Emerging Paradigms for a Sustainable Next-Generation Computing Landscape

Asst. Prof. Stefan Nastic

Distributed Systems Group,

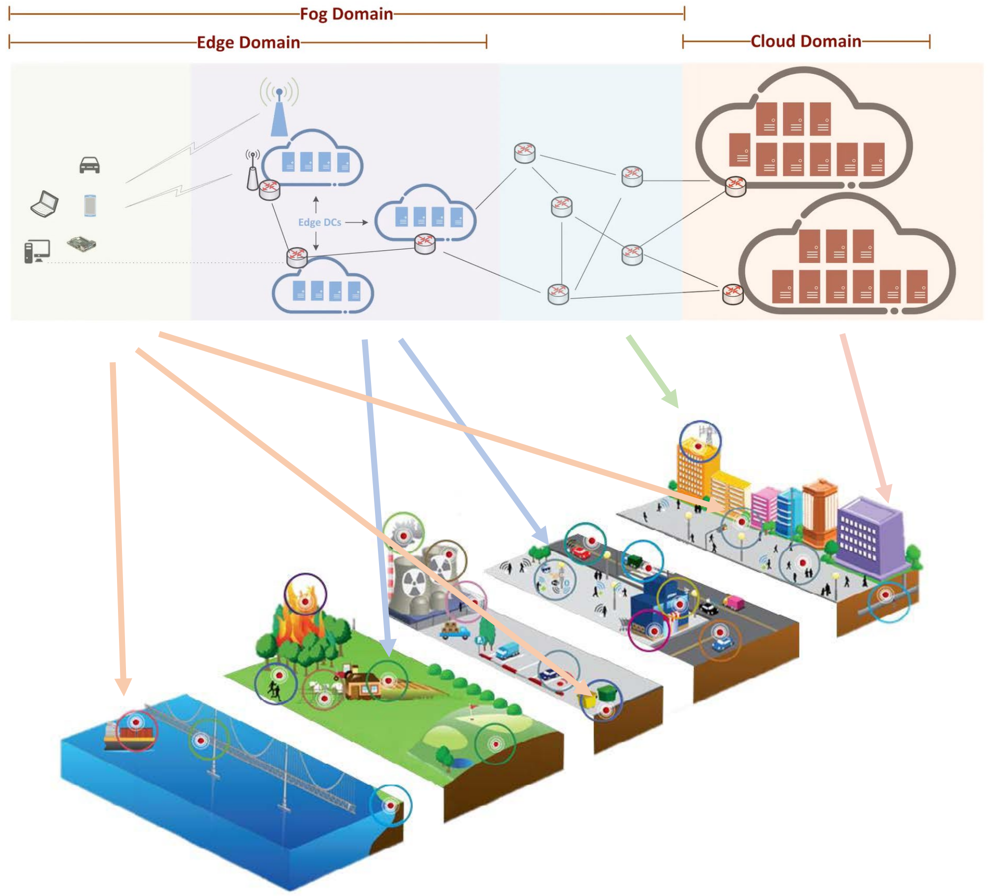
TU Wien Informatics

Public Lecture Series: Sustainability in Computer Science 2025

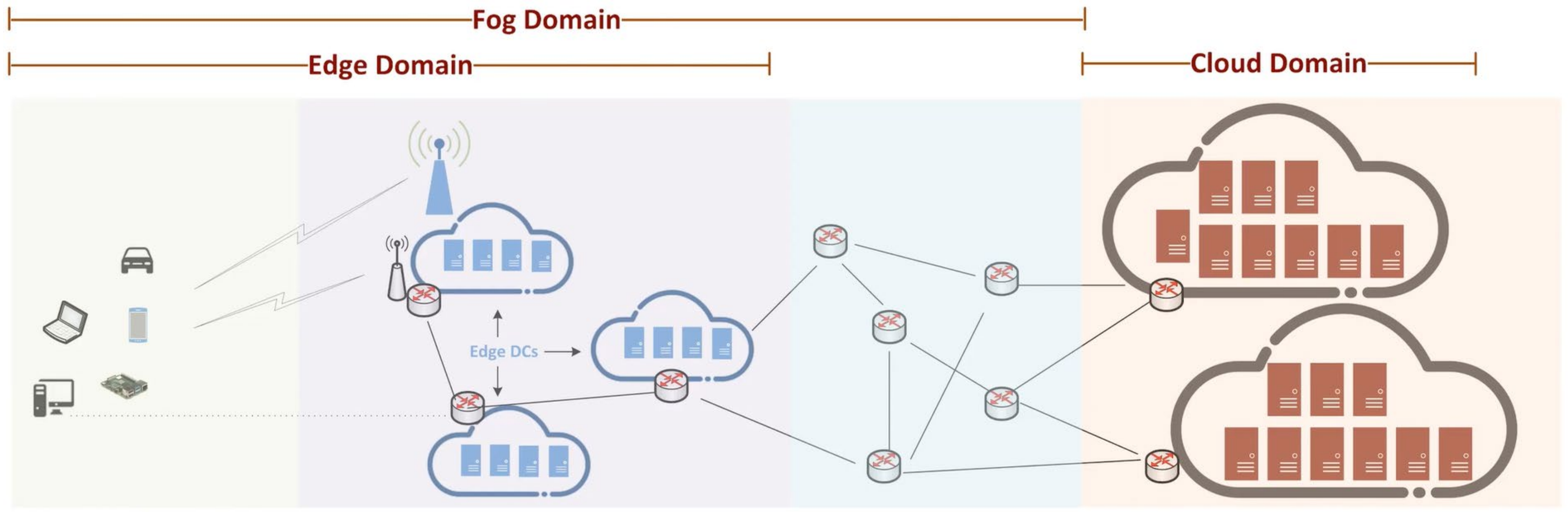
December 15th, 2025

Computing systems

- Distributed computing systems are at the heart of modern society
- They underpin our critical infrastructure and applications:
  - Smart Cities
  - Autonomous vehicles
  - Healthcare
  - Disaster Recovery
- Keeping pace with the demands of modern society is challenging:
  - Performance/Reliability/Scalability
  - Cost effectiveness
  - Energy efficiency
  - CO2 emission
  - Computing sustainability
  - …



Current computing landscape



End devices, IoT | **Edge devices, MAC** | **Core network, Internet** | **Cloud, Data centers**

End devices, IoT

Very limited compute
Low reliability
Volatility
Battery constraints
BUT cheap!

Edge devices, MAC

Edge of the network
Low latency to end devices
General-purpose compute nodes (but resource-limited)
Some redundancy
Higher cost of infrastructure

Core network, Internet

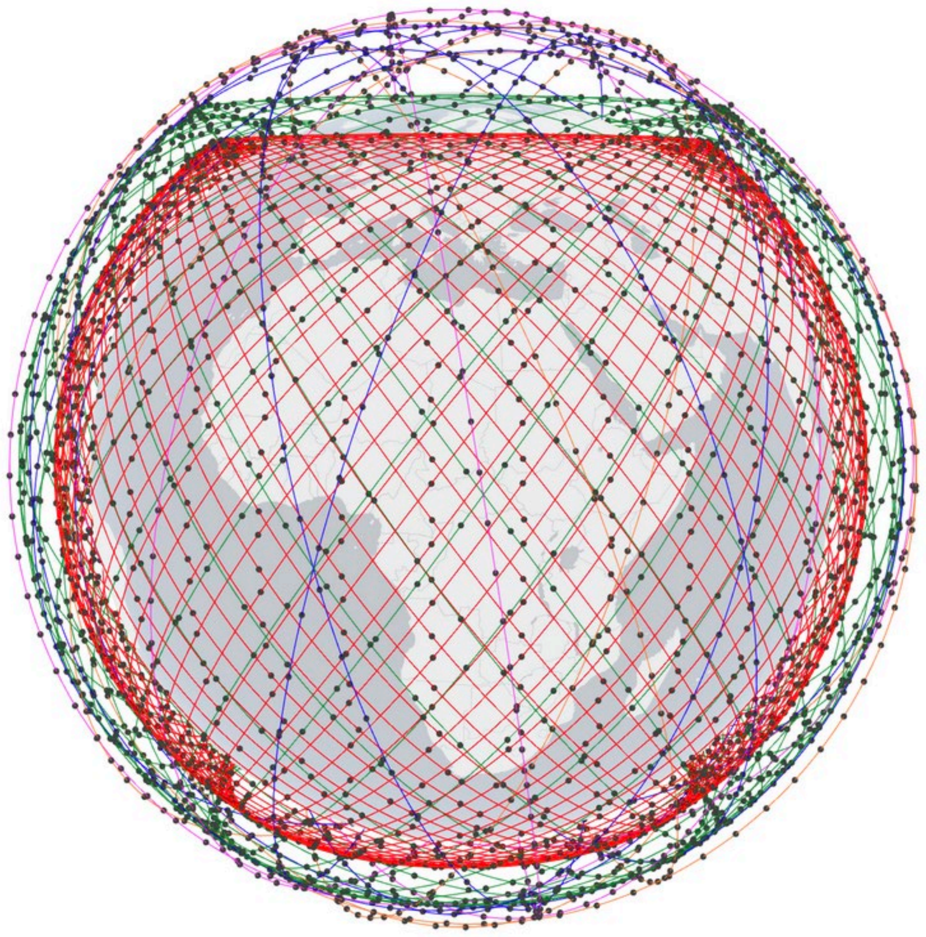
Cloud, Data centers

"Unlimited" compute & storage resources
High availability
Lower cost
High latency vs. Edge/Fog

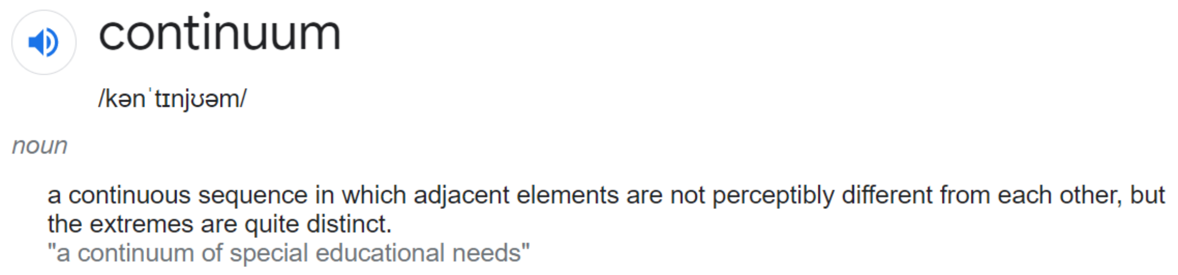
Compute power vs Latency vs Cost efficiency vs Reliability vs Sustainability

Edge-Cloud-Space continuum

- Lower Earth Orbit (LEO) Satellites (500 -2000 km)
  - Arranged in planes evenly spaced around the earth
  - Phase 1 of the Starlink constellation developed by SpaceX with 24 planes of 66 satellites each (Main Shell) →
  - Currently 7k+ sats; planned 30k+ sats (Gen2)
- Satellites can perform local computation → can act as regular compute nodes
- Satellites communicate with the core network through ground stations (via Ka-band RF) and within in constellation via (ISLs)
  - “Laser antenna” with high-speed and –bandwidth (100 Gbps)
  - ISLs typically arranged in a +Grid pattern, where each satellite keeps links to its successor and predecessor in its plane in addition to two cross-plane links


SpaceX Starlink

3D Compute Continuum in a nutshell



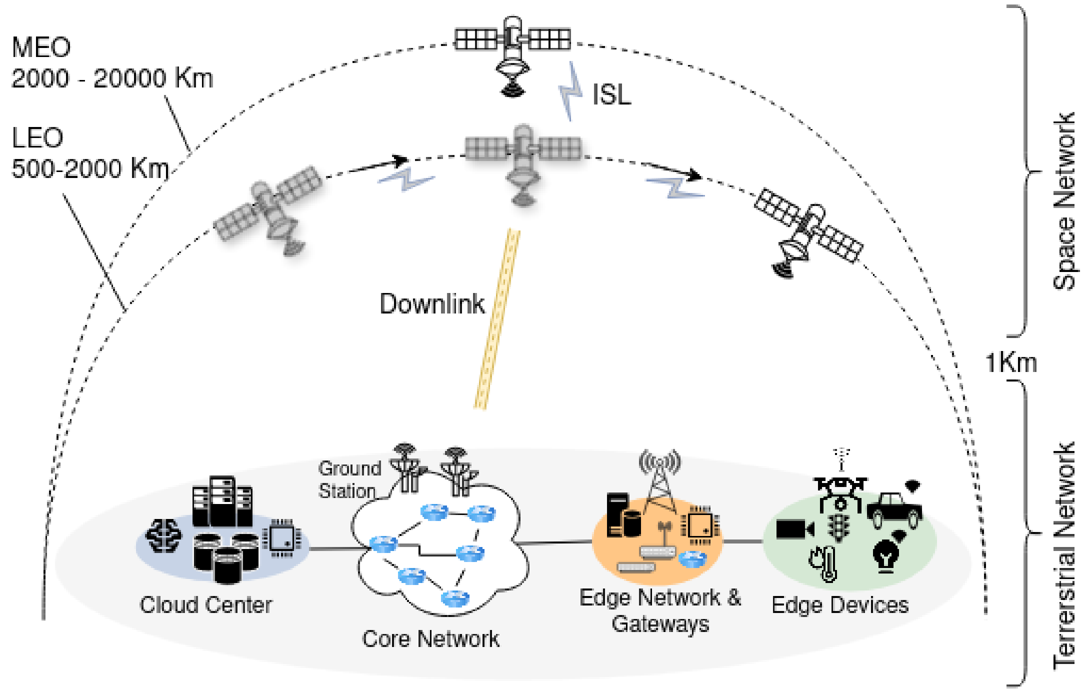
continuum

/kənˈtɪnjʊəm/

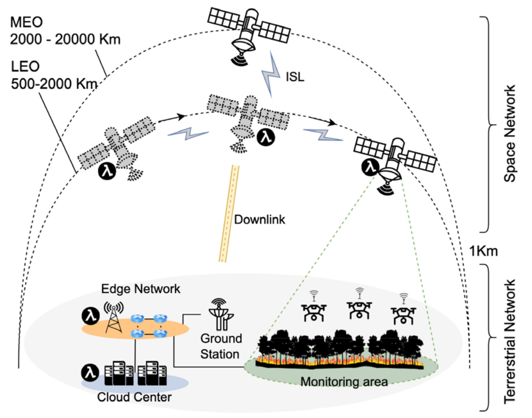
noun

a continuous sequence in which adjacent elements are not perceptibly different from each other, but the extremes are quite distinct.
"a continuum of special educational needs"

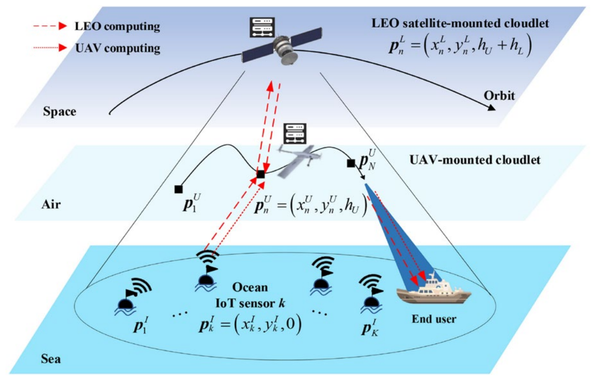
- Edge-Cloud-Space Continuum
- A large pool of **interconnected compute resources,** distributed across all spatial dimensions
- Usually hierarchically organized, logically layered, and **globally heterogeneous**



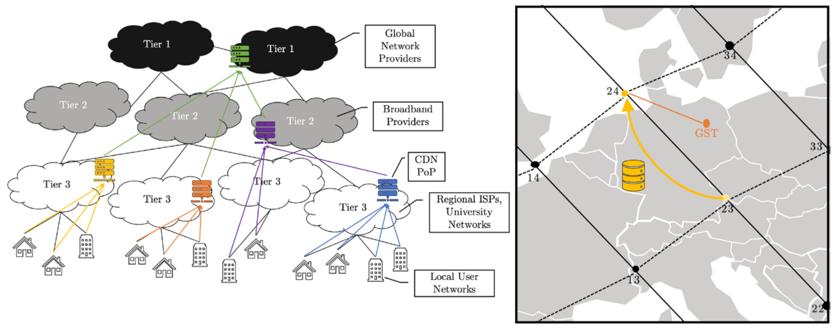
3D Compute Continuum Applications



Wildfires monitoring & recovery

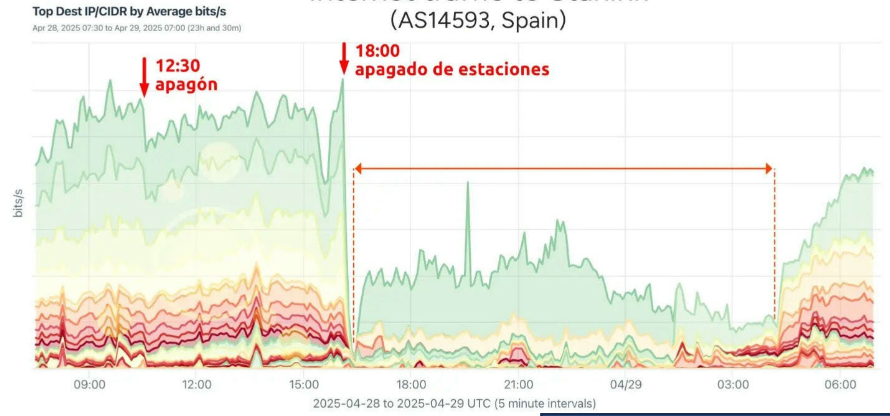

LEO computing
UAV computing
LEO satellite-mounted cloudlet
$p_n^L = (x_n^L, y_n^L, h_U + h_L)$
Space
Orbit
UAV-mounted cloudlet
Air
p_1^U
p_N^U
$p_n^U = (x_n^U, y_n^U, h_U)$
Ocean
IoT sensor k
$p_k^I = (x_k^I, y_k^I, 0)$
p_1^I
p_K^I
End user
Sea

Marine life monitoring

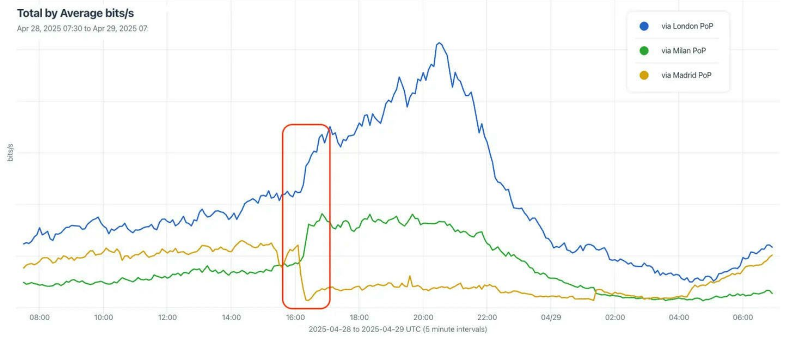


Content Delivery Networks (CDNs)

Internet traffic to Starlink
(AS14593, Spain)

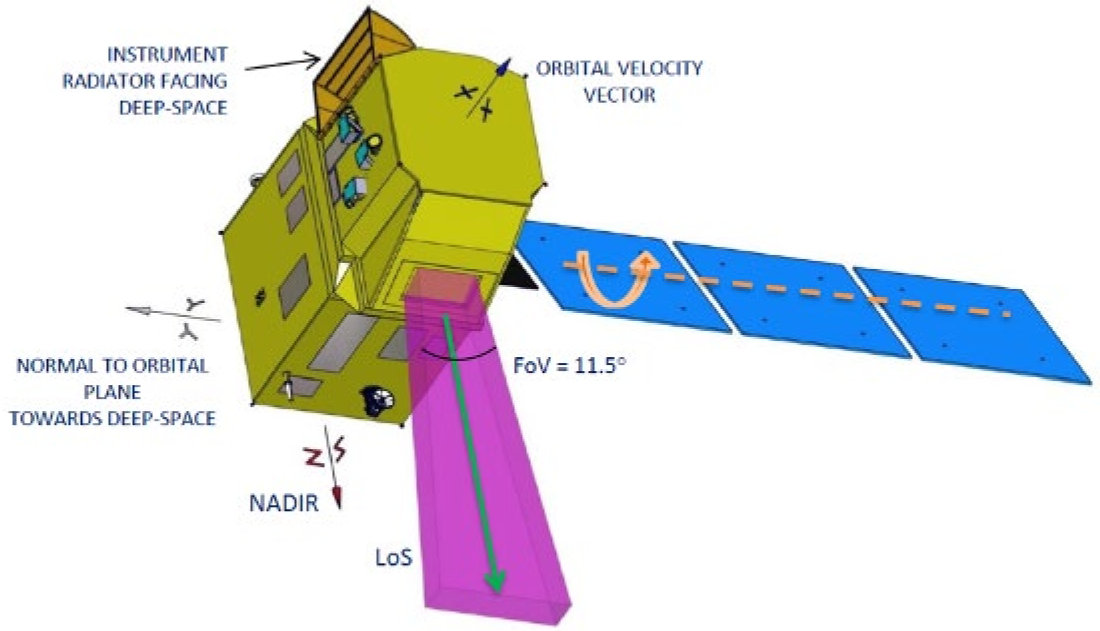


Internet traffic to Starlink
via PoPs in London, Milan, and Madrid



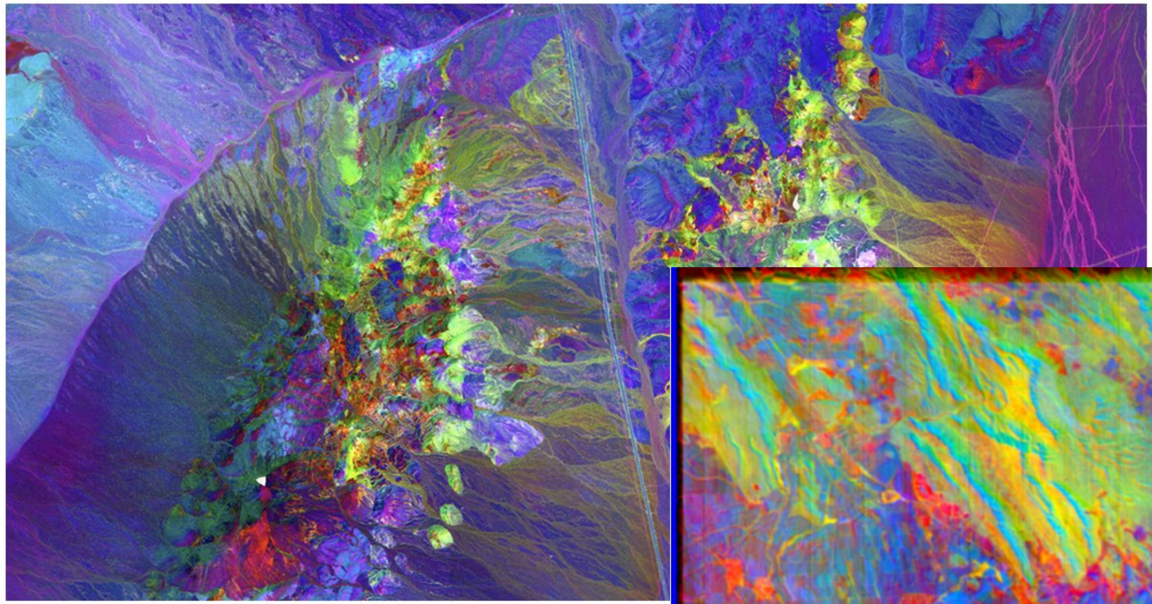
Dealing with large-scale blackouts

Sustainability in focus with hyperspectral imaging

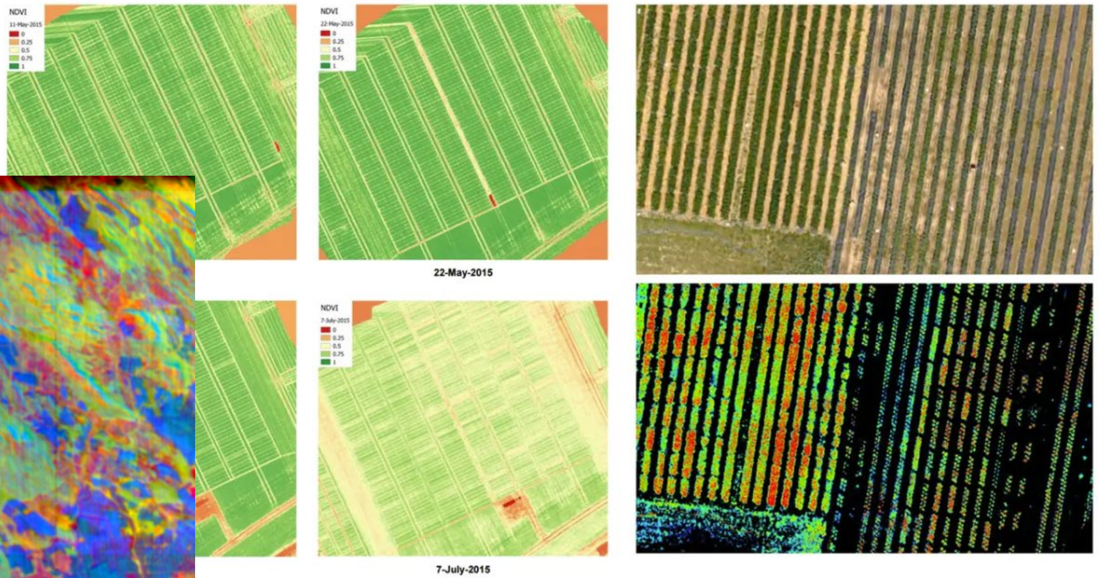

INSTRUMENT RADIATOR FACING DEEP-SPACE
ORBITAL VELOCITY VECTOR
X
Y
NORMAL TO ORBITAL PLANE TOWARDS DEEP-SPACE
FoV = 11.5°
Z
NADIR
LoS

- ESA‘s aims to provide *routine HS observations through the Copernicus Program*, supporting EU policies for natural resource management
- CHIME mission advances sustainable practices in agriculture, biodiversity, soil characterization, environmental preservation, …
- Many upcoming missions from ESA:
  - CIMR
  - CO2M
  - CRISTAL
  - LSTM
  - ROSE-L
  - …

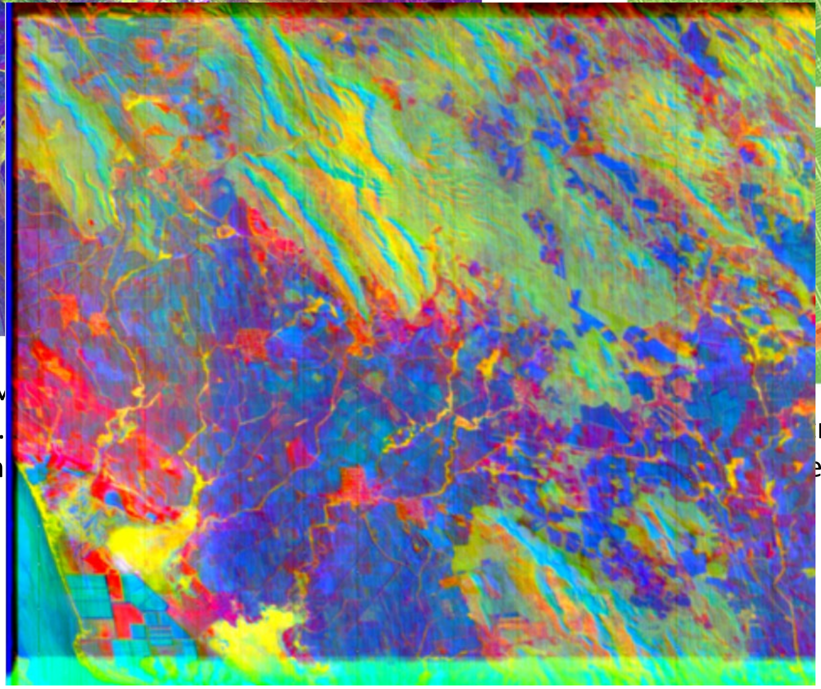
Sustainability in focus with hyperspectral imaging



Tanya Cross. Hyperspectral Imaging for Mining: EnM
Airborne Data. Online Dec 2025. https://earthdaily.
mining-enmap-satellite-data-versus-spectir-airborn



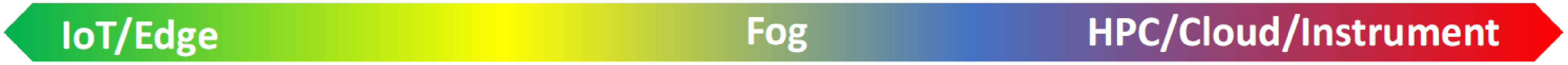
ral. Hyperspectral imaging for agriculture. Online Dec. 2025.
echyperspectral.com/en/applications/hyperspectral-imaging-agriculture



John O’Brien. Pixxel Hyperspectral Remote Sensing. Online Dec. 2025.
https://www.businesswire.com/news/home/20220328005027/en/Pixxel-Announces-%2425M-Investment-Led-By-Radical-Ventures-To-Advance-The-Worlds-Most-Detailed-Imaging-Satellites-for-Earth-Health-Monitoring

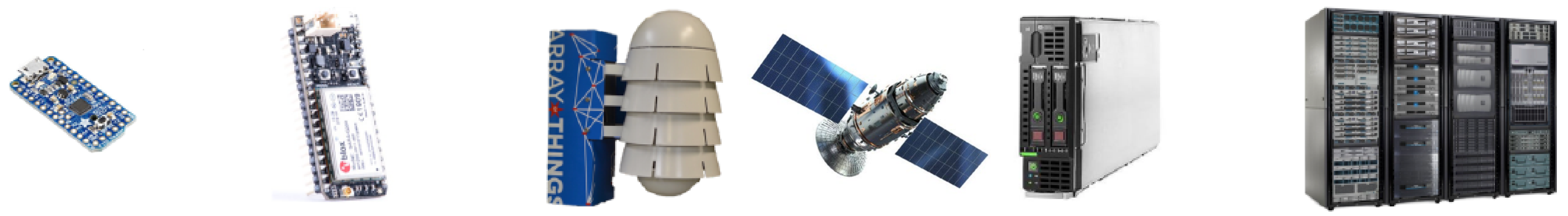
Device heterogeneity in the 3D Continuum

Differences of several magnitudes in terms of the number of available devices, their resources, reliability, device costs, their size ….



Size	Nano	Micro	Milli	Server	Fog	Campus	Facility
Example	Adafruit Trinket	Particle.io Boron	Array of Things	Linux Box	Co-located Blades	1000-node cluster	Datacenter
Memory	0.5K	256K	8GB	32GB	256G	32TB	16PB
Network	BLE	WiFi/LTE	WiFi/LTE	1 GigE	10GigE	40GigE	N*100GigE
Cost	\$5	\$30	\$600	\$3K	\$50K	\$2M	\$1000M

Count = 10^9
Size = 10^1



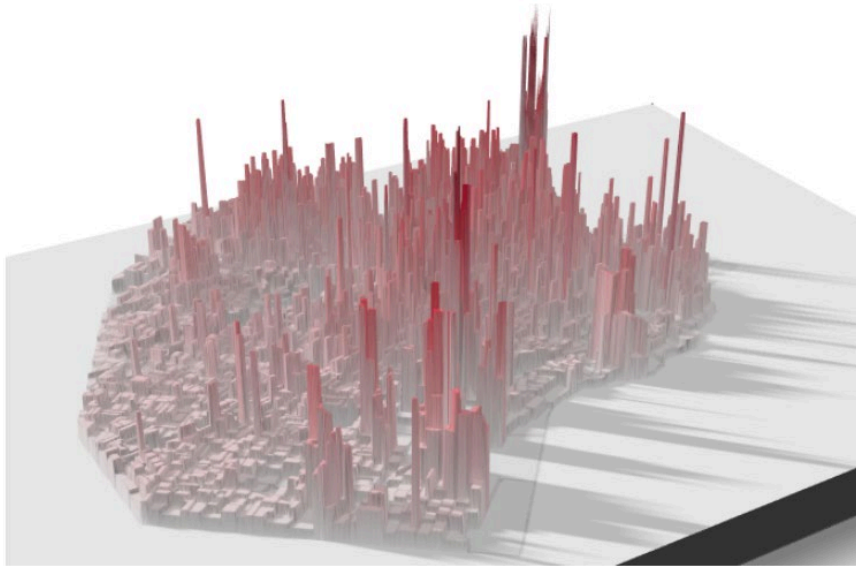
Count = 10^1
Size = 10^9

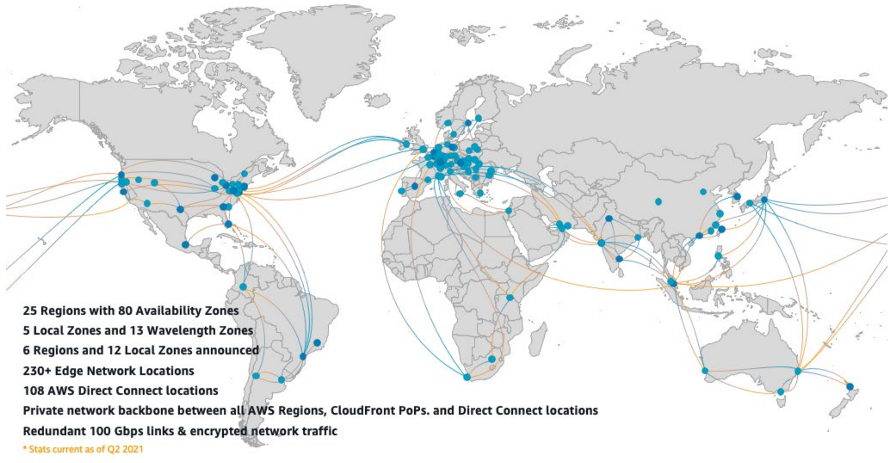
Managing the tradeoffs in the Computing Continuum!

Geo heterogeneity / computing continuum density

- Infrastructure/device density
- Data density
- User density
- Capability density
- Sporadic density shifts, e.g., a music festival or an event
- …

- Local vs global view on the infrastructure





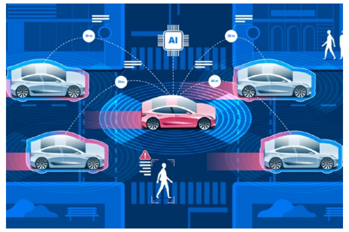
25 Regions with 80 Availability Zones
5 Local Zones and 13 Wavelength Zones
6 Regions and 12 Local Zones announced
230+ Edge Network Locations
108 AWS Direct Connect locations
Private network backbone between all AWS Regions, CloudFront PoPs. and Direct Connect locations
Redundant 100 Gbps links & encrypted network traffic
* Stats current as of Q2 2021

Software heterogeneity

Programming Languages

Application Domains

AI/ML



Autonomous Vehicles

Augmented Reality

Platforms

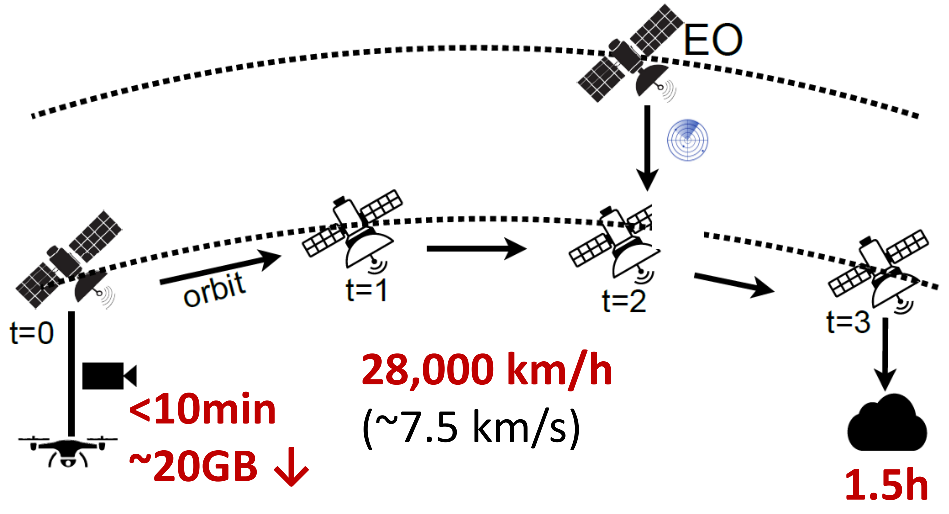
Google

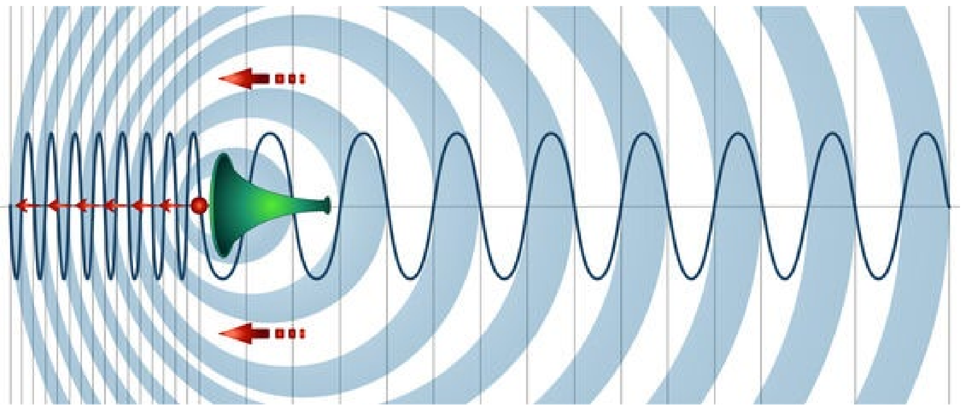
AWS

CloudFlair

Hypermobility of the compute infrastructure

- Satellite motion is governed by orbital dynamics and not user-defined
- Very high velocity
- Dynamic coverage footprint (only a few minutes to communicate)
- Dopler shift due to high-speed relative motion → signal deterioration
- Perturbed Keplerian orbits
  - Deviate from ideal, two-body Keplerian trajectory
- Data link variability
  - inclination, atmospheric conditions, antenna

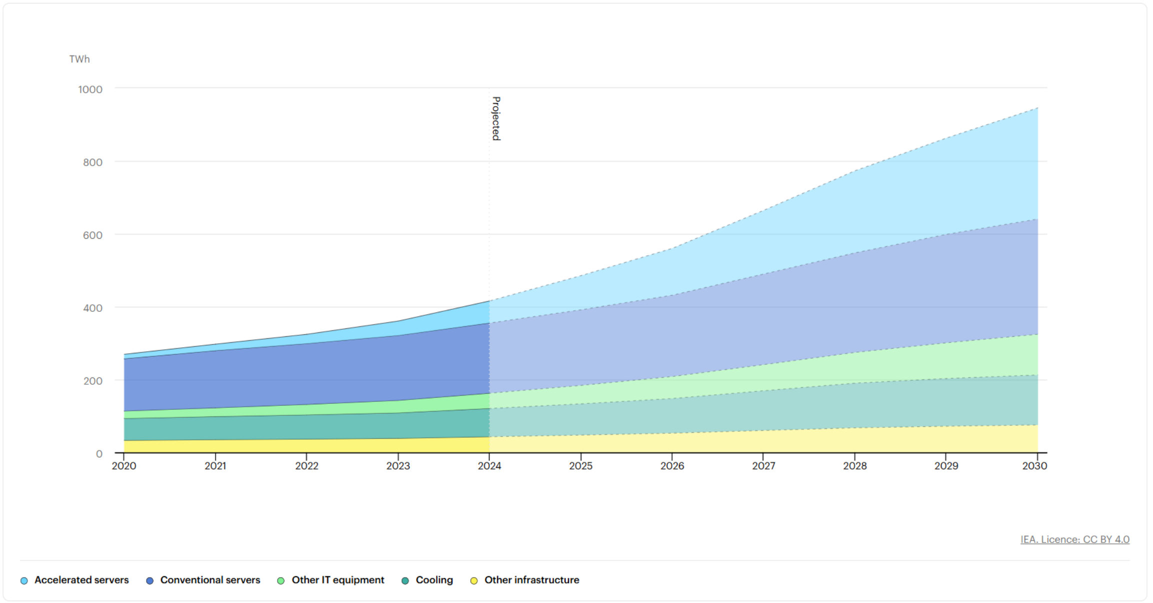




Growing energy demand of modern computing

- AI workload explosion
  - Training frontier models now requires tens to hundreds of MWh per run
  - Training a large transformer model emits ~300t of CO2 (vs a avg. person: 5t CO2 per year)
- Data center electricity growth
  - Data centers consume ~1–2% of global electricity today and expected to double by 2030
- Energy cost of data movement
  - Data tra
    than co

  → Increasi
    space p

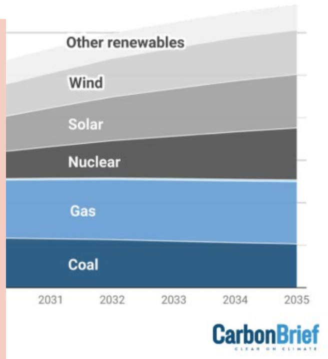


Emily M. Bender et al
Parrots: Can Languag
on Fairness, Accounta

The scalability of modern computing is increasingly bounded not only by compute, but also by energy availability and energy efficiency

- Energy is no longer a secondary concern—it is a limiting system resource shaping how we build and operate systems
- Energy efficiency now directly impacts cost, scalability, and carbon footprint

→ The ability to compute sustainably is becoming a first-class citizen!

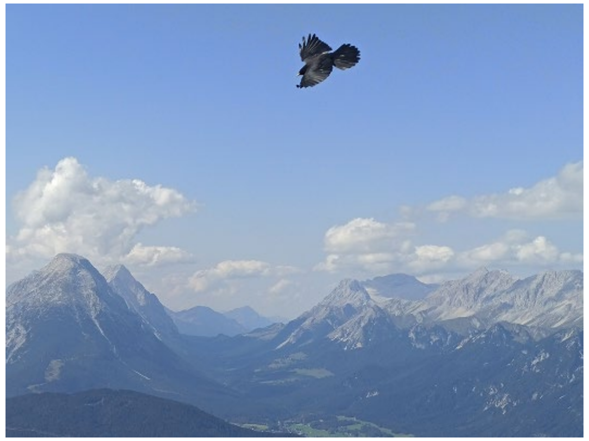


y of power to data centers

Freedom at last?

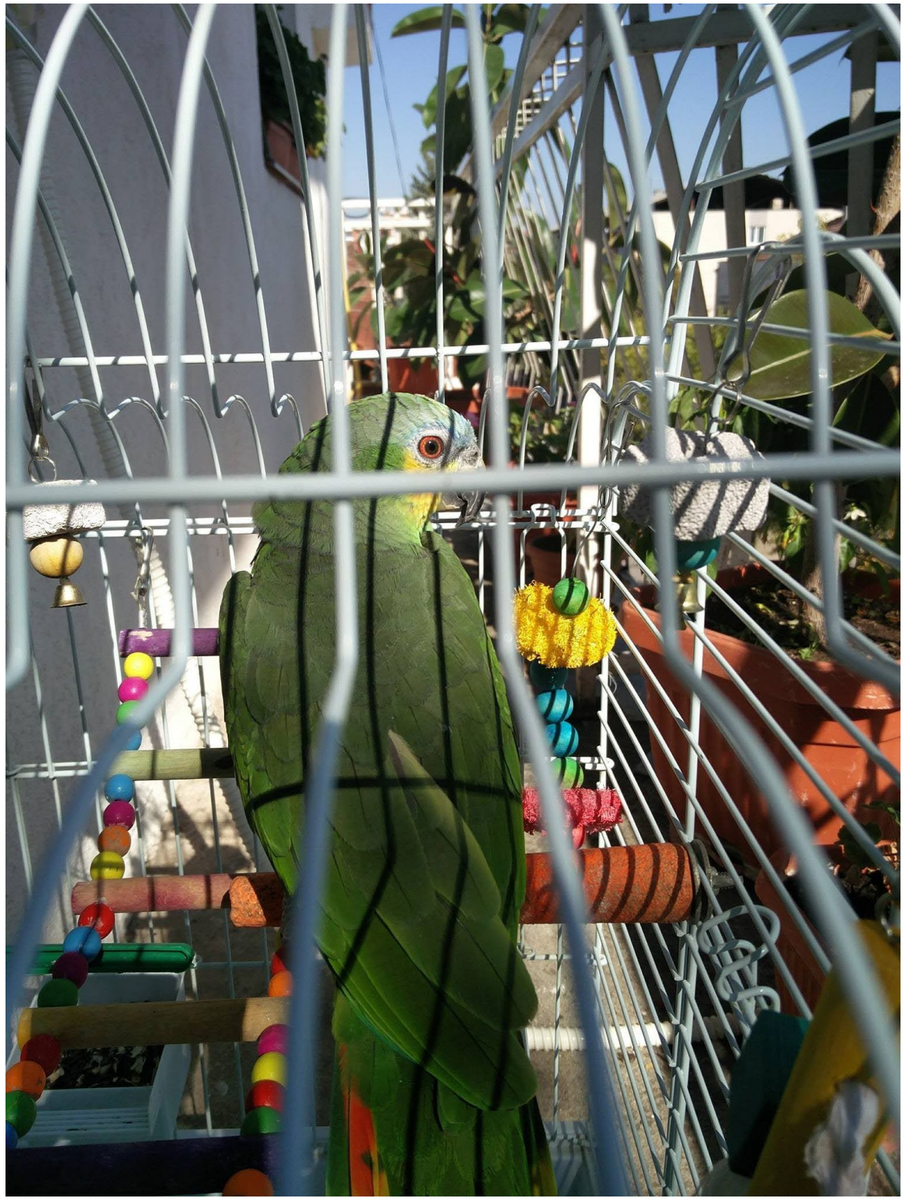
We can compute anywhere without worrying about the infrastructure!

(Utility-based compute model)



Is this already the reality?

Freedom?!



We are not locked in the cage!

Freedom?!

Limited freedom (at home) - the current state

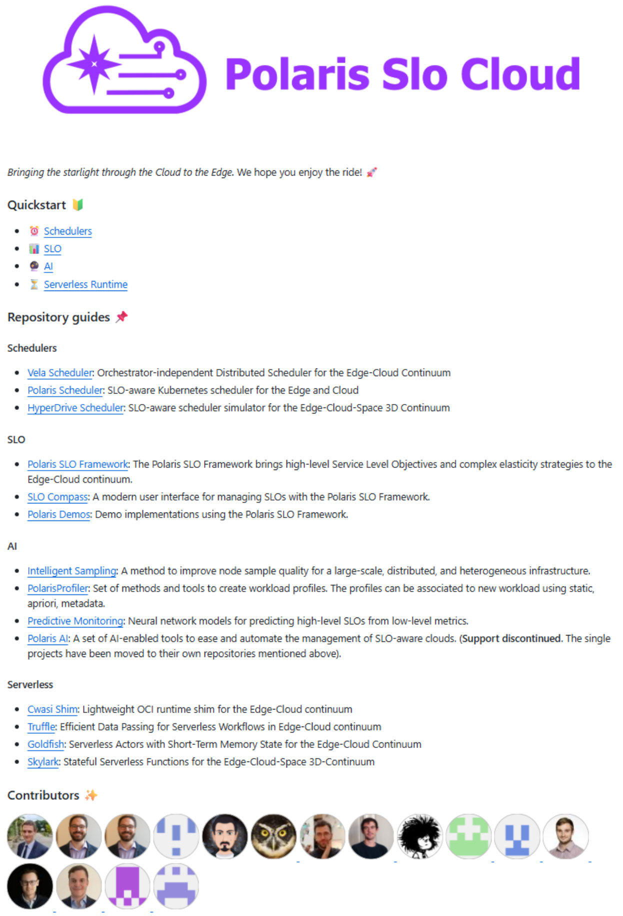
Freedom?!

We want full freedom

Project Polaris

- How can we **effectively manage** this new computing infrastructure while **efficiently managing the non-functional tradeoffs?**
- Polaris develops an ecosystem of tools and framework for managing the 3D continuum to guarantee that applications meet their KPIs
- Polaris is hosted by Linux Foundation

Nastic, S., Morichetta, A., Pusztai, T., Dustdar, S., Ding, X., Vij, D. and Xiong, Y., 2020. SLOC: Service level objectives for next-generation cloud computing. IEEE Internet Computing, 24(3), pp.39-50.



Polaris Slo Cloud

Bringing the starlight through the Cloud to the Edge. We hope you enjoy the ride! 🚀

Quickstart 🔰

- ⏰ Schedulers
- 📊 SLO
- 🔮 AI
- ⏳ Serverless Runtime

Repository guides 📌

Schedulers

- Vela Scheduler: Orchestrator-independent Distributed Scheduler for the Edge-Cloud Continuum
- Polaris Scheduler: SLO-aware Kubernetes scheduler for the Edge and Cloud
- HyperDrive Scheduler: SLO-aware scheduler simulator for the Edge-Cloud-Space 3D Continuum

SLO

- Polaris SLO Framework: The Polaris SLO Framework brings high-level Service Level Objectives and complex elasticity strategies to the Edge-Cloud continuum.
- SLO Compass: A modern user interface for managing SLOs with the Polaris SLO Framework.
- Polaris Demos: Demo implementations using the Polaris SLO Framework.

AI

- Intelligent Sampling: A method to improve node sample quality for a large-scale, distributed, and heterogeneous infrastructure.
- PolarisProfiler: Set of methods and tools to create workload profiles. The profiles can be associated to new workload using static, apriori, metadata.
- Predictive Monitoring: Neural network models for predicting high-level SLOs from low-level metrics.
- Polaris AI: A set of AI-enabled tools to ease and automate the management of SLO-aware clouds. (**Support discontinued**. The single projects have been moved to their own repositories mentioned above).

Serverless

- Cwasi Shim: Lightweight OCI runtime shim for the Edge-Cloud continuum
- Truffle: Efficient Data Passing for Serverless Workflows in Edge-Cloud continuum
- Goldfish: Serverless Actors with Short-Term Memory State for the Edge-Cloud Continuum
- Skylark: Stateful Serverless Functions for the Edge-Cloud-Space 3D-Continuum

Contributors ✨

https://github.com/polaris-slo-cloud

Overview of SLOs

- A Service Level Objective (SLO) is a specific, quantifiable target for the performance of an application, often expressed as a percentage over a period of time
  - Max 100 ms response time for 99% of the requests to a traffic prediction service

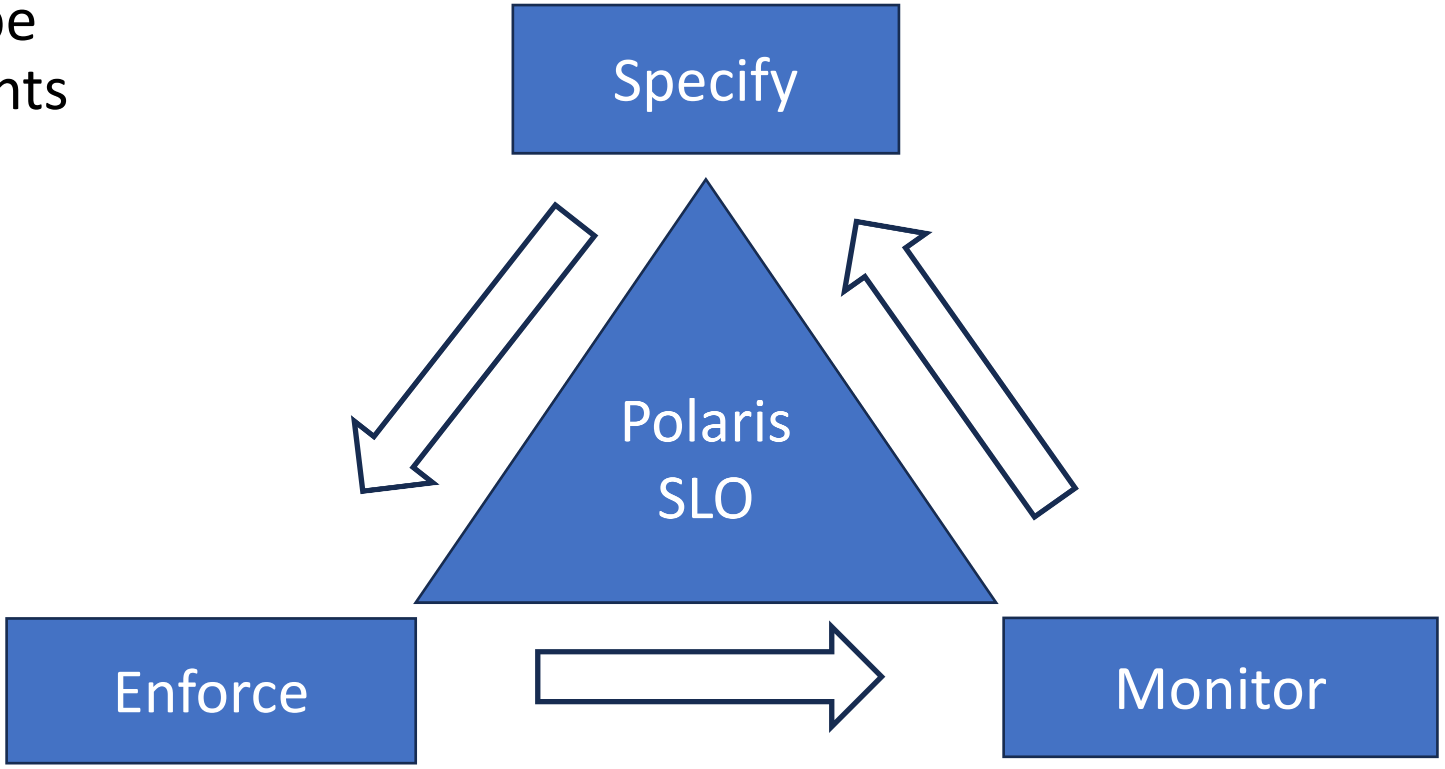
- **SLOs can be measured!**
- **SLOs can be enforced!**

Traditional SLOs

- Mainly focused on reactive, low-level, system metrics
- Unable to capture tradeoffs in non-functional requirement
- Hard to align with application business KPIs

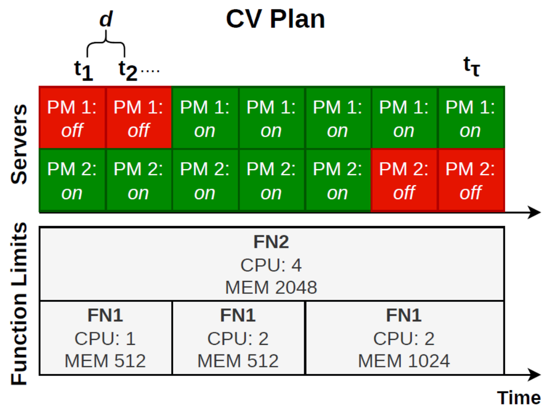
Polaris SLOs for the 3D Continuum

- Performance and reliability guarantees, whose impact can be reflected in business requirements and KPIs, e.g., cost efficiency

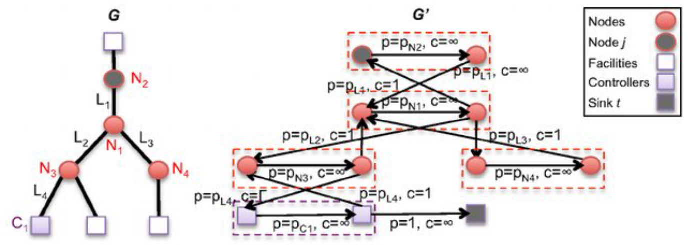


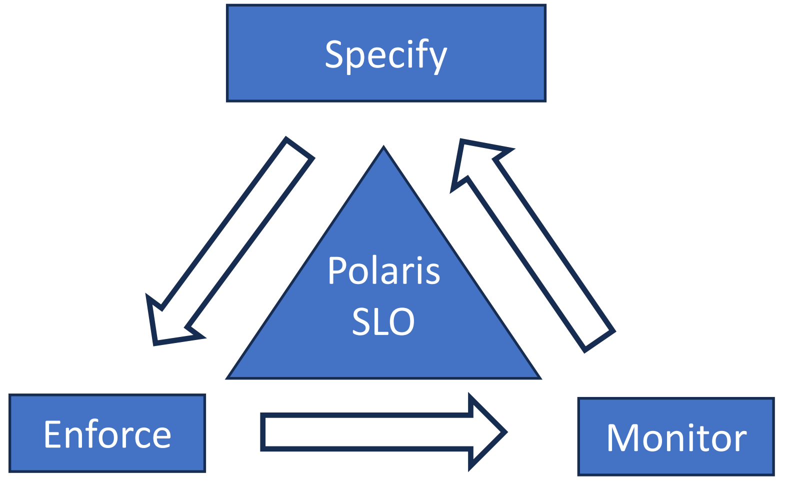
Pusztai, T., Nastic, S., Ding, X., Vij, D. and Xiong, Y., 2021, September. SLO script: A novel language for implementing complex cloud-native elasticity-driven SLOs. In 2021 IEEE International Conference on Web Services (ICWS) (pp. 21-31).

Polaris SLOs for the 3D Continuum

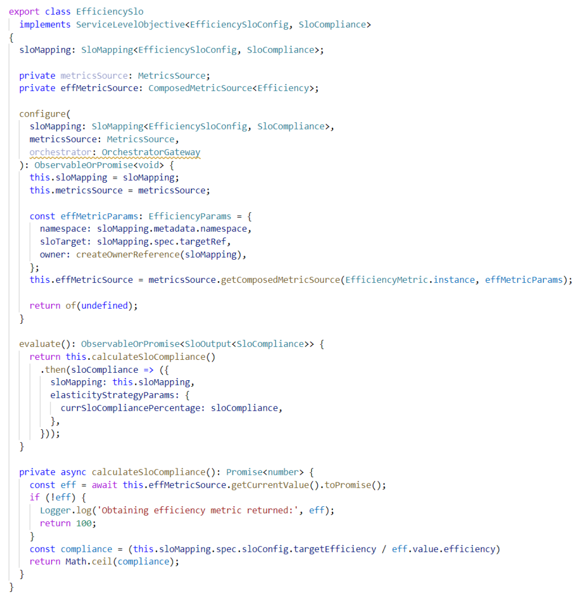


Serverless scaling, consolidation & scheduling strategies



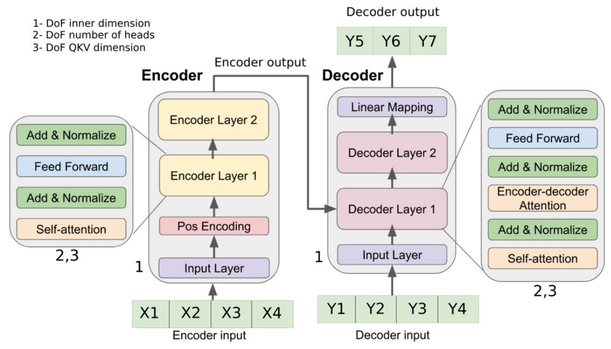


SLO Script



```
export class EfficiencySlo
  implements ServiceLevelObjective<EfficiencySloConfig, SloCompliance>
{
  sloMapping: SloMapping<EfficiencySloConfig, SloCompliance>;

  private metricsSource: MetricsSource;
  private effMetricSource: ComposedMetricSource<Efficiency>;

  configure(
    sloMapping: SloMapping<EfficiencySloConfig, SloCompliance>,
    metricsSource: MetricsSource,
    orchestrator: OrchestratorGateway
  ): ObservableOrPromise<void> {
    this.sloMapping = sloMapping;
    this.metricsSource = metricsSource;

    const effMetricParams: EfficiencyParams = {
      namespace: sloMapping.metadata.namespace,
      sloTarget: sloMapping.spec.targetRef,
      owner: createOwnerReference(sloMapping),
    };
    this.effMetricSource = metricsSource.getComposedMetricSource(EfficiencyMetric.instance, effMetricParams);

    return of(undefined);
  }

  evaluate(): ObservableOrPromise<SloOutput<SloCompliance>> {
    return this.calculateSloCompliance()
      .then(sloCompliance => ({
        sloMapping: this.sloMapping,
        elasticityStrategyParams: {
          currSloCompliancePercentage: sloCompliance,
        },
      }));
  }

  private async calculateSloCompliance(): Promise<number> {
    const eff = await this.effMetricSource.getCurrentValue().toPromise();
    if (!eff) {
      Logger.log('Obtaining efficiency metric returned:', eff);
      return 100;
    }
    const compliance = (this.sloMapping.spec.sloConfig.targetEfficiency / eff.value.efficiency)
    return Math.ceil(compliance);
  }
}
```

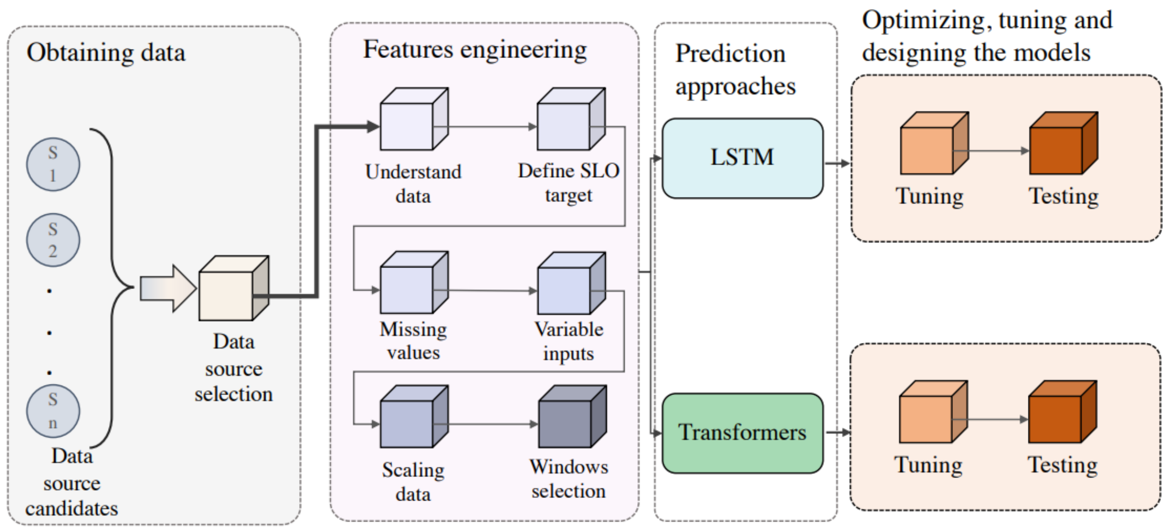
Predictive Monitoring



Pusztai, T., Morichetta, A., Pujol, V.C., Dustdar, S., Nastic, S., Ding, X., Vij, D. and Xiong, Y. (2022). A novel middleware for efficiently implementing complex cloud-native SLOs. In IEEE 14th International Conference on Cloud Computing (CLOUD) (pp. 410-420).

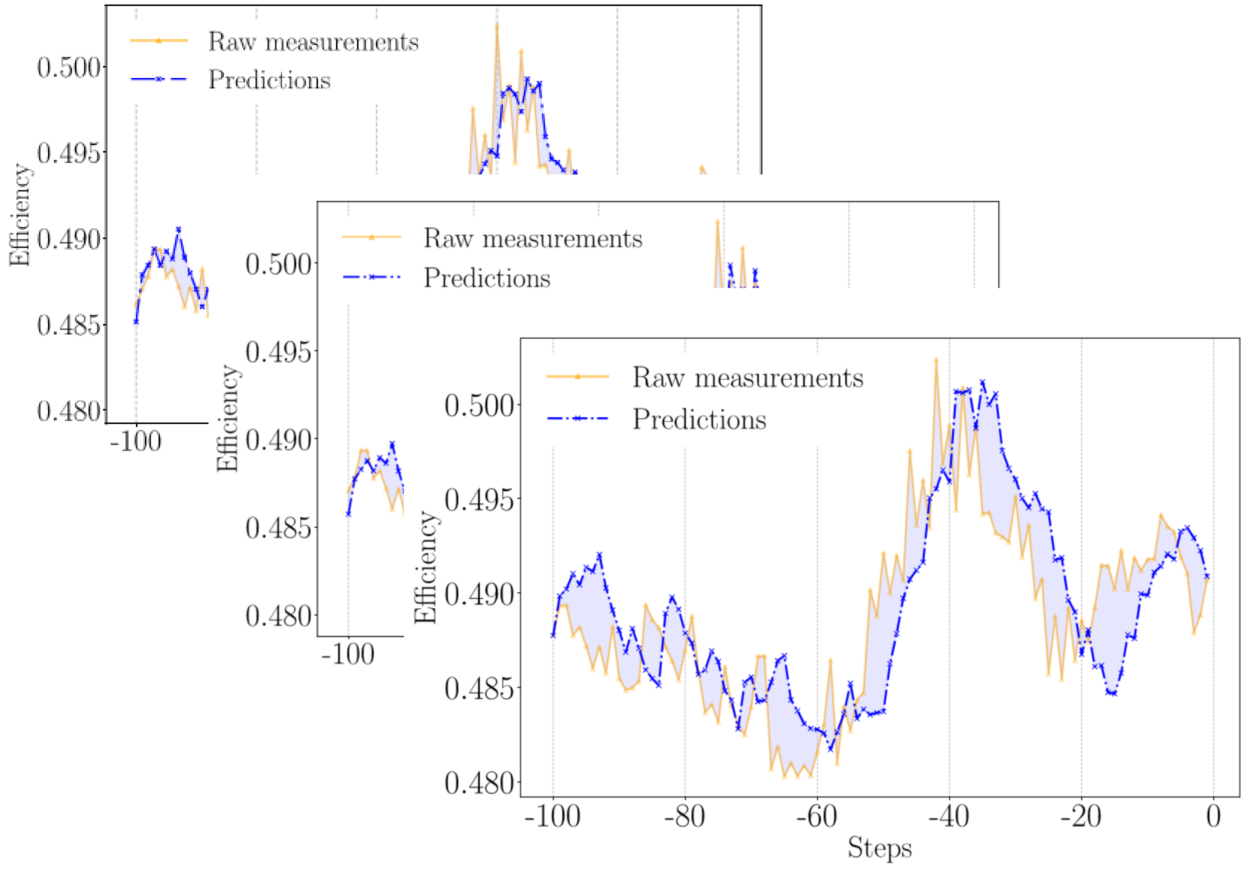
Predictive Monitoring High-Level SLO Metrics

- Predictive monitoring vs. reactive monitoring
- Two main prediction models
  - Long short-term memory (LSTM)
  - Temporal Fusion Transformers (TFTs)
- Polaris prediction models are designed so that they can predict the high-level SLO violations (e.g., cost efficiency) given only the low-level infrastructure metrics

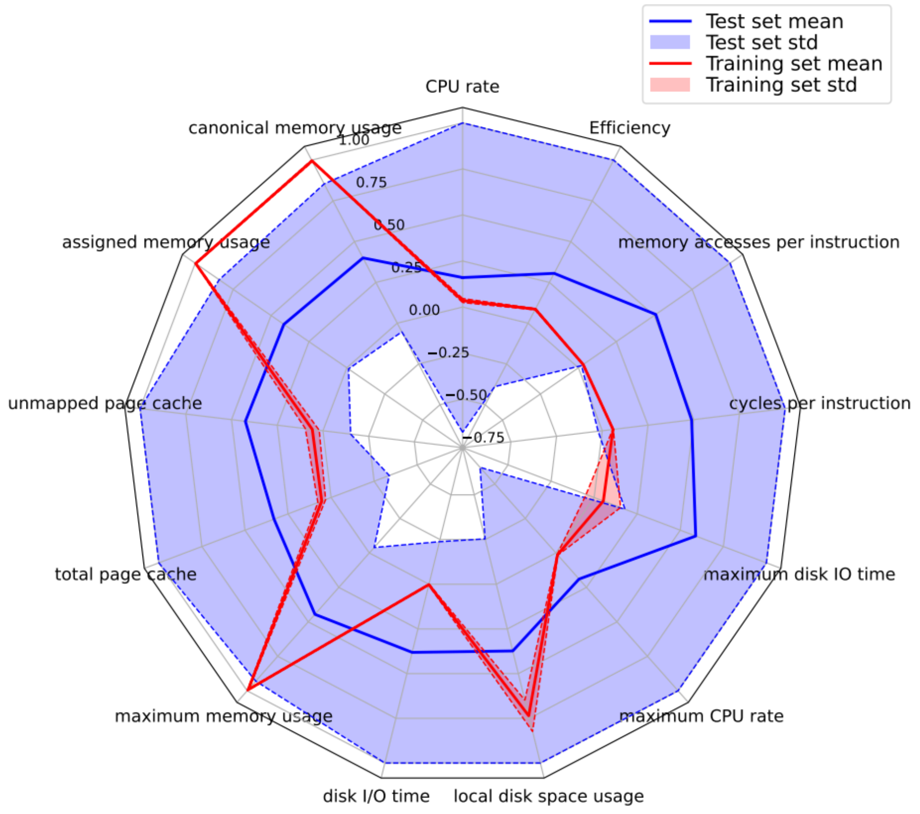


Morichetta, A., Casamayor Pujol, V., Nastic, S., Pusztai, T., Raith, P., … Zhang, Z. (2023). Demystifying Deep Learning in Predictive Monitoring for Cloud-Native SLOs. In IEEE 15th International Conference on Cloud Computing (CLOUD)

Predictive Monitoring of Cost Efficiency

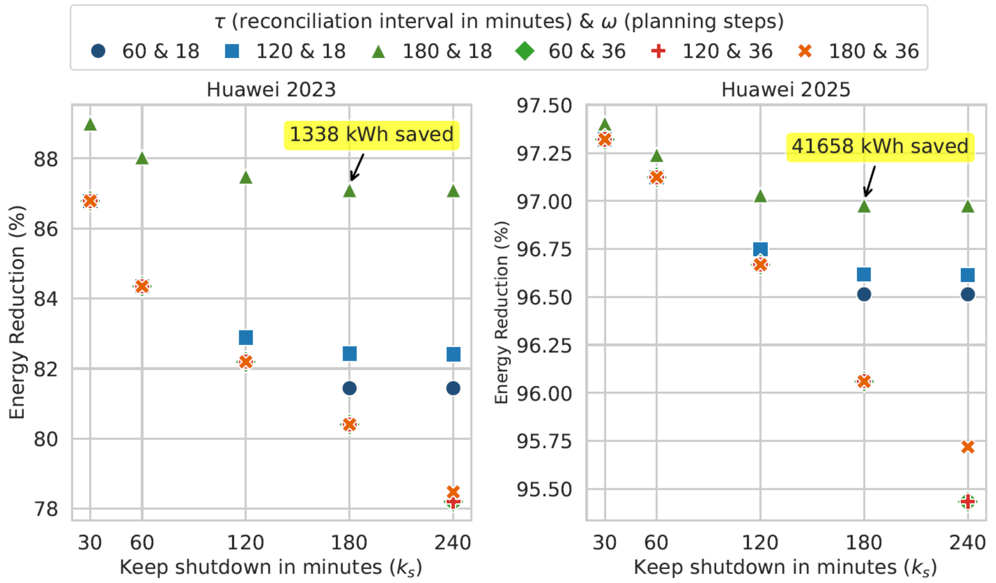


(c) Prediction for t+3

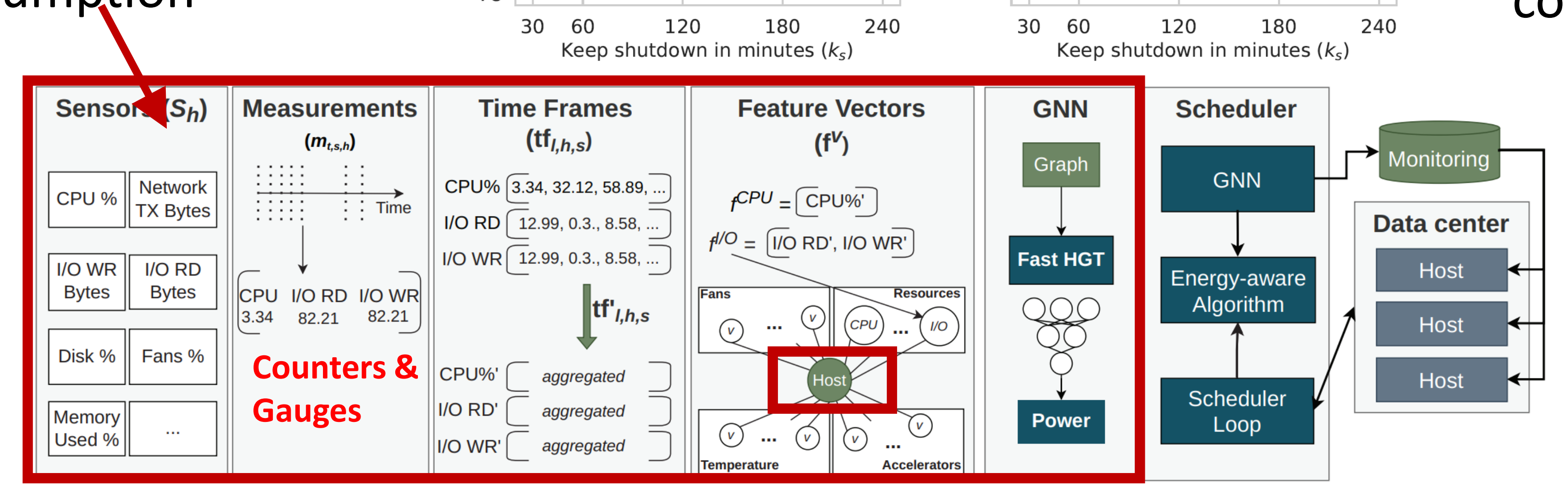


Predictive Monitoring of Energy Consumption

- **Predict energy consumption:** The model captures heterogeneous hardware and software that are inputs for the GHT-GNN, which outputs (min, max, avg) energy consumption



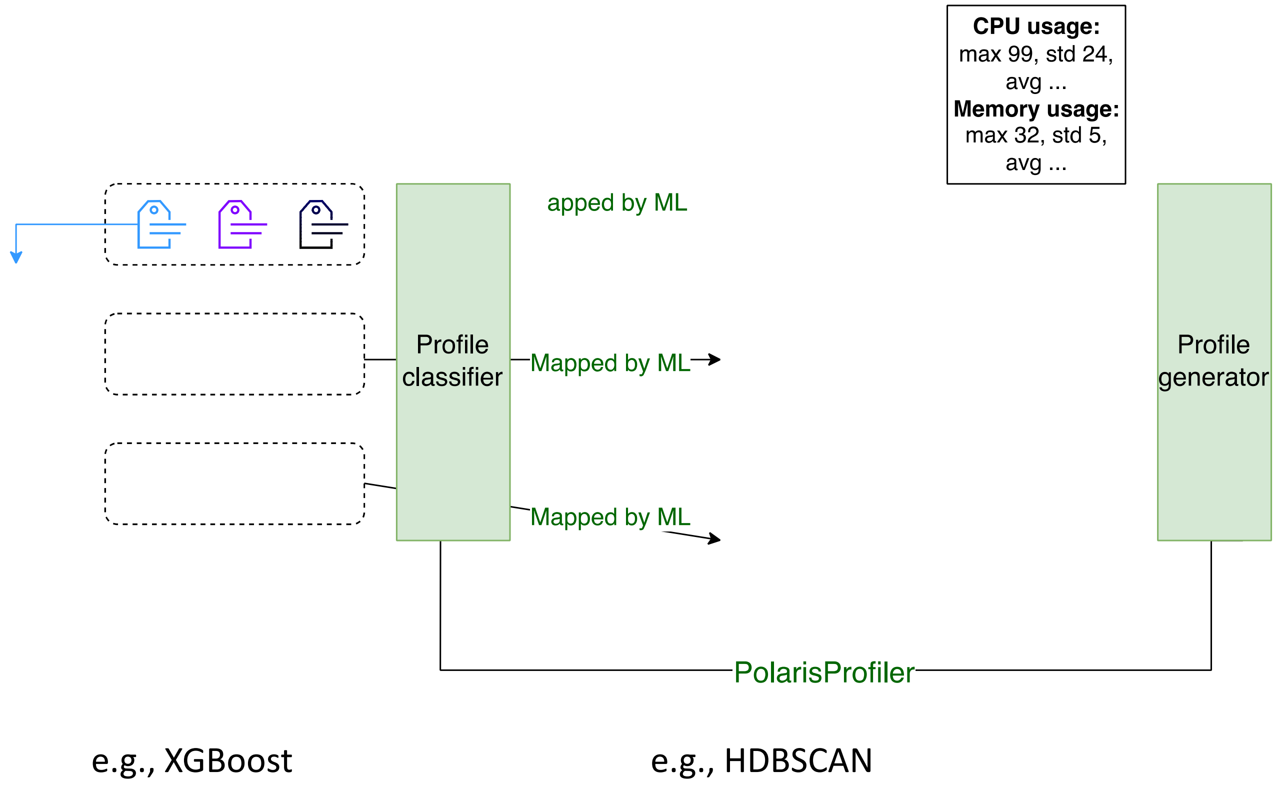
- **Energy-aware scheduling & consolidation:** The system builds applications' energy signature and provides custom opportunistic scheduling and consolidation algorithms



P. Raith et al., "Opportunistic Energy-Aware Scheduling for Container Orchestration Platforms Using Graph Neural Networks," 2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid '24)

Addressing Predictions Bootstrapping

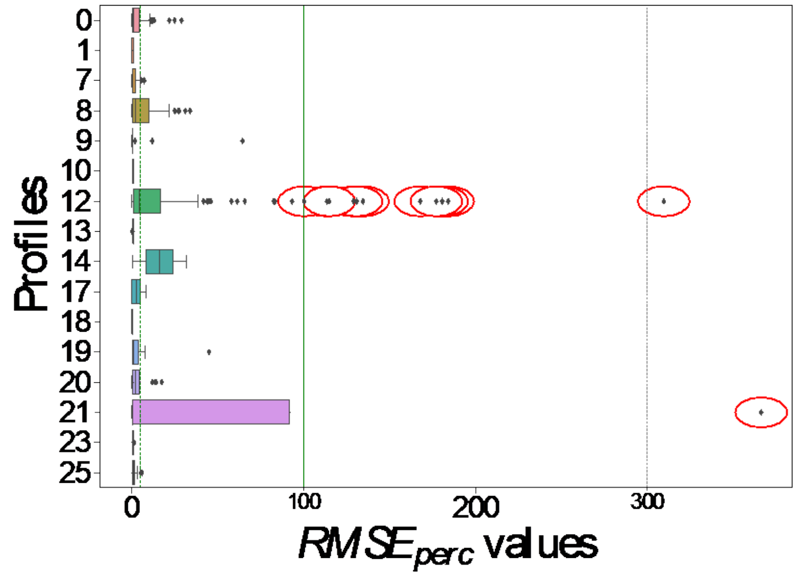
- Metadata-based Workload Profiling
- Main intuition: use apriori available meta data to derive **workload profiles**
  - Profile generation based on dynamic data
  - Profile classification based on static metadata

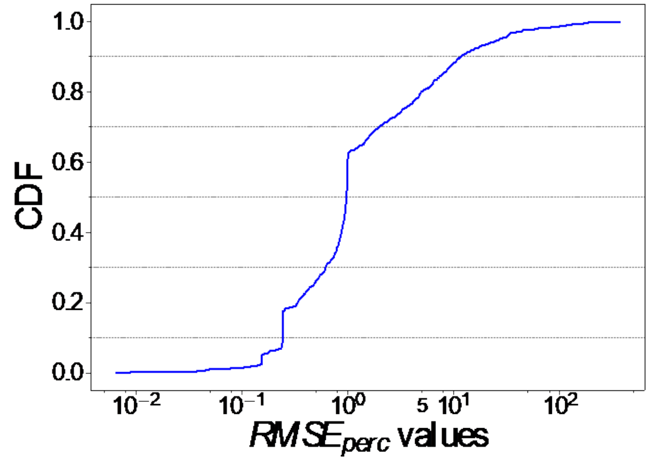


Morichetta, A., Casamayor Pujol, V., Nastic, S., Pusztai, T., Raith, P., … Zhang, Z. (2023). Demystifying Deep Learning in Predictive Monitoring for Cloud-Native SLOs. In IEEE 15th International Conference on Cloud Computing

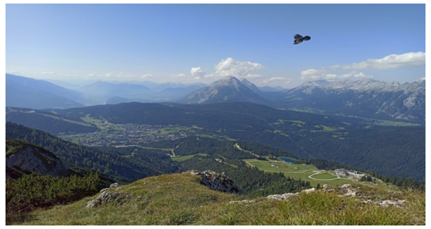
Metadata-based Profiler Results

- Predicting job duration
  - 1000 proportionally-sampled jobs - not used for training
  - Use avg duration in each profile as prediction
  - **RMSEperc below 5% for more than 80% of the profiled jobs**
- Main takeaway: We can learn a lot about the future workload by looking at its metadata





Core promises of Serverless computing



No infrastructure management

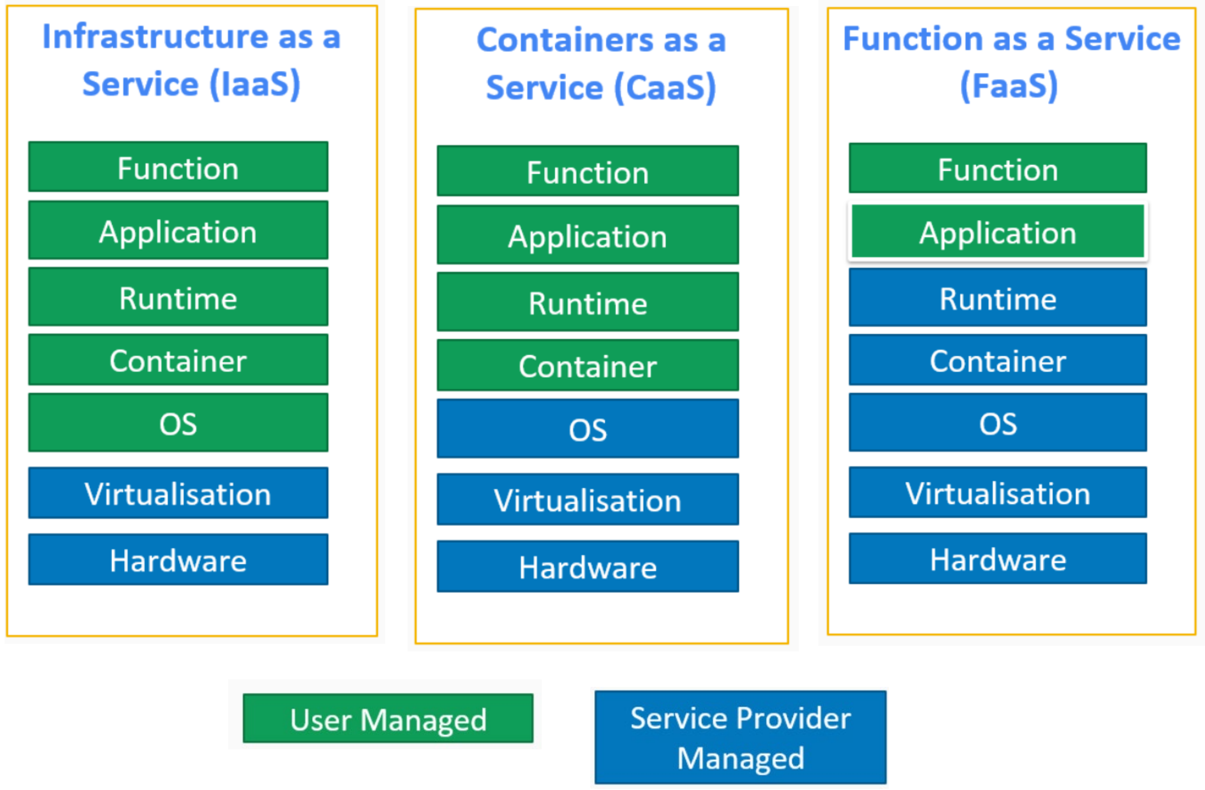
Flexible elastic scaling

Availability

Cost effectiveness

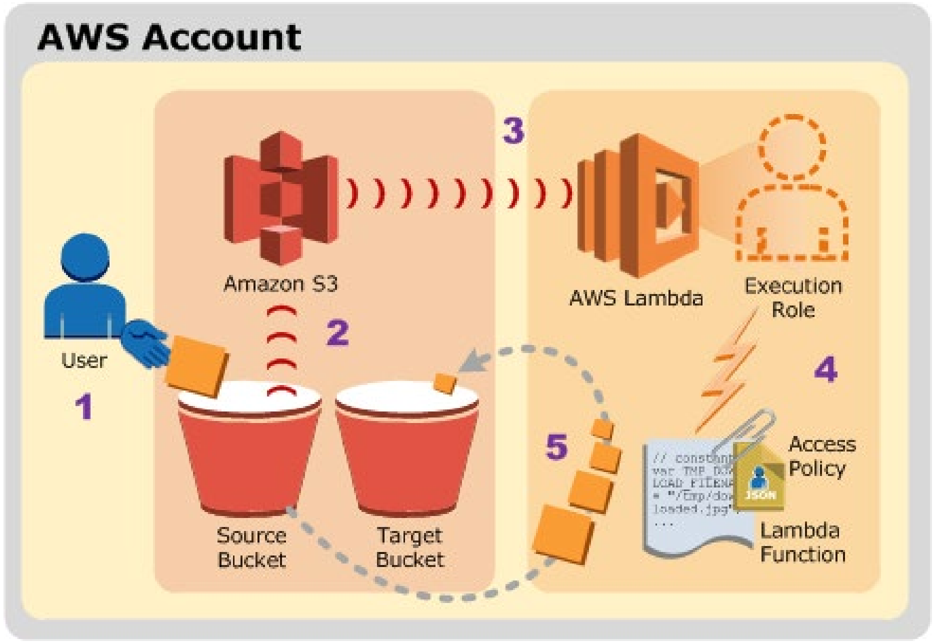
Serverless Computing in a Nutshell

- Serverless ≠ No Servers
- Natural evolution in Cloud Computing
- Novel Function as a Service (FaaS) programming paradigm
- Novel execution model



Function as a Service (FaaS) Model

- A **programming and execution model** where developers can deploy and run code in response to events
- Application is decomposed into "triggers" and "actions" (**stateless functions**)
  - Developers write functions and upload them to a FaaS platform
  - Functions are triggered by events, such as a user request or a change in data
- Advanced programming models built on top of this basic stateless model to enable stateful FaaS



Marcelino, C., Shahhoud, J., & Nastic, S. (2024). GoldFish: Serverless Actors with Short-Term Memory State for the Edge-Cloud Continuum. In Proceedings of the 14th International Conference on the Internet of Things (IoT 2024). New York, NY, USA:

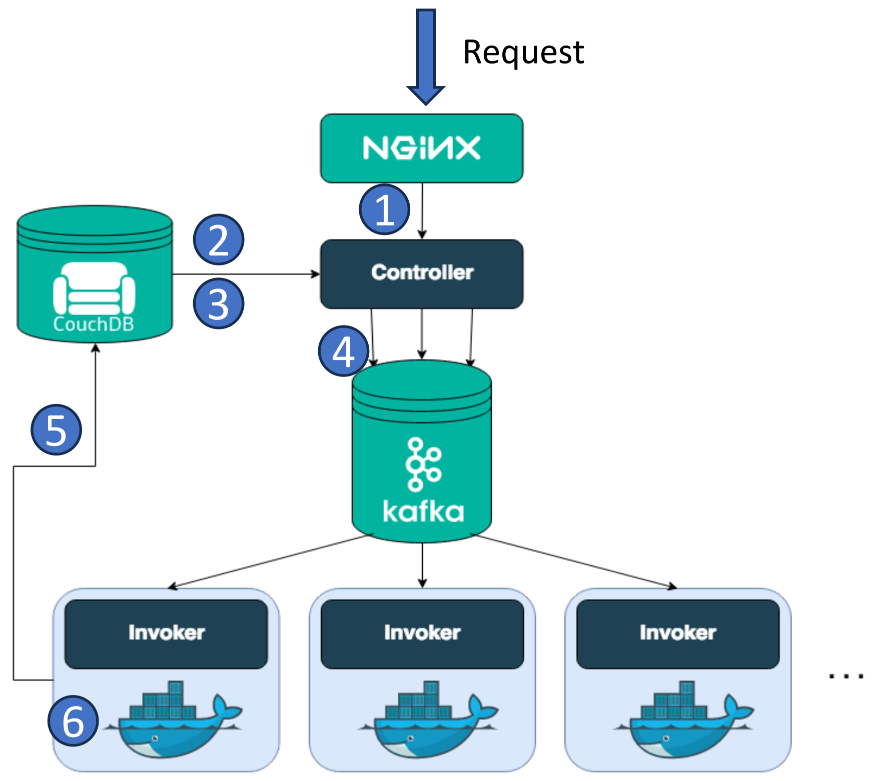
Function Isolation and Virtualization Models

- Each serverless function is deployed and executed within a **dedicated virtual container**
- Various isolation and virtualization models:
  - Containers (K-Native on Kubernetes)
  - MicroVMs - Firecracker (AWS Lambda)
  - V8 Isolates (Cloudflare, Deno Deploy)
  - WASM runtime (Fastly)
  - Unikernels (Mainly academia)
- The container is typically alive only during request processing -> **Scaled to zero**

Scale-to-zero architecture

- Function execution step-by-step (simplified)
  1. New request is forwarded by the ingress/router
  2. Authentication information is fetched from the database
  3. The action (function) metadata is fetched from the database
  4. A message is sent to the Invoker via an MQ
  5. The Invoker fetches the action (function) from the database
  6. The Invoker executed the function
- Steps 2 – 5 need to happen before the function gets invoked -> **Cold start issue!**

Performance vs. Cost Efficiency



Marcelino, C., & Nastic, S. (2023). CWASI: A WebAssembly Runtime Shim for Inter-function Communication in the Serverless Edge-Cloud Continuum. In The Eighth ACM/IEEE Symposium on Edge Computing (SEC 2023)

Serverless beyond the FaaS

Serverless "support BaaS services"

- API Gateways for synchronous (request-based) communication
- Object stores for state management
- Function orchestration for serverless workflows
- Database change streams for reacting to data changes (DynamoDB Streams)

Self-provisioning infrastructures

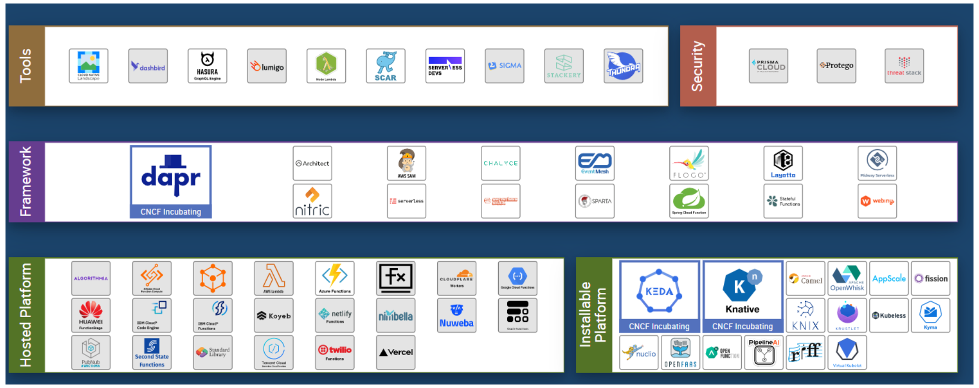
- Address the issue of having to provision and manage BaaS
- Extend the serverless paradigm beyond compute

Nastic, S. (2023). Self-Provisioning Infrastructures for the Next Generation Serverless Computing. SN Computer Science. Special Issue on Advances in Enterprise Information Systems.

Serverless managed Cloud services

- Serverless relational database (Amazon Aurora)
- Serverless container management (AWS Fargate)
- Serverless warehouse (Google BigQuery)

Larcher, T., Gritsch, P., Nastic, S., & Ristov, S. (2024). BAASLESS: Backend-as-a-Service (BaaS)-Enabled Workflows in Federated Serverless Infrastructures. IEEE Transactions on Cloud Computing, 1(1), 1–15.



Many Opportunities …

- Fine-grained pay-per-use model with preserved SLOs
- Cost-efficient resource usage with Scale-to-Zero
- Rapid elasticity & auto-scaling due to lightweight virtualization
- Native cloud offloading
- True utility-based resource consumption
- Disaggregation of compute and storage
- Stateless workflows and life-cycle

- Effortlessly parallelizing computing
- Extremely bursty workloads
- CUP-bound applications
- Developing scalable Edge-native solution
- True event-driven applications
- ….

Core promises of Serverless computing

No infrastructure management

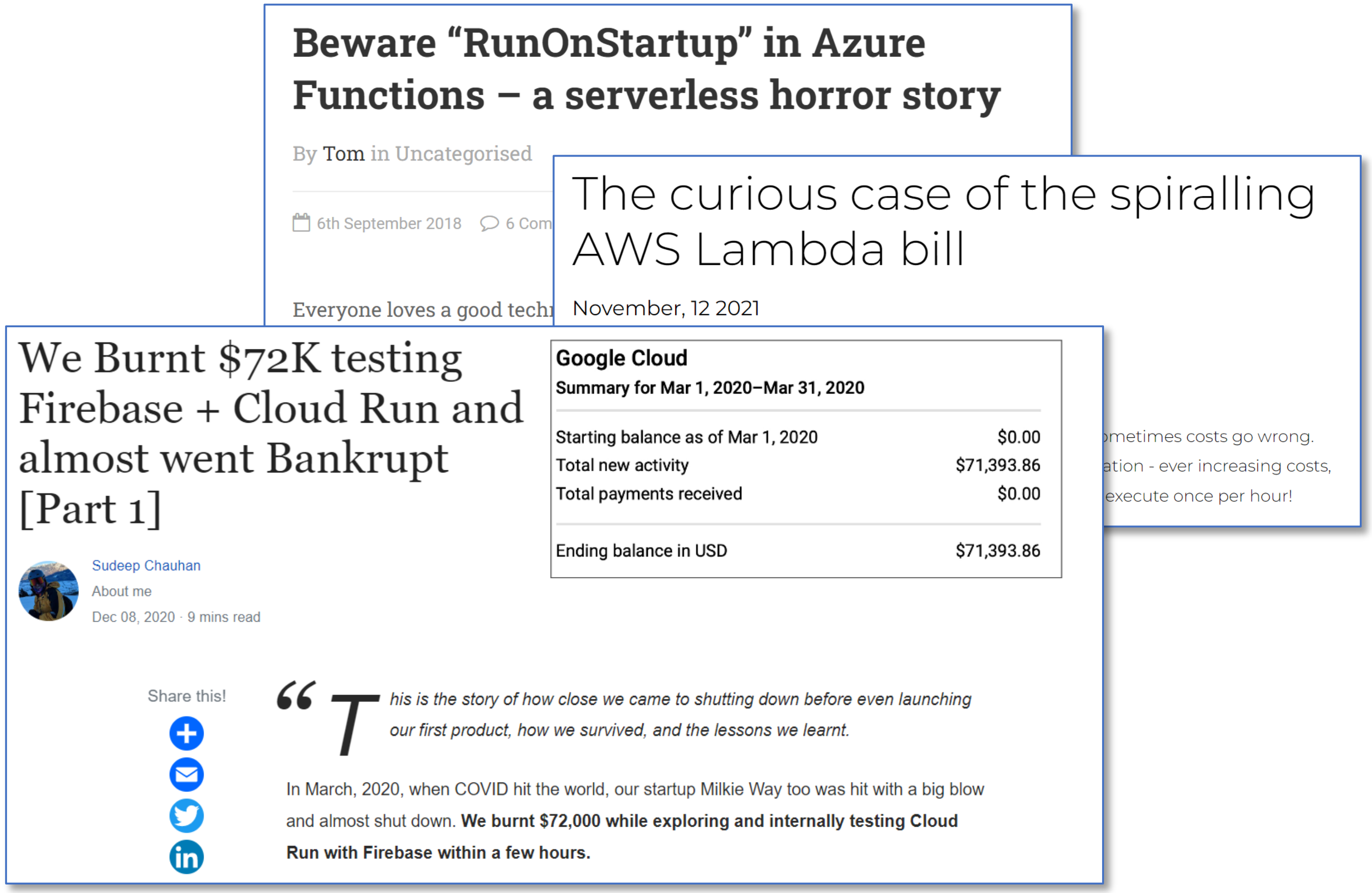
Flexible elastic scaling

Availability

Cost effectiveness

Serverless horror stories: A runaway function

- What is a runaway function?
- There are no default safeguards in place to prevent runaway functions



Beware "RunOnStartup" in Azure Functions – a serverless horror story

By Tom in Uncategorised

6th September 2018 6 Com

Everyone loves a good techi

The curious case of the spiralling AWS Lambda bill

November, 12 2021

metimes costs go wrong.
ation - ever increasing costs,
execute once per hour!

We Burnt $72K testing Firebase + Cloud Run and almost went Bankrupt [Part 1]

Sudeep Chauhan
About me
Dec 08, 2020 · 9 mins read

Google Cloud
Summary for Mar 1, 2020–Mar 31, 2020

Starting balance as of Mar 1, 2020	$0.00
Total new activity	$71,393.86
Total payments received	$0.00
Ending balance in USD	$71,393.86

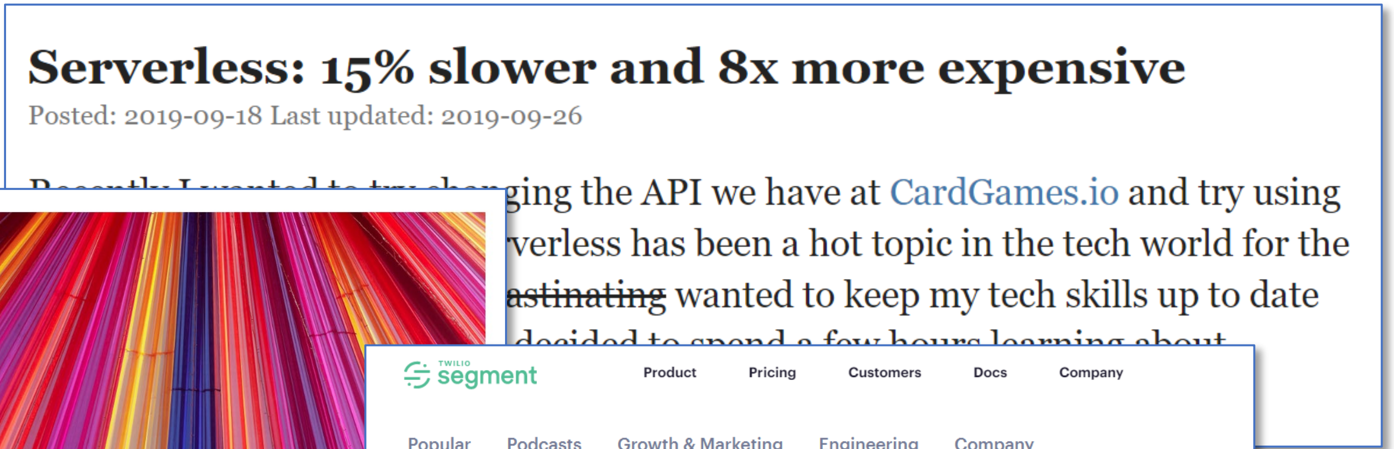
Share this!

"This is the story of how close we came to shutting down before even launching our first product, how we survived, and the lessons we learnt.

In March, 2020, when COVID hit the world, our startup Milkie Way too was hit with a big blow and almost shut down. **We burnt $72,000 while exploring and internally testing Cloud Run with Firebase within a few hours.**

Serverless horror stories: Service misconfiguration

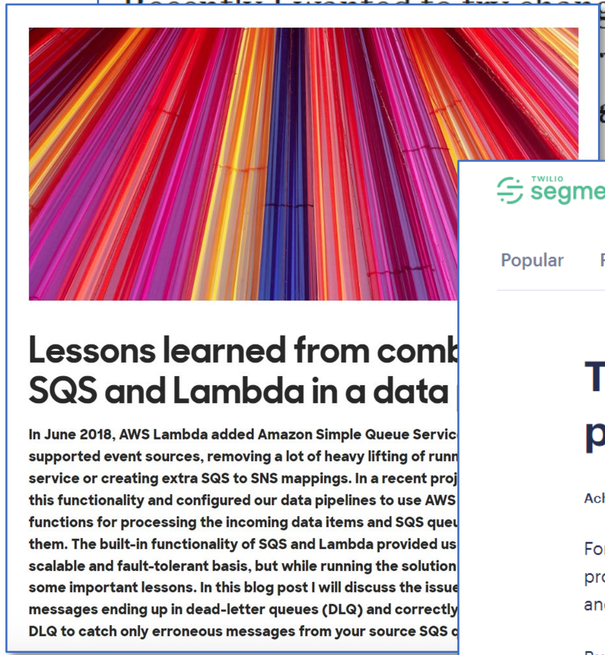
- Serverless functions don't live in isolation ("support BaaS services")
- Serverless says nothing about configuring & managing BaaS services
- A complex ecosystem with broad implications on
  - Costs
  - Performance
  - Reliability



Serverless: 15% slower and 8x more expensive

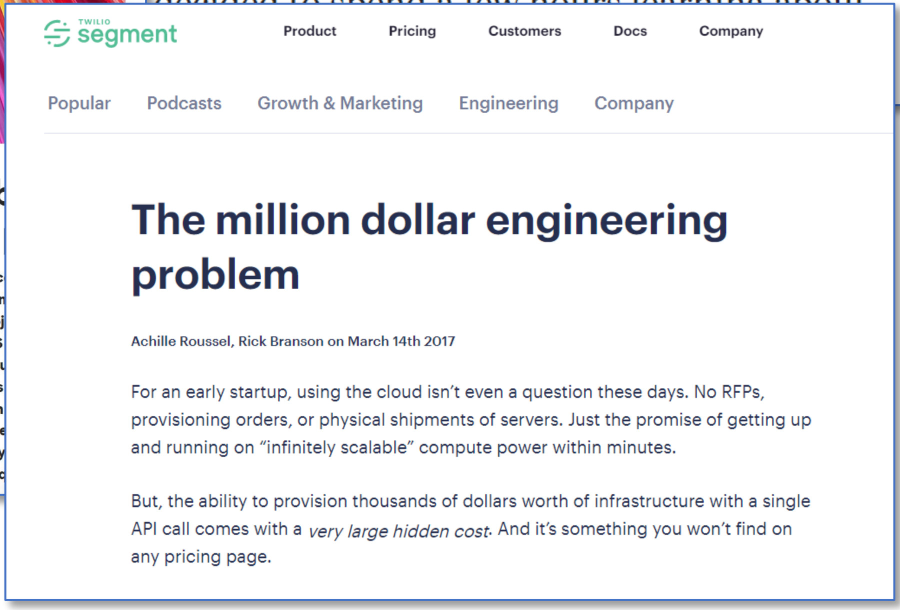
Posted: 2019-09-18 Last updated: 2019-09-26

ging the API we have at CardGames.io and try using
verless has been a hot topic in the tech world for the
~~astinating~~ wanted to keep my tech skills up to date



Lessons learned from comb
SQS and Lambda in a data

In June 2018, AWS Lambda added Amazon Simple Queue Servic
supported event sources, removing a lot of heavy lifting of runn
service or creating extra SQS to SNS mappings. In a recent proj
this functionality and configured our data pipelines to use AWS
functions for processing the incoming data items and SQS queu
them. The built-in functionality of SQS and Lambda provided us
scalable and fault-tolerant basis, but while running the solution
some important lessons. In this blog post I will discuss the issue
messages ending up in dead-letter queues (DLQ) and correctly
DLQ to catch only erroneous messages from your source SQS q



The million dollar engineering problem

Achille Roussel, Rick Branson on March 14th 2017

For an early startup, using the cloud isn't even a question these days. No RFPs, provisioning orders, or physical shipments of servers. Just the promise of getting up and running on "infinitely scalable" compute power within minutes.

But, the ability to provision thousands of dollars worth of infrastructure with a single API call comes with a *very large hidden cost*. And it's something you won't find on any pricing page.

Challenges with Serverless in the 3D Continuum

Performance Challenges

- **Function startup latency ("cold start")**
- Lack of performance isolation ("noisy neighbors")
- Inconsistent performance due to hardware heterogeneity (also present in Cloud only solutions!)
- Function scheduling and placing

Data Management Challenges

- **Sharing intermediate, ephemeral data in serverless workflows (fn1 → fn2)**
- State management for stateful computations
- Efficient and cost-effective caching solutions
- Function execution latency due to lack of data locality

Nastic, S., Dustdar, S., Philipp, R., Alireza, F., & Pusztai, T. (2022). A Serverless Computing Fabric for Edge & Cloud. In 4th IEEE International Conference on Cognitive Machine Intelligence (CogMi).

Challenges with Serverless in the 3D Continuum

Reliability Engineering Challenges

- SLO-aware provisioning of functions (beyond memory and CPU)
- Dealing with function failures considering SLOs (beyond simple retries)
- Network partitioning can render functions useless (detached storage)
- Interoperability and portability of functions (federated serverless computing, multi-cloud, Sky computing)

Software Development Challenges

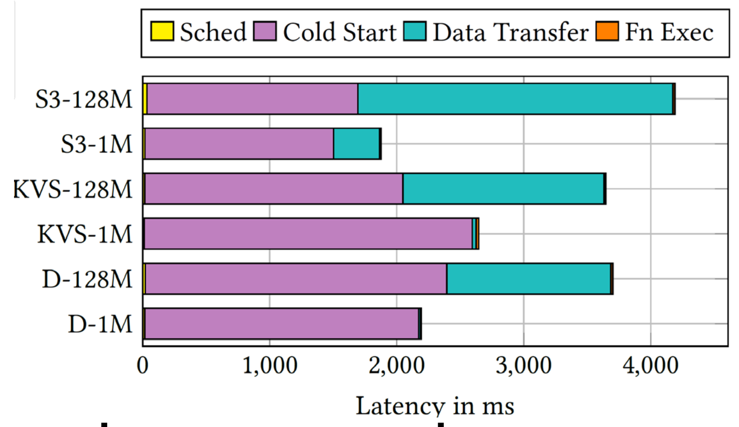
- A misconfiguration of services and resources
- Bad coding practices
- Wrong or incomplete error handling mechanism
- Testing of the functions, beyond the unit tests
- Resource quotas & limits

Nastic, S., Dustdar, S., Philipp, R., Alireza, F., & Pusztai, T. (2022). A Serverless Computing Fabric for Edge & Cloud. In 4th IEEE International Conference on Cognitive Machine Intelligence (CogMi).

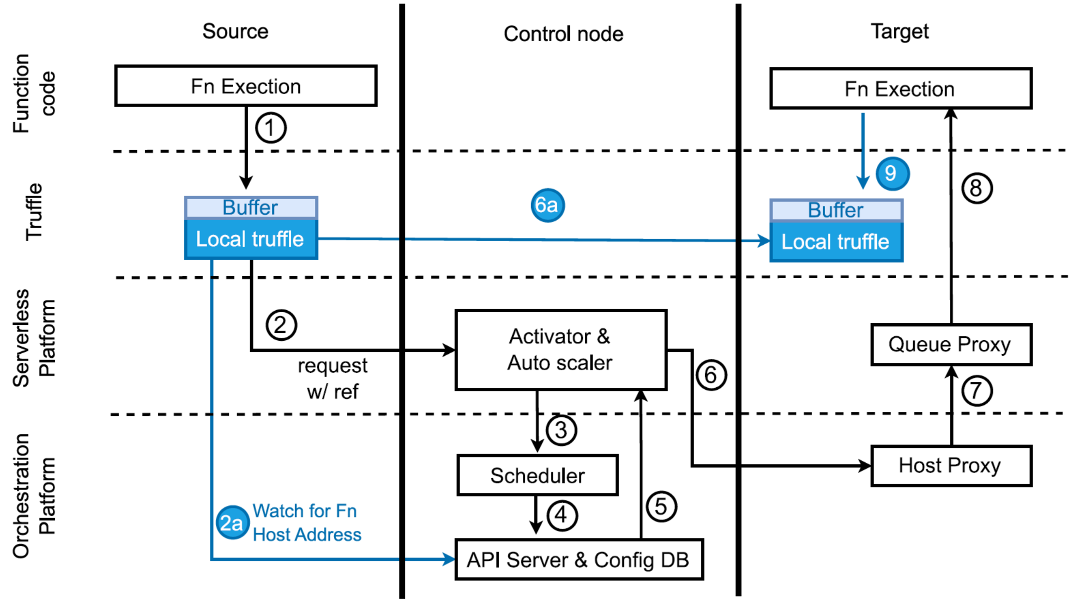
Leveraging cold starts for efficient data passing

- Main intuition: use cold start time to **do useful work vs wasting compute and energy**!
  - Why →
- Truffle is a solution that currently supports 2 modes:
  - State prefetching from storage
  - Inter-function data passing →
- Function exec path (black arrows)
  - 1) send request, 2) forward request, 3-8) platform internal invocation process
- Data prefetching path (blue arrows)
  - 2a) register watcher, 6a) transfer the data (payload), 9) fetch data from local Truffle
  - Note 6a happens in parallel to the steps 6-8

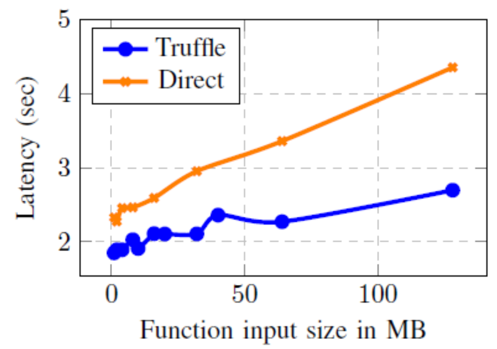
Marcelino, C., & Nastic, S. (2024). Truffle: Efficient Data Passing for Data-Intensive Serverless Workflows in the Edge-Cloud Continuum. In Proceedings of the IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC '24).



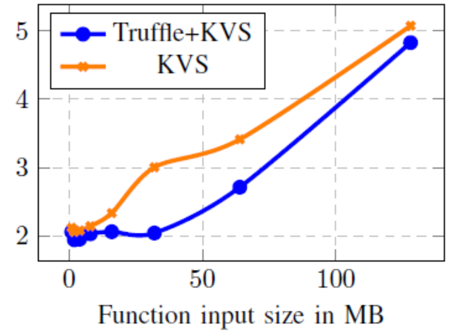
* Running over 100 Mbit network in the Edge-Cloud



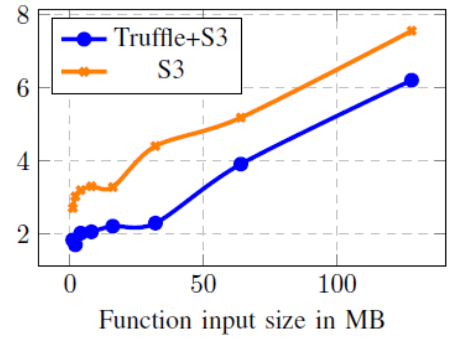
Truffle results



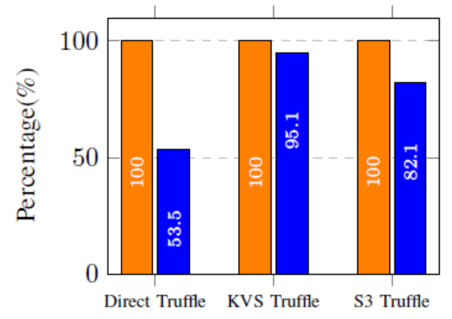
(a) Direct Data Passing



(b) KVS Data Passing

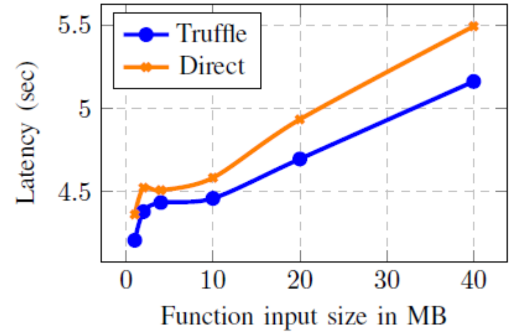


(c) S3 Data Passing

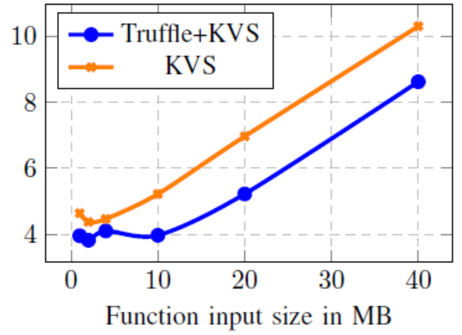


(d) Normalized Latency at 128MB

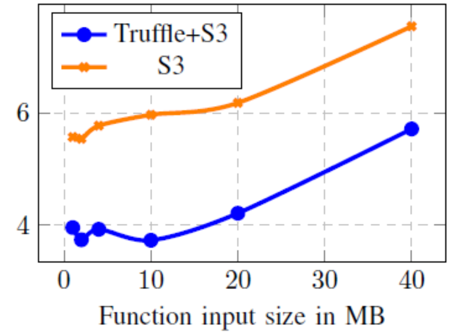
Chained functions latency



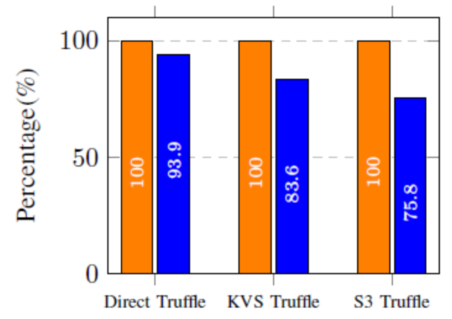
(a) Direct Data Passing



(b) KVS Data Passing



(c) S3 Data Passing



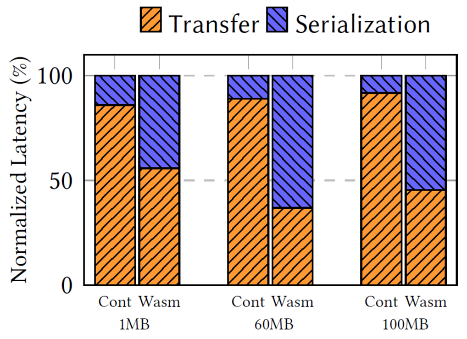
(d) Normalized Latency at 40MB

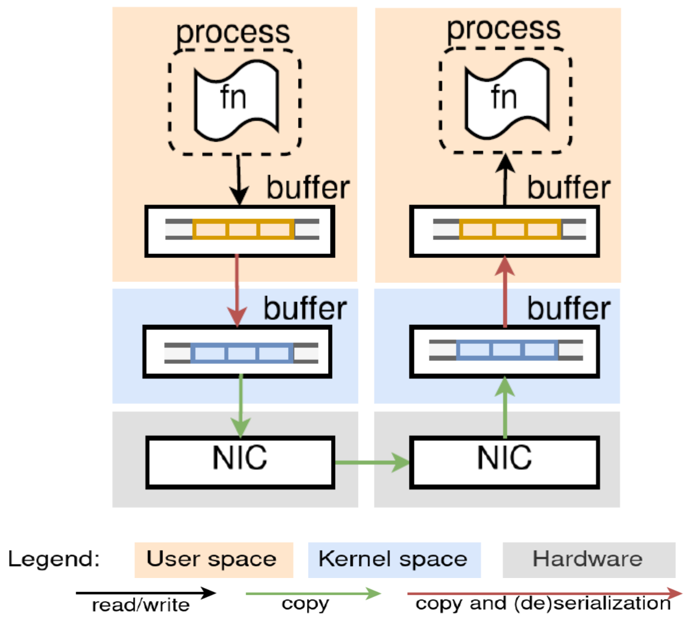
Video analytics latency

Minimal-copy serialization-free data transfers

- Data transfer particularly affects **stateless workflows** (Fn1→ Fn2)
  - Data serialization imposes overhead →
  - Data copying across user/kernel/network spaces imposes overhead (context switch)

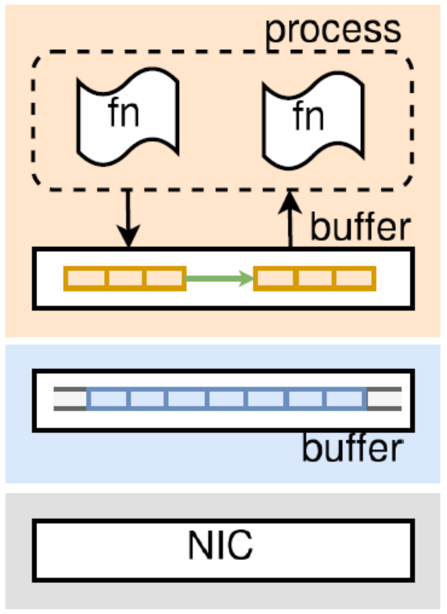
- Traditional data transfer mechanisms require several (de)serialization and copy operations (sys calls) →



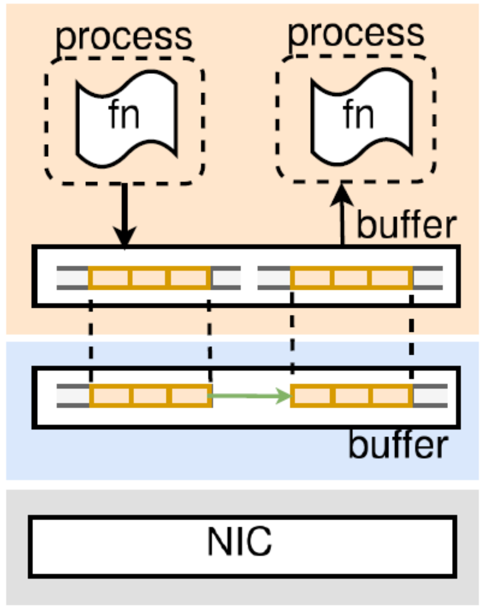


Minimal-copy serialization-free data transfers

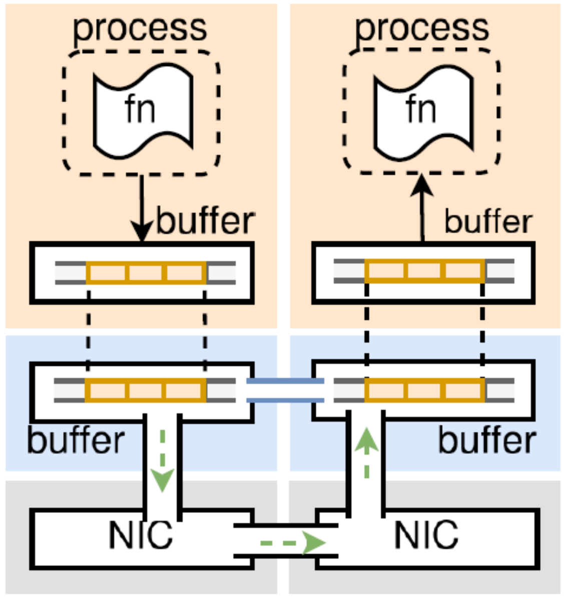
- Roadrunner defines 3 modes of data transfer between serverless functions and manages the functions' memory via shims
- Data transfer is automatically determined based on the functions' deployment



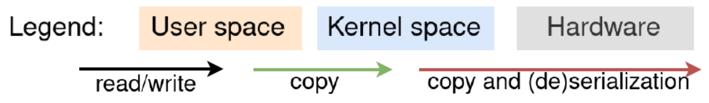
M1: In sandbox data transfer



M2: Kernel-space data transfer



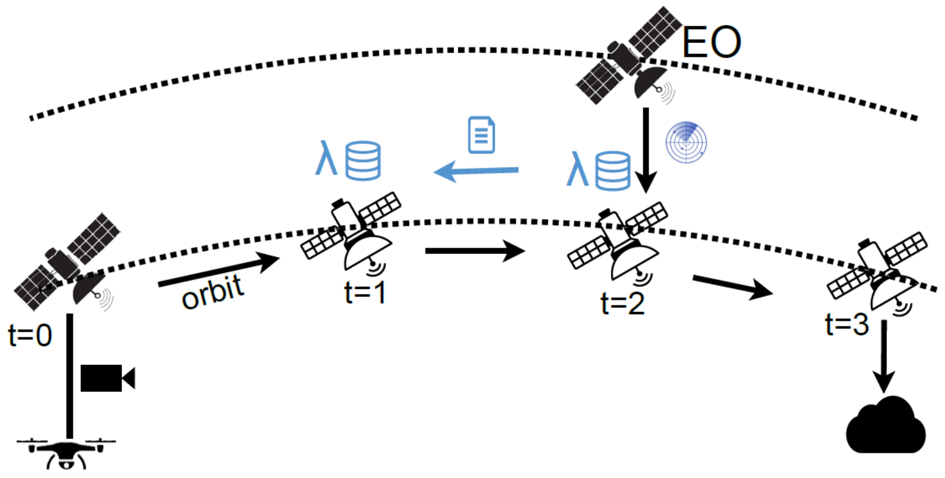
M3: Network data transfer

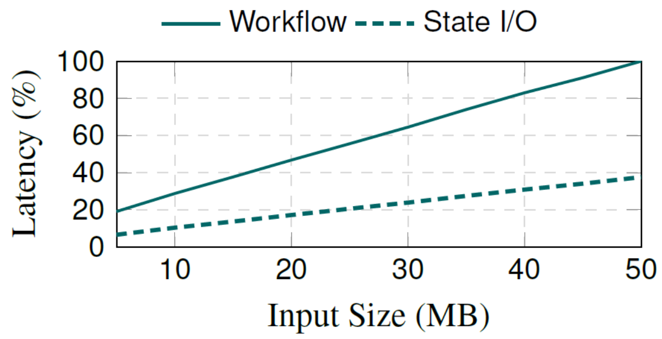


Marcelino, C., Pusztai, T., & Nastic, S. (2025). Roadrunner: Accelerating Data Delivery to WebAssembly-Based Serverless Functions. In Proceedings of the 26th ACM/IFIP International Middleware Conference (MIDDLEWARE 2025).

Dealing with infrastructure hypermobility

- Intuition: Propagate function state to "counteract" orbital movements →
- Databelt introduces an **SLO-aware state migration/propagation** mechanism that enables the function state to move continuously in orbit regardless of the movement of individual satellites





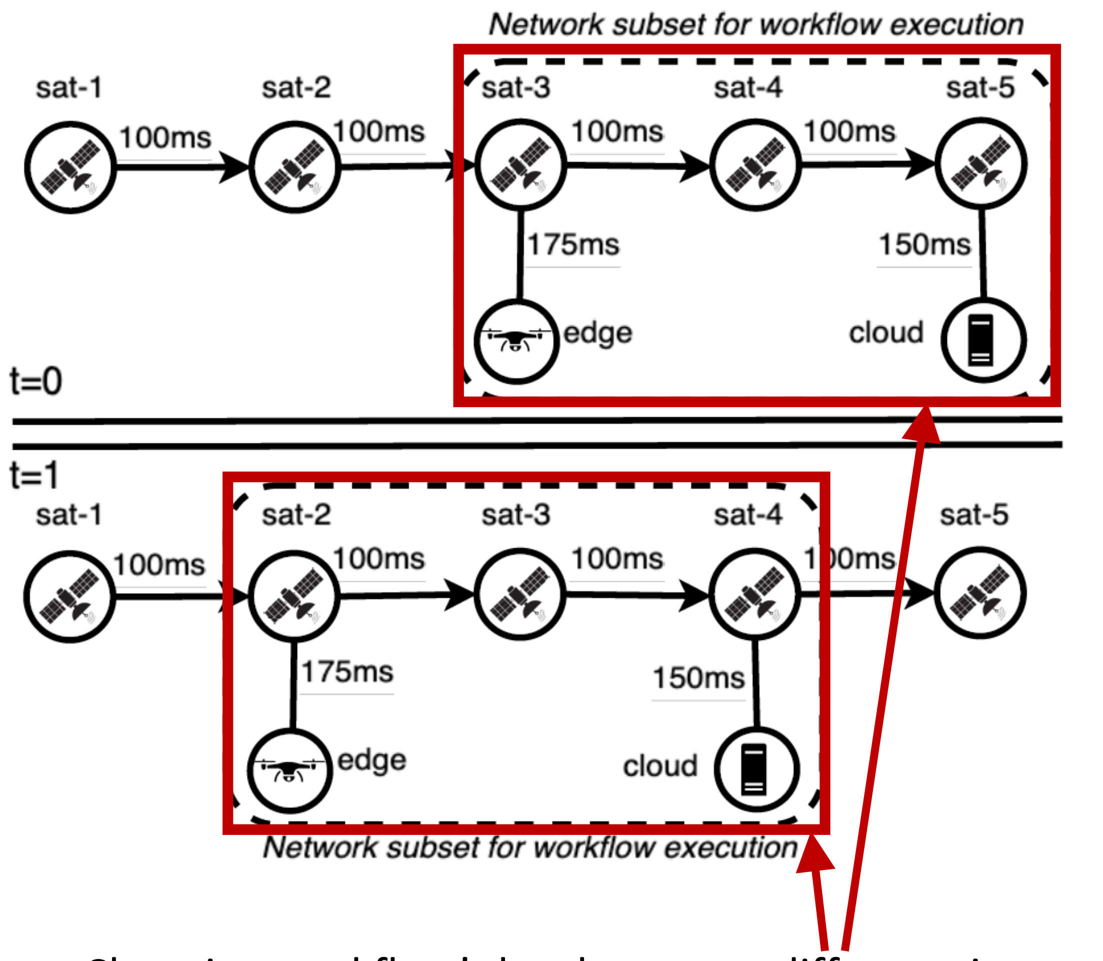
Normalized workflow duration vs. state loading

Cynthia Marcelino, Leonard Guelmino, Thomas Pusztai, Stefan Nastic. (2025) Databelt: A Continuous Data Path for Serverless Workflows in the 3D Compute Continuum. Journal of Systems Architecture (to appear)

Databelt in the 3D Continuum

Identify → Compute → Offload

1. Construct a real-time infra topology graph of all execution nodes
2. Select the most appropriate set of nodes (satellites), and compute the shortest path to each one
   - Predict which ones are likely to go out of range based on orbital dynamics
3. Proactively move the function state (data) to the best candidate node



Changing workflow’s local group at different times t

Constraints for in-orbit data placement

- Requirements to consider when scheduling and deploying functions or data to satellites
- Thermal management
  - Satellites are frequently exposed to solar radiation, hence temperature fluctuates from −120°C in shadow to +120°C in sunlight
  - Heat dissipation only possible via radiation (no conduction or convection)
- Energy Capacity
  - Satellites are usually solar powered (recharge in sunlight)
  - Executing a function must not deplete battery!
- Many more →

Formulated as hard/soft constraints which define the operational space under which a function can execute or state can be persisted/retrieved

$$T^n_{\text{orb}} + \sum_{i \in \mathcal{F}} T^{in}_{\text{exc}} \leq T^n_{\text{max}} \quad \forall n \in \mathcal{N}$$

$$\sum_{i \in \mathcal{F}} P_i \cdot x_{i,n} \leq P^n_{\text{avail}} \quad \forall n \in \mathcal{N}$$

T. Pusztai, C. Marcelino, S. Nastic, Hyperdrive: Scheduling serverless functions in the edge-cloud-space 3d continuum. In 2024 IEEE/ACM Symposium on Edge Computing (SEC), 2024, pp. 265–278.

Conclusions

1. **3D Compute Continuum** as the next generation of compute infrastructures from Cloud→Edge→Satellites
2. **Serverless computing** as the next step in the evolution of the distributed computing
3. The capabilities of the **Serverless 3D Continuum** enable future applications to effectively manage trade-offs between performance, scalability, reliability, and cost efficiency

Thank you for your attention!

Asst. Prof. Stefan Nastic

snastic@dsg.tuwien.ac.at

https://dsg.tuwien.ac.at