*

Datasets Service

[**Container:** Flask, Python, JWT, Pymongo]

Manages datasets and stores associated metadata

Uses [HTTP, JSON] →

Reads from and writes to [Pymongo] →

Database

[**Container:** Mongo, NoSQL]

Stores the metadata from each dataset

Workflows Service

[**Container:** Flask, Python, JWT, Pymongo, Condu]

Manages and translates the workflows to a more specific format to be sent to the orchestrator service for execution.

Uses [HTTP, JSON] →

Reads from and writes to [Pymongo] →

Database

[**Container:** Mongo, NoSQL]

Stores the users' workflows

Uses [Condu] ↓

Orchestration Service

[**Container:** Conductor, Dynomite]

Manages and executes the workflows

Reads from and writes to →

Database

[**Container:** Dynomite, Redis]

Stores the state of workflows and the metadata associated with workflows and tasks

Gets tasks [Condu] ↑

Reads from and writes to

Data science Services

[**Container:** Python]

Retrieve tasks for execution from the orchestration service; executes them and returns the results to the orchestration service.

Reads from and writes to →

Distributed File System

[**Container:** NFS]

Manages data such as user uploaded datasets and data produced in the data science services.

DS4NP Platform
[Software System]