KubeCon

CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon | CloudNativeCon

THE LINUX FOUNDATION OPEN SOURCE SUMMIT

AI_dev Open Source GenAI & ML Summit

China 2024

Uniting Sustainability and Edge Computing: Kepler & Open Horizon on RISC-V and Heterogeneous Systems

Speakers: Peng Hui Jiang (IBM) & David Yao (Persistent Systems)
Date: Thursday, August 22, 2024
Time: 16:25 - 17:00 HKT
Location: Level 1 | Hung Hom Room 1

Welcome & Introduction

Peng Hui Jiang

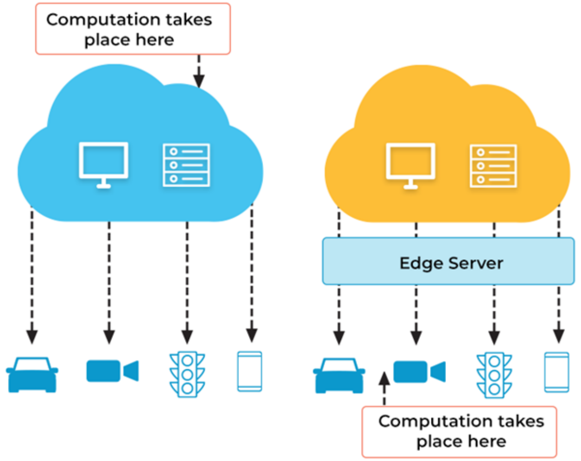
Architect and Senior
Software Engineer, IBM

David Yao

Program Director, IBM Cloud
Persistent Systems

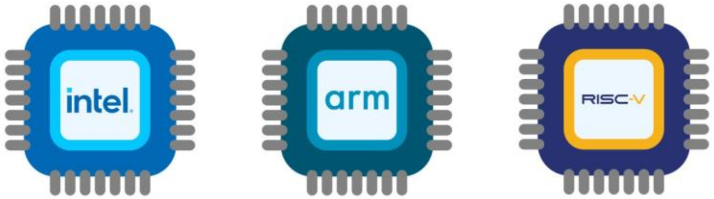
Problem Statement

- **The Dynamic Cloud-Edge Landscape**
  - **Evolution of Cloud-Edge Computing:**
    The landscape of cloud-edge computing is rapidly evolving. As the demand for processing power and efficiency grows, so do the challenges associated with energy consumption.
  - **Energy Consumption Challenge:**
    High energy consumption in edge computing impacts operational costs and environmental sustainability. Finding efficient solutions is crucial for meeting sustainability goals.



Sustainability Challenges

- **Current Sustainability Issues**
  - **Energy Inefficiencies:**
    Traditional edge computing solutions often lack efficient energy management due to the diverse hardware platforms and inefficient workload allocation.
  - **Hardware Diversity:**
    The presence of various hardware architectures (x86, ARM, s390, RISC-V) complicates efforts to optimize performance and energy usage.



Our Proposal

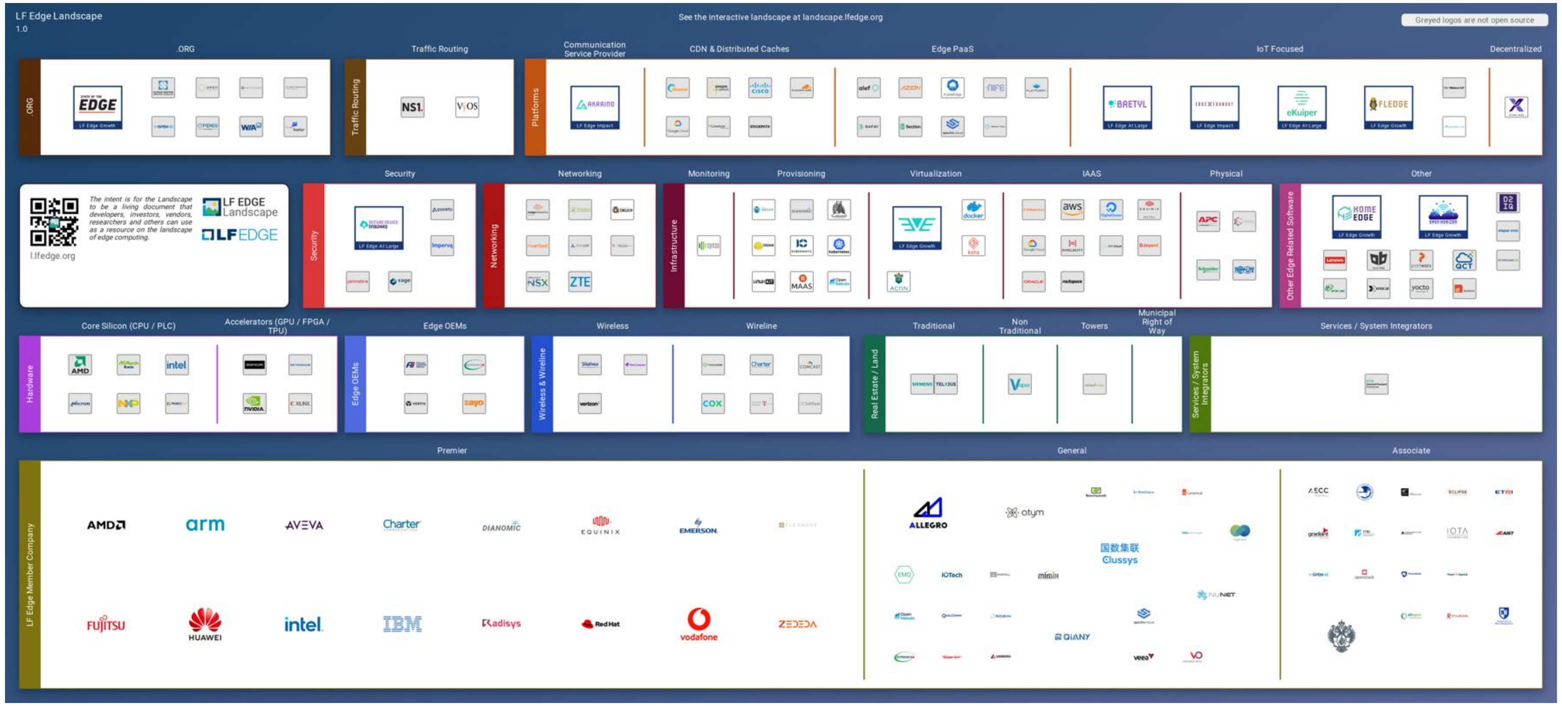
- **Integration of Kepler and Open Horizon**
  - **Kepler's Energy Estimation Capabilities:**
    Kepler offers advanced tools for energy estimation, enabling precise predictions and management of energy consumption across different workloads.
  - **Open Horizon's Workload Management:**
    Open Horizon provides autonomous workload management, improving resource allocation and operational efficiency.

KEPLER

OPEN HORIZON

Role of CNCF and LF Edge Ecosystem

- **Support for Diverse Hardware**
  - **Ecosystem Integration:** CNCF and LF Edge ecosystems support a wide range of hardware platforms, ensuring seamless integration and performance optimization.
  - **Hardware Compatibility:** These ecosystems facilitate compatibility with various hardware architectures, including x86, ARM, s390, and emerging RISC-V systems.



LF Edge Interactive Landscape

Specific Focus: RISC-V Architecture

- **Introduction to RISC-V**
  - **Overview of RISC-V:**
    RISC-V is an open-standard instruction set architecture (ISA) that provides flexibility and customization for edge computing applications.
  - **Relevance to Edge Computing:**
    RISC-V's growing adoption in edge devices offers potential for optimizing both performance and energy efficiency due to its modular and scalable nature.
  - **Advantages:**
    Customizability and cost-effectiveness of RISC-V make it an attractive choice for diverse edge computing scenarios, especially where tailored solutions are needed.

Chinese Market Insights

- **Edge Devices in the Chinese Market**
  - **Market Characteristics:**
    China has a significant presence of edge devices, particularly in sectors such as manufacturing, retail, and surveillance.
  - **Benefits of Integration:**
    Integrating Kepler and Open Horizon with various platform can greatly enhance energy efficiency and sustainability in these sectors, aligning with China's rapid technological advancements and industrial needs.
  - **Impact on Edge Devices:**
    The application of these technologies can lead to more efficient energy usage, reduced operational costs, and improved system resilience.

RISC-V

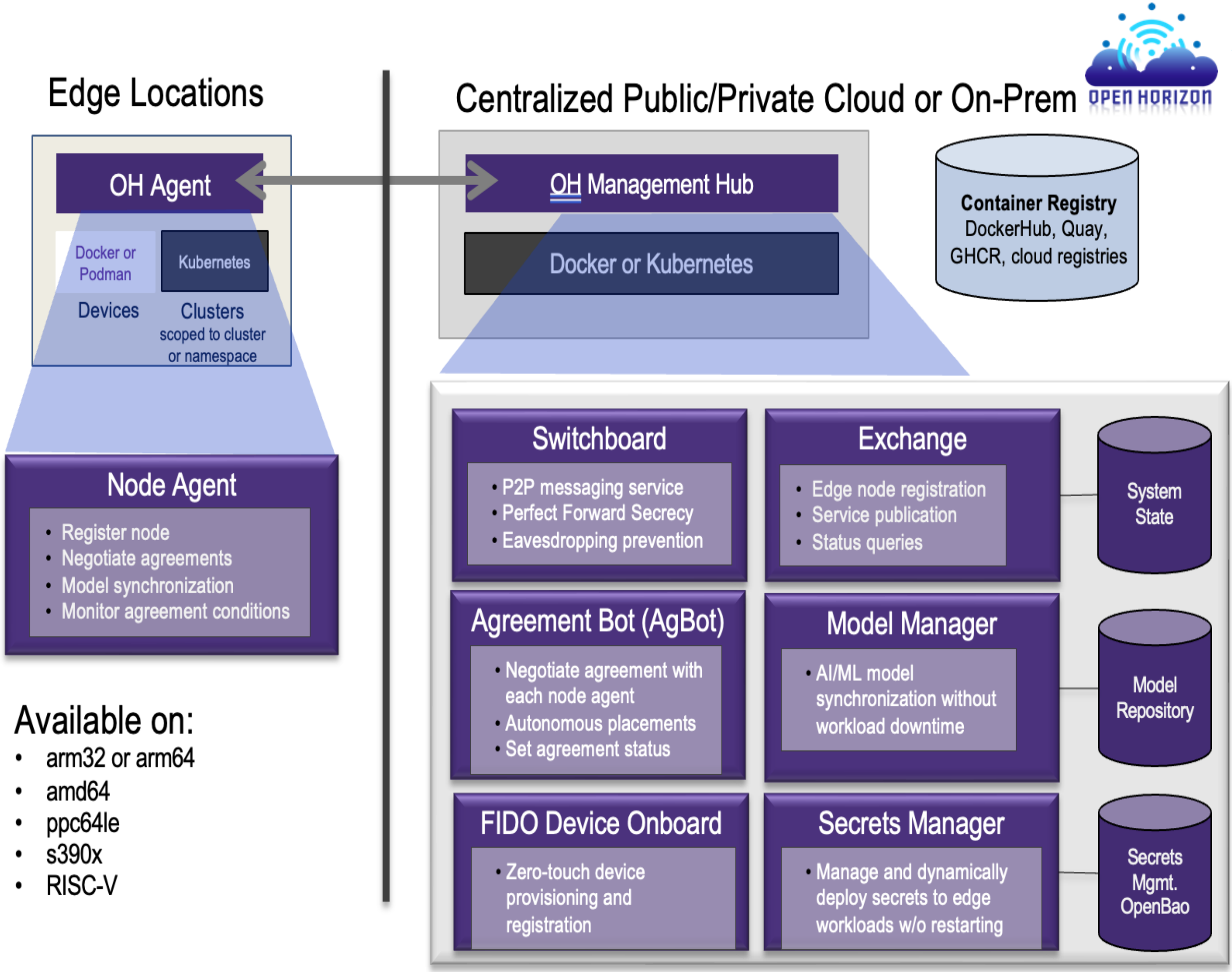
Overview of Open Horizon

Open Horizon – stage two LF Edge Project

https://www.lfedge.org/projects/openhorizon/

- Open Horizon is an open-source platform for **managing** the service software lifecycle of **containerized workloads** and related machine learning assets.
- Enables **autonomous management** of applications of edge computing nodes (clusters and devices) based on **Docker and Kubernetes** from a central management hub
- Containerized workload management and orchestration
- Model management – ML models can be re-deployed without restarting the consuming application

Main Open Horizon Components



Edge Locations

OH Agent

Docker or Podman

Kubernetes

Devices

Clusters scoped to cluster or namespace

Node Agent

- Register node
- Negotiate agreements
- Model synchronization
- Monitor agreement conditions

Centralized Public/Private Cloud or On-Prem

OPEN HORIZON

OH Management Hub

Docker or Kubernetes

Container Registry DockerHub, Quay, GHCR, cloud registries

Switchboard

- P2P messaging service
- Perfect Forward Secrecy
- Eavesdropping prevention

Exchange

- Edge node registration
- Service publication
- Status queries

System State

Agreement Bot (AgBot)

- Negotiate agreement with each node agent
- Autonomous placements
- Set agreement status

Model Manager

- AI/ML model synchronization without workload downtime

Model Repository

FIDO Device Onboard

- Zero-touch device provisioning and registration

Secrets Manager

- Manage and dynamically deploy secrets to edge workloads w/o restarting

Secrets Mgmt. OpenBao

Available on:

- arm32 or arm64
- amd64
- ppc64le
- s390x
- RISC-V

Key Principles of Open Horizon:

- **Open Source**: Trust and transparency are promoted by community-driven development
- **Simple to use**: Requires less tooling to monitor and maintain workloads than alternatives
- **More than hyperscale**: Supports fleet deployments with over 10K nodes
- **Connectivity-optional**: Service software lifecycle management shouldn't require an always-on Internet connection
- **OCI-compliant**: There are more container engines than Docker, and containers shouldn't have to be run as root
- **Single-minded**: It only solves one problem, and solves it well
- **Autonomous**: Deployment decision-making is automated through policy and constraint matching
- **Secure**: Communications are only initiated from the edge compute node, with fully encrypted messages, over secure protocols
- **Low-touch**: An administrator does not need to log in to destination nodes in order to deploy applications and metadata
- **No Code**: You can define, create, and deploy applications through configuration only

Open Horizon Empowers Industries

Boston Dynamics Robot Dog

https://www.ibm.com/case-studies/boston-dynamics

Going where no human can go

Boston Dynamics Robot Dog - Spot, a mobile robot with sensors and analytics on its back, Reduces costs associated with IoT instrumentation. Spot’s routine inspection routes and follow-up task assignments Improve worker efficiency.

Mayflower Autonomous Ship

https://mas400.com/

No captain. No crew. No problem.

6 AI powered cameras | 30 Onboard sensors | 15 Edge devices | 0 Humans on board

To sense its environment and make smart decisions without connecting to an onshore computer, IBM Edge Application Manager powered by Open Horizon manages 15 edge computing devices to process data onboard.

Smart Agriculture

https://wiki.lfedge.org/display/OH/Open+Horizon+Smart+Agriculture+SIG

Open Horizon + Liquid Prep + …

Texas A&M AgriLife

The Agriculture Innovation Mission for Climate

AgStack A Linux Foundation Project

AgriRegio

Overview of CNCF Kepler

Kepler Overview:

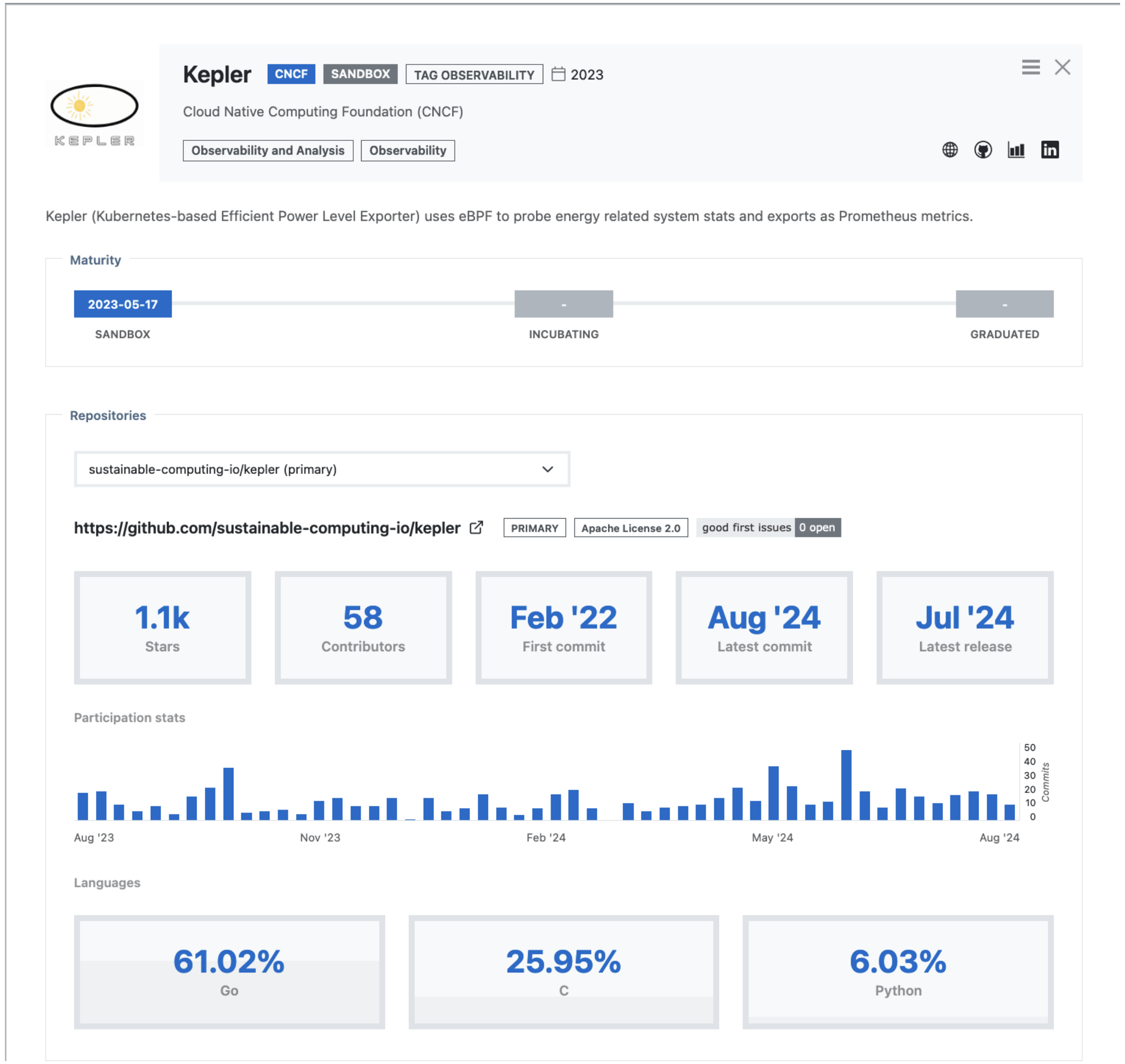
- Kubernetes-based Efficient Power Level Exporter (Kepler).
- Estimates power consumption at process, container, and Kubernetes pod levels.
- Designed to be extensible for contributions from industrial and research projects.

Kepler Architecture:

- Utilizes a BPF program integrated into the kernel's pathway.
- Extracts process-related resource utilization metrics.
- Collects real-time power consumption metrics from node components.

APIs and Models Used by Kepler:

- **Intel RAPL**: For CPU and DRAM power.
- **NVIDIA NVML**: For GPU power.
- **ACPI**: For platform (entire node) power.
- **Redfish/IPMI**: For platform power.
- **Regression-based Trained Power Models**: Used when real-time metrics are unavailable.



Kepler CNCF SANDBOX TAG OBSERVABILITY 2023

Cloud Native Computing Foundation (CNCF)

Observability and Analysis | Observability

Kepler (Kubernetes-based Efficient Power Level Exporter) uses eBPF to probe energy related system stats and exports as Prometheus metrics.

Maturity

SANDBOX	INCUBATING	GRADUATED
2023-05-17	-	-

Repositories

sustainable-computing-io/kepler (primary)

https://github.com/sustainable-computing-io/kepler PRIMARY Apache License 2.0 good first issues 0 open

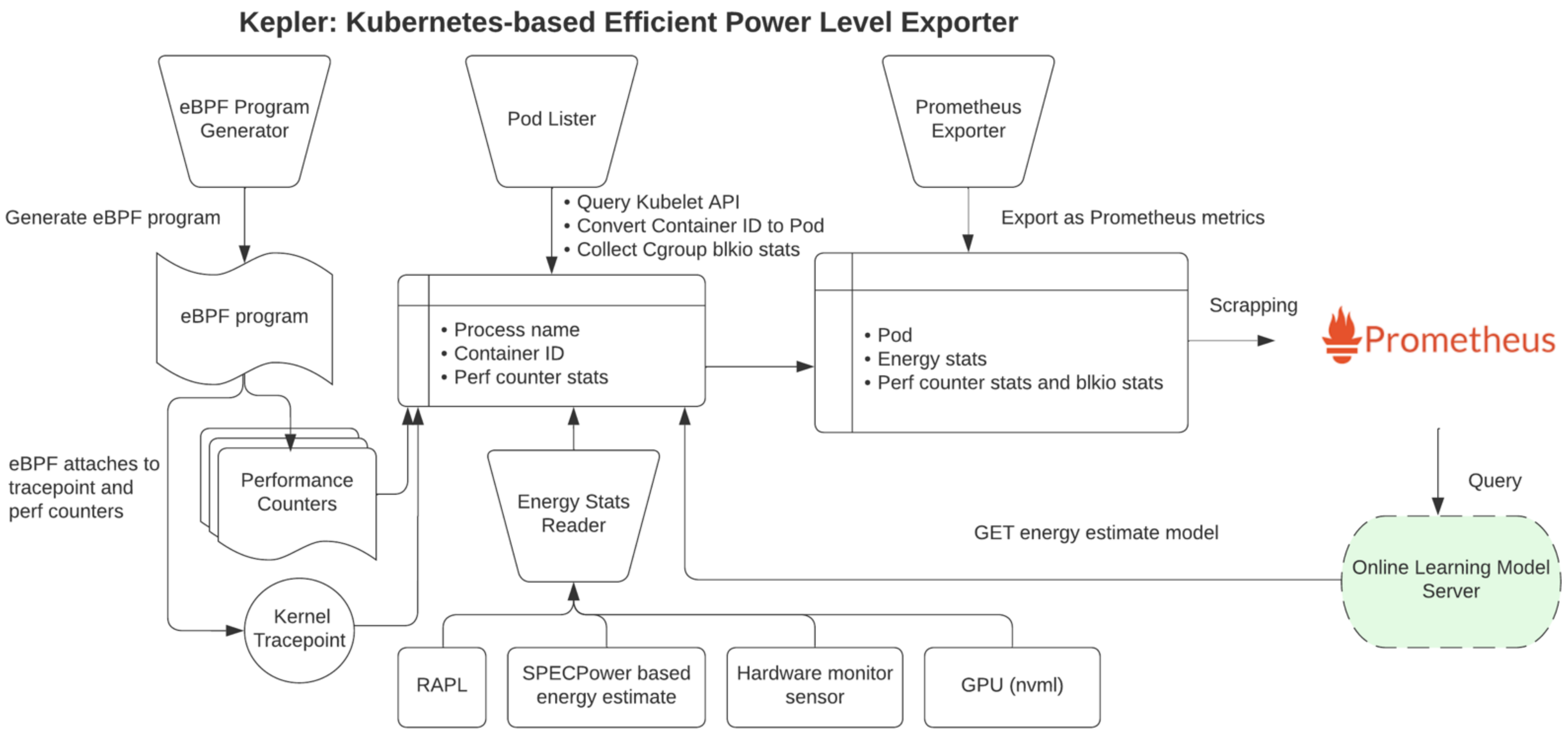
Stars	Contributors	First commit	Latest commit	Latest release
1.1k	58	Feb '22	Aug '24	Jul '24

Participation stats

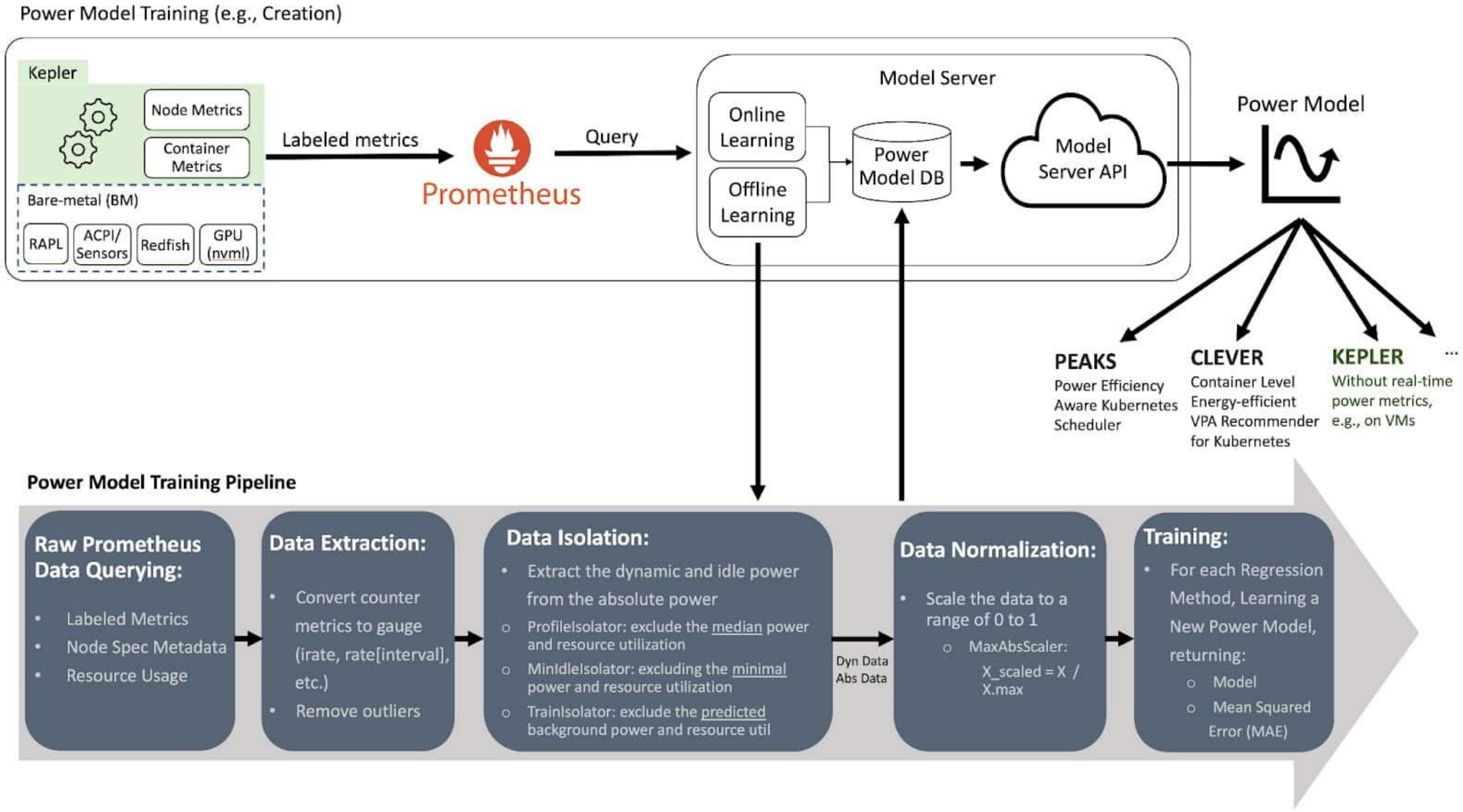
Languages

Go	C	Python
61.02%	25.95%	6.03%

Kepler - Architecture


Kepler: Kubernetes-based Efficient Power Level Exporter
eBPF Program Generator
Pod Lister
Prometheus Exporter
Generate eBPF program
• Query Kubelet API
• Convert Container ID to Pod
• Collect Cgroup blkio stats
Export as Prometheus metrics
eBPF program
• Process name
• Container ID
• Perf counter stats
• Pod
• Energy stats
• Perf counter stats and blkio stats
Scrapping
Prometheus
eBPF attaches to tracepoint and perf counters
Performance Counters
Energy Stats Reader
Query
GET energy estimate model
Online Learning Model Server
Kernel Tracepoint
RAPL
SPECPower based energy estimate
Hardware monitor sensor
GPU (nvml)

Kepler – Power Model Training



Power Model Training (e.g., Creation)

Power Model Training Pipeline

Raw Prometheus Data Querying:

- Labeled Metrics
- Node Spec Metadata
- Resource Usage

Data Extraction:

- Convert counter metrics to gauge (irate, rate[interval], etc.)
- Remove outliers

Data Isolation:

- Extract the dynamic and idle power from the absolute power
  - ProfileIsolator: exclude the median power and resource utilization
  - MinIdleIsolator: excluding the minimal power and resource utilization
  - TrainIsolator: exclude the predicted background power and resource util

Data Normalization:

- Scale the data to a range of 0 to 1
  - MaxAbsScaler:
    X_scaled = X / X.max

Training:

- For each Regression Method, Learning a New Power Model, returning:
  - Model
  - Mean Squared Error (MAE)

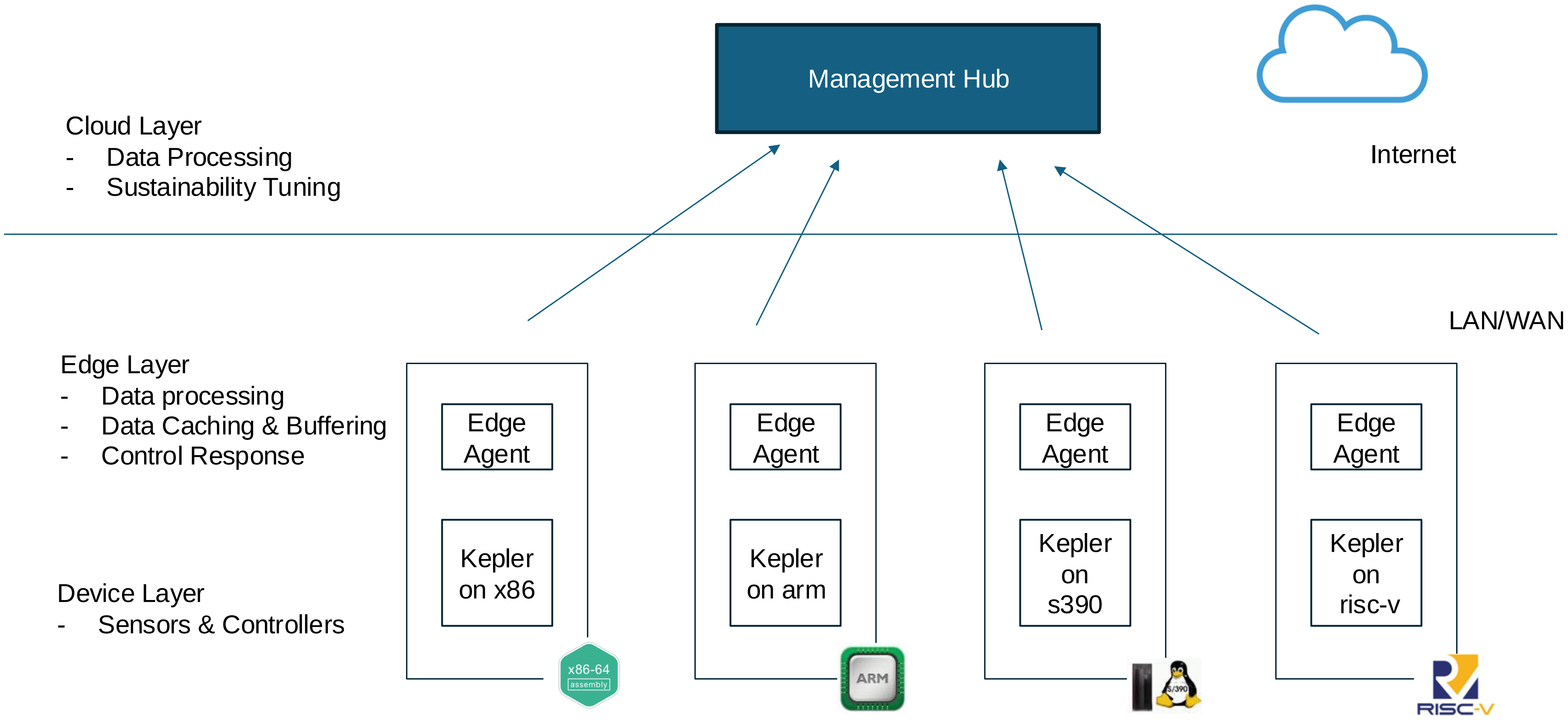
Monitoring and Optimization

- **System Monitoring**
  - **Monitoring Tools:**
    We will illustrate the tools and methods used to monitor distributed and heterogeneous systems with Kepler and Open Horizon.
  - **Optimization Techniques:**
    Our demonstration will cover real-time techniques for optimizing energy consumption and system performance.
  - **Example Scenarios:**
    We will provide examples of scenarios where monitoring and optimization lead to significant improvements in system efficiency and sustainability.

Use Case Demonstration

- **Building and Integrating Kepler & Open Horizon**
  - **Use Case Overview:**
    We will present a use case where Kepler and Open Horizon are deployed on various platforms to demonstrate their practical applications.
  - **Implementation Steps:**
    The demonstration will include steps to build and integrate Kepler and Open Horizon, showcasing how these technologies can be adapted to work effectively with various platforms.

Building out the edge computing using Open Horizon and Kepler



Cloud Layer
- Data Processing
- Sustainability Tuning

Edge Layer
- Data processing
- Data Caching & Buffering
- Control Response

Device Layer
- Sensors & Controllers

Management Hub

Internet

LAN/WAN

Edge Agent
Kepler on x86

Edge Agent
Kepler on arm

Edge Agent
Kepler on s390

Edge Agent
Kepler on risc-v

Installation of Open Horizon

Component	Version	Description
Open Horizon management hub	4.5.0	The Open Horizon management hub manages the edge nodes and the edge service software lifecycle on each edge node.
Agbot	2.30.0-1177	Agreement bot (agbot) instances are created centrally and are responsible for deploying workloads and machine learning models to Open Horizon edge nodes.
Exchange API	2.110.1-1003	The Exchange API provides a REST API to all of the Open Horizon resources (patterns, policies, services, nodes, and so on) used by all the other components in Open Horizon.
Model Management System (MMS)	1.9.10-1177	The Model Management System (MMS) facilitates the storage, delivery, and security of models and files needed by edge services. This enables edge nodes to easily send and receive models and files to and from the management hub.
Secure Device Onboard (SDO)	1.11.16-913	The Secure Device Onboarding (SDO) service enables SDO-enabled edge devices to be configured with zero touch.
FIDO Device Onboard (FDO)	1.0.0-110	The FDO component, FIDO Device Onboard, is a device onboarding scheme from the FIDO Alliance that enables technology created by Intel, which makes it simple and secure to configure edge devices and associate them with an edge management hub.
Secrets Manager	1.1.1-641	The Secrets Manager is the repository for secrets deployed to edge devices, enabling services to securely receive credentials used to authenticate to their upstream dependencies.
Edge node		Any edge device, edge cluster, or edge gateway where edge computing takes place.
Edge cluster agent	2.30.0-1177	The agent that is installed on edge clusters to enable node workload management by Open Horizon.
Edge device agent	2.30.0-1177	The agent that is installed on edge devices to enable node workload management by Open Horizon.
ESS	1.9.10-1177	The edge node part of MMS that makes AI models and files available to the edge services.

All-in-one installation instructions

```
curl -sSL https://raw.githubusercontent.com/open-horizon/devops/master/mgmt-hub/deploy-mgmt-hub.sh |
bash
```

```
----------- Summary of what was done:
  1. Started Horizon management hub services: Agbot, CSS, Exchange, FDO, Mongo DB, Postgres DB, Postgres DB FDO, Vault
  2. Created exchange resources: system organization (IBM) admin user, user organization (myorg) and admin user, and agbot
   Automatically generated these passwords/tokens:
     export EXCHANGE_ROOT_PW=K1nP4q2aFVJf5iupqkKjJ4hOWKUsiR
     export HZN_ORG_ID=root
     export HZN_EXCHANGE_USER_AUTH=root:K1nP4q2aFVJf5iupqkKjJ4hOWKUsiR      export EXCHANGE_HUB_ADMIN_PW=rka7SEry8fi5JdoBjgKMn3iVOJOb3y
     export HZN_ORG_ID=root
     export HZN_EXCHANGE_USER_AUTH=hubadmin:rka7SEry8fi5JdoBjgKMn3iVOJOb3y      export EXCHANGE_SYSTEM_ADMIN_PW=LgyHRehB599yNLYgcvdrb31IiqsDX1
     export HZN_ORG_ID=IBM
     export HZN_EXCHANGE_USER_AUTH=admin:LgyHRehB599yNLYgcvdrb31IiqsDX1      export AGBOT_TOKEN=ADqnHH3jRsLARV7GUpGP4XqdLXVZFY
     export HZN_ORG_ID=IBM
     export HZN_EXCHANGE_USER_AUTH=agbot:ADqnHH3jRsLARV7GUpGP4XqdLXVZFY      export EXCHANGE_USER_ADMIN_PW=VfflqBJhmrCslq9YsYXA96snTqaSLT
     export HZN_ORG_ID=myorg
     export HZN_EXCHANGE_USER_AUTH=admin:VfflqBJhmrCslq9YsYXA96snTqaSLT      export HZN_DEVICE_TOKEN=mW10wHWWTYFeQwiogK3VUT5aDETjHq
     export HZN_ORG_ID=myorg
     export HZN_EXCHANGE_USER_AUTH=node1:mW10wHWWTYFeQwiogK3VUT5aDETjHq    Important: save these generated passwords/tokens in a safe place. You will not be
able to query them from Horizon.
   Authentication to the Exchange is in the format <organization>/<identity>:<password> or $HZN_ORG_ID/$HZN_EXCHANGE_USER_AUTH.
  3. Installed and configured the Horizon agent and CLI (hzn)
  4. Created a Horizon developer key pair
  5. Installed the Horizon examples
  6. Created and registered an edge node to run the helloworld example edge service
  7. Created a vault instance: http://0.0.0.0:8200/ui/vault/auth?with=token
    Automatically generated this key/token:
      export VAULT_UNSEAL_KEY=ah3NPs17b4rdkc6bpMoVXdURLCGgVD4FV84o6b8Jz2g=
      export VAULT_ROOT_TOKEN=hvs.s4ZmwKfPcAvbmE69yIGTopa4    Important: save this generated key/token in a safe
place. You will not be able to query them from Horizon.
  8. Created a FDO Owner Service instance.
    Run test-fdo.sh to simulate the transfer of a device and automatic workload provisioning.
    FDO Owner Service on port 8042 API credentials:
      export FDO_OWN_SVC_AUTH=apiUser:vq05gBz8J5UYwsdNf7vRXpNVEsLEQL  9. Added the hzn auto-completion file to
~/.bashrc (but you need to source that again for it to take effect in this shell session)For what to do next,
see: https://github.com/open-horizon/devops/blob/master/mgmt-hub/README.md#all-in-1-what-next
Before running the commands in the What To Do Next section, copy/paste/run these commands in your terminal:
  export HZN_ORG_ID=myorg
  export HZN_EXCHANGE_USER_AUTH=admin:VfflqBJhmrCslq9YsYXA96snTqaSLT
```

Containers after all-in-one installation

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
00520ce2a46a	openhorizon/ibm.helloworld_amd64	/bin/sh -c /service…	15 minutes ago	Up 15 minutes		fbeb10882d3ed4888dd43410833c2591ca2b22d90128fcedc5e4063f6a81b7d6-ibm.helloworld
8a0438966fff	openhorizon/amd64_agbot	/bin/sh -c /usr/hor…	17 minutes ago	Up 16 minutes (healthy)	127.0.0.1:3110->8080/tcp, 0.0.0.0:3111->8083/tcp	agbot
47a3a2921ecd	openhorizon/fdo-owner-services	/bin/sh -c $WORKDIR…	17 minutes ago	Up 17 minutes (healthy)	0.0.0.0:8042->8042/tcp, :::8042->8042/tcp, 0.0.0.0:9008->9008/tcp, :::9008->9008/tcp	fdo-owner-services
c0ffd21c75dc	openhorizon/amd64_cloud-sync-service	/usr/edge-sync-serv…	17 minutes ago	Up 17 minutes (healthy)	0.0.0.0:9443->8080/tcp	css-api
1e7a9e4aadbb	openhorizon/amd64_vault	entrypoint.sh server	17 minutes ago	Up 17 minutes (healthy)	0.0.0.0:8200->8200/tcp	vault
ee2635fd1462	openhorizon/amd64_exchange-api	/bin/sh -c '/usr/bi…	17 minutes ago	Up 17 minutes (healthy)	8083/tcp, 0.0.0.0:3090->8080/tcp	exchange-api
1e4ac21e9077	postgres:13	docker-entrypoint.s…	17 minutes ago	Up 17 minutes (healthy)	0.0.0.0:5433->5432/tcp, :::5433->5432/tcp	postgres-fdo-owner-service
ed33f6dd36c7	mongo:4.0.6	docker-entrypoint.s…	17 minutes ago	Up 17 minutes (healthy)	27017/tcp	mongo
c189824bc51e	postgres:13	docker-entrypoint.s…	17 minutes ago	Up 17 minutes (healthy)	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	postgres

anax Agent installation

- Log into edge device and set the environment variables, replacing the <placeholders> with real values applicable for edge client:

```
export HZN_ORG_ID=<your exchange organization id>
export HZN_DEVICE_TOKEN=<specify a string value for a token>
export HZN_DEVICE_ID=<specify a string value to uniquely identify this device>
export HZN_EXCHANGE_USER_AUTH=<user id>:<password>
export HZN_EXCHANGE_URL=<management hub protocol and IP address>:3090/v1
export HZN_FSS_CSSURL=<management hub protocol and IP address>:9443/
export HZN_AGBOT_URL=<management hub protocol and IP address>:3111
export HZN_SDO_SVC_URL=<management hub protocol and IP address>:9008/api
```

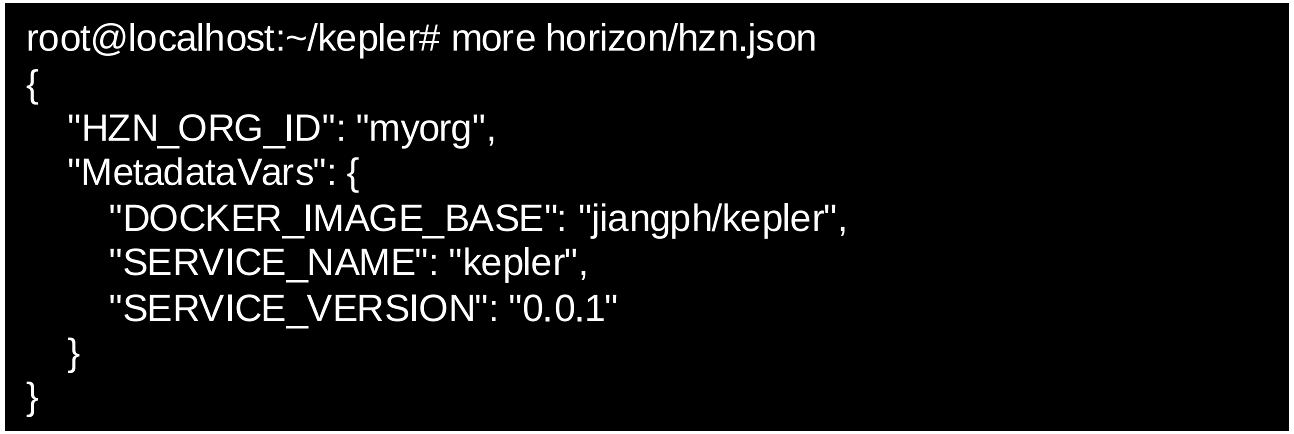
- Download and run the agent-install.sh script to get the necessary files from CSS (Cloud Sync Service), install and configure the Agent, and register edge device to run the edge service:

```
sudo -s -E curl -sSL https://github.com/open-horizon/anax/releases/latest/download/agent-install.sh | bash -s -- -
i css: -p IBM/pattern-ibm.helloworld -w '*' -T 120
```

Creating Kepler Edge Service

- export DOCKER_HUB_ID="<dockerhubid>"
- echo "<dockerhubpassword>" | docker login -u $DOCKER_HUB_ID --password-stdin
- cd to the directory in which you want to create your new service and then run this command to create the files for a simple edge service and associated Horizon metadata files:
- hzn dev service new -s kepler -i "$DOCKER_HUB_ID/kepler”

```
├── Dockerfile.amd64
├── Dockerfile.arm
├── Dockerfile.arm64
├── Makefile
├── horizon
│   ├── dependencies
│   ├── hzn.json
│   ├── pattern-all-arches.json
│   ├── pattern.json
│   ├── service.definition.json
│   ├── service.policy.json
│   ├── servicesecret
│   └── userinput.json
└── service.sh
```



```
root@localhost:~/kepler# more horizon/hzn.json
{
    "HZN_ORG_ID": "myorg",
    "MetadataVars": {
        "DOCKER_IMAGE_BASE": "jiangph/kepler",
        "SERVICE_NAME": "kepler",
        "SERVICE_VERSION": "0.0.1"
    }
}
```

Publishing A Pattern For Kepler Edge Service

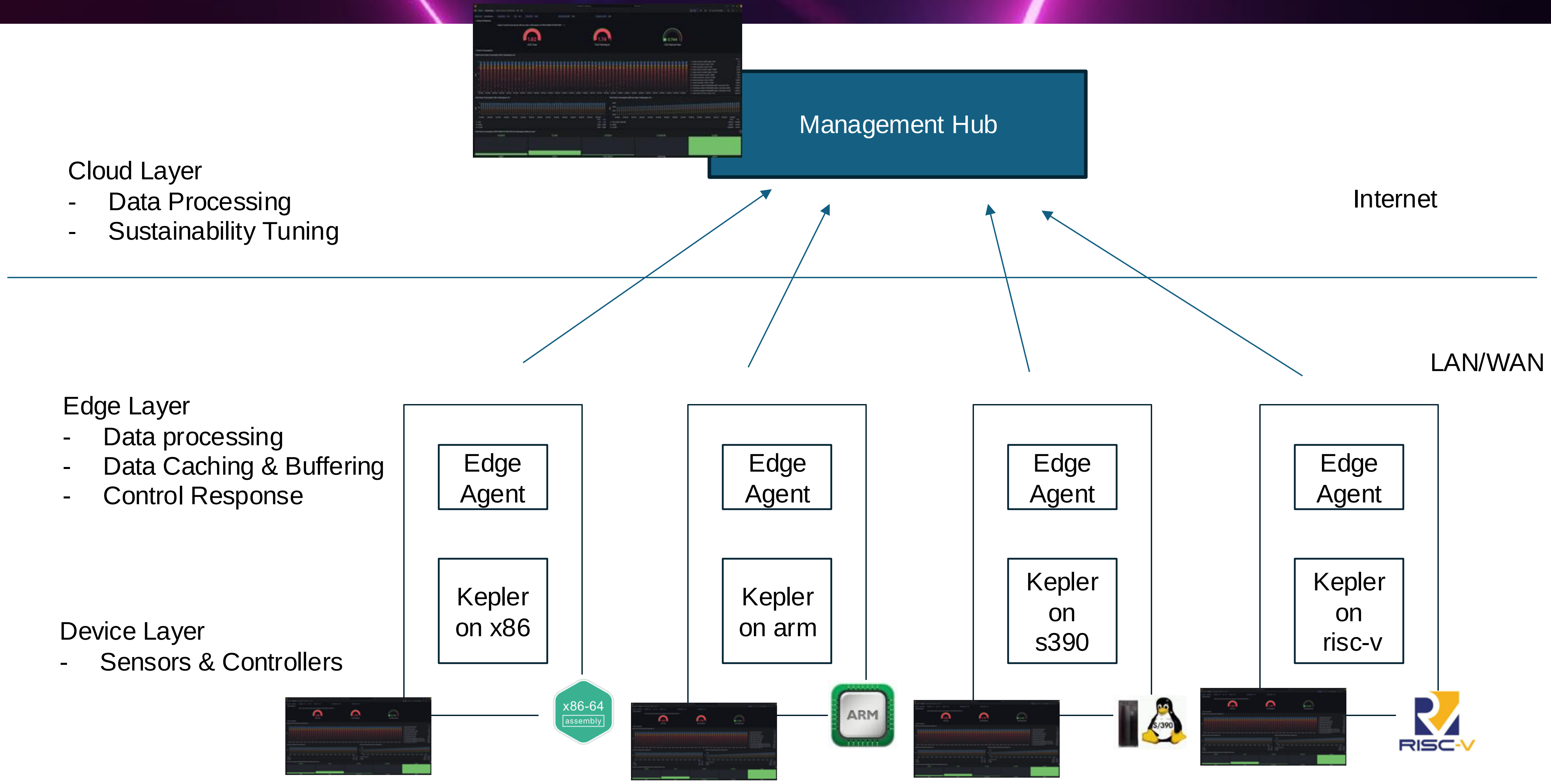
1. Publish and view your edge node deployment pattern in the Horizon Exchange:
   ```
   hzn exchange pattern publish -f horizon/pattern.json hzn exchange pattern list
   ```
2. Register your edge node with Horizon to use your deployment pattern:
   ```
   hzn register -p pattern-${SERVICE_NAME}-$(hzn architecture)
   ```
3. The edge device will make an agreement with one of the Horizon agreement bots (this typically takes about 15 seconds). Repeatedly query the agreements of this device until the agreement_finalized_time and agreement_execution_start_time fields are filled in:
   ```
   hzn agreement list
   ```
4. After the agreement is made, list the docker container edge service that has been started as a result:
   ```
   sudo docker ps
   ```
5. See the kepler service output:
   ```
   hzn service log -f $SERVICE_NAME
   ```
6. Unregister your edge node (which will also stop the kepler service):
   ```
   hzn unregister -f
   ```

Publishing A Pattern For Kepler Edge Service

```
export HZN_ORG_ID=myorg

export
HZN_EXCHANGE_USER_AUTH=admin:zFrOei3wPV8NjsBE
uaAkG8PBQGxDzO

export HZN_FSS_CSSURL=http://50.116.43.231:9443/

export HZN_EXCHANGE_URL=http://50.116.43.231:3090/v1

export HZN_AGBOT_URL=http://50.116.43.231:3111

export HZN_FDO_SVC_URL=http://50.116.43.231:9008/api

curl -sSL https://github.com/open-
horizon/anax/releases/latest/download/agent-install.sh | bash -
s -- -i anax: -k css: -c css: -p myorg/pattern-kepler-amd64 -w '*'
-T 120

hzn service log -f myorg/kepler
```

```
root@192-155-95-241:~# hzn agreement list
[
  {
    "name": "pattern-kepler-amd64_kepler_myorg_amd64 merged with pattern-kepler-
amd64_kepler_myorg_amd64",
    "current_agreement_id":
"73e5a8d8c4edf69fd283a932dfa25706ebd1d16b71de0d56aab0c0f959c490f6",
    "consumer_id": "IBM/agbot",
    "agreement_creation_time": "2024-08-06 03:41:22 +0000 UTC",
    "agreement_accepted_time": "2024-08-06 03:41:28 +0000 UTC",
    "agreement_finalized_time": "2024-08-06 03:41:29 +0000 UTC",
    "agreement_execution_start_time": "2024-08-06 03:41:43 +0000 UTC",
    "agreement_data_received_time": "",
    "agreement_protocol": "Basic",
    "workload_to_run": {
      "url": "kepler",
      "org": "myorg",
      "version": "0.0.1",
      "arch": "amd64"
    }
  }
]
```

Monitoring using Kepler



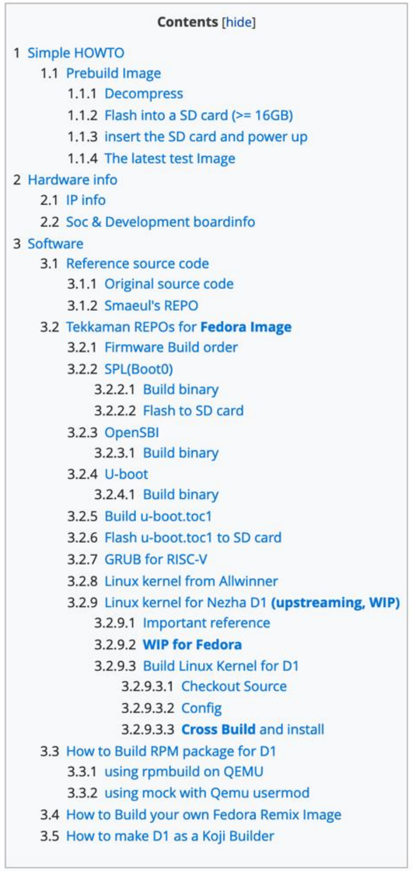
Cloud Layer
- Data Processing
- Sustainability Tuning

Edge Layer
- Data processing
- Data Caching & Buffering
- Control Response

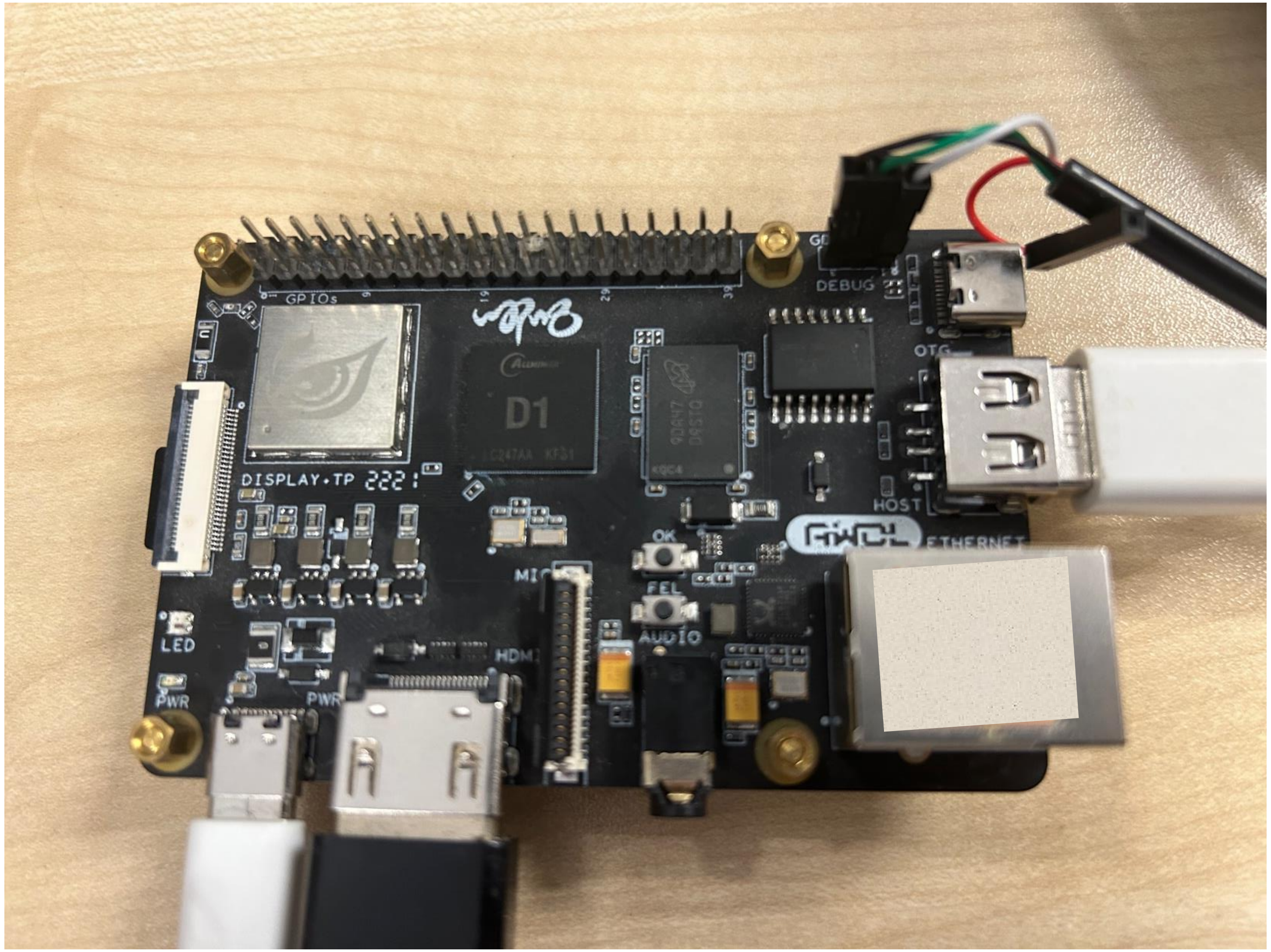
Device Layer
- Sensors & Controllers

Risc-V Experiment (哪吒)

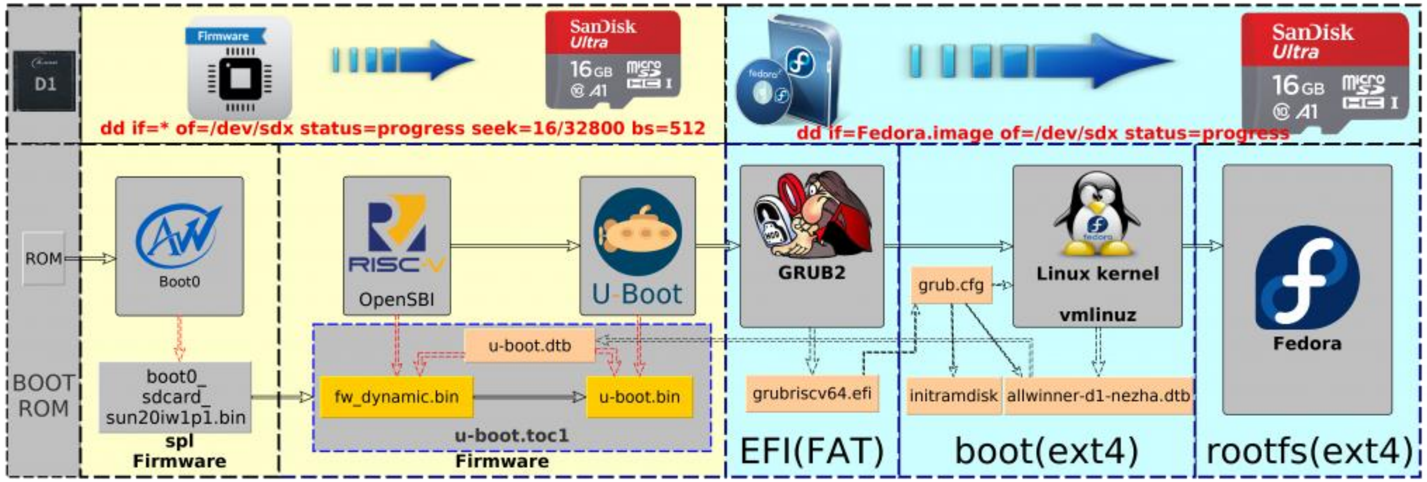
Nezha is a AIoT development board customized by AWOL based on Allwinner's D1-H chip. It is the world's first mass-produced development board that supports 64bit RISC-V instruction set and Linux system.

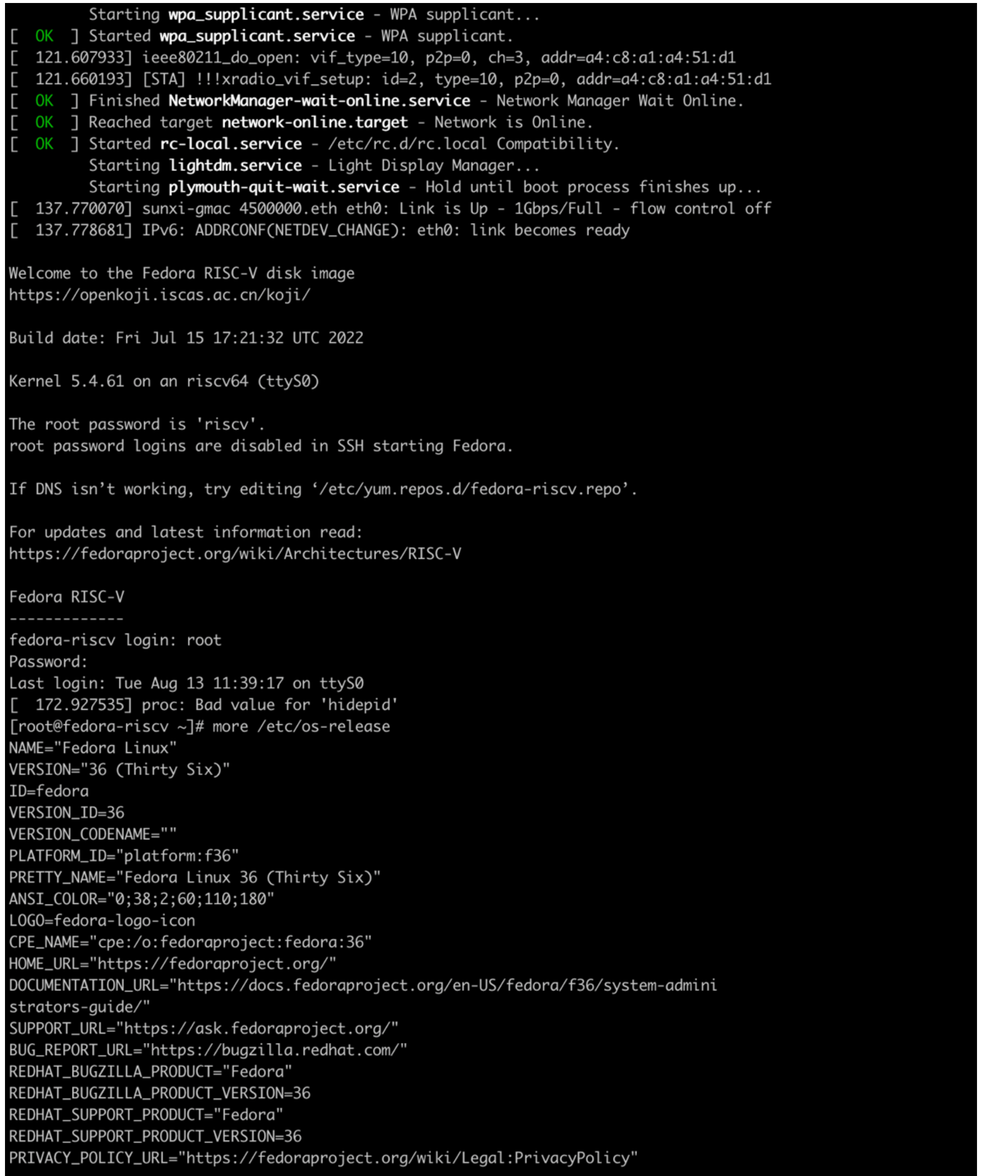


Contents [hide]



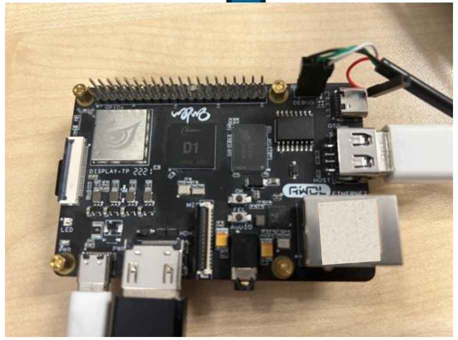
Risc-V Experiment (哪吒)

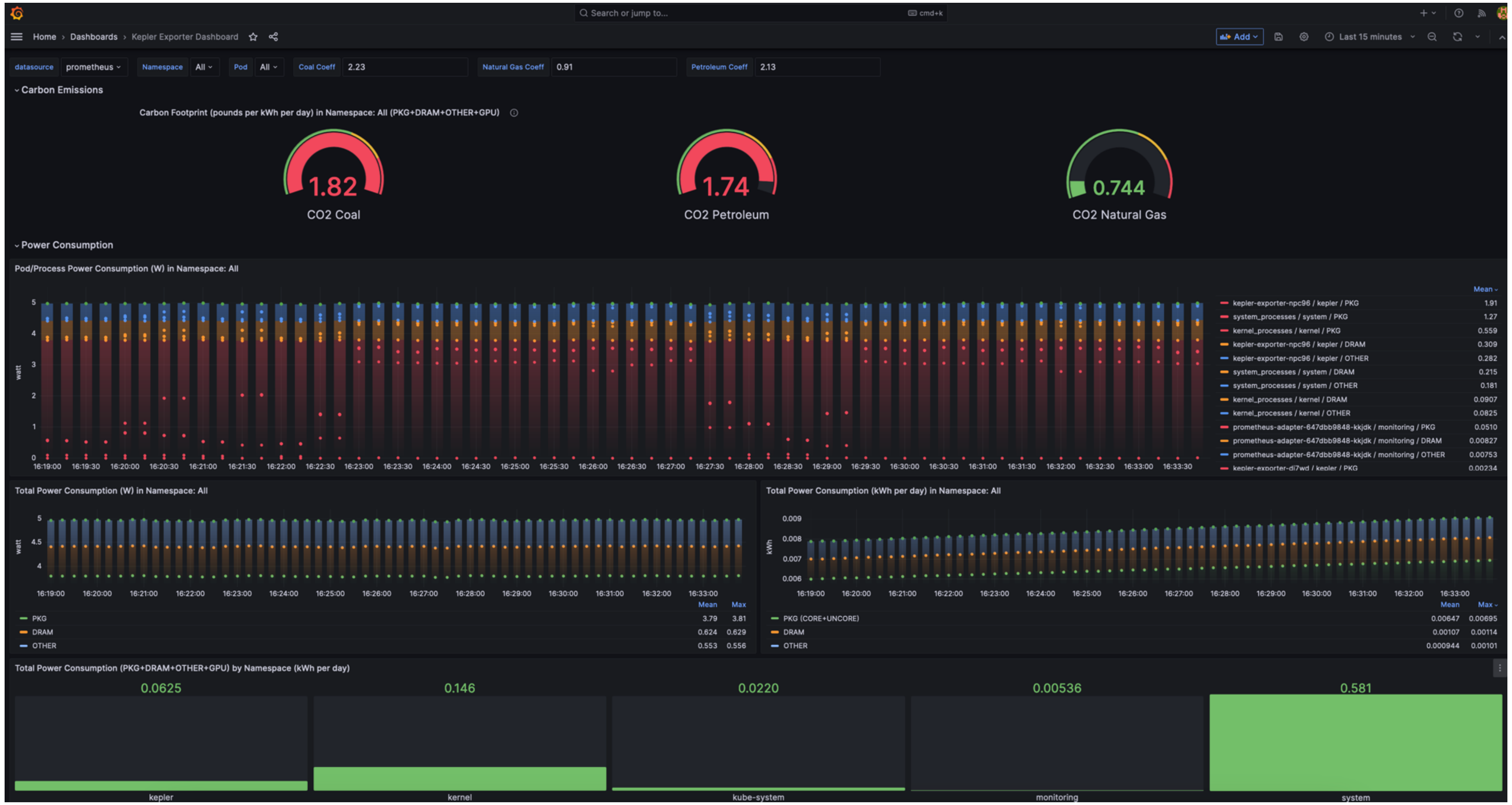




```
         Starting wpa_supplicant.service - WPA supplicant...
[  OK  ] Started wpa_supplicant.service - WPA supplicant.
[  121.607933] ieee80211_do_open: vif_type=10, p2p=0, ch=3, addr=a4:c8:a1:a4:51:d1
[  121.660193] [STA] !!!xradio_vif_setup: id=2, type=10, p2p=0, addr=a4:c8:a1:a4:51:d1
[  OK  ] Finished NetworkManager-wait-online.service - Network Manager Wait Online.
[  OK  ] Reached target network-online.target - Network is Online.
[  OK  ] Started rc-local.service - /etc/rc.d/rc.local Compatibility.
         Starting lightdm.service - Light Display Manager...
         Starting plymouth-quit-wait.service - Hold until boot process finishes up...
[  137.770070] sunxi-gmac 4500000.eth eth0: Link is Up - 1Gbps/Full - flow control off
[  137.778681] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

Welcome to the Fedora RISC-V disk image
https://openkoji.iscas.ac.cn/koji/

Build date: Fri Jul 15 17:21:32 UTC 2022

Kernel 5.4.61 on an riscv64 (ttyS0)

The root password is 'riscv'.
root password logins are disabled in SSH starting Fedora.

If DNS isn't working, try editing '/etc/yum.repos.d/fedora-riscv.repo'.

For updates and latest information read:
https://fedoraproject.org/wiki/Architectures/RISC-V

Fedora RISC-V
-------------
fedora-riscv login: root
Password:
Last login: Tue Aug 13 11:39:17 on ttyS0
[  172.927535] proc: Bad value for 'hidepid'
[root@fedora-riscv ~]# more /etc/os-release
NAME="Fedora Linux"
VERSION="36 (Thirty Six)"
ID=fedora
VERSION_ID=36
VERSION_CODENAME=""
PLATFORM_ID="platform:f36"
PRETTY_NAME="Fedora Linux 36 (Thirty Six)"
ANSI_COLOR="0;38;2;60;110;180"
LOGO=fedora-logo-icon
CPE_NAME="cpe:/o:fedoraproject:fedora:36"
HOME_URL="https://fedoraproject.org/"
DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f36/system-admini
strators-guide/"
SUPPORT_URL="https://ask.fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=36
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=36
PRIVACY_POLICY_URL="https://fedoraproject.org/wiki/Legal:PrivacyPolicy"
```

Risc-V Experiment (哪吒)





Home › Dashboards › Kepler Exporter Dashboard

datasource	Namespace	Pod	Coal Coeff	Natural Gas Coeff	Petroleum Coeff
prometheus	All	All	2.23	0.91	2.13

Carbon Emissions

Carbon Footprint (pounds per kWh per day) in Namespace: All (PKG+DRAM+OTHER+GPU)

CO2 Coal	CO2 Petroleum	CO2 Natural Gas
1.82	1.74	0.744

Power Consumption

Pod/Process Power Consumption (W) in Namespace: All

	Mean
kepler-exporter-npc96 / kepler / PKG	1.91
system_processes / system / PKG	1.27
kernel_processes / kernel / PKG	0.559
kepler-exporter-npc96 / kepler / DRAM	0.309
kepler-exporter-npc96 / kepler / OTHER	0.282
system_processes / system / DRAM	0.215
system_processes / system / OTHER	0.181
kernel_processes / kernel / DRAM	0.0907
kernel_processes / kernel / OTHER	0.0825
prometheus-adapter-647dbb9848-kkjdk / monitoring / PKG	0.0510
prometheus-adapter-647dbb9848-kkjdk / monitoring / DRAM	0.00827
prometheus-adapter-647dbb9848-kkjdk / monitoring / OTHER	0.00753
kepler-exporter-di7wd / kepler / PKG	0.00234

Total Power Consumption (W) in Namespace: All

	Mean	Max
PKG	3.79	3.81
DRAM	0.624	0.629
OTHER	0.553	0.556

Total Power Consumption (kWh per day) in Namespace: All

	Mean	Max
PKG (CORE+UNCORE)	0.00647	0.00695
DRAM	0.00107	0.00114
OTHER	0.000944	0.00101

Total Power Consumption (PKG+DRAM+OTHER+GPU) by Namespace (kWh per day)

kepler	kernel	kube-system	monitoring	system
0.0625	0.146	0.0220	0.00536	0.581

Kepler Dashboard

Benefits and Impact

- **Energy Efficiency**
  - **Optimized Energy Management:**
    The integration of Kepler and Open Horizon results in more precise energy management, leading to reduced energy wastage and lower operational costs.
  - **Environmental Impact:**
    Improved energy efficiency contributes to a lower carbon footprint, aligning with global sustainability goals.

Sustainability Goals

- **Contribution to a Greener Paradigm**
  - **Greener Cloud-Edge Computing:**
    Our approach supports a greener cloud-edge computing paradigm by enhancing energy efficiency and system resilience.
  - **Long-term Impact:**
    This initiative not only addresses current challenges but also sets the stage for sustainable technology solutions that can adapt to future demands.

Conclusion

- **Recap of Key Points**
  - **Summary of Key Points:**
    We explored how integrating Kepler and Open Horizon with CNCF and LF Edge supports sustainability in cloud-edge computing, especially with the RISC-V architecture.
  - **Key Takeaways:**
    We highlighted the benefits of energy efficiency, practical use cases, and the impact on the Chinese market.

Call to Action

- **Encouragement**
  - **Explore Technologies:**
    We encourage you to explore the integration of these technologies in your own systems to enhance energy management and sustainability.
  - **Contact Information:**
    For more information or further discussion, please feel free to reach out to us.

KEPLER

https://www.cncf.io/projects/kepler/

OPEN HORIZON

https://www.lfedge.org/projects/openhorizon/

Call to Action

CNCF环境可持续发展技术咨询小组简介

CNCF TAG ENV是一个技术咨询小组，由充满激情的个人推动，他们致力于制定指导方针、工具和实践，将可持续性融入云原生技术。其主要目的是分享减少技术基础设施对环境影响的工具和最佳实践，提出创新战略和实用解决方案。TAG ENV定期组织会议，包括推广2024年云原生可持续发展周等活动。TAG ENV是一个开放的团体，任何希望为该团体及其活动做出贡献和积极参与的人都可以这样做。该团体专注于透明度、包容性和贡献者之间的积极合作，激发创新，为更可持续的云原生和开源空间做出贡献创造更多机会。

您可以通过一下方式参与CNCF可持续技术指导小组的活动：

- 加入slack channel #tag-environmental-sustainability
- 加入可持续技术指导小组定期会 meetings对于亚太地区的朋友，您可以加入每双周周二，北京时间晚8:30（12PM GMT）召开的APAC定期会，下次会议时间定在7月23号： **Zoom:** Join Meeting **Passcode:** 77777
- 在github提交您的议题 GitHub issues
- 或参考这篇博客文章 深入探讨了如何开始为TAG ENV做出贡献

欢迎新的贡献者，我们很乐意让您加入我们！每一项贡献，无论大小，都可以在实现更环保、更可持续的技术未来方面发挥作用。

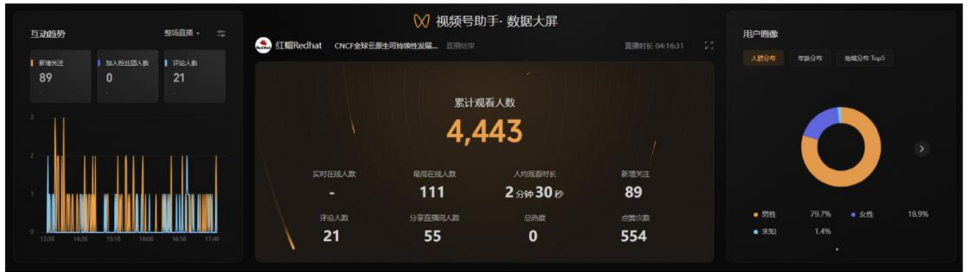
Call to action

CNCF云原生可持续发展周2024**招募**赞助商和志愿者

2023年国内CNCF云原生可持续发展周，我们以Kepler项目为例，**普及了可持**续计算的概念以及可持续计算在运维优化等软件生命周期中所扮演的角色。**我**们在本地去得了4000+**的在**线收看人数，在国际社区引发了广泛影响。借**此**，**国**际社区开始注重中国和亚太在可持续计算方面的影响，并设置了亚太时区的定期会。

2024年国内CNCF云原生可持续发展周，我们希望更多的进行国内外技术交流**和合作**。

- 为**国内**优质的项目/**案例提供CNCF云原生可持**续发展周2024这个国际舞台，让国外的同行听到我们的声音。
- **将国外**优秀的案例带到国内的会场，促进开源技术交流合作。


视频号助手· 数据大屏

互动趋势 整场直播

新增关注 89
加入粉丝团人数 0
评论人数 21

红帽Redhat CNCF全球云原生可持续性发展... 直播结束 直播时长 04:16:31

累计观看人数 4,443

实时在线人数 -
最高在线人数 111
人均观看时长 2分钟30秒
新增关注 89
评论人数 21
分享直播间人数 55
总热度 0
点赞次数 554

用户画像

男性 79.7%
女性 18.9%
未知 1.4%

Q&A

- **Invitation for Questions**
  - **Open Discussion:**
    We now open the floor for questions. Feel free to ask about the integration process, use cases, or any other related topics.
  - **Thank You:**
    Thank you for your attention and participation. We look forward to your questions and feedback.