Node Manager, Tracking and Feedback: Progress update and future roadmap

ESGF F2F Conference
Washington, DC, December 2016

Team lead: Sasha Ames

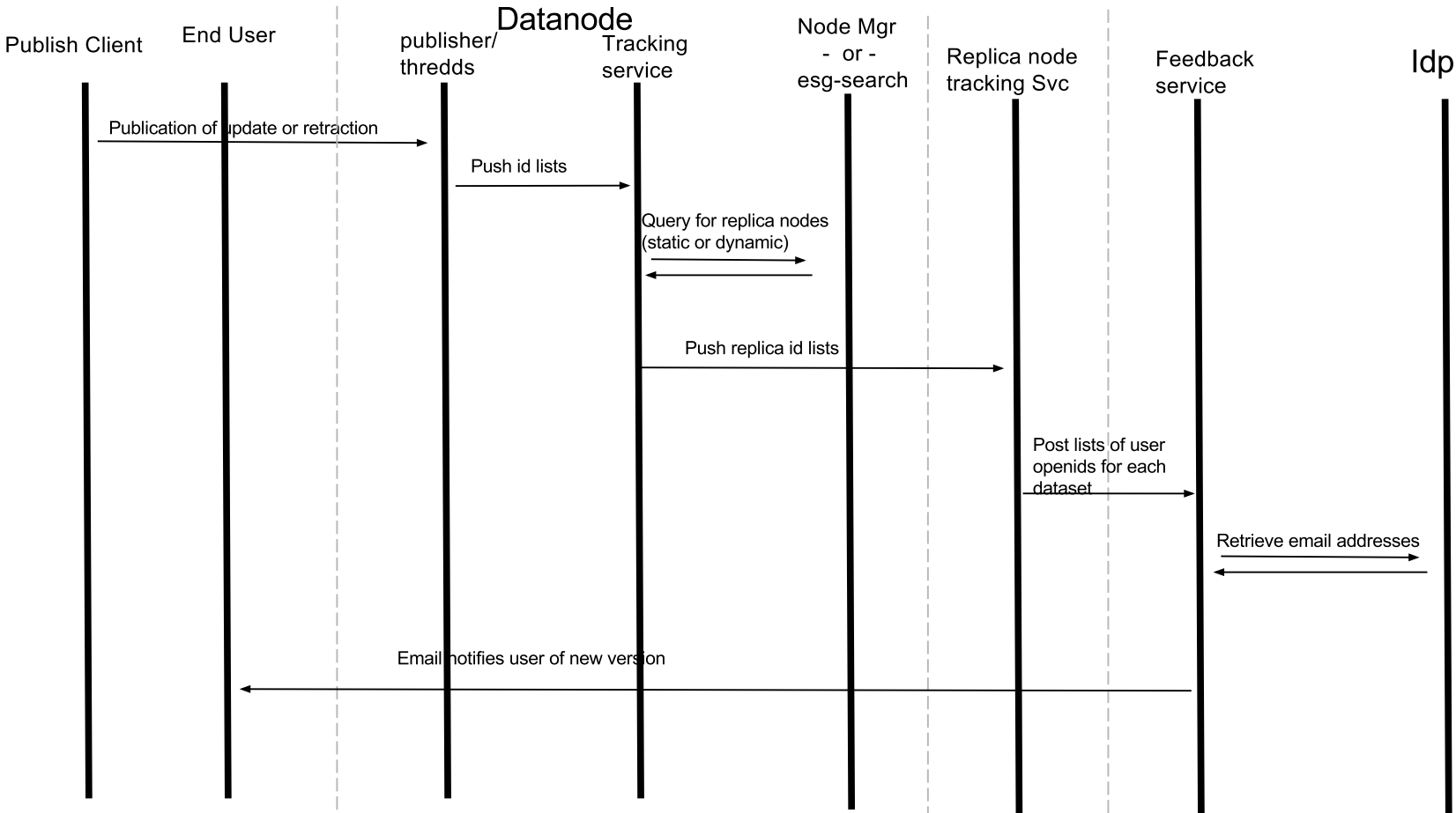Prashanth, Luca, Katharina, Nicolas

- **Node Manager**: "Core" ESGF module that maintains node status, federated metadata, properties, metrics and registry of services
- Django module integrated into apache for node-based deployment
  - Worker/monitoring daemon process running via control scripts
  - Task queue of requests through file system
- Testing performed with up to 7 nodes in cluster environment; five test-federation nodes
- Dashboard integration tests with served registration.xml
- Refactor and module conversion to SSL tested in virtual environment
- Code merged into github: https://github/ESGF/esgf-node-manager

- Provide ESGF data users with notification emails when certain "changes" occur based on previously downloaded content
  - Publication of updated dataset version
  - Retraction of dataset
- Status: architecture documented

ESGF — *Earth System Grid Federation*

Replica Node Tracking and Feedback Workflow

Publish Client | End User | Datanode — publisher/thredds | Tracking service | Node Mgr - or - esg-search | Replica node tracking Svc | Feedback service | Idp

Publication of update or retraction

Push id lists

Query for replica nodes (static or dynamic)

Push replica id lists

Post lists of user openids for each dataset

Retrieve email addresses

Email notifies user of new version

Missed milestones

- Node Manager: production deployment
- Tracking and Feedback: prototype implementation

2017 Roadmap

- Node Manager
  - Convert control scripts to esg-node
  - Integrate setup into installer
  - Federation production deployment
  - Implement secret key deployment based on signed certificates so we can identify "trusted" nodes
  - Determine if esgf_*.xml (non-static) files should be managed automatically as done by previous Java esgf_node_manager
- Tracking/Feedback
  - Begin services implementation
  - Integrate publisher notifications to tracking service
  - Feedback service: first version read user email address direct from PGSQL db
  - Query email address from IdP API

Working teams collaborations

- Node Manager
    - IWT – integrate setup into installer
    - CWT – resource availability
    - Stats – determine more metrics
    - Others – use of Node Manager for node status monitoring, sharing of metadata, etc.
        - Eg. make an IdP "not selectable" if determined to be down
- Tracking/Feedback
    - IdEA – query IdP for user emails
    - PWT, Errata – integration of tracking
    - CoG, search – Saved search UI

Additional Resources Needed

- Full/part time developer(s) would help out a lot