FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

DISTRIBUTED SYSTEMS GROUP

FogFrame: IoT service deployment and execution in the fog

Olena Skarlat,
Kevin Bachmann, Stefan Schulte

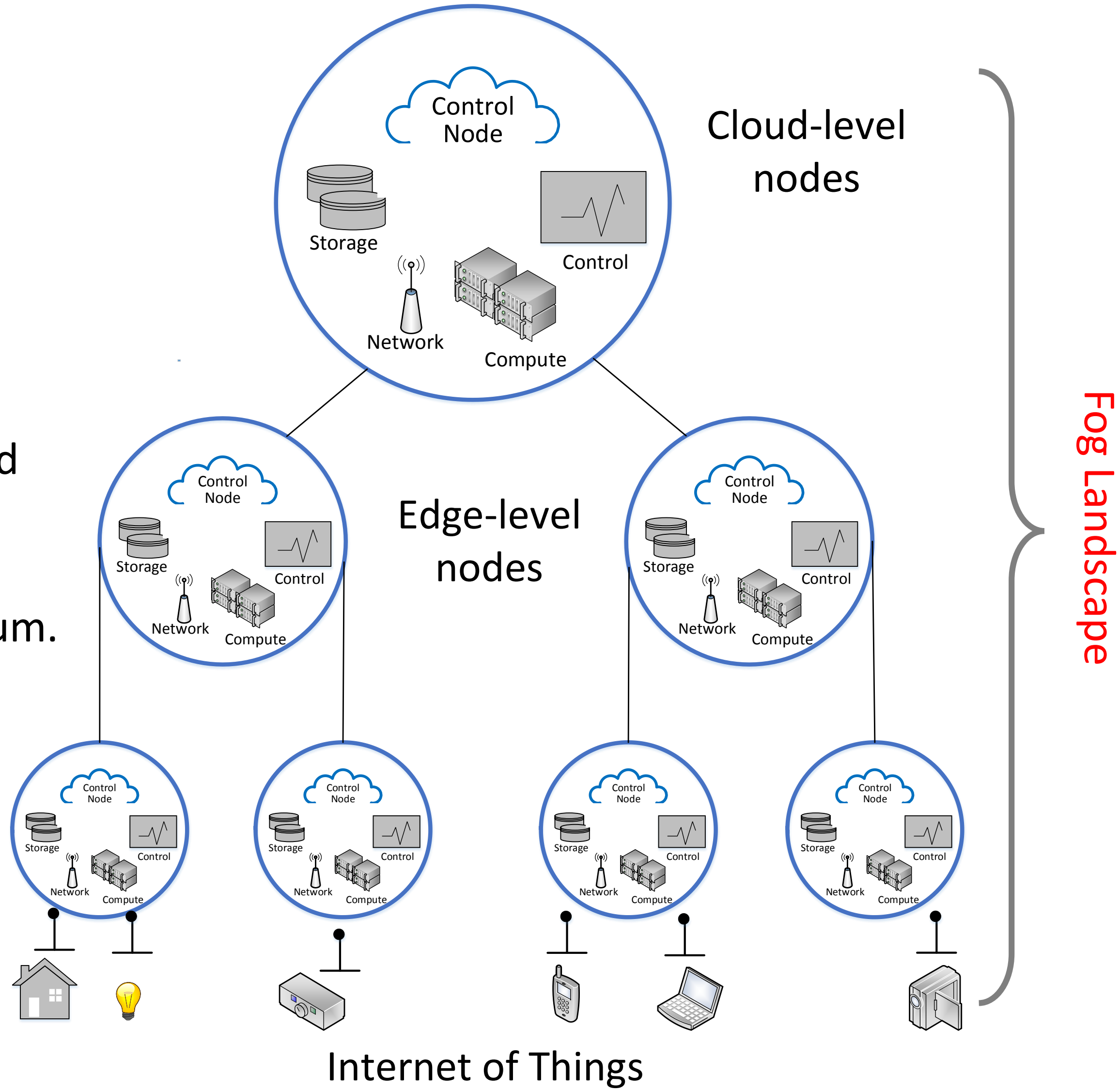
KuVS-Fachgespräch Fog Computing 2018

Distributed Systems Group, TU Wien, Austria
http://www.infosys.tuwien.ac.at/

What is a Fog Landscape?

OpenFog

A horizontal,
system-level architecture that distributes
computing, storage, control and networking functions
closer to the users
along a cloud-to-thing continuum.

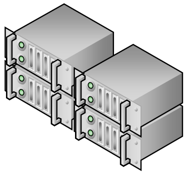


Research Questions

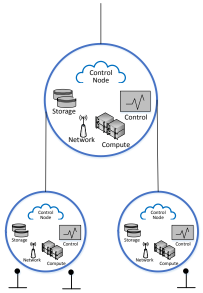
Challenge: to create and support an execution environment for IoT applications in the fog landscape.

- What are the mechanisms to provide virtualization of resources?

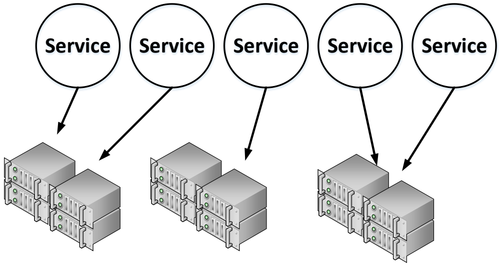
Compute Units



- What are the methodologies and tools to realize software that manages a fog landscape and executes services?



- How to perform and optimize resource provisioning and execute services?



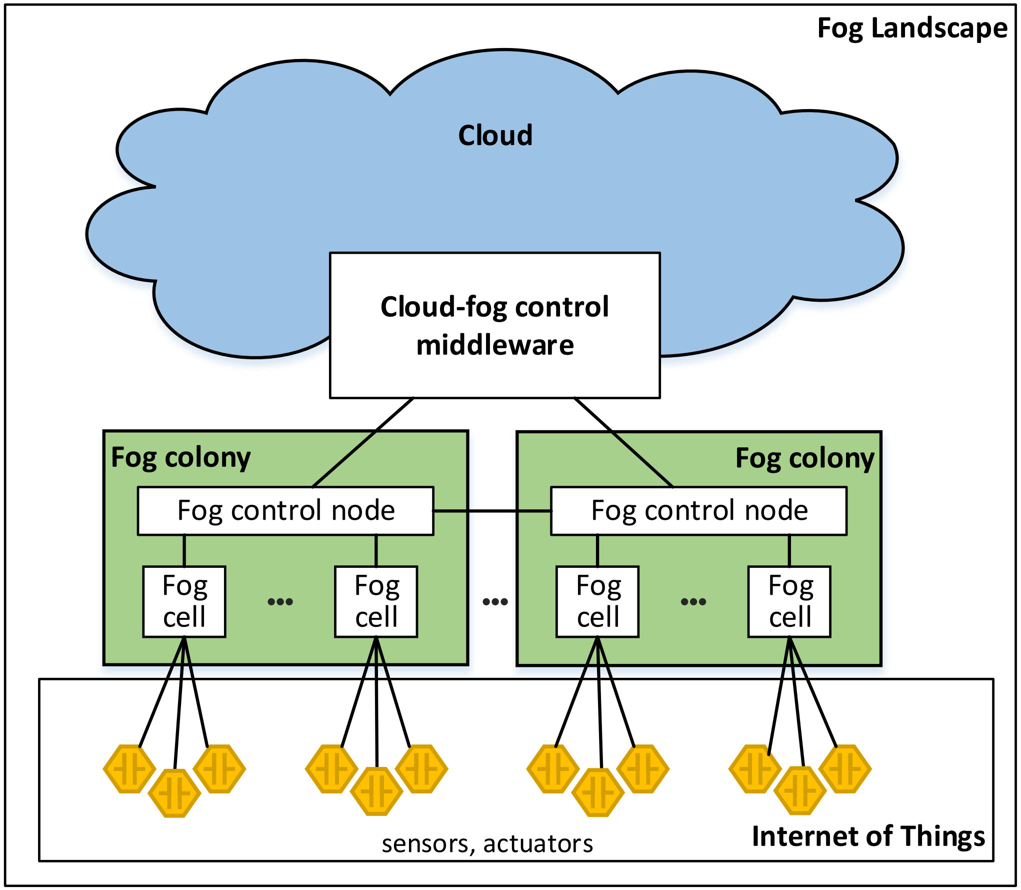
Fog Landscape Resource Model

Fog cells control IoT devices (sensors, actuators), can execute services

Fog control nodes control fog cells and execute services

A fog control node and connected fog cells form a fog colony, acts as a micro data center

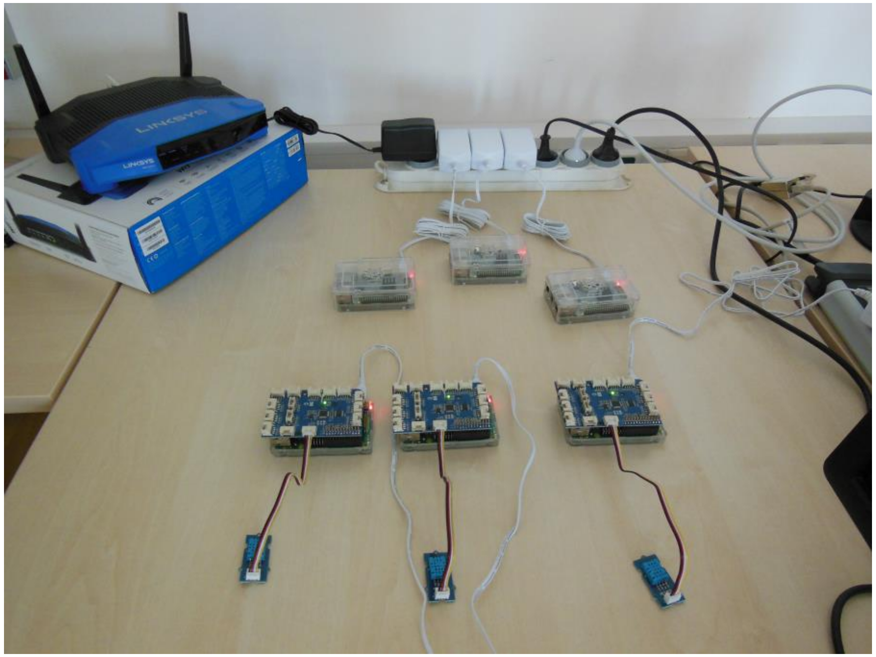
Cloud-fog middleware manages cloud resources and supports fog colonies

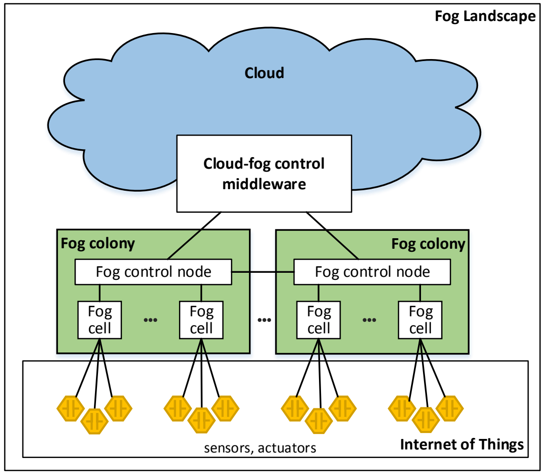


FogFrame: A Fog Computing Framework

Functionality of FogFrame:

- Coordinated control over a fog landscape
- Monitoring and analysis of resources
- Service placement plan
- Deploy and execute applications
- (Re-)configuration of the fog landscape based on runtime events

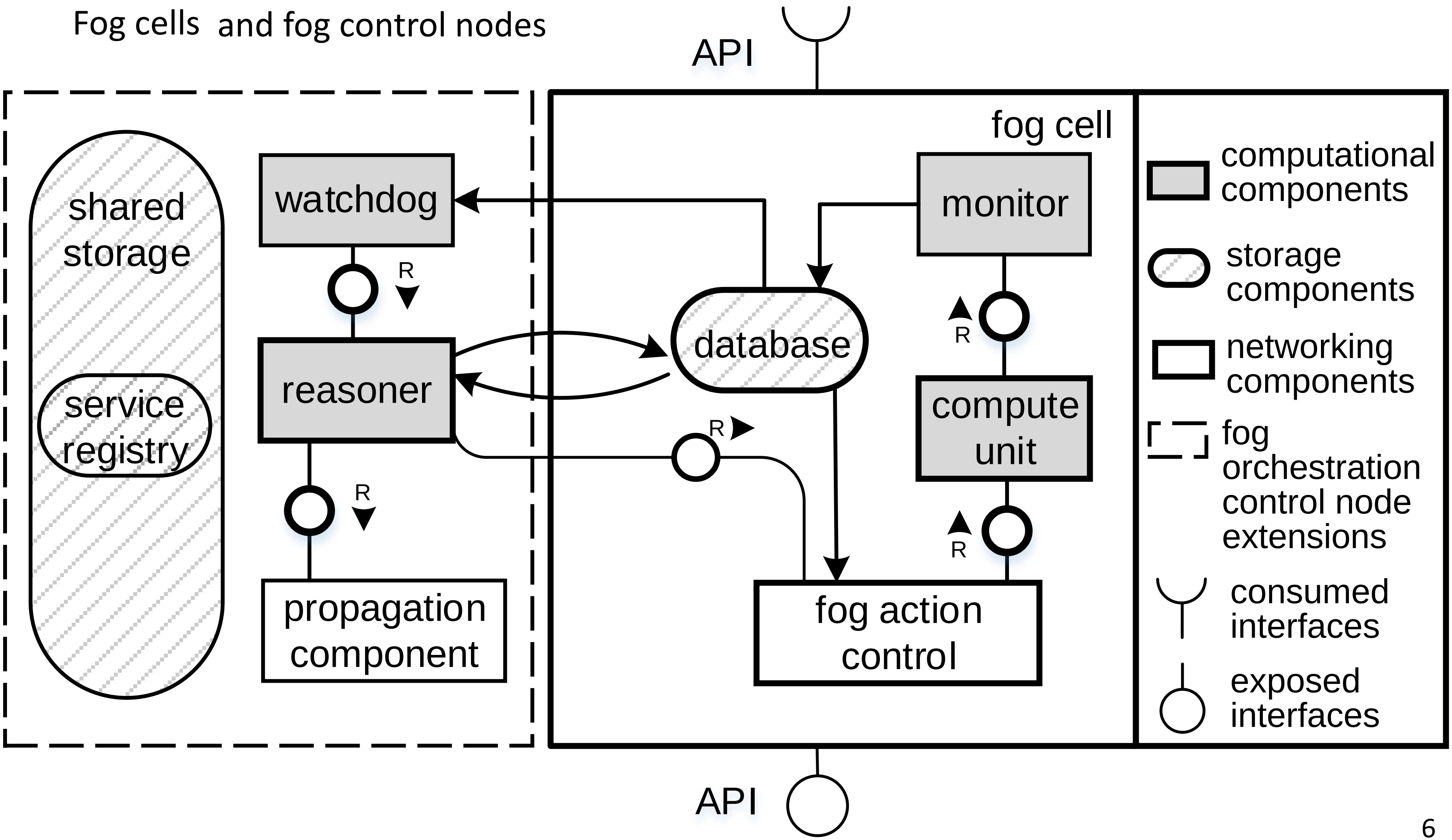




https://github.com/keyban/fogframe

FogFrame Architecture

Fog cells and fog control nodes

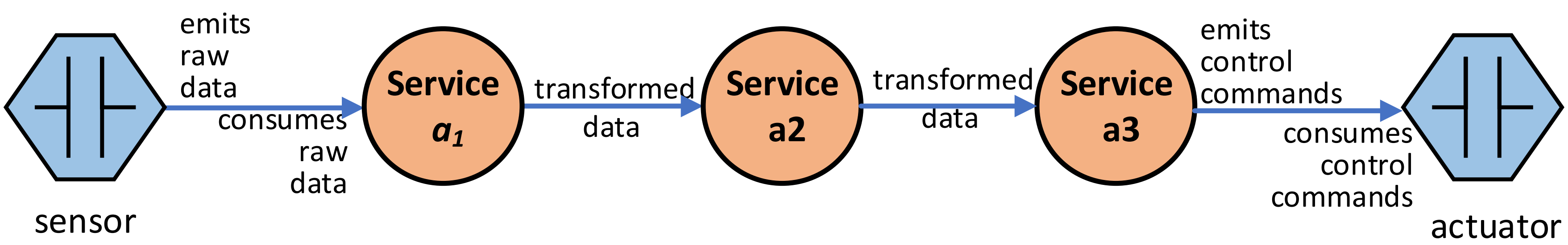


Application Model

Distributed Data Flow* application

- Quality of Service (QoS) requirements, e.g., deadline on deployment and execution time
- Set of services to be deployed
- Each service is characterized by demands in CPU, RAM, and storage,

  its service type (e.g., a certain sensor equipment is needed, or it is a purely cloud service)



*N. K. Giang, M. Blackstock, R. Lea, and V. C. Leung, "Developing IoT Applications in the Fog: a Distributed Dataflow Approach," in 5th International Conference on the Internet of Things (IoT). IEEE, 2015, pp. 155–162.

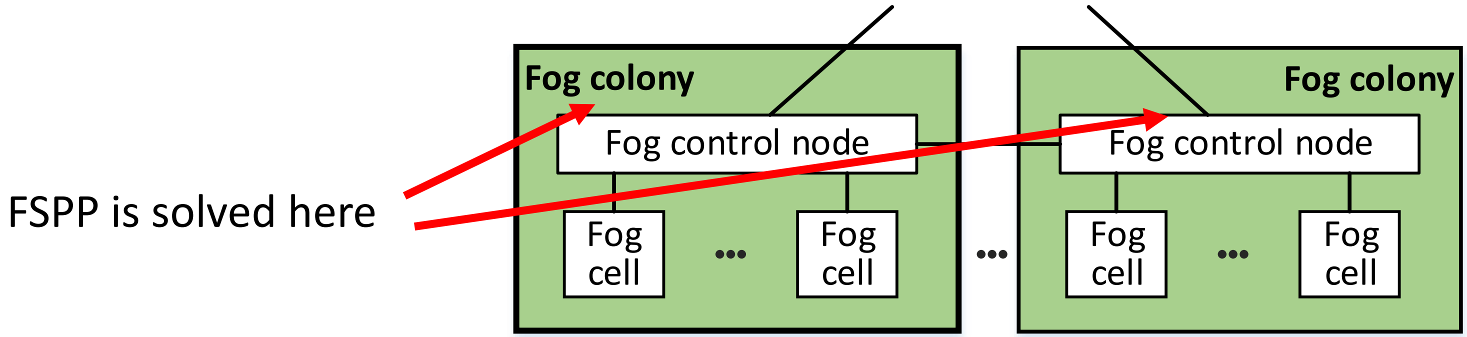
Service placement

Fog Service Placement Problem (FSPP)

Goal: to produce a service placement plan which maximizes the utilization of fog colonies while satisfying QoS

FSPP is solved by each head fog control node:

- Which services have to be executed in its own fog colony?
- Which services have to be executed locally on own resources of fog control node?
- Which services have to be propagated to the closest neighbor colony?
- Which services have to be propagated to the cloud?



Fog Service Placement Problem (FSPP)

Variables: decision variables

Goal: maximize fog colony resources utilization, while adhering to QoS parameters

$$\max \sum_{A_k}^{A} P(A_k) \left(\sum_{a_i}^{A_k} \left(\sum_{f_j}^{Res_{a_i}(F)} x_{a_i}^{f_j} + x_{a_i}^{F} \right) + |A_k| y_{A_k} \right) \qquad P(A_k) = \frac{1}{D_{A_k} - w_{A_k}^{t}}$$

Constraints:

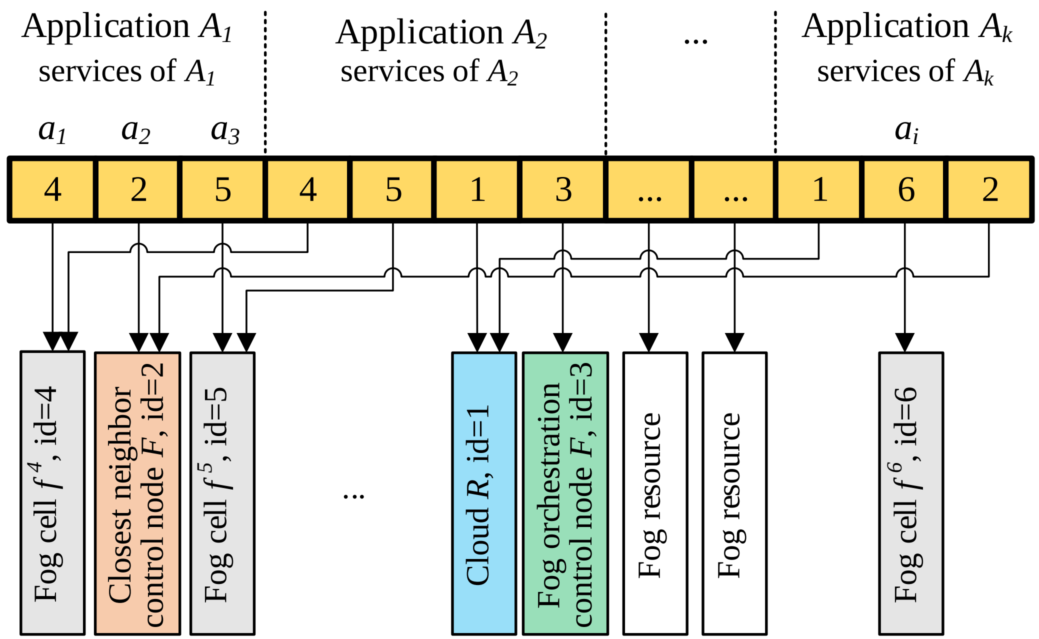
- Resource capacities
- Number of deployed containers
- Adherence to QoS
- Propagation

Solutions of FSPP

- Exact mathematical method
- First-fit heuristic algorithm
- Genetic algorithm

Solutions of FSPP - Genetic Algorithm

Chromosome:



Fitness: encourage if a chromosome fulfills constrains, apply penalties upon violations, „death penalties"

$$F(c) = \sum_{\beta_p \in \Psi} \omega_{\beta_p}(1 - 2\delta_{\beta_p(c)}) + \sum_{\beta_\gamma \in \Gamma} \omega_{\beta_\gamma}(1 - 2\delta_{\beta_\gamma(c)}) - \omega_p D(c)$$

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Solutions of FSPP - Genetic Algorithm

Parameters:

- 80%-uniform crossover
- Tournament selection
- 2% random gene mutation
- 20% elitism rate
- Population size of 1000 individuals

Stopping condition:

- Positive fitness (no death penalties applied)
- Tolerance value: dividing the incremental variance of the fitness values by the maximum fitness value over generations

Service Deployment

Challenge: how to deploy services in the heterogeneous environment of a fog landscape?

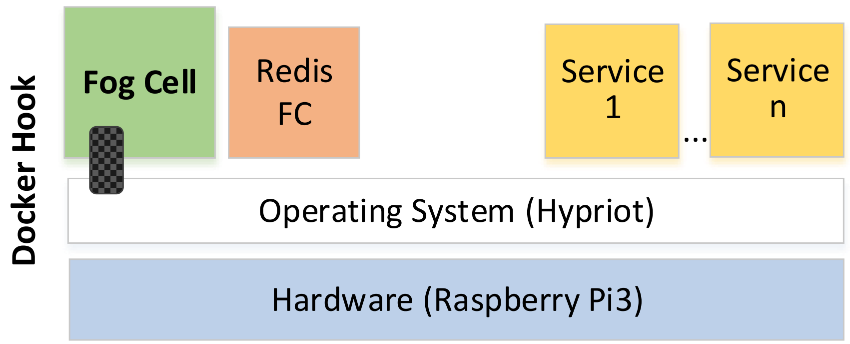
- Deployment in the cloud
- Deployment in the fog colonies

	Cloud	Fog colonies
Resources	VMs	Raspberry Pis
Processor architecture	64-bit Amazon Machine Image	ARM
Operating system	CoreOS	Hypriot
Base Docker image	FROM java:8	FROM hypriot/rpi-java
Service storage	Docker Hub	Shared storage

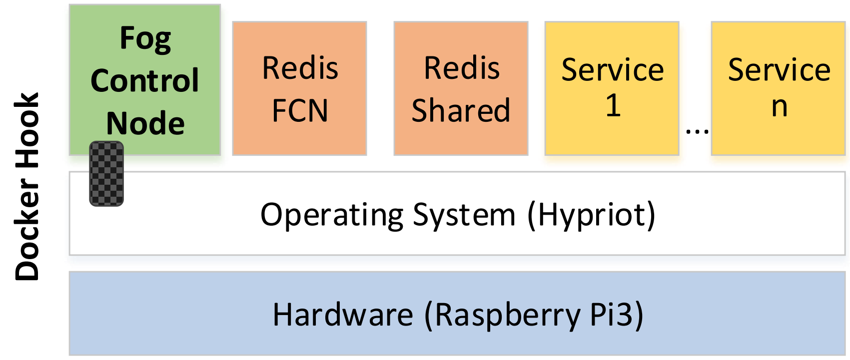
Service Deployment in Fog Colonies

Docker Hook instead of Docker-inside-Docker

Fog cells (FC)
- Redis FC container with a local database



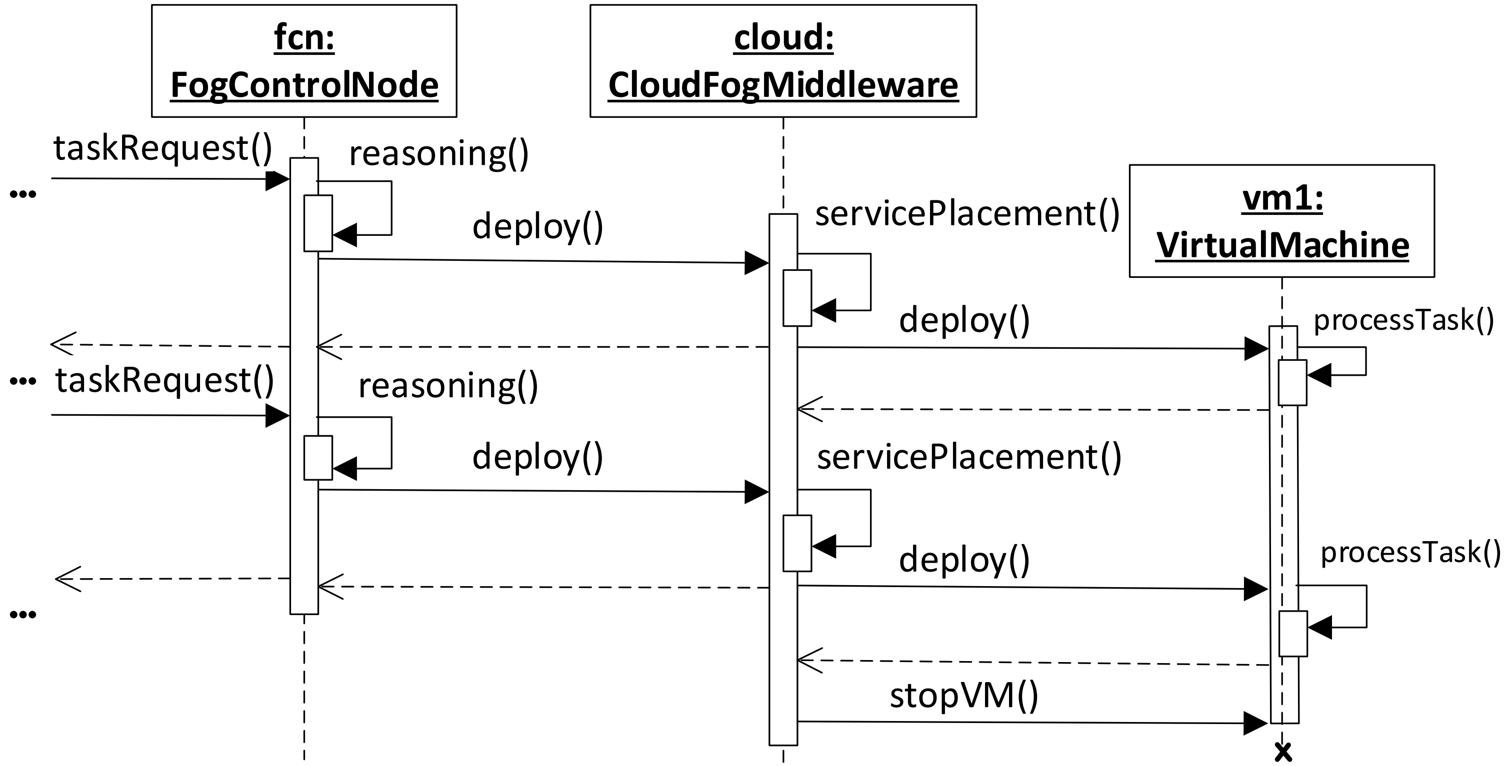
Fog control nodes (FCN)
- Redis FCN with a local database
- Redis Shared with the shared repository of service images



Service Deployment in the Cloud

AWS cloud (Openstack cloud)

CoreOS with Docker runtime preconfigured



Evaluation

Metrics:

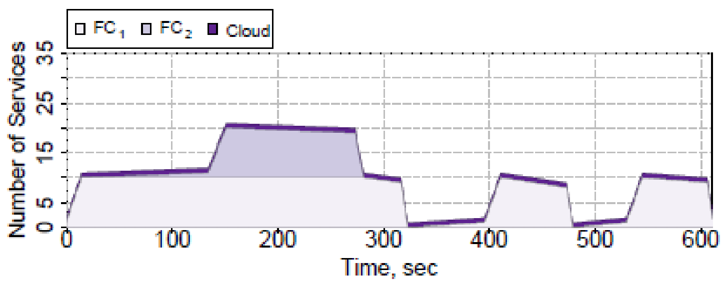
- Deployment time
- Utilization of resources (deployed containers)

Scenarios:

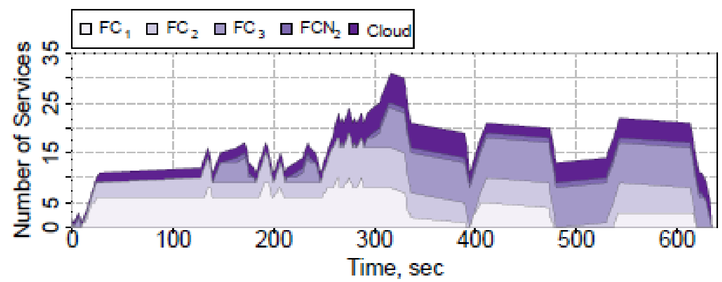
- Assessment of deployment time
- Different arrival patterns of application requests
- Different service placement algorithms
- Reaction on runtime events

Evaluation: different algorithms, arrival patterns

Constant arrival pattern of application requests: 10 services each 2 minutes

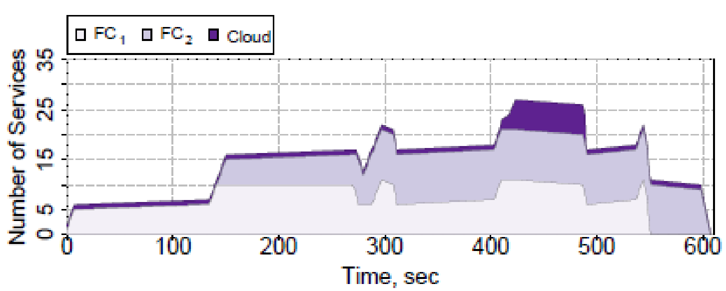


(a) First-fit algorithm

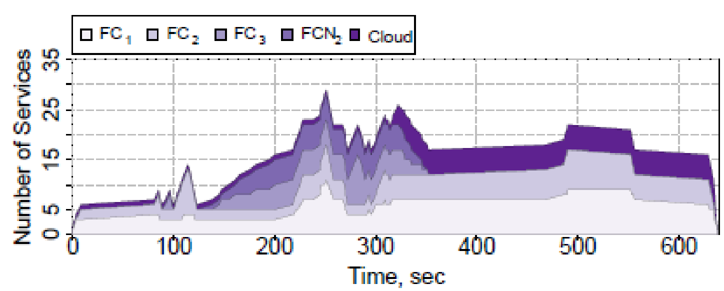


(b) Genetic algorithm

Pyramid arrival pattern: 5, 10, 15, 10, and 5 services each 2 minutes



(a) First-fit algorithm



(b) Genetic algorithm

Related Work

	Execution environment	Resource provisioning
[1] de Brito et al.	IoT testbed: Docker Swarm + OpenMTC M2M, VMs	Docker labels
[2] Tsai et al.	Distributed analytic platform: TensorFlow + Kubernetes, Raspberry Pis	By Kubernetes
[3] Yigitolu et al.	„Foggy“, Raspberry Pis	Orchestration server on every node, first-fit algorithm
[4] Brogi et al.	-	„FogTorch“, heuristic fail-first algorithm
[5] Vögler et al.	„DIANE“, simulated	Rule-based
[6] Xiao and Krunz	Simulated	Power consumption, QoE, „offload forwarding“
[7] Ni et al.	Simulated	Priced Timed Petri nets
[8] Saurez et al.	„Foglets“, simulated	Trigger-based: latency and resources

Conclusions

- Execution environment for IoT applications in a fog landscape
- FogFrame: placement, deployment, and execution of IoT applications
- Optimization problems for resource provisioning and service placement

Future work:

- Automated device discovery, fault tolerance mechanisms
- Availability, reliability of services and devices, cost in the optimization
- Heuristic and exact algorithms

Thank you for attention!

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Distributed
Systems Group

Olena Skarlat, MSc
Research Assistant

Vienna University of Technology
Institute of Information Systems
Argentinierstrasse 8/184-1, 1040 Vienna, Austria
T: +43 1 58801-18459 **F:** +43 1 58801-18491
E: o.skarlat@infosys.tuwien.ac.at
www.infosys.tuwien.ac.at

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Related Work

	Reference
[1] de Brito et al.	A Service Orchestration Architecture for Fog-enabled Infrastructures. M. S. de Brito, S. Hoque, T. Magedanz, R. Steinke, A. Willner, D. Nehls, O. Keilsa, F. Schreiner, FMEC 2017
[2] Tsai et al.	Distributed Analytics in Fog Computing Platforms using Tensorow and Kubernetes. P. H. Tsai, H. J. Hong, A. C. Cheng, C. H. Hsu, APNOMS 2017
[3] Yigitolu et al.	Foggy: A framework for continuous automated iot application deployment in fog computing. E. Yigitoglu, M. Mohamed, L. Liu, H. Ludwig, AIMS 2017
[4] Brogi et al.	QoS-aware Deployment of IoT Applications Through the Fog. A. Brogi, S. Forti, IEEE Internet Things J.
[5] Vögler et al.	Optimizing Elastic IoT Application Deployments. M. Vögler, J. Schleicher, C. Inzinger, S. Dustdar, Trans. Serv. Comput.

Related Work (ctnd)

	Reference
[6] Xiao and Krunz	QoE and power eciency tradeo for fog computing networks with fog node cooperation. Y. Xiao, M. Krunz, INFOCOM 2017
[7] Ni et al.	Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets. L. Ni, J. Zhang, C. Jiang, C. Yan, K. Yu, IEEE Internet Things J.
[8] Saurez et al.	Incremental Deployment and Migration of Geo-Distributed Situation Awareness Applications in the Fog. E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, DEBS 2016

Our papers

Resource provisioning for IoT services in the fog
O. Skarlat, S. Schulte, M. Borkowski, P. Leitner
SOCA 2016

Towards QoS-aware fog service placement
O. Skarlat, M. Nardelli, S. Schulte, S. Dustdar
ICFEC 2017

Optimized IoT service placement in the fog
O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, P. Leitner
SOCA Journal 2017

FogFrame: Service placement, deployment, and execution in the fog
O. Skarlat, K. Bachmann, S. Schulte
FGCS Journal 2018 (under review)