Policy Based Data Management iRODS

Reagan W. Moore   (DICE-UNC)

Arcot Rajasekar   (DICE-UNC)

Mike Wan   (DICE-UCSD)

Wayne Schroeder   (DICE-UCSD)

Mike Conway   (DICE-UNC)

Antoine de Torcy   (DICE-UNC)

Hao Xu   (UNC)

Sheau-Yen Chen   (DICE-UCSD)

Jason Coposky   (RENCI)

Lisa Stillwell   (RENCI)

Leesa Brieger   (RENCI)
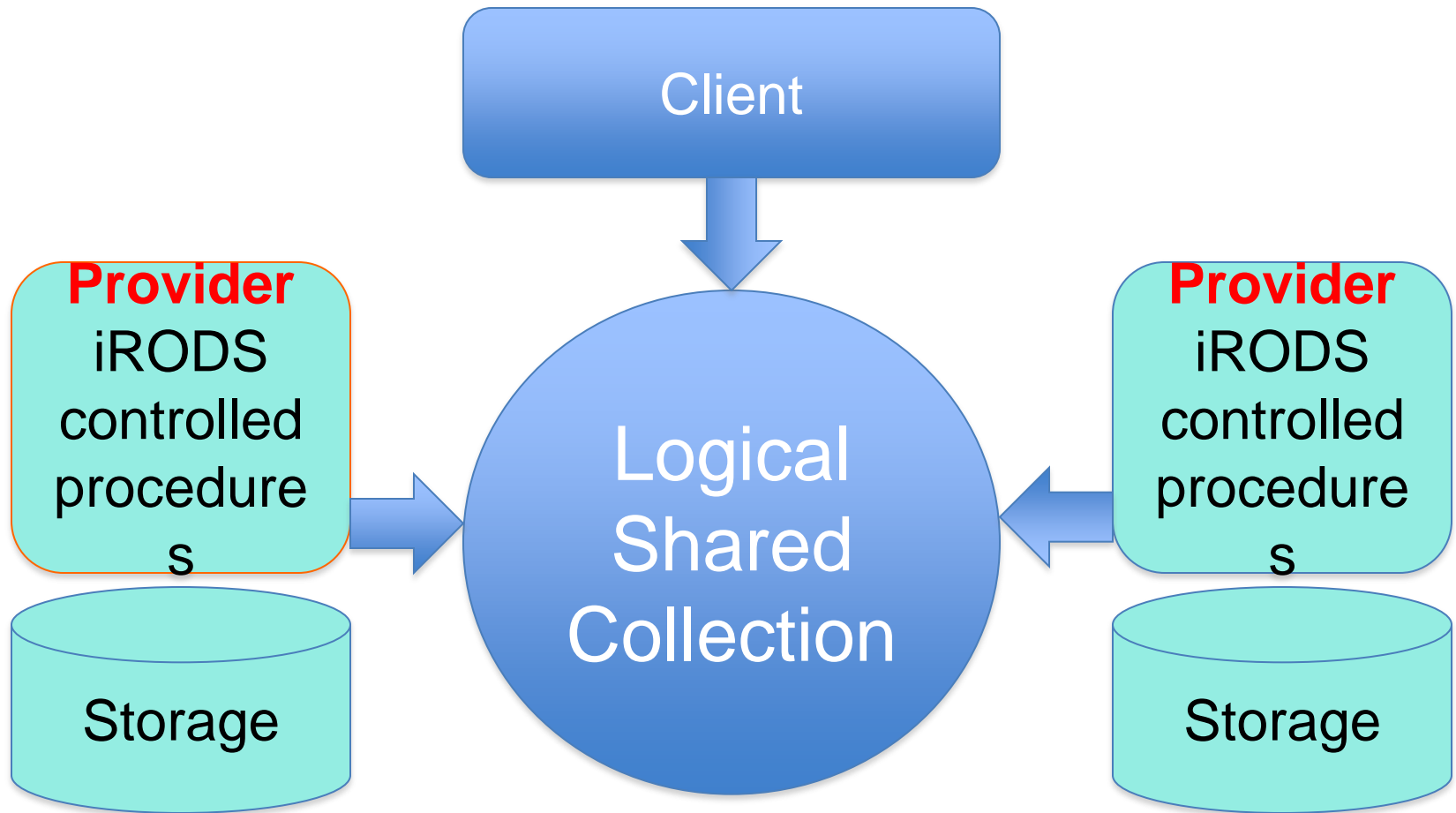
http://irods.diceresearch.org

Agenda

- 09:00 Introduction to iRODS technology – (Moore, UNC-CH)
- 10:00 Open discussion on data management needs – (Moore, UNC-CH)
- 10:30 Break
- 11:00 Installation of iRODS clients (Windows, Mac, Linux) – (Moore UNC-CH)
- 11:45 Initial demonstration of iRODS client access – (Moore, UNC-CH)
- 12:00 Lunch
- 14:00 Introduction to iRODS rules and micro-services – (Moore, UNC-CH)
- 14:45 Demonstration of federation of data grids - (Yutaka Kawai, KEK, Japan, Moore UNC-CH)
- 15:00 Simple rule examples including database queries (Moore, UNC-CH)
- 15:30 Break
- 16:00 Complex rule examples including scheduling (Moore, UNC-CH)
- 16:30 SRM development status– (Wei-Long UENG, ASGC)
- 17:00 Open discussion of iRODS applications in digital libraries, data grids, and preservation environments – (Moore, UNC-CH)
- 17:30 End of tutorial

Policy-based Data Sharing

Client

Provider
iRODS controlled procedures

Storage

Logical Shared Collection

Provider
iRODS controlled procedures

Storage

Consensus on Policies and Procedures controls the shared data

Applications

- International projects
  - Cyber Square Kilometer Array (radio astronomy), Cinegrid (movies)
- National data grids
  - Australia, New Zealand, Portugal, UK, France
- Federal agency archives
  - NASA Center for Climate Simulation, National Optical Astronomy Observatories
- Institutional repositories
  - French National Library, Carolina digital repository, Broad Institute genomics data, Sanger Institute

iRODS Version 3.0

- Released Sept. 30, 2011
- Next release scheduled for March, 2012
- New features
  - New rule engine
    - Strong parameter typing
    - Optimized performance (thousands of rules)
    - Expanded rule programming language
  - Rule versioning
  - Distributed rule base management
  - Soft links to external resources
  - New transport management for large files
  - Improved Java interface
  - Windows native port (C++ compilation)
  - Dropbox interface (iDrop)

Under Development

- Tickets (3.1)
  - Ability to give access to specified file for specified time duration

- netCDF library support (3.1)

- Pluggable Authentication Modules (3.2)
  - Use external identity resource

- Registration of workflows (3.2)
  - Micro-service structured objects
  - Re-execute workflows for derived data products
  - Policy-encoded objects
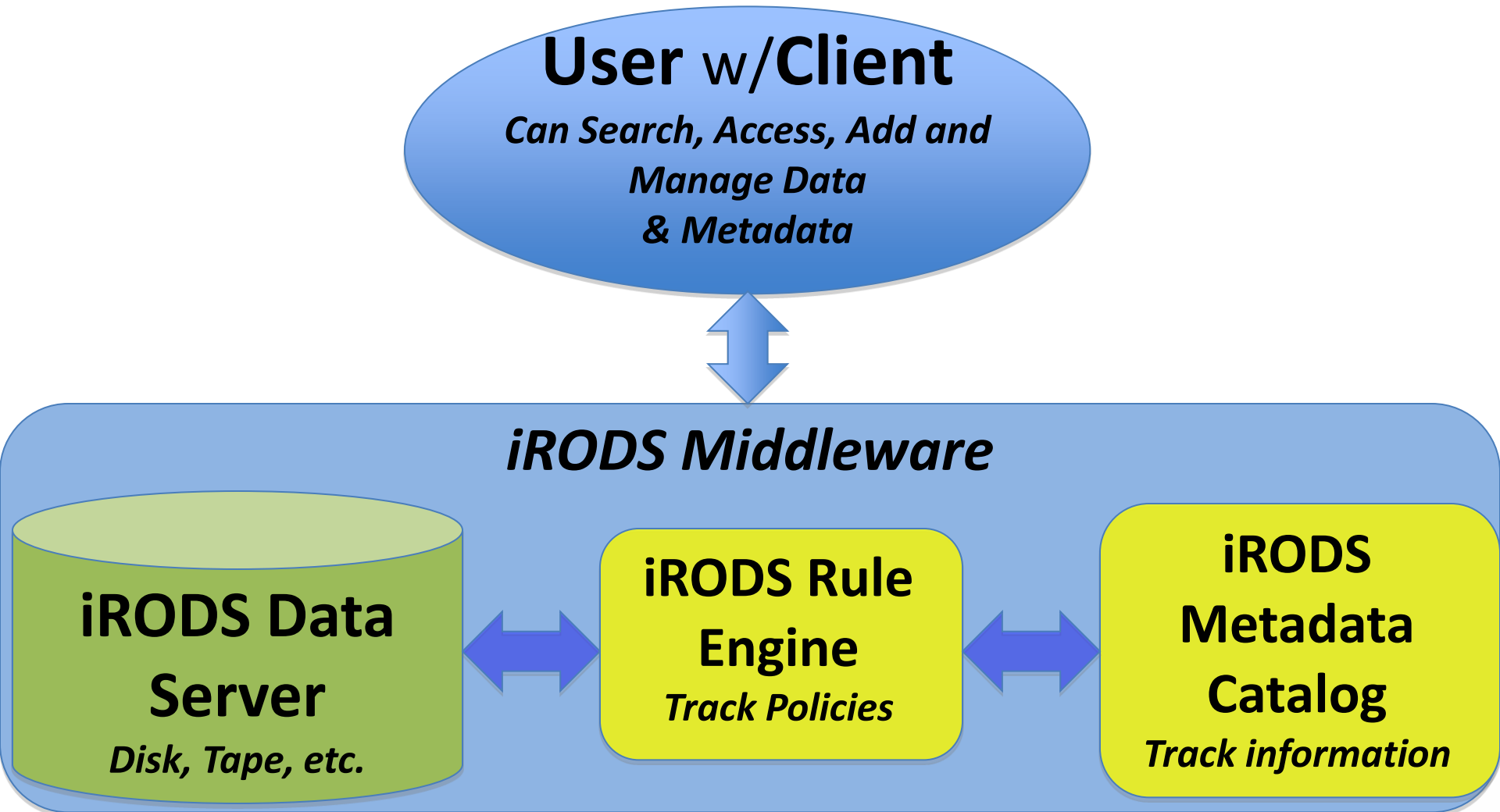
Policy-based Data Environments

- *Purpose*   - reason a collection is assembled

- *Properties*   - attributes needed to ensure the **purpose**

- *Policies*   - controls for enforcing desired **properties,**

-          **mapped to computer actionable rules**

- *Procedures* - functions that implement the **policies**

-          **mapped to computer actionable workflows**

- *Persistent state information* - results of applying the **procedures**

-          **mapped to system metadata**

- *Assessment criteria* - validation that **state information** conforms to the desired **purpose**

-          **mapped to periodically executed policies**

Overview of iRODS Architecture

User w/**Client**

Can Search, Access, Add and Manage Data & Metadata

iRODS Middleware

iRODS Data Server

Disk, Tape, etc.

iRODS Rule Engine

Track Policies

iRODS Metadata Catalog

Track information

Access distributed data with Web-based Browser or iRODS GUI or Command Line clients.

Data Management Applications
Generic Infrastructure

- Data grids – PB-size distributed collections
  - Astronomy – NOAO, CyberSKA, LSST
  - High Energy Physics – BaBar, KEK
  - Earth Systems – NASA (MODIS data set)
  - Australian Research Collaboration Service
  - Genomics – UNC-CH/RENCI

- Institutional repositories
  - Carolina Digital Repository

- Libraries
  - Texas Digital Libraries
  - Seismology - Southern California Earthquake Center

- Archives
  - Ocean Observatories Initiative

Data Virtualization

Access Interface

Policy Enforcement Points

Standard Micro-services

Standard I/O Operations

Storage Protocol

Storage System

Data Grid

Map from the actions requested by the client to multiple policy enforcement points.

Map from policy to standard micro-services.

Map from micro-services to standard Posix I/O operations.

Map standard I/O operations to the protocol supported by the storage system

Data Grid Clients (48)

API	Client	Developer	Language
Browser			
	DCAPE	UNC	
	iExplore	RENCI-Oleg	C++
	JUX	IN2P3	Jargon
	Peta Web browser	PetaShare	
	iDrop web browser	Mike Conway	Java
	Davis web interface	ARCS	
	Rich web client	Lisa Stillwell - RENCI	
Digital Library			
	Akubra/iRODS	DICE	Jargon
	Dspace	MIT	
	Fedora on Fuse	IN2P3	FUSE
	Fedora/iRODS module	DICE	Jargon
	Islandora	DICE	Jargon
	Curators Workbench	CDR-UNC-CH	Jargon
File System			
	Davis - Webdav	ARCS	Jargon
	Dropbox / **iDrop**	DICE-Mike Conway	Jargon
	FUSE	IN2P3, DICE,	FUSE
	FUSE optimization	PetaShare	FUSE
	OpenDAP	ARCS	
	PetaFS (Fuse)	Petashare - LSU	
	Petashell (Parrot)	PetaShare	

iRODS Clients (Cont.)

Grid			
	GridFTP - Griffin	ARCS	
	Jsaga	IN2P3	Jargon
	Parrot	UND - Doug Thain	
	SRM	Academia Sinica	
	Saga	KEK	
I/O Libraries			
	PRODS - PHP	Renci - Lisa Stillwell	
	C API	DICE-Mike Wan	C
	C I/O library	DICE-Wayne Schroeder	C
	Fortran	Schroeder	C
	Eclipse file system	CDR - UNC-CH	Jargon
	Jargon	DICE-Mike Conway	Jargon
	Pyrods - Python	SHAMAN-Jerome Fusillier	Python
Portal			
	EnginFrame	NICE / RENCI	Jargon
	Petashare Portal	LSU	Jargon

iRODS Clients (Cont.)

Tools			
	Archive tools-NOAO	NOAO	
	Big Board visualization	RENCI	
	iFile	GA Tech	
	i-commands	DICE	
	Pcommands	PetaShare	
	Resource Monitoring	IN2P3	
	Sync-package	Academica Sinica	
	URSpace	Teldap - Academica Sinica	
Web Service			
	VOSpace	IVOA	
	Shibboleth	King's College	
Workflows			
	Kepler - actor	DICE	Jargon
	Stork - interoperability	LSU	
	Workflow Virtualization	LSU	
	Taverna - actor	RENCI	

Policy Enforcement Points

- Currently have 74 locations within iRODS framework where policies are checked.
  - Each action may involve multiple policy enforcements points

- Policy enforcement points
  - Pre-action policy   (selection of storage location)
  - Policy execution   (file deletion control)
  - Post-action policy   (derived data products)

icommands	none	acChkHostAccessControl	acSetPublicUserPolicy	acAclPolicy	acSetRescSchemeForCreate	acRescQuotaPolicy	acSetVaultPathPolicy	acPreProcForModifyDataObjMeta	acPostProcForModifyDataObjMeta	acPreProcForDataObjOpen	acPostProcForOpen	acSetMultiReplPerResc	acPostProcForCreate	acPostProcForPut	acPostProcForCopy	acPostProcForRepl	acPostProcForPhymv	acPreProcForObjRename	acPostProcForObjRename	acPreProcForRmColl	acTrashPolicy	acDataDeletePolicy
icp		x	x	x	x	x	x	x	x	x	x		x		x							
icp -N 2		x	x	x	x	x	x	x	x	x	x		x		x							
iphybun		x	x	x	x	x	x	x	x	x		x										
irepl		x	x	x	x	x	x			x		x				x						
ibun -cD		x	x	x	x	x	x	x	x				x	x								
iput		x	x	x	x	x	x	x	x				x	x								
iphymv		x	x	x	x	x	x	x	x			x					x					
imv		x	x	x			x	x	x			x						x	x			
irm		x	x	x			x	x	x			x						x	x		x	x
irm -r collection		x	x	x			x	x	x			x						x	x	x	x	x

Policy Enforcement Points (74)

ACTION	PRE-ACTION POLICY	POST-ACTION POLICY
acCreateUser	acPreProcForCreateUser	acPostProcForCreateUser
acDeleteUser	acPreProcForDeleteUser	acPostProcForDeleteUser
acGetUserbyDN	acPreProcForModifyUser	acPostProcForModifyUser
acTrashPolicy	acPreProcForModifyUserGroup	acPostProcForModifyUserGroup
acAclPolicy	acChkHostAccessControl	acPostProcForDelete
acSetCreateConditions	acPreProcForCollCreate	acPostProcForCollCreate
acDataDeletePolicy	acPreProcForRmColl	acPostProcForRmColl
acRenameLocalZone	acPreProcForModifyAVUMetadata	acPostProcForModifyAVUMetadata
acSetRescSchemeForCreate	acPreProcForModifyCollMeta	acPostProcForModifyCollMeta
acRescQuotaPolicy	acPreProcForModifyDataObjMeta	acPostProcForModifyDataObjMeta
acSetMultiReplPerResc	acPreProcForModifyAccessControl	acPostProcForModifyAccessControl
acSetNumThreads	acPreprocForDataObjOpen	acPostProcForOpen
acVacuum	acPreProcForObjRename	acPostProcForObjRename
acSetResourceList	acPreProcForCreateResource	acPostProcForCreateResource
acSetCopyNumber	acPreProcForDeleteResource	acPostProcForDeleteResource
acVerifyChecksum	acPreProcForModifyResource	acPostProcForModifyResource
acCreateUserZoneCollections	acPreProcForModifyResourceGroup	acPostProcForModifyResourceGroup
acDeleteUserZoneCollections	acPreProcForCreateToken	acPostProcForCreateToken
acPurgeFiles	acPreProcForDeleteToken	acPostProcForDeleteToken
acRegisterData	acNoChkFilePathPerm	acPostProcForFilePathReg
acGetIcatResults	acPreProcForGenQuery	acPostProcForGenQuery
acSetPublicUserPolicy	acSetReServerNumProc	acPostProcForPut
acCreateDefaultCollections	acSetVaultPathPolicy	acPostProcForCopy
acDeleteDefaultCollections		acPostProcForCreate

iput ../src/irm.c   **checks 10 policy hooks**

srbbrick14:10900:ApplyRule#116:: acChkHostAccessControl
srbbrick14:10900:GotRule#117:: acChkHostAccessControl
srbbrick14:10900:ApplyRule#118:: acSetPublicUserPolicy
srbbrick14:10900:GotRule#119:: acSetPublicUserPolicy
srbbrick14:10900:ApplyRule#120:: acAclPolicy
srbbrick14:10900:GotRule#121:: acAclPolicy
srbbrick14:10900:ApplyRule#122:: acSetRescSchemeForCreate
srbbrick14:10900:GotRule#123:: acSetRescSchemeForCreate
srbbrick14:10900:execMicroSrvc#124:: msiSetDefaultResc(demoResc,null)
srbbrick14:10900:ApplyRule#125:: acRescQuotaPolicy
srbbrick14:10900:GotRule#126:: acRescQuotaPolicy
srbbrick14:10900:execMicroSrvc#127:: msiSetRescQuotaPolicy(off)
srbbrick14:10900:ApplyRule#128:: acSetVaultPathPolicy
srbbrick14:10900:GotRule#129:: acSetVaultPathPolicy
srbbrick14:10900:execMicroSrvc#130:: msiSetGraftPathScheme(no,1)
srbbrick14:10900:ApplyRule#131:: acPreProcForModifyDataObjMeta
srbbrick14:10900:GotRule#132:: acPreProcForModifyDataObjMeta
srbbrick14:10900:ApplyRule#133:: acPostProcForModifyDataObjMeta
srbbrick14:10900:GotRule#134:: acPostProcForModifyDataObjMeta
srbbrick14:10900:ApplyRule#135:: acPostProcForCreate
srbbrick14:10900:GotRule#136:: acPostProcForCreate
srbbrick14:10900:ApplyRule#137:: acPostProcForPut
srbbrick14:10900:GotRule#138:: acPostProcForPut
srbbrick14:10900:GotRule#139:: acPostProcForPut
srbbrick14:10900:GotRule#140:: acPostProcForPut

renci

Actionable Rule

- Name | Constraint | Workflow | Recovery

- Associate name with
    - Policy enforcement point
    - User defined rule

- Constraint is a test on any
    - Session variable
    - Persistent state attribute

- Workflow composed by chaining micro-services

- Recovery workflow composed from micro-services

Hello World

```
myTestRule {
# Rule to print out "Hello World"
    writeLine("stdout", "Hello World");
}
INPUT null
OUTPUT ruleExecOut
```

Policies – actionable rules

- Retention, disposition, distribution, arrangement
- Authenticity, provenance, description
- Integrity, replication, synchronization
- Deletion, trash cans, versioning
- Archiving, staging, caching
- Authentication, authorization, redaction
- Access, approval, IRB, audit trails, report generation
- Assessment criteria, validation
- Derived data product generation, format parsing
- Federation of independent data grids

Highly Controlled Environment

- All accesses are authenticated
  - GSI / Kerberos / Challenge-response / Shibboleth
- All operations are authorized
  - ACLs on files, storage
  - User groups, storage groups
- All policies evaluate a constraint
  - Constraints based on persistent state information and session information

iRODS Extensible Infrastructure

- Clients – specific to discipline and life cycle state
- Policies – specific to discipline
- Procedures – specific to discipline
- Remaining infrastructure is generic
  - Network transport
  - Authentication / Authorization
  - Distributed storage access
  - Remote execution
  - Metadata management
  - Message passing / distributed debugging
  - Rule engine

Extended Capabilities

- Replication
- Registration of files into the data grid
- Synchronization of remote directory
- Managed file transport (iDrop)
- Automated metadata extraction
- Queries on metadata, tags
- Server-side workflows (loop over result sets)
- Parallel I/O streams & RBUDP transport

Cloud Integration

- iRODS drivers for
  - EC2
  - S3
- Applications include
  - http://www.fiercegovernmentit.com/story/nasa-touts-nebula-faults-public-cloud/2011-08-22
  - http://nebula.nasa.gov/media/uploads/nasa-nebula-in-action.pdf

Integration of Data and Workflows

- **Client-side workflows**
  - Presentation / display
  - Client managed synchronization / transport
- Policy-driven workflows
  - Automation of administrative functions
  - Processing at the storage system
  - Data-intensive computations
- Traditional computer workflows
  - CPU-intensive computations

Expectations

- Data collection sizes will increase
  - Now petabytes, soon exabytes
    - 1 PB/year = 32 MB/sec
    - 1 PB/day = 11.6 GB/sec
  - Indexing done in the storage system
- Integration of data manipulation with storage controllers - DDN SFA10KE
  - Analyses done within the storage system

Future Applications

- Digital libraries
  - Continuous indexing of contents
- Scientific data collections
  - Extraction of features from data sets
  - Creation of derived data products
- Archives
  - Transformative migrations
  - Validation of assessment criteria

iRODS Development

- Realized Objects
  - Micro-service Structured Object (MSSO)
  - Register into iRODS the workflow needed to create a derived data product and the input parameter / files
  - Accessing the link causes the derived data to be generated and registered as a replica
  - Implemented using compound resource concept
  - The realized object workflow can reference other realized objects, and dynamically re-create an object when any underlying input resource is altered.
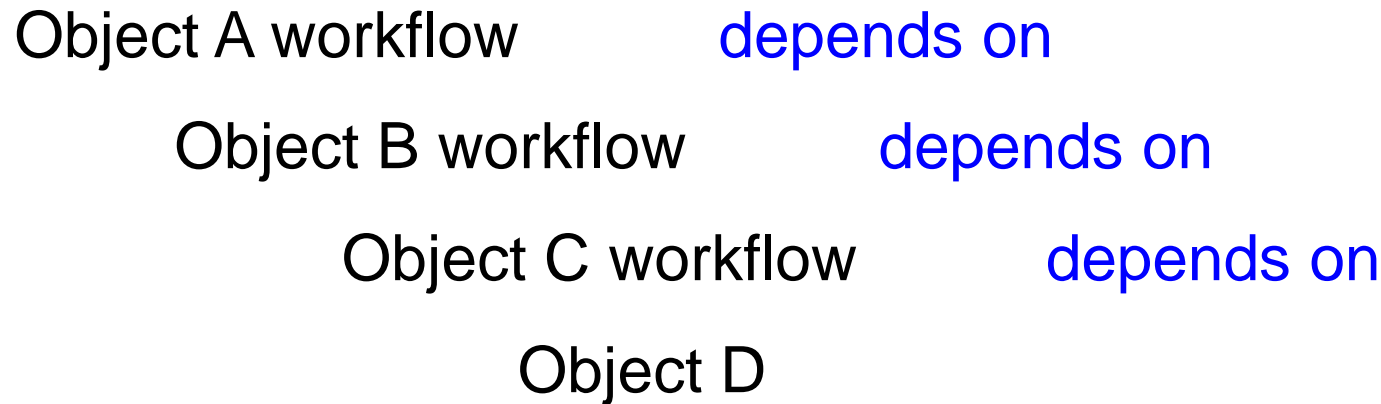 - Effectively manages provenance for workflows

Automated Re-processing Design

Object A workflow   depends on

    Object B workflow   depends on

        Object C workflow   depends on

        Object D

Change object D, and system automatically re-computes objects C, B, and A when object A is clicked

DataNet Federation Consortium
Data Driven Science

- Implement national data grid
  - Federate existing discipline-specific data management systems to enable national research collaborations

- Enable collaborative research on shared data collections
  - Manage collection life cycle as the user community broadens

- Integrate "live" research data into education initiatives
  - Enable student research participation through control policies

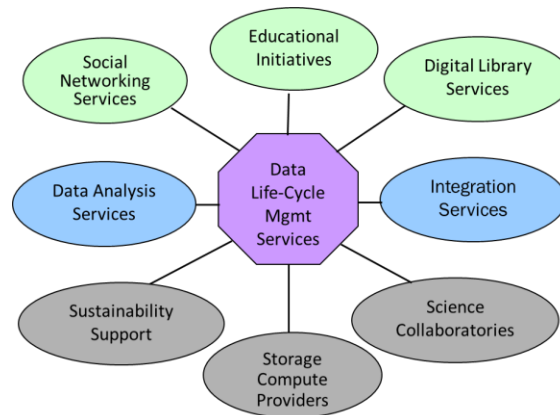| Project |
| Shared Collection |
| Processing Pipeline |
| Digital Library |
| Reference Collection |
| Federation |

Collection Life Cycle

Cyber-infrastructure Partners:
Univ. of North Carolina, Chapel Hill
Univ. of California, San Diego
Arizona State University
Drexel University
Duke University
University of Arizona
University of South Carolina

Science and Engineering Initiatives:
Ocean Observatories Initiative
the iPlant Collaborative
CUAHSI
CIBER-U
Odum Social Science Institute
Temporal Dynamics of Learning Center

Policy-based data management

National Science Foundation Cooperative Agreement: OCI-0940841

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

Replication Validation Rule

- Loop over collection in sets of 256 files
- Restart capability for long-running session
  - Tracks identity of the files previously verified
  - Sets TEST_DATA_ID attribute on the collection
- Control execution rate
  - Slow down rate when executing too fast
- Audit trail of all changes to files
- Identify and replace corrupted files
- Create missing replicas
  - Distribute new replicas across storage vaults

iRODS - Open Source Software

Reagan W. Moore

rwmoore@renci.org

http://irods.diceresearch.org