

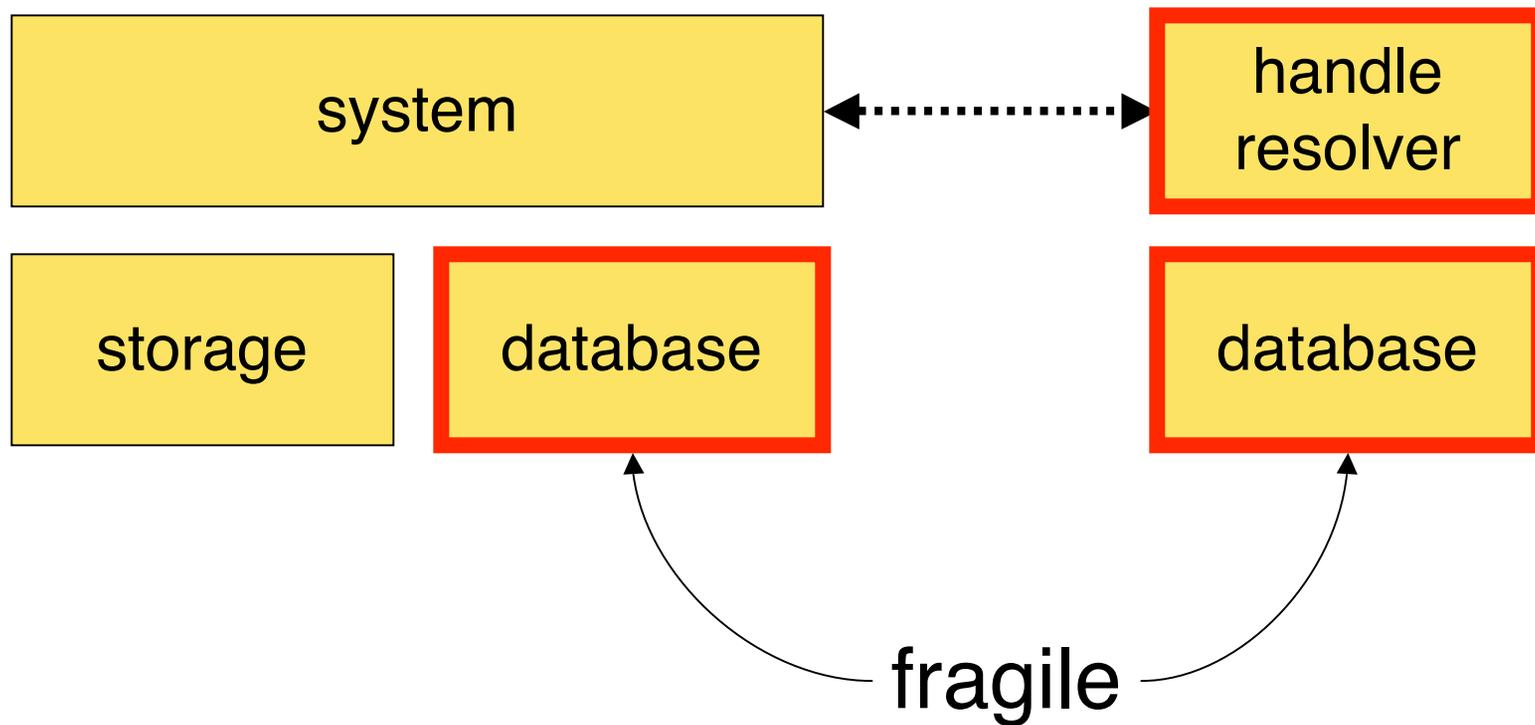
# NGDA Architecture Update

Greg Janée

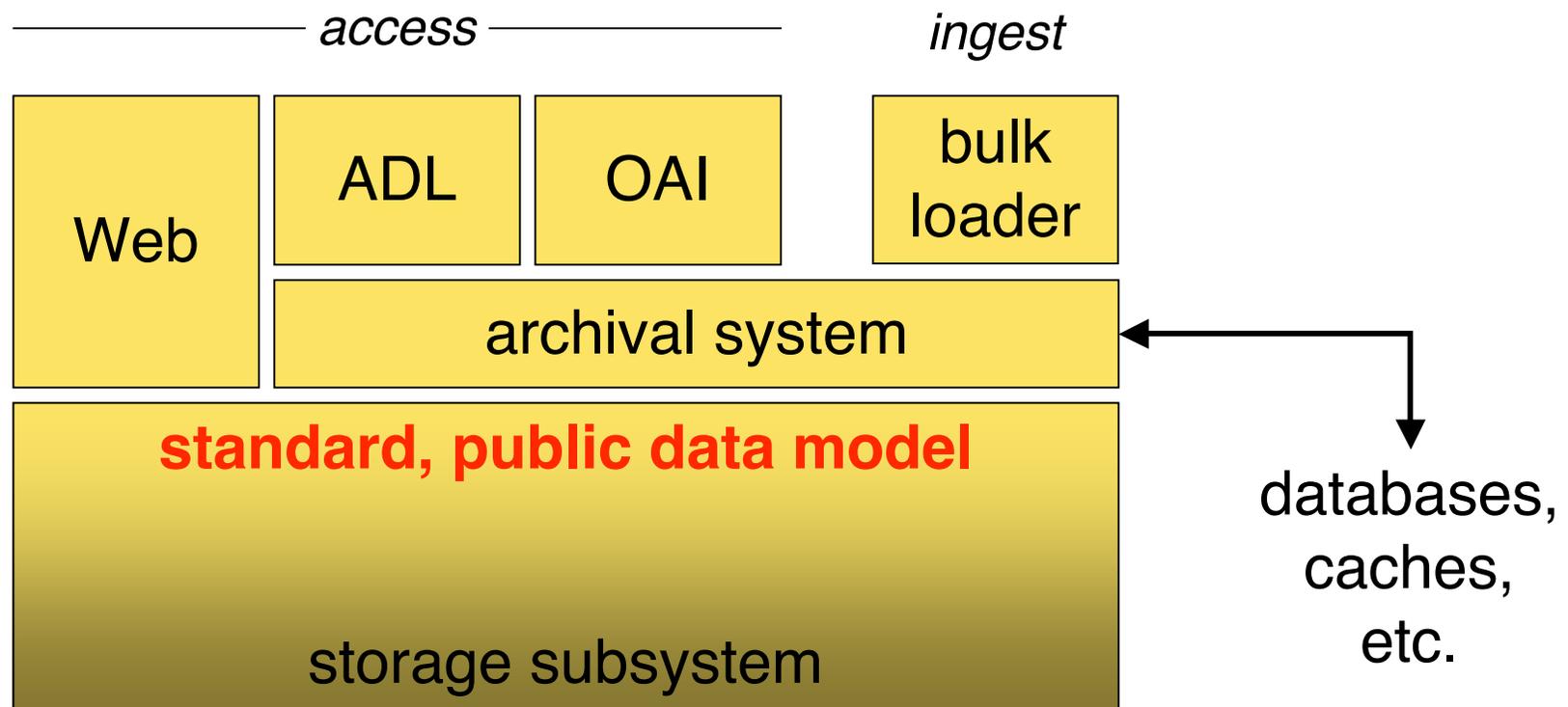
# Three motivations

- Archival has to be cheap & easy
  - little incentive
  - no funding
- Need to archive data semantics
  - key differentiator from text, audio, video
- Focus on *long-term* preservation
  - need to migrate whole systems

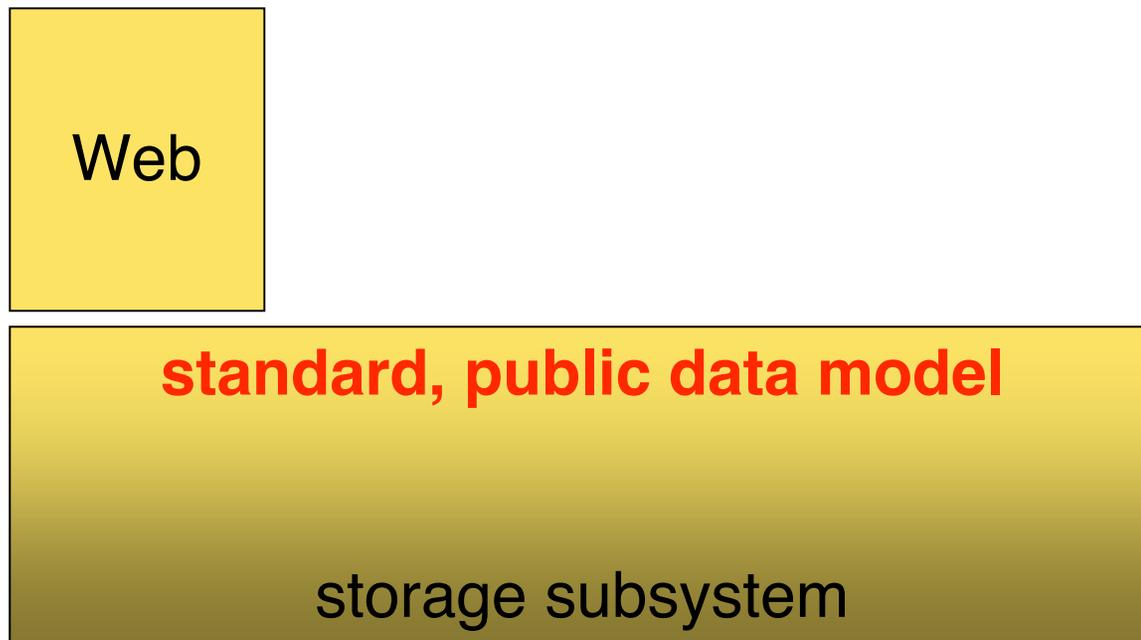
# Typical repository architecture



# NGDA architecture



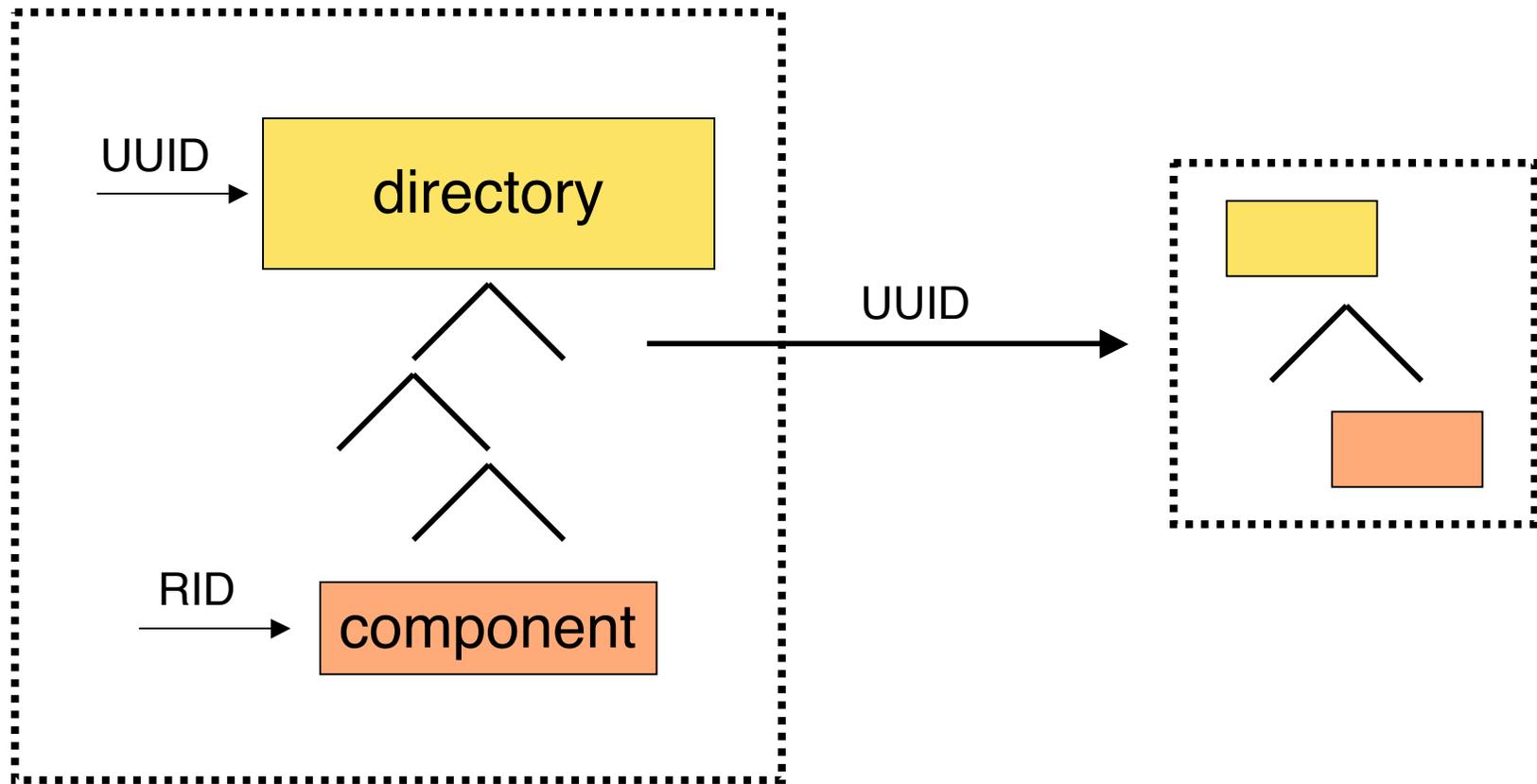
# Post-NGDA architecture



# Storage system requirements

- Req's:
  - associate UUIDs/RIDs with bitstreams
  - retrieve global/local bitstream by UUID/RID
  - determine (parent) UUID of any bitstream
  - list all UUIDs
- Satisfied by:
  - any filesystem
  - tag URIs for UUIDs
    - tag:library.ucsb.edu,2005:*identifier*

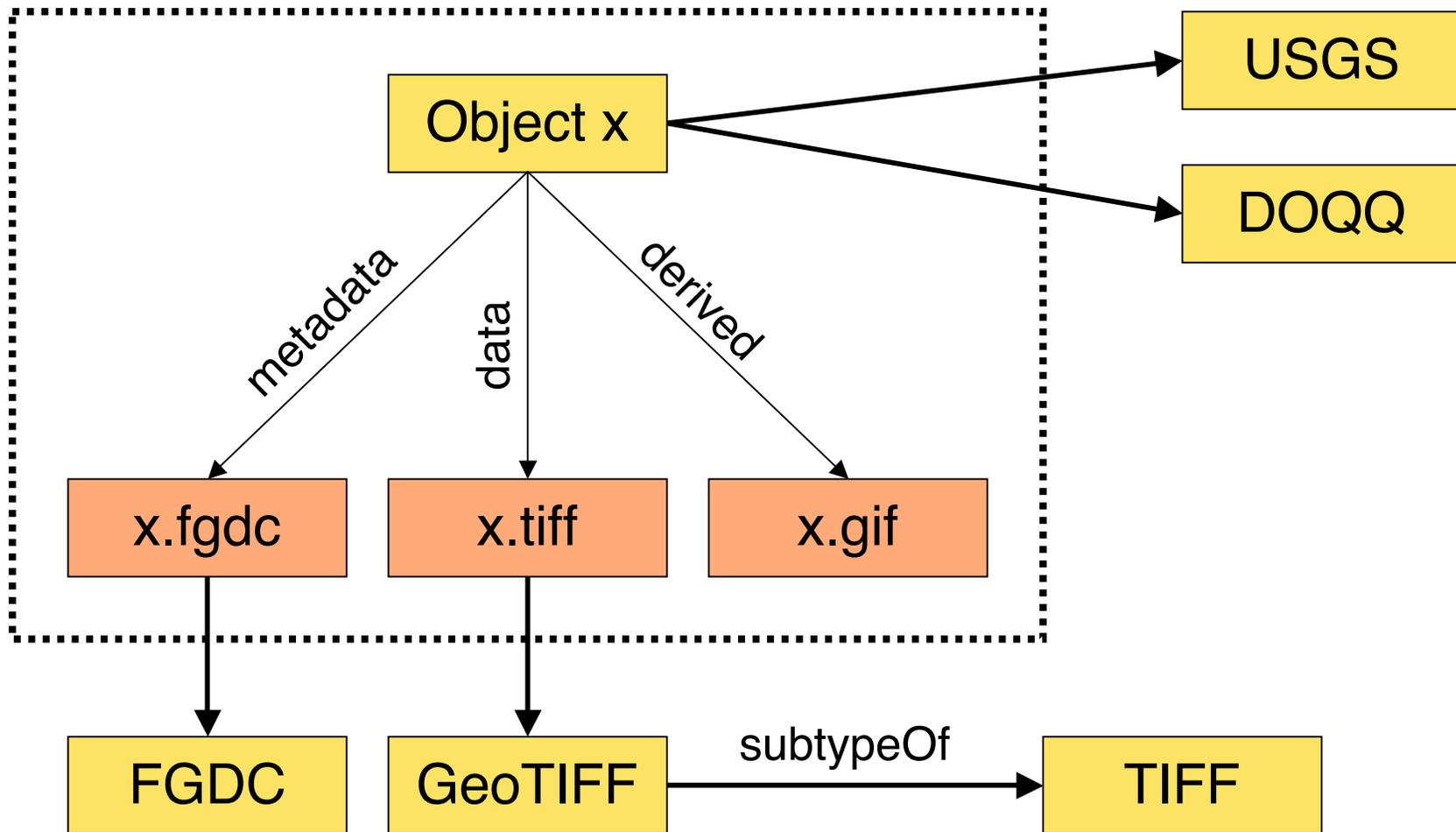
# Archival objects



# Archival objects

- Directory info per component
  - named relationship/position
  - format & semantics
    - by UUID references to definitions
  - fixity: checksum
  - provenance: isDerivative
  - policy: mutability
  - rights
- Components may be provided by archive itself

# Example



# Archives

- Archive = set of archival objects
  - no structure
  - no free-floating bitstreams
- In anticipation of federation:
  - associations may cross archive boundaries
  - archival objects may not

# Object types

- Content
- Format definition
- Semantic definition
- Provider
- Organizational structures
  - collection
  - series
  - ingest session

# Archive-provider agreement

- Defines
  - common structure of objects to be ingested
  - necessary validations
  - associations to other objects
  - policies, rights, etc.
- Represents choke point
  - requires human evaluation

# Deferred functionality

- Incremental ingest
- Object revisions
- Rights
- 3rd-party access
- Federation

# Status

- Starting development now
- Approach: iterative refinement