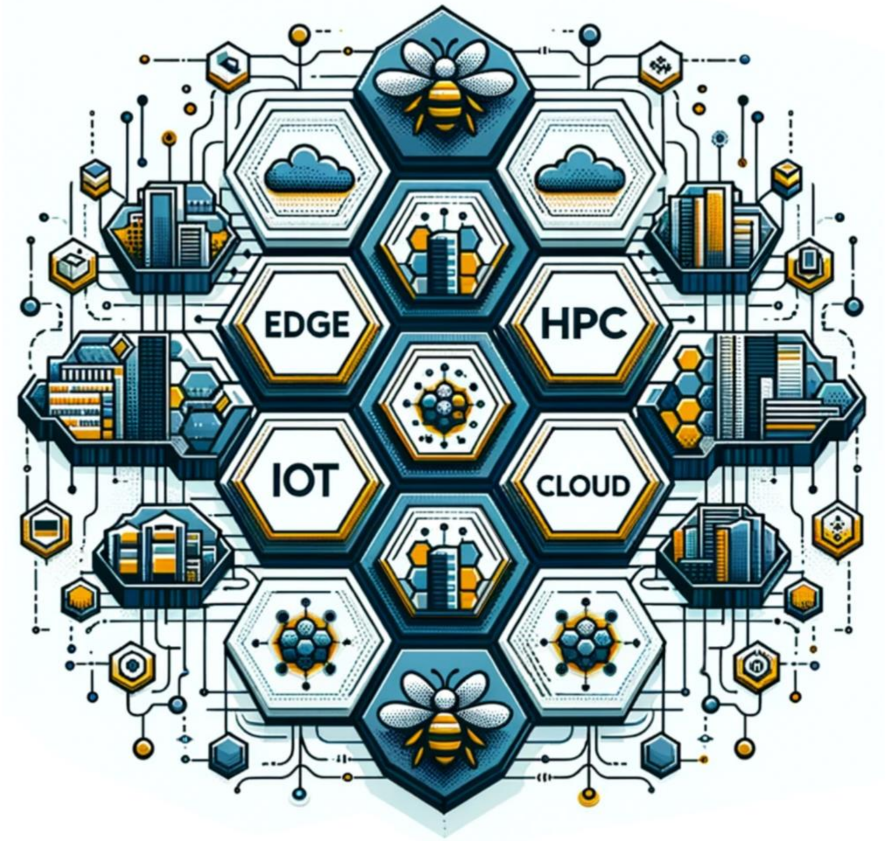
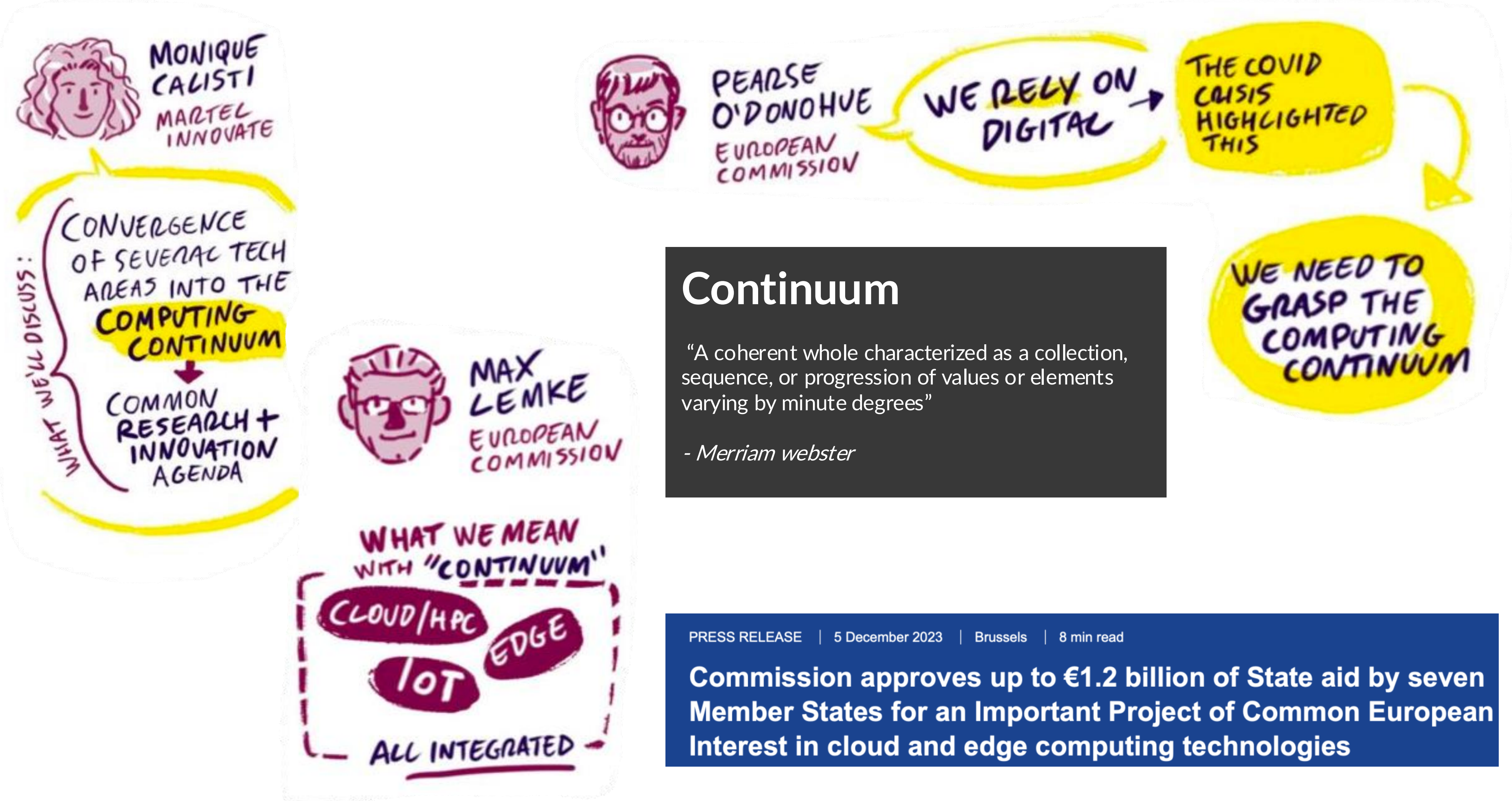


Distributed Quantum Computing

Erik Källman, RISE

QAS 2024


MONIQUE CALISTI
MARTEL INNOVATE
WHAT WE'LL DISCUSS:
CONVERGENCE OF SEVERAL TECH AREAS INTO THE COMPUTING CONTINUUM
COMMON RESEARCH + INNOVATION AGENDA
MAX LEMKE
EUROPEAN COMMISSION
WHAT WE MEAN WITH "CONTINUUM"
CLOUD/HPC
EDGE
IOT
ALL INTEGRATED
PEARSE O'DONOHUE
EUROPEAN COMMISSION
WE RELY ON DIGITAL
THE COVID CRISIS HIGHLIGHTED THIS
WE NEED TO GRASP THE COMPUTING CONTINUUM

Continuum

“A coherent whole characterized as a collection, sequence, or progression of values or elements varying by minute degrees”

- Merriam webster

PRESS RELEASE | 5 December 2023 | Brussels | 8 min read

Commission approves up to €1.2 billion of State aid by seven Member States for an Important Project of Common European Interest in cloud and edge computing technologies

Source: Executive Summary - Digital Autonomy in the Computing Continuum

Challenges Compute Infrastructure

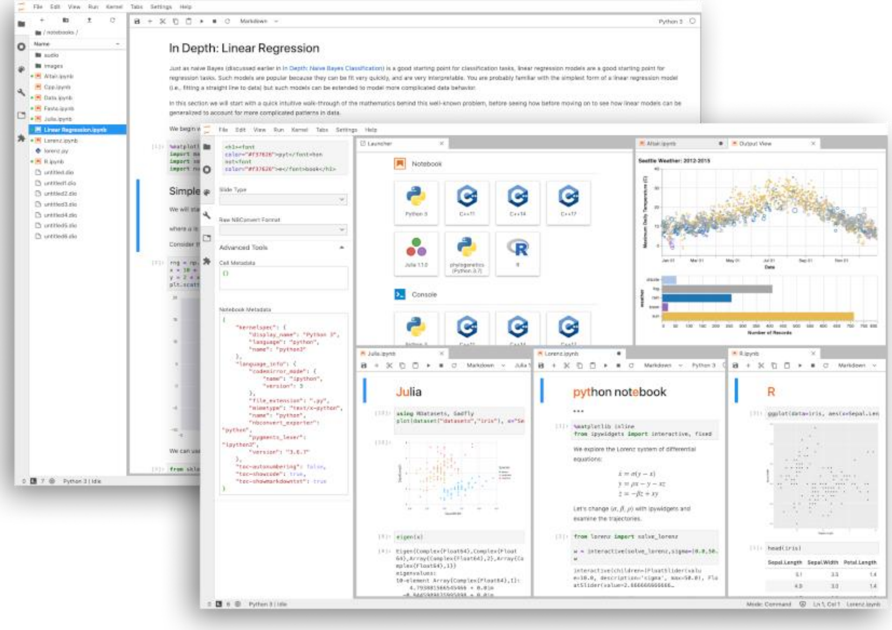
From a developer perspective …

- **User experience**
  - Complex login process: SSH to a login node
  - Setting up tunnels
  - Mastering Slurm jobs
  - When will my job run? Will someone kill my job?
- **Data management**
  - Determining data storage locations
  - Manual data transfers can be time-consuming and error-prone
- **Integration issues**
  - Connecting HPC systems with cloud to streamline workflows?
  - No APIs? Lack of automation tools (GitOps/CI/CD)
  - Multi-factor authentication
  - Sometimes no Internet access on compute nodes

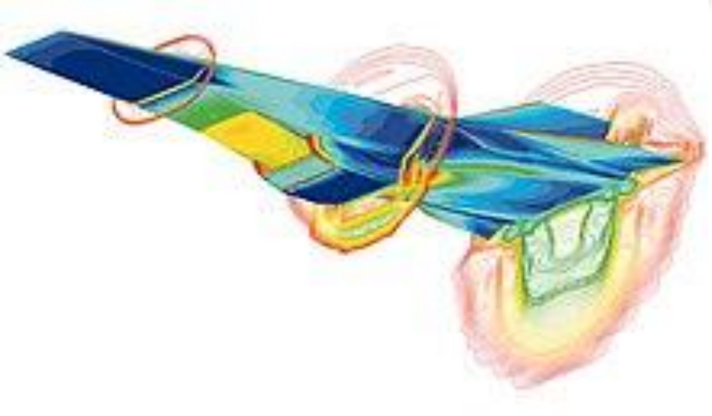
Generated by ChatGPT

High-Performance Computing

- **Scientific / Research workflows**
  - **Manual interaction**: Required to set up simulations or experiments
  - **Valuable outcomes over efficiency**: Quickly obtain accurate and valuable research results
  - **Exploratory**: Research workflows can be less predictable and require more hands-on adjustments
  - **Batch processing**: Requiring manual scripting and queue management



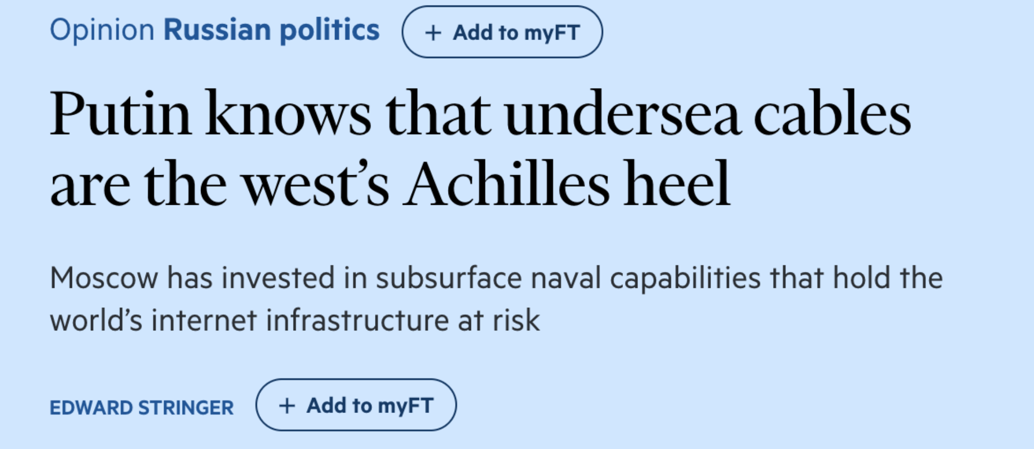
- **Why not use cloud platforms?**
  - Cloud platforms can be very complex and cumbersome to use for researchers
  - Cloud platforms like Kubernetes are not designed for HPC workloads *(not optimized for performance)*



Problems with Cloud Computing

- Dependency on network access
- Vendor lock-in
- Compliance and Regulations
- Security and privacy concerns
- Digital sovereignty

Sweden
Norway
Finland
Gulf of Bothnia
Estonia
Baltic Sea
Latvia


Opinion **Russian politics** + Add to myFT

Putin knows that undersea cables are the west's Achilles heel

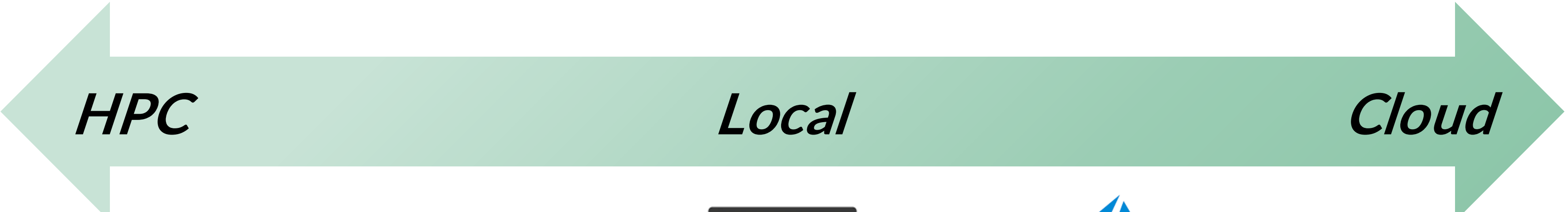
Moscow has invested in subsurface naval capabilities that hold the world's internet infrastructure at risk

EDWARD STRINGER + Add to myFT

Ideal for **scientific workflows, large-scale simulations, complex engineering computations**, and tasks requiring extensive computational power and high data throughput

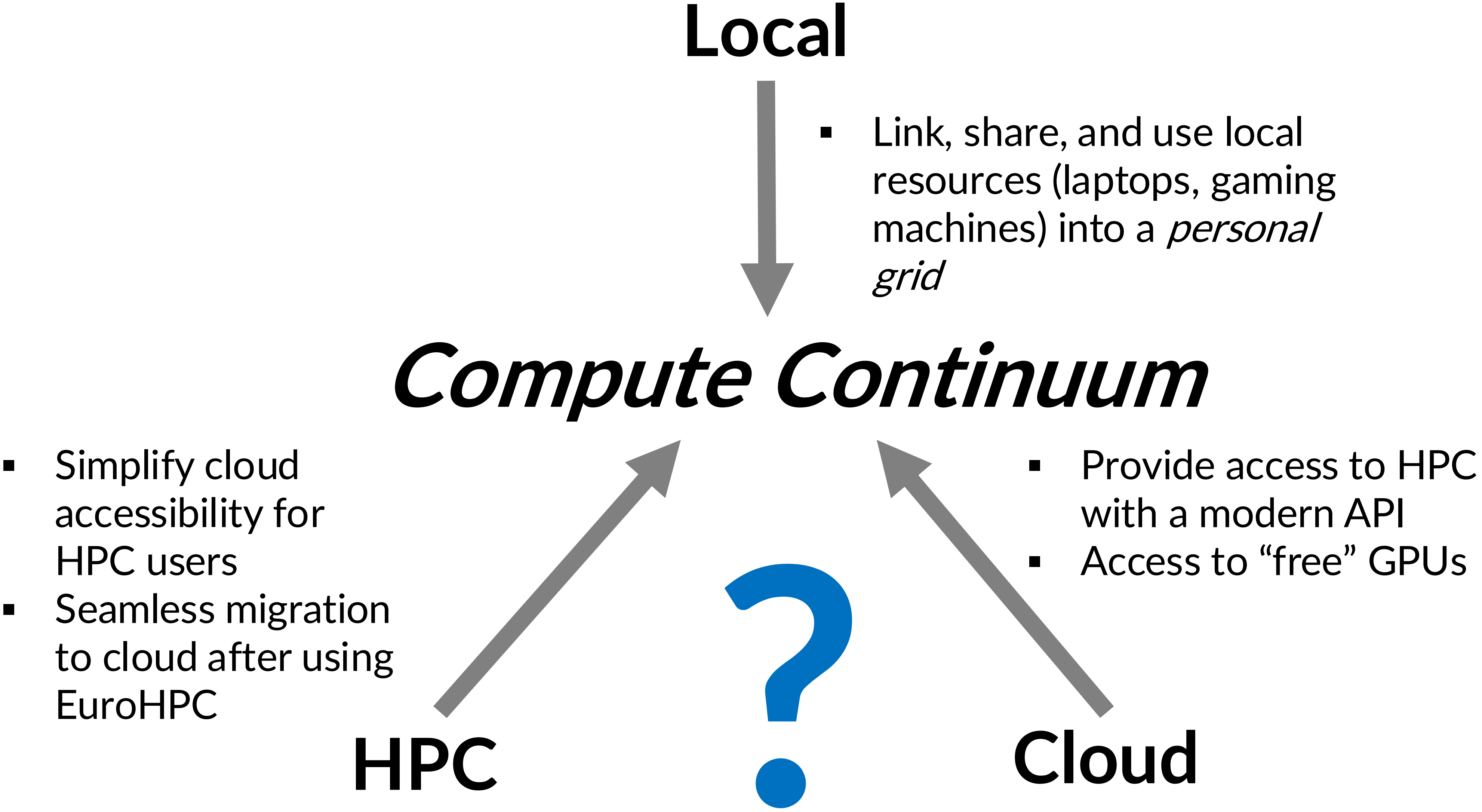
Ideal for **development, testing, and small-scale experimentation**. Suited for prototyping, debugging, and tasks that require immediate, hands-on access to computational resources

Ideal for **data storage, big data processing, machine learning, and production environments**. Optimized for scalable, distributed web services, and cost-effective resource management across global infrastructures



Azure

aws



Local

- Link, share, and use local resources (laptops, gaming machines) into a *personal grid*

Compute Continuum

- Simplify cloud accessibility for HPC users
- Seamless migration to cloud after using EuroHPC

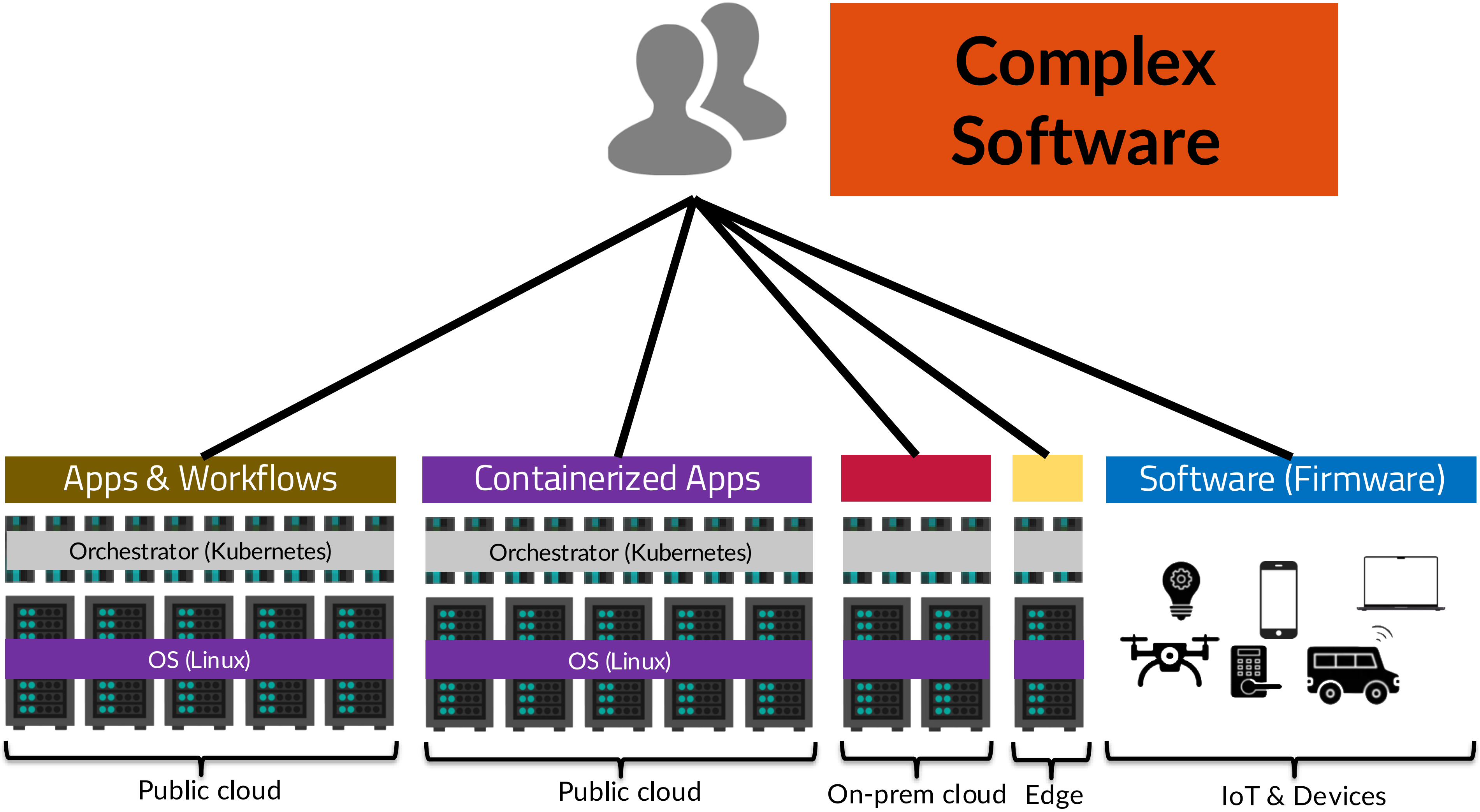
- Provide access to HPC with a modern API
- Access to “free” GPUs

HPC

?

Cloud

Complex Software



Apps & Workflows

Orchestrator (Kubernetes)

OS (Linux)

Public cloud

Containerized Apps

Orchestrator (Kubernetes)

OS (Linux)

Public cloud

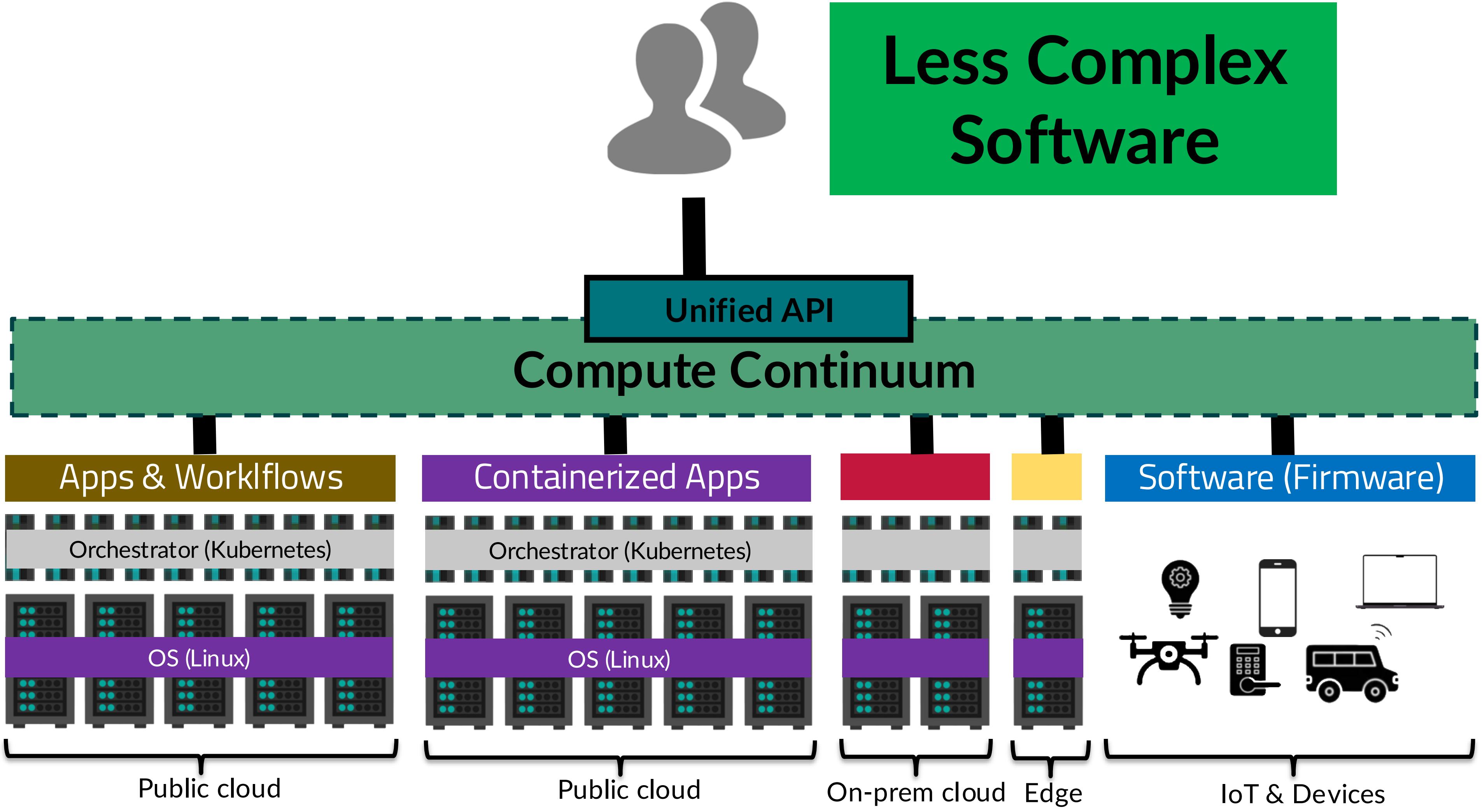
On-prem cloud

Edge

Software (Firmware)

IoT & Devices

Less Complex Software



Unified API

Compute Continuum

Apps & Worklflows

Orchestrator (Kubernetes)

OS (Linux)

Public cloud

Containerized Apps

Orchestrator (Kubernetes)

OS (Linux)

Public cloud

On-prem cloud

Edge

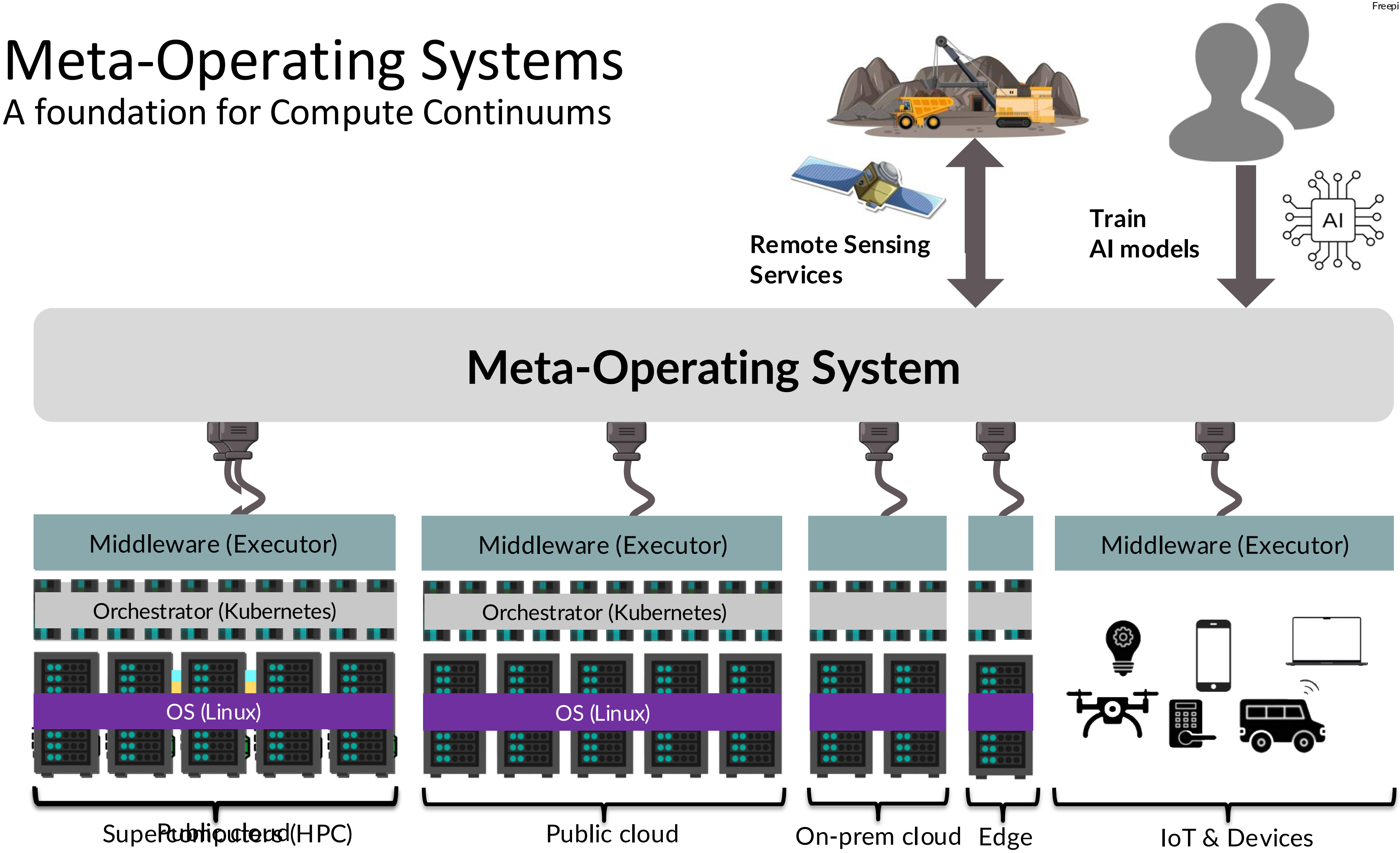
Software (Firmware)

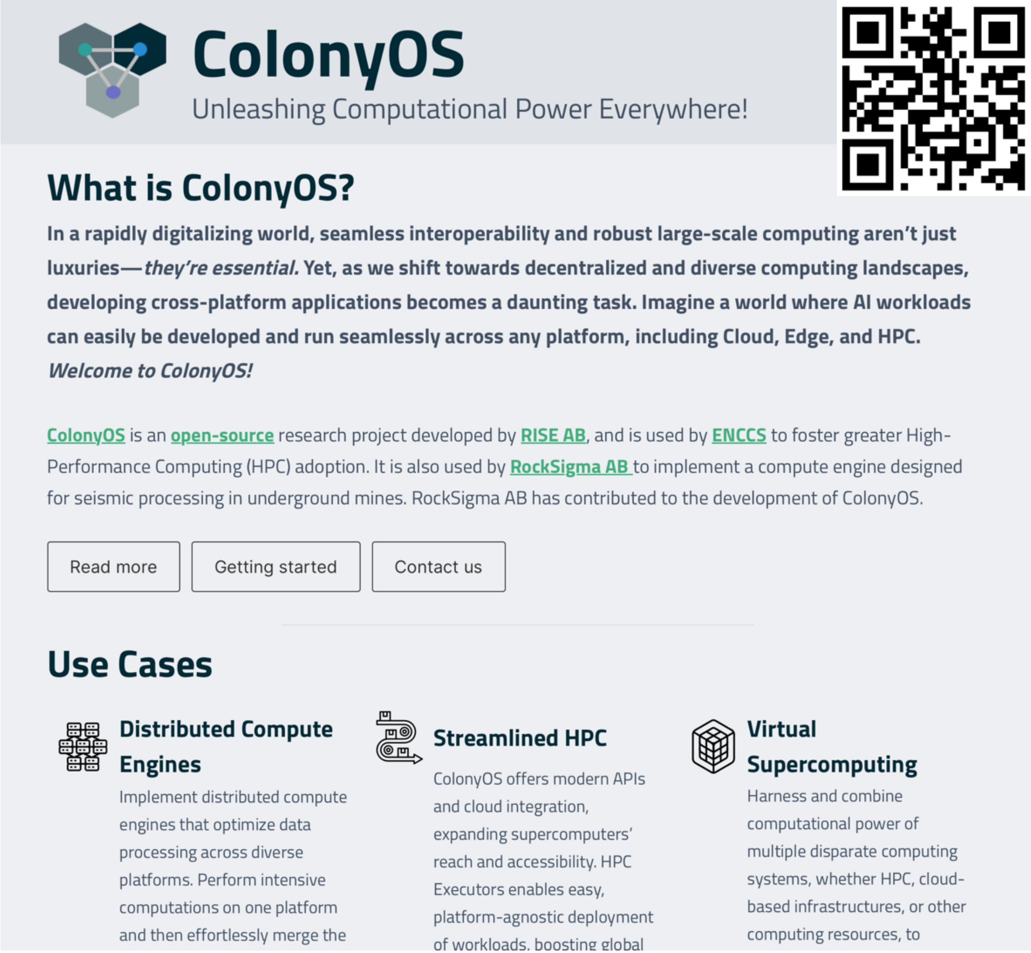
IoT & Devices

Images downloaded from Freepik and Vecteezy

Meta-Operating Systems

A foundation for Compute Continuums





ColonyOS

Unleashing Computational Power Everywhere!

What is ColonyOS?

In a rapidly digitalizing world, seamless interoperability and robust large-scale computing aren't just luxuries—*they're essential.* Yet, as we shift towards decentralized and diverse computing landscapes, developing cross-platform applications becomes a daunting task. Imagine a world where AI workloads can easily be developed and run seamlessly across any platform, including Cloud, Edge, and HPC.

Welcome to ColonyOS!

ColonyOS is an **open-source** research project developed by **RISE AB**, and is used by **ENCCS** to foster greater High-Performance Computing (HPC) adoption. It is also used by **RockSigma AB** to implement a compute engine designed for seismic processing in underground mines. RockSigma AB has contributed to the development of ColonyOS.

Use Cases

Distributed Compute Engines

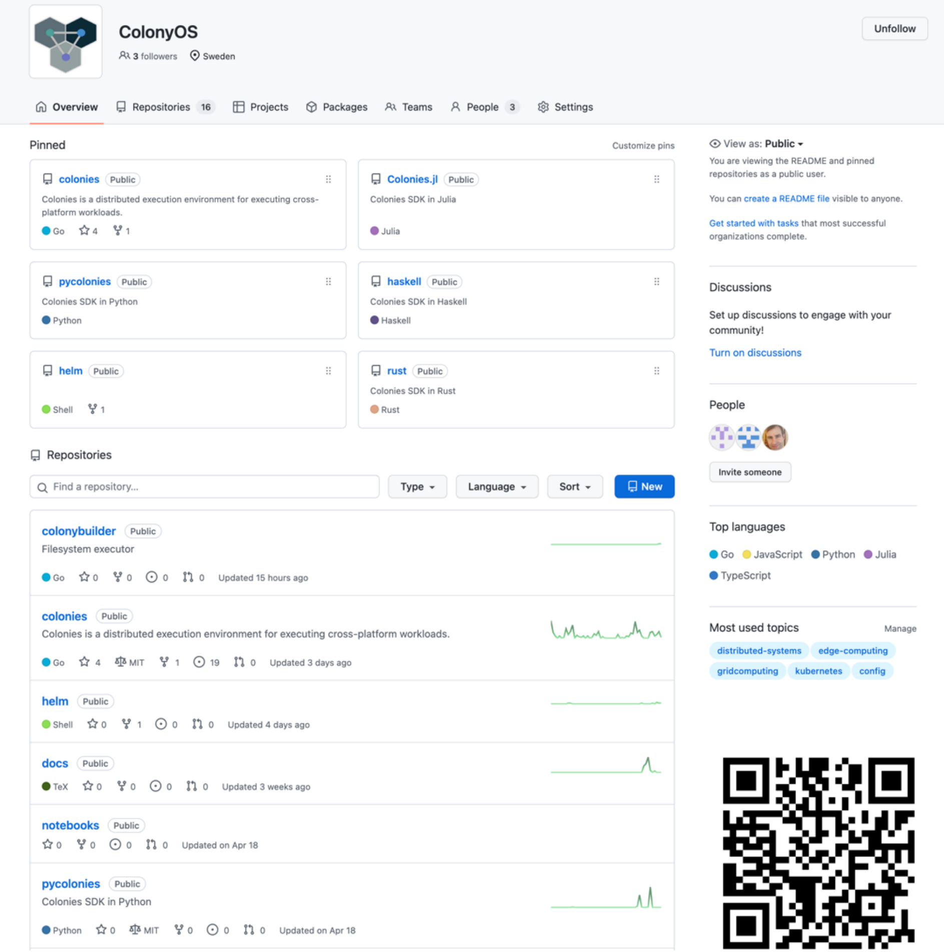
Implement distributed compute engines that optimize data processing across diverse platforms. Perform intensive computations on one platform and then effortlessly merge the

Streamlined HPC

ColonyOS offers modern APIs and cloud integration, expanding supercomputers' reach and accessibility. HPC Executors enables easy, platform-agnostic deployment of workloads, boosting global

Virtual Supercomputing

Harness and combine computational power of multiple disparate computing systems, whether HPC, cloud-based infrastructures, or other computing resources, to



ColonyOS

3 followers · Sweden

Pinned

colonies Public
Colonies is a distributed execution environment for executing cross-platform workloads.
Go ☆ 4 ⑂ 1

Colonies.jl Public
Colonies SDK in Julia
Julia

pycolonies Public
Colonies SDK in Python
Python

haskell Public
Colonies SDK in Haskell
Haskell

helm Public
Shell ⑂ 1

rust Public
Colonies SDK in Rust
Rust

Repositories

colonybuilder Public
Filesystem executor
Go ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated 15 hours ago

colonies Public
Colonies is a distributed execution environment for executing cross-platform workloads.
Go ☆ 4 MIT ⑂ 1 ⊙ 19 ⇅ 0 Updated 3 days ago

helm Public
Shell ☆ 0 ⑂ 1 ⊙ 0 ⇅ 0 Updated 4 days ago

docs Public
TeX ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated 3 weeks ago

notebooks Public
☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Apr 18

pycolonies Public
Colonies SDK in Python
Python ☆ 0 MIT ⑂ 0 ⊙ 0 ⇅ 0 Updated on Apr 18

View as: **Public**

You are viewing the README and pinned repositories as a public user.

You can create a README file visible to anyone.

Get started with tasks that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

Turn on discussions

People

Top languages

Go · JavaScript · Python · Julia · TypeScript

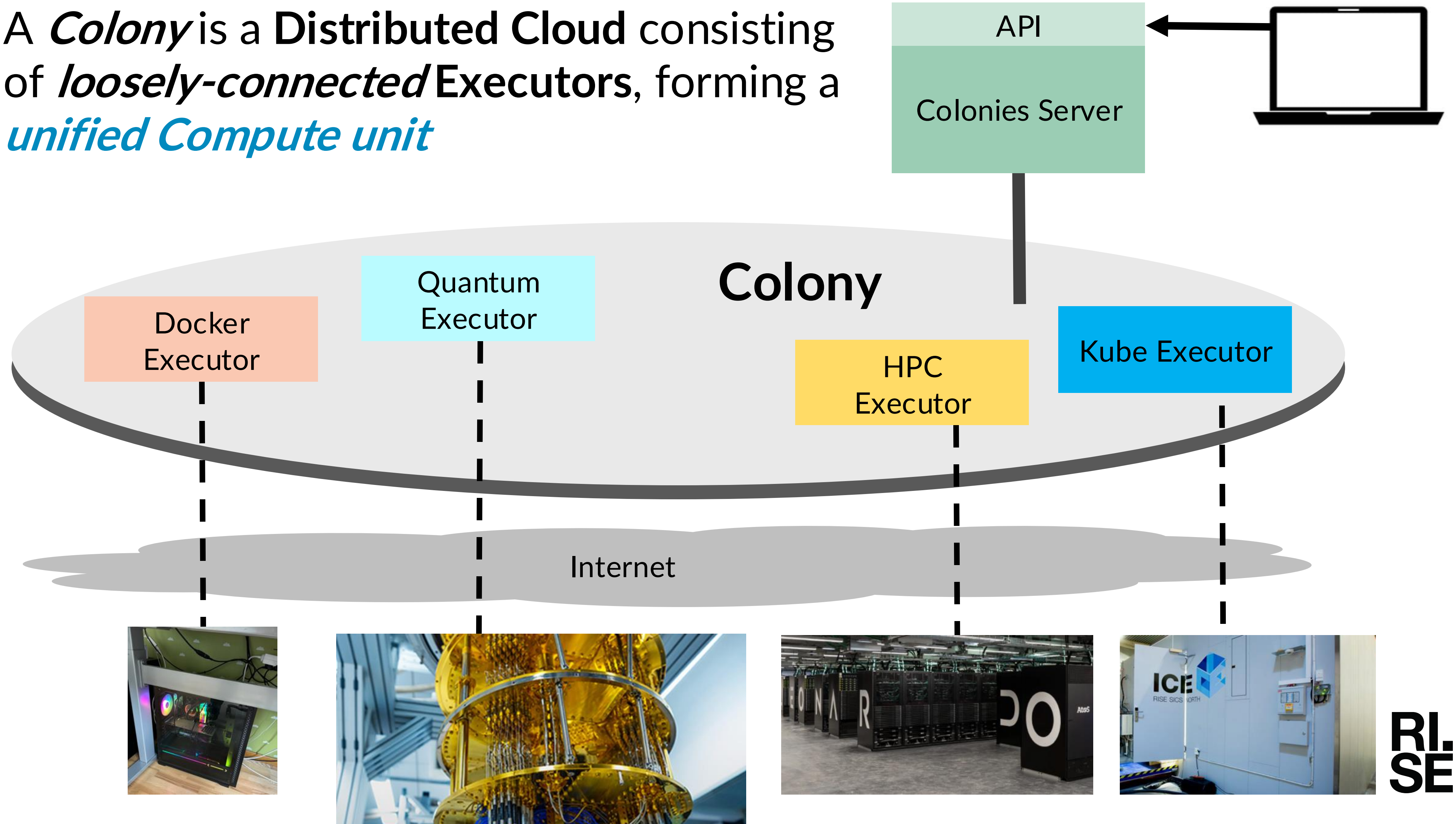
Most used topics

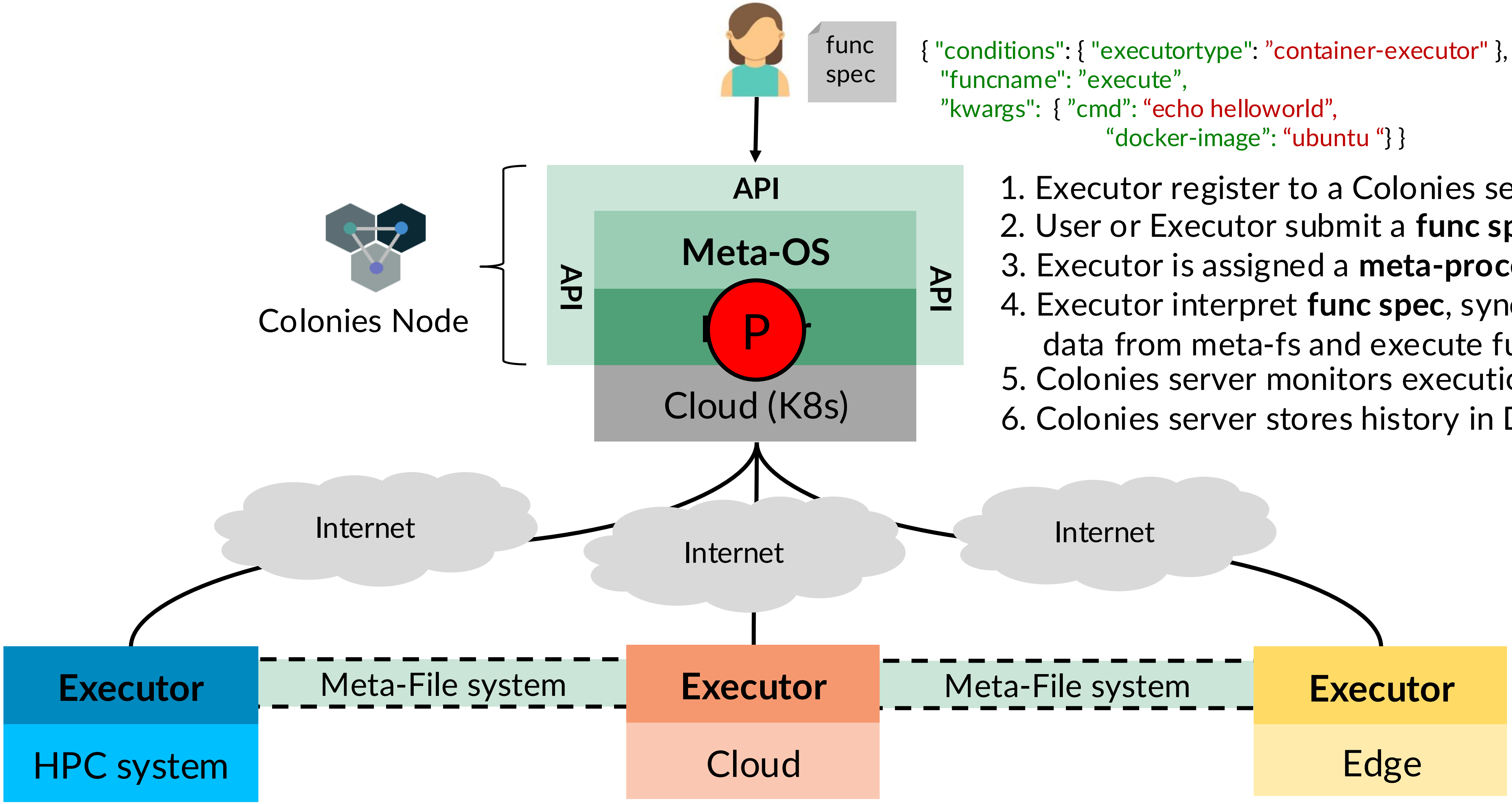
distributed-systems · edge-computing · gridcomputing · kubernetes · config

https://colonyos.io

https://github.com/colonyos

A *Colony* is a **Distributed Cloud** consisting of *loosely-connected* **Executors**, forming a ***unified Compute unit***





```
{ "conditions": { "executortype": ”container-executor" },
   "funcname": ”execute”,
   ”kwargs":  { ”cmd”: “echo helloworld”,
                          “docker-image”: “ubuntu “} }
```

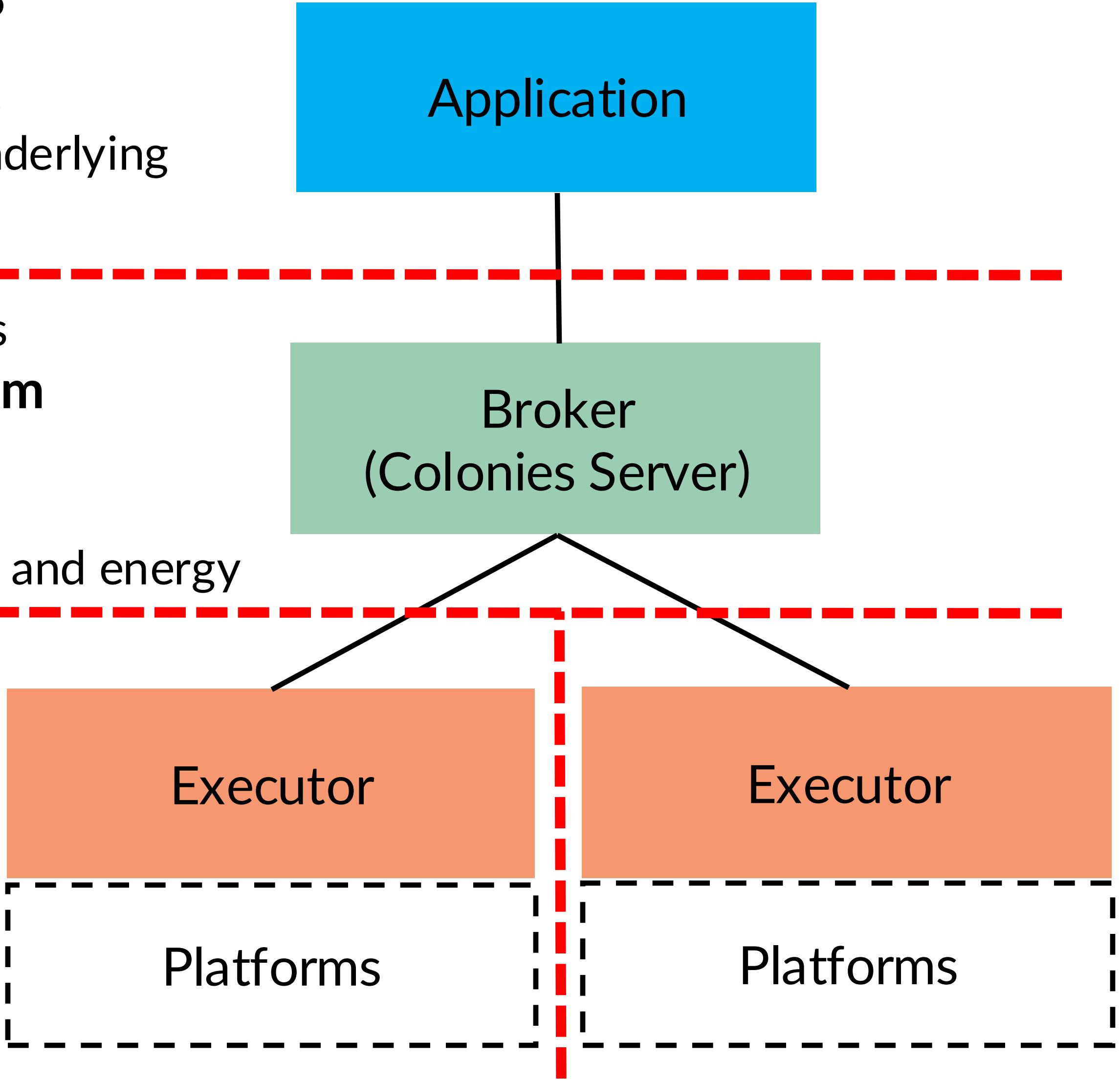
1. Executor register to a Colonies server
2. User or Executor submit a **func spec**
3. Executor is assigned a **meta-process**
4. Executor interpret **func spec**, sync data from meta-fs and execute func
5. Colonies server monitors execution
6. Colonies server stores history in DB

Separation of concerns

- Users describe meta function calls
- Do not need to understand the underlying platforms

- Abstracts away complex platforms
- Enables a **loosely coupled system**
- Ledger
- Dynamic allocation of resources
- Optimize performance, scalability, and energy

- Executors are microservices designed to execute specific functions
- Integrate with other platforms
- System integrator
- Reside anywhere on the Internet



Application

Broker (Colonies Server)

Executor

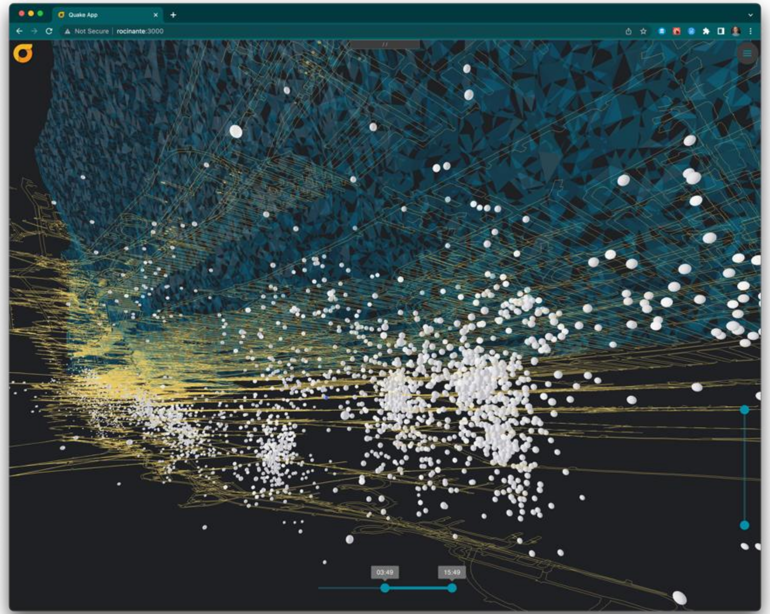
Executor

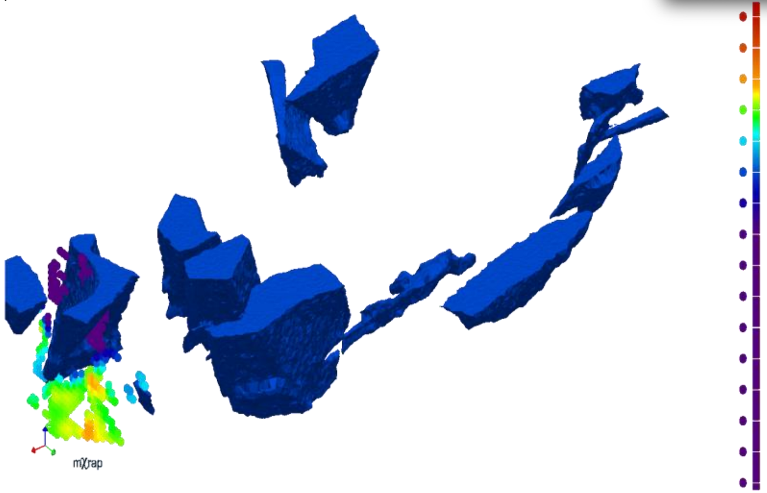
Platforms

Platforms

RockSigma AB

- Seismic processing underground mines
- Used by LKAB to analyze seismicity and process a massive amount of data from one the largest mines in the world (Kiruna/Malmberget)
- On-preem + cloud



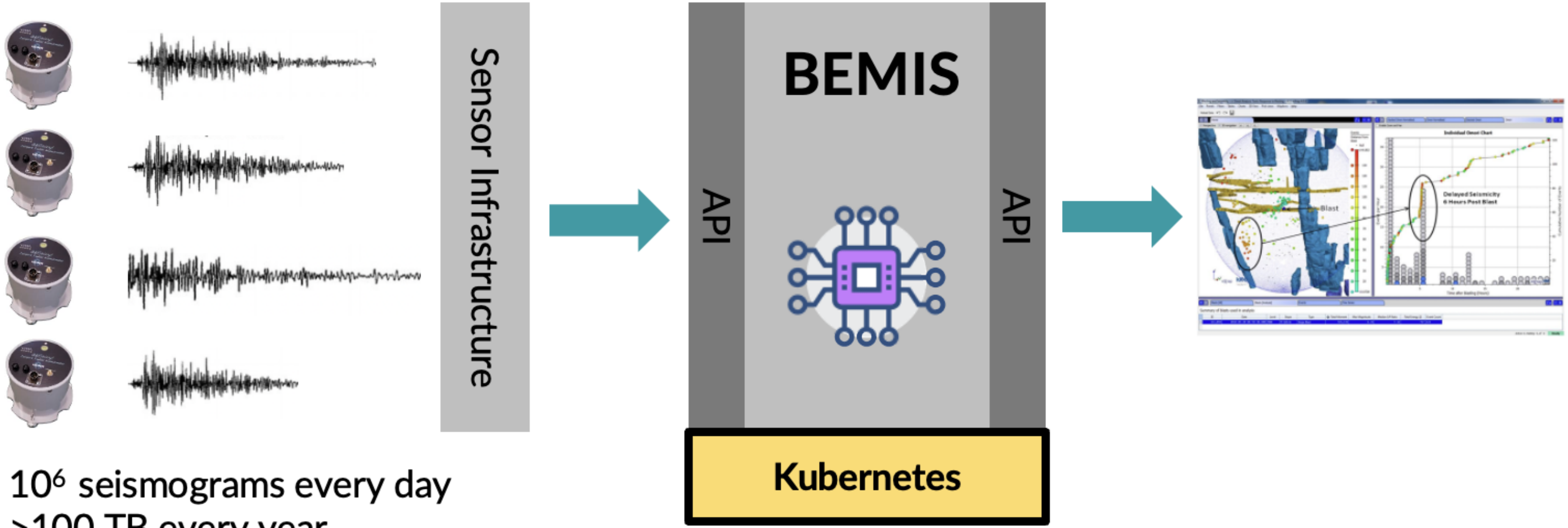


RockSigma AB

Seismograms

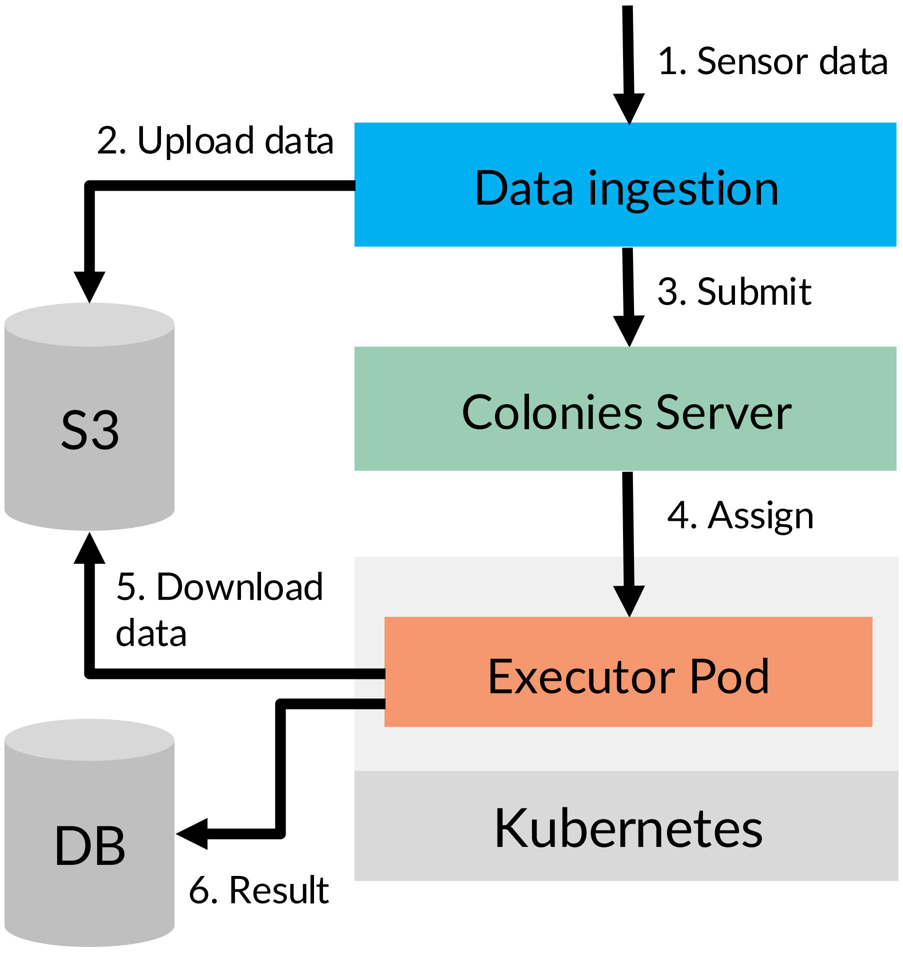
Seismic Processing Engine

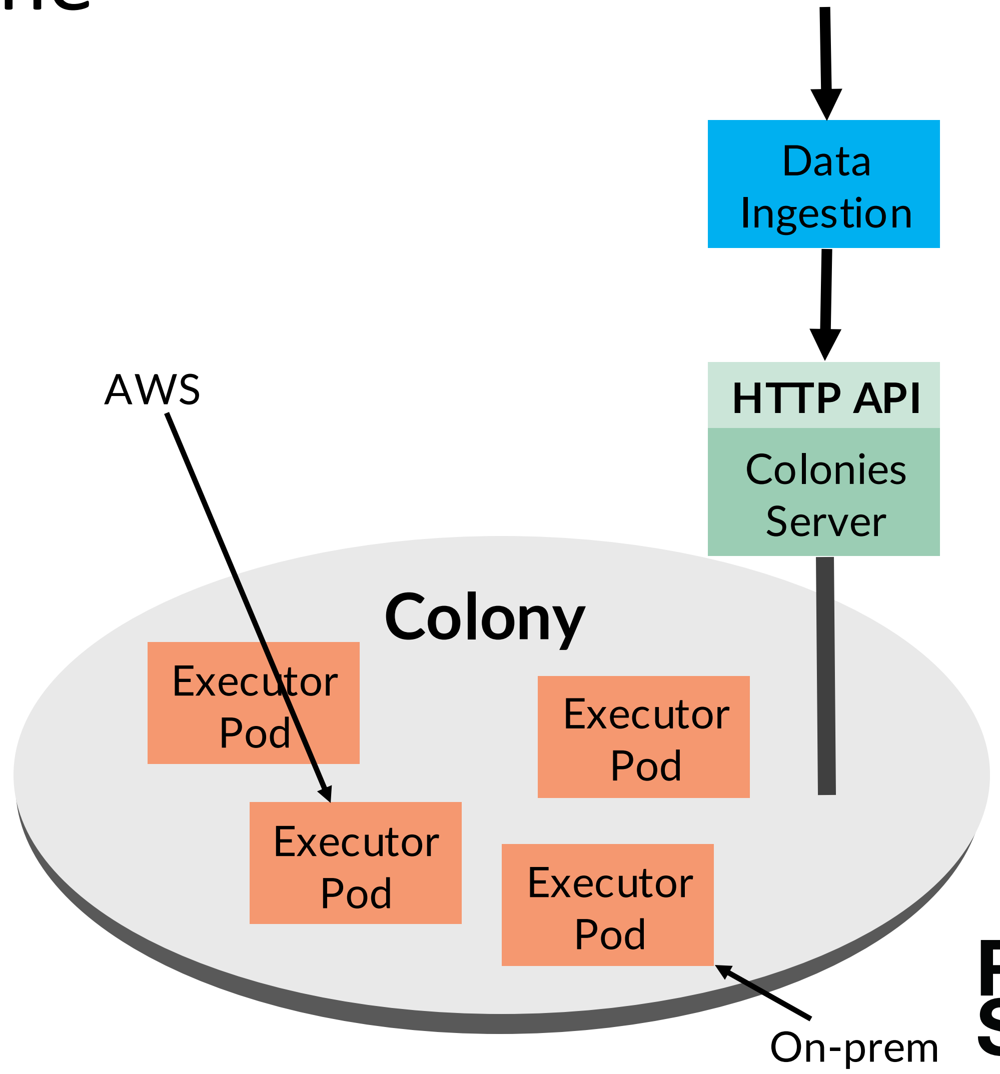
Visualization

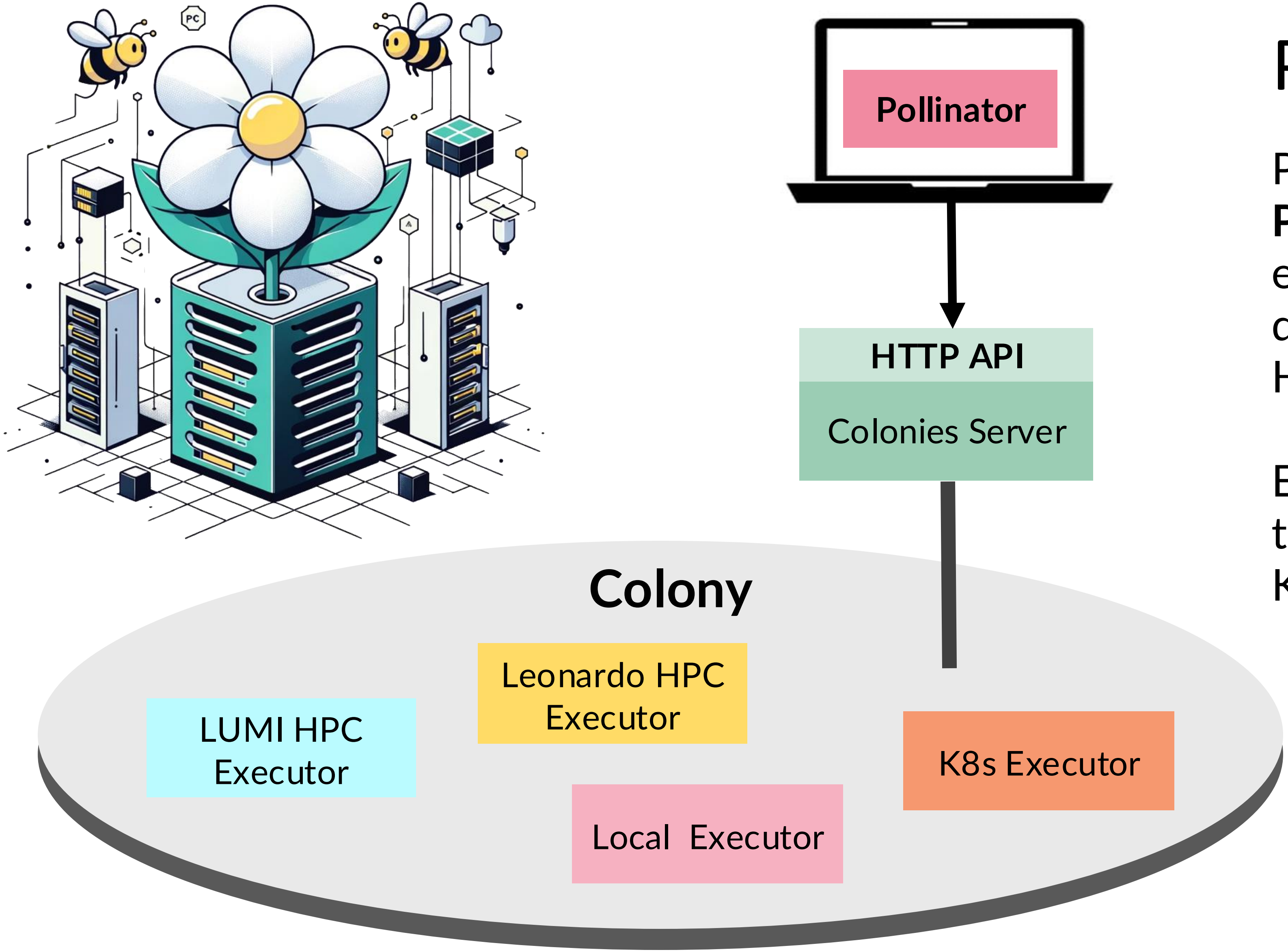


10^6 seismograms every day
>100 TB every year

A Seismic Processing Engine





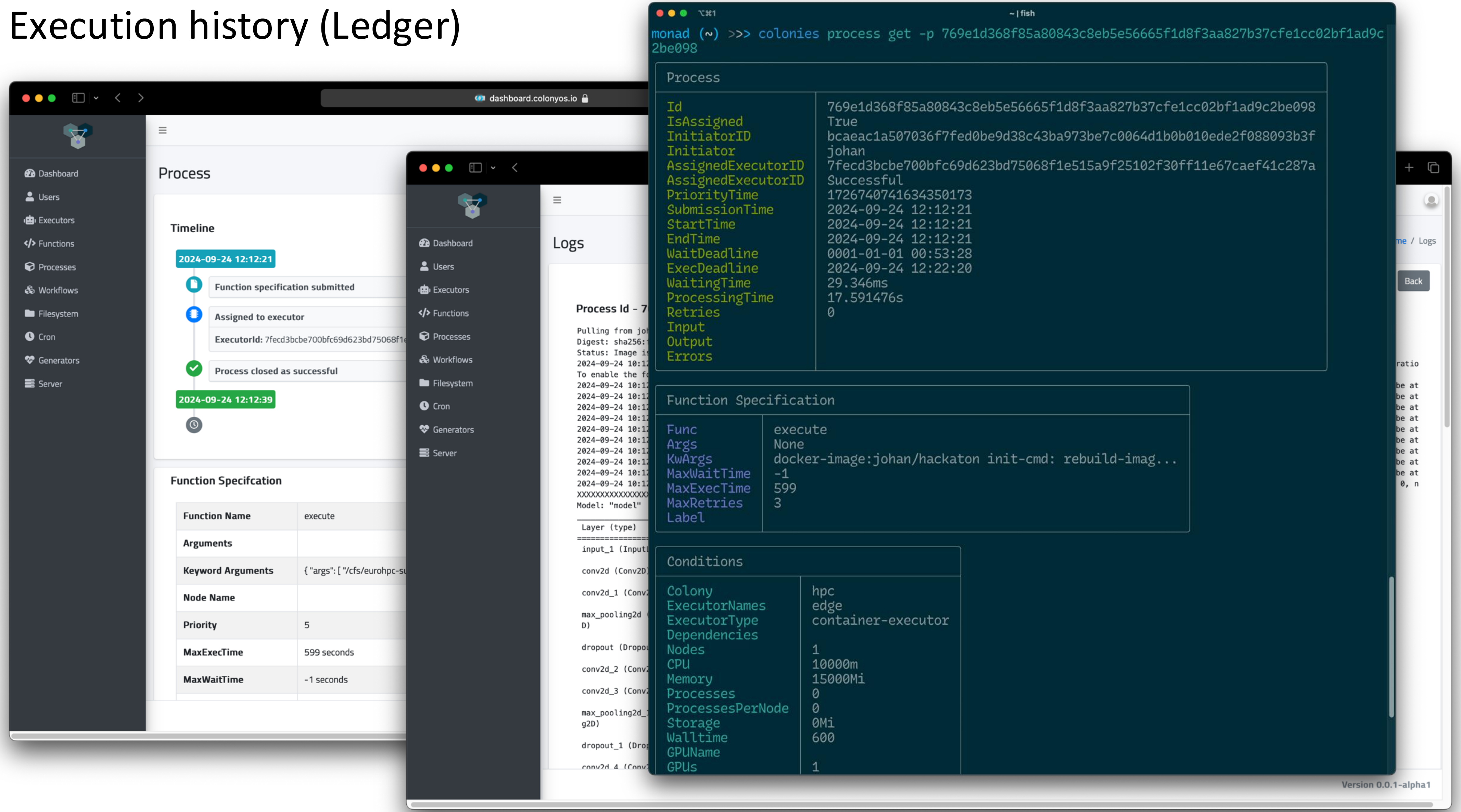


Pollinator

Pollinator provides a **PaaS alike** user experience for ML development on HPC & K8s

Eliminates the need to learn Slurm, Kubernetes

Execution history (Ledger)



Process

Timeline

2024-09-24 12:12:21

Function specification submitted

Assigned to executor

ExecutorId: 7fecd3bcbe700bfc69d623bd75068f1e

Process closed as successful

2024-09-24 12:12:39

Function Specifcation

Function Name	execute
Arguments	
Keyword Arguments	{ "args": ["/cfs/eurohpc-su
Node Name	
Priority	5
MaxExecTime	599 seconds
MaxWaitTime	-1 seconds

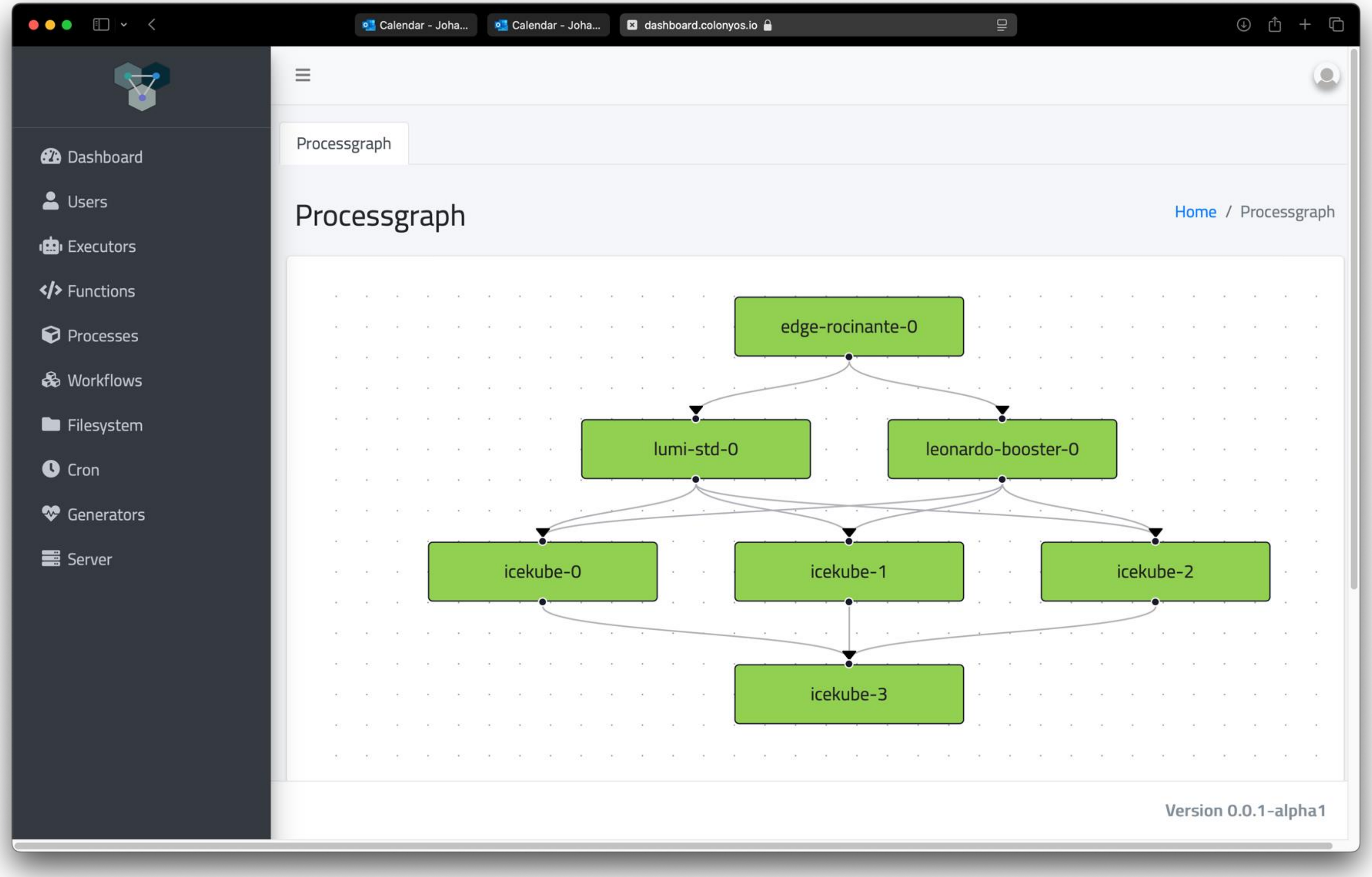
Logs

Process Id - 7

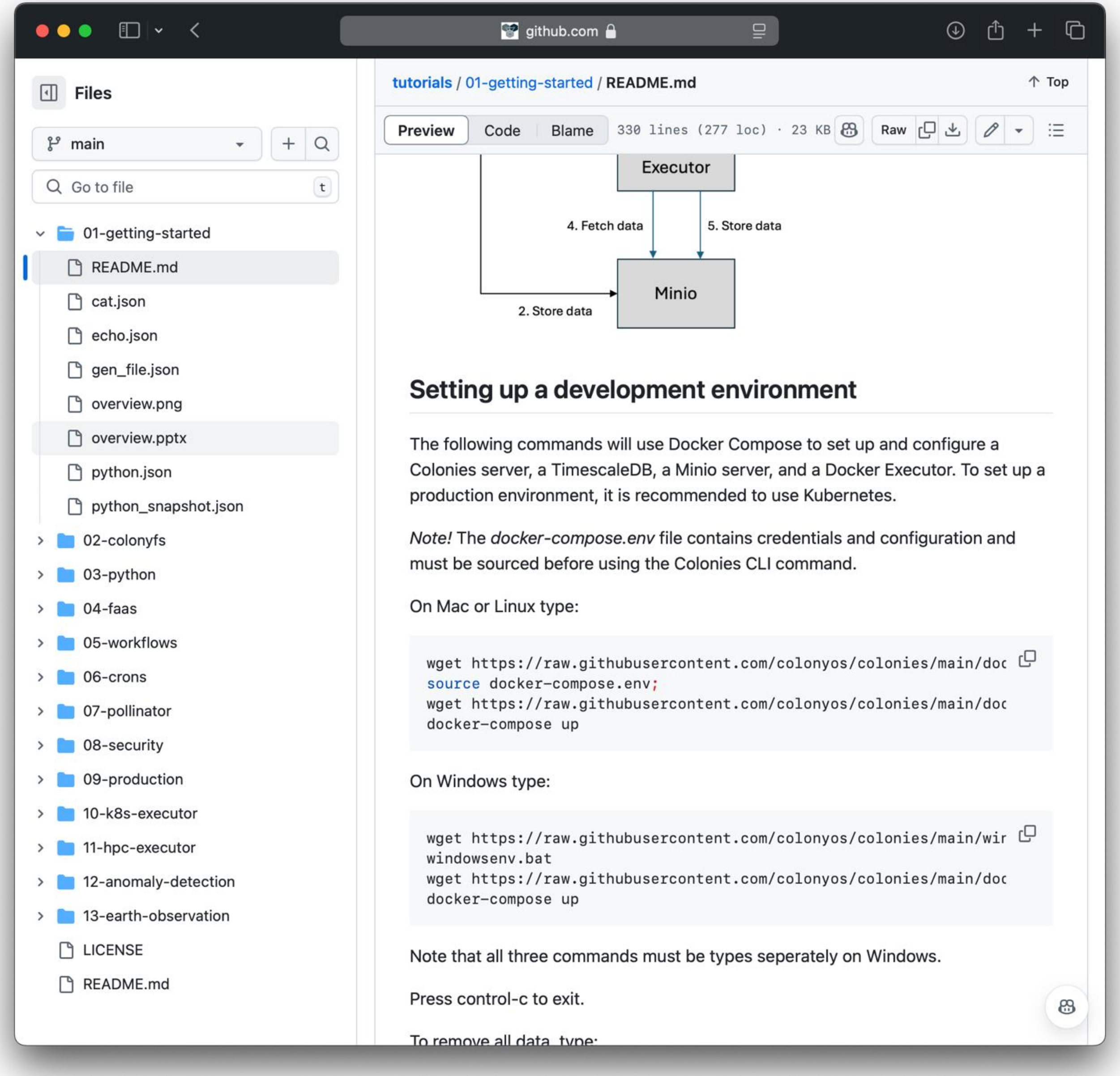
```
Pulling from joh
Digest: sha256:
Status: Image is
2024-09-24 10:12
To enable the fo
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
2024-09-24 10:12
XXXXXXXXXXXXXXX
Model: "model"
_______________
 Layer (type)
===============
 input_1 (InputL

 conv2d (Conv2D)

 conv2d_1 (Conv2

 max_pooling2d (
 D)

 dropout (Dropou

 conv2d_2 (Conv2

 conv2d_3 (Conv2

 max_pooling2d_1
 g2D)

 dropout_1 (Drop

 conv2d_4 (Conv2
```

```
~ | fish
monad (~) >>> colonies process get -p 769e1d368f85a80843c8eb5e56665f1d8f3aa827b37cfe1cc02bf1ad9c
2be098

 Process
 Id                   769e1d368f85a80843c8eb5e56665f1d8f3aa827b37cfe1cc02bf1ad9c2be098
 IsAssigned           True
 InitiatorID          bcaeac1a507036f7fed0be9d38c43ba973be7c0064d1b0b010ede2f088093b3f
 Initiator            johan
 AssignedExecutorID   7fecd3bcbe700bfc69d623bd75068f1e515a9f25102f30ff11e67caef41c287a
 AssignedExecutorID   Successful
 PriorityTime         1726740741634350173
 SubmissionTime       2024-09-24 12:12:21
 StartTime            2024-09-24 12:12:21
 EndTime              2024-09-24 12:12:21
 WaitDeadline         0001-01-01 00:53:28
 ExecDeadline         2024-09-24 12:22:20
 WaitingTime          29.346ms
 ProcessingTime       17.591476s
 Retries              0
 Input
 Output
 Errors

 Function Specification
 Func          execute
 Args          None
 KwArgs        docker-image:johan/hackaton init-cmd: rebuild-imag...
 MaxWaitTime   -1
 MaxExecTime   599
 MaxRetries    3
 Label

 Conditions
 Colony            hpc
 ExecutorNames     edge
 ExecutorType      container-executor
 Dependencies
 Nodes             1
 CPU               10000m
 Memory            15000Mi
 Processes         0
 ProcessesPerNode  0
 Storage           0Mi
 Walltime          600
 GPUName
 GPUs              1
```

Cross-platform workflows



Processgraph



Setting up a development environment

The following commands will use Docker Compose to set up and configure a Colonies server, a TimescaleDB, a Minio server, and a Docker Executor. To set up a production environment, it is recommended to use Kubernetes.

Note! The *docker-compose.env* file contains credentials and configuration and must be sourced before using the Colonies CLI command.

On Mac or Linux type:

```
wget https://raw.githubusercontent.com/colonyos/colonies/main/doc
source docker-compose.env;
wget https://raw.githubusercontent.com/colonyos/colonies/main/doc
docker-compose up
```

On Windows type:

```
wget https://raw.githubusercontent.com/colonyos/colonies/main/wir
windowsenv.bat
wget https://raw.githubusercontent.com/colonyos/colonies/main/doc
docker-compose up
```

Note that all three commands must be types seperately on Windows.

Press control-c to exit.

To remove all data, type:

https://github.com/colonyos/tutorials

RI.
SE