

Intellectual Property Protection in Model-Based Development – The Deltachain Approach

Thomas Weber

Karlsruhe Institute of Technology (KIT)

Karlsruhe, Germany

thomas.weber@kit.edu

Abstract

Industry is built atop the principle of intellectual property, i.e., knowledge about a system, a process, or something else that can be sold to a customer. Therefore, it is of utmost importance to protect this intellectual property from competitors. For the development of complex systems, companies need to collaborate, each contributing their share of intellectual property needed to develop the system. The exchange of overlapping parts, e.g., interfaces, is needed for the integration and ultimately building the system. Sharing the interface in an incremental way, i.e., by representing it as deltas, i.e., changes made to a model, improves collaboration efficiency. Continuous sharing instead of a version-based sharing can further increase the efficiency. We propose the Deltachain approach, which combines the delta-based representation of models with encryption-based access control, allowing us to use untrusted infrastructure for sharing models.

Keywords

Deltachain, Intellectual Property Protection, Delta-Based Representation, Encryption

ACM Reference Format:

Thomas Weber. 2026. Intellectual Property Protection in Model-Based Development – The Deltachain Approach. In *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE-Companion '26)*, April 12–18, 2026, Rio de Janeiro, Brazil. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3774748.3787625>

1 Introduction

While the thesis title suggests the Deltachain approach [18, 19] is limited to model-driven development [12], we first want to clarify that it is not, but is applicable to all artifacts that we can describe a semantic structure for. For that reason, we use the term artifact in this paper. The two aspects combined in this thesis, i.e., delta-based representation (DBR) [22] and encryption-based access control (EBAC) [5], do not depend on models in the sense of model-driven development, but rather in the sense of structure that can be assumed. Therefore, although the dissertation itself will be used and evaluated with models in the model-driven sense, the conceptual requirement is for the artifacts to be of a structure for which we can define meaningful deltas, as discussed in prior work [16], which we summarize in the next paragraph. The combination of

both DBR and EBAC allow us to provide a storage solution on untrusted third-party hardware while still being able to provide intellectual property protection and working in a change-based manner, improving collaboration productivity.

DBR. Meaningful delta refers to a notion of artifact representation with deltas, i.e., we derive an artifact by applying all the deltas that describe the steps necessary to create the state of the artifact [16, 22]. A fitting example is the representation of commits in Git¹, which consists of added and removed lines in text files or completely added or removed files. The replacement of lines is only needed as syntactic sugar to increase readability, although not technically necessary. The deltas for Git can be used because Git uses text files, which define a separator between parts of the artifacts, i.e., newline characters. For an artifact representation to be usable for our approach, we also need such a separator between meaningful parts of the artifact, to, e.g., define a delta which adds a UML class to a UML class diagram.

EBAC. Access control systems [10] restrict the access of subjects to objects according to rules. For the evaluation of these rules, usually a monitor is used, which takes access requests from subjects to objects and evaluates the based on the rules. If access is granted, the subject gains access to the object, e.g., to read or to modify it. This monitor needs to be trusted by all organizations sharing the object, which we deem infeasible for large-scale collaborations, especially in supply chains of complex cyber-physical systems. Instead of trusting a monitor, the sharing entity can also enforce access rights by using encryption and sharing the keys with the collaborating organizations that should gain access, called EBAC [5]. For EBAC, the whole artifact is divided into parts to which a certain set of entities should gain access. Afterward, the part is encrypted, and the keys used for the encryption are distributed to all entities that should gain access. One drawback of this approach is that every time something in the artifact changes, the new artifact needs to be encrypted and shared again. Furthermore, the approach is defined on specific artifact types.

Background & Motivation. For the sharing of artifacts, we want to be able to share only what has been changed, using deltas, and we want to be able to apply it to all kinds of artifacts. Therefore, we combine EBAC with DBR and as discussed, we want to build on a defined structure. In our case, this is, regarding the implementation, a metamodel, and the definition of deltas for that metamodel, similar to [4, 20]. Deltas have been used for other use cases, e.g., consistency preservation, and have proven very useful [9]. As metamodels as a concept are domain-independent, the Deltachain



This work is licensed under a Creative Commons Attribution 4.0 International License. *ICSE-Companion '26, Rio de Janeiro, Brazil*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2296-7/26/04

<https://doi.org/10.1145/3774748.3787625>

¹<https://git-scm.com/>, accessed 12.11.2025

approach supports all domains. The concrete metamodels used to describe domains are of course not domain-independent, but we built on the domain-independent meta-metamodel [12] and for the implementation on the Eclipse Modeling Framework [13]. Both come with their technical limitations but do not restrict the concept to a specific domain or tool-set.

Related Work. For being domain-independent, approaches like Git can be used. On top of Git, a layer to encrypt the data can be added, as shown in [11]. The two approaches Git-crypt² and git-encrypt³ use encryption of files inside a Git repository to restrict the access to them. However, the approach separates between public and private files and not between different groups of entities that should gain different fine-grained access to the artifacts inside the repository. In addition, our flexibility in regard to the delta definition allows for solutions tailored for a specific domain. For the specific domain of models and therein model repositories, a current challenge lies in the protection of intellectual property [2]. The proposed Deltachain approach can be used as a technical foundation for conceptual architectures for cross-organizational collaborations, e.g., as outlined in [7]. Close to our approach, not only by the name, are blockchains and especially blockchain databases [1]. Therein, the deltas are inflexible compared to our approach. Closely related to blockchain databases are distributed ledgers [15], which are similar to the Deltachain, but do not incorporate EBAC.

Organisational. This dissertation project is advised by Ralf Reussner⁴ at Karlsruhe Institute of Technology. I am currently in the midst of my fourth year out of five years planned.

2 Requirements

We defined seven requirements to our approach, which we discuss here shortly. For a longer version, we refer to the vision paper [18]. Please note that we derived the requirements from the cited papers in the sense, that the papers do pose the requirements for the approaches discussed themselves. The first requirement is that we can use the Deltachain as a single source of truth [8], i.e., that all aspects of the system that need to be described, can be appended to the Deltachain. We ensure that by using metamodel-independent deltas. The second requirement is that we do not want collaborators to provide copies of artifacts or artifact parts, but only the access is shared [5]. Note that this requirement goes beyond just using the Deltachain as a single source of truth, because we explicitly require sharing access. The third requirement is the protection of intellectual property [3], which we incorporate by using EBAC and respective mechanisms to distribute keys. The fourth requirement is that different information sharing scopes can be incorporated [14], starting from real-time collaboration and going to an infrequent synchronization, which are all possible ways to use the Deltachain. The fifth requirement is that we only want to exchange what has changed, i.e., we only want to store and distribute deltas and not model states [22]. We do so by explicitly defining and using a delta metamodel, which is defined on the meta-metamodel level to be applicable to all metamodels and therefore every domain. The sixth

requirement is that we want to be able to share the Deltachain on untrusted hardware [21], which we enable by encrypting the deltas before storing them on the Deltachain. The seventh requirement is that we are on the one hand, domain-agnostic, but that the Deltachain can be parametrized in a way, that we can still support domain-specific constructs [12]. We use metamodels to define the domains and our delta metamodel can be parametrized, so we support the seamless integration of domain-specific aspects into the Deltachain, going beyond traditional version control systems [23].

3 Approach

The Deltachain approach is built combining four contributions, which we outline in this section [18, 19]. The semantics of deltas deal with the way how to represent change in a meaningful way. Meaningful refers to the discussion in the introduction about DBR, so how well a delta represents the specifics of a domain, a task, or anything, that is of particular importance for a use case. We provide that by employing metamodels and a metamodel-independent but parametrizable delta metamodel. The Deltachain itself must be described and the means to derive an encrypted Deltachain and ways to store it have to be defined, being our second work package. The third work package is the derivation of encryption rules on the delta level from the access control rules, which are defined on the artifact level. The fourth work package is the integration of all prior work packages into a framework to support intellectual-property-preserving delta-based artifact sharing. Summarizing, we highlight selected benefits of our approach.

WP1: Semantics of Deltas. For the semantics of deltas we conducted a scoping literature review [16] to answer the research questions of how change is modeled, for which purpose it is modeled, and to what extent atomic and composite changes can be modeled. We also investigated the specific purpose of enhancing traceability for engineering reviews annotating composite changes in [6]. **Current State:** This work package is finished and published. **Contribution:** We investigated different approaches on how to model change and compared them in regard to the purposes they serve.

WP2: Delta Encryption and Storage. We proposed a Deltachain metamodel, encryption methods, and storage methods for the Deltachain in [17]. The delta metamodel in [17] is parametrized with a Git-specific metamodel defining files, lines in files, and blob files, enabling us to use Github⁵ repositories and their content as case studies for the evaluation of our approach. **Current State:** This work package is finished and published, though some completions for the complete prototype remain open. **Contribution:** The contribution of this WP is the definition of a metamodel for the Deltachain, the encryption methods used and how the Deltachain is handled, both in its encrypted and unencrypted form.

WP3: Derivation of Encryption Rules. We proposed the DEBAC approach [17] to derive encryption rules defined on deltas from access control rules, defined on artifact states, i.e., to grant or not grant read or write access to specific parts of the artifact. The aim of the DEBAC approach is to be able to use existing access control

²<https://github.com/AGWA/git-crypt>, accessed 11.11.2025

³<https://github.com/shadowhand/Git-encrypt>, accessed 11.11.2025

⁴https://dsis.kastel.kit.edu/staff_ralf_reussner.php, accessed 12.11.2025

⁵<https://github.com>, accessed 11.11.2025

system information and, by applying the Deltachain approach, additionally gain the benefit of being able to use untrusted third-party hardware. **Current State:** This work package is finished and published. **Contribution:** The contribution of this WP is the DEBAC approach, which enables the derivation of encryption rules, as well as their application to both encrypt and decrypt artifact parts in order to store or retrieve them from a Deltachain. The generated encryption rules define symmetric keys for the read access and mechanisms using asymmetric encryption to also restrict write access to the artifact stored on the Deltachain.

WP4: Framework. The framework integrates all components, providing a usable prototype to share intellectual property on untrusted hardware with other entities in an intellectual-property-preserving manner. **Current State:** This work package is conceptually finished but not published yet. Additionally, several aspects of the prototypical implementation are still lacking. **Contribution:** The contribution of this WP is a conceptual framework as well as a prototypical implementation allowing the use of a Deltachain in cross-organizational collaboration.

Benefits. Our approach supports incremental development without explicit interface sharing. Stakeholders can work on the latest artifact and interface versions by deriving them from the encrypted Deltachain, choosing when to publish updates. The Deltachain can be shared securely and freely via cloud services (e.g., AWS) since all deltas are encrypted. The ability to store any aspect of the system on the Deltachain creates a complete logical view of the system, though access remains distributed across organizations to protect intellectual property. Access control is decentralized: organizations grant keys for specific models or parts, enabling heterogeneous privileges without a central authority. While encryption introduces performance overhead and limits operations like branching or merging on the Deltachain itself, its primary role is secure cross-organization sharing without relying on commonly trusted hardware.

4 Planned Evaluation

Our evaluation will focus on validating the Deltachain approach and its ability to enable secure, incremental, cross-organizational development. We will assess (1) performance overhead from encryption, (2) usability of the framework for incremental development, and (3) effectiveness of IP protection in cross-organizational scenarios, once the prototypical implementation has been completed. We will also analyze scalability by measuring how the Deltachain performs with increasing numbers of deltas and stakeholders. Case studies will be derived from GitHub repositories and simulated multi-party collaboration, using commits and collaborators as heuristic.

Acknowledgments

This work was funded by the Topic Engineering Secure Systems of the Helmholtz Association (HGF) and supported by KASTEL Security Research Labs, Karlsruhe and supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - SFB 1608 - 501798263.

References

- [1] Daniel Adkins, Archita Agarwal, Seny Kamara, and Tarik Moataz. 2020. Encrypted blockchain databases. In *Proceedings of the 2nd ACM Conference on Advances in*

- Financial Technologies*. 241–254.
- [2] Francesco Basciani, Alfonso Pierantonio, Ludovico Iovino, et al. 2015. Model repositories: Will they become reality? A position statement. In *Proceedings of the 3rd International Workshop on Model-Driven Engineering on and for the Cloud 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2015), Ottawa, Canada, September 29, 2015*, Vol. 1563. CEUR-WS, 37–42.
- [3] Francesco Basciani, Alfonso Pierantonio, Ludovico Iovino, et al. 2015. Model repositories: Will they become reality? A position statement. In *MODELS (satellite events)*, Vol. 1563. CEUR-WS, Aachen, 37–42.
- [4] Erik Burger and Boris Gruschko. 2010. A change metamodel for the evolution of MOF-based metamodels. In *Modellierung 2010. Gesellschaft für Informatik eV*, 285–300.
- [5] XT Cai, Sheng Wang, Xin Lu, and WD Li. 2017. An encryption approach for product assembly models. *Advanced Engineering Informatics* 33 (2017), 374–387.
- [6] Johan Cederbladh, Eduard Kamburjan, David A Manrique-Negrin, Rakshit Mittal, and Thomas Weber. 2025. Traceability Support for Engineering Reviews of Horizontal Model Evolution. *Systems Engineering* (2025), e70001.
- [7] George A Gionis, Christoph Schroth, and Till Janner. 2011. Advancing interoperability for agile cross-organisational collaboration: a rule-based approach. In *Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business*. IGI Global, Hershey, PA, USA, 238–253.
- [8] Amer A Hijazi, Srinath Perera, Ali M Al-Ashwal, and Rodrigo Neves Calheiros. 2019. Enabling a single source of truth through BIM and blockchain integration. In *International Conference on Innovation, Technology, Enterprise and Entrepreneurship (ICITEE)*. Applied Science University Bahrain, Bahrain, 385–393.
- [9] Heiko Klare, Max E Kramer, Michael Langhammer, Dominik Werle, Erik Burger, and Ralf Reussner. 2021. Enabling consistency in view-based system development—the vitruvius approach. *Journal of Systems and Software* 171 (2021).
- [10] Ravi S Sandhu and Pierangela Samarati. 1994. Access control: principle and practice. *IEEE communications magazine* 32, 9 (1994), 40–48.
- [11] Russell G Shirey, Kenneth M Hopkinson, Kyle E Stewart, Douglas D Hodson, and Brett J Borghetti. 2015. Analysis of implementations to secure git for use as an encrypted distributed version control system. In *2015 48th Hawaii International Conference on System Sciences*. IEEE, 5310–5319.
- [12] Thomas Stahl, Markus Völter, and Krzysztof Czarnecki. 2006. *Model-driven software development: technology, engineering, management*. John Wiley & Sons, Inc., Hoboken, NJ, United States.
- [13] Dave Steinberg, Frank Budinsky, Ed Merks, and Marcelo Paternostro. 2008. *EMF: eclipse modeling framework*. Pearson Education.
- [14] Soyoung Sung, Yanghoon Kim, and Hangbae Chang. 2018. Improving collaboration between large and small-medium enterprises in automobile production. *Enterprise Information Systems* 12, 1 (2018), 19–35.
- [15] Ali Sunyaev. 2020. Distributed ledger technology. *Internet computing: Principles of distributed systems and emerging internet-based technologies* (2020), 265–299.
- [16] Thomas Weber, Johan Cederbladh, Sebastian Weber, Arne Lange, Antonio Cicchetti, and Ralf Reussner. 2025. How and Why is Change Modeled? – A Scoping Literature Review. In *MODELS-C*.
- [17] Thomas Weber, Katrin Quellmalz, Laurin Benz, Sebastian Weber, Erik Burger, and Ralf Reussner. 2025. Towards the Derivation of Encryption Rules for Encryption-Based Access Control for Delta-Based Model Representations – The DEBAC Approach. In *Second International Workshop on Model Management (MoM), co-located with the 28th International Conference on Model Driven Engineering Languages and Systems (MODELS 2025), Grand Rapids, MI, 5th-10th October 2025*. Institute of Electrical and Electronics Engineers (IEEE), 46.23.01; LK 01.
- [18] Thomas Weber and Sebastian Weber. 2024. Model Everything but with Intellectual Property Protection-The Deltachain Approach. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. 49–56.
- [19] Thomas Weber and Sebastian Weber. 2024. Towards a Single Source of Truth with a Freely Shareable Deltachain. In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 92–94.
- [20] Jan Willem Wittler, Timur Saglam, and Thomas Kühn. 2023. Evaluating model differencing for the consistency preservation of state-based views. *Journal of Object Technology* 22 (2023), 1–14.
- [21] Wenhao Xu, Hui Ma, Zishuai Song, Jianhao Li, and Rui Zhang. 2023. Gringotts: An Encrypted Version Control System with Less Trust on Servers. *IEEE Transactions on Dependable and Secure Computing* (2023).
- [22] Alfa Yohannis, Dimitris Kolovos, and Fiona Polack. 2017. Turning models inside out. In *CEUR Workshop Proceedings 1403*. The University of York, York, 430–434.
- [23] Nazatul Nurlisa Zolkifli, Amir Ngah, and Aziz Deraman. 2018. Version control system: A review. *Procedia Computer Science* 135 (2018), 408–415.