

Synthesizing Plausible Infrastructure Configurations for Evaluating Edge Computing Systems

Thomas Rausch
Clemens Lachner
Pantelis A. Frangoudis
Philipp Raith
Schahram Dustdar

@thrauat



Towards a Serverless Platform for Edge AI

Thomas Rausch
TU Wien

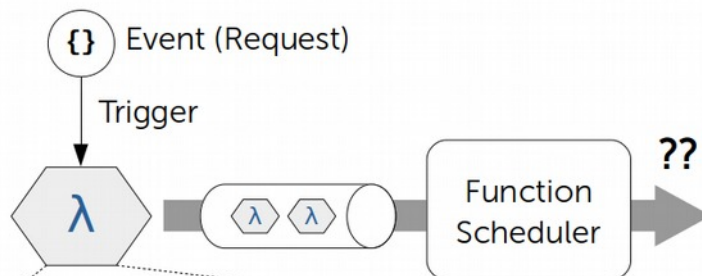
Waldemar Hummer
IBM Research AI

Vinod Muthusamy
IBM Research AI

Alexander Rashed
TU Wien

Schahram Dustdar

Deviceless Model

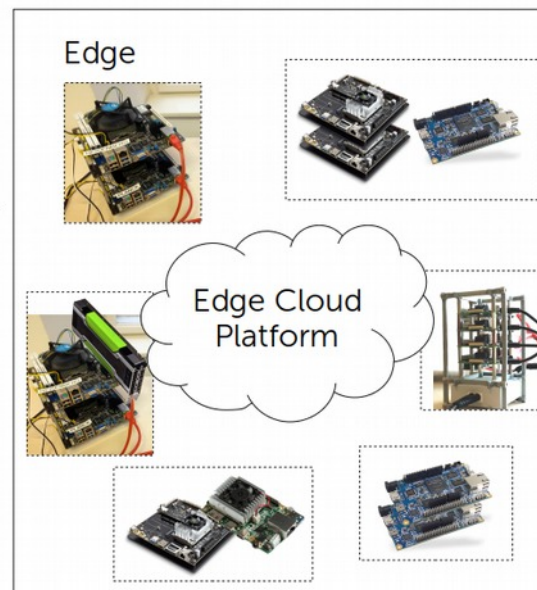


```

def handle(req):
    s3 = boto3.client('s3')
    with open(tmpfile, 'wb') as f:
        s3.download_fileobj('bucket', req['obj'], f)
        data = numpy.load(f)

    m = train_model(data, req['train_params'])
    s3.upload_fileobj(serialize(m), 'bucket', 'model')
    # ...
  
```

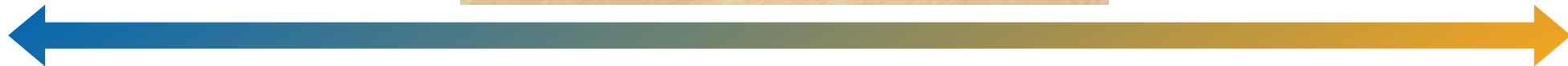
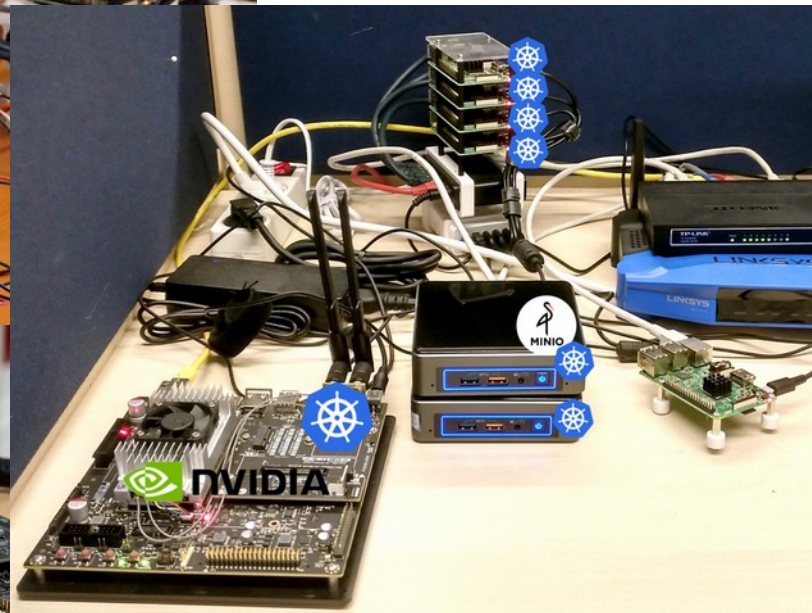
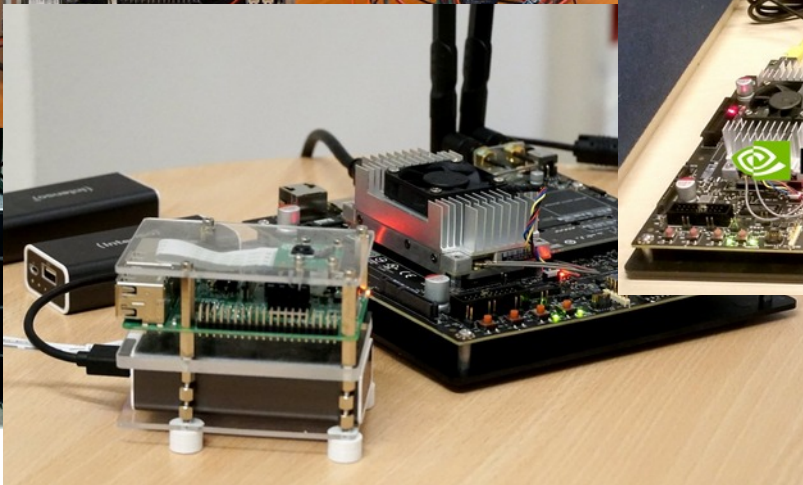
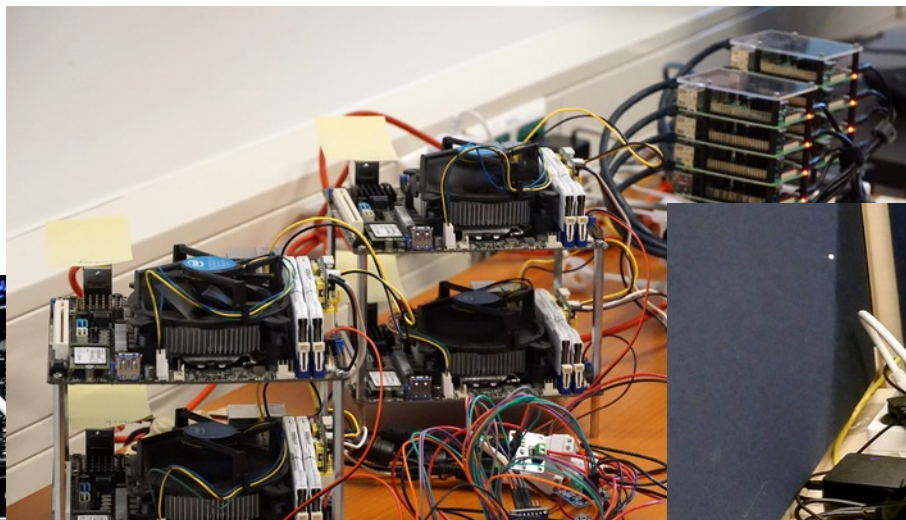
Edge Cloud



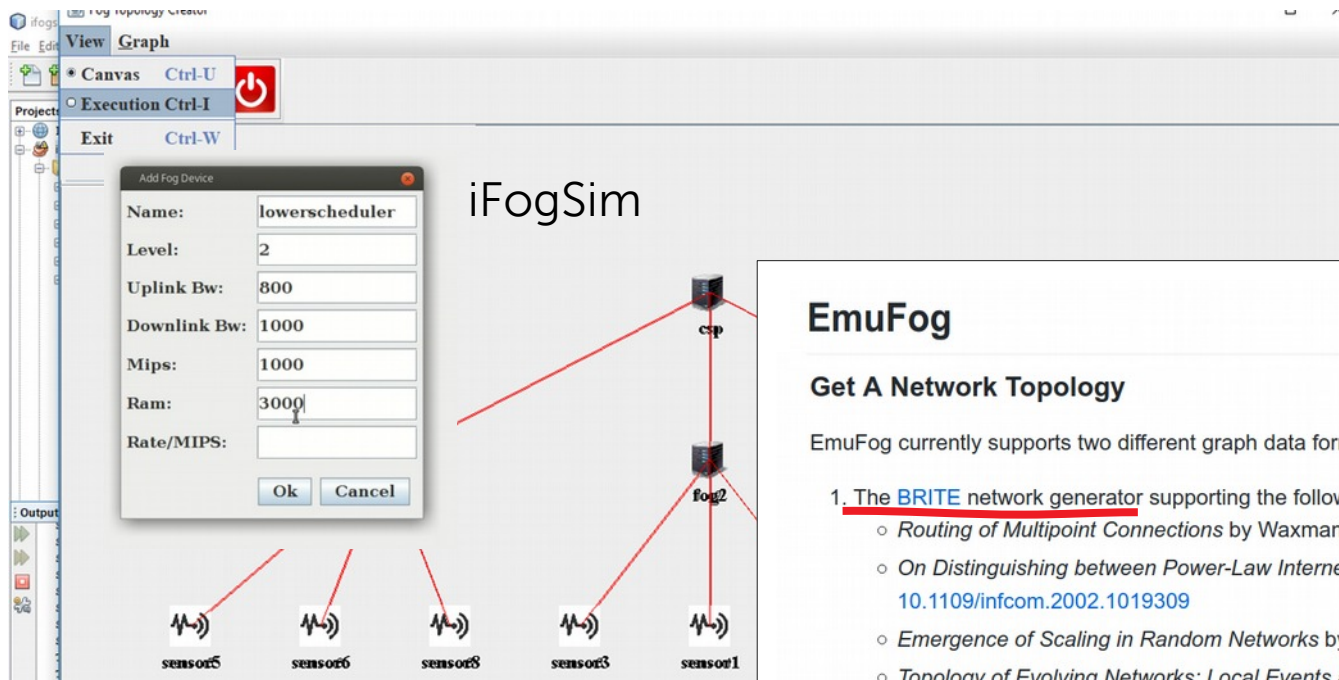
This paper proposes a serverless operating edge AI application to illustrate the challenges in edge cloud from AI lifecycle management. In a serverless model, we enable easy deployment of functions. We take a deviceless approach to sources transparently as cloud fine-grained control over server resources. We demonstrate the limitations of current schedulers, and present the

1 Introduction

Edge AI is one of the major drivers in recent advanced computing systems, and amidst the growing demand for low-latency and privacy-aware



Edge systems evaluation tools



EmuFog

Get A Network Topology

EmuFog currently supports two different graph data formats:

1. The BRITE network generator supporting the following models:
 - *Routing of Multipoint Connections* by Waxman, DOI: [10.1109/incom.2002.1019309](https://doi.org/10.1109/incom.2002.1019309)
 - *On Distinguishing between Power-Law Internet Topology Generators* by Bu and Towsley, DOI: [10.1109/incom.2002.1019309](https://doi.org/10.1109/incom.2002.1019309)
 - *Emergence of Scaling in Random Networks* by Barabási and Albert, DOI: [10.1126/science.286.5439.50](https://doi.org/10.1126/science.286.5439.50)
 - *Topology of Evolving Networks: Local Events and Universality* by Barabási and Albert, DOI: [10.1103/physrevlett.85.16](https://doi.org/10.1103/physrevlett.85.16)
2. The Macroscopic Internet Topology Data Kit from Caida including measured real world internet topologies

Gupta et al. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments

Mayer et al. EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures

What are representative infrastructure configurations for edge systems?

The Computing Landscape of the 21st Century

Mahadev Satyanarayanan
Carnegie Mellon University
satya@cs.cmu.edu

Wei Gao
University of Pittsburgh
WEIGAO@pitt.edu

Brandon Lucia
Carnegie Mellon University
blucia@andrew.cmu.edu

ABSTRACT

This paper shows how today's complex computing landscape can be understood in simple terms through a 4-tier model. Each tier represents a distinct and stable set of design constraints that dominate attention at that tier. There are typically many alternative implementations of hardware and software at each tier, but all of

2 A Tiered Model of Computing

Today's computing landscape is best understood by the *tiered model* shown in Figure 1. Each tier represents a distinct and stable set of design constraints that dominate attention at that tier. There are typically many alternative implementations of hardware and software at each tier, but all of them are subject to the same set of

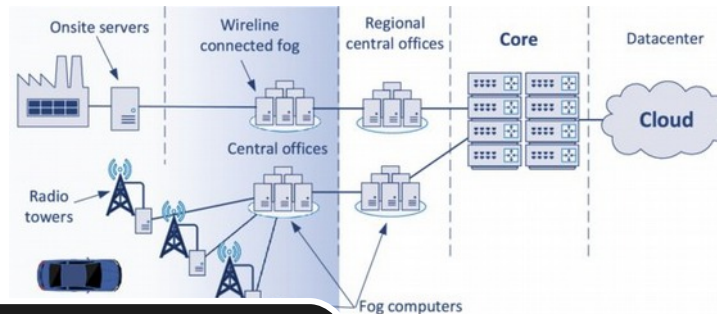
Urban Sensing



Array of Things, I



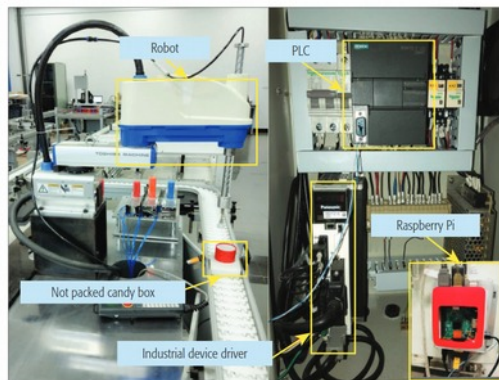
Multi-Access Edge Computing



Best distribution for fog and cloud servers

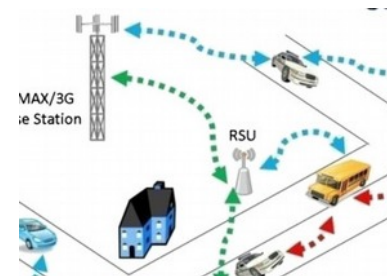
Details in the paper!

Industrial



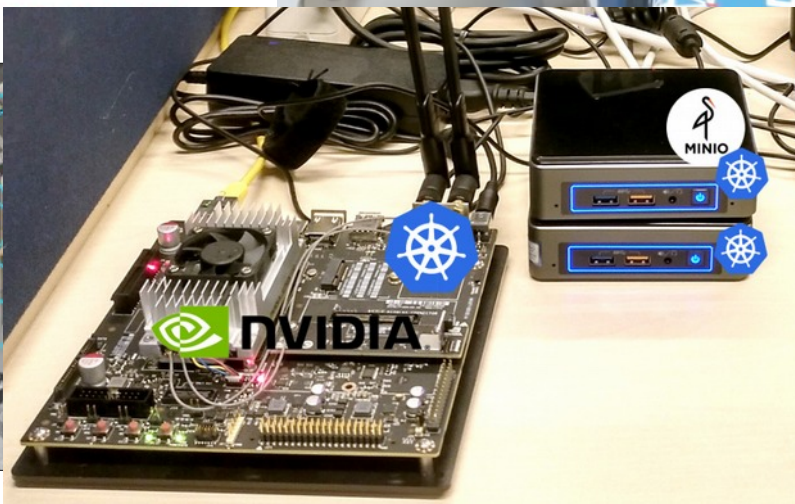
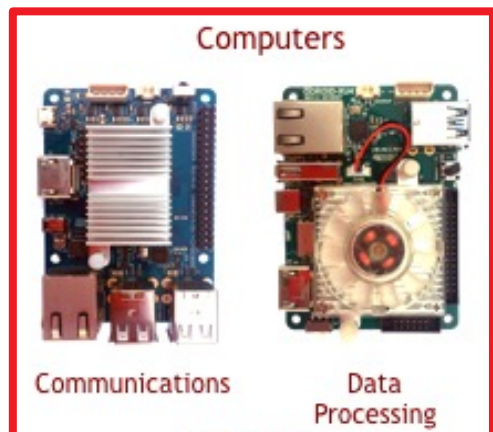
Chen et al. Edge Computing in IoT-Based Manufacturing

Networking



Eiza et al. Investigation of routing reliability of vehicular ad hoc networks

City-Scale Edge Computing Fabric



Edge Topology Synthesizer



This block shows the top navigation bar and repository header of a GitHub page. The dark navigation bar at the top contains the GitHub logo, links for 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing', a search bar, and 'Sign in' and 'Sign up' buttons. Below this, the repository path 'edgerun / ether' is displayed with a computer icon. To the right are buttons for 'Watch' (1), 'Star' (1), and 'Fork' (0). A secondary bar contains tabs for 'Code' (selected), 'Issues' (2), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Security' (0), and 'Insights'.

Edge Topology Synthesizer [under development]

This block displays the README content for the 'ether' repository. At the top, a summary bar shows '12 commits', '2 branches', '0 packages', '0 releases', and '1 contributor'. Below this is a header for 'README.md'. The main content starts with the title 'ether: Edge Topology Synthesizer' in a large, bold font. A paragraph follows, stating: 'Ether is a Python tool to generate plausible edge infrastructure configurations. It arose from the need to evaluate edge computing systems in different infrastructure scenarios where no appropriate testbeds are available.' This is followed by a section titled 'Use cases' with a horizontal line underneath. The text 'Some of the uses cases for ether include' is followed by a bulleted list: 'Evaluating resource allocation strategies', 'Creating topologies for network simulations', and 'Infrastructure capacity planing'.

12 commits 2 branches 0 packages 0 releases 1 contributor

ether: Edge Topology Synthesizer

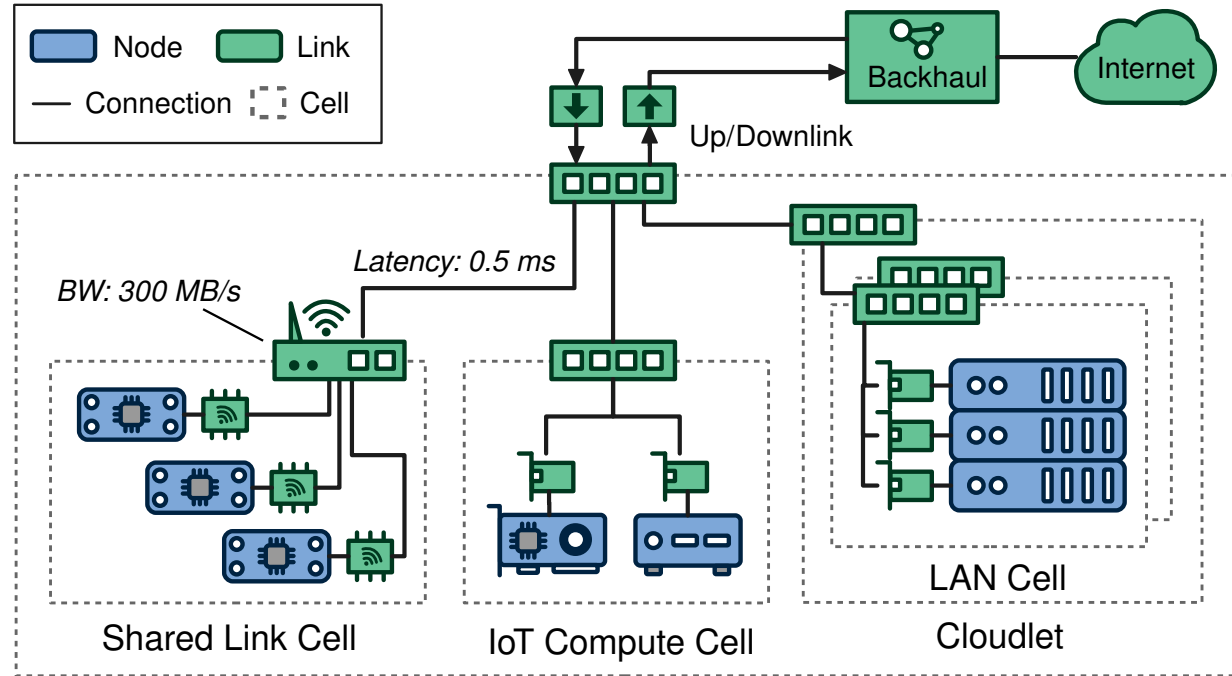
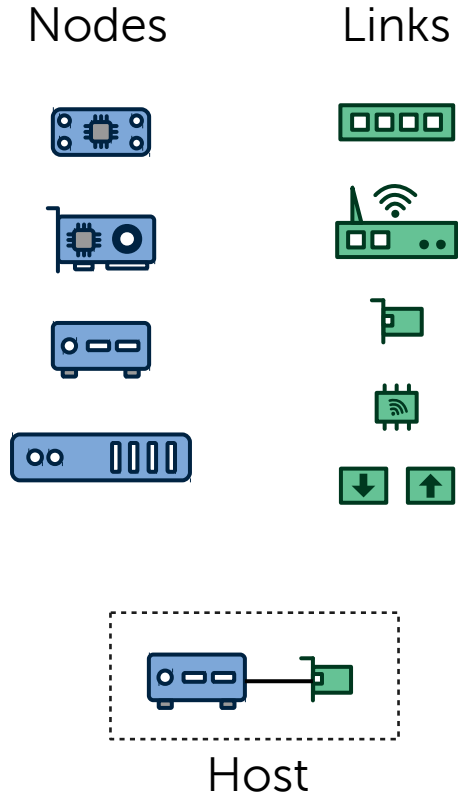
Ether is a Python tool to generate plausible edge infrastructure configurations. It arose from the need to evaluate edge computing systems in different infrastructure scenarios where no appropriate testbeds are available.

Use cases

Some of the uses cases for *ether* include

- Evaluating resource allocation strategies
- Creating topologies for network simulations
- Infrastructure capacity planing

Conceptual model



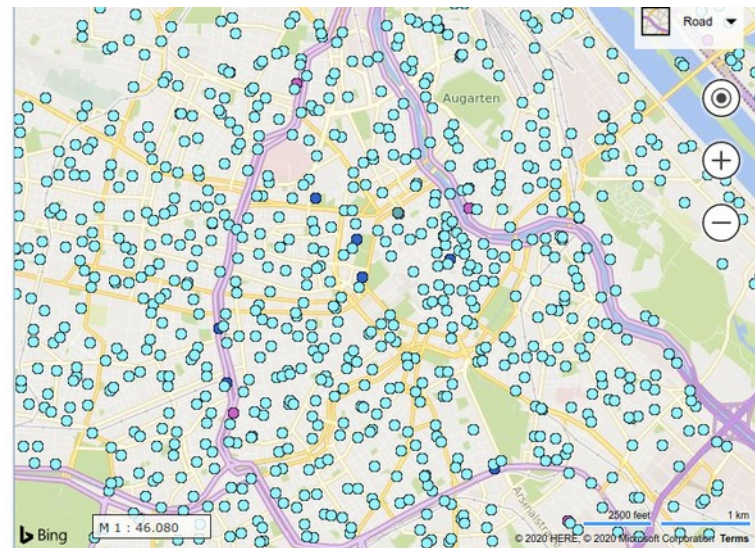
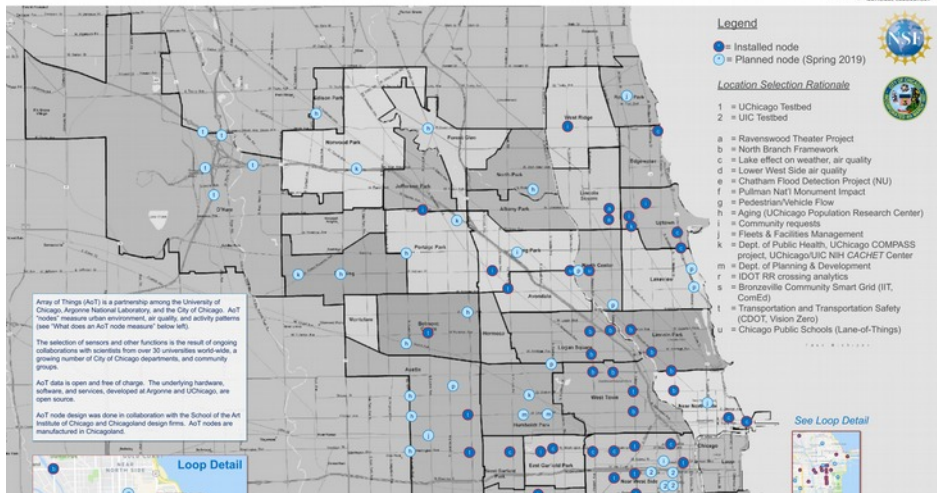
Industrial IoT Scenario

Parameterized cell synthesis

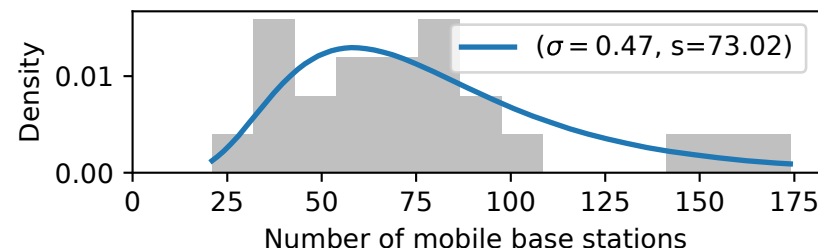
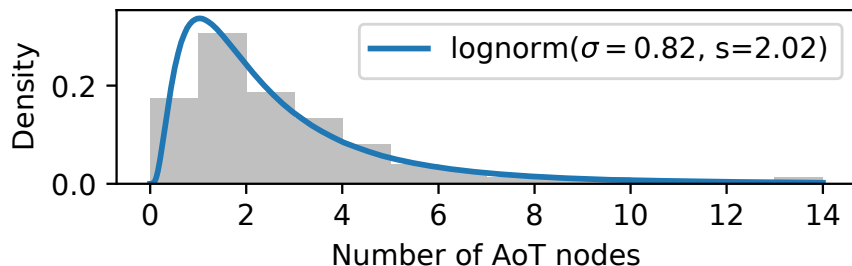
- Degree distributions? Probably not.
- Cell (cluster) size
- Heterogeneity:
How *different* are the devices in the cell?
- Density:
How many nodes are in this cell compared to others?
- ? What others would be useful

GeoCell Density

Array of Things – a Fitness Tracker for Cities



<https://www.senderkataster.at/karte>



Programmatic creation of topologies

```
topology = Topology()    (= networkx graph)

aot_node = IoTComputeBox(nodes=[nodes.rpi3, nodes.rpi3])
neighborhood = lambda size: SharedLinkCell(
    nodes=[
        [aot_node] * size,
        IoTComputeBox([nodes.nuc] + ([nodes.tx2] * size * 2))
    ],
    shared_bandwidth=500,
    backhaul=MobileConnection('internet_chix'))
city = GeoCell(
    5, nodes=[neighborhood], density=lognorm((0.82, 2.02)))
cloudlet = Cloudlet(
    5, 2, backhaul=FiberToExchange('internet_chix'))

topology.add(city)
topology.add(cloudlet)
```

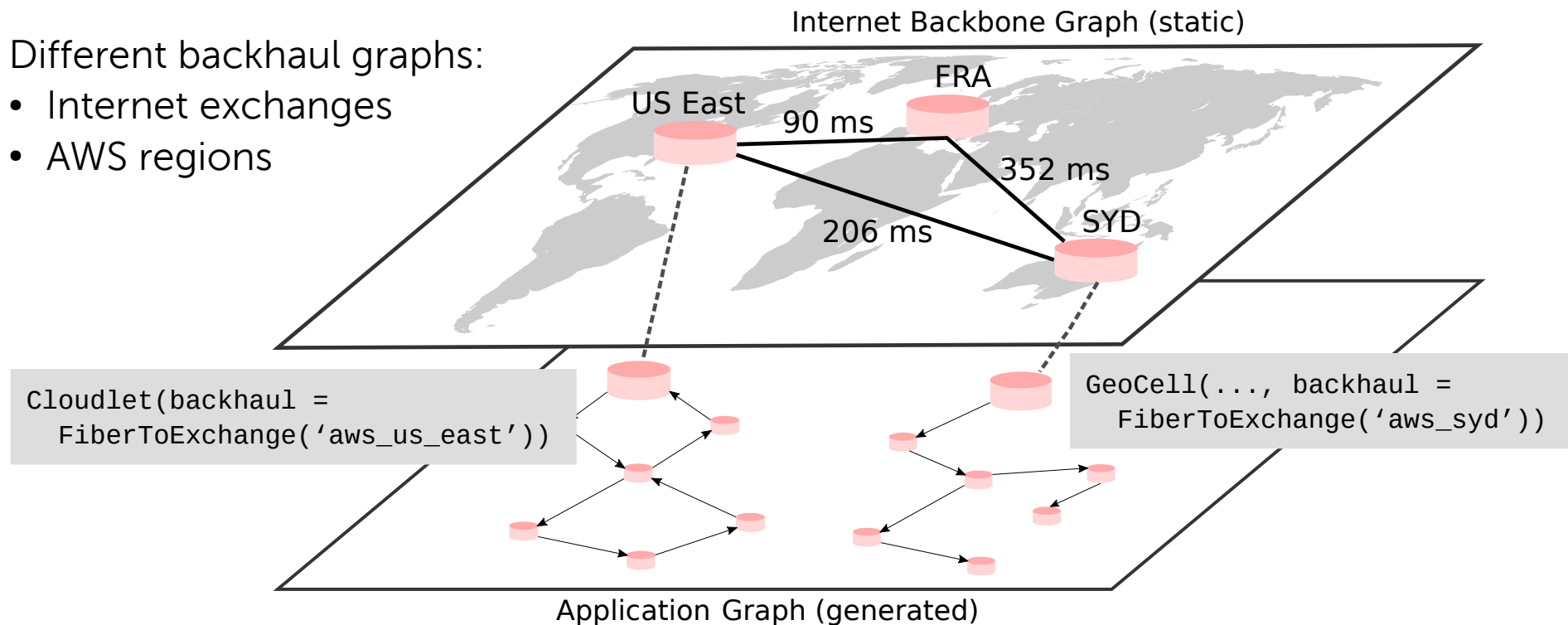


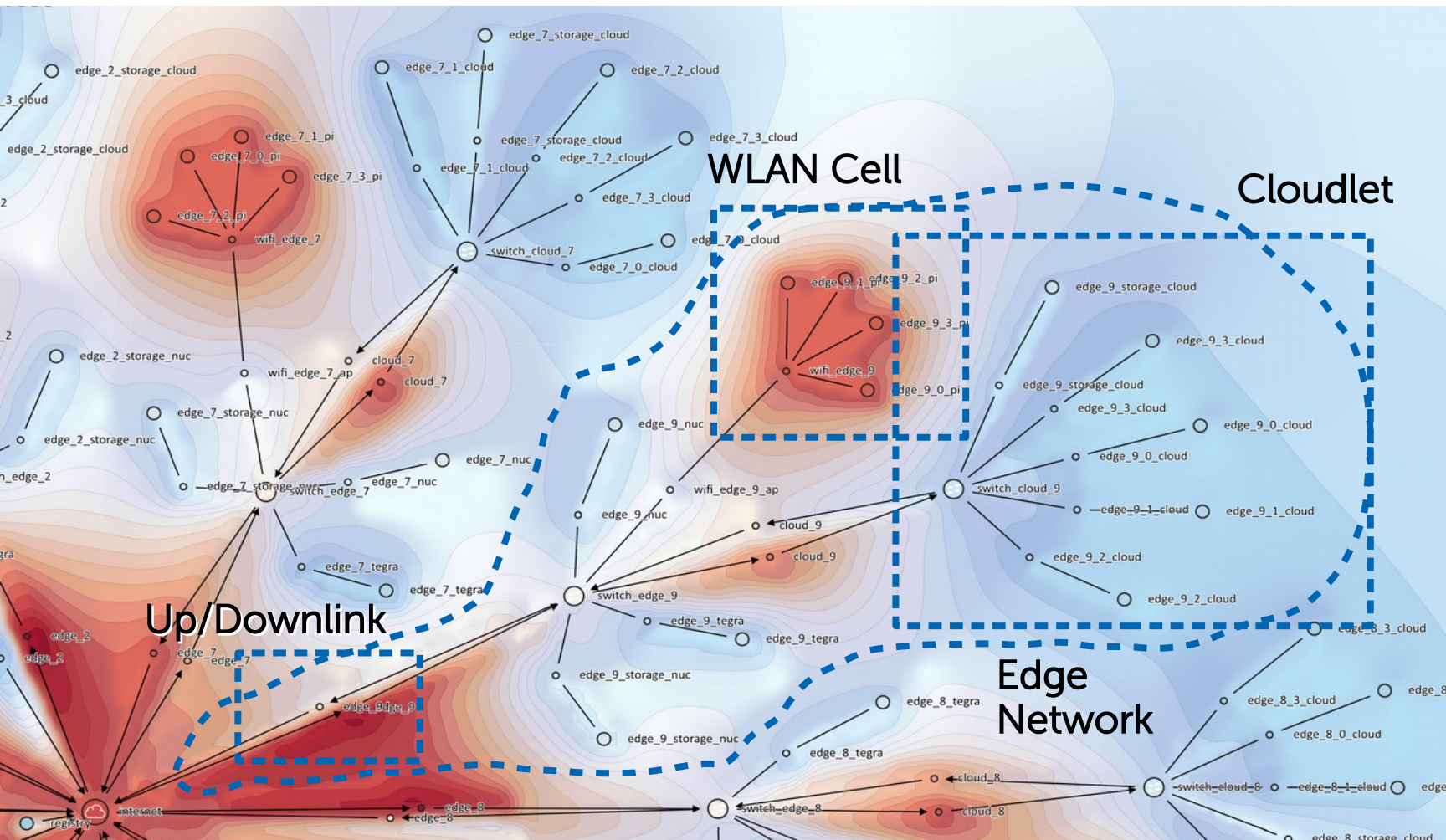
GraphML
GML
JSON
YAML
...

Interconnecting topologies

Different backhaul graphs:

- Internet exchanges
- AWS regions





Visualization by Reinhold Preiner using TAM: <https://github.com/rpreiner/tam>

Simulation data from faas-sim: <https://github.com/edgerun/faas-sim>



ether: Edge Topology Synthesizer

Ether is a Python tool to generate plausible edge infrastructure configurations. It arose from the need to evaluate edge systems in different infrastructure scenarios where no appropriate testbeds are available.

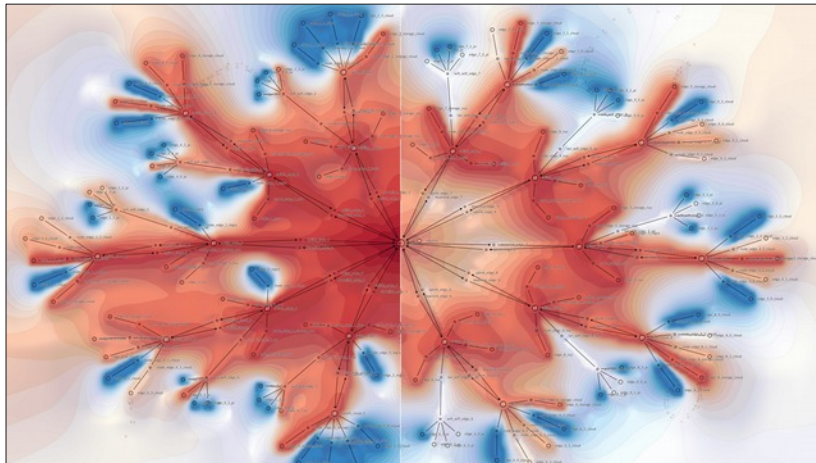
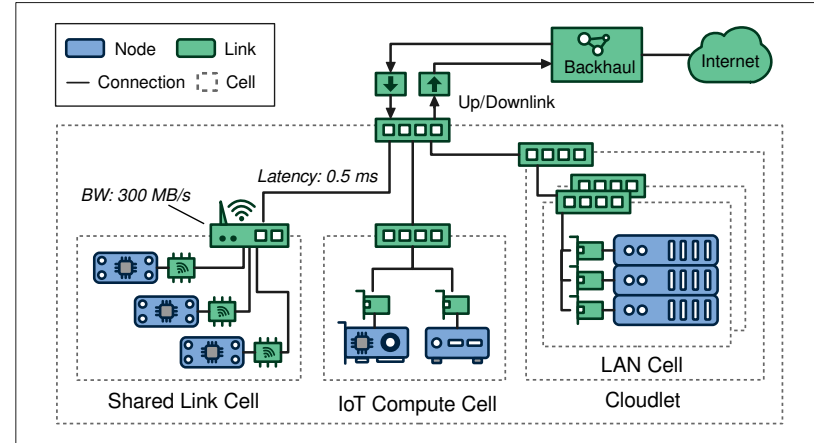
Use cases

Some of the uses cases for *ether* include

- Evaluating resource allocation strategies
- Creating topologies for network simulations
- Infrastructure capacity planning

github.com/edgerun/ether







Informatics

Dipl.-Ing. (MSc), BSc
Thomas Rausch
 Research Assistant

TU Wien
Information Systems Engineering
 Argentinierstrasse 8-194-02, Vienna, Austria
 T: +43 1 58801-184838
 E: trausch@dsg.tuwien.ac.at
<https://dsg.tuwien.ac.at/staff/trausch>



DISTRIBUTED
SYSTEMS GROUP