System for Offline Data Access (SODA)

P. Dwarakanath, P. Lundqvist

LIU/NSC

December 10 2014

Contents

What we are doing

- Integration of offline data to the ESGF data node.
- Use case: Serve EURO4M data stored in MARS via the ESGF data node.
- Solution should be general enough to be able to serve data from different offline storage back-ends.
- Should be responsible for triggering retrieval of offline data, cache management, and publication of offline data.

ESGF workflow

- Admin publishes datasets onto ESGF.
- Publication involves gathering of information from DRS (path), metadata (variable attributes such as time period), and filesystem (timestamps, checksums)
- ESGF currently only supports the NetCDF file format.
- Users can download wget scripts containing links to datasets selected by them using the ESGF web-frontend.
- Data is also presented via the THREDDS data server (TDS), allowing for data subsetting.

Design considerations for offline data handling

- Amount of offline data may be huge and have very long retrieval times; Publication would have to be done without having to retrieve data to disk
- Data format may not be NetCDF (MARS uses GRIB).
- Data unit may not be a file (MARS works with individual fields).
- File attributes (size, checksum, timestamps) may be unknown, prior to decompression and format conversion etc.

Design requirements and limitations

- Allow for efficient publishing onto ESGF without requiring actual retrieval of offline data.
- Make it easy to support different offline storage back-ends by providing a simple and well-defined plugin framework.
- Implement caching of offline content to local filesystem.
- Build upon existing ESGF code to make integration as seamless as possible.
- Rely on existing authentication infrastructure of ESGF

Design requirements and limitations

- Provide a separate, standalone file server, external to ESGF
- WPS implementation to ease adoption as WPS is popular across ESGF sites.
- THREDDS shall not be supported for offline data, at least in the initial version.
- No prospect of allowing for downloads of subsets of requested datasets until all of them are available.

SODA overview

- Separate service and plugins:
  - Defines common behaviour that each offline storage system needs to provide and put rest of the logic in separate service (SODA)
- Service:
  - WPS interface for handling publishing and data retrieval.
  - caching, cache replacement logic and task scheduling.
- Plugin:
  - Retrieving and returning metadata for offline data publication.
  - Format conversion from GRIB to NetCDF for requested datasets and writing to fileserver.
  - Configurable DRS and project specific attributes external to DRS.

Components of SODA

1. SODA WPS API
   - Provide semantics for efficient publication of offline content.
   - Initiation of data retrieval requests.
   - Asynchronous.
   - Invoked by ESGF Web-frontend; Not exposed directly to end users.
2. SODA Scheduler
   - A task queue for scheduling and executing asynchronous processes for data publication, retrieval and cache management.
   - Logic for handling overlapping requests.

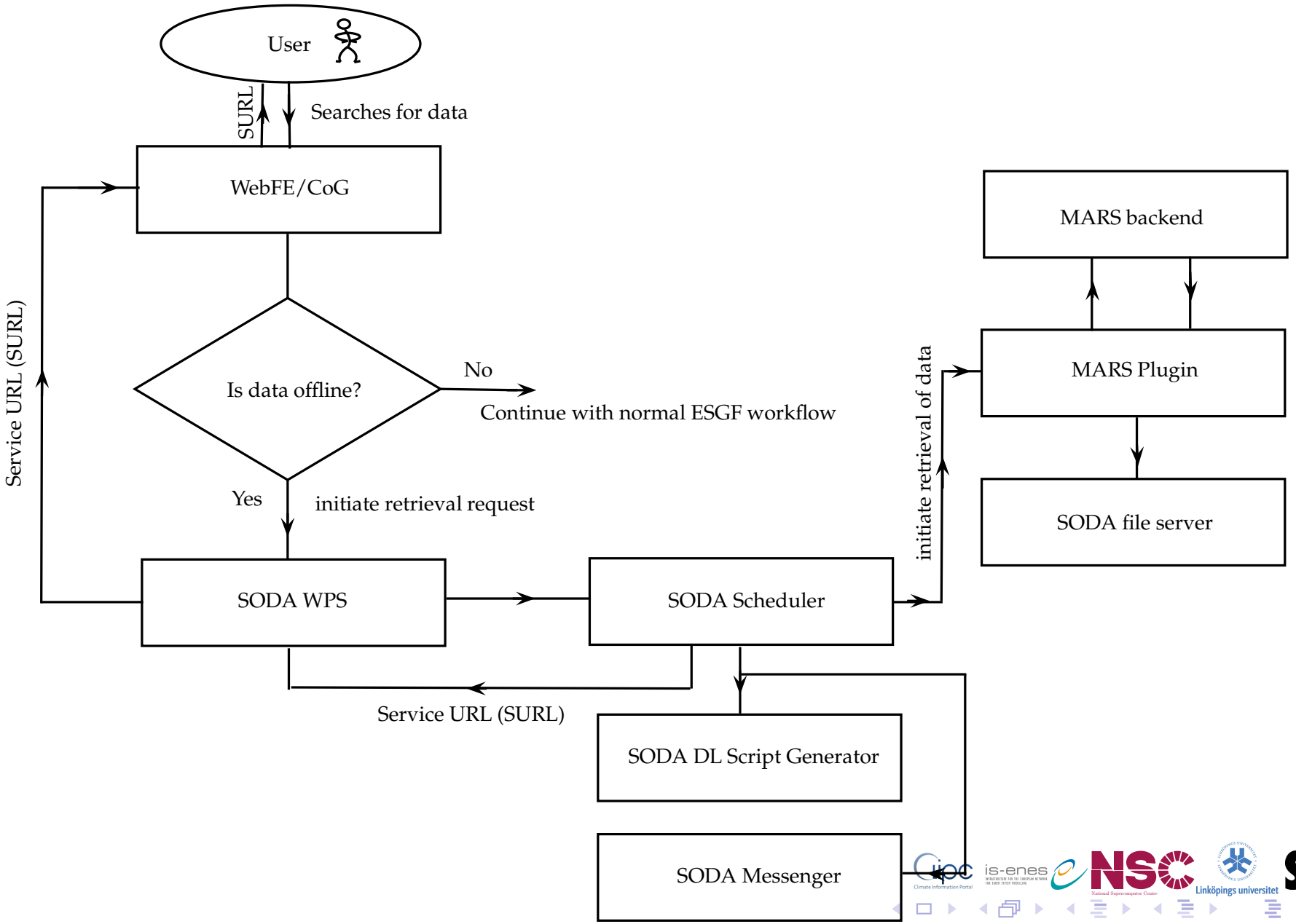
Components of SODA

3. SODA WPS client Python API
   - A Python API for easily making the defined WPS calls to a SODA instance.
4. SODA HTTP fileserver
   - Service files from cache to end user.
   - Collect download metrics and use it for cache management decisions.
5. SODA database
   - Task status, file availability status, download metrics.

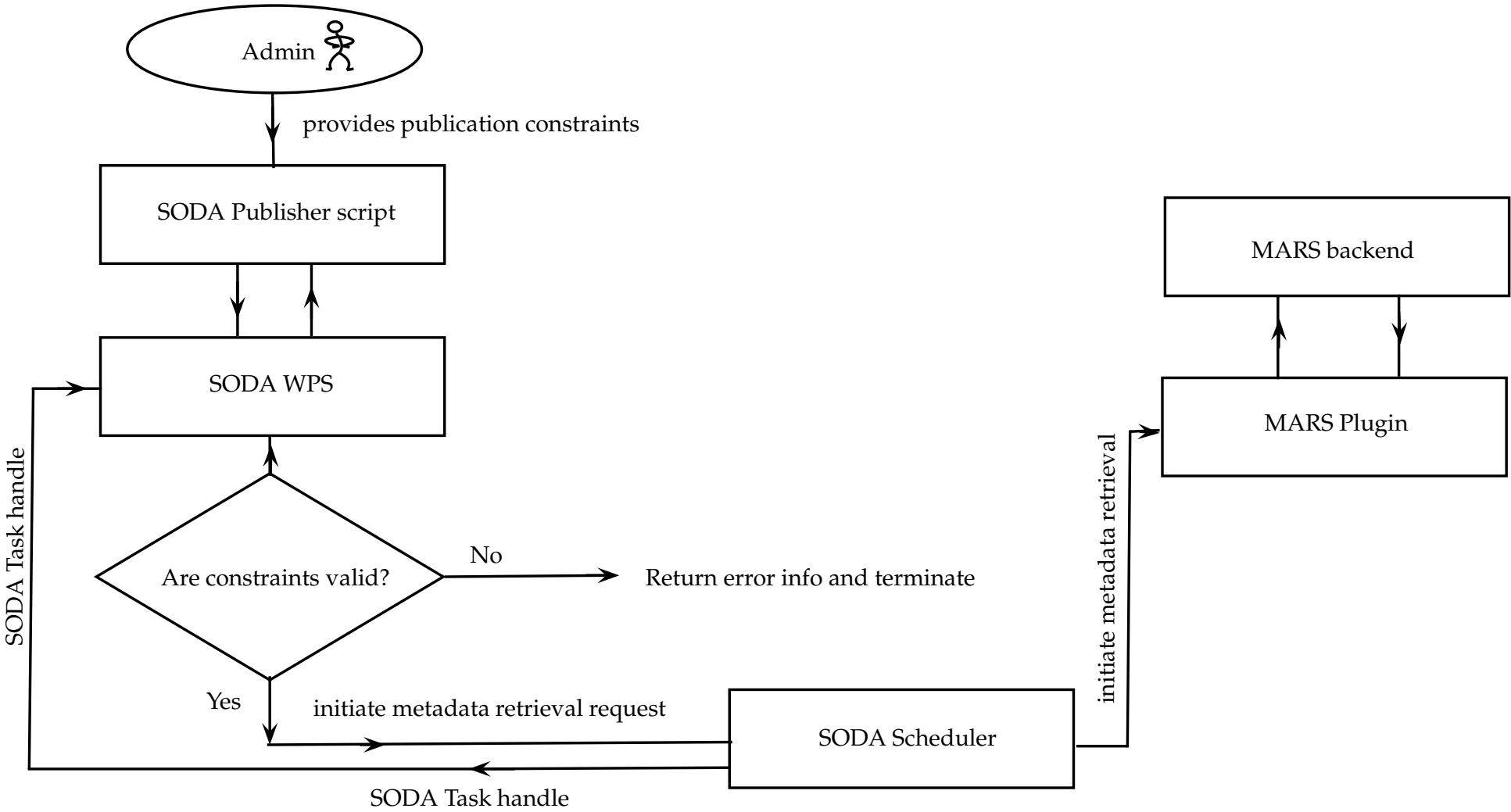
Components of SODA

6. SODA Messenger
   - System to notify users of task status by email.
7. SODA wget script generator
   - System to generate script with download URLs to datasets requested by users.
   - To use the same auth mechanisms as conventional ESGF wget scripts.
8. SODA Publisher
   - A component of SODA that **must** be collocated on the ESGF datanode for being able to publish data.
   - Uses SODA WPS client API for initiating data publication.
   - Implements a custom publisher tool, based on the ESGF publisher.
   - Uses metadata returned by SODA to publish data, instead of having to scan physical files.

Component Diagram for data download workflow



Component Diagram for data publication workflow



Features for future releases

- Advanced client that would allow for incremental downloading of files, without requiring that all requested data be available at the same instance.