



Development of Intelligent Mechanisms for Contextual Support of Writing and Convergence of Proposals in Participatory Processes

MIGUEL RODRIGUES DOMINGUES

Junho de 2026

Development of Intelligent Mechanisms for Contextual Support of Writing and Convergence of Proposals in Participatory Processes

Miguel Rodrigues Domingues

**Dissertation to obtain the Master's Degree in
Informatics Engineering, Specialization Area in
Software Engineering**

Supervisor: Goreti Marreiros

Porto, June 2026

Statement of Integrity

I hereby declare that I have conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore, the work presented in this document is original and authored by me, having not previously been used for any other end. The exceptions are explicitly recognised in the section “Ethical and Professional Considerations” of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P. PORTO.

ISEP, Porto, June 28, 2026

Dedication

To my parents and grandparents, who have always been there for me.

Thank you for your constant support, for everything you taught me, and for helping me become the person I am today. Much of what I have achieved is the result of the values, strength, and guidance you gave me throughout my life.

To my grandparents who are no longer here, I hope this achievement will make you proud.

Finally, I dedicate this work to my girlfriend, who supported me every day throughout this Master's journey. Thank you for giving me strength when things became difficult, for helping me look further, and for always encouraging me to push a little more.

Resumo

A participação cidadã em processos democráticos digitais tem vindo a ganhar importância nos últimos anos, sobretudo através de plataformas de participação eletrónica utilizadas em orçamentos participativos, consultas públicas e outras iniciativas de envolvimento cívico. Apesar deste crescimento, continuam a existir dificuldades relevantes na forma como os cidadãos formulam as suas contribuições. Muitas propostas são submetidas com pouca clareza, estrutura insuficiente, informação incompleta ou desalinhamento com as regras do processo participativo. Além disso, a existência de propostas semelhantes ou complementares pode levar à fragmentação de ideias, reduzindo a visibilidade de iniciativas com objetivos comuns e dificultando a construção de soluções mais fortes e colaborativas.

Esta dissertação foi desenvolvida no contexto do projeto CoParticipation e tem como objetivo estudar, conceber, implementar e avaliar mecanismos inteligentes de apoio contextual à escrita e à convergência de propostas em processos participativos digitais. O trabalho foca-se na utilização de técnicas de Processamento de Linguagem Natural e de Modelos de Linguagem de Grande Dimensão para apoiar os cidadãos durante a criação de propostas, sem substituir a sua decisão nem o processo formal de avaliação realizado pelas entidades responsáveis.

A investigação seguiu uma metodologia de *Design and Creation*, combinando revisão do estado da arte, análise de requisitos, desenho da solução, implementação de um protótipo funcional e avaliação experimental. A solução proposta estende o ecossistema da *framework* EMPATIA através de uma camada de assistência baseada em Inteligência Artificial, organizada como um serviço independente. O protótipo é composto por uma aplicação web desenvolvida em *Laravel* e *Livewire*, responsável pela interação com o utilizador e pela gestão de propostas, e por um serviço de Inteligência Artificial desenvolvido em *FastAPI*, responsável pelo processamento inteligente, pela comunicação com modelos de linguagem executados localmente e pela gestão de conhecimento contextual.

A solução implementada inclui dois mecanismos principais. O primeiro é a revisão assistida de propostas, organizada em duas etapas: validação regulamentar e melhoria da proposta. A validação regulamentar analisa a proposta face às regras do processo participativo e devolve uma explicação clara quando são identificados possíveis problemas. A etapa de melhoria analisa o título, a descrição e os campos dinâmicos da proposta, sugerindo alterações que possam melhorar a clareza, estrutura, coerência e adequação ao processo. As sugestões são sempre apresentadas ao cidadão, que pode aceitá-las ou rejeitá-las antes da submissão final.

O segundo mecanismo é a convergência semântica, cujo objetivo é identificar propostas semelhantes ou complementares antes da submissão. Para isso, a solução utiliza *embeddings*, pesquisa vetorial e verificação com modelos de linguagem, permitindo apresentar ao cidadão propostas relacionadas já existentes no mesmo processo participativo. Este mecanismo pretende reduzir a fragmentação de ideias e incentivar a reflexão sobre possíveis relações entre propostas.

Para apoiar estes mecanismos, foram integradas estratégias de *Retrieval-Augmented Generation*, bases de dados vetoriais, extração de regras a partir de documentos regulamentares, utilização de propostas históricas e adaptação de modelos através de *Low-Rank Adaptation*. Foram também construídos diferentes conjuntos de dados para desenvolvimento, demonstração, treino e avaliação, incluindo propostas históricas, propostas sintéticas de teste, exemplos de reescrita e exemplos de instrução para adaptação do modelo.

A avaliação combinou testes automáticos e avaliação por especialistas. Os testes automáticos compararam diferentes modelos e configurações, incluindo modelos base, configurações com recuperação de contexto e um modelo adaptado com *Low-Rank Adaptation*. Os resultados mostraram que a combinação de conhecimento contextual e adaptação do modelo melhora o desempenho na validação regulamentar e na revisão de propostas. Complementarmente, foram realizados testes aos requisitos não funcionais, avaliando aspetos como o tempo de resposta, a qualidade e manutenibilidade do código, a segurança das dependências e a usabilidade da interface Web, cujos resultados demonstraram que o protótipo satisfaz os requisitos definidos para este trabalho. A avaliação por especialistas envolveu profissionais com experiência em processos participativos, que analisaram a utilidade, clareza, adequação e potencial de adoção da solução. Os resultados foram positivos, especialmente na melhoria da clareza e estrutura das propostas, bem como na transparência das explicações apresentadas ao cidadão.

Os resultados obtidos indicam que mecanismos inteligentes baseados em Processamento de Linguagem Natural e Modelos de Linguagem de Grande Dimensão podem apoiar a escrita contextual de propostas, promover maior consistência estrutural e contribuir para a identificação antecipada de ideias relacionadas em processos participativos digitais. Embora o protótipo não constitua uma plataforma pronta para produção, demonstra a viabilidade da abordagem e fornece uma base para futuras integrações no ecossistema CoParticipation e na *framework* EMPATIA.

Palavras-chave: Participação Eletrónica, Processamento de Linguagem Natural, Modelos de Linguagem de Grande Dimensão, Apoio à Escrita, Convergência Semântica, Inteligência Artificial.

Abstract

Digital participatory processes allow citizens to contribute to public decision-making through online platforms. However, many proposals are submitted with limited clarity, incomplete information, or weak alignment with process rules. In addition, similar or complementary ideas are often submitted separately, increasing fragmentation and reducing opportunities for collaboration.

This dissertation addresses these challenges by designing, implementing, and evaluating a proof of concept for AI-assisted proposal writing and semantic convergence in digital participatory processes. The work was developed within the CoParticipation project and aligned with the EMPATIA framework. The prototype combines a Laravel and Livewire web platform with a FastAPI-based AI service responsible for regulatory validation, proposal improvement, document ingestion, retrieval, semantic convergence, and local language model interaction.

The solution uses Natural Language Processing, Large Language Models, Retrieval-Augmented Generation, embeddings, vector retrieval, and LoRA-based model adaptation. Its main functionalities are a two-stage proposal review workflow and a semantic convergence mechanism that identifies similar or complementary proposals before submission.

The evaluation combined automatic tests, non-functional quality assessment, and expert feedback. Results show that contextual knowledge and model adaptation improve regulatory validation and proposal review, while the prototype meets the defined quality requirements. Experts confirmed the usefulness and adoption potential of the solution. Overall, the approach enhances proposal clarity, supports regulatory awareness, identifies related proposals, and provides effective contextual assistance without compromising citizen control or administrative responsibility.

Keywords: Electronic Participation, Natural Language Processing, Large Language Models, Writing Support, Semantic Convergence, Artificial Intelligence.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Goreti Marreiros, for her guidance, availability, and support throughout the development of this dissertation. Her feedback and orientation were essential in helping me structure the research work and improve the quality of this document.

I would also like to thank Luís Cordeiro, my supervisor at OneSource, for the opportunity to develop this work within the context of the CoParticipation project and for the trust, support, and professional guidance provided throughout this process.

I am also grateful to the experts who dedicated their personal time to evaluate the proposed solution: Filipe Casimiro, Timóteo Barrocas, Simone Júlio, and Vânia Martins. Their experience in participatory processes and their feedback were extremely valuable for assessing the practical relevance of the developed prototype.

Finally, I would like to thank everyone who, directly or indirectly, supported me during this Master's journey. Their encouragement, advice, and patience contributed to the completion of this work.

Table of Contents

1	Introduction	1
1.1	Problem Background.....	1
1.1.1	Context of the CoParticipation Project	2
1.1.2	The EMPATIA Framework.....	3
1.2	Research Objectives	4
1.2.1	Main Objective	4
1.2.2	Goals.....	4
1.2.3	Expected Contributions	4
1.3	Ethical and Professional Considerations	5
1.4	Methodology Overview	5
1.5	Risk Management	6
1.6	Document Structure	7
2	State of the Art.....	9
2.1	Research Methodology	9
2.2	First Research Question.....	10
2.2.1	Protocol Definition and Research Question.....	10
2.2.2	Search Strategy and Study Identification	10
2.2.3	Screening and Study Selection	10
2.2.4	Study Classification	11
2.2.5	Data Extraction and Aggregation	12
2.2.6	Results Analysis and Critical Synthesis	12
2.3	Second Research Question.....	14
2.3.1	Protocol Definition and Research Question.....	14
2.3.2	Search Strategy and Study Identification	14
2.3.3	Screening and Study Selection	14
2.3.4	Study Classification	15
2.3.5	Data Extraction and Aggregation	16
2.3.6	Results Analysis and Critical Synthesis	16
3	Analysis	19
3.1	Proposed Solution.....	19
3.2	Domain Model	20
3.3	Requirements Engineering	21
3.3.1	Functional Requirements	22
3.3.2	Non-functional Requirements	25
3.3.3	Use Cases.....	26
4	Design	31
4.1	Architectural Model	31

4.2	Technological Stack	32
4.3	C4 Diagrams	33
4.3.1	Context Diagram	33
4.3.2	Container Diagram	34
4.3.3	Component Diagram	36
4.3.4	Deployment Diagram	37
4.4	Sequence Diagrams	37
4.4.1	Proposal Review Workflow	38
4.4.2	Semantic Convergence Workflow	38
4.4.3	Document Ingestion Workflow	38
4.5	AI Service Design	38
4.6	Knowledge Management Design	39
4.7	Proposal Review and Semantic Convergence Design	40
5	Proof of Concept Implementation	43
5.1	Implementation Overview	43
5.2	Web Platform Implementation	43
5.2.1	Participatory Process Configuration	44
5.2.2	Dynamic Proposal Form	45
5.3	Integration between Laravel and the AI Service	46
5.4	AI Service Implementation	47
5.4.1	API Request Processing	47
5.4.2	Prompt Engineering and Structured Output Control	48
5.4.3	Integration with Ollama	50
5.5	Proposal Review Implementation	51
5.5.1	Regulatory Validation	51
5.5.2	Proposal Improvement	52
5.6	Semantic Convergence Implementation	53
5.7	Embeddings and Retrieval Implementation	55
5.8	Document Ingestion Implementation	56
5.9	Dataset Construction	58
5.9.1	Historical Proposal Dataset	58
5.9.2	Process Seed Datasets	59
5.9.3	Proposal Rewrite Dataset	59
5.9.4	LoRA Training Dataset	60
5.10	Model Adaptation and Training	60
5.10.1	Preliminary Training Experiments	61
5.10.2	Final LoRA Training Strategy	61
5.11	Robustness and Error Handling	62
6	Experimental Evaluation and Discussion	65
6.1	Evaluation Objectives	65

6.2	Research Hypotheses	66
6.3	Evaluation Methodology.....	66
6.3.1	Evaluation Design	66
6.3.2	Evaluation Datasets and Metrics	67
6.3.3	Expert Evaluation	68
6.4	Automated Evaluation Results	69
6.4.1	Proposal Review and Regulatory Validation	69
6.4.2	Semantic Convergence Evaluation	70
6.4.3	RAG and LoRA Comparative Evaluation	71
6.4.4	Non-Functional Requirements Evaluation.....	72
6.5	Expert Evaluation Results.....	76
6.5.1	Questionnaire Results	76
6.5.2	Blind A/B Testing Results	79
6.6	Discussion	80
6.6.1	Analysis of Research Hypotheses.....	80
6.6.2	Main Findings	80
6.7	Threats to Validity.....	81
7	Conclusions and Future Work	83
7.1	Research Overview	83
7.2	Main Contributions.....	83
7.3	Achievement of the Main Objective	84
7.4	Limitations.....	85
7.5	Future Work	85
7.6	Final Remarks	86
	References.....	87
	Appendix A.....	91
	Appendix B.....	92
	Appendix C.....	93
	Appendix D.....	94
	Appendix E.....	95
	Appendix F.....	96
	Appendix G	97
	Appendix H	98

List of Figures

Figure 1 - Risk Matrix (Burstein and Zuckerman, 2024)	7
Figure 2 - PRISMA Diagram for the First Research Question.....	11
Figure 3 - Full Fine-Tuning vs LoRA Fine-Tuning vs RAG (Chawla, 2023)	13
Figure 4 - PRISMA Diagram for the Second Research Question	15
Figure 5 - SBERT Architecture (Bhaskar, 2023)	17
Figure 6 - Positioning within the CoParticipation Ecosystem.....	20
Figure 7 - Domain Model.....	21
Figure 8 - Use Case Diagram.....	27
Figure 9 - Context Diagram.....	34
Figure 10 - Container Diagram.....	35
Figure 11 - Component Diagram.....	36
Figure 12 - Deployment Diagram	37
Figure 13 - Internal Organisation of the AI Service	39
Figure 14 - Knowledge Management and Retrieval Strategy.....	40
Figure 15 - Proposal Review and Semantic Convergence Workflow.....	41
Figure 16 - Simplified Process Schema Example	44
Figure 17 - Dynamic Proposal Creation Form with Process-Specific Fields.....	45
Figure 18 - Simplified Laravel Communication Pattern	46
Figure 19 - Regulatory Validation Response Example	48
Figure 20 - Improvement Response Example.....	48
Figure 21 - First Part of Regulatory Validation Prompt.....	50
Figure 22 - Ollama List Command	51
Figure 23 - Regulatory Warning Displayed to the Citizen	52
Figure 24 - Inline Proposal Improvement Suggestions with Accept and Reject Options.....	53
Figure 25 - Semantic Convergence Workflow.....	54
Figure 26 - Semantic Convergence Modal with Similar and Complementary Proposals.....	55
Figure 27 - Function Responsible for Splitting Documents into Overlapping Text Chunks	56
Figure 28 - Document Ingestion Workflow	57
Figure 29 - Historical Dataset Structure Example	58
Figure 30 - Portuguese Process Seed Dataset Structure Example	59
Figure 31 - Proposal Rewrite Dataset Structure Example.....	59
Figure 32 - LoRA Training Dataset Structure Example.....	60
Figure 33 - LoRA Training Output Example.....	62
Figure 34 - Demonstration and Evaluation Session with Domain Experts.....	69
Figure 35 - Response Time Evaluation Results	73
Figure 36 - Lighthouse Desktop Results	75
Figure 37 - Questionnaire Answers	77
Figure 38 - Global Evaluation Questionnaire Answers	78
Figure 39 - A/B Testing Question Example.....	79
Figure 40 - Risk Management Matrix.....	93

Figure 41 - Proposal Review Sequence Diagram	94
Figure 42 - Semantic Convergence Sequence Diagram.....	95
Figure 43 - Document Ingestion Sequence Diagram.....	96
Figure 44 - Runtime Interaction of the Proof-of-Concept.....	97
Figure 45 - Mobile-friendly layouts.....	98

List of Tables

Table 1 - Classification of Studies for the First Research Question	12
Table 2 - Classification of Studies from the Second Research Question.....	16
Table 3 - Participatory Process Management Functional Requirements.....	22
Table 4 - Knowledge and Document Management Functional Requirements.....	22
Table 5 - Proposal Drafting Support Functional Requirements	23
Table 6 - Semantic Convergence Support Functional Requirements	24
Table 7 - Model and Evaluation Functional Requirements	25
Table 8 - Non-functional Requirements.....	26
Table 9 - UC01: Register Participatory Process.....	27
Table 10 - UC02: Ingest Process Documents	28
Table 11 - UC03: Draft Proposal.....	28
Table 12 - UC04: Request Proposal Review.....	29
Table 13 - UC05: Accept/Reject Improvement Suggestions.....	29
Table 14 - UC06: Choose Proposal Convergence Suggestion	30
Table 15 - UC07: Submit Proposal.....	30
Table 16 - Technological Stack	32
Table 17 - Interactions between Laravel and FastAPI	46
Table 18 - Main Processing Pipelines.....	47
Table 19 - Prompt Generic Structure.....	49
Table 20 - Validation Prompt Structure	51
Table 21 - Types of Information Stored in ChromaDB.....	56
Table 22 - Datasets Created.....	58
Table 23 - Main LoRA Configuration Parameters.....	61
Table 24 - Automatic Evaluation Test Groups	67
Table 25 - Comparative Evaluation Configurations.....	67
Table 26 - Expert Profiles.....	68
Table 27 - Proposal Review Automated Evaluation Results.....	70
Table 28 - Semantic Convergence Automated Evaluation Results	71
Table 29 - RAG and LoRA Comparative Evaluation Results.....	72
Table 30 - Code Quality and Maintainability Results.....	74
Table 31 - Package Vulnerabilities.....	74
Table 32 - Mapping Evaluation to Non-Functional Requirements.....	76
Table 33 - Average Scores per Category	77
Table 34 - Most Selected Improvements	78
Table 35 - Blind A/B Testing Results	79
Table 36 - Data Aggregation of Studies for the First Research Question.....	91
Table 37 - Data Aggregation of Studies for the Second Research Question	92

Acronyms

A/B	A/B Testing
AI	Artificial Intelligence
AlaaS	Artificial Intelligence as a Service
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
C4	Context, Containers, Components and Code
CA-RAG	Context-Aware Retrieval-Augmented Generation
CTG	Controllable Text Generation
ERDF	European Regional Development Fund
EU	European Union
F1	F1-score
FAIR	Findable, Accessible, Interoperable, Reusable
FR	Functional Requirement
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IEEE	Institute of Electrical and Electronics Engineers
ISEP	Instituto Superior de Engenharia do Porto
JSON	JavaScript Object Notation
JSONL	JSON Lines
LLaMA	Large Language Model Meta AI
LLM	Large Language Model
LoRA	Low-Rank Adaptation

ML	Machine Learning
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain
NFR	Non-functional Requirement
NLP	Natural Language Processing
OECD	Organization for Economic Co-operation and Development
PEFT	Parameter-Efficient Fine-Tuning
PICOCS	Population, Intervention, Comparison, Outcomes, Context, Study type
PLM	Pre-trained Language Model
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QA	Question Answering
RAG	Retrieval-Augmented Generation
REST	Representational State Transfer
SBERT	Sentence-BERT
TRL	Transformer Reinforcement Learning
UC	Use Case
UN	United Nations
VRAM	Video Random Access Memory

1 Introduction

This chapter describes the foundation of this dissertation by defining the context, objectives, and challenges that this investigation intends to address throughout the following sections. The chapter begins with the background of the problem, presenting the concept of electronic participation and its current limitations, particularly in processes that include a phase for citizen proposal formulation. Subsequently, the main objective and the specific goals are defined alongside the ethical and professional considerations that ensure the study's integrity. It also summarises the risk management approach adopted during planning to anticipate potential constraints, concluding with a detailed description of the document structure.

1.1 Problem Background

Electronic participation, commonly known as e-participation, enhances participatory processes through Information and Communication Technologies (ICT) and constitutes a central element of modern electronic democracy. This concept utilises specialised mechanisms to manage digital democratic practices while valuing the benefits of technology for both governments and citizens within political and governance processes (Mariani *et al.*, 2025). Digital platforms are currently used to manage various participatory processes, including participatory budgeting, voting, public consultations, and the collaborative formulation of public policies. The second decade of the 21st century recorded rapid growth in these mechanisms, moving beyond simple information provision and expanding significantly between the years 2012 and 2018 (UN, 2020). This expansion allows communities to influence public decisions more directly and promotes collaboration between institutions and the community, making deliberative processes a consolidated trend (OECD, 2025).

Despite the increased technological availability, the effectiveness of digital participation platforms remains limited by structural barriers that obstruct both the inclusion of diverse groups and the overall quality of participation. A digital divide exists that goes beyond simple infrastructure access, involving significant disparities in the digital literacy and specialised skills

required from the end users. Engaging in advanced forms of deliberation requires specific capacities to analyse complex political proposals and provide structured contributions that can be effectively integrated into governance (UN, 2020). Many current platforms function as simple asynchronous channels that fail to promote meaningful dialogues or encourage the collective reflection necessary for robust democracy (Borchers, Soroko, *et al.*, 2024). Furthermore, the absence of active moderation creates a gap in feedback during collaboration, preventing participants from identifying if their contributions are inconsistent or ignore technical regulations (Borchers, Soroko, *et al.*, 2024).

The quality and clarity of submitted proposals represent a persistent challenge for public administration, as citizens frequently present unrealistic requirements or describe their needs through abstract visions. This absence of clear detail makes it more difficult to carry out a thorough technical analysis of the proposals and creates obstacles for the municipal technicians responsible for evaluating the feasibility of each submission (Borchers, Soroko, *et al.*, 2024; OECD, 2025). These differences in how participants express themselves and in the level of detail they provide arise from the group's diverse demographic and educational backgrounds (Borchers, Soroko, *et al.*, 2024). When technicians must process hundreds or thousands of submissions, the lack of structure and text ambiguity makes the evaluation process unmanageable (Shortall *et al.*, 2022; OECD, 2025). Consequently, poorly formulated proposals are often excluded, which reduces project acceptance and generates significant frustration among the contributing citizens (Borchers, Soroko, *et al.*, 2024).

Finally, idea fragmentation and the lack of convergence undermine the effectiveness of mass participatory processes, as digital platforms often fail to identify redundant proposals sharing the same objectives (Borchers, Soroko, *et al.*, 2024). This dispersion of similar contributions dilutes public support for specific initiatives and prevents the construction of an informed consensus among the various stakeholders (Shortall *et al.*, 2022). Without effective convergence mechanisms, valuable opportunities for co-creating robust solutions are lost, and participants tend to reproduce existing ideas instead of developing new ones (Borchers, Soroko, *et al.*, 2024; Mariani *et al.*, 2025). The difficulty in scaling moderation to handle high data volumes aggravates the lack of interaction and prevents a truly collaborative and interactive debate (Shortall *et al.*, 2022).

Given this context, the research problem focuses on the absence of intelligent mechanisms for contextual writing support and proposal convergence, which compromises textual quality and limits citizen collaboration.

1.1.1 Context of the CoParticipation Project

This dissertation is developed within the CoParticipation project (CoParticipation, 2026), an innovation initiative funded by the COMPETE2030 Programme (Compete 2030, 2026) and co-financed by the European Union through the European Regional Development Fund (ERDF). The project aims to advance digital citizen participation by combining modern web technologies,

Artificial Intelligence (AI), and blockchain to create more transparent, inclusive, and efficient participatory processes.

CoParticipation addresses several key challenges found in current electronic participation platforms:

- **Proposal Quality:** The difficulty citizens face when trying to draft proposals.
- **Administrative Overload:** The heavy workload of process managers who must analyse and evaluate large volumes of public contributions.
- **Trust and Transparency:** The ongoing need to strengthen integrity and trust in digital democratic systems.

To tackle these issues, the project explores intelligent mechanisms that assist citizens during proposal creation, support decision-makers with AI-powered analysis tools, and secure participation records. The project is led by OneSource (OneSource, 2026), in collaboration with the partners Associação Oficina (*Oficina*, 2026) and Instituto Pedro Nunes (*Instituto Pedro Nunes*, 2026). The work in this dissertation contributes directly to these goals by developing intelligent contextual assistance for proposal writing and semantic convergence.

1.1.2 The EMPATIA Framework

The technological foundation of the CoParticipation platform is the EMPATIA framework (EMPATIA, 2026). This open-source electronic participation platform was originally created under the European Union Horizon 2020 research programme, bringing together universities, research centres, and technology companies to promote inclusive and effective digital democracy.

EMPATIA was designed as a flexible, modular framework that allows public institutions to deploy and manage a wide variety of citizen engagement initiatives, such as:

- **Participatory Budgeting:** Enabling citizens to directly decide on the allocation of public budgets.
- **Public Consultations:** Gathering community feedback on proposed local regulations and institutional policies.
- **Civic Engagement Programmes:** Supporting continuous dialogue and collaborative idea collection between citizens and municipalities.

Over the years, EMPATIA has demonstrated its scalability across multiple national and international initiatives, inspiring gamified variants like EMPAVILLE (EMPAVILLE, 2026).

Within the CoParticipation project, EMPATIA serves as the base software layer. This dissertation extends the core framework by integrating an AI service layer designed to provide real-time writing support and preventive similarity detection directly into the proposal submission workflow.

1.2 Research Objectives

This section presents the main objective of the dissertation together with the specific goals established to achieve it. These goals define the methodological path followed throughout the research, guiding the analysis, design, implementation, and evaluation of the proposed solution.

1.2.1 Main Objective

The main objective of this dissertation is to design, develop, and evaluate a proof-of-concept solution capable of supporting citizens during proposal creation in digital participatory processes through intelligent mechanisms based on Natural Language Processing (NLP) and Large Language Models (LLMs). More specifically, the work investigates how these technologies can support contextual writing, promote structural standardisation, and foster semantic convergence of proposals while preserving the citizen's original intent and respecting the specific rules of each participatory process.

1.2.2 Goals

To achieve the main objective, the following goals were defined:

- **State-of-the-Art Review:** To conduct a literature review organised around two research questions, utilising scientific research methods and frameworks that will be detailed in the dedicated state-of-the-art chapter.
- **Prototype Design and Development:** To design and develop a functional prototype as a proof of concept for the identified problems, applying knowledge acquired during the Master in Informatics Engineering and additional skills identified through skills management analysis.
- **Solution Evaluation:** To evaluate the effectiveness and usability of the developed prototype using appropriate metrics and assessment approaches.

1.2.3 Expected Contributions

The completion of this project is intended to improve digital participatory processes across several dimensions. Firstly, the proposed mechanisms are expected to enhance the quality and clarity of proposals by supporting citizens in formulating more structured and consistent proposals and by analysing them against the participatory process rules. Secondly, the introduction of semantic analysis and similarity detection aims to reduce idea fragmentation, thereby promoting convergence and collaboration among participants with shared goals.

From a scientific perspective, this work seeks to contribute scientifically and technologically by applying LLMs to a specific and relatively unexplored domain, reinforcing existing knowledge on AI use for writing support and deliberation.

1.3 Ethical and Professional Considerations

This project involves the processing of textual data from previous participatory processes, which requires a rigorous approach in compliance with the General Data Protection Regulation (GDPR) (EU, 2016) and the FAIR principles (Findable, Accessible, Interoperable, Reusable) (Wilkinson et al., 2016), ensuring both the lawful and transparent handling of personal data and the adoption of good data-management practices that promote openness, reuse, and long-term accessibility.

From an academic standpoint, the investigation follows the Code of Ethical Conduct of the Polytechnic Institute of Porto (Portugal, 2020), ensuring respect for scientific integrity norms and the correct referencing of all ideas and sources.

In a broader professional context, the work adopts the principles of the IEEE Code of Ethics (IEEE, 2023), which promote honesty, integrity, and transparency in the presentation of information and the application of technical skills. Additionally, the study observes the European Code of Conduct for Research Integrity (ALLEA, 2023), reinforcing the principles of reliability, accountability, and fairness throughout the scientific activity.

In the specific domain of Artificial Intelligence, the research ensures compliance with the European Union Artificial Intelligence Act (AI Act) (EU, 2024). This regulation establishes transparency requirements, risk assessment protocols, and supervision mechanisms for AI systems, guaranteeing that the development and application of the proposed solution respect both ethical and legal principles.

Artificial Intelligence tools were used in this dissertation in three ways, in accordance with institutional guidelines and scientific transparency principles:

- **Prototype Functionality:** The developed system integrates local LLMs, executed through Ollama, to provide the core intelligent features.
- **Software Development Process:** Generative AI assistants were used during implementation to support code writing, refactoring, and debugging tasks.
- **Thesis Document Revision:** AI-based language tools were employed for grammar correction, syntactic refinement, and consistency checks in English (United Kingdom) during the final editing phase of this document.

1.4 Methodology Overview

The work methodology adopted for this project is Design and Creation. This approach involves an iterative movement between the development of an artifact, in this case, the writing and convergence support prototype, and its evaluation. This allows for learning through the process of building solutions to identified practical problems (Oates, 2006). This methodology is frequently framed within the Design Science Research paradigm, which argues that scientific

knowledge is achieved through the design and implementation of innovative artifacts that solve relevant problems (Hevner *et al.*, 2004).

The work is grounded in the areas of NLP, Software Engineering, and Data Science. The methodology is structured into sequential and iterative phases to ensure feasibility and alignment with the proposed objectives:

1. **Research and Analysis:** Conducting a systematic literature review to establish the state of the art in fine-tuning techniques, RAG mechanisms, and intent analysis. This phase establishes the necessary theoretical foundation for development.
2. **Design and Architecture:** Defining the system's modular structure. Here, the identified requirements are translated into architectural diagrams, data flows, and technical specifications for the models to be used.
3. **Build:** Practical implementation of the functional prototype. In this phase, software engineering techniques are applied to integrate NLP modules, using datasets from real participatory processes to feed and adjust the system's behaviour.
4. **Evaluation:** The utility and performance of the artifact are validated according to previously stipulated metrics.

1.5 Risk Management

Risk management is a systematic process of identifying, analysing, and mitigating uncertain events that could compromise project objectives. Risk management frameworks allow for the anticipation of problems, assessment of their probability and impact, and the definition of appropriate responses in a structured manner (Aven, 2016). Adopting this approach contributes to greater predictability, transparency, and project control, reducing the likelihood of significant deviations from the initial planning.

In the context of this project, risk management is particularly relevant due to the reliance on external data, the technical complexity associated with integrating AI mechanisms, and the need for legal compliance and stakeholder acceptance. Identifying risks early helps align technical and scientific planning with realistic mitigation strategies suited to the scope of the work.

To support this process, each risk is evaluated using a qualitative scale from 1 to 5 for Likelihood and from 1 to 5 for Consequence/Impact, forming the risk matrix shown in Figure 1.

		Consequence				
		Negligible 1	Minor 2	Moderate 3	Major 4	Catastrophic 5
Likelihood	5 Almost certain	Moderate 5	High 10	Extreme 15	Extreme 20	Extreme 25
	4 Likely	Moderate 4	High 8	High 12	Extreme 16	Extreme 20
	3 Possible	Low 3	Moderate 6	High 9	High 12	Extreme 15
	2 Unlikely	Low 2	Moderate 4	Moderate 6	High 8	High 10
	1 Rare	Low 1	Low 2	Low 3	Moderate 4	Moderate 5

Figure 1 - Risk Matrix (Burstein and Zuckerman, 2024)

Based on the project’s context and objectives, five primary risks were identified and analysed, considering their causes, potential effects, likelihood, impact, and corresponding mitigation strategies. The most critical risk concerns the higher-than-expected technical complexity involved in integrating LLMs, NLP techniques, vector databases, and the proposed software architecture. This complexity could affect both the development schedule and the overall functionality of the proof of concept. Other relevant risks include the quality and availability of open datasets required for model adaptation and evaluation, the potential unavailability or modification of external APIs and third-party services, compliance with GDPR requirements, and the acceptance of the proposed solution by domain experts and stakeholders.

To minimise these risks, several mitigation strategies were defined during the planning phase. These include prioritising incremental development, maintaining fallback mechanisms for external dependencies, validating legal requirements related to data usage, employing alternative public datasets when necessary, and involving domain experts throughout the development and evaluation phases. The complete qualitative risk assessment is presented in Appendix C.

1.6 Document Structure

The present document is organised into seven chapters.

Chapter 1 introduces the investigation context, presents the CoParticipation project and the EMPATIA framework, defines the research main objective, goals and expected contributions, and discusses the ethical and professional considerations that guide the study.

Chapter 2 is dedicated to the state of the art, beginning with a description of the research methodology and followed by a literature review.

Chapter 3 analyses the proposed solution by defining its scope, outlining the domain concepts, and establishing the functional and non-functional requirements that guide the system's behaviour.

Chapter 4 presents the overall design of the solution, describing the architectural approach, the main technological decisions, and the modelling views that structure both the platform and the AI-driven components.

Chapter 5 details the implementation of the proof of concept, explaining how the different subsystems were developed and integrated, from the web application and AI service to the supporting data, models, and runtime environment.

Chapter 6 presents the experimental evaluation of the solution, outlining the methodology adopted, the different forms of assessment, and the main findings. It also discusses the implications of the results and the factors that may influence their validity.

Finally, Chapter 7 summarises the key contributions of the dissertation, highlights the main limitations of the work and concludes with directions for future research and development.

2 State of the Art

This chapter aims to analyse the existing scientific knowledge relevant to the proposed investigation through a literature review. The objective is to identify, organize, and discuss the most significant works related to the application of intelligent mechanisms based on NLP and LLMs for supporting writing and proposal convergence. This analysis allows for the theoretical framing of the research problem while identifying gaps and opportunities that justify the relevance of the proposed work within the field of digital democracy.

2.1 Research Methodology

The literature review follows a systematic methodology to ensure rigour, transparency, and reproducibility in the identification, selection, and analysis of relevant studies. The strategy is based on a Systematic Mapping Study, which allows the researcher to identify, categorize, and map the available evidence regarding a specific research domain. This approach is ideal for obtaining a global overview of the existing research while revealing the distribution of studies and identifying potential gaps in the literature (Petersen, Vakkalanka and Kuzniarz, 2015).

The review process begins with the formulation of research questions using the PICOCS framework, which considers Population, Intervention, Comparison, Outcomes, Context, and Study type. Defining each component of this framework allows for a more rigorous orientation of the search criteria and facilitates the construction of search strings for identifying relevant studies in digital libraries (Hosseini *et al.*, 2024).

To ensure transparency in article selection, the study utilises the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) flow. This model represents the information collection path through a flow diagram, documenting the process from initial identification to final inclusion. The PRISMA flow ensures that the whole process remains both auditable and transparent (Page *et al.*, 2021).

2.2 First Research Question

2.2.1 Protocol Definition and Research Question

The research question analysed in this section is as follows: "How can mechanisms based on NLP, LLMs, and other intelligent techniques be effectively adapted and applied to specific domains, namely supporting proposal writing in digital participatory processes?"

The primary objective of this review is to identify and analyse studies that address the adaptation of NLP and LLM techniques to specific contexts. Special attention is given to strategies for model contextualisation, personalization, and their integration into digital participation platforms or functionally similar domains.

2.2.2 Search Strategy and Study Identification

The identification of relevant studies was conducted through targeted searches in the IEEE Xplore database. The construction of the search strings was supported by the PICOCS model, which allowed for the structured organisation of the main elements of the research question:

- **Population:** Studies regarding digital platforms or specific NLP application domains.
- **Intervention:** The use of NLP, LLMs, or intelligent mechanisms for writing support or domain adaptation.
- **Comparison:** Not applicable, as this is a Mapping Study.
- **Outcomes:** Improvements in textual quality, contextualisation, domain adaptation, or writing support.
- **Context:** Digital participatory processes or functionally comparable contexts.
- **Study:** Empirical studies, methodological proposals, case studies, or relevant reviews.

Based on this framework, the following search strings were defined: ("Natural Language Processing" OR "Large Language Models" OR LLM) AND (domain-specific OR customization OR adaptation) AND (text writing OR semantic similarity).

2.2.3 Screening and Study Selection

The results obtained underwent a screening process consisting of several phases: removal of duplicates, title and abstract analysis, and full-text analysis when necessary. This process was documented through a PRISMA flow diagram, following the inclusion and exclusion criteria defined below:

Inclusion Criteria:

- Peer-reviewed scientific publications.
- Articles published between 2021 and 2025.

- Studies with direct relevance to NLP, LLMs, or intelligent mechanisms applied to specific domains.
- Methodological contributions related to the adaptation, contextualisation, or personalization of LLMs.
- Open access availability.
- Publications in the English language.

Exclusion Criteria:

- Non-scientific literature or works lacking methodological foundation.
- Opinion pieces, editorials, or incomplete articles.
- Studies without a clear link to writing support or domain adaptation.
- Publications in languages other than English.

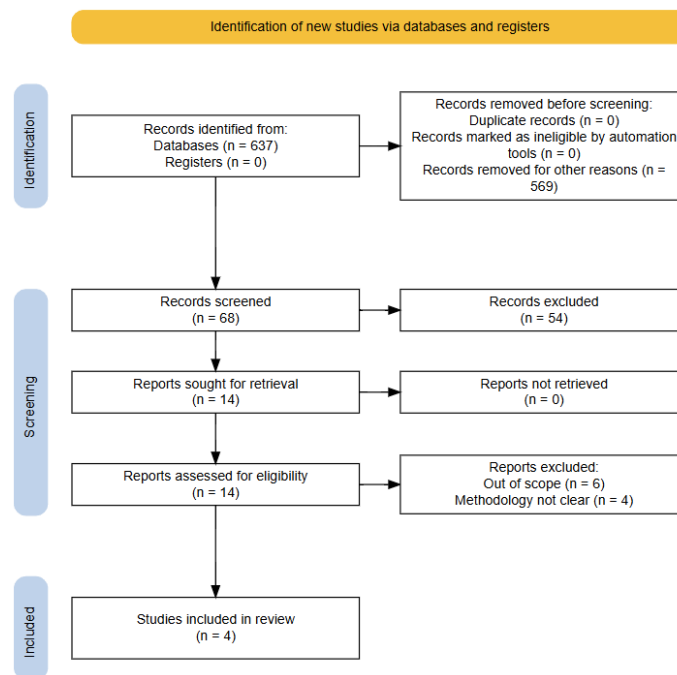


Figure 2 - PRISMA Diagram for the First Research Question

2.2.4 Study Classification

After the selection process, the studies were classified according to the following categories: type of technique utilized, domain adaptation strategy, and application domain.

Table 1 - Classification of Studies for the First Research Question

Study (Author, Year)	Technique Type	Adaptation Strategy	Domain
(Challa and Bright, 2025)	LLMs (LLaMA/GPT)	Low-Rank Adaptation (LoRA)	Domains with scarce resources
(Ding et al., 2023)	Large-scale PLMs	Delta-tuning (LoRA/Prefix-tuning)	Generic
(Budakoglu and Emekci, 2025)	LLMs	RAG, Fine-tuning	Quality Assurance
(Castillo-López, de Chalendar and Semmar, 2025)	Hybrid Systems (BERT+LLM)	Zero/Few-shot learning	Dialogues and Intent Recognition

2.2.5 Data Extraction and Aggregation

For each selected study, the most relevant data were extracted, including study objectives, technical approach, application context, reported results, and the limitations identified by the authors. The summarised results of this extraction process are available in Appendix A.

2.2.6 Results Analysis and Critical Synthesis

A review of the literature reveals that very few studies look at how LLMs can be adapted to participatory processes. For this reason, the search was broadened to include other similar domains that face comparable challenges.

The specialization of LLMs for domains with limited resources constitutes a recurring challenge, as full parameter fine-tuning involves high computational and financial costs (Challa and Bright, 2025). In this context, Parameter-Efficient Fine-Tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA), emerge as efficient solutions for adapting models to specific contexts. These approaches allow for the optimization of system performance without compromising economic feasibility, demonstrating that LLMs can be effectively specialised using a reduced number of trainable parameters (Ding *et al.*, 2023).

Contextual precision is another critical requirement identified in functionally similar areas. The use of Retrieval-Augmented Generation (RAG) allows systems to ground their responses in external knowledge bases, thereby reducing hallucinations and increasing the practical utility of generated answers (Budakoglu and Emekci, 2025). Comparative studies between RAG-based and fine-tuning approaches indicate that hybrid solutions tend to maximize semantic accuracy. This knowledge is highly transferable to the development of assistants capable of understanding the specific rules, norms, and constraints inherent to participatory processes.

Finally, literature regarding intent recognition in dialogues provides relevant contributions to this investigation. Combining Bidirectional Encoder Representations from Transformers (BERT) models with LLMs has shown significant improvements in detecting out-of-scope requests (Castillo-López, de Chalendar and Semmar, 2025). This functionality is essential to ensure that writing support remains focused on pertinent contributions that align with the overall goals of the participatory process.

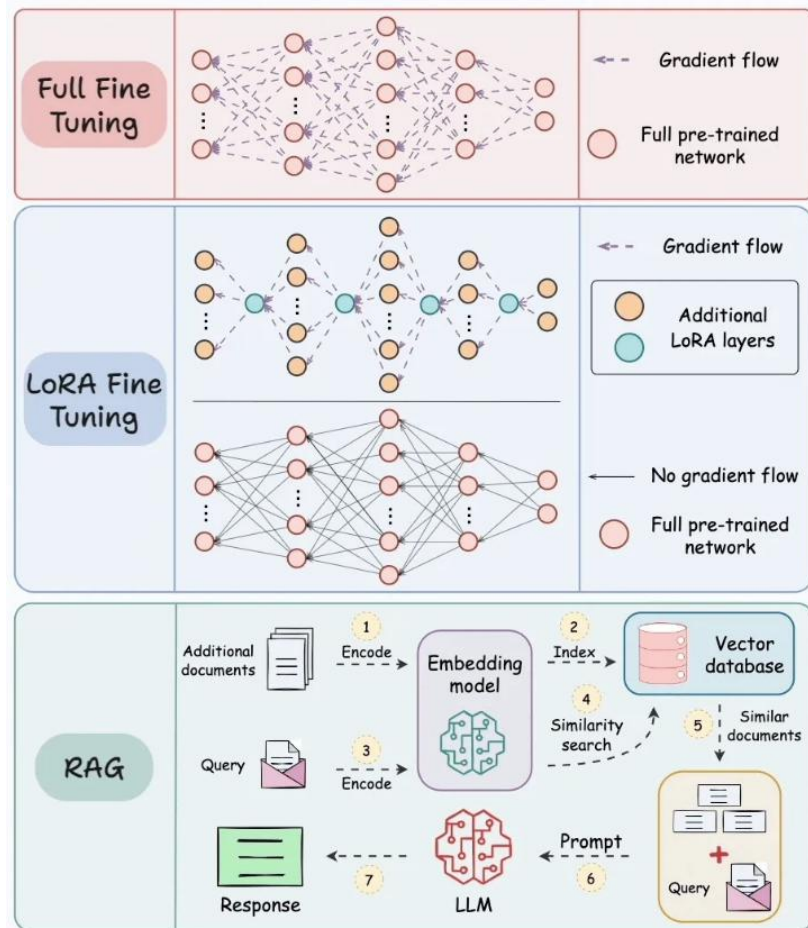


Figure 3 - Full Fine-Tuning vs LoRA Fine-Tuning vs RAG (Chawla, 2023)

2.3 Second Research Question

2.3.1 Protocol Definition and Research Question

The research question analysed in this section is: “Which approaches for AI-assisted writing support and semantic similarity detection are described in the literature, and how can they contribute to improving proposal quality and convergence in digital participatory processes?”

The objective of this review is to identify and analyse studies that evaluate AI-based writing support mechanisms and semantic similarity techniques applicable to identifying redundancies, clustering content, or promoting convergence between textual contributions. This analysis includes approaches from functionally similar contexts that demonstrate potential for adaptation to the research problem.

2.3.2 Search Strategy and Study Identification

The identification of studies was conducted through searches in the IEEE Xplore database. The construction of search strings was supported by the PICOCS model to systematically structure the main elements of the question:

- **Population:** Studies on assisted writing, collaborative systems, or digital platforms.
- **Intervention:** Use of AI, NLP or LLMs for writing support or semantic analysis.
- **Comparison:** Not applicable, as this is a Mapping Study.
- **Outcomes:** Improvement in textual quality, semantic similarity identification, redundancy reduction, or convergence promotion.
- **Context:** Digital participatory processes or comparable collaborative writing contexts.
- **Study:** Empirical studies, methodological proposals, case studies, or relevant reviews.

Based on this framework, the following search strings were defined: ("AI-assisted writing" OR "writing support systems" OR "text generation") AND ("semantic similarity" OR "text clustering" OR "content convergence") AND (collaborative OR electronic participation).

2.3.3 Screening and Study Selection

The results underwent a screening process involving duplicate removal, title and abstract analysis, and full-text review. This process was documented through a PRISMA flow diagram, following these criteria:

Inclusion Criteria:

- Peer-reviewed scientific publications published between 2021 and 2025.
- Studies addressing AI-assisted writing, NLP, or semantic similarity analysis.
- Works contributing to textual quality improvement or content convergence.
- Open access publications in English.

Exclusion Criteria:

- Non-scientific literature or works without methodological foundation.
- Opinion pieces, editorials, or incomplete articles.
- Studies without a clear link to assisted writing or semantic analysis.
- Publications in languages other than English.

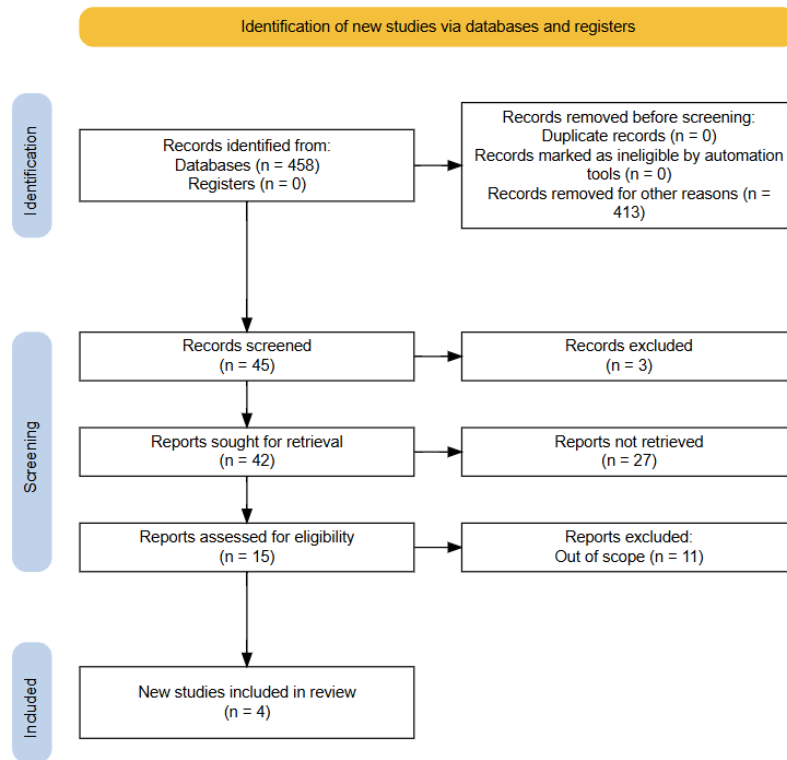


Figure 4 - PRISMA Diagram for the Second Research Question

2.3.4 Study Classification

Selected studies were organised into categories based on the type of writing support approach, the semantic analysis techniques used, and the application context.

Table 2 - Classification of Studies from the Second Research Question

Study (Author, Year)	Type of Writing Support	Semantic Analysis Techniques	Application Context
(Borchers, Cao, et al., 2024)	Support for analysis and structuring	Clustering and Summarization	E-Participation
(Alpdemir and Alpdemir, 2024)	Content generation	Sentence Transformers and Controllable Text Generation (CTG)	Article writing
(Collini et al., 2025)	Generation of reliable responses	Similarity Validation and Embeddings	RAG-based QA systems
(Arana-Catania et al., 2021)	Organisation of contributions	Sentiment Analysis and Topic Modelling	E-Participation

2.3.5 Data Extraction and Aggregation

For each study, data regarding objectives, technical approaches, results, and limitations were extracted and aggregated, with the full results available in Appendix B.

2.3.6 Results Analysis and Critical Synthesis

AI-assisted writing support tools have improved significantly, but their use in drafting proposals for digital participatory processes is still mostly unexplored. Most existing solutions focus on general content creation, which led this review to look at other related domains. This broader perspective helped show how writing support tools can improve text quality and encourage participants to produce more aligned and convergent contributions in digital citizenship contexts.

Controllable Text Generation (CTG) constitutes a relevant approach for supporting users in structuring ideas while preserving their personal style. Techniques such as CTG combined with sentence transformers, like Sentence-BERT (SBERT), ensure that generated text remains coherent with the initial topic and aligned with the proposal context (Alpdemir and Alpdemir, 2024). These approaches are particularly valuable for citizens without technical backgrounds, helping to reduce participation barriers. However, their application in digital governance platforms still requires scientific validation and formalized methodological proposals.

Content convergence is addressed in literature primarily through large-scale organisation and clustering mechanisms. Machine Learning (ML) systems allow for automated redundancy detection (Arana-Catania *et al.*, 2021), reducing the cognitive load on analysts and increasing the visibility of semantically similar proposals (Borchers, Cao, *et al.*, 2024). However, these approaches focus mainly on post-submission analysis phases rather than providing preventive

support during the drafting process. As a result, there are still no solutions that actively encourage convergence during the drafting of citizens' contributions.

The reliability of system-generated suggestions is ensured in recent literature through validation mechanisms based on semantic similarity. Approaches such as Context-Aware Retrieval-Augmented Generation (CA-RAG) help mitigate hallucinations by validating generated responses against relevant contextual information (Collini *et al.*, 2025). This type of validation is especially critical in participatory processes, where technical rigour, transparency, and credibility are determining factors for the acceptance and overall impact of citizen proposals.

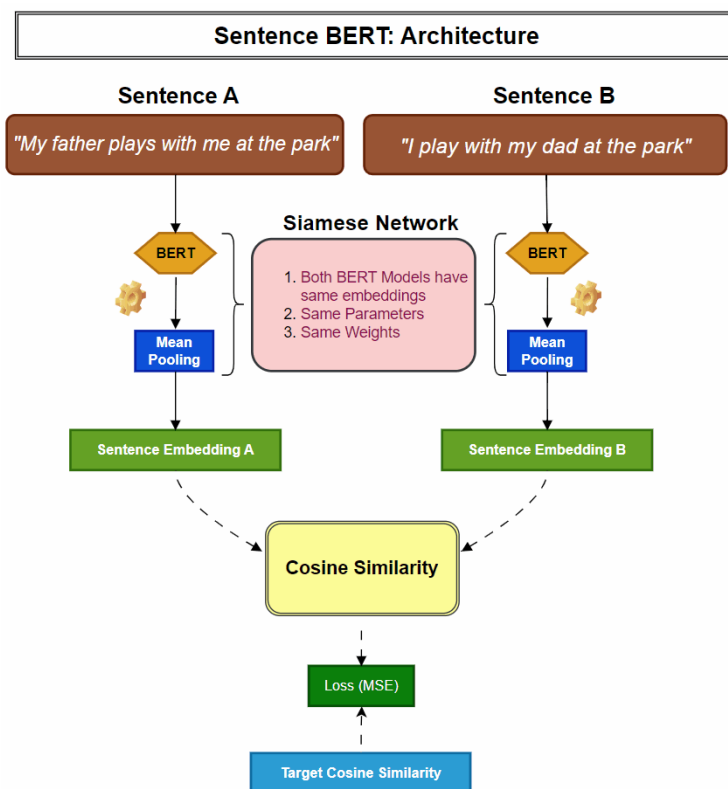


Figure 5 - SBERT Architecture (Bhaskar, 2023)

3 Analysis

The analysis phase defines the foundations of the solution. It clarifies what the system must support, identifies the main concepts in the domain, and specifies the requirements that guide the design, implementation, and evaluation work.

3.1 Proposed Solution

As described in Section 1.1, this work is part of the CoParticipation project, which aims to improve electronic participation platforms using intelligent tools. The contribution of this dissertation is to support citizens during proposal drafting, helping them write clearer and more complete proposals and encouraging the early convergence of similar or complementary ideas.

The proposed solution extends the existing EMPATIA framework, which continues to provide the core functionalities required to manage participatory processes, including process configuration, proposal submission, user interaction and platform administration. Instead of replacing these capabilities, this dissertation adds an AI-based assistance layer that works alongside EMPATIA and supports citizens during the proposal drafting stage.

The solution consists of a specialised AI module that provides two capabilities: proposal review and semantic convergence. The proposal-review component checks draft proposals against the rules and context of the participatory process and provides suggestions to improve clarity, completeness, and compliance. The semantic-convergence component identifies proposals that are similar or complementary, helping citizens see related ideas and encouraging collaboration before submission.

In the CoParticipation ecosystem, these features are implemented as independent services integrated into a shared Artificial Intelligence as a Service (AIaaS) layer. This setup allows different AI modules to evolve separately while relying on common elements such as language models, governance mechanisms, and processing pipelines. The module developed in this

dissertation fits into this structure and can operate alongside other intelligent services developed within the project. Figure 6 shows how the module connects to EMPATIA, the AIaaS layer, the Guardrails component, and the shared language model infrastructure.

A key element of this ecosystem is the Guardrails layer, which applies safety and governance rules across all AI services. It validates requests, filters inappropriate interactions, and applies safeguards before and after model execution. Because this layer is part of the shared infrastructure, it falls outside the scope of this dissertation and is not reimplemented in the prototype.

The scope of this work is therefore limited to the design, development, and evaluation of the AI module for proposal review and semantic convergence. The module interacts with the shared language model infrastructure and with process-specific knowledge sources and exposes its functionalities to EMPATIA through an API. This ensures integration within the broader CoParticipation ecosystem while keeping the focus on the specific research contribution of the dissertation.

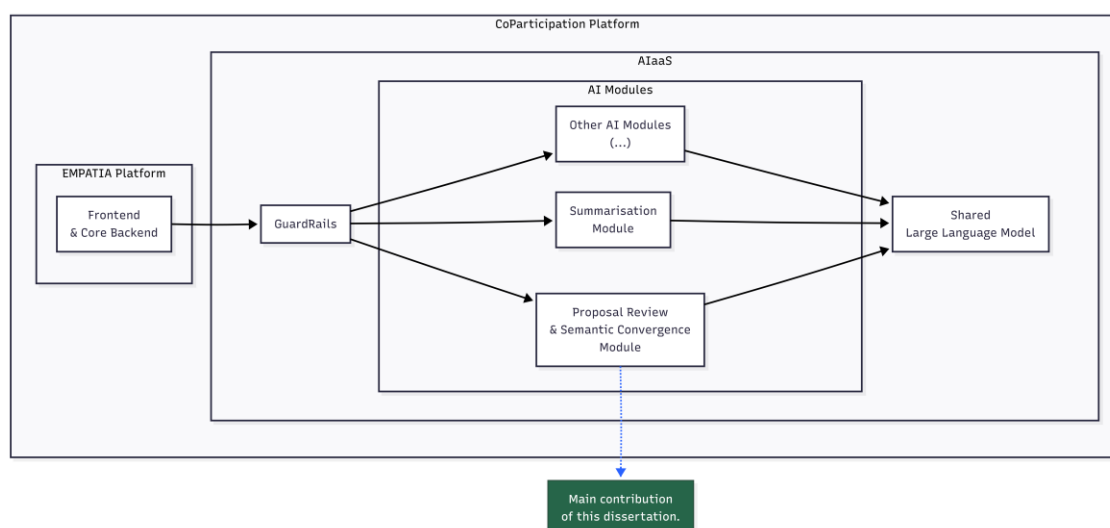


Figure 6 - Positioning within the CoParticipation Ecosystem

3.2 Domain Model

The domain model identifies the main concepts of the solution and clarifies how they relate to each other. It offers a conceptual view of the problem, independent of the technologies used in the prototype.

At the centre of the model is the **Participatory Process**, which represents a specific participation initiative, such as a participatory budgeting edition. Each process includes configuration data such as its name, locale, proposal schema, thematic areas, target audiences, and other constraints. A process may also include several documents, such as regulations or guidelines.

A **Proposal** represents a citizen submission within a participatory process. It contains fixed fields (e.g. title, description, locale, and status) and dynamic fields defined by the schema of the corresponding process. This approach allows different processes to define different submission requirements without changing the overall model.

Process Documents provide the regulatory and contextual information used by the AI components. When a document is ingested, its content may be divided into smaller parts (chunks) and converted into vector representations for retrieval. Relevant rules may also be extracted and stored as structured knowledge, which is later used during proposal validation.

The **Proposal Review Result** represents the outcome of the AI-assisted review. It may include an eligibility decision, explanations, suggestions for improving the text, corrections to dynamic fields, and justifications. Since the system is designed to support rather than replace the citizen, these suggestions must always be reviewed before being accepted.

The **SimilarityResult** represents the outcome of the semantic convergence mechanism. It links a draft proposal to existing proposals and classifies the relationship as similar, complementary, both, or irrelevant. It may also include similarity scores and a short explanation intended for the citizen.

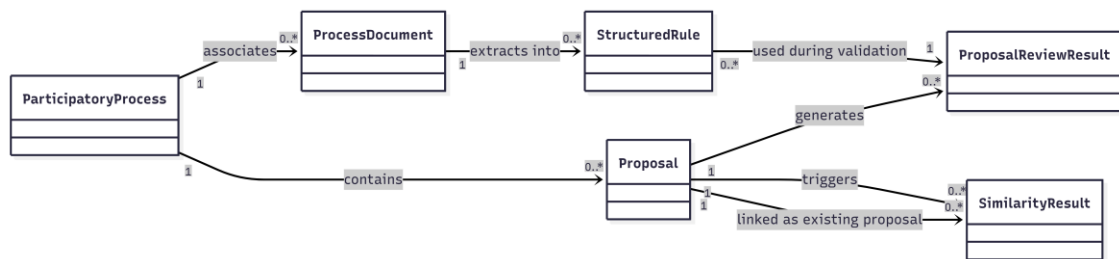


Figure 7 - Domain Model

3.3 Requirements Engineering

Requirements engineering was used to define how the system should behave and which quality attributes it must meet. The requirements were informed by the research questions, the limitations identified in the literature, the goals of the CoParticipation project, and the scope defined for the proof of concept.

The requirements are divided into functional and non-functional categories. Functional requirements describe what the system must do and were prioritised using the MoSCoW method (Clegg and Barker, 1994). Non-functional requirements describe the qualities and constraints that shape the design and implementation of the solution.

3.3.1 Functional Requirements

To support citizens during proposal creation and to improve the clarity and consistency of submissions, the system must provide a set of core capabilities. These requirements describe the expected behaviour of the solution within the CoParticipation ecosystem and define the services implemented in the prototype. For clarity, they are organised into five areas: participatory process management, knowledge and document management, proposal drafting support, semantic convergence support, and model/evaluation support.

Table 3 - Participatory Process Management Functional Requirements

ID	Title	Description	Priority
FR1.1	Register participatory process	The system must allow the registration of a participatory process, including its name, description and configuration needed to support proposal submission and AI assistance.	Must have
FR1.2	Support process-specific dynamic fields	The system must support proposal forms whose fields change according to the configuration of each participatory process.	Must have
FR1.3	Support multilingual processes	The system should support processes defined in different languages, allowing the prototype to operate in more than one linguistic context.	Must have

Table 4 - Knowledge and Document Management Functional Requirements

ID	Title	Description	Priority
FR2.1	Ingest process documents	The system must allow the ingestion of process-specific documents, such as regulations and guidelines, so that they can be used as contextual knowledge by the AI components.	Must have
FR2.2	Store document chunks	The system must divide ingested documents into smaller units and store them with metadata linking them to the original document and process.	Must have

ID	Title	Description	Priority
FR2.3	Extract regulatory rules	The system must extract structured rules from process documents to support eligibility validation.	Must have
FR2.4	Preserve traceability to source documents	The system should maintain links between extracted rules, document chunks, and the original documents.	Should have
FR2.5	Index historical proposals	The system must index historical proposals so they can be used as examples for proposal improvement and semantic comparison.	Must have

Table 5 - Proposal Drafting Support Functional Requirements

ID	Title	Description	Priority
FR3.1	Create proposal draft	The system must allow citizens to start a draft by providing a title, description, and the required process-specific fields.	Must have
FR3.2	Validate proposal eligibility	The system must analyse a proposal against the official rules of the active process and indicate whether it appears eligible or if any concerns were found.	Must have
FR3.3	Provide regulatory explanation	When issues are detected, the system must return a clear explanation to help citizens understand why the proposal may not meet the rules.	Must have
FR3.4	Analyse and improve proposal content	The system must analyse the proposal's fields, identify issues and suggest improvements or corrections while preserving the citizen's original intent.	Must have
FR3.5	Allow acceptance or rejection of suggestions	The system must allow citizens to accept or reject AI suggestions before submitting the proposal.	Must have
FR3.6	Preserve citizen intent	The system must avoid changing the proposal into a different idea and ensure suggestions remain aligned with the citizen's goal.	Must have

ID	Title	Description	Priority
FR3.7	Support budget calibration	The system should use proposal content and historical examples to suggest more realistic estimated values when needed.	Should have
FR3.8	Generate structured AI responses	The system must return AI outputs in a structured format that can be processed and shown consistently to the citizen.	Must have

Table 6 - Semantic Convergence Support Functional Requirements

ID	Title	Description	Priority
FR4.1	Index submitted proposals	The system must index submitted proposals so they can be used in future similarity searches.	Must have
FR4.2	Detect similar proposals	The system must compare a draft with existing proposals from the same process and detect semantically similar ones.	Must have
FR4.3	Detect complementary proposals	The system should identify proposals that are not duplicates but address related needs or objectives.	Should have
FR4.4	Classify proposal relationships	The system should classify retrieved proposals as similar, complementary, both, or irrelevant.	Should have
FR4.5	Explain convergence suggestions	The system must provide a clear explanation of why a proposal is considered similar or complementary.	Must have
FR4.6	Allow independent submission	The system must allow citizens to submit their proposal even when similar or complementary proposals exist.	Must have
FR4.7	Support convergence intention	The system could allow citizens to indicate interest in collaborating or merging with an existing proposal.	Could have

Table 7 - Model and Evaluation Functional Requirements

ID	Title	Description	Priority
FR5.1	Support multiple language models	The system must allow different open-source LLMs to be configured and tested within the same workflow.	Must have
FR5.2	Support domain-adapted models	The system must support a domain-adapted model trained for proposal review and participatory budgeting.	Must have
FR5.3	Support RAG configuration	The system must allow retrieved context (e.g., regulations, historical proposals) to be enabled or disabled for experimentation.	Must have
FR5.4	Provide automatic final decision-making	The system will not replace municipal experts or make final decisions on proposals.	Won't have
FR5.5	Implement toxicity, fraud, and abuse detection	The system will not implement these checks, as they belong to the shared AlaaS guardrail layer.	Won't have

The prioritisation reflects the proof-of-concept nature of the prototype. Must have requirements define the essential features needed to demonstrate the intelligent support mechanisms. Should have requirements improve adaptability, explainability, and evaluation. Could have requirements add optional extensions that are useful but not required to meet the research goals. Won't have requirements set clear boundaries for what is outside the scope of the dissertation.

3.3.2 Non-functional Requirements

Beyond functional behaviour, the solution was designed with several non-functional requirements that ensure flexibility, maintainability, and smooth integration into larger software ecosystems. These quality attributes guided the architectural choices and influenced how the prototype was implemented. Table 8 summarises the main non-functional requirements considered during the design process.

Table 8 - Non-functional Requirements

Identifier	Non-functional Requirement	Description
NFR1	Extensibility	The architecture should allow new AI features to be added without major changes to existing components.
NFR2	Modularity	System responsibilities should be separated into independent components to support isolated development, testing, deployment, and reuse.
NFR3	Explainability	The system should provide recommendations that are easy to understand within the context of the participatory process.
NFR4	Response time	AI-assisted interactions should be delivered within an acceptable timeframe (below 5 seconds) to keep the drafting experience fluid.
NFR5	Maintainability	The implementation should support future updates to language models, retrieval mechanisms, datasets, and domain-specific knowledge without major architectural changes.
NFR6	Adaptability	The intelligent mechanisms should be configurable for different participatory processes and domains by incorporating process-specific knowledge and model adaptations.
NFR7	Usability	The system must provide a responsive, mobile-friendly interface that works consistently across smartphones, tablets, and desktop devices.

These quality attributes influenced several architectural decisions described throughout this chapter, including the adoption of a modular AI service layer, the separation between user-facing and intelligent components, and the use of process-specific knowledge repositories and adaptable language models.

3.3.3 Use Cases

The use cases describe how the main actors interact with the system and summarise the key functionalities identified during requirements engineering. They provide a high-level view of how citizens and process managers use the prototype and show how user actions relate to the functional requirements. Figure 8 presents the use case diagram for the proposed solution, while the following tables describe each use case in more detail.

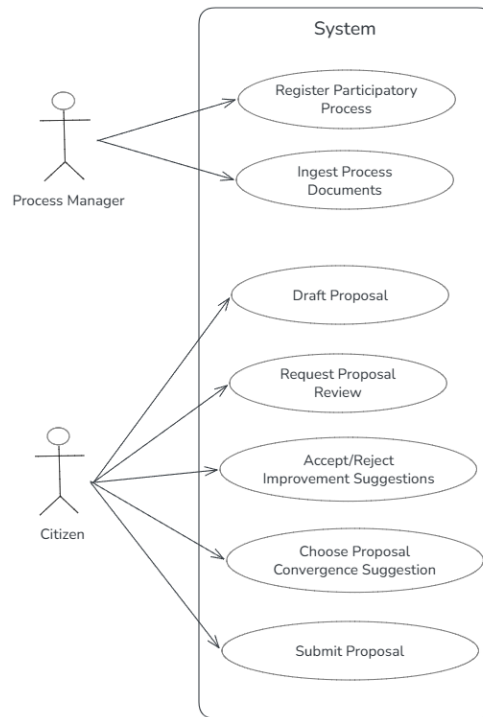


Figure 8 - Use Case Diagram

Table 9 - UC01: Register Participatory Process

Field	Description
ID	UC01
Name	Register Participatory Process
Primary Actor	Process Manager
Objective	Register a new participatory process and configure its basic information so that proposals can be submitted and analysed.
Preconditions	The process manager has access to the administration interface.
Main Flow	1. The manager starts the creation of a new process. 2. The manager provides the required information and configuration. 3. The system validates the information. 4. The system stores the configuration.
Alternative Flow	3a. If mandatory information is missing or invalid, the system reports the error and requests correction.
Postconditions	A new participatory process is registered and ready for document ingestion and proposal submission.
Associated FRs	FR1.1, FR1.2

Table 10 - UC02: Ingest Process Documents

Field	Description
ID	UC02
Name	Ingest Process Documents
Primary Actor	Process Manager
Objective	Upload regulations or supporting documents so they can be used as contextual knowledge during AI-assisted review.
Preconditions	A participatory process has already been registered.
Main Flow	1. The manager uploads one or more documents. 2. The system processes the documents. 3. The documents are indexed and linked to the process.
Alternative Flow	2a. If a document cannot be processed, the system reports the failure and aborts ingestion for that document.
Postconditions	The documents become available as contextual knowledge for review and validation.
Associated FRs	FR2.1, FR2.2, FR2.3, FR2.4

Table 11 - UC03: Draft Proposal

Field	Description
ID	UC03
Name	Draft Proposal
Primary Actor	Citizen
Objective	Create or edit a proposal before requesting AI assistance or submitting it.
Preconditions	A participatory process is open for participation.
Main Flow	1. The citizen opens the proposal form. 2. The citizen fills in the title, description, and dynamic fields. 3. The system stores the draft locally in the session.
Alternative Flow	2a. The citizen may revise the draft multiple times before continuing.
Postconditions	A proposal draft is available for AI-assisted review.
Associated FRs	FR3.1

Table 12 - UC04: Request Proposal Review

Field	Description
ID	UC04
Name	Request Proposal Review
Primary Actor	Citizen
Objective	Obtain AI-assisted feedback on proposal quality, regulatory compliance, and field consistency.
Preconditions	A proposal draft exists.
Main Flow	1. The citizen requests a review. 2. The system checks the proposal against the regulations. 3. If eligible, the system analyses the content and generates suggestions. 4. The results are returned to the citizen.
Alternative Flow	2a. If regulatory issues are found, the system reports them with explanations.
Postconditions	Review results and suggestions are available for inspection.
Associated FRs	FR3.2, FR3.3, FR3.4, FR3.6, FR3.7, FR3.8

Table 13 - UC05: Accept/Reject Improvement Suggestions

Field	Description
ID	UC05
Name	Accept/Reject Improvement Suggestions
Primary Actor	Citizen
Objective	Decide whether to apply the AI-generated improvements to the proposal draft.
Preconditions	Review results are available.
Main Flow	1. The citizen reviews the suggestions. 2. The citizen accepts or rejects each one. 3. The system updates the draft accordingly.
Alternative Flow	2a. The citizen rejects all suggestions and keeps the original draft.
Postconditions	The draft reflects the citizen's final decisions.
Associated FRs	FR3.5

Table 14 - UC06: Choose Proposal Convergence Suggestion

Field	Description
ID	UC06
Name	Choose Proposal Convergence Suggestion
Primary Actor	Citizen
Objective	Review similar or complementary proposals and decide whether to proceed independently or consider convergence.
Preconditions	A review has been completed and related proposals were identified.
Main Flow	1. The system presents similar or complementary proposals. 2. The citizen reviews the suggestions and explanations. 3. The citizen decides whether to continue with the current proposal.
Alternative Flow	1a. If no relevant proposals are found, no suggestions are shown.
Postconditions	The citizen has considered convergence options before submission.
Associated FRs	FR4.2, FR4.3, FR4.4, FR4.5, FR4.6, FR4.7

Table 15 - UC07: Submit Proposal

Field	Description
ID	UC07
Name	Submit Proposal
Primary Actor	Citizen
Objective	Submit the final proposal after reviewing any AI suggestions.
Preconditions	A draft exists and the citizen has completed any desired review or convergence steps.
Main Flow	1. The citizen submits the proposal. 2. The system stores it. 3. The proposal is indexed for future similarity searches.
Alternative Flow	1a. The citizen returns to the draft to make further changes.
Postconditions	The proposal becomes part of the participatory process and is available for future convergence analysis.
Associated FRs	FR3.1, FR4.1

4 Design

The design phase describes how the solution is structured and how its main components work together. It translates the requirements defined in the previous chapter into an architectural model, outlines the technological choices that support the prototype, and explains how the system's services and interactions were organised to deliver AI-assisted support within participatory processes.

4.1 Architectural Model

A modular service-oriented approach was adopted to improve maintainability and extensibility. In this setup, the web platform handles user interaction and process management, while AI-related tasks are delegated to an independent service exposed through interfaces. This separation allows the intelligent components to evolve independently from the web application and makes it easier to integrate new AI features in the future.

Another key principle is the separation between domain knowledge and language-model reasoning. Instead of relying solely on the internal knowledge of a model, the system incorporates external information such as regulations, supporting documents, and previously submitted proposals. This helps ensure that the recommendations are grounded in the specific context of each participatory process.

The architecture also divides reasoning tasks into separate processing stages. Eligibility validation, proposal improvement, and semantic convergence are handled independently. This decision was informed by early experiments (see Section 5.10), which showed that combining multiple tasks in a single inference step produced less reliable results. Assigning each task a clear objective improves consistency and simplifies development and maintenance.

To document the architecture at different levels of abstraction, the C4 model was used (Brown, 2018). The Context, Container, and Component diagrams describe the environment in which the solution operates, the main execution units, and their internal structure. These views are

complemented by a deployment diagram and sequence diagrams that illustrate runtime interactions between the main elements.

4.2 Technological Stack

The proof of concept required selecting technologies that balanced ease of development, local execution, extensibility, and suitability for experimentation. Table 16 summarises the principal technologies adopted and the reasoning behind their selection.

Table 16 - Technological Stack

Technology	Purpose	Justification
Laravel + Livewire	Web platform	EMPATIA is built on Laravel, so using the same stack simplified integration and ensured architectural consistency while allowing fast prototype development.
FastAPI	AI service	FastAPI offers a lightweight, high-performance framework for REST services in Python, making it suitable for implementing the AI module and its APIs.
SQLite	Relational data storage	SQLite requires no server setup and provides all the functionality needed for a proof of concept with minimal overhead.
ChromaDB	Vector database	ChromaDB integrates easily with Python and provides the vector storage and similarity search needed for RAG and semantic convergence.
Ollama	Local LLM execution	Ollama enables simple local deployment of open-source models with minimal configuration, supporting experimentation and reproducible evaluation.
Open-source LLMs	Language model inference	Open-source models avoid dependency on proprietary services and offer a good balance between performance, resource usage, and cost. Their comparison is discussed later.
LoRA	Model adaptation	LoRA was selected because it enables efficient fine-tuning with far fewer resources than full model training, as highlighted in the literature review.

Technology	Purpose	Justification
FastEmbed with sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	Embedding generation	This model provides multilingual support, fast inference, and strong performance for semantic similarity tasks.
Hugging Face Transformers ecosystem (with TRL and PEFT libraries)	Model training	The Hugging Face ecosystem is widely used for LLM development and supports parameter-efficient training methods such as LoRA, making it well suited for the prototype.

The selected technologies support the goals of the solution while keeping a modular architecture that allows experimentation with different models, retrieval strategies, and adaptation techniques. Their use also helps ensure that the prototype is reproducible and can be extended within the broader CoParticipation ecosystem.

4.3 C4 Diagrams

4.3.1 Context Diagram

The context diagram presents the proposed module within the broader CoParticipation environment. Citizens and process managers interact with the EMPATIA-based web platform, while AI-related requests pass through the shared AlaaS layer before reaching the proposal review and semantic convergence module. This separation keeps the dissertation focused on specialised AI assistance, while relying on the surrounding platform for user interaction, governance, and guardrail mechanisms.

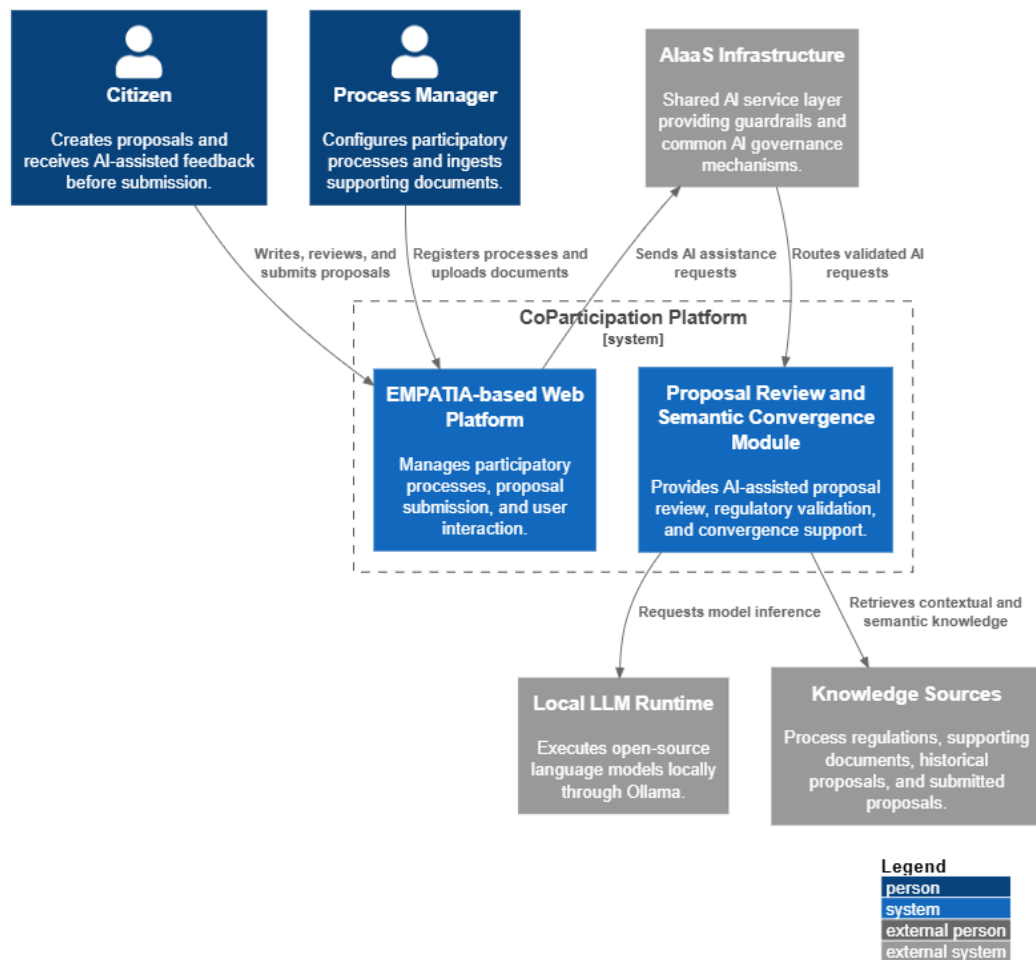


Figure 9 - Context Diagram

4.3.2 Container Diagram

The container diagram breaks the prototype into its main execution and storage units. The Laravel and Livewire application manages user interaction and operational data, while the Python AI service concentrates the experimental AI functionalities. This separation reflects the different technical requirements of the web workflow and the AI mechanisms and allows each part to evolve independently. The use of both relational and vector storage mirrors the nature of the data: structured process information is stored relationally, while documents and proposals require semantic retrieval through embeddings.

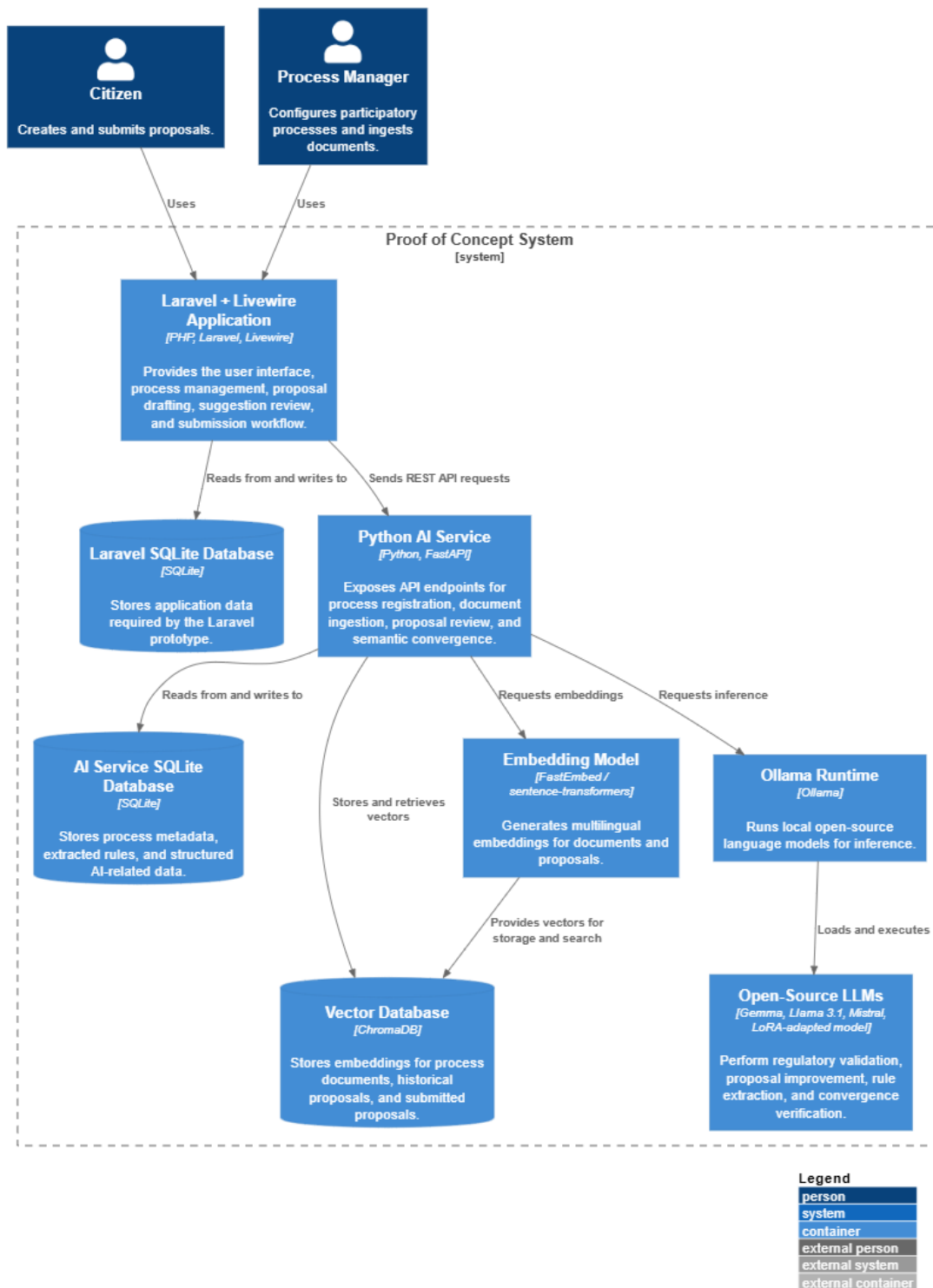


Figure 10 - Container Diagram

4.3.3 Component Diagram

The component diagram describes the internal structure of the Python AI service, where the main contribution of the dissertation is located. The service is organised into components for process management, document ingestion, rule extraction, proposal review, semantic convergence, retrieval, embeddings, and LLM communication. This design avoids concentrating all AI behaviour in a single component and makes the system easier to test, maintain, and extend. It also reflects the decision to separate eligibility validation, proposal improvement, and convergence analysis into distinct responsibilities.

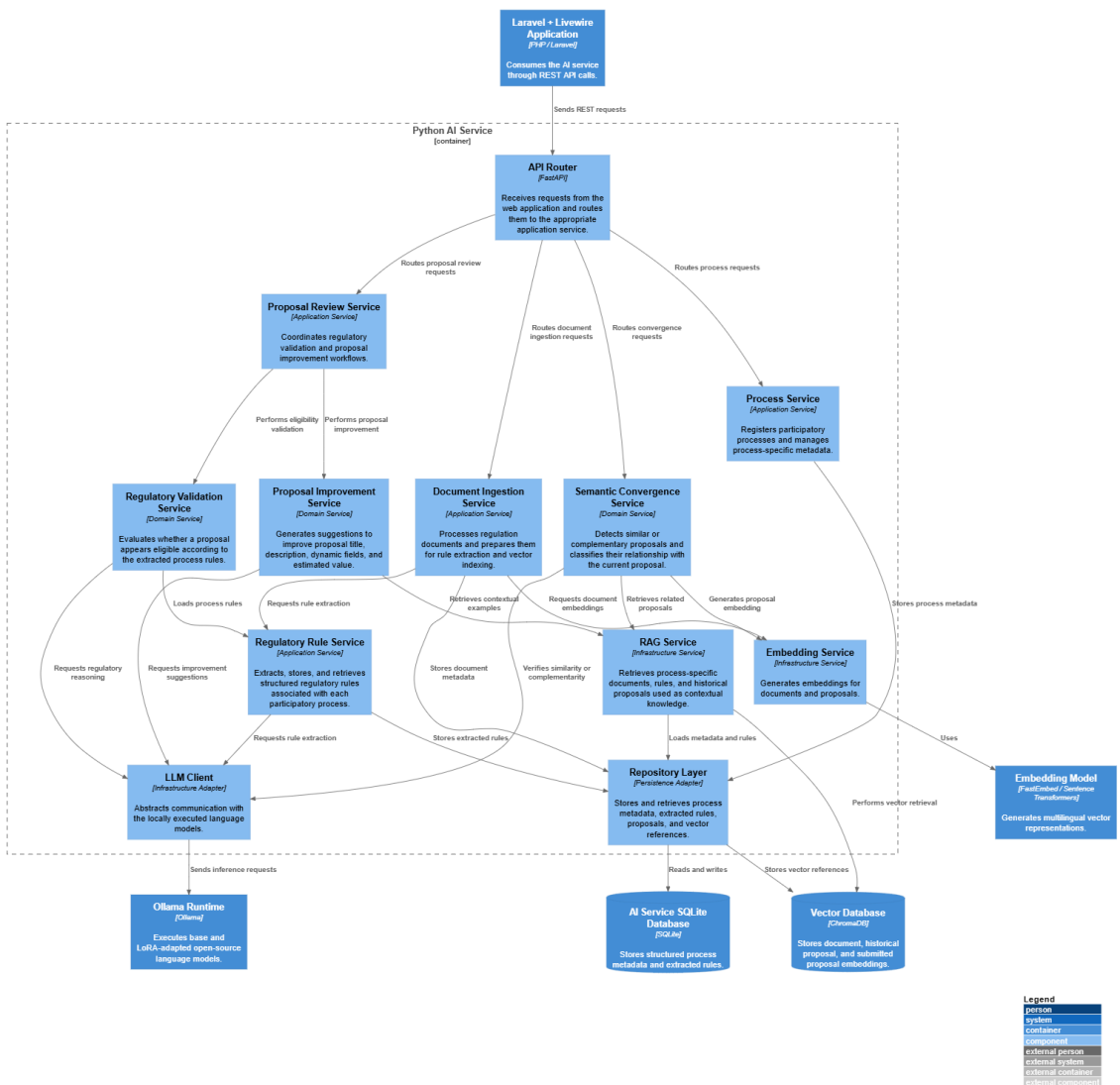


Figure 11 - Component Diagram

4.3.4 Deployment Diagram

The deployment diagram shows how the prototype runs at execution time. The architecture separates the user device, the Laravel application server, the Python AI service environment, and the local model runtime. This organisation suits the proof-of-concept nature of the implementation while keeping each layer logically independent. Local model execution avoids reliance on proprietary cloud services and supports reproducible experimentation, while SQLite and ChromaDB provide lightweight persistence suitable for a prototype.

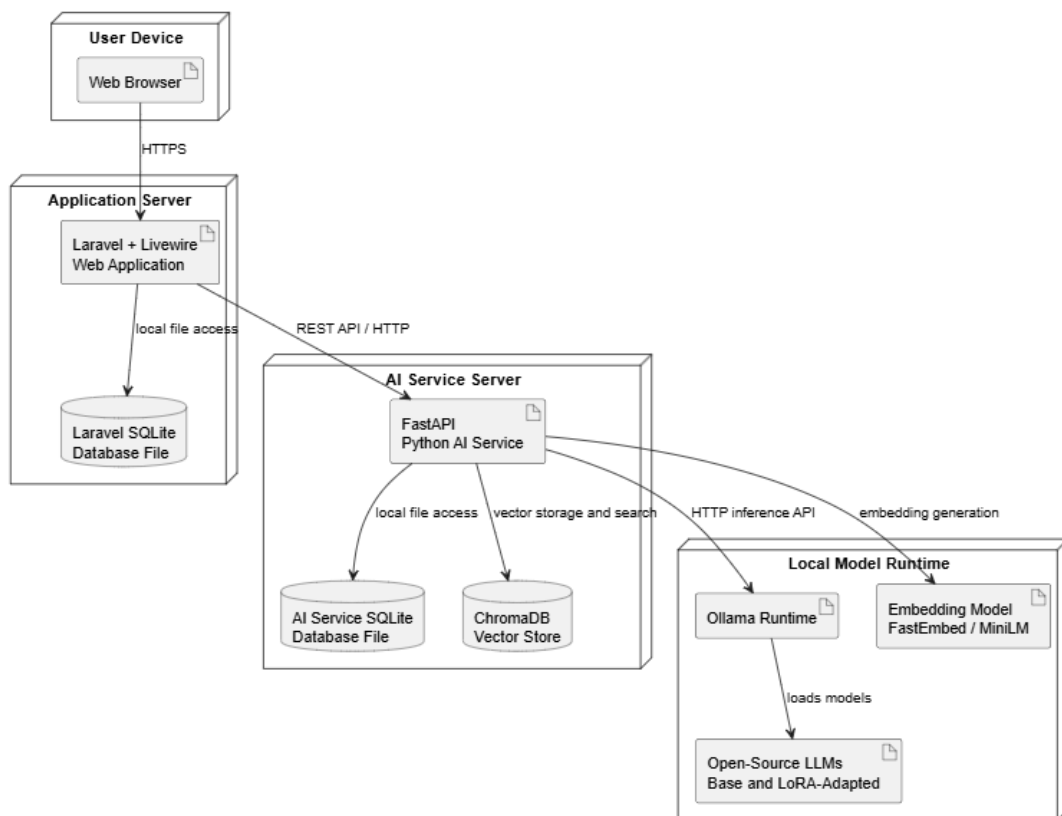


Figure 12 - Deployment Diagram

4.4 Sequence Diagrams

This section presents the sequence diagrams that describe the main workflows implemented in the solution. Each diagram shows how the web platform, the AI service, and the underlying knowledge and model layers interact. For readability, the detailed diagrams are included in Appendix D, Appendix E, Appendix F.

4.4.1 Proposal Review Workflow

This diagram (see Appendix D) represents the two-stage proposal review workflow. The system first checks the proposal against the rules of the participatory process. If the proposal appears eligible, the workflow continues to the improvement stage, where the system suggests refinements to the text and dynamic fields. If regulatory issues are found, the citizen is informed and must correct the proposal before requesting another review. Separating eligibility validation from writing suggestions makes the workflow clearer and more reliable.

4.4.2 Semantic Convergence Workflow

This diagram (see Appendix E) describes how the system identifies semantically related proposals before submission. The current proposal is converted into an embedding using only the title and description, since dynamic fields may not always be accurate. The vector database retrieves candidate proposals from the same process, and the language model evaluates whether each one is similar, complementary, both, or irrelevant. This two-step approach combines efficient retrieval with model-based reasoning and helps avoid weak or misleading convergence suggestions.

4.4.3 Document Ingestion Workflow

This diagram (see Appendix F) shows how process documents (e.g. regulations or guidelines) are ingested. The document is split into smaller chunks, embedded, and stored in the vector database for later retrieval. In parallel, the language model extracts structured regulatory rules, which are stored in the relational database with references to their source. This hybrid approach preserves the original document context while making regulatory validation more focused and easier to trace.

4.5 AI Service Design

The AI service is an independent component responsible for all intelligent processing in the prototype. While the Laravel application manages user interaction, proposal submission, and process management, the AI service performs regulatory validation, proposal improvement, semantic convergence, document ingestion, rule extraction, and communication with the language models. This separation keeps the web application independent of AI-specific implementation details and simplifies future maintenance.

The service exposes its functionality through REST API endpoints, allowing the web application to request AI operations without depending on their internal implementation. Details such as prompt construction, model selection, vector retrieval, and response processing are handled entirely within the AI service.

Internally, the service is organised into specialised components, each responsible for a specific task. This organisation makes the system easier to extend, test, and maintain, while also allowing different models, prompting strategies, or retrieval configurations to be evaluated without changing the web application.

The implementation also follows the AlaaS architecture adopted by the CoParticipation project. The developed module operates as one specialised AI service that can share language models, governance mechanisms, and guardrails with other AI capabilities. This approach promotes reuse and simplifies integration into the broader platform.

Finally, the AI service follows the transparency and human oversight principles established by the AI Act. All generated outputs are presented as recommendations rather than automatic decisions. Citizens remain responsible for accepting or rejecting the suggested changes, while regulatory validation provides advisory feedback without replacing the formal assessment performed by the responsible authority.

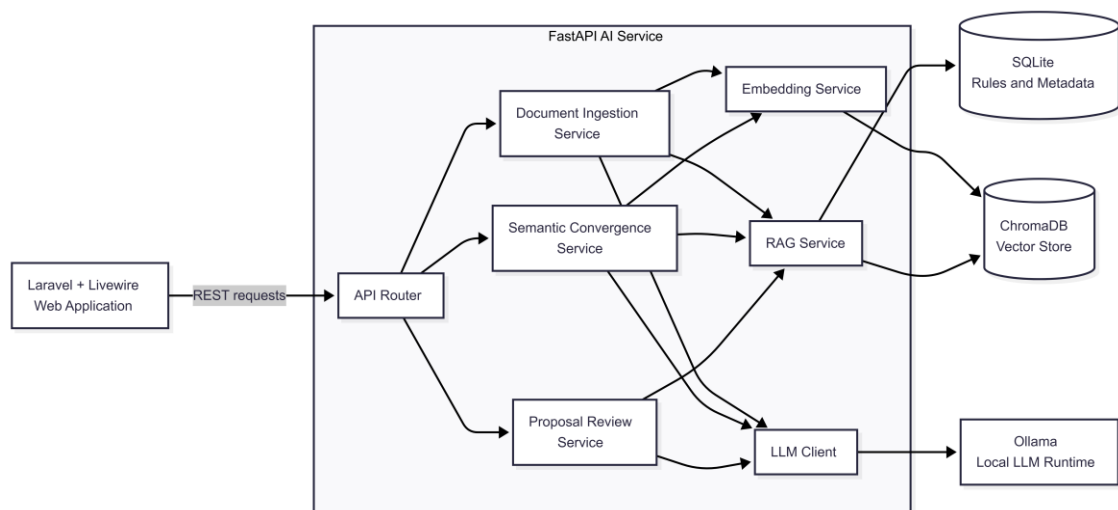


Figure 13 - Internal Organisation of the AI Service

4.6 Knowledge Management Design

The knowledge management design defines how the system stores and retrieves the information needed to support the AI mechanisms. Because participatory processes depend on regulations, dynamic fields, and previously submitted proposals, the AI service cannot rely only on the language model's internal knowledge. It must retrieve the specific information for the active participatory process.

The solution uses two types of storage: relational storage for structured information and vector storage for semantically searchable content. Relational storage keeps process metadata, extracted regulatory rules, and document references. Vector storage keeps regulation chunks, historical proposals, and proposals submitted in the current process.

During document ingestion, uploaded regulations are split into smaller chunks and stored with metadata linking them to the correct document and participatory process. At the same time, the system extracts regulatory rules and stores them separately for proposal validation. This avoids sending full documents to the language model and makes validation more efficient.

Historical proposals are stored as embeddings to provide examples during proposal review, while proposals submitted in the current process support the semantic convergence mechanism by helping identify similar or complementary ideas before submission.

By combining structured rules with semantic retrieval, the system offers context-aware assistance and remains flexible enough to incorporate new knowledge sources, retrieval methods, or embedding models in the future.

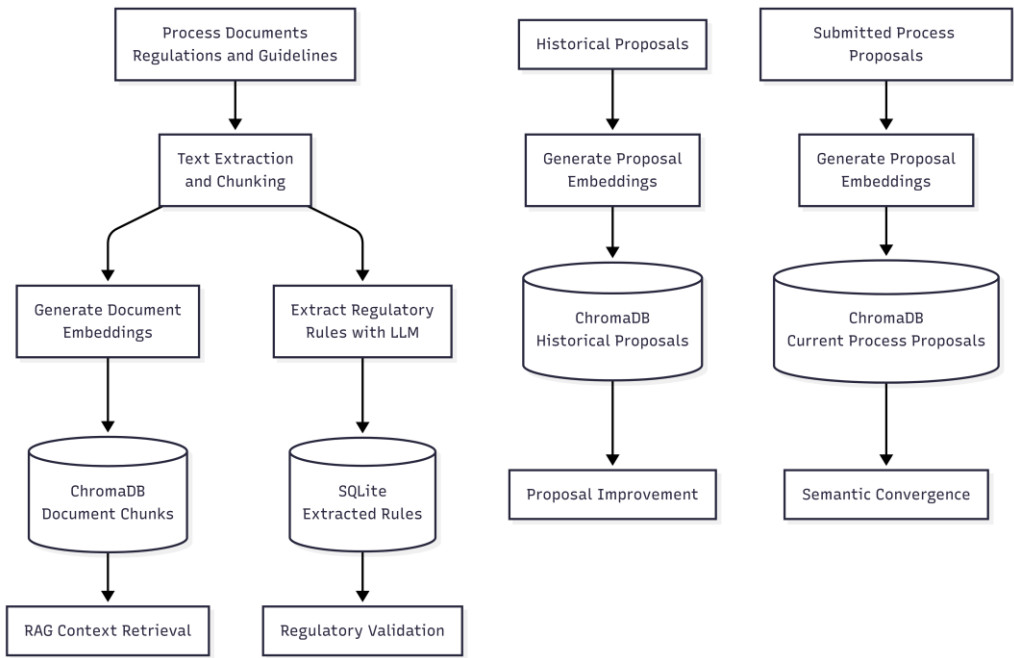


Figure 14 - Knowledge Management and Retrieval Strategy

4.7 Proposal Review and Semantic Convergence Design

The proposal review and semantic convergence mechanisms are the two main intelligent features of the solution. Both are designed to help citizens while they draft proposals, without replacing their decisions or the formal evaluation carried out by public administration.

The proposal review mechanism works in two steps. First, the system checks the proposal against the rules of the active participatory process. This step focuses only on eligibility and exclusion criteria and returns a structured result showing whether the proposal appears to follow the defined rules. If any issues are found, the citizen is informed and can revise the proposal before requesting another review.

Proposals that meet the regulatory requirements move to the improvement step. Here, the system analyses the title, description, and dynamic fields and suggests improvements to wording, structure, thematic area, target audience, or estimated value, while keeping the citizen's original intent. The citizen decides whether to accept or reject each suggestion. This separation between validation and improvement was intentional, as early experiments showed that combining both tasks in a single model call produced less reliable results.

The semantic convergence mechanism helps reduce the fragmentation of similar ideas. Before submission, the system generates an embedding from the proposal title and description and retrieves related proposals from the same participatory process. Dynamic fields are not used because they may still need correction during review.

The retrieved proposals are then checked by the language model to confirm their relationship with the current proposal. This extra step is important because semantic similarity alone cannot always distinguish between truly related proposals and those that only share similar wording. Each proposal is classified as similar, complementary, both, or irrelevant, with a short explanation shown to the citizen. This encourages reflection before submission, reduces fragmentation, and supports collaboration among participants with related ideas.

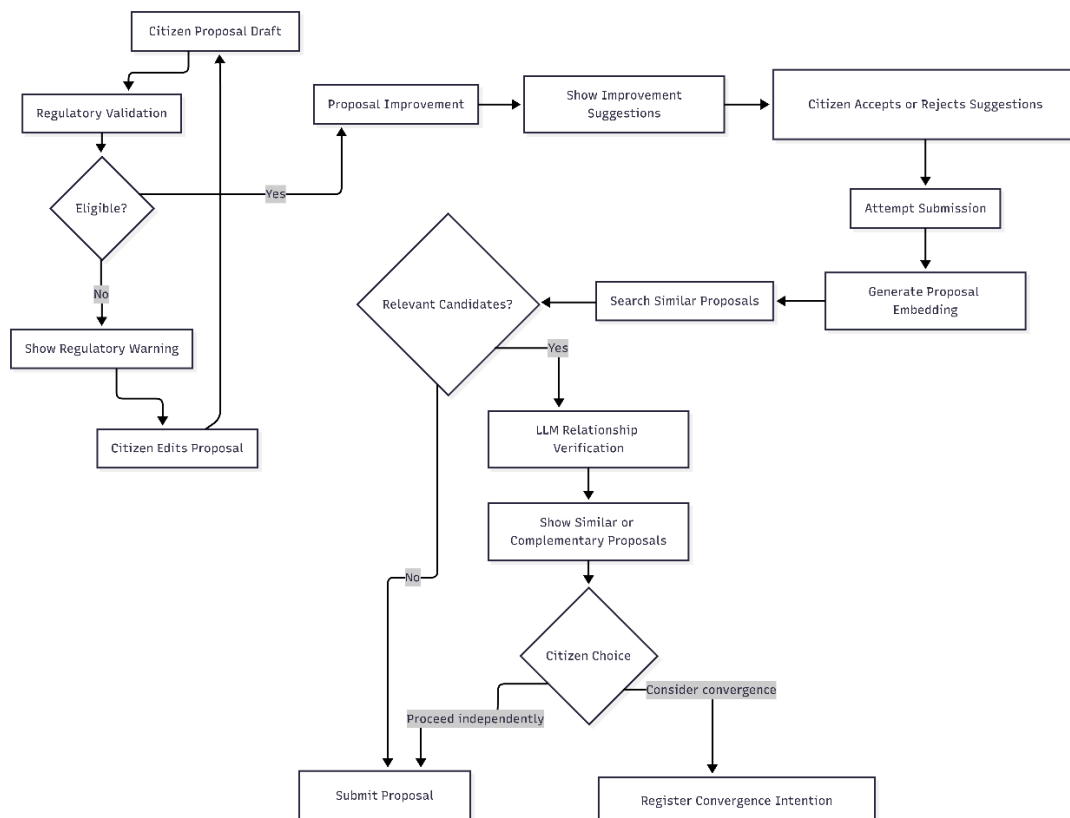


Figure 15 - Proposal Review and Semantic Convergence Workflow

5 Proof of Concept Implementation

This chapter describes how the proof of concept was implemented. Building on the analysis and design from the previous chapters, it explains how the solution was developed, covering the web platform, the AI service, the main processing pipelines, the supporting datasets, the model-adaptation process, and the mechanisms used to ensure stable and reliable operation.

5.1 Implementation Overview

The proof of concept was implemented as two separate applications. The first is a web platform developed with Laravel and Livewire, responsible for process management, proposal submission, and user interaction. The second is a Python-based AI service developed with FastAPI, responsible for executing the intelligent processing pipelines and interacting with local language models. The two applications communicate through REST endpoints exposed by the AI service.

Figure 44 in Appendix G illustrates the interaction between the main components of the prototype.

5.2 Web Platform Implementation

The web platform was implemented using Laravel and Livewire, following the same technology stack as the EMPATIA platform. It manages participatory processes, proposal submission, and the interaction between citizens and the AI service.

The application supports two main entities: participatory processes, which define the context and configuration of each process, and proposals, which represent the contributions submitted by citizens.

The prototype includes two main Livewire components:

- **ProposalForm:** Handles proposal drafting, AI review requests, suggestion management, convergence checking, and final submission.
- **ProposalList:** Displays existing proposals and supports filtering during testing and expert evaluation.

All AI operations are performed by the FastAPI service, while the web platform remains responsible for data persistence and user interaction.

5.2.1 Participatory Process Configuration

Each participatory process stores the information required for proposal submission, including its metadata and a JSON schema defining the fields that citizens must complete.

The schema is interpreted at runtime, allowing different processes to define different submission requirements without changing the database structure or modifying the proposal form. For example, one process may require parish, thematic area, estimated value, and target audience, while another may require only a location field.

A simplified example of a process schema is shown in Figure 16.

```
{
  "fields": [
    {
      "name": "title",
      "type": "text",
      "max_val": 100
    },
    {
      "name": "description",
      "type": "textarea",
      "max_val": 5000
    },
    {
      "name": "parish",
      "type": "select",
      "options": {
        "bonfim": { "en": "Bonfim", "pt": "Bonfim" },
        "campanha": { "en": "Campanhã", "pt": "Campanhã" }
      }
    },
    {
      "name": "thematic_area",
      "type": "select",
      "options": {
        "education": { "en": "Education & Training", "pt": "Educação e Formação" },
        "environment": { "en": "Environment & Sustainability", "pt": "Ambiente e Sustentabilidade" },
        "culture": { "en": "Culture & Heritage", "pt": "Cultura e Património" }
      }
    }
  ]
}
```

Figure 16 - Simplified Process Schema Example

The same schema is also sent to the AI service, allowing proposal review to validate both the proposal text and the submitted dynamic fields according to the requirements of the active process.

5.2.2 Dynamic Proposal Form

The proposal form is generated dynamically based on the schema of the active participatory process. Static fields such as title and description are always displayed, while dynamic fields are rendered according to their configured type (e.g. text inputs, numeric fields, dropdowns).

Before any AI request is performed, Laravel and Livewire validate the submitted data by checking required fields, numeric constraints, maximum lengths, and allowed values. This ensures that only structurally valid proposals are processed by the AI service.

The form preserves the current proposal state throughout the interaction, allowing citizens to iteratively review their proposal, apply or reject suggestions, make additional changes, and finally submit the revised version.

The figure displays two versions of a 'Submit Your Proposal' form side-by-side, illustrating how the form dynamically adapts to different participatory processes. Both forms share a common header with a 'Municipality' logo, the process name, and language toggles (EN, PT). A 'Proposals List' link is present in the top left of each form's content area.

Left Form: Orçamento Participativo Porto 2026

- PROPOSAL TITLE ***: Text input with placeholder 'Give your proposal a clear, descriptive title'.
- DESCRIPTION ***: Text area with placeholder 'Describe your proposal in detail — what problem does it solve? Who benefits? How would it be implemented?'.
- PARISH ***: Dropdown menu with placeholder 'Select an option...'.
- THEMATIC AREA ***: Dropdown menu with placeholder 'Select an option...'.
- ESTIMATED VALUE (€) ***: Text input with placeholder '€ 0.00'.
- TARGET AUDIENCE ***: Dropdown menu with placeholder 'Select an option...'.
- Action**: 'Review with AI' button.

Right Form: Local Improvements Initiative

- PROPOSAL TITLE ***: Text input with placeholder 'Give your proposal a clear, descriptive title'.
- DESCRIPTION ***: Text area with placeholder 'Describe your proposal in detail — what problem does it solve? Who benefits? How would it be implemented?'.
- LOCATION ***: Text input with placeholder 'Enter the location of your proposal'.
- Action**: 'Review with AI' button.

Both forms include a footer with the text '© 2026 Municipality · Participatory Budgeting'.

Figure 17 - Dynamic Proposal Creation Form with Process-Specific Fields

5.3 Integration between Laravel and the AI Service

Communication between the Laravel application and the Python AI service is performed through HTTP requests. The Laravel layer sends structured JSON payloads and receives structured JSON responses, allowing the frontend to display validation messages, improvement suggestions, and convergence results without handling raw model output.

All communication is encapsulated in a dedicated service class, which centralises the HTTP logic and prevents Livewire components from issuing direct API calls. This makes the integration easier to maintain and keeps the web layer focused on user interaction. Each request includes the process identifier and locale, enabling the FastAPI service to load the correct configuration, rules, documents, and vector collections. The locale setting primarily determines the language of the LLM response, and a simplified representation of this communication pattern is shown in Figure 18, while Table 17 summarises the main interactions between Laravel web app and FastAPI.

A code snippet in PHP showing an HTTP POST request. It starts with a comment in three colored circles (red, yellow, green). The code sets a timeout of 30 seconds, adds headers for 'X-Process-Id' and 'X-Locale' using variables \$processUuid and \$locale, and then posts \$proposalData to \$url.

```
$response = Http::timeout(30)
    ->withHeaders([
        'X-Process-Id' => $processUuid,
        'X-Locale' => $locale,
    ])
    ->post($url, $proposalData);
```

Figure 18 - Simplified Laravel Communication Pattern

Table 17 - Interactions between Laravel and FastAPI

Laravel Action	FastAPI Processing	Returned Result
Request regulatory validation	Loads process rules and evaluates eligibility	Eligibility status and explanation
Request proposal improvement	Reviews text and dynamic fields	Suggested title, description, fields, and explanation
Submit proposal	Indexes proposal for future retrieval	Submission confirmation
Check convergence	Retrieves and verifies related proposals	Similar/complementary proposal suggestions
Ingest process document	Processes document, chunks text, extracts rules	Ingestion status

5.4 AI Service Implementation

The AI service was implemented as a FastAPI application that centralises all intelligent behaviour and exposes it to the Laravel platform through REST endpoints. Internally, the service is organised into modules responsible for request handling, proposal review, document ingestion, RAG operations, embeddings, vector storage, and communication with locally executed language models.

Each request is processed through a dedicated pipeline that performs the operations required for the selected task. Depending on the endpoint, these operations may include:

- Input normalisation;
- Loading contextual information;
- Constructing prompts or retrieval queries;
- Calling the vector store or the selected LLM;
- Parsing and validating the model output;
- Formatting the final response.

Table 18 outlines the execution steps for each core processing pipeline.

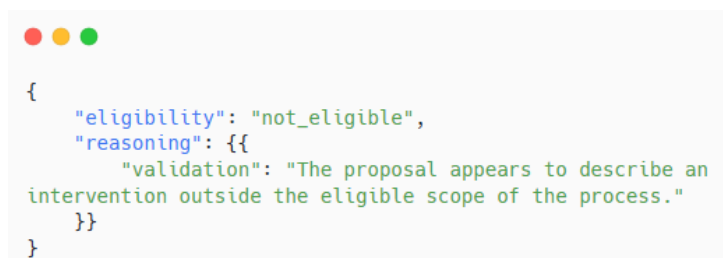
Table 18 - Main Processing Pipelines

Pipeline	Main Steps
Regulatory validation	Load process rules, build validation prompt, call LLM, parse eligibility result
Proposal improvement	Load schema and contextual examples, build improvement prompt, call LLM, parse suggestions
Semantic convergence	Generate embedding, retrieve candidates, optionally verify with LLM, return related proposals
Document ingestion	Extract text, chunk document, embed chunks, store vectors, extract and store rules
Proposal indexing	Generate proposal embedding and store it in the process-specific vector collection

5.4.1 API Request Processing

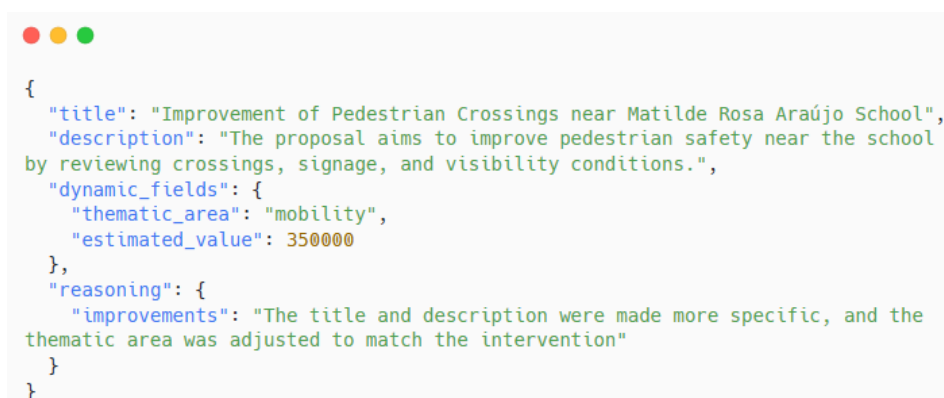
Each API operation receives structured data rather than raw text. For proposal review, the request includes the proposal title, description, locale, process identifier, and dynamic fields. This structure allows the AI service to build task-specific prompts and apply validation rules before and after the LLM response.

Before returning a response, every LLM output is converted into a predefined JSON structure. Invalid or incomplete responses are automatically repaired whenever possible. If recovery is unsuccessful, the service returns a controlled fallback response instead of exposing raw model output, ensuring that the Laravel application always receives predictable and valid data.



```
{
  "eligibility": "not_eligible",
  "reasoning": {
    "validation": "The proposal appears to describe an intervention outside the eligible scope of the process."
  }
}
```

Figure 19 - Regulatory Validation Response Example



```
{
  "title": "Improvement of Pedestrian Crossings near Matilde Rosa Araújo School",
  "description": "The proposal aims to improve pedestrian safety near the school by reviewing crossings, signage, and visibility conditions.",
  "dynamic_fields": {
    "thematic_area": "mobility",
    "estimated_value": 350000
  },
  "reasoning": {
    "improvements": "The title and description were made more specific, and the thematic area was adjusted to match the intervention"
  }
}
```

Figure 20 - Improvement Response Example

5.4.2 Prompt Engineering and Structured Output Control

Prompt engineering was used to make the behaviour of the LLMs predictable, task-specific, and suitable for integration with the web application. Because the system depends on structured responses that must be interpreted automatically, the prompts were designed not only to guide the model's reasoning but also to constrain the format and scope of the output.

Instead of using a single general-purpose prompt, the implementation uses specialised prompts for each operation:

- **Regulatory validation:** focuses only on eligibility and compliance with process rules.
- **Proposal improvement:** focuses on rewriting and improving the title, description, and dynamic fields.
- **Semantic convergence:** focuses on classifying relationships between proposals as similar, complementary, both, or irrelevant.

Separating these prompts reduces task interference and prevents the model from mixing regulatory decisions with writing suggestions.

Each prompt follows a structured organisation composed of several sections:

Table 19 - Prompt Generic Structure

Prompt Section	Purpose
System role	Defines the model's role (e.g., validator, reviewer, convergence analyser)
Task objective	Describes the specific operation to be performed
Context	Provides rules, schema constraints, historical proposals, or retrieved candidates
Proposal data	Includes the submitted title, description, locale, and dynamic fields
Behaviour constraints	Defines what the model must and must not do (e.g., preserve citizen intent)
Output schema	Forces the response to follow a predefined JSON structure

The prompts also include step-oriented reasoning instructions. The model is asked to analyse the proposal according to the relevant criteria before producing the final response. However, the system does not expose long reasoning traces to the citizen. Only the final structured result and a concise explanation are returned, keeping the interface clear and accessible.

Strict output control is applied across all AI operations. Each feature expects a minimal JSON structure:

- **Regulatory validation:** eligibility result + explanation.
- **Proposal improvement:** improved fields + justification.
- **Semantic convergence:** relationship classification + explanation.

This strategy provides three advantages:

- The FastAPI service can validate and normalise the model response before sending it to Laravel.
- The frontend can present suggestions consistently through interface components.
- The model is constrained to return only the required fields, reducing hallucinations and unnecessary content.

When invalid or incomplete JSON is produced, the AI service attempts to repair the response. If recovery is not possible, a controlled fallback is returned instead of exposing raw model output. This improves robustness and prevents malformed responses from disrupting the workflow.

Together with RAG and LoRA, prompt engineering forms one of the three adaptation mechanisms adopted in this dissertation. While LoRA specialises the language model and RAG

supplies contextual knowledge, prompt engineering guides how the model applies that knowledge to each task.

```
REGULATORY_VALIDATION_PROMPT = """
You are a strict regulatory eligibility validator for a municipal e-participation platform.

Your only task is to decide whether the citizen proposal is eligible according to the official
regulations of the current participatory process.

YOU MUST RESPOND IN THE FOLLOWING LANGUAGE: {locale}

The regulations are provided as a JSON array. Each object contains: type, confidence and rule_text
{rules_context}

- Use the schema only to understand submitted dynamic fields and explicit numeric min/max constraints.
- You should not validate if a dynamic field is not the best fit for the submitted proposal.
- The official regulations remain the primary source for eligibility and what you should validate.

{schema_context}

Evaluate only this proposal. Do not rewrite it. Do not make it more compliant.

Proposal Title:
{title}

Proposal Description:
{description}

Proposal Original Locale:
{original_locale}

Proposal Dynamic Fields:
{dynamic_fields}
...
```

Figure 21 - First Part of Regulatory Validation Prompt

5.4.3 Integration with Ollama

Ollama was used to run open-source language models locally. The FastAPI service communicates with Ollama through an internal LLM client abstraction that hides provider-specific details from the rest of the system. This makes it possible to switch models or adjust inference parameters without modifying the proposal-review logic.

The LLM client is responsible for:

- selecting the configured model.
- sending prompts to the Ollama runtime.
- defining generation parameters.
- receiving model responses.
- returning the generated text to the corresponding pipeline.

Model selection, maximum output tokens, context-window size, temperature, timeout, and keep-alive behaviour are all controlled through configuration. This flexibility was important during experimentation, as different models required different settings to produce complete and well-structured responses.

A warm-up mechanism was also implemented. When enabled, the service sends a small request to Ollama during startup so that the model is loaded before the first real interaction. This reduces the latency of the first AI-assisted request.

```
debian@Fixo-Miguel:/mnt/c/Users/migue$ ollama list
NAME                                ID                                SIZE    MODIFIED
llama3.1-8b-pb-expert:latest        1ca4a8ac715c                     4.9 GB  13 days ago
llama3.1:8b                          46e0c10c039e                     4.9 GB  2 weeks ago
gemma4:latest                       c6eb396dbd59                     9.6 GB  5 weeks ago
mistral:latest                      6577803aa9a0                     4.4 GB  2 months ago
```

Figure 22 - Ollama List Command

5.5 Proposal Review Implementation

The proposal review feature is implemented as a two-stage workflow. This is one of the most important implementation decisions in the prototype, because early tests showed that a single model call was too broad when it had to validate regulations, improve text, correct fields, and reason about budget values at the same time. The implemented workflow separates regulatory validation from proposal improvement.

5.5.1 Regulatory Validation

Regulatory validation is executed first. The FastAPI service receives the proposal and loads the extracted rules associated with the active process. These rules are then formatted into a compact context and inserted into a validation prompt.

The prompt focuses exclusively on determining whether the proposal appears to violate any process rules. It does not attempt to improve the text. The expected output is minimal: an eligibility decision and a citizen-friendly explanation.

Table 20 - Validation Prompt Structure

Pipeline	Main Steps
System role	Defines the model as a strict regulatory validator
Official rules	Provides extracted process rules
Proposal schema	Provides available fields and constraints
Submitted proposal	Provides title, description, and dynamic fields
Output format	Forces a structured JSON response

If the proposal is considered eligible, the Laravel application automatically proceeds to the improvement stage. If problems are detected, the citizen receives a warning and must revise the proposal before requesting another review.

The screenshot shows a web interface for submitting a proposal. At the top, there is a link to 'Proposals List' and a heading 'Submit Your Proposal'. Below the heading, a message states: 'Share your idea for improving our community. Our AI assistant will review your proposal and offer suggestions before final submission.' A red pill-shaped button with a white exclamation mark and the text 'NOT ELIGIBLE' is displayed. Below this, a yellow box contains the 'AI VALIDATION ANALYSIS' section. It includes a quote: 'The proposal is ineligible because its stated purpose—the repair of a religious building for internal parish community activities—directly and exclusively benefits a religious group, which violates the grounds for exclusion.' At the bottom of the yellow box are two buttons: 'Edit Proposal' (with a pencil icon) and 'Continue to AI Improvement' (with a right arrow icon). Below the yellow box, the 'PROPOSAL TITLE' field contains the text 'Rehabilitation of the São Martinho Church Roof'. The 'DESCRIPTION' field contains the text: 'Complete structural repair of the São Martinho Church roof, including replacement of beams, tiles, and waterproofing. The intervention is exclusively intended for the religious building and internal parish community activities.'

Figure 23 - Regulatory Warning Displayed to the Citizen

5.5.2 Proposal Improvement

The improvement stage is executed only after the proposal passes regulatory validation. Its purpose is to enhance the clarity and coherence of the proposal, not to decide whether it should be accepted.

The prompt receives the proposal title, description, dynamic fields, the process schema, and contextual examples retrieved through RAG. The model is instructed to preserve the citizen's intent and suggest improvements only when they make the proposal clearer, more complete, or more coherent.

The implementation supports improvements to the title, description, and dynamic fields. After receiving the model response, the AI service validates the returned JSON and normalises field values when needed. Numeric values may be converted to the expected type, and dynamic-field options are checked against the process schema before being returned to Laravel.

The Laravel interface then compares the current proposal values with the AI suggestions. Suggested changes are stored as pending suggestions and displayed inline, allowing the citizen to accept or reject each one individually.

This interaction ensures that the AI remains assistive. The system never overwrites the proposal automatically and the citizen remains responsible for deciding which suggestions to incorporate.

[← Proposals List](#)

Submit Your Proposal

Share your idea for improving our community. Our AI assistant will review your proposal and offer suggestions before final submission.

✓ ELIGIBLE PROPOSAL

AI VALIDATION ANALYSIS

"The proposal meets all stated requirements regarding thematic scope, budget limits, location, and general feasibility for public investment."

AI IMPROVEMENT RATIONALE

"The title was made more formal and specific by adding 'Rehabilitation of Public Football Field near ISEP' for better clarity. The description was refined to be more professional and action-oriented, clearly stating the goal is comprehensive rehabilitation and ensuring safety/functionality. All dynamic fields were reviewed: the location (Paranhos), target audience (All Citizens), and estimated value (\$50,000) remain appropriate as they accurately reflect the scope of rehabilitating a local public sports facility. The thematic area suggested (Sport & Recreation) is a better fit taking into account the proposals context."

! Our AI has suggested 3 improvement(s) for your proposal. Review them below before submitting.

PROPOSAL TITLE *

Football Field

✕ AI SUGGESTION

Rehabilitation of Public Football Field near ISEP

✕ ✓

DESCRIPTION *

Rehabilitate the small public football field in the neighborhood across from ISEP.

✕ AI SUGGESTION

We propose the comprehensive rehabilitation of the small public football field located across from ISEP. This project aims to restore the facility, ensuring it remains a safe and functional recreational space for local residents.

✕ ✓

PARISH *

Paranhos

THEMATIC AREA *

Culture & Heritage

✕ AI SUGGESTION

Sport & Recreation

✕ ✓

ESTIMATED VALUE (€) *

€ 50000

TARGET AUDIENCE *

All Citizens

Figure 24 - Inline Proposal Improvement Suggestions with Accept and Reject Options

5.6 Semantic Convergence Implementation

The semantic convergence feature is executed before final proposal submission. Its goal is to reduce fragmentation by informing citizens when similar or complementary proposals already exist in the same participatory process.

The implementation begins by generating an embedding from the proposal title and description. Dynamic fields are intentionally excluded because they may have been incorrectly selected and should not restrict retrieval.

The vector search provides a first filtering step. However, the implementation also supports LLM-based verification, since embedding similarity alone cannot reliably distinguish between duplicates, complementary ideas, and false positives. For example, two proposals may refer to the same location but suggest different interventions that could be combined.

The convergence response includes candidate proposals, similarity and complementarity information, and an explanation. If no relevant proposals are found, the frontend continues with the normal submission flow. When convergence suggestions are presented, the citizen may choose to proceed independently or indicate an intention to collaborate. In the prototype, collaboration intentions are stored with the proposal metadata for later inspection.

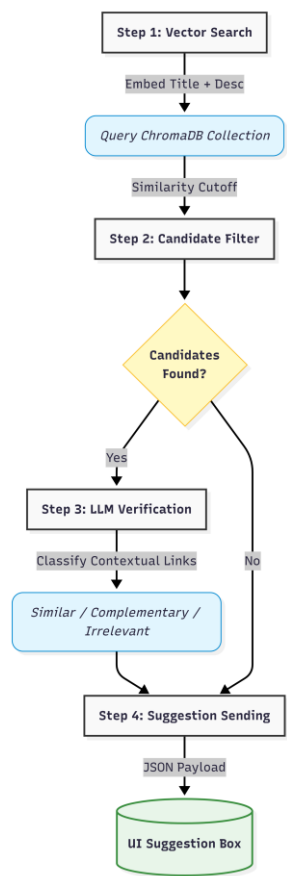


Figure 25 - Semantic Convergence Workflow

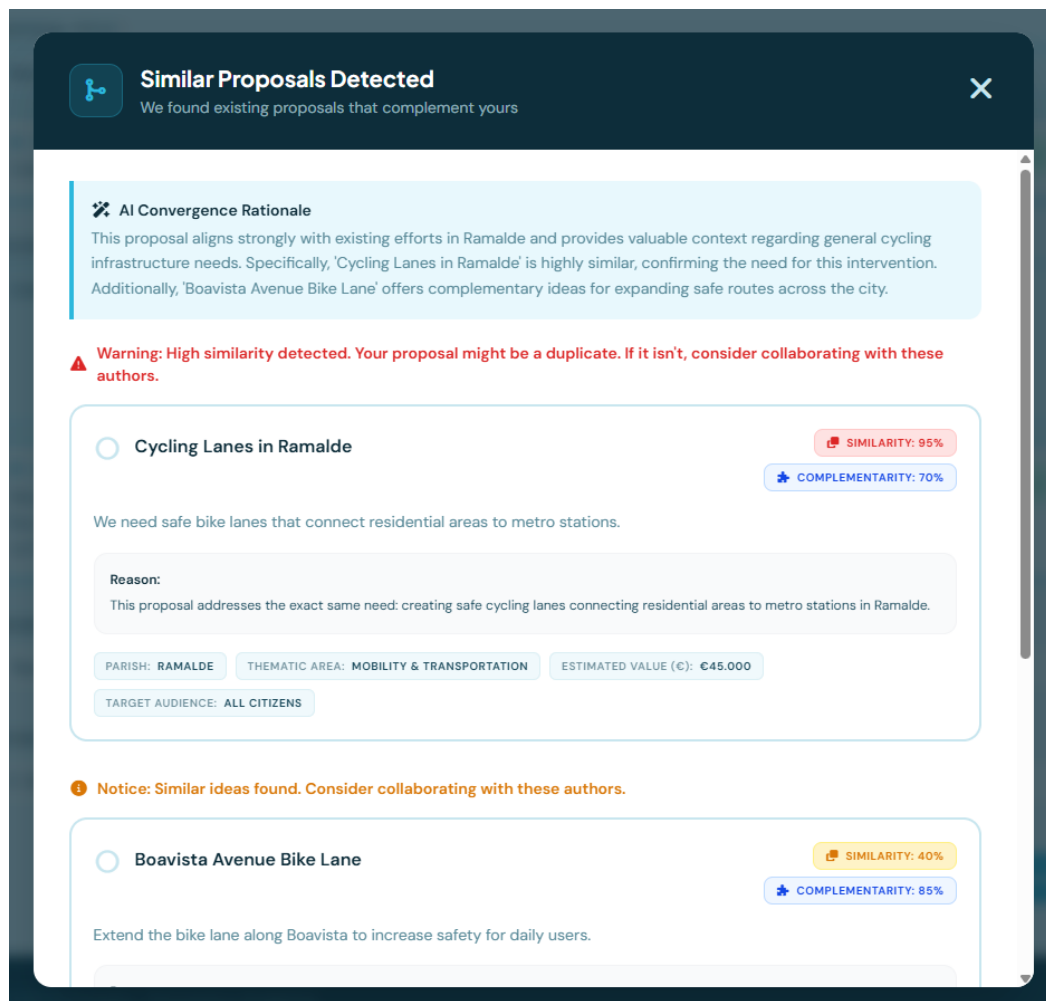


Figure 26 - Semantic Convergence Modal with Similar and Complementary Proposals

5.7 Embeddings and Retrieval Implementation

Embeddings are used in three main parts of the prototype: regulation document retrieval, historical proposal retrieval, and semantic convergence between submitted proposals. The selected embedding implementation uses FastEmbed with the multilingual model `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`. This model was chosen because the prototype includes Portuguese and English examples and therefore requires multilingual semantic representations.

The embedding process converts textual content into vector representations that can be stored and searched in ChromaDB. The implementation uses separate collections for different retrieval purposes. This avoids mixing unrelated data and makes the retrieval behaviour easier to configure and evaluate. The persistence directory, top-K values, and similarity thresholds are configurable, which allows the retrieval behaviour to be adjusted during experiments.

Table 21 - Types of Information Stored in ChromaDB

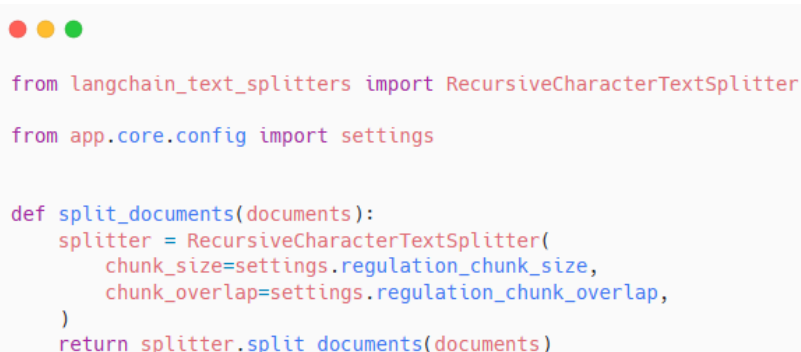
Collection Type	Stored Content	Purpose
Regulation chunks	Text chunks from process documents	Context retrieval and traceability
Historical proposals	Previously approved proposal examples	Proposal improvement and budget calibration
Process proposals	Submitted proposals from the active process	Similarity and convergence detection

5.8 Document Ingestion Implementation

Document ingestion is used to prepare process regulations and supporting documents for later AI-assisted review. This operation is mainly administrative and was implemented programmatically for the prototype rather than through a full administrative interface.

The ingestion pipeline begins when a process document is sent to the FastAPI service. The document text is extracted and divided into smaller chunks. Each chunk is embedded and stored in ChromaDB with metadata that identifies the original document, process, chunk, and source location when available. In parallel, the LLM is used to extract structured regulatory rules from the document. These rules are stored in the AI service relational database and later used during regulatory validation.

A key part of this pipeline is the configurable text-splitting mechanism, which is implemented using a recursive character-based splitter and is shown in Figure 27.



```

from langchain_text_splitters import RecursiveCharacterTextSplitter

from app.core.config import settings

def split_documents(documents):
    splitter = RecursiveCharacterTextSplitter(
        chunk_size=settings.regulation_chunk_size,
        chunk_overlap=settings.regulation_chunk_overlap,
    )
    return splitter.split_documents(documents)

```

Figure 27 - Function Responsible for Splitting Documents into Overlapping Text Chunks

The `chunk_size` parameter defines how much text is included in each segment. This value is configurable because different regulations vary in structure and density:

- shorter chunks improve retrieval precision but may lose context;
- longer chunks preserve context but may reduce retrieval accuracy.

The `chunk_overlap` parameter ensures that consecutive chunks share part of their content. This overlap is important because:

- regulatory rules often span multiple sentences or paragraphs;
- splitting text strictly by size may cut a rule in half;
- overlapping chunks preserve continuity and reduce the risk of losing meaning at boundaries.

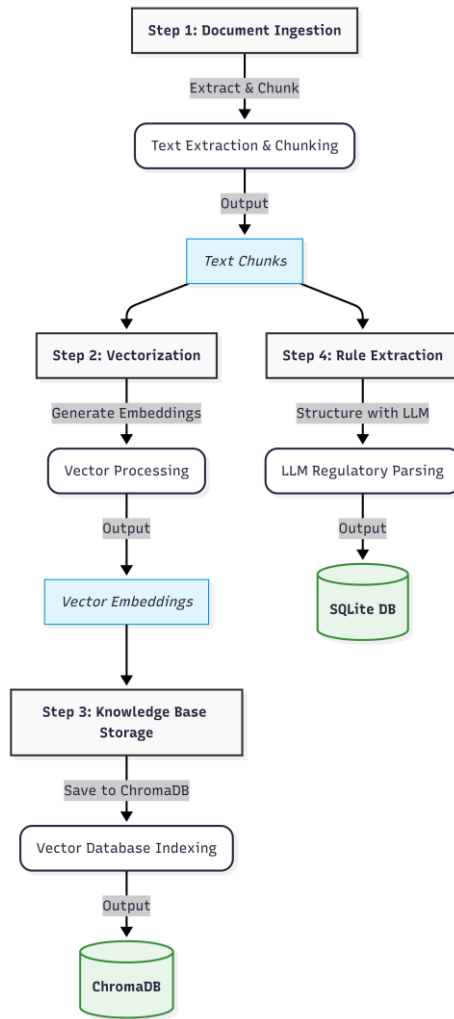


Figure 28 - Document Ingestion Workflow

This hybrid approach allows the system to use extracted rules for focused validation while preserving the original document chunks for traceability and future retrieval.

5.9 Dataset Construction

Several datasets were created to support development, demonstration, model adaptation, and evaluation. All datasets were stored in JSON or JSONL format to simplify processing by both the Laravel and FastAPI applications. Table 22 summarises the datasets used in the implementation.

Table 22 - Datasets Created

Dataset	Size	Language	Origin	Main Use
Historical proposal dataset	1500	Portuguese	Scraped from public participatory platforms	Historical grounding and semantic comparison
Example Process 1 seed proposals	50	Portuguese	Artificially generated	Prototype seeding and convergence testing
Example Process 2 seed proposals	10	English	Artificially generated	Multilingual demonstration
Proposal rewrite dataset	300	Portuguese	Artificially generated	Proposal improvement examples
LoRA training dataset	650	Mainly Portuguese	Manually curated instruction examples	Domain adaptation of the LLM

5.9.1 Historical Proposal Dataset

The historical proposal dataset was created by scraping publicly available information from Portuguese participatory platforms. The selected sources included Lisboa Participa, Cascais Participa, POP Penha, Orçamento Participativo Jovem, and OP Tavira. The collected records correspond to winning proposals or proposals that had passed a technical analysis phase. This made them useful as examples of valid participatory proposals. After scraping, a cleaning script was applied to retain only the fields relevant to the prototype. This dataset supports historical proposal retrieval, budget calibration, and semantic comparison.

To ensure compliance with data-protection requirements, all scraped information was publicly accessible, and no personal identifiers were collected. Any fields that could contain personal data were excluded or anonymised during processing, ensuring that the dataset respects GDPR principles and does not involve sensitive information.

```
{
  "title": "Disponibilização online gratuita de todas sebatas universitárias",
  "description": "Sendo as sebatas essenciais para a vida de um estudante...",
  "thematic_area": "Inovação social",
  "estimated_value": "60000"
}
```

Figure 29 - Historical Dataset Structure Example

5.9.2 Process Seed Datasets

Two process seed datasets were created to populate the prototype with realistic test data. The first contains 50 Portuguese proposals with dynamic fields such as parish, thematic area, estimated value, and target audience. The second contains 10 English proposals and was used to demonstrate that the system can support multilingual process configurations. These datasets were used to test proposal submission, indexing, retrieval, and convergence behaviour under controlled conditions.

```
{
  "title": "Ciclovias em Ramalde",
  "description": "Precisamos de ciclovias seguras que liguem as
zonas residenciais às estações de metro.",
  "dynamic_fields": {
    "parish": "ramalde",
    "thematic_area": "mobility",
    "estimated_value": 45000,
    "target_audience": "all"
  }
}
```

Figure 30 - Portuguese Process Seed Dataset Structure Example

5.9.3 Proposal Rewrite Dataset

The proposal rewrite dataset contains 300 examples of poorly formulated proposals and corresponding improved versions. The examples were artificially generated and manually validated to remove duplications and ensure quality. The dataset represents common problems in citizen submissions, such as vague descriptions, missing context, unclear objectives, and weak structure. This dataset helped define the expected style and structure of improved proposal outputs.

```
{
  "bad_input": "melhorar atravessamentos perto da escola matilde rosa araujo",
  "formal_output": {
    "title": "Melhoria dos Atravessamentos no Percurso para a Escola E.B.S. Matilde Rosa Araújo",
    "description": "A proposta consiste na identificação e requalificação de atravessamentos na
proximidade da escola...",
    "dynamic_fields": {
      "thematic_area": "Rede viária",
      "estimated_value": 350000
    }
  }
}
```

Figure 31 - Proposal Rewrite Dataset Structure Example

5.9.4 LoRA Training Dataset

The LoRA training dataset was created to adapt the model to the participatory budgeting domain. It contains 650 instruction examples divided into training, validation, and test splits.

The dataset covers the main AI-assisted tasks implemented in the prototype:

- regulatory validation.
- proposal improvement.
- semantic convergence classification.

The examples were designed to teach the model to detect incorrect thematic areas, unsuitable target audiences, unrealistic budgets, regulatory violations, vague descriptions, and relationships between proposals. The dataset was intentionally focused on behaviour expected from the application rather than on general text generation.

```
{
  "messages": [
    {
      "role": "system",
      "content": "You are a municipal assistant that evaluates whether retrieved proposals are similar, complementary, both, or irrelevant. Return valid JSON only."
    },
    {
      "role": "user",
      "content": "You are a municipal assistant. A citizen is submitting a new proposal.\nYour task is to evaluate candidate existing proposals retrieved by vector search..."
    },
    {
      "role": "assistant",
      "content": "{\n  \"proposals\": [\n    {\n      \"id\": \"1\", \"is_relevant\": true, \"relationship_type\": \"similar\", \"similarity_percentage\": 88.0, \"complementarity_percentage\": 45.0, \"reason\": \"'Iluminação junto à escola' aborda uma intervenção muito próxima da nova proposta.\\n\", \"convergence_rationale\": \"As propostas 'Iluminação junto à escola', 'Passadeiras elevadas perto da escola' não é a mesma intervenção, mas pode reforçar o impacto prático da nova proposta.\\n\", \"convergence_rationale\": \"As propostas 'Iluminação junto à escola', 'Passadeiras elevadas perto da escola' são úteis porque ajudam a comparar intervenções semelhantes ou medidas que reforçam o mesmo objetivo municipal.\\n\"}\n  ]\n}"
    }
  ],
  "task": "similarity_convergence",
  "meta": {
    "source": "synthetic_convergence",
    "category": "similarity_complementarity",
    "locale": "pt"
  }
}
```

Figure 32 - LoRA Training Dataset Structure Example

5.10 Model Adaptation and Training

The prototype supports open-source LLMs executed locally through Ollama. Three candidate models were initially considered: Gemma 4, Llama 3.1 8B Instruct, and Mistral. These models were selected because they are open-source, relatively small, suitable for local execution, and provide a favourable balance between performance and computational cost.

Initial manual tests indicated that Gemma 4 produced the strongest baseline results, followed by Llama 3.1 and Mistral. Llama 3.1 8B Instruct was selected for LoRA adaptation to evaluate whether parameter-efficient fine-tuning could improve a competitive base model and potentially allow it to match or surpass the strongest untuned baseline.

5.10.1 Preliminary Training Experiments

Several preliminary fine-tuning experiments were conducted before defining the final training strategy. The first experiment used a broad proposal-review dataset containing many valid proposal examples. Although this helped the model learn the expected JSON structure and basic rewriting behaviour, it introduced an imbalance: the model became overly permissive and frequently preserved user-submitted values even when the thematic area, target audience, or estimated budget were inconsistent with the proposal content.

A second training phase used a more problem-focused dataset. This dataset included incorrect budgets, unsuitable thematic areas, wrong target audiences, private or commercial benefits, and other problematic cases. This improved the model’s ability to detect isolated issues.

However, these experiments also revealed a deeper architectural problem. The original workflow expected the model to validate regulations, review text, correct dynamic fields, and suggest improvements in a single step. This caused the model to mix responsibilities and produce ambiguous outputs. As a result, the workflow was redesigned into two separate stages: regulatory validation and proposal improvement.

5.10.2 Final LoRA Training Strategy

The final LoRA training used Llama 3.1 8B Instruct as the base model. Training followed a supervised fine-tuning approach using instruction examples in chat format. The main training configuration is summarised in Table 23.

Table 23 - Main LoRA Configuration Parameters

Parameter	Value	Description
LoRA rank	16	Dimensionality of the low-rank matrices; controls adaptation capacity and memory usage.
LoRA alpha	32	Scaling factor applied to LoRA updates, influencing training stability and expressiveness.
Epochs	3	Number of full passes over the training dataset.
Learning rate	2×10^{-4}	Controls the magnitude of weight updates.

Parameter	Value	Description
Precision	4-bit model loading	Reduces VRAM consumption by loading the base model in 4-bit quantisation.
Optimiser	Paged AdamW 8-bit	Memory-efficient optimiser suitable for low-precision training.
Scheduler	Cosine	Gradually decays the learning rate following a cosine curve.

Training targeted the attention and feed-forward projection layers, allowing the model to be adapted with limited computational resources while preserving the base model weights.

Instead of using explicit controllable-generation mechanisms, the desired style was learned through supervised examples. The reference outputs consistently used structured, formal, citizen-oriented wording, with clear identification of the proposal objective and expected community benefit.

```
[venv] @nvidia:~/Riguel/home/elei/ai-training$ venv/bin/python train_lora.py
# Unsloth: Will patch your computer to enable 2x faster free finetuning.
Unsloth: Your Flash Attention 2 installation seems to be broken. Using Xformers instead. No performance changes will be seen.
# Unsloth Zoo will now patch everything to make training faster!
Loading model and tokenizer...
--((--))-- Unsloth 2024.6.1: Fast Llama patching. Transformers: 5.0.0
          \ NVIDIA GeForce RTX 4070. Num GPUs = 1. Max memory: 21.94 GB. Platform: Linux.
          / Torch: 2.10.0+cu120. CUDA: 8.9. CUDA Toolkit: 12.8. Triton: 3.0.0
          \ BF16+FP8 = YML. FA [Xformers = 0.0.35. FA2 = False]
          / Free license: http://github.com/unslothai/unsloth
          \
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!
Loading weights: 100%
Unsloth: Will load Unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit as a legacy tokenizer.
Setting up LoRA adapter config...
Unsloth 2024.6.1 patched 32 layers with 32 QKV layers, 32 O layers and 32 MLP layers.
Loading and splitting dataset...
Mapping chat template to datasets...
Initializing SFTTrainer...
warmup_ratio is deprecated and will be removed in v5.2. Use 'warmup_steps' instead.
# Unsloth: Padding-free auto-enabled, enabling faster training.
Starting training...
--((--))-- Unsloth - 2x faster free finetuning | Num GPUs used = 1
          \ Num examples = 1,020 | Num Epochs = 3 | Total steps = 306
          / Batch size per device = 1 | Gradient accumulation steps = 16
          \ Data Parallel: GPUs = 1 | Total batch size (1 x 16 x 3) = 48
          / Trainable parameters = 41,943,040 of 8,072,204,288 (0.52% trained)
          \
[0%] | 0/306 [00:00:07, 71it/s]
use_return_dict is deprecated! Use 'return_dict' instead!
Unsloth: Double buffering enabled (parallel H2O + compute) for backward pass.
Unsloth: Will smartly offload gradients to save VRAM!
{'loss': '2.254', 'grad_norm': '2.254', 'learning_rate': '0.0001125', 'epoch': '0.00777'}
{'loss': '1.514', 'grad_norm': '1.127', 'learning_rate': '0.0001999', 'epoch': '0.1975'}
[5%] | 15/306 [02:45:38:08, 0.44s/it]
Unsloth: Restored added_tokens_decoder metadata in ./outputs/checkpoint-25/tokenizer_config.json.
{'loss': '1.09', 'grad_norm': '0.6725', 'learning_rate': '0.000199', 'epoch': '0.2963'}
{'loss': '1.167', 'grad_norm': '0.6364', 'learning_rate': '0.0001909', 'epoch': '0.3951'}
{'loss': '1.178', 'grad_norm': '1.129', 'learning_rate': '0.0001927', 'epoch': '0.4938'}
{'eval_loss': '1.061', 'eval_runtime': '15.19', 'eval_samples_per_second': '11.85', 'eval_steps_per_second': '2.963', 'epoch': '0.4938'}
[10%] | 30/306 [05:40:28:54, 0.78s/it]
Unsloth: Restored added_tokens_decoder metadata in ./outputs/checkpoint-50/tokenizer_config.json.
{'loss': '1.075', 'grad_norm': '0.6455', 'learning_rate': '0.0001893', 'epoch': '0.5926'}
{'loss': '1.171', 'grad_norm': '0.4245', 'learning_rate': '0.000184', 'epoch': '0.6914'}
[20%] | 60/306 [08:38:26:04, 0.77s/it]
Unsloth: Restored added_tokens_decoder metadata in ./outputs/checkpoint-75/tokenizer_config.json.
{'loss': '1.091', 'grad_norm': '0.4381', 'learning_rate': '0.0001776', 'epoch': '0.7901'}
{'loss': '1.08', 'grad_norm': '0.4307', 'learning_rate': '0.0001747', 'epoch': '0.8889'}
{'loss': '1.08', 'grad_norm': '0.4082', 'learning_rate': '0.0001622', 'epoch': '0.9877'}
{'eval_loss': '1.056', 'eval_runtime': '15.06', 'eval_samples_per_second': '11.95', 'eval_steps_per_second': '2.987', 'epoch': '0.9877'}
[30%] | 90/306 [11:39:22:23, 0.81s/it]
Unsloth: Restored added_tokens_decoder metadata in ./outputs/checkpoint-100/tokenizer_config.json.
{'loss': '0.9848', 'grad_norm': '0.4549', 'learning_rate': '0.0001534', 'epoch': '1.079'}
{'loss': '0.9525', 'grad_norm': '0.4722', 'learning_rate': '0.0001439', 'epoch': '1.178'}
[40%] | 120/306 [14:20:19:36, 0.58s/it]
```

Figure 33 - LoRA Training Output Example

5.11 Robustness and Error Handling

The implementation includes several mechanisms to improve robustness during AI-assisted interactions.

First, all AI responses are expected to follow predefined JSON structures. Before returning a response to Laravel, the AI service parses and validates the output. If the model produces malformed JSON, the service attempts to repair it or returns a controlled fallback response.

Second, the Laravel interface prevents concurrent AI operations by showing loading states and disabling repeated actions while requests are being processed. This avoids duplicate requests and reflects the fact that LLM inference may take several seconds, especially when prompts include rules, schemas, and retrieved examples.

Third, the AI service uses configurable timeouts and model parameters. This allows the system to be tuned for different hardware conditions and model sizes.

Fourth, the system is designed to fail safely. Regulatory validation warnings do not represent final administrative rejection, improvement suggestions are not automatically applied, and convergence suggestions do not force the citizen to merge proposals.

These mechanisms also contribute to compliance with the transparency and human oversight principles of the AI Act by ensuring that AI-generated outputs remain advisory, explainable, and subject to human decision-making throughout the proposal submission process.

6 Experimental Evaluation and Discussion

This chapter presents the experimental evaluation of the proof of concept developed in this dissertation. The objective is to assess whether the implemented AI-based mechanisms effectively support citizens during proposal creation by improving proposal quality, identifying regulatory issues, and detecting similar or complementary proposals before submission.

The evaluation combines automatic testing with expert assessment. Automatic tests measure the technical behaviour of the implemented AI pipelines under different model configurations, while the expert assessment evaluates the usefulness, clarity, and suitability of the generated recommendations within the context of real participatory processes.

6.1 Evaluation Objectives

The main objective of the evaluation is to determine whether the proposed solution can provide useful AI-assisted support for digital participatory processes. In particular, the evaluation focuses on two central mechanisms: proposal review and semantic convergence.

The proposal review mechanism is evaluated according to its ability to support citizens in improving the clarity, structure, and coherence of their proposals. On the other hand, the semantic convergence mechanism is evaluated according to its ability to identify proposals that are similar, complementary, or potentially related to the proposal being created.

The evaluation also examines the contribution of the main AI techniques used in the prototype, namely open-source LLMs, Retrieval-Augmented Generation, embeddings, and LoRA-based model adaptation. Rather than assessing these techniques in isolation, the chapter analyses how they influence the overall behaviour of the implemented system.

The evaluation is therefore guided by four main questions:

1. Does the proposal review mechanism improve the clarity and structure of citizen proposals?

2. Does the system correctly identify regulatory problems and dynamic-field inconsistencies?
3. Does RAG improve the contextual accuracy of the model when applying process-specific rules?
4. Does the semantic convergence mechanism identify related proposals in a way that is useful for participatory processes?

6.2 Research Hypotheses

The experimental evaluation is guided by the following hypotheses:

- **H1 - Proposal review improves proposal quality:** The AI-based writing support mechanism improves the clarity, structure, and completeness of citizen proposals while preserving their original intent.
- **H2 - LoRA adaptation improves domain adequacy:** Adapting the base language model through LoRA improves its ability to follow the expected proposal review behaviour, apply participatory-process reasoning, and produce structured outputs.
- **H3 - RAG improves contextual accuracy:** Retrieval-Augmented Generation improves the model's ability to apply process-specific information, particularly regulations, eligibility criteria, and budget constraints.
- **H4 - Semantic convergence identifies related proposals effectively:** The combination of embeddings, vector retrieval, and LLM-based verification can identify proposals that are similar or complementary, helping reduce duplicate submissions and encouraging the aggregation of related ideas.

6.3 Evaluation Methodology

The evaluation methodology combines automatic evaluation and expert assessment. This mixed approach was adopted because the proposed solution contains both technical and human-centred components. Automatic tests are appropriate for measuring structured behaviours, such as JSON validity, eligibility classification, dynamic-field correction, and retrieval performance. Expert assessment complements these results by evaluating qualities that are difficult to measure automatically, including proposal clarity, usefulness of the recommendations, and suitability for real participatory processes.

6.3.1 Evaluation Design

The automatic evaluation was organised into three complementary groups, each addressing a different aspect of the implemented solution.

Table 24 - Automatic Evaluation Test Groups

Test Group	Purpose
Proposal review and regulatory validation	Evaluates regulatory validation, proposal improvement, dynamic-field correction, budget correction, and structured output compliance
Semantic convergence evaluation	Evaluates whether the system retrieves and classifies similar or complementary proposals correctly
RAG and LoRA comparison	Compares the effect of RAG and LoRA adaptation across different configurations

The tested models were:

- Gemma4:latest
- Llama3.1:8b
- Mistral:latest
- Llama3.1-8b-pb-expert:latest (LoRA-adapted model)

To analyse the contribution of retrieval and model adaptation, three execution configurations were compared.

Table 25 - Comparative Evaluation Configurations

Configuration	Description
Base without RAG	Base Llama 3.1 model without retrieval context
Base + RAG	Base Llama 3.1 model with process-specific retrieved context
LoRA + RAG	LoRA-adapted Llama 3.1 model with process-specific retrieved context

This design made it possible to evaluate not only which model performed best, but also whether the combination of RAG and LoRA provided measurable benefits over the base model.

6.3.2 Evaluation Datasets and Metrics

The proposal review evaluation used 20 test cases per model. These cases covered regulatory validation, proposal improvement, dynamic-field correction, budget correction, and JSON/schema compliance. The main metrics were pass rate, average processing time, eligibility accuracy, precision, recall, and F1-score.

The semantic convergence evaluation used 15 test cases per model. Fourteen cases contained expected similar or complementary proposals, while one case expected no relevant match. The evaluation measured ranking and retrieval quality using Precision@K, Recall@K, F1@K, Mean

Reciprocal Rank (MRR), Normalised Discounted Cumulative Gain (NDCG@K), relationship accuracy, and false positive rate.

The comparative evaluation compared the base model without RAG, the base model with RAG, and the LoRA-adapted model with RAG. This evaluation focused on regulatory pass rate, eligibility F1-score, improvement pass rate, convergence metrics, and relationship accuracy.

6.3.3 Expert Evaluation

The expert evaluation was conducted with four professionals with experience in participatory processes, participatory budgeting, public administration, and civic engagement. The purpose of this evaluation was to assess whether the system outputs were useful, understandable, and suitable for real participatory-process contexts. Table 26 summarises the expert profiles.

Table 26 - Expert Profiles

Expert	Role	Institution	Experience in Participation
Timóteo Barrocas	Senior Technician	Lisbon City Council	10 years of experience managing participatory processes
Filipe Casimiro	Head of Division, Participation Division	Lisbon City Council	7 years of experience managing participatory processes, including participatory budgeting, citizens' assemblies, and public consultations
Simone Júlio	Project Manager	Associação Oficina	13 years of experience in participatory democracy, participatory budgeting, process evaluation, municipal capacity building, EMPATIA, PHOENIX, and CoParticipation
Vânia Martins	Project Manager	Associação Oficina	12 years of experience in citizenship, participation, sustainability, participatory budgeting, PHOENIX, CoParticipation, and municipal participatory-process design

The expert evaluation had two components:

- **A 5-point Likert-scale questionnaire**, where experts assessed usefulness, proposal quality, regulatory validation, semantic convergence, transparency, user interaction, and adoption potential.

- **A blind A/B test**, where experts compared original citizen-style proposals with AI-improved versions without knowing which version had been generated by the system.

Before completing the questionnaire, the experts were introduced to the proof of concept and to its main workflows.

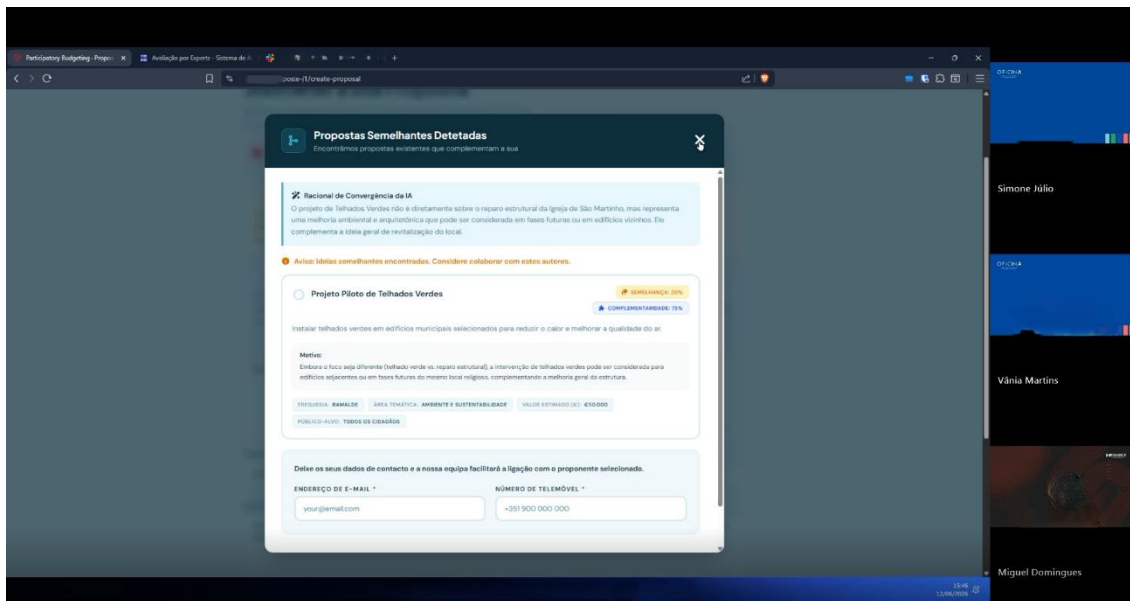


Figure 34 - Demonstration and Evaluation Session with Domain Experts

6.4 Automated Evaluation Results

The automatic evaluation results are presented in this section and are organised according to the three main test groups: proposal review and regulatory validation, semantic convergence, and RAG/LoRA comparison.

6.4.1 Proposal Review and Regulatory Validation

The proposal review evaluation included 20 test cases per model, covering regulatory validation, proposal improvement, dynamic-field correction, budget correction, and structured-output compliance. Table 27 presents the results.

Table 27 - Proposal Review Automated Evaluation Results

Model	Passed	Pass Rate	Avg. Time	Eligibility Accuracy	Precision	Recall	F1
gemma4:latest	15/20	0.75	3.337s	0.90	0.8889	1.0000	0.9412
llama3.1:8b	15/20	0.75	4.282s	0.90	0.8889	1.0000	0.9412
mistral:latest	14/20	0.70	2.824s	0.70	1.0000	0.6250	0.7692
llama3.1-8b-pb-expert:latest	19/20	0.95	2.269s	1.00	1.0000	1.0000	1.0000

The LoRA-adapted model (llama3.1-8b-pb-expert:latest) achieved the best results, correctly passing 19 out of 20 test cases (95%). It also reached perfect eligibility classification, with accuracy, precision, recall, and F1-score all equal to 1.0. This shows that the adapted model was the most reliable for the proposal-review workflow.

Both llama3.1:8b and gemma4:latest reached the same overall pass rate (75%) and an eligibility F1-score of 0.9412. They correctly identified all non-eligible proposals (recall = 1.0), but each incorrectly marked one eligible proposal as non-eligible. This may generate unnecessary warnings, although it is less critical than accepting ineligible proposals.

The weakest results were obtained by mistral:latest, with a pass rate of 70% and an eligibility F1-score of 0.7692. Although it achieved perfect precision, its lower recall (0.6250) means that some non-eligible proposals were incorrectly accepted, making it the least suitable model for regulatory validation.

All models produced valid JSON responses and followed the expected output format in every test case, showing that the prompt structure and validation mechanisms ensured stable integration with the web application.

The biggest differences appeared in dynamic-field correction. The LoRA-adapted model achieved a perfect correction rate, gemma4:latest reached 0.75, and both llama3.1:8b and mistral:latest reached 0.50. In contrast, all models handled budget validation correctly, with a budget-check rate of 1.0.

Overall, the results show that the proposal-review workflow works reliably and that LoRA adaptation clearly improves domain-specific reasoning and dynamic-field correction.

6.4.2 Semantic Convergence Evaluation

The semantic convergence evaluation measured the system's ability to retrieve and classify related proposals. The test set included proposals that were similar, complementary, both, or unrelated. Table 28 presents the results.

Table 28 - Semantic Convergence Automated Evaluation Results

Model	Precision @K	Recall @K	F1@K	MRR	NDCG @K	Relationship Accuracy	False Posit. Rate
gemma4:latest	0.7857	0.9406	0.8310	0.9286	0.9029	1.0000	0.000
llama3.1:8b	0.8095	0.9762	0.8548	0.9643	0.9569	0.8214	0.000
mistral:latest	0.8571	0.9405	0.8667	0.9643	0.9292	1.0000	0.000
llama3.1-8b-pb-expert:latest	0.7857	0.9405	0.8310	0.8929	0.8765	0.9643	0.000

All models achieved a false-positive rate of 0.0 when no related proposal existed, showing that the system does not create unnecessary convergence suggestions.

The mistral:latest model achieved the highest Precision@K (0.8571) and the highest F1@K (0.8667), giving the best balance between retrieving useful candidates and avoiding irrelevant ones.

The base llama3.1:8b model achieved the highest Recall@K (0.9762) and NDCG@K (0.9569), meaning it retrieved more relevant proposals and ranked them well. However, its relationship-classification accuracy (0.8214) was lower, showing that retrieving good candidates does not always mean correctly identifying their relationship.

The gemma4:latest model achieved perfect relationship-classification accuracy (1.0), and the LoRA-adapted model reached 0.9643. Because retrieval mainly depends on embeddings and vector search, these results suggest that the language model's main role is to interpret and classify the retrieved proposals rather than retrieve them.

Overall, the results show that combining vector retrieval with LLM-based verification provides accurate semantic convergence support. The vector database retrieves candidate proposals efficiently, and the language model determines whether they are similar, complementary, both, or unrelated.

6.4.3 RAG and LoRA Comparative Evaluation

The comparative evaluation analysed three configurations: base Llama without RAG, base Llama with RAG, and LoRA-adapted Llama with RAG. The objective was to understand the contribution of retrieval and model adaptation to the final system behaviour.

Table 29 - RAG and LoRA Comparative Evaluation Results

Configuration	Regulatory Pass Rate	Eligibility F1	Improvement Pass Rate	Conv. Precision@K	Conv. Recall@K	Conv. F1@K	Conv. NDCG@K	Relationship Accuracy
Base without RAG	0.70	0.7692	0.6	0.8095	0.9762	0.8548	0.9569	-
Base + RAG	0.90	0.9412	0.50	0.8095	0.9762	0.8548	0.9569	0.7857
LoRA + RAG	1.00	1.0000	0.70	0.7857	0.9405	0.8310	0.9556	0.9643

Regulatory validation results showed a clear benefit from RAG. Without retrieval, the base Llama model achieved a regulatory pass rate of 0.70 and an eligibility F1-score of 0.7692. With RAG enabled, these values increased to 0.90 and 0.9412, showing that process-specific context improves regulatory reasoning. The best results came from the LoRA + RAG configuration, which achieved a regulatory pass rate and eligibility F1-score of 1.00. This confirms that RAG and LoRA work well together: RAG provides the necessary context during inference, and LoRA helps the model interpret and apply that context more consistently.

Proposal-improvement results were more modest. The base model without RAG reached a pass rate of 0.60, compared with 0.50 for the base model with RAG and 0.70 for the LoRA-adapted model. This suggests that extra context alone does not always help writing-support tasks and may introduce noise unless the model has been adapted to separate regulatory information from writing guidance.

Semantic convergence remained stable across configurations because retrieval mainly depends on embeddings and vector similarity. Even so, the LoRA-adapted model achieved the highest relationship-classification accuracy (0.9643), compared with 0.7857 for the base model with RAG, showing that LoRA improves how the model interprets relationships after retrieval. The configuration without RAG was not evaluated for relationship-accuracy because candidate retrieval depends on it.

Overall, the results show that RAG is especially useful for regulatory validation, while LoRA improves domain-specific reasoning and relationship classification.

6.4.4 Non-Functional Requirements Evaluation

In addition to evaluating the functional behaviour of the intelligent mechanisms, complementary tests were conducted to verify whether the prototype satisfies the non-functional requirements defined in Chapter 3. The evaluation covered response time, software quality, maintainability, dependency security, and web usability.

The backend was assessed using automated quality tools, including pytest-cov, Coverage.py, Radon, Ruff, Pylint, Vulture, and pip-audit. The web application was evaluated using Lighthouse for both desktop and mobile environments, measuring performance, accessibility, best practices, SEO, and responsiveness.

Response Time Evaluation

Response time was evaluated for the three main AI-assisted endpoints: regulatory validation, proposal improvement, and semantic convergence detection. Each endpoint was tested with the four selected models: llama3.1:8b, llama3.1-8b-pb-expert:latest, gemma4:latest, and mistral:latest. For each model-endpoint pair, one warm-up request and three measured requests were executed. Figure 35 presents the obtained results.

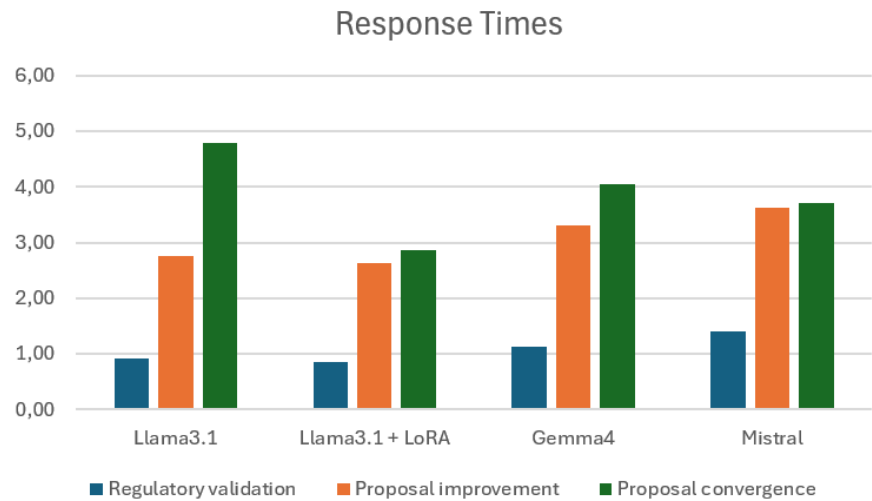


Figure 35 - Response Time Evaluation Results

All requests completed successfully, resulting in a 100% success rate across all tested model and endpoint combinations. Regulatory validation was consistently the fastest operation (0.862 - 1.409 s), followed by proposal improvement (2.643 - 3.622 s). Semantic convergence required the longest execution times because it combines vector retrieval with LLM-based verification, but every average remained below the defined five-second threshold.

Among the evaluated models, the LoRA-adapted Llama 3.1 achieved the lowest response time in all three endpoints, demonstrating that model adaptation did not introduce additional latency.

Software Quality and Maintainability Evaluation

The Python AI service was also evaluated using static analysis, test coverage, and maintainability tools. The results are summarised in Table 30.

Table 30 - Code Quality and Maintainability Results

Evaluation Aspect	Tool	Result
Application code quality	Pylint	10.00/10
Linting and static code quality	Ruff	All checks passed
Dead-code detection	Vulture	No high-confidence dead code detected
Average cyclomatic complexity	Radon	A
Full repository coverage	Coverage.py / pytest-cov	44%
Deterministic core coverage	Coverage.py / pytest-cov	95%

The application achieved the maximum Pylint score (10.00/10), while Ruff reported no linting issues and Vulture detected no high-confidence dead code. In addition, Radon classified the average cyclomatic complexity as Grade A, indicating low structural complexity across most modules. Some orchestration-heavy modules, particularly those related to RAG and proposal review, naturally present greater complexity because they coordinate document ingestion, rule extraction, vector retrieval, prompt construction, LLM calls, and response normalisation. Nevertheless, the overall result supports the maintainability of the implementation.

The automated test coverage for the deterministic core logic of the AI service (proposal service, similarity service, JSON utilities and text utilities) achieved 95% coverage, demonstrating that the most critical processing logic is extensively validated through automated tests.

Dependency Security Evaluation

Dependency security was evaluated using pip-audit. The tool reported one vulnerability associated with the ChromaDB dependency used by the prototype.

Table 31 - Package Vulnerabilities

Package	Version	Reported Vulnerability	Fix Available
Chromadb	1.5.9	CVE-2026-45829	No fixed version listed

The reported issue targets ChromaDB server endpoints. Since the prototype uses ChromaDB only as a local vector store accessed internally by the AI service, the practical impact is limited. Nevertheless, future deployments should monitor dependency updates and apply the corresponding security patches when available.

Web Interface Quality and Usability Evaluation

The Laravel and Livewire web application was evaluated using Lighthouse in both desktop and mobile environments. The desktop evaluation obtained the following scores:



https://app.localhost/1/create-proposal

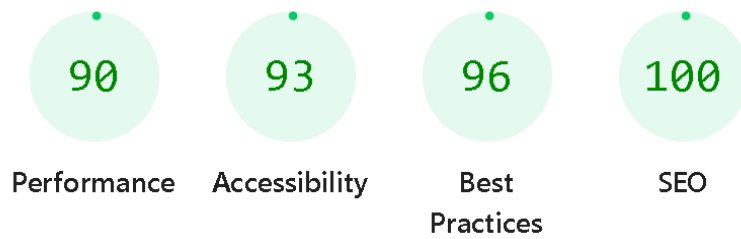


Figure 36 - Lighthouse Desktop Results

The mobile evaluation produced similar results, indicating that the interface remains usable and responsive across different screen sizes. These results support the usability and accessibility dimensions of the non-functional requirements, particularly because the proposal creation workflow is intended to be used by citizens with different levels of technical proficiency.

The desktop layouts, previously presented in Chapter 5, illustrate how the proof of concept is displayed in a clear and readable interface. The corresponding mobile layouts are included in Appendix H and confirm that the user interface remains responsive and functional across different screen sizes. Together, these observations provide additional evidence that the prototype supports the intended proposal-submission workflow on multiple devices.

Although Lighthouse does not replace user testing, the obtained results provide useful supporting evidence that the web interface satisfies basic quality expectations regarding performance, accessibility, best practices, and responsiveness.

Mapping to Non-Functional Requirements

The results presented in the previous sections were mapped to the non-functional requirements defined during the analysis phase. Table 32 summarises the outcome of this evaluation.

Table 32 - Mapping Evaluation to Non-Functional Requirements

Identifier	Non-Functional Requirement	Evaluation Source	Assessment
NFR1	Extensibility	Architectural and implementation evaluation	Satisfied
NFR2	Modularity	Architectural and implementation evaluation	Satisfied
NFR3	Explainability	Expert evaluation and structured AI responses	Satisfied
NFR4	Response Time	Endpoint benchmark results	Satisfied
NFR5	Maintainability	Static analysis and automated testing	Satisfied
NFR6	Adaptability	Model adaptation and configuration evaluation	Satisfied
NFR7	Usability	Lighthouse evaluation and expert assessment	Satisfied

Overall, the evaluation demonstrates that the prototype satisfies all defined non-functional requirements. The AI service meets the required response-time threshold, presents good maintainability and code quality indicators, and achieves high automated test coverage over its deterministic logic. In addition, the Lighthouse assessment and expert evaluation provide evidence that the web application offers a responsive and usable interface suitable for the objectives of this proof of concept.

6.5 Expert Evaluation Results

The expert evaluation complements the automatic experiments by assessing the practical usefulness of the proposed solution from the perspective of professionals with extensive experience in participatory processes. Since proposal quality depends not only on technical correctness but also on clarity, feasibility, and administrative relevance, expert judgement provides an important validation of the prototype.

6.5.1 Questionnaire Results

The questionnaire was completed by four domain experts. Results were highly positive across all evaluated dimensions, particularly writing support, transparency, regulatory validation, and adoption potential. Table 33 summarises the average scores by category.

Perguntas
Respostas
4
Definições

Este formulário não aceita respostas.
Gerir

Secção 1 de 9

Avaliação por Experts - Sistema de Apoio à Submissão de Propostas Participativas

Este questionário tem como objetivo recolher a opinião de experts sobre uma solução baseada em IA para apoiar cidadãos na criação, revisão e melhoria de propostas em processos participativos. A avaliação inclui a utilização livre da solução e uma comparação A/B entre propostas originais e versões melhoradas pela IA.

Instruções

Por favor, avalie a solução após interagir livremente com o formulário de criação de propostas. Nas questões de escala, 1 representa a avaliação mais negativa e 5 a avaliação mais positiva.

Figure 37 - Questionnaire Answers

Table 33 - Average Scores per Category

Evaluation Category	Average Score	Main Interpretation
General usefulness and adequacy	4.69 / 5	Considered useful and appropriate for municipal participatory processes
Writing support	4.85 / 5	Strong agreement that the system improves clarity, structure, and comprehensibility
Regulatory validation	4.81 / 5	Effective at identifying ineligibility and explaining regulatory issues
Semantic convergence	4.55 / 5	Useful functionality, with some room for improvement in relevance and distinction between similarity and complementarity
Transparency and user interaction	4.83 / 5	Transparent, simple to use, and appropriate for non-technical citizens
Global evaluation	5.00 / 5	All experts assigned the maximum overall rating

All experts agreed that the solution is useful for supporting citizens during proposal creation and that it can help improve the quality of submitted proposals. Writing support received particularly strong feedback: all experts gave the maximum score to the preservation of the original intent and to the clarity and structure of the improved versions.

The regulatory validation mechanism also received strong support. All experts gave the maximum score to the system's ability to identify ineligibility and to provide clear justifications.

This is a significant result, as regulatory validation is a sensitive task that must be helpful without being perceived as an automatic final decision.

The semantic convergence mechanism was positively evaluated as well. Experts agreed that it can help reduce duplicate proposals and encourage the aggregation of related ideas. However, this category had the lowest average score (4.55). The relevance of retrieved proposals received 4.25, and the clarity of the distinction between similar and complementary proposals received 4.5. These results indicate that the functionality is valuable but that the explanation and presentation of relationships can still be refined.

Transparency and user interaction were also highly rated. All experts gave the maximum score to the statement that the system should be used as support rather than as an automatic decision-maker, and to the transparency of the explanations provided. This reinforces the design choice of offering clear justifications and maintaining citizen control over final decisions. Experts were also asked which types of improvements they considered most useful. The most frequently selected categories were:

Table 34 - Most Selected Improvements

Improvement Type	Number of Experts
Textual clarity	4
Proposal structure	3
Grammatical correction	3
Thematic-area validation	3
Budget validation	2
Form adequacy	1

These results indicate that experts value both linguistic improvements and process-specific validation. The system is therefore useful not only as a text rewriting tool, but also as a contextual proposal-support mechanism.

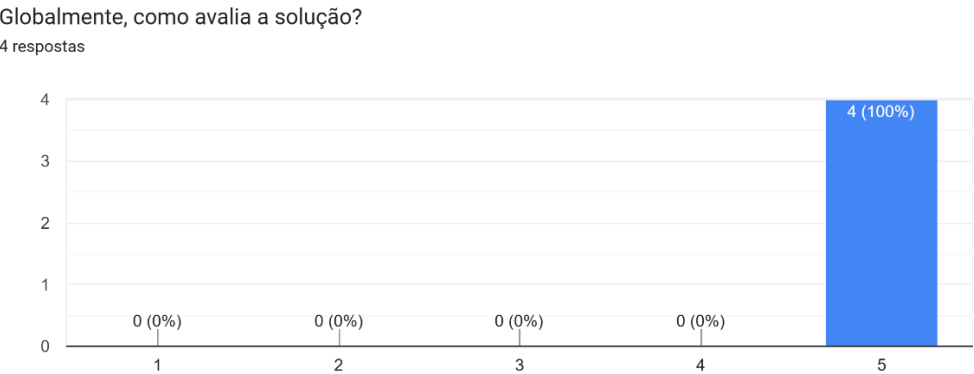


Figure 38 - Global Evaluation Questionnaire Answers

6.5.2 Blind A/B Testing Results

The blind A/B test asked experts to compare original proposal formulations with AI-improved versions. Each pair was anonymised as Option A and Option B, ensuring that evaluators did not know which version had been generated by the system. Across 15 proposal pairs and 4 evaluators, the test produced 60 individual assessments. In 58 of these, experts selected the AI-improved version as the better formulation, corresponding to a preference rate of 96.7%.

Table 35 - Blind A/B Testing Results

Result	Value
Number of proposal pairs	15
Number of experts	4
Total evaluations	60
AI-improved version selected	58
Original version selected	2
Expert preference for AI-improved version	96.7%

Opção A

- **Título:** Um Mundo de Igualdades

- **Descrição:** Projeto dirigido a escolas de todos os níveis de ensino contemplando a realização de palestras sobre racismo, desigualdades sociais, xenofobia, preconceito, homofobia, bullying e violência doméstica.

Opção B

- **Título:** Um Mundo de Igualdades: Palestras sobre Preconceito e Direitos Humanos em Escolas

- **Descrição:** Proposta de realização de palestras educativas em escolas de todos os níveis de ensino. O objetivo é sensibilizar a comunidade escolar para temas cruciais como racismo, desigualdades sociais, xenofobia, preconceito, homofobia, bullying e violência doméstica, promovendo um ambiente mais inclusivo e respeitoso.

4 respostas

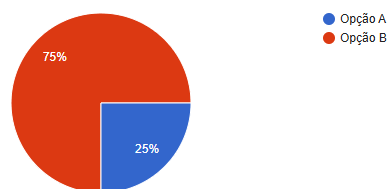


Figure 39 - A/B Testing Question Example

Three of the four experts selected the AI-improved version in all 15 pairs. The remaining expert selected it in 13 out of 15. The two cases where the original proposal was preferred illustrate that proposal improvement remains a subjective task and that a more complete or formal version is not always perceived as superior in every context.

Overall, the results provide evidence that the proposal-improvement mechanism produces text that domain experts consider clearer, more structured, and more suitable for participatory-process workflows.

6.6 Discussion

The evaluation results show that the proposed solution can effectively support citizens during proposal creation and contribute to clearer, more structured proposals and more consistent semantic convergence. This section discusses the findings in relation to the research hypotheses.

6.6.1 Analysis of Research Hypotheses

H1 - Proposal review improves proposal quality

H1 is supported. Both the automatic evaluation and the expert assessment support this hypothesis. The proposal-review mechanism consistently improved proposal quality while preserving the original intent, as demonstrated by the high proposal-review scores and the strong preference for the AI-improved proposals in the blind A/B evaluation.

H2 - LoRA adaptation improves domain adequacy

H2 is supported. The LoRA-adapted model achieved the best overall performance in proposal review and regulatory validation, demonstrating superior domain-specific reasoning and dynamic-field correction. Its advantages were less pronounced in retrieval-based tasks, where performance depends primarily on embeddings and vector search.

H3 - RAG improves contextual accuracy

H3 is supported. The results show that Retrieval-Augmented Generation substantially improves regulatory validation by providing process-specific context during inference. However, the proposal-improvement evaluation also indicates that additional context alone is insufficient unless the model has been adapted to use it effectively.

H4 - Semantic convergence identifies related proposals effectively

H4 is supported. The semantic convergence mechanism successfully retrieved and classified related proposals while avoiding false positives. Expert feedback also confirmed its usefulness, although improvements to the presentation of relationships between proposals could further enhance the user experience.

6.6.2 Main Findings

The evaluation produced six main findings.

- The two-stage proposal-review workflow successfully separates regulatory validation from writing improvement, producing more consistent and interpretable results.
- Structured JSON outputs enabled reliable integration between the AI service and the web application across all evaluated models.
- The combination of RAG and LoRA produced the strongest regulatory-validation performance, confirming that both techniques provide complementary benefits.
- Expert evaluation demonstrated that the generated recommendations are useful, understandable, and appropriate for participatory-process workflows.
- The proposed interaction model supports transparency and human oversight by presenting AI-generated recommendations while preserving citizen and administrative decision-making.
- The non-functional evaluation demonstrated that the prototype is technically robust, satisfying the defined requirements for response time, maintainability, adaptability, and usability.

6.7 Threats to Validity

Several threats to validity must be considered when interpreting the evaluation results.

- **Internal Validity:** The automatic evaluation uses manually created test cases and expected outputs, which reflect specific choices about what counts as a correct proposal review or semantic relationship. Although these criteria follow the goals of the system, some tasks involve subjective interpretation.
- **External Validity:** The evaluation focused on municipal participatory processes, mainly participatory budgeting. Because of this, the results may not fully generalise to other participation formats with different legal or organisational rules. The LoRA-adapted model was also trained for this domain. While the approach can be reused elsewhere, applying it to other municipalities, languages, or participation contexts would require additional adaptation and testing.
- **Construct Validity:** Proposal quality cannot be measured objectively with automated metrics alone. For this reason, the evaluation included expert judgement, such as the blind A/B comparison. Although experts strongly preferred the AI-improved proposals, these results represent perceived quality rather than an absolute measure.
- **Conclusion Validity:** The expert evaluation involved only four specialists, and the automatic tests used relatively small datasets. Although both evaluations provide consistent evidence supporting the solution, larger studies with more experts, citizens, and participatory processes would strengthen the statistical reliability of the conclusions.

7 Conclusions and Future Work

This chapter concludes the dissertation by summarising the work developed, presenting the main contributions, discussing the achievement of the dissertation's main objective, identifying the principal limitations, and outlining future work.

7.1 Research Overview

This dissertation addressed two major challenges in digital participatory processes: improving the quality of citizen proposals and reducing the fragmentation caused by similar or complementary submissions.

To address these challenges, a proof of concept was designed, implemented, and evaluated within the context of the CoParticipation project. The proposed solution integrates AI-assisted proposal review, regulatory validation, and semantic convergence into the proposal submission workflow through a Laravel-based web platform and a FastAPI AI service.

The evaluation combined automatic experiments with expert assessment, demonstrating both the technical feasibility of the proposed approach and its practical usefulness in participatory-process contexts.

7.2 Main Contributions

The main contributions of this dissertation are:

- The design and implementation of an AI-assisted architecture that integrates proposal review and semantic convergence into digital participatory platforms.
- A two-stage proposal-review workflow that separates regulatory validation from proposal improvement, producing more consistent and interpretable results.

- The integration of process-specific knowledge through Retrieval-Augmented Generation, allowing the language model to reason using regulations, proposal schemas, and historical proposals.
- A semantic convergence mechanism that identifies similar and complementary proposals before submission, encouraging collaboration and reducing idea fragmentation.
- An evaluation combining automatic testing, non-functional assessment, and expert validation, demonstrating the technical feasibility and practical usefulness of the proposed solution.

7.3 Achievement of the Main Objective

The main objective of this dissertation was to design, develop, and evaluate a proof-of-concept solution capable of supporting citizens during proposal creation in digital participatory processes through intelligent mechanisms based on NLP and LLMs.

The work developed throughout this dissertation demonstrates that this objective was successfully achieved. The proposed solution combines Retrieval-Augmented Generation (RAG), domain-adapted language models, semantic similarity techniques, and structured workflows to provide contextual assistance during proposal creation while preserving the citizen's original intent.

Contextual writing support can be achieved by analysing the proposal title, description, and dynamic fields in relation to the active participatory process. The prototype demonstrated that AI-generated suggestions can improve clarity, structure, and completeness while preserving the citizen's original intent.

Structural standardisation can be supported through schema-aware proposal review and structured AI responses. By validating dynamic fields, checking regulatory constraints, and returning results in predefined JSON formats, the system can provide consistent feedback that can be safely integrated into the web interface.

Semantic convergence can be supported through embeddings, vector retrieval, and LLM-based verification. This combination allows the system to retrieve related proposals and classify them as similar, complementary, both, or irrelevant. In this way, citizens can be informed about existing related ideas before submitting a new proposal.

In general, the results obtained demonstrate that intelligent mechanisms based on NLP and LLMs can effectively support contextual proposal writing, structural standardisation, and semantic convergence when they are grounded in process-specific knowledge, integrated into structured AI-assisted workflows. These mechanisms also function as transparent decision-support tools that preserve both citizen autonomy and the role of administrative evaluation.

7.4 Limitations

Although the obtained results are encouraging, this work presents several limitations.

First, the developed system is a proof of concept rather than a production-ready platform. Administrative features, operational monitoring, scalability, and deployment aspects were intentionally excluded from the dissertation scope.

Second, both the automatic and expert evaluations were conducted on relatively small samples. Larger datasets and broader participation would strengthen the statistical significance of the obtained results.

Third, the proposed approach was evaluated mainly in the context of municipal participatory budgeting. Applying it to other participatory processes may require additional adaptation of the language model, prompts, and process-specific knowledge.

Finally, semantic convergence remains partially subjective, since distinguishing between similar, complementary, and unrelated proposals depends on the objectives and context of each participatory process.

7.5 Future Work

Future work should begin with a broader evaluation involving more citizens, process managers, and municipal technicians. Testing the system in real participatory processes would provide stronger evidence about usability, acceptance, and impact on proposal quality.

Another important direction is the improvement of the semantic convergence mechanism. Future versions should refine the explanation of why proposals are considered similar or complementary and explore better ways of presenting convergence suggestions to citizens.

The collaboration workflow could also be expanded. In the current prototype, citizens may indicate an intention to collaborate, but the process is not fully developed. Future work could include mechanisms for contacting proposal authors, merging ideas, or supporting collaborative rewriting.

The administrative workflow could be extended with richer document-management capabilities, including rule inspection, manual validation, and knowledge-based management.

Finally, additional language models, embedding techniques, retrieval strategies, and training datasets should be evaluated to determine whether the observed results remain consistent across different technical configurations and participatory-process contexts. Integrating the developed prototype into the complete CoParticipation and EMPATIA ecosystem also represents a natural continuation of this work.

7.6 Final Remarks

This dissertation demonstrated that AI-based mechanisms can provide useful support during the creation of proposals in digital participatory processes. The proof of concept showed that proposal review, regulatory validation, and semantic convergence can be integrated into the submission workflow in a way that supports citizens without replacing their decisions.

The results indicate that the combination of structured prompts, RAG, embeddings, and LoRA adaptation can improve the behaviour of local open-source language models in this domain. Expert feedback further suggests that the system has practical value for improving proposal clarity, supporting regulatory awareness, and encouraging the aggregation of related ideas.

Beyond its technical contributions, the proposed architecture also demonstrates how AI-assisted mechanisms can be integrated into participatory platforms while respecting principles of transparency, human oversight, and responsible AI established by the AI Act.

In conclusion, the work contributes to the development of more supportive, transparent, and collaborative digital participation platforms. It also provides a foundation for future research and development within the CoParticipation project and the EMPATIA framework.

References

- ALLEA (2023) *The European Code of Conduct for Research Integrity*. DE: ALLEA - All European Academies. Available at: <https://doi.org/10.26356/ECOC> (Accessed: December 13, 2025).
- Alpdemir, Y. and Alpdemir, M.N. (2024) "AI - Assisted Text Composition for Automated Content Authoring Using Transformer-Based Language Models," *2024 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*. *2024 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 1–6. Available at: https://doi.org/10.1109/IC_ASET61847.2024.10596255.
- Arana-Catania, M. et al. (2021) "Citizen Participation and Machine Learning for a Better Democracy," *Digit. Gov.: Res. Pract.*, 2(3), p. 27:1-27:22. Available at: <https://doi.org/10.1145/3452118>.
- Aven, T. (2016) "Risk assessment and risk management: Review of recent advances on their foundation," *European Journal of Operational Research*, 253(1), pp. 1–13. Available at: <https://doi.org/10.1016/j.ejor.2015.12.023>.
- Bhaskar, T. (2023) *Why does BERT stand out in the sea of sentence embedding models?* Available at: <https://www.linkedin.com/pulse/why-does-bert-stand-out-sea-sentence-embedding-models-bhaskar-t-bi6wc/> (Accessed: January 4, 2026).
- Borchers, M., Cao, T.-B., et al. (2024) "Designing AI-Based Systems to Support the Analysis of Citizens' Inputs from E-Participation," *ECIS 2024 Proceedings* [Preprint]. Available at: https://aisel.aisnet.org/ecis2024/track23_designresearch/track23_designresearch/7.
- Borchers, M., Soroko, D., et al. (2024) "Hear Me Out: Supporting Citizens to Create Comprehensible Contributions on Urban Participation Platforms," *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2), pp. 1–26. Available at: <https://doi.org/10.1145/3686971>.
- Brown, S. (2018) *Software Architecture for Developers: Volume 1 - Architectural storytelling*. Jersey: Leanpub.
- Budakoglu, G. and Emekci, H. (2025) "Unveiling the Power of Large Language Models: A Comparative Study of Retrieval-Augmented Generation, Fine-Tuning, and Their Synergistic Fusion for Enhanced Performance," *IEEE Access*, 13, pp. 30936–30951. Available at: <https://doi.org/10.1109/ACCESS.2025.3542334>.
- Burstein, G. and Zuckerman, I. (2024) "Uncertainty Reduction in Operational Risk Management Process," *Risks*, 12(5), p. 77. Available at: <https://doi.org/10.3390/risks12050077>.
- Castillo-López, G., de Chalendar, G. and Semmar, N. (2025) "Intent Recognition and Out-of-Scope Detection using LLMs in Multi-party Conversations," in F. Béchet et al. (eds.) *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue. SIGDIAL 2025*, Avignon, France: Association for Computational Linguistics, pp. 504–512. Available at: <https://aclanthology.org/2025.sigdial-1.41/> (Accessed: December 28, 2025).
- Challa, V. and Bright, A. (2025) "Low-Resource Fine-Tuning of LLMs for Domain-Specific Tasks," *Universal Research Reports*, 12(4), pp. 45–56. Available at: <https://doi.org/10.36676/urr.v12.i4.1621>.
- Chawla, A. (2023) *Full-model Fine-tuning vs. LoRA vs. RAG*. Available at: <https://blog.dailydoseofds.com/p/full-model-fine-tuning-vs-lora-vs> (Accessed: January 4, 2026).
- Clegg, D. and Barker, R. (1994) *Fast-track: a RAD approach*. Wokingham, England ; Reading, Mass. : Berkshire, UK ; Redwood Shores, CA, USA: Addison-Wesley Pub. Co. ; Oracle (CASE method).
- Collini, E. et al. (2025) "Context-Aware Retrieval Augmented Generation Using Similarity Validation to Handle Context Inconsistencies in Large Language Models," *IEEE Access*, 13, pp. 170065–170080. Available at: <https://doi.org/10.1109/ACCESS.2025.3614553>.

Compete 2030 (2026) *Página Inicial, Compete 2030*. Available at: <https://compete2030.gov.pt/> (Accessed: June 18, 2026).

CoParticipation (2026) *CoParticipation - Plataforma de Próxima Geração para suporte de Processos Participativos (COMPETE2030-FEDER-02208000)*. Available at: <https://coparticipation.onesource.pt/pt> (Accessed: June 18, 2026).

Ding, N. et al. (2023) "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, 5(3), pp. 220–235. Available at: <https://doi.org/10.1038/s42256-023-00626-4>.

EMPATIA (2026) *Página Inicial*. Available at: <https://empatia-project.eu/> (Accessed: June 18, 2026).

EMPAVILLE (2026) *Página Inicial*. Available at: <https://empaville.org/pt> (Accessed: June 18, 2026).

EU (2016) *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*. Available at: <http://data.europa.eu/eli/reg/2016/679/2016-05-04> (Accessed: December 13, 2025).

EU (2024) *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)*. Available at: <http://data.europa.eu/eli/reg/2024/1689/oj> (Accessed: December 13, 2025).

Hevner, A.R. et al. (2004) "Design Science in Information Systems Research," *Management Information Systems Quarterly*, 28(1), pp. 75–106. Available at: <https://doi.org/10.2307/25148625>.

Hosseini, M.-S. et al. (2024) "Formulating research questions for evidence-based studies," *Journal of Medicine, Surgery, and Public Health*, 2, p. 100046. Available at: <https://doi.org/10.1016/j.jglmedi.2023.100046>.

IEEE (2023) *IEEE Code of Ethics | IEEE*. Available at: <https://www.ieee.org/about/corporate/governance/p7-8> (Accessed: December 13, 2025).

Instituto Pedro Nunes (2026) *Instituto Pedro Nunes*. Available at: <https://www.ipn.pt/> (Accessed: June 18, 2026).

Mariani, I. et al. (2025) "An Overview on E-Participation," in I. Mariani et al. (eds.) *Design Thinking as a Strategic Approach to E-Participation: From Current Barriers to Opportunities*. Cham: Springer Nature Switzerland, pp. 31–56. Available at: https://doi.org/10.1007/978-3-031-72160-1_3.

Oates, B.J. (2006) "Researching Information Systems and Computing," 26(6), p. 495. Available at: <https://doi.org/10.1016/j.ijinfomgt.2006.07.009>.

OECD (2025) *Tackling civic participation challenges with emerging technologies: Beyond the hype*. OECD Public Governance Policy Papers. Available at: <https://doi.org/10.1787/ec2ca9a2-en>.

Oficina (2026) *Oficina*. Available at: <https://www.oficina.org.pt> (Accessed: June 18, 2026).

OneSource (2026) *OneSource Consultoria Informática, Lda*. Available at: <https://onesource.pt> (Accessed: June 18, 2026).

Page, M.J. et al. (2021) "PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," *BMJ*, 372, p. n160. Available at: <https://doi.org/10.1136/bmj.n160>.

Petersen, K., Vakkalanka, S. and Kuzniarz, L. (2015) "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, 64, pp. 1–18. Available at: <https://doi.org/10.1016/j.infsof.2015.03.007>.

Shortall, R.M. *et al.* (2022) "Reason against the machine? Future directions for mass online deliberation," *Frontiers in Political Science*, 4(946589). Available at: <https://doi.org/10.3389/fpos.2022.946589>.

UN (2020) *E-participation: A Quick Overview of Recent Qualitative Trends*. UN Department of Economic and Social Affairs (DESA) Working Papers 163. Available at: <https://doi.org/10.18356/0f898163-en>.

Wilkinson, M.D. *et al.* (2016) "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, 3(1), p. 160018. Available at: <https://doi.org/10.1038/sdata.2016.18>.

Appendix A

Table 36 - Data Aggregation of Studies for the First Research Question

Study	Objectives	Approach	Context	Results	Limitations
(Challa and Bright, 2025)	Optimize LLMs for domain-specific tasks at a low cost	Parameter-Efficient Fine-Tuning (PEFT) via LoRA	Research labs and enterprises with limited hardware	Significant reduction in memory usage and training time	Dependency on the quality of the instruction data
(Ding <i>et al.</i> , 2023)	Analyze the effectiveness of optimizing a small number of parameters	Delta-tuning and unification of PEFT methods	Adaptation of large-scale models	Large-scale models can be effectively stimulated by few parameters	Complexity in selecting the ideal tuning strategy
(Budakoglu and Emekci, 2025)	Compare contextual accuracy between RAG and Fine-tuning	External data integration (RAG) vs. parameter adjustment	Applications requiring high factual precision	Hybrid models balance semantic accuracy and efficiency	RAG may suffer from context inconsistencies
(Castillo-López, de Chalendar and Semmar, 2025)	Improve reliability in task-oriented dialogues	Combining BERT's efficiency with LLM generalization	Conversational systems and out-of-scope detection	Performance improvement through cross-model information sharing	Requirement for annotated data for the BERT component

Appendix B

Table 37 - Data Aggregation of Studies for the Second Research Question

Study	Objectives	Approach	Context	Results	Limitations
(Borchers, Cao, <i>et al.</i> , 2024)	Facilitate the analysis of citizen inputs	Design Science Research and AI	E-Participation platforms	Reduction of the analysts' cognitive load	Difficulty in capturing subjective nuances
(Alpdemir and Alpdemir, 2024)	Generate texts that preserve the author's style	Controllable Text Generation (CTG)	Article and blog writing	High fluency content production	Requirement for short initial prompts
(Collini <i>et al.</i> , 2025)	Mitigate hallucinations in LLM responses	Contextual similarity validation	Technical information environments	Significant increase in factual precision	High computational cost of validation
(Arana-Catania <i>et al.</i> , 2021)	Optimize democratic processes with ML	Clustering and duplicate detection	Large-scale public consultations	Improved organisation and visibility of proposals	Risk of bias in machine learning models

Appendix C

Risk ID	Description	Cause	Effect	Risk Owner	Probability (1-5)	Impact (1-5)	PI Score	Expected Result, No Action	Risk Response Type	Response description
	Description of the risk	Cause of the risk	Effect on the project	Name of person who monitors the risk	Group sourced rough estimate of how likely this is to occur	Rough estimate of how significant the impact of this risk	Probability multiplied by Impact	What will happen if the risk becomes an issue and no action is taken	Decision made by group on how to respond to this risk (see above in blue)	How do you know it is time to put the response into play
1	Quality or unavailability of open data sources	Incomplete, outdated, or unavailable participatory data	Limitations in semantic analysis and PoC validation	Miguel Domingues	3	4	12	Planned features become unfeasible or unrepresentative	Mitigate	Evaluate multiple data sources and utilize alternative or synthetic datasets
2	Unavailability or changes in external APIs	Reliance on uncontrolled third-party services	Interruption of API-dependent functionalities	Miguel Domingues	2	3	6	Partial loss of proof-of-concept features	Mitigate	Implement fallback mechanisms and API abstraction layers
3	Higher than expected technical complexity	Integration of LLMs, NLP, and vector databases	Development delays or reduction of functional scope	Miguel Domingues	3	5	15	Schedule non-compliance or excessive solution simplification	Mitigate	Prioritize essential features and adopt incremental development
4	GDPR non-compliance	Improper use of personal data or lack of anonymization	Ethical and legal invalidation of the developed solution	Miguel Domingues	1	5	5	Inability to use, disclose, or publish the solution	Avoid	Use only anonymized data and review legal requirements from the project start
5	Low stakeholder acceptance of the solution	Lack of alignment with real stakeholder	Reduced practical impact of the proof of concept	Miguel Domingues	3	3	9	Solution results in limited scientific or practical value	Mitigate	Define usability criteria and align features with the specific domain

Figure 40 - Risk Management Matrix

Appendix D

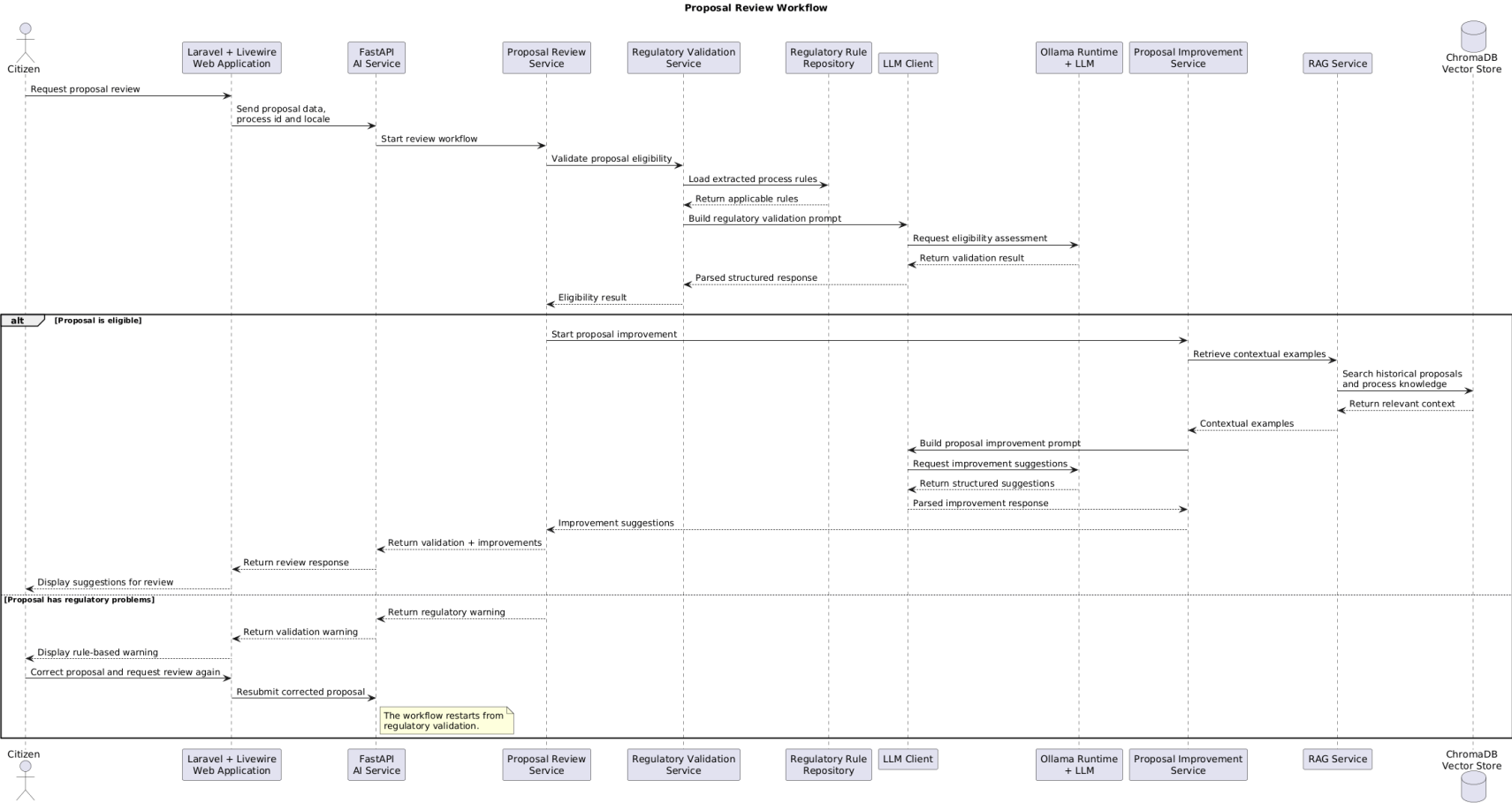


Figure 41 - Proposal Review Sequence Diagram

Appendix E

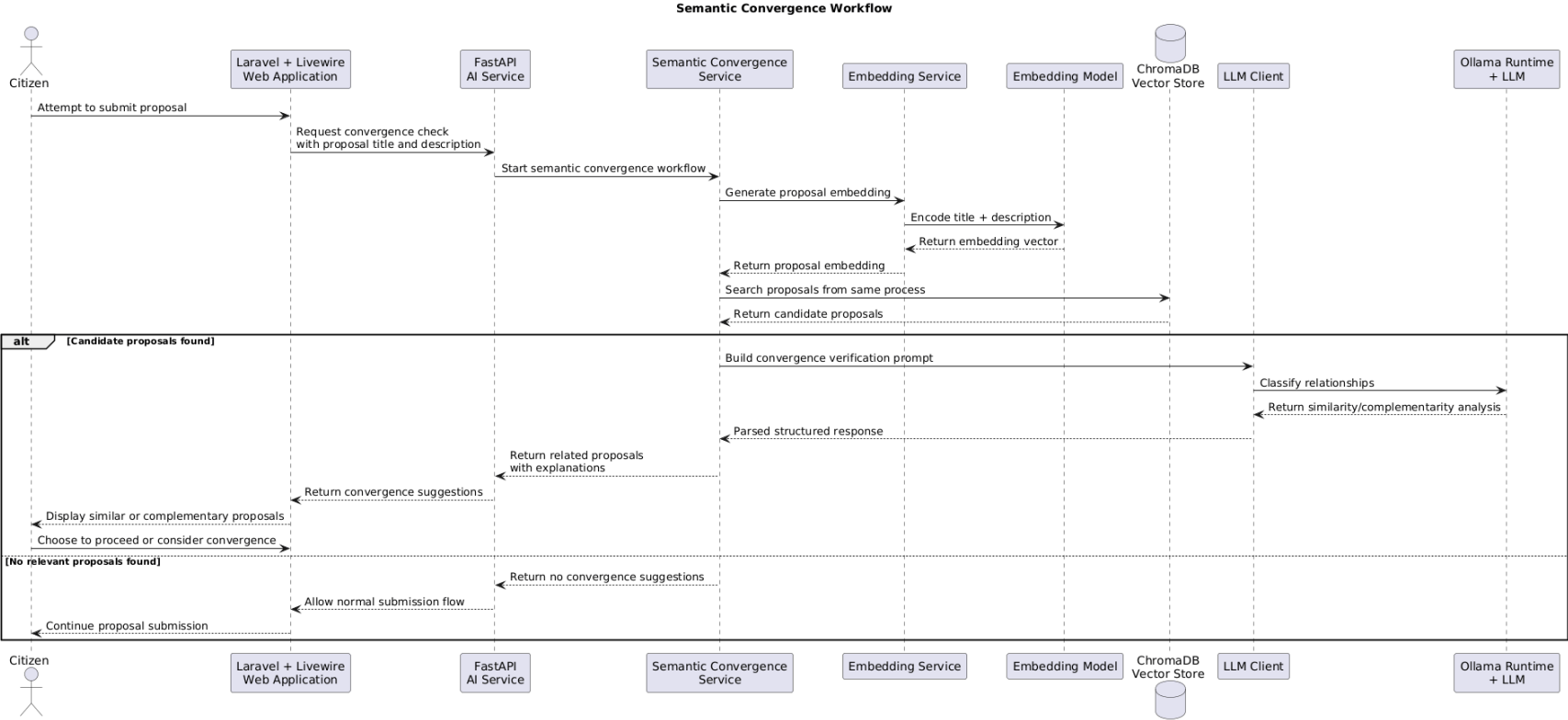


Figure 42 - Semantic Convergence Sequence Diagram

Appendix F

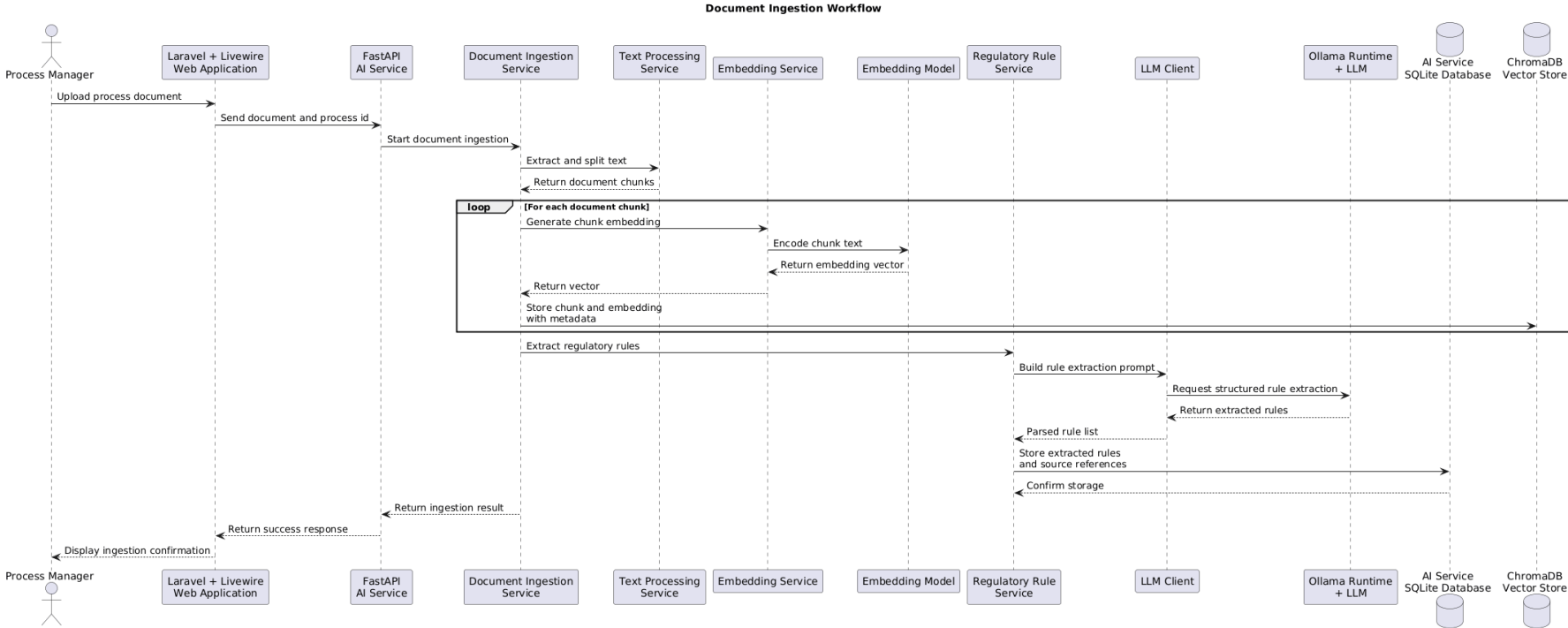


Figure 43 - Document Ingestion Sequence Diagram

Appendix G

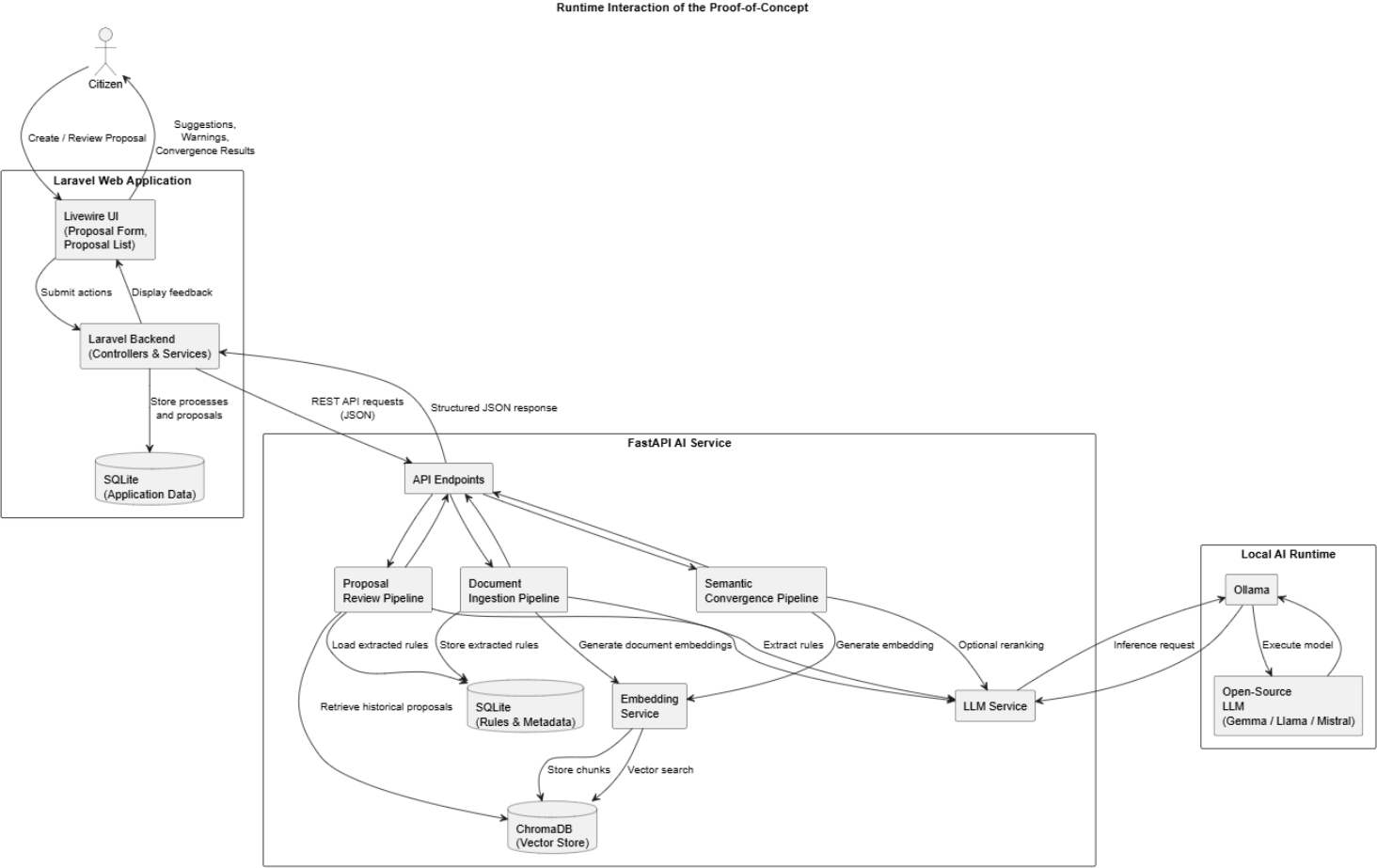


Figure 44 - Runtime Interaction of the Proof-of-Concept

Appendix H

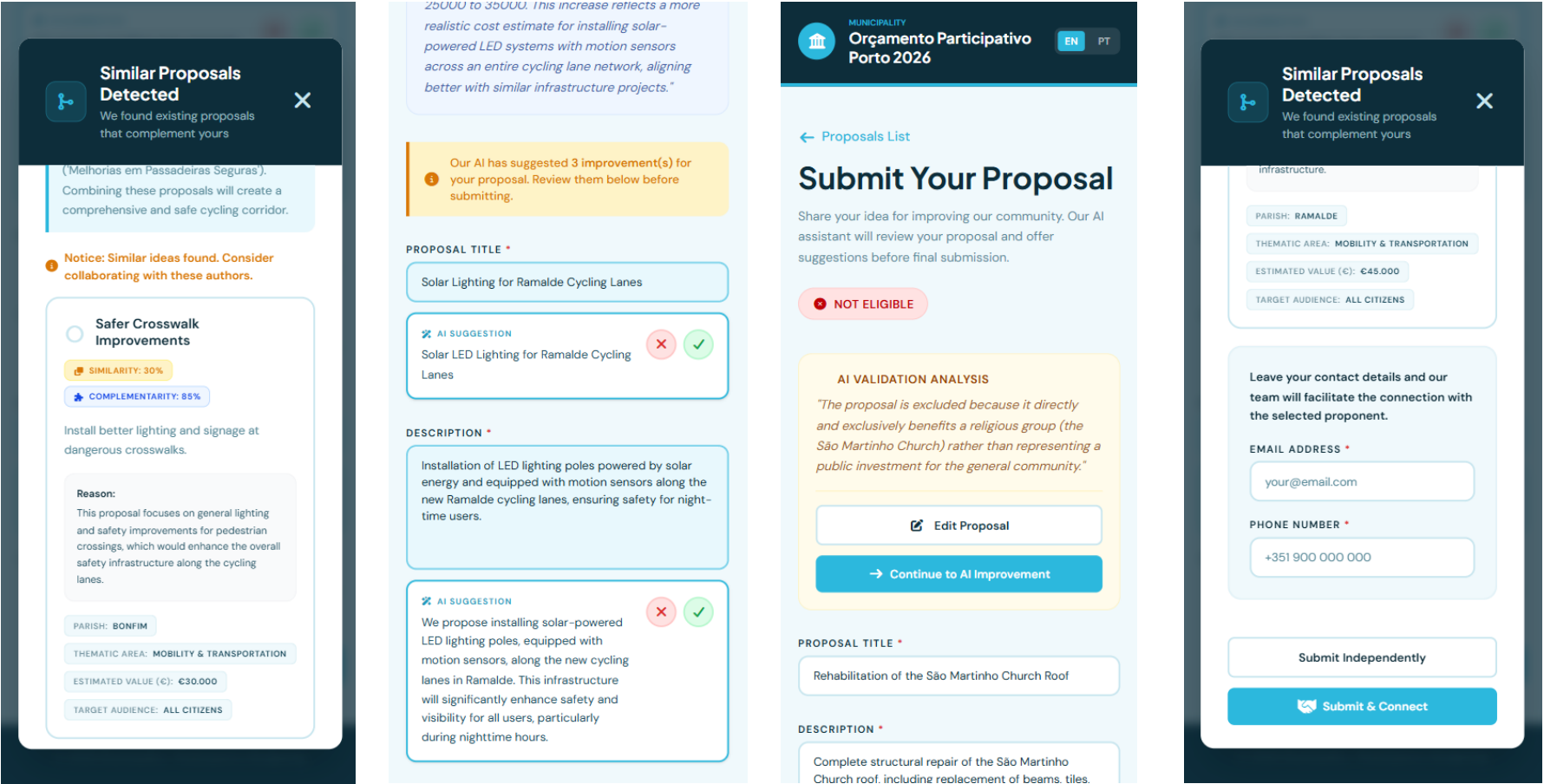


Figure 45 - Mobile-friendly layouts