



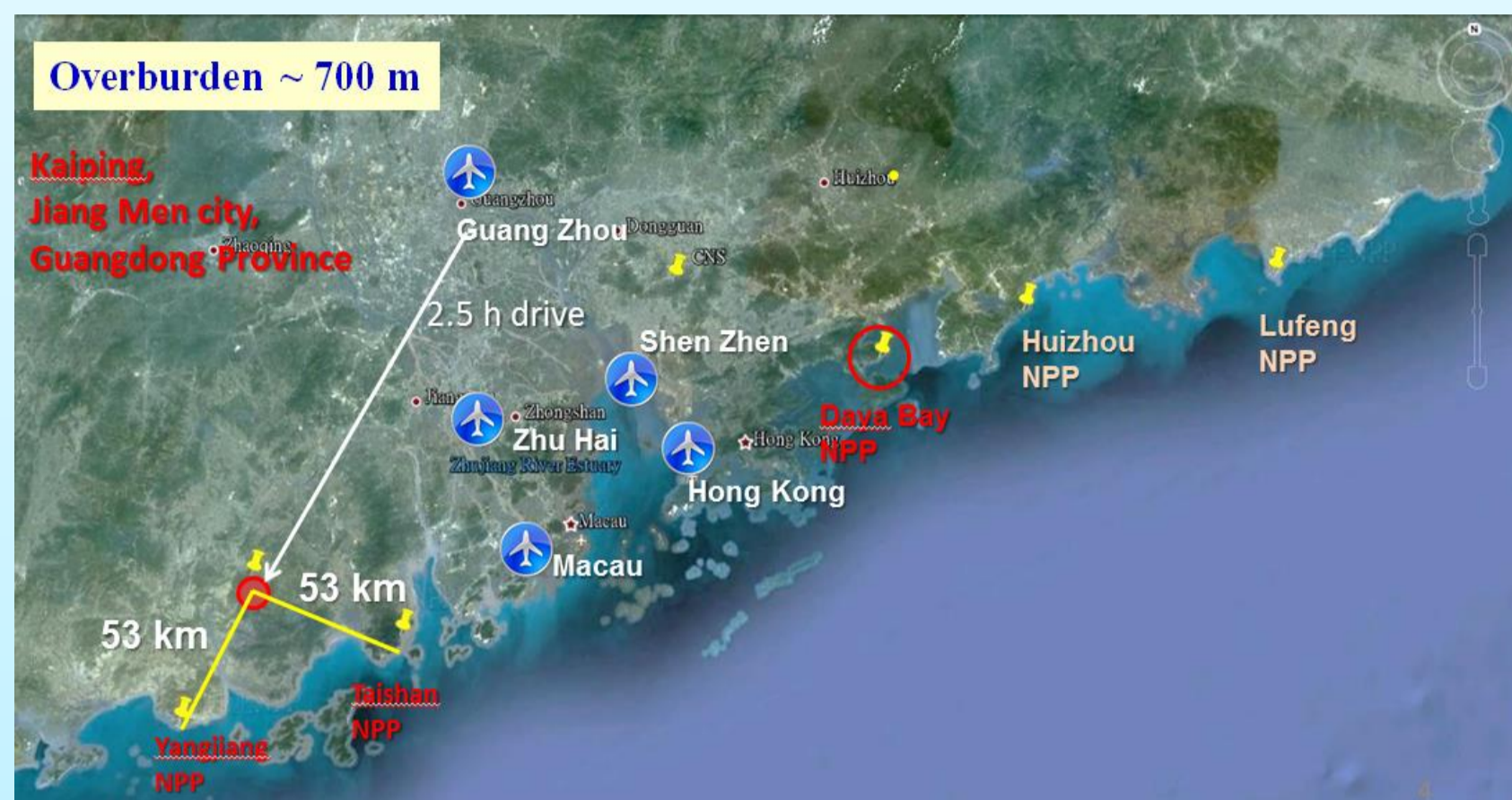
# A SOA Based Design of JUNO DAQ Online Software

Jin Li<sup>1,2</sup>, Minhao Gu<sup>1</sup>, Fei Li<sup>1</sup>, Kejun Zhu<sup>1,2</sup>  
 On behalf of the JUNO collaboration



1State Key Laboratory of Particle Detection and Electronics, Institute of High Energy Physics, CAS, Beijing, 100049, China  
 2University of Chinese Academy of Sciences, Beijing 100049, China (Email: [lifei@ihep.ac.cn](mailto:lifei@ihep.ac.cn); [lijin@ihep.ac.cn](mailto:lijin@ihep.ac.cn))

## Introduction



### JUNO:

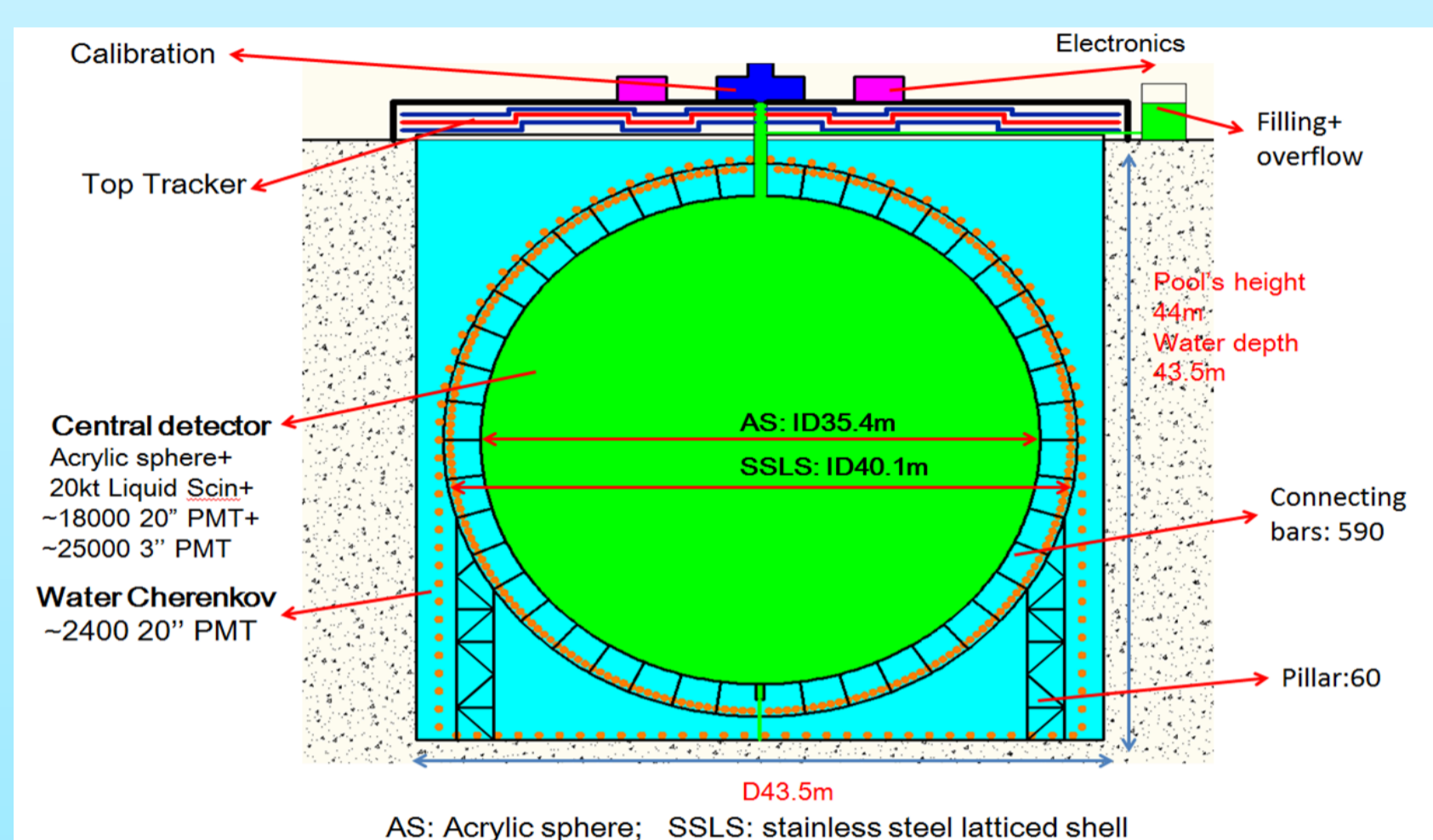
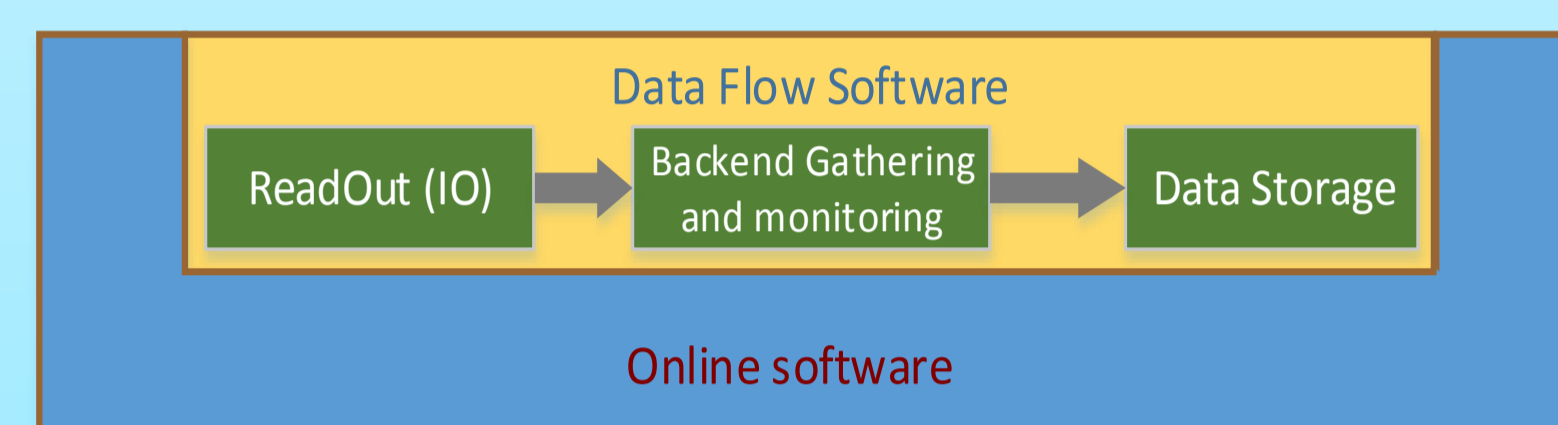
- determine neutrino mass hierarchy, precisely measure oscillation parameters
- the largest liquid scintillator detector with the best energy resolution in the world
- central detector with 18000 20" PMTs and 25000 3" PMTs, Water Cherenkov with 2400 20" PMTs

Since the numerous PMTs and large-scaled front-end electronic readout channels, DAQ will be:  
**Large scaled with thousands of software processes distributed over several hundreds of computing nodes**

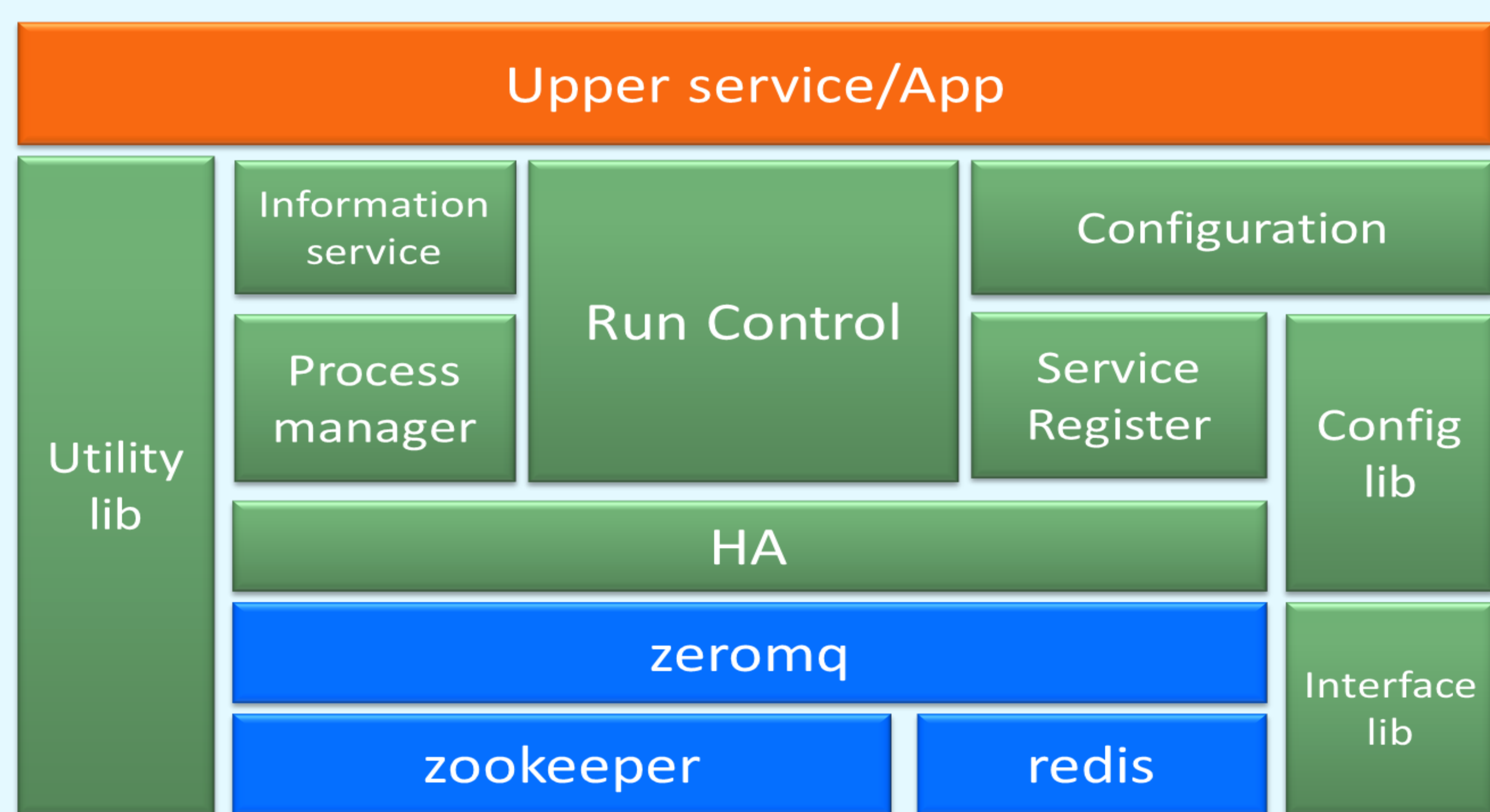
**Online Software** : A distributed system which is designed to control and monitor the whole DAQ system, plays a important global role during the whole data taking period, including configuring, monitoring, controlling, multi-processes management, information sharing...

### Online Software design:

- SOA : service oriented architecture--loosely-coupled, modular, reusable
- Message exchange--ZeroMQ
- PARTITION : DAQs run concurrently
- HA : high availability--Zookeeper
- Other functional modules: Configuration, Run Controller, Information sharing...



## Online Software architecture Design

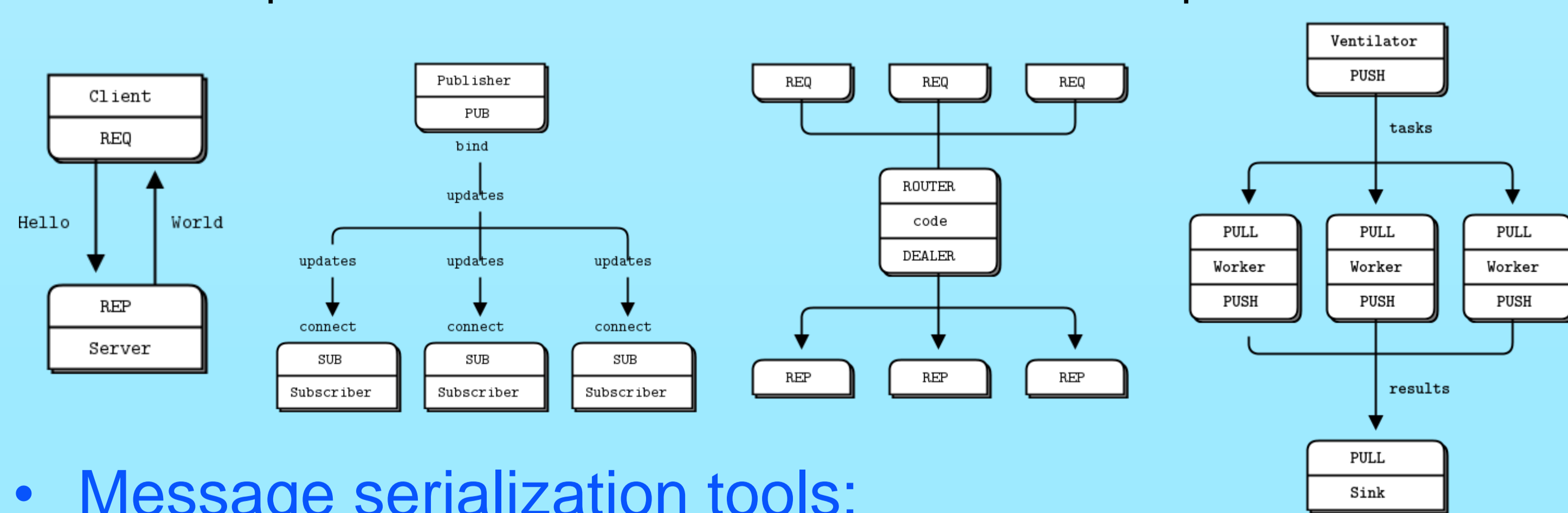


### The total DAQ architecture:

- Green : framework services, SOA based framework of the online software-- Run Control, Information service...
- Orange: user parts, Data flow system
- Blue : underlying layer-- ZeroMQ Zookeeper Redis...
- SOA: deployment methodology relying on the integration and interaction between loosely coupled services.
- Any service could be service provider or service consumer.
- Message exchanges among all services using TCP/IP encapsulated by ZeroMQ.

### Underlying communication layer: ZeroMQ

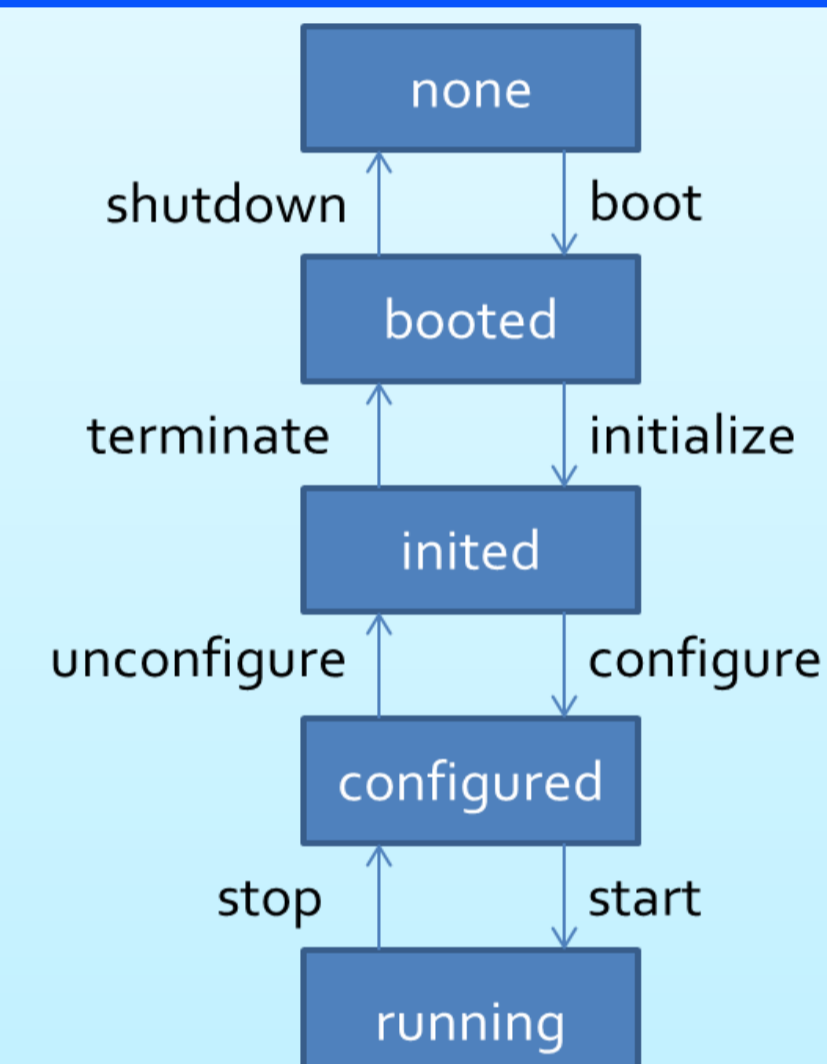
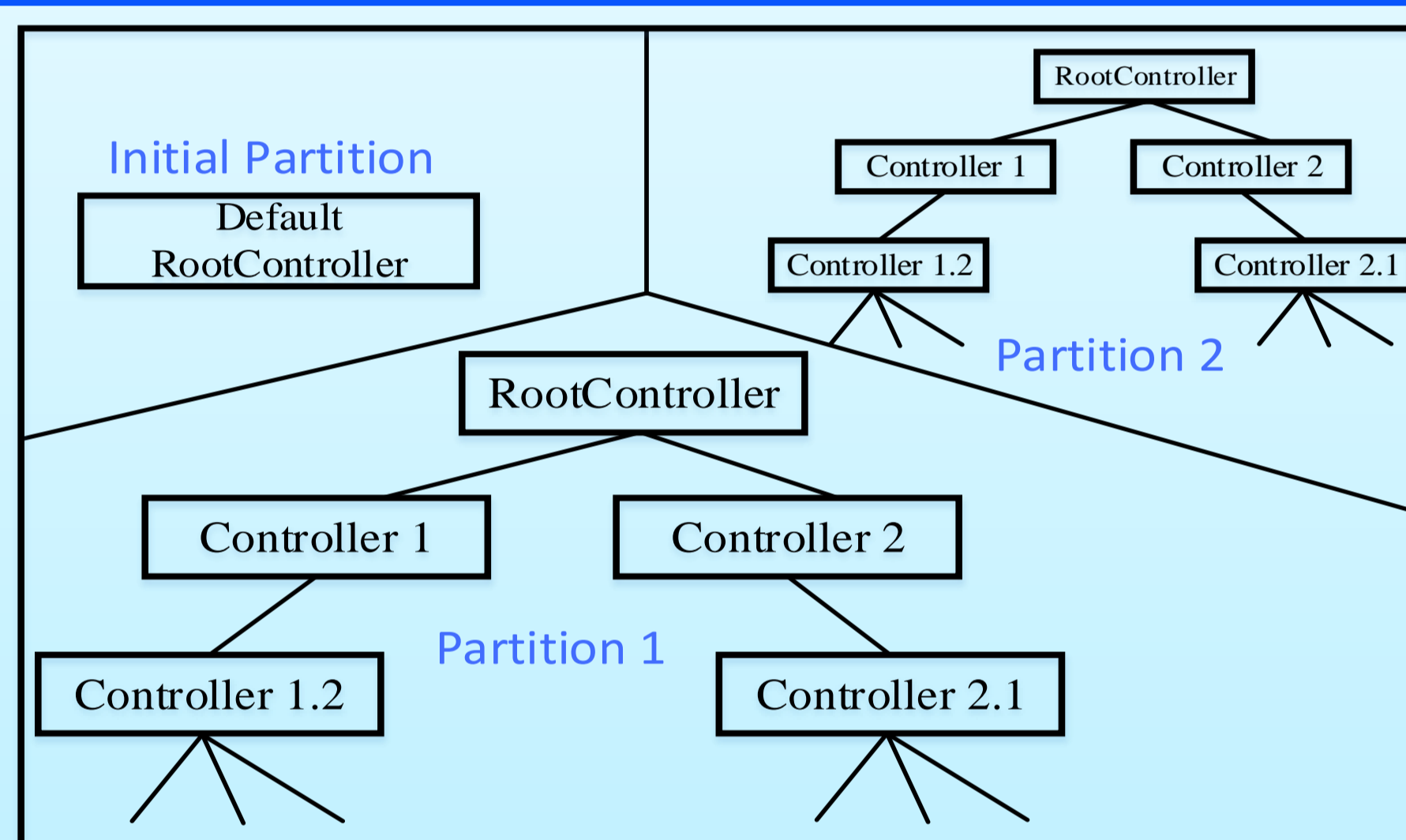
- ZeroMQ seems like traditional Berkeley socket, encapsulates low-levels statement and error handling complexities.
- advantages: zero brokers, low latency, cross-platform and support of multiple widely used programming languages.
- ZeroMQ provides several advanced communication patterns:



### Message serialization tools:

- Google Protocol buffers: serializing structured data
- Easy, language-neutral, platform-neutral

## Tree-like controller:



- Intention:** Multiple DAQs run concurrently, Operate a sub-set in stand-alone mode, Command dispatch and state collection
- The top level RootController taking the roles of the overall control and coordination of the system.
- Each controller level controls the next level controllers.
- State: each controller will wait a timeout time unless all children's state have been received and then switch the state.
- The 'Initial Partition' is a common component that accomplish initialization and public function.

## Information service:

- Histograms, event rates...
- Any DAQ components could publish its information to IS, and anyone could query information from IS.
- Redis is a good choice to act as database for its high performance:

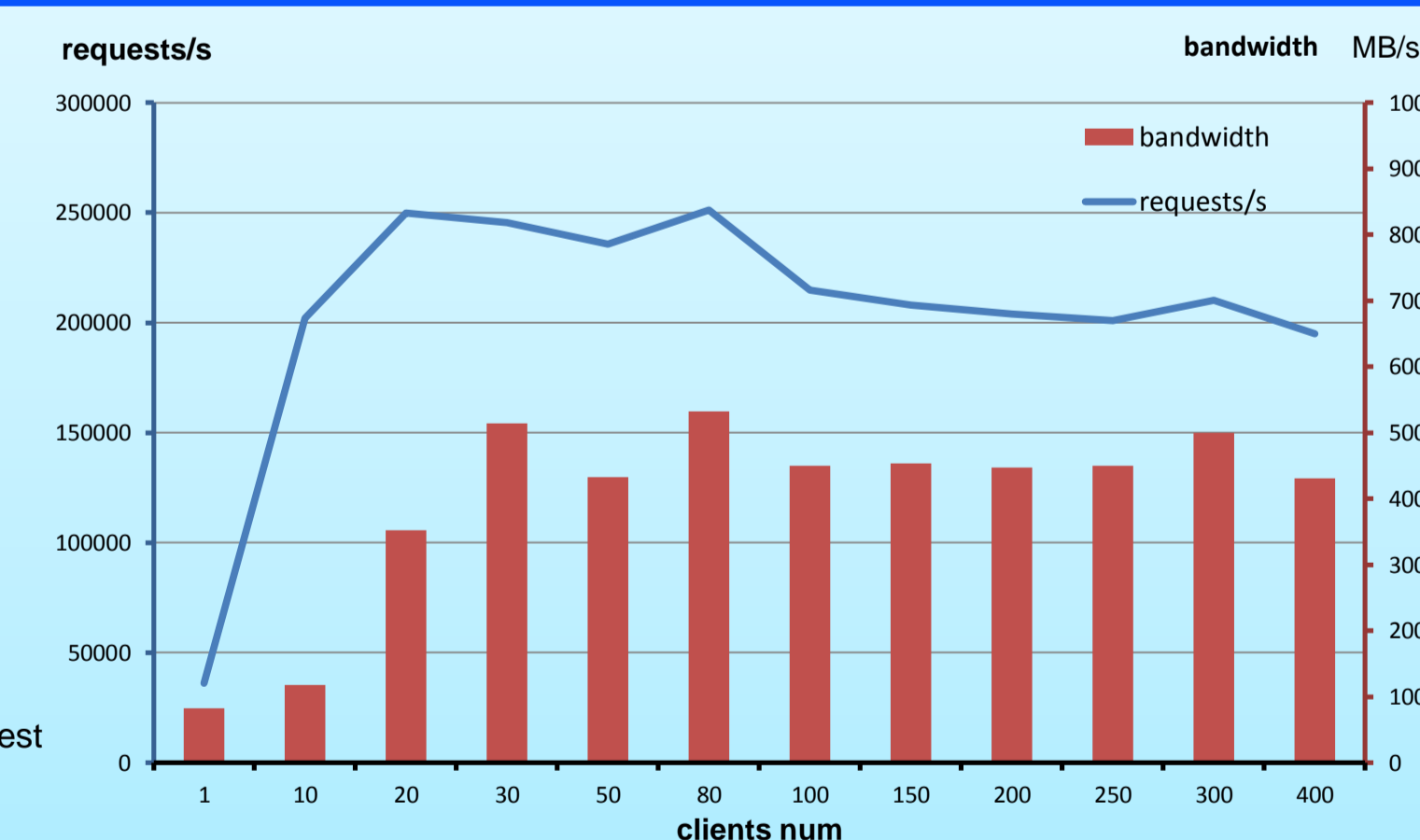


Fig : The blue line is the number of command which were successfully processed and returned per second for n clients. The red histograms are the Redis server's bandwidth for n clients. The test environment is: the length of each request is 2000 Bytes, and 4 commands are wrapped in one request which is also called pipeline =4.

### Load balance:

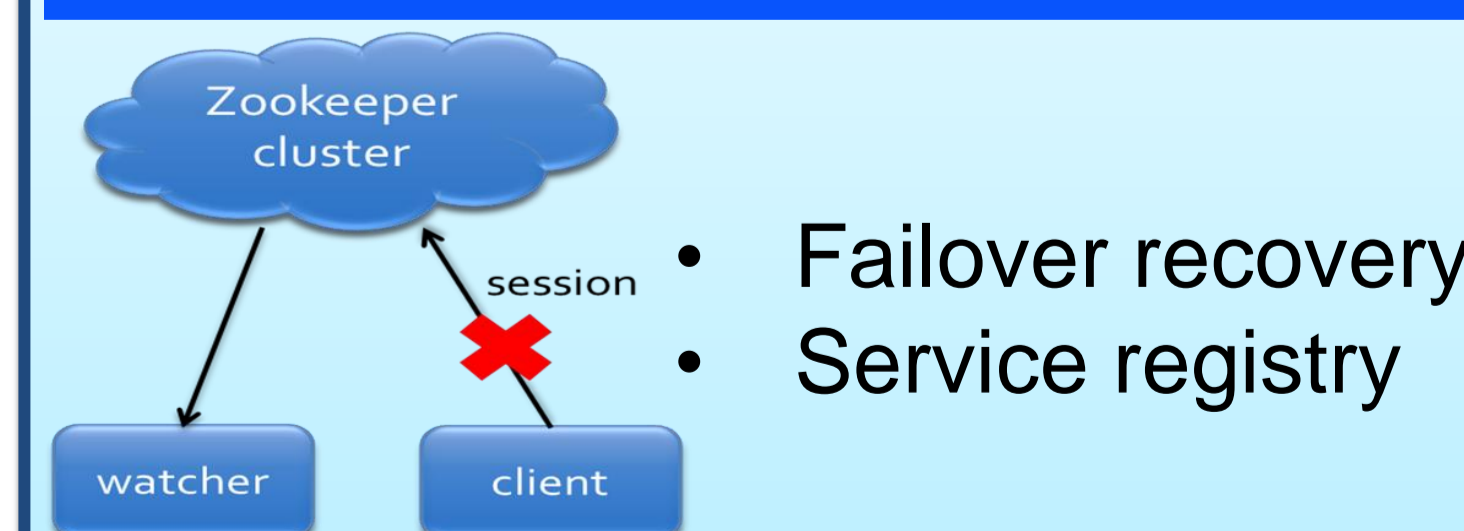
- Static way: specific define which IS server the information will send to.
- Dynamic way: dispatcher server determines in running time.

## Configuration:

```
<object type="Partition" id="part_tst">
  <rel name="Segment" type="segment" id="ros_seg" />
  <rel name="Segment" type="segment" id="eb_seg" />
  <rel name="Segment" type="segment" id="dfm_seg" />
  <rel name="Segment" type="segment" id="setup" />
  <rel name="IsControlledBy" type="service" id="RootController" />
  <rel name="InfoDestTo" type="service" id="def_iss"/>
</object>
```

- User Interface: XML format with XSD parser
- Configure Services provides queries about partition path, id, type, and relationship between levels that defined in XML file.
- Key words: *IsControlledBy*, *InfoDestTo*...

## HA module:



- Failover recovery
- Service registry
- All services have master/slave to ensure the failover recovery.
- The failover is monitored by zookeeper.

## Summary

- The architecture design of JUNO DAQ Online software has been achieved.
- The core functions have been accomplished and the whole system runs well with dummy data.
- The JUNO onsite is still under construction, however. So the design is in laboratory and the further development and optimization is ongoing.