

# PanDA Pilot Ideas

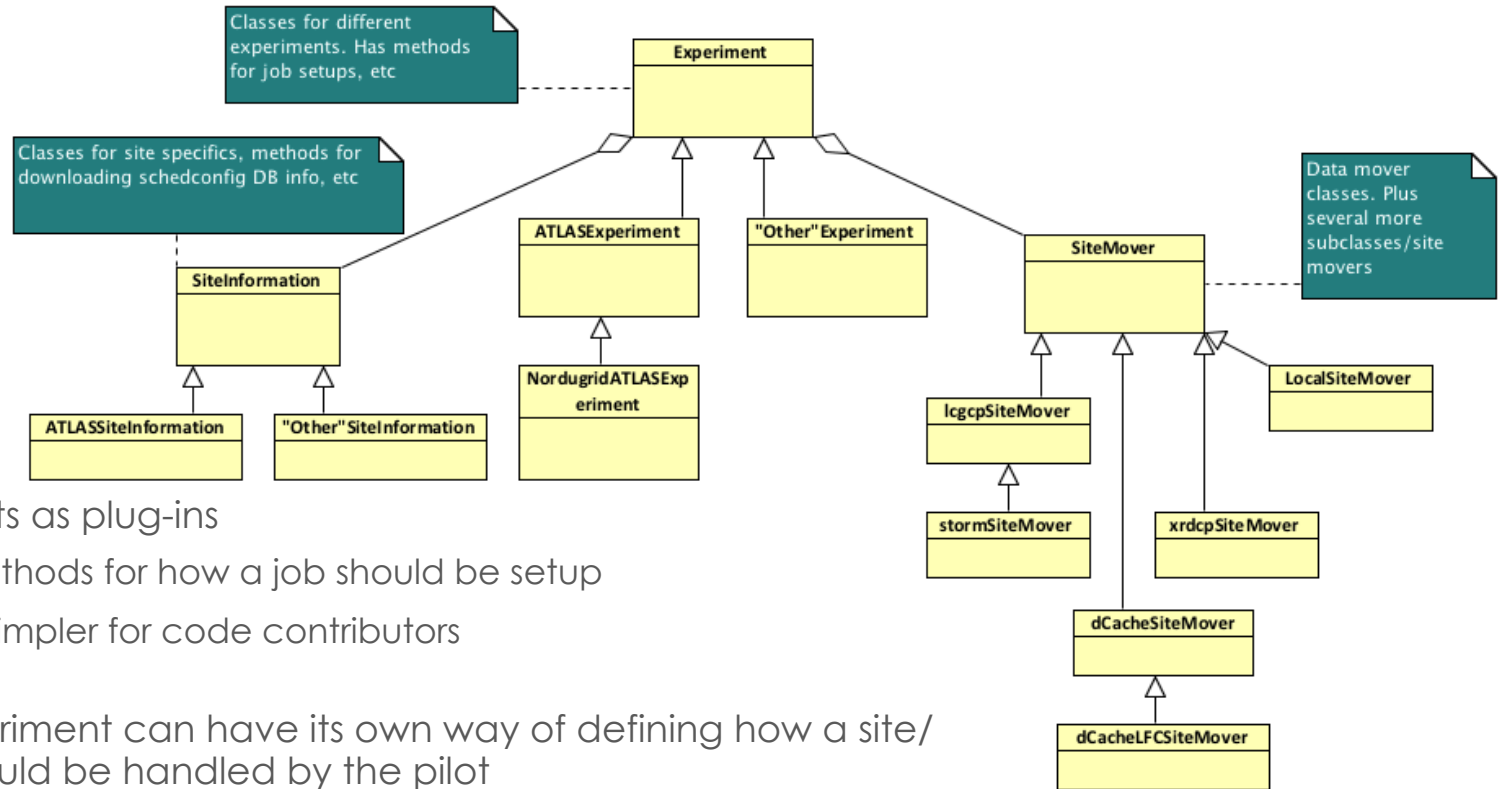
Paul Nilsson

UNIVERSITY OF TEXAS  ARLINGTON

# Next Generation Pilot

- Current pilot: 40k LOC
  - Several modules getting very big, change is needed
  - On-going refactoring process
    - Refactoring of pilot, runJob modules made it possible to “easily” add glExec, e.g.
    - Old unused code is removed
  - Major changes needed
    - Especially useful if pilot is to be used for other experiments
      - Pilot is rather ATLAS specific [although it can run generic jobs]
    - Long term project but changes will appear gradually
    - When the major new classes have been introduced, pilot will be considered ‘next generation’ [Code name: *Picard*]

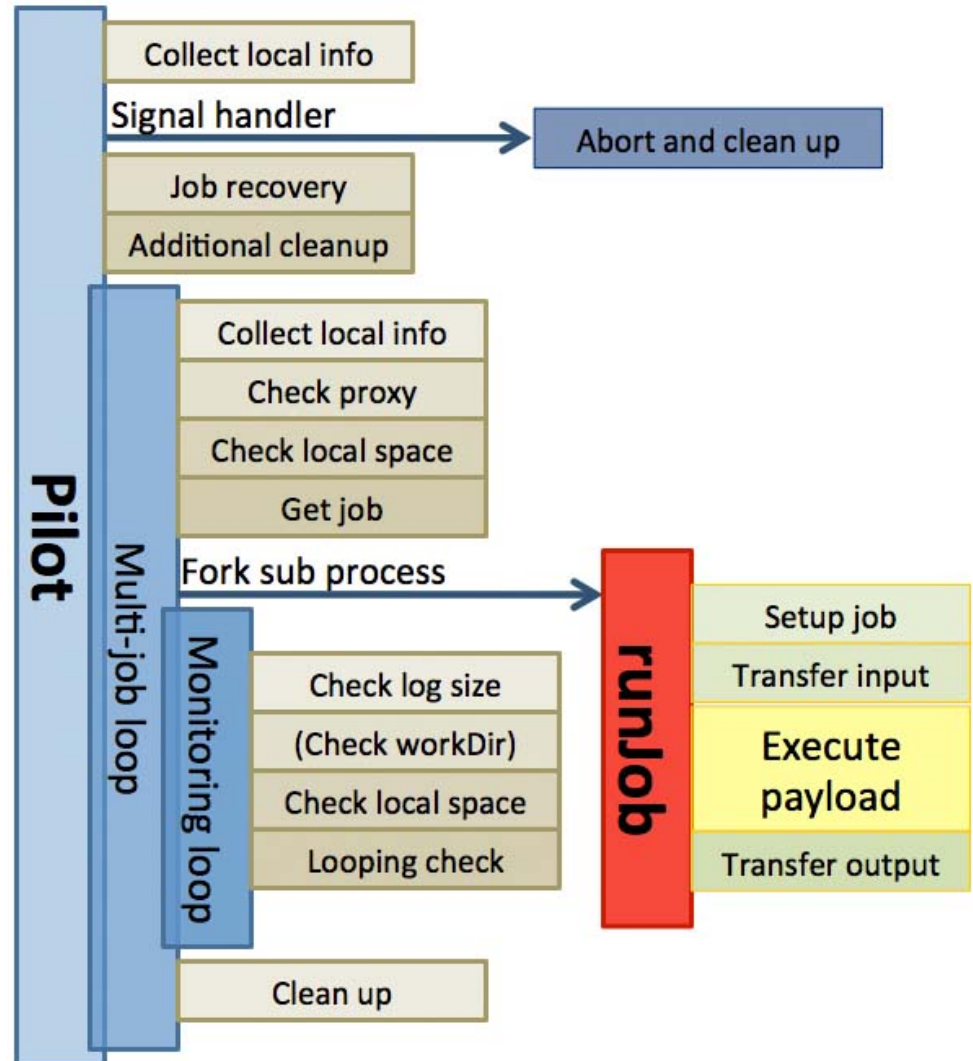
# Diagrams of Major Operations



- Experiments as plug-ins
  - E.g. methods for how a job should be setup
  - Much simpler for code contributors
- Each experiment can have its own way of defining how a site/queue should be handled by the pilot
  - E.g. ATLAS uses queuedata file from schedconfig DB
  - Note: not saying that "other" experiments should not use schedconfig DB
- SiteMover classes as before, but refactored and will take advantage of Experiment (will also know about SiteInformation)

# Pilot Workflow

- Workflow same as before
- Code will be organized in a better way
- The current experiment will be instantiated whenever it is needed
  - Major operations classes are singletons



# Job Recovery Update

- A job that fails during stage-out can leave the output locally
  - Remains will be picked up by later pilot, stage-out reattempted
- New version in development
  - Less complicated than previous version (thus easier to maintain)
  - Development on hold for a long time due to lack of interest
  - UK is showing interest to use it (so far only used in US), will be evaluated
- Job recovery is only used for production
  - Not to be used for user jobs due to the “slow” recovery process
  - Time to revise / refine that idea?
    - E.g. job retries can also be slow
    - What about using job recovery for certain types of user jobs? (long ones, jobs failing the second retry attempt and if the job can't be moved to another site, or even adding a client option `--useJobRecovery` although not many sites offer job recovery at the moment..)

# Stage-in/out Retries

- ❑ Pilot already supports multiple stage-in/out retries
  - ❑ Retry after 5/10 minutes sleep (stage-in / stage-out)
  - ❑ Only useful for certain failures, e.g. file corruption SE->WN (less useful for “real” SE problems since chances are the problem is still there after 5/10 minutes)
  - ❑ Add additional stage-out retry for really long running jobs (after a long break)?
- ❑ Stage-in using alternative replica
  - ❑ Currently max three alternative replicas
- ❑ In development, Federated XrootD: in case of stage-in failure get the file from another site (not only for “missing file” problems)
  - ❑ Discussing details with Wei Yang (SLAC)
- ❑ Stage-in from alternative SE (not using Federated XrootD technology)
  - ❑ In principle already supported by the pilot (via schedconfig.se)
  - ❑ Tuning needed?
- ❑ Stage-out to alternative SE
  - ❑ If stage-out fails at a certain site in a certain cloud, put it at the T1
  - ❑ Some thinking and development needed

# Error Diagnostics

- Currently pilot can identify 100 errors types
  - Should more fine grained stage-in/out errors be identified? (most common error: “stage-in/out failed”)
- “Trf” errors are reported separately from “pilot” errors (interpretation done in PanDA monitor / Bamboo error translator)
  - “Unknown errors” being fixed, several have been identified (I have a long list in my pocket..)
- New ErrorDiagnostics class in development
  - Trying to avoid doing ‘grep’ on stdout (keep to minimum)
  - But why not if the job has failed? Some grepping however is always done
  - Move current error interpreter code to ErrorDiagnostics class
    - E.g. methods for interpreting special errors (e.g. code 40 jobs)
    - Easier for pilot code contributors to simply add a method rather than having to understand where exactly to squeeze in code into pilot/runJob
- DEBUG mode for jobs
  - Next pilot version: server can instruct pilot to start sending tail of trf stdout (+ number of lines, always)
  - Create special command for users (“ppeek”?) that will instruct the server to send the DEBUG command to the pilot during next job update?
  - Tail of stdout will end up on the monitor job page (like log extracts)?

# Additional Useful(?) Features (1/2)

- Pilot source from /cvmfs
  - One less curl to the server [per wrapper]
  - Basically done already by Alessandro
  - Small change in the wrappers needed
- New FileHandler class in development
  - For locally available files
    - Currently pilot looks for DBRelease files in standard cvmfs location and skips stage-in accordingly
  - Generalization of DBReleaseHandler
  - Do the same for FSRelease files (always at the same location as DBRelease files or different?) Any other files?
- Nightlies
  - Support will be checked-in as soon as possible (done in dev pilot)
  - Currently in testing [using homePackage = rel\_6, release = 17.X.0]
- Automatic direct-access (suggested by Hiro)
  - Do copy-to-scratch for 'small' files and direct-access for 'large' files





# Additional Useful(?) Features (2/2)

- Direct-access for production jobs
  - Demonstrated in the past; didn't work last time I tried, some bug fix needed
- Send current “pilot job state” to server with updateJob? (not same as PanDA state)
  - Monitor could display last known job state: starting, setup, stagein, running, stageout, finishing
  - Displayed in pilot log (often difficult for people to tell what the pilot is doing when there is a problem)

## Prod pilot

```
17 Apr 09:40:38| pilot.py | --- Main pilot monitoring loop (job id 1469569460, iteration 87)
17 Apr 09:41:38| pilot.py | ---pollChildren: {'prod': [8884, <Job.Job instance at 0x4998e60>, 5676]}
```

## Dev pilot

```
12 Apr 11:10:41|pilot.py | --- Main pilot monitoring loop (job id 1466577535, state:stagein, iteration 2)
12 Apr 11:11:42|pilot.py | --- Main pilot monitoring loop (job id 1466577535, state:running, iteration 3)
```

- Replace internal time-out module (timed\_command)
  - Known problems with e.g. stdout handling; sometimes the command cannot finish in less than one minute (possibly related to stdout not flushed properly – not necessarily a pilot problem)
  - Current command based on relatively old methods, better technology exists
  - Nice little project for a volunteer.. (I have links to a couple of alternatives)