Overview of PICKSC Science Gateway

Qiyang Hu, Frank Tsung

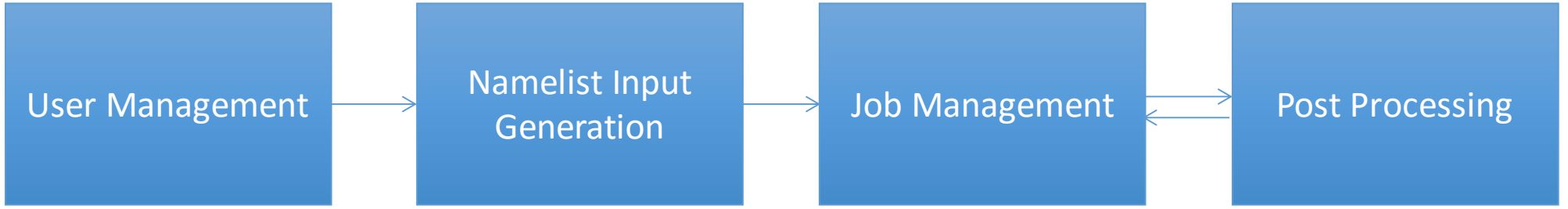
September 19, 2017

Introduction

- Gateway = Web Graphical User Interface:
  - PIC software: BEPS, OSIRIS
  - For developers, users, educators and students

- A proof-of-concept implementation
  - From scratch
  - Ready to expand

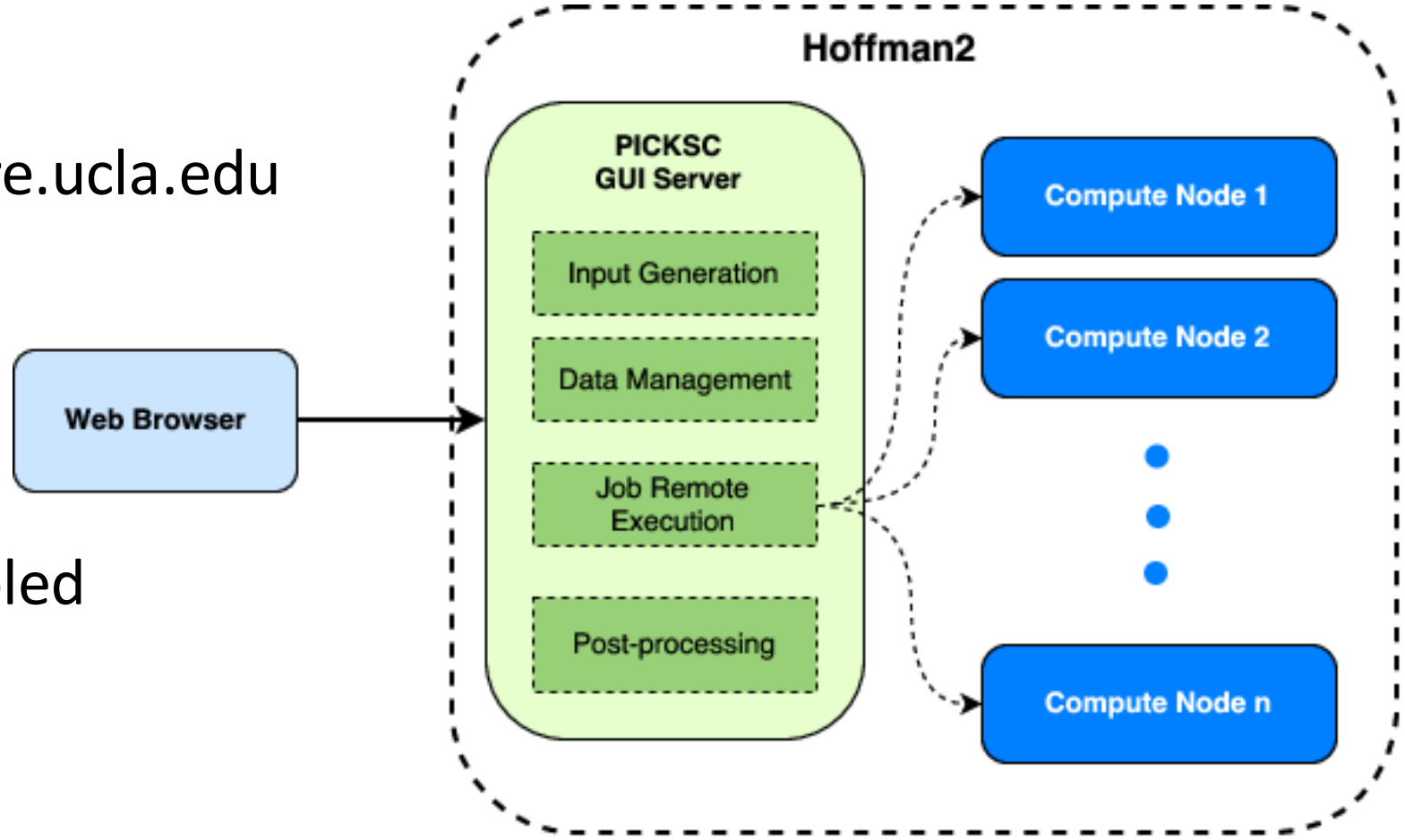
Work Modules for PICKSC Science Gateway



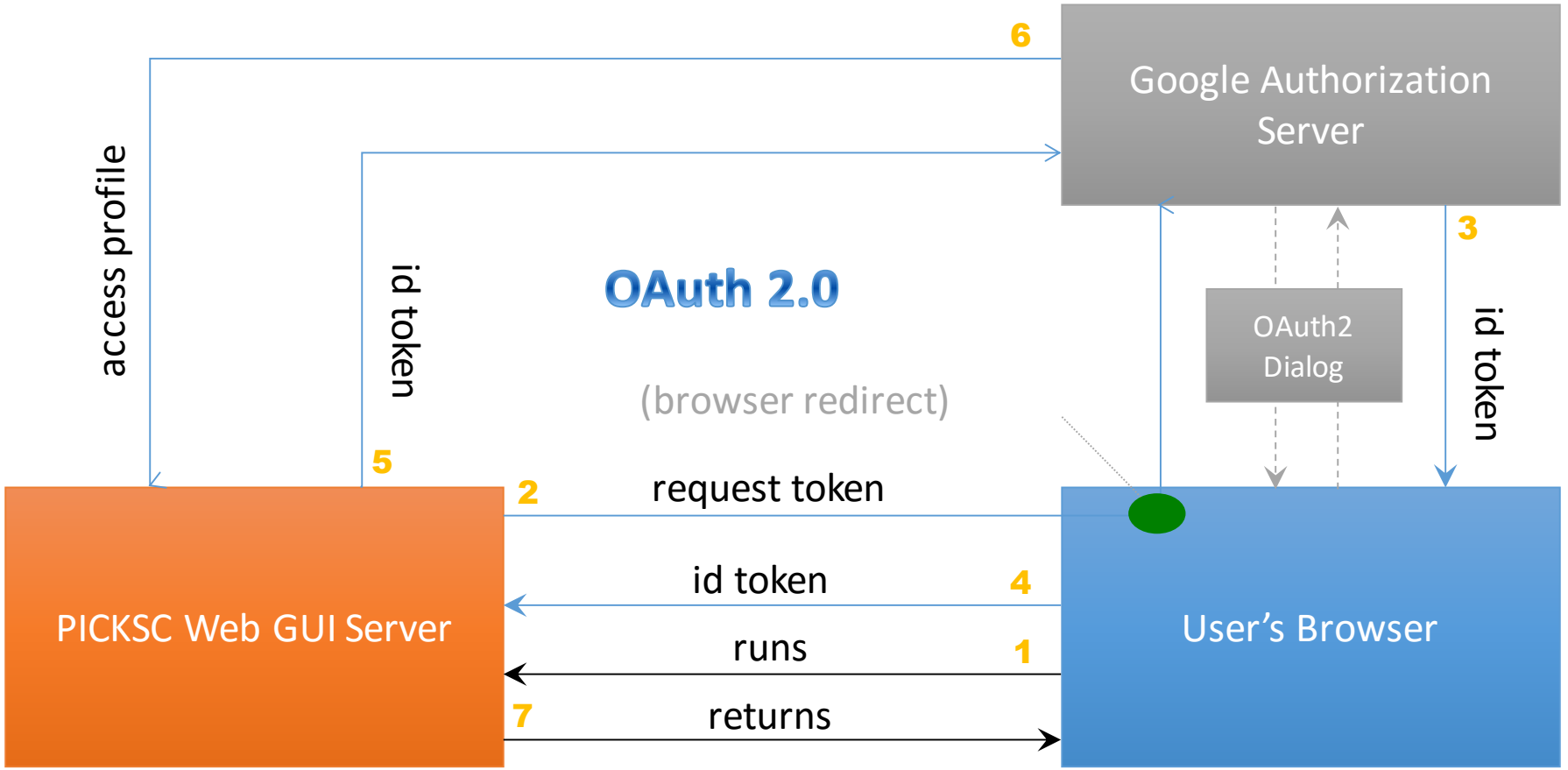
- LAMP Stacks: plain PHP, no framework
- ‘User’ and ‘Job’ modules have database tables in the backend
- 'Input' interface supports for hundreds or thousands of parameters
- 'Postprocess' includes the batch processing

Current Server Architecture

- Hostname: picksc.hoffman2.idre.ucla.edu
  - *only* user: picksc
- CentOS 6.7
  PHP: 5.3.3
  MySQL 5.1
- Google sign-in enabled
- InCommon Certs

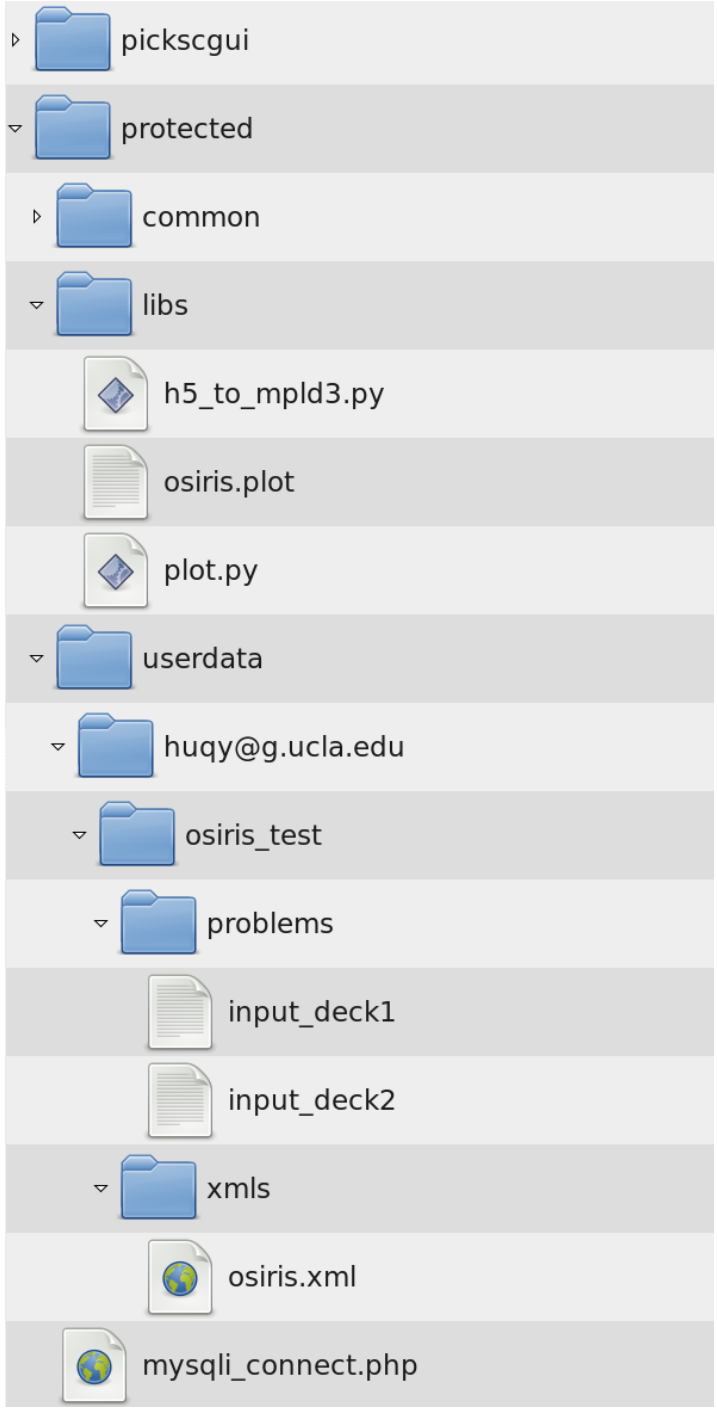


User Management: Google OAuth2



Input Module

- Inputs defined in a hierarchical description in XML file
  - Grouped in 3 level hierarchy
  - Can limit display subset of parameters that differ from defaults
  - Can handle multiple namelists/XMLs
  - Help available for each input
- Server implementation:
  - ‘protected’ folder
    - outside from web root
  - ‘common’
    - provide common template
  - ‘userdata’
    - user space
- Each Code should have:
  - Its own XML files
    - For multiple namelist sections
  - Its customized problems
    - Must specify/use XML filename



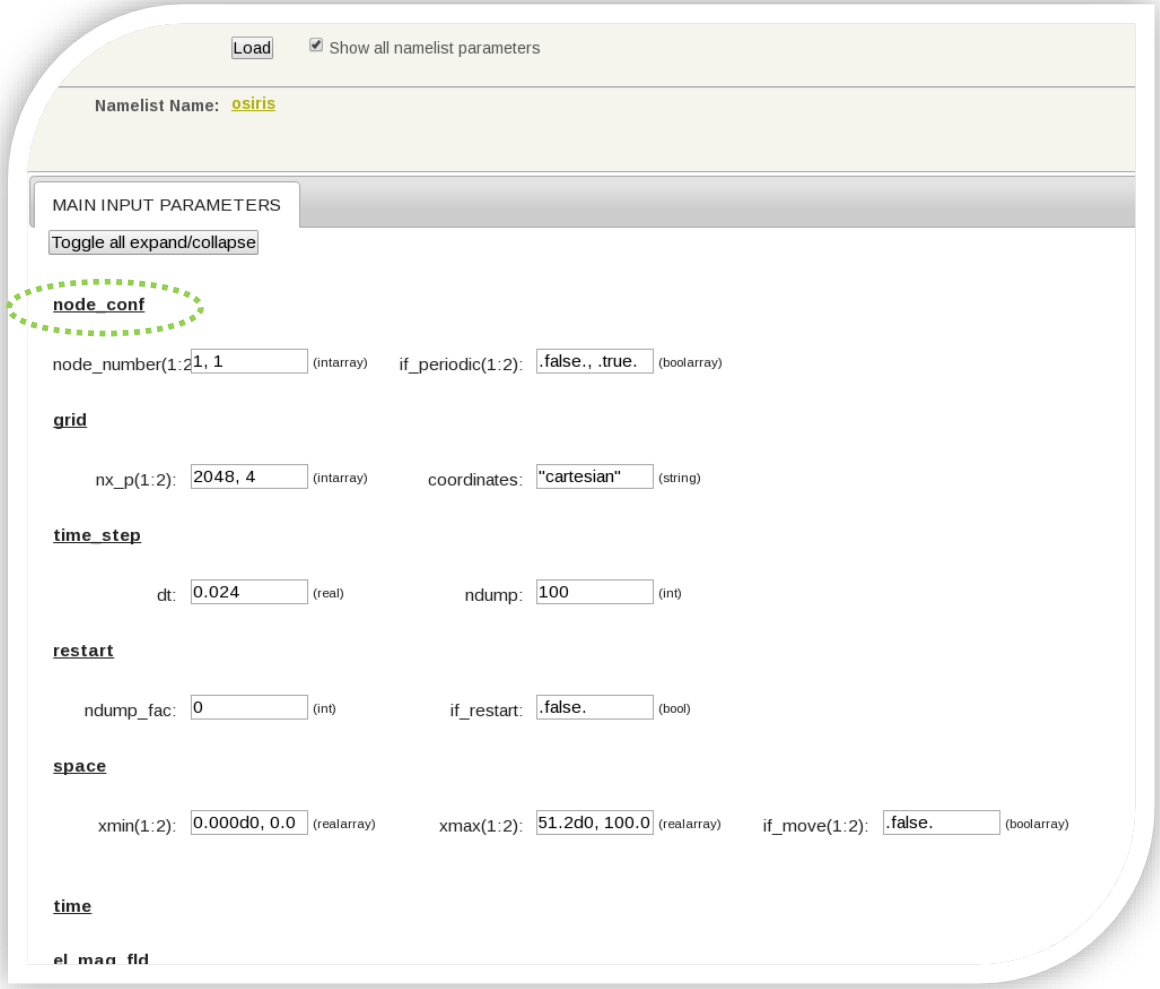
Smart Interface for Input

osiris.xml

```
−<application>
  <name>osirisTest</name>
  <desc>This XML is for OSIRIS</desc>
  <command>None</command>
 −<input>
  −<file type="text">
    <name>TEST</name>
    <filename>osiris</filename>
    <desc>The file contains namelist for OSIRIS</desc>
   </file>
  −<group>
    <name>MAIN INPUT PARAMETERS</name>
   −<desc>
      This group describes the spatial limits of the simulations
    </desc>
   −<group>
     <name>node_conf</name>
     <desc/>
    −<param type="intarray">
      <name>node_number</name>
      <size>2</size>
      <value>240,48</value>
     −<desc>
        specifies the number of nodes to use in each directio
        will be specified by setting all the items to 1. It is not
        direction, but this will guaranty better load balancing
      </desc>
     </param>
    −<param type="boolarray">
      <name>if_periodic</name>
      <size>2</size>
      <value>.false., .true.</value>
     −<desc>
        specifies if the boundary conditions for each directio
        or particle species.
      </desc>
     </param>
    </group>
   −<group>
     <name>grid</name>
     <desc>spatial grid </desc>
    −<param type="intarray">
```

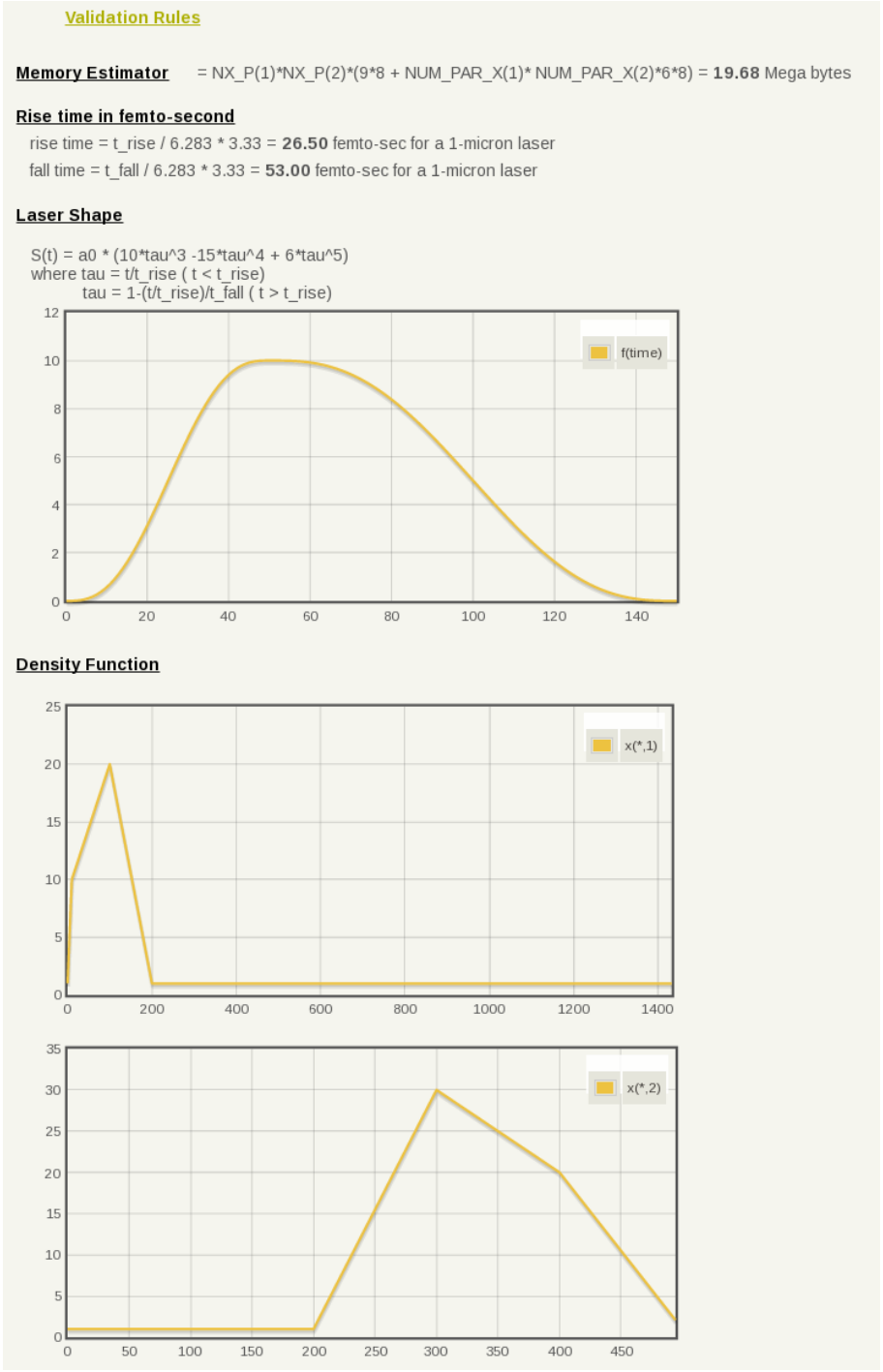
namelist template

```
node_conf
{
 node_number(1:2) = 1, 1,
 if_periodic(1:2) = .true., .true.,
}

grid
{
 nx_p(1:2) = 5000, 4,
 coordinates = "cartesian",
}

time_step
{
 dt = 0.19,
 ndump = 5,
}

restart
{
 ndump_fac = 0,
}

space
{
 xmin(1:2) = 0.000d0, 0.0,
 xmax(1:2) = 1000.0, 10.0,
 if_move(1:2) = .false.,
}

time
{
 tmin = 0.0d0,
```

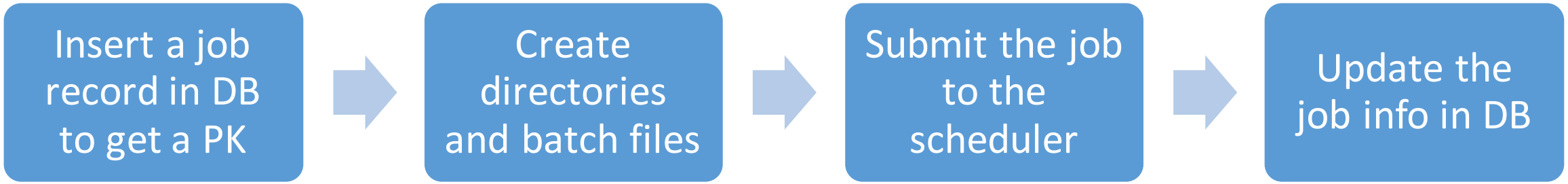


Input Validation

- All rules hard-coded in system
  - PHP + Javascript
- Chart plot:
  - Flot: Jquery-based
- In XML files:
  - <input>
    ...
    </input>
    <validation>
      <rule>3</rule>
      <rule>4</rule>
    </validation>



Manage jobs from web server



- Job has a separate ID (primary key) in DB
- All job info saved in `job' table in DB
- Input saved in namelist folder
- Output saved in a separate job folder

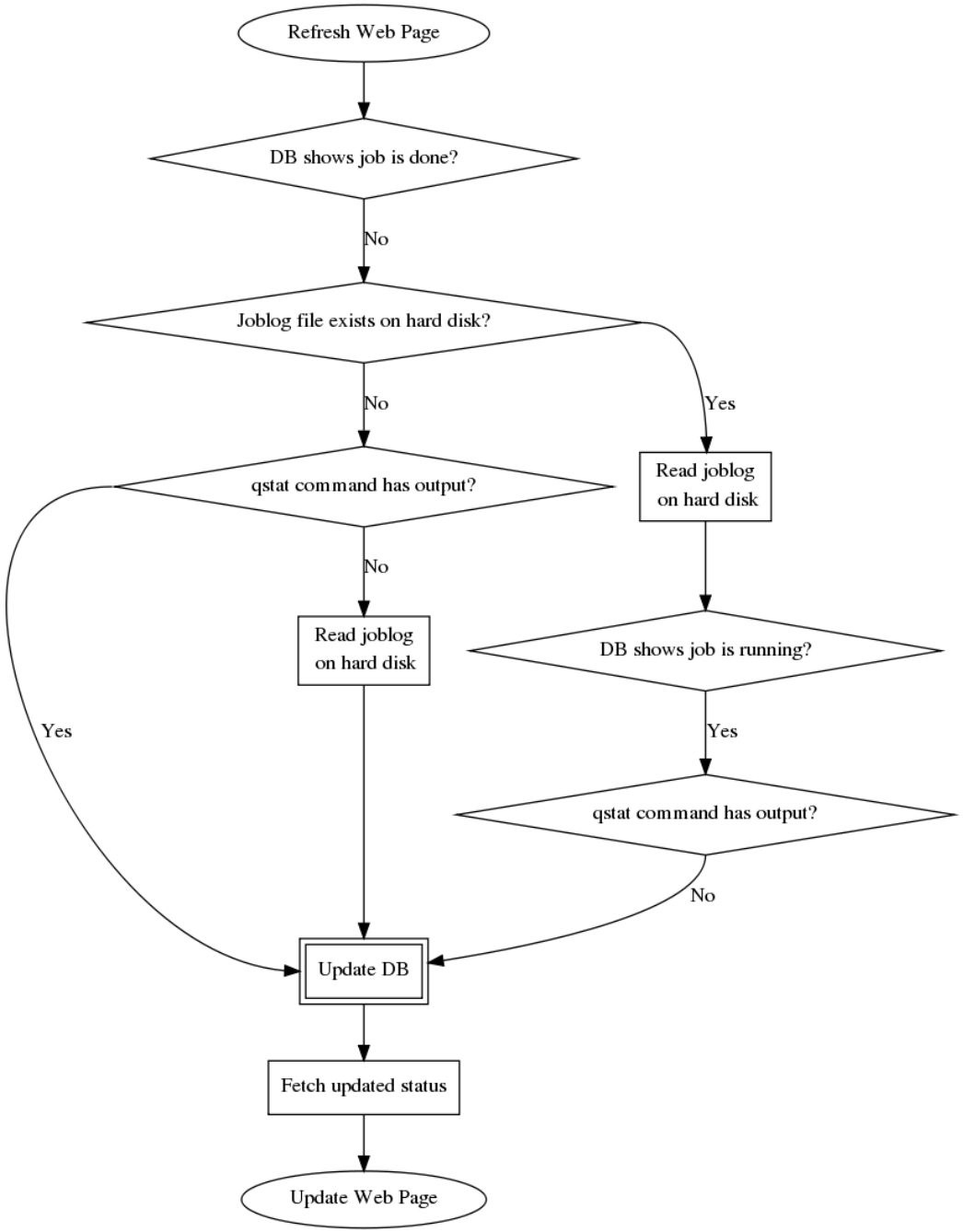
UGE Environment Setup

- 'picksc' (the apache user) is the solo user to UGE
- PHP runs bash *q*-related command
- Job submission:
  - Export bash environment variables
  - Batch cmd file:
    - Adopt IDRE script template
    - joblog & output files: in job space
  - Copy input file to job space
  - Job id: parsed from the stdout
    - Write to local DB

Job batch running to H2

- Manage files
  - Common libs in protected folder
    - Python post-process scripts
    - plot templates
  - Job-specific files: copied to job space
    - Customized parameters

- Improve the syncing process of the job status between H2 and web server.



Jobs folder

- Separate subfolder under ‘protected’ folder
- Named by the job id in DB
  - PK in jobs table
- Batch file and output files saved per job
- Deleting job will delete info in:
  1. DB
  2. Scheduler
  3. Corresponding folder

Interfaces for jobs

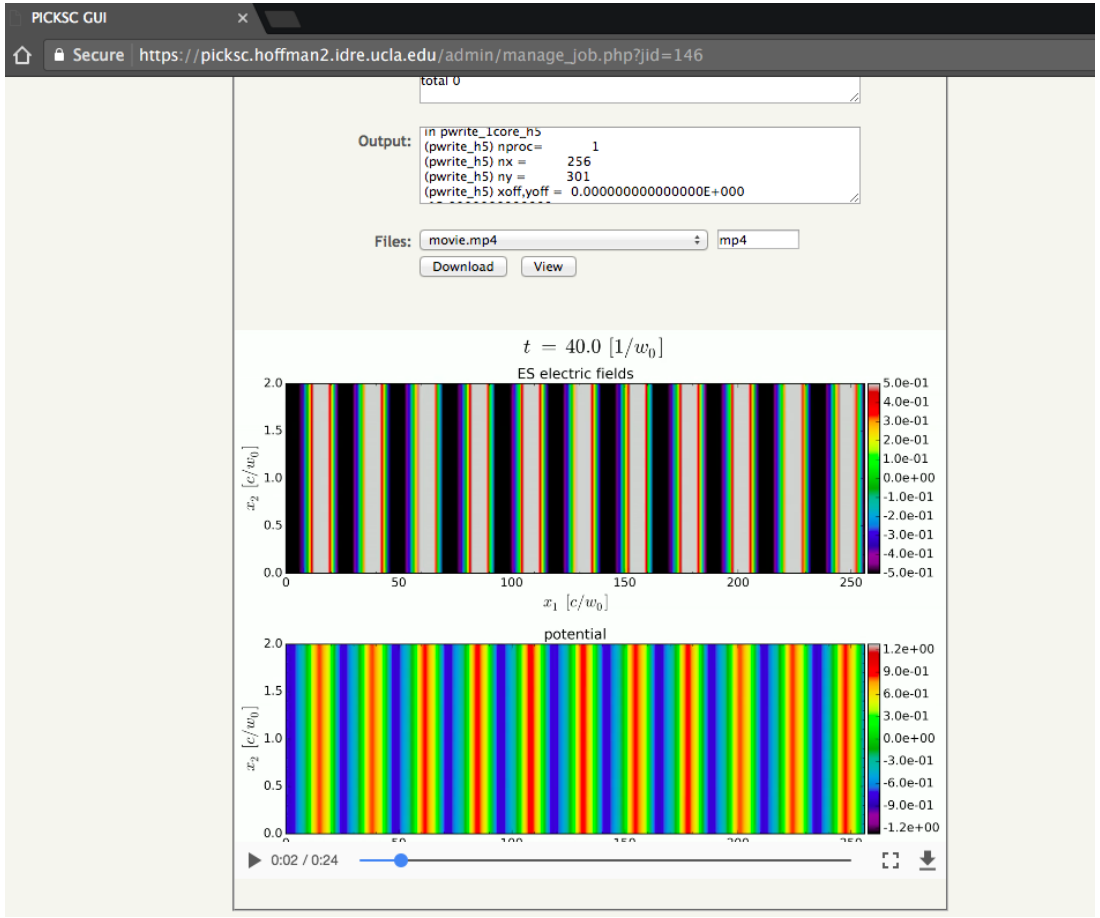
- General Users:
  - Run the executables from ‘common’ folder
  - Currently it can display the std out content
  - Manage their own jobs

- Admin Users:
  - Upload executables to ‘common’ folder
  - Manage all job input/outputs

Post-processing

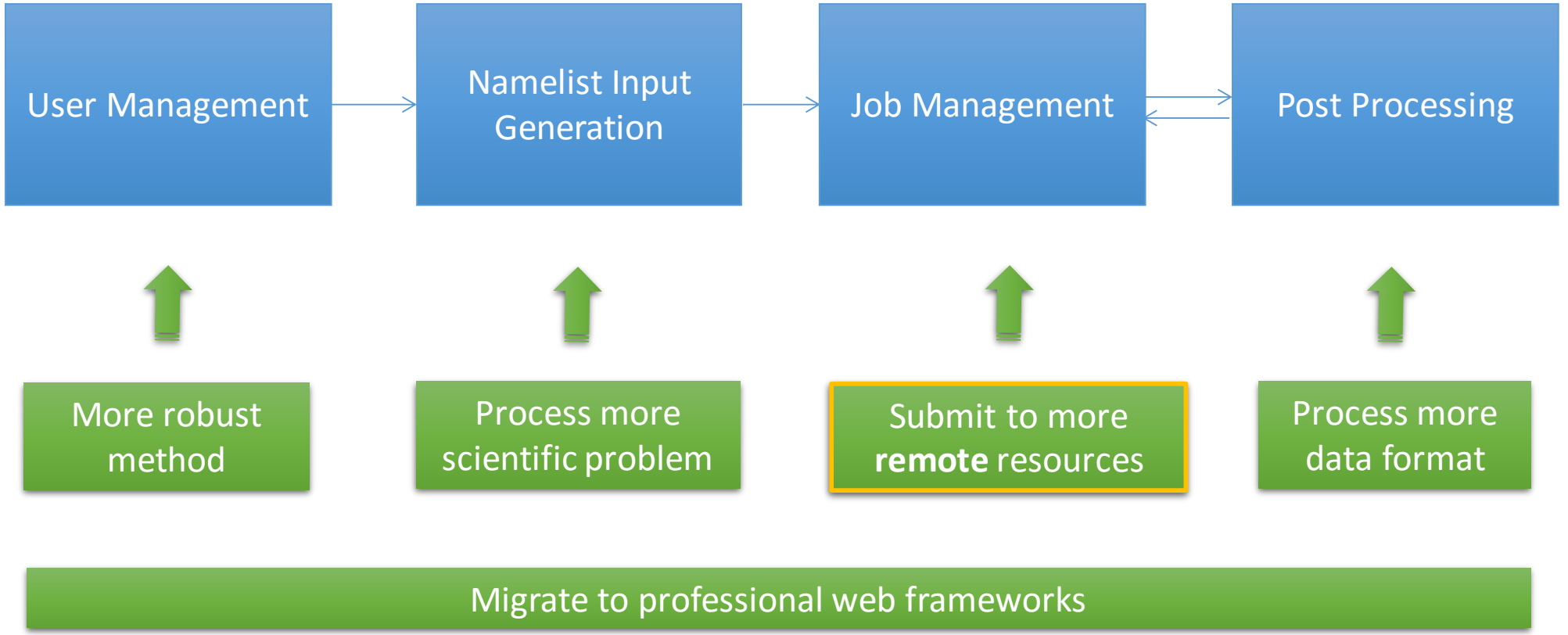
- More options for post-processing
  - IDL, matplotlib
  - Customized parameters
- Python libs installed in /protected/libs/ folder
  - visxd from Frank for IDL
  - plot.py from Frank for matplotlib
    - osiris.plot
  - mpld3
- Online view of hdf5 pictures

A quick screenshot for reviewing the animations

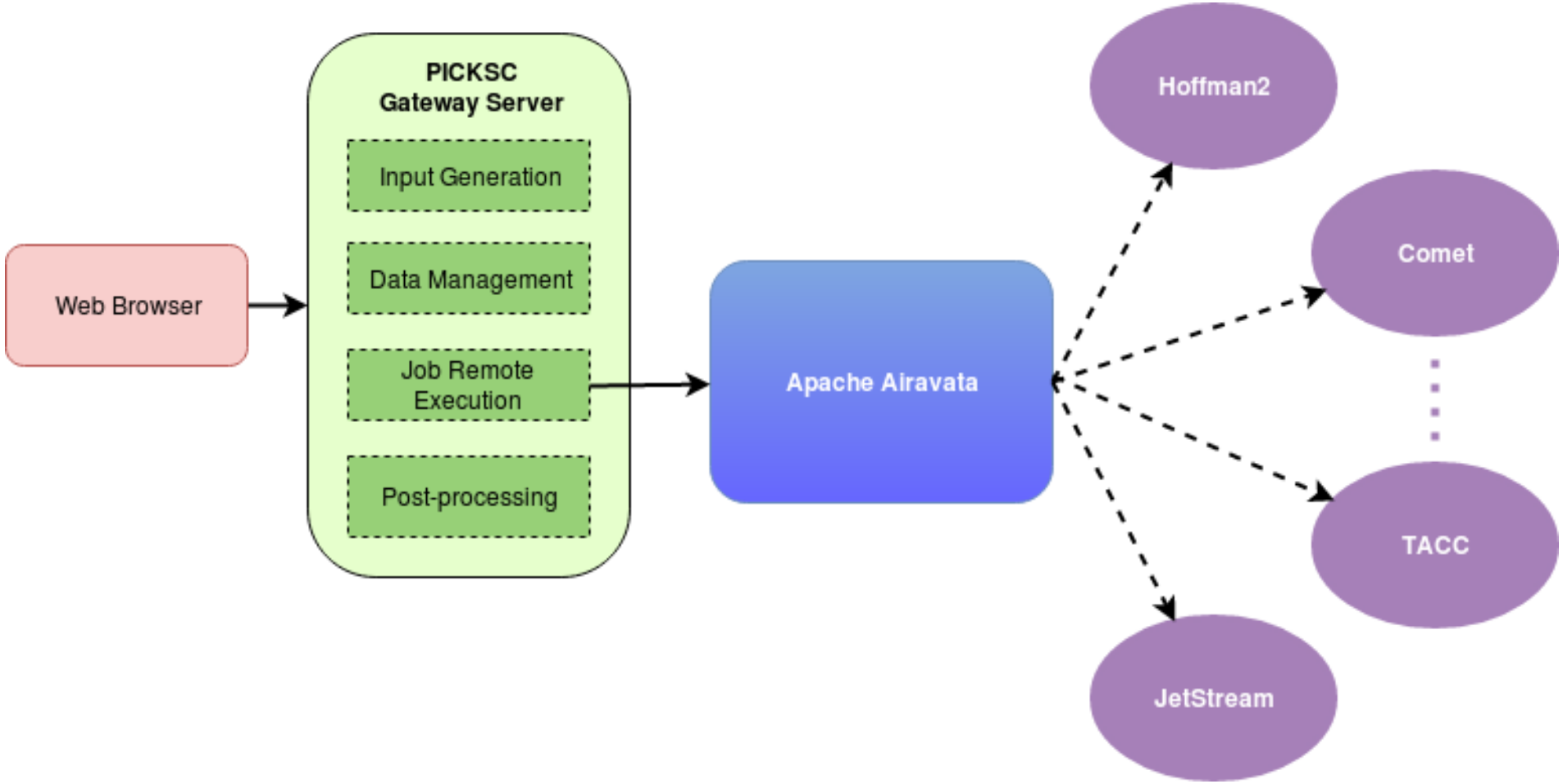


Live Demo

Future Developments



New Server Architecture



Apache Airavata as a black-box middleware