netidee PROJEKTE

LEO Trek

Interim Report | Call 19 | Project ID 7442

Content

1 Introduction 3

2 Status of Work Packages 3

2.1 AP1 – *Project Management* 4
2.2 AP2 – *Stardust Simulator* 4
2.2.1 Stardust: A Scalable and Extensible Simulator for the 3D Continuum 4
2.3 AP3 – *LEO Trek Scheduler* 5
2.3.1 HyperDrive – Serverless Workflow Scheduler for the 3D Continuum 5
2.3.2 ChunkFunc – SLO- and Input-aware Resource Optimizer for Serverless Workflows 6
2.4 AP4 – *LEO Trek Platform* 7
2.4.1 FedCCL: Federated Clustered Continual Learning Framework for Privacy-focused Energy 7
2.4.2 Cosmos: A Cost Model for Serverless Workflows in the 3D Compute Continuum 8
2.5 AP5 – *Documentation & Dissemination* 9

3 Implementation of Funding Conditions. 9

4 Summary of Plan Update 9

5 Dissemination and Networking 10

6 References 10

1 Introduction

All the objectives planned for the first half of the project (this report's reported period) have been successfully achieved and the objectives set for the second half of the project remain valid and realistic. Table 1 provides an overview of the key project output and the goals. The target number of academic publications for excellent success (five) has already been reached. Overall work is going according to plan.

Table 1: Project KPIs Overview

KPI	Current State	Goal
Academic publications	5	Minimum: 3 Ideally: 5
Open-source software projects	2	3
Blog posts	4	8

2 Status of Work Packages

The work packages are proceeding according to plan. Table 2 lists the academic publications made over the first half of the project and Table 3 lists the open-source software projects.

Table 2: LEO Trek Academic Publications

No.	AP	Publication
1	AP2	T. Pusztai, J. Hisberger, C. Marcelino, and S. Nastic, "Stardust: A Scalable and Extensible Simulator for the 3D Continuum," in 2025 IEEE International Conference on Edge Computing and Communications (EDGE), 2025.
2	AP3	T. Pusztai, C. Marcelino, and S. Nastic, "HyperDrive: Scheduling Serverless Functions in the Edge-Cloud-Space 3D Continuum," in 2024 IEEE/ACM Symposium on Edge Computing (SEC), 2024.
3	AP3	T. Pusztai and S. Nastic, "ChunkFunc: Dynamic SLO-aware Configuration of Serverless Functions," IEEE Transactions on Parallel and Distributed Systems, 2025.
4	AP4	M. Helcig and S. Nastic, "FedCCL: Federated Clustered Continual Learning Framework for Privacy-focused Energy Forecasting," in The 9th IEEE International Conference on Fog and Edge Computing (ICFEC), 2025.
5	AP4	C. Marcelino, S. Gollhofer-Berger, T. Pusztai, and S. Nastic, "Cosmos: A Cost Model for Serverless Workflows in the 3D Compute Continuum," in 2025 IEEE International Conference on Smart Computing (SMARTCOMP), 2025.

Table 3: LEO Trek Published Open-source Software

No.	AP	Software	URL
1	AP2	Stardust 3D Continuum Simulator	https://github.com/polaris-slo-cloud/stardust
2	AP3	HyperDrive Serverless Scheduler	https://github.com/polaris-slo-cloud/hyper-drive
3	AP3	ChunkFunc Serverless Workflow Optimizer	https://github.com/polaris-slo-cloud/chunk-func
4	AP4	FedCCL Federated Learning Framework	https://github.com/polaris-slo-cloud/fedccl

2.1 AP1 – *Project Management*

The project is going according to plan with no significant incidents or deviations from the work plan. The team has shown high ambition and produced substantial results, as detailed in the subsequent AP results. There were no noteworthy issues or delays. The project costs adhere to the plan.

There has been a change to the team: Thomas Pusztai's contract at the TU Wien expired on August 31, 2025 and thus, he had to leave the project team. Since he had to use up his remaining vacation, his last working day was July 30. His duties have been taken over by Cynthia Marcelino.

2.2 AP2 – *Stardust Simulator*

This work package has been carried out and concluded according to plan. It resulted in one academic publication and one open-source software artifact.

2.2.1 Stardust: A Scalable and Extensible Simulator for the 3D Continuum

Low Earth Orbit (LEO) mega constellations provide low latency communication between LEO and terrestrial nodes and among terrestrial nodes, extending the Edge-Cloud Continuum into an Edge-Cloud-Space 3D Continuum. Developing orchestration services and applications for the 3D Continuum, such as RapidREC, requires realistic simulations of the highly dynamic network conditions and node locations inherent to this environment. Unfortunately, existing simulators only allow for relatively small constellations to be simulated without scaling to a large number of host machines. *Stardust* is a scalable and extensible open-source simulator for the 3D Continuum. Our main contributions are:

1. *Stardust, a scalable and extensible next generation simulator* for the 3D Continuum with support for simulating LEO-, Cloud-, and Edge nodes in a scalable manner. Stardust enables experiments for evaluating networking and orchestration algorithms for the 3D Continuum.

It supports simulating mega constellations three times the size of the currently largest constellation, with almost 7k satellites on a single machine.

2. *A dynamic routing mechanism* that enables experimentation with different routing mechanisms by making the ISL routing protocol and the network path computation changeable. This allows, e.g., changing the default +Grid ISL routing to a different protocol or to introduce caching or hypergraph algorithms as a replacement for Dijkstra's algorithm to calculate node-to-node network paths.
3. *SimPlugin, a plugin mechanism* that serves as the integration point for custom logic that Stardust should execute at every step of the simulation. A SimPlugin has access to the complete infrastructure state and, thus, allows integrating, e.g., orchestration algorithms/software that should be evaluated using Stardust.

2.3 AP3 – *LEO Trek Scheduler*

This work package has been carried out and concluded according to plan. It resulted in the publication of two academic papers and two software artifacts.

2.3.1 HyperDrive – Serverless Workflow Scheduler for the 3D Continuum

HyperDrive is a platform and network Service Level Objective-aware scheduler for serverless functions in the 3D continuum. The 3D continuum expands the Edge-Cloud continuum to include low earth orbit (LEO) satellites. These satellites have enormously grown in number in the recent years and are projected to provide valuable compute resources, especially for Earth Observation (EO) data from satellites by avoiding unnecessary downlinking of the massive amounts of data. Satellite EO data can be used to survey the region around an accident to assess the state of the road network, predict congestion, and devise a plan for faster recovery. The contributions of HyperDrive include:

- A novel Serverless Platform that introduces novel components and mechanisms tailored to the unique characteristics of the 3D Continuum. HyperDrive enables functions to be seamlessly executed anywhere in the 3D Continuum, optimizing performance and reliability.
- A Serverless scheduler for the 3D Continuum that considers constraints such as resource capacity, application SLO requirements, and network load to minimize the end-to-end Serverless workflow latency. The HyperDrive scheduler, also considers satellite position and thermal conditions to enable function scheduling in the 3D Continuum. By considering edge, cloud, and space conditions, HyperDrive executes functions that meet every SLO requirement in the 3D Computing Continuum. HyperDrive achieves 71% lower end-to-end (E2E) network latency than the next best baseline approach.

The HyperDrive scheduler is designed to address the challenges that arise in the placement of serverless functions in the 3D Continuum using an optimization problem (see academic paper for this) and using a Multi Criteria Decision Making (MCDM) approach. The MCDM approach considers the vicinity of candidate nodes to the node of the previous function and source data, the resources

of the candidate node, the network SLOs, and the maximum allowed operating temperature of the candidate node, if it is a satellite.

2.3.2 *ChunkFunc – SLO- and Input-aware Resource Optimizer for Serverless Workflows*

ChunkFunc is a resource optimizer for serverless workflows. It assigns resource profiles to a serverless workflow's functions to ensure that the response time Service Level Objective (SLO) of the workflow is met, while minimizing costs. Unlike much of the state-of-the-art, ChunkFunc considers the size of the input data of a function when assigning resources. This ensures SLO compliance when the input is larger than average and saves costs when the input is smaller than average. This approach benefits applications with highly diverse input sizes, such as traffic analysis systems. During rush hour, the input to a periodically executed accident detection workflow is larger than average and during night, the input is smaller than average. ChunkFunc's contributions include:

- An SLO- and input data size-aware function performance model for determining optimized configurations in serverless workflows, depending on the input data size.
- ChunkFunc Profiler, which automatically builds performance models for serverless functions and workflows based on typical input data sizes. Profiling is automatic, users only deploy a function and specify typical input data. A novel, auto-tuned Bayesian Optimization approach reduces the profiling costs by up to 90% compared to exhaustive profiling and ensures high accuracy of the results.
- ChunkFunc Workflow Optimizer, which leverages various heuristics to dynamically adapt the resource configuration of functions in a workflow to meet a performance-based SLO (e.g., response time), while minimizing cost. Depending on the workflow it increases SLO adherence by a factor of 1.12 to 2.0 and reduces costs by up to 53% The Workflow Optimizer is extensible with arbitrary performance-based SLOs.

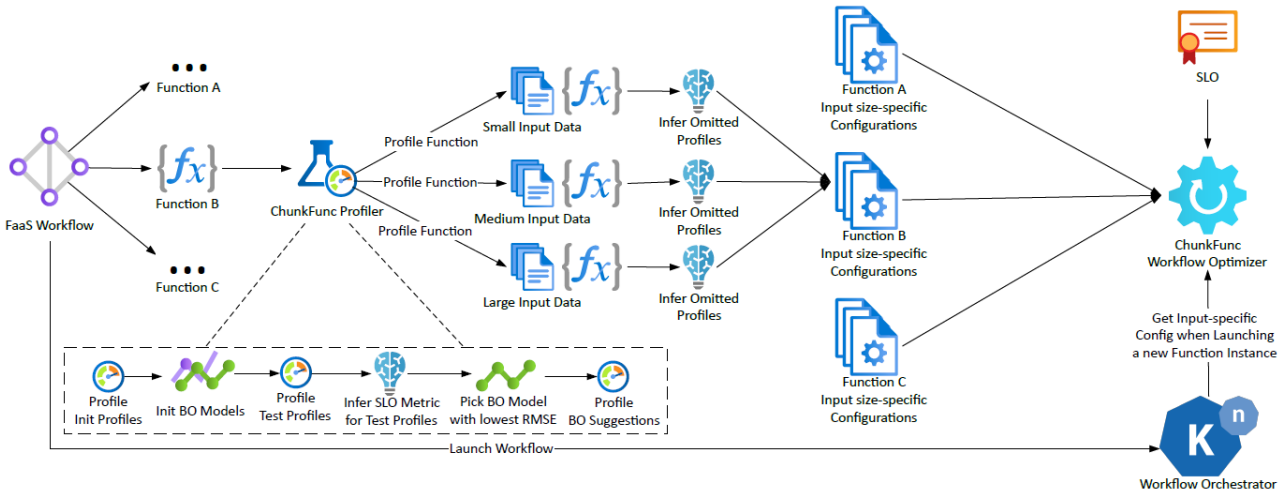

Figure 1: Overview of the ChunkFunc System and Lifecycle of a Serverless Workflow.

The ChunkFunc framework consists of two major components: The *Profiler* and the *Workflow Optimizer*. Figure 1 presents an overview of ChunkFunc and the lifecycle of a serverless workflow within the system. Upon their deployment, serverless functions are automatically picked up by the

ChunkFunc Profiler. It deploys function instances using various resource configurations to execute profiling runs with their typical input data sizes, without any user interaction. To reduce the number of profiling runs, while maintaining a high accuracy of the results, the choice of resource configurations is guided by Bayesian Optimization. Our BO Dynamic Hyperparameter Selection picks the hyperparameter that yields the most accurate results for a particular function type and input size combination. Finally, the input-specific performance profiles are leveraged by the ChunkFunc Workflow Optimizer, which provides a suitable resource profile, to meet the workflow's SLO and minimize cost, to the serverless orchestrator prior to invoking a function.

2.4 AP4 – *LEO Trek Platform*

This work package has started in June 2025 and is progressing as planned. It has so far resulted in the publication of two academic papers and one software artifact.

2.4.1 FedCCL: Federated Clustered Continual Learning Framework for Privacy-focused Energy

Privacy-preserving distributed model training is crucial for modern machine learning applications, yet existing Federated Learning approaches struggle with heterogeneous data distributions and varying computational capabilities. Traditional solutions either treat all participants uniformly or require costly dynamic clustering during training, leading to reduced efficiency and delayed model specialization. FedCCL (Federated Clustered Continual Learning) is a framework that addresses these challenges through a combination of pre-training clustering and asynchronous Federated Learning. Unlike most of the existing approaches that perform clustering during or after training [1] [2], FedCCL employs DBSCAN clustering based on static characteristics before training begins. This approach enables immediate model specialization while reducing coordination overhead. Furthermore, participants can belong to multiple clusters simultaneously, facilitating more nuanced knowledge sharing than strict partitioning approaches [3]. FedCCL's main contributions include:

- *FedCCL Framework:* A Federated Learning framework that integrates clustered pre-training with an enhanced asynchronous FedAvg algorithm. The framework operates through a two-phase approach, initially clustering clients based on their inherent system properties before training, followed by client-driven updates with model locking during aggregation. Mitigating the performance degradation typically seen in asynchronous Federated Learning with heterogeneous data while maintaining reduced overhead.
- *FedCCL Predict & Evolve:* Through our system property-based clustering approach, FedCCL creates a framework that provides a specialized model for newly joining clients without requiring prior exposure to their specific data distributions. In the *Predict* phase, new clients can immediately benefit from these highly specialized models to generate predictions. As clients begin contributing their own data, they enter the *Evolve* phase, where they participate in training and refining cluster-specific models. Our evaluation demonstrates this capability through robust generalization metrics, where models achieve nearly identical performance

levels for both training and independent populations, with mean error rates showing minimal degradation of only 0.14 percentage points for new installations.

2.4.2 Cosmos: A Cost Model for Serverless Workflows in the 3D Compute Continuum

To allow evaluating costs for serverless deployments in the 3D (Edge-Cloud-Space) Continuum efficiently, *Cosmos* constitutes a novel cost- and a performance-cost-tradeoff model for serverless workflows that identifies key factors that affect cost changes across different workloads and cloud providers. Common approaches for serverless cost estimation include: (a) Predictions [4, 5, 6] use models, such as ML and math models to estimate costs based on historical execution data. This enables the estimation and analysis of costs without executing or even deploying a workflow. However, these high-level predictions often fail to provide detailed cost breakdowns or to identify the main drivers of higher expenses. (b) Simulations [7, 8, 9] enable users to explore how costs behave under different parameter configurations. They offer valuable insights into performance and expenses across various workload patterns, highlighting important trade-offs. However, existing simulation tools often lack fine-grained parameters to identify which aspects contribute to higher costs.

Since current cost models are not detailed enough for precise performance-cost tradeoff decisions, users often err on the side of caution and incur higher costs to ensure performance. The Cosmos cost model enables the building of intelligent frameworks to optimize serverless costs and maximize performance. Our main contributions include:

- *Cosmos: A cost and a performance-cost tradeoff model for serverless workflows* that incorporates the heterogeneity and dynamic characteristics of the 3D Continuum. Cosmos isolates the main cost drivers while accounting for their interdependencies, providing an understanding of how different factors impact execution and cost, e.g., resource constraints, workload characteristics, communication overhead, and dynamic pricing.
- *A cost taxonomy that classifies the main cost drivers*, enabling their identification among invocation, compute, data transfer, state management, and BaaS. This provides insights into specific cost drivers for serverless workflows across the different layers of the 3D Continuum.

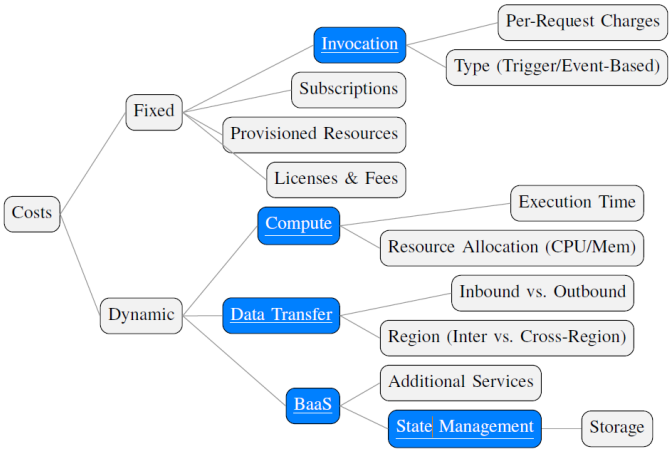

Figure 2: Serverless workflow costs drivers, highlighting key cost drivers: Invocation, Compute, Data Transfer, and State Management (partial view).

Figure 2 presents a taxonomy of the main cost drivers associated with serverless workflows, highlighting the focus of this analysis: Invocation, Compute, Data Transfer, and State Management. The main cost drivers are directly tied to the execution and performance of serverless functions, representing the most variable and impactful cost components in typical serverless workflows. Unlike some fixed costs, such as subscriptions and provisioned resources, which remain constant regardless of usage, the underlined drivers exhibit cost fluctuations based on function activity, data flows, and resource consumption.

2.5 **AP5 –** ***Documentation & Dissemination***

This work package is proceeding according to plan. Every software artifact includes documentation in its open-source repository. Additionally, each repository is accompanied by an academic publication.

3 Implementation of Funding Conditions.

No special conditions were defined for this project.

4 Summary of Plan Update

Overall, work is going according to plan. *AP2 Stardust Simulator* and *AP3 LEO Trek Scheduler* have been completed as scheduled and *AP4 LEO Trek Platform* has been started. Project results *7 Stardust Simulator for the 3D Continuum* and *8 LEO Trek Serverless Scheduler for the 3D Continuum* have been successfully completed.

The planned end date of the project at the end of November 2025 remains feasible.

5 Dissemination and Networking

All of LEO Trek's innovations were published in academic papers. Most of them (4 out of 5) were conference papers and were presented to the audience at the respective conferences. This has sparked conversations with the attendees and plans for a Horizon Europe research project in the near future.

6 References

[1] F. Sattler, K.-R. Müller und W. Samek, „Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE Transactions on Neural Networks and Learning Systems,* Bd. 32, Nr. 8, p. 3710–3722, 2021.

[2] A. Ghosh, J. Chung, D. Yin und K. Ramchandran, „An Efficient Framework for Clustered Federated Learning," *IEEE Transactions on Information Theory,* Bd. 68, Nr. 12, p. 8076–8091, 2022.

[3] E. Yoo, H. Ko und S. Pack, „Fuzzy Clustered Federated Learning Algorithm for Solar Power Generation Forecasting," *IEEE Transactions on Emerging Topics in Computing,* Bd. 10, Nr. 4, p. 2092–2098, 2022.

[4] S. Eismann, J. Grohmann, E. van Eyk, N. Herbst und S. Kounev, „Predicting the Costs of Serverless Workflows," New York, NY, USA, Association for Computing Machinery, 2020, p. 265–276.

[5] R. Cordingly, W. Shu und W. J. Lloyd, „Predicting Performance and Cost of Serverless Computing Functions with SAAF," 2020, p. 640–649.

[6] C. Lin und H. Khazaei, „Modeling and Optimization of Performance and Cost of Serverless Applications," *IEEE Transactions on Parallel and Distributed Systems,* Bd. 32, Nr. 3, p. 615–632, 2021.

[7] L. C. Da Silva, R. Medeiros und N. Rosa, „COSTA: A cost-driven solution for migrating applications in multi-cloud environments," 2023, p. 57–63.

[8] F. Liu und Y. Niu, „Demystifying the Cost of Serverless Computing: Towards a Win-Win Deal," *IEEE Transactions on Parallel and Distributed Systems,* Bd. 35, Nr. 1, p. 59–72, 2024.

[9] P. Garcia Lopez, A. Slominski, B. Metzler, M. Berhendt und S. Shillaker, „Serverless End Game: Disaggregation enabling Transparency," 2024, p. 9–14.