

Decentralized Publishing Architecture: POSSE to Federation

Architecture Documentation

October 23, 2025

Contents

1 Overview

This document describes a comprehensive architecture for decentralized content publishing that gives individuals (PERSON) complete control over their data while providing flexible hosting and publishing options. The system is built around the concept of POSSE (Publish on your Own Site, Syndicate Elsewhere) as a control center, with clear separation between data ownership, hosting, and publishing responsibilities.

1.1 Key Principles

- **Data Ownership:** PERSON maintains ultimate control over their content through POSSE
- **Flexible Hosting:** Content can be self-hosted or hosted on external HUBs
- **Protocol Agnostic:** Support for multiple publishing protocols (ActivityPub, etc.)
- **Gate Control:** Endpoints can enforce their own access and content rules
- **Feedback Loop:** Publishers receive reports on content acceptance and distribution

2 Actors and Roles

2.1 PERSON

The individual user who creates and controls content. PERSON has ultimate authority over:

- What content gets published
- Which audiences receive content
- Which protocols and endpoints to use
- Content modification and deletion

2.2 HUBOWNER

The owner of hosting or publishing infrastructure. HUBOWNER can:

- Provide hosting services for POSSE containers
- Provide publishing/syndication services
- Set HUB-level rules and policies
- Manage infrastructure (compute, storage, uptime)

3 Personal Data Layer

The PERSONAL layer represents the user's content creation and organization system.

3.1 Components

- **PERSON (role)**: The user's role within the personal data context
- **USER (identity, keys)**: Digital identity and cryptographic keys
- **TEXT (free-form)**: Unstructured text content created by the user
- **BUNDLES (curated)**: Organized collections of content
- **ITEMS (metadata/hooks)**: Structured items with metadata and integration hooks

3.2 Relationships

- PERSON controls USER identity and keys
- PERSON controls TEXT content creation
- PERSON manages BUNDLES organization
- BUNDLES contain some TEXT content
- BUNDLES create structured ITEMS

4 POSSE: The Control Center

POSSE (Publish on your Own Site, Syndicate Elsewhere) is the central control system where PERSON maintains all "leashes" - the rules and decisions about content publishing.

4.1 Components

- **POSSE (container)**: The data container holding all content and rules
- **RULES (audience, endpoints)**: PERSON's publishing rules specifying audiences and target endpoints
- **INGEST**: Process of bringing content into POSSE
- **SLA (POSSE)**: Service Level Agreement covering ownership, portability, and local policy

4.2 Key Functions

- **Content Control**: PERSON sets rules for what content goes where
- **Audience Management**: Defines who can access what content
- **Endpoint Selection**: Chooses which protocols and endpoints to use
- **Policy Enforcement**: Implements PERSON's content and access policies

5 Hosting Layer

The hosting layer provides infrastructure for running POSSE containers and related services.

5.1 Self-Hosted Option

When PERSON chooses to self-host:

- PERSON provides their own compute resources
- PERSON manages their own storage
- PERSON is responsible for uptime and maintenance
- Complete control over infrastructure and data

5.2 HUB-Hosted Option

When PERSON uses a HUB for hosting:

- HUBOWNER provides compute resources
- HUBOWNER manages storage infrastructure
- HUBOWNER ensures uptime and maintenance
- PERSON retains control over content and rules

6 Publishing Layer

The publishing layer handles the distribution of content from POSSE to various endpoints.

6.1 Self-Publishing

When PERSON chooses to self-publish:

- Direct syndication to target endpoints
- Direct relay to other services
- PERSON manages all publishing infrastructure
- Complete control over publishing process

6.2 HUB Publishing

When PERSON uses a HUB for publishing:

- HUB handles syndication to target endpoints
- HUB manages relay to other services
- HUBOWNER provides publishing infrastructure
- PERSON retains control over content and rules

7 Regime Layer

The regime layer defines the protocols and standards used for content distribution.

7.1 ActivityPub

ActivityPub is a decentralized social networking protocol that enables:

- Interoperable social networking
- Decentralized content distribution
- Standardized activity streams
- Federation between different platforms

7.2 Other Protocols

Support for additional protocols:

- RSS/Atom feeds
- Custom APIs
- Blockchain-based protocols
- Future protocol extensions

8 Gate Layer

The gate layer provides access control and content filtering at the endpoint level.

8.1 Access Control

- **Audience Rules:** Defines who can access content
- **Authentication:** Verifies user identity
- **Authorization:** Determines access permissions
- **Rate Limiting:** Controls access frequency

8.2 Content Filtering

- **Content Moderation:** Filters inappropriate content
- **Spam Prevention:** Blocks unwanted content
- **Quality Control:** Ensures content meets standards
- **Legal Compliance:** Enforces legal requirements

9 Federation Layer

The federation layer represents the distributed network of endpoints and services.

9.1 Fediverse

The decentralized social network including:

- Mastodon instances
- Pleroma servers
- Misskey instances
- Other ActivityPub-compatible services

9.2 Other Endpoints

Additional publishing targets:

- Social media platforms
- Blog networks
- News aggregators
- Custom applications

9.3 Discovery and Aggregation

- **Content Discovery:** Finding relevant content
- **Aggregation Services:** Collecting content from multiple sources
- **Search Engines:** Indexing and searching content
- **Recommendation Systems:** Suggesting relevant content

10 Feedback Loop

The system includes a feedback mechanism that reports back to POSSE about content distribution success.

10.1 Reporting

- **Acceptance Reports:** Which endpoints accepted content
- **Distribution Status:** Where content was successfully published
- **Error Reports:** What failed and why
- **Performance Metrics:** How well content performed

10.2 Learning

- **Rule Optimization:** Improving publishing rules based on success rates
- **Endpoint Selection:** Choosing better endpoints for future content
- **Content Strategy:** Adjusting content based on distribution success
- **Policy Updates:** Refining access and content policies

11 Implementation Considerations

11.1 Security

- **End-to-End Encryption:** Protecting content in transit
- **Access Control:** Implementing proper authentication and authorization
- **Data Integrity:** Ensuring content hasn't been tampered with
- **Privacy Protection:** Safeguarding user data and preferences

11.2 Scalability

- **Load Balancing:** Distributing traffic across multiple instances
- **Caching:** Improving performance with content caching
- **CDN Integration:** Using content delivery networks for global distribution
- **Database Optimization:** Efficient storage and retrieval of content

11.3 Interoperability

- **Protocol Support:** Implementing multiple publishing protocols
- **API Standards:** Following established API conventions
- **Data Formats:** Using standard data exchange formats
- **Version Management:** Handling protocol and API versioning

12 Conclusion

This architecture provides a comprehensive framework for decentralized content publishing that balances individual control with flexible hosting and publishing options. By separating concerns between data ownership (POSSE), hosting (HUB), and publishing (REGIME/GATE), the system enables both technical flexibility and user autonomy.

The key innovation is the POSSE as a control center where individuals maintain ultimate authority over their content while leveraging external infrastructure as needed. This approach ensures that users never lose control of their data while benefiting from shared infrastructure and services.

13 Architecture Diagram

The diagram above shows the complete flow from personal data creation through POSSE control to various hosting, publishing, and federation options. Each layer represents a distinct concern, with clear interfaces between them.

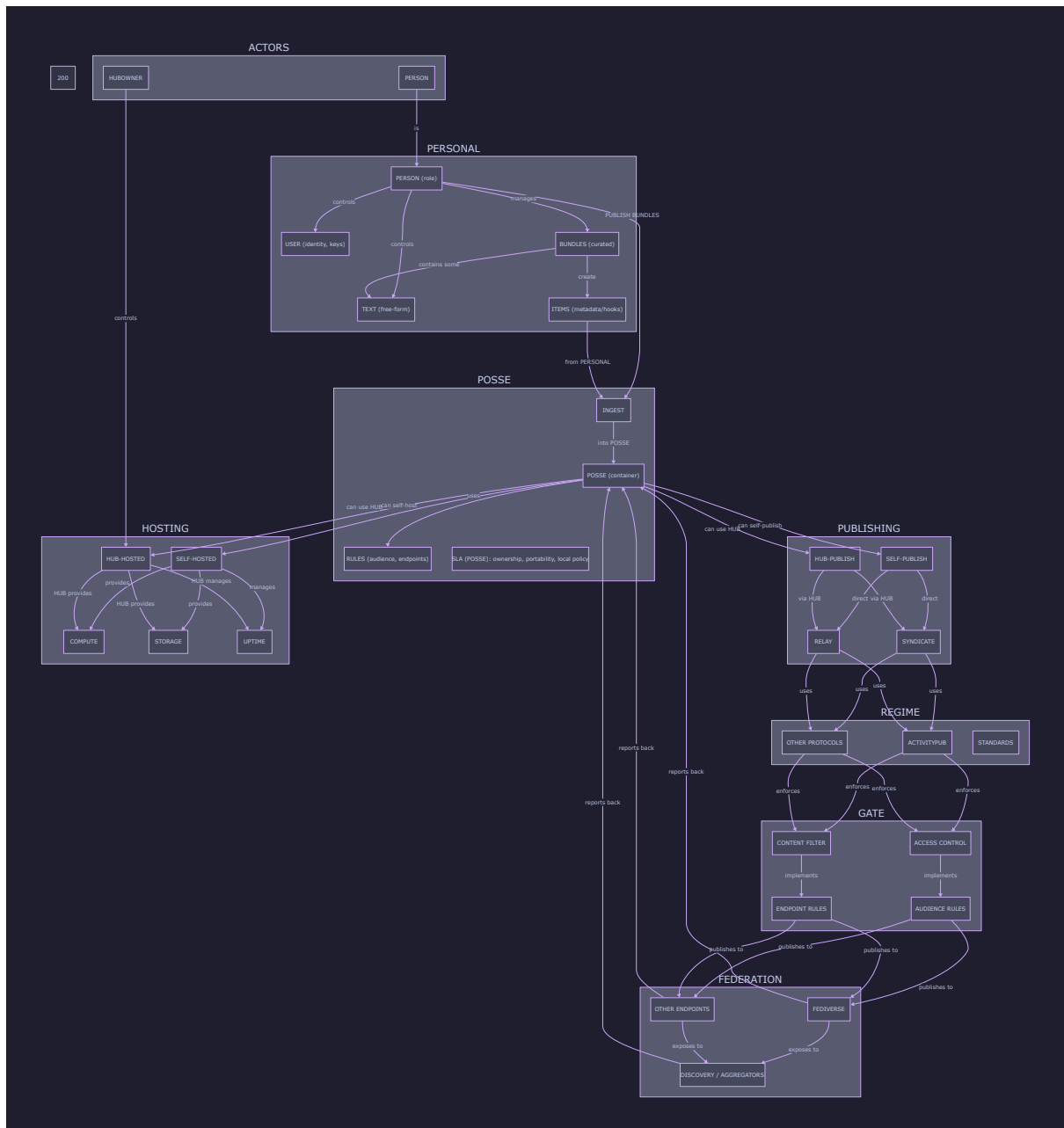


Figure 1: Complete Architecture Flow: From Personal Data through POSSE Control to Federation