Japan-Taiwan Data and AI Module Platform for Analyzing Remote Sensing data, Part 3

Hidemoto Nakada, Ryosuke Nakamura, Kyoung-Sook Kim, Jason Haga, Yusuke Tanimura, Ryousei Takano, Yoshio Tanaka (**AIST**)
Hsiu-Mei Chou, Hsi-En Yu, Chun Hung Huang, Weicheng Huang (**NCHC**)

Bo Chen, Scarlet Peng (**NSPO**)

This work is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and "Secure and Distributed Data Cloud for AI Platform between Taiwan and Japan" funded by MOST in Taiwan.

The Goal

- We want to share our
  - Programs
  - Computational resources
  - Data resources

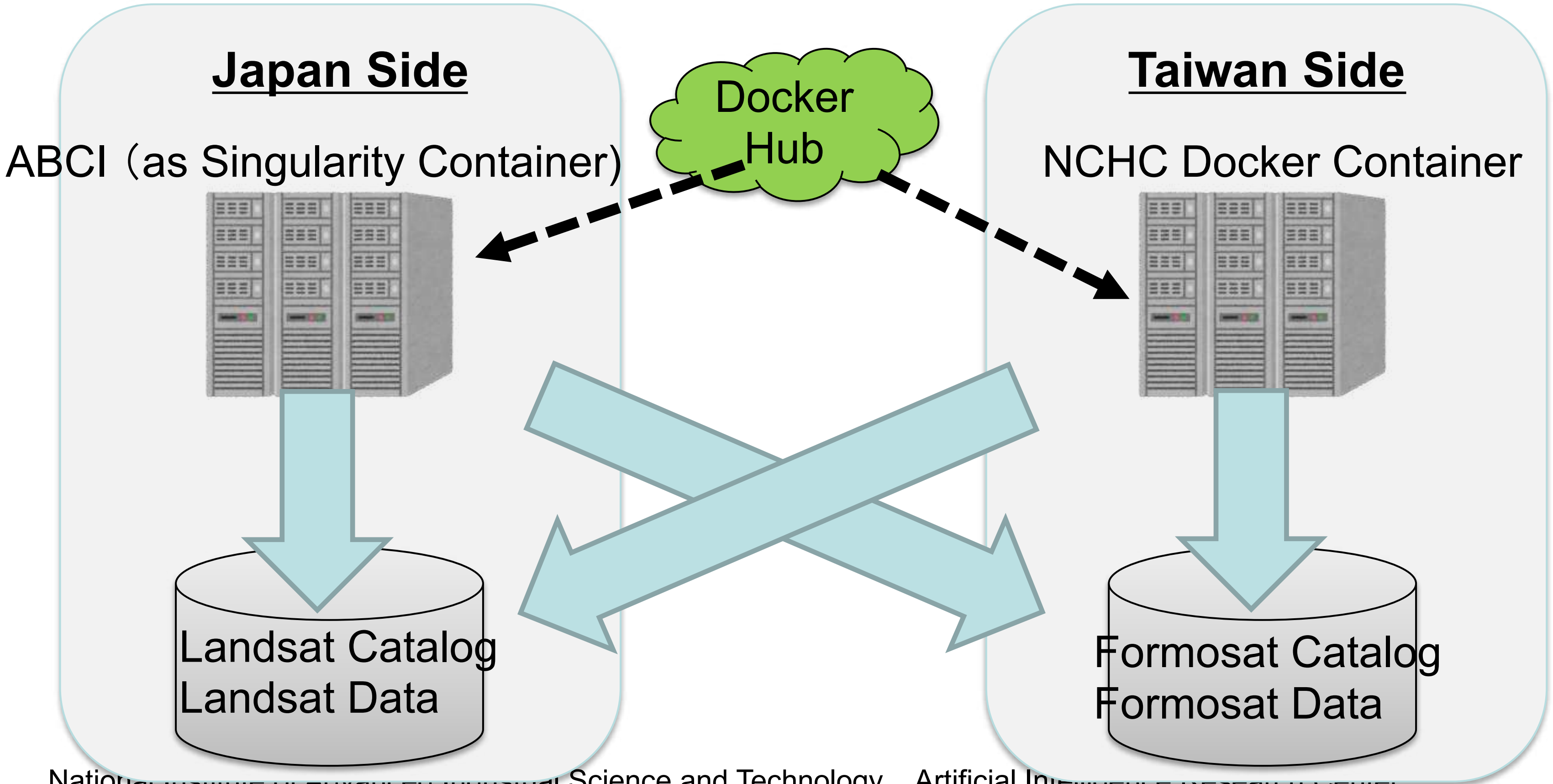
Background

- Program
  - AIST - Machine Learning module that detect objects in the satellite images
- Computers
  - AIST- ABCI – a cluster for AI
  - NCHC – Clusters
- Data
  - AIST – Landsat 8 satellite images
  - NCHC/NSPO – Formosat satellite images

→ How can we share them?

Demo at PRAGMA 36

- Sharing computing resources and data resources
- Interoperable ML module on Docker hub
- Deploy and run the module for any combination of the resources.

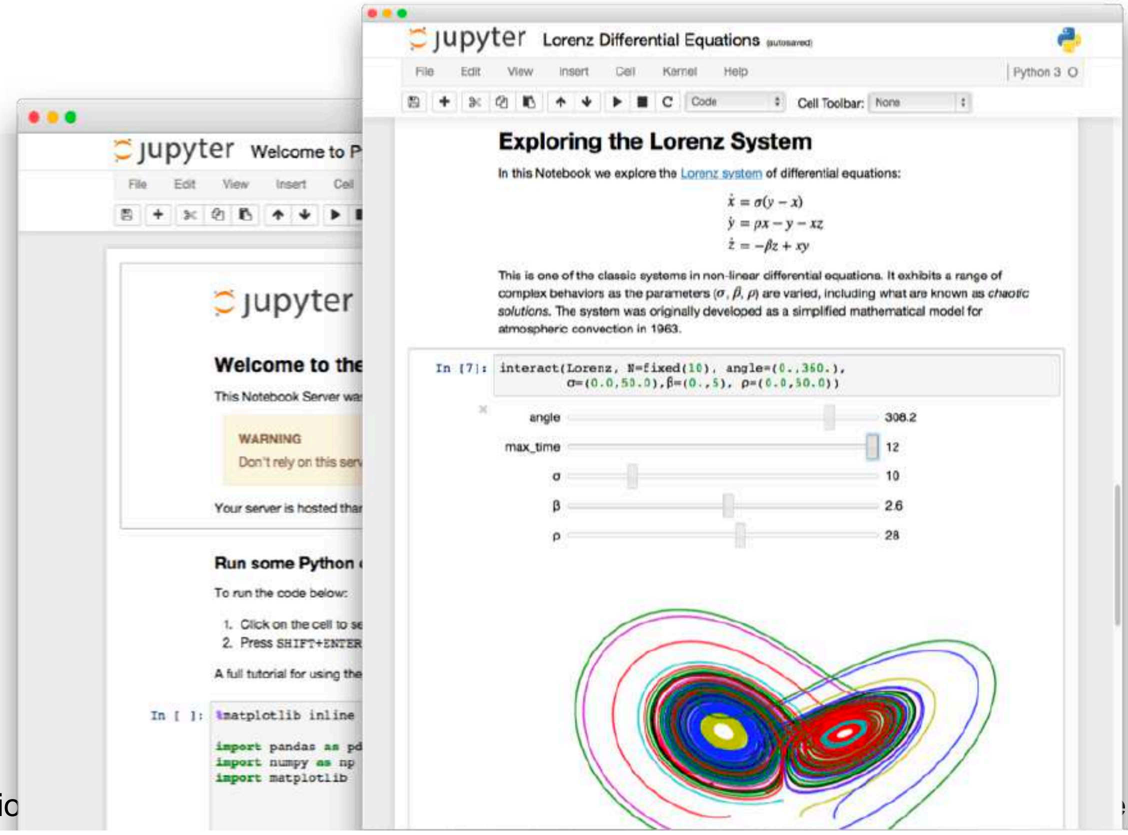


How can we share programs?

- Sharing code as a container is not enough
- **User Environment** is required
  - To explore data and codes

Jupyter Notebook

- De facto standard for data analysis tasks
- Iterative exploration on dataset



jupyter Welcome to P
File Edit View Insert Cell
jupyter Lorenz Differential Equations (autosaved)
File Edit View Insert Cell Kernel Help
Python 3
Code
Cell Toolbar: None

jupyter

Welcome to the

This Notebook Server was

WARNING

Don't rely on this serv

Your server is hosted thar

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

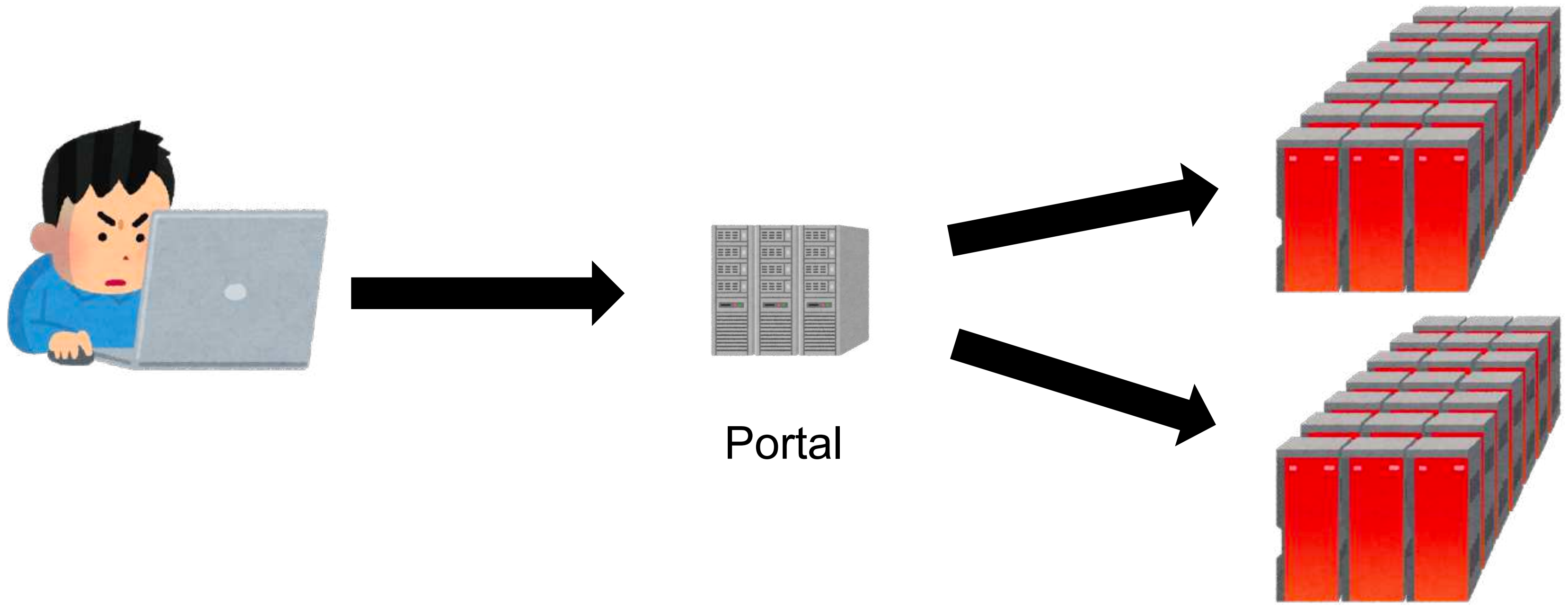
```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

angle	308.2
max_time	12
σ	10
β	2.6
ρ	28

https://jupyter.org/

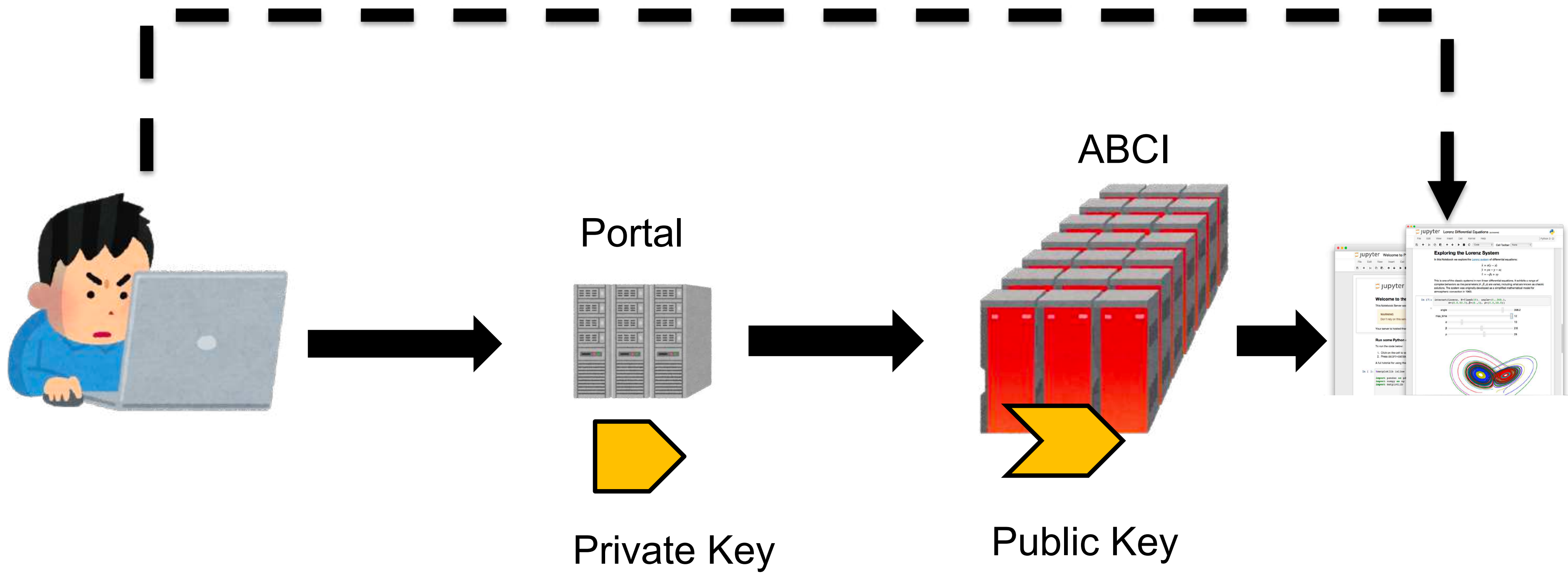
The Scenario

- Users log on to a portal via Web browser
- Select computational / data resources and code
- Start up notebooks on the specified computer resource and access it with the Web browser



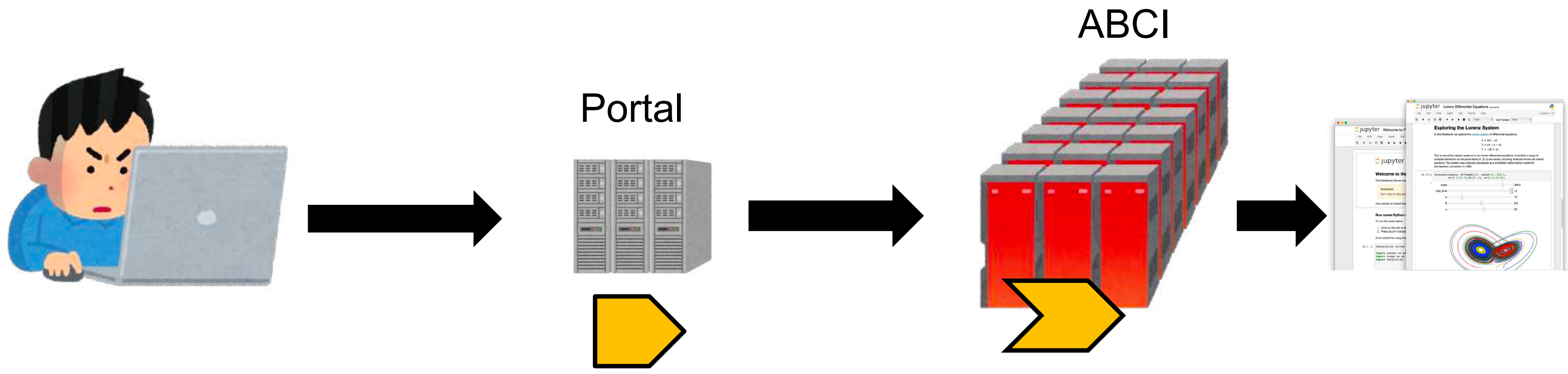
Naïve implementation

- Install the Private key on the Portal
- Directly connect to the notebook server



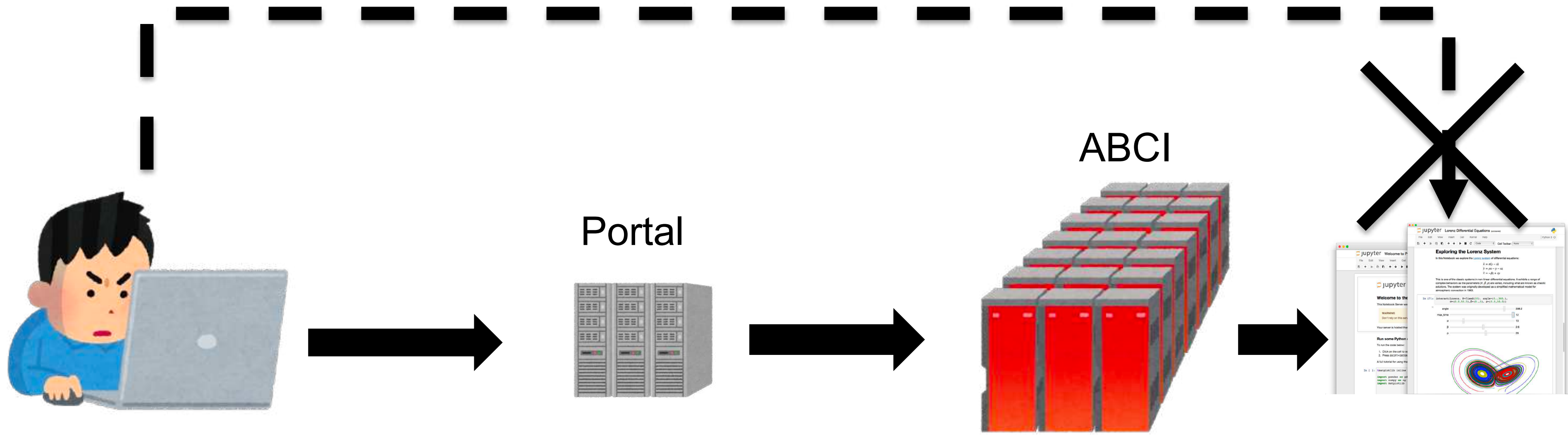
Technical Issue 1

- Where to keep private keys to log on the computational resources
  - To submit jobs to the computer, ssh connection is required
  - We don't want to keep user's private keys on the portal
    - If the portal is compromised, the attacker can do anything on the computational resources



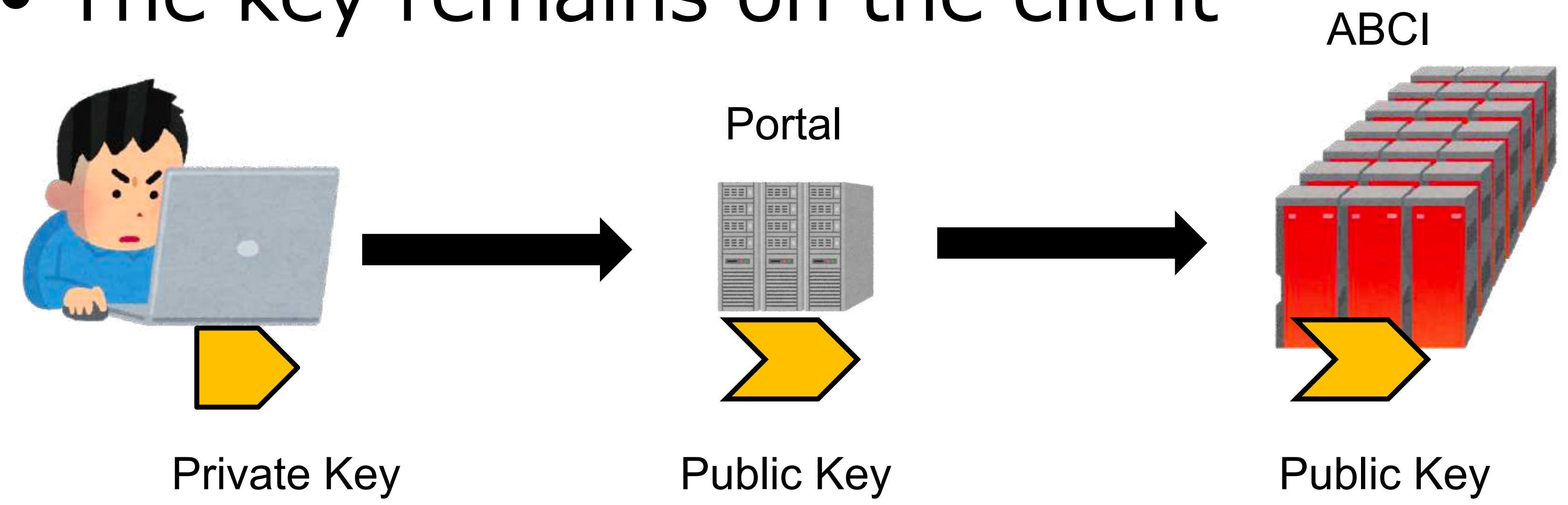
Technical Issue 2

- How to connect to the notebook on the computational resources
  - Supercomputers typically does not allow direct connections to the computational nodes
  - Outbound communication might be allowed, inbound communication is prohibited, typically.



How to Connect to supercomputers: Solution 1.

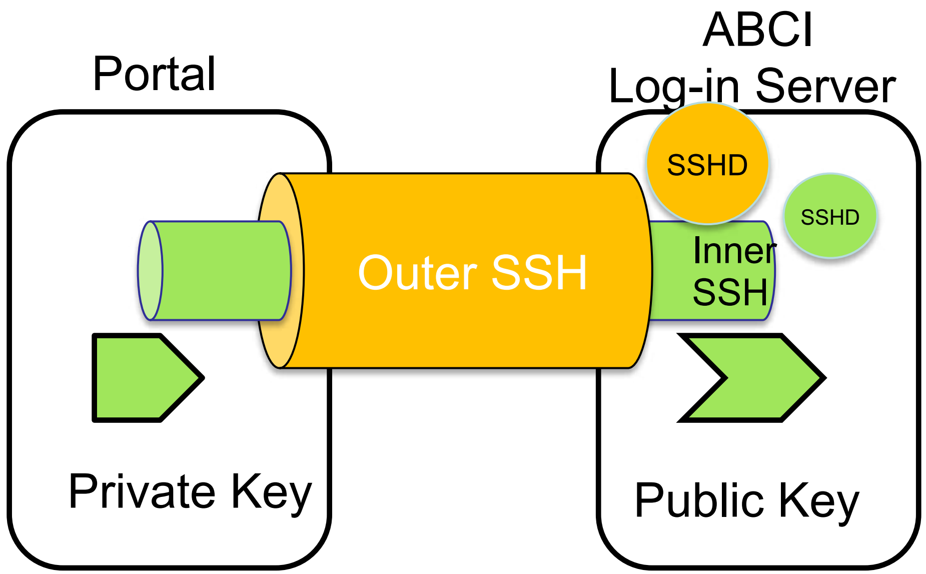
- Use ssh authentication forwarding
- The key remains on the client



- This is not enough since the user have to stay logged in on the Portal

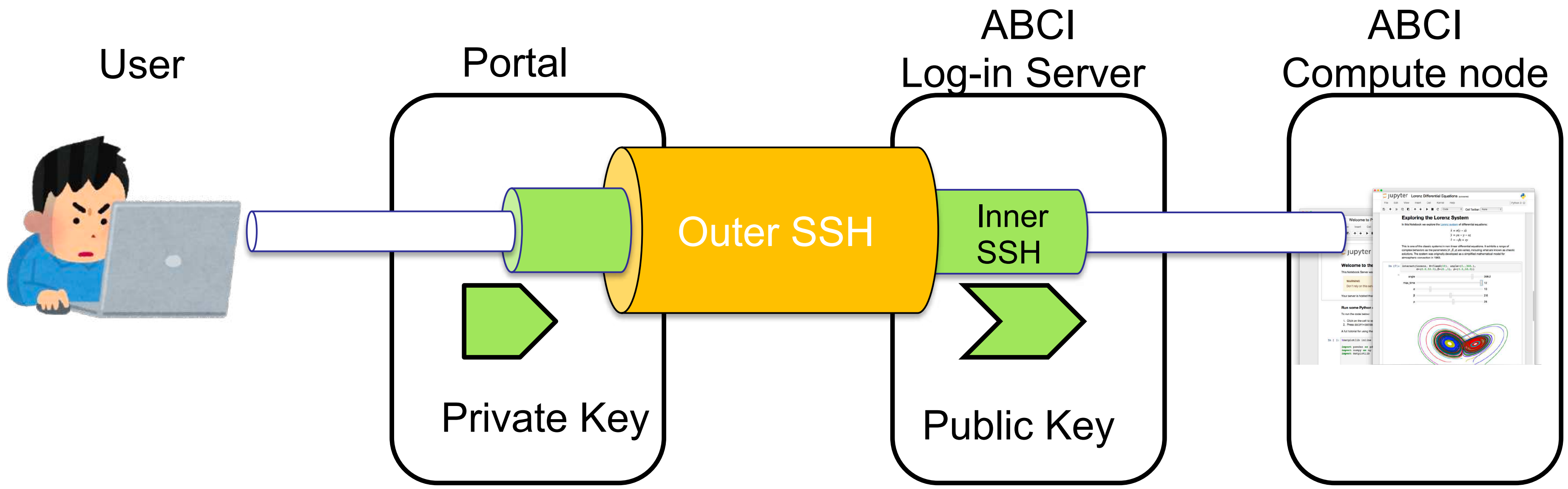
Solution 2: Nested SSH trick

- Establish a tunnel with the original keypair
  - Once the tunnel is established, the user can log out from the portal
- Invoke user-level SSHD, that works with another keypair with less capability
  - It can invoke container with notebooks only.



How to connect the notebook?

- Nested SSH tunneling!
- This is not enough, since the compute node is dynamically allocated
  - It is not possible to establish the tunnel when we start the inner SSH



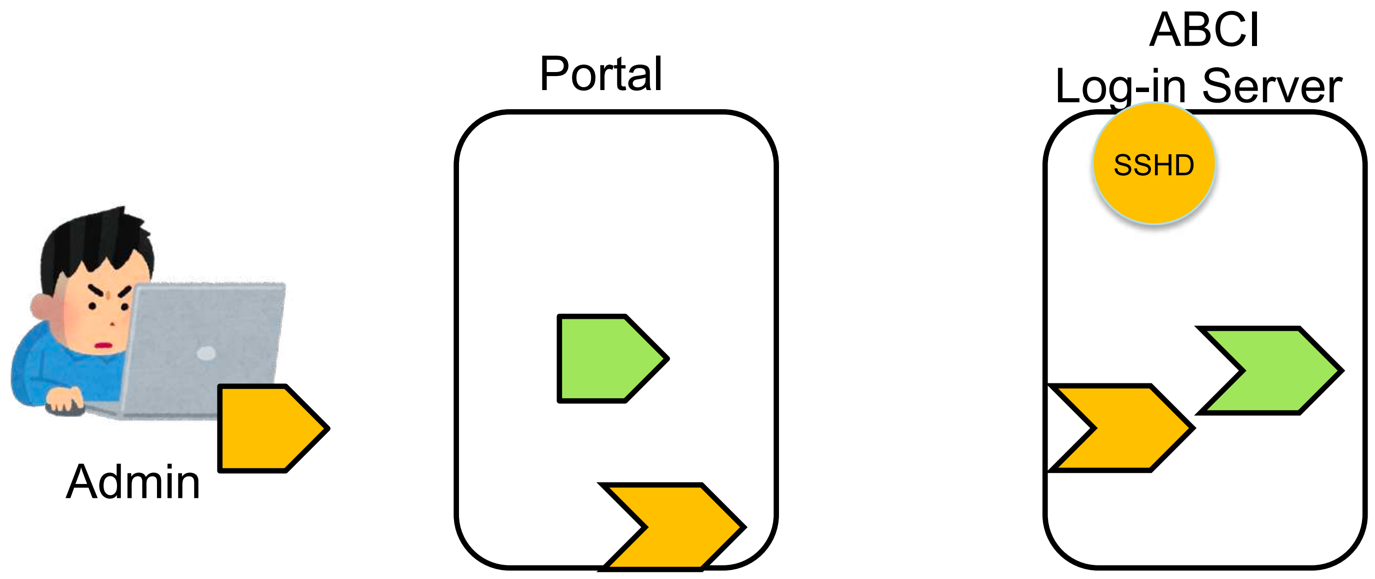
ON-the-fly tunnel management.

- We can add / delete tunnel **on the fly**
- Control Master allows us to dynamically share the connection.
- Control Master opens a socket file to control the sharing.

- > ssh –S SOCKET –O forward L:8000:HOSTNAME:8000 abci

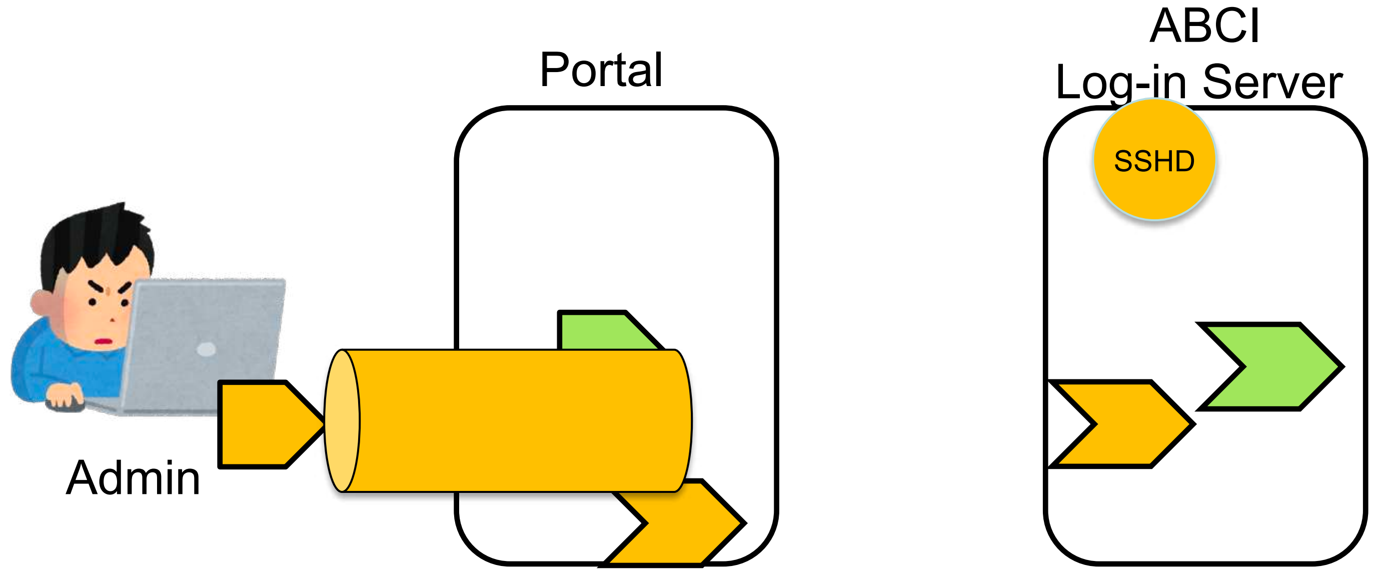
Summary

- Private Key is at the admin user's computer only
- Users can connect to the dynamically allocated computer resources



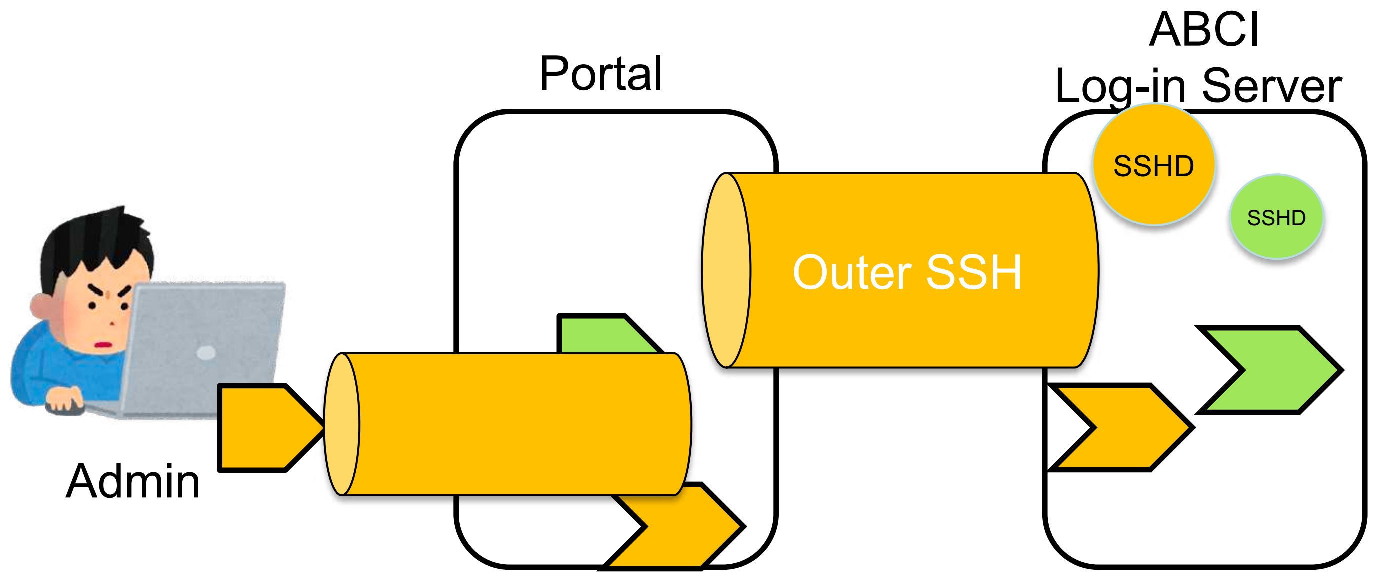
Step 1

- Admin logs in to the portal



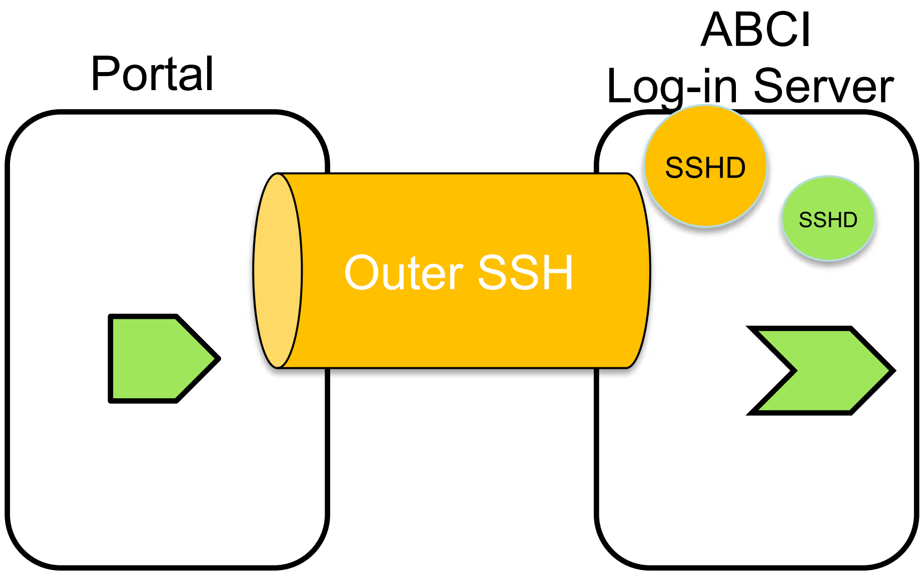
Step 2

- Establish the Outer SSH forwarding port for user-level inner SSH



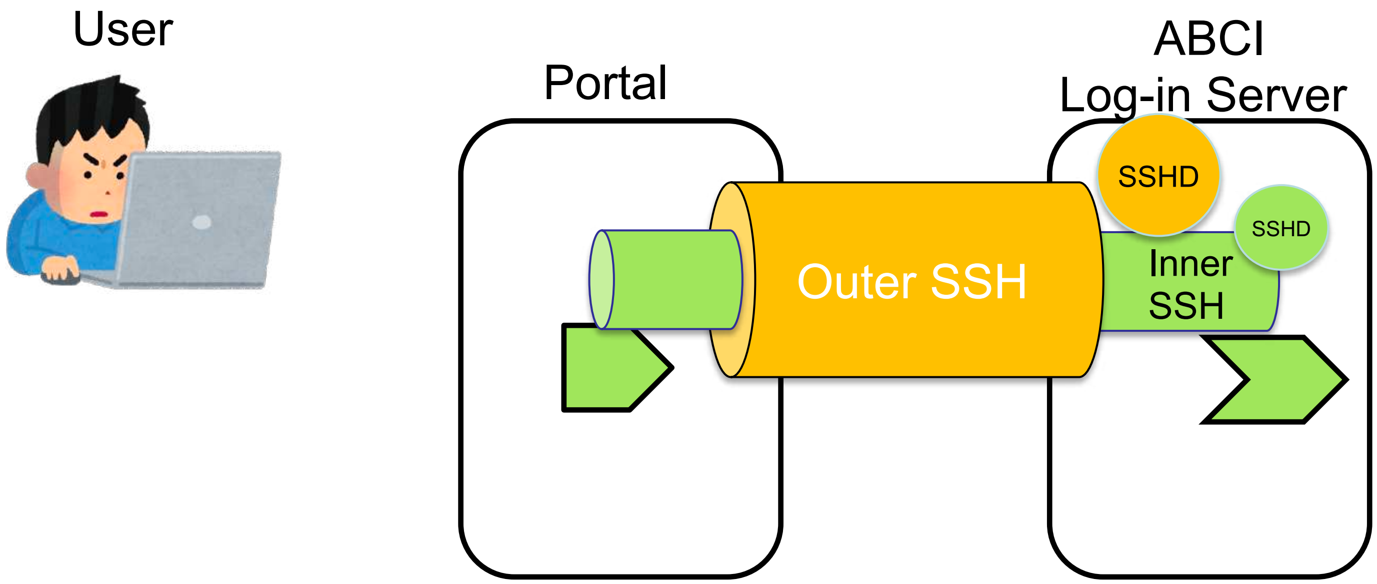
Step 2.5

- Admin logs out from portal.



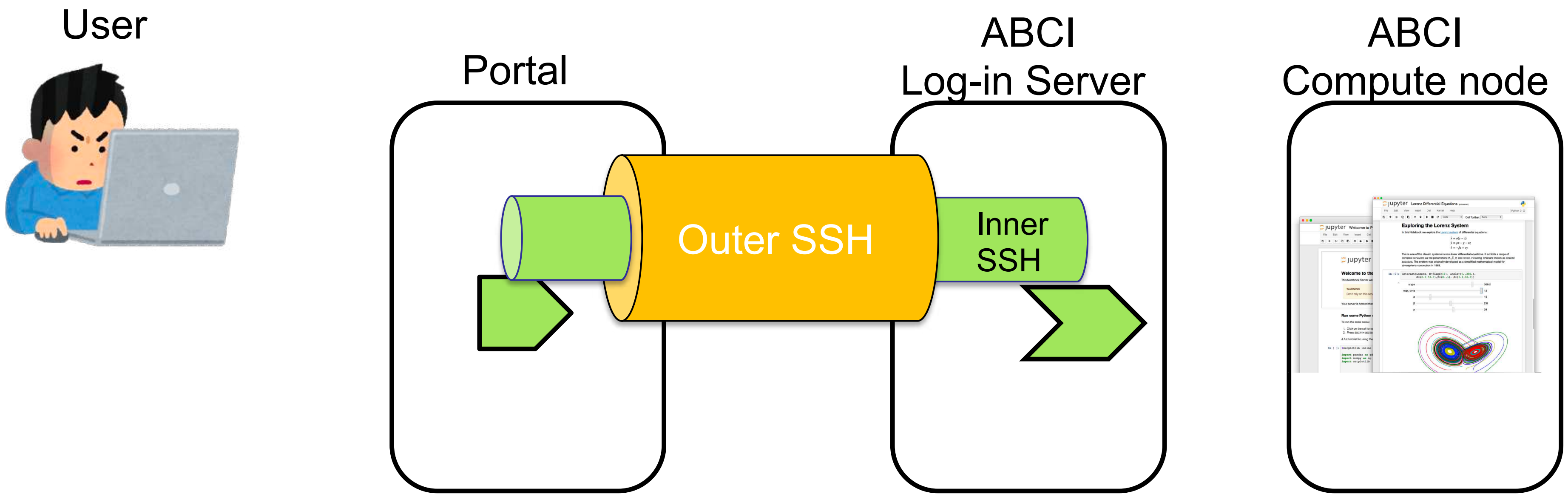
Step3

- User logs in the portal with **Web Browser**
  - and launch Jupyter notebook container process with inner SSH connection



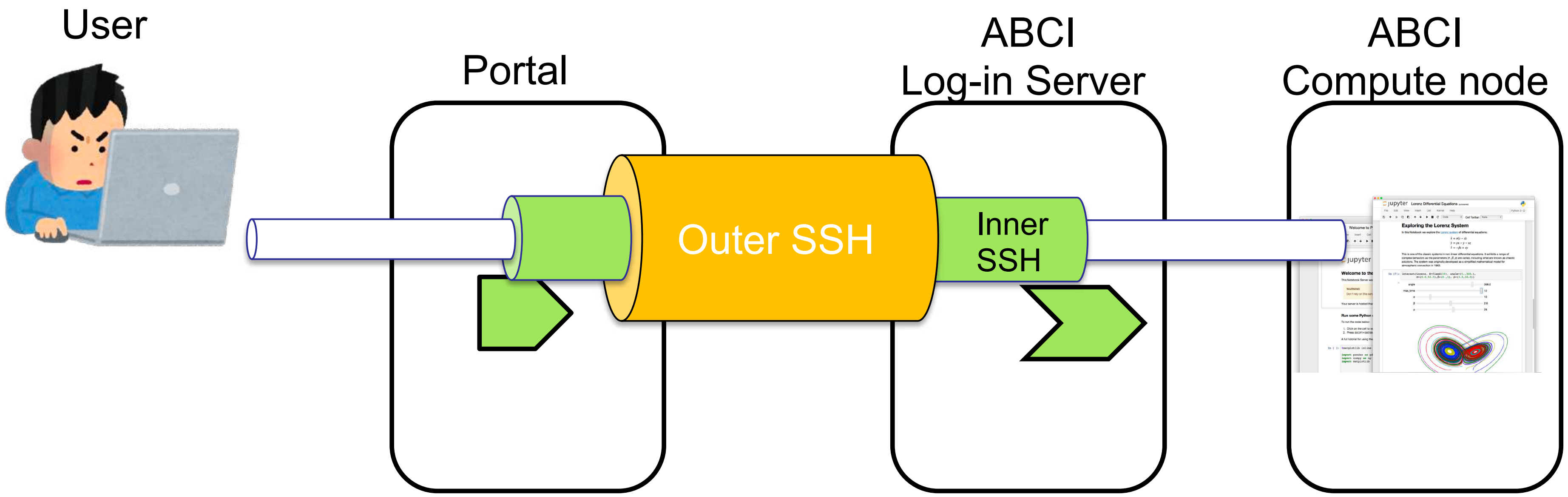
Step3.5

- Computer resources are allocated



Step4

- The Portal program add the port forwarding to give access to the user's browser.



Demo

Conclusion

- We believe Jupyter Notebook is a good candidate as a user interface for data exploration
- We can provide Notebooks on dynamically allocated computer resources with containers

Future work

- Investigate Jupyterhub and consider to integrate our portal with it
  - Jupyterhub uses outbound connection
- User authentication on the portal
- Notebook isolation
- Notebook sharing