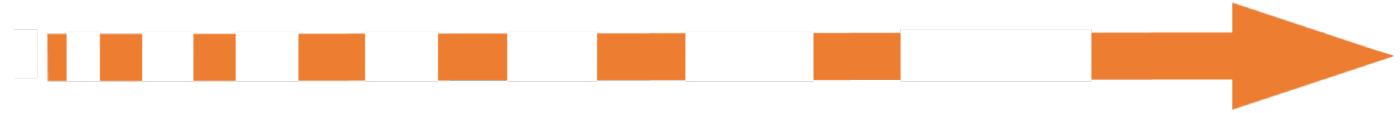


# Task 4.3 Data pre-processing at ingestion/Orchestrating workflows based on data movement



Data Management for extreme scale computing



Marica Antonacci

[marica.antonacci@ba.infn.it](mailto:marica.antonacci@ba.infn.it)



eXtreme DataCloud is co-funded by the Horizon2020  
Framework Program – Grant Agreement 777367  
Copyright © Members of the XDC Collaboration, 2017-2020

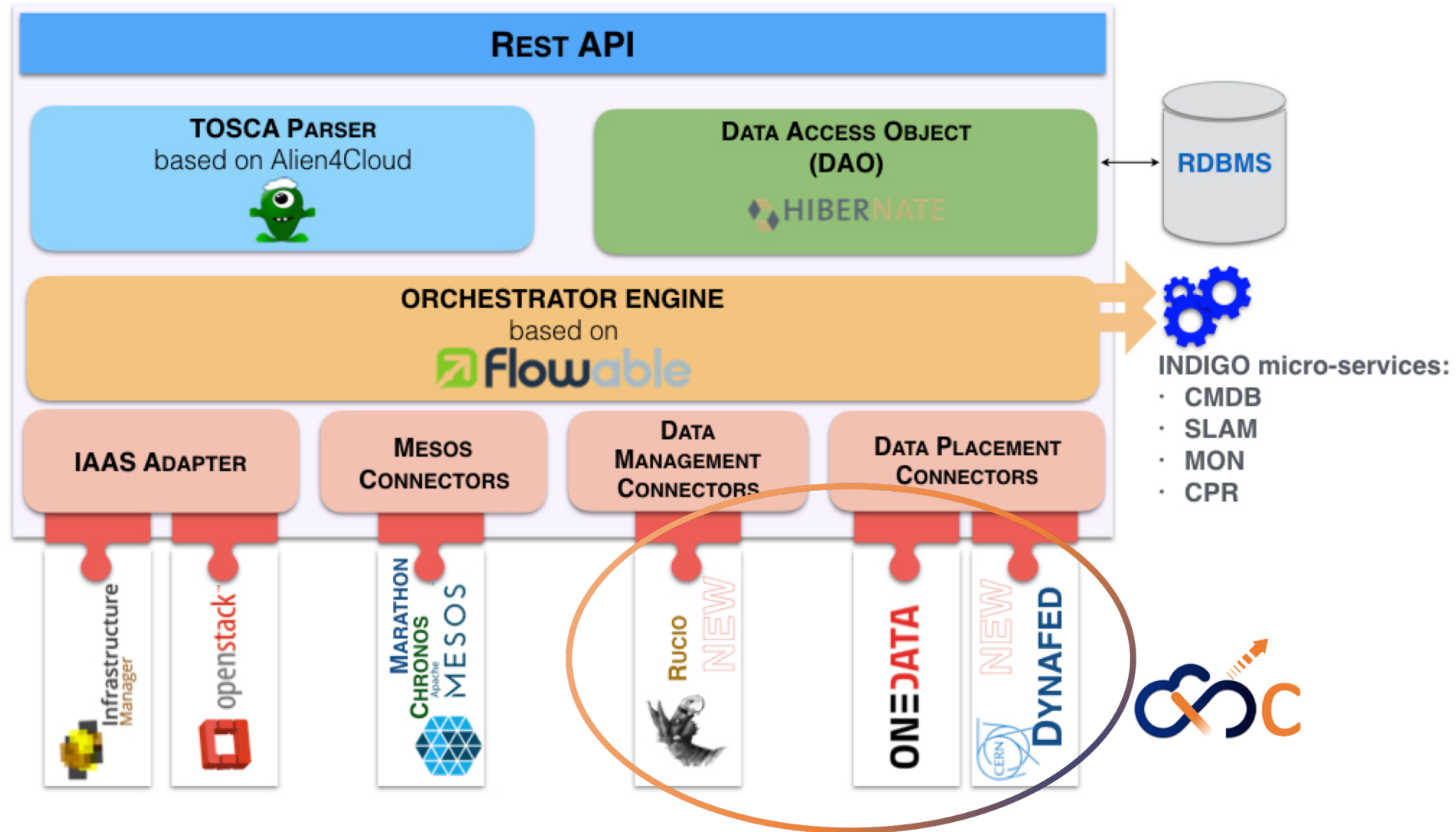
# Task 4.3 Goals:

- ✗ Implement a workflow engine...
  - exploiting the INDIGO PaaS Orchestrator capabilities of requesting execution over distributed Computing resources.
- ✗ ... to run user applications before performing the actual storing of data on the final storage destination into the infrastructure resources.
- ✗ Use cases:
  - run experiment-independent quality checks,
  - data skimming,
  - metadata extraction and/or translation
  - indexing...

# Progress Report:

- ✗ State of the art review (done)
  - INDIGO Orchestration solutions
  - Rucio Distributed Data Management System
- ✗ Analysis and testing activities on Rucio (in progress)
- ✗ Definition of the reference workflow for the data pre-processing at ingestion
- ✗ Contribution to the Architectural design (Deliverable 4.1)
- ✗ Enhancement of Onedata plugin in the Orchestrator (will be ready for XDC-1)
- ✗ Implementation of Dynafed plugin in the Orchestrator (will be ready for XDC-1)
- ✗ Implementation of a Java client for Rucio (in progress)
- ✗ Implementation of notifications in dCache

# Extending the INDIGO Orchestrator



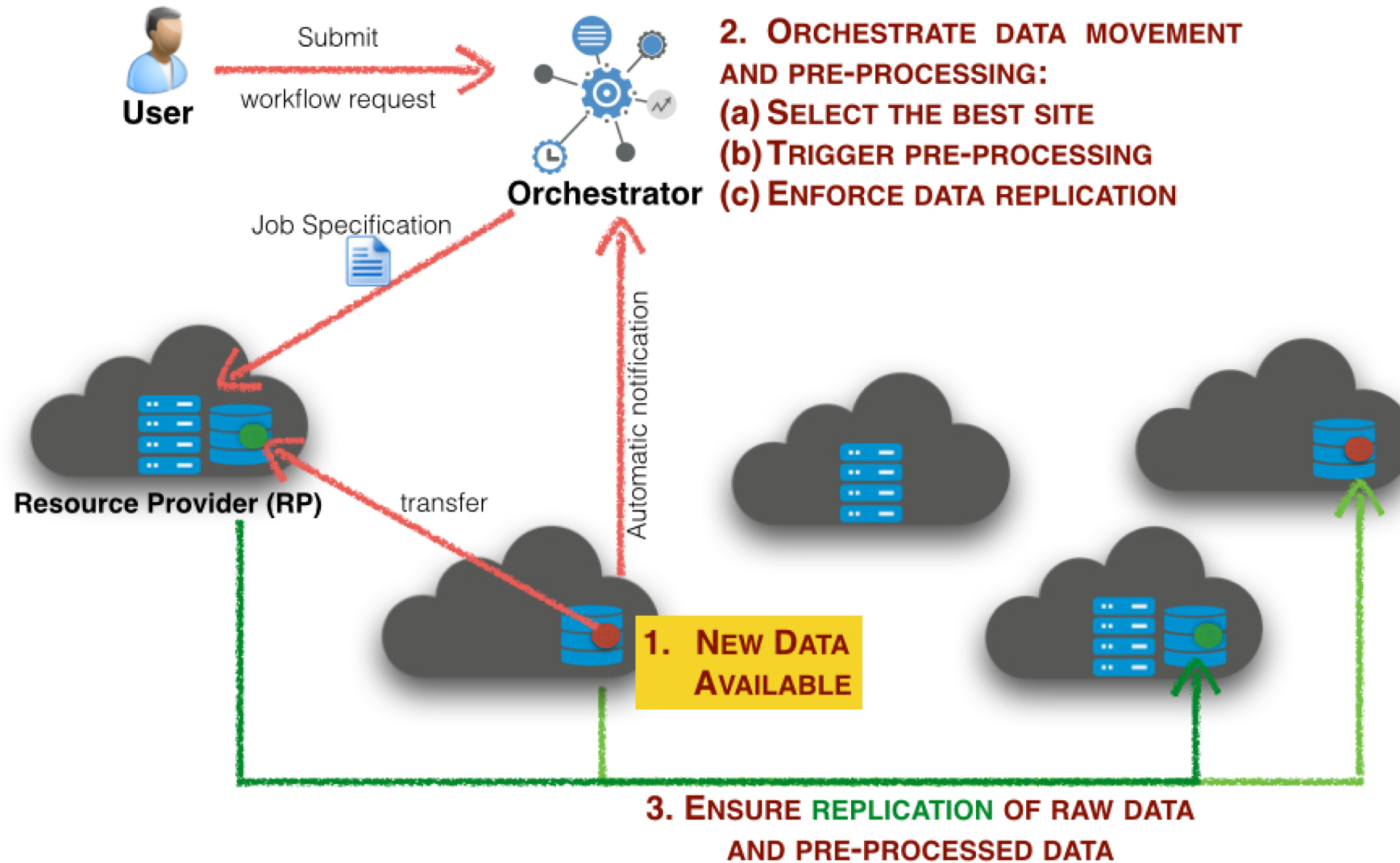
# Onedata Data Placement Plugin: improvements

- ✗ INDIGO prototype requires the user to provide the access token in the TOSCA template for mounting the user onedata space
- ✗ The XDC plugin will programmatically get a valid access token from Onedata using the user IAM (oauth2) token
  - ➡ The user will no longer be asked to provide the access token in the TOSCA template

# Dynafed Data Placement Plugin:

- ✗ Exploits HTTP/DAV to get the location of the file replicas
  - HTTP PROPFIND operation
- ✗ The information about the replicas is used by the Orchestrator to schedule the user job to the best computing site
  - Monitoring, health data and metrics about the storage endpoints might be used to improve the scheduling strategy – discussion on-going
- ✗ Openid-Connect Authentication integrated in Dynafed
  - Based on apache openidc module
  - The Orchestrator will interact with Dynafed endpoints using the INDIGO IAM oauth2 token

# Reference workflow: pre-processing at data ingestion



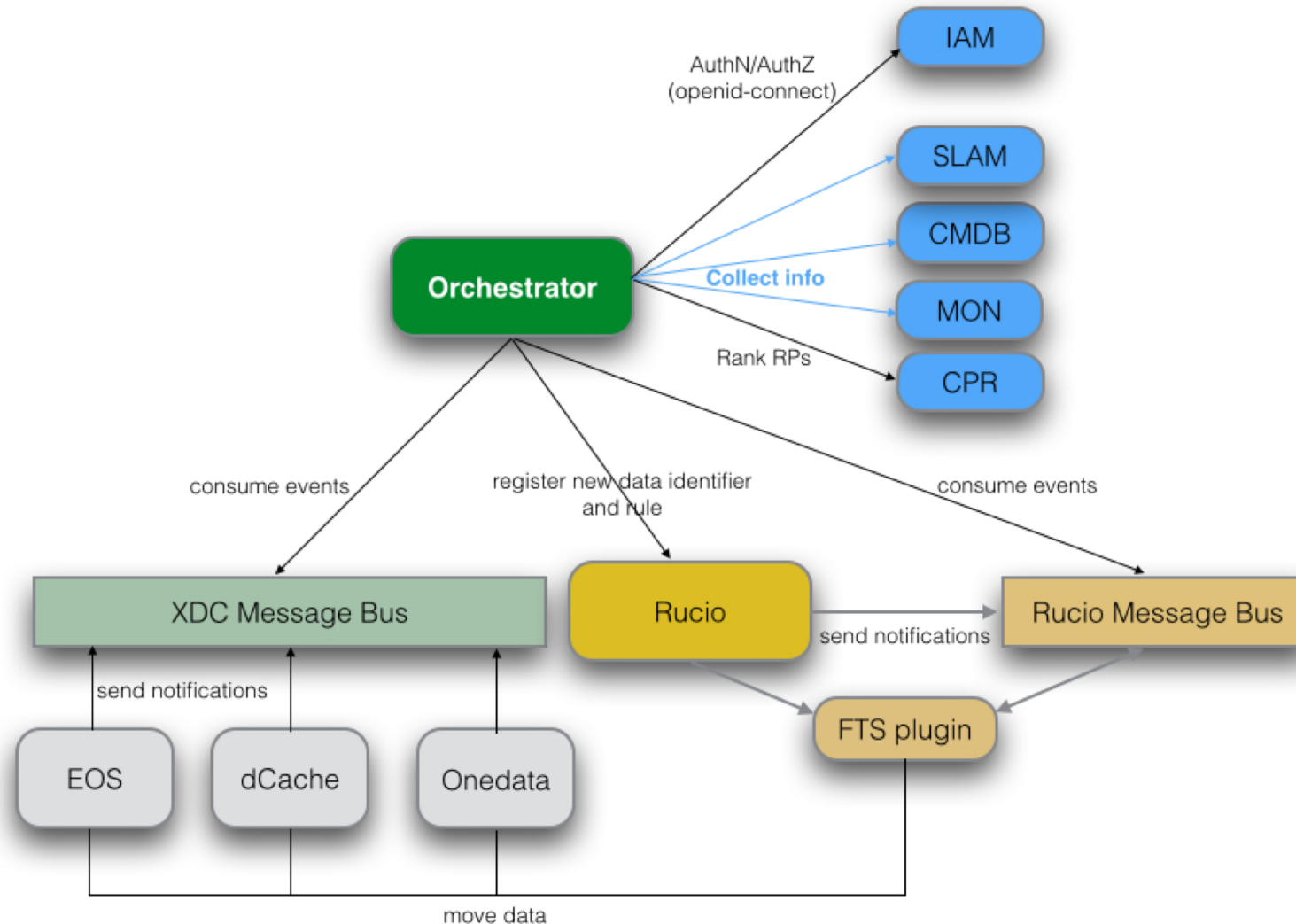
# Reference workflow: pre-processing at data ingestion



1. The user submits his workflow request to the Orchestrator including the following information:
  - the space to watch for incoming data
  - the application to be run on incoming data
  - the replication rule to be enforced on the incoming and/or preprocessed data.
2. The storage system, holding the watched storage, notifies the presence of new data by sending a message to the XDC message queue.
3. The INDIGO Orchestrator receives the notification and registers the ingested data file into Rucio, including the replica policy, specified by the user.
4. The Orchestrator selects the best compute site to perform the requested processing. To do so, it might collect information from different sources: CMDDB, SLAM, Monitoring, Storage endpoints.
5. The Orchestrator triggers a data movement through Rucio in order to copy the data to the selected compute center.
6. The Orchestrator gets notified on the completion of the data transfer, by listening to the Rucio Message Queue.
7. The Orchestrator triggers the processing Job by submitting the request to those computing clusters (Mesos/Kubernetes) available at the site.
8. As soon as the job output is produced, its availability is notified to interested parties, in particular Rucio, via the XDC message bus.
9. The data, generated by the processing step, is automatically registered into Rucio through the Orchestrator.
10. Rucio takes care that the policies requested for the raw and preprocessed data are applied.



# Orchestration: Draft Architecture



# dCache storage events

## ✗ Support has been added for storage events

- ➡ Allow non-dCache agent to get notified that something of interest happen inside dCache

## ✗ Two mechanisms to receive events

### ➡ Kafka

- ➡ exposes dCache internal events (currently) without any security. This is intended for trusted (or otherwise tightly integrated) services.

### ➡ Server Sent Events (SSE)

- ➡ generic interface that requires authentication. Out-of-the-box, it is available to all dCache users (there's no anonymous event subscription).

# Main Open points

- ✗ User requirements to be describe in TOSCA
- ✗ Review the reference workflows
  - ➡ Integration of the different components: authentication flow, QoS specification, notifications, etc.
- ✗ Testbed setup
  - ➡ PaaS
  - ➡ Rucio
  - ➡ Storage endpoints