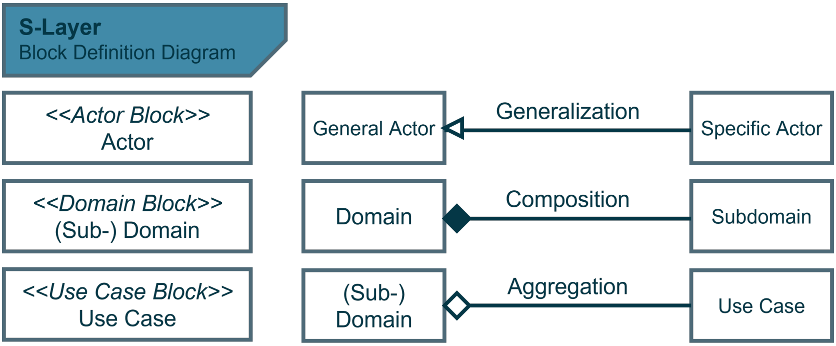


Figure 6.10: EDE S-Layer: Modeling elements.

Table 6.3 provides a comparison between the S-Layer and similar diagram types from SysML and UML, highlighting its domain-specific adaptations and constraints.

The S-Layer adopts the BDD notation but refines it to a deliberately concise, subject-oriented vocabulary, ensuring that every element and relationship is immediately interpretable by ecosystem stakeholders. Only three block categories are permitted: *Actor Blocks* represent the active entities that perform behavior within the ecosystem, while *Domain Blocks* and *Use Case Blocks*

Table 6.3: Comparison of S-Layer to SysML BDD and UML Class Diagram Features.

Feature	EDE (S-Layer)	SysML (BDD)	UML (Class Diagram)
Primary Use	Ecosystem design	Systems Engineering	Software Development
Scope	Socio-technical systems	Multidisciplinary systems (hardware, software, etc.)	Software and software-based systems
Primary Modeling Element	Actor Block, Domain Block, Use Case Block	Block	Class
Structural Relationships	Generalization, Composition, Aggregation	Association, Aggregation, Composition, Generalization, Dependency, Realization	Association, Aggregation, Composition, Generalization, Dependency, Realization
Additional / Specialized Relationships	–	Allocation, Item Flow, Binding Connector, «satisfy», «verify», «deriveReqt»	Template Binding, Package Import/Merge, Qualified Association

depict the passive structural context as identified by the EDE decomposition framework. A Domain may contain one or more Sub-domains; this exclusive whole-part relationship is expressed with *composition*. In contrast, Use Cases are intentionally kept independent of the Domains in which they may execute; their non-exclusive association is therefore modeled with *aggregation*. Beyond these two containment relationships, the palette is restricted to single-inheritance *generalization* hierarchies; all ambiguous or weak dependencies (e. g., «uses», «realizes», item flows, allocations) are omitted. By prohibiting multiple inheritance and low-semantics links, the S-Layer diagram remains compact and unambiguous while still capturing the complete static topology of the Digital Ecosystem.

The S-Layer serves both as a descriptive tool and as a stabilizing structure for the entire modeling process. By clearly delineating the participating Actors, Domains, and Use Cases, it facilitates the consistent development of functional, interactional, and behavioral views across the SFIB layers.

6.3.2 Function Layer

The Function Layer (F-Layer) delineates the functional decomposition of the Digital Ecosystem. Each diagram reflects the context of a specific Use Case. It defines the Use Case along with its associated Functions, Actors, and Roles. Positioned between the structural and processual layers, the F-Layer serves as a crucial link between static system components and their behaviors within concrete processes. It addresses the question, *"Which functions are realized?"* The expressibility of the F-Layer is illustrated in Figure 6.11.

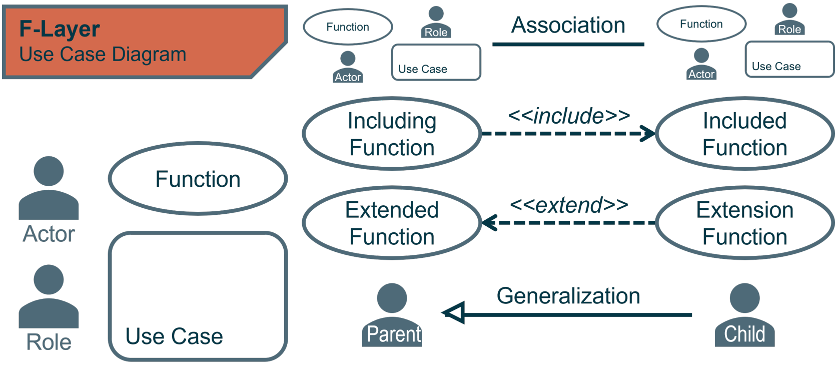


Figure 6.11: EDE F-Layer: Modeling elements.

Table 6.4 provides an overview of the terminology alignment between the EDE F-Layer and the documentation of UML/SysML UCDs.

The F-Layer employs the graphical conventions of a UML/SysML UCD while adapting the terminology to align with the EDE decomposition framework.

Table 6.4: Terminology alignment between the EDE F-Layer and UML/SysML UCDs.

EDE	UML / SysML	Notes
Use Case	System boundary	Defines the functional context being decomposed.
Function	Use Case	Represents an internal capability that realizes the EDE Use Case.
Actor	Actor	Same semantics; an active entity of the system connected to the use case.
Role	Actor «role» stereotype	Active entity with a specific role in the use case.

In this layer, the element typically identified as the system in UML is redefined as the Use Case— a bounded, value-creating scenario within the Digital Ecosystem. The term use case in UML is redefined as a Function, highlighting that it represents a capability that fulfills the EDE Use Case. UML actors can be represented as EDE Actors or Roles, preserving the same core semantics.

In practice, an F-Layer diagram focuses on a single Use Case and illustrates the corresponding Functions, Actors, and Roles. These diagrams provide the structural foundation for the subsequent I- and B-Layers; every PASS model references the Functions defined within this layer. Standard UML dependencies, such as «include» and «extend», are maintained between Functions, providing analysts with a modular and reusable approach to express functional variation without cluttering the diagram with implementation details.

6.3.3 Interaction Layer

The Interaction Layer (I-Layer) specifies the communication between Subjects within a particular Function. It addresses the question *“What is exchanged between whom?”* and elucidates how coordination and information flow occur within the Digital Ecosystem. The means of representation is a PASS SID. This expressibility is illustrated in Figure 6.12.

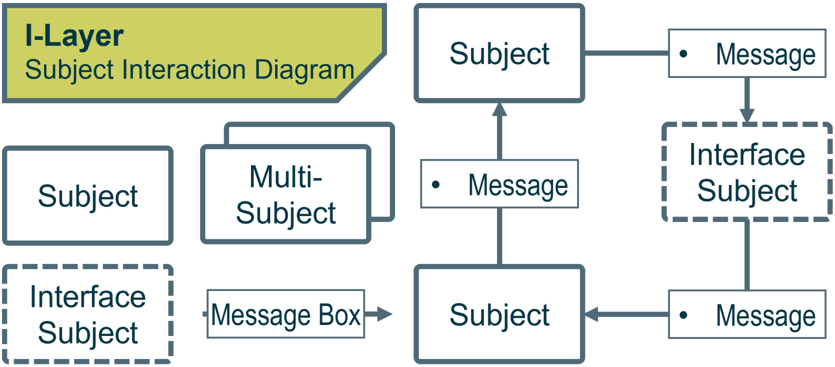


Figure 6.12: EDE I-Layer: Modeling elements.

This layer builds upon the F-Layer and accommodates the inclusion of an SID for each Function. Each SID delineates a set of interacting Subjects, the messages exchanged between them, and the structure of the communication process. The modeling abstraction of the I-Layer intentionally omits internal logic. Given that PASS is fully subject-oriented and optimized for modeling distributed systems, no enhancements to the notation are necessary to better align with the characteristics of Digital Ecosystems. Consequently, the I-Layer supports the use of all model elements specified in the Standard-PASS-Ontology for SIDs.

6.3.4 Behavior Layer

The Behavior Layer (B-Layer) defines the internal logic of a single Subject. It answers the question *"How does a Subject act in a given Function?"* and completes the SFIB model by specifying the behavioral patterns that result in the message exchanges between Subjects depicted on the I-Layer. The notation complies to PASS SBDs and is illustrated in Figure 6.13.

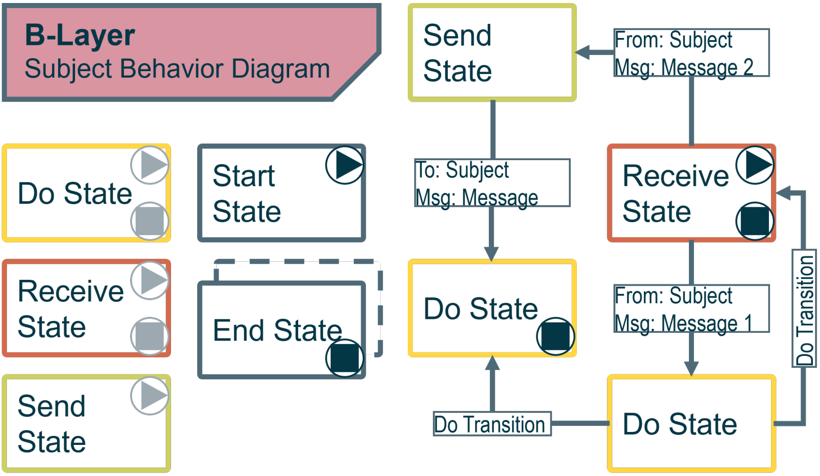


Figure 6.13: EDE B-Layer: Modeling elements.

Behavior in the B-Layer is described using SBDs derived from PASS. Conceptually, an SBD closely resembles a classical finite state machine: it contains exactly one well-defined start state and one or more explicit end states. The control flow consists of states and transitions that can form either linear paths or cyclic loops. In the context of the EDE methodology, each state corresponds to a Task, enabling the direct representation of the Subject's task sequence within the diagram.

Execution begins when the Subject enters the initial state; it then progresses through a series of *Do States*, *Receive States*, and *Send States*. Since each transition is event-driven, the resulting model is, in principle, executable while maintaining an abstract level.

Compared to general-purpose behavior notations, such as UML Activity Diagrams or flowcharts, SBDs explicitly focus on message-based coordination. They maintain a strict separation between a Subject's actions and its communication with other Subjects.

Finally, the B-Layer is integrally connected to the overall layering scheme: each SBD is directly linked to one Subject in an SID on the I-Layer, and every SID is, in turn, associated to a Function in the F-Layer. This chain of references ensures behavioral completeness and traceability throughout the entire model hierarchy.

6.3.5 Conclusion

The SFIB layering concept of the EDE Synthesis Model provides a structured and scalable framework for modeling Digital Ecosystems. By differentiating between structural, functional, interactional, and behavioral aspects, the model improves clarity and modularity in both analysis and design. This is further strengthened by adhering strictly to the paradigm of subject-orientation.

Each layer serves a distinct purpose: the S-Layer defines the static composition of the ecosystem; the F-Layer captures purpose-driven functionality; the I-Layer models communication among distributed roles (subjects); and the B-Layer specifies executable behavior at the subject level. Together, these layers form a coherent modeling framework that enables traceability from abstract actor constellations to concrete system behaviors.

The integration of SysML-inspired diagrams, which emphasize structure and function, with PASS-based diagrams, which focus on interaction and behavior, ensures that the model is both expressive and formally grounded. Furthermore, the layered design supports gradual refinement, model reuse, and compatibility with MBSE approaches.

As Digital Ecosystems increase in complexity, the SFIB model provides a methodological foundation that harmonizes abstraction, specificity, and implementability, making it suitable for both academic research and practical system architecture tasks.

6.4 Representation of the EDE Synthesis Model

The `synthesisModel`[2] JSON schema provides a structured representation of entities within the Synthesis Model. The primary entities include actors, roles, domains, use cases, and functions. Additionally, the relationships connecting these entities can also be defined. To ensure proper versioning and model management, each `synthesisModel` must include a metadata section. This core structure is illustrated in Figure 6.14. The schema adheres to JSON Schema Draft 2020-12[3]. The following subsections detail the specifications of JSON objects permitted in a `synthesisModel` and their properties. All required properties of an object are denoted with an asterisk (*) in the itemizations.

6.4.1 Metadata

Metadata provides essential identifying and descriptive information for the `synthesisModel`. The structure of the `metadata` object is illustrated in Figure 6.14.

The properties of the metadata object are as follows:

- **id*** *(type: string)*: Unique identifier for the entire `synthesisModel`. This identifier must remain consistent across versions of the same model to ensure traceability and reference. The pattern of the `id` is not specified, as it should conform to the ORCHESTRATORS naming conventions.

[2] The notation `synthesisModel` is used from here on out to denote that a representation of a Synthesis Model according to the here described schema definition is meant.

[3] `"schema":"https://json-schema.org/draft/2020-12/schema"`

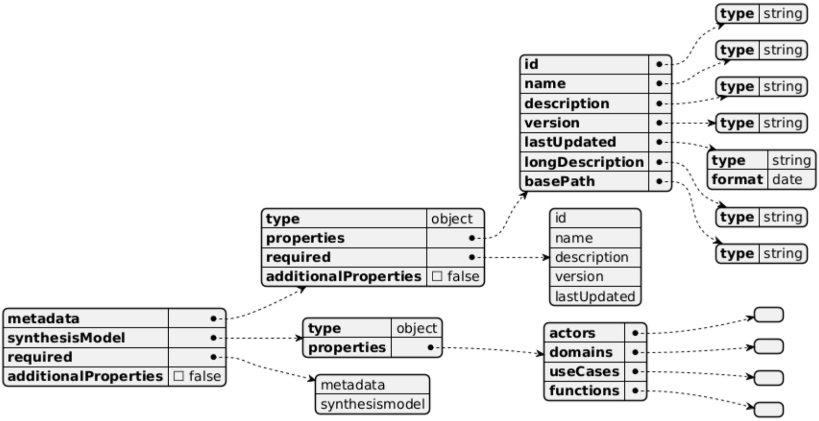


Figure 6.14: JSON Schema structure and metadata of the `synthesisModel`.

- **name*** *(type: string)*: Human-readable name of the `synthesisModel`. This serves as the primary display title for the model in user interfaces and documentation.

- **description*** *(type: string)*: A concise textual description providing an overview of the model's purpose and scope. This single-paragraph description helps users quickly understand the model's intent.

- **version*** *(type: string)*: Version number of the model, typically following semantic versioning (e. g., "1.0.0"). This tracks the evolutionary development of the model over time. The version number is associated with a planned or actual release of the model and does not reflect all commits in the model's repository.

- **lastUpdated*** *(type: string, format: date)*: Date of the most recent update to the model, following the ISO date format (YYYY-MM-DD). This provides temporal context for the model's currency.

- **longDescription** *(type: string)*: Extended textual description that elaborates on the model's context, applications, and relationships to other models. This optional field allows for more detailed explanations than the concise description field.

- **basePath** *(type: string)*: Relative path to the directory containing additional resources associated with the model, such as documentation, images, or supplementary files. This helps maintain organized references to external content.

6.4.2 Core Structure of Synthesis Model Entities

The `synthesisModel` object consists of four primary entity collections: `actors`, `domains`, `useCases`, and `functions`. Each collection is mandatory and is implemented as an array of entity objects. The schema enforces a consistent structure across all entity types while permitting entity-specific attributes.

Common properties shared among these entities include:

- **id*** *(type: string, format: uuid4)*: Universally Unique Identifier (UUID) conforming to the RFC4122 v4 format. This ensures that each entity has a globally unique reference that can be reliably used for cross-entity references and external system integration.

- **shortID** *(type: string, max length: 8)*: Abbreviated identifier used for compact notation and references in diagrams or relationships. While not required, this is particularly useful in visualizations and shorthand references where full UUIDs would be unwieldy. Typically, it involves truncating the UUID to its first eight characters.[4]

[4] The first eight characters of a UUID represent 32 bits, allowing for about 4.3 billion unique combinations. Even though it's only a portion of the full UUID, this is usually more than enough to avoid collisions in practice. The chance of duplicates remains extremely low unless you're generating millions of identifiers, due to the large number of possible values.

- **entityType*** *(type: string, enum: ["actor", "domain", "useCase", "function"])*: Classification that identifies the entity's type within the model. This enforces consistent type identification and enables type-specific validation and processing.
- **name*** *(type: string)*: Full descriptive name of the entity. This serves as the human-readable label for the entity in diagrams, listings, and documentation.
- **shortName** *(type: string, max length: 8)*: Abbreviated name used for display in space-constrained contexts such as diagrams or tables. Unlike shortID, which is used for references, this is meant for human readability in compact visualizations.
- **description*** *(type: string)*: Textual description explaining the entity's purpose, characteristics, and role within the model. This provides context and clarification to aid in understanding the entity's function and significance. This property should not exceed a one-paragraph description.
- **MDrelativePath** *(type: string)*: Relative path to a markdown documentation file providing further elaboration on the entity's details. This facilitates the separation of extensive documentation from the model structure while maintaining linked references.

6.4.3 Actors

Actors represent active entities at the ecosystem level. The full schema for `actor` objects is illustrated in Figure 6.15.

In addition to the common properties listed above, `actor` objects possess specific attributes:

- **isA** *(type: string, format: UUID reference)*: References another Actor entity by its UUID, establishing an inheritance relationship. When

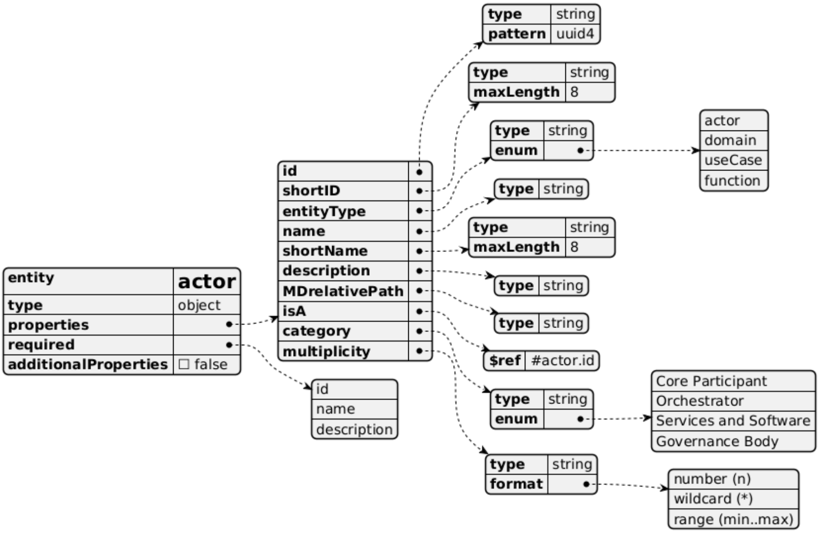


Figure 6.15: JSON Schema for EDE `actor` objects.

specified, the Actor inherits characteristics from the referenced parent Actor, facilitating hierarchical Actor taxonomies and minimizing redundancy in Actor definitions.

- **category** *(type: string, enum: ["Core Participant", "Orchestrator", "Services and Software", "Governance Body"])*: Classifies Actors into predefined functional categories that reflect their roles within the ecosystem:

  – *Core Participant*: Primary actors directly engaged in the ecosystem's main value exchanges - see Section 2.1 for a definition of CORE PARTICIPANT

  – *Orchestrator*: Actors coordinating and facilitating interactions between other Actors - see Section 2.1 for a definition of ORCHESTRATOR

  - *Services and Software*: Service providers or technological components supporting the Digital Ecosystem and its functionality
  - *Governance Body*: Regulatory, oversight, or standards-setting actors establishing rules and policies

- **multiplicity** *(type: string, format: number, wildcard, or range*: Indicates the permitted number of Actor instances that can coexist concurrently within an implemented system. This cardinality constraint is essential for modeling the scalability of ecosystems and the relationships among instances. The multiplicity property supports various formats, each with specific semantic meanings:
  - Number (n): Exactly the specified number of instances must exist (e. g., "1")
  - Wildcard ("*"): Any number of instances can exist, including none
  - Range ("min..max"): Between the minimum and maximum (inclusive) number of instances can exist (e. g., "1..5"). A Wildcard can be used to indicate an unbounded upper range (e. g., "1..*")

6.4.4 Domains

Domains group Use Cases and may contain hierarchical Sub-Domains, as illustrated in Figure 6.16.

In addition to the common properties, `domain` objects include:

- **useCases** *(type: array of strings, format: UUID references)*: Collection of references to `useCase` entities associated with this Domain. This property establishes the Domain's functional scope by linking to the specific Use Cases contained within it, enabling functional grouping and organization.

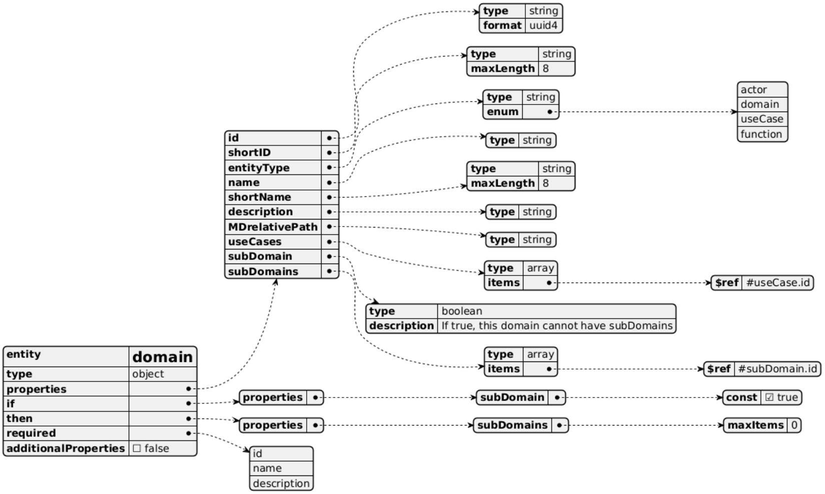


Figure 6.16: JSON Schema for EDE `domain` objects.

- **subDomain** *(type: boolean)*: Flag indicating whether this Domain can be included as a Sub-Domain within another Domain. When set to `true`, the Domain is designated as eligible for inclusion in Domain hierarchies, but cannot itself contain further Sub-Domains (enforcing a single level of nesting).
- **subDomains** *(type: array of strings, format: UUID references)*: References to `domain` entities that are hierarchically contained within this Domain. This enables the representation of nested Domain structures, allowing for the organization of Domains into coherent taxonomies with parent-child relationships.

The schema enforces a crucial constraint: Domains cannot contain further Sub-Domains when marked as `subDomain = true`. This restriction prevents multi-level nesting, thereby avoiding excessive complexity. The constraint is implemented through a conditional validation rule:

Listing 6.1: JSON structure with if/then logic for `subDomain`.

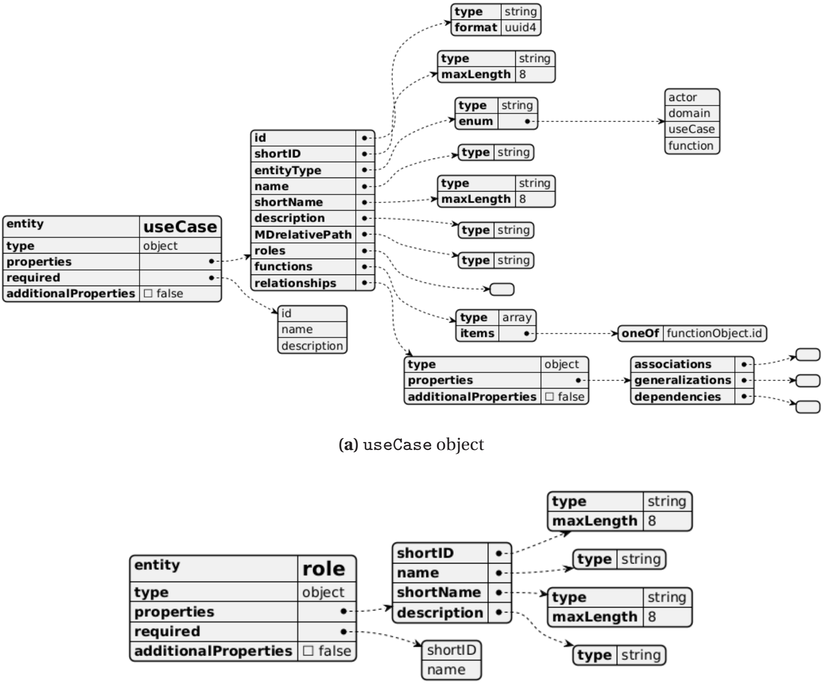
```
{
  "if": {
    "properties": {
      "subDomain": { "const": true }
    }
  },
  "then": {
    "properties": {
      "subDomains": {
        "maxItems": 0
      }
    }
  }
}
```

6.4.5 Use Cases

Use Cases connect all entities within their scope. The structural breakdown of these in the `synthesisModel` is illustrated in Figure 6.17.

Along with the common properties, `useCase` objects exhibit the following specialized attributes:

- **roles** *(type: array of objects)*: Collection of Role definitions specific to this Use Case. Roles represent active entities at the Use Case level. Each `role` object contains:
  - **shortID*** *(type: string, max length: 8)*: A compact identifier for the Role, used in relationship references and diagrams.
  - **name*** *(type: string)*: A descriptive name for the Role that indicates its function or responsibility.



(a) `useCase` object

(b) `role` object

Figure 6.17: JSON Schema for EDE `useCase` objects.

  - **shortName** *(type: string, max length: 8)*: An abbreviated Role name for compact display.

  - **description** *(type: string)*: An explanation of the Role's responsibilities, capabilities, and significance within the Use Case. If the Role requires further elaboration, this can be provided in the markdown file linked to the `useCase` object.

- **functions** *(type: array of strings, format: UUID references)*: References to `function` entities that implement or support this Use Case. This establishes a connection between abstract Use Case descriptions and specific Functions, forming a directed graph that reflects structural decomposition and enhances traceability.

- **relationships** *(type: object)*: A structured definition of the connections among entities within the context of the Use Case. This complex object represents the associations, generalizations, and dependencies that delineate the internal relationships of the Use Case.

6.4.5.1 Relationships

Relationships between entities are explicitly defined within a Use Case. The corresponding `relationship` objects adhere to the structure illustrated in Figure 6.18.

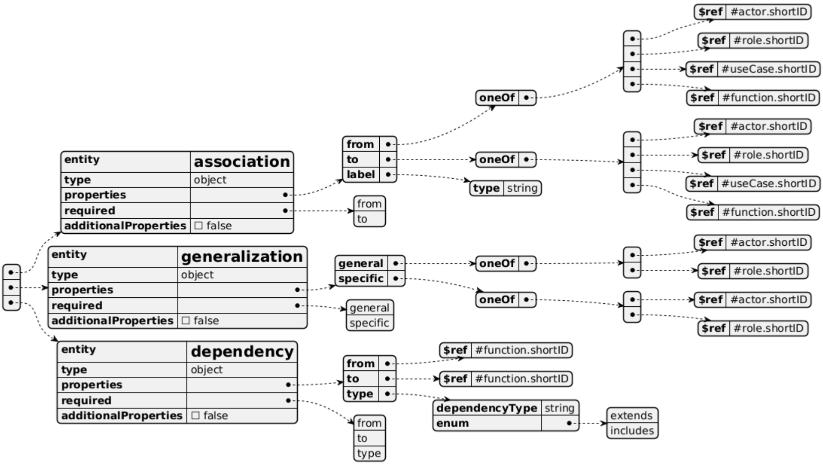


Figure 6.18: JSON Schema for EDE `relationship` objects.

The `synthesisModel` distinguishes among three types of `relationship` objects:

- **associations** *(type: array of objects)*: Connections between entities that represent general relationships or communication pathways. Associations denote weak relationships, modeling interactions such as "communicates with", "interacts with", or "uses", depending on the types of entities (active or passive) linked through the association. Each `association` object contains:

  - **from*** *(type: string, reference to shortID)*: Source entity of the relationship, identifying the initiator or origin point.

  - **to*** *(type: string, reference to shortID)*: Target entity of the relationship, identifying the recipient or destination.

  - **label** *(type: string)*: Optional descriptive text characterizing the nature of the association, such as "manages", "requests", or "provides data to".

- **generalizations** *(type: array of objects)*: Hierarchical relationships that express patterns of specialization and generalization, similar to inheritance in object-oriented design. Generalization is commonly employed to link Roles with Actors and vice versa, as well as to establish hierarchies among Roles. Each `generalization` object contains:

  - **general*** *(type: string, reference to shortID)*: Reference to the more general (parent) Role or Actor, from which characteristics are inherited.

  - **specific*** *(type: string, reference to shortID)*: Reference to the more specific (child) Role or Actor, which inherits and potentially extends the general entity.

- **dependencies** *(type: array of objects)*: Function-to-Function relationships expressing inclusion or extension[5] patterns. These relationships model functional composition and architectural dependencies. Each dependency contains:
  - **from*** *(type: string, reference to shortID)*: Source Function that incorporates or extends another Function.
  - **to*** *(type: string, reference to shortID)*: Target Function that is included or extended.
  - **type*** *(type: string, enum: ["extends", "includes"])*: Nature of the dependency relationship.
    - *extends*: The source Function extends or augments the capabilities of the target Function, adding additional functionality while preserving the original behavior.
    - *includes*: The source Function incorporates the target Function as part of its implementation, making the target Function a constituent component.

The `synthesisModel` schema enforces uniqueness constraints on all relationship arrays to prevent duplicate connections (`"uniqueItems": true`), thus ensuring a clear and non-redundant model of entity interactions.

6.4.6 Functions

Functions encapsulate defined actions or processes within the Digital Ecosystem, and they are allocated to specific Use Cases. Figure 6.19 illustrates the representation of `function` objects.

[5] Applied to Function-to-Function dependencies, `includes` indicates that one function always invokes another as part of its behavior, while `extends` suggests that the additional functionality is invoked only in specific scenarios.

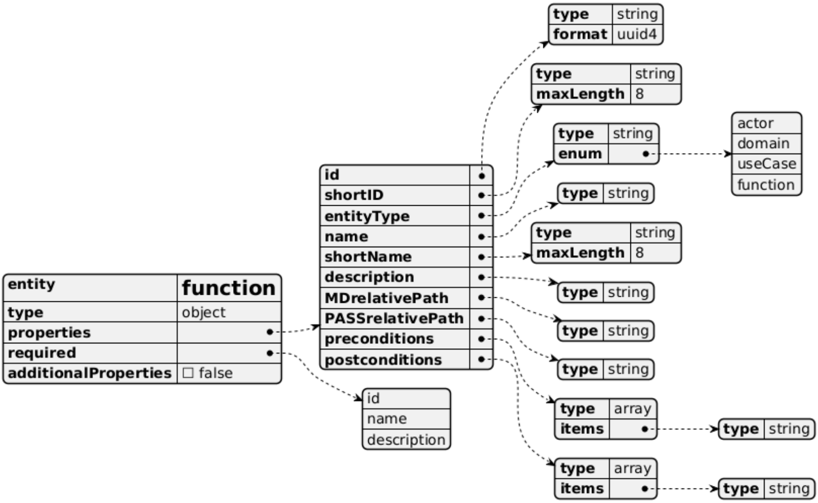


Figure 6.19: JSON Schema for EDE `function` objects.

Beyond the common properties, `function` objects possess the following specific attributes:

- **PASSrelativePath** *(type: string)*: Relative path to the detailed PASS model representation in JSON-LD (see Section 6.4.7. This allows linking the Function to a formal process definition expressed in PASS notation.
- **preconditions** *(type: array of strings)*: Collection of conditions or states that must be satisfied before the Function can be executed. These describe the required input states, available resources, or contextual requirements that must be present. Preconditions serve as both documentation and potential runtime validation checkpoints.
- **postconditions** *(type: array of strings)*: Collection of conditions or states that will be true after the Function's successful execution. These describe the Function's effects, guarantees, or produced output states.

Postconditions document the Function's contract with its consumers and can be used for verification.

6.4.7 JSON-LD for PASS Process Models

PASS models are maintained as independent artifacts, separate from the `synthesisModel`. This separation facilitates modular development while ensuring consistent references across models. Specifically, PASS models are referenced through the `PASSrelativePath` property within the `function` objects, thereby establishing a connection between information for the SF-Layers and IB-Layers.

All PASS models adhere to the Standard-PASS-Ontology [6], which provides a common vocabulary for subject-oriented process modeling.

Although the Standard-PASS-Ontology utilizes RDF/XML as its canonical serialization format[7], this format is not optimal for integration with contemporary web technologies and data pipelines. Therefore, PASS models are converted and stored as JSON-LD[8], which preserves the semantics of the original ontology while enhancing interoperability and usability.

Since both RDF/XML and JSON-LD are serializations of RDF data, the conversion between them is straightforward. Listing 6.2 demonstrates how this conversion can be achieved using a concise Python script.

[6] The Standard-PASS-Ontology is defined and maintained at `https://github.com/I2PM/Standard-PASS-Ontology` and further explained in the PASS Standard Book available at `https://github.com/I2PM/PASS-Standard-Book-Tex-Project`.

[7] RDF/XML is an XML-based serialization format for RDF (Resource Description Framework) data. While it is a W3C standard, its syntax is often verbose and not easily human-readable. See `https://www.w3.org/TR/rdf-syntax-grammar/`.

[8] JSON-LD (JavaScript Object Notation for Linked Data) is a lightweight Linked Data format based on JSON. It allows the representation of RDF data using a compact and developer-friendly syntax, making it easier to integrate semantic data into web technologies. See `https://json-ld.org/`.

Listing 6.2: Convert OWL to JSON-LD using Python and rdflib

```python
from rdflib import Graph

# Load the OWL file
g = Graph()
g.parse("PASS.owl", format="xml")  # OWL is RDF/XML

# Serialize to JSON-LD
g.serialize(destination="PASS.jsonld", format="json-ld")
```

6.4.8 Validation and Constraints

The `synthesisModel` schema implements a comprehensive set of validation rules to ensure data integrity, consistency, and appropriate relationships among all model entities. These constraints can be categorized as follows:

6.4.8.1 Structural Validation

- **Required Properties** *(type: constraints)*: For each entity type, certain properties are essential to ensure complete and valid definitions:
  - All primary entities require `id`, `name`, and `description` to establish identity and provide essential documentation.
  - Relationship objects necessitate both source and target endpoints, as well as the relationship type, to ensure a comprehensive definition of connections.
  - Role objects require `shortID` and `name` for identification and user-friendly labeling.

- **Additional Properties** *(type: constraint: "additionalProperties": false)*: The schema enforces strict property sets by disallowing undefined properties across the model. This ensures consistency and prevents unintentional misspellings of properties or unstructured extensions.

- **Schema Version** *(type: metadata)*: The schema complies with the Draft 2020-12 JSON Schema standard and utilizes advanced validation features, including conditional validation, format assertions, and reference resolution.

6.4.8.2 Type and Format Validation

- **Pattern Constraints** *(type: regex patterns)*: Several properties must conform to specific regular expression patterns to ensure validity:

  - UUIDs must match the pattern in Listing 6.3, enforcing RFC4122 v4 UUID format with specific version and variant bits:

Listing 6.3: UUID Pattern

```
{
  "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-4[0-9a-
      f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}$"
}
```

  - Multiplicity values must match the pattern in Listing 6.4, enabling numeric values, ranges, and wildcard notations (e. g., "*" for "many"):

Listing 6.4: Multiplicity Pattern

```
{
  "pattern": "^(0|1|\\*|[1-9]\\d*)
      (\\.\\.(0|1|\\*|[1-9]\\d*))?$"
}
```

- **String Length Limits** *(type: numeric constraints)*: Properties with explicit length restrictions to ensure compact representation:
  - `shortID` and `shortName` are limited to maximum 8 characters, enforcing their role as brief, display-friendly identifiers
- **Date Format** *(type: format: date)*: The `lastUpdated` property in the `metadata` object must comply with the ISO 8601 date format (YYYY-MM-DD), thereby facilitating chronological tracking and sorting.
- **Enumerated Values** *(type: enum arrays)*: Several properties must be selected from predefined value sets to ensure consistent classification:
  - `entityType` must be one of: "actor", "domain", "useCase", "function", establishing clear type boundaries.
  - Actor `category` must be one of: "Core Participant", "Orchestrator", "Services and Software", "Governance Body", providing a controlled taxonomy for Actor classification.
  - Dependency `type` must be one of: "extends", "includes", limiting the dependency semantics to well-defined relationships.

6.4.8.3 Reference Validation

- **ID References** *(type: JSON schema $ref)*: The schema validates the references between entities to ensure referential integrity:

  - Actor's `isA` must reference a valid Actor `ID`, enforcing proper inheritance chains.
  - Domain's `useCases` must reference valid Use Case `ID`s, ensuring Domains only contain existing Use Cases.
  - Domain's `subDomains` must reference valid Domain `ID`s, maintaining consistent Domain hierarchies.
  - Use case's `functions` must reference valid Function `ID`s, facilitating functional breakdown.

- **Relationship Endpoints** *(type: oneOf/anyOf constraints)*: The schema validates relationship endpoints to ensure coherent connections:
  - Association endpoints (`from`, `to`) must reference existing entity `shortID`s, preventing dangling connections.
  - Generalization endpoints (`general`, `specific`) must reference valid Actor or Role `shortID`s, limiting generalizations to appropriate entity types.
  - Dependency endpoints (`from`, `to`) must reference valid Function `shortID`s, ensuring Function dependencies connect actual Functions.

6.4.8.4 Hierarchy and Relational Constraints

- **Domain Hierarchy Rules** *(type: conditional validation)*: Complex constraints within the `domain` object:
  - Domains marked as `"subDomain":true` cannot have any SubDomains (enforced via `if/then` conditional schema), preventing multi-level nesting.

  - All Domains referenced in `subDomains` array must themselves have `"subDomain":true` set, ensuring consistency in the Domain model.
  - Circular references in Domain hierarchies are prevented through validation rules, avoiding infinite loops in the Domain tree.

- **Uniqueness** *(type: array constraints)*: Validation to prevent duplicate entities and relationships:
  - All relationship arrays (associations, generalizations, dependencies) enforce `uniqueItems:true`, preventing redundant connections.
  - Entity IDs must be unique within their respective collections, ensuring clean entity identification.

The schema validation rules collectively ensure that any `synthesisModel` document conforming to the schema has a consistent structure, maintains referential integrity, and accurately represents the intended Domain model, including its entities and relationships. These technical validations provide a foundation for reliable model exchange, analysis, and visualization.

The `synthesisModel` schema provides a comprehensive and systematically structured foundation for modeling Digital Ecosystems within the EDE methodology, enabling a clear and precise representation of SFIB-Layer information. By formalizing entity properties, relationships, and constraints, this schema facilitates model analysis, supporting both qualitative and quantitative assessments across diverse application scenarios.

6.5 Practical Application of the EDE Methodology

This section provides an example of the practical application of the EDE methodology. It acts as a supplementary guide for practitioners, demonstrating how the theoretical concepts previously described can be operationalized through specific tools, workflows, and technical configurations. The focus here is on establishing an environment that facilitates the CORE SERVICES cycle of the EDE approach.

In the previous sections, the conceptual foundation and structure of the `synthesisModel` within the EDE methodology were introduced and discussed. Building on this, the following presents a comprehensive workflow that facilitates distributed modeling, collaborative development, and automated model integration. This is particularly pertinent in the context of Digital Ecosystems, where multiple ORCHESTRATORS and CORE PARTICIPANTS independently contribute to the evolving ecosystem framework.

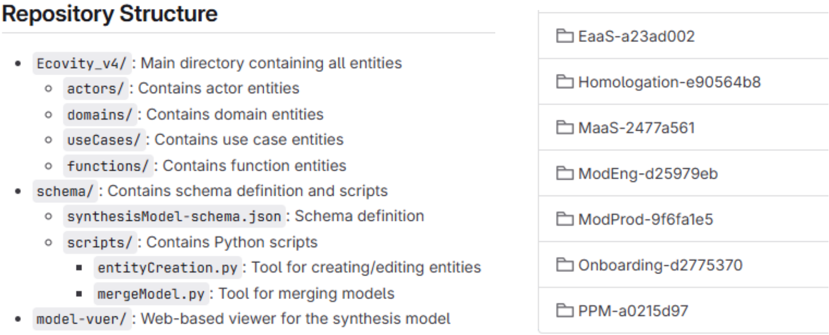
The technical implementation utilizes a GitLab-managed repository for federated entity management, Python tools for entity creation and automated merging, and a specialized web-based Model Viewer application. This approach emphasizes modularity, reusability, and interoperability and provides a transferable blueprint for applying these principles in practice.

6.5.1 Federated Repository Management

The collaborative modeling of Digital Ecosystems within the EDE methodology can be improved through a Git-based repository setup. A central GitLab repository provides the technical foundation for distributed modeling efforts. This configuration allows all ORCHESTRATORS and CORE PARTICIPANTS of a Digital Ecosystem to work concurrently on the model outline while maintaining consistency and traceability of changes.

6.5.1.1 Structure of the Repository

All entities of the `synthesisModel` can be managed independently within the repository. In the root folder, entities are organized into directories by primary types and stored as modular JSON files, each detailing a specific ecosystem entity. The entities are categorized by type: *Actors*, *Domains*, *Use Cases*, and *Functions*. A dedicated Python-based tool (see Section 6.5.2) generates a subdirectory for each of these entities, as illustrated in Figure 6.20.



Repository Structure

- `Ecovity_v4/` : Main directory containing all entities
  - `actors/` : Contains actor entities
  - `domains/` : Contains domain entities
  - `useCases/` : Contains use case entities
  - `functions/` : Contains function entities
- `schema/` : Contains schema definition and scripts
  - `synthesisModel-schema.json` : Schema definition
  - `scripts/` : Contains Python scripts
    - `entityCreation.py` : Tool for creating/editing entities
    - `mergeModel.py` : Tool for merging models
- `model-vuer/` : Web-based viewer for the synthesis model

EaaS-a23ad002
Homologation-e90564b8
MaaS-2477a561
ModEng-d25979eb
ModProd-9f6fa1e5
Onboarding-d2775370
PPM-a0215d97

(a) Repository Structure **(b)** `useCase` Directories

Figure 6.20: EDE Application: Structure of the repository.

A UUID is assigned to each entity upon creation, ensuring conflict-free integration of entities from various locations or contributors into a unified model. Additional properties can be configured using the Python tool or manually within the JSON file. Markdown files can be utilized to enhance the descriptions of entities. PASS process models for Functions can be independently created, exported to JSON-LD, stored in the entity or assets folder, and linked to the entity within the JSON object.

6.5.1.2 Support for Federated and Intersecting Ecosystems

While this example focuses on a single central repository, the approach inherently supports federated collaboration across multiple Digital Ecosystems. The use of UUIDs facilitates the seamless integration of entities from various repositories or ecosystems.

This flexibility allows Digital Ecosystems to intersect and share entities across organizational or domain boundaries, similar to the overlapping regions of a Venn diagram. Entities developed within one ecosystem can be reused, extended, or referenced in another, thereby enhancing interoperability and collaboration across ecosystems without necessitating centralized control.

6.5.2 Creation of the `synthesisModel` Entities

To facilitate the definition of entity descriptions for the `synthesisModel`, a dedicated Python-based GUI tool is provided. This tool enables contributors to create, edit, and relate ecosystem entities in a structured and schema-compliant manner, thereby eliminating the need for manual management of JSON files or directory structures.

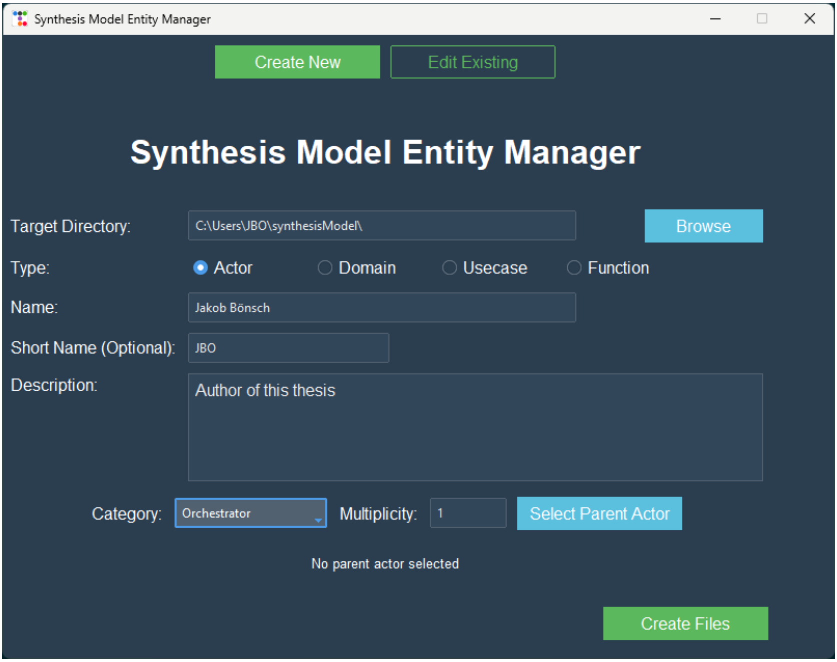
6.5.2.1 Purpose and Integration in the Workflow

The tool is central to the GitLab-based workflow. It enables ORCHESTRATORS and CORE PARTICIPANTS to independently create entities, each uniquely identified by UUIDs. This approach eliminates merge conflicts and facilitates contributions from distributed teams.

Each entity (i. e., *Actor*, *Domain*, *Use Case*, or *Function*) is stored as a JSON file within a directory named after the entity's short name and UUID suffix. Additionally, a Markdown file for human-readable documentation is generated alongside the JSON file.

6.5.2.2 Usage and Features

The graphical tool facilitates the creation and editing of all four primary `synthesisModel` entities. Figure 6.21 depicts the main GUI window.


Synthesis Model Entity Manager
Create New
Edit Existing
Synthesis Model Entity Manager
Browse
Select Parent Actor
Create Files

Target Directory: C:\Users\JBO\synthesisModel\

Type: ● Actor ○ Domain ○ Usecase ○ Function

Name: Jakob Bönsch

Short Name (Optional): JBO

Description: Author of this thesis

Category: Orchestrator Multiplicity: 1

No parent actor selected

Figure 6.21: Main interface of the entity-creation tool with type selection and metadata input.

The tool is intentionally streamlined yet powerful. It offers two distinct working modes, namely *Create New* and *Edit Existing*, allowing contributors to either introduce new entities or update those already under version control without switching applications. When a new entity is created, the program assigns a UUID and embeds that identifier in both the file name and the enclosing directory. This convention virtually eliminates merge conflicts when multiple teams push to the same Git repository. All data entered through the GUI conforms to the formal `synthesisModel` JSON schema. For convenience, operations involving cross-references—such as allocating Use Cases or adding relationships—are managed in dedicated pop-up dialogs. These dialogs present filtered lists of eligible targets and visual summaries of existing relations, facilitating the establishment or revision of inter-entity connections without manually editing JSON.

6.5.2.2.1 Creating a new entity.

1. (Optional) Change the target directory.
2. Choose *Create New* mode and select the desired entity type.
3. Enter a short name and a descriptive text.
4. Fill in type-specific details (e. g., category, multiplicity).
5. Press *Create Files.*

Upon submission, the tool assigns a UUID and generates a JSON file along with a corresponding Markdown file, following the structure outlined below for each entity:

- `<shortname>-<uuid8>/` - `<uuid8>` = first 8 characters of the UUID
  - `<uuid>.json` — structured entity definition
  - `description.md` — human-readable description

These files are located in their respective directories (e. g., `actors/`, `useCases/`) within the repository. This organization ensures compatibility with the merge tools and the GitLab workflow.

6.5.2.2.2 Editing an existing entity.

1. Choose *Edit Existing* mode.
2. Select the entity in the tree view; its metadata is loaded into the form.

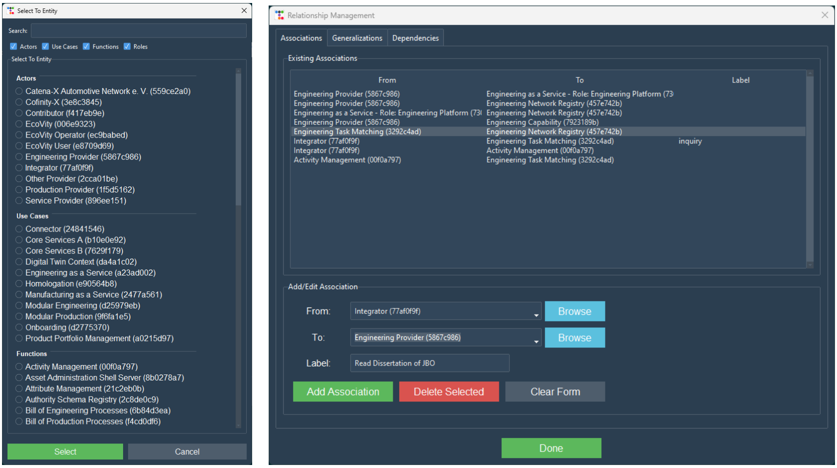
3. Modify any field as required.

4. Click *Update*.

The modified properties of the JSON object are updated while preserving the previous input, particularly the UUID, to ensure compatibility for downstream merging. No changes are made to the Markdown file.

6.5.2.2.3 Managing relationships.

Pressing *Manage Relationships* opens the dialogue displayed in Figure 6.22. Here, the user can establish or modify *associations*, *generalizations* or *dependencies*. Changes are written back to the JSON object.



Select To Entity

Search:

☑ Actors ☑ Use Cases ☑ Functions ☑ Roles

Select To Entity

Actors

○ Catena-X Automotive Network e. V. (559ce2a0)
○ Cofinity-X (3e8c3845)
○ Contributor (f417eb9e)
○ EcoVity (006e9323)
○ EcoVity Operator (ec9babed)
○ EcoVity User (e8709d69)
○ Engineering Provider (5867c986)
○ Integrator (77af0f9f)
○ Other Provider (2cca01be)
○ Production Provider (1f5d5162)
○ Service Provider (896ee151)

Use Cases

○ Connector (24841546)
○ Core Services A (b10e0e92)
○ Core Services B (7629f179)
○ Digital Twin Context (da4a1c02)
○ Engineering as a Service (a23ad002)
○ Homologation (e90564b8)
○ Manufacturing as a Service (2477a561)
○ Modular Engineering (d25979eb)
○ Modular Production (9f6fa1e5)
○ Onboarding (d2775370)
○ Product Portfolio Management (a0215d97)

Functions

○ Activity Management (00f0a797)
○ Asset Administration Shell Server (8b0278a7)
○ Attribute Management (21c2eb0b)
○ Authority Schema Registry (2c8de0c9)
○ Bill of Engineering Processes (6b84d3ea)
○ Bill of Production Processes (f4cd0df6)

(a) Selection of related entities

Relationship Management

Existing Associations

From	To	Label
Engineering Provider (5867c986)	Engineering as a Service - Role: Engineering Platform (73	
Engineering Provider (5867c986)	Engineering Network Registry (457e742b)	
Engineering as a Service - Role: Engineering Platform (73	Engineering Network Registry (457e742b)	
Engineering Provider (5867c986)	Engineering Capability (7923189b)	
Engineering Task Matching (3292c4ad)	Engineering Network Registry (457e742b)	
Integrator (77af0f9f)	Engineering Task Matching (3292c4ad)	inquiry
Integrator (77af0f9f)	Activity Management (00f0a797)	
Activity Management (00f0a797)	Engineering Task Matching (3292c4ad)	

Add/Edit Association

From: Integrator (77af0f9f)

To: Engineering Provider (5867c986)

Label: Read Dissertation of JBO

(b) Definition of associations

Figure 6.22: Relationship management within a Use Case.

6.5.3 Consolidating the `synthesisModel`

After entities have been created and committed to the GitLab repository, the Python script `mergeModel.py` is utilized to consolidate the distributed components of the `synthesisModel` into a single, schema-compliant file. This merged model serves as the foundation for visualization and subsequent processing.

6.5.3.1 Functionality Overview

The merge script performs the following key tasks:

1. Recursively collects all entity JSON files (*Actors*, *Domains*, *Use Cases*, *Functions*).
2. Automatically links Markdown descriptions if available.
3. Assembles a hierarchical `synthesisModel` structure with updated metadata.
4. Validates the resulting model against the provided JSON schema.
5. Copies all source files to the `model-vuer` application directory for visualization.

6.5.3.2 Usage Example

The script is executed from the repository root directory as follows:

Listing 6.5: Running the merge script for the synthesisModel

```
python schema/scripts/mergeModel.py
```

The user is prompted to enter a model name. If no input is provided, a default name is used (e. g., `EcoVity_v4_r0`).

The script generates two primary outputs:

- `model-vuer/public/synthesismodel/<model-name>.json`
  A consolidated `synthesisModel` file for visualization.
- `model-vuer/public/synthesismodel/assets/<model-name>/`
  A directory containing all source files (JSON and Markdown) for the model viewer application.

6.5.3.3 Validation and Quality Assurance

The script validates the generated `synthesisModel` against a defined JSON schema (`schema/synthesisModel-schema.json`) to ensure structural consistency and compliance.

In future work, this validation process can be directly integrated into the GitLab CI pipeline to automatically verify all contributions before merging them into the main branch of a given release.

6.5.4 Model Viewer

The visualization of the `synthesisModel` is achieved through the *model-vuer*, a web application based on Vue 3, designed to offer an interactive and user-friendly interface for exploring Digital Ecosystem models.

6.5.4.1 Core Technology Stack

The application follows a modern frontend architecture and incorporates the following technologies:

- **Vue 3:** Reactive frontend framework using the Composition API.
- **TypeScript:** Provides type safety across the application.
- **Vite:** Fast build tool and development server.
- **Element Plus:** UI component library for consistent interface elements.
- **Pinia:** State management with persistent storage support.
- **Vue Router:** For handling application navigation.
- **Markdown & PlantUML:** For content editing and diagram rendering.
- **webPASS-Viewer:** Web-based viewing of PASS models directly from JSON-LD, built on Vue Flow.
- **Vue Flow:** Customizable Vue 3 component for interactive flowcharts.

6.5.4.2 Integration with the GitLab Workflow

The `synthesisModel` generated by the merge script is stored in the `model-vuer/public/synthesismodel/` directory. Upon deployment, the model-vuer application can select all models in this directory for visualization.

This automated integration facilitates a seamless workflow encompassing entity creation, repository management, model merging, and visualization within a containerized, reproducible environment.

6.5.5 Conclusion

This section demonstrates the practical application of the EDE methodology. The presented toolchain facilitates the distributed and collaborative development of Digital Ecosystem models by integrating structured repository management, automated model merging, and interactive visualization. The described application setup is used like that in IntWertL.

The GitLab-based repository setup allows all Core Participants and Orchestrators of a Digital Ecosystem to independently contribute entities, supported by UUID-based identification and standardized directory structures. The Python tools facilitate the creation, editing, and merging of entities in a schema-compliant manner, thereby ensuring model consistency across distributed contributions.

Finally, the model-vuer application provides an intuitive web-based interface for exploring and analyzing the resulting Synthesis Model. The automated deployment process guarantees that new versions of the model can be integrated and visualized with minimal effort.

Overall, this approach exemplifies the core principles of the EDE methodology—modularity, interoperability, and federated collaboration—and provides a transferable blueprint for modeling and managing complex, distributed Digital Ecosystems.

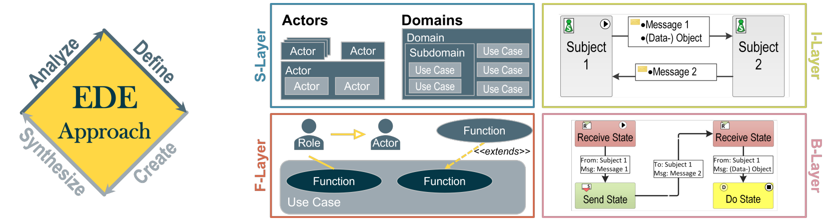
6.6 Summary

The *Engineering Digital Ecosystems (EDE)* methodology serves as a generalized foundation for the model-based and subject-oriented design of Digital Ecosystems. This methodology constitutes the main artifact of this thesis and addresses the two primary research questions:

RQ1 How can the design process of Digital Ecosystems be supported by a *model-based methodology*?

RQ2 What kind of *subject-oriented model* can be used to describe a Digital Ecosystem?

Therefore, the EDE methodology consists of two complementary components, as illustrated in Figure 6.23.



(a) EDE Double Diamond

(b) Synthesis Model

Figure 6.23: Core components of the EDE methodology.

First, it presents a structured methodological approach that guides the design process. This four-step process adapts the Design Council's Double Diamond model and addresses both Core Values and Core Services of the Digital Ecosystem. The EDE Double Diamond serves as the response to RQ1. The second component, the *Synthesis Model*, corresponds to RQ2. It facilitates the subject-oriented and model-based (re-) presentation of Digital Ecosystems. At its core is a layered presentation of Structure, Function, Interaction, and Behavior (SFIB). Completing the EDE methodology is the formal model representation of the `synthesisModel` as a JSON schema. Together, these components address the fundamental challenges of EDE, specifically the inherent complexity, federated structure, and diversity of stakeholders typical of Digital Ecosystems.

To facilitate the implementation of the EDE methodology, this chapter includes best practices, methodological toolboxes, and an example of practical application. The EDE methodology presented here is validated and discussed in the subsequent chapter.

7 Validation and critical reflection

DSR contributions are persuasive only to the extent that they are supported by relevant, rigorous, and useful evidence. Consequently, this chapter consolidates insights gained from the *IntWertL* case study and situates the proposed EDE methodology within the theoretical framework of DSR.

It aims to achieve three complementary objectives:

1. to demonstrate how the research satisfies the seven DSR guidelines articulated by Hevner et al. (2004);
2. to critically reflect on the hypotheses that underpin the EDE methodology; and
3. to delineate the methodological and practical limitations that temper the generalizability of the results and frame a future research agenda.

Therefore, the following section 7.1 maps each guideline to specific decisions and outcomes of the study. The subsequent section 7.2 offers a critical reflection on the six guiding hypotheses, while section 7.3 delineates the substantive and procedural limitations of the EDE methodology. Collectively, the chapter provides a transparent and coherent account of how this thesis both adheres to and expands upon the canon of DSR.

7.1 Validation of the application of the Design Science Research Guidelines

This section discusses the application of all seven DSR guidelines of Hevner et al. (2004) in this research study, thereby demonstrating its contribution as effective DSR. Hevner distills this into two fundamental questions:

> The fundamental questions for design-science research are, "What utility does the new artifact provide?" and "What demonstrates that utility?" Evidence must be presented to address these two questions. That is the essence of design science. Contribution arises from utility. If existing artifacts are adequate, then design-science research that creates a new artifact is unnecessary (it is irrelevant). If the new artifact does not map adequately to the real world (rigor), it cannot provide utility. If the artifact does not solve the problem (search, implementability), it has no utility. If utility is not demonstrated (evaluation), then there is no basis upon which to accept the claims that it provides any contribution (contribution). Further more, if the problem, the artifact, and its utility are not presented in a manner such that the implications for research and practice are clear, then publication in the [Information Systems] literature is not appropriate (communication). (Hevner et al. 2004, p. 91)

Consequently, the EDE methodology is validated against these guidelines to demonstrate its utility.

7.1.1 Problem Relevance

The significance of designing Engineering Digital Ecosystems is clearly outlined in Chapter 3. German federal ministries and the European Union have allocated hundreds of millions of euros in grants for research projects aimed at developing, implementing, and establishing Digital Ecosystems. Most of these funds are matched with equivalent industrial investments.

Nonetheless, the analysis of the academic and industrial research landscape regarding design methodologies for Digital Ecosystems identified four major research gaps. This research study aims to contribute knowledge to the first two of these gaps:

1. Lack of holistic modeling methodology for Digital Ecosystems.
2. Insufficient integration of formalized modeling paradigms into Digital Ecosystem design.

Therefore, this study is deemed relevant as it addresses two primary research questions associated with the identified gaps:

RQ1 How can the design process of Digital Ecosystems be supported by a model-based methodology?

RQ2 What kind of subject-oriented model can be used to describe a Digital Ecosystem?

7.1.2 Research Rigor

Based on Hevner et al. (2004), research rigor comprises three aspects: the application of appropriate methods in conducting research, the utilization of the knowledge base, and the implementation of artifacts in suitable environments.

The first aspect is addressed in Chapter 4. This chapter outlines the rationale for selecting the SDSM approach, the design of the evaluation based on FEDS, and the distillation of Level 1 results into the EDE methodology.

Second, the knowledge base is employed to inform design decisions throughout the iterations of the SDSM, as well as directly for the EDE methodology. The EDE Approach is based on the Design Council's Double Diamond framework and principles of MBSE. While the Synthesis Model employs JSON schema for representation, it bases its presentation on subject-orientation, SysML, and PASS.

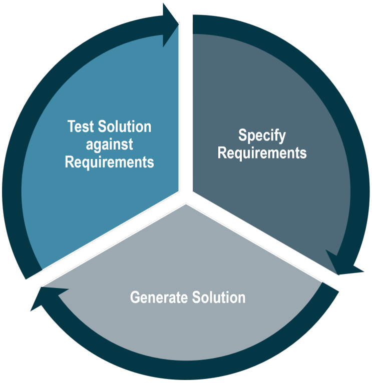
Lastly, the IntWertL project offers an ideal implementation environment. It is sufficiently large to encompass over twenty organizations and a variety of stakeholders from all relevant Engineering Digital Ecosystem domains, while remaining small enough to allow comprehensive observation of all ecosystem design activities.

7.1.3 Design as a Search Process

For the EDE methodology, we seek an appropriate design approach, as well as a satisfactory means of presenting and representing the Synthesis Model. This design process is inherently iterative. Given the extensive solution space for modeling approaches and techniques, we do not seek the optimal solution but rather aim for a "satisficing" (Simon 1996) solution, meaning a solution that is sufficiently effective. The search process employed through the SDSM is illustrated in detail in Chapter 5. In line with the search for a satisficing solution, we focus on demonstrating that the identified solutions are effective rather than explaining why they work. Consequently, the SDSM approach iterates requirement specification, solution generation, and testing, as illustrated in Figure 7.1.

The search process in this work operates on three levels:

- **SDSM Cycle:** Full nine-step SDSM approach. Possibility to define another focal point (e. g., process-centric, or structure-centric).



SDSM Steps:

1. Analyze Specific Problem Domain
2. Define Specific Requirements
3. General Problem and Requirements
4. General Solution
5. Ex Ante Evaluation (General)
6. Design Specific Solution
7. Ex Ante Evaluation (Specific)
8. Implement Specific Solution
9. Ex Post Evaluation

Figure 7.1: Iterative Generate-Test cycle employed in the SDSM search process.

- **Specialization Loop:** The three generate-test loops in one SDSM cycle. Starting from the general solution, through the specific solution, until the specific solution in its application domain.
- **Micro-Iterations**: Refinements of the solutions based on the immediate evaluation feedback.

7.1.4 Design as an Artifact

Hevner et al. (2004, p. 82) do not limit the term artifacts to only include instantiations "but also the constructs, models, and methods applied in the development and use of information systems". Thus, this work presents two types of artifacts: First, the instantiations of Level 1 artifacts in Chapter 5. Second, and more importantly, the EDE methodology, its design principles, methods, and the Synthesis Model, presented in Chapter 6, constitutes an artifact in its own right.

7.1.5 Design Evaluation

Each instantiated design artifact is evaluated meticulously. The corresponding evaluation design is elaborated on in Section 4.2.3. Building on the FEDS of Venable et al. (2016) four steps are taken: First, the goals of the evaluation are established; second, the evaluation strategy is selected; third, evaluation properties are defined; and finally, the individual evaluation episodes are planned. The evaluation goals include relevance, enrichment, usability, and timeliness. As no other artifacts addressing the same problem exist, direct comparison is not feasible. Nonetheless, three primarily formative evaluation steps are conducted during each SDSM cycle. The results of these evaluations are presented throughout Chapter 5.

The EDE methodology is validated by the collective findings from these Level 1 evaluations. However, as this is a nascent approach to designing Digital Ecosystems, further evaluation through future work is necessary. For now, the utility of the EDE methodology is substantiated by its established existence and rigorous development.

7.1.6 Research Contributions

This research offers two distinct contributions to the field:

- **Level 1 | Design artifacts:** Instantiations of the Digital Ecosystem design are true inventions as both the solution maturity, and the application domain maturity are to be considered low. These design artifacts are the Synthesis Models V1-V3 from Chapter 5.
- **Level 2 | Methodology:** The EDE methodology meets the criteria for exaptation as well known solutions are combined in a new way and applied to a new application domain. The corresponding emergent knowledge contributions, i. e. the EDE Double Diamond Approach and the Synthesis Model, are presented in Chapter 6.

The hypotheses discussed subsequently may serve as a basis for developing more advanced knowledge contributions, such as a comprehensive design theory for Digital Ecosystems. However, although some interesting findings support each of these hypotheses, further research is necessary to integrate this and other knowledge into a cohesive Level 3 knowledge contribution.

7.1.7 Research Communication

DSR necessitates effective communication of research to both technical and managerial audiences. In the context of Digital Ecosystems, this extends to include ORCHESTRATORS, CONTRIBUTORS, and CORE USERS. Consequently, this thesis aims to engage a diverse audience, including but not limited to: researchers in Digital Ecosystems, model-based design, subject-orientation, and DSR, as well as industry representatives from the engineering and software development sectors.

The need for effective communication is emphasized through a clear structure throughout the thesis. In particular, the EDE methodology in Chapter 6 is presented to facilitate stakeholders in easily locating relevant information. Each step of the EDE Double Diamond is systematically outlined, detailing its purpose, tasks, and applications. Furthermore, the Synthesis Model is first introduced in conceptual terms, followed by its notation, and is then specifically addressed for technical audiences in its representational form.

Chapter 5 addresses all practitioners seeking a deeper understanding of the iterative artifact creation and search process. It provides valuable insights into how instantiations of the Synthesis Model can manifest and address various concerns associated with the specific needs of CORE PARTICIPANTS.

Chapters 3, 4 and 7 are targeted towards the research community, while Chapters 1, 2 and 8 offer a shared framework for all readers.

7.2 Critical Reflection

Based on the current state of research and applications discussed in Chapter 3, six hypotheses (H1 - H6) are proposed: the first three focus on **RQ1** (model-based methodology), while the remaining three concentrate on **RQ2** (subject-oriented model). Each of these hypotheses guided the design of the EDE methodology.

H1 **CORE VALUES and CORE SERVICES need to be considered in Digital Ecosystem design.**

H2 **Digital Ecosystems are complex enough that a model-based approach only makes sense.**

H3 **A Digital Ecosystem is a system of systems and CORE PARTICIPANTS perform a complex choreography.**

H4 **Subject-Orientation is well suited for socio-technical Digital Ecosystems.**

H5 **A Digital Ecosystem model requires multiple perspectives.**

H6 **Digital Ecosystems evolve over time.**

Subsequently, each of these hypotheses will be discussed in terms of how it is motivated by the state of the art, its application in the iterative design of Synthesis Model V1-V3, and its reflection in the EDE methodology.

7.2.1 Core Values and Core Services need to be considered in Digital Ecosystem design.

This hypothesis aims to prioritize the inclusion of Core Values rather than solely relying on Core Services. Design invariably starts with an exploration of the problem space, which, in the context of Digital Ecosystems, involves understanding the Core Values. Effective engineering of Core Services can only begin when these elements are aligned among stakeholders. Initiating with Core Values is expected to shorten subsequent consensus cycles, reduce design conflicts, and serve as a stable reference for iterative improvements in functionality.

This need is recognized in the first two steps of the SDSM during each iteration. In particular, I1 and I3, through their extensive vision workshops, place significant emphasis on formulating Core Values for the IntWertL ecosystem. Moreover, the design and implementation of the specific solution are evaluated for alignment with the previously defined Core Values.

The inclusion of both Core Values and Core Services is evident in the EDE methodology, as the EDE Double Diamond Approach comprises the Core Values cycle and the Core Services cycle, as illustrated in Figure 7.2.

7.2.2 Digital Ecosystems are complex enough that a model-based approach only makes sense.

The analysis of the state of the art demonstrates that informal "box-and-line" notations are prevalent tools for modeling Digital Ecosystems. Moreover, there is a recognized need for models, possibly even multi-perspective models, to effectively describe Digital Ecosystems. Formalizing the employed modeling techniques should introduce minimal overhead while significantly enhancing cross-stakeholder understanding and reducing rework. The hypothesis posits that measurable improvements in clarity and defect avoidance will occur when compared to informal box-and-line sketches.

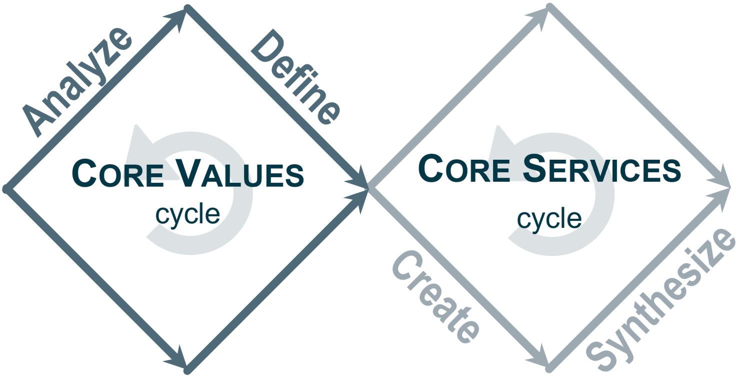


Figure 7.2: Core Values cycle and Core Services cycle of the EDE methodology.

In accordance with this hypothesis, all three iterations utilize formal modeling languages to support the model-based approach. By constraining modeling elements and integrating informal annotations, these models achieve universal comprehensibility. However, as demonstrated in I1 with the process explorations, these modeling tools can not only aid in articulating the design of the Digital Ecosystem but also in analyzing its problem space. The ex post evaluations revealed, that stakeholders created their own models for perspectives that where not included in the current version of the Synthesis Model. However, if their models were not properly connected to the Synthesis Model framework, this led to inconsistencies and contradictory models from different participants. Additionally, some of these models were so informal that multiple lengthy meetings were necessary to explain the content and notation, rendering the model basically useless for communication.

By the third iteration at the latest, it becomes evident that Digital Ecosystems are sufficiently complex to necessitate standardized modeling. In this context, eleven use cases involved modeling fifty-nine functions, not to mention the numerous related subjects and tasks that would arise if all process models were fully elaborated.

Consequently, the EDE methodology employs a model-based approach, introducing the Synthesis Model as an outcome of the CORE SERVICES cycle. Both the Creation and Synthesis steps require model-based work. However, in the earlier Analyze and Define stages, employing incomplete models or fragments is advised based on best practices and suggested methods. This model-based approach helps manage complexity by providing a common frame of reference and enhancing interpretability, particularly for socio-technical systems.

7.2.3 A Digital Ecosystem is a system of systems and CORE PARTICIPANTS perform a complex choreography.

A Digital Ecosystem encompasses more than a singular, monolithic workflow. It can only be defined through the interconnected use cases in which the CORE PARTICIPANTS are involved. The full spectrum of activities within this complex system of systems cannot be centrally orchestrated. Thus, it resembles a natural ecosystem, from which the concept and term is derived:

Imagine an old-growth temperate forest as an ecosystem of interacting participants: Canopy trees capture sunlight and fix carbon, providing the structural “platform” on which the system rests. Fungi establish an underground network that transfers water, nitrogen, and chemical signals among otherwise competing tree species. Pollinators, such as bees, moths, and birds, navigate the underbrush, facilitating genetic exchange among understory plants. Beetles and earthworms decompose fallen wood and leaf litter, recycling nutrients back into the root-fungus network. Predators, including owls and martens, regulate herbivore populations, thereby preventing the overgrazing of saplings. No single organism schedules or directs these exchanges. Yet, through countless local interactions—each species "performing its role" and responding to its neighbors—the forest exhibits a stable, self-regulating choreography. The removal or distortion of any key actor causes the entire ecosystem to falter, illustrating that the system's health arises from decentralized cooperation rather than central control.

The process-centric approach in I1 aimed to reflect the choreography of actors through interconnected SIDs. However, this approach was insufficient to adequately capture the dynamic nature of a Digital Ecosystem. Therefore, I2 and I3 placed the CORE SERVICES at the center of consideration. These are separable building blocks for describing the Digital Ecosystem. This structure resembles that of Catena-X and other X-projects, which are organized around use cases. Each of these use cases comprises interrelated yet independent processes among different actors. This organization is characteristic of federated choreography rather than centralized orchestration.

This separation of functionality is evident in the EDE methodology framework. Within this framework, Use Cases and Functions can be identified and described independently. The Synthesis Model supports the modeling of stand-alone processes for each Function, allowing their integration into Use Cases. However, the Use Cases are not interconnected within a comprehensive Digital Ecosystem process or workflow. Nonetheless, Use Cases delineate the characteristic functionality of the Digital Ecosystem. The concept of systems of systems is further emphasized by the ability to integrate entities from different Synthesis Models, analogous to a Venn Diagram, to create a new Digital Ecosystem.

7.2.4 Subject-Orientation is well suited for socio-technical Digital Ecosystems.

Subject-orientation originates from research in distributed systems and is thus tailored for federated Digital Ecosystems. This is supported by literature that indicates most descriptions of Digital Business Ecosystems (DBEs) primarily focus on process descriptions, with actors being the most commonly utilized elements. This emphasis on actors and processes also serves as the foundation for subject-oriented modeling. However, explicitly incorporating a subject-oriented description, rather than solely relying on implicit semantics, significantly enhances the expressiveness and comprehensibility of these descriptions. Moreover, the paradigm of subject-orientation can be applied to other modeling perspectives, extending beyond process modeling. The essence remains in the clear identification of all active entities.

Subject-orientation is particularly advantageous when multiple active entities engage in shared processes. Socio-technical systems, such as Digital Ecosystems, inherently comprise multiple active entities, specifically at least one human and one technical actor.

Consequently, all three iterations of SDSM incorporated the general requirement to clearly identify active entities within the utilized notations. This alignment was evaluated and found satisfactory for all three iterations. The subject-oriented focus received positive feedback from stakeholders, with PASS generating the most interest among the various modeling notations employed throughout IntWertL. This may suggest an enhanced reception of subject-oriented models; however, a comparative analysis for validation is beyond the scope of this study.

The notation choices made for the Synthesis Model facilitate consistent subject-orientation throughout the EDE methodology. As demonstrated, the EDE methodology supports a persona decomposition that identifies active entities, namely actors, roles, and subjects, across various levels. Each of these entities is described in the `synthesisModel` at its respective level in a manner applicable solely to active entities. Consequently, all elements in each layer of the Synthesis Model can be easily classified as either active or passive as is illustrated in Figure 7.3.

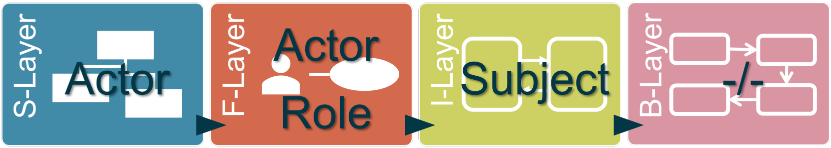


Figure 7.3: Enumeration of active entities on each SFIB-Layer.

7.2.5 A Digital Ecosystem model requires multiple perspectives.

Digital Ecosystems unite numerous Core Participants from diverse backgrounds and encompass various Core Services across different domains. Therefore, it is highly unlikely that a single perspective can fully capture the complexity of a Digital Ecosystem. Consequently, both experience from MBSE and and the literature on DBE modeling suggest the inclusion of different types of models. However, since subject-orientation has primarily been viewed in the context of process modeling, this raises the question of how to implement subject-orientation in other model types.

This finding is supported by I1 and I2 of the SDSM, indicating that neither the process-centric nor the structure-centric approach is satisficing to comprehensively describe Digital Ecosystems. I2 has demonstrated that omitting process information leads to the "wild growth" of incoherent and informal descriptions, as developers and adopters rely on more precise process information. This point is further emphasized in I1, where the Synthesis Model V1 lacks behavioral information, resulting in the development of the scenario process model as a foundation for MVP creation and validation. Furthermore, it has been determined that separating interests and creating new perspectives through context differentiation is advantageous. This allows stakeholders to concentrate on the content that is relevant to them.

Both distinctions of perspectives are incorporated into the EDE methodology. First, the SFIB-Layers of the Synthesis Model provide four distinct perspectives through their respective model types. In this context, the S-Layer and F-Layer aggregate information that establishes a reference framework for subsequent context separation. Thus, the structural description serves as the basis for functional decomposition, thereby facilitating context separation. Consequently, S-Layer and F-Layer must engage with managerial stakeholders. However, a process description is essential for the adoption and implementation of the use cases. In this regard, the IB-Layers are utilized.

Second, the separation of context for both model utilization and maintenance offers additional perspectives. As a result, FIB-Layers incorporate multiple diagrams specifically adapted to the context derived from the next higher Synthesis Model layer. Each diagram can be understood and designed independently of the others within the same layer. This allows adopters and developers with practical interests to access the necessary information without requiring a comprehensive understanding of all interconnected components within the Digital Ecosystem, while still maintaining coherence among all diagrams.

7.2.6 Digital Ecosystems evolve over time.

Constant growth, iterative evolution, and open innovation are fundamental to Digital Ecosystems. Therefore, their only constant is change. The representation of the subject-oriented model must facilitate easy and ideally federated maintainability. This necessitates rigorous versioning and release planning of the model artifacts.

The three SDSM cycles demonstrate that the Synthesis Model undergoes intentional evolution over time. This evolution encompasses not only the emergence of Digital Ecosystems but also constitutes a defining characteristic of the ecosystem itself. Therefore, clear identification of the model version is essential for effective stakeholder communication. Observations from the SDSM iterations suggest that developing subsequent versions in a federated manner is beneficial; additionally, sharing updates and information about upcoming releases improves stakeholder engagement and acceptance.

This evolution is natively supported by the EDE Synthesis Model. Firstly, its metadata includes not only an ID but also a version property. Since the ID is not constrained by a specific pattern, it can be readily employed for release management and versioning across various levels. Secondly, integrating new use cases or entire domains is straightforward. The same applies to the ecosystem personas, particularly as roles and subjects are already defined at the use case and functional levels, respectively. Section 6.5 clearly describes a suitable approach to support evolution in the application of the Synthesis Model and EDE methodology.

Furthermore, the EDE approach is also based on the concept of evolution. It not only iterates the CORE SERVICES cycle to further refine the functional architecture, but it also intends to regularly revisit the CORE VALUES of the Digital Ecosystem. This allows for the integration of this evolution into the fundamental definitions of purpose, vision, mission, and values of the entire ecosystem.

7.3 Limitations of the EDE Methodology

Although the EDE methodology adheres to the seven DSR guidelines outlined in Hevner et al. (2004), several limitations hinder the strength and generalizability of its contributions. These limitations should guide the interpretation of the presented results and inform the design of future studies aimed at extending this research.

1. **Single–case setting.**
   Validation is based on a longitudinal case study (*IntWertL*) conducted in the German mechanical engineering sector, specifically targeting SMEs in the automotive industry. While this environment offers a

rich context for exploratory evaluation, claims of external validity require replication across other industries, ecosystem scales, and cultural contexts.

2. **Predominantly formative evaluation.**
   The evidence of utility primarily derives from formative evaluation episodes of design artifacts. A rigorous summative comparison with alternative Digital Ecosystem design artifacts remains to be conducted.

3. **Adoption challenges of subject-orientation.**
   Subject-oriented modeling has demonstrated benefits in federated, socio-technical environments. However, its cognitive overhead and unfamiliar notation may hinder adoption in domains where process-centric thinking is not well established or where alternative process modeling methods are more commonly used.

4. **Limits of JSON.**
   The JSON-based `synthesisModel` currently lacks expressiveness and semantic reasoning capabilities. Additionally, there's no assessment of efficiently integrating inputs from multiple contributors. Future research should investigate if utilizing graph databases or OWL encodings could enhance reasoning and facilitate version control.

5. **Tool support and automation.**
   Prototype scripts facilitated model generation; however, no integrated toolchain, such as an IDE plugin or web workbench, currently exists. The absence of mature tooling limits practitioner adoption and compromises repeatability.

6. **Incremental evolution not yet empirically verified.**
   Although the methodology prescribes iterative evolution cycles, only three releases have been observed. The long-term governance mechanisms and continuous application of the EDE methodology remain uncertain.

7. **Limited summative analysis.**
   Utility was validated solely through construction and demonstrated through qualitative stakeholder feedback and acceptance of artifacts. However, the application of the EDE methodology itself was beyond the scope of this study. Consequently, summative evaluation with practitioners of the EDE methodology will broaden the evidence base for claims of efficiency.

Further investigation is necessary to compare the effectiveness of alternative methodologies in the context of Engineering Digital Ecosystems. This statement is not a critique of the current work but rather a call for further research in this emerging field. As research progresses from this foundational study, rigorous assessment and comparison with other methods across various contexts are essential to substantiate claims of general applicability.

7.4 Summary

This chapter situates the EDE methodology within the framework of DSR and consolidates the empirical insights obtained from the *IntWertL* case study.

Initially, a structured mapping against the seven DSR guidelines of Hevner et al. (2004) indicated that the study addresses a problem of significant practical relevance, employs rigorous methods and a transparent knowledge base, conceptualizes design as an iterative search process, produces artifacts of demonstrable utility, evaluates those artifacts through multiple formative episodes, articulates clear research contributions at two levels (instantiated design artifacts and the EDE methodology itself), and effectively communicates the results to both academic and practitioner audiences.

Second, a critical reflection revisits six guiding hypotheses derived from the current state of the art. The analysis demonstrated how each hypothesis influenced the three iterations of the SDSM presented in Chapter 5 and is now integrated into the EDE methodology, the Double Diamond framework, and the Synthesis Model. The discussion highlighted the benefits of incorporating Core Values, embracing a model-based approach, recognizing Digital Ecosystems as systems of systems, adopting subject-oriented modeling, providing multi-perspective views, and facilitating continuous evolution.

Third, the chapter identifies seven limitations that impact the generalizability of the findings: the single-case setting, the predominance of formative evaluation, the adoption of a subject-oriented approach, issues related to the current JSON representation, limited tool support, emerging evidence on ecosystem evolution, and the absence of summative evaluation of the methodology itself. These constraints establish a clear agenda for future research.

8 Conclusion and Outlook

This final chapter summarizes the key points of the study, draws conclusions, and offers an outlook with suggestions for future research.

8.1 Conclusion

Digital Ecosystems are socio-technical systems whose emergent properties present significant potential for securing competitive advantages. However, effective design is obstructed by the complex interaction patterns among Core Participants. Although both academia and industry recognize the critical importance of modeling these ecosystems, practical modeling approaches remain inadequate, and no formal modeling languages have been established. Therefore, this work proposes the comprehensive Engineering Digital Ecosystems (EDE) methodology for the model-based and subject-oriented design of Digital Ecosystems.

The conclusion is organized into three subsections. First, it delineates the status quo and the initial context of the research. Next, it summarizes the Design Science Research (DSR) approach utilized to examine the solution space and gain insights. Finally, it reassesses the resulting EDE methodology.

8.1.1 Summary of the initial situation

Chapter 1 introduces the potential of Digital Ecosystems for their participants. These networks enhance flexibility, resilience, and adaptability by emphasizing collaborative value creation, particularly for SMEs. By promoting coopetition, the Digital Ecosystem serves as an incubator for innovation, fostering the emergence of disproportionate value creation. However, the development of these complex socio-technical systems, involving multiple stakeholders, necessitates adequate resources. This motivates the main objectives of this work: a *model-based methodology* and a *subject-oriented model* for Engineering Digital Ecosystems. Based on this, the chapter details the research outline.

Chapter 2 establishes the foundational context essential for understanding the remainder of this work. Initially, the term *Digital Ecosystem* and its primary components are defined: CORE VALUES influence CORE SERVICES. ORCHESTRATORS oversee these components and facilitate the development of interoperable and competitive SOLUTIONS. Meanwhile, CORE PARTICIPANTS seamlessly adapt roles and engage in coopetition when utilizing or contributing to SOLUTIONS. Next, the term *model-based* within Model-Based Systems Engineering (MBSE) is clarified, introducing the Systems Modeling Language (SysML) and focusing on its diagrams: Block Definition Diagram (BDD) and Use Case Diagram (UCD). The paradigm of *subject-orientation* and its application through the Parallel Activity Specification Scheme (PASS) are additionally presented. Lastly, *design* is explored, connecting human-centered design thinking, the Double Diamond model's divergence-convergence process, and the disciplined approach of DSR.

Chapter 3 reviews current research and industry practices in Digital Ecosystems and their modeling. In academia, there is a recognized need for improved modeling methods for Digital Business Ecosystems (DBEs) as current practices primarily employ informal and non-standardized notations that focus on a singular perspective. Subject-oriented modeling demonstrates promise, as it aligns with the distributed nature of Digital Ecosystems and offers the necessary expressiveness for process information, which is considered integral. On the industry side, initiatives such as IDS, Gaia-X, and Catena-X are analyzed as they advance operational ecosystems. The analysis of the state-of-the-art in academia and industry identifies four research gaps, validating the utility of the two research questions. These findings form the foundation for formulating six guiding hypotheses for the EDE methodology regarding: CORE VALUES and CORE SERVICES, the model-based approach, the system of systems concept, subject-orientation, integration of multiple perspectives, and the evolution of Digital Ecosystems.

At this stage, existing research gaps are clearly identified, accompanied by two distinct research questions (RQ1 & RQ2) that guide the research process. The six hypotheses (H1-H6) will inform the development of the EDE methodology and will provide the foundation for qualitative evaluation.

8.1.2 Design Science Research

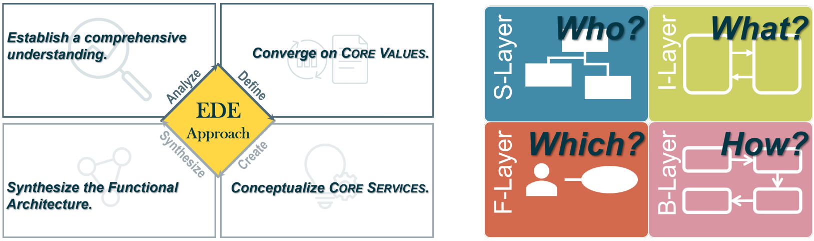
This study employs DSR as its methodological foundation. Chapter 4 introduces the DSR paradigm and the design guidelines by Hevner et al. (2004). In addition to the research approach, the application context, IntWertL, is also introduced. Utilizing a comprehensive DSR methodology framework, the Soft Design Science Methodology (SDSM) is chosen and adapted for this study by developing a specific evaluation strategy based on the Framework for Evaluation in DSR (FEDS). This approach ensures that evaluations validate the contribution of each artifact version in addressing the design issue and generating transferable knowledge. The adapted SDSM comprises nine steps across three iterations, resulting in three general solutions, three specific design artifacts, and nine evaluation episodes. The chapter concludes by discussing how insights from the SDSM are transformed into the primary contribution: the EDE methodology.

Chapter 5 presents findings from the three SDSM iterations conducted within the context of IntWertL. Both the design artifacts and their evaluation results, along with the design process itself, inform the EDE methodology; thus, special emphasis is placed on presenting all nine SDSM steps for each of the three iterations. The first iteration, which employed a process-centric approach with PASS SIDs, established the foundation for the MVP and the IntWertL ecosystem. However, a Digital Ecosystem cannot be confined to a single process. The second iteration addressed structural issues by utilizing a modified BDD for representation, enabling the development of business and operational models, as well as the breakdown of functional architecture. The third iteration integrated the previous approaches by employing BDDs, UCDs, SIDs, and SBDs for a cohesive representation.

In the context of DSR, business needs from the IntWertL environment, alongside applicable foundations and methodologies from the knowledge base, inform the iterative build-test cycle. The resulting artifacts are then utilized within the IntWertL ecosystem. Moreover, and crucial to DSR, insights gained from all three iterations contribute to the development of the EDE methodology, thereby enriching the DSR knowledge base.

8.1.3 Summary of the EDE methodology

The EDE methodology constitutes the primary contribution of this study and is detailed in Chapter 6. It establishes a generalized framework for the model-based and subject-oriented design of Digital Ecosystems. As depicted in Figure 8.1, the EDE methodology comprises two complementary components: the EDE Double Diamond Approach and the Synthesis Model.



(a) EDE Approach (Double Diamond)

(b) Synthesis Model

Figure 8.1: Core components of the EDE methodology.

To address RQ1, a structured methodological approach is presented, that guides the model-based design process. This approach consists of four steps and modifies the Design Council's Double Diamond model into two cycles:

- **CORE VALUES Cycle** (Analyze, Define), which elicits and refines the ecosystem's problem domains, purpose, vision, mission and value propositions;
- **CORE SERVICES Cycle** (Create, Synthesize), which concretizes the functional architecture that will deliver those values.

The *Synthesis Model*, corresponding to RQ2, facilitates a subject-oriented and model-based (re-) presentation of Digital Ecosystems through interconnected layered views of SFIB (Structure, Function, Interaction, and Behavior). This encompasses a formal `synthesisModel` representation in JSON schema. These elements address EDE's core challenges: complexity, federated structure, and diverse stakeholders.

The chapter outlines best practices, recommended methods, and a practical application example to assist practitioners in implementing the EDE methodology.

In Chapter 7, the EDE methodology is critically examined. First, structured mapping through the seven DSR guidelines by Hevner et al. (2004) confirms its effectiveness as a DSR artifact. Next, a reflection on six guiding hypotheses demonstrates advantages of including CORE VALUES with CORE SERVICES, utilizing a model-based approach, recognizing Digital Ecosystems as systems of systems, applying subject-oriented modeling, offering diverse perspectives, and facilitating ongoing evolution. Finally, the chapter identifies seven limitations that affect the generalizability of the findings: reliance on a single case, predominance of formative evaluation, adoption of a subject-oriented approach, challenges with the current JSON representation, insufficient tool support, early evidence of ecosystem evolution, and lack of conclusive summative evaluation.

Through its Validation by Creation, the EDE methodology represents the first satisficing methodology of its kind. It effectively addresses both research questions: first, by advocating a model-based engineering approach through the EDE Double Diamond; and second, by providing a comprehensive subject-oriented Synthesis Model for characterizing Digital Ecosystems. Additionally, the coherence and efficacy of the entire methodology rely on the six originally proposed hypotheses, which serve as the foundation for every aspect of EDE's design and validation.

8.2 Outlook

To conclude, this outlook will address unresolved methodological issues and propose directions for future research.

These issues can be categorized into two groups: tasks for further validation of the EDE methodology and areas requiring additional development. All specific insights are derived from a single case study. Future research should apply the EDE methodology in designing other Digital Ecosystems with varying levels of maturity across diverse domains, including aerospace, automotive, plant engineering, railway, and robotics. This entails following the complete approach for designing these ecosystems and creating a Synthesis Model of existing Digital Ecosystems within these fields. A summative evaluation of these efforts will provide substantial evidence regarding the utility of the methodology. The research should also investigate the quality of SOLUTIONS and their design processes that instantiate the CORE SERVICES based on a Synthesis Model as further validation for the methodology.

Additionally, while all six hypotheses have been shown to be supported, further evaluation is necessary to determine whether the application of these hypotheses can be inferred as causative for the utility of an artifact within the EDE context. Furthermore, the activities should investigate which additional hypotheses should be considered to develop a comprehensive design theory regarding the embedded phenomena in Digital Ecosystems.

Lastly, two primary avenues exist for enhancing the EDE methodology. The first involves refining the prototypical tools that facilitate its application, particularly in the creation and visualization of the Synthesis Model. In terms of creation, this may incorporate interactive methods for editing and viewing, including synchronous shared editing functionality. Regarding visualization, there is significant potential to improve the user experience of the model viewer through dedicated application styling and more coherent diagramming. Additionally, this includes revisiting the current JSON schema and analyzing whether alternative means of representation may be superior. For instance, it is plausible that the use of the Web Ontology Language could yield benefits in terms of expressibility; however, a simplified YAML structure might enhance readability and support a diagramming-in-code approach.

Secondly, the EDE methodology is employed to design the functional architecture of a Digital Ecosystem, i. e. its CORE SERVICES. However, it currently does not encompass the design of SOLUTIONS, business cases, and operating models. Nevertheless, all of these elements depend on the information provided in the Synthesis Model. Future research should investigate whether notation refinements, additional layers, or another dimension could support the respective design processes while ensuring adherence to CORE SERVICES specifications. In this context, an interesting avenue of research could be to employ ALPS for modeling CORE SERVICES. The process descriptions for instantiating SOLUTIONS can then be automatically checked for compliance, thereby reducing certification efforts. Extending the Synthesis Model to include SOLUTIONS would position it as a holistic artifact for both the development and operation of Digital Ecosystems.

Overall, this study serves as a foundation rather than a conclusion for the rigorous design of Engineering Digital Ecosystems.

List of Publications

Gereon C. Bönsch, Jakob Bönsch, Matthes Elstermann, Maximilian Kuhn, Hendrik Lauf, Michael Riesener, and Günther Schuh. Process Documentation for the Agile Development of Wind Turbines Using Subject-Oriented Modeling. In Francesco Gabriele Galizia and Marco Bortolini, editors, *Production Processes and Product Evolution in the Age of Disruption*, pages 506–515, Cham, 2023a. Springer International Publishing. ISBN 978-3-031-34821-1. doi: 10.1007/978-3-031-34821-1_55.

Jakob Bönsch, Matthes Elstermann, Andreas Kimmig, and Jivka Ovtcharova. A subject-oriented reference model for Digital Twins. *Computers & Industrial Engineering*, 172:108556, October 2022a. ISSN 0360-8352. doi: 10.1016/j.cie.2022.108556. URL `https://www.sciencedirect.com/science/article/pii/S0360835222005617`.

Jakob Bönsch, Katharina Reh, and Jivka Ovtcharova. Subject-Oriented Business Process Models of SMEs: Case Study, Best Practices and Evaluation. In Matthes Elstermann, Stefanie Betz, and Matthias Lederer, editors, *Subject-Oriented Business Process Management. Dynamic Digital Design of Everything – Designing or being designed?*, pages 79–94, Cham, 2022b. Springer Nature Switzerland. ISBN 978-3-031-19704-8. doi: 10.1007/978-3-031-19704-8_5.

Jakob Bönsch, Svenja Hauck, Matthes Elstermann, and Jivka Ovtcharova. An Approach to Create a Common Frame of Reference for Digital Platform Design in SME Value Networks. In Matthes Elstermann, Anke Dittmar, and Matthias Lederer, editors, *Subject-Oriented Business Process Management. Models for Designing Digital Transformations*, pages 63–82, Cham, 2023b. Springer Nature Switzerland. ISBN 978-3-031-40213-5. doi: 10.1007/978-3-031-40213-5_5.

Jakob Bönsch, Lucas Greif, Svenja Hauck, Simon Kreuzwieser, Anjela Mayer, Felix Longge Michels, and Jivka Ovtcharova. Virtual Engineering: Hands-on Integration of Product Lifecycle Management, Computer-Aided Design, eXtended Reality, and Artificial Intelligence in Engineering Education. *Chemie Ingenieur Technik*, 96(11):1460–1474, 2024a. ISSN 1522-2640. doi: 10.1002/cite.202300169. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.202300169`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.202300169.

Jakob Bönsch, Vladimir Jelschow, Moritz Haking, Sleiman Bobbou, and Oliver Deisser. Intelligente Wertschöpfungsnetzwerke für individuelle Fahrzeugentwicklung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 119 (12):902–906, December 2024b. ISSN 2511-0896. doi: 10.1515/zwf-2024-1173. URL `https://www.degruyter.com/document/doi/10.1515/zwf-2024-1173/html`. Publisher: De Gruyter.

Matthes Elstermann, Jakob Bönsch, and Jivka Ovtcharova. Subject-Oriented Value-Stream Mapping with SiSi. In Michael Freitag, Aseem Kinra, Herbert Kotzab, Hans-Jörg Kreowski, and Klaus-Dieter Thoben, editors, *Subject-Oriented Business Process Management. The Digital Workplace – Nucleus of Transformation*, pages 232–250, Cham, 2020. Springer International Publishing. ISBN 978-3-030-64351-5. doi: 10.1007/978-3-030-64351-5_16.

Matthes Elstermann, Jakob Bönsch, Andreas Kimmig, and Jivka Ovtcharova. Human-Centered Referential Process Models for AI Application. In Alfred Zimmermann, Robert J. Howlett, Lakhmi C. Jain, and Rainer Schmidt, editors, *Human Centred Intelligent Systems*, pages 56–65, Singapore, 2021. Springer. ISBN 978-981-16-3264-8. doi: 10.1007/978-981-16-3264-8_6.

Lucas Greif, Jakob Bönsch, Karan Lashkare, Francesco Arcidiacono, Tobias Zuehlke, and Jivka Ovtcharova. Robuste Bauteilidentifikation mittels digitaler Fingerabdrücke. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 120(s1):232–235, March 2025. ISSN 2511-0896. doi: 10.1515/zwf-2024-0128. URL `https://www.degruyterbrill.com/document/doi/10.1515/zwf-2024-0128/html`. Publisher: De Gruyter.

Andreas Kimmig, Moritz Schöck, Edgar Mühlbeier, Florian Oexle, and Jürgen Fleischer. Wertstromkinematik – Produktionssysteme neu gedacht: Interdisziplinäres Forscherteam arbeitet an der Produktionstechnik der Zukunft (Teil 2). *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 116(12): 935–939, December 2021. ISSN 2511-0896. doi: 10.1515/zwf-2021-0207. URL `https://www.degruyterbrill.com/document/doi/10.1515/zwf-2021-0207/html`. Publisher: De Gruyter.

Simon Kreuzwieser, Andreas Kimmig, Felix Michels, Rebecca Bulander, Victor Häfner, Jakob Bönsch, and Jivka Ovtcharova. Human-Machine-Interaction in Innovative Work Environment 4.0 – A Human-Centered Approach. *New Digital Work – Digital Sovereignty at the Workplace. Ed.: A. Shajek*, page 68, 2023. doi: 10.1007/978-3-031-26490-0_. URL `https://publikationen.bibliothek.kit.edu/1000159373`. ISBN: 9783031264894.

Victor Kurtz, Jakob Bönsch, and Jivka Ovtcharova. Understanding and Harnessing the Potential of Conversational AI for S-BPM. In Matthes Elstermann, Stefanie Betz, and Matthias Lederer, editors, *Subject-Oriented Business Process Management. Dynamic Digital Design of Everything – Designing or being designed?*, pages 41–57, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19704-8. doi: 10.1007/978-3-031-19704-8_3.

Bibliography

Ron Adner. Ecosystem as Structure: An Actionable Construct for Strategy. *Journal of Management*, 43(1):39–58, January 2017. ISSN 0149-2063. doi: 10.1177/0149206316678451. URL https://doi.org/10.1177/0149206316678451. Publisher: SAGE Publications Inc.

Marco Ardolino, Anna Bino, Maria Pia Ciano, and Andrea Bacchetti. Enabling digital capabilities with technologies: A multiple case study of manufacturing supply chains in disruptive times. *Systems*, 13(1), 2025. ISSN 2079-8954. doi: 10.3390/systems13010039. URL https://www.mdpi.com/2079-8954/13/1/39.

Richard Baskerville, Jan Pries-Heje, and John Venable. Soft design science methodology. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, DESRIST '09, pages 1–11, New York, NY, USA, May 2009. Association for Computing Machinery. ISBN 978-1-60558-408-9. doi: 10.1145/1555619.1555631. URL https://dl.acm.org/doi/10.1145/1555619.1555631.

Maria Bengtsson and Sören Kock. ”coopetition” in business networks—to cooperate and compete simultaneously. *Industrial Marketing Management*, 29(5):411–426, 2000. ISSN 0019-8501. doi: https://doi.org/10.1016/S0019-8501(99)00067-X. URL https://www.sciencedirect.com/science/article/pii/S001985019900067X.

Tim Brown. Design thinking. *Harvard business review*, 86(6):84, 2008.

Richard Buchanan. Wicked Problems in Design Thinking. *Design Issues*, 8(2):5–21, 1992. ISSN 0747-9360. doi: 10.2307/1511637. URL https://www.jstor.org/stable/1511637. Publisher: The MIT Press.

Marc Burkhalter, Christian Betz, Stefanie Auge-Dickhut, and Reinhard Jung. Orchestrating Value Co-Creation in Business Ecosystems. In Karen Wendt, editor, *Theories of Change: Change Leadership Tools, Models and Applications for Investing in Sustainable Development*, pages 257–291. Springer International Publishing, Cham, 2021. ISBN 978-3-030-52275-9. doi: 10.1007/978-3-030-52275-9_16. URL https://doi.org/10.1007/978-3-030-52275-9_16.

Jakob Bönsch, Svenja Hauck, Matthes Elstermann, and Jivka Ovtcharova. An Approach to Create a Common Frame of Reference for Digital Platform Design in SME Value Networks. In Matthes Elstermann, Anke Dittmar, and Matthias Lederer, editors, *Subject-Oriented Business Process Management. Models for Designing Digital Transformations*, pages 63–82, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-40213-5. doi: 10.1007/978-3-031-40213-5_5.

Jakob Bönsch, Vladimir Jelschow, Moritz Haking, Sleiman Bobbou, and Oliver Deisser. Intelligente Wertschöpfungsnetzwerke für individuelle Fahrzeugentwicklung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 119 (12):902–906, December 2024. ISSN 2511-0896. doi: 10.1515/zwf-2024-1173. URL https://www.degruyter.com/document/doi/10.1515/zwf-2024-1173/html. Publisher: De Gruyter.

Abide Coskun-Setirek, Carmela Annosi , Maria, Hurst , William, Dolfsma , Wilfred, , and Bedir Tekinerdogan. Architecture and Governance of Digital Business Ecosystems: A Systematic Literature Review. *Information Systems Management*, 41(1):58–90, January 2024. ISSN 1058-0530. doi: 10.1080/10580530.2023.2194063. URL https://doi.org/10.1080/10580530.2023.2194063. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10580530.2023.2194063.

Agostinho da Silva and Antonio J Marques Cardoso. Enhancing sme competitiveness through coopetition networks: A service-dominant logic approach. *Management and Production Engineering Review*, 2024.

Lenny Delligatti. *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison–Wesley, Boston, 2014. ISBN 978-0-13-343033-2.

Aline Dresch, Daniel Pacheco Lacerda, and José Antônio Valle Antunes. Design Science Research. In Aline Dresch, Daniel Pacheco Lacerda, and José Antônio Valle Antunes Jr, editors, *Design Science Research: A Method for Science and Technology Advancement*, pages 67–102. Springer International Publishing, Cham, 2015. ISBN 978-3-319-07374-3. doi: 10.1007/978-3-319-07374-3_4. URL `https://doi.org/10.1007/978-3-319-07374-3_4`.

Matthes Elstermann. *Executing Strategic Product Planning - A Subject-Oriented Analysis and New Referential Process Model for IT-Tool Support and Agile Execution of Strategic Product Planning*. KIT Scientific Publishing, Karlsruhe, 2020. ISBN 978-3-7315-0972-1. doi: 10.5445/KSP/1000097859. Section: 334.

Albert Fleischmann. *Distributed Systems*. Springer, Berlin, Heidelberg, 1994. ISBN 978-3-642-78614-3 978-3-642-78612-9. doi: 10.1007/978-3-642-78612-9. URL `http://link.springer.com/10.1007/978-3-642-78612-9`.

Albert Fleischmann, Werner Schmidt, Christian Stary, Stefan Obermeier, and Egon Börger. *Subjektorientiertes Prozessmanagement: Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern*, volume 4. Hanser München, 2011.

Sanford Friedenthal, Alan Moore, and Rick Steiner. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, Waltham, MA, 3 edition, 2014. ISBN 978-0-12-800202-5.

Gaia-X European Association for Data and Cloud AISBL. Gaia-X: Technical Architecture, June 2020. URL https://www.bmwk.de/Redaktion/EN/Publikationen/gaia-x-technical-architecture.html. Accessed February 4, 2026.

Gaia-X European Association for Data and Cloud AISBL. Gaia-X Trust Framework — Release 22.10, October 2022. URL https://docs.gaia-x.eu/policy-rules-committee/trust-framework/22.10/. Accessed February 4, 2026.

Shirley Gregor and Alan R. Hevner. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37(2):337–355, 2013. ISSN 0276-7783. URL https://www.jstor.org/stable/43825912. Publisher: Management Information Systems Research Center, University of Minnesota.

Matthew Hause. The SysML modelling language. In *Fifteenth European systems engineering conference*, volume 9, pages 1–12, 2006. URL https://www.researchgate.net/profile/Matthew-Hause/publication/260359652_The_OMG_Modelling_Language_SYSML/links/5d706494299bf1cb80884eb5/The-OMG-Modelling-Language-SYSML.pdf. Accessed February 4, 2026.

Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105, 2004. ISSN 0276-7783. doi: 10.2307/25148625. URL https://www.jstor.org/stable/25148625. Publisher: Management Information Systems Research Center, University of Minnesota.

INCOSE. Systems Engineering Vision 2025, 2014. URL https://www.incose.org/. Accessed February 4, 2026.

Shun Ishihara and Taka Matsutsuka. Towards Interoperable Data Spaces: Comparative Analysis of Data Space Implementations between Japan and Europe, January 2025. URL http://arxiv.org/abs/2501.15738. Accessed February 4, 2026.

Jan Kaiser, Duncan McFarlane, Gregory Hawkridge, Pascal André, and Paulo Leitão. A review of reference architectures for digital manufacturing: Classification, applicability and open issues. *Computers in Industry*, 149: 103923, August 2023. ISSN 0166-3615. doi: 10.1016/j.compind.2023.103923. URL `https://www.sciencedirect.com/science/article/pii/S0166361523000738`.

James F. Moore. *The death of competition : leadership and strategy in the age of business ecosystems*. Wiley, 1996. URL `https://cir.nii.ac.jp/crid/1130282273219843328`.

Object Management Group. OMG Systems Modeling Language (SysML) Version 1.6. Technical Report formal/19-06-03, OMG, June 2019. URL `https://www.omg.org/spec/SysML/1.6`. Accessed February 4, 2026.

Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007. Publisher: Taylor & Francis.

Heinrich Pettenpohl, Markus Spiekermann, and Jan Ruben Both. International Data Spaces in a Nutshell. In Boris Otto, Michael ten Hompel, and Stefan Wrobel, editors, *Designing Data Spaces : The Ecosystem Approach to Competitive Advantage*, pages 29–40. Springer International Publishing, Cham, 2022. ISBN 978-3-030-93975-5. doi: 10.1007/978-3-030-93975-5_3. URL `https://doi.org/10.1007/978-3-030-93975-5_3`.

Plattform Industrie 4.0. Manufacturing-X: Shaping the Data Ecosystem for the Manufacturing Industry – Comprehensive Concept Paper, 2023. URL `https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Manufacturing-X_long.html`. Accessed February 4, 2026.

Jan Pries-Heje, Richard Baskerville, and John Venable. Soft Design Science Research: Extending the Boundaries of Evaluation in Design Science Research. In *Proceedings from the 2nd International Conference on Design Science Research in IT (DESRIST)*, pages 18–38, 2007.

Jan Pries-Heje, John Venable, and Richard Baskerville. Soft design science methodology. In *Situated Design Methods*. The MIT Press, 07 2014. ISBN 9780262321006. doi: 10.7551/mitpress/9936.003.0008. URL `https://doi.org/10.7551/mitpress/9936.003.0008`.

Werner Schmidt, Albert Fleischmann, and Oliver T. Gilbert. Subjektorientiertes Geschäftsprozessmanagement. *HMD Praxis der Wirtschaftsinformatik*, 46(2):52–62, April 2009. ISSN 2198-2775. doi: 10.1007/BF03340343. URL `https://doi.org/10.1007/BF03340343`.

Prince Kwame Senyo, Kecheng Liu, and John Effah. Digital business ecosystem: Literature review and a framework for future research. *International Journal of Information Management*, 47:52–64, August 2019. ISSN 02684012. doi: 10.1016/j.ijinfomgt.2019.01.002. URL `https://linkinghub.elsevier.com/retrieve/pii/S0268401218305991`.

Abdulrazak F. Shahatha Al-Mashhadani, Muhammad Imran Qureshi, Sanil S. Hishan, Mohd Shamsuri Md Saad, Yamunah Vaicondam, and Nohman Khan. Towards the Development of Digital Manufacturing Ecosystems for Sustainable Performance: Learning from the Past Two Decades of Research. *Energies*, 14(10):2945, January 2021. ISSN 1996-1073. doi: 10.3390/en14102945. URL `https://www.mdpi.com/1996-1073/14/10/2945`. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Herbert A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 3rd edition, 1996.

Ralph J. Smith. Engineering. *Encyclopaedia Britannica*, March 2024. URL `https://www.britannica.com/technology/engineering`. Accessed February 4, 2026.

Chen Hsi Tsai, Jelena Zdravkovic, and Janis Stirna. Modeling Digital Business Ecosystems: A Systematic Literature Review. *Complex Systems Informatics and Modeling Quarterly*, 0:1–30, April 2022. doi: 10.7250/csimq.2022-30.01.

John Venable, Jan Pries-Heje, and Richard Baskerville. FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, January 2016. doi: 10.1057/ejis.2014.36. URL `https://www.tandfonline.com/doi/abs/10.1057/ejis.2014.36`. Publisher: Taylor & Francis.

John R Venable, Jan Pries-Heje, and Richard L Baskerville. Choosing a Design Science Research Methodology. In *ACIS 2017 Proceedings*, 2017.

Roel J. Wieringa. The Design Cycle. In Roel J. Wieringa, editor, *Design Science Methodology for Information Systems and Software Engineering*, pages 27–34. Springer, Berlin, Heidelberg, 2014. ISBN 978-3-662-43839-8. doi: 10.1007/978-3-662-43839-8_3. URL `https://doi.org/10.1007/978-3-662-43839-8_3`.