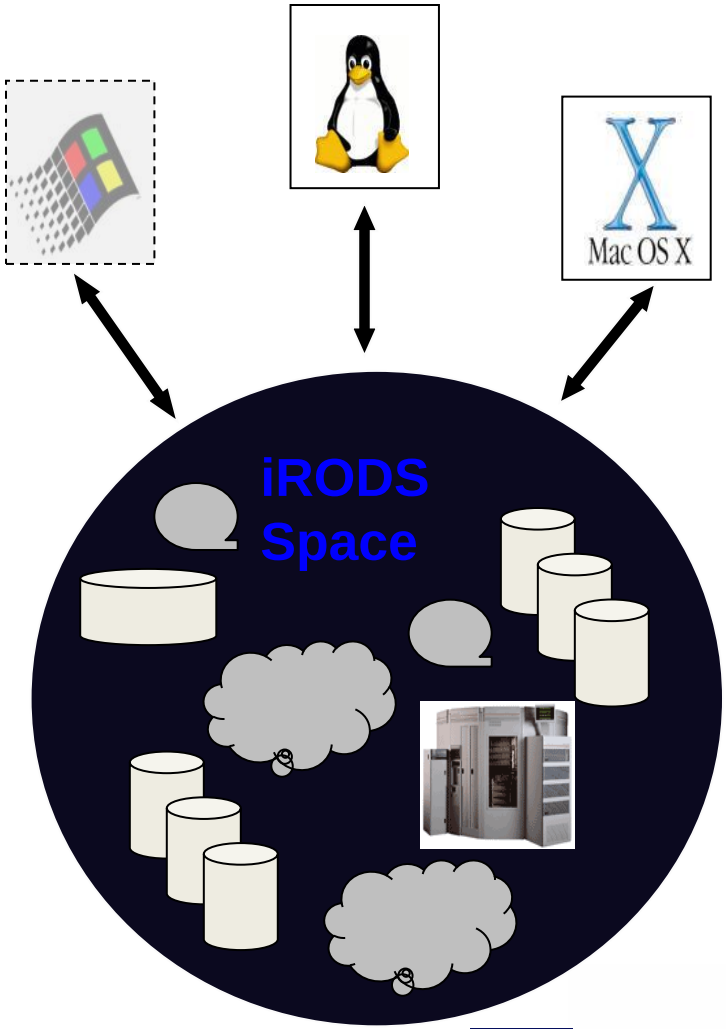
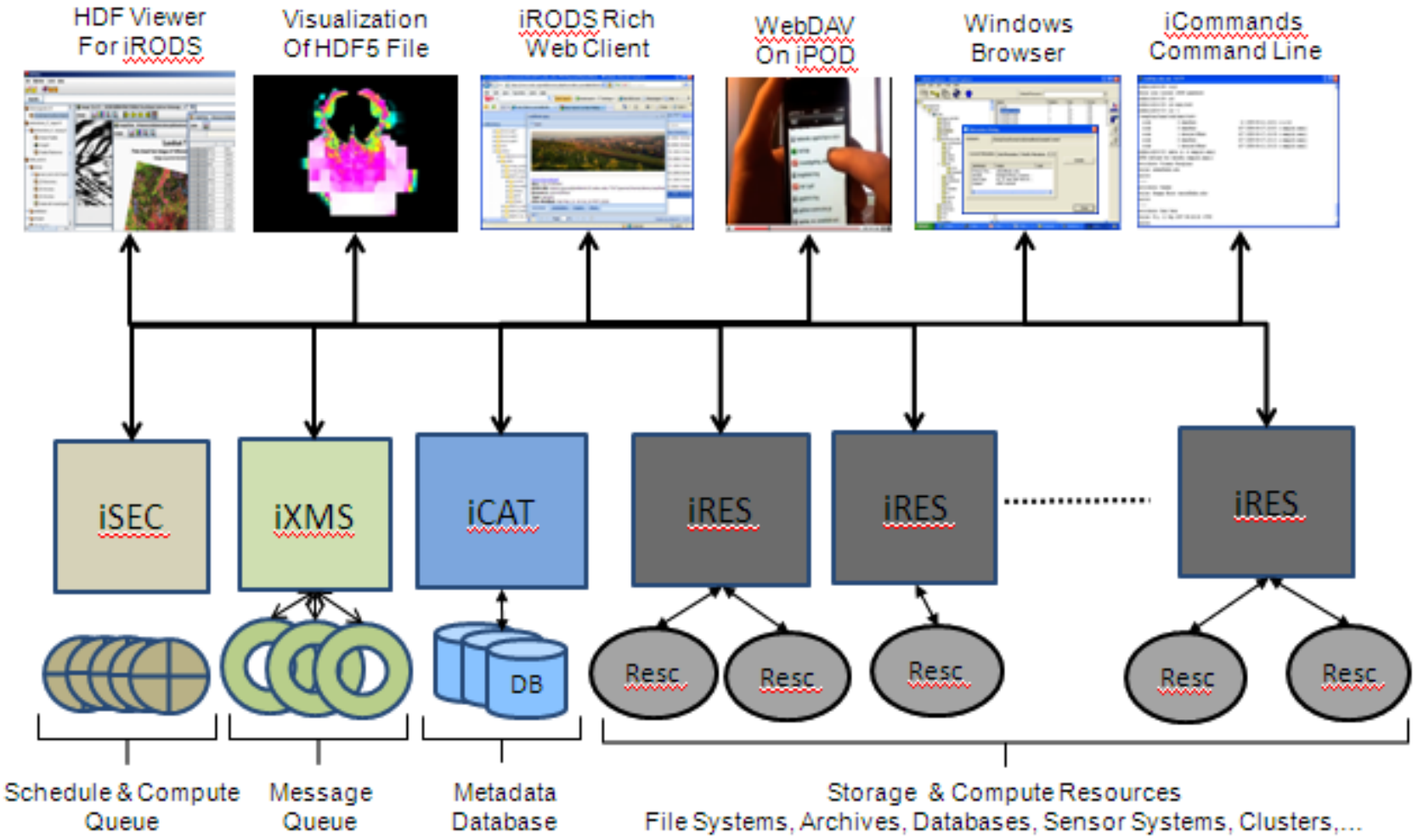
Micro-Service Objects (MSO)

Micro-Service Structured Objects (MSSO)



iRODS Distributed Data Management



D·I·C·E

i·R·O·D·S

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

UCSD

NSF

THE NATIONAL ARCHIVES ARCHIVES.GOV

renci

Data Served by iRODS

- Static Information through drivers
  - Files in storage systems
    - Traditional file systems: Disks, tapes, NFS
    - Cutting-edge Systems: Clouds (S3), Object Servers (WOS)
    - Backend uses POSIX APIs to access files
  - DBO – Database Objects
    - Tabular data from databases
    - Backend uses SQL through ODBC or native API

How to Add a New Access method in iRODS

- Through Storage Drivers
  - Add a driver file in server/drivers directory
    - 17 APIs need to be written – some can be NULL
  - Add information in iRODS core code
    - Some information in header files
    - Some information in iCAT
  - Add information in config.mk file
  - Add information in Makefile as needed
- Needs to change iRODS core
- Upgrades to new releasescan be troublesome
  - Unless integrated into release svn

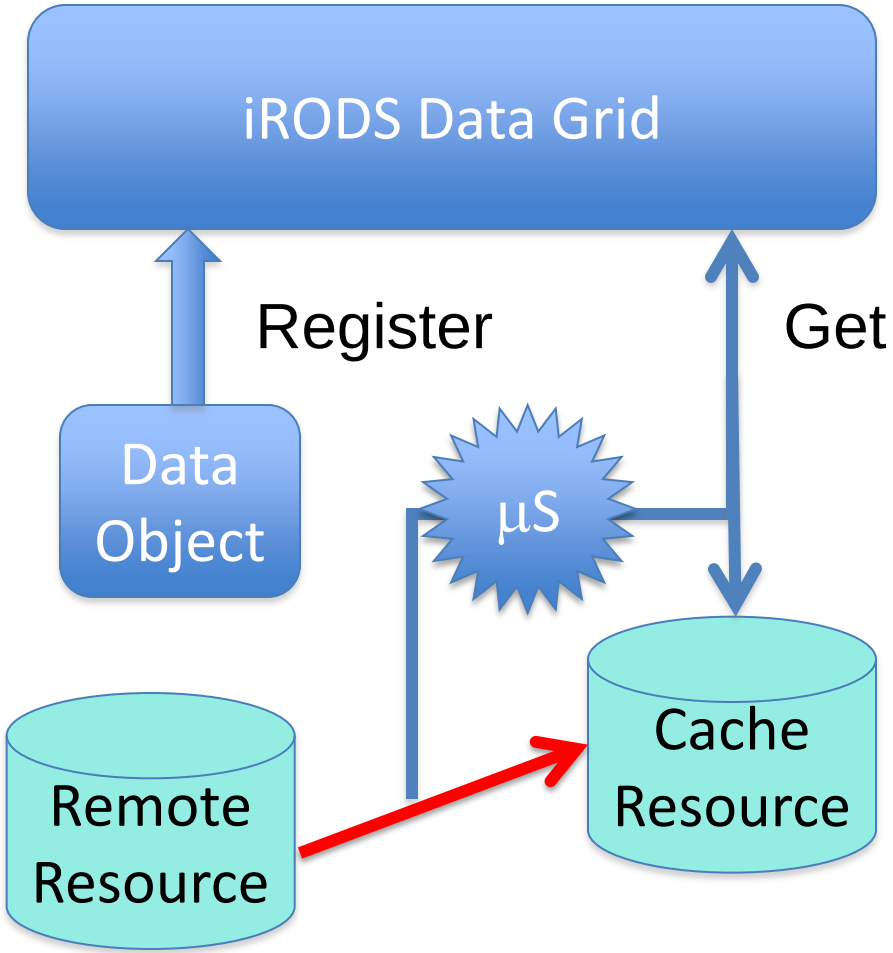
Another way to add a driver

- Universal Mass Storage Driver
- Implemented by Jean-Yves
- In Server/bin/commands directory
  - Executable script that
  - Create routines that can call executable commands to perform get/put/stat functions
- Easy to use.
  - No changes in core code – upgrades easy
- One nit-picky problem
  - Only one per universal resource per server

MSO

- Micro-Service Object
- Aim: Create an iRODS object
  - which can access an external resource
  - Just by writing a few micro-services
- Built on access methods already in iRODS
  - compound resource model
    - Already used by DDN's WOS and Amazon's S3
    - Not much dissimilar to Universal Storage of Jean-Yves.
- Uses micro-services in modules
  - Upgrades is easy
  - As many resource types as you want

Simple Access Example



Steps

1) Register "access information" for an object (ex. http url)
2) On "get", retrieve a copy from remote resource and cache in data grid and send to client
3) On subsequent "get", provide copy from cache If copy is deleted, go back to original site

MSO micro-services

- Three micro-services are needed to be implemented for each resource-type
  - Get, Put, Stat
  - No need to register each resource-type in iCAT
    - Requires creation of at least one mso resource and “mso resource group” for data grid
    - Resource group needed to identf cache resource
  - Micro-servcs created in “mso” module
    - In directory, ….
    - Signature of micro-service has to follow a template.
    - Add the micro-service signature (as normally done).

Sample “get” mso micro-service

How to use MSO

- Enable the mso micro-services:
  - Stop irods
  - Run gmake clean followed by gmake
  - Start irods
- Register an object
- Retrieve using any iRODS access mechanism
  - Iget, web access, JARGON, ….
- Available mso-types: http/s, ftp, z39.50, srb data grid, irods data grid (register a federated) objet

Getting a file from the web

ireg -D mso -R msoResc14 -G msoRescGroup "//http://irods.org/pubs/iRODS_Fact_Sheet-0907c.pdf" /tempZone/home/rods/webfile

Remote object is registered as webfile in the directory /tempZone/home/rods

Can retrieve a copy of the web page

iget webfile

Next Step - MSSO

- Micro-service Structured Object
- Aim: Can a “workflow” be identified as an iRODS object?
  - Executing the workflow should add the results of the workflow into iRODS
  - But workflows are complex (several steps)
  - They need parameters and input files
  - They may also produce more than one output file
  - Can one “iget” a workflow result?
  - Different runs may produce different set of results
- Concept of a “Realizable Object”
  - Different from “normal iRODS static objects
  - Elevates an executable to an “object” level
    - Brings in all iRODS-related management

Realizable Objects

- Register into iRODS the workflow needed to create a derived data product
- Accessing the link causes the derived data to be generated and registered as made accessible to the client
- Elevates provenance information to first class object
  - Workflow and parameter files (and if needed all executables) can be registered in iRODS
  - Provides repeatabilty and verifiability

MSSO – under development

- Available in 3.2
- Builds on proven iRODS concepts:
  - ‘structured files’ in iRODS
    - Tar, gzip, tgz, bundle, …
  - ‘mounted collections’ in iRODS
    - Mounting local directories, linking collections
- Inherits all iRODS bells & whistles
- Phase 1:workflows written in RODS rule language
- Phase 2: Integrate it to run iRODS Workflow Orchestration Server
- Phase 3: Elevate it to run workflows in any executable language

iRODS - Open Source Software

http://irods.diceresearch.org

NSF OCI-0940841 "DataNet Federation Consortium"
NSF OCI-1032732 "SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Application"
NSF OCI-0848296 "NARA Transcontinental Persistent Archives Prototype"
NSF SDCI-0721400 "Data Grids for Community Driven Applications"

D·I·C·E i·R·O·D·S THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL UCSD NSF THE NATIONAL ARCHIVES ARCHIVES.GOV renci