

Publication as a Service: Globus Publish to ESGF

Sasha Ames (LLNL), Rachana Ananthakrishnan
(UofC/Argonne), and Lucasz Lacinski (UofC)

 Lawrence Livermore
National Laboratory



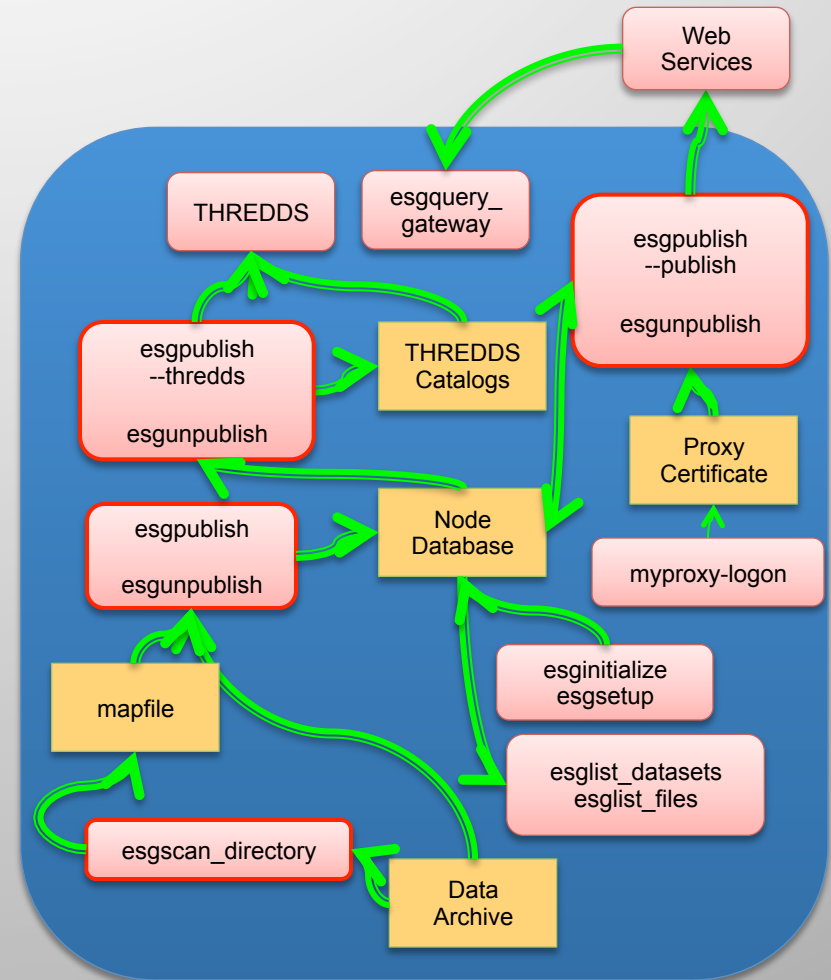
LLNL-PRES-XXXXXX

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC



ESGF Publication

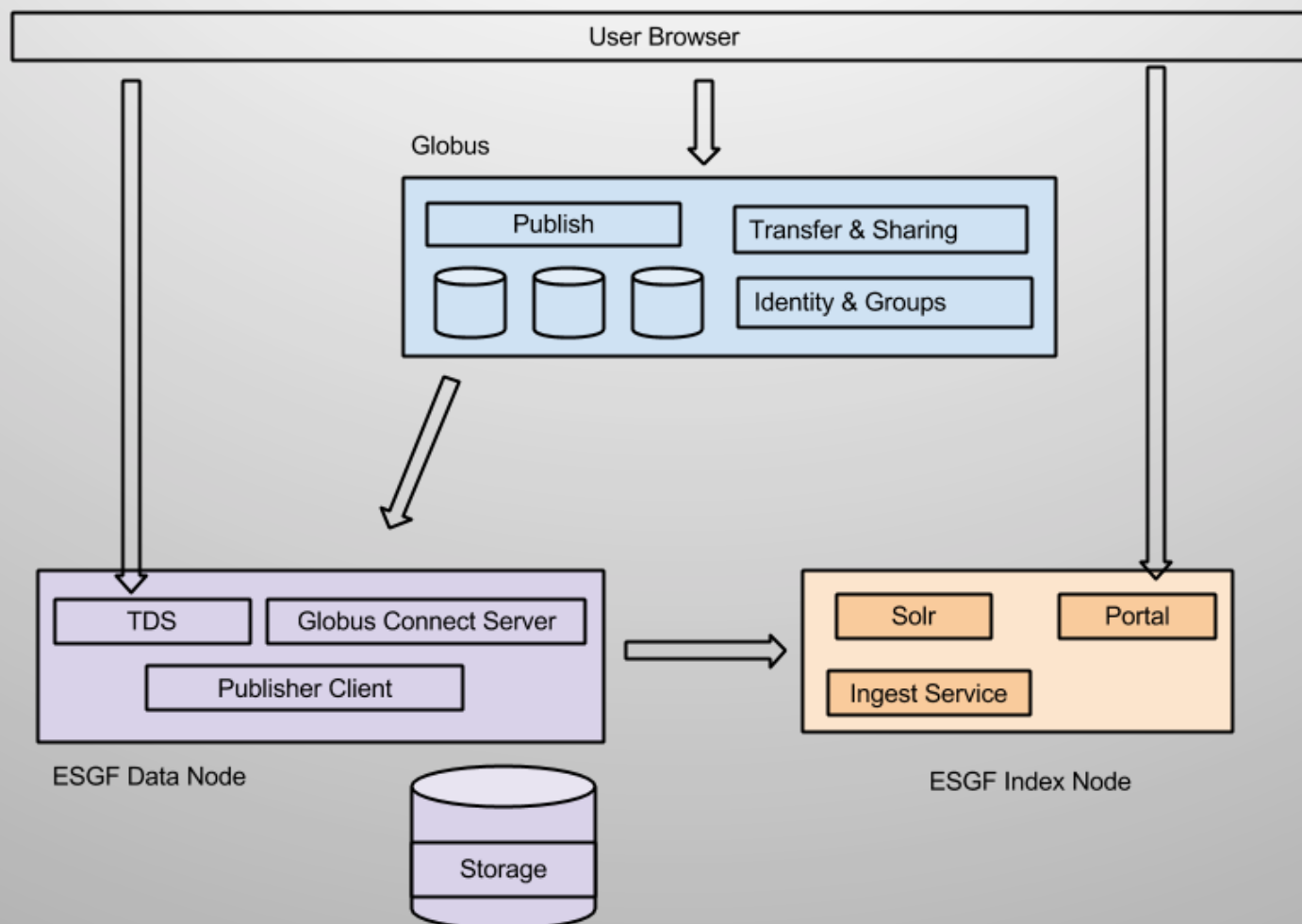
- Scripts to manage listing of files (datasets) and organize metadata for data discovery
- Multiple phases of writing metadata to three stores:
 - PostgreSQL db
 - TDS
 - SOLR (index node/gateway)
- Requires:
 - Data moved to node-mounted storage
 - Project setup for searchable facets
 - Hands-on command-line invocation
- Onerous process requires skill in systems administration



Publication as a Service

- Web based UI initiates asynchronous publication process in perspective of user
 - Leverages Globus Publish
- Data set movement coordinated via Globus Transfer
- Service coordinates invocation of publication scripts in a standardized fashion
- User to be notified of success/error at later date/time

Publication as a Service Workflow



Video demonstration of Globus Publisher service

Improvements to Publisher (in progress)

- `esgscan_directory` update
 - report errors in cases where absent
 1. Generic message better than none
 2. Need specific “parse” error message
 - (use “debug” mode)
- “Dry run” publish
 - completes CF checks without write to database
 - Useful for error reporting within Pub. As a Service Framework
- `meta_synchro.py` tool
 - Details of how the three metadata stores differ.
 - Presence/absence of data sets in each comparison category
 1. PSQL / TDS
 2. TDS / SOLR
 3. SOLR / PSQL

Considerations for Revamping the Publisher

- More flexibility for configuration sets
 - Move away from “Project” as sole concept for organization
- Invoke CDSCAN; create “.xml collections”
 - Supports tools that rely on .xml holdings of metadata, eg. VISUS
 - Robust validation of data sets for later use with UVCDAT
- Design coordination with services for automatic replication of updates
- Leverage compute resources
 - Calculate checksums
 - Run CMOR to bring data sets up to project-specified QC compliance