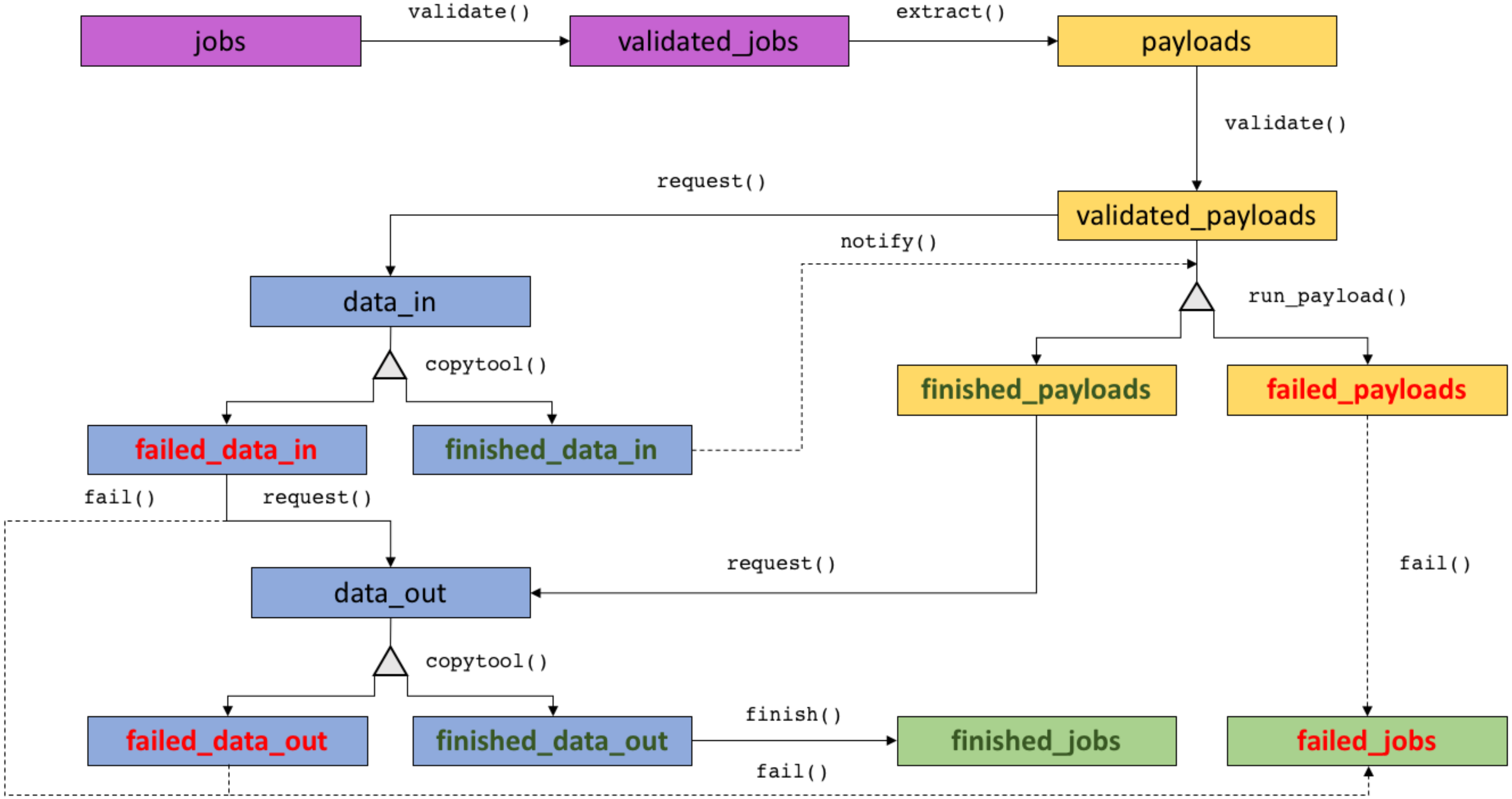
Paul Nilsson

PILOT DEVELOPMENT

Pilot 2 – Internal Queues Updates

- Previous queue implementation had problems with log transfers
- New Job class written, pending code merge
  - Currently job dictionary is moved around the queues, soon job object



Pilot 2 – Error Handling and Reporting

- Error handling has been expanded
  - Currently identifying 12 different error types and can report them to the server when necessary
    - In addition to payload errors reported in job report (also extracted and reported)
    - More error codes+handling will be added in groups

Pilot 2 – Event Service

- Event Service component integrated with main (generic) workflow
  - Pilot automatically uses the relevant code for event service or for normal jobs
- Code merge in progress (with Paul's dev pilot)
  - Additional ES functions migrated from Pilot 1
    - Related to mangling of job parameters for merge jobs
- Testing to begin ~tomorrow

Pilot 2 – Copytools and Information Service

- Copytool developed for gfal-copy
  - In addition to rucio, xrdcp, mv
- To be done: LSM copytool
- Development of Data Component pending delivery of Information Service component
  - Information Service component provides interface to AGIS/schedconfig
  - Work in progress, expect first prototype next week

Pilot 2 – Containers

- If needed, OS to be used for payload should be sent by server along with job specs
  - Otherwise use default image (taken from schedconfig or pilot config file)
- Pilot will always verify that desired copytool is available locally
  - If not, try to use container that has middleware if allowed by site
- Discussing details in biweekly container meetings
  - https://twiki.cern.ch/twiki/bin/view/AtlasComputing/ADCContainersDeployment
- Some coding remains - testing on UK site on OSCER in US

Pilot 2 – Nordugrid and Harvester

- Nordugrid mode
  - Added support for file based job definitions instead of downloading from server
- Harvester mode
  - Job retrieval updated to support Harvester work flow
    - When the pilot is launched, a job definition file for the initial job is present in the launch directory (i.e. same as for Nordugrid)
    - When the job has completed, and if there is time for another job, pilot will place a job request file in the launch directory
    - Harvester will see the job request file and place another job definition file for the pilot to discover
    - (Pilot removes the job request file when the new job definition file is found)
  - Implemented in Pilot 1+2

“HPC Pilot”

D. Oleynik

- Pilot application for managing execution of PanDA jobs on HPC under Harvester control
  - Currently being implemented as stand-alone code using Pilot 2 API – will see later if we can implement is as a dedicated Pilot 2 workflow
- Main functionality:
  - Setup of execution process: prepare environment, perform specific preparation for execution (ex: for OLCF managing of working directories for ATLAS job on RAM disk, etc)
  - Propagate job execution information to Harvester (intercommunication with Harvester)
  - Post processing of execution: cleanup, analyzing of results of execution (exit code, messages)
- Harvester creates working directory for each HPC Pilot job and places json file with job definition in launch directory
- Stage In/Out and zipping of logs are managed by Harvester
- **Work plan**: Finalize debugging of prototype and identify detailed description of required functionality
  - Most of functions already implemented in different places: Pilot 1+2, MiniPilot and HPC MiniPilot -> being moved to/reimplemented in Pilot 2