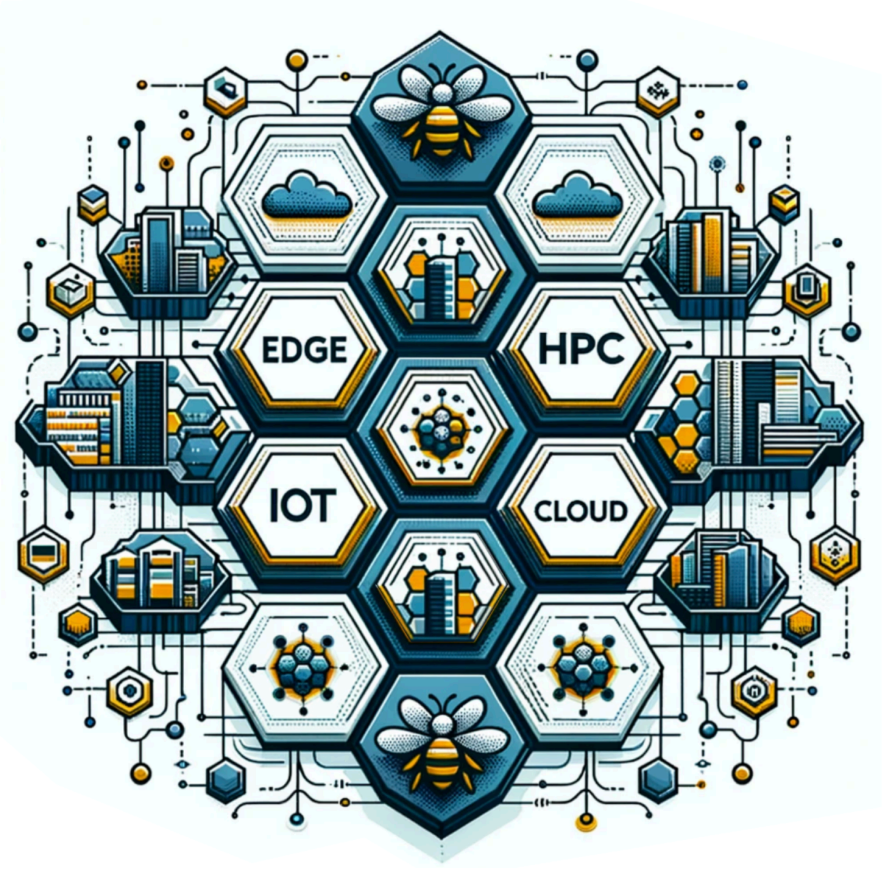
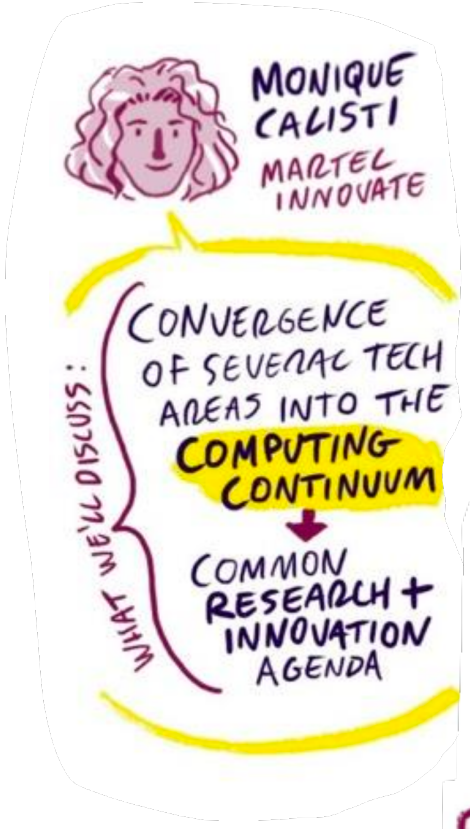
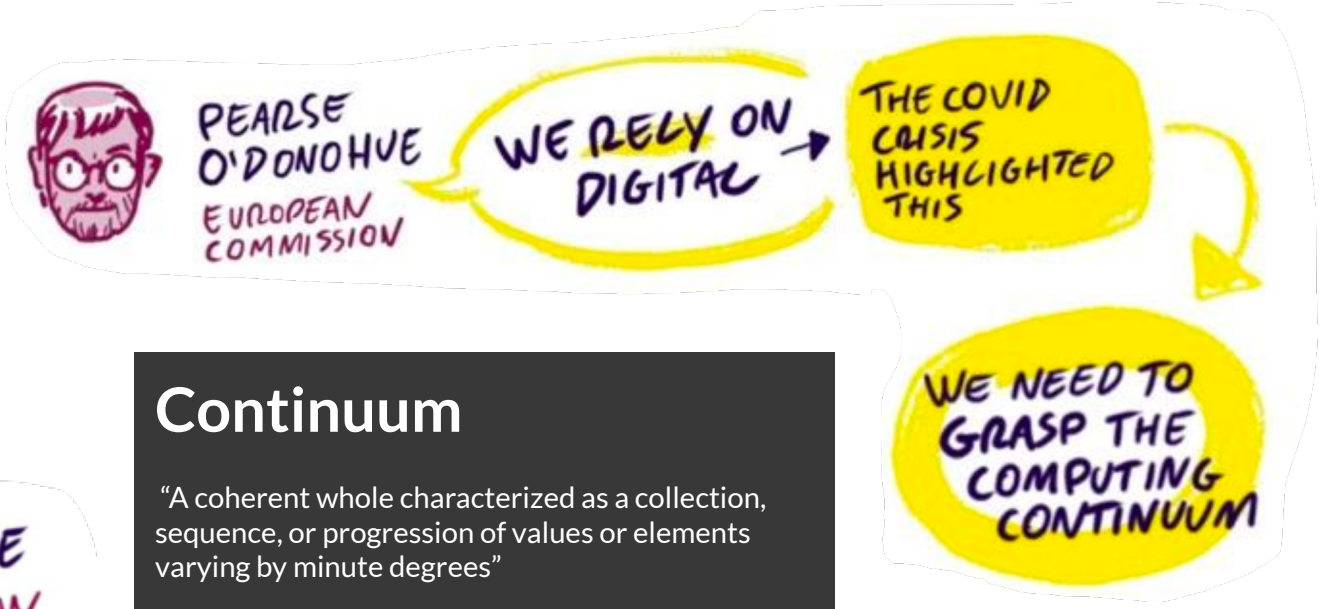


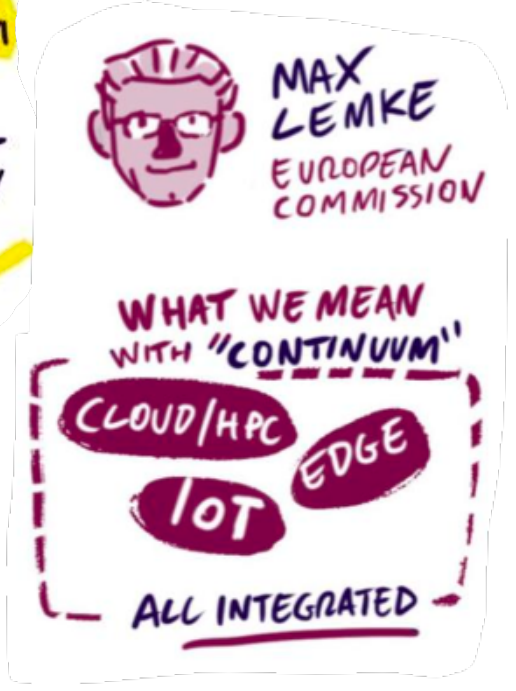
Distributed Quantum Computing

Erik Källman, RISE

QAS 2024







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 …

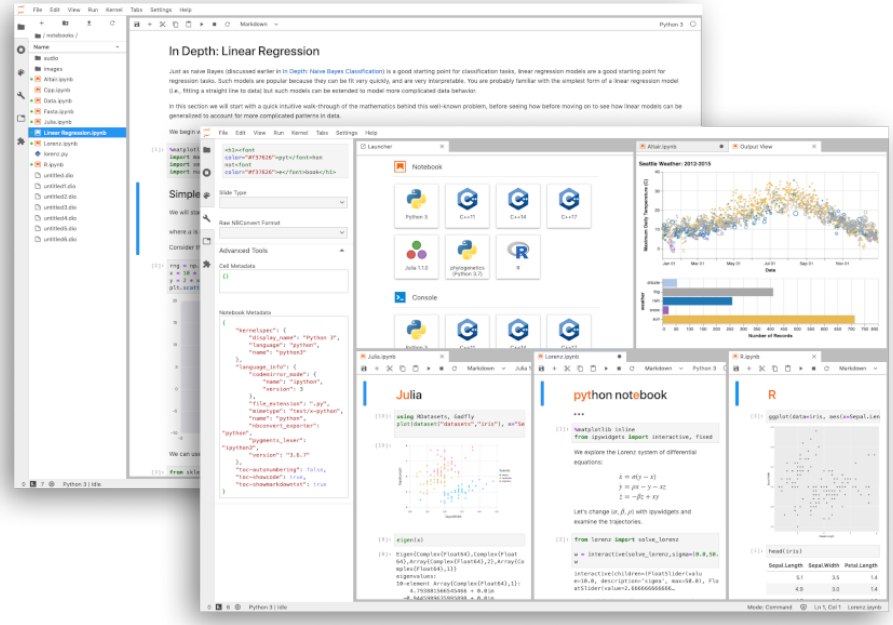
Generated by ChatGPT

- **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

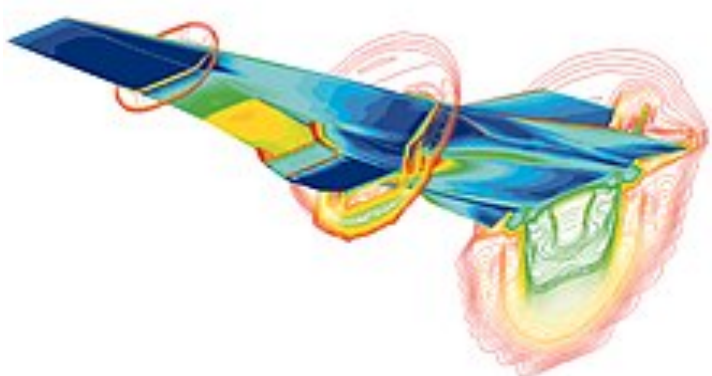
RI.
SE

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



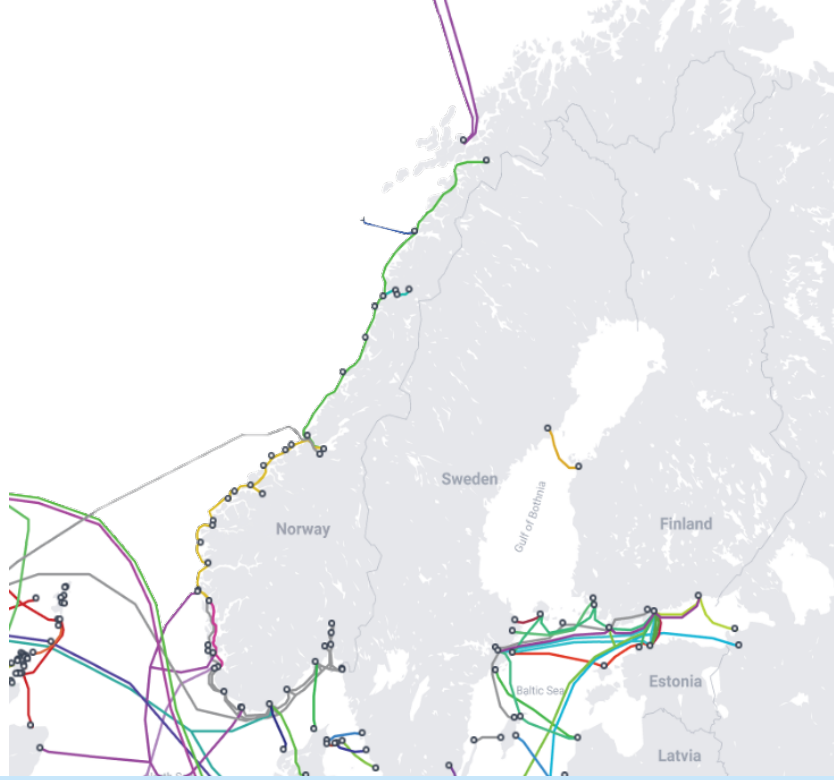
- Why not use cloud platforms?
  - Cloud platforms can be very complex and cumbersome to use for researchers
  - Cloud platforms like Kubernetes are not designed for HPC workloads *(not optimized for performance)*



RI.
SE

Problems with Cloud Computing

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



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

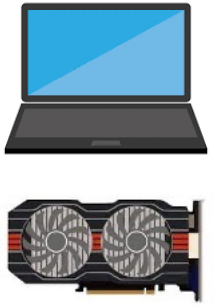
EDWARD STRINGER + Add to myFT

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

HPC ***Local*** ***Cloud***



Local

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

Compute Continuum

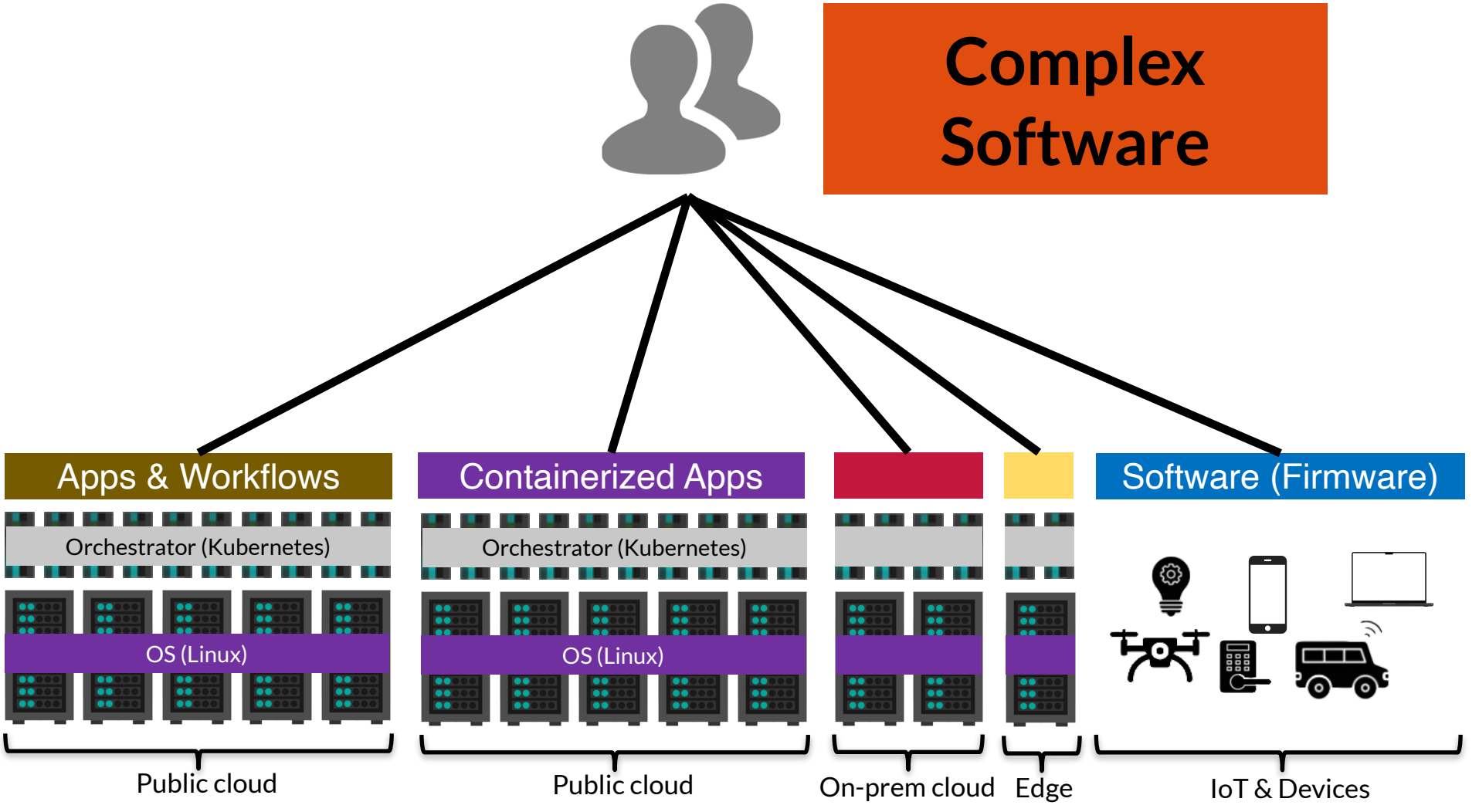
- Simplify cloud accessibility for HPC users
- Seamless migration to cloud after using EuroHPC

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

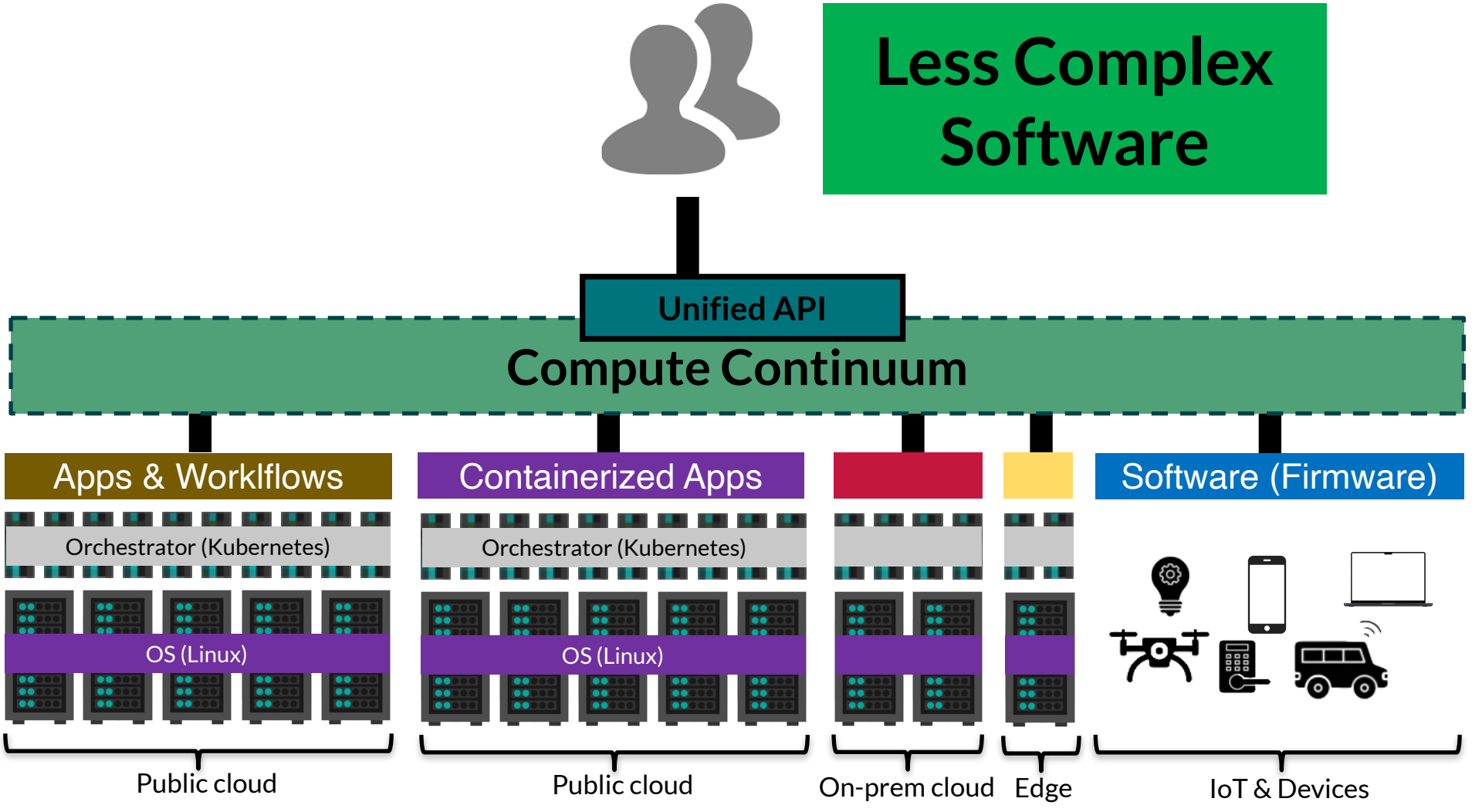
HPC

?

Cloud


Complex Software
Apps & Workflows
Orchestrator (Kubernetes)
OS (Linux)
Containerized Apps
Orchestrator (Kubernetes)
OS (Linux)
Software (Firmware)
Public cloud
Public cloud
On-prem cloud
Edge
IoT & Devices

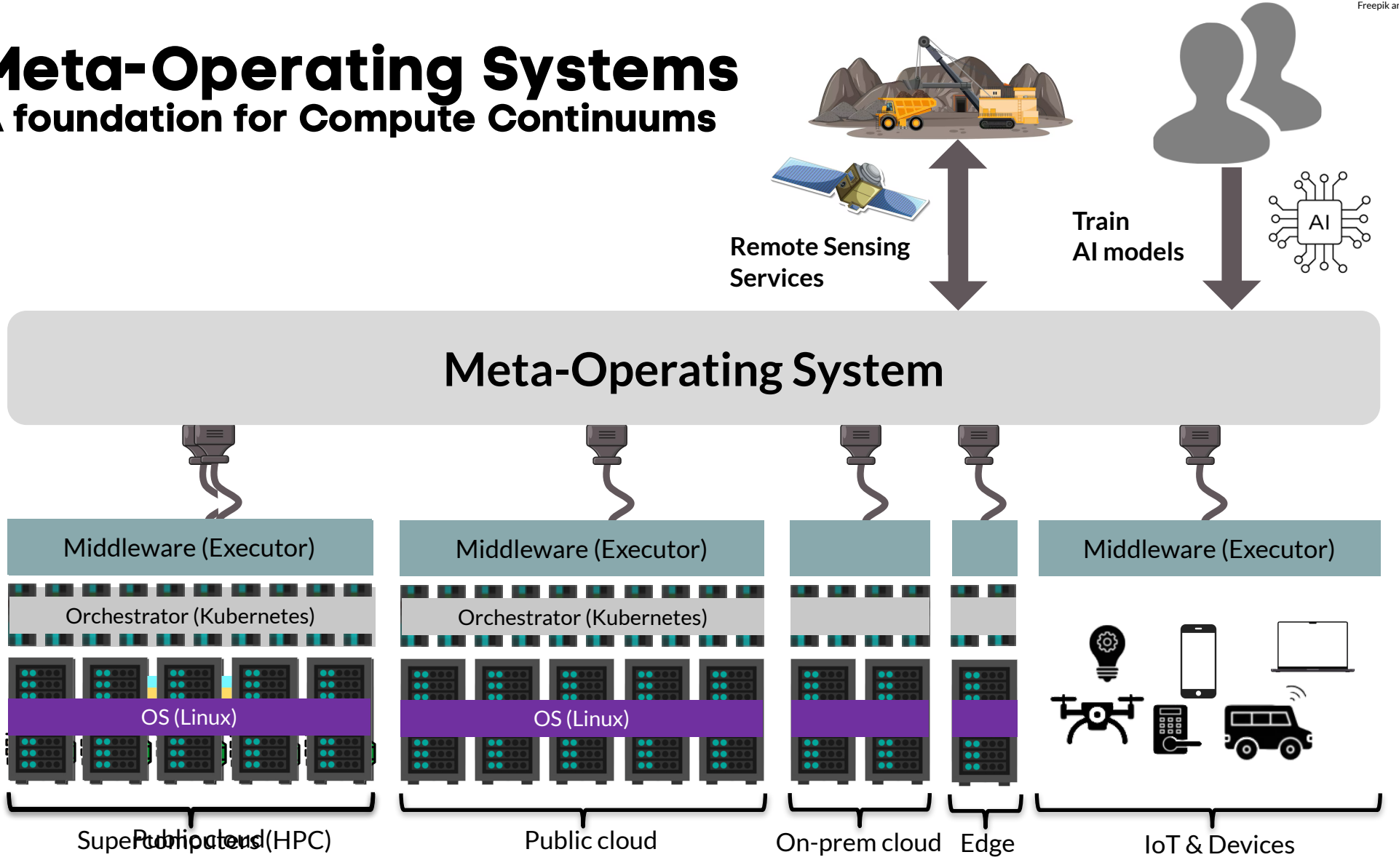
RI.
SE


Less Complex Software
Unified API
Compute Continuum
Apps & Worklflows
Containerized Apps
Software (Firmware)
Orchestrator (Kubernetes)
Orchestrator (Kubernetes)
OS (Linux)
OS (Linux)
Public cloud
Public cloud
On-prem cloud
Edge
IoT & Devices

RI.
SE

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.

Read more | Getting started | Contact us

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

https://colonyos.io

ColonyOS

3 followers · Sweden

Unfollow

Overview | Repositories 16 | Projects | Packages | Teams | People 3 | Settings

Pinned

Customize pins

- colonies (Public) — Colonies is a distributed execution environment for executing cross-platform workloads. — Go
- 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
- rust (Public) — Colonies SDK in Rust — Rust

Repositories

Find a repository... | Type | Language | Sort | New

- colonybuilder (Public) — Filesystem executor — Go — Updated 15 hours ago
- colonies (Public) — Colonies is a distributed execution environment for executing cross-platform workloads. — Go, MIT — Updated 3 days ago
- helm (Public) — Shell — Updated 4 days ago
- docs (Public) — TeX — Updated 3 weeks ago
- notebooks (Public) — Updated on Apr 18
- pycolonies (Public) — Colonies SDK in Python — Python, MIT — 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

Invite someone

Top languages

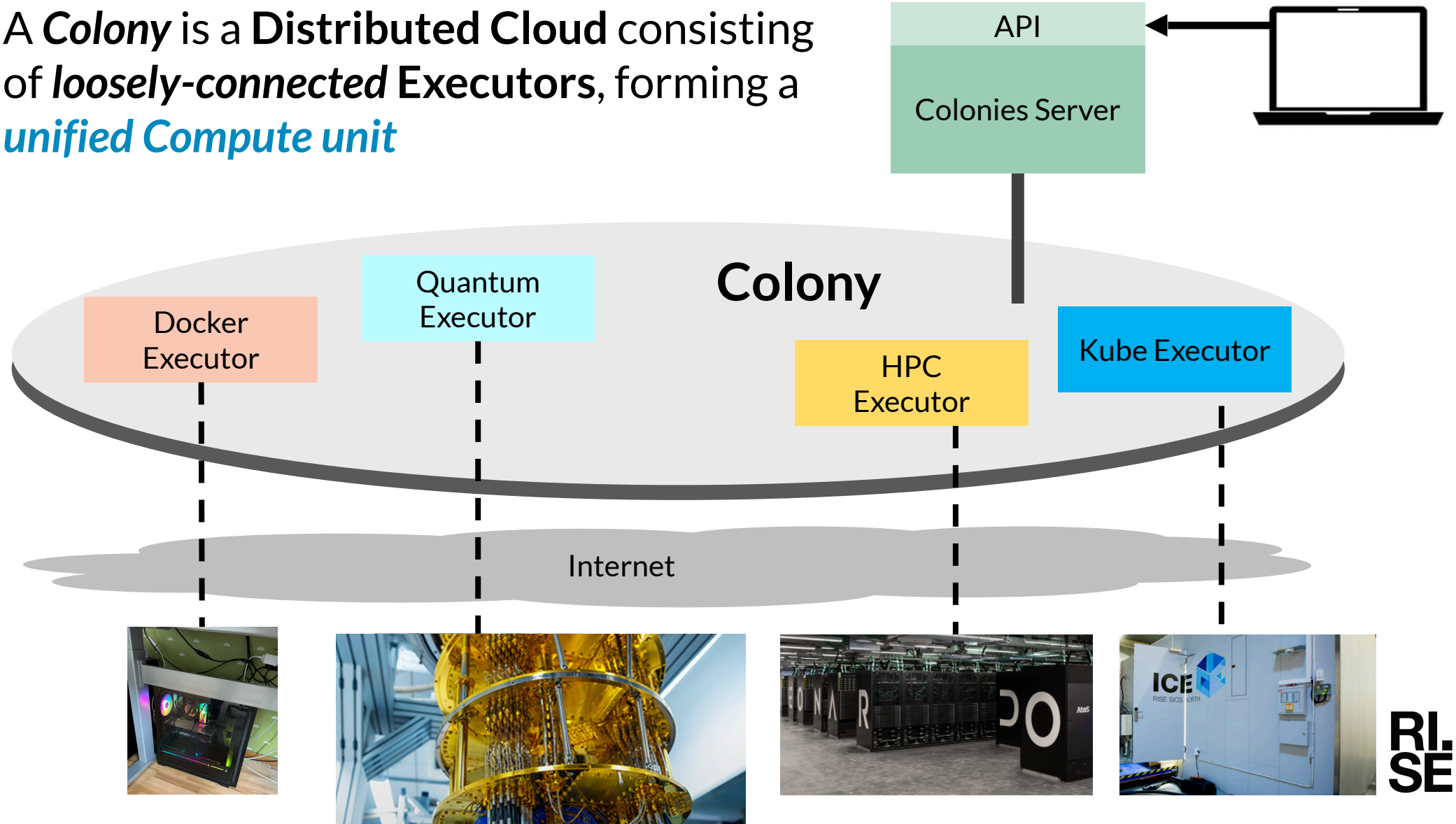
Go, JavaScript, Python, Julia, TypeScript

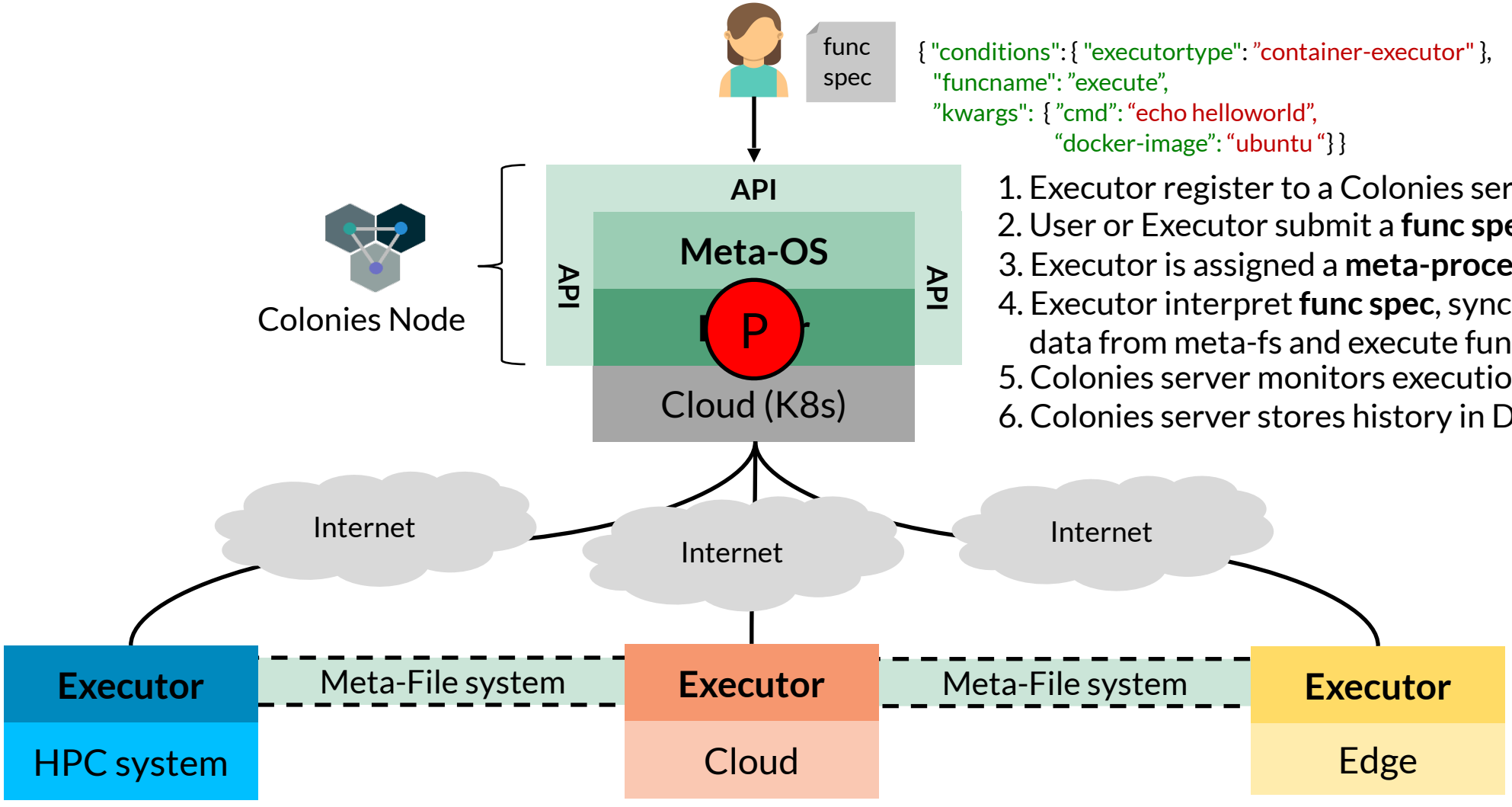
Most used topics — Manage

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

https://github.com/colonyos

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



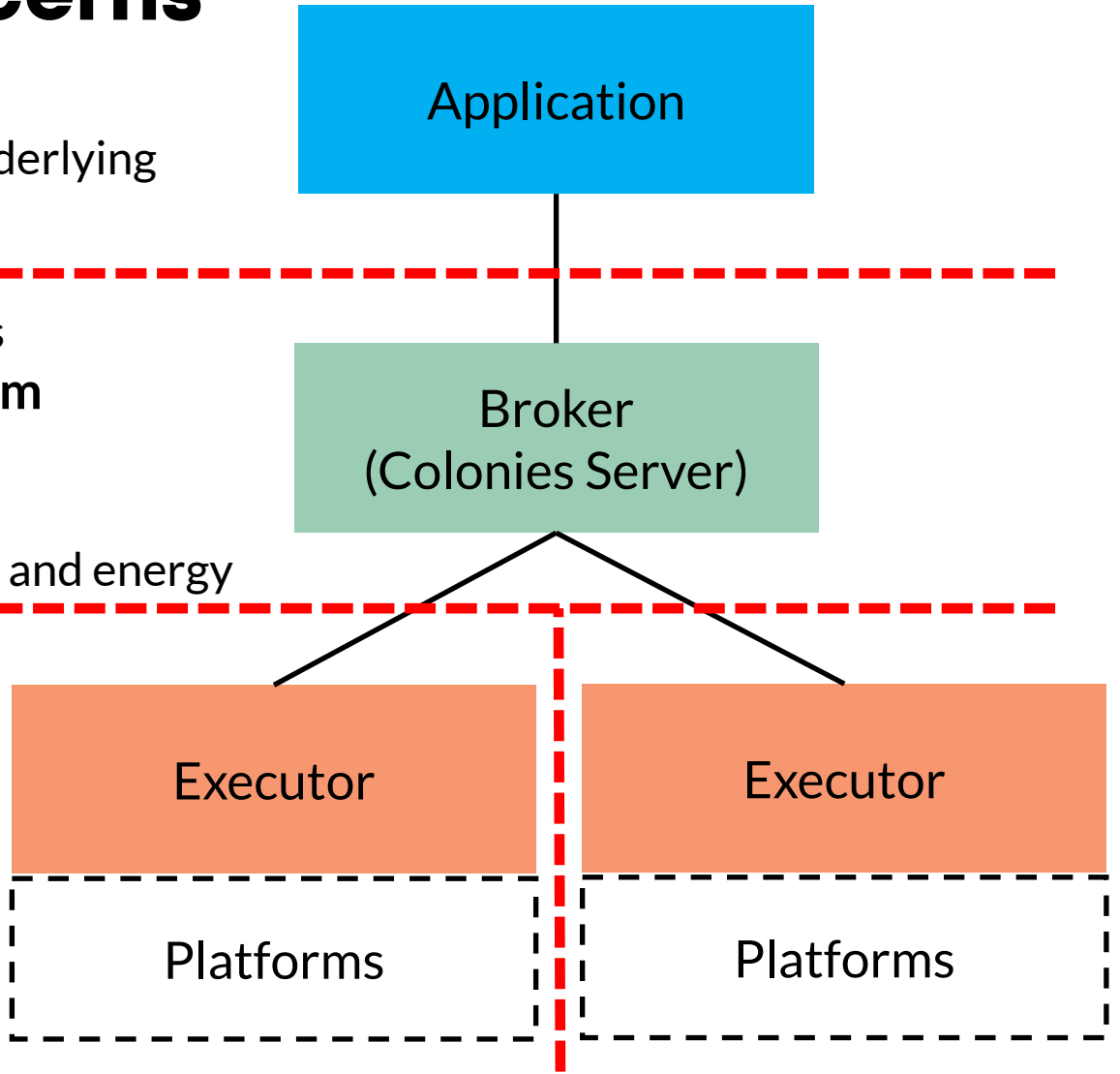

func
spec
{ "conditions": { "executortype": "container-executor" },
"funcname": "execute",
"kwargs": { "cmd": "echo helloworld",
"docker-image": "ubuntu "} }
Colonies Node
API
Meta-OS
API
API
P
Cloud (K8s)
1. Executor register to a Colonies server
2. User or Executor submit a func spec
3. Executor is 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
Internet
Internet
Internet
Executor
HPC system
Meta-File system
Executor
Cloud
Meta-File system
Executor
Edge

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

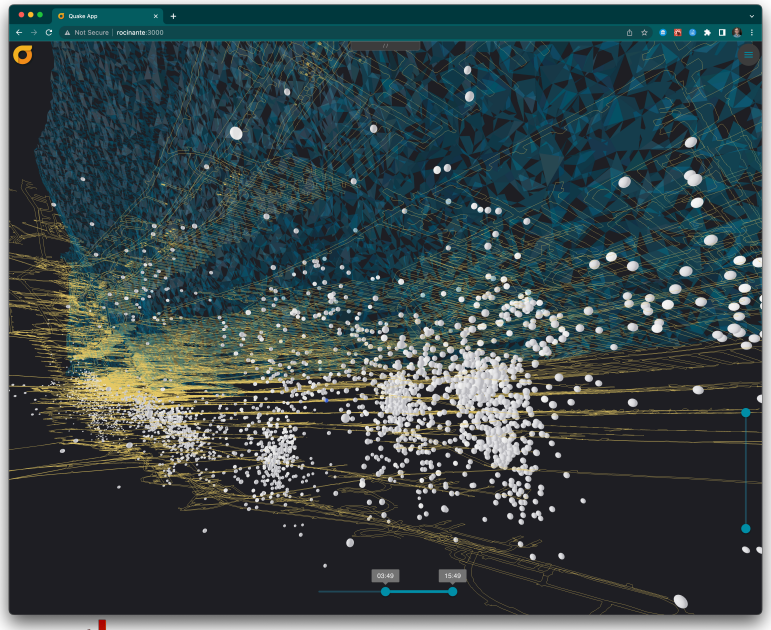
- Executors are microservices designed to execute specific functions
- Integrate with other platforms
- System integrator
- Reside anywhere on the Internet

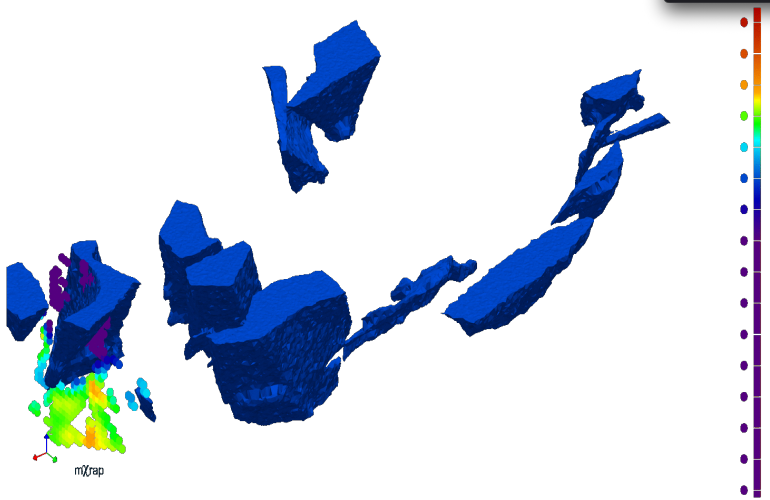


RI.
SE

RockSigma AB

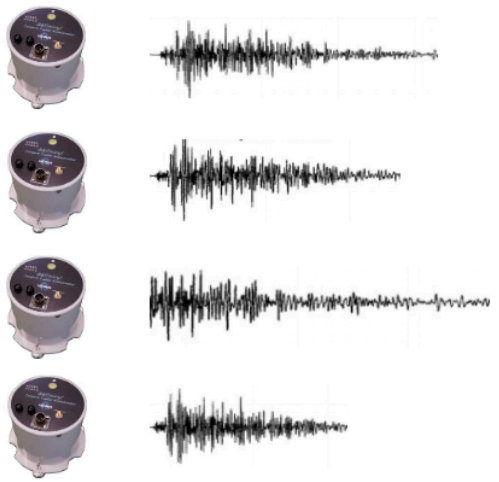
- Seismic processing underground mines
- Used by LKAB to analyze seismicity and process a massive amount of data from one the largest mines in the world (Kiruna/ Malmberget)
- On-preem + cloud





RockSigma AB

Seismograms

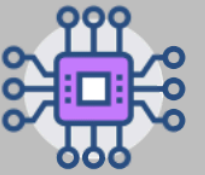


Sensor Infrastructure

Seismic Processing Engine

API

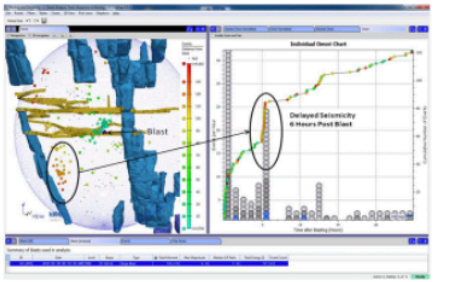
BEMIS



API

Kubernetes

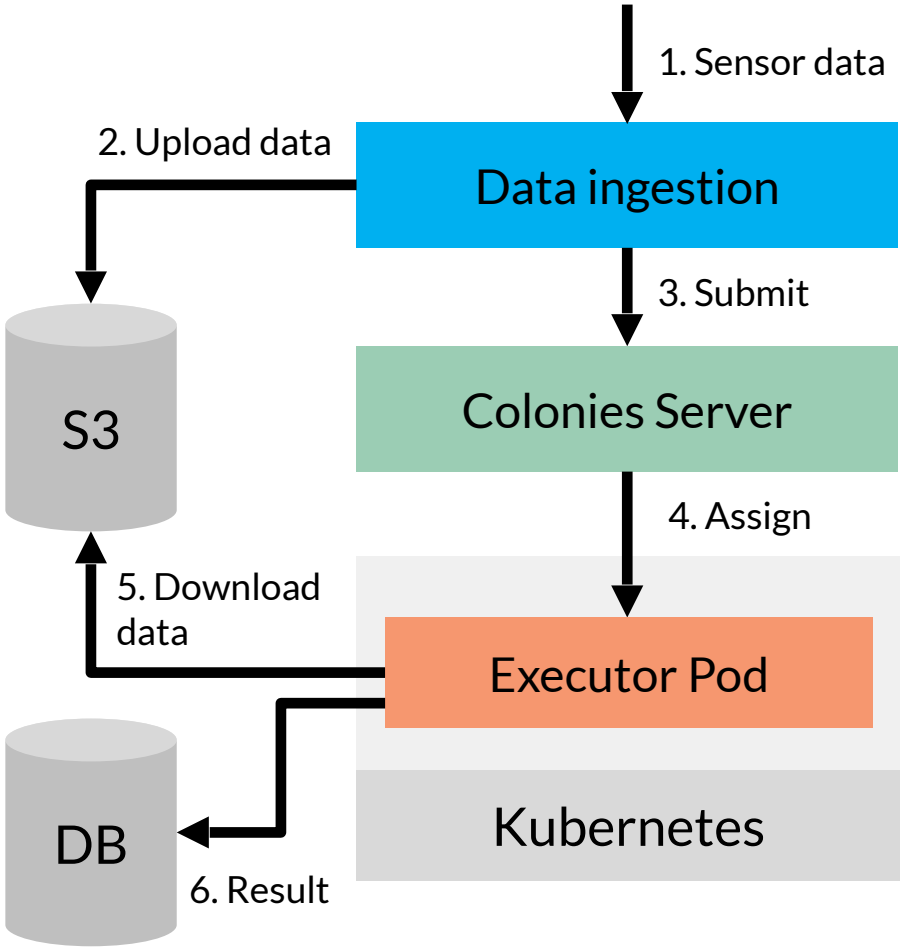
Visualization

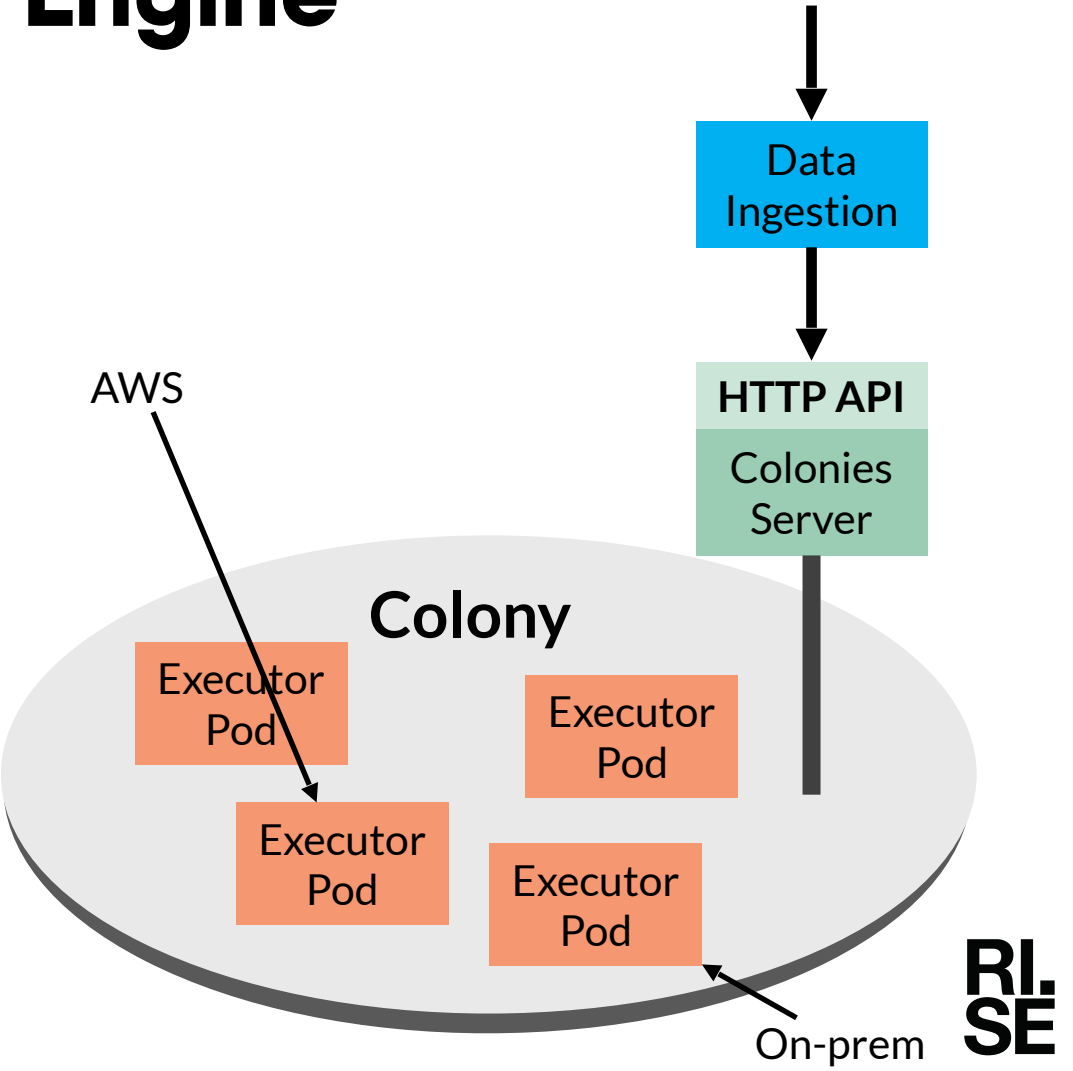


10^6 seismograms every day
>100 TB every year

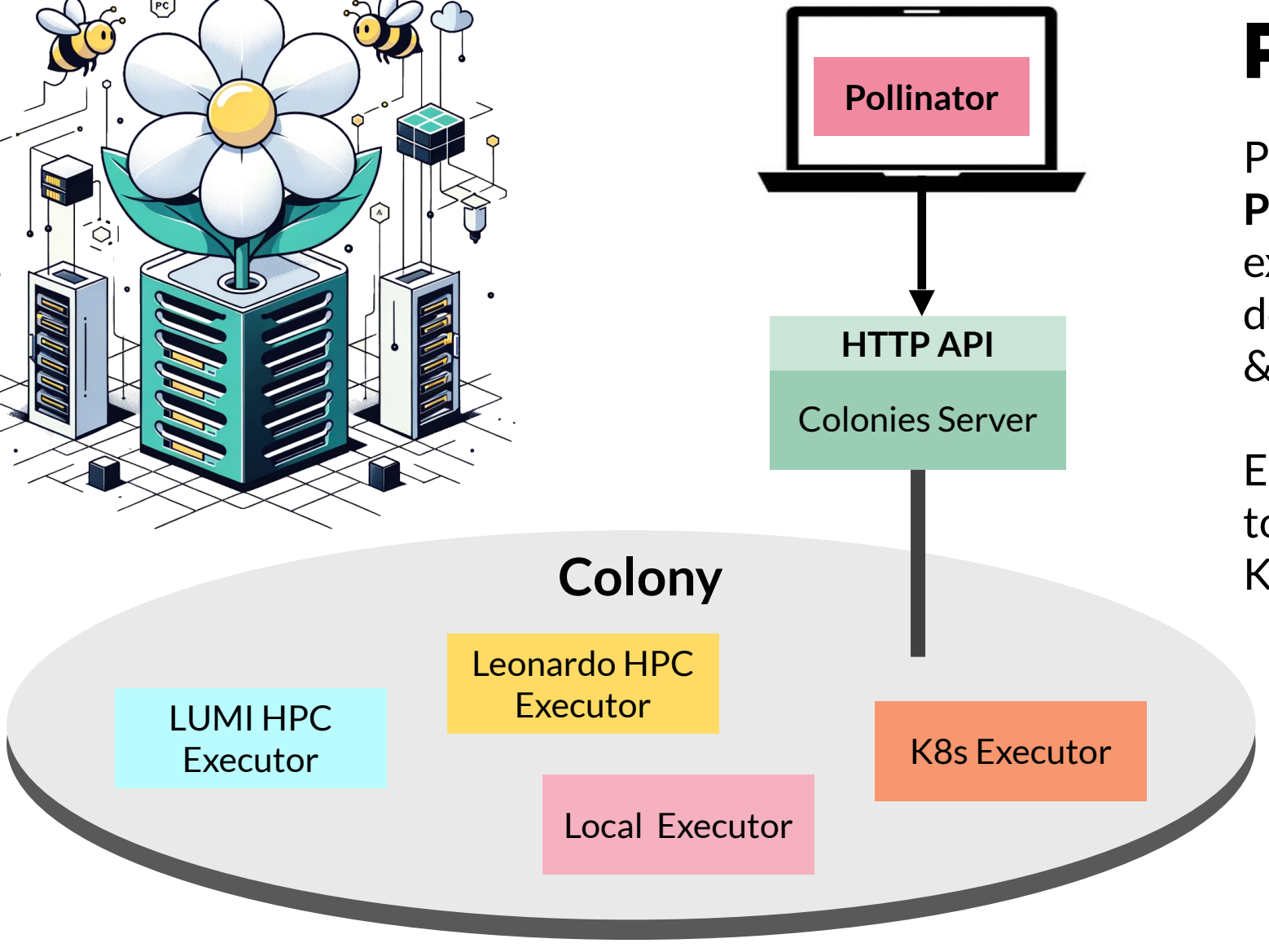
RI.
SE

A Seismic Processing Engine





RI. SE



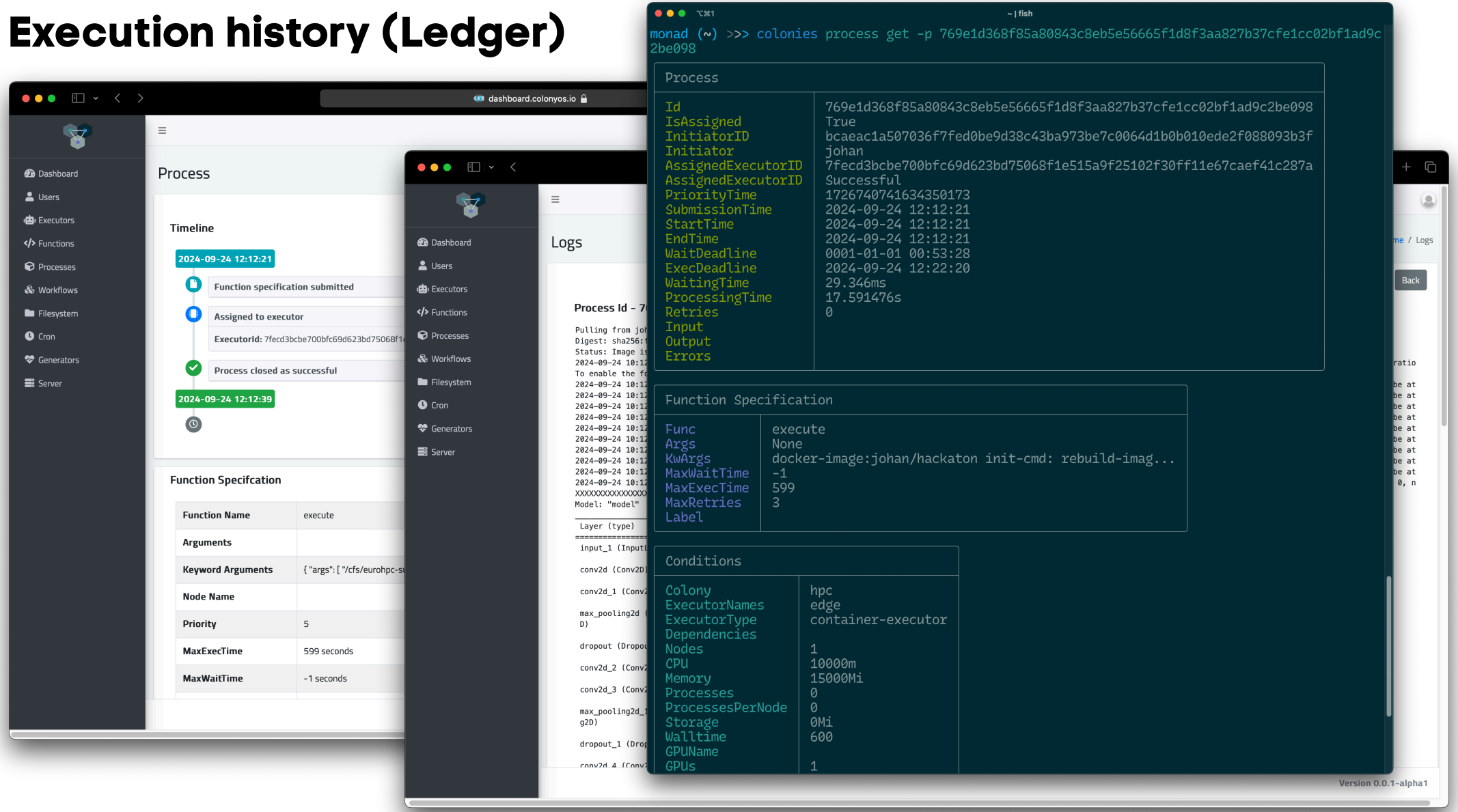
Pollinator

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

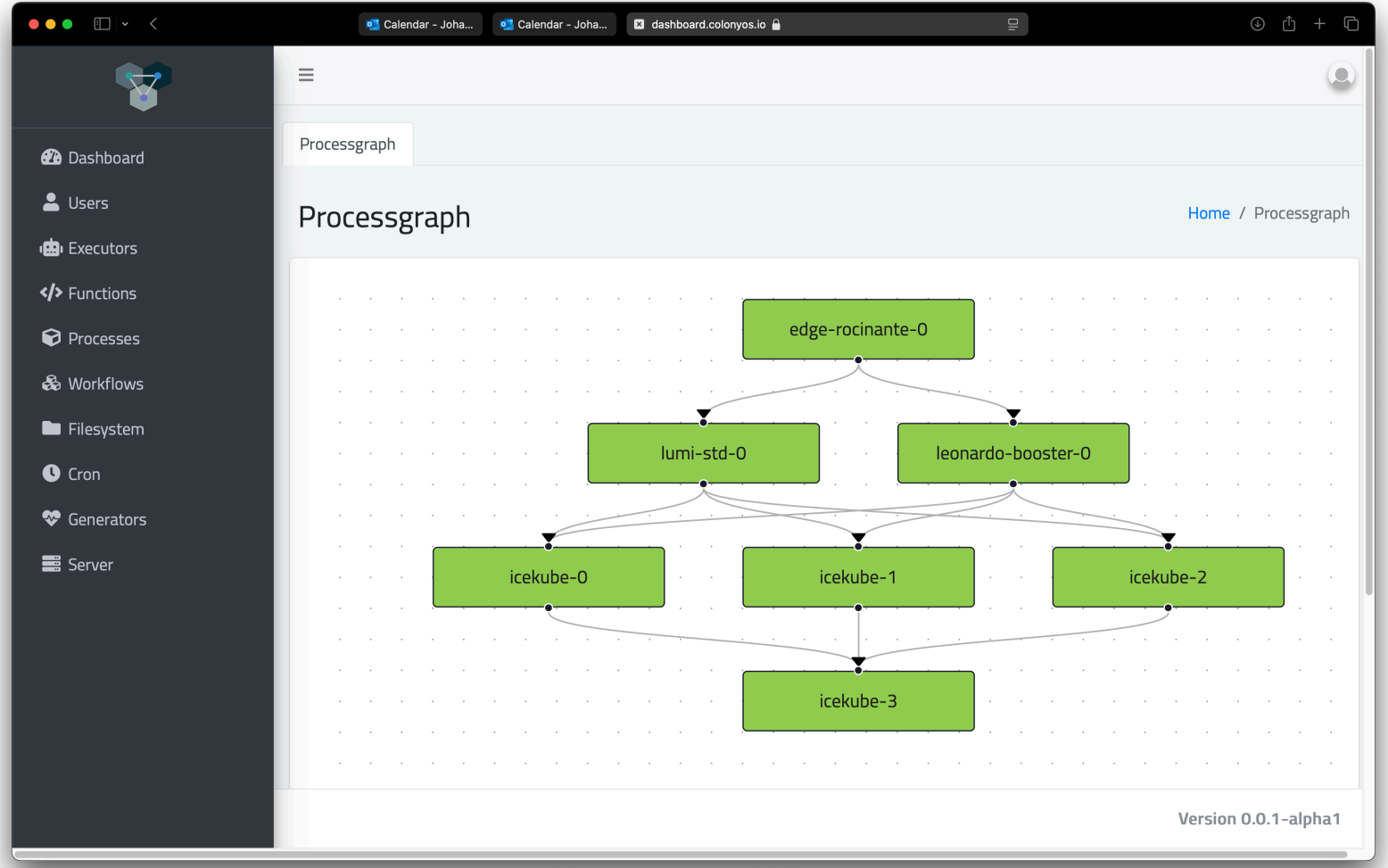
Eliminates the need to learn Slurm, Kubernetes

RI.SE

Execution history (Ledger)



Cross-platform workflows



github.com

Files

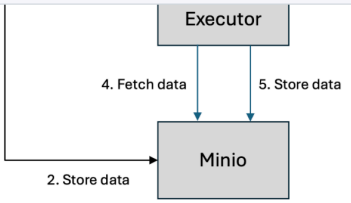
main

Go to file

- 01-getting-started
  - README.md
  - cat.json
  - echo.json
  - gen_file.json
  - overview.png
  - overview.pptx
  - python.json
  - python_snapshot.json
- 02-colonyfs
- 03-python
- 04-faas
- 05-workflows
- 06-crons
- 07-pollinator
- 08-security
- 09-production
- 10-k8s-executor
- 11-hpc-executor
- 12-anomaly-detection
- 13-earth-observation
- LICENSE
- README.md

tutorials / 01-getting-started / README.md ↑ Top

Preview | Code | Blame | 330 lines (277 loc) · 23 KB | Raw



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