End-to-end digital preservation system handling ingestion, storage, access, and integrity verification for decades-long retention.

What I delivered:
-built the complete pipeline end to end
-involved in design discussions
-advocated to technical and non-technical
stakeholders, community outreach
-collaborated other institutions and open source
developers
-dev, test and production environments

Requirements:
-user needs to be able to create AIPs
-user needs to be able to generate metadata
-user needs to be able to check preserved content
-content needs many, regular back ups
-sensitive documents must be safe to access
-budget contraints -- design around open source and existing system
-multiple long term storage locations
-fixity checking and reassurance, we prevent issues like bit rot
-an easy user flow to ingest content
-automation features

INGEST CONTENT

send AIP

Designed for decades-long retention
Handles:
Media refresh
Tape migration
Bit-rot detection
Cheap
Copies on failure
Cold datastore
Handles large files reliably in big chunks, scales horizontally
Hot copies elsewhere!!!

send AIP

Metadata form

-manual user entry
-pulls some meta from
ASpace by ID

send metadata →

Archivematica

-packages content
-verifies metadata
-.7z compression

send AIP

TSM1

TSM2

-Search/discover
-user auth
-flexible metadata json w strawberry

send AIP and
metadata

updating database

fixity check
archipelago

Archipelago

-front end access
-user authorization

fetch metadata by content id →

ArchivesSpace

contains partial
metadata/records

USER SERVICE

Monitoring Service

-produces graphs
-sends emails
-fixity checking, mostly on archipelago

USER SERVICE

fixity check TSM

DP Postgres Database

-content id
-og checksum
-current checksums
-exact locations

MONITOR CONTENT

S3 Minio bucket datastore ACTING BEHIND
DRUPAL
-content management system
-strawberry fields for metadata parsing
-AWS S3 Minio bucket for large
item storage

update database

VIEW CONTENT

-Fixity checking gives consistency guarantee
-TEST, DEV AND PROD environments for CI/CD
-Consistency was priority
-Safety in back ups, automating parts of the system
-Open source development of APIs to connect it all