White Paper

Blockchain is dead, long live Blockchain

An Engineering Approach for the Application of Distributed Ledger Technology

fortiss

Authors:

Tomas Bueno Momčilović
fortiss GmbH
Munich, Germany

Matthias Buchinger
fortiss GmbH
Munich, Germany

Dian Balta
fortiss GmbH
Munich, Germany

kontakt@fortiss.org

Content

Glossary

Abbreviation	Long version
IAM	Identity and access management
IBFT	Istanbul Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work

1. Introduction

An engineering approach for making evidence-based decisions on the value of using distributed ledger technology (DLT), choosing the most appropriate type, and designing a technology bundle for a specific case.

The successes of Bitcoin and Ethereum placed blockchain technologies on the radar of companies and governments worldwide. From the adoption of Bitcoin in El Salvador (Lopez & Livni, 2021), to billions of estimated investments in blockchain solutions globally ($65 billion estimated for 2021; PWC, 2020), this tool has become a go-to example of an innovation that can disrupt entire industries with its appropriate application. Nonetheless, unrealized investments followed the high expectations, inducing a wave of expressed disappointment in the actualized potential after more than 10 years of prototyping (Suichies, 2015).

As a subset of distributed ledger technology (DLT), the public blockchain with a cumbersome proof-of-work mechanism for determining consensus is only one example of a decentralized ledger. Private and hybrid solutions with unchangeable (i.e., immutable) entries, as well as cheaper and faster consensus mechanisms than proof-of-work, emerged in new applications in the past decade. To bring attention to these emerging options, more than 100 decision trees and questionnaires have attempted to help users find their way through the preconditions and potential categories of DLT, yet none with a holistic, design-oriented approach.

This white paper introduces a comprehensive framework for prototyping DLT applications, which is also available as an online tool. Starting from the important question on "should I use DLT?" can help users avoid risky trial-and-error in creating expensive prototypes whose value may never be actualized. Upon completion of the questionnaire, users obtain high-level recommendations, traceable requirements, and a canvas of DLT and non-DLT components for a complete application. As the only framework to merge all preconditions and possibilities into one location, the proposed tool fosters early collaboration between managers and engineers, thus leading to provably useful applications.

2. What is DLT?

Distributed ledger technology (DLT) applies cryptographic principles to economic needs. It is a new paradigm for storing transactions on a commonly accessible ledger, wherein participants who transact with each other do not need to rely on centralized control for trust and security. The most prominent DLT is the public blockchain, which represents the first generation of solutions that have since been improved.

The technology represents a whole category of solutions that go beyond the original Bitcoin white paper (Nakamoto, 2008) in at least three manners. Firstly, newer and more sustainable protocols[1] make the process of transacting and verifying trustless, cumbersome and energy-expensive than the original Proof-of-Work protocol. Secondly, emerging solutions rely on a combination of methods for linking transactions (e.g., directed acyclic graphs), not relying purely on one way of chaining transaction blocks. Finally, there are now many ways to compartmentalize who can write or read transactions, providing ways for transaction data to be decentralized but remain private.

The classic blockchain provided a technically demanding method for coordinating relatively uncomplicated and common currency exchange, but opened a large room for innovation. From electricity to supply chains, modern DLT solutions can substitute complex mechanisms of exchanging value in dynamic networks, or separate a centralized system of control into distributed compartments. Achieving both by transforming systems into ecosystems with human and automated participants is the future.

Core Properties

Distributed ledgers are immutable, trustless, decentralized, multiparticipant ecosystems, fundamentally differing from centralized ledgers and other database solutions.

- **Immutable:** Transactions written in the distributed ledger cannot be subsequently changed. This is useful for recording financial and non-financial transactions, since the ledger records the mistakes, misuse (e.g., double spending) or breaches of contract that lead to harms.
- **Trustless:** The ledger shifts the trust from people to software. By verifying the code, individual participants should be able to trust that the software will function as intended, so that they can safely interact with other participants without knowing their intentions.
- **Decentralized:** The structure of participants in the ledger is horizontal (i.e., democratic and consensus-based) rather than vertical (i.e., hierarchical and control-based). Even when access to the ledger is restricted for privacy or other reasons, the limited control any party can exert minimizes the risk of a single point of failure. This is particularly appropriate for consortia and other cooperating actors who are not under the same roof.
- **Multiparticipant:** To be truly immutable and distributed, ledgers must be used by multiple participants. However, the security of the ledger is not threatened if the interests of the participants are misaligned, so this solution is apt for cases where trust cannot be easily established, but everyone benefits from a common solution.
- **Ecosystem:** Modern DLT solutions include complementary components[2], and are embedded in an ecosystem of connectable DLT tools (i.e., side-chains) and non-DLT extensions (i.e., off-chains), fixing throughput, storage, and functionality constraints of the ledgers.

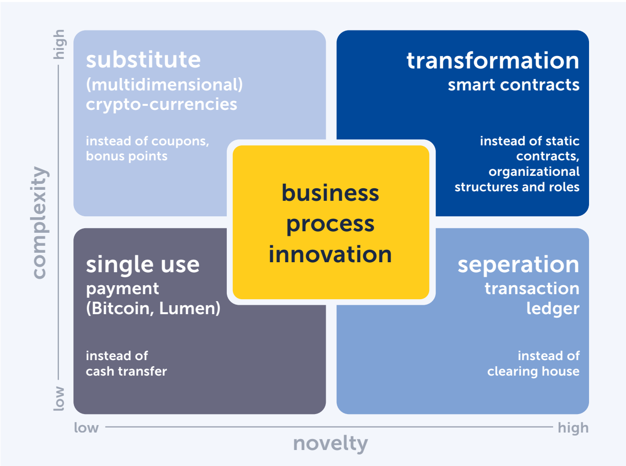


Figure 1: The value of DLT process innovation

1 Including, but not limited to: Proof-of-Stake, Proof-of-Authority, Proof-of-History, Proof-of-Elapsed-Time, and Practical Byzantine Fault Tolerance.
2 Examples include: token wallets, smart contracts, peer-to-peer file storage solutions, or Internet-of-Things (IoT) sensors and actuators.

2. What is DLT?

Framework for DLT Utilization

This paper proposes a framework for determining the usability of DLT in practical cases beyond cryptocurrencies. Using three pieces of the DLT puzzle - prototypes, decision-making frameworks, and technical literature – we provide a tool for designing a prototype from an idea in 86 questions. Although the framework requires managers and engineers to commit substantial effort initially, it reduces the total time and effort that users would need to prepare the documentation and implement the idea otherwise.

Fundamentally, the tool helps users achieve three goals simultaneously:

1. prove beyond reasonable doubt that the DLT is a valuable technology for a given use case, in order to obviate the need for expensive trial-and-error and reduce the risk of project failure or underperformance compared to the conventional alternatives;

2. align the needs of internal teams and external stakeholders with the technical possibilities, by setting realistic expectations and explicit requirements, to avoid painful transitions; and

3. generate a blueprint containing all layers, customized component recommendations, and a complete architecture with DLT and non-DLT elements, which can be immediately implemented using traditional software engineering methods.

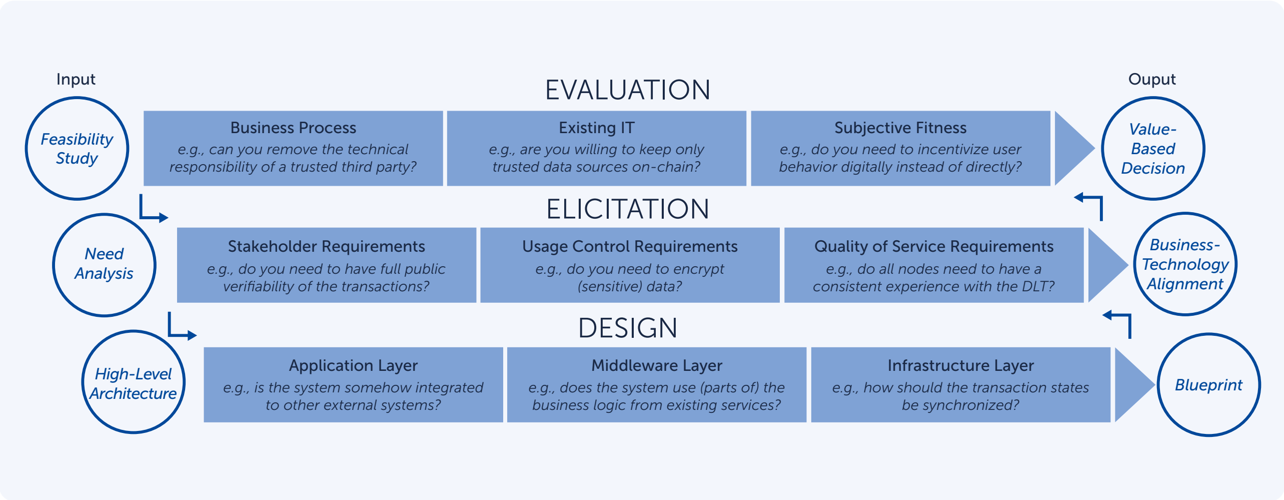


Figure 2: Three-stage framework for DLT prototyping

3. Stage I: Evaluation

The goal of the first stage is to satisfy all conditions related to hard technical and organizational constraints introduced by DLT and understand the risks under the soft constraints. The topic of **business process** is the important first step that users must take to understand the role of DLT underneath the hype. Unlike storing transactions in a database or coordinating activities via emails, the implementation of this solution requires users to undertake management changes toward rather unintuitive behaviors.

Instead of transitioning toward trusting well-defined systems and their codes (also known as "trustlessness"), one example of an unintuitive change is the reduction of the technical (and possibly, entire) responsibility of a trusted third party or intermediary. Another is the introduction of transaction fees to cover the additional computing power, which can unintentionally disincentivize the intended use of the ledger. In addition to these changes, users must have a proper and legal reason for using DLT over other alternatives and possess something (i.e., contracts, assets, or exchange) that can be represented digitally.

Particularly, two types of tools used to analyze the business process and the involved stakeholders have been proven helpful in the past. The first type of tool is the stakeholder dependency diagram. **Figure 3** shows a diagram, which depicts all involved stakeholders with their activities in the process and their goals. Furthermore, the diagram reveals the dependencies between the actors, the power distribution, and the available governance mechanisms.

Multiple questions for the business process topic, such as the absence or presence of a trustworthy third party in the process, can be answered with the development of the aforementioned diagram. An example of the stakeholder diagram is shown in the appendix.

The second helpful tool for the evaluation of the business process is the use of a BPMN process flow diagram as shown in **Figure 4**. This process flow diagram is a formal method of diagramming an organizational process with its inputs and outputs as an interface for defining the important elements of the current system, problems, and potential solutions.

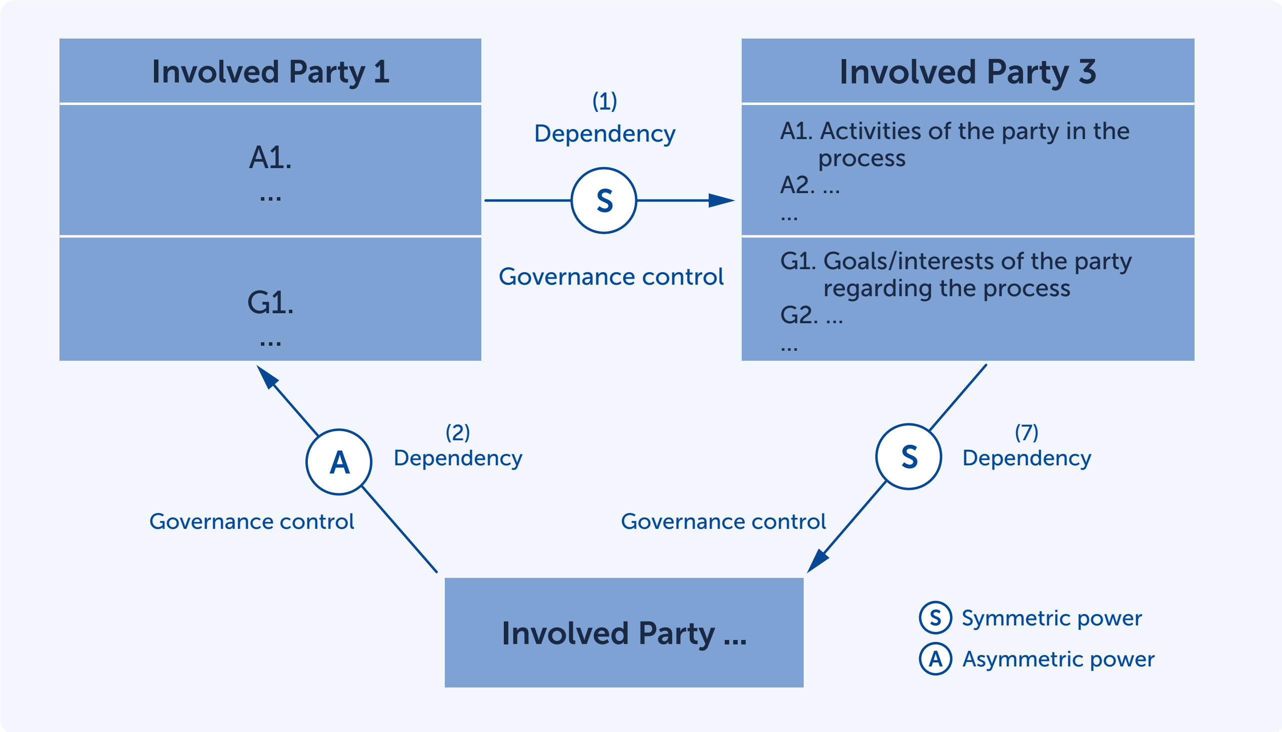


Figure 3: Notation for stakeholder dependency diagram

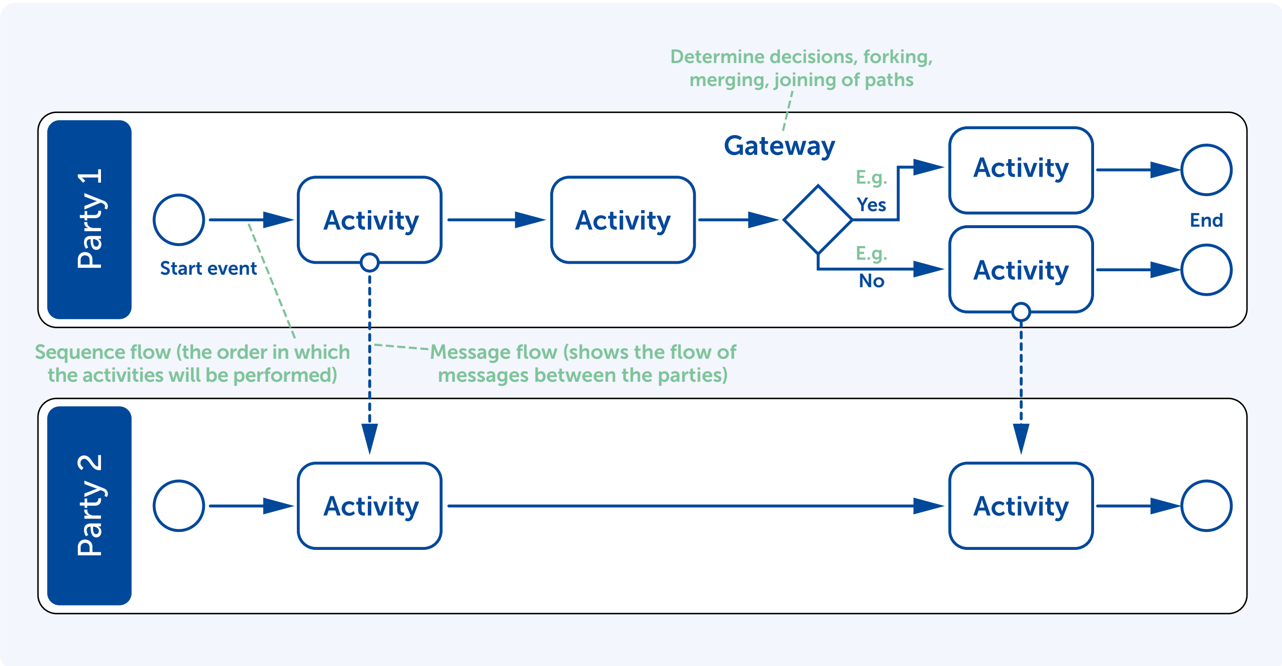


Figure 4: Notation for BPMN process flow diagram

3. Stage I: Evaluation

The development of this kind of diagram helps identify the current issues in the process, as shown in the example of a BPMN process flow diagram presented in the appendix. These issues can be further taken as starting points for the development of requirements and needs for a solution and the evaluation (if DLT is the right technology to solve these needs).

As a technically demanding solution, DLT can still be a poor choice for organizations under its current limitations. The topic of **existing IT** focuses on providing guidelines related to throughput, energy use, on-chain storage capacity data, and hardware requirements. DLT performs worse than its alternatives in almost all dimensions. Thus, the use case should neither strictly depend on speed (such as banking transactions) nor scale (e.g., big data), and the business value should justify adopting the low-performing solution. Additional technical properties of a use case that would go against DLT would include providing read and write access only to one entity, wherein regular databases perform the job, or giving only write access to untrustworthy sources without setting up verification mechanisms (for the discussion of why DLT cannot perform the job of IoT sensors, Wust & Gervais, 2017). Notably, because what is stored on the ledger cannot be changed, not all states or data (e.g., user data) are appropriate.

Subjective fitness is concerned with needs that are specific to a use case, such as the need to decentralize away from problematic or corrupt intermediaries, or secure oneself against attacks. While not the best solution, encryption and immutability of entries in DLT can provide security, privacy, and transparency that are necessary for the work of governmental institutions or the protection of intellectual property in businesses. Answering the questions negatively does not disqualify a case, but users must be aware of the risks and costs of using expensive consensus mechanisms to achieve simple objectives. Similarly, automatically executable smart contracts are useful for cases where arduous contract verification is unnecessary, or alternatively may require investment into verification mechanisms, such as sensors, optical character recognition, and an interface for uploading and storing supporting documents.

Examples of added value provided by DLT are summarized in Figure 5.

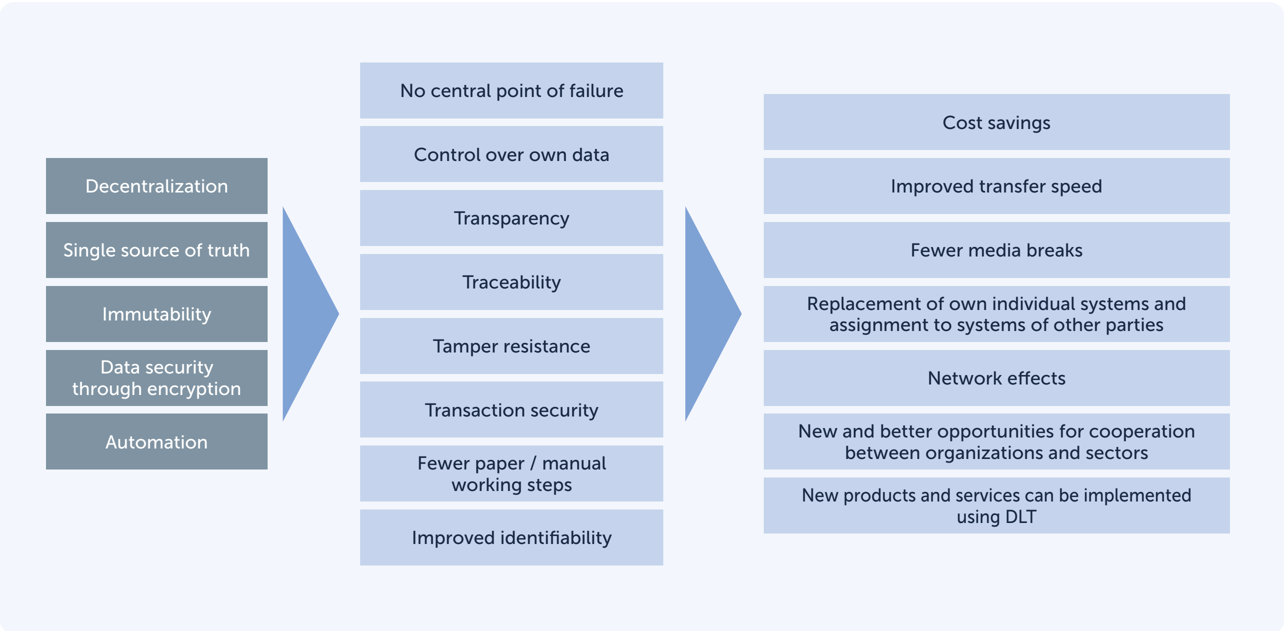


Figure 5: Examples of provided added value by DLT

4. Stage II: Elicitation

Elicitation translates the needs into technical requirements. Given the changes triggered by implementing DLT, stakeholder requirements are the most substantial. They range from a simple definition of participants to making their needs and incentives explicit. For example, private organizations emphasize non-disclosure of business secrets, whereas public or civic organizations may need full public verifiability of transactions. Different incentives also require different reward and punishment mechanisms to constrain behavior. Cryptocurrencies use coins to reward the miners, but organizations can also rely on tokens for direct monetary or in-kind reward. As for punishment, high fees in certain transactions can reduce unwanted activity.

Usage control and quality of service (QoS) requirements are not extensive, but require direct collaboration of managers and engineers. Important technical trade-offs are considered when deciding on substantially high-level issues. Contrary to the popular belief, nodes in current-generation blockchain and other mechanisms need not be fully transparent or equal. Within limits, users are given control over functionalities, so they can sacrifice e.g., open access for security, or full immutability to satisfy the "right-to-be-forgotten." As for QoS, scalability can be an issue. Users should have a plan for big data, whether in specific transactions or across all transactions over time. Similarly, security is also important, as participants can threaten the ledger (e.g., 51% attack). Finally, users should specify version compatibility and feasible performance needs.

The questions in the first and second stages of the framework depend on the domain knowledge of the users, who can generate the inputs using their preferred off-the-shelf tools. Multiple tools such as a Stakeholder Dependency Diagram can help (see Figure 6), as demonstrated in Balta et al. (2019). They highlight what is important to the organization and its stakeholders, preventing the users from being overly attached to a specific solution (i.e., "law of the instrument").

Depending on domain knowledge, such tools can be used by senior managers who comprehensively understand their organization and stakeholders, as well as specialized users (such as systems engineers) who can interview relevant actors and find documentation, such as contracts, which support the created view. In any case, easily retrievable and documented input from a comprehensive feasibility study would provide traceable answers to questions that directly concern values and requirements, thereby reducing the risk of reaching the prototyping design stage with a poorly fitting project.

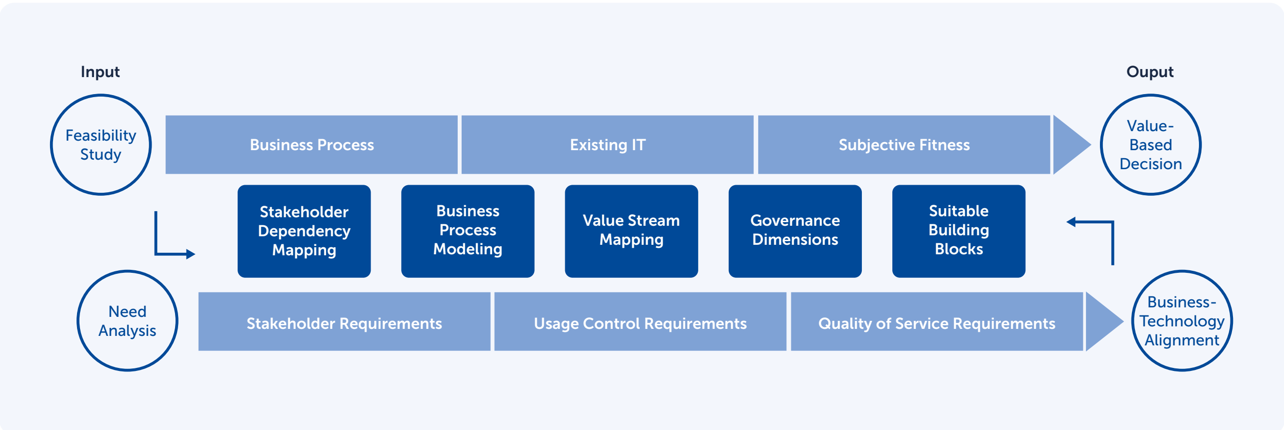


Figure 6: Summary of potential tools for answering the topics in the first and second stages

5. Stage III: Design

The findings from the first two stages serve as input for the third stage, namely the design stage, as shown in Figure 7. This stage leads to the following: the realization in the evaluation stage, in which substantial amounts of data must be stored, and the recommendation of an additional off-chain database component at the design stage because storing big data on a DLT is usually inefficient and costly. The colored arrows in **Figure 7** show further examples of interdependencies and information flow in the tool. The information flow through the tool results in more effective results being obtained in the design stage when the process is more deeply analyzed in in the first stage.

The design stage comprises three layers that must be addressed by the engineers. The application layer covers typical questions in software application, including the interaction between the DLT system and its users, devices, and web services (e.g., through APIs), as well as the integration of components and external systems, such as IoT devices, wallets, and central administration. This layer also involves classical system administration issues, such as monitoring of performance and logging.

The middleware layer is concerned with programmable logic that can be implemented on the DLT through smart contracts or on other components.

The infrastructure layer covers all relevant questions for determining the storage of data: which data should be stored on- and off-chain, how sensitive data are treated, how many transactions per month and data per transaction are processed, and other similar questions.

The 4+1 view model (Figure 8) can be used to depict the results of the third stage in an understandable and common way. The 4+1 view model is used to describe the architecture of a software-intensive system from the perspective of different stakeholders, such as end users, developers, or project managers. The four views of the model include logical, development, process, and physical views, and the "+1" is a scenario view. Different types of UML diagrams can be used to depict the four main views.

- The logical view deals with the functionality of the system for the end user and can be depicted with a UML class diagram.
- The development view describes the system from the viewpoint of a developer and handles software management. Therefore, a UML component diagram is suitable.
- The process view depicts the processes between the components of the system and is also important to understand the runtime behavior. A UML sequence diagram can be used for this purpose.
- The physical view illustrates the system from the viewpoint of the system architect. It also deals with the distribution of software components and is closely related to the development view. Therefore, the suitable UML deployment diagram is based on the UML class diagram.
- The scenarios are mostly represented through written text and describe application scenarios.

Examples of the diagrams for all five views are provided in the appendix.

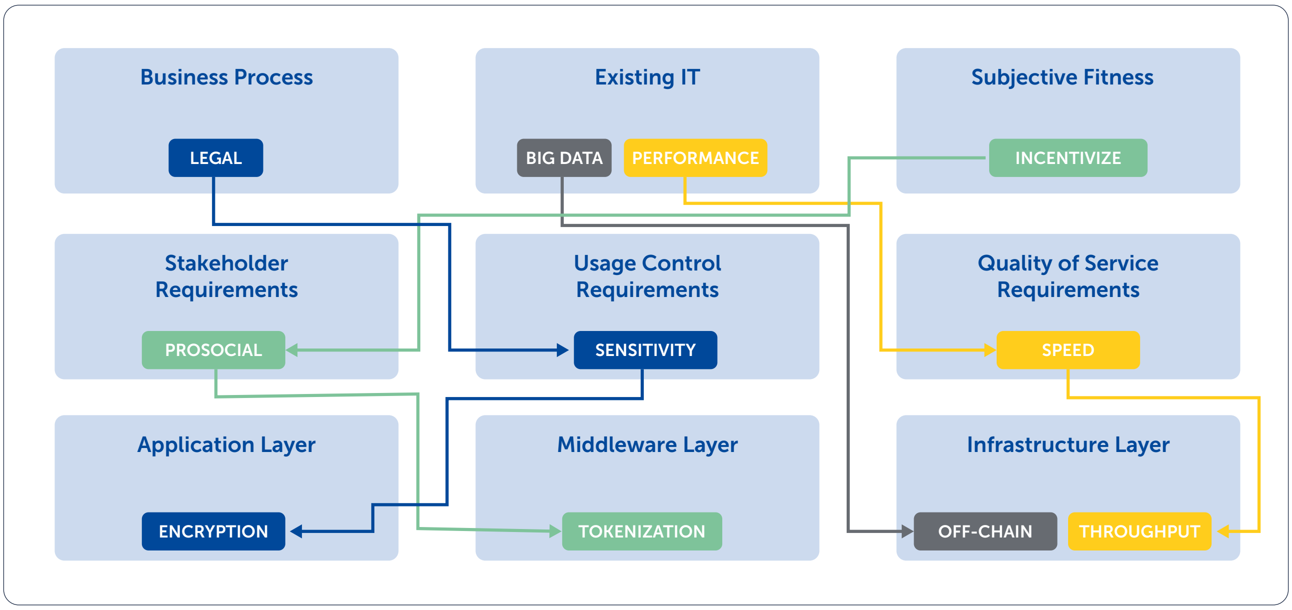


Figure 7: Information flow through the tool

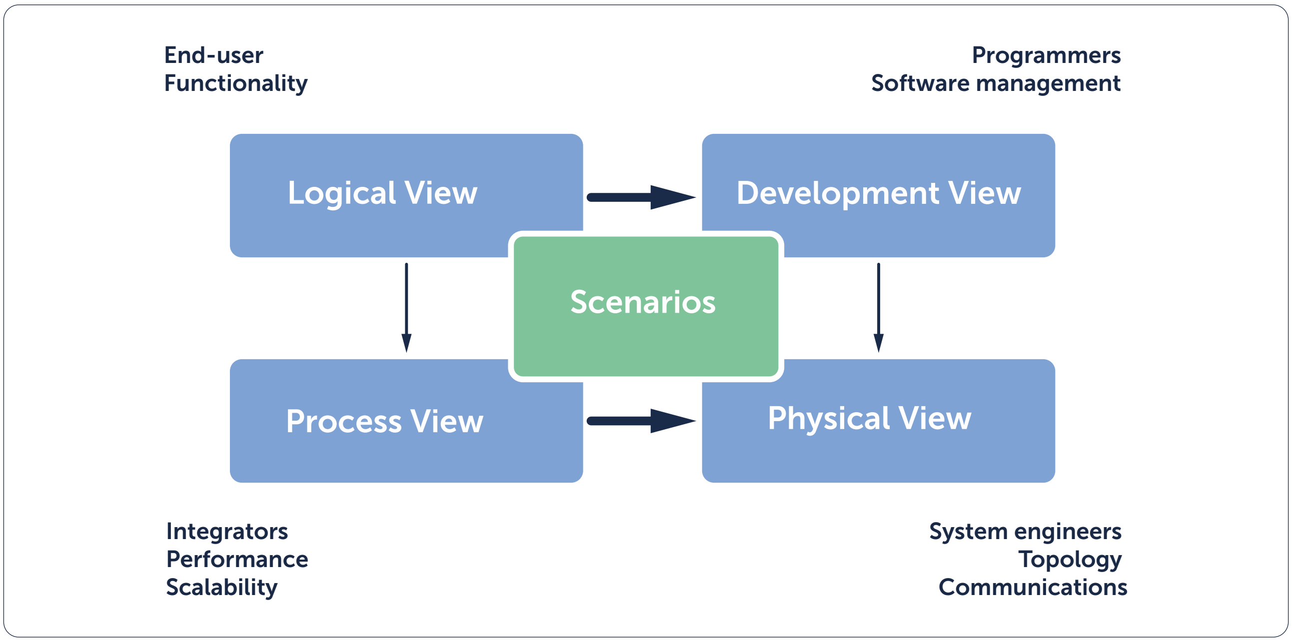


Figure 8: „4+1" view model according to Kruchten (1995)

6. DLT Use-Case-Solver

The DLT Use-Case-Solver is a tool developed to simplify and standardize the conclusions from the analysis. This tool is based on the three stages presented in **Figure 2** and combines the results of the analyses of case studies, existing decision frameworks, and technical papers. The tool is built similarly to a questionnaire, and questions relevant to each stage are asked one after the other. By linking the questions in the background, the questions regarding the second and third stages can change depending on what was answered in the first stage. Moreover, the tool does not replace the work necessary to analyze a case; therefore, the above-explained tools, such as the stakeholder dependency diagram, should be performed before using the DLT Use-Case-Solver tool.

However, this tool reveals that each step in the analysis is important for decision making. The DLT Use-Case-Solver also specifies the impact of the analysis results on the decision regarding the use of DLT and the architecture for a DLT-based solution. Therefore, this tool can help decision makers in answering the following three questions at the beginning of a potential DLT project:

1. Should I use DLT in my project in the first place?
2. Which DLT type is the best fit for my project?
3. What would the software architecture for my DLT-based project look like?

The answers to these questions do not only evaluate whether a DLT is useful (which is the only property of other analyzed decision frameworks), but also provides substantially deep analyses of the added value of DLT in the specific use case and its reasonable implementation together with other necessary components.

This canvas helps standardize and accelerate the development process and increases the likelihood of success of a DLT project. Moreover, architectures of existing DLT projects can be reevaluated based on the best practice results of the tool.

As a publicly available web app available on any browser, the tool is accessible to everybody and easy to use without further instructions: https://dlt.fortiss-demo.org/.

The tool was developed to be used by managers and technical engineers. However, a solid understanding of the business processes, the business environment, and boundaries is necessary to answer all questions correctly, especially for the stages of DLT evaluation and elicitation. Therefore, the aforementioned tools help perform the analysis comprehensively. Deep technical knowledge, specifically regarding technical requirements and the current IT landscape, is also crucial for the stage of DLT design.

As a result of the tool, if DLT is assessed as suitable for the analyzed case. Then, a basic software architecture for a DLT-based system is derived and presented as a canvas with three main stages. The canvas can be used to develop the different views of the 4+1 view model because it provides the most information needed for the different views.

- In the **application layer**, specific requirements regarding user interfaces and user management, as well as logging and monitoring aspects, are provided.
- In the **middleware layer**, backend components, as well as the business logic and data model, are analyzed from a concept perspective.
- The **infrastructure layer** is subdivided into the network, processing, and storage layers. Relevant aspects to diverse networks are analyzed in the network layer.

An example for a filled output canvas of the DLT Use-Case-Solver can be found in the appendix.

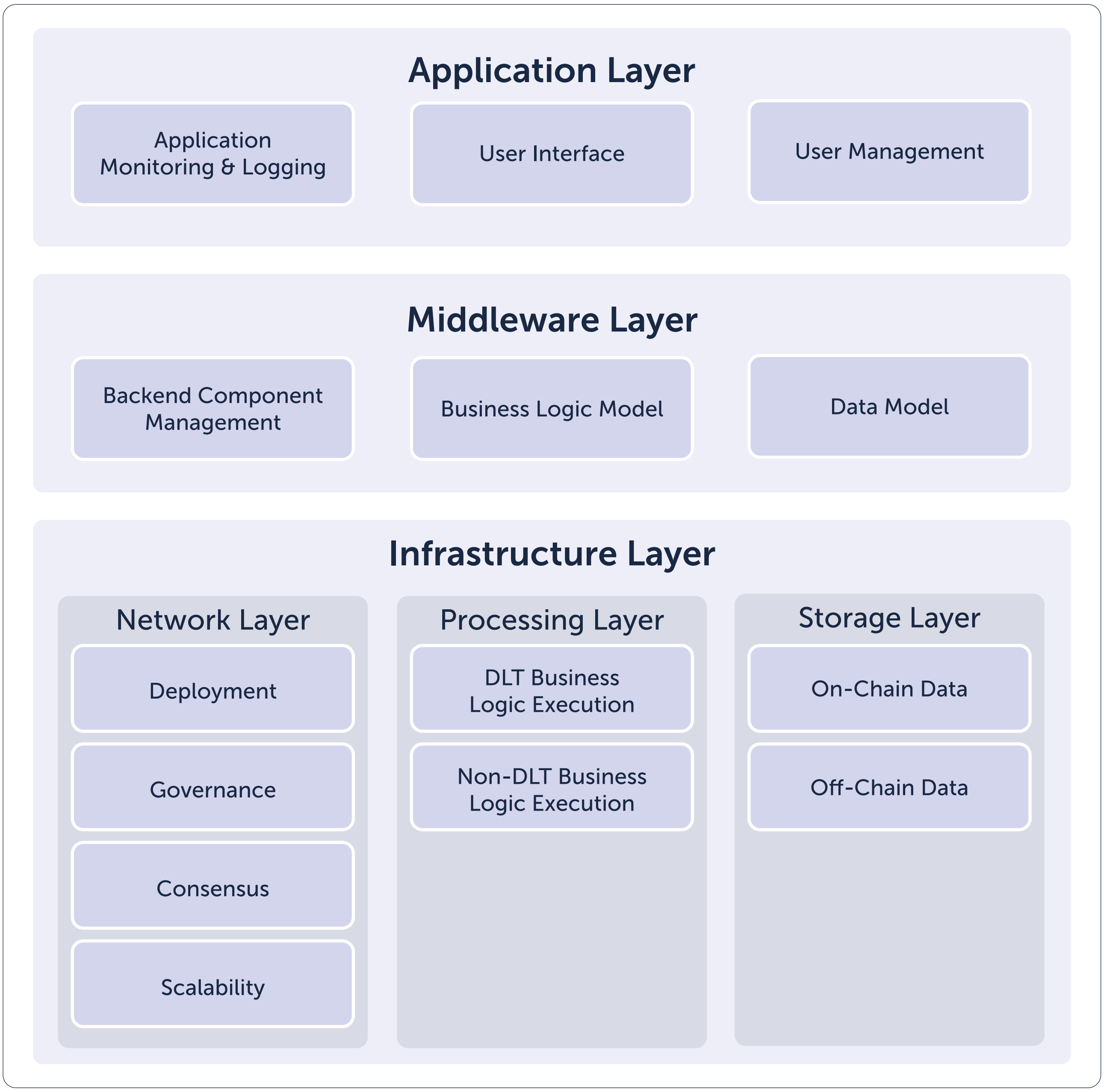


Figure 9: Output canvas of the DLT Use-Case-Solver

7. Application in Practice

The developed tool streamlines the analysis of possible DLT use cases, especially the architecture design process for promising cases. However, one major characteristic of DLT is its applicability only in an environment with numerous stakeholders. The evaluation process (e.g., with the stakeholder diagram) attempts to capture the needs and desires of all stakeholders to incentivize them to participate in the DLT solution, but the evaluation performed by one stakeholder is still mostly biased. Therefore, the opinions of all required stakeholders must be obtained to increase the chance of success in real projects. The tool must be processed by various stakeholders and the results must be compared and synchronized. In reality, the inclusion of all stakeholders can only be ensured by an agile, cyclical evaluation process. As illustrated in **Figure 10,** an agile process may lead users to rethink their approach as they strive to align business and technology or create a design blueprint and prototype. Similarly, the same stages can be used to further specify needs or align with other stakeholders, especially when the DLT implementation would introduce potentially contested change.

The analyses of existing DLT decision frameworks revealed that a framework recommending specific software architectures does not really exist. Design recommendations were either merely evaluations of performance, implementations of single DLT use cases, or suggested reference architectures. Thus, the proposed tool is quite novel and still needs verification and evaluation with new use cases or by practitioners. Furthermore, the cases, which served as the main source for the architecture design part, were all under proof-of-concept states and had various limitations that led to quite unrealistic work in production scenarios where other conditions apply and some requirements change. The tool also applies mainly to the recommendation of software architecture for proofs-of-concept or minimum viable products.

Finally, regarding separate branches for each industry, the tool thus far is an industry-agnostic version that neglects specific industry requirements or scenarios. Instantiation of the tool in practice may increase its applicability, so this step will be pursued in the future. Moreover, the technologies in the output canvas are recommended on a component level, because multiple options are available based on preference. Consequently, specific advantages and disadvantages of any single technology are not declared because these areas evolve rapidly and information risks becomes outdated. Still, we strive to provide examples of fitting technologies for each component on the tool website: https://dlt.fortiss-demo.org/dlt-analysis.

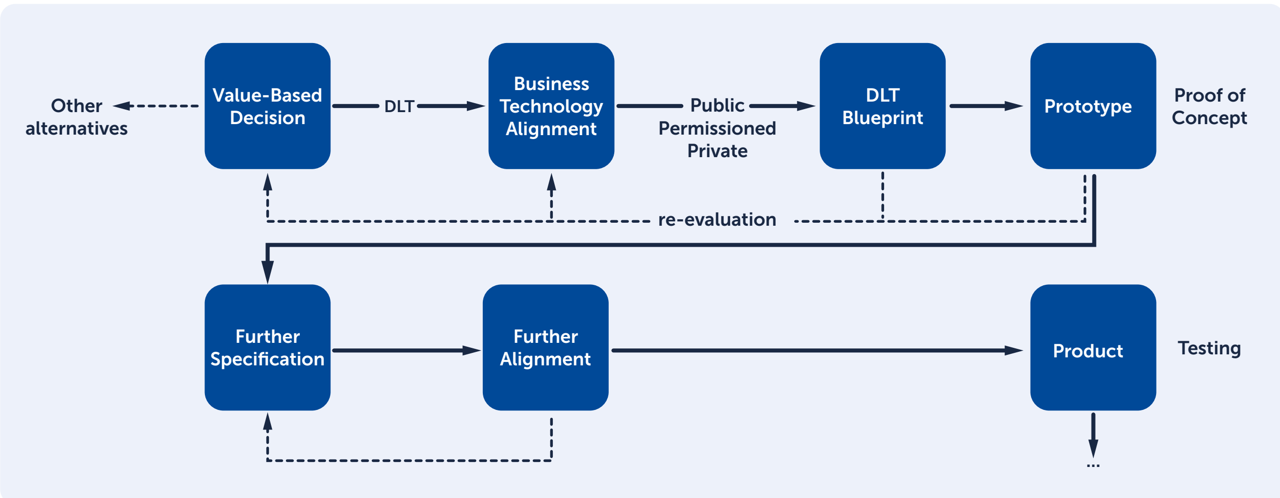


Figure 10: Expected outputs of each stage and the potential for iteration

8. Acknowledgements

We are grateful for the contribution of all our partners with their use cases along the practical course at TU Munich. We are also grateful for the contribution of all the student participants in the course, who analyzed the uses cases and developed prototypes.

We would also like to thank all member of our research group for their support in terms of reviewing and providing feedback on the numerous versions of this paper. Additionally, we would like to thank former student and research members of our team, who contributed substantially to the course and our DLT research: Vincent Delitz, Anastasios Kalogeropoulos, Saurabh Singh, and Marcus M. Dapp.

Finally, our special thanks go to Professor Helmut Krcmar who enabled us to organize the course at the TU Munich, supported our efforts at fortiss and served as inspiration to pursuing the goal of designing information systems in a distributed setup.

9. References

Abdo, J. B. & Zeadally, S. (2021). Neural network-based blockchain decision scheme. Information Security Journal: A Global Perspective, 30(3), 173–187. DOI: 10.1080/19393555.2020.1831658

Ammous, S. (2016). Blockchain Technology: What is it Good for? SSRN, published online. DOI: 10.2139/ssrn.2832751. Retrieved from https://ssrn.com/abstract=2832751

Balta, D., Kalogeropoulos, A., Kuhn, P. & Krcmar, H. (2019). In: Search For Consensus. In: Shefali V. (ed), Proceedings of Ongoing Proceedings of Ongoing Research, Practitioners, Posters, Workshops, and Projects of the International Conference EGOV-CeDEM-ePart 2019.

Balta, D., Greger, V., Wolf, P., & Krcmar, H. (2015). E-government stakeholder analysis and management based on stakeholder interactions and resource dependencies. 2015 48th Hawaii International Conference on System Sciences, 2456–2465. IEEE.

Belotti, M., Božić, N., Pujolle, G. & Secci, S. (2019). A vademecum on blockchain technologies: When, which, and how. IEEE Communications Surveys &Tutorials, 21(4), 3796–3838. DOI: 10.1109/COMST.2019.2928178

Birch, D., Brown, R. G. & Parulava, S. (2016). Towards ambient accountability in financial services: Shared ledgers, translucent transactions and the technological legacy of the great financial crisis. Journal of Payments Strategy & Systems, 10(2), 118.

Buchinger, M., Balta, D. & Krcmar, H. (2020). Distributed Ledger Technology in the Banking Sector: A Method for the Evaluation of Use Cases. 15th International Conference on Wirtschaftsinformatik. DOI: 10.30844/wi_2020_b6-buchinger

Buterin, V. (2013). Ethereum white paper. Ethereum. Retrieved from https://ethereum.org/en/whitepaper/

CompTIA (2019). Blockchain Decision Tree. CompTIA. Retrieved from https://connect.comptia.org/content/infographic/blockchain-decision-tree

Degenhart, B., Zachau, S., Joseph, S., Kriti, S. & Kobinger, L. (2019). FINFOO [Git Repository]. 2019 DLT4PI Seminar, Technical University of Munich. Retrieved on 10th November 2021 from https://git.fortiss.org/Blockchain/student-practical-courses/dlt4pi-ss19/fin4

Gartner. (2019). Gartner 2019 Hype Cycle for Blockchain Business Shows Blockchain Will Have a Transformational Impact across Industries in Five to 10 Years. Gartner. Retrieved from https://gartner.com/en/newsroom/press-releases/2019-09-12-gartner-2019-hype-cycle-for-blockchain-business-shows

Gourisetti, S. N. G., Mylrea, M. & Patangia, H. (2019). Evaluation and demonstration of blockchain applicability framework. IEEE Transactions on Engineering Management, 67(4), 1142–1156. DOI: 10.1109/TEM.2019.2928280

Gräbe, F., Kannengießer, N., Lins, S. & Sunyaev, A. (2020). Do not be fooled: Toward a holistic comparison of distributed ledger technology designs. 53rd Annual Hawaii International Conference on System Sciences. DOI: 10.24251/HICSS.2020.770

Habes, A., Alhomsi, B., Lurch, F. & Adikari, M. (2019). A prototype of an anti-counterfeiting blockchain for IBO GmbH [Git Repository]. 2019 DLT4PI Seminar, Technical University of Munich. Retrieved from https://git.fortiss.org/Blockchain/student-practical-courses/dlt4pi-ss19/ibo

Hribernik, M., Zero, K., Kummer, S. & Herold, D. M. (2020). City logistics: Towards a blockchain decision framework for collaborative parcel deliveries in micro-hubs. Transportation Research Interdisciplinary Perspectives, 8, 1–8. DOI: 10.1016/j.trip.2020.100274

Hunhevicz, J. & Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. Advanced Engineering Informatics, 45. DOI: 10.1016/j.aei.2020.101094

Koens, T. & Poll, E. (2018). What blockchain alternative do you need? In J. Garcia-Alfaro et al. (eds.), Data Privacy Management, Cryptocurriencies and Blockchain Technology, 113–129. DOI: 10.1007/978-3-030-00305-0_9

Kruchten, P. B. (1995). "The 4+ 1 view model of architecture." In: IEEE software 12.6, pp. 42–50. issn: 0740–7459

Lapointe, C. & Fishbane, L. (2019). The blockchain ethical design framework. Innovations Technology Governance Globalization, 12(3–4), 50–71. DOI: 10.1162/inov_a_00275

Litan, A. (2021). Hype Cycle for Blockchain 2021; More Action than Hype. Gartner. Retrieved from https://blogs.gartner.com/avivah-litan/2021/07/14/hype-cycle-for-blockchain-2021-more-action-than-hype/

Lo, S. K., Xu, X., Chiam, Y. K. & Lu, Q. (2017). Evaluating suitability of applying blockchain. 2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS). DOI: 10.1109/ICECCS.2017.26

Lopez, O. & Livni, E. (2021 October 7). In Global First, El Salvador Adopts Bitcoin as Currency. The New York Times. Retrieved from https://www.nytimes.com/2021/09/07/world/americas/el-salvador-bitcoin.html

Luo, H. & Yan, D. (2021). Blockchain architecture and its applications in a bank risk mitigation framework. In: Economic Research-Ekonomska Istraživanja, S. 1–19. DOI: 10.1080/1331677X.2021.1986672.

Maden, A. & Alptekin, E. (2020). Evaluation of factors affectingthe decision to adopt blockchaintechnology: A logistics companycase study using Fuzzy DEMATEL. Journal of Intelligent & Fuzzy Systems, 39, 6279–6291. DOI: 10.3233/JIFS-189096

Marr, B. (2021, July 2). The 5 big problems with blockchain everyone should be aware of. Bernard Marr. https://bernardmarr.com/the-5-big-problems-with-blockchain-everyone-should-be-aware-of/

Maull, R., Godsiff, P., Mulligan, C., Brown, A. & Kewell, B. (2017). Distributed ledger technology: Applications and implications. Strategic Change, 26(5), 481–489. DOI: 10.1002/jsc.2148

Mendhurwar, S. & Mishra, R. (2021). 'Un'-blocking the industry 4.0 value chain with cyber-physical social thinking. Enterprise Information Systems, published online, 1–49. DOI: 10.1080/17517575.2021.1930189

Mulligan, C., Zhu Scott, J., Warren, S. & Rangaswami, J. P. (2018). Blockchain beyond the hype: A practical framework for business leaders. World Economic Forum. Retrieved from http://www3.weforum.org/docs/48423_Whether_Blockchain_WP.pdf

9. References

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved from https://bitcoin.org/bitcoin.pdf

Panetta, K. (2017). Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017. Gartner. Retrieved from https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017

Platt, M., Hasselgren, A., Roman-Belmonte, J. M. et al. (2021). Test, trace, and put on the Blockchain?: A viewpoint evaluating the use of decentralized systems for algorithmic contact tracing to combat a global pandemic. JMIR Public Health and Surveillance, 7(4), 1–14. DOI: 10.2196/26460

PWC. (2020). Time for trust: How blockchain will transform business and the economy. PWC. Retrieved from https://www.pwc.com/gx/en/industries/technology/publications/blockchain-report-transform-business-economy.html

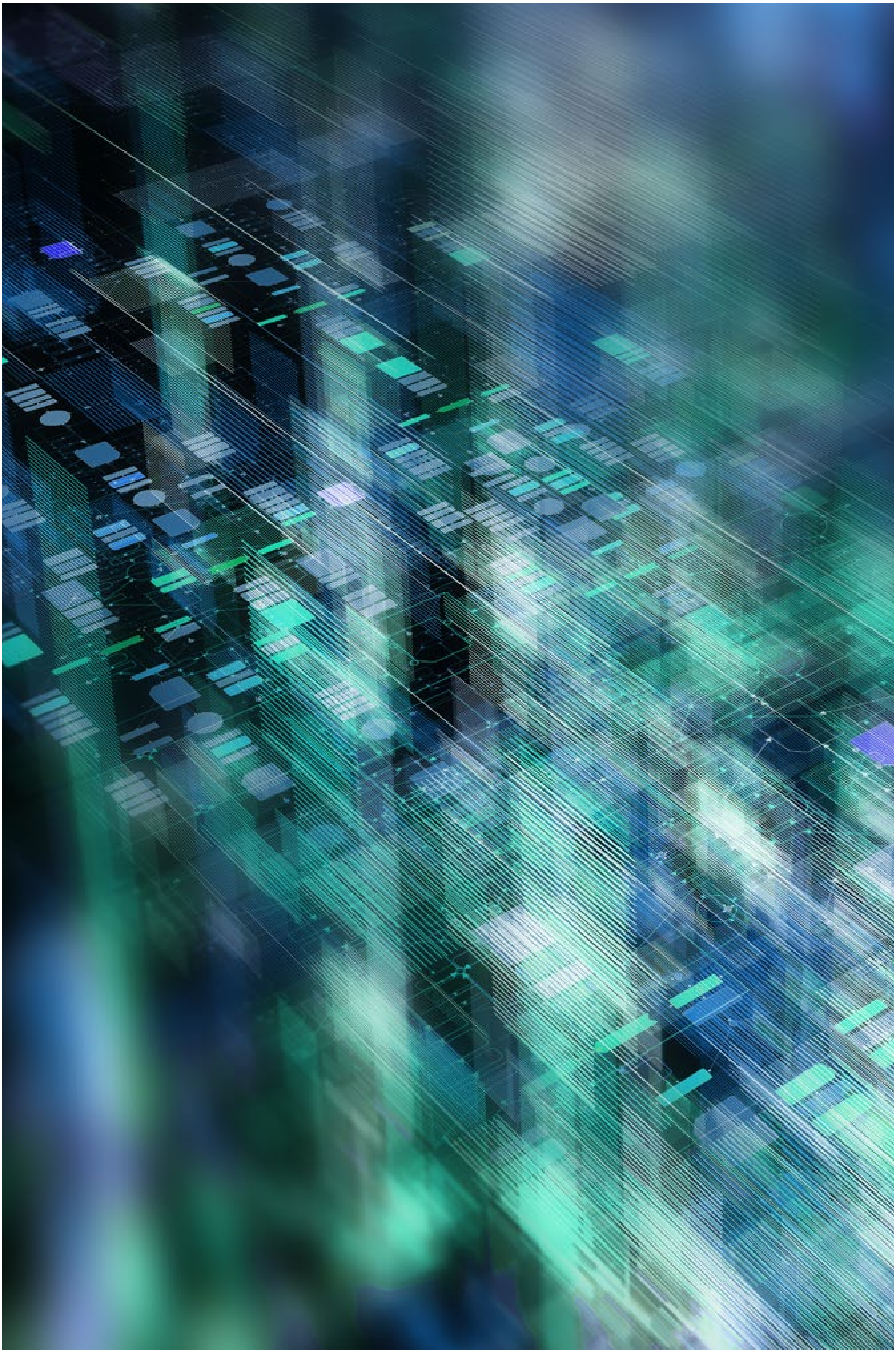
Suichies, B. (2015). Why Blockchain must die in 2016. LinkedIn Pulse. Retrieved from https://www.linkedin.com/pulse/why-blockchain-must-die-2016-bart-suichies/

Treiblmaier, H. (2019). Toward more rigorous blockchain research: Recommendations for writing blockchain case studies. Frontiers in Blockchain, 2(3), DOI: 10.3389/fbloc.2019.00003

Tsoniotis, N., Kourtesis, D., Tsiakmakis, S., Christaras, V., Makridis, M., Kounelis, I., & Fontaras, G. (2019). Feasibility study and prototyping of a blockchain-based transport-service pricing and allocation platform. Publications Office of the European Union. DOI: 10.2760/60436

Wüst, K. & Gervais, A. (2017). Do you need a blockchain? 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). DOI: 10.1109/CVCBT.2018.00011

Zhao, H., Zhang, M., Wang, S., Li, E., Guo, Z. & Sun, D. (2021): Security risk and response analysis of typical application architecture of information and communication blockchain. In: Neural Computing and Applications 33 (13), S. 7661–7671. DOI: 10.1007/s00521-020-05508-z



Apendix

A. Overview of the variety of use cases and the incentives for using DLT

One source of best practices in DLT is prototypes developed by organizations with strong incentives for successful DLT operation. The success of a project must be justified beyond the successes achieved by existing alternatives to avoid making DLT and blockchain only expensive prototypes. Two common questions in informal decision trees, such as "is the use case a real business problem?" and "can existing alternatives solve the problem?", respectively address the problems of testing and proving value beyond doubt.

In a student lab seminar named "DLT for Process Innovation" and organized by the research institute fortiss and the Department for Informatics from the Technical University of Munich, running yearly since 2017, teams of two to six graduate students have developed 24 DLT-focused prototypes for various cases. Herein, the various incentives, challenges, and properties of these projects, some of which are currently in use, were explored. Project repositories are available here: https://git.fortiss.org/Blockchain/student-practical-courses. Details about the use case can be found here: https://researchgate.net/project/DLT4PI

The cases were introduced and supported by companies and public entities. Initial correlations between some case characteristics and architecture decisions could be determined through a detailed analysis of all cases and prototypes, as shown in Figure 11.

As visible in the second column set, *trust* and *efficiency* were the most important incentives for prototyping DLT apps. Functionalities related to immutable storage, traceability of supply chains, and the use of financial or other incentives were primarily meant to build trust, whereas secure distribution, billing and effective coordination were ways of pursuing of efficiency. Mostly private and some hybrid DLTs were employed in order to provide functionalities to a limited number of (also mostly private) actors. Notable exceptions are two applications in the „special public interest" category, where private DLTs enable the coordination of energy trade between households and industrial actors without the extensive involvement of third parties, but where access to the application is controlled and permission granted upon request. General public applications relied on both public and private DLTs, with an emphasis on functionalities that digital incentives and storage provide for coordinating traffic, volunteering, real-estate trade, as well as supporting anti-corruption transparency-related projects.

		Secure information distribution & verification	Process & supply chain traceability	Billing	Financial incentive & Colored Coins	Effective coordination of multiple parties	Immutable, transparent information storing	Trust	Efficiency	Transparency	Control	Public DLT	Private DLT	Hybrid DLT	General public application	Cross-organizational business application	Public special application
Main functionality	Secure information distribution & verification	100%						56%	89%	22%	33%	33%	67%		22%	78%	
	Process & supply chain traceability		100%	13%				100%	75%	38%	13%		75%	25%		88%	13%
	Billing		50%	100%				50%	100%		50%		50%	50%		50%	50%
	Financial incentive & Colored Coins				100%			100%			100%	67%	33%		100%		
	Effective coordination of multiple parties					100%			100%	50%	100%	50%	50%		50%	50%	
	Immutable, transparent information storing						100%	100%		100%			100%		100%		
Incentive	Trust	29%	47%	6%	18%		6%	100%	59%	24%	35%	24%	65%	12%	35%	59%	6%
	Efficiency	47%	35%	12%		12%		59%	100%	24%	35%	24%	65%	12%	18%	71%	12%
	Transparency	29%	43%			14%	14%	57%	57%	100%	29%		100%		14%	86%	
		33%	11%	11%	33%	22%		67%	67%	22%	100%	33%	56%	11%	56%	44%	
DLT Type	Public DLT	50%			33%	17%		67%	67%		50%	100%			67%	33%	
	Private DLT	38%	38%	6%	6%	6%	6%	69%	69%	44%	31%		100%		19%	69%	13%
	Hybrid DLT		100%	50%				100%	100%		50%			100%		100%	
Scope	General public application	29%			43%	14%	14%	86%	43%	14%	71%	57%	43%		100%		
	Cross-organizational business application	47%	47%	7%		7%		67%	40%	40%	27%	13%	73%	13%		100%	
	Public special application		50%	50%				50%	100%				100%				100%

Figure 11: Summary of the dependencies of the different characteristics of the analyzed use cases

Figure 12 shows that the organizations which approached the students with a challenge came from all three (public, private, and academic) sectors; however, the most numerous (58%) were private sector projects.

Half of these projects centered on improving the manufacturing process using the immutable and decentralized nature of distributed ledgers to improve compliance, efficiency, and profit, with others from the pharmacological, food, shipping, and financial industry following similar incentives. Of the other non-private projects, many were driven not only by pro-social incentives for improving transparency, reducing corruption, and promoting positive behavior in communities but also enhancing societal efficiency in trading energy and real estate or managing traffic.

The main functionalities provided by the prototypes rely on the fundamental properties of distributed ledgers. For example, keeping certain data (e.g., transactions) on-chain is preferable. Decentralized ledgers allow users to trace and audit events in a process that may be subject to accidental or intentional changes for the worse, thereby making fault detection, root cause analysis, or supply chain trust-building easy.

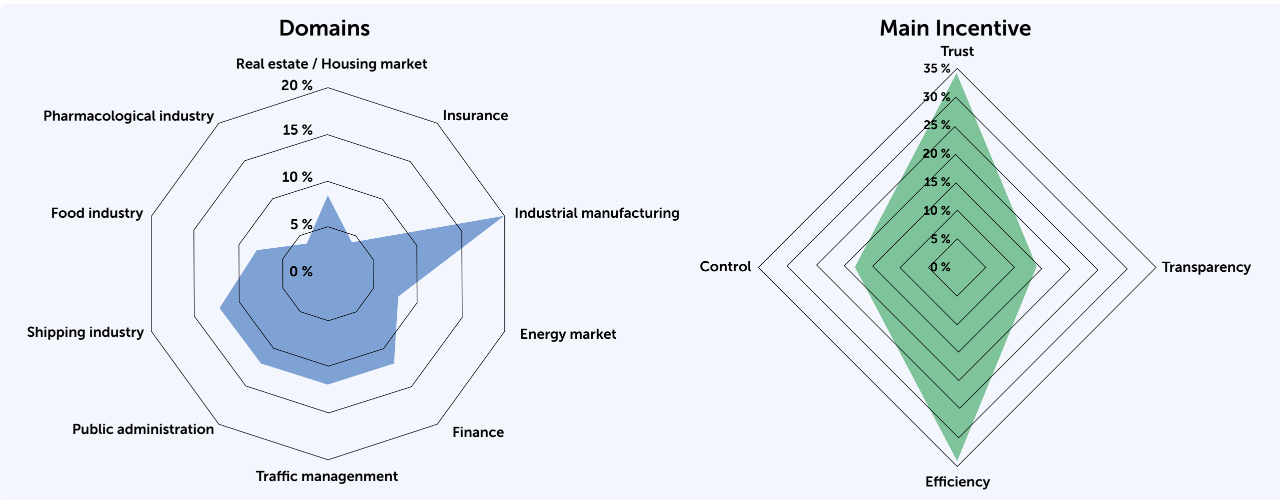


Figure 12: Domains of use cases and main incentives of the partners considering DLT application

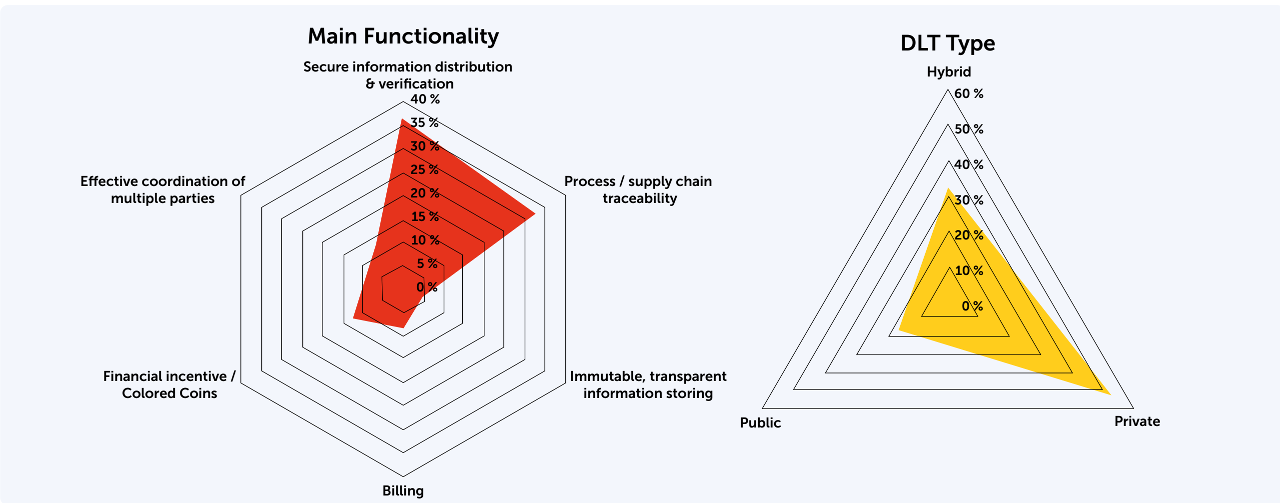


Figure 13: Main functionalities considered in the use cases and used DLT types to implement them

In high-stakes environments, such as airplane parts manufacturing, tracing the origins of subpar materials or parts is essential. Elsewhere, public verifiability of certain transactions is necessary for legal and social reasons (e.g., transparency); therefore, this information cannot be changed by any party. This condition is something that public or hybrid DLT solutions offer because these applications have been created from the start to be publicly, or at least widely, accessible.

Auditing is the most sought functionality. However, prototypes provided other functionalities to improve user behavior or stakeholder relationships with an entire ecosystem around DLT apps. A few recently implemented examples are as follows.

- **Tokenization:** DLT applications often come with the possibility to incentivize certain activities with tokens or coins or disincentivize them with transaction fees. In several prototyped apps, tokens mediate the trade of self-generated energy or real estate for money or volunteering for in-kind rewards without the need for a third party in most transactions[3].

- **Easy and secure setup:** As distributed (cloud or peer-to-peer) solutions, several DLT apps and their nodes are easy to set up and maintain by technical users and can be further simplified for non-technical users (e.g., the general public). In the case of decentralized clinical trials mediated by secure DLT apps, ease of use has proven to be essential to ensure the continuous interaction of patients with the app.

- **Encryption:** Most hybrid and private solutions provide encryption for sensitive data, thus creating a secure channel for communicating sensitive transactions. Despite limited throughput, this method has proven useful in cases where verification of patent provenance (without revealing the patent to everyone) or sharing personal health data is important.

- **File storage:** Given the storage limits of consensus mechanisms, DLT apps can be linked with a number of off-chain solutions for file storage. In examples that involved billing, files of invoices or photographic proof of delivery were stored off-chain, only linking to the unique identifiers of the slightly bulky financial information on-chain to reduce the cost of maintaining the ledgers.

- **Rule-based execution:** Smart contracts provide the ability to immediately execute an action (e.g., send payment) once certain conditions are satisfied (e.g., contract is fulfilled). In combination with AI or IoT solutions that serve as sensors for conditions beyond those which are traceable by code, this approach is quite useful. For example, one prototype proposes the use of smart contracts to reduce the human workload in reviewing supplied documents based on appropriate legal rules to accelerate the processing of immigration applications while also maintaining a decentralized trail to ensure the entire process is legally sound.

3 In cases of (legal) contention or technical issues, third parties often cannot be substituted, but their role can be reduced for the majority of non-contentious cases.

Domain	Secure infomration distribution & verification	Process & supply chain traceability	Billing	Financial incentive & Colored Coins	Effective coordination of multiple parties	Immutable, transparent information storing	Trust	Efficiency	Transparency	Control	Public DLT	Private DLT	Hybrid DLT	General public application	Cross-organizational business application	Public special application
Real estate / Housing market	50 %					50 %	100 %	50 %	50 %		50 %	50 %		100 %		
Insurance	100 %							100 %			100 %				100 %	
Industrial manufacturing		100 %					100 %	60 %	40 %			80 %	20 %		100 %	
Energy marked		50 %	50 %				50 %	100 %				100 %				100 %
Finance				100 %			100 %			100 %	67 %	33 %		100 %		
Traffic management	33 %				67 %		33 %	100 %		100 %	33 %	67 %		67 %	33 %	
Public administration	67 %					33 %	67 %	67 %	67 %	33 %	33 %	67 %		33 %	67 %	
Shipping industry	67 %	33 %	33 %				67 %	100 %		67 %		67 %	33 %		100 %	
Food industry	50 %	50 %					100 %	50 %	50 %			100 %			100 %	
Pharmacological industry	100 %							100 %	100 %			100 %			100 %	

Figure 14: Summary of the dependencies of the domain of the cases and different characteristics

B. Literature on DLT decision making

Given the potential gains yet practical limitations, additional discussion surrounding DLT has attracted hype and cynicism. Published research and expertise are crucial sources of knowledge behind decision-making, but individual views can nonetheless contradict each other. To understand the problem, more than 200 publications were studied, revealing 107 frameworks.

Most (61) were informal or open-ended, which often have no clear connection to actual use cases or sources and bring no original contribution except for the question, "Do you have a real business case?" (Lewis, 2016). Unfortunately, open-ended frameworks provide approaches that are too vague to help in deciding on how to proceed with DLT.

These frameworks rely on subjective decision making that requires either technical expertise in IT-oriented issues initially or acceptance of risk in testing out assumptions regarding the business. In areas where experimentation is costly[4] and decisions must be made by non-technical users, open-ended workflows do not provide a sufficient interface for reducing prototyping costs. The remaining (36) close-ended and peer-reviewed frameworks provide clear answers in the form of mostly decision trees or questionnaires, as shown in Figure 16.

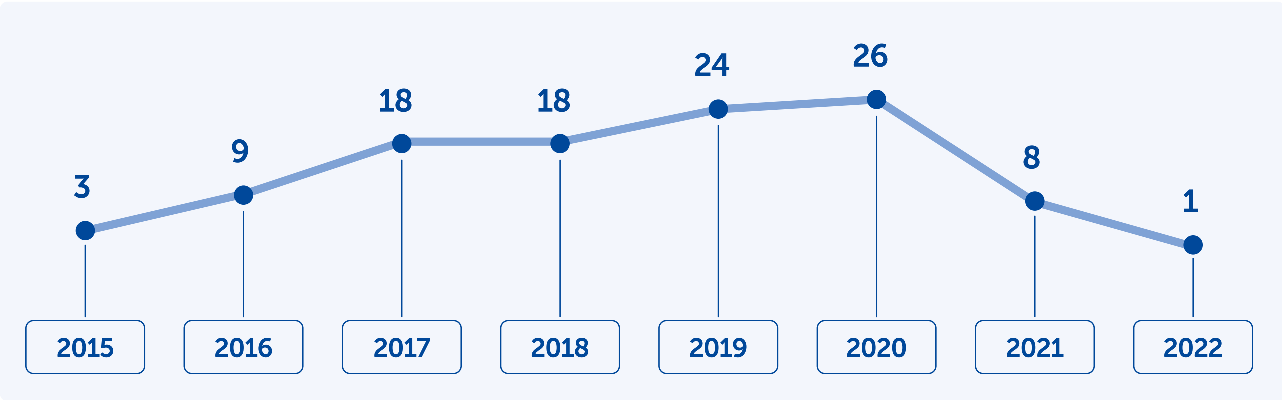


Figure 15: Decision frameworks by year

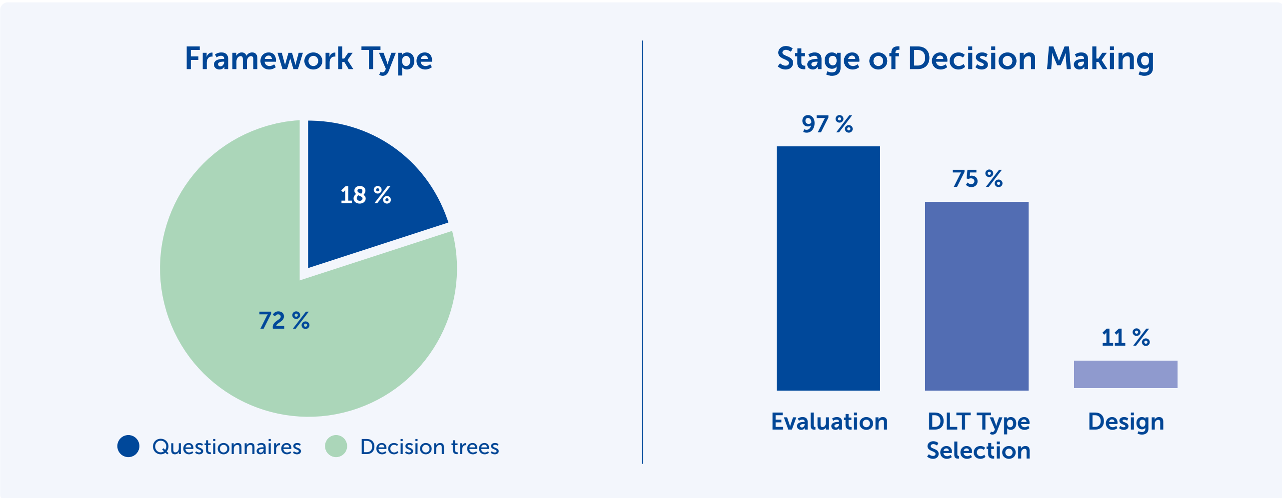


Figure 16: Types of decision frameworks and stages of decision making

4 Some tools allow for quick prototyping of experimental DLT apps. However, most require technical knowledge to use and commitment to maintenance.

Evaluation is the dominant category of issues addressed by the frameworks. DLTs are remarkably specific solutions for problems that cannot be solved (as effectively) by traditional databases or trusted third parties; thus, almost all frameworks disqualify DLT prototyping where this is not the case. Once the value of a DLT has been proven, most frameworks also guide users toward criteria for the **selection** between a public, hybrid, or private DLT solution. For example, a need for privacy, control, or public verifiability filters out what components will be available in the later stages. Moreover, except for high throughput, state-of-the-art, full public verifiability cannot be achieved with private DLT solutions. Finally, only a few publications deal with technical questions relevant to **design**. These publications center on smart contracts, throughput, or off-chain capabilities but do not offer a complete component recommendation. Therefore, the existing frameworks lead to 32 unique evaluation criteria and 27 further questions (**Figure 17**) for elicitation of various requirements. The listed categories overlap because users still have available options within the constrained set despite the hard constraints determined by the first stage.

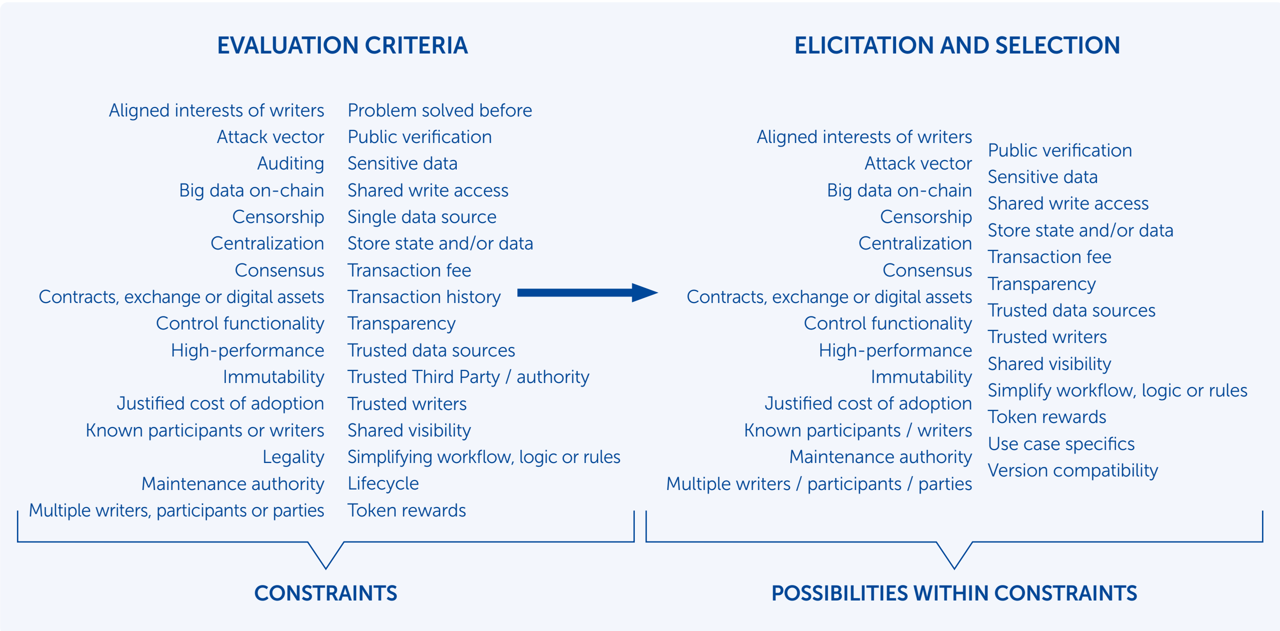


Figure 17: Evaluation criteria for requirement elicitation

C. Technical literature

Lacking design recommendations, the decision-making literature is most suited for specifying the details of the case itself within DLT-specific constraints. However, sufficient knowledge of the components behind a DLT application and its surrounding ecosystem is necessary to prototype a custom DLT solution considering case requirements. Technical papers on distributed ledgers (including blockchains) provide the necessary technical specifications in 27 unique questions, which include the following:

- **Technical DLT architecture and taxonomy:** Specific terminology and architecture blueprints for DLT-based systems, such as data management (on- and off-chain), frontend, and smart contracts. It also includes a first meta-architecture or layers.

- **Enterprise architecture use cases:** Use cases and their respective DLT architectures from applications in the industry or academic settings; similar to the student prototypes.

- **DLT design patterns:** Different approaches to on- and off-chain combinations, specific uses of smart contracts, or handling of key management, among other patterns.

- **Procedural guidance for DLT development:** Workflows that help in the design of individual layers as well as the selection of appropriate components.

- **Performance analysis:** Comparative analysis of the throughput and capacity of different solutions across the public–hybrid–private trichotomy, consensus mechanisms (e.g., Proof-of-Work, Proof-of-Stake, …), or types of distributed ledgers (e.g., blockchain, Hashgraph).

- **Decentralized software components:** Comprehensively cover decentralized components, and Interplanetary File System as the most prominent solution in most cases.

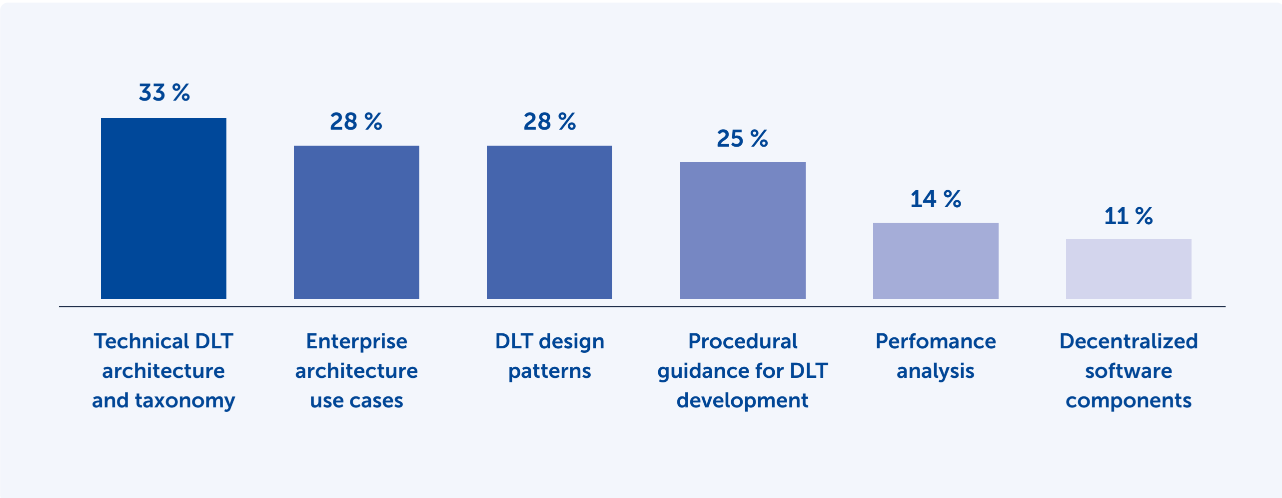


Figure 18: Technical specifications addressed in analyzed technical papers

D. Examples of helpful tools for use case analysis

Stakeholder dependency diagram

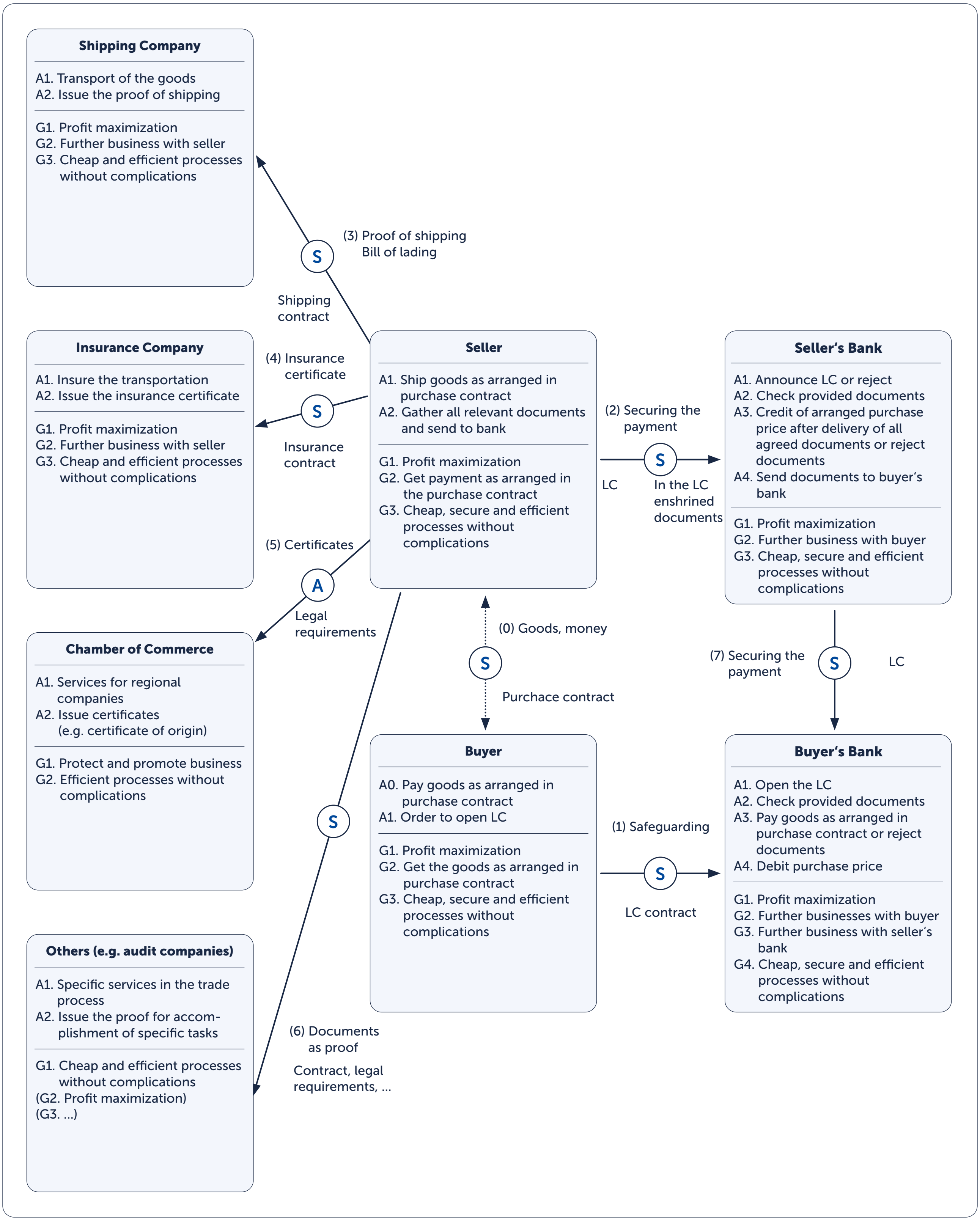


Figure 19: Example of a stakeholder dependency diagram from a letter of credit case

BPMN process flow diagram

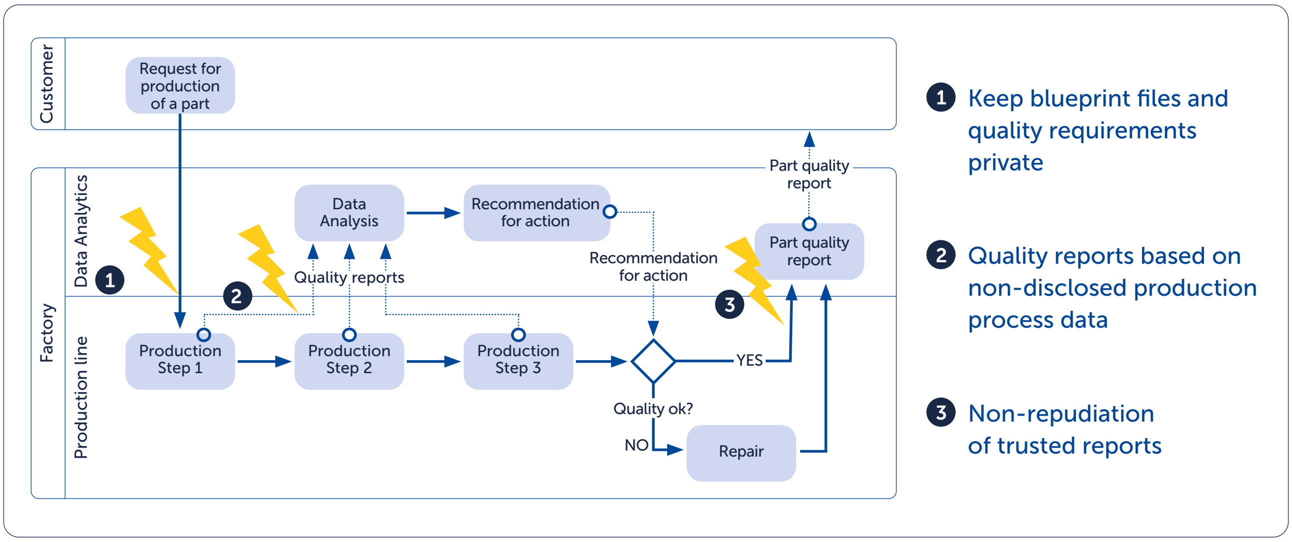


Figure 20: Example of BPMN process flow diagram to identify current issues taken from a patent registration case

E. 4+1 View Model Diagrams

UML class diagram

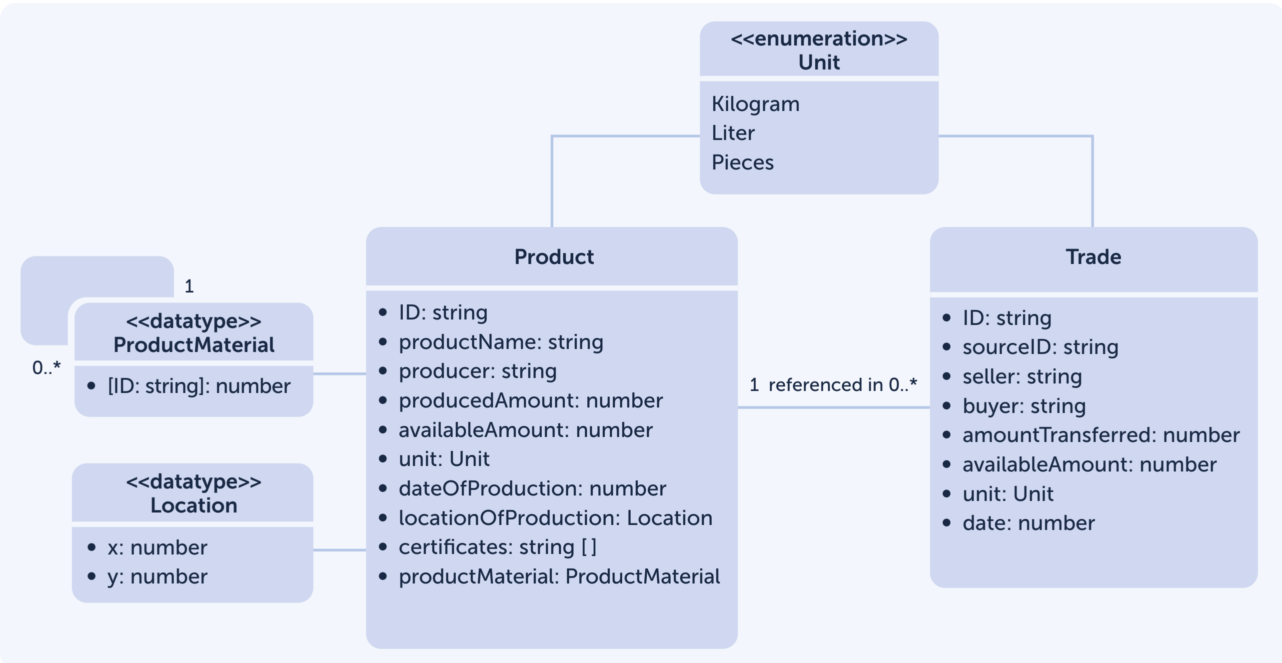


Figure 21: Example for UML class diagram for the logical view taken from a digital commodity exchange case

UML component diagram

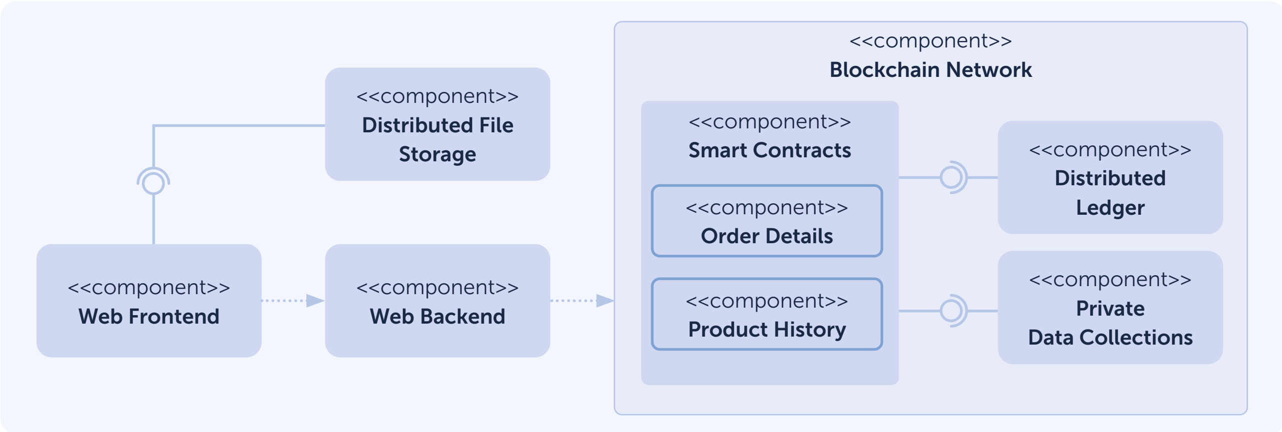


Figure 22: Example for UML component diagram for the development view taken from an industrial supply chain case

UML sequence diagram

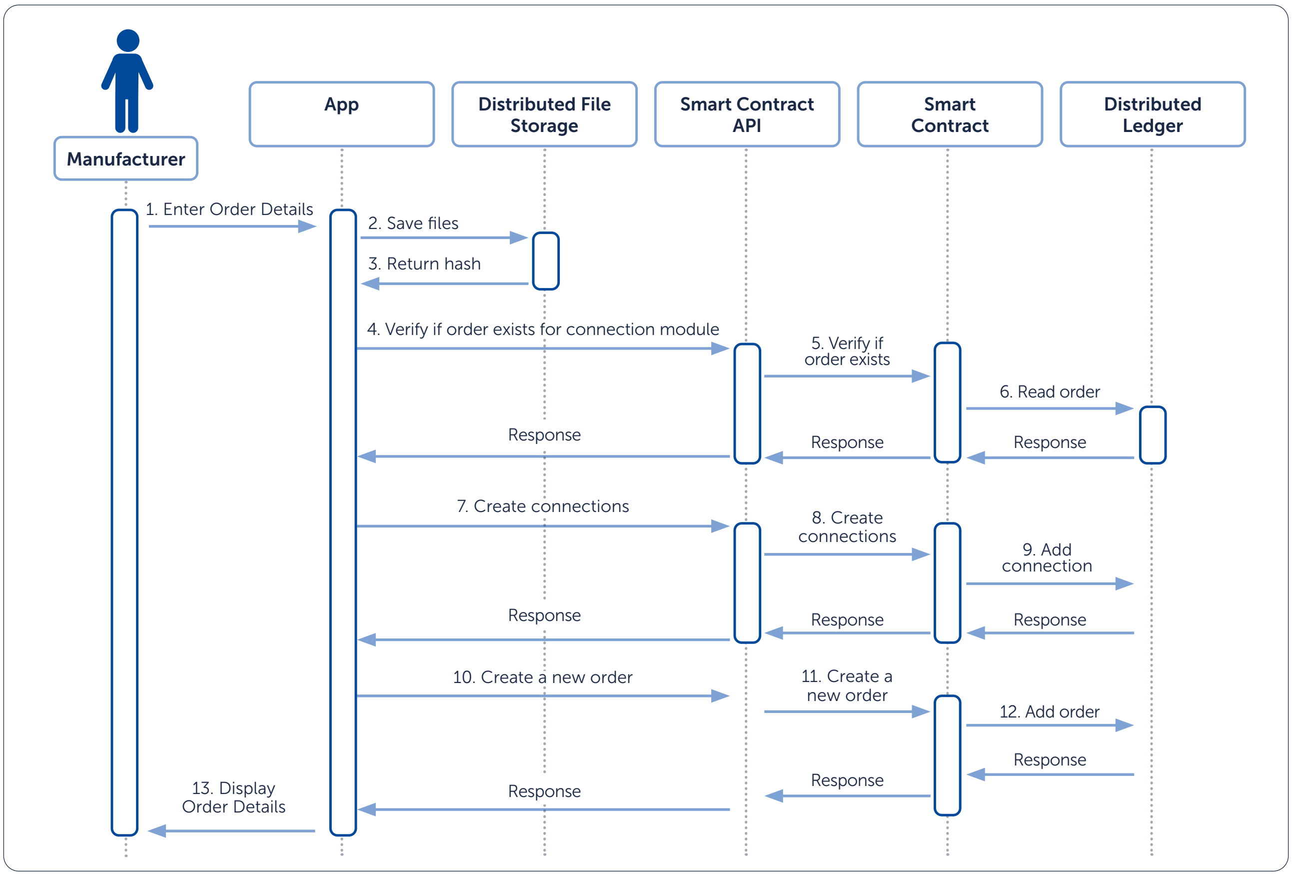


Figure 23: Example for UML sequence diagram for the process view taken from an industrial supply chain case

UML deployment diagram

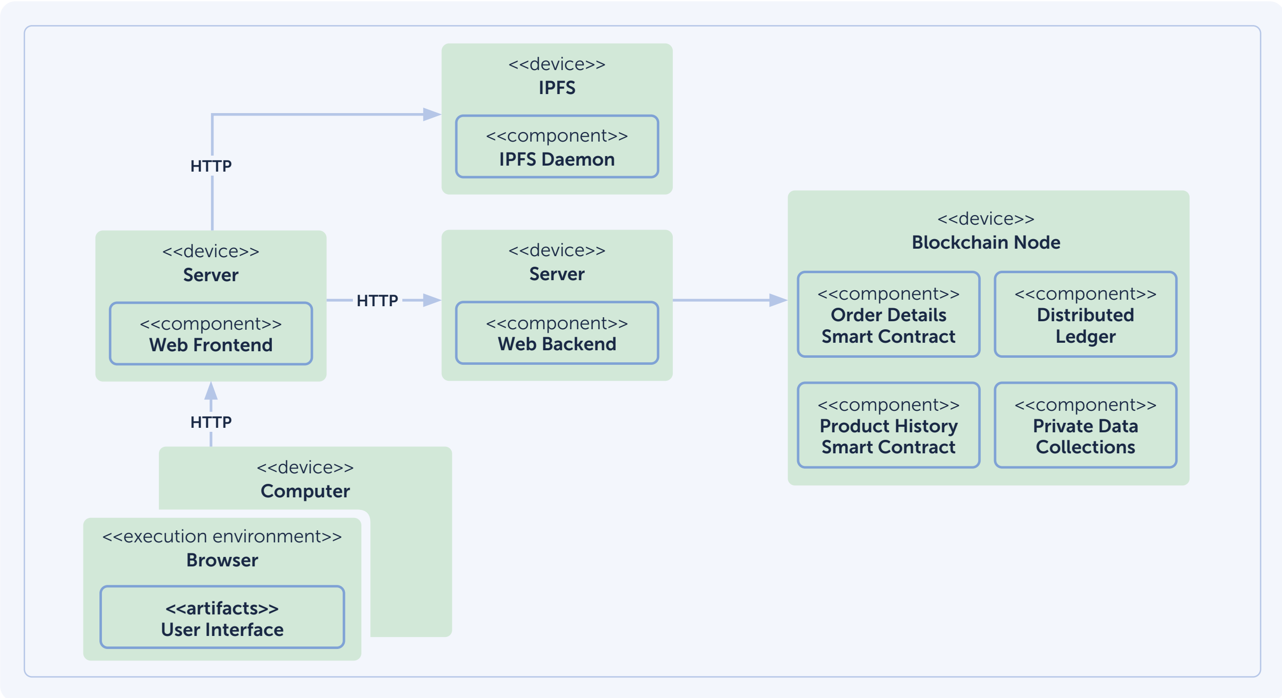


Figure 24: Example for UML deployment diagram for the physical view taken from an industrial supply chain case

Scenarios

Scenario 1:
Akwasi, the head of the family, wants to sell land

Akwasi wants to sell a part of his statutory land. He registers first to the distributed ledger through the app on his smartphone. His identity is confirmed by the government. Akwasi then draws the surface he wants to sell on the satellite map interface provided by the app. He sets a price for the land and submits a sell request for a customary-held property. His request is sent to the chief of the community in Trede. He checks the request and price and accepts it. The request is then forwarded to the Customary Land Secretariat (CLS) of Trede. CLS also approves his request. Finally, the land commission checks for any statutory land conflicts for that ground and provides approval. The transaction is executed because the transaction and price are approved.

Scenario 2:
Mawusi, wants to buy land

Mawusi wants to buy some land. She registers first to the distributed ledger through the web interface from her laptop. Her identity is then confirmed by the government. Mawusi searches for lands for sale in the region Trede, which is her first choice. She founds a sale proposal for statutory land. She likes the ground, which can be registered in a statutory manner. She then sends a buy request with the proposed price. The land transaction should already be approved by the Chief and CLS of the community in Trede to get listed. The land commission confirms the absence of statutory land conflicts for that ground. The transaction is processed successfully, and the land is finally owned by Mawusi after her payment.

Scenario 3:
Corporation A, potentially wants to buy large agricultural surfaces

Corporation A wants to work in a sustainable manner. Therefore, he wants to know exactly (if he can) which grounds are for sale in customary and statutory land management systems. Thus, the company registers to the Fit4Ghana Blockchain, receives the authority's approval of registration after presenting identification papers of the company, and acquires access to lands for sale. He can see the available grounds and send requests to buy them. Any attempt from the land commission, the chief of the community, or the company to execute land transactions outside the blockchain system is forbidden by law. Corporation A can only buy approved lands by the stakeholder through the blockchain application and cannot come to any conflict as a consequence of such an action. Corporation A buys some grounds in Trede through the blockchain system but far away less than what other corporations could have done without the blockchain system, ensuring a sustainable future for Trede.

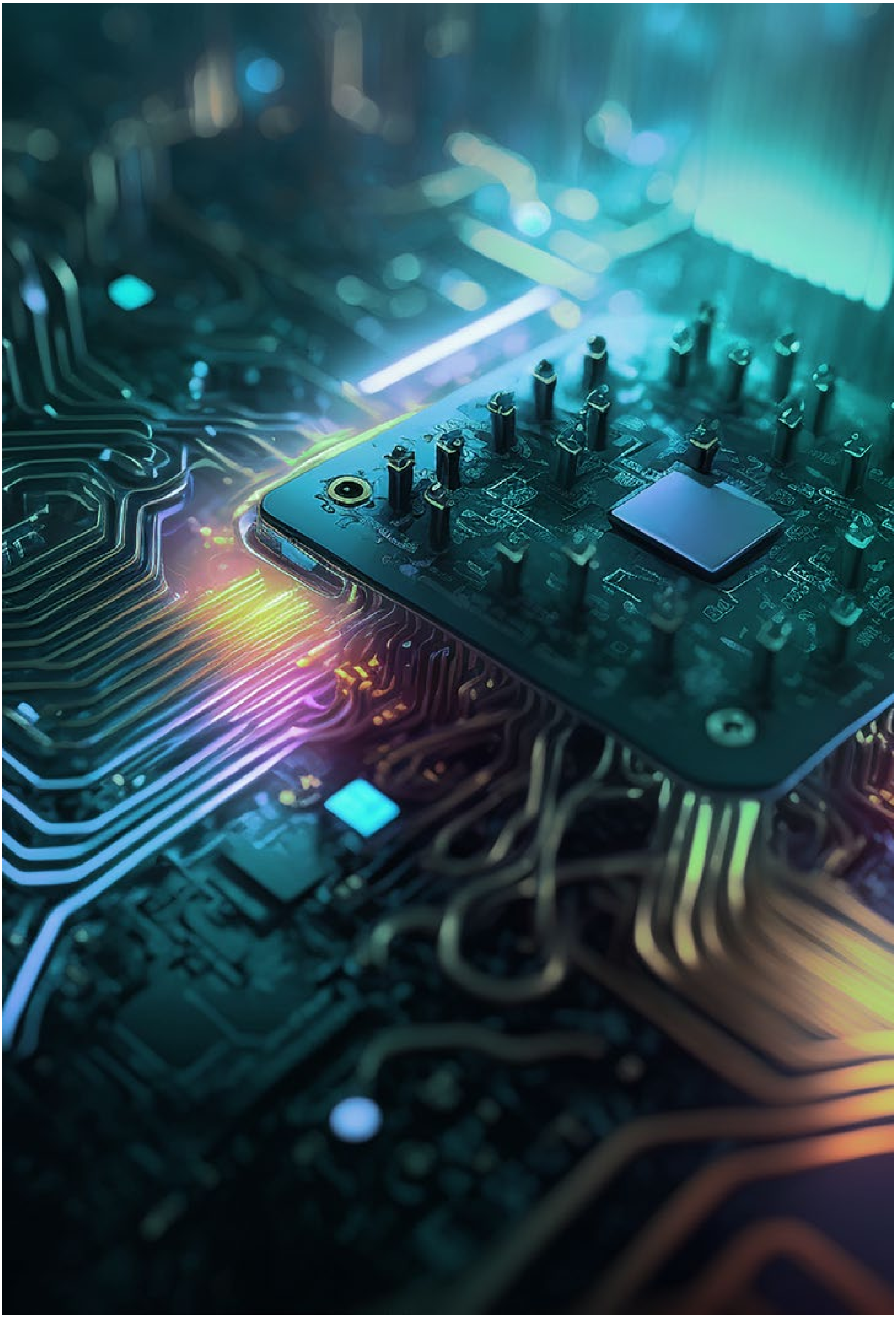
Scenario 4:
Kafui, community chief, approves land transactions

Kafui is the Chief of the Community of Trede. He deals frequently with land conflicts and external parties willing to buy land from his community. He wants to track all land transactions easily and avoid conflicts and land grabbing. Kafui is registered by the authorities in the Fit4Ghana Blockchain as a recognized chief of a community. He receives notifications and is prompted to decide upon the request each time some members of the community want to sell land. He can check the ground the members of the community want to sell as well as the price. If he agrees with the offer, then he receives a confirmation of the land transaction. He can also oversee all the registered grounds and land transactions of his community.

F. Example of DLT-Use-Case-Solver output canvas

Figure 25: Example DLT-Use-Case-Solver output canvas application and middleware layer

Figure 26: Example DLT-Use-Case-Solver output canvas infrastructure layer



Imprint

Published by
fortiss GmbH
Guerickestraße 25
80805 München

Editing
enago, München

Layout
fortiss GmbH

Print
viaprinto GmbH & Co. KG

ISSN Print
2699-1217

ISSN Online
2700-2977

1st issue
January 2023

Photo credits
Title: AdobeStock 273179184
Page 7: AdobeStock 343465704
Page 17: AdobeStock 354335934
Page 21: AdobeStock 191892412
Page 37: AdobeStock 551790029
Page 38: Fortiss GmbH @ Kathrin Kahle

Find here more
fortiss White Paper

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