HYPER-AI

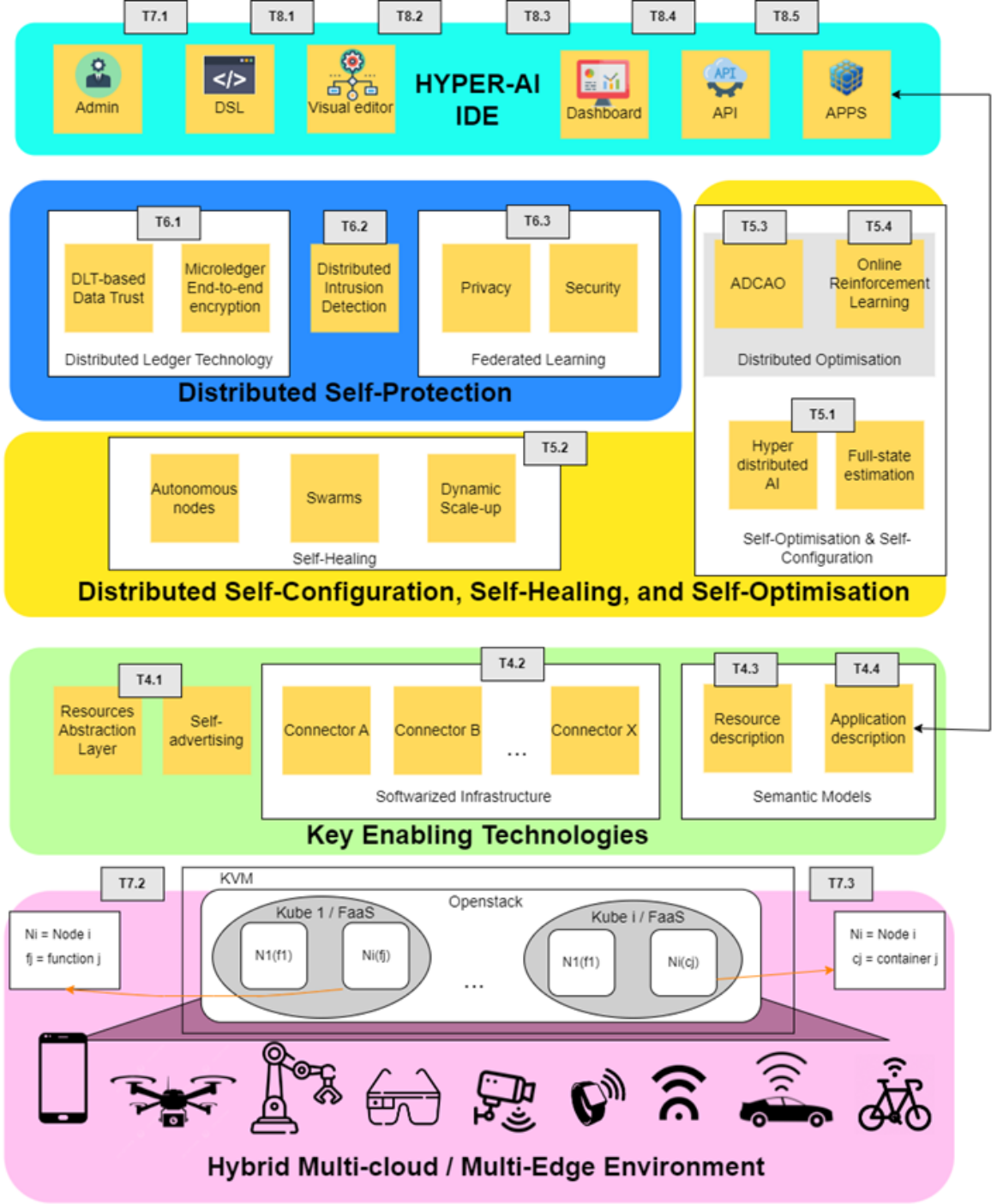
Revolutionising big data applications with autonomous cloud-to-edge resources

Key Technologies & Innovations in HYPER-AI

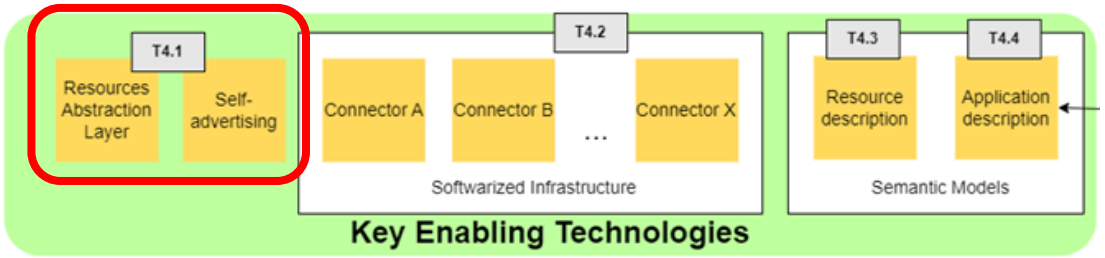
Vassilis Papataxiarhis, NKUA

1st HYPER-AI Webinar

High-Level Architecture



Resources Abstraction & Self-advertising mechanisms

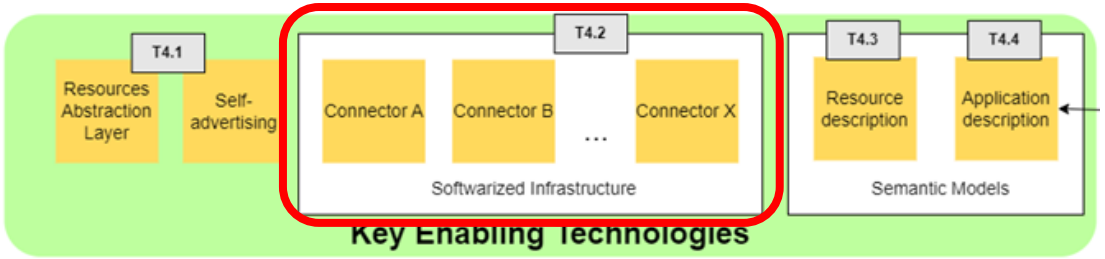


Goal: Implement a full set of functionalities for the registration and the lifecycle management of each node of the CC.

Activities

- Abstraction of the CC nodes to facilitate the identification of available assets in all layers of the infrastructure.
- Open framework to abstract the necessary computational resources (i.e., the nodes of the continuum)
- Enable simpler and faster development, deployment, operation and execution of cloud-edge applications utilizing heterogeneous resources within the continuum.
- The resources abstraction will alleviate the vendor lock-in issue and ensure interoperability.
- Implement self-advertising mechanisms to enable the discovery of nodes.
- Allow the registration of new nodes (i.e., “I am here and available to offer these capabilities”) and enable the reservation and the engagement of resources (i.e., book a new node to participate in the computing swarm, assess the level and the efficiency of collaboration).

Cognitive Cloud Softwarized Infrastructure

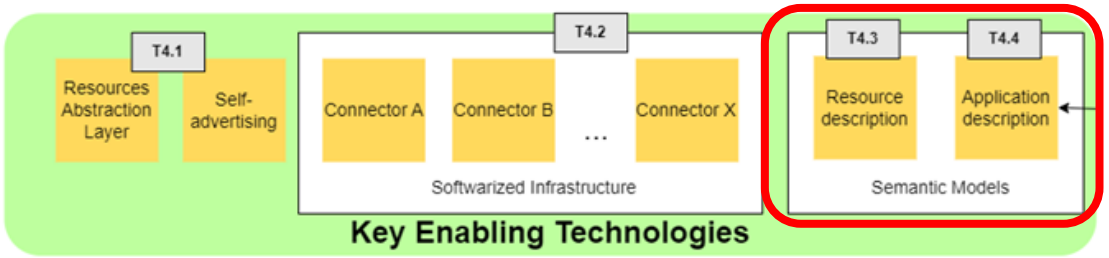


Goal: Interconnect and manage heterogeneous, multi-modal entities with various computational, storage and network capabilities in a common Computing Continuum empowered by decentralized/swarmed network intelligence.

Activities

- Develop a set of connectors that will implement the horizontal HYPER-AI functions in each one of the nodes of the continuum.
- The envisioned open connectors will be implemented on top of the nodes' operating systems (OS) to enable the transparent management and interoperation of the swarming nodes.
- The management of the computing swarms will be agnostic to the underlying hardware or software details.
- Implement open connectors for (a) a set of different cloud-edge resources including (but not limited to) cloud server components, smartphones and tablets, augmented reality equipment such as smart glasses, and (b) a set of different operating systems including among other Linux-based OS and Android.

Semantic Models

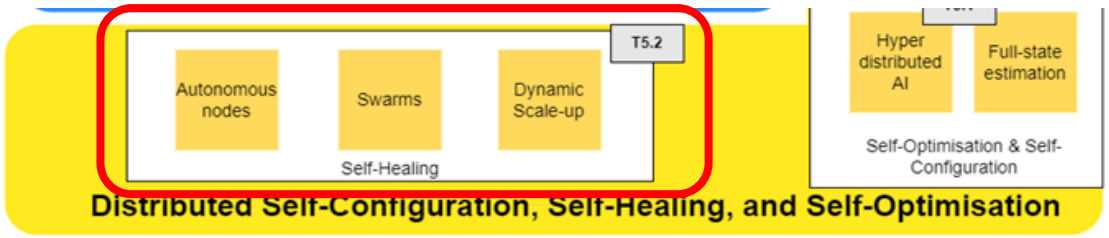


Goal: Deliver a set of open models for the semantic description of resources composing the different layers of the CC.

Activities

- Use semantic modeling to represent new resources registered to the platform.
- Design open models (ontologies) to semantically describe the capabilities of the different system resources (e.g., functional properties, CPU, RAM, GPU availability, energy costs).
- Abstract the multi-cloud/multi-edge computing infrastructures, capturing the high-level information that will enable the representation of system capacity.
- Deliver models that will facilitate the conceptual description of artifacts and software dependencies.
- Help optimise application deployments over the computing continuum.
- Build an Abstraction Component Library based on an open-source Infrastructure as a Code language such as Kubernetes Resource Model (KRM) and OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA).

Autonomous Self-managed Computing Nodes and Swarms

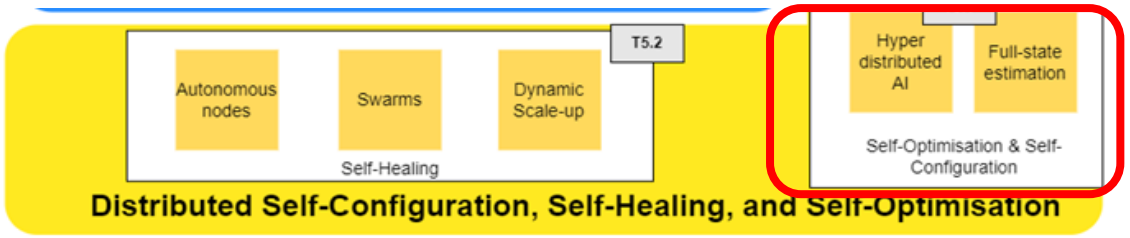


Goal: Build mechanisms to opportunistically pool/authorize resources and enable the exploitation of different parts of the network which are unused but available.

Activities

- Creation of adaptive mechanisms that allow for the opportunistic (autonomous lifecycle management of nodes) pooling of network resources.
- Development of dynamic AI-based real-time workload-batches' allocation i.e., autonomous matching of application workloads and processes to formed nodes based on network, computational, storage dynamics, as well as likelihood of fault events, security threats, cost targets, application requirements (e.g., data locality, QoS constraints, workload characterizations etc.).

Self-optimization and self-configuration

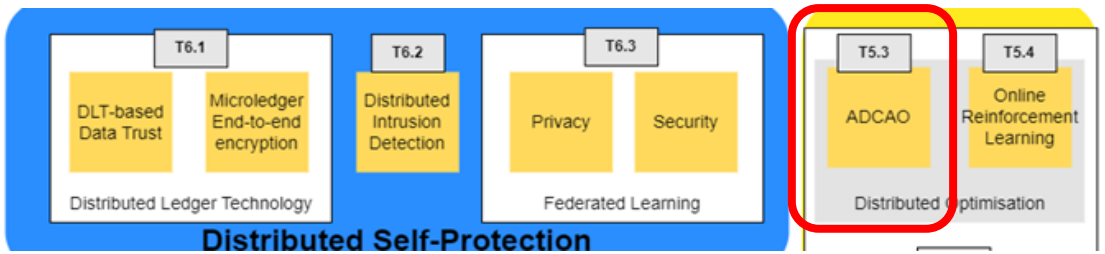


Goal: Hyper distributed resources modeling AIs for reliable full-state estimation across the computing hierarchy

Activities

- Detection of abnormal continuum/system states and enhancement of situational awareness and system full-state estimation in a non-fully-observed distributed continuum system.
- Use of advanced deep AI technologies to achieve self-awareness during the runtime phase. Development of a situation assessment and forecasting mechanism.
- Establishment of a specialized data curation mechanism to ensure the quality of operationally streamed data.

Asynchronous distributed cognitive adaptive optimization

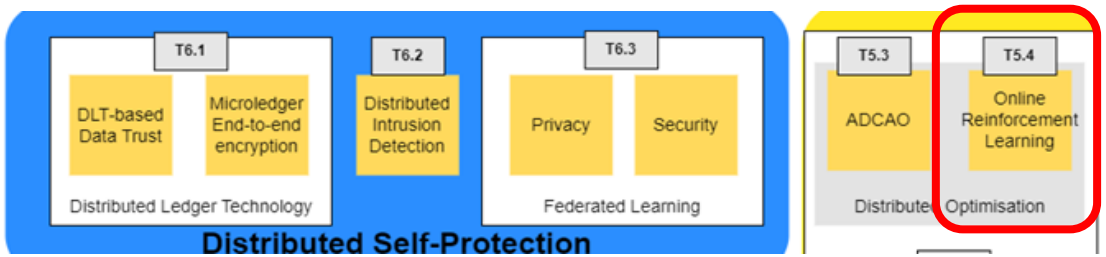


Goal: Build an asynchronous distributed optimization mechanism for real-time computing (memory, processing, storage) resources reconfiguration

Activities

- Exploit the real-time situation awareness mechanisms capabilities to optimize local policies by a cooperative multi-agent scheme.
- Each agent will have a different lifecycle (asynchronous) according to the assigned computing node management.
- Select the most suitable system configuration policy (pre-trained) agent based on the contextual situation.
- Each of these agents will be asynchronously trained using data from its dedicated lifecycle and utilization cases.
- The goal is to enhance robustness, performance, and fault tolerance across the system/continuum/network.

Distributed optimization mechanism for real-time data-related (big data locality and transmission) resources management

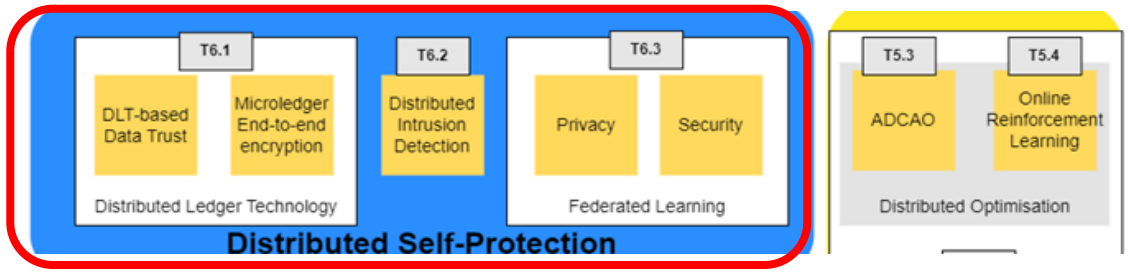


Goal: Deliver the intelligence components that will allow for the transformation of abstract task descriptions into concrete deployment plans by reasoning over the task descriptions, the relevant data and the available infrastructure

Activities

- Optimize both static and real-time, online application-data-related resources based on application needs.
- Intelligence components will be developed to convert abstract application's task descriptions into specific deployment plans.
- A collaborative hierarchical multi-agent reinforcement learning (MARL) strategy will be employed to enhance coordination among the meshed smart-nodes by T5.2.
- To address the cold-start problem associated with launching new deployments, multi-agent RL models will be trained and validated through simulations.

Distributed Self Protection

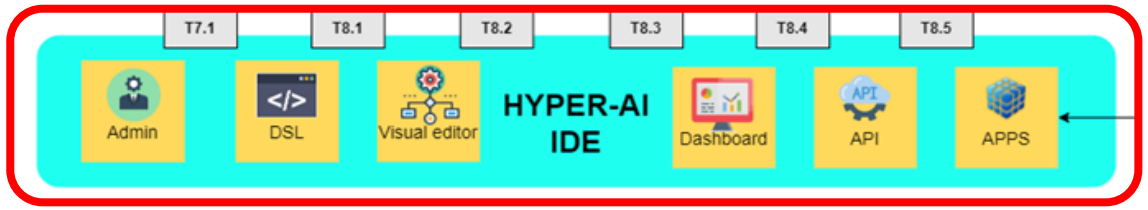


Goal: Build security, privacy and trust mechanisms for both the data and the network in a multi-edge/multi-cloud computing continuum

Activities

- Design and implementation of a decentralized data trust and security framework based on DLT to provide a transparent and secure data management solution that enhances trust in digital transactions while protecting sensitive data.
- Develop an owner-centric encryption mechanism that ensures secure data transmission and storage at the application level that provides end-to-end encryption that is owned and controlled by the data owner, thereby ensuring maximum privacy and security.
- Design and develop an AI-based distributed intrusion detection system for Cognitive Cloud Continuum architectures, providing an effective and efficient mechanism for detecting and preventing cyber-attacks on distributed systems.
- Investigate and develop solutions for privacy and security issues in Federated Learning, ensuring that sensitive data is protected during the learning process, while maintaining the accuracy of the learning models.

Application Submission IDE

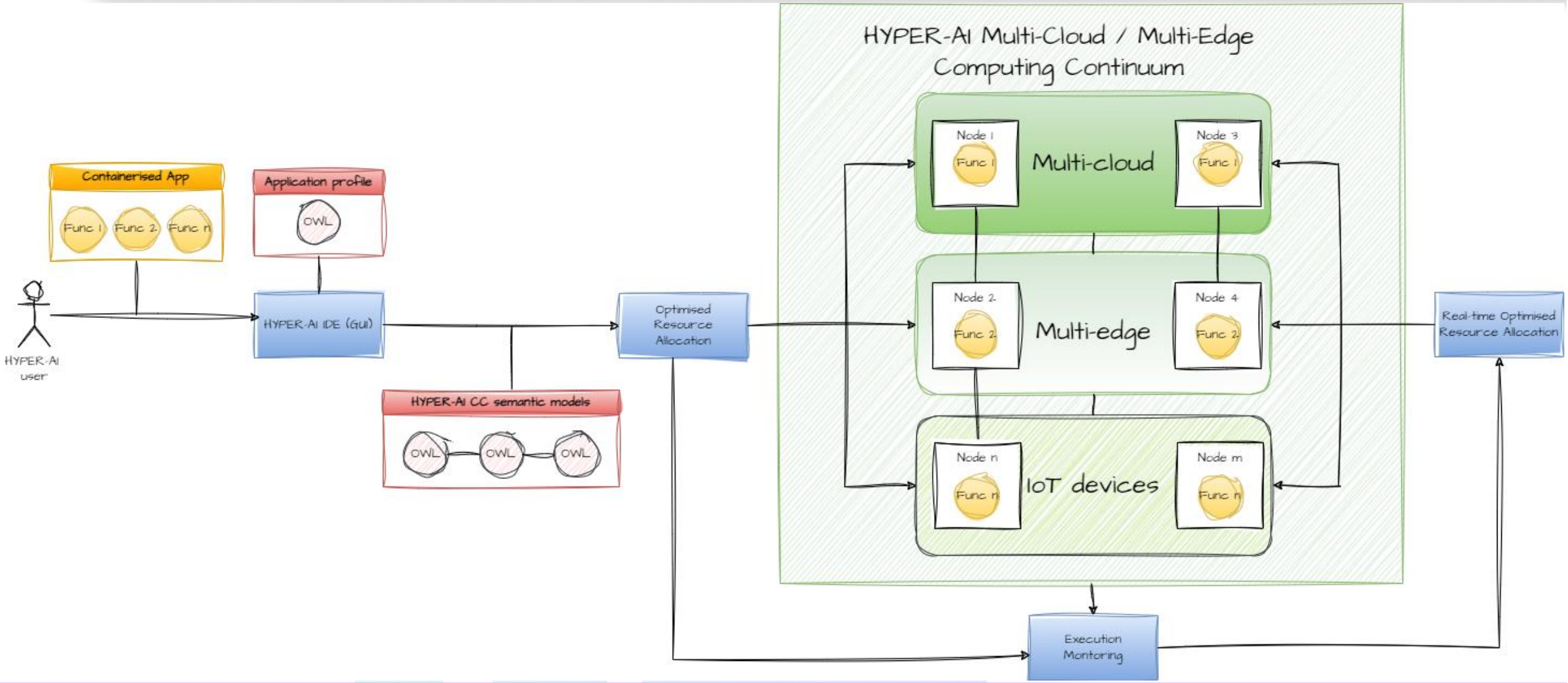


Goal: Develop an IDE to facilitate app deployment to the HYPER-AI Computing Continuum by non-experts.

Activities

- Develop abstractions for edge computing applications, capturing higher-level information that will enable the conceptual description of functional and non-functional app requirements as well as dependencies between the different software modules.
- Develop a front-end (i.e., human-machine interface, data visualisations) that will allow HYPER-AI users to specify the details of the deployment workflow and orchestrate the use of the framework.
- Tools for supporting the users through the whole lifecycle of the application (design, deployment, runtime)
- Link the semantic models for the representation of the computing continuum and the application requirements with the application deployment workflow.
- Candidate technologies => Domain Specific Language (DSL)

Example Workflow



HYPER-AI

Revolutionising big data applications with autonomous cloud-to-edge resources

Thank you