ARCITECTA™

www.arcitecta.com

sgi®

Mediaflux™

OPERATING SYSTEM FOR META+DATA

Optimised Packaging of Data

Jason Lohrey

Chief Technology Officer
Arcitecta Pty. Ltd.

5th December 2012

Overview ..

This presentation will look at solutions the following problem…

How to reduce the number and size of files for:

- Storage – reducing the load on file systems and DMF
- Network transmission – reducing the load on IP networks when transmitting data

Presentation in two acts ..

- Act 1: What we are doing..
- Act 2: What you would like us to do/interested in..

What's the problem?

.. There are too many files.

Storage:

- Every file requires:
  - An i-node (128-512 bytes per file)
  - Possibly indexes for path/name retrieval
  - Modulo block size

- In the case of DMF:
  - Each file over a threshold is migrated to tape

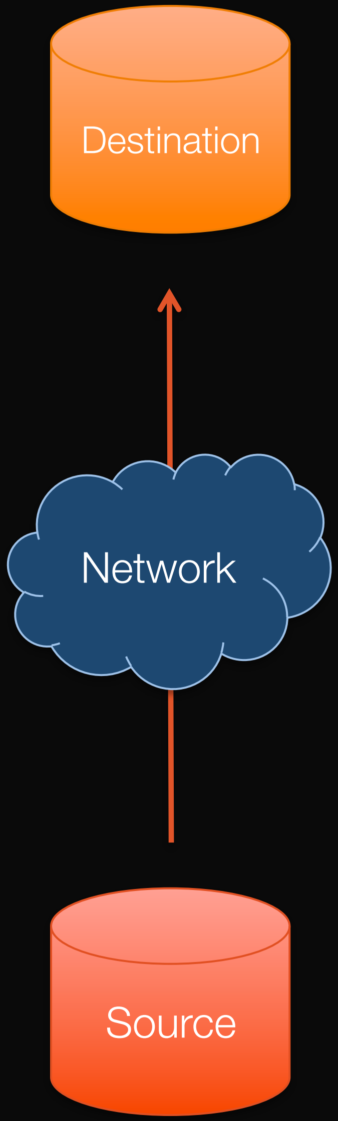
What's the problem?

What happens when you try to transfer 100,000 or a million files from one place to another?

Network:

Transferring across a network requires a "transaction" per file:

- NFS:
  - Lookup
  - Read + Attributes (pre/post)
- CIFS …



What's a solution?

We need to step back a bit.. and look at the bigger picture.

There are two classes of problems that lend themselves to reorganizing the data:

File clustering – often files are related (part of a project, product, set..)

File compression – many file types can be well compressed

Container Format:

Arcitecta Archive Format (AAR):

- Up to 2^{63} byte total file size
- Up to 2^{63} bytes per entry
- Uncompressed, or Deflate compression (*later*: adaptive compressors)
- Streamed or random access
- Created/Read by:
  - Aterm
  - Web-clients
  - Stand-alone AAR.jar tool

Container Format:

Arcitecta Archive Format (AAR):

- Up to 2^{63} byte total file size
- Up to 2^{63} bytes per entry
- Uncompressed, or Deflate compression (*later*: adaptive compressors)
- Streamed or random access
- Created/Read by:
    - Aterm
    - Web-clients
    - Stand-alone AAR.jar tool

Container Format:

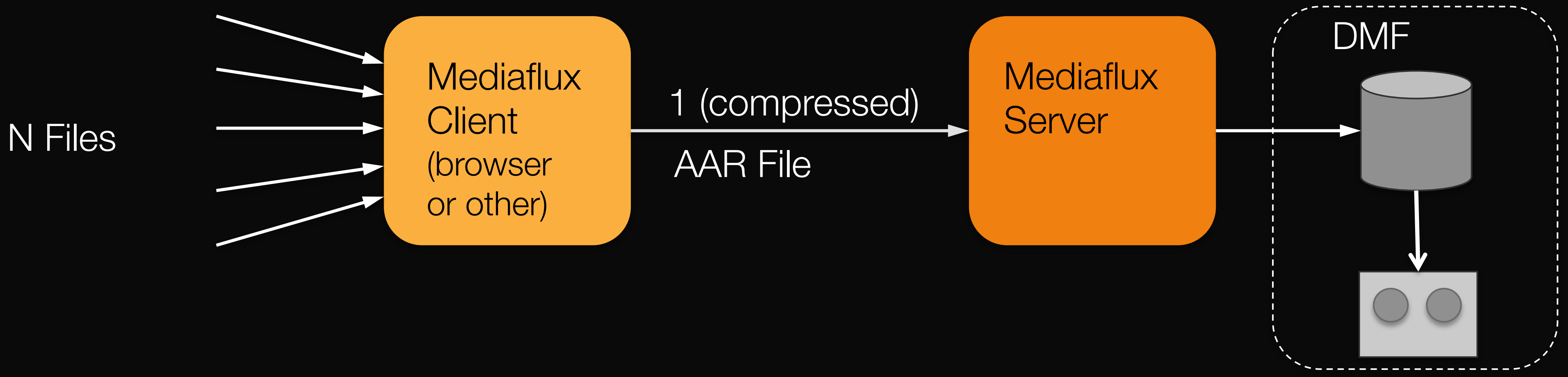
Arcitecta Archive Format (AAR):

- Parallel compression/decompression
- Segment/chunk checksums
- Splitting and splicing (without decompressing)
- Extraction of TOC for external (e.g. database) storage

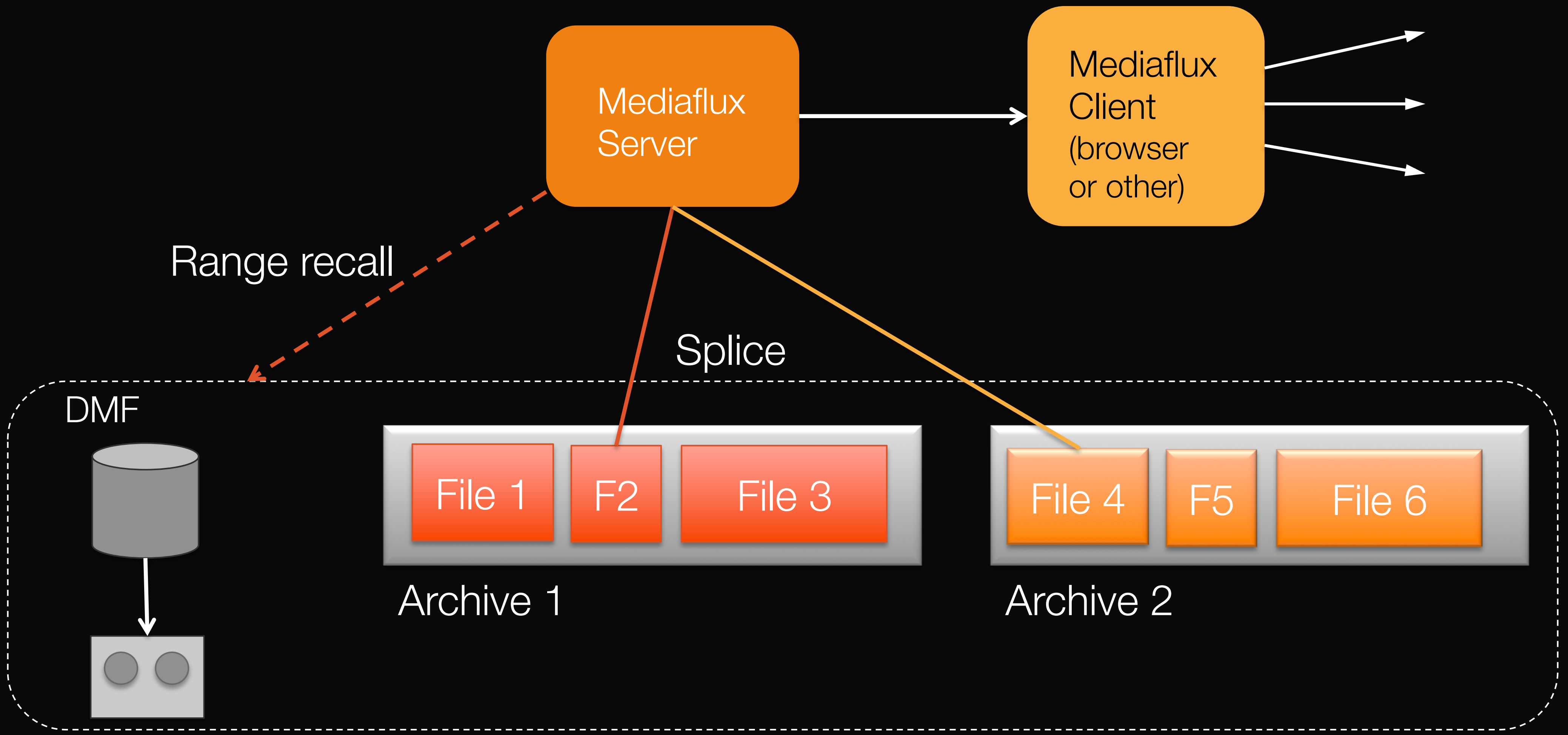
Pipeline:

Mediaflux can coalesce many files into a single Arcitecta Archive (AAR), and automatically re-inflate on extraction. This:

- Significantly reduces the number (and potentially size) of files managed by server side storage (e.g. HSM)
- Significantly reduces the number of files transmitted versus other mechanisms such as Windows Explorer, etc.



Extraction:

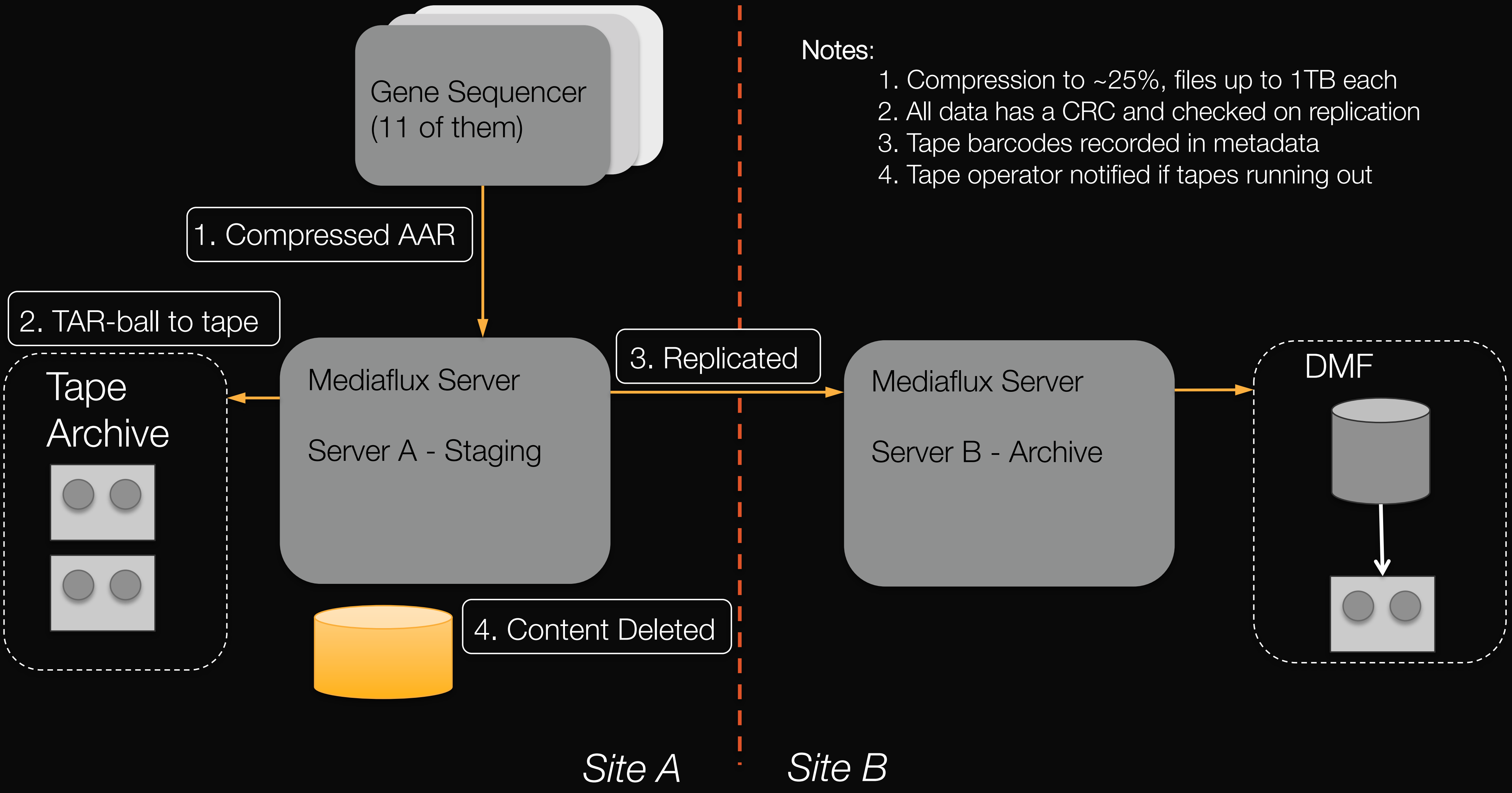


Mediaflux Server
Mediaflux Client (browser or other)
Range recall
Splice
DMF
File 1
F2
File 3
Archive 1
File 4
F5
File 6
Archive 2

University of Queensland
Institute of Molecular Bioscience
Ingestion

Notes:

1. Compression to ~25%, files up to 1TB each
2. All data has a CRC and checked on replication
3. Tape barcodes recorded in metadata
4. Tape operator notified if tapes running out



IMB:

A typical example is:

10,237 files containing Long Mate Pair genome sequence data totaling 269GB are packaged into one file of less than 92GB. Hundreds of these runs.

They can then extract the entire ensemble, or selectively extract individual files for HPC processing.

Defence:

A typical example is:

Controlled Image Base (CIB) – typically 1000-1500 files per *product.*

DTED

ADRG

ESRI Shape files

Etc..

20,000 CIB archives = 20 to 35 million files

File Compilation Profiles (FCP):

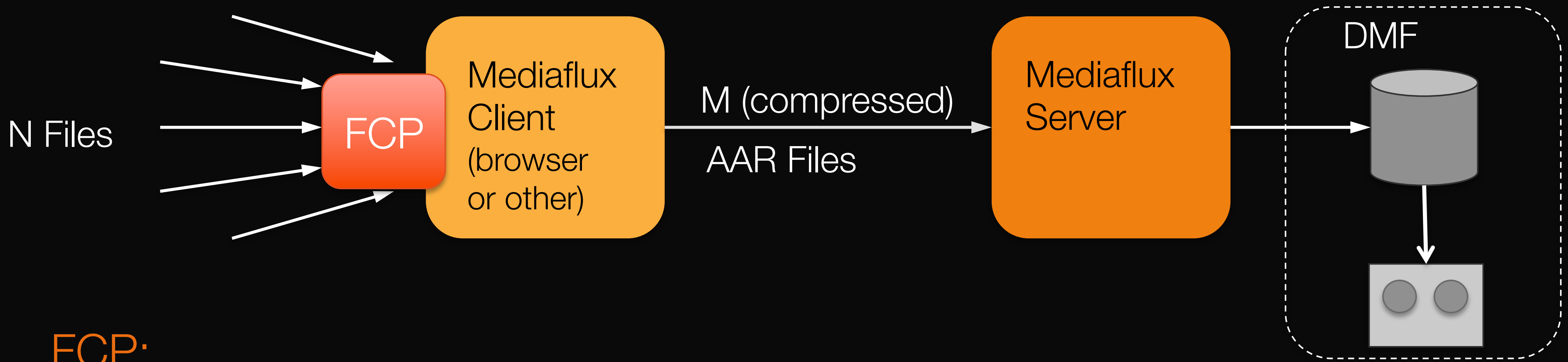
FCP's provide a lexical analyzer for the structure of a file system.

- Allows semantically aware:
  - Clustering
  - Type recognition
  - Metadata extraction
- Looks at:
  - File names
  - Magic numbers
  - Patterns
  - File contents

Sample FCP:

```
profile DICOM {

   construct DCM-FILE {
      match {
         file contains bytes 4449434d at 128
      }
   }

   construct DCM-SET {
      match {
         group unnamed {
            construct DCM-FILE
         }
      }

      encapsulate as archive level 6
      logical type "dicom/set"

      consume yes

      consumer {
         service "dicom.ingest"
            arguments "<engine>pss</engine><arg name='pss.asset.namespace.root'/><arg
name='pss.id.subject.by'>patient.id</arg><arg name='some.variable'/>"

            add "namespace" value at "arg[@name='pss.asset.namespace.root']"

            add "variable:some.value" value at "arg[@name='some.variable']"
      }
   }

}
```

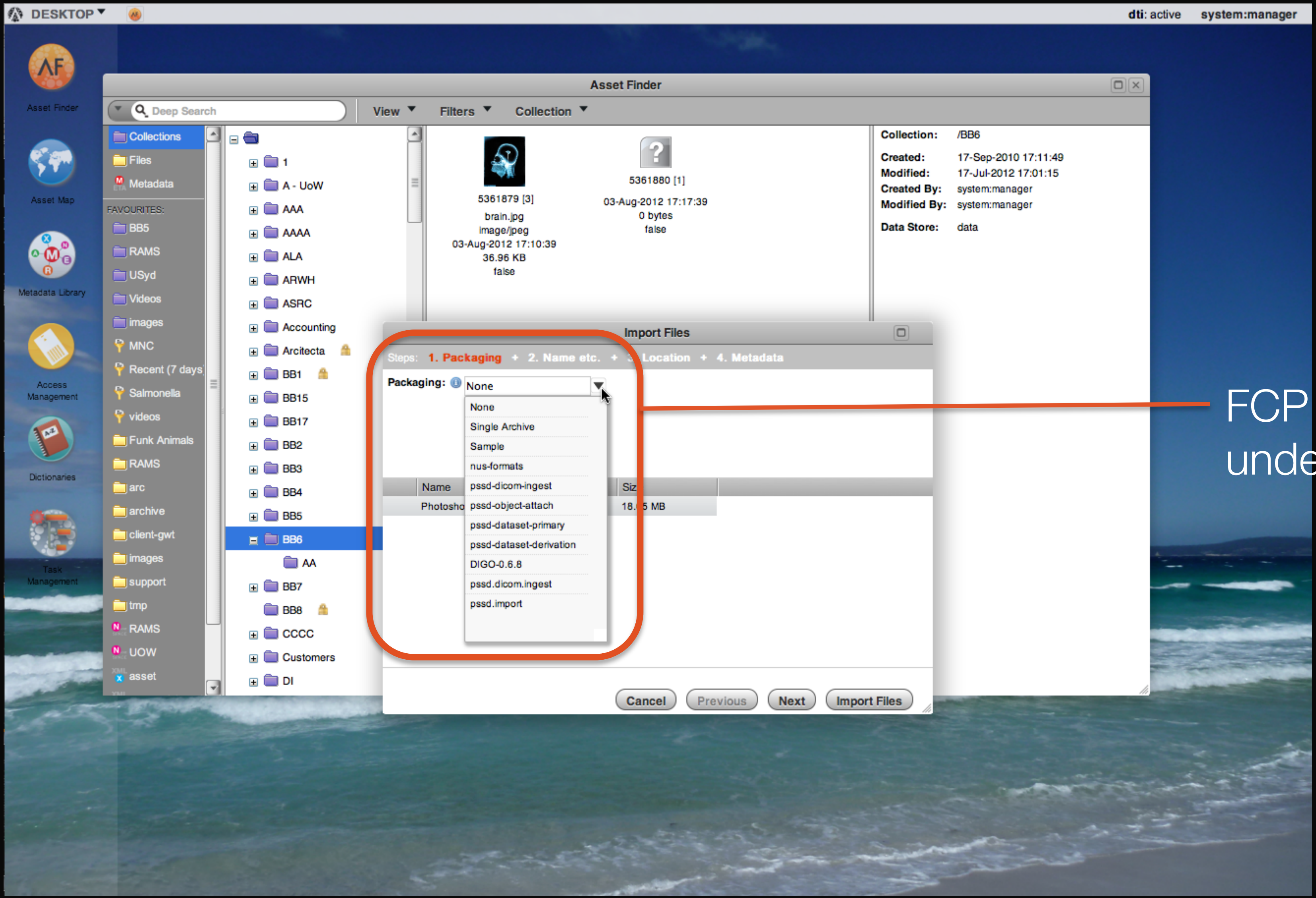
FCP + AAR:



FCP:

- Can be:
  - Initiated by a human
  - Scripted/automated
- Can walk through complete file systems

Desktop – Selecting FCP:



5361879 [3]
brain.jpg
image/jpeg
03-Aug-2012 17:10:39
36.96 KB
false

5361880 [1]
03-Aug-2012 17:17:39
0 bytes
false

Collection:	/BB6
Created:	17-Sep-2010 17:11:49
Modified:	17-Jul-2012 17:01:15
Created By:	system:manager
Modified By:	system:manager
Data Store:	data

Packaging: None

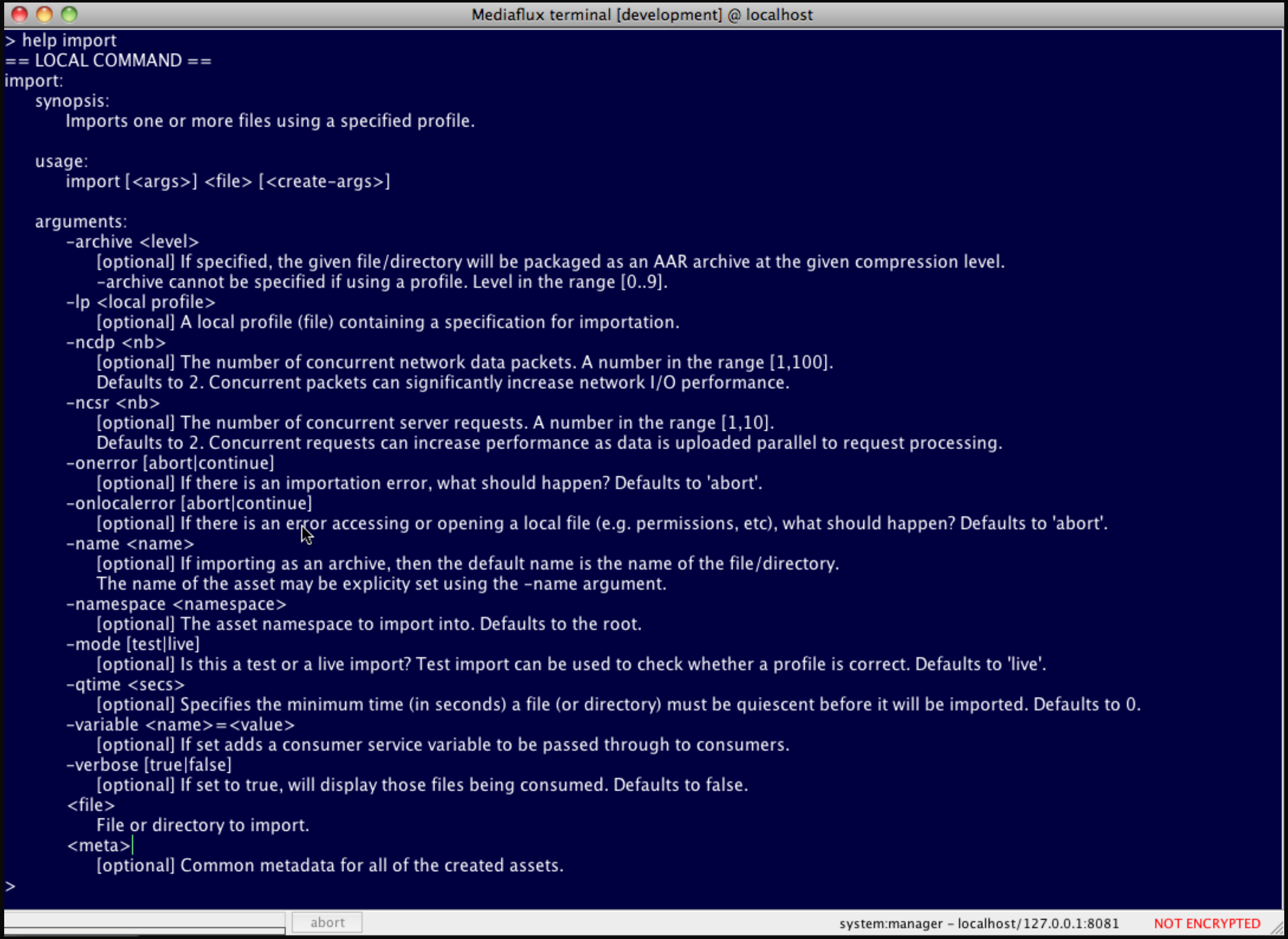
Name	Siz
Photosho	18.65 MB

FCP chosen under *Packaging*

Command Line:

ATERM:

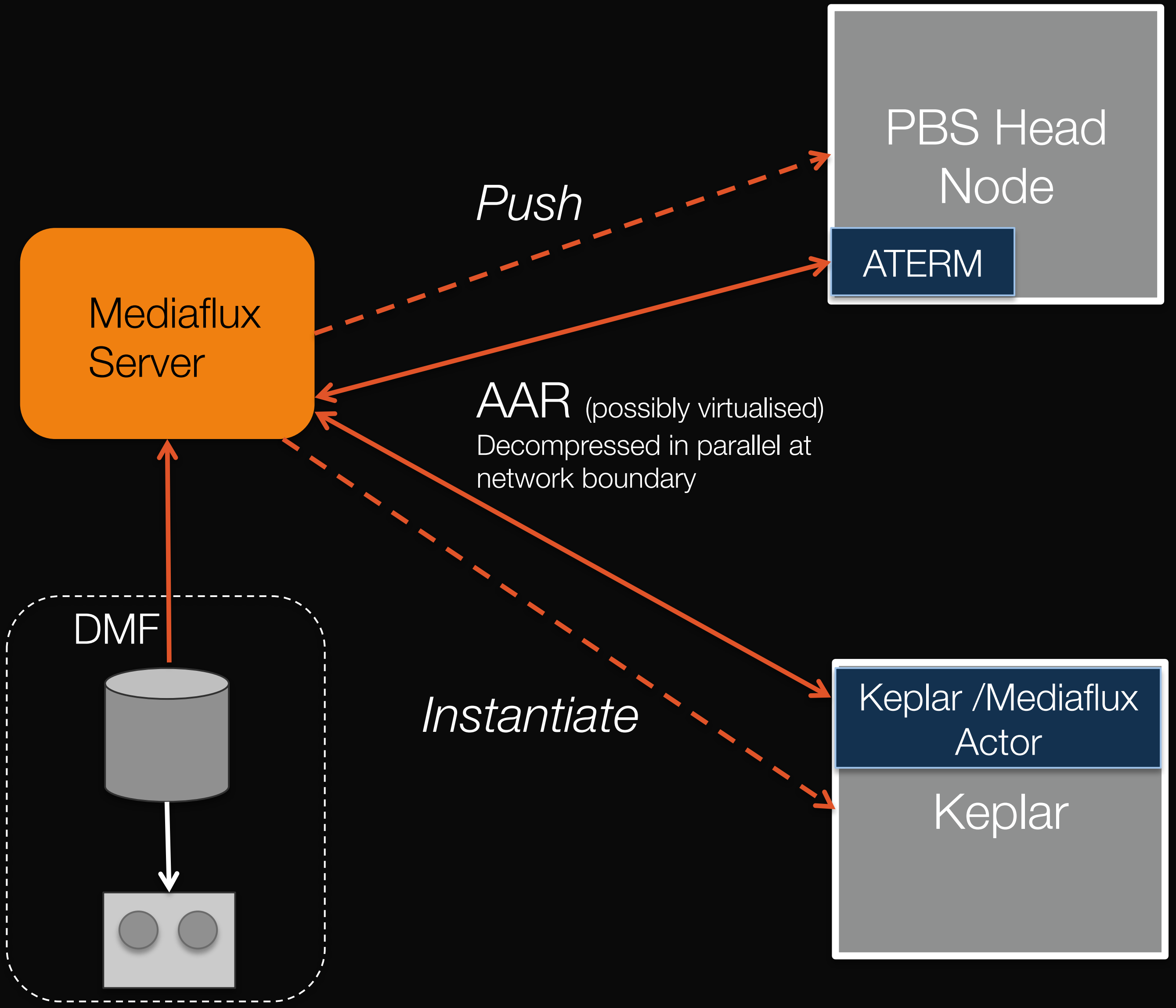
- Select FCP
- Can be scripted.



Mediaflux terminal [development] @ localhost
abort
system:manager – localhost/127.0.0.1:8081
NOT ENCRYPTED

```
> help import
== LOCAL COMMAND ==
import:
    synopsis:
        Imports one or more files using a specified profile.

    usage:
        import [<args>] <file> [<create-args>]

    arguments:
        -archive <level>
            [optional] If specified, the given file/directory will be packaged as an AAR archive at the given compression level.
            -archive cannot be specified if using a profile. Level in the range [0..9].
        -lp <local profile>
            [optional] A local profile (file) containing a specification for importation.
        -ncdp <nb>
            [optional] The number of concurrent network data packets. A number in the range [1,100].
            Defaults to 2. Concurrent packets can significantly increase network I/O performance.
        -ncsr <nb>
            [optional] The number of concurrent server requests. A number in the range [1,10].
            Defaults to 2. Concurrent requests can increase performance as data is uploaded parallel to request processing.
        -onerror [abort|continue]
            [optional] If there is an importation error, what should happen? Defaults to 'abort'.
        -onlocalerror [abort|continue]
            [optional] If there is an error accessing or opening a local file (e.g. permissions, etc), what should happen? Defaults to 'abort'.
        -name <name>
            [optional] If importing as an archive, then the default name is the name of the file/directory.
            The name of the asset may be explicity set using the -name argument.
        -namespace <namespace>
            [optional] The asset namespace to import into. Defaults to the root.
        -mode [test|live]
            [optional] Is this a test or a live import? Test import can be used to check whether a profile is correct. Defaults to 'live'.
        -qtime <secs>
            [optional] Specifies the minimum time (in seconds) a file (or directory) must be quiescent before it will be imported. Defaults to 0.
        -variable <name>=<value>
            [optional] If set adds a consumer service variable to be passed through to consumers.
        -verbose [true|false]
            [optional] If set to true, will display those files being consumed. Defaults to false.
        <file>
            File or directory to import.
        <meta>
            [optional] Common metadata for all of the created assets.
>
```

HPC:


PBS Head Node
ATERM
Push
Mediaflux Server
AAR (possibly virtualised)
Decompressed in parallel at network boundary
DMF
Instantiate
Keplar /Mediaflux Actor
Keplar

POSIX:

May expose the contents of an archive via:

- POSIX
- NFS

May post process in the server to analyze and convert to archives (not as optimal).

Can use file pattern matching (FCP)

Tricky Things (need to solve):

POSIX view of containerized files is read-only… what are the semantics of a file write?

How do you create derivative sets?

Check out, check-in deltas?

Tricky Things (can be solved):

POSIX view of containerized files is read-only… what are the semantics of a file write?

How do you create derivative sets?

Check out, check-in deltas?

Using:

- Multiple archives
- Split and join

… Anything else?

ARCITECTA™

Mediaflux™

OPERATING SYSTEM FOR META+DATA

www.arcitecta.com