

AI4Gov

Trusted AI for Transparent Public Governance
fostering Democratic Values

Deliverable 3.1 Decentralized Data Governance, Provenance and Reliability V1


30-10-2023

Version 1.0



Funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Agency. Neither the European Union nor the granting authority can be held responsible for them.

PROPERTIES	
Dissemination level	Public
Version	1.0
Status	Final Version
Beneficiary	UBITECH
License	 <p>This work is licensed under a Creative Commons Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0). See: https://creativecommons.org/licenses/by-nd/4.0/</p>

AUTHORS		
	Name	Organisation
Document leader	Ntalaperas Dimitris	Ubitech
Participants	Xanthi Papageorgiou	Ubitech
	Nikos Kalatzis	Ubitech
	Dimitris Kotios	UPRC
	George Manias	UPRC
Reviewers	Alenka Gucek	JSI
	Matej Kovacic	JSI
	Spiros Borotis	MAG

VERSION HISTORY				
Version	Date	Author	Organisation	Description
0.1	02/10/2023	Ntalaperas Dimitris	Ubitech	ToC
0.5	8/10/2023	Ntalaperas Dimitris, Xanthi Papageorgiou	Ubitech	ToC refinement, SotA, Model
0.6	17/10/2023	Ntalaperas Dimitris, Xanthi Papageorgiou, Nikos Kalatzis	Ubitech	Technology, editing
0.7	18/10/2023	Dimitrios Kotios, George Manias	UPRC	Data Governance Framework, editing
0.9	20/10/2023	Ntalaperas Dimitris, Xanthi Papageorgiou	Ubitech	Prototype description editing, Consolidation of 1 st draft fro internal review
0.9	26/10/2023	Alenka Gucek, Matej Kovacic	JSI	Review comments
1.0	30/10/2023	Dimitris Ntalaperas, Dimitrios Kotios	UBI, UPRC	Final draft addressing internal review
1.1	31/10/2023	Spiros Borotis	MAG	Final review and editing

Table of Contents

Abstract	7
1 Introduction.....	8
1.1 Purpose and scope.....	8
1.2 Document structure.....	8
1.3 Previous iterations	9
2 State of the Art	10
2.1 Blockchain technology	10
2.1.1 <i>Public Blockchains</i>	13
2.1.2 <i>Permissioned Blockchains</i>	13
2.1.3 <i>Read-write access</i>	14
2.2 Smart Contracts	15
2.3 Decentralised Storage.....	16
2.4 Blockchain Governance.....	18
2.4.1 <i>Off-chain governance</i>	19
2.4.2 <i>On-chain governance</i>	20
2.4.3 <i>Governance in HyperLedger Fabric</i>	20
3 Decentralisation in AI4Gov	22
3.1 Data information in AI4Gov	22
3.2 Decentralised Data Storage in the AI4Gov platform.....	25
3.3 Architecture for Decentralised Data Governance.....	27
3.3.1 <i>Business layer</i>	27
3.3.2 <i>Application layer</i>	29
3.3.3 <i>General requirements</i>	30
4 Technological enablers	34
4.1.1 <i>HyperLedger Fabric</i>	34
4.1.2 <i>Governance mechanisms under HyperLedger Fabric</i>	43
4.1.3 <i>OpenDSU</i>	46
5 Data Governance Framework	48
5.1 General Guidelines and Policies.....	48
5.2 Applicable Regulations and EU Guidelines.....	51
5.2.1 <i>General Data Protection Regulation (GDPR)</i>	51
5.2.2 <i>EBSI Conformance</i>	53
5.2.3 <i>Ethics Guidelines for Trustworthy AI</i>	54
5.2.4 <i>EU Artificial Intelligence Act</i>	55
6 Conclusions.....	57
7 References	58
APPENDIX A – Basics of blockchain by example	60
APPENDIX B – Data Governance Framework Questionnaires	63
APPENDIX C – ALTAI-driven Questionnaire	66

List of figures

Figure 1: A centralised versus a decentralised infrastructure	11
Figure 2: AI4Gov Reference Architecture	26
Figure 3: Business layer of the decentralised architecture	29
Figure 4: Application layer of the decentralised architecture	30
Figure 5: HyperLedger Fabric example configuration. Green, purple and grey colours correspond to org0, org1 and org2, respectively. Org0 is the orderer. A1 is an application that invokes P1's endpoint, and A2 invokes P2's endpoint, which should run the same chaincode after endorsement. CC1 (the channel configuration), L1 (the ledger) and S5 (the chaincode) are blue; this denotes that they do not correspond to a single organisation but are common to all.....	35
Figure 6: HyperLedger Fabric two-channel example configuration.	38
Figure 7: Deployment of the decentralised test infrastructure.....	40
Figure 8: Inspecting the transactions and blocks via the HyperLedger Explorer	41
Figure 9: Sample file containing the invocation of custom rule	42
Figure 10: Verifiable Credentials in EBSI (source: https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pagelid=555222155)	46
Figure 11: dApps under the OpenDSU Framework	47
Figure 12: AI Act defined levels of risk	56
Figure 13: Evaluation of hashes.....	60
Figure 14: Inclusion of a difficulty problem for PoW.....	61
Figure 15: Mining a block	61
Figure 16: Example of blocks formed into a blockchain	61
Figure 17: Distributed ledger.....	62
Figure 18: How a change invalidates the blockchain.....	62

List of Tables

Table 1: Categorisation of Blockchains based on read/write permissions	14
Table 2: Types of data and end users per pilot case.....	22
Table 3: Decentralised Data Governance policies in AI4Gov.....	33
Table 4: Summary of symbols appearing the the HyperLedger example network architecture depicted in Figure 5	36
Table 5: Correspondence of AI4Gov user roles to HLF identity types	45
Table 6: DFG Questionnaire	65

Abbreviations

Abbreviation	Description
ALTAI	Assessment List for Trustworthy Artificial Intelligence
CC	Channel Configuration
DAO	Decentralized Autonomous Organization
dApp	decentralized (decentralised) Applications
DSU	Data Sharing Unit
EBSI	European Blockchain Services Infrastructure
eIDAS	electronic Identification and Trust Services
ESSIF	European Self Sovereign Identity Framework
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
HLF	HyperLedger Fabric
IPFS	InterPlanetary File System
NFT	Non-Fungible Token
OpenDSU	Open Data Sharing Unit

PoS	Proof of Stake
PoW	Proof of Work
SotA	State of the Art
SSApp	Self-Sovereign Application
SSI	Self-Sovereign Identity
WP	Work Package

Abstract

The present document presents the first iteration of the Decentralised Data Governance Model for AI4Gov and describes the mechanisms for achieving and implementing the required provenance and reliability features of AI4Gov while also respecting the privacy guidelines set by GDPR. After reviewing the State of the Art of blockchain technologies, the application of blockchain technology for smart contracts and data decentralisation is described along with how permissioned blockchain technologies can be used to implement any custom set of rules for accessing the data, defining smart contract logic, and endorsing changes in the blockchain state. Though at this early stage of the project, the full set of policies cannot be defined, the document describes the mechanism and the technology enablers based on the HyperLedger Fabric framework that will allow the realisation of the Model in the future in a flexible manner that will allow customisation of the Governance model and adaptation of it to future needs. Furthermore, the OpenDSU framework for realising applications respecting the tenets of Self-Sovereign Identity is briefly described, as well as how this framework is going to be leveraged for the implementation of a wallet application. Lastly, the strategy for achieving conformance with EBSI is laid out by pointing out the steps needed to fulfil this goal.

1 Introduction

1.1 Purpose and scope

The present document offers the 1st iteration of the decentralised data governance framework of AI4Gov. Its major constituents are the first set of specifications, which, based on the user stories, offer a set of guidelines for decentralised data governance and the first prototype architecture and deployment, which implements and demonstrates a set of business scenarios that conform to the general guidelines of the framework.

More specifically, a review of the existing State of the Art regarding blockchain governance models is performed and based on this, a set of policy and technology enablers are identified; these enablers are meant to facilitate a compliant decentralised infrastructure for AI4Gov that will allow both policymakers and organisations to make use of the benefits of decentralised technology in a manner that is guaranteed to offer safeguards for sensitive and secret data protection, while, at the same time, being fully GDPR compliant. The framework that governs all data-related processes is fully described for all data being handled in AI4Gov, both the ones accessed via decentralised mechanisms and the ones that are not.

At the architecture and technical level, the first version of the infrastructure using blockchain is presented. A review of blockchain technologies is given, together with an explanation of the specific technology, based on HyperLedger Fabric, that is to be followed by AI4Gov. The prototypical deployment based on this architecture is also presented in the current report.

Lastly, an analysis of the Decentralized Application (dApp) framework that is being developed for AI4Gov is given. The framework will be based on the Wallet and a dApp marketplace that will be implemented using the OpenDSU framework. The way that the AI4Gov Wallet is going to achieve compliance with the European Blockchain Services Infrastructure (EBSI) will also be documented; EBSI compliance will allow for the possibility of integration with the EBSI, which is the European blockchain infrastructure proposed by the EU.

1.2 Document structure

The present document is structured as follows:

- Section 1 contains the present introduction
- Section 2 gives a State-of-the-art (SotA) analysis of blockchain technologies, the storage mechanisms under the decentralised regime and the ways that a blockchain can be governed.
- Section 3 applies the disciplines analysed in Section 2 and applies them in AI4Gov; it defines the main mechanisms by which data policies will be defined and endorsed in AI4Gov.
- Section 4 describes the technology enablers and the way that these will be leveraged to achieve the policy mechanisms described in Section 3

- Section 5 describes the Data Governance Framework of AI4Gov, i.e., the framework that entails the processes and definition that govern all data handled in the project. It also lays out the steps needed to achieve EBSI conformance, and it also presents a short guideline on how AI4Gov will ensure compliance with the “right to be forgotten” right of GDPR; this needs some special consideration to ensure compatibility with the immutable nature of the decentralised technologies used.
- Section 6 gives the conclusions of the present work.

1.3 Previous iterations

This is the first iteration of the Decentralized Data Governance Model, with the second iteration being reported in *D3.2 Decentralized Data Governance Model V2*.

2 State of the Art

2.1 Blockchain technology

Blockchain-based data structures, in the sense of an immutable data structure that can store new information only in an append-like manner, were known in computer science long before the advent of Bitcoin and cryptocurrencies. Haber and Stornetta, for example, have proposed a blockchain-like structure for time-stamping a digital document since 1991 (Haber & Scott Stornetta, 1991). However, it was with the seminal white paper of Satoshi Nakamoto (Nakamoto, 2017) which established the possibility of creating a new cryptocurrency, named Bitcoin, that the blockchain technology became widely known and used by the general public. The main idea behind Nakamoto's¹ proposal is to use a shared ledger to record transactions between peers. While, by itself, the idea of a ledger is not new (indeed, banks have been keeping ledgers to record stored assets and loans since at least the 13th century), the revolutionary idea was that this ledger is not now kept and administrated within a single entity, but rather is shared between participants of the networks. Transactions in this setting are signed and broadcast through the network and, by a consensus mechanism that involves all partners, become part of the blockchain. The history of transactions agreed upon by the network is all that is needed to track asset ownership.

This distinction can be seen graphically in Figure 1. The centralised architecture serves nodes via a set of services and data that are accessed through a network. In the decentralised approach, each node holds a local copy of all data, and they interact via the network as equal peers.

¹ It should be noted that it is not yet clear if Nakamoto is a real person, or a pseudonym used by some other individual(s)

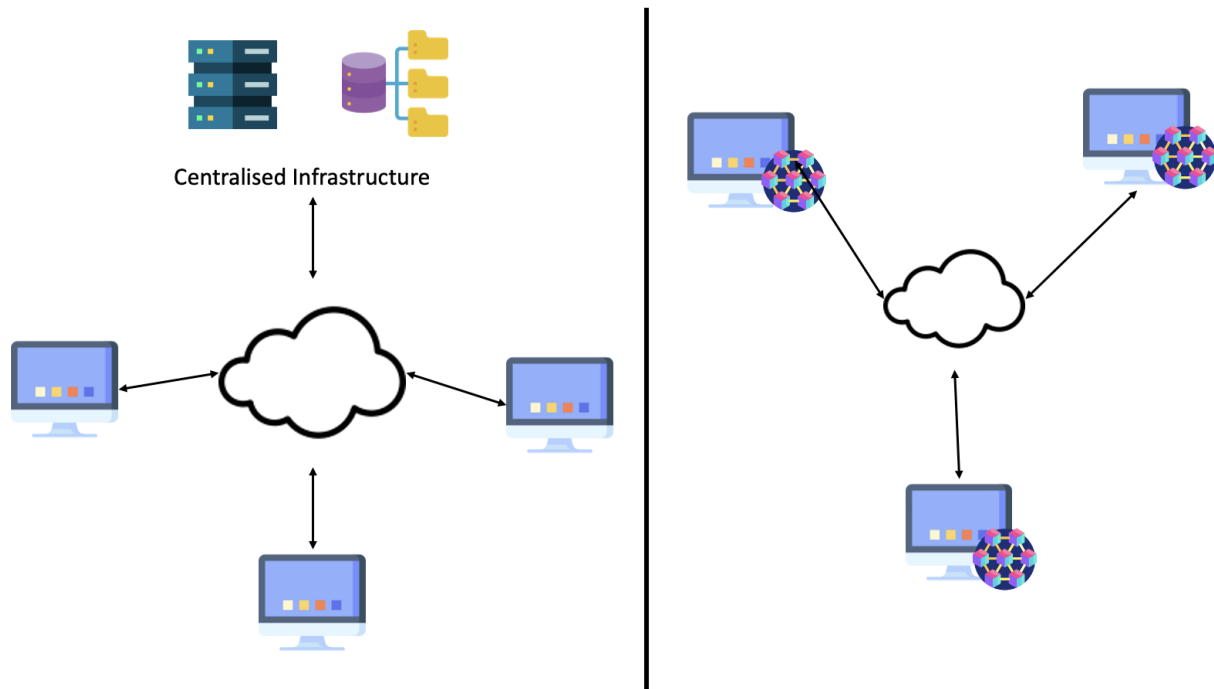


Figure 1: A centralised versus a decentralised infrastructure.

The details on how consensus is achieved, how cryptocurrency is created or charged for transactions, etc, are defined by the specific technology used. In Bitcoin, for example, special operated nodes, called miners, collect transactions and combine them to create hashed blocks. However, the Bitcoin protocol dictates that in order for a block to be acceptable, its hash should be below a certain value that is determined by the current difficulty level (Bitcoin has a complex way of setting the difficulty that takes into the total hash created by all miners at an instance). To do so, a miner should include a seed in their computation and make consequent tries until the target value is reached. As there is no known algorithm for reversing a hash, the only way to do so is by brute force (Menezes et al., 2018). This Proof Of Work (PoW) mechanism achieves two things:

- It is extremely difficult to commit fraudulent transactions. As the hash rate problem is a known one-way problem (i.e., difficult to solve, but easy to verify a solution), fraudulent transactions containing double spending or involuntary transfer of funds will be discarded by the network, as the amount of resources needed to create a block with the hashes set by the difficulty level demands an extreme amount of computational resources that, for isolated attackers, are dwarfed by the total computational power of the network (if an attacker however, achieves to produce more than 50% of the total computational power, then she/he/they can take over the blockchain. This is known as the 51% attack).
- By rewarding an amount of Bitcoin to the successful miners, this mechanism acts as the money-issuing mechanism of the currency. In parallel, it gives incentives to miners to invest in equipment and electricity costs to mine new blocks.

For a visual example of how blockchain works under the PoW scheme, please refer to APPENDIX A – Basics of blockchain by example, which offers a review of how blocks are mined and how the consistency of the blockchain is agreed by the peers via the PoW mechanism.

Although PoW is a reliable and proven, both theoretically and in practice, method for awarding miners and ensuring the validity of the recorded transactions, it has become under scrutiny due to its environmental footprint. Indeed, as Bitcoin and cryptocurrencies have become popular, more and more miners are joining cryptocurrency networks; that means increased available hash rate and thus increased auto-adjusted difficulty of the mining problems to ensure steady block production. This computation comes with an ever-increasing electricity cost, thereby greatly increasing CO₂ emissions (Mora et al., for example, estimated a potential push of 2 Celsius degrees just from Bitcoin emissions (Mora et al., n.d.)). The Proof Of Stake (PoS) has been suggested and implemented by many blockchains as an alternative consensus mechanism that overcomes this issue. In a PoS mechanism, a node is chosen via a selection mechanism to suggest new blocks. This node “stakes” a certain amount of cryptocurrency as the new block is validated by the other nodes. If it is approved, the block is added, and the node is rewarded with an amount of the blockchain cryptocurrency. If the other nodes detect an invalid block, they disregard the block, and the staked amount is removed (“slashed”) from the node’s account. The incentive for nodes to participate is, exactly as in the case of PoW, the possibility of a reward, while foul play is averted by penalising the fraudulent node. Although PoS is much more environment friendly, it has been criticised for removing certain facets of the decentralised nature of blockchain; examples of criticism are that only one node at a time suggests blocks and that only “rich” participants own a significant amount of the blockchain's cryptocurrency can suggest blocks and get rewards.

As we have seen, the first, historically and still the major one in terms of global transaction volume, use case of blockchain technology is the creation and circulation of cryptocurrency. Due to the anonymous nature of transactions and the highly speculative trading of cryptocurrency, cryptocurrencies have acquired bad fame, with negative opinions ranging from considering it to be a dangerous and unstable asset to being an “out-of-the-book” scam. Recent examples, such as the Tera-Luna crash (Briola et al., 2022), the collapse of major exchanges^{2 3 4} and the collapse of Non-Fungible Tokens (NFTs, see Section 2.3) prices, seem to validate these opinions. However, while there is a strong possibility that cryptocurrencies prove in the end to be a failure or scam, the technological framework of the blockchain has proven to be applicable in other e-gov and business use cases. Certification of credentials under the Verifiable Credentials scheme (Sedlmeir et al., 2021) or verification of stages of the supply chain (Gurtu & Johny, 2019) (see also IBM’s Food Trust solution⁵) are typical use cases where blockchain can be a strong enabler. We will investigate such use cases, especially in the context of AI4Gov, in the present report.

² <https://www.investopedia.com/what-went-wrong-with-ftx-6828447>

³ <https://www.coindesk.com/markets/2022/07/15/the-fall-of-celsius-network-a-timeline-of-the-crypto-lenders-descent-into-insolvency/>

⁴ <https://www.cnn.com/2022/07/11/how-the-fall-of-three-arrows-or-3ac-dragged-down-crypto-investors.html>

⁵ <https://www.ibm.com/products/supply-chain-intelligence-suite/food-trust>

2.1.1 Public Blockchains

Although not explicitly stated, the discussion of the previous section (Section 2.1) involved what is called “public blockchains”. As its name suggests, a public blockchain is a blockchain in which anyone can join and participate in transactions, mining or other operations the blockchain offers. The only thing required is a key pair (a public and private key); this key pair is also called a wallet and can be used by the user to sign transactions and receive funds (aka cryptocurrency). Apart from the wallet, which the user can freely create, public blockchains do not contain any login mechanism, nor do they have any user registry that stores the user base of the blockchain. Users are identified only by their address (which is commonly derived from their public key) and are therefore anonymous. Any mechanism of retrieving any kind of information from the blockchain involves the inspection of the ledger history and will reveal only information about wallets, which do not necessarily correspond in a one-to-one manner to users (a user may have multiple wallets in their possession, with no way of determining this relationship).

This public nature of blockchains is one of the main drivers of implementing the consensus mechanisms described above (Section 2.1). They provide the means by which currency is circulated in the blockchain (via rewards to miners/stakers or via transaction fees) and the mechanism by which all nodes agree on the status of the blockchain. The Bitcoin and the Ethereum Network are two examples of public blockchains.

2.1.2 Permissioned Blockchains

While public blockchains may be ideal for cryptocurrency issuing and trading, it is easy to see that adoption of them for use cases involving the public sector or business can introduce many difficulties. First of all, the anonymity of users is something not commonly suited for business applications, where the identification of each user is almost universally needed, commonly together with a role-based mechanism. Furthermore, consensus cannot be reached by the same mechanisms as that of the public blockchains, by demanding that end users mine blocks or stake cryptocurrencies. Even if these roles are limited to special infrastructure roles, the fact that the blockchain has an inherent cryptocurrency needed for transactions and consensus is not something that can be easily adapted to each business case.

To this end, a special category of blockchain technologies has been implemented, the so-called permissioned blockchains. In a permissioned blockchain, each user needs to be registered, usually by an administrator(s) or at least by a registering policy defined by the existing nodes. Registration, authentication and authorisation are typically performed by identifying the users with their credentials, typically issued by Certificate Authority (CA). The users that are enrolled have access to the blockchain, but additional conditions may limit visibility aspects; for example, the infrastructure may have channels which possess a separate ledger, and the users can use only the ledger(s) in which they are enrolled.

Furthermore, the consensus regarding mined blocks and order of transactions is performed differently in permissioned blockchains, with the exact mechanism being dependent on the specific technology used. A typical mechanism is the usage of a special service called the Orderer. The Orderer collects transaction requests from nodes, validates them and decides upon the order

in which these are to be inserted into the blockchain. To the question of what incentives nodes have to run the Orderer service, the answer is that this depends on the value added by the blockchain solution. A consortium that consists of banks, for example, that benefits from a permissioned blockchain, can share the cost by each of them running orderer nodes in parallel or by rotation or even by renting this functionality to a trusted third party. A public service offered by governments, on the other hand, has the incentive of creating or adding value for citizens and, thus, the incentive of running an orderer in the government’s infrastructure.

The HyperLedger Fabric (HLF)⁶ developed by the Linux Foundation⁷, which is also the technology that is to be adopted in AI4Gov, is an example of a permissioned blockchain.

2.1.3 Read-write access

This section takes a little digression to denote that, although the above categorisation into public and permissioned blockchain is sufficient for the purposes of the project and of the present document, we should state for completeness that there can be more granularity in the access levels of a blockchain. In a blockchain, there are two ways in which users interact with the blockchain:

- Read access, which allows users to view the contents of the blockchain.
- Write access, which allows users to perform transactions on the blockchain. By transactions, we mean anything that alters the blockchain, from simple asset transfer to smart contract (see Section 2.2) deployment and state-altering smart contract invocation.

In this context, a public blockchain is a blockchain that can be read by anyone, while a private is a blockchain with restricted read access. A permissioned blockchain is a blockchain in which only registered users/wallets can alter the state of the blockchain, while a permissionless is a blockchain in which anyone can perform state-altering transactions (see Table 1).

Table 1: Categorisation of Blockchains based on read/write permissions

	Permissionless	Permissioned
Public	Anyone can view the chain and anyone has the same access rights	Anyone can view the blockchain, but not all wallets have the same rights
Private	Anyone can create data and/or contracts, but viewership is limited	Viewing the blockchain and write-access rights only to registered users

⁶ <https://www.hyperledger.org/projects/fabric>

⁷ <https://www.linuxfoundation.org/>

The Ethereum and Bitcoin blockchains mentioned in Section 2.1.1 are public permissionless blockchains. HyperLedger Fabric, on the other hand, mentioned in Section 2.1.2, is a private permissioned blockchain. A public permissioned blockchain is one that allows visibility of data, but enrolling into the blockchain requires some form of registration that must be approved by a party determined by the governance model of the blockchain. Sovrin network (Windley, 2021), which aims at establishing a service for self-sovereign identity, is one example of a public permissioned blockchain. The last case seems self-contradictory, as free write access seems to imply free read access. However, there are use cases and corresponding blockchains that require limited read access with unlimited write access. This case happens when there is the need to allow anyone to enter data or deploy smart contracts on the blockchain, but we give this party the right to limit the viewership of the data they entered to a list of users/wallets that the party requires. Holochain (Kiyak et al., 2022) is an example of a private permissioned blockchain.

As already hinted, this level of granularity in categorisation will not be needed for the purposes of the present report; the term *public blockchain* will be used instead of public permissionless blockchain, and the term *permissioned blockchain* will be used instead of private permissioned blockchain unless otherwise needed.

2.2 Smart Contracts

Although performing direct transactions of the form *party A gives X amount of cryptocurrency to party B* is an obvious use case of cryptocurrency and blockchain-based transaction recording, it is often the case in finance that transactions are not performed unconditionally. There may be the need for some backward checks, the need for a third party (e.g., an arbiter) to arbitrate or broker the transaction, etc. Blockchains have the capability of performing such tests by running appropriately written code. The main idea is the following:

- A wallet writes and deploys the bytecode containing the business logic.
- The bytecode is inserted as data directly into a block and, once mined, becomes part of the blockchain.
- When a party wishes to invoke the business logic, it accesses it by its address and required parameters. The code is executed by performing any actions and transactions implemented in its bytecode.

It can be understood that such a code can have certain limitations. First of all, as it will be stored in the blockchain, it should be compiled in a format that can be understood by the nodes of the blockchain so that they can run and validate the code. Secondly and most importantly, this code should be deterministic, especially if it alters the state of the blockchain. Otherwise, the nodes that tried to produce blocks would not agree on the output of the code; if the code involves a transaction, there would then be no consensus regarding which run is the correct one. As non-determinism is not only introduced by random generators but also can be introduced by external services, blockchain code cannot use directly external sources of information (e.g., it cannot run an external service to obtain market prices directly and compute charges based on them, as these can change in each run by each one of the blocks). There are ways to overcome some of these

limitations to an extent (for example, by invoking the Chainlink (Breidenbach et al., 2021) oracle network), but this limitation is something that the code developers should be aware of.

Although Bitcoin had its own version of scripting language (called Script), the true revolution in blockchain coding came with the advent of the Ethereum network and its Ethereum Virtual Machine (EVM) (V.M. 2014). The main game changer was the introduction of a Turing complete language for the development of blockchain code, also called a smart contract. While Script only allows for a limited functionality (e.g., loops are not possible in Script), the EVM can run code that is Turing complete. Apart from the new capabilities that are introduced in the area of what is called Decentralized Finance (DeFi), the possibility to create smart contracts has far-reaching implications for other areas of business.

Using smart contracts, we can, in theory, implement code that is shareable, visible and transparent to all parties. By implementing an agreed set of conditions by which the code is accepted (endorsed) and by which its versioning and maintenance are governed, parties can perform transactions directly without the needs of physical third parties (brokers or arbiters) with undisputed outcomes. In fact, the roles originally fulfilled by brokers and/or arbiters can be performed by the code of the smart contract, which conforms to the business logic that all participants agree; this is the rationale behind the *Code is Law* (Hassan & De Filippi, 2017) mantra, often chanted by the crypto community. Even beyond the scope of general finance, smart contracts can be used to verify one's identity and conditionally retrieve shared attributes that can be validated by all parties, e.g., citizens can prove their nationality to a requesting authority without the need to provide unnecessary and redundant documentation.

2.3 Decentralised Storage

One aspect that is crucial for many business scenarios and that also introduces concerns regarding data privacy is that of data storage. Blockchain is, at the very basic level, a data structure and, as such, can, in theory, hold any kind of data, including large report files or even multimedia. However, the limitations of following this direct approach immediately become apparent:

- Storage in the blockchain means that the data becomes available to all parties, something that seriously challenges data privacy and GDPR compliance. Even if they are stored encrypted, there is still the possibility of potential decryption of the data, either because the password was compromised by whatever means or because research has found flaws in the mathematical algorithm used for encryption. In any case, the idea that data, even encrypted, remain in the blockchain forever seems to contradict the basic *right to be forgotten* tenet of GDPR.
- Using the blockchain as a database is extremely inefficient. Copies of all files need to be copied and shared through all peers, and searching or retrieving the file may require traversing a long series of blocks, especially if the file cannot be stored in a single block due to block size limitations.

The above difficulties are inherent to many blockchain technologies; after all, they were first designed to offer the possibility of recording anonymous cryptocurrency transactions, which is a use case with no need to preserve sensitive documents and data.

However, as the possibility of using blockchain in a series of non-cryptocurrency-related business cases was identified, new methods have been devised to cope with these limitations. Even within the crypto world, a new use case, the Non-Fungible Token (NFT) (Wilson et al., 2021), has been implemented and made extremely popular in the crypto community. Originally created in 2014 (Cascone, 2021), an NFT is basically a unique identifier that is mined into a blockchain and points to a file containing a piece of art, such as a painting, a video, etc. The unique identifier proves ownership of the corresponding piece of art and can be traded between parties directly or via smart contracts; any such trade transfers the ownership of the NFT. NFTs themselves are distinguishable from each other, as each one, even if they contain the same picture, is unique; hence the “non-fungible” terminology in their naming.

NFTs have become extremely popular, with millions of dollars being traded for their respective market. They have drawn a lot of criticism, ranging from mere dismissal of the idea to them being a highly speculative or even fraudulent market. Although the recent price crash of NFTs⁸ seems to confirm the notion that NFTs may have been a bubble or even an outright scam, the fact remains that they demonstrated the use case of using the blockchain to refer to off-chain data in a transparent way. While many NFT holders may feel scammed, they at least can prove, at any time, ownership of their now worthless asset⁹; while not very useful to them now, this capability points out at possibilities for custom business case scenarios which involve decentralised data handling and transfer.

In a business scenario, file sharing and proof of ownership are critical enablers as they allow the exchange and usage of information and reports. The main lesson is that we do not need to store files in the blockchain but pointers (or anchors) that point to the location of the file. As these pointers can be mined via smart contracts, they can be defined to prove and validate ownership of the file. The file itself can be stored in an encrypted form that only the owner of the file can decrypt via her/his keys that identify her/him. As an example, suppose that a company generates reports regarding sensor data that measure the acidity of their product at various stages in the supply chain. These reports can be anchored to the blockchain as they are generated. When a party requests acidity numbers for a special batch and stage, the report is retrieved via the identifier and displayed to the requesting party; a hash of the file can also be stored with the anchor to ensure that the off-chain data have not been modified.

There are various frameworks and libraries for storing and retrieving data off-chain, each with its own set of features concerning access ways and what kind of information is retrieved (e.g., ownership, files, zero-knowledge proofs, etc.). One such framework is the OpenDSU framework, which has already been successfully used for developing self-sovereign systems for data

⁸<https://markets.businessinsider.com/news/currencies/nft-market-crypto-digital-assets-investors-messari-mainnet-currency-tokens-2023-9>

⁹ To be precise, this is not entirely true. There is no inherent restriction in the NFT mechanism that hashes the asset and enforces the hashes to be the same. For example, Moxie Marlinspike demonstrated how one could create an NFT that can change depending on the platform displayed, and that turns into a poop emoji when someone buys it (<https://moxie.org/2022/01/07/web3-first-impressions.html>), further justifying the infamy of NFTs. Again, the fact remains that NFTs displayed a use case for storing files off-chain. In a serious business setting, the anchoring can be made secure by enforcing hashes, as we will see in the present report.

governance in pharmaceutical-related use cases (Ursache et al., 2022). OpenDSU is a framework for implementing decentralised applications (dApps) that allows full control of the end user's identity and content (the term SSApps (Self Sovereign Application) is used and proposed in the OpenDSU documentation exactly to stress the self-sovereign nature of the OpenDSU-based dApps). OpenDSU is blockchain agnostic and allows the deconstruction not only of the underlying data that a dApp contains but also of its entire execution environment.

The versatility of OpenDSU makes it one of the key enabler technologies that are going to be used in the AI4Gov for the implementation of the relevant dApps and wallet.

2.4 Blockchain Governance

The governance model of an enterprise solution is a crucial aspect of its lifecycle that determines what changes are going to be performed and how they will be implemented and integrated into the existing solution. In traditional, non-blockchain-oriented solutions, governance is performed by one or a set of governing bodies and implemented by the development team, with potential overlapping members who can be part of both the development team and the governing body. In a web-bank scenario, for example, a legislative governing body may decide that all web-bank applications should have two-factor authentication. Another governing body within the bank (e.g., the executive board) may authorise the update of the application to conform with the new legislature. The enterprise architecture team is another governing body that will decide how the architecture is to be adapted/refactored and maybe provide the technical specifications. The development team, lastly, will perform the changes.

It is obvious that the above model encounters some natural obstacles when applied to a blockchain solution.

- In a decentralized setting, who is the one who decides that a change should be made and adopted?
- Provided that a change is decided by whatever means, how is it going to be deployed into the whole decentralised network?

The decentralised governance models that are designed to handle these challenges can be briefly categorised into: a) off-chain governance models and b) on-chain governance models; see also the review in (Fischer & Valiente, 2021). We will give a brief overview in the following subsections. However, it is important to note that in many blockchains, especially the public ones, there is no clear distinction between the governance bodies and the development teams. In fact, developers who write smart contracts and actively contribute to the blockchain are typically participants and even key players in any governance bodies or mechanisms that the blockchain possesses. This can seem reasonable in the public blockchains since on-chain operations rely mainly on the protocol(s) of the blockchain and how these are implemented; therefore, It makes perfect sense for the developers to be central actors of the blockchain's governance. In the permissioned case, which is typically adopted in corporal and organisation settings, we cannot expect the developers to have special privileges, and governance will generally depend on the blockchain's consortium agreement. Even if aspects of policy regulations are deferred to on-chain

procedures, decisions to endorse or reject proposed changes will depend on the internal governance of each peer (organisation).

2.4.1 Off-chain governance

The off-chain governance model is typically used by PoW blockchains. As its name implies, decisions take place by processes not involving the blockchain. Depending on the blockchain, the model can range from a total democracy, in which all parties vote on the decision, to a “benevolent dictator” model, where one person has full decisive power. The governance body can use a variety of tools, ranging from conferences, forums, physical meetings, etc., to reach a decision. If a decision requires changes to the blockchain state (e.g., reset to a previous state to cancel some transactions that exploited previous erroneous code) or to the blockchain code (e.g., an increase of block size), this is implemented by the developers.

With the consensus reached and the changes developed, the blockchain then splits into what is called a *blockchain fork*. What this means is that after a certain block, the new blockchain code only accepts blocks that are mined using the new rules determined by the update. Any blocks that are added using the code implementing the old rules are discarded by the nodes running the code implementing the new rules.

It is to be noted that, in theory, there is no real authority that can impose nodes to adopt changes. The majority of peers and developers adopt the new rules either because they were part of the majority voting or accepted the vote outcome (in democratic models) or because they respect the authority of the dictator. It can, and in fact did, happen that certain nodes disagree with the new rules and continue to use the old rules by continuing to add blocks to their own fork using the old rules. Bitcoin Cash is such an example, where some nodes disagreed with a proposed upgrade (the so-called Segwit upgrade) that was meant to tackle several issues of Bitcoin, including measures to limit the impact of the small block size of BitCoin, and created their own fork implementing their own changes, by directly changing the block size to a larger size. Another example of a benevolent dictator model is the hard fork proposed in the Ethereum blockchain by its creator (and dictator) Vitalik Buterin in 2016. The hard fork was meant to remedy a situation that was caused by the vulnerability of a widely used smart contract, which caused the theft of Ethereum cryptocurrency equivalent to ~70M in 2016 prices¹⁰. The new fork was not accepted by some purists who insisted that the “Code is Law” mantra is followed to the letter and that, bug or no bug, the state of the blockchain should not be retroactively altered. The ones who did not accept Vitalik’s fork continued to use the previous version of the blockchain; the respective cryptocurrency is now known as Ethereum Classic.

It is to be noted that in all cases, the fact that the decided forks have been accepted as the “correct” ones and have kept the original name of the respective cryptocurrency and network happened solely because, in the end, the majority of nodes accepted the changes and established its dominance by producing a much greater hash rate, or total staked coins, than that of the

¹⁰ <https://www.bitstamp.net/learn/crypto-101/ethereum-dao-hack/>

“rebels”. From a technical perspective, if one were to inspect the various forks of the blockchain, she/he would find nothing that denotes that a certain fork is the “official” Bitcoin while another is the Bitcoin Cash. As is the case in many areas ranging from political party splintering to religious schisms, the majority usually keeps the “brand name” and establishes a stronger claim to continuation.

2.4.2 On-chain governance

An on-chain governance model is typically used by PoS blockchains and is one that occurs solely within the blockchain. The enabler for such a governance model is the smart contract technology. Special smart contracts implement the functionality of what is called a Decentralized Autonomous Organization (DAO)¹¹. A DAO implements, via the set of rules established by the smart contract, the governance model of the blockchain. For example, a DAO may provide a mechanism for voting on suggested upgrades, during which all or a subset of nodes can vote, either sharing an equal voting power (one vote per user) or by having a voting power analogous to their total staked cryptocurrency, or by any other mechanism dictated by the DAO. Accepted proposals are coded into smart contracts and are then executed in the blockchain, possibly with a further round of votes to ensure validation (i.e., there is an approval on the “how” the accepted changes were implemented).

2.4.3 Governance in HyperLedger Fabric

In permissioned and/or private blockchains, the governance models described above may not directly apply; however, they still perform the keyways that a blockchain is governed and therefore by combining certain aspects, they can create custom governance models for permissioned private blockchains.

For the case of HyperLedger Fabric specifically, which will be the blockchain that is going to be used in AI4Gov there can be different governance models ruling the blockchain’s code and the smart contracts’ (called chaincode in the HyperLedger Fabric ecosystem) lifecycle governance.

First of all, regarding the source code of HyperLedger Fabric, this follows an open governance model, with the rules specified by the Linux Foundation, which is the foundation that developed the HyperLedger ecosystem. Governance of source code is thus 100% off-chain. However, HLF offers the capability of defining special rules for accepting new chaincode or upgrading existing ones, by setting the appropriate endorsement policies; these policies implement a kind of on-chain governance model for chaincode consensus.

One last item that we need to consider as a digression is the distinction between what we call in the present report a blockchain network and a blockchain Network (with capital N). A blockchain network is any network that is based on some existing technology. For example, a set of business partners or friends may get the code of the Ethereum blockchain and create their own small

¹¹ DAOs are also used in chains governed off-chain to implement custom agreements (e.g., loan system with collateral rules). In fact, the original hack in the Ethereum that caused its hard fork was in one of its DAOs. While important for chains governed off-chain, DAOs are indispensable for on-chain governance.

Ethereum network based on the Ethereum protocol. The Ethereum Network, on the other is the global network that was created by Vitalik Buterin; when someone hears about the price of “Ether” it refers to the cryptocurrency created on this Network. The distinction is somewhat similar to the terminology used for internet (any network that bridges local networks of various technologies via a network layer) and Internet (the global internet that everyone is familiar and uses). In the HyperLedger setting however, this distinction is not relevant. There is no global HyperLedger Fabric network, and networks developed using HyperLedger Fabric (or any other solution from HyperLedger, such as Besu or Aries), typically have their own name that is associated with the business case they implement (e.g., the European Blockchain Services or EBSI, IBM Supply Chain Intelligence Suite, etc.).

3 Decentralisation in AI4Gov

After having investigated the main use cases and governance models of blockchain technology, the current section will provide the general guidelines for how blockchain will be adopted in AI4Gov to fully utilise the technology's added value and potential. The analysis will focus on the following issues:

- Identification of the data that is going to be used by AI4Gov's pilot cases in terms of type, volume and privacy of data. Though this identification will firstly be derived for the three pilot cases, there will be an effort to identify characteristics of data of potential future adopters of the platform.
- Identification of main data handling and normalisation scenarios, also based on the current architecture of AI4Gov (documented in D2.3).
- Defining the appropriate decentralised data governance model that defines how data are stored, accessed, and used.
- Definition of the general requirements that decentralised code (chaincode) should fulfil in order to be able to execute the scenarios defined by the needs of data normalisation and use case scenarios.
- The blockchain network topology and configuration that achieves the objectives defined by the decentralised data governance model and the chaincode requirements.
- The on-chain governance model that is applicable to the AI4Gov use cases.

3.1 Data information in AI4Gov

Based on the specification scenarios conducted in D6.1, a set of data sets being handled and the future user stories that the pilots wish to implement have been documented. For the full analysis, the reader may refer to Section 4.2 of D6.1; here, we will aggregate information gathered by the analysis that concerns the data used and how they are expected to be used for the realisation of future scenarios. This overall view is depicted in Table 2, where data and end-user information are grouped per pilot.

Table 2: Types of data and end users per pilot case

Use Case	Pilot	Data	Type	Users
Water Management drinking water	DPB	-Sewage Treatment data	Static/Streaming	-Workers at the municipal consortium for water management

Water Management – sewage water		<ul style="list-style-type: none"> -Water cycling billing data -Streaming sensor data 		-Local administration
IRCAI global 100 projects	JSI	<ul style="list-style-type: none"> -IRCAI data of projects submitted (textual description, URLs) -Event Registry data (news and event items) -OECD AI policy initiatives 	Static	<ul style="list-style-type: none"> -Teams in private or public Institutions/Organizations that are submitting projects to the IRCAI Global Top 100 program. -Government -Corporate -Researchers
SDG Observatory				
OECD policy document analysis				
Parking tickets monitoring	VVV	<ul style="list-style-type: none"> -Census data -Household water data -Tourist data (arrivals, overnight stay, cruise data) -Airport traffic data -Municipality events attendance data 	Static	Policy makers
Waste management – Pay as you Throw				

While most data refer to reports of several kinds and aggregation of information, we can already see that the characteristics can impose certain limitations on the underlying data governance model. On the one hand, there are report data (mostly in the case of JSI scenarios) that are generally public and do not contain sensitive information. Sharing these files via a central storage mechanism or by anchoring them into a blockchain in an unencrypted format can be done without special considerations. On the other hand, census data, that VVV has access to, certainly contains sensitive data, either in the form of personal information or by exposing information that an attacker can utilise to ascertain certain attributes of the data subjects. These, of course, cannot be stored directly in a central database; an anonymisation process must take place before this is

done. Even anchoring these data in the blockchain in encrypted format may cause some GDPR compliance issues, as was explained in Section 2.3 (see also Section 5.2.1 of the present report).

In terms of efficiency, the volume of data used can range from small reports of the size of the order of several megabytes to collections of stream data that can take up to several gigabytes. Stream data will not be handled by the blockchain, as there is no clear use case for this, and, in general, it is extremely inefficient for handling such data (even traditional databases may face performance issues; typically, special time series databases are used for this purpose, such as InfluxDB¹²). For the static case, for efficient storage and processing of large files, should they be incorporated into a decentralised storage solution, the special considerations put forth in Section 2.3 should be taken into account.

Currently, there is no clear picture of which of the described data should be stored in a centralised manner and which ones should be anchored to the blockchain. Data that need a mutually verified ownership or a transparent history of changes for purposes of accountability should be anchored to the blockchain. Other report data and their validity can be easily verified by the generating authority itself (e.g., by accessing its website) and may be stored centrally. In any case, however, the decentralised data governance model should take into account all cases that can be encountered by the pilots and by future adopters of the AI4Gov platform. These can be summarised in the following list:

- Allow the possibility of encrypting anchored data.
- Limit decryption of anchored data to users/wallets that can prove ownership of the anchored data or to users/wallets that the owner has explicitly given access to.
- In case of data corruption or voluntary changes of the anchored data, the anchor should be invalidated so that the requesting party never retrieves a file that has changes that occurred after the anchoring.
- Allow versioning of anchored files so that verified and agreed versions of the file can be tracked through the blockchain.

Apart from the use case analysis, Task 3.2 produced a data questionnaire that was distributed to all partners of the consortium. This questionnaire collected information about all data that each organisation handles, such as their organisation and documentation, their interoperability aspects, accessibility aspects, ethical issues etc. This information, especially the one corresponding to the pilots will be of crucial importance when the Decentralized Data Governance Model is instantiated for each use case. For example, if standards are used, the smart contracts that anchor them to the blockchain will ensure that these are respected. If interoperability aspects of the data must be retained, likewise, the smart contracts will make sure that interoperability is retained as data are transformed and/or uplifted when used in various scenarios.

The reader can refer to APPENDIX B – Data Governance Framework Questionnaires for a full listing of all questions of the questionnaire.

¹² <https://www.influxdata.com/>

3.2 Decentralised Data Storage in the AI4Gov platform

Before we describe the architecture and design of the decentralised data storage infrastructure and smart contracts, we first give a brief recap of where the blockchain infrastructure resides within the general AI4Gov framework. Figure 2 depicts the overall AI4Gov reference architecture as described in D2.3 The blockchain infrastructure enables the following:

- It provides the backbone that runs the smart contract that the various dApps that will be developed for AI4Gov will invoke. dApps in AI4Gov will be developed for business scenarios that benefit from transparency in code execution and require agreement from all partners. As an example, a policy recommendation rule engine, can run as a smart contract that is endorsed by all peers; in this manner, all parties will agree upon the conclusions of the engine and each peer can prove that they receive a specific output from the rule engine.
- For data that are going to be stored via blockchain anchoring, it provides all necessary smart contracts for performing the anchoring and also provides any access functionality required.
- For data that need to be aligned to a specific schema or transformed under certain rules, it can provide the necessary mutually endorsed smart contract that provides the required transformations via the commonly endorsed policy; the data can then be stored centrally or anchored to the blockchain.

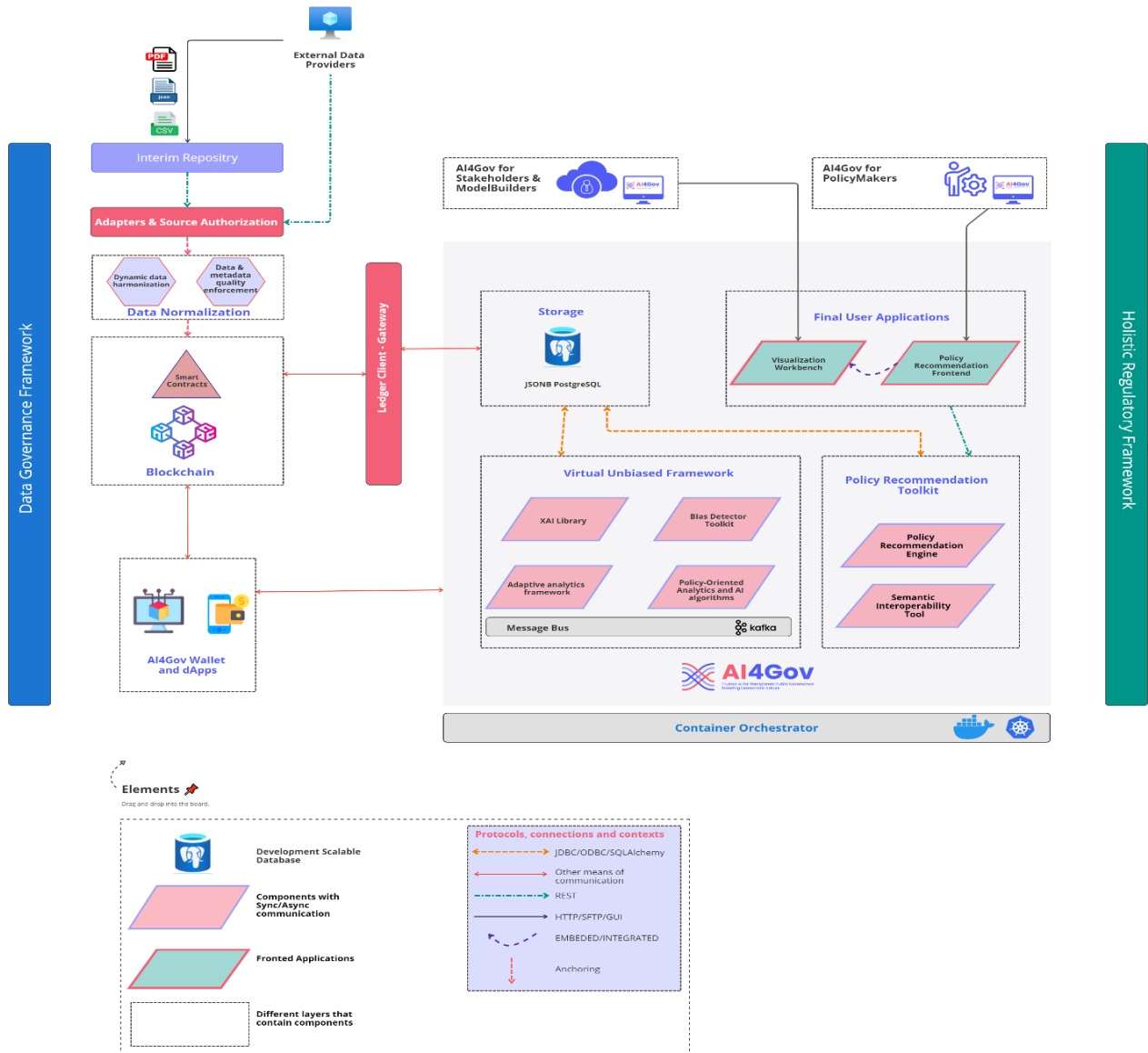


Figure 2: AI4Gov Reference Architecture

The topology of the network will be such that all pilots will have at least one node participating in the network, which, at least for the prototypical implementation, will reside on AI4Gov’s VM infrastructure. There will be at least one channel that is common to all organisations, with additional channels being added and configured as needed (see Section 4.1.1 for a brief description of the notion of channels).

One closely related feature to the ones listed above concerns the *output* of the various AI4Gov modules. The AI modules that are to be developed in WP4, for example, will give a set of tools for detecting bias in documents and will further utilise explainable AI techniques to give insight into the results of bias analysis. These reports, containing both the results of the analysis and the

explanations, will also be stored in the AI4Gov platform. While not clear at the present moment, a decentralised data storage mechanism and data validation mechanism may also be beneficial for end users by, for example, anchoring the explainable data to record the rationale for a result produced at a specific point in time. In this case, the explainability report should conform to the common standards agreed upon so that the same results will have the same semantics in all settings and can thus be applied in a straightforward and commonly agreed manner in, for example, a definition of a new policy. The main point is that whatever transaction needs to store data, it should do so via a smart contract that clearly defines and enforces the schema of the stored data. The business logic of the contract should be endorsed by at least the peers who may require the data for verification and analysis.

3.3 Architecture for Decentralised Data Governance

In this section, we will briefly describe the main architectural concepts of the AI4Gov decentralised infrastructure. Although the main interactions with other components of the AI4Gov platform are explained in the AI4Gov Reference architecture (D2.3), the present analysis will focus on the decentralised governance models. While the models refer both to data and the business logic implemented by the decentralised infrastructure (i.e., smart contracts), we will refer to both of them as the *Decentralised Data Governance Model*, for short, unless the distinction between data and code needs to be referenced explicitly.

We will make the analysis following the Archimate modelling approach¹³. At this stage, it is very early to produce a technical layer of the model; therefore, the analysis of the pilot cases will focus on the core aspects of the Business and Application layers of the Archimate model, at least at level zero¹⁴. While even these two layers will be refined at the 2nd iteration, as details involving the usage scenarios (e.g., access based on user roles, endorsement policies, etc.) will become clearer as the component design is integrated into a complete architecture design, we can still discern the main aspects of the business and application functionality. These aspects are general enough to allow for future specification and granularisation depending on the exact data and execution flows that are going to be implemented for the pilots.

3.3.1 Business layer

Starting from the business aspects of the model, the level zero business view can be seen in Figure 3. The actual users and user roles are not defined explicitly they are rather referenced by the generic “User” and “User Role” terms. The important thing to note is that the infrastructure will

¹³ <https://www.archimatetool.com/>

¹⁴ Level 0 diagrams are typically defined for Data Flow Diagrams. Here, by level 0 we denote that we will consider mainly the interactions between business and application elements, which, by themselves, will be treated as black box. V2 will define all elements, including the technical one, fully.

allow for users and role-based access to the blockchain and the corresponding services. The main roles have been identified in D2.3 and are the following:

- AI Model Builder
- Ethical Expert
- Individual/Citizen
- Policy Maker
- Admin
- External Sources (external systems)

The enrolling policies and the part that each role will play in the endorsement policy, both of smart code definition and of smart contract execution (aka on-chain governance), will be defined as the architecture is further refined and the data flow scenarios are better understood. The business layer of the next iteration will accommodate this to explicitly indicate the relation of each role with the governance models implemented. As we will also see in the Technology section (Section 4), the technology enablers that are going to be used allow for custom endorsement policies that separate between roles, as the requirements denote. These may not need to be confined to the business roles identified for AI4Gov; any business role that may be needed from the perspective of a future adopter of the AI4Gov platform can be mapped to identity types and be used for the definition and endorsement of policies.

For the solution, we define a single product, the “AI4Gov Decentralised infrastructure and contracts”, which may be further broken down into finer granularity in the future level one iteration of the architecture. The product composes various services, and it implements the “Decentralised Data Governance Model” contract, which is composed of two sub-contracts, one referring to the data policy and one referring to the smart contract code policy. The exact details of what these policies entail are purposefully not defined; they can be whatever is agreed by the governance bodies of the pilots and future adopters, which together form the AI4Gov blockchain consortium. We will explain in the application layer (Section 3.3.2) and in the technology section (Section 4) that the solution is flexible enough to allow for the definition and implementation of a variety of policies.

Further analysing the business aspects of the architecture, the main service that the “Decentralised Data Governance Model” offers is the “Provide decentralisation” service, which offers all the benefits of decentralisation for transparent data storage and code execution. This is denoted by the set of values which drive the need for the functionality (traceability, accountability, providence, transparency). As the main service involves both data alignment functionalities and smart contract functionality, this is denoted as two sub-functionalities.

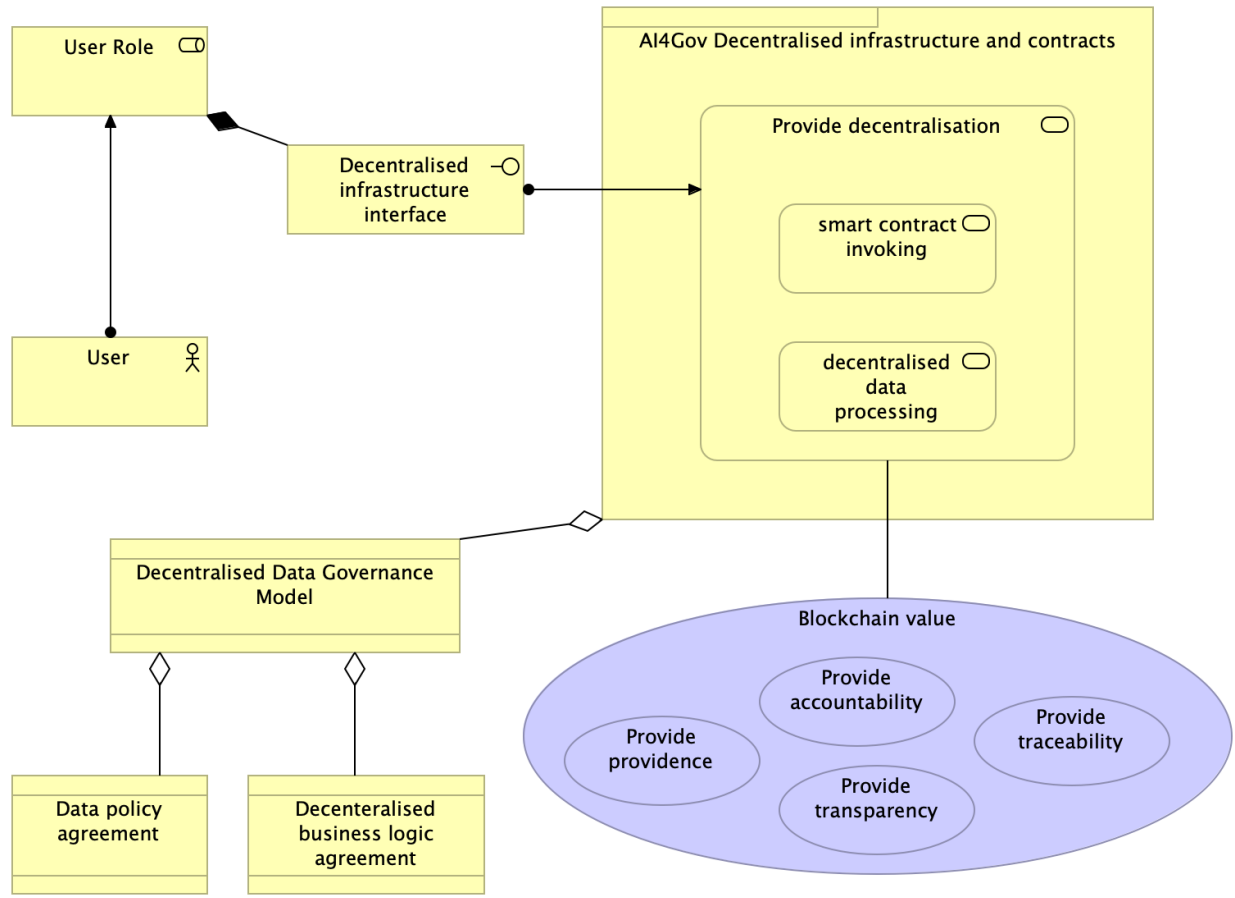


Figure 3: Business layer of the decentralised architecture

3.3.2 Application layer

Considering the application layer, the level zero diagram that depicts the main parts of the application can be seen in Figure 4. A wallet component interacts via a “Gateway” interface to access the “Data Governance Policy” service (although the term “gateway” is hinted by the mechanism provided by the HLF’s gateway mechanism¹⁵ by which users the ledger externally, from an architecture point of view this refers to any means of interfacing with a decentralised infrastructure). The service is realised by the “decentralisation” functionality, which consists of sub-functionalities having to do with chaincode invocation and execution and data alignment processes. At this point, the functionality is referenced generically, with no component being explicitly defined for implementing the required functionality. Furthermore, for the chaincode execution, the various sub-functionalities (cases where smart contract execution provides value to the execution flows of the pilots) need to be defined. The “Rule Engine” and “AI validation” functionalities have been added as examples to hint at possible future use cases and may not be the same during the 2nd iteration of the document. The main point to disseminate here is that the

¹⁵ <https://hyperledger-fabric.readthedocs.io/en/latest/gateway.html>

functionality allows for the definition of any custom business logic that needs to be executed in a decentralised manner via a smart contract.

As a last point, the “Schema definition” functionality needs to access the source data files to define schemas and enforce alignment. The “Static Documents” data object is meant to refer to any kind of report, from source files to AI bias detection results. These are static data objects. Streaming data will not be considered in the present context, as the decentralised infrastructure will be used to store only static data. There will be some process by which stream data will be used to derive some static files for analysis (e.g., reports, aggregation, etc.); however, this process will be specific to the stream and not yet specified. In Figure 4, this is denoted as a “Report Generation Service” business service that is implemented by a “Report Generation Service” application service, with the business service being connected to the streaming data and the reports by using generic association relation. In this manner we keep the definition of the process of converting stream data to static as general as possible; the breakdown of this process at the business, application and technical level will occur in D3.2 which is the 2nd version of the data framework.

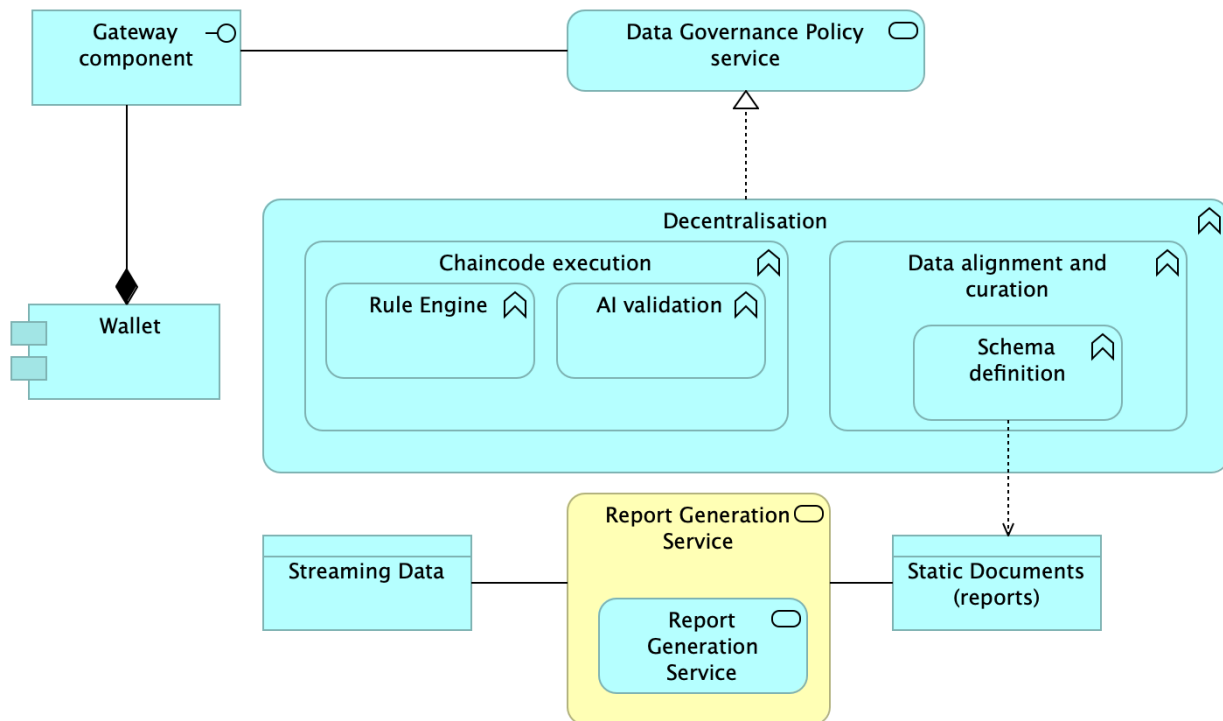


Figure 4: Application layer of the decentralised architecture

3.3.3 General requirements

Consolidating the main points of the business and application aspects, as well as the goals of the pilot use cases of AI4Gov, a set of general requirements for the Decentralised Data Governance model can be retrieved. This set is depicted in Table 1 and is deliberately defined to provide the possibility of choosing specific policies for certain aspects of the model.

Briefly, the source code of the blockchain will be governed off-chain and outside the AI4Gov project since the consortium does not take part in the development of any blockchain technology. The source code of the blockchain will be maintained by the respecting governance body that implements the blockchain solution (for the HyperLedger Fabric technology, as is going to be documented in Section 4, this body will be the Linux Foundation).

For data handling, four distinct operations are considered, namely, insertion, reading, versioning and deletion of files. Insertion and versioning will be allowed only by the data owner of the files while accessing for read purposes will be allowed both by the owner and by the users included by the user access list of the file defined by the user. Although these definitions derive from the requirements, their enforcements in the blockchain are implemented via on-chain mechanisms, and are therefore of the “on-chain” type. As we will also see in the technology section (Section 4), in the HyperLedger Fabric framework, any transaction with the blockchain, including data anchoring must be implemented via a smart contract (also called chaincode); in this sense the governance model for data handling can be seen, from a smart contract perspective, as a subset of the smart contract governance. Indeed, this fact further justifies referring to both governance models (data and code) under the same “Decentralised Data Governance Model”, as, in the end, their implementation falls back to the same mechanisms.

For smart contracts, two cases should be distinguished:

- Definition and deployment of smart contracts. This involves the agreement of smart contract functionality and consensus mechanisms by which this agreement is reached, after which the smart contract is deployed.
- Execution of smart contracts. Smart contract execution that changes the state of the blockchain must be approved by the network. This set of policies describes the mechanism and consensus required for an execution of a smart contract to be accepted and its result stored in the blockchain.

Smart contract execution should allow for various policies depending on the scenario. The data handling scenario, for example, which we just covered, will be executed via smart contracts. In this scenario, when an organization wishes to upload some data concerning their activity, they should not wait for endorsement by other peers, the code that creates the anchored file should have a single endorsement. However, the code definition that describes the logic of how this anchoring is done (e.g., chunk size, encryption schema etc.) affects all peers and its update should therefore be decided under a schema requiring broader consensus. In contrast to most public blockchains where such on-chain policies are “hard coded” into the blockchain protocol and any changes in the consensus mechanism require hard forks of the blockchain, our architecture allows for on-chain management depending on the scenario; the HyperLedger Fabric which will be adopted by AI4Gov is an enabler for having such custom and configurable policies.

One special case that needs to be considered is the deletion of file. Under GDPR, the right to be forgotten seems to enforce the implementation of mechanisms for deleting files. Moreover, from a technical point of view, and since the files are just anchored in the blockchain and they physically reside in an off-chain storage, there is no mechanism which restricts the owner of just deleting

the file¹⁶. At first glance, this seems to limit the accountability feature of the blockchain, as party can “prove” that they provided some kind of information of the blockchain and then delete it with now apparent mechanism of third parties to verify that the proof was indeed submitted or deleted.

There are two measures handle this situation, both of which will be implemented in the decentralised data handling mechanism of AI4Gov:

- The first is implemented trivially by requiring that the anchor contains the hash of the anchored file. If the file is deleted and then a party tries to retrieve the file, they will be informed that the file is invalid, pointing out to data modification after the anchoring. If a requesting party requires proof, this then cannot be generated and, although the party cannot discern what exactly happened, the owner cannot claim that they possess the proof at the specific point in time. While this mechanism prevents the owner from retroactively changing evidence, it still has the drawback that proof of the original action is lost.
- The second one is to implement smart contracts that, when anchoring files, also store in the anchor any information required for evidence that is then permanently stored in the blockchain. For example, if an owner wishes to prove that a water sample is non-acidic, they can store the full report of the data and store the total pH value as metadata on the blockchain with the anchor. The smart contract agreed for this business logic will take the source file, compute the pH, store it and then anchor the report. This transaction, depending on the endorsement policy, will be validated by other peers so that when the pH value is stored in the blockchain, it will be in a mutually agreed way, agreed by specific business logic, and implemented in a specific smart contract. Thus, the result cannot be fabricated or denied later, even if the original report becomes unavailable.

Policy description	Type	Location	Policy(ies) adopted
Maintenance and upgrades of the blockchain source code	Source code	Off-chain	External governance
Read access of anchored data	Data	Off-chain and On-chain	Data owner and user access list defined by data’s owner

¹⁶ There are file systems that, exactly as the blockchain, are distributed and in which files are never deleted. The InterPlanetary File System (IPFS) is one such an example. Though IPFS will be considered (and is in fact used in the current framework’s prototype) for storing public documents efficiently, it will not contain any files that contain sensitive information.

Insertion of anchored data	Data	Off-chain and On-chain	Data owner
Versioning of anchored data	Data	Off-chain and On-chain	Data owner
Deletion of anchored data	Data	Off-chain	Data owner
Maintenance of smart contracts code	Smart contract code	On-chain	<ul style="list-style-type: none"> • Single node • Majority vote • Minimum number of endorsements • Unanimous vote
Validation of smart contract invocation results	Smart contract code	On-chain	<ul style="list-style-type: none"> • Single node • Majority vote • Minimum number of endorsements • Unanimous vote

Table 3: Decentralised Data Governance policies in AI4Gov

4 Technological enablers

This sub-section will describe the main aspects of the technology stack that will be used to implement the Decentralised Data Infrastructure together with all mechanisms which will allow the realisation of the Decentralized Data Governance Model and facilitate the usage of smart contracts and dApps by end users.

As explained, for the purposes of AI4Gov, a permissioned blockchain solution will be used, which is the most logical solution based on the need to filter user enrollment and access. The HyperLedger Fabric, developed by Linux Foundation, is the most widely used blockchain framework implementing a permissioned blockchain, with many adopters across the world. The European Blockchain Services Infrastructure (EBSI)¹⁷ and the IBM Blockchain¹⁸ are two typical examples of blockchains that are based on the HyperLedger Fabric.

The HyperLedger ecosystem, moreover, offers many other frameworks that could be used in the future to enhance the functionality of the infrastructure. HyperLedger Aries¹⁹, for example, facilitates Zero Knowledge Proof (ZKP)(Aad, 2023), a process by which the truth of a statement can be ascertained, proved and validated without returning any other kind of information related to the data that supports the claim (for example, such a query might be “does VVV have over 10 thousand registered voters”? A ZKP process would not need to retrieve the whole transaction and data history but only validate the truth or false of the statement). Integration of these frameworks in the solution, if the need arises, becomes much easier if the solution is based on the HyperLedger Ecosystem.

As mentioned, all smart contracts will be accessed via a wallet mechanism together with its dApp marketplace which will contain all required dApps that will be defined, designed and implemented for AI4Gov. For the wallet, the OpenDSU framework will be utilized. OpenDSU is a framework that strongly enforces the Self-Sovereign Identity (SSI) model in the implementation of dApps. Under the SSI, the users can claim and identify their identity in the eco-system, via which they can access the resources that they own.

The core aspects of these technologies that constitute the enablers for AI4Gov will be briefly described in the following sections; the reader can read the official documentation for more technical details.

4.1.1 HyperLedger Fabric

HyperLedger Fabric is a permissioned private blockchain solution that was developed by the Linux Foundation. In this section, we will describe its main aspects and how these affect and can be used to implement the Decentralised Data Governance Model.

¹⁷ <https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home>

¹⁸ <https://www.ibm.com/blockchain>

¹⁹ <https://www.hyperledger.org/projects/aries>

First of all, we will have a look at the main concepts that constitute a network. Figure 5 depicts such as an example taken from the official documentation of HLF²⁰. Table 4 gives a summary of the various items appearing in Figure 5 for reference; a more detailed explanation of the terms follows in the form of giving definitions and analysing the various terms appearing in them.

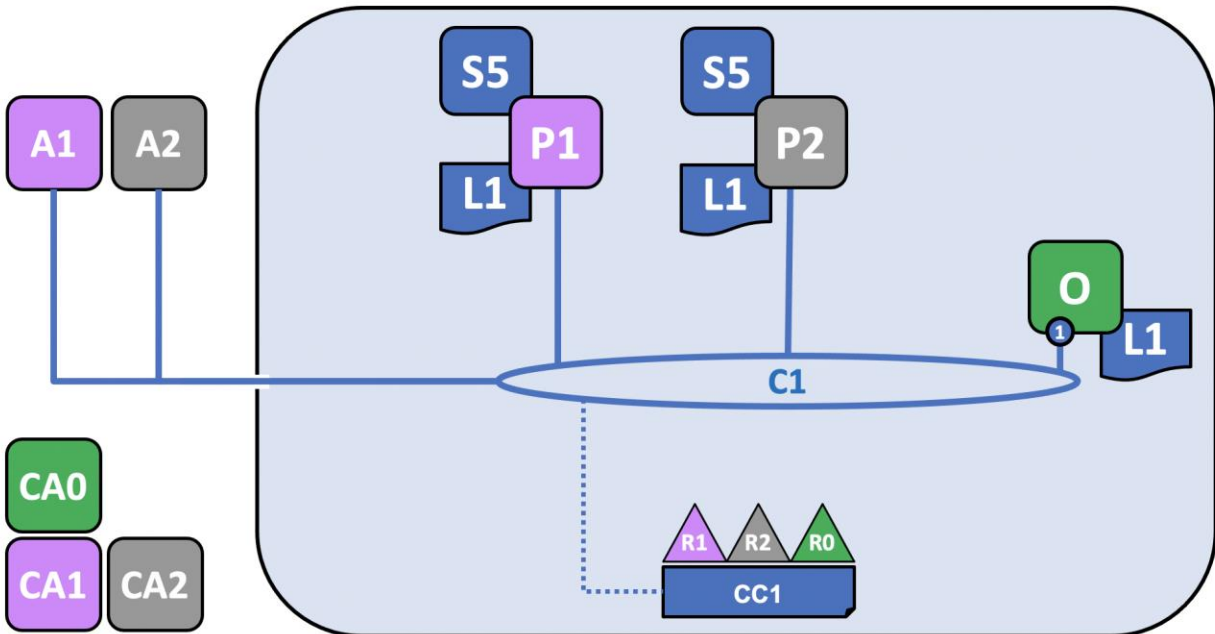


Figure 5: Hyperledger Fabric example configuration. Green, purple and grey colours correspond to org0, org1 and org2, respectively. Org0 is the orderer. A1 is an application that invokes P1’s endpoint, and A2 invokes P2’s endpoint, which should run the same chaincode after endorsement. CC1 (the channel configuration), L1 (the ledger) and S5 (the chaincode) are blue; this denotes that they do not correspond to a single organisation but are common to all.

Legend	Explanation	Colour scheme
R0, R1, R2	The organisations	Green, purple, grey depending on organisation
P1, P2	The peers (one for each one of the R1 and R2 orgs)	Purple, grey depending on the organisation

²⁰ <https://hyperledger-fabric.readthedocs.io/en/release-2.5/network/network.html>

Legend	Explanation	Colour scheme
CA0, CA1, CA2	Certification authorities	Purple, grey depending on the organisation
C1	Channel	N/A
CC1	Channel Configuration	Blue (common to all organisations)
L1	Ledger	Blue (common to all organisations)
S5	Chaincode	Blue (common to all organisations)
A1, A2	Applications (A1(2) is developed by R1(2) and invokes P1(2)'s endpoint	Purple, grey depending on the organisation the peer belongs to

Table 4: Summary of symbols appearing the the HyperLedger example network architecture depicted in Figure 5

Three organisations, R1, R2 and R3, wish to create a blockchain solution. They create a network configuration called CC1. This jointly agreed network configuration lists the definition of the organisation, their roles and the policies that apply to each role.

Definition 4.1 (Channel Configuration): A configuration that is agreed upon by the organisations and that defines which are the organisations of the channel, what are their roles and which are the policies regarding each role.

Once a channel configuration is defined, an empty channel will be created. A channel is a kind of network in which an organisation can interact via the set of policies defined in the channel configuration. A blockchain solution may have multiple channels, with users being part of any number of channels with various roles.

Definition 4.2 (Channel): A channel is a network between organisations. A channel has its own distributed ledger and can be accessed only by the participating organisations. Its policies are defined in the Channel Configuration.

In the example, channel C1 has been created and joined by the organisations. As mentioned, the organisation definitions must be provided for the configuration; however, since we have a permissioned blockchain, their identities should be proved by some agreed means. This role is accomplished by the Certificate Authorities (CAs) who provide the required X.509 certificates

(Cooper et al., 2008); it is via these certificates that ownership of components is proved. For AI4Gov, issuing of certification via Self-Sovereign Identity mechanisms and/or through eIDAS²¹ will be investigated.

The way that HLF maps certificates to member organisations is by a mechanism called the Membership Service Provider (MSP).

Definition 4.3 (Membership Service Provider): *An MSP is a data structure that defines how organisations are linked to a root CA.*

Organisations join the network as peer nodes. Peers play the role of nodes in HLF and may initiate transactions, invoke chaincode, propose changes, etc. Each peer has a copy of the ledger for each network it belongs to (peers can join multiple channels, and each channel has its own ledger). In the example, two peers, P1 and P2, have joined the channel, each one with its own copy of the ledger and its own copy of chaincode denoted by L1 and S5, respectively.

At this point, it is worth giving some clarification for terms that are generally used interchangeably but have subtle differences, especially in the HLF setting.

The terms ledger and blockchain have been used interchangeably, but in HLF, a blockchain is a sub-component of a ledger:

Definition 4.4 (Ledger): *A ledger is a collection of a blockchain and the state database of the channel. The blockchain of a channel contains all recorded transactions that took place in the channel. The state database contains the values of each recorded asset at a given time.*

Basically, the state database acts as an efficiency mechanism in HLF. Instead of having to transverse the whole blockchain to get the value of a recorded asset, this can be directly retrieved by the state database. In contrast to the blockchain, the state database is not immutable but is constantly updated to changes that are recorded by new blocks.

Another pair of terms is smart contract and chaincode. We give the following definitions:

Definition 4.5 (Smart Contract): *Any business logic that governs transaction and data access is called a smart contract*

Definition 4.6 (Chaincode): *The code that implements smart contracts is called chaincode*

Chaincode implements smart contracts, and under this distinction, a single chaincode package may refer to multiple smart contracts. As this distinction is irrelevant in most contexts, these terms are usually used interchangeably.

The term asset mentioned refers to any digitally represented piece of property that can have its ownership transferred via transactions invoked by chaincode; in fact, in HLF, and in contrast with other blockchains that allow direct transactions, the only way a state-altering transaction can occur is via chaincode.

²¹ <https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation>

Definition 4.7 (Asset): A key-value pair that is stored in the ledger and denotes transferable objects between users.

A special node of the R0 runs the ordering service denoted by O. The ordering service is responsible for collecting endorsed transactions and transforming them into a sequence of blocks that is then distributed back to the peers to be added to their blockchain copy.

Definition 4.8 (Ordering Service): The ordering service is run by one or more nodes and is responsible for collecting endorsed transactions, ordering them into a sequence of blocks and distributing them back to peers. Orderer nodes contain a copy of the blockchain but not of the world state.

The ordering service basically offers similar functionality as the PoW and PoS mechanisms of public blockchain. In a permissioned setting, however, there is no inherent reward system to give incentives for mining, instead, it is the mutual benefit of the joining parties that gives the incentive to “mine” blocks.

External applications may access the decentralized infrastructure and invoke chaincode. These are denoted as A1 and A2 in Figure 5 and can be connected to the HLF framework via its Gateway mechanism; the Gateway is the main mechanism by which external applications (dApps) can communicate with the underlying blockchain infrastructure.

After reviewing the main aspects of HLF, we can better understand how a fully developed HLF infrastructure would work. Figure 6 depicts a two-channel scenario, also from the official documentation, that is an extension of the basic one-channel scenario. Although a new CA was added for the new peer, the CA setting remains the same, as each user’s identity is unique and independent of the number of channels or the user’s participation in them. As such, they remain one for each user.

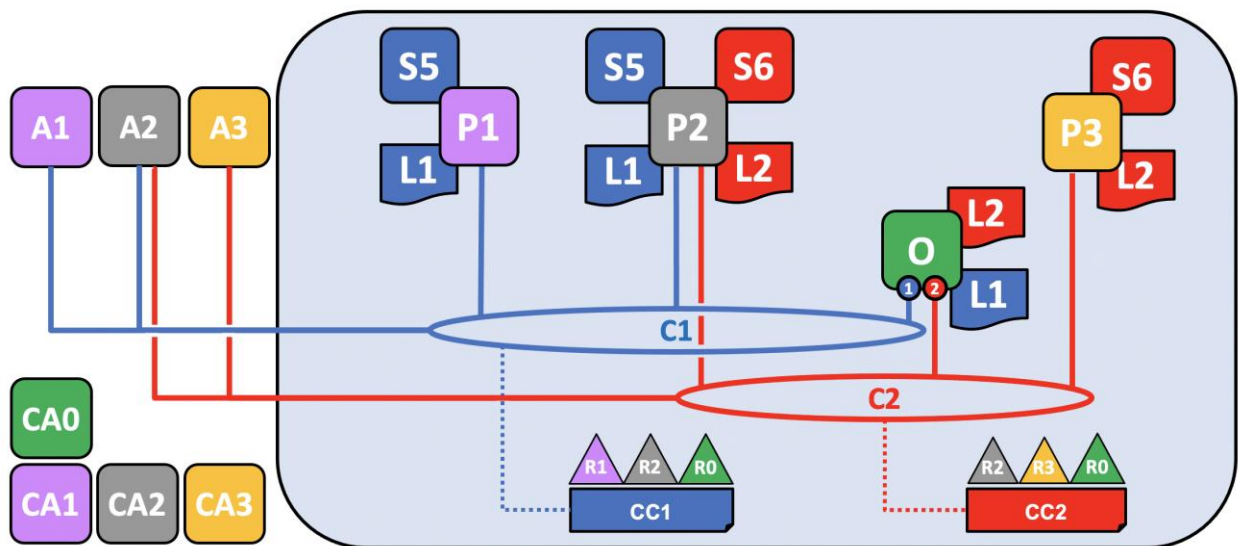


Figure 6: Hyperledger Fabric two-channel example configuration.

A second channel has been created and has its own configuration, denoted CC2. Only P2 from the original channel participates in the new channel along with another peer, P3. We can see that P2, which participates in both channels, has two copies of a ledger and a chaincode package, one for each channel in which it participates. The two channels also have the same orderer, which, likewise, has two copies of the blockchain. We will see in the next section (Section 4.1.2) how the capability of having multiple channels described here, together with the policy mechanism of HLF, allows for the implementation of custom governance models based on the business scenarios that are designed for each channel. From the present discussion, we can already see how HLF can customise the decentralised data flows and policies depending on the use case. For example, AI experts may share a dedicated ledger, which is used to validate ML analysis results by requiring that each one of them reproduce the same results. The AI experts may then participate in another channel shared with policymakers that may require read access to some data from generated reports (e.g., tracking bias trends).

4.1.1.1 Sample governance and execution scenario

Although the Decentralized Data Governance Framework is a work in progress and will be fully finalised by the time V2 of the Framework is released (D3.2), following an agile methodology, a prototypical backbone infrastructure has been implemented together with a mechanism to define flexible smart contract logic via smart contracts. This is in parallel with the evolution of the framework and with the implementation of other modules of the AI4Gov platform. As modules are developed and integrated, the business logic that can benefit from decentralisation will be incorporated into smart contracts; in this manner, the Decentralized Data Governance framework will be implemented into code in small consecutive cycles by updating its requirements and architecture, and then realising said updates. This process seems to mimic the Architecture Development Method (ADM) phase cycle of The Open Group Architecture Framework (TOGAF) (Lankhorst et al., 2005), and is, in fact, inspired by it, however the iterations also include the agile phases of development and testing.

Here, we design a generic scenario in which it is assumed that there is a Rule Engine implemented with a business logic that defines how rules are executed. A policymaker would wish to define a custom policy and check if certain KPIs are met. A development HLF infrastructure has been deployed in AI4Gov's VM infrastructure, which deploys a test network containing two peers and an orderer. Figure 7 depicts a snapshot of the test infrastructure running in the form of containerised services that were deployed using Docker. The test infrastructure consists of two organisations, each one having a Fabric-CA service²² (ca_org1 and ca_org2 containers), a CouchDB²³ service (couchdb0 and couchdb0 containers) that stores the world state of the ledger, and two peer nodes running (peer0.org1 and peer0.org2). Two more containers run the

²² This CA is used for development purposes; in a production environment, real and authoritative CAs will be used. Usage of eIDAS nodes will also be considered.

²³ <https://couchdb.apache.org/>

chaincode as a service²⁴ (dev-peer0.org1.example.com-rules_1.0 and dev-peer0.org1.example.com-rules_1.0 containers). A separate orderer node runs as a microservice (orderer.example.com); the orderer has its own Fabric-CA service running (ca_orderer).

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND	CR
e24f65691abc	hyperledger/explorer:latest	Up 23 minutes	0.0.0.0:8080->8080/tcp, ::8080->8080/tcp	explorer.nynetwork.com	"docker-entrypoint.s..."	Ab
5b6e28d54f2c	hyperledger/explorer-db:latest	Up 23 minutes (healthy)	5432/tcp	explorerdb.nynetwork.com	"docker-entrypoint.s..."	2
5b4c2d7c9f	dev-peer0.org1.example.com-rules_1.0	Up 22 minutes	0.0.0.0:8379543e4bd50f07e31ad0ae02afa74db075f9e0c5156f4fb76c223234d3ba7-ccc04e17e55ba00cb55c6e9cead850324b7032a9a64214f1cefb4320ec99b->chaincode-peer.add.	dev-peer0.org1.example.com-rules_1.0	"chaincode-peer.add..."	22
8379543e4bd50f07e31ad0ae02afa74db075f9e0c5156f4fb76c223234d3ba7	dev-peer0.org2.example.com-rules_1.0	Up 22 minutes	0.0.0.0:8379543e4bd50f07e31ad0ae02afa74db075f9e0c5156f4fb76c223234d3ba7-b7d38d4aeb592c9f67b1dd65b18ffcfc456d128a8e41264e55d65f4643b1301->chaincode-peer.add.	dev-peer0.org2.example.com-rules_1.0	"chaincode-peer.add..."	22
844d1b441159	hyperledger/fabric-tools:latest	Up 23 minutes		cli	"/bin/bash"	23
f269127c9c38	hyperledger/fabric-peer:latest	Up 23 minutes	0.0.0.0:7051->7051/tcp, ::7051->7051/tcp, 0.0.0.0:9444->9444/tcp, ::9444->9444/tcp	peer0.org1.example.com	"peer node start"	23
041c0a894085	hyperledger/fabric-peer:latest	Up 23 minutes	0.0.0.0:9051->9051/tcp, ::9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445/tcp, ::9445->9445/tcp	peer0.org2.example.com	"peer node start"	23
f0597e1a6a71	couchdb:3.1.1	Up 23 minutes	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp, ::7984->5984/tcp	couchdb1	"tini -- /docker-ent..."	23
7b1df7ba4db0	hyperledger/fabric-orderer:latest	Up 23 minutes	0.0.0.0:7050->7050/tcp, ::7050->7050/tcp, 0.0.0.0:7053->7053/tcp, ::7053->7053/tcp, 0.0.0.0:9443->9443/tcp, ::9443->9443/tcp	orderer.example.com	"orderer"	23
7bc410f73da	couchdb:3.1.1	Up 23 minutes	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp, ::5984->5984/tcp	couchdb0	"tini -- /docker-ent..."	23
075151a100c7	hyperledger/fabric-ca:latest	Up 23 minutes	0.0.0.0:7054->7054/tcp, ::7054->7054/tcp, 0.0.0.0:17054->17054/tcp, ::17054->17054/tcp	ca_org1	"sh -c 'fabric-ca-se..."	23
4b9dcf11a700	hyperledger/fabric-ca:latest	Up 23 minutes	0.0.0.0:9054->9054/tcp, ::9054->9054/tcp, 7054/tcp, 0.0.0.0:19054->19054/tcp, ::19054->19054/tcp	ca_orderer	"sh -c 'fabric-ca-se..."	23
cfe4109094c2	hyperledger/fabric-ca:latest	Up 23 minutes	0.0.0.0:8054->8054/tcp, ::8054->8054/tcp, 7054/tcp, 0.0.0.0:18054->18054/tcp, ::18054->18054/tcp	ca_org2	"sh -c 'fabric-ca-se..."	23

Figure 7: Deployment of the decentralised test infrastructure

We also assume that the user can define custom policies via a set of attributes that should have a schema that is compliant with the policy definition. Let's assume a simple scenario and how the technology can allow for customisation of it using off-chain and on-chain governance mechanisms. For the base scenario, we will assume that a consortium of policymakers agrees to implement a blockchain solution to verify that a policy conforms to certain environmental concerns. First of all, all policies should target low carbon emissions; let's call this value *threshold*. We can define a simple chaincode function *checkCarbonThreshold*, which, after the policy ID is given, it runs the check and returns *true* or *false* depending on the satisfaction of the target carbon emission KPI. Therefore, after the user enters the policy, two actions are performed:

- The policy definition is retrieved based on the ID entered, along with any metadata (e.g., timestamp), and the request data are entered in the ledger to record the request.
- The *checkCarbonThreshold* smart contract is run, and the policy is approved based on the result. The execution may also be stored in the blockchain for later inspection.

Figure 8 depicts the state of the ledger for the simple scenario using the visualisation tool HyperLedger Explorer²⁵. Via this interface or any visualisation tool that can read from the blockchain, peers can read and inspect all the transactions in the blockchain.

Suppose now that the legislature is changed and the carbon limit is decreased. It is easy to change the chaincode to perform the new check, but how is the new change going to be endorsed by the network? This will be defined by the policy. In this example, and since we have an issue of legal adherence, the policy may define that the chaincode may be changed by only a peer belonging to the admin organisation (or even an admin user) and that all peers should endorse changes once

²⁴ Chaincode as a service is a feature that was introduced in Fabric 2.0, and it allows the deployment of chaincode as a microservice that has a runtime that is independent of the peer services. This way, the chaincode can be managed without having to execute the commands within the peer environment, but by invoking the commands externally.

²⁵ <https://wiki.hyperledger.org/display/explorer>

it is approved by the admin. For other types of checks, a majority vote could suffice for changing the definition.

Scenarios can become more complicated as we allow for more functionality. For example, specific municipalities may implement a policy by adding additional restrictions, such as minimum proximity of infrastructure installation to settlements. More chaincode functions can be added that perform these checks; since further checks may not be legally binding and their specifics can be decided between the peers (e.g., municipalities), the endorsement policy can involve only such organisations (unanimous vote, majority vote, or whatever other schema needed).

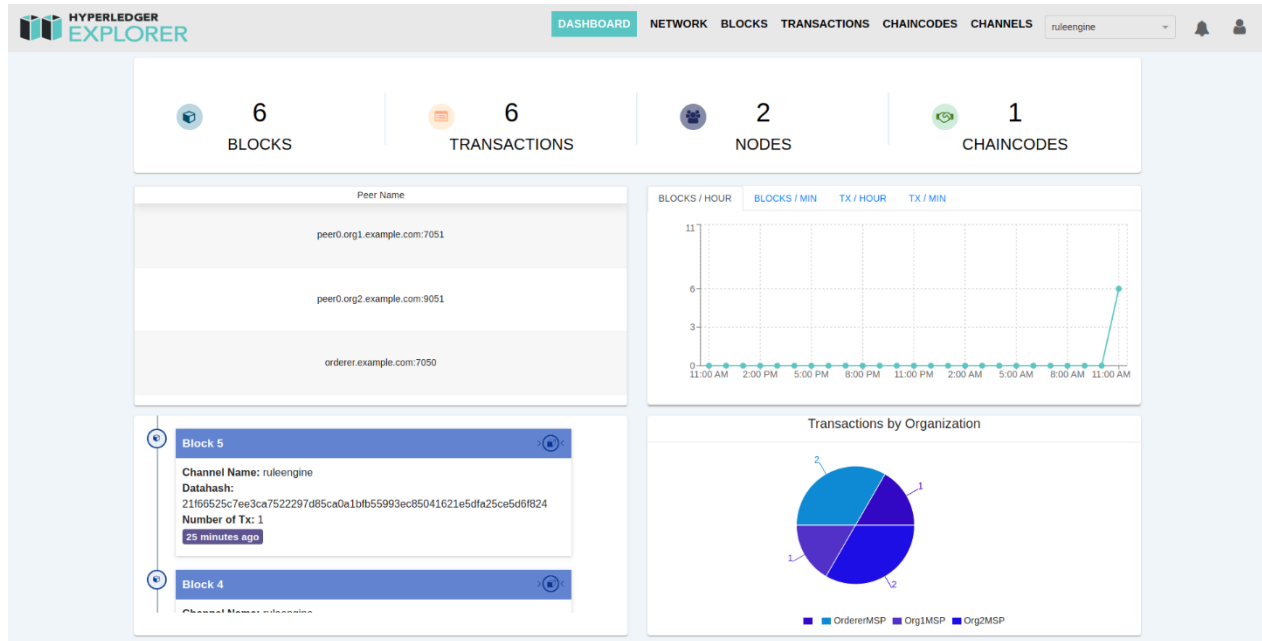


Figure 8: Inspecting the transactions and blocks via the HyperLedger Explorer

We can go even further than that. In a real-world scenario, it can be very common that a series of conditions may be needed to be verified, for which there does not exist a ready-to-use chaincode. Of course, the peer can implement such changes and propose them for endorsement, but this may be time-consuming and counter-productive. Specific to AI4Gov, this may be the case when a policymaker needs to search if she/he can create a new efficient policy by combining various aspects of existing policies and checking if the results pass certain criteria. She/he wishes to make this check using the established rules existing on smart contracts and demonstrate the validity of her/his results regarding the new policy to the whole consortium in a transparent way. The design, as this is also implemented in the prototype, allows for this by defining a special “super chaincode”, which accepts a series of checks that need to be performed in the form of a boolean expression. The function, in the prototypical implementation, is called *ruleEngine*, and an example input it can accept is depicted in the example listing depicted in Figure 9.

```

[
  "OR",
  {
    "check": "checkEmissionThreshold",
    "valueType": "literal",
    "values": [
      {
        "value": 1.7,
        "units": "ton"
      }
    ],
    "condition": "≤"
  },
  [
    "AND",
    {
      "check": "checkEmissionThreshold",
      "valueType": "literal",
      "values": [
        {
          "value": 1.75,
          "units": "ton"
        }
      ],
      "condition": "≤"
    },
    {
      "check": "factoryProximity",
      "valueType": "function",
      "values": [
        {
          "value": "func:distanceFromSettlement",
          "operands": "$City"
        }
      ],
      "condition": true
    }
  ]
]

```

Figure 9: Sample file containing the invocation of custom rule

In this scenario, the peer invokes the *ruleEngine* function and defines an expression, which needs to be evaluated against the rule engine. Let's call the custom policy that the user tries to validate "LocalFactoryEmissionPolicy". The expression is satisfied if LocalFactoryEmissionPolicy is true, (i.e., if the factory produces less than 1.7 tons of emissions) OR if the factory produces less than 1.75 AND is far from settlements. The tonnage checks involve condition evaluations based on literal values. The proximity check is denoted by a value which is of type *func*, which means that the "factoryProximity" function accepts an operand that is evaluated at run-time. The run-time evaluation involves the invocation of function *distanceFromSettlement* to evaluate the distance

of the factory from the nearest settlement. If this distance is above the tolerance that the *factoryProximity* smart-contract imposes, then the expression is satisfied. In this scenario, the policymaker allows for a small relaxation of the emission criteria, if the factory is built in an isolated location.

To demonstrate the ability to anchor data, the specific custom rule runs by submitting the custom execution to a file system that is then anchored to the blockchain. For this example, a local IPFS network was set up. From a business perspective, a user has defined and executed a custom rule off-chain, which was executed using conditions which they are executed and governed on-chain, while the submitted policy and its execution for specific values were stored off-chain and anchored in the blockchain. Although simple, this scenario demonstrates the versatility allowed by the underlying infrastructure. It also shows how accountability is achieved by anchoring data on the blockchain. In this example, the custom policy that the user has submitted for validation will be stored in IPFS; any policymaker or AI expert who wishes to inspect the custom policy definition can retrieve both the results and the specific version of the file that defined the rules and validate that the results can be reproduced.

4.1.2 Governance mechanisms under HyperLedger Fabric

The key enabling element of the HLF framework that can be used to implement the Decentralized Data Governance model is the policy mechanism of HLF. Policies can be set to define how a number of parameters of the infrastructure can be changed. Policies, among other things, define:

- How a peer joins or is removed from a channel
- How a change in the channel's chaincode base is approved by peers
- How a change of the ledger transaction is approved by peers.

These definitions require an initial off-chain agreement so that the initial configuration and policies can be bootstrapped. However, the initial policies may also include rules regarding how these can change. The so-called *Modification* Policy defines how many of the peers of the channel must agree before a change in the channel configuration takes place. That means that under HLF, an initial off-chain agreement may change on-chain if the initial configuration allows.

As mentioned, the policy definition allows for great flexibility in expressing policies. HLF offers various ways of defining rules for policy. For example, the Signature policy defines via a logical expression which types of users from each organisation must consent for the policy to be satisfied. Such a policy can be of a form like *OR('Org1.member', 'Org2.peer')*, which means that either a member of Org1 or a user with a *peer* role from Org2 must agree. Signature policies can be combined by a set of AND, OR and NOutOf operators to define custom rules. It is easy to verify that such rules allow for custom Governance models. If, for example, one policy contains only Org1.Admin defined for most aspects of the policies, including the Metapolicy policy, this model resembles the "benevolent dictator" model defined in 2.4.1. If the policies apply to the transaction policy domain as well, it goes even beyond that as peers cannot even endorse (validate) transaction. This is very close to a centralized infrastructure scenario, the main difference being that peers have always access to the shared information for inspection and verification of all changes.

Signature policies can be aggregated into a tiered hierarchy of policies, called sub-policies, into an *ImplicitMeta* policy. *ImplicitMeta* policies themselves can also be nodes of the hierarchical tree of an *ImplicitMeta* policy. These hierarchical definitions allow extension of the Signature policies by more easily including rules about generic types of users instead of referring to types within single organisations. Thus, they allow for flexible policies that do not depend only on the existing user base of the channel but provide mechanisms to define how future users will affect the governance of the chain.

4.1.2.1 *Smart Contract Governance*

Smart Contract Governance refers to how the business logic implemented by smart contracts in the blockchain is governed by members of the consortium. In HLF, this is directly translated into how the chaincode is endorsed in the network. This is defined in the so-called *LifecycleEndorsement*, which has a majority voting system by default (for existing and new peers) but can be configured to contain any policy rule. Although the exact mechanism of the Chaincode Lifecycle will be analysed more in-depth in V2, where the exact set of policies will also be defined, we can give a high-level description of how a new chaincode is submitted in an HLF channel.

- The code is packaged. This can be performed by any number of organisations.
- The code is installed on the peers. This step should be performed by all organisations that need to use the chaincode.
- The chaincode is approved. It is in this step that the *LifecycleEndorsement* is checked to endorse or not the new definition.
- The code is committed. If approval is achieved, the endorsements are collected from one organisation which submits the new chaincode. The new chaincode is now part of the channel's ledger.

4.1.2.2 *Data governance*

Data governance refers to how consensus is reached when the state of the blockchain is changed by transactions that aim to alter it. In HLF, all transactions are performed through chaincode, so data governance is equivalent to chaincode execution endorsement, i.e., the model which defines how the network decides to accept or reject a change proposed by a chaincode invocation. Similar to all cases described above, the endorsement policy is majority voting by default but can be configured to conform to any complex rule.

4.1.2.3 *User Roles in governance*

As mentioned, the way that each of the identified types of users of AI4Gov will be included in the decentralized data governance model and what obligations and rights it will have will be fully defined in V2. However, these roles can be to a first approach be mapped with HLF's user types. Briefly, HLF defines the following identity types:

- Admin: Admins perform administrative tasks such as registering a peer to a channel.
- Peer: A peer endorses or commits transactions; they are basically the nodes that are responsible for executing and maintaining the business logic defined in the code base of a ledger.

- Client: Clients interact and perform transactions on the network; roughly it corresponds to an end user invoking the services of the blockchain.
- Orderer: An orderer is a special node that, alone or with other orderer nodes, implements the ordering service.

Based on these types, the correspondence between AI4Gov roles and HLF identity types can be seen in Table 5. A special note is that for now, the orderer role has been assigned to the Admin group of AI4Gov. Ordering in HLF can be performed via various protocols, and since it involves a type of consensus (parties agree on the algorithm and execution that decides how sequence blocks are ordered), it may be considered to be shared by other user roles as well. This analysis, together with the decision about which exact ordering protocol will be implemented for the Decentralised Data Governance Model, will be documented in V2.

AI4Gov Role	HLF Identity Type	Description
AI Model Builder	peer	The AI Model builders must be able to install and maintain chaincode that runs AI models as smart contracts
Ethical Expert	client/peer	Ethical experts will be clients as they will invoke chaincode via dApps to retrieve results. The possibility for peers is included, in the case of ethical experts need to define and maintain smart contract logic for their business scenario.
Individual/Citizen	client	Individuals will access the network through dApps that invoke chaincode as clients.
Policy Maker	client/peer	Policy makers will be clients as they will invoke chaincode via dApps to retrieve results. The possibility for peers is included, in the case policy makers need to define and maintain smart contract logic for their business scenario.
Admin	admin/orderer	The admins will perform administrative tasks on the network, including ordering
External Sources	client	Any external source will access the network as a client.

Table 5: Correspondence of AI4Gov user roles to HLF identity types

4.1.3 OpenDSU

The OpenDSU framework is an approach to creating Self-Sovereign Identity (SSI) enabled Decentralized Applications (also called SSApps within the OpenDSU terminology). SSI (Baars, 2016; Preukschat & Reed, 2021) is a model of digital identity that allows the identity holder to have full control of her/his identity without the need for an arbitrary identity provider (e.g., Google). Under SSI users are identified and verified under the Verifiable Credentials Scheme (VCs) (“Verifiable Credentials Data Model 1.0,” n.d.). Under this model, users hold a unique identity that, in the context of decentralised SSI, is called Decentralized Identifier (DID). Using their DID, users can establish their identity to certificate issuers, which can then sign their credentials (an alumni member, for example, can use a dApp offered by their university and receive a certification that they have obtained a diploma). If a verifier then needs to check the validity of the user’s credentials, the verifier can resolve the signed DID document and confirm that this document was signed by a DID that belongs to the university (see Figure 10 for a graphical depiction of how SSI is implemented in EBSI).

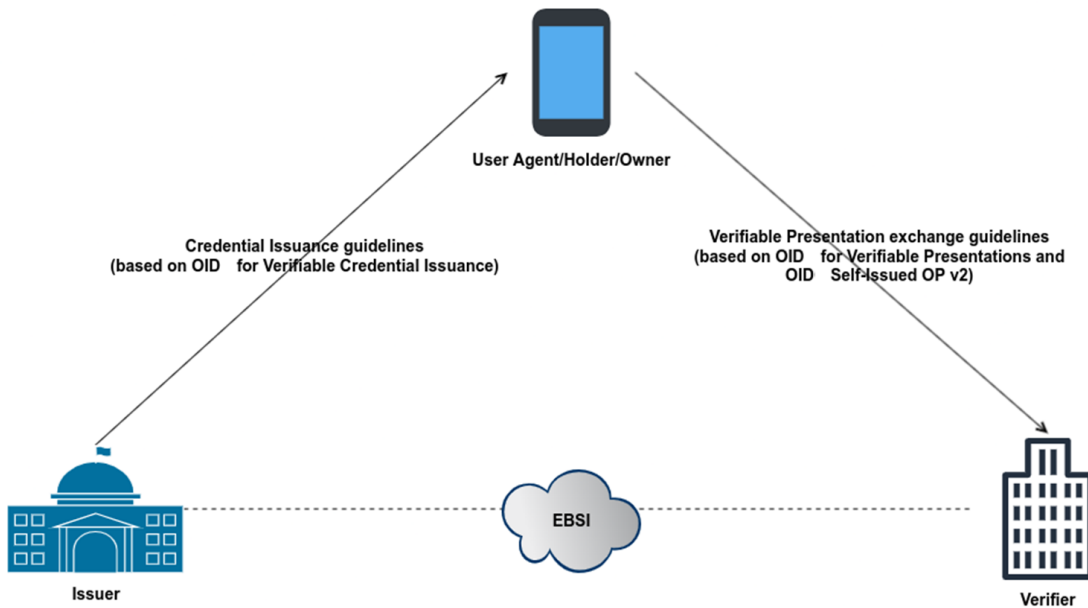


Figure 10: Verifiable Credentials in EBSI (source: <https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=555222155>)

Figure 11: dApps under the OpenDSU Framework shows the rough idea behind an OpenDSU-based dApp.

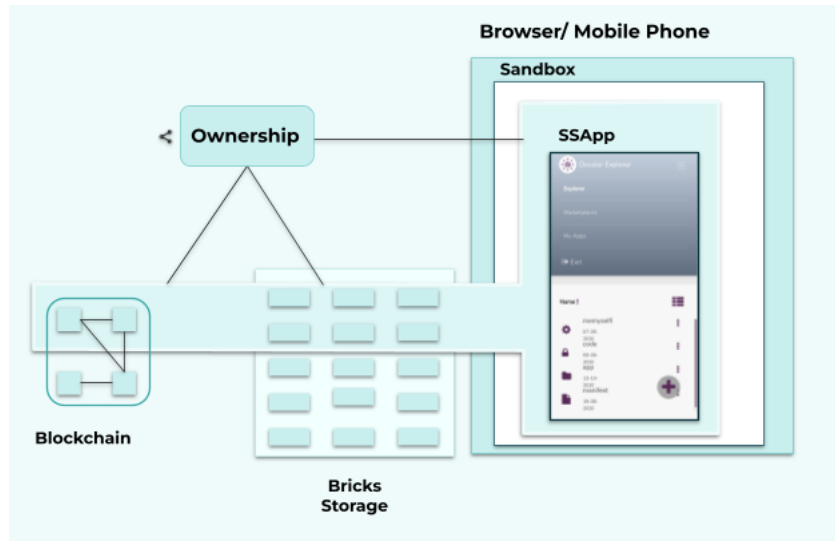


Figure 11: dApps under the OpenDSU Framework

The application, both in terms of components and in terms of data, is shared and stored in a special data storage called bricks storage. OpenDSU is agnostic regarding the details of the file storage implementation of a Brick Storage; they can be realised in a file system, the IPFS, stored in a SharePoint folder, etc. The bricks are encrypted and are anchored into a blockchain infrastructure; again, OpenDSU is blockchain agnostic, and any blockchain technology can, in theory, be used. Each time a user wishes to execute a dApp, they do so via their wallet, which, at a very basic level, translates into their keys. By using their keys, they, and only they, can retrieve the shards from the Bricks Storage by referencing the blockchain and reconstructing the whole execution environment of the respective dApp together with any data that the dApp requires.

With Self-Sovereign Identity being a major goal and driver for identity management both globally and at the EU level (see the ESSIF initiative²⁶), OpenDSU is a great enabler for achieving SSI within AI4Gov. While core backbone decentralized services will be run as chaincode and managed directly within the HLF infrastructure, any functionality that is exposed to end users via dApps, will go through the OpenDSU framework to provide SSI capabilities to end users.

²⁶ <https://essif-lab.eu/>

5 Data Governance Framework

5.1 General Guidelines and Policies

The Data Governance Framework (DGF) is a structured and comprehensive set of guidelines, policies, and procedures that govern how data is managed, shared, and protected within the AI4Gov Project. This framework serves as a crucial instrument for ensuring that data-related activities align with the EU's legal and regulatory landscape, particularly with regard to data protection and privacy. Within this context, the Data Governance Framework project plays a pivotal role in navigating the complexities of data management while complying with EU data protection laws. This framework acts as a structured roadmap that not only empowers project partners to harness the potential of data but also safeguards the rights and interests of individuals whose data is processed.

The DGF is aligned with the Data Governance Act while also taking into consideration key regulations such as GDPR, AI Regulation, EU AI Act and the Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment. To provide a concrete framework, policies and guidelines are generated for each of the above regulations that all partners within AI4Gov should take into consideration, focusing on all of the following factors:

1. **Compliance with Regulations:**

This factor emphasises the need to comply with data protection and privacy regulations. It includes ensuring that data handling practices align with the legal requirements imposed by such regulations, with a focus on key laws like the General Data Protection Regulation (GDPR).

- Stay informed about relevant data protection regulations in your region and industry.
- Appoint a Data Protection Officer (DPO) to oversee compliance.
- Regularly update data governance policies to align with evolving regulations.

2. **Data Ownership:**

Data ownership refers to the clear definition of who has control over the data. It ensures that data rights and responsibilities are well-defined among partners, especially in collaborative initiatives. Clear data ownership definitions help prevent disputes and maintain responsible data management.

- Define data ownership in partnership agreements, specifying rights and responsibilities.
- Establish data governance committees with representatives from each partner to address ownership concerns.
- Create data access and usage policies that respect data ownership and provide guidelines for shared data.

3. **Data Security:**

Data security is vital to safeguard data against unauthorised access and breaches. It involves implementing security measures such as encryption for data at rest and in transit, access controls, and regular security audits to identify and mitigate potential risks.

- Implement encryption for data at rest and in transit.
- Enforce access controls, ensuring that only authorised personnel can access and modify data.
- Regularly conduct security audits and vulnerability assessments to identify and mitigate risks.

4. Data Quality:

Data quality ensures that data used for analysis and decision-making is accurate, consistent, and reliable. It involves the development of data quality standards, validation processes, data profiling, and data cleaning to maintain high-quality data.

- Develop data quality standards and validation processes to maintain data accuracy and consistency.
- Implement data profiling and cleaning procedures to rectify inaccuracies and inconsistencies.
- Provide training to ensure that personnel understand the importance of data quality and their role in maintaining it:

5. Privacy by Design:

Privacy by design emphasises the integration of privacy safeguards into AI development and data handling processes from the project's outset. It includes practices like privacy impact assessments (PIAs) and data anonymisation to protect individual identities.

- Incorporate privacy impact assessments (PIAs) into the development of new projects and data initiatives.
- Use data anonymisation or pseudonymisation techniques to protect individual identities.
- Continuously assess and update privacy measures to adapt to changing risks and challenges.

6. Data Sharing Agreements:

Data sharing agreements are essential for defining the terms and conditions of data sharing, access, and usage. They ensure clarity and compliance in data sharing practices, including specifying data ownership and responsibilities. Regular review and updates are necessary to adapt to changing conditions.

- Draft comprehensive data sharing agreements that clearly specify data ownership, permitted uses, and responsibilities.

- Include provisions for data retention and disposal to maintain compliance with regulations.
- Regularly review and update data-sharing agreements to reflect changing needs and conditions.

7. Data Lifecycle Management:

Data lifecycle management involves a structured approach to data handling, ensuring data consistency from acquisition to disposal. It includes the development of data lifecycle plans, regular audits, and documentation of data retention and disposal processes.

- Develop a data lifecycle management plan to ensure data is handled consistently from acquisition to disposal.
- Regularly audit data storage and processing practices to identify inefficiencies or compliance issues.
- Document data retention and disposal processes to maintain transparency and compliance.

8. Ethical Considerations:

Ethical considerations focus on responsible AI practices and the prevention of bias and discrimination in AI applications. This involves conducting fairness and bias assessments, providing training to raise ethical awareness, and promoting transparency in AI development and deployment.

- Conduct fairness and bias assessments on AI models to identify and mitigate potential bias.
- Provide training to personnel involved in AI and data projects to raise awareness of ethical concerns.
- Encourage transparency by documenting AI model development and deployment processes.
- Take Assessment List for Trustworthy Artificial Intelligence (ALTAI)²⁷ for self-assessment into consideration.

9. Accountability:

Accountability ensures clear lines of responsibility within the partnership. It involves the appointment of Data Stewards to oversee data governance, defining roles for incident response and GDPR compliance, and establishing a Data Governance Committee for oversight.

²⁷ European Commission, Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment, <https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment>

- Appoint Data Stewards within each partner organization to oversee data governance practices.
- Create clear lines of responsibility for incident response and GDPR compliance.
- Establish a Data Governance Committee with representatives from all partners to oversee accountability.

10. **Monitoring and Compliance:**

Continuous monitoring, audits, and compliance assessments are essential to identify and rectify issues, ensuring ongoing data governance. This factor involves regular internal audits, documentation of practices, and the establishment of mechanisms for reporting and addressing data governance concerns and breaches promptly.

- Regularly conduct internal audits and compliance assessments to identify and rectify issues.
- Provide a mechanism for stakeholders to report data governance concerns or breaches for quick resolution.
- Take AI4Gov’s Data Management Plan (D1.2) into consideration regarding monitoring activities and compliance.

5.2 [Applicable Regulations and EU Guidelines](#)

5.2.1 [General Data Protection Regulation \(GDPR\)](#)

The General Data Protection Regulation (GDPR) is a comprehensive data protection and privacy regulation that was enacted by the EU, replacing the Data Protection Directive (95/46EC). The project’s data management plan includes all actions and guidelines for GDPR compliance; since it applies to all data, including decentralised data, it also applies to the Decentralized Data Governance Model. With decentralised storage, however, there is a caveat that needs special consideration.

The immutable nature of the blockchain means that any information stored there will always be there and will never be removed as long as at least one node continues to operate and keep a copy of the chain. This seems to contradict the “right to be forgotten” right of GDPR. Even if we assume that personal data is encrypted in the blockchain, there is still the possibility that a future data breach or exploit will break the encryption.

The following considerations and actions are specific to the Decentralized Data Governance Model and aim at fully respecting the “right to be forgotten” tenet of GDPR.

- All files are stored off-chain and can be accessed via the blockchain only through anchors; although the anchors will be immutable, the anchored files can be deleted, invalidating the anchor in the process.

- Unencrypted files in plain text will only contain public data (such as publicly available reports).
- Any file that has potentially sensitive information will be anonymised before being stored and anchored; even then, it will be in encrypted format.
- Anonymised and encrypted files can only be unencrypted by users owning the appropriate key pair. Only the data controllers will have such keys.
- Anonymised and encrypted files can be deleted by the data controllers from the off-chain storage. The anchor will become invalid and will not be able to get verified.
- Distributed storage technologies, such as IPFS, will only be used for public data.

On top of the above points, the OpenDSU mechanism of the wallet performs further encryption and sharing of data and allows fine-tuned data control to the wallet's owner, thus applying further data protection for end-users.

GDPR Compliance Guidelines

1. Legal and Regulatory Compliance (Article 5):
 - Ensure that all data processing activities comply with the GDPR and relevant data protection regulations.
2. Data Classification and Sensitivity (Article 5):
 - Classify data based on sensitivity and importance to determine appropriate safeguards and handling requirements.
3. Data Protection Officer (DPO) (Article 37, Article 38):
 - Appoint a Data Protection Officer if required by the GDPR and define their responsibilities.
4. Data Inventory and Mapping (Article 30):
 - Create a comprehensive data inventory and mapping to understand data flows, storage locations, and processing purposes.
5. Data Minimization (Article 5):
 - Collect and process only the data necessary for the project's objectives, adhering to the principle of data minimization.
6. Data Subject Rights (Articles 12-23):
 - Ensure that data subjects can exercise their rights, such as the right to access, rectify, and delete their data.
7. Data Processing Legal Basis (Articles 6 and 9):
 - Identify and document the legal basis for data processing activities within the project.
8. Data Protection Impact Assessments (DPIAs) (Article 35):
 - Conduct DPIAs for high-risk data processing activities and take measures to mitigate identified risks.
9. Privacy by Design and by Default (Article 25):
 - Integrate privacy into the project's design and development processes to ensure data protection is a core consideration.
10. Data Security (Article 32):
 - Implement strong data security measures, including encryption, access controls, and regular security audits.
11. Data Breach Response Plan (Articles 33 and 34):

- Develop a clear and documented plan for responding to and reporting data breaches in compliance with the GDPR.
12. Consent Management (Article 7):
 - If applicable, establish a consent management system for collecting, recording, and managing consent from data subjects.
 13. Third-Party Data Processors (Article 28):
 - Ensure that any third-party data processors involved in the project comply with GDPR and have appropriate data processing agreements in place.
 14. Data Transfer Mechanisms (Chapter V):
 - Implement lawful mechanisms for international data transfers, such as Standard Contractual Clauses (SCCs).
 15. Data Retention and Deletion (Article 5):
 - Define data retention policies and procedures to ensure data is not kept longer than necessary for the intended purposes.
 16. Data Access and Portability (Article 20):
 - Provide mechanisms for data subjects to access and receive their data, adhering to GDPR's data portability requirements.
 17. Training and Awareness (Article 39):
 - Conduct training for project stakeholders to increase awareness of data protection principles and GDPR compliance.
 18. Data Governance Policies and Procedures (Article 5):
 - Develop clear data governance policies and procedures that outline how data should be managed and processed within the project.
 19. Data Sharing Agreements (Article 28):
 - If data sharing occurs with external entities, establish clear data sharing agreements that include data protection clauses.
 20. Data Documentation and Records (Article 30):
 - Maintain detailed records of data processing activities, agreements, and compliance measures.
 21. Regular Auditing and Monitoring (Article 32):
 - Implement regular audits and monitoring to ensure ongoing compliance with the GDPR and other data protection regulations.
 22. Incident Response Plan (Articles 33 and 34):
 - Develop a response plan for handling and reporting data incidents as required by the GDPR.

5.2.2 EBSI Conformance

Created in 2018, the European Blockchain Services Infrastructure (EBSI) is the EU's "official" blockchain infrastructure. It operates with nodes across EU countries with the goal of offering its

services to organisations and citizens across Europe. Its business use cases currently aim at three domains, namely Verifiable Credentials, Track and Trace and Trusted Data Exchange. During the EBSI demo day, held in May 2022, various scenarios proved the ability to verify credentials using the underlying EBSI infrastructure; that means that EBSI, though still an active and ever-growing project, has proved its efficiency for cross-border credential certification.

If the trend continues, EBSI will be adopted in production, and it will be the main infrastructure for cross-border, SSI-enabled, cross-border transactions. As such, potential integration with the AI4Gov blockchain infrastructure and dApp ecosystem will be investigated during the design and implementation of the smart contracts and dApps required for the execution of the pilot use cases.

Two main aspects, however, can be identified at the present moment:

- **Wallet conformance.** A goal that can be set from the present moment is that the wallet that will be implemented for AI4Gov will conform to EBSI standards. This conformance is verified by a series of tests offered by EBSI. The tests differ depending on the role of the Wallet user (end-user or holder, issuer, verifier). For AI4Gov, it is expected that Holder Wallets will be implemented; however, if any issuer or verifier wallet application is needed, this, too, shall be tested for EBSI conformance. All DIDs used in AI4Gov will be fully compliant with the EBSI guidelines.
- **Usage of EBSI services.** This is related to the first one in the sense that EBSI services can be used only by conformant applications. This aspect will investigate which of the EBSI services that are offered or planned to be implemented can be used for AI4Gov (Identity provision via an authorisation endpoint or via the SSI eIDAS bridge is an example).

5.2.3 Ethics Guidelines for Trustworthy AI

The Ethics Guidelines for Trustworthy Artificial Intelligence was presented in April 2019 by EU's High-Level Expert Group on AI.²⁸ The Guidelines put forward a set of 7 key requirements that AI systems should meet in order to be deemed trustworthy.

1. Human agency and oversight:

- AI systems should empower human beings, allowing them to make informed decisions and fostering their fundamental rights. At the same time, proper oversight mechanisms need to be ensured, which can be achieved through human-in-the-loop, human-on-the-loop, and human-in-command approaches.

2. Technical Robustness and safety:

- AI systems need to be resilient and secure. They need to be safe, ensuring a fall back plan in case something goes wrong, as well as being accurate, reliable and reproducible. That is the only way to ensure that also unintentional harm can be minimized and prevented.

²⁸ European Commission, Ethics guidelines for trustworthy AI, 2019, <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>

3. Privacy and data governance:
 - Besides ensuring full respect for privacy and data protection, adequate data governance mechanisms must also be ensured, taking into account the quality and integrity of the data and ensuring legitimised access to data.
4. Transparency:
 - Data, system and AI business models should be transparent. Traceability mechanisms can help achieve this. Moreover, AI systems and their decisions should be explained in a manner adapted to the stakeholders concerned. Humans need to be aware that they are interacting with an AI system and must be informed of the system's capabilities and limitations.
5. Diversity, non-discrimination and fairness:
 - Unfair bias must be avoided, as it could have multiple negative implications, from the marginalisation of vulnerable groups, to the exacerbation of prejudice and discrimination. Fostering diversity, AI systems should be accessible to all, regardless of any disability, and involve relevant stakeholders throughout their entire life circle.
6. Societal and environmental well-being:
 - AI systems should benefit all human beings, including future generations. It must hence be ensured that they are sustainable and environmentally friendly. Moreover, they should take into account the environment, including other living beings, and their social and societal impact should be carefully considered.
7. Accountability:
 - Mechanisms should be put in place to ensure responsibility and accountability for AI systems and their outcomes. Auditability, which enables the assessment of algorithms, data and design processes plays a key role therein, especially in critical applications. Moreover, adequate an accessible redress should be ensured.

In the scope of the Data Governance Framework, a questionnaire has been created based on the Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment and is available in Appendix C. As the underlying AI models generated within the scope of AI4Gov are not mature enough to be assessed during this first deliverable iteration, the partners results will be included in D3.2, which is the second version of this document.

5.2.4 EU Artificial Intelligence Act

In April 2021, the European Commission proposed the first EU regulatory framework for AI as part of the EU's broader efforts to address the ethical and legal challenges posed by AI technologies. The AI Act, officially known as the Artificial Intelligence Act, serves as a comprehensive regulatory framework within the European Union, designed to standardize the development and deployment of artificial intelligence technologies. Its central mission is to ensure that AI systems are created and employed in a manner consistent with the ethical, legal, and safety standards upheld by the EU. The AI Act encompasses a wide range of AI applications, classifying them as either high-risk or low-risk based on their potential to cause harm, as it can be illustrated in the Figure 12:

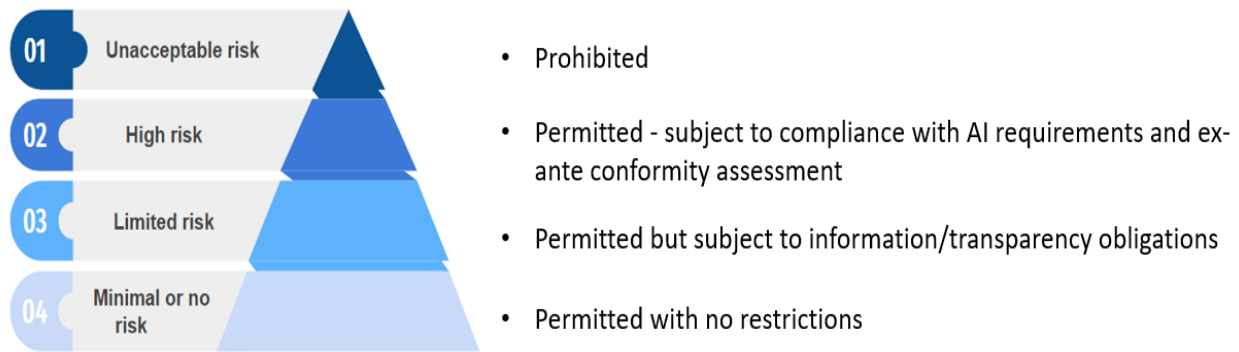


Figure 12: AI Act defined levels of risk

The regulation expressly prohibits certain AI practices, including government-based social scoring and the exploitation of individuals' vulnerabilities.

Transparency and accountability are also foundational principles of the AI Act, necessitating clear documentation, user information, and explanations for AI system decisions. Data governance, focusing on data quality and data protection, is also a critical component of the regulation. For non-compliance with its provisions, the AI Act outlines penalties and fines, with monetary penalties that can range up to €30 million or 6% of the violating entity's global annual turnover, contingent on the severity of the breach.

To oversee and coordinate the application of the regulation across EU member states, the European AI Board was established. Furthermore, the AI Act introduces a system for AI testing and certification to ensure that AI systems meet its requirements, thus fostering trust and accountability in the development and utilisation of artificial intelligence technologies within the European Union. The proposal of the EU AI Act will become law once both the Council (representing the 27 EU Member States) and the European Parliament agree on a common version of the text.

6 Conclusions

The present report provided the first iteration of the Decentralised Data Governance framework for AI4Gov. Though the decentralised data policies are yet to be defined per pilot case and per data type, V1 of the framework provides both the guidelines and the technology enablers that will allow fine-tuning of policies during pilot case execution. Off-chain policies are expected to govern core operational characteristics of the network (e.g., topology) and enforce data policies that are not expected or allowed to change in the future (e.g., core aspects of GDPR compliance). On-chain policies, on the other hand, will involve the whole consortium, both pilots of AI4Gov and future adopters, and will allow the collaboration in changes needed in the implemented business logic and/or the data flow scenarios.

Furthermore, and to align with current trends in Self-Sovereign Identity, the Decentralized Data Governance model defines a mechanism for implementing dApps based on SSI (also called SSApps) by leveraging the OpenDSU framework. The dApps will be contained in a Wallet, which will be developed with the aim of achieving EBSI conformance by being validated by the EBSI conformance test suite officially offered by EBSI.

Lastly, considerations regarding GDPR's right to be forgotten and its compatibility with the immutable nature of decentralised ledgers were taken into account; the report documented the methodological and technical mechanisms by which the right to be forgotten can be respected, even in a decentralised setting.

7 References

- Aad, I. (2023). Zero-Knowledge Proof. *Trends in Data Protection and Encryption Technologies*, 25–30. https://doi.org/10.1007/978-3-031-33386-6_6
- Baars, D. (2016). *Towards self-sovereign identity using blockchain technology*. <http://essay.utwente.nl/71274>
- Breidenbach, L., Cachin, C., Chan, B., Coventry, A., Ellis, S., Juels, A., Koushanfar, F., Miller, A., Magauran, B., Moroz, D., Nazarov, S., Topliceanu, A., Tramèr, F., & Zhang, F. (2021). Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Naorib.Ir*. <https://naorib.ir/white-paper/chinlink-whitepaper.pdf>
- Briola, A., Vidal-Tomás, D., Wang, Y., Letters, T. A.-F. R., & 2023, undefined. (2022). Anatomy of a Stablecoin's failure: The Terra-Luna case. *Elsevier*. <https://www.sciencedirect.com/science/article/pii/S1544612322005359>
- Cascone, S. (2021). Sotheby's Is Selling the First NFT Ever Minted – and Bidding Starts at \$100. *Artnet News*. <https://news.artnet.com/market/sothebys-is-hosting-its-first-curated-nft-sale-featuring-the-very-first-nft-ever-minted-1966003>
- Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. <https://doi.org/10.17487/RFC5280>
- Fischer, A., & Valiente, M. C. (2021). Blockchain governance. *Internet Policy Review*, 10(2). <https://doi.org/10.14763/2021.2.1554>
- Gurtu, A., & Johny, J. (2019). Potential of blockchain technology in supply chain management: a literature review. *International Journal of Physical Distribution and Logistics Management*, 49(9), 881–900. <https://doi.org/10.1108/IJPDLM-11-2018-0371/FULL/HTML>
- Haber, S., & Scott Stornetta, W. (1991). How to time-stamp a digital document. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 537 LNCS, 437–455. https://doi.org/10.1007/3-540-38424-3_32
- Hassan, S., & De Filippi, P. (2017). The Expansion of Algorithmic Governance: From Code is Law to Law is Code. *Field Actions Science Reports, Special Issue 17*.
- Kiyak, Y. S., Poor, A., Budakoğlu, I. İ., & Coşkun, Ö. (2022). Holochain: a novel technology without scalability bottlenecks of blockchain for secure data exchange in health professions education. *Discover Education*, 1(1). <https://doi.org/10.1007/S44217-022-00013-Y>
- Lankhorst, M., Iacob, M. E., Jonkers, H., Van Der Torre, L., Proper, H. A., Arbab, F., De Boer, F. S., Bonsangue, M., Hoppenbrouwers, S. J. B. A., Van Zanten, G. V., Groenewegen, L., Van Buuren, R., Slagter, R. J., Campschroer, J., Steen, M. W. A., Stam, A. W., Wieringa, R. J., Van Eck, P. A. T., Krukkert, D., ... Janssen, W. P. M. (2005). Enterprise architecture at work: Modelling, communication, and analysis. In *Enterprise Architecture at Work: Modelling, Communication, and Analysis*. <https://doi.org/10.1007/3-540-27505-3>

- Menezes, A. J. (Alfred J.), Van Oorschot, P. C., & Vanstone, S. A. (n.d.). *Handbook of applied cryptography*.
- Mora, C., Rollins, R., Taladay, K., ... M. K.-N. C., & 2018, undefined. (n.d.). Bitcoin emissions alone could push global warming above 2 C. *Nature.Com*. Retrieved October 15, 2023, from <https://www.nature.com/articles/s41558-018-0321-8%C2%A0>
- Nakamoto, S. (2017). Bitcoin White Paper. In *Blockchain* (Vol. 23, Issue 4).
- paper, V. B., & 2014, undefined. (n.d.). A next-generation smart contract and decentralized application platform. *Finpedia.Vn*. Retrieved October 18, 2023, from https://finpedia.vn/wp-content/uploads/2022/02/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
- Preukschat, A., & Reed, D. (2021). *Self-sovereign identity*. [https://books.google.com/books?hl=en&lr=&id=BfQ1EAAAQBAJ&oi=fnd&pg=PA1&dq=21\)+Self-sovereign+identity+:+decentralized+digital+identity+and+verifiable+credentials&ots=iCMGsy4ITq&sig=R6E_KkdK1-EX1x_CJt_MAF6OKug](https://books.google.com/books?hl=en&lr=&id=BfQ1EAAAQBAJ&oi=fnd&pg=PA1&dq=21)+Self-sovereign+identity+:+decentralized+digital+identity+and+verifiable+credentials&ots=iCMGsy4ITq&sig=R6E_KkdK1-EX1x_CJt_MAF6OKug)
- Sedlmeir, J., Smethurst, R., Rieger, A., & Fridgen, G. (2021). Digital Identities and Verifiable Credentials. *Business and Information Systems Engineering*, 63(5), 603–613. <https://doi.org/10.1007/S12599-021-00722-Y>
- Ursache, C., Sammeth, M., & Alboaie, S. (2022). *OpenDSU: Digital Sovereignty in PharmaLedger*. <http://arxiv.org/abs/2209.14879>
- Verifiable Credentials Data Model 1.0. (n.d.). *Www.W3.Org*. Retrieved October 22, 2023, from <https://www.w3.org/TR/vc-data-model/>
- Wilson, K. B., Karg, A., & Ghaderi, H. (2021). Prospecting non-fungible tokens in the digital economy: Stakeholders and ecosystem, risk and opportunity. *Business Horizons*, 65(5), 657–670. <https://doi.org/10.1016/j.bushor.2021.10.007>
- Windley, P. J. (2021). Sovrin: An Identity Metasystem for Self-Sovereign Identity. *Frontiers in Blockchain*, 4. <https://doi.org/10.3389/FBLOC.2021.626726/FULL>

APPENDIX A – Basics of blockchain by example

In this appendix, we will give a visual example of how a blockchain works, based on the online interactive demo of Anders Brownworth²⁹.

Starting from the basics, Figure 13 depicts how a series of transactions can be evaluated into a hash. This information denotes that Alice sends 10\$ to Bob, Tom sends 20\$ to Bob, and Bob invokes a smart contract by which he lends 20\$ to Peter. Bob pays 2.8\$ as transaction fees to invoke the smart contract. The transactions that the smart contract performs are not depicted to make the example easier to follow.

SHA256 Hash

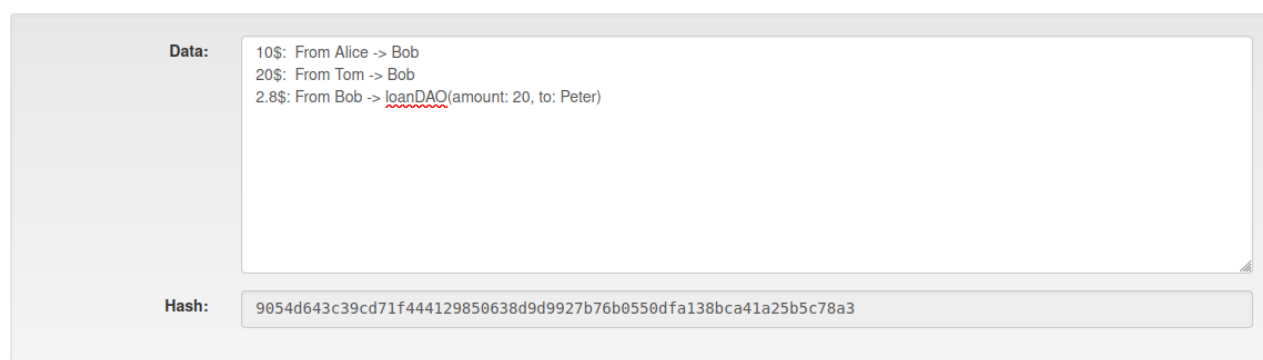


Figure 13: Evaluation of hashes

Let's introduce a mining problem with a set difficulty. The problem states that the hash must start with four zeros. The SHA256 hash that was computed (and indeed, most probably, any hash that records such transactions) is unlikely to have this characteristic. The hash is, therefore, invalid (Figure 14). To mine a valid block, the miner nodes include a special field (named nonce) and include this in the hash. The problem is solved when by trying a series of nonces, they compute a hash with four leading zeros (Figure 15).

These blocks can then be linked into a blockchain (Figure 16). An important aspect of a blockchain is that the hash is computed not only from the transaction data and the nonce but it also includes the hash of the previous block. Therefore, if any change happens in these blocks, the blockchain is invalidated.

Figure 17 depicts how a ledger becomes distributed by having peers hold one copy of the ledger each. If a peer proposes a corrupted block by changing anything in the above computation, the blockchain becomes invalidated (Figure 183), and this version will be discarded by the other peers of the network. The only way fraudulent blocks can be inserted is when a party has the majority of the computational power of the network. Then, by outperforming other miners, the party can

²⁹ <https://andersbrownworth.com/blockchain/>

consistently propose valid hashes with fake transaction data, which the network, because of the PoW mechanism, will accept.

Block

Block: # 1

Nonce: 72608

Data: 10\$: From Alice -> Bob
20\$: From Tom -> Bob
2.8\$: From Bob -> loanDAO(amount: 20, to: Peter)

Hash: 82569ea27abf15f501ef3a1ec2fd0ca4b2ae202dcd997181897ef8129781f75a

Mine

Figure 14: Inclusion of a difficulty problem for PoW

Block

Block: # 1

Nonce: 33989

Data: 10\$: From Alice -> Bob
20\$: From Tom -> Bob
2.8\$: From Bob -> loanDAO(amount: 20, to: Peter)

Hash: 00001c1d19c4d5f376820bc316ae02f488a57fed56410221c64312a67f2f4649

Mine

Figure 15: Mining a block

Block: # 1

Nonce: 33469

Data: 10\$: From Alice -> Bob
20\$: From Tom -> Bob
2.8\$: From Bob -> loanDAO(amount: 20, to: Peter)

Prev: 00

Hash: 00000c429e142987e159a5a26af2a8298ca4aa58b872e0512534a928a0

Mine

Block: # 2

Nonce: 12471

Data: 10\$: From Peter -> Alice
3.2\$: From Alice -> loanDAO(amount: 15, to: Peter)

Prev: 00000c429e142987e159a5a26af2a8298ca4aa58b872e0512534a928a0

Hash: 00001997978b0103c08895c08082477a71c25229ca122c38a29642a49fe

Mine

Block: # 3

Nonce: 48428

Data: 1.2\$: From Peter -> loanDAO(amount: 20, to: "Bob", action: "REPAY")

Prev: 00001997978b0103c08895c08082477a71c25229ca122c38a29642a49fe

Hash: 000024c3020ba3c87b38894641ca296697f86783ca34b1f82405092f

Mine

Figure 16: Example of blocks formed into a blockchain

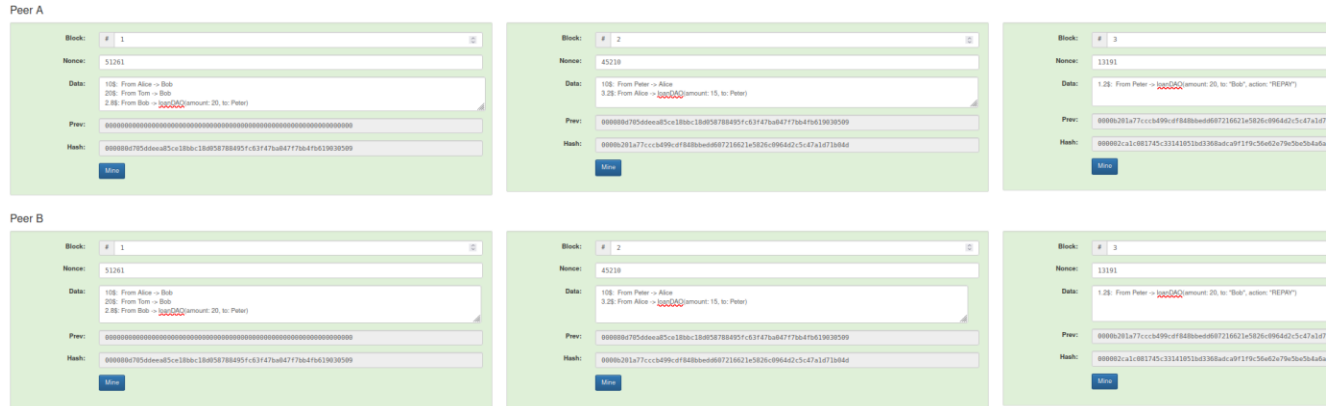


Figure 17: Distributed ledger

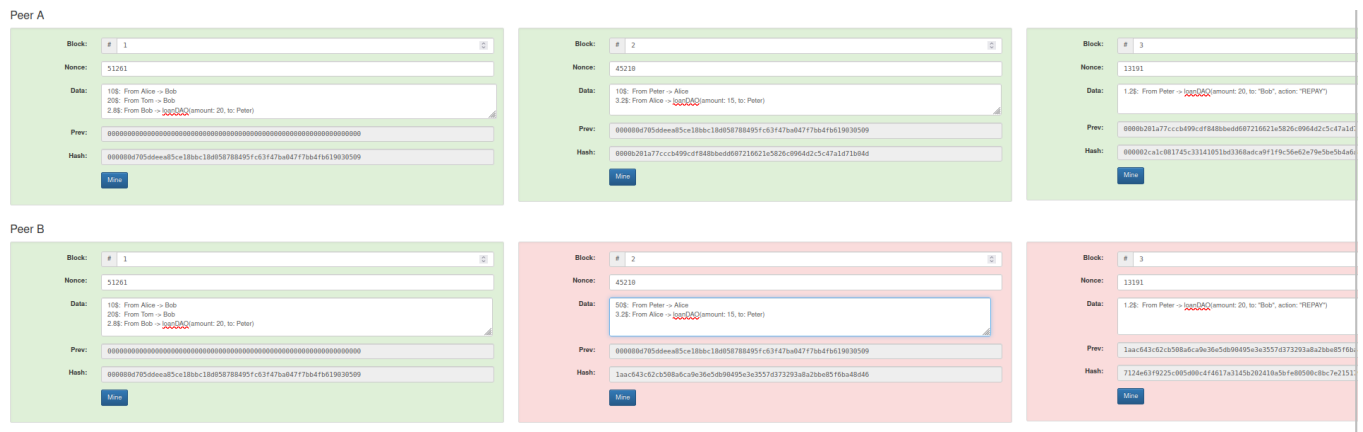


Figure 18: How a change invalidates the blockchain

APPENDIX B – Data Governance Framework Questionnaires

In parallel with the pilot definition, there was an effort to define aspects of data and data flows of all partners of the AI4Gov consortium. These aspects included:

- summary
- organisation documentation and metadata,
- accessibility data interoperability
- reuse and sharing,
- security and perseveration
- ethical aspects

See for the complete questionnaire.

Although the framework is still at an early stage to define all flows for all partners and use cases, the responses from all partners, especially from pilots, indicated that the data collected show a diverse nature, both on their technical aspects (size, format, etc.) and on the required privacy setting. The Decentralized Data Governance Framework described in the present report can be configured to access and transform the data according to the schemas and business logic defined, enhancing, apart from the transparency aspect of the data, both explainability in the case of reports and interoperability in the case of semantically similar data that use different schemas.

Data Summary: *The following questions aim to provide an overview of what types of data will be generated, collected and/or shared within AI4Gov project.*

- What type of data will you **produce or generate** during the Project?
- What type of data will you **collect** during the Project?¹
- How will you collect the data? In what formats?
- Will the provenance of the data be thoroughly documented using the appropriate standards?
- Will you use pre-existing data? From where?
- What is the expected size of the data that you intend to generate or re-use?
- Are there tools or software needed to create/process/visualize the data?
- Is there a storage and backup strategy in place?

2. Data Organization, Documentation and Metadata: *The following questions are intended to understand the plan for organizing, documenting, and using descriptive metadata to assure quality control and findability of the respective data.*

- What standards will be used for documentation and metadata (e.g., Digital Object Identifiers)?
- Do you use any best practices/guidelines for managing the data to publish (i.e., make available to third parties)?
- Do you use any tool for checking that the data are well formatted?
- What directory and file naming convention will be used?
- What project and data identifiers will be assigned?

- Is there a community standard for metadata sharing/integration?
- Will metadata be offered in such a way that it can be harvested and indexed?

3. Data Accessibility: The following questions aim to identify any data access and ownership concern.

- What steps will be taken to protect privacy, security, confidentiality, intellectual property or other rights?
- Does your data have any access concerns? Describe the process someone would take to access your data.
- Who checks the correct execution of the access process (e.g., lab, University, funder)?
- What procedures have you developed for the safe transfer of personal or sensitive data?
- Do you plan to make any research publications based on the data collected, processed or generated within the project? If yes, is it going to be openly available?

4. Data Interoperability: *What data and metadata vocabularies, standards, formats or methodologies will be followed to make your data interoperable and facilitate data exchange.*

- Will your data include qualified references² to other data (e.g. other data from your project, or datasets from previous research)?
- Will you data use a formal, accessible, shared, and broadly applicable language for knowledge representation?
- In case it is unavoidable that you use uncommon or generate project specific ontologies or vocabularies, will you provide mappings to more commonly used ontologies?

5. Data Reuse and Sharing: The following questions are intended to clarify how the collected data will be released for sharing and to evaluate their reproducibility.

- If you allow others to reuse your data, how will the data be discovered and shared? List the categories of data that will be made re-usable or openly accessible.
- Will the process of data generation be reproducible? What would happen if collected data got lost or became unusable later?
- What is the audience for the reused data? How are they potentially utilizing the data?
- Any restrictions on who can re-use the data and for what purpose?

6. Data Security and Preservation

The following questions are intended to clarify how the collected data will be preserved and archived.

- Will the data be safely stored in trusted repositories for long term preservation and curation? In what format?
- What provisions are or will be in place for data security (including data recovery as well as secure storage/archiving and transfer of sensitive data)?
- Who decides what data or what categories of data will be kept and for how long?
- Who will maintain the data for the long-term?
- Have you prepared a formal risk assessment addressing each of the major risks to data security and potential solutions?

- Any special privacy or security requirements (e.g., personal data, high-security data)?
- The GDPR requires personal data not be kept longer than necessary for the purpose for which it was stored. What protocol(s) will you put in place to ensure you delete personal data that is no longer required to be stored?

7. Ethical Aspects:

- What types of personal data do you intend to collect, generate or process?
- What types of sensitive data do you intend to collect, generate or process?
- Will any of the data subjects be children or vulnerable people?
- Will you be collecting personal or sensitive data from people who have not given their explicit consent to participate in the Project?
- If you collected personal data, as defined by the GDPR, which of the six Art. 6.1 bases will you rely on for the processing of each category of personal data?
- If you collected sensitive data, as defined by the GDPR, which of the ten Art. 9 bases will you rely on for the processing of each category of sensitive data?
- Have you already gained consent for data preservation and sharing from any data subject(s)?
- Will you engage in large scale or big data processing?
- Will any entity (including any service provider) outside of the E.U. have access to personal or sensitive data? If yes, please provide further information.

Table 6: DFG Questionnaire

APPENDIX C – ALTAI-driven Questionnaire

This questionnaire is prepared on the basis of the “Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment”, elaborated by the High Level Expert Group on Artificial Intelligence set up by the EC, which is the same group which delivered the Ethics Guidelines for Trustworthy AI. **Please refer to the guiding questions insert in ALTAI** to prepare your input: in each of the section of the questionnaire you can find in red the indication of the pages where the related guiding questions can be retrieved in ALTAI.

REQUIREMENT #1 Human Agency and Oversight
AI systems should support human agency and human decision-making, as prescribed by the principle of respect for human autonomy. This requires that AI systems should both: act as enablers for a democratic, flourishing and equitable society by supporting the user’s agency; and uphold fundamental rights, which should be underpinned by human oversight. Please assess your AI system in terms of their respect for human agency and autonomy as well as human oversight.
Human Agency and Autonomy
This subsection deals with the effect AI systems can have on human behaviour in the broadest sense. It deals with the effect of AI systems that are aimed at guiding, influencing or supporting humans in decision making processes, for example, algorithmic decision support systems, risk analysis/prediction systems (recommender systems, predictive policing, financial risk analysis, etc.). It also deals with the effect on human perception and expectation when confronted with AI systems that 'act' like humans. Finally, it deals with the effect of AI systems on human affection, trust and (in)dependence. <i>Guiding questions: please refer to ALTAI, p. 8-9</i>
Human Oversight
This subsection helps to self-assess necessary oversight measures through governance mechanisms such as human-in-the-loop (HITL), human-on-the-loop (HOTL), or human-in-command (HIC) approaches. Human-in-the-loop refers to the capability for human intervention in every decision cycle of the system. Human-on-the-loop refers to the capability for human

intervention during the design cycle of the system and monitoring the system's operation. Human-in-command refers to the capability to oversee the overall activity of the AI system (including its broader economic, societal, legal and ethical impact) and the ability to decide when and how to use the AI system in any particular situation. The latter can include the decision not to use an AI system in a particular situation to establish levels of human discretion during the use of the system, or to ensure the ability to override a decision made by an AI system.

Guiding questions: please refer to ALTAI, p. 9

REQUIREMENT #2 Technical Robustness and Safety

A crucial requirement for achieving Trustworthy AI systems is their dependability (the ability to deliver services that can justifiably be trusted) and resilience (robustness when facing changes). Technical robustness requires that AI systems are developed with a preventative approach to risks and that they behave reliably and as intended while minimising unintentional and unexpected harm as well as preventing it where possible. This should also apply in the event of potential changes in their operating environment or the presence of other agents (human or artificial) that may interact with the AI system in an adversarial manner.

Guiding questions: please refer to ALTAI, p. 10-12

Resilience to Attack and Security

General Safety

Accuracy

Reliability, Fall-back plans and Reproducibility

REQUIREMENT #3 Privacy and Data Governance

Closely linked to the principle of prevention of harm is privacy, a fundamental right particularly affected by AI systems. Prevention of harm to privacy also necessitates adequate data governance that covers the quality and integrity of the data used, its relevance in light of the domain in which the AI systems will be deployed, its access protocols and the capability to process data in a manner that protects privacy.

Guiding questions: please refer to ALTAI, p. 13-14

Privacy

Data Governance

REQUIREMENT #4 Transparency

A crucial component of achieving Trustworthy AI is transparency which encompasses three elements: 1) traceability, 2) explainability and 3) open communication about the limitations of the AI system.

Guiding questions: please refer to ALTAI, p. 15-16

Traceability

Explainability

Communication

REQUIREMENT #5 Diversity, Non-discrimination and Fairness

In order to achieve Trustworthy AI, we must enable inclusion and diversity throughout the entire AI system's life cycle. AI systems (both for training and operation) may suffer from the inclusion of inadvertent historic bias, incompleteness, and bad governance models. The continuation of such biases could lead to unintended (in)direct prejudice and discrimination against certain groups or people, potentially exacerbating prejudice and marginalisation. Harm can also result from the intentional exploitation of (consumer) biases or by engaging in unfair competition, such as the homogenisation of prices by means of collusion or a non-transparent market. Identifiable and discriminatory bias should be removed in the collection phase where possible. AI systems should be user-centric and designed in a way that allows all people to use AI products or services, regardless of their age, gender, abilities or characteristics. Accessibility to this technology for persons with disabilities, which are present in all societal groups, is of particular importance.

Guiding questions: please refer to ALTAI, p. 17-19

Avoidance of Unfair Bias

Accessibility and Universal Design

Stakeholder Participation

REQUIREMENT #6 Societal and Environmental Well-being

In line with the principles of fairness and prevention of harm, the broader society, other sentient beings and the environment should be considered as stakeholders throughout the AI system's life cycle. Ubiquitous exposure to social AI systems in all areas of our lives (be it in education, work, care or entertainment) may alter our conception of social agency, or negatively impact our social relationships and attachment. While AI systems can be used to enhance social skills, they can equally contribute to their deterioration. This could equally affect peoples' physical and mental well-being. The effects of AI systems must therefore be carefully monitored and considered. Sustainability and ecological responsibility of AI systems should be encouraged, and research should be fostered into AI solutions addressing areas of global concern, for instance the Sustainable Development Goals.³² Overall, AI should be used to benefit all human beings, including future generations. AI systems should serve to maintain and foster democratic processes and respect the plurality of values and life choices of individuals. AI systems must not undermine democratic processes, human deliberation or democratic voting systems or pose a systemic threat to society at large.

Guiding questions: please refer to ALTAI, p. 20-21

Environmental Well-being

Impact on Work and Skills

Impact on Society at large or Democracy

REQUIREMENT #7 Accountability

The principle of accountability necessitates that mechanisms be put in place to ensure responsibility for the development, deployment and/or use of AI systems. This topic is closely related to risk management, identifying and mitigating risks in a transparent way that can be explained to and audited by third parties. When unjust or adverse impacts occur, accessible mechanisms for accountability should be in place that ensure an adequate possibility of redress.

Guiding questions: please refer to ALTAI, p. 22-23

Auditability

Risk Management