



Mediaflux™

OPERATING SYSTEM FOR META+DATA

Dynamic File Systems

Jason Lohrey

Chief Technology Officer
Arcitecta Pty. Ltd.

5th December 2012

Introduction ..

Typically, once we create a file system there is only one layout... the layout we created.

However, we have a lot of metadata, can we use that to dynamically create a file system?

... that is, a file system that is not static, but rather **changes shape dynamically** according to some defined rules..

Metadata

Information about.. information

- Information about an asset (that may not be in the content)
- Stored as binary XML
- Automatically generated:
 - Revision history (audit trail: who, when, what)
 - Plug-in meta data generation
- User generated
- Validated against a Mediaflux document definition
- Server-side manipulation & merging
- Unstructured and/or structured querying

Applying Metadata

At Any Time..

- Metadata simply “attached” as required to an asset
- Can attach any number of different fragments of metadata

```
asset.query :where \
xpath(revision/status)='ACCEPTED'
```

```
asset.query :where \
text contains 'spotted quoll'
```

Revision:

status: ACCEPTED

by: Bill Smith



Note: Very fine example of a Spotted Quoll

Evolving Metadata

At Any Time

- Who knows what metadata might be required in 2/5/10 .. years?
- Can evolve by:
 - Adding to existing definitions
 - Defining and “attaching” new definitions
- Evolve incrementally or by direct update

Revision:

```
status: enumeration {  
    ACCEPTED,  
    REJECTED  
}, max-occurs 1
```

by: String, max-occurs

1



Revision:

```
status: enumeration {  
    ACCEPTED,  
    REJECTED  
}, max-occurs 1
```

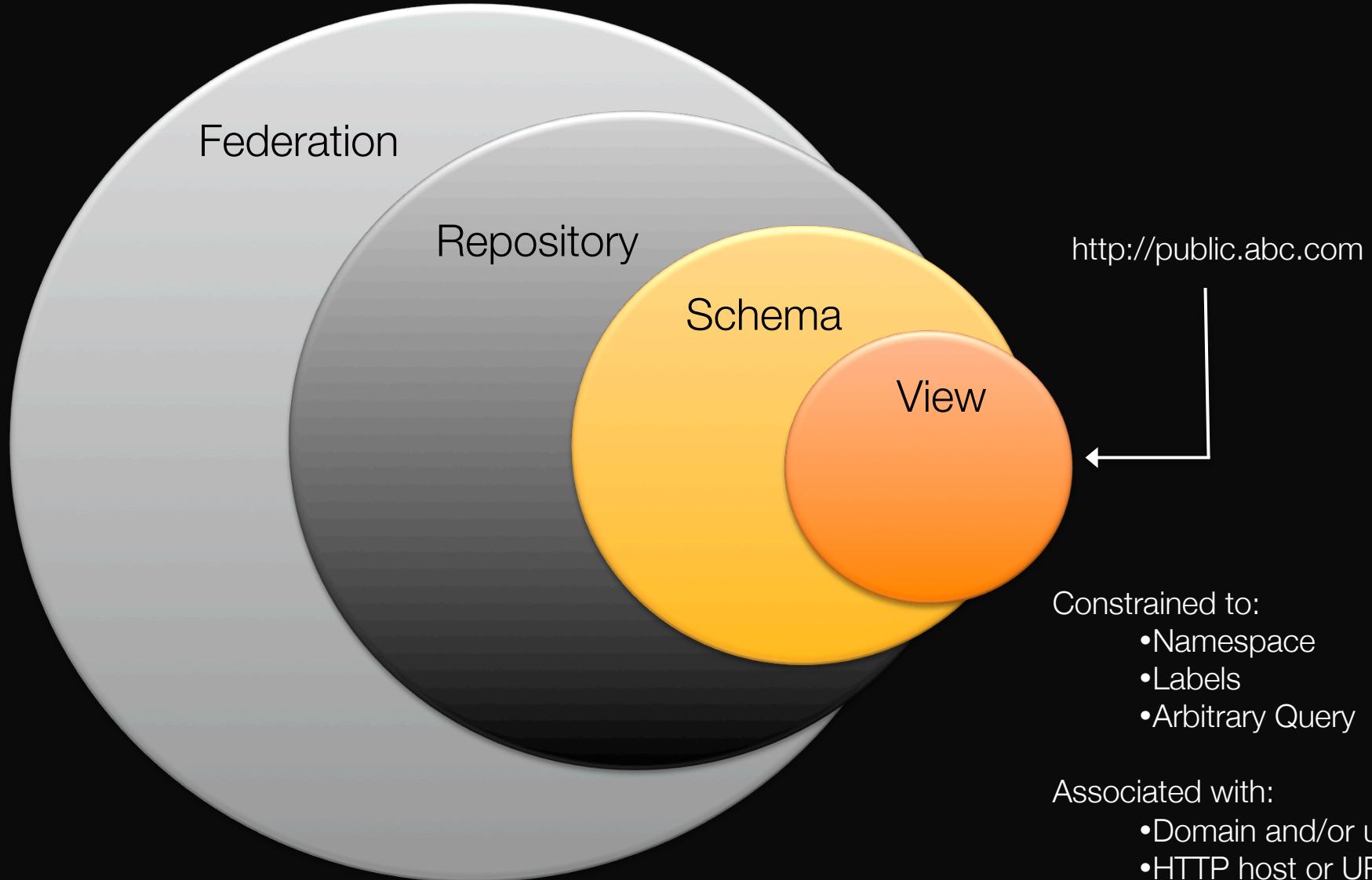
by: String, max-occurs 1

publish-date: Date, max-occurs
1

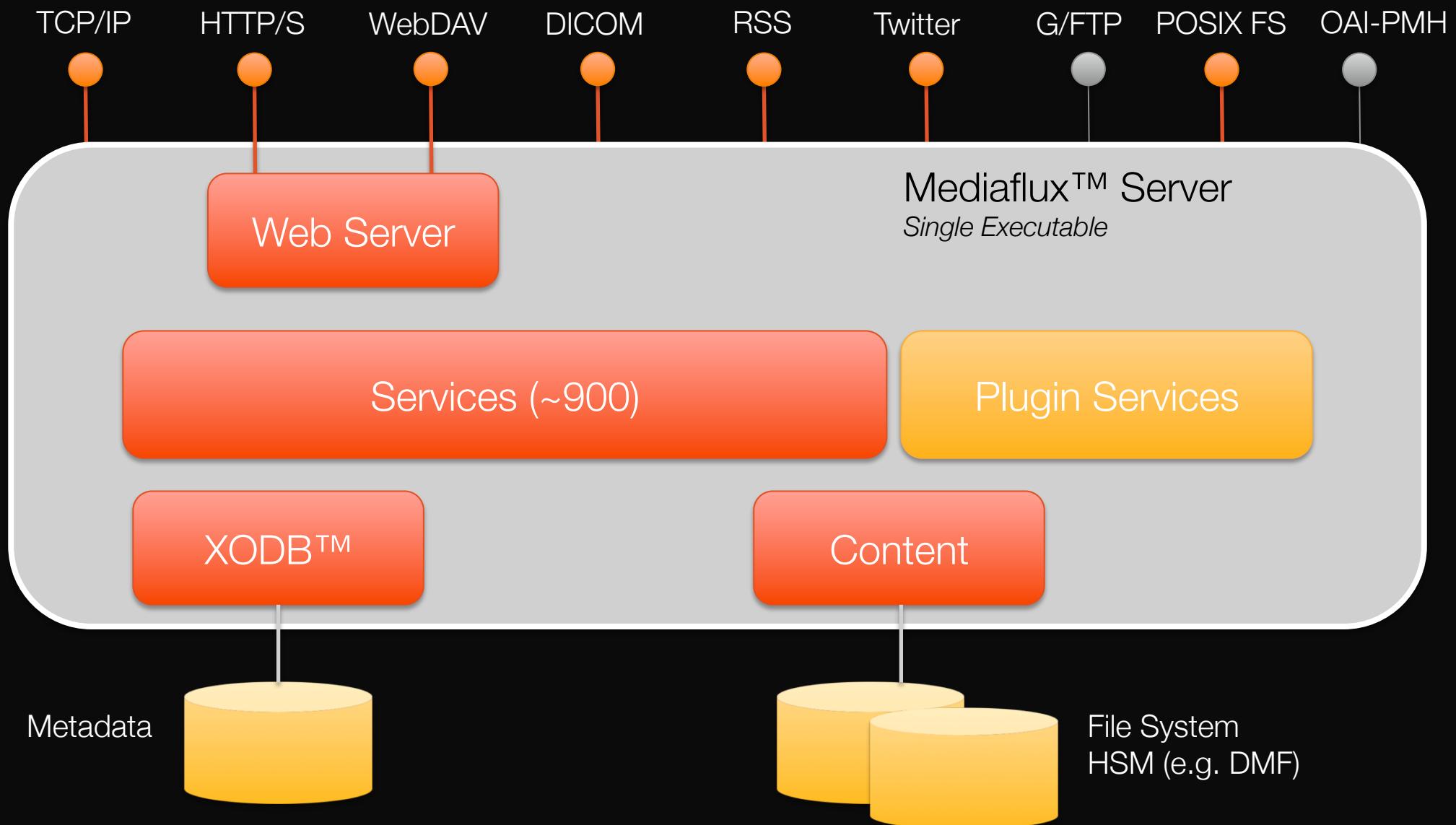
ANATOMY OF AN ASSET



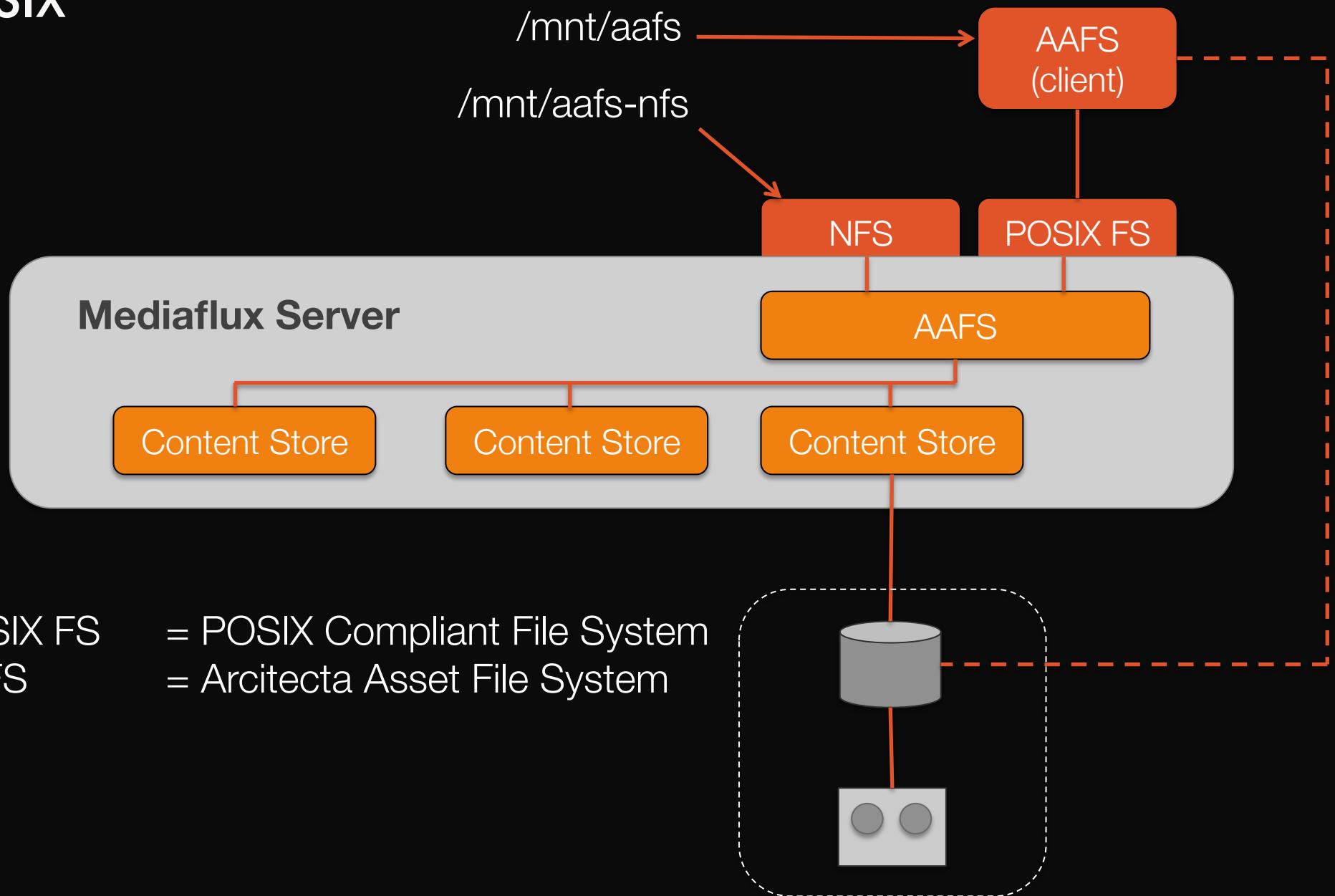
Views



Architecture



POSIX



Semantics..

Namespace = directory
Asset = file

Access Control Lists mapped into permissions for:

- Owner
- Group
- Other

Identity Map

POSIX provides little information for identity..

Kernel:

- UID
- GID
- PID
- Mount point

UID Map:

501 -> user “fred”
502 -> user “admin”

NFS:

- UID
- GID
- IP address
- Mount point

GID Map:

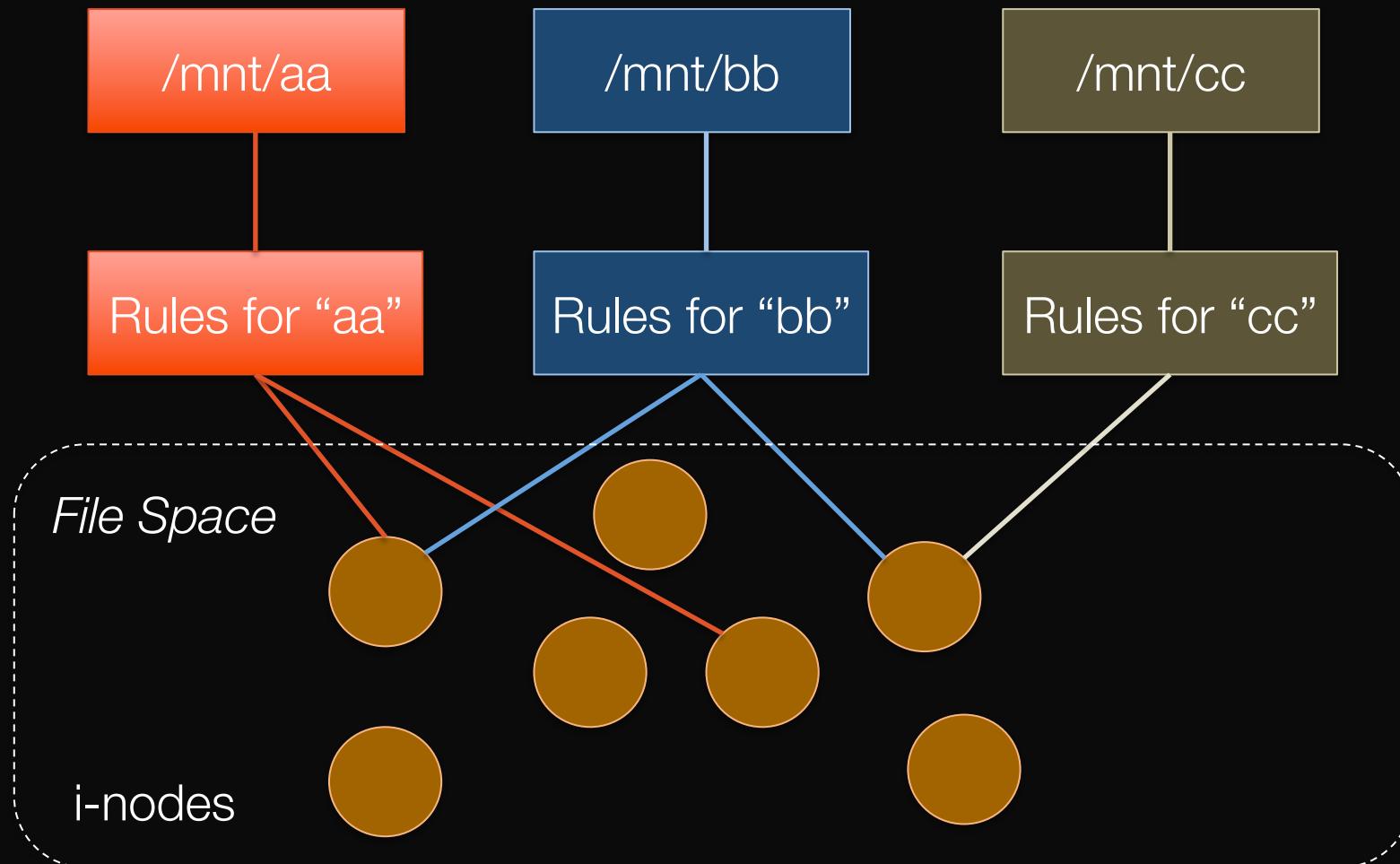
30 -> role “staff”
88 -> role “admins”

Identity Can be Used..

When mounting, we can specify the rules for the structure of the file system:

- Based on a “view”:
 - Arbitrary query
 - Labels (e.g. only show “PUBLISHED” versions)
- Based on Identity:
 - who you are
 - where you have come from
 - And your authority
- Based on a template

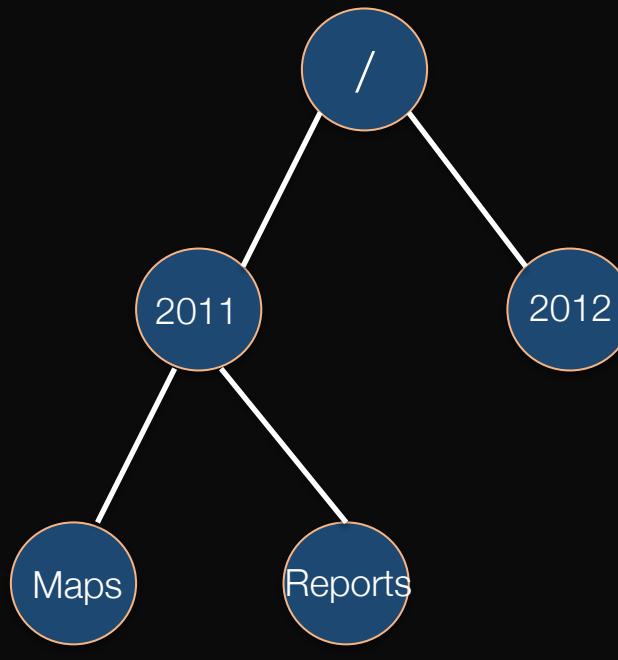
There can be many mounts on the same files ..



Templates ..

Using **functions**, specify the virtual directories:

- Level 1 = year(ctime)
- Level 2 = month(ctime)
- Level 3 = xvalue('myproduct/type')



Functions ..

The Mediaflux function language:

- Built in functions
- Plugin functions
- Calling a plugin service
- Expressions:
 - Boolean
 - Computation

```
> function.list
:function "add"
:function "and"
:function "average"
:function "boolean"
:function "cast"
:function "choose"
:function "collection"
:function "contains"
:function "contains-all-words"
:function "contains-any-words"
:function "contains-word"
:function "dateformat"
:function "dateparse"
:function "day"
:function "default"
:function "distinct"
:function "divide"
:function "ends-with"
:function "equals"
:function "error"
:function "escape-uri"
:function "gt"
:function "gte"
:function "if"
:function "isdate"
:function "join"
:function "length"
:function "lowercase"
:function "lt"
:function "lte"
:function "matches"
:function "max"
:function "min"
:function "month"
:function "multiply"
:function "normalize-space"
:function "not"
:function "notequals"
:function "now"
:function "null"
:function "number"
:function "or"
:function "replace"
:function "service"
:function "sort"
:function "starts-with"
:function "string"
:function "substring"
:function "subtract"
:function "uppercase"
:function "weekday"
:function "words"
:function "xattribute"
:function "xattributes"
:function "xcount"
:function "xelement"
:function "xelements"
:function "xexists"
:function "xnode"
:function "xnodes"
:function "xvalue"
:function "xvalues"
:function "year"
:function -usage "faql" "xvalue"
:function -usage "faql" "xvalues"
:function -usage "csv" "cellvalue"
```

Expressions ..

year(xvalue('patient/visitdate')) - year(xvalue('patient/birth'))

xvalue('image/cloud-cover/percentage') <= 10

xvalue('image/cloud-cover/percentage') > 10

xvalue('person/height') / xvalue('person/weight') > 35

if (...)

Functions:

- Can be complex
- Can have conditionals

```

asset.import.metadata.xml :namespace "convicts" \
:batch-size 1000 \
:apath "/ccc" \
:mpath -as "convicts:ccc" ". " \
:transform < \
    :map -document "convicts:ccc" -xpath "/event/@when" \
        "default(xvalue('/event/@when'),'UNKNOWN')" \
    :map -document "convicts:ccc" -xpath "/event/deathDate" \
        "dateformat( dateparse( xvalue('/event/deathDate') , 'dd MMM, yyyy', 'ddMMM, \
yyyy', 'MMM, yyyy', ' MMM, yyyy', 'MMM,yyyy', 'yyyy', ' , yyyy', ' , yyyy', ' , yyyy', '\\Unknown/missing\\', yyyy', \
'\\ Unknown/missing\\', yyyy', 'M, yyyy', 'M , yyyy') , 'dd-MMM-yyyy')" \
    :map -document "convicts:ccc" -xpath "/event/birthDate" \
        "dateformat( dateparse( xvalue('/event/birthDate') , 'dd MMM, yyyy', 'ddMMM, \
yyyy', 'MMM, yyyy', ' MMM, yyyy', 'MMM,yyyy', 'yyyy', ' , yyyy', ' , yyyy', ' , yyyy', '\\Unknown/missing\\', yyyy', \
'\\ Unknown/missing\\', yyyy', 'M, yyyy', 'M , yyyy') , 'dd-MMM-yyyy')" \
    :map -document "convicts:ccc" -xpath "/event/baptismDate" \
        "dateformat( dateparse( xvalue('/event/baptismDate') , 'dd MMM, \
yyyy', 'ddMMM, yyyy', 'MMM, yyyy', ' MMM, yyyy', 'MMM,yyyy', 'yyyy', ' , yyyy', ' , yyyy', ' , yyyy', '\\Unknown/ \
missing\\', yyyy', '\\ Unknown/missing\\', yyyy', 'M, yyyy', 'M , yyyy') , 'dd-MMM-yyyy')" \
    :map -document "convicts:ccc" -xpath "/event/deathYear" \
        "dateformat( dateparse( xvalue('/event/deathYear') , 'dd MMM, yyyy', 'ddMMM, \
yyyy', 'MMM, yyyy', ' MMM, yyyy', 'MMM,yyyy', 'yyyy', ' , yyyy', ' , yyyy', ' , yyyy', '\\Unknown/missing\\', yyyy', \
'\\ Unknown/missing\\', yyyy', 'M, yyyy', 'M , yyyy') , 'dd-MMM-yyyy')" \
    :map -document "convicts:ccc" -xpath "/event/birthYear" \
        "dateformat( dateparse( xvalue('/event/birthYear') , 'dd MMM, yyyy', 'ddMMM, \
yyyy', 'MMM, yyyy', ' MMM, yyyy', 'MMM,yyyy', 'yyyy', ' , yyyy', ' , yyyy', ' , yyyy', '\\Unknown/missing\\', yyyy', \
'\\ Unknown/missing\\', yyyy', 'M, yyyy', 'M , yyyy') , 'dd-MMM-yyyy')" \
> \
:in file:/xml-to-doc/raw_ccc/raw_ccc_standard.xml

```

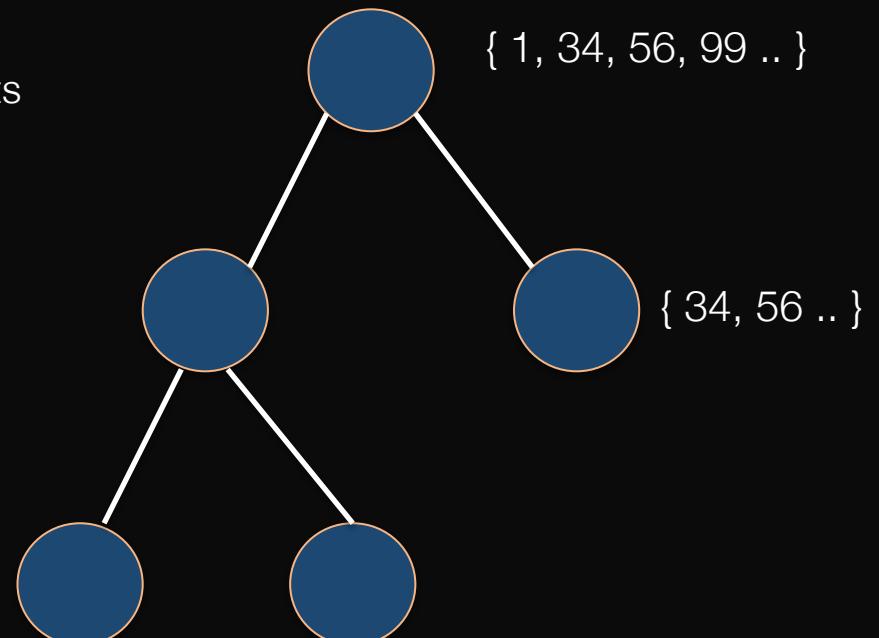
Transformations ..

- Metadata:
 - Virtual “side-car” XML in arbitrary format, defined by mount rules. E.g.
 - ISO 19115
 - JP 2064
 - Dublin Core
 - RIF-CS
 -
- Data:
 - Specifying the formats for data, requiring dynamic transformation (and caching)
E.g.:
 - JPEG-2000
 - NITF

Performance ..

- The **lookup** performance is a function of the dynamic computation:
 - Standard asset query, or
 - Multi-level, function indexes
 - Dynamically computed and cached functional sets
- ... and your authorization levels.

Thereafter, all i-node level operations.



Where/when to use?

- Dynamic file systems excellent for:
 - Controlled publication of managed data
 - Pipelining workflows for applications that can only use POSIX
 - Multiple presentations of the same data (possibly as required by different applications)
 - Interchange

The power of metadata!

Mediaflux™

OPERATING SYSTEM FOR META+DATA

www.arcitecta.com