



# Neutrino 2026, University of California, Irvine, United States

## Offline Software and DCI System of JUNO Experiment



Tao Lin<sup>2</sup>, Zhengyun You<sup>1</sup>, Xiaomei Zhang<sup>2</sup>, Yuning Su<sup>1</sup> (On behalf of the JUNO collaboration)  
 1. Sun Yat-sen University 2. IHEP

### 1. JUNO Experiment



Fig. 1 The location of JUNO experiment site

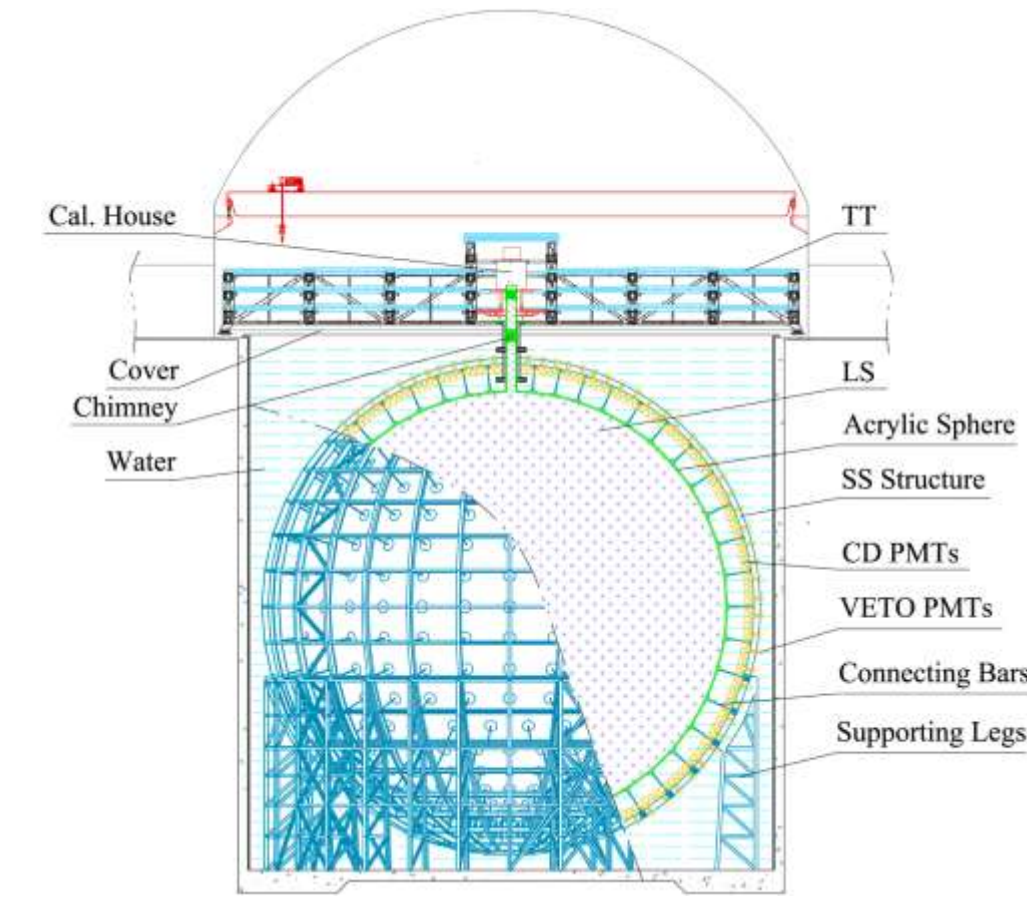


Fig. 2 The structure of JUNO detector

- ◆ The Jiangmen Underground Neutrino Observatory (JUNO) is located in Kaiping City, Jiangmen, Guangdong Province, China, around 53 km from Yangjiang and Taishan nuclear power stations.
- ◆ The primary scientific goals of JUNO:
  - Determine neutrino mass ordering at  $3\sigma$  significance with  $\sim 7$  years of data-taking.
  - Achieve precise measurements of neutrino oscillation parameters.
  - Observe and study supernova neutrinos, solar neutrinos, atmospheric neutrinos and geo-neutrinos, as well as conduct other cutting-edge scientific researches.

### 2. JUNO Offline Software System

- ◆ Introduction
  - Data processing and analysis of neutrino experiments differ from those in collider experiments, i.e. Reactor antineutrinos are detected mainly through the inverse beta decay (IBD) process.

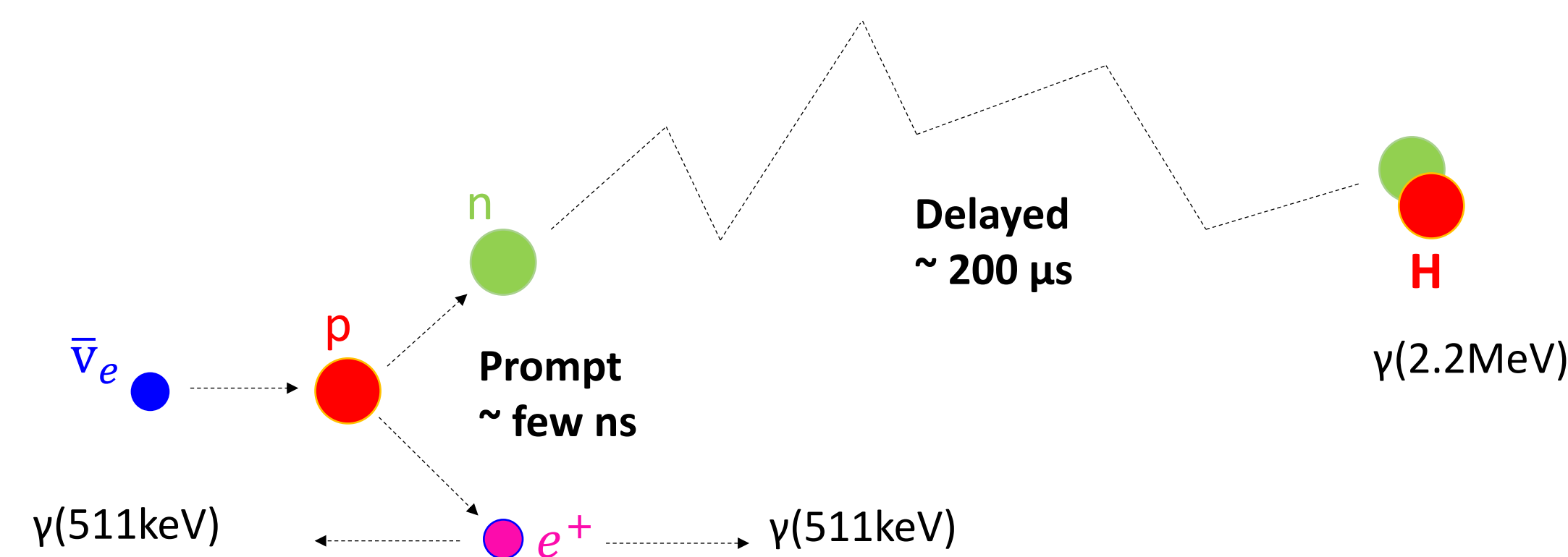


Fig. 3 Diagram of Signal Generation in IBD Process

- Since 2012, the JUNO Offline Software system (JUNOSW) has been developed to meet the simulation, reconstruction, calibration, and analysis needs of the JUNO experiment.
- It is based on the SNIKER (Software for Non-Collider Physics Experiment) framework.

#### ◆ Core Software:

- ✓ Framework
- ✓ Event Data Model
- ✓ Geometry Management
- ✓ Database System
- ✓ Analysis Framework

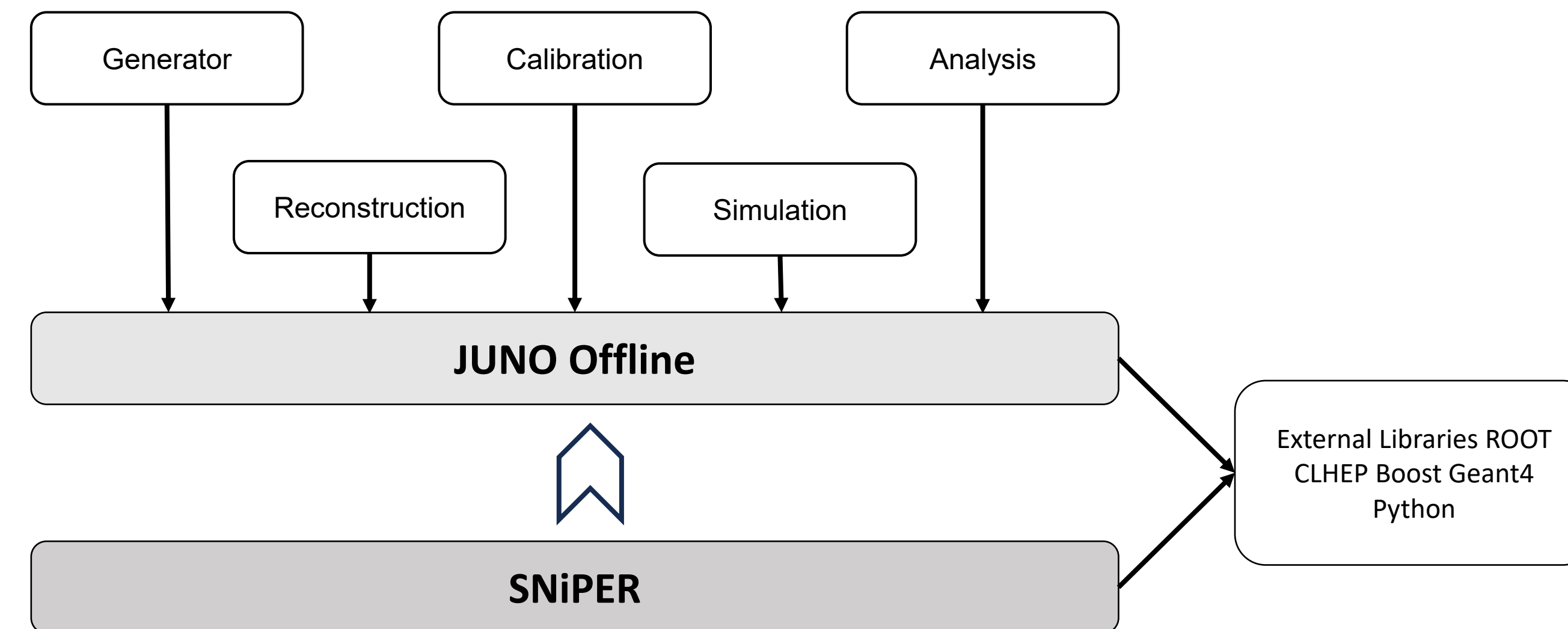
#### ◆ Application:

- ✓ Simulation
- ✓ Reconstruction
- ✓ Calibration
- ✓ Physics Analysis

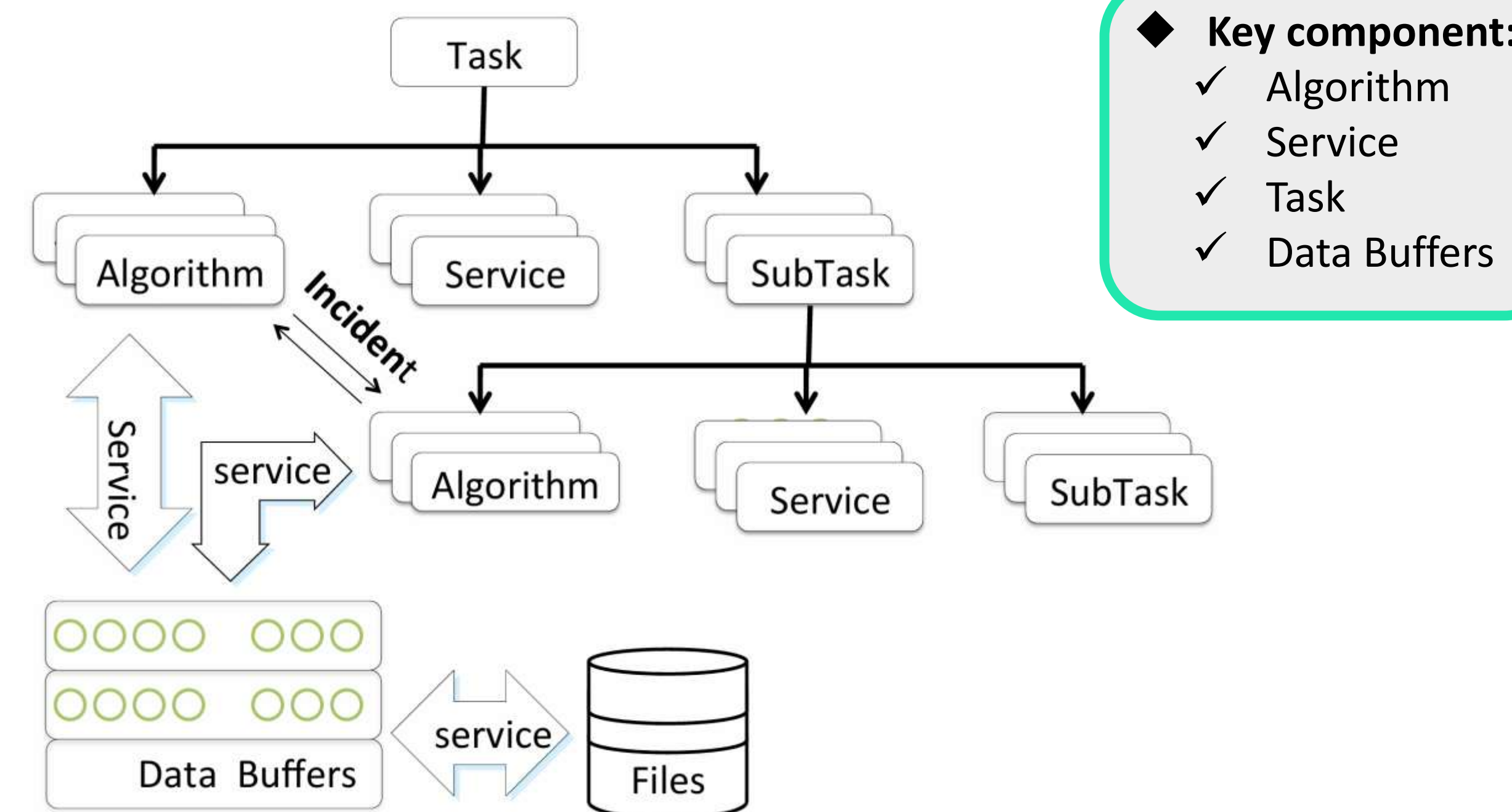
#### ◆ Modern Technologies:

- ✓ C++17
- ✓ Python3
- ✓ cmake
- ✓ gitlab

### 2. JUNO Offline Software System



- ◆ SNIKER Framework: a general-purpose software framework
  - Lightweight: depends mainly on Boost.Python.
  - Highly modularized and extensible
  - Developed for JUNO and later adopted by several other experiments, including LHAASO, STCF, HERD, and nEXO.



- ◆ Key component:
  - ✓ Algorithm
  - ✓ Service
  - ✓ Task
  - ✓ Data Buffers

### 3. DCI System

- ◆ Why do we need DCI (Distributed Computing Infrastructure) system?
  - JUNO is an international experiment whose computing resources are provided by multiple data centers worldwide. (IN2P3, IHEP, JINR, CNAF, SDU.....)
  - To use resources and share data efficiently across distributed data centers, the DCI system was developed to:

- ✓ integrate heterogeneous resources
- ✓ hide complexity from users
- ✓ provide a simple way for users to use distributed resources with:
  - Global identity and a unified user interface

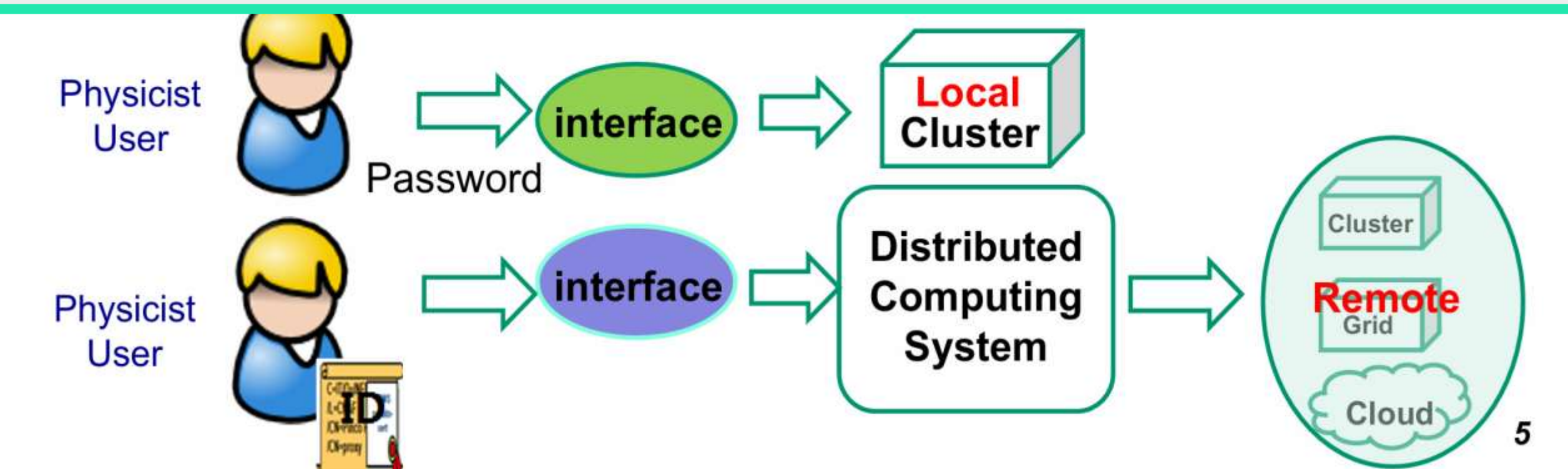


Site	Site type	MaskStatus
GRID-IN2P3.fr	GRID	Active
CLUSTER-SDU.cn	CLUSTER	Active
CLOUD-IHEP.CLOUD.cn	CLOUD	Active
GRID-IHEP.cn	GRID	Active
CLOUD-IFN-PADOVAN.it	CLOUD	Active
GRID-JINR-CONDOR.ru	GRID	Active
GRID-IFN-CNAF.it	GRID	Active

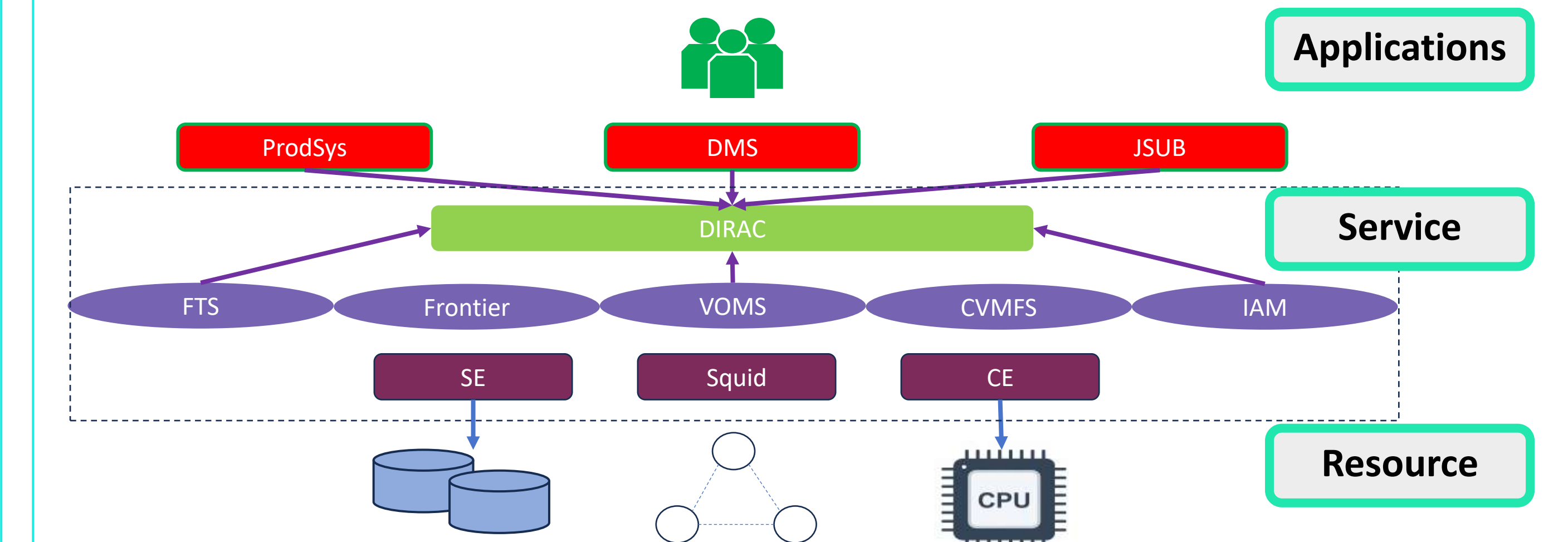
### 3. DCI System

➢ DCI vs. local cluster:

- ◆ Similar procedure:
  - ✓ Need authentication to tell the system who you are
  - ✓ Interface or commands to submit jobs or access data
  - ✓ Access JUNO software from CVMFS
- ◆ Differences:
  - ✓ In DCI, jobs are running in remote resources
  - ✓ In DCI, data is accessed from remote storage



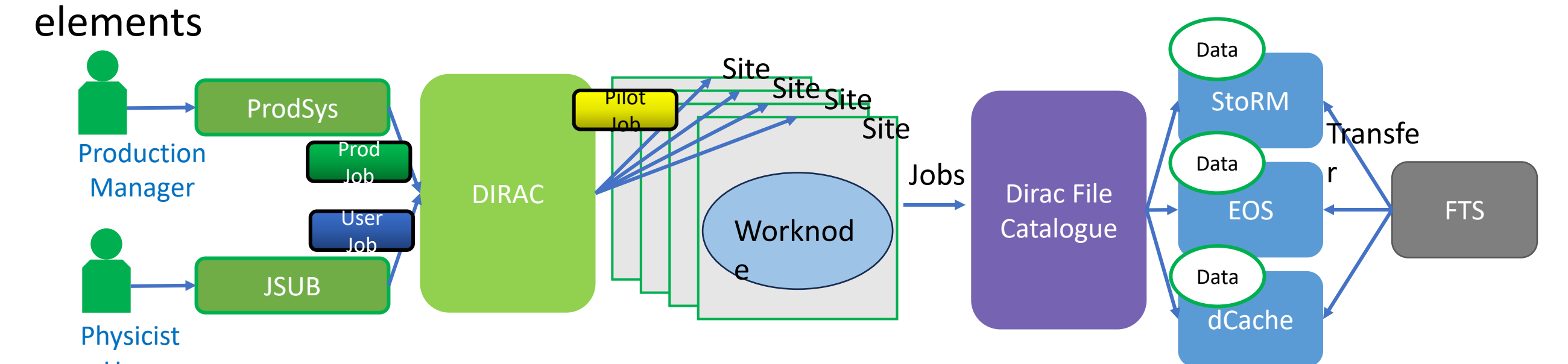
◆ How does DCI system work?



- The DCI system is built on DIRAC, leveraging both its WMS (Workflow Management System) and DMS (Data Management System).
- the infrastructure is structured into three layers: resource, service, and application.

◆ How do users use the DCI system?

- ✓ Get Grid Certificate and join JUNO VO group
- ✓ Set up DCI environment and prepare proxy with cert
- ✓ Use JSUB to create and submit jobs to the DCI, Check job status through JSUB or DIRAC.
- ✓ Output data are registered in the DIRAC File Catalogue and can be accessed from storage elements



[1] X. Huang et al. (JUNO Collaboration), Offline Data Processing Software for the JUNO Experiment, PoS ICHEP2016, 1051 (2017).  
 [2] X. Zhang (JUNO Collaboration), JUNO distributed computing system, EPJ Web Conf. 295, 04030 (2024).