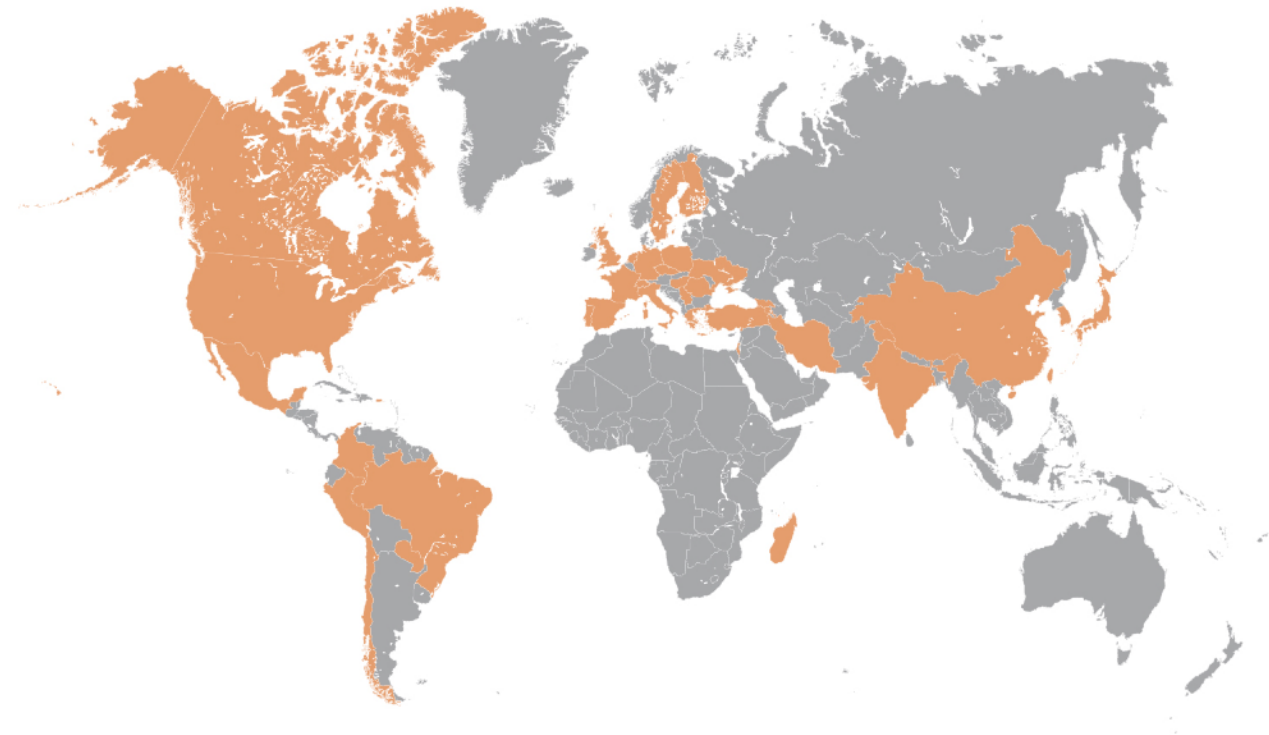


Just-in-time workflow matching for DUNE

Andrew McNab
for the DUNE Collaboration

European HTCondor Workshop
September 2023

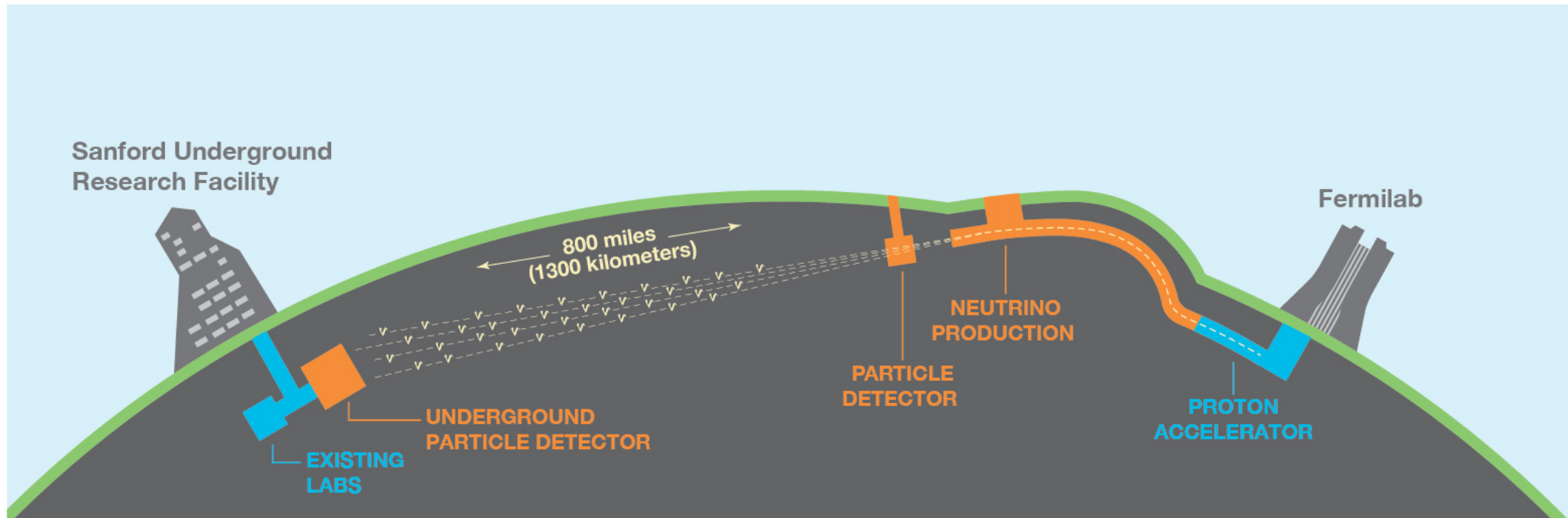


Overview

- DUNE computing context
- Just-in-time workflow design: justIN
- Some key components

What will DUNE be?

- Build huge tanks of liquid argon in a gold mine in South Dakota.
- Send neutrinos from Fermilab to South Dakota.
- Detect what neutrino flavours arrive in the liquid argon.
- Look out for supernova neutrinos and proton decays at the same time.



DUNE Computing Requirements

- Raw data will go from South Dakota to Fermilab and then out to other sites
 - Raw data capped at 30 PB/year
 - Annual tape increments will be about double that
 - with disk requirements increasing at a slower rate.
 - These estimates include simulation and derived data
- That's about 10-15% of the current requirements of a large LHC experiment like ATLAS or CMS, in a full data taking year
- So this is very achievable but it is a major increase in scale compared to the previous Fermilab neutrino experiments
 - **In particular, we need to use sources of funding beyond Fermilab and outside the US to reach the necessary scale**

DUNE Computing Environment

- Fermilab neutrino experiments have a long heritage of integrated applications software, data handling, and workload management
- Simulated and real data are processed using the LArSoft framework for liquid argon time project chamber detectors like DUNE and its predecessors.
- Data files have been managed with SAM - which began on the Tevatron in the 1990s.
 - Manages file metadata, file replica locations, and records which files have been processed during a particular processing campaign
 - LArSoft integrates with SAM to get the next unprocessed file from within the application
 - But SAM has become harder to maintain and expand
- Strategic decision to replace SAM file catalogues with:
 - Rucio for replica management and a new Fermilab product, MetaCat, for metadata

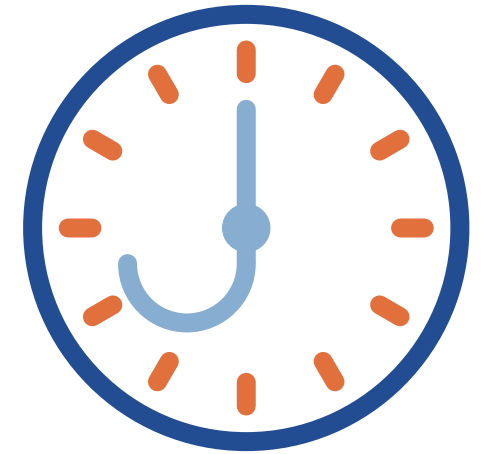


DUNE Workload Environment

- Fermilab has a major investment in HTCondor as its local batch system
 - And in GlideInWMS which allows remote grid (pilot) jobs to join the same HTCondor pool and execute user jobs
- On top of this, Fermilab has developed Jobsub, which now takes the form of customised wrappers around condor_submit, condor_q, etc
- Due to DUNE's scale and specific requirements, in 2022 Fermilab set up a dedicated HTCondor Global Pool for DUNE to allow DUNE-specific configurations
 - This uses HTCondor schedds at Fermilab and offsite: BNL and RAL-PPD so far; CC-IN2P3 and CERN are planned.
- But we still needed a way of managing workflows, that implements SAM's just-in-time file allocation to jobs, but can also generate the jobs and get them to the right sites "near" where the unprocessed files are

Just-in-time workflow system

Just-in-time workflow system

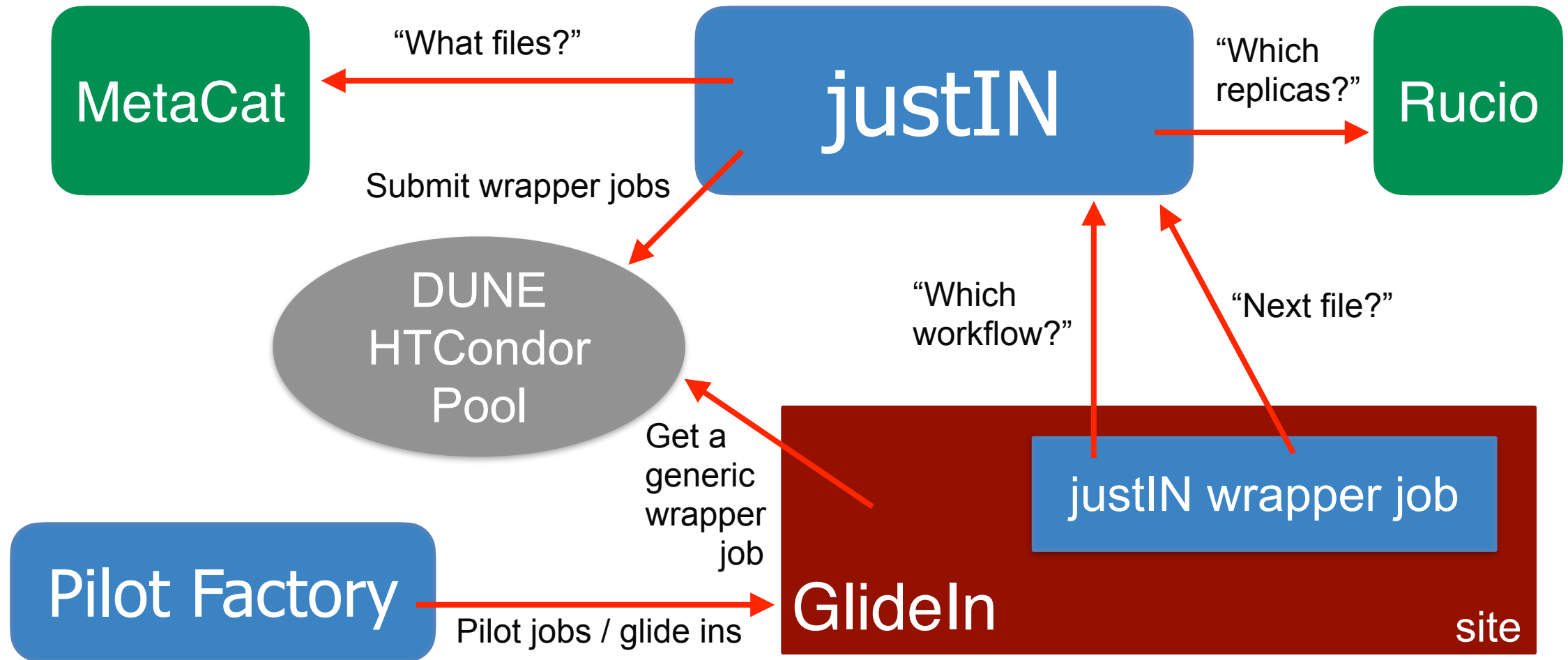


- To satisfy these Rucio/MetaCat/LArSoft/GlideInWMS constraints we designed a just-in-time workflow management system: **justIN**
- Central idea: once we get a job slot at a site, we choose and run a workflow that needs “nearby” unprocessed files, and we feed it the details (URLs, Rucio DIDs, ...) of nearby unprocessed files to work on.
- We have a central database that caches workflow requests from users, relevant Rucio file locations, and information about sites and storages
 - Agents update this information as necessary
- justIN job factory submits jobs to the DUNE HTCondor Global Pool
- justIN services respond to queries and tell the jobs what to do
- Web and command line interfaces allow submission/monitoring of workflows

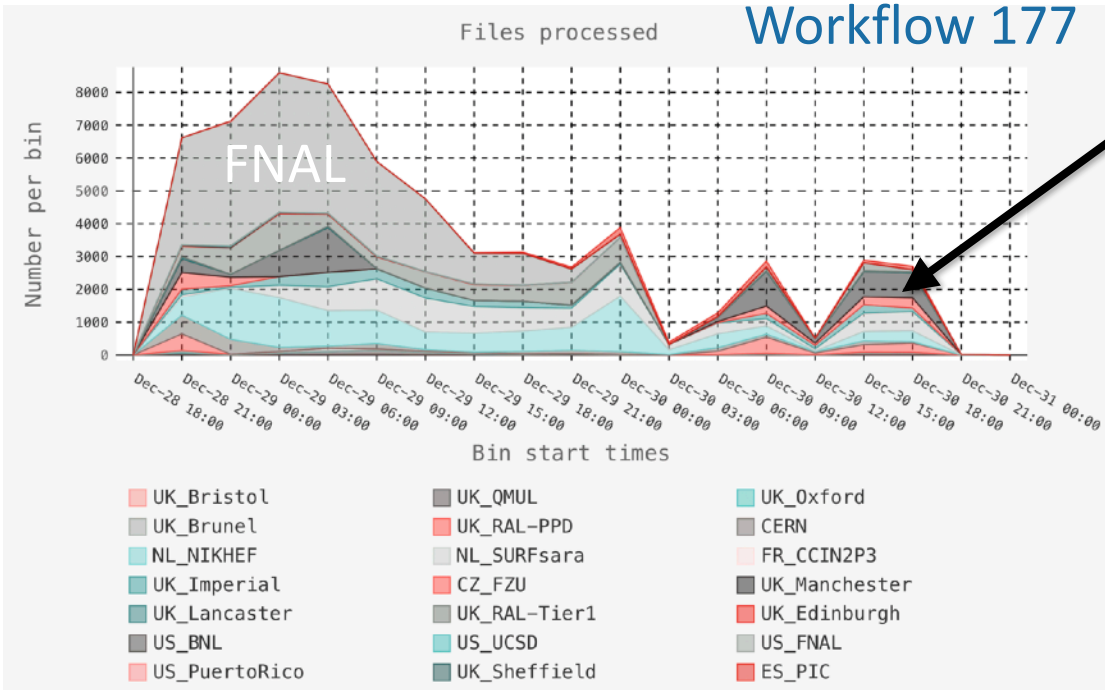
justIN workflows and stages

- Users (including production team) submit requests to run workflows to justIN
 - Workflows consist of one or more stages, with output files from one stage available to the next stage via Rucio-managed storage
- A MetaCat Query Language expression defines the inputs of the first/only stage
 - `"files from dune:all where core.run_type='dc4-vd-coldbox-bottom' and dune.campaign='dc4' limit 10"`
- For each stage, the user defines
 - Job requirements of memory, processors, duration and maximum network “distance” to storage with input files
 - A list of preferred output storages to try
 - A jobscript to be run — in containers using the same Apptainer image at all sites

justIN with generic wrapper jobs

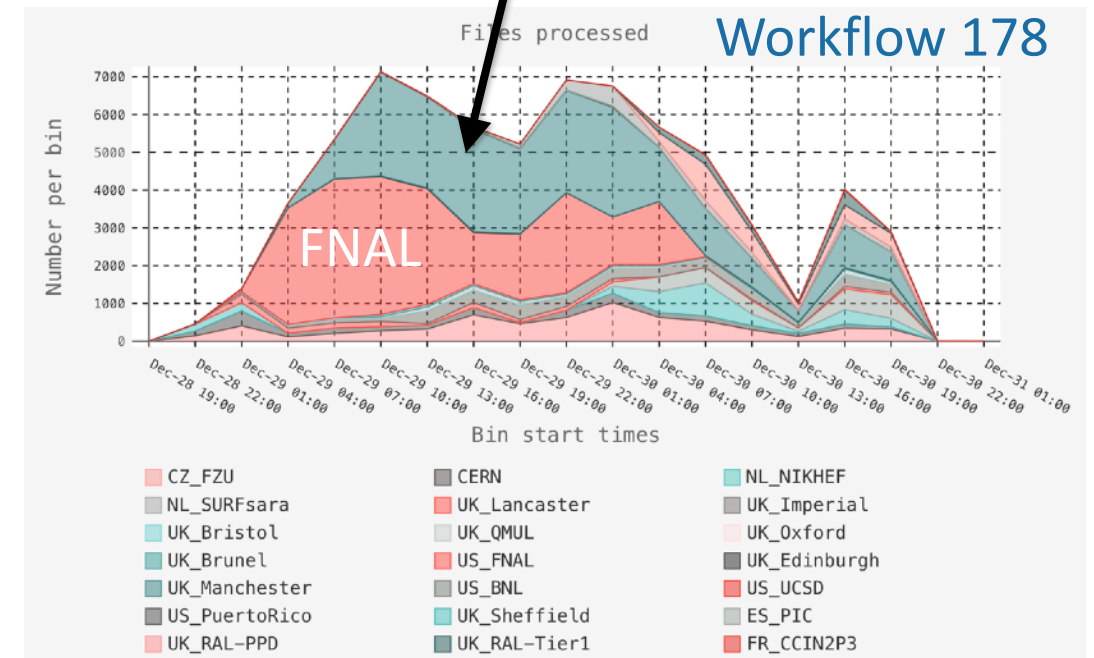


justIN in DUNE Data Challenge DC4



Not much Manchester

Lots of Manchester!



Job allocations follow the data to suitable sites, using replica locations originally from Rucio

Key components of justIN

justIN wrapper jobs

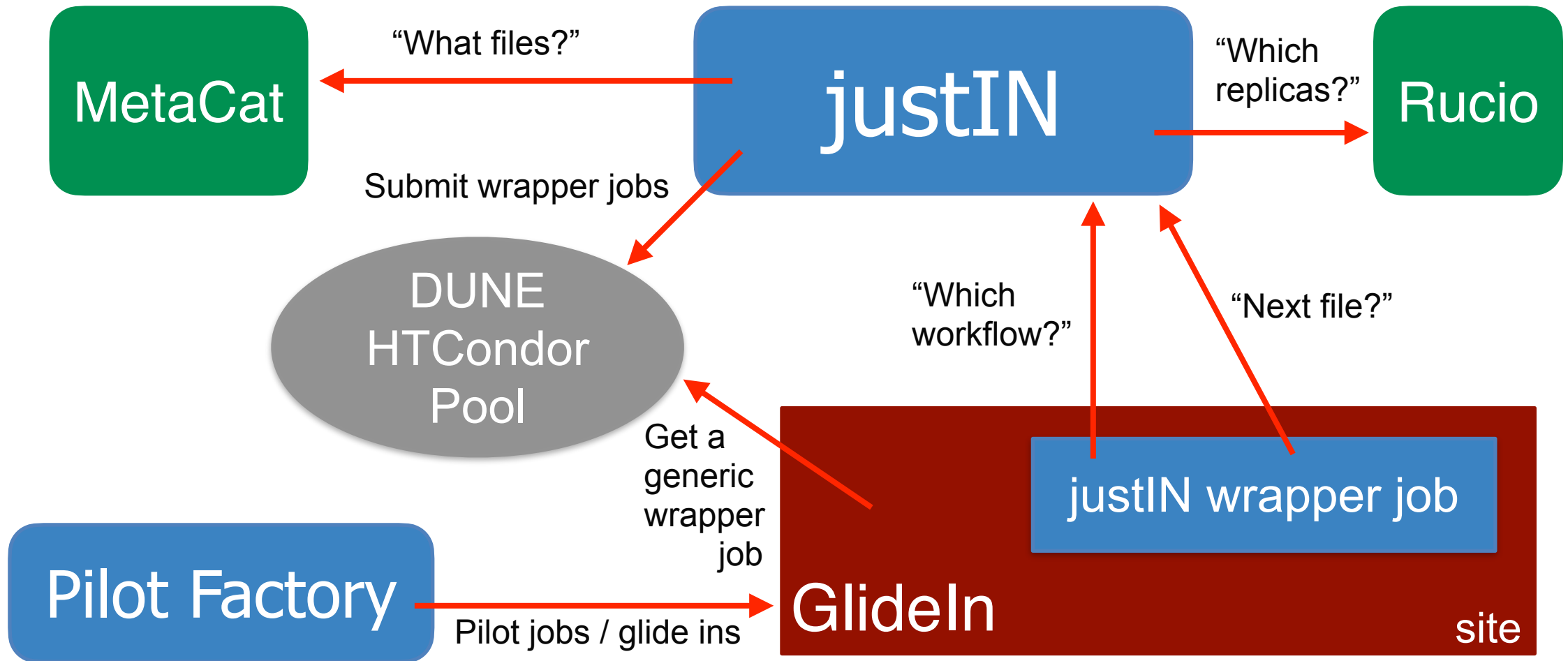
- Wrapper jobs manage the X.509 / tokens needed for the user's jobscript
- They execute the jobscript within an inner Apptainer container if possible
 - If site does not support this, run trusted jobs from DUNE production team
- They register and upload output files in MetaCat and Rucio storage using a privileged certificate that users do not have access to
 - Including retries and then failovers to other storages if necessary
 - Optionally uploading user files to Fermilab scratch instead using their SciToken
- Reporting fatal errors, running a heartbeat process in the background
- In *production*, they are entirely generic: the wrapper asks what workflow to work on
 - But in the current *development* instance, they are submitted for specific workflows and users, and rely on the just-in-time ClassAds mechanism for replica-aware matching



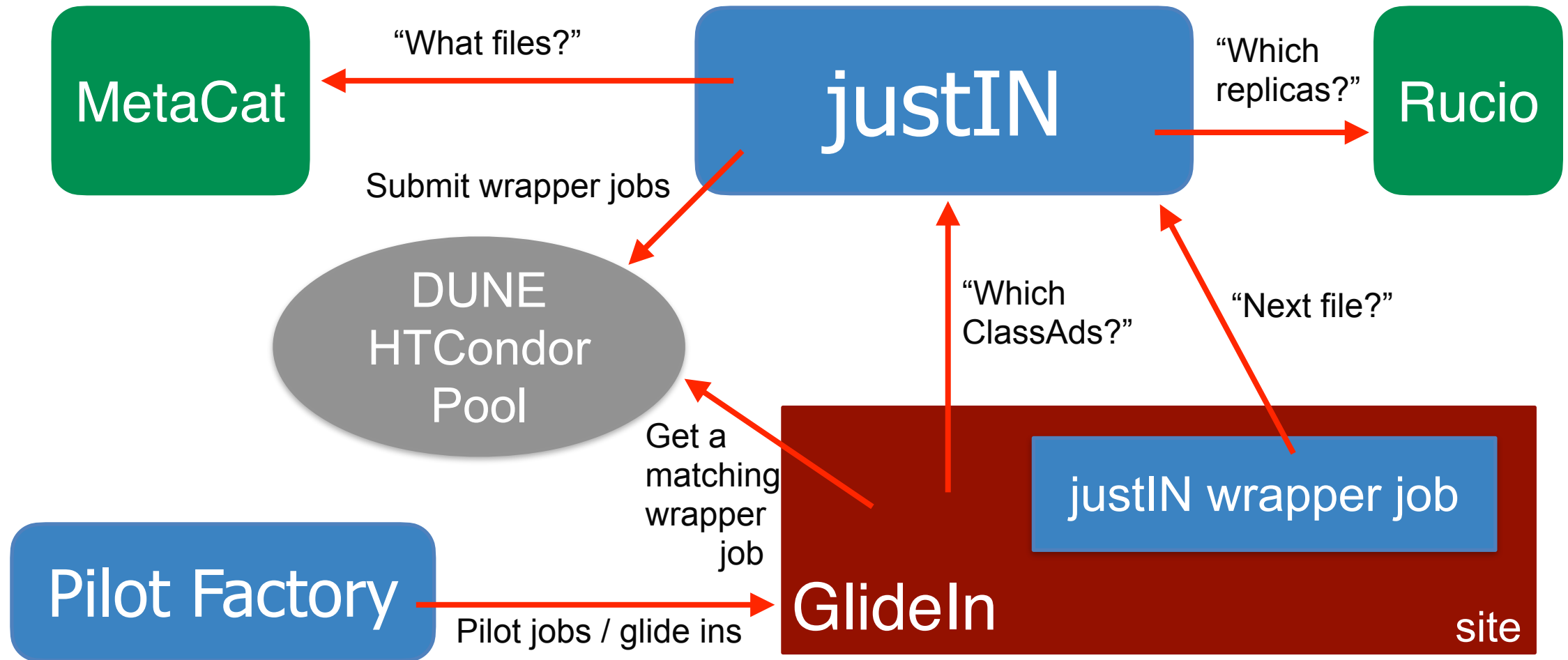
Just-in-time ClassAds

- This is the system we will deploy after the summer production campaigns are finished
- Instead of wrapper jobs asking justIN what workflow to work on, the GlideIn asks this
- Remember in GlideInWMS, pilot factories are submitting jobs to sites which then start HTCondor daemons in the job slot, which then contact the DUNE HTCondor job pool
- We have added a custom script to the GlideIns which contacts justIN and asks for extra ClassAds to publish, to rank justIN workflows **for that job slot** based on nearby unprocessed files.
 - “If I were to run a justIN workflow in my slot, what is *your* order of preference?”
- This causes the GlideIn to pull in one of the justIN jobs from the pool, based on the given rank, with user priority doing tie-breaking between equally suitable workflows
- We can use all the nice HTCondor user priority machinery, and the existing Fermilab HTCondor user and group monitoring and accounting - for justIN and non-justIN jobs

Recap: justIN with generic wrapper jobs



justIN with just-in-time ClassAds



justIN and SciTokens / WLCG Tokens

- We achieved our major goal to remove the need for user X.509 earlier this year
- Users can authenticate to the justIN web dashboard with CILogon (like IAM) and the Fermilab SSO — with OIDC between the dashboard and CILogon in the background
- This reveals the user's ID token (**not** an HTCondor IDTOKEN!), including eduPersonPrincipalName ("amcnab@fnal.gov") and WLCG groups ("/dune/production")
 - And lets justIN cache a refresh token for that user
- At the command line, users are offered a link to a page on the web dashboard to authenticate to justIN. This puts a 7 day session ID in a local file.
- Jobs that need user tokens can obtain them via justIN's cache.
- So **justIN does not actually need to trust access tokens**, but it gets user details from ID tokens and can manage access tokens for uploading outputs.

justIN schedd at RAL-PPD

- justIN submits its wrapper jobs to a dedicated HTCondor schedd within the DUNE HTCondor pool
 - Operated by RAL-PPD on the RAL OpenStack service
- The justIN central service and schedd VMs are run by the same team and use user ID based authentication between them.
- All jobs are submitted as the dunejustin user
- This enables the just-in-time ClassAds model, allowing us to submit dedicated jobs
 - So they have accounting users and groups for that type of workflow
 - `group_dune.prod_keepup / group_dune.prod_mcsim / group_dune.analysis`
- There are also latency and firewall advantages to having a dedicated schedd

Summary

- DUNE requires a major step up in scale compared to previous Fermilab neutrino experiments
 - Necessitates using storage and CPU outside Fermilab and the US
- At the same time, new metadata and replica catalogues are being adopted
- DUNE has developed a just-in-time workflow system, justIN
 - Works with the new catalogues
 - But leverages the existing investment in GlideInWMS/HTCondor
 - While delivering fine-grained matching of jobs to sites based on replica locations
 - Validated by data challenges and current Monte Carlo production

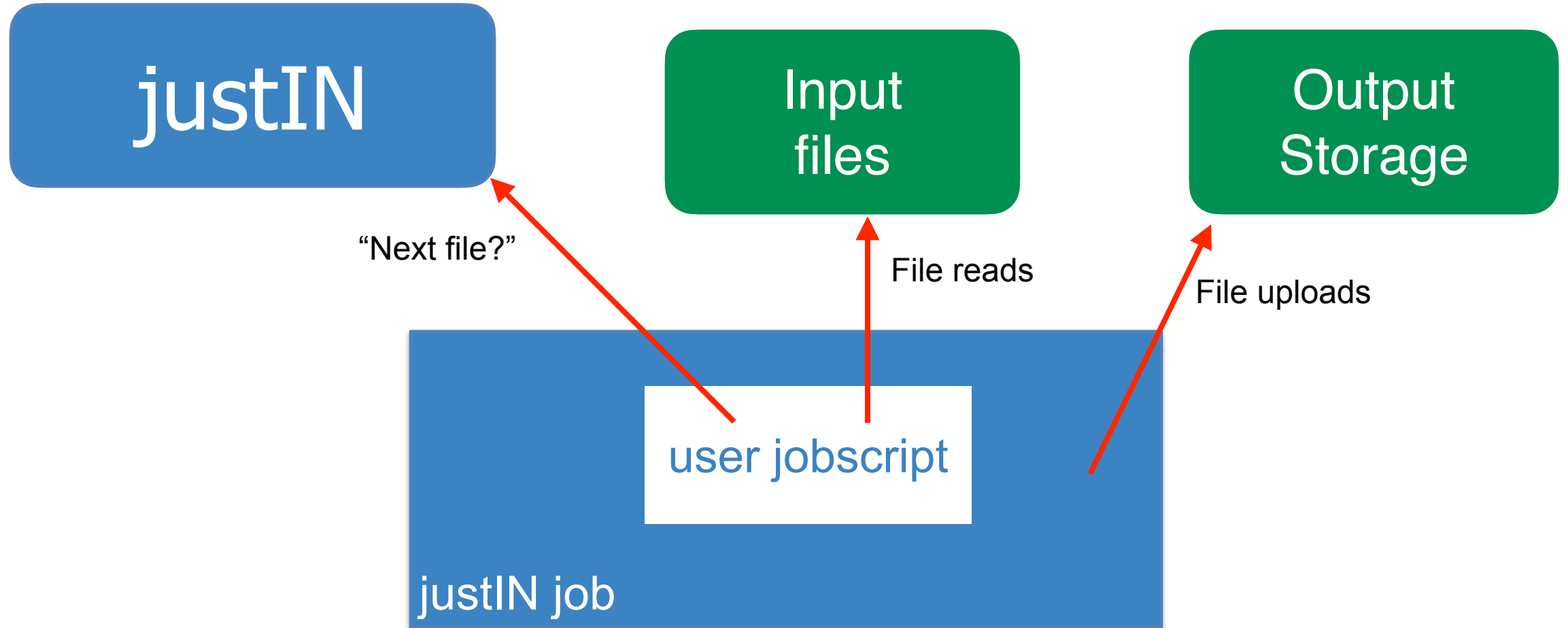


Links / references

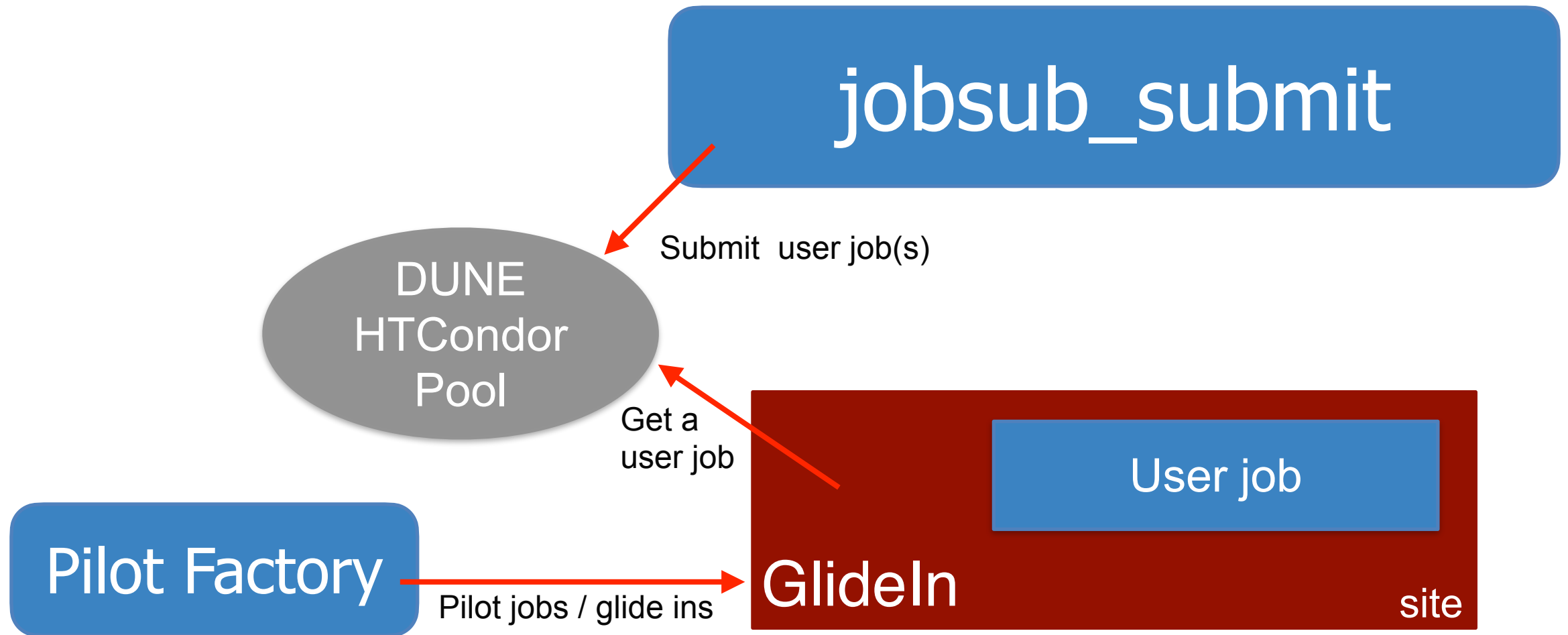
- DUNE
 - <https://www.dunescience.org/>
- Rucio
 - <https://rucio.cern.ch/>
- MetaCat
 - <https://metacat.readthedocs.io/>
- justIN
 - <https://justin.dune.hep.ac.uk/docs/>

Backup

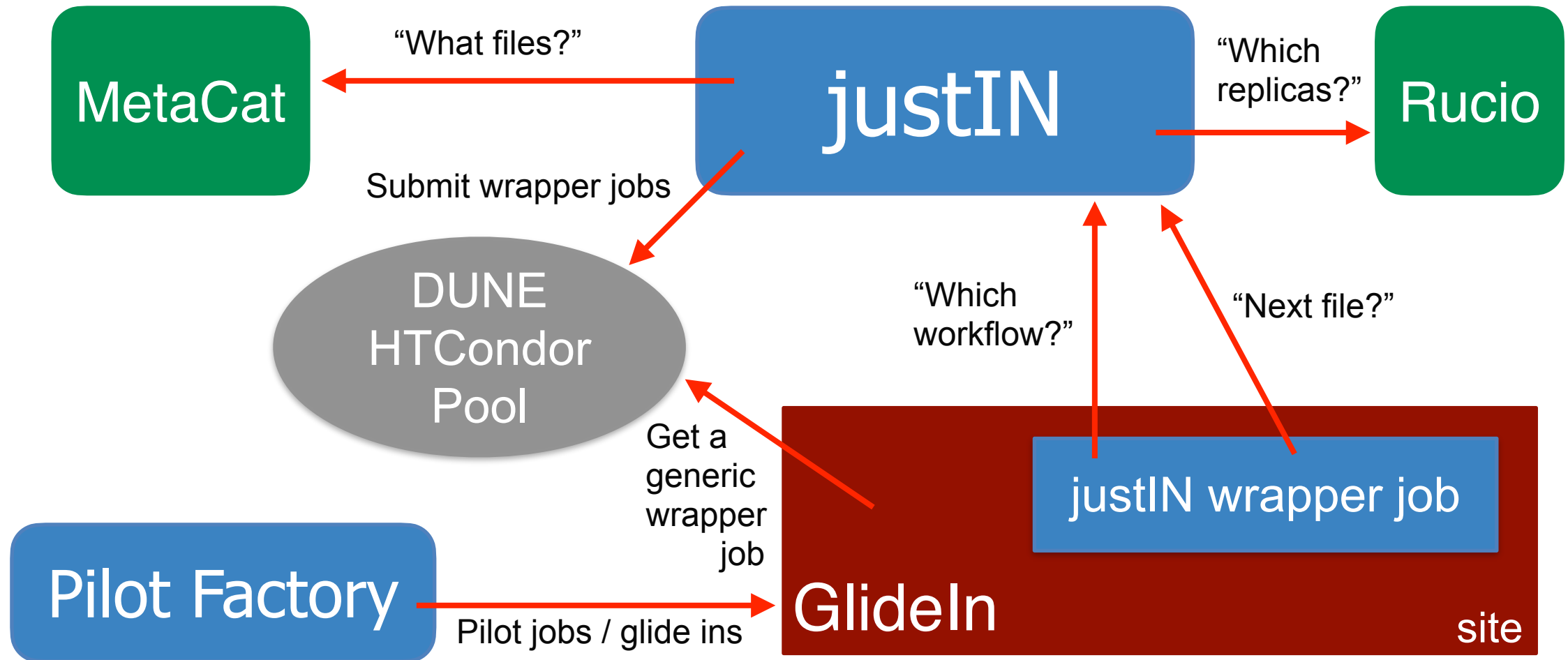
justIN wrapper jobs



GlideInWMS and Fermilab's jobsub



justIN with generic wrapper jobs



justIN with just-in-time ClassAds

