Integrating Islandora and Archivematica

Mark Jordan

Islandora Camp 2012

Charlottetown, August 2, 2012

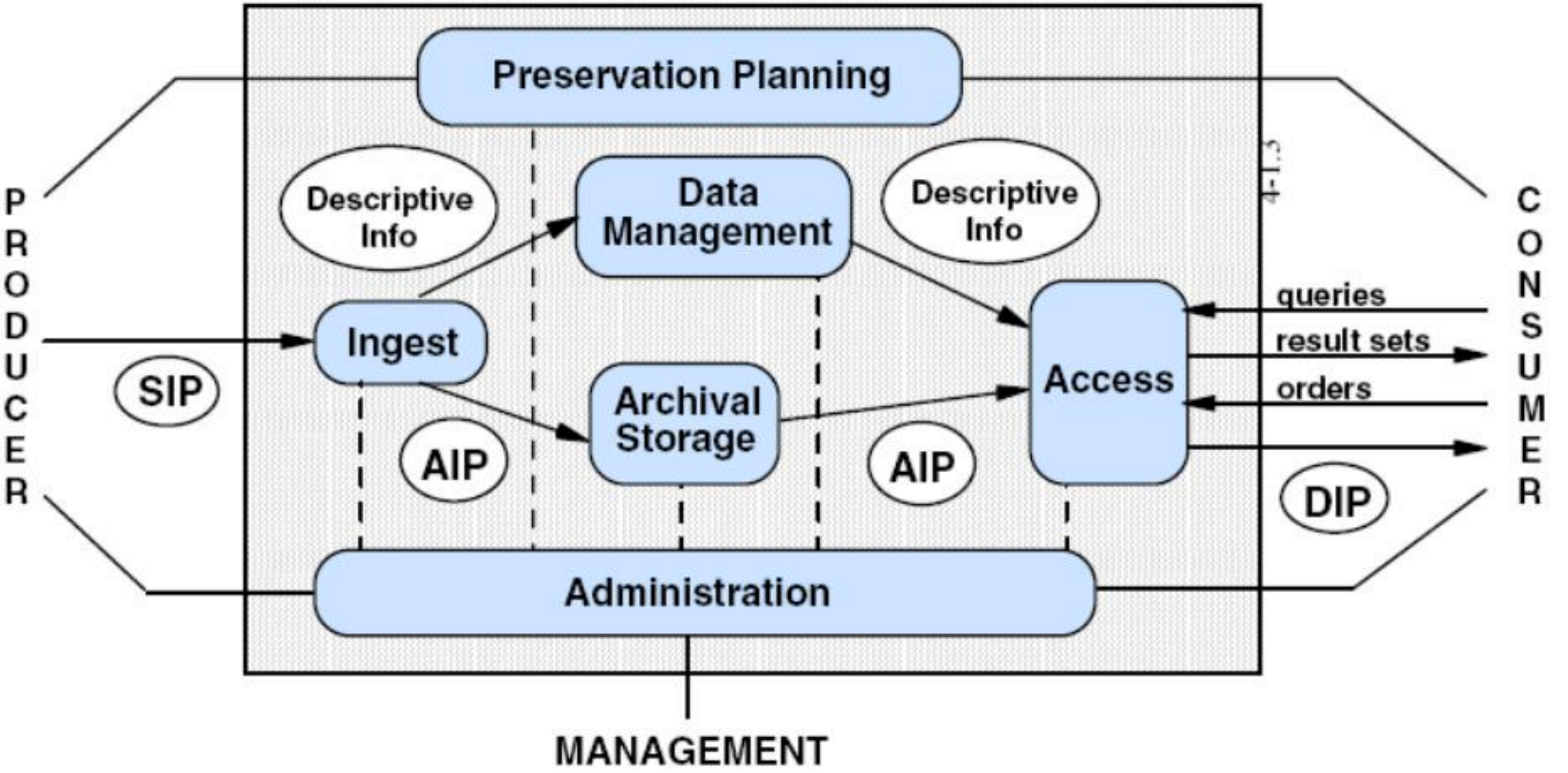
Outline

- Some important digital preservation standards
- Overview of Archivematica
- CONTENTdm and Archivematica integration (UBC example)
- Use cases for Islandora and Archivematica integration
- Strategies for integrating Islandora and Archivematica

Digital preservation strategies

- Normalization
  - On ingestion, convert files to standard, open, proven formats
- Migration
  - Convert files to current standardized formats
- Emulation
  - Recreation of the original environment the digital content was created and used in
- Significant characteristics
  - Those characteristics or properties of a digital object that must be preserved in order to ensure the continued accessibility, usability, and meaning of that object

OAIS



From *Reference Model for an Open Archival Information System (OAIS)*. Washington, DC: CCSDS Secretariat, 2002, page 4-1. http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf

PREMIS

- Digital objects
- Intellectual entities
- Agents
  - Subelements: agentIdentifier, agentType, agentName
- Events
  - Subelements: eventIdentifier, eventType, eventDateTime, eventOutcomeInformation
    - eventTypes: digital signature validation, normalization, virus check, fixity check, replication
- Rights
- Relationships

METS

- METS header
- Descriptive metadata section
- Administrative metadata section
  - PREMIS can be stored here
- File section
- Structural map section
  - Can be physical or logical
- Structural link section
- Behavior section

Overview of Archivematica

- Developed and supported by Artefactual Systems Inc.
- GNU Affero General Public License
- 0.9 beta to be released August 13; 1.0 to be released January 2013
- Clients include City of Vancouver Archives, University of British Columbia Library, SFU Archives, Rockefeller Archive Center, UNESCO
- 10+ workshops over the last 12 months

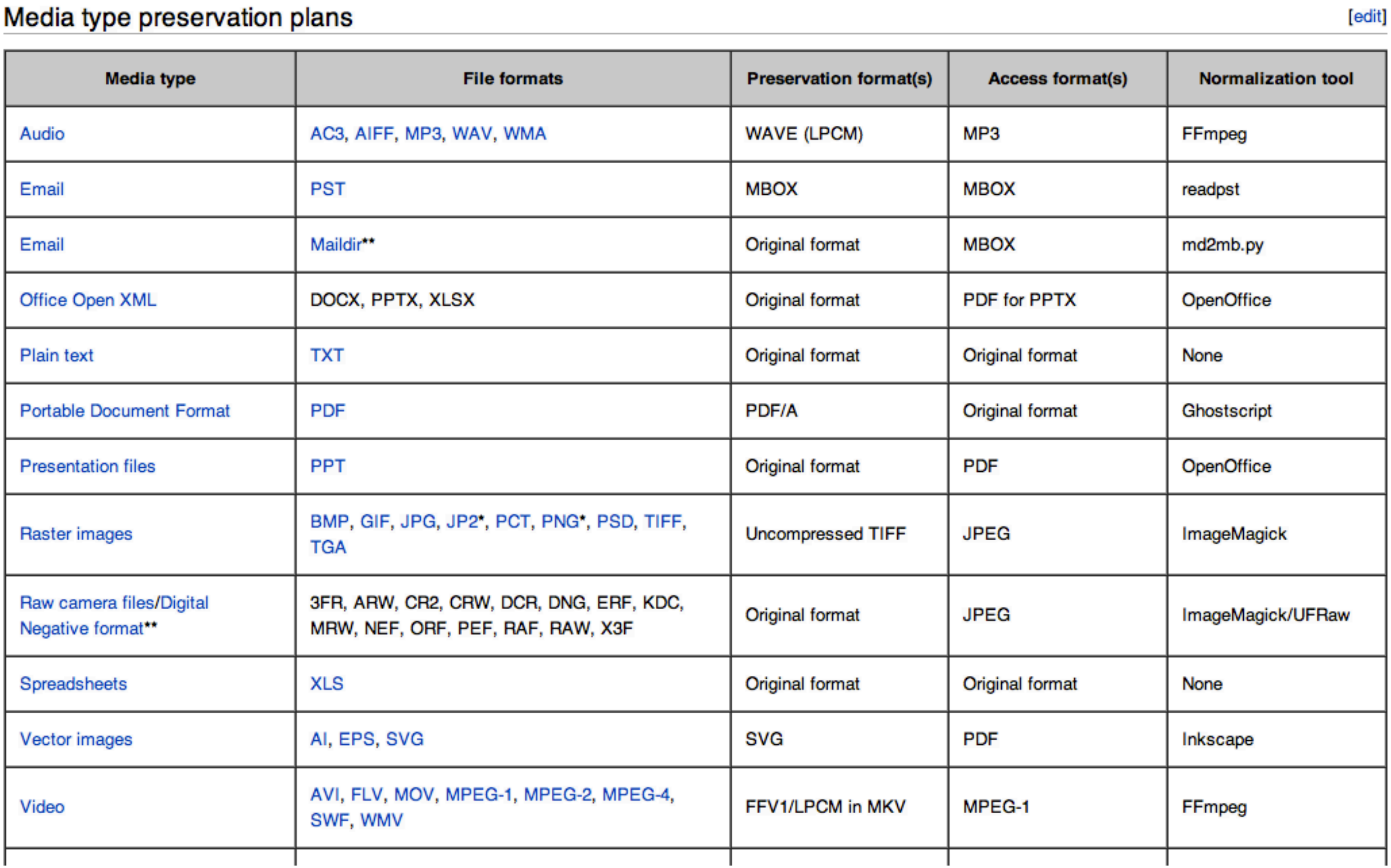
Features

- "Open Source OAIS"
    - SIP to AIP to DIP
- Microservices design pattern
- Dashboard
- Single install
- Distributed processing and storage architecture
- Storage agnostic
- Media-type preservation plans
- Lowers the barriers to best-practice digital preservation
- Uses METS, PREMIS, BagIt

Example microservices

- Transfer
  - Rename with transfer UUID
  - Include default Transfer processingMCP.xml file
  - Scan for viruses
  - Generate METS.xml document
- Ingest
  - Normalize access
  - Normalize preservation
  - Verify checksums generated on ingest
  - Generate DIP
  - Prepare AIP

Media-type preservation plans

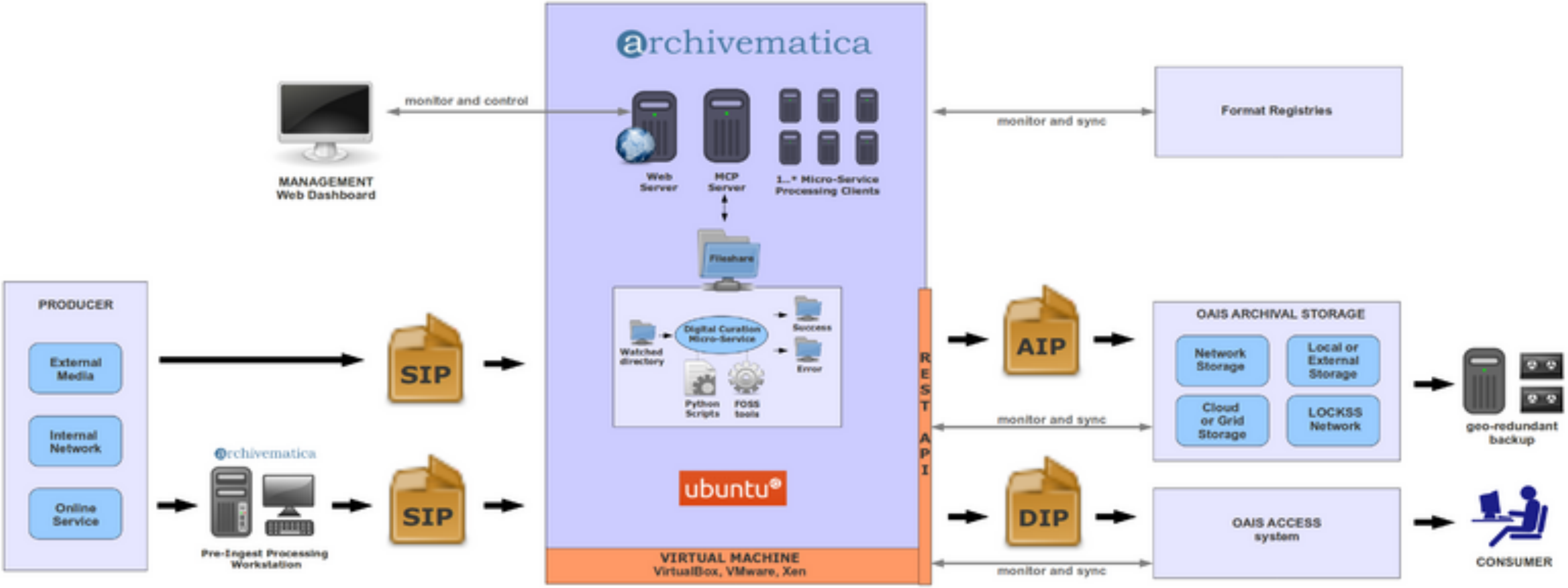


Media type preservation plans

[edit]

Media type	File formats	Preservation format(s)	Access format(s)	Normalization tool
Audio	AC3, AIFF, MP3, WAV, WMA	WAVE (LPCM)	MP3	FFmpeg
Email	PST	MBOX	MBOX	readpst
Email	Maildir**	Original format	MBOX	md2mb.py
Office Open XML	DOCX, PPTX, XLSX	Original format	PDF for PPTX	OpenOffice
Plain text	TXT	Original format	Original format	None
Portable Document Format	PDF	PDF/A	Original format	Ghostscript
Presentation files	PPT	Original format	PDF	OpenOffice
Raster images	BMP, GIF, JPG, JP2*, PCT, PNG*, PSD, TIFF, TGA	Uncompressed TIFF	JPEG	ImageMagick
Raw camera files/Digital Negative format**	3FR, ARW, CR2, CRW, DCR, DNG, ERF, KDC, MRW, NEF, ORF, PEF, RAF, RAW, X3F	Original format	JPEG	ImageMagick/UFRaw
Spreadsheets	XLS	Original format	Original format	None
Vector images	AI, EPS, SVG	SVG	PDF	Inkscape
Video	AVI, FLV, MOV, MPEG-1, MPEG-2, MPEG-4, SWF, WMV	FFV1/LPCM in MKV	MPEG-1	FFmpeg

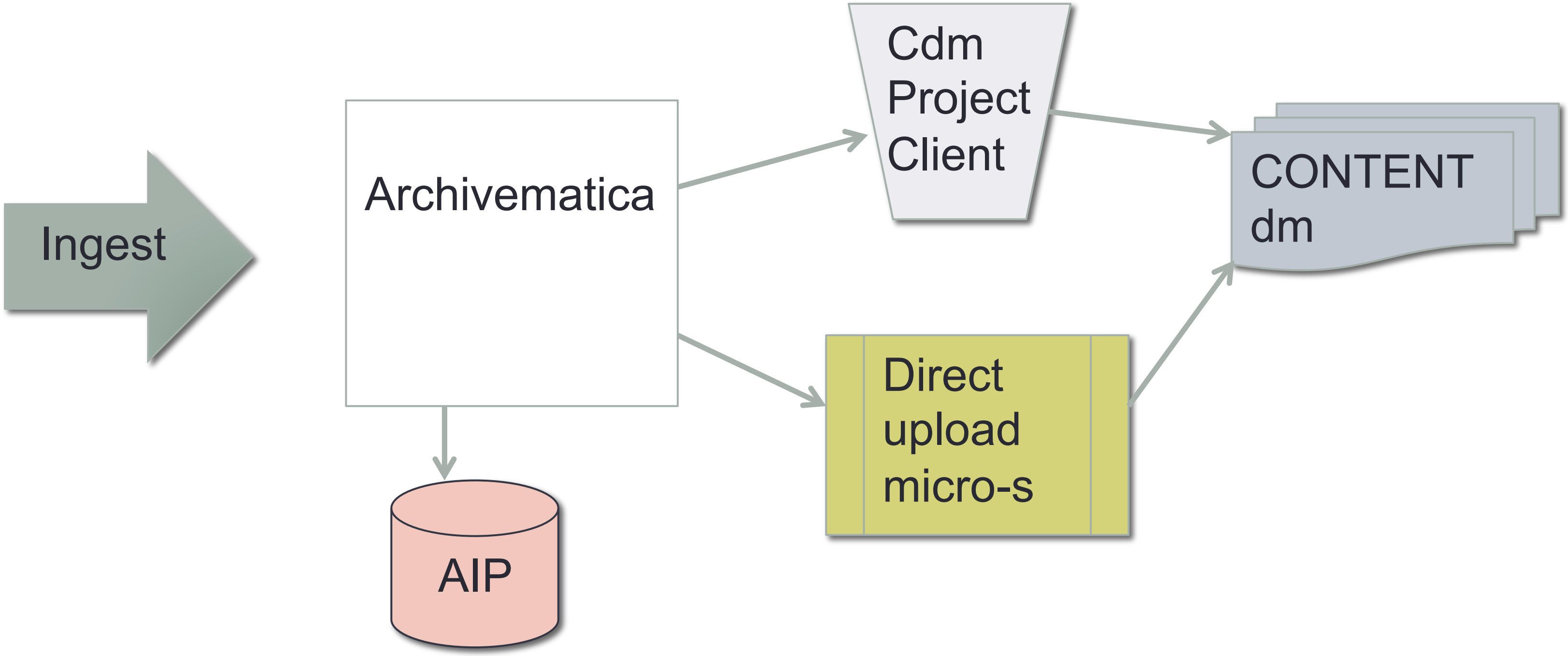
Architecture



https://www.archivematica.org/wiki/File:Archivematica-0.8-beta-architecture.png

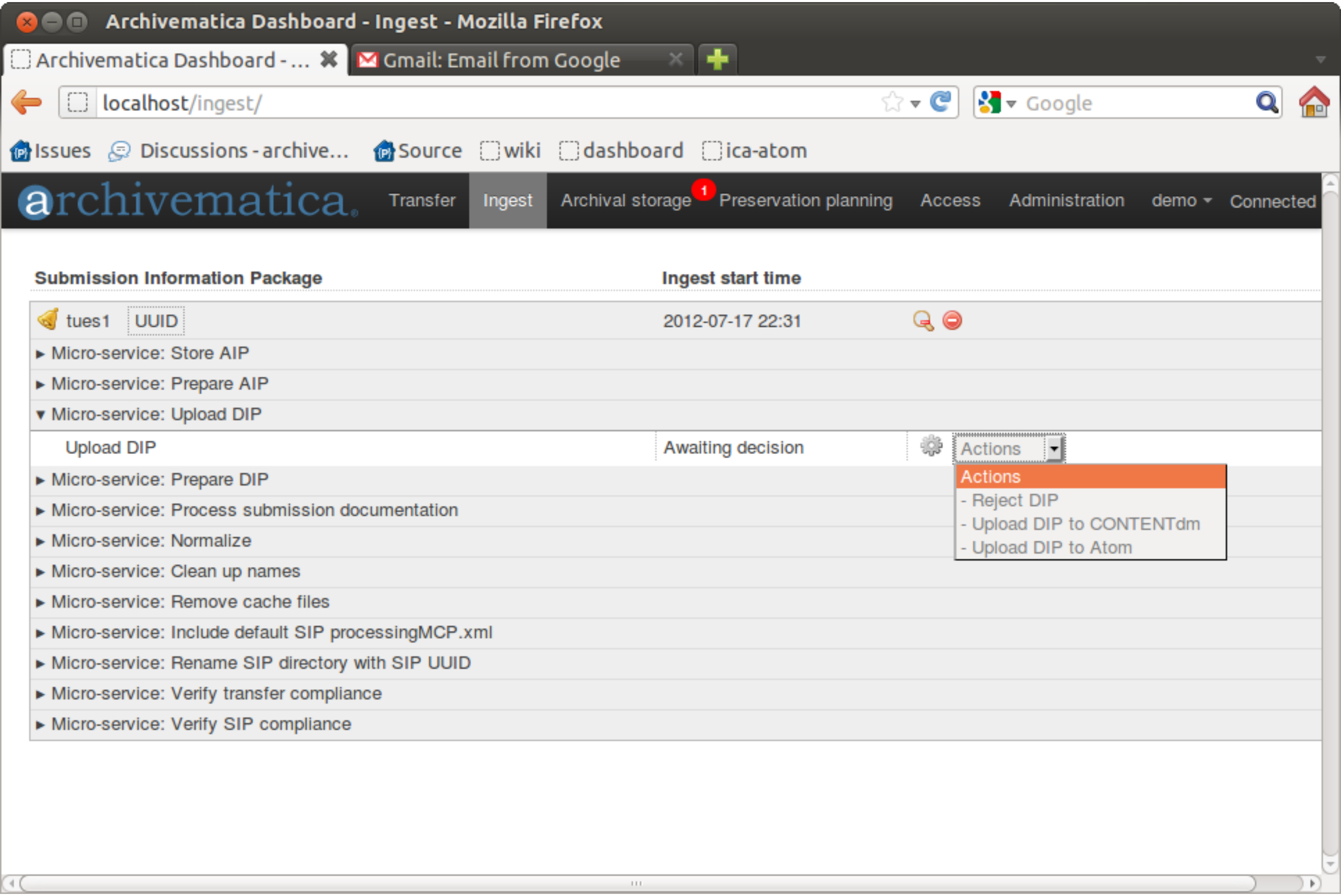
CONTENTdm and Archivematica

- Developed as part of UBC's Archivematica pilot project
- Workflow: ingestion into Archivematica, DIP is generated and uploaded to CONTENTdm

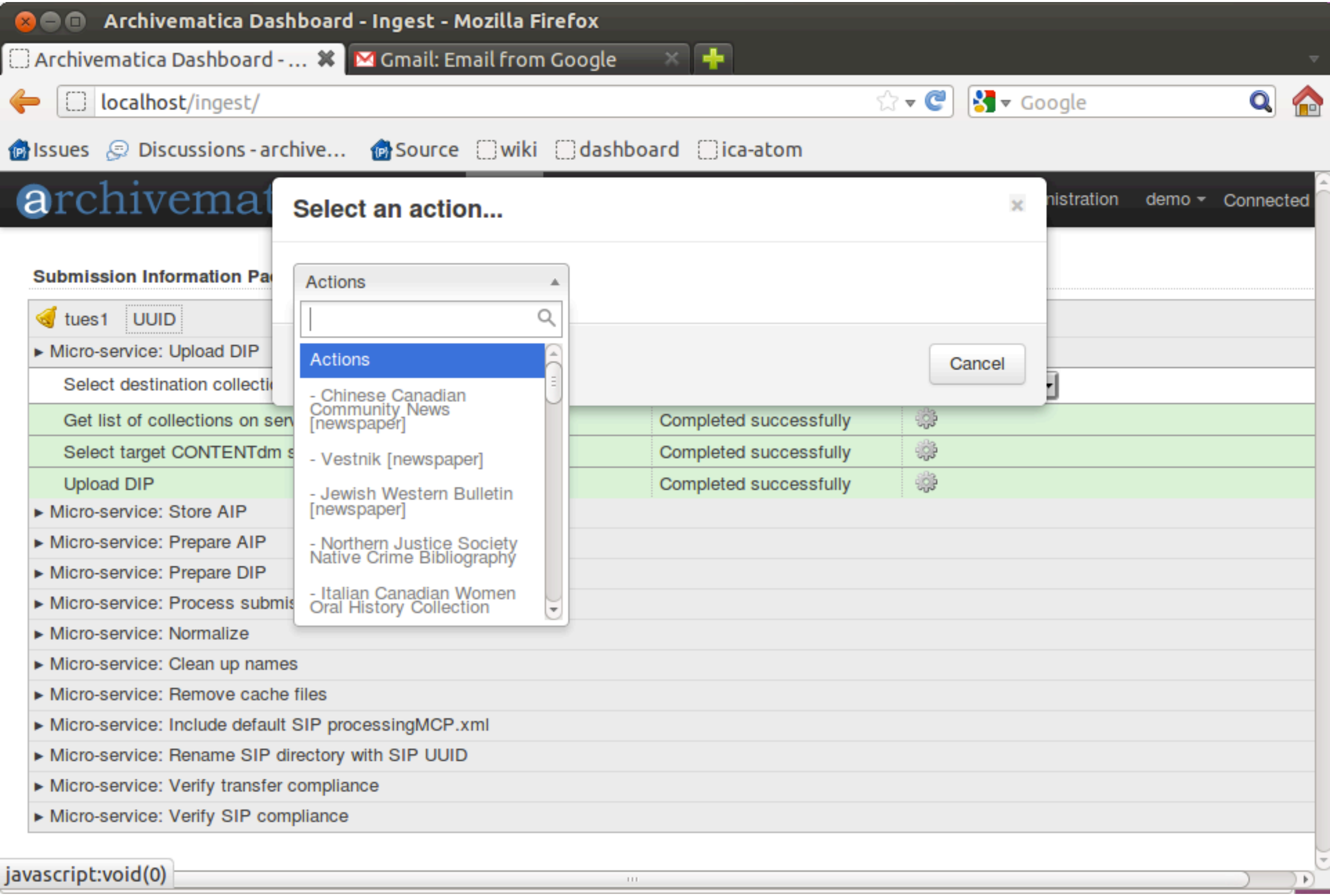


Required microservices

- restructureDIPForContentDMUpload.py
- getContentdmCollectionList.py
- upload-contentDM.py



Submission Information Package	Ingest start time
tues1 UUID	2012-07-17 22:31
▸ Micro-service: Store AIP	
▸ Micro-service: Prepare AIP	
▾ Micro-service: Upload DIP	
Upload DIP	Awaiting decision
▸ Micro-service: Prepare DIP	
▸ Micro-service: Process submission documentation	
▸ Micro-service: Normalize	
▸ Micro-service: Clean up names	
▸ Micro-service: Remove cache files	
▸ Micro-service: Include default SIP processingMCP.xml	
▸ Micro-service: Rename SIP directory with SIP UUID	
▸ Micro-service: Verify transfer compliance	
▸ Micro-service: Verify SIP compliance	



Select an action...

Actions

- Actions
- Chinese Canadian Community News [newspaper]
- Vestnik [newspaper]
- Jewish Western Bulletin [newspaper]
- Northern Justice Society Native Crime Bibliography
- Italian Canadian Women Oral History Collection

Cancel

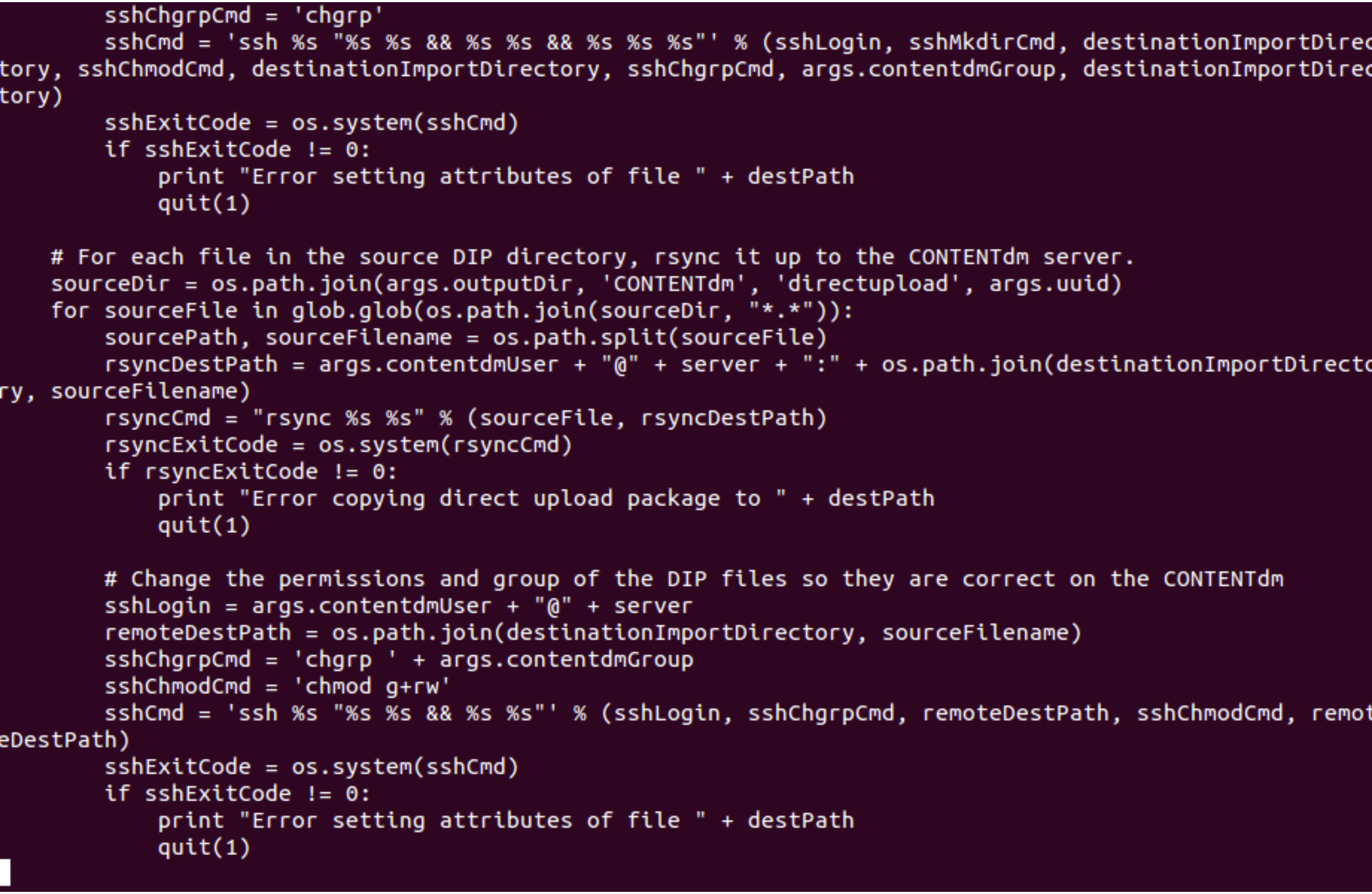
Submission Information Pa

tues1 UUID

▸ Micro-service: Upload DIP

Select destination collecti		
Get list of collections on serv	Completed successfully	⚙
Select target CONTENTdm s	Completed successfully	⚙
Upload DIP	Completed successfully	⚙

▸ Micro-service: Store AIP
▸ Micro-service: Prepare AIP
▸ Micro-service: Prepare DIP
▸ Micro-service: Process submis
▸ Micro-service: Normalize
▸ Micro-service: Clean up names
▸ Micro-service: Remove cache files
▸ Micro-service: Include default SIP processingMCP.xml
▸ Micro-service: Rename SIP directory with SIP UUID
▸ Micro-service: Verify transfer compliance
▸ Micro-service: Verify SIP compliance

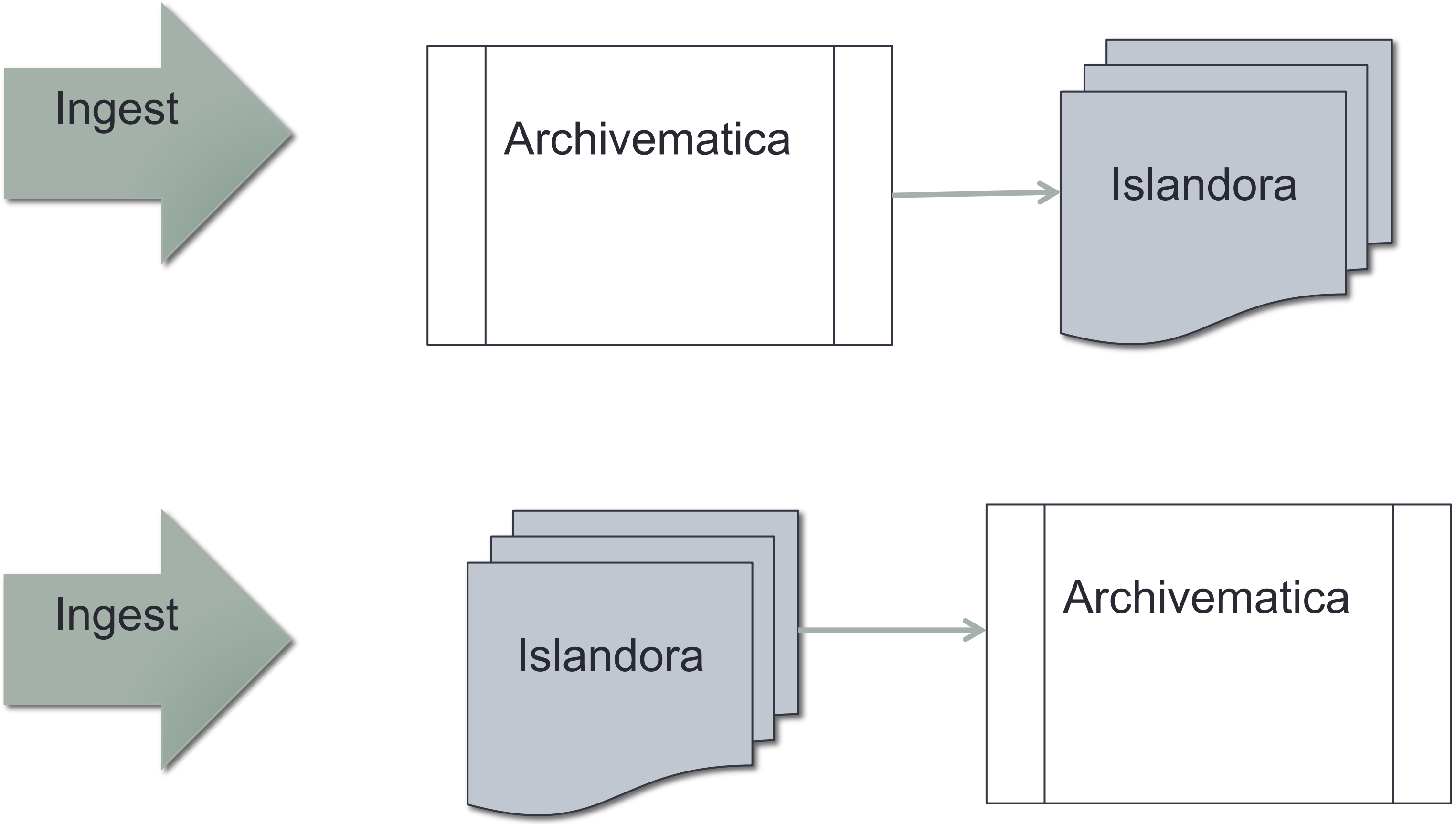


```
        sshChgrpCmd = 'chgrp'
        sshCmd = 'ssh %s "%s %s && %s %s && %s %s %s"' % (sshLogin, sshMkdirCmd, destinationImportDirec
tory, sshChmodCmd, destinationImportDirectory, sshChgrpCmd, args.contentdmGroup, destinationImportDirec
tory)
        sshExitCode = os.system(sshCmd)
        if sshExitCode != 0:
            print "Error setting attributes of file " + destPath
            quit(1)

    # For each file in the source DIP directory, rsync it up to the CONTENTdm server.
    sourceDir = os.path.join(args.outputDir, 'CONTENTdm', 'directupload', args.uuid)
    for sourceFile in glob.glob(os.path.join(sourceDir, "*.*")):
        sourcePath, sourceFilename = os.path.split(sourceFile)
        rsyncDestPath = args.contentdmUser + "@" + server + ":" + os.path.join(destinationImportDirecto
ry, sourceFilename)
        rsyncCmd = "rsync %s %s" % (sourceFile, rsyncDestPath)
        rsyncExitCode = os.system(rsyncCmd)
        if rsyncExitCode != 0:
            print "Error copying direct upload package to " + destPath
            quit(1)

        # Change the permissions and group of the DIP files so they are correct on the CONTENTdm
        sshLogin = args.contentdmUser + "@" + server
        remoteDestPath = os.path.join(destinationImportDirectory, sourceFilename)
        sshChgrpCmd = 'chgrp ' + args.contentdmGroup
        sshChmodCmd = 'chmod g+rw'
        sshCmd = 'ssh %s "%s %s && %s %s"' % (sshLogin, sshChgrpCmd, remoteDestPath, sshChmodCmd, remot
eDestPath)
        sshExitCode = os.system(sshCmd)
        if sshExitCode != 0:
            print "Error setting attributes of file " + destPath
            quit(1)
```

Why integrate Islandora and Archivematica?

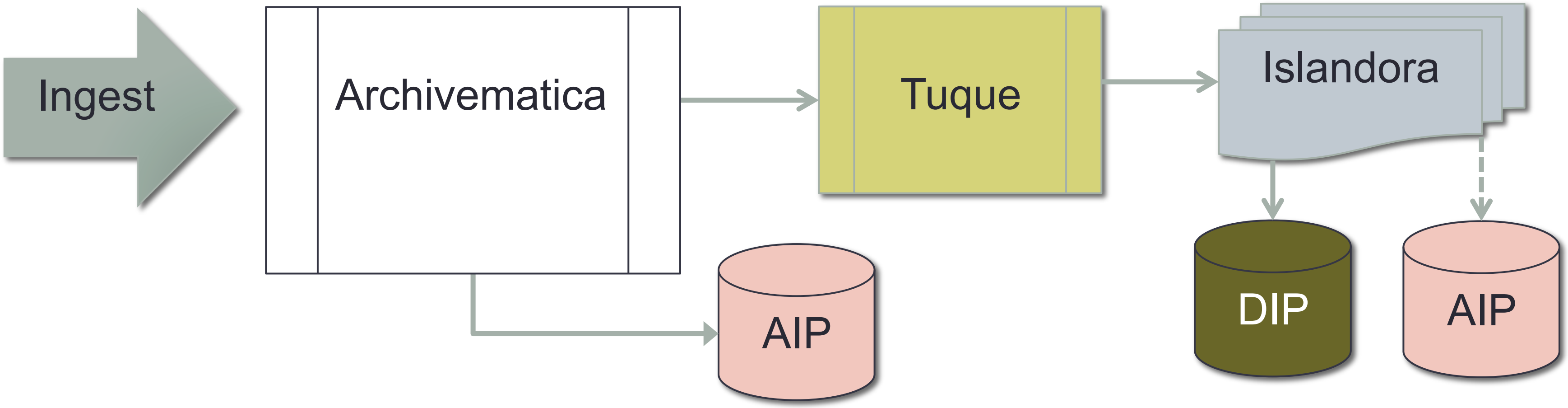
- Ingestion-to-access workflows are orthogonal to digital preservation workflows
- Archivematica is a ready-to-deploy digital preservation stack
    - OAIS, PREMIS, media-type preservation plans, standardized normalization, METS
- Integration of Islandora and Archivematica is an application of the UNIX pipeline / microservice philosophy
    - Both systems are open, flexible, and based on standard tools
    - Both use a Service-Oriented Architecture

Integration use cases



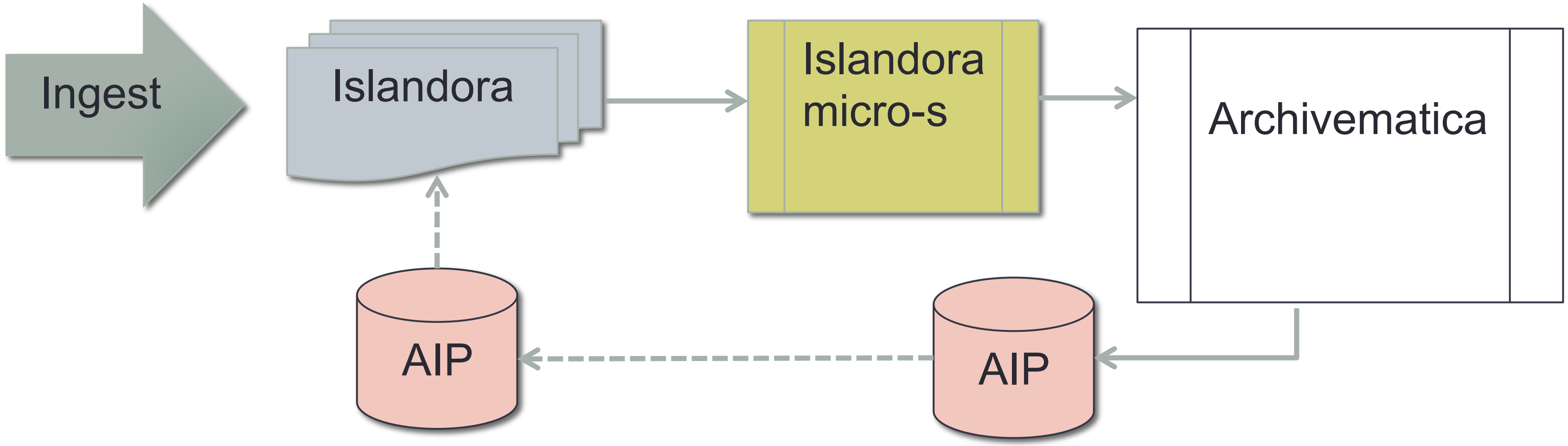
Integration details: Archivematica first

- Similar to CONTENTdm “Upload DIP” model
- Convert Archivematica DIP into a Islandora import package, then ingest via Tuque / some other API
- Could add the zipped AIP as a datastream
- Archivematica would need to know which collection / content model to use



Integration details: Islandora first

- Islandora microservices create an Archivematica transfer package and move it to the transfer directory
- Archivematica's automated workflow kick in and push the transfer from SIP through to AIP
- Could add the zipped AIP as a datastream (e.g., through Tuque)
- Archivematica can remain totally agnostic to Islandora content models



Development required

	Archivematica	Islandora
Archivematica first	• Microservices to convert DIP to Islandora import packages • AIP synchronization services	NULL
Islandora first	• Automated workflow trigger interface • AIP synchronization services	• Microservices to create an Archivematica transfer package

Summary

- Islandora rocks!
- Archivematica rocks!
- Each does its own thing well
- It just makes sense to chain the two together (if you want full digital preservation functionality)
- We have the technology, let’s make a $6,000, configure-and-play super-stack