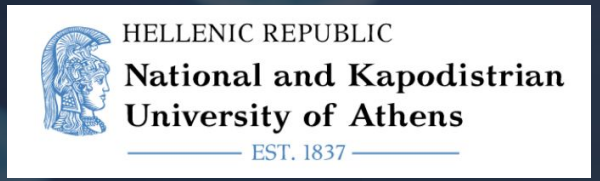
Hyper-CC / CLOUD COMPUTING 2026

# Hypertool: A Resource Abstraction and Lifecycle Management Framework for Heterogeneous Cloud Environments

Michalis Loukeris, Nektarios Deligiannakis, Vassilis Papataxiarhis, Panagiotis Kechriniotis, Syed Mafooq Ul Hassan, Nicolas Louca, Herodotos Herodotou, Stathes Hadjiefthymiades

**Michalis Loukeris**
National and Kapodistrian University of Athens
mloukeris@di.uoa.gr

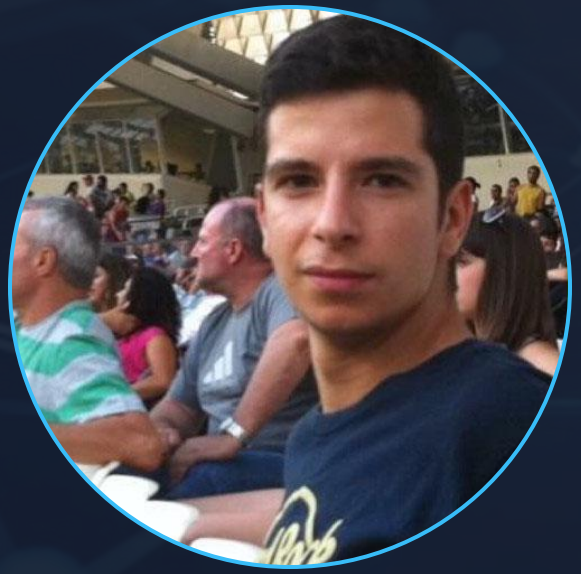



# Presenter’s Resume

## Michalis Loukeris

Researcher @ NKUA | P-Comp Group

- Working on the EU-funded HYPER-AI project
- Focus on Kubernetes orchestration & resource abstraction
- Expertise in cloud-to-edge lifecycle management



# Research Group & Projects

## Workgroup

**P-Comp Lab, NKUA**

## Interests

- Middleware for context-aware services
- Trust-based security for autonomous networks
- Game theoretic resource allocation models

## Current Projects

### HYPER-AI

Framework for AI-enabled orchestration across the continuum.

### EO4EU Platform

Next-generation EO data processing and analysis. Integrates Machine Learning, Knowledge Graphs, and Extended Reality to tackle platform intelligence and systems support.

# Introduction & Motivation

### The Problem

Standard platforms (e.g., K8s) use basic models limited to static CPU/RAM.

### The Challenge

Modern workloads demand awareness of acceleration, energy, and security.

### Complexity

Manual node lifecycle management is error-prone and requires high expertise.

### Solution: Hypertool

**Platform-Agnostic Framework**

Automated registration and extensible resource representation across clusters.

# Hypertool Architecture Overview

## Dual-Component Design

Combines a Command Line Interface (CLI) utility and a management DaemonSet.

## CLI Profiler

Used by administrators for node onboarding and lifecycle governance.

## DaemonSet

Ensures a monitoring pod runs on every worker node for continuous health checks.

## Declarative Model

Translates complex API interactions into Custom Resource Definitions (CRDs).

# CLI Functionality (Admin Interface)

- **Lifecycle Management**
  Centralized commands for discovery, self-advertisement, and registration.

- **Dry-Run Mode**
  Allows administrators to simulate operations and preview attributes before execution.

- **Robustness**
  Features sophisticated error management and automatic rollback mechanisms.

- **Visibility**
  Includes a verbose mode for diagnostic logs and deep troubleshooting.

# Continuous Monitoring (DaemonSet)

- **Continuous Lifecycle Management:** Maintains the ongoing management of nodes after their initial registration.
- **Automated Deployment:** Automatically schedules a dedicated monitoring pod onto every new worker node that joins the cluster.
- **Self-Healing Resilience:** Utilizes native Kubernetes reconciliation loops to stay operational and withstand localized failures.
- **Dynamic Polling Interval:** Runs a continuous execution loop every five minutes to keep resource information up to date.
- **API Patching & Recalculation:** Dynamically computes system attributes and updates the orchestration API during each polling interval.

# Extended Resource Attributes

- **Computational Performance:** Estimates GFLOPs via synthetic sub-second workloads for benchmarking.
- **Energy Efficiency:** Calculates performance-per-watt (GFLOPs/Joule) using hardware introspection or ML regressors.
- **Monetary Cost:** Assigns qualitative tiers (very low to very high) using K-Means clustering of hardware specs.
- **Trust & Security:** Evaluates operational stability (restarts, OOM kills) and security compliance (CIS benchmarks, CVEs).
- **Network & Hardware:** Monitors bandwidth/latency and detects specialized accelerators like GPUs, TPUs, and FPGAs.

# Validation & Results

## Local Validation

Tested on Minikube with a 12-core system; successfully computed all resource metrics.

## On-Premises Validation

Evaluated inter-node coordination and DaemonSet scaling on a multi-node virtualized cluster.

## Public Cloud Validation

Demonstrated profiling of diverse compute capacities and cost tiers in production-grade environments.

# Executive Summary & Outlook

## CURRENT STATUS

### System Summary

Hypertool bridges the gap between static orchestrator schemas and complex modern workload requirements.

### Operational Maturity

Demonstrated stability and ease of integration across local, on-prem, and public cloud infrastructure.

## FUTURE OUTLOOK

### Network Topology

Expanding the schema for deeper network topology metrics to enhance visibility.

### Predictive ML

Integrating predictive ML models for proactive anomaly detection and self-healing capabilities.

# Any Questions?

## Thank You!

**Contact:** mloukeris@di.uoa.gr

**Acknowledgments:** Supported by the HYPER-AI project (Horizon Europe Grant Agreement 101135982)