prostep ivip

Recommendation

Collaborative Digital Twin (CDT)

prostep ivip Recommendation

Collaborative Digital Twin (CDT)

Recommendation PSI 32

Version 1.0

Management Summary

The prostep ivip project group Collaborative Digital Twin (CDT) addressed one of the most pressing challenges in modern engineering: enabling secure, efficient, and interoperable collaboration across organizational boundaries. Increasing product complexity – particularly in automotive, aerospace, and industrial sectors – requires a shift from document-based, asynchronous processes to model-based, data-centric and synchronous approaches. While Digital Twins (DTs) have emerged as a major enabler for this transformation, a key aspect – the collaborative development of Digital Twins – remains insufficiently addressed.

The CDT initiative, launched by prostep ivip in cooperation with the German Association of the Automotive Industry (VDA), provides a comprehensive framework for cross-enterprise collaboration based on Digital Twins. More than 80 experts from 26 companies and 9 research institutions contributed to this effort, ensuring broad industrial relevance. The project group thus bundles the knowledge and activities for the collaborative, cross-company development of DTs. The project delivers four major outcomes:

- **Semantic Foundation** – A layered ontology framework and aspect models establish a common language for engineering data, enabling semantic interoperability across domains and IT systems. This approach aligns with initiatives such as Catena-X and supports lifecycle-wide consistency.
- **IT Architecture Blueprint** – CDT defines a modular, federated architecture for Digital Twin collaboration. It emphasizes data sovereignty, secure identity and contract management, and integration into existing IT landscapes. The architecture leverages open standards and supports interoperability with Gaia-X-compliant data spaces.
- **Operationalization through User Journeys** – Practical workflows illustrate how CDT concepts can be applied in real-world scenarios, from requirements management and system design to simulation, integration, and manufacturing engineering. These user journeys were validated through a multi-partner demonstrator, proving the feasibility of synchronous, point-based data exchange without traditional file transfers.
- **Cross-Domain Configuration and Variant Management** – By introducing structured methodologies for feature modeling, variation point identification, and configuration rule definition, CDT supports scalable product line engineering and traceability across the entire lifecycle, which is a critical capability for managing product complexity in collaborative environments.

Implications for industry are significant: CDT enables reduced engineering effort, faster decision-making, and improved data consistency, translating into up to 10% shorter development cycles and lower costs. Successful implementation, however, requires a holistic transformation across five dimensions – mindset, relationships, methods, technology, and information. Companies must adopt trust-based collaborative models, agile practices, and standardized data exchange principles to fully leverage the benefits of Digital Twins.

Looking ahead, CDT lays the foundation for Live Collaboration in Engineering (LCE), the next phase of cross-enterprise digitalization. By harmonizing standards, fostering interoperability, and promoting open architectures, CDT provides a clear roadmap for industry to transition from fragmented processes to a connected, collaborative ecosystem.

"With the CDT proof of concept, we have reached a true milestone: Instead of sending entire files back and forth, we can now collaborate directly on the actual data and individual values. This reduces complexity, avoids interface losses and unnecessary data overload. For me, this is a quantum leap in collaboration – efficient, transparent, and future-oriented." – ***Dr. Sebastian Handschuh, Mercedes-Benz AG***

“The results from the Implementors Lab have shown how quickly and how significantly we can advance cross-company digital collaboration with CDT and prostep ivip. What used to take days or weeks now happens in seconds – and across different IT environments. This is not just technological progress; it opens entirely new possibilities for agile, connected development. We must now turn these tangible opportunities into reality.” – ***Peter Gerber, Schaeffler Technologies AG & Co. KG***

Disclaimer

prostep ivip Recommendations (CDT Recommendations) are recommendations that are available for anyone to use. Anyone using these recommendations is responsible for ensuring that they are used correctly.

This CDT Recommendation reflects the prevailing state-of-the-art at the time of publication. Anyone using CDT Recommendations must assume responsibility for his or her actions and acts at their own risk. The prostep ivip Association and the parties involved in drawing up the CDT Recommendation assume no liability whatsoever.

The prostep ivip Association encourages anyone encountering an error or the possibility of an incorrect interpretation when using the CDT Recommendation to contact the prostep ivip Association (psi-issues@prostep.org) immediately so that errors can be rectified.

Copyright

I. All rights to this CDT Recommendation, in particular the copyright, rights of use, and sale such as the right to duplicate, distribute or publish this CDT Recommendation remain exclusively with the prostep ivip Association and its members.

II. This CDT Recommendation may be duplicated and distributed unchanged, for instance for use in the context of creating software or services.

III. It is not permitted to change or edit this CDT Recommendation.

IV. A suitable notice indicating the copyright owner and the restrictions on use must always appear.

Figures

Tables

1 Introduction

Digital Twins are one of the key enablers to cope with the increasing complexity of modern product development. The key to the success of DTs is their collaborative and cross-company creation. In this chapter, the initial situation and motivation for the project group Collaborative Digital Twins is outlined. The goal and the set-up of the project group are described as well as the structure of this recommendation.

1.1 Initial Situation & Motivation

In recent years, the landscape of product development and engineering has undergone a profound transformation. The increasing complexity of modern products – especially in sectors like automotive, aerospace, and industrial manufacturing – demands a new level of collaboration across companies, domains, and disciplines. Products today are no longer purely mechanical; they are cyber-physical systems that integrate electronics, software, and data-driven services. As a result, engineering processes must evolve from isolated, document-based workflows to interconnected, model-based, and data-centric approaches.

However, the current state of digital engineering is characterized by asynchronous data exchange, manual data transformation, and limited semantic interoperability. Each stakeholder – OEMs, Tier 1 and Tier 2 suppliers, service providers – often works within its own IT environment, using proprietary tools and data formats. This leads to significant inefficiencies, such as:

- Redundant data entry and transformation efforts
- Delays in communication and decision-making
- Inconsistent data baselines and versioning
- Limited traceability and accountability across the product lifecycle

These challenges are particularly evident in collaborative engineering scenarios, where multiple partners must co-develop complex systems. One promising approach to addressing with these challenges is Digital Twins (DTs). Within this project, DTs are defined as…

"… a virtual identifiable representation of a real-world entity (product, concept, process), which is bidirectionally connected to it at a specified frequency and fidelity."

The concept stems from an approach presented in the early 2000s by Grieves & Vickers[1] and has generated significant industry interest. In 2018, according to Gartner Research[2], DTs were considered to be at the peak of inflated expectations. The value proposition is based on manifold potentials and benefits. Especially in engineering, Digital Twins offer virtual prototypes, reducing cost and time and improving customer integration, leading to increased functionality.

While individual companies have made significant advances in developing internal digital twin solutions, these efforts often lack interoperability and are constrained by proprietary data models, isolated IT infrastructures, and inconsistent terminologies. Existing initiatives such as Catena-X, Tractus-X, and Eclipse Dataspace Connector (EDC) have made important contributions to data sharing and interoperability, yet a comprehensive framework for synchronous, cross-enterprise engineering collaboration based on digital twins was still missing.

Within the prostep ivip Association, multiple project groups had been working on topics related to digital twins, including systems engineering, model-based development, and data standards. However, these efforts were largely siloed, and a consolidated position or strategy on collaborative

[1] Grieves & Vickers, 2017 - Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. https://doi.org/10.1007/978-3-319-38756-7_4
[2] https://www.gartner.com/en/documents/3867164

digital twins had not yet been established. This fragmentation limited the ability of member companies to align their digital transformation strategies and to leverage synergies across domains and industries.

1.2 Goal Definition

In order to enable the envisioned cross-enterprise collaboration based on DTs, companies need to undergo a transformation in five areas – (1) mindset, (2) relationships, (3) methods, (4) technology, and (5) information (see Figure 1).

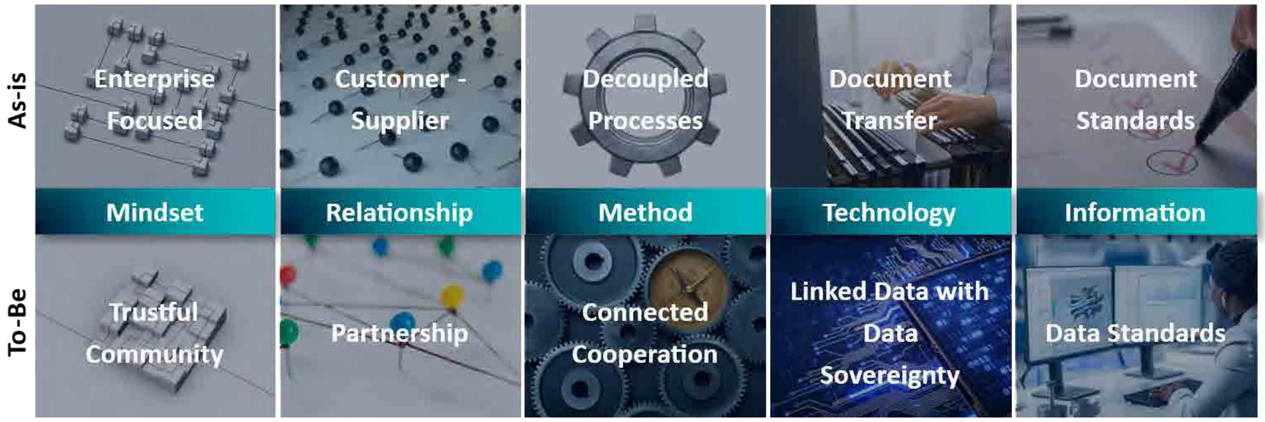


Figure 1: Dimensions of Change in the Context of Collaborative Digital Twins. Companies need to address all five categories for successful implementation.

Current engineering collaboration is characterized by an “As-Is” state that is enterprise-focused, based on customer-supplier hierarchies, and reliant on decoupled processes and document-based data exchange. This approach leads to inefficiencies, delays, and limited transparency. To unlock the full potential of digitalization, the industry must transition to a “To-Be” state that emphasizes:

Mindset: From isolated enterprise thinking to a trustful community built on openness and shared goals.

Relationship: From transactional customer-supplier interactions to strategic partnerships fostering co-innovation.

Method: From decoupled processes to connected cooperation, enabling synchronized workflows across organizations.

Technology: From document transfer to linked data with data sovereignty, ensuring secure, real-time exchange of engineering information.

Information: From fragmented document standards to harmonized data standards, supporting semantic interoperability and lifecycle consistency.

This holistic transformation requires consistent implementation and change across all levels, as highlighted in the CDT framework. To leverage this change, the project group Collaborative Digital Twins (CDT) was established.

1.3 Set-up of the Project Group Collaborative Digital Twins

The CDT project group is a joint initiative of the prostep ivip Association and the German Association of the Automotive Industry (VDA). The project group was established in 2022 with the goal of developing a definition and conceptual framework for Digital Twins from the perspective of prostep ivip,and consolidating ongoing activities. Despite the relevance and high level of attention in publications and within the community, there has been no clear positioning of the prostep ivip association on the topic of Digital Twins. In addition, several existing project groups are working on topics related to Digital Twins or producing results that can be leveraged in building Digital Twins.

CDT therefore serves as a hub, bringing together the relevant discussions within prostep ivip on Digital Twins from methodological, technical, and legal perspectives, enabling member companies to take the path toward collaborative Digital Twins.

The project group brings together a diverse consortium of stakeholders, including industrial users, implementors, service providers, and academic partners. In total, more than 80 participants from 26 companies and 9 research institutions have contributed to the work, ensuring a broad representation of perspectives and expertise. This multi-disciplinary composition reflects the complexity of the challenge: enabling collaborative digital twins requires not only technical solutions but also harmonized processes, governance models, and semantic standards.

The CDT project group was organized into six work packages that together covered all essential aspects of collaborative digital twin development. **Work Package 0** managed organizational tasks and strategic networking. **Work Package 1** established the semantic foundation by defining ontologies and aspect models for interoperability. **Work Package 2** focused on practical collaboration scenarios, developing user journeys that demonstrate cross-enterprise engineering processes. **Work Package 3** delivered blueprints for digital twin-ready IT architectures, providing scalable and vendor-neutral solutions. **Work Package 4** created interfaces to related projects and initiatives, promoting harmonization and synergy. Work package 5 “Human Competencies of Competence Holders” was put on hold due to missing resources and commitment. Finally, **Work Package 6** addressed configuration and variant management, enabling collaborative handling of complex product structures.

A key factor in operationalizing CDT is the institutional interplay - CDT does not exist in isolation; it builds upon and aligns with existing initiatives and standards (see Figure 2). While prostep ivip and the “Verband der Automobilindustrie” (VDA) provide strategic recommendations based on the needs of the companies, CDT developed use cases and requirements on how to implement these recommendations in practice. Catena-X aligns with the use cases and develops data standards[3], which are implemented in software components provided e.g. by Tractus-X or Catena next. Organizations such as Cofinity-X are responsible for the operationalization of the developed approaches, which will then be implemented and tested by the companies. The lessons learned by industry are then fed back to the associations for further development and projects.

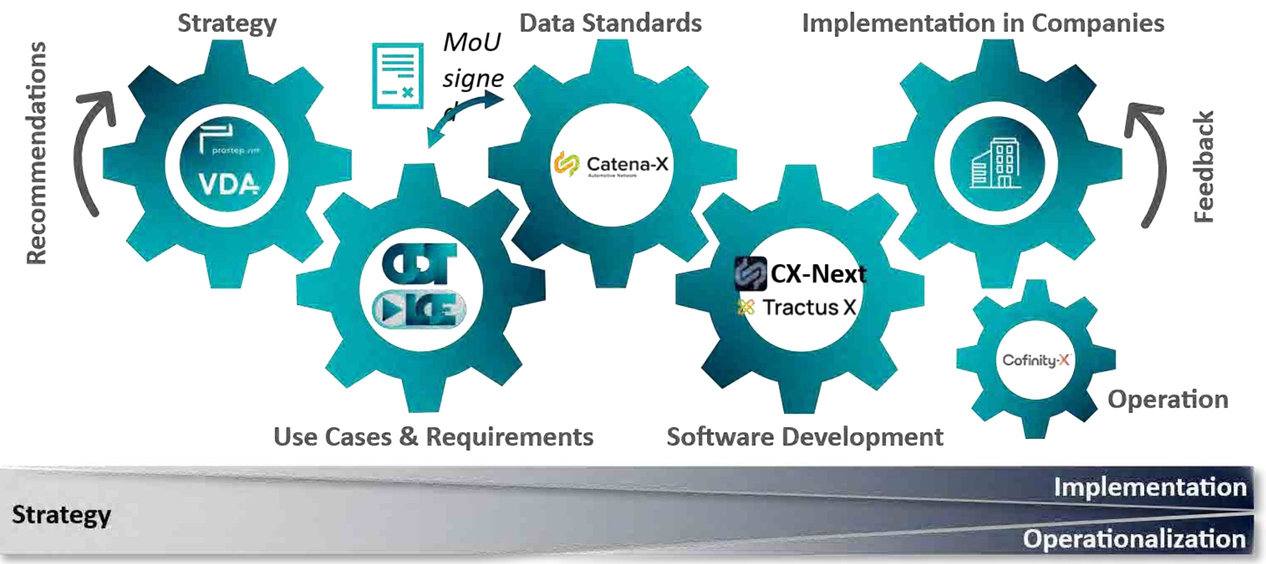


Figure 2: Interplay of Organizations in the Context of the Project Group “Collaborative Digital Twins”.

[3] In 2025 prostep ivip and the Catena-X association have signed a memorandum of understanding. All results are well aligned between the two associations and will be collaboratively further developed in future.

1.4 Structure of the Recommendation

This recommendation is structured in seven chapters starting with a Management Summary, providing a concise overview of the project's objectives, key results, and strategic implications. This is followed by the Introduction (Chapter 1), which sets the context by outlining the initial situation and motivation for the project. It explains why collaborative digital twins are essential for modern engineering and describes the working environment in which CDT was conceived.

The second chapter (0), State of the Art – Current Challenges in Collaborative Product Development & Existing Initiatives, offers a comprehensive analysis of existing practices and technologies. It examines the limitations of current approaches, reviews relevant initiatives such as Catena-X, and interprets their implications for CDT.

Chapter three (0), Terminology & Ontologies, establishes the semantic foundation for collaborative digital twins. It defines key terms, introduces aspect models, and presents the CDT ontology framework, which enables interoperability across domains and systems.

The fourth chapter (0), Blueprint for DT-ready IT Architectures, translates conceptual work into technical guidance. It describes architecture principles, building blocks, and interaction flows, culminating in implementation strategies for CDT-compliant infrastructures.

Chapter five (5), User Journeys for Collaborative Digital Twins, illustrates practical application scenarios. These user journeys demonstrate how CDT concepts can be implemented in real-world engineering processes, from requirements management to manufacturing engineering.

The sixth chapter (6), Cross-Domain Configuration & Variant Management, addresses the complexity of managing product variants in collaborative environments. It provides methods and demonstrator results that support consistent configuration across multiple partners. While chapter 0 captures general Digital Twin related terminology, terminology specific to configuration and variant management is described here.

Finally, the report concludes with Implications for Industry, Summary & Outlook (0). This chapter synthesizes the findings, discusses their relevance for different stakeholders, and outlines next steps, including the transition toward the follow-up initiative, Live Collaboration in Engineering (LCE).

2 State of the Art – Current Challenges in Collaborative Product Development & Existing Initiatives

Collaborative product development has become a central driver of innovation, competitiveness, and knowledge sharing in modern industries. Increasing global connectivity and the need for cross-disciplinary expertise make collaboration essential across value chains. Digital Twins (DTs) play a pivotal role in this development, serving as both a technological foundation and an enabler for deeper cooperation across organizations.

To capture the current state of industry practice, more than 20 companies across aerospace, automotive, electrical and mechanical engineering, nuclear energy, and consortia structures were surveyed and interviewed. The focus was on understanding the maturity of collaborative product development, the role of DTs in this context, and existing barriers to adoption.

To ensure that the survey results were directly relevant to the overarching project context, the selection of recipients was made with a clear focus on industries and stakeholders involved in collaborative product development with Digital Twins. The participants were identified primarily through established networks and existing project contacts, which ensured a high level of trust and willingness to provide detailed insights. By addressing “warm contacts” in companies and consortia already engaged in related initiatives, the survey could capture informed perspectives while maintaining a strong link to the practical challenges faced in industry.

2.1 Assessment of the State of the Art

Digital Twins are already widely recognized as a fundamental basis for operations management and process integration. Their presence in daily engineering and production workflows also makes them a good indicator of an organization’s readiness for collaborative development.

To evaluate the maturity and current practices in collaborative product development using Digital Twins, a structured survey was designed and distributed among 59 companies across different industries, including aerospace, automotive, electrical and mechanical engineering, nuclear energy, and established industrial consortia.

The survey was deliberately designed with open-ended questions. This format enabled respondents to describe their practices, expectations, and challenges in their own words, thereby providing richer qualitative insights beyond simple multiple-choice metrics. The responses were later categorized and analyzed against expert definitions and reference frameworks, ensuring comparability across industries.

The survey addressed six thematic categories:

1. General Understanding
    a. Respondents were asked to define what a Digital Twin (DT) means in their organizational context and describe how this definition is currently applied.
    b. This provided a baseline of shared terminology and revealed potential differences in interpretation.
2. Scope of Use
    a. Questions focused on which aspects of DTs were prioritized (e.g., 3D models, maintenance documents, software integration, data models, or real-time synchronization).
    b. Participants were also asked in which lifecycle phases they apply DTs (engineering, production, operational/product twin, enterprise-level integration).
3. Context of Use

   a. This section explored how DTs are envisioned for deployment: internally, as customer-facing services, or as part of supply chain integration.
   b. Respondents were also asked whether collaboration between DTs was part of their scope and what objectives such collaboration would serve.

4. Planned Implementation
   a. This part addressed the use of data-sharing infrastructures such as Catena-X or Manufacturing-X.
   b. It also assessed whether companies planned to adopt standards like the Asset Administration Shell (AAS), OPC UA, or overarching data exchange platforms, or whether they favored proprietary IT architectures.
5. Intellectual Property Protection
   a. Respondents described their provisions for IP protection in DT projects, including security measures and governance mechanisms to safeguard sensitive data.
6. Configuration Management
   a. Finally, companies were asked whether and how they incorporate configuration management in DT projects.
   b. Follow-up questions clarified whether they strive for a complete (100%) DT implementation or only partial adoption, and what decision processes guide these choices.

By structuring the survey along these categories, it was possible to map both the breadth of Digital Twin adoption and the depth of collaborative practices across industries using open-ended questions. The approach also highlighted key areas where alignment exists (e.g., general definition of DTs) as well as where fragmentation persists (e.g., standards adoption, data sharing, IP protection).

The purpose of the survey was to obtain a broad overview based on open-ended questions.. These open-ended responses provide a broader understanding during the analysis process and can be used for different analytical purposes.

Figure 3 shows the participating companies and how the survey was conducted in two iterations. . The responses were analyzed and then summarized for each industry and per question..

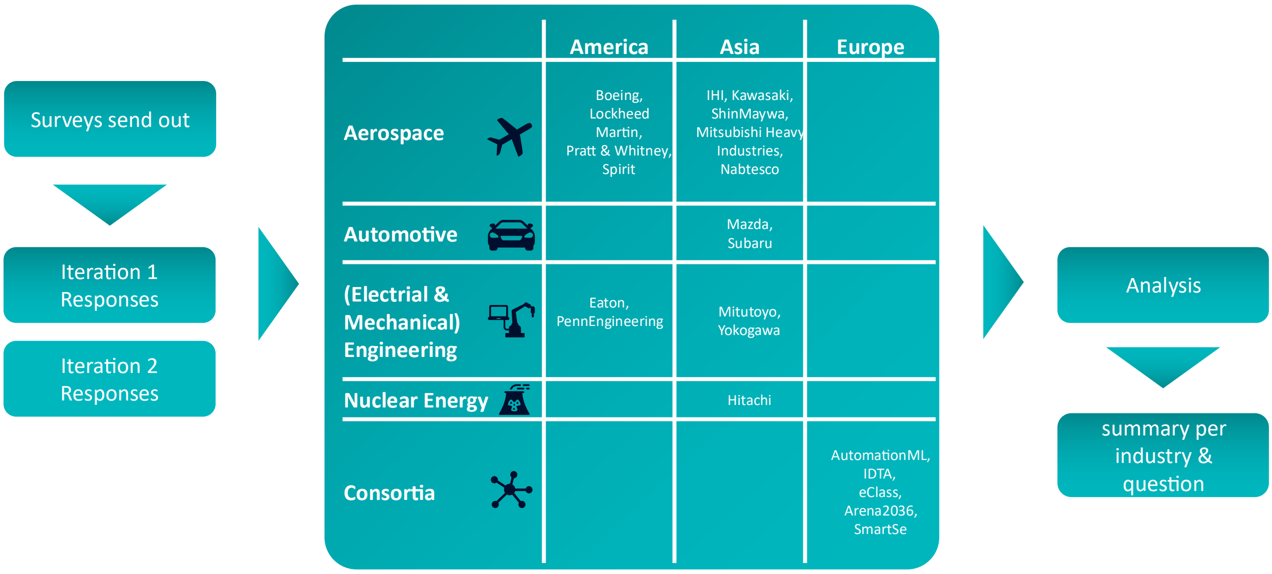


Figure 3: Survey Workflow

2.2 Analysis of the State of the Art

The open-ended responses from the survey were systematically grouped into seven thematic clusters to enable coherent interpretation and comparison of patterns across different industries. While the initial clustering was based on the existing survey structure, it was refined during analysis to better capture recurring topics and commonalities, ensuring the qualitative richness of individual company perspectives was preserved.

The seven clusters cover: General (DT definitions and current application), Adoption (maturity of use across lifecycle phases), Aspects of DTs (technical dimensions like real-time data and 3D models), Context of Use (deployment strategies and collaboration objectives), Data Sharing and Collaboration (internal/external exchange and data spaces), Configuration Management (integration of CM practices and selective vs. complete DT implementation), and IP Protection and Vision (governance, security, and strategic outlook).

This seven-cluster framework established a coherent structure for the analysis, making it possible to identify both industry-specific differences and overarching challenges, such as gaps in standardization and interoperability. The company answers are interpreted within the context of alignment in the respective cluster—meaning their statements are compared with general definitions from experts and third parties—which is visualized through "alignment colors" (

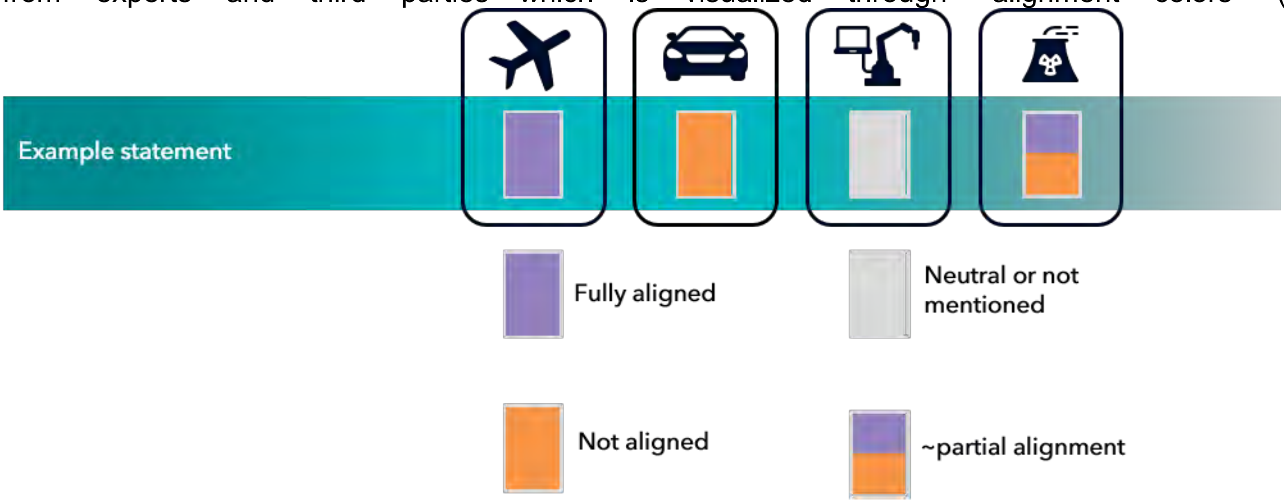


Figure 4) for easy interpretation.

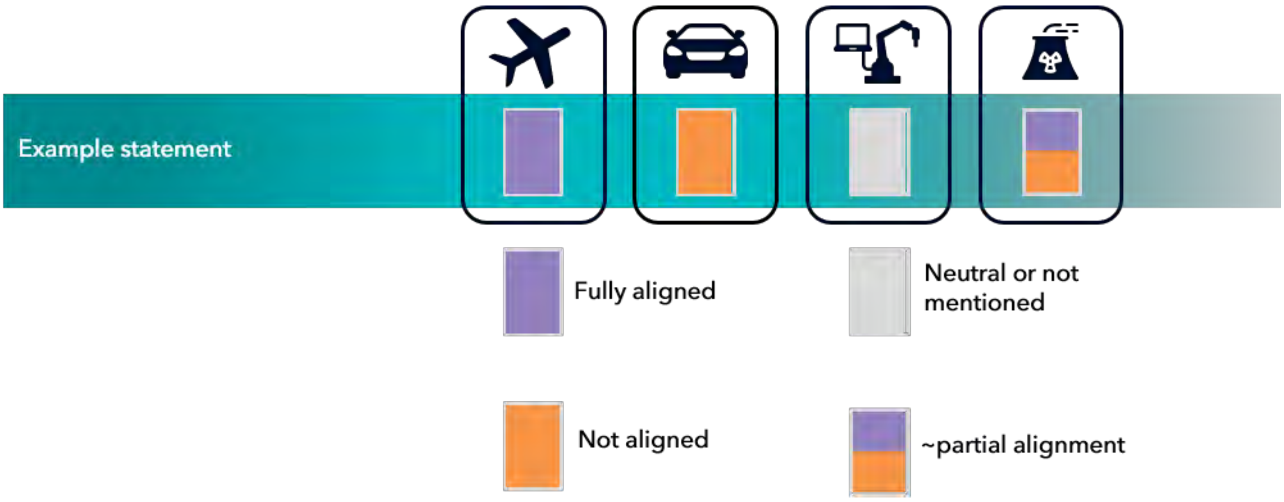


Figure 4: Survey Evaluation Schema

2.2.1 General Understanding

The analysis of the survey results shows that there is already a broad agreement across industries on the fundamental definition of Digital Twins. Most companies view DTs as digital representations of physical assets, systems, or processes that support data integration and decision-making. This shared understanding provides a solid conceptual basis for collaboration. Nevertheless, practical application remains fragmented. In many cases, companies are still running pilot projects or isolated implementations. Established, large-scale use cases are rare, and integration levels differ considerably between sectors. This uneven picture highlights the importance of harmonized standards if the full potential of collaborative product development is to be realized. Figure 5 shows a detailed distribution of responses across questions and industries.

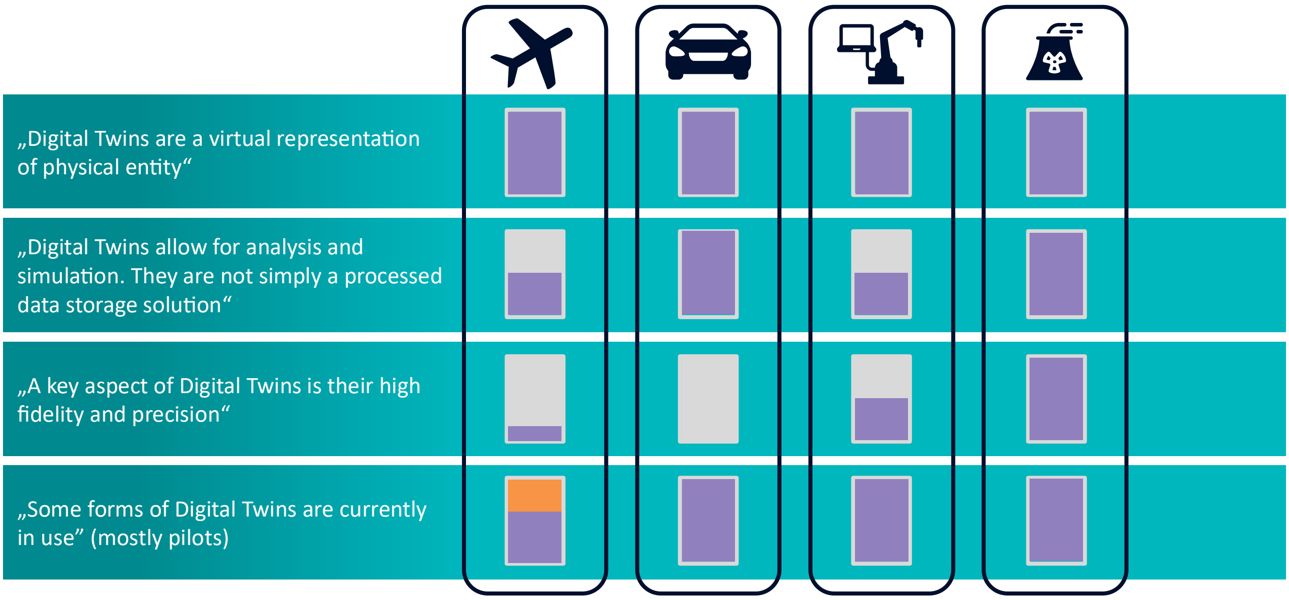


Figure 5: Evaluation of General Understanding

2.2.2 Adoption

With regard to adoption, most companies currently use Digital Twins internally, particularly in production and in design and engineering. These two lifecycle stages form the main entry points for DTs, especially in the automotive and engineering industries. By contrast, areas with high potential—such as maintenance, repair, overhaul, and supply chain integration—are only rarely addressed. This imbalance shows that the technology is established enough to be used in core operations, but that opportunities for cross-organizational collaboration are not yet being exploited. Many applications are still in the testing stage, which further delays the emergence of standardized solutions and large-scale rollouts. Figure 6 shows a detailed distribution of responses across questions and industries.

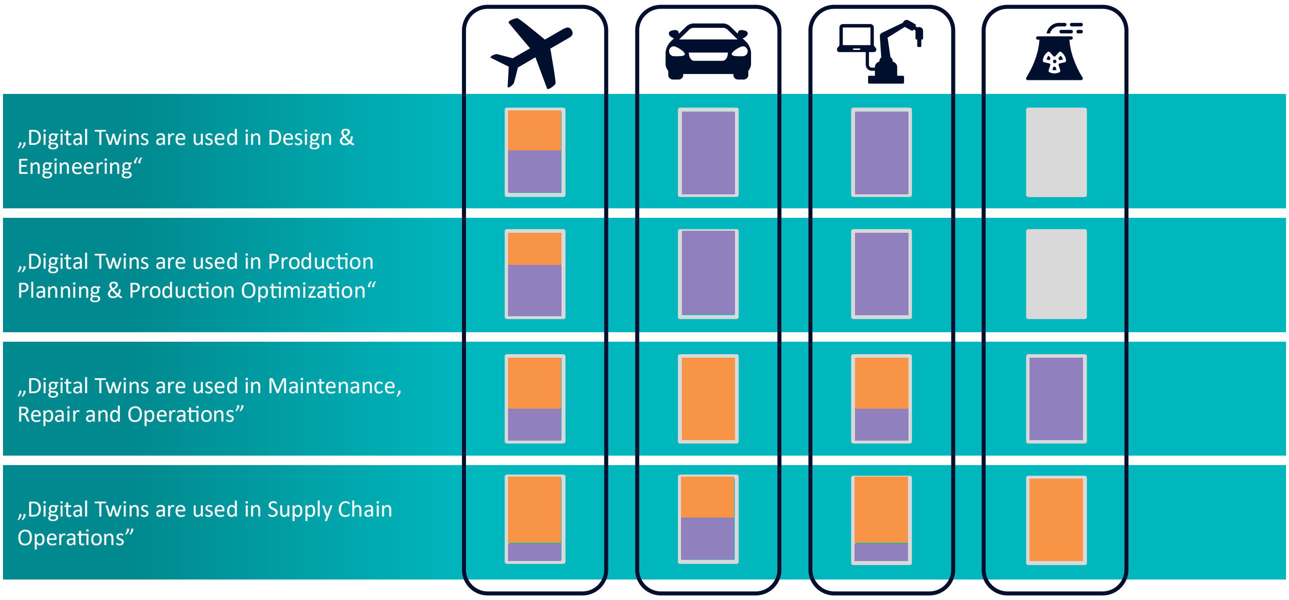


Figure 6: Evaluation of Adoption

2.2.3 Aspects of Digital Twins

The technological aspects of DTs are implemented in highly diverse ways, with no clear pattern that follows industry boundaries. Real-time data synchronization, software integration, 3D models, and data management practices are used heterogeneously, pointing to a generally low maturity of implementation. The automotive and engineering sectors appear to be ahead of others, demonstrating that the technological foundations are well defined, but the lack of common focus areas risks slowing down the development of branch-spanning collaboration. Figure 7 shows a detailed distribution of responses across questions and industries.

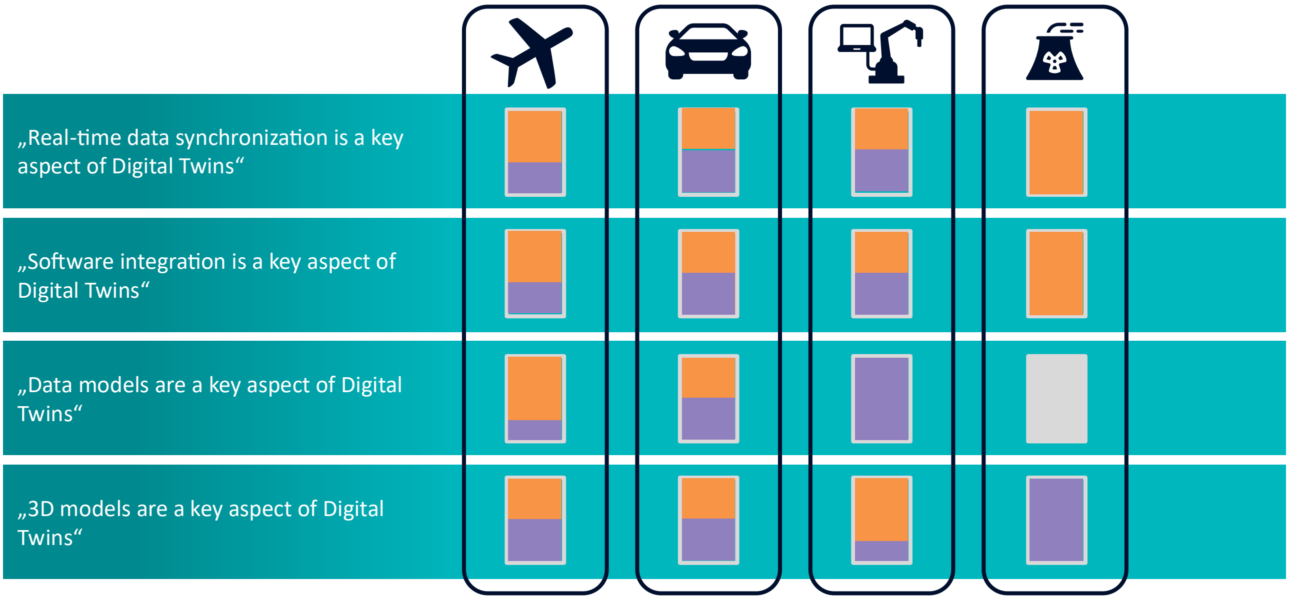


Figure 7: Evaluation of Aspects

2.2.4 Context of Use

The context in which Digital Twins are deployed also differs between industries. Engineering companies are among the first to commercialize DT-based services, offering them directly to customers. Aerospace and automotive firms concentrate more on integrating their suppliers, a reflection of the high safety standards and strong competitive pressures in these industries. In the nuclear sector, companies such as Hitachi face particularly strong restrictions, as security requirements limit the degree of openness and external collaboration. In most cases, collaboration between Digital Twins is not a central element of current projects, although where it is considered, it is typically linked to supply chain integration, lifecycle synchronization, and the joint development of products and services. Figure 8 shows a detailed distribution of responses across questions and industries.

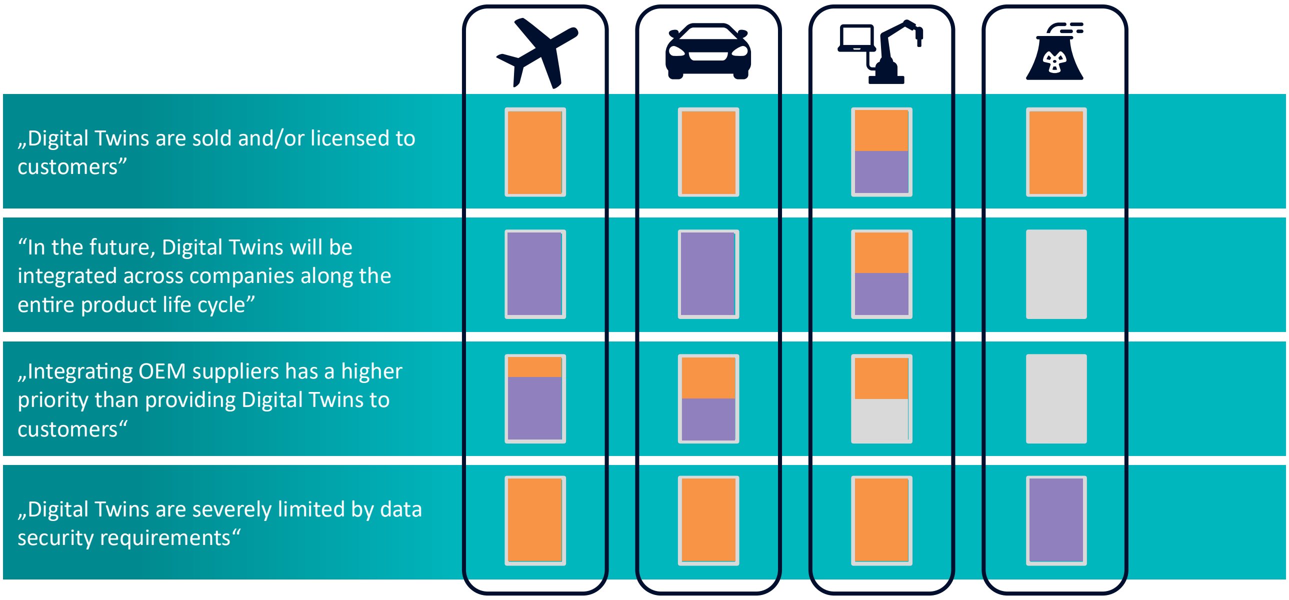


Figure 8: Evaluation of Context of Use

2.2.5 Data Sharing and Collaboration

Data sharing and collaboration practices remain limited overall. Data spaces such as Catena-X are not yet widely known among the surveyed companies, even though such infrastructures could play an important role in standardization and secure information exchange. Automotive manufacturers tend to rely on proprietary systems and maintain selective collaborations with a few trusted suppliers, while aerospace and engineering companies show greater interest in standardized solutions. Cross-company collaborations involving DTs already exist but are still rare and often confined to bilateral partnerships. Figure 9 shows a detailed distribution of responses across questions and industries.

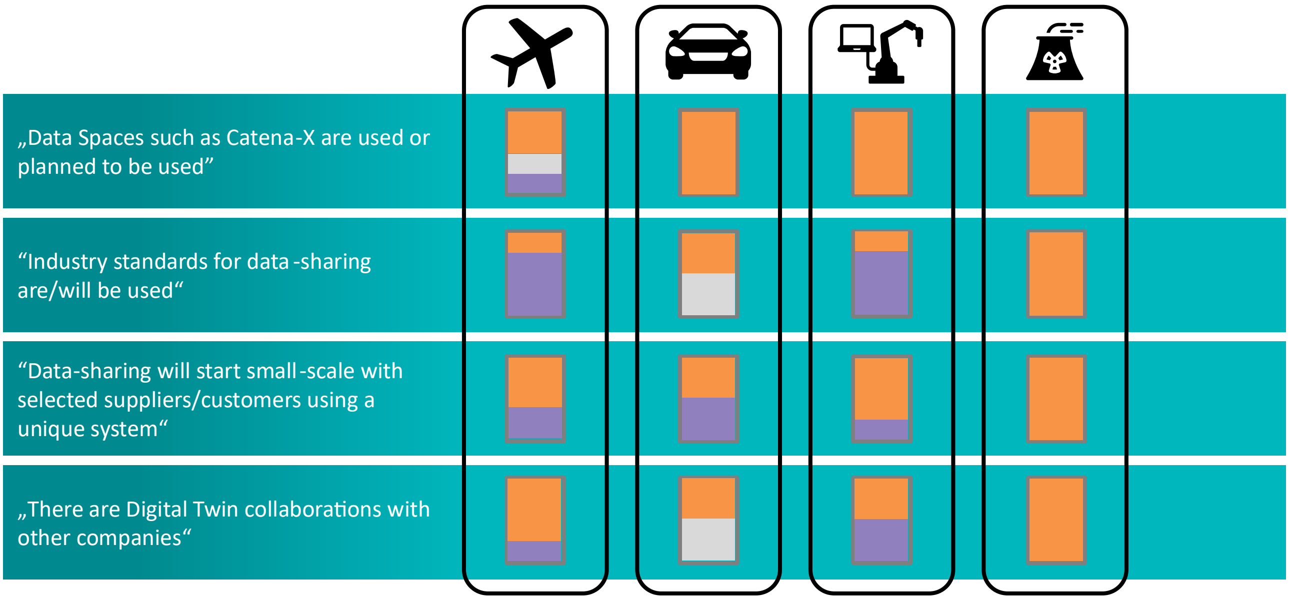


Figure 9: Evaluation of Data Sharing and Collaboration

2.2.6 Configuration Management

Configuration management is recognized across all industries as an essential element for Digital Twin collaboration, but its implementation is uneven. Some companies, such as those in the nuclear energy sector, must restrict themselves to software approved by regulatory authorities, in this case the IAEA. Others have introduced proprietary configuration management systems that are not interoperable with those of their partners. Only a minority of companies aim for complete digital twin implementation, while most adopt the technology selectively in areas where it provides the clearest value. This pragmatic approach helps manage resources but contributes to an overall heterogeneous maturity landscape. Figure 10 shows a detailed distribution of responses across questions and industries.

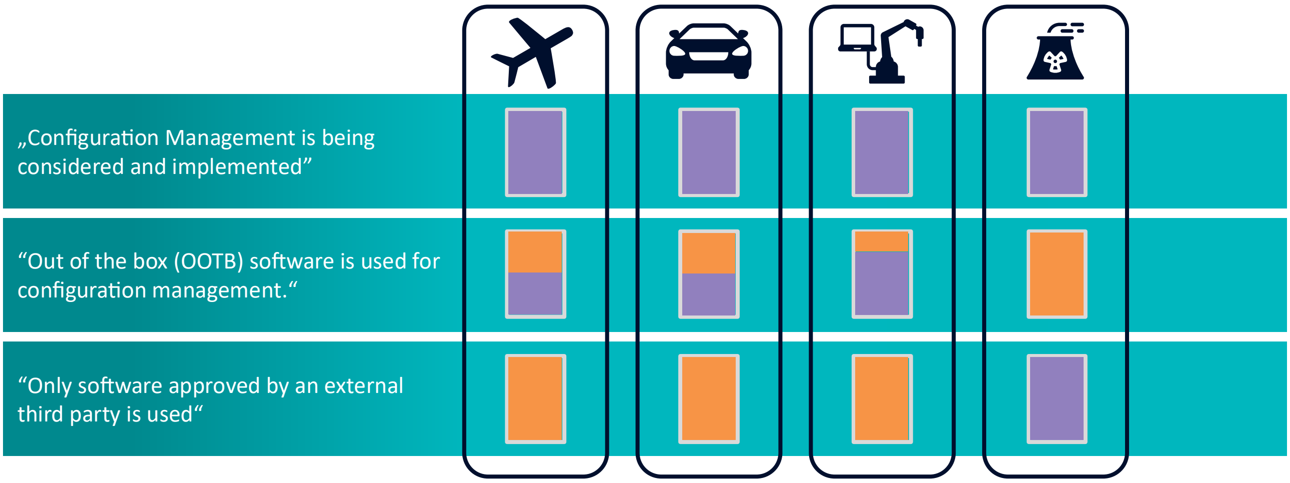


Figure 10: Evaluation of Configuration Management

2.2.7 IP Protection & Vision

Finally, intellectual property protection and strategic vision form an important but underdeveloped area. All respondents emphasized the need to safeguard IP, but most rely on conventional measures such as access controls, nondisclosure agreements, and patents. Few have developed specific protective frameworks for Digital Twins. While this creates a basic level of trust, it also limits the willingness to share sensitive data across organizational boundaries. In terms of vision, companies generally focus on use cases that promise tangible benefits within their own industry. Broader cross-sector perspectives are rare, indicating that the collaborative potential of DTs is not yet fully embedded in long-term strategic planning. Figure 11 shows a detailed distribution of responses across questions and industries.

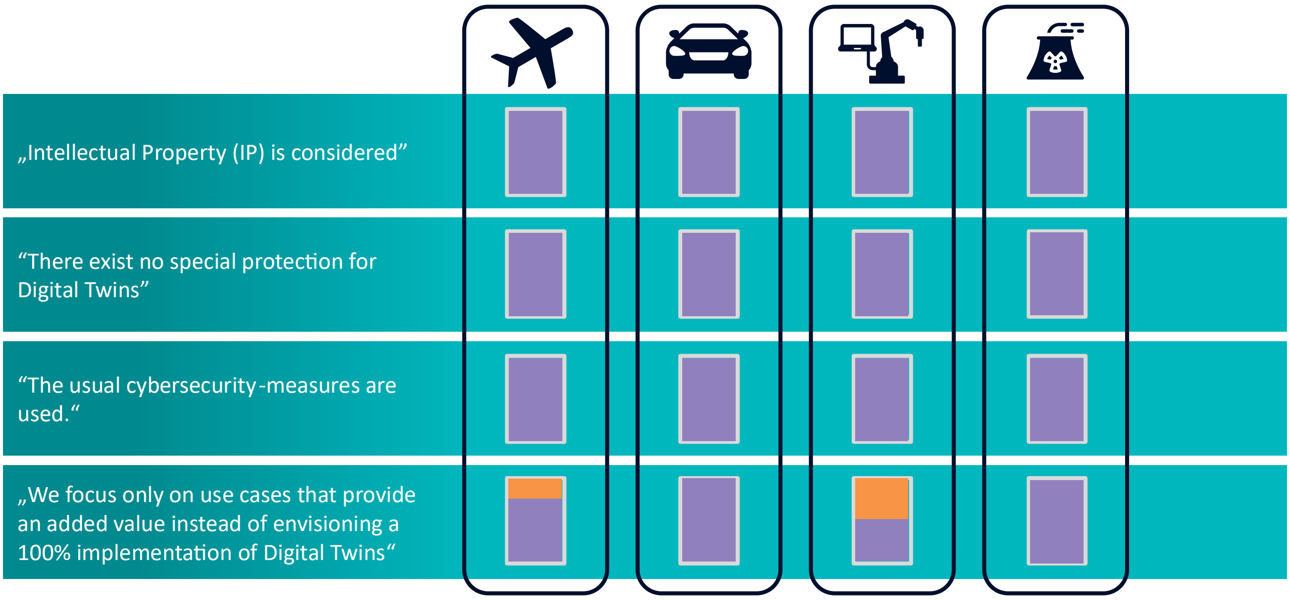


Figure 11: Evaluation of IP Protection and Vision

Overall, the analysis reveals a landscape of partial adoption and fragmented practices. While the conceptual basis of Digital Twins is widely shared, industries differ in how far they have advanced in applying and integrating the technology. Opportunities for collaboration exist, but they are hampered by uneven maturity, lack of standardization, and concerns about data protection and intellectual property.

2.3 Conclusion and Interpretation of the State of the Art

The survey results demonstrate that Digital Twins have become an established concept across industries, but their implementation and collaborative use remain highly uneven. While all companies share a common understanding of what a Digital Twin is, their approaches to adoption vary widely, reflecting sector-specific requirements, levels of maturity, and organizational priorities.

An important outcome of the survey is the limited awareness and adoption of collaborative data space initiatives, particularly those emerging in the European context such as Catena-X, Manufacturing-X, and related -X ecosystems. These initiatives are designed to establish secure, standardized, and interoperable infrastructures for data exchange across industries. Their aim is to enable companies to share Digital Twin information seamlessly, while maintaining control over access rights and protecting intellectual property.

The survey responses show that most companies are either unfamiliar with these initiatives or have not yet engaged with them in a structured way. This represents a critical gap, as -X initiatives directly address many of the challenges identified in the analysis: the lack of common standards, the need for trustworthy governance models, and the difficulty of scaling cross-company collaboration. Without broader adoption of such ecosystems, efforts to integrate Digital Twins across supply chains are likely to remain fragmented and isolated.

Consortia in Europe stand out as significantly more advanced in applying Digital Twins collaboratively. They make active use of data spaces and standardized frameworks, which enables them to connect partners and exchange data in structured ways. By contrast, private companies often follow more heterogeneous strategies that emphasize short-term return on investment. Their implementations tend to be isolated, designed to meet immediate internal needs rather than to support long-term collaboration. This divergence creates a split landscape in which consortia move towards standardized, interoperable ecosystems, while private industry remains fragmented. Figure 12 summarizes this.

Consortia

- **Heterogeneous use cases of all sizes**
  - sector specific use cases
  - industry-spanning simulation models for the whole supply chain
  - mostly consider all life-cycle phases
- **Emphasize on**
  - stronger connectivity between partners
  - enabling collaboration and integration between different companies' Digital Twins.
  - minimization of implementation hurdles
- Develop and work with **industry standards**
- **Data Spaces** are being integrated

vs.

Private Industry

- **Company-specific use cases of smaller size**
  - industry-specific solutions
  - focus on internal use
  - first collaborations between two partners
  - focus on individual life-cycle phases
- **Emphasize on**
  - Return on Investment (ROI)
  - pilots to identify most beneficial use cases
- ~50/50 distribution between **industry standards and tailored systems**
- Use of **companies' data storage**

Figure 12: Summary of Findings

A comparison of industries highlights clear differences in maturity. Engineering companies, the automotive sector, and established consortia already use Digital Twins across multiple stages of the product lifecycle, ranging from design and production to operational applications. By contrast, aerospace and nuclear energy companies still show considerable untapped potential. In aerospace, the emphasis on supplier integration has not yet translated into broader collaborative ecosystems, while in the nuclear sector strict regulatory and security requirements continue to restrict the scope of adoption. Figure 13 summarizes this.

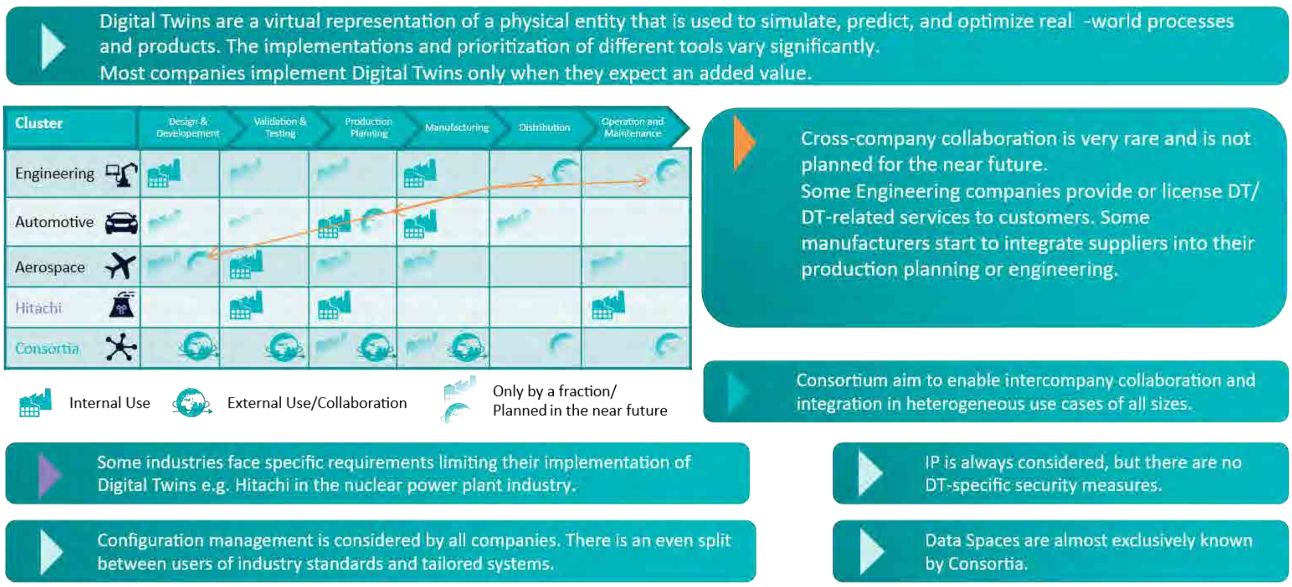


Figure 13: Survey Key Results

Taken together, these findings present a realistic picture of the current implementation phase of Digital Twins. They reveal both progress and gaps: on the one hand, the technology is widely recognized and increasingly integrated into product development processes; on the other, large parts of the value chain remain disconnected, and collaboration across company boundaries is still rare. The analysis underscores the need for standardization, governance frameworks, and trusted environments for data exchange in order to unlock the full potential of collaborative product development with Digital Twins.

For this reason, a dedicated follow-up action within the project will focus on researching these initiatives in greater depth. The goal will be to map their relevance to different industries, assess their maturity and adoption potential, and identify pathways for companies to participate. By systematically evaluating initiatives such as Catena-X, the project can provide clear recommendations for industry partners and highlight opportunities to accelerate collaborative Digital Twin adoption through shared infrastructures.

3 Terminology & Ontologies in the Context of Collaborative Digital Twins

In collaborative digital twin development, precise terminology and well-structured semantic models are essential to ensure interoperability, clarity, and consistency across domains and disciplines. This chapter provides a conceptual foundation by introducing key terms and semantic constructs, that are central to a shared understanding within the project. It begins with a focused selection of terminology that supports cross-functional communication and continues with an exploration of ontologies and aspect models, which serve as structuring mechanisms for knowledge representation and system modeling. Together, these elements form the semantic backbone of the collaborative framework and enable scalable, modular, and semantically aligned engineering practices.

3.1 Terminological Foundations for a Shared Understanding

In collaborative digital twin development, a shared understanding of key concepts is essential to ensure consistency across disciplines, domains, and stakeholders. This chapter introduces and defines a curated set of terms that are central to the project's conceptual and technical foundation[4]. Unlike comprehensive domain glossaries, which cover a broad range of terms, the focus here is on terminology that is particularly relevant within the context of this initiative. These definitions aim to support clear communication, reduce ambiguity, and foster alignment across teams working on architecture, data integration, semantic modeling, and solution design. This also applies to consistent collaboration across work packages with the Collaborative Digital Twin project.

The following table provides a selection of key terms that are essential to the collaborative development of digital twins within the CDT project. These terms have been extracted from a broader multi-domain glossary with domain specific terms and are highlighted here due to their relevance across the project teams and disciplines. The aim is to foster a shared understanding and consistent usage throughout the initiative.

[4] The extended, complete glossary is available to prostep ivip members: https://prostep-ivip.atlassian.net/wiki/x/LYcY

Table 1: Selection of key terms essential to the collaborative development of Digital Twins.

Key Term	Explanation	External References
Architecture Building Block	An Architecture Building Block is a package of functionality that describes a single aspect of the overall model. It defines what functionality will be implemented, captures business and technical requirements, is technology aware and solution-neutral, and directs and guides the development of SBBs.	Based on TOGAF
Aspect	An aspect is a domain-specific view on information and functionality associated with a specific digital twin with a reference to a concrete Aspect Model. An aspect is a software service to retrieve data at runtime or live data of a Twin from a data source or to trigger operations. An aspect implementation includes an interface (API) offering meaningful data for consumption and ensures data compliance to the Aspect Model specification via a defined interface."	Glossary :: Bosch Semantic Stack Documentation
Aspect Model	An Aspect Model is a formal, machine-readable description of the structure and semantics of data related to a specific aspect. Aspect Models are specified using the Semantic Aspect Meta Model (SAMM), a specification provided open source at the Eclipse Semantic Modeling Framework (ESMF). It is used to organize digital representation of information, capture the semantics of the domain, and create a clear context for related data. Aspect models can be reused in similar use cases and are stored in an aspect model catalog to provide quick and easy access for authorized users and applications. They support various formats (e.g. XML, JSON) and ensure consistency and interoperability in data representation.	Glossary :: Bosch Semantic Stack Documentation
Collaboration	A structured, trust-based form of joint work in which equal participants contribute to a shared goal—such as the creation, enrichment, or use of digital twins—within a federated data space. Unlike cooperation, which supports individual objectives, collaboration implies shared ownership, integrated contributions, and collective responsibility.	
Collaborative Digital Twin Framework	The Collaborative Digital Twin Framework (CDT Framework) describes a consistent and trustfully linked framework to enable timely collaboration (cross company / cross domain) utilizing digital twin technologies.	
Core Ontology	A core ontology is a generic, high-level semantic model that defines the fundamental concepts and relationships common to multiple domains or scenarios.	

Key Term	Explanation	External References
Data Space	A Data Space is a federated, decentralized, digital infrastructure enabling secure data exchange while maintaining data sovereignty. The participants of the data space can decide how, when, and with whom they share their data. To ensure a compliant and trusted way of data exchange all data providers and consumers need to mutually agree to rules. This ensures interoperability and trust, facilitating cross-company collaboration	http://docs.gaia-x.eu/technical-committee/architecture-document/22.10/federation_service/ https://catenax-ev.github.io/docs/operating-model/why-introduction
Digital Shadow[5]	A digital shadow depicts the externally created data profile of a component or a product. The data profiles are created by means of algorithms that include acquisition, analysis, evaluation, and consolidation of data. The digital shadow is used as the foundation for decisions.	Iske, S.: Medienbildung im Kontext digitaler Personenprofile. In book: Von der Bildung zur Medienbildung, Verständig, D., Holze, J., Biermann, R. (edts.), Springer VS, Wiesbaden, 2016, pp. 257-280, https://link.springer.com/chapter/10.1007/978-3-658-10007-0_13
Digital Twin	A Digital Twin is a virtual identifiable representation of a real-world entity (product, concept, process), which is bidirectionally connected to it at a specified frequency and fidelity.	Grieves, M. (2011): Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management. Space Coast Press (2011), ISBN 978-0982138007 LInk
Domain Ontology	A domain ontology is a formal, structured representation of knowledge specific to a particular area of expertise. It defines the key concepts, relationships, and rules within that domain, enabling consistent interpretation, reasoning, and data integration across systems and stakeholders.	
Solution Building Block	A Solution Building Block is a candidate solution that conforms to the specification of the assigned Architecture Building Block (ABB). It defines what product, or component will implement the functionality, defines the implementation, fulfills business requirements, and is product- or vendor-aware. For example, a SBB can be a tool or a tool chain.	Based on TOGAF

[5] A sensor stream from a physical twin feeds a dashboard without feedback mechanisms. The dashboard contains or displays a digital shadow.

This selection of terms provides a shared linguistic and conceptual foundation for the collaborative development of digital twins and serves as a reference point for the semantic structures and modeling approaches discussed in the following sections.

3.2 Semantic Structures and Aspect Models in the Collaborative Digital Twin Framework

In the context of the Collaborative Digital Twin (CDT) framework of the prostep ivip association, semantic interoperability is a foundational requirement for enabling consistent, meaningful data exchange across the entire product lifecycle. This is especially crucial in complex, multi-stakeholder environments such as automotive, aerospace, and mechanical engineering, where digital twins must integrate heterogeneous data sources, systems, and domain perspectives.

To achieve this, the CDT framework leverages semantic structures—notably ontologies and aspect models—as key enablers of interoperability. These structures provide a shared vocabulary and formal logic that allow systems to interpret and reason about data in a consistent way, regardless of origin or format.

3.2.1 Ontologies and Aspect Models

This section introduces the foundational concepts of ontologies and aspect models, explaining how they enable semantic interoperability by providing structured, machine-readable representations of domain knowledge and modular views of product data. The chapters also explore their interrelation, and present relevant initiatives and standards—both within the prostep ivip association and externally—that support the development and adoption of semantic structures in the Collaborative Digital Twin framework.

3.2.1.1 Ontology Definition

Ontologies are formal, structured representations of knowledge within a specific domain. They define concepts (classes), relationships (properties), and constraints in a way that is both human-readable and machine-interpretable[6]. By establishing a shared vocabulary and logic, ontologies enable consistent understanding and reasoning across systems, organizations, and lifecycle stages.

Ontologies are used for a variety of purposes, including:

- **Semantic interoperability**: Ensuring consistent meaning across systems.
- **Data integration**: Linking heterogeneous data sources through shared semantics.
- **Reasoning and inference**: Enabling intelligent systems to derive new knowledge from existing data.
- **Knowledge management**: Structuring domain expertise for reuse and scalability.
- **Digital twin modeling**: Providing the semantic layer for describing physical and virtual entities.

A key characteristic of ontologies is that they operate under the Open World Assumption (OWA). This means that the absence of a statement or fact in an ontology does not imply its falsehood – rather, it is simply unknown or not yet defined. While this assumption supports flexibility and extensibility, it also introduces several challenges in practical applications. For instance, datasets may be incomplete, making it difficult to perform quality checks or validate data consistency, since one can never be certain that all relevant information is present. Additionally, varying levels of detail across different data sources can hinder uniform data handling, especially during data exchange between systems with different modeling granularity.

---

[6] Introduction to Ontology, Springer. DOI 10.1007/978-3-642-01904-3_3

3.2.1.2 Aspect Model Definition

Aspect models are modular, semantic representations of specific views or facets of a product, process, or system. Unlike ontologies, which define general domain knowledge, aspect models focus on structuring data relevant to a particular context—such as a product's material composition, manufacturing parameters, or lifecycle status. They are designed to be reusable, composable, and interoperable across different systems and stakeholders[7].

They can play a central role in enabling consistent data exchange and interpretation throughout the product lifecycle. By encapsulating domain-specific semantics and linking them to standardized data structures, aspect models allow digital twins to be built and understood collaboratively, even when data originates from different organizations or IT systems. Example uses for aspect models are:

- **Standardizing data exchange** across organizational boundaries.
- **Structuring digital twin data** in a modular and scalable way.
- **Enabling semantic validation** and automated reasoning.
- **Supporting lifecycle integration**, from design to recycling.
- **Facilitating compliance and reporting**, e.g. for sustainability or regulatory requirements.

In contrast to ontologies, aspect models typically operate under the Closed World Assumption (CWA). This means that any information not explicitly stated in the model is assumed to be false or irrelevant. While this assumption supports data validation, schema enforcement, and predictable behavior in operational systems, it also introduces limitations. For example, it requires that all relevant data be present and complete at the time of exchange, which can be challenging in distributed environments. Furthermore, differences in modeling granularity or completeness between stakeholders can lead to incompatibilities or data loss during integration. Therefore, aspect models must be carefully designed to balance strictness with flexibility, especially when used in collaborative, cross-organizational scenarios.

3.2.1.3 Combining Ontologies and Aspect Models

Ontologies and aspect models serve complementary roles in the semantic structuring of data. Ontologies provide the semantic foundation—defining the concepts, relationships, and rules of a domain—while aspect models apply this foundation to specific data contexts. In other words, ontologies describe *what things mean*, and aspect models describe *how those things are represented and exchanged* in a given scenario.

It is possible to reference ontology concepts from inside aspect models, allowing concepts to be shared across multiple models in different contexts and levels of detail. However, the differences in regard to their flexibility (OWA vs. CWA) must be considered and managed to avoid semantic mismatches and ensure robust data.

3.2.1.4 prostep ivip Ontologies

Within the prostep ivip association, the development of ontologies has been an ongoing topic. A working group is delegated to curate a Top Level Ontology (psiOnt[8]), which defines concepts that are shared between other psi working groups to enable straightforward integration. This working group is also driving the effort that other working groups, such as the VEC, share their models as ontologies as well.

The most elaborated ontology within the psi space is the Digital Data Package (DDP[9]) ontology. This ontology covers various domains across the entire lifecycle, with the goal of helping users share

---

[7] https://docs.bosch-semantic-stack.com/concepts/aspect-model.html
[8] https://prostep-ivip-e-v.github.io/TLO/
[9] https://www.prostep.org/en/projects/digital-data-package

their data as simply as possible and with minimal information loss. The DDP schema and ontology are currently prepared for standardization within the CASCaDE[10] project of the GfSE.

3.2.1.5 Semantic Models in Engineering

Ontologies are also a well-observed topic in the industry outside of the prostep ivip association. Most remarkable in this context are the efforts of the World Wide Web Consortium (W3C[11]), Catena-X-based initiatives, including GAIA-X and Eclipse Tractus-X and, considering data-exchange, the Open Services for Lifecycle Collaboration (OSLC[12]) standard and vocabularies, which also rely on ontologies. Aside from these major contributors, use-case-centric data models, such as ReqIF, are available, even if they are not necessarily based on ontologies or aspect models.

Of the initiatives listed, W3C and OSLC vocabularies typically utilize ontologies and follow a modularized approach. Each ontology will cover a previously defined scope and use case. The common ground for W3C ontologies are the framework vocabularies RDF(S) and OWL[13], whereas the OSLC vocabularies share common concepts that build on RDF(S) and OWL. However, the shared concepts are of a generic nature, domain and engineering knowledge is handled mainly in the domain vocabularies of OSLC.

In Catena-X-related initiatives, aspect models are being used[14]. These initiatives address very concise scopes within each aspect model and share only fundamental concepts, such as the formalization of identifiers, between models. Considering the range of available models and vocabularies as well as velocity in publishing new models, Catena-X-related initiatives are currently the forerunner of the industry. This is based on their strategic decision to publish first releases as quickly as possible and accepting the risk that shortcomings and other issues will only be found later in the field, requiring new releases, potentially with breaking changes.

3.2.2 CDT Ontology Framework

In the context of the Collaborative Digital Twin (CDT) framework, ontologies serve as the semantic backbone for data exchange. They ensure that different stakeholders—such as OEMs, suppliers, and software systems—can interpret data in the same way, even when it originates from diverse sources. This is essential for enabling traceability, automation, and integration across the product lifecycle.

For this purpose, an ontology framework is presented as a foundational component of the overall CDT architecture. It provides the semantic infrastructure necessary for enabling machine-readable, uniform data models that support consistent processing and exchange across systems, domains, and organizational boundaries. Designed to be domain-agnostic and extensible, the CDT Ontology Framework can be adapted to any use case, scenario, or industry. While the CDT initiative focuses on providing the framework and guidance for its use, the architecture itself supports broad adoption across cross-domain and cross-partner collaborations.

3.2.2.1 Semantic Architecture

The semantic architecture of the CDT Ontology Framework is designed to support scalable, modular, and interoperable knowledge representation across domains and organizations. It proposes a layered structure that is complemented by a generic core ontology, see Figure 14. This approach enables semantic consistency while allowing for domain-specific and enterprise-specific extensions.

---

[10] https://cascade.gfse.org/
[11] https://www.w3.org/groups/
[12] https://open-services.net/
[13] https://www.w3.org/TR/owl2-syntax/
[14] https://catenax-ev.github.io/docs/next/standards/CX-0143-UseCaseCircularEconomyDigitalProductPassportStandard/CX-0143-SEM

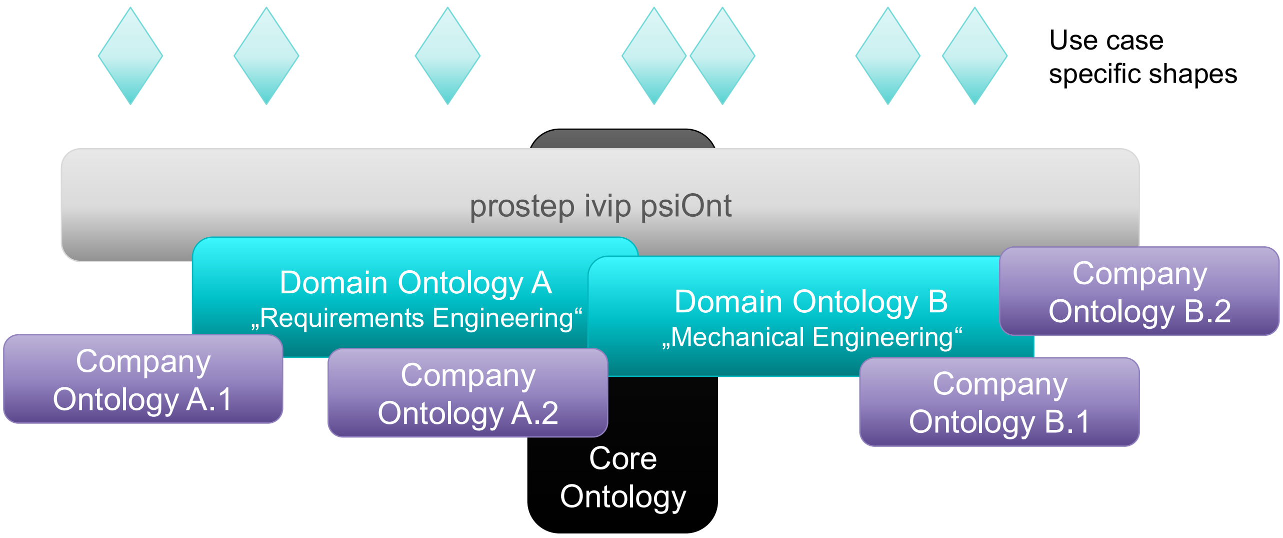


Figure 14: Overview of the CDT Ontology Framework Architecture

The focus of the proposed ontology architecture is the **Core Ontology**, which provides the essential concepts and relationships common to collaboration use cases. This ensures a shared semantic base that supports uniform data processing and interpretation across the CDT framework.

The semantic center of the **layered structure** is Domain Ontologies, which capture the semantics of specific engineering disciplines—such as Requirements Engineering and Mechanical Engineering. These ontologies formalize domain knowledge and provide reusable building blocks for modeling lifecycle data. Further specialization is achieved through company-specific ontologies, which extend domain ontologies to reflect enterprise-specific terminology, processes, and data structures. These ontologies allow individual organizations to tailor the semantic framework to their internal needs while maintaining compatibility with the broader CDT ecosystem. At the top of the architecture, upper-level ontologies such as the prostep ivip Top-Level Ontology (TLO) serve as a unifying semantic layer that connects the semantic layer across domains. It provides high-level abstractions and ensures alignment across domains and companies.

Additionally, for exchange purposes **use-case-specific shapes**—represented as structured aspect models—are derived from this ontology stack to support concrete data exchange scenarios.

This architecture enables a flexible yet standardized approach to semantic modeling, allowing stakeholders to define, extend, and exchange digital twin data in a consistent and machine-readable manner.

3.2.2.2 Methodology

The development of the CDT Ontology Framework followed an iterative and collaborative approach, combining expert knowledge, user journey analysis, and established ontology engineering practices. The methodology ensured that the resulting semantic structures were both theoretically sound and practically applicable to the CDT use cases. The overall methodology combined the processes to create the core, domain and enterprise ontologies, followed by the collection of competency questions for the creation of aspect models (see Figure 15).

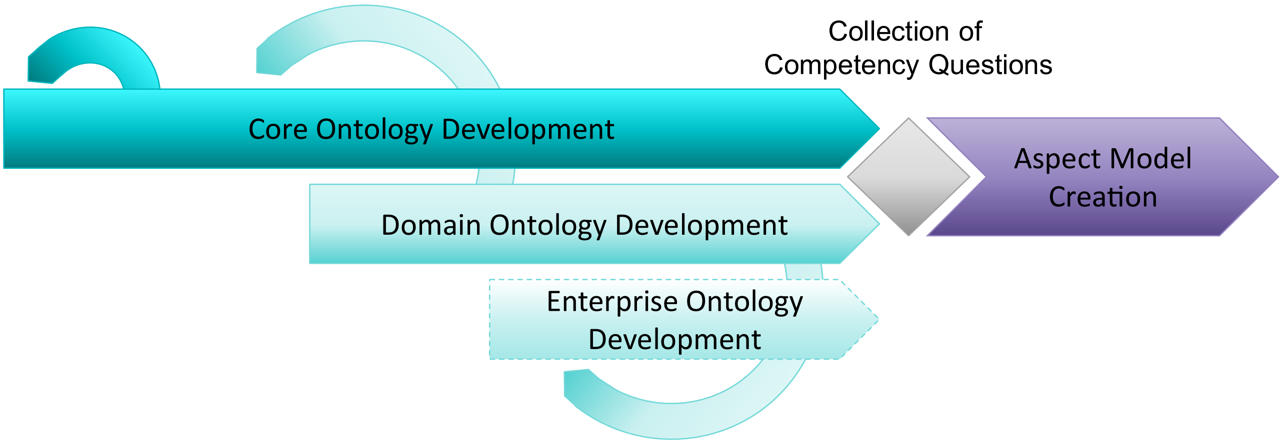


Figure 15: Applied Methodology for the Ontology Framework

When modeling any ontology, a uniform process was followed, see Figure 16. The process began with an analysis of user journeys and glossary terms collected within the CDT project. These artifacts provided the initial set of concepts and relationships relevant to the CDT context. Based on this foundation, modeling workshops were conducted where domain and ontology experts collaborated to identify and relate concepts in a meaningful manner.

To enrich this foundation, related ontologies and data models from industry and research were studied. Based on these findings, conceptual diagrams were created to sketch the ontology structure. These diagrams were presented to the project team and discussed in domain expert interviews. Once the interviews resulted in a stable version of the ontology, it was formalized in TTL and shared via GitHub[15].

Figure 16: Applied Ontology Modeling Process

After each process step, a revised version was produced and re-evaluated in subsequent sessions. If necessary, earlier steps were revisited with new findings in mind. This iterative cycle was repeated for each user journey covered. When new user journeys introduced additional requirements, updates were made to the core ontology and, where necessary, to the associated aspect models.

Once the ontologies reached a mature state, competency questions were collected in workshops with domain experts. Using these competency questions, both the designed ontologies were verified and the required aspect models created. To ensure the validity of the competency questions and avoid manipulation to make the questions fit the ontology, the designers of the ontology were not involved in this step. The questions focused on data exchange between partners as described in the user journeys. The collected questions were reviewed for relevance and refined where necessary.

Aspect models were then created using the SAMM Aspect Model Editor[16], which provides an integrated development environment for modeling SAMM-compliant aspect models. Each aspect model was designed to represent the data required to answer the competency questions, including conditions specified by the questions.

The modular architecture described in the previous chapter guided the modeling efforts, with a strong focus on the core ontology. Domain ontologies were either reused or created as proof-of-concept without extensive workshops, as they were not intended to be final deliverables. Validation of the ontologies was achieved by collecting and answering competency questions derived from the use

[15] https://github.com/prostep-ivip-e-V/cdt
[16] https://docs.bosch-semantic-stack.com/oss/aspect-model-editor.html

cases. Due to budget constraints, the enterprise ontology layer was not implemented in this phase of the project. However, the architecture remains prepared to accommodate company-specific extensions in future iterations.

The methodology was iterative and collaborative, involving one ontology expert working closely with multiple domain experts. Collaboration tools included Confluence for glossary management and conceptual diagrams, and Protégé as the ontology engineering environment. The SAMM Aspect Model Editor was used for aspect model creation.

Challenges encountered included the need to elaborate certain user journey definitions to create meaningful domain ontologies and to obtain sufficient input from domain experts. These lessons highlight the importance of clear requirements and active stakeholder engagement in semantic modeling projects. Furthermore, while it was helpful to collect competency questions for formalizing aspect models, the lack of competency questions for creating the ontologies itself caused issues, especially regarding finding the appropriate abstraction level and scope of the developed core and domain ontologies.

3.2.2.3 Modeling Principles

Considering current modeling best practices and compatibility with tools for generating aspect models, five modeling principles have been agreed upon and are recommended for creating domain or enterprise ontologies in the context of prostep ivip ontologies with or without combination with aspect models.

- The base URI of the ontology must not contain upper case letters or special characters
- Local names of classes, and object and datatype properties must be camel cased, with class names starting with a capital letter
- Labels and comments from the Resource Description Framework Schema (RDFS) must be used to describe the concepts in natural language using language strings
- Properties with multiple concepts in domain (starting node) or range (ending node) must be defined once per valid combination and named as follows:
    - *BasePropertyName-range-forDomain*, e.g. *hasState-MaturityState-forRequirement*
    - Each property that has been modelled this way must be a sub property of the base property it refers to in its name.

The last modeling principle allows applications and reasoners to utilize the logic of precise and meaningful properties to infer knowledge about the connected entities or define precisely the required information. At the same time, the base properties can be used for querying, reducing query complexity. Following the recommended naming principle will also support users, and domain and modeling experts in understanding the purpose and specification of the properties quickly.

3.2.2.4 Core Ontology

The **CDT Core Ontology** establishes the semantic foundation for collaboration in the CDT framework. It models the essential entities and interactions required for exchanging and managing engineering data across organizational boundaries. For this purpose, the core ontology covers six aspects of collaboration:

1. Versioning
2. Internal and external information objects
3. Internal and external information workflows
4. Security classification

5. Feedback between collaboration partners
6. Collaborative change processes

At its center is the abstract concept of an **Information Object**, representing work artifacts such as CAD models or system components. To introduce a versioning concept, **Concept Artifacts** (representations without version specificity) and **Versioned Artifacts** (precise versions of a concept artifact) are introduced. This approach has been modelled in alignment with the OSLC Configuration Management vocabulary concepts for versioned resources[17]. When sharing information with a partner, a dedicated versioned artifact, the **Exchange Object**, must be used to ensure traceability and integrity during collaboration. The concepts and their relations are presented in Figure 17.

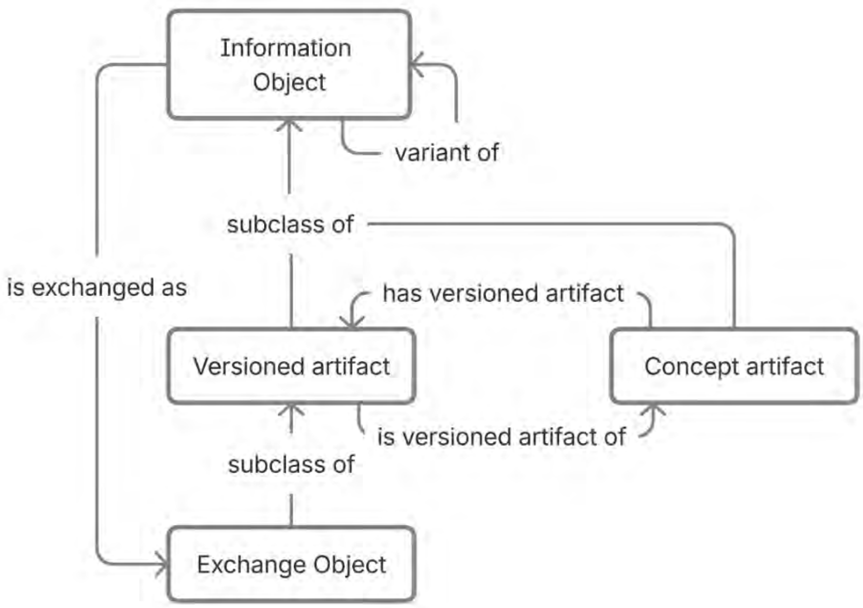


Figure 17: Information and Exchange Objects including Versioning

Additional properties define lifecycle states and metadata, such as **Workflow State**, **Maturity State**, and **Collaboration State** for both information objects in general and exchange objects specifically (Figure 18). These states are modelled as enumerations and include standard values as proposed by the OSLC[18] vocabulary.

[17] https://docs.oasis-open-projects.org/oslc-op/config/v1.0/os/config-vocab.html
[18] https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/errata01/os/change-mgt-vocab.html

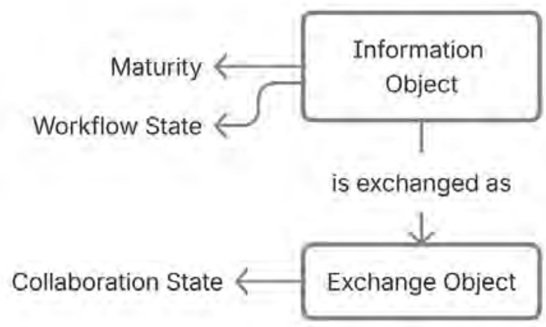


Figure 18: Workflow, Maturity and Collaboration State

Furthermore, concepts to add a **Security Classification** and allowed interactions to information and exchange objects directly are introduced in the core ontology, see Figure 19. With the security classification added internally to any information object, it is possible to automatically deduce the allowed interactions on an exchange object based on the partner the object is shared with. An **Update State** may indicate which actions the exchange partner may perform on the exchange object, e.g. read, update or delete. As the rules for such deductions may vary for each enterprise and within one enterprise for each collaboration partner, the core ontology aims to be generic in both the security classification and update state, recommending creating individualized subclasses when adapting to the proposed ontology architecture.

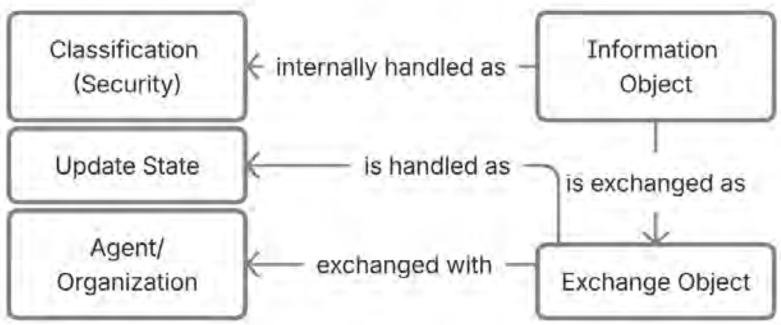


Figure 19: Security Classification

Collaboration feedback on an exchange object between partners is captured through **Reactions** (Figure 20). While reactions are modelled as a generic concept, it is further specified to **Comments** for discussions, **Change Requests** for proposed modifications, and **Changes** for implemented updates. Each reaction may have an identifier allowing tracking to internal source systems, a timestamp on when the reaction was given, a reason why this reaction was given as well as its content in form of its description.

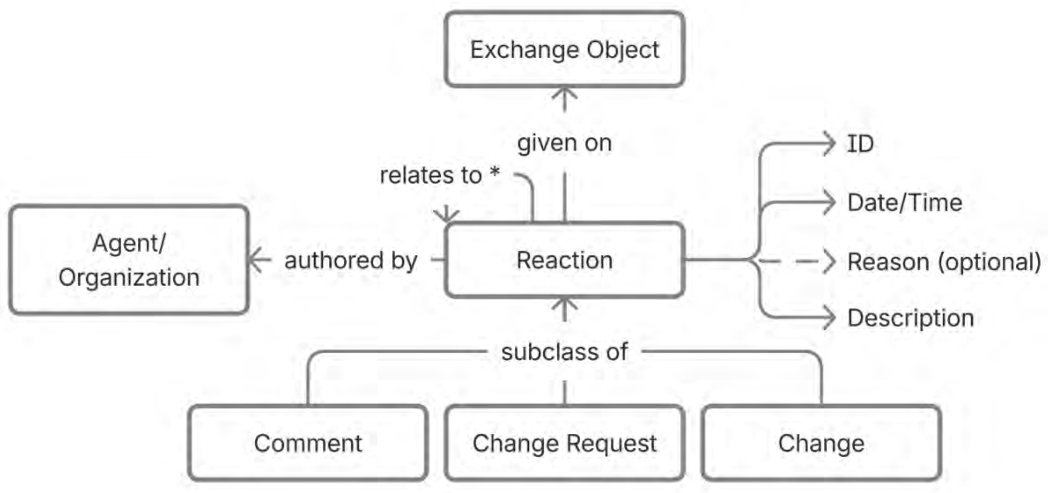


Figure 20: Reactions on Exchange Objects

It is also possible to relate a reaction to another reaction. The **relates to** property again serves as a generic property which has been extended by more specific relations for approving, mitigating, rejecting and resolving comments, as well as concepts to relate a change request to the change that has implemented it and vice versa.

The overall ontology is visualized in Figure 21 and published on the CDT project of the prostep ivip GitHub[19] account.

[19] https://github.com/prostep-ivip-e-V/cdt

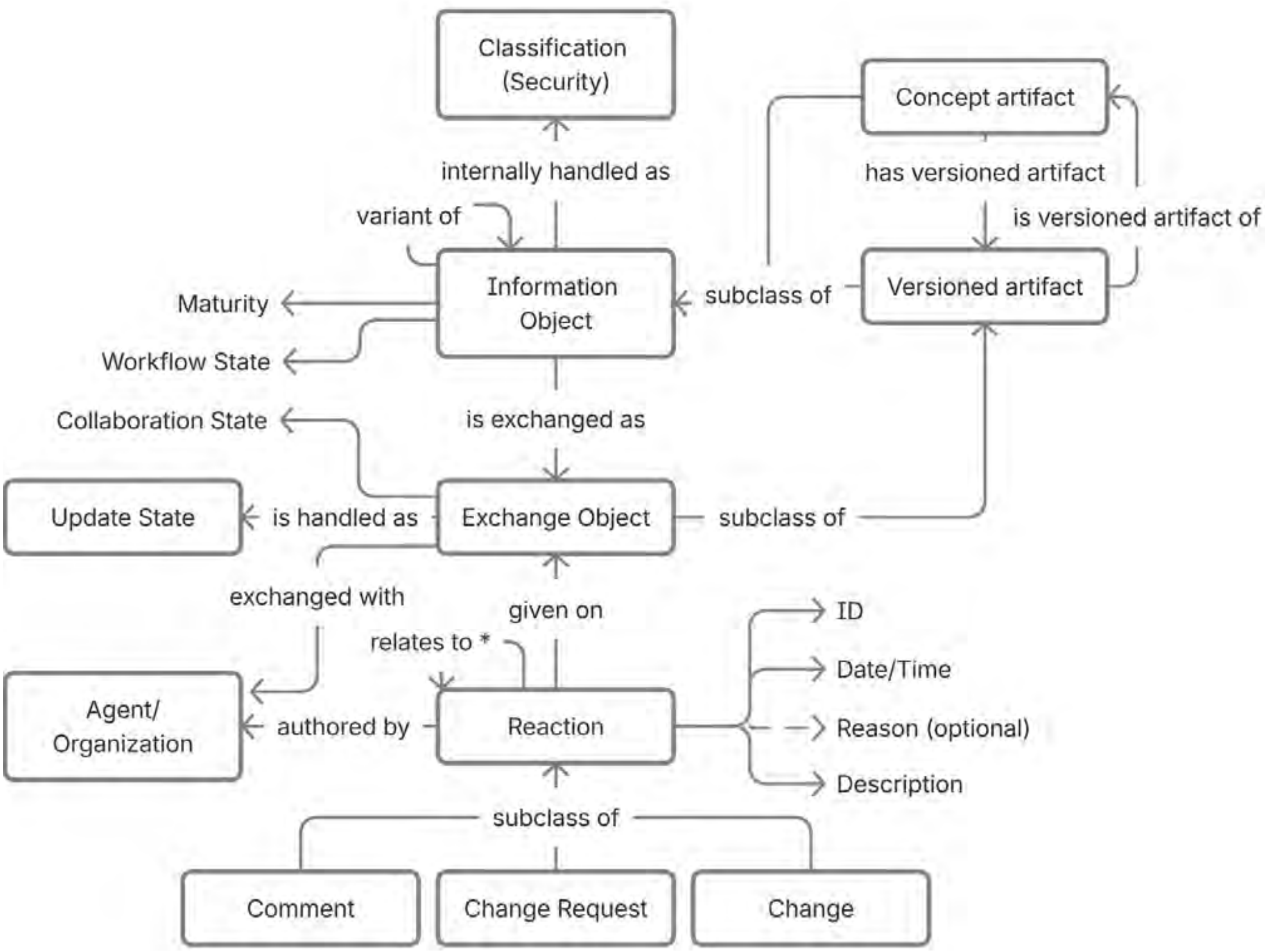


Figure 21: Overview of Entire Core Ontology

3.2.2.5 Domain and Enterprise Ontologies

Within the CDT project, a proof-of-concept requirements management and systems engineering ontology have been modelled based on the underlying user journeys. Both ontologies rely on concepts that are covered in other standards, ontologies and data models and can be found on the CDT project on the prostep ivip GitHub page. It is recommended that fitting existing domain ontologies are utilized where possible when adapting the CDT ontology architecture.

Enterprise ontologies have not been created.

3.2.2.6 Aspect Models

Aspect models in the CDT Ontology Framework serve as structured representations of data exchange scenarios, derived from competency questions that reflect real collaboration needs. They provide a machine-readable specification of what information should be exchanged and under which conditions, ensuring semantic consistency across partners. To align with the Catena-X research initiative, we used the Semantic Aspect Meta Model (SAMM)[20] to create the aspect models.

Within a modeling workshop, initially 11 competency questions were collected from domain experts. These questions were analyzed regarding their relevance to the user journey, applicability to the available domain ontologies and revised or merged if needed. The resulting seven competency questions have been grouped by the concepts on which the query conditions are set:

[20] https://docs.bosch-semantic-stack.com/oss/samm-specification.html

- **Requirements** (*GET requirements that I am / my organization is allowed to see*, *GET the type of requirement, GET all available requirements for relevant components shared by the OEM*).
- **Comments** (Which requirements have received new comments?, *Which comments have been resolved by my project partners?*).
- **Comments per Requirement** (e.g., *GET all available comments on a requirement (set), GET an update on new comments on a requirement (set)*).

Based on these competency questions, seven aspect models were developed. The models were created using the SAMM Aspect Model Editor, ensuring compliance with the Catena-X Semantic Aspect Meta Model (SAMM) specification. Each aspect model includes a structured schema for the data required to answer the competency question and semantic references to the CDT Core Ontology for consistent interpretation. The aspects have been published on the research project's GitHub[21] repository.

3.2.2.7 Compatibility with Catena-X Aspect Models

As described in section 3.2.1, the CDT ontology framework differs from the Catena-X approach based on the applied technologies. While the CDT ontology framework is using the entire RDF, RDFS and OWL capabilities, promoting the Open World Assumption for easy extendibility, Catena-X focuses on the exchange process solitarily, using SHACL[22] shapes and low logical profiles, and following the Closed World Assumption. Depending on the considered use cases, both approaches may be valid and feasible, and bear benefits over the other.

Aside from the technical differences, the scope of both initiatives varies. The CDT ontology framework does not only consider the individual exchange scenarios, but on a continuous Digital Twin in collaboration. The aspect models for data exchange are seen as snapshots of the Collaborative Digital Twin, that integrate seamlessly into the overarching knowledge graph. The ontologies serve to create semantic relationships between information objects between different data exchange points. On the other hand, Catena-X aspect models are created for one individual data exchange scenario and do not utilize concepts from domain and cross-domain ontologies. Hence, a "requirement" class in one aspect model is not the same as a "requirement" class in another one. However, it is possible to add references to thesauri, ontologies or other data models via annotation properties.

Finally, the CDT ontology framework proposed the generic core ontology that defines shared, generic concepts regarding collaboration. This helps to build a collaborative environment that can handle exchange data, such as comments, security classification, etc., in a uniform manner. In Catena-X aspect models only generic properties, such as identifiers, are relying on shared concepts. This leads to every working group defining their own approach to handling and model exchange meta data. Creating a collaborative environment therefore requires more effort, as the core functionalities may be adjusted frequently.

Concluding, the CDT ontology framework is compatible with Catena-X aspect models but provides more stability in regard to handling core functionalities of exchange data and creating a continuous Collaborative Digital Twin. Both initiatives may be focusing on the same scenario, i.e. collaboration, but their scopes differ. A combination of the CDT ontology framework that is mapped to the Catena-X aspect models may be able to offer the best of both worlds.

3.2.2.8 Conclusion

The CDT project recommends an ontology-based framework as the semantic backbone for collaborative digital twin environments. This framework ensures interoperability by providing a

---

[21] https://github.com/prostep-ivip-e-V/cdt/tree/main/aspectModels
[22] https://www.w3.org/TR/shacl/

shared vocabulary and formal logic for consistent interpretation of data across diverse systems, domains, and organizations. It combines ontologies—which define domain concepts and relationships—with aspect models that structure data for specific exchange scenarios. The architecture is layered, featuring a generic core ontology for common collaboration concepts, domain ontologies for discipline-specific semantics, and optional enterprise extensions for company-specific needs. Together, these components enable scalable, modular, and machine-readable data models that support traceability, automation, and lifecycle integration.

To operationalize this approach, the CDT methodology emphasizes iterative development, stakeholder involvement, and alignment with established standards. Aspect models derived from competency questions ensure that data exchange scenarios are semantically consistent and practically relevant. However, the application of competency questions for creating the ontologies itself proved valuable and should not be skipped. We recommend to create two sets of competency questions, one for the ontologies and one for the aspect models, as each has their own, very different scope and modeling approach.

While the CDT framework promotes an open-world assumption for flexibility and extensibility, it remains compatible with initiatives like Catena-X, which follow a closed-world approach for strict validation. By providing a robust semantic infrastructure, the CDT ontology framework facilitates seamless collaboration, reduces integration effort, and lays the foundation for continuous, cross-domain digital twins.

4 Blueprint for DT-ready IT Architectures

The IT architecture of the CDT framework is designed to enable secure, scalable, and interoperable collaboration between business partners across organizational boundaries. It is built on a federated, modular, and open design that ensures data sovereignty while supporting integration into existing IT landscapes. While the primary focus lies on the product creation phase – represented by the Type-Twin[23] – the architecture is designed to be lifecycle-independent.

This architectural blueprint is detailed in the following chapters, which cover the key architecture constraints and principles, the structure and roles of the ABBs, supported architecture variants , and the basic interaction flow.

4.1 Architecture Constraints and Principles

The CDT architecture is governed by a set of architecture constraints and guiding principles that ensure interoperability and alignment with the project's objectives. These principles define the architectural boundaries for all components and integrations within the framework. An overview of the architectural constraints and principles described in this section is provided in tabular form in Table 2. The constraints and guiding principles define the already fixed boundary conditions (= constraint) and underlying general rules / guidelines (= guiding principles). They reflect a level of consensus among all stakeholders and form the basis for making the needed decisions. They can be of type business, data, application and technology. The guiding principles have to be considered within all architecture options and the target architecture.

Each constraint/ principle must be described using the following pattern (Source: TOGAF[24]):

- **Name**: Should both represent the essence of the rule as well as be easy to remember. Specific technology platforms should not be mentioned in the name or statement of a principle. Avoid ambiguous words in the Name and in the Statement such as: "support", "open", "consider", and for lack of good measure the word "avoid", itself, be careful with "manage(ment)", and look for unnecessary adjectives and adverbs (fluff). In case of a given constraint the source of the constraint must be mentioned. E.g. by referring to a Bosch corporate standard, a already taken decision etc.
- **Statement**: Should succinctly and unambiguously communicate the fundamental rule. For the most part, the principles statements for managing information are similar from one organization to the next. It is vital that the principles statement be unambiguous.
- **Rational**: Should describe the requirement more in detail: Highlight the business benefits of adhering to the principle, using business terminology. Point to the similarity of information and technology principles to the principles governing business operations. Also describe the relationship to other principles, and the intentions regarding a balanced interpretation. Describe situations where one principle would be given precedence or carry more weight than another for making a decision.
- **Implications**: Should highlight the requirements, both for the business and IT, for carrying out the principle - in terms of resources, costs, and activities/tasks. It will often be apparent that current systems, standards, or practices would be incongruent with the principle upon adoption. The impact to the business and consequences of adopting a principle should be clearly stated. The reader should readily discern the answer to: "How does this affect me?" It is important not to oversimplify, trivialize, or judge the merit of the impact. Some of the implications will be identified as potential impacts only, and may be speculative rather than fully analyzed.

[23] A type twin refers to a digital representation of a type of products, while an instance twin represents a specific product instance with a unique identifier.
[24] http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap29.html

A core principle is the use of direct data exchange based on federated data storage. Rather than relying on centralized data hubs, the CDT framework ensures that data remains within the persistence of its owner. This approach reduces legal and security complexities while preserving data sovereignty. To enable this, the architecture requires harmonized data models and formats, local mapping capabilities, and a standardized data connector with API support.

To minimize integration efforts, particularly in brownfield environments the CDT architecture reuses existing collaboration network building blocks such as those developed within Catena-X. At the same time, it remains independent of formal network membership, ensuring that it can function autonomously.

To support traceability and consistency across the product lifecycle, the use of persistent references is required. These references must remain valid over time and be resolvable independently of specific technologies or platforms.

To allow variant and configuration management, the data must always be clearly associated with specific product variants or configurations. Versioned data is retrieved and stored locally. As product data often spans over multiple domains and organizations, cross-domain variant management is a critical requirement.

Openness is another fundamental principle of the CDT IT-architecture. All central components must comply with the Code of PLM Openness (CPO). The CPO criteria serve as a benchmark for evaluating the architectural elements, ensuring that the framework remains open and adaptable.

Collaboration within the CDT framework is based on Digital Twins, which serve as semantic, machine-readable representations of assets. Therefore, the architecture must enable the definition, creation, cataloging, and provisioning of twins and their related aspects. Semantic alignment is achieved through the use of ontologies and aspect models.

Although the CDT architecture is designed to be lifecycle-agnostic, it places a particular emphasis on the product creation phase, focusing on typologizing data (Type-Twin). The framework supports the integration of data and aspects from all lifecycle phases, but its structure is optimized for typologizing and contextualizing data during the early stages of product development.

To ensure interoperability with other Digital Twin frameworks, the CDT architecture supports the exchange and reuse of aspects across different ecosystems. This is achieved through the use of common standards, such as those promoted by the Industrial Digital Twin Association (IDTA).

Finally, the architecture supports multiple data exchange mechanisms, including push-based, pull-based, and notification-driven approaches. Each mechanism has specific architectural implications and requires different Architectural Building Blocks (ABBs) to ensure seamless and secure communication between participants.

Table 2: Architecture constrains and principles

Name	Statement	Rationale	Implications
Direct Data Exchange based on Federated Data Storage	Data must be exchanged directly between the involved business partners without any additional storage on a „central hub“.	By not storing the data centrally, the data remains under the sovereignty of the business partners involved and many legal and data security aspects become much simpler without a third party partner. Data exchange may include the downloading of files, but	Harmonized data models and exchange formats required (Local) capabilities required to map into harmonized data models / exchange formats Involved parties need a data connector with responding capabilities.

Name	Statement	Rationale	Implications
		not in the form of exported and sent documents.	Avoid visualization showing data in the middle between two partners.
Usage of Basic Collaboration Network Building Blocks	As far as possible, the CDT target architecture to be developed shall make use of already existing building blocks of Collaboration Networks.	In order for the member companies to be able to implement the CDT target architecture in their brownfield, conflicts to the current architecture shall be avoided. As many companies are part of initiatives building up industry sepcific or cross industry Collaboration Networks like Catena-X or Gaia-X, sticking to the architecture building blocks of these Networks will minimize the implementation effort for the CDT member companies. This constraint carries less weight that Constraint 1 (Decentral Data Exchange).	This requires a profound and up-to-date knowledge of the activities in these collaboration networks, as their results may be subject to change. It must be checked whether CDT member companies can gain the full benefit of the CDT target architecture even if they are not part of any Collaboration Network.
Persistent References	References used in data objects (e.g. reference from product data to a related CAD model) must be valid over the complete lifetime of the data object.	So that references do not run into the void, they must be valid and usable over the entire lifetime of the object.	Concept for „stable/persistent references“ required (Local) capabilities required to provide and mange stable/persistent references which are independent from techologies and concepts of the local systems providing the endpoints of the referenced objects
Variant & Configuration Managment	It must be ensured that the exchanged data can be precisely assigned to a specific variant or configuration, leveraging Stable Reference Management.	***Versioned shared data*** is pulled into the partner's IT environment and forms a permanent copy. Each time data is accessed, it is checked for updates.	Cross Domain Variant & Configuration Managment is required for each partner

Name	Statement	Rationale	Implications
Ensuring of Openness in the sense of CPO	(Central) SBBs must be compliant with the Code of PLM Openness (CPO)	To ensure exchangeability and integration of centrally used SBBs, the requirements defined by CPO must be supported.	All components for the central part of the solution should be evaluated using the CPO criteria
Collaboration based on Digital Twins	Basis for the collaboration of business partners is the Digital Twin concept.	Based on semantic models, Digital twins offer a well-documented and computer-processable representation of assets and asset definitions, making it easier for business partners to understand and communicate about the asset. This shared understanding enables collaboration and helps avoid misunderstandings or misinterpretations.	Capabilities for Digital Twin and Aspect management required. This includes definition, creation, cataloguing, lookup, provisioning, etc. of twins and aspects. Aligned semantics are formed in the ontology and aspect models. Necessity for lean description of digital twins (e.g. KBs of aspects and meta data instead of GBs of CAD and simulation files)
Cover Aspect of the Entire Life Cycle with Focus on Aspects of the Product Creation Process	It must be possible to integrate aspects of any lifecycle phases, whereas the focus is on aspects of the product creation process (describing the “typologizing” data, Type-Twin).	Direct implication from the project scope of CDT: „While the CDT project focuses on aspects of the product creation process (describing the “typologizing” data, Type-Twin), the digital twins managed in the CDT Framework may contain aspects from other lifecycle phases and it may as well be used in other lifecycle phases than development and may be integrated into other Digital Twin frameworks.”	The architecture must be generic and independent from the product lifecycle the aspect is for
Ability for integration into other Digital Twin frameworks	It must be possible to integrate Aspects from other Digital Twin based frameworks as well as provide Aspects to other frameworks	Direct implication from the project scope of CDT: „While the CDT project focuses on aspects of the product creation process (describing the “typologizing” data, Type-Twin), the digital twins managed in the CDT Framework may contain aspects from other lifecycle phases and it may	Common Standards (e.g. IDTA) for Digital Twins must be used Detailed knowledge of other Digital Twin based frameworks (e.g. Catena-X) required

Name	Statement	Rationale	Implications
		as well be used in other lifecycle phases than development and may be integrated into other Digital Twin frameworks."	
No Live Collaboration	Live collaboration is a corner case that is not included in the current CDT user journeys.	Live Collaboration is not part of the Project scope of CDT.	the architecture does not have to fulfill any requirements to enable live collaboration.
Data exchange mechanisms	Data exchange can be performed using push/constant pull mechanisms or notifictaions about updated data.	Depending on the use case different data exchange mechanisms are required.	Different exchange mechanisms set different requirenments for ABBs.

4.2 Architecture Building Blocks (ABBs)

The CDT architecture is structured around a modular set of Architecture Building Blocks (ABBs) shown in Figure 22. The ABBs are grouped into four functional clusters: Integration, Digital Twin Management, Collaboration, and Data Management. These clusters are explained in the following chapters.

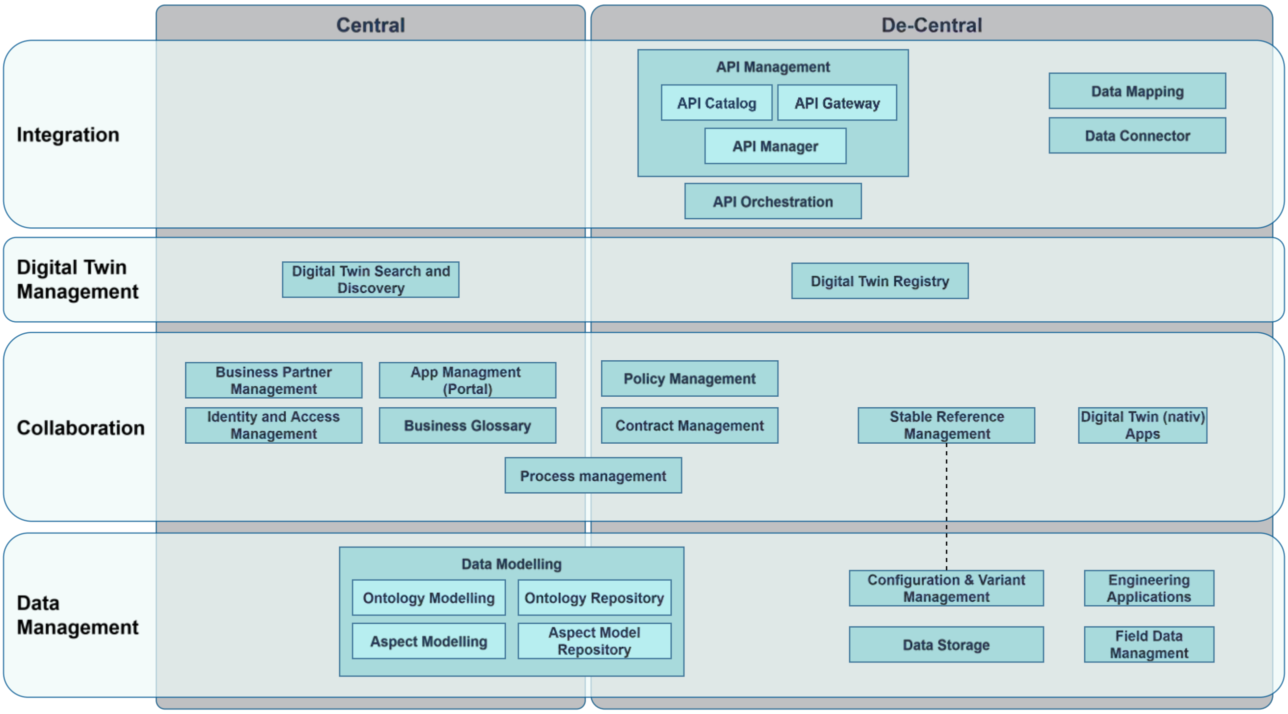


Figure 22: Architecture building blocks of the CDT architecture

4.2.1 Integration Cluster

The Integration Cluster includes the ABBs Data Mapping, Data Connector, API Management (comprising API Manager, API Catalog, and API Gateway), and API Orchestration. These ABBs provide the functionality required for data exchange between heterogeneous systems.

Data Mapping

This ABB provides functionality to support the transformation of data from source to target formats. This includes protocol mapping, format conversion for unstructured data, structural mapping for structured data, and key-value translation. For example, it enables the conversion of a proprietary JSON-based product structure into a STEP file or the transformation of a vendor-specific 3D CAD model into a neutral JT format.

Data Connector

This ABB provides functionality to connect to a Collaborative Digital Twin Framework.

API Management

This ABB provides functionality to support the process of managing APIs throughout their lifecycle, from the early design and build stages to their publication, consumption, and monitoring. An API management platform typically covers the functionality of API Manager, API Catalog and API Gateway

API Orchestration

API orchestration is the process of integrating one or more applications into a single unified offering. Often, it's used to merge API calls into a single front-end, automate processes, or merge multiple internal APIs from a user experience perspective.

4.2.2 Digital Twin Management Cluster

The ABBs of the Digital Twin Management Cluster are essential for managing the lifecycle and discoverability of Digital Twins and their related aspects.

Digital Twin Registry

This ABB provides functionalities to create and store Digital Twins and related meta data. It supports the assignment of aspects to twins, including references to aspect models and endpoints, and control the access to them.

Digital Twin Search and Discovery

This ABB provides functionality to search for and discover digital twins and their related aspects including the aspect endpoints based meta data.

4.2.3 Collaboration Cluster

The Collaboration Cluster provides the functionality for governance, identity, and process management necessary for secure and compliant cross company collaboration. It also includes tools for semantic alignment and reference management.

Stable Reference Management

This ABB provides functionality to provide, manage and resolve stable references to work products over the complete life-cycle of a work product.

Digital Twin (native) Apps

This ABB provides functionality to access and visualize data, accessible via Digital Twin aspects and present them to users.

The Business Glossary

This ABB provides functionality to manage glossaries serve as references for business term meaning (term semantic).

Business Partner Management

This ABB provides functionality to provide unique IDs for Business Partners, govern, consolidate and enrich Business Partner Data and screen Business Partners against various lists (e.g. sanction party, fraud).

Contract Management

This ABB provides functionality to support creating, executing, and managing contracts between two or more parties. It involves the administration of contracts throughout their lifecycle, from initial negotiation and drafting to contract execution, performance monitoring, and eventual contract closure.

Policy Management

This ABB provides functionalities to define, maintain and release a documented set of basic principles and associated guidelines, formulated and enforced by the governing body or an assigned committee of an organization that direct and limit decisions and actions in pursuit of its objectives.

Identity and Access Management

This ABB provides functionality to manage processes, policies, and technologies required to manage and control access to digital resources within an organization. It involves the management of user identities, company identities, authentication, authorization, and privileges.

App Management (Portal)

This ABB provides functionality to manage the processes, tools, and strategies used to manage and govern applications within a portal environment. It involves the administration, deployment, monitoring, and maintenance of applications that are hosted and accessed through a portal.

Process Management

This ABB provides functionality to model, orchestrate and manage business processes in the context of collaborative engineering projects and digital twins.

4.2.4 Data Management Cluster

The Data Management Cluster provides the functionality for storing, modeling, and managing the data and semantics that underpin Digital Twins. It supports both structured and semantic data management across domains.

Data Storage

This ABB provides functionality to persistently store relevant engineering and operational data in a structured and secure manner. This includes support for various data types—such as structured, unstructured, and metadata – ensuring that information is reliably captured, organized, and retrievable throughout the system lifecycle.

Engineering Applications

This ABB provides functionality to enable the practical implementation of engineering principles, methodologies, and technologies across various domains (e.g. system modeling). Authoring tools are a key component of engineering applications. They provide capabilities for the creation and manipulation of engineering data, allowing users to generate, edit, and manage structured technical content such as models, specifications, and documentation.

Field Data Management

This ABB provides functionality for managing data generated by physical assets in real-world environments. This includes the ability to store, process, and track operational data such as sensor readings, performance metrics, and status information. The goal is to ensure data integrity, traceability, and availability for analysis, diagnostics, and decision-making.

Cross Domain Variant & Configuration Management

This ABB provides functionalities to manage product variants and their configurations across various domains, leveraging Stable Reference Management. It ensures that exchanged data is accurately associated with a specific variant or configuration, while enabling clear differentiation and traceable management.

Data Modeling

This ABB provides functionality to enable the definition, management, and storage of both structural and semantic models that form the foundation of the CDT framework. It includes Aspect Modeling, which provides the capability to create and maintain aspect models. These models are stored in the Aspect Model Repository along with descriptive metadata for each model entry. Additionally, Ontology Modeling the functionality to create, change and manage ontology models. The resulting ontologies are stored in the Ontology Repository.

4.3 Architecture Variants for Data Exchange

The presented CDT framework supports different architectural variants for data exchange, each designed for different technical and operational requirements. These variants define how data flows between participants and which infrastructure components are involved in the exchange process. In the following a variant without temporary data storage und a variant with temporary data storage is described.

Variant 1: No Temporary Data Storage

This variant represents the default approach within the CDT framework (Figure 23). Data exchange is handled directly via API calls, without the need for intermediate file transfer or centralized data storage. Metadata required for the exchange – such as file location and access credentials – is embedded within the API payload. This approach aligns with CDT's federated architecture principles, ensuring that data remains with the source and is accessed only when needed. It minimizes legal and security complexity and supports real-time, lightweight data exchange across organizational boundaries. Most use cases within CDT can be realized using this variant.

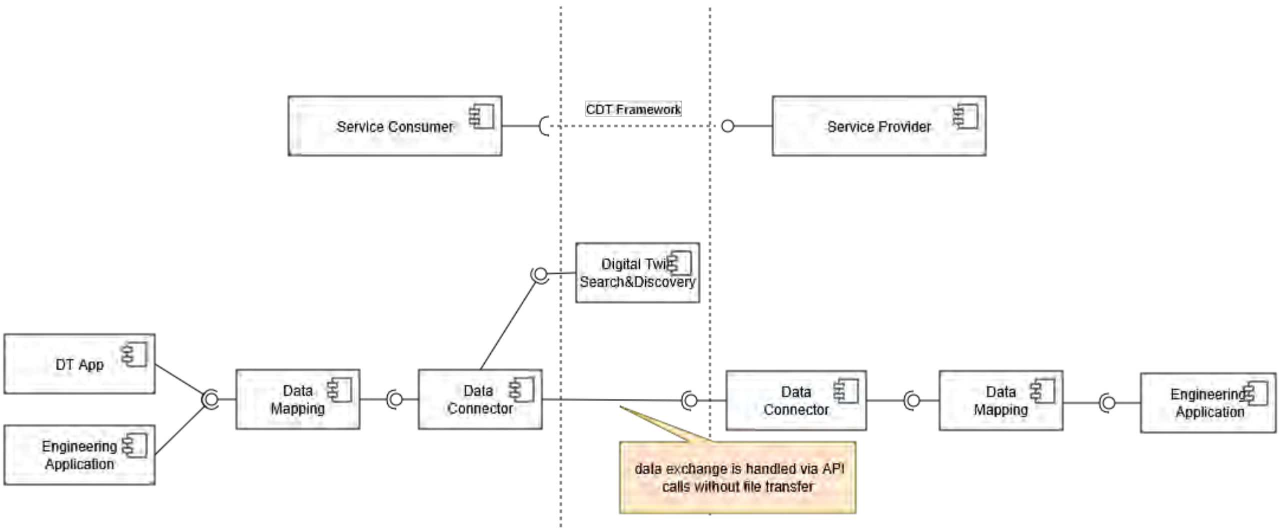


Figure 23: Data exchange without temporary data storage (variant 1)

Variant 2: Temporary Data Storage

In scenarios where file transfer is required, due to data size limitations or system capabilities, this variant introduces a temporary data storage component (Figure 24). The service provider pushes the data to this storage, and the consumer retrieves it via pull mechanisms. Metadata about the transferred file like location or access credentials is still handled through API calls as explained in variant 1. While this approach enables the exchange of large files, it is only used when necessary. In most cases, the CDT IT architecture does not rely on temporary data storage.

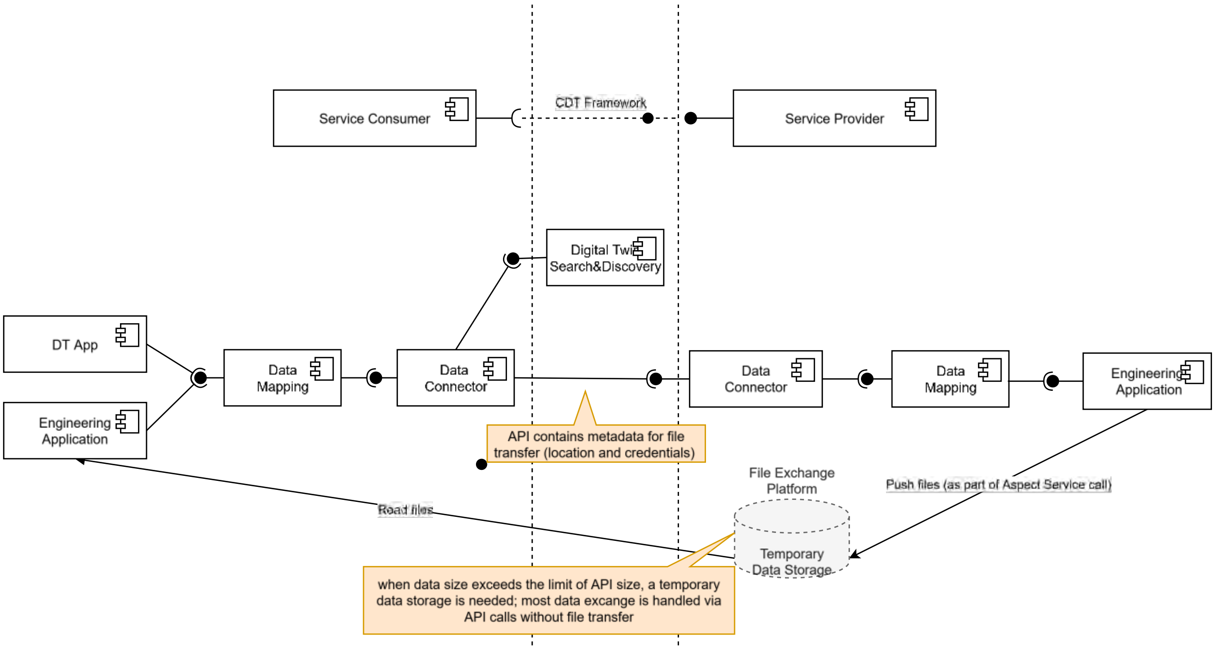


Figure 24: Data exchange with temporary data storage (variant 2)

4.4 Basic Interaction Flow (BIF)

The Basic Interaction Flow (BIF) describes the standardized sequence of steps that describe how data exchange between two parties is initiated, authorized, and executed within the CDT framework. It outlines the roles of the consumer, the CDT ABBs and the data provider. This sequence ensures that all interactions comply with the architecture constraints and principles described in chapter 4.1. The complete BIF is visualized in Figure 25.

The process begins with the consumer logging into the CDT platform (Step 1). The platform's integrated Identity and Access Management (IAM) component validates the user's authorization (Step 2). If the authorization is successful, the consumer is granted access to the Digital Twin Search and Discovery service (Step 3), where they can search for available Digital Twins and their associated aspect models. Once the desired aspect is found, the Contract Management component checks whether the consumer has the legal permission to access and use the data (Step 4). If the contractual conditions are met, the CDT platform publishes an aspect endpoint (Step 5), which the consumer can integrate into its Data Connector (Step 6).

The consumer then calls the published aspect endpoint (Step 7). Before the consumer receives the data, the provider needs to validate the request within the providers data connector (Step 8). If the request is authorized, the call is forwarded to the appropriate internal endpoint within the provider environment (Step 9), which connects to either Engineering Applications, Field Data Management systems or other systems providing the requested data (Step 10). The requested data is retrieved, mapped to the agreed aspect structure (Step 11), and made available via the provider's Data Connector (Step 12). On the consumer side, the received data is mapped from the aspect structure to the consumer's internal data structure (Step 13), enabling its use in internal Digital Twin Apps or Engineering Applications (Step 14).

This interaction flow ensures that data exchange is secure, traceable, and compliant with both technical and legal requirements. It also reflects the modular and federated nature of the CDT architecture, where each step is supported by dedicated ABBs.

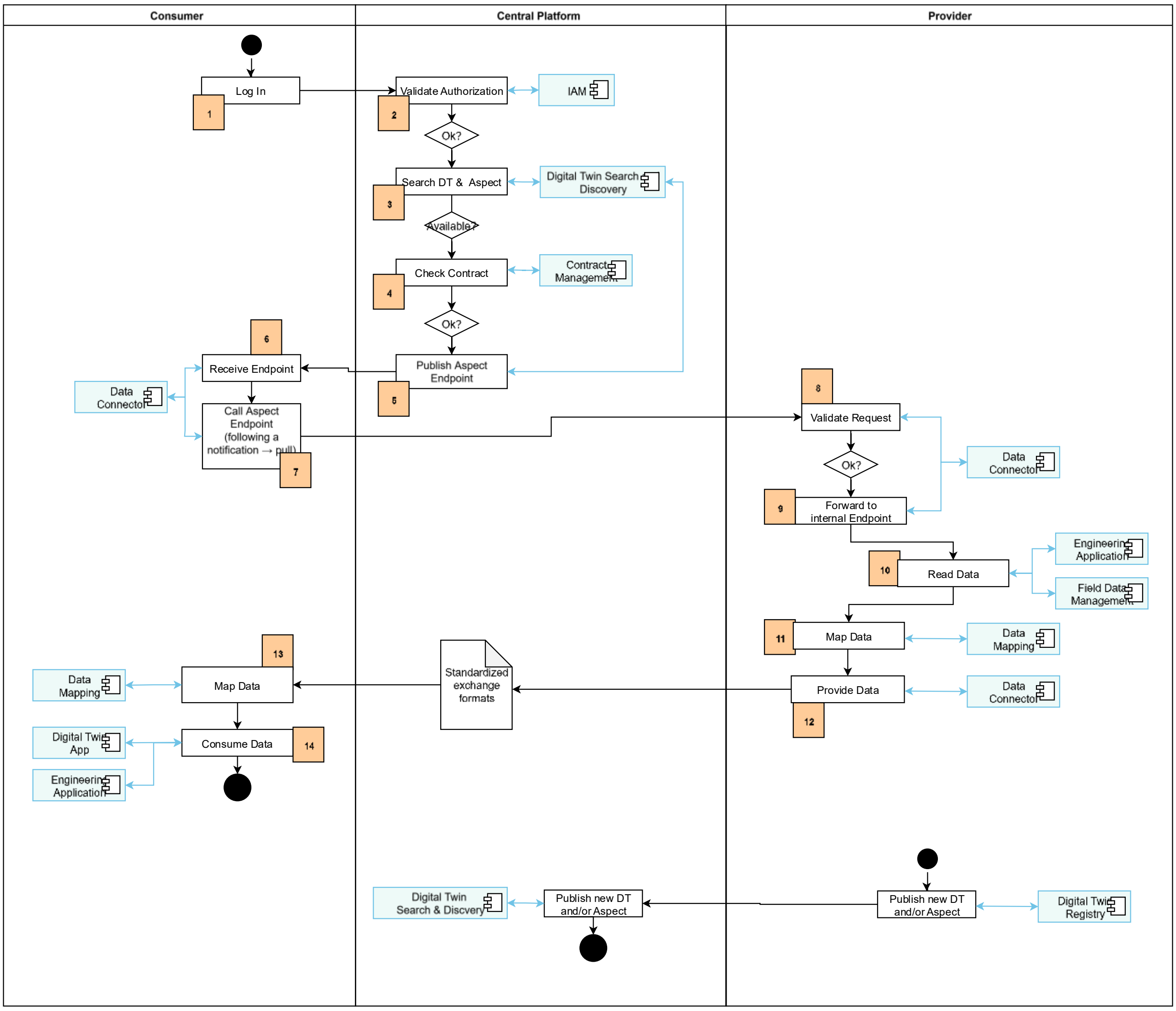


Figure 25: Basic interaction flow of the CDT architecture

4.5 Implementation Strategy for CDT Architecture

The implementation of the CDT architecture requires a clear strategy for establishing a suitable framework for integrating engineering applications. Different implementation strategies exist, each with specific advantages and challenges.

One approach is to use existing dataspaces such as Catena-X or Manufacturing-X. These ecosystems provide standardized components, including identity management, contract handling, and federated catalogs This option offers high interoperability and compliance with industry standards, while ensuring data sovereignty through a federated architecture. It also facilitates integration into broader industrial networks. However, it introduces dependencies on membership and governance structures and may require additional effort to adapt company specific aspects.

A second option is to build a proprietary data space using open-source components, e.g., Tractus-X. This approach provides maximum flexibility and avoids licensing costs while maintaining compatibility with Catena-X standards. It is particularly suitable for organizations with strong internal IT capabilities and a need for full control over customization. The main drawback is the significant implementation and operational effort, including expertise in security and infrastructure management.

Alternatively, companies can opt for a commercial data space solution offered by providers such as T-Systems or Cofinity-X. These services deliver “Data space-as-a-Service” and “Data space-Connector-as-a-Service”, including hosting, operation, and support under defined service-level agreements. This option enables rapid deployment and reduces internal resource requirements. However, it comes with higher recurring costs and limited flexibility for customization.

Beyond the data space setup, integration of engineering applications is also needed. Direct integration through vendor-provided APIs aligned with CDT aspect models would be ideal, but this scenario is highly unlikely in practice due to the prevalence of proprietary interfaces. A more pragmatic solution is the use of data mapping technologies, which transform proprietary data structures into CDT-compliant aspect models. While this approach offers flexibility and leverages existing APIs, challenges remain for example in accessing granular artifacts from complex architecture models.

An overview of the implemented CDT Architecture can be found in Figure 26. The CDT Demonstrator illustrates how the proposed architecture can be implemented in practice. The demonstrator utilized Eclipse Data Space Components, Catena-X principles, and mapping solutions to enable secure, synchronous data exchange between heterogeneous requirements management tools without transferring entire files. Only the necessary data points were exchanged, ensuring data sovereignty and reducing manual effort. This approach demonstrated significant business value, including faster collaboration across organizational boundaries, improved process efficiency, and scalability for additional use cases such as simulation and integration. The architecture consists of three partner environments, each hosting its own requirements management application (e.g., 3DExperience, ReqMan). These applications are connected to the data space via Data Mapping and API Management components, which transform proprietary data into standardized aspect models. The connection to the data space is established through Eclipse Dataspace Connectors (EDC), ensuring secure and controlled data exchange.

As a central component of the architecture lies the Federated Catalog and Identity Hub, which manage service discovery, identity, and access rights across all participants. The data space enables synchronous, point-based data exchange rather than file-based transfers, preserving data sovereignty and reducing manual effort. Semantic alignment is achieved through Ontology Modeling and Management, Aspect Modeling, and a Business Glossary, which ensure consistent interpretation of exchanged data. Process orchestration is supported by Process Modeling capabilities, enabling governance and compliance.

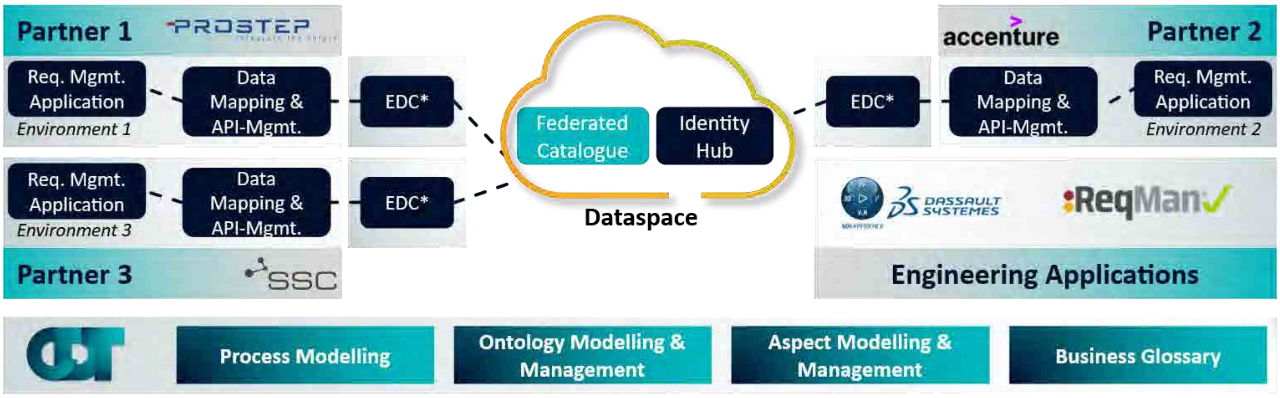


Figure 26: Set-up of the implementors laboratory for the CDT demonstrator.

For pilot projects, commercial data space services or open-source components (e.g., Tractus-X) are suitable starting points. For large-scale adoption, integration into existing Gaia-X compliant data spaces (e.g., Catena-X) is recommended to benefit from established standards and marketplace offerings. Regarding engineering application integration, mapping solutions should be prioritized as an immediate solution, while long-term efforts should focus on API standardization to enable seamless interoperability. An overview of implementations options can be found in Table 3.

Table 3: Implementation options

Item	Implementation Options	Links
Subscription to data space (incl. BPN)	T-Systems (Connect & Integrate)	See Website (Connect & Integrate)
Subscription to data space (incl. BPN)	Cofinity-X (Dataspace OS)	See Website (Dataspace OS)
Data space connector	EDC as a Service	+ 3-10 PD Cofinity-X, T-Systems
Data space connector	Internal implementation based on open-source components (Tractus-X)	5-100 PD (Tractus-X)
Connecting enterprise infrastructure to EDC	Internal implementation	5-10 PD
Data mapping for engineering applications	Internal implementation	3-5 PD
Data mapping for engineering applications	Implementor for data mapping	OpenPDM

5 User Journeys for Collaborative Digital Twins

As part of the CDT project, six User Journeys (UJs) (see Figure 27) were developed to explore and refine collaborative workflows that mimic what occurs during actual product development. This recommendation presents these User Journeys. The UJs were designed with the goal of being independent from specific tools and existing frameworks. This flexibility ensures that the UJs can be applied across organizations engaged in cross-enterprise engineering collaboration, particularly between OEMs and their partners, regardless of the tools or frameworks they currently use. By considering the existing tool landscapes and operational methodologies within different industries, the user journeys provide a scalable and adaptable approach that can be seamlessly integrated into diverse engineering environments.

However, it is important to note that the user journeys are not intended to be a one-size-fits-all solution. Each organization will need to adapt and tailor the user journeys to fit their specific context, business processes, and technological setup. As such, the user journeys should be seen as a flexible framework rather than a prescriptive, step-by-step recipe for implementation. The goal is to provide a foundation that can be detailed and customized to the unique requirements of each organization, enabling effective cross-enterprise collaboration while maintaining alignment with existing systems and workflows.

Figure 27: User Journey Overview

Each user journey is introduced with a simplified swim lane diagram, followed by a detailed table that provides an in-depth description of the steps involved, along with user-journey-specific requirements and methods. These user journeys represent a high-level starting point, developed by CDT members, with the aim of refining and adapting them for cross-partner collaboration. The intent is to offer a foundational framework that can be further tailored to suit the unique needs and contexts of the different stakeholders involved in the collaboration.

5.1 General Set-Up for the User Journey Implementation

5.1.1 Mars Rover Dataset

The Mars Rover dataset[25] is a reference dataset maintained within the prostep ivip Association. It was created to provide a realistic data basis for testing concepts developed within different project groups. Representing a complex cyber-physical system, the dataset includes artifacts from different lifecycle stages. The dataset contains several types of engineering information, such as a structured set of requirements, detailed CAD models, and system architecture models. This allows the Mars Rover dataset to be used for different industrial use cases.

For the CDT Demonstrator, only the requirements subset of the Mars Rover dataset was utilized to implement User Journey 0 (Collaboration Model Setup) and User Journey 1 (Collaborative Requirements Content Management). These requirements provided the necessary input for validating the principles of the CDT IT-Architecture and used ABBs. However, the dataset also includes CAD data and system models, which can be leveraged for future extensions of the CDT Demonstrator, such as collaborative system design, simulation, and integration scenarios.

The requirements set of the Mars Rover used in the demonstrator is structured to reflect the complexity of a real-world engineering scenario. It covers different domains and lifecycle aspects, including functional and performance specifications as well as structural definitions for key components, and operational constraints. The requirements address fundamental aspects such as motion capabilities, energy management, control and navigation, environmental conditions, and compliance with safety and legal standards. The Mars Rover dataset provides a robust basis for the CDT Demonstrator, enabling the implementation of UJ0 and UJ1 in a realistic and industry-relevant context. Its scope and inclusion of additional artifacts such as CAD data and system models make it an ideal candidate for extending the CDT framework to further user journeys and industrial use cases.

5.1.2 Overarching Requirements

To ensure seamless collaboration across all user journeys (UJ0, UJ1, UJ2, UJ3, UJ5), a set of overarching requirements (Table 4) has been identified. These requirements provide a foundational framework, ensuring interoperability, security, and accountability for cross-enterprise engineering collaboration. It is important to note that these overarching requirements apply across all user journeys. Specific requirements that are unique to each user journey will be introduced in their respective chapters. Table 4 below summarizes the key overarching requirements, categorized by their focus area.

Table 4: Overarching Requirements Ensuring Seamless Collaboration Across User Journeys

ID[26]	Focus Area	Requirement
R100	Access Rights	Every time data is exchanged between the OEM and the partner, access rights related to both contractual agreements and data usage shall be verified before the data is consumed.
R101		End users shall only be granted access to data that they are authorized to view or modify, based on their specific role and project permissions.
R102		Users granted project access shall be provided with context-relevant information at the highest level, appropriate to their role and project scope.

[25] https://prostep-ivip.atlassian.net/wiki/spaces/MR/overview?homepageId=983042
[26] For making referencing easier, all steps, requirements, and methods have been assigned a unique identifier in scope of the CDT project.

ID[26]	Focus Area	Requirement
R200	IP Management	IP ownership must be clearly defined by the creator and displayed on all shared artifacts, enabling transparency on intellectual property across all parties.
R201		Each artifact that includes intellectual property must be tagged with references to the terms and conditions governing its use, agreed between the OEM procurement agent and the partner contract manager.
R300	Notification Management	The workflow system shall support task orchestration between the OEM and partner, sending notifications for task sequence updates in real-time.
R301		Any change in the status of shared artifacts (e.g., requirements) shall trigger automatic notifications to all subscribed users, informing them of the update at each workflow stage.
R400	Traceability	Every action taken by each partner within the project must be traceable, documenting who performed the action, what was done, when, and why (with comments or linked objects). This is necessary to ensure accountability and to manage liability issues.
R401		OEM procurement agent & partner contract manager shall be able to review/have access to all exchanges between OEM & Partner

These requirements act as guiding principles for the development and implementation of collaborative workflows within the context of the CDT. By addressing critical areas such as access rights, IP management, notifications, and traceability, they aim to establish a robust and secure framework for cross-enterprise engineering collaboration.

5.1.3 Roles

Within the defined UJs, role definitions are used to structure tasks based on competencies. It is important to note that these definitions are intended as a flexible framework rather than a fixed standard, recognizing that roles and their designations can vary significantly across companies and industries.

The provided role definitions group competencies and process-related activities to create a shared understanding of responsibilities within cross-enterprise collaboration. Enterprises are encouraged to adapt these definitions to their specific contexts by mapping the outlined tasks and responsibilities to their own role structures, based on competencies and activities performed.

In some cases, a role may exist on both the OEM and partner side, while in others, roles are exclusive to one side. The Stakeholder column below indicates where a particular role resides.

Table 5: Adapting Role Definitions to Enterprise Needs

Role Name	Alternative Titles	Stake-holder	Details	Occurrence
Project Leader	Chief Engineer, Program Manager, Chief Engineer	OEM + Partner	• Authorizes the project and triggers work initiation. • Coordinates resources and resolves conflicting targets across disciplines.	UJ 0-5

Role Name	Alternative Titles	Stake-holder	Details	Occurrence
			• Sets project goals and oversees Board Reviews. • Focuses on overall delivery (cost, quality, time) and manages schedules, budgets, and contractual frameworks.	
Lead Engineer	Design Manager, Perform-ance Leader	OEM + Partner	• Manages engineering tasks within their technical discipline or domain scope. • Sets high-level targets and negotiates task breakdowns. • Provides technical leadership and conducts reviews. Includes roles like requirements managers, simulation managers, and lead systems engineers.	UJ 0-5
Specialist		OEM + Partner	• Executes technical activities (design, simulation, testing) to generate project deliverables. • Reports to Lead Engineer. • Focuses on detailed, task-specific content creation.	UJ 0
Design Quality Assessor	Technical Auditor, Quality Auditor	OEM	• Defines quality criteria, standards, and auditing processes for their domain. • Ensures compliance with quality processes and evaluates deliverables at milestone reviews. • Maintains enterprise-wide quality standards in collaboration processes.	UJ 0+2
Procurement Agent	Supply Chain Specialist, Supplier Manager	OEM	• Manages partner relationships, including IP protection and collaboration frameworks. • Establishes platform access, role definitions, classification, and collaboration environments.	UJ 0-5

Role Name	Alternative Titles	Stake-holder	Details	Occurrence
Contract Manager	External Contract Coordinator	Partner	• Negotiates collaboration and contractual frameworks with the OEM. • Ensures compliance with agreements and protects intellectual property.	UJ 0
Design Engineer / Requirements Manager **(Specialist)**		OEM + Partner	• Responsible for the conceptualization and detailed design of components or systems • Define requirements (and related features and functions) and specifications • Breaks down requirements from high level to system and component level, together with system engineers and simulation engineers • Ensures requirements are consistent with stakeholder expectations and project goals • Manages requirement changes and ensures traceability	UJ 1
System Engineer **(Specialist)**		OEM + Partner	• Manages system-level architecture and ensures alignment with project requirements. • Oversees interface definitions and control documents, ensuring consistent communication between sub-systems • Integrates supplied sub-systems, evaluates change requests, and manages updates to system designs and architecture • Drives maturation, freezing, and release of sub-system designs linked to project requirements	UJ 2
Simulation Expert **(Specialist)**		OEM + Partner / Service Provider	• Agrees on modeling approaches, architecture design standards, and specifications for simulation models • Develops system simulation models, proposing interface definitions for sub-systems. • Supports requirements engineer in breaking down high level targets and	UJ 3

Role Name	Alternative Titles	Stake-holder	Details	Occurrence
			checking feasibility / balance by simulation • Conducts simulations to assess the performance and reliability of designs and record requirements coverage • Freezes and releases integrated system simulation models and results baselines	
Testing Expert (Specialist)		OEM + Partner / Service Provider	• Defines testing scope and objectives, trying to optimize testing plan for requirements coverage, lead time and cost. • Ensures the reliability and dependability of testing by adherence to quality and methodology standards • Designs and performs tests to verify product or system performance against requirements. • Defines configuration, manufactures and assembles system prototypes, integrating components to match defined configurations	UJ 5

These role definitions are intentionally generic to accommodate the diverse needs of cross-enterprise collaboration. Enterprises may use different names for the same roles or combine multiple roles into one, depending on their organizational structure and operational requirements. When implementing the workflows and User Journeys defined in this document, it is recommended to map the tasks and responsibilities outlined here to the relevant roles within your enterprise.

5.2 User Journey 00: Collaboration Model Setup

The overarching user journey establishes the foundational processes and technical alignment required to enable all subsequent user journeys, covering early-phase collaboration, system configuration, and the closure of the CDT. From a business viewpoint, it somehow digitalizes part of the contractual framework among the involved parties.

5.2.1 UJ0 Flowchart

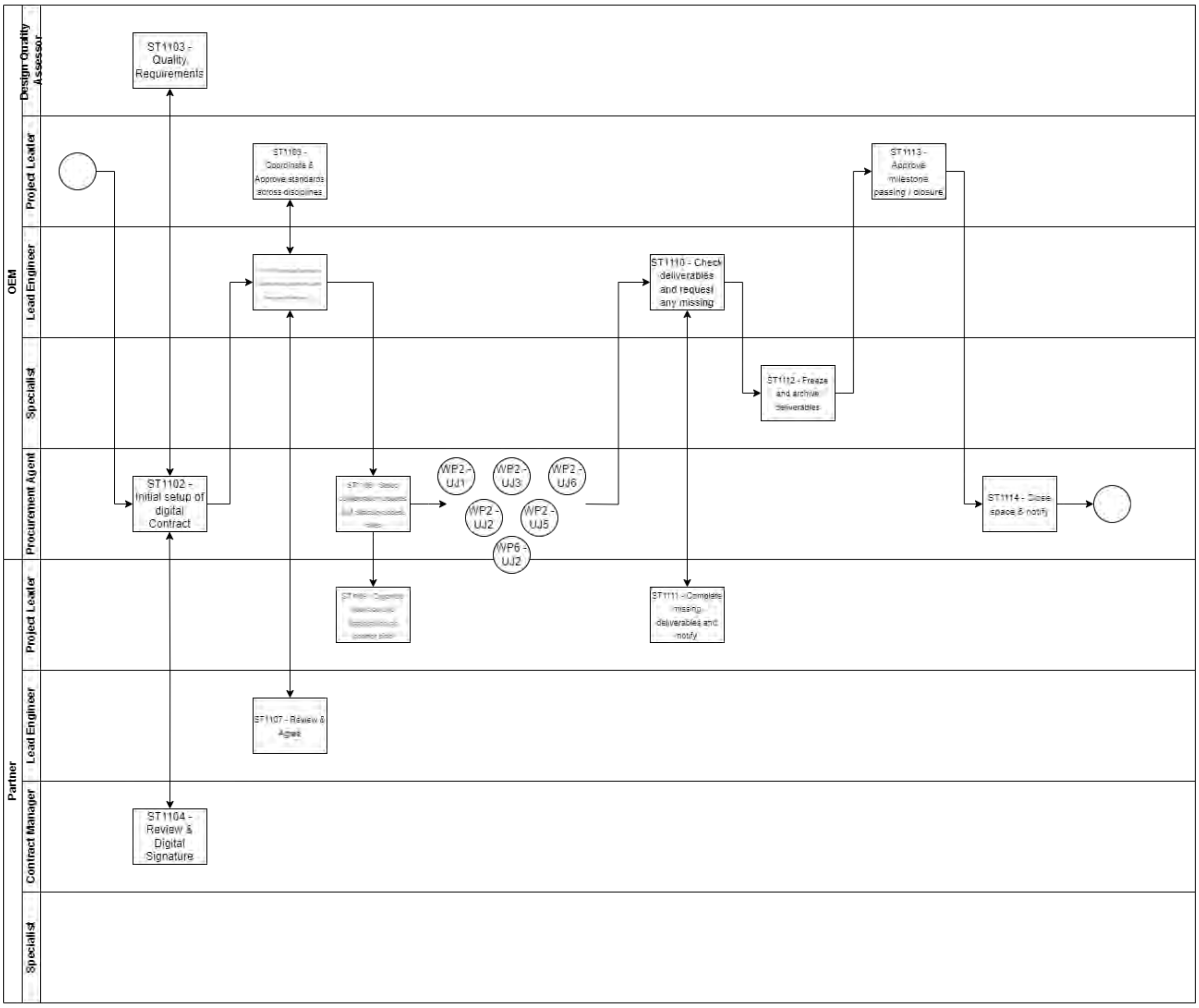


Figure 28: Flowchart UJ0

Table 6: User Journey 0 Steps

ID	Explanation / Comment
ST1102	Confirm Partners and work content. Propose contractual framework & agree with partner contract manager. Set up contract. Approve disclosure to partner supply chain & corresponding rules. Define liability principles. Agree on quality requirements, integration & quality assurance roles, responsibilities & accountability.
ST1103	Provide quality requirements, etc. to go into contractual terms.
ST1104	Share comments on & agree on contractual framework. Request disclosure to partner supply chain. Sign contract.

ID	Explanation / Comment
ST1105	Propose way-of-working method / collaboration framework (including data structure/formats and compliance to international standards, collaboration workspace/information transfer mechanisms, process for data exchange rules and process solving issues and escalations)
ST1106	Coordinate definition of Ways of working & data structure/formats /standards across disciplines
ST1107	Agree on collaboration framework
ST1108	Set Up Interfaces and Organization: 1. Nominate Focal Points (responsible for overall coordination of activities related to the program) 2. Designate an appropriate representative who is responsible for: Set up and management of the collaboration environment; Hardware set up; Software set up; Account set up and role assignment; Creation, maintenance, and deletion of user accounts; Data archiving/data retention rules 3. Define a Support System Set up technical platform enabling dataflows and to monitor data access and IP. In case of (partially) centralized architecture: Set up shared platform additionally enabling OEM and Partner to share, link, comment and manage configuration of project data. Create project collaboration space. Provide access to partner with dedicated role(s) Notify Partner
ST1109	Organize Interfaces/data flow & organization on partner side
ST1110	Check data exhaustivity vs contractual requirements, identify & request missing partner data
ST1111	Complete missing data & notify OEM
ST1112	Archive data (=make data available for OEM & Partner in read-only status for the mid-term after collaboration phase closure)
ST1113	Approve closure of the project or collaboration phase with partner.
ST1114	Close access to collaboration platform and archive contents. Notify partner.

5.2.2 UJ0 Requirements & Methods

Table 7: User Journey 0 Requirements & Methods

Type	ID	Link to Step	Explanation / Comment
Requirement	R1000	N/A	Manage artefacts (e.g. requirements) maturity status & corresponding modification rights, coverage status
Requirement	R1001	ST1108	Versioning capability and consideration in access rights & roles concept (WP6)
Requirement	R1002	ST1104	Contractual Framework shall cover following aspects: • Applicable regulations are identified, supporting the project (consider worldwide business relationships, governments, and regional authorities) • Statement of Work has been defined (work scope defines category of supplier relationship, such as design and build to spec, design or other intellectual services, equipment) • Export control and Intellectual Property (IP) agreement • Data Exchange Rules and Processes • Project Management Terms • Monitoring and Management of Contract Execution and Contractual Coverage of Evolution Requests • Anticipation and Mitigation of Contractual Risks (Shared risk analysis, mitigation action plan, Contract amendment) • Review of contractual IP expectations for each participating stakeholder and partner • Expansion upon contractually-defined IP protection • Definition and implementation of IP protection-compliant processes for the collaboration
Method	M1000	N/A	Consider strategies for: • Integration of existing artefacts • Use-case driven architecture • Model purpose definition • Legal / Obligations to provide proof & Compliance
Method	M1001	N/A	Consider: • Level of alignment / detailed modeling • Interface & Mapping definition

Type	ID	Link to Step	Explanation / Comment
Method	M1002	N/A	The OEM/Steering partner act as an integrator and ensures the communication of relevant data between Partners in more complex collaborations e.g. to develop an ADAS Platform

5.2.3 UJ0 Demonstrator Result and Recommendations

The CDT demonstrator, implemented within the Implementors Laboratory, serves as proof of concept for the Collaborative Digital Twin framework. It operationalizes the CDT architecture and validates key principles through the implementation of User Journeys 0 and 1. This section focuses specifically on User Journey 0 (Collaboration Model Setup), which establishes the foundational processes and technical alignment required for all subsequent collaborative workflows. UJ0 ensures governance, identity management, and technical onboarding before any data exchange occurs, thereby creating the prerequisites for secure and efficient collaboration. In the demonstrator, three heterogeneous partner environments, provided by Accenture, SSC, and PROSTEP AG, were integrated, each using different distinct technologies for data mapping and connectivity to the dataspaces. In the demonstration scenario one OEM (represented by PROSTEP) is exchanging various requirements with partners (represented by Accenture and SSC). Accordingly, each environment is represented by one requirements management system hosted by the mentioned partners. The following section explains how UJ0 was implemented within these environments and highlights the key challenges.

Accenture implemented its environment using Polarion as the requirements management tool. The team configured and deployed the Polarion environment, including repository setup and user access control, and created the Mars Rover project within Polarion. The alignment with the CDT architecture and its Architecture Building Blocks was achieved through external mapping and data exchange mechanisms aligned with the overall demonstrator architecture. During implementation, Accenture faced challenges, particularly in understanding the operational model and integration approach for Data Space and the T-Systems connector and understanding the Data space APIs.

SSC implemented its environment using the T-Systems Catena-X Portal as the primary gateway to the data space. The integration relied on APIs provided by the Connect & Integrate App, which exposed the EDC core API. SSC used these APIs to establish connectivity and enable data exchange, ensuring compliance with Catena-X principles. The C&I App effectively served as the connector within the CDT architecture, providing the necessary interface for secure and standardized communication. While the overall approach was similar to other Catena-X environments, SSC's implementation featured a slightly different user interface and was characterized by a high degree of automation. The main challenge during implementation was the extremely short time frame following the design phase, which required rapid deployment of a novel technological solution.

PROSTEP AG represented the OEMs engineering infrastructure consisting of CATIA Magic Cyber Systems Engineer as architecture application as well as Dassault´s 3DExperience Platform as requirements management system. This infrastructure was seamlessly connected to the EDC by the OpenPDM integration technology. PROSTEP's standardized connectors enabled the integration of the 3DExperience Platform and established a link to the EDC connector provided by T-Systems, granting access to the Catena-X data space.

The implementation of UJ0 within the CDT Demonstrator demonstrated that a federated data space approach can integrate heterogeneous environments. Despite differences in tools and integration strategies, all partners were able to connect through standardized EDC interfaces and using CDT principles. Key lessons learned include the importance of early clarification of integration concepts, such as the operational model of the data space and API usage. This Implementation forms the foundation for extending the CDT Demonstrator to additional user journeys and industrial use cases for collaborative engineering.

5.3 User Journey 01: Collaborative Requirements Content Management

This user journey facilitates requirements delivery from the OEM while fostering interactive collaboration with the partner to ensure mutual alignment and input. The digital collaboration and tracing of requirements setting approval, acceptance and change is particularly important as it is strongly linked to contractual responsibilities and compliance obligations.

5.3.1 UJ1 Flowchart

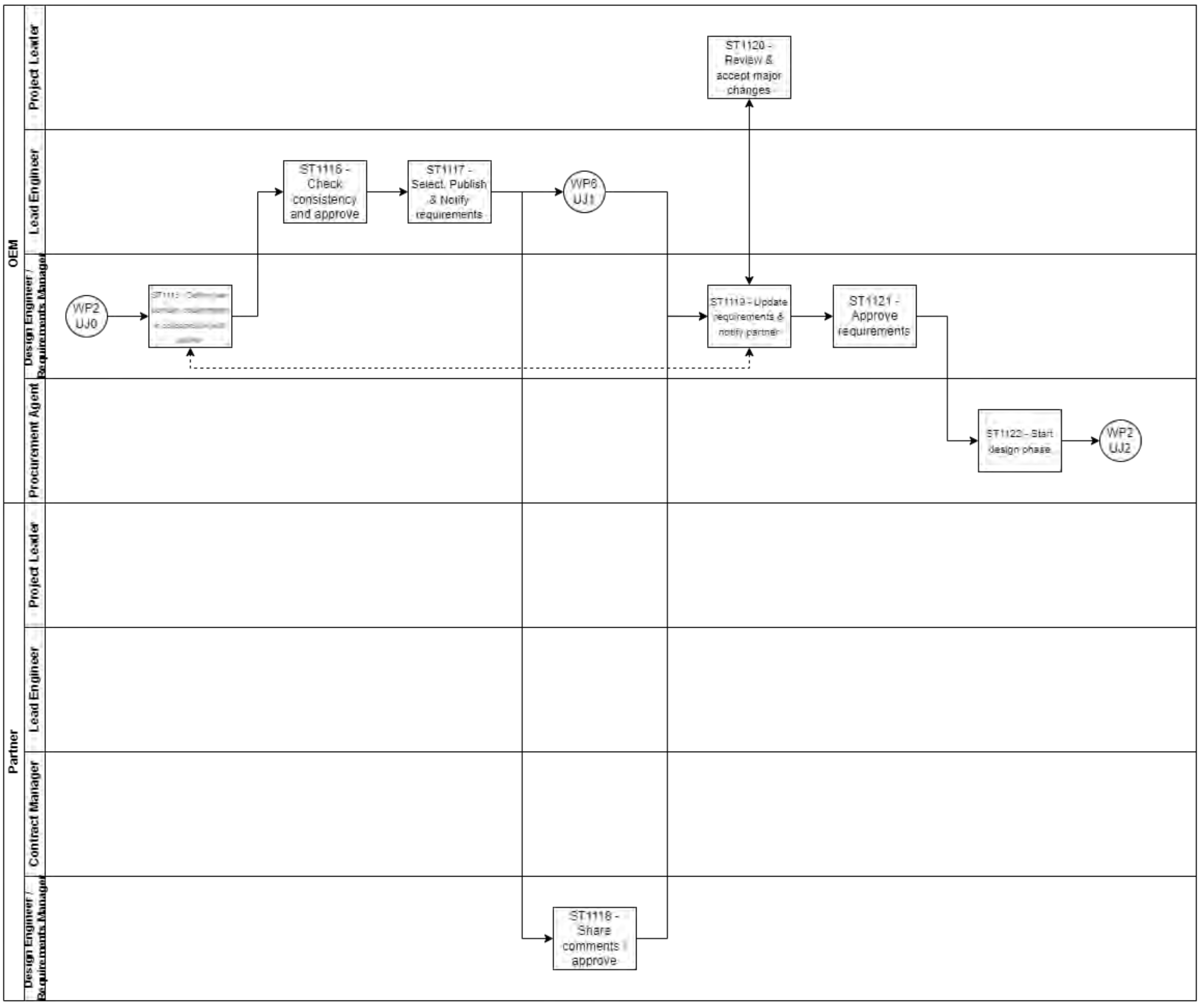


Figure 29: Flowchart User Journey 1

Table 8: User Journey 1 Steps

Step ID	Step Explanation
ST1115	Define requirements (and related features and functions) and specification in collaboration with partner
ST1116	Check consistency & approve requirements (and related features and functions) identify requirements incompatibilities and define corresponding variation points
ST1117	Classify Data (e.g. Export Control / Third Party Proprietary) Select data to be shared / Generate 100% or 120% view from 150% configuration (features/functions/requirements), define if to be visible, commentable, modifiable and/or downloadable and notify partner
ST1118	Share comments & proposals for information objects (features/functions/requirements)
ST1119	Accept & update requirement set with relevant info & notify partner
ST1120	Accept requirements update if schedule and/or budget impact
ST1121	Approve requirement set
ST1122	Approve system design phase start

5.3.2 UJ1 Requirements & Methods

Table 9: User Journey 1 Requirements & Methods

Requirement ID	Requirement Explanation / Comment
R1003	Manage requirement maturity status & corresponding modification rights, test coverage status for specific requirements
R1004	Manage various types of requirements, i.e. customer, functional, FuSaf, internal, standards/normative, regulations... with different attributes and relations to other entities
R1005	manage a common requirement lifecycle status / workflow view that is either unique for everybody, or has an unambiguous mapping
R1006	Notify automatically subscribers of requirement status update (for all steps)
R1007	Partner may also need to involve his supply chain

5.3.3 UJ1 Demonstrator Result and Recommendations

This section describes the implementation of UJ1 – Collaborative Requirements Content Management within the CDT demonstrator. UJ1 builds upon the collaboration model established in UJ0 and demonstrates how requirements can be exchanged and synchronized across heterogeneous environments without relying on traditional file-based transfers. This user journey validates the ability of the CDT architecture to enable secure, point-based data exchange through a federated data space, ensuring data sovereignty and semantic consistency. The demonstrator covered three heterogeneous partner environments, provided by Accenture, SSC, and PROSTEP AG, each using different requirement management systems. This diversity reflects real-world complexity and highlights the flexibility of the CDT architecture described in Chapter 5.5. To achieve interoperability, all partners agreed on a common data model based on a Catena-X aspect model for requirements, which provided a unified understanding of the data and harmonized semantics across systems. This alignment was essential for enabling consistent interpretation and traceability throughout the collaboration process. The following paragraphs describe the implementation of UJ1 within these environments and summarize key challenges.

Accenture implemented its environment with a strong focus on data mapping and traceability. The team developed mapping logic aligned with requirement IDs provided by PROSTEP AG to ensure consistent synchronization across systems. To support secure and scalable storage, Azure Blob Storage was provisioned as the persistent data layer within Accenture's cloud environment, enabling reliable handling of requirement data. Furthermore, a dedicated data transfer interface was created to import data into Polarion and publish updates to the EDC. The main challenges encountered included handling REQIF import failures, as Polarion rejected the provided reqif-files due to format inconsistencies or unsupported attributes. Additionally, attempts to migrate requirements from Excel led to incorrect mapping of requirement IDs, which impacted traceability and required corrective measures.

SSC implemented its environment using ReqMan as the requirements management tool. ReqMan provided an API for synchronizing incoming data, enabling automated consumption of updates from the data space. The synchronization process was executed by a fully automated script that covered all steps of the Catena-X data sharing paradigm, including catalog requests, contract negotiation on the control plane, and data transfer on the data plane. This script could be triggered manually via terminal commands, menu buttons, or executed automatically as a batch job. For outgoing data, ReqMan offered a menu button to initiate synchronization, sending data according to Catena-X principles with assets, access policies, usage policies, and contract definitions. The actual data behind the asset was stored in Azure Blob Storage. To establish the connectivity to the data space SSC used the EDC, provided over the T-Systems Connect & Integrate App, purely via APIs and without relying on the C&I graphical interface to achieve a higher degree of automation. A Catena-X aspect model for requirements served as the common data format, and the ReqMan requirements management Tool provided seamless mapping to this model. SSC's implementation was highly automated and designed for ease of use, minimizing user effort despite the complexity of the underlying data sharing process. The main challenges were defining the scope of the implementation in relation to project objectives and ensuring feasibility within a very short timeframe.

PROSTEP AG implemented its environment using a fully integrated approach based on the OpenPDM platform. The solution handled connectivity to both 3DEXPERIENCE and the EDC, as well as mapping to and from the Catena-X aspect model. Data assets were uploaded to an AWS S3 storage as part of the exchange process. The workflow included selecting a requirement by ID, exporting it from 3DEXPERIENCE, mapping it to the Catena-X requirement model, uploading it to S3, and registering a new asset and contract definition in the EDC. For data consumption, the process involved selecting the target ID and contract ID, initiating the transfer, extracting and mapping the data back to the 3DEXPERIENCE format, and updating the dataset accordingly within the tool. This implementation leveraged out-of-the-box OpenPDM components for process orchestration, 3DX and EDC connectivity, and asset upload. While the demonstrator processes were triggered manually, they were designed to support full automation in future deployments. PROSTEP AG's solution was fully integrated into its platform, allowing components to be added or replaced for connecting other systems such as ALM-, PLM-, or ERP-Systems. The main challenges included defining a consistent mapping and unique identifiers for requirements across partners. Additionally, the initial set of requirements has been imported into CATIA Magic Cyber Systems Engineer according to the demonstration scenario, where the requirements have been defined. CATIA Magic Cyber Systems Engineer was seamlessly integrated to 3DEXPERIENCE, in which the requirements were managed.

The implementation of UJ1 within the CDT Demonstrator successfully proved that collaborative requirements management across heterogeneous environments is feasible when supported by a federated data space and a shared semantic foundation. By agreeing on a common Catena-X aspect model for requirements, all partners ensured interoperability and semantic consistency, enabling synchronization of requirement changes without exchanging entire files. This approach reduces manual effort, improves traceability, and strengthens data sovereignty. The demonstrator also highlighted critical success factors for scaling this concept: standardized dataspace connectors, robust mapping strategies, and automation to minimize complexity for end users. The implementation provides a basis for extending the CDT architecture to additional user journeys and industrial use cases for engineering collaboration across organizational boundaries.

5.4 User Journey 02: Collaborative System Design

Extensive collaboration between the OEM and partner enables streamlined feedback during the design phase, supporting agile and efficient system development. By working on the same baseline models, there is a faster iteration on architecture blocks, on their interfaces and requirements coverage.

5.4.1 UJ2 Flowchart

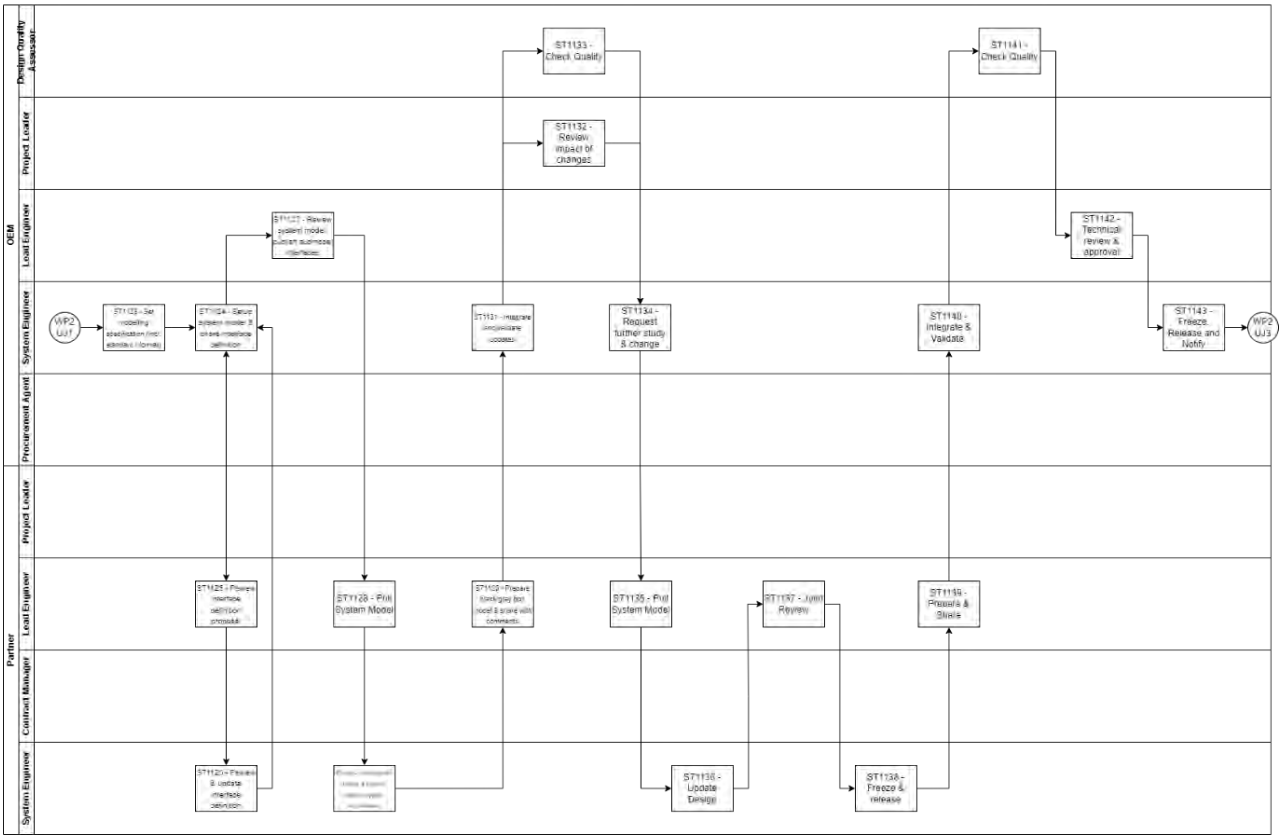


Figure 30: User Journey 2 - Collaborative System Design

Table 10: User Journey 2 Steps

ID	Step Explanation / Comment
ST1123	Align with Partner System Engineer on modeling approach & framework, Architecture System Design standards, modeling specifications
ST1124	Initiate the system architecture & behavioral models including proposal for definition of interfaces to sub-systems

ID	Step Explanation / Comment
ST1125	Review & Share interfaces definition proposal
ST1126	Review & update interfaces definition proposal (interface control document)
ST1127	Check consistency. Select shareable architecture system design subset / Generate 100% or 120% view from 150% configuration incl. interfaces for partner. Classify Data in the collaboration space (e.g. Export Control / Third Party Proprietary) & define access rights (visible, commentable, modifiable &/or downloadable). Notify partner
ST1128	Check/translate & share OEM model data with system engineer
ST1129	Initiate sub-system architecture design & prepare Share comments & proposals for requirements/ system design update
ST1130	Share & link own sub-system architecture elements as a black box (incl. interface control document incl. behaviors & links to related requirements) with OEM. Share comments & proposals for requirements / architecture system design update
ST1131	Integrate all supplied (sub)systems & analyse partner change request on requirements / system design
ST1132	Analyse schedule and/or budget impact & accept requirement update
ST1133	Check consistency & quality at vehicle system level. Identify need for Partner sub-system architecture element update
ST1134	Request change on Partner system architecture element
ST1135	Check/translate & share OEM model data with system engineer
ST1136	Mature and freeze sub-system design (incl. links to related requirements)
ST1137	Organize technical review & approve the sub-system model
ST1138	Release integrated sub-system model baseline & notify project team
ST1139	Share & link own sub-system architecture elements (incl. interface control document & links to related requirements) with OEM
ST1140	Integrate updated sub-system design, update system design accordingly and freeze it
ST1141	Check consistency & quality at vehicle system level
ST1142	Organize technical review & approve the integrated system model
ST1143	Release integrated system model baseline & notify project team & Partner

5.4.2 UJ2 Requirements & Methods

Table 11: User Journey 2 Requirements & Methods

Type	ID	Link to Step	Explanation / Comment
Requirement	R1008	ST1133 / ST1141	Automated checks to verify integrated model compliance to Design standards & Quality requirements (requires high model consistency & alignment on modeling framework & rules across the value chain)
Requirement	R1109	ST1124 / ST1125	TBD: Functional & technical requirements of Interface
Requirement	R1110	ST1126 / ST1124	Connection between overall collaboration platform and OEM & Partner data management platform
Requirement	R1111	ST1130 / ST1131	Automated checks to verify partner model elements compliance to System Design standards & Quality requirements
Method	M1003	N/A	In Agile mode, invite OEM to partner sprint/system demos to provide feedback on system under development

5.5 User Journey 03: Collaborative Simulation

In this user journey, the OEM prepares the product simulation environment for overall development integration and validation, while the partner retains responsibility for executing subsystem development simulations, ensuring an integrated yet distributed workflow.

5.5.1 UJ3 Flowchart

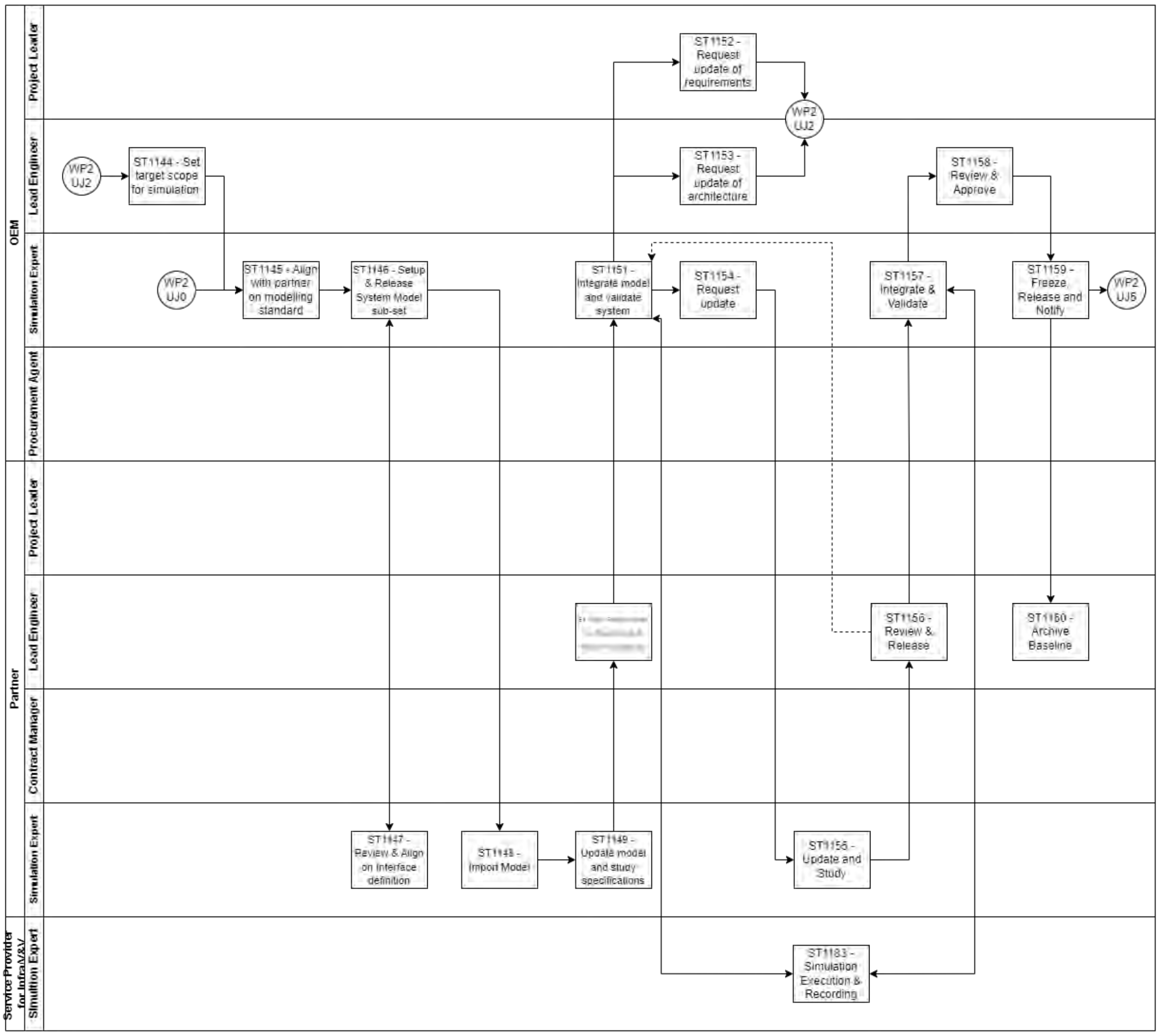


Figure 31: User Journey 3 - Collaborative Simulation

Table 12: User Journey 3 Steps

ID	Step Explanation / Comment
ST1144	Set target scope for simulation: requirements, detail, etc.
ST1145	Align with Partner Simulation expert on modeling standards and interfaces
ST1146	Release overall system structure model. Classify Data in the collaboration space (e.g. Export Control / Third Party Proprietary) & define access rights (visible, commentable, modifiable &/or downloadable). Select data to be shared / Generate 100% (or 120%) view from 150% configuration. Notify partner
ST1147	Technical agreement, including review & update interfaces definition proposal
ST1148	Access / Pull OEM model

INTERNAL

ID	Step Explanation / Comment
ST1149	Identify data (e.g. meta data, evidence) to be shared with the OEM. Complete simulation system structure with own sub-system elements
ST1150	Approve that the model behavior reflects the component expected requirements / behavior & release
ST1151	Integrate simulation model, executes V&V, performs simulation and record whether requirements are not met
ST1154	Request update on system model
ST1155	Update system structure with own sub-system elements
ST1156	Approve that the model behavior reflects the component expected requirements / behavior & release
ST1157	Integrate simulation model, executes V&V, performs simulation and record whether requirements are met
ST1158	Reviews and approve requirements fulfilment
ST1159	Freeze and release integrated system simulation model & results baseline & notify project team & Partner
ST1160	Freeze and release integrated sub-system model baseline & notify project team.
ST1183	Execute simulation and record whether requirements are met / not met. Involve tool vendor, if necessary, on complex cases.

5.5.2 UJ3 Requirements & Methods

Table 13: User Journey 3 Requirements & Methods

Type	ID	Link to Step	Explanation / Comment
Requirement	R1112	N/A	Same data model for UJ3, UJ5 (UJ5 = extension of UJ3)
Requirement	R1113	ST1146	Data model compliant with credible simulation process recommendation from prostep ivip + Model identity card (MIC core) from Prostep + SystemX + JAMBE Note: additional upcoming: ontology data model for simulation from SmartSE
Requirement	R1114	ST1146	Automated checks to verify partner model elements compliance to System model standards & Quality requirements. If black / grey box, check on quality / credibility is limited (ref. SmartSE upcoming guidelines for modeling quality requirements).
Requirement	R1115	ST1148, ST1151, ST1155	All partners shall be able to change parameters exposed by other partners.
Requirement	R1116	ST1158	OEM shall be able to automatically evaluate the performance by matching simulation results to

Type	ID	Link to Step	Explanation / Comment
			requirement where numerical KPIs are available (coverage analysis)
Method	M1004	N/A	Integration & Quality responsibility could be taken by a designated party, eg. if there are several OEMs & Partners involved for ADAS. Accountability in case of issues needs in this case to be defined before the collaboration start

5.6 User Journey 04: Collaborative Integration & Tests

This user journey extends the previous user journey with the inclusion of subsystem prototypes for integration and validation, ensuring systems meet functional and performance standards.

5.6.1 UJ4 Flowchart

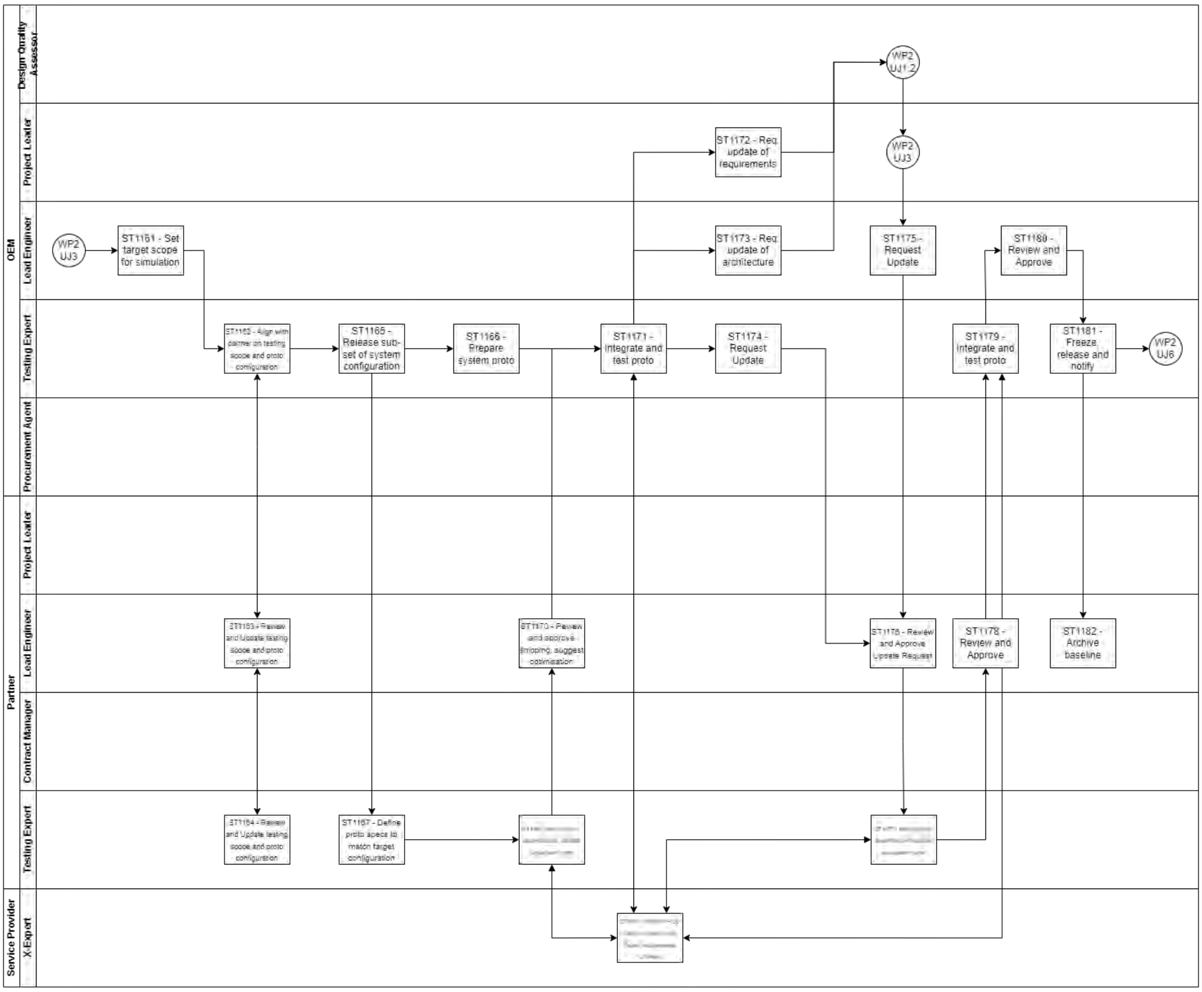


Figure 32: User Journey 4 - Collaborative Integration & Tests

Table 14: User Journey 4 Steps

ID	Explanation / Comment
ST1161	Define target objectives & scope for testing based on system requirements (from first version of the Design Verification Plan)
ST1162	Align with Partner Testing expert on testing scope & objectives, Muster/Maturity level, KPIs, product configuration & principles, expected OEM & Partner technical input on both sides
ST1163	Review & update testing scope & objectives, requirements, product configuration & principles
ST1164	Review & update testing scope & objectives, product configuration & principles
ST1165	Release overall system prototype configuration definition (incl. Mechanical, E/E, SW components) and requirements. Share sub-set of System configuration (e.g. based on 100% or 120% view from 150% configuration), simulation & testing environment data with the Partner.
ST1166	Manufacture & assemble system prototype
ST1167	Define sub-system prototype configuration & tests parameters/environment based on OEM requirements &/or system prototype configuration (incl. Mechanical, E/E, SW components)
ST1168	Manufacture & assemble sub-system prototype components (e.g. SW, ECU, mechanical parts). Validate sub-system by test and/or simulation. Deliver physical test sub-system prototype together with sub-set of configuration data (e.g. based on 100% or 120% view from 150% configuration), simulation/testing environment & tests parameters to OEM
ST1169	Perform HiL tests, analyse results and record requirements are met/not met
ST1170	Approve that the test sub-system behavior satisfies requirements / reflects the expected behavior & release, suggest requirements &/or system design update to optimise costs with similar performance
ST1171	Integrate system prototype, perform HiL tests, analyse results and record that requirements are not met
ST1173	Request update on system architecture &/or design
ST1174	Request update on system parameters (for SiLS or new SW for ECU in HiLS)
ST1175	Identify need to update sub-system design & request sub-system design update
ST1176	Update sub-system design
ST1177	(re)Manufacture & assemble sub-system prototype components. Validate sub-system by simulation and/or test. Deliver physical test sub-system prototype together with configuration data & tests parameters to OEM
ST1178	Approve that the test sub-system satisfies requirements / reflects the expected behavior & release
ST1179	Integrate system prototype, perform HiL tests, analyse results and record that requirements are met

ID	Explanation / Comment
ST1180	Reviews and approve requirements fulfilment
ST1181	Freeze and release integrated system test configuration & results. Baseline & notify project team & Partner
ST1182	Freeze and release integrated sub-system model baseline & notify project team.

5.6.2 UJ4 Requirements & Methods

Table 15: User Journey 4 Requirements & Methods

Type	ID	Link to Step	Explanation / Comment
Requirement	R1112	N/A	Same data model for UJ3, UJ5 (UJ5 = extension of UJ3)
Requirement	R1117	N/A	OEM shall be able to automatically evaluate the performance by matching testing results to requirement where numerical KPIs are available (coverage analysis)
Requirement	R1118	N/A	TBC: OEM should be able to share a 100% or 150% structure model for the partner to develop system for a specific product or a line-up (consistently with UJ2 and WP6)
Method	M1005	ST1165	For a release of a given function, a test catalog shall be defined by the Lead engineer and aligned with the Partner e.g. in a Design Verification Plan
Method	M1006	N/A	Define backwards from process end the data / technical input to be exchanged along the process
Method	M1007	ST1162	Consider detailed and simplified / dummy model depending on the level of the tests
Method	M1008	N/A	Model catalog (index) incl. link to relevant models (potentially stored on several repositories) to inform about which models & granularity are made available for the test(s)
Method	M1009	N/A	Consider strategies for parameters / specifications management together with system model management (also relevant to WP6)
Method	M1010	N/A	Consider strategies for: • Development based on control model • Validation based on single instances or complete product line
Method	M1011	N/A	Standardize meta-models between simulation & test in order to make planning, decisions and documentation more interchangeable and re-usable between simulation and test.

5.7 User Journey 05: Collaborative Manufacturing Engineering

This user journey describes a collaboration model in which a supplier delivers a complete product and production system, such as an engine and its fasteners, directly to an OEM's assembly line. The package includes the physical components, integration services and a digital twin with complete inventory data. This approach involves important decisions that are made early on to define requirements and ensure coordination. The following diagram illustrates the integrated partner delivery model for complex manufacturing. It shows how a supplier provides not only a pre-developed product subsystem (e.g., an engine), but also its physical fastening system and a digital twin. The process highlights key milestones such as the make-or-buy decision and final supplier selection to ensure seamless integration at the OEM. The following sections detail the implementation and key benefits of this integrated partnership.

5.7.1 UJ5 Flowchart

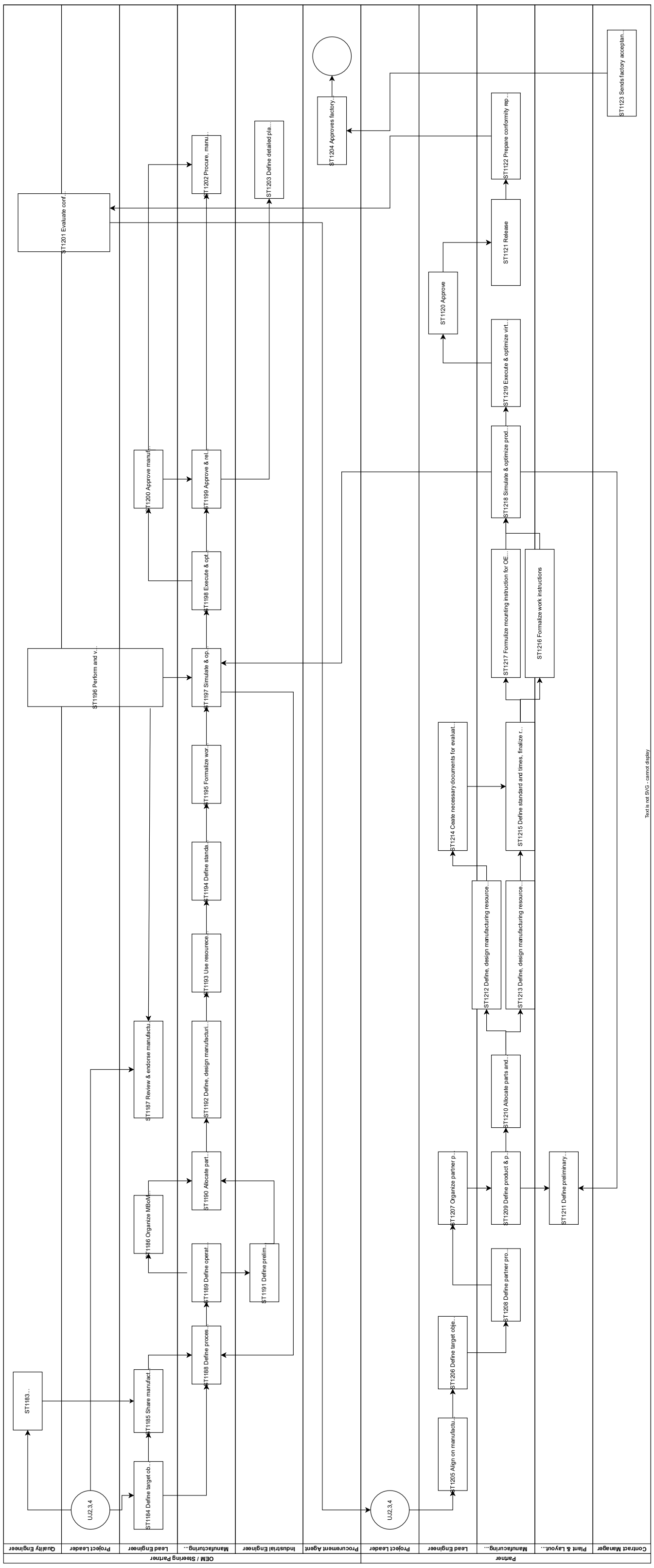


Figure 33: User Journey 5 Collaborative manufacturing engineering/definition of production means

5.7.2 UJ5 Requirements & Methods

Table 16: User Journey 5 Requirements & Methods

ID	Explanation / Comment
ST1183	Define detailed quality requirements for the product/system, covering specifications, standards, and acceptance criteria.
ST1184	Define target objectives, scope, and budget frame for all required manufacturing activities and define what must be achieved, what is included/excluded, and available funding.
ST1185	Share manufacturing & logistics requirements with Partner (space…) to communicate OEM needs to the partner responsible for manufacturing/production.
ST1186	Organize MBoM review & release to collect the Manufacturing Bill of Materials, reviews correctness, and formally releases it.
ST1187	Review & endorse manufacturing data (between EBOM and MBOM) to validate the consistency between engineering (EBOM) and manufacturing structure (MBOM).
ST1188	Define process flow & MBOM to design the overall manufacturing workflow and finalizes the MBOM structure.
ST1189	Define operations & steps to break down the manufacturing into discrete operations and sequential steps.
ST1190	Allocate parts & items to steps to map physical items to their corresponding operations or manufacturing steps.
ST1191	Define preliminary plant & production lines layout to create a first draft of the factory physical layout (stations, lines, flows).
ST1192	Define, design manufacturing resources (machines, jigs, tools) & allocate them to operations to determine what equipment is required and assigns resources to each manufacturing step.
ST1193	Use resources to create 3D views to generate 3D manufacturing visualizations (e.g., station design, ergonomics).
ST1194	Define standard and times; finalize routings to specify work durations, standard operations times, and production routing.
ST1195	Formalize work instructions to create detailed work instructions for operators.
ST1196	Perform and validate PFMEA to conduct Process Failure Mode and Effects Analysis to identify and mitigate risks.
ST1197	Simulate & optimize production flows & operations to use digital tools to simulate and improve manufacturing efficiency and throughput.
ST1198	Execute & optimize virtual commissioning and freeze to run virtual testing of equipment, automation, and logic before physical commissioning.
ST1199	Approve & release for final manufacturing definitions before execution.

ID	Explanation / Comment
ST1200	Approve manufacturing process for confirmation that the process design meets requirements.
ST1201	Evaluate conformity of partner pre-entries and validate quality requirements to check whether the partner's proposed manufacturing configuration meets quality standards.
ST1202	Procure, manufacture & validate resources for acquisition or to produce physical resources; ensures they meet specifications.
ST1203	Define detailed plant & production lines layout / plant configuration / deliver plant digital twin to finalize layout including detailed configuration and virtual model of the factory.
ST1204	Approves factory acceptance test to start production line (physically) to officially go-ahead to start the physical Factory Acceptance Test (FAT).
ST1205	Align on manufacturing, logistics & interface requirements with OEM to coordinate between OEM and partner and ensure process alignment.
ST1206	Define target objectives & scope for manufacturing engineering activities (partner view) and define its own engineering scope within OEM boundaries.
ST1207	Organize partner product subsystem & production subsystem MBOM, review & release for structuring its MBOM for its subsystems and releases it.
ST1208	Define partner product subsystem process flow (incl. production & MBOM) to make clear manufacturing flow for its subsystem.
ST1209	Define product & production subsystem manufacturing operations & steps (partner view) for detailed operations for the subsystem.
ST1210	Allocate parts & items to steps (partner view) to map subsystem parts to partner manufacturing steps.
ST1211	Define preliminary plant & production lines layout (partner) to first draft the layout of partner's portion of manufacturing.
ST1212	Define, design manufacturing resources for OEM integration (machines, jigs, tools for mounting, transport…) (partner view) to design resources required to integrate into OEM process.
ST1213	Define, design manufacturing resources for partner (machines, jigs, tools…) (partner view) to design its internal manufacturing equipment.
ST1214	Create necessary documents for evaluation of implementation by OEM to provide documentation needed for OEM validation.
ST1215	Define standard and times; finalize routings (partner) for timing and routing.
ST1216	Formalize work instructions (partner) to create detailed assembly instructions.

ID	Explanation / Comment
ST1217	Formulate mounting instruction for OEM to integrate the product subsystem while using the production subsystem to describe how the OEM integrates the partner subsystem into the final assembly.
ST1218	Simulate & optimize production flows & operations (partner) to run its own process simulation and optimization.
ST1219	Execute & optimize virtual commissioning and freeze (partner) for commissioning of partner systems.
ST1120	Approve subsystem manufacturing design.
ST1121	Release of partner subsystem process.
ST1122	Prepare conformity report and share with OEM to send conformity validation documents.
ST1123	Sends factory acceptance test to start production line (physically) for dispatching officially FAT to OEM to begin physical production-line validation.

6 Cross-Domain Configuration & Variant Management

6.1 Role of Configuration and Variant Management in Digital Twins

Configuration Management (CM) and Variant Management (VM) are fundamental enablers for the effective implementation of Digital Twins, supporting the justification for integrating Cross-Domain Configuration & Variant Management within the Collaborative Digital Twin (CDT) framework. Together, CM and VM guarantee the integrity, traceability, and regulatory compliance of the digital representation of products throughout their lifecycle.

Alignment and Consistency

WP6 facilitates the alignment of variability between original equipment manufacturers (OEMs) and suppliers, harmonizing diverse product configurations to enable seamless collaboration. This process ensures that all CDT stakeholders operate consistently with the latest released product data, thereby establishing a unified single source of truth.

Traceability and Change Control

Comprehensive traceability is maintained by transparently linking changes and updating histories across multiple systems and partners. Strict change control procedures involve stakeholders in impact analyses and change boards, preventing unforeseen disruptions and ensuring controlled evolution of product configurations.

Lifecycle Support and Compliance

CM and VM extend beyond design to support subsequent product lifecycle phases, such as maintenance and servicing, ensuring continuity beyond the CDT project scope. Additionally, they serve as enablers of compliance with regulatory and quality standards, which is essential for operation within regulated industrial environments.

6.2 Conflicting Requirements as the Foundation for Variant Management

Conflicting requirements represent the fundamental rationale for implementing Variant Management (VM) within product development processes. Diverse client specifications necessitate differentiated product offerings, where certain requirements coexist harmoniously while others are inherently contradictory, thereby necessitating the creation of product variants and structured variability strategies.

Trade-Off Analysis and Architectural Implications

Following the identification of conflicting requirements, a systematic trade-off analysis evaluates the costs of maintaining variability against potential market opportunities. This analysis directly influences product structure decisions, promoting the adoption of modular product architectures that accommodate variant complexity while maintaining manufacturability and cost efficiency.

Strategic Variant Planning and Justification

Each product variant is explicitly justified by traceable (contradicting) requirements, ensuring that variability is purposefully driven rather than arbitrarily introduced. Early integration of variant strategy during initial requirement phases prevents costly late-stage modifications and reduces development cycle times.

Optimization and Cost Control

A disciplined VM approach actively prevents the proliferation of unnecessary variants and options, thereby minimizing associated development, production, and maintenance efforts. This optimization balances market responsiveness with operational efficiency, ensuring sustainable product portfolio management.

6.3 Product Line Engineering and Configuration Management

Product Line Engineering (PLE) provides the methodological framework for systematically constructing products based on shared and variable components, emphasizing platform thinking and product family approaches. Configuration Management and Variant Management (CM & VM) function as operative tools within this framework, defining legitimate component combinations and ensuring that each variant is comprehensively described, internally consistent, and fully traceable. The Collaborative Digital Twin (CDT) represents the digital implementation of these PLE concepts, enabling effective management of product complexity across distributed organizations.

Configuration Hierarchy and Modular Architecture

The configuration hierarchy and structure (eBoM), distinct from the manufacturing Bill of Materials (mBoM), provides a systematic organization of product variability. Modularization serves as a key mechanism for enabling component reuse, reducing redundancy, and facilitating efficient variant management across the product portfolio.

Variant Management in Inter-Organizational Collaboration

The role of VM in partner collaboration depends fundamentally on the nature of shared product information. When discrete, pre-defined variants are exchanged between partners, VM plays a limited collaborative role. However, when shared variability is managed across organizational boundaries, critical aspects require explicit discussion and agreement. These include the scope of variability—restricting shared variability to system elements directly relevant to the collaborative effort—and intellectual property (IP) considerations, as variability structures may inadvertently expose proprietary product information not intended for exchange. Careful delineation of these boundaries ensures both effective collaboration and appropriate IP protection.

6.4 Semantic Harmonization for Collaborative Engineering: The Role of Unified Terminology in the Collaborative Digital Twin

6.4.1 Rationale

The effective collaboration of original equipment manufacturers (OEMs), suppliers, and engineering tool vendors across distributed value chains requires more than technological interoperability, it demands semantic alignment and shared conceptual frameworks.

We recognize that engineering collaboration at scale depends fundamentally on establishing common terminology and unified semantics across all participating organizations. Without standardized language, terminology ambiguities, tool-specific nomenclature, and domain-specific variations create barriers to seamless information exchange, impede decision-making processes, and introduce costly rework and misinterpretation.

This challenge is particularly acute in the domains of configuration management, fundamental for collaboration, and variant management, where multiple industrial sectors employ divergent approaches to product configuration, feature modeling, and lifecycle management. While ISO/IEC standards (10007, 15288, 26550, 26580) and IEEE 828 provide normative frameworks, their implementation varies significantly across organizations and tool ecosystems. A unified glossary addressing configuration management fundamentals, variant management core concepts, and business logic rules serve as the essential semantic bridge that brings OEMs, suppliers, and tool vendors into alignment. By establishing a consolidated terminology derived from international standards and supplemented with industry-specific considerations, the CDT project facilitates not merely technical interoperability but genuine collaborative engineering, where all parties operate with shared understanding, consistent definitions, and transparent cross-organizational communication.

This semantic harmonization extends beyond lexical standardization to enable the effective implementation of the digital twin concept itself. Configuration management provides the temporal dimension, tracking changes, versions, baselines, and traceability, while variant management adds the structural dimension, managing features, variations, and product families. Effectivity mechanisms bridge both dimensions, determining when variants apply across temporal and structural domains. When OEMs, suppliers, and tool vendors employ disparate terminology for these concepts, the digital twin becomes fragmented, with each party maintaining incompatible data models and configuration structures. Conversely, unified terminology enables the construction of integrated digital representations that reflect the complete product configuration across all lifecycle phases and across the entire collaborative ecosystem. Thus, the comprehensive variant management glossary represents not merely a reference document but a foundational enabler of true collaborative digital engineering.

6.4.2 Consolidation Efforts and Terminology Standardization

While Chapter 0

Terminology & Ontologies in the Context of Collaborative Digital Twins discusses the project-specific terminology used within this recommendation, the comprehensive glossary consolidation within this chapter represents a substantial effort to harmonize configuration management and variant management terminology across diverse organizational, tool, and industry contexts.

Key Consolidation Achievements

Several critical terms were standardized through systematic consolidation efforts. Change Impact Analysis was renamed from the broader "Impact Analysis" to provide more comprehensive terminology aligned with ISO 10007 and IEEE 828 standards. Configuration Item consolidated multiple related designations like "Configurable Unit" and "Configurable Assembly", into a single standardized term reflecting ISO standards. Feature Model represents a significant consolidation, merging "Variation Model," "Variability Model," and "Feature Catalog" under ISO/IEC 26580 preferred terminology, while retaining "Feature Catalog" as an enumeration-focused variant. Product Line was renamed from "Product Family", aligning with ISO/IEC 26580 standard designation. Product Variant consolidated the ISO/IEC 26580 term "Member Product" into a single, unified concept. Variant Configuration brought together "Bill of Features" and related terms under a single semantic umbrella. Additionally, “Maturity” merged "Maturity Status" and "Coverage Status" through identification of identical definitions and scope, and “Parameter” underwent normalization of singular/plural forms for consistency.

Critical Term Restorations

The consolidation identified several terms previously marked as removed but requiring restoration due to their critical importance. Most significantly, “Effectivity” was marked for mandatory restoration, recognized as a fundamental bridge concept between Configuration Management (managing changes over time) and Variant Management (managing variations across product families). Effectivity encompasses multiple subtypes, “Date Effectivity”, “Serial Number Effectivity”, “Lot Effectivity”, and “Parameter Effectivity”, each determining when specific configurations, features, or part versions apply in production or deployed products. This concept, derived from ISO 10007 and widely implemented in PLM tools, was essential to restore given its role in determining temporal and conditional applicability of variant configurations.

New Terms and Extensions

The consolidation process identified three significant conceptual gaps requiring new term definitions. “Feature Lifecycle State” was added to articulate the technical maturity, market readiness, and intellectual property protection phases of features across the product lifecycle. This term encompasses progression through phases from Concept through Ideation, IP Assessment, In Development, Enabled/Available, Active deployment, Planned Deprecation, Deprecated, and ultimately Obsolete/End-of-Life, with explicit integration of IP protection phases including Patent lifecycle (Provisional, Pending, Issued, Maintenance, Expired), Trademark status, and Trade Secret management. “Parametric Variation” was introduced as an extension to the Feature Model, accommodating continuous or discrete numeric parameter variations such as engine displacement, battery capacity, and tolerance ranges. While ISO/IEC 26580 concentrates on categorical feature variations, parametric variation addresses quantitative distinctions essential in many industrial domains. “Configuration Status Accounting” and “Configuration Audit” were identified as missing Configuration Management prerequisites, completing the ISO 10007 and IEEE 828 compliance framework.

Standards Alignment and Consolidation Rationale

The consolidation systematically aligned terminology with international standards. ISO/IEC 26550 and 26580 formed the normative foundation for variant management core concepts, ensuring that terminology such as Feature, Feature Model, Product Line, Product Variant, and Variant Configuration maintains semantic consistency with authoritative standards. The three-level hierarchical structure, Configuration Management Fundamentals, Variant Management Core, and Business Logic, emerged as the consolidation framework, positioning Configuration Management concepts as foundational prerequisites and organizing Business Logic rules governing their integration within Cross-Domain Configuration Management (CDCM) contexts.

Tool-Specific Terminology Mapping

The consolidation documented systematic mapping of CDT glossary terms to tool-specific terminology, recognizing that practitioners operate within diverse software ecosystems. SAP implementations employ "Characteristics" for Features and "Master Data Characteristics" for Feature Models; PTC Windchill uses "Option" and "Option Pool" respectively; PureVariants aligns directly with glossary terminology; and Teamcenter implements its own proprietary variants. This mapping ensures that the glossary provides coherence without imposing rigid terminology that conflicts with established tool vocabularies, thereby enabling practitioners to maintain communication across tools while adopting standardized conceptual frameworks.

6.5 Configuration & Variant Management User Journeys (UJ) & Connecting Points to “Main” CDT User Journeys

The development of the Collaborative Digital Twin (CDT) requires a structured understanding of how configuration and variant management activities unfold within multi-partner engineering ecosystems. To support this objective, Work Package 6 (WP6) defined a sequence of five dedicated User Journeys (WP6 UJ1–UJ5) that capture practical workflows, roles, and information exchanges required to establish and maintain product variability in a collaborative context. Each journey represents a coherent step in the evolution from conceptual feature definition toward rule-based configuration governance.

These User Journeys document the outcomes of a series of workshops held between 2024 and 2025 with representatives from multiple industrial partners. Together, they illustrate how configuration and variant management (CVM) processes interact with broader CDT processes; particularly requirement management, model integration, and digital twin instantiation.

6.5.1 WP6 UJ1: Define Feature Model with Potential Re-Use of Existing Features (Models)

The following subsections describe each of the five WP6 User Journeys in detail. They follow a consistent structure, outlining purpose, process steps, involved roles, and connecting points to the overarching CDT User Journeys within the Collaborative Digital Twin.

The first WP6 user journey (Figure 34: User Journey “Define Feature Model with Potential Re-Use of Existing Features (Models)”) establishes the foundation of variant management by defining a structured feature model that captures both new and reusable functional or architectural elements. The goal is to represent product capabilities at an abstract level, decoupled from implementation details, thereby enabling systematic analysis of potential reuse across product families.

Within this step, participating partners identify relevant feature sources, such as legacy systems, prior configurations, or requirement catalogs. Harmonization of naming conventions and alignment with the consolidated CDT terminology ensures consistency with the standardized Feature and Feature Model definitions derived from ISO/IEC 26580. The resulting model serves as the initial reference for downstream variability identification and configuration rule development.

Critical outcomes of UJ1 include the establishment of traceability between features and underlying requirements, the creation of a controlled repository for reusable features, and the introduction of version control mechanisms enabling collaborative feature evolution across organizational boundaries.

In WP6 UJ1, Product Management begins by defining the scope through consolidation of requirements, ensuring that business intent and product constraints are captured. The Feature Model Engineer then checks for suitable features in the existing feature catalog, creates the new feature model, and identifies scope-specific deltas and missing features, requesting consistency support as needed. After completing the definition, the engineer may update the feature catalog and finally release the validated feature model for downstream use. The Design Engineer contributes by checking technical consistency and supporting the modeling process where required. The process concludes as Product Management approves the feature model, establishing a controlled, traceable artefact that serves as the entry point for all subsequent configuration and variant management activities within the CDT.

This user journey interfaces primarily with CDT UJ1 (Collaborative Requirements Content Management) by consuming consolidated requirement information as input for feature definition and ensuring traceability back to stakeholder needs. It also supports CDT UJ2 (Collaborative System Design) by providing the structured variability framework that system architects use to allocate functions and define configurable system architectures within the Collaborative Digital Twin.

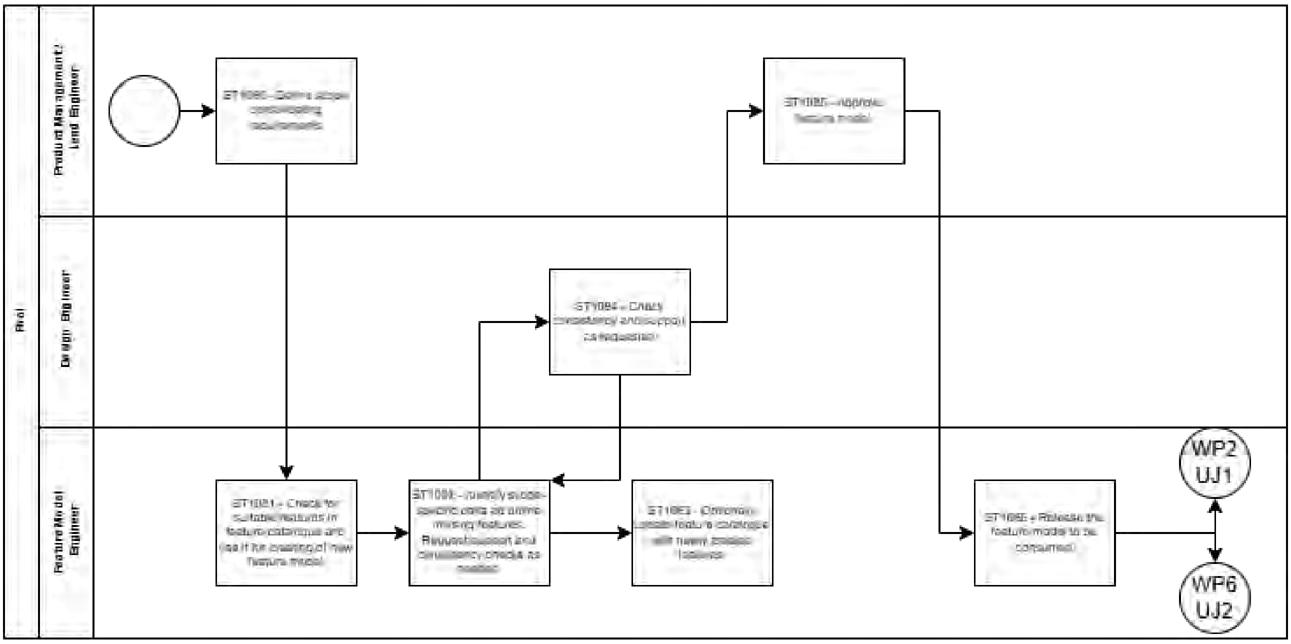


Figure 34: User Journey “Define Feature Model with Potential Re-Use of Existing Features (Models)”

6.5.2 WP6 UJ2: Define a new Product Line with Potential Re-Use of existing Features / Functions / Requirements and Allocated Specifications

Building upon the feature model, the second user journey extends the focus from individual features to an integrated product line perspective (Figure 35: User Journey "Define a new Product Line with Potential Re-Use of Existing Features / Functions / Requirements and Allocated Specifications”.). The objective is to define a coherent structure of products that leverages existing functions and specifications where feasible, while introducing new capabilities driven by market and stakeholder needs. The result is a 150% product structure that represents the complete design space, covering all potential variants before filtering for specific market realizations.

Within this step, stakeholder consolidate requirement inputs and align on the intended scope of the product line. This process combines business-driven definition with engineering feasibility, ensuring that variability is introduced deliberately. The resulting product line definition enables early evaluation of design trade-offs, reuse potential, and platform strategies across OEMs and suppliers.

Critical outcomes of WP6 UJ2 include the establishment of a structured product line baseline, the linkage of product structures with configuration data, and the preparation of a product configurator environment that governs variant creation and release. This ensures that product line definition, configuration logic, and variant realization remain synchronized throughout the digital twin lifecycle.

In WP6 UJ2, the Product Lead initiates the process with a Product Definition Letter, setting the overall scope and intent. The Product Manager consolidates stakeholder and market requirements to define the boundaries of the product line and later creates product variants within the configurator. The Product Structure Manager transforms this scope into a 150% Bill of Materials, defining the needed variability for each product, while the Configuration Manager sets up and releases the Product Configurator, linking it to the product structure. This hand-off from business intent to executable configuration ensures that product line definition, structure, and rules evolve coherently under clear ownership.

This user journey links directly to CDT UJ2 (Collaborative System Design), providing the product line context, structural variability, and configurator logic required to instantiate specific system architectures. It also connects to CDT UJ4 (Collaborative Integration and Tests) by enabling configuration-controlled validation of product variants, ensuring that the defined product line can be efficiently verified, tested, and manufactured within the CDT framework.

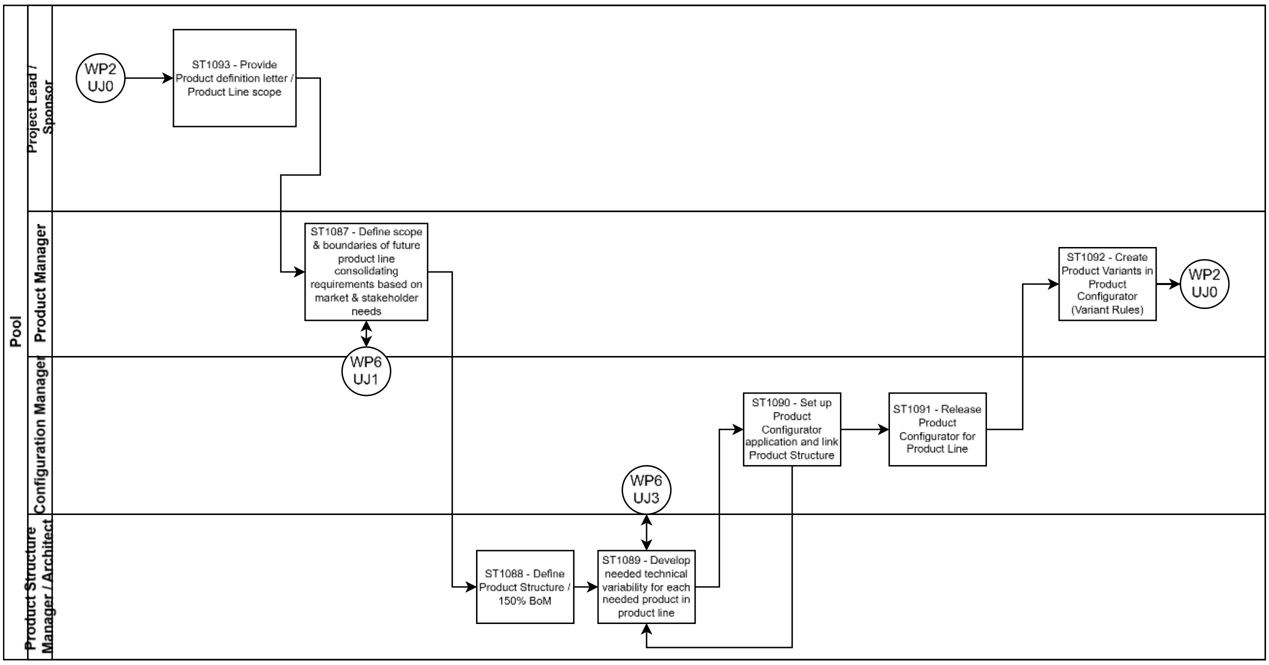


Figure 35: User Journey "Define a new Product Line with Potential Re-Use of Existing Features / Functions / Requirements and Allocated Specifications”.

6.5.3 WP6 UJ3: Identify Variation Points Based on Contradicting Requirements

The third user journey (Figure 36: User Journey "Identify Variation Points Based on Contradicting Requirements") addresses the analytical phase of variability identification. By systematically examining contradicting or mutually exclusive requirements across customer segments, markets, or regulatory domains, the process derives explicit variation points within the product architecture.

Contradicting requirements act as indicators of legitimate variability. Structured analysis methods, such as requirement clustering, constraint modeling, or functional dependency mapping, are used to expose where products must diverge to satisfy differing needs. The identification of variation points ensures that variability remains purposeful, controlled, and directly linked to documented rationale, thereby avoiding uncontrolled complexity growth.

Outcomes of WP6 UJ3 include the creation of traceable links between conflicting requirements and the corresponding variation points in the feature or system model. This provides the logical bridge between requirement engineering and configuration management, ensuring that all subsequent variant definitions can be justified by measurable business or technical drivers.

In WP6 UJ3, the Requirements Engineer initiates the process by maintaining and updating requirements and checking the requirements database for conflicts, resulting in a structured findings report. The Design Engineer then identifies potential variation point candidates, jointly with the Feature Model Engineer assessing technical impacts and required efforts. Based on this analysis, the Product Manager decides whether to deactivate conflicting requirements or create new variation points. Once this decision is made, the Requirements Engineer removes or adjusts the affected requirements in the documentation or implements the variation point and corresponding feature values into the requirements environment. Finally, the Feature Model Engineer integrates the new variation point into the system design, closing the loop between requirement analysis and technical realization.

This user journey aligns closely with CDT UJ1 (Collaborative Requirements Content Management), as it refines and resolves requirement conflicts that drive product variability. It also connects to CDT UJ2 (Collaborative System Design) by ensuring that each established variation point has a validated architectural representation, forming the foundation for consistent variant definition and downstream configuration control.

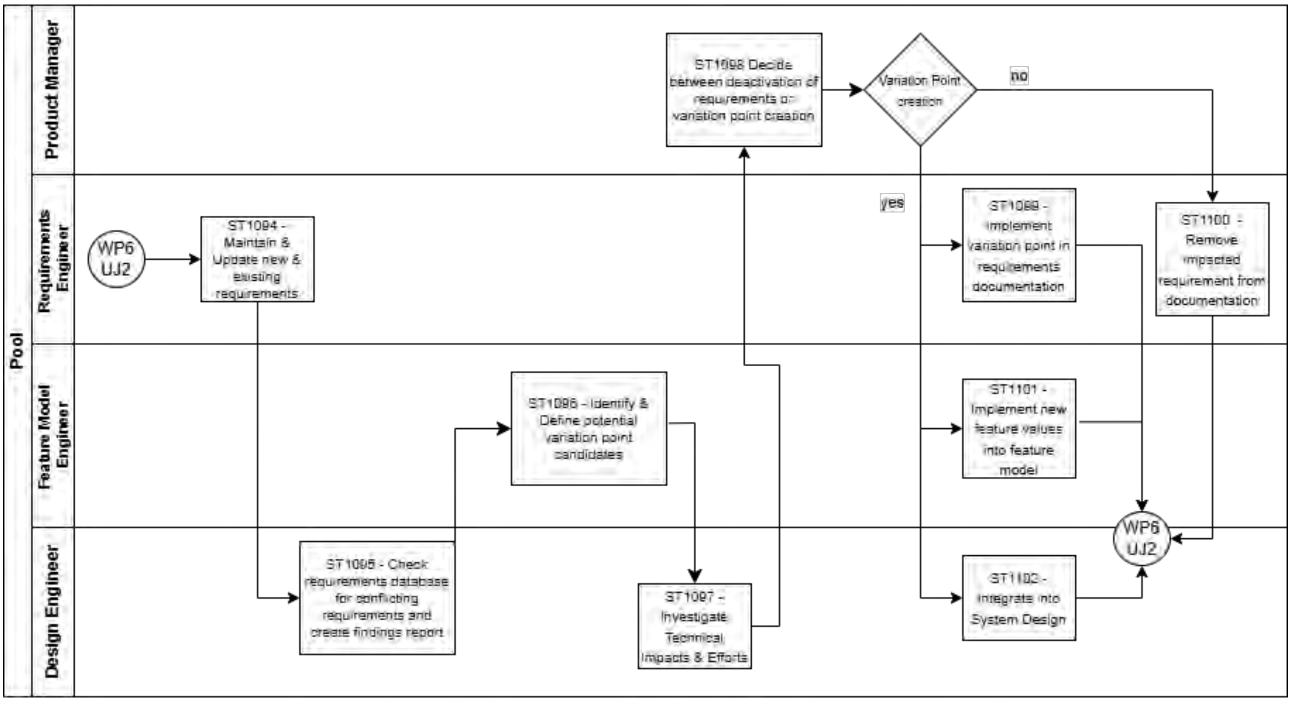


Figure 36: User Journey "Identify Variation Points Based on Contradicting Requirements"

6.5.4 WP6 UJ4: Identify / Designate Variants Allocated to Variation Points

The fourth user journey (Figure 37: User Journey "Identify / Designate Variants Allocated to Variation Points") translates the abstract variation points into tangible variant definitions. For each identified variation point, the set of valid variant realizations is defined, documented, and associated with relevant configuration data.

This step operationalizes the transition from conceptual variability to implementable product structures. Variants may represent alternative functional implementations, different physical realizations, or parameterized configurations. The process ensures that each variant is uniquely identifiable, versioned, and embedded within the overall configuration hierarchy.

In collaborative engineering contexts, particular attention is given to ownership and governance of shared variants. Partners agree on which variants are globally managed across the CDT environment and which remain locally controlled within specific organizational scopes. Establishing this boundary prevents information leakage while maintaining interoperability.

The outcome of WP6 UJ4 is a coherent mapping of variation points to concrete variant definitions, forming the prerequisite for consistent configuration rules and automated variant selection mechanisms.

In WP6 UJ4, the Configuration Manager begins by reviewing the identified variation points and prioritizing them based on impact and feasibility. The Product Manager then checks for existing variant definitions associated with each variation point, after which the Feature Model Engineer either maps available variants or initiates the creation of new ones. The Requirements Engineer defines variant-specific requirements, and the Design Engineer develops design solutions which are integrated by the Product Structure Manager for each variant into the product structure. The Feature Model Engineer subsequently updates the feature model with new variants, after which the Configuration Manager defines applicability and configuration rules. The Configuration Manager concludes the process by validating the variants across requirements, design, and structure, documenting variant management at the variation point, and publishing the updated models to inform all stakeholders. Clear ownership of variant definition and approval is essential, with the Configuration Manager acting as process integrator across all contributing roles.

This user journey connects directly to CDT UJ2 (Collaborative System Design) and CDT UJ4 (Collaborative Integration and Tests). It provides the technical and configuration basis for verifying variant-specific designs and test configurations within the CDT, ensuring that each variant is correctly represented, validated, and traceable across all lifecycle phases.

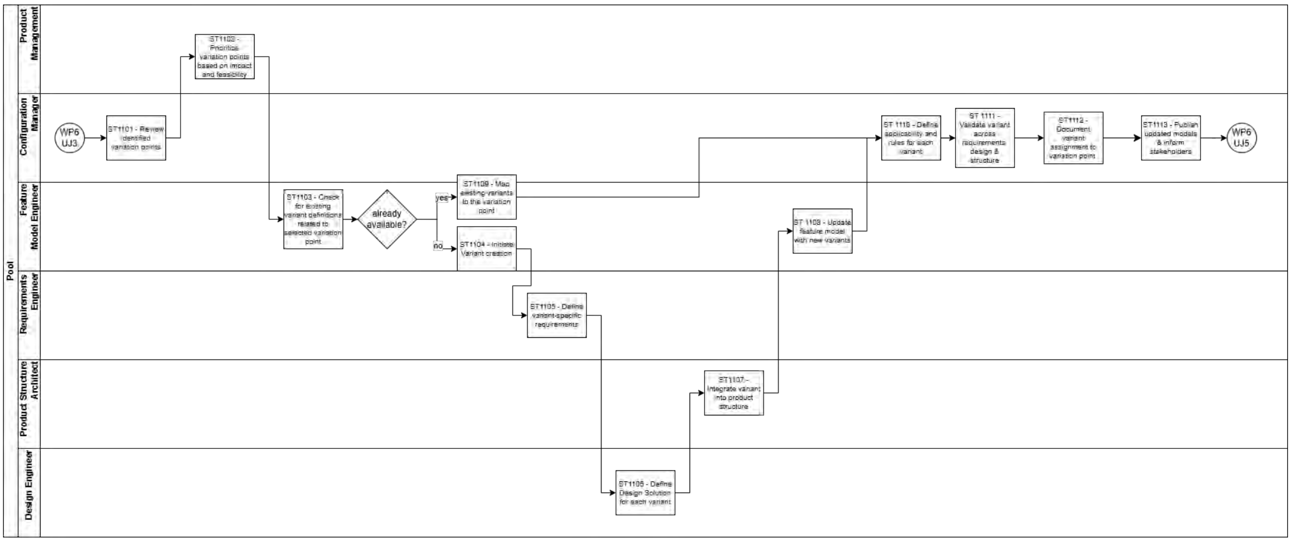


Figure 37: User Journey "Identify / Designate Variants Allocated to Variation Points"

6.5.5 WP6 UJ5: Define and Update Configuration Rules and Variability Model (Variation Dependency Identification)

The fifth user journey (Figure 38: User Journey "Define and Update Configuration Rules and Variability Model") consolidates prior results into an executable variability model governed by configuration rules. These rules define valid feature combinations, dependency constraints, and incompatibility conditions that collectively determine valid configurations within the product line.

Rule definition involves cross-domain collaboration among systems engineers, domain experts, and tool specialists. The process captures logical relations such as “requires,” “excludes,” or “implies,” expressed in machine-interpretable form suitable for implementation in configuration tools. The variability model is maintained under configuration control, ensuring that rule updates follow formal change processes analogous to those applied to product data.

An essential element of this step is variation dependency identification. Through analysis of real-world use cases and engineering data, hidden dependencies between features and system parameters are uncovered and formalized as explicit rules. The resulting rule set enhances configuration validity checking, supports automated product configuration, and provides traceable documentation of variability rationale. WP6 UJ5 thus represents the culmination of the CVM user journey sequence, establishing a governed, rule-based framework for managing product variability throughout the digital twin lifecycle.

In WP6 UJ5, the Feature Model Engineer initiates the process by identifying logical and structural dependencies between variation points, while Product Management defines relevant business constraints and the Design Engineer determines technical constraints between variants. The Requirements Engineer contributes by formulating requirement-level dependency rules, which the Configuration Manager then consolidates into a consistent rule set. The Feature Model Engineer models the resulting configuration logic within the variability model, and the Product Structure Manager embeds this logic in the product structure. The Configuration Manager subsequently assesses and tests the consistency and completeness of all rules together with relevant stakeholders. If the review identifies inconsistencies, the process loops back to the Feature Model Engineer for updates; if approved, the engineer releases the updated variability model, establishing the authoritative configuration rule baseline.

This user journey directly supports CDT UJ2 (Collaborative System Design) and CDT UJ5 (Collaborative Manufacturing Engineering). It ensures that configuration constraints defined during design are consistently propagated into manufacturing and lifecycle processes, enabling rule-based configuration validation, variant realization, and controlled product updates within the Collaborative Digital Twin environment.

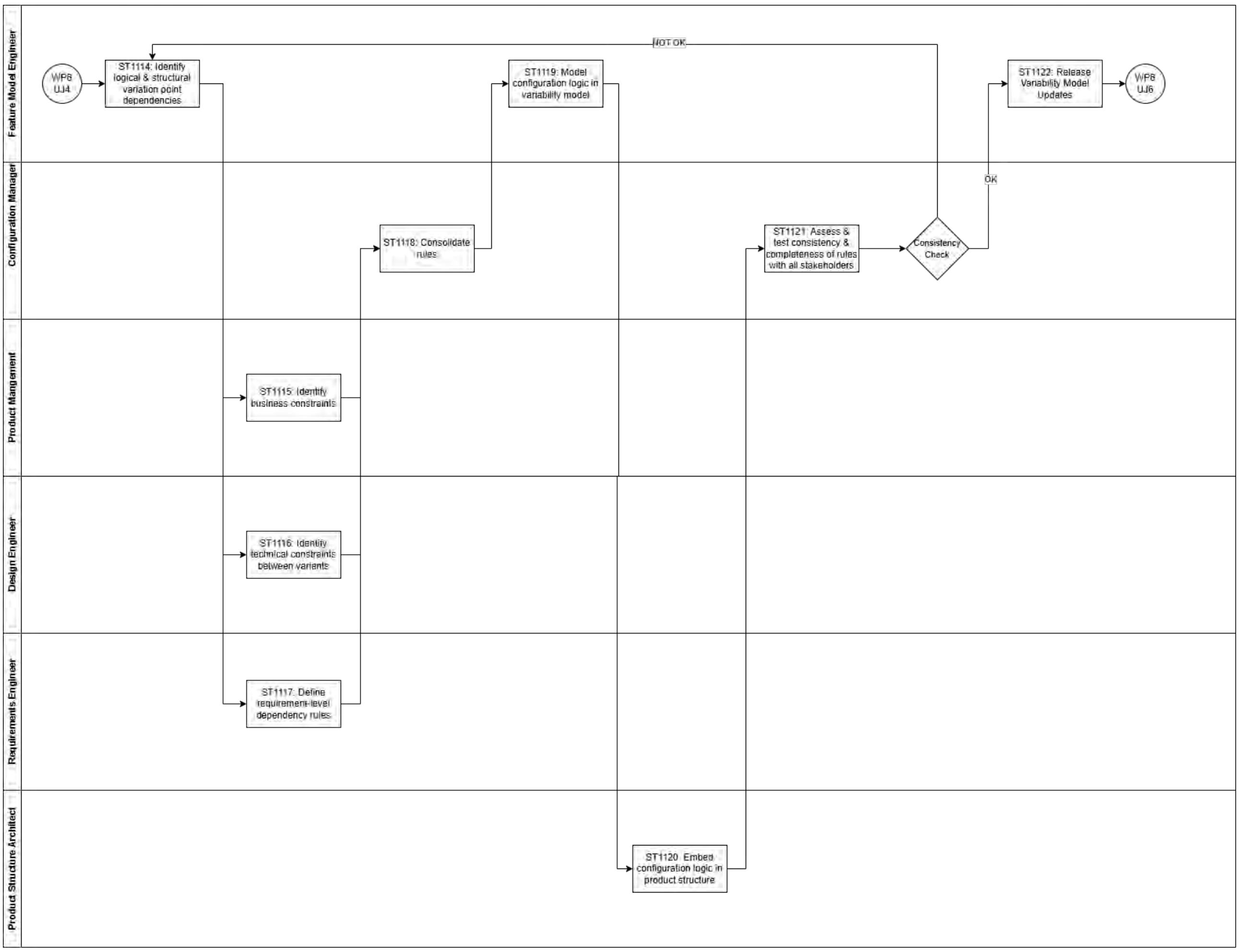


Figure 38: User Journey "Define and Update Configuration Rules and Variability Model"

Collectively, the five WP6 User Journeys establish a closed, iterative process: From feature definition to rule-based configuration governance. They serve as the conceptual and procedural foundation for the Collaborative Configuration and Variant Management Demonstrator described in the following chapter.

6.5.6 Further User Journey Candidates

Within the CDT Project, two further User Journeys have been identified but their development was not covered due to a thematic focus on the Design Engineering aspects. Therefore, the CDT Work can be continued in a following project and investigate:

- “Configure & Release Product Variants for Manufacturing”
- “Manage the lifecycle of released Variants across the complete product lifecycle”

6.6 Collaborative Configuration & Variant Management Demonstrator

The practical application of the WP6 configuration and variant management (CVM) approach is demonstrated using the Mars Rover dataset, a widely established reference within prostep ivip (see Chapter 6.1). The scenario illustrates how requirements, features, configuration rules, and the product structure can be systematically linked to ensure consistent management of product variability across all engineering domains. Through this example, the demonstrator operationalizes the concepts introduced in WP6 UJ1–UJ5 and shows how configuration-controlled product evolution can be realized in a collaborative engineering environment.

In complex product development settings, managing variability efficiently is essential for maintaining coherence between capability definitions, feature-level abstractions, structural design elements, and 150% BoM representations. Within the Mars Rover scenario, requirements define product capabilities such as mission environment or payload capacity. These capabilities are translated into features and governed by configuration rules—the logical mechanisms that determine permissible feature combinations. These, in turn, shape the configurable Bill of Materials (BoM) and define the valid set of product variants. The integration of these artifacts ensures traceability, reduces inconsistencies, and supports the transition from Engineer-to-Order (EtO) practices toward a more scalable Configure-to-Order (CtO) paradigm.

6.6.1 Mars Rover Data Set Extension

The demonstrator material represents consolidated results of the prostep ivip CDT WP6 working group. It shows how new or modified requirements are incorporated into the product line through coordinated updates to the feature catalog, feature model, variation points, variant definitions, and configuration rules. This integrated approach links engineering processes with downstream functions such as sales and production planning, forming a continuous digital thread within the Collaborative Digital Twin.

Figure 39 illustrates the CDT variant management concept, connecting the abstract variability space in the feature catalog and feature model with the concrete implementation level represented by the product catalog and configurable BoM catalog. Valid feature selections result in instantiated Bills of Features (BoF) and 100% Bills of Materials (BoM), providing the product instance required for manufacturing, costing, and customer quotation.

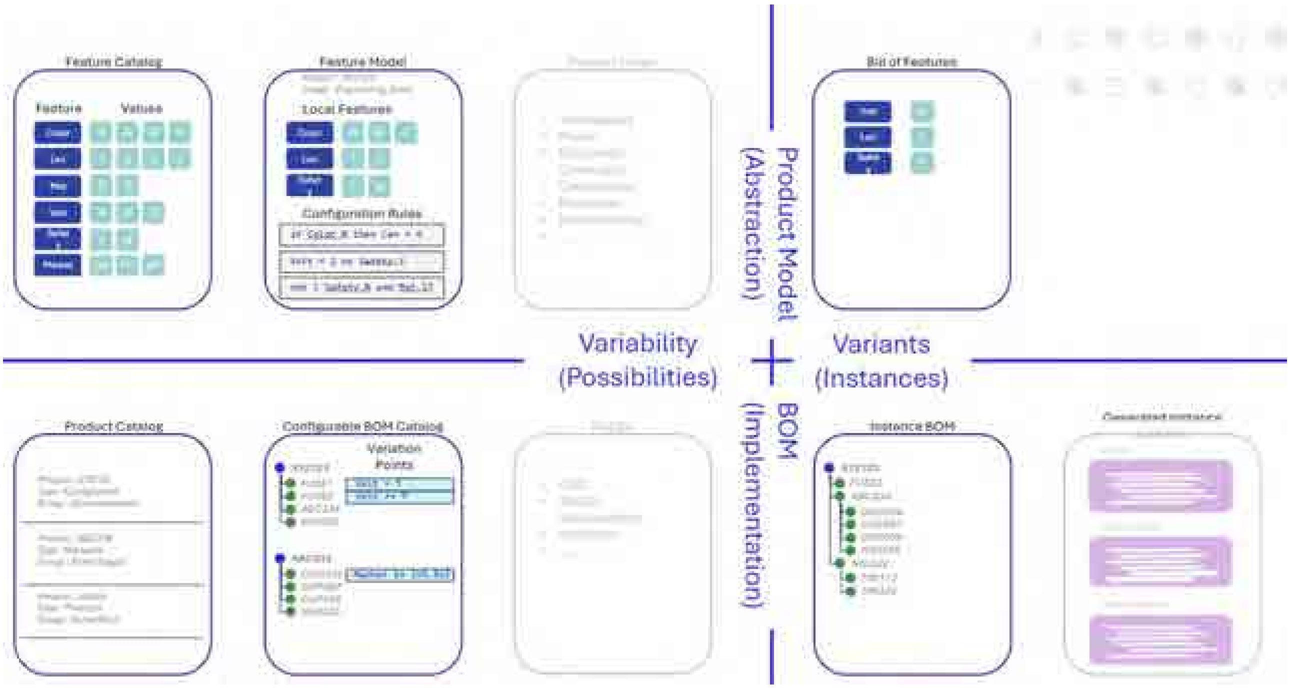


Figure 39: CDT Variant Management Concept

Linking WP6 User Journeys to the Demonstrator Workflow

The Mars Rover storyline reflects a complete application of the WP6 user journeys:

- WP6 UJ3 (Identify Variation Points Based on Contradicting Requirements)

The process begins when new customer requirements—such as mission environment or payload capability—introduce contradictions with existing specifications. Requirements Engineers and Design Engineers perform feasibility checks, identify variation points, and evaluate whether the requirement leads to feature extension or deactivation.

- WP6 UJ1 (Define Feature Model with Re-Use of Existing Features)

Once a variation point is confirmed, the Feature Model Engineer updates the feature catalog and feature model, adding new features or feature values where required (e.g., new environment conditions or payload categories). Logical dependencies, such as exclusions based on environmental constraints, are introduced or refined.

- WP6 UJ4 (Identify / Designate Variants Allocated to Variation Points)

New or adapted variants are defined at the variation point. Requirements Managers describe variant-specific conditions, Design Engineers develop associated technical realizations, and the Product Structure Manager integrates the new solutions into the 150% BoM. This ensures that logical feature variability is consistently reflected in the physical product structure.

- WP6 UJ5 (Define and Update Configuration Rules and Variability Model)

Configuration Managers consolidate business, technical, and requirement-level constraints into formalized configuration rules. These rules govern the applicability of features and maintain consistency within the variability model.

- WP6 UJ2 (Define a New Product Line with Re-Use of Features / Functions / Specifications)

During instantiation, valid feature combinations form Bills of Features, which can be transformed into 100% BoMs supporting product-specific documentation, costing, and manufacturing preparation. This step completes the EtO–CtO bridge by translating feature selections into executable product configurations.

Figure 40 visualizes the storyline of this WP 6 demonstrator, emphasizing the controlled propagation of variability across requirements, features, rules, and product structures as new feature requests arise.

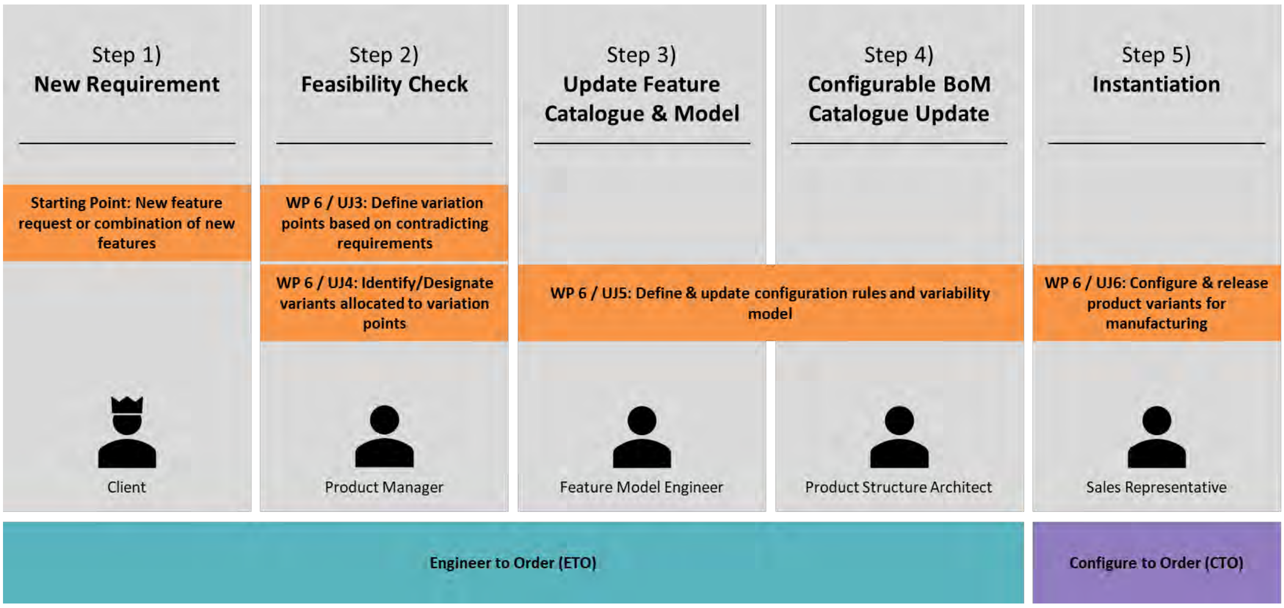


Figure 40: Storyline of WP6 Demonstrator

The following is a dive into the data following the fictional storyline

Step 1) A client raising a new feature/requirement

Client: "Is it possible to use my communication unit with 6 kg on the mars?"

The external request raising from the client is about to use a communication unit for usage in a product on mars mission. Background is the product development team want to reuse the specification from the latest moon rover project as a base to develop a mars rover in an efficient way. In this case, the dedicated communication unit is an advanced version, which has with 6 kg more mass than the moon version. Besides that, the gravity on moon is less than on mars. This implicated a requirement for a higher payload and impact the whole construction of the mars rover compared to the moon version. The new requirement needs to be analyzed in a new context, include different conditions of the system of interest – Here in this example regarding the chassis and the steering motor. The first activity is to check, if the needs are compatible with the current requirements. In this case the requirements list needs a new requirement for the higher payload of 6 kg.

Id	Name	Text
2.3.3	Motion over slopes	The Mars Rover shall be able to move on slopes which fullfill the following condition: * slope angle uphill max. 20 degree * slope angle downhill max. 30 degree * slope angle sidehill max 15 degree
2.6	Max Mass	The Mars Rover shall have a total mass of not more than 12 kg
2.6.1	Mars Rover Mass	The Mars Rover shall have a total mass of not more than 10 kg
2.6.2	Payload	The Mars Rover shall have a payload of 2 kg.
2.9	Operating on Mars	The Mars Rover shall be able to operate on the planet Mars.
2.9.1	Operating Atmosphere	The Mars Rover shall be able to operate in Mars atmosphere for at least 25 y
2.9.2	Mars Gravity	The Mars Rover shall be able to operate under Mars gravity of 3.71 m/s^2.

Figure 41: Exemplary Compatibility Checks on Requirements Level

Step 2) Feasibility Check

Product Manager: "After detailed (impact) analysis we decided to cater for this wish, invest in the development and to include it in our Product Line."

This decision for a new requirement is implemented by the Requirements Engineer by creating an additional one as seen in Figure 42.

Id	Name	Text
2.6	Max Mass	The Mars Rover shall have a total mass of not more than 12 kg
2.6.1	Mars Rover Mass	The Mars Rover shall have a total mass of not more than 10 kg
2.6.2	Payload	The Mars Rover shall have a payload of 2 kg.
2.6.3	Payload	The Mars Rover shall have a payload of 6 kg.
2.8	Landing	The Mars Rover shall be able to land on planet Mars.

Figure 42: Extension of Requirements by Adding an Additional Requirement

Step 3) Update Feature Catalog & Model

Feature Model Engineer:

> "We have two contradicting requirements.
>
> This needs to be reflected in our company's feature catalogue and Rover Product Line feature model so I can link the new requirement with features."

The decision also impacts on the feature catalog and feature model (Figure 43). The catalog and model for the product line need to be refined in terms of features, values and configuration rules. The features Environment and Payload (turquois) with mars and 6 kg (dark orange) are added. The assessment and definition of configuration rules, e.g. the exclusion of the feature Mobility "Fly" when selecting the Environment "Moon" is an important step to inherit new variants in the future.

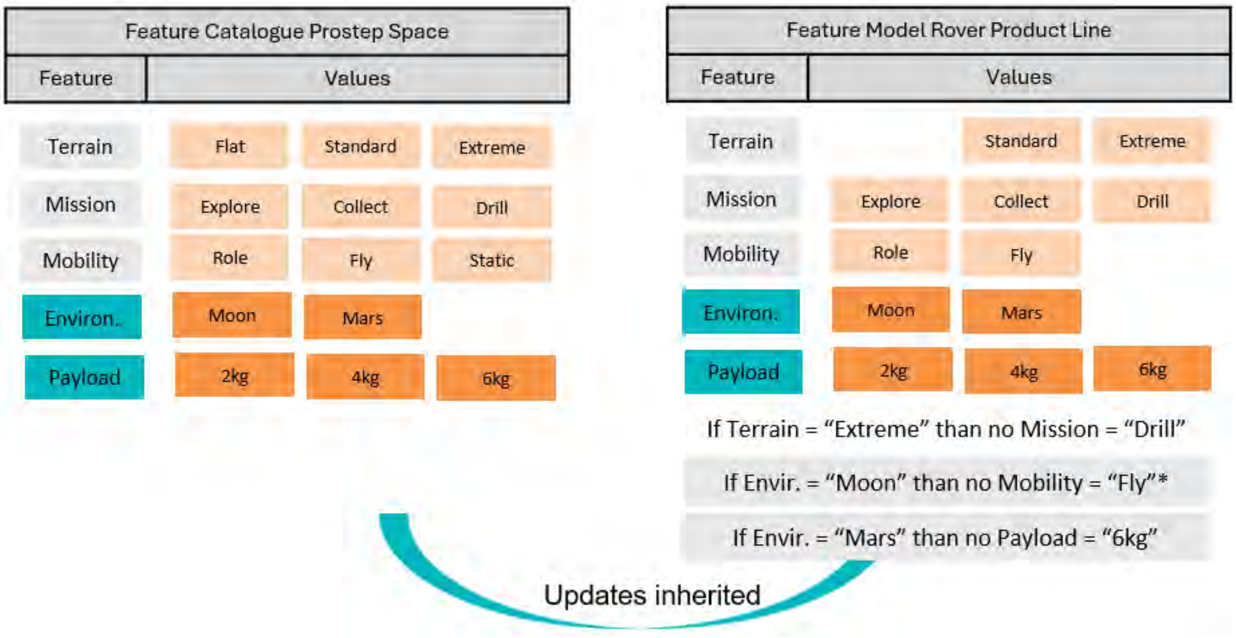


Figure 43: Extension of Feature Catalog & Feature Model

Step 4) Configurable BoM Catalog Update

Product Structure Architect:

"The Feature Model Engineer informed me about the updates of the feature model. My duty is to create a placeholder in our Configurable BoM, link them with the Feature Model, and trigger the Development of Design Solutions"

Since higher payloads are now supported, the configurable BoM and linked data need an update. Therefore, variants are created and developed and link them to the feature model. Design Engineers are implementing new Design Solutions and annotate features to architecture, e.g. for strengthened chassis and a stronger steering motor (Figure 44).

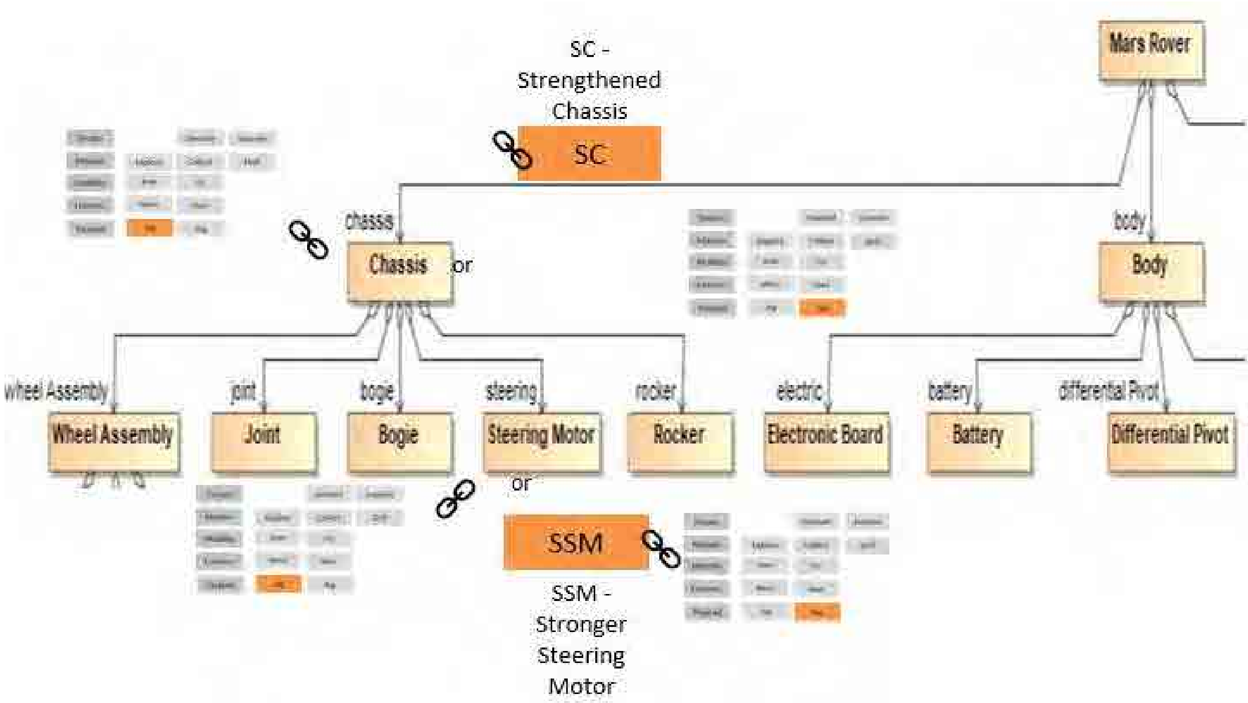


Figure 44 Annotation of Features in the Architecture Diagram

Step 5) Instantiation deriving 100% Feature Selection & BoM (Figure 45**)**

Sales Representative:

"For my new client, I translate needs into features, resulting in a Bill of Features for their individual Configure to Order (CtO) Rover. Manufacturing takes my 100% BoM and starts production."

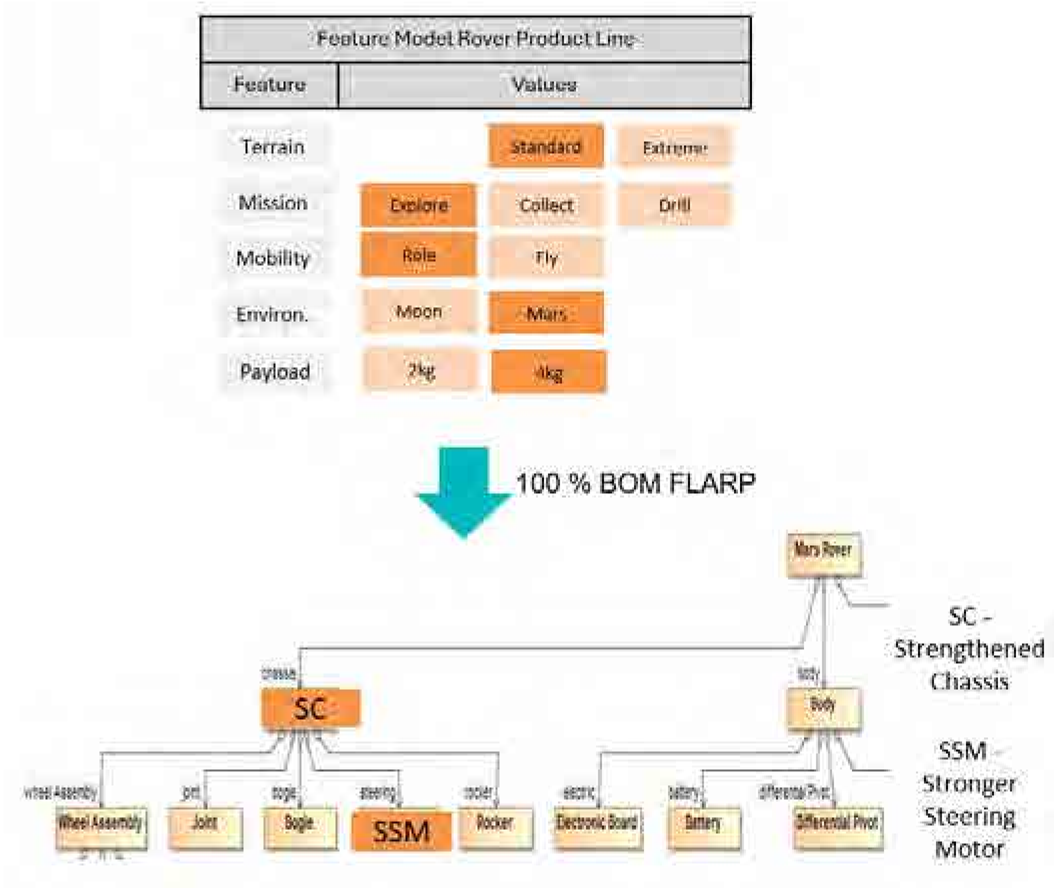


Figure 45: Instantiation of 100% Feature Selection and 100% BOM

Conclusion

The Mars Rover demonstrator shows how the WP6 CVM methodology enables cross-domain consistency, structured variability definition, and automated configuration execution within a collaborative environment.

By tightly integrating requirements, feature modeling, configuration rules, and product structures, the demonstrator validates the feasibility of CDT-enabled variant management and highlights the scalability of the approach, from customer-driven changes to fully configured product instances.

6.6.2 Mars Rover Resume

Since requirements can be contradicted, they cannot be implemented in a single system version. Their variation leads to the creation of product lines. CDT uses requirements to systematically plan variants as assets for reuse across product generations. The practical example with the Mars Rover covers essential concepts for the application of configuration and variant management (CM/VM) and clearly illustrates the end-to-end modeling of a Mars Rover product line – from the definition of features and the modeling of rules to the automated generation of variants and parts lists. The implementation of these methods in modern PLE toolchains, e.g. ALM with feature-based VM tools and CAD, particularly regarding the simulation of causal relationships and automatic variant generation, is an emerging field of activity. The development and implementation of standardized principles for designing, naming, structuring and applying functions, restrictions and configuration rules to ensure consistency and scalability across product architectures is a key task. In addition, performing KPI-driven impact analyses for configure-to-order (CTO) scenarios is a useful way to evaluate how configuration decisions affect reuse potential, costs, development time and resource utilization.

6.6.3 AI in Collaborative Configuration & Variant Management

AI will profoundly reshape configuration solutions within the manufacturing domain, with the strongest impact occurring in two key areas: The modeling of product variability and the execution of configuration processes. Since any configuration solution depends on well-structured product models that define features, constraints, and rules, AI will increasingly support the definition, analysis, and optimization of these models. By assisting in the extraction of reusable elements, harmonizing terminology, and identifying inconsistencies or missing rules, AI contributes to more robust and scalable product model structures that form the foundation of reliable configuration workflows.

Equally transformative is the impact of AI on the configuration process itself. Today, configuration is predominantly a human-driven activity, where customers, sales representatives, or engineers manually select features that satisfy all applicable constraints. In the evolving digital enterprise, AI agents will assume an active role as design collaborators, procurement negotiators, and optimization engines. These agents will request, propose, and validate configurations autonomously, using structured intents instead of manual user interaction, thereby streamlining decision-making across organizational boundaries.

A representative scenario is the design of a manufacturing system in which an AI design assistant retrieves valid subsystem configurations directly from suppliers' digital platforms. Negotiation and validation occur through formalized configuration logic rather than human intervention, enabling faster and more consistent integration across the supply chain. In this emerging paradigm, configuration shifts from a traditional user interface to a digital protocol: The standardized mechanism through which intelligent systems ensure compatibility, compliance, and optimal performance within complex, interconnected manufacturing ecosystems.

7 Conclusion

7.1 Implications for Industry

The results of the Collaborative Digital Twin (CDT) project have far-reaching implications for industrial practice, particularly in sectors characterized by complex, multi-disciplinary product development such as automotive, aerospace, and mechanical engineering. The transition from document-based, asynchronous collaboration to a model-based, semantically aligned, and digitally connected ecosystem represents a paradigm shift that will influence organizational structures, technology strategies, and business models.

The demonstrator results confirm that collaborative digital twins can significantly reduce manual effort, improve data consistency, and accelerate decision-making. For industry, this translates into tangible benefits: shorter lead times, reduced engineering rework, and enhanced traceability across lifecycle stages. Moreover, the ability to synchronize requirements, system models, and simulation data across partners supports early design convergence, mitigating costly late-stage changes. Based on the feedback of the project group this includes up to over 10% design & configuration management hours reduction and up to over 10% development lead time reduction. These efficiencies contribute directly to improved competitiveness and cost control in increasingly volatile markets.

Industry must recognize that collaborative digital twins are not merely a technological enhancement but a catalyst for organizational change. Traditional customer–supplier hierarchies and sequential development processes are insufficient to meet the demands of shortened development cycles, increased product complexity, and growing regulatory requirements. CDT demonstrates that successful implementation requires a holistic transformation across five dimensions: **mindset, relationships, methods, technology, and information**. Companies must move toward trust-based partnerships, adopt agile and iterative development practices, and establish governance frameworks that enable real-time collaboration while safeguarding intellectual property.

To foster these changes in cross-enterprise collaboration standards for data exchange, semantic modeling, and configuration management are needed. CDT addressed this challenge by introducing a layered ontology framework as well as aspect models that provide semantic interoperability across domains and organizations. These ontologies are compliant with the work of other organizations such as Catena-X and accepted standards such as Req-IF. CDT laid the groundwork for explicit application of these standards in industry. Industry needs to adopt and implement **standard-compliant data models**, align with initiatives such as **Catena-X**, and adopt open architectures that support federated data spaces. Failure to do so risks perpetuating fragmented IT landscapes and increasing integration costs.

The implementation of these data models is supported by the developed CDT blueprint for digital twin-ready IT architectures emphasizing federated data exchange, semantic alignment, and modular building blocks. This approach ensures data sovereignty—a critical requirement in an era of heightened IP sensitivity and regulatory scrutiny. Industrial organizations must prepare to integrate **data space technologies** (e.g., Eclipse Data Space Connector) and implement secure identity, contract, and policy management mechanisms. These capabilities will become essential for compliance with emerging data governance frameworks such as Gaia-X and for enabling scalable collaboration across global supply chains. The CDT blueprint is a industry-agnostic template, that can be directly adopted by industry.

As product portfolios expand and customization becomes a key differentiator, configuration and variant management emerge as critical success factors. CDT provided methodologies and user journeys for managing variability in a collaborative context, ensuring that product structures remain consistent and traceable across organizational boundaries. Industry must adopt these principles to avoid uncontrolled complexity, maintain compliance, and enable scalable product line engineering. This will require integration of configuration logic into digital twin frameworks and alignment with standards such as ISO/IEC 26580.

7.2 Summary & Discussion

The Collaborative Digital Twin (CDT) project group has demonstrated how cross-enterprise collaboration in engineering can fundamentally be improved through a combination of semantic interoperability, federated IT architectures, and standardized collaboration processes. CDT did not focus on the specific development of Digital Twins, but on a concept for a connected ecosystem where data, models, and processes are shared securely and consistently across organizational boundaries.

The project delivered three major outcomes. First, it established a semantic foundation through a layered ontology framework and aspect models, enabling a common understanding of engineering data across domains. This approach addresses one of the most persistent barriers to collaboration: the lack of harmonized terminology and data structures. Second, CDT introduced a blueprint for digital twin-ready IT-architectures, designed around modular building blocks and federated data space technologies. This architecture ensures data sovereignty, scalability, and interoperability, aligning with emerging standards such as Gaia-X and Catena-X. Third, CDT operationalized these concepts through user journeys that illustrate practical collaboration scenarios, validated by a multi-partner demonstrator. These journeys cover critical phases of the product lifecycle—from requirements management and system design to simulation, integration, and manufacturing engineering—showing that synchronous, point-based data exchange is feasible without relying on traditional file transfers. These user journeys were successfully implemented by the implementors lab as a demonstrator. The demonstrator highlights the transformative power of the approach of CDT, guiding companies in streamlining and improving cross-company collaboration based on Digital Twins.

A key capability in future product development and the adoption of Digital Twins is the management of cross-domain configuration and variant management. CDT provided a structured methodology for defining feature models, identifying variation points, and establishing configuration rules within a collaborative environment. This approach supports the transition from engineer-to-order practices toward scalable configure-to-order paradigms, reducing complexity and improving reuse potential.

Beyond technical achievements, CDT emphasizes the need for organizational and cultural transformation. Implementing collaborative digital twins requires companies to adopt new collaboration models based on trust, transparency, and shared governance. Agile practices, role redefinition, and investment in competencies such as semantic modeling and model-based systems engineering are essential to realize the full potential of CDT. Here, some legal aspects of this transition are yet to be specified.

The project also highlighted the importance of ecosystem alignment. CDT does not replace existing initiatives but complements them, creating bridges between standards and technologies. Harmonization with Catena-X and related frameworks is critical to avoid fragmentation and enable interoperability on a scale.

In summary, CDT provides a clear roadmap for industry to transition from fragmented, document-based processes to a federated, semantically aligned collaboration environment. The benefits—reduced manual effort, improved data consistency, accelerated decision-making, and enhanced traceability—translate directly into shorter development cycles, lower costs, and higher product quality. While challenges remain in areas such as IP protection, regulatory compliance, and large-scale adoption, CDT establishes the foundation for future initiatives like Live Collaboration in Engineering (LCE), paving the way for real-time, collaboration across the entire V-model.

7.3 Outlook

Beyond technology, the implementation of collaborative digital twins demands cultural adaptation. Organizations must foster openness, transparency, and shared accountability while maintaining robust IP protection. This involves redefining roles, introducing agile collaboration models, and investing in workforce competencies related to semantic modeling, data governance, and model-based systems engineering. Resistance to change—whether due to legacy processes or perceived risks—must be addressed through clear communication of business value and phased adoption strategies.

Further, the developed approaches have to be operationalized in order to be easily adoptable by industry. Here, software vendors are required to adapt their products to the demands of the proposed IT-architecture. Additionally, industry needs to commit to common data standards to leverage their adoption and improve their effectiveness. As value chains become more and more complex, the developed solutions have to be adjusted. Especially industry-, domain-, and company-agnostic approaches are needed to bring together companies along the entire value chain. Lastly further user journeys shall be developed to cover all collaborative aspects of a products lifecycle.

These open points will be addressed by the prostep ivip project group LCE – Live Collaboration in Engineering – starting 2026. Based on the results of CDT further use cases will be developed and exemplary implemented with a special focus on software-defined products. Additionally, a stronger focus will be set on elaborating the industry-agnostic changes required to foster this innovation.

Abbreviations

AAS	Asset Administration Shell
ABB	Architecture Building Block
AI	Artificial Intelligence
ALM	Application Lifecycle Management
API	Application Programming Interface
AWS	Amazon Web Services
BIF	Basic Interaction Flow
BOM	Bill of Material
BPN	Business Partner Number
CAD	Computer-Aided Design
CDCM	Cross-Domain Variant & Configuration Management
CDT	Collaborative Digital Twin
CM	Configuration Management
CPO	Code of PLM Openness
CTO	Configure to Order
CVM	Configuration & Variant Management
CWA	Closed World Assumption
DDP	Digital Data Package
DT	Digital Twin
ECN	Engineering Change Notice
ECP	Engineering Change Proposal
ECR	Engineering Change Request
EDC	Eclipse Data Space Components Connector
ERP	Enterprise Resource Planning
ESMF	Eclipse Semantic Modeling Framework
IAEA	International Atomic Energy Agency
IDTA	Industrial Digital Twin Association e.V.
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
JT	Jupiter Tesselation
KPI	Key Performance Indicator
LCE	Live Collaboration in Engineering
MBSE	Model-based Systems Engineering
OEM	Original Equipment Manufacturer

OPC UA	Open Platform Communications Unified Architecture
OSLC	Open Services for Lifecycle Collaboration
OWA	Open World Assumption
OWL	Web Ontology Language
PI	Program Increment
PLE	Product Line Engineering
PLM	Product Lifecycle Management
PO	Purchase Order
RDF	Resource Description Format
REQIF	Requirements Interchange Format
RFLP	Requirement-Functional-Logical-Physical
SAMM	Semantic Aspect Meta Model
SHACL	Shapes Constraint Language
STEP	Standard for the Exchange of Product Model Data
TLO	Top-Level Ontology
TOGAF	The Open Group Architecture Framework
UJ	User Journey
URI	Uniform Resource Identifier
VDA	Association of the Automotive Industry
VM	Variant Management
WP	Work Package

Cross-Domain Variant and Configuration Management Definitions

Term	Definition
Alternative Part	BOM-term indicating that one part may be used instead of another part, defined by the interchangeability
Architecture Building Block (ABB)	An Architecture Building Block is a package of functionality that describes a single aspect of the overall model. It defines what functionality will be implemented, captures business and technical requirements, is technology aware and solution-neutral, and directs and guides the development of SBBs.
Artefact	An artefact represents the results of a process. For instance, requirement analysis, models, review results, or a glossary. An artefact may be structured hierarchically.
Aspect Model	An Aspect Model is a formal, machine-readable description of the structure and semantics of data related to a specific aspect. Aspect Models are specified using the Semantic Aspect Meta Model (SAMM), a specification provided open source at the Eclipse Semantic Modeling Framework (ESMF). It is used to organize the digital representation of information, capture the semantics of the domain, and create a clear context for related data. Aspect models can be reused in similar use cases and are stored in an aspect model catalog to provide quick and easy access for authorized users and applications. They support various formats (e.g. XML, JSON) and ensure consistency and interoperability in data representation.
Baseline	An immutable set of versioned Configuration Items, including software, hardware, and documentation, or Configuration that are used as a reference point for future changes or updates.
Bill of Features (BoF)	See: Variant Configuration
Bill of Material (BoM)	A Bill of Material (short: BoM) is a comprehensive list of raw materials, components, and assemblies required to manufacture a product, along with the quantities needed. Depending on their context they can be structured differently and named Engineering BoM, Manufacturing BoM, Design BoM, …
Branch	OMG Definition: “Branch is an indirect subclass of Record (via CommitReference) that represents an independent line of development in a project” Previously:”A variant of a configuration item, such as a software component or a document, that is created by making changes to the original configuration item” (without effect to the original)
Capability	An expression of a system, product, function, or process ability to achieve a specific objective under stated conditions.
Comment	A comment is a business relevant information that emerges from collaboration processes and is an additional information to other business objects but not part of the product data.

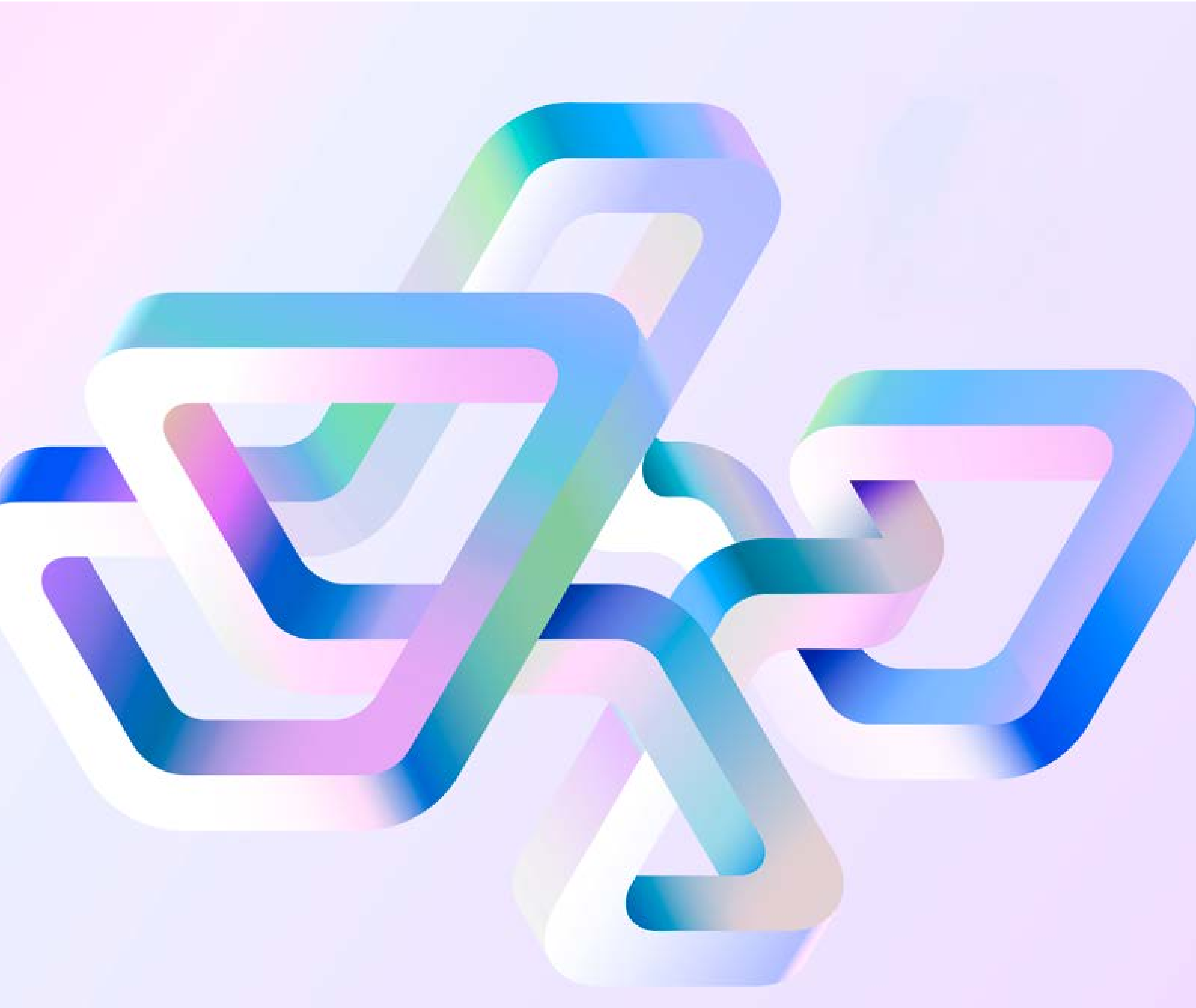
Term	Definition
Configurable BoM	A Superset-BOM that additionally contains selection conditions to include and exclude parts towards specific needs.
Configuration	A configuration is the result of a Configurable BOM configured for a specific purpose leading to a structure of selected Configuration Items and sub-configurations
Configuration Item (CI)	A CI is an administration/management envelope for an individual work product. Its purpose is the identification of collaborative and re-usable content allocated to a single release responsibility.
Configuration Rule	Defines a logical relationship between features or product using a Boolean-Logic.
Configure-to-Order (CTO)	A customer can compose an end product by selecting features linked with pre-defined component variants.
Coverage status	Find an overarching definition that can be considered valid not only for requirements but also for product engineering data
Data Space	A Data Space is a federated, decentralized, digital infrastructure enabling secure data exchange while maintaining data sovereignty. The participants of the dataspace can decide how, when, and with whom they share their data. To ensure a compliant and trusted way of data exchange all data providers and consumers need to mutually agree to rules. This ensures interoperability and trust, facilitating cross-company collaboration.
Digital data package (DDP)	A digital data package (DDP) is a container file that contains three types of items: 1.Domain-specific neutral format files (i. e. ReqIF, SysML, JT, STEP, FMI, SSP, I++, etc.), or URLs to resources (i. e. using OSLC, RDF) 2. A dictionary, listing all relevant entities & relations, contained in the neutral format files, each with a reference to the source 3. A visual and interactive content overview of all contained entities & relations
Digital Shadow	A digital shadow depicts the externally created data profile of a component or a product. The data profiles are created by means of algorithms that include the acquisition, analysis, evaluation, and consolidation of data. The digital shadow is used as the foundation for decisions.
Digital Twin (DT)	A Digital Twin is a virtual identifiable representation of a real-world entity (product, concept, process), which is bidirectionally connected to it at a specified frequency and fidelity."
Domain	A domain is a specific area of expertise or functionality within a digital twin, encompassing related configurations, parameters, and data relevant to that aspect of the product or system.
Engineering Application	An Engineering Application is an application that supports the engineering of a product or service within its different lifecycle phases, e.g. in design, modelling, analysis, simulation, and management (e.g. CAD software, PLM systems, simulation software, etc.).
Engineer-to-Order	The product is adapted to specific customer requirements. Unlike CTO, this adaptation requires engineering effort.

Term	Definition
Feature	Characteristic of a product conceivable by the customer (e.g., OEM or end user). In case of variability, the same feature can allocate different characteristics
Feature Catalogue	Definition in PLE norm (ISO 26580) A Model of the collection of all of the features and feature constraints available across the entire product line
Feature Model	Model of the collection of all the feature options and feature constraints available across the entire product line.
Federation	Federation refers to a structural approach that facilitates the decentralized exchange of data between multiple participants within and across data spaces. It enables the direct exchange of data between participating partners without the need for centralized storage. The federation is characterized by harmonized processes, formats, and semantics/ontologies which ensure interoperability across different systems and technologies.
Function	A specific purpose or activity that a system, component, or process is designed to perform to achieve a desired effect or outcome. (see ISO15288)
HiL (Hardware-in-the-Loop)	HiL means a test environment in which control units in the development stage can be tested in real-time operation on a model of their later operating environment. This makes it possible to analyze the developed device at an early stage.
Instance	A specific realization reflecting a variant configuration which is derived from 150% or configurable BoM/ engineering artifacts. A Customized product instance with individual properties.
Interchangeability	Property of a component or system to be replaced with another without requiring additional modifications, ensuring same functionality and performance.
Link Item (LI)	A link item (LI) is a reference point, or, an "anchor", inside a CI. It can be compared with a chapter (LI) or section heading inside a document or book (CI). It is versioned only if the containing CI is versioned, same as a chapter in a book is not published (versioned), but only the whole book.
Ontology	An ontology provides a formal naming and definition of the artefacts of a domain and how they are related, by defining a set of concepts and categories that represent that domain.
Parameter	A set of values that is used to control the operation of a function or that is used by a function to compute one or more outputs
Parametrization	The process of defining and configuring parameters that determine the specific characteristics and behaviour of a system, allowing for customization and flexibility in its application.
Persistent Reference	A Persistent Reference is a unique reference used in data objects (e.g. a reference from product data to a related CAD model) that consistently refer to a digital resource. They must be valid throughout the complete lifetime of the data object, regardless of changes in location, format, or ownership of the resource.

Term	Definition
Platform	A foundational system or environment that supports the development, integration, and operation of various applications, services, or products, often providing core functionalities and common standards.
Portfolio	A collection of products, services, or projects offered by a company, managed together to achieve strategic objectives and address diverse market needs.
Product Asset Instance	Definition in PLE norm: An instantiation of a shared asset specific to a Member Product, automatically produced by the PLE Factory Configurator, corresponding to a Bill-of-Features for that Member Product
Product Family	See “Product Line”
Product lifecycle management	Product lifecycle management (PLM) is the process of managing the entire lifecycle of a product from its inception through the engineering, design and manufacture, as well as the service and disposal of manufactured products. PLM integrates people, data, processes, and business systems and provides a product information backbone for companies and their extended enterprises.
Product Line	ISO/IEC 26580: family of similar products with variations in features and functions
Product Line Engineering Artifacts	Work products and artifacts associated with the product line, including all content needed by any of the product variants.
Product Model	A product model is defined by a set of features, feature values, and rules. The rules declare the available value combinations for features. The Product Model is the input to the configurator that helps users find valid values for the features.
Product structures / BoMs	Description of the structure of a product in terms of its nested bill of material items (BOM items). Note 1: BOM items for tangible products Note 2: Distinct types of bill of materials may only include one level of hierarchy. Note 3: BOM item may also include documents or processing instructions. Note 4: BOM has different usages according to the product lifecycle phase (Sales, Engineering, Software, Electronic, Manufacturing, ...) Note 5: BOM may differ by structural type (see specializations)
Product Variant	A product belonging to product line - i.e. it is derived from product line.
Product Variant Engineering Artifacts	Work products and artifacts specific to a product variant which are derived from product line model based on the corresponding variant configuration.
Selection Condition	Logical expression referring to features that specifies when to include an assembly or part in a mBOM.
Shared Asset	ISO 26580: Software and systems engineering lifecycle digital artefacts that compose a part of a delivered Member Product or support the engineering process to create and maintain a Member Product

Term	Definition
Shared Asset Superset	ISO 26580: representation of a Shared Asset that includes all content needed by any of the Member Products
SiL (Software-in-the-Loop)	Software-in-the-Loop testing (SiL testing) means testing embedded software, algorithms or entire control loops with or without environment model on a PC.
Solution Building Block	A Solution Building Block is a candidate solution that conforms to the specification of the assigned Architecture Building Block (ABB). It defines what product, or component will implement the functionality, defines the implementation, fulfills business requirements, and is product- or vendor-aware. For example, a SBB can be a tool or a tool chain.
Solution Space	A collection of valid feature/value combinations for a given subset of the features in a product model.
Stream	A mutable configuration (especially relates to cross domain configurations in CDCM-Context). Contained configuration items can be streams or baselines.
Superset BOM	A BOM that contains optional and alternative elements from which a BOM for a single product can be derived.
System model	A system model represents aspects of a system and its interactions with its environment. It is a conceptual representation that helps us understand, analyze, and design complex systems.
Test cases	In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.
Unit	A discrete component or entity within a system that performs a specific function or holds a measurable value, often part of a larger structure or process.
Unit of Configurability	Version-independent object under configuration management control.
Variability (Variance)	ISO 26550: Characteristics that may differ among members of the product line
Variability Model	ISO 26550: Explicit definition for product line variability Note: It introduces variation points, types of variation for the variation points, variants offered by the variation points, variability dependencies, and variability constraints. Variability models may be orthogonal to or integrated in other models such as requirements or design models. There are two types of variability models: Application variability models and domain variability models.
Variant	ISO 26550: One alternative that may be used to realize particular variation points ISO 26580: Single alternative that can be used to realize a particular Variation Point
Variant Configuration	The specification for a product variant based on the feature selection.

Term	Definition
Variation	A variable artifact in a product line engineering artifact. Variation is controlled by its underlying variation restriction (e. g. boolean terms; syn.: variation condition) which is based on the features defined in the feature model.
Variation Model	Model that contains variation points, links between variation points and requirements, variation rules (dependencies between variation points) and parametrization of variation points
Variation Point	A model element or domain artifact within shared asset superset content that marks a location of variability. It is triggered by conflicting requirements and defines where alternative variants may apply and is resolved through feature selections to produce a specific product asset instance, typically for technical or commercial differentiation.
Version	Reproducible, immutable and referable state of a configuration item and its contents.
Work Product	Result of a process activity that is (usually) released for utilization and subsequent processing. Considering the differentiation between core processes (value adding) and support processes (no value adding), core process work results must always be treated as CIs.



prostep ivip association

Heinrich-Hertz-Straße 3-7
64295 Darmstadt
Germany

Phone +49-6151-9287336
psev@prostep.com
www.prostep.org

ISBN 978-3-948988-46-3
Version 1.0, 2026-6