

Paul Nilsson

---

# PILOT 2.0

# Introduction

---

- New PanDA Pilot Project launched in April 2016
  - Project to span the next few years
  - Presented at PanDA workshop in April, WFMS review in May, TIM in June (plus other presentations in ADC and Pilot Dev meetings) [For more info..]
  - Development and support of the old pilot (“Pilot 1.0”) will continue as it remains the production pilot until new pilot (“Pilot 2.0”) is ready
- Motivation
  - Some of the Pilot 1.0 code base is getting a bit too old and is difficult to maintain
  - Refactoring is a slow process that has already been going on for years and does not always have highest priority
  - More manpower made available to alleviate a steady increase of feature requests
  - New features/workflows are often challenging to implement/support
    - E.g. experiment as a plug-in, glExec, failovers, object stores, event service, HPC:s, ..
- “Complete” rewrite
  - Keeping some recent new developments (not cut-and-paste)
  - Getting rid of all legacy code and outdated mechanisms
  - Rethink of basic pilot flow

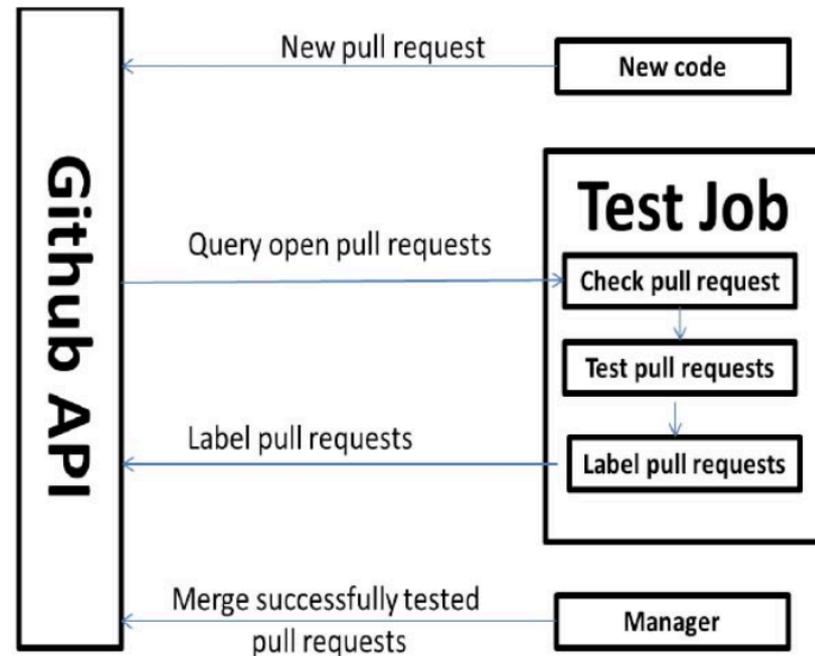
# Documentation and Code Repository

---

- Goal: Pilot 2.0 should be “completely” documented
  - Auto-documentation of code
    - Documentation tools to be evaluated (e.g. sphinx, pydoc, doxygen)
  - General overview, major workflows and algorithms on wikis
    - General Pilot 2.0 wiki:  
<https://twiki.cern.ch/twiki/bin/view/PanDA/Pilot2> (in progress)
    - Site movers wiki:  
<https://twiki.cern.ch/twiki/bin/view/PanDA/SiteMovers> (also covering Pilot 1.0)
- Pilot 2.0 repository already located in GitHub
  - Dev branch: <https://github.com/PanDAWMS/pilot-2.0>

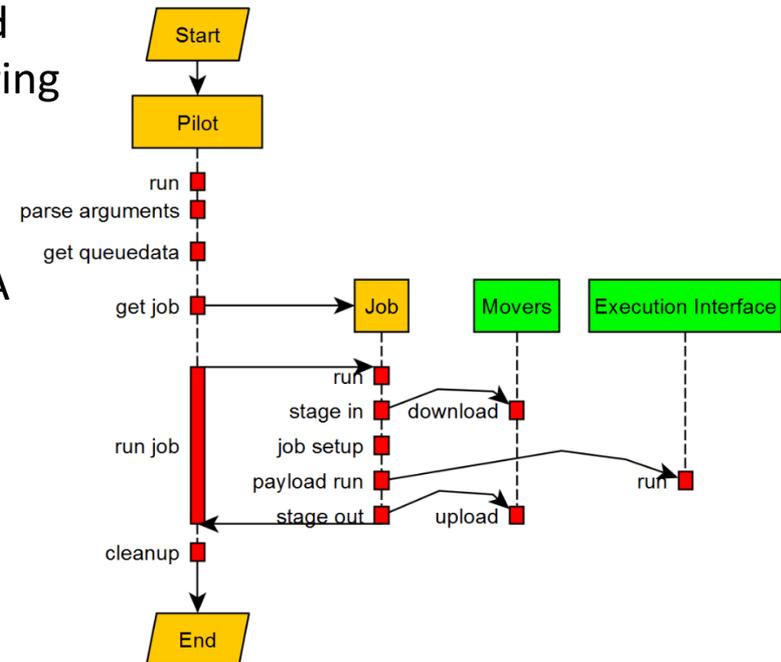
# Testing Framework

- A pull request into Pilot GitHub will trigger a series of tests that need to succeed before release can be approved
- Use pattern validation to ..
  - Avoid typing errors
  - Keep code clean
  - Follow coding convention
- Unit tests are added to pilot for new functionalities
- Automatic test workflow
  - Full test of all test functions
  - Pull requests are labelled “OK” or “FAIL” automatically with API
  - For “FAIL”, add failure message to GitHub with API
- Manager merges successfully tested pull requests
- Dev and RC pilot will automatically be created (by cron job - currently manually created)



# MiniPilot

- A minimal pilot has been developed by Daniel Drizhuk (Kurchatov Inst.)
  - First Pilot 2.0 code to have been developed
  - To be used by the developers primarily during the initial development and testing stage
  - For module and component testing
  - Can eventually result in a SimplePilot for external use / starting point for new PanDA users
- Documentation/instructions in GitHub
  - <https://github.com/PanDAWMS/pilot-2.0/tree/dev/lib/minipilot>
- Easy to use by design
- Code is clean and simple
- Using proper/standard [python] logging
- Following coding conventions
  - Enforced by testing framework



# Site Movers

---

- A new site mover architecture in development since summer of 2015
  - Implemented in “Pilot 1.0” but will largely survive in Pilot 2.0
  - Major development effort and a great simplification (using new AGIS functionality leading to removal of several complex schedconfig fields [to be removed after site migration])
  - Essentially ready, major copy commands supported (rucio, xrdcp, lcg-cp, lsm, dccp)
    - Standing by with GFAL2 and OS site movers for non-Rucio using experiments
  - Some functionality not implemented yet (e.g. alt. stage-out to nucleus site) or tested (object store transfers, to be handled by rucio)
    - Currently being discussed
- Migration plan for sites to use new site movers is in progress
  - Validation of DDM endpoints for support of particular protocols
    - Most endpoints already tested and verified with stand-alone script
  - Validation of content of information system
  - Validation of mover implementation
  - Corner case study
    - Validation of opportunistic resources and specific storage implementations
  - Post-migration clean-up
    - AGIS and schedconfig cleanup and optimization, retirement of outdated pilot code
  - Many site admins have already been contacted
  - Site migration responsible: Danila Oleynik (UTA)

# Component Model

- Recent F2F discussion based on Pilot 1.0 workflows resulted in first draft of Pilot 2.0 component model
- Comparisons with other models to be done
  - E.g. Radical Pilot (S.Jha et al)
- First version of Pilot 2.0 code structure based on component model already created and available in GitHub
  - For initial testing of workflows, testing system
  - Currently only skeleton code
- Component Model and Architecture document to be written

Job recovery
- find unstaged files
- stage-out files
- cleanup

Benchmark
- evaluate execution environment speed

Pilot setup
- establish signal handling
- retrieve queuedata
- setup local environment

Sandbox setup
- establish local security
- retrieve hardware, OS and HPC data
- check memory limits

Job control
- get jobs from global scheduler or local scheduler
- validate job definition versus available resources

Event control
- get event ranges
- update events

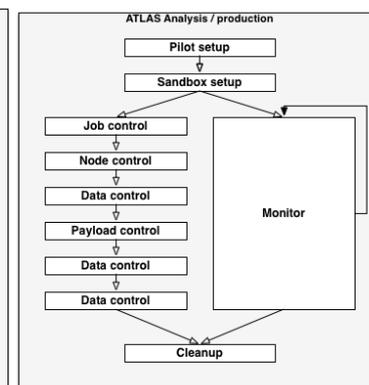
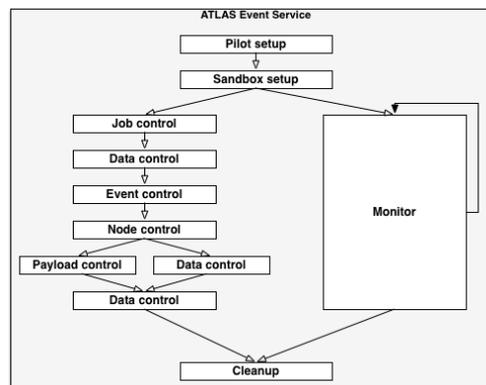
Cleanup
- controller cleanup
- pilot cleanup

Node control
- setup accounting
- payload preparation
- payload submission
- payload inter process communication
- payload verification

Monitor
- monitor payload progress
- measure memory
- measure CPU
- send heartbeat to server

Payload control
- setup accounting
- payload setup
- payload submission
- payload execution
- output verification

Data control
- setup accounting
- stage-in
- stage-out



# Harvester Integration

---

- Harvester can simplify some of the workflows in the pilot
  - Logic can be moved away from pilot to Harvester
    - Will simplify pilot code
  - Harvester can decide which pilot modules to use
    - Especially useful on HPC:s where only a reduced set of pilot components are needed
      - E.g. multi-jobs on HPC:s; getJob functionality not needed
- The modular approach of Pilot 2.0 will facilitate the integration with Harvester
  - Note: Pilot 1.0 is also modular and Harvester will initially launch components (HPC code) from ‘old’ pilot
- Details are being discussed
- See slides from Tadashi for many more details

# Timeline

---

- MiniPilot ready for use by pilot developers (incl. documentation)
  - Presented at Grid 2016 conference (D. Drizhuk)
- Testing system currently being designed (implementation being tested, designed by Wen Guan, University of Wisconsin)
  - Presented at Grid 2016 conference (D. Drizhuk)
- Functional requirements have been written down for most workflows
- Component model discussion done in July 2016
  - F2F meeting with five of the developers present
- Design phase initiated in July 2016
  - Skeleton Pilot 2.0 components created for first design iteration
  - Component model and architecture document in ~six months (from now)
- Implementation and testing phase for basic functionalities
  - ~Six months (i.e. in ~one year from now)
- Final product for original requirements
  - Including additional implementations and extensions
  - ~Two years

# Core Pilot Team

---

Name	Time allocation	Tasks (to be further defined)
Paul Nilsson	25% Pilot 2.0, 75% Pilot 1.0 -> 100% Pilot 2.0	Project leader; general overview and management
Alexey Anisenkov	20-50%	Site movers, RunJob* classes
Daniil Drizhuk	Up to 50%	MiniPilot, RunJob* classes
Mario Lassnig	25%	Site movers (Rucio optimization; ADC Data Management)
Danila Oleynik	>=50%	HPC pilot, event service, general pilot workflow
Wen Guan	>=50%	Event service, HPC pilot, job manager
Pavlo Svirin	(100% ALICE Pilot)	HPC for ALICE integration

+ Collaboration with Shantenu Jha (and two of his Post Docs / Rutgers, Radical/SAGA), Shao-Ting Cheng (AMS) – and there will be others as well

# Organization

---

- Pilot team is spread out over the world, sometimes present at CERN
  - Paul, Wen, Mario, Pavlo stationed at CERN - would be good to have Danila here as well, Alexey and Daniil are in Russia (occasionally at CERN)
  - Vidyo conferences
  - Face-to-face meetings
- Project transparency
  - Report development status frequently (in various meetings)
  - Todo/wishlist wiki ('major' Pilot 1.0 features will also be added here)
  - Meeting minutes [at least after vidyo conferences]
    - Distributed to PanDA mailing list
    - Archived on indico pages (PanDA Pilot Development Meetings)