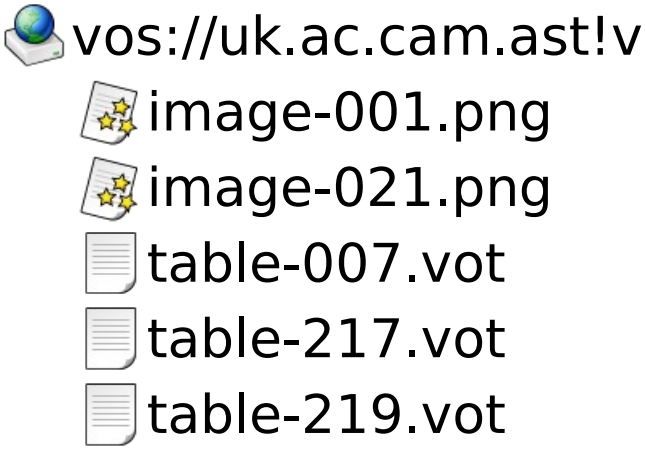
# Implementing VOSpace 1.x

Dave Morris
AstroGrid,
IoA Cambridge

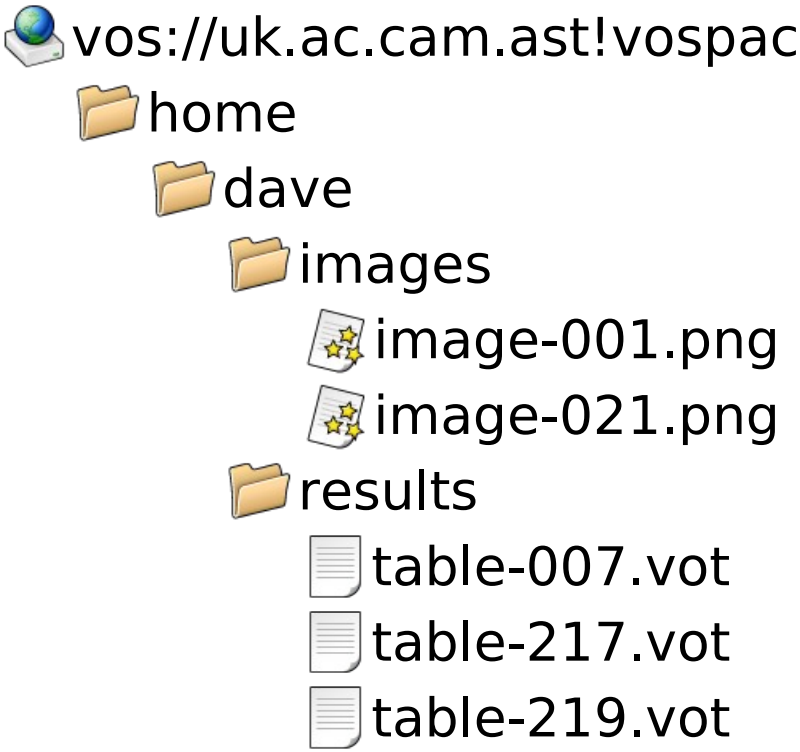
# IVOA VOSpace 1.0




- Support for more than one protocol
- Support for more than one view
- Asynchronous 3rd party transfers

- Published as IVOA working draft

# IVOA VOSpace 1.1 Things to add ....




- Containers
- Hierarchical space within each service

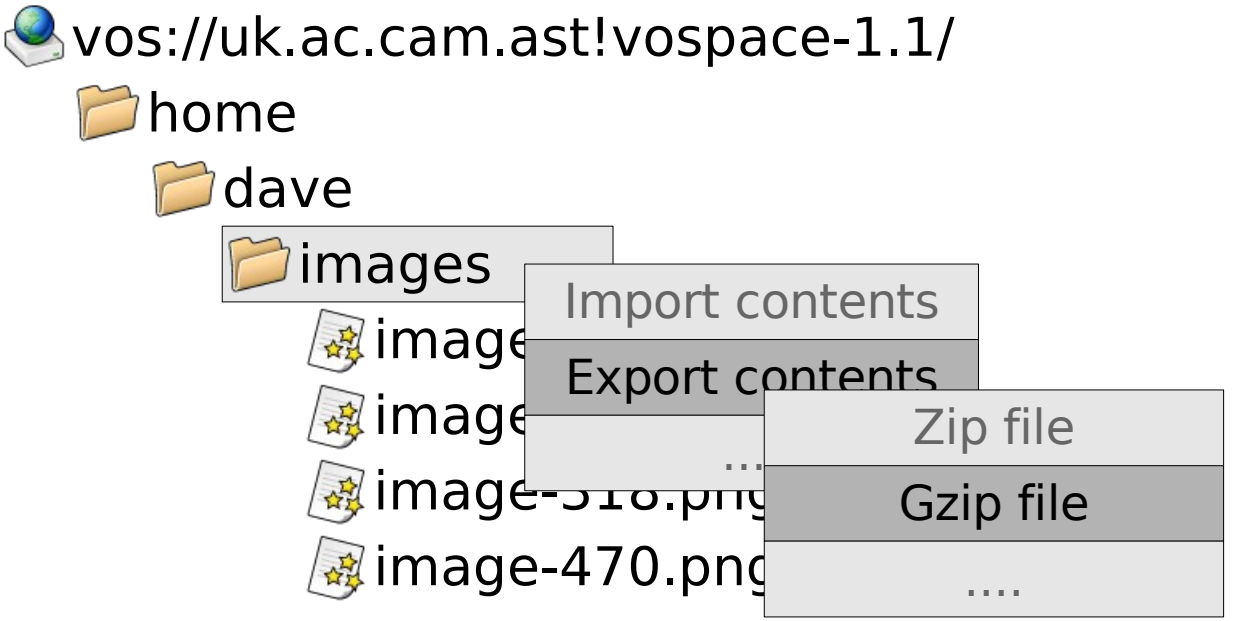
# IVOA VOSpace 1.1 Things to add ....


- Containers
- General agreement on behaviour

  - Still to do
  - Details of capabilities and metadata

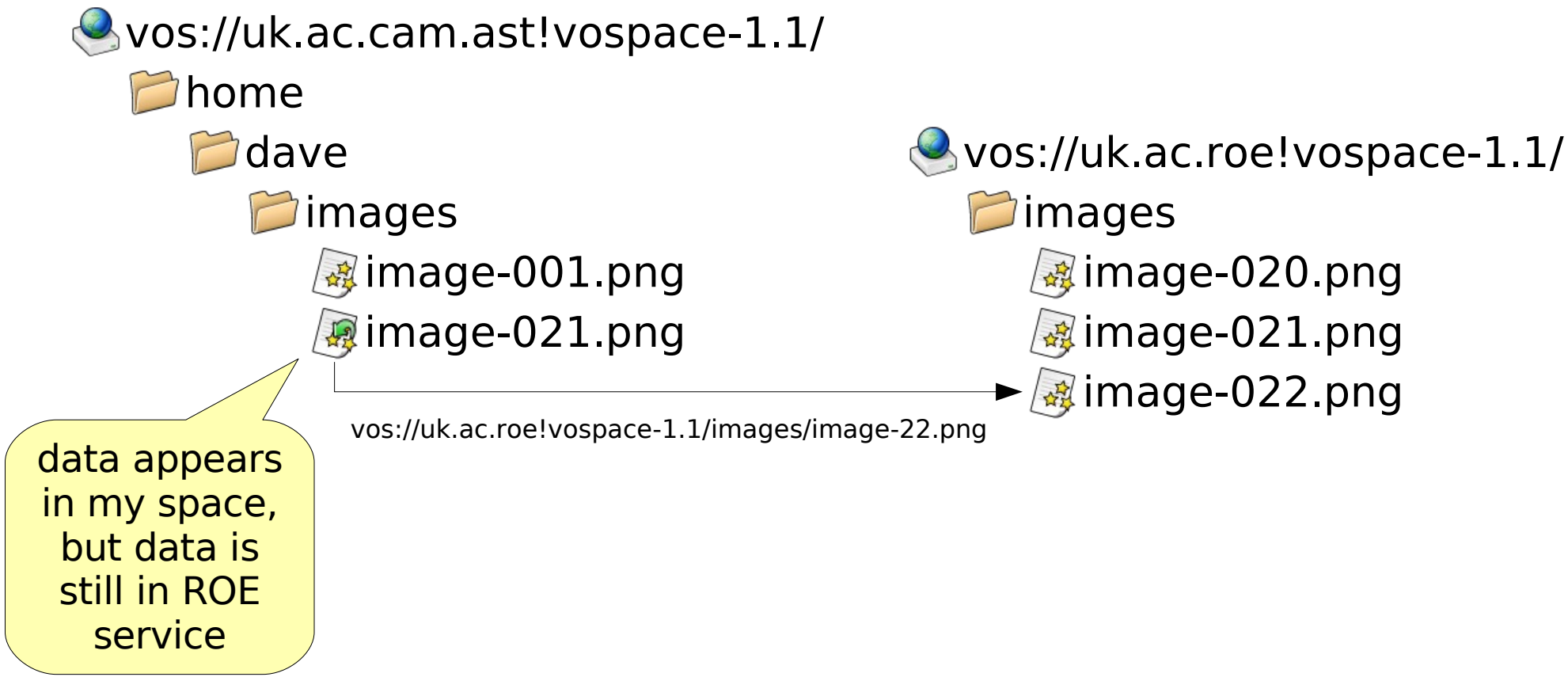
    *“This container accepts the following data types”*

# IVOA VOSpace 1.1 Things to add ....



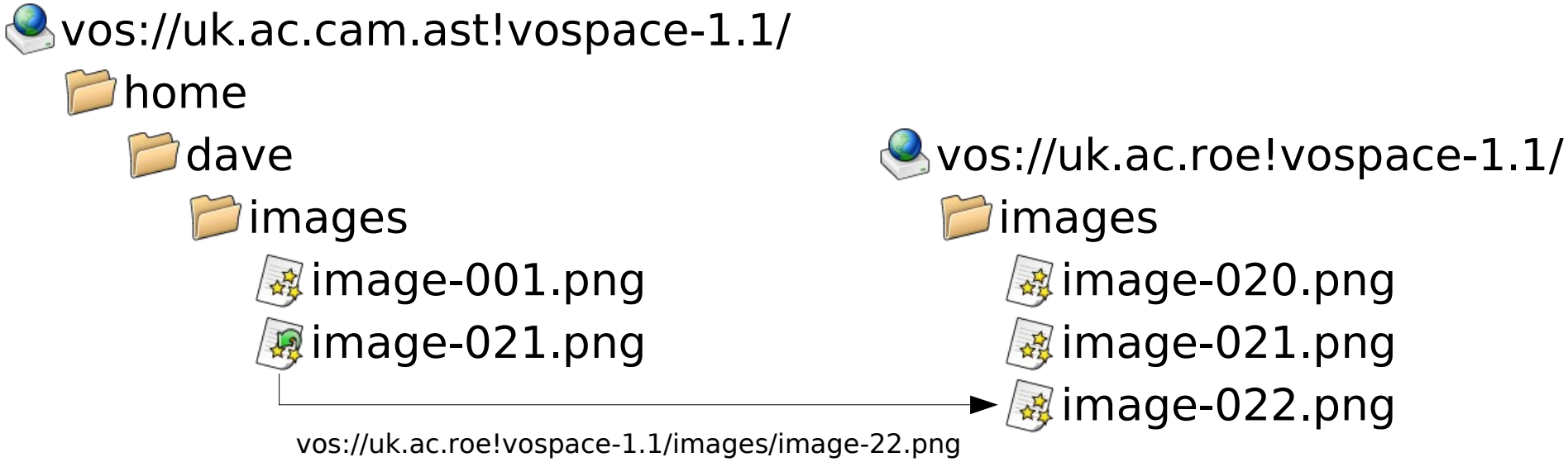

- Container transfers
- Transfer contents as a single object

# IVOA VOSpace 1.1 Things to add ....




- Inter space links
- Links to objects in other services

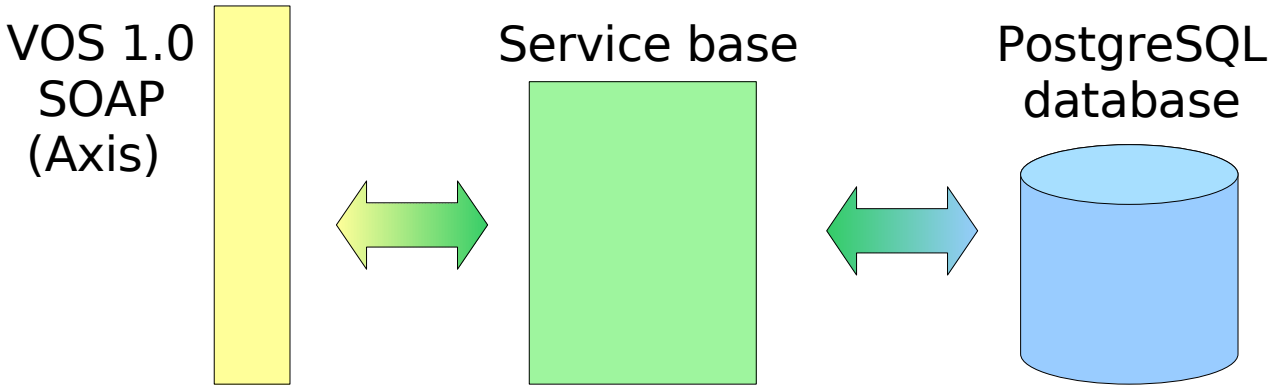
# IVOA VOSpace 1.1 Things to add ....




- Inter space links
- General agreement on behaviour

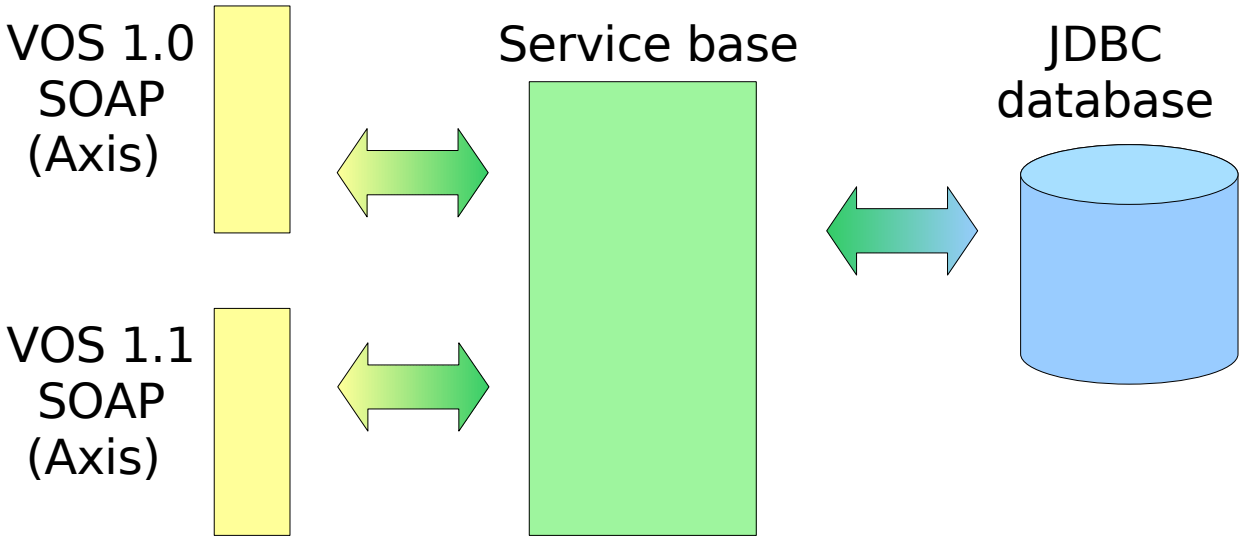
  - Still to do
  - Final details of schema and exceptions

# AstroGrid VOSpace 1.x  Implementation ....



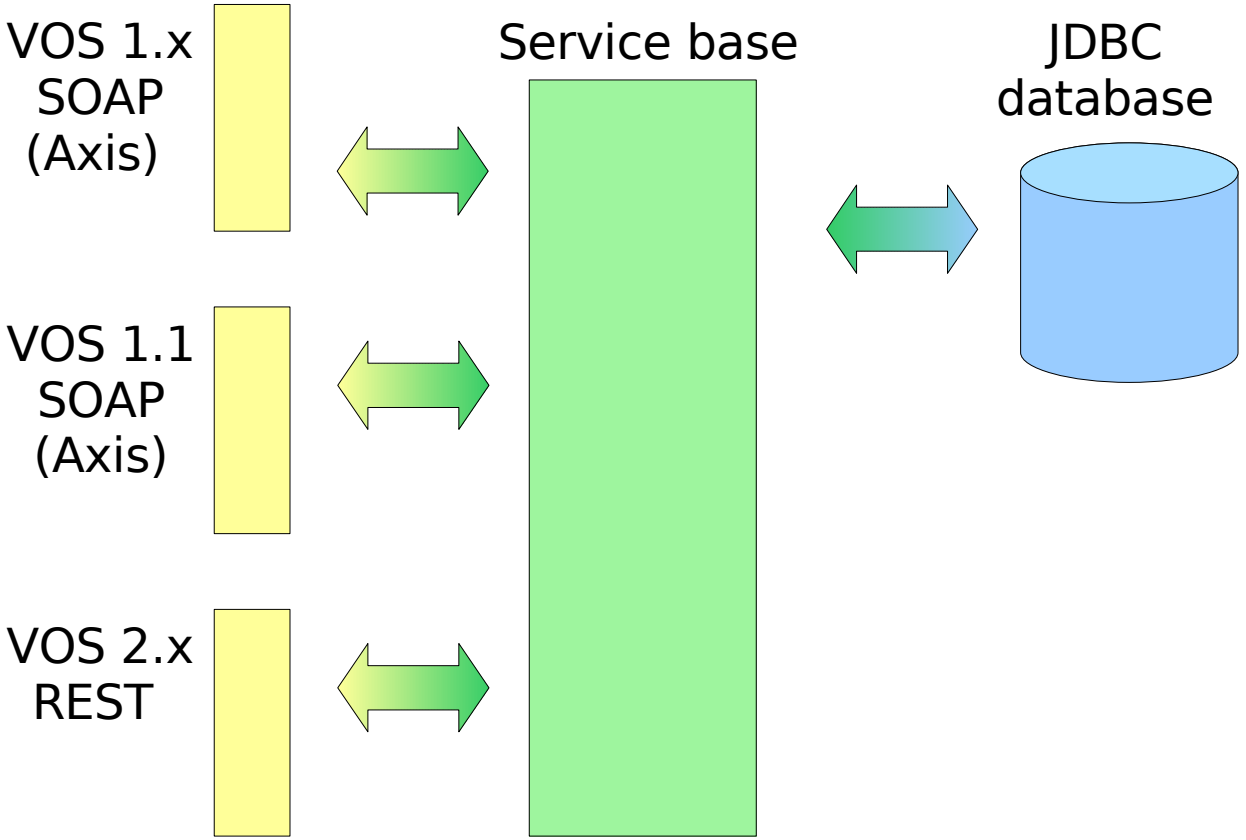

- Core service base
- Stores node metadata in RDMBS
  - Current implementation uses PostgreSQL
  - Capable of using any JDBC compliant RDBMS

# AstroGrid VOSpace 1.x  Implementation ....



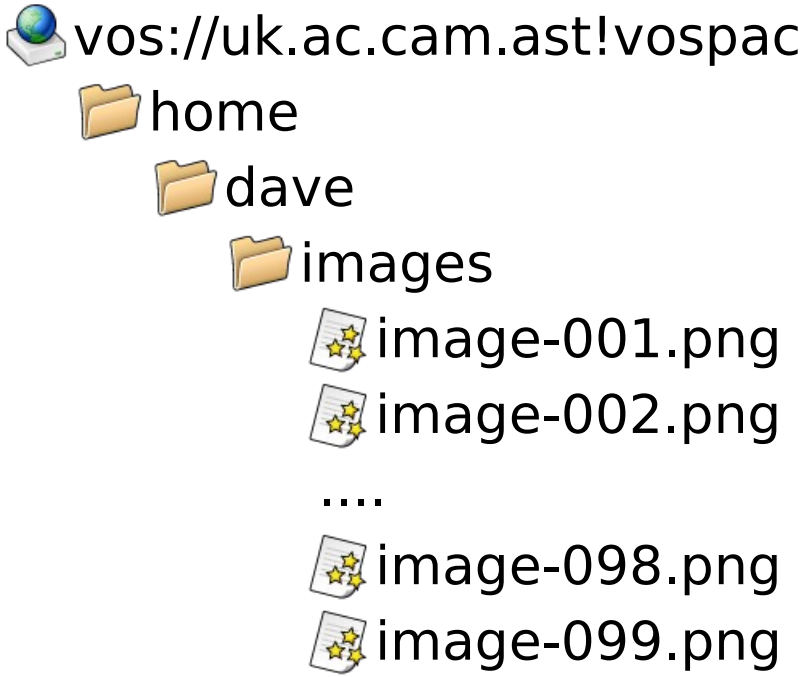

- Support for more than one version
  - VOS-1.0
  - VOS-1.1

# AstroGrid VOSpace 1.x  Implementation ....




- Support for more than one interface
  - VOS-1.x SOAP services
  - VOS-2.x REST services

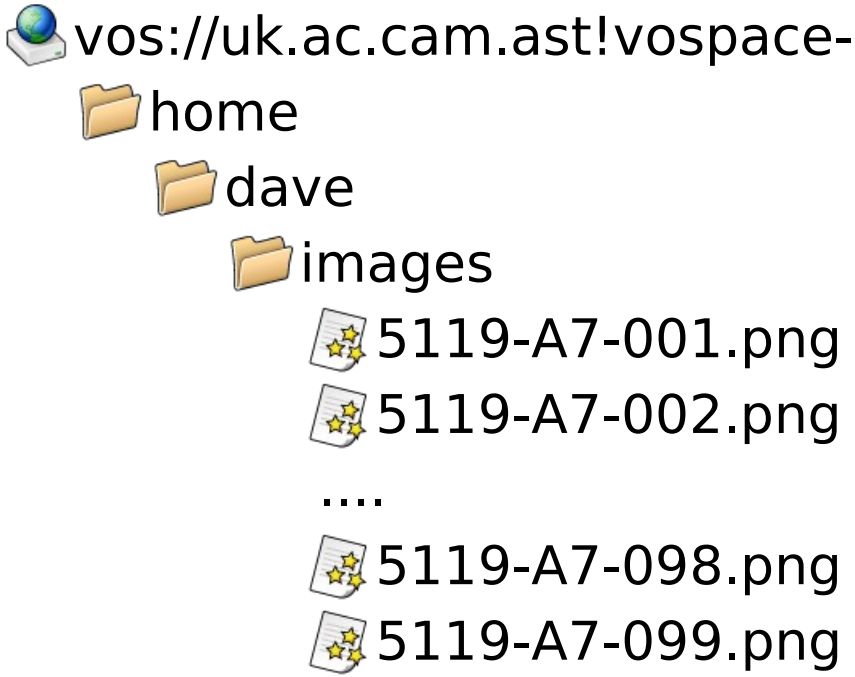
# IVOA VOSpace 1.1 Things to change ....




- ListNodes
  - Request 1st 'page'
  - Response contains 'token'
  - Request next 'page' using token
- Tricky to implement server side
- UI developers don't need it
- Not compatible with REST

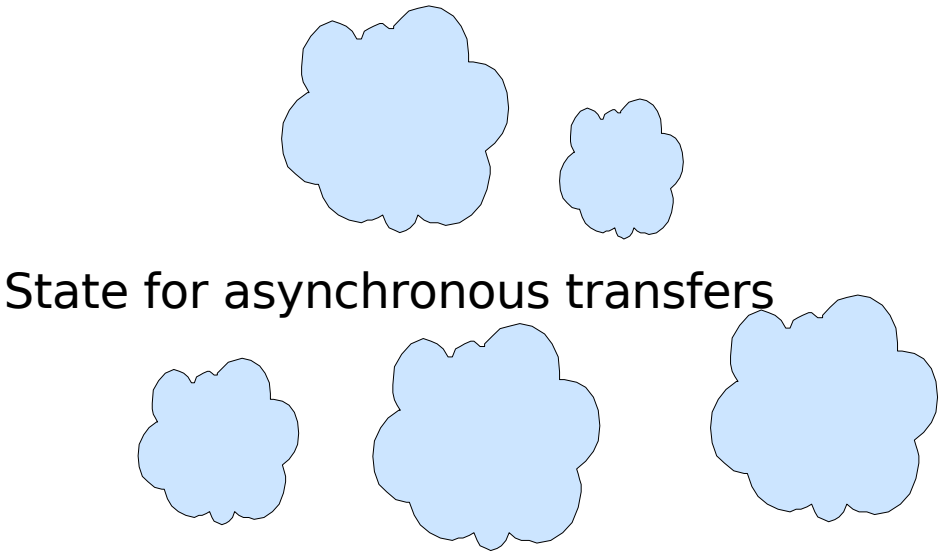
- REST style would be simpler
- List everything in one response

# IVOA VOSpace 1.1 Things to change ....




- Generated names
  - vos://null
  - Not clear to users
  - Won't work with containers
- Reserved name .auto
- Applied to any path
  - vos://service/path/.auto
  - vos://service/path/5119-A7-100

# IVOA VOSpace 1.x Things to think about ....




How do we represent the state of a transfer

- As a VOS Node with properties ?
- As a UWS object ?

Initiate a transfer, response contains representation of state

- Transfer has state
- Transfer has nested state for each protocol

# IVOA VOSpace 1.x Things to think about ....

Naming of protocol and view params

- Current form uses 'string' names
- Causes problems representing state of a transfer as a Node

- Current schema makes it difficult to identify a parameter using a URI.

- Can we refactor these to look like properties ?

  Re-use existing schema elements rather than define new ones

  Protocol param == Node property

- Or ... do we actually need protocol and view params ?

# IVOA VOSpace 1.x Summary

- VOSpace 1.1
  - Inter space links
    - General agreement on behaviour
    - Still to do
    - Final details of schema and exceptions
  - Containers
    - General agreement on behaviour
    - Still to do
    - Details of capabilities and metadata