Designing Incentive Controls for Federated Learning in Aviation

Implemented on-chain

Bruno Lapré

Delft University of Technology

TU Delft

Designing Incentive Controls for Federated Learning in Aviation

Implemented on-chain

by

Bruno Lapré

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on the 10th of March 2025

Student number:	4291751
Faculty:	Technology, Policy and Management
Project duration:	September, 2024 – March, 2025
Thesis committee:	
Dr. ir. M. G.A. de Reuver,	TU Delft, chair
Dr. ir. M. Tuler de Oliveira,	TU Delft, first supervisor
Dr. ir. R. van Bergerm,	TU Delft, second supervisor
Asteris Apostolidis, PhD	KLM, external advisor

Cover: Tokyo Skytree, image by Author

TU Delft

Preface

I've been known to go on a bit too long when given the opportunity and this thesis is no exception, so for once I'll keep something short. This thesis is the result of months of work and probably a bit too long of a period beforehand which I can however look back on with joy and satisfaction.

I'd like to thank, in no particular order, for my entire long student career: My parents, D.S.V. "Nieuwe Delft", the many colleagues I've had across Wines and Whiskies, LiteBit and Hanno, and my friends who I can rely on for coffee, lunch or wine anytime.

Particularly for this thesis, I'd like to thank my supervisors, especially Marcela, for the support, the Desert Bus for Hope charity for keeping me sane, and my partner for the constant mutual strength throughout our respective graduations.

Thank you for reading.

Bruno Lapré
Delft, March 2025

Executive Summary

In aviation, like many other industries, machine learning is a promising way to utilize the vast quantities of data generated and apply them to improve many processes, such as maintenance and flight delay management. As a method of machine learning, Federated Learning (FL) is a promising alternative, keeping sensitive and valuable data private while still allowing training to be done.

Problem areas around FL occur around cooperation and participation: More participants result in better models, but trusting the system and other participants is a high barrier with costs of data transformation and usage being very high. Rewards are necessary but need to be distributed fairly among participants.

The current state of research into Federated learning in aviation is focused on technical implementations and adopting new methods of training. However, institutional perspectives and incentivization are critical aspects of helping successfully implement FL in aviation in the face of regulation and lack of participation due to the fears of free-riding and untrustworthy participants. These areas of research are active but still have gaps regarding implementation and institutional perspectives.

Research Questions Based on previous work building an FL architecture, and within the context of a master thesis, this research addresses the problem and knowledge gap by performing a design science research using the following design goal as a research question: *How can a blockchain-based incentive organizing system be designed and implemented to increase participation and trust in federated learning systems within aviation partnerships?*

Subquestions are posed to guide the research steps, consisting of:

- RQ1: How should existing requirements regarding the FL framework for aviation be modified or updated to consider design around the context of incentives?
- RQ2: What kind of institutional setup is fitting for FL in aviation and how does it inform FL incentive mechanism design?
- RQ3: Which alternative design choices should be considered and selected to best fulfil the requirements within the institutional design?
- RQ4: How does an implemented MVP of the design perform and behave to meet the requirements?
- RQ5: How well do the design and MVP fulfil the role as a solution to the problem of trustworthy incentives for FL in aviation?

Methods The design science research is appended with an institutional analysis using a modified institutional analysis to guide the design. Further research steps included reviewing existing requirements relevant to this context and updating the system requirements based on literature. The design steps were continued with a draft of a design based around smart contracts on a private blockchain, that was implemented and tested. The requirements and context regarding trustworthy FL in aviation were used to validate and evaluate the design.

Results The research resulted in an institutional setup in which the FL reward and organization happened within a regulatory and behavioural context. Requirements were built around security, regulatory adaptability and equitable rewarding, and modularity. Based on the process, a conceptual model of the design was generated, consisting of smart contracts and an external part controlling the FL process and rewarding.

The smart contracts orchestrate FL task registration, access to workflow, and sensitive data methods, as well as the rewarding method. Contribution calculation is done outside of the design to allow for changing algorithms and implementations, while the smart contracts deployed two on-chain tokens

as rewards distributed based on contribution scores calculated: An ERC-1155 NFT-like token MAT that can be burned to gain access to the model, and can be sold on external marketplaces, and an ERC-20 RewardToken that shows participation and enables receiving kickbacks from market sales and providing a small amount of voting power for system governance.

The design was implemented and tested for validity, passing the basic use case tests and fulfilling the requirements adequately. Additionally, the scalability and speed of the implemented contracts were tested, with positive results.

Discussion Key findings were effective use-case for blockchain in FL design, the effectiveness of institutional analysis for helping guide and understand actor behaviours in FL in regulated sectors, and the potential for using dual-token rewards for rewarding models to incentivize participation.

Limitations include a smaller scope for requirements elicitation, being based mostly on previously created requirements, lack of real-world validation of the designed artefact, and a similarly not fully validated institutional design the design was based on.

Future research includes further validation in real-world applications with actual implemented FL and an incentive mechanism, as well as tests around the partnership using numerical analysis or serious gaming simulations.

Advice For users looking to build FL systems in highly regulated sectors such as aviation, knowledge of the broader field is important to be able to understand and adjust to technological developments and regulatory pressures. It is not impossible to design systems that are future-proof by utilizing techniques that safeguard both reliability and adaptability such as the upgradable blockchain-based design presented here. Further developments in this direction should take care to involve experts across the sector, keep up with behavioural and technological changes and investigate innovative designs.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
BC	Blockchain
Cosem	Complex Systems Engineering and Management
DSR	Design Science Research
ERC-20	Ethereum Request for Comment #20
ERC-1155	Ethereum Request for Comment #1155
EVM	Ethereum Virtual Machine
FGM	Finalised Global Model
FL	Federated Learning
IP	Intellectual Property
KC	Knowledge Commons
MAT	Model Access Token
MVP	Minimum Viable Product
ML	Machine Learning
NFT	Non-Fungible-Token
PoA	Proof of Authority
TPS	Transactions per Second
XAI	Explainable Artificial Intelligence

1

Introduction

Industrial Machine Learning in sectors like aviation provides many opportunities for predictive maintenance, optimizing processes, and accident prevention, but has problems regarding participation and trust in the surrounding systems needed for effective learning. Environmental gains, lower costs, and a higher level of safety can all be achieved through successful implementations of Machine Learning (ML), using the vast amounts of data generated by the various devices and sensors already on board most civil aircraft and the surrounding industry. However, even though it is an active field of research, FL is not yet widely used and applied across aviation and other sectors for reasons both specific for the sector and the method in general. In this introduction the problems of participation and trust around ML in aviation are presented from the perspective of Federated Learning (FL), a method of ML that shows promise in helping solve these problems. Problems whose potential solutions' design space is explored throughout this thesis.

1.1. Background

Machine learning is a broad field of research and applications of algorithms that use data to create models to help predict patterns and behaviours. The usage of ML in aviation has utilities along broad categories, and different projects have already provided fruitful progress in utilizing ML for aviation by following governing organisation's guidelines such as EASA's AI roadmap (EASA, 2023) and EUROCONTROL's air traffic control partnerships with AI (EUROCONTROL, n.d.) companies, and through broader research for predictive maintenance (Stanton et al., 2023 Adryan and Sastra, 2021). Using ML can help find known unknowns in the vast quantities of data through the training of models, but the data owners need to be able to collect, transform and utilize this data to perform ML. This means that, besides the cost associated with data processing and training, it might be necessary to share the information between stakeholders to train better and more useful models, as more - relevant - data generally results in better models.

The most well-known method, centralized ML stores all data in a single data warehouse to train models, which is an aspect that raises concerns from potential participants and data-sharing parties regarding privacy, data access and storage. For airlines in particular, the data can contain privacy-sensitive information regarding passengers, pilots, and staff, or contain sensitive information about performance or trade secrets. Often personal information has limits to being shared freely due to GDPR regulations Commission, 2020 or, for employees, collective labour agreements such as the data sharing limits set for the Data4Safety EU program by KLM for limitations around data sharing KLM, n.d. Other concerns regarding data access and storage are also related to security, as having a monolithic data storage design introduces a single point of failure.

A proposed method of ML is Federated Learning (FL), a distributed method where the training of a model is performed locally, and a global model with more predictive power is generated by only sharing the found parameters, not the data itself. This global model is then redistributed to the contributors, after which further training iterations can be done to improve the algorithms further. This method, initi-

ated through research with Google for data training using mobile devices, is enabled through modern high bandwidth networks and applicable for the aviation industry through developments in the Industrial Internet of Things, where smaller devices and sensors can contribute their data easier and faster through more powerful infrastructure, resulting in improvements for processes and maintenance. If applied properly, FL as a distributed method of ML can produce models with similar capabilities to ML while keeping the actual data at the stakeholders. However, it is more important with FL that stakeholders collaborate, and importance needs to be placed on gathering and governing participants to create better models and solve collaborative issues.

To have a successful and trustworthy FL system, participants need to collaborate more: More participants with quality data will generally result in a higher model fit with better applications and results. Problems occur in participation because the needs of different actors in the field of aviation can differ, with some joining with a certain expectation of rewards or model applications. This situation means that collaboration is not often the most optimal game theoretical action because of the risk of free riders, malicious behaviour from others, and data leaks through reverse engineering the models. Participating is already a costly operation from initialization costs and computational intensity, without an actual guarantee of successful results. Therefore the expected benefits and rewards from participating in an FL system need to be higher than the expected costs and potential loss of any strategic position to competitors turned FL partners, and the system needs to be trustworthy for participants to join in.

Trust and good faith participation in an FL network is vital for its success, with trustworthy FL being an active field of research, such as in the comprehensive overview by Tariq et al., 2024. Controlling who joins in the FL training tasks and rewarding or punishing bad actors in a trustworthy way is important for a successful process. Trust is gained with both transparency and fairness in the process, where the participants know what is happening to their data and how rewards are distributed. The issue becomes complex in the negotiation between actors within the institutional boundaries: Aviation actors such as airlines, manufacturers or maintenance groups which can both collaborate and be competitors of each other have varying types of data to offer and want to keep it private, but also need to share enough to train the model. This issue is amplified with the risk that, in FL, data can be leaked from the global model using certain methods (Hitaj et al., 2017). However, the more secure the FL algorithms, the lower the quality of the final model. Additionally, publicly rewarding participants based on their contributions may also leak how much data on a certain topic they have. Combined with the regulations set by the EU and existing agreements on data sharing and privacy, this complexity shows a necessity for a trustworthy transactional layer where rules are set and trusted so participants, end users, and governing bodies can trust the FL process and models.

Proper incentivization of FL participation is a method and subsection of trustworthy FL research that focuses on solutions and methods to calculate rewards with regard to actor behaviour based on their performance. An ongoing research direction, a range of solutions, and steps for further research are proposed Zhan et al., 2021, with different reward methods based on total contribution, reputation, or game theoretical models of behaviour. However, just having an incentive mechanism to fairly reward participants does not guarantee the behaviour of actors, especially within regulated sectors, with the institutional context also relevant for designing and understanding the behaviours and needs of potential participants and model users.

A focus on the institutional context of FL based on the stakeholders and their behaviour, and organizing the incentives for FL is important to adapt. Data collaboration such as in FL is an ongoing policy direction, as exemplified in the EU's broader data strategy (Commission, 2020), and adapting to institutional changes is important for long-term participation in FL systems. It is important to understand the institutional environment to predict and control participants in the face of regulatory concerns and socio-economic circumstances surrounding data sharing for FL.

1.2. Problem

The problem surrounding FL in aviation concerns trust in, and incentives for, the potentially participating actors within a partnership in institutional boundaries. FL is a promising method to train models in a competitive field such as aviation, where broad data availability isn't being fully utilized because of concerns around data sharing, but needs, to increase model quality and collaboration between par-

ticipation parties who need to be rewarded fairly and transparently. The lack of trust between parties themselves and FL systems stemming from competitiveness and potential abuse of FL participation is a problem that reduces the willingness to participate, also given the costs involved and the, for aviation already strict, regulatory landscape adapting to ML developments. The problem is within the dilemma of needing to share data but actors being hesitant to share data, amplified by a need for trustworthy and high enough incentives to overcome these hesitations

This problem is at the intersection of different research fields regarding, with many research directions to tackle this problem are possible. Fitting within a TU Delft Complex Systems Engineering and Management (Cosem) Master thesis scope, this situation can be handled as a socio-technical problem stemming from applying cutting-edge technology within a specific institutional scope that concerns behaviours of different actor types and stakeholders.

1.2.1. Thesis Direction

To further research in solutions towards this problem, this thesis will continue and expand upon the work of a previous Master Thesis by van de Walle, 2024, in which design science was used to propose a framework for FL in aviation, focusing on broad architectural design decisions based on expert interviews and feedback. Specifically, the reward and cooperation aspect, suggested in Walle's thesis as a blockchain and token-based system, showed a promising direction for solving the problem, which could be expanded upon and become the focus of continued research. This thesis explicitly builds further upon Walle's work, with a focus on design around rewarding participation and organising cooperation while maintaining the important trust pillars for FL and the aviation sector

This thesis will present a design of science-based research for implementing a trustworthy reward layer consisting of incentive mechanisms and trust-building systems for FL in aviation using the current state-of-the-art research and technologies. Literature defining the state-of-the-art research surrounding the problem will be found in chapter 2, followed by a review and adaptation of the design requirements found by van de Walle, 2024 for the context of this problem in chapter 4. A broad institutional analysis follows that in chapter 5 to place the design in an institutional context. A design will be drafted, defined and partially implemented for testing in chapter 6, and validated and tested further in chapter 7. A critical discussion on the process and results will be provided in chapter 8.

2

Literature

Furthering the understanding of the problem, exploring the current gap in knowledge and moving further towards a methodology for this research to help bridge that gap requires a literature overview from the angles of technological background, trustworthiness, participation and institutional context. For the background of FL, the technology, its applications and opportunities within aviation, trustworthiness, as well as the incentives for participation are explored. Additionally, literature from an institutional context is analysed to help understand stakeholder behaviour and the rules that need to be followed to understand the current status of those research directions.

The current state-of-the-art research in FL is focused mostly on techniques and advancing learning methods for higher accuracy, robustness, and reliability, although other research directions for FL could be more relevant to the problems of participation and trust. Advanced research is being done into the mathematics and computer science to further methods and model generation in directions outside of the technical scope of this research, but it will briefly be touched upon. Besides the technical advances around FL, contextual aspects such as incentives, cooperation and governance are relevant subjects of active research directions and will be explored. The state of the art from an institutional perspective will also be explored to understand the research state of the FL context regarding actor behaviour and rules, with institutional being defined around the background rules and regulations, overview of stakeholders and partnerships and the specific characteristics of FL within them.

Aviation as a sector is used as the context of the literature and thesis in general as a broad and promising field for FL applications. Defined for this thesis as the broader collection of organisations, companies and institutions, aviation does for this thesis not only include airlines, but also manufacturers, airports and support staff. This collection is highly interconnected and is a competitive field that can work on the cutting edge. This broad definition and analysis scope puts the problem and research further into a broad aviation sector wide scope.

An overview of the current literature surrounding FL for aviation was collected through various channels and reviewed for the technical background, trustworthiness and for institutional context. Collection of relevant literature was performed through a search of relevant keyword combinations per background subsection on Google Scholar and Scopus, sorted by search term relevance and data, supplemented with both forward and backwards snowballing and recommended readings from supervisors. Keywords used for the search were combinations, variations and synonyms of terms relevant to the subtopics of FL in general, trustworthiness, incentives and institutional research. For the background of FL and its applications in aviation, the terms used were ”Federated Learning”, “Aviation”, “Applications”, “Distributed Learning”, ”Data Sharing”, and “Data Usage”. For trustworthiness and incentives, “Trustworthy”, “Fairness”, “Incentives”, “Incentive Mechanism”, “Rewards” and “Token Rewards” were utilized. Terms used for the institutional context were “Regulations”, “Collaboration”, “Partnerships”, and “Governance”.

The selection of works to be included and expanded upon was performed based on recency, the potential for snowballing and relevance for the problem of organising incentives and trustworthy FL, although no specific cut-off point for date was used to keep the potential selection broad.

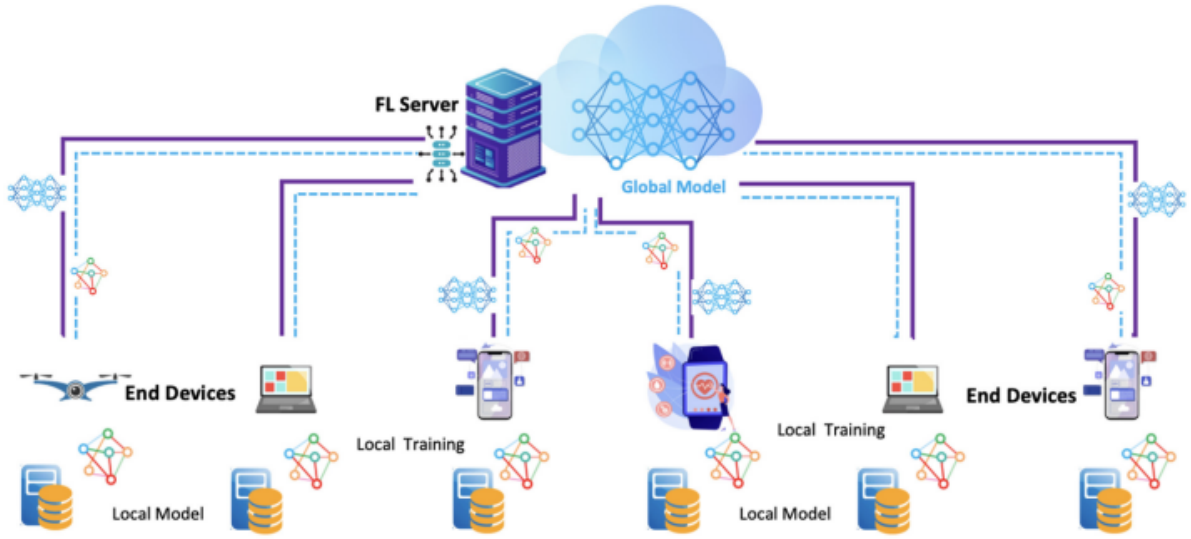

Figure 2.1: Centralized FL, Tariq et al., 2024

2.1. Federated Learning Background

A broad understanding of the technical aspects and works in FL both in and outside the aviation sector is important to set the stage for the understanding of how to design within them. As current literature on the technical aspects is highly advanced and on a rapid pace, the basics of FL as well as the broader research directions regarding trustworthy FL and incentives for participation are explored.

2.1.1. Federated Learning

Federated learning is a method of machine learning for training a model in a distributed way, requiring collaboration between and local training by the participants. Initially suggested through McMahan et al., 2023 with Google, it has spurred its own extensive research field (Yang et al., 2019) and has advantages over centralised machine learning regarding privacy, storage and network costs.

FL is generally initialised with an initial model for a specific training task on a central location, which is transferred to the participants, who do local training using their own data and update the global model with their found models. Although other architectures also exist, figure 2.1 from Tariq et al., 2024 shows a diagram of such a Centralized FL setup. Using training rounds, the updated global model is shared back to the participants, and the global model is continuously improved as an aggregate of the local models. It has gained traction as a distributed learning method in Industrial applications such as in aviation, utilizing faster machines and high-speed networks in which data to be collected at a rapid enough pace (Bugshan et al., 2023). Different architectures for FL can be designed and developed within three categories: data-distribution-based, often vertical or horizontal, scale-driven, cross-silo or cross-device, and coordination-based, centralized, decentralized or hierarchical (Tariq et al., 2024). Whatever method is chosen, FL is a method that comes with costs of computation and communication (Banabilah et al., 2022) for both the initializing parties as well as the participants. Real-world implementations of FL are still scarce, but some have progressed to a deployed state, such as FATE, as the first industrial-grade open framework for the secure computation of FL tasks (Y. Liu et al., 2021).

Current developments in the technical aspects of FL are advances in implementations and improving aspects like privacy, convergence and speed. Kairouz et al., 2021 presents an overview of advances and open problems. The primary questions in FL research posed here are based on efficiency and effectiveness, threat models, the privacy of user data, attacks and failures, ensuring fairness and system challenges. As a critical indicator for effectiveness, participation must remain high through ensuring these challenges are met, especially fairness. Besides technical developments, critical developments are also being made in making FL trustworthy, an important part of the indicated problem for further applications in sectors like aviation.

2.1.2. FL In aviation

As indicated in the introduction, within aviation FL has multiple potential use cases and promising directions of research that show the relevance and importance of further research and development in this field. In general, the EASA AI roadmap (EASA, 2023) expects the sector to be positively impacted by AI/ML for aircraft design, environment and more. For example predictive maintenance (Stanton et al., 2023) and flight delay prediction (Guo et al., 2021), with ML being actively used as well, in for example temperature prediction (Apostolidis et al., 2022). Additionally, flight safety (Oehling and Barry, 2019) and environmental gains (Wu et al., 2024) are included in the possible applications of ML and therefore potentially FL in aviation. With well-utilized data, both economic and sustainability goals can be better achieved when using ML and FL, although the research and guidelines show models need to be safe and reliable.

Critical in aviation is safety and reliability when using AI/ML models generated on data: importance is placed in the EASA AI roadmap as well on making sure models used in aviation are trustworthy, and can be assessed on safety, security and ethics level if relevant. Human oversight is important in almost all applications of AI/ML to prevent biases and wrongful use, exemplified in the concepts introduced by Kyriakou and Otterbacher, 2023. Additionally, making sure knowledge exists where the model came from, the explainability, is an important part of safe and reliable FL (Saraf et al., 2020). For broader Industrial IoT, Jagatheesaperumal et al., 2024 suggests design strategies to enable trustworthy FL for interoperability and robustness including adaptable learning architectures and an adaptive hybrid approach between probabilistic and deterministic models to thrive in a changing landscape of FL and industries such as aviation.

Designing an FL framework for predictive maintenance in aviation was previously done in the thesis by van de Walle, 2024, where sector-specific requirements were gathered based on themes of privacy and security, regulatory compliance, management of IP, Distribution of Benefits, Impact of FL, and Adoption and Trust. The designed framework, utilising a consortium blockchain, received expert feedback on the trade-off between data privacy and performance, erring on the side of performance, as well as moving towards rewarding participants in the form of tokens for access. The responses from the sector stakeholders indicated through that thesis show a high level of interest in further research in FL for aviation. The key results from this thesis, the framework itself based on expert interviews and research, is presented in appendix D

The possibilities for ML and FL in aviation are varied and promising, with research and development not only towards the applications but also towards necessary safety and trustworthy aspects. To make sure the applications perform well and are reliable in a regulated sector like aviation, the concept of trustworthy FL is an important and distinct research direction to understand.

2.1.3. Trustworthy FL

An important aspect of FL that can enable participation and regulatory compliance is the concept of trustworthiness. Tariq et al., 2024 presents a detailed overview of which challenges and points of interest exist for designing and developing Trustworthy FL, allowing for attracting actors through fairness and expectations in the FL process, and keeping within governance policy directions of trust and safety of the used models. Pillars for trustworthy FL are defined therein as Interpretability and Explainability, Fairness, Transparency, Accountability, and Privacy and Robustness. Around these pillars, the state of research and directions are briefly explained in this subsection, as well as the pillars' relevance in the aviation sector.

Interpretability and Explainability are key concerns for usage of ML in critical services to make sure the model is understood. Often cited by lawmakers and governing institutions and sometimes defined as explainable AI (XAI), knowing why a model behaves a certain way or how it could have been influenced by the training process is important to deal with its results or behaviour and can help expand safety. Especially in sectors and applications noted as high-risk such as aviation, XAI is important. Work is being done to provide explainable FL models such as through explainable vertical FL (Chen et al., 2022) and methods for keeping track of the reputation of training participants (Y. Wang and Kantarci, 2021).

Fairness towards participants through proper client selection and rewarding the contributions of those clients properly helps with making sure participants are attracted, stay engaged and provide quality

contributions. Within aviation, the stakeholders are diverse but the competitive field with often thin margins makes fairness important across the field. Increasing fairness can be done using trustworthy client selection by providing incentive mechanisms and rewards based on multiple methods as collected by Zhan et al., 2021.

Transparency and Accountability provide insight in how participants have contributed to the model, as well as how rewards are distributed if any. This is useful for increasing trust in the models as well as the overarching systems that govern the FL process. Aviation stakeholders might be interested in knowing how rewards and responsibilities are distributed and organised so trust is increased in a fair reward construction. Solutions exist for contribution evaluation based on Shapely Values, which calculate how much a participant has contributed, such as Tastan et al., 2024 , and Shi et al., 2024, which additionally rewards participants based on reputation and distribution of model gradients.

Enabling data Privacy and having Robust FL is vital for participants to have trust in the FL models and processing systems surrounding the training. Making sure no data is leaked from both the local aggregated data and the global model that is distributed is a key concern for participating parties across all sectors. FL is not a perfect system and has vulnerabilities and risks: Bouacida and Mohapatra, 2021 provides an overview of these threats and attacks possible from clients, aggregators or outsiders. Attacks can focus on data, algorithms and federations. Although privacy in FL is expected to be high because data is not actually shared to a central server, studies show finding identifiable data from the global model is possible with some work (Melis et al., 2019, Hitaj et al., 2017). Developments in this field move towards solutions like differential privacy (Ouadrhiri and Abdelhadi, 2022) and homomorphic encryption (Y. Liu et al., 2021), which can however decrease the accuracy of the global model. When designing FL, privacy versus performance is an important trade-off to be made by its participants. In sectors like aviation, it has been indicated by experts that the performance and subsequent reliability of the model are more important than the security in such trade-offs, mostly to keep up with safety regulations (van de Walle, 2024).

Trustworthy FL is a distinct and active research area wherein many solutions and problems are being researched within multiple pillars of trust relevant to broader applications as well as FL. As an important aspect of FL relevant for aviation, increasing trust in the FL process is an understood goal that helps with the problem found. A specific aspect of trustworthiness is rewarding participants in a fair and open way, something that is also a research direction to be investigated further.

2.1.4. Incentives for FL

As part of the design for trustworthy FL, incentive mechanisms as means to reward contribution is an important aspect to be aware of, but has some inherent difficulties compared to simple rewarding structures. To make participants join any system, incentives are needed to make sure the expected rewards are higher than the costs to join or actively participate. Incentives can come in many shapes and forms but should be related to the quality of the activity of the participant and outweigh any opportunity or financial costs. Zhan et al., 2021 explains how standard, existing incentive mechanisms that rely solely on utility and game theoretical optimums do not apply well in FL because of difficulty in evaluating the training data value of each client and of modelling the learning performance of different FL algorithms.

Incentive mechanisms for FL is an active research field with many potential solutions along different axes of evaluating contributions. Three categories for incentive mechanisms were suggested in Tariq et al., 2024: Game Theory, Reputation-based and Contribution-based. Zhan et al., 2021 also shows a taxonomy of incentive mechanism designs for federated learning driven by different aspects: Data Quality, Data Quantity, Client reputation, and by Resource allocation, both communication and computation. Recent suggestions include the work by Zhao et al., 2024, classifying participants as diligent, inefficient or malicious and designing a self-safeguarding incentive mechanism tailored to the various types of participant behaviour. Examples also include Cai et al., 2024 who propose utilising contract theory and a reputation mechanism, while Z. Wang et al., 2022 proposes a blockchain-based architecture for rewarding based on reputation points and Shapley values. As an active research field, the proposals for calculating rewards are varied, with some criticizing the usage of purely the Shapley value as too limited or computationally intensive (Z. Liu et al., 2022). Although no single solution has been broadly chosen as the best fitting incentive mechanism for any given application, the general consensus is that contribution-based equitable rewards help with trustworthy FL, and show the need

for fit-for-purpose designs.

Specifically for aviation, incentives have some key concerns that should be addressed regarding economic and sector-specific properties. As a highly competitive industry, the economics of air transport is complicated with problematic financial sustainability, especially airlines (Tretheway and Markhvida, 2014), giving a need for high return on investments. As a potential boon for economic sustainability, FL and ML participation can be part of the business model by improving processes. Additionally, the aviation sector is slowly moving towards sustainable developments (ICAO, 2022) for which ML models, including FL, can be useful. The benefits the model itself can have on the broader goals of the sector can, for more altruistically or environmentally conscious stakeholders, be an incentive in and of itself.

The current state of literature show that trustworthy FL in aviation has the incentivization of participants in a fair and transparent way as a key concept, fitting within the found problem. Furthering insights in the institutional context will help expand the understanding of the technical aspects found so far.

2.2. Institutional Context

Understanding the state of research in the direction of institutional analysis is important to guide design around behaviour, rules and collaboration. To build and expand trustworthy FL, potential participant behaviour and their interactions need to be understood. Work that has been done in this direction was collected using the previously discussed methods and is presented from the perspective of the resources, the formal rules that need to be followed, as well as insights from partnerships and organisation research fields relevant to FL. This section is set up using terms from the Institutional Analysis and Design (IAD) framework (Ostrom, 1990), a well-known method of doing institutional analysis and used here to guide the reporting of research done for the context that informs the actions taken, based on resource characteristics and formal rules. Additionally, the actors and partnerships that are possible within an FL organisation are analysed based on existing literature. The state of the literature within the IAD compiled here is transposed and later in chapter 5 analysed further.

2.2.1. Resource Characteristics

The resources utilized for FL are data, with the end model also being a resource that is being produced by the FL process. The characteristics of the resources are analysed to understand behaviours and needs around the process. Data is generated, transformed and potentially trained with during the FL process, and the resulting models, both local and global, have their own unique features.

Data needed for FL needs to be found, gathered and transformed to be useful in the FL process and is a non-exhaustive resource able to be generated from any number of sources. The data is generated and exists from numerous different types of sources, from aircraft data to personnel information. The data generated can be in various formats or data types, but is within aviation often standardised according to existing agreements (Keller, 2016). Gathering the data is often a costly step where the data needs to be found first, and then processed and stored, needing expertise and computational abilities like storage and processing power. Transforming and processing the data is another costly step, but is needed to make the data useable.

If an FL task is specified, a workflow to prepare the data is often part of the preparation, where certain data has to be collected and transformed in a specific way. This step can consist of sharing an example model or dataset, from which the participant has to take steps to prepare their own data. This workflow can also contain private data. An example of this is Zeng et al., 2023 optimizing for communication overhead.

A key characteristic of both data and the models generated is that it could be a unique intellectual property (IP). A concern also found in van de Walle, 2024, the generated models are the result of work, generated from owned data, and are the property of those who generated it, but ownership can be shared and agreements need to be made for the ownership of the model. Such concerns result in the need to define ownership and potential limits for the usage or distribution of the model.

The generated models are also non-exhaustible and infinitely copyable, but might need some knowledge about how to use them effectively. Additionally, a key feature of such a model is its quality, and its ability to predict, which is an indication of the value of the model. The value of a Finalised Global

Model (FGM), the end result of an FL task where the model does not get significantly better, comes from that predictive power applied to the stakeholder's own processes, or its value as a sold or leased model Song et al., 2019. In the worst cases as discussed previously, the FGM might have some characteristics that leak parts of or characteristics of the original training data Jin et al., 2021.

The costs and reward values are the resources that guide the behaviour of stakeholders, as economic participants are interested only in the value of the end result, so the worth of the internal FGM application plus values from sharing the FGM plus potential incentive rewards, are higher than the costs to gather and transform the data and participate in the FL training task.

2.2.2. Formal Rules

The formal rules are those that need to be followed by the sector and all participants, and include laws and regulations that govern and limit the playing field. The EU is the focus of the formal rules inventory as it has one of the most comprehensive set of rules, regulations and policy directions in the world: The AI Act is one of the first comprehensive AI/ML regulations implemented, defining a risk-based approach for regulating AI and ML, including FL.

The AI Act itself and the broader policy direction of the EU regarding ML-related developments define restrictions and requirements on development and research. The AI act is based on a risk-based assessment, and Amidi et al., 2023 finds that FL, when the resulting models are used for critical processes like aviation, could be seen as a high-risk application. Participants and the system as a whole need to follow the obligations set for high-risk AI systems such as risk assessment, traceability of results and compliance Additionally, Woisetschläger, Mertel, et al., 2024 and again Woisetschläger, Erben, et al., 2024 analyse FL within the EU regulatory framework and ask the questions about who the service provider is, and highlights the importance of audibility, verifiability, integrity, and privacy in FL to work within the EU regulatory landscape. They give some key research issues regarding FL under the AI Act by concluding that data governance regarding bias and privacy, energy efficiency robustness and quality management are big challenges for FL.

The GDPR, an important act that often needs to be consulted when any kind of data processing occurs, is relevant for privacy only if personal data for individuals is being shared: as it needs to be purposeful and allow for the subject's rights to be enforced. Removing individual data from a completed model is a challenging technique known as unlearning. If personal data is being used as part of the training, FL is not secure enough (Truong et al., 2021) and in such situations, the GDPR has to be followed with all the rights of the data owners. An example of this previously mentioned is the KLM pilot collective labour agreement limiting and regulating data shared for sharing data regarding flight safety (KLM, n.d.).

Regulation on the usage of FL models and algorithms for aviation is based on general safety regulations as well as guidelines from high-level organisations such as the EASA. Airworthiness and regulations are arranged in EU regulations like EU 2018/1139 and EU 2015/640, which provide comprehensive guidelines on safety as a priority. Additionally, information system (IS) security in aviation is impacted also through EU 2023/203 and EU 2022/1645, setting requirements for IS safety. The EASA provides guidelines such as the AI Roadmap (EASA, 2023), helping organisations to come to terms with the regulations and how to implement solutions. Other International organisations also have visions and regulations on model and algorithm usage and data protection. The European Centre for Algorithmic Transparency (ECAT), hosted by the Joint Research Centre (JRC) is set up for controlling the transparency of algorithms used in important systems. When collaborating for FL, participants need to prioritise safety and IS security, while also staying within the guidelines of international organisations.

Important to note is how the research into and the application of ML/AI including FL can outpace regulatory space. Even the AI act might not be sustainable for short-term developments, and big swings in heavy regulations might de-incentivize potential developments or applications Wachter, 2024. The risk that is associated with participating in FL is therefore influenced by the ability of both the FL processes and research, and the regulators to adapt to changing winds.

The current formal rules dictating how FL participants should behave are mostly focused on security, assessment of risks and traceability, and safety. Working within a developing and cutting-edge field does not mean that vital organisations can bypass the rules for rapid development, but rather have an obligation to be as compliant as possible in the face of an evolving regulatory landscape, especially in

the EU.

2.2.3. Partnerships and organisations

The collaborations that potential stakeholders undergo to perform FL is important to understand to understand the dynamics and behaviours within them. Organisational structures formed around such collaborations are influenced by external and internal factors and shape the behaviour of key actors. These actors as well as their potential behaviour are here explained based on literature.

The stakeholders for an FL training system consist of the trainers, controlling parties as well as external influences through regulatory bodies and support staff. Needed for FL are parties actively willing and participating in the process, support staff that help with developing and transforming data as needed, governing actors that help organise and provide governance for the collaboration of actors and regulatory bodies that influence the system directly or through formal rules (Bonawitz, 2019). For aviation, van de Walle, 2024 applies the FL architecture to airlines, OEMs and more actors that can perform different roles within the system.

Although multiple ways of organising FL have been proposed, a consensus about which is best fitting along multiple axes has yet to be reached. FL has a variety of technical architectures and designs possible which also have an influence on governance design. A horizontal FL system needs different agreements than a fully decentralized one. As FL is still a developing research area, other types of partnerships or collaboration setups are based on existing partnerships, like the fully consortium-controlled model suggested by van de Walle, 2024, or are not widely implemented and are waiting for further developments. An important aspect of forming potential collaborations for FL is how it compares to actors just building an ML model in-house. Bi et al., 2024 uses game theory in cross-silo horizontal FL to show the behaviour of potential participants, providing actionable insights such as how relatively smaller participants are a better fit and how rewarding participants is important to push institutions towards joining if otherwise unwilling.

Zuziak et al., 2023 proposes the usage of partnership types called Data Collaboratives using decentralised learning - such as FL - as a way to shift from data ownership to data access and sharing-focused economy while keeping control of the data. A key opportunity for data collaboration through FL is dealing with a lack of data usage: a tragedy of the anti-commons where data is underused because of complex and overlapping ownership and privacy issues. Moving data towards usage as a common good by engaging groups in collaborations is, therefore, a key method for stimulating data collaborations like FL. Ostrom's design principles (Wilson et al., 2013) are also suggested to be used for data resource sharing and moving (potential) participants towards collective action and sharing their data. Important when defining a data collaboration is data governance, wherefore Peregrina et al., 2022 lays the groundwork by proposing a model which includes multiple assessment steps, a training configuration agreement and the definition of a data governance model for enabling FL.

With multiple types of organisational structures possible, it is important to note how the actors behave within the different organisations. The differing types of actors within a specific kind of collaboration are influenced by their roles in the system and the governance model utilized. Guided by the rules set within such a partnership, the technical infrastructure that binds the collaborators together for performing actual FL tasks and rewarding is the last aspect of the organisational structure that is analysed.

2.2.4. Infrastructure

The proposed aviation FL framework by van de Walle, 2024 suggests a consortium blockchain-based token-based reward scheme, showing interest in Blockchain as a technical aspect of the design. The blockchain-based implementation provides a transparent, trustworthy layer that controls important aspects of the FL system for rewards, learning and governance and is potentially a key technology in secure distributed learning, as suggested in the systematic overview by Li et al., 2022. Additionally, the design used by Majeed et al., 2023 shows a direction using tokens as rewarding mechanisms, used in a public blockchain. Further applications of blockchain in FL are used in Rejeb et al., 2021 and in Y. Ma et al., 2020 for trustworthy security and access control for FL in general by C. Ma et al., 2022 and Shayan et al., 2020. Aviation-specific blockchain cases include Deng et al., 2024, using blockchain as the knowledge distillation method and data exchange process validators, and Nguyen et al., 2021 using blockchain for digital twin analysis for air mobility.

Using blockchain as a governance tool has been done in multiple applications. Distributed Autonomous Organisations (DAOs) can, often in public cryptocurrency projects, govern themselves using reward tokens that are used for voting and can provide value, although some real-world implementations have resulted in catastrophic problems when exploits were found in for example the original TheDAO (DuPont, 2017). The DAO design remains actively used, even though in public implementations the voting and governance are often not effective (Sharma et al., 2024), but in general governance systems such a design that provides some influence as a reward is promising (Kaal, 2021).

Design choices need to be made around infrastructure and collaboration based on the specific stakeholders wherein the FL task happens. There is no set best way of organising both the actors as the FL process itself within the context of the formal rules and resource characteristics, but more than enough opportunities and potential solutions exist. The institutional design needed to further design insights in this problem context will be further created in chapter 5.

2.3. Knowledge Gap

As an active research field, FL is constantly developing and rapid progress is being made on many aspects. For the context of participation and trust by the aviation industry specifically, indeed technical applications and solutions for both trustworthiness and institutional design are aplenty, but more perspective and designs for organising and controlling around incentives are needed.

On the technical aspects, much development is being done, but the multitude of technical designs and infrastructure as well as the evolving nature of methods for both FL itself and incentivization asks for a fit-for-purpose method of both selecting and adjusting appropriate methods of FL and rewarding. Even though Industrial IoT as a broader sector is making strides in adapting FL, aviation-specific research is still developing the applications and best practice methods.

The aviation sector as a whole, consisting of various interconnected and competing organisations, needs clarity and guarantees to cooperate on processes like FL. The difficulties they face regarding trust and collaboration while keeping up with regulatory concerns show the need for further research into solution spaces that can catch those issues.

For incentivization in FL, similar strides are made with many different methods and developments being done to attract participants who can trust the rewarding methods. Although behaviours have been modelled, not many links have been made between actor behaviour within a defined system boundary that has certain regulatory and economic expectations and the used incentive designs.

Organising FL within a partnership or collaboration has not been well explored in real-life situations, with initial steps being taken only in specific sectors or without long-term analysis of behaviours or success. The evolving landscape of technology and a slower regulatory system catching up with the developments make fitting designs and solutions a complex problem with open questions about what a fitting institutional field looks like and how which rules need to be followed.

When moving towards a potential implementation of (part of) the FL framework proposed by van de Walle, 2024 and looking at the state-of-the-art research, a knowledge gap arises regarding the systematic design of incentives and FL techniques within the institutional context and specific sectors. The FL framework suggests a specific incentive mechanism, but the current research state has not converged on any best method yet. Additionally, the institutional context of the aviation sector for utilizing FL methods is not a broadly explored topic. Defining the potential solution space needs furthering the understanding of both the institutional context and technical options and the interaction between them that works for an incentive mechanism. How a governance model can be adapted to such a system is also not a thoroughly explored research area.

Designing for FL in aviation means a fit-for-purpose design for a specific coalition of actors with broad goals. On the meeting point between incentive mechanisms, institutional context, and data collaboration governance, therefore, lies a research area where designs can be made and tested to aid effective incentivization and governance of FL in aviation.

2.3.1. Research Direction

Current research is highly focused on technical specifications and trustworthy FL. As part of the Fairness pillar of trustworthy FL, methods of reliable incentivization and contribution evaluation are important topics. When combined with an institutional perspective, the partnership design, incentives and mechanisms to control and distribute rewards in a trustworthy manner is a promising research direction this thesis will follow.

The design and implementation of the technical and institutional aspects of the proposed FL framework for aviation can bring useful insight into trustworthy FL through incentives, data collaboration and institutional design. This thesis will continue to explore that topic and take steps in research design by adapting parts of the FL Framework's work, adapting particular requirements for design, creating an institutional context and continuing design steps.

3

Methodology

Based on the problem and current state of the literature, the methodology of the rest of this thesis introduced in this chapter was built from the research questions around designing for FL in aviation with roots in Design Science. The methods for each sub-question based on research steps are presented and a visual overview of the research is provided further in this chapter.

3.1. Research question

Framed as a design problem, this thesis is a Design Science Research (DSR) for implementing and testing a section of the FL framework proposed by van de Walle, 2024. Specifically the blockchain-based incentive mechanism and the applicable governance features, which highly influence actor behaviour towards participation and therefore should be verified. The research focuses on designing a reward scheme within FL for aviation within a defined partnership setup and implementing and testing parts of the design.

The research question is shaped as a design goal that helps provide key insights: *How can a blockchain-based incentive organizing system be designed and implemented to increase participation and trust in federated learning systems within aviation partnerships?*

Key definitions for this research are the incentive systems, FL systems and aviation partnerships. Incentive systems are any process or system that works to pull potential participants into the system through fair rewards. The FL systems are any implemented method of performing FL using the data of (a combination of) actors to generate models for participants in the system. Aviation partnerships consist of any group of actors within the aeronautics space such as airlines and manufacturers, but also include but are not limited to airports, maintenance providers and travel associations, who share the common goal of performing FL.

This research will address the knowledge gap by providing insights into how a system that organises a reward mechanism could look within an explored institutional context of collaborations and regulations, as well as how such a system can influence participants. Requirements based on those gathered by van de Walle will be used to see if an implementation of a blockchain-based incentive mechanism performs as expected and is needed to help progress research and applications of FL in aviation.

3.2. Approach

This section describes the approach and steps taken to fulfil the research, with detailed methods following in section 3.3. This research is a research design using the methodological body of knowledge Introduction to Design Science by Johannesson and Perjons, 2021, with a focus on first building the institutional context and then moving towards technical artefact design. The goal was to design and implement an infrastructure around FL and its incentive mechanism on a blockchain setup, within a specific institutional design, inspired by the FL for aviation framework van de Walle, 2024. The end design was also to be implemented and tested as a minimum viable product (MVP) to show its validity

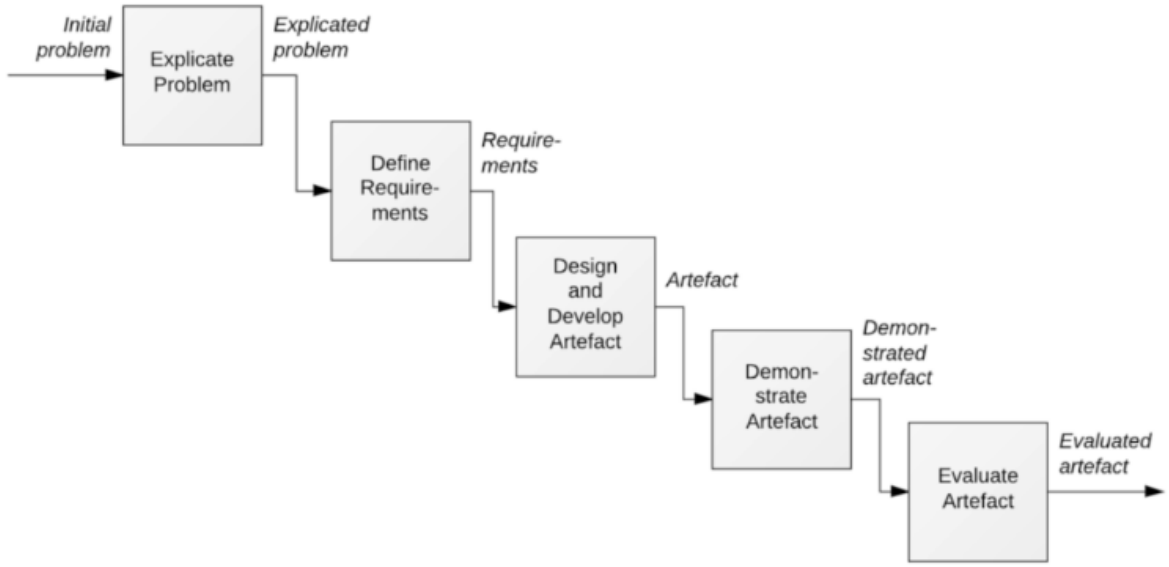

Figure 3.1: Design Science Research

as a trustworthy incentive layer.

The Design Science Research methodology guided the research steps. The steps provided for Design Science are iterative phases, that loop back around themselves and are often reoccurring based on found results. These steps were used to define the research steps of this thesis: Starting from a problem explication, moving towards requirements used to design and develop an artefact suitable to solve the found problem. That artefact is then demonstrated to fulfil its role and evaluated in the broader context of the problem. The steps are illustrated with their in and outputs in figure 3.1, with every step feeding its result into the next phase. Although the process seems very linear, design science is explicitly an iterative approach, with many steps iterating back and progress being made incrementally.

This research was mostly design and development-focused, with additional research on the institutional environment as an additional research step, and the implementation, although every step was addressed. With the problem being identified in chapter 1 and much of the requirements engineering within FL for aviation being done through van de Walle, 2024, those steps were not the main focus of the research: the existing requirements were analyzed and updated to more directly to deal with problem FL faces around incentives. Additionally, extensive evaluating with, for example, actual FL tasks and with feedback from stakeholders were left as further research directions to remain within the scope and timeframe of a 6 month Master thesis.

3.2.1. Research Steps

Based on the DSR design steps, this thesis methodology was based on specific research steps that loosely looped back into each other while being distinct steps. Each step resulted in a subquestion to help the main design goal and had its own distinct methods associated with it.

Requirements

Using the explicated problem definition and the state of the research, the requirements previously suggested and verified by van de Walle, 2024 were analysed and adapted. First, the requirement elicitation methods and results from Walle's analysis were used to extract the requirements and verify their relevance to this design. As the context and the to-be-designed system and artefact(s) in this research is a subsection of the FL framework, the expected outcome was most requirements would remain, possibly with some adaptations, to guide and constrain this design in a relevant way. These adaptations would be made based on the context of the problem of incentives and collaboration, and the context from their origin in the original work.

Institutional Design

From the requirements and problem, steps had to be taken to set the stage for further artefact design and development using an institutional backdrop. To gain insight into the kind of partnership to design for, what kind of collaborations can be agreed upon and in what context the system would behave, institutional analysis is a useful tool. Based on the literature gathered and using an adapted institutional Analysis framework, the institutional context and setups were to be defined, consisting of the roles stakeholders play, the actions they can take in participating in FL, and their behaviour. This institutional design could then be used as a backdrop for the further design steps within the requirements set.

Artefact Design

With the institutional design made, the requirements set and the problem clearly defined, design can be done around potential artefacts that solve the problem. Again the design science research book of methods (Johannesson and Perjons, 2021) was the leading body for the steps taken for this step. First, initial designs and options were collected and potential features were combined and evaluated to best perform within the limits set and the playing field. Selection of design features followed, with the end goal being a clearly defined artefact that could help solve the problem.

Implementation

The designed artefact was then to be implemented as a minimum viable product (MVP) to prove its effectiveness, test the limits of the design if implemented and for potential future usage in further research. The implementation as well as the full features of the design were to be evaluated to show their effectiveness and expandability.

Evaluation

A finalised design was then critically reflected on regarding the requirements and the problem context to find how well the design fulfils its role and how it can be adapted further. As design can be continued indefinitely, this step is important to put a cap on the design and reflect on the future of designing in this direction, finding possibilities and research directions that further the field. Additionally, the limits and problems with the design are reflected upon.

Sub questions

The sub-questions were created from the research steps to provide key results and insights from the research. Every step looping back into itself provides a design process where separate steps influence each other but did need to come to a conclusion at some point to remain within scope. The research steps were transposed onto the research sub-questions that help answer the main research question and expect certain results from the steps:

- RQ1: How should existing requirements regarding the FL framework for aviation be modified or updated to consider design around the context of incentives?
- RQ2: What kind of institutional setup is fitting for FL in aviation and how does it inform FL incentive mechanism design?
- RQ3: Which alternative design choices should be considered and selected to best fulfil the requirements within the institutional design?
- RQ4: How does an implemented MVP of the design perform and behave to meet the requirements?
- RQ5: How well do the design and MVP fulfil the role as a solution to the problem of trustworthy incentives for FL in aviation?

3.3. Methods

The methods used are taken mostly from the research design body of knowledge, supplemented by the institutional analysis framework, adapted to research in an FL context, and author experience. Each research step previously defined is further made explicit here for the appropriate sub-question with the explicit research methods used. The steps and flow of the research and this thesis are additionally presented in figure 3.2.

3.3.1. Requirements (re)defining - RQ1

The requirements used in the thesis of van de Walle, 2024 will be reviewed and expanded upon in this phase. First, the requirements will be evaluated in their effectiveness in evaluating and shaping the design of incentive mechanisms and governance and their fit within that scope. This process will be done using informed arguments and insights from literature found by van de Walle. Additionally, new requirements will be added if necessary through existing literature regarding the institutional context and existing incentive designs and similar systems. This step will result in the requirements utilized for the further design steps.

3.3.2. Institutional background - RQ2

To describe the institutional background and be able to identify the limits coming from laws and governance designs currently relevant for FL in aviation, this research will contain an institutional analysis using previously found literature and new insights, resulting in an institutional design as a context for the rest of the artefact.

The institutional analysis will be performed using the Institutional Analysis and Design (IAD) Framework based on the work by Ostrom, 2011 adapted for usage in the context of FL. As some arguments are made that this framework, initially built for usage in the economic commons, is not a full fit for data as a resource, the IAD is supplemented by the knowledge commons (KC) research framework Sanfilippo et al., 2018, which is an adaptation of the IAD for the knowledge commons like personal information, research and data. The frameworks are briefly touched upon and expanded to argue for their utilization, and then used to frame the continuation and further analysis of the context, a set of argued-for action situations that are relevant, and the actors and their roles in the institutional playing field. This field is then further defined as an institutional design in which the rest of the artefact design will occur.

For the Context, both the formal rules and the resource characteristics are important. On the laws and regulation level, boundaries set by EU law and directives as well as by overarching organizations will be further investigated and clarified using existing literature that investigated the institutional context and legal documents, part of which is already expanded upon in chapter 2. Specifically the EU and aviation governing organisations fit within the scope and have the most comprehensive institutional insight specifically for KLM and other European airlines. International organizations are investigated as well, but they are expected to have less clearly defined and strict rules. A brief review of the relevant resource characteristics and their impact on the system will also be done based on the found literature insights.

The context, actors and partnership setups will inform the design by being adapted into an institutional field containing the actors and contextual limits. This institutional setup will be the baseline of the design, informing the necessary design choices made further.

3.3.3. Design process - RQ3

The design process based on the broad guidelines from Johannesson and Perjons, 2021 will iteratively develop a design for an artefact and the features to be implemented to fulfil the requirements. Starting with drafting initial designs and ideas using kernel knowledge from author experience and existing technological developments, rough drafts will be iteratively developed and expanded upon to result in a complete design which can fulfil the requirements and help solve the problem

This process starts by utilizing kernel knowledge from a short knowledge-gathering phase wherein a base of information on cutting-edge technologies and designs surrounding FL organisation and incentive mechanisms is found. This will mostly consist of finding incentive mechanism designs in FL, other solutions in broader FL organisations, and finding how well they match the institutional setup found before.

Based on the possibilities, drafts of designs will be made and iterated using whiteboard sessions and solo brainstorming ideas and combinations of technologies. As a loose and unstructured creative set of methods, in agreement with Johannesson and Perjons, 2021, the end results from these iterations based on the problem and requirements to solve result in varying designs and ideas for an artefact that might not have a fully traceable origin. As such, the drafts and basic design ideas will be reported on as well as the finalised designs based on the potential alternatives, but the process itself can't be fully

explained.

A design selection procedure will result in a chosen basic design and inform the final artefact. The alternative designs and ideas will be analysed and rated based on their capabilities of fulfilling the requirements and performing their role in the problem area. Arguments will be given on the alternatives and potential design choices, and problems and opportunities for each solution direction will be argued for based on the problem and institutional context. The alternatives and design choices deemed most promising for design solutions will be adapted into a finalised design that turns into the designed artefact that can be implemented and evaluated.

The end result of this phase will be a design and setup of the artefact as a collection of design choices within the system based on the arguments given. The artefact will be defined as a conceptual model and in a UML-like format for ease of understanding and further implementation.

3.3.4. Implementation and testing - RQ4

Implementation starts with arguing for a Minimal Viable Product of the final design to implement, fitting within the scope of development time. The design, assumed as a collection of smart contracts, will be implemented as smart contracts using Solidity for the Ethereum Virtual Machine (EVM), the standard for smart contract deployment. Best practices, other similar implementations, and existing boilerplate designs from other EVM-based smart contracts will form the basis of the implementation. The resulting set of smart contracts should provide an implementation of the design chosen with the basic functionality validated, and fit within the FL for Aviation Framework suggested by van de Walle. The contracts are to be deployed on a private blockchain.

To test the performance and ability of the implemented design to meet the requirements, the deployed systems are tested on aspects of speed, complexity and scalability. These metrics are chosen to provide insight for further research if the design is feasible for larger-scale deployment and usability. A Python script runs the key methods of the contract and measures response time and speed, and scalability was tested based on the file and bytecode sizes of the implementation.

An important note is the decision to not test or build with actual data or even FL training as those methods and capabilities are outside the author's knowledge field and scope of the thesis.

The result of this step is a set of deployed smart contracts, test results and insights from these implementations and tests. These form the basics for the evaluation of the designed artefact.

3.3.5. Evaluation - RQ5

Evaluation of the artefact includes reflecting on the design based on the requirements and ability to solve the problem within multiple scenarios and circumstances, as well as evaluating the success of the design within the aviation sector and other sectors of interest. Based on the design, arguments will be made on the upgradability, future implementations possible on the basic framework, and the process of the implemented smart contracts.

3.3.6. Research Overview

For ease of overview of this research, table 3.1 shows an overview of the steps with the basic methods per sub-question and the associated chapter, while the figure 3.2 provides a visual overview of the steps with outputs based on the research per step. Additionally, a discussion of the research and the results is presented in chapter 8. This thesis continues with the actual steps, starting with the requirements for adaptations in the next chapter.

Research Step	Sub Question	Methods	Chapter(s)
Requirement Adaptation from previous works	RQ1	Informed arguments, literature	chapter 4
Institutional Analysis to create design context	RQ2	Institutional Analysis, Literature	chapter 5
Initial Design steps for concepts and ideas	RQ3	Creative methods, iterative brainstorming, literature for inspiration and kernel knowledge	chapter 6
Design Selection	RQ3	Informed arguments from alternatives and design concepts	chapter 6
Implementation of MVP	RQ4	Programming parts of the design and testing suite	chapter 6
Testing the implementation	RQ4	Run tests for speed, complexity and feasibility	chapter 7
Evaluating the design	RQ5	Informed arguments on successful fulfilment of requirements	chapter 7

Table 3.1: Steps and methods

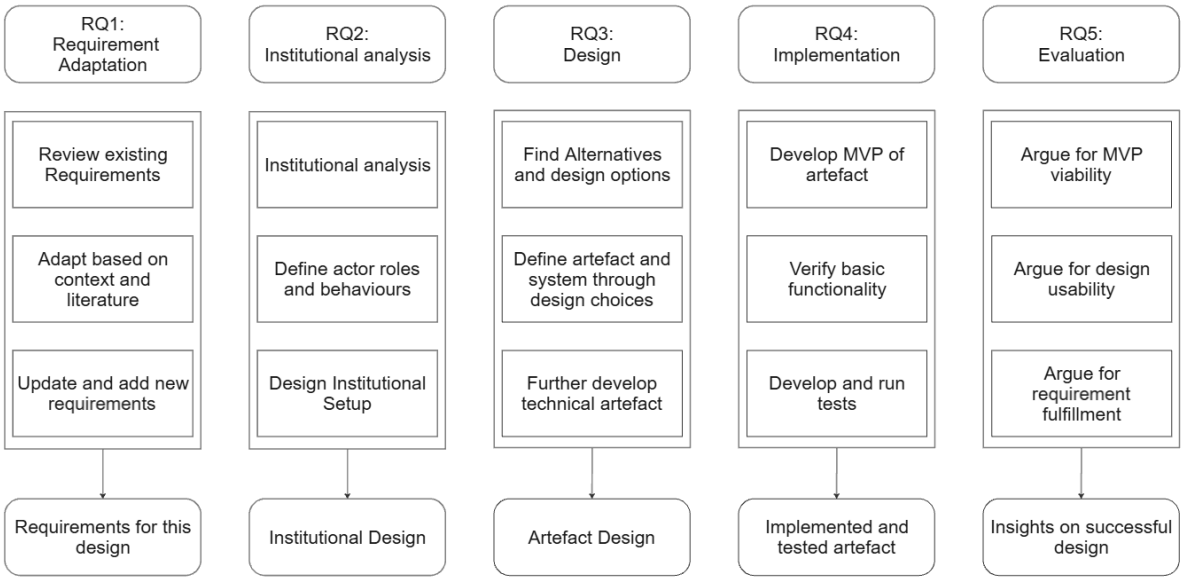

Figure 3.2: Research Diagram

4

Requirements

Defining requirements within the design science research method is going through the steps of outlining the artefact to be designed and eliciting requirements. Work has already been done on both these steps as part of the work by van de Walle, 2024 as a main part of the thesis, giving them high validity for potential re-use. The existing requirements will be analysed and a new and updated set will be argued for to guide the design.

The artefact to be designed is part of a broader system and partnership for Federated learning within the aviation sector. The artefact should tackle the problem of incentives and coordination surrounding incentives. The mechanism and method of governing and distributing these incentives within the system is the main focus of the artefact design, meaning the artefact will look like an IT infrastructure that fulfils roles regarding rewarding, control and organisation for FL processes.

4.1. Existing Requirements

The initial requirements found through key stakeholder interviews and exploratory work by van de Walle, 2024 address the needs and concerns for predictive maintenance in aviation using FL. These form a validated basis for the overarching architecture which needs to be rescoped for designing around incentives for FL. For each requirement, the scope fit and relevancy to this design problem are evaluated for this DSR. Additionally, the proposed solutions for these requirements are briefly touched on to validate that the proposal does not interfere with or otherwise influence the requirements for this incentive-based DSR. This first section is therefore heavily based on the requirements analysis and subsequent implementations and reflections on them done by de Walle.

4.1.1. Inclusion Criteria

Arguing for the inclusion of existing requirements means reviewing them against the background of their initial selection and scoping them to a better fit within the incentive context. The criteria used are threefold: is the requirement itself and the process and reasoning behind that requirement in the initial work relevant still for this context of incentivization design; can it actually be used as a requirement to shape the IT artefact; and is the work done to fulfil that requirement helpful in designing the incentivization systems. For each requirement, these criteria are used to decide if the requirement should be kept, adjusted or is not relevant to this design process. table 4.1 gives an overview of the existing requirements and the status within this design process, and from here the existing requirements by van de Walle are indicated as **rx**, where x is the requirement ID.

4.1.2. Requirement review

Security measures (**r1**) were an important and agreed-upon requirement, increasing trust and protecting the information. Security will always remain important in designs regarding data, especially in a regulated field such as aviation. When looking at rewards and organisation for this design, this requirement is still of high importance, although the purpose of the original design was to secure the

ID	Requirement	Status
r1	The system must have strict security measures.	Kept and adapted
r2	The system must have a reliable data-sharing framework.	Merged into r1
r3	The system must ensure compliance with existing legal standards	Kept as non-functional
r4	The system must be adaptable to evolving regulations.	Kept and adapted
r5	The system must develop a consortium-based approach for managing co-created IP	Out of scope
r6	The system must establish mechanisms for equitable benefit sharing.	Kept and adapted
r7	The system must develop transparent processes for value attribution.	Kept and adapted
r8	The system must be more accurate than current predictions.	Out of scope
r9	The system must ensure that AI models are explainable and transparent	Out of scope
r10	The system must have training programs for stakeholders	Out of scope
r11	The system must appoint a trusted, neutral entity for collaborative management and fair governance.	Kept and adapted

Table 4.1: Existing Requirements

aggregation and training. Adapting it for this artefact design is therefore a minor adaptation of the purpose, focusing security on the rewards and collaboration infrastructure. The original research fulfilled **r1** using various encryption methods for the data.

Reliable data sharing framework (**r2**) was again based on the high focus on security on data sharing for trust and active participation. Although relevant for the broader FL implementation, this requirement is too broadly scoped to adapt to this design. Furthermore, Walle proposes a consortium blockchain as a layer to store model metadata updates. That design choice is one of the bases of this design process, with this artefact being an expansion or alternative to that proposal. Trust in the protocols however is an important factor for participant acceptance and contribution, so reliability is added to r1 to provide a broadly fitting requirement.

Existing legal standards (**r3**) reign within the same context of aviation and provide boundaries for the design. Both regulators and participants are interested in compliance to make sure the system is legal and credible. Work by Walle resulted in a set of systems to control ongoing compliance and check the system against it, which could be adapted towards this artefact. The scope is still fitting and the requirement can, in accordance with the current state of literature and the following institutional analysis, as well as the original work, be adapted for this design.

The evolving regulations (**r4**) requirement, defined as a non-functional requirement, is an expansion of r3 based on the experts' expressed importance of monitoring and adapting to a changing regulatory landscape. Additional trust can be placed in the system if it is adaptable, and this requirement will be adapted as a functional requirement as a key aspect of the design. The proposed solution to this requirement was also similar to r3 and is also not of high concern to be worked around for this design process.

A consortium-based approach (**r5**) was selected as a requirement for fair IP management and for a governance structure within the consortium. The ownership and control of IP created from shared data was an issue pressed by aviation stakeholders, resulting in a requirement that aims to provide a structure that allows for all contributors to benefit and prevent monopolization of the resulting model. Although the importance of IP control is important, this requirement is a better fit for the higher level FL architecture and less for this specific rewarding infrastructure. Furthermore, the IP licensing method as a proposed solution gives adequate and out-of-scope protection. However, the importance of shared governance and control over the system is kept and adapted to different requirements.

Equitable benefit sharing (**r6**) is a key aspect of both the original framework architecture and this design goal as a contribution to the motivation of participants. A fair distribution that all participants agree on is a key aspect and fully within the scope and is kept as a requirement. The solutions proposed in the original thesis can also be the foundations of this design, with rewards being distributed as tokenized model access earned based on the contribution score, controlled with smart contracts.

Transparency for value attribution (**r7**) was set as a requirement for building trust and engagement by allowing participants to understand how and why they are rewarded. Again, a requirement key to the design of this artefact as well as the general architecture, is adapted as a requirement. Van de Walle proposed RFFL (Xu et al., 2022) as the algorithm running on a blockchain to calculate the contribution transparently, which could be one of the solutions for this design.

Accuracy for predictions (**r8**) is explicitly a requirement set for the entire FL system. Although more participants generally result in more accurate FL models and the goal of increasing trust is increasing participation, this requirement falls out of the scope of this particular design. The artefact should perform a key part in fulfilling this requirement as part of the broader architecture, however, but will not explicitly be built with this as a requirement.

Explainability and transparency for the models themselves (**r9**) are important for the aviation sector to understand biases, problems and reasons the model behaves in certain ways. This important requirement is based on the needed trust from the industry and regulatory approval but falls outside of the scope of the reward systems.

Training programs for stakeholders (**r10**) is a non-functional requirement towards programs to increase the knowledge and engagement of aviation stakeholders for the technical aspects of data gathering, security and distribution around FL. Necessary for any participant, knowledge of the FL workflows

and methods is also a key aspect of this design, being needed for any participant to actually use the designed artefact. However, enshrining this in the artefact itself is out of scope, as the design focuses mostly on the broader cooperation and technical aspects, and additionally, participants are expected to be supported by the support staff as part of adapting the design.

The need for a trusted entity for collaborative management and fair governance (**r11**) is a very broad requirement for ensuring trustworthy management and fair governance. This entity would control the parameter aggregation and promote trustworthy collaboration. A consortium was chosen as this entity by Walle to oversee the activities mediate disputes, and perform similarly to the decision-makers in the institutional design. Scoping away from the parameter control, the artefact should enshrine some part of the trustworthy governance surrounding rewarding and FL task orchestration.

4.2. Updating Requirements

The requirements kept from the original are appended with new design-specific requirements. These requirements are based on insights from the literature on trustworthy FL and institutional context to help solve the design problem found. The artefact designed through these requirements should result in a design that fulfils them, resulting in an artefact that can help increase trustworthy FL partnership participation. Per the theme of the requirement, the relevant requirements are formulated and explained based on the kept and adapted existing requirements, architecture solutions, literature and institutional insights gathered in the literature. The new requirements are shown in table 4.2, and use **Rx** as the ID. The purpose of the requirement is also provided to indicate the reason for the requirement being included, where relevant also adapted from van de Walle, 2024.

4.2.1. Security

R1: The system must have strict access security measures

Based on the original security requirements by van de Walle, this requirement expects the designed artefact to secure data and the models shared and created within the system. Adapted towards access regulation, making sure only those who should be able to access sensitive data and models, the design must ensure methods for securing and allowing access. Additionally, traceability of access and the ability to gain insight into the usage and behaviour within the system is also expected.

Purpose: To securely control access to models, training data and the FL tasks.

R7: The system must implement access control methods to FL tasks, information and models

Controlling access for participants towards the tasks, workflow information and intermediary models is important to retain data security and trust in the system. Additionally, according to Bi et al., 2024, central orchestrating parties need to be able to enforce (temporary) banning and otherwise control participation for the FL task. Therefore, this requirement was added to ensure the system creates methods for such access control.

Purpose: To protect FL tasks, information and models and keep participants in check

4.2.2. Regulations

R2: The system must ensure compliance with existing legal standards

Fully adapted from the existing requirements and still relevant, the legal standards themselves must be enshrined into the rules of the system, as the regulatory environment can halt the entire system if mistakes are made. This includes current and future versions of the EU AI Act, GDPR and the set of formal rules further to be defined in chapter 5. The requirement is adapted as a *non-functional* requirement, meaning the artefact itself should not have direct functions addressing this requirement, but still should stay within the regulatory confinements.

Purpose: To meet legal and regulatory requirements

R3: The system must be adaptable to evolving regulations

The ever-changing regulatory landscape and the international aspects of aviation, where regulations might be very different, indicate the need to stay compliant in the long run depending on the participants. Speedy adaptability in such situations is vital for the usage of FL and is, just like in the existing

ID	Requirement	Purpose	Based on
R1	The system must have strict access security measures	To securely access control to models, training data and the FL tasks	van de Walle, 2024
R2	*The system must ensure compliance with existing legal standards*	To meet legal and regulatory requirements	van de Walle, 2024
R3	The system must be adaptable to evolving regulations	To remain compliant with future legal changes	van de Walle, 2024
R4	The system must establish mechanisms for equitable rewards and benefit sharing	To maintain motivation and quality participation among participants	van de Walle, 2024
R5	The system must implement transparent processes for value attribution	To build trust and engagement among participants	van de Walle, 2024,Bi et al., 2024 & Inst. Insights
R6	The system must appoint a trusted, neutral controlling body	To ensure fair system governance and control FL tasks	van de Walle, 2024
R7	The system must implement access control methods to FL tasks, information and models	To protect FL tasks, information and models and keep participants in check	Bi et al., 2024, Inst. Insights
R8	The system must enable the sale of model access	To provide additional revenue streams and insight into model quality	Inst. Insights
R9	The system must establish methods of additional contribution-based governance	To promote self-correcting behaviour and promote engagement	Inst. Insights
R10	The system must provide modularity regarding contribution and rewarding methods	To adapt to new developments and changing behaviours and developments in the system	Tariq et al., 2024

Table 4.2: System Requirements

requirement, adapted for this design.

Purpose: To remain compliant with future legal changes

4.2.3. Incentives

R4: The system must establish mechanisms for equitable rewards and benefit sharing
As established in the problem statement, insights from the literature and Walle's thesis, benefit sharing is a key concept to incentivization and rewarding participation with the goal of maintaining contributions and participation. Keeping the rewards based on the contribution, the artefact can make sure participation levels remain high and of good quality.

Purpose: To maintain motivation and quality participation among participants

R5: The system must implement transparent processes for value attribution
Based on R4, value contribution-based rewards need transparency in the process for the participants to trust the rewards they are given. Additionally, Bi et al., 2024 also shows that insight into how others are contributing and are being rewarded helps build trust in the system.

Purpose: To build trust and engagement among participants

R8: The system must enable the sale of model access
Increasing the possibilities of revenue to increase rewards and incentives in the model can be built using external markets, balancing the price of model access according to the quality based on market demands. Adding an additional revenue source is similar to the Tokenization of the model for sale suggested in the revised framework by Walle's design. Adding market mechanisms to the system can help with improving the incentives and gives participants more dimension to which to understand the value of their shared work.

Purpose: To provide additional revenue streams and insight into model quality

4.2.4. Governance

R6: The system must appoint a trusted, neutral controlling body
Adapted from the a requirement by Walle, having a central authority trusted to ensure the FL processes occur in a trustworthy manner is also a key requirement. The role of such an entity is set as the decision-makers, having governance power in the collaboration and FL task decision-making.

Purpose: To ensure fair system governance and control of FL tasks

R9: The system must establish methods of additional contribution-based governance
In addition to the central authority from R6, participant governance outside those circles based on contributions and dedication to the collaboration provides more incentives to play fair and keep the system running well. Some power to the participants means they also feel some responsibility, helping the system perform more trustworthy.

Purpose: To promote self-correcting behaviour and promote engagement

4.2.5. Modularity

R10: The system must provide modularity regarding contribution and rewarding methods
As a highly active research field, designing concrete artefacts for FL that implement certain technologies and developments comes with risks of outdated or differently behaving systems. The multifaceted approach possible for FL in both design and reward and other dimensions is very high (Tariq et al., 2024), with no current convergence toward a specific design fit-for-purpose in aviation. To capture the long or medium-term dynamic changes of the system, the possibility of adapting the actual implementations both in and outside of the system is important. Modularity as a design concept is useful to utilize in an IT infrastructure artefact as often upgrades are possible that make key functions behave differently, in this case, the contribution and rewarding methods.

Purpose: To adapt to new research into best practices and changing behaviours and developments in the system

As part of the design process, requirements were made based mostly on the existing set of requirements with some additions made for the context of incentives. The resulting set of requirements captures the systems' need for security, compliance and adaptability to regulations, equitable rewarding based on transparent value attribution, governance through a neutral body with additional contribution-based inputs, access control for the tasks, workflow info and models and modular mechanisms for measuring contributions and rewards.

5

Institutional Analysis

This chapter contains the institutional analysis, introducing the framework used to find the factors influencing the behaviour actors for the system and finding fitting roles for certain action situations. The goal of the institutional analysis is to define and propose an institutional field and partnership type and the rules participants might have to follow in such a collaboration. An adapted framework will be used to expand upon the contextual features of the system found in the literature, as well as to define a playing field wherein the design will progress. The end result will be an overview of potential actors in and around the system, their behaviour regarding the system and each other.

5.1. Analysis Framework

Analysing institutional contexts and participant interactions within certain action situations can be done using the Institutional Analysis and Design (IAD) framework, which is adapted here for usage in FL for aviation. The work by Ostrom, 2011 worked towards the IAD framework for usage for governing the commons, a concept in economics about a type of resource that can have shared utilization and responsibility. Arguments have been made that distributed data sharing and learning, like FL, can be analysed similarly, with data itself being the shared resource that results in useful models using the IAD framework as the base of analysis (Hess and Ostrom, 2005). Further works suggests an adaptation of the IAD is can be a useful tool, with the Knowledge Commons (KC) framework (Sanfilippo et al., 2018) being a suggested adaptation of the IAD that focuses on Knowledge Commons - any kind of data or intellectual property that can be utilized to learn from in this context. This KC framework is similar in usage to the IAD Framework. The eight design principles Ostrom (Ostrom, 1990) found in successful common resource managements are also followed in the institutional setup: Clearly defined boundaries, fitting circumstances, participatory decision making, monitoring of the commons, gradual sanctions, easy conflict resolution, legitimacy and being networked.

The key features of both frameworks and design principles are combined to result in the analysis framework that is used to understand and describe an institutional setup for the rest of the design to function in. To guide the suggested design, parts of this model are used to infer the context and resulting playing field for potential participants of an FL partnership using insights from literature and context. The resulting framework that is used for this chapter is shown in figure 5.1, with the context consisting of the resource characteristics, participant characteristics and the formal rules the system needs to follow. This chapter will expand upon the action situations and how the found context features can be interpreted to form a dynamic playing field.

5.2. Context

The context wherein FL happens consists of the formal rules such as laws, the characteristics of the resource - data - and potential partnership designs. These contextual features shape the behaviour and design space of participants and components of the system by providing limits and directions of system dynamics. Research and background information has been gathered previously in chapter 2,

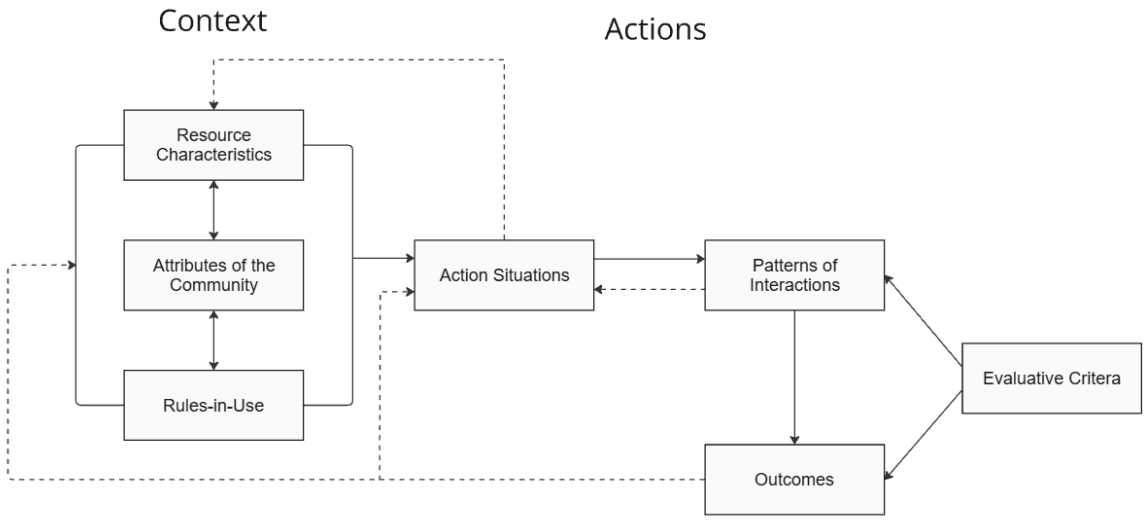

Figure 5.1: Institutional Analysis Framework

and is fit here in the Context part of the analysis framework.

5.2.1. Formal rules

As found in the literature, the formal rules surrounding FL in aviation are mostly focused on security, risk assessment, traceability and safety, and indicate that the formal rules firmly request compliance from the high-risk aviation sector. Especially in the EU, utilizing any kind of ML/AI, so including FL, generated models needs to follow risk assessments, and traceability of the model and data and the aviation industry must, naturally, remain within safety regulations.

The most important set of regulations and rules that impacts the usage of ML are the AI Act, which puts strong limits on the high-risk applications found in aviation and the GDPR putting limits on personal data sharing and general aviation regulations defining the security and safety of operations. These regulations limit completely free-for-all data sharing and the utilization of models generated by FL and put a responsibility on the ones creating and using ML/AI models. This points to a need to control the access to the models and potential participants so that abuse or unregulated data sharing is not allowed or punished.

A key concern is the regulatory field keeping up with the changing regulations. As the technology and regulations are in a permanent arms race with home team advantage to the technology, it can often be assumed that regulations have to always play catch up, resulting in certain uncertainties regarding the coming regulations. As adapted as a system requirement, being able to adapt key aspects of the FL systems and organisations to keep in line with the evolution of regulations should be an important aspect of dealing with the long-term changes to improve the longevity of the systems.

5.2.2. Resource Characteristics

As found in the literature, key resource characteristics revolve around data availability and transformation, IP of both data and the model, privacy of the contents and the value of the finalised model. These characteristics influence the behaviour and the design needed for the institutional context.

Preparing the data needed for FL brings high costs and is sensitive to attacks and malicious behaviours while often being a guarded IP or resource containing secret information. Adapting data to being able to be used is a process that can come with high costs and the need for technical know-how from potentially participating actors. This can be resolved with the utilization of support staff or external consulting parties, both coming with more costs. Risks are also associated with gathering the data and sharing the models trained on the data.

The Finalised Global Model (FGM) is the key resource that results from the work done in the FL tasks, but can also be sensitive to data leaks or illicit sharing. This model can be utilized in technical applica-

tions it was trained for, predicting or confirming expected behaviours in technical processes.

The behaviour of participating actors is affected by the characteristics of their data and the FGM. The high costs associated with transforming and preparing the data for the workflow make participating only worth it if the expected valuation of the FGM is higher than these costs. The risks associated with malicious behaviour, potential data leakage or an unsuccessful FL process where convergence is not reached might also be reasons to not participate. Such behaviour needs to be captured in partnerships to make sure participants are aware of the rules set to diminish such risks and encourage active participation.

Actor-based risks also include data manipulation and free riding, therefore malicious behaviour might be beneficial because of the characteristics of FL. Scenarios can occur where data manipulation or other attacks are beneficial to individual actors if control or access to the FGM is beneficial. Adversarial machine learning attacks can be utilized, and every development in IS brings data security issues in itself that need to be accounted for.

The value of participation in the FL partnership for actors comes from the application of the FGM and the marginal gains from utilizing said models. Economically the unit of value generated from participation in the FL task is the gains from the effective application of an FGM or any monetary reward from selling access, where the quality of the model is affected by the data provided by the participants. Additionally, benefits can be gathered from the goodwill of other participants, non-economic benefits such as sustainability improvements, or in-house research insights.

5.2.3. Partnerships

The current technical challenges of FL as well as the need for cooperation show the need for collaboration of the actors within an organisational structure. Not many fully formalised FL structures have been set up so far, so this section will propose further steps using a partnership known as a Data Collaborative for FL, as well as briefly discuss other options. This type of partnership will be expanded upon as the baseline institutional playing field for the design to work in.

Data collaboratives are broad partnerships between multiple types of organisations with the purpose using their respective data in a productive way. A data collaborative according to Zuziak et al., 2023 definition has the goal of performing at least one collaborative computing task, with the potential capability of tracking participants' contributions. The 4 key principles to classify such a partnership as a data collaborative are: the infrastructure for decentralised data storage, shared model governance, implementation-agnostic universality and the minimal utility to perform at least one analytical operation. These key principles align with the purposes of a partnership for FL, which needs a way of performing and distributing model training, a way of controlling the process through any means and needs to be successful at least once to be worth considering.

Such a data collaborative consists of participants who are willing to utilize their data towards a common goal that benefits all those participating, in this case the training of an FL model. Aspects of actor behaviour in this collaborative include their willingness to join, certain expectations on how the process will go and the usage of the final model. The need for the model, which steps they need to take to train and what kind of (economic) benefits the end result could result in influence how interested actors are in joining.

Governance of data collaboratives is an ongoing research area and actual implementations of the collaboratives rarely progress past their early stages. Bartolomucci and Leoni, 2024 propose an extensive governance model for such collaboratives. An alternative model is proposed by Ruijer, 2021 from a more public perspective. Both models contain governance structures that have some sort of controlling party as the governing instance in the collaborative.

Partnership types for FL that organise the participants is a research field in its own right, for this research, a data collaborative with a set number of participants is proposed and expanded upon further through this analysis. This setup will be used continuously for this design as it is both a fitting partnership design as well as one with active and current research on governance.

5.3. Action Situations

The action situation is a concept from the research frameworks where the actual activity happens and is the point of focus for the analysis. An action situation, influenced by the defined Context, is a situation where parties interact in a certain way regarding the resource, resulting in some kind of outcome. This method of analysis is used to contextualize certain activities. For this institutional analysis and to work towards an institutional setup, a few action situations are given related to the process of FL in aviation, from starting the partnership to training to finishing up the tasks. The action situation selected does not form a comprehensive overview of all situations but is a selection to provide enough behavioural insights for further development and design.

Joining the partnership is an action situation wherein potential participants who know about the partnership or are currently not actively participating might move towards active participation. This situation is influenced by the need for FL task workflow information to be shared, the potential sensitivity of said workflow contents, such as example data showing what kind of projects the proposer is working on, the potential rewards from participation and trust in the process itself: Are the models handled securely, is the system fair and are rewards distributed fairly? The result of this process is actors who are interested and except the resulting value to be higher than the cost turning into active participation. The effect of these action situations are fed back into the rest of the system, as lessons can be learned from why participants would not be interested in joining, with more participants also providing more benefits to all already existing participants as FL model quality is expected to rise with high-quality participants joining.

The data collection and preparation phase is a process that needs technical know-how and investments to find, transform and train on the data owned by the participant. Based on the workflow for a suggested FL task, from a research proposal or from another source, some data has to be gathered to be able to train models. This data gathering also includes data discovery, storage, transformation, and all other necessary steps based on the resource's characteristics. This process can be costly especially if the participating party has little work done on this before, although it can be accelerated by support from other parties or the collaboration as a whole. This action situation mostly has an impact on the field by raising the initial costs and barrier to entry for potential participants.

Training tasks are the core concepts of the FL process that guide the actual FL training itself and are used further in this research as the tentpole around which further design steps are made. To conceptualize the FL process for further design, the FL training process is modelled as separate Tasks with distinct steps: The initial proposal of the FL task, with certain expectations based on the goal, such as engine performance, and workflow information such as an example model and data set so potential participants know if they can join in the task. Such a proposed task can be kept internally or published externally for example. A proposed task can then be initialized if enough or a certain amount of participants have accepted joining the task, which then expects them to actually contribute as the FL training gets underway. The training rounds are where the actual FL happens: based on the horizontal centralized FL model, a basic example global model is transferred to the participants who can then do local training based on the model and their data, improving gradients and fine-tuning the network as much as possible based on their data. After a certain time or local accuracy is reached, the participants then share their local model back to the centralized server, where the global model is now adapted with the improved gradients from the local models that perform better for the task. These steps are iterated in rounds until the global model reaches a certain threshold for performance or some other end criterium such as time passed. Some incentive mechanisms calculate the added value the participant created per round. The wrap up then concludes the training Task, finalising the model and, if relevant, calculating the contributions of the participants. Depending on the rules set, the model and any rewards are distributed among participants.

System governance is where decisions are made regarding the workings of the partnership. Applied here in a simple way, the decision power is given to a privileged set of actors, such as the consortium in van de Walle's FL design, who can influence certain system elements such as limits to access to models or tasks, how and which tasks are selected for proposal and how rewards are calculated. These decisions are influenced by the context's formal rules, where it is also the responsibility of the governing parties to adapt to regulations. The actions the governing parties can take influence the field directly, both on the potential effectiveness of the partnership as well as the degree of participation.

5.4. Playing field

To further design insights, a conceptual design of a potential collaboration is drawn using insights from the context as well as the potential behaviour of actors within such a partnership. This playing field is based on the found institutional characteristics of the system and contains the description of potential actors and their roles, the rules they have to follow and their interactions for an FL process.

5.4.1. Actors

Actors in the FL system are broadly specified into multiple categories with their own roles and purposes in the action arenas. Each actor type is also assigned possible types of behaviour based on their potential goals. Table 5.1 compiles the roles and their descriptions as well as some examples of actors fulfilling such a role in the aviation sector. The goals and behaviours of each actor role are briefly

Role	Description	Example actors
Data Owner	Data creator who has the ability or intent to transform the data into usable formats for FL	Manufacturers, Airlines, OEMs, airports, data buyers.
Training Contributor	A data owner or other party that has access to data relevant to the active FL training and actively contributes using a specified workflow.	Data Owners that are willing to contribute and are active in the current FL task.
Model End-user	Any actor that uses the model for their purposes	Manufacturers, Airlines, OEMs, airports, research institutes.
Decision maker	Any actor given the power of decision-making in the collaborative with the purpose of removing, adding or otherwise modifying rules, training tasks and more.	Consortium members, Initializing parties, elected board.
Regulatory Body	An actor that has high-level control over the system and the rules	EASA, EUROCONTROL, IATA.
Supporting role	Specific actor that provide technical, implementation, data analytical or other support to the actors in the system	IT consultants, research groups.

Table 5.1: Roles and Participants

explained and are used to further define the rules the collaborative has to have to fulfil its goals of succeeding in FL tasks. For each role an example actor is described to help understand what kind of actor can fulfill such a role. All economic participants are expected to have the need to have the potential reward be higher than the cost to get that reward.

Data owners are aware of the data they have and the potential it has and want to generally use their resources for the best result. Knowing their data has value inclines them to utilize it if it is worth something, even though costs are involved. While purely economic actors want economic benefits, more altruistic actors might also want broader benefits through for example environmental improvements. Behaviours include deciding to join in the training tasks and pay the costs, or suggesting training tasks. Example data owners are parts manufacturers, airlines, often creators of their own data, as well as data buyers who can procure data from other parties to gain some economic benefit.

Training Contributors are those actively participating and want to get a return on the costs they are currently spending for training the model. They can help with this shared goal by providing and participating efficiently in the training tasks, or when maliciously inclined be lazy and not provide many useful tasks or even keep feeding back similar parameters. If the end result of model access is worth it, these behaviours are possible, to the detriment of the entire collaborative. Different types of behaviour can be modelled: In work by Zhao et al., 2024 participants are modelled as either Diligent, Inefficient or Malicious depending on their contribution using Shapley values, while Song et al., 2019 uses a Con-

tribution index. Bi et al., 2024 uses game theory to model the behaviours as either participating or defecting during rounds by not continuing to share their local model. These can be the same as the data owners, but with the intention of actually contributing.

Model End-users use the FGM for their own goals and research and might be willing to pay for it. Depending on the model, any actor, including external parties, might be interested in using it. Having information about what the model can do is essential for external parties, while internal parties who helped train the model should have an idea about its effectiveness. The actors that can fulfill this role are very broad and can include the same ones that are data owners, but can also be research institutes or policy makers.

Decision-makers in the partnership are those who hold power and have the incentive to change or keep the rules set. Behaviours are based on their goals, which can similarly be economic egoistic or altruistic. Their power is more complex as decisions can shape the entire system's behaviour, but generally, they can work towards cooperation or competition. If elected in good faith they would generally move towards either what is or will be a popular decision. In general, these actors are also the ones who have the power and mandate to propose FL Tasks and compile workflows. Examples of decision makers in an FL collaboration can be members of an FL consortium, or a subsection of them, or a separate elected board of actors within or outside the system.

Regulatory bodies enact their policies by influencing decision-making and the formal rules the system is bound by. They can be influenced and change their political direction, but generally move broadly in a constructive or restrictive direction regarding policy on AI/ML. These can include as an example known, broad governing organisations with soft or hard power such as EASA and EUROCONTROL, but also advisory or advocacy groups such as IATA that can ask for certain guarantees surrounding the usage or application of data and models.

Supporting roles are a specific group of actors that are important to the collaborative, but whose behaviours are not of direct influence to the system itself. This includes technical support, expert knowledge about utilizing or implementing training tasks, data gathering and transformation and other technical aspects of FL that an organisation might not be able to do themselves. Example actors include research groups or IT consultancy companies that work with the data owners and training contributors to help for their own or a broader scientific benefit.

5.4.2. Rules of participation

The designed partnership informs what kind of rules participants have to follow. These rules, partially using Ostrom's language for rules, are set interaction rules, between and from actors, and rewarding rules, for the system's behaviour towards participants. These rules can be used to provide basics for further research and as specific guidelines for the design for this thesis.

Interacting with the FL process for actors is set by rules about joining and leaving an FL training task, knowing how to perform the tasks and the ability to join in decision-making. An actor may join the training task and become a contributor if they follow the workflow to fulfil the current task and may leave when the FL tasks are deemed finished. The decision-makers, with the initializers of the current FL task, may decide when that is, as well as decide a selection procedure on who may join. Leaving before the task if finished, or not adding enough value according to set criteria will result in a penalty of some sort. Joining the decision-maker role is reserved for those decided by the initialisers of the collaborative and who prove later in the existence of the partnership to have a vested interest.

In the institutional design, rewards are distributed on successful FL tasks. What events these are and what the actual rewards consist of can be adjusted during the lifecycle of the partnership, but the design and behaviour of those processes fall outside of the scope of this design and research. The basic rule is that participants get rewarded based on performance. A part of the rewards will almost always be access to the FGM, as suggested by van de Walle, 2024 in the form of an access token. Additionally, some economic benefits should be spread equitably over all participants to incentivise participation through more than only model access. Selling access over the open market, as set as a system requirements, with the rewards flowing back into the system is also one of those rewards.

Rewards can also be non-economic for those participants more interested in providing insights or sup-

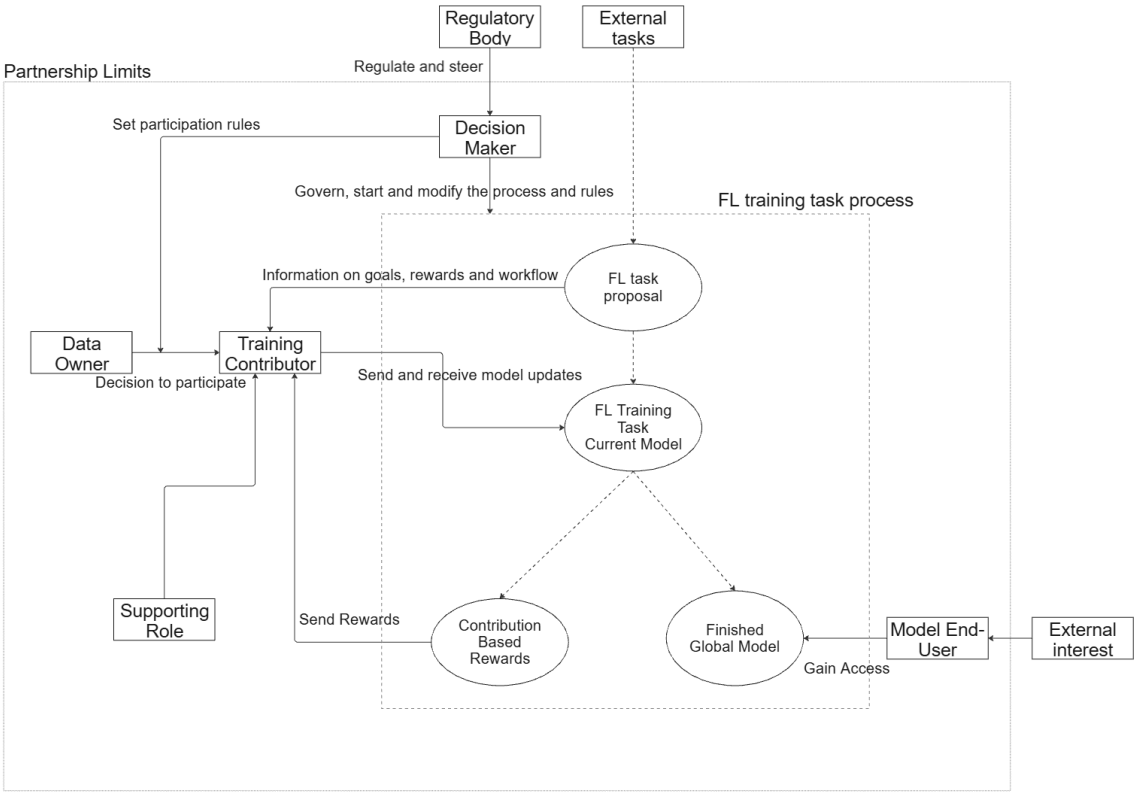

Figure 5.2: Institutional Field

porting for example the sustainable development goals. These actors are interested in the utility and knowledge of the usefulness of the FGM. As research and experience regarding the behaviour and incentivization of participants evolve over time, the ability to change the reward schemes becomes an important aspect of the partnership: Having to reform and re-negotiate the entire system under such developments is more resource-intensive than adjusting certain aspects.

5.4.3. Institutional Field

Based on the found and suggested roles and behaviour, informed further by Ostrom's design principles, a conceptual institutional playing field is suggested that fits within the set requirements: within this context, the rest of the design process will happen. The partnership design is based on the key roles of the FL process as well as how they interact with each other and with key activities that FL consists of. A centralized FL architecture fits best to keep control over the rules and reduce overhead by providing a central server that needs controlling and oversight, although hierarchical and similar non-decentralized FL architectures could also fit within this field. The external field consists, besides the regulatory bodies, of potentially interested external users that might turn into end-users of the FGM and external potential FL tasks that might come from outside the system.

Ostrom's design principles used to help make sure the data resources existent in the system are used efficiently are reflected in the proposed institutional designs by applying them to the actors and properties of FL. The system boundaries are defined by having a pre-selected group of data owners who can potentially participate in the training. The rules are specifically made for FL in aviation, with strict formal rules in mind with no compromises on security. Making decision-making participatory is vital according to Ostrom, so a possibility for anyone to ascend to the decision-maker role was included. Monitoring the process needs to be transparent, and reward mechanisms can act as a soft-sanctioning method. As a closed system with clear rules, conflict resolution can be arranged quickly with short lines and fast decisions. The partnership exists explicitly within the formal rules set for AI/ML and aviation guidelines, and internal governance is legitimised by the consortium owning the partnership.

This institutional field serves as an indication of how actors interact with each other and the FL system within the context of a partnership. Aspects of the field and the behaviour of actors can be seen infigure 5.2 as a visual overview.

Decision makers control both the FL task selection and initialization, additional rules and control come from these actors about allowing interested parties to join as contributors and about reward distribution. The role is powerful and enshrined also in the consortium defined by van de Walle, 2024 to, besides having IP ownership, make decisions about the long-term behaviour and rules of the partnership. The decision maker can select FL tasks to start, propose new ones, and change the rules of participation and how the contributions and rewards are calculated and distributed. Additionally, governance is done by setting limits on model access and complying with regulations regarding the models. If for example a model is found to be biased, it is the responsibility of the decision makers to adjust access rules. On a higher level, the regulatory body and the legal framework control the decisions the decision maker has to make, as well as set rules where models can be used.

5.5. Example scenario

To help explain the playing field and the situation in which the design is placed, an example scenario of a potential real-world application of FL is shown here. This scenario contains influence from every role within the playing field and considers the forming and basic activities of a partnership for FL. Within that scenario, the action situations are explained and the activities are shown. The scenario is plausible, but not verified as an actual potential situation.

A group of organisations in the aviation sector decide to collaborate on an FL project concerning temperature fluctuations on the wings of a specific aircraft model during landing. The manufacturer of that model is interested in understanding the impact on the materials and has decided to initialize an FL process. With 4 airliners that utilize this plane, the manufacturer agrees to set up a collaboration for the duration of this FL process to organise the process. These 5 participants agree to share their role as decision makers with equal say in the process and agreements. Internal research then finds more data owners in a university research group and a data broker, who agree to participate as potential contributors and are accepted by all decision makers.

These 7 Training contributors agree to join a proposed FL process with a set of expectations and restrictions: The decision makers propose, with the help of an IT consultancy firm specializing in ML, a task with appropriate workflows to find a model that can help understand wing temperatures. This FL task proposal contains an example model and dataset information such as parameters and dimensions. This proposal is shared with all the participants and also includes an agreed-upon reward structure based on contribution. One participant however secretly wants to not contribute because of the costs and only is interested in the end result of the process.

The FL task can now begin, and sensor data from the planes in use and found is used to train and improve local models based on the shared example model. The updated parameters are shared with the global model, which is updated to the best-performing version. These rounds continue until a predefined performance threshold is found where the model does not perform meaningfully better. For this scenario, the contribution score is calculated through Shapley values, and at the end of the training, every participant gets a score based on that value. Initial rewards are then distributed. The malicious actor, which has only fed in slightly adjusted values based on random noise or limited data, is found out through this value and cut from the rewards.

The final model is then set and distributed according to the agreements, either through tokenised access points or wholesale. The manufacturer wants to use the FGM to test new wing materials, while some insights from the model show the airlines that a different angle of approach during cold weather can improve behaviour, which can help pilots. An extra model end user is found when after publication of the potential use cases, a different manufacturer purchases model rights to help improve their material usage. They purchase the model for an amount which is distributed according to every training contributor's contribution percentage.

The usage of the model is controlled by regulatory bodies, who are invested in the model when it is applied to planes, as some biases or risks can exist. A regulatory agency that assists in testing

aviation safety tests the findings of the model and asks the training contributors about how the data was gathered. Additionally, they suggest the consortium add certain rules on who can join the system.

This example scenario ends up with actors performing their indicated roles and will be used to show behaviour as the design takes shape. The institutional design built around the context for certain action arenas helps with providing a baseline for technical design for incentives in FL. The insights gained from the analysis revolve mostly around the need to adapt to the changing regulatory and technical landscape, while the need for a trustworthy organisational layer is required to fulfil both the task management as well as the rewarding aspect.

6

Design

The design goal of this thesis is for an artefact in an IT infrastructure that allows for rewarding, controlling and organising Federated Learning. The design process within design science research consists of generating ideas and iteratively selecting and assessing these potential designs. Idea generation was based on real-life implementations of similar systems and broad knowledge based on FL, incentives, and orchestration. Important steps to stay within the scope are defining explicitly what is being designed to fulfil the requirements and defining clear assumptions that delineate the design edges. This chapter reports on almost the entire process, starting with alternative generation and assessments against the requirements, the sketches and initial design thoughts, moving into the designed artefact itself with detailed explanations about the parts.

6.1. Design Process

The design science research body of knowledge (Johannesson and Perjons, 2021) provides tools and guidelines on how to move from initial ideas to a designed artefact in 4 sub-activities: Imagine and Brainstorm, Assess and Select, Sketch and Build, and Justify and Reflect. Idea generation was performed by iteratively brainstorming and drafting using broad knowledge bases regarding FL and trustworthy organizing systems, as well as incentive design for FL done by other research. The assessment process was partially an iterative approach, reflecting on initially drafted design alternatives and their fit for the system and requirements, updating ideas continuously, and subjective alternative selection procedure. Evaluation of iterations and the design were also tested with some of the trust evaluation criteria in FL by Tariq et al., 2024, with a focus on incentive mechanisms and contribution evaluation, but also robustness, privacy, client selection, and accountability. Finalising iterations on the design were based on informal feedback and reflections on the artefact's ability to fulfil the requirements.

The baseline and background for the design were based on the initial framework by van de Walle, 2024 and the institutional design fromchapter 5 as the setup of the actor field, with rules and design governed by a trusted central party. Alternatives were based on different potential design choices with their own set of pros and cons, that could be built together in a design.

6.1.1. Alternative generation

By brainstorming, whiteboarding and general creative processes, both alternative solutions and potential designs were generated that would fulfil the functions needed to fulfil the requirements as a cohesive design of the artefact fit to help solve the design problem. Knowledge of the topic area found through literature and author knowledge was utilized to inspire and shape the alternatives, with some aspects of the design having a broader design space than others. Although the base design was shaped to fit within the FL architecture by van de Walle, 2024 and around the institutional design, alternatives that did not fit within these ideas were also considered to promote creative solutions. The most promising potential alternatives on key aspects of rewarding, control and infrastructures are reported and discussed in this subsection, as well as some issues with them.

Reward Alternatives

Rewarding participants is a key incentive design that should function as one of the most important aspects of the artefact. These rewards can come in many forms and multiple methods were considered based on three distinct categories: Monetary or other economic gain, model access under the assumption the model is of value and altruistic, where participants are interested in the general good the model can do.

Economic gains can be expressed purely as monetary gains by providing contributors in the system with a direct reward. This could come from a pool of money collected by the consortium in the partnership, from sponsorships or grants or from investments. Reward payments were considered as a direct deposit for initial participation, payments per round and based on contribution. Compensation for organisation size and data availability as well as the possibility to adjust rewards based on previous contributions and behaviour were also considered.

Direct issues with providing purely monetary rewards are first of al the need for these rewards to be higher than the costs, which cannot be guaranteed and runs into problems when the initial costs vary per participant depending on how much investments are needed to start performing FL. Additionally, the money needs to come from somewhere and, with no guaranteed cash flows for the collaborative, a negative feedback loop can occur with no money coming in to incentivize participation, resulting in even less money coming in as the performance drops.

Rewarding participants with model access is a suggested option that gives participants the right to the utilization of the model. As a singular reward scheme, this is only interesting in the situation where the model is of use to the interested parties. Contributing participants who supply local models are expected to have use of the model trained partially on their data, but no guarantees are expected for usefulness for their specific organisation or use case. Access can be granted as one-time uses, for example using access tokens, allowing a specific amount of hours of access to the model or simply giving the entire model and its parameters to all participants.

Utilizing access as a reward implies the exclusion of the final model. In principle, controlling model access is doable using various access control methods, where based on whatever alternative is chosen, rules and policies can be applied to provide access to the model when needed. Critically, however, fully controlling who has access to the model is made difficult through participants training local models based partially on the global model, and can therefore have a version of the model that, even though it might be of lower accuracy than the FGM, can still be utilized without going through the access methods.

A reward scheme purely based on model access expects the model to be of high quality, useability and performance, else the reward has little use. In such cases, without high expectations about the FL process and the resulting model, participation can remain low as the resulting incentives are lower than the expected costs.

Altruistic stakeholders can be interested in rewards that are not purely monetary but provide broader value to the world such as environmental gains, passenger satisfaction or safety. Even though these utilizations of the FGM also provide financial gains, having a focus on these general improvements is not uncommon in certain organisations that also have the opportunity and data to contribute to FL. For some, having just the expectation of providing resources for FL for 'the greater good' can be seen as no reward for more economic participants. Additionally, the practical implementation of such rewards should include a method of proving the entire process actually provided the expected or promised benefits, needing some kind of verification of the results in the real world. Such difficulties and increased costs to prove efficacy further make a solitary use of non-financial rewards unlikely to succeed in a competitive sector such as aviation.

Rules and control Alternatives

Trustworthy rules that needed to be implemented had design space within contracts and agreements, fees and fines, and technologies such as using smart contracts and cryptographic proofs. The design space was explored in these categories of methods useful in steering actor behaviour.

Contracts and agreements as classic methods of controlling actors are workable methods that will always have a place in partnerships between organisations. Contracts would be set about behaviour and rewarding participation, setting the rules based on negotiations between the parties. For the system

itself, having contracts will always be the basis of collaboration and setting up the initial partnership. However, contracts are inflexible and take time to be adjusted to changing situations and developments, resulting in the rules set within contracts being looser than might be necessary.

Controlling behaviour using fees, fines and other material or immaterial reward or punishment methods is in economics and policy creation a frequently used method. Using such encouragement of proper behaviour in a system can be used to make sure participants do or do not fulfil their expected roles. The effectiveness of such methods depends on the impact it has on the participant: If the value of for example a fine is a fraction of invested costs or total turnover an organisation might not care at all about how the behaviour impacts the system or is viewed as internally and externally. Malicious behaviour might continue, or the rewards might not be significant enough to encourage participation.

Technological advances have allowed for enshrining rules and behaviours in trustworthy data stacks that can be implemented and verified by any participant. Specifically, smart contracts were analysed as potential solutions to help with verifying and trusting the actions of participants, an important aspect of control and trust in the system.

Smart contracts were one of the suggested solutions in the architecture of van de Walle, 2024, controlling important aspects of the entire FL organisation. Smart contracts are useful for automatically controlling or documenting actions and behaviours of actors and are very useful for control in a trustworthy and transparent way. Issues occur on utilizing smart contracts in certain applications depending on implementations: mistakes in the code can introduce problems and unintended behaviours that continue automatically until the smart contract is fixed. Additionally, although smart contract transparency is seen as a benefit, not all data and information can be made public in such a way that anyone can read it. Important to note is how almost all smart contract implementations rely on a blockchain as the immutable ledger to store the contract rules and apply the results of the contract rules.

Infrastructure Alternatives

On the broader infrastructure side of the to-be-designed artefact, alternatives were gathered for implementations using traditional architectures, secure distributed computing environments and blockchain. These alternatives were analysed for their usefulness in securely organising the FL process and reward schemes, and how to contain the other design choices.

Making an FL design within a traditional server architecture would entail providing a fully centralized design where the parameter server and controlling parties would also organise the workflow sharing, participant registration and other orchestration services. Having a central location for the global model is generally part of the FL architecture, except in decentralized designs, but the orchestration aspect can be designed differently. Using central architecture relies on the full trust of the administrator of that server, and the single point of failure makes it a potentially risky design. However, the method is easiest to implement and can reduce overhead, as it is already part of most FL architectures anyway.

Using secure distributed computing environments on different servers or cloud solutions to control when and where participants can gain access to critical information or data such as workflows or sandboxes is a useful method for both FL training locations as well as orchestration services.

Blockchain is an important part of the initial proposal, utilizing its key features of immutability and transparency to provide a trusted ledger of transactions and control rules and decisions made for all participants. Various implementations and methods exist for building the system on-chain, including public or private chains, hidden transactions and bespoke chains or using existing implementations such as the EVM or as Solana applications. The design choices to be considered here are utilizing a public, private or a consortium chain, or even considering not utilizing an EVM chain or a blockchain at all.

A public chain has the benefit of being accessible without any special needs and has an immediate broad appeal, but comes at the costs of high transaction costs, hesitancy regarding privacy, and the inability to intervene in the chain if something goes wrong the governing parties want to fix. Private and consortium chains have many overlapping features, where access is only granted to a select few. The key difference is how the chain is validated, where in consortium chains validation power is explicitly shared among the members, whereas in private chains these *may* be run by one actor allowing for easier chain reorganisation.

Even though some of the computation can be done on-chain, a blockchain can on occasion be nothing more than a slow database, so only necessary features should be put on chain. Computation is slow, so unless a custom blockchain can deal with the complex computations of FL, off-chain applications will always be a part of the system. However, smart contracts without a blockchain controlling the outcome and results are rare and not deemed very useful.

6.1.2. Alternative Assessments

To make design choices, the alternative methods and implementations were assessed based on the requirements. For each functional requirement, so excluding the non-functional R2 about regulatory compliance, the potential success of the alternatives found in the previous subsection was argued for how well it would aid in fulfilling that functionality. For each requirement, relevant alternatives that might be used in any combination are commented on how well they would fulfil the requirement and how they could inform the design.

R1 Access Security

Fulfilling this requirement needs solutions that securely control access to the models, training data and FL tasks within the system, providing interesting solutions within the control and infrastructure alternatives.

Smart contracts that control the access based on the set rules on a mostly immutable blockchain provide a useful design space when combined with some distributed computing. Smart contracts can be deployed to control the rules about who can access the models or the published FL Tasks, and can be used to verify the access or right of a signer's wallet according to the set rules. If the models or task info are put onto one secure server, it introduces a single point of failure but is a fitting architecture where the immutable chain can verify if a signer has access. Additionally, the blockchain setup needs to be secure against attacks. A public chain is the most secure as its decentralized nature is very secure, while a private chain can be vulnerable if enough validator nodes are attacked.

R3 Adaptable to regulations

Remaining compliant with future legal changes by being adaptable to regulations is important as an alternative, and can be fulfilled using technological or organisational choices.

Adaptability is not a simple implementation, but promising options exist in adjustable contracts that can change how the system behaves. Adjusting any part of a system based on trust on the fly inherently comes with some concerns, even though the adjustments might be based on compliance. Utilizing upgradable smart contracts on chain is a technical solution that works, but needs agreement by participants. Additionally, smart contracts might be an implementation risk in some scenarios as they are not legally binding or within standard regulations, increasing the risk when mistakes are made in the code.

Some aspects of potential rewards and incentives are harder to adapt, as it might not be possible to recall rewards or access tokens. If the model generated is deemed to be unusable due to regulatory concerns, the worth of the model also drops to zero, damaging the economic benefits and reputation of the system. These are inherent problems with having to be adaptable to regulations, but if the rewards are split per task and the general benefits are high enough, both adaptability to regulations as high enough economic benefits can be sustained.

R4 Equitable rewarding and benefit sharing

The rewarding and benefit sharing in an equitable way must be part of the system to build trust and engagement among participants, which the alternatives around rewarding and control are useable for.

Monetary rewards that can be controlled using smart contracts are useful to design areas for benefit sharing, while model access, if the model is of high quality, is also valuable. A direct deposit for services rendered in the FL process is an option but needs liquidity from the partnership themselves, which is not guaranteed to be available. Additionally, even though the monetary rewards can be fair, in high-cost organisations the amount gained might not be worth the resources spent as the sizes could be diminishing. An alternative is slashing rewards based on misbehaviour, which could also be implemented with decent expected results.

Rewards in the form of model access are an inherent part of the FL system, where participants get guaranteed model access, potentially up to a certain amount. Providing equitable rewards this way can also be done through leasing access or wholesale selling of the FGM, which, if the model is good, gets a bigger reward as well. If these reward shares are distributed fairly based on some of the many existing algorithms that define a contribution score or Shapley Value, such rewards are equitable. Smart contracts could be a key solution in this space as the sharing and rewarding can be organised in an auditable and transparent way, increasing trust in fairness.

R5 Transparent value attribution

Value attribution in a transparent way is important for trust and engagement in FL, so the alternatives assessed on this requirement include those that increase transparency.

The highest transparency can be found in utilizing smart contracts that control the value attributions. Smart contracts are able to verifiably store and manage the added value from the participants in the tasks. The calculations needed for value attribution is very high, and having that on-chain needs a high-performance chain, as often blockchains are not fast enough. Many attribution calculations exist, so having a way of doing multiple is useful.

R6 Trusted, neutral controlling body

Governance is a key concept and having a trusted and neutral body in control helps with fair governance in addition to controlling FL tasks. Already set up within the institutional design, adding certain technological design choices should help those in charge with their ability to enact changes.

Classic contracts give the likeliest best design space for helping assign such roles and responsibilities while utilizing a blockchain-based infrastructure can put the responsibility and control amount varied parties as verifying nodes for a blockchain or applications on the chain.

R7 Access Control Methods

Access control methods need to be able to protect FL tasks, and information and keep control over the participants.

Smart contracts are scored highly for their ability to automatically set and check for access rights in a broad environment. This method is both transparent and immutable, giving participants insights into what policies are enacted. These can work in tandem with a central server hosting setup where a service is only run when certain on-chain conditions are met for the signer trying to log in. Basic passwording systems can also function as an alternative for accessing the tasks or the network but are less well suited for transparently keeping track of which actor is allowed access.

R8 Model access sale

Enabling model access sales gives additional revenue streams and acts as a method of giving insight in model quality through market dynamics.

Methods of model access include tokenizing access runtime distinct access privileges, or unlimited access. Controlling access to the model is important to be able to sell the model, which itself should be stored in a private secure server or cloud. Utilizing model access through smart contracts on the blockchain very highly and having some contractual agreements in writing for IP and usage of the model also is a potential design choice.

R9 Contribution-based governance

The promotion of self-correcting behaviour by giving the participants some extra responsibilities through contribution-based governance methods can be implemented in the context of the key alternatives.

Smart contracts are scored as very useful as they are adjustable, controlling aspects that can enshrine voting and governance. On-chain voting and governance is a common practice in public blockchain projects and has shown some potential in useful behaviours. Alternatively, giving more voting power or a place as a governing decision-maker for consistent high-performing participants is an option that requires less technical overhead and can be more informal or enshrined in founding contracts.

R10 Modularity

Allowing changing contributions and rewarding methods is important to adapt to new developments in the sector and respond to the changing behaviours of the actors involved.

Multiple methods exist to fulfil this functionality, with traditional contracts being quite slow and inflexible, smart contracts being adaptable but with some concerns about trust if they get upgraded and an agnostic approach to the infrastructure as most modern IT architectures can deal with modularity.

Alternative assessment conclusions

The key alternatives that seemed the most promising were in the area similar to the architecture design of van de Walle, 2024, utilizing smart contracts and blockchain. Supplemented by traditional contracts using any kind of IT infrastructure supplemented by a blockchain to govern and control both access and rewards seemed to point to promising design space.

Assessment of the design space in three main areas of rewarding, control and infrastructure pointed to multiple promising methods of fulfilling the requirements within the context of the FL system field from chapter 5. Further steps were to make combinations and sketches of these broad alternatives and specify the promising ones further to shape an artefact design.

6.1.3. Sketching and Iteration

Various combinations of design options were used and sketched up using the digital imaging and whiteboarding platform Miro. Iteration and further developments were performed using unspecified criteria and informal feedback relevant to the design. After some open-ended rounds of iteration, the basic design was reached based on the institutional design from figure 5.2 and the direction taken by van de Walle, 2024, utilizing a blockchain-based system with tokens as contribution-based rewards that can be used for governance, model access, selling model access and getting kickbacks from model sales.

The conceptual and architectural design, based on the FL for aviation framework and the institutional design, converged on a design after two iterations. Explicit choices were made on having active participants be rewarded *and* have an influence on decision-making processes as both an incentive mechanism and a way of encouraging participants to provide maximum value to the FL system.

6.1.4. Reflection and Decisions

Justifying the selection and design of this design needed to be done for certain aspects of the design, after which further detailing of the artefact's implementation could progress.

First of all, the baseline design of Walle and the institutional field that points to a blockchain-based system with tokenized rewards is a broadly used method of contribution and interaction in public cryptocurrency projects. Inspiration was taken from Distributed Autonomous Organisations (DAOs) that, often in public cryptocurrency projects, govern themselves using reward tokens that are used for voting and can provide value. The case for the usage of BC in the first place when, after all, a simple database could also work, is argued for by it being necessary for smart contract functionality, having high scores in the alternative assessments against the requirements and fitting in the architecture by Walle. The institutional design shaped the artefact as one with defined roles and rules while using Majeed et al., 2023 as the key inspiration for using tokens as rewarding mechanisms, although used in a private blockchain instead. BC was compared with only formal rules between actors organized with a standard server as coordination architecture, which is worse at checking the clients and transparency in the process. Blockchain-based smart contracts remained the main method of the design because it is an oft-used solution for key requirements on transparent and trustworthy processes (Rejeb et al., 2021) and trustworthy security (Y. Ma et al., 2020) and access control for FL in general (C. Ma et al., 2022). Convergence was reached on the detailed technical solutions and architecture after further iteration rounds.

6.1.5. Design Choices

Finalising the design, each design choice made and the implications on the broader system are explained. Key concepts are introduced here with the reasoning behind those design decisions based on the analysis of alternatives the idea generation phase and the context of institutional design and requirements.

The context of the design is limited to a set of smart contracts, the control of these contracts and their interactions with a black-box periphery where the actual FL happens. Additionally, the calculations of the contribution value are also done outside of the directly designed scope. This decision means the design scope remains small while the artefact will still play an important role in the FL process, but some control over the system's ability to interact with the periphery relies on out-of-scope designs. The FL calculations and hosting of the model done outside of the context of the on-chain system and contracts, modelling it as the periphery, expanded on later.

Smart contracts on a private blockchain are the key IS parts of the design that control the FL task registration, participation and rewarding the participants. Additionally, it is used as a check for access, being used to verify if a certain participant is allowed to access information such as workflow or an FGM. The choice for smart contracts means that much of the control flow is set up transparently, with all the policies set in a way participants can see, and also semi-immutable, with the contracts and blockchain state being leading unless some extreme measure is taken by the validators or through contract changes.

The choice for a private blockchain results in a permissioned chain that only a select group of participants can have access to, with the node validators that run the chain being the responsibility of the governing parties like the consortium members and decision makers. This means the blockchain is robust as long as enough participants are aligned, and the chain will remain an immutable ledger of relevant transactions as long as all validators behave.

Upgradability and adjustment are enabled with the ability to be upgraded after deployment of the smart contracts and usage of the proxy contract pattern, which allows easy and flexible upgrade patterns while keeping the smart contracts running. This explicit design choice is meant to capture the changing environment, allowing for changes, but has some impact on the trustworthiness by being less rigid. A certain reward mechanism or contribution calculation method can be found better or worse, and implementing that into the system must be a well-thought-out choice during the lifespan of the partnership.

Tokens were a design choice that captured rewards, governance and model access. Reward tokens are chosen as part of the rewarding mechanism, being generated based on the contribution of the participant. The decision to make the rewards purely a token means that these tokens represent the contribution to the model, but also as a method of measuring contribution to the partnership in general. Having the tokens be the exclusive initial reward mechanism does introduce some challenges, as no direct monetary rewards are distributed. The reward tokens will act as a method of governance, giving some governing power based on the contribution. Having the reward tokens also function as some sort of stake in the system means that the incentive to contribute well is appended by the ability to also adjust the governance direction of the system. Additionally, the reward token can be used for profit distribution from model sales.

Access to the model is put behind a token that provides access to using the model. The access token represents the ability to use the model, which has the effect that model access can be transferred and tracked on a per-token basis. This also allows the token, represented on the private chain, to be transferred to a public chain or other markets where the tokens can be sold. This does however impact the level of access to the FGM the contributors get, as they also might have to use the model, or can get infinite access.

Governance design choices are made around introducing extra voting power to the ones contributing by giving the reward tokens extra capabilities in the form of voting power in certain decisions. The reward tokens can be used to provide input in governance decisions ranging from FL tasks, reward mechanisms and more that can be decided by the partnership. This does create some complications regarding governance capabilities where decisions are influenced by more participants.

A design that relies on multiple tokens as the reward structure that both enables model selling and extra governance power based on contributions is a promising design direction that will be expanded upon to indicate how well it can fulfil its goals. The next section will provide insight into the actual further design.

6.2. Design

The artefact is a collection of smart contracts on a private blockchain, with the decision-makers and owners of the network being the validators using proof of Authority. The smart contracts control the rules set for the participants and allow for the system to organize FL and the rewarding of the participants based on their contribution. The artifact fits within the existing and potential future systems as the organizing layer for the partnership, and servers as a modular interaction point that controls and registers organizational and co-operational aspects of the FL tasks and process. The final converged design after the process was described in a conceptual model of the broader system figure 6.1. This design is explained in sections on its functions, infrastructure, the smart contracts it exists of, the reward systems and the peripheral aspects to interact with the external systems. The technical setup with the contracts and the necessary functions both in and outside the system is presented in figure 6.2. The explanation sections include how the design aspect fulfils the requirements found in table 4.2 and potential alternative design choices that were not made but were found most promising.

6.2.1. Infrastructure

The key infrastructural design choices resulted in a setup of a fully **private blockchain** validated by trusted **consortium validators**. Key components defining the infrastructure are the Private BC network and the consortium validators. The consortium validators are the controllers of the blockchain using a validation method known as Proof of Authority (PoA), where transactions and blocks on the blockchain are validated by trusted and approved nodes. As long as enough of the nodes themselves remain secure, the integrity of the blockchain is secured. The validators are controlled by known participants of the consortium, making sure all consortium members have an equal say and ability to validate the network and are able to be held accountable if a validator misbehaves by adjusting or stopping transactions. For general PoA implementations, at least 4 trustworthy validators are needed to ensure chain integrity.

The components Private BC network and the Consortium validators address specific requirements regarding security and transparency. This infrastructure allows for strict access control and security measures (**R1, R7**) by providing a blockchain network that is fully controlled by trusted parties, enabling full control over access and rights within the network. Additionally, the inherent quality of BC as a readable ledger provides transparency (**R5**) for any and all implementations running over the network. Additionally, the responsibilities of the validators fulfil the role of implementation for the controlling body (**R6**), ensuring integrity.

Alternative solutions include a public chain such as Ethereum or Solana being the main infrastructure. This design option adds potential participants and publicity and has a more developed chain security and development team, but actively hinders privacy concerns and creates overhead, introducing gas costs and becoming a highly interesting and public target for attacks.

6.2.2. Smart Contracts

A set of six smart contracts was designed, with three of them being functional and providing key functionality for orchestration, rewarding, and governance, two implementations of Tokens, and a proxy contract controlling the upgradeability. The key functionality of the contracts are described, with their interactions, and how they fulfil requirements.

FL Orchestrator

The orchestration contract controls and monitors the state of the FL tasks run by the system, including the participants, the rounds, and potential metadata, and can initiate and run FL tasks from multiple sources. Important functions of the FL orchestrator contract are *proposing an FL task* with a set of participants and information around the task, *initializing an FL task* that's been proposed after participants had time to register, *starting an FL task* and *updating FL task status* through round updates, and *finishing an FL task* manually or automatically, based on accuracy, time or elapsed rounds, triggering the **FL Rewarder** with the appropriate task information. Only permissioned actors are allowed to do manual updates or proposals.

The orchestrator does not host the FL information itself but rather the identifier of the FL task, which can be linked to the workflow information. The Orchestrator control the state so any implementation

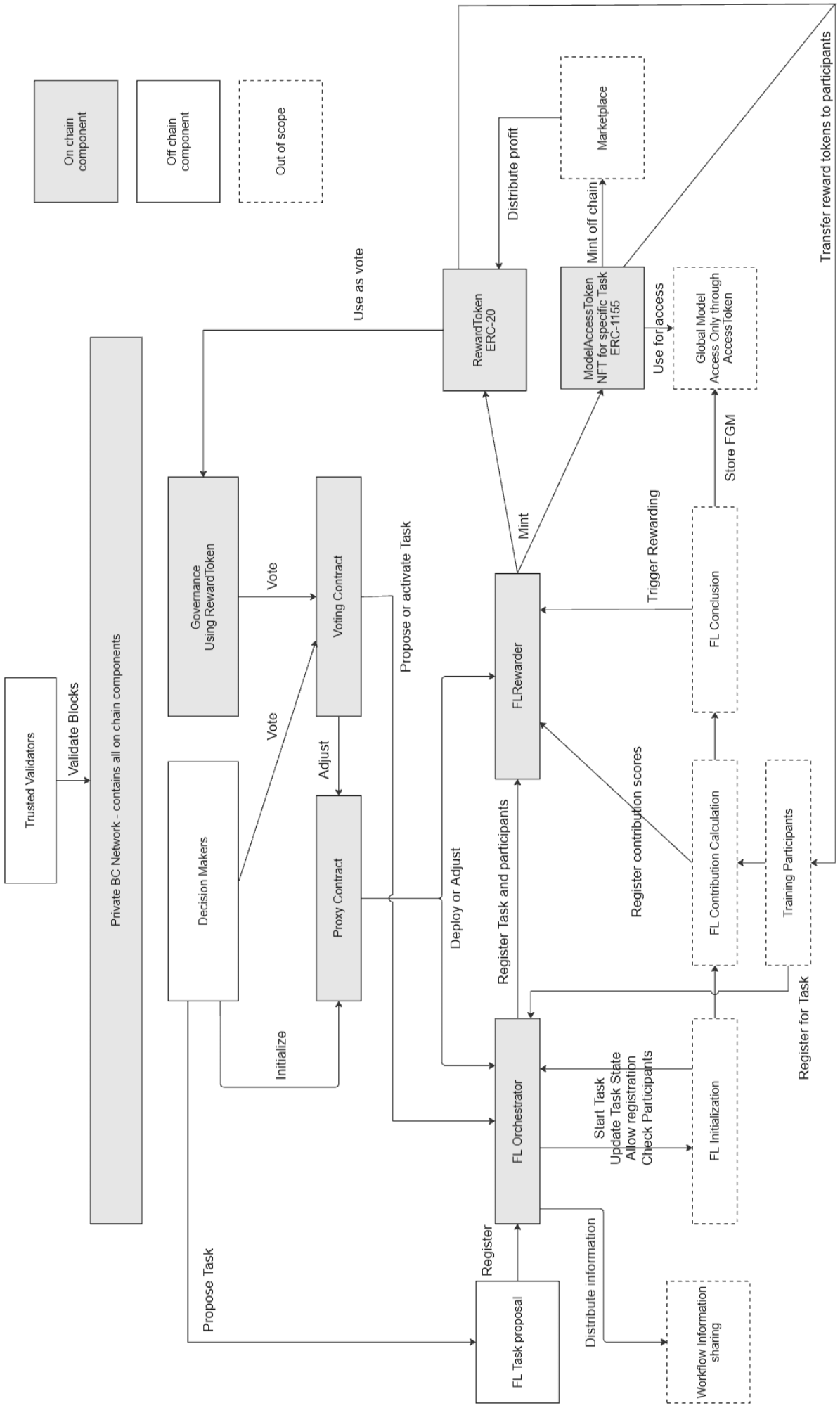

Figure 6.1: Artefact Design

can interact with the Orchestrator to read who is *allowed access to* the current model, workflow information or the parameter server to upload the local model. Therefore, the *status getting* functionality is accessible by out-of-network participants if needed.

The orchestrator works to fulfil the requirement around access control for sensitive data and task contributions (**R7**) by creating role and status-based access to these critical data locations and systems.

Alternative solutions for this part of the design are putting the entire FL process in a smart contract, like Shayan et al., 2020 have done. This method had issues with the upgradability and throughput of the system, making it less appealing also considering the requirement for modularity (**R10**).

FL Rewarder

Built inside the rewarding contract are functionalities to receive the contribution values and distribute rewards to the participants based on those values when the FL task is done. The specific design choice was made for this artifact to not initially do any specific contributions calculations, but rely on other implementations based on the best-fitting incentive mechanism algorithm. This was done to ensure flexibility and modularity to capture and design for the broad spectrum of potential solutions for contribution calculation, such as purely Shapley value or other game theory calculation, captured by only taking an end-score that is calculated in a secure microservice off-chain. Furthermore, the upgradeability of this smart contract allows for adjustments of the methods later in the lifespan of the system. The FL Rewarder *receives contribution scores* for the current round or Task, and, when the task is completed, *rewards participants* based on their score and the total amount of rewards set for that task or in the contract.

Requirements fulfilled through these design aspects are those regarding rewarding and benefit sharing (**R4**) and transparency in the processes (**R5**). The Rewarder is the main method of calculating the total rewards, as calculated by any method, and can automatically mint the rewards to the applicable addresses in a fully transparent way. Both the implementation and results can be viewed by anyone with access to the chain. The choice to make the rewarder algorithm agnostic provides a benefit to the modularity requirement (**R10**), making the system more adaptable to continuous development.

Possibilities for alternative solutions include of course implementing a specific incentive mechanism, such as any from the list of Zhan et al., 2021 or Tariq et al., 2024, some of them even already being implemented as a smart contract. Although many of these implementations are interesting and tested, some issues arise with overhead and performance, and of course, the future-proof modularity expected (**R10**) to keep up with developments in the field.

Reward Tokens

The tokens designed as rewards in accordance with the architecture design and based on existing standards are two types of tokens with their own distinct goals and designs. The standards used are ERC-20 for transferable and own-able tokens and ERC-1155 tokens, a token that can function as a Non-fungible-Tokens (NFT) that are unique and have a specific ID associated with them that can point to metadata. The FL rewarder contract mints these tokens to the accounts that fulfilled the Task.

The RewardToken is a generic ERC-20 token that is minted to the participants based on their contribution and to the original owners of the system, the consortium members in this example. Revenue generated from selling model access can be distributed among the holders of the token.

The ModelAccessToken (MAT) is an ERC-1155 that contains a link to the specific FL task associated with it. Each MAT is linked to that specific finished Task alone and only minted when that task is completed. More information can also be associated with that token, such as a description of the quality and capabilities of the model. The tokens are *burned* (destroyed) to gain access to the FGM. The microservice associated with the FGM can check if the participant has done this action and allow access to run the model. Additional access tokens can be minted when more accesses are needed, and sold by integrating them with public chain contracts. The MAT can be publicized to be sold on for example the public Ethereum chain as wrapped MAT ERC-20 tokens, and sold on public marketplaces to gain access. This process needs further development work Rewards from these sales can be automatically spread over holders of the RewardToken by getting the total amount of tokens owned.

Key roles the reward tokens fulfil are aiding the rewarding scheme requirement (**R4**) and the ability for model access sale (**R8**). The tokens themselves are the reward as both FGM access time, a valuable resource if the model is of high quality, and a governance and kick-back token that can be thought of as a virtual 'share' in the system. Selling the MAT outside of the system if implemented correctly allows for potential higher revenues and rewards, and increases the visibility and relevance of the consortium itself.

Other solutions to the reward tokens were different implementations of the token as well as not using tokens but just storing states in contracts. The ERC-1155 implementation, which is used as a combination of ERC-20 and ERC-721, the default NFT implementation, can be utilized to create distinct tokens in bulk minting commands but has some compatibility issues with current best practices in design, so it is possible to revert to using a simpler ERC-721 as MAT. The option to not use tokens but just register the rewards and right to access the model as a state in a contract is an alternative that fulfils similar roles, but has overhead problems surrounding easy transfer and selling, and also putting a lot of state storage responsibility of the associated contracts, making storage and upgrading slower and more complex.

FL Governance

Providing governing power over parts of the system is the FL governance contract, using votes to steer or control elements of the network such as changing reward values or creating more access tokens. A commonly used method of decentralizing governance, tokens are used similarly to company shares to vote on proposals. The governance contract is supplemental to the power decision-makers and the consortium have, allowing participants to have some power in changing the rules or suggesting FL tasks. The contract allows for *voting proposals*, *voting* and can automatically *trigger proposals* to enact successful proposals. The RewardToken is the token used for voting, although most of the power is still at the original instigators. These minimum balances can be adjusted, with full voting power about starting a task or making big changes never going fully to the token holders, giving them more of an advisory role in the system.

This governance method fulfils requirements mainly as a method of additional governance from participants based on the work they did (**R9**) by giving some power to active participants.

Governance is not an essential aspect of the system and is used to formalize the voting power of participants into code. As seen in DuPont, 2017 theDAO, where an exploit lost millions of dollars, code as the law is not always the best solution, so a powerful contract as a governance method adds some risks. A realistic alternative solution is formal contact and negotiations outside of the BC network, which could also be based on the activity and results of the participant. However, the actual power of the smart contract solution is lower, as the trusted consortium still controls the validators, and both formal and informal talk and negotiation are expected to happen anyway both in and outside the scope of this system.

Proxy Contract

A key method in providing upgradability in secure Smart Contract design is the proxy design pattern. This development method is often used to provide the ability to upgrade contracts easily by having users only interact with the proxy contract, which itself points to the different implementations of other contracts. This means that upgrading or changing the contracts used only needs a change to the contract address in the proxy contract. The proxy contract itself is only the interaction point to other contract functions, and it is able to *store* state and *point* to the other contracts doing the work. This design sets the Orchestrator and Rewarder contracts as proxy-upgradable, given those being the most important and stateful contracts.

Requirement (**R10**) is fulfilled using this proxy upgrading method, allowing the critical contracts to be upgraded to implement different contribution and rewarding functions or changing the methods of controlling the FL tasks. Similarly, adjustments to regulatory changes (**R3**) regarding for example participants or limits around access are done more easily with a proxy contract.

The concept of implementation changes in an immutable ledger like a BC is inherently difficult. Upgrading contracts is always possible, but removing the contracts from the networks is not possible once

deployed, just removing any direct references to it. An alternative method of upgrading is to just upgrade the contract without a proxy pattern: This method works but needs to allow for state transference, as some roles or statuses of Tasks might be stored in a single contract only.

6.2.3. Periphery

FL Process

An FL architecture and system is designed as a black box outside of the system, allowing flexibility in how the FL is performed, with only the interaction functions being part of the system. This is modelled conceptually as a vertical FL setup with a central, trusted, and secured server as a parameter aggregator and model storage location, although another architecture such as hierarchical could also be used. The key steps of FL are modelled as Initialization, Training Rounds, and Conclusion, with an FGM as the result after an arbitrary number of rounds until convergence. The FL process relies on the FL Orchestrators' state of the current task: The Task, with a unique Task ID, is linked to a set of participants and proceeds through training while the FL Orchestrator keeps track of the round status. Access to workflow information, sandbox environments, and the ability to contribute to the model are all controlled by checking the Orchestrator's state. Similarly, contribution scores are sent to the FL Rewarder when the FL process goes through training, sharing the end contribution values of each participant as calculated by an algorithm like Cai et al., 2024 or Z. Wang et al., 2022. The values themselves are publicly stored on the rewarder contract for each participant and after the Task concludes, the rewarding function can be activated.

The external processes do not directly contribute to functional requirements but need to fit within the legal environment to ensure compliance (**R2**). Contributing to the requirements are the interaction points

Alternative solutions include having the FL process inside the system implemented as smart contracts such as Shayan et al., 2020, but keeping it as an external process allows for much more modularity as discussed previously.

External

Other external parts of the designs are aspects of the behaviour of the training participants and the external market. The participants are modeled to interact with the system as individual addresses on the network that are fully controlled by their private keys. This key is the responsibility of that participant and controls both their rewarding address and their identity as a participant in the Task. The participant ID can be transposed to a new address in case the private key access is lost or compromised. The external marketplace can be any form of marketplace where access to the FGM can be sold. This can include public blockchain networks such as Ethereum or Solana, where the MAT can be wrapped as a tradeable token and sold, to be later unwrapped on the private chain to get access to the FGM, or on another B2B trading platform as an agreed upon method. Either way, profits should be returned to the holders of the RewardToken or Consortium owners, depending on the agreements sent at formation.

The ability to work with an external market fulfils requirement **R8** regarding sales, and keeping the private keys as secure methods of interacting with the system for participants helps with ensuring access security (**R1**).

Other design solutions can be different methods of sales, such as selling infinite model access or some kind of subscription model over tokenized access. Additional external interactions can be designed such as collaborative participants sharing access, or controlling external access using whitelisted IP ranges. All such designs have less integration with the rest of the design and fit less in the verified broader architecture designed by van de Walle, 2024.

6.3. Technical Components

To provide a different viewpoint of the technical aspects of the designed artefact, an overview of the technical components as smart contracts and as other IS parts of the artefact as well as the users directly interacting with those parts are presented infigure 6.2.

The key necessary components of the artefact are:

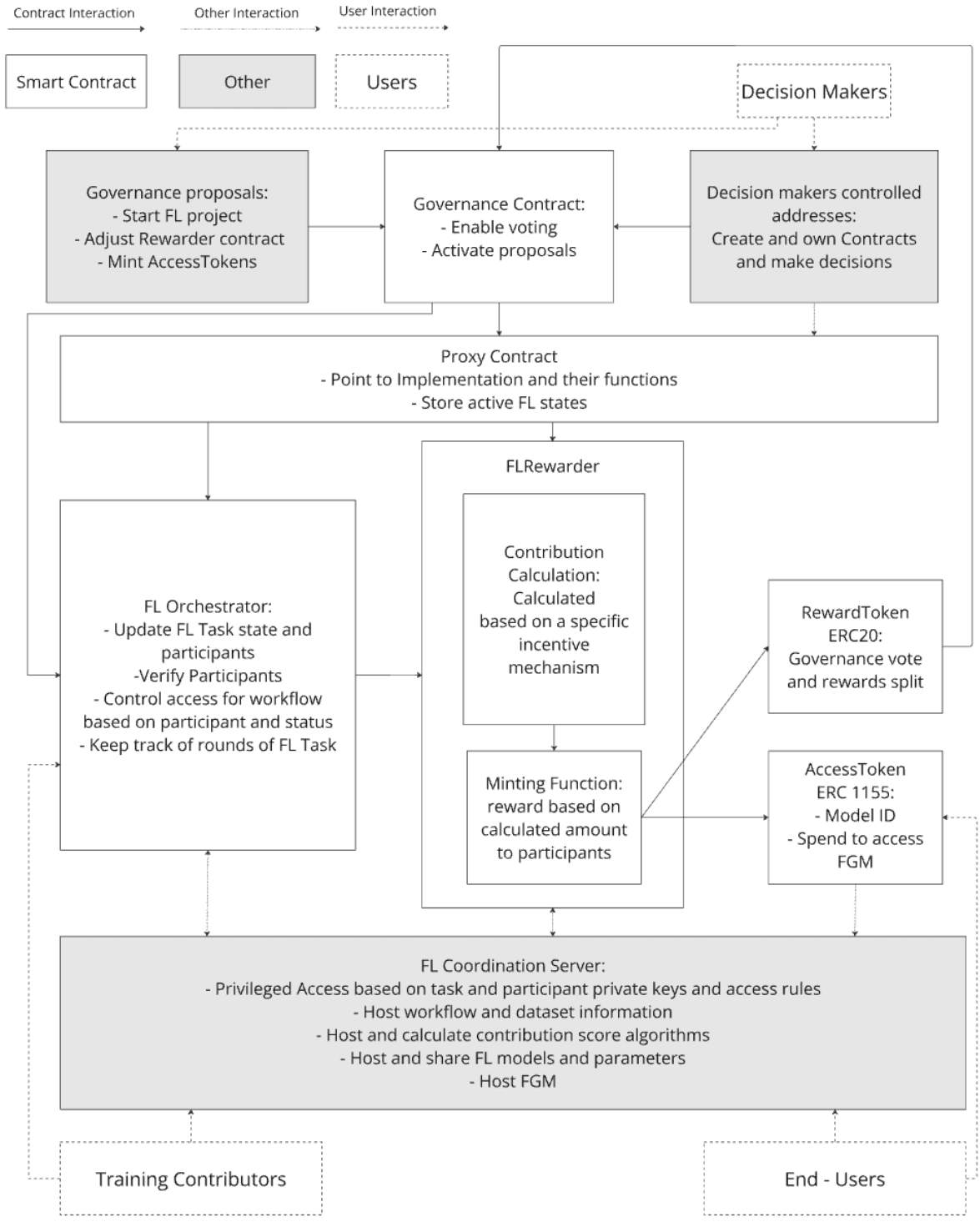

Figure 6.2: Technical Design

Governance Contract allows voting on proposals about FL Tasks and proposals, adjusting rewards or otherwise changing aspects of the system.

Proxy Contract exists to simplify upgrading implementations by only pointing to the implementation contracts and storing active states. Upgrading existing contracts can then be done by deploying a new contract and upgrading the proxy contract to point to the new contract addresses, using the same or similar function callbacks.

FLOrchestrator Contract keeps track of the state of any FL tasks undertaken by the system, including its ID, any metadata, participants and links to workflow. The orchestrator can be interacted with to register as a participant, query to verify if a participant is indeed part of FL tasks, set limits or adjust the allowed participants and start the functions of the FLRewarder if Tasks are completed.

FLRewarder Contract keeps track of the contribution scores that are sent to it about that specific FL task and the contribution the participants made to either the round or the total task, depending on the algorithm used. When a round is completed, the minting function can be used to reward the participants the tokens based on the score in that FL task.

RewardToken an owneable ERC-20 token that allows for governance by its usage as a voting token, and getting a proportional fraction of the kickback from any sales made of models generated by the system.

ModelAccessToken an owneable ERC-1155 NFT token that is linked to a model ID that can be spent, by burning the token, to gain model access from the coordination server. Can be deployed or 'wrapped' in a public chain to allow for its sale on public marketplaces

A private blockchain that hosts all the smart contracts and keeps track of the values, rules, and transactions. Validated using PoA by the consortium owners and any decision-makers also given that right.

Virtual roles on the blockchain that contracts can use to limit actions taken and control participant's role to allow certain parties to update FL tasks in the Orchestrator, trigger the rewarding contracts, or mint new AccessTokens when needed.

The External Coordination Server is the black-box modelled idealized location where the FL process actually happens, where model parameters are stored and where participants interact with during FL itself. The server hosts any models, workflow, dataset information and other related FL Task data, including the servers running the contribution calculation algorithm. To allow access to privileged data or running of the models, microservices should query the chain using the private keys of the requester and access rules to provide access if proven they are allowed to. Additionally, the finalized global model (FGM) is hosted here, needing to have an MAT burned by an interested to allowed access, either for a timed access or a certain amount of work cycle, depending on the rules set for that model.

Interactions with the system are in the form of:

- Participants registering for an FL Task
- Governance proposals being suggested to the FLGovernor
- Model sales generating revenue and being distributed
- Requests for more MAT if necessary
- The partnership deciding to adjust, remove, adjourn or initialize the system
- Decision makers to suggest FL tasks or change rules regarding participation and rewards
- Upgrades and changes being deployed as new or adjusted smart contracts using the Proxy pattern
- Participants use their private keys to register for an FL task, request workflow or model access, use their tokens to vote or check the status of the Task.

SC	Function	Sender	Input	Result
FL Orchestrator	proposeTask	OrchestratorRole	initialParticipants[], rewardPool, initialAccessTokens	Task with new id created with a set amount of access and reward tokens and the status Proposed
	startTask	Orchestrator Role	taskId	Task status is set to Started
	progressTask	Orchestrator Role	taskId	Task round is increased
	finishTask	Orchestrator Role	taskId, finalScores[]	Resolves the training task with optionally final scores, and trigger FL Rewarder calcRewards
	registerForTask	msg.sender	taskId	Participant is added to Task if Task status is Proposed
FL Rewarder	roundScore	Contributor Role	taskId, round, participants[], scores[]	Add the participant's score to a Task round
	calculateRewards	Contributer Role, FL Orchestrator	taskId, participants, finalScores, flrTokenPool, MATAmount	Finalizes the Task by giving each participant's address a final score value
	claimRewards	msg.sender	taskId	Minting of the tokens based on the final score value
FLRewardToken	mint	FL Rewarder	address, tokenAmount	Mints tokens to specified participant address
MAT	mint	FL Rewarder	address, tokenAmount, taskId	Mints tokens linked to that taskId to specified participant address
	burn	msg.sender	tokenAmount	Burns an amount of tokens and signals an Event that the tokens have been burned for that ID

Table 6.1: Smart Contract methods

The system behaviours and interactions are captured in the methods built. The basic functionality of the system is performed with the unique methods explained in table 6.1.

6.3.1. Assumptions

Designing a more specific system in a complex environment needs to be based on specific assumptions regarding behaviour and the implementations of other systems that might interact with the design. These assumptions are partially based on existing implementations and literature. The existence of these assumptions shapes the design and implementation but also provides the necessity for modularity in the design, which itself is also an expected feature.

Technical assumptions about the design were set based on scope and knowledge about potential solutions. Most importantly, the actual FL process would not be done within the designed artefact, contrasting other blockchain-based FL solutions that run part of the calculations on-chain, such as Bao et al., 2019. Data and parameter aggregation and sharing with the global model is done securely through a different system. The artefact controls who is allowed to access these pieces of information and controls the rewarding mechanisms using trusted execution environments. Solutions for this exist, such

as reading the state of a blockchain to enable controlled distributed data processing such as eFLINT (van Binsbergen et al., 2022). Additionally, access to the FGM is similarly locked in a trusted execution environment until a certain state such as an access token is acquired, which the server or container running the model has to check to work. The workflow information needed to perform and contribute to the FL Task can vary and contain IP about the task and data required. Access to the workflow should similarly be privileged, with an access key or flag needed before being able to access this secured information or sandbox for the initial model. Examples include a microservice that starts or stops access based on a certain state on a blockchain. A similar method was applied in Abuzied et al., 2024.

Institutional assumptions include a broad alignment with the rules and design set in chapter 5, although it is expected some of the design may be expandable beyond this specific collaboration. Nevertheless, this design process was performed under the baseline assumptions set in the chapter of 7 participants in a Data Collaborative. These participants might have different goals such as free-riding or attacking behaviour, but are for the designed assumed willing and able to implement any technical speciations and at least 4 actors have enough trust in each other to form the consortium of decision makers. Additionally, it is assumed there is a market for models generated through this FL process, and external parties are interested in joining, using, or proposing FL tasks for this collaboration.

The design of a set of smart contracts as the IT infrastructure was done using analysed and argued for design choices that work together to fulfill the system requirements. The smart contracts fulfill interconnected roles to organise and control an FL process by creating a verifiable and transparent method of registering the calculated contribution and FL information, while also utilizing tokens as both rewards and access control methods. Although much of the FL process itself is set aside as a black box that the smart contract artefact interacts with, this design will be further tested and verified to be able to fulfill its role as incentive control system.

7

Validation and Evaluation

The artefact is validated and evaluated by implementing a basic version to test if it satisfies the requirements set, as well as showing findings from the entire design process. The final design was demonstrated and evaluated artificially in a simulated situation and formally using the system requirements. Additionally the minimum viable product is tested for scalability in terms of performance and complexity.

7.1. Implementation

To show the proof of concept, an MVP of the artefact from figure 6.2 was implemented and tested as a set of smart contracts on a private blockchain. Deployed on Hyperledger Besu, with IBFT2.0 Proof of Authority, the set of five smart contracts was implemented in Solidity using Hardhat and Zeppelin smart contract boilerplates. The off-chain interactions for updating FL tasks, calculating scores and getting models were simulated using Python as interactions with the smart contracts to stay in scope and time frame. Similarly, the implementation of the Governance contracts as MVPs was not implemented using the proxy pattern, but should in any production deployment following this design.

The smart contracts were implemented in Solidity, the standard for smart contract development, using a combination of new contracts and OpenZeppelin boilerplate contracts, the industry standard for secure designs, for the ERC-20, ERC-1155 and the Governor contracts. The source code can be found on https://doi.org/10.5281/zenodo.149596564, a live repository with a functional UI on https://github.com/BrunoLapre/FLDesign and an overview of the most important functions of the contracts are found in appendix A. A basic local Hyperledger Besu environment was spun up using the basic methods for a test network, utilizing Docker Compose containers as validators on a personal desktop Windows 10 Linux Subsystem (WSL) using an Ubuntu distro, which doubled as the virtual host for testing and development. The relevant hardware on the desktop was a Ryzen 2700X CPU, with the VM on a 256GB SSD with 16GB of dedicated memory. The solidity files were compiled and deployed using Hardhat and Hardhat Ignition, utilizing a set of deployment module files, found in appendix B, to specify the deployment. The smart contract addresses were used and interacted with using a Next.Js-based webpage and MetaMask to give commands and update the contract state using the private keys of public testing addresses. Each contract's unique functions were tested to see if the expected input, output and interactions behaved as intended.

In table 7.1 the implemented contracts are shown with their behaviours and affected components as well as which methods and events are used in the contracts for that functionality. The diagram in figure 7.1 shows a visual overview of with which components these interactions move, where the FL servers contain the idealized location with permissioned access and the FL and rewarding algorithms. Interactions with letters are part of the same or similar processes but with different or indirect connections between components.

Interaction	Description	Components	SC Methods
0	Validate the blockchain by running Nodes	Decision maker, BC Validator node	
1	Propose FL Task with participants, total reward and model access tokens	Decision Maker account, FLOrchestrator	proposeTask()
2a,b	Register for FL Task	Training contributor account, FLOrchestrator	registerForTask()
3	Get workflow information	Training Contributor, FL Server	isParticipant()
4	Check if workflow information requester is registered	FL Server, FLOrchestrator	isParticipant()
5	Start Task, register Task as started	Decision Maker account, FLOrchestrator	startTask()
6	Trigger FL Server to accept updates	FLOrchestrator, FLServer	event TaskStatusUpdated
7	Update Rounds and Task status	FLServer, FLOrchestrator	progressTask()
8	Finish Task and Trigger Rewarding	Decision Maker account, FLOrchestrator, FLRewarder	finishTask()
9	Get total and/or round scores, calculate rewards	FLRewarder, FLServer	calculateRewards()
10a,b,c,d	Mint the appropriate reward and model access tokens to participants	FLRewarder, RewardToken, MAT, Training Contributor accounts	claimRewards()
11	Burn token with ModelID to gain access	MAT, FL Server	burnforModelAccess()
12	Allow access	FL Server, End-user Account	event TaskStatusUpdated
13	Purchase access tokens	End-user Account, MAT	
14a,b	Vote using the reward tokens	RewardToken owners, FLGovernor	
15	Update or use contracts based on voting proposals	FLGovernor, FLOrchestrator or FLRewarder	

Table 7.1: Interactions in Implementation

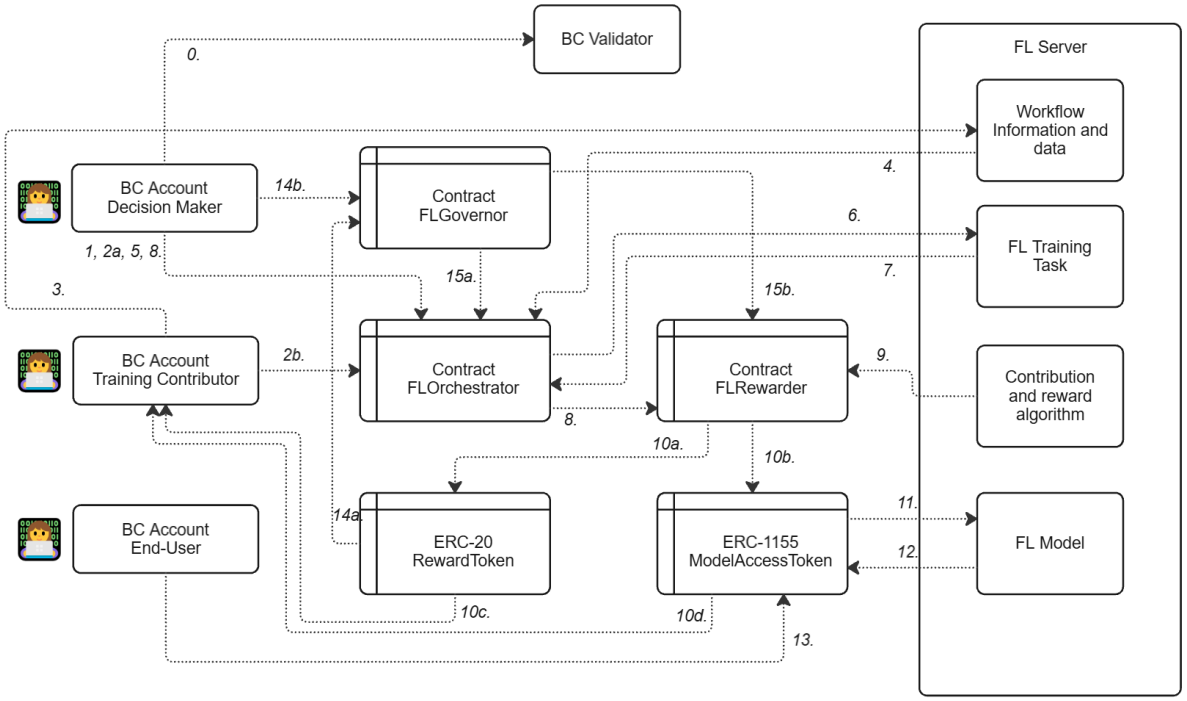

Figure 7.1: Implementation interactions

7.2. Validation

To validate the system, expected behaviours should be by the problem situation and the institutional design to show the problem is being solved with this artefact: *Does this collection of smart contracts organize FL and incentivization for aviation participants in a trustworthy manner?*

7.2.1. Basic functionality

To demonstrate the usefulness of the artefact, the designed smart contracts need to be successful in controlling FL processes in at least one situation, providing the example institutional setup defined in chapter 5. The situation simulated was a collaborative of 7 participants *p*, one The experimental setup for the validation was deploying the smart contracts on a private blockchain on a local machine and interacting with the contracts for a basic functionality needed for a successful FL process orchestration and rewarding. The process steps needed to provide basic functionality were the interactions in table 7.1. Each interaction was performed between the associated components using a web application and results were observed. With every possible interaction, all components performed as expected, verifying basic functionality: the expected results and if the tested implementations fulfilled these expectations are presented in table 7.2. The not implemented functions and interactions were still possible with the design but were, again, left out of scope.

The example scenario provided in chapter 5 is used to explain how the implemented contracts can be used in such a process. The original participating parties decide to utilize this design and start by running blockchain validators themselves, for a total of 5 validators for the chain. They and other eventual parties each notify each other which address is associated with them to allow them to validate their transactions on the chain are valid. The initialising manufacturer proposes a Task with ID 0, as it's the first one on the system. On a server they upload all the relevant information linked to Task ID 0. The interested participants register on the orchestrator contract and can access the workflow information. When it's confirmed all participants are registered, the consortium can vote or more simply the initializer can start the task. This sends an event out to the FL server, where out of context of the chain FL can occur. Every round can be registered to be completed with optional contribution scores. For this scenario the contribution score is a Shapley Value, which is calculated when the FL process finishes. When the manufacturer or enough decision-makers decide the FL model is no longer meaningfully improving, the finishTask method can be used to register the FL task as completed on-chain, starting

Inter.	Expected behaviour	Results
0	Any number of validators controlled by specific servers can be set.	Success, nodes can be set, needs at least 4
1	Propose a Task with a set of participants and total tokens as a reward.	Success, task registered in FLOrchestrator contract
2	Account can register for a Task if allowed	Success, the account can be registered if not already in, if the task has not yet started
3	Only access workflow and other data when part of the associated Task	FLServer not implemented in the MVP, possible with the design
4	FLOrchestrator needs to be able to prove requester is registered in active and associated Task	FLOrchestrator can be queried to show the account is registered
5	Allowed account can set Task to started	Success, task status set to started in FLOrchestrator contract
6	FLOrchestrator needs to trigger the FL process itself on the server	Possible using the task started Event emitted by FLOrchestrator
7	FL Server can update the rounds	Possible through the FLOrchestrator, FLServer not implemented
8	Able to finish task, which then triggers the rewarding function of the Rewarder	Success, the decision maker can force a task to finish, which triggers the Reward calculations
9	Contribution and associated reward calculated for that Task per Participant address	Success, score calculated using arbitrary values in the MVP: When every participant contributes equally, scores are equal.
10	Reward tokens minted and distributed	Success, and minted tokens directly to the participant accounts.
11 & 12	MAT burned for access to the Model	Partial success, burning of token is possible but no associated trigger on FLServer
13	Other accounts outside the system can purchase the MAT	Not directly implemented in MVP, possible to transfer or wrap an ERC-1155 token for later selling.
14	Voting functionality works	Success, boilerplate implementation functions as expected with the RewardToken
15	Proposals can directly influence the Orchestrator and Rewarder	Partial success, the proxy pattern was not applied so upgrading was not tested, however, the Orchestrator was triggered using a vote on the proposal.

Table 7.2: Functionality Results

ID	Requirement	Fulfilled by	Interaction
R1	The system must have strict access security measures	FLOrchestrator, Private BC	*i4, i12,*
R2	The system must ensure compliance with existing legal standards	*Non-functional requirement*	
R3	The system must be adaptable to evolving regulations	Proxy design, Consortium control,	*i15*
R4	The system must establish mechanisms for equitable rewards and benefit sharing	FLRewarder, Off-chain mechanism, Marketable MAT	*i9, i10*
R5	The system must implement transparent processes for value attribution	Private BC, Smart contracts	
R6	The system must appoint a trusted, neutral controlling body	Consortium validators, FLGovernance, Voting with RewardToken	*i0*
R7	The system must implement access control methods to FL tasks, information, and models	FLORchestrator, FL Coordination Server	*i3, i4, i11*
R8	The system must enable the sale of model access	Marketable MAT, wrapping function	*i13*
R9	The system must establish methods of additional contribution-based governance	Voting with RewardToken, FLGovernance	*i14*
R10	The system must provide modularity regarding contribution and rewarding methods	Proxy Design, FLRewarder FL Coordination server	*i15*

Table 7.3: System Requirements Fulfilment

the rewarding process. Based on the scores sent, MAT and FLR tokens are sent out to the participants. Usage of the model can be done by simple showing an address has an MAT, or by burning the tokens, depending on the FL model settings defined off-chain. The FLR token amounts are used to provide voting power through the governance contract, and the amount of FLR tokens are used to calculate any reward distribution from on or off chain model sales.

7.2.2. Fulfilment of requirements

The requirements for the design in table 7.3 are tested against this smart contract design. Although the design choices were informed by the requirements and built to fulfil the requirements, a short critical reflection is done to argue how well the requirement is fulfilled by the respective components and the interactions of the design. Interactions that were not implemented in the MPV were still part of the design and were deemed possible to be implemented, so were considered to help fulfil the requirement; the interactions are noted as *i0...i15*. R2 does not have an associated component or interaction as it is a non-functional requirement, and R5 does not have associated interactions as the capability for transparency is fulfilled not through an internal interaction but by the ability to look at the transaction history and deployed smart contracts.

Security Validation

Security (**R1**) and access control (**R7**) are both captured in the inherent blockchain and smart contract design, with the private chain controlled by consortium validators providing both a secure and transparent ledger only editable by trusted sources, and the MAT being useable as an access method for the finalized model. The access is governed by these immutable states of the smart contracts on-chain,

only changing at appropriate moments, if everything is running as expected.

The blockchain as the basis for the design was also a design choice made in the validated design of van de Walle, 2024, and the literature agrees with BC as secure as long as the method of chain integrity, in this design PoA, remains secure. Attacks on the security of the BC mean attacking the integrity, which can only be done through attacks on Validators so that a controlling majority has been compromised. The potentially sensitive data surrounding the FL task such as workflow and initial parameters are still stored on a system outside the direct design, so rigid security is needed for security to be as tight as possible. Additionally, as access is linked to the private keys of the participant trying to access a certain secure environment, any leakage of the private keys poses high risks of potentially unidentifiable security breaches. IP detection and two-factor authentication beyond the private key access can be implemented.

Regulatory Validation

For the requirements in a regulatory field (**R2** & **R3**) the design stays within the set regulations as it does not currently generate any models itself and many aspects of the artefact can, even when deployed, be modified to adapt to the regulations using the proxy design. The non-functional requirement **R2** is fulfilled as long as none of the external systems in the periphery move outside of the compliance. Additionally, any sweeping or important changes to the regulatory environment or the implementation of this design in a different regulatory sector or geographical location with different rules can be adapted to by the consortium changing aspects of the design. However, if any aspect outside of the system does fail to meet compliance, this specific system has no inherent checks or limits to stop that, besides the formal rules the consortium has to follow in that case, which are not governed by automated systems. Solutions such as those proposed by van de Walle, 2024 as a Compliance database with a regulatory update monitor might be adapted to work with this design.

Incentives Validation

Equitable benefit sharing and rewarding (**R4**) is the key design aspect of the entire system, providing the capability to mint both rewarding and access tokens, and slashing the rewards of potential free riders, using the FLRewarder contract in combination with the Off-chain systems. Transparency of the valuation process **R5** is fulfilled through the readable BC as a ledger for the distribution transactions as well as partial calculations in the contracts. The access token is a solution for providing access to the model in a transferable way, providing the possibility of the sale of model access (**R8**).

Although the token rewards themselves are implemented as unique design aspects, the actual calculation for contribution and resulting rewards are not fully implemented in this design. The full functional requirement therefore is not fully fulfilled, as other design and engineering choices might make the rewards less equitable, and are not a direct part of this design. However, this is an intended design choice, creating a more rigid design that can implement a wide range of algorithms for the incentivization of FL.

Similarly, transparency is limited to the on-chain environment: added value attribution is mostly performed off-chain using a separately designed algorithm out of the scope of this design. Although the results of these calculations are shown on the chain, the full transparency needed for this requirement is not guaranteed with just this designed set of smart contracts. Additionally, the pseudonymous nature of BC addresses (formatted as 0x....abc123) means the identity of participants can be known and shared, resulting in situations where, depending if participants have not publicly shared or were not forced to share their on-chain identity, transparency about contributions and rewarding is lessened.

The marketing of model access could also behave in a different way than expected, being submitted to market mechanics and following in the footsteps of many other failed utility tokens in the cryptocurrency market. Care must be taken to adjust expectations and information dissemination about what the token gives access to, and precautions must be taken to adjust the expected rewards from the sale of models to other parties in or outside of the sector.

Governance Validation

The governance and management aspect of the system (**R6**) is appointed as a collection of participants in the decision-maker role with the power of both changing aspects of the contracts, helping (**R10**)

as well as organizing FL tasks, appended with marginal voting power from RewardToken holders for contribution-based governance (**R9**)

The neutrality and trustworthiness of the decision makers is an important aspect and can be trusted as long as the most powerful entity, the consortium itself, can be trusted. Although the neutrality of the Reward token holders turned voters might be lessened, as they are not chosen but rewarded with governance power, they are expected to behave in the interest of the system that gives them such rights and rewards. However, behaviours can change, turning the actions of active participants with a lot of voting power or even powerful actors controlling the consortium against the interest of the system or becoming biased. If the game theory strategy goes towards non-cooperation, the entire system can still fail to perform productive collaborations.

Modularity Validation

By utilizing the proxy designs and allowing upgrades across all smart contract implementations, as well as allowing the actual incentive calculations to be done using various differing designs, the modularity requirement (**R10**) is fulfilled.

Although modularity is high in some aspects, contribution and rewarding methods are not updateable during tasks and rely on the possibility of updating and implementing different algorithms on the organization server that also do not interfere with the processes of the on-chain system. Possible problems can occur when an algorithm is changed in the middle of a long FL process or the values shared to the FLRewarder are wrong due to some mistake, which will be permanently added to the chain unless also smart contracts are upgraded in response.

7.2.3. Context Validation

The last validation step is fitting the system in the FL architecture proposed by van de Walle, 2024. For reference, the latest validated model from that thesis is included in appendix D. This design implements part of the FL architecture, Distributing tokens, Contribution evaluation, and the Blockchain and associated Contracts. The institutional design does not fully overlap with the surrounding context of van de Walle's design but has much of the same properties as the consortium model influenced by regulators. The key properties, a controlling organization, blockchain-based tokenization, and specific incentive mechanisms are aligned enough to validate the design in that context. Additionally, the requirements most of this design was based on are themselves based on that work, increasing validity.

7.3. Testing

The MVP was tested to show its effectiveness and usability by measuring the speed of key methods, the complexity of the contracts and the size impact on the nodes. These tests were performed to find how a similar system would be able to perform and fulfil its role: A system that is too slow or unusable would not be able to scale well, so acceptable performance indicators are important. The indicators tested here consist of the speed, to make sure many access calls to the contracts do not take so long to make interactions take too long at scale, the complexity of the contracts, which impacts the on-chain size, and the file-sizes of the contracts and blockchain environment on the validators, relevant to make sure scalability is possible. A python script found in appendix appendix C, including instructions for replication, was used to run the contracts running on the local Hyperledger blockchain network as indicated for speed and complexity, and the on-disk size of the validators was also found, with the results aggregated in tables and graphs presented in this section.

The Python script ran 500 iterations of all the relevant and unique methods of the Orchestrator and Rewarder contracts to test speed and complexity performance. The methods chosen to be tested were non-exhaustive, as many of the contracts were extensions of existing OpenZeppelin boilerplate contracts - already known and tested - which were not themselves evaluated here, and selected only if they were actual transformative function calls. Testing the contracts meant going through every step of the FL Task process once: Propose a task (proposeTask), initiate the task (startTask), go to the next round (progressTask), add some arbitrary score a participant added to that round (roundScore), finish up the Task (progressTask), calculate the rewards (calcRewards), and the claiming of the rewards by a participant (claimRewards). Additionally, the basic get methods were tested to make sure all the data was as expected and correct. The script called the functions 100 times with the same data and FL

Method	TPS
proposeTask	23.57
startTask	138.2
progressTask	200.43
finishTask	208.75
roundScore	136.67
calcRewards	31.84
claimRewards	16.05

Table 7.4: Method TPS

Method	Gas
proposeTask	173,041
startTask	52,287
progressTask	52,237
finishTask	35,860
roundScore	85,049
calcRewards	149,149
claimRewards	274,666

Table 7.5: Method Gas costs

task information and recorded the duration of the code block performing the function call execution for speed and gas costs. Scalability was tested by checking the impact of running the contracts - therefore storing data on-chain - on the file sizes of the 4 validator nodes.

7.3.1. Speed

Measuring the performance of smart contracts is a complex procedure that is based on blockchain processing speeds and contract complexities, as well as the specifications of the chain itself. For this MVP, the speed was measured similarly to public blockchains in Transactions per Second (TPS), to see the maximum amount of transactions that can be included in the validation processes.

The TPS is based on multiple factors: the chain design itself and contract complexity, and is not necessarily a measure of the actual performance, which can be dependent on other factors such as how busy the network is but can be used to understand the upper limits of the number of times the smart contracts can be used. A high TPS is an indication that many transactions can simultaneously be performed per block validated. The so-called block speed of the Hyperledger Besu under the IBFT 2.0 validation method is about 5 seconds, meaning a new block with the transactions is validated and added to the network every 5 seconds. The test script measured how many transactions with the respective method call would on average be executed per second. For the simpler calling functions, which rely on reading the chain and not the actual transaction speed, the time to get a response from the chain was measured.

The average TPS per method is presented in table 7.4. The graphs of both the simple get methods to get information from the chain, as well as the average TPS for Orchestrator and Rewarder functions, are presented in figures figure 7.3 and figure 7.2.

7.3.2. Complexity

The so-called Gas costs were calculated at the moment of interaction with the methods and are indicative of the complexity and efficiency of the implementation on the chain. From every interaction with the methods, the used gas costs in wei, the base units for costs in EVM ecosystems, were recorded. The gas costs were consistent among every contract call and recorded in table 7.5 and as a graph in figure 7.4

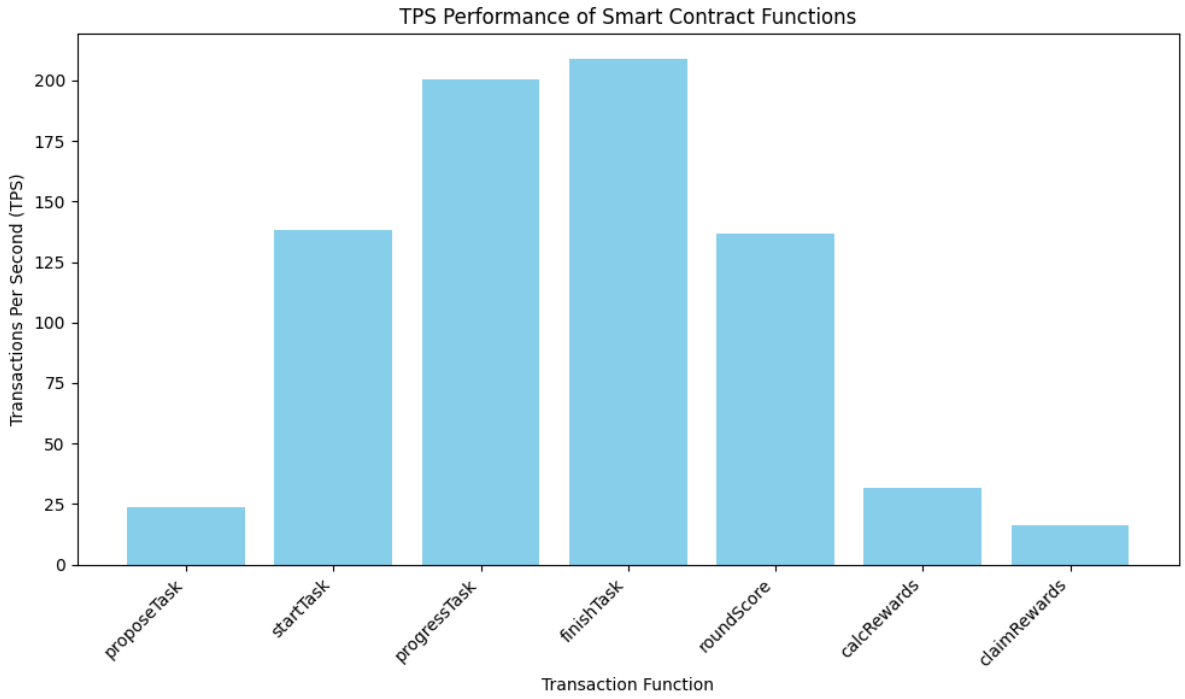

Figure 7.2: Methods TPS bar graphs

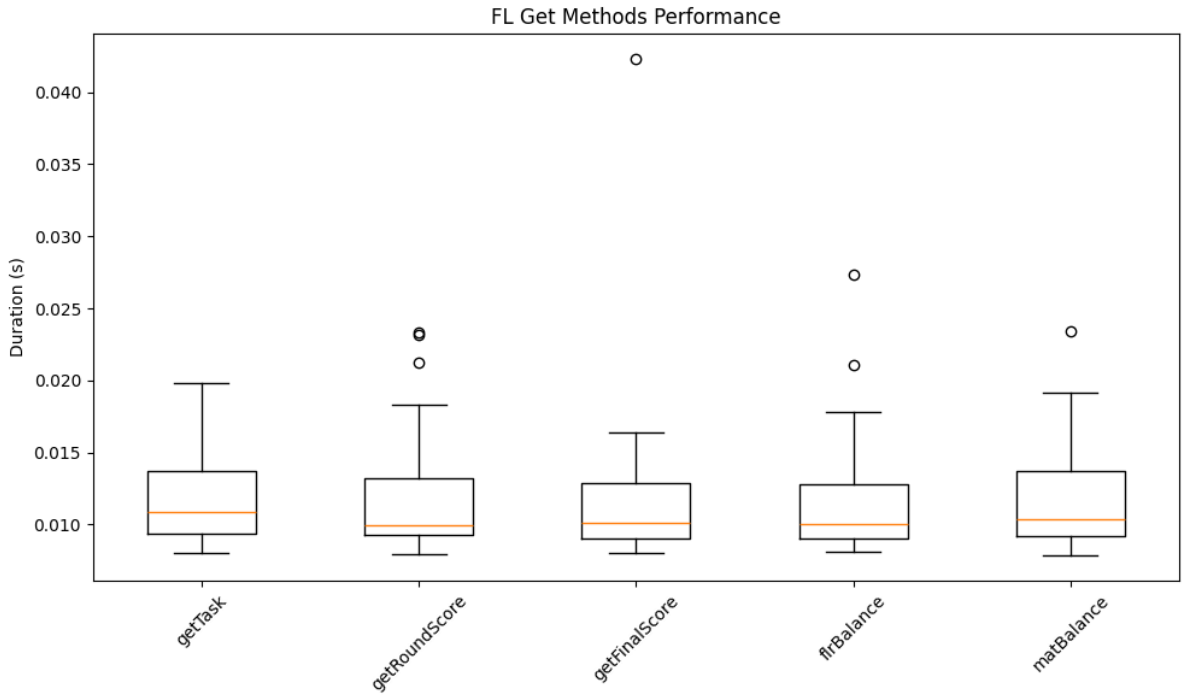

Figure 7.3: Get methods performance in seconds

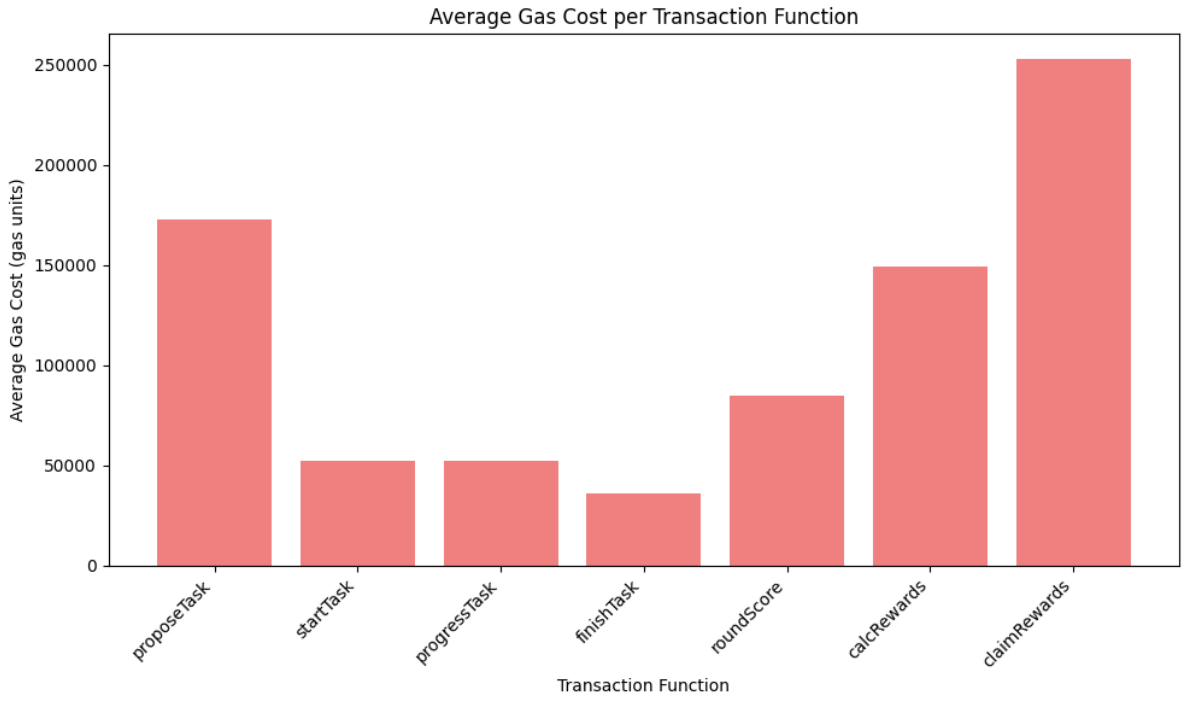

Figure 7.4: Average Gas Cost Per Transaction

7.3.3. Size

The on-disk size of the validators and data pertaining to the orchestration tasks was about 23 GB with all the relevant docker data and images. Scaled up, compared to the giant sizes of the datasets being trained on running into hundreds of terabytes, these image sizes are not impactful.

On-chain, the size of the contracts is measured in bytecode, with the state-changing interactions having a small impact on the size. The sizes of the contracts are in the size range of 30 kilobytes, which are the optimized post-deployment values, which are also around the average size of smart contract bytecode. The impact of having larger on-chain contracts is mostly related to initial deployment costs and time and size increases of on-disk files for the validators. As a private chain, control can be kept over the contents of the chain if any problems arise, so no big issues is expected.

7.3.4. Testing conclusions

Testing the MVP revealed the basic functionality of the implemented contracts behaving as expected and within performance expectations and should be able to scale towards industrial applications. As a small-scale test, it revealed insights into the scalability by giving an indication of the range of the speed of the smart contract transactions, gas fees and file sizes.

Based on a block transaction speed of 5 seconds per block validated and added to the chain, the smart contracts TPS were in line with the gas costs and had useable speeds for scaling up. As the transaction speeds are not expected to be needed to perform instantly, the speeds are within limits and are expected to scale well: In the case where transactions take a very long time or the network is busy, some events can take a while, but the ordering of the memory pool of the EVM is built to handle such issues.

Gas costs indicate that variations levels of complexity exist in the various contracts, which is expected as some expand OpenZeppelin contracts with some complexity built in, which are however optimised. The implications for these gas costs from this implementation are that the complexity of the contract calls varies but is never on a very high cost. On the public Ethereum chain, for example, contract interaction gas prices for things such as sending Ether or minting tokens are about 1000 times more, depending on the activity on the network. Although the gas costs themselves do not cost anything as the transaction fees are de facto waived on a private chain, the complexity does show the contracts as

implemented are not overly complex and should be able to be expanded upon.

The size of the contracts on the chain and the impact of running the chain on a local validator on storage space is nearly negligible in the current MVP implementation.

The technical tests of the MVP were successful in showing the potential for scalability by showing how neither speed, complexity nor file sizes were problematically high.

7.4. Further Evaluation

Besides the requirement-based evaluation performed in the previous section, key criteria of trustworthy FL. Performance such as speed and gas costs of the contracts and the impact on the accuracy of the model are often part of FL-related designs and implementations but were left out of scope because the private PoA chain cares little about gas fees and costs, and no fast-access is required from participants so speed is not a relevant factor unless it is extremely slow and unusable. Impact on accuracy is also not expected of the smart contracts, giving no on-chain calculations for the tasks performed.

7.4.1. Trustworthiness

The trust evaluation criteria collected from literature by Tariq et al., 2024 is used to find if the implemented design fulfils the trust aspect of the expected system. The broad collection of criteria includes the relevant to this design criteria of Transparency, Robustness, Incentive Mechanisms, Contribution Evaluation, and Client Selection. Each criterion is briefly touched on to evaluate the total system's qualities regarding these criteria.

Transparency for model behaviour and decisions made is captured adequately through the transparent decisions on-chain and insight into proposals and changes, but the off-chain aspects of the design do not guarantee full transparency. Additionally, this design does little to clarify model behaviour besides giving broad insights into the balancing of the contributions.

Robustness in the face of adversarial behaviours or attacks is ensured through the BC design, even though validators can still be compromised, the PoA setup is robust against attacks. Other attack angles such as malicious contributors can be caught during tasks if their contribution remains low, while lazy participants are punished by slashing rewards. However, individual participants are as weak as the weakest security feature in the systems and can risk wrong results or changing behaviour.

Incentive Mechanisms for improved participation and attracting trustworthy participants were a key concept and design goal, which the dual-token design captures.

Contribution evaluation done off-chain but controlled through reliable smart contracts helps with increasing trust in the system if they consider multiple factors. Depending on the implemented off-chain methods for calculating contribution, the contribution calculation mechanisms generally help with the trustworthiness.

Client Selection to filter for trustworthy clients is possible in the system, but no direct rules or directions are set to pre-select based on trustworthiness. Only after at least one round can insight be gained if certain clients behave in a trustworthy manner if they score high enough on contributions.

8

Discussion

This thesis conducted design science research on building systems around the incentivization of Federated Learning in the aviation sector with the help of an institutional analysis and work done by a previous thesis. The research provided several insights in designing around institutions for FL, technical implementations and the potential for scaling such a solution. Key contributions to the field are institutional insight into FL for regulated sectors and a unique variation of FL rewarding and orchestration on a private blockchain.

8.1. Findings

The research questions were posed as a design goal: *How can a blockchain-based incentive organizing system be designed and implemented to increase participation and trust in federated learning systems within an aviation partnership?* The design science research process was followed based on sub-questions about the requirements (RQ1), institutional setup fit (RQ2), technical solutions (RQ3), implementation and performance (RQ4), and evaluation (RQ5).

For RQ1 existing requirements from previous work were analysed in chapter 4 and evaluated for this context, keeping some aspects of the design and guiding the design towards a fit within an FL for aviation architectural model. Adapting these requirements required some re-scoping from the broader FL system this research utilizes by van de Walle, 2024. The requirements were adapted into those that fit better for a system that aims to solve the problem of incentivization participation instead.

RQ2 was answered using an institutional analysis in chapter 5, finding a data collaborative based on the consortium controlling the FL system to be a good fit, with applicable rules and the need for proper rewarding. An adapted IAD framework was used to find contextual features and defined roles that are expected in an FL process. The defined playing field consists of data owners turning into training contributors that perform round based FL tasks, controlled by decision makers influenced by regulatory bodies. Economic incentives are added using a market mechanism of selling access tokens to give the system value.

The inventory of solutions needed for RQ3 was part of the design process in chapter 6, ending up with blockchain-based reward tokens using flexible and modular smart contracts as fitting solutions. A dual token design that rewards participants based on their calculated contribution with both access tokens as well as reward tokens that also signify some kind of stake in the system was adapted in a set of smart contracts. This design puts the FL process itself aside in a black-box and focuses on the smart contracts that organise FL tasks and rewards.

RQ4 needed an implementation of the design, showing the proof of concept by writing and testing the basic functionality at the start of chapter 7. The contracts were developed and deployed to be tested and validated to fulfil their expected roles within the designed system. Within an example scenario and with regard to the requirements, the contracts worked as a minimum viable product within an actual deployable blockchain environment.

Finally, RQ5 results in chapter 7 that the basic design functions well and performs its role as an orchestration and incentive system fit for FL within all the requirements set and can be scaled up to real-world applications. The requirements are fulfilled at least basically, but still need further development as the FL process itself is not fully adapted for the contracts.

An important part of the design science process is the aspect that a perfect solution is not being found, especially not in a short research time by a single person. This design was to find a solution that satisfied the requirements and could be provably useable and expanded upon by further researchers in this direction. The findings, in this case, were fitting within this goal, having designed and tested a set of smart contracts that fulfil requirements adapted from previous work within the context of a found institutional design. The new requirements or the design were however not validated through feedback such as interviews. Therefore this designed artefact is far from a perfect solution, as will be discussed, but does perform its role in progressing insights for design in FL and incentivization thereof.

The chosen design focuses on satisfying the requirements found fitting within previous work done, and performs well in that aspect, being able to be fair and reliable through its BC design providing immutability and transparency. The choice for modularity and adaptability with voting input from high-performing participants works to strengthen the trust of already participating actors in the system. The design is strong in its ability to adapt to better or more fitting reward mechanisms and regulatory concerns, but the adaptability choice does sacrifice some reliability, as no guarantees can be made the system remains as it is.

The implemented contracts, which were not a comprehensive implementation of the system, performed as expected but were also tested for performance. The tests ran found little troubles with the systems' ability to scale up, with the speed and complexity within expected limits. Therefore the contracts are expected to be able to scale and perform a functioning role in further development and research.

8.2. Interpretations

The results from this thesis were centred around the design process for both an institutional field and a set of smart contracts for an FL incentive system. In this process, sub-results were found that have an impact on future research in this direction as well as the application of such designs in large-scale operations. The findings are interpreted in this section.

8.2.1. Institutional

From an institutional perspective, FL training systems are not very common or researched much, with much institutional research done more within ML in general and data sharing. The current research understanding is that FL is a promising direction for collaborations, but more real-life applications need to be studied to advance understanding of the institutional and collaborative impact of FL. This work furthers the research by setting up a market-based approach that informs a design process.

Utilizing an institutional analysis to find the answer to the relevant sub-question was performed as part of the literature analysis and the application of a modified analysis framework to understand how actors can behave within certain contexts and action situations. This resulted in an institutional setup where actors have different roles within FL training, where FL tasks are performed by contributors and governed by decision-makers, with regulators having the ability to guide the rules within. The design of the playing field sheds light on the need for adaptability to regulatory and technological changes, as new developments can change how the system performs. The highly regulated nature of aviation makes ML-based development a target for scrutiny from regulators, something the current literature does agree with as shown in the active research regarding the AI act. Therefore, based on this design and the current state of research, being able to adapt aspects of the system is important to be able to in future applications.

The collaboration type drafted for this system was one where governing power was enshrined in a set of stakeholders as a consortium that had a vested interest in the results. This design also added the need for some contribution-based governing capabilities or a sense of ownership to further incentivize participants to behave in a productive way, as they could then benefit more. The data collaborative as partnership form was based on current research supporting such designs for FL, and is seen in this research as a fitting model of collaboration. The addition of contribution-based governing in the system

was modelled after existing DAO models, which do not always perform as well, however, as seen in public examples. Therefore, attention needs to be directed towards the institutional functionality and behaviour of giving some influence to participants through contribution and partial ownership aside from consortiums.

Model training contributing actors were modelled as performing well, working poorly or working lazily towards the goal. This fits within the system to underline the need for proportional reward and an understanding of how the actors behave. Active research in the field of FL actor behaviour also models actors this way and requires incentivization methods to steer them towards active participation to diminish the free-riding risks and poor model performance. A trustworthy incentive layer that is shown to be functional and scalable in this research therefore shows the need for further work in technical applications and actor modelling. The market-based approach built for this expects to give increasing value to well-performing actors through models being worth more and sold and therefore can increase the use of such incentivization methods.

Because of its usefulness in helping guide the design and understanding of actor behaviour, institutional research for FL in regulated sectors such as aviation seems like a useful activity to help guide compliant and adaptable designs for FL in the face of a multitude of actor behaviours and evolving regulations. The results found here show that a market-based approach to incentivization of FL in aviation can be a useful research direction.

8.2.2. Design

The completed design shows possibilities for utilizing and implementing smart contracts on a private blockchain for FL organizing and rewarding. Based on existing literature, this is not a completely novel concept, but the design for this research provides new insights with some different design choices and adapted within a specific sector.

Utilizing blockchain for the highly desired properties of transparency and immutability for the reward process was a design choice both initially suggested by the thesis van de Walle, 2024 this research was based on and a commonly used solution in other FL research. Using smart contracts helps with automated and transparent FL organisation and can be both reliable through BC's immutability, as well as flexible with the ability to upgrade the contracts in the face of new developments. The broader FL literature suggests many different kinds of rewarding and organisation methods, so this design is one of many, but does have a wider design space, capturing both FL organisation and rewarding through the Orchestrator and Rewarder contracts, than most others. The validation of this implementation and development of the original idea aligns with the literature and can therefore be seen as a successful step in furthering design insights.

Rewarding participants based on contributions calculated from different sources and rewarding based on a dual-token system is an important design decision based on other similar implementations but is uniquely organised in a private chain that provides voting and revenue kick-backs to the holders of the reward token and a flexible model access token. Through the arguments made regarding the behaviour and interaction within the designed contracts, the effectiveness of this method seems promising in fulfilling the set requirements and fitting within the context of similar designs. In the current state of research for FL rewarding this is yet another option, but one that can be tested further through the smart contracts built and is based on a unique design.

The implementation of the design as proof of concept demonstrated the basic functionality of the smart contracts within a private blockchain, showing with tests around speed and scalability such a solution can scale up towards production use cases in the sector. The results of the tests and general evaluation show a promising design that can guide further design steps and testing in the field.

8.3. Implications

This research and its results advance the broader field of incentivization and organisation in FL in both practical and theoretical ways by providing a full design cycle including MVP implementation, showing the potential for real-world use, and advancing the understanding of designing around FL through the insights from designing both a technical artefact and an institutional field.

8.3.1. Practical

For implementing FL in aviation, key implications are around the usage of blockchain and tokens for rewarding participants, flexibility in design, and building around certain behaviours. The results of the research point to potential practical steps to be taken for further development based on behaviour and technical implementations.

Blockchain as a method for implementing around FL is reaffirmed as a valid and useful design paradigm, with a private blockchain allowing both internal transparency as well as security. This research finds that utilizing private chains for immutable transparency is the key application of such technology, and the potential to scale contracts within a chain like Hyperledger Besu does show its real-life usefulness as a functional method of incentive organisations. As an oft-maligned technical solution with few use cases besides cryptocurrencies of dubious value, the found implication that BC can be useful for such an FL process therefore is promising for both FL design and BC usability in general.

Using contribution-based reward tokens as an incentive that provides a revenue share, some voting power and model access is a promising direction of incentivization that increases economic benefits and adds a level of self-control in the system by giving active participants some responsibility. Such economic models and semi-ownership can be used to develop robust FL partnerships and provide a new dimension of potential incentivization designs. The results show that such a design basically functions by fulfilling set requirements and scenarios, but needs more exploring in practical applications as that was not done in this research.

Design flexibility and modularity are important in an active research field where new implementations and methods can improve performance over the previous iterations. In a field like ML, having the best model by using the most accurate or most efficient learning methods is a key concern, and in the aviation sector, being adaptable to regulations for sharing or using models is needed for continuous compliance and safety. Being able to adapt implementations as they are validated can help keep the system competitive without starting anew, improving the longevity of a successful partnership.

8.3.2. Theoretical

In the research field surrounding FL, developments were made through the research in this thesis on institutional setups, trustworthiness and design methods.

Utilizing institutional analysis and design for FL is a promising direction that can be applied more in the research areas of data collaboratives and broader ML organisations. This research found a use for applying an adapted IAD to help guide design and understand actor behaviours. Having insight into who should be responsible and which rules should be enforced can inform design choices when implementing FL. Furthermore, the usage of such insights can guide the technical designs and methods for dealing with actor behaviours and dealing with the regulatory environment. Therefore, Further research into FL for aviation and other highly regulated environments could benefit from institutional analyses used for designing partnerships and responsibilities within such systems.

The institutional field suggested fits within the context of a broader and verified FL architecture for aviation and can help progress the understanding of stakeholder behaviour in more research. Data collaboratives or similar partnerships with shared governance through a consortium are promising and previously found directions for FL governance research directions, a research field that can also gain insights from more general data sharing and ML research projects. Through further research, The designed market-based approach can be further developed for organising FL rewards.

Utilizing trustworthiness as a key criterion in the evaluation of FL designs is important for regulated fields in the face of the AI Act demanding explainable AI/ML. The system behaviour was validated through both the requirements and aspects of trustworthy FL. Although many aspects of trustworthiness exist, key trustworthiness factors such as incentive mechanisms and contribution evaluation are reiterated as important and should therefore remain a key aspect in the design around FL as both a design goal and method of evaluation design.

The design process provided insights into the differing methods and design alternatives that can be used and combined with functional artefacts. Building artefact designs around the myriad of possible implementations and methods that are available is a challenging exercise, but the potential success of

a modular fit-for-purpose design strategy that is exampled here for aviation can be adapted for other fields.

8.4. Reflections

This thesis used a variety of methods and resulted in a design based heavily on another thesis as a precursor, so some reflections are in order regarding the methods used, the design itself some assumptions made about it, and general limitations of the research.

8.4.1. Methods

The broader methodological selection was a DSR research based on work by van de Walle, 2024 shaped as a research further designing and actually implementing the rewarding aspect of the FL architecture for aviation. The steps were adapted from the DSR body of work but were appended with an institutional analysis to provide some more background, help the design process and legitimise the work as a Cosem master thesis. The design steps themselves, the addition of the DSR uncommon institutional step and the time spent actually engineering smart contracts introduced some challenges and influenced the research results.

A key aspect of the research to reflect on was the reliance on van de Walle's thesis on FL architecture. This thesis relied heavily on this and had effects on the results. Specifically, requirements and the problem field came from that thesis, which itself did contain many validating aspects for the work, but could not always be utilized to validate the requirements and problem field for this research. As such, even though many validated aspects were taken over, the methods relied on some aspects too much of van de Walle's work. If that work did have issues impacting its worth, this research would become much less relevant as the problem and requirements from van de Walle's work shaped a lot of the methods and processes for this DSR. However, enough additional and unique work in the institutional analysis, adaptation on van de Walle's requirement and especially design and implementation was done that relied less on that thesis, so the work performed here can still stand on its own.

For a Cosem thesis about design, a balance had to be made between actual engineering, where a product is being made versus design science which has a more rigorous approach. This balance was attempted, but a clearer decision from the initial start of the thesis to focus heavily on design or on engineering would have made a more clear-cut thesis that either goes into more development time or has more critically reflecting design steps.

Some steps of the DSR had more activities in them than others: the requirement and problem specification steps in particular were heavily based on the work by van de Walle, with most of the requirements being taken from that thesis. The method of adjusting them was based on argumentations and some literature works but could have been verified more. Additionally, the problem specification was taken as a given and only expanded further in the introduction with potentially lacking sources from the industry itself specific for this case.

The iterative nature of many of the steps was challenging to report on, with a creative process that dictated some of the initial design process being perhaps not the most rigorous scientific method. As such, the repeatability of some aspects of the research and the illustration of the exact sources for design insights were difficult to capture. Going back and forth between steps, mostly about the design and validation, made for a messy process that could have been captured and demarcated earlier by setting a set amount of time or iterations between validation steps, design and implementations.

Alternative generation and making the design choices were not based on very rigorous or validated methods, stemming mostly from the author's experience and other works or projects found. This should have been clarified beforehand, with a strict approach to finding alternatives and design options per design area of interest, perhaps based on an additional dedicated alternative finding literature review. Currently, the design can suffer from biases by the author based on experience or a lack of knowledge of certain solutions.

8.4.2. Institutional methods

The institutional aspect was difficult to fit within the DSR process and should have been fitted better within the design steps. A more valid approach was to design the institutional setup as a DSR itself,

either being a design project within the design or being a completely separate entity. This work put the institutional works kind of in the middle, drawing inspiration and insights from the institutional analysis without verifying and validating much of the work done in that section. Utilizing more traditional institutional design language such as the grammar for rules could have made the work more valid and understandable. The resulting institutional insights are therefore harder to utilize by other research as its less valid and are currently for this thesis mostly useful only as a design step.

The institutional analysis result converged quickly on the already defined market-based model sale through van de Walle's work and author bias. Other designs and variants should have been considered, as they also provide different solutions to the design issue. A governmentally subsidized research group that does FL or a non-profit cooperative that is guided by a consortium could have been valid alternative options that were not explored at all. The impact of the lack of broader exploration of such options weakens further the institutional validity of the design.

Real-world actor behaviour was not captured well and not measured, interviews could have been added or numerical simulations could have predicted system behaviour with differing results. As of now, the design assumes much of the behaviour from the non-validated institutional design, lowering the validity of the design. As such, the design should be tested more either in scaled-up situations or with expert interviews. Inspiration was taken from the responses of expert interviews in van de Walle's research however, slightly mitigating this issue.

In future similar research that does want to have a similar setup to this DSR with an institutional design based directly on other work, key changes regarding the methods should include: Having a clear idea of the function and fit of institutional analysis, a plan to argue for and explain all the design steps, plan for some kind of real-world validation such as interviews or simulations and make sure the decision between engineering and designing is explained and argued for.

8.4.3. Design

Reflections on the design itself consist of insights from design choices and the experimental results. The design itself, a set of smart contracts that interact with each other for FL task organisation and reward, and a black-box periphery that does the actual FL process is based on an institutional design and was also implemented and tested.

Some naive choices were made in the process, the biggest being the black-boxing of much of the FL process. This can reduce validity as unexpected and unexplored system behaviours can influence the design. Having no simulated or implemented FL that runs a training task and calculates the participant contribution that interacts with the smart contracts does not impact the design itself, but the potential and currently unknown behaviours can have an effect on the usability in future implementations.

Having the design be purely blockchain was a very early design decision based on van de Walle's work, the author's intention and experience, and a reliance on smart contracts. This design choice is argued for, but some bias was definitely present in this decision. The result of this bias indicates that an exploration of potential alternative infrastructure design space is relevant to include in further explorations of this design work.

The token design introduced in this design rewards participants with both a reward ERC-20 token and a Model Access Token, forgoing direct payments and rewards. This design choice heavily relies on the model being functional and proving some usefulness: If the FGM does not perform and the training tasks consistently do not produce useful models, the partnership does not provide any value at all. Even though this is as expected, as a consortium producing worthless models should be worthless, the fear of participating without expected rewards is not fully captured by the token design, as no baseline reward exists. In successful collaborations, however, the tokens get worth more with both the model sales increasing for more kick-back rewards and interest in joining can grow.

The MAT as an access token has some concerns regarding usability and transferability both within and outside the private chain. Transferring the MAT to another address means giving them both access and transfer rights, meaning that access to the model can be moved to parties that might try to reverse data from the model, and might be completely unknown in their intentions or origins. Additional protection can be added in the interface between the model server and potential users such as IP checks or the

verification of the real identity of a pseudonymous wallet address before giving actual access. Additionally, the capacity token having the token be sellable on a public-facing blockchain, which then requires interaction between different blockchains, is not fully explored and has some ramifications: The other chain needs to trust the private chain fully to have the MAT on the public chain be a reliable commodity. This requires some cryptographic proof or some other method of organising the public sale of the MAT, either as a separate entity on a public chain or by selling on the private chain only for direct payments to the consortium, bypassing the blockchain.

The results of the tests on speed and scalability tested the response times of the contract calls, file sizes and gas fees. Although useful insights, because of the iterative approach, these tests, if not satisfactory, could have resulted in adjustments of the implementation to improve the tests. This means the test results in their current state indicate the best results within the author's engineering abilities and can be improved on by a more experienced Solidity engineer.

The impact of these found issues on the design and the usability for actual implementations and future research include the need for further validation of the impact of actual FL tasks and the features of such a task, such as speed and impact on actor behaviour and the long term effect of token rewarding. The actual implementation in a real-world application should take note of the reliance on OpenZeppelin standards, the developments in the EVM development sector and other potential BC options, as well as a need for engineering knowledge around smart contracts and data discovery, and most importantly, selection of contribution score calculators.

8.5. Future research

Significance of this research is from the dual-token design and as an implementation of an important aspect of the previously designed FL architecture for Aviation. Future directions of research are indicated from this work for collaboration designs for FL, other usages of this design and testing directions. Exploring the institutional framework's validity in different collaboration designs, particularly in environments with varying levels of trust requirements or alternative governance structures. This could include public collaborations or scenarios where participants have less inherent trust in each other, providing insights into the robustness and flexibility of the system. Testing and further validation of both design and institutional fields can be done using serious gaming and simulations to gain insight into more complex system behaviour as models. Furthermore, testing the behaviour of actors in the system in different situations using game theory models, simulations or gamification could also strengthen insight into the design. The potential applicability to non-FL machine learning designs could validate its broader utility in collaborative learning environments.

9

Conclusion

Designing around cutting-edge developments such as Federated Learning for a highly regulated and complex sector such as aviation is a challenge where aspects of compliance, dynamics and technical solutions are found. This thesis performed design science research and institutional analysis of incentivization and organisation structures for FL in aviation, resulting in both a technical artefact and theoretical insights. The goal and research question was about: *How can a blockchain-based incentive organizing system be designed and implemented to increase participation and trust in federated learning systems within aviation partnerships?*

The research direction was explored by splitting up the research into steps based on the Design Science Research methodology and with the addition of an institutional analysis to understand the context and situation of the participants. The research was based on previous work that defined an architecture for FL in aviation, wherein rewards in a consortium of aviation players were distributed based on their contribution. The problem area was further defined as one of trustworthy organisation and proper incentivization for participation in the face of high costs and free-riding risks.

The requirements from the previous work were adapted and refocused into 10 requirements, one of which was non-functional, that shaped further design based on security, regulations, incentives, governance and modularity in the face of a changing regulatory and technical environment. An institutional analysis resulted in a simple design of the basic organisation and functioning of a data collaborative that fulfils FL tasks controlled by a consortium with actors in the roles of participant, and data owner, participating in different manners and support, overseen by regulatory bodies.

Through iterative design steps, this research further developed as an artefact a system for FL collaboratives in aviation: a private blockchain-based smart contract architecture for managing incentives, collaboration and access control with a dual-token reward system providing incentives as both financial and governance rights. The tokens, representing model access and the reward, can be distributed based on a set distribution amount, and can be utilized for model sale, additional voting power in decisions and can represent partial ownership of the system to receive any profits from model distribution. The design was purposely made relatively sparse of FL and contribution calculation methods internally and was modular to allow for future adaptations of technologies or regulatory adaptations.

The artefact was tested and validated, showing that, with abstracted FL processes and some technical assumptions, smart contracts can be used as infrastructure for incentive mechanisms while also orchestrating the tasks themselves. Using basic tests for functionality as well as tests for speed, scalability and complexity this MVP is expected to be able to scale and be adapted for future usage and research.

Although the design has limitations around real-world testing and applications, it provides a strong foundation for future research steps for trustworthy FL. Modular design is useful to deal with the evolving FL landscape in aviation, both regulatory as well as technically. Future research direction should focus on real-world implementations and validation using alternative incentives and contribution calculation

mechanisms, as well as testing methods with real-world situations and expert reviews, broadening the validity of the design.

This thesis shows the design and demonstration of the effectiveness of a blockchain-based method of FL control and reward using a dual-token system, although significant work still needs to be done to transform the design to a validated method usable in production environments. The challenge of design around FL incentivization in aviation remains complex, but this design science research adds at least one part of the puzzle.

References

Abuzied, Y., Ghanem, M., Dawoud, F., Gamal, H., Soliman, E., Sharara, H., & ElBatt, T. (2024). A privacy-preserving federated learning framework for blockchain networks. *Cluster Computing*, *27*(4), 3997–4014. https://doi.org/10.1007/s10586-024-04273-1

Adryan, F. A., & Sastra, K. W. (2021). Predictive maintenance for aircraft engine using machine learning: Trends and challenges [Number: 1]. *AVIA*, *3*(1). https://doi.org/10.47355/avia.v3i1.45

Amidi, A., Giannopoulos, A., & Trakadas, P. (2023). Federated learning techniques: From ethics to regulation. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4588972

Apostolidis, A., Bouriquet, N., & Stamoulis, K. P. (2022). AI-Based Exhaust Gas Temperature Prediction for Trustworthy Safety-Critical Applications. *Aerospace*, *9*(11), 722. https://doi.org/10.3390/aerospace9110722

Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., & Jararweh, Y. (2022). Federated learning review: Fundamentals, enabling technologies, and future applications. *Information Processing And Management*, *59*(6), 103061. https://doi.org/https://doi.org/10.1016/j.ipm.2022.103061

Bao, X., Su, C., Xiong, Y., Huang, W., & Hu, Y. (2019). Flchain: A blockchain for auditable federated learning with trust and incentive. *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, 151–159. https://doi.org/10.1109/BIGCOM.2019.00030

Bartolomucci, F., & Leoni, F. (2024). Designing an effective governance model for data collaboratives. *Research-Technology Management*, *67*(4), 49–61. https://doi.org/10.1080/08956308.2024.2351331

Bi, X., Gupta, A., & Yang, M. (2024). Understanding partnership formation and repeated contributions in federated learning: An analytical investigation. *Management Science*, *70*(8), 4974–4994.

Bonawitz, K. (2019). Towards federated learning at scale: Syste m design. *arXiv preprint arXiv:1902.01046*.

Bouacida, N., & Mohapatra, P. (2021). Vulnerabilities in federated learning. *IEEE Access*, *9*, 63229–63249. https://doi.org/10.1109/ACCESS.2021.3075203

Bugshan, N., Khalil, I., Rahman, M. S., Atiquzzaman, M., Yi, X., & Badsha, S. (2023). Toward trustworthy and privacy-preserving federated deep learning service framework for industrial internet of things. *IEEE Transactions on Industrial Informatics*, *19*(2), 1535–1547. https://doi.org/10.1109/TII.2022.3209200

Cai, H., Gao, L., Wang, J., & Li, F. (2024). Reliable incentive mechanism in hierarchical federated learning based on two-way reputation and contract theory. *Future Generation Computer Systems*, *159*, 533–544. https://doi.org/10.1016/j.future.2024.05.045

Chen, P., Du, X., Lu, Z., Wu, J., & Hung, P. C. (2022). EVFL: An explainable vertical federated learning for data-oriented artificial intelligence systems. *Journal of Systems Architecture*, *126*, 102474. https://doi.org/10.1016/j.sysarc.2022.102474

Commission, E. (2020, February 19). *Shaping europe's digital future: Commission presents strategies for data and artificial intelligence*. Retrieved October 8, 2024, from https://ec.europa.eu/commission/presscorner/detail/en/ip_20_273

Deng, W., Li, X., Xu, J., Li, W., Zhu, G., & Zhao, H. (2024). Bfkd: Blockchain-based federated knowledge distillation for aviation internet of things. *IEEE Transactions on Reliability*.

DuPont, Q. (2017). Experiments in algorithmic governance. *Bitcoin and beyond*, 157.

EASA. (2023, May 10). *Artificial intelligence roadmap 2.0* (Roadmap). EASA.

EUROCONTROL. (n.d.). *Embracing artificial intelligence to respond to the challenges in aviation* [Eurocontrol]. Retrieved July 8, 2024, from https://www.eurocontrol.int/artificial-intelligence

Guo, L., Wei, Q., & Chenyu, H. (2021). Research on flight delay prediction based on horizontal and vertical federated learning framework. *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 38–44.

Hess, C., & Ostrom, E. (2005). A framework for analyzing the knowledge commons: A chapter from understanding knowledge as a commons: From theory to practice.

Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep models under the GAN: Information leakage from collaborative deep learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 603–618. https://doi.org/10.1145/3133956.3134012

ICAO. (2022). *ICAO Environmental Report 2022* (tech. rep.). ICAO. Retrieved January 11, 2024, from https://www.icao.int/environmental-protection/Pages/envrep2022.aspx

Jagatheesaperumal, S. K., Rahouti, M., Alfatemi, A., Ghani, N., Quy, V. K., & Chehri, A. (2024). Enabling trustworthy federated learning in industrial iot: Bridging the gap between interpretability and robustness. *IEEE Internet of Things Magazine*, *7*(5), 38–44. https://doi.org/10.1109/IOTM.001.2300274

Jin, X., Chen, P.-Y., Hsu, C.-Y., Yu, C.-M., & Chen, T. (2021). Cafe: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems*, *34*, 994–1006.

Johannesson, P., & Perjons, E. (2021). *An introduction to design science* (Second edition). Springer.

Kaal, W. A. (2021). Blockchain solutions for agency problems in corporate governance. In *Information for efficient decision making: Big data, blockchain and relevance* (pp. 313–329). World Scientific.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Nitin Bhagoji, A., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., … Zhao, S. (2021). Advances and open problems in federated learning [Number: 1–2]. *Foundations and Trends® in Machine Learning*, *14*(1), 1–210. https://doi.org/10.1561/2200000083

Keller, R. M. (2016). Ontologies for aviation data management. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 1–9.

KLM. (n.d.). CAO voor KLM-vliegers op vleugelvliegtuigen. https://www.uitvoeringarbeidsvoorwaardenwetgeving.nl/mozard/document/docnr/1344134,,926616,/CAO%20voor%20KLM-vliegers%20op%20vleugelvliegtuigen%20%282014-2023%29%20%28schoon%29.pdf

Kyriakou, K., & Otterbacher, J. (2023). In humans, we trust: Multidisciplinary perspectives on the requirements for human oversight in algorithmic processes. *Discover Artificial Intelligence*, *3*(1), 44. https://doi.org/10.1007/s44163-023-00092-2

Li, D., Han, D., Weng, T.-H., Zheng, Z., Li, H., Liu, H., Castiglione, A., & Li, K.-C. (2022). Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey [Number: 9]. *Soft Computing*, *26*(9), 4423–4440. https://doi.org/10.1007/s00500-021-06496-5

Liu, Y., Fan, T., Chen, T., Xu, Q., & Yang, Q. (2021). Fate: An industrial grade platform for collaborative learning with data protection. *Journal of Machine Learning Research*, *22*(226), 1–6. http://jmlr.org/papers/v22/20-815.html

Liu, Z., Chen, Y., Yu, H., Liu, Y., & Cui, L. (2022). GTG-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on Intelligent Systems and Technology*, *13*(4), 1–21. https://doi.org/10.1145/3501811

Ma, C., Li, J., Shi, L., Ding, M., Wang, T., Han, Z., & Poor, H. V. (2022). When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Computational Intelligence Magazine*, *17*(3), 26–33.

Ma, Y., Sun, Y., Lei, Y., Qin, N., & Lu, J. (2020). A survey of blockchain technology on security, privacy, and trust in crowdsourcing services. *World Wide Web*, *23*, 393–419.

Majeed, U., Khan, L. U., Hassan, S. S., Han, Z., & Hong, C. S. (2023). FL-incentivizer: FL-NFT and FL-tokens for federated learning model trading and training. *IEEE Access*, *11*, 4381–4399. https://doi.org/10.1109/ACCESS.2023.3235484

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2023, January). Communication-Efficient Learning of Deep Networks from Decentralized Data [arXiv:1602.05629 [cs]]. https://doi.org/10.48550/arXiv.1602.05629
Comment: [v4] Fixes a typo in the FedAvg pseudocode. [v3] Updates the large-scale LSTM experiments, along with other minor changes.

Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. *2019 IEEE symposium on security and privacy (SP)*, 691–706.

Nguyen, T. A., Kaliappan, V. K., Jeon, S., Jeon, K.-s., Lee, J.-W., & Min, D. (2021). Blockchain empowered federated learning with edge computing for digital twin systems in urban air mobility. *Asia-Pacific International Symposium on Aerospace Technology*, 935–950.

Oehling, J., & Barry, D. J. (2019). Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data. *Safety Science*, *114*, 89–104. https://doi.org/10.1016/j.ssci.2018.12.018

Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge university press.

Ostrom, E. (2011). Background on the institutional analysis and development framework. *Policy Studies Journal*, *39*(1), 7–27. https://doi.org/10.1111/j.1541-0072.2010.00394.x

Ouadrhiri, A. E., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE Access*, *10*, 22359–22380. https://doi.org/10.1109/ACCESS.2022.3151670

Peregrina, J. A., Ortiz, G., & Zirpins, C. (2022). Towards data governance for federated machine learning [Series Title: Communications in Computer and Information Science]. In C. Zirpins, G. Ortiz, Z. Nochta, O. Waldhorst, J. Soldani, M. Villari, & D. Tamburri (Eds.), *Advances in service-oriented and cloud computing* (pp. 59–71, Vol. 1617). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-23298-5_5

Rejeb, A., Keogh, J. G., Simske, S. J., Stafford, T., & Treiblmaier, H. (2021). Potentials of blockchain technologies for supply chain collaboration: A conceptual framework. *The International Journal of Logistics Management*, *32*(3), 973–994.

Ruijer, E. (2021). Designing and implementing data collaboratives: A governance perspective. *Government Information Quarterly*, *38*(4), 101612. https://doi.org/10.1016/j.giq.2021.101612

Sanfilippo, M., Frischmann, B., & Standburg, K. (2018). Privacy as commons: Case evaluation through the governing knowledge commons framework. *Journal of Information Policy*, *8*, 116–166. https://doi.org/10.5325/jinfopoli.8.2018.0116

Saraf, A. P., Chan, K., Popish, M., Browder, J., & Schade, J. (2020). Explainable artificial intelligence for aviation safety applications. *AIAA AVIATION 2020 FORUM*. https://doi.org/10.2514/6.2020-2881

Sharma, T., Potter, Y., Pongmala, K., Wang, H., Miller, A., Song, D., & Wang, Y. (2024). Unpacking how decentralized autonomous organizations (daos) work in practice. *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 416–424. https://doi.org/10.1109/ICBC59979.2024.10634404

Shayan, M., Fung, C., Yoon, C. J., & Beschastnikh, I. (2020). Biscotti: A blockchain system for private and secure federated learning. *IEEE Transactions on Parallel and Distributed Systems*, *32*(7), 1513–1525.

Shi, Z., Zhang, L., Yao, Z., Lyu, L., Chen, C., Wang, L., Wang, J., & Li, X.-Y. (2024). FedFAIM: A model performance-based fair incentive mechanism for federated learning. *IEEE Transactions on Big Data*, *10*(6), 1038–1050. https://doi.org/10.1109/TBDATA.2022.3183614

Song, T., Tong, Y., & Wei, S. (2019). Profit allocation for federated learning. *2019 IEEE International Conference on Big Data (Big Data)*, 2577–2586. https://doi.org/10.1109/BigData47090.2019.9006327

Stanton, I., Munir, K., Ikram, A., & El□Bakry, M. (2023). Predictive maintenance analytics and implementation for aircraft: Challenges and opportunities [Number: 2]. *Systems Engineering*, *26*(2), 216–237. https://doi.org/10.1002/sys.21651

Tariq, A., Serhani, M. A., Sallabi, F. M., Barka, E. S., Qayyum, T., Khater, H. M., & Shuaib, K. A. (2024). Trustworthy federated learning: A comprehensive review, architecture, key challenges, and future research prospects. *IEEE Open Journal of the Communications Society*, *5*, 4920–4998. https://doi.org/10.1109/OJCOMS.2024.3438264

Tastan, N., Fares, S., Aremu, T., Horvath, S., & Nandakumar, K. (2024). Redefining contributions: Shapley-driven federated learning [Version Number: 1]. https://doi.org/10.48550/ARXIV.2406.00569

Tretheway, M. W., & Markhvida, K. (2014). The aviation value chain: Economic returns and policy issues [Special Issue of the First European Aviation Conference Re-Inventing the Aviation Value Chain Nov 22-23, 2012]. *Journal of Air Transport Management*, *41*, 3–16. https://doi.org/https://doi.org/10.1016/j.jairtraman.2014.06.011

Truong, N., Sun, K., Wang, S., Guitton, F., & Guo, Y. (2021). Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security*, *110*, 102402. https://doi.org/10.1016/j.cose.2021.102402

van Binsbergen, L. T., Kebede, M. G., Baugh, J., van Engers, T., & van Vuurden, D. G. (2022). Dynamic generation of access control policies from social policies [12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare]. *Procedia Computer Science*, *198*, 140–147. https://doi.org/https://doi.org/10.1016/j.procs.2021.12.221

van de Walle, B. (2024, June). *Architectural framework for federated learning in aviation maintenance* [Msc Thesis]. TU Delft.

Wachter, S. (2024). Limitations and loopholes in the eu ai act and ai liability directives: What this means for the european union, the united states, and beyond. *Yale Journal of Law and Technology*, *26*(3).

Wang, Y., & Kantarci, B. (2021). Reputation-enabled federated learning model aggregation in mobile platforms. *ICC 2021 - IEEE International Conference on Communications*, 1–6. https://doi.org/10.1109/ICC42927.2021.9500928

Wang, Z., Yan, B., & Dong, A. (2022). Blockchain empowered federated learning for data sharing incentive mechanism. *Procedia Computer Science*, *202*, 348–353. https://doi.org/10.1016/j.procs.2022.04.047

Wilson, D. S., Ostrom, E., & Cox, M. E. (2013). Generalizing the core design principles for the efficacy of groups [Evolution as a General Theoretical Framework for Economics and Public Policy]. *Journal of Economic Behavior And Organization*, *90*, S21–S32. https://doi.org/https://doi.org/10.1016/j.jebo.2012.12.010

Woisetschläger, H., Erben, A., Marino, B., Wang, S., Lane, N. D., Mayer, R., & Jacobsen, H.-A. (2024). Federated learning priorities under the european union artificial intelligence act [Version Number: 1]. https://doi.org/10.48550/ARXIV.2402.05968

Woisetschläger, H., Mertel, S., Krönke, C., Mayer, R., & Jacobsen, H.-A. (2024). Federated learning and AI regulation in the european union: Who is responsible? – an interdisciplinary analysis [Version Number: 2]. https://doi.org/10.48550/ARXIV.2407.08105

Wu, C., Wang, Y., & Tao, L. (2024). Machine learning-enabled techno-economic uncertainty analysis of sustainable aviation fuel production pathways. *Chemical Engineering Journal Advances*, *20*, 100650. https://doi.org/10.1016/j.ceja.2024.100650

Xu, R., Baracaldo, N., Zhou, Y., Anwar, A., Kadhe, S., & Ludwig, H. (2022, July 15). DeTrust-FL: Privacy-preserving federated learning in decentralized trust setting. Retrieved August 30, 2024, from http://arxiv.org/abs/2207.07779

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications [Number: 2]. *ACM Transactions on Intelligent Systems and Technology*, *10*(2), 1–19. https://doi.org/10.1145/3298981

Zeng, D., Liang, S., Hu, X., Wang, H., & Xu, Z. (2023). Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, *24*(100), 1–7.

Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P., & Guo, S. (2021). A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 1–1. https://doi.org/10.1109/TETC.2021.3063517

Zhao, H., Sui, M., Liu, M., Zhu, C., Xun, W., Xu, B., & Zhu, H. (2024). Are you diligent, inefficient, or malicious? a self-safeguarding incentive mechanism for large-scale federated industrial maintenance based on double-layer reinforcement learning. *IEEE Internet of Things Journal*, *11*(11), 19988–20001. https://doi.org/10.1109/JIOT.2024.3367875

Zuziak, M. K., Hinrichs, O., Abdrassulova, A., & Rinzivillo, S. (2023). Data collaboratives with the use of decentralised learning. *2023 ACM Conference on Fairness, Accountability, and Transparency*, 615–625. https://doi.org/10.1145/3593013.3594029

A

Contract UMLs

Found here are UML-like diagrams with the most important aspects of each contract's implementation and functions.

```
FLGovernor is Governor

constructor (_token)

votingDelay()
votingPeriod()
quorum()
proposalThreshold()

#inherited:
propose(address[] targets, uint256[] values, bytes[] calldatas, string description)
```

```
RewardToken is ERC20


constructor(address admin, address minter){
   _grantRole(DEFAULT_ADMIN_ROLE, admin)
   _grantRole(MINTER_ROLE, minter)
}

mint(address to, uint256 amount) onlyRole(MINTER_ROLE){
   _mint(to, amount)
}
```

```
ModelAccessToken is ERC1155


constructor() ERC1155("URL") {
        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _grantRole(MINTER_ROLE, msg.sender);
        _grantRole(URI_SETTER_ROLE, msg.sender);
    }

mint(address account, uint256 id, uint256 amount, bytes memory data) onlyRole(MINTER_ROLE){
}

mintBatch(address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data)
onlyRole MINTER_ROLE{
   _mintBatch(to, ids, amounts, data);
}

burnForModelAccess( unit256 taskId, address holder) {
   burn(msg.sender, taskId, amount)
   emit ModelAccessUsed (taskId, msg.sender)
}
```

```
FLOrchestrator is AccessControl

IRewardsCalculator = FLRewarder.address

constructor(){
   _grantRole(DEFAULT_ADMIN_ROLE, sender)
   _grantRole(ORCHESTRATOR_ROLE, sender)
}

enum TaskStatus{
PROPOSED
STARTED
FINISHED

struct Task{
uint 256 id
TaskStatus status
uint256 currentRound
address[] participants
bool exists
}

event TaskCreated
event TaskStatusUpdated
event ParticipantAdded
event ParticipantRegistered
event RoundUpdated

proposeTask(address[] participants) onlyRole ORCHESTRATOR_ROLE{
newTask{id, TaskStatus.PROPOSED, currentRound0, participants, exists: true}
}

startTask(taskID) onlyRole ORCHESTRATOR_ROLE{
task.status = TaskStatus.STARTED
}

registerForTask(taskID){
task.participants.push(sender)
}

progressToNextRound(taskID) onlyRole ORCHESTRATOR_ROLE{
task.currentRound++
}

finishTask(taskID) onlyRole ORCHESTRATOR_ROLE{
task.TaskStatus.FINISHED
rewardsCalculator.calculateRewards(taskID, participants)
}

isParticipant(taskID, participant) returns bool
```

```
FLRewarder is AccessControl

IFLRToken = RewardToken.address
IModelAccessToken = ModelAccessToken.address
IFLOrchestrator = FLOrchestrator.address

constructor(){
   _grantRole(DEFAULT_ADMIN_ROLE, sender)
   _grantRole(REWARDER_ROLE, sender)
   _grantROLE(CALCULATOR_ROLE, sender)
}

struct ParticipantScore{
roundScores
totalScore
bool hasContributed
}

struct ModelInformation{
string taskName
string modelType
uint256 participantCount
uint256 totalRounds
uint256 finalAccuracy
}

struct TaskRewards {
   taskID
   uint256 RewardTokenPool
   uint256 ModellAccessAmount
}

recordRoundScores(taskID, round, participants, scores) onlyRole CALCULATOR_ROLE{
for each participant in task:(
   score = roundscore
   hasContributed= true
)}

finalizeTaskScores(taskID, round, participants, scores) onlyRoleCALCULATOR_ROLE{
for each participant in task:(
   score = sum of all roundscore
   hasContributed= true
)}

claimRewards(taskID) {
   require(sender !hasClaimed)
   require(orchestrator.isParticipant)
   require(sender hasContributed)

   RewardTokenAmount = (RewardTokenPool / score) / totalTaskScore

   RewardToken.mint(sender, RewardTokenAmount)
```

B

Deployment files

Using Hardhat ignition to deploy the files in the /contracts directory

```
const GOVERNANCE_PARAMS = {
    VOTING_DELAY: 1,                // 1 block
    VOTING_PERIOD: 50400,           // ~1 week with 12 sec block time
    QUORUM_PERCENTAGE: 4,           // 4%
  };

  const ROLES = {
    MINTER_ROLE: ethers.utils.keccak256(ethers.utils.toUtf8Bytes("MINTER_ROLE")),
    ORCHESTRATOR_ROLE: ethers.utils.keccak256(ethers.utils.toUtf8Bytes("ORCHESTRATOR_ROLE")),
    URI_SETTER_ROLE: ethers.utils.keccak256(ethers.utils.toUtf8Bytes("URI_SETTER_ROLE")),
  };

  module.exports = {
    GOVERNANCE_PARAMS,
    ROLES,
  };
```

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");

module.exports = buildModule("Governance", (m) => {
  const tokens = m.useModule(require("./tokens"));
  const orchestrator = m.useModule(require("./orchestrator"));

  const VOTING_DELAY = 1;
  const VOTING_PERIOD = 50400;
  const QUORUM_PERCENTAGE = 4;

  const governance = m.contract("FLGovernor", [
      tokens.flrToken
    ]
  );

  return { governance };
});
```

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");

module.exports = buildModule("Main", (m) => {
  const tokens = m.useModule(require("./tokens"));
  const orchestrator = m.useModule(require("./orchestrator"));
  const rewarder = m.useModule(require("./rewarder"));
  const governance = m.useModule(require("./governance"));

  return {
    tokens,
    orchestrator,
    rewarder,
```

```
    governance
  };
});
```

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");

module.exports = buildModule("Orchestrator", (m) => {
  const tokens = m.useModule(require("./tokens"));
  const orchestrator = m.contract("FLOrchestrator");

  return { orchestrator };
});
```

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");

module.exports = buildModule("Rewarder", (m) => {
  const tokens = m.useModule(require("./tokens"));
  const orchestrator = m.useModule(require("./orchestrator"));

  const rewarder = m.contract("FLRewarder",
     [
      tokens.flrToken,
      tokens.modelAccessToken,
      orchestrator.orchestrator
    ]
  );

  return { rewarder };
});
```

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");

module.exports = buildModule("Tokens", (m) => {
  // Deploy FLRToken with defaultAdmin and minter addresses
  const flrToken = m.contract("FLRToken", [
      "0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266", // defaultAdmin
      "0x70997970C51812dc3A010C7d01b50e0d17dc79C8"  // minter
    ]
  );
  const modelAccessToken = m.contract("ModelAccessToken", []);
  return { flrToken, modelAccessToken };
});
```

C

Testing Code

The following Python code tests the deployed contract on a local chain. To replicate the tests please follow these steps:

1. Deployment of the contracts found in the GitHub on https://github.com/BrunoLapre/FLDesign
2. Adjust the values in the Python script to align with the local file locations and deployments
3. Run the script in a notebook or interactive window

```
from web3 import Web3
import json
from typing import Dict, Any, List, Tuple
import time
import numpy as np
import matplotlib.pyplot as plt

BESU_RPC_URL = "http://localhost:8545"
web3 = Web3(Web3.HTTPProvider(BESU_RPC_URL))
#Public test account 0xfe3b557e8fb62b89f4916b721be55ceb828dbd73 keys
PRIVATE_KEY = "0x8f2a55949038a9610f50fb23b5883af3b4ecb3c3bb792cbcefbd1542c692be63"
#This should look like /../deployments/chain-1337/
FILE_LOC = 'Deployment Location here'

with open(f'{FILE_LOC}deployed_addresses.json', 'r') as file:
    contracts = json.load(file)

# Load contract ABI
def load_contract_abi(contract_name: str) -> Dict[str, Any]:
    with open(f'{FILE_LOC}{contract_name}.json', 'r') as file:
        contract_data = json.load(file)
        return contract_data['abi']

# --- Contract Setup (Keep this) ---
orchestrator = web3.eth.contract(
    address = web3.to_checksum_address(contracts['Orchestrator#FLOrchestrator']),
    abi = load_contract_abi("Orchestrator#FLOrchestrator")
)
rewarder = web3.eth.contract(
    address = web3.to_checksum_address(contracts['Rewarder#FLRewarder']),
    abi = load_contract_abi("Rewarder#FLRewarder")
)
account = web3.eth.account.from_key(PRIVATE_KEY)


# --- Transaction Sending Functions (Modified to return tx_hash) ---
def proposeTask_tx(orchestrator, taskId):
    prop_tx = orchestrator.functions.proposeTask([account.address],5000,5000).
        build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
```

```
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(prop_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def startTask_tx(orchestrator, taskId):
    start_tx = orchestrator.functions.startTask(taskId).build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(start_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def progressTask_tx(orchestrator, taskId):
    next_tx = orchestrator.functions.progressToNextRound(taskId).build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(next_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def finishTask_tx(orchestrator, taskId):
    f_tx = orchestrator.functions.finishTask(taskId,[50000]).build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(f_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def roundScore_tx(rewarder, taskId):
    rec_tx = rewarder.functions.recordRoundScores(taskId,0,[account.address],[1234]).
        build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(rec_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def calcRewards_tx(rewarder, taskId):
    calc_tx = rewarder.functions.calculateRewards(taskId,[account.address],[5000],5000,500).
        build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
    signed_tx = web3.eth.account.sign_transaction(calc_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash

def claimRewards_tx(rewarder, taskId):
    claim_tx = rewarder.functions.claimRewards(taskId).build_transaction({
        'from': account.address,
        'nonce': web3.eth.get_transaction_count(account.address),
        'gas': 2000000,
        'gasPrice': web3.eth.gas_price
    })
```

```
    signed_tx = web3.eth.account.sign_transaction(claim_tx, PRIVATE_KEY)
    tx_hash = web3.eth.send_raw_transaction(signed_tx.raw_transaction)
    return tx_hash


# --- TPS and Gas Measurement Function ---
def measure_tps_and_gas(transaction_function, num_transactions: int, *args_for_tx_func) -> Tuple[float, float, float, float, List[int]]:
    tx_hashes: List[bytes] = []
    start_time = time.perf_counter()
    gas_costs: List[int] = [] # To store gas used for each successful transaction

    print(f"Sending {num_transactions} transactions for {transaction_function.__name__}...")
    for _ in range(num_transactions):
        tx_hash = transaction_function(*args_for_tx_func)
        tx_hashes.append(tx_hash)

    receipts = []
    success_count = 0
    print("Waiting for receipts...")
    for tx_hash in tx_hashes:
        try:
            receipt = web3.eth.wait_for_transaction_receipt(tx_hash, timeout=60) # Timeout
            receipts.append(receipt)
            success_count += 1
            gas_costs.append(receipt.gasUsed) # Record gas used
        except Exception as e:
            print(f"Error waiting for receipt for tx {tx_hash.hex()}: {e}")

    end_time = time.perf_counter()
    elapsed_time = end_time - start_time
    tps = success_count / elapsed_time if elapsed_time > 0 else 0

    avg_gas_cost = np.mean(gas_costs) if gas_costs else 0
    std_dev_gas_cost = np.std(gas_costs) if len(gas_costs) > 1 else 0
    std_dev_tps = 0 # Standard deviation of TPS is less relevant in a single run, can be calculated across multiple runs

    print(f"--- TPS and Gas Test Results for {transaction_function.__name__} ---")
    print(f"Transaction Function: {transaction_function.__name__}")
    print(f"Number of Transactions Sent: {num_transactions}")
    print(f"Successful Transactions: {success_count}")
    print(f"Elapsed Time: {elapsed_time:.2f} seconds")
    print(f"Effective TPS: {tps:.2f} transactions/second")
    print(f"Average Gas Cost: {avg_gas_cost:.2f} gas")
    print(f"Std Dev Gas Cost: {std_dev_gas_cost:.2f} gas")
    print(f"----------------------------------------------------")

    return tps, std_dev_tps, avg_gas_cost, std_dev_gas_cost, gas_costs


def test_tps_and_gas():
    print('Checking which Tasks exists...')
    taskId = 0

    while True:
        try:
            getinfo_tx = orchestrator.functions.getTask(taskId).call()
            if "0xFE3B557E8Fb62b89F4916B721be55cEb828dBd73" in str(getinfo_tx):
                taskId += 1
                continue
            else:
                print(f'{taskId} Does not exist')
                break
        except Exception as e:
            print(f"No taskId {taskId}: {e}")
            break
    task_id_tps = taskId
    num_tx_per_func = 100 # Number of transactions per function to test

    tps_results = {}
```

```
    gas_results = {}
    gas_costs_detail = {} # To store gas costs for boxplot

    test_functions = [
        ("proposeTask", proposeTask_tx, orchestrator, task_id_tps),
        ("startTask",   startTask_tx,   orchestrator, task_id_tps),
        ("progressTask",progressTask_tx, orchestrator, task_id_tps),
        ("finishTask",  finishTask_tx,  orchestrator, task_id_tps),
        ("roundScore",  roundScore_tx,  rewarder, task_id_tps),
        ("calcRewards", calcRewards_tx, rewarder, task_id_tps),
        ("claimRewards",claimRewards_tx, rewarder, task_id_tps),
    ]

    for func_name, tx_func, contract, task_id in test_functions:
        tps, std_dev_tps, avg_gas, std_dev_gas, detailed_gas_costs = measure_tps_and_gas(
            tx_func, num_tx_per_func, contract, task_id)
        tps_results[func_name] = tps
        gas_results[func_name] = avg_gas
        gas_costs_detail[func_name] = detailed_gas_costs # Store for boxplot


    print("\n--- Overall TPS Results ---")
    for method, tps_value in tps_results.items():
        print(f"{method} TPS: {tps_value:.2f}")

    print("\n--- Overall Gas Cost Results ---")
    for method, gas_value in gas_results.items():
        print(f"{method} {method} Avg Gas Cost: {gas_value:.2f}")


    # --- Generate Graphs ---
    methods = list(tps_results.keys())
    tps_values = list(tps_results.values())
    gas_values = list(gas_results.values())
    gas_costs_boxplot_data = [gas_costs_detail[method] for method in methods]


    # TPS Bar Graph
    plt.figure(figsize=(10, 6))
    plt.bar(methods, tps_values, color='skyblue')
    plt.xlabel("Transaction Function")
    plt.ylabel("Transactions Per Second (TPS)")
    plt.title("TPS Performance of Smart Contract Functions")
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout()
    plt.savefig("tps_performance_bar.png") # Save graph to file
    plt.show()


    # Gas Cost Bar Graph
    plt.figure(figsize=(10, 6))
    plt.bar(methods, gas_values, color='lightcoral')
    plt.xlabel("Transaction Function")
    plt.ylabel("Average Gas Cost (gas units)")
    plt.title("Average Gas Cost per Transaction Function")
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout()
    plt.savefig("gas_cost_bar.png") # Save graph to file
    plt.show()

    # Gas Cost Box Plot (Optional - for distribution)
    plt.figure(figsize=(12, 7))
    plt.boxplot(gas_costs_boxplot_data, labels=methods)
    plt.xlabel("Transaction Function")
    plt.ylabel("Gas Cost (gas units)")
    plt.title("Gas Cost Distribution per Transaction Function")
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout()
    plt.savefig("gas_cost_boxplot.png") # Save graph to file
    plt.show()
```

```
249
250 print(f"Connected to network: {web3.is_connected()}")
251 print(f"Current block number: {web3.eth.block_number}")
252
253
254 if __name__ == "__main__":
255     test_tps_and_gas()
```

D

FL Architecture (Walle, 2024)

Sourced from van de Walle, 2024

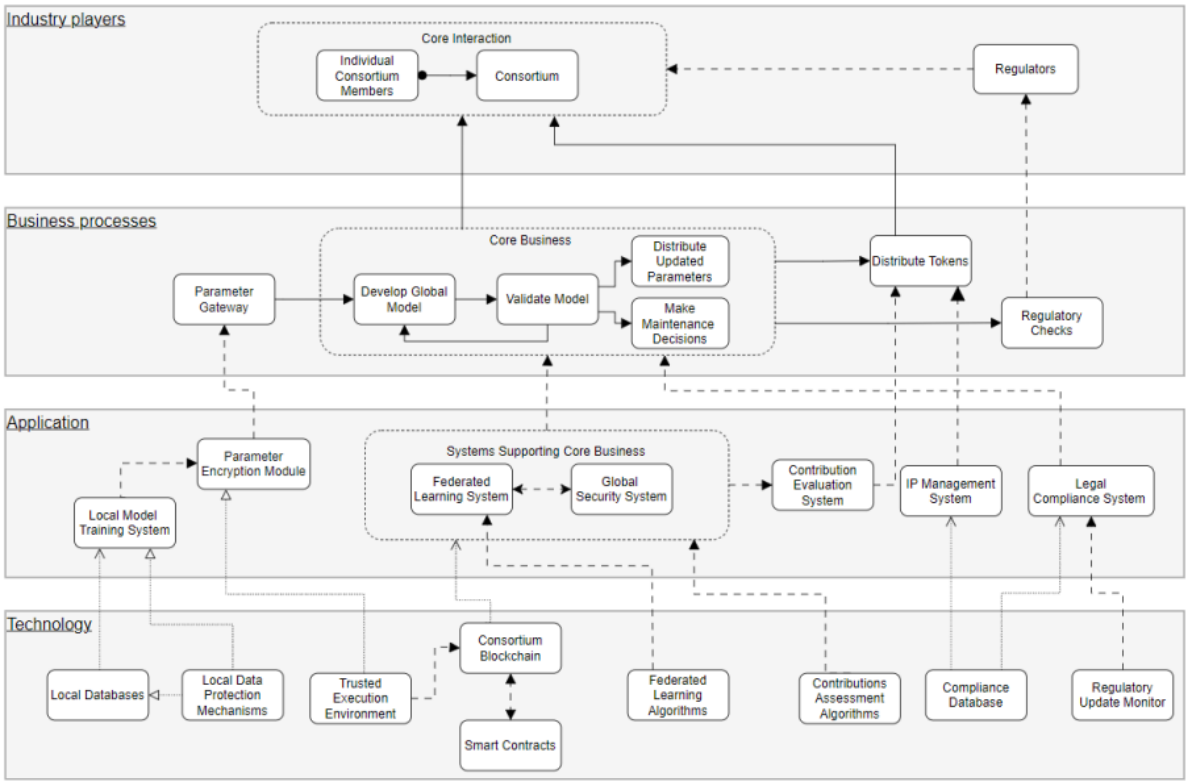

Figure D.1: Validated Architecture model