Big Data Management for Science

Terrell G. Russell, Ph.D.
Data Management Research Scientist
Renaissance Computing Institute (RENCI)
University of North Carolina at Chapel Hill
tgr@renci.org
@terrellrussell

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

renci
RESEARCH ENGAGEMENT INNOVATION

Background and Justification

- Generating Data → Managing Data
- Grids → Policy
- Automation and Auditing

- iRODS
  - History
  - Philosophy / Goals
  - Capabilities
  - Implementation

- E-iRODS
  - History
  - Philosophy / Goals
  - Process
  - Status


i·R·O·D·S

iRODS

- History
  - Second generation code. Born from original project named Storage Resource Broker (SRB). Developed at UCSD San Diego Supercomputing Center (SDSC).
  - Began in mid-2005, 0.5 release in Dec 2006, 1.0 release in Jan 2008, 3.1 release in Mar 2012, 3.2 release next month.
  - Most of the development team moved to SILS and RENCI at UNC-Chapel Hill in 2008-2009.

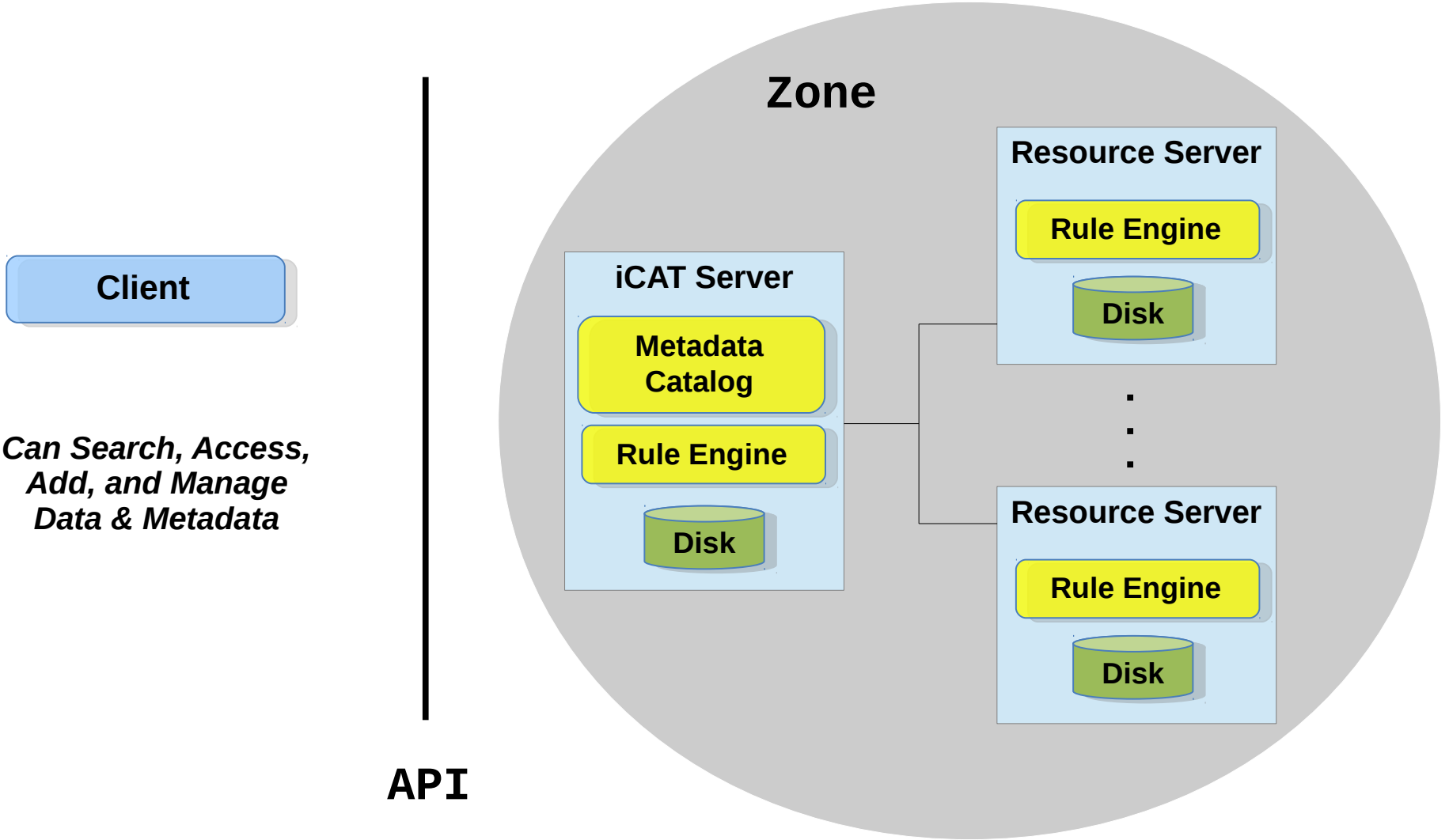
- Philosophy / Goals
  - Open source
  - Compile and run on as many systems as possible
  - Maximum flexibility and connectivity to existing infrastructure
  - Designed with federation in mind from the beginning
  - Provide computer actionable rules and policy atop a virtual filesystem
  - Provide full auditing capabilities regarding access, fixity, and provenance for every file ingested into the system

iRODS

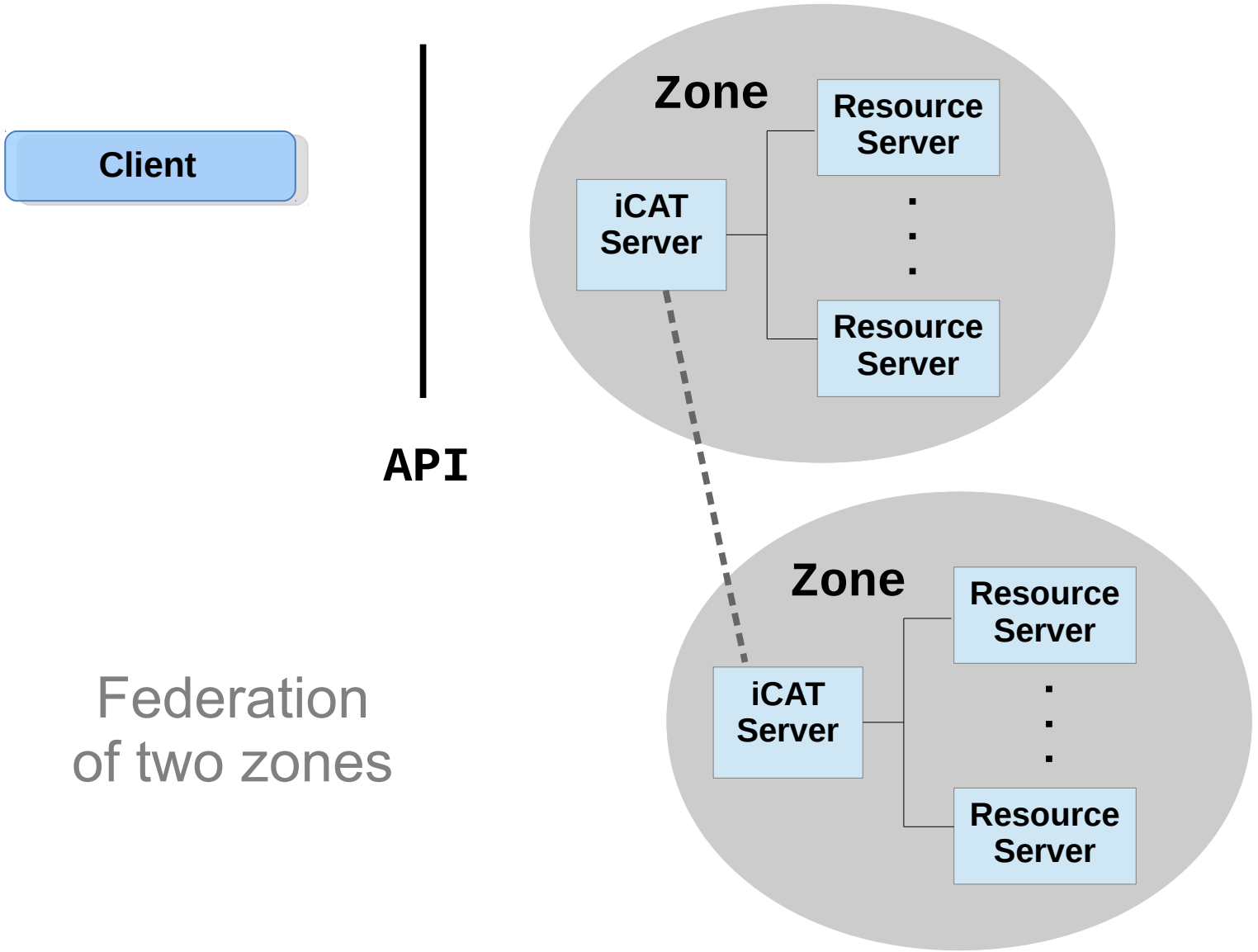
- Capabilities
  - Can connect to many different types of storage systems and databases
  - C and Java APIs
  - Rule Engine with immediate or delayed execution
  - Policy enforcement points before and after nearly every action that can be taken within the system
  - Audit Trails, Metadata, Access Control, Authentication
  - Zone to Zone Federation

- Implementation
  - C
  - Installation via perl scripts
  - Most options are compile time variables
  - Compile in place, including dependencies like PostgreSQL and unixODBC
  - Sparse documentation of how things work under the hood or why they were designed the way they are

iRODS Architecture



iRODS Architecture



Federation of two zones

iRODS Policy Enforcement Points

- Currently 74 locations where policies are checked and rules can be executed based on appropriate conditions
  - Authentication
  - Access control
  - Pre-action policy (PreProcForPut)
  - Policy execution
  - Post-action policy (PostProcForPut)
  - Many others...

iRODS Around the World

- Data grids – Petabyte-scale 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


irods
enterprise data management

E-iRODS

- History
  - Developed by RENCI at UNC-Chapel Hill
  - Forked from community 3.0 code in early 2012
  - Tracking bug fixes and selected new features
  - 3.0b2 is available at http://www.e-irods.org

- Philosophy / Goals
  - Open source
  - Minimal movement in the core
  - Functionality provided by plugins
  - Easy installation and upgrades
  - Use system-provided software
  - Sustainability
  - Maintenance
  - Support

E-iRODS - Process

- Refactorization
  - Object oriented
  - Dependency inversion
  - Simple modular architecture
- Issue Tracking
  - Prioritized
  - Issue ownership
  - Every commit is tagged
- Documentation
  - Doxygen
  - Administration manual(s)
  - Compiled from source as well
- Automate Everything
  - Continuous Integration
    - Code coverage
    - Static analysis
  - Testing
    - Unit tests
    - Functional tests
    - Regression tests
    - Topology tests
    - Federation tests
    - Test independence
  - Packaging
    - EPM generates RPM, DEB, DMG across multiple versions of operating systems
    - Dependency management

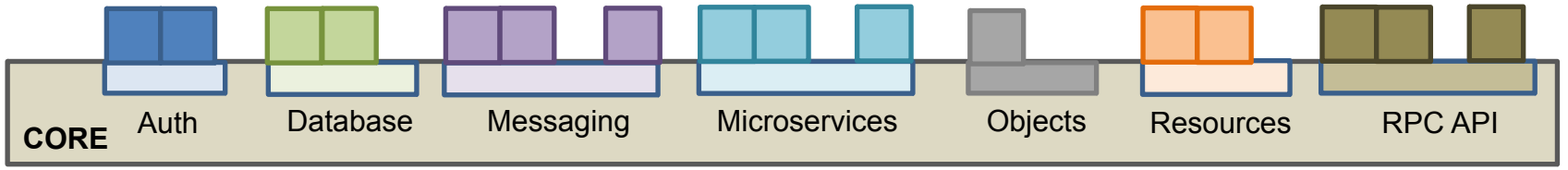
iRODS / E-iRODS Core is a substrate upon which new functionality may be added via six fixed interfaces. The core is designed to be a small, stable broker of extensible services.

Interfaces for Extensibility:
Authentication, Database, Messaging, Microservices, Objects, Resources, RPC API

Plugins extend the functionality of iRODS / E-iRODS relevant to a given interface. They are self contained, dynamically loadable, and could be proprietary.

iRODS/E-iRODS includes a plugin dependency model. Plugins may be inter-dependent and provide new functionality via multiple plugins.

A Bundle of plugins can provide a set of features to support newly created first-class objects within iRODS / E-iRODS such as Tickets or Workflows.



E-iRODS

- Big Data Management for Science
  - Federation / Control
  - Traceability
  - Replicability
  - Workflow Support
  - Auditing
  - Preservation
  - Access

Big Data Management for Science

Terrell G. Russell, Ph.D.
Data Management Research Scientist
Renaissance Computing Institute (RENCI)
University of North Carolina at Chapel Hill
tgr@renci.org
@terrellrussell

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

renci
RESEARCH ENGAGEMENT INNOVATION