iRODS Technologies at UNC

E-iRODS:

Enterprise iRODS at RENCI

Presenter: Leesa Brieger
leesa@renci.org

renci

UNC Chapel Hill Investment in iRODS

DICE and RENCI: research units of UNC CH

- DICE
  - Developers of iRODS
  - Headed by Reagan Moore
  - Development and support team
    - Arcot Rajasekar
    - Wayne Schroeder
    - Mike Wan
    - Sheau-Yen Chen
    - Hao Xu
    - Antoine DeTorcy
    - Mike Conway

  (DICE-UNC and DICE-UCSD)

- RENCI
  - Developers of E-iRODS
  - Director of RENCI: Stan Ahalt
  - Development and support team
    - Jason Coposky
    - H. Johnson
    - Howard Lander
    - Terrell Russell
    - Michael Stealey
    - Lisa Stillwell
    - Leesa Brieger
    - Charles Schmitt

iRODS Technologies

- iRODS
  - Research-oriented/community code
  - Funded by research organizations

- E-iRODS
  - Represents RENCI's (and UNC's) long-term support for iRODS
  - Targeting new funding models for sustainability – moving beyond traditional public research funds
  - First release based on iRODS 3.0
    - second beta came out in June 2012
    - www.e-irods.org
  - Inspired by the RedHat/Fedora open source model
  - E-iRODS Consortium to fund Enterprise support infrastructure

E-iRODS Goals

- Production-ready
- Supports deployment by sys admin without IT staff for development and customization
- Greater modularity: microservices, authentication, resources, …
- Dynamic extensibility: (hot) plug-ins
- Greater security - hardening, signed plug-ins
- User-friendly - installation and use
- Admin-friendly - upgrades and administration
- High availability - scalable, fail-over capabilities
- Dependency inversion
- Back-porting to the iRODS code

renci

E-iRODS Processes

- Refactoring
  - simple modular architecture
  - object-oriented
- Issue Tracking
  - prioritizing issues
  - issue ownership
  - every commit is tagged
- Testing
  - unit, functional, regression
  - cppcheck for static analysis
  - Valgrind for dynamic memory analysis
  - gcov for coverage metrics
  - variety of topologies
  - federated grid tests
- Automate Everything
  - Continuous Integration (CI)
- Packaging
  - EPM metapackager generates RPM, DEB, DMG across multiple OS versions
  - dependency management inherited from OS tools: apt (DEB), yum(RPM)
- Documentation
  - Doxygen
  - administration manuals - compiled from source during packaging
  - user manuals

Collaborative Development and Test Environment at RENCI

Git – distributed revision control system

GForge – project management system

- hosting & version control
- bug-tracking
- messaging
- **https://code.renci.org/gf/project/**

Hudson/Jenkins – Continuous Integration environment: incremental quality control

Nexus – Maven repository that tracks dependencies and bundles for check-out (Java)

renci

Continuous Integration

Automated via Hudson (moving to Jenkins)

A risk reduction technique

Push code frequently to the repository

Build & test for each new commit in order to catch defects as early as possible

Periodic testing as well…

Automated CI removes a level of burden from developers and provides constant insight to the state of the project

renci

Continuous Integration (CI)

RENCI E-iRODS Testing Environment

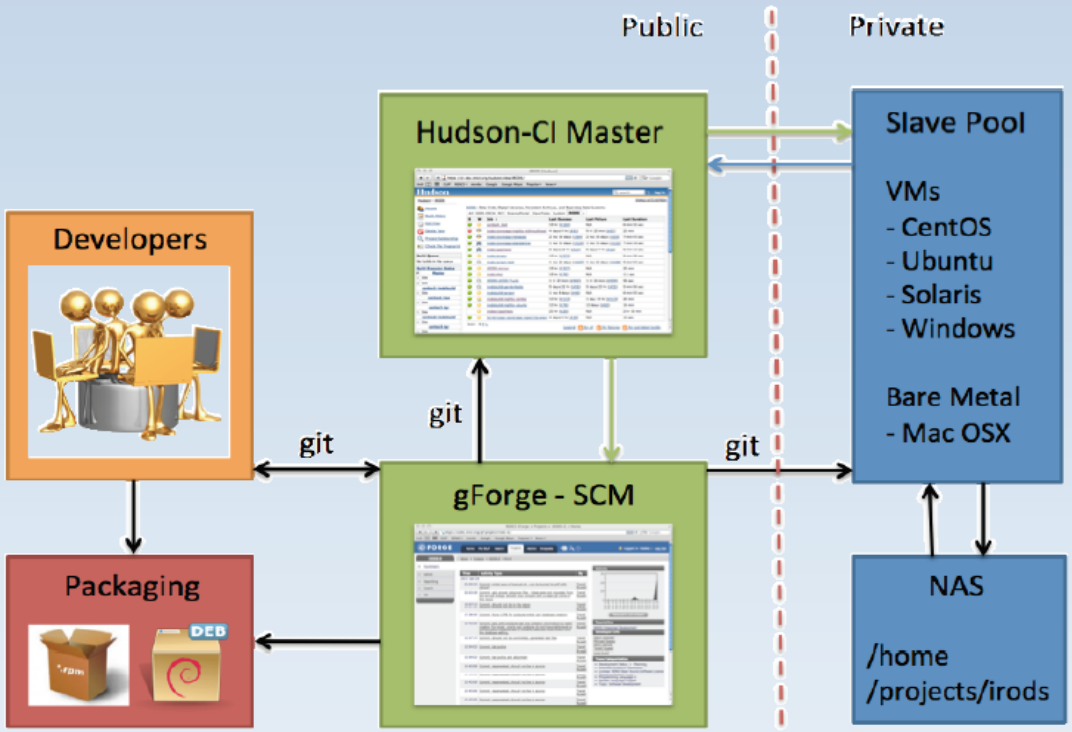
Hudson launches task on Slave Pool

Slave Pool runs script to "deploy a Gridbundle"

- Gridbundle – topological definition of an n-zone iRODS network (json)

Tests run against the resulting "live grid"

- Automated and/or manual testing
- Aggregates test results from various machines



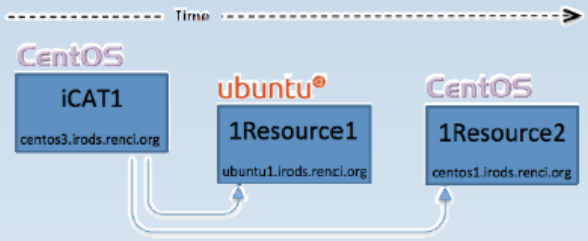
- Automated system testing
- Frequent code pushes to repository
- Continuous Build & Test
- Reduces risk by detecting defects early
- Reduces testing staff time

Testing Framework

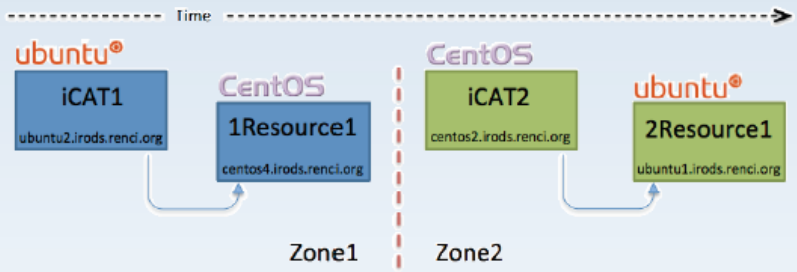
Deploying a Gridbundle
Validate gridbundle is well-formed
Validate testbed capacity is sufficient
For each iCAT server
- Send celery request, wait for success
- Populate gridbundle data structure (IP and hostname)
- For each resource server
  > Send celery request, wait for success
  > Populate gridbundle data structure (IP and hostname)

Write out populated gridbundle to livegrid.json

Single Zone: One iCAT with two resources



Federation: Two iCATs with one resource each



- Predefined grid bundles
- Automated generation of distributed E-iRODS zone
  - NSF Exo-genie infrastructure for dynamically configured virtual networks
- IP and hostname stitching
- Scripted tests and expected outcomes

Testing

- 100% test coverage **of server-side APIs** across selected platforms and topologies: n-way testing across all combinations

- Packages released as of June 2012
  - DEB (Debian, Ubuntu)
  - DMG (MacOSX) – Unix client (icommands)
  - RPM (RHEL, CentOS, Fedora, SUSE)

- Topologies
  - Single zone: iCAT server + 2 non-iCAT servers
  - Federation: two single zones

- Consortium Certification
  - Planned:
    - Solaris
    - Windows (MSI)
    - MacOSX (servers)

E-iRODS View

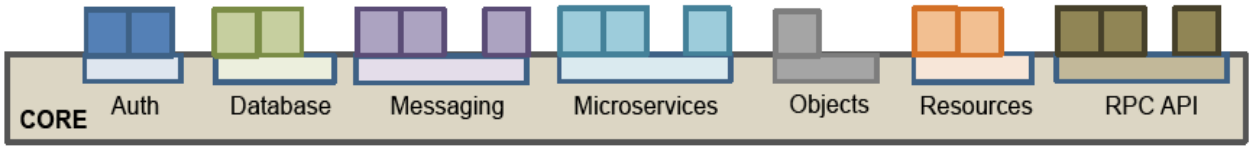
iRODS / E-iRODS Core is a substrate upon which new functionality may be added via fixed interfaces. The core is designed to be a small, stable broker of extensible services.

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.



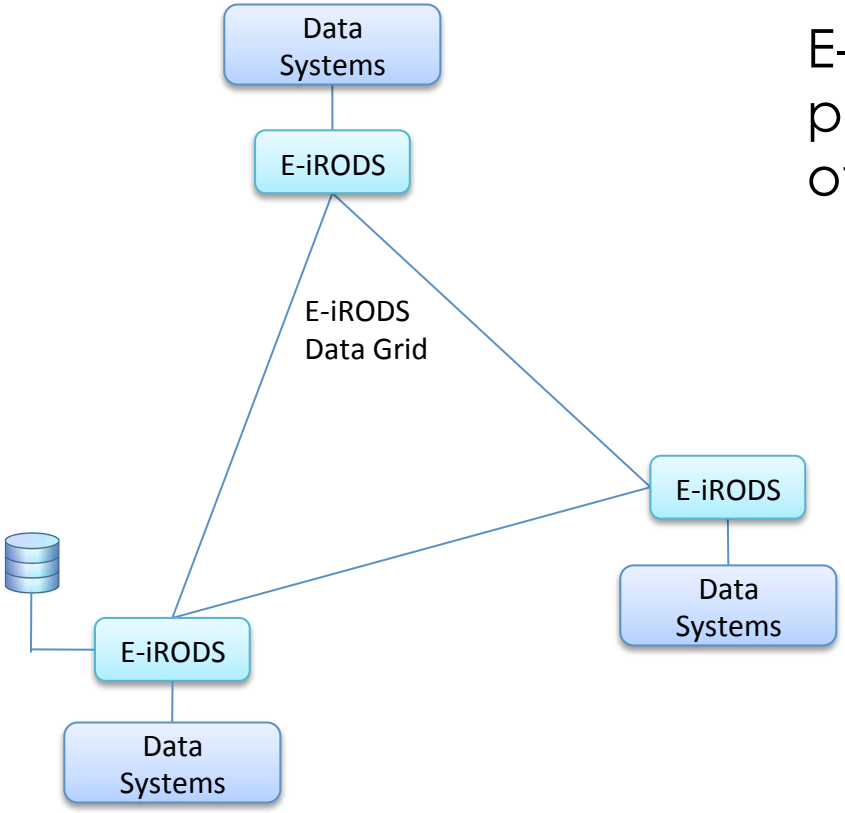
E-iRODS 3.0 will include pluggable microservices and pluggable resources.

Pluggable Microservices

- To be included in E-iRODS 3.0
- Planned for next release of iRODS (3.3)
- Hot pluggability –
  - no need to shut down data grid to update functionality
  - data grid can activate some plug-ins (done by data admin, not sys admin)
- 'Signed' microservices provides greater security
- Plug-in dependency model facilitates upgrade of the data grid

renci

E-iRODS Resources

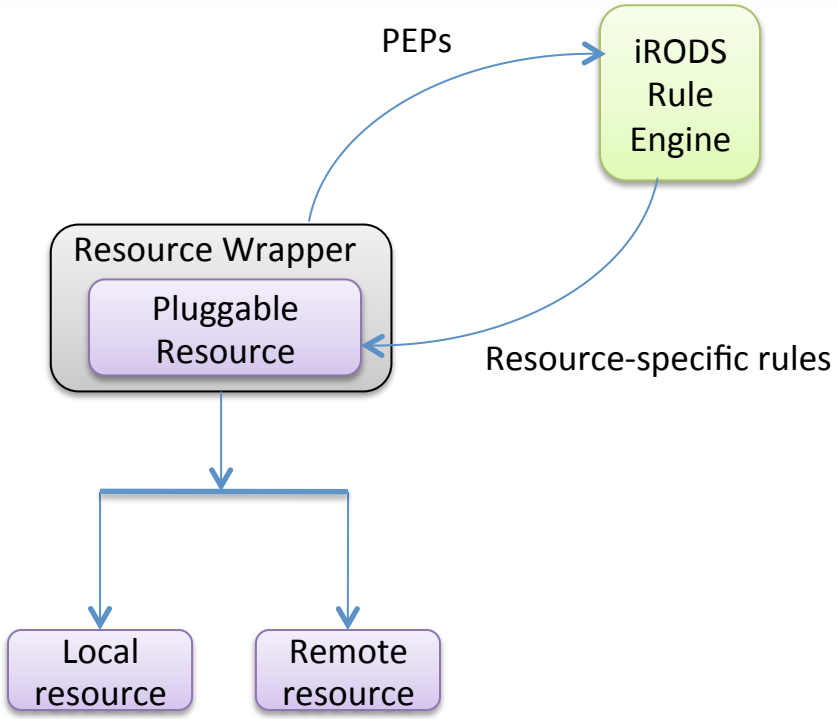


E-iRODS:
production management
of distributed data assets

- unified view and management of disparate storage systems
- policy-driven data management
- private/community data cloud provisioning
- extensible and open-source

renci

Policy-Managed Pluggable Resources



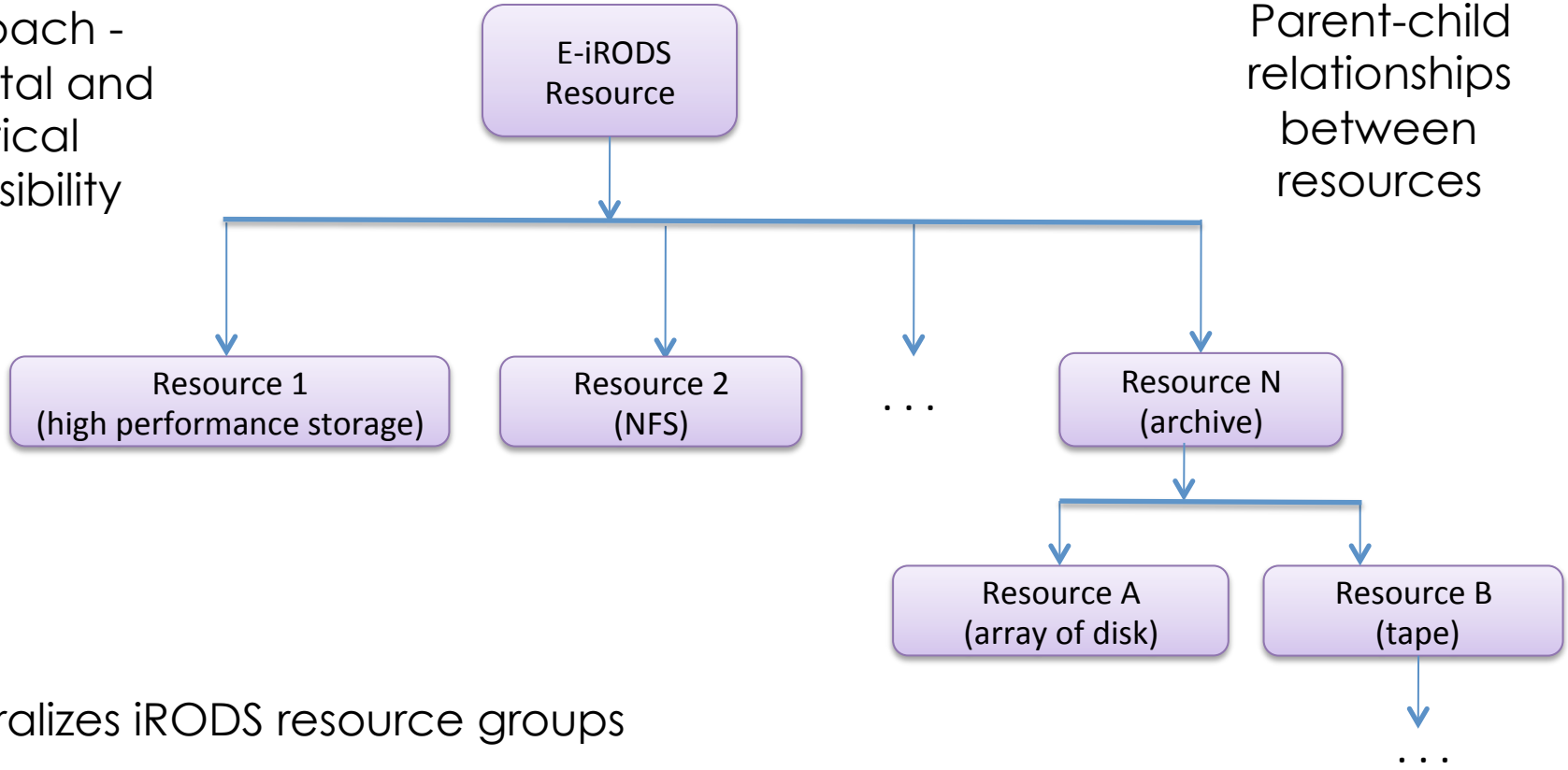
- User develops pluggable resource
- Code inspection allows for auto-generated Policy Enforcement Points (**PEPs**)
- Grid admin can then develop standard iRODS policy-enforcing rules specific to the resource

Use Case Example

- Pluggable resource replicates off-site to ensure high availability, with an exception…
- iRODS rule turns off remote replication for files tagged with 'Protected Health Information' metadata

E-iRODS Composable Resources

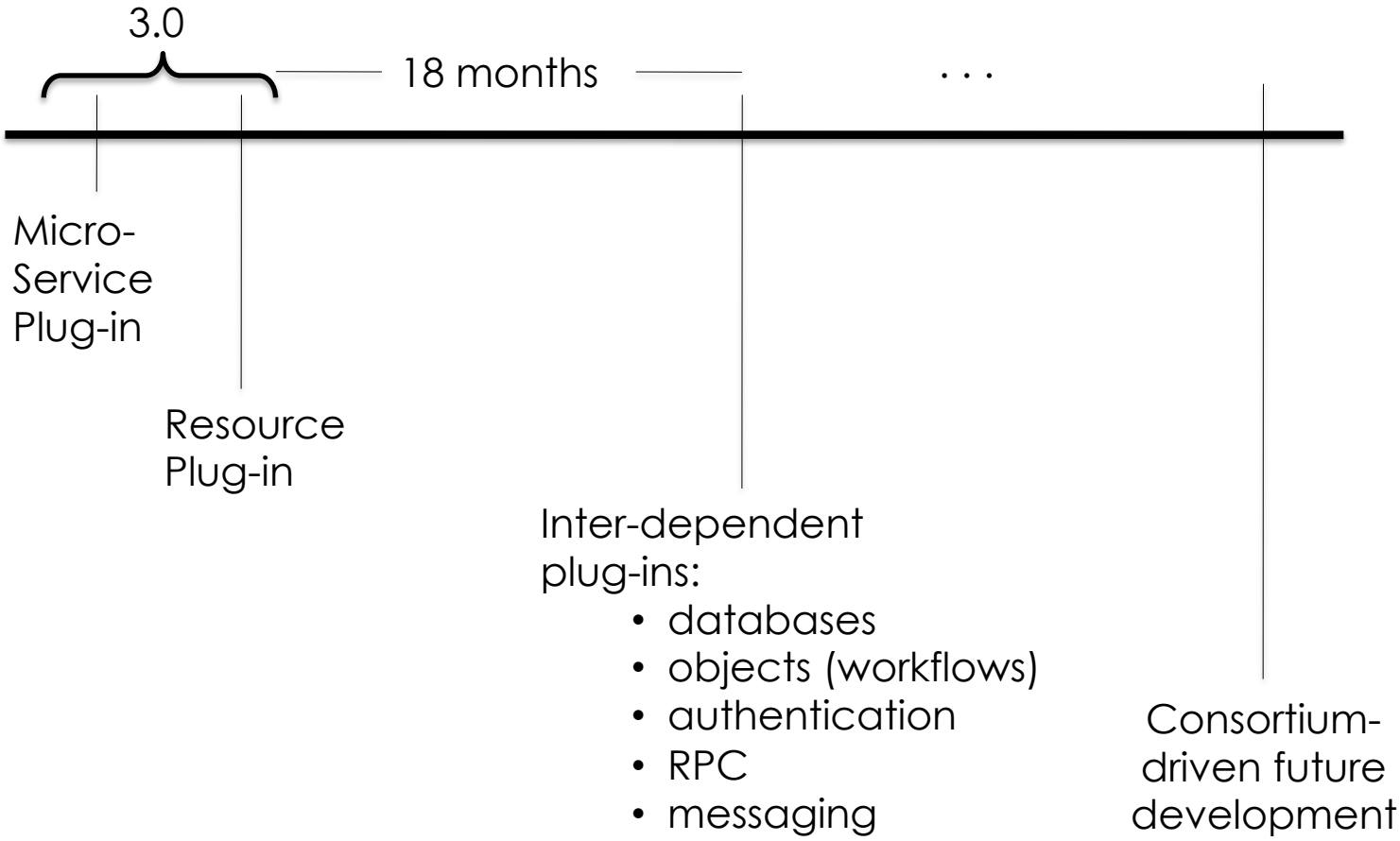
Tree-based approach - horizontal and vertical extensibility



Parent-child relationships between resources

- Generalizes iRODS resource groups
- Operations introduce policy at the storage level
- To be included in E-iRODS 3.0 with smart resources for HSM, round robin, replication, load balancing

Anticipated Timeline for Future Development



E-iRODS Consortium

Summary

- Membership dues will fund basic E-iRODS development; research funding will keep iRODS development going
- E-iRODS remains completely open source (binary and source code)
- The Consortium will gather together organizations to
  - Protect the longevity of the iRODS technology
  - Allow input and coordination on the roadmap
  - Unite organizations with common (data) goals
- Membership levels
  - General
    - Access to release roadmap
    - Privileged access to (paid) consulting and technical support
  - Professional
    - Voting rights to release roadmap
    - Non-voting seat on governing board
  - Sustaining
    - Voting seat on governing board

E-iRODS Consortium

- Contacts:

  Charles Schmitt (cschmitt@renci.org) and
  Leesa Brieger (leesa@renci.org)

- Draft charter document available on request

- We are looking for discussions with potential partners
  - Suggestions and ideas welcome
  - To reach consensus before formal establishment of the consortium

renci