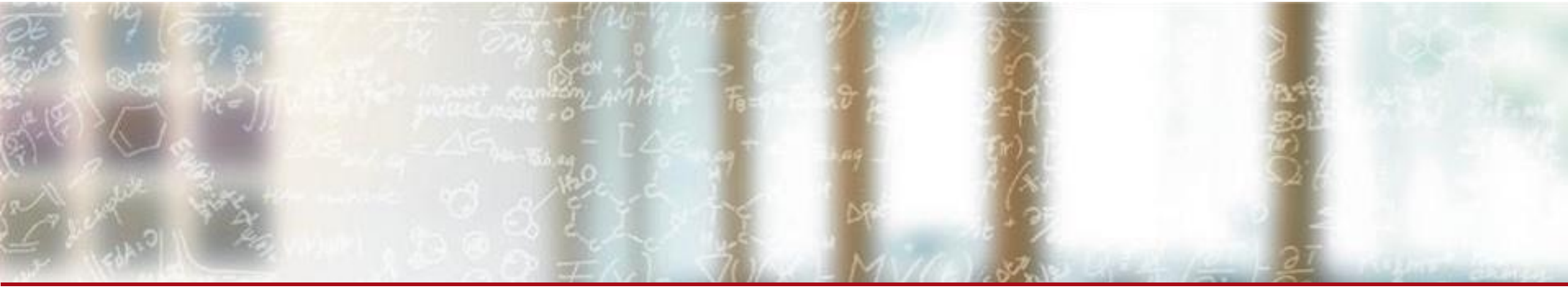
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



FirecREST: an API for HPC
Introduction

Juan Pablo Dorsch, CSCS (ETH-Zürich)

FirecREST Training
October 3rd, 2023

Agenda

FirecREST training – Agenda for 03.10.2023

Time	Topic
10:30 – 10:45	Welcome coffee and registration
10:45 – 11:30	FirecREST introduction
11:30 – 12:30	pyFirecREST and FirecREST CLI, hands on
12:30 – 13:30	Lunch break (at Ristorante Stadio)
13:30 – 14:00	Showcase: scientific web portal using pyFirecREST
14:00 – 15:30	Hackathon: hands on
15:30 – 15:45	Coffee break
15:45 – 17:00	Hackathon: hands on

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

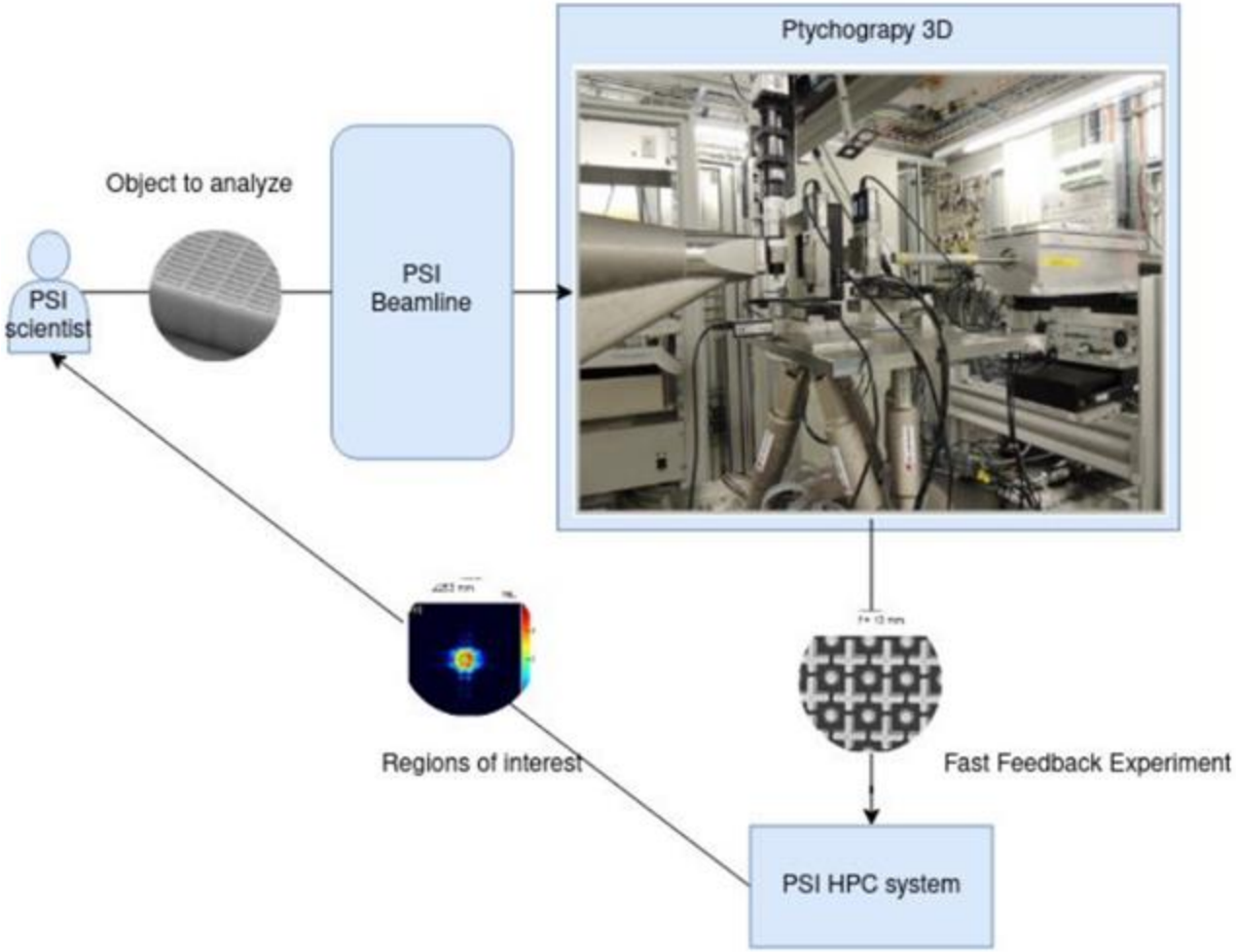
Motivation

Advanced Scientific Workflow on HPC

- SELVEDAS
  - The **Service for Large Scale Volume Experiment Data Analysis** is a collaboration project from SwissUniversities between Paul Scherrer Institut (PSI) and CSCS.
  - PSI operates the Swiss Light Source (SLS) which is a third-generation synchrotron light source (infrared spectrum range)
  - SLS allows the study of surfaces of new structures and the bulk properties of novel materials:
    - protein crystals (pharmaceutical research)
    - magnetic properties of surfaces (storage technologies)
    - low friction surfaces
    - solar cells, etc

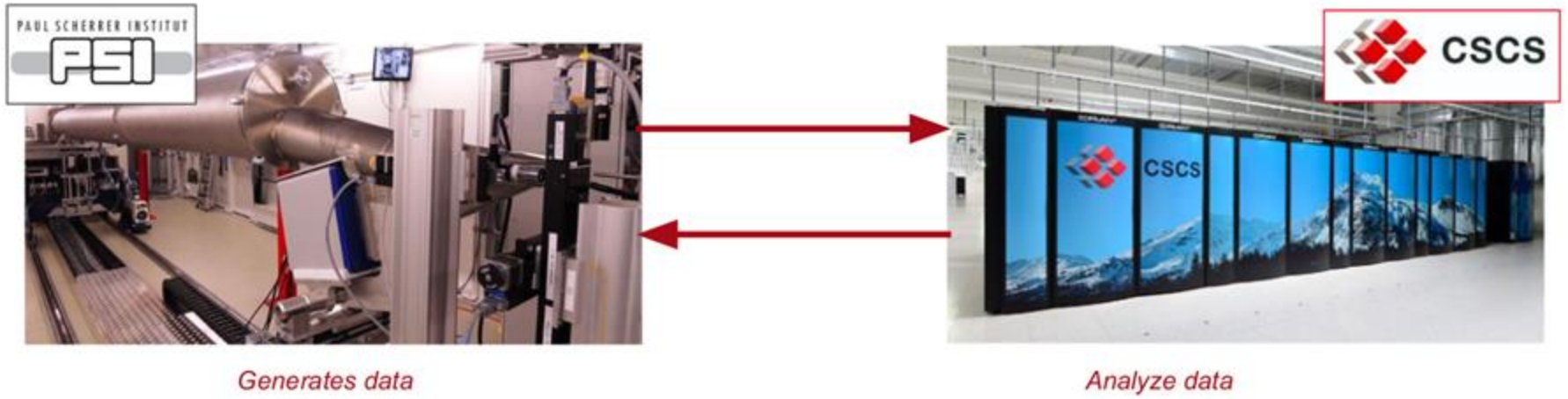
Advanced Scientific Workflow on HPC

- Scientists bring objects to the beamline and do research on their internal nanostructure using a 3D Ptycography scan
- As soon as detector images can be taken scientists need fast ptychographic reconstructions to find the interesting region of the object
- The problem is reconstruction has to be fast: PSI needs a Super Computer



Advanced Scientific Workflow on HPC

- SELVEDAS project provides the workflow between PSI beamline and CSCS computational resources to fast feedback experiment.
- PSI generates the data and via a web portal sends it to CSCS HPC systems to analyze.
- After analysis is done, results must be sent back to PSI scientists in the web portal.



Advance Scientific Workflows on HPC

- Projects/Users need to integrate HPC into advanced workflows
  - Custom solutions using “classic” HPC
    - Challenging for users (lack of development standards)
    - Challenging for HPC provider (multiple workflows → multiple support)
- Need for a standard modern interface to HPC resources
  - HPC clusters
  - Job schedulers
  - Filesystems operations
  - Moving data within filesystems
  - External data transfer from and to HPC filesystems
- Interface with existing HPC services/infrastructure
  - Avoid building parallel services

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

FirecREST

FirecREST in a nutshell

- **FirecREST is a web-enabled API to HPC resources**

FirecREST in a nutshell

- **FirecREST is a web-enabled API to HPC resources**

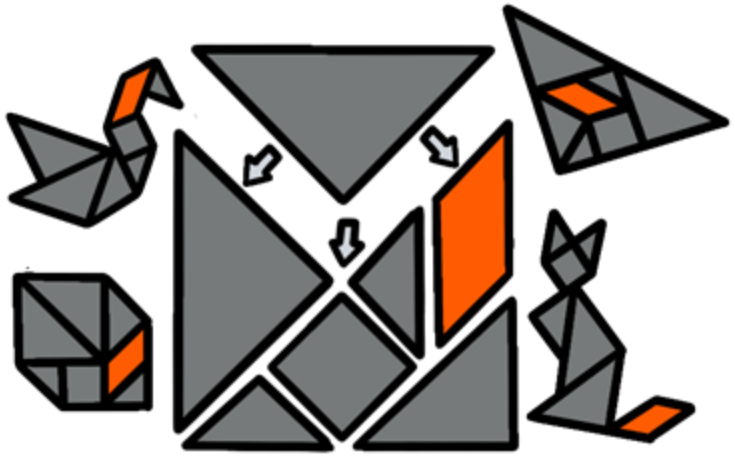
- Presents standard programming interface
  - Based on RESTAPI concept
  - Independent of programming language (HTTP)
  - Translates web requests into HPC business logic
  - Parses back HPC results into web-friendly format

FirecREST in a nutshell

- **FirecREST is a web-enabled API to HPC resources**
- Presents standard programming interface
- Provides web interface for classic HPC
  - Creation of web applications over HPC
  - Enables support for multiple devices

FirecREST in a nutshell

- **FirecREST is a web-enabled API to HPC resources**
- Presents standard programming interface
- Provides web interface for classic HPC
- Allows modular design to support different workflows and HPC systems
  - Abstracts HPC resources into components and objects



FirecREST features

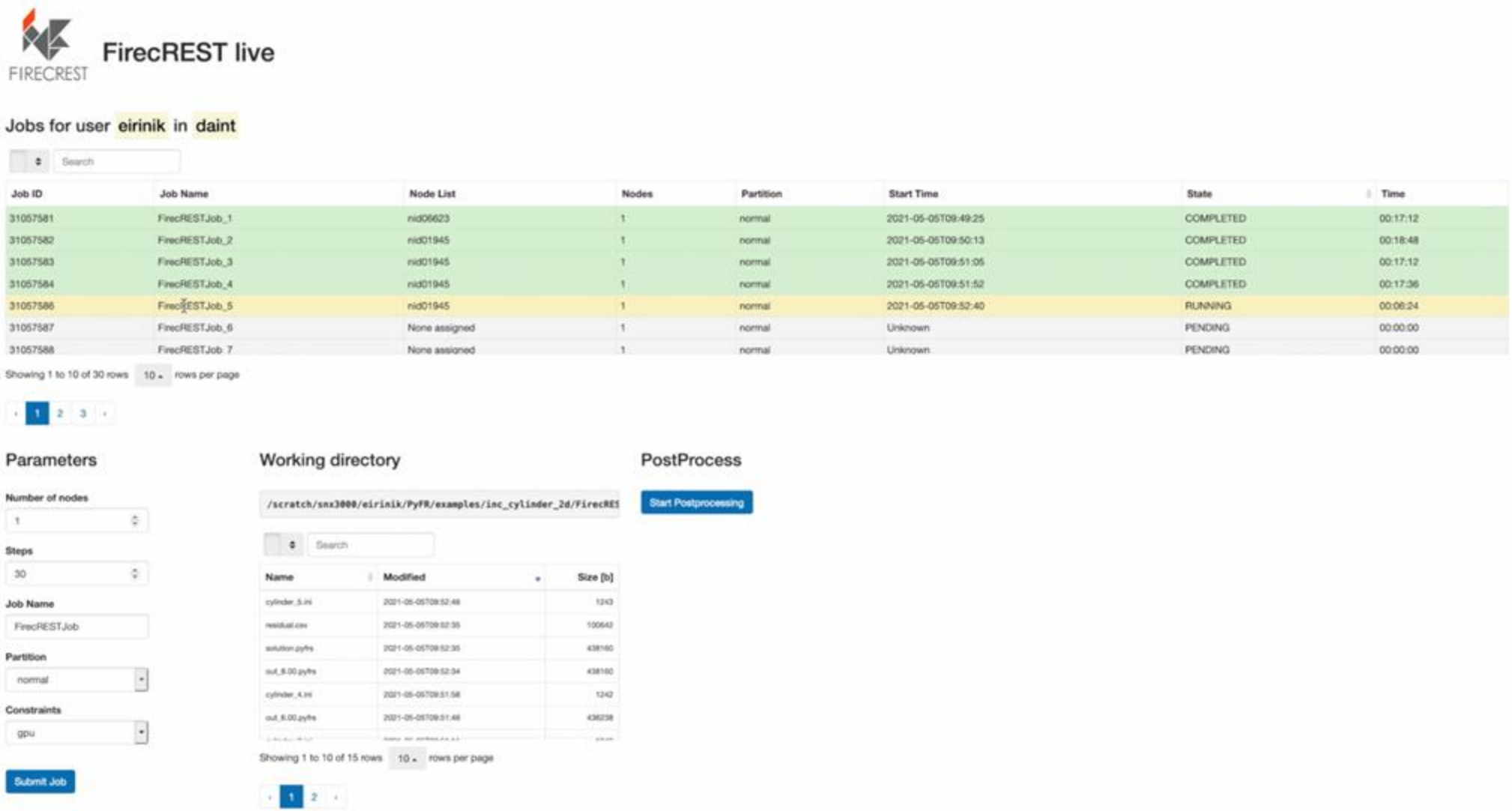
- **FirecREST is a web-enabled API to HPC resources**
- Presents standard programming interface
- Provides web interface for classic HPC
- Allows modular design to support different workflows and HPC systems
- Integrates with authentication and authorization layers
  - Relies on standard IAM solutions for authentication

FirecREST API

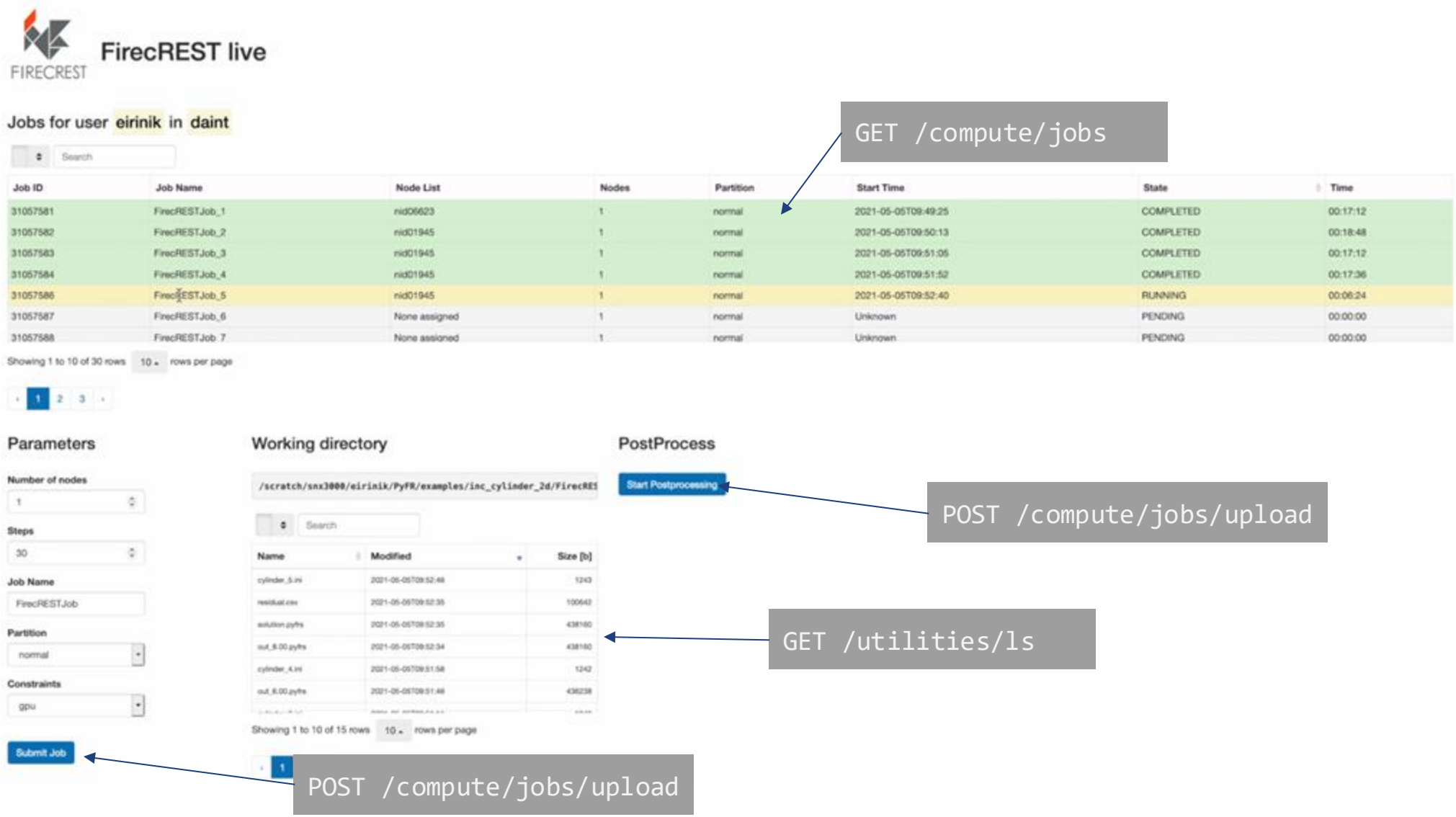
The FirecREST API

- OpenAPI documentation: https://firecrest-api.cscs.ch

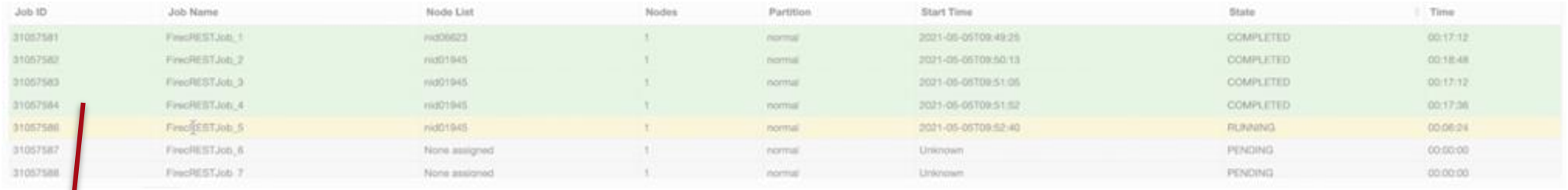
HPC web portal



HPC web portal (where FirecREST calls are applied)



HPC web portal (in the background)



Job ID	Job Name	Node List	Nodes	Partition	Start Time	State	Time
31057581	FirecRESTJob_1	nid00623	1	normal	2021-05-05T09:49:25	COMPLETED	00:17:12
31057582	FirecRESTJob_2	nid01945	1	normal	2021-05-05T09:50:13	COMPLETED	00:18:48
31057583	FirecRESTJob_3	nid01945	1	normal	2021-05-05T09:51:05	COMPLETED	00:17:12
31057584	FirecRESTJob_4	nid01945	1	normal	2021-05-05T09:51:52	COMPLETED	00:17:36
31057585	FirecRESTJob_5	nid01945	1	normal	2021-05-05T09:52:40	RUNNING	00:06:24
31057587	FirecRESTJob_6	None assigned	1	normal	Unknown	PENDING	00:00:00
31057588	FirecRESTJob_7	None assigned	1	normal	Unknown	PENDING	00:00:00

```
<table
id="table"
class="table table-condensed"
data-toggle="table"
data-height="400"
data-pagination="true"
data-url="/list_jobs"
data-auto-refresh="true"
data-auto-refresh-interval="15"
data-auto-refresh-status="true"
data-auto-refresh-silent="false"
data-search="true"
data-row-style="rowStyle"
data-sort-name="state"
>
<thead>
    <tr>
    <th data-sortable="false" data-field="jobid">Job ID</th>
    <th data-sortable="false" data-field="name">Job Name</th>
    <th data-field="nodelist">Node List</th>
    <th data-field="nodes">Nodes</th>
    <th data-field="partition">Partition</th>
    <th data-sortable="false" data-field="start_time">Start Time</th>
    <th data-sorter="stateSorter" data-sortable="true" data-field="state">State</th>
    <th data-sortable="false" data-field="time">Time</th>
    </tr>
</thead>
</table>
```

HPC web portal (in the background)

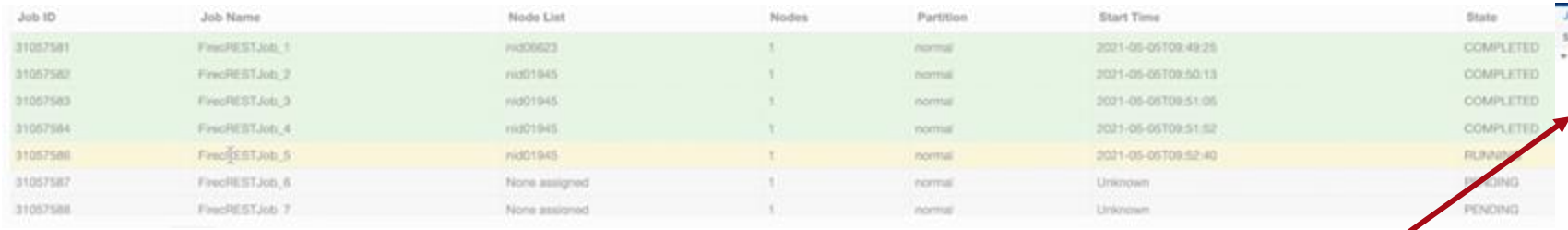
Job ID	Job Name	Node List	Nodes	Partition	Start Time	State	Time
31057581	FirecRESTJob_1	nid06623	1	normal	2021-05-05T09:49:25	COMPLETED	00:17:12
31057582	FirecRESTJob_2	nid01945	1	normal	2021-05-05T09:50:13	COMPLETED	00:18:48
31057583	FirecRESTJob_3	nid01945	1	normal	2021-05-05T09:51:05	COMPLETED	00:17:12
31057584	FirecRESTJob_4	nid01945	1	normal	2021-05-05T09:51:52	COMPLETED	00:17:36
31057585	FirecRESTJob_5	nid01945	1	normal	2021-05-05T09:52:40	RUNNING	00:06:24
31057587	FirecRESTJob_6	None assigned	1	normal	Unknown	PENDING	00:00:00
31057588	FirecRESTJob_7	None assigned	1	normal	Unknown	PENDING	00:00:00

```
<table
id="table"
class="table table-condensed"
data-toggle="table"
data-height="400"
data-pagination="true"
data-url="/list_jobs"
data-auto-refresh="true"
data-auto-refresh-interval="15"
data-auto-refresh-status="true"
data-auto-refresh-silent="false"
```

```
@app.route("/list_jobs", methods=["GET"])
@oidc.require_login
def list_jobs():

    list_jobs_request = requests.get("https://firecrest.cscs.ch/compute/jobs", headers={"X-Machine-Name": "daint"})

    if list_jobs_request.ok: # if HTTP request was OK
        jobs_running = list_jobs_request.json()
        return {"rows": jobs_running}

    # return empty list if HTTP request wasn't successful
    return {"rows": []}
```

```
    <th data-sortable="false" data-field="time">Time</th>
    </tr>
</thead>
</table>
```

HPC web portal



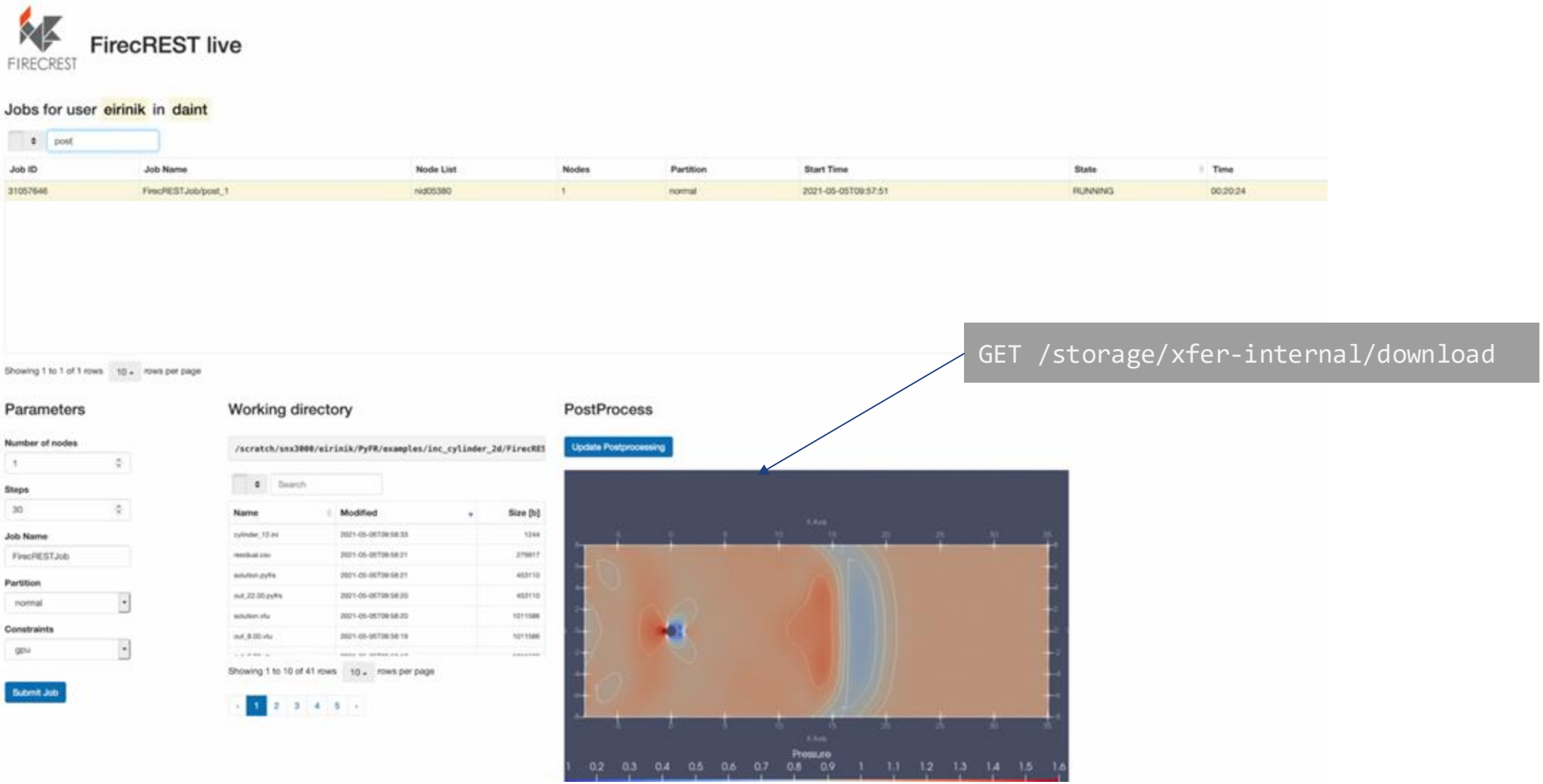
Job ID	Job Name	Node List	Nodes	Partition	Start Time	State
31057581	FirecRESTJob_1	nid06623	1	normal	2021-05-05T09:49:25	COMPLETED
31057582	FirecRESTJob_2	nid01945	1	normal	2021-05-05T09:50:13	COMPLETED
31057583	FirecRESTJob_3	nid01945	1	normal	2021-05-05T09:51:05	COMPLETED
31057584	FirecRESTJob_4	nid01945	1	normal	2021-05-05T09:51:52	COMPLETED
31057586	FirecRESTJob_5	nid01945	1	normal	2021-05-05T09:52:40	RUNNING
31057587	FirecRESTJob_6	None assigned	1	normal	Unknown	PENDING
31057588	FirecRESTJob_7	None assigned	1	normal	Unknown	PENDING


```
<table
id="table"
class="table table-condensed"
data-toggle="table"
data-height="400"
data-pagination="true"
data-url="/list_jobs"
data-auto-refresh="true"
data-auto-refresh-interval="15"
data-auto-refresh-status="true"
data-auto-refresh-silent="false"
```

```python
@app.route("/list_jobs", methods=["GET"])
@oidc.require_login
def list_jobs():


    list_jobs_request = requests.get("https://firecrest.cscs.ch/compute/jobs", headers={"X-Machine-Name": "daint"})

    if list_jobs_request.ok: # if HTTP request was OK
        jobs_running = list_jobs_request.json()
        return {"rows": jobs_running}

    # return empty list if HTTP request wasn't successful
    return {"rows": []}
```

```
    <th data-sortable="false" data-field="time">Time</th>
    </tr>
</thead>
</table>
```

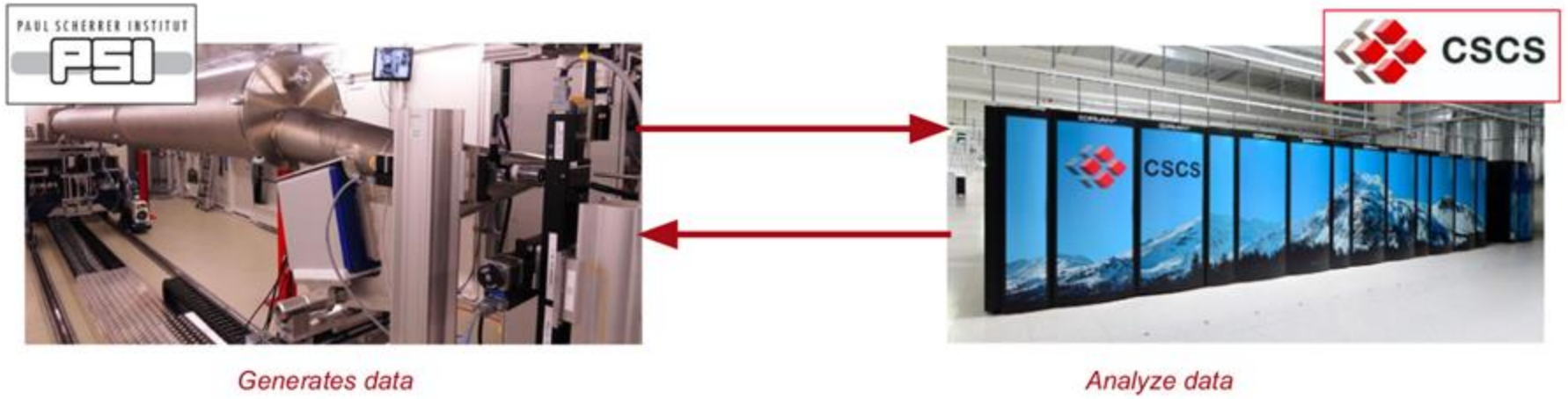
HPC web portal



FirecREST use case: Solving SELVEDAS

Advanced Scientific Workflow on HPC

- SELVEDAS project provides the workflow between PSI beamline and CSCS computational resources to fast feedback experiment.
- PSI generates the data and via a web portal sends it to CSCS HPC systems to analyze.
- After analysis is done, results must be sent back to PSI scientists in the web portal.

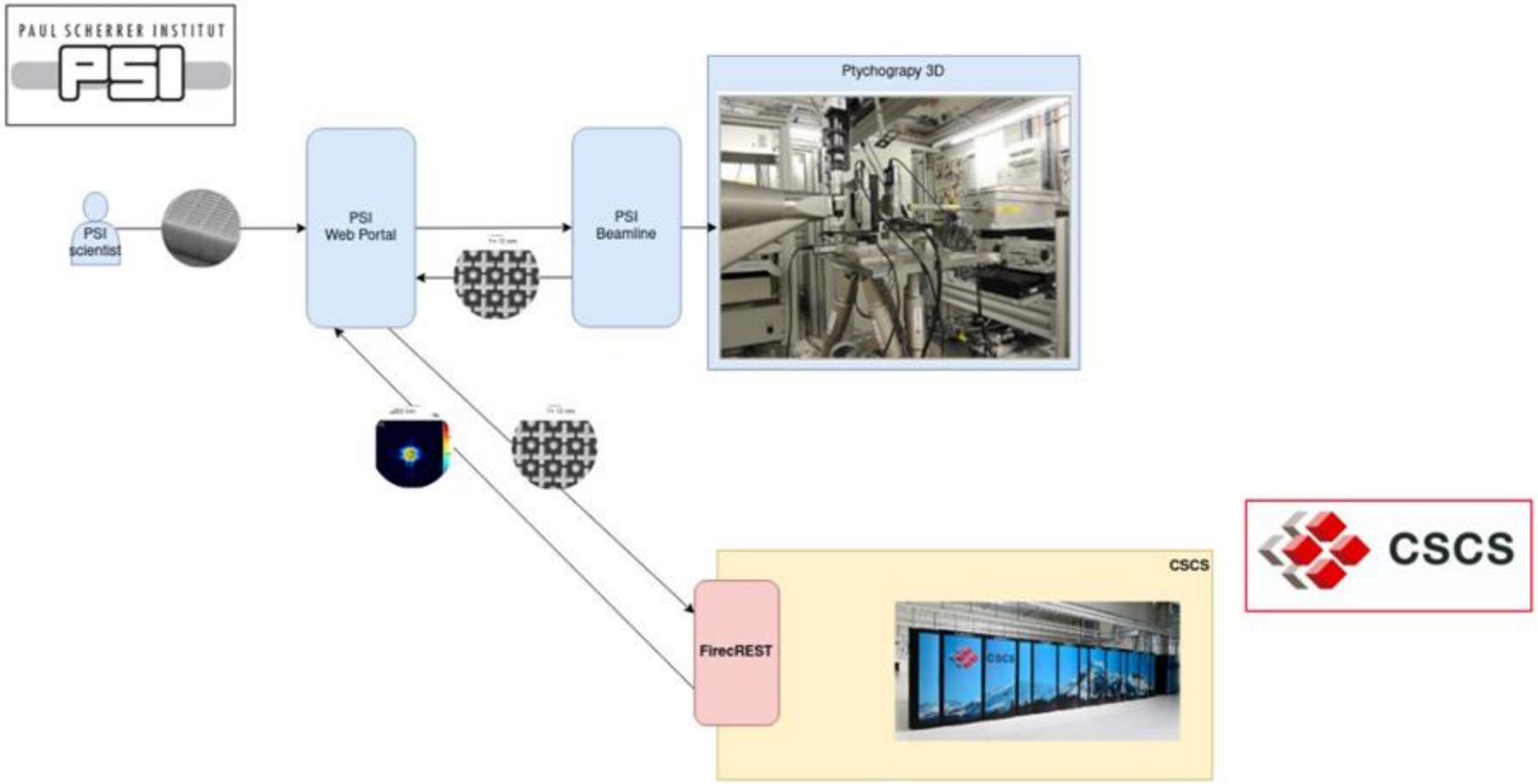


Advanced Scientific Workflow on HPC

- SELVEDAS project provides the workflow between PSI beamline and CSCS computational resources to fast feedback experiment.
- PSI generates the data and via a web portal sends it to CSCS HPC systems to analyze.
- After analysis is done, results must be sent back to PSI scientists in the web portal.

Scientific Workflow

Proposed solution (PSI PoV)



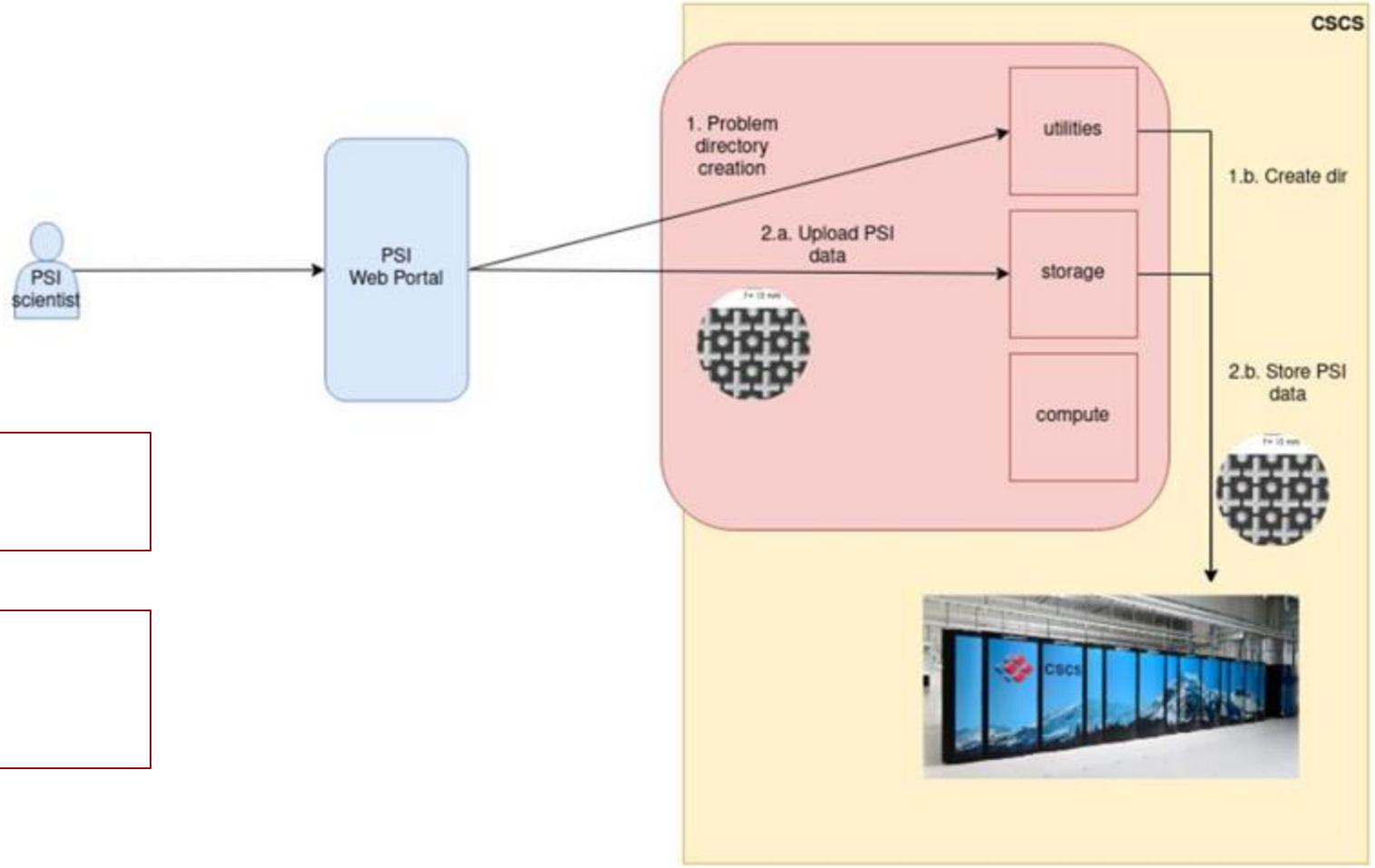
Scientific Workflow

Implementation

START

1. POST /utilities/mkdir
   -d “targetPath=/path/to/newdir”

2. POST /storage/xfer-external/upload
   -d “targetPath=/path/to/newdir”
   -d “sourcePath=/local/file”



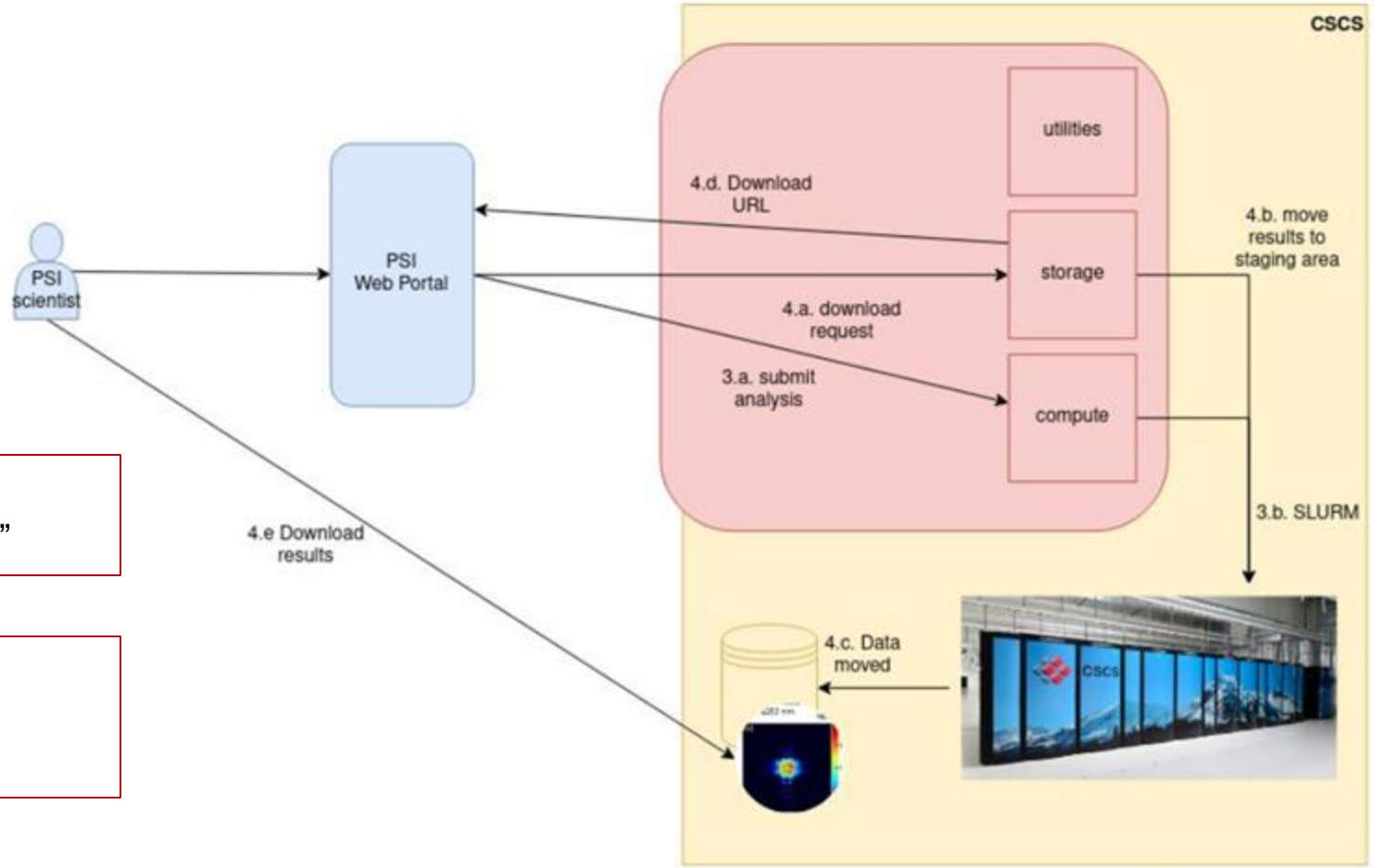
Scientific Workflow

Implementation

3. POST /compute/jobs/path
   -d “targetPath=/path/to/newdir/sbatch.sh”

4. POST /storage/xfer-external/download
   -d “sourcePath=/path/to/newdir/results”

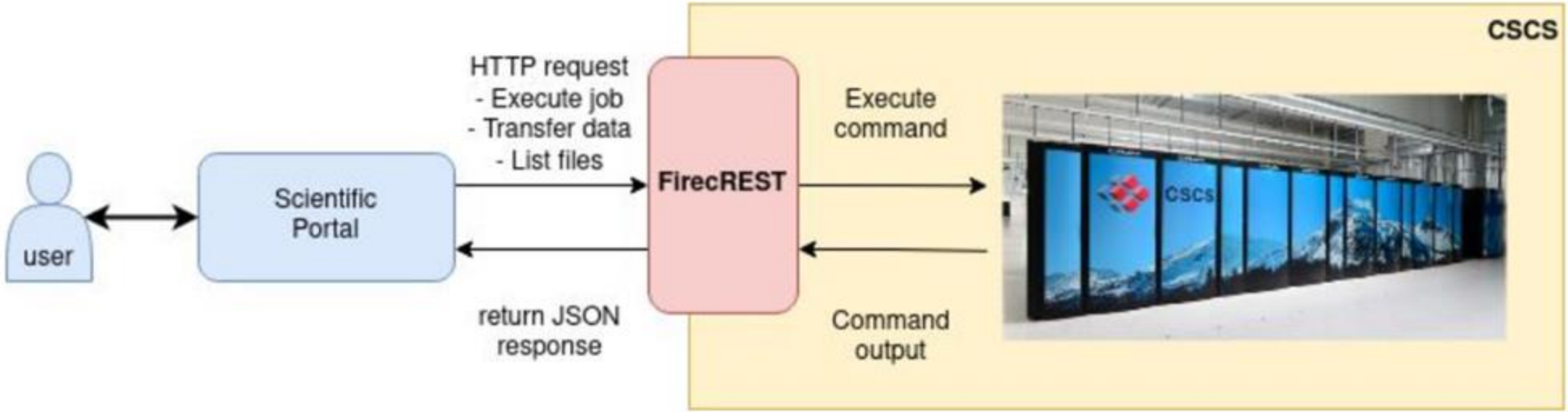
END



FirecREST internals

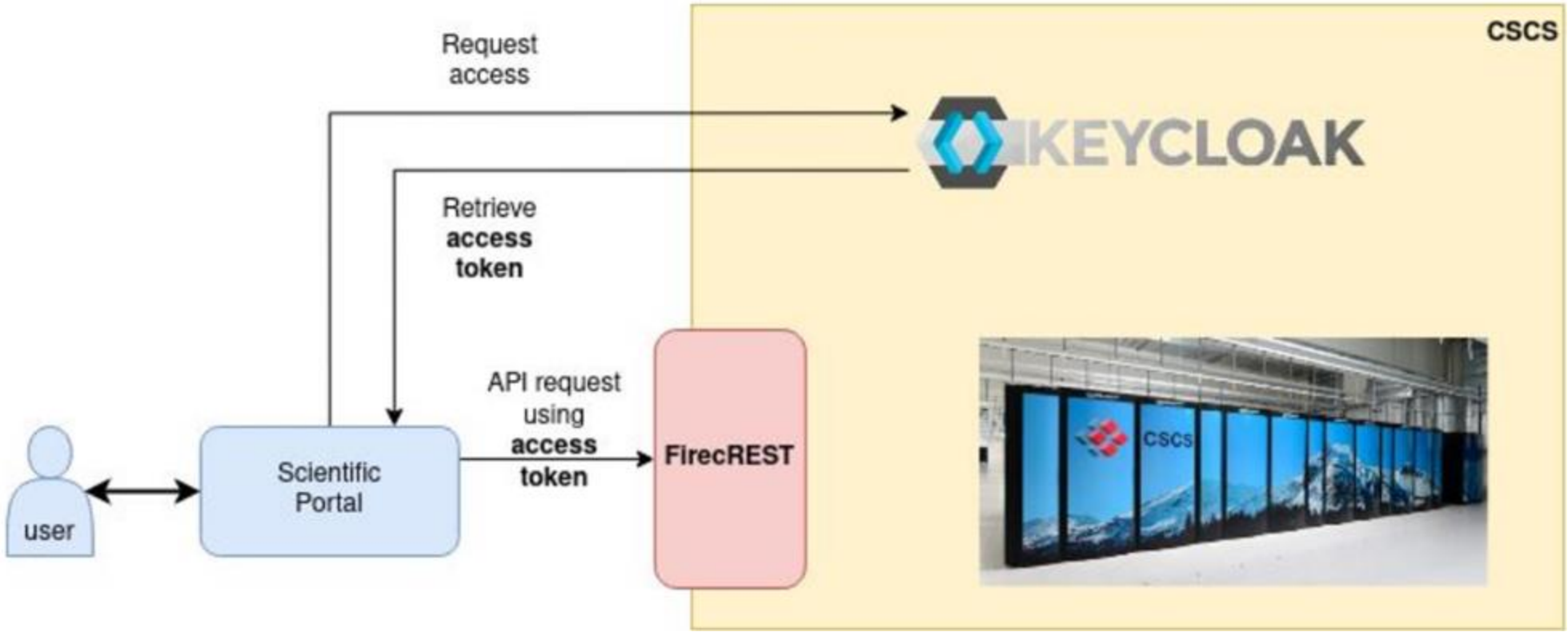
FirecREST internals

- A (very) simplified approach



FirecREST internals

- Authentication
  - IAM relies on an OpenID Connect server (OIDC)
  - FirecREST trusts in access token from trusted sources
  - JSON Web Tokens (JWT) standard is used as access tokens



FirecREST internals

- Authentication
  - Token format

```
>>> import jwt, json
>>> token =
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiw
ibmFtZSI6IlVzZXIgMDEiLCJzY29wZSI6ImZpcmVjcmVzdCIsInByZWZlcnJlZF91
c2VybmFtZSI6InVzZXIwMSIsImlhdCI6IjE1MTYyMzkwMjIiLCJqdGkiOiJlOTdhY
jIyNy1kNmMwLTQ2MGQtOGQxZS0zOWI4NmZlZDgzZGIiLCJleHAiOiIxNTcxMzQ3MT
c3In0.oP7ff8430xU9-Q9usdQY2lufPhqV9mXXOR3uX3KV2Vs"

>>> decoded = jwt.decode(token, options = {"verify_signature":
False})

>>> print(json.dumps(decoded, indent=2))
{
  "sub": "1234567890",
  "name": "User 01",
  "scope": "firecrest",
  "preferred_username": "user01",
  "iat": "1516239022",
  "jti": "e97ab227-d6c0-460d-8d1e-39b86fed83db",
  "exp": "1571347177"
}
```

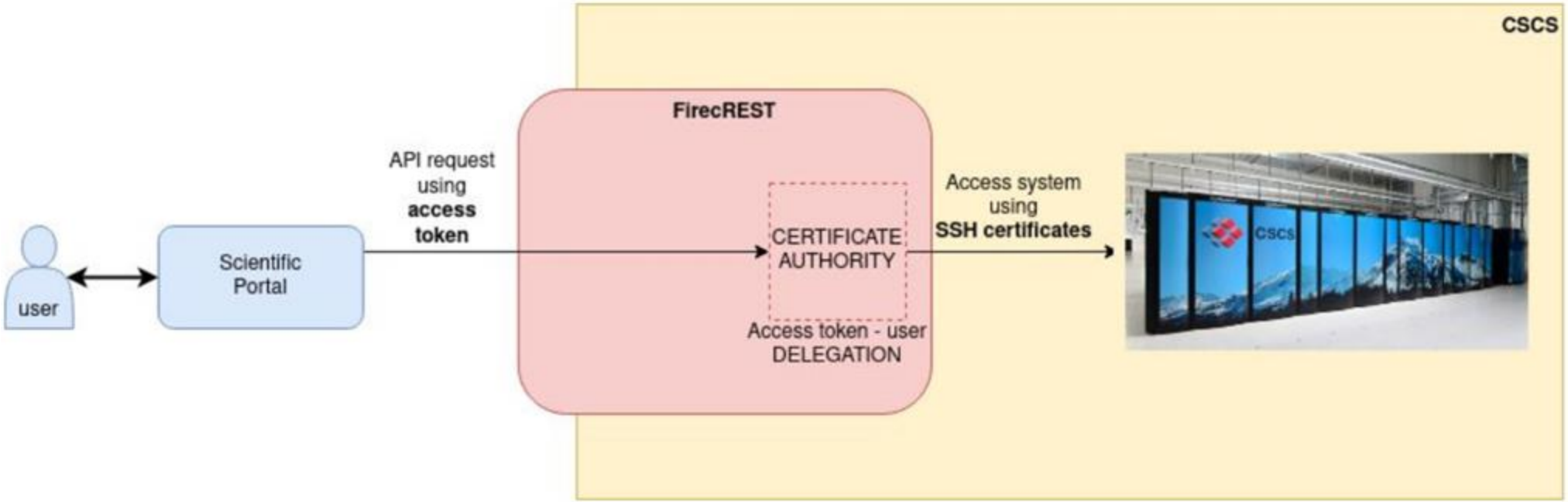
Encrypted token

System username

Expiration datetime

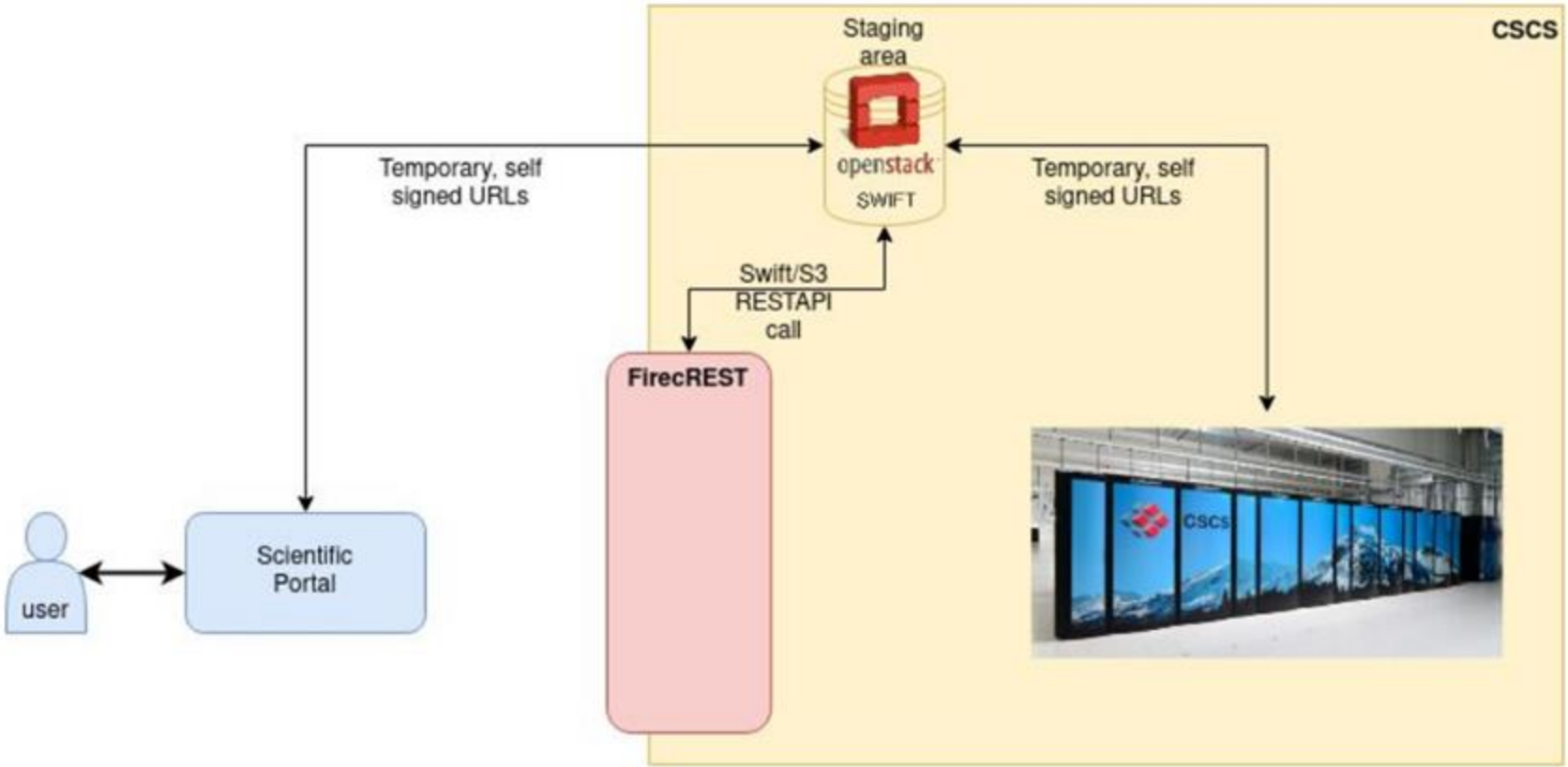
FirecREST internals

- Command execution
  - FirecREST translate JWT into **user credentials** for HPC systems
  - These credentials can only be used for a **short period** of time and for **specific commands and parameters**



FirecREST internals

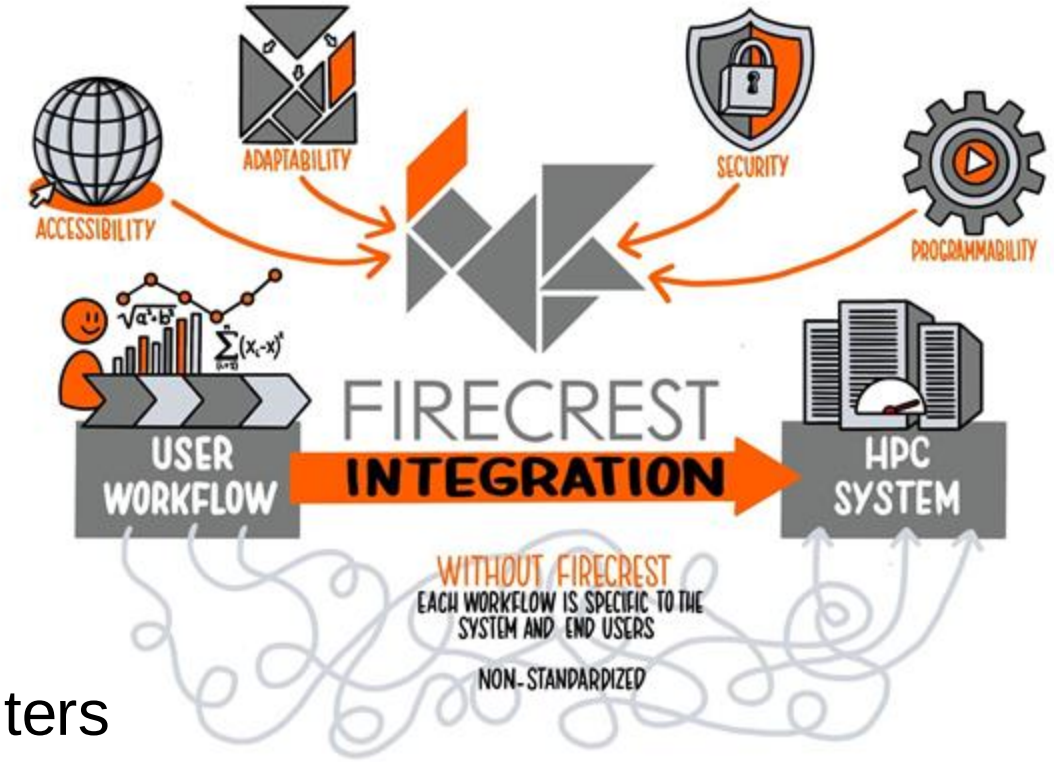
- External data transfers
  - FirecREST uses more in-site infrastructure and decouple transfers from the API
  - Object storage (OpenStack SWIFT or Amazon S3) as staging area



Conclusions

Conclusions

- FirecREST facilitates the integration of advanced scientific workflows for HPC, and allows the scientific and academic community to be closer to HPC
- Facilitates, as well, the work of HPC providers, by giving a common entry point for supporting developers
- Enforces security on remote access and command execution providing logs and filters
- Integrates existing infrastructure and standards widely used by HPC providers



Status at CSCS and Roadmap

- FirecREST is installed interfacing `daint` and `eiger` systems
- FirecREST team is working actively on supporting several vClusters provided by CSCS using a standard methodology of deployment
- FirecREST team is providing a recipe to deploy FirecREST for other HPC Centers
- The team is also working on the specification for version 2, gathering requirements from use cases and internally at CSCS

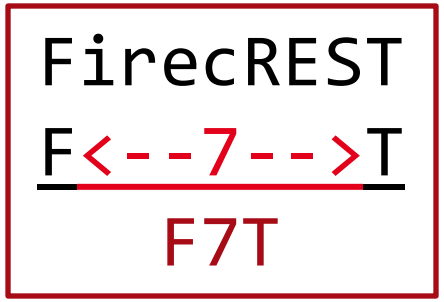
Conclusions

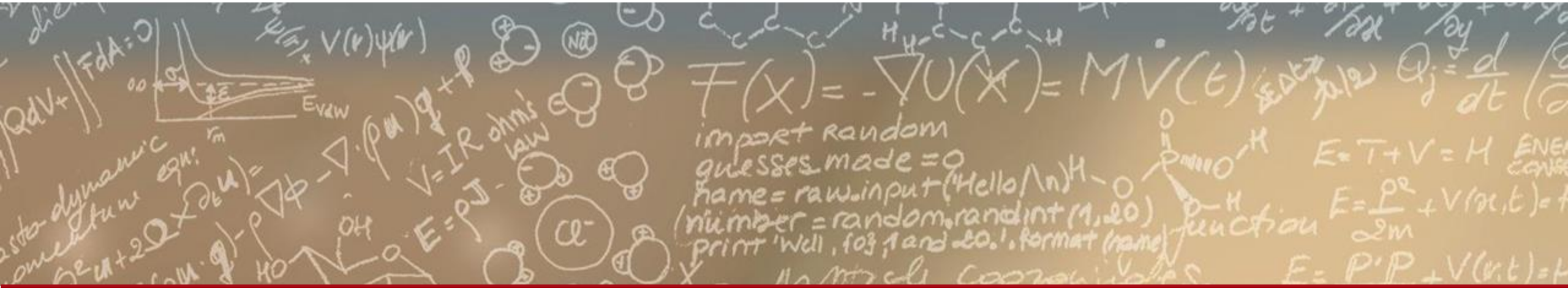
- More on FirecREST

  - Official page at CSCS: https://products.cscs.ch/firecrest/
  - FirecREST specification (latest): https://firecrest-api.cscs.ch
  - Official GitHub repository: https://github.com/eth-cscs/firecrest
  - Official Docs: https://firecrest.readthedocs.io
  - Python Library and CLI Docs: https://pyfirecrest.readthedocs.io

Requirements for today's training

- GitHub repository: https://github.com/eth-cscs/firecrest-training-2023
- OIDC Client Registration service: https://oidc-dashboard-prod.cscs.ch
- Training account for `daint` (username and password provided by CSCS)
- When submitting jobs use `--reservation=firecrest_api` in your Sbatch file
- Just a comment… you might see today the naming "**`f7t`**"





Thank you for your attention.