



Funded by
the European Union

Horizon Europe

EUROPEAN COMMISSION

European Climate, Infrastructure and Environment Executive Agency (CINEA)

Grant agreement no. 101160684



Use of open-source P2P energy sharing platforms for energy Democratization

Deliverable D 3.1

U2Demo architecture definition and guidelines

Document Details

Due date	31-04-2025
Actual delivery date	31-04-2025
Lead Contractor	Exaion / Energy Web
Version	1.0
Prepared by	Nitin Gavhane (Energy Web)
Reviewed by	Gilles Deleuze (EXAION) & Glenn Reynders (Ku Leuven)
Dissemination Level	Public

Project Contractual Details

Project Title	Use of open-source P2P energy sharing platforms for energy democratization
Project Acronym	U2Demo
Grant Agreement No.	101160684
Project Start Date	01-09-2024
Project End Date	29-02-2028
Duration	42 months

Document History

Version	Date	Contributor(s)	Description
0.1	07/04/2025	Energy Web	Draft for architecture definition & guidelines
0.2	17/04/2025	EXAION, Ku Leuven	Deliverable revision
1.0	30/04/2025	Energy Web	Deliverable finalisation

Disclaimer

This document has been produced in the context of the U2Demo¹ project. Views and opinions expressed in this document are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgment

This document is a deliverable of U2Demo project. U2Demo has received funding from the European Union's Horizon Europe programme under grant agreement no. 101160684.



**Funded by
the European Union**

¹ <https://u2demo.eu/>

Executive Summary

This deliverable is the first of Work Package (WP) 3 related to the development of the U2Demo platform. The deliverable provides a global architecture definition and guidelines for the U2Demo platform, aiming to facilitate the development and integration of innovative peer-to-peer energy trading and sharing solutions. The document presents a high-level architecture overview, articulates critical design principles and technical specifications, and outlines essential considerations regarding security, privacy, and implementation guidelines. The ultimate objective is to establish a robust, scalable, and secure open-source foundation for subsequent project phases.

This deliverable is critical for the following activities of WP3 but also for the definition of the requirements that should be considered in the development of algorithms in WP4. However, the architecture presented in this document can evolve to accommodate all the need of algorithms.

Table of Contents

Executive Summary	4
Table of Contents	5
List of Figures.....	6
Acronyms	7
1 Introduction	8
1.1 Scope and Objectives	8
1.2 Structure of the deliverable.....	8
2 U2Demo Platform Architecture Overview	9
2.1 High-level Architecture Diagram.....	9
2.2 Components and Interactions.....	11
2.2.1 Digital Spine Data Exchange.....	11
2.2.1.1 The Energy Web Chain (EWC)	12
2.2.1.2 EW Self-Sovereign Identities Hub (SSI-Hub).....	12
2.2.1.3 Authentication Strategies between the Client GW and the Message Broker	15
2.2.1.4 DDHub Client Gateway	16
2.2.2 Information Exchange requirements.....	19
3 Design Principles and Requirements	20
3.1 Functional Requirements	20
3.2 Non-functional requirements	20
Standards and.....	21
3.3 Interoperability.....	21
4 Technical Requirements	22
4.1 Microservices Definition	22
4.2 Blockchain/Distributed Ledger Technology Integration	22
4.3 APIs and Data Structures.....	22
5 Security and Privacy Considerations.....	24
5.1 Data Management and Privacy Compliance.....	24
5.2 Cybersecurity Guidelines	24
6 Implementation Guidelines.....	26
6.1 Development Practices and Standards.....	26
6.2 Recommended Tools and Technologies	26
References.....	28
Appendix A.....	29
Table A1: Technology Stack Responsibilities per Component	29
Table A2: Deployment Model Options.....	30

List of Figures

Figure 1: Digital Spine High Level Integration Diagram	11
Figure 2: SSI-Hub Entity Hierarchy.....	13
Figure 3: Sample Role Associations	14
Figure 4: Authentication Strategies between the Client GW and the Message Broker	15
Figure 5: DDHub Client GW User Interface.....	16
Figure 6: Sample Client GW Setup by Organization	16
Figure 7: DDHub Client GW Screen for RESTful and WSS APIs	17
Figure 8: RESTful and WSS API Documentations	17

Acronyms

AI	Artificial Intelligence
CINEA	European Climate, Infrastructure and Environment Executive Agency
CI/CD	Continuous Integration/Continuous Delivery
CSDM	Common Semantic Data Model
D&C	Dissemination and Communication
DDHub	Decentralized Data Hub
DID	Decentralized Identifiers
DL	Description Language
ECs	Energy Communities
ENISA	European Union Agency for Cybersecurity (Former: European Network and Information Security Agency.)
EU	European Union
EVM	Ethereum Virtual Machine
EWC	Energy Web Chain
EWT	Energy Web Token
FR	Functional Requirements
GDPR	General Data Protection Regulation
GW	Gateway
HSMs	Hardware Security Modules
IAM	Identity & Access Management
IPFS	InterPlanetary File System
IT	Information Technology
KPI	Key Performance Indicators
M	Month
mTLS	Mutual Transport Layer Security
NIS2	Network and Information Security 2
NFR	Non-Functional Requirements
NGSI-LD	Next Generation Service Interface with Linked Data
OWL	Web Ontology Language
P2P	Peer-to-peer
REST	Representational State Transfer
SDK	Software Development Kit
SI	Standards and Interoperability
SSI	Self-Sovereign Identities
TTL	Time to live
VC	Verifiable Credentials
W3C	World Wide Web Consortium
WP	Work Package

1 Introduction

1.1 Scope and Objectives

The U2Demo project aims to democratize energy markets through an innovative and open-source peer-to-peer (P2P) trading platform and energy-sharing technologies. This document specifically addresses the global architecture definition necessary to support the development of the U2Demo platform.

This deliverable provides a comprehensive overview of the architecture specifications, design guidelines, and technical framework required for the successful development, deployment, and operation of the U2Demo platform. It includes detailed descriptions of system architecture, including modular components and layers, all tailored to support a distributed, P2P energy sharing ecosystem. The document will be complemented by Deliverable 3.2 where the main challenges related with the use of standards and interoperability will be addressed.

Furthermore, this deliverable outlines the software development methodologies, toolchains, and integration strategies to be adopted throughout the project lifecycle, with a focus on maintaining openness, transparency, and community-driven collaboration. It addresses key technical decisions such as the selection of blockchain or distributed ledger technologies, smart contract design, user interface considerations, and API standards to facilitate third-party extensions and integration with smart grid infrastructures.

The deliverable serves as a foundational reference for all project stakeholders—including developers, system architects, policy advisors, community partners, and deployment teams—ensuring alignment in vision, technical execution, and long-term sustainability. It is intended not only as a blueprint for current development efforts but also as a living document that can evolve alongside the platform to support future innovation and broader adoption.

1.2 Structure of the deliverable

After this introductory section, Section 2 presents the high-level architecture of U2Demo platform and the main components. Section 3 presents the design principles and the functional and non-functional requirements, including interoperability. The technical requirements are detailed in Section 4 mainly in aspects related to microservices, blockchain and APIs. Section 5 addresses aspects related with security and privacy and Section 6 the main guidelines and drivers for implementation.

2 U2Demo Platform Architecture Overview

This section describes the high-level architecture envisioned for the U2Demo platform. It aims to clearly present how different components and technologies interact and integrate within the overall system. This platform should consider the requirements and architectures used in Energy Communities (ECs) [1]. ECs are collective energy initiatives in which citizens, local authorities, and small businesses cooperate to produce, consume, store, and sell renewable energy. Enabled by European legislation such as the Renewable Energy Directive (EU) 2018/2001 [2] and the Electricity Market Directive (EU) 2019/944 [3], these communities aim to decentralize the energy system, increase local resilience, and empower consumers. By promoting renewable energy and local participation, ECs contribute to decarbonization and democratization of the energy sector [4]. They also offer socio-economic benefits such as lower energy bills, increased awareness of sustainability, and stronger community ties.

Governance models for ECs vary widely, depending on factors such as ownership structure, decision-making processes, and the roles of different stakeholders. Common models include cooperative structures, public-private partnerships, and hybrid arrangements that integrate multiple actors. Effective governance ensures transparency, inclusivity, and fair distribution of benefits and responsibilities [5]. Several distributed energy resources can be managed inside the ECs resulting in a very complex problems [6] that should be solved to achieve effective advantages for the members of the EC.

2.1 High-level Architecture Diagram

The diagram **presented in Figure 1 illustrates the high-level** overview of the Energy Web Digital Spine architecture and how the core components of the U2Demo platform interact with each other. It highlights the data flows and integration points between elements such as the Decentralized Data Hub (DDHub) Client Gateway, Message Broker, SSI-Hub, and participant systems. A detailed description of each component and its role in the architecture is provided in Section 22.2.

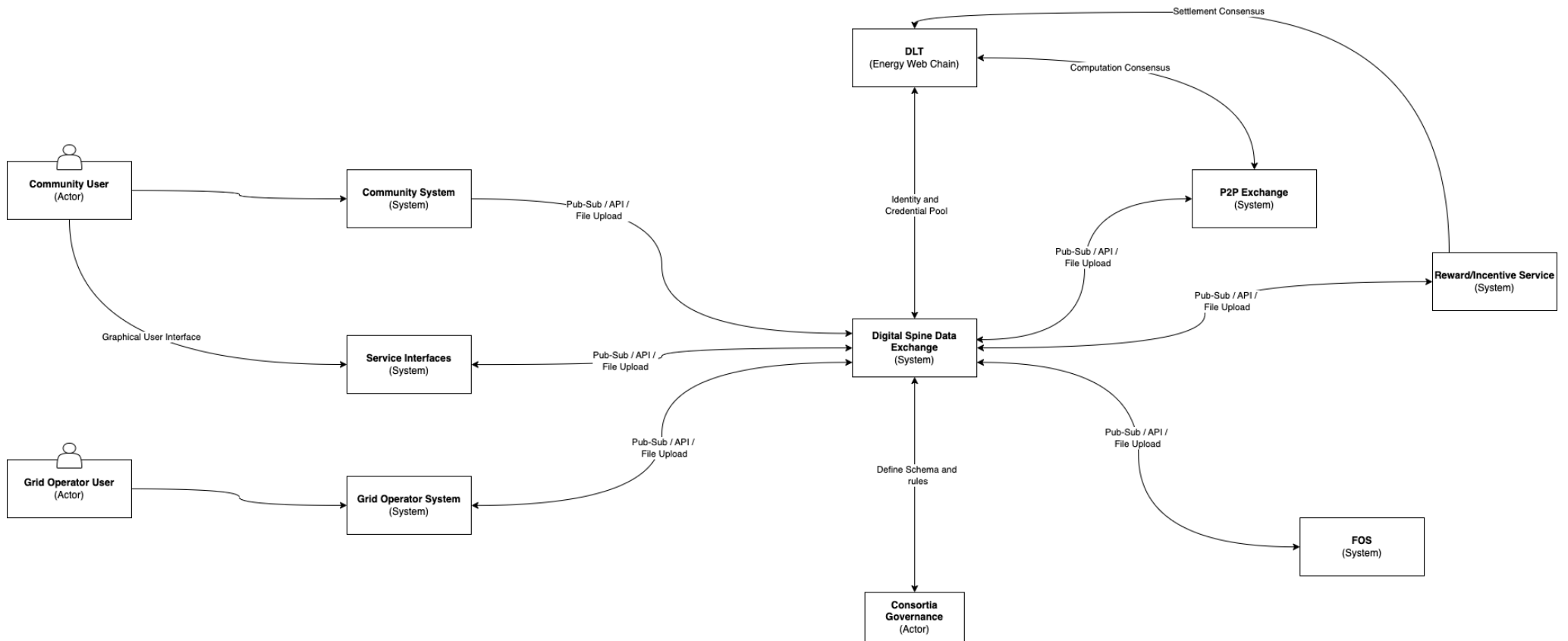


Figure 1: Overall Architecture

2.2 Components and Interactions

Description of key platform components, such as blockchain infrastructure, Identity and credentials management, P2P trading algorithms, and analytics services. It highlights how these elements communicate, the protocols used, and the data flows involved.

2.2.1 Digital Spine Data Exchange

The Digital Spine, presented in Figure 2 is a decentralized data exchange solution by Energy Web which helps facilitate the transfer of business-critical information among participants in a consortium or network to orchestrate varying business solutions. Also, Digital Spine is fully compliant with open-source requirements. All core components, such as DDHub, the SSI-Hub, and the associated Software Development Kit (SDKs), are published under permissive open-source licenses, ensuring transparency, accessibility, and extensibility for consortium partners and third-party developers. The architecture encourages modular integration and adheres to community-driven development practices.

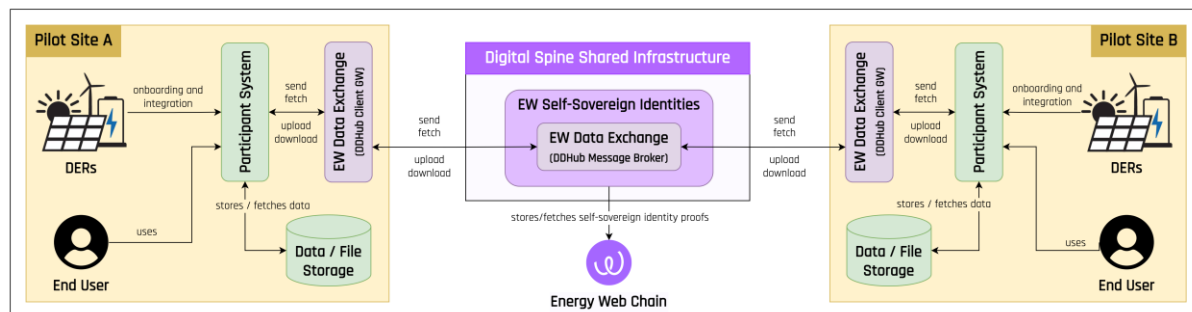


Figure 2: Digital Spine High Level Integration Diagram

The Digital Spine was selected as the foundational architecture due to its proven alignment with U2Demo’s core goals: decentralized data exchange, secure identity management, and interoperability. Built by Energy Web and adopted in multiple industrial energy applications, the Digital Spine offers a robust, open-source framework that supports General Data Protection Regulation (GDPR) compliant data handling, peer-to-peer communication, and modular integration with external systems. Its native support for Self-Sovereign Identities (SSI) and Verifiable Credentials (VCs) makes it the most suitable and future-proof infrastructure for enabling trusted interactions between diverse actors in energy communities.

The design choices of the Digital Spine are grounded in decentralization, data sovereignty, and interoperability, principles that are core to the U2Demo project. Each participant maintains full control over the data they produce and consume, with the flexibility to store this data in their preferred systems. This enables seamless integration with existing Information Technology (IT) environments and supports diverse operational contexts across pilot sites. By anchoring only identity-related metadata (such as Decentralized Identifiers (DIDs) and credential proofs) on-chain, the architecture remains lightweight and efficient while enabling a

“trust-but-verify” or even “trustless” model of interaction. In this setup, participants can exchange data securely without needing to trust a central intermediary, as authenticity and authorization are cryptographically verifiable. A key architectural principle is the separation of concerns: the Digital Spine facilitates secure, standardized, and verifiable data exchange, while the P2P exchange and other domain-specific applications remain focused on their own specialized logic, such as market operation, community coordination, or service delivery. This modularity allows each subsystem to function independently within its area of expertise, enhancing flexibility, robustness, and long-term maintainability. Digital Spine was selected because it offers a mature, modular, and open-source foundation specifically designed for decentralized energy ecosystems. Its alignment with SSI standards and successful deployment in real-world energy use cases make it a clear and future-ready choice for U2Demo.

2.2.1.1 The Energy Web Chain (EWC)

The Energy Web Chain (EWC) is a public, open-source, enterprise-grade blockchain built on the Ethereum Virtual Machine (EVM) and specifically designed for applications in the energy sector. It provides a decentralized infrastructure to support innovative energy solutions, particularly those involving digital identity and asset management. The native cryptocurrency, the Energy Web Token (EWT), facilitates on-chain transactions and interactions within the network.

EWC serves as the foundational layer for managing self-sovereign identities (SSIs) in the Digital Spine framework. Decentralized Identifiers (DIDs), along with their associated public keys, hashes of DID documents, and proofs of verifiable credentials, are securely stored on the blockchain. Meanwhile, the actual DID documents—containing the descriptive data linked to each identity—are stored off-chain using the InterPlanetary File System (IPFS), a decentralized, peer-to-peer file storage protocol. It’s important to note that the EWC is exclusively used for identity-related data, and no other types of information are stored on the blockchain.

EWC is the backbone of self-sovereign identities in Digital Spine. DIDs, their respective public key, hash of their DID document, and the proof of their verifiable credentials are stored on-chain. However, the DID document which contains the actual information associated with the DID is stored on IPFS (InterPlanetary File System), a decentralized, peer-to-peer file sharing system.

2.2.1.2 EW Self-Sovereign Identities Hub (SSI-Hub)

The EW SSI-Hub is a decentralized, digital identity solution that provides seamless identity and access management features through DIDs and VCs. This allows entities to have full control over their identity-related data and how it’s shared, rather than relying on centralized authorities.

In Digital Spine, DIDs identify data senders and recipients while VCs determine the roles a DID possess in the organization and/or application-level business perspective. Data senders and recipients are not end-users, rather they represent an organization or a specific participant system of an organization in the perspective of the data exchange network.

"Each identity can be assigned roles during enrollment through the EW Switchboard. Both the identity and the cryptographic proof of the assigned roles are stored on-chain. Currently, only identity-related metadata is recorded on the blockchain, while any data transmitted or received via the data exchange remains entirely separate from the blockchain infrastructure.

Figure 3 shows how roles are associated and defined for their respective organizations and applications. This hierarchy can be created using EW Switchboard. For the U2Demo project, this hierarchy is yet to be defined.

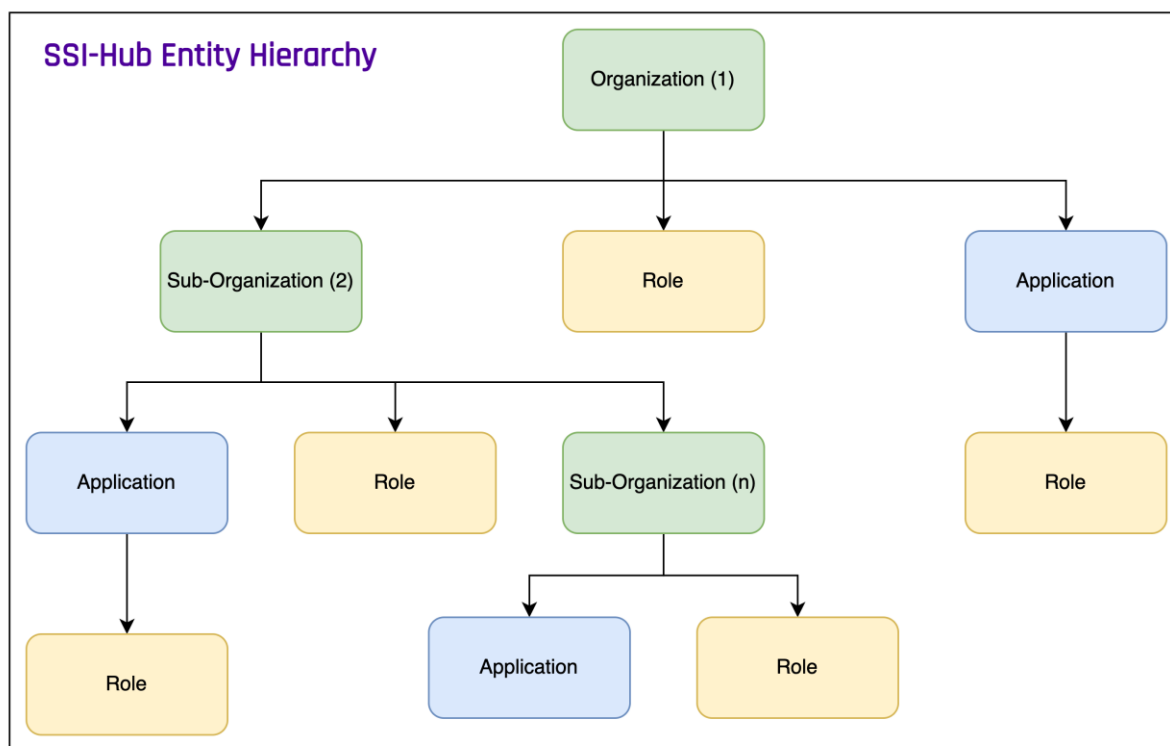


Figure 3: SSI-Hub Entity Hierarchy

In Digital Spine, there are two (2) roles required for each application context: **user** and **topiccreator**. All roles under an application must require the **user** role.

The **user** role in Message Broker ensures that the DID has access to the general APIs of DDHub Message Broker while the **topiccreator** role ensures that the DID is allowed to

manage data schemas (topics) in any other application which the DID has also a **topiccreator** role with. Figure 4 shows an example of how roles can be assigned:

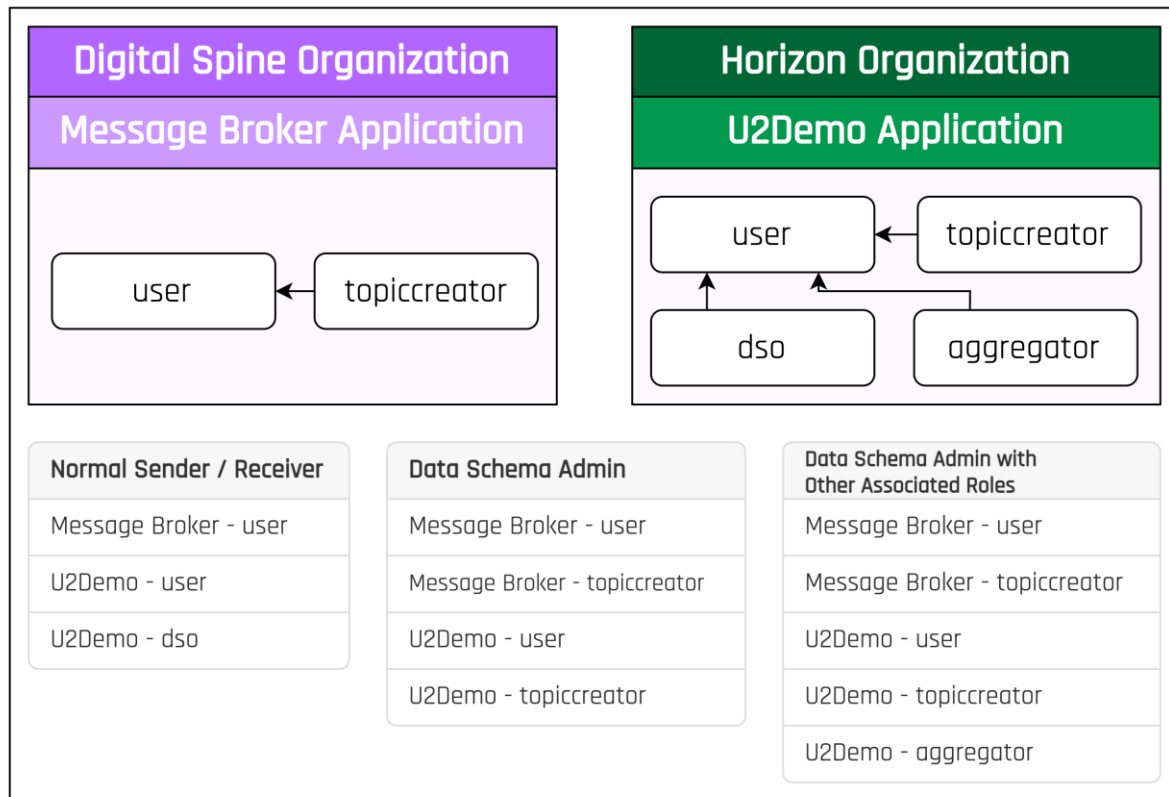


Figure 4: Sample Role Associations

In the example above, there can be different combinations of roles associated with a DID. A DID can have roles which only allow it to be just a normal sender or receiver of messages. A DID can also have roles which allows it to have access to data schema (topic) management features on top of having messaging capabilities. Other roles may also be associated with a DID for further messaging channel restrictions management.

2.2.1.3 Authentication Strategies between the Client GW and the Message Broker

The scope of SSI-Hub’s IAM is only within the integration points between the DDHub Client GW and the DDHub Message Broker. A different IAM strategy is applied between the Participant System and the DDHub Client Gateway (GW) which will be described further in this document.

Since the DDHub Client GW and the DDHub Message Broker may exist in different, independent environments, it is paramount to ensure a higher level of trust and security between them. Thus, mutual Transport Layer Security (mTLS) protocol needs to be enabled and setup as a two-way authentication between them. mTLS requires both the client and the server to authenticate each other using digital certificates. The digital certificate is generated and provided by the organization that houses the message broker. The participant admin will then need to configure the digital certificate using the DDHub Client GW.

The DDHub Client GW communicates with the DDHub Message Broker using Representational State Transfer (RESTful) APIs. To access these APIs, the DDHub Client GW authenticates using DID-based Auth0 strategy whereby identity token is generated from the DID private key to produce access and refresh tokens. These tokens have configurable Time to live (TTLs) to ensure optimal expiry for robust implementation of security practices. Figure 5 present the authentication strategies between the Client GW and the Message Broker.

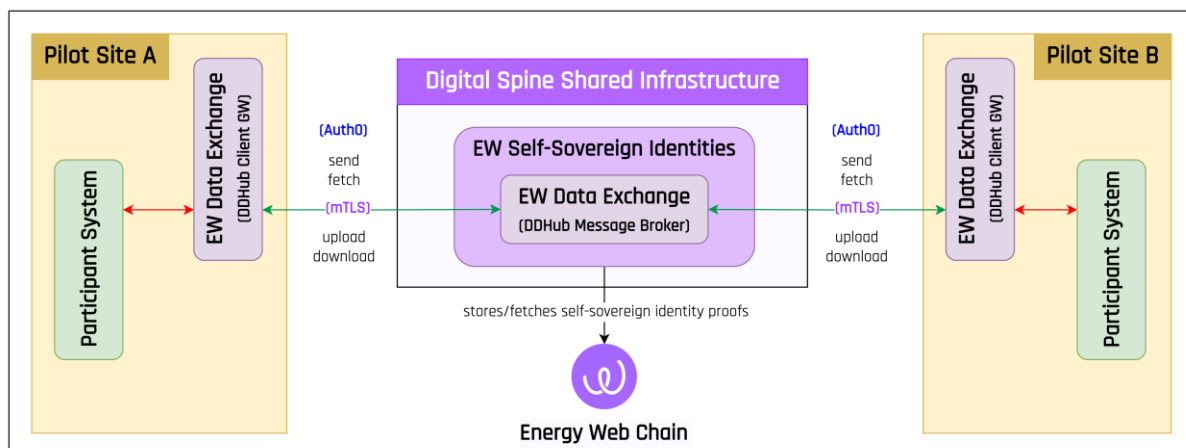


Figure 5: Authentication Strategies between the Client GW and the Message Broker

2.2.1.4 DDHub Client Gateway

A DDHub Client GW instance (see Figure 6) of the EW Data Exchange (DDHub) is required to be deployed in each participant environment. The admin uses it to manage the identity and its roles, data schemas (topics), and data channels. The DDHub Client GW and EW Switchboard provides intuitive graphical user interface for admin-related tasks which includes but not limited to data schema management, channel management, and gateway settings.

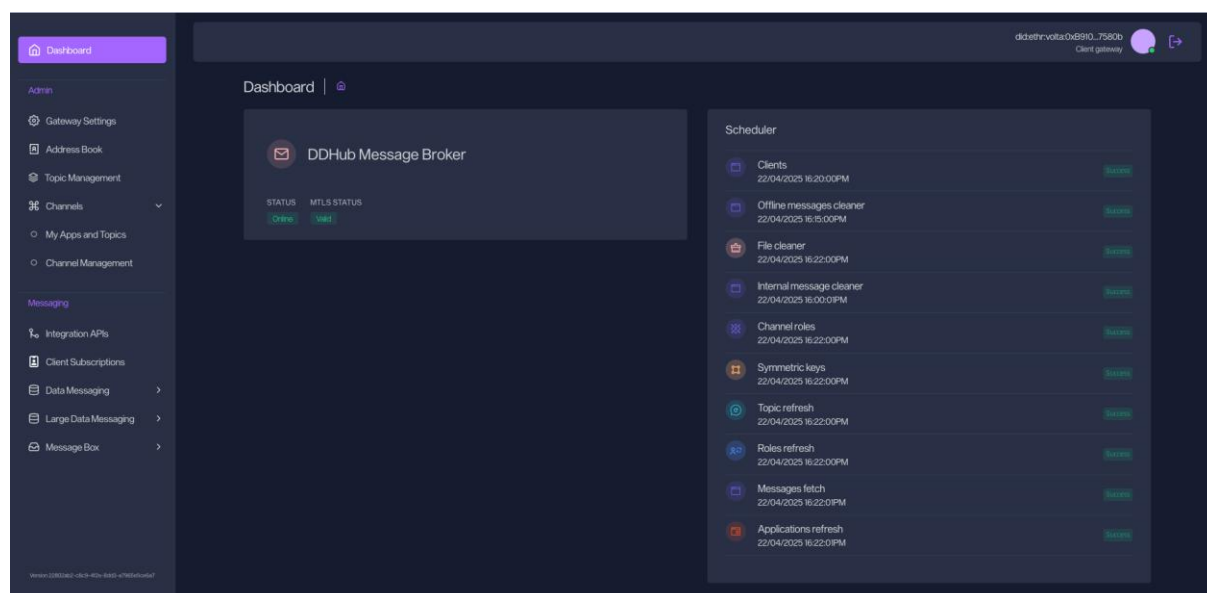


Figure 6: DDHub Client GW User Interface

Behind the DDHub Client GW is a single or a group of participant systems or services depending on how data flows from one system to another. A participant system may take the form of a Community System, Grid Operator System, FOS, P2P Exchange, Rewards Service, etc. Figure 7 and Figure 8 present a sample Client GW and DDHub Client GW, respectively.

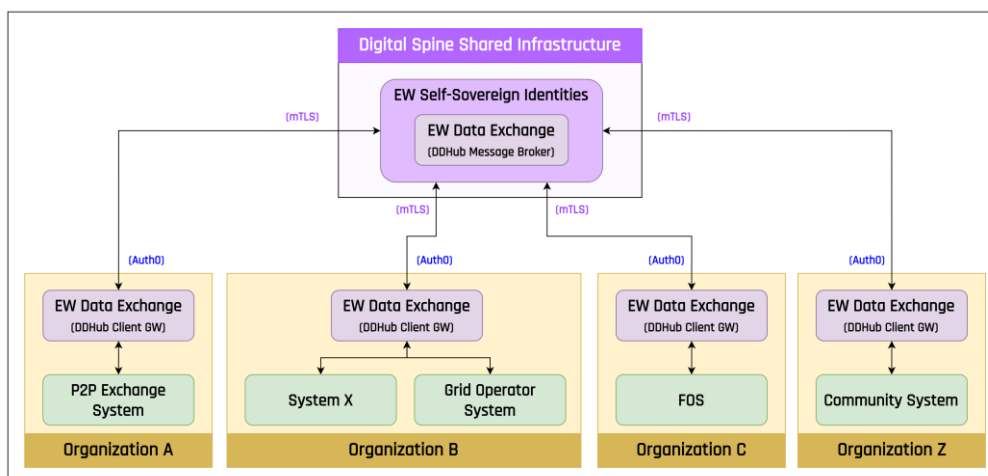


Figure 7: Sample Client GW Setup by Organization

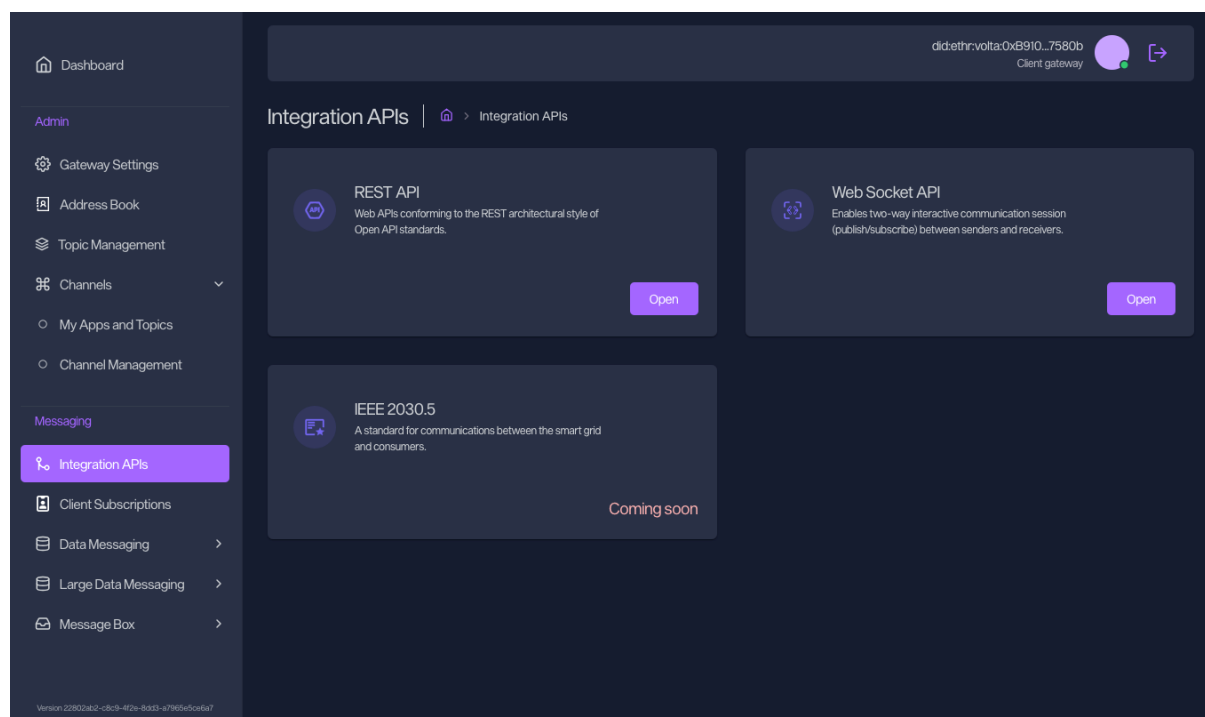


Figure 8: DDHub Client GW Screen for RESTful and WSS APIs

The EW Data Exchange exposes RESTful and WSS APIs (Figure 9) via the DDHub Client GW for participant systems to send to and receive data from other participants. Access to these APIs must be within the private or protected premises of the participant. These APIs are designed for system-to-system integration and must not be used otherwise.

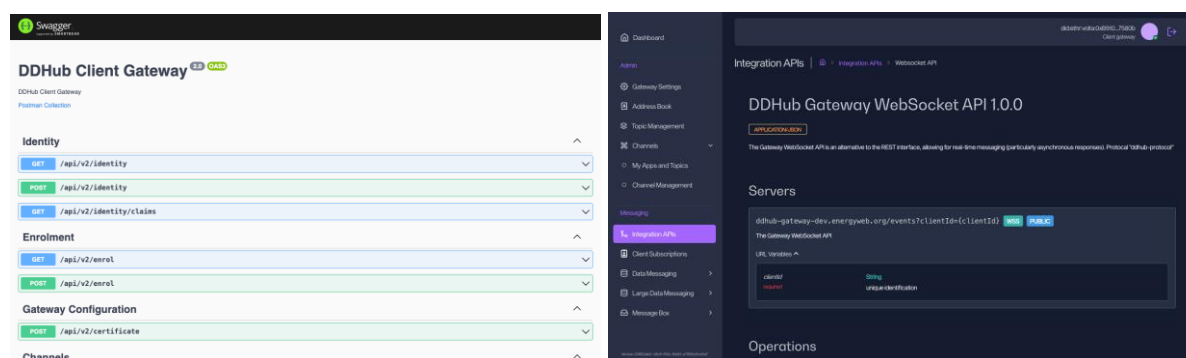


Figure 9: RESTful and WSS API Documentations

Documentation of available APIs are available via the DDHub Client GW user interface.

An API key must be enabled for secure system-to-system integration with the DDHub Client GW. The API key is configured as an environment variable or a value set in the integrated key value of the DDHub Client GW.

In any case, for example a P2P local services exchange between participants, DERs owned by the participant are onboarded and integrated into their respective systems. The participant system takes care of sharing data and instructions to the other participants in the network through the DDHub Client GW. In addition, the participant system is responsible for storing and fetching data coming from the DERs and other participants. The general architecture for an end user is presented in Figure 10.

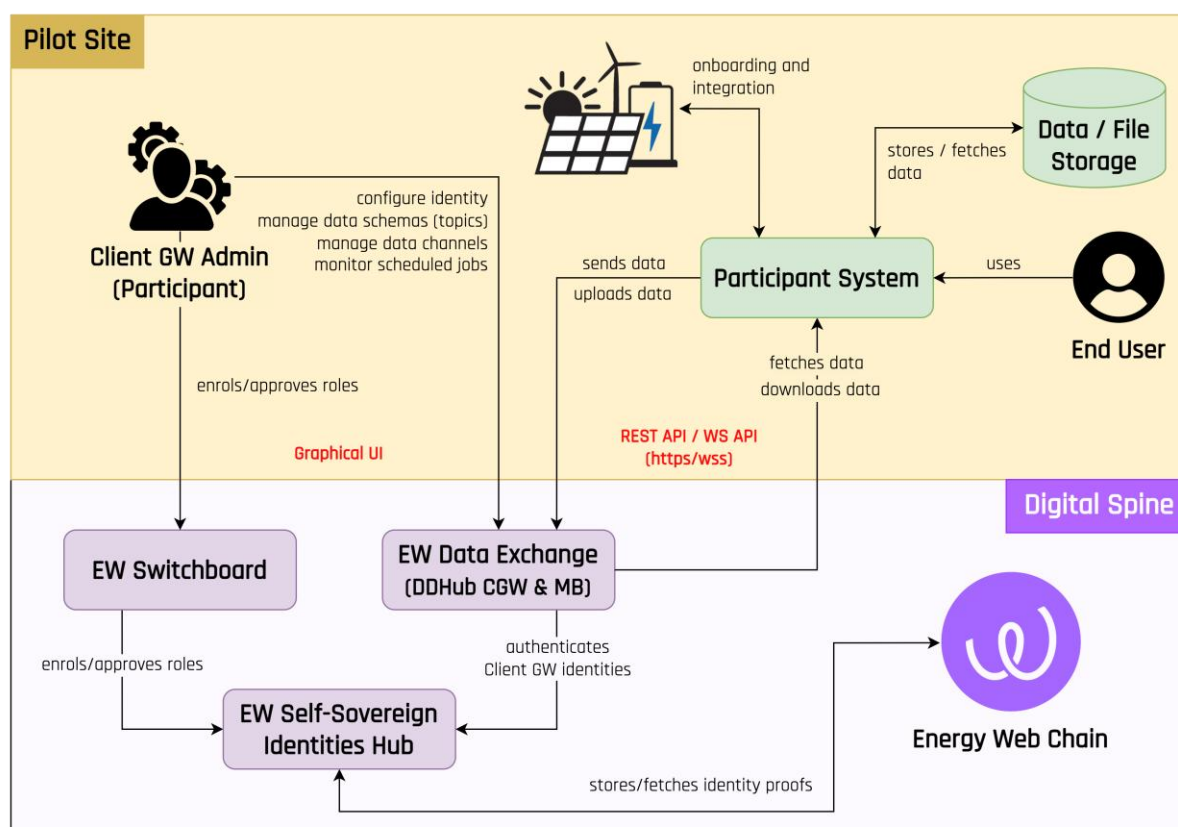


Figure 10: General Architecture

The Digital Spine infrastructure only stores data temporarily to aid the data exchange and ensure at-least-once delivery. By default, data persists in the DDHub Storage for 24 hours. The requirements for the integration in the Pilots sites are the following:

1. Defined business unit to group business use-cases (example: a project, an application)
2. Defined business unit roles and governance requirements
3. Defined use-cases, data flows, and data schemas
4. Prepared identities and issued roles
5. Dedicated DDHub Message Broker for the project
6. Deployed Client GWs for each pilot site

2.2.2 Information Exchange requirements

The P2P Exchange System's use-cases and data flows need to be defined before the actual architecture can be drawn. These requirements are being defined in Task 1.4, centered in the demonstration requirements, and in Task 2.1, based on a more theoretical and harmonized view. How Digital Spine interacts with the components of this system is highly dependent in the aforementioned prerequisites.

In a P2P energy community, effective and secure information exchange is critical for coordinating energy production, consumption, and storage among decentralized actors. Participants—including prosumers, consumers, and aggregators—must continuously share data on energy availability, demand forecasts, and market preferences. This exchange enables real-time matching of supply and demand while maintaining transparency and trust among users. Furthermore, standardized data protocols and interoperable digital platforms are essential to ensure that the decentralized system functions reliably and scales efficiently across various technical and regulatory environments.

Flexibility services, such as demand response and distributed storage dispatch, are increasingly provided by P2P communities to system operators and flexibility operators. These services help stabilize the grid, defer infrastructure investments, and manage congestion or voltage issues. To enable this, operators must access timely, granular data on the status and availability of flexible assets within the community. This requires a bidirectional information flow: system operators issue requests or market signals, and community platforms or aggregators respond with available flexibility volumes, activation timelines, and technical constraints. Ensuring data privacy, integrity, and low-latency communication is essential for the success of these transactions.

Another important aspect that will be considered in the project are the sharing mechanism and reward aspects. These mechanisms must be transparent, fair, and dynamically adjusted based on market conditions, system needs, and individual contributions. Smart contracts and blockchain-based systems can automate the settlement of these rewards, reducing administrative costs and ensuring tamper-proof accounting. By linking performance metrics—such as responsiveness, reliability, and volume of flexibility delivered—to financial or token-based incentives, communities can align individual behavior with collective grid-support goals. Ultimately, well-designed reward functions encourage active engagement, increase participation rates, and enhance the overall value proposition of energy communities in the energy transition.

3 Design Principles and Requirements

3.1 Functional Requirements

To support the development of U2Demo platform and P2P and energy sharing mechanism, the platform must fulfill a range of functional requirements that ensure secure, transparent, and efficient operation. This section presents some of the Functional Requirements (FR) that should be considered.

FR1: Secure Identity & Access Management

Users (operators, aggregators, prosumers) shall authenticate via EWF's SSI-based identity hubs and be assigned roles/permissions according to GDPR-aligned consent scopes.

FR2: Data Ingestion & Normalization

The system shall ingest real-time telemetry from DERs and grid assets (via MQTT/HTTP) and normalize it to a common semantic schema.

FR3: Marketplace Transactions

Support peer-to-peer energy trading workflows, with order matching, settlement and tokenization managed on the DLT layer.

FR4: Analytics & Visualization

Provide dashboards for grid balancing forecasts, asset utilization, and privacy-preserving KPIs, with drill-down to device-level data.

FR5: Policy & Rule Enforcement

Enforce user-defined business rules (e.g. dynamic pricing, curtailment thresholds) at runtime via a policy engine.

3.2 Non-functional requirements

Concerning the Non-Functional Requirements (NFR), the following have been identified:

NFR1: Scalability

System must support horizontal scaling to handle up to <X> messages per day, leveraging containerized microservices and Kubernetes autoscaling.

NFR2: Security & Privacy

All data at rest and in transit shall be encrypted (TLS 1.3, AES-256); personal data handling must comply with GDPR principles of minimization and purpose limitation.

NFR3: Availability & Resilience

Target 99.9 % uptime with multi-zone deployment, active-passive failover, and canary-style rollouts to minimize disruptions.

NFR4: Observability

End-to-end tracing (Jaeger), metrics (Prometheus), and centralized log aggregation (ELK) must be in place to achieve <5 min MTTR for critical incidents.

NFR5: Performance

Average API response time ≤ 200 ms under nominal load; data pipeline end-to-end latency ≤ 1 second for real-time workflows.

3.3 Standards and Interoperability

Finally, concerning the requirements related to standards and interoperability (SI), the following should be considered.

SI1: Semantic Data Model

- Adopt Next Generation Service Interface with Linked Data (NGSI-LD) for entity–property–relationship modelling to ensure consistent cross-domain semantics.
- Align with TM Forum’s Common Semantic Data Model (CSDM) for asset and service definitions, enabling seamless integration with telco and utility catalogs.

SI2: Ontology Framework

- Define a U2Demo Ontology in Web Ontology Language/Description Language (OWL/DL), capturing domain concepts (distributed energy resources, markets, roles, policies) and their interrelations.
- Publish as a machine-readable schema with SPARQL endpoints for semantic queries and reasoning services.

SI3: Protocol Adapters

- Provide pluggable adapters for IEC 61850, OCPP 1.6J, MQTT-SN and REST/JSON to integrate legacy and modern devices.

SI4: API Governance

- Expose RESTful and GraphQL endpoints with OpenAPI/Swagger specifications, versioned and stored in an Artifactory registry.

SI5: Data Exchange & Consent

- Implement World Wide Web Consortium (W3C) compliant Verifiable Credentials for consent management, enabling auditable, user-driven data sharing.

4 Technical Requirements

4.1 Microservices Definition

The microservices architecture for the U2Demo platform is still evolving and will be progressively refined in line with the development timelines of WP3 and WP4. In particular, Task 3.2 (focused on standards and security) and Task 3.3 (focused on the technical implementation of the platform) will shape the final structure of containerized services, API interfaces, and data schema management. The design is being built around Energy Web's DDHub integration approach, which enables each participant to deploy and operate their own integration containers, either on-premise or in the cloud, ensuring modularity, autonomy, and alignment with the project's decentralization principles. Finalized service definitions, informed by pilot deployment and integration feedback, will be documented in upcoming deliverables such as D3.3 and D4.3.

At this stage, the design draws heavily on Energy Web's hands-on experience with similar architectures. That experience informs the current setup, where microservices enable secure, identity-driven data exchange through RESTful and WebSocket APIs, support role-based access via verifiable credentials, and provide dynamic schema validation. This foundation offers both a tested technical baseline and a flexible framework for future enhancements.

4.2 Blockchain/Distributed Ledger Technology Integration

The integration of blockchain technology within the U2Demo platform focuses on establishing trust, verifiability, and decentralized governance of roles and interactions. Rather than storing transactional data or payloads on-chain, the system will anchor only critical identity metadata, such as DIDs, role assignments, and credential proofs, on the EWC. This lightweight use of the distributed ledger ensures tamper-evidence and transparency without introducing unnecessary performance overhead. Participants maintain full control over their data and logic execution, while verifiers across the network can independently validate authenticity and authorization using verifiable credentials. The upcoming work in Task 3.2 will define shared standards for identity formats and credential schemas, while Task 3.3 will implement this integration across the DDHub architecture, including message signature validation, DID-based authorization, and role-based access enforcement.

4.3 APIs and Data Structures

The detailed API specifications and data schema definitions for U2Demo will be developed iteratively in line with the progress of Task 3.2 (standardization and security) and Task 3.3 (platform implementation). These specifications will be driven by the needs of pilot integrations and the emerging standards across the U2Demo ecosystem, including ontology work and schema alignment activities under WP2 and WP6. RESTful and WebSocket (WSS) APIs will

be exposed through the DDHub Client Gateway, enabling participants to publish, subscribe, and validate messages based on dynamic schema definitions.

Each API will be role-gated using verifiable credentials and DID-based authentication. Topic-level schemas and messaging protocols will follow established patterns for secure, schema-driven data exchange, where JSON Schema-based payload validation and channel-level access control help ensure interoperability, traceability, and data integrity across diverse participants. These design principles will be aligned with semantic models such as NGS-LD and TM Forum's CSDM where applicable and formally described in upcoming deliverables including D3.3, D4.2, and D4.3.

5 Security and Privacy Considerations

5.1 Data Management and Privacy Compliance

The U2Demo platform adopts a privacy-by-design approach, ensuring that data management practices align with the core principles of the GDPR, including data minimization, purpose limitation, and user consent. SSI and VCs form the basis of identity and access management, giving participants full control over their identity data and the permissions granted to others. No personal data is stored on-chain; only hashed references (e.g. credential proofs and DID metadata) are anchored to the Energy Web Chain, ensuring compliance with data sovereignty and immutability requirements. Energy Web's solutions, including the DDHub, Switchboard, and EW-IAM components, have been implemented in projects requiring strict alignment with EU data protection frameworks. These tools enable auditable, decentralized data access while allowing each participant to govern its own information in line with organizational or national compliance policies.

5.2 Cybersecurity Guidelines

Cybersecurity within the U2Demo architecture is addressed through a defense-in-depth strategy, with clear responsibilities distributed across all layers of the platform. While Energy Web provides foundational identity, access, and data exchange components (e.g. DDHub, EW-IAM, Switchboard), cybersecurity assurance is a shared responsibility involving all platform participants, from infrastructure hosts to application developers and integration partners.

At the infrastructure level, each participant hosting a DDHub Client Gateway is responsible for securing its runtime environment, whether on-premise or cloud-based. This includes enforcing operating system hardening, firewall configuration, vulnerability scanning, and endpoint monitoring. Participants are also expected to manage the secure storage of cryptographic key material (DID private keys and TLS certificates), either through Hardware Security Modules (HSMs) or trusted key vaults.

All data exchanged within the platform is encrypted in transit using mTLS. At the application layer, fine-grained access control is enforced using verifiable credentials that are issued and validated against known DIDs. This ensures that only authorized entities can publish, subscribe, or retrieve messages via the DDHub environment. Message integrity is guaranteed using cryptographic signatures, and topics/schemas are versioned and validated at runtime to prevent injection or corruption of data.

The platform promotes a “zero-trust” architecture, where authentication and authorization are required for every interaction, even between known systems. Channel-level governance ensures that participants retain sovereignty over their communication domains and can independently define access rules and data retention policies.

To support operational resilience, the system architecture includes at-least-once delivery guarantees, message persistence with configurable expiry, and retry logic for failed transmissions. Logging, audit trails, and health monitoring are built into both the client and broker components to detect anomalies and support incident response.

As the project evolves, the consortium will continue to assess risks and align with European Union (EU) cybersecurity frameworks such as the Network and Information Security 2 (NIS2) Directive [7], European Union Agency for Cybersecurity (ENISA) technical guidelines, and ISO/IEC 27001. Additional security policies will be formalized through Task 3.2 to define secure integration standards, certificate renewal processes, incident handling procedures, and cross-organizational security coordination mechanisms.

6 Implementation Guidelines

6.1 Development Practices and Standards

The implementation of the U2Demo platform will follow modern, modular, and secure software engineering practices. The microservices architecture will be based on containerized components (Docker/Kubernetes), aligned with Task 3.3 (Platform Implementation), and built with a focus on scalability, maintainability, and portability. The codebase will adhere to open-source best practices, including the use of public version control (e.g., Git), Continuous Integration/Continuous Delivery (CI/CD) pipelines, code linting, static analysis tools, and documentation aligned with the OpenAPI standard for RESTful interfaces.

Each participant-facing service will be exposed via secure APIs documented using Swagger/OpenAPI, and version-controlled for lifecycle management. Semantic versioning will be adopted to ensure compatibility across deployments. The message schemas and APIs will be defined under Task 3.2 (Standardization & Security) using JSON Schema for payload validation, with an emphasis on schema evolution and backward compatibility.

Role-based access control, cryptographic key handling, DID management, and secure enrollment workflows will be implemented using Energy Web's IAM and Switchboard components, which have been field-tested in multi-stakeholder energy projects. Integration security, including certificate handling, mTLS setup, and DID token issuance, will follow a standardized onboarding process to be documented and automated as part of the implementation timeline.

6.2 Recommended Tools and Technologies

The U2Demo implementation will leverage a suite of open-source and enterprise-grade technologies to support decentralized, scalable data exchange. Based on previous experience from deployments of DDHub and EW-DOS, the following toolsets are recommended:

- Containerization & Orchestration: Docker and Kubernetes for deployment across participant environments.
- Identity and Access Management: EW-IAM client libraries, DIDKit (or equivalent), and Switchboard for role-based access control using verifiable credentials.
- Data Exchange: DDHub Client Gateway SDK and Broker, exposing RESTful and WebSocket endpoints with schema validation and pub/sub capabilities.
- Message and Schema Management: NATS JetStream for message persistence, JSON Schema for data contracts, and optional Apache Camel connectors for backend integration patterns.
- Monitoring & Observability: Prometheus for metrics collection, Grafana for dashboards, and centralized logging (e.g., ELK or Loki).
- DevOps and Testing: GitHub Actions or GitLab CI for pipelines, with tools like Postman, Dredd, or REST Assured for contract/API testing.

To ensure portability across participants, the system will support multiple deployment models: containerized (Docker/K8s), embedded (Node.js SDK), or managed by an integration service provider. Participants can either self-host their Client Gateway or connect to a trusted provider's DDHub container via secure credentials. Some of the technology stack responsibilities and deployment models are presented in Appendix A.

These implementation guidelines will be refined and documented as development progresses and pilots begin onboarding. Final practices and integration instructions will be captured in D3.3 and D4.3, with additional tooling and process recommendations adapted based on participant feedback.

References

- [1] E. Barabino *et al.*, “Energy Communities: A review on trends, energy system modelling, business models, and optimisation objectives,” *Sustain. Energy Grids Netw.*, vol. 36, p. 101187, Dec. 2023, doi: 10.1016/j.segan.2023.101187.
- [2] European Parliament and of the Council, *Directive on the promotion of the use of energy from renewable sources*. 2018. [Online]. Available: Directive (EU) 2018/2001
- [3] *Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU (recast) (Text with EEA relevance.)*, vol. 158. 2019. Accessed: Jul. 05, 2024. [Online]. Available: <http://data.europa.eu/eli/dir/2019/944/oj/eng>
- [4] S. Ahmed, A. Ali, and A. D’Angola, “A Review of Renewable Energy Communities: Concepts, Scope, Progress, Challenges, and Recommendations,” *Sustainability*, vol. 16, no. 5, p. 1749, Feb. 2024, doi: 10.3390/su16051749.
- [5] M. Kubli and S. Puranik, “A typology of business models for energy communities: Current and emerging design options,” *Renew. Sustain. Energy Rev.*, vol. 176, p. 113165, Apr. 2023, doi: 10.1016/j.rser.2023.113165.
- [6] E. Gomes, L. Pereira, A. Esteves, and H. Morais, “PyECOM: A Python tool for analyzing and simulating Energy Communities,” *SoftwareX*, vol. 24, p. 101580, Dec. 2023, doi: 10.1016/j.softx.2023.101580.
- [7] European Parliament and of the Council, *Directive on Measures for a high common level of cybersecurity across the Union*. 2022. [Online]. Available: <https://eur-lex.europa.eu/eli/dir/2022/2555/oj/eng>

Appendix A

Table A1: Technology Stack Responsibilities per Component

Component	Primary Technology	Purpose	Responsible Party
DDHub Client Gateway (GW)	Docker, Node.js, REST/WSS APIs	Local message handling, DID-based auth, schema validation	Each Participant
DDHub Message Broker	NATS JetStream, File Store, JWT	Secure message routing and storage (at-least-once delivery)	Energy Web / Hosting Partner
Switchboard (Role & DID Management)	EW-IAM, DIDs, Verifiable Credentials	Issue, manage and validate identities and roles	Energy Web
Schema Registry	JSON Schema, Apache Camel (optional)	Validate messages against agreed topic schemas	Energy Web / WP3 Team
Monitoring & Observability	Prometheus, Grafana, ELK Stack	Metrics, logging, error tracking	Participants + WP3 DevOps Leads
Deployment Tooling	GitHub Actions, Terraform, Helm	CI/CD, IaC, container deployment	WP3 Lead (EXAION) + Participant Ops Teams
Key Management	Vault (e.g., Azure Key Vault, HashiCorp)	Secure storage of private keys and credentials	Each Participant

Table A2: Deployment Model Options

Deployment Model	Description	Use Case Fit	Hosting Responsibility
Self-hosted Container (Docker)	DDHub Client GW deployed in participant's own cloud/on-prem environment via Docker	Most flexible; ideal for mature IT teams	Participant
Kubernetes Cluster	DDHub Client GW deployed via Helm in a managed or bare-metal K8s cluster	For scalable, production-grade ops	Participant / Managed Provider
Integration Provider (Hosted)	Participant connects to a third-party-managed DDHub Client GW via secure credentials (e.g. Auth Token)	For lean teams with limited DevOps	Certified Service Provider
Embedded SDK (Node.js)	Lightweight SDK embedded into existing backend systems (JavaScript/TypeScript)	Ideal for integrators and small systems	Participant