mSpace System Documentation

This document provides a comprehensive overview of the mSpace system, inspired by DSpace but implemented using Gin, GORM, and MinIO. It covers system design, metadata management, workflows, API endpoints, file storage, access control, and troubleshooting guidelines.

# Table of Contents

1. 1. Online Access to Your Digital Assets
2. 2. Metadata Management
3. 3. Licensing
4. 4. Persistent URLs & Identifiers
5. 5. Getting Content into mSpace
6. 6. Getting Content out of mSpace
7. 7. User Management
8. 8. Access Control
9. 9. Usage Metrics
10. 10. Digital Preservation
11. 11. System Design
12. 12. Common Errors & Troubleshooting

# 1. Online Access to Your Digital Assets

mSpace provides web-based access to digital assets stored in MinIO. Users can navigate through Communities, Collections, and Items, with support for presigned URLs to securely access files.

Supported file types include PDF, DOCX, CSV, TXT, and images (PNG, JPEG).

Presigned URLs allow time-limited secure access for download and upload.

Browser compatibility: Chrome, Firefox, Safari, Edge (latest versions).

# 2. Metadata Management

Metadata in mSpace follows a flexible JSON schema, supporting Dublin Core-like fields.

Authority control ensures consistency in metadata fields such as author, subject, and date.

Choice management allows administrators to define controlled vocabularies.

# 3. Licensing

Each Community and Collection can have its own license policy.

Submitters must grant a license when uploading items to mSpace.

Creative Commons licenses are supported for published items.

# 4. Persistent URLs & Identifiers

mSpace uses UUIDs as primary identifiers for Communities, Collections, and Items.

Bitstreams (files) are referenced by unique object keys in MinIO.

Future support planned for Handle or DOI integration.

# 5. Getting Content into mSpace

Users can manually submit items through the web UI or via REST API.

Submission workflow: User uploads item → Reviewer validates metadata → Admin approves/rejects → Item is published.

CLI tools for bulk import are planned.

Externally hosted files can be registered by storing metadata with a remote file URL.

API-driven imports allow automation of large-scale content ingestion.

# 6. Getting Content out of mSpace

mSpace exposes a REST API to retrieve metadata, items, and collections.

Example: GET /api/items/{id} returns item metadata and file references.

Future roadmap includes OAI-PMH export for interoperability.

CLI export tools will support batch downloads.

# 7. User Management

Users (E-Persons) are registered with unique accounts stored in the database.

Authentication uses JWT tokens.

Future plans include subscription features (alerts on new items).

Groups allow role-based management of permissions.

# 8. Access Control

Authentication is required for item submission and administrative actions.

Authorization follows role-based access control (RBAC).

Public items are accessible via presigned URLs.

# 9. Usage Metrics

mSpace records usage statistics at the Item, Collection, and Community levels.

System statistics include storage usage, number of users, and API activity.

Integration with Prometheus and Grafana is possible for advanced monitoring.

# 10. Digital Preservation

Checksum validation ensures file integrity.

The system periodically re-checks files stored in MinIO.

Corrupted files can be re-uploaded and re-validated.

# 11. System Design

The data model follows a hierarchy: Community → Collection → Item → Metadata.

Foreign key constraints ensure items belong to valid collections and communities.

MinIO provides scalable object storage with presigned URL support.

Database migrations are managed via GORM AutoMigrate, with careful ordering of entities.

# 12. Common Errors & Troubleshooting

Foreign Key Violations: Ensure referenced Community/Collection exists before creating items.

MinIO Upload Errors: Check bucket policy and network connectivity.

Presigned URL Expiration: Ensure the client downloads the file before the URL expires.

CORS Errors: Configure allowed origins in MinIO and Gin middleware.

Migration Conflicts: Apply migrations in order (Community → Collection → Item → Metadata).