Middleware development notes

- Result set/accumulator improvements
- Collection hierarchies
- Collection monitoring
- Ingest/management services
- Vocabulary refinements

Result sets and accumulators

- Result accumulator
  - now: craps out on first error
  - new: runs until all threads finish or time out

- Result set
  - tracks middleware and per-collection errors
  - maintains per-collection counts
  - still merges & ranks results
  - maintains mapping between results & collections?
  - interaction with collection hierarchies?

Collection hierarchies

- The problem:
  - flat namespace
  - people want hierarchies
    - aggregation
    - logical grouping
    - grouping & easy importing of remote collections

Kinds of collections

- From implementation perspective: two kinds of collections
  - leaf collections
    - "real," have items
  - non-leaf collections
    - pure aggregations
    - may or may not expose sub-collections
    - collection-level metadata
      - static: custom or auto-generated
      - dynamic: auto-generated
    - independent of collection location
      - proxies act like symlinks or NFS mounts
      - forms a logical namespace of distributed content
- From client perspective: all collections look alike*

Exposing the hierarchy

- Configuration service
  - reports hierarchy, collection statuses
- Query results
- Aggregates with hidden contents act like leaf collections
- Result sets?

Identifiers

- Now:
  - `collection:local-id`
- New:
  - `collection/path/...:local-id`
  - `collection/path/...:hidden/path/...:local-id`
- Someday: GUID prefixes
- Implementation note: identifier renamers must be uniformly employed

Collection monitoring

- New collection-level service: status
  - possible answers: up, down, partially up
  - aggregates return info on sub-collections
  - bucket99 driver: executes test query

- Separate thread polls collections, maintains global map

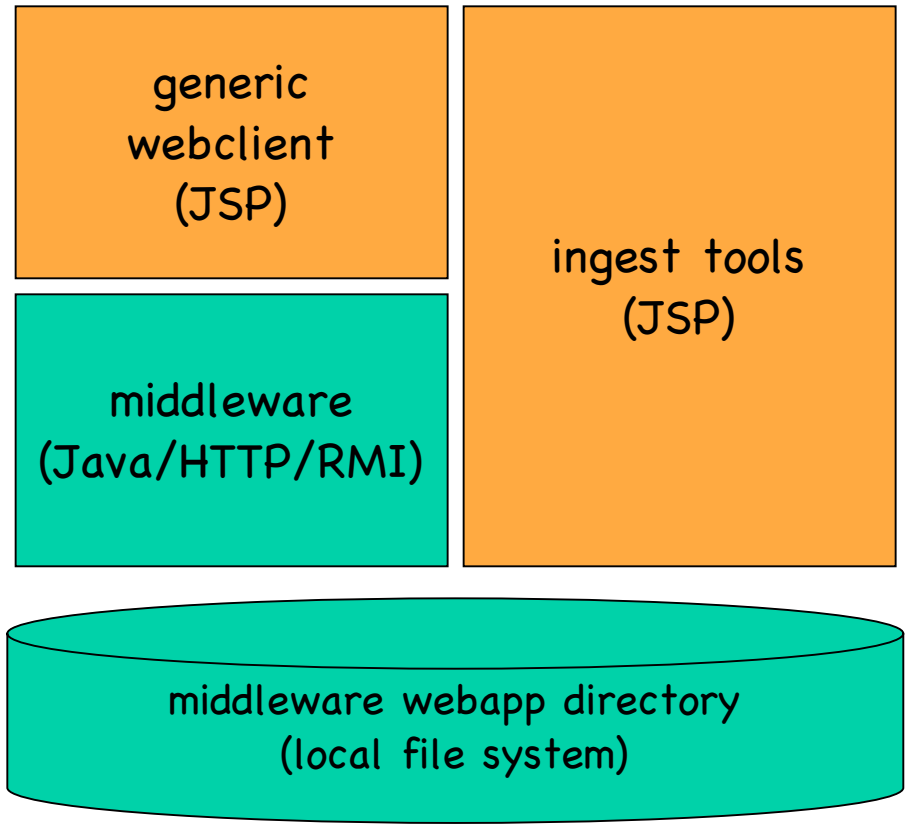
Ingest/management services

- Add item
  - 3 ADL views plus other, arbitrary views
  - returns collection-local identifier
- Replace view
- Delete item

- Create collection
  - name, static metadata
- Replace collection static metadata
- Delete collection

- New: these are "lc" (local collection) specific

Ingest big picture



Vocabulary refinements

- Definition: a thesaurus *C* is a refinement of a thesaurus *P* if it has the following properties:
  - every preferred term in *C* is (transitively) narrower than one or more preferred terms in *P*
  - every non-preferred term in *C* has a *use-instead* relationship to one or more preferred terms in *P* and/or in *C*
- Thesaurus protocol must support cross-thesaurus references
- Hierarchical bucket is still associated with a root thesaurus
  - refinements support per-collection customization

adl:types

- Base: DCMI Type Vocabulary
  - http://dublincore.org/documents/dcmi-type-vocabulary/

- MIL extension:
  - DCMI:Dataset
    - digital elevation model
  - DCMI:Image
    - map
    - remote-sensing imagery
      - satellite image
      - aerial photograph

adl:formats

- Offline
  - digital
  - physical
- Online
  - text
  - image
  - video
  - audio
  - application