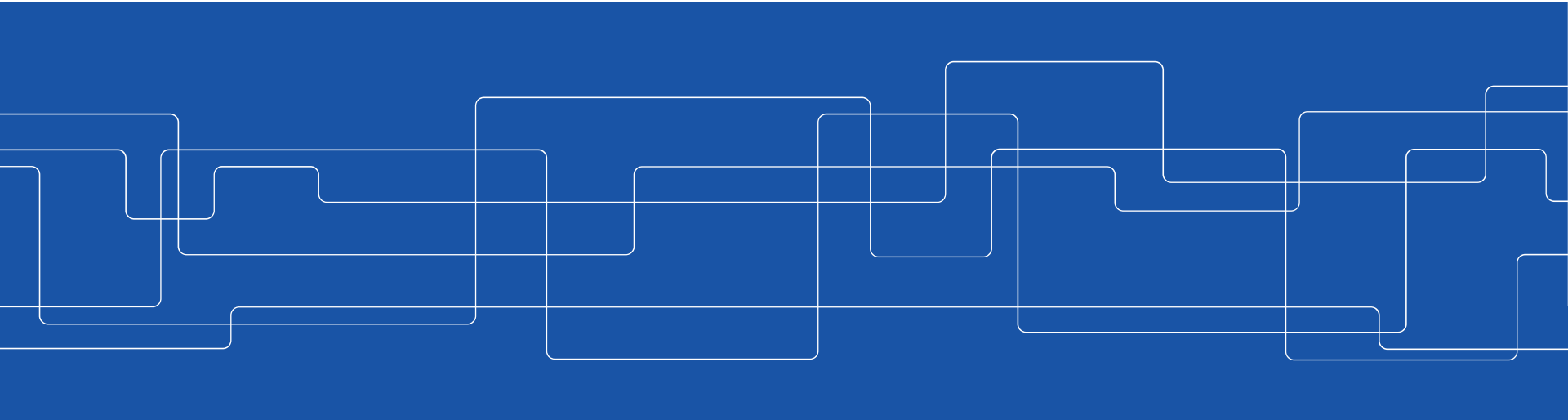




Status and Prospects of Kanki: An Open Source Cross-Platform Native iRODS Client Application

Ilari Korhonen
KTH PDC Center for High Performance Computing
iRODS UGM 2016, June 9th 2016





About Project Kanki

- Cross-platform iRODS client application for Linux and Mac OS X (currently...)
- Available as open source with a 3-clause BSD license
 - <https://www.github.com/ilarik/kanki-irodsclient>
- Released by University of Jyväskylä, Finland (my previous employer until May 31st 2016)
- As I started at KTH PDC June 1st 2016, I will now continue the development with my new employer



About KTH PDC

- Supercomputing center based at the KTH Royal Institute of Technology in Stockholm, Sweden
- PDC = Parallel Dator Centrum
- Largest supercomputing site in the nordic countries
- Services available for researchers thru SNIC
- SNIC = Swedish National Infrastructure for Computing
- I will be working with the SNIC national iRODS project



Releases of the Kanki iRODS Client

- First release was made September 11th 2015 in source form at GitHub
- Since then 9 releases have been made
- Also, since version 1.0.3 I have been distributing binary releases at GitHub for CentOS/RHEL and for OS X since version 1.0.7
- Latest release is version 1.1.0b1 released Nov 16, 2015
- Bleeding edge can be cloned from the develop branch



Some Features of the Kanki iRODS Client

- Basic iRODS object operations in the GUI:
 - Put, get, delete, rename, move, create collection
- Put and get are recursive of course and running on separate threads for multiple background transfer jobs
- Rule exec queue display
- Schema-wise configurable metadata editor
- Search of iRODS objects based on arbitrary criteria



Some Features of the Kanki iRODS Client

- Support for all the iRODS authentication modules on Linux and Mac OS X
- Support for SSL and TCP transport modes
- Also some miscellaneous features (selection of storage resource, possibility to enable or disable verification of checksums on put and get against the iCAT stored hashes if available)
- Error log which captures all iRODS API errors



Prospective Features (see conference paper for details)

- Synchronization of iRODS collections with local directories
- Access Control List editor for iRODS objects
- Schema validation in the metadata manager
- Rule submission for execution into the grid
- Exhaustive rule engine execution queue management
- Data grid analytics with visualization views (VTK!)



Build and Release Workflows

- The build and packaging process can and has been fully automated (via Jenkins CI previously at JYU)
- Linux builds can be done in disposable LXC containers launched from purpose-built Docker images
- OS X builds on can also be done via Jenkins CI jobs, which build irods-icommands and the iRODS client app into an OS X installer package (icommands bundled)
- So far the releases have been built with Qt qmake

Build and Release Workflows

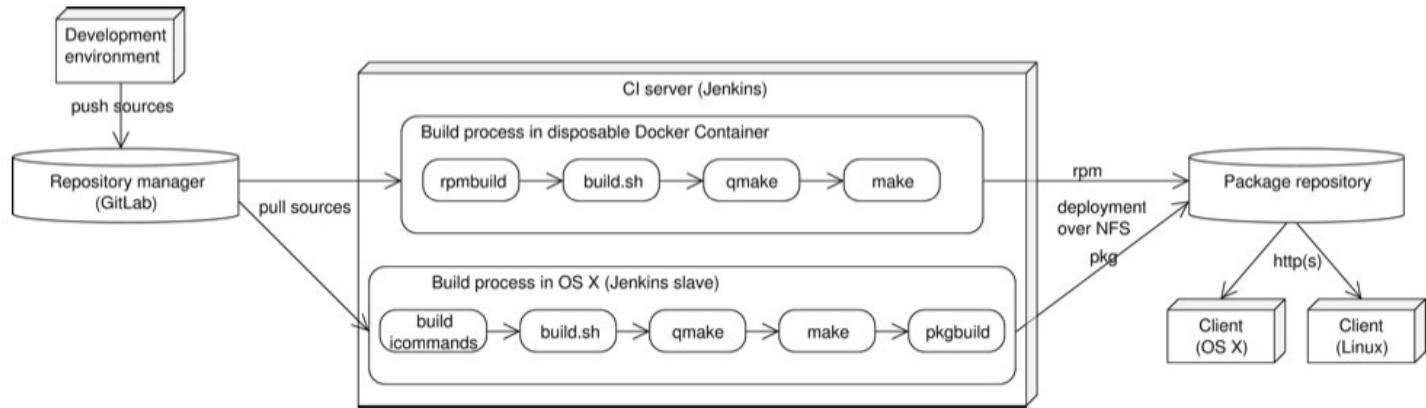


Figure courtesy of Miika Nurminen (from our iRODS UGM 2016 conference paper)



Build and Release Workflows

- Currently a single C++11 codebase, builds with gcc and clang
 - OS X builds bundle icommands and plugins into iRODS.app
 - CMake build environment in process (works on OS X)
 - iRODS/Qt/VTK integration done as a proof-of-concept
-
- Stay tuned for the next release (1.1.0) !



Demo and Questions

- Contact Information:
 - Ilari Korhonen (PDC Center for High Performance Computing, KTH Royal Institute of Technology, Stockholm, Sweden)
 - E-mail: ilarik@kth.se or ilarik@pdc.kth.se

Thank you for your interest !