

# SPEC-07: Core Services Module

## 1. Background

The **Core Services** module comprises foundational microservices that power authentication, user management, configuration, feature toggles, and telemetry ingestion across the Hearthlink ecosystem. Core Services underpin all other modules (Alden, Alice, Mimic, Synapse, Vault, Sentry) by providing stable, secure APIs for identity, permissions, settings, and analytics.

## 2. Requirements (MoSCoW)

### Must have

- OAuth2.0/OpenID Connect provider issuing JWTs and supporting introspection
- User Profile Service with CRUD APIs and encrypted-at-rest storage
- Feature Flag Service for dynamic toggles with rollout and audit logging
- Configuration Store: versioned key-value settings scoped by environment
- Metrics Ingestion API: high-throughput bulk write to time-series DB (Prometheus/Timescale)

### Should have

- Analytics Query API: pre-aggregated telemetry for dashboards
- Centralized Rate-Limit Service with policy management

### Could have

- Billing & Usage reporting for cost analysis
- Service Catalog UI for API discovery

### Won't have (this increment)

- GraphQL Gateway
- Advanced A/B testing engine

## 3. Method

### 3.1 Architecture Diagram

```
@startuml
package "Core Services" {
    [Auth Service] --> [User DB]
    [Profile Service] --> [User DB]
    [Feature Flag Service] --> [Config Store]
    [Config Store] --> [Config DB]
    [Metrics API] --> [Metrics Store]
}
```

```

    [RateLimit Service] --> [RateLimit Store]
}
package "Consumers" {
    [UI Modules]
    [Synapse]
    [Sentry]
    [Vault]
    [Mimic]
}
[Consumers] --> [Auth Service]
[Consumers] --> [Profile Service]
[Consumers] --> [Feature Flag Service]
[Consumers] --> [Config Store]
[Consumers] --> [Metrics API]
[Consumers] --> [RateLimit Service]
@enduml

```

### 3.2 Data Schema

```

@startuml
table Users {
    + user_id      : UUID [PK]
    + email        : VARCHAR [UNIQUE]
    + password_hash: VARCHAR
    + created_at   : TIMESTAMP
    + last_login   : TIMESTAMP
}

table Roles {
    + role_name    : VARCHAR [PK]
    + description  : TEXT
}

table UserRoles {
    + user_id      : UUID [FK]
    + role_name    : VARCHAR [FK]
    + assigned_at  : TIMESTAMP
}

table FeatureFlags {
    + flag_name    : VARCHAR [PK]
    + enabled      : BOOLEAN
    + rollout_pct  : INT
    + metadata     : JSON
    + updated_at   : TIMESTAMP
}

```

```

table Configurations {
  + key          : VARCHAR [PK]
  + value        : JSON
  + version      : INT
  + environment  : VARCHAR
  + updated_at   : TIMESTAMP
}

table Metrics {
  + metric_id    : BIGINT [PK, auto]
  + namespace    : VARCHAR
  + name         : VARCHAR
  + labels       : JSON
  + value        : DOUBLE
  + timestamp    : TIMESTAMP
}

table RateLimits {
  + service      : VARCHAR [PK]
  + key          : VARCHAR [PK]
  + limit        : INT
  + window_sec   : INT
  + updated_at   : TIMESTAMP
}
@enduml

```

## 4. UI Components & Wireframes

### 4.1 User Management Panel

```

+-----+
| User Management                               |
| [Create User] [Bulk Import ▼] [Refresh]      |
|-----|
| [Users Table: ID, Email, Roles, Last Login]  |
+-----+

```

Component	Function	Data/API Call
UsersTable	Lists users with pagination and filters	GET /v1/users?limit=50&page=1
CreateUserButton	Opens new user form	N/A
BulkImportDropdown	Handles CSV/YAML import	POST /v1/users/import

Component	Function	Data/API Call
RefreshButton	Reloads user list	GET /v1/users
UserActionsCell	Edit/Delete per user	PUT /v1/users/{id}, DELETE /v1/users/{id}

## 4.2 Roles & Permissions Panel

```
+-----+
| Roles & Permissions |
| [Add Role] [Export Roles] |
|-----|
| [Roles Table: Role Name, Description, Assigned] |
+-----+
```

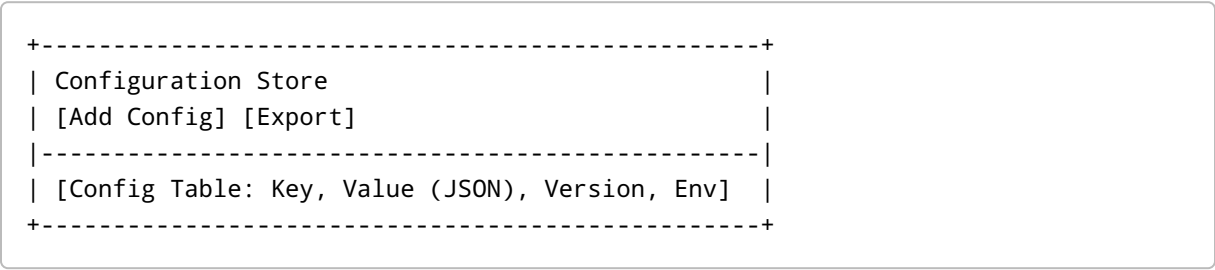
Component	Function	Data/API Call
RolesTable	Lists roles	GET /v1/roles
AddRoleButton	Opens form to create role	N/A
ExportRolesButton	Exports role list	GET /v1/roles/export
AssignRoleDialog	Map users to roles	POST /v1/userroles

## 4.3 Feature Flags Panel

```
+-----+
| Feature Flags |
| [New Flag] [Import Flags] [Toggle All] |
|-----|
| [Flags Table: Name, Enabled, Rollout%, Updated] |
+-----+
```

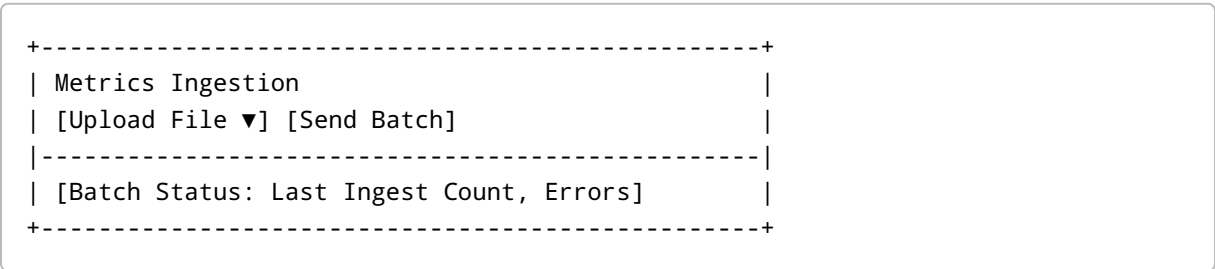
Component	Function	Data/API Call
FlagsTable	Lists feature flags	GET /v1/flags
NewFlagButton	Creates or updates a flag	N/A
ImportFlagsButton	Bulk import flags	POST /v1/flags/import
ToggleFlagSwitch	Enable/disable flag	PUT /v1/flags/{name}

4.4 Configuration Store Panel



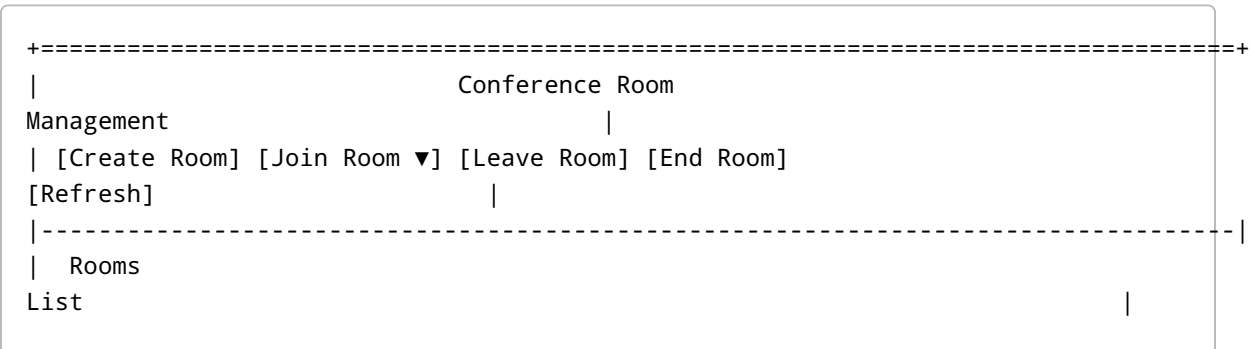
Component	Function	Data/API Call
ConfigTable	Shows config entries	GET /v1/config?environment=prod
AddConfigButton	Opens entry form	N/A
ExportConfigButton	Exports config JSON	GET /v1/config/export
EditConfigDialog	Edit value/version/env	POST /v1/config

4.5 Metrics Ingestion Panel



Component	Function	Data/API Call
UploadFileDropdown	Selects metrics file for batch ingest	N/A
SendBatchButton	Sends file payload	POST /v1/metrics/bulk
BatchStatusCard	Displays last ingest results	N/A

4.6 Full-Screen Conference Room Interface



	+-----				
+					
	ID	Name	Participants	Status	Created At
	+-----				
+					
	+-----+		+-----		
+					
	Room Detail Pane		Participants List		
	- Name: <Room Name>		- User A		
	- ID: <UUID>		- User B		
	- Status: Active		- ...		
	- Created: <Timestamp>				
	- Settings: [Edit] [Delete]		[Add Participant ▼]		
	+-----+		+-----		
+					
+=====+					

Component	Function	Data/API Call
ConferenceHeaderBar	Global actions and room filters	N/A
RoomsListTable	Full-screen table of rooms with sorting and pagination	GET /v1/conference/rooms
RoomDetailPane	Detailed view/edit of selected room settings	GET /v1/conference/rooms/{id}, PUT /v1/conference/rooms/{id}
ParticipantsList	Lists and manages participants	GET /v1/conference/rooms/{id}/participants
AddParticipantDropdown	Adds users to room	POST /v1/conference/rooms/{id}/participants
CreateRoomButton	Opens modal to configure new room	POST /v1/conference/rooms
JoinRoomButton	Joins selected room	POST /v1/conference/rooms/{id}/join

Component	Function	Data/API Call
LeaveRoomButton	Leaves current room	POST /v1/conference/rooms/{id}/leave
EndRoomButton	Ends and archives room	POST /v1/conference/rooms/{id}/end
RefreshButton	Reloads room list and details	GET /v1/conference/rooms

## 4.7 Full-Screen Project Command Workspace

```

+=====+
|                                     Project Command
|                                     |
Workspace                           |
| [New Command] [Import SOP ▼] [Export Commands]
| [Refresh]                           |
|-----|
| Commands
Catalog                             |
| +-----+
+   |
| | ID   | Type           | Status       | Triggered By | Created At   |
| |-----|
+   |
| |-----+ +-----+
+   |
| | Command Detail Pane      | | SOP Template Library
| | |
| | - ID: <UUID>              | | - SOP A
| | |
| | - Type: <Type>            | | - SOP B
| | |
| | - Status: Pending/Running | | - ...
| | |
| | - Output: (Log or result) | | [Search SOP ▼]
| | |
| | - Actions: [Execute] [Cancel] | +-----+
+   |
| +-----+
+=====+

```

Component	Function	Data/API Call
CommandWorkspaceHeader	Global actions: new, import/export, refresh	N/A
CommandsCatalogTable	Full-screen table listing commands with filters	GET /v1/commands
CommandDetailPane	Detailed view of selected command including logs and results	GET /v1/commands/{id}, PUT /v1/commands/{id}
SOPTemplateLibrary	Sidebar listing SOP templates for new commands	GET /v1/commands/sop-templates
ExecuteCommandButton	Triggers execution of selected command	POST /v1/commands/{id}/execute
CancelCommandButton	Cancels a running command	POST /v1/commands/{id}/cancel
ExportCommandsButton	Exports command list as JSON or CSV	GET /v1/commands/export
RefreshButton	Reloads command catalog and details	GET /v1/commands

## 5. API Endpoints Project Command Panel

```
+-----+
| Project Command                               |
| [New Command] [Import SOP ▼] [Execute]       |
|-----|
| [Commands List: ID, Type, Status, TriggeredBy] |
+-----+
```

Component	Function	Data/API Call
CommandsListTable	Lists project commands	GET /v1/commands
NewCommandButton	Opens form to create a command via SOP template	N/A
ImportSOPDropdown	Bulk import SOP definitions	POST /v1/commands/import
ExecuteButton	Trigger selected command	POST /v1/commands/{id}/execute
CommandStatusBadge	Displays real-time status of command execution	GET /v1/commands/{id}/status



## 5. API Endpoints. API Endpoints

Service	Method	Path	Description	Auth Scope
Auth Service	POST	/oauth/token	Obtain access token	public
	GET	/oauth/introspect	Validate token	auth.introspect
Profile Service	POST	/v1/users	Create user	profile.write
	GET	/v1/users	List users	profile.read
	PUT	/v1/users/{id}	Update user	profile.write
	DELETE	/v1/users/{id}	Deactivate user	profile.write
Feature Flag Service	GET	/v1/flags	List flags	flag.read
	POST	/v1/flags	Create/update flag	flag.write
Config Service	GET	/v1/config?environment={env}	Fetch config values	config.read
	POST	/v1/config	Create/update config	config.write
Metrics API	POST	/v1/metrics/bulk	Ingest metrics batch	metrics.write
Rate-Limit Service	GET	/v1/ratelimits/{service}/{key}	Get rate limits	ratelimit.read

## 6. Implementation

1. Provision databases (PostgreSQL & TimescaleDB) and Redis for rate-limits.
2. Deploy Auth & Profile services with OIDC flows (Keycloak or custom).
3. Implement Feature Flag and Config services (e.g. Unleash, custom Go service).
4. Build Metrics ingestion endpoint: handle JSON/CSV payloads efficiently.
5. Develop rate-limit sidecar using Redis token bucket algorithm.
6. Secure mTLS and audit logging for all service calls.
7. Write unit, integration, and performance tests (target 5k req/sec).

## 7. Milestones

Milestone	Timeline	Owner
Auth & Profile MVP	Week 1-2	Security Team

Milestone	Timeline	Owner
Feature Flags & Config MVP	Week 3	Backend Team
Metrics Ingestion & Rate-Limit	Week 4	SRE/DevOps Team
UI Panel Implementations	Week 5	Frontend Team
Security Hardening & Auditing	Week 6	Security Team
Performance & Fault Testing	Week 7-8	QA Team

## 8. Gathering Results

- Auth token latency <50 ms; introspection <20 ms.
- User management operations success rate >99.9%.
- Feature flag toggle latency <20 ms.
- Config reads/writes <50 ms.
- Metrics ingestion throughput >20k points/sec.
- Rate-limit accuracy >99.9% under bursts.

## 9. References & Dependencies

- **Integration Blueprints:** appendix\_b\_integration\_blueprints.md
- **UI Blueprints:** appendix\_c\_ui\_blueprints.md
- **DevOps Guide:** \_DEVELOPMENT\_OPERATIONS\_GUIDE.md
- **Security Policies:** VOICE\_ACCESS\_POLICY.md & SOP\_Role\_Assignment.md

## Need Professional Help in Developing Your Architecture?

Please contact me at [sammuti.com](https://sammuti.com) :)