White Paper

Platform Engineering

Towards pervasive, robust and trustworthy platforms

fortiss

Authors:

Tomas Bueno Momčilović
fortiss GmbH
Munich, Germany
momcilovic@fortiss.org

Dian Balta
fortiss GmbH
Munich, Germany
balta@fortiss.org

Alexander Perzylo
fortiss GmbH
Munich, Germany
perzylo@fortiss.org

Contributors:

Dominik Mittel
Ingmar Kessler
Junsheng Ding
Mahdi Sellami
Peter Kuhn
Yannick Landeck
Yuchen Zhou
fortiss GmbH
Munich, Germany

Content

1. Abstract

We are researchers in the competence field of Platform Engineering at fortiss and conduct cross-domain research for developing pervasive, robust, and trustworthy platforms. Our approach encompasses a broad spectrum of methods, including semantic knowledge representation, formal requirement and capability matching, as well as traceability mechanisms for verifiable argument-based transactions between heterogeneous actors. This practice-proven engineering method and its associated tools allow for rapid prototype development, capturing relevant perspectives and facets for making informed decisions and steering the journey towards effective platformization.

2. Platformization is omnipresent

While software is "eating the world"[1], platforms are determining whose software will do it and how. What started as a vision of a few maverick startups has become a phenomenon that grew into the goliaths of today, such as Amazon, Android or OpenAI.

It set the stage for a wider movement of platformization, representing a softwarebuilding purpose – *for the users, by the superusers* – and drawing an increasing number of digital service pioneers.

Platforms are the software-defined infrastructures of the digital world. Taking inspiration from wooden platforms that support the builders of our physical cities, platforms serve as the foundation, the support, and the material upon which digital cityscapes are built. Like bricks in high-rises, expert-optimized, packaged, and reusable pieces of code are the building blocks that may be stacked on top of each other. *Platformization* is thus the process of arranging these blocks so that the digital buildings are livable and useful to their users. Given the potential of platforms, the European Union has taken initiative to establish their "controllability, portability and interoperability"[2]

Unsurprisingly, platformization is also transforming traditional industries, where software has usually played a subordinate role. Automakers are orchestrating the swarm intelligence of automated vehicles, stoplights, and individuals on platforms[3]. Farmers are monitoring their crops remotely via dashboards that integrate machines both on land and in the air[4]. Manufacturers are tracking complex components that pass through many human and robotic hands via distributed ledgers and knowledge graphs[5]. Energy distributors are managing their transformation to environmentally-friendly sources via federated platforms (see Fig. 1). Even governments, which are said to be traditionally innovation-reluctant, are combining their fragmented know-how to automate administration and build proactive public services[6].

Only few cases truly benefit from complete isolation. Still, not every platform will be a success. To avoid being puzzled when the costs are high and impact is critical, we start with a question from the technological point of view:

How can a platform be engineered for success?

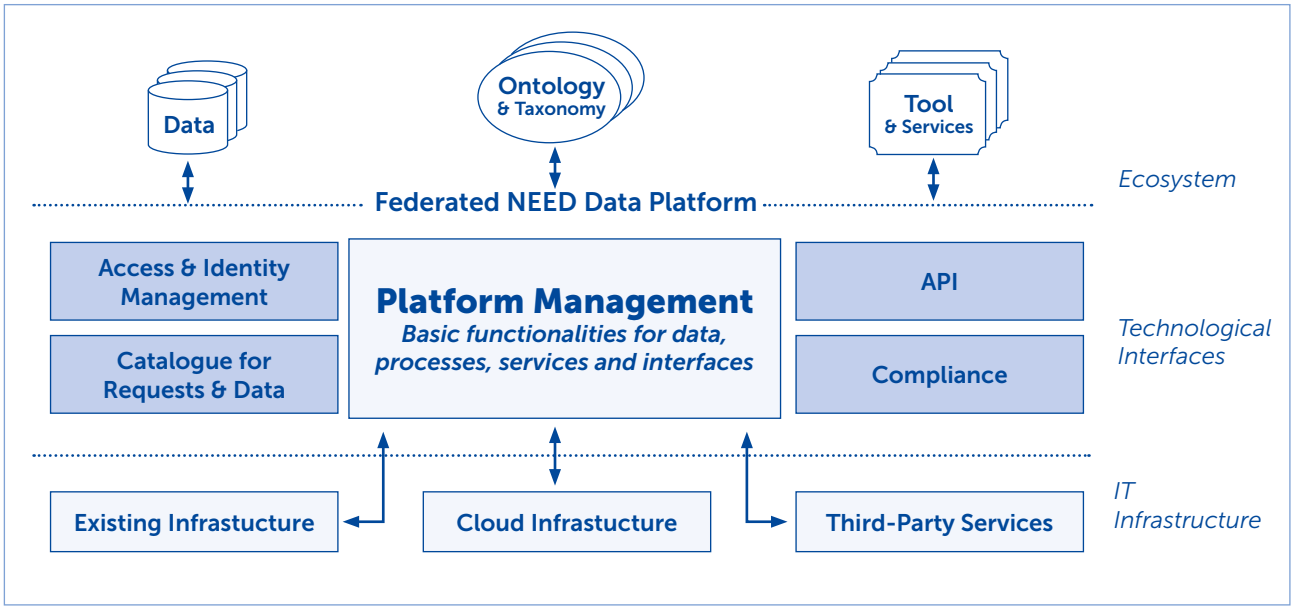


Figure 1: An exemplary architecture of a platform for the energy domain (Source: Project NEED[7]).

1 Andreessen, M. (2019). Why Software is Eating the World. Andreessen-Horowitz. Link: https://a16z.com/why-software-is-eating-the-world/ **2** Platform GAIA-X: https://gaia-x.eu **3** Echeto, J., Santos, M. & Romana, M. G. (2022). Automated vehicles in swarm configuration: Simulation and analysis. Neurocomputing, 501, 679-693. **4** Project FarmExpert: https://www.fortiss.org/en/research/projects/detail/farmexpert-40 **5** Project DiProLeA: https://www.fortiss.org/en/research/projects/detail/diprolea **6** Project DRP: https://www.fortiss.org/en/research/projects/detail/drp **7** Project NEED: https://www.fortiss.org/en/research/projects/detail/need

3. The platformization puzzle

Rarely do modern organizations build their own operating systems anymore; even inhouse tools can hardly compete with generic but well-maintained services. The paradigm shift that came with the Internet led organizations to distribute their monolithic systems towards common infrastructures. At face value, platforms seem like the continuation of this trend.

However, not all networks lead to functional platforms. Digital ecosystems are inherently complex (see Fig. 2), and the success of hosting a central node depends on the host's capacity to not only build and orchestrate the network, but grow, maintain, and endure pressures. The risk of failure is not negligible: as of a 2019 study[8], most (80%, US-based) platform hosts could not deal with the ever-growing interdependencies between components and actors with competing incentives, nor their dynamic behavior, and resorted back to legacy solutions[9]. Only large players with dedicated teams managed to power through, leaving others to depend on their often-monopolistic services.

Hosting a platform is a grand goal, but solving this puzzle with dynamically changing pieces is not trivial. Every industry today is impacted by the growing body of regulation and standards, posing critical requirements for compliance and certification. The advent of technological breakthroughs further complicates the picture. Organizations must ask:

- How do we select and integrate suitable technologies?
- How do we orchestrate the elements of a growing, intertwined, and heterogeneous ecosystem?
- How do we build capacity and trustworthiness with the knowledge distributed across domains and actors?
- How do we maintain trust and capacity in an open, dynamic network of actors, whose incentives are not aligned?

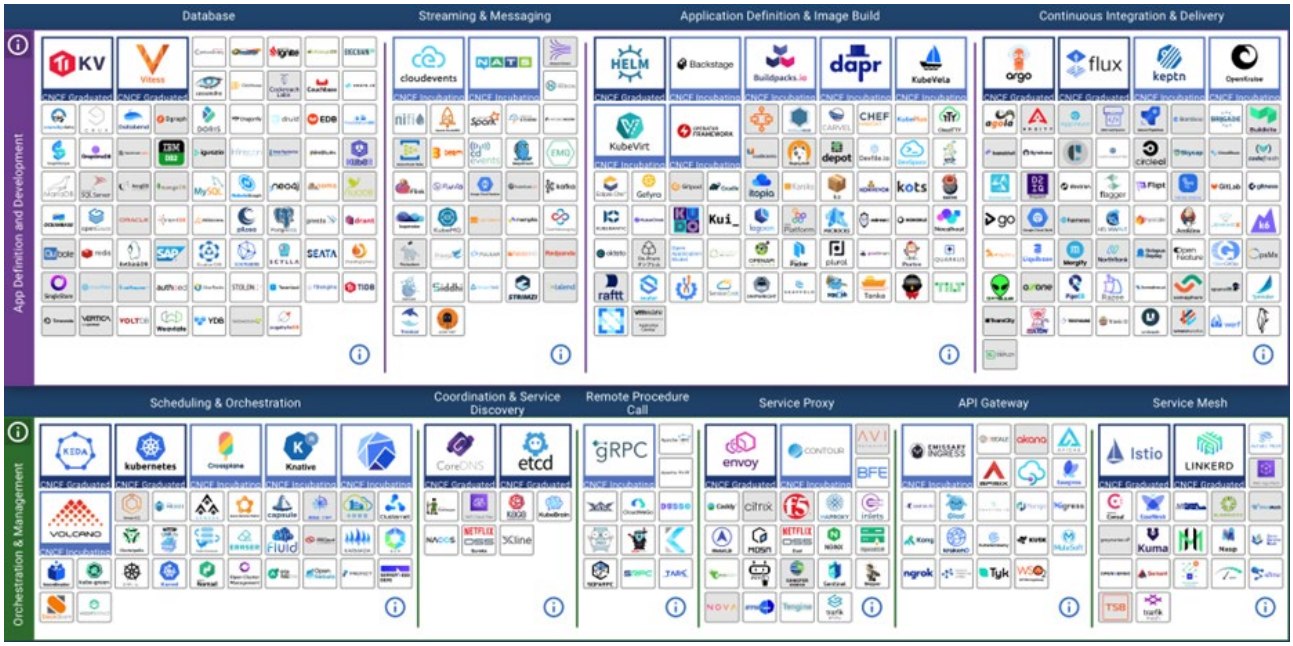


Figure 2: One part of the puzzle; a sketch of technologies for cloud-based platforms (Source: CNCF[10]).

8 Cusumano, M. A., Gawer, A., & Yoffie D. B. (2019). The Business of Platforms: Strategy in the Age of Digital Competition, Innovation, and Power. Boston, MA: Harvard Business School. **9** Van Alstyne, M. W., Parker, G. G., & Choudary, S. P. (2016). Pipelines, Platforms, and the New Rules of Strategy. Harvard Business Review. Link: https://hbr.org/2016/04/pipelines-platformsand-the-new-rules-of-strategy
10 Cloud Native Computing Foundation (CNCF). (n.d.). CNCF Cloud Native Landscape. Link:https://landscape.cncf.io/

4. Platform Engineering in a nutshell

We map out the platformization puzzle and provide two central missing pieces: an engineering method and a tailored toolbox. The engineering method, referred to as model-instantiate-apply and abbreviated as **MIA**, provides a succinct process-like summary of our approach to tackling platformization challenges. The tailored toolbox builds atop established software tools and knowledge models. We solve the remaining puzzle with three parallel streams of research activities (cf. Fig. 3), to achieve the crucial qualities of a successful platform: *pervasiveness, robustness, and trustworthiness*.

First, a platform can only be *pervasive* if it is provably useful to its intended users and lightens their workloads. We refer to this as **knowledge augmentation**. All platforms provide knowledge, but not all do it successfully. Instead of enforcing rules of conversation that overwhelm, discourage, or distract users away from valuable work, platforms should be enabled to autonomously reason about given contexts and ask for appropriate information as needed, where needed. We enable this by modeling and instantiating the world of the users, so that both humans and machines can describe and understand the application domain in their own lingo, and the platform translates and coordinates.

Second, a platform can only be *robust* if it is useful across time and problem-space. We refer to this as **capability exploration**. Even the most dedicated organizations cannot reliably solve all problems of a growing number of systems, processes, and stakeholders. Nor should they: Platforms should support actual users in solving their problems, their way. We enable this by capturing the logic of software and hardware alike, and integrating user-picked tools as interfaces with the platform, so that everyone can draw on the best problem interpretations and reality-grounded solutions.

Third, a platform can only be *trustworthy*, if its members can work responsibly and unconcerned. We refer to this as **accountability design**. Platforms that betray their users or owners fail, regardless of their past value or creators' intentions. Trust requires mechanisms that prevent wrong information[11] from being shared with wrong members[12], and timely and corrective reactions to wrong actions. We believe that 'trustless', inscrutable code for enforcing rules is not enough, and that platforms should let appropriate and accountable individuals establish the rulebook, and decide *which* rules are applied *when*. We enable this by assuring correctness with traceable, interlinked, and verifiable claims.

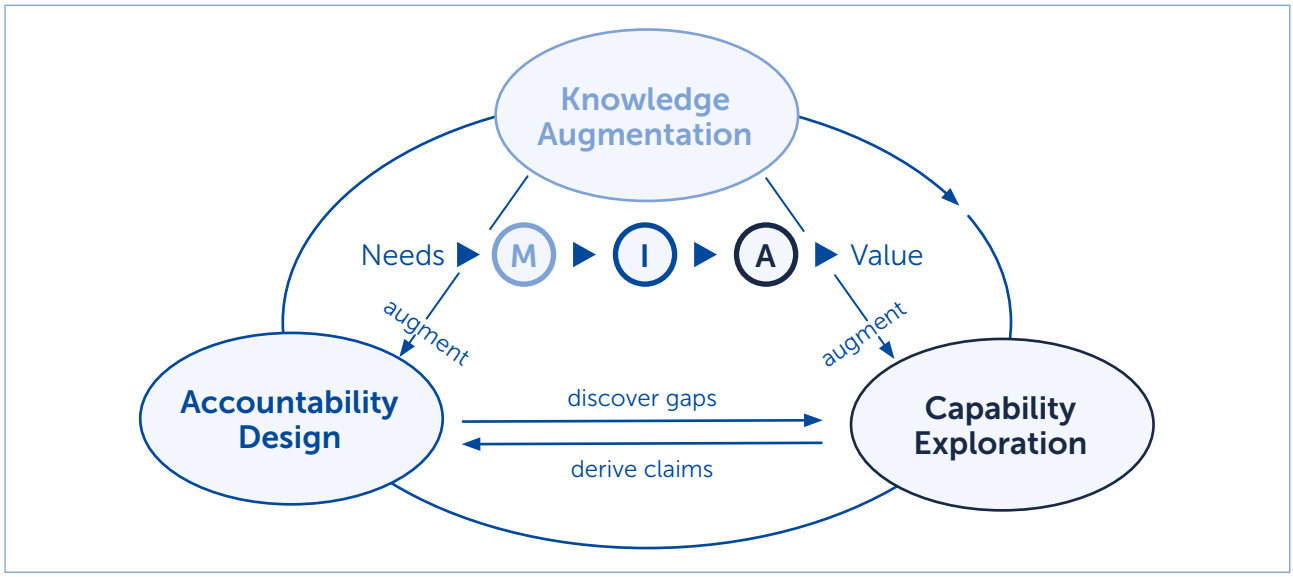


Figure 3: Our approach to Platform Engineering.

11 A common problem is privacy and intellectual property protection. See, for example: Foerderer, J., Kude, T., Mithas, S., & Heinzl, A. (2018). Does Platform Owner's Entry Crowd Out Innovation? Evidence from Google Photos. Information Systems Research, 29 (2), 1-18.
12 One known problem from sharing too much information about the platform includes "forking": i.e., the unsanctioned cloning of a repository and business logic. See, for example: Karhu, K.,Gustafsson, R., & Lyytinen, K. (2018). Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks.

5. Model, instantiate, apply (MIA)

Platforms are built on a foundation of knowledge representing the needs, systems, and ecosystems of stakeholders. The approach to representing and reasoning about this knowledge is what the **MIA method** is all about (cf. Fig. 4). It involves the formal modeling of knowledge, the instantiation of these models for specific descriptions of platform or stakeholder resources, and their combination and deployment for particular applications.

When *modeling*, we aim at establishing a common understanding between all stakeholders. Together with domain experts, we capture the relevant concepts, attributes, and relations of the domain, as well as the needs and requirements from the ecosystem of specifications, standards, and regulation. This knowledge is formally encoded in ontologies and enables the automatic construction of knowledge graphs, which provide a common, semantic, and interlinked vocabulary across the platform (i.e., a reusable and shareable semantic model).

When *instantiating*, we support the owners of each physical or digital asset to formally represent relevant properties and capabilities. With the technical means and the personal aid that we provide, owners transform their data (semi)automatically, e.g., making existing data available through virtualization. New data is made to be directly retrievable from the knowledge graph itself, and often encoded in the Web Ontology Language (OWL) and Resource Description Framework (RDF).

When *applying*, we support the complementors (i.e., the "superusers" and developers) to deliver the value to the users. Developers can connect their application software to the knowledge graph APIs, select and retrieve the relevant semantic models and asset representations, and perform reasoning and querying tasks on the integrated data from heterogenous sources. Applying further makes the program logic of applications explicit to other platform complementors, reducing the isolation of logic in source code, and bridging the knowledge gaps of different departments and organizations.

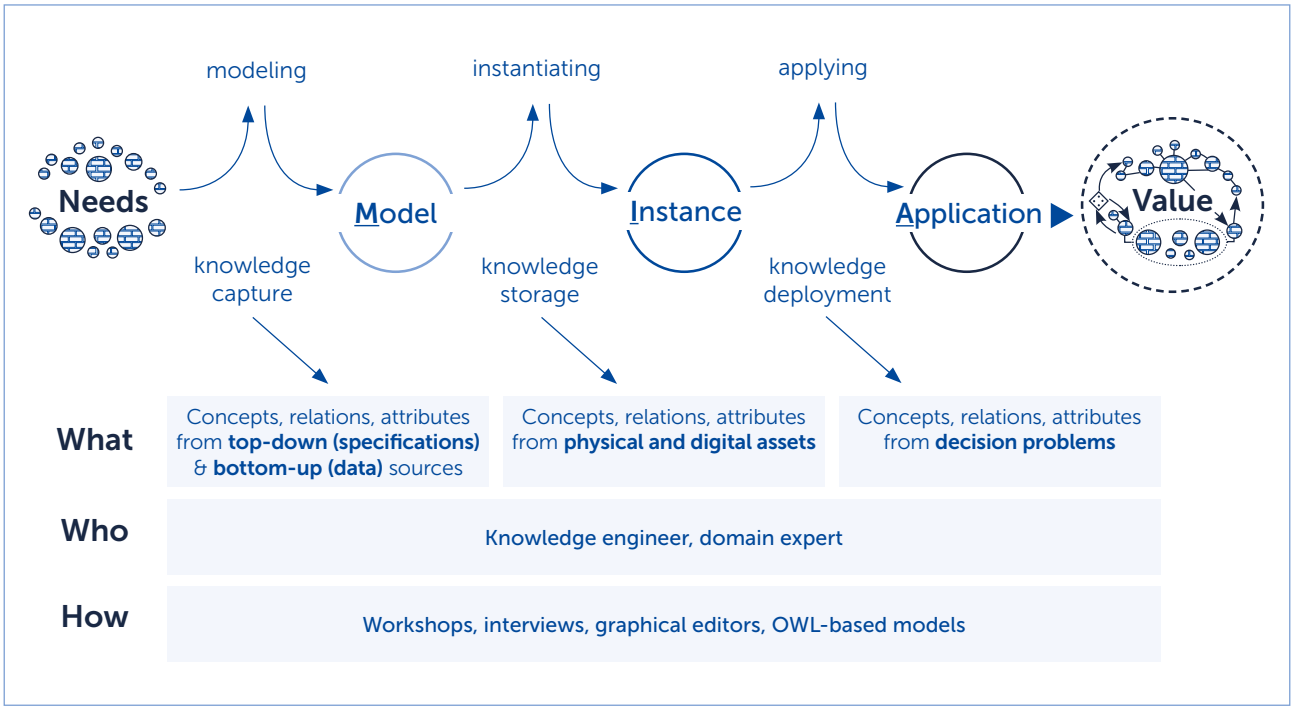


Figure 4: From needs to value with the MIA method.

6. Platform Engineering applied across domains

We introduce platforms to the use-cases that need pervasive, robust, trustworthy, and collaboration-enabling infrastructure at any level. As a research institute, we apply scientific solutions to your challenges, and shape your ideas and infrastructure into proofs-of-concept that demonstrate the feasibility and valuecreation potential of platforms.

Our portfolio includes several industrial and public challenges (cf. Fig. 5). Readers living in Bavaria may have interacted with one of our platforms. Our team has collaborated with Bavarian public departments to deliver their data[13] and services[14] to the digital world following the introduction of the Online-zugangsgesetz (OZG). We integrated (dis)embedded and (dis)embodied systems into semantic twins to platformize large airlines, small and medium-sized manufacturers[15], and distributed energy networks[16].

Moreover, our approach has been applied to assuring accountability and compliance of platform-driven manufacturing[17,18] and financial[19] operations, with graph-based traceability mechanisms (incl. distributed-ledger technologies) that make capabilities auditable.

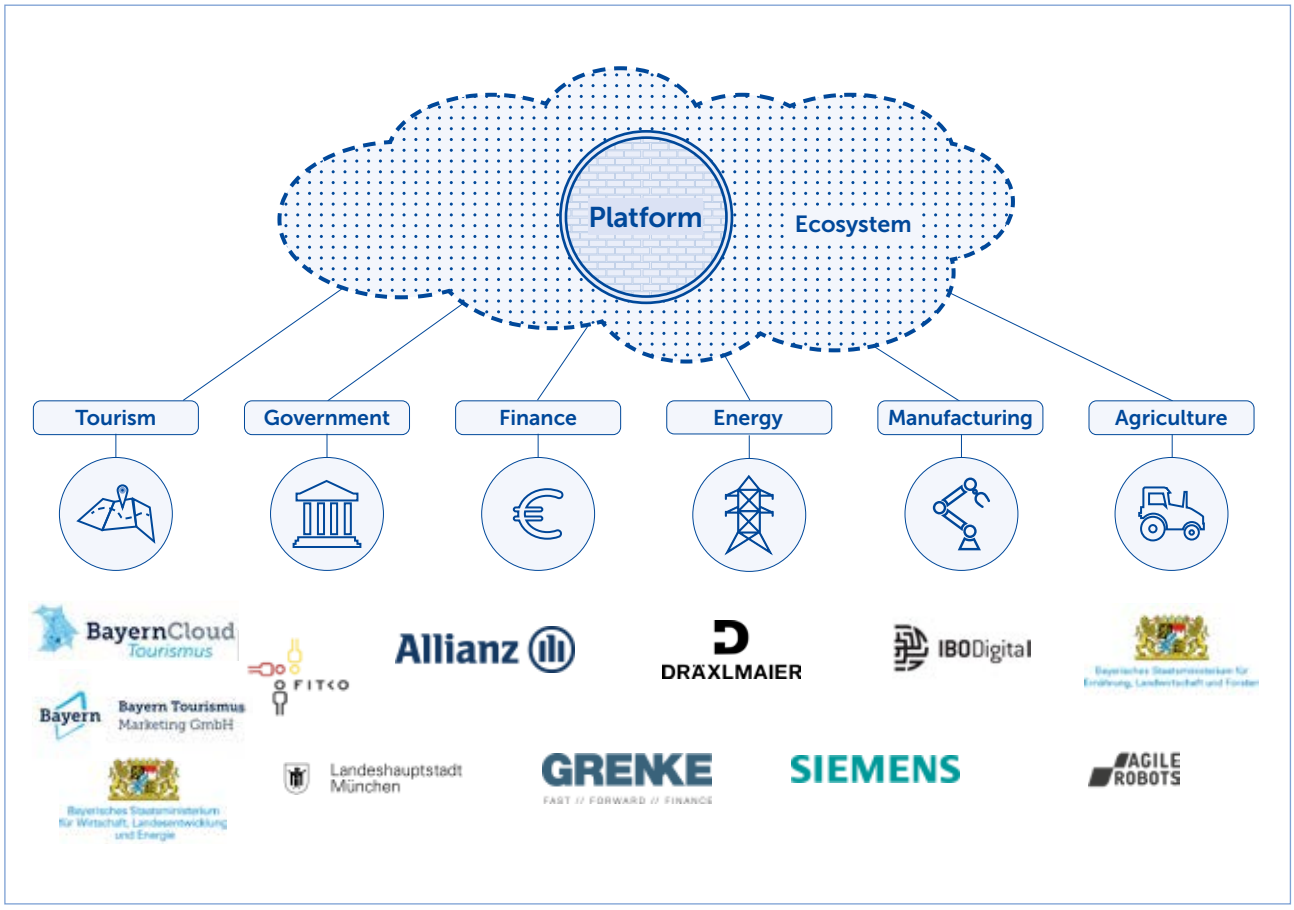


Figure 5: Industries and partner organizations where our team has made an impact.

13 Project KI-SusCheck: https://www.fortiss.org/en/research/projects/detail/ki-sus-check **14** SmartButler, IBM & Aberto „VaterSmart" Hackathon: https://www.fortiss.org/en/news/details/asmart-butler-for-the-administration-of-tomorrow **15** Project SMErobotics: https://www.fortiss.org/en/research/projects/detail/smerobotics **16** Project BEST: https://www.fortiss.org/en/research/projects/detail/best **17** Project DiProLeA; see above. **18** Project ASCA4IE: https://www.fortiss.org/en/research/projects/detail/asca4ie **19** Projects FLA and FinComp: https://www.fortiss.org/en/research/projects/detail/fla

7. Exemplary use-case

One prominent area of our impact involves high-precision manufacturing and delivery of parts in safety-critical industries. Our customized platforms are built from the needs and infrastructure of the manufacturing company and its partners.

To build parts that ensure safety of downstream products and systems, employees must take extreme care to capture, store, and deploy knowledge:

- record all functionalities, dependencies, and provenance (i.e., thorough bills-of-materials);
- manufacture carefully or delegate to trusted partners; and
- request and provide evidence regarding the product quality and changes throughout its lifecycle.

Our role has been to enable the highly impactful work to happen as seamlessly as possible, by guiding all users and partners through the platformization streams. We modeled the world of domain experts (product engineers), i.e., their products, manufacturing processes, involved hardware and software resources, and related skills, in order to formally and semantically specify their requirements and development activities. Combining the mentioned models, we established mechanisms for quality verification, validation, and matching of process and product requirements with manufacturing capabilities. Automatically generated and immutably stored claims and supporting evidence of those mechanisms enable users to control version changes, detect faults, and analyze root causes in problems.

The resulting proof-of-concept platform (see Fig. 6) represents an integration of three layers: a core layer containing a semantic knowledge base and a blockchain-based trust layer; a complementor's layer for attaching partner tools that interact with the semantic knowledge in the core layer; and a user's layer that provides customer specific interfaces and connections to their machines and devices.

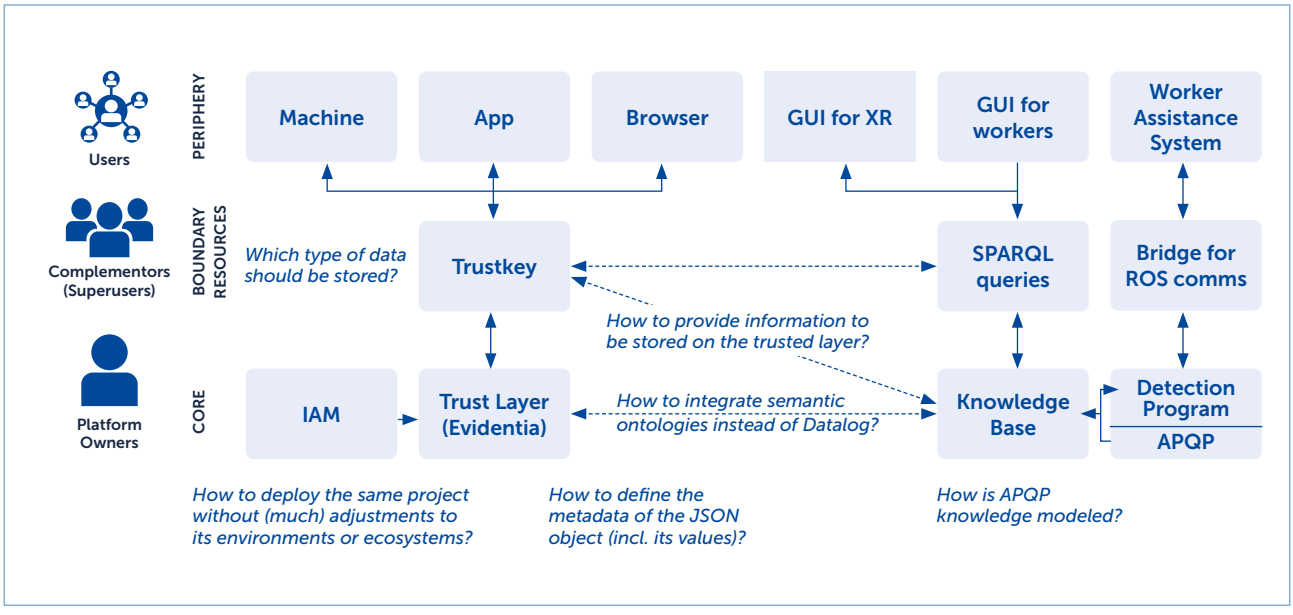


Figure 6. Simplified analysis of a manufacturing platform and its challenges (Source: Project DiProLeA[20]).

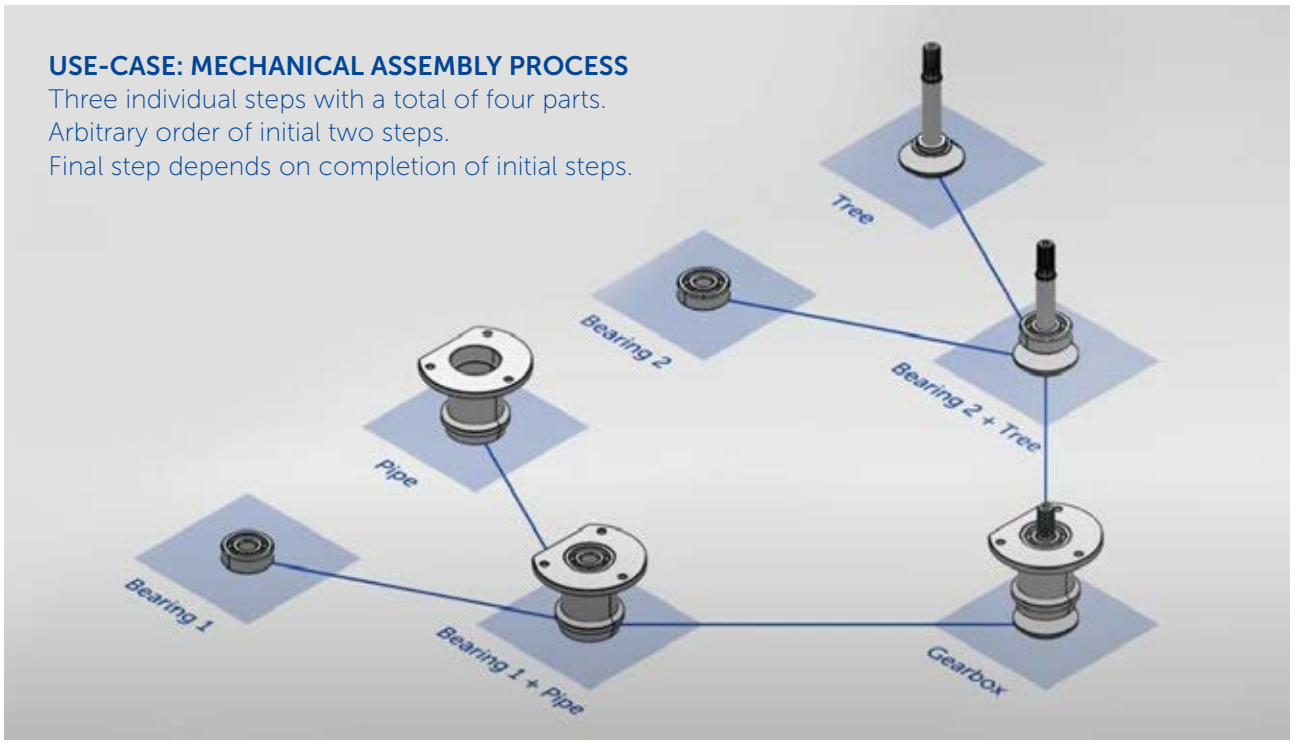
20 Project DiProLeA; see above.

8. Demonstrator

We provide a glimpse of our work with a platform-demonstrator in the fortiss Labs[21]. The demonstrator is a platform for teaching industrial robots to assemble workpieces by instructing them visually. This intuitive example of human-robot interaction is a meld of a drag-and-drop interface, a KUKA iiwa robotic arm, and a custom ontology-based task representation and interpretation layer for translating human instruction to machine action.

The demonstrator reflects our concept to address the trend towards mass customization in the manufacturing sector. Mass customization often leads to smaller batch sizes with automation approaches not being financially viable due to their inefficiency in programming and operation. A suitable solution must lower the need for extensive training of operators and shorten the time of adapting the production system to new manufacturing tasks through intuitive instruction paradigms. In addition, the under-lying platform can be scaled up to production networks, which support the distribution of tasks across organizations by matching organizational capabilities with manufacturing requirements and creating verifiable claims between them.

This demonstrator is the result of extensive work with the industry on the challenges of remote, rapid, and reliable mechanical assembly. More importantly, it shows the value and feasibility of a validated platform, whose users (i.e., cross-domain operators with no prior knowledge of robot programming) formulate correct task sequences, interpretable prompts for the assembly, and tools for high-precision actions. Anyone can use and extend such a system, whether from the field, the factory floor, or the office.



Photos: fortiss Labs platform for the robot-based assembly of workpieces with a drag-and-drop interface[21].

21 fortiss Labs webpage: https://www.fortiss.org/en/research/fortiss-labs


fortiss
fortiss

Imprint

Published by
fortiss GmbH
Guerickestraße 25
80805 München

Layout
fortiss GmbH

Print
viaprinto GmbH & Co. KG

ISSN Print
2699-1217

ISSN Online
2700-2977

1st issue
February 2024

Photo credits
Title: AdobeStock 637057734
Page 5: AdobeStock 666702324
Page 9: fortiss GmbH
Page 13: fortiss GmbH
Page 14: fortiss GmbH @ Kathrin Kahle

Find here more fortiss Whitepaper

fortiss is the Free State of Bavaria research institute for software-intensive systems based in Munich. The institute collaborates on research, development and transfer projects together with universities and technology companies in Bavaria and other parts of Germany, as well as across Europe. The research activities focus on state-of-the-art methods, techniques and tools used in Software & Systems-, AI- and IoT-Engineering and their application with cognitive cyber-physical systems.

fortiss is legally structured as a non-profit limited liability company (GmbH). The shareholders are the Free State of Bavaria (majority shareholder) and the Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.

fortiss GmbH
Guerickestraße 25
80805 München
Deutschland
www.fortiss.org
Tel: +49 89 3603522 0
E-Mail: info@fortiss.org

fortiss