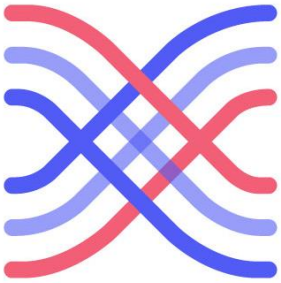


AI4Gov

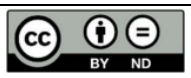
Trusted AI for Transparent Public Governance
fostering Democratic Values

Deliverable 3.4: Policy Recommendation Toolkit V2

29-03-2025

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
Beneficiary	UBI
License	 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/

AUTHORS		
	Name	**Organisation**
Document leader	Xanthi Papageorgiou	UBI
Participants	Nikos Kalatzis	UBI
	Konstantinos Tzelaptsis	UBI
	Septimiu Nechifor	SIE
	Raluca Maria Repanovici	SIE
	Iuliana Stroia-Vlad	SIE
	Dimitris Kotios	UPRC
Reviewers	George Manias	UPRC
	Silvina Pezzetta	WLC

VERSION HISTORY				
Version	**Date**	**Author**	**Organisation**	**Description**
0.1	01/02/2025	Xanthi Papageorgiou	UBI	ToC
0.3	10/02/2025	Xanthi Papageorgiou	UBI	Final Requirements
0.5	01/03/2025	Xanthi Papageorgiou, Nikos Kalatzis, Septimiu Nechifor, Raluca Maria Repanovici, Iuliana Stroia-Vlad	UBI, SIE	Final Architecture
0.7	12/03/2025	Xanthi Papageorgiou, Nikos Kalatzis, Konstantinos Tzelaptsis, Septimiu Nechifor, Raluca Maria Repanovici, Iuliana Stroia-Vlad Dimitris Kotios	UBI, SIE, UPRC	Prototype description
0.8	13/03/2025	Xanthi Papageorgiou, Dimitris Kotios	UBI, UPRC	Editing
0.9	26/03/2025	Xanthi Papageorgiou,	UBI	1st draft for internal review
0.9.1	27/03/2025	George Manias	UPRC	1st Internal peer review
0.9.2	29/03/2025	Silvina Pezzetta	WLC	2nd Internal peer review
1.0	29/03/2025	Xanthi Papageorgiou	UBI	Final draft addressing internal review comments

Table of Contents

Abstract **7**

1 Introduction **8**
1.1 Purpose and scope 8
1.2 Document structure 8
1.3 Updates since the previous version 9

2 Use Cases and requirements **10**
2.1 Semantic Interoperability 10
2.2 Open Democracy 13
2.2.1 Trust in an open democracy *14*
2.2.2 AI4Gov Implementation of homomorphic encryption *17*
2.3 AI In Policy Making 21

3 Architecture **23**
3.1 Business Layer 24
3.1.1 Semantic Alignment View *26*
3.1.2 Open Democracy View *26*
3.1.3 AI-based Policy Recommendation *27*
3.2 Semantic Layer 28
3.3 Application Layer 29
3.4 Technology Layer 30

4 Prototype Development **32**
4.1 Policy Recommendation Toolkit (PRT) 33
4.1.1 PRT Overview *33*
4.1.2 PRT Policy Creator *34*
4.1.3 PRT Explorer *43*
4.1.4 PRT Recommender *48*
4.1.5 PRT Modify Policy *49*
4.1.6 PRT Statistics *50*
4.1.7 PRT Wallet Registration *50*
4.2 Citizens' Wallet 51

5 Data Governance Framework **60**
5.1 DGF Overview 60
5.2 DGF KPIs introduced within the Policy Recommendation Toolkit 62

6 Conclusions **65**

7 References **66**

8 Appendix **67**

List of figures

Figure 1: The Ali-Cave. The Prover knows the combination of the lock that is deep in the cave. She wants to prove to the Verifier that she knows the code without disclosing it 15
Figure 2: The Prover follows a path and opens the door. The Verifier shouts a random path. The Prover is expected to appear on the entrance corresponding to the path that the Verifier called 16
Figure 3: Homomorphic encryption. Applying f directly to c(m) produces the output c(f(m)). Decrypting it, we get the same output as we would get if we applied the function directly in the plaintext data 17
Figure 4: Policy voting choices 18
Figure 5: Citizens' Votes as stored in Blockchain 19
Figure 6: Homomorphic encryption implementation on AI4Gov 20
Figure 7: AI4Gov Reference Architecture 24
Figure 8: PRT Architecture – High-level view 25
Figure 9: Semantic Alignment View 26
Figure 10: Open Democracy View 27
Figure 11: AI Recommendation Service 28
Figure 12: Semantic Layer View 29
Figure 13: PRT Architecture – Application Layer 30
Figure 14: PRT Architecture – Technology Layer 31
Figure 15: Homepage of the Policy Recommendation Toolkit (top view with functionalities) 33
Figure 16: Homepage of the Policy Recommendation Toolkit (bottom view with the categories) 34
Figure 17: Policy Creator interface 34
Figure 18: Fill form with the appropriate values in Waste Management category 35
Figure 19: Policy creation submitting process screen 36
Figure 20: Total policy recommendation and analytics results in Waste Management category 37
Figure 21: Fill form with the appropriate values in Traffic Management category 38
Figure 22: Total policy recommendation and analytics results in Traffic Management category 39
Figure 23: Fill form with the appropriate values in Drinking Water category 40
Figure 24: Total policy recommendation and analytics results in Drinking Water Management category 41
Figure 25: Fill form with the appropriate values in Sewage Water Management category 42
Figure 26: Total policy recommendation and analytics results in Sewage Water Management category 43
Figure 27: Policy Explorer 44
Figure 28: Policy Endorsement 44
Figure 29: Proof invitation for Policy Voting by citizen 45
Figure 30: Policy Vote by citizen (Accepted proof request) 46
Figure 31: Policy Vote by citizen (Successfully complete proof request) 47
Figure 32: Policy Voting by citizen 47
Figure 33: Policy Vote by citizen (Citizen rejected) 48
Figure 34: Recommendation results and KPI / constraints interface 49
Figure 35: Modification of policies and editing interface 49
Figure 36: Comparative analytics 50
Figure 37: Sharing QR code for citizen invitation 51
Figure 38: Form of attributes of verifiable credential 51
Figure 39: Citizens' wallet – Initial screen 52
Figure 40: Boarding invitation generated by HyperLedger Aries 53
Figure 41: Accepting the invitation 54
Figure 42: Accepting the Credential 54
Figure 43: Saved credentials in the wallet 55
Figure 44: Select policy to Vote 56
Figure 45: View the policy details 56
Figure 46: Vote Policy for three options (positive, negative, neutral) 57
Figure 47: Successfully voting 58
Figure 48: Ballot results after completing vote 59
Figure 49: Visualization Workbench via mobile 59
Figure 50: High-level illustration of DGF structure (1/2) 61
Figure 51: High-level illustration of DGF structure (2/2) 62

List of Tables

Table 1: Semantic Interoperability in AI4Gov Use Cases 11
Table 2: Digital Open Democracy in AI4Gov Use Cases 13
Table 3: Vote Policy request 17
Table 4: AI-enabled Policy Making in AI4Gov 21
Table 5: Request to AI analytics for Waste Management Category 36
Table 6: Request to AI analytics for Traffic Management Category 38
Table 7: Request to AI analytics for Drinking Water Category 40
Table 8: Request to AI analytics for Sewage Water Category 42
Table 9: DFG KPIs introduced to the PRT platform 63

Abbreviations

Abbreviation	Description
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
SSI	Self-Sovereign Identity
PRT	Policy Recommendation Toolkit
ZKP	Zero Knowledge Proofs
DGF	Data Governance Framework

Abstract

This document presents the final implementation of the Policy Recommendation Toolkit (PRT), which aims to facilitate organizations in policy-making and provide a transparent governance model that gives the opportunity to citizens to audit these processes of policy-making and to actively participate in the formation of them via blockchain-enabled co-creation. In this final implementation, the requirements derived from the pilots are finalized; the requirement engineering process follows an agile methodology, and the core aspects of the system are identified. These aspects are consolidated in the PRT architecture, which is another outcome of the present deliverable. Additionally, a wallet has been implemented based on Verifiable Credential in order to engage the citizens in the policy recommendation process. Integration of Blockchain Infrastructure with citizen wallet and PRT have been analysed in the final version of the architecture. Implementation of the PRT and integration into both the Visualization Workbench and the decentralized infrastructure has finalized; the final platform is given in the present report.

1 Introduction

1.1 Purpose and scope

The present document offers the 2nd and final iteration of the Policy Recommendation Toolkit (PRT). Following the user stories, it derives the set of user requirements and, based on these requirements, presents the architecture that fulfils them. In accordance also with the methodology followed in D3.1 and D3.2, the final version of the architecture is presented and incudes the business and application layers. The technical layer (application and infrastructure) is also be presented in this 2nd and final iteration of the document.

Although the pilot use cases focus on data and actors from organizations, the present document extends, whenever applicable, the use cases to citizens and have them to participate in the policy-making process, improving participatory governance and policy-making. Following the model of open democracy and recognizing that policies primarily affect the citizens, we leverage the decentralized infrastructure defined in D3.1 and D3.2 to facilitate the participation of citizens in the policy-making process. This is achieved via various means that are implemented via self-governed smart contracts, which also allow citizens to form opinions, provide feedback to policymakers and vote for recommended policies. The developed wallet component facilitates citizen engagement through a DAO-based voting mechanism. The citizen wallet integration leverages a Verifiable Credential based solution in order to ensure trust between citizens and public authorities. These mechanisms have been implemented, with the present document providing the overview of the technology enablers and the implementation that complement those proposed in the Decentralized Data Governance Framework (D3.1 and D3.2) and are specific to the citizen user group. Furthermore, the implementation is described.

1.2 Document structure

The present document is structured as follows:

- Section 1 contains the present introduction.
- Section 2 lists the requirements and briefly presents the new technology enablers that have been leveraged for implementing the PRT.
- Section 3 presents the various layers of the architecture.
- Section 4 presents the developments and the finalization of the PRT along with the citizens' wallet.
- Section 5 gives a series of horizontal Key Performance Indicators (KPIs) that have been developed and integrated into the Policy Recommendation Toolkit, following the rules and guidelines of the Data Governance Framework (DGF).
- Section 6 gives the conclusions of the present work.

1.3 Updates since the previous version

In this section, the major updates incorporated in the context of this deliverable are highlighted, in comparison to the first version of the series of deliverables related to the Policy Recommendation Toolkit.

Firstly, Section 2 introduces a new subsection that elaborates on AI4Gov's implementation of homomorphic encryption, focusing on its key features and practical applications.

Next, Section 3 presents the finalized architecture, introducing a new subsection that details the Technology Layer, which enhances the overall implementation of the final PRT and citizen wallet.

Also, Section 4 details the refined PRT and implemented citizen wallet, providing a comprehensive overview of their finalized structure and functionalities.

Finally, Section 5 introduces a series of horizontal Key Performance Indicators (KPIs), developed and integrated into the Policy Recommendation Toolkit in accordance with the Data Governance Framework's (DGF) rules and guidelines.

2 Use Cases and requirements

A policy-making process can be defined as a collaborative process that involves interest groups and analytical frameworks with the goal of forming a common set of goals and actions (Thatcher et al., 2015). To facilitate the optimal design of policies, it is essential that large groups of affected stakeholders are able to form networks in order to communicate ideas and needs and form policies using a co-creation process so that all stakeholder interests are imprinted in the resulting policies. In the domain of public policies, in particular, this process needs not only to involve large segments of the public sector but also to be transparent to the public; the citizens should be able both to co-design policies and audit them.

Modern trends in digitalization and AI can help these processes grow and enforce transparency in various facets of their execution. Although the facets of the processes that can be enhanced by digitalization are interconnected, we can roughly separate them into the following categories.

- Semantic interoperability: Policies often depend on terminology, models, and datasets that are used and understood at different levels by interested stakeholders. Semantic interoperability mechanisms ensure that when policies are defined, their constituents have a specific meaning that is understood unambiguously by all interested parties.
- Promotion of inclusiveness, responsiveness and accountability by enabling the model of Open Democracy (Landemore, 2020). Technological enablers such as the blockchain technology can help institutions and citizens to participate in activities of policy formation in its various phases, from consultation to voting. Decentralization can also act vertically through all aspects of digitalization by enforcing trust.
- Recommender systems can help produce optimal policies by solving optimization problems and suggesting candidate policies according to the constraints set, greatly reducing the complexity of designing a policy from scratch. With the advent of AI, these optimization processes can produce solutions that are very close to global optima; moreover, by using generative AI, new areas and potential actions can be explored by searching through the available datasets.

In the following sub-sections, the following axes are described, together with a mapping that shows how they can be leveraged to enhance policy-making in the AI4Gov Pilot Cases.

2.1 Semantic Interoperability

From the first days of the Semantic Web (Semantic Web – W3C), the goal of semantic interoperability is to provide unambiguous meaning to data exchanged between information systems. In practice, this can be very difficult since these meanings depend on context and are often shared between systems and processes that were not designed initially to be working together. The word “safe” for example, can have a different meaning depending on the domain (e.g., mean an acceptable level of emission in the domain of green growth or applied in the working conditions in labour policies). When, as it is commonly the case, policies involve multiple

domains and different datasets are combined, the categorical labels of data that have a semantic relation, have to be grouped together. While this process can be done manually by a data curator, it is often tedious and, in the case of large datasets, could prove impossible. Novel techniques that involve Knowledge Graphs and AI, however, can be used to cluster entities that are semantically interlinked.

For the use cases of AI4Gov, the requirements for semantic interoperability can be seen in Table 1. We distinguish two cases:

- The pilots in isolation. This is represented in the first three rows of the table. For this case, the main requirement is to uplift the data to a taxonomy so that it can be readily interpreted and consumed by a 3rd party.
- Collaboration between pilots, in the two last rows. This is the case in which water management policy design (DPB) and waste management policy design (VVV) can be benefited by policy data maintained by JSI. How this is done is investigated in the AI Policy Making Section, however it requires that data from the two sources are aligned via an appropriate model.

Table 1: Semantic Interoperability in AI4Gov Use Cases

Use Case	Pilot	Data	Users	Requirements
Water Management – drinking water	DPB	-Sewage Treatment data -Water cycling billing data -Streaming sensor data -Citizen wallet data -Explainability reports -Bias reports	-Workers at the municipal consortium for water management -Local administration -Citizens	-Uplift data
Water Management – sewage water				
IRCAI global 100 projects	JSI	-IRCAI data of projects submitted (textual	-Teams in private or public Institutions/Organizations that are submitting	-Uplift report metadata
SDG Observatory				

OECD policy document analysis		description, URLs) -Event Registry data (news and event items) -OECD AI policy initiatives - Explainability reports - Bias reports	projects to the IRCAI Global Top 100 program. -Government -Corporate -Researchers	
Parking tickets monitoring	VVV	-Census data -Household water data -Tourist data (arrivals, overnight stay, cruise data) -Airport traffic data - Municipality events attendance data -Citizen wallet data - Explainability reports - Bias reports	-Policy makers -Citizens	Uplift data
Waste management – Pay as you Throw				
Water Management policies	DBP -JSI	Data from DBP and JSI	-Policy makers -Citizens	Align data

Waste Management policies	VVV -JSI	Data from DBP and JSI	-Policy makers -Citizens	Align data

2.2 Open Democracy

While, in theory, citizens have access to public information, can monitor government and participate in public consultation, participation in these processes is often hindered in practice. Citizens often have to actively search for the appropriate channels, while efficient participation in the public consultation may be poisoned by deep fake, paid digital accounts and bots.

Blockchain is a technology enabler that can facilitate inclusive democratic processes while helping avoid the aforementioned caveats. The blockchain enabler and underlying infrastructure that is going to be used in AI4Gov to implement a fully decentralized data governance framework has been documented in D3.1 and D3.2. In this section, we are describing how the decentralized infrastructure, along with the usage of smart contracts and the on-chain data governance framework, can be used to activate citizens in policy making. The requirements for this are listed in Table 2. Briefly, the affected actors can be separated into two categories:

- Policymakers (as members of public institutions) propose policies and define the governance policies by which the policies can be endorsed.
- Citizens can vote on the proposed policies and can demand to retrieve explainability reports if the policies are based on the output of an AI algorithm.

In case where an AI output is produced deterministically, it can also run on the blockchain as a smart contract. In this case, the citizens have an extra tool for auditing, as they can validate the models by rerunning them on their nodes.

The cases have been limited to two pilots, namely DPB and VVV, since these pilots directly involve policymaking that has the potential to engage stakeholders from the whole spectrum, from public servants to citizens. The special user group “Governing body” is reserved for the users that have the right to alter the policies by which the blockchain is governed (e.g., change the voting system from unanimous to majority, change the definition of the code running a recommender system, etc.). The physical users to which this group corresponds may vary depending on the use case. For the DPB and VVV, these would be the policy makers (the users of the institutions).

Table 2: Digital Open Democracy in AI4Gov Use Cases

Use Case	Pilot	Data	Users	Requirements
Water Management – drinking water	DPB		-Policy Maker	-Propose policy -Alter policy

		-Sewage Treatment data -Water cycling billing data		-Vote policy
			-Governing body	-Define governance model
Water Management – sewage water			-Citizens	-Vote policy -Audit AI
Parking tickets monitoring	VVV	-Census data -Household water data -Tourist data (arrivals, overnight stay, cruise data) -Airport traffic data -Municipality events attendance data	-Policy makers	-Propose policy -Alter policy -Vote policy
			Governing body	-Define governance model
Waste management – Pay as you Throw			-Citizens	-Propose policy -Alter policy -Vote policy

2.2.1 Trust in an open democracy

One of the key challenges that any digital platform that tries to implement mechanisms of open democracy has to face is that of trust. Citizens avoid entering an open and inclusive platform if it lacks the appropriate transparency and trust mechanisms. Any such platform should guarantee that:

- Any feedback and consultation that is signed by the citizen cannot be altered or isolated in any way.

- Both the governance process and the mechanisms by which this process can change are clear to citizens.
- Any piece of evidence (e.g., OECD report) that is used for forming a policy can be retrieved and inspected by citizens.
- Secrecy of vote should be possible.

The first three elements are typical use cases of a blockchain infrastructure: the technological enablers and the mechanisms for enforcing them are documented in D3.1. The secrecy of the vote, however, is a new requirement that is specific to the PRT and has to be treated separately.

At first sight, providing vote secrecy seems incompatible with the nature of the blockchain; ballots need to be counted by the smart contract that implements the voting mechanism and have to be recorded into the blockchain in plain view of all peers. However, by following certain cryptographic protocols, such as the Zero Knowledge Proof (ZKP) (Feige et al., 1987), this can be used to prove certain statements without disclosing further information.

A graphical way of understanding the basic idea behind ZKP is the story of the Ali Baba cave. The setting is depicted in Figure 1. The cave has a door that connects paths A and B. The door has a code that the Prover knows. She wants to prove to the Verifier that, indeed, she knows the code without disclosing the code itself.

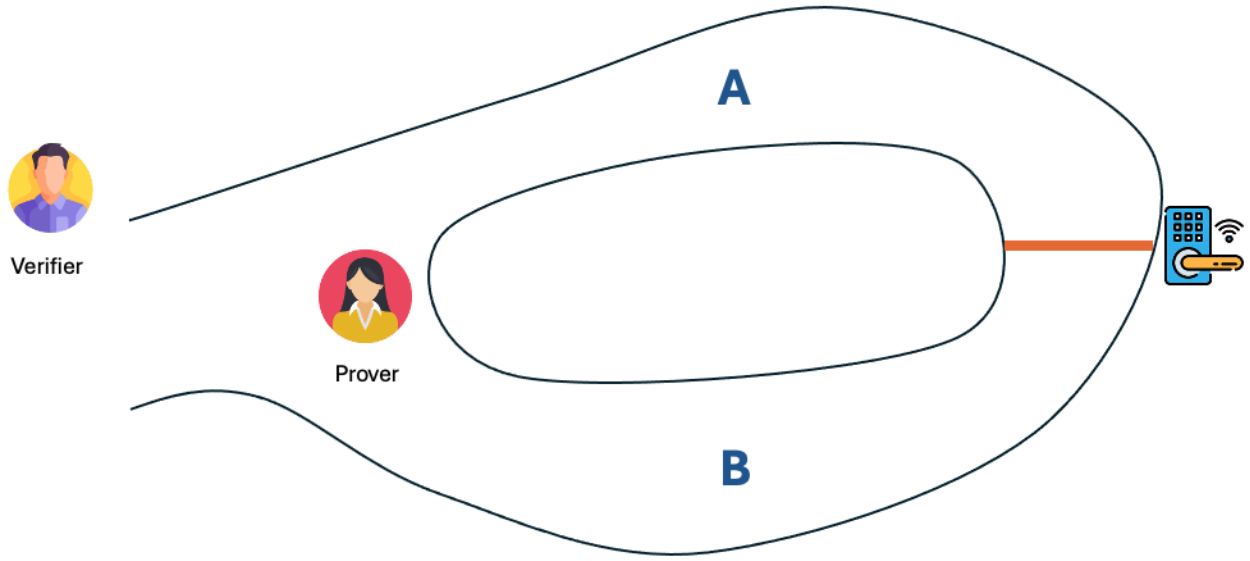


Figure 1: The Ali-Cave. The Prover knows the combination of the lock that is deep in the cave. She wants to prove to the Verifier that she knows the code without disclosing it

A straightforward way to achieve this is to have the Prover and Verifier both randomly choose a path, Figure 2. First, the Prover follows the path to reach the door without the Verifier seeing, and then the Verifier goes to the entrance and shouts his choice. The Prover then has to follow the path that the Verifier called and appear in the corresponding entrance. In case the prover does not know the password, she cannot cross the door and must, therefore, return by the way she took. This path has a 50% probability of coinciding with the path that the Verifier called. If she knows the password however, she can appear on the called path. By repeating the experiment

enough number of times to eliminate the chance of luck, the Prover can prove to the Verifier that she, indeed, knows the password to the door.

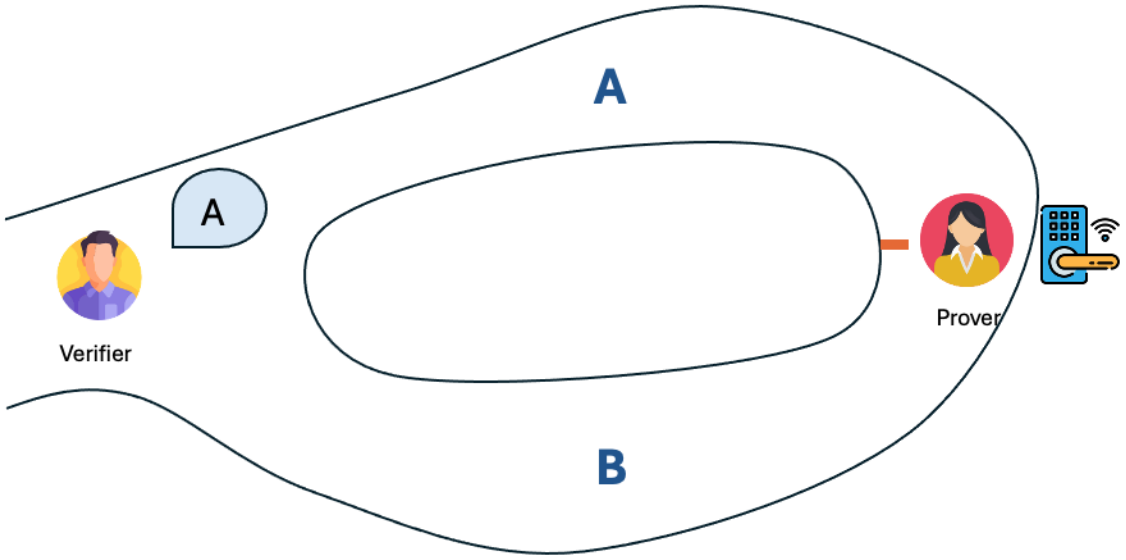


Figure 2: The Prover follows a path and opens the door. The Verifier shouts a random path. The Prover is expected to appear on the entrance corresponding to the path that the Verifier called

Extending the above reasoning to the voting system, it can be seen that the information that is needed for deciding the outcome of a vote is not the individual ballots but rather:

- The aggregates of all the ballots
- The knowledge that a voter has cast a ballot to avoid double voting.

ZKP mechanisms can be applied to prove that a voter has cast a ballot. For computing the aggregates, various schemes based on ZKP exist, such as the "Commitment Scheme". For the purposes of the PRT however, the most promising solution that is now under development is that based on the Homomorphic Encryption. Homomorphic Encryption is a technique that allows operations on encrypted data without the need to decrypt it. The main idea of the mechanism is depicted in Figure 3. A plaintext *m* can be encrypted in the cipher *c(m)*, and consequently, if *f* is any function, the message f(m) is encrypted into the cipher *c(f(m))*. If the encryption is such that by applying *f* to *c(m)* we get the same cipher *c(f(m))* as that we would get if we encrypted *f(m)* directly, then the encryption scheme is a homomorphic encryption that allows computation of function *f* directly on the encrypted data.

As an example of this, consider the Pallier function defined by:

$$C(m) = g^m r^n mod n^2$$

with *g,n* being the public key and *r* a random number.

then

$$C(m_1)C(m_2) = g^{m_1} {r_1}^n mod n^2 \; g^{m_2} {r_2}^n mod n^2 = g^{(m_1+m_2)} (r_1 r_2)^n mod n^2$$
$$= C(m_1 + m_2)$$

It can be seen the cipher contains the sum of the encrypted sum of the messages; this is exactly what is required by a voting system.

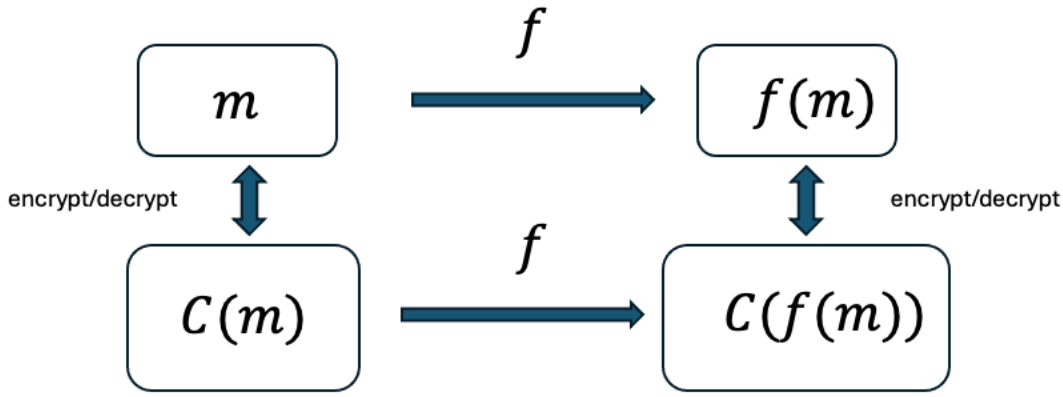


Figure 3: Homomorphic encryption. Applying f directly to c(m) produces the output c(f(m)). Decrypting it, we get the same output as we would get if we applied the function directly in the plaintext data

2.2.2 AI4Gov Implementation of homomorphic encryption

The methodology adopted for the citizen voting system is based on the Paillier cryptosystem (Will et al., 2015), a homomorphic encryption algorithm that ensures secure computations on encrypted data. The Paillier cryptosystem is a probabilistic asymmetric encryption algorithm (public, private key) that relies on the difficulty of computing discrete logarithms in a composite modulus. A key feature of this cryptosystem is its additive homomorphism, allowing encrypted values to be summed without decryption. This property has been leveraged in the voting system to aggregate all encrypted votes securely, producing the final ballot result without exposing individual votes, ensuring privacy and integrity in the election process (for more details see Appendix).

The voting system is based on citizen wallet where the citizens store their identity. The wallet utilizes a verifiable credential solution in order to enable a decentralized Identity management system. Citizens can present proofs in order to be verified without disclosing their identity details. After the verification, citizen can vote on a selected policy. Citizen wallet has access to the public key via blockchain. Therefore, the user cast a vote request as shown below (Table 3).

Table 3: Vote Policy request

Endpoint URL	/api/transactions/VotePolicy
HTTP Method	POST

URL Parameters	policyName (e.g. Waste Reduction)
	credential_id (e.g. a5fb980e-d342-4d3f-a64b-67f79f2f44f79)
	vote (e.g. 1)

Response Example	{
	"response": {
	"message": "Successfully submit a transaction",
	"result": "Policy Voted"
	},
	"timestamp": "2025-03-21T15:11:36.680Z"
	}

The citizen has three voting options, Upvote (1) Downvote (-1) and Neutral (0), Figure 4. The request promotes the preferred vote of citizen regarding the selected policy. The encrypted vote is stored on the blockchain, on condition that the citizen has not voted on that policy before and a successful message is received as a response.

Figure 4: Policy voting choices

The wallet encrypts the vote (m) with the public key using the Paillier cryptosystem before submission to the blockchain gateway. The encrypted vote C(m) is received in the gateway and if the citizen has not voted the policy, yet the C(m) is stored in the blockchain. The encrypted Vote is stored in the blockchain with below format, Figure 5.

```
key: "Policy63453de12d0a2ed231abcd00b7f00046030f4f7baa8cadf2271618496453665ce4e5706536d625d32303d5
  b8453ea0dec6bb08d99a8dd3c0d97630e3ee24f219"
is_delete: false
value: "{"PolicyID":"63453de12d0a2ed231abcd00b7f00046030f4f7baa8cadf2271618496453665ce4e5706536d625d3
  2303d5b8453ea0dec6bb08d99a8dd3c0d97630e3ee24f219","PolicyName":"waste management","KPIs":["GreenCit
  y","Reduce city taxes","Increase recycling","Waste Collection Efficiency","Waste Reduction"],"Category":"Waste Ma
  nagement","Votes":{"a5fb980e-d342-4d3f-a64b-67f79f2f44f79":"3708507436285716497233918903191783891 2\n1
  6123977029830541523807925450408061483490513852982706186127581417495096278773580511576378415 7
  2359447196827338382476910344732491253734585327853112360008375385485427437169001230838971150 7
  4138647573066459398420555350570720149629973858140496817000003791266412860101382766226810934 9
  3041244170305720645901070518842456379214272565664684753911025560176256308718405623519816332 5
  6289793261057900134198329714764792589223401530160784630044287583025401218509503901091784919 7
  9517355711266649895217865346344575154373605870861535218858022591422511657993455191764645045 7
  1795927717849566911500552392886956970216147712410552954279224139945704359608520551571624381 5
  0655875664340316792120253278623488917028890347105563721790372560370963690556603467160048078 6
  3522617043455569173591513326189781411309692180051643047725090795937834859641552143932170362 7
  3415987099698807575952250739548656467015573204065175131800259062093391613577224411744442294 1
  5723138908807974469937397738879820162739543722509668183267807116324206247906213297383476893 7
  3816600303701510713406221648529922640826252953261188812471977827874163933053769766304381547 4
  5444815684019902458577263812072083372648475018178091685197033525633847679967143101251738 5
  0"},"CreatedBy":"VVV","CreatedAt":"Wednesday, 26-Mar-25 16:21:51 UTC","Endorsers":[],"Endorsed":false,"Statu
  s":"","Tag":""}"
```


Figure 5: Citizens' Votes as stored in Blockchain

The multiple encrypted votes can be retrieved from the gateway so as to apply a homomorphically aggregated (f operation) without decryption C(f(m)). The result of C(f(m)) can be decrypted and get the final result of the ballot f(m), Figure 6. The related users can have access to ballot results every time they select to see the result for a specific policy. Users can access updated results dynamically, as the system recalculates the tally whenever a new vote is cast.

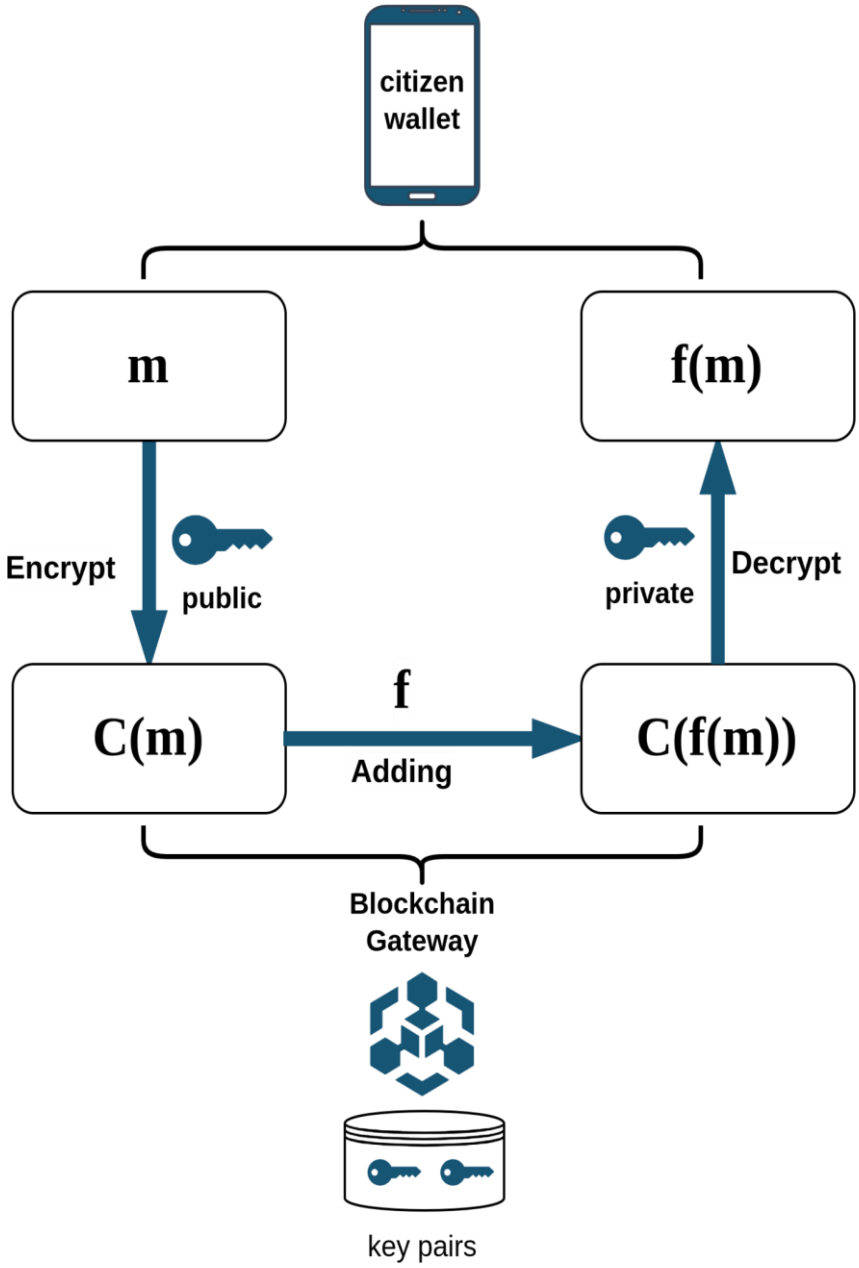


Figure 6: Homomorphic encryption implementation on AI4Gov

2.3 AI In Policy Making

As the number of datasets that correspond a) to the underlying domain(s) of the policy under consideration and b) to the number of opinions formed in public discourse keeps growing, processing of this information via AI, both traditional and generative can lead to recommendations of new policies that benefit from insight gained by these data, that is hard to get via traditional means. AI, in this sense, can lead to breakthroughs in policy-making. However, some caveats can identified:

- The datasets may be poisoned by errors and/or bias.
- In the current state of social media, many of the data points have themselves been generated using generative AI; these data points may too poison the AI models.
- Citizens and representatives cannot verify the source or the validity of the data; even worse, they cannot identify if the AI has been trained on such poisoned datasets and if, thus, can be trusted.

As with open democracy, decentralization can help in enforcing trust in AI models by demanding that:

- Each AI is assigned a decentralized identifier (DID), which is attached to any report it generates
- Any AI derived result or report is anchored in the blockchain together with the metadata of the AI that produced it, so that stakeholders can audit the AI to retrieve information such as its algorithm, its training parameters, its learning corpus etc.

In addition, these caveats are addressed by complying whit GDPR and the recently passed AI Act. As an extra layer of trust, AI that produces deterministic results can run on-chain as a smart contract. The execution of an AI that is implemented as smart contract is fully reproducible by any peer; in this sense stakeholders can validate the results independently. The requirements that codify the above considerations can be viewed in Table 4.

Table 4: AI-enabled Policy Making in AI4Gov

Use Case	Pilot	Data	Users	Requirements
Water Management – drinking water	DPB	-Sewage Treatment data -Water cycling billing data	Policy Makers	-Set criteria -Get recommended policies -Generate new policies

Water Management – sewage water			Citizens	-Audit AI -Get Explainability report
Parking tickets monitoring	VVV	-Census data -Household water data -Tourist data (arrivals, overnight stay, cruise data) -Airport traffic data -Municipality events attendance data	-Policy makers	-Set criteria -Get recommended policies -Generate new policies
Waste management – Pay as you Throw			Citizens	-Audit AI -Get Explainability report
Water Management policies	DBP -JSI	Data from DBP and JSI	-Policy makers	Get relevant reports
Waste Management policies	VVV -JSI	Data from DBP and JSI	-Policy makers -Citizens	Get relevant reports

3 Architecture

Following the requirements laid out in Section 2, the architecture of the Policy Recommendation Toolkit can be derived. This architecture describes the toolkit to a level of granularity that is at a lower level than the one described in the AI4Gov Reference architecture that was described in D2.4 (Figure 7). The red boxes indicate the backend of the PRT, while the green box corresponds to its backend. Elements of the PRT overlap with the Decentralized Data Framework, which was described in D3.1 and D3.2, while the front end is tightly integrated with the Visualization Workbench. As such, certain elements of the architecture of the PRT are referred to elements described in D3.1 and D3.2.

For a uniform presentation, the same approach that was followed in D3.1 and D3.2 to describe the architecture will be followed here. The Archimate modeling language (Archi – Open Source ArchiMate Modelling) of The Open Group will be followed and the architecture will be described in the Business and Application Layer; as there is a strong semantic component to the PRT, a special Semantic View will also be given. As is the case with the Decentralized Data Framework, further decompositions of the architecture elements, along with its technical layers, will be given in this 2nd iteration of the deliverable.

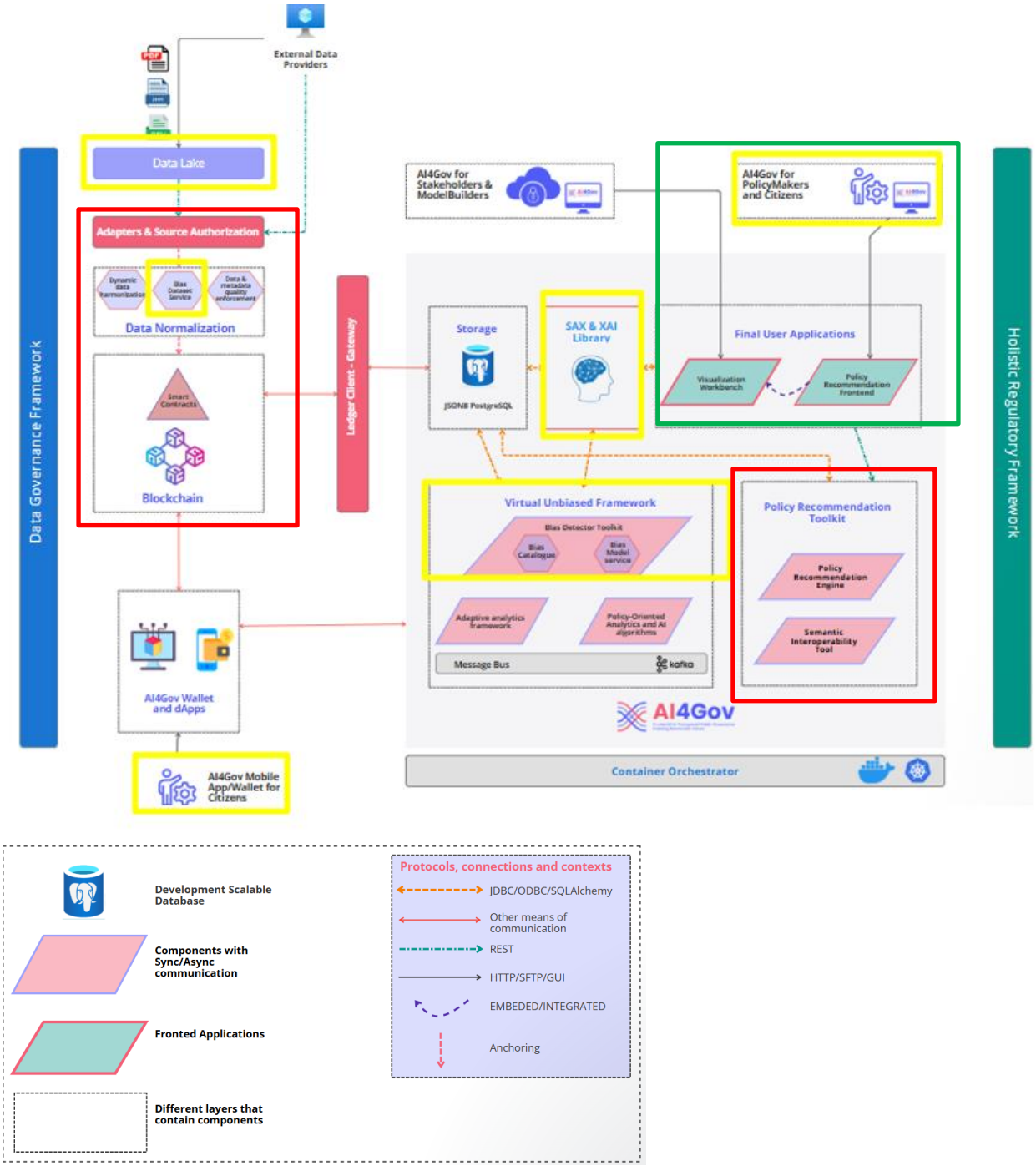


Figure 7: AI4Gov Reference Architecture

3.1 Business Layer

From a business perspective, the main value of the PRT is to facilitate policy making. However, as it was already seen by the requirement analysis, it does so by incorporating various

interconnected functionalities and affects the various stakeholders in a different way. To this end, a high-level business viewpoint will be given, that will be accompanied by the different views that describes how these different aspects.

The high level viewpoint is presented in Figure 8. A circular value stream provides clear and unambiguously defined policies that are enhanced by the engagement of the stakeholders. The stakeholders co-create the policies by using a platform that promotes trust. The policies are published and are then governed and audited via the open platform. The business services for materializing this value stream are:

- The Semantic Alignment service is leveraged to produce policies with clear semantics.
- The Open Democracy DAO implements all functionality that allows stakeholders to participate and govern the policy-making process in a democratic and decentralized manner using the underlying blockchain infrastructure.
- The AI Recommendation service uses AI to recommend policies based on criteria set either by the policy makers or by the democratic process agreed upon in the DAO service.

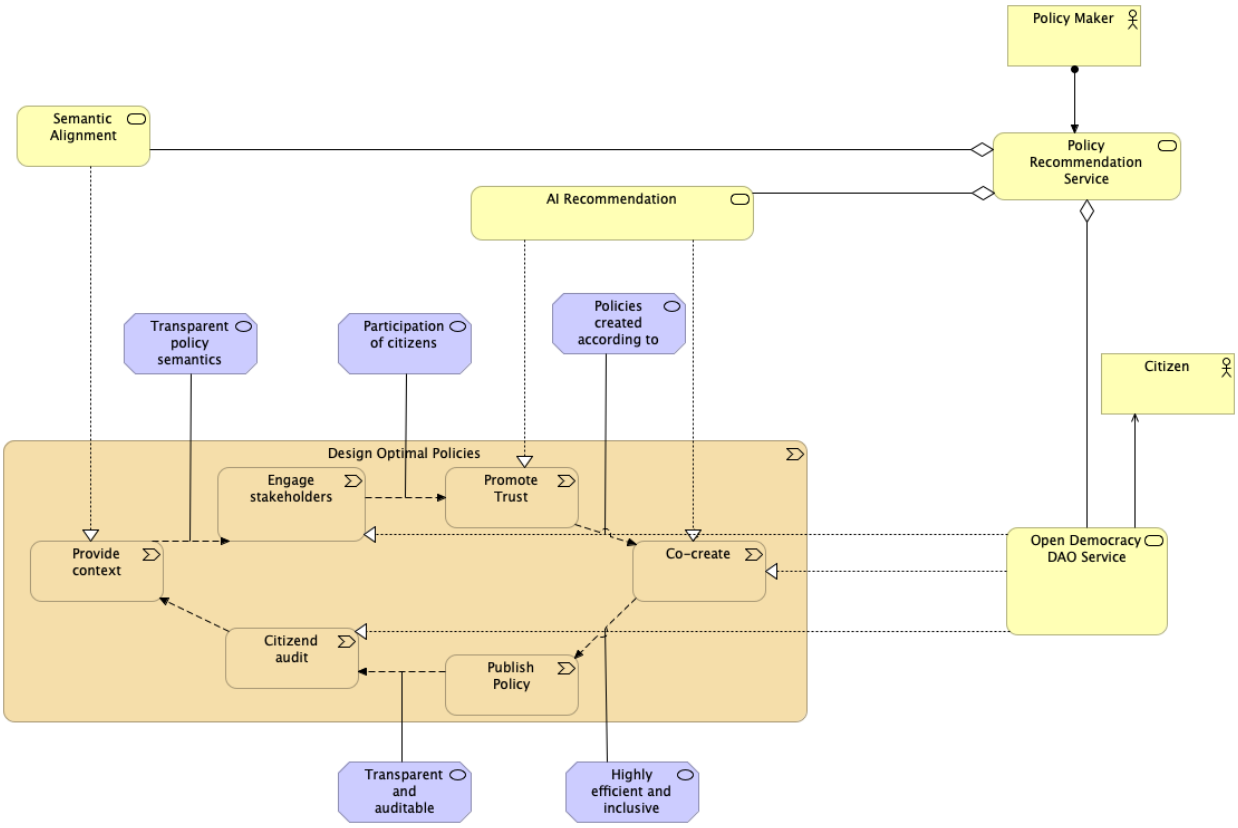


Figure 8: PRT Architecture – High-level view

3.1.1 Semantic Alignment View

Figure 9 depicts the Semantic Alignment View that shows how the Semantic Alignment service is decomposed. It is realized by an underlying business process, which is served by two sub-services. The Data Uplift Service performs the translation of data headers, relations, and metadata from the source format to the common vocabulary, while the Data Alignment Service maps data between data sets using common ontologies.

These services are in turn realized by internal business processes, which are composed respectively by internal business functions, which define that uplift and alignment should be performed by using discovery services that are based on AI.

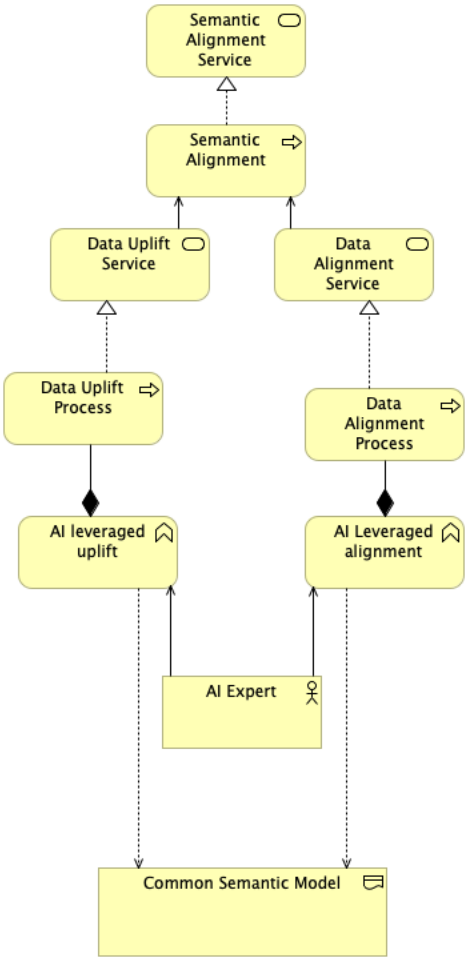


Figure 9: Semantic Alignment View

3.1.2 Open Democracy View

The Open Democracy DAO is a Decentralized Autonomous Organization[1] realized by the elements depicted in the Open Democracy View in Figure 10. The Service is realized by three processes that

[1] https://www.investopedia.com/tech/what-dao/

cover all main aspects of the service, mainly voting, consulting (opinion forming) and auditing. The functions that implement this view are all based on business logic implemented in smart contracts using the underlying Decentralized Data Governance infrastructure.

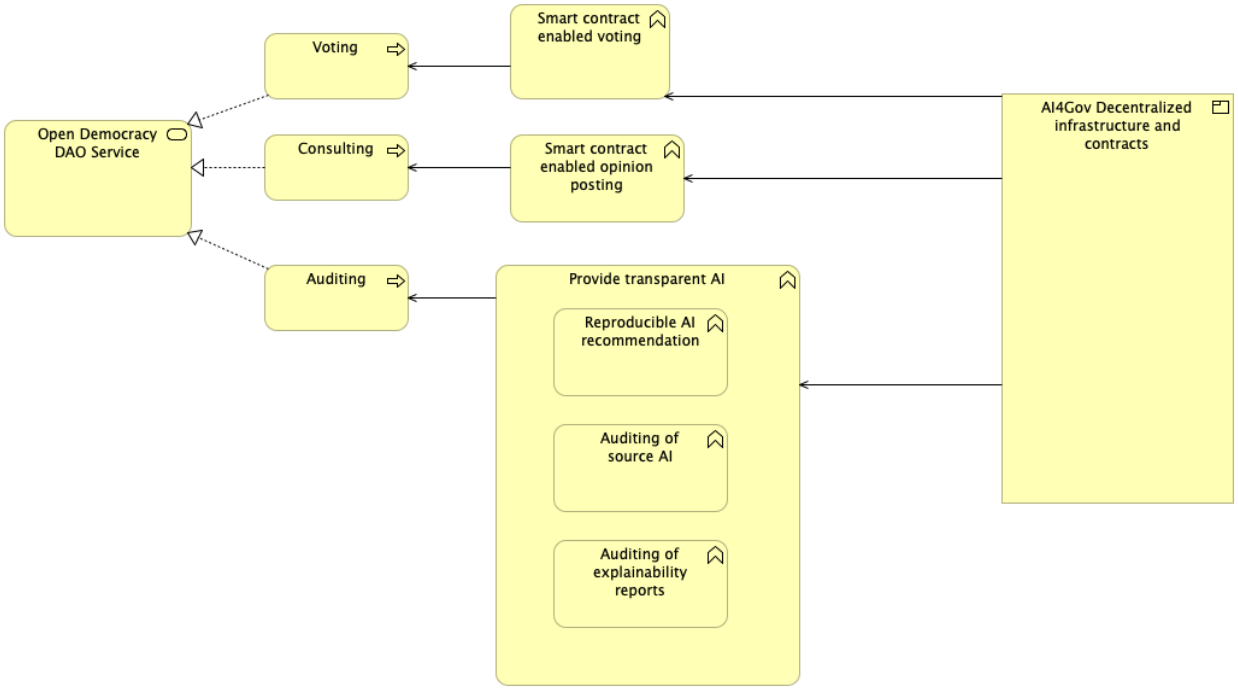


Figure 10: Open Democracy View

3.1.3 AI-based Policy Recommendation

The AI-based Policy Recommendation View is depicted in Figure 11. The realization of the AI Recommendation Service takes place via two functions, mainly the one that is responsible for the recommendation itself and the one that is responsible for anchoring any AI and explaining ability reports that happened off-chain to the policy that was recommended based on this AI. The on-chain AI analytics function is served by the AI4Gov Decentralized Infrastructure and Contracts (defined in D3.1 and D3.2). External AI Services that are to be implemented in WP4 can be anchored in the blockchain following the processes of data anchoring defined in the Decentralized Data Governance Framework and, in turn, served to the AI Recommendation Service via the On-chain AI analytics function.

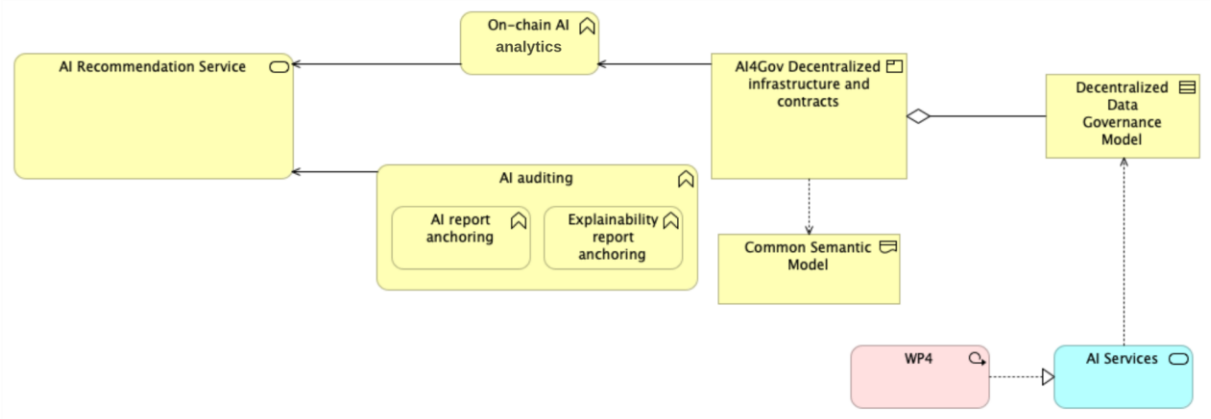


Figure 11: AI Recommendation Service

The on-chain AI analytics function is about to recommend some KPIs based on analytics which are getting as response from AI services. The recommended KPIs are appended to the policy which is written on-chain.

3.2 Semantic Layer

In order not to be confused with the Semantic Alignment View, which was described in 3.1.1, the Semantic Layer depicts the organization of data and information, whereas the Semantic Alignment View describes the process of performing semantic alignment. This layer is depicted in Figure 12. Sources of data include:

1. AI and explainability reports that are generated by external AI services and are linked into the blockchain via anchors. The anchoring business function describes this process and is used to serve the recommendation system based on AI.
2. The source and the aligned data come from data that are retrieved on-site; the first one corresponds to raw data as these are collected by pilots, while the latter one corresponds to data that have been uplifted by the semantic models.
3. Finally, the policy is a serialized document that describes policies and is stored on-chain.

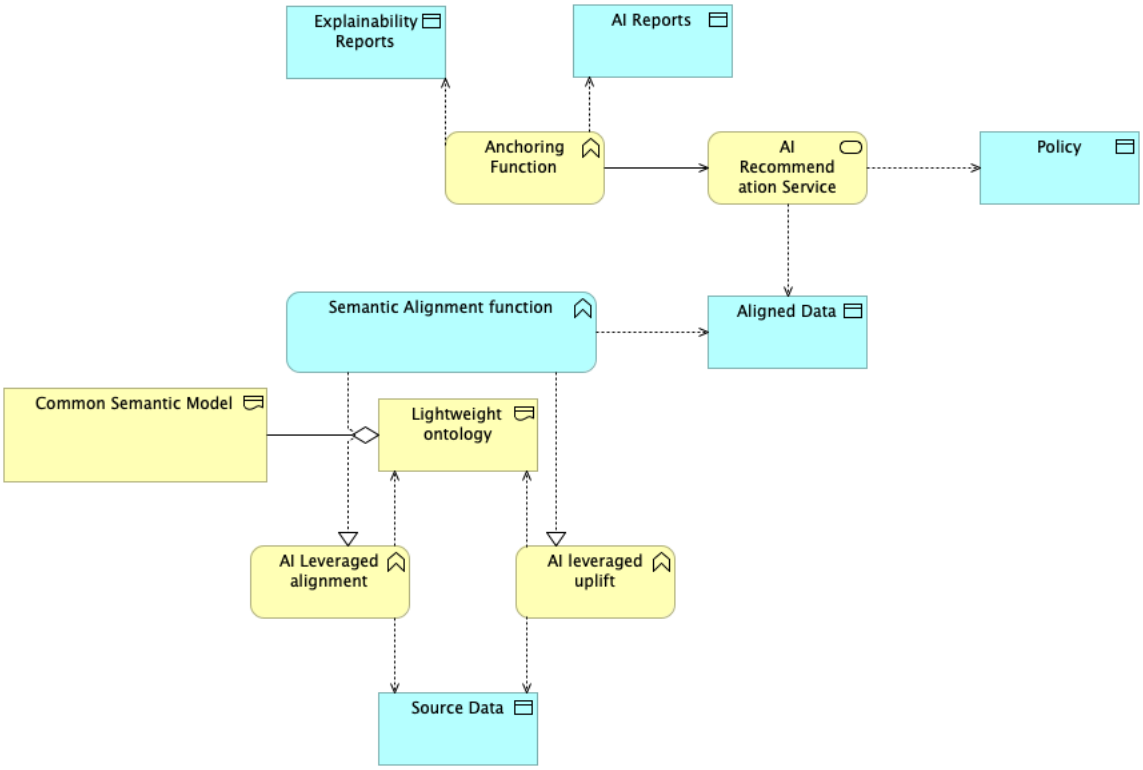


Figure 12: Semantic Layer View

3.3 Application Layer

The high-level application layer of the PRT is depicted in Figure 13: PRT Architecture – Application LayerFigure 13. The toolkit is a component that is assigned to two services: the DAO service, which implements the Open Democracy functionality, and the AI Recommendation Service, which implements On Chain AI and mapping of policies to AI reports via blockchain anchoring. For completeness, all relevant business services that are realized by the corresponding application services are also listed in the diagram. The Semantic alignment is not directly assigned to the component, but rather serves other functionalities of the PRT, namely the recommendation service itself. The role of the main application services is to serve the core business functions that compose the PRT, such as the auditing/voting systems and the recommender system.

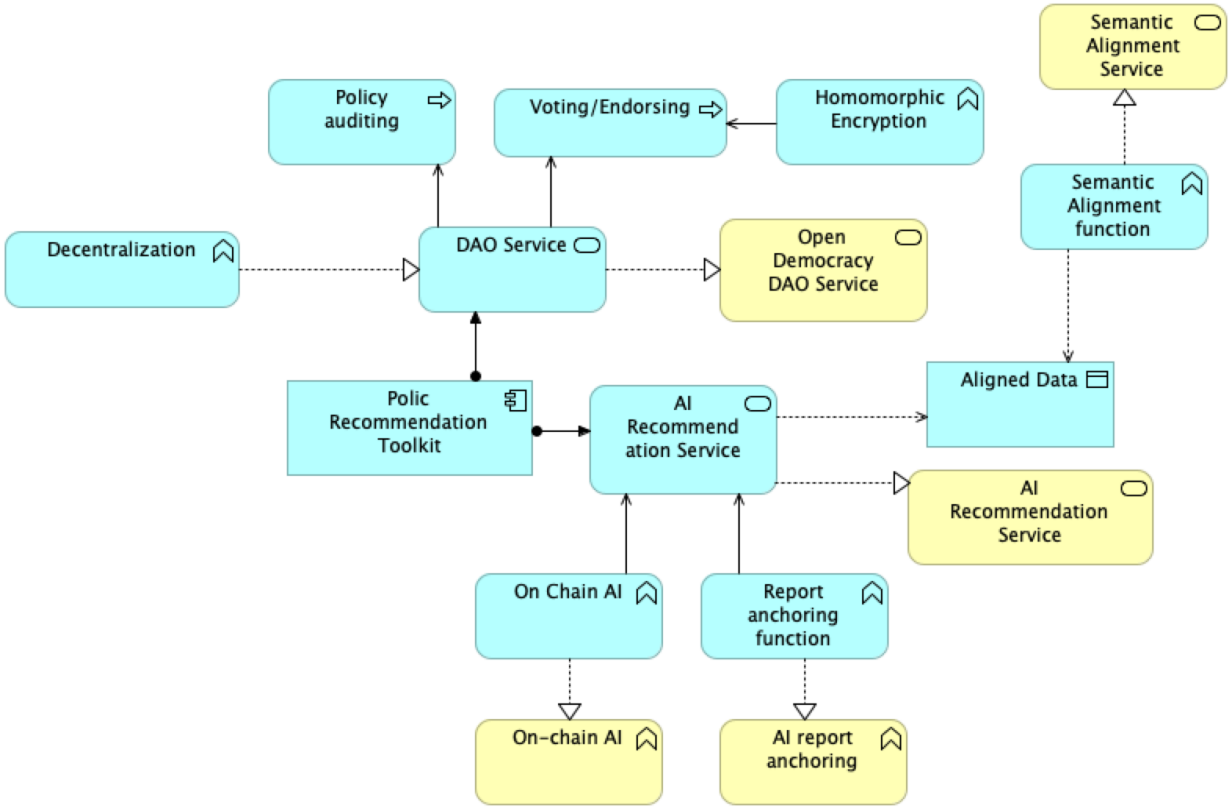


Figure 13: PRT Architecture – Application Layer

3.4 Technology Layer

The Technology Layer (or Technology View) provides a structured representation of the hardware, software, and networking infrastructure that supports the application processes, Figure 14. We have two Hyperledger Aries instances: one that creates the artifact for the Citizen Wallet and another that creates the Next.js artifact, i.e., the Visualization Workbench. The critical difference between them is the decoupled Aries Controller, which is separately deployed on its own server.

Additionally, the Mediator Component ensures that the appropriate data supporting the Verifiable Credentials workflow is written to the Hyperledger Indy blockchain solution. The BIE Infrastructure, along with the two Hyperledger Aries instances, forms the Shared Blockchain Interface.

The APK and Next.js artifacts both realize the Policy Recommendation Toolkit (PRT), while the Shared Blockchain Interface serves the PRT. The Node.js server inside the BIE Infrastructure realizes the BIE Service, whereas the Hyperledger Aries nodes realize the Aries Service. Together, they support the AI Recommendation and DAO Services of the Application Layer.

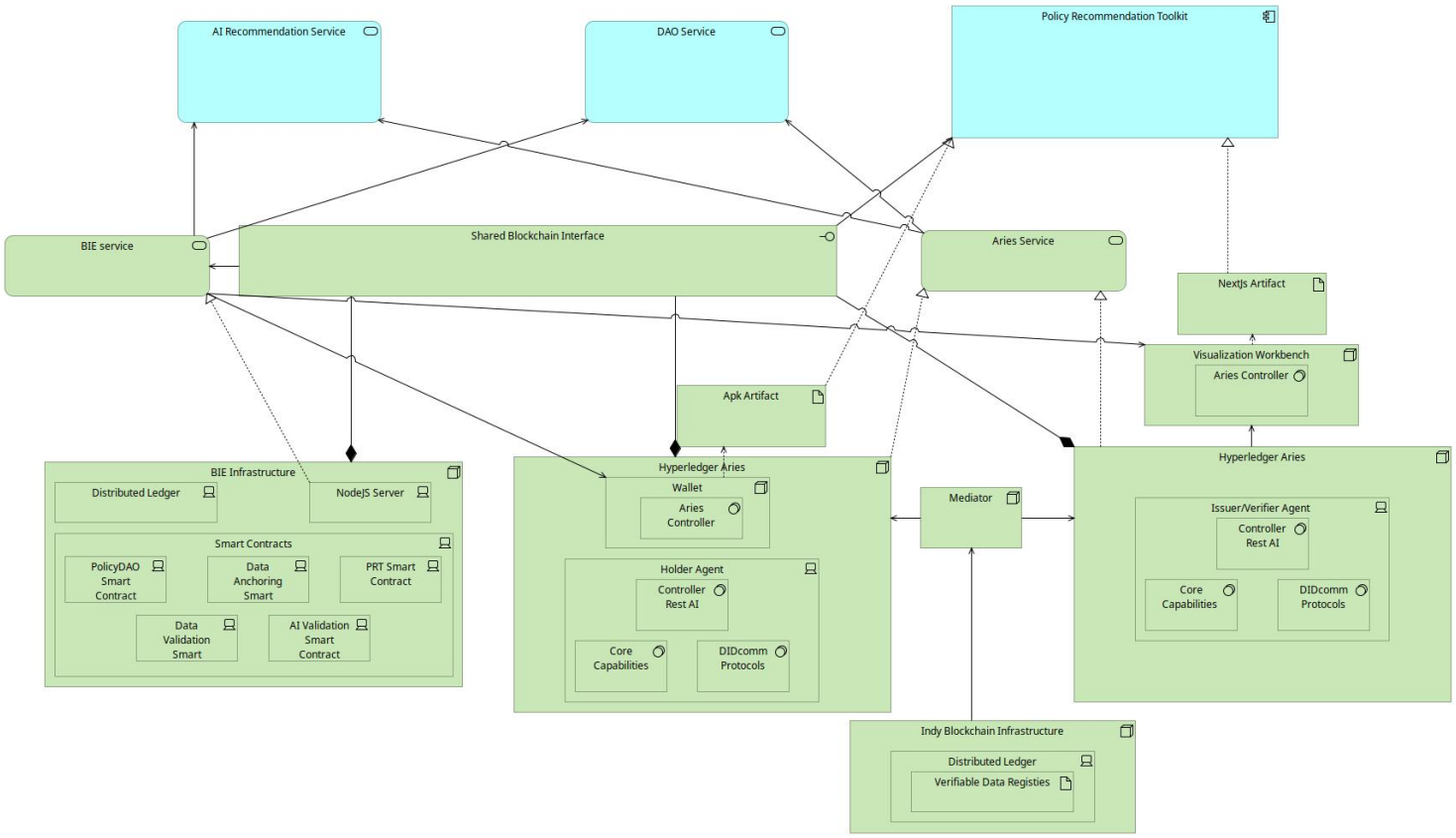


Figure 14: PRT Architecture – Technology Layer

4 Prototype Development

The implementation of the PRT was following an iterative process by which a continuous requirements engineering process is followed in parallel with pilot activities (WP6) to constantly update the requirements and the software releases following suit to implement a subset of these requirements in each cycle.

In parallel with the implementation of PRT, integration activities have also been performed; both the decentralized infrastructure and the PRT code base are integrated into the Visualization Workbench as far as the desktop elements are concerned. Apart from that, based on the comments of the first review in order to include the new case of open democracy we have implemented a decentralized mobile application (dApp) for citizens. This wallet borrows design elements from the Visualization Workbench in order to create a unified user experience. Furthermore, an AI-enabled toolkit for open and inclusive policy-making involving public authorities and citizens has been implemented.

The overall status of the implementation in relation to all the functionalities that the PRT introduces can be summarized as follows:

- The mechanism for aligning KPIs into a lightweight semantic model has been implemented. The model was developed based on KPIs from the VVV and DPB pilots and is used for prototyping purposes.
- Insertion and updating of policies have been implemented as a smart contract using the underlying decentralized infrastructure that is described in D3.1 and D3.2.
- Mechanisms for voting policies have also been implemented. Voting can be performed by both institutions and citizens via their dApps.
- Smart contracts for automatically filtering out popular policies have been implemented. These smart contracts identify popular policies that are endorsed by peers (e.g., other municipalities) or by peers upon conditions set into the smart contract.
- An on-chain recommendation system has been implemented as a smart contract as a prototype of on-chain AI recommendation algorithms. The prototypical recommendation system accepts a set of hard and soft KPIs and returns all policies that fulfill the hard KPIs set by the user. The policies are then ranked according to their performance on the soft KPIs.
- The inclusion of taxonomies and vocabularies into the Decentralized Governance Model and the utilization of those taxonomies to define action items and KPIs of policies.
- The implementation of mechanisms for defining custom blockchain governance models and deploying DAOs on the blockchain.
- The implementation of a mechanism for automated smart contract definition and deployment. This task is common with activities carried out in T3.1; in the context of T3.3 the focus was on applications that allow users to set the basic rules and meta-parameters of a recommender smart contract (e.g., threshold values for accepting policies, sequence of conditions etc.) and govern how this updated business logic is deployed in the blockchain.

- The implementation of fully autonomous citizen dApps has been packaged and deployed on Android devices. This Android application allows citizens to monitor policies and participate in voting and giving feedback on them.

4.1 Policy Recommendation Toolkit (PRT)

4.1.1 PRT Overview

The Policy Recommendation Toolkit (PRT) and its functionalities are integrated in to Visualization Workbench (more details in D4.4) which is publicly available in the Ai4Gov cluster (https://cluster-ai4gov.euprojects.net/). The PRT includes interactive graphical components for providing each one of the major functionalities that fulfil the framework's objectives. There is more than one navigation option among the available tools through the application's overview screen, which are available to the policy maker; this can be chosen via the initial page (Figure 15 and Figure 16). As depicted in Figure 15, the four main functionalities are described, allowing the user to navigate to one of them. On the same screen, there are four categories in which the policy maker can create a new policy, as depicted in Figure 16.

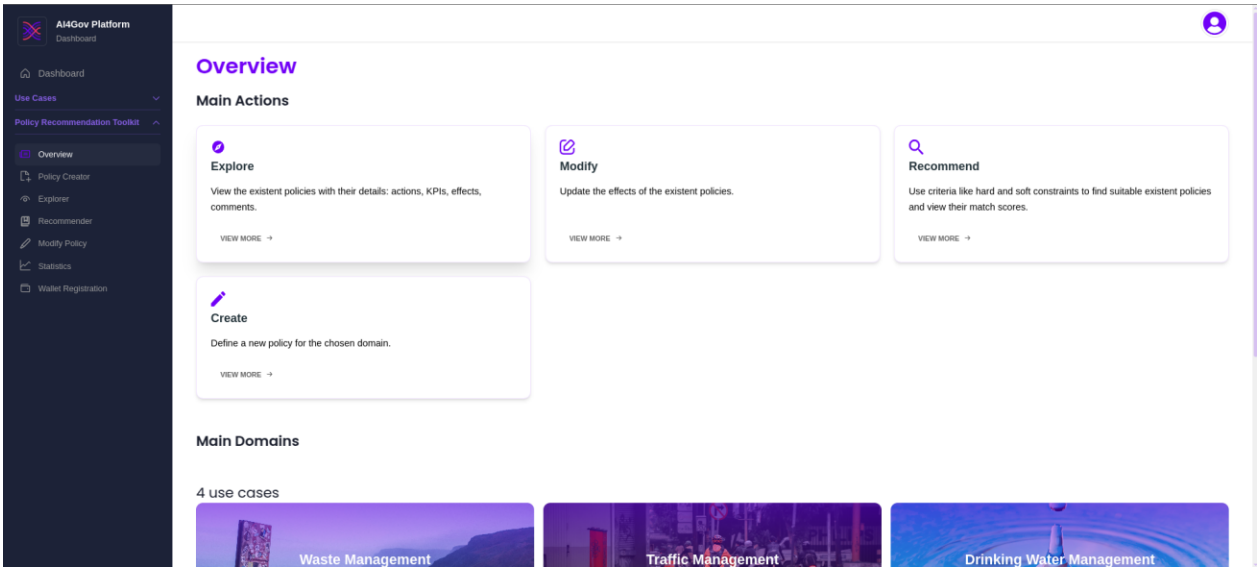


Figure 15: Homepage of the Policy Recommendation Toolkit (top view with functionalities)

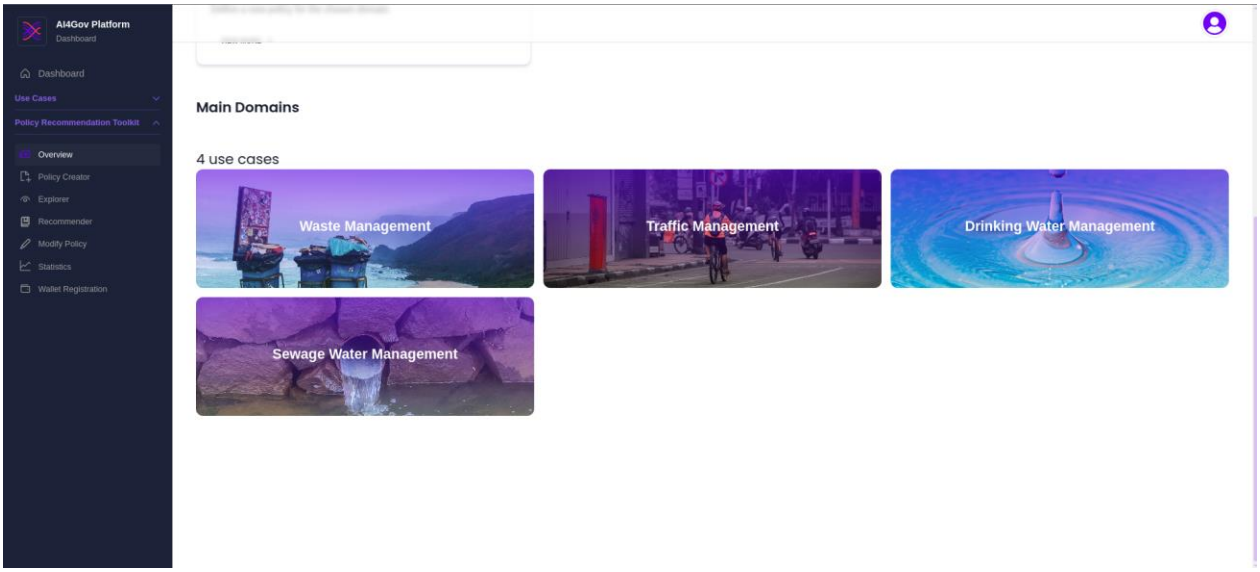


Figure 16: Homepage of the Policy Recommendation Toolkit (bottom view with the categories)

4.1.2 PRT Policy Creator

Then the user can create a policy from the relative menu, selecting a specific category (waste management, traffic management, drinking water management, sewage water management). Following the process of introducing a new policy for each of the existing categories, as will be analysed in detail below. Initially, for the first category of waste management, the user is required to complete the following form, as depicted in Figure 17.

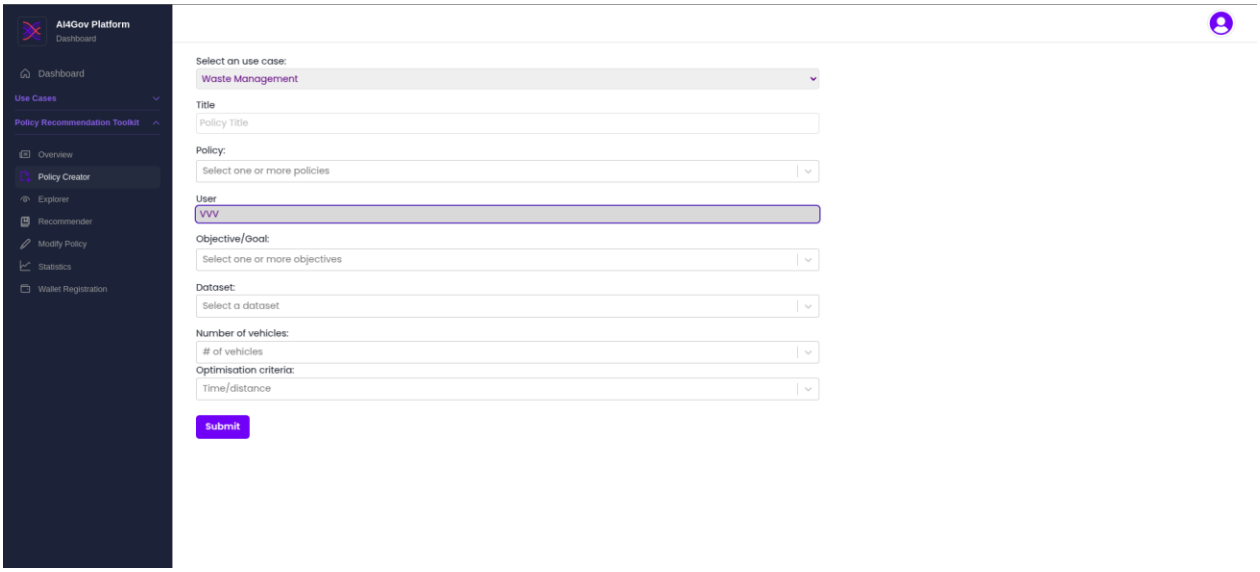


Figure 17: Policy Creator interface

In this specific form, the policy creator is required to fill in the following details. First, the title of the policy is entered in the first field. In the second field, labelled 'policy,' one or more policies related to the specific category are selected. The third field, 'user,' contains a fixed value with the user's name and cannot be modified.

Next, in the 'objective/Goal' field, one or more objectives expected to be achieved with this policy are entered. The following field is related to the dataset of the AI models that will be used for this specific category. In the last two fields, the number of available vehicles in the municipality that are ready to be used is entered, followed by the criteria that the policy maker wants to optimize. In this example, the criteria include time and distance.

The final form, once completed, depicted in Figure 18.

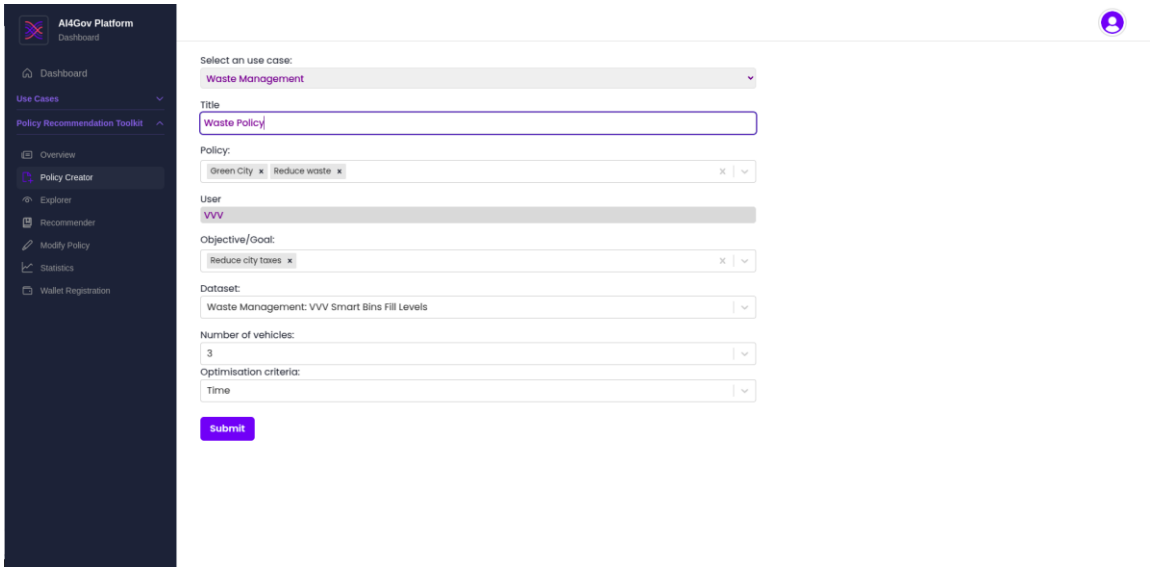


Figure 18: Fill form with the appropriate values in Waste Management category

When the user presses the 'Submit' button, certain processes start running in the background, Figure 19.

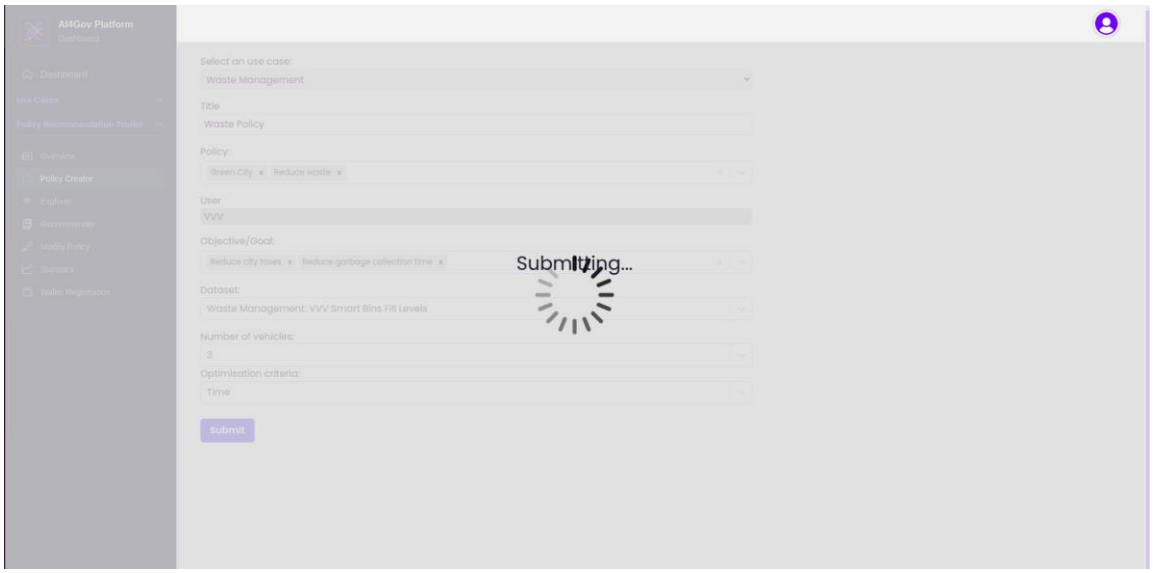


Figure 19: Policy creation submitting process screen

First, all the information entered in the form is collected and properly prepared to be sent as a request to the analytics. This is the body that will be incorporated into the API request, Table 5.

Table 5: Request to AI analytics for Waste Management Category

Endpoint URL	/ai4gov_routing_optimization_api
HTTP Method	POST

Request Body	<pre>{
	"distance_type":"optimize distance",
	"vehicle_capacities": 5
	}</pre>

Response Example	<pre>{ "total_time": 0,
	"total_distance": 10845,
	"total_load": 3133,
	"total_fuel_cost": 6.42
	}</pre>

Therefore, the user receives feedback in the form of a window displaying all relevant information. Initially, it includes the title of the policy that has been entered, the user who submitted it, the

KPIs generated by the analytics, as well as the predicted values produced by the algorithm. These values include the distance travelled, the time required, the fuel consumed, and the fuel cost incurred, as depicted in Figure 20. The final policy is stored in the blockchain.

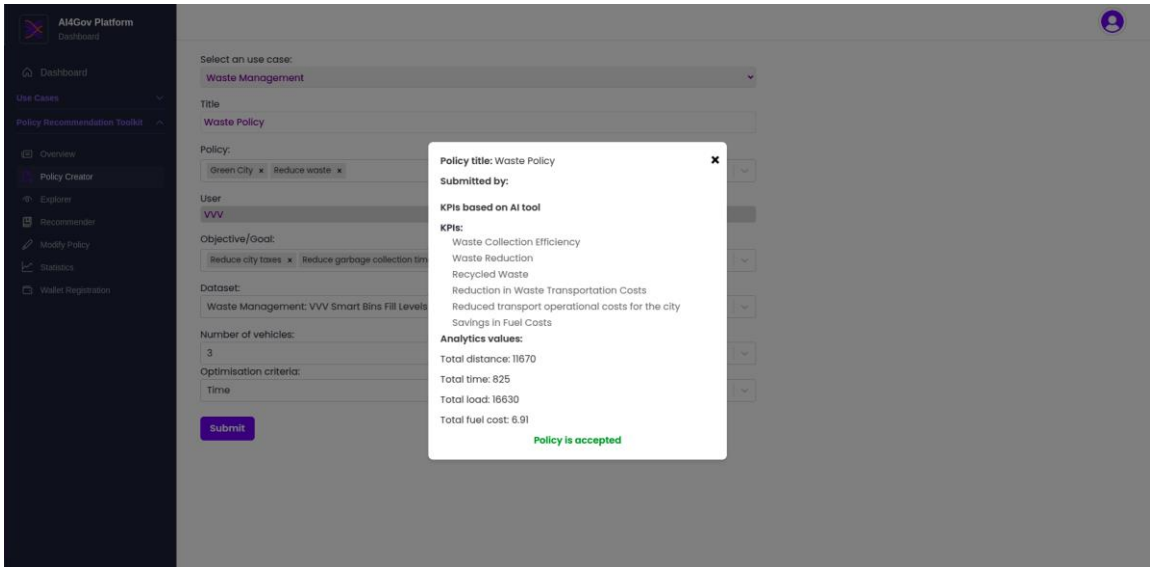


Figure 20: Total policy recommendation and analytics results in Waste Management category

The next category concerns traffic management. This category is selected from the drop-down menu in the first field. In this specific form, the policy creator is required to fill in the following details. First, the title of the policy is entered in the second field. In the third field, labeled 'policy,' one or more policies related to the specific category are selected. The fourth field, 'user,' contains a fixed value with the user's name and cannot be modified.

Next, in the 'objective/Goal' field, one or more objectives expected to be achieved with this policy are entered. The following field is related to the dataset of the AI models that will be used for this specific category. And in the last two fields, the violation for which we want to make a prediction is initially entered, and in the second part, the part of the week we want to examine. In the application, there are two options: the duration of the week or the weekend. The final form, once the form has been completed, as depicted in Figure 21.

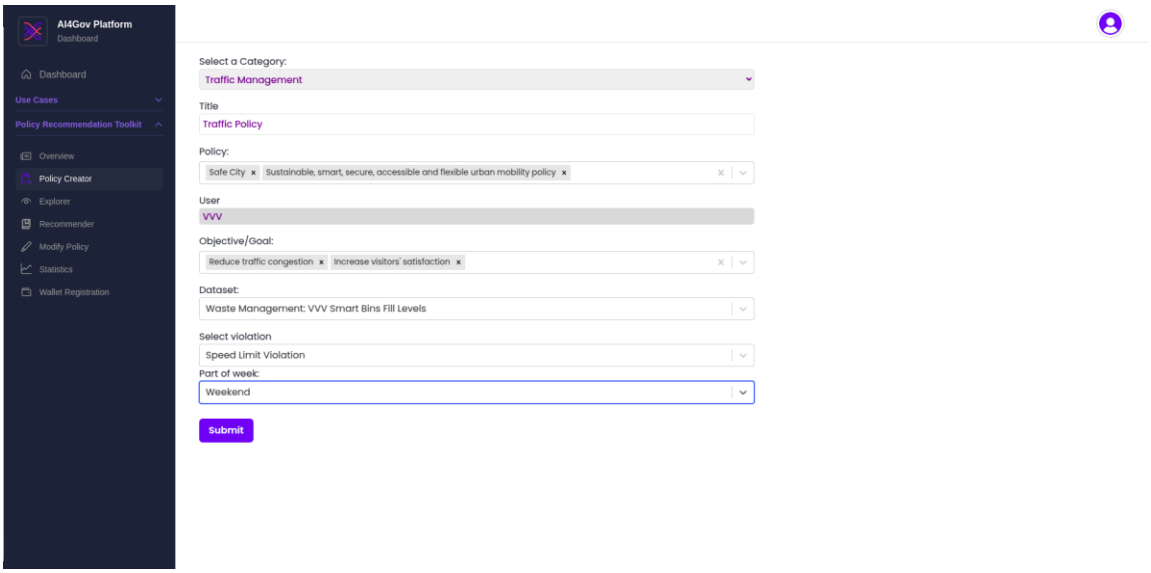


Figure 21: Fill form with the appropriate values in Traffic Management category

When the user presses the 'Submit' button, certain processes start running in the background. This is the body that will be incorporated into the API request, Table 6.

Table 6: Request to AI analytics for Traffic Management Category

Endpoint URL	/ai4gov_prt_traffic_violations_api
HTTP Method	POST

Request Body	{
	"violation":"Speed Limit Violation",
	"part_of_day": "weekday"
	}

Response Example	{ "area": 4, "distance_to_be_covered": 109.0, "fuel_cost": 12, "num_police_cars": 3, "time_needed": 130.8 }

Once the process just described is completed, the user receives feedback in the form of a window displaying all relevant information. Initially, it includes the title of the policy that has been entered, the user who submitted it, the KPIs generated by the analytics, as well as the predicted values produced by the algorithm. These values include the cost of fuel, the distance that needs to be covered, the time required to cover the distance, and the number of police cars, as depicted in the Figure 22.

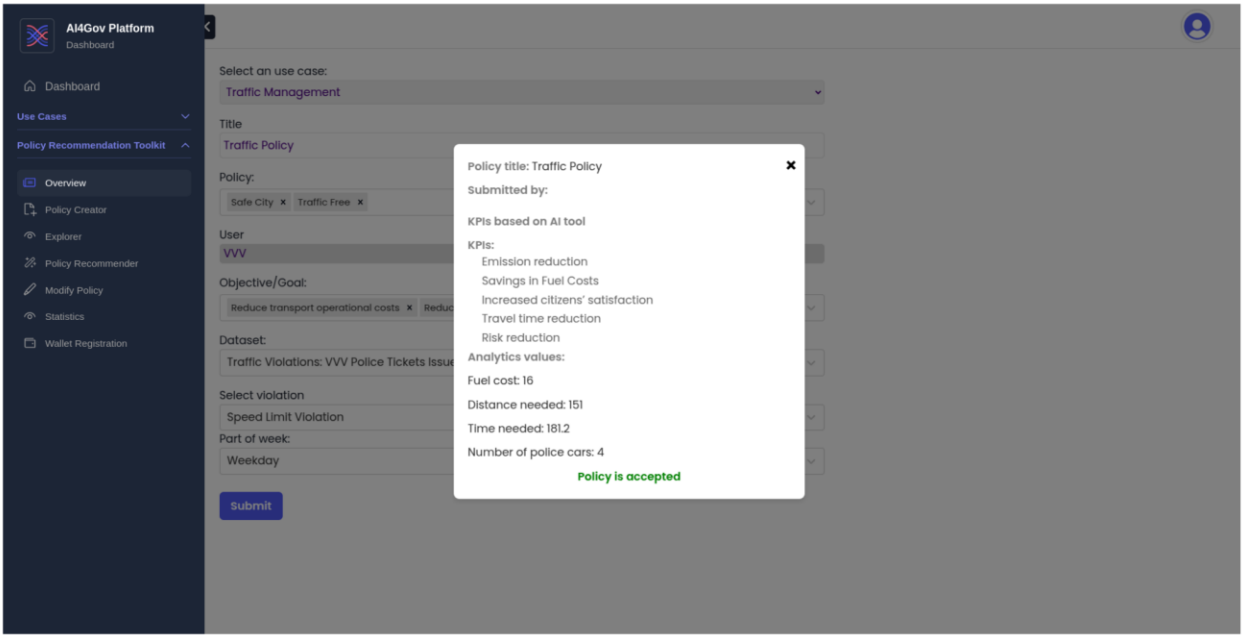


Figure 22: Total policy recommendation and analytics results in Traffic Management category

The next category concerns the quality of drinking water. This category is selected from the drop-down menu in the first field. In this specific form, the policy creator is required to fill in the following details. First, the title of the policy is entered in the second field. In the third field, labeled 'policy,' one or more policies related to the specific category are selected. The fourth field, 'user,' contains a fixed value with the user's name and cannot be modified.

Next, in the 'objective/Goal' field, one or more objectives expected to be achieved with this policy are entered. The following field is related to the dataset of the AI models that will be used for this specific category. In the last field, the user is required to choose between the three options ATALAYA, HIGUERA LA REAL, VALDERE DE LLERENA, which concern the area for which the prediction will be made, as depicted in Figure 23.

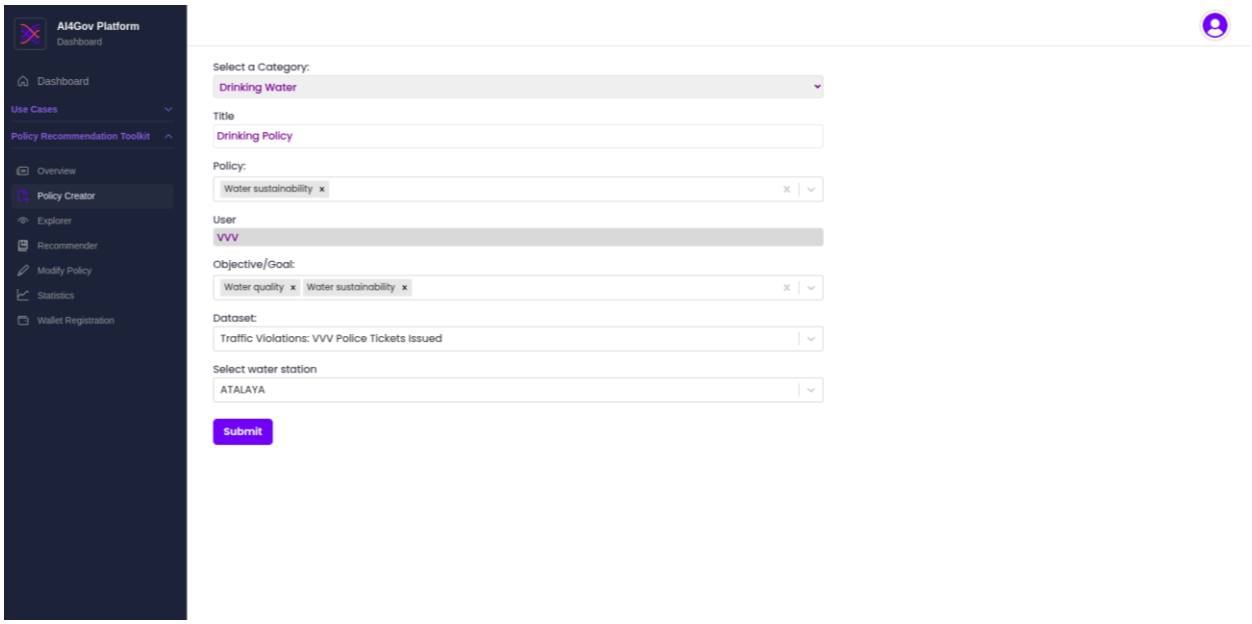


Figure 23: Fill form with the appropriate values in Drinking Water category

When the user presses the 'Submit' button, certain processes start running in the background. A form is created that contains the necessary information that has been entered, as well as an additional CSV file that includes the history of previous reports regarding water quality, Table 7.

Table 7: Request to AI analytics for Drinking Water Category

Endpoint URL	/ai4gov_water_management_api
HTTP Method	POST
URL Parameters	use_case (e.g. drinking water - quality)
	entity (e.g. ATALAYA)

Response Example	{
	"prediction_cl": 0.6213,
	"prediction_level": 90.685,
	"prediction_ph": 8.1142,
	}

Once the process just described is completed, the user receives feedback in the form of a window displaying all relevant information. Initially, it includes the title of the policy that has been entered, the user who submitted it, the KPIs generated by the analytics, as well as the predicted values produced by the algorithm. These values include the prediction of CL, PH and level as well, as depicted in the Figure 24.

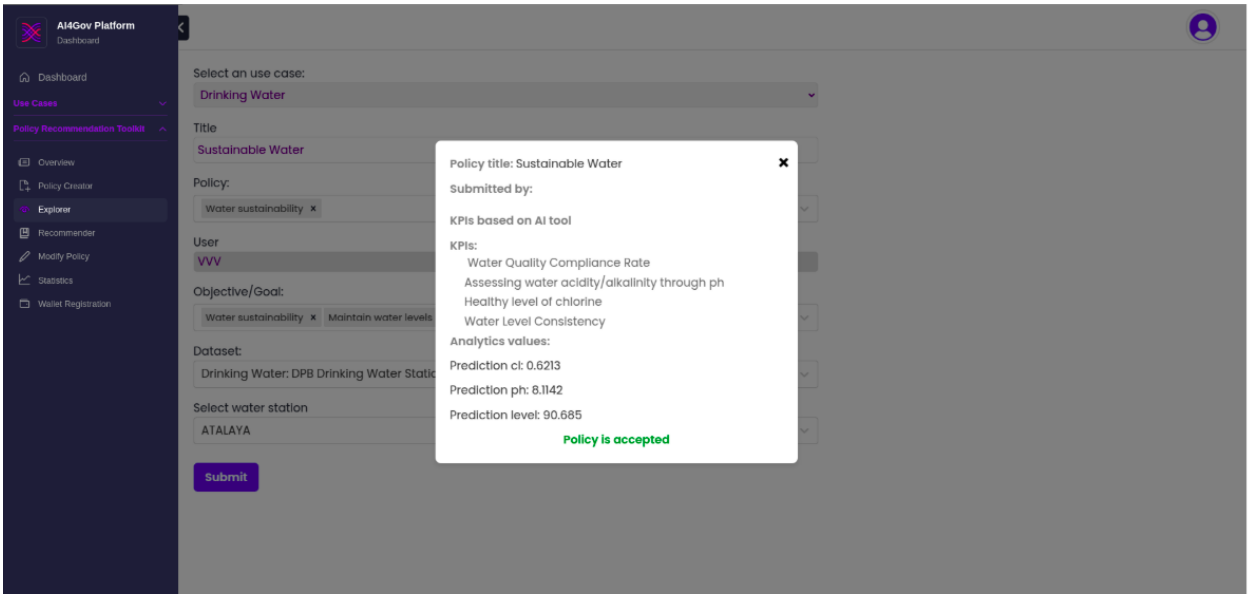


Figure 24: Total policy recommendation and analytics results in Drinking Water Management category

The next and the last category concerns the sewage water. This category is selected from the drop-down menu in the first field. In this specific form, the policy creator is required to fill in the following details. First, the title of the policy is entered in the second field. In the third field, labeled 'policy,' one or more policies related to the specific category are selected. The fourth field, 'user,' contains a fixed value with the user's name and cannot be modified.

Next, in the 'objective/Goal' field, one or more objectives expected to be achieved with this policy are entered. The following field is related to the dataset that will be used for the above. In the last field, the user is required to choose between the three options EDAR Cheles, EDAR Oliva de la Frontera, EDAR Torremayor, which concern the area for which the prediction will be made. As depicted in Figure 25.

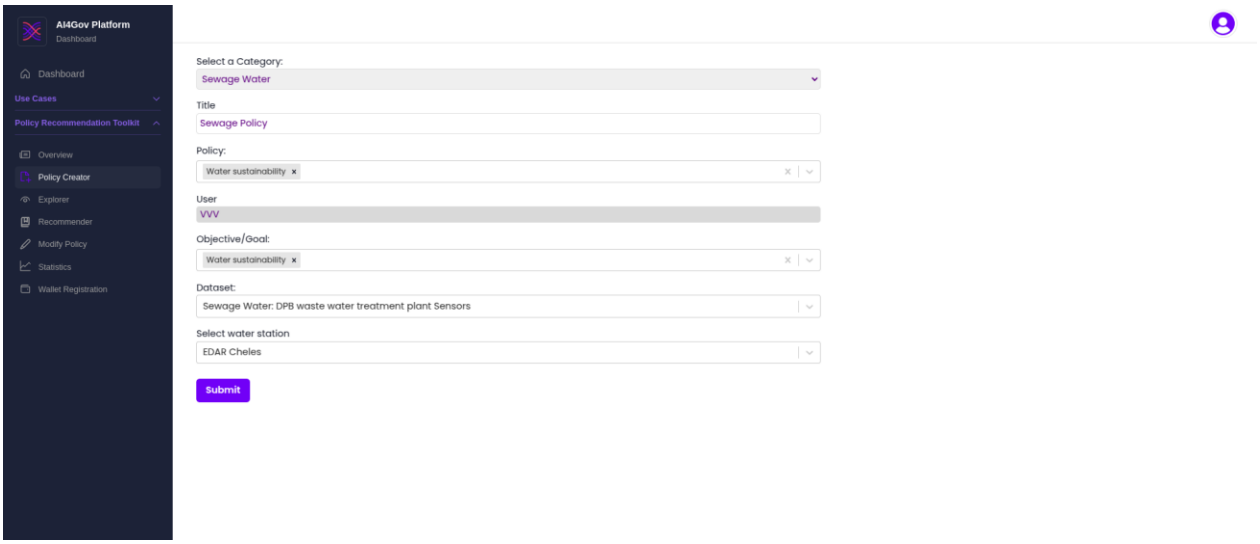


Figure 25: Fill form with the appropriate values in Sewage Water Management category

When the user presses the 'Submit' button, certain processes start running in the background. A form is created that contains the necessary information that has been entered, as well as an additional CSV file that includes the history of previous reports regarding energy consumption, Table 8.

Table 8: Request to AI analytics for Sewage Water Category

Endpoint URL	/ai4gov_water_management_api
HTTP Method	POST
URL Parameters	use_case (e.g. sewage water - WWTP energy consumption)
	entity (e.g. EDAR Cheles)

Response Example	{
	"prediction_cl": 0.6213,
	"prediction_level": 90.685,
	"prediction_ph": 8.1142,
	}

Once the process just described is completed, the user receives feedback in the form of a window displaying all relevant information. Initially, it includes the title of the policy that has been

entered, the user who submitted it, the KPIs generated by the analytics, as well as the predicted values produced by the algorithm. These values include the prediction of energy consumption as well, as depicted in the Figure 26.

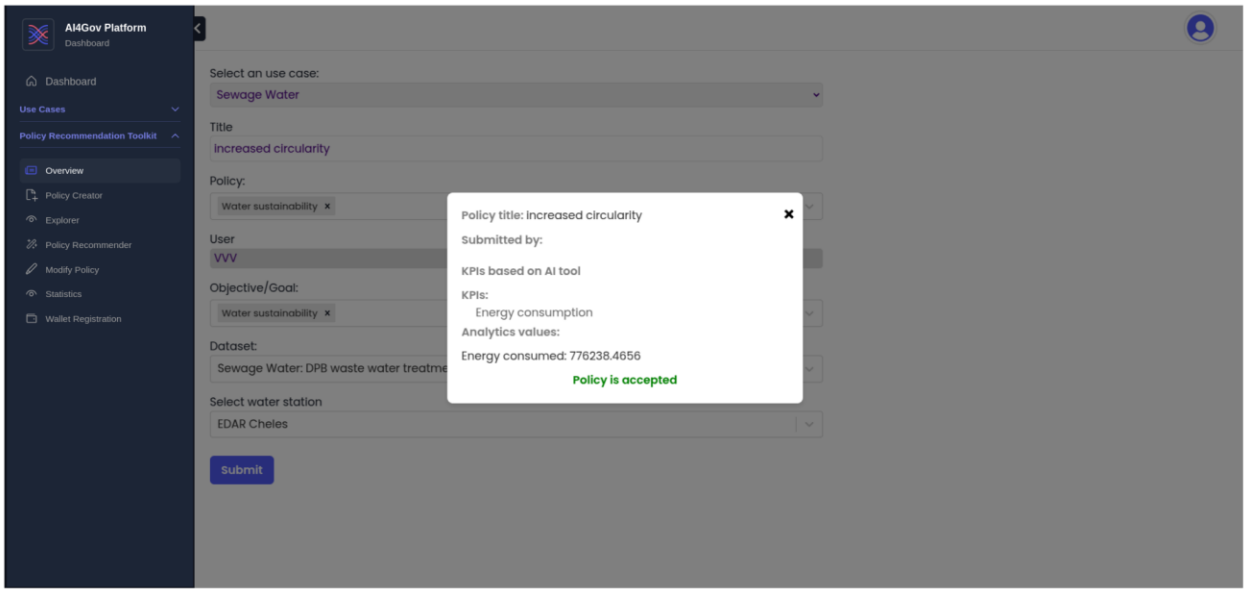


Figure 26: Total policy recommendation and analytics results in Sewage Water Management category

4.1.3 PRT Explorer

The user can navigate among the existing policies (Figure 27). Then a policy can be chosen, and the user can examine and read the details of each policy and endorse the preferable policy (Figure 28).

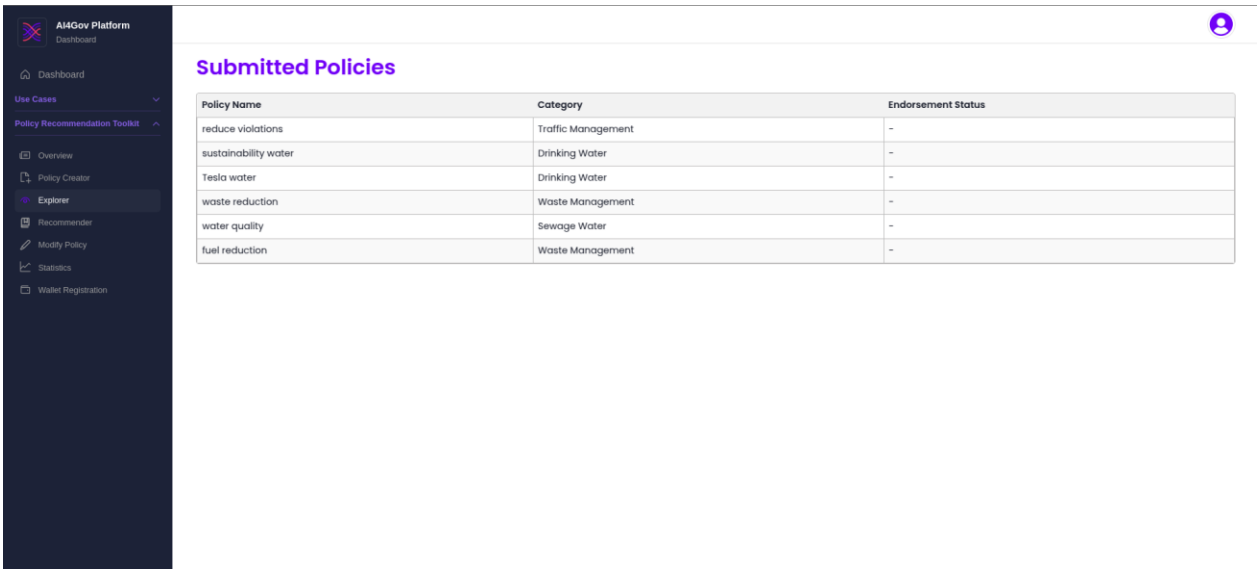


Figure 27: Policy Explorer

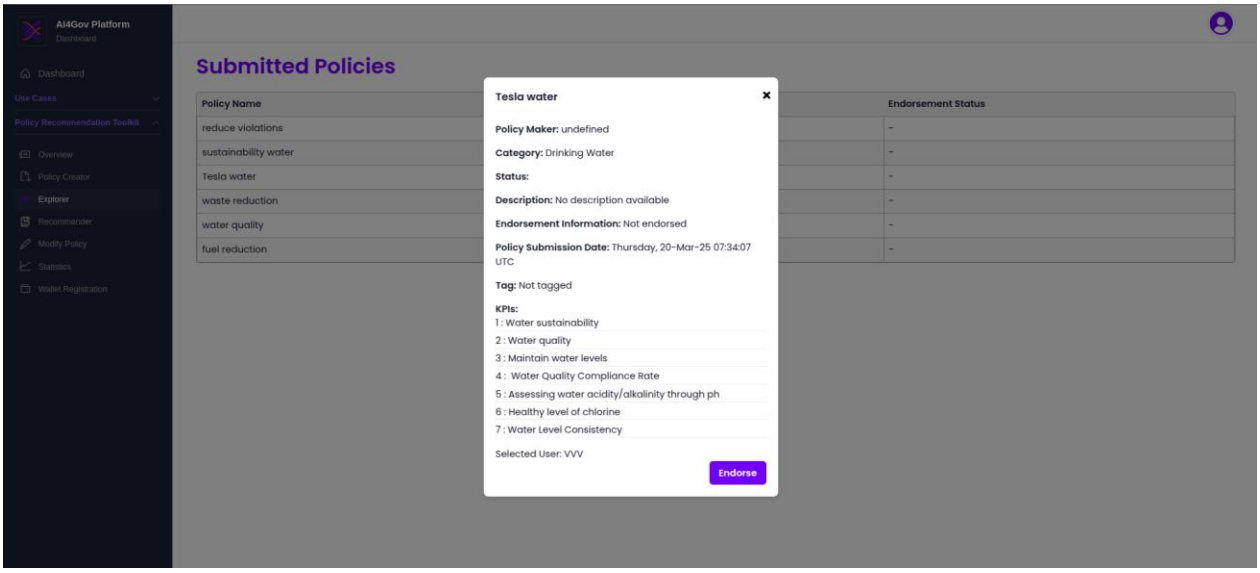


Figure 28: Policy Endorsement

In the case where the user has logged into the application as a “citizen”, this particular screen functions differently. As depicted in Figure 29, instead of a list of policies, a QR code appears, which the citizen is prompted to scan using the mobile application in order to prove that they are a verified user.

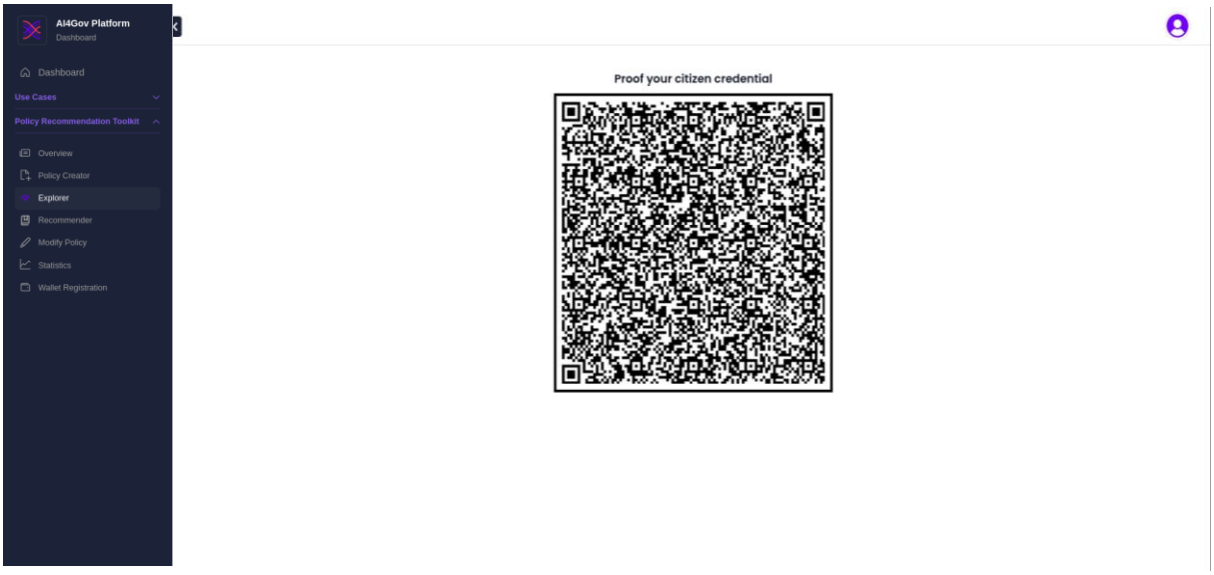


Figure 29: Proof invitation for Policy Voting by citizen

By following the two-step verification process, the user scans the QR code from their personal wallet. Once the user scans the QR code, they essentially send a request with their credentials to the platform. There, it is verified via blockchain whether the user actually possesses the required credential. If there is no discrepancy, the system proceeds with sending the proof request. As the process depicted in Figure 30.

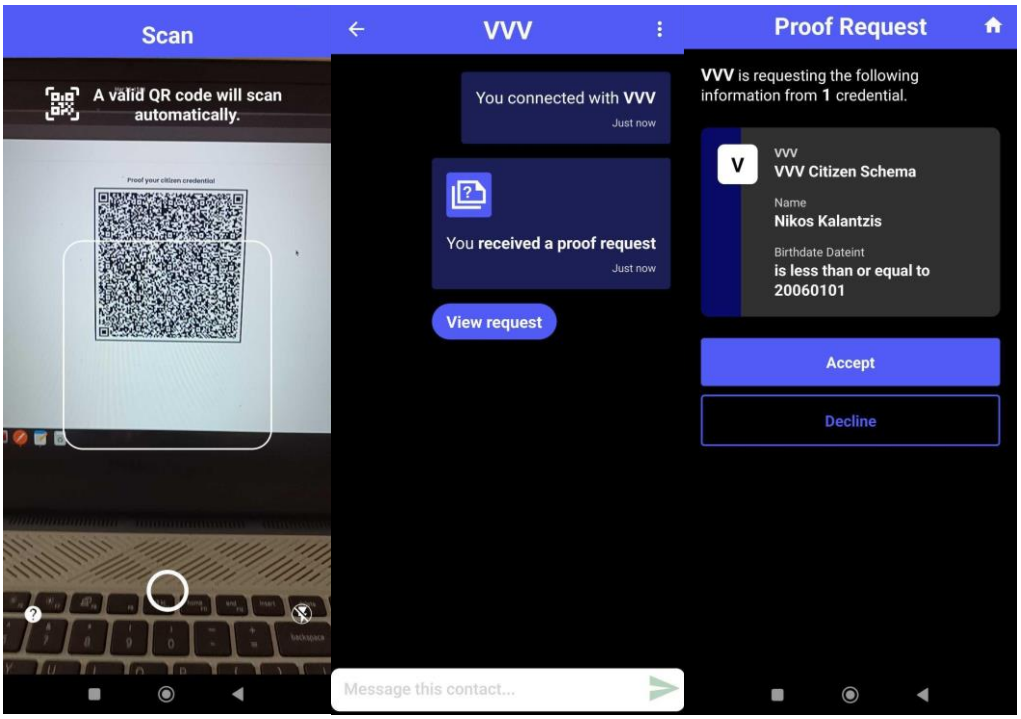


Figure 30: Policy Vote by citizen (Accepted proof request)

Then, the user accepts the proof request and enters the list of stored policies through the platform in order to review them and possibly vote on one that interests them as depicted in Figure 27. On the wallet's side, the completion of the process is displayed as depicted in Figure 31. Then a policy can be chosen, and the user can examine and read the details of each policy and vote the preferable policy (Figure 32).

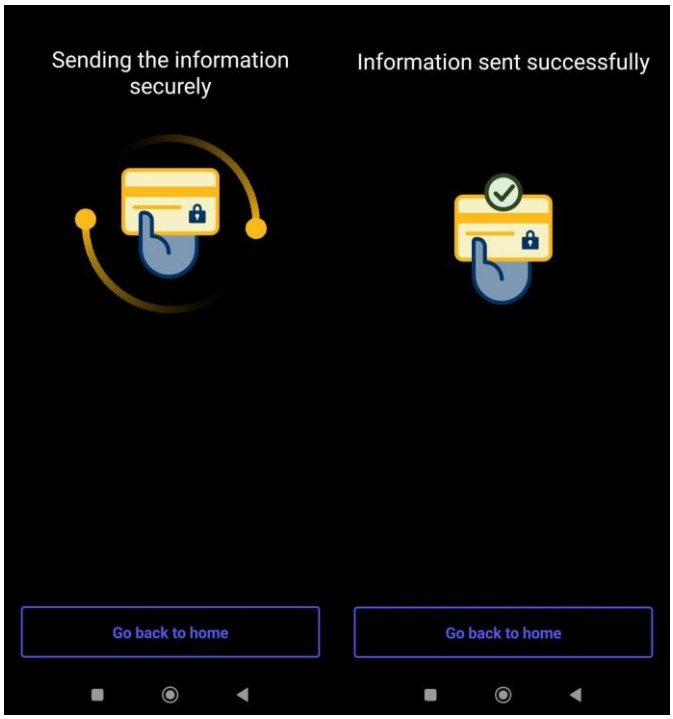


Figure 31: Policy Vote by citizen (Successfully complete proof request)

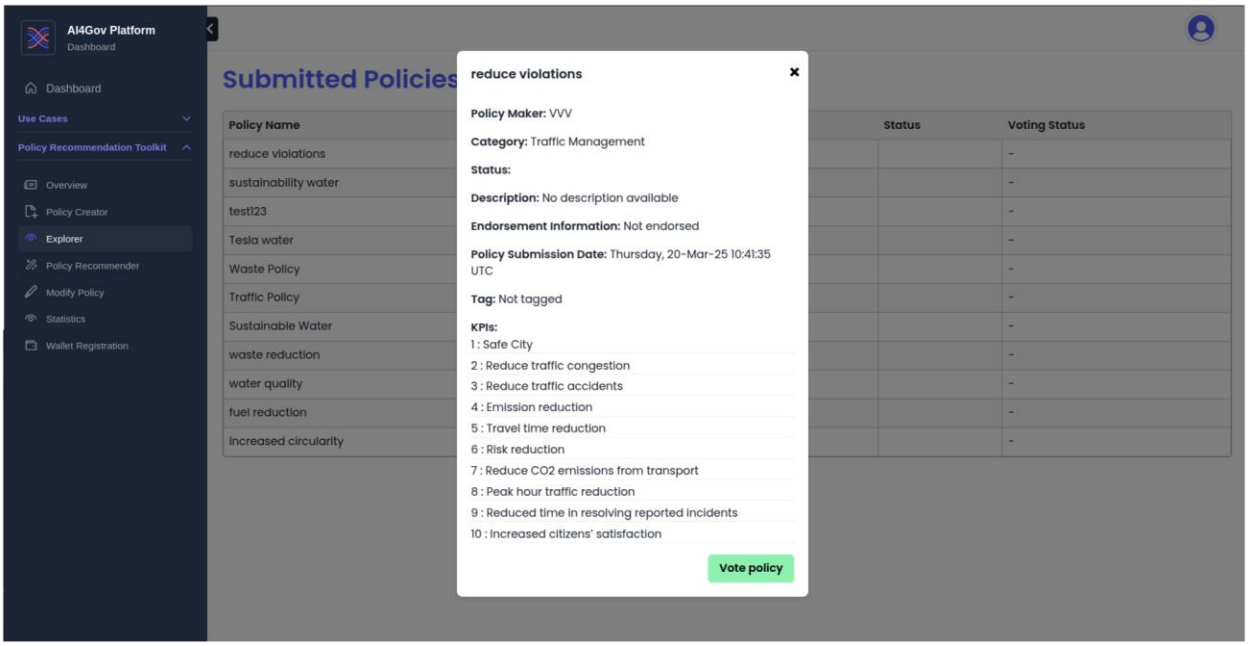


Figure 32: Policy Voting by citizen

If the user has not been certified by an organization, the process is rejected by the system, the user receives the corresponding message as depicted in Figure 33, and the platform does not proceed further in the process of voting on a policy.

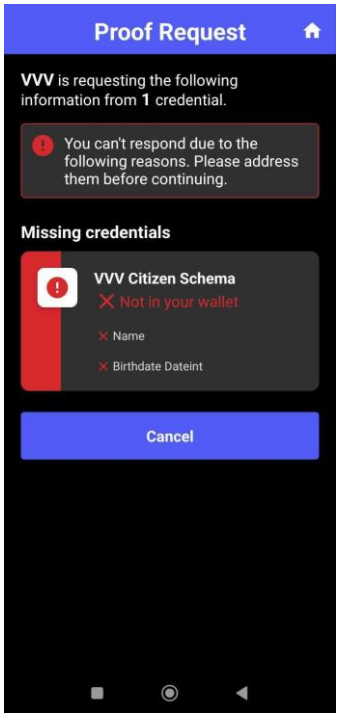


Figure 33: Policy Vote by citizen (Citizen rejected)

4.1.4 PRT Recommender

The user is selecting KPIs as hard and soft constraints and retrieves as a response the policies that fulfil the KPIs of hard constraints, Figure 34. The retrieving policies have a rate regarding the soft KPIs that are fulfilled.

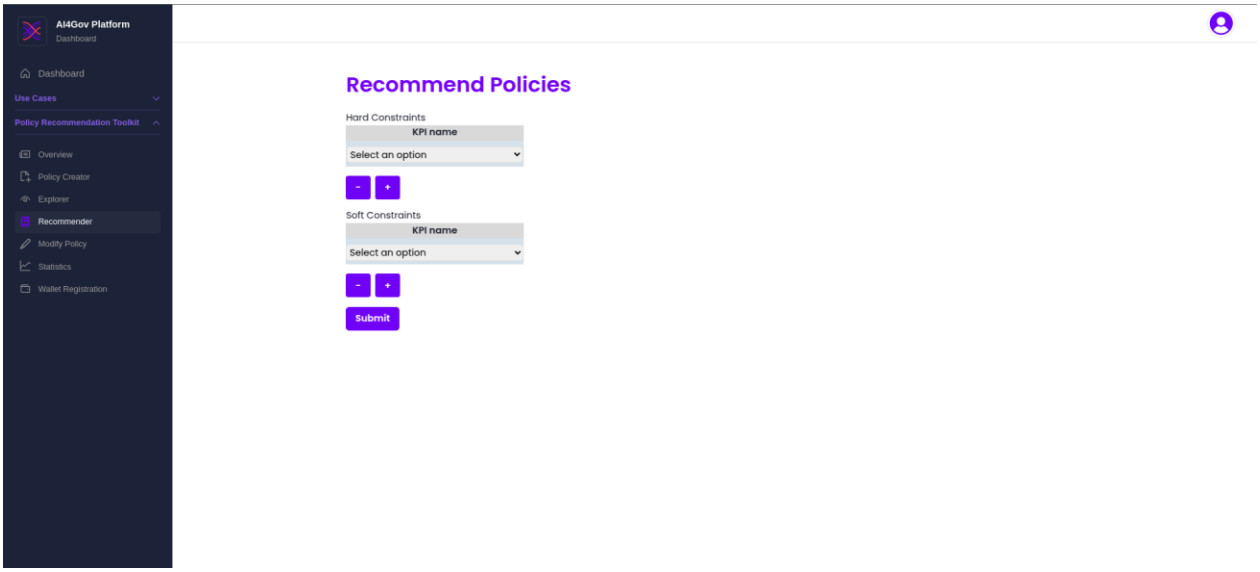


Figure 34: Recommendation results and KPI / constraints interface

4.1.5 PRT Modify Policy

There is an editing option via addition or re-definition of the relevant KPIs (Figure 35).

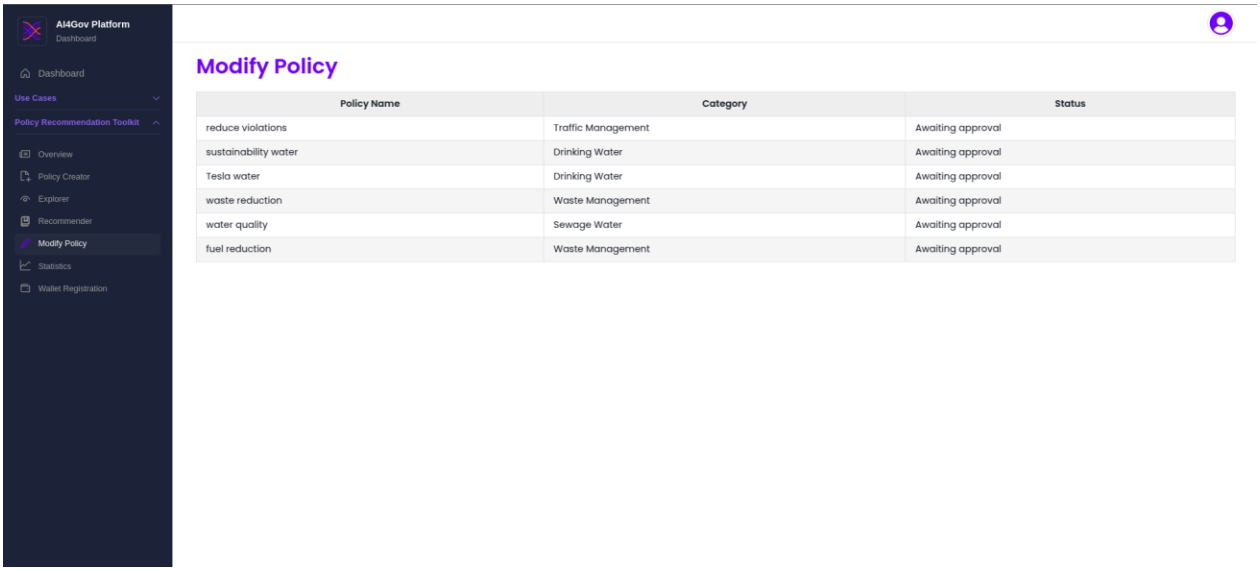


Figure 35: Modification of policies and editing interface

4.1.6 PRT Statistics

Also, there is a statistics menu that describes some analytics about the categories, the status and the endorsement of policies, Figure 36.

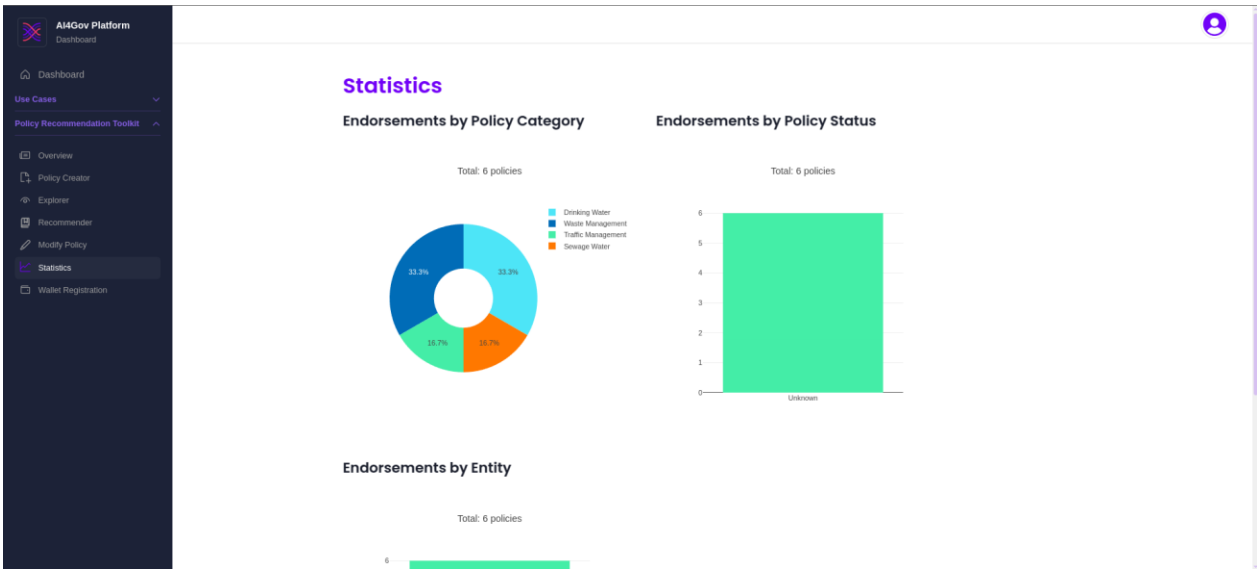


Figure 36: Comparative analytics

4.1.7 PRT Wallet Registration

Finally, there is a menu about the credential issuance operation. The first step is to share a QR code for establishing a connection between the issuer (VVV or DPB) and the holder (citizen wallet), Figure 37. After the connection is established, the Issuer can fill the form about the attributes of holder/citizen and send the credential offer to the holder in order to store this to the wallet, Figure 38.

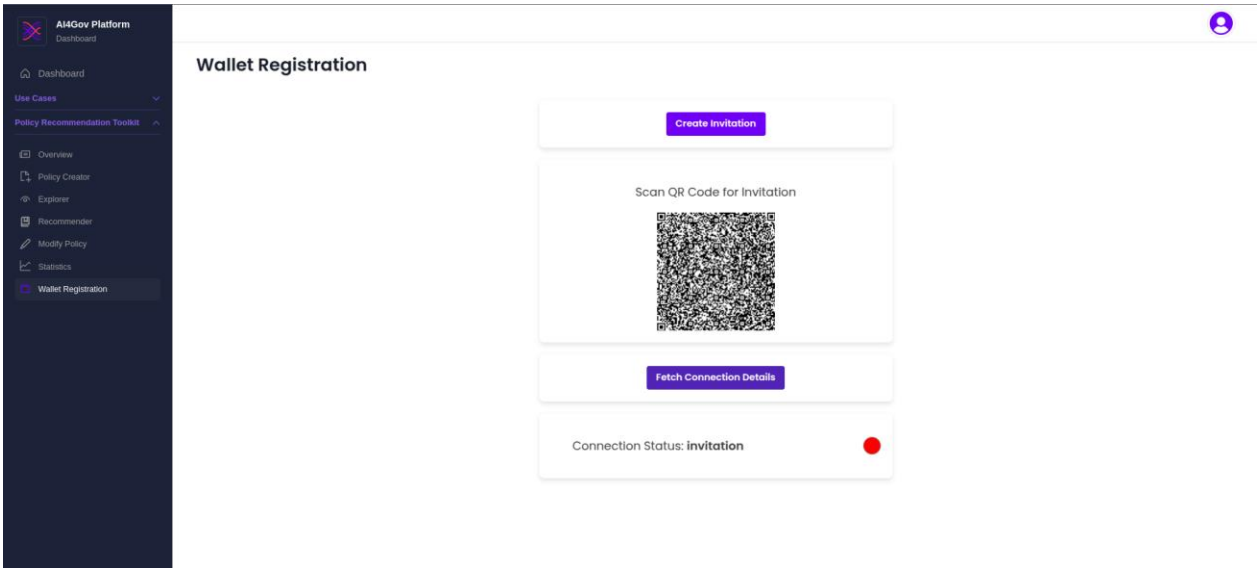


Figure 37: Sharing QR code for citizen invitation

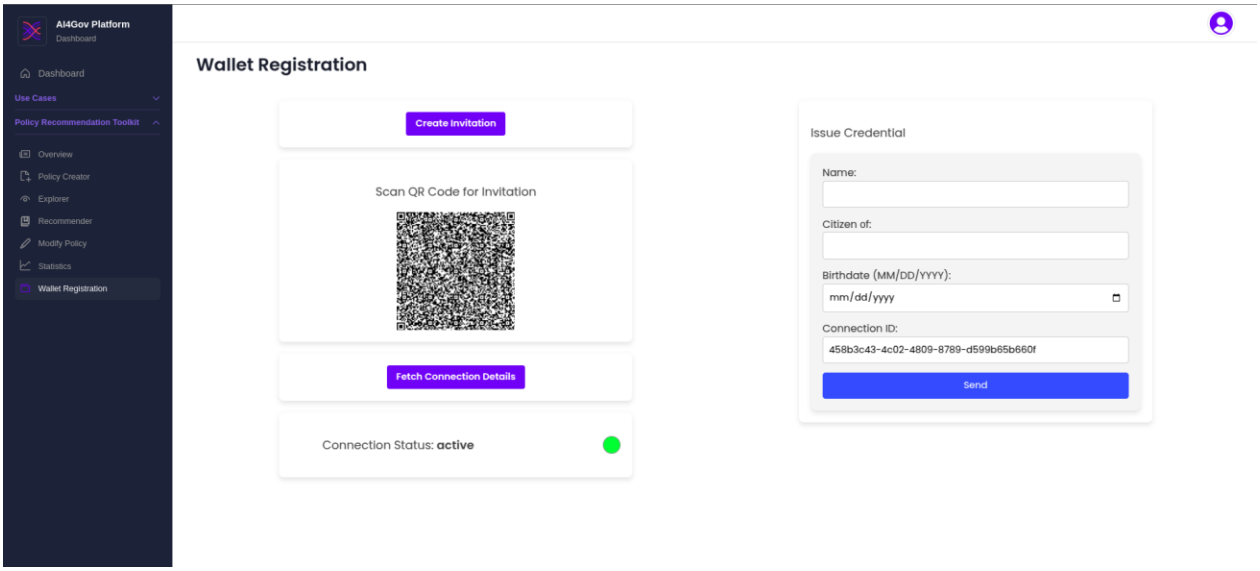


Figure 38: Form of attributes of verifiable credential

4.2 Citizens' Wallet

Monitoring and evaluation of policies can also be performed by citizens via an appropriate dApp (Citizen Wallet). We start with an empty citizen wallet that holds no credentials (Figure 39).

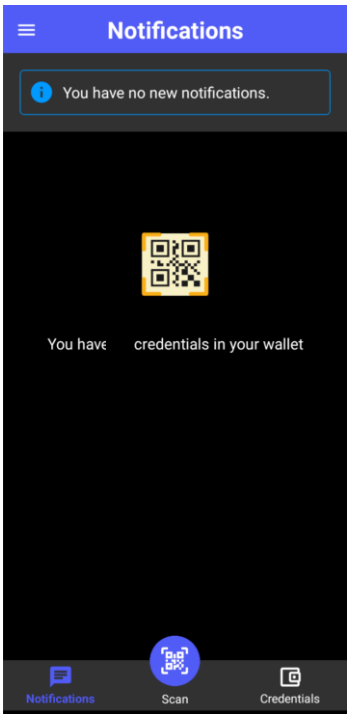


Figure 39: Citizens' wallet – Initial screen

An organization that distributes the wallet, such as the government or a municipality, can generate invitations to agents, thus allowing them to board the platform. One such invitation captured by the backend of the Aries infrastructure is depicted in Figure 40. This QR code is depicted in the Visualization Workbench.

The user can now accept the invitation by scanning the transmitted QR code (Figure 41 left); after some time, she/he is connected with the issuer of the invitation, which, in this scenario, is VVV (Figure 41 right). After the invitation, the issuer can issue a full credential and offer it to the citizen (Figure 42 left); if the citizen accepts, she/he now has a credential filled with all the attributes sent by the issuer (Figure 42 right). The citizen can verify that the credential presented in her/his screen is the same as the one recorded in the blockchain; she/he is free to reject the credential, if a mismatch is identified. This credential can now be presented to any party requiring proof under the VC scheme[2].

[2] A common misunderstanding is that this scheme proves the truth of the claims the holder presents. This is not entirely true. To be perfectly precise the holder can prove that the issuer has signed the validity of the claim. For example, a holder can prove that VVV confirms that the subject's name is John Papadopoulos. Whether this claim is true or not, and more importantly whether it can be trusted or not, depends upon the level of trust that the verifier has towards the issuer.

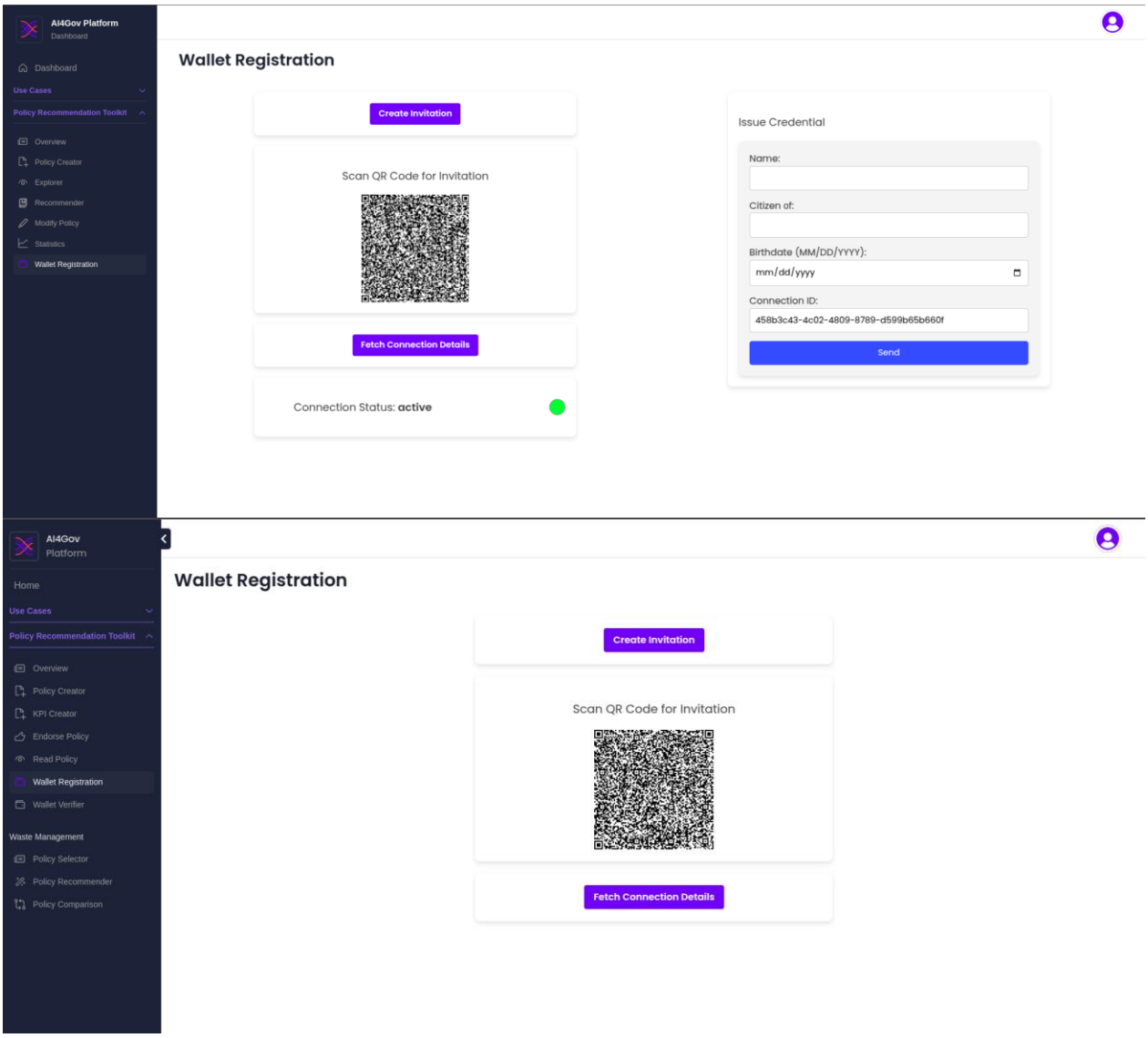


Figure 40: Boarding invitation generated by HyperLedger Aries

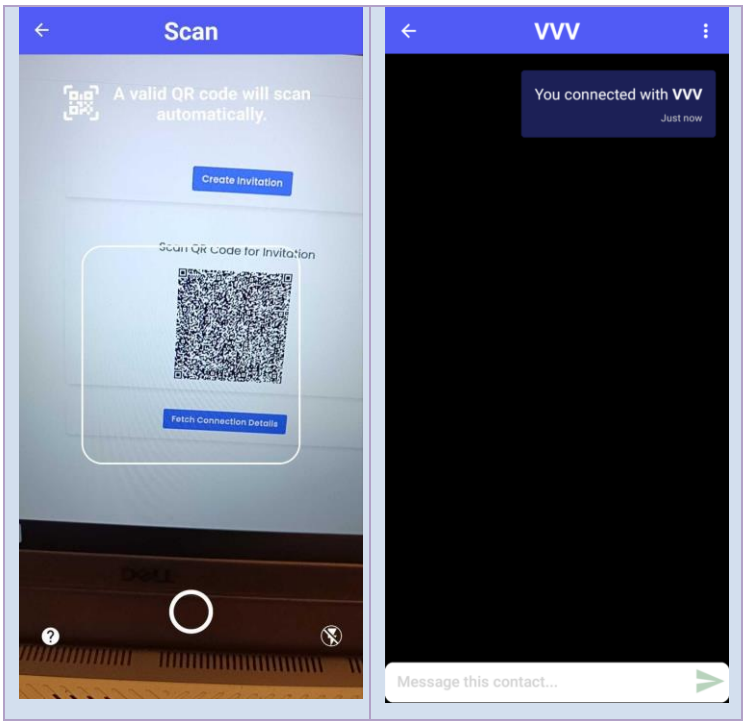


Figure 41: Accepting the invitation

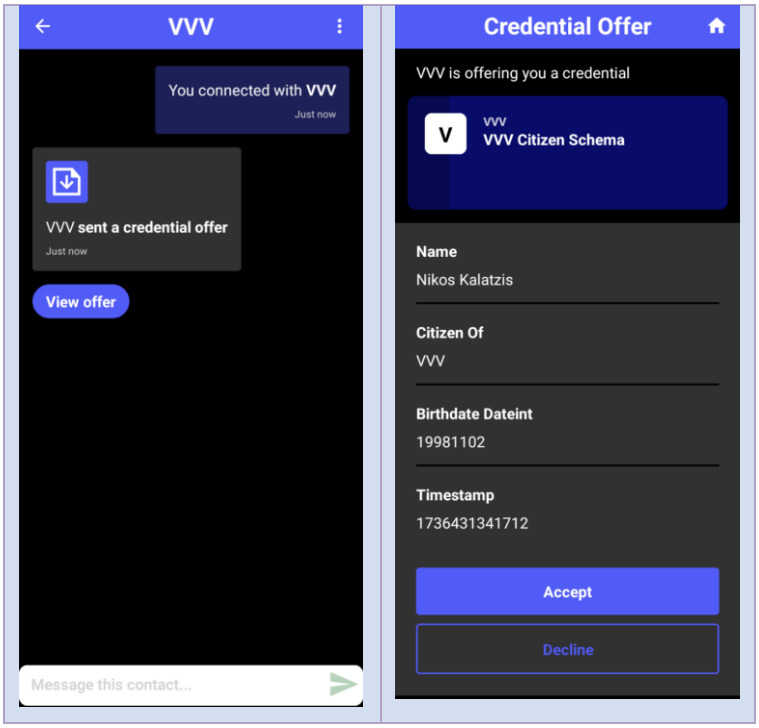


Figure 42: Accepting the Credential

As previously described, an Android application was deployed for mobile devices, allowing citizens to monitor policies and participate by voting and providing feedback. This process is presented below. Initially, the user utilizes the wallet on their personal mobile phone, simultaneously verifying their identity. From the application's home screen, they select the "**Vote Policy**" tab in the bottom menu. On this screen, the user's stored credentials are displayed, as depicted in the Figure 43.

Figure 43: Saved credentials in the wallet

After the user selects the appropriate credentials, they are taken to the next screen, where a list of all policies is displayed, as depicted in Figure 44.

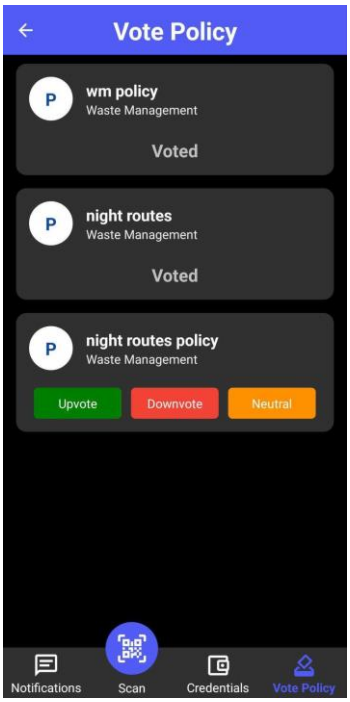


Figure 44: Select policy to Vote

If the user wishes to see more details regarding the policies, they simply need to click on one, and a window will appear with the relevant information about the specific policy, Figure 45.

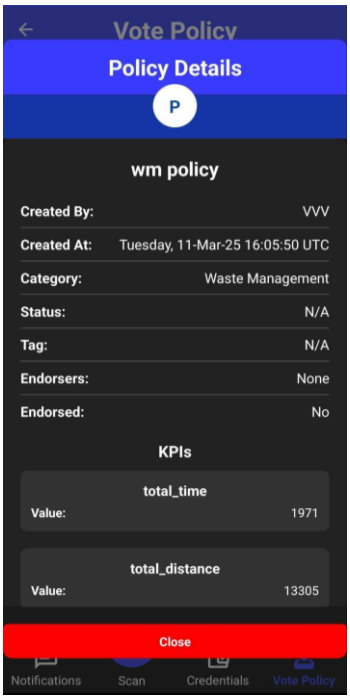


Figure 45: View the policy details

For each policy, there are three different voting options: positive, negative, and neutral. In this case, the citizen has the ability to choose one of the three. If the citizen with the given credentials has already voted on a policy, the buttons do not appear. Instead, a message is displayed stating that the specific policy has already been voted on by the user. In the first case, 'positive' means they approve of the specific policy; in the second case, 'negative' means they disapprove the policy; and in the last case, 'neutral' means they do not express any opinion on the specific policy. For each of these choices, a window appears confirming the voting decision, as depicted in Figure 46.

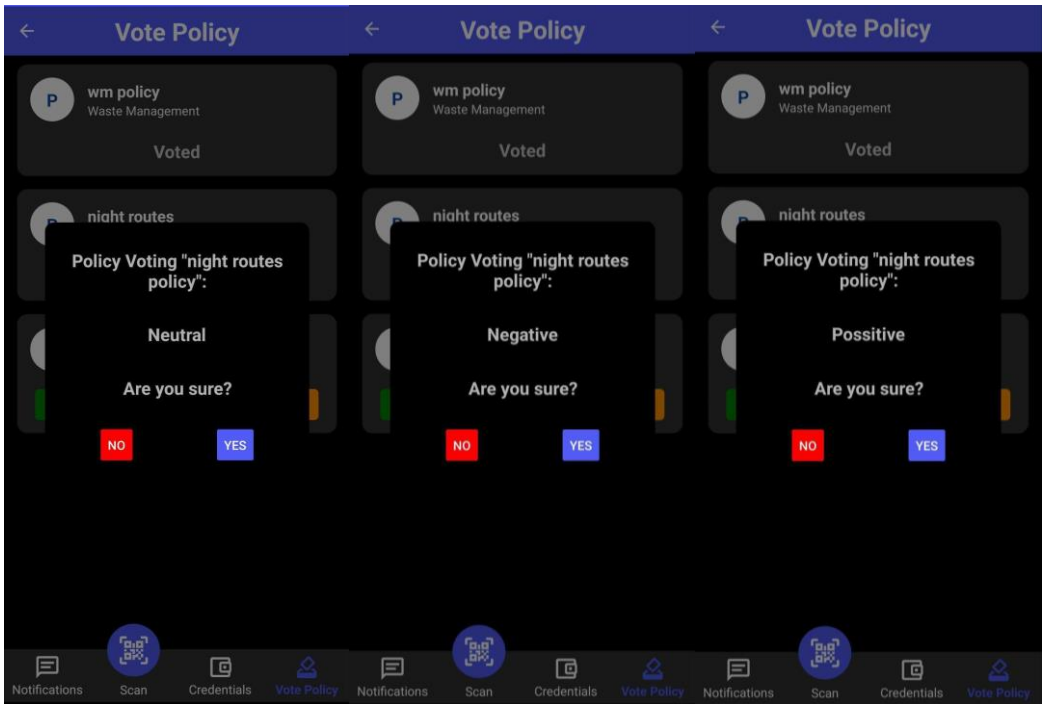


Figure 46: Vote Policy for three options (positive, negative, neutral)

After the voting process is completed, a content window appears to inform the user that the process has been completed, along with a thank-you message, as depicted in Figure 47.

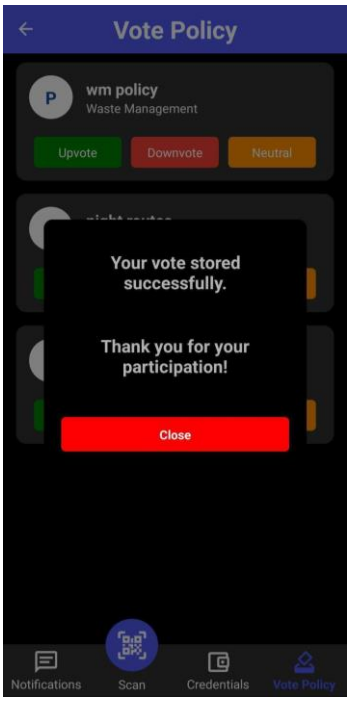


Figure 47: Successfully voting

Finally, the voting process has been completed. The user now returns to the previous screen and can no longer vote for the same policy again. Additionally, they can check the policy details by tapping on it, with the difference that in the new window that appears, the current ballot results will also be displayed, Figure 48. At this point, it is worth noting that the result reflects the difference between the participation counts rather than the total number of votes. This helps to avoid bias.

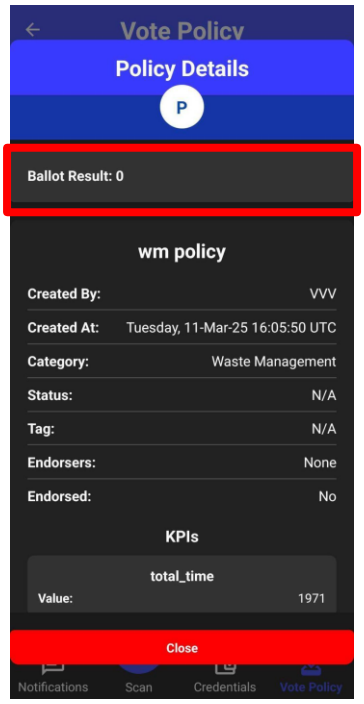


Figure 48: Ballot results after completing vote

The citizen can access the Visualization Workbench and use all the relevant functionalities by using the wallet. As depicted in Figure 49.

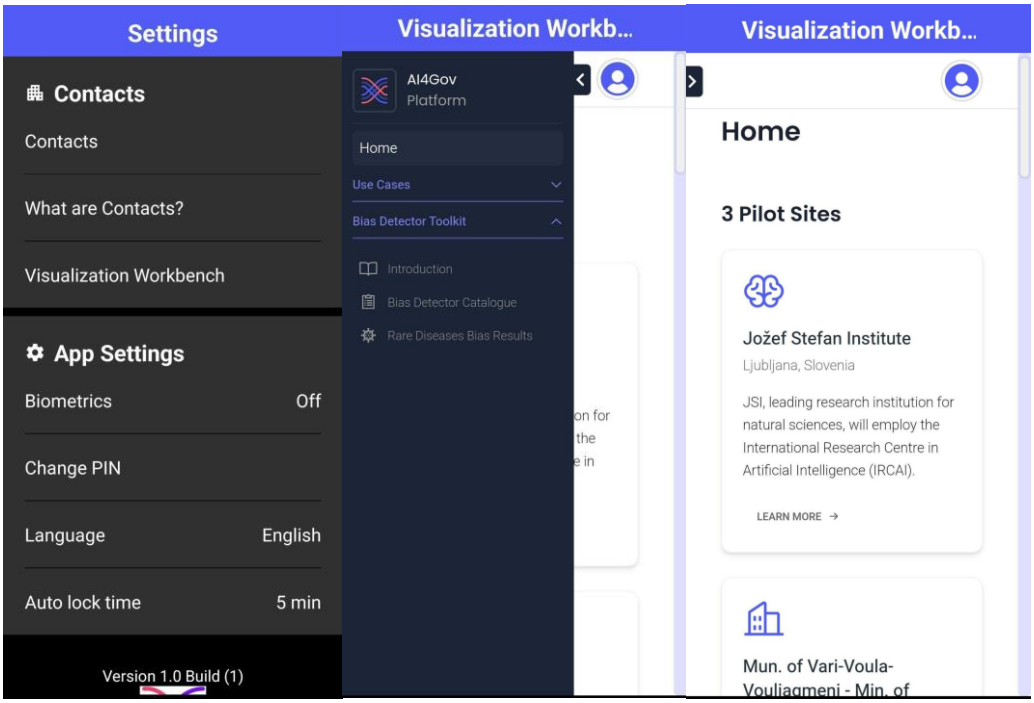


Figure 49: Visualization Workbench via mobile

5 Data Governance Framework

5.1 DGF Overview

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 was firstly introduced in D3.1: Decentralized Data Governance, Provenance and Reliability V1, with the final version which introduced a comprehensive set of rules and guidelines sourcing from the AI Act being integrated within D3.2: Decentralized Data Governance, Provenance and Reliability V2.

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. A high-level of the available information of the DGF as well as its structure are illustrated in the Figure 50 and Figure 51, respectively:

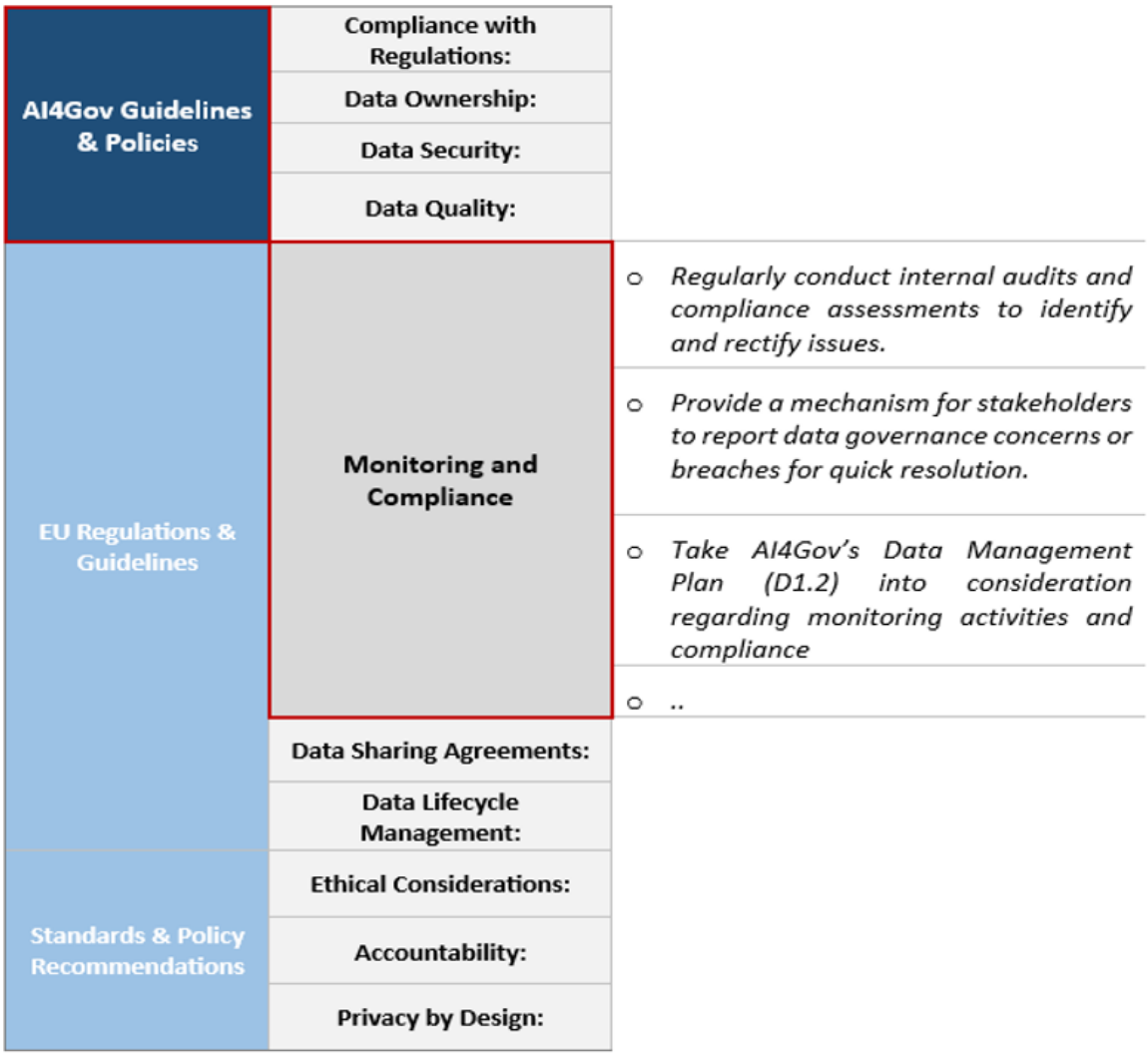


Figure 50: High-level illustration of DGF structure (1/2)

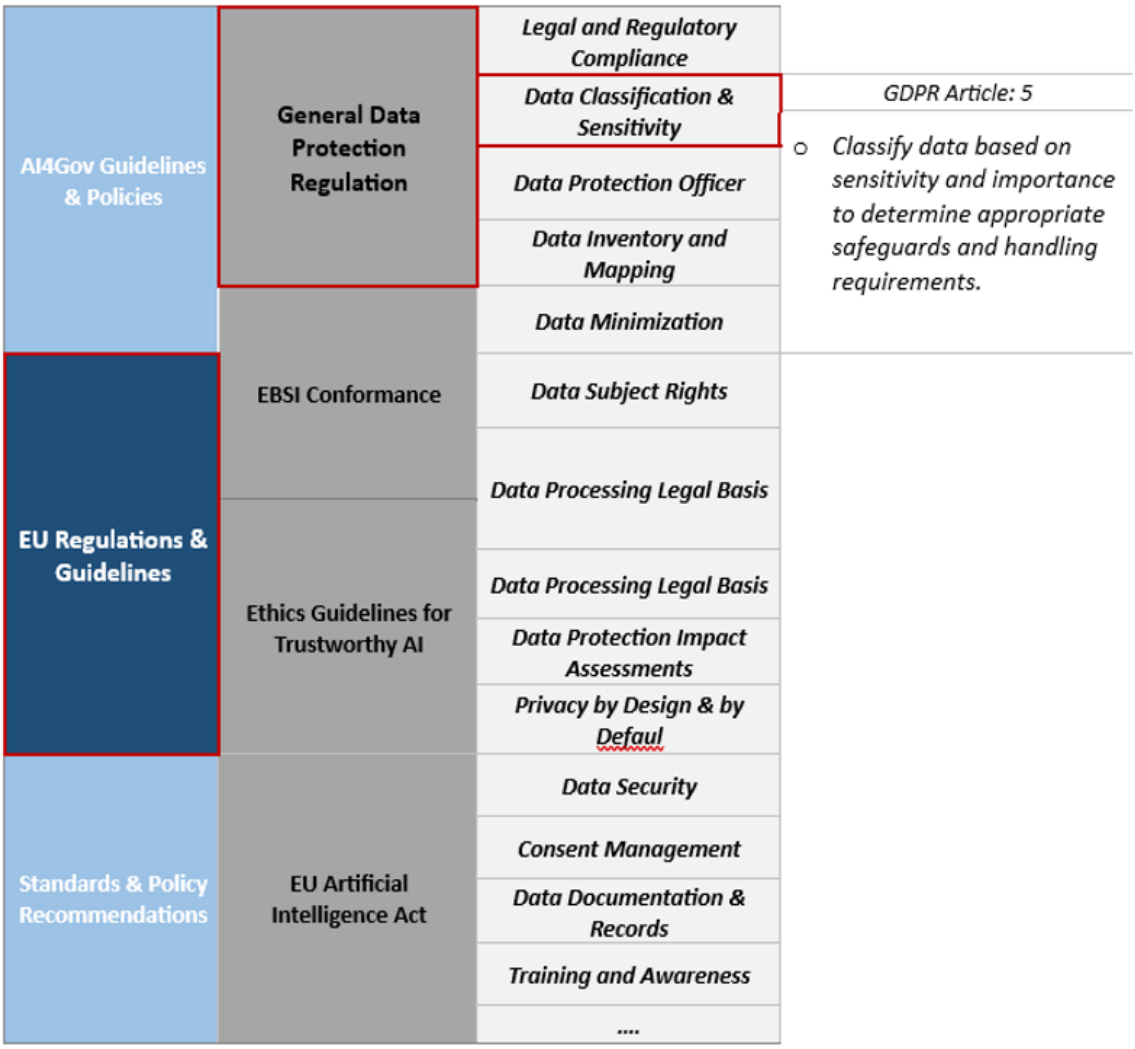


Figure 51: High-level illustration of DGF structure (2/2)

5.2 DGF KPIs introduced within the Policy Recommendation Toolkit

In alignment with the rules and guidelines set forth by the DGF, a series of horizontal Key Performance Indicators (KPIs) have been developed and integrated into the Policy Recommendation Toolkit Deliverable. These KPIs serve as measurable criteria to assess the extent to which organizational policies and AI systems comply with essential requirements stemming from the AI Act, such as risk classification, transparency, data protection, human oversight, and security. By incorporating these cross-cutting KPIs, the PRT enables consistent evaluation across different policy domains, ensuring that all technical implementations within the project uphold the principles of trustworthy AI and adhere to EU regulatory standards. Furthermore, these indicators provide project partners with actionable insights to guide decision-making, monitor compliance, and foster continuous improvement in data and AI governance practices.

As many of the policies center around the AI models developed within the pilots under WP6, the KPIs have been designed to specifically address the AI implementation aspects of the Data Governance Framework. These KPIs aim to evaluate how effectively the deployed AI systems align with the governance, compliance, and ethical standards outlined in the DGF. The KPIs introduced for this purpose are presented in

Table 9. These KPIs have been incorporated in the final prototype of PRT.

Table 9: DFG KPIs introduced to the PRT platform

AI Category	Description	KPIs (connected to underlying Pilot's AI models)	Variable	Measurement	Threshold
1. AI Risk Classification Policy	Measures the accuracy of risk-level classification for AI systems according to the AI Act's risk framework (unacceptable, high, limited, minimal). Ensures proper safeguards are applied based on risk category.	AI Model Risk Classification Accuracy	ai_model_risk_classification	(Number of correctly classified AI systems / Total AI systems) × 100	≥ 80% on
2. AI Transparency and Explainability	Assesses how many AI systems provide clear, accessible explanations for their decisions. Supports transparency obligations by enabling stakeholders to understand and evaluate AI outcomes.	AI Decision Explainability Rate	ai_decision_explanation	(Number of AI systems with explainability documentation / Total AI systems) × 100	≥ 90% on
3. AI Data Protection Policy	Evaluates the extent to which personal data processed by AI systems is anonymized or pseudonymized, enhancing privacy protection and compliance with GDPR and the AI Act.	AI Personal Data Anonymization Rate	ai_personal_data_anonymized	(Number of anonymized AI data points / Total AI data points) × 100	≥ 90% on
4. AI Ethical Considerations	Tracks the presence of unfair bias in AI models, aiming to ensure fairness, non-discrimination, and ethical AI usage. Helps mitigate risks related to marginalization or unjust outcomes.	Bias Detection in AI Models	ai_bias_detection	(Number of AI models flagged for bias / Total AI models) × 100	≤ 5% on

5. AI Human Oversight Mechanism	Monitors how many AI systems integrate mechanisms for human oversight or intervention. Ensures accountability and allows human control over critical AI decisions.	Human-In-The-Loop Rate	human_in_the_loop_integration	(Number of AI models with human oversight / Total AI models) × 100	≥ 80% on
6. AI Security and Robustness Policy	Measures compliance of AI systems with security standards and audit results. Promotes resilience against cyber threats and maintains system robustness and reliability.	AI System Security Compliance	ai_security_compliance	(Number of AI systems passing security audits / Total AI systems) × 100	≥ 90% on
7. AI Governance & Monitoring	Evaluates the proportion of AI systems that meet AI Act requirements. Ensures alignment with legal standards, reducing the risk of regulatory penalties and fostering trust.	AI Regulatory Compliance Rate	ai_regulatory_compliance	(Number of AI systems compliant with AI Act / Total AI systems) × 100	≥ 90% on
8. AI Impact Assessment Policy	Tracks the completion rate of risk and impact assessments for potential high-risk AI systems. Facilitates early identification of potential harms and supports risk mitigation planning.	AI Risk Impact Assessment Completion	ai_risk_impact_assessment	(Number of AI impact assessments completed / Total AI systems) × 100	≥ 80% on

Moreover, the development and inclusion of these KPIs extends beyond the original scope and intent of the Data Governance Framework, which was conceived primarily as an internal reference document to guide project partners in the responsible collection, production, and processing of data. As such, while these indicators offer a forward-looking perspective on operationalizing trustworthy AI governance, their current function remains advisory, supporting further enhancements and iterations of the platform rather than immediate deployment.

6 Conclusions

In this report, the final iteration of the requirements of the PRT was given, together with the final version of the architecture that fulfils these requirements. As efficient policymaking should allow for citizen feedback and co-creation, the requirements defined in the initial pilot definitions have been expanded, as to include a citizen component that is expected to actively increase citizen participation in policymaking. The technology enablers and components that promote openness and inclusiveness that have been documented in D3.1, D3.2 and D3.3, in this final version of PRT have been implemented and integrated, namely AI Recommendations, ZKP, Homomorphic Encryption and Citizen Wallet. Going beyond the original scope of the PRT implementation, special emphasis was given to the implementation of a Citizen Wallet as an added key enhancement. This wallet empowers citizens to form opinions, provide feedback to policymakers and vote for recommended policies leveraging cryptographic algorithms to foster a secure, trustworthy and transparent framework for open democratic engagement.

7 References

Feige, U., Fiat, A., & Shamir, A. (1987). ZERO KNOWLEDGE PROOFS OF IDENTITY. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 210–217. https://doi.org/10.1145/28395.28419

Landemore, H. (2020). *Open democracy: Reinventing popular rule for the twenty-first century*. https://books.google.com/books?hl=en&lr=&id=fe7xDwAAQBAJ&oi=fnd&pg=PA247&dq=open+democrcy+helene+landemore&ots=sw3kkUpnpM&sig=vdlESk8SoBWBN9XU_zEM9PuJUt0

Semantic Web - Wikipedia. (n.d.). Retrieved April 22, 2024, from https://www.w3.org/standards/semanticweb/

Thatcher, M., & J. B.-I. E. of the S., & 2015, undefined. (n.d.). Issue Networks: Iron Triangles, Subgovernments, Policy Communities, Policy Networks. *Iris.Luiss.ItM Thatcher, J BraunsteinInternational Encyclopedia of the Social & Behavioral Sciences (Second Edition), 2015•iris.Luiss.It*. Retrieved April 19, 2024, from https://iris.luiss.it/handle/11385/184949

Mark A. Will, Ryan K.L. Ko, Chapter 5 - A guide to homomorphic encryption, Editor(s): Ryan Ko, Kim-Kwang Raymond Choo, The Cloud Security Ecosystem, Syngress, 2015, Pages 101-127, ISBN 9780128015957, https://doi.org/10.1016/B978-0-12-801595-7.00005-7. (https://www.sciencedirect.com/science/article/pii/B9780128015957000057)

8 Appendix

AI4Gov Implementation of homomorphic encryption

Key Generation

1. Selection of Two Large Prime Numbers
    - Two large prime numbers, **p** and **q**, are randomly chosen such that their product **n = p × q** is used as part of the public key.
2. Computation of Public and Private Keys
    - The public key **(n, g)** is computed, where **g** is a generator chosen such that it satisfies the necessary mathematical properties for encryption.
    - The private key **(λ, μ) λ** is derived using **λ = lcm(p-1, q-1)**, and an auxiliary value **μ** is calculated for decryption.
3. Key Distribution
    - The **public key (n, g)** is published on the **blockchain**, making it accessible to all voting clients (citizen wallets) for encrypting votes.
    - The **private key (λ, μ)** is securely stored by the election authority and is never shared, ensuring that only authorized personnel can decrypt the final result.

Encryption, Decryption and addition process

The encryption formula for a given public key and a message m is calculated to cipher as c= E(m) with the below function:

$$E(m) = c = \ g^m \cdot r^n \ mod\ n^2$$

The decryption formula for a given private key and a cipher c is calculated to plain text m = D(c) with the below function:

$$D(c) = m = \frac{(c^\lambda mod\ n^2) - 1}{n} \mu\ mod\ \ n$$

The homomorphic addition of plaintexts m_1 and m_2 is calculated as shown in the below function:

$$D(E(m_1, r_1)\ \cdot\ E(m_2, r_2)\ mod\ n^2) = m_1 + m_2 mod\ n$$

By observing the above operations it is concluded that the addition does not work for negative results. Since the Paillier cryptosystem is based on modulo arithmetic, if it is tried to add numbers that produce a negative result the number that we getting from decryption function is a number very close to the n. Because the result of this $0 < \frac{(c^\lambda mod\ n^2) - 1}{n} < n$. So the formula needs to be

modified.

There are two ways to manage this issue.

1. The negative result can be handled by checking if the result of the addition $f(m)$ is a number close to n. If this is true then the final result should be $f(m) - n$. The way that the check can take place is by assigning a threshold about the amount of digits of the $f(m)$. If the digits are too many then it is assumed that the result is very close to n. So the final result that should be $f(m) - n$. In other case the result is $f(m)$.

   The below code describes this solution for two encrypted votes

```
...
// Homomorphic addition: Enc(x1) * Enc(x2) mod n^2
let totalEncrypted = (encryptedVote1 * encryptedVote2) % (publicKey.n ** BigInt(2))
let finalCount = privateKey.decrypt(totalEncrypted);
// n = p * q
if ((finalCount.toString()).length > 20)
finalCount = finalCount - publicKey.n
...
```

2. Another approach is to shift the values of votes in order to not produce negative results. In this case the potential numbers of votes are -1, 0 and 1. So it is suffice to shift those numbers by 1 and the new range can be modified to 0, 1 and 2. After this modification the process can be followed properly and the in final result should be reduced the amount that is added by shifting the numbers range. This amount can calculated as $l \cdot a$ (l is number of calculated votes and a is the shift index). So the final result is:

   filanResult = result - $l \cdot a$

   The bellow code describes this solution for three encrypted votes

```
// Define shift value (k) for handling negative numbers
const k = 1;
const x1 = -1;
const x2 = 1;
const encodedX1 = x1 + k;
```

```
const encodedX2 = x2 + k;
// Encrypt the encoded numbers
const encryptedX1 = publicKey.encrypt(BigInt(encodedX1));
const encryptedX2 = publicKey.encrypt(BigInt(encodedX2));
// Homomorphic addition: Enc(x1) * Enc(x2) * Enc(x2) mod n^2
const encryptedSum = publicKey.addition(encryptedX1, encryptedX2, encryptedX2);
// Decrypt the result
const decryptedSum = privateKey.decrypt(encryptedSum);
// Decode the result (shift back)
const finalResult = Number(decryptedSum) - 3* k;
```

The case that has been selected as a solution in this specific scenario is the first.