# Document Management System

## Solution Design Document

### Document Control

| **Version** | **Date** | **Author** | **Status** | **Comment** |
| --- | --- | --- | --- | --- |
| V1.0 | 05/06/2025 | Technical Team | Draft | Initial Solution Design Document |

## 1. Introduction

### 1.1 Purpose

This Solution Design Document outlines the technical architecture and implementation approach for the new Document Management System (DMS). This system will replace the current SharePoint site for procedure documents repository which has significant performance issues, particularly when handling the volume of approximately 45,000 documents.

### 1.2 Scope

The solution encompasses the design, development, and implementation of a robust, reliable, and scalable Document Management System capable of:

* Efficiently storing and managing large volumes of documents
* Providing effective access control and permission management
* Supporting document workflow processes
* Offering enhanced search and retrieval capabilities
* Maintaining compliance with audit requirements

### 1.3 References

* Business Specification Document (GDS BRM: GD & SSBI SOP Repository - PRJ06805)
* Current SharePoint Repository Assessment
* High-Level Design Diagram

## 2. System Overview

### 2.1 Business Context

The current SharePoint site for procedure documents repository has reached its performance limits and crashes under the load of approximately 45,000 documents. A new platform is required to resolve these performance and scalability challenges while providing enhanced functionality for document management.

### 2.2 Key Business Objectives

1. Implement a storage solution with capacity aligned with current and future requirements
2. Provide effective and efficient access management (critical for SSBI and SSTB Japan)
3. Develop workflow capabilities to address document addition, modification, archival, and recertification
4. Consider AI/Bionics options to enhance search, indexing, analysis, and insights generation

### 2.3 Delivery Approach

The solution will be delivered in three MVPs (Minimum Viable Products):

* **MVP 1**: Core document storage, access management, workflows, and basic UI
* **MVP 2**: Enhanced metadata management, advanced search, and reporting capabilities
* **MVP 3**: AI-assisted features, customization options, and external system integrations

## 3. Architectural Design

### 3.1 System Architecture Overview

The Document Management System will be implemented using a multi-layered architecture:

#### 3.1.1 User Layer

* Different user types: Uploaders, Reviewers, Approvers, Viewers, Admins, and System Admins
* Each user type will have specific permissions and access rights

#### 3.1.2 Presentation Layer

* Web Portal UI built with React.js
* Responsive design supporting various devices and screen sizes
* Key UI components:
  + Landing Page - Document browsing and access
  + Upload Form - Document submission with metadata
  + Review UI - Document review and approval workflows
  + Search UI - Advanced search functionality
  + Admin UI - System administration
  + AI Copilot UI (MVP3) - AI-assisted features

#### 3.1.3 API Gateway / API Management

* Centralized entry point for all frontend requests
* API versioning, throttling, and monitoring
* Security enforcement point

#### 3.1.4 Authentication and Authorization Layer

* SailPoint integration for user authentication
* Role-based access control
* Security token validation

#### 3.1.5 Application Services Layer

* **Document Management Service**: Storage, retrieval, version control, PDF conversion
* **Workflow Engine**: Process definition, task assignment, status tracking
* **Access Management Service**: Role-based control, permission management
* **Notification Service**: Email notifications based on workflow triggers
* **AI Services** (MVP3): Document validation, intelligent search, copilot assistance

#### 3.1.6 Business Services Layer

* Search and Indexing
* Reporting
* Audit and Compliance
* Document Lifecycle Management
* Bulk Operations
* Automated Recertification

#### 3.1.7 Data Access Layer

* Database abstraction layer
* Data access patterns and optimization

#### 3.1.8 Data Repositories

* Document Repository: For storing actual document files
* Metadata Database: For document metadata and attributes
* User and Access Database: For user profiles and access rules
* Workflow Database: For workflow definitions and states
* Logs: For system and audit logging
* Reference Data: For taxonomy and other reference information

### 3.2 Technology Stack

#### 3.2.1 Frontend Technologies

* **Framework**: React.js with TypeScript
* **UI Components**: Material UI or similar component library
* **State Management**: Redux or Context API
* **API Communication**: Axios or Fetch API

#### 3.2.2 Backend Technologies

* **API Framework**: Node.js with Express or .NET Core
* **Workflow Engine**: Camunda BPM or custom workflow solution
* **PDF Processing**: PDF.js or similar library
* **Email Service**: SMTP integration with templating engine

#### 3.2.3 Database Technologies

* **Document Storage**: Object storage (e.g., AWS S3, Azure Blob Storage)
* **Metadata and User Data**: Relational database (e.g., PostgreSQL, SQL Server)
* **Search Engine**: Elasticsearch or similar for advanced search capabilities

#### 3.2.4 Infrastructure

* **Deployment**: Containerized application deployed on Kubernetes or similar
* **CI/CD**: Automated build and deployment pipeline
* **Monitoring**: Application performance monitoring and logging

#### 3.2.5 Security

* **Authentication**: SailPoint integration, JWT tokens
* **Authorization**: Role-based access control
* **Encryption**: Data at rest and in transit encryption
* **Audit**: Comprehensive audit logging

## 4. Detailed Component Design

### 4.1 Document Management Service

#### 4.1.1 Storage and Retrieval

* Document files will be stored in a scalable object storage system
* Each document will have a unique identifier
* Versioning will track all changes to documents
* PDF conversion will standardize document format for viewing

#### 4.1.2 Metadata Management

* Structured metadata storage in relational database
* New metadata fields as specified in business requirements:
  + Relationship between SOPs and supporting procedures
  + Automated Unique Document Identifier
  + Alternate Owner
  + Confidential Flag
  + Org Unit
  + Legal Entity
  + Aging
  + Region
  + Country
* Metadata validation rules and constraints

### 4.2 Workflow Engine

#### 4.2.1 Workflow Definitions

* Document addition workflow
* Document modification workflow
* Document archival workflow
* Periodic recertification workflow

#### 4.2.2 Workflow States

* Draft
* Under Review
* Approved
* Published
* Archived
* Rejected

#### 4.2.3 Task Assignment

* Based on document metadata (owner, alternate owner)
* Role-based assignments
* Escalation paths for overdue tasks

### 4.3 Access Management Service

#### 4.3.1 User Management

* Integration with SailPoint for user identity
* User groups and roles mapping
* User profile information

#### 4.3.2 Permission Management

* Document-level permissions
* Role-based access control
* Special handling for confidential documents

#### 4.3.3 Access Audit

* Comprehensive logging of all access events
* Reporting on user access patterns

### 4.4 Notification Service

#### 4.4.1 Notification Types

* Document upload notifications
* Review request notifications
* Approval/rejection notifications
* Recertification reminders
* System alerts and reports

#### 4.4.2 Notification Delivery

* Email delivery with tracking
* In-system notifications
* Aggregated daily digests option

### 4.5 Search and Indexing

#### 4.5.1 Search Capabilities

* Metadata-based search (MVP1)
* Full-text search within documents (MVP2)
* Advanced filters and facets
* Saved searches

#### 4.5.2 Indexing Strategy

* Real-time indexing of new and updated documents
* Scheduled re-indexing for consistency
* Search result relevance scoring

### 4.6 AI Services (MVP3)

#### 4.6.1 Document Validation

* Automated metadata validation
* Content validation against templates
* Consistency checking

#### 4.6.2 Intelligent Search

* Natural language query processing
* Semantic search capabilities
* Personalized search results

#### 4.6.3 Copilot Assistance

* Document creation assistance
* Metadata recommendations
* Process guidance

## 5. Data Model

### 5.1 Document Entity

Document {

id: UUID (primary key)

uniqueDocumentIdentifier: String

title: String

version: String

status: Enum (Draft, InReview, Approved, Published, Archived)

documentType: Enum (SOP, SupportingProcedure, etc.)

filePath: String

fileType: String

fileSize: Number

createdBy: UUID (foreign key to User)

createdDate: DateTime

modifiedBy: UUID (foreign key to User)

modifiedDate: DateTime

confidentialFlag: Boolean

processTaxonomyId: UUID (foreign key to ProcessTaxonomy)

parentDocumentId: UUID (foreign key to Document) // For supporting procedures

}

### 5.2 Document Metadata

DocumentMetadata {

id: UUID (primary key)

documentId: UUID (foreign key to Document)

owner: UUID (foreign key to User)

alternateOwner: UUID (foreign key to User)

orgUnit: String

legalEntity: String

region: String

country: String

businessUnit: String

functionalLeader: UUID (foreign key to User)

lastRecertificationDate: DateTime

nextRecertificationDate: DateTime

aging: Number // Days since last update

}

### 5.3 User Entity

User {

id: UUID (primary key)

sailPointId: String

username: String

email: String

displayName: String

department: String

isActive: Boolean

lastLoginDate: DateTime

}

### 5.4 Group Entity

Group {

id: UUID (primary key)

name: String

description: String

sailPointGroupId: String

}

### 5.5 User-Group Association

UserGroup {

userId: UUID (foreign key to User)

groupId: UUID (foreign key to Group)

primary key: (userId, groupId)

}

### 5.6 Permission Entity

Permission {

id: UUID (primary key)

name: String

description: String

resourceType: Enum (Document, Feature, etc.)

action: Enum (Read, Write, Delete, Approve, etc.)

}

### 5.7 Group-Permission Association

GroupPermission {

groupId: UUID (foreign key to Group)

permissionId: UUID (foreign key to Permission)

primary key: (groupId, permissionId)

}

### 5.8 Document Version

DocumentVersion {

id: UUID (primary key)

documentId: UUID (foreign key to Document)

versionNumber: String

filePath: String

changeDescription: String

createdBy: UUID (foreign key to User)

createdDate: DateTime

status: Enum (Draft, Published, Archived)

}

### 5.9 Workflow Instance

WorkflowInstance {

id: UUID (primary key)

workflowType: Enum (Addition, Modification, Archival, Recertification)

status: String

currentStep: String

documentId: UUID (foreign key to Document)

initiatedBy: UUID (foreign key to User)

initiatedDate: DateTime

completedDate: DateTime

dueDate: DateTime

}

### 5.10 Workflow Task

WorkflowTask {

id: UUID (primary key)

workflowInstanceId: UUID (foreign key to WorkflowInstance)

taskType: String

status: Enum (Pending, InProgress, Completed, Rejected)

assignedTo: UUID (foreign key to User)

assignedDate: DateTime

completedDate: DateTime

dueDate: DateTime

comments: String

}

### 5.11 Process Taxonomy

ProcessTaxonomy {

id: UUID (primary key)

code: String

name: String

description: String

parentId: UUID (foreign key to ProcessTaxonomy)

level: Number

isActive: Boolean

}

### 5.12 Notification

Notification {

id: UUID (primary key)

recipientId: UUID (foreign key to User)

subject: String

message: String

type: Enum (WorkflowNotification, SystemAlert, etc.)

status: Enum (Pending, Sent, Failed)

createdDate: DateTime

sentDate: DateTime

relatedEntityType: String

relatedEntityId: UUID

isRead: Boolean

}

## 6. API Design

### 6.1 Document Management APIs

#### 6.1.1 Document Upload

POST /api/documents

Request:

- multipart/form-data with document file and metadata

Response:

- 201 Created with document ID and status

#### 6.1.2 Document Retrieval

GET /api/documents/{documentId}

Request:

- Path parameter: documentId

Response:

- 200 OK with document metadata and download URL

#### 6.1.3 Document Update

PUT /api/documents/{documentId}

Request:

- Path parameter: documentId

- multipart/form-data with updated document file and/or metadata

Response:

- 200 OK with updated document details

#### 6.1.4 Document Deletion/Archival

DELETE /api/documents/{documentId}

Request:

- Path parameter: documentId

Response:

- 204 No Content on success

#### 6.1.5 Document Search

GET /api/documents/search

Request:

- Query parameters for search criteria

Response:

- 200 OK with paginated list of matching documents

### 6.2 Workflow APIs

#### 6.2.1 Start Workflow

POST /api/workflows

Request:

- Workflow type and related document ID

Response:

- 201 Created with workflow instance ID

#### 6.2.2 Get Workflow Status

GET /api/workflows/{workflowId}

Request:

- Path parameter: workflowId

Response:

- 200 OK with workflow details and current status

#### 6.2.3 Complete Workflow Task

PUT /api/workflows/{workflowId}/tasks/{taskId}

Request:

- Path parameters: workflowId, taskId

- Task completion details (approval/rejection, comments)

Response:

- 200 OK with updated task and workflow status

### 6.3 User and Access Management APIs

#### 6.3.1 User Authentication

POST /api/auth/login

Request:

- User credentials or SailPoint token

Response:

- 200 OK with authentication token and user details

#### 6.3.2 User Management

GET /api/users

POST /api/users

PUT /api/users/{userId}

DELETE /api/users/{userId}

#### 6.3.3 Group Management

GET /api/groups

POST /api/groups

PUT /api/groups/{groupId}

DELETE /api/groups/{groupId}

#### 6.3.4 Permission Management

GET /api/permissions

POST /api/permissions

PUT /api/permissions/{permissionId}

DELETE /api/permissions/{permissionId}

### 6.4 Notification APIs

#### 6.4.1 Send Notification

POST /api/notifications

Request:

- Notification details (recipients, subject, message)

Response:

- 201 Created with notification ID

#### 6.4.2 Get User Notifications

GET /api/notifications

Request:

- Query parameters for filtering

Response:

- 200 OK with paginated list of notifications

#### 6.4.3 Mark Notification as Read

PUT /api/notifications/{notificationId}/read

Request:

- Path parameter: notificationId

Response:

- 200 OK with updated notification status

## 7. User Interface Design

### 7.1 Landing Page

* Dashboard view showing:
  + Recently accessed documents
  + Documents pending action
  + Quick search
  + Filter by taxonomy, owner, BU, region, country
* Navigation to other sections

### 7.2 Document Upload Form

* Multi-step form with:
  + File upload section
  + Metadata input section with validation
  + Document relationship mapping
  + Preview before submission

### 7.3 Review UI

* Document preview
* Side-by-side comparison with previous version
* Approval/rejection actions
* Comment section
* Change summary view

### 7.4 Search UI

* Advanced search criteria
  + Metadata fields
  + Full-text search (MVP2)
  + Date ranges
  + Document status
* Search results presentation
  + Sorting options
  + Filtering options
  + Export capability

### 7.5 Admin UI

* User and group management
* System configuration
* Audit logs
* Reporting dashboard

### 7.6 AI Copilot UI (MVP3)

* Natural language query interface
* Document suggestions
* Intelligent assistant for document creation

## 8. Security Design

### 8.1 Authentication

* Integration with SailPoint for SSO
* JWT token-based session management
* Multi-factor authentication support (if required)

### 8.2 Authorization

* Role-based access control
* Document-level permission management
* API endpoint security

### 8.3 Data Protection

* Encryption of data at rest
* Encryption of data in transit (TLS/SSL)
* Special handling for confidential documents

### 8.4 Audit and Compliance

* Comprehensive audit logging of:
  + Document access events
  + Document modifications
  + Workflow actions
  + Administrative actions
* Audit log retention policy
* Audit report generation

## 9. Integration Design

### 9.1 SailPoint Integration

* User authentication
* User group synchronization
* Access management

### 9.2 External System Integrations (MVP3)

* **Archer**: Control changes notification
* **BPM**: Process taxonomy synchronization
* **iCAMS**: Compliance management integration
* **MWD**: Workforce data for orphan detection

### 9.3 Email Integration

* SMTP service for sending notifications
* Email templates for different notification types
* Email tracking and delivery status

## 10. Performance Design

### 10.1 Scalability Approach

* Horizontal scaling for web and application tiers
* Database sharding or partitioning for metadata storage
* Content distribution network for document delivery

### 10.2 Performance Targets

* Document upload: < 5 seconds for files up to 10MB
* Document retrieval: < 3 seconds
* Search results: < 2 seconds
* UI responsiveness: < 1 second for most operations

### 10.3 Caching Strategy

* Browser caching for static resources
* Application-level caching for reference data
* Database query caching

## 11. Deployment Design

### 11.1 Environment Strategy

* Development environment
* Testing/QA environment
* UAT environment
* Production environment

### 11.2 Deployment Approach

* Containerized application components
* Orchestration with Kubernetes or similar
* Infrastructure as Code for environment provisioning

### 11.3 Backup and Recovery

* Regular database backups
* Document storage replication
* Disaster recovery procedures

## 12. Testing Strategy

### 12.1 Testing Approaches

* Unit testing for individual components
* Integration testing for API interfaces
* Performance testing for scalability validation
* Security testing for vulnerability assessment
* User acceptance testing for business validation

### 12.2 Test Data Management

* Synthetic test data generation
* Production data sanitization for testing
* Test data refresh procedures

## 13. Implementation Plan

### 13.1 MVP1 Implementation

1. Core document storage and retrieval system
2. Basic metadata management
3. Essential user interfaces
4. Fundamental workflow capabilities
5. User and access management integration

### 13.2 MVP2 Implementation

1. Enhanced metadata management
2. Bulk operations capabilities
3. Advanced search features
4. Reporting functionality
5. Automated archival and orphan reporting

### 13.3 MVP3 Implementation

1. AI-assisted document validation
2. Intelligent search capabilities
3. Custom home page features
4. External system integrations
5. Copilot functionality

## 14. Migration Strategy

### 14.1 Data Migration Approach

1. **Assessment**: Analyze existing SharePoint repository structure and content
2. **Mapping**: Define mapping between old and new metadata structures
3. **Cleaning**: Identify and resolve data quality issues
4. **Migration**: Batch migration process with validation
5. **Verification**: Ensure all documents and metadata are correctly transferred

### 14.2 Migration Tools

* Custom migration scripts
* ETL processes for metadata transformation
* Validation tools for data integrity checks

### 14.3 Cutover Strategy

* Phased migration approach
* Read-only period for SharePoint during final migration
* Post-migration validation period

## 15. Operational Considerations

### 15.1 Monitoring and Alerting

* Application performance monitoring
* Error and exception tracking
* Resource utilization monitoring
* Automated alerting for critical issues

### 15.2 Maintenance Procedures

* Regular database maintenance
* Document storage optimization
* Patch and update management
* Performance tuning

### 15.3 Support Model

* Tiered support structure
* Issue tracking and management
* Knowledge base for common issues
* User training materials

## 16. Risks and Mitigation

### 16.1 Technical Risks

1. **Performance under full load**: Conduct thorough performance testing with realistic data volumes
2. **Data migration complexity**: Detailed migration planning and pilot migrations
3. **Integration challenges**: Early proof-of-concept for key integrations

### 16.2 Business Risks

1. **User adoption**: Comprehensive training and intuitive UI design
2. **Business continuity during transition**: Phased approach and fallback options
3. **Regulatory compliance**: Early involvement of compliance stakeholders

## 17. Conclusion

This Solution Design Document provides a comprehensive technical blueprint for implementing the new Document Management System. The proposed architecture addresses the key business requirements while providing a scalable and future-proof solution. The phased MVP approach allows for incremental delivery of value while managing implementation risks.

## Appendices

### Appendix A: Glossary of Terms

* **SOP**: Standard Operating Procedure
* **BU**: Business Unit
* **MVP**: Minimum Viable Product
* **SSBI**: Self-Service Business Intelligence
* **SSTB**: Standard Settlement Terms and Billing

### Appendix B: Reference Architecture Diagram

[High-Level Architecture Diagram included in the original documentation]

### Appendix C: Detailed Workflow Diagrams

[To be developed during detailed design phase]