

Provide Site Resource for Interactive Data Analysis

Jingyan Shi

shijy@ihep.ac.cn

Computing Center, Institute of High Energy Physics, Chinese Academy of Science

On Behalf of the INK Team



The 9th Asian Tier Center Forum (ATCF9)

Interactive aNalysis worKbench (INK)

- 1 Motivation**
- 2 Design and Development**
- 3 Current Status and Next Steps**
- 4 Summary**

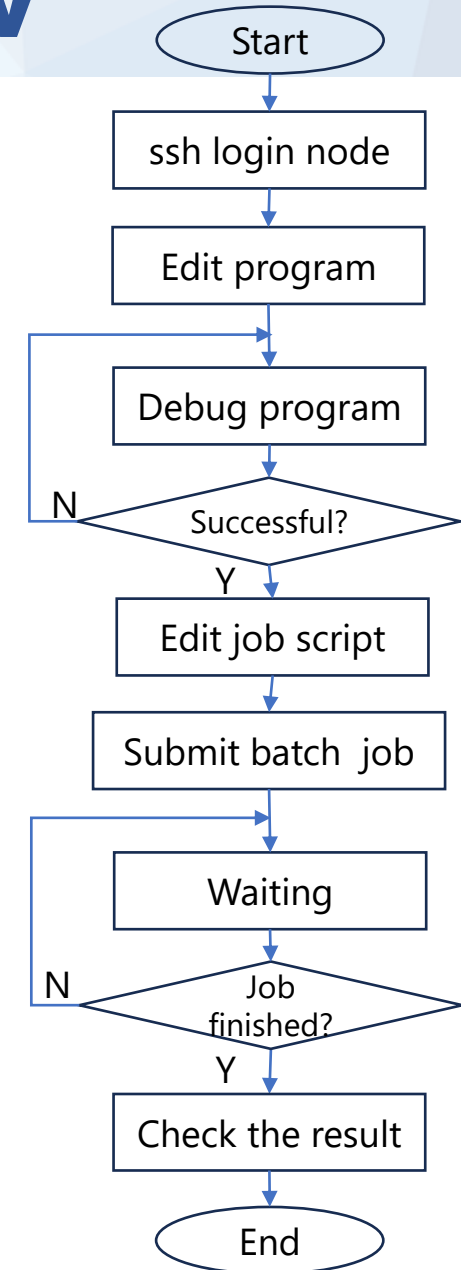
Traditional HEP Data Analysis Workflow

● Characteristics

- Closed environment
- Offline Processing
- Backend Resource Utilization

● Limitations

- Requires SSH access to login nodes for coding / debugging
- User direct file access limited to login node
- Interactive tools (VSCode, Jupyter, etc.) strain login node resources
- Need to wait the the job completion to get data result



HEP Data Analysis Way is Deeply Changed

- **New development tools:**

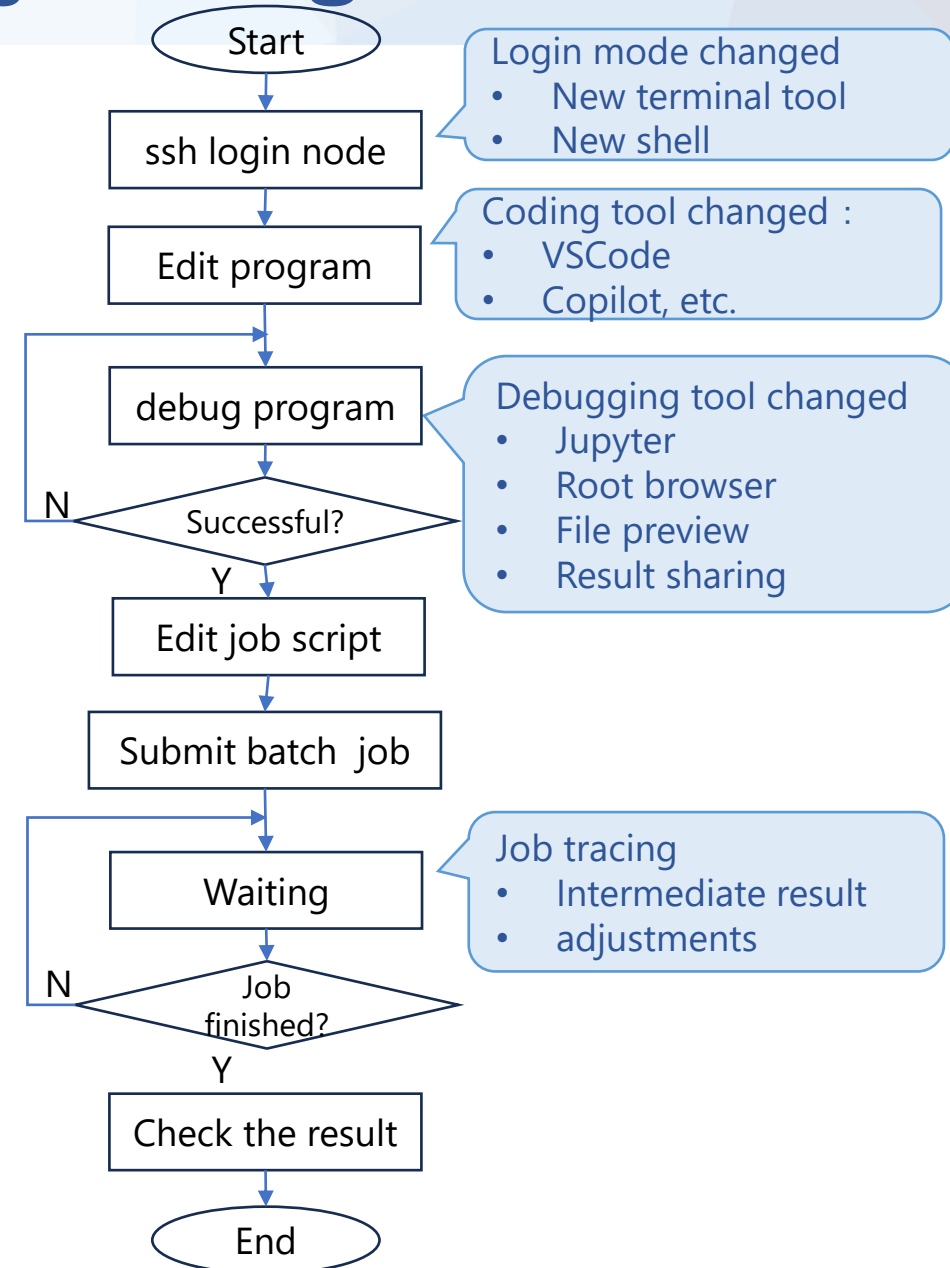
- Feature-rich IDE (VSCode, JetBrains), Interactive Programming Environment (JupyterLab), etc..

- **New coding assistance:**

- Copilot, ROOT Browser, file preview, etc..

- **New Demand for debugging & execution**

- Online plotting and result sharing
- Simplified, on-demand runtime environment setup (esp. for new comer)
- Runtime environments switching on demand



HEP Data Processing Requirements Collected

- **From Users:**

- IDE is hard to use with SSH remote mode login farm
 - Extra burden to the login node – more and more slowly
 - Disconnection happened frequently
- X-forwarding always gets stuck, when drawing figure or using root browser.
- Forbidden port forwarding due to the security reason, cannot use Jupyter Lab easily outside campus

- **From HEP Software Developers:**

- Computing resource environment is hard to deploy and use.
- No way to access file from the developer's own machine

- **Analysis facility**

- Loosely definition: infrastructure and services that provide integrated data, software and computational resources to execute on or more elements of analysis workflow
- Batch system: infrastructure of AF
 - **Transition from “Closed & Backend” to “Open & Frontend”**

Taken from
<https://hepsoftwarefoundation.org/activities-archive/analysisfacilitiesforum.html>

Solution: Interactive Platform

- **Core Concept**

- Run **interactive tools** (VSCode, Jupyter, etc.) **as cluster jobs**, delivered via web.
- Provide **secure API interfaces** for cluster job management and file access

- **Key Benefits**

- No Lag – Tools run in-container, zero client-side load
- No dependency on ssh – Used without ssh login
- Guaranteed Resources – Dedicated memory/storage, no login node strain
- Dev-Friendly – Build & integrate custom apps with cluster APIs

- **Web-First Access**

- 🌐 Zero Setup – Browser-only,
- 📁 FileOps Simplified – Upload/download/view/share files directly
- ⚙️ Easy Integration – Unified APIs for lab software/web services

INK: Interactive aNalysis worKbench

- **Visionary goal**

- Scientific research with just a network connection

- Shift backend / offline computing and storage resources to the frontend / interactive for enhanced accessibility
 - Enable flexible integration of resources and applications
 - Provide customized interactive tools for HEP experiments
 - Offer plus-and-play deployment solution for the HEP computing sites

- **Immediate goal**

- Scientific research with just a web page

- Provide web-based interactive analysis tool based on HTCondor cluster

- Leverage web technologies : VSCode, Jupyter, Rootbrowse, VNC, Enode
 - Run on the dedicated HTCondor job slot : no interference each other

- Enable external file system access

- Files could be accessed via web page directly
 - Upload / Download / View

- Provide API interfaces for cluster job submission and file access for the workflow of the experiment

Exploring INK: User Dashboard

Navigation Bar

Home Dashboard Eshell Files Ink Jobs Help zhangxuantong

Services

- Dashboard
- Eshell
- Ink Jobs
- Files

Applications

- Vscode
- Jupyter
- RootBrowse
- Enode

Services

Application CentOS7 **AlmaLinux9**

Ink Jobs Statistics

Current Interactive Jobs: 61 jobs

Statistics of Interactive Jobs

● Enode 0 ● Jupyter 8 ● Vscode 42 ● Root 1/2

My Ink Jobs

Operation	Job Type	Job ID	OS for Job	Job Status
Connect	jupyter	10786.0	AlmaLinux9	RUNNING
-	-	-	-	-
-	-	-	-	-

User Job Info

IHEP Computing Overview

Platform Overview: HTC CLUSTER | HPC CLUSTER

Running Jobs

Queueing Jobs

Exploring INK: Cluster Files Manager

The screenshot shows the INK Cluster Files Manager interface. At the top left is the INK logo. The navigation bar includes links for Home, Dashboard, Eshell, Files, Ink Jobs, Help, and a user profile for 'shijy'. Below the navigation bar is a search bar and buttons for 'Goto', 'Upload', and 'Create Dir'. A 'Refresh' button and a 'Show hidden files' checkbox are also present. The main content area shows the current directory path as '/home/cc/shijy/Ink'. Below this is a table listing files and directories with columns for Name, Permissions, Owner, Group, Size, Last Modified Time, and Operation. Callouts highlight the 'Address Bar', 'Current Directory', 'Upload File Create Directory', 'Preview', 'Share', 'Download', and 'Delete' buttons.

Address Bar

Current Directory

Upload File Create Directory

Preview

Share

Download

Delete

Name	Permissions	Owner	Group	Size	Last Modified Time	Operation
Data_1	drwxr-xr-x	shijy	u07	4 KB	2025-09-24 23:15:15	
RawData	drwxr-xr-x	shijy	u07	4 KB	2025-09-24 23:15:15	
testjobs	drwxr-xr-x	shijy	u07	4 KB	2025-09-24 23:15:43	
BESIII_Detector.jpeg	-rw-r--r--	shijy	u07	383.05 KB	2025-06-26 06:46:01	
Data_ana_1.root	-rw-r--r--	shijy	u07	770.52 KB	2025-06-05 02:03:43	
JUNO_DISPLAY.png	-rw-r--r--	shijy	u07	527.09 KB	2025-04-02 16:45:51	
JUNO_oscillation.png	-rw-r--r--	shijy	u07	2.02 MB	2025-04-02 16:45:38	
PhysRevLett.110.252001.pdf	-rw-r--r--	shijy	u07	365.55 KB	2025-04-02 16:45:25	
SVM.png	-rw-r--r--	shijy	u07	256.97 KB	2025-04-02 16:45:17	
Zc3900.png	-rw-r--r--	shijy	u07	36.18 KB	2025-06-26 06:45:45	
mos2_nopt_corr_img_fill(3).fits	-rw-r--r--	shijy	u07	2.10 MB	2025-06-04 08:06:46	

Exploring INK: EShell

```

hanx at lxlogin003 in ~
$
hanx at lxlogin003 in ~
$ bash -c 'cat /etc/motd'
*****
* Welcome to lxlogin003. http://www.centos.org
* User Manual: http://www.centos.org/docs/5.4/sect1/01.html
*****
hanx at lxlogin003 in ~
$ ls
sfs          CE          cron.sh     DIRAC       HTC          ink-test    junofs      LoginNodeActivitybackup  nginx.conf  PKI          sleep.py    TencentCloud
backup       cron_new.sh cron_task.sh HERD        ink          ISO         lnk_guide   mon             Ops          private     scratchfs   startup     test.sh
beefs       cron_old.sh cvmfs       herdfs      ink_guide   jpg         LoginNodeActivity  my_project  Other-study  public      scripts     T2T3       tra

hanx at lxlogin003 in ~
$ ls
sfs          CE          cron.sh     DIRAC       HTC          ink-test    junofs      LoginNodeActivitybackup  nginx.conf  PKI          sleep.py    TencentCloud
backup       cron_new.sh cron_task.sh HERD        ink          ISO         lnk_guide   mon             Ops          private     scratchfs   startup     test.sh
beefs       cron_old.sh cvmfs       herdfs      ink_guide   jpg         LoginNodeActivity  my_project  Other-study  public      scripts     T2T3       tra

hanx at lxlogin003 in ~
$ cd ink

hanx at lxlogin003 in ~/ink
$ ls
app_443.conf  app_login.info  hep_jupyter-lab.err.3956036.0  hist.png  ink-rootbrowse  jupyter.log  rootbrowse_hanx.cxx  start-vcn.sh.out  vscode.log
app_check2.py  fastlink-dev    hep_jupyter-lab.err.6338211.0  import.ipynb  ipacc.ipynb  rb2.py      rootfiles            start-vcnode.sh  write_in_rootbrowse.cxx
app_check3.py  fimplot.py      hep_jupyter-lab.out.3956036.0  import_ROOT_test.ipynb  joba-202010109-083329  rb3.py      sample.ipynb        test.py
app_check4.py  firefox.desktop hep_jupyter-lab.out.6338211.0  ink         juno-event-display.sh  rb.py       sample.pdf          traffic_json
app_check5.sh  hep_jupyter-lab hist.pdf                    ink-front    jupyterlab.sh  rootbrowse  start-vcn.sh        Untitled.ipynb

hanx at lxlogin003 in ~/ink
$ cd rootfiles

hanx at lxlogin003 in ~/ink/rootfiles
$ pwd
/home/ce/hanx/ink/rootfiles

hanx at lxlogin003 in ~/ink/rootfiles
$ ls
[hanx@lxlogin003 rootfiles]$ ls
AtlasExample.root  fillpatterns.root  gaxis_func.root  graph_twopad.root  tutorials_math.root
bigcanvas.root    fillrandom.root    gaxis.root       JPsi_evolution.root  VertexXY.root
dani1o6.root      fitlinear56.root   general.root     QResults.root
exclusion.root     fitslicesy.root    gr2.root         RobotoMono-Medium.woff2
ff3.root          folders.root        graph2d.root     RobotoMono-Regular.woff2
fontools.root     graph.root          sigmc.root

hanx@lxlogin003 ~]$

PID USER      NLWP PRI    NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
3362489 tengli   1  20  0  839M 352M 221M R 100.5 0.1 13:26.07 python devrun.py --seed 181655938 --nev
3559666 luoxj    75  20  0 11.7G 790M 52736 R 100.5 0.3 0:22.05 python /afs/hep.ac.cn/users/l/luoxj/Dat
3426102 tengli   1  20  0 597M 550M 290M R 99.9 0.2 9:05.31 python devrun.py --seed 181655938 --nev
5187442 qiuda    1  20  0 1044M 593M 292M I 99.3 0.2 28:00.52 pythons /eos/ks2/herd/qiuda/herdos/herdo
8508687 wangnayan 128  20  0 12.0C 315M 12644 R 99.9 0.1 3:11.60 python 1.test.generate_top.py 16 md TOS
8558100 wucy     6  20  0 1148M 510M 166M S 53.5 0.2 0:09.61 softwarenew/software2.0/dec2trigger /eos
3560319 chenzewen1 2  20  0 374M 80032 18944 S 25.6 0.0 0:01.47 /cvmfs/lhcb.cern.ch/lhcbdirac/versions/v
3536124 chenhuuyi9 1  20  0 1336M 985M 125M R 24.4 0.4 0:56.68 /dybfs2/users/liji16/Libs/root-6.32.08-i
5391 cvmfs    21  20  0 17.5C 154M 2048 S 23.2 0.1 2d19h53m /usr/bin/cvmfs2 -o rw,system_mount,fsnam
3559503 huangzhiha 18  20  0 900M 179M 79872 D 19.0 0.1 0:02.69 python 1.py
2971359 zhangfz   1  20  0 231M 14948 3584 S 17.2 0.0 7:01.16 http -u zhangfz
6780 cvmfs    1  20  0 4519M 413M 5120 S 11.3 0.2 1h45:50 /usr/bin/cvmfs2 -o rw,system_mount,fsnam
3541061 hanx     1  20  0 230M 15972 4096 R 10.7 0.0 0:05.68 http
4367 root    61  10 -10 5098M 2146M 10240 S 5.4 0.8 1d03h50m /usr/bin/eosxd /eos/user -o rw,fsname=us
2832040 xuwei    5  20  0 988M 636M 3816 S 5.4 0.2 18h37:46 btm
3535222 xpzhang 1  20  0 21476 9188 5632 S 4.8 0.0 0:07.19 sshd: xpzhang@motty
3535223 xpzhang 1  20  0 223M 4608 4096 S 4.8 0.0 0:07.83 rsync --sender -vlogDtpre.ilafx
349919 xuejingqin 12  20  0 41.4C 311M 5692 S 4.2 0.1 41:25.47 /workfs2/juno/xuejingqin/.vscode-server/ l
4159 cvmfs    2  20  0 36108 13616 5120 S 3.6 0.0 2h55:36 /usr/bin/cvmfs2 --cachesgr__ . 10 11 429
3122703 zhangyihan 1  20  0 20244 8060 5632 S 3.6 0.0 1:14.81 sshd: zhangyihan@motty
3560429 chenshiqia 1  20  0 215M 1536 1536 S 3.6 0.0 0:00.06 sleep 10s
8440525 cvmfs    1  20  0 12.5C 263M 8192 S 3.0 0.1 47:25.63 /usr/bin/cvmfs2 -o rw,system_mount,fsnam
4097848 root    61  10 -10 7391M 1716M 12800 S 3.0 0.7 3d09h55m /usr/bin/eosxd /eos/lhaaso -o rw,fsname=
2472 root    58  20  0 53140 29700 0 S 2.4 0.0 3h24:20 /usr/share/filebeat/bin/filebeat -c /etc
1157816 root    61  10 -10 6307M 935M 14336 S 2.4 0.4 19h38:10 /usr/bin/eosxd /eos/hxmt -o rw,fsname=hx
3122704 zhangyihan 1  20  0 9792 7680 5120 S 2.4 0.0 0:47.70 /usr/libexec/openssh/sftp-server
353953 xuejingqin 14  20  0 43.4C 730M 9788 S 1.8 0.3 54:05.17 /workfs2/juno/xuejingqin/.vscode-server/ l
2279139 root    55  10 -10 716M 188M 15360 S 1.8 0.1 1h31:15 /usr/bin/eosxd /eos/wcda -o rw,fsname=wc
1490495 root    61  10 -10 2524M 632M 12800 S 1.2 0.2 3h59:55 /usr/bin/eosxd /eos/argo -o rw,fsname=ar

```

SSH to login node in webpage terminal by one click.

font-size:

theme:

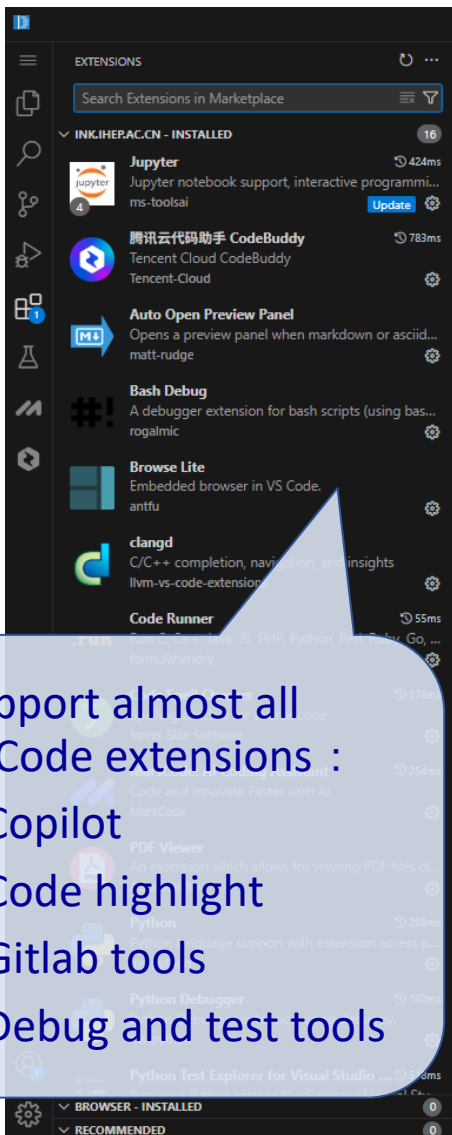
Copy Shortcut: **Ctrl+Shift+c**

Paste Shortcut: **Ctrl+Shift+v**

Support user customization

Support Tmux/Screen and other shell tools

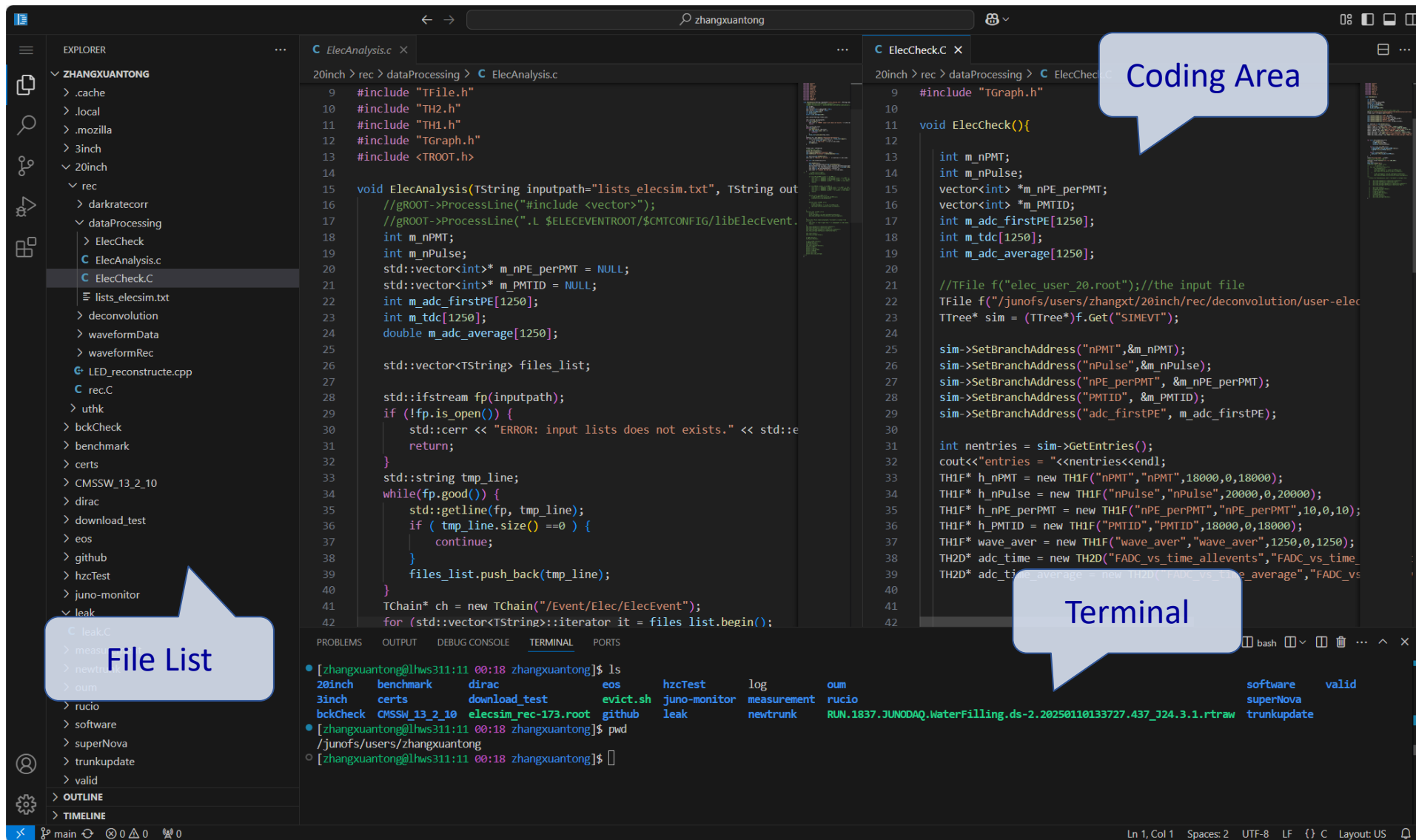
Exploring INK: VSCode



Support almost all VSCode extensions :

- ✓ Copilot
- ✓ Code highlight
- ✓ Gitlab tools
- ✓ Debug and test tools

The screenshot shows the VS Code Extensions Marketplace with various extensions installed, including Jupyter, CodeBuddy, Auto Open Preview Panel, Bash Debug, Browse Lite, clangd, and Code Runner.



Coding Area

```
#include "TFile.h"
#include "TH2.h"
#include "TH1.h"
#include "TGraph.h"
#include <TROOT.h>

void ElecAnalysis(TString inputpath="lists_elecsim.txt", TString out
//gROOT->ProcessLine("#include <vector>");
//gROOT->ProcessLine(".L $ELEVENTROOT/$CMTCONFIG/libElecEvent.
int m_nPMT;
int m_nPulse;
std::vector<int>* m_nPE_perPMT = NULL;
std::vector<int>* m_PMTID = NULL;
int m_adc_firstPE[1250];
int m_tdc[1250];
double m_adc_average[1250];

std::vector<TString> files_list;

std::ifstream fp(inputpath);
if (!fp.is_open()) {
    std::cerr << "ERROR: input lists does not exists." << std::e
    return;
}
std::string tmp_line;
while(fp.good()) {
    std::getline(fp, tmp_line);
    if ( tmp_line.size() ==0 ) {
        continue;
    }
    files_list.push_back(tmp_line);
}
TChain* ch = new TChain("/Event/Elec/ElecEvent");
for (std::vector<TString>::iterator it = files_list.begin();
```

Terminal

```
[zhangxuantong@lhws311:11 00:18 zhangxuantong]$ ls
20inch benchmark dirac eos hzcTest log measurement oum software valid
3inch certs download_test evict.sh juno-monitor leak newtrunk rucio superNova
bckCheck CMSSW_13_2_10 elecsim_rec-173.root github leak newtrunk RUN.1837.JUNODAQ.WaterFilling.ds-2.20250110133727.437_J24.3.1.rtraw trunkupdate
[zhaxuantong@lhws311:11 00:18 zhangxuantong]$ pwd
/junofs/users/zhangxuantong
[zhaxuantong@lhws311:11 00:18 zhangxuantong]$
```

The screenshot shows the VS Code interface with a File List on the left, a Coding Area in the center, and a Terminal at the bottom. The coding area contains C++ code for data processing. The terminal shows the execution of commands like 'ls' and 'pwd'.

Exploring INK: JupyterLab

The screenshot shows the JupyterLab Launcher interface. On the left is a file browser with a search bar and a list of files and folders. The main area is divided into three sections: Notebook, Console, and Other. Each section contains several kernel and editor icons. A red circle highlights the 'JUNO_PYTHON' kernel icon in the Notebook section. A blue callout bubble points to the 'JUNO_PYTHON' icon, listing available kernels. Another blue callout bubble points to the 'Terminal', 'Text File', and 'Markdown File' icons in the Other section, listing popular editors. A third blue callout bubble points to the 'firefox' file in the file browser, listing Jupyter plugins.

File browser contents:

Name	Modified
Desktop	4mo ago
Documents	4mo ago
Downloads	last mo.
firefox	3mo ago
kr5	9mo ago
milvus	3mo ago
priles	4mo ago
Pictures	4mo ago
Public	4mo ago
temp	3mo ago
Templates	4mo ago
venv	3mo ago
Videos	4mo ago
cm_agent	yesterday
dirac03.rescan.log	yesterday
dirac03.tar.gz	yesterday
files.log	8mo ago
firefox-136.0.4.t...	3mo ago
log	2mo ago
remove.sh	2mo ago
test.py	8mo ago
Untitled1.ipynb	3mo ago
untitled.txt	3mo ago
Untitled1.ipynb	2mo ago
Untitled2.ipynb	2mo ago
x509up_u10664	3mo ago

Launcher sections:

- Notebook:** Python 3 (ipykernel), fermiPy 1.4.0, Julia 1.11.5, JUNO_PYTHON, Python (LCG 107), Python 3.12 with ROOT v6.34.04, ROOT C++ v6.34.04
- Console:** Python 3 (ipykernel), fermiPy 1.4.0, Julia 1.11.5, JUNO_PYTHON, Python (LCG 107), Python 3.12 with ROOT v6.34.04, ROOT C++ v6.34.04
- Other:** Terminal, Text File, Markdown File, Julia File, Python File, Show Contextual Help

Jupyter Plugins: firefox

Jupyter Kernels:

- ✓ Python3/ROOT/Julia/FermiPy
- ✓ JUNOSW environment

Some popular editors:

- ✓ Terminal
- ✓ Text editor
- ✓ Markdown editor

Exploring INK: ROOT Browser

The screenshot displays the ROOT 7 browser interface. On the left, a file tree shows the path `hanx > ink > ink-rootbrowse > sample.root`. A table lists files with their names and sizes. A callout bubble labeled "File Path" points to the file list, and another labeled "ROOT File Contents" points to the file details.

The main area contains two histograms: "ximass" and "lammass". The "ximass" histogram has a peak at approximately 1.321. A callout bubble labeled "Interaction with ROOT Objects" points to the histogram. A statistics box for "ximass" shows: Entries: 121612, Mean: 1.321, Std Dev: 0.004413. A tooltip for the peak shows: ximass bin = 51, x = [1.321, 1.322], entries = 18159.

The "lammass" histogram has a peak at approximately 1.116. A statistics box for "lammass" shows: Entries: 121612, Mean: 1.116, Std Dev: 0.002702.

At the bottom, a command line interface shows the command `root [1] h1002->Show()` and the output of the `Show()` command, listing various ROOT objects and their values.

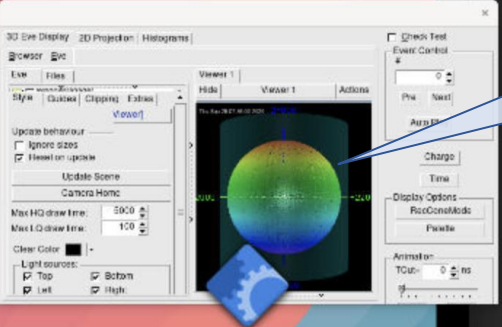
A callout bubble labeled "ROOT Command Line Interface" points to the command line.

Exploring INK: VNC

VNC is a cloud desktop, running in IHEP job cluster, so it has perfect network connection and zero I/O delay

Terminal

```
File Edit View Terminal Tabs Help
hasSim @
hasCalib 1
hasRec @
hasSimus @
initialize JVisEvtMgr successfully
initHisto
initPatQProj
initPatTProj
initEve
initEveGeom
initEvePml
Cd Digit Size : 43212
initEvePmt MainWin name:
initEveRec
initEveSimOp
```



JUNO Detector Visualization

noVNC

Ctrl

Alt

A

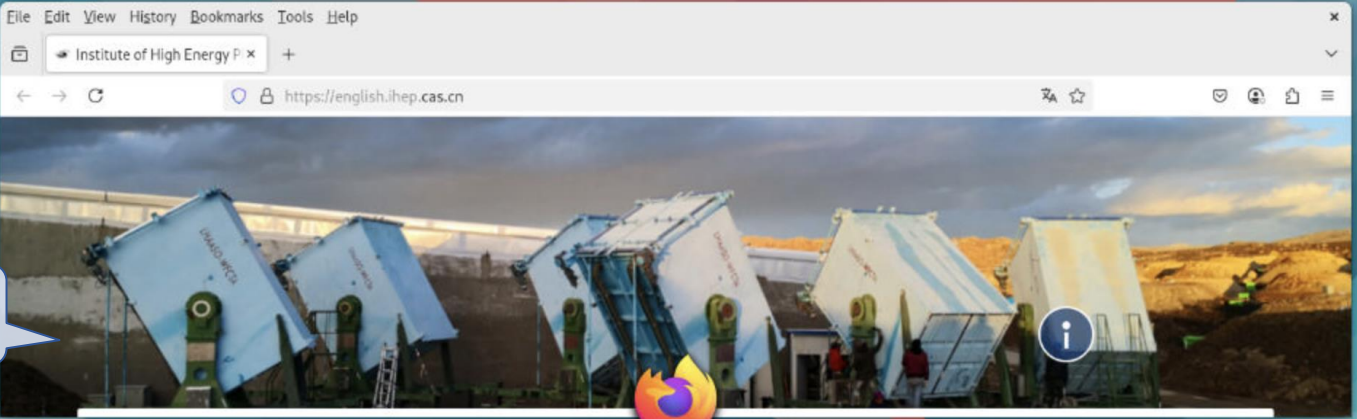
Windows

Arrows

Esc

Grid

Firefox browser



Terminal icon

Firefox icon

3D visualization icon

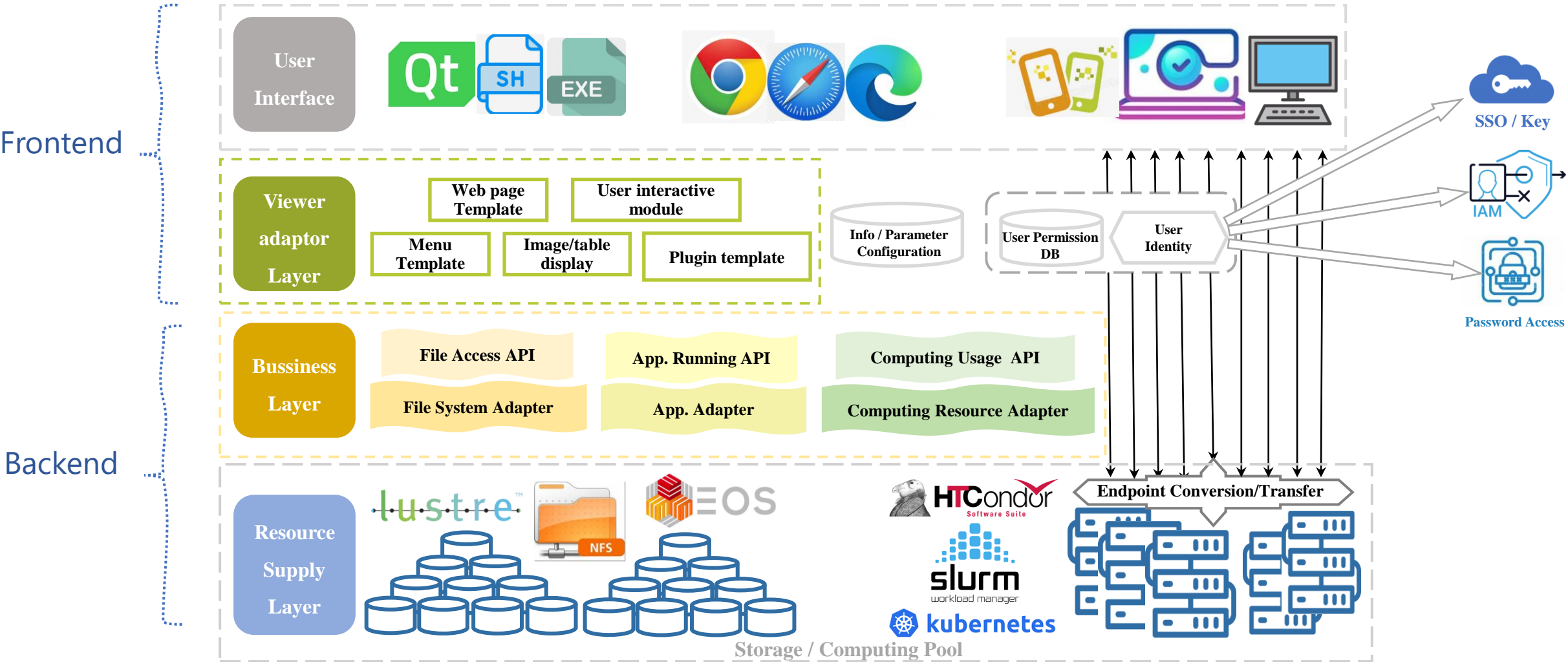
Applications icon

Other Applications

Key Development Challenges

- **Loose and flexible software architecture**
 - Customizable: each HEP experiment configure its own resource and applications
 - Configurable: resources allocation, applications, default settings
 - Ready-to-use deployment for site
- **Secure authentication & authorization**
 - Support diverse methods (SSO, token / Key, passwd, commercial account)
 - Secure for the computing and storage resource usage
 - Maintain consistent access policies for external/internal users
- **Seamless resource & application integration**
 - Various resources integration: HTCondor, Slurm, Kubernetes, EOS, Lustre etc.
 - Flexible application integration mechanism

Loose and Flexible Architecture



INK Modular Architecture

INK Modular Architecture

- Hierarchical layers structure
- Separation of frontend and backend
 - Frontend:
 - User interface layer: user way to connect resource
 - Web-based portals
 - User software
 - Viewer adapter layer: user portal generator
 - Backend:
 - Business logic layer: standardized API for the frontend interaction
 - Adapter layer: integrates diverse resources (compute, storage, applications)

Authentication and Authorization

● Authentication

- INK leverages the cluster's existing authentication mechanism(SSO, IAM, Password/Key etc.)
- Mutual trust established among INK components

● Authorization

- Dedicated INK user permission database governs
 - Available resources
 - Allowed Applications
 - Resources access rights

INK Authentication and Authorization at IHEP

● IHEP cluster authentication

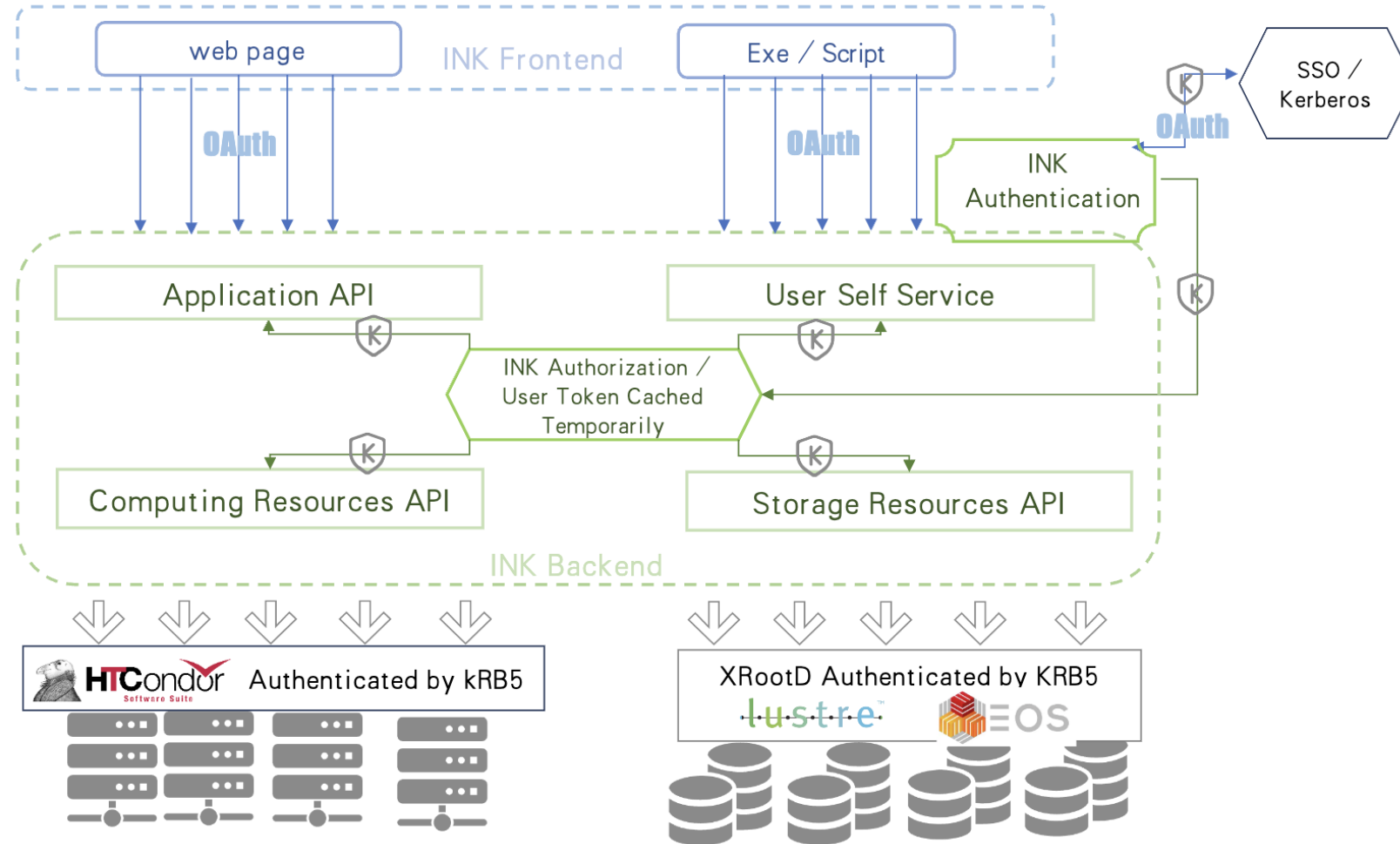
● Kerberos 5

- user authentication
- Computing & storage usage
 - HTCondor & XRootD

● OAuth for INK frontend/ backend authentication

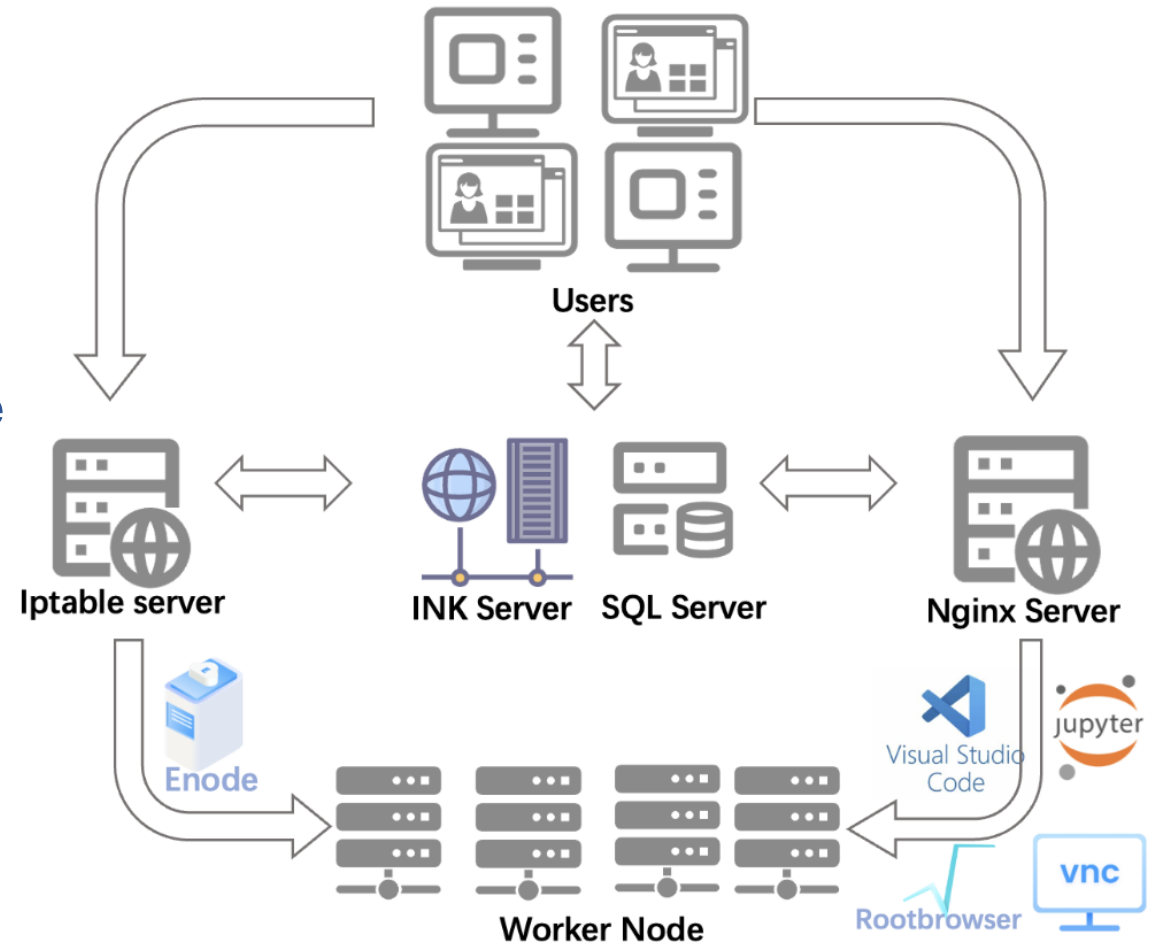
● A authorization mechanism developed

- Kerberos 5 service provides short-term user ticket
- INK caches and refreshes user tokens during the job execution



Run Application ON Worker Node

- App jobs submitted by INK
- Frontend(user) Connect to applications running on worker nodes via port forwarding
 - **Nginx Server**: handle port forwarding for VSCode, Jupyter, VNC, and Rootbrowser
 - **Iptable Server**: handle port forwarding for Enode
 - Totally **transparent** to user
- File stream connection between file storage and frontend
 - For file upload / download
 - Files viewer running inside browser
 - Fits files
 - Root files



Port forwarding for application running at worker node

Customization for Experiments (Ongoing Work)

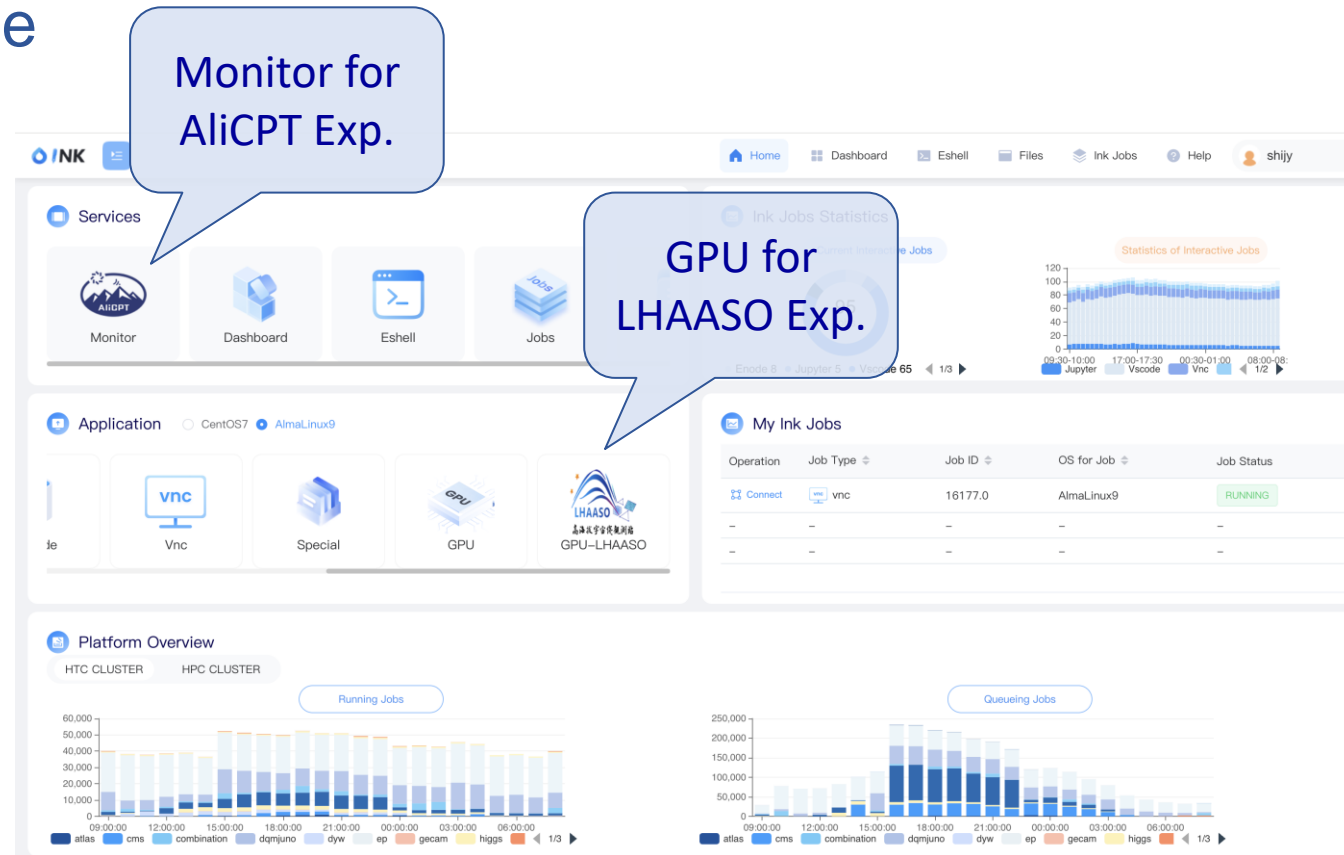
- Resources and applications database records:

- **Experiment-specific configuration**

- Resources of the experiments
- Applications of the experiments

- **User-specific configuration**

- Personalized resources and applications permission →





Current Status

- Publicly accessible at <https://ink.ihep.ac.cn>
 - First official release: Mar. 2025
 - Latest release: 2.1.0
- IHEP user access the web page directly without any extra work
 - 1 cpu core with 6GB memory for each job slots
 - Limit 24 hours walltime for the INK job
 - Provide special job slots for some special exp. user
 - Bigger memory
 - More CPU cores
- **Quite positive user feedback received!**

Next Steps

- **Customization for more experiments**
 - Typical exp. runtime environment setup behind Applications
 - Typical exp. workflow for data processing
- **More functions will be added**
 - **Job action monitor: Monitor job's resource consumed in real time and showed inside INK**
 - Memory consumed, file I/O etc.
- **Open Sourcing: Target release in October, 2025.**

Summary

- **Traditional batch system face limitations (Offline, closed and backend) for HEP data analysis now**
- **INK focus on providing an interactive usage of the cluster**
 - Provides users with a “frontend usage” for “backend resources”
 - It supports global security credentials, enabling seamless and password-free access to all resources
 - By bridging backend resources and frontend display, it allows interactive tools like VSCode to be presented via web pages.
- **Received positive feedback from users since the first version release published**
- **Further development is ongoing based on the HEP requirements**
 - Comments and suggestions are welcomed

Thank You!
Questions?