The DiRAC Data Movement Project

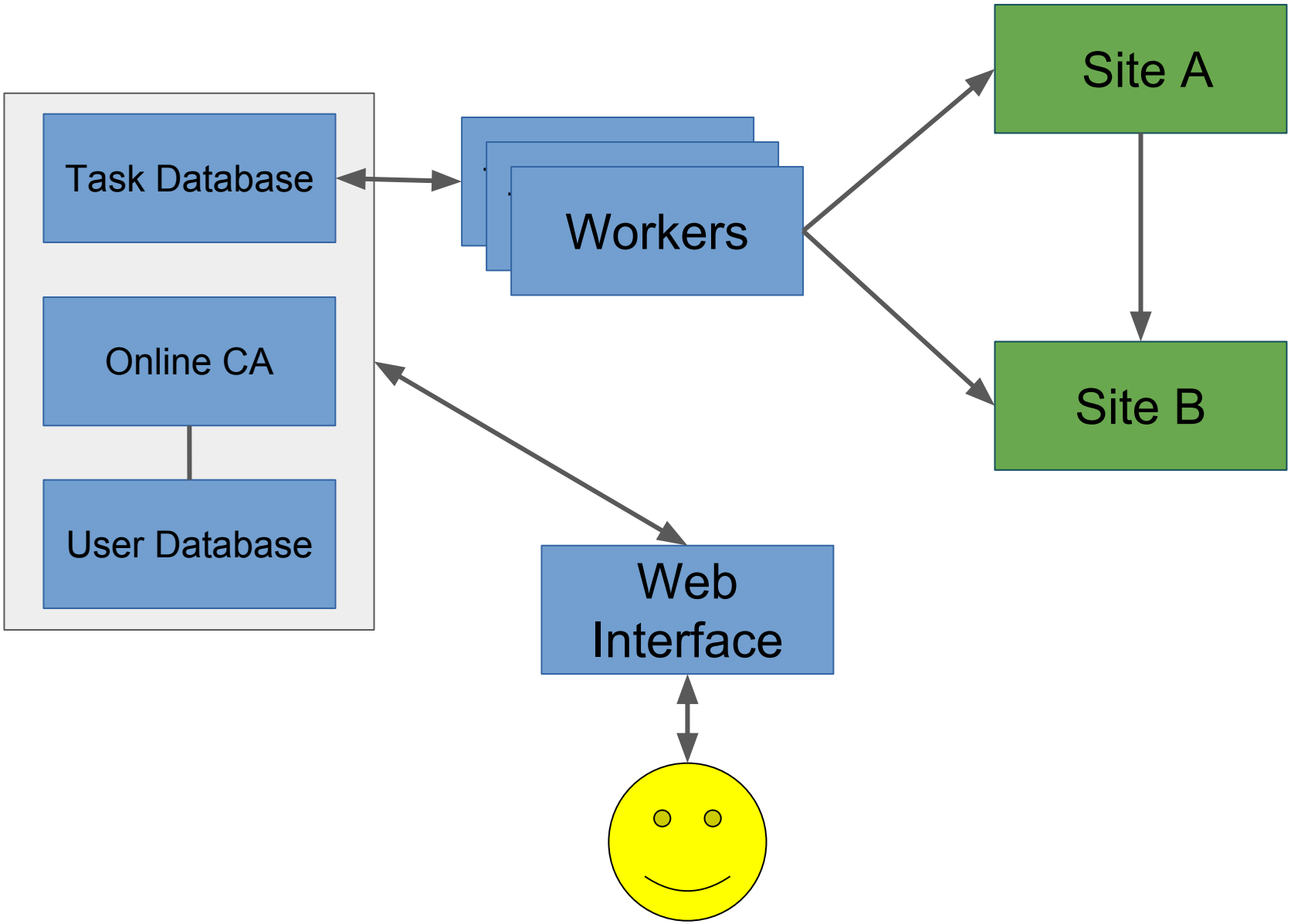
High Energy Physics Group, Imperial College

Imperial College London

Project Aims

- Use Case:
  - Facilitate data transfers across sites which don't have a shared authentication framework.
  - Must have a simple to use interface to appeal to a broad user base.
- Basic Idea:
  - LHC data distribution infrastructure & tools are well established and not specific to particle physics.
  - Build on this experience and reuse appropriate grid technologies where appropriate (X.509, GSIFTP, GFAL2 & FTS).
  - Separate (and hide) back-end technologies from the users.

System Architecture



Implementation

- Components built in Python around RESTful style web services.
- GridFTP for storage access.

- Targeted at RHEL7 versions of packages: Portable to newer operating systems (Ubuntu, Fedora, EL8).
- Wrapper scripts for easy installation/deployment.

- Fully Open Source (exact license TBD).

User Interface & Authentication

- JavaScript driven UI for managing files.
- Users register with the data mover:
  - This only gives them access to the transfer system, not the data transfer endpoints.
  - Potential for federated login.
- An X.509 certificate for the account is silently issued by the CA on account creation.

Endpoint (Site) Access

- Accessing an endpoint (site) which hasn't been used before triggers "secondary" authentication.
  - User logs into the site through the Web interface with their institution credentials.
  - Their autogenerated DN is added against the correct account in the grid-mapfile on the endpoint GridFTP server(s).
- All operations (listing, copying & deleting files) transparently submit tasks into the back-end database.
  - An X.509 proxy is automatically delegated for each task.

Back-end (Task Queue)

- Tasks in the operation database are selected by priority:
  - E.g listing tasks should be executed immediately, latency negligible for bulk copy
- Worker processes pick up a task via a separate (REST) interface and process them with the appropriate plugin.
  - All plugins are currently wrappers around the grid gfal-* tools.
- The design is scalable.
  - Workers can run remotely on multiple hosts.

Before you ask: Why not FTS?

- Back-end needs to be able to perform multiple operations: list, delete & copy.
- FTS only ‘copy’ is currently actually used:
  - Support for delete option unclear
    - Would have to implement a worker model for delete.
  - List option poorly documented, unclear
- FTS would be another service and database that needs installation, maintenance and monitoring.

Future Work

- Potential to make use of federated authentication infrastructure.
  - Even easier for users to login if they can use an existing federation account.
  - Integrates with UKT0.
  - Site authentication could be done against existing identity providers.
  - Federated on-line X.509 CAs are also available.

- Support for automated tasks such as backups or rule-based replication.

Project Status

- Currently in initial development phase.
  - Prototype versions of all modules have been written.
- Aiming for initial demonstration at the end of the March.

Any Questions?

Imperial College London

Backup Slides

Why GridFTP?

- Readily available for a number of platforms.
- Supported via the Grid Community Forum (https://gridcf.org).
- Very easy to configure in basic modes but with support for scalability if required.

Why not VOMS?

- VOMS effectively allows users to join groups:
  - Requires explicit (administrator) approval, which can be slow.
    - Creates poor user experience due to the delay.
  - Users want to access their files, not just generic group files, so some user specific mapping would still be required.
  - Would need support for the groups at the given sites.
    - Adds complexity for sysadmins & may not fit some sites' configurations at all.