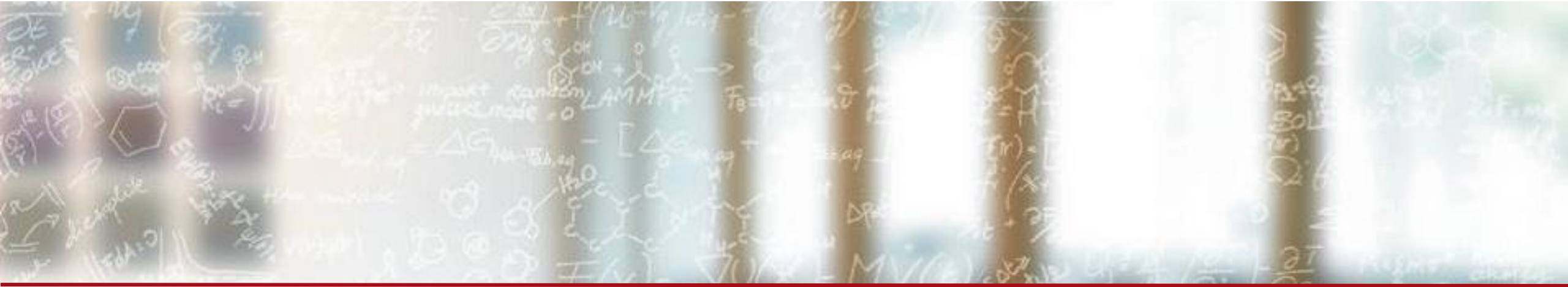




**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



# Enhancing Infrastructure Management and Datacenter Visibility and Security at CSCS with MarmotGraph

Massimo Benini, CSCS / ETH Zürich  
HPC&AI Advisory Council, Locarno April 23 2026

Problem: how to master the complexity explosion in managing Data Centers

## *“Modern HPC centers are undergoing a profound transformation”*

- No longer a single monolithic supercomputer
- HPC-Cloud convergence
- Heterogeneous HW
- **Software-Defined Model:** CSCS has moved toward infrastructure fragmented into versatile clusters tailored for specific communities
- **The "vCluster" and "vService" Challenge:** Utilizing technologies like "vCluster" provides flexibility but significantly increases metadata complexity
- **Dynamic Tracking:** Tracking the dynamic, shifting state of the center is increasingly difficult with traditional methods
- Hundreds of projects and thousands of users
- Matrix team organization

# Isolated silos

*“There is a lack of a holistic overview of the center and its assets”*

- Data centers rely on **isolated** “sources of truth”:
  - openDCIM → HW mgmt
  - NetBox → Network mgmt
  - RedHat Satellite → images mgmt
- **Lack of Traceability:** These tools function as isolated silos, hindering cross-domain queries
- Different tools, different levels of abstraction
- Different update intervals
- Capture contextual and unstructured information

**Our solution:** Leveraging neuroscientific tools to manage the next generation of software-defined supercomputing.

# A brief history

- Built for the FET flagship project "The Human Brain Project" (HBP)
- First release in April 2018
- Continued development and operation during the successor project "EBRAINS 2.0"
- Associated to CSCS since 2024
- Extension from a "neuroscientific" project towards a more generic, domain-agnostic solution



# The FAIR principle

- Describe (neuro-)scientific data so it can be:
  - Found by others
  - Is **accessible** (not only by humans)
  - Is **interoperable** (works with various tools and services)
  - Is **reusable**
- Cover everything from "whole brains" down to the microscopic level, from neurorobotics to electrophysiology...

# Our approach: The centralized aggregation

- Putting real world entities into context
  - Extract information from multitude of tools
  - Alignment on the "meta layer" based on a center-wide, tool-unspecific metadata structure
  - Make contradicting information transparent
- This results in...
  - Annotations of the same entity in multiple levels of detail
  - Continuously evolving annotation schemes (new sources of information, new use-cases...)
- Relational databases are just not dynamic enough! We need a graph!
  - But "Thinking in graph" is hard and not very "user friendly"...

# The problem sounds familiar...

- CSCS already develops and operates the "EBRAINS Knowledge Graph" - a graph management system for making neuroscientific data FAIR (findable, accessible, interoperable and reusable)
- Similar problems
  - Multi-level annotations
  - Big heterogeneity
  - Complex interlinking of resources
- So, let's reuse the technology...



# MARMOT GRAPH

- **Management Applications for Rich Metadata Objects in a Graph**
- A rich toolset and service layer (incl. API) on top of graph engines
- Strengths in multi-source tracking, fine granular permission management and validation / publication mechanisms.
- Emphasizing the collaborative process of quality assurance between automation and "human expertise"

# More than "just" a graph database



**Explore**



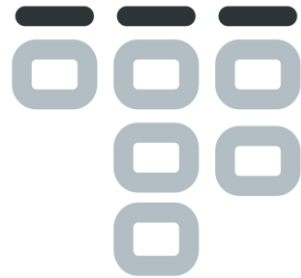
**Enrich**



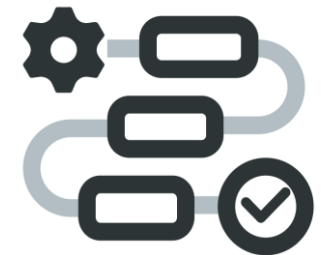
**Validate**



**Consume**

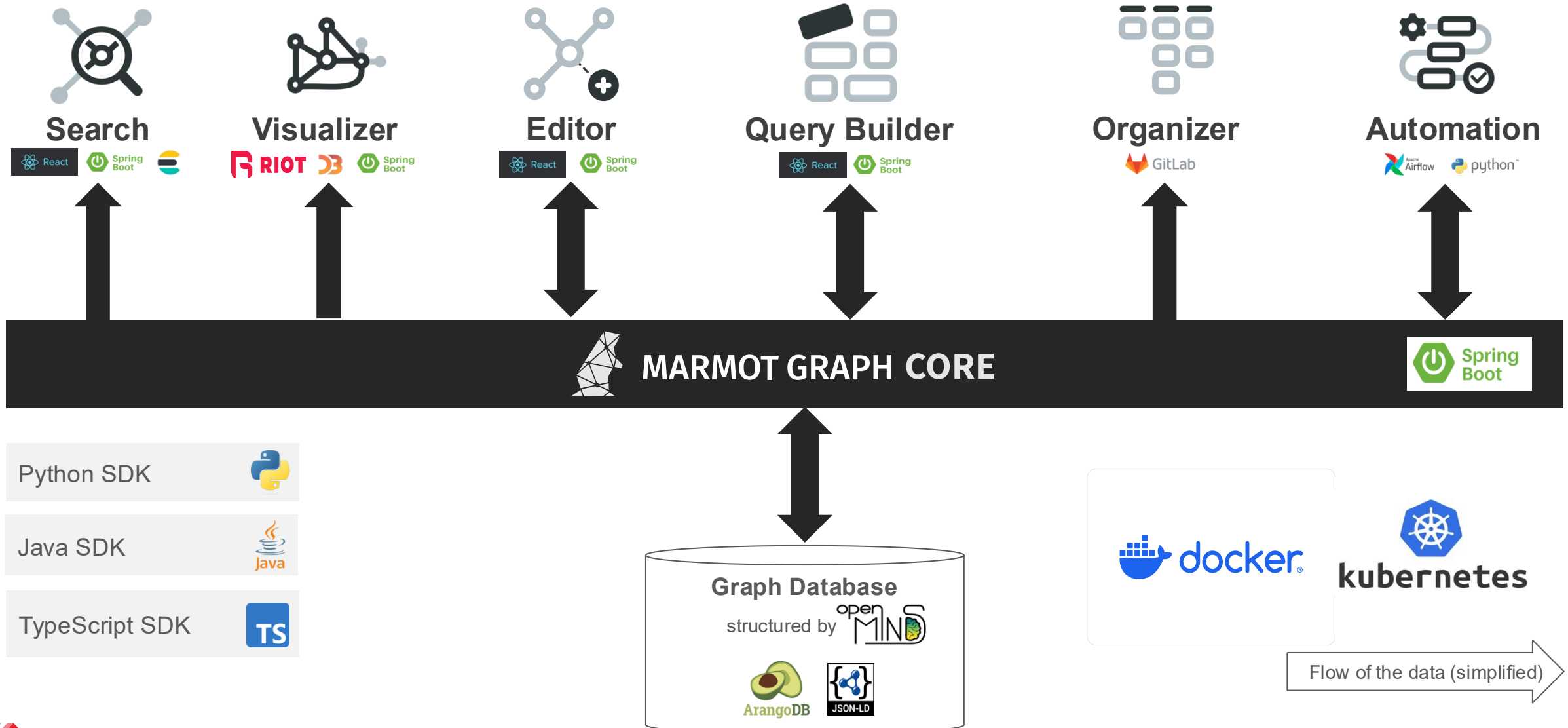


**Organize**

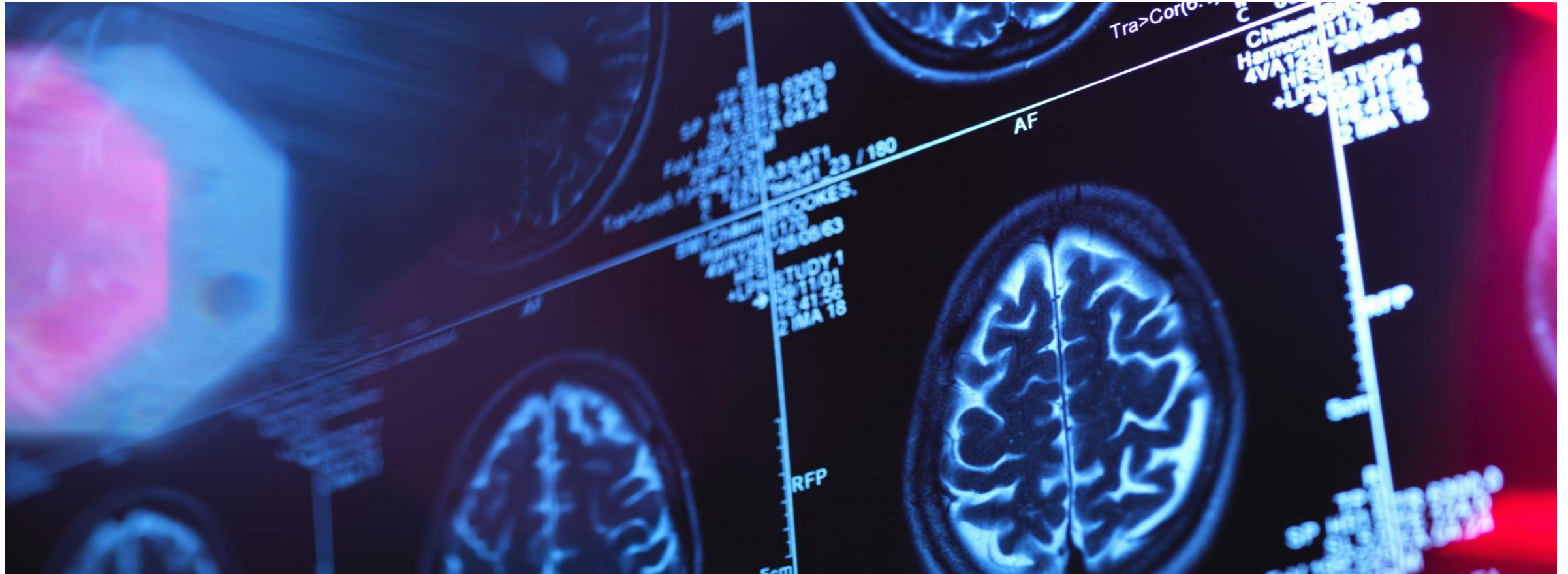


**Automate**

# More than "just" a graph database



# Bridging the gap: From neuroscience to HPC



# Based on the lessons learned from neuroscience

- Development of the metadata framework for linked neuroscientific data

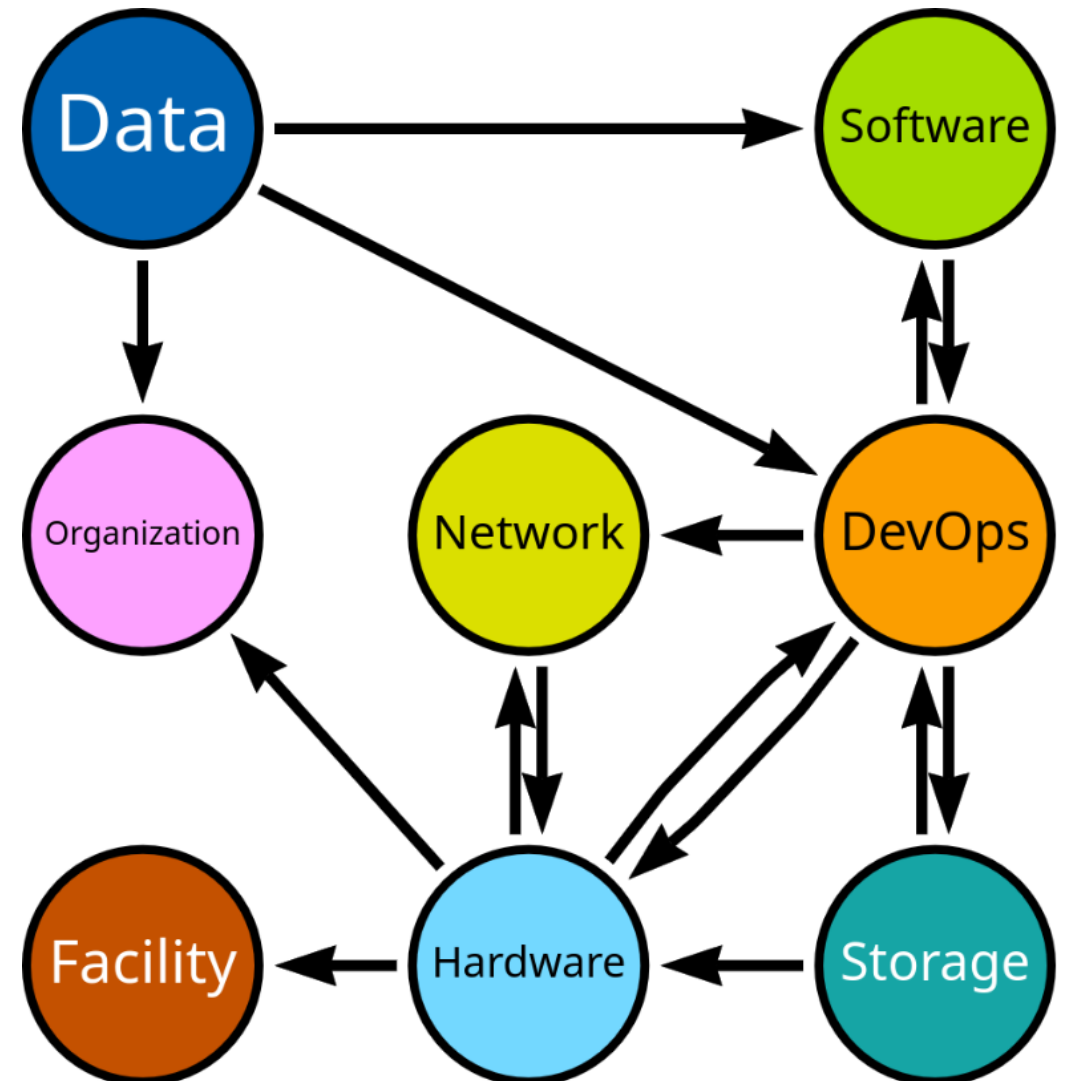


<https://openminds.om-i.org>

- Fast feedback loops, continuous technical feasibility checks and testing for practicability

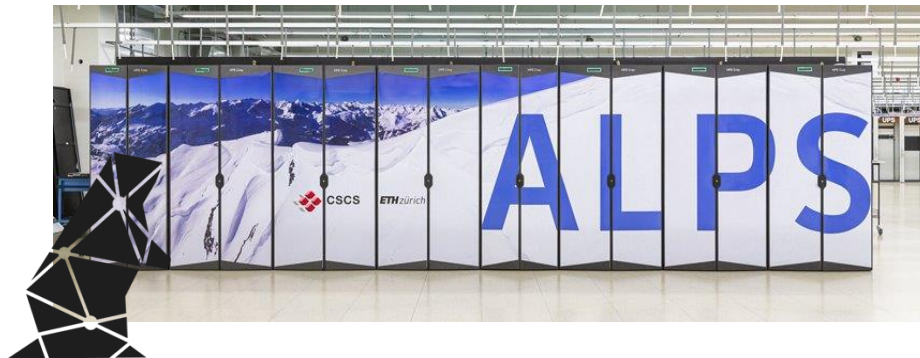
# Building the model

- *The model architecture was developed through a rigorous bottom-up modeling methodology designed to be applicable to **any** supercomputing center.*
- *Decompose the ecosystem into functional domains: Facility, Hardware, Network, DevOps, Software, Data, Organization and identify granular entities (nodes)*
- *Both neuroscience and supercomputers share similarities: they handle complex, multidimensional relationships among entities and support multiple levels of details*



# Model extensions for CSCS

- Extension of the generic model with CSCS customizations
  - Custom entities such as "vCluster" or "vService"
  - Organization structures of the centre
  - Making the tool multi-tenant capable
  - Dataset Catalog for Observability
- Creation of CSCS automation pipelines incl. center-specific business logic such as naming conventions / identifier mappings, etc.



# Our motivation

- Gathering an overview across multiple domains of the center
  - Build a central entry point for the collection of facts across the center (as a supporting tool for incident management, asset inventory, etc.)
  - En"code" implicit knowledge (such as naming conventions, etc.)
  - Link information for the detection of inconsistencies and the inference of new knowledge around the center
- Ensure the continuity of specialized applications to address the day-to-day needs of the various domains



# The current graphs

## CSCS KG

~240'000 entities\*  
(3.4 Mio triples, 75 types)

## EBRAINS KG

>4.5 Mio entities\*  
(>31 Mio triples, ~250 types)

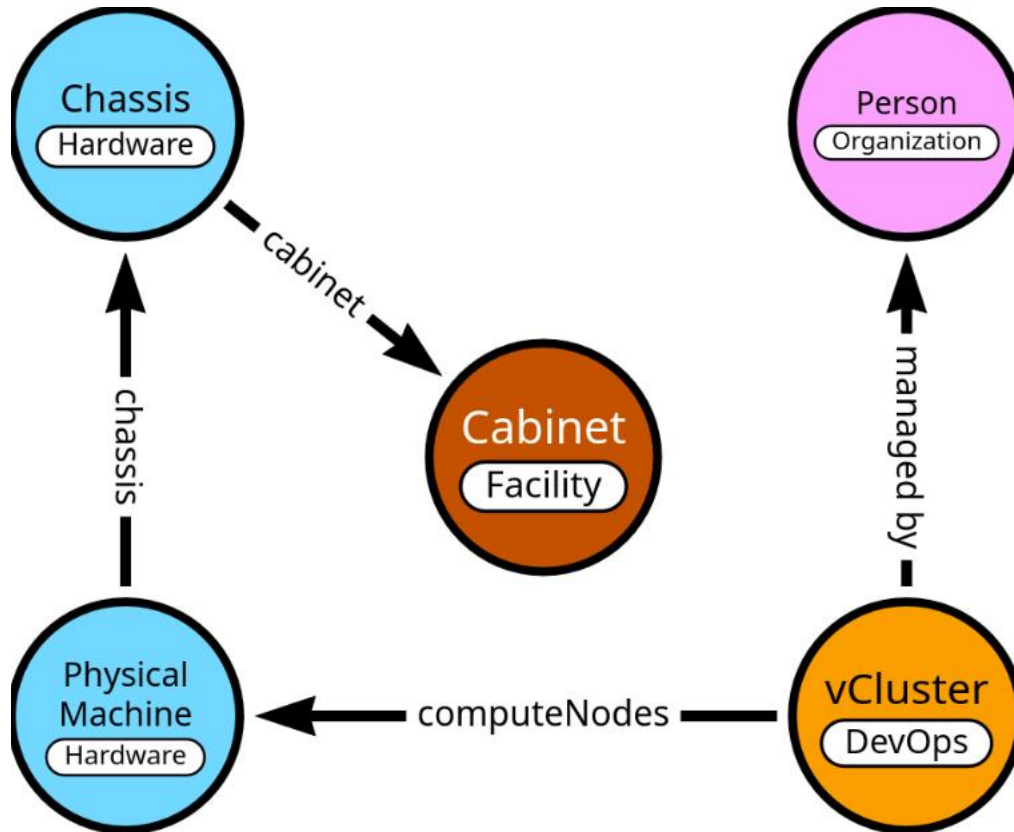
\*State on Oct 22nd, 2025



# Use cases in day-to-day work

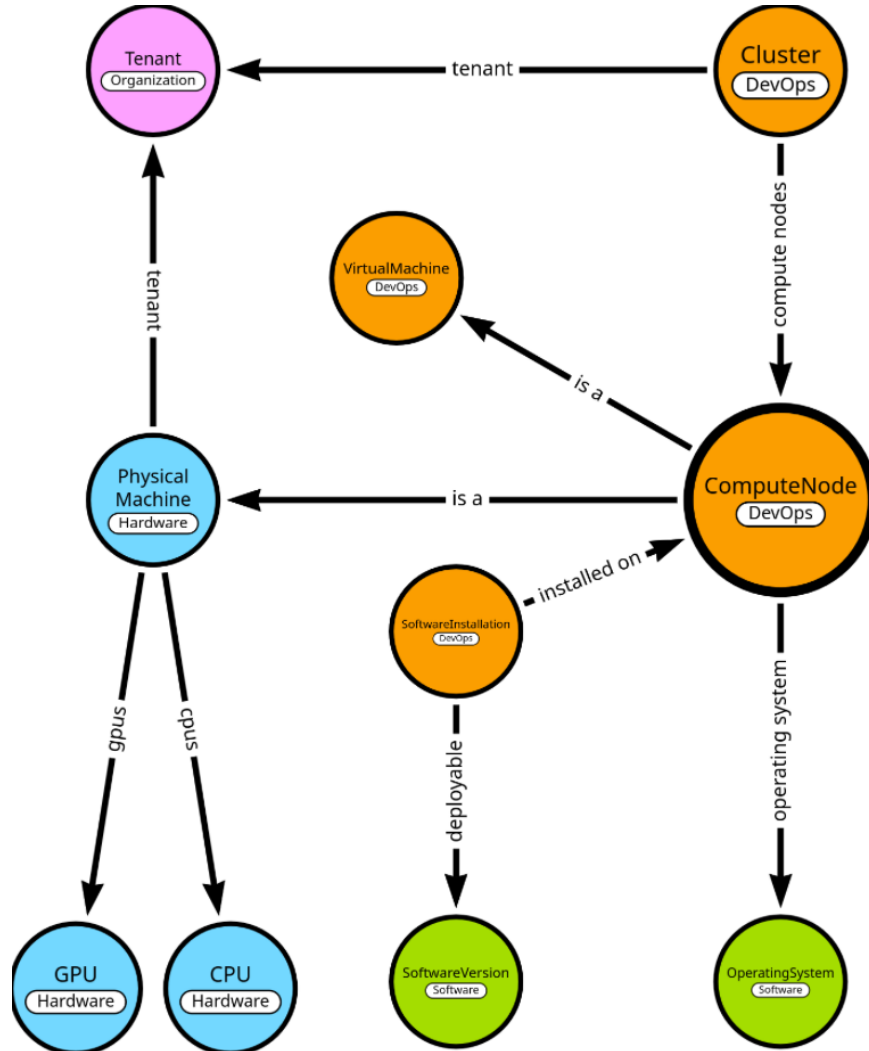


# Assessing HW failure impact



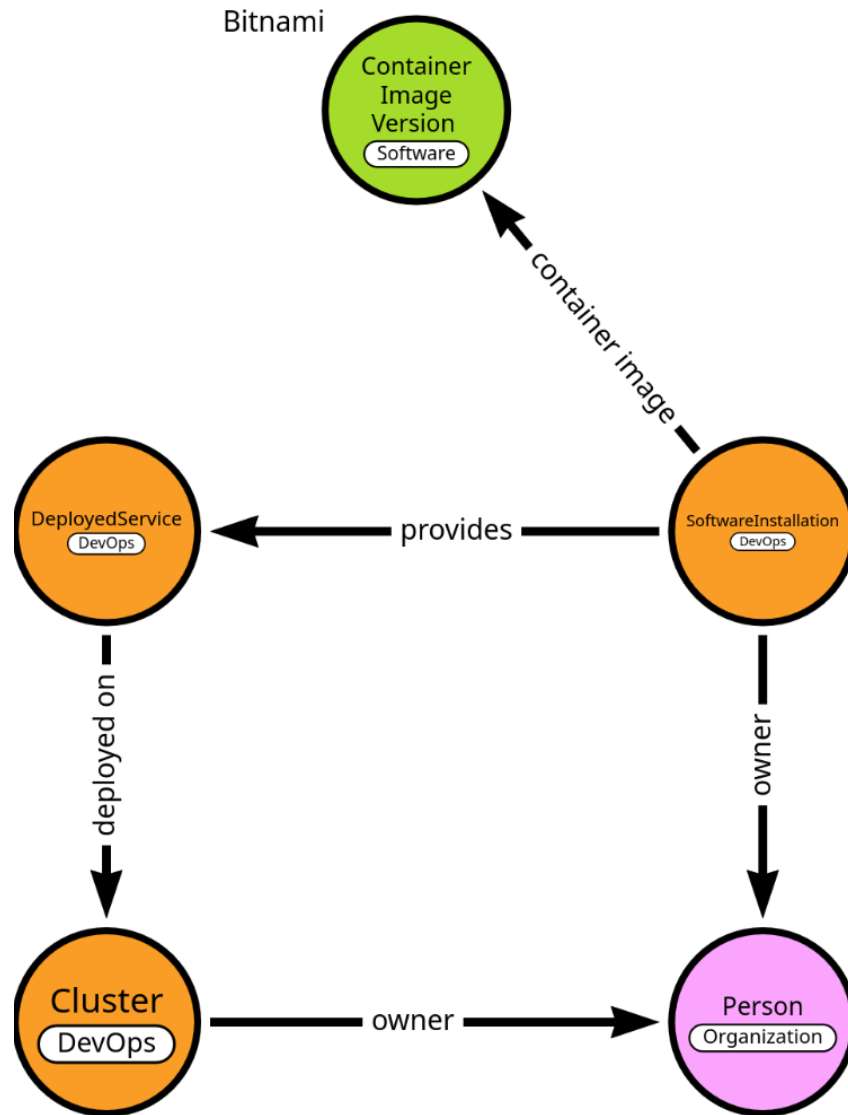
- ✓ Starting from the facility domain (the Cabinet)
- ✓ Traverse the edges to Hardware domain (the Compute nodes)
- ✓ Follow the semantic links into the DevOps domain (vClusters and Vservices)
- ✓ Notify the people involved

# Mapping HW assets and SW



- ✓ Starting from the Tenant node
- ✓ Traverse the graph into the HW assets to identify the bare-metal nodes, GPUs, storage etc etc
- ✓ Traverse the graph into the SW assets to list Operative Systems, drivers, applications installed on the node
- ✓ The result is a detail asset and software JSON manifest to be easily consume for reporting

# Mitigating deployment risks: the bitnami case



- ✓ Start traverse the graph from the Image Registry Node
- ✓ Traverse the edges to the Deployed Service nodes
- ✓ This single query provides a list of all deployments using the Bitnami repositories

# CVE impact analysis: identify the “blast radius”

- Before Marmotgraph:

**Software Scanning:** Security teams manually searched for vulnerable package versions.

•**Infrastructure Correlation:** DevOps teams were contacted to identify which clusters or services were affected.

•**Network Mapping:** Manual consultation of network configs to check for public internet exposure.

- After Marmotgraph:

A single global traversal starts directly from the **CVE-ID**

•**Physical details:** machine serial number and cabine/rack location (**Facility Domain**)

•**Connectivity:** IP address and domain to determine external reachability (**Network domain**).

•**Impact:** The specific vCluster and services that will be interrupted by a reboot (**DevOps domain**).

•**Accountability:** The specific CSCS employee and lead who must authorize the patch (**Organization domain**).

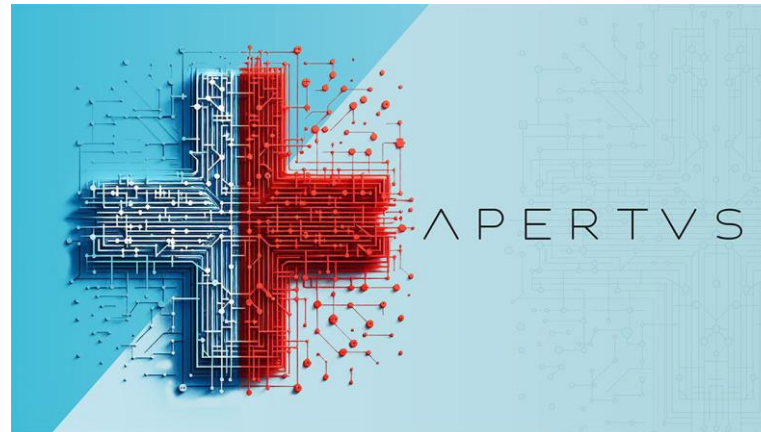
Shorter MTTR!

# Why caring about knowledge graphs? There's AI!

- Yes, AI can help – but...
  - Hallucination is still a problem (and very dangerous when facts matter)
  - Semantics are hard for everyone (incl. humans)
  - Outdated, bad and contracting data remains a problem
  
- So, no AI at all?

# Why caring about knowledge graphs? There's AI!

- AI can help in curation, validation, inference and interfacing (e.g. via GraphRAG/MCP) - and yes, we're experimenting with it...



- But the establishment and maintenance of a reliable source of truth (incl. modelling) will always require human expertise because building a KG is more than serializing and interpreting information...

# Where can I find more information?

- <https://marmotgraph.org>
- Via [support@marmotgraph.org](mailto:support@marmotgraph.org)



Claudio Goffi



Oliver Schmid



Annapaola Iolanda  
Santarsiero



Massimo Benini

