jargon.update();

iRODS User Meeting 2013

Garching, Germany

Mike Conway - DICE (www.irods.org)

michael_conway@unc.edu

Topic

Let’s talk about client development.

‘Client’ includes API for both Java and PHP, and also includes interfaces for the desktop and webtop, as well as mobile devices.

Desired Topics?

Items

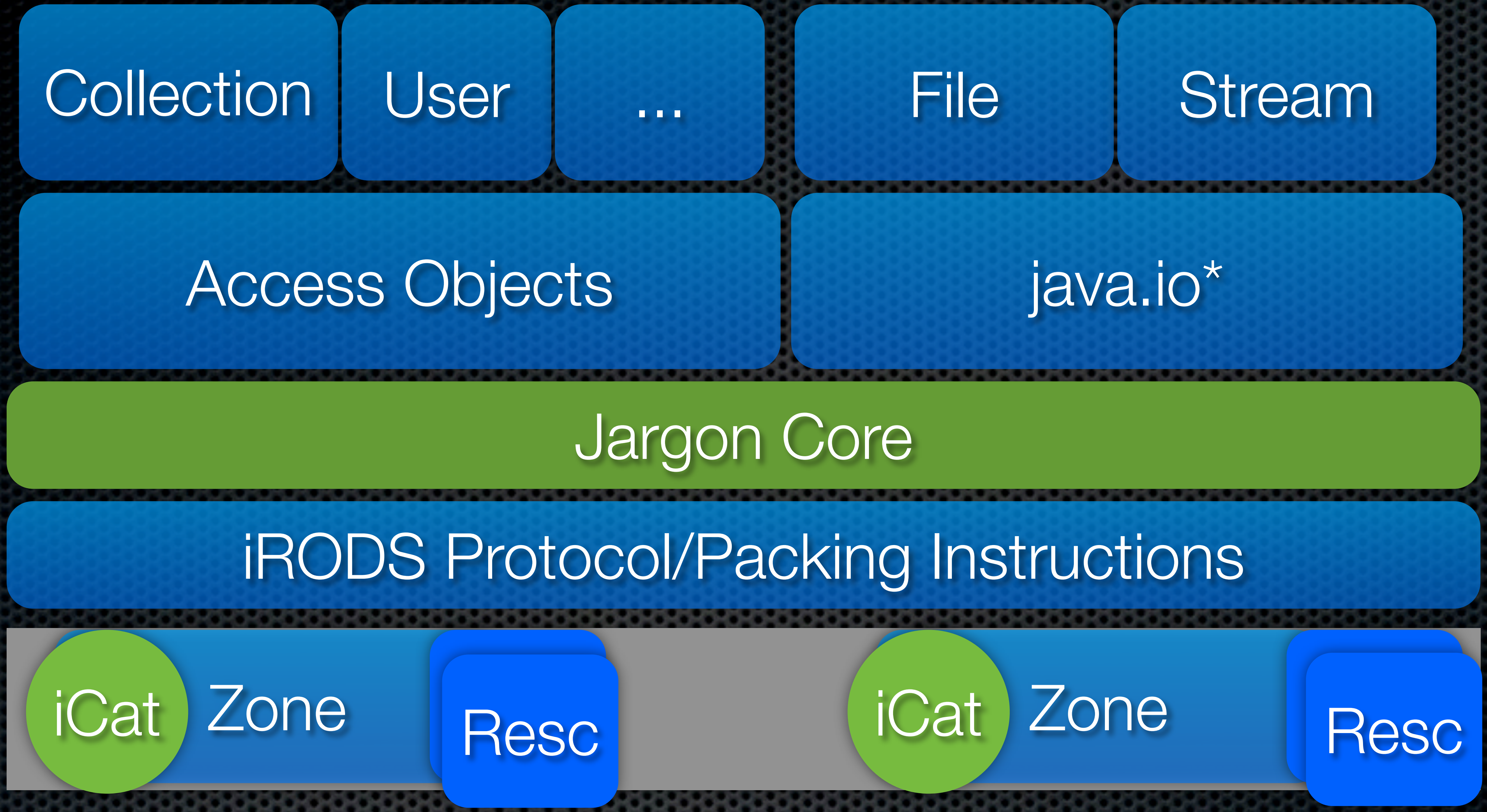
- Current State
- Strategy
- Release Highlights
  - Jargon 3.0 and iDrop 2.0
- Roadmap
  - Jargon
  - iDrop
  - PHP

Current State

- Jargon 3.3 is released and recommended for all versions of iRODS.
- iDrop 2.0 is released and recommended for all versions of iRODS.
- Jargon ‘classic’ (Jargon Trunk) is at EOL and will only get bug fixes.
- PRODS/PHP API is alive and kicking and getting community attention.

Jargon Core

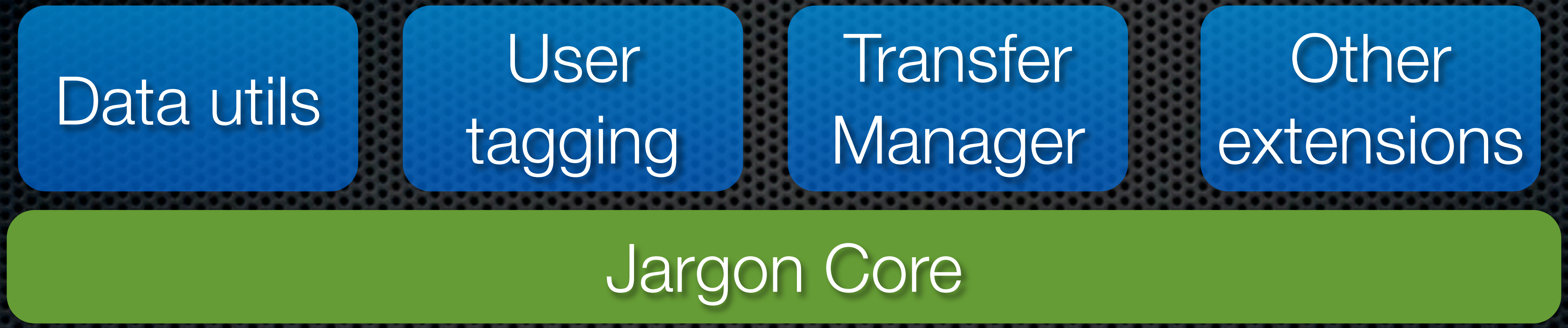
- Jargon is the current client API. Jargon-core is the main library
  - Low level implementation of iRODS protocol, which you should never see
  - ‘Public’ API with POJO’s in and out that represents the ICAT domain model as well as core iRODS services



Jargon Core in the stack

Jargon is a driver for the grid, and should intuitively present iRODS services across federations.

Everything above core can be tested by using mocks!



Data utils

User tagging

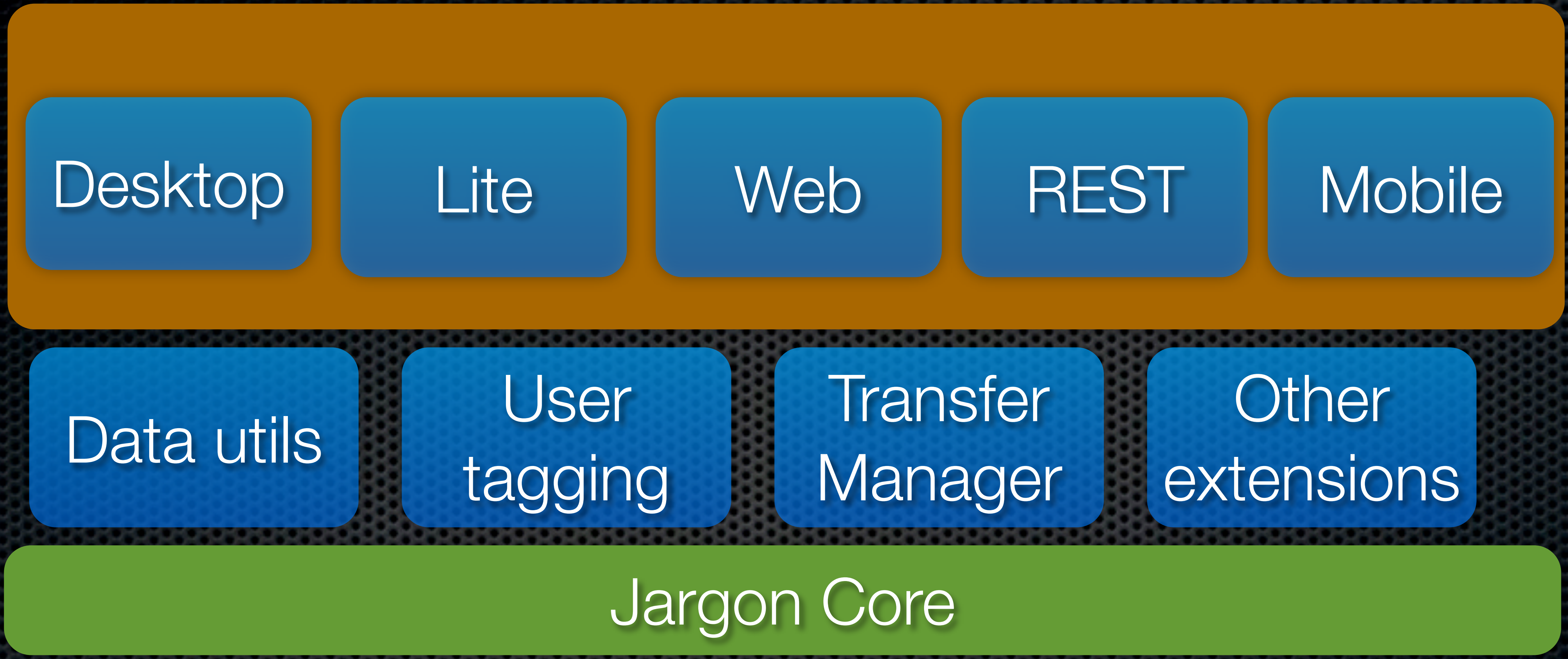
Transfer Manager

Other extensions

Jargon Core

Jargon services above core

Jargon-core is used to develop higher level services. Keep the core as small as possible!



Desktop

Lite

Web

REST

Mobile

Data utils

User tagging

Transfer Manager

Other extensions

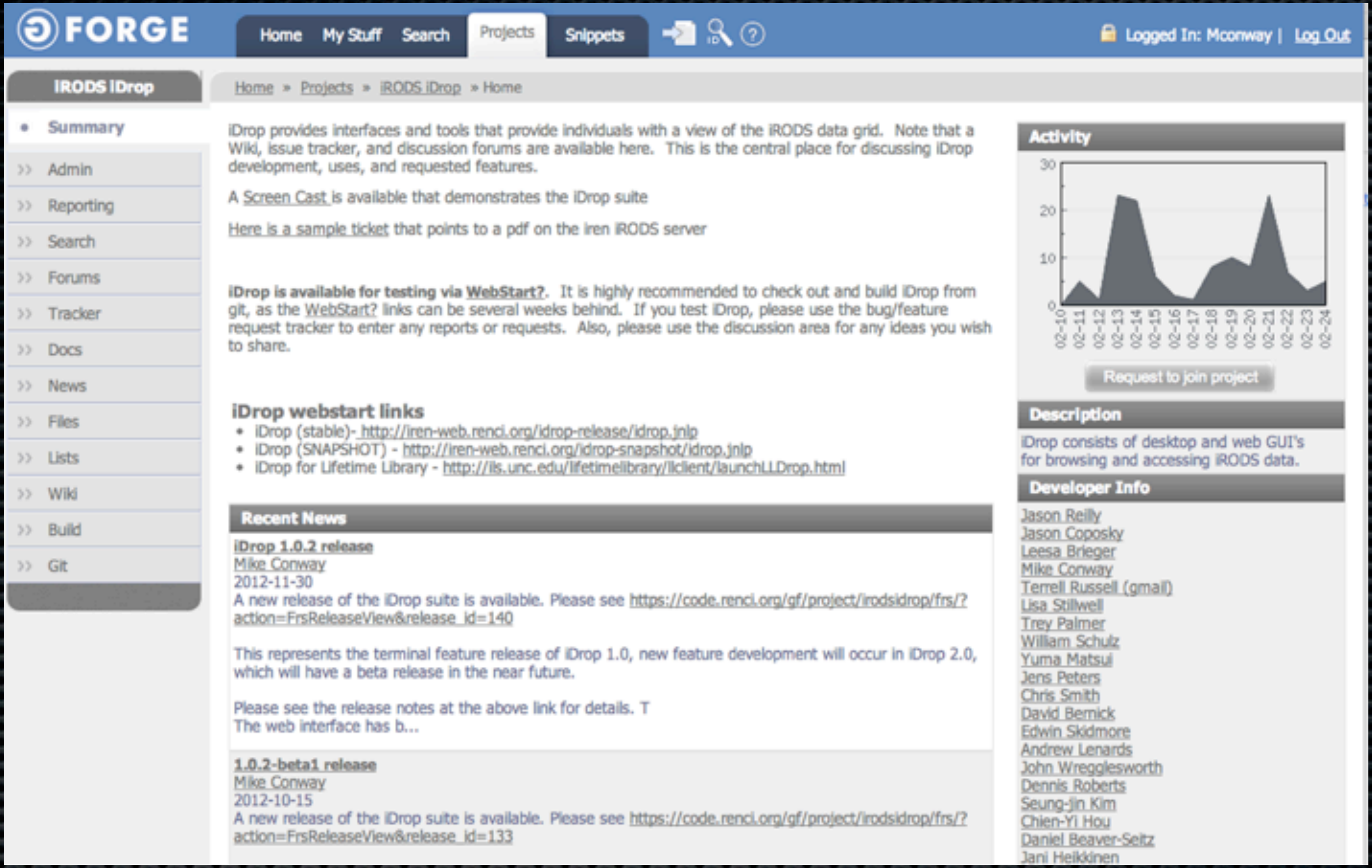
Jargon Core

iDROP

iDrop is a suite of interfaces and web API that provide a supported, out-of-the-box set of clients.

Release!

- Jargon 3.3.0 Feature Release
  - Lots of features and fixes to support iDrop Web2
- iDrop Suite 2.0 (Phase I)
  - Swing GUI redesign (with iPlant) Phase I
  - Web interface redesign
    - Twitter Bootstrap



iDrop provides interfaces and tools that provide individuals with a view of the iRODS data grid. Note that a Wiki, issue tracker, and discussion forums are available here. This is the central place for discussing iDrop development, uses, and requested features.

A Screen Cast is available that demonstrates the iDrop suite

Here is a sample ticket that points to a pdf on the iren iRODS server

iDrop is available for testing via WebStart?. It is highly recommended to check out and build iDrop from git, as the WebStart? links can be several weeks behind. If you test iDrop, please use the bug/feature request tracker to enter any reports or requests. Also, please use the discussion area for any ideas you wish to share.

iDrop webstart links

- iDrop (stable)- http://iren-web.renci.org/idrop-release/idrop.jnlp
- iDrop (SNAPSHOT) - http://iren-web.renci.org/idrop-snapshot/idrop.jnlp
- iDrop for Lifetime Library - http://ils.unc.edu/lifetimelibrary/llclient/launchLLDrop.html

Recent News

iDrop 1.0.2 release
Mike Conway
2012-11-30
A new release of the iDrop suite is available. Please see https://code.renci.org/gf/project/irodsidrop/frs/?action=FrsReleaseView&release_id=140

This represents the terminal feature release of iDrop 1.0, new feature development will occur in iDrop 2.0, which will have a beta release in the near future.

Please see the release notes at the above link for details. T
The web interface has b...

1.0.2-beta1 release
Mike Conway
2012-10-15
A new release of the iDrop suite is available. Please see https://code.renci.org/gf/project/irodsidrop/frs/?action=FrsReleaseView&release_id=133

Description

iDrop consists of desktop and web GUI's for browsing and accessing iRODS data.

Developer Info

Jason Reilly
Jason Coposky
Leesa Brieger
Mike Conway
Terrell Russell (gmail)
Lisa Stillwell
Trey Palmer
William Schulz
Yuma Matsui
Jens Peters
Chris Smith
David Bernick
Edwin Skidmore
Andrew Lenards
John Wregglesworth
Dennis Roberts
Seung-jin Kim
Chien-Yi Hou
Daniel Beaver-Seitz
Jani Heikkinen

Projects and Code

http://code.renci.org

Codebase available via GIT and Maven

Demos of iDrop

Jargon Core Roadmap

- 3.3.1 tune up release in 45 days
    - Tune ups for ‘listing’ methods needed
- 3.3.2 feature release 90 days
    - Mounted collections
    - Workflows and MSOs
    - Kerberos
    - Expanded GenQuery support
- 3.3.3 Optimization and Refactoring only release
    - JMeter automated benchmarks against eIRODS grid

iDrop Roadmap

- iDrop Swing GUI Phase II
  - Grid Memory
  - New ‘transfer engine’ with restructured queue manager functions and better management of restarts
  - Client side rule engine

iDrop Keystore

- “Keystore” functionality can remember all grids accessed, and can pre-seed grids with deployment.

iDrop Transfer Database

- Adjust local database
  - Transfer
    - Transfer Attempt
      - Transfer File

Client Side Action Engine

- Transfer Callback mechanism exists to provide hooks
  - Before transfer
    - Before file
      - Within file
    - After file
  - After transfer

Client Side Action Engine

- Attach handlers to events by host/zone/action
  - a.host.name:zone:put
  - another.host.name:zone:put
- Handlers implement interface that
  - injects context
  - provides condition hook
  - provides execution hook

iDrop new components

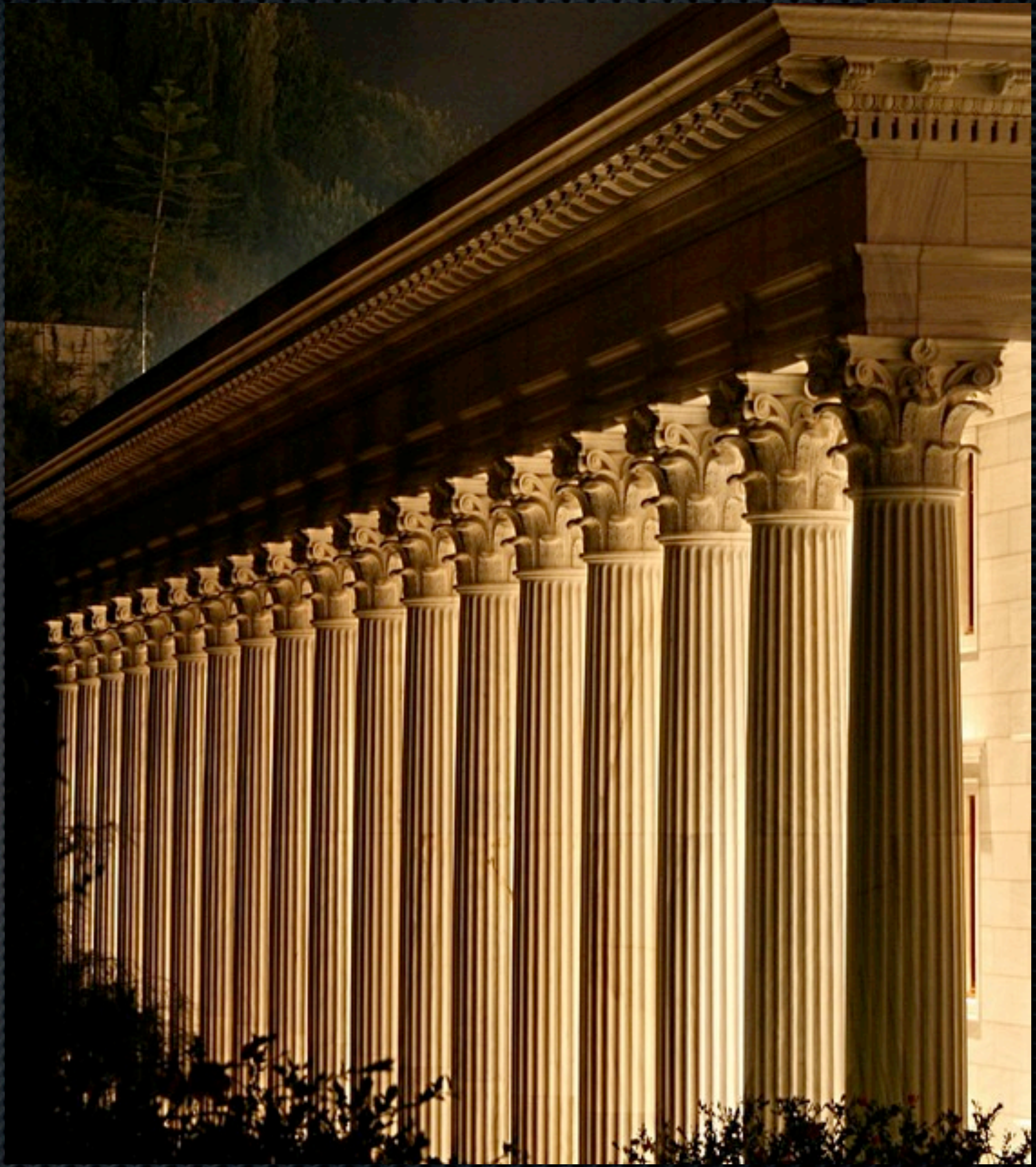
- iDrop new components
  - iDrop-REST - basic put, get, metadata, rule functions for files and collections. This has been prototyped using Spring REST services
  - iDrop-Mobile - client of REST API. We’ve been evaluating PhoneGap for this effort.

PHP and PRODS

- PHP is not going away, getting community support
  - Tickets
  - PAM
- PHP efforts need to focus on the SUBSET of capabilities required for PRODS and eventual Drupal/ Wordpress module support

iRODS is a platform

- The challenge is to view iRODS as a platform for application development
  - Grid
  - Catalog
  - Metadata
  - Policy



Future Imperatives

- Going forward, these are strategic considerations

Client feature development needs to be more closely linked to server feature plans

A server feature is not a feature unless clients can access it

Client communication patterns are changing - one connection per operation giving way to multiple operations per connection

Discovery and Introspection of heterogeneous capabilities on a grid is required

New classes of metadata?

Client operations need to migrate to server

Migration to Protobuf? Per invocation protocol specification

Hot-spot detection and migration/resolution. Evolution of new API on server, via Protobuf?

Efficient transfer of row/ column data a-la TDS?