Policy-Based Distributed Data Management Systems

Reagan W. Moore

Arcot Rajasekar

Mike Wan

{moore,sekar,mwan}@diceresearch.org

http://irods.diceresearch.org

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

UCSD

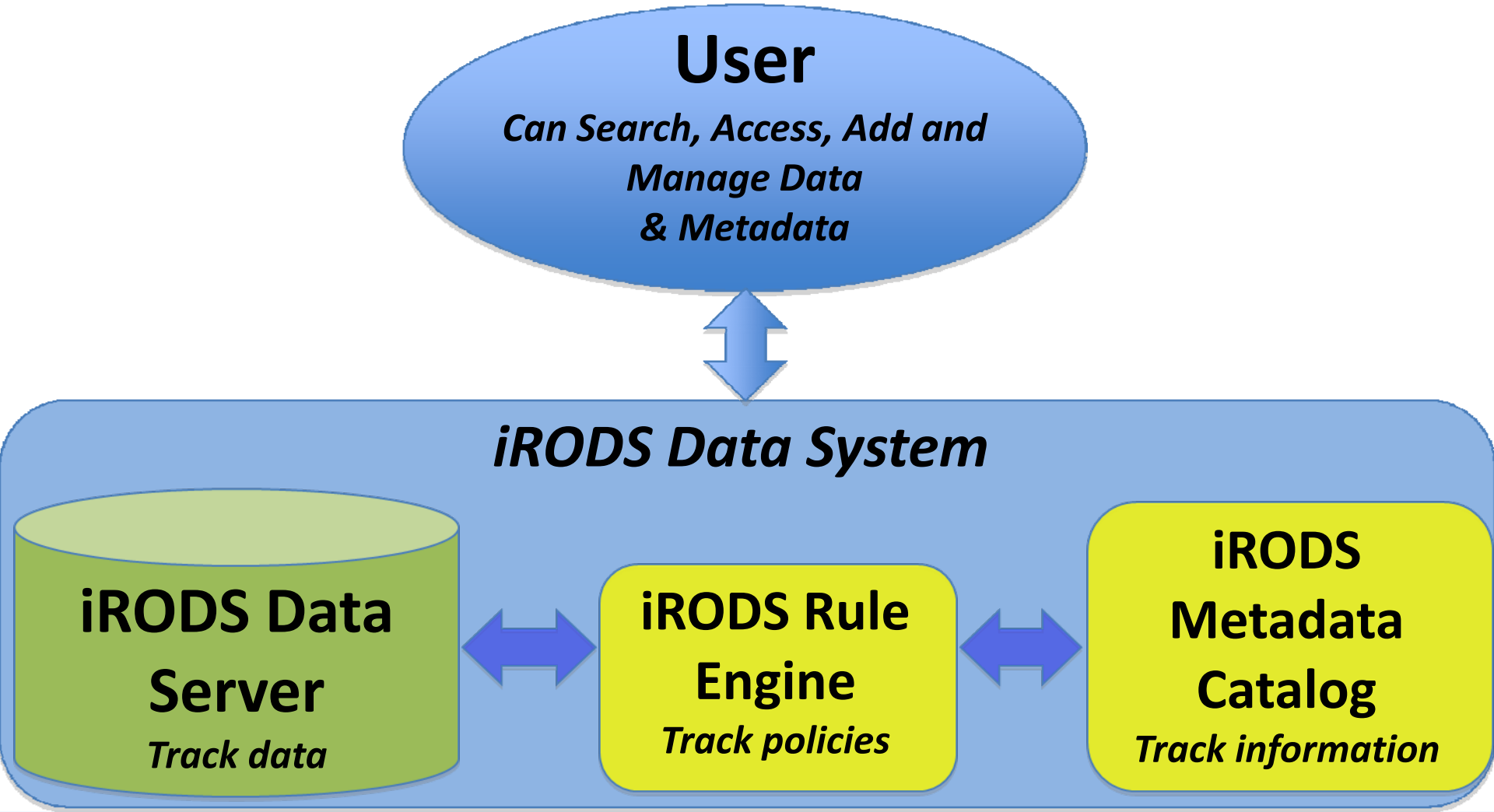
Abstract

Digital repositories can be defined by their policies and procedures. The integrated Rule Oriented Data System explicitly characterizes policies as computer actionable rules and procedures as computer executable micro-services. By tuning the policies and procedures, different data management applications can be created, including digital libraries for publishing data, persistent archives for preserving data, and data grids. Rules can be implemented that validate assessment criteria, automate administrative management functions, and enforce management policies. We examine the design criteria behind the creation of policy-based distributed data management systems, and the capabilities that are enabled.

Data Management Challenges

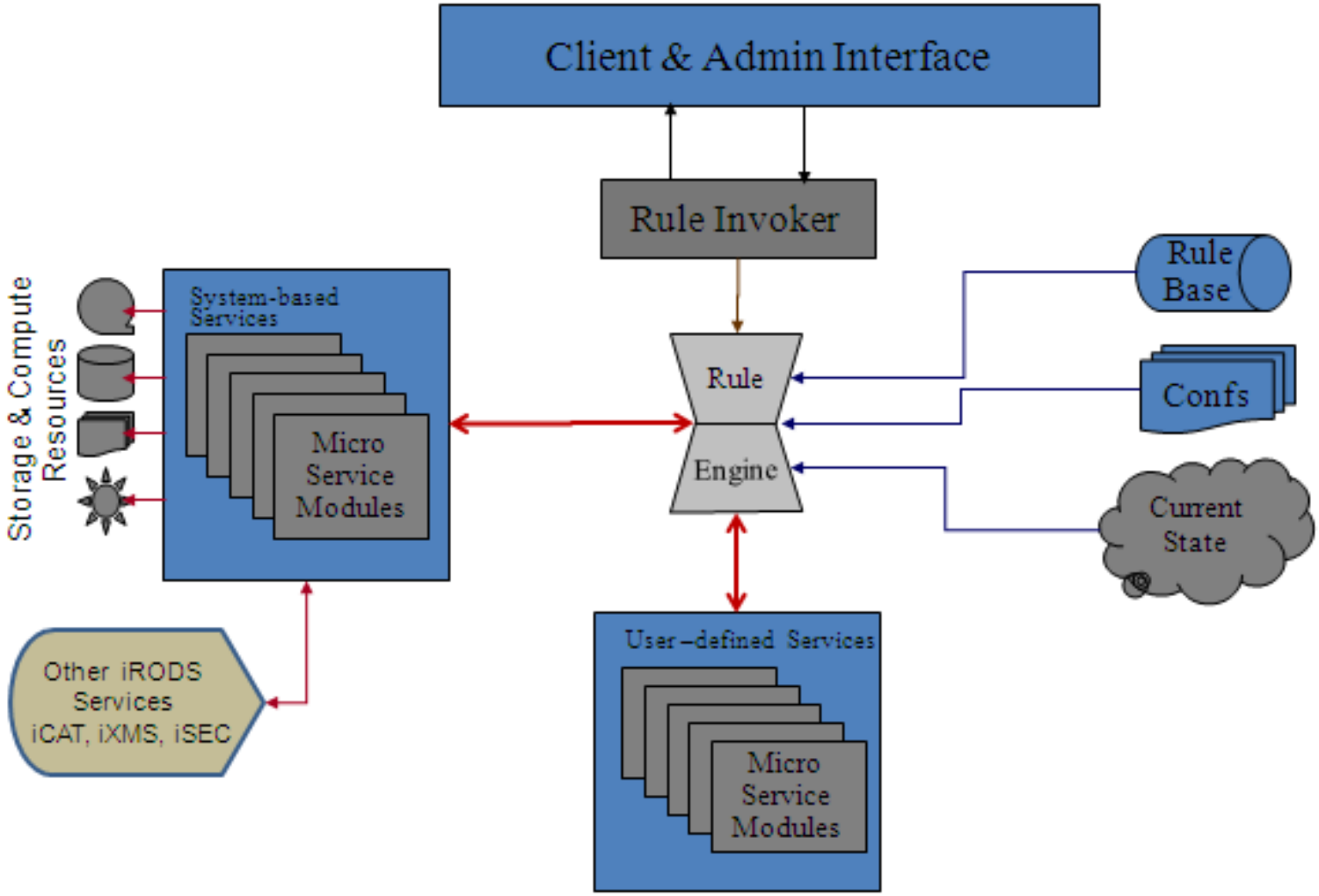
- Data driven research generates massive data collections
  - Data sources are remote and distributed
  - Collaborators are remote
  - Wide variety of data types: observational data, experimental data, simulation data, real-time data, office products, web pages, multi-media
- Collections contain millions of files
  - Logical arrangement is needed for distributed data
  - Discovery requires the addition of descriptive metadata
- Long-term retention requires migration of output into a reference collection
  - Automation of administrative functions is essential to minimize long-term labor support costs
  - Creation of representation information for describing file context
  - Validation of assessment criteria (authenticity, integrity)

Overview of iRODS Data System



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

iRODS Resource Server



iRODS Micro-Services

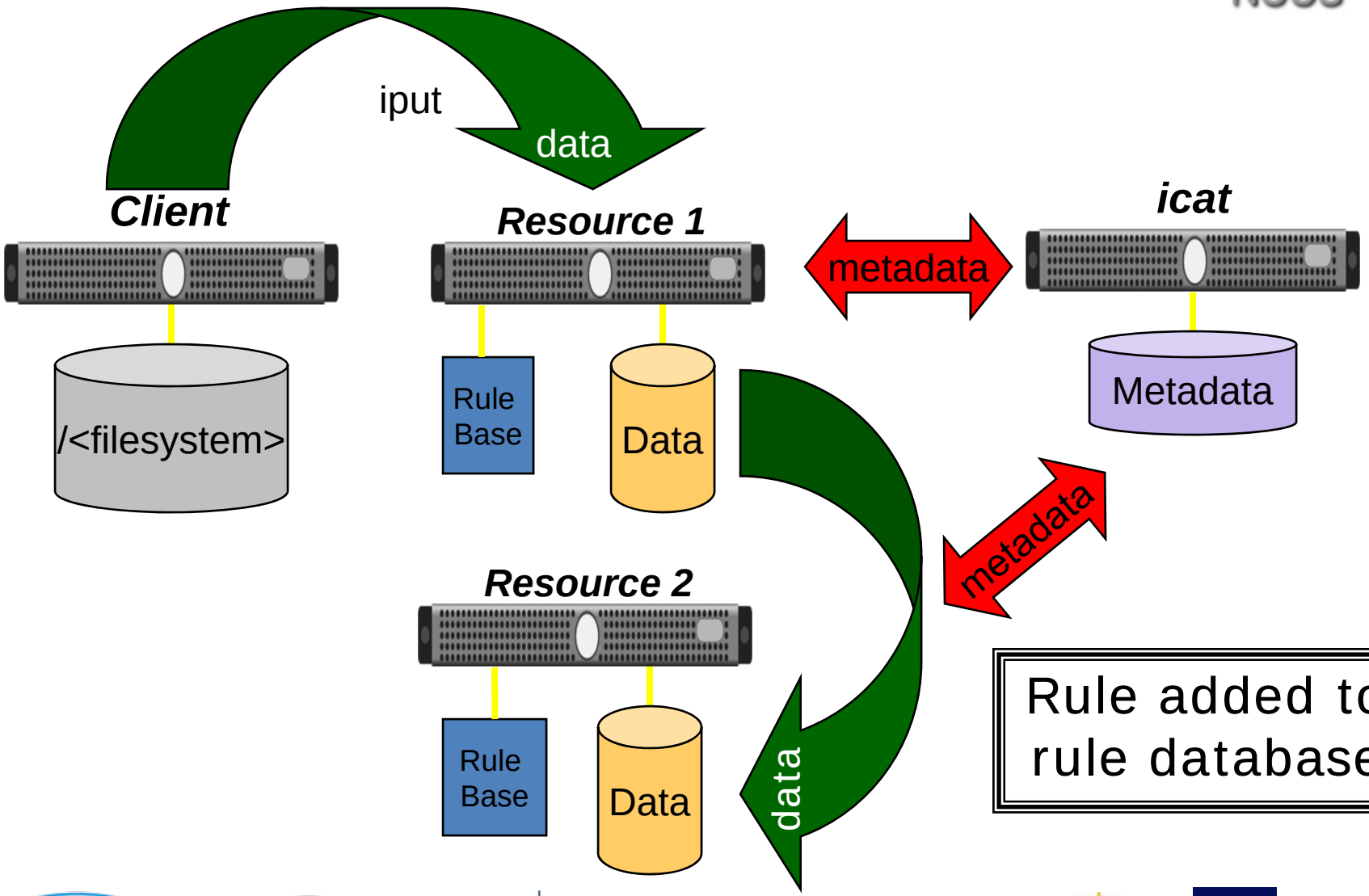
- Function snippets that wrap a well-defined process
  - Compute checksum
  - Replicate file
  - Integrity check
  - Zoom image
  - Get SDSS image cutout
  - Search PubMed
- Written in C or Python (PHP, Java soon)
  - Recovery micro-services to handle failure
  - Web services can be wrapped as micro-services
- Can be chained to perform complex tasks
  - Micro-services invoked by rule engine

iRODS Rules

- Server-side workflows

  Action | condition | workflow chain | recovery chain
- Condition - test on any attribute:
  - Collection, file name, storage system, file type, user group, elapsed time, IRB approval flag, descriptive metadata
- Workflow chain:
  - Micro-services / rules that are executed at the storage system
- Recovery chain:
  - Micro-services / rules that are used to recover from errors

iput With Replication

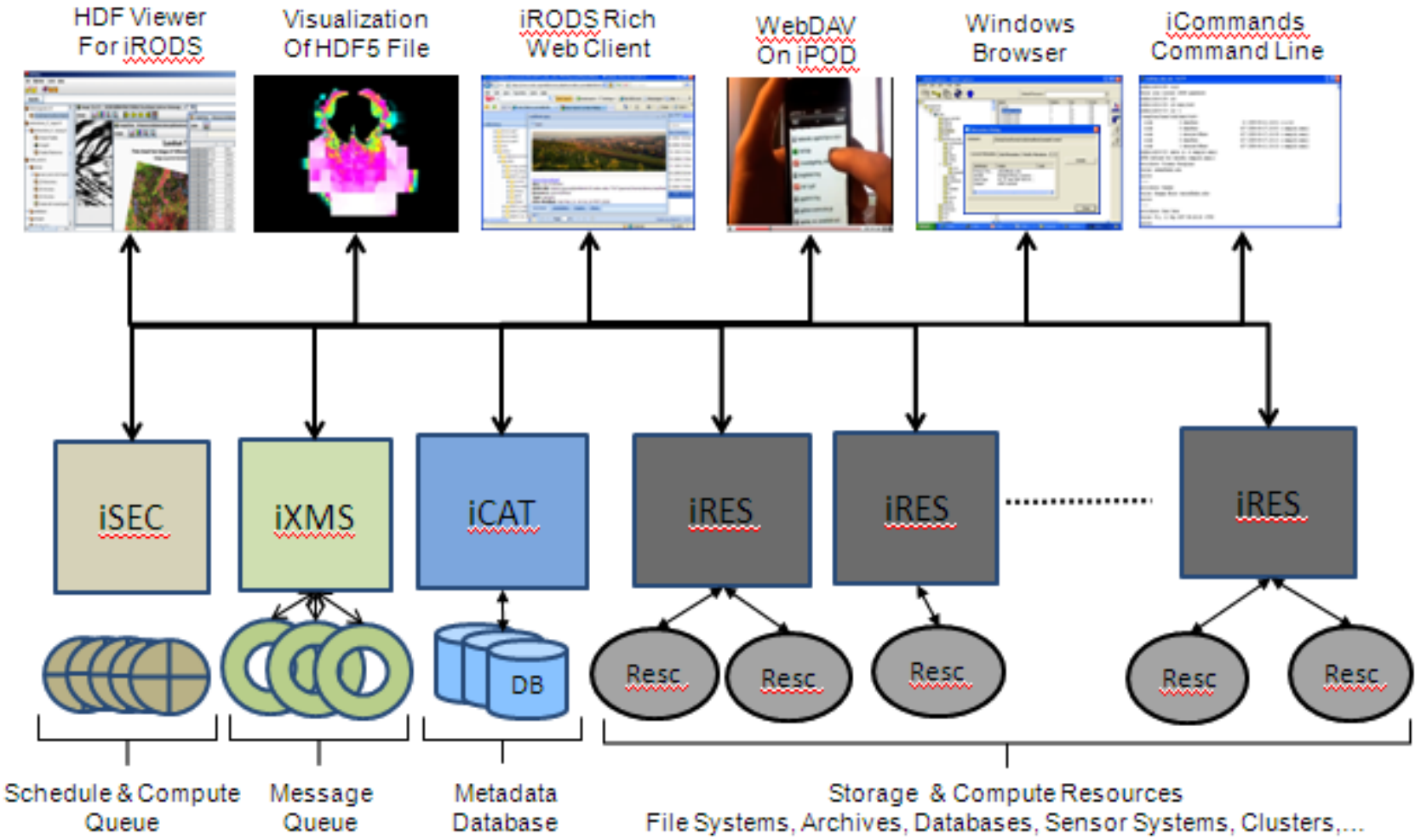


Rule added to rule database

Policy-Virtualization: Automate Operations

- System-centric Policies & Obligations:
  - Manage retention, disposition, distribution, replication, integrity, authenticity, chain of custody, access controls, representation information, descriptive information requirement, logical arrangement, audit trails, authorization, authentication
- Domain-specific Policies:
  - Identification & Extraction of Metadata
  - Ingestion Control for Provenance Attribution
  - Processing of Data on Ingestion
    - Creation of multi-resolution images, type-identification, anonymization,…
  - Processing of Data on Access
    - IRB Approval for data access, Data sub-setting, Merging of multiple images, conversion, redaction, …

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

SRB

Policy/rule execution

- Immediate - enforced at time of action invocation
- Deferred - applied at a future time
- Periodic - applied at defined interval
- Interactive - applied on demand

- iSEC scheduler / batch system supports
  - Local workflows
  - Distributed workflows
  - Deferred and periodic workflows
  - (Launch micro-services on clusters, clouds, supercomputers)

Checksum Validation Rule

```
myChecksumRule{
    msiMakeQuery("DATA_NAME, COLL_NAME, DATA_CHECKSUM",*Condition,*Query);
    msiExecStrCondQuery(*Query,*B);
    assign(*A,0);
    forEachExec (*B)  {
            msiGetValByKey(*B,COLL_NAME,*C);
            msiGetValByKey(*B,DATA_NAME,*D);
            msiGetValByKey(*B,DATA_CHECKSUM,*E);
            msiDataObjChksum(*B,*Operation,*F);
            ifExec  (*E != *F)  {
                 writeLine(stdout,file *C/*D has registered checksum *E and computed checksum *F);
            }
            else {
                 assign(*A,*A + 1);
            }
    }
    ifExec(*A > 0)  {
          writeLine(stdout, have *A good files);
    }
}
```

*Condition can be COLL_NAME like '/ils161/home/moore/genealogy/%'


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


Preservation is an Integral Part of the Data Life Cycle

- Organize project data into a shared collection
- Publish data in a digital library for use by other researchers
- Enable data-discovery & data-driven analyses
- Preserve reference collections for use by future research initiatives
- Analyze new collection against prior state-of-the-art data
- Define & Enforce Policies for long-term management and curation


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


To Manage Long-term Preservation

- Define desired preservation properties
  - Authenticity / Integrity / Chain of Custody / Original arrangement
  - Life Cycle Data Requirements Guide
- Implement preservation processes
  - Appraisal / accession / arrangement / description / preservation / access
- Manage preservation environment
  - Minimize costs
  - Validate assessment criteria to verify preservation properties

ISO MOIMS
repository assessment criteria

- Are developing 150 rules that implement the ISO assessment criteria

90	***Verify descriptive metadata and source against SIP template and set SIP compliance flag***
91	***Verify descriptive metadata against semantic term list***
92	***Verify status of metadata catalog backup (create a snapshot of metadata catalog)***
93	***Verify consistency of preservation metadata after hardware change or error***

iRODS Evaluations

- NASA Jet Propulsion Laboratory
  - iRODS selected for managing distribution of Planetary Data System records
- NASA National Center for Computational Sciences
  - iRODS chosen to manage archive of simulation output and serve as access data cache for distribution
- AVETEC appraisal for DoD HPC centers
  - iRODS now provides all required capabilities
- French National Library
  - iRODS rules control ingestion, access, and audit functions
- Australian Research Coordination Service
  - iRODS manages data distributed between academic institutions

Development Team

- DICE team
  - Arcot Rajasekar - iRODS development lead
  - Mike Wan - iRODS chief architect
  - Wayne Schroeder - iRODS developer
  - Bing Zhu - Fedora, Windows
  - Lucas Gilbert - Java (Jargon), DSpace
  - Paul Tooby - documentation, foundation
  - Sheau-Yen Chen - data grid administration
- Preservation
  - Richard Marciano - Preservation development lead
  - Chien-Yi Hou - preservation micro-services
  - Antoine de Torcy - preservation micro-services


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


Foundation

- Data Intensive Cyber-environments
  - Non-profit open source software development
  - Promote use of iRODS technology
  - Support standards efforts
  - Coordinate international development efforts
    - IN2P3 - quota and monitoring system
    - King's College London - Shibboleth
    - Australian Research Collaboration Services - WebDAV
    - Academia Sinica - SRM interface


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


For more information:

Reagan W. Moore
rwmoore@renci.org
http://irods.diceresearch.org

NSF OCI-0848296 "NARA Transcontinental Persistent Archives Prototype"
NSF SDCI-0721400 "Data Grids for Community Driven Applications"


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL