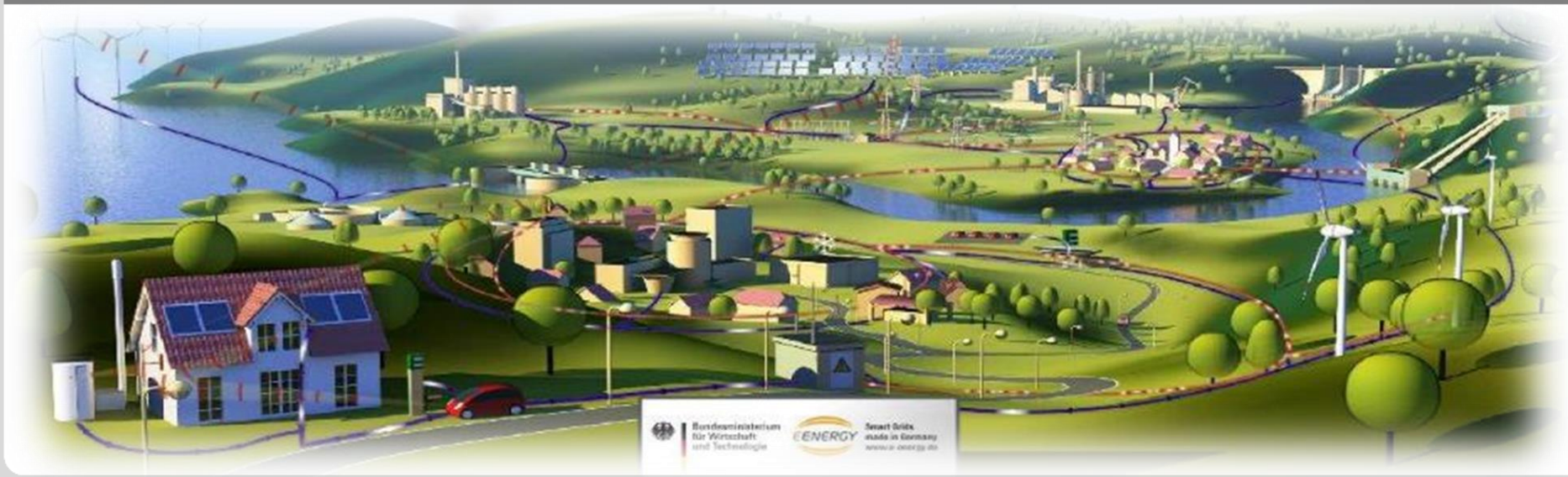
Karlsruhe Institute of Technology

Microservice-based Architecture for the Integration of Data Backends and Dashboard Applications in the Energy and Environment Domains

Jannik Sidler, Eric Braun, Christian Schmitt, Thorsten Schlachter and Veit Hagenmeyer

INSTITUTE FOR AUTOMATION AND APPLIED INFORMATICS (IAI)

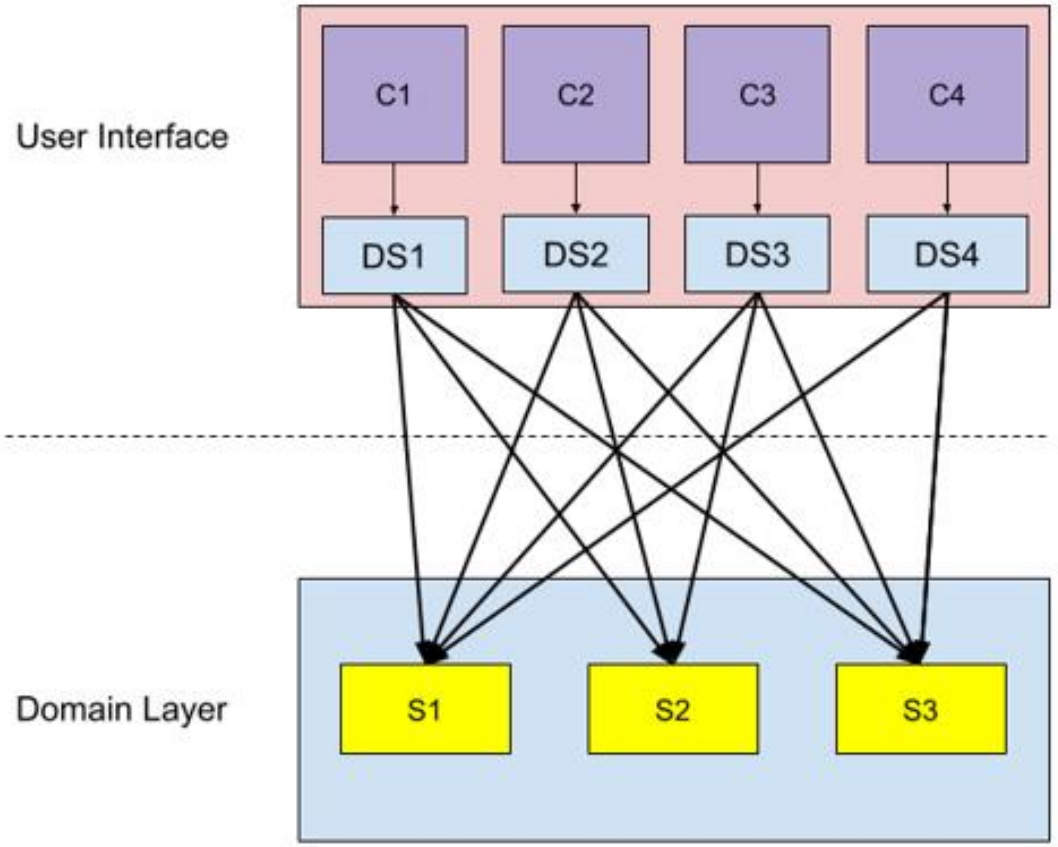


Motivation – General Aspects

- climate change
- renewable energies
- IAI – research in the energy and environment domain
  - various applications have already been created
- important components already exist
  - Generic Microservice Backend (GMB)
  - frontend framework based on reusable web components
- connecting layer still missing
  - goal: reduction of data transmission
- What is the purpose of this connecting layer?

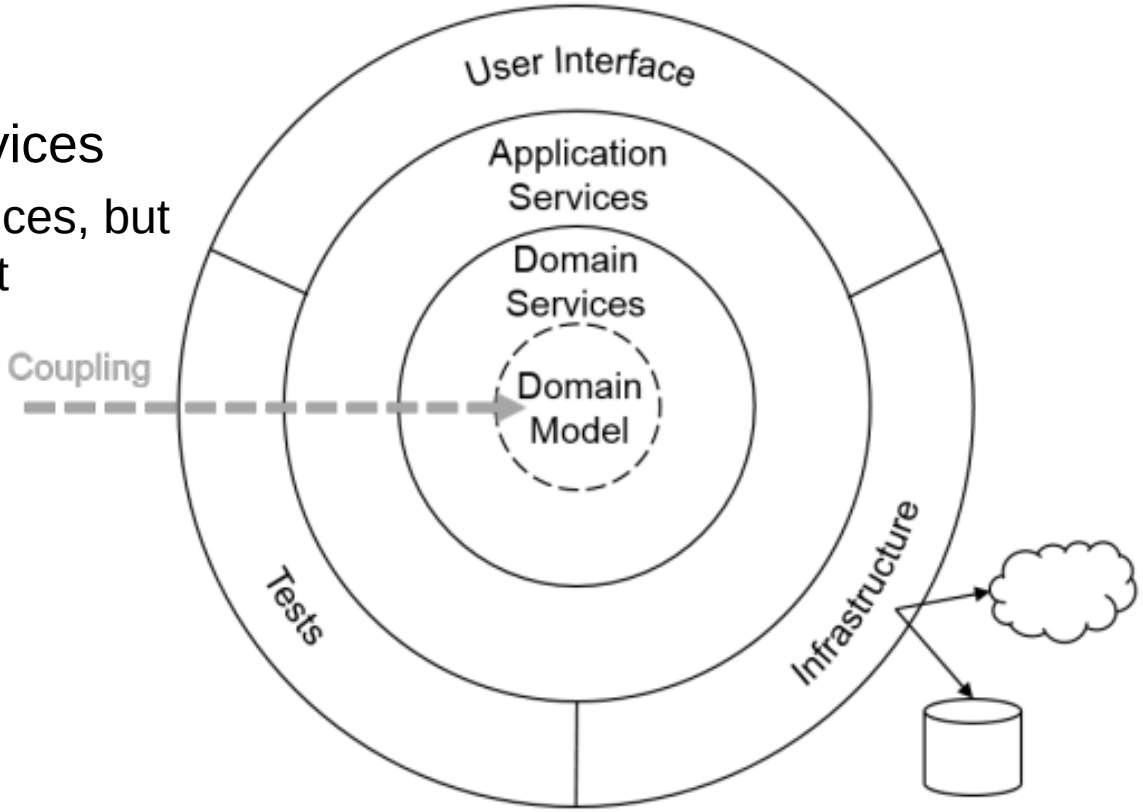
Motivation – Many Requests

- requests from frontend to backend are expensive
  - high physical distance
  - transmission of data
  - often high amount of requests necessary



Foundations – Onion Architecture

- idea derived from onion architecture
- separation of domain and application logic
- usage of application services
  - based on domain services, but principally independent

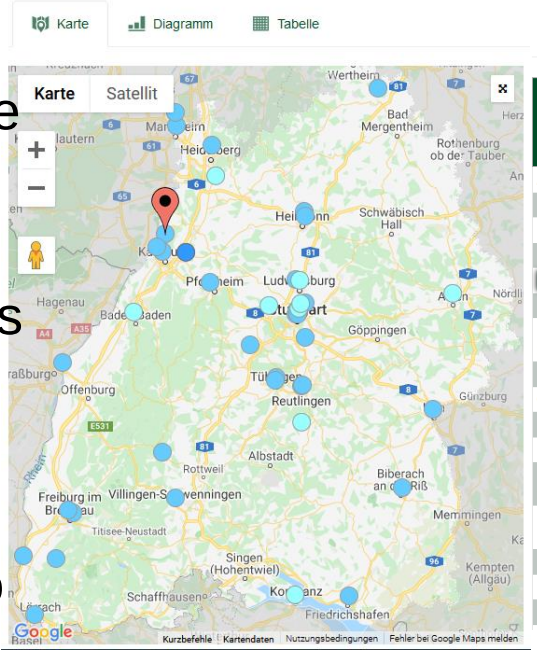


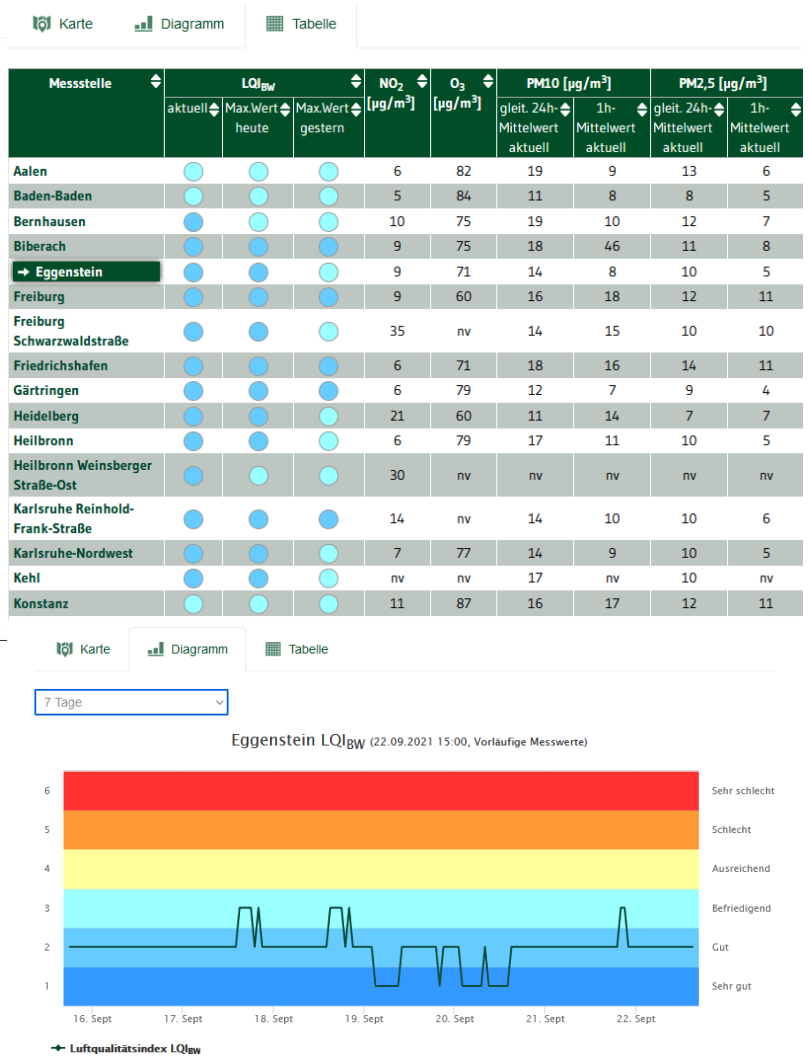
Foundations - Microservices

- encapsulate defined pieces of functionality
- commonly used in today's software design
  - good maintainability
  - good performance
  - well suited for modern server infrastructures
    - therefore efficient usage of servers
- usable in combination with the onion architecture

Air measurement

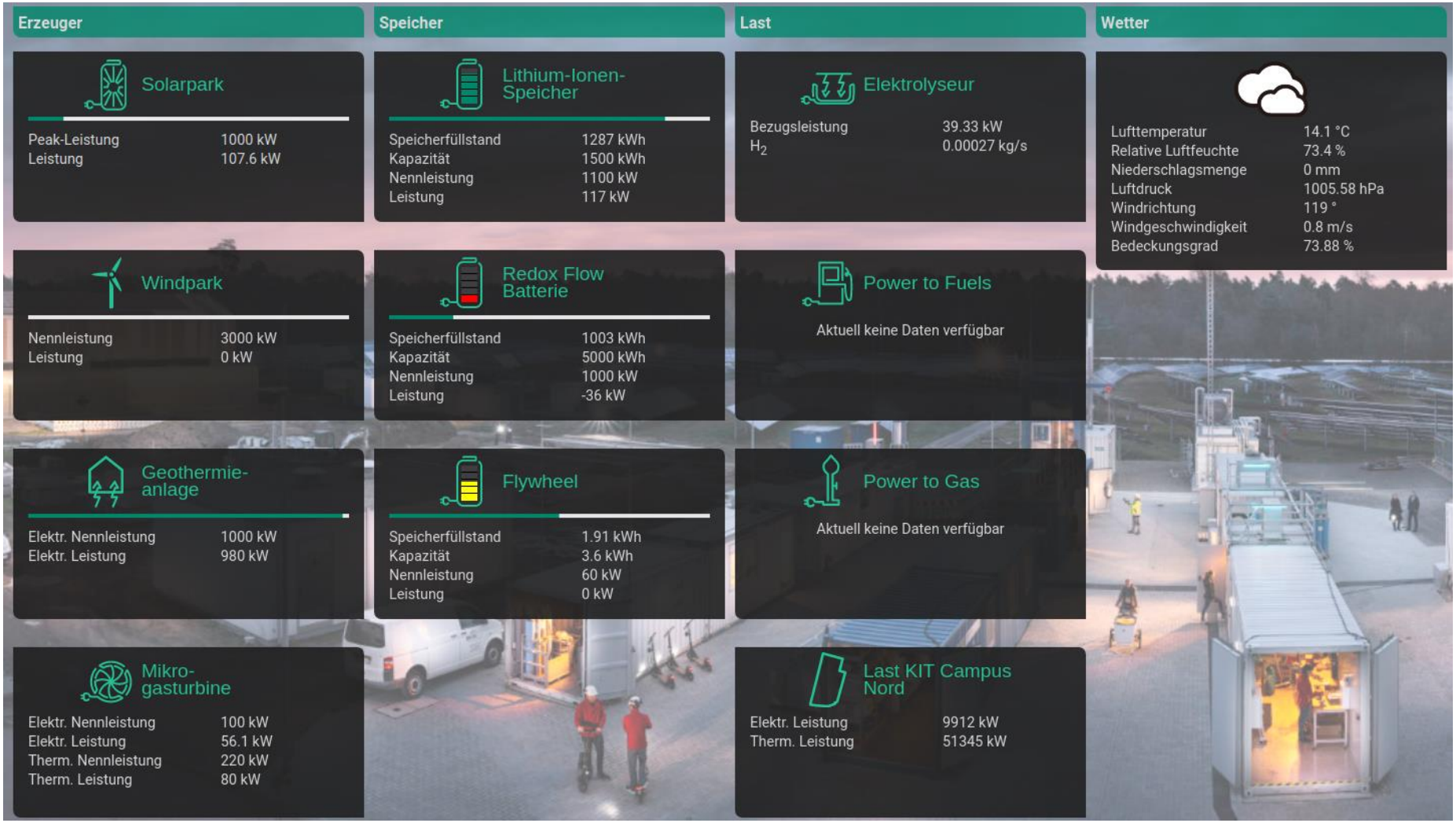
- domain services provide/aggregate measurement data
- frontend accesses domain services
- frontend processes requested data to generate format for UI components
- → user's hardware/network has to do the job!





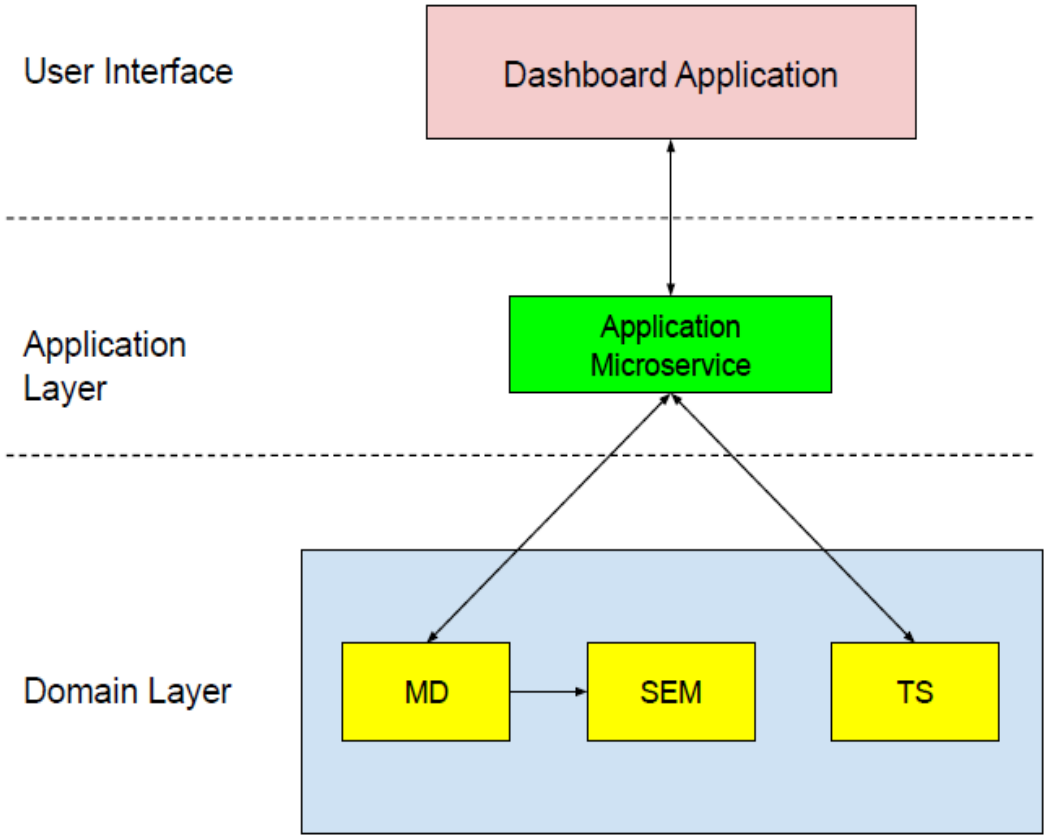
Messstelle	LQI$_{BW}$ aktuell	LQI$_{BW}$ Max.Wert heute	LQI$_{BW}$ Max.Wert gestern	NO_2 [µg/m³]	O_3 [µg/m³]	PM10 [µg/m³] gleit. 24h-Mittelwert aktuell	PM10 [µg/m³] 1h-Mittelwert aktuell	PM2,5 [µg/m³] gleit. 24h-Mittelwert aktuell	PM2,5 [µg/m³] 1h-Mittelwert aktuell
Aalen				6	82	19	9	13	6
Baden-Baden				5	84	11	8	8	5
Bernhausen				10	75	19	10	12	7
Biberach				9	75	18	46	11	8
→ Eggenstein				9	71	14	8	10	5
Freiburg				9	60	16	18	12	11
Freiburg Schwarzwaldstraße				35	nv	14	15	10	10
Friedrichshafen				6	71	18	16	14	11
Gärtringen				6	79	12	7	9	4
Heidelberg				21	60	11	14	7	7
Heilbronn				6	79	17	11	10	5
Heilbronn Weinsberger Straße-Ost				30	nv	nv	nv	nv	nv
Karlsruhe Reinhold-Frank-Straße				14	nv	14	10	10	6
Karlsruhe-Nordwest				7	77	14	9	10	5
Kehl				nv	nv	17	nv	10	nv
Konstanz				11	87	16	17	12	11

Energy Measurement



Software Architecture (1)

- domain layer consists of three microservices
- dashboard sends request to AppMS
- AppMS requests corresponding data from backend
- AppMS and domain microservices are hosted on the same server infrastructure

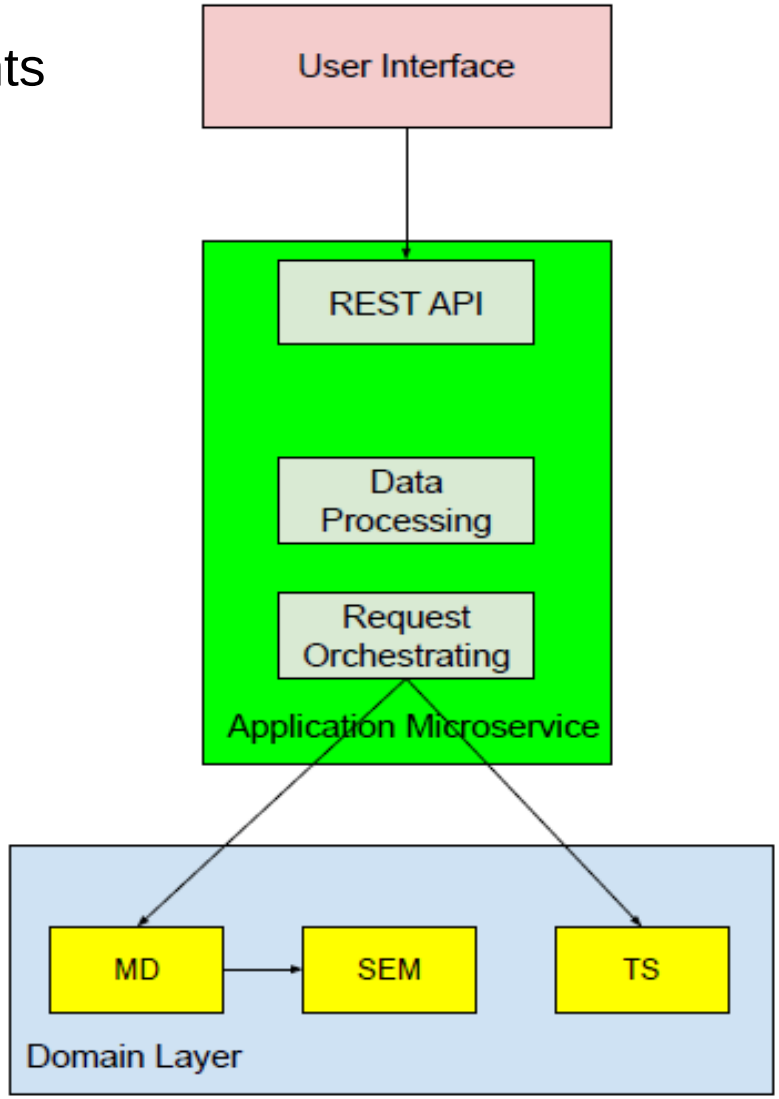


Software Architecture (2)

- domain layer consists of three microservices
  - Masterdata Service (MD)
    - provides master data of different objects in the context, e.g. measurement stations
  - Semantic Service (SEM)
    - validates master data objects
    - provides structural and semantic information for masterdata
  - Timeseries Service (TS)
    - manages time series data
    - aggregation, filtering, …

Software Architecture (3)

- AppMS has three base components
  - REST API
    - provides callable enpoints
    - called by client (User Interface)
  - Data Processing
    - generate output format
    - perform data processing steps
  - Request Orchestrating
    - make requests to the domain microservices



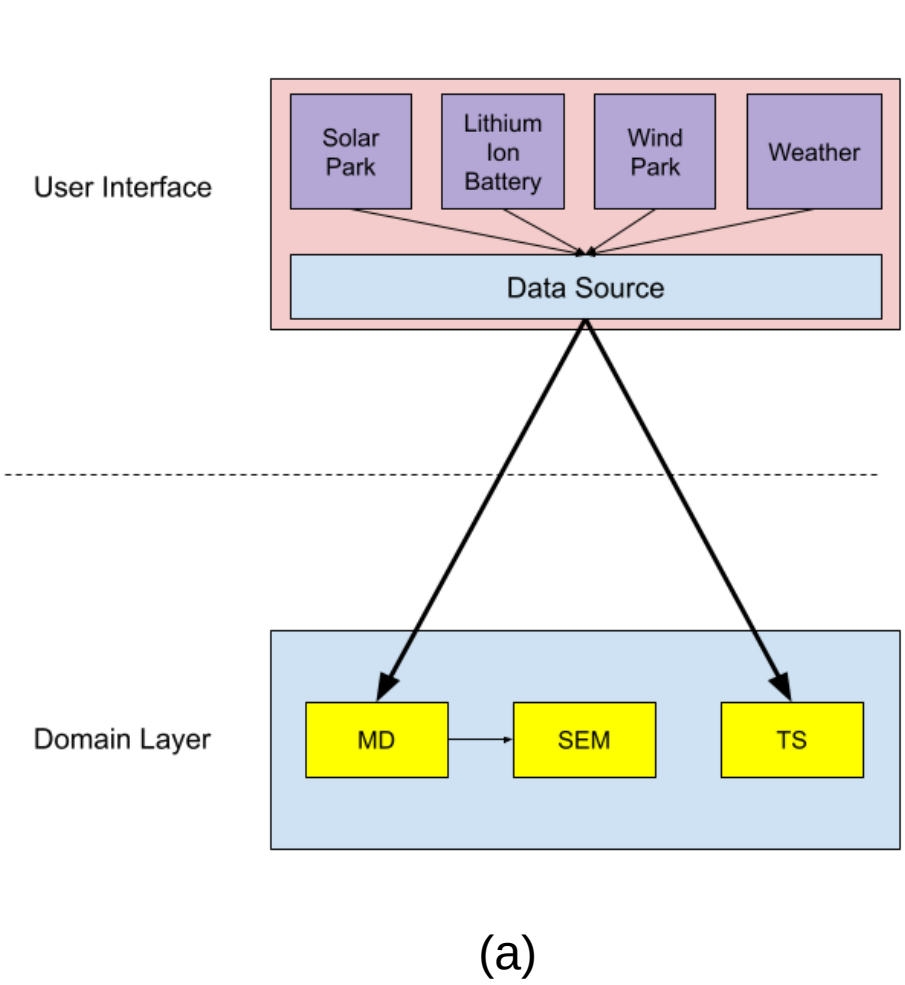
Evaluation (1)

- previously shown scenarios are used for validation
  - air measurement: no AppMS
  - energy measurement: with AppMS
- main purposes of AppMS:
  - reduction of network load
  - reduction of hardware requirements for end users
- network load aspect is examined in detail
- hardware requirements difficult to measure

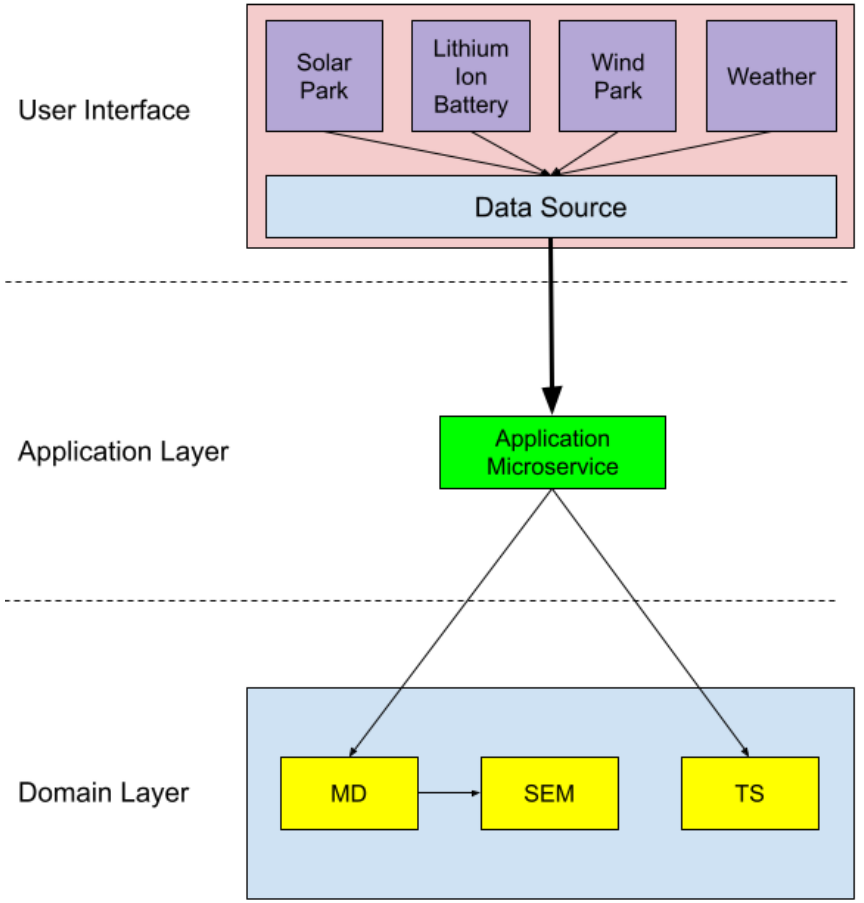
Evaluation (2)

- requests can be cheap or expensive
  - cheap request: made in same infrastructure/server environment
  - expensive request: made between two different infrastructures/server environments
- goal: reduce number of expensive requests
- two different scenarios
  - architecture uses no AppMS (a)
  - architecture uses AppMS (b)

Evaluation (3)



(a)



(b)

Evaluation (4)

Type	Number of expensive requests
no AppMS	8
with AppMS	1

- it is assumed that each component may need different resources
  - → worst case scenario (regarding required requests)
- data source can only make one request at once
  - it can call the application microservice which can make requests to all required resources

Conclusion

- extension of existing software architecture
  - usage of application service
  - more efficient sequence of requests
  - less resources for end users
- approach fits modern software architectures
- positive effect on power consumption (presumably)
- foundation for future works

Thank you…

… for your attention!

References

- Sidler et al. Design of a Web-Service for Formal Descriptions of Domain-Specific Data (2020)
  - https://link.springer.com/chapter/10.1007%2F978-3-030-39815-6_20
- Hippchen et al. Designing Microservice-Based Applications by Using a Domain-Driven Design Approach (2017)
  - https://cm.tm.kit.edu/download/domain_driven_microservice-architecture.pdf
- Palermo. The Onion Architecture (2008)
  - https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/