

# Public Service Registry (PSR)

## Business Requirements Document (BRD)

**Version:** 6.0

**Date:** Oct 12, 2026

**Status:** Final Review

**Document Control**

| Version | Date | Author | Changes |
|---|---|---|---|
| 6.0 | Oct 12, 2026 | Cross-Functional Team | Final Review |

## Table of Contents


**1. Overview..... 5**
1.1 Purpose and Scope..... 5
1.1.1 Purpose..... 5
1.1.2 In Scope (Public Service Registry)..... 5
1.1.3 Out of Scope (Other Blocks)..... 6
1.2 Business Value..... 6
1.2.1 For Citizens and Businesses (MyBhutan Portal)..... 6
1.2.3 For Government Service Providers (PSR admin interface)..... 6
1.2.4 For Government Leadership..... 6
1.3 Success Metrics..... 7
**2. Public Service Registry Overview..... 7**
2.1 PSR Overview..... 7
2.1.1 Digital Government Ecosystem..... 7
2.2 PSR Roles and Responsibilities..... 8
2.3 Integration with Building Blocks..... 9
**3. Usage Scenarios..... 10**
**4. Public Service Registry Capabilities and Requirements..... 10**
4.1 PSR Capabilities..... 10
4.1.1 Service Catalog Management Capabilities..... 10
4.1.2 Service Discovery & Search Capabilities..... 11
4.1.3 Service Information Management Capabilities..... 11
4.1.4 Service Classification and Taxonomy Capabilities..... 11
4.1.5 Service Orchestration and Journey Management Capabilities..... 12
4.1.6 Multi-Channel Access Capabilities..... 12
4.1.7 Integration and Interoperability Capabilities..... 12
4.2 PSR Functional Requirements..... 13
4.2.1 FR-1: Service and Systems Catalog Management Requirements..... 13
4.2.2 FR-2: Service and Systems Discovery and Search Requirements..... 15
4.2.3 FR-3: Service and Systems Information Management Requirements..... 15
4.2.4 FR-4: Service and System Classifications and Taxonomy Requirements..... 16
4.2.5 FR-5: Service and Systems Orchestration and Journey Requirements..... 23
4.2.6 FR-7: Multi-Channel Access Requirements..... 24
4.3 Non-Functional Requirements..... 24
4.3.1 Performance Requirements..... 25
4.3.2 Availability and Reliability Requirements..... 25
4.3.3 Scalability Requirements..... 26
4.3.4 Security Requirements..... 26

4.3.5 Usability Requirements....27
4.3.6 Compliance Requirements....28
4.3.7 Operational Requirements....28
**5. Data, Security and Compliance....28**
5.1 Data Model....28
5.1.1 Indicative Service Catalog Data (e.g. PostgreSQL)....29
5.1.2 Indicative Journey and Orchestration Data....31
5.1.3 Indicative Redis Cache Data Structures....33
5.2 Data Security....34
5.2.1 Encryption Standards....34
5.2.2 Key Management....34
5.2.3 Data Retention and Disposal....35
5.3 Security Controls....35
5.3.1 Authentication and Authorization....35
5.3.2 API Security....36
5.3.3 Data Privacy....37
5.4 Compliance Requirements....37
5.4.1 Data Protection Compliance....37
5.4.2 Open Data Standard Compliance....38
5.4.3 Accessibility Standard Compliance....38
5.4.4 API and Interoperability Standards....38
**6. Appendices....39**
Appendix A: Glossary....39
Appendix B: API Reference Summary....39
Appendix C: Error Codes....40
Appendix D: Definitions....41
**7. Annexure 1: Example User Scenarios for Public Service Registry (PSR)....42**
Scenario A: Citizen Discovering Birth Registration Services....42
Scenario B: Business Owner Starting New Enterprise....44
Scenario C: Cross Agency Service Bundle for Marriage....47
Scenario D: Government Service Provider Publishing new Vaccination Program....49
**8. Process Flow....53**
8.1. Overview....53
8.2. Agency Onboarding Process Flow....55
8.3. Agency Onboarding — Roles and Responsibilities....57
8.4. Service Listing Lifecycle....57
8.5. Citizen Discovery and Consumption Flow....58
8.6 Service Management Process Flows....61

8.6.1 Classification Change Request..... 61
8.6.2 Create, Approve and Publish a Service.....62
8.6.3 Define and Publish a Journey..... 63
8.6.4 Update a Published Service..... 64
8.6.5 Roll Back to an Earlier Version..... 64
8.6.6 Service Ownership Transfer.....65
8.6.7 Deprecate and Archive a Service.....66

# 1. Overview

## 1.1 Purpose and Scope

### 1.1.1 Purpose

The Public Service Registry (PSR) is a centralized catalog and discovery platform that maintains comprehensive information about all government services available to citizens, businesses, and other government entities. Operating as the "service directory" for digital government, PSR enables easy discovery, access, and management of government services across all agencies and departments.

### 1.1.2 In Scope (Public Service Registry)

- **Service catalog management**: Create, update, publish, and retire service definitions.
- **Service and Agency Mapping:** Map agencies with the services and cross agency service mapping. Agencies should fetch from agency registries.
- **Service discovery and search**: faceted filters for category, agency, cost, location, delivery channel, Location-based search and Eligibility-based filtering
- **Service metadata management**: Comprehensive service information including eligibility, requirements, costs, and delivery channels.
- **Service classification and taxonomy**: Hierarchical categorization and life event grouping.
- **Service relationships**: Prerequisites, bundles, dependencies, and alternatives.
- **Multi-channel service access**: MyBhutan Portal for citizens and businesses, the PSR admin interface for agency staff, and API integration.
- **System Catalog Management:** Create, update, delete systems.
- **Service and System Mapping:** Mapping of services with the systems. One to many and many to one relationship between services and systems.
- **System Capabilities Catalogue:** Adequately mapped system and systems catalogue with system capabilities up to 5 layers.
- **Journey orchestration:** Journey management and composite service workflows.
- Integration and Interoperability: Enables secure API integration, real-time updates, data exchange, federation, and backward-compatible system connectivity.
- Wherever service mapping is required, it should be implementable.

### 1.1.3 Out of Scope (Other Blocks)

- **Service delivery execution** (handled by individual service systems)
- **Payment processing** (handled by Payment Block)
- **Document management and storage** (handled by Document Management Block)
- **Identity verification** (handled by Digital Identity Block)
- **Notifications and alerts** (handled by Notification Block)
- **Citizen-facing portal** (the MyBhutan Portal, displays PSR data to citizens and businesses)

**Key Distinction:** PSR is a service information and discovery platform, not a service delivery platform. It provides comprehensive information about services and enables access, but the actual service delivery is handled by respective service systems.

## 1.2 Business Value

### 1.2.1 For Citizens and Businesses (MyBhutan Portal)

- **Easy service discovery**: Find government services quickly through intuitive search
- **Comprehensive information**: Understand requirements, costs, and timelines before applying
- **Better understanding of requirements :** Enables clearer identification and alignment of citizen service needs and expectations.
- **Progress tracking**: Monitor application status across multiple services
- **Bundled services**: Access related services together (e.g., business registration package)

### 1.2.3 For Government Service Providers (PSR admin interface)

- **Centralized service management**: Manage service information in one place
- **Reduced duplication**: Identify overlapping services across agencies
- **Faster service launch**: Standardized process for publishing new services

### 1.2.4 For Government Leadership

- **Inter-agency coordination**: Facilitate cross-agency service delivery
- **Digital transformation**: Foundation for integrated government services

### 1.3 Success Metrics

| Category | Metric | Target |
|---|---|---|
| **Availability** | System Uptime | ≥ 99.9% |
| **Performance** | Search Response Time | ≤ 200ms (95%) |
| **Performance** | Service Detail Load Time | ≤ 500ms (95%) |
| **User Experience** | Service Discovery Success Rate | ≥ 90% |
| **User Experience** | User Satisfaction Score | ≥ 4.2/5.0 |
| **Coverage** | Services Cataloged | 100% of citizen-facing services |
| **Adoption** | Monthly Active Users | Target growth: 10% quarterly |
| **Data Quality** | Service Information Completeness | ≥ 95% |
| **Data Quality** | Information Currency | < 30 days staleness |
| **Integration** | API Response Time | ≤ 100ms (95%) |

# 2. Public Service Registry Overview

## 2.1 PSR Overview

### 2.1.1 Digital Government Ecosystem

The PSR operates as a foundational building block within the broader digital government ecosystem:

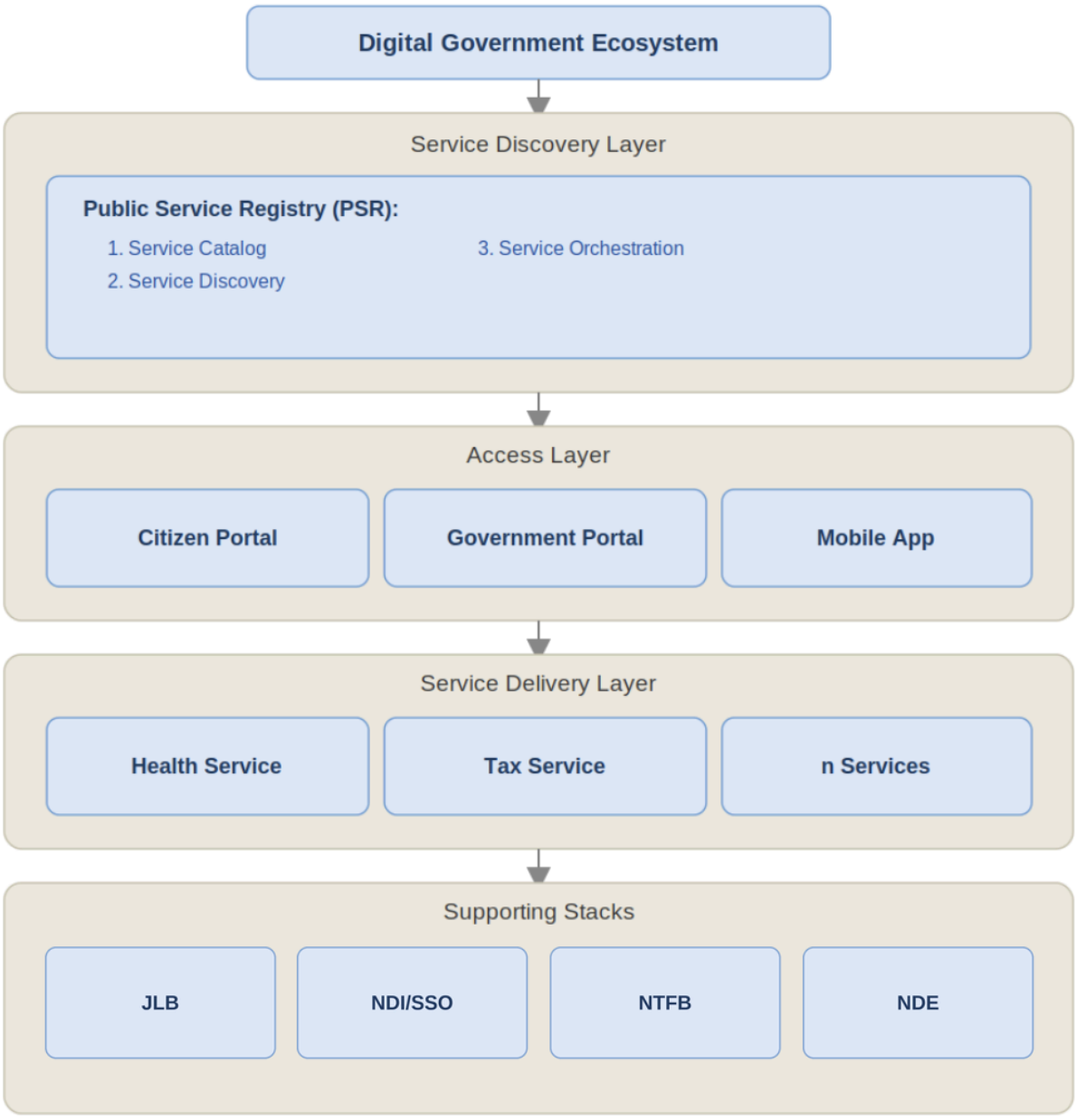



*Figure 1: Example Overview of the PSR*

## 2.2 PSR Roles and Responsibilities

**Primary Responsibility:** Centralized service catalog and discovery platform for all government services

**PSR Receives:**

- Service definitions from government agencies
- Service updates and changes from service owners
- Search queries from citizens and businesses

- Service usage data from delivery systems

**PSR Provides:**

- Comprehensive service catalog with detailed information
- Multi-modal search and discovery capabilities
- Service recommendations based on user profile and context
- Service orchestration for complex journeys
- APIs for third-party integration

**PSR Manages:**

- Service lifecycle (draft → review → published → deprecated → archived)
- Service metadata (eligibility, requirements, costs, timelines)
- Service relationships (prerequisites, bundles, alternatives)
- Service classifications and taxonomies

**Critical Boundary:** PSR does not execute service delivery or process applications. It provides information and facilitates discovery and access, but actual service fulfillment is handled by respective service systems.

## 2.3 Integration with Building Blocks

| Building Block | Integration Purpose | Pattern |
|---|---|---|
| **NDI/SSO (Identity Provider)** | Authenticate users of the PSR admin interface (Super Admin, Agency Admin, Service Owner, Reviewer, Agency Head); PSR then applies its own role-based, agency-scoped access | Federated (OIDC) |
| **MyBhutan portal** | Public area: browse, search and view services. Signed-in area: sends citizen and business profile attributes to PSR and receives eligibility verdicts; shows service details, recommendations, journey progress and quality scores; hands users over to agency systems | REST API |
| **MDMB** | Look up each agency's official golden record and hierarchy when the agency is created; provide location records (Dzongkhag, Gewog) for service locations | REST API via NDE |

| | | |
|---|---|---|
| **NDE** | Publish service events (published, updated, deprecated, archived, ownership changed) to subscribing blocks, and carry requests between PSR and MDMB, UCS | Event-driven + REST API |
| **Notification Block** | Send announcements, reminders, next-step alerts and status notifications to citizens and agency staff | Async via NDE + REST API |
| **Journaling Block** | Record the audit trail of service changes, approvals, ownership transfers and access logs for compliance | Async via NDE |
| **DMS** | Store service media files (images, videos, PDFs) in the Document Store (DOCS) and return references to PSR | REST API |
| **UCS** | Read the service catalog, owning agency and service-system mapping to show services in support case forms and route cases to the right agency | REST API via NDE |
| **Observability Platform** | Monitor and report PSR health, performance and availability | REST API |

# 3. Usage Scenarios

**Refer to Annexure for Example Scenarios**

# 4. Public Service Registry Capabilities and Requirements

## 4.1 PSR Capabilities

### 4.1.1 Service Catalog Management Capabilities

- **Service lifecycle management** with workflow-driven state transitions (draft → review → published → deprecated → archived)
- **Version control** for service definitions with rollback and audit history capability
- **Rich metadata management** including eligibility, requirements, fees/costs, processing timelines, delivery channels, required documents, legal basis, and service locations
- **Service ownership tracking** with clear assignment to government departments/agencies
- **Government Agency management** allowing the Super Admin to create and manage agencies and designate an Agency Admin for each agency

- **Agency onboarding** allows the Super Admin to record onboarding requests received from agencies and create the agency and its Agency Admin in PSR.
- **Role-based service management** allowing designated Agency Admins to create, update, submit for approval, and publish services within their assigned agency
- **Agency-Service Mapping** with Service Providing Agencies using MDM
- Government Agency system lifecycle management
- Systems and services mapping process
- **Bulk import/export** of service definitions for migration and backup
- **Service templates** for standardized service creation across different categories
- **Approval workflows** with configurable review and authorization chains

### 4.1.2 Service Discovery & Search Capabilities

- **Faceted search** with filters for category, agency, cost, location, delivery channel
- **Location-based search** for services available in specific geographic areas
- **Eligibility-based filtering** that shows only services user qualifies for

### 4.1.3 Service Information Management Capabilities

- **Comprehensive service details** including eligibility, prerequisites, documents, process steps, costs, and timelines
- **Service relationships** (prerequisites, related services, bundles, alternatives, dependencies)
- **Dynamic content** that adapts based on user profile and context
- **Rich media support** (images, videos, infographics, PDFs)
- **Interactive elements** (calculators, eligibility checkers, document checklists)
- **Contact information** with multiple channels (phone, email, chat, in-person)
- **Service status visibility** see whether a service is active, deprecated, under maintenance, or upcoming

### 4.1.4 Service Classification and Taxonomy Capabilities

- **Hierarchical categorization** (e.g., Health → Preventive Care → Immunization)

- **Life event grouping** (birth, education, employment, marriage, business, retirement, death)
- **Target audience segmentation** (citizens, businesses, government entities, NGOs)
- **Service type classification** (G2C, G2B, G2G, G2E)
- **Multi-dimensional tagging** for flexible organization
- **Custom taxonomies** per agency requirements
- Classification change requests, allowing agencies to request new categories or life events

### 4.1.5 Service Orchestration and Journey Management Capabilities

- **Service bundling** for related services (e.g., birth registration package)
- **Sequential workflows** with dependency management
- **Parallel service processing** support
- **Conditional routing** based on user selections or eligibility
- **Progress tracking** across multiple services
- **Integration with external workflow systems for orchestration visibility and status tracking**

### 4.1.6 Multi-Channel Access Capabilities

- **PSR admin interface**: a responsive web application where agency staff and the Super Admin manage agencies, services, journeys and the taxonomy
- **API access** for third-party integration

### 4.1.7 Integration and Interoperability Capabilities

- **RESTful APIs** with comprehensive documentation
- **GraphQL support** for flexible querying
- **Webhook notifications** for real-time updates
- **Event-driven integration** via Message Broker (e.g. Kafka)
- **Standard data formats** (JSON, XML, CPSV-AP)
- **Federation support** for distributed catalogs
- **Bulk data exchange** for synchronization
- **API versioning** for backward compatibility

- **Integration with external Workflow/BPM systems**

## 4.2 PSR Functional Requirements

| Particulars | Definition |
|---|---|
| **MUST** | Critical for system operation, legal or regulatory requirement or business process requirement |
| **SHOULD** | Important for user experience, significant business value |
| **COULD** | Nice to have, future enhancements |
| **WON'T** | Out of scope for current phase |

### 4.2.1 FR-1: Service and Systems Catalog Management Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|
| **Service Catalog Management** | | | |
| FR-1.1 | Agency onboarding request | MUST | The onboarding request is received. The Super Admin records the request in PSR with: agency name, parent agency, MDMB agency record, business contact, and the person nominated as Agency Admin. The Super Admin confirms the agency's MDMB record, creates the agency and assigns the Agency Admin role to the nominated person. |
| FR-1.2 | Government Agency and Agency Admin management | MUST | Super Admin can create, update and manage government agency records and assign/designate an Agency Admin for each agency; Agency Admin access is restricted to their assigned agency. the Agency Admin assigns Service Owner, Reviewer, Agency Head and Journey Owner roles within their agency. |
| FR-1.3 | Create and edit service definitions | MUST | Support all metadata fields defined in data model; rich text editor for descriptions, unique, permanent |

| | | | |
|---|---|---|---|
| | with comprehensive metadata | | service ID generated by PSR at creation that never changes or is reused (including on versioning, ownership transfer or recategorisation) |
| FR-1.4 | Implement service lifecycle management (draft → review →approve/reject →updated/versioned → published → deprecated → archived) | MUST | State transitions with validation; automated workflows; audit trail of state changes, service is locked for editing while in review and unlocked when returned to draft on rejection. |
| FR-1.5 | Version control and roll back for service definitions | MUST | Track all changes; compare versions; one-click rollback to any previous version; rollback capability; rollback impact analysis (show dependent services/journeys affected); rollback approval workflow for major versions; change history with user attribution and reason for rollback |
| FR-1.6 | Service approval workflow with configurable rules | MUST | Multi-level approval; role-based routing; notifications; approval history; rejection with reasons. |
| FR-1.7 | Bulk import/export of service data | MUST | Support CSV, Excel, JSON formats, (Word and PDF formats for reporting, viewing); validation on import(security scanning); error reporting; template download. |
| FR-1.8 | Service templates for standardized creation | SHOULD | Templates per service category; customizable fields; pre-filled common data; template versioning |

| | | | |
|---|---|---|---|
| FR-1.9 | Service archival with retention management | MUST | Configurable retention period; automated archival; restore capability; compliance with regulations |
| FR-1.10 | Agency-based service management | MUST | Designated Service owner can create, update and publish services belonging to their assigned agency, subject to the configured lifecycle and approval rules |
| FR-1.11 | Agency-Service mapping | MUST | PSR shall associate each service with its responsible/service-providing agency using the defined MDM structure; mapping changes shall be authorized and auditable |

### 4.2.2 FR-2: Service and Systems Discovery and Search Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|
| **Service Discovery and Search Requirements** | | | |
| FR-2.1 | Full-text search across all service metadata | MUST | Search service names, descriptions, keywords; relevance ranking; < 200ms response time (95%) |
| FR-2.2 | Faceted search with multiple filters | MUST | Filter by category, agency, cost, location, delivery channel, Service type, audience; dynamic filter options; filter combinations |
| FR-2.3 | Location-based service discovery | MUST | Filter services by district/location; "services near me"; geospatial queries, service locations selected from MDMB Location golden records (e.g., Dzongkhag, Gewog), not entered as free text. |
| FR-2.4 | Eligibility-based filtering | SHOULD | Show only services user qualifies for; based on profile attributes; dynamic eligibility check |

### 4.2.3 FR-3: Service and Systems Information Management Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|

| | | | |
|---|---|---|---|
| FR-3.1 | Display comprehensive service details | MUST | Show all metadata fields; structured layout; mobile-responsive; print-friendly |
| FR-3.2 | Service relationship management | MUST | Define prerequisites, related services, bundles, alternatives; visual relationship graph |
| FR-3.3 | Rich media content support | MUST | Upload images, videos, PDFs; media library; version control for media(stored in Document Store). |
| FR-3.4 | Interactive service elements | SHOULD | Eligibility checkers; fee calculators; document checklists; embedded forms |
| FR-3.5 | Contact information management for each service | MUST | Display multiple contact channels (phone, email, chat, in-person) |
| FR-3.6 | Service status visibility | SHOULD | Users can see whether a service is active, deprecated, under maintenance, or upcoming. Upcoming is set automatically from the approved effective date. Under maintenance is set by the Agency Admin with a reason and expected end date. Upcoming and under-maintenance services are visible but cannot be started. |

### 4.2.4 FR-4: Service and System Classifications and Taxonomy Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|
| FR-4.1 | Create and manage hierarchical category structure for services and level of | MUST | • Support minimum 5 levels of category depth (e.g., Health → Preventive Care → Immunization → Child Vaccination → HPV)<br>For systems - e.g. 5 layer menu structure. |

| | | | |
|---|---|---|---|
| | capabilities for systems | | • Parent-child relationship validation with referential integrity<br>• Circular reference prevention (category cannot be its own ancestor)<br>• Ability to move categories within hierarchy with dependent service updates<br>• Display category breadcrumb navigation (Home > Health > Preventive Care)<br>• Category reordering within same level via drag-and-drop or display_order field<br>• Soft delete (show affected services before deletion) |
| FR-4.2 | Define and manage life event taxonomy | MUST | • Pre-configured life events: Birth, Education, Employment, Marriage, Business Start/Operations, Property, Retirement, Death, Immigration, Health Crisis<br>• Custom life event creation by the Super admin<br>• Life event metadata: name, description, target audience, icon, typical age range, associated categories<br>• Service-to-life-event mapping (many-to-many relationship)<br>• Life event-based service discovery page (e.g., "I'm having a baby" shows all related services)<br>• Life event lifecycle stages support (e.g., Marriage: Before, During, After)<br>• Life event activation/deactivation without deleting data |

| | | | |
|---|---|---|---|
| | | | • Import/export life event configurations<br>• Validation: Each service must be mapped to at least one life event or marked as "General Service" |
| FR-4.3 | Multi-dimensional tagging system | SHOULD | • Free-form tag creation by service owners (subject to approval workflow)<br>• Tag categories: Keywords, Topics, Sectors, Beneficiaries, Special Programs<br>• Service can have unlimited tags across multiple tag categories<br>• Tag search and filtering in service discovery (e.g., "tags:covid,vaccination,children")<br>• Tag auto-suggestions based on existing tags and service content<br>• Tag management: merge duplicate tags, rename tags (propagate to all services), delete unused tags<br>• Tag normalization: lowercase, trim whitespace, handle synonyms |
| FR-4.4 | Service type classification enforcement | MUST | • Mandatory service type selection during service creation: G2C (Government to Citizen), G2B (Government to Business), G2G (Government to Government), G2E (Government to Employee)<br>• Service type-specific metadata templates (e.g., G2B requires business registration fields)<br>• Service type-based filtering in search |

| | | | |
|---|---|---|---|
| | | | • Service type displayed prominently on service detail page<br>• Separate service catalogs/views per service type for targeted user experience<br>• Validation rules per service type (e.g., G2C must have citizen eligibility criteria)<br>• Service type change tracking in audit logs (changing type may impact eligibility)<br>• Default landing pages per service type (e.g., the MyBhutan Portal's business view shows only G2B services)<br>• Cross-type service relationships (e.g., G2C birth registration may link to G2B child business account) |
| FR-4.5 | Target audience segmentation | MUST | • Multi-select target audiences: Citizens, Businesses, Government Entities, NGOs, Foreign Nationals, Students, Seniors, Persons with Disabilities<br>• Audience-specific service views with personalized homepage<br>• Audience profile attributes for eligibility checking (e.g., age, income, disability status)<br>• Audience-based recommendations (e.g., show senior citizen services to users aged 60+)<br>• Audience exclusions (mark services as "Not applicable to" specific audiences) |

| | | | |
|---|---|---|---|
| | | | • Audience priority ranking for services serving multiple audiences (primary vs secondary)<br>• Dynamic audience detection from user profile (NDI integration)<br>• Validation: Each service must have at least one target audience<br>• Audience filter in search: "Show me services for businesses" |
| FR-4.6 | Agency-based categorization and hierarchy | MUST | • Complete government agency hierarchy: Ministries → Departments → Divisions → Sections<br>• Agency metadata: name, type, parent agency, contact info, website, logo<br>• Service-to-agency mapping (each service owned by exactly one agency)<br>• Agency service portfolio page (all services offered by an agency)<br>• Inter-agency service relationships (services requiring coordination across agencies)<br>• Agency Admin role (manage only their agency's services)<br>• Agency-based service search and filtering (e.g., "Show all Ministry of Health services")<br>• Agency hierarchy changes reflected in service ownership (handle mergers, restructuring If two departments merge, the PSR can bulk-reassign all services to the new agency.) |

| | | | |
|---|---|---|---|
| | | | • Agency contact information displayed on service detail pages<br>• Agency-level approval workflows (agency head must approve before system-wide publication)<br>• Validation: Cannot delete agency with active services (must reassign first) |
| FR-4.7 | Cross-taxonomy relationships and mappings | SHOULD | • Define relationships between categories, life events, and agencies (e.g., "Birth" life event maps to "Health" and "Civil Registration" categories)<br>• Category-to-life-event recommendations (when user browses "Health > Child Services", suggest "Birth" life event)<br>• Agency-to-category specialization (identify which agencies specialize in which categories)<br>• Taxonomy consistency validation (prevent orphaned services not mapped to any taxonomy)<br>• Bulk taxonomy operations (assign multiple services to same category/life event)<br>• Taxonomy versioning<br>• Visual taxonomy explorer (interactive tree/graph view of all relationships)<br>• Export taxonomy structure for external systems (JSON, XML, CSV) |

| | | | |
|---|---|---|---|
| FR-4.8 | Taxonomy navigation optimization | MUST | • Taxonomy-based breadcrumb navigation (e.g., Home > Health > Preventive Care > Immunization > Service Name)<br>• Taxonomy-based mega menu navigation (dropdown showing category hierarchy)<br>• Taxonomy-based URL slugs (e.g., /services/health/preventive-care/immunization)<br>• Mobile-optimized taxonomy navigation (e.g collapsible menus, swipe gestures) |
| FR-4.9 | Service Ownership Transfer Management | MUST | • Support deliberate, permanent transfer of service ownership between agencies, departments, or divisions while maintaining historical records and user continuity<br>• Transfer workflow:System provides formal transfer workflow: Submit Request → Current Owner Approves → New Owner Accepts → Execute Transfer<br>• Required justification: Transfer requires mandatory justification field (e.g., "Government restructuring", "Agency mandate change as per Cabinet decision<br>• Historical archive: Previous ownership is archived, not deleted.<br>• Department/Division level: Transfers supported at any level: Agency → Agency, Department → Department, Division → Division, cross-level (Division → Department) |

| | | | |
|---|---|---|---|
| | | | • Bulk transfer: Support transferring multiple services at once (e.g., all services from Department X to Department Y)<br>• Notification:All service provider notified of transfer: service owners, agency admins, users with saved services, users with pending applications<br>• Audit trail: Complete audit log: Transfer ID, service ID, from_agency, to_agency, from_department, to_department, requested_by, approved_by_current, approved_by_new, effective_date, justification, status |
| FR-4.10 | Classification change request | MUST | Agency Admin can request a new category or life event, giving name, parent, justification and affected services. Super Admin checks for an existing match, shared or agency-specific use, hierarchy rules and metadata, then approves or rejects with reasons. Approved items update the taxonomy version. The agency is notified, the service cannot be submitted until the request is decided, and all actions are audited. |

### 4.2.5 FR-5: Service and Systems Orchestration and Journey Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|
| FR-5.1 | Create service bundles for life events | MUST | Group related services into a journey; define service sequence; manage journey metadata; support life event templates. Each journey is owned by a Service Owner of the owning agency; the journey owner can |

| | | | |
|---|---|---|---|
| | | | include only published services and cannot edit services owned by other agencies; participating agencies are notified when their services are included |
| FR-5.2 | Define service dependencies and sequence | MUST | Define prerequisites and dependencies; define service order; prevent users from starting services before required prerequisites are met. |
| FR-5.3 | Conditional service routing | SHOULD | Define different service paths based on user responses or eligibility; support branching and dynamic journey paths. |
| FR-5.4 | Parallel service processing | SHOULD | Allow independent services to proceed in parallel; track each branch; show overall journey progress. |
| FR-5.5 | Mixed sequential-parallel journeys | SHOULD | Support journeys combining sequential steps, parallel branches, and conditional paths. |
| FR-5.6 | Partial rollback in journeys | SHOULD | Allow eligible journey steps or branches to be restarted without affecting other completed branches. |

### 4.2.6 FR-7: Multi-Channel Access Requirements

| ID | Requirement | Priority | Acceptance Criteria |
|---|---|---|---|
| FR-6.1 | Responsive PSR admin interface | MUST | Responsive design (desktop, tablet, mobile); consistent experience; <500ms page load; used by agency staff and the Super Admin; citizen-facing display is provided by the MyBhutan Portal |
| FR-6.2 | RESTful API access | MUST | Comprehensive API coverage; rate limiting; documentation; versioning; authentication |

## 4.3 Non-Functional Requirements

This section contains minimal NFR from the business perspective. However, it should cover all the NFR defined under the NFR SOP maintained by the DSD division.

### 4.3.1 Performance Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Search response time | ≤ 200ms (95%) | Query execution timing from database/search index |
| Service detail page load | ≤ 500ms (95%) | Full page render time including media |
| API response time | ≤ 100ms (95%) | End-to-end API call timing |
| Search index update | ≤ 5 minutes | Time from service update to search availability |
| Throughput - Search | ≥ 2,000 req/s per instance | Load testing with realistic queries |
| Throughput - API | ≥ 5,000 req/s per instance | Sustained load testing |
| Database query time | ≤ 50ms (95%) | Database query execution time |
| Cache hit ratio | ≥ 90% | Cache effectiveness for frequently accessed services |
| Embeddable widget initial load | ≤ 300ms (95%) | Time from widget render start to interactive state |
| Parallel workflow branch synchronization | ≤ 500ms (95%) | Time from last parallel branch completion to join point execution |

### 4.3.2 Availability and Reliability Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Service availability | ≥ 99.9% monthly | Uptime monitoring (max 43.2 min downtime/month) |
| Scheduled maintenance | ≤ 4 hours/month | Planned maintenance windows |

| | | |
|---|---|---|
| Mean Time To Recovery (MTTR) | ≤ 30 minutes | Automated recovery and manual intervention |
| Mean Time Between Failures (MTBF) | ≥ 720 hours (30 days) | System stability tracking |
| Disaster Recovery RTO | ≤ 2 hours | Cross-region failover testing |
| Disaster Recovery RPO | ≤ 30 minutes | Data replication lag measurement |
| Error rate | < 0.5% | Failed requests / total requests |
| Data loss tolerance | Zero | Backup and replication verification |

### 4.3.3 Scalability Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Horizontal scalability | Linear scaling to 50+ instances | Load testing with gradual increase |
| Service catalog capacity | Support 50,000+ services | Database capacity and search index testing |
| Concurrent user capacity | 100,000 simultaneous users | Load testing with realistic user behavior |
| Search index size | 10GB+ indexed data | Elasticsearch cluster sizing |
| Media storage | 1TB+ (expandable) | Cloud storage with auto-scaling |
| Auto-scaling response time | ≤ 3 minutes | Kubernetes HPA metrics observation |
| Database growth | 50% annual growth support | Capacity planning and partitioning |

### 4.3.4 Security Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Data encryption at rest | AES-256 | All RDBMS (e.g. PostgreSQL), NoSql( e.g. MongoDB), File System (S3 or similar) encrypted |

| | | |
|---|---|---|
| Data encryption in transit | TLS 1.3 minimum | All network communication encrypted |
| Authentication | OAuth 2.0, OIDC | Integration with NDI |
| Authorization | RBAC for admin, API key for integration | Role-based access control testing |
| API security | API Gateway with rate limiting | Request throttling, DDoS protection |
| Input validation | All inputs validated | SQL injection, XSS prevention testing |
| Vulnerability patching | Critical: 7 days, High: 30 days | Security update tracking |
| Penetration testing | Quarterly | Third-party security audit |
| Security headers | Standard security headers | OWASP security header compliance |
| Data privacy | GDPR, local privacy laws | Privacy impact assessment |

### 4.3.5 Usability Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Learnability | New users can find service < 3 minutes | User testing with novice users |
| Task completion rate | ≥ 90% for service discovery | User testing scenarios |
| User satisfaction | ≥ 4.0/5.0 rating | User surveys (System Usability Scale) |
| Error rate | < 5% user errors | User testing error tracking |
| Help documentation | Comprehensive help available | Documentation coverage review |
| Mobile usability | Consistent mobile experience | Mobile device testing (iOS, Android) |

| | | |
|---|---|---|
| Accessibility | WCAG 2.1 AA compliance | Automated accessibility testing + manual audit |

### 4.3.6 Compliance Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Data protection | GDPR, local privacy laws | Legal review and compliance audit |
| Open data standards | CPSV-AP (EU standard) | Schema validation |
| API standards | OpenAPI 3.0 specification | API documentation compliance |
| Accessibility | WCAG 2.1 AA | Accessibility audit |
| Data retention | 7 years for audit logs | Archival process validation |
| Service metadata | ISO 15836 (Dublin Core) | Metadata schema validation |
| Interoperability | GovStack building block standards | Integration testing |

### 4.3.7 Operational Requirements

| Requirement | Target | Measurement Method |
|---|---|---|
| Monitoring coverage | 100% of critical components | Monitoring dashboard verification |
| Alerting response | < 5 minutes for critical | Alert delivery testing |
| Log retention | 90 days (operational), 7 years (audit) | Log archival verification |
| Backup frequency | Daily incremental, weekly full | Backup schedule verification |
| Backup restoration | < 1 hour for critical data | Restoration drill |
| Deployment frequency | Weekly (non-disruptive) | CI/CD pipeline testing |
| Rollback capability | < 15 minutes | Rollback procedure testing |

# 5. Data, Security and Compliance

## 5.1 Data Model

This section describes data model for the services and FRs but not limited to it;

### 5.1.1 Indicative Service Catalog Data (e.g. PostgreSQL)

```
sql
-- Services table
CREATE TABLE services (
    service_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    service_name VARCHAR(500) NOT NULL,
    service_name_i18n JSONB,
    service_type VARCHAR(20) NOT NULL, -- G2C, G2B, G2G, G2E
    category_id UUID REFERENCES categories(category_id),
    agency_id UUID REFERENCES agencies(agency_id),
    description TEXT,
    description_i18n JSONB,
    eligibility_criteria JSONB,
    required_documents JSONB,
    processing_time JSONB, -- {min_days, max_days, average_days}
    cost JSONB, -- {amount, currency, fee_structure}
    delivery_channels TEXT[], -- ['online', 'service_center', 'mobile']
    service_locations UUID[], -- Array of location IDs
    contact_info JSONB,
    legal_basis TEXT, -- Laws/regulations governing this service
    service_url VARCHAR(500),
    status VARCHAR(20) NOT NULL, -- DRAFT, UNDER_REVIEW, PUBLISHED,
DEPRECATED, ARCHIVED
    version INT DEFAULT 1,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    created_by UUID REFERENCES users(user_id),
    updated_by UUID REFERENCES users(user_id),
    published_at TIMESTAMP,
    deprecated_at TIMESTAMP
);
```

```
-- Categories table (hierarchical)
CREATE TABLE categories (
    category_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    category_name VARCHAR(200) NOT NULL,
    category_name_i18n JSONB,
    parent_category_id UUID REFERENCES categories(category_id),
    category_level INT,
    display_order INT,
    icon_url VARCHAR(500),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- Agencies table
CREATE TABLE agencies (
    agency_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    agency_name VARCHAR(300) NOT NULL,
    agency_name_i18n JSONB,
    agency_type VARCHAR(50), -- Ministry, Department, Agency
    parent_agency_id UUID REFERENCES agencies(agency_id),
    contact_email VARCHAR(255),
    contact_phone VARCHAR(50),
    website_url VARCHAR(500),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- Service Relationships
CREATE TABLE service_relationships (
    relationship_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    source_service_id UUID REFERENCES services(service_id),
    target_service_id UUID REFERENCES services(service_id),
    relationship_type VARCHAR(50), -- PREREQUISITE, RELATED, BUNDLE,
ALTERNATIVE, DEPENDENCY
    description TEXT,
```

```
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(source_service_id, target_service_id, relationship_type)
);
-- Service Versions (audit trail)
CREATE TABLE service_versions (
    version_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    service_id UUID REFERENCES services(service_id),
    version_number INT NOT NULL,
    service_data JSONB NOT NULL, -- Complete service definition at this version
    change_description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    created_by UUID REFERENCES users(user_id)
);
-- Service Approvals
CREATE TABLE service_approvals (
    approval_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    service_id UUID REFERENCES services(service_id),
    version_number INT,
    approver_id UUID REFERENCES users(user_id),
    approval_status VARCHAR(20), -- PENDING, APPROVED, REJECTED
    comments TEXT,
    approved_at TIMESTAMP,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

### 5.1.2 Indicative Journey and Orchestration Data

```
-- Journeys table
CREATE TABLE journeys (
    journey_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    journey_name VARCHAR(300) NOT NULL,
    journey_name_i18n JSONB,
    description TEXT,
```

```
    description_i18n JSONB,
    life_event VARCHAR(100), -- birth, marriage, business_start, etc.
    target_audience VARCHAR(50), -- citizens, businesses, government
    total_estimated_time_days INT,
    total_estimated_cost DECIMAL(15,2),
    status VARCHAR(20) DEFAULT 'ACTIVE', -- ACTIVE, INACTIVE, ARCHIVED
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- Journey Steps
CREATE TABLE journey_steps (
    step_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    journey_id UUID REFERENCES journeys(journey_id),
    service_id UUID REFERENCES services(service_id),
    step_number INT NOT NULL,
    required BOOLEAN DEFAULT TRUE,
    prerequisite_steps UUID[], -- Array of step_ids that must be completed first
    parallel_allowed BOOLEAN DEFAULT FALSE,
    estimated_time_days INT,
    step_description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- User Journey Progress
CREATE TABLE user_journey_progress (
    progress_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL,
    journey_id UUID REFERENCES journeys(journey_id),
    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completed_at TIMESTAMP,
    current_step INT,
```

```
    progress_percentage DECIMAL(5,2),
    status VARCHAR(20) DEFAULT 'IN_PROGRESS', -- IN_PROGRESS,
COMPLETED, ABANDONED
    steps_status JSONB -- Array of {step_id, status, started_at, completed_at}
);
```

### 5.1.3 Indicative Redis Cache Data Structures

```
# Service Detail Cache
Key: service:{service_id}
Type: String (JSON)
TTL: 5 minutes
Value: Complete service JSON
# Search Results Cache
Key: search:{query_hash}:{filters_hash}
Type: String (JSON)
TTL: 10 minutes
Value: Array of service_ids with metadata
# Popular Services Cache
Key: popular:services:{category_id}
Type: Sorted Set
Score: Popularity score
Members: service_ids
TTL: 1 hour
# User Preferences Cache
Key: user:{user_id}:preferences
Type: Hash
TTL: 30 minutes
Fields: saved_services, recent_searches
# Real-Time Counters
Key: service:{service_id}:views:today
Type: String (Integer)
```

TTL: 24 hours

Value: View count

*# Session Data*

Key: session:{session_id}

Type: Hash

TTL: 1 hour (sliding)

Fields: user_id, current_service, search_history, timestamp

## 5.2 Data Security

### 5.2.1 Encryption Standards

- Data at Rest:
  - PostgreSQL: Transparent Data Encryption (TDE) with AES-256
  - MongoDB: Encryption at filesystem level
  - S3/Cloud Storage: Server-side encryption (SSE-KMS)
  - Search: Encrypted at-rest via storage encryption
- Data in Transit:
  - External Traffic: TLS 1.3 only
  - Internal Service-to-Service: Mutual TLS (mTLS) via Istio
  - Database Connections: TLS for PostgreSQL, MongoDB, Search
  - Kafka: TLS encryption enabled
- Sensitive Data Handling:
  - User PII (from NDI): Not stored in PSR, fetched on-demand with caching (short TTL)
  - API Keys: Hashed with SHA-256
  - Admin credentials: Delegated to NDI

### 5.2.2 Key Management

**Key Hierarchy:**

Root Key (KMS - AWS KMS, Azure Key Vault, HashiCorp Vault)

↓

Data Encryption Keys (DEK) - per service, rotated every 90 days

↓

Individual data item encryption

- Key Rotation:
    - DEKs rotated every 90 days
    - API keys rotated every 180 days
    - Re-encryption of data with new keys within 30 days

### 5.2.3 Data Retention and Disposal

| Data Type | Retention Period | Disposal Method |
|---|---|---|
| Published services | Indefinite (or until deprecated) | Soft delete → Archive after 2 years |
| Deprecated services | 5 years post-deprecation | Archive to cold storage |
| User ratings/reviews | Service lifetime + 1 year | Anonymization, then deletion |
| Analytics data | 3 years | Aggregation, then deletion of raw data |
| Search logs | 1 year | Aggregation, then deletion |
| Audit logs | 7 years | Archival to cold storage via JLB |
| Cache data | As per TTL | Automatic Redis expiration |
| Complaints | 7 years post-resolution | Archival to cold storage |

## 5.3 Security Controls

### 5.3.1 Authentication and Authorization

- Authentication:
    - User authentication delegated to NDI (citizens, businesses)
    - Admin authentication via NDI/SSO
    - API authentication via API keys (for system integrations)
- Authorization:
    - Public Access: Service search and viewing (no authentication required
    - Authenticated Access: Personalized recommendations, saved services, journey tracking
    - Admin Access: Service creation, editing, approval, analytics
    - API Access: Rate-limited based on API key tier
    - Role-Based Access Control (RBAC):

| Role | Permissions |
|---|---|
| Citizen/Business | View services, search, save favorites, rate services |
| Deprecated services | 5 years post-deprecation |
| Service Owner | Create/edit services for their agency, view analytics for their services |
| Service Approver | Review and approve services |
| Agency Admin | Manage all services for their agency, manage team members |
| System Admin | Full access to all services, system configuration, user management |
| API Client | Read-only access to service catalog, analytics (based on tier) |

### 5.3.2 API Security

- Rate Limiting:

| Client Type | Rate Limit | Burst | Window |
|---|---|---|---|
| Anonymous IP | 100 req/min | 200 | 1 minute |
| Authenticated User | 600 req/min | 1200 | 1 minute |
| Basic API Client | 1000 req/min | 2000 | 1 minute |
| Premium API Client | 10000 req/min | 20000 | 1 minute |

- Input Validation:
    - JSON schema validation for all API requests
    - SQL injection prevention (parameterized queries, ORM)
    - XSS prevention (output encoding, CSP headers)
    - File upload validation (type, size, malware scanning)
- Security Headers:

    Strict-Transport-Security: max-age=31536000; includeSubDomains

    X-Content-Type-Options: nosniff

    X-Frame-Options: SAMEORIGIN

    Content-Security-Policy: default-src 'self'; img-src 'self' https:; script-src 'self' 'unsafe-inline'

X-XSS-Protection: 1; mode=block

Referrer-Policy: strict-origin-when-cross-origin

### 5.3.3 Data Privacy

**Privacy by Design:**

- Minimal data collection (only necessary information)
- Data minimization in search indexes
- Anonymization of analytics data
- User consent for optional data collection

**User Rights:**

- **Right to Access:** Export user's saved services, ratings, journey progress
- **Right to Rectification:** Edit user preferences and saved data
- **Right to Erasure:** Delete user account and associated data (except audit logs)
- **Right to Portability:** Export data in JSON format
- **Right to Object:** Opt-out from personalized recommendations

## 5.4 Compliance Requirements

### 5.4.1 Data Protection Compliance

**GDPR / Local Privacy Laws:**

- Data Protection Impact Assessment (DPIA) conducted
- Privacy policy clearly communicated to users
- Consent management for optional features
- Data breach notification procedures
- Data Processing Agreement (DPA) with vendors

**Implementation:**

- User consent tracking for analytics cookies
- Opt-in for marketing communications
- Clear privacy policy and terms of service
- Regular privacy audits

### 5.4.2 Open Data Standard Compliance

**CPSV-AP (Core Public Service Vocabulary-Application Profile):**

- Service catalog structured according to CPSV-AP schema
- Interoperability with international service catalogs
- Standardized metadata for cross-border service discovery and integration

**ISO 15836 (Dublin Core):**

- Metadata elements follow Dublin Core standards for discovery, interoperability and management
- Ensures long-term preservation and discoverability

### 5.4.3 Accessibility Standard Compliance

**WCAG 2.1 AA:**

- All UI components accessible via keyboard
- Screen reader compatibility
- Sufficient color contrast (4.5:1 for normal text)
- Resizable text without loss of functionality
- Alternative text for images
- Captions for videos

**Implementation:**

- Automated accessibility testing in CI/CD pipeline
- Manual accessibility audit quarterly
- User testing with assistive technologies
- Accessibility statement published

### 5.4.4 API and Interoperability Standards

**OpenAPI 3.0:**

- Comprehensive API documentation
- Machine-readable API specification
- Interactive API explorer (Swagger UI)

**GovStack Building Block Standards:**

- Compliance with GovStack specifications
- Standardized interfaces for integration
- Reusable components

# 6. Appendices

## Appendix A: Glossary

| Term | Definition |
|---|---|
| **Service** | A government service available to citizens, businesses, or other entities |
| **Life Event** | Significant events in a person's or business's life that require government services (e.g., birth, marriage, business start) |
| **Journey** | A collection of related services grouped together for a specific life event or goal |
| **Service Bundle** | A pre-defined group of services that are commonly used together |
| **Eligibility** | The criteria that determine whether a user qualifies for a service |
| **Service Owner** | The government agency or department responsible for delivering a service |
| **Service Lifecycle** | The stages a service goes through from creation to archival |
| **CPSV-AP** | Core Public Service Vocabulary-Application Profile, an EU standard for service catalogs |
| **WCAG** | Web Content Accessibility Guidelines |
| **RBAC** | Role-Based Access Control |
| **API** | Application Programming Interface |
| **NDI** | National Digital Identity (Identity Provider) |

## Appendix B: API Reference Summary

**Service Catalog APIs:**

- GET /api/services - List all services (with filters)
- GET /api/services/{id} - Get service details
- POST /api/services - Create new service (admin)

- PUT /api/services/{id} - Update service (admin)
- DELETE /api/services/{id} - Delete service (admin)

**Search APIs:**

- GET /api/search?q={query} - Search services
- GET /api/search/suggest?q={query} - Auto-complete suggestions
- GET /api/services/popular - Get popular services
- GET /api/services/recommended - Get personalized recommendations

**Journey APIs:**

- GET /api/journeys - List all journeys
- GET /api/journeys/{id} - Get journey details
- GET /api/journeys/{id}/progress - Get user's progress in journey
- POST /api/journeys/{id}/start - Start a journey

**Analytics APIs:**

- GET /api/analytics/services/{id} - Get service analytics
- GET /api/analytics/search - Get search analytics
- GET /api/analytics/dashboard - Get dashboard data

## Appendix C: Error Codes

| Code | Message | Description |
|---|---|---|
| 200 | OK | Request successful |
| 201 | Created | Resource created successfully |
| 400 | Bad Request | Invalid request parameters |
| 401 | Unauthorized | Authentication required |
| 403 | Forbidden | Insufficient permissions |
| 404 | Not Found | Resource not found |
| 409 | Conflict | Resource already exists |
| 422 | Unprocessable Entity | Validation error |
| 429 | Too Many Requests | Rate limit exceeded |

| | | |
|---|---|---|
| 500 | Internal Server Error | Server error |
| 503 | Service Unavailable | Service temporarily unavailable |

## Appendix D: Definitions

| Particulars | Definition |
|---|---|
| Must | Critical for system operation, legal or regulatory requirement or business process requirement. |
| Should | Important for user experience, significant business value |
| Could | Nice to have, future enhancements |
| Won’t | Out of scope for current phase |

# 7. Annexure 1: Example User Scenarios for Public Service Registry (PSR)

## Scenario A: Citizen Discovering Birth Registration Services

**Step 1: Life Event Selection**

- **User:** Logs into the MyBhutan Portal → selects the **"Birth of Child"** life event.

**Step 2: Service Bundle Display**

- **PSR:** Returns the bundled services:
    - Birth Registration (Priority: Urgent)
    - Vaccination Scheduling
    - Millennium Education Policy Subscription
    - Education Enrollment
    - Bank Account Opening for Child

**Step 3: Service Details Review**

- **User:** Clicks on **"Birth Registration"**
- **PSR:** Provides service details, displayed by the MyBhutan Portal:
    - Eligibility: All citizens
    - Required Documents: Hospital birth certificate, Parent IDs
    - Processing Time: 3 working days
    - Cost: Free (information only)
    - Delivery Channel: Online + Service Center
    - Prerequisites: None
    - Next Steps: Vaccination Scheduling (recommended within 30 days)

**Step 4: Personalized Eligibility Check**

- **MyBhutan Portal:** Sends attributes from the citizen profile to PSR.
- **PSR:** Evaluates the approved eligibility rules and returns a verdict for each service.
- **MyBhutan Portal:** Displays: **"You are eligible for all services"**

**Step 5: Service Initiation**

- **User:** Clicks **"Start Application"**
- **MyBhutan Portal:** Hands the user over to the **Birth Registration System**, which provides the application form. PSR does not process the application.

**Step 6: Progress Tracking**

- **User:** Returns to the **MyBhutan Portal**
- **PSR:** Provides journey progress, based on status updates from the agency systems:
    - Birth Registration: In Progress (Step 2 of 4)
    - Vaccination Scheduling: Not Started (Recommended by Day 25)
    - Benefits Application: Available (Prerequisites met)

**Key PSR Capabilities Demonstrated:**

- Life event-based service discovery
- Service bundling and prioritization
- Comprehensive service information display
- Eligibility evaluation using citizen profile attributes sent by the MyBhutan Portal
- Hand-off to agency service delivery systems
- Multi-service progress tracking

**Success Criteria:**

- User finds all relevant services within 2 minutes
- Eligibility check completes in <2 seconds
- Service information is complete and accurate
- Smooth transition to the application system
- Progress visible across all services

## Scenario B: Business Owner Starting New Enterprise

**Persona:** Tshering Wangmo, Entrepreneur

**Context:** Starting a restaurant business; needs to complete all required registrations.

**Step 1: Business Type Selection**

- **User:** Logs in to the **MyBhutan Portal** (business account). → searches **"start restaurant business"**
- **PSR:** search to identify:
    - Business type = Restaurant
    - Action = Start new

**Step 2: Comprehensive Service Journey**

- **PSR:** Returns the **"Starting a Restaurant Business"** journey (cross-agency; owned by a Service Owner of the lead agency designated by the Super Admin).

**Sequential Steps (must be completed in order):**

- **Business Name Registration** (Ministry of Industry, Commerce and Employment): Ready to Start | 2 days | 500 BTN
- **Taxpayer Registration** (Department of Revenue and Customs): Prerequisite required (Step 1) | 3 days | Free
- **Food Safety License** (Bhutan Food and Drug Regulatory Authority): Prerequisites required (Steps 1, 2) | 10 days | 2,000 BTN
- **Business Operating License** (Ministry of Industry, Commerce and Employment): Prerequisites required (Steps 1, 2, 3) | 7 days | 3,000 BTN

**Parallel Steps (can be done any time):**

- Employee Registration (MoICE)
- Fire Safety Inspection (Fire Department)
- Environmental Clearance (if applicable)

**Estimates (calculated by PSR from the services in the journey; fees are for information only):**

- Total Time: 4–6 weeks
- Total Cost: 5,500 BTN + inspections

**Step 3: Detailed Requirements**

- **User:** Clicks on **"Business Name Registration"**
- **PSR:** Provides:
    - **Eligibility:** Bhutanese citizens or registered foreign entities
    - **Required Documents:** Citizenship ID; proposed business name (3 alternatives); business plan summary; lease agreement or property ownership proof
    - **Application Process (carried out in the agency system):** search name availability; submit application with documents; pay registration fee (through the Payment Block, not PSR); receive approval certificate
    - **Delivery Channels:** Online portal (preferred), Service center

**Step 4: Service Dependencies Visualization**

- **MyBhutan Portal.:** Shows the journey map provided by PSR:
    - **4.1:** Complete **Name Registration**
    - **4.2:** Complete **Tax Registration** (after 4.1)
    - **4.3:** Obtain **Food License** (after 4.1 and 4.2)
    - **4.4:** Obtain **Operating License** (after 4.1, 4.2 and 4.3)
    - **4.5 (Parallel):** Register **Employees**; complete **Fire Safety Inspection** (any time after starting)
    - **Current Status:** You are on Step 1 of 4 mandatory steps

**Step 5: Save and Track Journey**

- **User:** Clicks **"Start This Journey"**
- **PSR:** Starts journey tracking:
    - Saves progress, updated from agency system status

    - Monitors completion of prerequisites and unlocks next steps
    - Requests reminders and next-step alerts, which the Notification Block sends

**Step 6: Cross-Agency Data Reuse**

- **PSR:** Defines which outputs of one step are reused in later steps. The data itself moves through the MyBhutan Portal. auto-fill and agency systems; PSR holds no application data.
    - Business name from Step 1 auto-populates in Steps 2, 3, 4
    - Tax registration number from Step 2 flows to later steps
    - Citizenship ID does not need to be re-submitted

**Key PSR Capabilities Demonstrated:**

- Semantic search understanding business context
- Complex service journey orchestration
- Sequential and parallel service dependencies
- Visual journey mapping
- Cost and time estimation
- Definition of data reuse across services
- Journey progress tracking and reminders

**Success Criteria:**

- User understands the complete business setup process in <5 minutes
- All required services identified and sequenced correctly
- Clear visibility into costs, timelines and requirements
- Data reuse across services reduces application time by >60%
- User completes the journey 30% faster than without PSR

## Scenario C: Cross Agency Service Bundle for Marriage

**Persona:** Karma and Deki, Engaged Couple

**Context:** Planning to get married; need to understand all related government services. The scenario is a hypothetical service journey.

**Step 1: Life Event Selection**

Karma and Deki log in to the **MyBhutan Portal.** and search for "getting married." PSR returns the **"Marriage"** journey (cross-agency; owned by a Service Owner of the lead agency designated by the Super Admin; services from other agencies are included as read-only).

**Step 2: Before Marriage Services**

**Required:** Marriage Registration (Notary Services): 14 days' notice required; Free; Documents: ID, Proof of age (18+), Divorce decree (if applicable); both parties must be present.

**Recommended (Optional):** Pre-Marriage Counseling (Department of Social Welfare): 2 sessions over 2 weeks; Free.

**Step 3: At Marriage Services**

Marriage Certificate Issuance: generated by the agency system after the ceremony; Immediate (digital), 3 days (physical); Digital free, Physical 50 BTN.

**Step 4: After Marriage Services**

- **Immediate Actions:** Update ID Documents (5 working days, Free); Add Spouse to Health Insurance (National Health Insurance; immediate online; premium adjustment based on family size)
- **Financial Services:** Joint Bank Account Opening (if desired; links to banking service directory); Update Tax Filing Status (Department of Revenue; before next tax filing; Free)
- **Property & Legal:** Property Registration Update (if applicable); Nomination/Beneficiary Updates

**Step 5: Personalized Checklist**

The MyBhutan Portal. sends profile attributes to PSR. PSR evaluates the journey's conditional

rules (using attributes from PSR's published eligibility attribute list) and returns the extra steps that apply:

- Karma has vehicle registration → add "Update vehicle registration"
- Deki has government employment → add "Update HR records"
- Both have property → add "Update property records"

**Timeline View:** Week 1: Marriage registration · Week 2: Pre-marriage counseling (optional) · Week 4: Ceremony · Week 5: ID updates, health insurance · Week 6: Tax status, bank accounts · Month 2: Property updates

**Step 6: Smart Reminders**

PSR schedules reminders based on the journey timeline; the Notification Block sends them:

- 14 days before: "Register marriage date"
- 7 days before: "Prepare required documents"
- Marriage day: "Congratulations! Here's your post-marriage checklist"
- 1 week after: "Don't forget to update ID and health insurance"
- 1 month after: "Complete remaining services"

**Step 7: Progress Tracking**

On the MyBhutan Portal's **"My Services"** dashboard, PSR provides progress based on status updates from the agency systems:

- Marriage Journey Progress: 65% complete
- Marriage Registration (Completed) · Marriage Certificate (Received) · ID Update (Completed) · Health Insurance Update (In Progress) · Tax Status Update (Pending) · Bank Account (Not Started)

**Key PSR Capabilities Demonstrated:**

- Comprehensive life event service bundling
- Temporal sequencing (before/during/after)
- Conditional journey steps based on citizen profile attributes

- Reminder scheduling (sent by the Notification Block)
- Cross-agency service coordination under a lead agency
- Progress tracking across multiple services
- Optional vs. mandatory service distinction

**Success Criteria:**

- All relevant services identified and presented logically
- Personalized checklist generated based on user profile
- Timeline helps users plan service completion
- No services missed due to lack of awareness
- User satisfaction with journey completeness

## Scenario D: Government Service Provider Publishing new Vaccination Program

**Persona:** Dr. Sonam Tshering, Service Owner, Department of Health

**Context:** Launching a new HPV vaccination program; needs to publish the service information.

**Before you start:** The Super Admin has created the Department of Health in PSR and designated its Agency Admin. The Agency Admin has assigned the roles: Dr. Sonam (Service Owner), Dr. Kinley (Reviewer) and the Secretary of Health (Agency Head, final approver).

**Step 1: Service Creation**

Service Owner → PSR Admin Interface. → clicks "Create New Service"

PSR → shows the service creation form with template selection: Healthcare Service (selected), Education Service, Business Service, Civil Service, Custom.

**Step 2: Basic Information Entry**

Service Owner → fills in service details:

- Service Name: HPV Vaccination Program
- Service ID: generated by PSR (permanent, never reused)
- Service Code: HLTH-VAC-HPV
- Category: Health → Preventive Care → Immunization

- Service Type: G2C (Government to Citizen)
- Target Audience: Citizens (primary), Students
- Description: "Free HPV vaccination for adolescents aged 9–14"
- Service Provider: Department of Health, Immunization Program; Contact: immunization@health.gov.bt; Hotline: 1122 (toll-free)

**Step 3: Life Event Classification**

Service Owner → tries to link the life event "Child Health"; it doesn't exist yet.

Service Owner → raises a classification change request (name, parent, justification, affected services).

Super Admin → checks for an existing match, shared vs agency-specific use, hierarchy rules and metadata → approves. "Child Health" is added and the taxonomy version is updated.

The service stays in Draft while the request is open and cannot be submitted until it is decided.

**Step 4: Eligibility Definition**

Service Owner → defines eligibility rules using PSR's published eligibility attribute list:

- **Rules (evaluated by PSR):** Age 9–14 years; Bhutanese citizens and residents
- **Guidance (shown to users, not used as filters, because the attributes are not on the list):** No prior HPV vaccination; parental consent required for minors
- **Special Considerations:** Priority for girls aged 11–12; catch-up program ages 13–26 (limited availability)

**Step 5: Service Requirements**

Service Owner → specifies requirements:

- Required Documents: Child's citizenship ID or birth certificate, parent/guardian ID, vaccination card, school enrollment certificate (if applicable)
- Prerequisites: Health screening (done at vaccination site), no current illness or fever
- Prohibitions: Pregnancy, severe allergic reactions to previous vaccines, immunocompromised (consult doctor)

**Step 6: Service Delivery Details**

Service Owner → configures delivery:

- Delivery Channels: Health Centers (Primary), Schools (Mass campaigns), Mobile vaccination units (Rural), Online (Not applicable)
- Locations: all 20 Dzongkhags, 205 health centers, selected from MDMB location records; Monday–Friday 9:00 AM–4:00 PM; appointments recommended, walk-ins accepted
- Fees (information only): Vaccination, Consultation, Certificate, all free
- Processing Time: Appointment same day to 7 days; procedure 15–20 minutes; observation 15 minutes; certificate issued immediately
- Effective From: on publication

**Step 7: Service Relationships**

Service Owner → defines relationships:

- Related Services: Routine Childhood Immunization, School Health Services, Adolescent Health Program
- Prerequisites: None
- Recommended Follow-up: Second dose after 6–12 months, annual health checkup
- Delivered by: Immunization Information System (from the System Catalog)

**Step 8: Content & Resources**

Service Owner → uploads supporting materials to the media library; files are stored in DMS (DOCS) and referenced by PSR: service brochure (PDF, English & Dzongkha), video "HPV Vaccination explained" (3 minutes), infographic (benefits and safety), FAQ document (20 questions), consent form template, location map of vaccination centers, mobile vaccination schedule.

**Step 9: Approval Workflow**

PSR → checks the mandatory rules (classification complete, classification request decided, required fields filled).

Service Owner → submits for review. Status: Draft → Review; the service is locked for editing.

- Reviewer: Dr. Kinley (Program Director) checks completeness, accuracy and policy compliance → Approved

- Final Approver: Secretary of Health → final authorisation → Approved for Publication
- If rejected at either level, the service returns to Draft with reasons and is unlocked.
- No one can review or approve their own submission.

**Step 10: Publication**

Service Owner → clicks "Publish Service" (only the approved version can be published).

PSR → executes publication:

- **Immediate Actions:** service searchable in the catalog, added to Health → Immunization, linked to the "Child Health" life event (ages 9–14), API updated, change recorded in the audit trail
- **Event:** PSR publishes a service event through NDE; the MyBhutan Portal, UCS picks it up.
- **Announcements (sent by the Notification Block, not PSR):** email to health centers, SMS to opted-in parents of children aged 9–14, push to Citizen App users

**Step 11: Service Updates**

- Service Owner → updates the hotline number.
  PSR → classifies this as **minor** → publishes directly as version 1.1, keeps the change log.
- Service Owner → adds catch-up pricing and clarifies catch-up eligibility.
  PSR → classifies this as **major** (fees and eligibility changed) → sends it to the Reviewer and Agency Head → once approved, the Service Owner publishes version 2.0; the MyBhutan Portal re-checks eligibility, and users who saved the service are notified.

**Key PSR Capabilities Demonstrated:**

- Role-based service creation (Service Owner, Reviewer, Agency Head)
- Comprehensive service definition and metadata management
- Classification change requests
- Rich media managed in PSR, stored in DMS (DOCS)
- Multi-level approval workflow with review lock
- Service relationship and system mapping

- Event-based publication through NDE
- Automatic minor/major change classification and version control

**Success Criteria:**

- Service published within 2 hours of approval
- All required information captured completely
- Service discoverable through all consuming channels without per-channel setup
- Consuming systems updated through NDE events
- Real-time visibility into service performance
- Ability to update service information quickly

# 8. Process Flow

## 8.1. Overview

This section describes how PSR works end to end: how an agency is onboarded, how its services are created, approved and published, how citizens discover and use them, and how services are maintained and retired. All of these flows follow the principle set out in Sections 1.1.3 and 2.2: PSR is a service information and discovery platform. It manages the service catalog, taxonomy, eligibility rules, service relationships and journeys. It does not deliver services, process applications or payments, store application documents, verify identity or send notifications; those are handled by the agency service systems, the Payment Block, the DMS, NDI/SSO and the Notification Block.

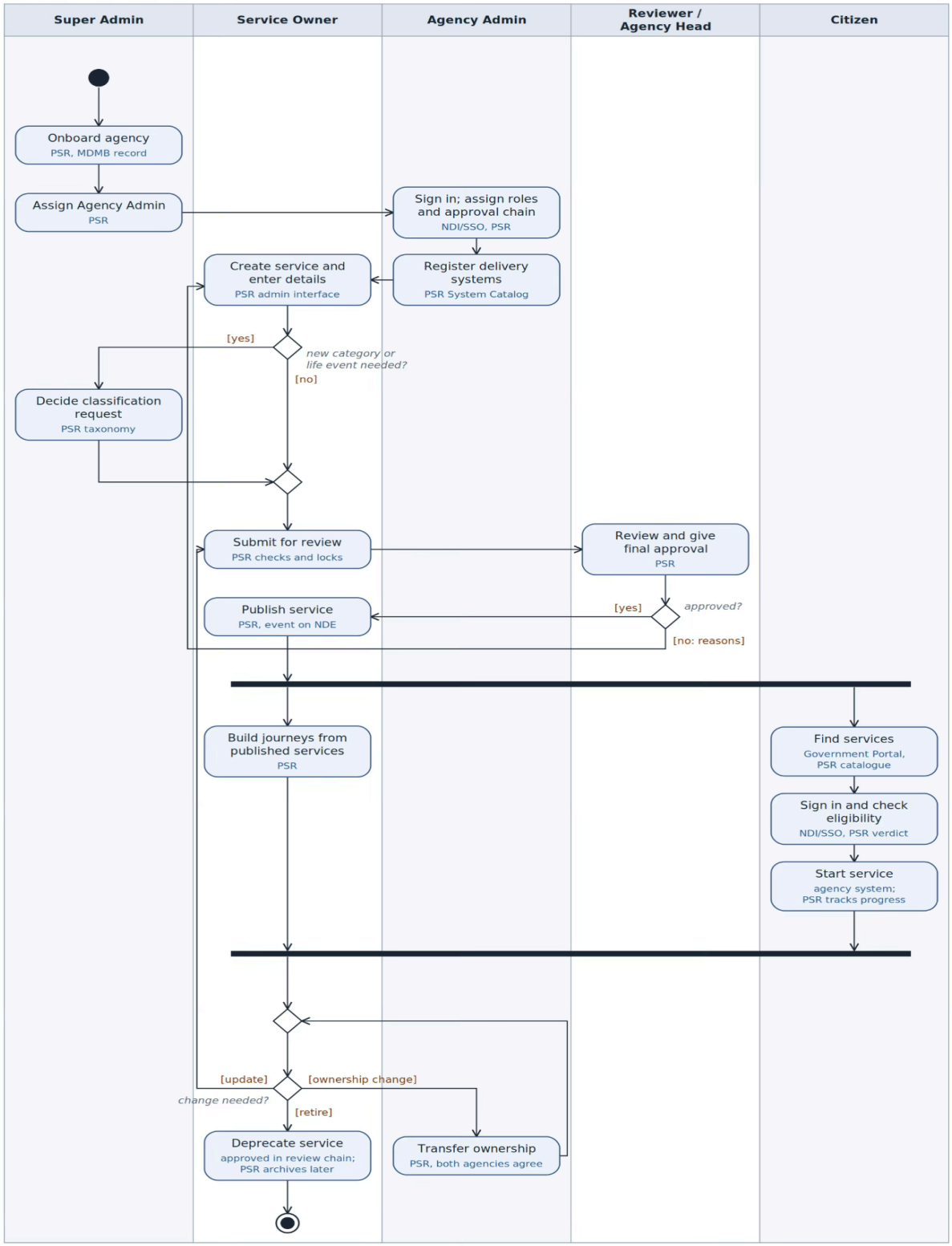



*Figure 2: PSR process flow*

## 8.2. Agency Onboarding Process Flow

A new agency must complete the following steps before any of its services become discoverable through PSR.

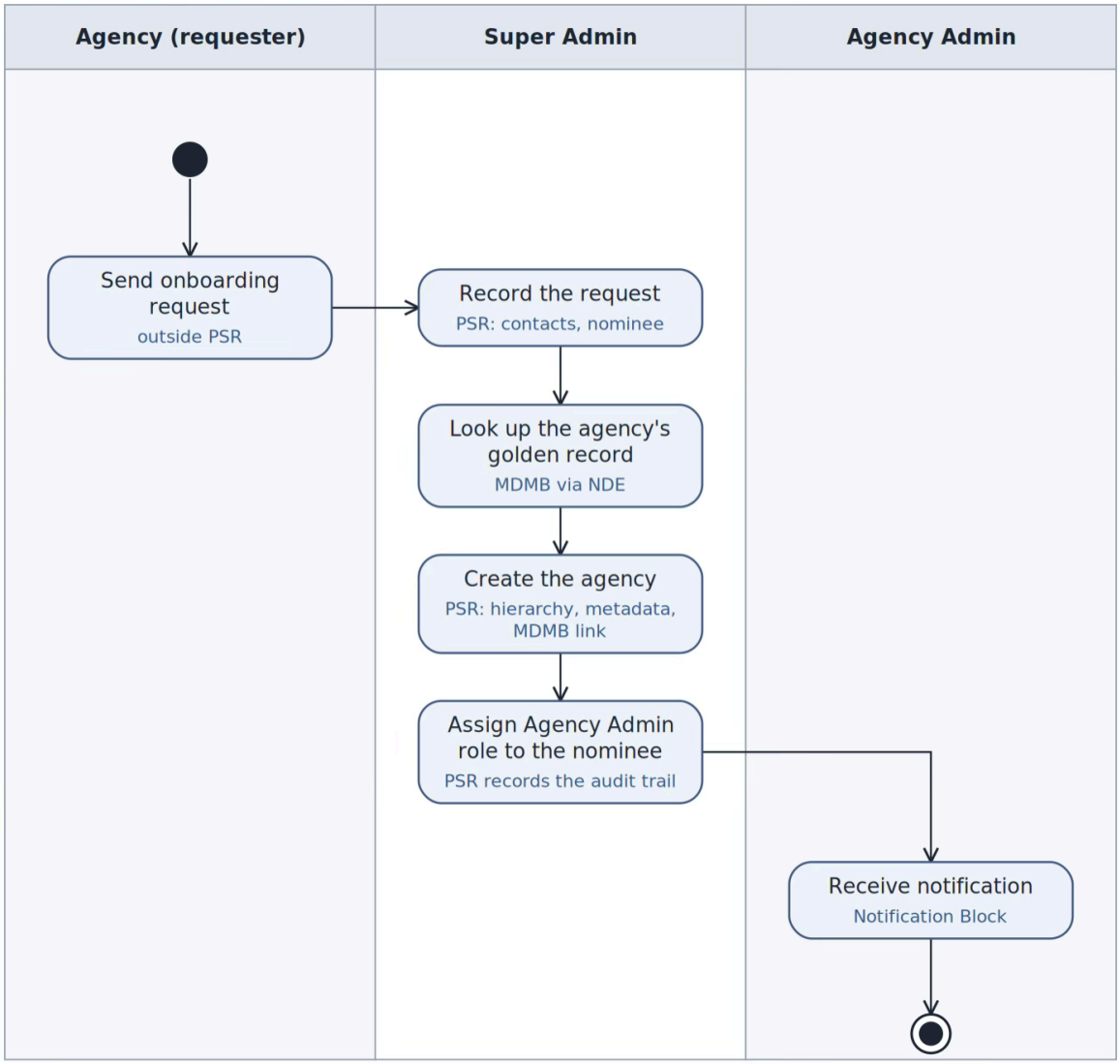



*Figure 3: Agency Onboarding and Role Setup*

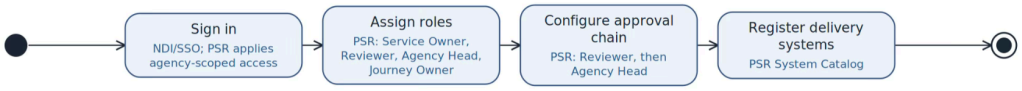



*Figure 4: Role Setup and Approval Chain*

**Step 1: Onboarding Request**
The agency submits a formal onboarding request to be onboarded to PSR, addressed to the catalog custodian. The request names the agency, its parent agency, a business contact, and the person nominated as Agency Admin.

**Step 2: Request Recorded**
The Super Admin records the request in PSR with the agency name, parent agency, MDMB agency record, business contact and nominated Agency Admin.

**Step 3: MDMB Record Check**
The Super Admin checks that the agency exists as a golden record in MDMB, accessed through NDE. If it does not exist, the agency's golden record is first registered in MDMB, where a data steward reviews it for duplicates before it is published. This keeps agency data consistent across all building blocks.

**Step 4: Agency Created**
The Super Admin creates the agency in PSR, linked to its MDMB record. The record includes its place in the government hierarchy (Ministry → Department → Division → Section) and its metadata: name, type, parent agency, contact information, website and logo.

**Step 5: Agency Admin Assigned**
The Super Admin assigns the Agency Admin role to the nominated person. PSR records the action in the audit trail and notifies the Agency Admin.

**Step 6: Agency Admin notified**

The Notification Block notifies the Agency Admin, which ends onboarding.

**Step 7: Agency Admin Sign-In**
The Agency Admin signs in with their own NDI/SSO identity. NDI/SSO confirms who they are, and PSR decides what they can do, limiting their access to their own agency.

**Step 8: Role Assignment**
The Agency Admin assigns the agency's roles: Service Owner, Reviewer, Agency Head and Journey Owner. One person may hold more than one role, but no one may review or approve their own submission.

**Step 9: Approval Chain Configuration**
The Agency Admin configures the agency's approval chain: the Reviewer as the first level and the Agency Head as final approver.

**Step 10: System Catalog Registration**
Registers the systems that deliver the agency's services in the System Catalog, so that each service can later be mapped to the system that delivers it.

## 8.3. Agency Onboarding — Roles and Responsibilities

| Step | Responsible actor | System involved |
|---|---|---|
| 1. Onboarding request | Agency | Off-platform |
| 2. Request recorded | Super Admin | PSR |
| 3. MDMB record check | Super Admin; MDMB data steward if registration is needed | MDMB via NDE |
| 4. Agency created | Super Admin | PSR |
| 5. Agency Admin assigned | Super Admin | PSR, JLB |
| 6. Receive notification | Agency Admin | Notification Block |
| 7. Sign-in | Agency Admin | NDI/SSO, PSR |
| 8. Role assignment | Agency Admin | PSR |
| 9. Approval chain configuration | Agency Admin | PSR |
| 10. System Catalog registration | Agency Admin | PSR System Catalog |

## 8.4. Service Listing Lifecycle

Every service moves through the lifecycle:

- **Draft:** the service can be edited.
- **Review:** the service is submitted and locked for editing. A rejection returns it to Draft with reasons.
- **Approved:** the Reviewer and Agency Head have approved it.
- **Published:** The Service Owner has published it and it is live.

- **Updated / Versioned:** editing a published service creates a new version. Minor changes go live directly; major changes go back to review.
- **Deprecated:** the service is being phased out, but remains visible with its status flagged.
- **Archived:** once the retention period ends, the service is archived and removed from search, and it can be restored.

While a service is published, users see one of three display statuses: **Upcoming** (before its effective date), **Active**, or **Under maintenance**.

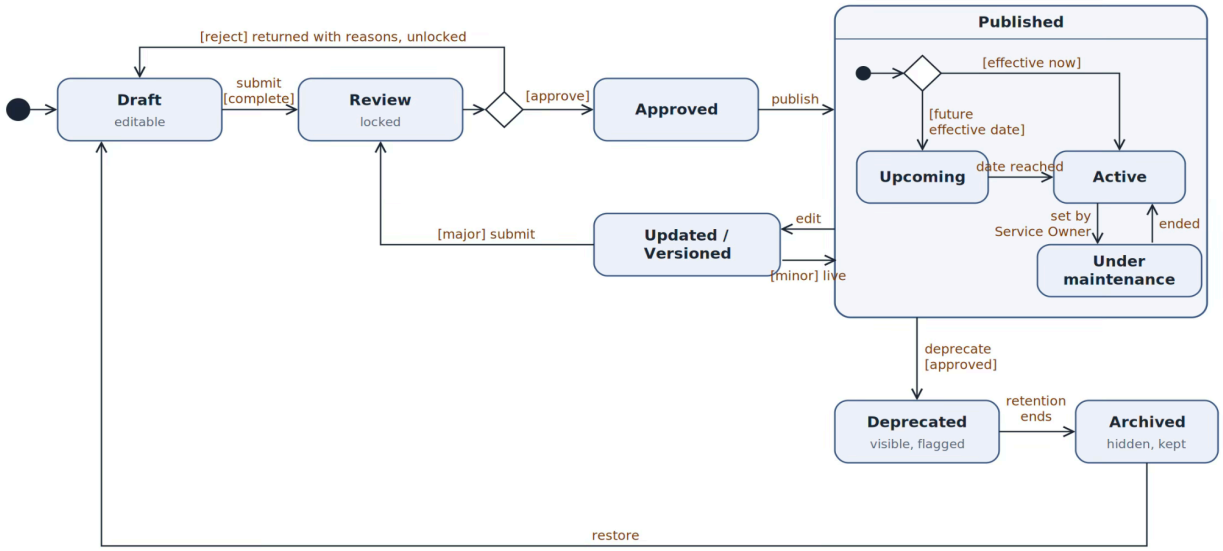



*Figure 5: Service Lifecycle*

## 8.5. Citizen Discovery and Consumption Flow

This flow describes what citizens and businesses experience once a service is published. It covers how they find a service, check whether they are eligible, and are handed over to the agency system that delivers it. PSR provides the service information, eligibility verdicts and journey progress. The application itself, documents and payment are handled by the agency system, the DMS and the Payment Block, not by PSR.

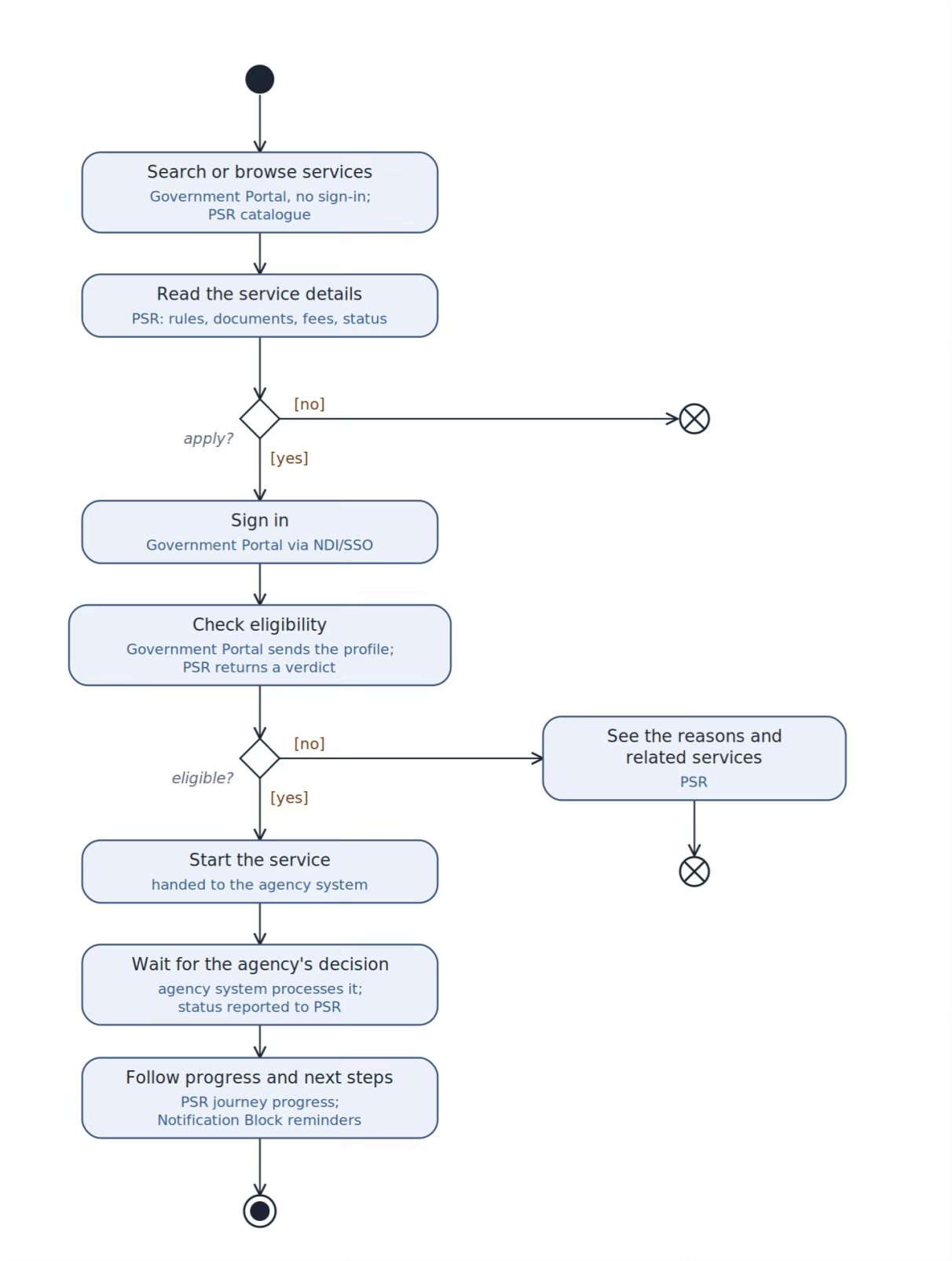



*Figure 6: Citizen Discovery, Eligibility and Hand-off*

**Step 1: Search or Browse Services**
The citizen searches or browses services on the MyBhutan Portal, using full-text or natural language search, filters, or a life event (FR-2.1, FR-2.2, FR-2.3, FR-4.2). The MyBhutan Portal reads this information from PSR.

**Step 2: View Service Details**
PSR returns the results and service details, and the MyBhutan Portal displays them: eligibility, required documents, fees (for information only), processing time, delivery channels, locations, contacts, status and interactive tools (FR-3.1, FR-3.4, FR-3.5, FR-3.6).

**Step 3: Decide to Apply**
If the citizen doesn't proceed, the flow ends. If they choose to start an application, the MyBhutan Portal asks them to sign in.

**Step 4: Sign In**
The citizen signs in to the MyBhutan Portal with NDI/SSO.

**Step 5: Eligibility Check**
The MyBhutan Portal sends attributes from the citizen's profile to PSR. PSR evaluates the approved eligibility rules and returns a verdict for each service, and the MyBhutan Portal shows only the services the citizen is eligible for, plus unconditional ones (FR-2.4). PSR uses the attributes only to evaluate the request and does not store them.

**Step 6: Not eligible**

If the citizen is not eligible, they see the reasons and related services, and the journey ends.

**Step 7: Start the Service**
If eligible, the citizen starts the service, or the next step of a journey. The MyBhutan Portal hands them over to the agency system that delivers it, which provides the application form. From this point, the application, document submission and payment are handled outside PSR.

**Step 8: Progress Tracking**
The agency system processes the application and reports its status. PSR updates the citizen's journey progress and unlocks the next steps once their prerequisites are met (FR-5.2, FR-5.4).

**Step 9: Progress and Reminders**
MyBhutan Portal shows the citizen's progress across all services in the journey on their dashboard. Reminders and next-step alerts are sent by the Notification Block, not by PSR.

**Roles and responsibilities**

| Step | Responsible actor | System involved |
|---|---|---|

| | | |
|---|---|---|
| 1. Search or browse | Citizen | MyBhutan Portal, PSR |
| 2. View service details | Citizen | MyBhutan Portal, PSR |
| 3. Decide to apply | Citizen | MyBhutan Portal |
| 4. Sign in | Citizen | MyBhutan Portal, NDI/SSO |
| 5. Eligibility check | MyBhutan Portal, PSR | MyBhutan Portal, PSR |
| 6. See reasons and related services | Citizen | MyBhutan Portal, PSR |
| 7. Start the service | Citizen | MyBhutan Portal, agency system |
| 8. Progress tracking | Agency system, PSR | Agency system, PSR(status) |
| 9. Progress, next steps and reminders | Citizen, MyBhutan Portal, Notification Block, PSR | MyBhutan Portal, Notification Block,PSR |

## 8.6 Service Management Process Flows

This section describes the processes that agencies and the Super Admin use to manage services after onboarding: classifying, creating and publishing services, building journeys, updating and rolling back services, transferring ownership, and retiring services.

### 8.6.1 Classification Change Request

When a service needs a category or life event that doesn't exist in PSR's taxonomy, the Agency Admin raises a classification change request. The request gives the proposed name, its parent, a justification and the services affected (FR-4.10). PSR records the request and holds the affected services in Draft until it is decided. The Super Admin checks for an existing match, decides whether the item should be shared or agency-specific, and checks the hierarchy rules and metadata (FR-4.1, FR-4.2). Then the Super Admin approves or rejects the request. If approved, the item is added and the taxonomy version updated (FR-4.7). If rejected, the reasons and a suggested alternative are recorded. Either way, the affected services are released, the decision is audited, and the agency is notified through the Notification Block.

| Activity | Responsible actor | System involved |
|---|---|---|
| Raise classification change request | Agency Admin | PSR |
| Record request and hold affected services | PSR | PSR |
| Review and decide request | Super Admin | PSR |
| Update taxonomy version (if approved) | PSR | PSR |
| Release affected services and audit | PSR | PSR, Journaling Block |
| Notify agency of decision | Notification Block | Notification Block |

### 8.6.2 Create, Approve and Publish a Service

The Service Owner creates a service from a template, and PSR generates a permanent service ID (FR-1.3, FR-1.8). The Service Owner enters the service details: classification, eligibility rules, requirements, delivery channels, locations selected from MDMB records, fees (for information only) and relationships (FR-2.3). They also upload media, which is stored in DMS (DOCS) (FR-3.3). Before submission, PSR checks the mandatory rules. Once submitted, the service moves to Review and is locked for editing (FR-1.4). The Reviewer checks completeness, accuracy and policy compliance, and the Agency Head gives final authorisation (FR-1.6, FR-4.6). A rejection at either level returns the service to Draft with reasons. Once approved, the Service Owner publishes it (FR-1.10). PSR makes it searchable, records the audit trail and publishes an event through NDE. The MyBhutan Portal, UCS and the Notification Block send any announcements.

| Activity | Responsible actor | System involved |
|---|---|---|
| Create service and enter details | Service Owner | PSR |
| Generate permanent service ID | PSR | PSR |
| Store media files | DMS | DMS (DOCS) |
| Validate mandatory rules; lock for review | PSR | PSR |
| Review service | Reviewer | PSR |
| Final authorisation | Agency Head | PSR |

| | | |
|---|---|---|
| Publish approved service | Service Owner | PSR |
| Publish event; refresh consumers | PSR, consuming channels | NDE, MyBhutan Portal, UCS |
| Send announcements | Notification Block | Notification Block |

### 8.6.3 Define and Publish a Journey

A journey group published services around a life event so citizens can follow them as one guided path (FR-5.1). The Journey Owner, a Service Owner of the owning agency, starts from a life event template. For a cross-agency journey, the Super Admin first designates the lead agency, whose Service Owner owns the journey. The Journey Owner adds published services and arranges them in sequence, with prerequisites, parallel branches and conditional paths (FR-5.2 to FR-5.5). PSR allows only published services and keeps other agencies' services read-only. It calculates the total time and cost, and participating agencies are notified that their services were included. The journey then goes through the Reviewer and Agency Head. A rejection returns it to the Journey Owner with reasons. Once approved, the Journey Owner publishes it, and PSR publishes an event through NDE.

| Activity | Responsible actor | System involved |
|---|---|---|
| Select life event template | Journey Owner (Service Owner) | PSR |
| Designate lead agency (cross-agency journeys) | Super Admin | PSR |
| Add services; set sequence, parallel and conditional paths | Journey Owner | PSR |
| Validate services; calculate total time and cost | PSR | PSR |
| Receive inclusion notice | Participating agency | Notification Block |
| Review journey | Reviewer | PSR |
| Final authorisation | Agency Head | PSR |
| Publish journey | Journey Owner | PSR, NDE |

### 8.6.4 Update a Published Service

When the Service Owner edits a published service, PSR creates a new version as a draft instead of changing the live one (FR-1.4, FR-1.5). PSR then classifies the change by the fields edited. Minor changes, such as wording, translations, contacts, media or hours, go live directly. Major changes, to eligibility, fees, required documents, prerequisites, prohibitions, service type, category, owner or relationships, go through the Reviewer and Agency Head (FR-1.6), and a rejection returns the version to the Service Owner with reasons. Once the version goes live, PSR updates the change log and audit trail and publishes an event through NDE. Consumers refresh their data, the MyBhutan Portal re-checks eligibility if the rules changed, and the Notification Block notifies users who saved the service.

| Activity | Responsible actor | System involved |
|---|---|---|
| Edit published service | Service Owner | PSR |
| Create new version; classify change | PSR | PSR |
| Review major change | Reviewer | PSR |
| Final authorisation of major change | Agency Head | PSR |
| Publish approved major version | Service Owner | PSR |
| Go live; update change log and audit trail | PSR | PSR, Journaling Block |
| Refresh consumers; re-check eligibility | Consuming channels | NDE, MyBhutan Portal, UCS |
| Notify users who saved the service | Notification Block | Notification Block |

### 8.6.5 Roll Back to an Earlier Version

If a published change turns out to be wrong, the Service Owner can roll the service back to an earlier version (FR-1.5). Before anything changes, PSR shows the impact: which services and journeys depend on the service. The Service Owner then decides whether to proceed. Rolling back a major version needs approval from the Reviewer and Agency Head; minor versions roll back directly. PSR restores the selected version, records who rolled it back and why, and publishes an event through NDE so consumers refresh their data.

| Activity | Responsible actor | System involved |
|---|---|---|

| | | |
|---|---|---|
| Select earlier version | Service Owner | PSR |
| Show impact on dependent services and journeys | PSR | PSR |
| Decide whether to proceed | Service Owner | PSR |
| Approve rollback (major versions) | Reviewer, Agency Head | PSR |
| Restore version; record who and why | PSR | PSR, Journaling Block |
| Publish event; refresh consumers | PSR, consuming channels | NDE, MyBhutan Portal, UCS |

### 8.6.6 Service Ownership Transfer

Ownership of one or many services can be transferred between agencies, departments or divisions (FR-4.6, FR-4.9). A transfer can be started by an Agency Admin directly, or prompted by a restructuring notice from MDMB through NDE. The requesting Agency Admin raises a transfer request listing the services, the target owner and a justification, and PSR records it. The current owner approves the request and the new owner accepts it; if either declines, the request is closed with reasons. Once both agree, PSR executes the transfer, archives the previous ownership and records the audit trail (FR-1.11). It also publishes an event through NDE, and the Notification Block notifies service owners, Agency Admins, users who saved the services and users with pending applications.

| Activity | Responsible actor | System involved |
|---|---|---|
| Send restructuring notice (optional trigger) | MDMB | MDMB, NDE |
| Raise transfer request | Requesting Agency Admin | PSR |
| Record request | PSR | PSR |
| Approve transfer | Current owner | PSR |
| Accept transfer | New owner | PSR |
| Execute transfer; archive previous ownership; audit | PSR | PSR, Journaling Block |

| | | |
|---|---|---|
| Publish ownership event | PSR | NDE |
| Notify affected parties | Notification Block | Notification Block |

### 8.6.7 Deprecate and Archive a Service

When a service is being phased out, the Service Owner requests deprecation and links a replacement service where one exists (FR-3.2). PSR validates the state change (FR-1.4), and the request goes through the Reviewer and Agency Head (FR-1.6). If rejected, the service stays Published and the reasons are returned. Once approved, PSR sets the service to Deprecated: it stays visible with its status flagged (FR-3.6). PSR then publishes an event through NDE, flags any journeys that include the service, and consumers show the deprecated status and the replacement. When the retention period ends, PSR archives the service automatically, removes it from search, and publishes an archival event so consumers remove it. An archived service can be restored.

| Activity | Responsible actor | System involved |
|---|---|---|
| Request deprecation; link replacement | Service Owner | PSR |
| Validate state transition | PSR | PSR |
| Review deprecation | Reviewer | PSR |
| Final authorisation | Agency Head | PSR |
| Set Deprecated; publish event; flag journeys | PSR | PSR, NDE |
| Show deprecated status and replacement | Consuming channels | MyBhutan Portal, UCS |
| Archive after retention period | PSR | PSR |
| Remove archived service | Consuming channels | NDE, MyBhutan Portal, UCS |