Architecture

Suggested Architecture, a work in progres

Tasks to solve

- Ingest into the archive
- Automatic QA of a batch (valid data and metadata)
- Manuel QA of a batch
- Creation of preservation metadata
- Creation of dissemination copy

Principles

- Early Ingest
- QA, preservation- and dissemination tasks take place in the archive
- Autonomous components, rather than a fancy workflow
- Automatic QA should be runnable by Ninestars

Elements of the Solution

Existing Elements

- Bit repository
- DOMS
- Hadoop platform
- DOMS GUI

New Elements

- Presentationinterface for pages (Prototype from Toke)
- Early Ingest-platform
- Platform for validation of metadata in DOMS
- Extraction of Files/metadata for manual QA

Autonomous Components

Overall Description

An autonomous component is watching the system, and discovers work to be done

Suggested solution:

- The Component performs a query to discover tasks to do
- The query is to a Summa index of the DOMS
- If there are tasks to do, a task is started
- The component is responsible for making sure that it does not start a task it has already started. This can safely be done in memory.
- The result of the task is written back to the Batch object in DOMS as metadata
- Information about the execution of the taks is written back to DOMS as preservation metadata. Probably as a Premis Event.
- Event data and the like is harvested frequently by Summa for indexing, and thus form the input for other autonomous components

Ingest into the Archive

Overall description:

Given a batch, put all jp2-files in the bitrepository, and all xml-files (including ALTO) into DOMS

Suggested solution:

- Hotfolder is watched for new batches
- All files with the extension *.jp2 is put into the bit repository (.md5-files are used for fixitychecks). The solution should be configurable to support other extensions
- All other files are put into DOMS
- The directory structure is preserved in the bitrepository File ID
- The directory structure is preserved in DOMS by having one object per directory, one datastream per file. The directory names are preserved in the object lables, the filenames in the datastream labels

Surveilance interface

Overall description:

A graphical web interface to show the current state for each batch

Suggested solution:

- Each batch is represented as a row in a table
- Each column represents an autonomous component, and the cell shows if this component have completed it's work on the batch
- Each cell in the table is a simple lamp, that shows if the taks have been completed. Pressing the lamp will fetch more information.
- The web interface integrates with MF PAK to get states from batches before and after digitisation
- This interface could possibly be used to approve or reject a batch

Bitarkivet og bånd

Overordnet beskrivelse:

Filer skal gemmes i bitarkivet på ten ben, et nearline og et offline. Begge de involverede ben vil have en anseelig mændge cache, men vil være tape-backed.

Foreslået løsning:

- Filerne lægges på de to ben ved modtagelse,
- Filer i cache vil blive rullet på bånd når et bestemt, men pt. udefineret (tidsbestemt, eller eksplicit ved godkendelse af batch), signal sendes.
- Filer fra batches der fejler validering vil blive slettet igen.
- Vi antager at vi kan validere og afvise et batch før det bliver nødvendigt at rulle det på bånd, men i værste fald er der spildt noget plads på båndet
- Nøglerne der giver adgang til at slette fra bitmagasinet skal eksplicit skaffes hvis der skal fjernes et ekstra batch. Som udgangspunkt kan nøglerne også slette allerede godkendte batches.

Opgaver på data (jp2-filer)

Overordnet beskrivelse:

Alle operationer på data køres som hadoop-jobs på filer i bitmagasinet. Herunder karakterisering og generering af formidlingskopier.

Foreslået løsning:

- jpylizer køres i map-skridt af map/reduce-job. Resultatet lægges i DOMS (som datastream) i reduce-skridt.
- Udtræk af histogram køres i map-skridt af map/reduce-job. Resultatet lægges i DOMS i reduce-skridt.
- Generering af formidlingskopi køres i map-skridt af map/reduce-job. Filen skrives direkte til formidlingsstorage. Eventuelle fejl skrives tilbage til DOMS i reduce-skridt.
- Der laves ingen validering i map/reduce-job. Dette foretages som efterbehandling af metadata.

Validering af metadata (i DOMS)

Overordnet beskrivelse:

Alle jobs der validerer metadata opfattes som jobs på et helt batch. Metadatavalidering kan være lokalt for én xml-fil eller kræve kendskab til sammenhæng mellem flere xml-filer i et batch - f.eks. validering af samme batchnummer i alt metadata eller validering af fortløbende sidenumre

Foreslået løsning:

- Validering implementeres som gennemløb af en træstruktur (svarende til filstruktur-hierarki), med mulighed for validering i hver knude
- Validering af en enkelt XML-fil kan foretages med XML-schema og schematron
- Træet kan gennemløbes med et filsystem eller DOMS som nederste niveau

- Resultatet af en validering skrives tilbage til DOMS
- En fejlet validering skal medføre en notifikation til Supervisor
- Hellere små autonome skridt end ét stort workflow

Manuel QA

Overordnet beskrivelse:

Der skal foretages manuel QA på filer og metadata udvalgt efter statistisk princip. Derudover skal der evt. foretages manuel QA på filer vi kan identificere som mistænkelige (f.eks. et helt mørkt batch). Manuel QA kræver adgang til et system til at inspicere jp2-filer og metadata. Vi fravælger i første omgang at lave et egentligt workflowstyringsprogram.

Foreslået løsning:

- Der etableres en fremvisningsløsning til avissider i bitmagasinet. Toke har lavet en første udgave.
- Der etableres mulighed for direkte links til objekter i DOMS GUI.
- Der udtrækkes hvilke sider der skal checkes udfra den statistiske visning.
- Ud fra dette udtræk laves et regneark (csv-fil) med link til fremvisning og DOMS GUI.
- Til dette ark tilføjes mistænkelige objekter med tilsvarende links og en begrundelse for hvorfor de er mistænkelige
- Vi laver ingen support af hvordan regnearkene benyttes til at godkende batches

Fejlede batches

Overordnet beskrivelse:

Batches fejles eller godkendes som et hele. Man kan ikke afvise dele af et batch, og leverandøren vil aflevere et nyt batch.

Foreslået løsning

- Hvis et batch fejler, skal hele batches slettes fra Bitarkivet
- Slette fra bitmagasinet involverer anskaffelse af slette nøglen
- Hvis et batch fejler, skal hele batches slettes fra DOMS
- Doms kan slette uden en bestemt slette nøgle, da alt bliver bevaret alligevel, og kan genskabes.
- Et genafleveret batch skal behandles på samme måde som et nyt batch