



D8.4: Exploitation and Sustainability Roadmap

Revision: v.1.1

Work package	WP 8
Task	Task 8.3
Due date	2026-03-31
Submission date	2026-04-22
Deliverable lead	CMS
Version	1.1
Authors	Carlos Coutinho (CMS), Gonçalo Lacerda (EDP), Luís Pisco (EDP), Maria Chiara Campodonico (MTL), Sotirios Spantideas (NKUA), Miroslav Popovic (UNS), Lidija Fodor (UNS), Ivan Kaštelan (UNS), Alceste Scalas (DTU), Nobuko Yoshida (OXF), Ping Hou (OXF), Filippo Vannella (TID), Carla Ferreira (NOVA), João Leitão (NOVA), Giovanni Granato (GMV), David Vázquez (GMV)
Reviewers	Carla Ferreira (NOVA), Panos Trakadas (NKUA), Maria Chiara Campodonico (MTL)
Abstract	This document presents the TaRDIS Exploitation and Sustainability Roadmap, which defines the strategies to maximise the uptake and use of the project’s results beyond the project closure. It consolidates the TaRDIS Key Exploitable Results (KERs). The roadmap aligns exploitation actions by focusing on adoption readiness, modular reuse, demonstrators, and quantified evidence of benefits and trade-offs. It specifies exploitation routes (open ecosystem uptake, partner product integration, pilots, services, and research/education transfer), governance and IP/ownership considerations, KPI-based monitoring, and a sustainability plan covering repository stewardship, maintenance responsibilities, release management, and pathways to following funding and development.
Keywords	Exploitation, sustainability



Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	2026-03-20	1st version of the template for comments	Carlos Coutinho (CMS)
V1.0	2026-04-20	1st version of the draft document	Carlos Coutinho (CMS)
V1.1	2026-05-08	Integration of reviewers comments	Carla Ferreira (NOVA)

DISCLAIMER



Funded by
the European Union

Funded by the European Union (TARDIS, 101093006). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

COPYRIGHT NOTICE

© 2023 - 2026 TaRDIS Consortium

Project funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	to specify R, DEM, DEC, DATA, DMP, ETHICS, SECURITY, OTHER*	
Dissemination Level		
PU	<i>Public, fully open, e.g., web (Deliverables flagged as public will be automatically published in CORDIS project's page)</i>	✓
SEN	<i>Sensitive, limited under the conditions of the Grant Agreement</i>	
Classified R-UE/ EU-R	<i>EU RESTRICTED under the Commission Decision No2015/ 444</i>	
Classified C-UE/ EU-C	<i>EU CONFIDENTIAL under the Commission Decision No2015/ 444</i>	
Classified S-UE/ EU-S	<i>EU SECRET under the Commission Decision No2015/ 444</i>	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.



EXECUTIVE SUMMARY

This deliverable, **D8.4 - Exploitation and Sustainability Roadmap**, reports the outcomes of Task **T8.3** and describes how the TaRDIS consortium will maximise the uptake, reuse, and long-term maintenance of project results beyond the project lifetime. It updates and consolidates the exploitation and sustainability strategy previously defined in **D8.2** [13].

TaRDIS reduces the engineering complexity of building correct, resilient, and efficient heterogeneous swarms and decentralised edge systems by combining:

- A language-independent, event-driven programming model and APIs (reported in **D3.5** [4]).
- A toolbox and runtime platform spanning heterogeneous devices (reported in **D6.3** [8]).
- Distributed AI and AI-based orchestration mechanisms (reported in **D5.3** [7]).
- Validation evidence, guidelines, and quantified evaluation procedures derived from the use cases (reported in **D7.3** [10] and **D7.4** [11]).

This roadmap defines **four project-level Key Exploitable Results (KERs)**, recognised by the European Commission's **Innovation Radar** and positioned in the “Exploring” market maturity category (see section 4.5 for details):

- KER-1: Babel Framework and ecosystem.
- KER-2: Decentralised Data Management Framework.
- KER-3: Improved Actyx Middleware.
- KER-4: Privacy-Preserving and Efficient Decentralised Training.

These KERs are supported by the enabling assets developed in the project, namely in WP3-WP7, including membership and broadcast protocols, anti-entropy mechanisms, **Babel-Swarm** and **Babel-Android**, decentralised storage systems such as **PotionDB** and **Nimbus**, and monitoring/reconfiguration tooling. Their feasibility and applicability have been validated through demonstrators and use-case integrations (e.g., messaging and voting applications running on heterogeneous devices; domain-specific demonstrators in Telefónica, GMV, and EDP; and technology transfer into Actyx's proprietary framework) and showcased in public events and conferences.

Given that several results are at an early market maturity stage, this roadmap adopts a **TRL-aware exploitation approach** focused on adoption readiness and risk reduction: packaging, modular reuse, demonstrators, integration guidance, and measurable evidence of benefits and trade-offs based on the project evaluation methodology (**D2.3** [3]) and validation outcomes (**D7.3** [10] and **D7.4** [11]). It leverages a project **quantification framework**, defined in deliverable **D7.1** [9] to support credible, measurable claims and to transparently manage key adoption constraints, particularly around privacy-by-design and trust assumptions in decentralised learning and operational robustness in dynamic environments.

This roadmap also defines exploitation routes tailored to each KER and stakeholder group: It aims to open the ecosystem uptake through reusable modular components and reference implementations; it promotes partner-led product integration and technology transfer (including proprietary integration paths where applicable); it includes pilot-oriented adoption supported

by packaging, documentation, and operational guidance; and research/education transfer through publications and open artefacts. Sustainability is addressed through planned governance for code and documentation stewardship, maintenance responsibilities, code hosting and release and versioning practices, and continuity measures including follow-on funding pathways and entrepreneurship support.

Finally, the roadmap is reinforced by the TaRDIS participation in the Horizon Results Booster programme, which strengthens the KER prioritisation and value propositions, exploitation strategy and business planning, access to additional funding, and reporting quality. Together, these measures provide a coherent pathway from validated project outcomes to sustained post-project impact and adoption.

The objective of this document is to ensure that the TaRDIS stack is adopted beyond the project through:

- Commercial uptake by industrial partners (direct productisation and service offerings)
- Open-source/community uptake (developer adoption, academic follow-up)
- Standardisation and interoperability influence
- Sustained maintenance of key components and reference implementations
- Measurable impact on KPIs relevant to trustworthy/resilient decentralised intelligence at the edge.

TABLE OF CONTENTS

Executive Summary	3
1 Introduction	10
1.1 Context and purpose of the document	10
1.2 The TaRDIS project	10
1.3 Exploitation and Sustainability.....	11
1.4 Document Layout.....	11
2 Exploitation objectives.....	12
3 Exploitation Concept and Methodology	13
TRL and maturity assessment approach	13
4 Exploitation Results.....	14
4.1 KER-1: Babel Framework and Ecosystems.....	16
4.2 KER-2: Decentralised Data Management Framework	16
4.3 KER-3: Improved Actyx Middleware	17
4.4 KER-4: Privacy Preserving and Efficient Decentralised Training	17
4.5 Listing in the EC Innovation Radar	18
4.6 KER taxonomy used in this roadmap	18
5 Support From Horizon Results Booster	19
5.1 KER-1: Babel Framework	19
NOVA Edge Spin-off Business Plan	25
5.2 KER-3: Improved Actyx Middleware	28
5.3 KER-4: Privacy-Preserving Decentralised Training	31
6 Updated Individual Exploitation Plans.....	34
6.1 Partner: NOVA School of Science & Technology (NOVA).....	34
6.2 Partner: Technical University Of Denmark (DTU).....	42
6.3 Partner: University of Oxford (UOX).....	45
6.4 Partner: University of Novi Sad (UNS)	51
6.5 Partner: National and Kapodistrian University of Athens (NKUA)	57
6.6 Partner: EDP NEW (EDP).....	65
6.7 Partner: GMV	75
6.8 Partner: Caixa Mágica Software (CMS)	82
6.9 Partner: Telefónica Innovación Digital (TID).....	86
6.10 Partner: Martel Innovate (MTL)	90
7 Exploitation Activities.....	92
7.1 Community Building and Ecosystem Engagement	92
7.2 Dissemination Integrated with Exploitation	92
7.3 Sustainability and maintenance post-project	93

Sustainability operating model.....	93
Continuity mechanisms	93
7.4 KPIs and Measurable Evidence	94
Common indicators (project-level).....	94
Evidence base.....	94
7.5 Exploitation Through Improving KERs in Future EC Projects	94
Identification of Potential Future EC Calls for TaRDIS follow-up.....	94
Strategy for Applying to Future EC Calls for TaRDIS follow-up.....	101
Roadmap for Applying to Future EC Calls for TaRDIS follow-up	102
8 Exploitation Pathways and Target Users	103
8.1 Developer and Integrator Uptake	103
8.2 Partner-driven commercialisation and services	103
8.3 Research and education pathway	103
9 Intellectual Property and Sustainability	104
10 Exploitation Deviations, Barriers and Mitigation	106
10.1 Specific Exploitation Risks per Innovation Radar KER	106
10.2 TRL-aware Exploitation Roadmap.....	106
How Having a Low TRL Changes the Exploitation Objective	107
What a Low TRL specifically impacts for TaRDIS.....	107
11 Conclusion	111

LIST OF FIGURES

Figure 1: Banner on the EU Swarm Workshop 202592

LIST OF TABLES

Table 1: TaRDIS Key Exploitable Results15
Table 2: Future EC calls for follow-up of TaRDIS research.....94

ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CO₂	Carbon Dioxide
CO₂e	Carbon Dioxide Equivalent (standardised measure of greenhouse gas impact expressed as CO ₂)
CRDT	Conflict-Free Replicated Data Type
DCR	Dynamic Condition Response (workflow/choreography model)
DEM	Demonstrator (deliverable type)
DEP	Digital Europe Programme
DMP	Data Management Plan (deliverable type)
DRL	Deep Reinforcement Learning
DSO	Distribution System Operator
DTU	Technical University of Denmark
EC	European Commission
ECS	Engineering of Computer Based Systems (conference series)
ETSI	European Telecommunications Standards Institute
EU	European Union
EV	Electric Vehicle
FAUNO	Federated AI Network Orchestrator
FL	Federated Learning
FLaaS	Federated Learning as a Service
FSM	Finite State Machine
GDPR	General Data Protection Regulation
GNSS	Global Navigation Satellite System
HE	Horizon Europe
HFL	Hierarchical Federated Learning
HRB	Horizon Results Booster
HVAC	Heating, Ventilation and Air Conditioning
IA	Innovation Action (Horizon Europe action type)
IDE	Integrated Development Environment
IFC	Information Flow Control
IoT	Internet of Things
IP	Intellectual Property / Internet Protocol (context-dependent)
IPR	Intellectual Property Rights
ISL	Inter-Satellite Link
JDK	Java Development Kit
KPI	Key Performance Indicator
KER	Key Exploitable Result
LEO	Low Earth Orbit
LOI	Letter of Intent

LSTM	Long Short-Term Memory (neural network type)
MARL	Multi-Agent Reinforcement Learning
MES	Manufacturing Execution System
ML	Machine Learning
MOM	Manufacturing Operations Management
MPST	Multiparty Session Types
MPT-FLA	MicroPython Testbed for Federated Learning Algorithms
MQTT	Message Queuing Telemetry Transport
MRE	Minimal Reproducible Example
MTL	Martel Innovate (partner organisation)
NKUA	National and Kapodistrian University of Athens
NOVA	NOVA School of Science and Technology (NOVA University Lisbon)
NuCRDT	Non-uniform CRDT
ODTS	Orbit Determination and Time Synchronisation
OEM	Original Equipment Manufacturer
OSS	Open-Source Software
PTB-FLA	Python Testbed for Federated Learning Algorithms
P2P	Peer-to-Peer
PAT	Process Analysis Toolkit (model checking tool)
PNT	Positioning, Navigation and Timing
PoC	Proof of Concept
PV	Photovoltaic (solar generation)
PUS	Packet Utilisation Standard (space systems)
PU	Public (dissemination level)
RIA	Research and Innovation Action (Horizon Europe action type)
RL	Reinforcement Learning
SBOM	Software Bill of Materials
SDLC	Software Development Life Cycle
SME	Small and Medium-sized Enterprise
SW	Software
TaRDIS	Trustworthy And Resilient Decentralised Intelligence for Edge Systems
TCC	Transactional Causal Consistency
TDMA	Time Division Multiple Access
TID	Telefónica Innovación Digital (partner organisation)
TRL	Technology Readiness Level
UOX	University of Oxford
UNS	University of Novi Sad
VS Code	Visual Studio Code
WP	Work Package

1 INTRODUCTION

1.1 CONTEXT AND PURPOSE OF THE DOCUMENT

This document provides the Exploitation and Sustainability Roadmap for **TaRDIS** (Trustworthy and Resilient Decentralised Intelligence for Edge Systems), which addresses the engineering complexity of building, deploying, and evolving heterogeneous swarms and decentralised edge systems. The roadmap defines how TaRDIS results will be exploited during and after the project, and how key software and knowledge assets will be sustained to ensure continued impact beyond the project lifetime.

The roadmap complements the project's technical work by translating research and engineering outputs into a coherent set of Key Exploitable Results (KERs), stakeholder-facing value propositions, exploitation routes, and sustainability commitments. It also provides the basis for consistent reporting and monitoring of exploitation progress through measurable indicators.

1.2 THE TARDIS PROJECT

TaRDIS focuses on significantly easing the effort required to build correct and efficient decentralised applications spanning diverse devices, networks, and programming environments. The project advances the state of the art by combining a novel programming paradigm with a toolbox supporting both development-time and runtime needs of swarm and decentralised systems. TaRDIS aims to:

- develop a language-independent, event-driven programming model offering distribution abstractions and decentralised machine learning primitives
- build a development environment for correct-by-design heterogeneous swarms, supported by embedded semantic analyses
- develop schemes for decentralised intelligence suitable for heterogeneous swarm environments
- develop decentralised runtime algorithms and protocols supporting the programming model in execution
- ensure interoperability across a significant number of devices and programming languages.

These objectives reflect the project's positioning at the intersection of programming languages and tools, distributed systems, edge computing, and decentralised intelligence, with the goal of enabling trustworthy and resilient swarm applications in a wide range of real-world settings.

TaRDIS exploitation follows a standard Horizon Europe logic linking research and engineering progress to adoption and impact:

Research outputs → **reusable toolbox components (software + specifications)** → **validation in use cases and demonstrators** → **adoption assets (documentation, examples, guidelines, benchmarks)** → **uptake by developers/integrators/platform providers** → **long-term sustainability and follow-on development.**

This roadmap reports how TaRDIS operationalises this logic through a prioritised KER portfolio, targeted stakeholder engagement, measurable progress indicators, and sustainability commitments.

1.3 EXPLOITATION AND SUSTAINABILITY

TaRDIS is highly intensive on the concepts of software and tooling. For such projects, having a long-term impact depends not only on scientific advances, but on whether the resulting components can be adopted, integrated, maintained, and evolved by stakeholders outside the project consortium.

This requires many activities for ensuring a strong footprint in the IT stack, such as packaging research outputs as reusable building blocks and reference implementations, demonstrating its value through validation assets, demonstrators, and quantified evidence of improvements and trade-offs, reducing adoption barriers via interoperability, and integration with existing infrastructures. Additionally, it is also important to define clear strategies for intellectual property management (IPR), licensing, and governance, as well as post-project sustainability planning for repositories, documentation, maintenance responsibilities, and follow-up funding.

1.4 DOCUMENT LAYOUT

The main purpose of the Exploitation and Sustainability Roadmap is to report the efforts taken by the project team as outcomes of task T8.3 “Exploitation and sustainability” to accomplish the joint exploitation strategy of the TaRDIS results. It builds on the definitions of its predecessor deliverable D8.2 [13], which specified the process followed to perform the task, the joint exploitation approach, and the guidelines for exploitation comprising exploitable results, potential addressees, stakeholders, IPR, and tools for enabling exploitation actions.

The following sections detail the reporting of activities which will steer the project towards achieving its Exploitation and Sustainability objectives, including:

- identifying and prioritising the TaRDIS KERs, including those recognised by the European Commission’s Innovation Radar¹
- characterising stakeholder segments and adoption scenarios relevant to TaRDIS outcomes
- defining exploitation routes and actions for each KER (open ecosystem uptake, partner integration, pilots, research transfer)
- providing a TRL-aware exploitation strategy and associated risk management
- setting out sustainability measures (governance, maintenance, release management, continuity funding)
- defining KPIs and a monitoring approach to track exploitation progress and post-project continuation

By providing this structure, the team aims to ensure that exploitation planning is tightly aligned with technical progress, validation activities, and the practical conditions required for a real-world uptake.

¹ <https://project-TaRDIS.eu/news/2025/09/18/TaRDIS-technologies-enter-the-european-commissions-innovation-radar>

2 EXPLOITATION OBJECTIVES

TaRDIS exploitation objectives are grounded in the project mission: to reduce the effort required to build correct and efficient heterogeneous swarms and decentralised edge systems through a language-independent event-driven programming model, correctness-by-design development environment, decentralised intelligence primitives, runtime protocols, and interoperability across devices and languages. Further detail can be asserted on the project's web site².

The basics of the project's exploitation objectives are to ensure that developers coming from SMEs and other companies become easily involved with the TaRDIS environment and toolbox. This is achieved through a set of project activities which, despite being part of the research and development of swarm applications, have a focused purpose to engaging developers into the use of the TaRDIS toolbox, namely:

- **Enabling the adoption by developers and system integrators** that aim to develop and build decentralised/edge/swarm applications through stable APIs, documentation, and examples, such as the programming model and APIs definition referenced in the project deliverables D3.5 [4] and D3.6 [5], but also the planned tooling releases
- **De-risking the deployment and operation** of such applications via the use of best-practices, best-of-breed tools (D3.6 [5]), and correctness-by-design tooling (e.g., semantic analyses documented in the project deliverable 4.3 [6]) and validated runtime mechanisms
- **Creating reusable building blocks** that partners, developers and third-party entities can integrate into products/services and open ecosystems, such as the ones documented in the project deliverable D6.3 [8]
- **Driving the uptake through evidence** by providing demonstrators and a quantification framework (D7.1) that allows TaRDIS to express benefits and trade-offs in measurable terms (e.g., performance/overhead, reliability, operational complexity, privacy/trust constraints).
- **Influencing the practice and standards** in decentralised intelligence, swarm computing and edge programmability such as the ones reported in the project deliverable D5.3 [7]
- **Ensuring sustainability post-project** by providing maintained repositories (e.g., CodeLab³), reported in the project deliverable D1.3 [1], promoting governance and fostering a community pathway post-funding (disseminating the project in universities, e.g., FISTA⁴ and in industry-referenced conferences such as ADRForum⁵, or even online⁶), or by assuming an active positioning in the European swarm computing ecosystem through cluster activities such as organising the Swarm Workshop 2025⁷
- **Obtaining recognition of the quality** of the TaRDIS components by the European Commission's Innovation Radar⁸, indicating innovation potential and visibility as a reference.

² <https://project-TaRDIS.eu>

³ <https://codelab.fct.unl.pt/di/research/TaRDIS>

⁴ <https://fista.iscte-iul.pt>

⁵ <https://adrforum.eu>

⁶ <https://www.youtube.com/@TaRDISproject>

⁷ <https://swarmworkshop.eu>

⁸ <https://project-TaRDIS.eu/news/2025/09/18/TaRDIS-technologies-enter-the-european-commissions-innovation-radar>

3 EXPLOITATION CONCEPT AND METHODOLOGY

For this report, the term “exploitation” covers:

- Scientific exploitation: publications, open artefacts, method transfer to research and education.
- Technological exploitation: software/tooling reuse, integration in third-party systems, reference implementations.
- Commercial exploitation: incorporation into products/services and support offerings by partners or third parties.
- Ecosystem exploitation: community building and (where applicable) standardisation influence.

TaRDIS explicitly integrates communication and dissemination with exploitation and sustainability planning (D8.1 [12], and D8.3). The project’s work package structure includes a dissemination/exploitation/standardisation-oriented WP (as indicated by planned deliverables D8.1, D8.3, D8.4, D8.5), which provides the governance vehicle for exploitation monitoring and reporting.

TRL and maturity assessment approach

TaRDIS reports maturity using TRL estimates and adoption readiness indicators. TRL estimates are self-assessed by each KER owner based on evidence from:

- Integration and validation activities in the use cases (D7.3 [10] and D7.4 [11]).
- Platform/toolbox releases and demonstrations (D6.3 [8])
- Documented evaluation methodology (D2.3 [3]).

Where results remain exploratory or lack validation in relevant environments, TRL estimates are capped accordingly. Adoption readiness is further tracked via availability of versioned releases, documentation, minimal reproducible examples, and operational guidance (see sections 7.3 and 7.4).

In this roadmap, “adoption readiness” refers to the practical conditions under which an external user can take up a TaRDIS asset with bounded effort. Concretely, this means: a tagged release with documented dependencies and supported platforms; a public repository with a README and licence; at least one runnable example reproducing a benchmarked result; and a documented deployment or integration path. KERs report against these conditions in their respective sections, allowing the consortium to distinguish results that are technically validated from those that are also ready for third-party uptake.

4 EXPLOITATION RESULTS

Key Exploitable Results (KERs) are the project outcomes with the clearest potential to generate impact beyond research activities. A KER can be a software component, a framework, a methodology, a protocol, a dataset, a demonstrator, a reference architecture, or a set of guidelines, if it is identifiable, transferable, and can be taken up by stakeholders outside the consortium.

TaRDIS identified a broad set of exploitable results in D8.2. In this final roadmap, the consortium organises exploitation around **four project-level Key Exploitable Results (KERs)** aligned with the innovations featured in the European Commission's Innovation Radar. This alignment supports consistent external communication and reporting.

For execution purposes, TaRDIS applies an **internal prioritisation**:

- Project-level KERs that receive primary post-project packaging and go-to-market effort
- Enabling assets that strengthen the Project-level KERs (e.g., adapters, benchmarks, demos, validation guidelines). This approach concentrates resources without discarding important exploitable outputs.

In practice, defining KERs serves three main purposes:

- **Focus and prioritisation:** Research projects often produce many outputs. KER identification selects the subset with the strongest potential for adoption and impact, so that exploitation efforts (packaging, validation, documentation, outreach) are concentrated where they matter most
- **Traceability from results to impact:** KERs provide a clear link between technical work packages and exploitation actions. They allow the consortium to report “what was produced”, “who can use it”, “how it will be used”, and “what evidence shows progress”, using measurable indicators (e.g., releases, integrations, pilots, uptake metrics).
- **Coordination across partners:** In multi-partner projects, KERs help clarify ownership, access rights, licensing expectations, and partner roles in exploitation. This reduces fragmentation and enables coherent joint exploitation narratives.

A robust KER description typically includes: a short definition, target stakeholders, value proposition, maturity/TRL, dependencies, exploitation routes (open-source uptake, partner integration, pilots, services, research transfer), IP/licensing approach, risks, and KPIs.

Deliverable 8.2 [13] identified eight candidate exploitable results. This roadmap consolidates them into four project-level KERs, grouping closely related assets and removing those that have since matured into supporting components rather than standalone exploitable outputs. Table 1 summarises the project-level KERs, aligned with the TaRDIS objectives and technical documentation.

Table 1: TaRDIS Key Exploitable Results

KER	What it is	Primary exploitation value	Status
<p>KER-1: Babel Framework and Ecosystems</p>	<p>Babel ecosystem extended for swarm and mobile environments, including Babel-Swarm (autonomic behaviours) and Babel-Android, plus multiple implementations of membership/communication services integrated into the ecosystem. (WP6 evidence: D6.3 [8])</p>	<p>Composable, adaptable building blocks for heterogeneous swarm apps including mobile</p>	<p>Open ecosystem uptake; partner integration; demos and integration guides; training and community building</p>
<p>KER-2: Decentralised Data Management Framework</p>	<p>Two complementary decentralised data systems: PotionDB (partial geo-replication, TCC, materialised views via incremental maintenance based on NuCRDTs) and Nimbus (fully decentralised swarm storage, strong eventual consistency via CRDTs, explicit partial replication, access control, persistence, reactive notifications, Babel Protocol Commons APIs). (WP6 evidence: D6.3 [8])</p>	<p>Data availability and consistency under cloud-edge and swarm dynamics; reduced replication cost; offline tolerance</p>	<p>Integration into toolbox/runtime; pilots where locality / bandwidth / availability dominate; developer uptake via APIs and reference implementations</p>
<p>KER-3: Improved Actyx Middleware</p>	<p>Technology transfer into Actyx's proprietary framework: improvements explored for reliable and durable event broadcast and durable event storage; Actyx imported project ideas; further WP6 involvement limited due to proprietary core. (WP6 evidence: D6.3 [8])</p>	<p>Direct industrial uptake pathway (integration into a market-facing proprietary middleware)</p>	<p>Product integration in industry; market deployment via CoNSERT and CMS; potential joint showcases; report evidence via partner confirmations, release notes, and demonstrator references</p>
<p>KER-4: Privacy-Protecting and Efficient Decentralised Training</p>	<p>Decentralised training and orchestration direction supported by toolbox/runtime integration work (WP6) and quantified constraints/needs identified in D7.1 [9] (privacy-by-design gaps, differential privacy noise injection, residual trust in aggregators)</p>	<p>Enables decentralised intelligence under edge constraints; roadmap addresses privacy and trust barriers</p>	<p>R&D-to-pilot hardening; pilots and controlled deployments; measurable trade-offs using D7.1 procedures; partner services and follow-on funding</p>

4.1 KER-1: BABEL FRAMEWORK AND ECOSYSTEMS

Description: KER-1 comprises the Babel framework and the surrounding ecosystem extensions that enable composable, reusable runtime services for decentralised and swarm applications across heterogeneous environments, including mobile platforms and autonomic behaviours.

Evidence: D3.5 [4] documents the programming model and an overview of APIs and correctness properties supported by workflow/protocol-based interactions. WP6 reports the evolution of the Babel ecosystem, including the development of **Babel-Swarm** (autonomic behaviours and swarm-oriented enhancements) and **Babel-Android** (mobile support), as well as the integration of multiple implementations of membership and communication protocols to improve composability and adaptability in dynamic swarm environments (D6.1, D6.2, D6.3 [8]).

Exploitation status: Exploitation focuses on ecosystem uptake and developer adoption through modular reuse, stable interfaces, examples/demos, and integration guidance. Given the early maturity, emphasis is placed on repeatable demonstrators, documentation, and integration assets that reduce adoption friction.

4.2 KER-2: DECENTRALISED DATA MANAGEMENT FRAMEWORK

Description: TaRDIS implements its decentralised data management framework through two complementary systems suited to different scenarios:

- **PotionDB** is a partially geo-replicated key-value database designed for geo-distributed settings where each replica stores only part of the data. In D6.3 [8], it is presented as a solution for supporting recurrent queries efficiently through materialised views, while preserving Transactional Causal Consistency and reducing unnecessary update propagation.
- **Nimbus** is a fully decentralised key-value database for volatile swarm environments, requiring no dedicated cloud or edge infrastructure. In D6.3 [8], it is presented as supporting offline operation, automatic reconciliation after reconnection, partial replication, access control, persistence, and reactive client notifications.

Evidence: This KER is extensively described in TaRDIS deliverables D6.2 and D6.3 [8], including their data models, APIs, consistency models, replication strategies, and status, and has been demonstrated in videos on the project's Youtube channel⁹ and on the EU Swarm Workshop in Brussels¹⁰.

Exploitation status: Exploitation is structured around:

- Integration into TaRDIS toolbox/runtime and validation in use cases.
- PotionDB supports exploitation in domains where adopters need geo-distribution, controlled replication cost, and efficient repeated analytics-style queries without constantly incurring remote-access overhead.
- Nimbus supports exploitation in swarms (robots / drones / satellites-in-emulation / harsh-environment telemetry) where the key adoption barrier is the lack of reliable infrastructure and the need for scalable decentralised storage that tolerates churn and partitions.

⁹ <https://www.youtube.com/@TaRDISproject>

¹⁰ <https://swarmworkshop.eu>

- Partner-led pilots where data locality, bandwidth, and availability constraints dominate.
- Developer uptake via clear APIs (Babel Protocol Commons; transactional KV API; view definitions) and reference implementations.

4.3 KER-3: IMPROVED ACTYX MIDDLEWARE

Description: KER-3 captures the TaRDIS contributions that were transferred into the Actyx middleware product line, focusing on improvements for reliable and durable event broadcast and durable event storage in decentralised settings.

Evidence: WP6 reports investigation and prototyping activities connected to improving reliable/durable event dissemination and storage, and indicates that part of the work has been imported into Actyx's product, while further WP6 involvement is constrained by the proprietary nature of the core middleware (D6.3 [8]). Use-case integration and validation activities covering the Actyx use case are reported in the WP7 deliverables (D7.3 [10] and D7.4 [11]).

Exploitation status: This KER represents a strong exploitation pathway at low TRL: direct transfer into a market-facing proprietary product. The primary exploitation path is **direct industrial technology transfer and product integration**, supported by partner-internal roadmaps, release notes, and continued validation in industrial contexts. Where software components are not publicly releasable, exploitation evidence is tracked via integration milestones, demonstrator evidence, and business uptake indicators (see section 7.4).

4.4 KER-4: PRIVACY PRESERVING AND EFFICIENT DECENTRALISED TRAINING

Description: This KER comprises TaRDIS results enabling privacy-preserving and efficient decentralised training and orchestration for edge systems, including federated learning service approaches and supporting lightweight AI techniques, validated through use-case integration and project evaluation methodology.

Evidence: Deliverable D5.3 [7] reference federated learning challenges (privacy-by-design, differential privacy noise injection) and distributed trust considerations (role of aggregator). D7.1 [9] identifies exploitation-critical challenges for federated/decentralised learning, including privacy-by-design limitations, the need for differentially private noise injection, and residual trust dependencies on aggregation/control points. D6.3 [8] reports integration work and use-case progress relevant to decentralised learning (e.g., Telefónica FLaaS evolution, hierarchical aggregation, and split learning exploration).

Exploitation status: primary exploitation route is technical reuse inside the toolbox/platform and adoption by practitioners needing decentralised ML on the edge; additional R&D may be needed for hardened privacy/security features. Exploitation is staged:

- Validated prototypes and demonstrators.
- Pilot trials with measurable trade-offs.
- Hardening of privacy/trust mechanisms towards higher TRLs, supported by the D7.1 [9] quantification procedures.

4.5 LISTING IN THE EC INNOVATION RADAR

The European Commission's **Innovation Radar**¹¹ is a mechanism that identifies and profiles innovations emerging from EU-funded projects and assesses their innovation potential and market maturity. When project results are featured in the Innovation Radar, it supports exploitation in two complementary ways:

- **External visibility and credibility:** The Innovation Radar recognition provides an externally framed description of key innovations and their maturity, improving stakeholder trust and making it easier to communicate results to industry, investors, and public-sector adopters.
- **Sharper exploitation reporting:** Becoming Innovation Radar entries encourage projects to articulate results in a market- and adoption-oriented manner, connecting technical outputs to clear value propositions, stakeholder segments, and realistic routes to uptake.

4.6 KER TAXONOMY USED IN THIS ROADMAP

TaRDIS distinguishes:

- **Project-level KERs:** The four consortium-wide KERs (see section 4.1 to 4.4) aligned with the EC Innovation Radar (see section 4.5).
- **Partner-level exploitable results:** Exploitable components owned by individual partners and potentially supporting one or more project-level KERs (detailed in section 6).
- **Enabling assets:** Integration adapters, demonstrators, benchmarks, guidelines, and operational tooling that increase adoption readiness of project-level KERs (primarily evidenced in **D6.3** [8], **D7.3** [10], and **D7.4** [11]).

Although the **KER-2: Decentralised Data Management Framework** was recognised as an Innovation Radar result, TaRDIS will deprioritise post-project investment in this KER, scaling exploitation effort according to partner capacity and expected near-term adoption pathways, following a Horizon Results Booster (see section 5) recommendation to concentrate post-project effort on the other three KERs. In practice, the consortium will still pursue:

- Continued open artefact availability and documentation for PotionDB and Nimbus.
- Targeted follow-up funding and pilots to further harden performance, operational tooling, and reproducible benchmarks before wider industrial uptake.

¹¹ <https://innovation-radar.ec.europa.eu>

5 SUPPORT FROM HORIZON RESULTS BOOSTER

The Horizon Results Booster (HRB) is a free service offered by the European Commission to Horizon Europe projects to strengthen the exploitation of their research results. Through expert coaching and facilitated workshops, the HRB helps project teams move their key exploitable results (KERs) closer to market, producing three core outputs: a characterisation table, an exploitation roadmap, and a business plan for any planned spin-off.

TaRDIS used Horizon Results Booster (HRB) support to strengthen exploitation readiness of low-TRL software outcomes. HRB activities supported:

- Consolidation and prioritisation of the project KER portfolio.
- Stakeholder-facing framing of value propositions and adoption scenarios.
- Clarification of exploitation routes (open ecosystem, partner integration, pilots, services, research transfer).
- Reinforcement of sustainability planning (governance, responsibilities, and continuity mechanisms).

These actions address common exploitation barriers in multi-component toolboxes, including fragmentation, unclear ownership boundaries, and insufficient adoption assets.

5.1 KER-1: BABEL FRAMEWORK

TaRDIS participated in the HRB programme focusing on the Babel Ecosystem KER, coordinated by NOVA FCT as IP owner and technical leader. The full outputs are provided in deliverable D6.3 [8] (Babel Ecosystem Exploitation Package). The sections below summarise each output.

KER	Individual intellectual asset	Protection measure	Reasoning
KER 1 - Babel Framework and Ecosystem	Babel core framework	Copyright (automatic) + Open source	Framework value grows with adoption; copyright protects authorship, licence controls reuse
	Babel - Swarm extensions (self-configuration, security primitives)	Copyright + separate licence	Allows selective openness; advanced features can remain controlled
	Babel-Android variant	Copyright + licence aligned with core	Keeps ecosystem coherent while allowing platform-specific growth
	Documentation, tutorials, reference examples	Copyright with possible open access	Encourages learning and adoption; low commercial risk, can be reused in paid training and certification
	Project name and branding	Trademark	Prevents confusion and misuse; critical for ecosystem trust

Description	The Babel Ecosystem is a unified programming environment for building smart, privacy-preserving IoT and home automation solutions that configure themselves, can operate without a connection to the Internet, and support interoperability across a wide range of devices. Developers taking advantage of Babel Ecosystem and integrators get faster development at lower costs, while end-users get stronger privacy, and solutions that are reliable even when the internet doesn't. Under the hood, Babel provides a wide range of modular components for building reliable and efficient decentralised systems - such as membership abstractions (that dynamically track devices that are part of the system) and
--------------------	---

	reliable communication mechanisms (send once, everyone gets it) - alongside other decentralised protocols that provide powerful abstractions easily to support diverse applications in the IoT and home automation domains.
Target market/end users	<p>The primary market for the Babel Ecosystem includes European SMEs and startups developing IoT and home automation solutions that require, or can benefit from, devising solutions that provide reliability, privacy, and offline functionality - such as smart home controllers, energy efficiency systems, or building automation platforms. The secondary market consists of device manufacturers and system integrators seeking a lightweight, interoperable software-based solution that allows to have a wide range of heterogeneous devices without relying on centralised cloud services or platforms to ensure inter-operation.</p> <p>The Babel Ecosystem addresses a growing need among technology providers and integrators to deliver interoperable, privacy-preserving, cloud-independent, and resilient solutions that ensure that the control and ownership of the system (as well as its data) remain under its exclusive control. According to Statista, the IoT market in Europe recorded a revenue of over 225 billion U.S. dollars in 2023 and is expected to increase to 411 billion U.S. dollars by 2028 [15]. Companies in almost all industries are adopting and investing in IoT solutions to increase efficiency, productivity, and security across their business activities.</p>
End-users needs / problems	<p>Key pain points:</p> <ul style="list-style-type: none"> • Dependence on cloud services increases latency, operational costs, and privacy risks, which is a major pain for final users. • Fragmented IoT and home automation development tools hinder integration and interoperability, which limits the ability and effectiveness of integrators and applications/systems developers. • Ensuring resilience and fault-tolerance in heterogeneous, resource-constrained systems is complex, which requires developers and integrators to seek specialised support to handle these challenges, which is expensive, and time consuming. • Innovation cycles are slow due to high integration and prototyping costs, which affects integrators and developers on one hand, and limits the ability of end-users to have access to the innovative solutions that they demand.
Competitive advantages	<p>Key advantages:</p> <ul style="list-style-type: none"> • Lower infrastructure costs by significantly reducing cloud dependence and avoiding vendor lock-in problem, which is a major advantage for developers and integrators that can avoid costs and dependency on cloud infrastructures, and end-users as they benefit from additional privacy and simplicity. • Privacy and sovereignty preserved by running logic closer to final users and their infrastructures, which is a benefit for end-users that retains the ownership of their deployments as well as all data produced/manipulated by it. • Competitive advantage through adoption of decentralised, future-proof architectures, which is a significant benefit for integrators and developers. • Faster deployment of decentralised services with reusable frameworks and self-configuration features, which makes the development and integration cycle faster and more effective. • Confidence in system resilience and fault tolerance, which is a benefit for end-users. • Unified APIs and development abstractions across devices (embedded, home automation, mobile) reduce engineering effort and enable faster innovation, which benefits integrators and developers alike.
Use model	<p>The Babel Ecosystem will be exploited following a hybrid model that combines open-source and community-driven, and commercial components.</p> <p>The core libraries and runtime frameworks will be released under a permissive open-source license (e.g., Apache 2.0) to foster transparency, trust, and generate community adoption, allowing the wide community of experiments and thinkers to contribute with their own tools which can further drive innovation and allow us to identify needs and emerging trends.</p> <p>A spin-off company of NOVA FCT (NOVA Edge) will be established to develop specialised components and services that extend the open-source core. These components will be licensed to developers, integrators, and device manufacturers, supporting commercial exploitation in the IoT and home automation domains. Licensing models can be tailored to</p>

	<p>specific device types or product lines, which allows to avoid the issuing of individual licenses per device that would be impossible to validate since the solution can operate without Internet connections.</p> <p>This model encourages broad adoption by lowering entry barriers for innovators and SMEs while allowing to explore additional revenue sources from consulting, integration, and training services. Over time, the open-source community will drive growth and experimentation, while licensed components will provide reliability and long-term support for production deployments.</p>
<p>Early Adopters</p>	<p>Early adopters of the Babel Ecosystem are expected to include European SMEs and startups developing IoT and home automation solutions that require privacy, interoperability, and offline functionality. These organizations are typically innovation-driven, open to experimentation, and limited by the high costs and rigidity of cloud-centric platforms.</p> <p>Additional early adopters include system integrators and device manufacturers seeking flexible, lightweight, efficient, and correct software solutions to enhance interoperability and long-term maintainability in their products.</p> <p>The Babel spin-off and the open-source community will serve as the main channels to engage these adopters through early access to beta versions, technical support, and opportunities to co-create new modules and new abstractions to foster innovation and enable the creation of solutions for diverse domains. Such collaborations will help validate the technology in real IoT and home automation scenarios, while establishing reference deployments that can server a key demonstrator of the reliability, simplicity, flexibility, and the additional benefits brought by the Babel ecosystem to the broader market.</p>
<p>Adopters' problems/needs</p>	<p>Key adopter pain points:</p> <ul style="list-style-type: none"> • Over-reliance on cloud: higher latency, high operational costs, privacy concerns, vendor lock-in on solutions that are developed and deployed on client premises, this is a significant limitation for integrators and developers, but also a significant hurdle for end-users that depend on cloud operators and platforms running there, losing sovereignty and control over their deployments. • High integration costs and long development cycles for developers of IoT solutions and the high cost (and time) for innovation and exploring new ideas, which negatively affects developers of solutions. • Difficulty ensuring resilience (i.e., fault-tolerance and robustness) in decentralised or resource-constrained environments composed of different heterogenous devices, which is a significant and costly pain for developers and integrators alike. • Fragmentation of tools across embedded, home automation, and IoT development ecosystems, which limits the capacity of integrators to effectively take advantage of different alternatives in the market when building solutions.
<p>Alternative solution</p>	<p>Current solutions heavy rely on cloud-based services, potentially requiring subscription or registration on different platforms that are used by different manufacturers of IoT devices. This solution creates a full dependence on Internet connectivity, third-party operators, cloud-infrastructures, and creates both difficulties in deployment and resilience but also raises concerns related with the privacy of end-users' data and exposition of behaviours.</p> <p>One can summarize the current alternative solutions in the specific groups:</p> <ul style="list-style-type: none"> • Cloud-centric platforms (e.g., AWS IoT, Azure IoT, Google Home, Alexa): offer scalability and convenience but create dependence on third-party infrastructures, increase operational costs, and present significant privacy limitations. These systems are incompatible with offline functionality and become unavailable without an Internet connection. • Proprietary edge or gateway frameworks (e.g., Cisco solutions, IKEA home solutions): provide partial decentralization but remain tied to vendor-specific hardware or services, reducing interoperability and long-term flexibility. • Open-source or DIY frameworks (e.g., Arduino-based solutions): allow experimentation at low cost but often lack adequate tooling and support to build complex systems, provide reliability, and adequate communication abstractions, requiring significant customization effort, and being limited to the devices that can be supported.

<p>Unique Value Proposition</p>	<p>The Babel Ecosystem empowers developers, integrators, and device manufacturers to build smart, privacy-preserving IoT and home automation systems that operate autonomously, offline, and without dependence on centralised cloud infrastructures.</p> <p>Unlike existing frameworks, Babel offers a self-configuring, modular, and decentralised architecture that supports interoperability across diverse devices and environments. Its reusable protocol abstractions – such as membership management and reliable communication – simplify development while ensuring resilience and long-term maintainability.</p> <p>By combining open-source accessibility with robust, production-grade extensions, Babel provides a scalable and sustainable foundation for creating next-generation IoT and home automation solutions that respect user privacy, reduce costs, and keep full control in the hands of those who deploy and use IoT and home automation solutions.</p>
<p>Competitors</p>	<p>The Babel Ecosystem operates within the space of different alternatives for IoT and home automation platforms, each offering relevant strengths but also significant limitations that the Babel ecosystem can address. Alongside solutions developed on the Babel ecosystem, there is also room for complementary alternatives that can interoperate with it.</p> <ol style="list-style-type: none"> 1. Cloud-based IoT platforms (e.g., AWS IoT Core, Microsoft Azure IoT, Google Home, Alexa) offer excellent scalability, device management, and integration with cloud analytics, many times by having centralised integration with third-party device developers that end-users have to register and interconnect. However, they require Internet connectivity and need the availability of centralised infrastructures, which increases operational costs and raises privacy concerns for end-users. Babel can complement these systems by providing the fundamental mechanisms for local, autonomous operation while still inter-operating with services from the cloud to the edge, however in a way that provides additional control for end-users. 2. Proprietary edge or home automation ecosystems (e.g., Cisco Solutions, IKEA Home) provide reliable solutions with user-friendly integration but are bound to specific vendors, hardware, or proprietary protocols. Babel’s open and modular design still offers the opportunity of interoperability with such systems while avoiding lock-in and supporting custom extensions for broader device compatibility. 3. Open-source and DIY solutions (e.g., Arduino) encourage experimentation and rapid prototyping, while empowering innovation at a low cost, but lack resilience and standardization for dependable operation in large-scale or commercial deployments. Babel’s modular runtime and reusable distributed protocols can strengthen these ecosystems by adding reliability and decentralised coordination without sacrificing openness and community-driven innovation. <p>By combining openness with robustness, Babel offers a unique bridge between experimental open frameworks and commercial IoT ecosystems, creating new opportunities for collaboration, integration, and co-development across the IoT and home automation landscape, while at the same time offering an alternative to existing solutions that is capable of providing reliability and privacy to end-users in a fully decentralised way that is able to cope with heterogeneous devices.</p>
<p>Timing</p>	<p>The Babel Ecosystem is currently at TRL 6, with its core components – including the runtime, modular communication abstractions, and prototype integrations in IoT and home automation environments – successfully validated in relevant settings.</p> <p>Within the next 12 to 24 months, the focus will be on industrial-grade consolidation and community expansion, guided by the following milestones:</p> <ul style="list-style-type: none"> • +3 months: Open-source release of the Babel core components and documentation; establishment of the initial developer community. • +8 months: Pilot deployments with IoT and home automation SMEs validating performance, interoperability, and offline operation in real environments, for TRL 7. • +10 months: Creation of the NOVA FCT spin-off, focusing on developing and licensing specialised modules for professional use. • +24 months: Transition to TRL 8, with several validated, reliable deployments and a growing community of contributors and adopters. <p>This phased approach ensures a balanced transition from research to exploitation, combining open-source community engagement with the gradual introduction of commercial</p>

	services. By the time Babel reaches TRL 8, it will be ready for wider market adoption, supported by strong technical validation, a clear IP strategy, and established pilot references.
IP Strategy	<p>The Babel Ecosystem will adopt a hybrid IP strategy balancing community-driven innovation with commercial protection:</p> <ul style="list-style-type: none"> • Core framework released under a permissive open-source license (e.g., Apache 2.0) to encourage adoption and community contribution. • Advanced modules, integration tools, and enterprise-grade features may be licensed under commercial agreements. • Trademarks (e.g., “Babel-Ecosystem” or another) will be registered to protect branding and ensure quality assurance across ecosystem contributions. • Filing a patent for the solution which will be implemented by the computer. This may be explored for specific runtime optimizations or decentralised protocol mechanisms unique to the Babel ecosystem. • A possibility of dual licensing model: i) for academic purposes used for free and; ii) commercial licensing exploitation by paid licenses and/or service or collaborative contracts. • Licensing will be achieved by commercial agreement to make specific tools and components available for developers and integrators. Licensing for device manufactures will be based on agreements, per component, for specific device types and models, which allows the licensing to be compatible with the fact that the solution does not require Internet connectivity to operate.
Exploitation Roadmap	
Actions	<p>Within the first six months after the project’s completion, NOVA FCT will focus on consolidating the technical, legal, and organizational basis for the exploitation of the Babel Ecosystem. The main objectives are to ensure sustainability, secure IP, and prepare for commercial and community-driven adoption in the IoT and home automation sectors primarily.</p> <p>Planned actions:</p> <ol style="list-style-type: none"> 1. Finalize, unify and stabilize the open-source core components (Babel, Micro-Babel, Babel-Android, Babel-Swarm) and establish their governance between NOVA FCT and the main researchers that have been driving the design and development of Babel, potentially leading to the creation of a spin-off owned partially between these entities. 2. Develop a commercialization plan defining the business model (hybrid open-source and licensed modules as well as consulting and training services), pricing structure, and potential partnerships with third-party entities for integration and potentially also for the training services. 3. Implement the IP protection strategy: file a patent for the Babel core mechanisms and decentralised protocol execution model; register the “Babel Ecosystem” trademark (it should be noted that a Internet Standard Draft has already been prepared and delivered with the overall idea of the framework for a more general purpose, hence the patent must be focused on specific components and features tailored for IoT and home automation, including self-configuration and specialised support distributed protocols). 4. Identify Deploy pilots that can be used for demonstration/validation in IoT and home automation contexts, focusing on interoperability, resilience, and privacy-preserving operation without cloud reliance. 5. Launch a developer and adopter community, including public repositories (GitHub), technical documentation, tutorials, and a communication channel for feedback and support, potentially already under the spin-off created for the exploitation of the Babel Ecosystem. 6. Prepare dissemination and promotional activities: product one-pager, website, and presentation materials targeting IoT developers and integrators and home automation domains for presentation in fairs and industry/start-up driven events.

	<p>7. Initiate funding applications for scaling activities (e.g., EIC Transition, national innovation grants, or collaboration with industrial partners, or venture capitalists - taking advantage of the network and brand NOVA as much as possible).</p>																				
<p>Roles</p>	<p>The exploitation of the Babel Ecosystem will be coordinated and led by NOVA FCT, which retains full ownership of the results and is responsible for ensuring their technical evolution, community governance, and commercial valorization. The roadmap foresees a progressive distribution of roles as the ecosystem evolves.</p> <p>NOVA FCT (IP Owner and Technical Leader)</p> <ul style="list-style-type: none"> • Maintains and evolves the open-source core components of Babel, ensuring stability, quality, and alignment with research outcomes. • Coordinates the creation of the spin-off company, providing technical guidance and ensuring strategic continuity. • Provides the IP to the Spin-off. <p>Babel Spin-off (Commercial and Operational Arm)</p> <ul style="list-style-type: none"> • Leads community building, training, and outreach actions to expand adoption across industry. • Oversees the protection of intellectual property, manages the Babel trademark, and defines licensing terms for commercial modules. • Manages the commercialization of advanced Babel modules and professional services (consulting, integration, certification, and training). • Builds and maintains industrial partnerships with SMEs and device manufacturers to co-develop or license new modules. • Ensures market visibility and brand recognition through communication campaigns and participation in trade and technology events. <p>SMEs and Early Adopters (Validation Partners)</p> <ul style="list-style-type: none"> • Participate in pilot deployments during the validation phase to assess Babel's performance, interoperability, and usability in real IoT and home automation environments. • Provide feedback and co-create extensions or applications that demonstrate Babel's practical value and potential in different market segments. <p>Open-Source Community (Collaborative Development)</p> <ul style="list-style-type: none"> • Contributes to the continuous improvement of the open-source core through feedback, testing, and new module contributions. • Acts as a driver of innovation, ensuring that Babel remains relevant, interoperable, and adaptable to new technological trends. <p>This clear distribution of roles ensures that NOVA FCT maintains scientific and strategic leadership, while the spin-off and early adopters expand Babel's impact across industrial and open-source ecosystems.</p>																				
<p>Milestones</p>	<table border="1"> <thead> <tr> <th data-bbox="395 1514 775 1563">Milestone</th> <th data-bbox="775 1514 963 1563">Timeline</th> <th data-bbox="963 1514 1393 1563">KPIs / Indicators</th> </tr> </thead> <tbody> <tr> <td data-bbox="395 1563 775 1671">M1 - Open-source release of the integrated Babel ecosystem core components</td> <td data-bbox="775 1563 963 1671">3 months</td> <td data-bbox="963 1563 1393 1671">Public repositories available; documentation and examples published;</td> </tr> <tr> <td data-bbox="395 1671 775 1778">M2 - Creation of the Spin-off</td> <td data-bbox="775 1671 963 1778">4 months</td> <td data-bbox="963 1671 1393 1778">Spin-off established with clear ownership and transfer of IP from NOVA FCT to the spin-off.</td> </tr> <tr> <td data-bbox="395 1778 775 1854">M3 - IP protection established</td> <td data-bbox="775 1778 963 1854">5 months</td> <td data-bbox="963 1778 1393 1854">Patent filed and trademark registered</td> </tr> <tr> <td data-bbox="395 1854 775 1930">M4 - Pilot deployments defined</td> <td data-bbox="775 1854 963 1930">5 months</td> <td data-bbox="963 1854 1393 1930">2 distinct pilots are defined and specified.</td> </tr> <tr> <td data-bbox="395 1930 775 2002">M5 - Business Plan and Commercial Model</td> <td data-bbox="775 1930 963 2002">6 months</td> <td data-bbox="963 1930 1393 2002">Licensing and pricing strategy defined</td> </tr> </tbody> </table>			Milestone	Timeline	KPIs / Indicators	M1 - Open-source release of the integrated Babel ecosystem core components	3 months	Public repositories available; documentation and examples published;	M2 - Creation of the Spin-off	4 months	Spin-off established with clear ownership and transfer of IP from NOVA FCT to the spin-off.	M3 - IP protection established	5 months	Patent filed and trademark registered	M4 - Pilot deployments defined	5 months	2 distinct pilots are defined and specified.	M5 - Business Plan and Commercial Model	6 months	Licensing and pricing strategy defined
Milestone	Timeline	KPIs / Indicators																			
M1 - Open-source release of the integrated Babel ecosystem core components	3 months	Public repositories available; documentation and examples published;																			
M2 - Creation of the Spin-off	4 months	Spin-off established with clear ownership and transfer of IP from NOVA FCT to the spin-off.																			
M3 - IP protection established	5 months	Patent filed and trademark registered																			
M4 - Pilot deployments defined	5 months	2 distinct pilots are defined and specified.																			
M5 - Business Plan and Commercial Model	6 months	Licensing and pricing strategy defined																			

	M6 - Community basis established	6 months	Creating of online presence and tools for community drive contributions. >= 1000 Repository visits per month; >= 25 users active per month.
Costs	1 st Year - 250,000€ - For developers (consolidation, documentation, preparing pilots), IP filling, creation of spin-off, business plan, and dissemination activities: <ul style="list-style-type: none"> • 120,000€ - Developers and technical staff • 100,000€ - IP filling and creation of spin-off • 30,000€ - Dissemination activities 2 nd and 3 rd Year - 350,000€ - For developers (advanced modules to be licensed, improvements, interactions with community), maintenance, dissemination: <ul style="list-style-type: none"> • 210,000€ - Developers and technical staff • 50,000€ - Pilot development and validation: equipment, licenses, and integration tools for demonstration sites. • 60,000€ - Community Management • 30,000€ - Dissemination activities 		
Revenues	0-1 year - 40,000€ - Mostly from consulting and training: <ul style="list-style-type: none"> • 40,000€ - Consulting 1-3 year - 300,000€ - Licencing, technical support contracts, consulting, training: <ul style="list-style-type: none"> • 50,000€ - Training Activities • 100,000€ - Consulting • 150,000€ - Licensing 		
Other sources of coverage	<p>Internal research funding from NOVA FCT will provide initial support during the first six months after project completion, covering the costs of spin-off preparation and technical consolidation. This support includes granting free ownership of the IP in exchange for NOVA's participation in the spin-off, as well as the allocation of technical staff for initial diligence, setup, and documentation efforts.</p> <p>Applications to European innovation programs (e.g., EIC Transition or equivalent Horizon Europe innovation actions) are planned within the first 6 to 9 months post-project, aiming to co-finance the unification and consolidation of the existing Babel components and to support the identification of early adopters for commercial case studies. These applications will align with the roadmap's first validation phase (M9 milestone) to ensure continuity of development and pilot deployment.</p> <p>Partnerships with industrial stakeholders in the IoT and home automation sectors will be pursued from the first year onward, both to provide pilot sites and to co-develop specialised modules that demonstrate Babel's practical value. These collaborations will serve as the foundation for early licensing opportunities and as validation cases for future funding proposals and should be established in 12 months.</p> <p>Venture capital investment will be sought in the second year, once the spin-off is established and initial market validation has been achieved, to support scaling operations, marketing, and team expansion.</p> <p>Funding timelines may introduce uncertainties, particularly regarding open calls for funding that have variable response times. To mitigate these risks, NOVA FCT will ensure operational continuity through internal support and ongoing collaborations, maintaining essential development and moving forward with community establishment activities while awaiting external funding results. Contingency actions include adjusting pilot deployment schedules and accelerating contacts with industrial partners for co-funded demonstration building activities if external funding is delayed.</p>		

NOVA Edge Spin-off Business Plan

NOVA Edge is the technology spin-off of NOVA School of Science and Technology (FCT NOVA) dedicated to commercialising the Babel Ecosystem, a programming environment for

building autonomous, privacy-preserving, and decentralised IoT and domotics systems. The company adopts a hybrid open-source and commercial model: the core of Babel is freely available, fostering a global developer community, while NOVA Edge generates revenue from licensing advanced modules, integration and consulting services, training programmes, and participation in collaborative research and innovation projects. The financial plan anticipates approaching break-even at the end of Year 4 and effectively reaching it in Year 5.

Organisation

NOVA Edge will be majority-owned by NOVA FCT, ensuring alignment with scientific objectives and continuity in long-term research. Technical leadership is provided by João Leitão (CTO, 15% time) and Nuno Preguiça (Co-CTO, 15% time). Engineering capacity comes from Diogo Jesus (Product Manager), João Brilha (Senior Engineer), and Rafael Matos (Senior Engineer). Operational responsibilities are organised around: Babel Core and Integrated Devices; Protocols and System Integration; Community and Ecosystem Development; Business Development and Sales. NOVA FCT's IRIS office will support administrative, legal, and technology-transfer functions.

Financial Plan

Over the first five years NOVA Edge aims to: reach break-even in Year 5; reach a turnover above €700,000 in Year 5; maintain a gross margin above 50% from Year 4 onward; achieve an EBITDA margin of ~10% by Year 5; maintain liquidity above 10% of yearly expenditure; demonstrate long-term sustainability through diversified revenue streams.

Personnel Costs

All personnel costs follow Portuguese labour law: 14 months of salary per year and employer social-security contributions (23.75%). CTO and Co-CTO contributions appear both as costs and as in-kind institutional contributions (reflecting NOVA FCT's commitment). Personnel accounts for roughly 70% of total expenditure.

Role of Employee	Annual Salary (€)	Employer Additional Taxes	Annual Cost (€)	Total
Chief Technical Officer (João Leitão) – 15%	8,302	23.75%	10,274	
Co-Chief Technical Officer (Nuno Preguiça) – 15%	10,752	23.75%	13,306	
Product Manager (Diogo Jesus)	50,000	23.75%	61,875	
Senior Engineer (João Brilha)	40,000	23.75%	49,500	
Senior Engineer (Rafael Matos)	40,000	23.75%	49,500	
Community Manager (to be hired)	28,000	23.75%	34,650	
Sales Manager (to be hired)	65,000	23.75%	80,437	
Junior Engineer (hired start Y3)	35,000	23.75%	43,312	
Junior Engineer (hired start Y4)	35,000	23.75%	43,312	
Junior Engineer (hired start Y5)	35,000	23.75%	43,312	

Summary of personnel costs per year:

Year	Filled Positions	Total Personnel Costs (€)
Year 1	CTO, Co-CTO, Product Manager, 2 Senior Engineers, Community Manager, Sales Manager	299,542
Year 2	Same as Year 1	299,542
Year 3	Same as Year 2 + 1 Junior Engineer	342,854
Year 4	Same as Year 3 + 1 Junior Engineer	386,166
Year 5	Same as Year 4 + 1 Junior Engineer	429,478

Operational Costs

Operational costs represent all non-personnel recurring expenses. These values remain stable across the first five years, since office space, cloud infrastructure, and support services do not scale significantly with the early team size.

Type of Cost	Annual Cost (€)
Office/lab space	5,000
Cloud services, hosting, open-source repositories	4,000
Stationary, Printing, Equipment depreciation	3,000
Access to testbed equipment	10,000
Accounting + Auditing Services	6,000
Legal Support + IP Protection	10,000
Marketing and Dissemination Activities	30,000
Total operational costs per year	68,000

Valuation of In-Kind Contribution of NOVA FCT

To ensure the viability of the company, NOVA FCT (major owner of the spin-off) will contribute in-kind. Values reported are per year and are not expected to change significantly during the first five years of operation.

Contribution	Estimated Value (€)
Use of Laboratory and Office Space	5,000
Access to testbed equipment	10,000
IRIS Administrative and Legal Support	16,000
Time of CTO (João Leitão) – 15%	10,274
Time of Co-CTO (Nuno Preguiça) – 15%	13,306
Total In-Kind Contribution per Year	54,580

Founder Capital Contribution

Founder	Initial Capital Contribution (€)	Investment Fraction
CTO (João Leitão)	5,000	33.33%
Co-CTO (Nuno Preguiça)	5,000	33.33%
Product Manager (Diogo Jesus)	2,500	16.67%
Senior Engineer (João Brilha)	1,250	8.33%
Senior Engineer (Rafael Matos)	1,250	8.33%
Total Founder Capital	15,000	100%

At incorporation, NOVA Edge will receive €15,000 in initial paid-in capital from the founding team. This initial capital supports early liquidity and complements institutional in-kind contributions from NOVA FCT.

Revenue Model

Four main revenue streams: licensing of advanced Babel modules; consulting and integration services; training and certification activities; research and development project participation.

Licensing Revenue Forecast:

Year	SME Licences (€5K)	Enterprise Licensing (€20K)	Licence Revenue over the Year (€)
Year 1	0	0	0
Year 2	1	0	5,000
Year 3	2	1	30,000
Year 4	4	2	60,000
Year 5	6	3	90,000

Consulting, Training, and Research Projects Revenue:

Year	Consulting (€)	Training (€)	R&D Participation (€)	Project	Total Revenue over the Year (€)
Year 1	30,000	10,000	0		40,000
Year 2	60,000	20,000	65,000		145,000
Year 3	80,000	25,000	130,000		235,000
Year 4	120,000	35,000	215,000		370,000
Year 5	150,000	40,000	265,000		455,000

Summarisation of Revenue:

Year	Licensing (€)	Consulting/Training/R&D (€)	Total Revenue (€)
Year 1	0	40,000	40,000
Year 2	5,000	145,000	150,000
Year 3	30,000	235,000	265,000
Year 4	60,000	370,000	430,000
Year 5	90,000	455,000	545,000

Projected Income (P&L):

Year	In-Kind (€)	Personnel Costs (€)	Operational Costs (€)	Revenue (€)	EBITDA (€)	EBITDA Margin	Net Result (€)
Year 1	54,580	299,542	68,000	40,000	-327,542	—	-327,542
Year 2	54,580	299,542	68,000	150,000	-217,542	—	-217,542
Year 3	54,580	342,854	68,000	265,000	-145,854	—	-145,854
Year 4	54,580	386,166	68,000	430,000	-24,166	-5.6%	-24,166
Year 5	54,580	429,478	68,000	545,000	+47,522	8.7%	+47,522

Funding strategy: NOVA FCT internal seed (first 6 months); applications to EIC Transition / Horizon Europe innovation actions (months 6–9 post-project); industrial partnerships from Year 1; venture capital investment from Year 2 onward, once the spin-off is established and initial market validation has been achieved. Funding timelines may introduce uncertainties; NOVA FCT will ensure operational continuity through internal support and ongoing collaborations.

5.2 KER-3: IMPROVED ACTYX MIDDLEWARE

KER	Individual intellectual asset	Protection measure	Reasoning
KER 2 - Improved Actyx Middleware	Middleware source code	Copyright + proprietary licence	Core value asset; copyright + licence enforce commercial use
	Workflow management logic under eventual consistency	Copyright + trade secrets	Hard to patent, valuable in practice; secrecy preserves advantage
	Industrial integration logic (e.g., grade runtime behaviour) and performance tuning methods	Trade secrets + NDAs	Know-how is hard to replicate and critical for customers
	Documentation & operator manuals	Copyright	Enables paid training and support services
	Brand and product identity	Trademark	Prevents competitors from free riding on reputation

Description	The improved Actyx middleware is a decentralised middleware designed to enable the development of highly resilient applications in the manufacturing sector. It supports fully decentralised operations, ensuring critical industrial processes continue uninterrupted even when central infrastructure is unavailable. The middleware leverages strong eventual consistency to manage workflows and data in a decentralised environment, optimising reliability and scalability in mission-critical operations.
Target market/end users	<ul style="list-style-type: none"> Manufacturing companies of all sizes, with a particular emphasis on small- to medium-sized enterprises (SMEs) that are increasingly turning to open-source solutions. Vertical sectors such as white-goods manufacturers and other industrial domains requiring tailored shop floor management applications. Organizations seeking flexible, scalable, and cost-efficient IT solutions for decentralised manufacturing operations.

<p>End-users needs / problems</p>	<ul style="list-style-type: none"> ● Need for highly resilient, decentralised applications that ensure uninterrupted operations, even when central infrastructure fails. ● Demand for faster, lower-cost application development tailored to specific companies or sectors. ● Pressure to adapt to Industry 4.0 trends, including automation, real-time analytics, and digital transformation. ● Growing importance of SaaS, AI, and generative AI capabilities in industrial software solutions. ● SMEs, in particular, face the challenge of finding affordable yet powerful open-source solutions that can compete with proprietary systems while supporting scalability.
<p>Competitive advantages</p>	<ul style="list-style-type: none"> ● The improved Actyx middleware offers fully decentralised operations, ensuring resilience and continuity in mission-critical processes. ● Its open-source model allows for wide adoption and customization by UNIWA or other organizations, reducing development costs and accelerating time-to-market. ● Built on strong eventual consistency, it optimizes reliability and scalability in decentralised environments. ● Facilitates sector-specific solutions (e.g., white-goods, automotive, or other manufacturing domains), giving flexibility compared to generic centralised software. ● Positioned in a fast-growing market (industrial software, MES, MOM) that is increasingly embracing open-source, SaaS, and AI-driven innovations, aligning well with industry trends.
<p>Use model</p>	<p>The improved Actyx middleware (KER) will be exploited primarily through an open-source use model, making it widely accessible to manufacturing companies, particularly SMEs. Adoption will be supported by the provision of services such as training, integration, and customization, enabling faster and lower-cost development of sector-specific applications (e.g., white-goods or automotive manufacturing). While the core remains open source, complementary modules and adaptations can be commercialised through technology transfer, licensing, or contract research with industrial partners. In parallel, results will be disseminated via publications and contributions to standards, reinforcing credibility and fostering wider uptake across the growing industrial software market.</p>
<p>Early Adopters</p>	<p>The most likely early adopters are small- to mid-sized manufacturing companies (SMEs) that are increasingly open to open-source, cost-efficient, and flexible IT solutions for shop floor management. Larger industrial players with strong digital transformation agendas may also adopt the middleware in pilot projects, but SMEs represent the primary early customer base due to their need for affordable, adaptable solutions.</p>
<p>Adopters' problems/needs</p>	<ul style="list-style-type: none"> ● SMEs often lack resources for complex, centralised IT infrastructure and need resilient, low-cost solutions that ensure operations continue even without central servers. ● Manufacturers need flexibility and faster software development cycles to keep up with Industry 4.0 trends (● There is a demand for tailored solutions (automation, analytics, AI-driven processes). ● sector- or company-specific) rather than generic platforms. ● Ensuring scalability, interoperability, and reliability in distributed environments remains a challenge, especially when integrating diverse devices and systems.
<p>Alternative solution</p>	<p>Commercial MES (Manufacturing Execution Systems) and MOM (Manufacturing Operations Management) software (e.g., Siemens Opcenter, Rockwell FactoryTalk, SAP Digital Manufacturing) already serve this market, offering robust centralised shop floor management solutions. However, these are often costly, complex, and vendor-locked, making them less attractive to Sessom open-source or hybrid solutions exist, such as Odoo Manufacturing or open-source MES platforms, but most lack the decentralised resilience and event-driven model of the improved Actyx middleware. Cloud-based SaaS shop floor tools (like Tulip, Katana, or SafetyCulture) provide flexibility but remain dependent on centralised infrastructures and subscription models, which may not fit all manufacturers' needs.</p>

Unique Value Proposition	The improved Actyx middleware differentiates itself by combining open-source accessibility, fully decentralised resilience, and sector-specific adaptability , directly addressing SMEs' cost and flexibility challenges while offering reliability comparable to enterprise-grade proprietary systems.
Competitors	<ul style="list-style-type: none"> • Proprietary MES / MOM Vendors • Cloud-based SaaS Shop Floor & Operations Tools • Open-Source / Hybrid Competitors
Timing	The improved Actyx middleware is already available as an open-source solution , meaning its core components are ready for immediate adoption and pilot use in industrial settings. This significantly shortens the time to market, as organizations can begin experimenting and integrating the middleware without delay. However, the development of sector-specific applications, integration services, and training modules is expected to take an additional 6 to 18 months , depending on the complexity of each use case. Broader industrial adoption and ecosystem integration are projected within 2 to 3 years , in line with ongoing Industry 4.0 trends and the increasing preference for open-source solutions in manufacturing.
IP Strategy	<ul style="list-style-type: none"> • Background IP: Original Actyx middleware and foundational technologies developed by ACT (open source). • Foreground IP: Improvements, refinements, and sector-specific adaptations developed under TaRDIS (led by UNIWA and partners). • IP Management Activities: Clear documentation of contributions, consistent licensing, monitoring adoption, and participation in open source/standards communities. • Exploitation Approach: Open-source core complemented by dual-licensing, service offerings, technology transfer, and potential patents for novel algorithms or industrial modules.
Exploitation Roadmap	
Actions	For this KER, the next steps include investigating its usability and limitations through real-world testing and feedback from early adopters, ensuring that the middleware can effectively meet industrial needs. In parallel, the consortium will assess opportunities to establish patents or other IPR protection schemes for novel components that extend beyond the open-source core. Efforts will also focus on identifying and engaging interested stakeholders , both as potential customers and as partners for future collaborations, to accelerate adoption and co-development. Finally, the project will work on developing integrated modules tailored to specific use cases and market segments , supporting targeted applications and laying the groundwork for future implementations and commercialization opportunities.
Roles	UNIWA and CMS will lead all the aforementioned activities.
Milestones	<ul style="list-style-type: none"> • KER efficiency and real-time operation evaluation (M3) • Patent/IPR establishment (M18) • Business plan formulated (M20) • Prototype complete (M24) • KPI: KER demonstrated to at least 3 stakeholders (M24)
Costs	<p>For the further development of the tools to be TRL 9 for the market, additional effort is required.</p> <p>At least 40 PMs will be used if further funding is received for this purpose.</p> <p>At the 3rd year, a loan of 50.000 euros would be necessary, if needed, for paying part of the salaries of 2 employees. Regarding the costs of realisation of a prototype, there is no indication of the cost of hardware at that time and since there are several problems in the production chain, no foreseen cost can be provided.</p>
Revenues	Further funding of 10 to 15k will be needed.

Other sources of coverage	Future EU project funding: dedicated industry investments targeting specific customer segments will be required to achieve TRL9 within two years following the project's completion.
----------------------------------	--

5.3 KER-4: PRIVACY-PRESERVING DECENTRALISED TRAINING

KER	Individual intellectual asset	Protection measure	Reasoning
KER 3 - Privacy-Preserving and Efficient Decentralised Training	Privacy-preserving mechanisms (e.g., DP implementations) and core FL mechanisms	Copyright + open and controlled licensing	Algorithms gain value through validation and trust. Users pay for compliance assurance.
	Compliance tooling	Auditing, logging, reporting modules	Direct monetisation point for regulated sectors
	Handling of heterogeneous devices	Copyright + trade secrets (partial)	Practical tuning is competitive advantage.
	Service operation know-how	Trade secrets	Core value for "AI as a service" offerings
	Branding and service naming	Trademark	Builds trust in privacy-critical markets

Description	The FLaaS v2 framework extends the Federated Learning as a Service (FLaaS) platform developed under TaRDIS, enhancing its capacity to support privacy-preserving, heterogeneous, and decentralised AI training. It integrates differential privacy mechanisms, Split Learning techniques for managing data and model heterogeneity, and decentralised intelligence features to enable autonomous, resilient learning across distributed environments. This evolution makes FLaaS v2 a versatile solution for data-sensitive industries, where regulatory compliance, scalability, and data sovereignty are critical.
Target market/end users	<ul style="list-style-type: none"> • Data-sensitive industries such as manufacturing, energy, and IoT, requiring privacy-preserving learning. • Cloud service providers and AI infrastructure operators aiming to offer federated AI services. • Research institutions and innovation labs working on decentralised AI and privacy technologies. • Public-sector and regulated domains (e.g., health, utilities) seeking GDPR-compliant distributed AI solutions.
End-users needs / problems	<ul style="list-style-type: none"> • Guaranteeing privacy and security in distributed learning environments. • Reducing dependency on centralised infrastructure and single points of failure. • Managing heterogeneous data sources and device capabilities efficiently. • Complying with regulatory frameworks (e.g., GDPR, AI Act) while maintaining learning performance. • Lowering costs and technical barriers for adopting federated AI solutions.
Competitive advantages	<ul style="list-style-type: none"> • Integrates privacy-preserving techniques such as differential privacy and Split Learning. • Fully decentralised operation ensuring resilience and autonomy. • Modular, open design enabling interoperability and easy extension. • Supports diverse environments with heterogeneous devices and networks. • Aligns with European digital sovereignty and trustworthy AI priorities.

Use model	FLaaS v2 will be exploited under a mixed model combining open-source accessibility with service-based extensions. The core platform will remain open to foster adoption by industry and research, while added-value components such as specialised privacy modules, integration services, and deployment support will be commercialised. Partners will leverage technology transfer, consulting, and customisation services to ensure sustainability and market reach.
Early Adopters	<ul style="list-style-type: none"> • SMEs and mid-sized manufacturers exploring AI integration under strict data governance. • Telecommunications and cloud providers (e.g., Telefonica) adopting federated AI services. • Research institutions piloting distributed AI frameworks in collaborative experiments.
Adopters' problems/needs	<ul style="list-style-type: none"> • Need for frameworks supporting hybrid (cloud-edge) federated learning. • Lack of accessible, regulatory-compliant distributed training solutions. • Challenges in managing diverse device data and model versions. • Need for cost-efficient AI infrastructure for decentralised applications.
Alternative solution	<ul style="list-style-type: none"> • Commercial FLaaS alternatives by Google, IBM, or NVIDIA provide strong cloud integration but are centralised and closed source. • Open-source frameworks like Flower or FedML support federated learning but lack privacy-preserving and decentralised extensions. • Decentralised AI platforms (e.g., Ocean Protocol, SingularityNET) address autonomy but not industrial or regulatory requirements.
Unique Value Proposition	FLaaS v2 uniquely combines privacy-preserving federated learning, heterogeneity management, and decentralised AI capabilities within a European open-source framework. It directly supports data sovereignty, regulatory compliance, and industrial scalability – bridging the gap between research prototypes and real-world deployable distributed intelligence solutions.
Competitors	
Timing	Core components are available from the baseline FLaaS release, enabling immediate evaluation. Validation and refinement through pilot use cases (e.g., wake-word detection) are ongoing. Market-ready versions, including integration services and dedicated modules, are expected within 18-24 months following project completion.
IP Strategy	<p>Background IP: FLaaS baseline framework and original design developed under TaRDIS.</p> <p>Foreground IP: Privacy, decentralisation, and heterogeneity enhancements by Telefonica, UNIWA, and partners.</p> <p>IP Management: Clear documentation of partner contributions, open-source licensing (e.g., Apache 2.0 / GPL), and potential patenting for novel privacy-preserving algorithms.</p> <p>Exploitation: Open-source release with optional service commercialisation and technology transfer agreements.</p>
Exploitation Roadmap	
Actions	Next steps will focus on validating our framework in real-world industrial scenarios and expanding its adoption through collaborations with early adopters and technology partners. These actions include refining differential privacy mechanisms, optimising decentralised coordination protocols, and ensuring scalability across diverse devices. Parallel efforts will target regulatory alignment, business model refinement, and dissemination through workshops and scientific publications.
Roles	<ul style="list-style-type: none"> • Telefonica I+D (TID): Lead partner for framework enhancement, validation, and exploitation. • CMS: Support in open-source management and community engagement

	<ul style="list-style-type: none"> • UNIWA: Support in academic dissemination • Other consortium partners: Pilot testing, integration, and co-development of specialised modules.
Milestones	<ul style="list-style-type: none"> • M6: Initial FLaaS prototype validated in lab environment. • M12: Integration of privacy and heterogeneity modules. • M18: Industry pilot validation completed (e.g., wake-word use case). • M21: Business plan finalised and IP strategy consolidated. • M24: Framework ready for market-oriented exploitation and partnership expansion.
Costs	To achieve higher TRL maturity, approximately 30 person-months will be required for continued development, validation, and support activities. An estimated €50,000 loan or co-funding will be necessary within two years post-project for technical staff and prototype evolution.
Revenues	Initial revenues will derive from collaborative projects, service contracts, and training activities. Additional revenue streams are expected from integration and consultancy services for industrial partners adopting FLaaS v2.
Other sources of coverage	Future EU funding opportunities (e.g., Horizon Europe, Digital Europe) and private-sector co-investment will be explored to support industrialisation. Strategic collaborations with industry clusters, open-source communities, and digital innovation hubs will be leveraged for sustainability.

6 UPDATED INDIVIDUAL EXPLOITATION PLANS

To wrap up the Exploitation and Sustainability Roadmap for the TaRDIS project, the consortium took a last, in-depth look at every partner’s exploitation plan. This wasn’t just a repeat of the mid-term planning, it was a chance to capture where things actually landed at the end: what’s ready, what works, and what everyone learned through the entire project.

Each partner was asked to update their plan with specifics: which Key Exploitable Results (KERs) they own or shape, how mature those results are now (with proof from prototypes, integrations, or use-case work), and exactly how they’ll move things forward after the project wraps up, whether that’s open-sourcing, rolling out new products or services, launching pilots, sharing results with the research community, or chasing new follow-on proposals. Everyone also spelled out their IP and licensing strategy, plus their commitments around keeping things alive: who handles governance, looks after repositories, secures resources, and ensures everything doesn’t just fade away.

This whole process serves three clear goals. First, it delivers a solid, auditable connection between what TaRDIS delivered and who’s taking it forward, along with concrete actions and timelines. That turns a high-level plan into something credible and real. Second, it lines up the final project narrative with each partner’s actual situation, technology readiness, dependencies, challenges, or commercial plans, so there’s less risk of fragmentation and more focus on how everything fits together. Third, it gives us a foundation for proper post-project follow up, with partner KPIs, milestones, and risk management collected in one solid sustainability plan.

All of this feeds directly into this deliverable. We use it to cross-check and improve the final list of TaRDIS KERs (including what’s highlighted by the Innovation Radar), document exactly what each partner promises to do and who’s responsible, and build a clear set of action points for exploitation and sustainability, backed up, where it makes sense, by new funding or entrepreneurial opportunities. The whole point is to make sure the impact of TaRDIS keeps growing even after the official project ends.

6.1 PARTNER: NOVA SCHOOL OF SCIENCE & TECHNOLOGY (NOVA)

Organisation name	NOVA School of Science & Technology		Short	NOVA
Contact person for exploitation	Name	Carla Ferreira	Partner Type	University
	Email	carla.ferreira@fct.unl.pt		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input checked="" type="checkbox"/> GMV <input checked="" type="checkbox"/> EDP <input checked="" type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Babel Framework and Ecosystem
- KER-2: Decentralised Data Management Framework (Nimbus, PotionDB, Arboreal)
- KER-3: TaRDIS Toolbox Integration Layer and Observability Infrastructure

KER Title	Babel Framework and Ecosystem
------------------	--------------------------------------

Short Description	
<p>A Java framework and multi-target runtime ecosystem for building decentralised, event-driven distributed protocols. The ecosystem (Babel-Swarm, Babel-Android, Micro-Babel) enables protocol-centric programming across heterogeneous device classes - from server-class machines to Android mobile devices and resource-constrained embedded hardware (ESP32, RP2040). Protocols are composable units that interact exclusively through asynchronous events, simplifying the development of complex distributed applications. The Babel Ecosystem has been identified as the primary Key Exploitable Result of WP6, validated at TRL 6 in IoT and smart-home environments, and is already being adopted by companies in Portugal for industrial IoT innovation.</p>	
Innovation Radar alignment	<input checked="" type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> prototype <input type="checkbox"/> validated in lab TRL estimate 6 <input checked="" type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready
Validation evidence	<p>Large-scale 5,000-node decentralised messaging experiment (D6.3 [8]) demonstrating 99.49% message delivery reliability and ~1.2s average end-to-end latency; TaRDIS Messaging App demonstrator running across cloud servers, Raspberry Pi nodes, Android smartphones, and ESP32 embedded devices simultaneously; 48+ hours continuous battery operation on commodity Android hardware; ParadigmShift early-adopter validation in commercial IoT product context.</p>
Pain Point (i.e., what problem does it solve?)	
<p>IoT development today requires distinct programming models, frameworks, and toolchains for each hardware tier (cloud, mobile, embedded), forcing developers to implement the same logic multiple times and creating fragile, cloud-dependent architectures. This results in high development costs, tight vendor lock-in to cloud providers, poor resilience when connectivity is intermittent, and difficulty building systems that span heterogeneous device fleets. No existing open platform bridges all three tiers (JVM/server, Android, bare-metal embedded) with a unified abstraction while supporting decentralised, peer-to-peer operation without a cloud broker.</p>	
Value proposition (i.e., why it is better than the alternatives?)	
<p>Our unified programming ecosystem helps IoT solution providers build innovative, resilient, and interoperable IoT systems while reducing over-reliance on the cloud and lowering development costs. Babel provides a single, coherent API that compiles and runs natively on JVM servers, Android smartphones, and ARM/ESP32 embedded hardware running FreeRTOS, eliminating the need for separate protocol stacks, reducing integration effort, and enabling resilient operation in disconnected or intermittently connected environments without any cloud dependency.</p>	
Target users / customers	
<ul style="list-style-type: none"> • Primary adopters: IoT platform developers and embedded systems companies building decentralised IoT products; technology startups requiring resilient, cloud-independent device communication (e.g., ParadigmShift). • Secondary adopters: Telecom operators (e.g., Telefónica) deploying IoT infrastructure; smart city platform providers; industrial IoT system integrators; energy sector operators managing distributed assets (e.g., EDP); research institutions and academia. • Sectors/markets: IoT and Edge Computing; Telecommunications; Smart Cities and Building Automation; Industry 4.0 and IIoT; Energy and Utilities; Connected Mobility. 	
Dependencies and integration requirements	
<ul style="list-style-type: none"> • TaRDIS components needed: Babel-Swarm (JVM runtime), Babel-Android, Micro-Babel (C/FreeRTOS for embedded hardware); Nimbus decentralised key-value store (KER-2) for applications requiring distributed state management. • Third-party dependencies: JVM (Java 11+), Android SDK (API 26+), FreeRTOS (MIT licence), standard build toolchains (Gradle, CMake). MQTT gateway optional for interoperability with legacy broker-based systems. All third-party licences are compatible with Apache 2.0. • Hardware/network assumptions: Operates on ARM Cortex-M microcontrollers (e.g., ESP32), Android smartphones (ARM), and commodity JVM servers. Designed for intermittently connected, bandwidth- 	

constrained, and mesh/ad-hoc network topologies; does not require persistent internet connectivity or a cloud broker.

- **Skills required to adopt:** Java/Kotlin (for Babel-Swarm and Babel-Android); C (for Micro-Babel on FreeRTOS); basic familiarity with distributed systems and publish-subscribe messaging concepts. No prior knowledge of swarm computing or epidemic protocols is required.

Unique technical features

- Cross-tier unified runtime: same programming model and API across JVM servers, Android smartphones, and ARM/ESP32 bare-metal devices (FreeRTOS), making it the only open-source framework spanning all three tiers without a cloud broker.
- Protocol-centric design: communication protocols (epidemic propagation, ACO-based routing, gossip) are first-class citizens in the framework, enabling resilient, peer-to-peer message delivery without central infrastructure.
- Validated at scale: 5,000-node large-scale experiment achieving 99.49% message delivery reliability and ~1.2s average latency; 48+ hours battery operation on commodity Android devices (D6.3 §6).
- Disconnection-tolerant by design: operates in intermittently connected, mesh, and ad-hoc network environments with no dependency on continuous internet access or cloud broker availability.
- Open-source core with commercial extensibility: Apache 2.0 licence for the runtime core; advanced modules and managed deployment support available under commercial licence, enabling a sustainable dual-licensing business model.

Evidence and artefacts

Public deliverable(s): D6.1 (WP6 initial software release), D6.2 (WP6 intermediate software release), D6.3 (WP6 final software release and exploitation plan [8]), available on the TaRDIS project website.

Code repository (public/private): Public: <https://github.com/babel-project> (Apache 2.0). Includes Babel-Swarm, Babel-Android, and Micro-Babel source code, build scripts, and example applications.

Demo video/slide deck: TaRDIS Messaging App demo video (multi-device deployment across server, Raspberry Pi, Android, and ESP32); large-scale 5,000-node experiment recording. Available through TaRDIS project dissemination channels.

Benchmark/evaluation: 5,000-node large-scale decentralised messaging experiment: 99.49% message delivery reliability, ~1.2s average end-to-end latency, 48+ hours mobile battery life. Full methodology and results documented in D6.3 [8].

Publications: multiple peer-reviewed papers on Babel routing protocols and decentralised IoT middleware (ACM/IEEE venues), and an IETF Internet Standards Draft submitted for Babel-based protocol standardisation. A full list is provided in the TaRDIS dissemination report (D8.5 [14]).

Exploitation Plan (Next 12-24 Months)

Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release	<input checked="" type="checkbox"/> Services (integration, support, training)
	<input type="checkbox"/> Integration into partner product/service	<input checked="" type="checkbox"/> Research exploitation (follow-on projects)
	<input type="checkbox"/> Joint exploitation with other partners	<input checked="" type="checkbox"/> Standardisation / pre-standardisation influence
	<input checked="" type="checkbox"/> External pilots / proofs of concept	<input type="checkbox"/> Other:

Concrete actions and milestones

Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Stable open-source release of the Babel Ecosystem (Babel-Swarm v1.0, Babel-Android v1.0, Micro-Babel v1.0), with full documentation and a migration guide.	NOVA	Q2 2026	Internal dev capacity; IP clearance	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Developer community engagement: workshops, tutorials, developer wiki, active	NOVA	Q3 2026	M1 completed; conference participation	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done

	GitHub issue support, and publication of the Babel developer guide.				
M3	ParadigmShift pilot live; two additional IoT company LOIs signed; commercial support offers published; spin-off or licensing entity formation initiated.	NOVA / ParadigmShift	Q4 2026	M1, M2; legal/financial advisory for spin-off	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): stable open-source releases of Babel-Swarm v1.0, Babel-Android v1.0, and Micro-Babel v1.0 (Apache 2.0); Nimbus v1.0; and API documentation, developer tutorials, and integration guides published on GitHub and the project website. • Pilot deliverables: ParadigmShift commercial pilot live deployment; at least 2 additional pilot agreements signed with IoT companies; reproducible benchmark package for the 5,000-node experiment made publicly available. • Business outputs (offers, proposals): Commercial support and integration service offering published; dual-licensing model operationalised; spin-off or licensing entity formation initiated; follow-on funding application(s) submitted (e.g., EIC Transition, Horizon Europe). 					

KER Title	Decentralised Data Management Framework (Nimbus, PotionDB, Arboreal)				
Short Description					
A set of decentralised storage and replication components supporting swarm and cloud-edge environments. Nimbus provides a fully decentralised CRDT-based key-value store suited to volatile swarms with partial replication and offline tolerance. PotionDB provides transactional causal consistency and efficient repeated queries via materialised views maintained with non-uniform CRDT techniques. Arboreal targets hierarchical cloud-edge replication scenarios.					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input checked="" type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:				
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input type="checkbox"/> validated in lab <input checked="" type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready			TRL estimate	4-5
Validation evidence	Integration and demonstrations within TaRDIS toolbox iterations and use-case development/validation activities (D6.3 [8]; D7.3 [10]; D7.4 [11]).				
Pain Point (i.e., what problem does it solve?)					
Edge/swarm systems suffer from intermittent connectivity, churn, constrained bandwidth, and heterogeneous trust/infrastructure. Conventional databases assume stable infrastructure and full replication or central coordination, making them costly or brittle in swarms.					
Value proposition (i.e., why it is better than the alternatives?)					
<ul style="list-style-type: none"> • Provides data availability and consistency under churn/partitions with explicit partial replication options. • Supports both swarm-first (Nimbus) and geo/cloud-edge transactional scenarios (PotionDB/Arboreal). • Integrates with the TaRDIS runtime ecosystem and APIs, reducing integration cost for decentralised applications. 					
Target users / customers					
<ul style="list-style-type: none"> • IoT/edge platform developers and system integrators building decentralised applications. • Robotics/swarm application developers needing local-first state and reconciliation. • Telco/edge infrastructures and industrial operators needing resilient distributed state. 					

Dependencies and integration requirements					
<ul style="list-style-type: none"> • Babel ecosystem runtime and protocol stack for dissemination/communication. • Standard JVM/Python/C build toolchains. • Operational baseline: nodes with intermittent connectivity; no always-on cloud required for Nimbus. 					
Unique technical features					
<ul style="list-style-type: none"> • Two differentiated designs for different deployment realities (swarm vs geo/cloud-edge). • CRDT-based semantics with explicit partial replication controls (Nimbus). • Transactional causal consistency and view maintenance mechanisms (PotionDB). 					
Evidence and artefacts					
Deliverables: D6.3 [8] Use-case evaluation/validation: D7.3 [10] / D7.4 [11].					
Exploitation Plan (Next 12-24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input checked="" type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Package reproducible benchmarks + deployment scripts	NOVA	6 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Publish “when to use Nimbus vs PotionDB vs Arboreal” adoption guide	NOVA	9 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Identify 1-2 pilot candidates where data locality/bandwidth dominate	NOVA	18 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done

KER Title	TaRDIS Toolbox Integration Layer and Observability Infrastructure				
Short Description	Tooling and runtime support enabling deployment-time and runtime observability and controlled reconfiguration of decentralised applications, including telemetry collection/aggregation and operational control interfaces used during demonstrators and validations.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: TaRDIS observability & reconfiguration enabler				
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input type="checkbox"/> validated in lab <input checked="" type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	4-5		
Validation evidence	D6.3 (toolbox components) [8] and final validation/guidelines (D7.4) [11]				
Pain Point (i.e., what problem does it solve?)					

Decentralised swarms are difficult to operate. Debugging, monitoring, and safe runtime changes are a major barrier to adoption.					
Value proposition (i.e., why it is better than the alternatives?)					
Operational tooling reduces adoption risk by enabling observability, controlled reconfiguration, and repeatable deployments, the key for moving from demos to pilots.					
Target users / customers					
System integrators, platform operators, industrial pilots, DevOps teams working on edge systems.					
Dependencies and integration requirements					
Open artefacts, integration services.					
Exploitation Plan (Next 12-24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input checked="" type="checkbox"/> Services (integration, support, training) <input type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Package reproducible benchmarks + deployment scripts	NOVA	6 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Publish “when to use Nimbus vs PotionDB vs Arboreal” adoption guide	NOVA	9 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Identify 1-2 pilot candidates where data locality/bandwidth dominate	NOVA	18 Months		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done

IP, licensing and access rights

Background IP needed
<ul style="list-style-type: none"> • Your background needed: NOVA’s prior research on Babel routing protocols, JVM middleware, and distributed systems (pre-TaRDIS publications and open-source prototypes) forms the foundational background IP leveraged by the Babel Ecosystem developed in TaRDIS. • Third-party IP/licenses: FreeRTOS (MIT); Android SDK (Apache 2.0 + proprietary components); standard JVM libraries (Apache 2.0 / LGPL). All third-party licences are compatible with Apache 2.0 distribution of the Babel Ecosystem core.
Foreground ownership
<input checked="" type="checkbox"/> Single partner-owned <input type="checkbox"/> Joint ownership - partners: <input type="checkbox"/> Unsure / to be agreed
Licensing intentions (software / docs)
<ul style="list-style-type: none"> • Preferred licence (if open source): Apache 2.0 for the Babel Ecosystem core (Babel-Swarm, Babel-Android, Micro-Babel) and Nimbus. • Components intended to remain proprietary: Advanced protocol modules (e.g., enhanced security layers, managed deployment tooling, enterprise SLA support infrastructure); commercial integration adapters. These will be available under a commercial licence alongside the open-source core.

- **Constraints/notes (patents, NDAs, data restrictions):** A patent filing is under way for novel routing and consensus mechanisms developed in TaRDIS. There are no NDAs constraining publication or open-source release, and no personal data are involved. Trademark registration for the “Babel” framework is under consideration.

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> ● Components: Babel-Swarm (JVM), Babel-Android, Micro-Babel (C/FreeRTOS), Nimbus decentralised key-value store, all of which will be actively maintained post-project under the NOVA GitHub organisation. ● Documentation: API reference documentation, developer tutorials, integration guides, and architectural overview maintained on GitHub and project website; developer wiki with community contribution model. ● Demo/benchmark artefacts: TaRDIS Messaging App (source + deployment scripts); reproducibility package for the 5,000-node large-scale experiment (dataset, scripts, configuration) made publicly available on GitHub. 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input checked="" type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify): Notes: Hosted under the NOVA GitHub organisation (https://github.com/babel-project). Open contribution model welcoming community pull requests. Planned transition to a spin-off entity to provide commercial maintenance continuity post-project.

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # software releases: 3 (Babel-Swarm v0.x, Babel-Android v0.x, Micro-Babel v0.x) during project; Nimbus prototype release. ● # demos delivered: 2 (TaRDIS Messaging App multi-device demo; 5,000-node large-scale experiment). ● # external stakeholders engaged: 5+ (Telefónica, GMV, EDP, Actyx, ParadigmShift), plus additional contacts through the Horizon Results Booster programme. ● Repo metrics (stars/forks/downloads/issues): Public GitHub repository at https://github.com/babel-project; metrics to be tracked post open-source release (v1.0). ● # pilots initiated: 1 (ParadigmShift early-adopter commercial pilot, building an IoT product on Babel/Micro-Babel). ● Publications/citations (optional): 5+ peer-reviewed papers at ACM/IEEE venues on Babel protocols and decentralised IoT middleware; 1 IETF Internet Standards Draft. Full list in TaRDIS dissemination report and D6.3 [8].
Expected metrics (next 12-24 months)
<ul style="list-style-type: none"> ● Target releases: 2 stable open-source releases within 12 months (Babel Ecosystem v1.0; Nimbus v1.0); quarterly patch releases thereafter. ● Target pilots: 3 active pilots within 24 months (ParadigmShift + 2 additional IoT companies); target revenue of €40K year 1, growing to €300K by year 3. ● Target adopters/users: 10+ developer teams using Babel in production or active evaluation within 24 months; 50+ GitHub stars within 12 months of v1.0 release. ● Target training/outreach: 2 workshops/tutorials at major IoT or distributed systems conferences within 12 months; developer webinar series; engagement in IETF standardisation process.
Quantified improvements (if available)
If you used D7.1 quantification procedures or other metrics, summarise:

- **Metric(s):** Message delivery reliability in large-scale decentralised IoT networks (% of messages successfully delivered end-to-end).
- **Baseline:** ad hoc MQTT-based IoT deployments in intermittently connected networks typically achieve around 80-85% delivery reliability under representative disconnection patterns.
- **Result:** 99.49% message delivery reliability at 5,000 nodes, with approximately 1.2 seconds average end-to-end latency and 48+ hours of continuous battery operation on commodity Android devices (D6.3 [8] Section 6).
- **Evidence link:** D6.3 [8], Section 6; <https://github.com/babel-project> (reproducibility package).

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Key-person dependency (High): development of the Babel Ecosystem is concentrated in a small NOVA research team. The departure of key researchers after the project could stall maintenance and community growth. 2. Market adoption barrier (Medium): IoT developers accustomed to cloud-centric, broker-based frameworks may face a significant paradigm shift when adopting decentralised programming models, leading to slower-than-expected uptake. 3. Funding gap (Medium): post-project maintenance costs (approximately €250K/year) must be covered by commercial revenue or follow-on funding before project financial reserves are exhausted; delays in commercialisation could create a sustainability gap.
Mitigation actions
<ol style="list-style-type: none"> 1. Key-person dependency (High): Hire or retain post-doctoral researchers after the project; form a spin-off entity to provide economic continuity and stable employment for the core team; and grow an open-source community to distribute maintenance responsibility beyond the NOVA team. 2. Market adoption barrier (Medium): Invest in developer outreach (workshops, video tutorials, comprehensive documentation); leverage ParadigmShift as a public success story; and target early-adopter IoT companies with strong pain points related to cloud dependency as lighthouse customers. 3. Funding gap (Medium): Pursue Horizon Europe follow-on projects and EIC Transition/Accelerator funding to bridge the post-project gap; operationalise the commercial support and integration service offering from Q2 2026 to generate early revenue (target: €40K in year 1).
Support needed from consortium / Horizon Results Booster / EC ecosystem
<p>From consortium: Introductions to IoT pilot partners via use case leads (Telefónica, GMV, EDP, Actyx) for potential pilot agreements; joint dissemination and co-marketing of the Babel Ecosystem at TaRDIS events and publications.</p> <p>From Horizon Results Booster: Continued advisory support for go-to-market strategy refinement, investor introductions, and partnership development beyond the project lifetime.</p> <p>From EC ecosystem: Visibility in EC IoT and Edge Computing initiatives; access to EIC Transition and EIC Accelerator funding instruments to support spin-off formation and scale-up.</p>

Partner Statement

NOVA LINCS, as leader of Work Package 6 in TaRDIS, has developed the Babel Ecosystem - a unified, protocol-centric programming platform that enables the seamless development of decentralised IoT applications across heterogeneous device tiers, from cloud servers and Android smartphones to bare-metal embedded hardware. The Babel Ecosystem (comprising Babel-Swarm, Babel-Android, and Micro-Babel) represents the primary Key Exploitable Result from NOVA's participation in TaRDIS and has been validated in a 5,000-node experiment that achieved 99.49% message delivery reliability while operating continuously for more than 48 hours on commodity mobile devices. ParadigmShift, an external company, has already adopted Babel and Micro-Babel as the foundation for a commercial IoT product, providing early validation of the technology's market readiness at TRL 6. NOVA's exploitation strategy, developed through the Horizon Results Booster programme, centres on an open-source core release under Apache 2.0, a dual-licensing commercial model for advanced modules, support and integration services, and the creation of a spin-off or licensing entity to ensure post-project

sustainability. Revenue projections reach €300K by year 3 after project completion. In parallel, NOVA continues its active IETF engagement to support the standardisation of Babel-based protocols. NOVA is committed to ensuring that the TaRDIS investment delivers lasting economic and societal impact through open innovation, commercial exploitation, and the democratisation of resilient IoT infrastructure for European industry.

6.2 PARTNER: TECHNICAL UNIVERSITY OF DENMARK (DTU)

Organisation name	Technical University of Denmark		Short	DTU
Contact person for exploitation	Name	Alceste Scalas	Partner Type	University
	Email	alcsc@dtu.dk		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input type="checkbox"/> EDP <input checked="" type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Composing large swarms for factory automation

KER Title	Composing large swarms for factory automation			
Short Description				
DTU developed theoretical methods and practical tools to correctly reuse and compose swarm components. In the setting of the ACT use case, this enables the modular design and verification of complex factory automation protocols.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input checked="" type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	3-4	
Validation evidence	https://github.com/DTU-SSE/machines/tree/master/demos/warehouse-factory			
Pain Point (i.e., what problem does it solve?)				
Previous methods and tools for defining and implementing swarms for factory automation were not compositional. Therefore, reusing existing swarm elements and automation specifications was error-prone and required significant manual effort.				
Value proposition (i.e., why it is better than the alternatives?)				
To the best of our knowledge, there are no comparable alternatives.				
Target users / customers				
<ul style="list-style-type: none"> • Primary adopters: Users of the Actyx toolkit - https://github.com/Actyx/Actyx • Secondary adopters: Design & development teams responsible for factory automation • Sectors/markets: Factory automation, industrial production 				
Dependencies and integration requirements				
<ul style="list-style-type: none"> • TaRDIS components needed: Actyx toolkit • Third-party dependencies: None • Hardware/network assumptions: IP-based network; sufficient hardware specs for running a minimal Linux distribution 				

<ul style="list-style-type: none"> • Skills required to adopt: TypeScript programming 					
Unique technical features					
<ul style="list-style-type: none"> • Tools for automatically verify the correct implementation of swarm components • Tools for automatically adapt and compose swarm components • Based on formal mathematical results 					
Evidence and artefacts					
Code repository (public/private):					
<ul style="list-style-type: none"> • https://github.com/DTU-SSE/machines • https://github.com/DTU-SSE/machines/tree/master/demos/warehouse-factory • https://github.com/DTU-SSE/swarm-monitor/blob/main/DEMO.md • https://github.com/DTU-SSE/latency-experiment 					
Exploitation Plan (Next 12-24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input checked="" type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input checked="" type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Release as open-source software	DTU, ACT	2026-03-31		<input type="checkbox"/> planned <input type="checkbox"/> ongoing <input checked="" type="checkbox"/> done
M2	Updated documentation and demo artifact	DTU	2026-05-31		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): https://github.com/DTU-SSE/machines • Pilot deliverables: N/A • Business outputs (offers, proposals): N/A 					

IP, licensing and access rights

Background IP needed					
<ul style="list-style-type: none"> • Your background needed: None. • Third-party IP/licenses: None. 					
Foreground ownership					
<input type="checkbox"/> Single partner-owned <input checked="" type="checkbox"/> Joint ownership - partners: ACT, DTU <input type="checkbox"/> Unsure / to be agreed					
Licensing intentions (software / docs)					
<ul style="list-style-type: none"> • Preferred licence (if open source): Apache 2.0 and MIT • Components intended to remain proprietary: None • Constraints/notes (patents, NDAs, data restrictions): None 					

Sustainability and maintenance (post-project)

What will you maintain after project end?
--

<ul style="list-style-type: none"> ● Components: https://github.com/DTU-SSE/machines ● Documentation: Included in artifact to be publicly released by 31 May 2026 (see milestone 2 above) ● Demo/benchmark artefacts: <ul style="list-style-type: none"> ○ https://github.com/DTU-SSE/machines/tree/master/demos/warehouse-factory ○ https://github.com/DTU-SSE/swarm-monitor/blob/main/DEMO.md ○ https://github.com/DTU-SSE/latency-experiment 	
Maintenance commitment	Governance and hosting preference
<input checked="" type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input type="checkbox"/> TaRDIS consortium repository (shared) <input type="checkbox"/> Partner-owned repository <input checked="" type="checkbox"/> External foundation/community (specify): GitHub

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # software releases: 1 ● # demos delivered: 3 ● # external stakeholders engaged: 1 ● Repo metrics (stars/forks/downloads/issues): N/A (repository published recently) ● # pilots initiated: 0 ● Publications/citations (optional):
Expected metrics (next 12-24 months)
<ul style="list-style-type: none"> ● Target releases: 1 ● Target pilots: N/A (best effort) ● Target adopters/users: N/A (best effort) ● Target training/outreach: N/A (best effort)
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> ● Metric(s): N/A ● Baseline: N/A ● Result: N/A ● Evidence link: N/A

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Reduced uptake if the target industrial ecosystem/tooling used for integration (e.g., middleware stack) evolves or becomes less accessible, requiring additional maintenance effort/alternative integration targets.
Mitigation actions
<ol style="list-style-type: none"> 1. Best-effort advertisement 1. Possible adoption as teaching material for DTU courses on concurrent and distributed systems.
Support needed from consortium / Horizon Results Booster / EC ecosystem
None

Partner Statement

The TaRDIS results have already impacted the DTU teaching for the MSc in Computer Science: DTU has already successfully completed 4 MSc projects based on TaRDIS research. The TaRDIS results will also influence future teaching activities - especially for courses related

to programming, formal methods, and concurrent and distributed systems. DTU plans to foster the industrial adoption of the TaRDIS toolkit through new MSc projects and collaborations with its industrial partners.

6.3 PARTNER: UNIVERSITY OF OXFORD (UOX)

Organisation name	University of Oxford		Short	UOX
Contact person for exploitation	Name	Nobuko Yoshida	Partner Type	University
	Email	nobuko.yoshida@cs.ox.ac.uk		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input type="checkbox"/> EDP <input type="checkbox"/> Actyx <input checked="" type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Type-Based Methods and framework for Communication Behaviour Analysis
- KER-2: Scribble (NuScr) - Protocol Specification and Analysis Tool

KER Title	Type-Based Methods and framework for Communication Behaviour Analysis			
Short Description				
This result comprises type-based methods and a framework, grounded in multiparty session types, for analysing communication behaviour in distributed systems. It supports formal specification and static analysis of communication protocols, enabling reasoning about properties such as communication safety, deadlock freedom, and liveness. The framework extends existing approaches to handle dynamic participation (e.g., late joining participants) and failure-prone execution and includes mechanised foundations to ensure the correctness of the analysis methods.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: Type-based communication behaviour analysis methods and framework			
Current maturity	<input checked="" type="checkbox"/> research concept <input type="checkbox"/> demo-ready	<input checked="" type="checkbox"/> prototype <input type="checkbox"/> pilot-ready	<input checked="" type="checkbox"/> validated in lab	TRL estimate
Validation evidence	Validation is demonstrated through formalisation of the methods, their implementation in supporting tools, and evaluation in academic publications and case studies involving communication protocol analysis in distributed systems			
Pain Point (i.e., what problem does it solve?)				
This framework addresses the challenge of ensuring correct communication in distributed systems, where components interact through complex protocols and operate without a complete global view. Errors such as inconsistent message sequences, deadlocks, and protocol violations are difficult to detect using conventional approaches. The framework provides a type-based foundation for specifying and analysing communication behaviour, enabling early detection of such issues and improving system reliability.				
Value proposition (i.e., why it is better than the alternatives?)				
Compared to informal specifications and testing-based approaches, the framework enables systematic and early detection of communication errors through type-based analysis. It provides formal guarantees on key behavioural properties and ensures consistency between interacting components by deriving local behaviours from global protocol specifications. This reduces development effort and improves reliability in distributed systems.				
Target users / customers				
<ul style="list-style-type: none"> • Primary adopters: <ul style="list-style-type: none"> ○ Researchers and practitioners in formal methods and distributed systems ○ Developers designing communication protocols and distributed applications 				

- **Secondary adopters:**
 - Organisations developing cloud, edge, and microservices-based systems
 - Tool developers integrating formal analysis into development workflows
 - Academic users for teaching and research
- **Sectors/markets:**
 - Distributed systems and cloud computing
 - Software engineering and developer tools
 - Telecommunications and networked systems
 - Edge and IoT systems

Dependencies and integration requirements

- **TaRDIS components needed:** Alignment with TaRDIS communication models and abstractions
- **Third-party dependencies:**
 - Supporting tools implementing type-based analysis (e.g., Scribble/NuScr)
 - Standard development environments
- **Hardware/network assumptions:** No specific hardware requirements; applicable to distributed and networked environments
- **Skills required to adopt:**
 - Knowledge of distributed systems and communication protocols
 - Familiarity with formal methods or type-based approaches

Unique technical features

- Type-based protocol specification: Formal modelling of communication behaviour using multiparty session types
- Global-to-local consistency: Derivation of local endpoint behaviours from global protocol descriptions
- Static verification of communication properties: Supports checking of safety, deadlock freedom, and liveness
- Support for dynamic systems: Applicable to environments with evolving participants and partial system knowledge
- Integration with tool-supported workflows: Can be combined with tools such as Scribble for practical analysis

Evidence and artefacts

Public deliverable(s):

- D4.1 - Report on the Desirable Properties for Analysis (M6)
- D4.2 - Report on the Initial Toolset (M18)
- D4.3 - Final Report on Toolset (M34)

Code repository (public/private):

- Supported via tools such as Scribble (NuScr): <https://github.com/nuscr/nuscr>

Demo video/slide deck:

- Scribble (NuScr) web interface (interactive): <https://nuscr.dev/nuscr/>

Benchmark/evaluation:

- Evaluation demonstrated through formal analysis, tool-supported validation, and case studies in distributed communication protocols

Exploitation Plan (Next 12-24 Months)

Exploitation route(s)	<input type="checkbox"/> Open-source release	<input type="checkbox"/> Services (integration, support, training)
	<input checked="" type="checkbox"/> Integration into partner product/service	<input checked="" type="checkbox"/> Research exploitation (follow-on projects)
	<input checked="" type="checkbox"/> Joint exploitation with other partners	<input checked="" type="checkbox"/> Standardisation / pre-standardisation influence
	<input type="checkbox"/> External pilots / proofs of concept	<input type="checkbox"/> Other:

Concrete actions and milestones

Milestone	Description	Owner	Target Date	Dependencies	Status
-----------	-------------	-------	-------------	--------------	--------

M1 Advancement of type-based methods for communication correctness	Further develop type-based methods based on multiparty session types to ensure communication correctness properties, including communication safety, deadlock freedom, and liveness, in distributed systems.	UOX	Month 12	Ongoing research activities	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M2 Extension to dynamic, compositional, and advanced analysis techniques	Extend the framework to support additional features for analysing communication behaviour in distributed systems, and investigate complementary techniques such as compositional verification, model checking, and mechanised verification to enhance analysis capabilities.	UOX	Month 18	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3 Exploration of advanced methods and dissemination	Explore the integration of emerging techniques, such as large language models for protocol analysis support, and disseminate results through publications and follow-on research, promoting adoption of type-based approaches in distributed systems.	UOX	Month 24	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done

Expected outputs (what will be delivered)

- **Releases (software/docs):** Academic publications reporting the developed methods and results
- **Pilot deliverables:**
 - Validation through case studies and protocol analyses in TaRDIS use cases
 - Demonstrations of advanced features
- **Business outputs (offers, proposals):**
 - Follow-on research proposals and collaborations
 - Contributions to academic dissemination and potential influence on standardisation efforts

KER Title	Scribble (NuScr) - Protocol Specification and Analysis Tool				
Short Description					
Scribble (NuScr) is a protocol description and analysis tool based on type-based techniques (multiparty session types). It enables the specification of global communication protocols, their projection to endpoint behaviours, and supports static checking of communication correctness. The tool helps detect communication mismatches early and supports the development of reliable distributed applications.					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Improved Actyx Middleware		<input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training		
	<input checked="" type="checkbox"/> Not applicable / other: Communication behaviour analysis and protocol verification tool				
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> demo-ready	<input checked="" type="checkbox"/> prototype <input type="checkbox"/> pilot-ready	<input checked="" type="checkbox"/> validated in lab	TRL estimate	
Validation evidence	Validation is demonstrated through the implementation of the Scribble (NuScr) tool, its publicly available repository and web interface, and its use in academic publications and case studies for specifying and analysing communication protocols based on multiparty session types.				
Pain Point (i.e., what problem does it solve?)					
Scribble solves the problem of ensuring correct communication between distributed system components by providing a formal, type-based language for specifying and analysing communication protocols. It enables the definition of global interaction protocols and their projection to local behaviours, allowing early detection of communication mismatches and helping ensure that implementations follow consistent interaction patterns. This					

reduces errors such as deadlocks, incorrect message ordering, and inconsistent assumptions between components.

Value proposition (i.e., why it is better than the alternatives?)

Scribble provides a type-based, formal approach to specifying and analysing communication protocols, enabling correctness guarantees that are difficult to achieve with conventional testing and debugging methods. Unlike informal or ad hoc approaches, it supports early detection of communication errors at the design stage, reducing development time and improving reliability. Its ability to derive local behaviours from a global protocol ensures consistency across distributed components, which is typically challenging to achieve manually.

Target users / customers

- **Primary adopters:**
 - Software developers and engineers building distributed and concurrent systems
 - Researchers and practitioners working on formal methods and communication protocols
 - Developers of middleware, APIs, and distributed services
- **Secondary adopters:**
 - Organisations developing cloud, edge, and microservices-based applications
 - Tool developers integrating protocol verification into development environments
 - Academic users for teaching and research in distributed systems and formal methods
- **Sectors/markets:**
 - Distributed systems and cloud computing
 - Software engineering and developer tools
 - Telecommunications and networked systems
 - Edge and IoT systems

Dependencies and integration requirements

- **TaRDIS components needed:**
 - Integration with TaRDIS communication models and APIs for applying protocol specifications within the TaRDIS ecosystem
 - Optional use within the TaRDIS development environments for workflow integration
- **Third-party dependencies:**
 - Supporting tools implementing type-based analysis (e.g., Scribble/NuScr)
 - Standard development environments
- **Hardware/network assumptions:**
 - No specific hardware requirements; operates in standard development environments
 - Applicable to systems deployed across distributed, cloud, edge, or networked environments
- **Skills required to adopt:**
 - Basic knowledge of distributed systems and communication protocols
 - Familiarity with formal specification concepts (e.g., type-based approaches)
 - Software development experience for integrating generated artefacts into applications

Unique technical features

- **Global protocol specification:** Supports high-level description of multiparty communication protocols from a global perspective.
- **Projection to local behaviours:** Automatically derives role-specific endpoint behaviours from global protocols, ensuring consistency across participants.
- **Type-based verification:** Uses multiparty session types to enable static checking of communication correctness properties.
- **Endpoint finite state machine generation:** Produces endpoint FSMs that can be used for analysis and correct-by-construction code generation.
- **Toolchain and web interface:** Provides both a command-line tool and a web-based interface (NuScr) for rapid prototyping and analysis.

Evidence and artefacts

Public deliverable(s):

<ul style="list-style-type: none"> • D4.2 - Report on the Initial Toolset (M18) • D4.3 - Final Report on Toolset (M34) <p>Code repository (public/private):</p> <ul style="list-style-type: none"> • Scribble (NuScr) repository (public): https://github.com/nuscr/nuscr <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> • Scribble (NuScr) web interface (interactive demo): https://nuscr.dev/nuscr/ <p>Benchmark/evaluation:</p> <ul style="list-style-type: none"> • Evaluation demonstrated through use in research case studies and experimental validation of communication protocols based on multiparty session types 					
Exploitation Plan (Next 12-24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input checked="" type="checkbox"/> Joint exploitation with other partners <input checked="" type="checkbox"/> External pilots / proofs of concept		<input checked="" type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input checked="" type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1 Enhancement of Scribble (NuScr) toolchain	Enhance the Scribble (NuScr) toolchain with improved support for advanced communication protocol features, including better usability, extended language constructs, and support for analysing more complex interaction patterns.	UOX	Month 12	Ongoing research results and tool development	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M2 Extension and application of Scribble in TaRDIS use cases	Extend Scribble to support additional protocol features (e.g., dynamic participation and advanced interaction patterns) and promote its use in TaRDIS use cases and related distributed applications.	UOX (with WP partners)	Month 18	Use cases (WP6/WP7), ongoing tool development	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M3 Dissemination and community adoption of Scribble (NuScr)	Promote Scribble (NuScr) through publications, tutorials, workshops, and online resources, and encourage its adoption by researchers and developers working on distributed systems and communication protocols.	UOX	Month 24	Research outputs, community engagement, and dissemination activities	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): <ul style="list-style-type: none"> ○ Updated releases of the Scribble (NuScr) toolchain, including extended support for advanced protocol features ○ Documentation, tutorials, and examples for protocol specification and analysis • Pilot deliverables: <ul style="list-style-type: none"> ○ Application of Scribble in selected TaRDIS use cases for specifying and analysing communication protocols ○ Demonstrations through case studies and integration into development workflows • Business outputs (offers, proposals): <ul style="list-style-type: none"> ○ Research proposals and collaborations for further development and adoption of Scribble ○ Training materials and workshops for developers and researchers 					

IP, licensing and access rights

Background IP needed
<ul style="list-style-type: none"> • Your background needed: Existing research on type theory and prior Scribble developments • Third-party IP/licenses: Open-source dependencies used by the Scribble (NuScr) toolchain.
Foreground ownership
<input checked="" type="checkbox"/> Single partner-owned <input type="checkbox"/> Joint ownership - partners: <input type="checkbox"/> Unsure / to be agreed
Licensing intentions (software / docs)
<ul style="list-style-type: none"> • Preferred licence (if open source): Open-source licence (e.g., MIT / BSD-style licence - consistent with existing Scribble repository) • Components intended to remain proprietary: None • Constraints/notes (patents, NDAs, data restrictions): No known IP restrictions; released as open-source software for research and development use

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> • Components: Scribble (NuScr) toolchain and associated extensions for protocol specification and analysis • Documentation: User documentation, tutorials, and examples for using Scribble • Demo/benchmark artefacts: <ul style="list-style-type: none"> ○ Web interface (NuScr) for prototyping ○ Example protocols and case studies 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input checked="" type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input checked="" type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> • # software releases: Multiple releases of the Scribble (NuScr) toolchain (ongoing development) • # demos delivered: <ul style="list-style-type: none"> ○ Demonstrations through case studies and formal analysis examples ○ Web-based interface (NuScr) available for interactive use ○ Demonstrations through research case studies and TaRDIS activities • # external stakeholders engaged: Academic research community in formal methods and distributed systems • Repo metrics (stars/forks/downloads/issues): Public repository available: https://github.com/nuscr/nuscr • # pilots initiated: Validation through TaRDIS use cases and protocol analyses
Expected metrics (next 12-24 months)
<ul style="list-style-type: none"> • Target releases: Continued integration of methods into supporting tools (e.g., Scribble), 1-2 improved releases of the Scribble (NuScr) toolchain, including enhancements to protocol features and usability • Target pilots: Application in multiple use cases and related distributed systems • Target adopters/users: Increased uptake in research and formal methods communities • Target training/outreach: Publications, workshops, and academic dissemination

Risks, Blockers, and Support Needed

Top risks
<ol style="list-style-type: none"> 1. Complexity of formal methods limiting adoption by non-expert users. 2. Challenges in scaling analysis to large and heterogeneous distributed systems. 3. Dependence on tool support for practical adoption. 4. Limited adoption beyond the research community. 5. Complexity of integrating formal methods into existing development workflow 6. Resource constraints for long-term tool maintenance and evolution
Mitigation actions
<ol style="list-style-type: none"> 1. Simplify usage through tool-supported workflows (e.g., Scribble). 1. Validate methods through case studies and real-world scenarios 2. Investigate complementary techniques (e.g., compositional analysis, model checking). 3. Disseminate results through publications and training activities. 4. Provide documentation, tutorials, and examples to lower the adoption barrier 4. Integrate Scribble into developer tools and workflows (APIs/IDE) 5. Promote usage through publications, workshops, and community engagement 6. Encourage open-source contributions to support sustainability
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> • Support for integration with tools and use cases across WP4-WP7 • Dissemination and visibility within research and industrial communities • Opportunities for follow-on research and collaboration • Support for dissemination and outreach activities (workshops, training, visibility) • Collaboration with partners for integration into use cases and toolchains • Guidance on exploitation and standardisation opportunities

Partner Statement

UOX will leverage the results of TaRDIS to advance the state of the art in formal methods for distributed systems, particularly in the analysis of communication behaviour using type-based techniques. The project has enabled the development and validation of methods and tools, such as Scribble, that support correct-by-construction communication protocols. UOX will continue to develop and extend these approaches through follow-on research, collaborations, and integration into future projects. The results will be disseminated through high-impact publications, open-source tooling, and engagement with the research and developer communities. In addition, UOX will promote the adoption of these techniques in practical distributed systems, contributing to improved reliability and correctness in emerging cloud, edge, and decentralised environments.

6.4 PARTNER: UNIVERSITY OF NOVI SAD (UNS)

Organisation name	University of Novi Sad		Short	UNS
Contact person for exploitation	Name	Ivan Kaštelan	Partner Type	University
	Email	ivan.kastelan@uns.ac.rs		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input checked="" type="checkbox"/> GMV <input type="checkbox"/> EDP <input checked="" type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Decentralised Monitorization and Reconfiguration Management tool
- KER-2: PTB-FLA and MPT-FLA FL frameworks
- KER-3: PTB-FLA–Babel Adapter / Stack
- KER-4: Flower-based FL tool

KER Title	Decentralised Monitorization and Reconfiguration Management tool			
Short Description				
The TaRDIS project has developed and demonstrated a powerful prototype for decentralised monitoring and configuration management in swarm-based systems. This milestone showcases an innovative system architecture, novel APIs, and runtime configuration strategies, all supporting the dynamic and intelligent orchestration of distributed edge applications.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem	<input type="checkbox"/> Decentralised Data Management Framework		
	<input type="checkbox"/> Improved Actyx Middleware	<input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training		
	<input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept	<input checked="" type="checkbox"/> prototype	<input type="checkbox"/> validated in lab	TRL estimate
	<input type="checkbox"/> demo-ready	<input type="checkbox"/> pilot-ready		

KER Title	PTB-FLA and MPT-FLA FL frameworks			
Short Description				
PTB-FLA and MPT-FLA are lightweight federated learning frameworks for federated learning and distributed peer-data exchange-based algorithm development and experimentation in edge-oriented environments. PTB-FLA provides a simple Python-based framework for developing and testing FL algorithms, while MPT-FLA extends this approach toward distributed execution on networked edge and IoT devices. Together, they support educational use, research prototyping, and the development of decentralised AI solutions.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem	<input type="checkbox"/> Decentralised Data Management Framework		
	<input type="checkbox"/> Improved Actyx Middleware	<input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training		
	<input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept	<input checked="" type="checkbox"/> prototype	<input checked="" type="checkbox"/> validated in lab	TRL estimate
	<input checked="" type="checkbox"/> demo-ready	<input type="checkbox"/> pilot-ready		4-5
Validation evidence	Validated through experimental use in FL algorithm development, academic publications, and GMV use case. MPT-FLA has also been validated in networked edge/IoT scenarios.			
Pain Point (i.e., what problem does it solve?)				
Existing federated learning frameworks are often too heavyweight, cloud-oriented, or difficult to adapt for resource-constrained edge and IoT environments. PTB-FLA and MPT-FLA address this by providing lightweight, accessible frameworks for designing, testing, and running FL algorithms in decentralised and edge-oriented settings.				
Value proposition (i.e., why it is better than the alternatives?)				
The frameworks are lightweight, easy to understand, and suitable for rapid prototyping, teaching, and research. They reduce setup complexity and make FL algorithm development more accessible, especially for educational and experimental use cases. MPT-FLA further extends this value by supporting execution on constrained and networked devices.				
Target users / customers				
<ul style="list-style-type: none"> ● Primary adopters: Universities, research groups, AI/ML developers 				

<ul style="list-style-type: none"> ● Secondary adopters: Industrial R&D teams, MSc/PhD students ● Sectors/markets: Higher education, research, edge AI, IoT, distributed intelligent systems
Dependencies and integration requirements
<ul style="list-style-type: none"> ● TaRDIS components needed: None except PTB-FLA/FLA ● Third-party dependencies: Python/Micropython ● Hardware/network assumptions: PTB-FLA runs on standard PCs; MPT-FLA targets networked PCs and IoT/edge devices ● Skills required to adopt: Basic Python programming, basic machine learning knowledge, basic understanding of federated learning and networking
Unique technical features
<ul style="list-style-type: none"> ● Lightweight federated learning framework family for edge-oriented development ● PTB-FLA supports simple Python-based FL algorithm prototyping ● MPT-FLA extends the approach to MicroPython and distributed edge/IoT execution ● Allows development of distributed applications based on TDM peer data exchange ● Low setup complexity ● Well suited for teaching, prototyping, and research
Evidence and artefacts
<p>Public Deliverables: D3.1, D3.2, D3.3, D3.4, D3.5, D4.1, D4.2, D4.3, D5.1, D5.2, D5.3, D7.2, D7.3</p> <p>Code repository (public/private):</p> <ul style="list-style-type: none"> ● https://github.com/miroslav-popovic/ptbfla ● https://codelab.fct.unl.pt/di/research/TaRDIS/wp5/ptb-fla <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> ● https://project-TaRDIS.eu/blog/2024/06/25/the-ptb-fla-successor-mpt-fla-advances-to-edge-systems/

KER Title	PTB-FLA–Babel Adapter / Stack				
Short Description					
The PTB-FLA–Babel Adapter / Stack enables integration of the PTB-FLA framework family with the Babel communication and middleware ecosystem within TaRDIS. It provides a bridge between lightweight FL algorithm development and broader decentralised execution environments, supporting interoperability and more realistic distributed experimentation.					
Innovation Radar alignment	<input checked="" type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:				
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	4		
Validation evidence	Validated through adapter development and demonstrations showing integration of PTB-FLA with the Babel-based stack as well as the GMV Use case.				
Pain Point (i.e., what problem does it solve?)					
A standalone FL framework is useful for algorithm development, but exploitation in TaRDIS requires integration with the broader communication and execution ecosystem. The adapter solves this interoperability gap by connecting PTB-FLA-based FL development with Babel-enabled distributed environments.					
Value proposition (i.e., why it is better than the alternatives?)					
The adapter extends the usefulness of PTB-FLA beyond standalone prototyping by enabling integration with the TaRDIS ecosystem without requiring changes to the already implemented decentralised applications. This creates a practical path from lightweight algorithm development to distributed deployment and experimentation, without requiring a complete redesign of the application.					

Target users / customers
<ul style="list-style-type: none"> ● Primary adopters: Existing PTB-FLA users, TaRDIS partners, research groups working with Babel ● Secondary adopters: Industrial R&D teams exploring FL middleware integration ● Sectors/markets: Researchers, distributed simulations, large scale benchmarking of algorithms.
Dependencies and integration requirements
<ul style="list-style-type: none"> ● TaRDIS components needed: PTB-FLA, Babel ● Third-party dependencies: Java, Python, Maven ● Hardware/network assumptions: Distributed/networked execution environment compatible ● Skills required to adopt: Python programming, basic understanding of distributed systems, networking, middleware integration knowledge
Unique technical features
<ul style="list-style-type: none"> ● Connects PTB-FLA with the Babel ecosystem ● Bridges lightweight FL development and distributed deployment ● Reuses PTB-FLA-based application logic without major redesign ● Strengthens the exploitation path of UNS FL frameworks inside TaRDIS
Evidence and artefacts
<p>Public deliverable(s): D5.3, D7.4</p> <p>Code repository (public/private):</p> <ul style="list-style-type: none"> ● https://github.com/LinguineP/PTB-FLA_BabelAdapter ● https://codelab.fct.unl.pt/di/research/TaRDIS/toolkit/Documentation/-/wikis/TaRDIS-Tools-and-APIs/Managed-Swarm-Elements/PTB-FLA-and-MPT-FLA ● https://codelab.fct.unl.pt/di/research/TaRDIS/wp6/demonstrations/gmv/adapters/ptb-fla <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> ● https://project-TaRDIS.eu/blog/2024/11/13/ptb-fla-babel-stack-gets-rolled-out-to-the-public-ptb-fla-got-the-plug-in-adapter-for-babel/

KER Title	Flower-based FL tool				
Short Description					
The Flower-based federated learning (FL) tool represents an approach for model training, data preparation, and evaluation and inference. It provides a flexible, user-centric interface and supports multiple FL algorithms (e.g., FedAvg, pFedMe) and model architectures (autoencoders, transformers, and CNNs). It was applied to tasks such as anomaly detection with noisy labels and classification and evaluated on a synthetic dataset provided by Actyx.					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Not applicable / other:	<input type="checkbox"/> Decentralised Data Management Framework <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training			
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> demo-ready	<input checked="" type="checkbox"/> prototype <input type="checkbox"/> pilot-ready	<input checked="" type="checkbox"/> validated in lab <table border="1"> <tr> <td>TRL estimate</td> <td>4-5</td> </tr> </table>	TRL estimate	4-5
TRL estimate	4-5				
Validation evidence	Validated through experiments regarding FL model training. Also tested on a synthetic data set, provided by Actyx.				
Pain Point (i.e., what problem does it solve?)					
It enables FL model training setup for non-expert users, by an intuitive interface, that provides applicable choices for the tasks needed to be solved. Also, it provides a straightforward approach for performing the model training.					
Value proposition (i.e., why it is better than the alternatives?)					
The tool is user-centric, i.e., easy to use and extend. It is applicable to experiment with various tasks on different data sets, in order to obtain valuable insights.					

Target users / customers					
<ul style="list-style-type: none"> ● Primary adopters: Universities, research groups, AI/ML developers ● Secondary adopters: Industry practitioners or private sector organizations ● Sectors/markets: Higher education, research, AI/ML, data-driven industries 					
Dependencies and integration requirements					
<ul style="list-style-type: none"> ● TaRDIS components needed: None, it can work as a standalone tool ● Third-party dependencies: Python, the Flower framework, and common ML libraries (e.g., PyTorch, TensorFlow, NumPy). ● Hardware/network assumptions: The tool runs on standard PCs, on cluster environments, and other federated settings ● Skills required to adopt: Basic understanding of the FL ML task to be solved, basic Python knowledge. 					
Unique technical features					
<ul style="list-style-type: none"> ● Flexible, user-centric interface for configuring federated learning workflows ● Support for multiple federated learning algorithms (e.g., FedAvg, pFedMe) ● Compatibility with diverse model architectures, including autoencoders, transformers, and CNNs ● Integrated pipeline for data preparation, training, and evaluation ● Applicability to varied ML tasks, such as anomaly detection and classification ● Modular and extensible design based on the Flower framework 					
Evidence and artefacts					
<p>Public deliverable(s):</p> <ul style="list-style-type: none"> ● D5.1: Initial report on distributed AI and AI-based orchestration, D5.2: Second report on distributed AI and AI-based orchestration, D5.3: Final report on distributed AI and AI-based orchestration ● D2.3, D3.1, D3.2, D3.3, D3.4, D3.5, D3.6, D7.2, D7.3, D7.4 <p>Code repository (public/private):</p> <ul style="list-style-type: none"> ● https://github.com/lidijaf/Flower-based_FL_tool <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> ● https://project-TaRDIS.eu/news/2025/06/17/new-federated-learning-flower-framework-based-tool-demonstration-showcases-interactive-tool-and-advanced-algorithms/ 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release		<input checked="" type="checkbox"/> Services (integration, support, training)		
	<input type="checkbox"/> Integration into partner product/service		<input checked="" type="checkbox"/> Research exploitation (follow-on projects)		
	<input checked="" type="checkbox"/> Joint exploitation with other partners		<input type="checkbox"/> Standardisation / pre-standardisation influence		
	<input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Finalize and maintain PTB-FLA, MPT-FLA and PTB-FLA–Babel Adapter repositories and supporting documentation for public/research use	UNS	2026		<input type="checkbox"/> planned <input type="checkbox"/> ongoing <input checked="" type="checkbox"/> done
M2	Use PTB-FLA and MPT-FLA results in teaching activities, student projects, and graduate-level courses	UNS	2026		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done

M3	Prepare follow-on research proposals and collaboration opportunities with TaRDIS partners based on PTB-FLA / MPT-FLA, Flower-based FL tool and related KERs	UNS	2026-2027		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M4	Maintain and incrementally improve the Flower-based FL tool repository and documentation for research use	UNS	2026		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M5	Support extensions of the Flower-based FL tool (e.g., additional FL algorithms or model configurations) and additional validation on various data, as needed	UNS	2026		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): Updated repositories, documentation, examples • Pilot deliverables: Internal demos, research prototypes • Business outputs (offers, proposals): Research proposals, project applications 					

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> • Components: PTB-FLA, MPT-FLA, PTB-FLA–Babel Adapter repositories and selected examples, Flower-based FL tool • Documentation: Installation instructions, usage notes, teaching materials, technical documentation • Demo/benchmark artefacts: Selected demos, evaluation examples, and benchmark artefacts where feasible 	
Maintenance commitment	Governance and hosting preference
<input checked="" type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> • # software releases: 0 (No formal releases) • # demos delivered: 0 • # external stakeholders engaged: • Repo metrics (stars/forks/downloads/issues): • # pilots initiated: 0 • Publications/citations (optional):
Expected metrics (next 12–24 months)
<ul style="list-style-type: none"> • Target releases: 1–2 repository/documentation updates

- **Target pilots:** 1–2 research or collaboration demonstrations
- **Target adopters/users:** Students, researchers, and interested project partners
- **Target training/outreach:** Integration into courses, student projects, workshops, and research dissemination

Partner Statement

As an educational institution, UNS will benefit from TaRDIS results primarily through educational and research activities. The know-how and the research results from TaRDIS KERs will be used to improve the courses, teaching material and teaching methodology to modernize the curriculum and offer state-of-the-art skills to students, mostly at the upper undergraduate and graduate level. Developed frameworks, Flower and PTB-FLA, will also enable strengthened research opportunities, enabling the UNS team to develop new project ideas and apply for new grants in the future. Optionally, UNS may collaborate with other partners within the use case to commercialize the use case KERs, following the agreed IP management plan

6.5 PARTNER: NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS (NKUA)

Organisation name	National and Kapodistrian University of Athens		Short	NKUA
Contact person for exploitation	Name	Sotirios Spantideas	Partner Type	University
	Email	sospanti@gmail.com		
TaRDIS use case(s)	<input checked="" type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input checked="" type="checkbox"/> EDP <input type="checkbox"/> Actyx <input checked="" type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Decentralised Federated Learning Framework (Fedra)
- KER-2: Lightweight ML methods: Early-Exit, Knowledge Distillation and Pruning
- KER-3: Decentralised Early Exit Inference Tool (D-exit)

KER Title	Decentralised Federated Learning Framework (Fedra)				
Short Description					
Fedra tool provides a fully decentralised framework for federated learning in swarm systems, using peer-to-peer communications among the participating nodes					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:				
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready			TRL estimate	
Validation evidence	The Fedra framework has been used in the EDP energy use case to train forecasting models among different smart homes/nodes				
Pain Point (i.e., what problem does it solve?)					

Fedra offers a resilient, fully decentralised federated learning scheme, specifically adapted for swarm systems to enable collaborative intelligence sharing across different nodes.

Value proposition (i.e., why it is better than the alternatives?)

The federated learning framework is lightweight and fully decentralised, in the sense that it does not need a central point of aggregation and it can operate even if swarm nodes are offline or unresponsive. In addition, due to its decentralised nature, it is resilient to attacks since no single point of failure exists in the framework.

Target users / customers

- **Primary adopters:**
 - Researchers and practitioners in distributed federated learning systems
 - Developers that design decentralised ML applications in swarm systems
- **Secondary adopters:**
 - Organisations or companies that develop swarm systems and applications
 - Academic users for teaching and research
- **Sectors/markets:**
 - Distributed ML systems and collaborative learning
 - Telecommunications and networked systems
 - Edge and IoT systems
 - Software engineering and developer tools

Dependencies and integration requirements

- **TaRDIS components needed:** None
- **Third-party dependencies:** libp2p for p2p communication and whichever libraries are required for the ML model training (e.g., Pytorch) that depend on the specific model
- **Hardware/network assumptions:** No specific hardware requirements; applicable to distributed and networked environments
- **Skills required to adopt:**
 - Knowledge of ML training and basic ML operations (data processing and training pipelines)
 - Networking principles are required to understand the p2p communication logic
 - Familiarity with federated learning general concepts is recommended

Unique technical features

- The fully decentralised FL scheme of Fedra enables the intelligence sharing among the swarm nodes in the form of model parameters.
- The parameters of the ML models are acknowledged peer-to-peer (P2P) between the FL participants, boosting the data privacy.
- Each swarm node aggregates the received model weights into a global ML model, incorporating the distilled intelligence of all nodes.
- The framework is model agnostic, designed to accommodate from simple ML models (e.g., regressors, classifiers, etc.) to complex neural network models
- The FL scheme exhibits enhanced resilience due its fully decentralised architecture, since no single point of failure exists and supports adaptive learning topology.

Evidence and artefacts

Public deliverable(s):

- D5.2: Second report on Distributed AI and AI-based orchestration
- D5.3: Final report on distributed AI and AI-based orchestration

Code repository (public/private):

- <https://github.com/anaskalt/fedra>

Demo video/slide deck:

- [TaRDIS Fedra+Pruning Demo](#)

Benchmark/evaluation:

<ul style="list-style-type: none"> KPIs for Fedra tool are included in D5.3: Final report on distributed AI and AI-based orchestration 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input checked="" type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Additional development of the Fedra framework with more challenging ML models (e.g., CNNs, DQNs, etc.)	NKUA	Month 12	Research outcomes and community engagement	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Additional development and testing of the Fedra framework in more challenging environments (e.g., node churning)	NKUA	Month 18	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Additional testing of the Fedra framework in real environments (swarm nodes that are deployed in wireless environments)	NKUA	Month 24	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> Releases (software/docs): Academic publications reporting the developed methods and results and update of the GitHub release Pilot deliverables: Validation in use cases that typically require collaborative FL training in swarm systems, as well as demonstrations of advanced features and presentations in upcoming conferences Business outputs (offers, proposals): Use of Fedra framework in networking concepts to follow-up research proposals and collaborations, as well as contributions to standardisation efforts on decentralised FL frameworks 					

KER Title	Lightweight ML methods: Early-Exit, Knowledge Distillation and Pruning				
Short Description					
Fedra tool provides a fully decentralised framework for federated learning in swarm systems, using peer-to-peer communications among the participating nodes					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem		<input type="checkbox"/> Decentralised Data Management Framework		
	<input type="checkbox"/> Improved Actyx Middleware		<input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training		
<input checked="" type="checkbox"/> Not applicable / other:					
Current maturity	<input type="checkbox"/> research concept	<input checked="" type="checkbox"/> prototype	<input checked="" type="checkbox"/> validated in lab	TRL estimate	
	<input type="checkbox"/> demo-ready	<input type="checkbox"/> pilot-ready			
Validation evidence	Knowledge Distillation (KD) functionality has been integrated with FLaaS in the TID use case				
Pain Point (i.e., what problem does it solve?)					

These tools solve the problem of computational resources and energy required at the edge of the networks when required to infer ML models. These resources are typically quite limited in unstable wireless environments and swarm systems.

Value proposition (i.e., why it is better than the alternatives?)

By using a different variety of lightweight methods, an optimal trade-off can be achieved between energy and computational resources savings and accuracy/latency of the inference process. In swarm environments, each node can in principle host a ML model for various tasks (e.g., anomaly detection, forecasting, etc.). The inference process of the ML models is a continuous process that requires node energy that may deplete its resources and battery. Using the developed lightweight tools, this continuous process is becoming more energy-efficient and faster at the expense of degradation of the model accuracy

Target users / customers

- **Primary adopters:**
 - Researchers and practitioners in distributed edge systems
 - Developers that design ML models for swarm systems
- **Secondary adopters:**
 - Organisations or companies that develop swarm systems and ML applications that require low-latency inference
 - Academic users for teaching and research
- **Sectors/markets:**
 - Distributed ML systems
 - Telecommunications and networked systems
 - Edge and IoT systems
 - Software engineering and developer tools

Dependencies and integration requirements

- **TaRDIS components needed:** None
- **Third-party dependencies:** Whichever libraries were required for the training of the original ML model training (e.g., Pytorch), i.e., before its lightweight version
- **Hardware/network assumptions:** No specific hardware requirements; applicable to distributed and networked environments
- **Skills required to adopt:**
 - Knowledge of ML training and basic ML operations (data processing and training pipelines)
 - Knowledge of the basic architecture of the DNN models in case of early-exit method and the KD method

Unique technical features

- Knowledge distillation technique can be effortlessly integrated with federated learning environments for providing more lightweight training
- The early-exit method is appropriate for swarm systems, since different nodes can host different parts of the ML model, boosting the resilience of the ML inference process
- The pruning method is DNN model agnostic, providing an overall model compression technique that reduces the required memory size and CPU per inference sample.

Evidence and artefacts

Public deliverable(s):

- D5.1: Initial report on Distributed AI and AI-based orchestration
- D5.2: Second report on Distributed AI and AI-based orchestration
- D5.3: Final report on distributed AI and AI-based orchestration

Code repository (public/private):

- <https://github.com/levgiorg/EarlyExit>
- <https://github.com/levgiorg/KnowledgeDistillation>
- <https://github.com/levgiorg/Pruning>

Benchmark/evaluation:						
<ul style="list-style-type: none"> KPIs for the lightweight tools and trade-off between ML model accuracy and inference latency/resource savings are included in D5.3: Final report on distributed AI and AI-based orchestration 						
Exploitation Plan (Next 12–24 Months)						
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input checked="" type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept			<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input checked="" type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status	
M1	Additional development is planned for the early-exit tool to include more complex models	NKUA	Month 12	Research outcomes and community engagement	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done	
M2	Integration is planned for lightweight tools with different FL frameworks to enable lightweight training pipelines at edge systems	NKUA	Month 18	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done	
M3	Additional testing of the lightweight tools with open-source FL frameworks to test lightweight training and inference pipelines at swarm systems	NKUA	Month 24	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done	
Expected outputs (what will be delivered)						
<ul style="list-style-type: none"> Releases (software/docs): Academic publications reporting the developed methods and results and update of the GitHub repositories Pilot deliverables: Validation in use cases that typically require lightweight inference of decreased inference latency in swarm systems, as well as demonstrations of advanced features and presentations in upcoming conferences (e.g., ANTS 2026) Business outputs (offers, proposals): Use of the lightweight tools in general IoT use cases that are resource-constrained devices. In addition, these tools can be utilised in follow-up research proposals and collaborations. 						

KER Title	Decentralised Early Exit Inference Tool (D-exit)				
Short Description					
D-exit is a tool that can be combined with the early-exit method for ML model inference. D-exit offers the deployment of ML models in swarm systems, where different parts of the model are hosted in different swarm nodes.					
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem		<input type="checkbox"/> Decentralised Data Management Framework		
	<input type="checkbox"/> Improved Actyx Middleware		<input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training		
<input checked="" type="checkbox"/> Not applicable / other:					
Current maturity	<input type="checkbox"/> research concept		<input checked="" type="checkbox"/> prototype		TRL estimate
	<input type="checkbox"/> demo-ready		<input type="checkbox"/> pilot-ready		

Validation evidence	The D-exit framework has been demonstrated as a standalone tool, combined with the early-exit method
Pain Point (i.e., what problem does it solve?)	
D-exit offers the deployment of the ML model that has been trained in an early-exit version directly to a swarm system, abstracting all the required communications between peers from the end-user/developer	
Value proposition (i.e., why it is better than the alternatives?)	
The decentralised early-exit of inference framework for swarm systems (D-Exit) is based on a peer-to-peer communication, distributing the inference process across swarm nodes, reducing the computational burden of individual participants and minimizing the single point of failure risk.	
Target users / customers	
<ul style="list-style-type: none"> ● Primary adopters: <ul style="list-style-type: none"> ○ Researchers and practitioners in distributed edge systems ○ Developers that design ML models for swarm systems ● Secondary adopters: <ul style="list-style-type: none"> ○ Organisations or companies that develop swarm systems and ML applications that require low-latency inference and system resilience ○ Academic users for teaching and research ● Sectors/markets: <ul style="list-style-type: none"> ○ Distributed ML systems ○ Telecommunications and networked systems ○ Edge and IoT systems ○ Software engineering and developer tools 	
Dependencies and integration requirements	
<ul style="list-style-type: none"> ● TaRDIS components needed: None ● Third-party dependencies: libp2p for p2p communication and whichever libraries are required for the ML model training (e.g., Pytorch) that depend on the specific model ● Hardware/network assumptions: No specific hardware requirements; applicable to distributed and networked environments ● Skills required to adopt: <ul style="list-style-type: none"> ○ Knowledge of ML training and basic ML operations (data processing and inference pipelines) ○ Knowledge of the basic architecture of the DNN models that will be deployed in the swarm 	
Unique technical features	
<ul style="list-style-type: none"> ● The D-exit framework implements early-exit offloading strategies that allow for adaptive computation based on the complexity of the input. Uncomplicated inferences are completed at the swarm nodes that host the primary model part, reducing the latency and overall computational load, while effectively utilizing the available network resources ● D-Exit distributes the model inference process across multiple swarm nodes, reducing the computational burden on any single point in the system and minimizing the single point of failure risk. ● The swarm nodes assume the roles of IoT/edge/cloud nodes of the typical hierarchical computing architecture, depending on the model branch that they host and can be used interchangeably to feed-forward an inference sample. ● The communication between the swarm members is conducted in a fully distributed peer-to-peer manner, providing zero dependencies on centralised nodes, while supporting dynamic adaptation to varying network conditions. ● The D-Exit framework enhances the resilience of the distributed early-exit, since it allows for seamless addition or removal of nodes, enabling the application to scale dynamically with respect to the swarm size. 	
Evidence and artefacts	
<p>Public deliverable(s):</p> <ul style="list-style-type: none"> ● D5.2: Second report on Distributed AI and AI-based orchestration 	

<ul style="list-style-type: none"> D5.3: Final report on distributed AI and AI-based orchestration <p>Code repository (public/private):</p> <ul style="list-style-type: none"> https://github.com/anaskalt/dexit <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> TaRDIS D-exit demo <p>Benchmark/evaluation:</p> <ul style="list-style-type: none"> KPIs for the D-exit tool are included in D5.3: Final report on distributed AI and AI-based orchestration 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input checked="" type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Additional development of the D-exit framework with complex ML models (e.g., CNN models for object detection)	NKUA	Month 12	Research outcomes and community engagement	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Additional development and testing of the D-exit framework in more challenging environments (e.g., node churning and peer complementarity)	NKUA	Month 18	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Additional testing of the D-exit framework in real environments (e.g., UAV swarm nodes)	NKUA	Month 24	Research outcomes and community engagement	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> Releases (software/docs): Academic publications reporting the developed methods and results and update of the GitHub repository Pilot deliverables: Validation in use cases that typically require sharing of the computational burden of the inference process in swarm systems, and demonstrations of advanced features and presentations in upcoming conferences (e.g., IEEE DESSERT 2026) Business outputs (offers, proposals): Use of D-exit framework in general IoT use cases in follow-up research proposals and collaborations 					

IP, licensing and access rights

Background IP needed
<ul style="list-style-type: none"> Your background needed: Existing research on decentralised systems related to ML model training (including collaborative learning) and inference pipelines Third-party IP/licenses: Open-source dependencies (e.g., libp2p, Pytorch, etc.) used from all tools (Fedra, Pruning, Early-exit, Knowledge Distillation, D-exit)
Foreground ownership
<input checked="" type="checkbox"/> Single partner-owned <input type="checkbox"/> Joint ownership – partners: <input type="checkbox"/> Unsure / to be agreed

Licensing intentions (software / docs)
<ul style="list-style-type: none"> ● Preferred licence (if open source): Open-source licence for all tools (e.g., MIT / BSD-style licence, consistent with existing licences in the repositories) ● Components intended to remain proprietary: None ● Constraints/notes (patents, NDAs, data restrictions): No known IP restrictions; released as open-source software for research and development use

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> ● Components: Fedra, D-exit, Pruning, Knowledge Distillation, Early-Exit ● Documentation: User documentation, tutorials, and examples for using the 5 provided repositories ● Demo/benchmark artefacts: None 	
Maintenance commitment	Governance and hosting preference
<input checked="" type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # software releases: 0 (No formal releases) ● # demos delivered: 2 demos in the framework of the mid-review ● # external stakeholders engaged: Academic research community in distributed systems and decentralised applications ● Repo metrics (stars/forks/downloads/issues): ● # pilots initiated: Validation through TaRDIS use cases and protocol analyses ● Publications/citations (optional): See above
Expected metrics (next 12–24 months)
<ul style="list-style-type: none"> ● Target releases: 1–2 repository/documentation updates ● Target pilots: 1–2 repository/documentation updates ● Target adopters/users: Students, researchers, and interested project partners that are working in swarm systems or ML applications in decentralised configurations ● Target training/outreach: Integration into courses, student projects, workshops, and research dissemination, as well as conference publications, workshops, and academic dissemination
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> ● Metric(s): <ul style="list-style-type: none"> ○ Fedra: <ul style="list-style-type: none"> ▪ Transmission Overhead (K-O-3.3): 25% reduction vs FedAvg ▪ Model Compression (K-O-3.4): 15% increase vs original model ▪ Training Time (K-O-3.5) ▪ Training Latency (K-B-07) ▪ Accuracy (K-B-10) ▪ RAM Requirements (K-B-08) ○ Lightweight tools and D-exit: <ul style="list-style-type: none"> ▪ Transmission Overhead (K-O-3.3), ▪ CPU Usage (K-B-06)

<ul style="list-style-type: none"> ▪ Inference Latency ▪ Model Compression (K-O-3.4) <ul style="list-style-type: none"> ● Baseline: Reported in D7.5 for the methodology and D5.3 [7] for results ● Result: Reported in D5.3 [7] for results ● Evidence link: Reported in D5.3 [7] for results
--

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Challenges in scaling analysis to larger distributed systems that required FL training or deployment of decentralised models. 2. Unavailability of equipment for running tests in real-word environments (e.g., UAV swarms).
Mitigation actions
<ol style="list-style-type: none"> 1. Run validation in simulated environment with an increasing number of peers/swarm nodes to evaluate the impact of scaling 1. Disseminate results through publications and training activities 2. Encourage open-source contributions to the public repositories.
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> ● Support from the consortium for dissemination and communication activities (workshops, training, visibility) ● Collaboration with consortium partners for further integration into interested use cases and dissemination for extended uses cases besides TaRDIS ● Follow-up for exploitation and standardisation opportunities from partners and EC ecosystem

Partner Statement

As a university, which is an educational and research motivated institute, NKUA will utilize TaRDIS results primarily in educational and research activities. In specific, Fedra, D-exit and the lightweight tools will be showcased to students and research practitioners to demonstrate how swarm systems can leverage the developed functionalities for purposes of ML applications at the network edge. In addition, NKUA interested researchers will continue to develop and extend the developed methods through follow-up research directions, presenting the achieved results in relevant workshops and conferences, as well as enhancing the collaborations with other partners/organizations and integrate the developed functionalities into future projects.

6.6 PARTNER: EDP NEW (EDP)

Organisation name	EDP NEW		Short	EDP
Contact person for exploitation	Name	Gonçalo Lacerda	Partner Type	Industry Large Company
	Email	goncalo.lacerda@edp.pt		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input checked="" type="checkbox"/> EDP <input type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Multi-Level Energy Community Architecture for Prosumers, Community Orchestrators and Grid Interaction
- KER-2: Decentralised Energy and Monetary Transaction Mechanisms Between Prosumers and Community Orchestrators for Energy Communities

- KER-3: Federated Learning–Based Forecasting and Decision Support for Energy Communities
- KER-4: Increased Low-Voltage Grid Resilience and CO₂ Reduction through Local Renewable Exchange

KER Title	Multi-Level Energy Community Architecture for Prosumers and Grid Interaction			
Short Description				
A distributed architecture enabling coordinated operation of households, energy communities, and the distribution grid. It defines clear roles for prosumers, community orchestrators, and grid operators, allowing local renewable production and consumption to be managed and balanced before interacting with the wider grid.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: Multi-level energy community architecture			
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	5-6	
Validation evidence	Validated in a controlled laboratory environment at EDP LABELLEC, integrating real EV chargers, smart meters, solar generation emulation, and community/grid coordination scenarios as reported in D7.1–D7.3. It was also tested on a demo which is a more controlled environment within 8 Raspberry Pi devices that represent 6 prosumers and 2 community orchestrators and all the possible transactions between them.			
Pain Point (i.e., what problem does it solve?)				
Centralised grid architectures struggle to cope with high penetration of distributed renewable energy resources and active prosumers, leading to local imbalances, congestion, and reduced quality of service				
Value proposition (i.e., why it is better than the alternatives?)				
<ul style="list-style-type: none"> • Enables local balancing before grid intervention • Reduces grid stress without immediate reinforcement • Supports scalable deployment of energy communities • Provides a clear operational separation of roles across levels 				
Target users / customers				
<ul style="list-style-type: none"> • Primary adopters: Energy Communities <ul style="list-style-type: none"> ○ Producers of Energy ○ Consumers of energy ○ Prosumers • Secondary adopters: <ul style="list-style-type: none"> ○ Other DSOs ○ Industry-related Consumers/Producers • Sectors/markets: <ul style="list-style-type: none"> ○ Electricity distribution ○ Smart grids and energy communities 				
Dependencies and integration requirements				
Third-party dependencies: <ul style="list-style-type: none"> • Grid emulation and energy metering interfaces • TaRDIS runtime services Hardware/network assumptions: <ul style="list-style-type: none"> • All the prosumers/community orchestrators were validated previously. 				

<ul style="list-style-type: none"> • Smart meters, EV chargers, local computation nodes. • IP-based communication between households, communities, and grid. <p>Skills required to adopt:</p> <ul style="list-style-type: none"> • Power systems engineering • Distributed system integration 					
Unique technical features					
<ul style="list-style-type: none"> • Hierarchical household–community–grid architecture; • Explicit community orchestrator role; • Event-driven coordination between distributed actors; • Technology-agnostic integration of prosumer assets; • Designed for high DER and EV penetration. 					
Evidence and artefacts					
<p>Public deliverable(s): D7.1, D7.2, D7.3, D7.4, D7.5</p> <p>Code repository (public/private): TaRDIS · GitLab</p> <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> • Two videos containing a recorded demo for two different points of view (POVs): <ul style="list-style-type: none"> ◦ https://drive.google.com/file/d/1GsWMYQSZV583H7KjNq52i72ffvRK5fX2/view?usp=drive_link ◦ https://drive.google.com/file/d/14hvv34xdLB8OwIHUQc4ZnpPynrdZcY5/view?usp=drive_link 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Internal technology transfer report: what parts can be absorbed into EDP R&D toolchain, and what needs hardening	EDP	3 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Regulatory & compliance scan for community-level decentralised exchange mechanisms and data handling	EDP	6 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Limited-scope pilot definition (1 community, constrained assets, clear KPIs) aligned with D7.x KPIs and operational constraints	EDP	12 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M4	Pilot execution or follow-up funding submission (HE/DEP/national) to raise TRL and operationalise the stack	EDP	18 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					

- **Releases (software/docs):** Besides the project deliverables, there's no expected releases of software or documents after the project ending
- **Pilot deliverables:** D7.1, D7.2, D7.3, D7.4 and D7.5
- **Business outputs (offers, proposals):** There are some contacts with EDP Comercial regarding a possible interest in trying to make this as a commercial product of EDP, however the TRL is too low and needs to be improved to reach EDP's request.

KER Title	Decentralised Energy and Monetary Transaction Mechanisms Between Prosumers		
Short Description			
A decentralised mechanism enabling energy and monetary transactions among prosumers within and across energy communities. It supports agreement establishment, matching of energy surplus and deficit, and accounting of exchanged energy and value, including EVs acting as flexible prosumer assets			
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input checked="" type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:		
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	5-6
Validation evidence	Validated through multiple scenarios (6 different scenarios) involving simulated prosumers, community orchestrators and communities, including exchange agreement creation and transaction execution, as described in D7.2 and evaluated in D7.3.		
Pain Point (i.e., what problem does it solve?)			
Local renewable energy surplus is often underutilised due to lack of mechanisms for decentralised exchange and settlement between prosumers.			
Value proposition (i.e., why it is better than the alternatives?)			
<ul style="list-style-type: none"> ● Enables local energy exchange without central market dependence ● Increases utilisation of locally produced renewable energy ● Encourages active prosumer participation ● Reduces reliance on grid-level balancing services 			
Target users / customers			
<ul style="list-style-type: none"> ● Primary adopters: Energy Communities <ul style="list-style-type: none"> ○ Producers of Energy ○ Consumers of energy ○ Prosumers ● Secondary adopters: <ul style="list-style-type: none"> ○ Other DSOs ○ Industry-related Consumers/Producers ● Sectors/markets: <ul style="list-style-type: none"> ○ Electricity distribution ○ Smart grids and energy communities 			
Dependencies and integration requirements			
Third-party dependencies: <ul style="list-style-type: none"> ● Grid emulation and energy metering interfaces ● TaRDIS runtime services Hardware/network assumptions: <ul style="list-style-type: none"> ● All the prosumers/community orchestrators were validated previously. 			

<ul style="list-style-type: none"> • Smart meters, EV chargers, local computation nodes. • IP-based communication between households, communities, and grid. <p>Skills required to adopt:</p> <ul style="list-style-type: none"> • Power systems engineering • Distributed system integration 					
Unique technical features					
<ul style="list-style-type: none"> • Community-level matching of energy surplus and deficit • Decentralised agreement establishment • Integrated energy and monetary accounting • Support for heterogeneous prosumer assets • Event-driven transaction execution 					
Evidence and artefacts					
<p>Public deliverable(s): D7.1, D7.2, D7.3, D7.4, D7.5</p> <p>Code repository (public/private): TaRDIS · GitLab</p> <p>Benchmark/evaluation: KPI K-U-02.</p>					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Milestone	Description	Owner
M1	Internal technology transfer report: what parts can be absorbed into EDP R&D toolchain, and what needs hardening	EDP	3 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Regulatory & compliance scan for community-level decentralised exchange mechanisms and data handling	EDP	6 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Limited-scope pilot definition (1 community, constrained assets, clear KPIs) aligned with D7.x KPIs and operational constraints	EDP	12 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M4	Pilot execution or follow-up funding submission (HE/DEP/national) to raise TRL and operationalise the stack	EDP	18 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): Besides the project deliverables, there's no expected releases of software or documents after the project ending • Pilot deliverables: D7.1, D7.2, D7.3, D7.4 and D7.5 					

- **Business outputs (offers, proposals):** There are some contacts with EDP Comercial regarding a possible interest in trying to make this as a commercial product of EDP, however the TRL is too low and needs to be improved to reach EDP's request.

KER Title	Federated Learning–Based Forecasting and Decision Support for Energy Communities			
Short Description				
A federated learning approach enabling forecasting of energy production and consumption and supporting decentralised decision-making at community level. Learning is performed without centralising sensitive data, improving scalability and privacy in energy community operation.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	5	
Validation evidence	Evaluated through forecasting and planning scenarios using federated learning components integrated into the EDP use case, as reported in D7.2 and D7.3.			
Pain Point (i.e., what problem does it solve?)				
Centralised forecasting solutions raise scalability and privacy concerns and are poorly suited to highly distributed energy communities.				
Value proposition (i.e., why it is better than the alternatives?)				
<ul style="list-style-type: none"> ● Preserves data locality and privacy ● Scales with number of prosumers and communities ● Improves forecasting accuracy through collaboration ● Reduces central computation and communication overhead 				
Target users / customers				
<ul style="list-style-type: none"> ● Primary adopters: Energy Communities <ul style="list-style-type: none"> ○ Producers of Energy ○ Consumers of energy ○ Prosumers ● Secondary adopters: <ul style="list-style-type: none"> ○ Other DSOs ○ Industry-related Consumers/Producers ● Sectors/markets: <ul style="list-style-type: none"> ○ Electricity distribution ○ Smart grids and energy communities 				
Dependencies and integration requirements				
Third-party dependencies: <ul style="list-style-type: none"> ● Grid emulation and energy metering interfaces ● TaRDIS runtime services Hardware/network assumptions: <ul style="list-style-type: none"> ● All the prosumers/community orchestrators were validated previously. ● Smart meters, EV chargers, local computation nodes. ● IP-based communication between households, communities, and grid. Skills required to adopt: <ul style="list-style-type: none"> ● Power systems engineering 				

<ul style="list-style-type: none"> • Distributed system integration 					
Unique technical features					
<ul style="list-style-type: none"> • Federated forecasting of generation and consumption • Privacy-preserving model aggregation • Community-aware decision support • Edge-based training and inference • Compatibility with dynamic energy communities 					
Evidence and artefacts					
<p>Public deliverable(s): D7.1, D7.2, D7.3, D7.4, D7.5</p> <p>Code repository (public/private): TaRDIS · GitLab</p> <p>Benchmark/evaluation: K-B-07, K-B-08, K-B-10, K-O-3.5</p>					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Milestone	Description	Owner
M1	Internal technology transfer report: what parts can be absorbed into EDP R&D toolchain, and what needs hardening	EDP	3 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Regulatory & compliance scan for community-level decentralised exchange mechanisms and data handling	EDP	6 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Limited-scope pilot definition (1 community, constrained assets, clear KPIs) aligned with D7.x KPIs and operational constraints	EDP	12 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M4	Pilot execution or follow-up funding submission (HE/DEP/national) to raise TRL and operationalise the stack	EDP	18 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): Besides the project deliverables, there's no expected releases of software or documents after the project ending • Pilot deliverables: D7.1, D7.2, D7.3, D7.4 and D7.5 • Business outputs (offers, proposals): There are some contacts with EDP Comercial regarding a possible interest in trying to make this as a commercial product of EDP, however the TRL is too low and needs to be improved to reach EDP's request. 					

KER Title	Increased Low-Voltage Grid Resilience and CO ₂ Reduction through Local Renewable Exchange		
Short Description			
An operational outcome demonstrating improved low-voltage grid resilience and significant CO ₂ emission reduction through local renewable energy exchange within energy communities. The approach reduces grid dependency while improving quality of service under high DER and EV penetration.			
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: Grid resilience and sustainability outcome		
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	6
Validation evidence	Scenario-based validation showing 92–95% CO ₂ reduction and increased active prosumer participation, documented in D7.3 Use Case KPIs		
Pain Point (i.e., what problem does it solve?)			
High Electrical Vehicle and renewable penetration create operational challenges for low-voltage grids, threatening reliability and decarbonisation objectives			
Value proposition (i.e., why it is better than the alternatives?)			
<ul style="list-style-type: none"> ● Achieves measurable CO₂ reduction without grid reinforcement ● Improves local resilience through community-level balancing ● Supports regulatory and sustainability objectives ● Increases the amount of renewable energy usage that is being produced on community-level 			
Target users / customers			
<ul style="list-style-type: none"> ● Primary adopters: Energy Communities <ul style="list-style-type: none"> ○ Producers of Energy ○ Consumers of energy ○ Prosumers ● Secondary adopters: <ul style="list-style-type: none"> ○ Other DSOs ○ Industry-related Consumers/Producers ● Sectors/markets: <ul style="list-style-type: none"> ○ Electricity distribution ● Smart grids and energy communities 			
Dependencies and integration requirements			
<p>Third-party dependencies:</p> <ul style="list-style-type: none"> ● Grid emulation and energy metering interfaces ● TaRDIS runtime services <p>Hardware/network assumptions:</p> <ul style="list-style-type: none"> ● All the prosumers/community orchestrators were validated previously. ● Smart meters, EV chargers, local computation nodes. ● IP-based communication between households, communities, and grid. <p>Skills required to adopt:</p> <ul style="list-style-type: none"> ● Power systems engineering ● Distributed system integration 			
Unique technical features			

<ul style="list-style-type: none"> • Community-based renewable prioritisation • Multi-level resilience logic • Integration of EVs as prosumer assets • CO₂-aware operational evaluation 					
Evidence and artefacts					
Public deliverable(s): D7.1, D7.2, D7.3, D7.4, D7.5 Code repository (public/private): TaRDIS · GitLab <ul style="list-style-type: none"> • Benchmark/evaluation: KPIs K-U-01 and K-U-02. 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
	Concrete actions and milestones				
Milestone	Description	Owner	Milestone	Description	Owner
M1	Internal technology transfer report: what parts can be absorbed into EDP R&D toolchain, and what needs hardening	EDP	3 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Regulatory & compliance scan for community-level decentralised exchange mechanisms and data handling	EDP	6 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Limited-scope pilot definition (1 community, constrained assets, clear KPIs) aligned with D7.x KPIs and operational constraints	EDP	12 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M4	Pilot execution or follow-up funding submission (HE/DEP/national) to raise TRL and operationalise the stack	EDP	18 Months	-	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): Besides the project deliverables, there's no expected releases of software or documents after the project ending • Pilot deliverables: D7.1, D7.2, D7.3, D7.4 and D7.5 • Business outputs (offers, proposals): There are some contacts with EDP Comercial regarding a possible interest in trying to make this as a commercial product of EDP, however the TRL is too low and needs to be improved to reach EDP's request. 					

Sustainability and maintenance (post-project)

What will you maintain after project end?

<ul style="list-style-type: none"> ● Components: All the components that were developed during the project will be analysed for future usage of EDP and/or further research and development of these same components. ● Documentation: All the documentation produced during the project ● Demo/benchmark artefacts: All the demo and benchmark artefacts produced during the project will be analysed for future usage of EDP and/or further research and development of these same components 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input checked="" type="checkbox"/> Not planned (explain): The vast majority of the components present within EDP Use Case were developed by other partners that are in the project, so the responsibility for a maintenance commitment of these components belongs to them	<input checked="" type="checkbox"/> TaRDIS consortium repository (shared) <input type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # demos delivered: 2 ● # external stakeholders engaged: Several external stakeholders were engaged throughout several conferences and events where EDP was present for the sake of TaRDIS ● # pilots initiated: 1
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> ● Metric(s): <ul style="list-style-type: none"> ○ Grid balancing efficiency at community level ○ Utilisation of locally produced renewable energy ○ CO₂-equivalent emissions reduction (environmental impact) ○ Effort to set up and operate energy communities (process efficiency) ● Baseline: <ul style="list-style-type: none"> ○ Baseline implementation relies on a distributed but largely manual orchestration of energy matching between households, communities, and the grid. ○ Data exchange and energy balancing are performed using client-server architectures and non-automated coordination between aggregators and grid operators. ○ No TaRDIS components are used; communication, coordination, and forecasting are handled using state-of-the-art but non-swarm-native solutions. ○ Quantitative baseline values are not yet numerically established, as D7.1 focuses on baseline description and KPI definition ● Result: <ul style="list-style-type: none"> ○ D7.1 does not yet report quantified numerical improvements for the EDP use case. <p>Expected improvements are clearly identified and will be quantified in later stages:</p> <ul style="list-style-type: none"> ○ Improved short-term grid balancing through distributed, peer-to-peer coordination. ○ Increased local consumption of renewable energy, reducing reliance on national grid balancing. ○ Potential reduction of CO₂-equivalent emissions due to improved utilisation of locally produced renewable energy. ○ Reduced operational effort for establishing and managing energy communities through TaRDIS abstractions and tooling. ○ Measurement and comparison against the baseline are planned using use-case-specific KPIs defined in D7.1 and will be executed after TaRDIS integration (reported in D7.2). ● Evidence link:

Deliverable D7.1 – “ Report on the expected improvements and quantification procedures ”, <ul style="list-style-type: none"> ○ Section 2.1 (EDP baseline description and challenges) ○ Section 4 (Areas of improvement and KPI definitions) ○ Section 4.5 (KPIs applicable to the EDP use case)

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. The regulation aspect within Portugal and/or EU. 2. The interest of the marketplace itself in creating this new type of energy community environment and its implications. 3. EDP Comercial interest in transforming the tools developed in this Use Case into a commercial product, usable by EDP.
Mitigation actions
<ol style="list-style-type: none"> 1. Communicate with the National and European regulatory bodies and try to ascertain what's possible. 2. Do some questionnaires to several different types of energy producers, consumers and/or energy communities to analyse if people would be interested in this new environment. 3. Try to showcase as best as possible on an internal level to convince EDP Comercial to invest in the technologies developed within the TaRDIS project.

Partner Statement

EDP will capitalise on TaRDIS by integrating its decentralised intelligence and federated learning capabilities into future distribution grid planning and operational support services. The project enabled EDP to validate a multi-level energy community architecture, that allows local renewable energy and prosumer participation to be coordinated before interacting with the wider grid, directly addressing challenges arising from high EV and PV penetration.

Through TaRDIS, EDP demonstrated decentralised energy and monetary transaction mechanisms between prosumers, supporting new energy community business models and increased citizen engagement.

The results obtained will be used in early-stage grid redesign studies and scenario-based operational analysis to reduce congestion, improve quality of service, and maximise local renewable utilisation.

In the longer term, EDP foresees replicating the swarm-based decision and coordination principles explored in TaRDIS to other energy domains, contributing to resilient, low-carbon electricity systems aligned with European energy transition objectives.

6.7 PARTNER: GMV

Organisation name	GMV		Short	GMV
Contact person for exploitation	Name	David Vázquez Enríquez	Partner Type	Industry Large Company
	Email	dvazquez@gmv.com		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input checked="" type="checkbox"/> GMV <input type="checkbox"/> EDP <input type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Distributed ODTs Algorithms for LEO Constellations
- KER-2: Framework for Distributed Algorithm Validation and Simulation

KER Title	Distributed ODTS Algorithms for LEO Constellations		
Short Description			
Development and validation of distributed Orbit Determination and Time Synchronization (ODTS) algorithms for LEO satellite constellations. The solution demonstrates equivalent performance to centralised approaches in terms of accuracy while enabling autonomous and scalable on-board processing.			
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: Distributed ODTS algorithms for space systems		
Current maturity	<input type="checkbox"/> research concept <input type="checkbox"/> demo-ready	<input type="checkbox"/> prototype <input type="checkbox"/> pilot-ready	<input checked="" type="checkbox"/> validated in lab <div style="background-color: #800040; color: white; padding: 2px; text-align: center; font-weight: bold;">TRL estimate</div>
Validation evidence	<ul style="list-style-type: none"> • Simulation-based validation, results compared with centralised baseline. • KPI evaluation in D7.3 and D7.5. • Performance comparison under failure scenarios and scalability conditions 		
Pain Point (i.e., what problem does it solve?)			
Centralised ODTS solutions: <ul style="list-style-type: none"> • Do not scale well for megaconstellations • Require heavy ground segment dependency • Are not resilient to node failures 			
Value proposition (i.e., why it is better than the alternatives?)			
<ul style="list-style-type: none"> • Enables onboard autonomy and reduced ground operations • Scalable to large constellations • Increased robustness to failures (no single point failure) • Aligns with future autonomous satellite swarm concepts 			
Target users / customers			
<ul style="list-style-type: none"> • Primary adopters: <ul style="list-style-type: none"> ○ Satellite operators ○ Space system integrators • Secondary adopters: <ul style="list-style-type: none"> ○ European Space Agency ○ NewSpace companies • Sectors/markets: Space (navigation, Earth observation, telecommunications) 			
Dependencies and integration requirements			
TaRDIS components needed: <ul style="list-style-type: none"> • PTB-FLA (communication layer) • Babel (distributed deployment) Third-party dependencies: <ul style="list-style-type: none"> • Python • Java Hardware/network assumptions: <ul style="list-style-type: none"> • Inter-satellite links (ISLs) • Distributed onboard computing Skills required to adopt: <ul style="list-style-type: none"> • GNSS, ODTS, filtering expertise 			

<ul style="list-style-type: none"> Distributed systems understanding 					
Unique technical features					
<ul style="list-style-type: none"> Fully distributed ODTs estimation scalable to large constellations Robust to node failures and partial connectivity Designed and validated for onboard execution 					
Evidence and artefacts					
<p>Public deliverable(s):</p> <ul style="list-style-type: none"> D7.1, D7.2, D7.3, D7.4 and D7.5 <p>Code repository (public/private):</p> <ul style="list-style-type: none"> Private (GMV internal) <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> Demo video available in: https://drive.google.com/drive/folders/1yApr1hOwSG49DSGCBqWSYMxfQEzQheMj?usp=drive_link <p>Benchmark/evaluation: KPIs K-U-05, K-U-06, K-U-07 and K-U-08</p>					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Milestone	Milestone	Milestone	Milestone	Milestone
M1	Finalize and maintain the distributed ODTs algorithms evaluation framework repository and develop support documentation for internal use	GMV	2026		<input type="checkbox"/> planned <input type="checkbox"/> ongoing <input checked="" type="checkbox"/> done
M2	Internally disseminate the results achieved with TaRDIS toolbox, extending its potential use to a broader range of algorithm development engineers	GMV	2026		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Additional improvements on the distributed framework, performing more tests and introducing more capabilities such as the re-scheduling algorithm (already unitarily tested)	GMV	2027		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> Releases (software/docs): No public software or documentation releases are expected after the project ending. However, internal dissemination of the results achieved with TaRDIS will be performed, so that other algorithm engineers and architects can benefit from the developed framework Pilot deliverables: No other pilots or demos are expected to be released after the project ends 					

- **Business outputs (offers, proposals):** The lessons learned during the course of TaRDIS in the areas of swarm applications, system decentralization, communications, machine learning and other domains, as well as the decentralised framework for the evaluation of distributed ODTs algorithms developed with Babel and PTB-FLA tools, will represent a technical advancement for GMV.
- These outcomes will enable improved strategic positioning in the field of autonomous navigation solutions and may serve as a foundation for future commercial proposals and project applications.

KER Title	Framework for Distributed Algorithm Validation and Simulation		
Short Description			
A development and validation framework that enables algorithm engineers to design, test, and evaluate distributed ODTs algorithms using TaRDIS tools without requiring complex distributed hardware setups			
Innovation Radar alignment	<input checked="" type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input checked="" type="checkbox"/> Not applicable / other: Distributed ODTs algorithms for space systems		
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	
Validation evidence	<ul style="list-style-type: none"> • Integration of algorithms into TaRDIS distributed environment • Demonstrated reduction of development complexity 		
Pain Point (i.e., what problem does it solve?)			
<ul style="list-style-type: none"> • Engineers cannot easily test distributed algorithms • Real distributed setups are expensive and impractical, especially when treating with systems like megaconstellations • Communication layers may be complex to implement for non-specialised algorithm developers 			
Value proposition (i.e., why it is better than the alternatives?)			
<ul style="list-style-type: none"> • Reduces development effort • Enables fast prototyping of distributed systems • Abstracts communication complexity • Allows reproducible resting scenarios 			
Target users / customers			
<p>Primary adopters:</p> <ul style="list-style-type: none"> • Internal engineering teams (GMV) <p>Secondary adopters:</p> <ul style="list-style-type: none"> • Research teams • Space system developers <p>Sectors/markets:</p> <ul style="list-style-type: none"> • Space engineering 			
Dependencies and integration requirements			
<p>TaRDIS components needed:</p> <ul style="list-style-type: none"> • PTB-FLA • Babel <p>Third-party dependencies:</p> <ul style="list-style-type: none"> • Python • Java <p>Hardware/network assumptions:</p> <ul style="list-style-type: none"> • Single machine with multi-process capability 			

<ul style="list-style-type: none"> Several machines may be needed if the system has too many nodes or the processing cost for each node is high. <p>Skills required to adopt:</p> <ul style="list-style-type: none"> Algorithm development Basic distributed systems 					
Unique technical features					
<ul style="list-style-type: none"> Emulation of distributed systems on a single machine or a low number of machines Configurable communication topology Separation of algorithm logic from communication layer Reduces need for HW infrastructure 					
Evidence and artefacts					
<p>Public deliverable(s):</p> <ul style="list-style-type: none"> D7.1, D7.2, D7.3, D7.4 and D7.5 <p>Code repository (public/private):</p> <ul style="list-style-type: none"> Private (GMV internal) PTB-FLA: https://github.com/miroslav-popovic/ptbfla Babel: Babel · GitLab PTB-FLA - Babel adapter: https://github.com/LinguineP/PTB-FLA_BabelAdapter <p>Demo video/slide deck:</p> <ul style="list-style-type: none"> Demo video available in: https://drive.google.com/drive/folders/1yApr1hOwSG49DSGCBqWSYMxfQEzQheMj?usp=drive_link Benchmark/evaluation: KPIs K-U-05 and K-U-06 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Finalize and maintain the distributed ODTS algorithms evaluation framework repository and develop support documentation for internal use	GMV	2026		<input type="checkbox"/> planned <input type="checkbox"/> ongoing <input checked="" type="checkbox"/> done
M2	Internally disseminate the results achieved with TaRDIS toolbox, extending its potential use to a broader range of algorithm development engineers	GMV	2026		<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Additional improvements on the distributed framework, performing more tests and introducing more capabilities such as the re-scheduling algorithm (already unitarily tested)	GMV	2027		<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					

- **Releases (software/docs):** No public software or documentation releases are expected after the project ending. However, internal dissemination of the results achieved with TaRDIS will be performed, so that other algorithm engineers and architects can benefit from the developed framework
 - **Pilot deliverables:** No other pilots or demos are expected to be released after the project ends
 - **Business outputs (offers, proposals):** The lessons learned during the course of TaRDIS in the areas of swarm applications, system decentralization, communications, machine learning and other domains, as well as the decentralised framework for the evaluation of distributed ODS algorithms developed with Babel and PTB-FLA tools, will represent a technical advancement for GMV.
- These outcomes will enable improved strategic positioning in the field of autonomous navigation solutions and may serve as a foundation for future commercial proposals and project applications.

IP, licensing and access rights

Background IP needed
<ul style="list-style-type: none"> ● Your background needed: Existing research on ODS techniques and navigation concepts, graph theory for ISL scheduling determination, and decentralised systems. ● Third-party IP/licenses: Open-source dependencies such as concrete python libraries.
Foreground ownership
<input checked="" type="checkbox"/> Single partner-owned <input type="checkbox"/> Joint ownership – partners: <input type="checkbox"/> Unsure / to be agreed
Licensing intentions (software / docs)
<ul style="list-style-type: none"> ● Preferred licence (if open source): N/A ● Components intended to remain proprietary: All to remain proprietary ● Constraints/notes (patents, NDAs, data restrictions): N/A

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> ● Components: All the components that were developed during the project will be analysed for future usage of GMV and/or further research and development of the same components ● Documentation: User documentation, tutorials, examples for using the framework, test results documentation. ● Demo/benchmark artefacts: All the components that were developed during the project will be analyzed for future usage of GMV and/or further research and development of the same components. 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input checked="" type="checkbox"/> Not planned (explain): The responsibility for maintenance commitment of external components used within the GMV use case (such as Babel and PTB-FLA) does not belong to GMV. GMV will, however, internally maintain the outputs from its use case, such as the decentralised ODS algorithms and the distributed framework for algorithms testing and evaluation	<input checked="" type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):
Notes	
TaRDIS consortium repository for the TaRDIS tools used within GMV’s use case, and a GMV-owned repository for its private internal results	

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # software releases: N/A (no standalone releases) ● # demos delivered: 1 main integrated demo (GMV use case demonstrator in D7.4) ● # external stakeholders engaged: Several stakeholders involved, mainly the EC and consortium-level stakeholders) ● Repo metrics (stars/forks/downloads/issues): N/A (no public repository) ● # pilots initiated: 1 (validation performed within project scope in the main integrated demo) ● Publications/citations (optional):
Expected metrics (next 12–24 months)
<ul style="list-style-type: none"> ● Target releases: N/A (no public releases, only integration into internal GMV solutions expected) ● Target pilots: N/A (no more than the one within TaRDIS scope) ● Target adopters/users: <ul style="list-style-type: none"> ○ Internal GMV teams (navigation and algorithms engineering) ○ Potential uptake in future customer projects ● Target training/outreach: Internal knowledge transfer sessions, and possible contribution to future project proposals.
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> ● Metric(s): <ul style="list-style-type: none"> ○ K-U-05 - Achievable distributed on-board ODTs performances versus the classical centralised on-ground ODTs. Quantitatively measured against known ground ODTs performances. Same order of magnitude is expected. ○ K-U-06 - Reduction of the use of computational resources: memory, CPU time, and energy. Quantitatively measured against known ground ODTs performances. Several orders of magnitude reduction are expected ● Baseline: <ul style="list-style-type: none"> ○ K-U-05 - Error in meter or sub-meter level ○ K-U-06 - Ground-based ODTs processing baseline (centralised approach, one node obtains the solution for the entire constellation) ● Result: <ul style="list-style-type: none"> ○ K-U-05 - Error slightly higher but near to the baseline, as expected. ○ K-U-06 - Several orders of magnitude of reduction in resources usage. ● Evidence link: <ul style="list-style-type: none"> ○ Deliverables D7.3 and D7.5

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. High complexity of distributed ODTs algorithms. Distributed estimation introduces challenges in convergence, stability and tuning, especially under realistic communication constraints. 2. Scalability limitations linked to simulation and hardware resources. Even with TaRDIS support, simulating very large constellation requires significant computational resources 3. Gap between simulation and operational deployment. Transitioning from simulation environments to real on-board systems involves constraints (hardware, certification, real-time requirements) that may limit direct applicability.
Mitigation actions
<ol style="list-style-type: none"> 1. Progressive validation approach, starting from pure computational simulation moving towards hardware-in-the-loop setups.

<ol style="list-style-type: none"> 2. Use of scalable tools and hardware purchases to extend simulation capabilities. 3. Incremental integration into existing GMV navigation frameworks. 3. Continued research to refine distributed estimation techniques and ensure robustness.
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> • Support for dissemination and outreach activities within research and industrial communities. • Collaboration with partners for tools integration

Partner Statement

GMV will exploit the results of TaRDIS by integrating the developed distributed ODTs algorithms and validation methodologies into its internal navigation software toolchain. These results will support the transition towards more autonomous and scalable satellite constellation operations, reducing reliance on ground infrastructure and enabling advanced swarm-based concepts.

The knowledge and capabilities gained through TaRDIS will be directly leveraged in future research and development activities, proposals, and customer-driven projects, strengthening GMV’s position in the field of distributed space systems. In particular, the ability to efficiently design, test, and validate distributed navigation algorithms represents a key competitive advantage for next-generation LEO constellations.

While the developed components are not intended as standalone commercial products, they constitute critical enablers for future high-value services and system-level solutions in the space sector.

6.8 PARTNER: CAIXA MÁGICA SOFTWARE (CMS)

Organisation name	Caixa Mágica Software		Short	CMS
Contact person for exploitation	Name	Carlos Coutinho	Partner Type	SME
	Email	carlos.coutinho@caixamagica.pt		
TaRDIS use case(s)	<input checked="" type="checkbox"/> Telefónica <input checked="" type="checkbox"/> GMV <input checked="" type="checkbox"/> EDP <input checked="" type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: TaRDIS IDE (VS Code Extension)

KER Title	TaRDIS IDE (VS Code Extension)			
Short Description				
A VS Code-based integration layer that orchestrates TaRDIS tools behind consistent workflows (setup → build/run → inspect), complemented by a structured “developer experience” catalogue and tool usage matrix to guide adoption across SDLCs and use cases.				
Innovation Radar alignment	<input checked="" type="checkbox"/> Babel Framework and Ecosystem <input checked="" type="checkbox"/> Decentralised Data Management Framework <input checked="" type="checkbox"/> Improved Actyx Middleware <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab <input checked="" type="checkbox"/> demo-ready <input type="checkbox"/> pilot-ready	TRL estimate	4-5	
Validation evidence	D3.6 report + released demonstrator (CodeLab + Zenodo).			

<p>Pain Point (i.e., what problem does it solve?)</p> <p>TaRDIS is a multi-tool, multi-language toolbox; without an integration surface, developers face high onboarding friction, fragmented workflows, and error-prone configuration (dependencies, runtimes, execution environments). The IDE reduces context switching and standardises execution and inspection paths.</p>
<p>Value proposition (i.e., why it is better than the alternatives?)</p> <ul style="list-style-type: none"> • Uses VS Code, a widely adopted, extensible, cross-platform IDE, reducing adoption friction versus bespoke tooling. • Provides workflow-level integration (projects/templates, dependency automation, Dockerised execution, documentation viewer) instead of forcing users into disparate CLIs and ad-hoc scripts. • Adds a recipe catalogue and SDLC mapping that translates toolbox capabilities into repeatable engineering practice, improving repeatability and teachability. • Use-case storyboards were developed for each TaRDIS pilot to describe the developer approach in each case, but a generic storyboard is still under development.
<p>Target users / customers</p> <ul style="list-style-type: none"> • Primary adopters: <ul style="list-style-type: none"> ◦ Developers and engineers building decentralised/swarm applications with TaRDIS tooling (consortium and external). ◦ Tool owners/integrators who need a unified UX for heterogeneous components. • Secondary adopters: <ul style="list-style-type: none"> ◦ Research groups and advanced students learning swarm programming + verification workflows. ◦ System integrators/consultancies needing reproducible demos and onboarding paths. • Sectors/markets: <ul style="list-style-type: none"> ◦ Industrial automation/manufacturing, energy systems/communities, aerospace systems, telecom/smart-home digital services.
<p>Dependencies and integration requirements</p> <ul style="list-style-type: none"> • TaRDIS components needed: Depending on scenario/recipe: Babel projects & protocols; PTB-FLA; DCR editor/compiler/runtime; FAUNO; FEDRA; configuration management; nuScr; plus other TaRDIS tools integrated or referenced in D3.6. • Third-party dependencies: Visual Studio Code; Node.js; Java 21 JDK; Maven; Python and venv; Docker; Git • Hardware/network assumptions: Standard dev workstation; Docker-capable environment; network access for cloning repos and pulling dependencies (where applicable). • Skills required to adopt: Basic software development skills (VS Code, Git); language-specific skills depending on tool (Java/Maven, Python/venv, Docker); familiarity with distributed systems helpful but recipes aim to lower barrier.
<p>Unique technical features</p> <ul style="list-style-type: none"> • Single VS Code “cockpit” for multi-project swarm development (Babel/PTB-FLA/DCR). • Automated dependency/configuration steps (e.g., pom.xml updates, venv creation, Docker run orchestration). • Embedded documentation viewer and guided flows. • Recipe catalogue + tool usage matrix mapping tools to SDLC phases and use cases. • Optional TaRDIS Assistant concept (chatbot guidance).
<p>Evidence and artefacts</p> <p>Deliverables:</p> <ul style="list-style-type: none"> • D3.2, D3.4, D3.6; and architectural framing in D2.3. <p>Code repository (public/private):</p> <ul style="list-style-type: none"> • https://codelab.fct.unl.pt/di/research/TaRDIS/toolkit/ide/vscode/vscode-TaRDIS-extension
<p>Exploitation Plan (Next 12–24 Months)</p>

Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release	<input checked="" type="checkbox"/> Services (integration, support, training)			
	<input checked="" type="checkbox"/> Integration into partner product/service	<input checked="" type="checkbox"/> Research exploitation (follow-on projects)			
	<input checked="" type="checkbox"/> Joint exploitation with other partners	<input type="checkbox"/> Standardisation / pre-standardisation influence			
	<input checked="" type="checkbox"/> External pilots / proofs of concept	<input type="checkbox"/> Other:			
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Publish TaRDIS IDE extension to VS Code Marketplace	CMS	12 months	Packaging/signing, licensing checks, documentation finalisation	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Convert D3.6 “recipes” into an online cookbook with IDE-linked steps and reproducible examples	CMS + NOVA + tool owners	12-18 months	Partner validation of tool steps; stable example repos	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Establish maintenance governance: issue triage, release cadence, compatibility matrix (IDE ↔ tools ↔ OS)	CMS	6 months	Access to repos, agreement on versioning	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): VS Code Marketplace release(s) of the TaRDIS IDE extension; periodic updates; compatibility matrix; onboarding documentation; recipe cookbook. • Pilot deliverables: One or more external Proofs of Concept (PoC) demonstrating end-to-end recipe execution with selected toolchains (e.g., DCR+Babel energy workflow; PTB-FLA+Babel ODTs workflow). • Business outputs (offers, proposals): Service offer for onboarding / integration / training; follow-on project proposals leveraging IDE + recipes as dissemination/adoption package. 					

IP, licensing and access rights

Background IP needed
<ul style="list-style-type: none"> • Your background needed: CMS know-how and code for VS Code extension integration/orchestration patterns; documentation assets • Third-party IP/licenses: VS Code extension ecosystem; dependencies under their respective OSS licenses; confirm compatibility for redistribution.
Foreground ownership
<input checked="" type="checkbox"/> Single partner-owned <input type="checkbox"/> Joint ownership – partners: <input type="checkbox"/> Unsure / to be agreed
Licensing intentions (software / docs)
<ul style="list-style-type: none"> • Preferred licence (if open source): Apache-2.0 • Components intended to remain proprietary: None • Constraints/notes (patents, NDAs, data restrictions): None

Sustainability and maintenance (post-project)

What will you maintain after project end?
<ul style="list-style-type: none"> • Components: TaRDIS VS Code extension (core), integration connectors/workflows, packaging scripts, compatibility matrix. • Documentation: Installation guide, tool integration guides, recipe cookbook

<ul style="list-style-type: none"> • Demo/benchmark artefacts: Curated example workspaces (Babel/PTB-FLA/DCR) aligned with selected recipes. 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input checked="" type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months) <input type="checkbox"/> Not planned (explain):	<input type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):
Notes	
<p>As this is strongly dependent on tools evolution, it would be interesting to establish a lightweight joint triage channel with tool owners to manage breaking changes and keep integrations stable across versions</p>	

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> • # software releases: 1 archival release on Zenodo (DOI assigned); additional internal iterations reflected by D3.2/D3.4/D3.6 • # demos delivered: 3 demonstrators (TaRDIS IDE extension) (D3.2, D3.4, D3.6). • # external stakeholders engaged: None • Repo metrics (stars/forks/downloads/issues): • # pilots initiated: None • Publications/citations (optional): https://doi.org/10.5281/zenodo.19211021
Expected metrics (next 12–24 months)
<ul style="list-style-type: none"> • Target releases: 2–4 tagged releases (including Marketplace releases + maintenance updates). • Target pilots: 1–2 external PoCs using recipe-driven onboarding (e.g., energy community stack; ODTS stack; Manufacturing stack). • Target adopters/users: 20–50 developers reached via trainings/workshops and downloads • Target training/outreach: 2 workshops/webinars + 1 published “TaRDIS IDE cookbook” site.
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> • Metric(s): Onboarding time (install → run first end-to-end scenario), number of manual configuration steps, reproducibility success rate across OS. • Baseline: manual multi-tool setup via disparate CLIs/scripts • Result: IDE-guided end-to-end workflow with reduced context switching • Evidence link: <ul style="list-style-type: none"> ○ https://codelab.fct.unl.pt/di/research/TaRDIS/toolkit/ide/vscode/vscode-TaRDIS-extension ○ https://doi.org/10.5281/zenodo.19211021

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Integration fragility / dependency drift across TaRDIS tools (version mismatches; changing repos; OS differences). 2. Adoption friction if distribution remains “manual install/debug mode” rather than Marketplace. 3. Sustainability risk if maintenance responsibilities across tool owners are unclear (IDE depends on evolving tools).
Mitigation actions
<ol style="list-style-type: none"> 1. Maintain a compatibility matrix (IDE ↔ tool versions ↔ OS prerequisites); add CI smoke tests running key recipes. 2. Publish via VS Code Marketplace; provide a single canonical onboarding path (“one-hour success”).

3. Set up governance with tool owners (lightweight steering + triage); document integration contracts per tool.
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> Alignment on licensing and long-term hosting expectations (especially if some tool repos are partner-owned). Support to recruit external early adopters (HRB/EC channels, Innovation Radar network) for 1–2 post-project PoCs. Coordination with tool owners to stabilise interfaces and publish versioned releases.

Partner Statement

CMS will continue to exploit TaRDIS results by maintaining and evolving the TaRDIS IDE as the primary adoption layer for the toolbox, converting project outcomes into repeatable developer workflows and recipe-driven guidance, and enabling external uptake through open distribution, documentation, and targeted pilots and support services.

6.9 PARTNER: TELEFÓNICA INNOVACIÓN DIGITAL (TID)

Organisation name	Telefónica Innovación Digital		Short	TID
Contact person for exploitation	Name	Filippo Vannella	Partner Type	Industry Large Company
	Email	filippo.vannella@telefonica.com		
TaRDIS use case(s)	<input checked="" type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input type="checkbox"/> EDP <input type="checkbox"/> Actyx <input type="checkbox"/> Other/None:			

Your Key Exploitable Results

- KER-1: Privacy-preserving learning through decentralised training in smart homes

KER Title	Privacy-preserving learning through decentralised training in smart homes			
Short Description				
This KER extends FLaaS with differential privacy-preserving mechanisms to support more private deployment of FL workflows in resource-constrained edge environments.				
Innovation Radar alignment	<input type="checkbox"/> Babel Framework and Ecosystem <input type="checkbox"/> Decentralised Data Management Framework <input type="checkbox"/> Improved Actyx Middleware <input checked="" type="checkbox"/> Privacy-Preserving & Efficient Decentralised Training <input type="checkbox"/> Not applicable / other:			
Current maturity	<input type="checkbox"/> research concept <input checked="" type="checkbox"/> prototype <input checked="" type="checkbox"/> validated in lab	TRL estimate	3-4	
Validation evidence	The knowledge distillation functionality has been validated in combination with FLaaS to support privacy-preserving distributed intelligence scenarios and forms the basis of a published research output.			
Pain Point (i.e., what problem does it solve?)				
Standard federated learning can be susceptible to attacks (e.g., inverse gradient) compromising privacy of user. This KER addresses the need to train model privately, and with robustness to attacks.				
Value proposition (i.e., why it is better than the alternatives?)				
By combining FL with privacy preserving learning, this result enables practical edge deployment than standard FL alone.				
Target users / customers				
<ul style="list-style-type: none"> Primary adopters: 				

<ul style="list-style-type: none"> ○ Edge AI teams within Telefónica ○ Developers building lightweight distributed AI services ● Secondary adopters: <ul style="list-style-type: none"> ○ Enterprises with constrained edge environments ○ Providers of privacy-preserving AI solutions ○ Research teams in edge intelligence and distributed ML ● Sectors/markets: <ul style="list-style-type: none"> ○ Telecommunications ○ Edge AI ○ IoT ○ Distributed ML services ○ Privacy-aware digital services 					
Dependencies and integration requirements					
<ul style="list-style-type: none"> ● TaRDIS components needed: <ul style="list-style-type: none"> ○ FLaaS core component ○ Optional integration with other TaRDIS lightweight AI methods ● Third-party dependencies: <ul style="list-style-type: none"> ○ ML frameworks supporting teacher/student training workflows for knowledge distillation ○ Deployment infrastructure for edge/cloud experiments ● Hardware/network assumptions: <ul style="list-style-type: none"> ○ Distributed training/deployment environment ○ Edge nodes with constrained resources ○ Network support for federated update exchange ● Skills required to adopt: <ul style="list-style-type: none"> ○ Familiarity with FL and knowledge distillation ○ ML engineering and deployment skills ○ Basic cloud/edge orchestration skills 					
Unique technical features					
<ul style="list-style-type: none"> ● Combines FLaaS with knowledge distillation for efficient collaborative learning ● Better fit for constrained edge infrastructures than conventional FL-only pipelines ● Supports reduced model complexity while preserving collaborative learning benefits ● Suitable for privacy-preserving edge intelligence scenarios ● Reusable as a building block for future edge AI offerings and pilots 					
Evidence and artefacts					
<p>Public deliverable(s): D7.3, D7.4, D7.5</p> <p>Code repository (public/private): public repository (MIT licences)</p> <ul style="list-style-type: none"> ● FLaaS Server: https://github.com/TianyueChu/FLaaS-Server ● FLaaS: https://github.com/TianyueChu/FLaaSAndroid 					
Exploitation Plan (Next 12–24 Months)					
Exploitation route(s)	<input checked="" type="checkbox"/> Open-source release <input checked="" type="checkbox"/> Integration into partner product/service <input type="checkbox"/> Joint exploitation with other partners <input type="checkbox"/> External pilots / proofs of concept		<input checked="" type="checkbox"/> Services (integration, support, training) <input checked="" type="checkbox"/> Research exploitation (follow-on projects) <input type="checkbox"/> Standardisation / pre-standardisation influence <input type="checkbox"/> Other:		
Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status

M1	Consolidate FLaaS as a reusable software component with improved documentation, interfaces, and deployment guidance	TID	Month 12	Internal technical prioritisation and repository maintenance	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Validate FLaaS in additional edge/cloud scenarios relevant to telecom and distributed AI services	TID	Month 18	Availability of pilot environments and integration support	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Assess integration opportunities of FLaaS into follow-up Telefónica innovation activities, proposals, or service concepts	TID	Month 24	Business alignment, internal interest, and external collaboration opportunities	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done

Expected outputs (what will be delivered)

- **Releases (software/docs):** Updated FLaaS software baseline, improved documentation, integration guidelines, and where appropriate repository or demo updates.
- **Pilot deliverables:** Validation in additional edge intelligence and privacy-preserving distributed training scenarios; proof-of-concept demonstrations in smart home environments.
- **Business outputs (offers, proposals):** Internal assessment for product/service integration, follow-up R&D proposals, and positioning of FLaaS as a reusable enabler for distributed AI services.

IP, licensing and access rights

Background IP needed

- **Your background needed:** Telefónica know-how in distributed systems, edge/cloud services, AI service integration, and software/service orchestration.
- **Third-party IP/licenses:** N/A

Foreground ownership

- Single partner-owned
- Joint ownership – partners:
- Unsure / to be agreed

Licensing intentions (software / docs)

- **Preferred licence (if open source):** MIT
- **Components intended to remain proprietary:** N/A
- **Constraints/notes (patents, NDAs, data restrictions):** FlaaS patent filed by telefonica, Privacy preserving FlaaS patent in filing.

Sustainability and maintenance (post-project)

What will you maintain after project end?

- **Components:** FlaaS Server and FlaaS Android repos
- **Documentation:** Included in the repos
- **Demo/benchmark artefacts:** Integration with internal unit product

Maintenance commitment

- Best-effort (no SLA)
- Active maintenance (issues + periodic releases)
- LTS commitment (duration: ____ months)
- Not planned (explain):

Governance and hosting preference

- TaRDIS consortium repository (shared)
- Partner-owned repository
- External foundation/community (specify):

KPIs and Measurable Evidence

Current metrics (to date)
<ul style="list-style-type: none"> ● # software releases: 1 ● # demos delivered: 1 ● # external stakeholders engaged: 0 ● Repo metrics (stars/forks/downloads/issues): N/A ● # pilots initiated: 0 ● Publications/citations (optional):
Expected metrics (next 12–24 months)
<ul style="list-style-type: none"> ● Target releases: 1 ● Target pilots: 1 ● Target adopters/users: N/A ● Target training/outreach: N/A
Quantified improvements (if available)
<p>If you used D7.1 quantification procedures or other metrics, summarise:</p> <ul style="list-style-type: none"> ● Metric(s): Accuracy, privacy budget ● Baseline: FLaaS without DP ● Result: Trade-off between privacy budget and accuracy (see Deliverables D7.3, D7.4, D7.5) ● Evidence link: D7.5

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Limited post-project resources for productisation or operational hardening 2. Integration complexity with real operational edge/cloud environments 3. Need for clearer market pull or internal business alignment for transition from prototype to service offering.
Mitigation actions
<ol style="list-style-type: none"> 1. Prioritise reuse in follow-up R&D and innovation activities 2. Consolidate documentation and modular interfaces to lower integration effort 3. Leverage joint exploitation and dissemination with partners where relevant.
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> ● Support for visibility, dissemination, and exploitation positioning ● Guidance on Horizon exploitation narratives, market positioning, and standardisation opportunities ● Collaboration with partners for joint demonstrations and cross-component integration

Partner Statement

Telefónica expects to exploit TaRDIS results primarily by advancing FLaaS as a reusable building block for privacy-preserving distributed intelligence in edge/cloud environments. The component is relevant to future telecom and digital-service (e.g., smart homes) scenarios where collaborative model training is needed without centralising sensitive data. Beyond its research value, FLaaS provides a basis for follow-up innovation, proofs of concept, and potential integration into future Telefónica service concepts related to edge AI and distributed analytics. Telefónica also sees value in extending FLaaS with lightweight AI techniques, such as knowledge distillation, to make federated learning more practical for constrained edge deployments. In this way, TaRDIS contributes to Telefónica's longer-term capabilities in distributed AI, privacy-preserving analytics, and intelligent edge services.

6.10 PARTNER: MARTEL INNOVATE (MTL)

Organisation name	Martel Innovate		Short	MTL
Contact person for exploitation	Name	Maria Chiara Campodonico	Partner Type	SME
	Email	mchiara.campodonico@martel-innovate.com		
TaRDIS use case(s)	<input type="checkbox"/> Telefónica <input type="checkbox"/> GMV <input type="checkbox"/> EDP <input type="checkbox"/> Actyx <input checked="" type="checkbox"/> Other/None: None			

Your Key Exploitable Results

- None

Concrete actions and milestones					
Milestone	Description	Owner	Target Date	Dependencies	Status
M1	Integration of TaRDIS methodologies into Martel's innovation management service portfolio	MARTEL	M6	Internal alignment, service design	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
M2	Leverage TaRDIS results and references in Horizon Europe and Digital Europe proposals	MARTEL	ongoing	Funding calls availability	<input type="checkbox"/> planned <input checked="" type="checkbox"/> ongoing <input type="checkbox"/> done
M3	Expansion of business network and partnerships via TaRDIS ecosystem	MARTEL	M12	Partner engagement continuity	<input checked="" type="checkbox"/> planned <input type="checkbox"/> ongoing <input type="checkbox"/> done
Expected outputs (what will be delivered)					
<ul style="list-style-type: none"> • Releases (software/docs): Methodological frameworks for dissemination and exploitation (internal use) • Pilot deliverables: • Business outputs (offers, proposals): <ul style="list-style-type: none"> ○ New consulting service packages (go-to-market, dissemination, exploitation) ○ Participation and coordination in future EU project proposals ○ Strategic partnerships and B2B collaborations 					

Sustainability and maintenance (post-project)

What will you maintain after project end?	
<ul style="list-style-type: none"> • Components: <ul style="list-style-type: none"> ○ Communication and dissemination frameworks ○ Stakeholder engagement methodologies • Documentation: <ul style="list-style-type: none"> ○ Reusable templates and guidelines • Demo/benchmark artefacts: N/A 	
Maintenance commitment	Governance and hosting preference
<input type="checkbox"/> Best-effort (no SLA) <input type="checkbox"/> Active maintenance (issues + periodic releases) <input type="checkbox"/> LTS commitment (duration: ____ months)	<input type="checkbox"/> TaRDIS consortium repository (shared) <input checked="" type="checkbox"/> Partner-owned repository <input type="checkbox"/> External foundation/community (specify):

<input type="checkbox"/> Not planned (explain):	
---	--

Risks, Blockers, and Support Needed

Top risks (max 3)
<ol style="list-style-type: none"> 1. Dependency on external funding cycles 2. Market competition in innovation consulting services.
Mitigation actions
<ol style="list-style-type: none"> 2. Focus on high-value consulting services and niche expertise 2. Leverage strong ecosystem positioning and partnerships 2. Continuously update methodologies based on project experience
Support needed from consortium / Horizon Results Booster / EC ecosystem
<ul style="list-style-type: none"> ● Continued collaboration within EU ecosystem initiatives ● Access to follow-up funding opportunities ● Engagement with Horizon Results Booster and EC networks

Partner Statement

Martel’s exploitation of TaRDIS focuses on transforming project experience into high-value consulting services and long-term strategic positioning within the European cloud–edge–IoT ecosystem. By leading dissemination and exploitation activities, Martel has developed advanced methodologies for stakeholder engagement, market alignment, and innovation uptake. These capabilities will be integrated into its service portfolio, enabling the company to support future EU projects, deep-tech start-ups, and industrial players. Leveraging its expanded network and proven track record, Martel will actively pursue new funding opportunities and partnerships, ensuring sustained impact and continued contribution to European innovation ecosystems.

7 EXPLOITATION ACTIVITIES

7.1 COMMUNITY BUILDING AND ECOSYSTEM ENGAGEMENT

TaRDIS has engaged in European ecosystem-building activities, including the co-organisation of the EU **Swarm Workshop 2025**¹² for two days starting on 2025-09-26 (see Figure 1), alongside other EC-funded projects in the Horizon Europe HORIZON-CL4-2022-DATA-01-03 swarm computing research cluster (OASEES, P2CODE, SmartEdge, and OpenSwarm). Such cluster activity directly supports exploitation by aligning technical directions across projects, creating shared audiences of potential adopters, and promoting reuse and interoperability across outcomes.



Figure 1: Banner on the EU Swarm Workshop 2025

Other similar events and details were reported in deliverable D8.5 ([14]).

7.2 DISSEMINATION INTEGRATED WITH EXPLOITATION

The TaRDIS deliverable D8.1 [12] sets out a strategy that explicitly aims to engage a critical mass of stakeholders and keep them engaged through continuous outreach, aligning dissemination actions with exploitation and sustainability objectives.

Besides this, TaRDIS maintains a public list of scientific publications¹³ related to the project's themes (distributed systems, protocols, verification/typing, decentralised intelligence, etc.), supporting research uptake.

¹² <https://swarmworkshop.eu>

¹³ <https://project-TaRDIS.eu/scientific-publications>

7.3 SUSTAINABILITY AND MAINTENANCE POST-PROJECT

TaRDIS sustainability focuses on ensuring that key software artefacts, documentation, and reproducibility assets remain accessible and maintainable beyond the project lifetime, with clear responsibilities and minimal dependency on project-only infrastructure.

Sustainability operating model

- **Ownership and stewardship:**
 - Public project-facing artefacts (deliverables, dissemination): TaRDIS website and public deliverables index, e.g.: <https://project-TaRDIS.eu/deliverables/>.
 - Each maintained repository has a declared owner/maintainer team (see section 6: partner plans).
- **Release management:** maintained components publish versioned releases and a compatibility note (API/toolbox/OS dependencies). Where applicable, APIs are aligned with the programming model and API specification (see **D3.5** [4]).
- **Minimum sustainability package per project-level KERs:**
 - A minimal reproducible example (MRE).
 - A deployment guide.
 - A validation/evidence note pointing to the final validation guidance (see D7.4 [11]).
 - A public reference to the relevant architecture and evaluation methodology (see D2.3 [3]) when applicable.
- **Security and vulnerability handling:** maintained repositories publish a security contact and a dependency update policy; for pilot-oriented components, producing a minimal Software Bill-of-Materials (SBOM) is recommended.
- **Archival and long-term access:** for critical releases and documentation snapshots, partners are encouraged to publish immutable archives (e.g., Zenodo DOI snapshots), complementing live repositories.

Continuity mechanisms

Continued development is supported by:

- Partner product integration and services (see section 8.2).
- Open ecosystem uptake (see section 8.1).
- Follow-on funding and pilots (see section 7.5).

Two transitions are critical to continuity at project end. First, the handover from project teams to maintainer teams: each KER specifies a named maintainer or maintaining organisation in its partner section, and the project repository index records that assignment so external users can locate the current owner after project closure. Second, the transition from project-funded effort to one of the three continuity routes above: partner integration absorbs effort into existing product roadmaps; ecosystem uptake distributes effort across external contributors; and follow-on funding restores dedicated effort under a new instrument. Where none of these routes is realistically available for a given asset, the consortium documents the asset as “archived” rather than “maintained,” with a Zenodo snapshot and a clear notice in the repository, to avoid misleading prospective adopters.

7.4 KPIs AND MEASURABLE EVIDENCE

TaRDIS tracks exploitation progress through a common set of indicators, complemented by KER-specific metrics reported by owners (Section 6). Indicators are grouped into: (i) adoption readiness, (ii) uptake, (iii) sustainability, and (iv) market/pilot traction.

Common indicators (project-level)

- **Adoption readiness:** number of versioned releases; availability of installation guides; MRE availability; presence of conformance tests; reproducible benchmark packages.
- **Uptake:** external pilots/PoCs; number of external adopters (identified organisations); repository activity metrics where applicable.
- **Sustainability:** named maintainers; maintenance commitment level; issue response process; release cadence.
- **Market traction:** LOIs, internal product integration decisions, service offerings published.

Evidence base

- Validation evidence and guidelines: D7.4 [11].
- Use-case development evidence: D7.3 [10].

7.5 EXPLOITATION THROUGH IMPROVING KERs IN FUTURE EC PROJECTS

For TaRDIS results at “Exploring” maturity, follow-on funding and pilot actions are core exploitation instruments to increase TRL, strengthen operational robustness, and reduce adoption risk. TaRDIS will use a structured continuation pipeline that maps candidate calls and funding instruments to KER maturation needs, specifies pilot commitments and KPI evidence requirements, and packages proposal-ready adoption assets (deployment recipes, reproducible benchmarks, and validation evidence) derived from D6.3 [8], D7.3 [10] and D7.4 [11].

Identification of Potential Future EC Calls for TaRDIS follow-up

For the short-term after the close-out of the TaRDIS project, a set of closely related calls were already identified (see Table 2) to ensure the project outcomes are properly followed, which will be described in the following subsections.

Table 2: Future EC calls for follow-up of TaRDIS research

Call / Topic ID	Type	Why	TaRDIS KER match
HORIZON-CL4-2027-04-DATA-03 “New approaches for decentralised, federated and sustainable AI data processing”	RIA	Direct continuation to harden decentralised training/privacy and address D7.1 constraints (privacy-by-design, trust assumptions, efficiency trade-offs)	KER-4 KER-2
HORIZON-CL4-2027-04-DATA-08 “Demand-side 3C pilot demonstrators on	IA	Pilot-driven route to operationalise TaRDIS runtime/tooling in telco-edge cloud settings; demonstrates	KER-1 KER-2 KER-4

Call / Topic ID	Type	Why	TaRDIS KER match
converged Telco Edge Cloud Infrastructure”		decentralised intelligence benefits with real KPIs	
HORIZON-CL4-2027-04-DIGITAL-EMERGING-05 “AI-Driven Robotics for Industry: Enabling System Integration and Adoption”	IA	Direct industrial adoption path for “robotics swarms”: integration + deployment + operations; clear exploitation and sustainability leverage	KER-1 KER-2 KER-4

TaRDIS will prioritise follow-on proposals that directly mature its four project-level KERs. In particular, **HORIZON-CL4-2027-04-DATA-03** provides a direct continuation route for Privacy Preserving and Efficient Decentralised Training, enabling TaRDIS to address constraints highlighted in D7.1 [9] (e.g., privacy-by-design gaps, trust assumptions in aggregation, with clear and measurable trade-offs) while scaling decentralised AI processing across heterogeneous infrastructures. In parallel, **HORIZON-CL4-2027-04-DATA-08** offers a pilot-driven path to validate TaRDIS swarm runtime/tooling and decentralised intelligence in converged Telco Edge Cloud infrastructures, strengthening adoption readiness through demonstrators with operational KPIs and stakeholder engagement beyond the consortium.

Finally, TaRDIS will also be interested in industry-facing follow-ups that translate swarm runtime/tooling into concrete deployment environments with clear exploitation routes. **HORIZON-CL4-2027-04-DIGITAL-EMERGING-05** is particularly aligned with “robotics swarms” adoption, enabling integration and operationalisation of decentralised intelligence and coordination in industrial multi-robot systems. Together, these three calls form a coherent continuation portfolio: maturing decentralised training and privacy features (RIA), validating runtime/tooling at scale in edge infrastructures (IA), and demonstrating industrial adoption in robotics integration contexts (IA), thereby maximising the long-term exploitation and sustainability of TaRDIS outcomes.

HORIZON-CL4-2027-04-DATA-03

Call & Type	HORIZON-CL4-2027-04-DATA-03 “New approaches for decentralised, federated and sustainable AI data processing” (RIA) ¹⁴
Opens	2026-11-17
Closes	2027-03-18
Best TaRDIS KER match	KER-4 Privacy-Preserving & Efficient Decentralised Training KER-2 data management
TRL Lift	TRL 4-5 to 5-6 (from prototype to relevant environment)

¹⁴ <https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/HORIZON-CL4-2027-04-DATA-03>

Consortium improvement	Privacy/cryptocurrency specialists
-------------------------------	------------------------------------

Concerning this call, **HORIZON-CL4-2027-04-DATA-03: New approaches for decentralised, federated and sustainable AI data processing**, TaRDIS has as strong attributes, among others, the KER for decentralised training / privacy-preserving learning. The main proposal idea shall be “Privacy-preserving decentralised training at the edge/swarms with measurable trade-offs (efficiency vs privacy), trust-minimising aggregation, and operational integration with swarm runtimes.”

TaRDIS has identified (see Innovation Radar, section 4.5) a key innovation on Privacy-Preserving and Efficient Decentralised Training, backed by runtime/toolbox work for decentralised systems (Babel ecosystem, monitoring/reconfiguration, decentralised data). This topic is a direct continuation path to mature TaRDIS decentralised training from “Exploring” maturity to higher-TRL prototypes/pilots through rigorous evaluation and sustainability-by-design.

TaRDIS is primed for this. Our work stands out on Innovation Radar for privacy-preserving, efficient decentralised training. We’ve put in serious runtime and toolbox engineering (think Babel ecosystem, dynamic monitoring, reconfiguration, and decentralised data handling). This opportunity lets us push TaRDIS further, taking it from experimental stages to mature, real-world pilots and higher-TRL prototypes, all while building in sustainability from the ground up.

Proposed Concept

We’re creating a next-generation stack for **privacy-preserving decentralised training across the cloud-edge-device continuum** that works smoothly across cloud, edge, and device layers, targeting settings where data cannot be centralised and nodes are heterogeneous and intermittently connected.

The future project will aim to deliver:

- **Trust-minimising training architectures** (e.g., hierarchical aggregation, decentralised coordination, split learning pathways for constrained devices).
- **Privacy-by-design mechanisms** integrated into training pipelines (differential privacy options, secure aggregation/robustness mechanisms where applicable).
- **Efficient decentralised training orchestration** under realistic constraints (bandwidth, energy).
- **A quantification framework** to report measurable trade-offs (accuracy, privacy, overhead, resilience).

TaRDIS differentiators

Most current approaches assume perfect networks. In contrast, TaRDIS embraces the messy reality: devices drop in and out, connections are flaky, hardware is all over the place, and the system structure isn’t neat or flat. Additionally, privacy and trust are centrally focused as engineering design goals, with clear, honest reporting on trade-offs. And with solutions that link directly into operational deployment, so they actually work in the field. Hence, we consider as TaRDIS differentiators:

- A move beyond “federated learning in ideal networks” to swarm/edge reality: partial connectivity, heterogeneity, hierarchical topologies.

- Treating privacy and trust as first-class engineering constraints; reports transparent trade-offs using measurement methodology (an approach already taken in TaRDIS D7.1 [9]).
- Integrating decentralised training with runtime/monitoring/reconfiguration for operational deployment (more than just algorithms).

KERs expected

- A plug-and-play decentralised training toolkit, composed by libraries, ready-to-use reference systems, and configurable privacy settings.
- Reference integrations with decentralised runtimes and data layers (pluggable and connectable into edge stacks).
- A reproducible benchmarking suite, evaluation artifacts, and practical “adoption playbooks” for pilot projects, extending the developer approach already reported on TaRDIS deliverable D3.6 [5].
- Optional support for commercial rollouts, through partner-led services or embedding straight into platforms.

Target stakeholders / adopters

- Edge AI platform vendors, telco edge operators, industrial IoT integrators and similar.
- Any organization needing strong privacy-preserving learning from distributed data: smart infrastructure, energy, mobility, manufacturing.
- Research groups and the open-source community in decentralised machine learning.

Pilot Plan

Running 2 or 3 real pilots to prove the technology in challenging environments, e.g., Decentralised, privacy-first training under real-world constraints, with hierarchical and split models for devices with minimal resources, requiring solid and measurable evidence on overhead and system resilience.

Consortium

The core TaRDIS group (experts in decentralised ML and runtime) shall lead the way, and the plan is including telecom and industrial adopters, researchers, privacy/security specialists, edge platform providers, and Testing & Experimentation Facility validation partners.

Risks & mitigations

- **Privacy overhead too high:** The team can report trade-offs and offer tailored configuration profiles.
- **Trust assumptions and models are unclear:** The project shall have an explicit threat model and show how the architecture shall address those threats.
- **Integration complexity:** TaRDIS has an already proven record of building modular interfaces and clear reference deployments that reduce complexity.

HORIZON-CL4-2027-04-DATA-08

Call & Type	HORIZON-CL4-2027-04-DATA-08 “Demand-side 3C pilot demonstrators on converged Telco Edge Cloud Infrastructure” (IA) ¹⁵
Opens	2026-11-17
Closes	2027-03-18
Best TaRDIS KER match	KER 1 Babel ecosystem KER 2 Data management KER 4 decentralised intelligence
TRL Lift	TRL 4-5 to 7-8 (from prototype to pilot operational demonstration)
Consortium improvement	Telco edge/cloud infrastructure providers, industry/public sector site owners

This call, **HORIZON-CL4-2027-04-DATA-08: Demand-side 3C pilot demonstrators on converged Telco Edge Cloud Infrastructure**, mentions decentralised intelligence and IoT-edge solutions in pilot demonstrators. This is a strong route to validate TaRDIS runtime/tooling in near-operational. TaRDIS has as strong points, among others, the KER for Babel ecosystem + monitoring/reconfiguration work. The main proposal idea shall be “Swarm-capable, correct-by-design runtime/tooling integrated into telco edge cloud stacks, with decentralised orchestration/observability and multi-device interoperability.”

TaRDIS provides a strong base for decentralised intelligence and swarm runtime/tooling: Babel ecosystem extensions, decentralised membership/communication primitives, decentralised storage (e.g., Nimbus/PotionDB), and monitoring/reconfiguration mechanisms. This topic is ideal to push TaRDIS components into near-operational pilots on telco edge cloud infrastructure.

Proposed Concept

The development and setup of pilots having swarm-native, decentralised intelligence applications on Telco Edge Cloud (the 3C vision¹⁶). According to that, the TaRDIS consortium can show how distributed applications can be developed across different edge devices, keeping them observable and safely reconfigurable at scale, while managing data and models across cloud and edge, and, most importantly, making central coordination mostly obsolete. The most important topics of the proposed concept are, therefore building and running demand-side pilot demonstrators that deploy swarm-native decentralised intelligence applications that demonstrate:

- Runtime portability and interoperability across devices/edge nodes.
- Observability and safe reconfiguration of decentralised applications at scale.
- Data and model distribution suited to the cloud-edge.

¹⁵<https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/horizon-cl4-2027-04-data-08>

¹⁶<https://digital-strategy.ec.europa.eu/en/news/commission-announces-eu75-million-euro-3c-project-build-federated-telco-edge-cloud-infrastructure>

- Decentralised intelligence behaviours that reduce reliance on central coordination while improving resilience.

TaRDIS differentiators

- TaRDIS is not proposing a proof-of-concept or a demo in a test lab but actually deploying decentralised intelligence against the telco edge constraints (network slicing, multi-domain, real SLA needs).
- The use of swarms and decentralised data/membership tools to deal with heterogeneity.
- Centring on real operations: telemetry for controlling and tweaking live systems.

KERs expected

- Pilot-validated runtime/tooling modules compatible with telco edge environments.
- A set of repeatable “deployment recipes” extending the ones described in TaRDIS deliverable D3.6 [5], which may include e.g., reference architectures, Infrastructure as Code (IaC) and containers, monitoring dashboards that are easy to pick up and execute.
- Evidence packages: KPIs on latency, resilience, operational cost, and decentralisation benefits.
- Strong path to exploitation through pilots: partner product integrations, service offerings, and follow-on adoption.

Target stakeholders / adopters

- Telco edge operators and infrastructure players.
- Industrial edge AI users, integrators, and anyone deploying at the edge, especially where autonomy is important.
- Public sector ops that need robust, decentralised systems

Pilot Plan

- Decentralised telemetry, monitoring, and control.
- Decentralised data sharing and swarm storage, even when devices churn.
- Optionally, throw in decentralised learning/model-sharing, connecting to DATA-03 initiatives.

Consortium

Telco partners, edge infrastructure providers, system integrators, and the TaRDIS tech team. We may consider adding more deployment-focused SMEs (like CMS) and pilot sites with real operational demands and measurable KPIs.

Risks & mitigations

- **Integrating with telco stacks is challenging:** The consortium will most likely need to clearly set the scope and reference architectures up front.
- **Security/compliance problems:** The project will need to align with telco-grade standards and monitoring from the start.
- **Project complexity is too high:** The pilots need to be tailored to a few tightly scoped, measurable scenarios.

HORIZON-CL4-2027-04-DIGITAL-EMERGING-05

Call & Type	HORIZON-CL4-2027-04-DIGITAL-EMERGING-05 “AI-Driven Robotics for Industry: Enabling System Integration and Adoption” (IA) ¹⁷ ,
Opens	2026-11-17
Closes	2027-03-18
Best TaRDIS KER match	KER 1 Babel ecosystem KER 2 Data management KER 4 decentralised intelligence
TRL Lift	TRL 4-5 to 7-8 (from prototype to pilot operational demonstration)
Consortium improvement	Robotics Original Equipment Manufacturers (OEMs), industrial end-users, safety/assurance partner

With respect to this call, **HORIZON-CL4-2027-04-DIGITAL-EMERGING-05: AI-Driven Robotics for Industry: Enabling System Integration and Adoption**, TaRDIS aims to make life easier for complex, mixed fleets of robots (and sensors, and edge devices). TaRDIS includes decentralised programming models, runtime and data tools, fuelled by AI and ML. It fits industrial robotics because the real problem in stake is getting all these different robots to work together, securely, reliably, and without drowning in operational overhead. The main proposal idea shall be “Multi-robot/swarm software stack for decentralised intelligence: runtime + data layer + privacy-preserving decentralised training, with observability and safe reconfiguration in industrial environments.”

Proposed Concept

The project will aim to deliver an integration/adoption package for AI-driven robot swarms in industry. The consortium will use decentralised intelligence to make robot teams more resilient and less dependent on central control. In detail, the main points are to:

- Plug TaRDIS runtimes and tools straight into industrial robotics stacks.
- Enable decentralised data sharing and team coordination for robots.
- Deliver strong ops support, monitoring and dynamic reconfiguration that’s designed for harsh industrial realities.
- Showcase applied AI in real-world jobs by inclusion of privacy-first, decentralised learning when it makes sense.

TaRDIS differentiators

- Considering the legacy of TaRDIS, the consortium will not be just dropping in another robotics algorithm. This project includes the whole stack, including tools, runtime, ops.

¹⁷<https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/HORIZON-CL4-2027-04-DIGITAL-EMERGING-05>

- The TaRDIS approach looks at robot coordination as a distributed systems' problem that needs strong observability, correctness, and resilience.
- The project outcomes are more than technological demos, they're hands-on deliverables like integration kits, architectures, and deployment playbooks.

KERs expected

- Middleware and tools that can be dropped into multi-robot systems.
- Reference designs for decentralised coordination/data exchange.
- Industrial demos, with clear KPIs for uptime, time-to-reconfigure, reduced operator effort, scalability.
- Commercial value for partners: integration services, new platform features, and training.

Target stakeholders / adopters

The target stakeholders shall be essentially robotics OEMs, industrial integrators, operators in warehouses, factories, inspection, and smart manufacturing.

Pilot Plan

The project aims to build at least two industry-grade demos, e.g., a robot fleet coordination in a warehouse, and maybe an inspection/maintenance swarm.

Pilots shall take into result in clear data: KPIs, integration cost/time, with impact clearly shown.

Consortium

Robotics vendors or integrators, real industrial users, edge platform providers, together with the TaRDIS stack owners.

Risks & mitigations

- **Operating with real industry poses serious safety risks:** This will require strong monitoring, conservative autonomy, and staged rollouts.
- **Integration challenges:** The TaRDIS consortium already has the experience of working with a real industrial customer on the Actyx pilot. The best way to address this is to standardize on architectures and build modular adapters.
- **Data privacy:** The consortium will provide support on-premises, privacy-respecting approaches from the start.

Strategy for Applying to Future EC Calls for TaRDIS follow-up

TaRDIS will treat follow-on proposals as structured exploitation actions, with clear gates and responsibilities:

1. Topic-to-KER mapping and prioritisation (portfolio gate)

- **Action:** Maintaining a live mapping between each candidate call topic and TaRDIS KERs, including which partners own/contribute to each KER and which adoption scenario each topic enables (RIA maturation vs IA pilots).
- **Output:** Maintain Table 2 up-to-date and reviewed quarterly.

2. Pilot and adopter identification (market/pilot gate)

- **Action:** For IA-oriented topics (e.g., telco edge pilots, robotics adoption), identify pilot sites and gather early commitment signals (e.g., letters of intent, pilot scope notes, constraints and KPIs).
 - **Output:** Pilot candidate registry, with KPIs aligned to the D7.1 [9] quantification procedures.
- 3. Technical packaging for proposal readiness (asset gate)**
- **Action:** Package exploitable components as proposal-ready assets: demo scripts, reproducible deployments, reference architectures, integration notes, and an evidence pack (benchmarks, demo videos, use-case progress).
 - **Output:** Build an “Adoption Kit” per KER, including current TRL and what work remains to reach pilot readiness.
- 4. Consortium building and role allocation (consortium gate)**
- **Action:** Confirm partner roles (technology owners, integrators, pilot operators, evaluation partners) and fill missing competencies:
 - Privacy/security/crypto for HORIZON-CL4-2027-04-DATA-03.
 - Ops/telco edge partners for HORIZON-CL4-2027-04-DATA-08.
 - Robotics OEM / end-users / safety companies for HORIZON-CL4-2027-04-DIGITAL-EMERGING-05.
 - **Output:** Draft consortium structure and partner exploitation commitments per proposal.
- 5. Proposal drafting, review, and submission (delivery gate)**
- **Action:** Produce proposal outlines that reuse and extend the TaRDIS exploitation headlines: KER-based value proposition, TRL-aware plan, quantified KPIs, sustainability plan, and exploitation routes.
 - **Output:** Submitted proposals and a documented “lessons learned” file for future iterations.

Roadmap for Applying to Future EC Calls for TaRDIS follow-up

The TaRDIS participation in Horizon Results Booster **Module E (Access to other funding & entrepreneurship support)** (see section 5) strengthened this continuation pipeline by:

- Supporting the identification of the most suitable calls and complementary funding instruments for each KER.
- Improving proposal-to-market alignment (which exploitation route is realistic for each KER).
- Supporting investor/accelerator pathways where relevant.
- Ensuring continuity planning and sustainability commitments are credible beyond the project end.

This creates a structured exploitation bridge from “Exploring” maturity to pilot-ready outcomes through follow-on funding and entrepreneurship support.

8 EXPLOITATION PATHWAYS AND TARGET USERS

8.1 DEVELOPER AND INTEGRATOR UPTAKE

The primary pathway for TaRDIS impact is expected to be the adoption of the programming model/toolbox/runtime by:

- Software engineers building decentralised edge/swarm applications
- System integrators deploying heterogeneous swarms
- Platform providers seeking higher-level programmability and correctness support

This pathway is supported by the project's explicit focus on a development environment and correctness-by-design analyses (D3.6 [5]) and the concrete description of model/API design and intended use in D3.5 [4].

8.2 PARTNER-DRIVEN COMMERCIALISATION AND SERVICES

Commercial exploitation is expected to occur primarily through:

- embedding TaRDIS components into partner platforms/products,
- offering integration, customisation, and support services,
- using use-case evidence to justify performance/correctness claims for customers.

The specific plans driven by individual partners were listed in section 6.

8.3 RESEARCH AND EDUCATION PATHWAY

The programming model, APIs, runtime ideas, and decentralised intelligence methods can be exploited via:

- publications + open artefacts,
- teaching materials and courses,
- follow-on projects and proposals.

9 INTELLECTUAL PROPERTY AND SUSTAINABILITY

The TaRDIS exploitation requires a balanced approach between open dissemination (to maximise uptake) and protection of partner-specific foreground (to enable commercial advantage). D8.1 [12] frames sustainability and exploitation as coordinated with outreach.

TaRDIS has established the technical and ecosystem foundations for exploitation: a defined programming model/API direction (D3.5 [4]), a release-driven tooling plan (deliverables list), community-building actions at European level (Swarm Workshop 2025, see section 5), and external visibility through Innovation Radar recognition (see section 4).

TaRDIS has delivered a coherent set of research and engineering outcomes that substantially reduce the effort and risk of building **trustworthy, resilient decentralised intelligence** across heterogeneous edge and swarm environments. The project results are backed by a complete chain of evidence, spanning:

- A consolidated architecture and evaluation methodology (**D2.3** [3]).
- A stable programming model and APIs supporting a language-independent, event-driven approach (**D3.5** [4]).
- Final results on distributed AI and AI-based orchestration (**D5.3** [7]).
- A final integrated platform/toolbox release covering communication, coordination, data management, observability, and execution across heterogeneous devices (**D6.3** [8]).
- End-to-end use-case development and validation, including integration outcomes and actionable guidelines (**D7.3** [10] and **D7.4** [11]).
- Dissemination and ecosystem building activities, consolidated in **D8.5** [14], further reinforce visibility and uptake potential beyond the project lifetime.

Exploitation readiness is demonstrated not only through documentation and software assets, but also through **concrete demonstrations at events such as the EC Swarm Workshop** (see section 7.1) and public showcases where the TaRDIS stack was executed in realistic settings, validating the feasibility of the approach and providing credible evidence of benefits and trade-offs. This is critical for software-intensive, multi-component innovations where adoption depends on operational confidence, reproducibility, and integration guidance as much as on novelty.

The roadmap's project-level Key Exploitable Results (KERs), aligned with the European Commission's Innovation Radar, are supported by clear exploitation pathways and sustainability commitments:

- **KER-1 (Babel Framework and Ecosystem)** shows particularly strong post-project momentum. Beyond technical validation and large-scale demonstrations, KER-1 is already driving the creation of a **dedicated spin-off pathway**, providing a credible mechanism for long-term stewardship, productisation, maintenance continuity, and market-facing service offerings. This significantly strengthens sustainability compared to typical research outputs that rely solely on best-effort academic maintenance.

- **KER-3 (Improved Actyx Middleware)** provides compelling evidence of industrial pull and technology transfer: the project results have been supported and evaluated with the involvement of **SW Machines GmbH**, an industrial actor **outside the TaRDIS consortium**, which engaged indirectly through Actyx AG. Importantly, SW Machines GmbH continued to support and assist with evaluation activities after Actyx AG left the consortium, evidencing the practical relevance of the outcomes and strengthening the credibility of TaRDIS exploitation beyond partner-internal intentions. This is a strong indicator of real-world interest, particularly in the manufacturing domain where operational constraints and reliability requirements are stringent.

Overall, TaRDIS concludes with a credible, TRL-aware exploitation and sustainability pathway: key results are packaged as reusable building blocks, validated through multiple heterogeneous use cases, complemented by guidelines for adoption, and reinforced by both **industrial uptake signals** and **continuity mechanisms** (including spin-off formation and external industrial support). The consortium is therefore positioned to maximise post-project impact through open ecosystem uptake, partner-driven product integration and services, targeted pilots, research and education transfer, and follow-on funding aimed at raising TRL and operational readiness while preserving the project's core goals of trustworthiness, resilience, and interoperability.

10 EXPLOITATION DEVIATIONS, BARRIERS AND MITIGATION

The following shows how TaRDIS addresses some of the typical risks and barriers for software-toolbox exploitation in research projects:

- Barrier:** tooling perceived as “research prototype only”
Mitigation: correctness-by-design focus, iterative releases, and use-case validation (definition of clear objectives on each TaRDIS deliverable and the inclusion of a deliverables structure).
- Barrier:** interoperability complexity across devices/languages
Mitigation: explicit project objective on interoperability and runtime support; validated via use cases.
- Barrier:** sustainability after project end
Mitigation: The inclusion of an explicit sustainability roadmap in this deliverable (D8.4) and an integrated outreach/exploitation planning (in deliverable D8.1 [12]).

10.1 SPECIFIC EXPLOITATION RISKS PER INNOVATION RADAR KER

Innovation Radar KER	Exploitation Risk	Evidence	Mitigation
Babel Framework	Fragmentation of ecosystem integrations	Typical for frameworks; depends on plugin strategy	Clear governance + stable APIs + reference integrations
Decentralised Data management framework	Performance/resilience claims not transferable across environments	Needs robust benchmarks	Use D7.1 [9] procedures; publish reproducible benchmarks
Improved Actyx Middleware	Dependency and licensing alignment; integration complexity	Middleware integration risk	Modular packaging; clear compatibility matrix
Privacy-Preserving decentralised training	Privacy-by-design and trust concerns; overhead	D7.1 [9] highlights privacy/trust issues	Threat model + trust-minimising architecture; measured trade-offs

10.2 TRL-AWARE EXPLOITATION ROADMAP

Given the early-stage maturity of the project’s innovations (Innovation Radar ‘Exploring’ category), TaRDIS exploitation is primarily focused on technology transfer and adoption readiness: delivering reusable modular building blocks, validated demonstrators, integration

assets, and a quantification framework that enables credible assessment of benefits and trade-offs. Commercial-scale deployment is not expected within the project timeframe; instead, TaRDIS establishes the technical and operational foundations required for industrial pilots and follow-on productization.”

Having a low TRL doesn't make exploitation impossible, but it changes the concept of “exploitation”. For TaRDIS, the Innovation Radar classification (see section 4.5) is essentially an external signal of that same reality: early-stage innovation with promise and commitment, not yet product-grade deployment at scale.

How Having a Low TRL Changes the Exploitation Objective

Considering that TaRDIS KERs still have a relatively low TRL, exploitation should be framed less as “commercial roll-out” and more as **technology transfer and adoption readiness building**, with three near-term exploitation goals:

1. De-risking adoption

The key deliverable for adopters is not a finished product; it's credible evidence that the approach works and is worth further investment. TaRDIS is already doing the right kind of work here: WP6 provides runtime building blocks and demonstrators (Messaging/Voting apps) and integration into use cases (Telefónica/GMV/EDP/Actyx), which are classic low-TRL de-risking assets.

2. Creating reusable building blocks (not monoliths)

Low TRL is much easier to exploit when outputs are modular: protocols, APIs, libraries, adapters, reference implementations. D6.2 shows exactly this kind of modularity (HyParView variants, gossip/flood broadcast, anti-entropy, PotionDB views, Nimbus storage, telemetry/reconfiguration). Modular building blocks can be exploited earlier via reuse in other projects and prototypes, even if they're not yet “enterprise ready”.

3. Positioning for follow-on funding and industrial pilots

A realistic exploitation target is conversion into pilots and follow-on productization funding. That can be Horizon follow-ups, national programmes, or partner-funded maturation. The exploitation report should therefore treat low TRL as a **roadmap**: “here is what exists now; here is what must be hardened; here is who will do it.”

What a Low TRL specifically impacts for TaRDIS

Adoption barriers increase (and shift)

For systems like TaRDIS (distributed runtime + data layer + decentralised ML), low TRL typically means:

- **Operational risk is high:** adopters worry about monitoring, failure modes, upgrades, debugging, and security hardening. WP6's monitoring and namespace-based reconfiguration work directly targets this (telemetry, aggregation, reconfiguration engine) (see D6.3 [8]).
- **Integration challenges:** early adopters don't want to rewrite their stack. The adapters and unified abstraction work (Cassandra/Hyperledger Fabric etc.) are exploitation enablers precisely because they reduce switching costs (see D6.3 [8]).

- **Trust/security/privacy gaps:** in decentralised ML, the adoption decision is often blocked by privacy-by-design and trust in aggregation/control points. D7.1 explicitly flags these as challenges (e.g., differential privacy noise injection and central aggregator trust).

Impact claims

Despite the KERs' low TRL, deliverable D7.1 [9] is an important exploitation instrument, as it defines **how** improvements will be quantified and documents known limitations and assumptions. That allows TaRDIS to claim as big exploitation achievements:

- The establishment of a quantification methodology and baselines.
- Demonstrating feasibility in heterogeneous demos and in-use-case integration.
- Identifying the remaining blockers for real-world adoption, which allows the consortium to have a plan.

TaRDIS Strategy to handle Low-TRL Exploitation

With outputs which are strong but still research-grade, the strongest exploitation channels that TaRDIS is proposing are:

1. **Industrial technology transfer inside partners:** WP6 work is influencing the proprietary Actyx framework (reliable/durable event broadcast + durable event storage). That is a clear sign of exploitation even if code is not public, because it represents the project adoption into a market-facing product line.
2. **Open-source reference implementations and community reuse:** Low TRL tools can still be heavily exploited if they become the reference implementation people build on. TaRDIS has all its public open-source code available on the project repositories on <https://codelab.fct.unl.pt/di/research/TaRDIS>.
3. **Validated demonstrators as “adoption assets”:** The project already showed numerous (see section 7.2) Messaging/Voting demos on heterogeneous devices. This is the sort of artefacts that convinces engineers and product owners the approach is workable.
4. **Pre-standardisation influence:** At low TRL, it's often premature to standardise, but the project can influence emerging best practices through APIs, patterns, and interoperability conventions (especially around swarm programming abstractions and decentralised data/ML). This precisely was reported on project Deliverable D8.5 [14].

Component-level impact of low TRL and exploitation consequences

This subsection aligns low-TRL exploitation implications explicitly to each Innovation Radar component and to WP6/D7.1 evidence.

Babel Framework and ecosystems (KER-1)

Low-TRL exploitation implications:

Adoption depends on stability of interfaces, clarity of integration patterns, and demonstrated portability across device classes. At low TRL, adopters will often trial Babel as a framework layer only if it is easy to integrate and comes with strong examples, rather than as a “full platform replacement”.

Evidence of readiness-building:

WP6 extended the Babel ecosystem with **Babel-Swarm** (autonomic behaviours) and **Babel-Android** (mobile support), and integrated multiple protocol implementations (membership, broadcast, anti-entropy), improving composability for heterogeneous swarms (see section 4.1 and deliverable D6.3 [8]).

Exploitation strategy considering a low TRL:

- Focus on “developer onboarding” assets: reference apps, integration guides, compatibility matrices, and repeatable demo deployments.
- Encourage uptake through research/community channels and early adopter pilots, not broad commercial roll-out.

Decentralised Data management framework (KER-2)

Low-TRL exploitation implications:

Data-layer components face strong adoption scrutiny: consistency guarantees, failure behaviour, replication overhead, and operational tooling. At low TRL, the exploitation risk is not only performance, but misapplication (using a geo-distributed transactional design in a highly dynamic swarm scenario, or vice-versa).

Evidence of readiness-building:

WP6 provides two complementary designs, PotionDB and Nimbus (see section 4.2 and deliverable D6.3 [8]).

Exploitation strategy considering a low TRL:

Provide scenario-based adoption guidance (“when to use PotionDB vs Nimbus”), plus reproducible benchmarks and evaluation evidence as soon as available (see deliverable D6.3 [8]).

Improved Actyx Middleware (KER-3)

Low-TRL exploitation implications:

Even though the overall project outputs remain research-grade, exploitation can be considered very strong in this KER as there are reports of a real industry customer integrating these results into a product. For proprietary middleware, exploitation evidence shifted from open release metrics to technology transfer evidence.

Evidence of direct industrial uptake:

WP6 and WP7 (see Deliverable D7.3 [10]) reported exploring improvements to Actyx to provide reliable and durable event broadcast and durable storage of propagated events, and notes that Actyx has already imported some WP6 ideas/work.

Exploitation strategy considering a low TRL:

This technology transfer is one of the project’s most strong exploitation outcomes (know-how and design integration), despite the resulting artefacts are not openly released (proprietary). Deliverables D7.3 and D7.5 show evidence of this technology uptake.

Privacy-Preserving and Efficient Decentralised Training (KER-4)

Low-TRL exploitation implications:

Decentralised training is often blocked not by algorithmic feasibility but by privacy, trust, and compliance requirements. Low TRL implies that privacy-by-design and trust-minimising

architectures may not yet be fully implemented or validated, limiting near-term exploitation to pilots and controlled environments.

Evidence of constraints and roadmap drivers:

D7.1 [9] explicitly highlights privacy-by-design limitations in federated learning services, and the need to inject differentially private noise to protect against adversarial attacks, and the continuing trust role of central aggregators.

Evidence of enabling infrastructure and integration direction:

WP6 progressed telemetry acquisition, aggregation, telemetry-driven control, and ML-oriented control endpoints, and reports collaboration with Telefónica to enhance FLaaS with decentralised interactions, hierarchical aggregation, and split learning for constrained devices. Additionally, these outcomes are planned to be exploited directly by pilot partner Telefonica (TID).

Exploitation strategy considering a low TRL:

Here the best exploitation strategy is to keep working on developments of the FLaaS platform, and support TID into promoting the KER internally at Telefonica while expecting to increase the TRL of this KER, possibly through another funding project (see section 0).

11 CONCLUSION

TaRDIS has established the technical and ecosystem foundations for exploitation: a defined programming model/API direction (D3.5 [4]), a release-driven tooling plan (deliverables list), community-building actions at European level (Swarm Workshop 2025, see section 5), and external visibility through Innovation Radar recognition (see section 4).

TaRDIS has delivered a coherent set of research and engineering outcomes that substantially reduce the effort and risk of building **trustworthy, resilient decentralised intelligence** across heterogeneous edge and swarm environments. The project results are backed by a complete chain of evidence, spanning:

- A consolidated architecture and evaluation methodology (**D2.3** [3]).
- A stable programming model and APIs supporting a language-independent, event-driven approach (**D3.5** [4]).
- Final results on distributed AI and AI-based orchestration (**D5.3** [7]).
- A final integrated platform/toolbox release covering communication, coordination, data management, observability, and execution across heterogeneous devices (**D6.3** [8]).
- End-to-end use-case development and validation, including integration outcomes and actionable guidelines (**D7.3** [10] and **D7.4** [11]).
- Dissemination and ecosystem building activities, consolidated in **D8.5** [14], further reinforce visibility and uptake potential beyond the project lifetime.

Exploitation readiness is demonstrated not only through documentation and software assets, but also through **concrete demonstrations at events such as the EC Swarm Workshop** (see section 7.1) and public showcases where the TaRDIS stack was executed in realistic settings, validating the feasibility of the approach and providing credible evidence of benefits and trade-offs. This is critical for software-intensive, multi-component innovations where adoption depends on operational confidence, reproducibility, and integration guidance as much as on novelty.

The roadmap's project-level Key Exploitable Results (KERs), aligned with the European Commission's Innovation Radar, are supported by clear exploitation pathways and sustainability commitments:

- **KER-1 (Babel Framework and Ecosystem)** shows particularly strong post-project momentum. Beyond technical validation and large-scale demonstrations, KER-1 is already driving the creation of a **dedicated spin-off pathway**, providing a credible mechanism for long-term stewardship, productisation, maintenance continuity, and market-facing service offerings. This significantly strengthens sustainability compared to typical research outputs that rely solely on best-effort academic maintenance.
- **KER-3 (Improved Actyx Middleware)** provides compelling evidence of industrial pull and technology transfer: the project results have been supported and evaluated with the involvement of **SW Machines GmbH**, an industrial actor **outside the TaRDIS consortium**, which engaged indirectly through Actyx AG. Importantly, SW Machines GmbH continued to support and assist with evaluation activities after Actyx AG left the consortium, evidencing the practical relevance of the outcomes and strengthening the credibility of TaRDIS exploitation beyond partner-internal intentions. This is a strong indicator of real-world interest, particularly in the manufacturing domain where operational constraints and reliability requirements are stringent.

Overall, TaRDIS concludes with a credible, TRL-aware exploitation and sustainability pathway: key results are packaged as reusable building blocks, validated through multiple heterogeneous use cases, complemented by guidelines for adoption, and reinforced by both **industrial uptake signals** and **continuity mechanisms** (including spin-off formation and external industrial support). The consortium is therefore positioned to maximise post-project impact through open ecosystem uptake, partner-driven product integration and services, targeted pilots, research and education transfer, and follow-on funding aimed at raising TRL and operational readiness while preserving the project's core goals of trustworthiness, resilience, and interoperability.

REFERENCES

- [1] TaRDIS Consortium. “Deliverable 1.3: Final report on the data management of the project”, March 2026, <https://project-TaRDIS.eu/download/d1-3-data-management-plan>.
- [2] TaRDIS Consortium. “Deliverable 2.2: Report on overall requirement analysis”, February 2024, <https://www.project-TaRDIS.eu/download/d2-2-report-on-overall-requirements-analysis>.
- [3] TaRDIS Consortium. “Deliverable 2.3: Report on architecture specification and evaluation methodology”, June 2024, <https://www.project-TaRDIS.eu/download/d2-3-TaRDIS-architecture-definition-and-technical-specifications>.
- [4] TaRDIS Consortium. “Deliverable 3.5: Third Report on Programming Model and APIs”, September 2025, <https://project-TaRDIS.eu/download/d3-5-third-report-on-programming-model-and-apis>.
- [5] TaRDIS Consortium. “Deliverable 3.6: Final release of TaRDIS development environment”, March 2026, <https://project-TaRDIS.eu/download/d3-6-final-release-of-TaRDIS-development-environment>.
- [6] TaRDIS Consortium. “Deliverable 4.3: Report on the final analyses’ toolset”, March 2026, <https://project-TaRDIS.eu/download/d4-3-final-report-on-toolset>.
- [7] TaRDIS Consortium. “Deliverable 5.3: Final report on distributed AI and AI-based orchestration”, March 2026, <https://project-TaRDIS.eu/download/d5-3-final-report-on-distributed-ai-and-ai-based-orchestration>.
- [8] TaRDIS Consortium. “Deliverable 6.3: Report and prototype of the final iteration of the platform”, March 2026, <https://project-TaRDIS.eu/download/d6-3-report-on-the-final-iteration-of-TaRDIS-toolbox-components>.
- [9] TaRDIS Consortium. “Deliverable 7.1: Report on the expected improvements and quantification procedures”, September 2023, <https://www.project-TaRDIS.eu/download/d7-1-report-on-the-expected-improvements-and-quantification-procedures>.
- [10] TaRDIS Consortium. “Deliverable 7.3: Report on development of the use cases with TaRDIS toolbox”, February 2026, <https://project-TaRDIS.eu/download/d7-3-report-on-development-of-the-use-cases-with-TaRDIS-toolbox>.
- [11] TaRDIS Consortium. “Deliverable 7.4: Report on the final validation of the toolbox and guidelines”, February 2026, <https://project-TaRDIS.eu/download/d7-4-report-on-the-final-validation-of-the-toolbox-and-guidelines>.
- [12] TaRDIS Consortium. “Deliverable 8.1: Dissemination and communication strategy and plan”, March 2023, <https://www.project-TaRDIS.eu/download/d8-1-dissemination-and-communication-strategy-and-plan>.
- [13] TaRDIS Consortium. “Deliverable 8.2: Exploitation and Sustainability Strategy and Plan”, August 2024
- [14] TaRDIS Consortium. “Deliverable 8.5: Outreach activities final report”, March 2026, <https://project-TaRDIS.eu/download/d8-5-outreach-activities-final-report>.
- [15] Statista, “Internet of Things (IoT) in Europe - statistics & facts”, <https://www.statista.com/topics/4123/internet-of-things-iot-in-europe>, accessed in 2026-02-10