Storage specification

Target release	0.1
Epic	
Document status	DRAFT
Document owner	Richard Lehane
Designer	Richard Lehane
Developers	Danny Archer
QA	

Goals

- Ensure secure, audit-able, accessible storage of digital State Archives
- Comply with OAIS Reference Model and digital preservation best practices

Background and strategic fit

The existing storage implementation for the digital State Archives is light-weight, is based on common standards such as CDL's Pairtree specification, and is flexible. It does have limitations however relating to: fine-grained control over access to Digital State Archives and logging of significant events (such as preservation actions, metadata changes, access) over time.

A new storage specification will:

- build on the existing storage implementation
- be **immutable**, meaning that no information stored in the Digital Archives is ever modified or deleted and that changes are additive rather than destructive
- support granular access to digital archives, meaning ability to control access to information within AIPs (e.g. metadata fields) by defining flexible rules for generation of DIPs from AIPs
- support logging of changes over time
- have rich context via the use of JSON LD.

Assumptions

- the Digital Archives team can define and customise its AIP storage (i.e. isn't required to adopt an off-the-shelf solution with its own fixed storage implementation)

Requirements

#	Title	User Story	Importance	Notes
1	Store detailed technical information about preservation processes	A project officer has run a particular version of siegfried, with a particular signature file, on a particular date, to produce file format identifications that are associated with the object. It isn't desirable to store this granular information in the metadata file.	Must be able to log	
2	Retain original metadata, and account for each addition or migration.	When project officer submits a SIP to the digital archive, or amends a AIP each change will be retained and each version can be obtained.	Storage must be immutable	
3	Metadata that controls access to information within AIPs	Given a complex access direction, e.g. one that applies to the metadata as well as the files within an object, be able to control within the manifest what files can be made available in a DIP and how metadata files should be changed in order to allow access in a DIP.	Access must be granular	

User interaction and design

SIP | AIP structure

SIPs have the same structure as AIPs. They differ in content only in that:

- references within SIPs to other SIPs in same consignment are relative #{1}
- AIPs have an additional log entry "ingestion" that is added on migration into repository.

```
foo/
|-- metadata.json
|-- manifest.json
|-- logs/
|---- 1492534569.json
|-- patches/
|---- access/
|------ 0.json
|---- manifest/
|------ 0.json
|---- metadata/
|------ 0.json
|------ 1.json
|-- versions/
|---- 0/
|------ file.doc
|---- 1/
|------ preview.png
|---- 2/
|------ text
```

DIP structure

DIPs omit patches and logs and apply access rules as defined in the (fully patched) manifest.json file, in relation to the present date, and context of the DIP generated.

```
foo/
|-- metadata.json
|-- manifest.json
|-- versions/
|---- 0/
|------ file.doc
|---- 1/
|------ preview.png
|---- 2/
|------ text
```

metadata.json

The metadata file includes essential contextual information describing the digital object. This includes recordkeeping metadata (e.g. creator, access directions, disposal classes) as well as metadata specific to the functioning of the digital object in its original environment (i.e. domain specific metadata such as ABN number).

This file is immutable. Changes defined are defined in patches/metadata sub-folder according to JSON patch http://jsonpatch.com/. These changes are applied during DIP generation. Rather than cumulatively apply changes, the last patch file (i.e. highest numbered) is the only one applied.

manifest.json

The manifest file supports the functioning of the digital object in the Digital State Archive environment. It does this by:

- identifying all the files that comprise the digital object and ensuring fixity of those files by storing checksums
- linking machine actionable access rules to the metadata and files that comprise the object. These rules determine what information from an AIP is included during DIP generation.
- and supporting functionality of the Digital State Archive access layer by linking to display targets (e.g. a versions/1/preview.png file); full text for indexing (e.g. a versions/2/text file).

This file is immutable. Changes defined are defined in patches/metadata sub-folder according to JSON patch http://jsonpatch.com/. These changes are applied during DIP generation. Rather than cumulatively apply changes, the last patch file (i.e. highest numbered) is the only one applied.

The specification for this file is: Manifest Specification

Versions

The versions folder contains all the various manifestations or views of the object. This includes the original form of the digital object, any transformed forms for preservation or access purposes, any preview forms e.g. a thumbnail view of an object, and any text forms e.g. transcripts or extracted text.

Logs

The logs folder contains individual log files named by the UNIX time at which the event occurred e.g. 1492534569.json.

These log files record signicant events that occurred during SIP ingestion and throughout the lifetime of the object within the Digital State Archive. Events might incluce format identification, ingestion, DIP generation, fixity calculation/verification etc.

PREMIS event terms are preferred for content of log files e.g. premis:eventIdentifier; premis:eventType (use http://id.loc.gov/vocabulary/preservation/eventType for controlled list); premis:eventDateTime; premis:eventDetailInformation; premis:eventOutcomeInformation.

Patches

(The patches folder and its subfolders are all optional and should only be included where an AIP has patch files).

The patches folder contains JSON patch files (http://jsonpatch.com/). These patches have two roles:

- allow updates to the manifest.json and metadata.json files over time while maintaining the immutability of the Digital State Archive
- control access to information within the metadata.json file during DIP generation.

The patches folder contains three sub-folders: access/ manifest/ and metadata/.

Patch files that are within manifest and metadata folders define changes to the metadata.json and manifest.json files. Patch files within these folders are labelled with incrementing integers i.e. 0.json, 1.json etc. The rule for application of these patches is: *last only* i.e. the highest most patch number is the only one applied.

Patch files within the access folder define the application of access rules to the metadata.json file during DIP generation (application of access rules to the manifest.json file is handled by the manifest file itself). Patch files within this folder is labelled with incrementing integers i.e. 0.json, 1.json etc. The rule for application of these patches is: apply the *relevant patch file* per the rules set out in the (fully patched) manifest.json file.

Storage commands

Assuming a command line tool named "repo". These are the type of commands that should be supported by the storage specification.

DIP generation

repo dipgen -level publish 9819098 // default - publish means a DIP that can be published online

repo digpen -level open 9819098 // open means a DIP that can't be published online but can be made available e.g. in reading room or by email

repo dipgen -level closed 9819098 // closed means a DIP that is closed e.g. can be provided only with permission agency

AIP generation

repo aipgen PATH

include a dry run flag that just logs changes

SIP/AIP/DIP validation

repo validate PATH/ID // checks manifest; recalculates checksums; schema validation

include flags to define level of validation

AIP patching

repo patch 9819098 PATH-to-SIP

include a dry run flag that just logs changes

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
Authenticity/ signing: Will the whole AIP be signed? Will subsets of the file be signed? Can signature files be versioned? Case example - if the log file is edited to contain JSON patch edits, which signature (if any) will be used to verify the future AIP? Is the manifest going to be versioned (as JAR is capable of)?	

Not Doing

Examples of related specifications

JAR manifest: http://docs.oracle.com/javase/7/docs/technotes/guides/jar/jar.html#Manifest_Specification

A JAR file is essentially a zip file that contains an optional META-INF directory.

manifest.json
view
--> thumb, display target, fields
index
---> text, fields

log.json
--> timestamps, users, ID

metadata.json

Zero Install Manifest Specification: http://0install.net/manifest-spec.html

0install generates a 'manifest' file. The manifest lists every file, directory and symlink in the tree, and gives the digest of each file's content. Here is a sample manifest file for a tree containing two files (**README** and **src/main.c**) and using the **sha1** algorithm:

F 0a4d55a8d778e5022fab701977c5d840bbc486d0 1132502750 11 README
D 1132502769 /src
F 83832457b29a423c8e6daf05c6dbcba17d0514dd 1132502769 17 main.c

'D' nodes correspond to directories, and their line format is:

```
"D", space, [mtime, space,] full path name, newline
```

F' and 'X' nodes correspond to files and executable files, respectively, and their line formats are:

```
"F", space, hash, space, mtime, space, size, space, file name, newline
"X", space, hash, space, mtime, space, size, space, file name, newline
```

References

European Archival Records and Knowledge Preservation. (various). Project Deliverables - E-Ark Project. Retrieved September 8, 2016, from http://www.eark-project.com/resources/project-deliverables

Australian research data provenance interest group minutes, from https://drive.google.com/folderview?id=0B3urY1fwzAEcflAwQlI4MzM2R0RpeVhzYXV3SXNObnZlZ0VpYmZvaFA2MFFsd0lqbzVCbnc&usp=drive_web

Standard Vocabularies and Authority Files

Used for a 2015 application profile (may be out of scope here).

Abbreviation or label	**Full title and Web access address**
Apple Uniform Type Identifier	https://developer.apple.com/library/ios/documentation/Miscellaneous/Reference/UTIRef/Articles/System-Declared
Archivematica	https://www.archivematica.org/wiki/Main_Page
CRS	Commonwealth Record Series System http://recordsearch.naa.gov.au/manual/index.htm
DPTR	Digital Preservation Technical Registry http://ndha-wiki.natlib.govt.nz/current-initiatives/technical-registry/
FITS	Flexible Image Transport System http://fits.gsfc.nasa.gov/
ISAAR (CPF)	International Standard Archival Authority Record for Corporate Bodies, Persons and Families, 2nd Edition http://w val-authority-record-for-corporate-bodies-persons-and-families-2nd-edition.html
ISO 15511 : 2011	Information and documentation -- International standard identifier for libraries and related organizations (ISIL) http

ISO 3166	ISO Country Codes http://www.iso.org/iso/country_codes.htm
ISO 8601 : 1988 (E)	Data elements and interchange formats - Information interchange - Representation of dates and times (W3C-DTF
LOCeventTypes	http://id.loc.gov/vocabulary/preservation/eventType <LOCeventTypes>: ingestion, fixity check, message digest ca quarantine, unquarantine, unpacking (proposed to working group), name cleanup (proposed to working group), vi (i.e. creation of normalized versions of ingested objects), registration (assigning an accession number to a set of signature validation, capture, compression, decryption, deletion and replication.
LOCpreservationLevelRole	http://id.loc.gov/vocabulary/preservation/preservationLevelRole
MIMETYPE	Internet Assigned Numbers Authority (IANA) Media Type http://www.iana.org/assignments/media-types/media-typ
NDIIP ECHO DEPository	The National Digital Information Infrastructure and Preservation Program http://www.ndiipp.illinois.edu/ - mainly fo METADATA_TRANSFORMATION = the transformation of one metadata format into another METADATA_CREATION = the creation of a new metadata record METADATA_MODIFICATION = the modification of a metadata record that does not change the format METADATA_DELETION = the deletion of a metadata record
NLA Mediapedia	National Library of Australia Mediapedia: Physical Format Carrier Resource http://mediapedia.nla.gov.au/home.pl
PREMIS	Preservation Metadata: Implementation Strategies http://www.loc.gov/standards/premis/
PRONOM	UK National Archives format and software registry http://apps.nationalarchives.gov.uk/PRONOM/
Term and Code List for Resource Description and Access (RDA) Carrier Types	http://www.loc.gov/standards/valuelist/rdacarrier.html
UDFR	Unified Digital Format Registry http://www.udfr.org/