Autopoietic Cognitive Edge-cloud Services

Deliverable 4.3
Coordination methodology and library implemented as software for realizing autonomy and actionability

Grant Agreement Number: 101093126

hiro MICRODATACENTERS
inesc id lisboa
ipto
sixsq
Lakeside Labs SELF-ORGANIZING NETWORKED SYSTEMS
POLITÉCNICA
MARTEL Innovate
TECHNISCHE UNIVERSITÄT DARMSTADT
University of Ljubljana Faculty of Computer and Information Science
IDSIA SUPSI
DATAPOWER CONSULTING

Autopoietic Cognitive Edge-cloud Services	
Project full title	Autopoietic Cognitive Edge-cloud Services
Call identifier	HORIZON-CL4-2022-DATA-01
Type of action	RIA
Start date	01/01/2023
End date	31/12/2025
Grant agreement no	101093126

Funding of associated partners
The Swiss associated partners of the ACES project were funded by the Swiss State Secretariat for Education, Research and Innovation (SERI).

D2.2 – ACES Kernel Components	
Author(s)	Melanie Schranz
Editor	Melanie Schranz
Participating partners	LAKE, UL, SIXSQ, SUPSI, HIRO, MAR, UPM, TUD
Version	1.0
Status	Final
Deliverable date	M30
Dissemination Lvl	PU - Public
Official date	30/06/2025
Actual date	01/07/2025

Executive Summary

The ACES (Autonomous Cognitive Edge Systems) project integrates a suite of software components designed to enable intelligent orchestration, real-time monitoring, anomaly detection, and cognitive management across the edge-to-cloud continuum. This deliverable summarizes all the implemented and integrated software components.

At the core of its orchestration capability is the Swarm Scheduler, which uses agent-based modeling and swarm intelligence to autonomously distribute workloads across clusters, enhanced by a dual-layer design for both inter- and intra-cluster coordination. Complementing this is the Workload Actions Manager, a Kubernetes-integrated service that manages lifecycle actions such as create, move, delete, and swap operations through an asynchronous, fault-tolerant API. The ML Calibrator further enhances orchestration by leveraging Bayesian learning to self-tune the scheduler's parameters, adapting to system performance without requiring predefined models. Together, these components ensure that ACES can dynamically manage workload distribution, resource optimization, and task orchestration at scale.

The platform’s observability and anomaly detection capabilities provide continuous insight into system behavior and operational health. A robust monitoring pipeline gathers telemetry across workloads, infrastructure, and applications, integrating tools like OpenTelemetry, Prometheus, NATS Jetstream, and Grafana to deliver real-time visibility and informed decision-making. Machine learning-based anomaly detection models operate at both intra-service and inter-service levels, identifying local disruptions such as resource spikes and broader system anomalies like communication failures or cascading faults. An advanced Security Risk Alert system employs a spatiotemporal CNN model to detect cyber threats by transforming telemetry data into visual representations, ensuring prompt attack identification. Supporting these functions is a dedicated metrics management pipeline and a Cognitive Framework that centralizes machine learning data and model handling, enabling consistent training, storage, and retrieval of AI artifacts essential for autonomous system operation.

Disclaimer

The ACES consortium consists of the following partners:

No	PARTNER ORGANISATION NAME	ABBREVIATION	COUNTRY
1	INSTITUTO DE ENGENHARIA DE SISTEMAS E COMPUTADORES, INVESTIGACAO E DESENVOLVIMENTO EM LISBOA	INESC ID	PT
2	HIRO MICRODATACENTERS B.V	HIRO	NL
3	TECHNISCHE UNIVERSITAT DARMSTADT	TUD	DE
4	LAKESIDE LABS GMBH	LAKE	AT
5	UNIVERZA V LJUBLJANI	UL	SI
6	UNIVERSIDAD POLITECNICA DE MADRID	UPM	ES
7	MARTEL GMBH	MAR	CH
8	SCUOLA UNIVERSITARIA PROFESSIONALE DELLA SVIZZERA ITALIANA	SUPSI	CH
9	INDIPENDENT POWER TRANSMISSION OPERATOR SA	IPTO	EL
10	DATAPOWER SRL	DP	IT
11	SIXSQ SA	SIXSQ	CH

Document Revision History

DATE	VERSION	DESCRIPTION	CONTRIBUTIONS
02/06/2025	0.1	Table of contents	LAKE, HIRO
18/06/2025	0.2	Partner Input	LAKE, HIRO, UL, UPM, TUD, SUPSI, MARTEL
24/06/2025	0.3	Finalization of inputs	LAKE
30/06/2025	0.4	Internal review	UL, SUPSI
30/06/2025	0.5	Finalization	LAKE
30/06/2025	1.0	Final version after project coordination review	INESC

Authors

AUTHOR	PARTNER
Melanie Schranz	LAKE
Maksimilian Mamiliaev	HIRO
Rui Min	HIRO
Thien Nguyen	TUD
Lorris Cannelli	SUPSI
Vito Cianchini	MARTEL
Gregor Gabrovsek	UL
Aymen Yahyaoui	UPM

Reviewers

NAME	ORGANISATION
Matjaz Juric	UL
Loris Cannelli	SUPSI

List of terms and abbreviations

ABBREVIATION	DESCRIPTION
ACES	Autopoiesis Cognitive Edge-cloud Services
AI	Artificial intelligence
API	Application programming interface
CF	Cognitive framework
CNI	Container network interface
COTS	Commercial off-the-shelf
CPU	Central processing unit
CQRS	Command and query responsibility segregation
CRDT	Conflict-free replicated data type
CRI	Container runtime interface
DB	Database
DDoS	Distributed denial-of-service
DKB	Distributed knowledge base
DKBRS	Distributed knowledge base replication service
EC	European Commission
EMDC	Edge micro data centre
GPU	Graphics processing unit
HTTP	Hypertext transfer protocol
IaC	Infrastructure as code
ID	Identifier
JSON	JavaScript object notation

MitM	Man-in-the-middle attack
ML	Machine learning
NIC	Network interface controller
NIDS	Network intrusion detection system
RAM	Random access memory
REST	Representational state transfer
RIA	Research and Innovations Action
RPP	Rapprochement protocol
SDN	Software-defined networking
SERI	Swiss State Secretariat for Education, Research and Innovation
UI	User interface
WP	Work package
YAML	YAML Ain't Markup Language

Table of contents

1 Orchestrator

1.1 Swarm Scheduler

The Swarm Scheduler component in ACES uses agent-based modeling and swarm intelligence for autonomous workload distribution and scheduling across the edge continuum. Its dual-layer design enables inter- and intra-cluster orchestration. Developed with Python's MESA and Go, it integrates with Kubernetes and message queues to ensure efficient, concurrent workload management.

Details: D4.5 – Distributed knowledge base and data management systems

ACES Git Link:
Simulated scheduler: https://github.com/ACES-EU/emergent-scheduler
Integrated scheduler: https://github.com/ACES-EU/resource-management-service

1.2 Workload action manager

The Workload Actions Manager (WAM) is a Kubernetes-integrated service that orchestrates lifecycle operations for Smart AIOps entities via a lightweight, fault-tolerant API. It accepts create, delete, move, and swap directives from both centralized reschedulers and decentralized agents, executing them in parallel and asynchronously through existing Kubernetes controllers and custom scheduler plugins. High availability is ensured by running multiple replicas behind a single Service, while action requests and completion notifications flow through a persistent message queue to guarantee delivery and state tracking.

Details: D4.1 - Goal representations corresponding to SLAs and SLOs for use in AI-/ML-/swarm-based methodology realising autonomy and actionability, D4.2 - Action language and library

ACES Git Link:
Workload Actions Manager: https://github.com/ACES-EU/workload-actions-manager

1.3 Tuning swarm scheduler parameter ML model

The ML Calibrator component in ACES leverages Bayesian learning to iteratively tune the hyperparameters of the swarm scheduler, based on observed system performance. Designed as a decentralized, self-learning module inspired by autopoiesis, it adapts without relying on predefined analytical models. Implemented in Python, it is fully integrated within the Kubernetes environment to support scalable and autonomous calibration at the edge.

Details: D4.5 – Distributed knowledge base and data management systems

ACES Git Link:
Simulated ML calibrator: https://github.com/ACES-EU/Self-hyperparameter-tuning
Integrated ML calibrator: https://github.com/ACES-EU/Autopoietic-Emergent-Scheduler

2 Monitoring

2.1 Monitoring metric collection

The Monitoring and Observability component in ACES provides end-to-end telemetry collection and real-time system visibility across the edge-to-cloud continuum. It enables proactive anomaly detection and informed decision-making by gathering and analyzing metrics, logs, and traces from workloads, infrastructure, and applications. The architecture integrates several open-source tools, including OpenTelemetry for instrumentation, Prometheus for metrics collection and alerting, NATS Jetstream for telemetry streaming, and Grafana for visualization. The component ensures telemetry is not only collected but also aggregated, analysed, and made available to other ACES modules via event streaming.

Details: D3.3 - Cross-level inter-agent knowledge model

ACES Git Link:
Monitoring and Observability component: https://github.com/ACES-EU/kc-monitoring-observability

3 Anomaly detection

Traditional monitoring technologies frequently miss small irregularities or new risks when microservice architectures become more complicated. The ACES platform incorporates machine learning-based elements to identify security threats and performance problems in real time within Kubernetes systems. Two parallel machine learning anomaly detection pipelines are implemented to catch various scopes of anomalous behavior. The intra-service and the inter-service ADCs. When combined, these models offer a multifaceted viewpoint on anomalies that include both localized and system-wide behavioral changes. To detect cyberattacks, the Security Risk Alert module combines these elements with a spatiotemporal CNN that transforms logs, system metrics, and traffic data into RGB-like visuals. Constructed using PyTorch and included into the ACES pipeline, it guarantees timely notifications and facilitates secure, robust operations.

3.1 Intra-Service Anomaly Detection Component (ML model)

To find anomalous activity within specific microservices, the Intra-Service Anomaly Detection methodology makes use of resource consumption indicators, including CPU utilization, memory allocation, and service uptime. The model can identify early warning indications of local problems such as CPU spikes, memory leaks, or unplanned service breakdowns by consistently tracking their metrics. It uses machine learning methods that are lightweight, optimized for real-time analysis, and have low computational overhead, which makes it ideal for deployment in contexts with limited resources, such as Kubernetes clusters.

ACES Git Link:

Anomaly Detection intra-service component: https://github.com/ACES-EU/Anomaly_Detection_ML

3.2 Inter-service Anomaly Detection Component (ML model)

Finding communication anomalies between microservices that can point to more extensive systemic or cascading problems is the main goal of the Inter-Service Anomaly Detection component. In order to identify trends that either precede or reflect major system instability, it keeps an eye on important interaction metrics, including increasing request latency, failed service dependencies, and abrupt changes in service topology. A dynamic, contextual understanding of microservice interactions over time is made possible by Temporal Knowledge Graphs (TKGs) and Graph Neural Networks (GNNs), upon which this component is based. The system can accurately discover complex anomalies that traditional point-based monitoring frequently misses by capturing the temporal behavior and structure of service communication through the modeling of service interactions as changing graphs.

ACES Git Link:

Anomaly Detection inter-service component: https://github.com/ACES-EU/TKG_Anomaly_detection_v1

3.3 Security risk alert (ML model)

The Security Risk Alert component in ACES is designed to detect potential threats against the ACES system. It utilizes our innovative Spatiotemporal CNN model for attack detection. Inspired by 3D image classification techniques, our approach transforms network traffic data, system metrics, and Kubernetes log into RGB image representations, enabling precise detection of cyberattacks. Developed using Python and the PyTorch framework, it will be integrated with Kubernetes and message queues

to ensure that any malicious activities against the ACES system are promptly detected and alerted for further countermeasures.

Details: D4.4 – AI/ML Algorithms for non-functional aspects support.

ACES Git Link:
Intrusion Detection System: https://github.com/ACES-EU/KubeFocus

4 Metric collection

4.1 Collect the services' metric

The ACES platform features a metrics collection pipeline tailored for service-level observability across edge-cloud deployments. Central to this is the Metrics Consumer, a Python-based component that ingests telemetry data from the Pull-Push Metrics Pipeline via NATS. The architecture cleanly separates dynamic time series metrics, stored in TimescaleDB, from static infrastructure metadata, which is persisted in Neo4j, ensuring semantic consistency with the ACES Data Model. To support efficient data lifecycle management, a Prefect-based ETL workflow periodically offloads historical metrics to MinIO object storage, reducing database load while preserving long-term analytical value. This design enables scalable, real-time monitoring of services, and forms a critical foundation for perception, orchestration, and autonomous decision-making across the ACES platform.

Details: D3.4 - Multi-layer Perception

ACES Git Link:
Metric management component: https://github.com/ACES-EU/EMDC-Metrics-Management

5 Cognitive Framework

5.1 Lifecycle Model Management

The Cognitive Framework component in the ACES project provides a set of functionalities for services that are based on machine learning in order to ease and unify the general approach. Designed as a gateway it ensures that read/write operations on data needed for model learning and model predictions are centralized and executed via itself. It means that the CF provides an API in order to read and write data both from Message Broker and data bases. Another pivotal role of the CF is being a unified storage for machine learning models. This includes retrieving already pretrained models based on the goal as well as storing models that have been trained by other services outside the CF.

Details: D4.5 – Distributed knowledge base and data management systems

ACES Git Link:
Cognitive Framework: https://github.com/ACES-EU/Cognitive-Framework

6. Conclusion

This deliverable summarized all the implemented and integrated ACES software components designed to enable intelligent orchestration, real-time monitoring, anomaly detection, and cognitive management across the edge-to-cloud continuum.