



DAISY : Data analysis integrated software system for X-ray experiments

Data Analysis and Software Integration System

Haolai Tian, Yu Hu, Zhibin Liu, Qiulan Huang, Hao Hu, Fazhi Qi

tianhl@ihep.ac.cn(On behalf of Computing Center, IHEP and C&C, HEPS)

HEPiX Spring 2021 online Workshop, 15th March 2021

Overview

Missions & Requirements

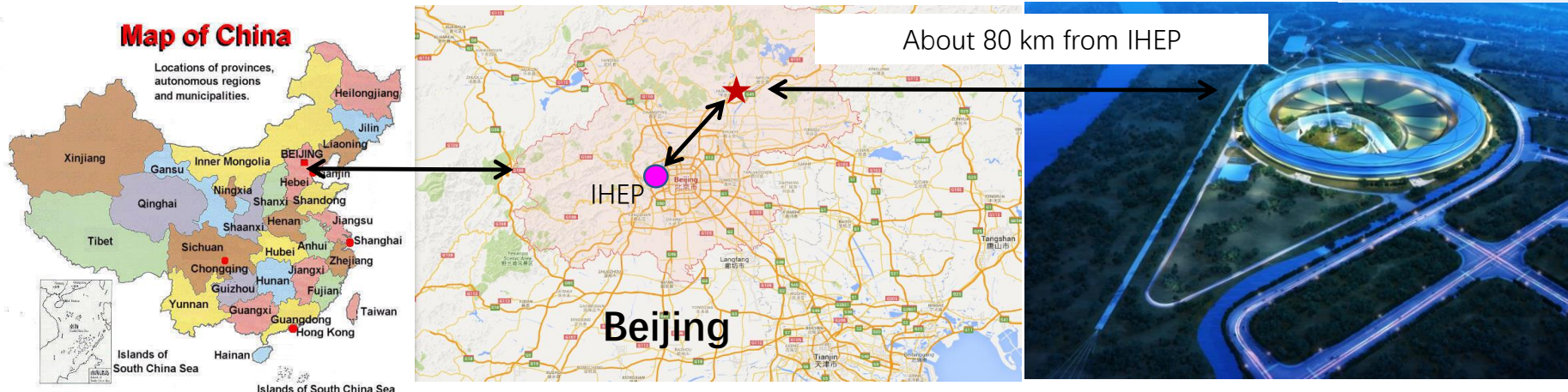
Architecture and Design

HEPS Testbed

HEPS: High Energy Photon Source

- New light source in China — High energy, high brightness
- Located in Beijing - about 80KM from IHEP
- Officially approved in Dec. 2017,
- The construction was started at the end of 2018
- The whole project will be finished in mid-2025

Main parameters	Unit	Value
Beam energy	GeV	6
Circumference	m	1360.4
Emittance	<u>pm</u> ·rad	< 60
Brightness	<u>phs</u> /s/mm ² /mrad ² /0.1%BW	>10 ²²
Beam current	mA	200
Injection		Top-up

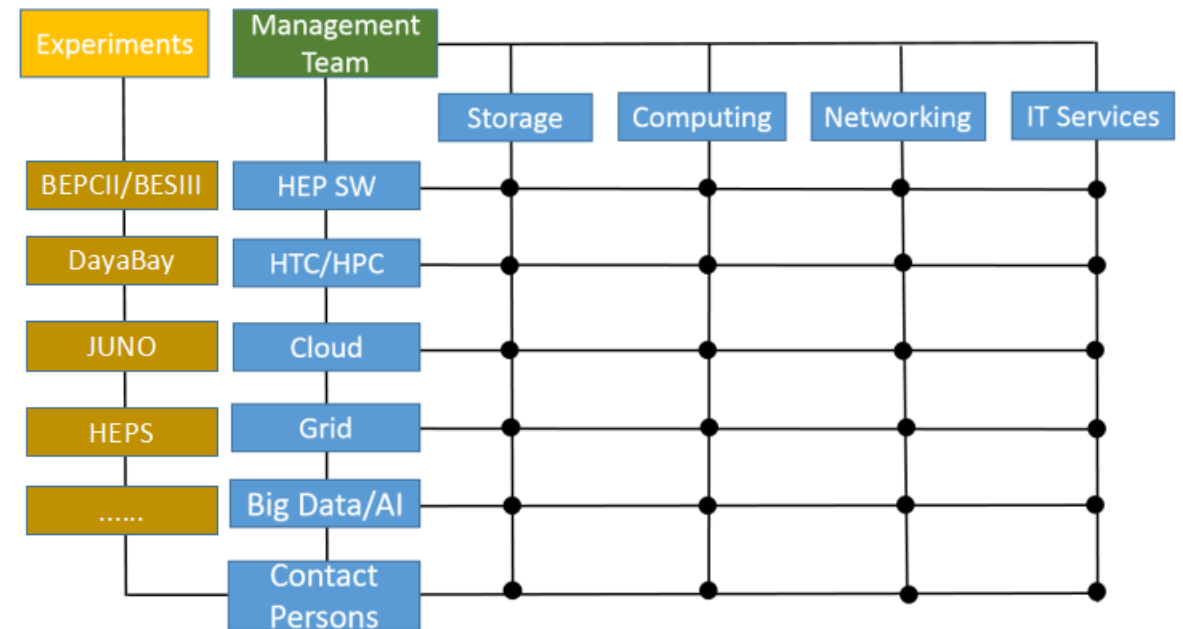


HEPS CC: the Computing & Communication System for HEP

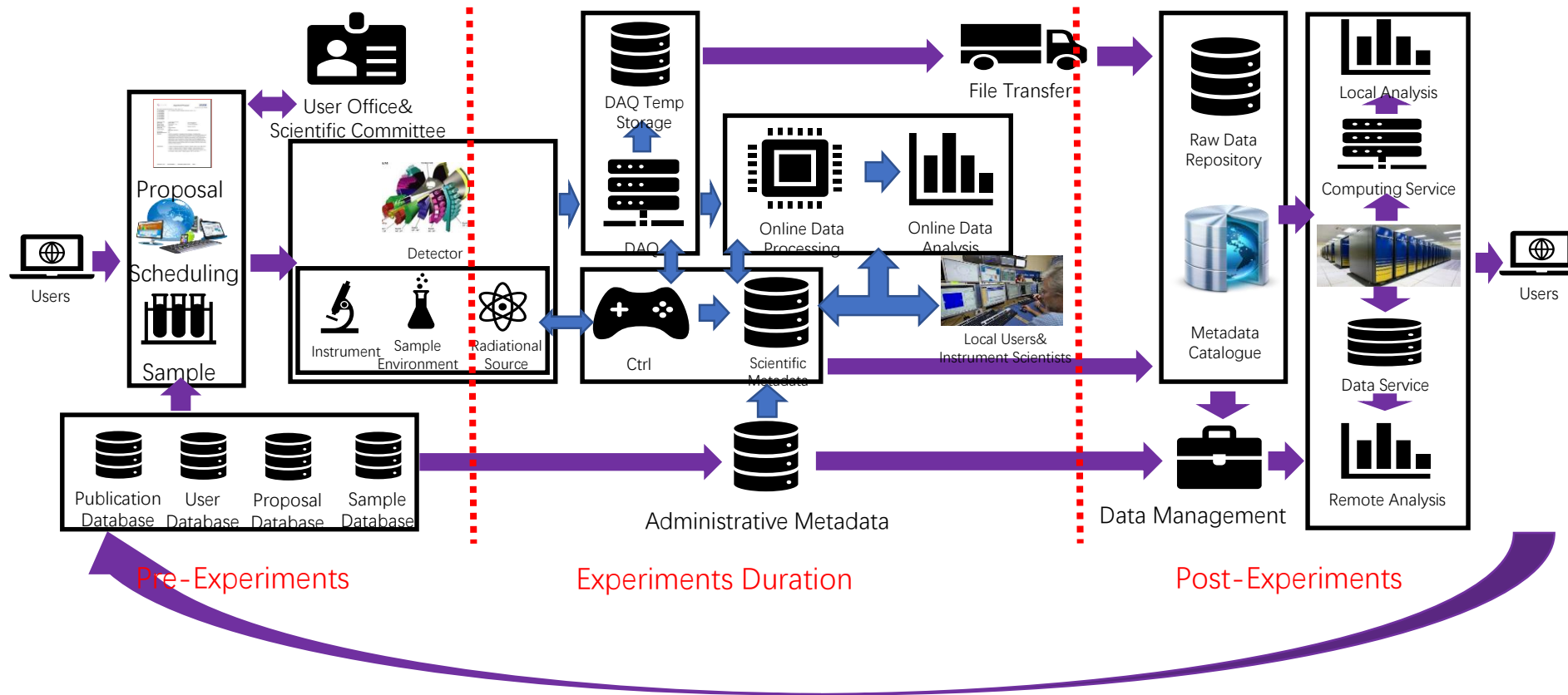
- 30+ members
 - Most of the members are coming from IHEP CC
 - 3 from CSNS/Computing and Software group
 - 1 from Beamline
- 7 workgroups are set up according to the tasks
 - Infrastructure , Network, Computing & Storage, Scientific Software, Data management, Database & Public Service , Monitoring, Security

Matrix management

Sharing talents and skills

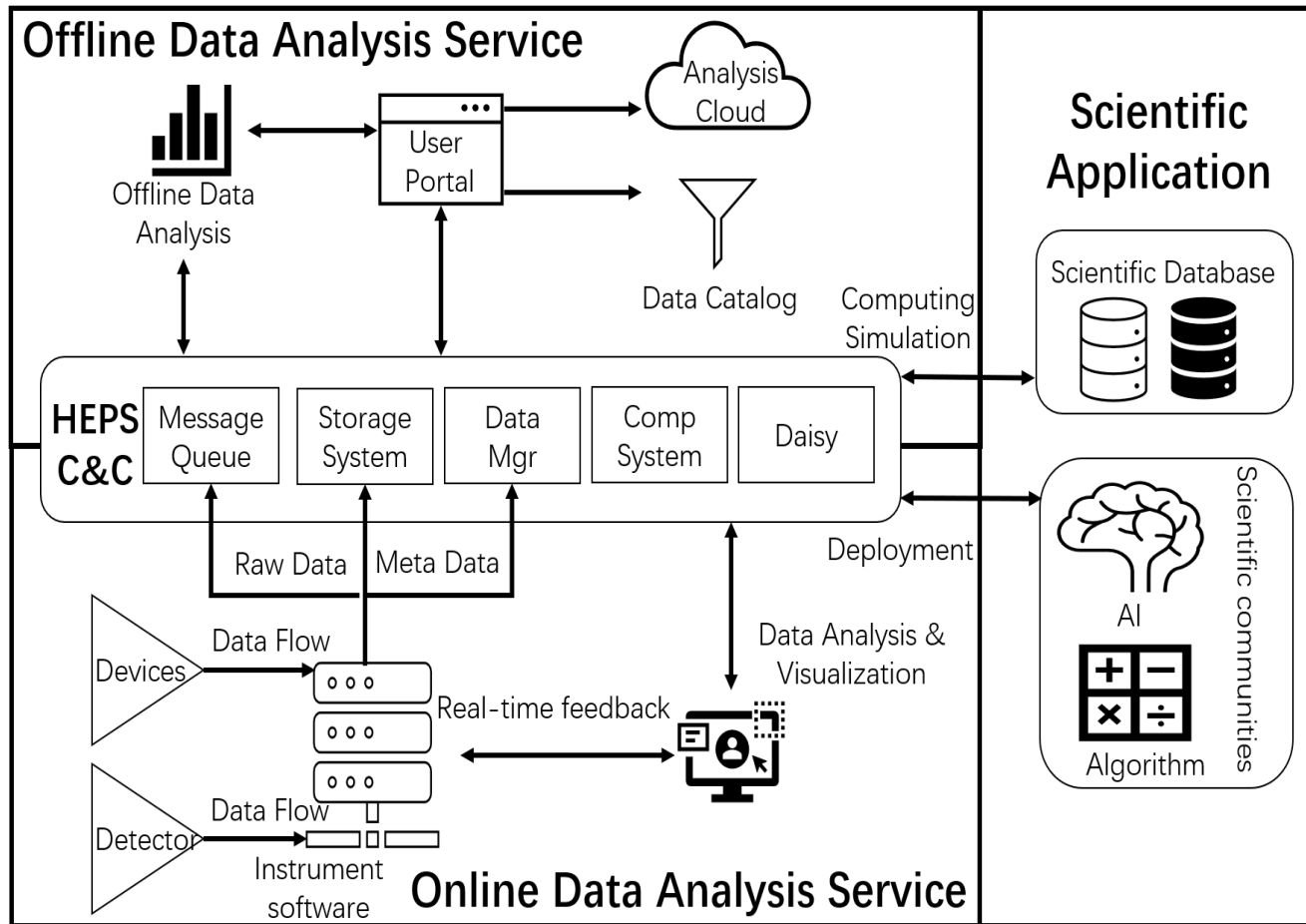


IT Services & Beamline Experiments



IT services are needed during the life-cycle of the Beamline experiments

Missions of Analysis Software



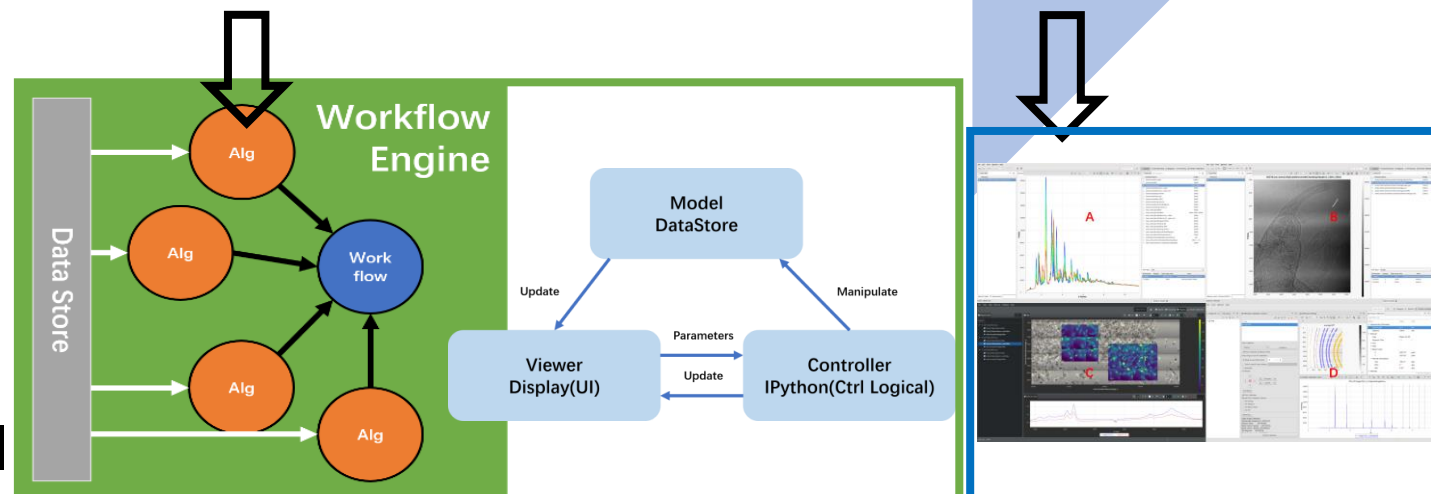
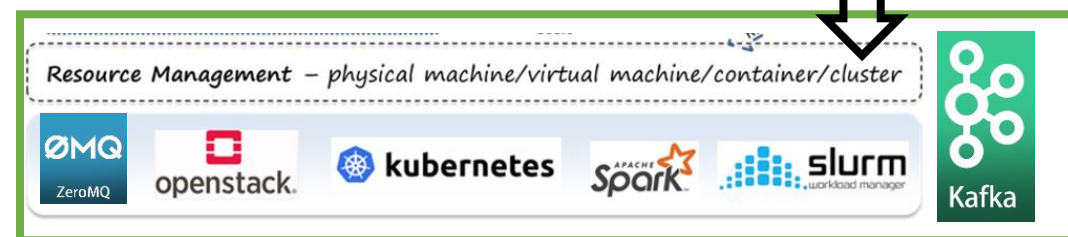
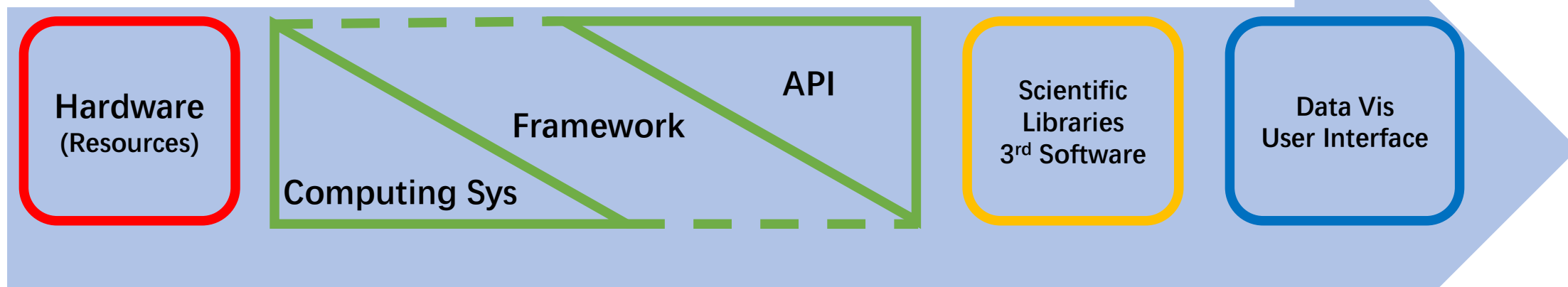
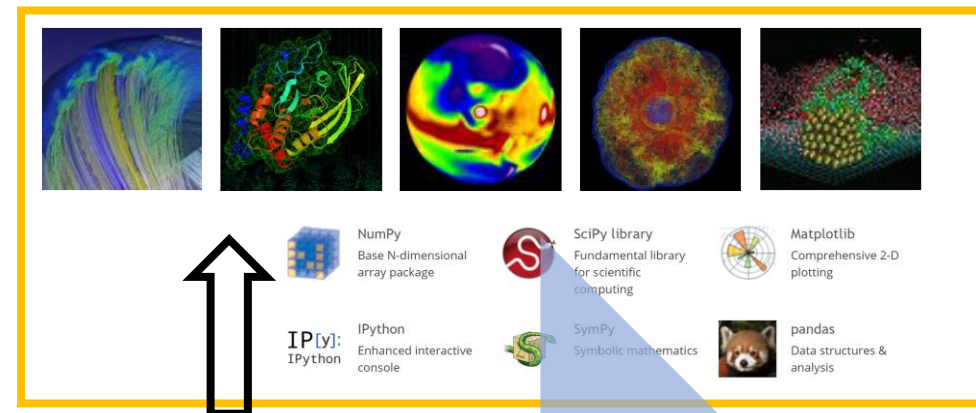
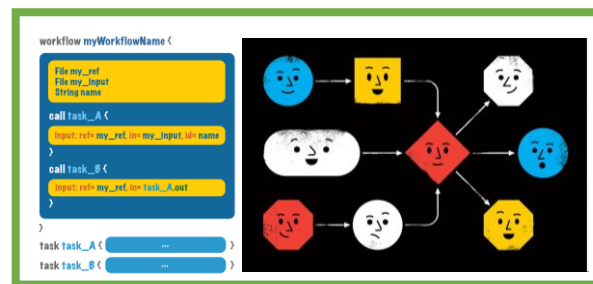
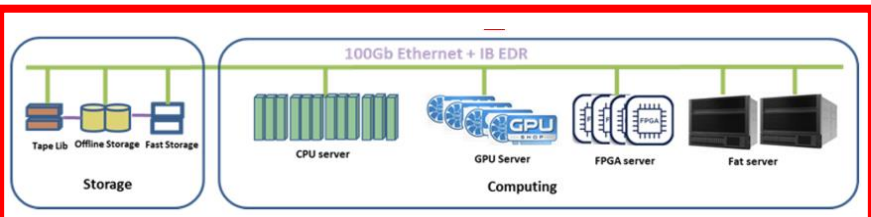
- **‘Online’ data processing and analysis**
 - Mission: Guide the experiment (visualize/estimate/characterize)
 - Files/Stream (whole/fraction)
- **‘Offline’ and remote data analysis**
 - Mission: Reduction, Reconstruction, Modelling and Simulation
 - Huge data volume (co-location of data storage and computing power)
- **Computing infrastructure**
 - HPC clusters (Spark/Slurm)
 - single workstation (Virtual machine)
 - Access: JupyterHub (notebook app)
 - Remote desktop (traditional app)
- **Software deployment and container**
 - Mission: Re-use and reproduce
 - The use of container to archive software environment

Overview

Missions & Requirements

Architecture and Design

HEPS Testbed



1. Hardware independence
2. A set of Standard Calls & GUI
3. Integration of 3rd Software

Inspired by Many Projects



Visualisation

Comprehensive visualisation of data in 1D, 2D and 3D



Development Environments

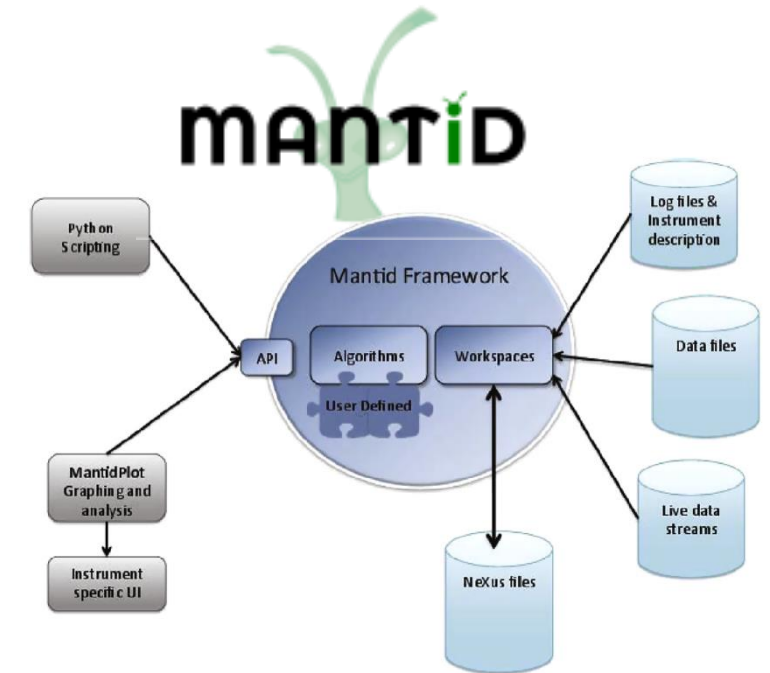
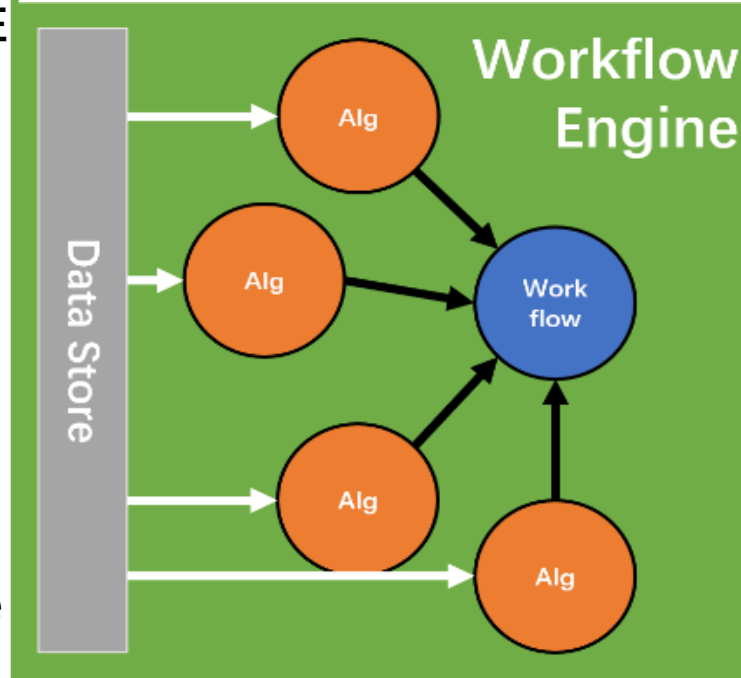
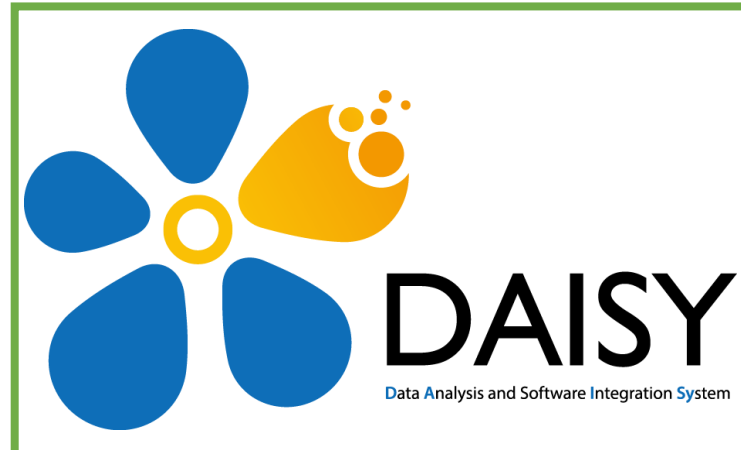
Integrated Python, Jython and Eclipse plug-in development, debugging and execution



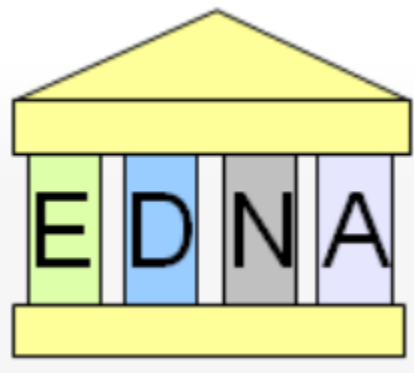
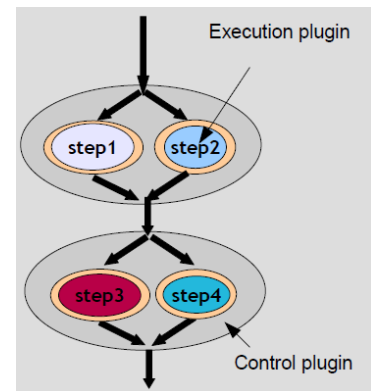
Processing and Workflows

for visual algorithms analysing scientific data

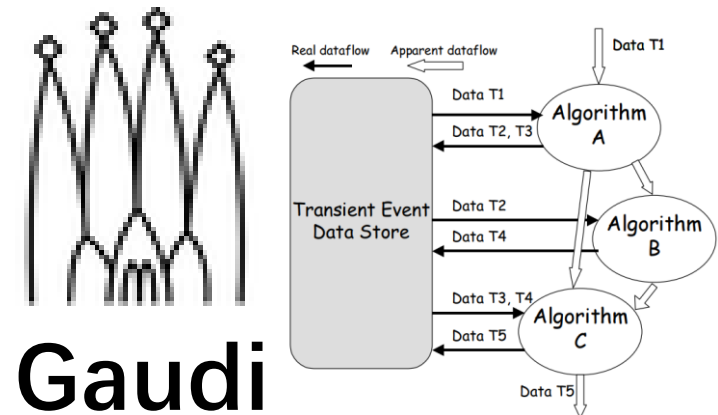
Data Visualization and Processing & IDE



Decouple Execution and Data Obj Module

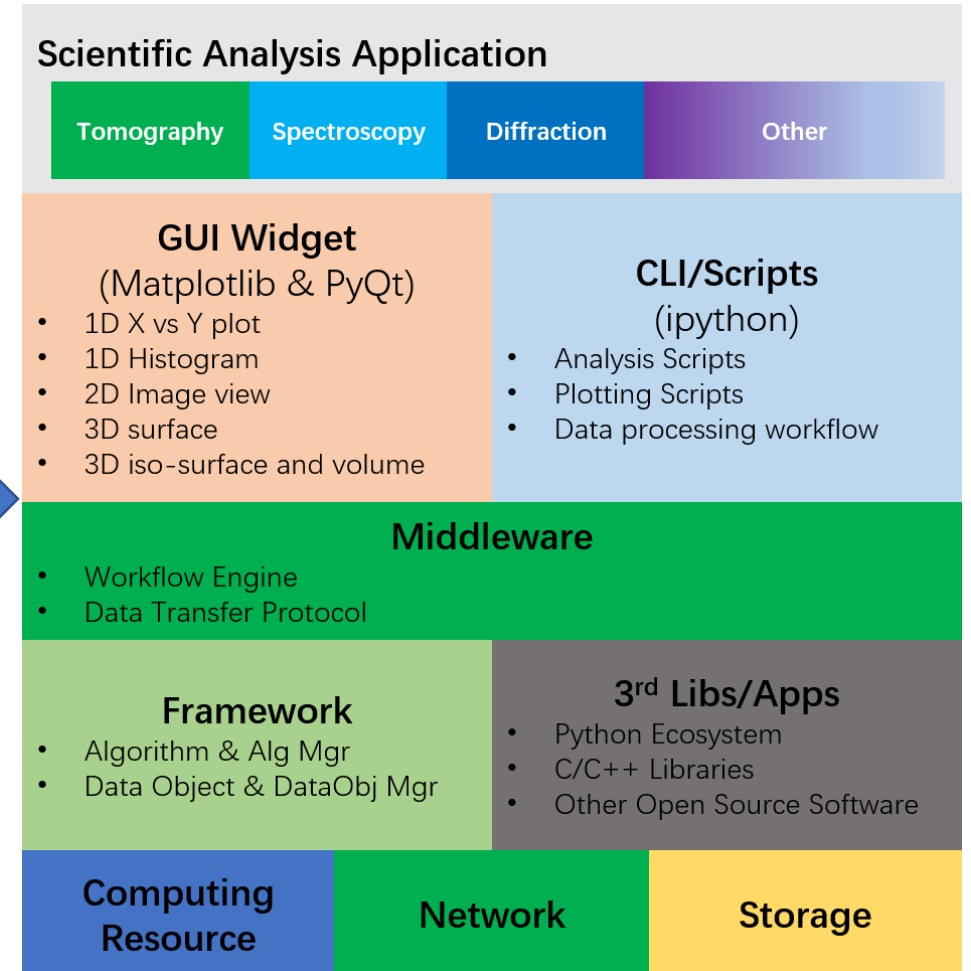
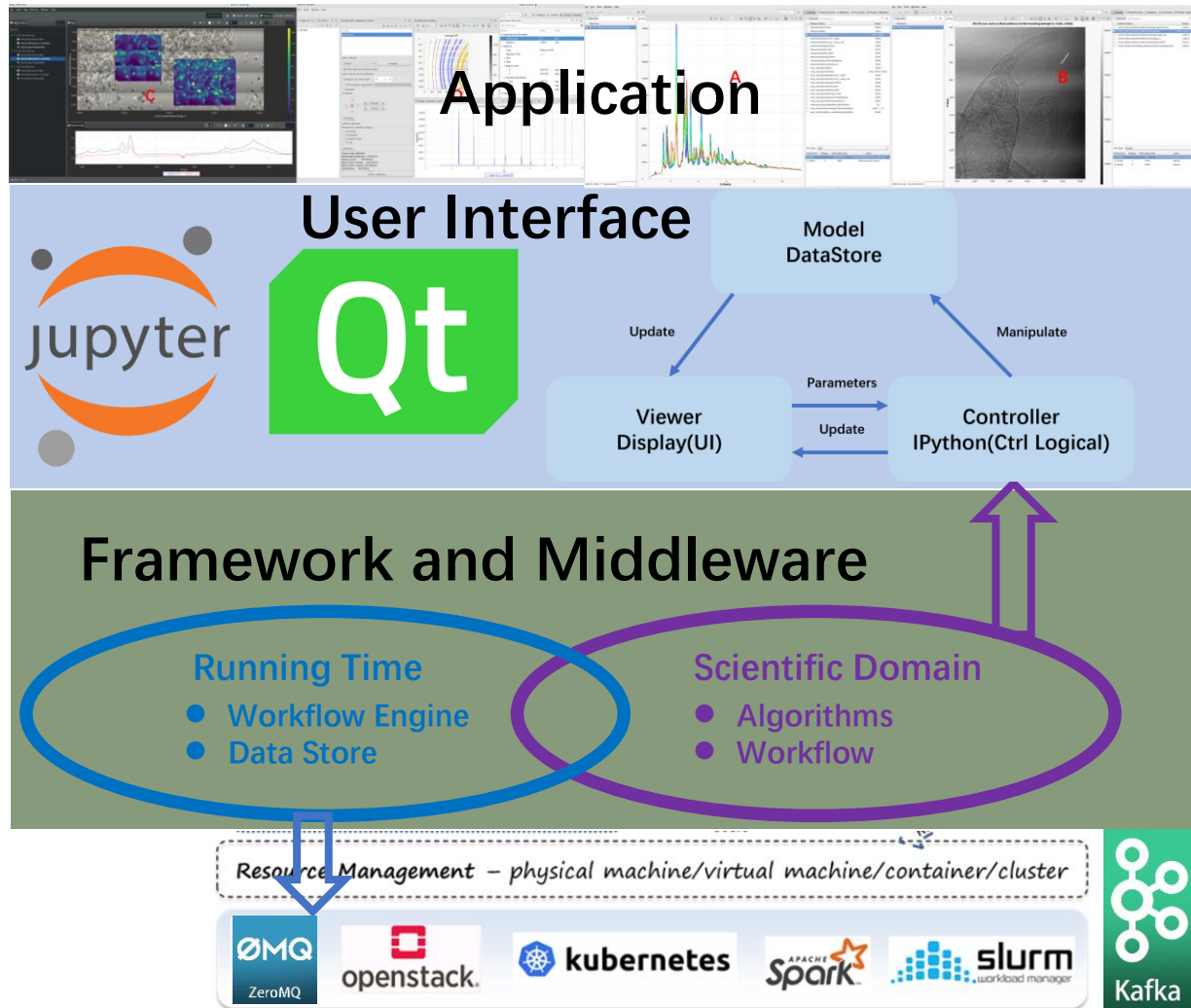


Decouple Control and Execution Module



Integrate with existing code via Algorithm

Software Framework Overview



Scientific Domain: Algorithm and Workflow

A

```
import numpy as np
import tomopy
from Daisy import DaisyAlg
class AlgTomopyRecon(DaisyAlg):
    def __init__(self, name):
        super().__init__(name)

    def initialize(self):
        self.data = self.get("DataStore").data()
        self.LogInfo("initialized, Tomopy Reconstruction")
        return True

    def execute(self, input_dataobj, theta, center, alg_type, output_dataobj):
        projs = self.data[input_dataobj]
        thetas = self.data[theta]
        dataobj = tomopy.recon(projs, thetas, center=center, algorithm=alg_type)
        self.data[output_dataobj] = dataobj
        return True

    def finalize(self):
        self.LogInfo("finalized")
        return True
```

B

```
@Daisy.Singleton
class WorkflowCTReconstruct(Daisy.PyWorkflow):
    def execute(self):
        self.engine['loadhdf5'].execute(input_path='/entry/tomo', output_dataobj='tomodata')
        self.engine['loadhdf5'].execute(input_path='/entry/dark', output_dataobj='darkdata')
        self.engine['loadhdf5'].execute(input_path='/entry/flat', output_dataobj='flatdata')
        self.engine['normalize'].execute(projs_dataobj='tomodata', darks_dataobj='darkdata', \
                                         flats_dataobj='flatdata', output_dataobj='normdata')
        self.engine['angles'].execute(input_dataobj='normdata', output_dataobj='thetas')
        self.engine['minuslog'].execute(input_dataobj='normdata', output_dataobj='mlogdata')
        self.engine['reconstruct'].execute(input_dataobj='mlogdata', theta='thetas', \
                                           center=1030, alg_type='fbp', output_dataobj='recodata')
        self.engine['savehdf5'].execute(input_dataobj='recodata', output_path='/entry/reco')

wf = WorkflowCTReconstruct('WorkflowCTReconstruct')
wf.initialize(workflow_engine='PyWorkflowEngine', \
              workflow_environment = init_dict, algorithms_cfg = cfg_dict)
wf.execute()

data = wf.data_keys()
algs = wf.algorithm_keys()

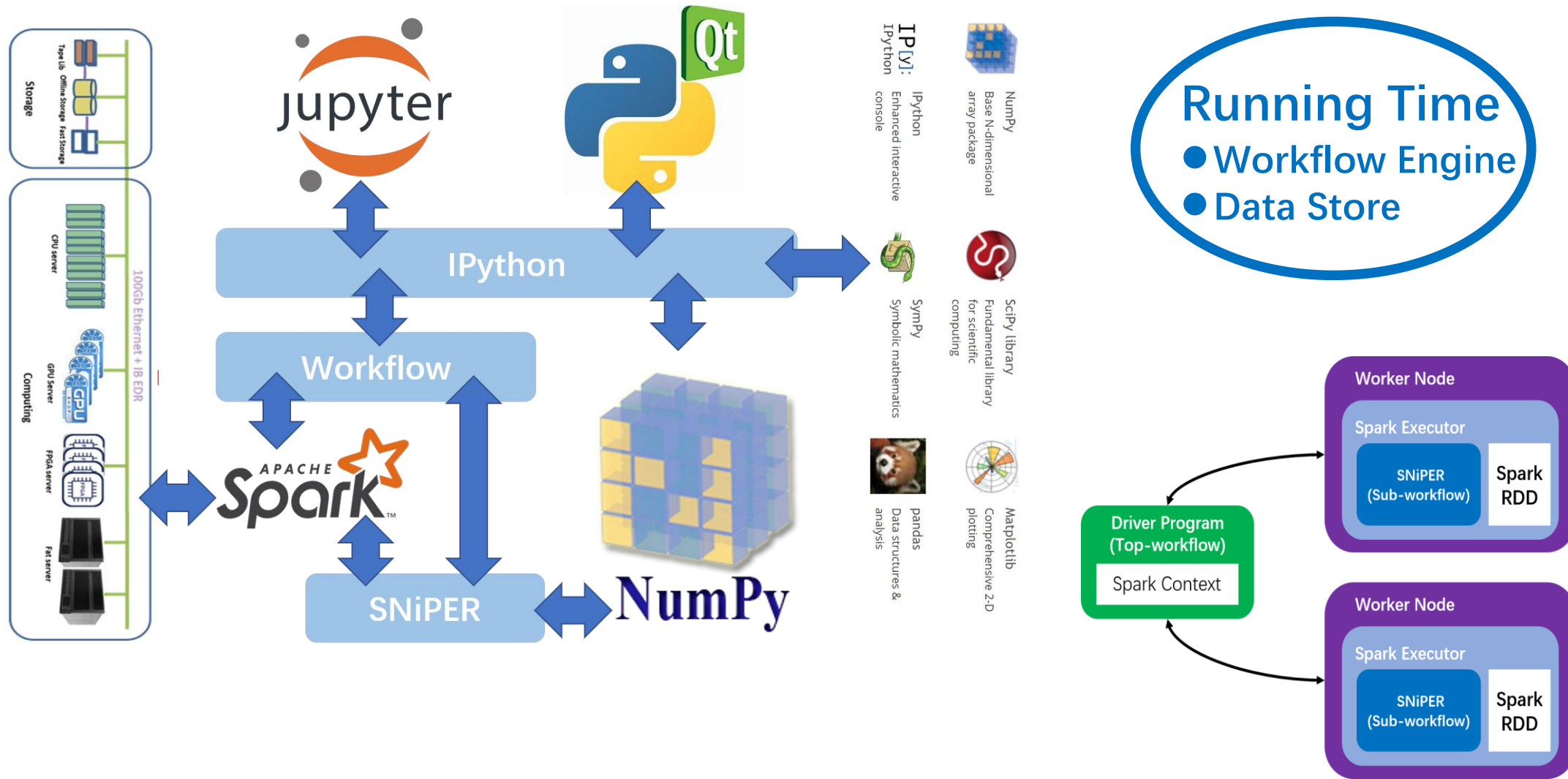
wf.finalize()
```

Scientific Domain

● Algorithms

● Workflow

Running Time Management: Workflow Engine and Data Store



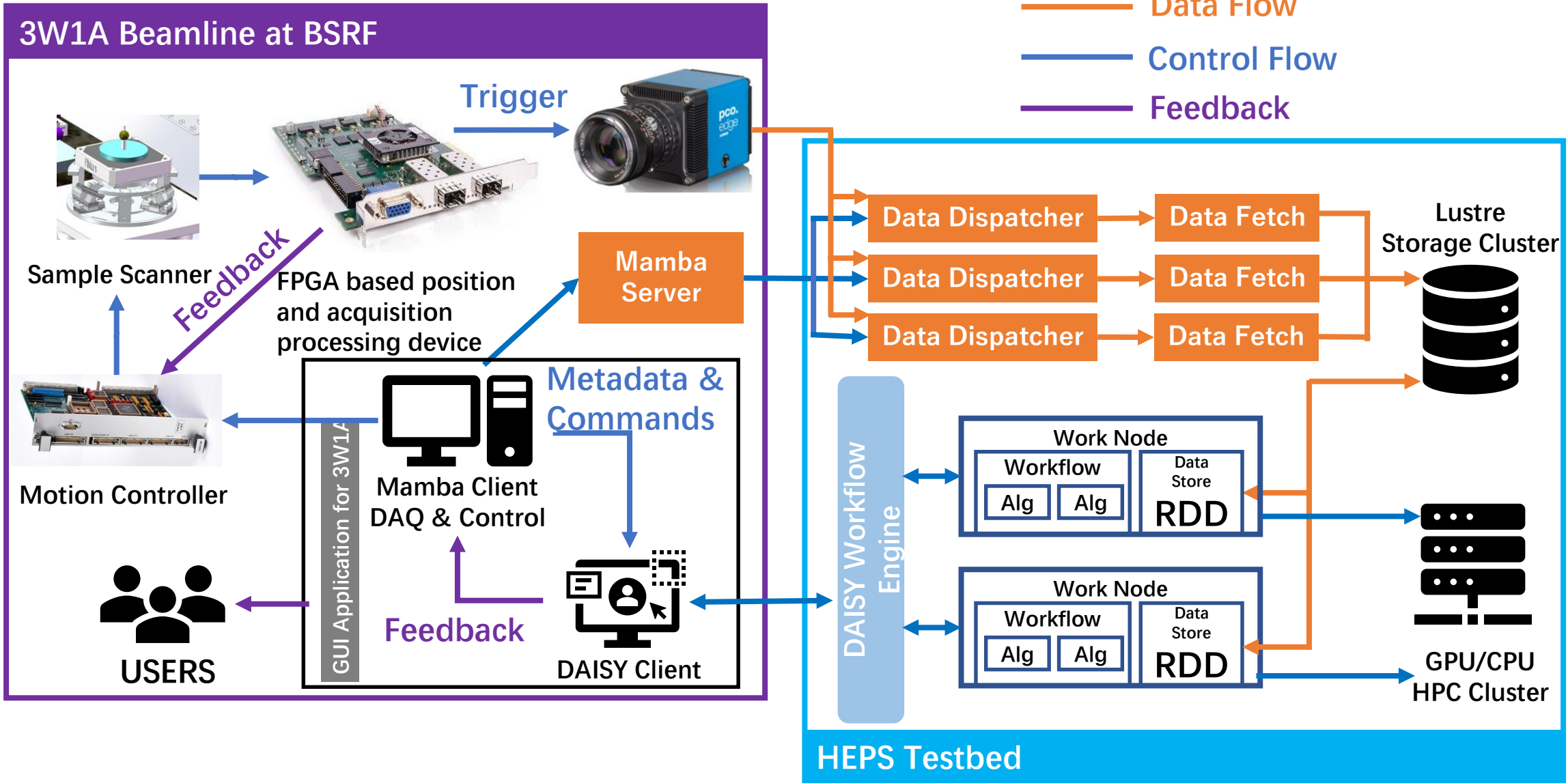
Overview

Missions & Requirements

Architecture and Design

HEPS Testbed

HEPS Testbed @ 3W1 at BSRF



The Workflow Configurations

```
init_dict = {'loaddata':{'class_name':'LoadHDF5',\
    'init_paras':{'inputfile_name':'scan_00575_data_000001.h5'}\
},\
'loadmask':{'class_name':'LoadHDF5',\
    'init_paras':{'inputfile_name':'scan_00575_master.h5'}\
},\
'azintalg':{'class_name':'AlgFAIIntegrate',\
    'init_paras':{'wavelength':'0.7293'}\
},\
'savedata':{'class_name':'SaveHDF5',\
    'init_paras':{'outputfile_name':'test_scan.h5'}\
}\
}
```

```
cfg_dict = {'azintalg':{'directDist':169,\
    'centerX':1049.967,\
    'centerY':1063.892,\
    'pixelX':75,\
    'pixelY':75,\
    'PlanRotation':64.66877,\
    'tilt':0.3753,\
    'azimin':93,\
    'azimax':115,\
    'radmin':10.0,\
    'radmax':20,\
    'nth':40}\
}
```

The Request for Resources

Resource Management – physical machine/virtual machine/container/cluster

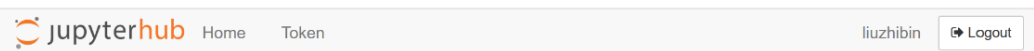


**Metadata/Data
Storage and Catalogue**

Refer to the [HEPS DMS report](#):

The Design of Data Management System(DMS) at HEPS

Reconstruction Service for Tomography



1. Login (JupyterHub)

Server Options

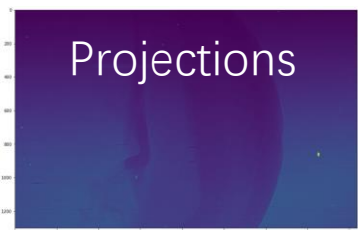
CT 3D reconstruction
CT 3D reconstruction service based on tomopy.

Deep Learning
Mainstream deep learning framework, contains tensorflow, pytorch, keras

Spark
Jobs can be submitted to the spark cluster

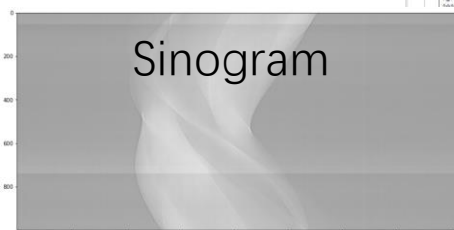
Start

3. Visualization (Jupyter notebook widget)



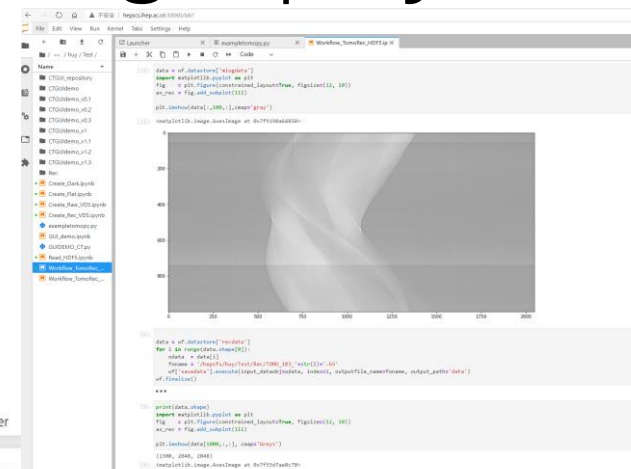
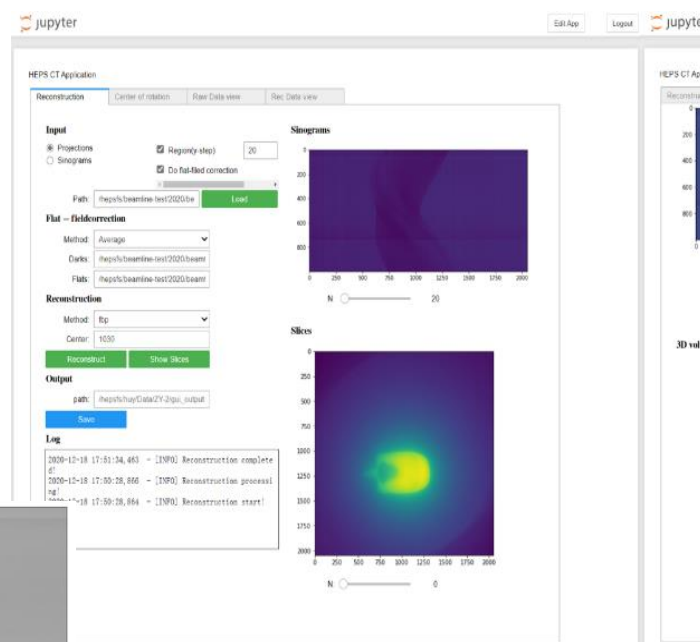
Projections

preprocessing



Sinogram

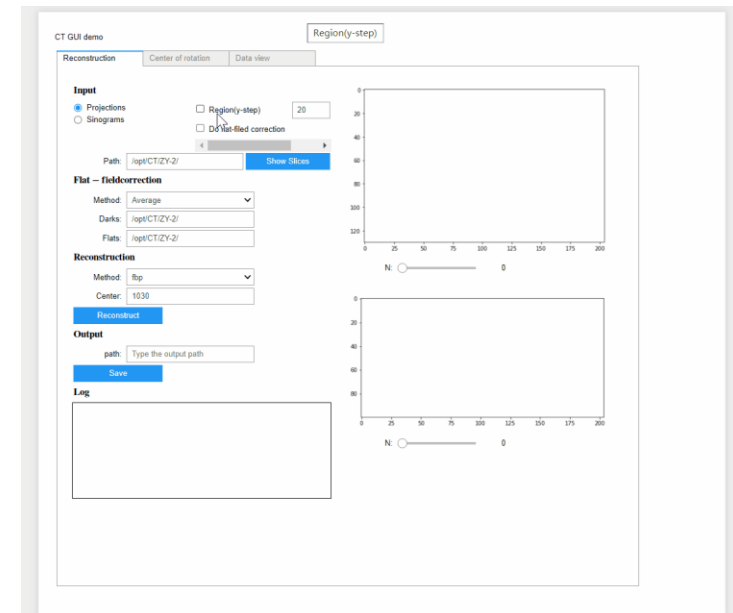
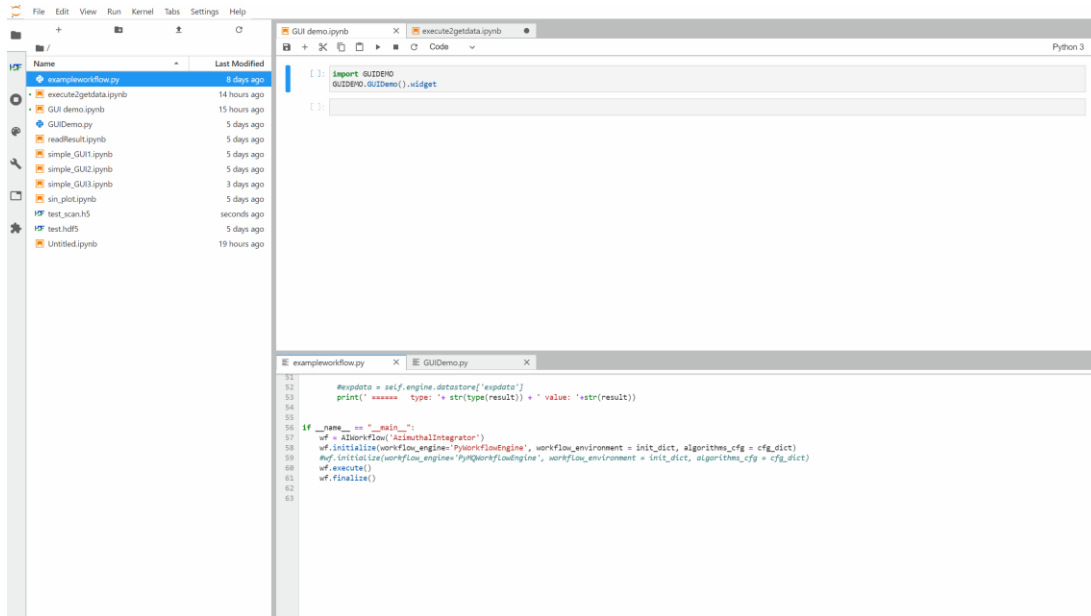
2. Workflow (JupyterLab)



reconstruction

CT Slices

Demo: PyFAI & Tomopy



The background features a large, stylized circular graphic composed of multiple concentric rings. The rings on the left are in shades of blue, while the rings on the right transition into shades of green. The word "Thanks" is centered within the white space of the circle.

Thanks