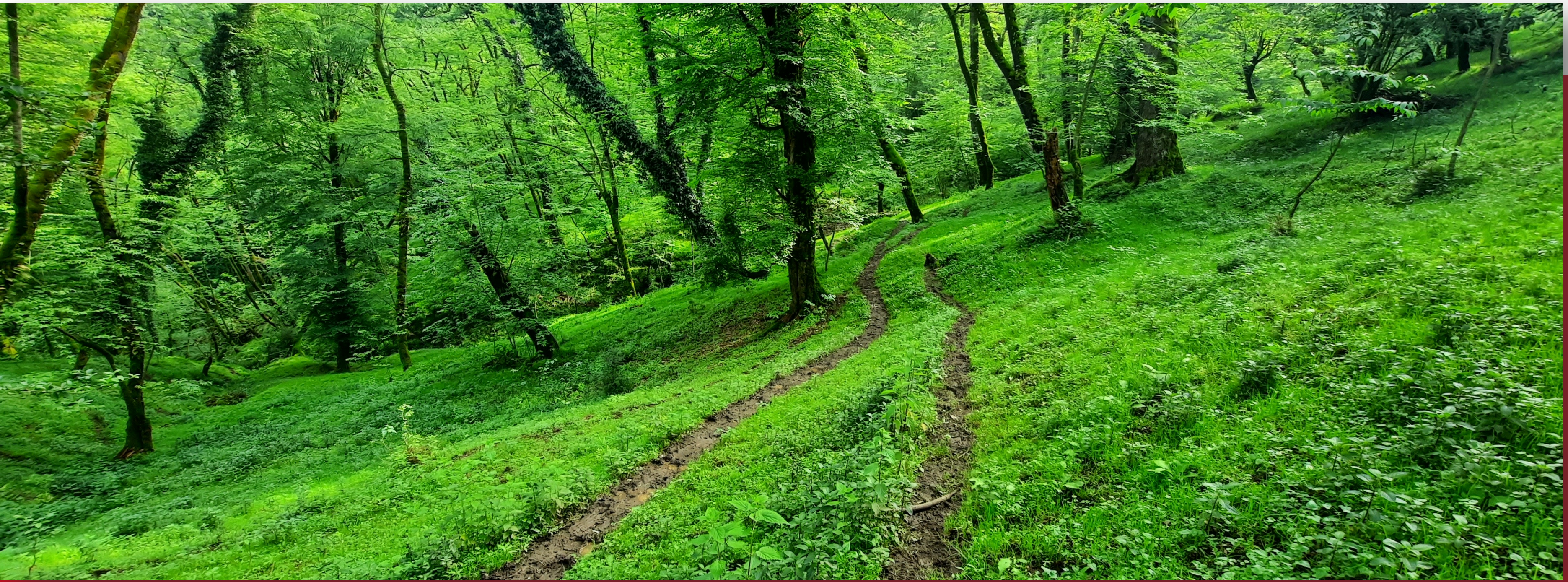
WOSCx3

TECHNISCHE UNIVERSITÄT BERLIN



GeoFaaS: An Edge-to-Cloud FaaS Platform

M. Malekabbasi, T. Pfandzelter, T. Schirmer, D. Bermbach | Scalable Software Systems

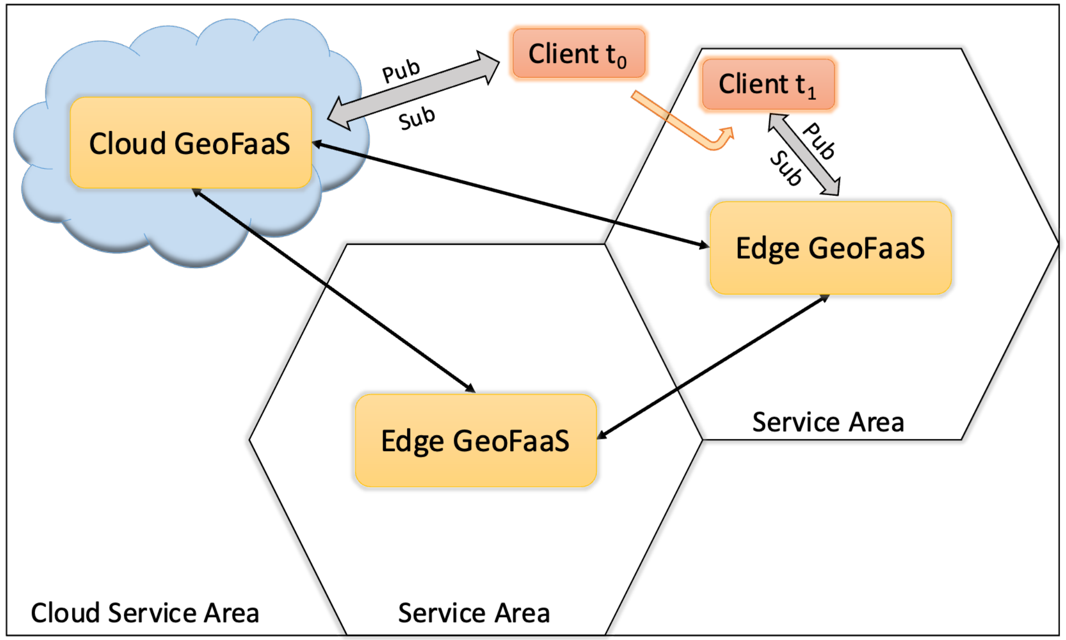
Edge-to-Cloud FaaS & Serverless

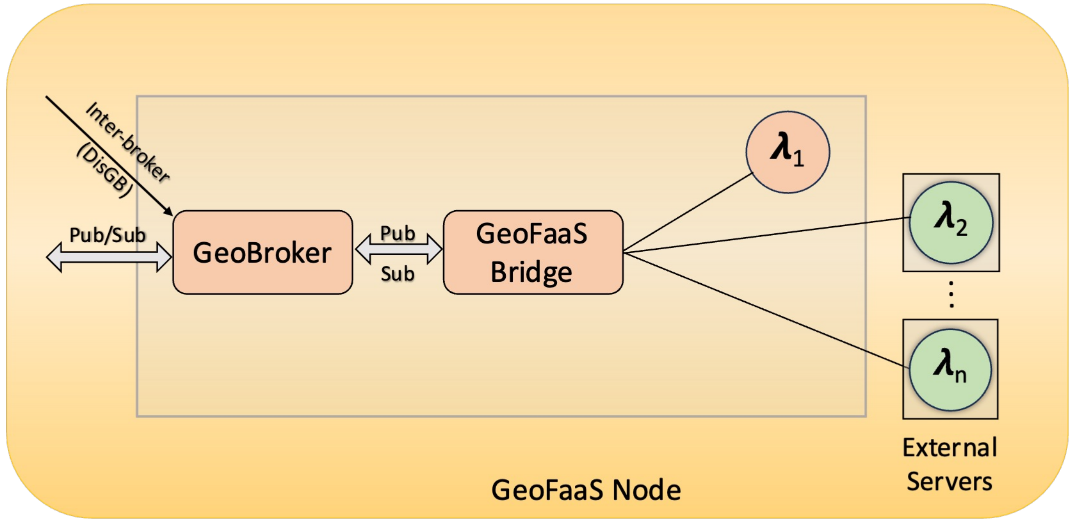
- Imagine (near) future with mobile clients
- Their characteristics:
  - Geographically-distributed
  - Limited battery and computation power
  - many latency-sensitive applications
- Problem: edge-to-cloud systems complex dev & management.
- Solution: FaaS abstraction is promising. Now even stateful [11].

- BUT, we are not considering client's location!
  - In a changing network, physical distance effectively approximates latency [13], [14].
  - Client location measurement is cheap (no interaction with the system)

Geo-aware Function-as-a-Service

- A geo-distributed FaaS platform, across e2c continuum
- Transparency for end clients
  - i.e. publish on *"f1/call"* to call f_1
  - Through a client library for the common serverless abstraction
- GeoFaaS node has three key elements
  - (Distributed) Geo-aware Message Broker ("DisGB")
  - FaaS server(s) (local or in the same data center)
  - Bridge, a middleware between the other two



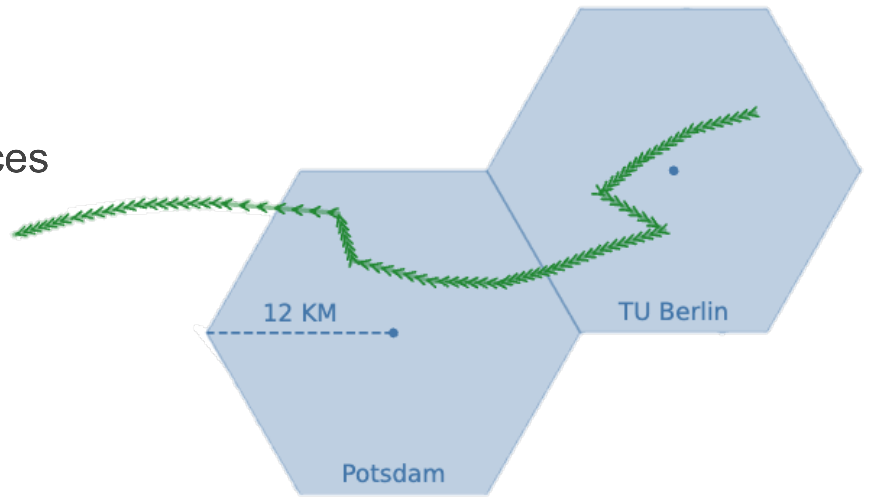


Evaluation

- Prototype:
  - Kotlin prototype and client library.
  - Topics for a sample function f_1 (Table I)
- Experiment setup:
  - 2 *RPi*[1] Edge nodes, and one *GCP*[2] Cloud node (Figure)
  - Each node, running tinyFaaS, GeoBroker and GeoFaaS Bridge instances
  - All clients on a *Rpi*, in the same network as the edges
- Scenarios:
  1) "Distance/Latency Change"
  2) "High Load", transparent offloading for uninterrupted service
  3) "Outage", system's resilience against GeoFaaS Bridge failure

TABLE I
TOPICS CREATED BY *GeoFaaS* FOR THE $f1$ FUNCTION.

#	Topic	Explanation
1	`/f1/call`	Client calls function (Bridge subscribes)
2	`/f1/ack`	Bridge acknowledges call (client subscribes)
3	`/f1/result`	Client subscribes for result (Bridge publishes)
4	`/f1/nack`	Edge Bridge offloads call (Cloud subscribes)
5	`/f1/call/retry`	Client direct cloud call (Cloud subscribes)



1. Raspberry Pi 4 B
2. Google Cloud Platform VM

Source & Experiments:
Github.com/OpenFogStack/GeoFaaS

Reults

- FaaS’s physical distance impacts response times. GeoFaaS effectively routes clients to the nearest servers.
- GeoFaaS offloads requests for transparent client responses under high load.
- GeoFaaS with reliable DisGB routes to cloud in case of edge (internal) failure.

- Full results in our paper (under review)
  - http://arxiv.org/abs/2405.14413

mm@3s.tu-berlin.de

References

[3] Pfandzelter, Tobias, and David Bermbach. "tinyfaas: A lightweight faas platform for edge environments." 2020 IEEE International Conference on Fog Computing (ICFC). IEEE, 2020.

[5] Russo, Gabriele Russo, et al. "Serverledge: Decentralized function-as-a-service for the edge-cloud continuum." 2023 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, 2023.

[6] Oliveira, Bárbara, et al. "Function-as-a-Service for the Cloud-to-Thing continuum: a Systematic Mapping Study." 8th International Conference on Internet of Things, Big Data and Security-IoTBDS. 2023.

[13] Hasenburg, Jonathan, and David Bermbach. "DisGB: Using geo-context information for efficient routing in geo-distributed pub/sub systems." 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC). IEEE, 2020.

[18] Ciavotta, Michele, et al. "DFaaS: Decentralized function-as-a-service for federated edge computing." 2021 IEEE 10th International Conference on Cloud Networking (CloudNet). IEEE, 2021.