Fog Computing

Bermbach | Part 1: From Cloud to Fog Computing

Agenda

Technische Universität Berlin

Lectures

- From Cloud to Fog Computing
- Fog Data Management
- Application Engineering
- Midterm Test
- Q&A
- Final Test

Prototyping

- Kick-off
- Group work
- Final Presentations

Reading assignments

- Kick-off
- Group work
- Final Presentations

Mobile Cloud Computing

EINSTEIN CENTER Digital Future

From Cloud to Fog Computing

Fog Computing has been called *the natural evolution of cloud computing*, i.e.,

What is cloud computing?

comes before

What is fog computing?

Recap

CLOUD COMPUTING

NIST Definition: Cloud Computing

"Cloud computing is a model for enabling
ubiquitous, convenient, on-demand network access to a
shared pool of configurable computing resources
(e.g., networks, servers, storage, applications, and services) that can be
rapidly provisioned and released with
minimal management effort or service provider interaction."

NIST Cloud Computing Characteristics

On-Demand Self-Service

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad Network Access:

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource Pooling:

The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Cloud Computing Characteristics

Rapid Elasticity:

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time

Measured Service:

Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Main Cloud Vendors

aws

Microsoft Azure

IBM Cloud

SAP Cloud Platform

Example

AMAZON WEB SERVICES

The AWS Global Infrastructure

- 20 regions: e.g., US West (California, Oregon), EU (Ireland); 4 more planned
- 61 availability zones for scalability and fault-tolerance purposes

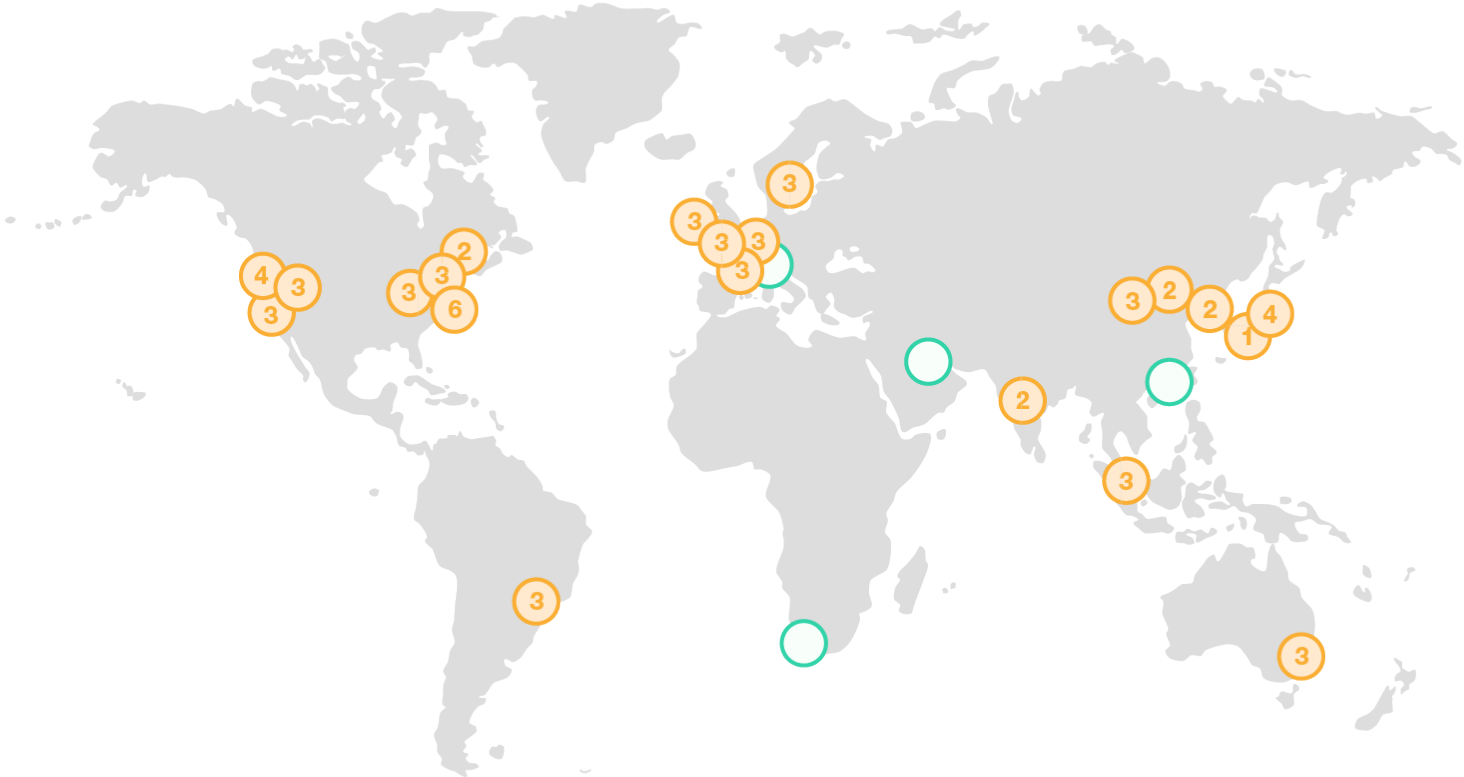


Image: https://aws.amazon.com/about-aws/global-infrastructure

Basic AWS Developer Model

Technische Universität Berlin

- Create AWS account; Sign up as developer
- Choose services
- Opt. sign up for services
- Opt. agree to service licenses (read carefully!)
- Understand pricing model; choose billing model
- Get identifiers and tools as needed
- Start coding

Mobile Cloud Computing

EINSTEIN CENTER Digital Future

Amazon Web Services

ELASTIC COMPUTE CLOUD (EC2)

EC2 Core Concepts

Amazon Machine Image (AMI): an encrypted file stored in Amazon storage, containing all the information necessary to boot instances of a customer's software

- An AMI is like a bootable root disk, which can be pre-defined or user-built
- Public AMIs: Pre-configured, template AMIs
- Private AMIs: User-built AMI containing private applications, libraries, data and associated configuration settings

Instance: the running system based on an AMI

- All instances based on the same AMI begin executing identically. An instance can be launched in a few minutes
- Any information on them is lost when the instances are terminated or if they fail
- Data can be made persistent by use of the Amazon storage services

EC2 Purchasing Options

On-Demand

- Pay by the hour with no long-term commitments
- For spikey, short-term, and unpredictable workloads
- t3.2xlarge costs currently $0.3328 per hour

Spot

- Pay for unused capacity which price fluctuates by the hour
- For workloads with flexible start and end times
- t3.2xlarge costs currently $0.1002 per hour

Reserved

- Pay a fixed price for resources reserved just for you
- For steady, long-term, and predictable workloads
- t3.2xlarge costs currently $0.1950 per hour when payed upfront for a year

EC2 Instance Types

General purpose instances
High CPU instances
High memory instances
GPU instances

Burst instances vs. regular instances
EBS-optimized vs. ephemeral storage

Amazon Web Services

LAMBDA

AWS Lambda

- Amazon's serverless computing offer
- Allows stateless code execution in response to events
- No servers to manage
- Scales continuously

Pay per Request

- Compute time is bought in 100ms increments
- Compute power
- No idle servers that need to be payed for

Simple Resource Model

- Only choose how much memory should be allocated for a function
- AWS Lambda proportionally allocates CPU power, network bandwidth, and disk I/O

Lambda Features

Bring your own Code

- Currently supports Java, Node.js, C#, and Python
- No limitation on third party libraries

Extends other AWS Services

- Add custom logic to resources such as AWS S3
- Good way to build a back-end service for applications that are triggered on-demand

Monitoring and Logging

- Build in metrics for number of requests, latencies, errors, …
- Logging via CloudWatch

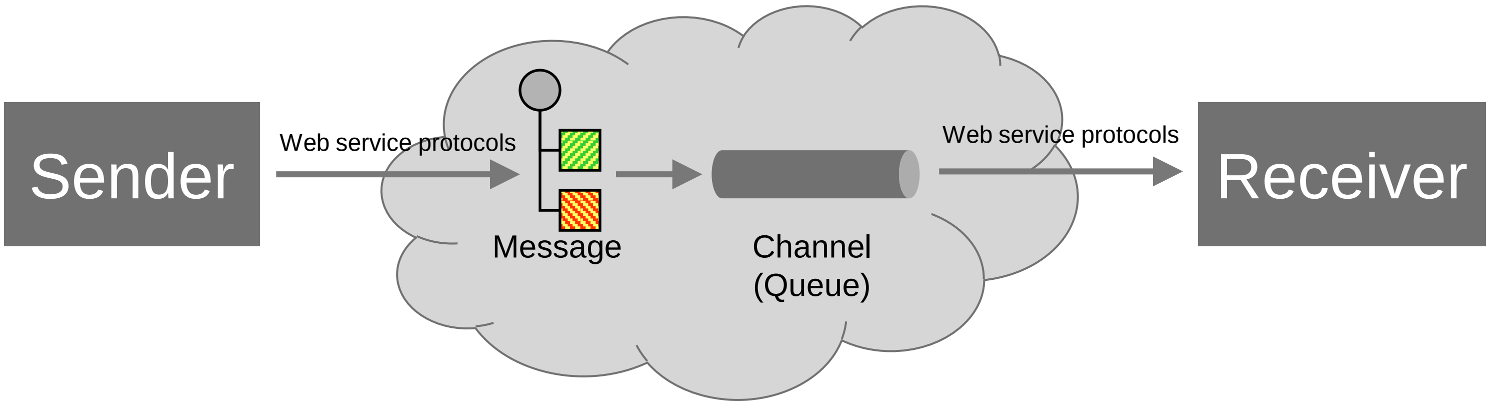
Fault Tolerance and Downtimes

- Compute capacity is maintained across multiple availability zones
- Predictable and reliable operational performance
- No maintenance windows or schedule downtimes

Amazon Web Services

SIMPLE QUEUE SERVICE (SQS)

AWS SQS Functionality



- Developers can create an unlimited number of Amazon SQS queues with an unlimited number of messages
- Messages can be sent and read simultaneously
- A server can check queues at any time for messages
- When a message is received, it becomes “locked” while being processed. This keeps other computers from processing the message simultaneously
- The message body can contain up to 256 KB of text in any format
- Messages can be retained in queues for up to 14 days

Standard Queues vs FIFO Queues

Standard Queue	FIFO Queue
Unlimited Throughput Nearly unlimited number of transactions per second	**High Throughput** Up to 300 transactions per second
At-Least-Once Delivery Messages are delivered at least once, but sometimes more often	**Exactly-Once Delivery** Messages are delivered exactly once
Best-Effort Ordering Messages might be re-ordered	**FIFO Ordering** Message order is preserved

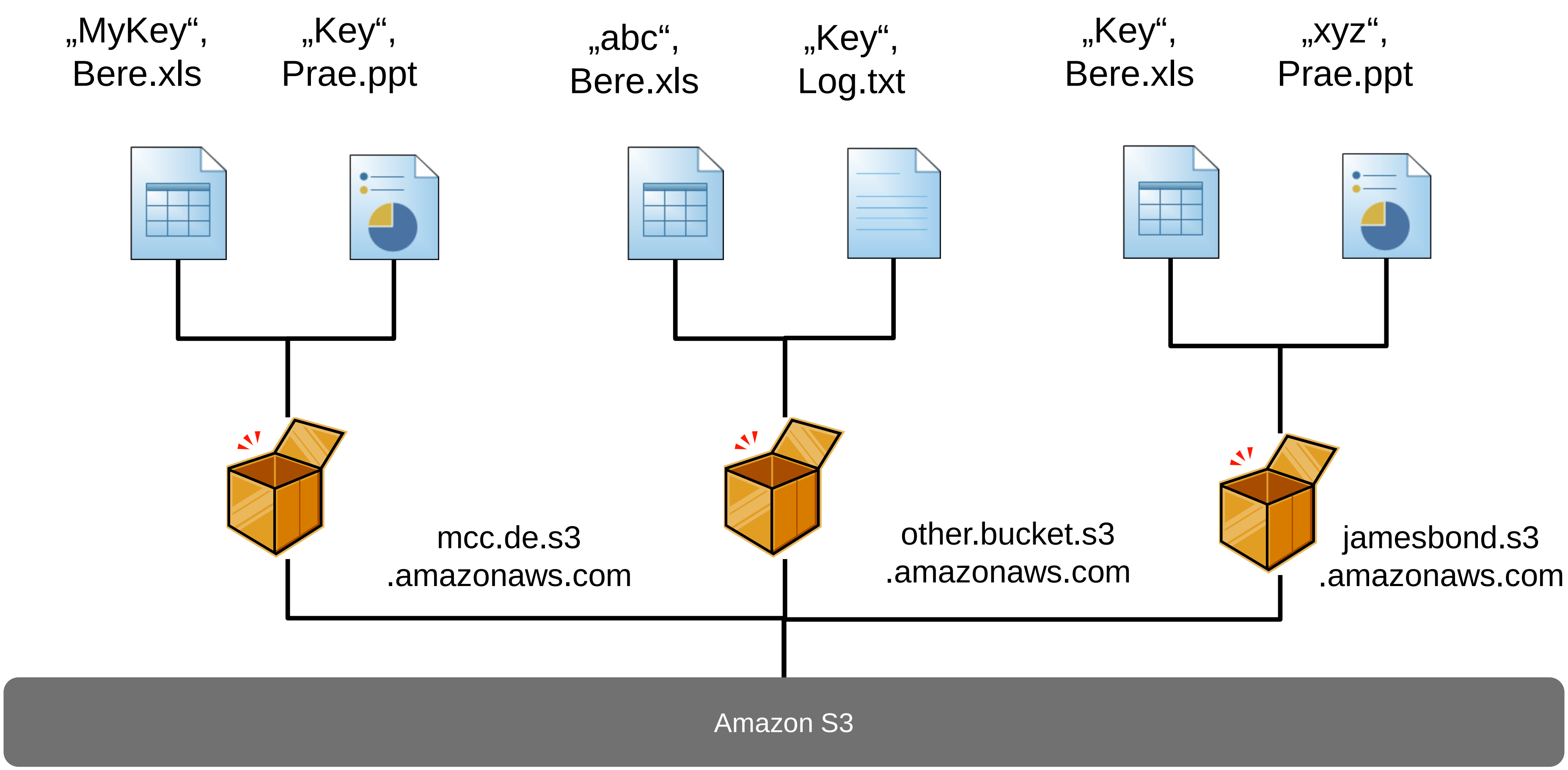
Amazon Web Services

SIMPLE STORAGE SERVICE (S3)

AWS Simple Storage Service (S3)

- Simple key-value store
- Basic web services interface to store and retrieve any amount of data (as objects), at any time, from anywhere on the web
- Based on the same “highly scalable, reliable, secure, fast, inexpensive” infrastructure that Amazon uses to run its global network of web sites
- Write, read, and delete objects containing from 1 byte to 5 terabytes of data each
  – Each object is stored in a bucket
  – Buckets partition the namespace of objects stored in Amazon S3 at the top level, names must be unique across all of Amazon S3
- Uses standards-based REST and SOAP interfaces
- Amazon S3 *Service Level Agreement* addressing reliability guarantees

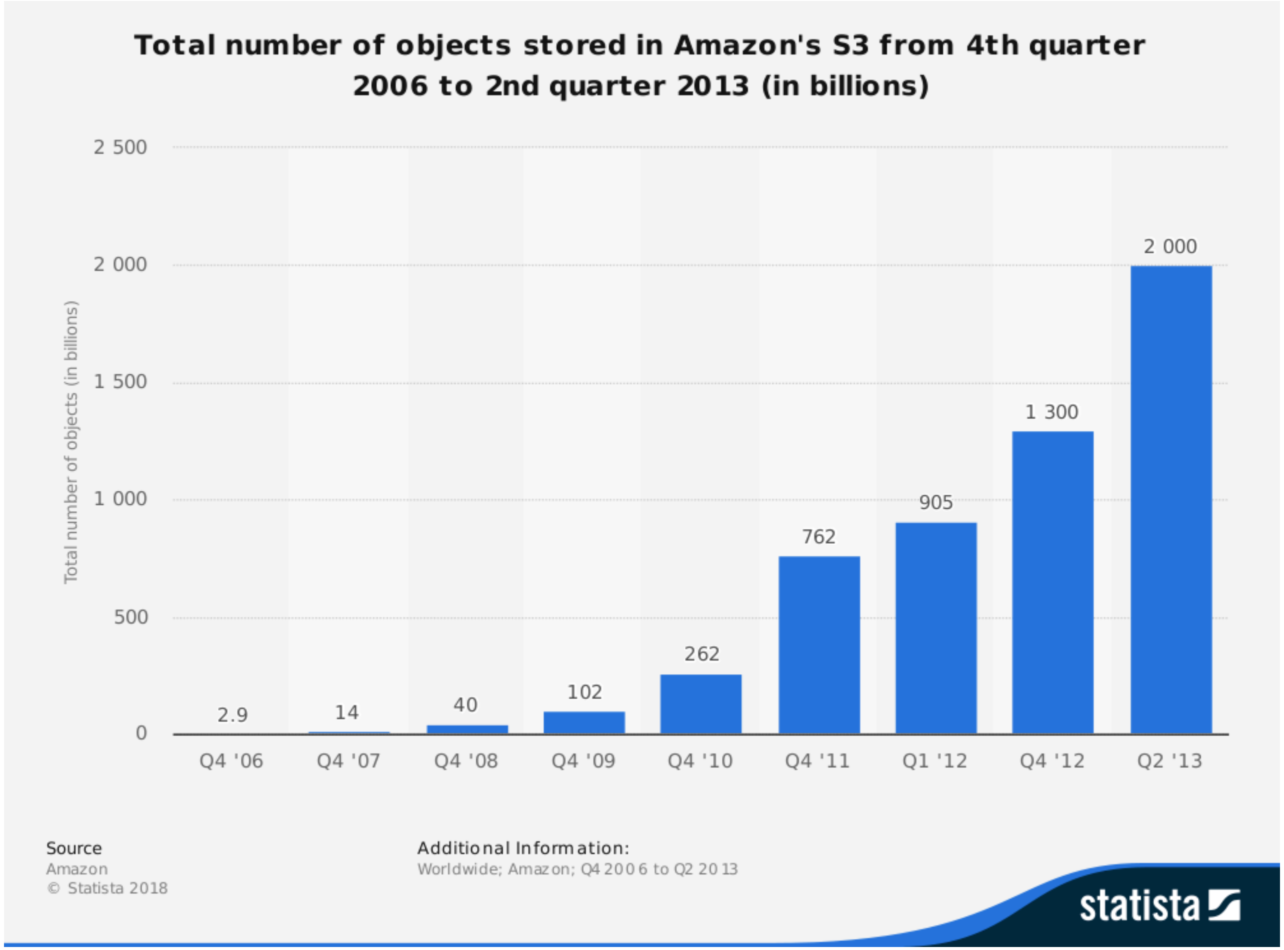
S3 Namespace



S3 Bucket Policy & Data Protection

- Bucket policies define access rights for Amazon S3 resources
- A bucket owner can write a bucket policy to allow/deny bucket-level permissions
  - Deny permission on any objects in the bucket
  - Grant permission on objects in the bucket only if the bucket owner is the object owner. For objects owned by other accounts the object owner must manage permissions using ACLs (Access Control Lists)
- Data can be protected using data encryption
  - The **client-side encryption** uploads the encrypted data to Amazon S3. In this case, the client manages the encryption process, the encryption keys, and related tools
  - The **server-side encryption** feature encrypts the object data before saving it on disks in its data centers and decrypts it when downloading the objects. Server-side encryption frees the client from the tasks of managing encryption, encryption keys, and related tools

The Number of Objects Stored in S3 Grows Exponentially (slightly outdated though)



Amazon Web Services

DYNAMO DB

AWS Dynamo DB

- NoSQL database that supports key-value and document data models
- Designed to run high-performance, internet-scale applications
    - Scales to >10 trillion requests per day
    - Peaks to >20 million requests per second
    - Over petabytes of storage
- Data can be replicate across regions
    - Globally distributed applications can access data locally
    - Gives single-digit millisecond read and write performance
- Per default, reads are *eventually consistent*

Dynamo DB Table

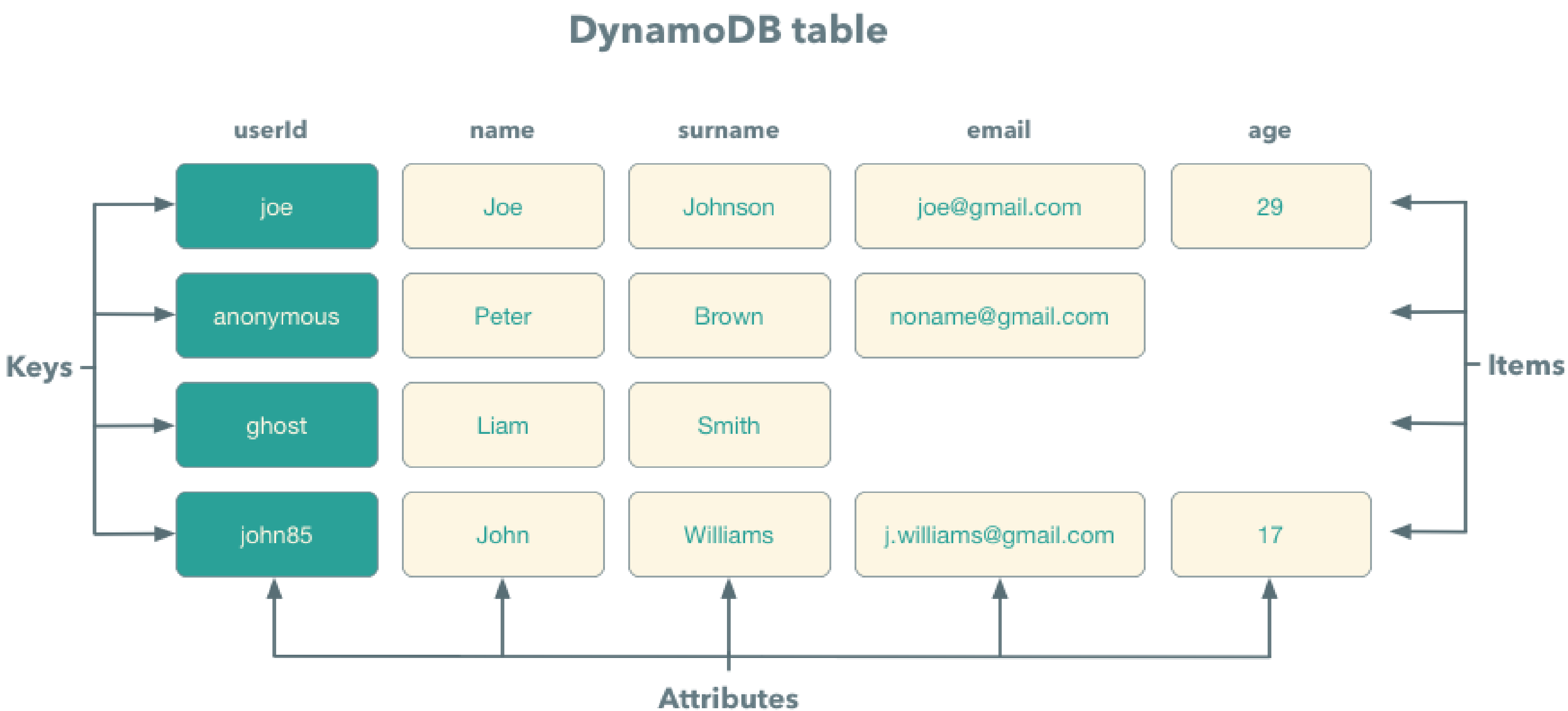
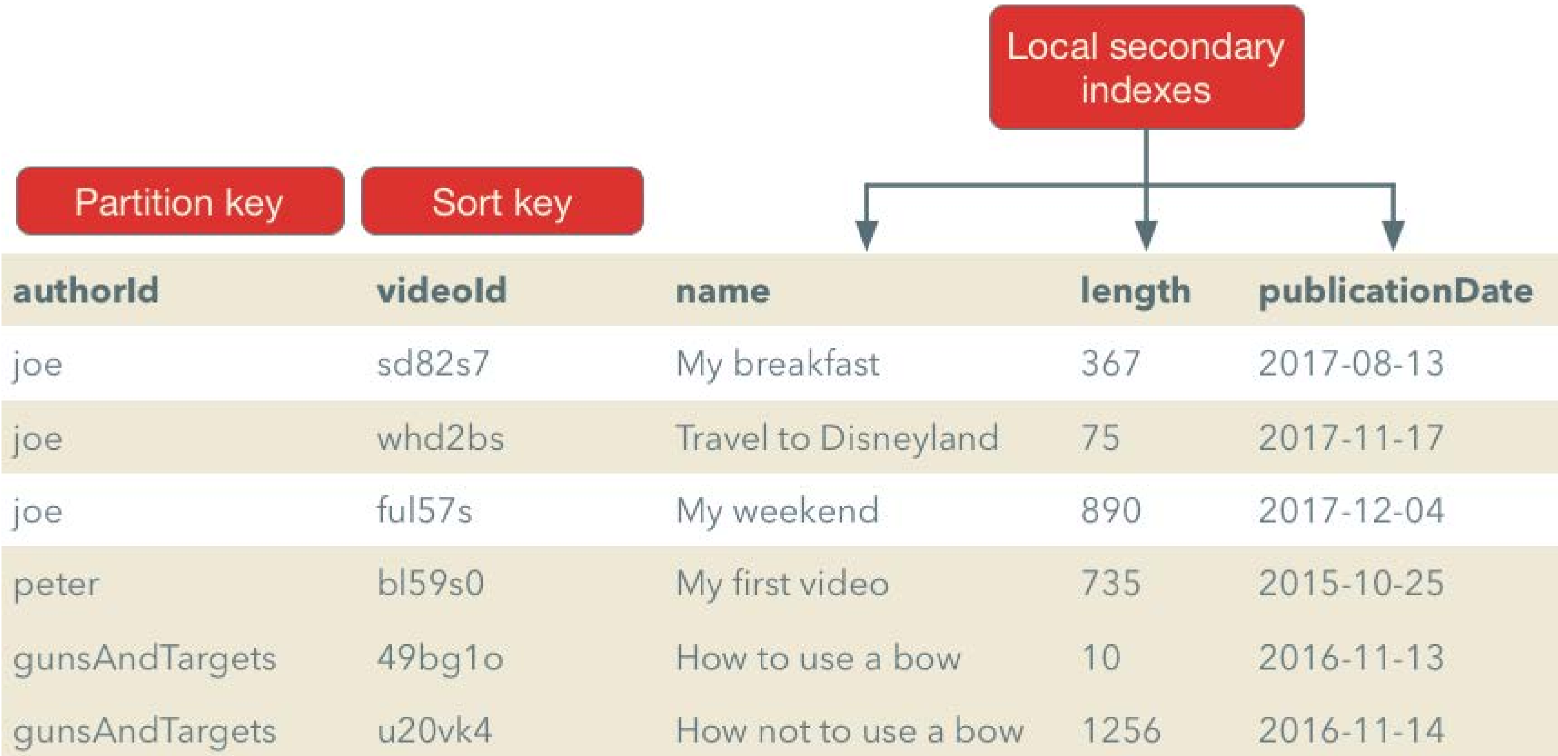


Image: https://brewing.codes/2017/11/13/dynamo-data-modeling/

Uses simple key, allows only retrieval by key

Partition and Sort Key, Local Secondary Index



authorId	videoId	name	length	publicationDate
joe	sd82s7	My breakfast	367	2017-08-13
joe	whd2bs	Travel to Disneyland	75	2017-11-17
joe	ful57s	My weekend	890	2017-12-04
peter	bl59s0	My first video	735	2015-10-25
gunsAndTargets	49bg1o	How to use a bow	10	2016-11-13
gunsAndTargets	u20vk4	How not to use a bow	1256	2016-11-14

Image: https://brewing.codes/2017/11/13/dynamo-data-modeling/

Complex key: select items based on partition key, then run range queries on sort key

Use local sec. index the same way as sort key

Global Secondary Index

Original table

Partition key	Sort key				
authorId	**videoId**	**name**	**length**	**publicationDate**	**season**
joe	sd82s7	My breakfast	367	2017-08-13	Food
joe	whd2bs	Travel to Disneyland	75	2017-11-17	My travel
joe	ful57s	My weekend	890	2017-12-04	My travel
peter	bl59s0	My first video	735	2015-10-25	null
gunsAndTargets	49bg1o	How to use a bow	10	2016-11-13	Bows and arrows
gunsAndTargets	u20vk4	How not to use a bow	1256	2016-11-14	Bows and arrows

↓

Global secondary index

Partition key	Sort key				
season	**videoId**	**name**	**length**	**publicationDate**	**authorId**
Food	sd82s7	My breakfast	367	2017-08-13	joe
My travel	whd2bs	Travel to Disneyland	75	2017-11-17	joe
My travel	ful57s	My weekend	890	2017-12-04	joe
Bows and arrows	49bg1o	How to use a bow	10	2016-11-13	gunsAndTargets
Bows and arrows	u20vk4	How not to use a bow	1256	2016-11-14	gunsAndTargets

Image: https://brewing.codes/2017/11/13/dynamo-data-modeling/

Creates a copy of the original table that is updates asynchronously

Building Cloud Applications
Example: Grep The Web (2008)

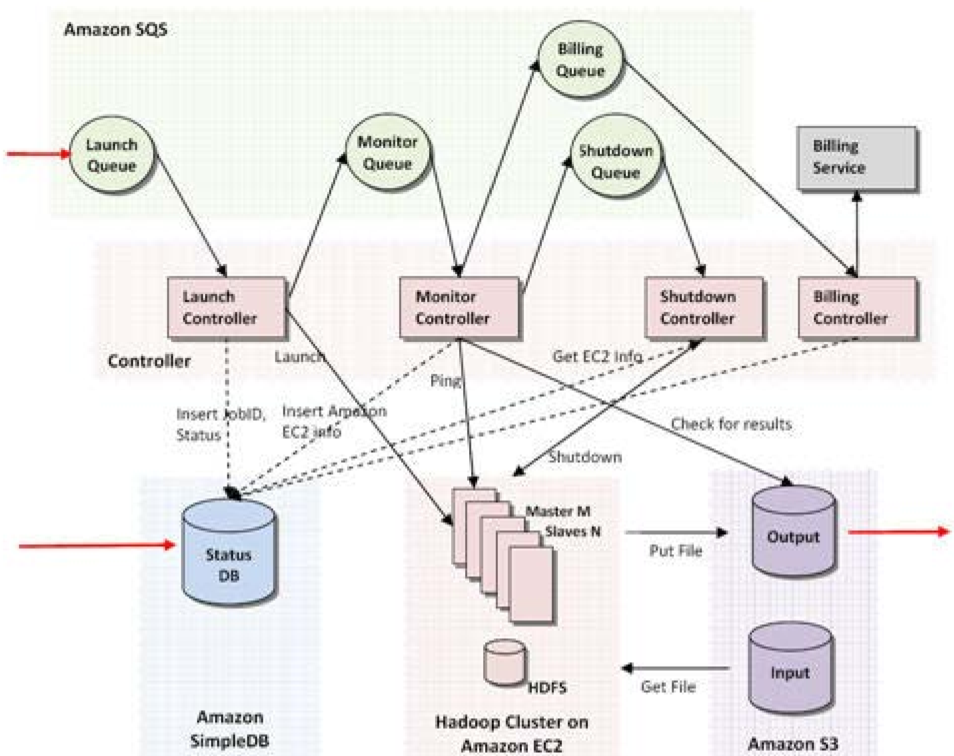
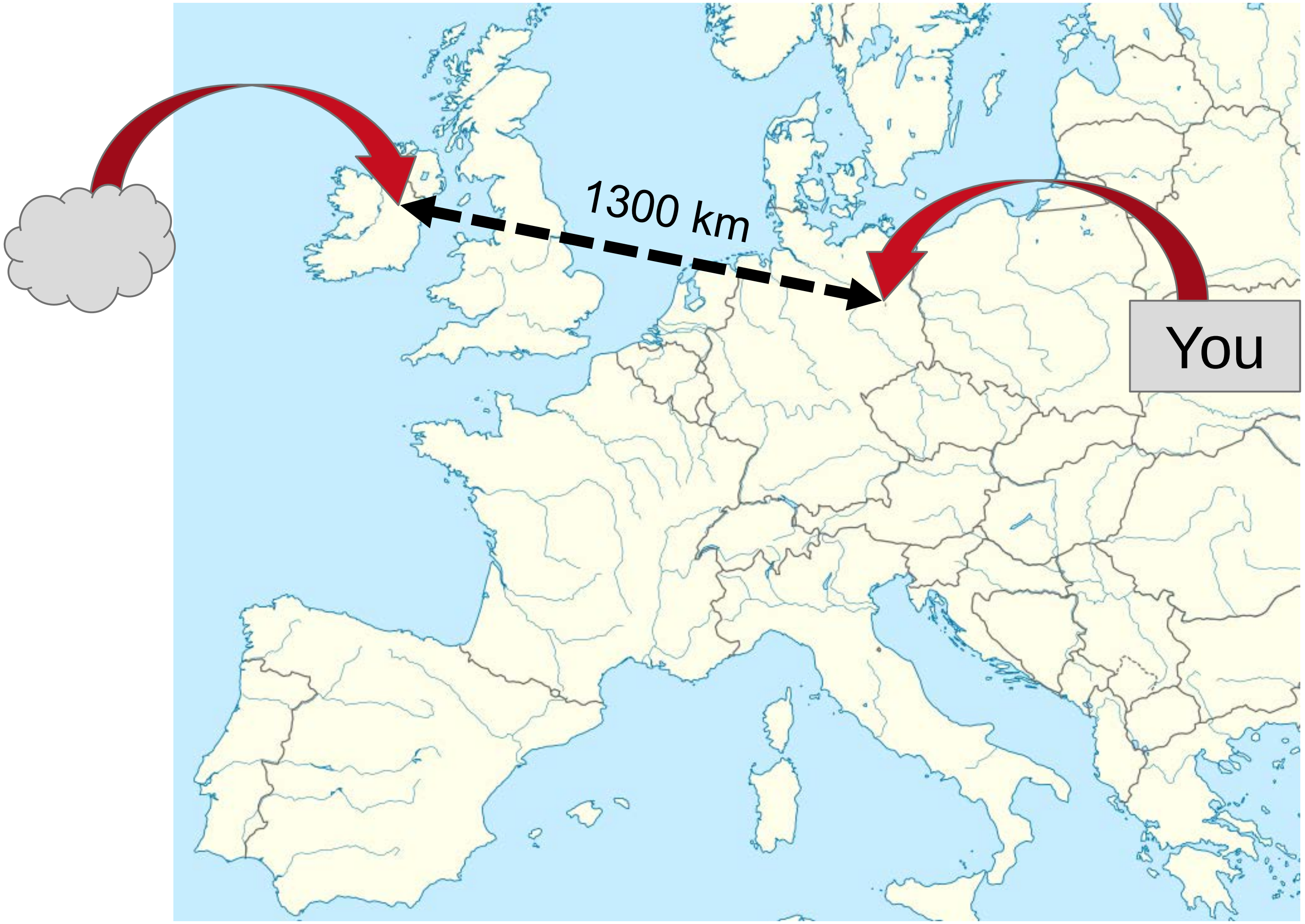


Image: https://aws.amazon.com/de/blogs/aws/white-paper-on/

FROM CLOUD TO FOG

Cloud Drawback



ping = 37ms

=> extra 0.5(+)m reaction time in city traffic

Why is Cloud Computing not Enough?

- Requires continuous connectivity, which is not feasible in many real world situations
- Latency might be to high for specific use cases
- Bandwidth limitations
- Regulations / privacy requirements

➢ Fog computing can solve some of these issues

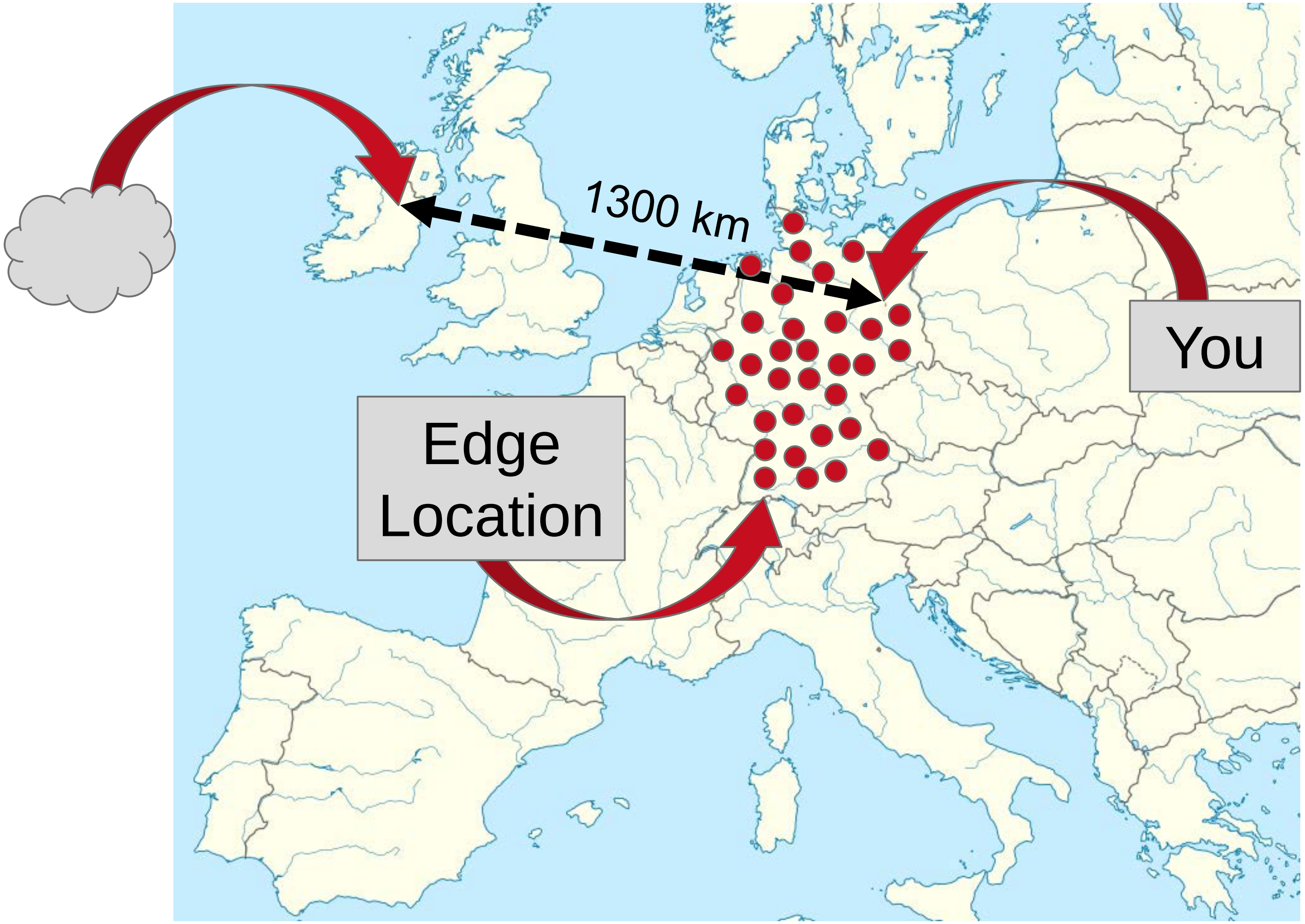
5G

IoT

eHealth

Autonomous driving

Fog computing to the rescue



What is the Edge?

- Refers to the outskirts of an administrative domain
- Applications are strongly associated with the Edge location
  - For end users, the Edge might be a mobile phone, car, or boat
  - For telecom companies, the Edge might be a point close to the end user but controlled by the provider
  - For large enterprises, the Edge might be a retail store or factory as that's where their application runs

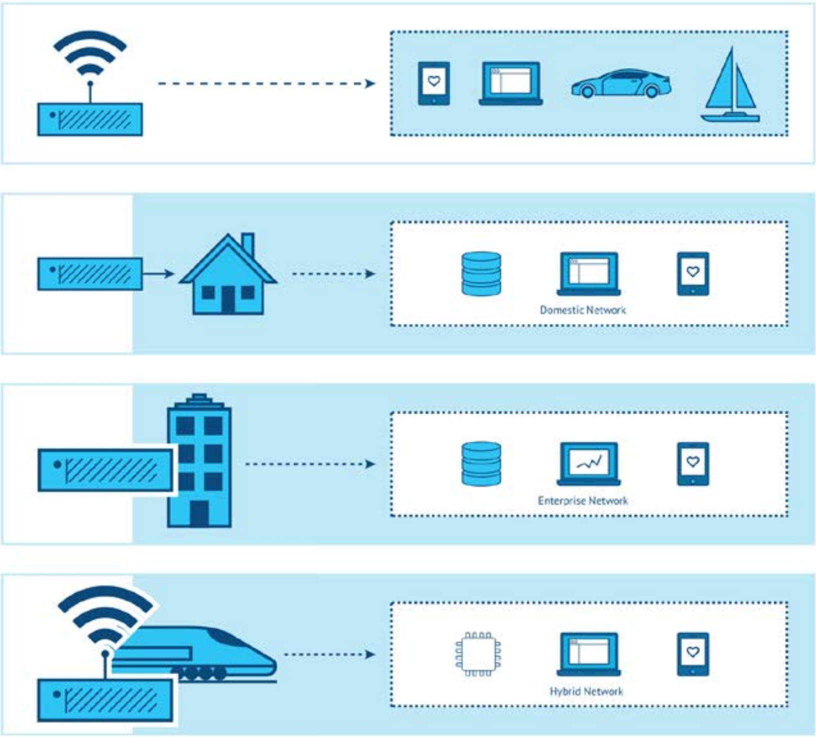


Image: https://www.openstack.org/edge-computing/cloud-edge-computing-beyond-the-data-center

Edge Computing is Not New

Long history from thick clients interacting with mainframe servers to content delivery networks (CDN).

Edge Computing in 2005:
Static data from the cloud is copied to the edge to enable faster access to it for end users (e.g., media).

Edge Computing in 2019:
Data is created at the edge (and in the cloud) and needs to be synced everywhere. Applications run at the edge.

FOG COMPUTING

What is Fog Computing?

- Fog computing combines cloud resources with edge devices and potential intermediary nodes in the network in between
- Fog Computing provides the ability to analyze data near the Edge for improved efficiency (regarding delay or bandwidth), or to operate while disconnected from a larger network (autonomy)
- At the same time, cloud services can be used for tasks that require more resources or elasticity

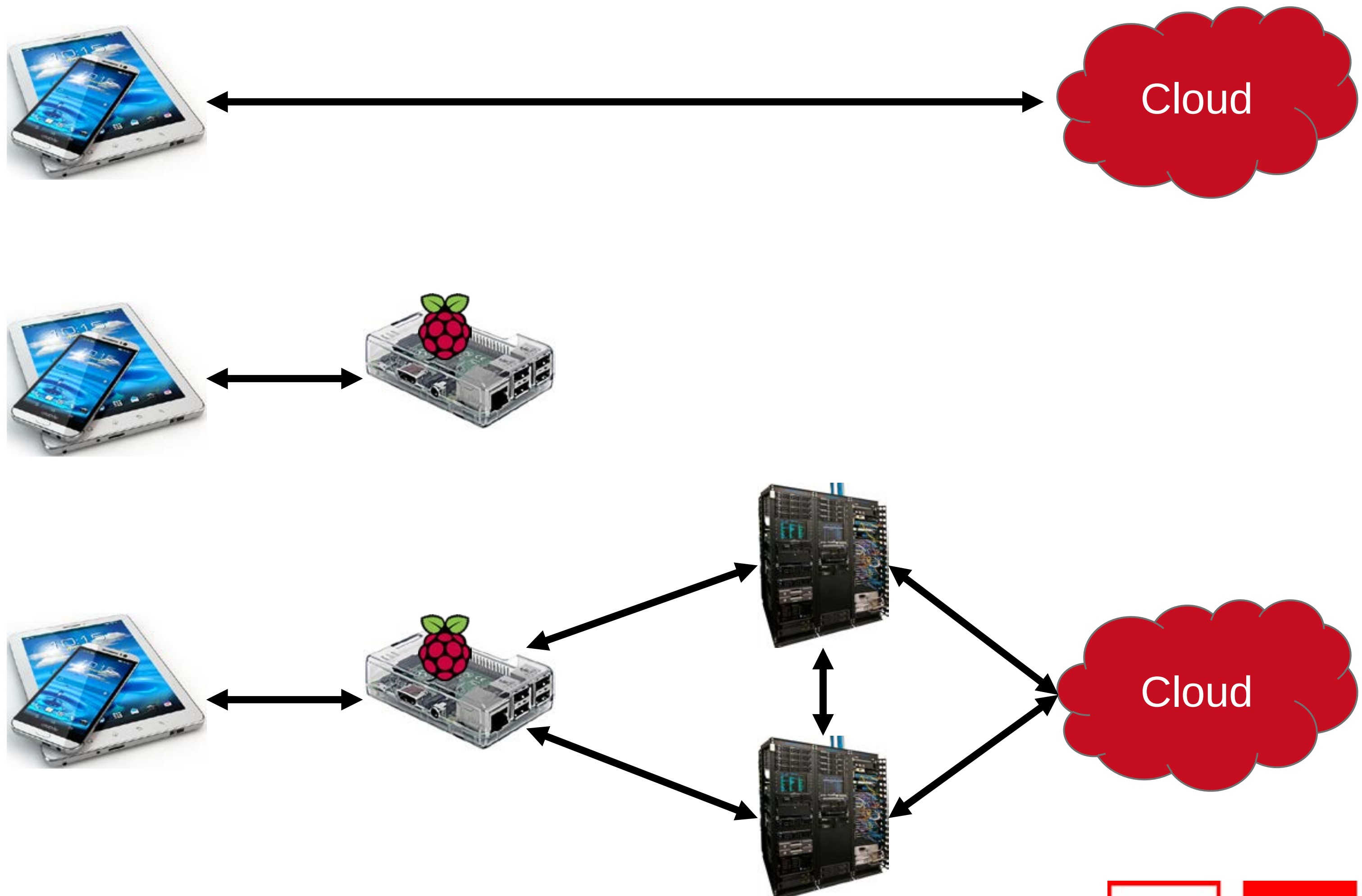
➢ Fog computing is an extension of the Cloud model as applications can reside in multiple layers of a network’s topology, including a backend Cloud

Cloud, Edge, and Fog Computing

Cloud

Edge

Fog



Fog Computing Characteristics

Technische Universität Berlin

- Runs required computations near the end-user and data to avoid latency, network, and other migration costs (including bandwidth)
- Uses lower latency storage at or near the Edge
- Uses low latency communication at or near the Edge rather than requiring all communications to be routed and synchronized through the backbone network
- Implements elements of management at or near the Edge rather than being primarily controlled through the Cloud
- Uses the Cloud for strategic tasks that require a large data context or huge amount of storage/compute power
- Multi-tenancy on a massive scale is required for some use cases

Mobile Cloud Computing

EINSTEIN CENTER Digital Future

Fog Computing is Geo-Distributed

- The physical location is significant as an application that needs to run close to its users needs to be in the "right part" of the Fog
- Entire pool of sites is dynamic as the physical separation comes with unreliable connections in between sites
- Sites are remote and potentially unmanned (requires administration via the network)
- "Fog nodes" deployed at a site can have various sizes and scales, e.g., from single device to complete data center
- Sites may be resource-constrained; adding capabilities (at the Edge) might not be possible due to space or power restrictions

Fog Computing

TYPICAL USE CASES

Use Cases

IoT
- Smart Home
- Industry 4.0

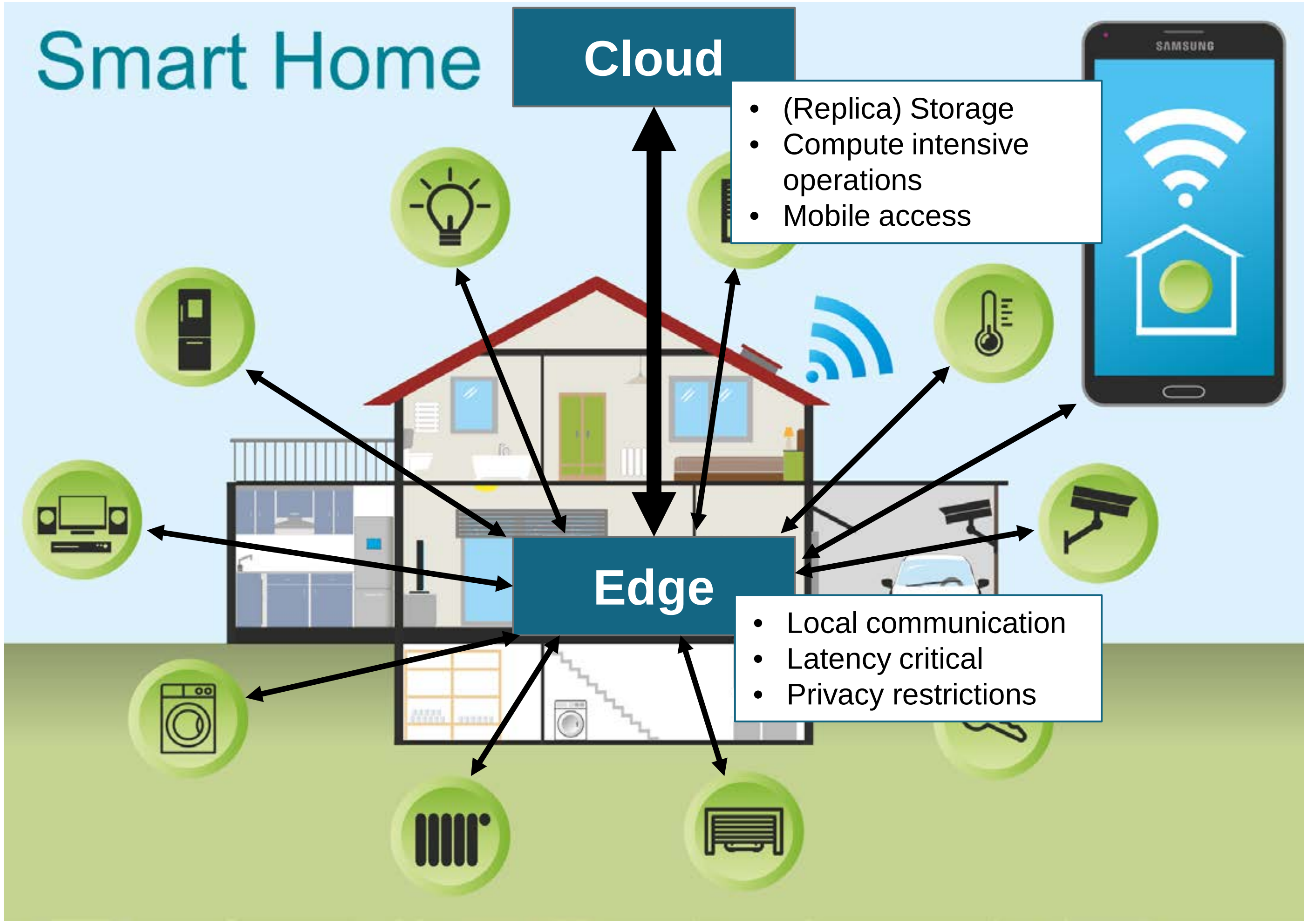
eHealth

Connected Vehicles

5G

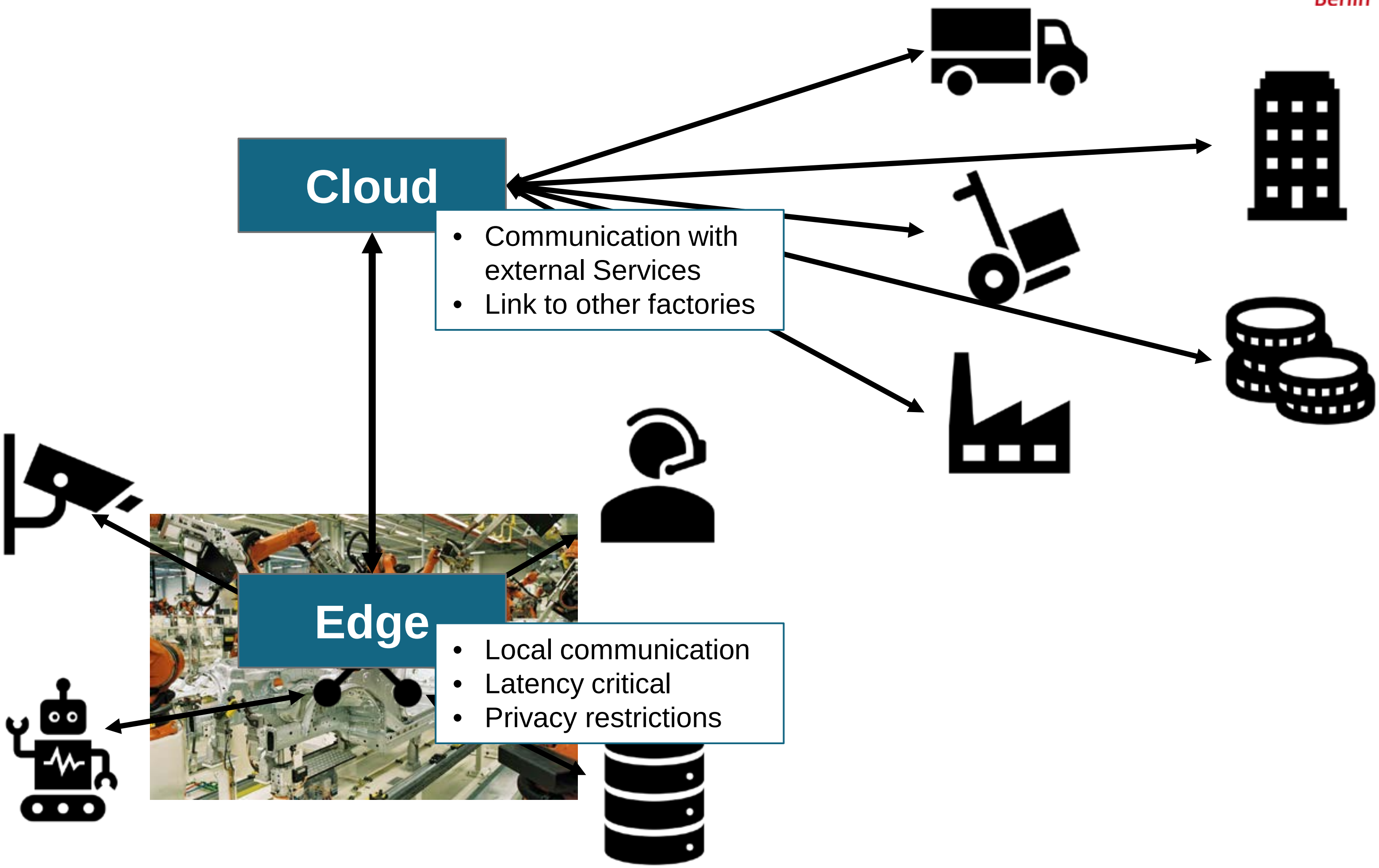
eHealth

Smart Home

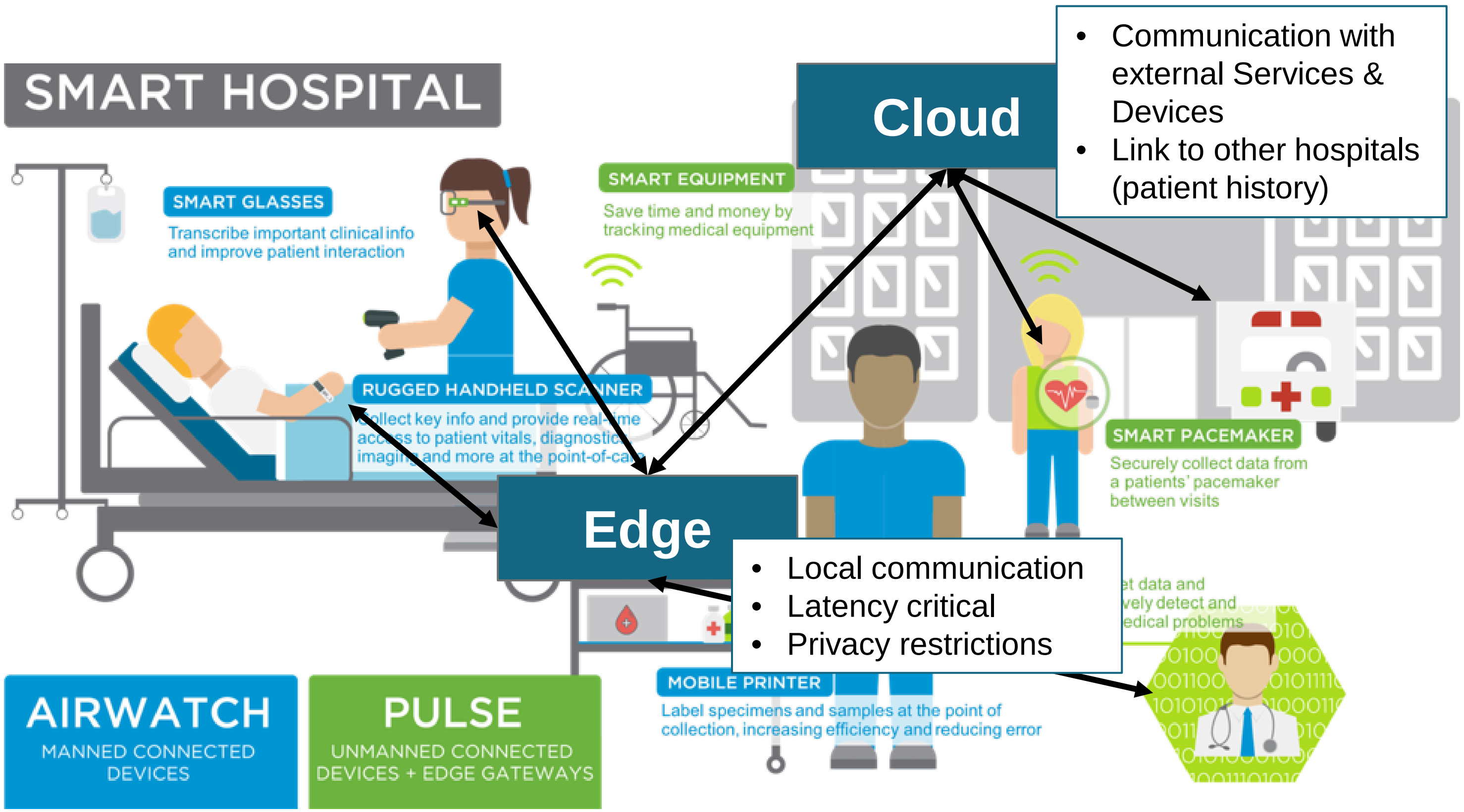


http://homeenergy.org/show/article/nav/issues/id/2197/magazine/156

Industry 4.0

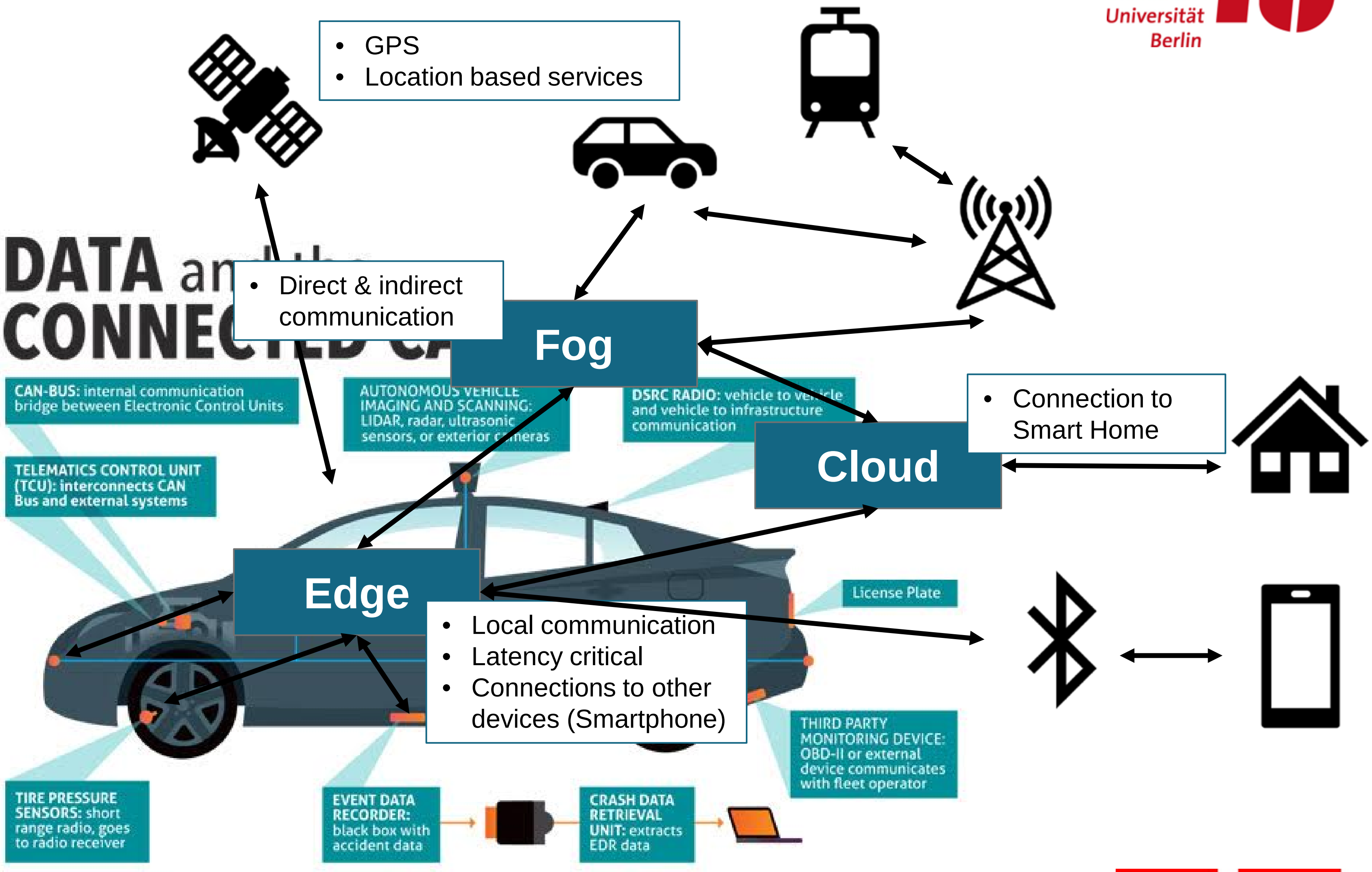

Cloud
Communication with external Services
Link to other factories
Edge
Local communication
Latency critical
Privacy restrictions

eHealth



https://blogs.vmware.com/edge/2018/02/20/managing-connected-devices-healthcare-part-4/

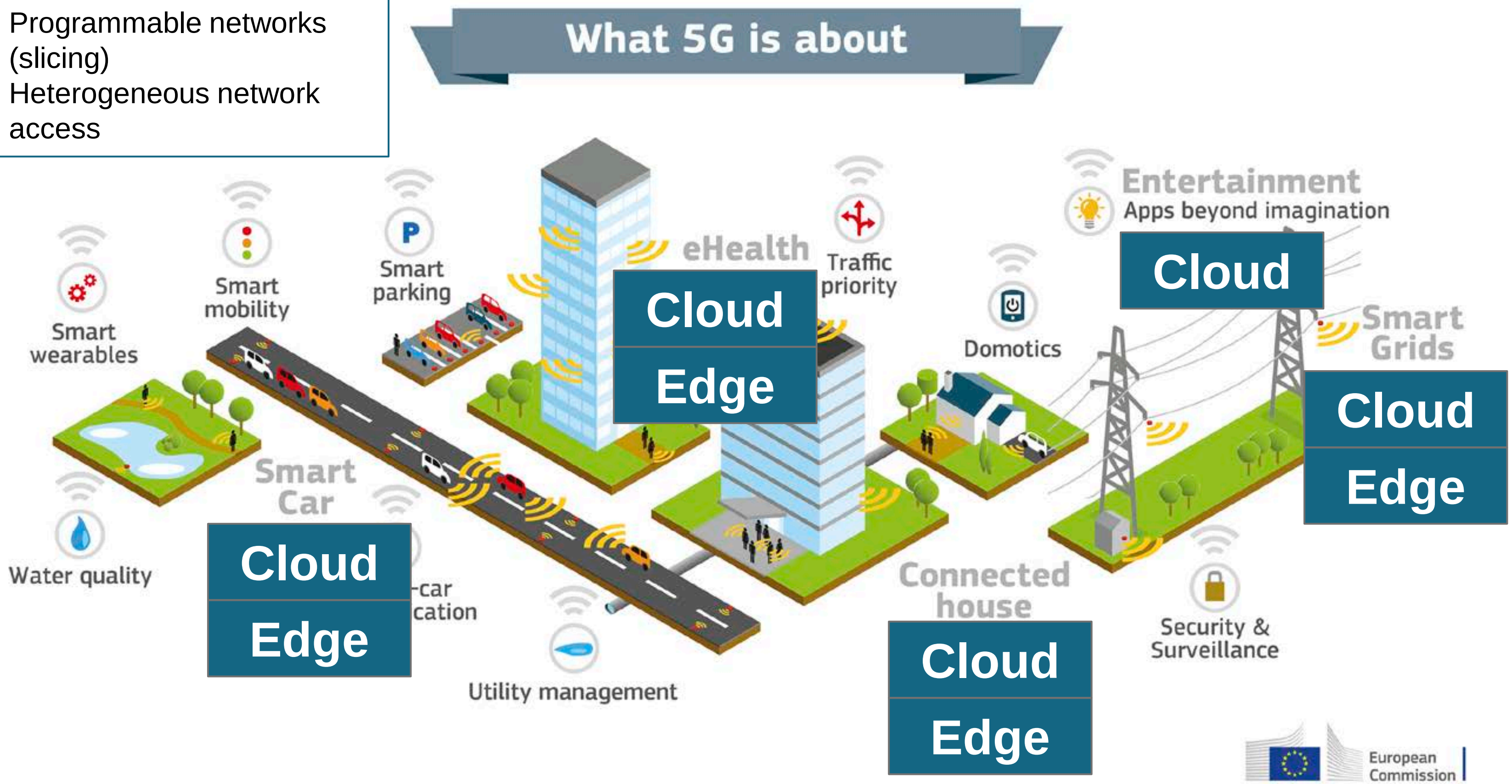
Connected Vehicles



http://www.iteveryone.com/real-time-applications-of-iot/

5G

- More bandwidth & speed
- Massive amount of connections
- Programmable networks (slicing)
- Heterogeneous network access



https://www.lteto5g.com/5g-internet-2020-eu-release-700-mhz-band-wireless-broadband/

Fog Computing Benefits many Areas

Data Collection, Analytics & Privacy

- Sending data over limited network connections to a centralized analytics system is counterproductive
- Pre-analyzing data near the Edge can save bandwidth
- Also enables anonymization of sensitive data before it is sent to the Cloud

Security

- IoT devices are often vulnerable to attacks
- Moving security closer to these devices enables higher performance security applications
- Also adds additional layers of security to protect the core against breaches and risks

Fog Computing Benefits many Areas

Compliance Requirements

- Broad range, e.g., geofencing, data sovereignty, and copyright enforcement
- Fog computing makes it possible to
  - Restricting data access based on geography and political boundaries
  - Limiting data streams depending on copyright limitations
  - Storing data in places with specific regulations

Real-Time

- AR/VR, connected cars, telemedicine, industry 4.0, or smart city applications can be very sensitive to latency or jitter
- Doing necessary computations closer to the Edge helps to reduce latencies

CHALLENGES TO ADOPTION

Main Obstacle for Adoption of Fog Computing

Inherent Obstacles

- Result from the very idea of using Fog resources
- Technical constraints, e.g., limit of computational power
- Logical constraints, e.g., tradeoffs in distributed systems
- Market constraints, e.g., there are currently no managed Edge services

External Obstacles

- Result from external entities
- Government agencies
- Attackers

Inherent Obstacles for Fog Computing Adoption

No Edge Services

- No on-demand Edge infrastructures available (yet)
- Fog application providers currently have to build, manage, and run their own physical “Edge machines” and setup Cloud integration
- Will be available sooner or later, e.g., AWS Greengrass is a first step

Lack of Standardized Hardware

- Edge machines come in a variety of flavors
    - Raspberry Pi
    - Google Edge TPU
    - …
- Software stacks need to be adapted / have to run everywhere
- Application architecture need to be modularized so that they can deliver some or all service features depending on the available resources

Inherent Obstacles for Fog Computing Adoption

Management Effort

- To cover the "entire Edge", many Fog nodes are needed
- As no managed Fog infrastructure service exists, management is left to the application providers

Managing QoS

- Issues such as network latencies, message reordering, message loss, network partitioning, or byzantine failures are more pronounced in geo-distributed systems than in centralized systems
- At the same time, Fog application domains such as IoT or autonomous driving have *stronger* quality requirements

Inherent Obstacles for Fog Computing Adoption

Technische Universität Berlin

No Network Transparency

- Fog application services can no longer be based on high-level abstraction (e.g., AWS regions or availability zones)
- Applications need to be aware of actual network topologies, as two Edge nodes may be …:
    - … connected directly
    - … connected at a higher hierarchy level
    - … connected via the internet backbone
- Otherwise, there are unexplainable performance variance across nodes

➢ Calls for the introduction of novel, more fine-grained topology abstractions that can be handled inside application services

Mobile Cloud Computing

EINSTEIN CENTER Digital Future

External Obstacles for Fog Computing Adoption

Physical Security

- Traditional approaches such as onsite security staff is not possible for small and geo-distributed sites
- In the Fog, physical security may mean:
    - Attaching hardware on the top of a street light's pole instead of on eye level
    - Protecting hardware with a fire-resistant coating to counter vandalism
    - …
- Attackers can easily gain a physical attack vector into software systems
- How can physical attacks be detected, and how should a Fog node react?

External Obstacles for Fog Computing Adoption

Legal and Regulatory Requirements

- Some application domains such as health care require data to be held in certain physical locations
- Cloud data centers can be certified if they exist in certain regions
- Seems impractical for Fog nodes
- Liquid Fog-based applications might have trouble fulfilling certain aspects of privacy regulations such as transparency about the storage location of personal data

Open Challenges

- Addressing storage latency over WAN connections
- Reinforced security at the Edge:
    - Monitoring the physical and application integrity on each site
    - Autonomously enable corrective actions when necessary
- Monitoring resource utilization across all nodes simultaneously
- Orchestration tools that manage and coordinate many Edge sites and workloads, potentially leading toward a peering control plane or “self-organizing Edge”
- Orchestration of a federation of Edge platforms (or cloud-of-clouds) has to be explored and introduced to the IaaS core services
    - Automated Edge commission/decommission operations
    - Software Deployment and Upgrades
    - Load balancing across geographically distributed hardware

Open Challenges

- Methods to synchronize state that tolerate discontinuous network links
- Ways to deal with network partitioning issues (short and long-term)
- Tools to manage Edge application life cycles, including:
    - The definition of advanced placement constraints in order to cope with requirements of application components
    - The provisioning/scheduling of applications in order to satisfy placement requirements (initial placement)
    - Data and workload relocations according to internal and external events
- Location Awareness: only deploy applications where needed
- Ways to share resources with neighbors, as the ability to expand becomes more limited when moving towards the Edge
- Benchmarking and Testing approaches
- Ways to comply with privacy regulations

OPENFOG ARCHITECTURE

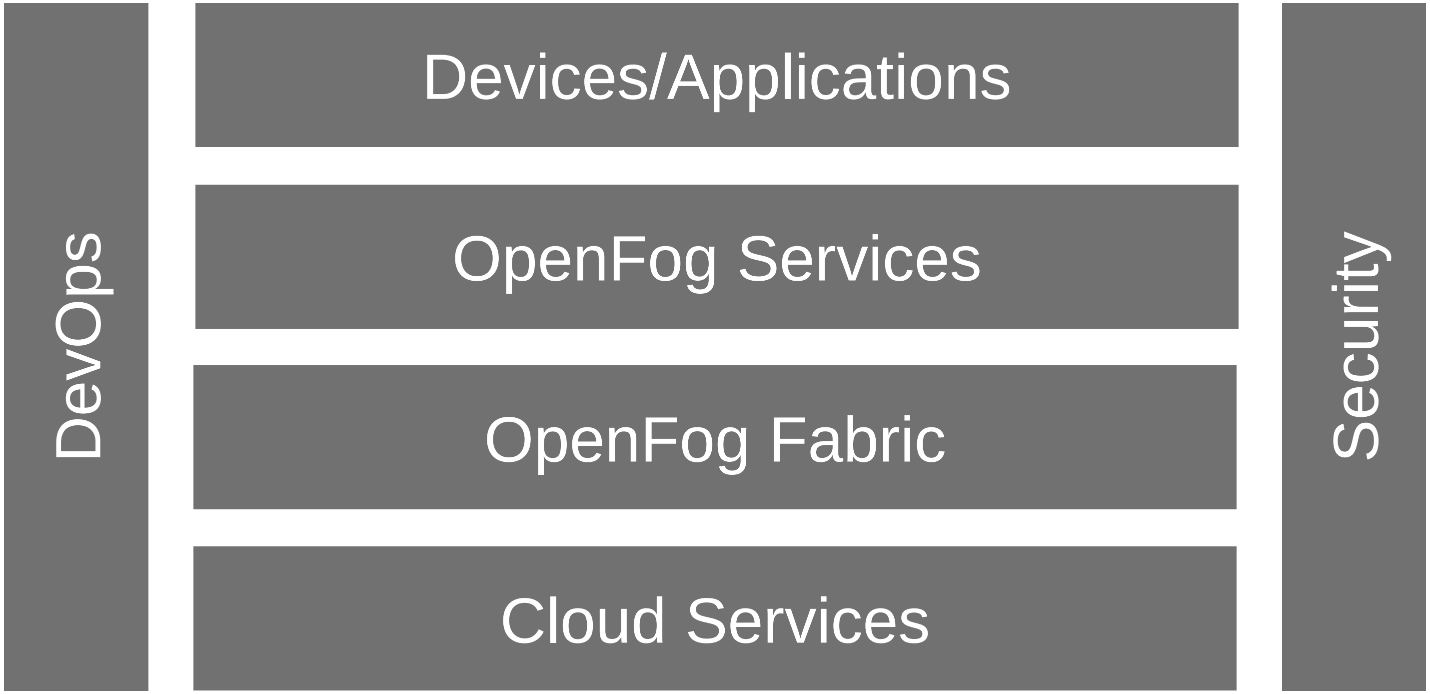
The OpenFog Architecture

- Is a system-level architecture that extends elements of compute, networking, and storage across the Cloud through the Edge of the network
- Defined by the OpenFog Consortium (formed in November 2015)
- Goal is to provide a set of methods and tools to create a continuum between Edge and Cloud
- Based on the premise that an open architecture is essential for the success of an ubiquitous Fog Computing ecosystem for IoT platforms and applications

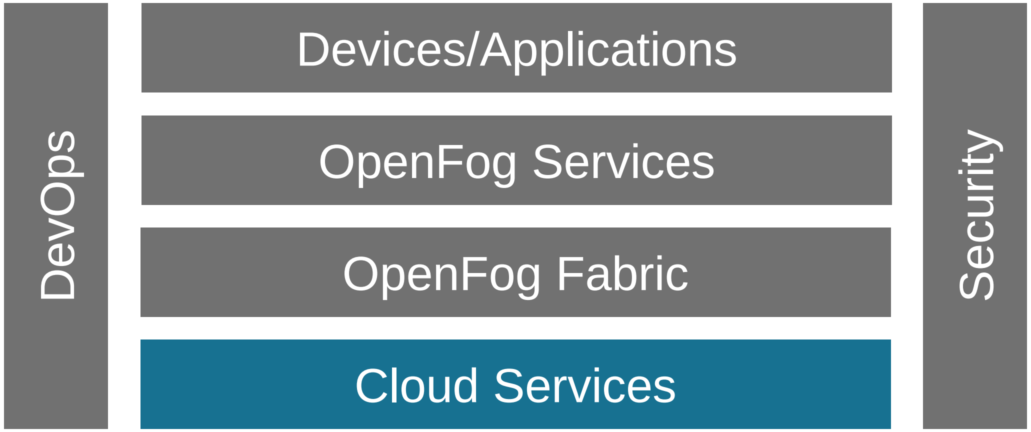
Open Fog Architecture: CEAL Advantages

- Cognition: awareness of client-centric objectives to enact autonomy
- Efficiency: dynamic pooling of local unused resources from participating end-user devices
- Agility: rapid innovation and affordable scaling under a common infrastructure
- Latency: real-time processing and cyber-physical system control

OpenFog: Building Blocks to Enable “Fog as a Service”

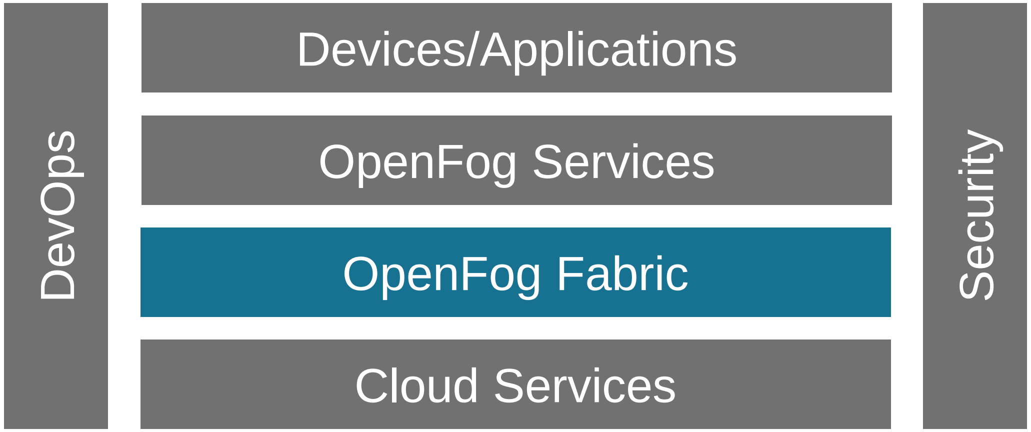


OpenFog: Building Blocks to Enable “Fog as a Service”



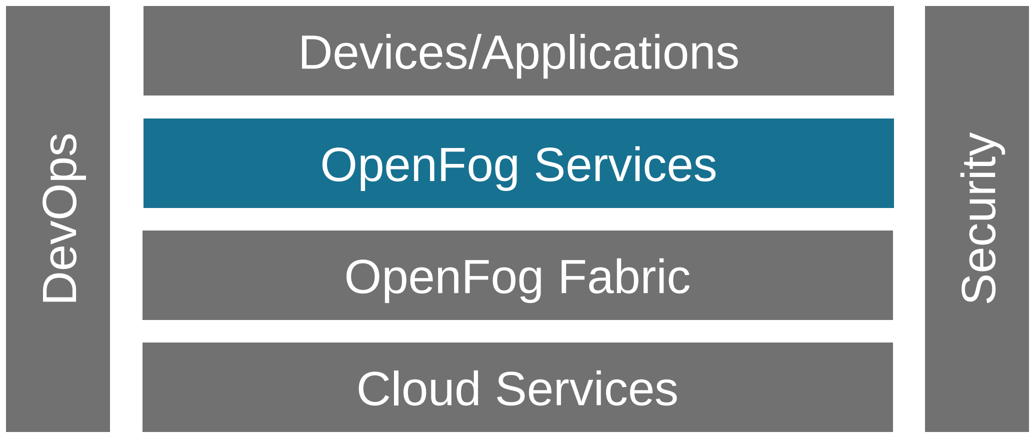
- Take advantage of the Cloud for computational work that needs to operate on a larger data scope
- Should be leveraged in a way that does not impede operational autonomy

OpenFog: Building Blocks to Enable “Fog as a Service”



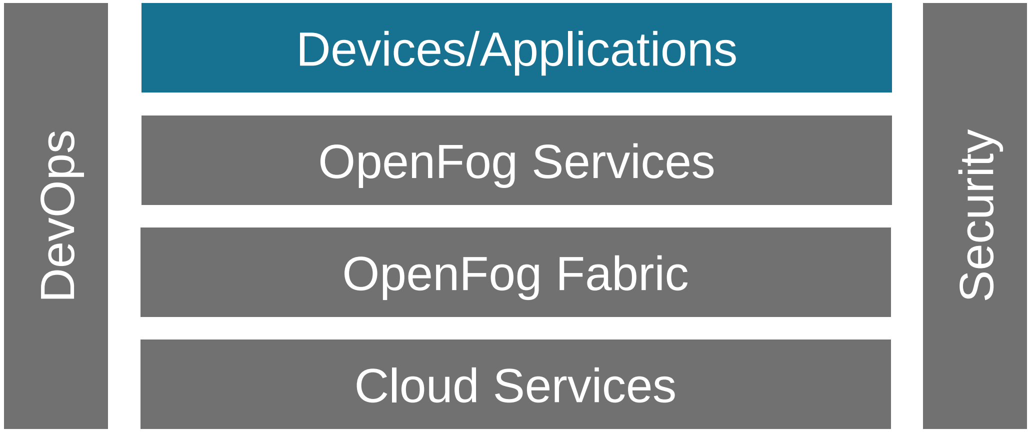
- Allows the construction of a homogenous computational infrastructure
- Infrastructure delivers services to the surrounding ecosystem
- Homogenous Infrastructure is usually build upon heterogenous hardware

OpenFog: Building Blocks to Enable “Fog as a Service”



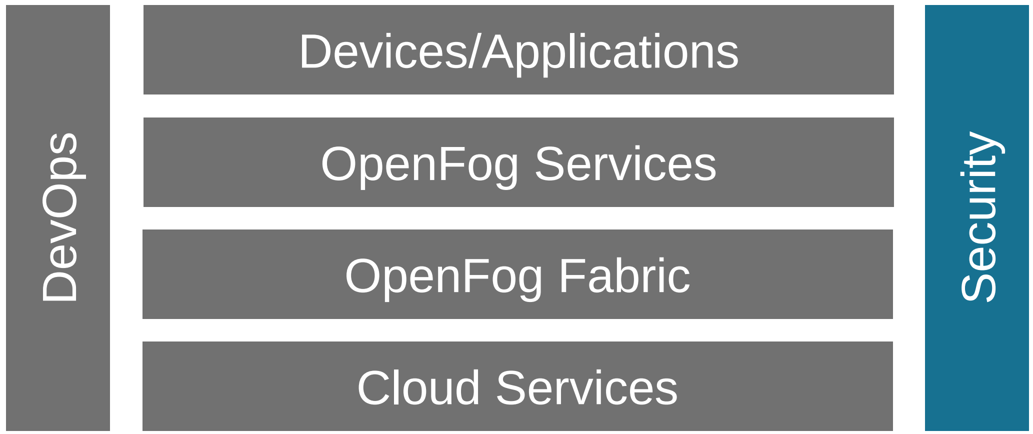
- Built upon the OpenFog Fabric infrastructure
- Most likely a micro-service architecture
- Services may include SDN, NFV, content delivery, device management, event processing, video encoding, protocol bridging, traffic offloading, crypto, analytics, etc.

OpenFog: Building Blocks to Enable "Fog as a Service"



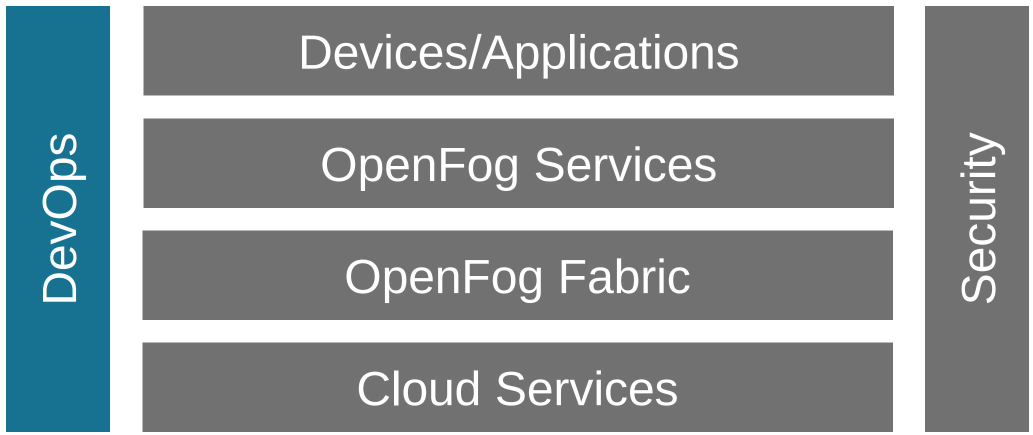
- Edge sensors, actuators, and applications
- Running standalone, within a Fog deployment, or spanning Fog deployments

OpenFog: Building Blocks to Enable “Fog as a Service”



- Discrete units of functionality are wrapped with access control mechanisms
- Data transfers between participating endpoints are secured through state of art information security practices

OpenFog: Building Blocks to Enable “Fog as a Service”



- Drive the agility of software upgrades and patching
- Build upon and control the continuous integration process

OpenFog Pillars

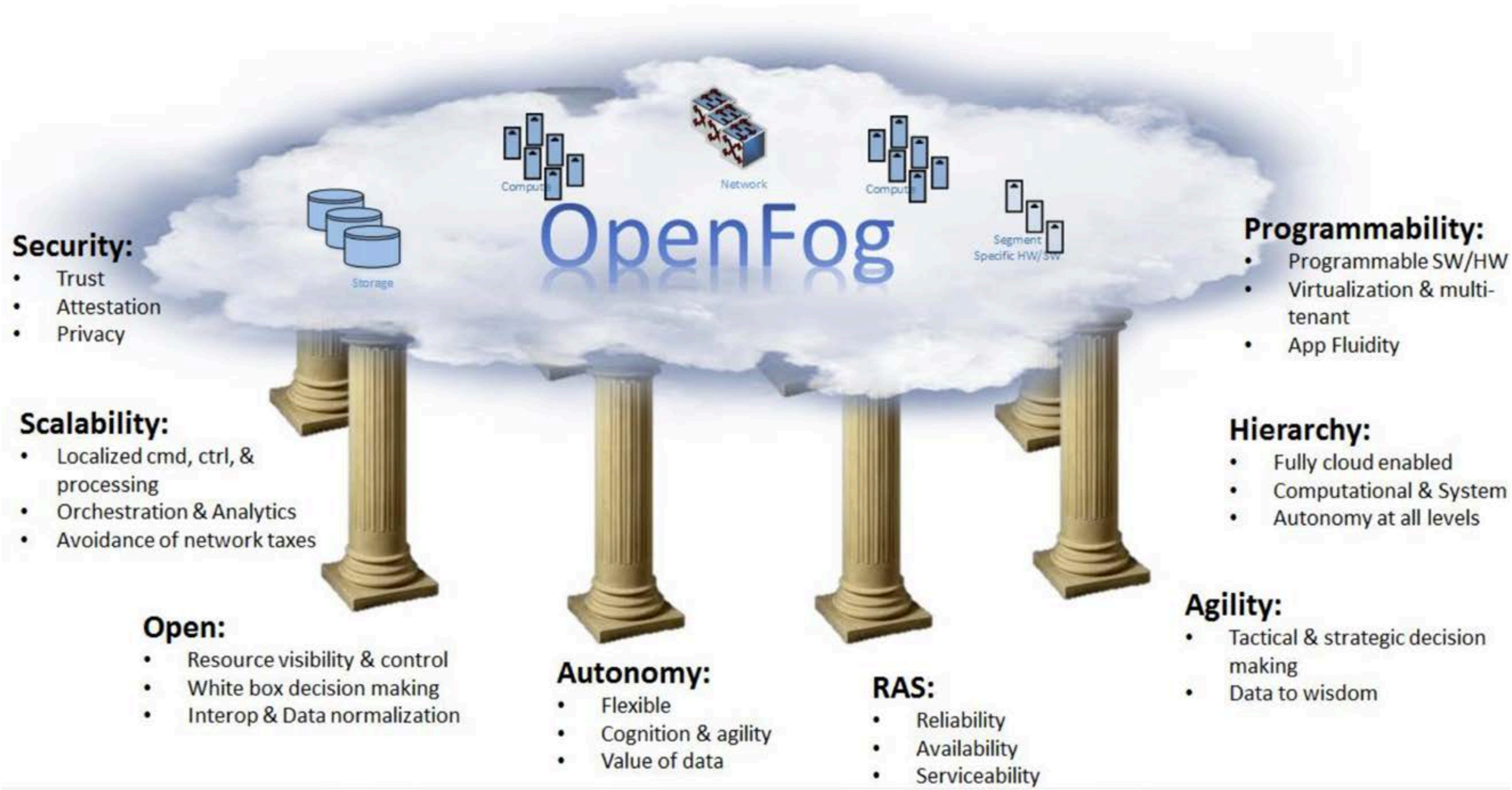


Image: OpenFog Consortium White Paper (www.OpenFogConsortium.org)

Summary

- Fog Computing combines cloud, edge, and resources in between to address
    - Latency needs
    - Bandwidth limitations
    - Privacy concerns

- Challenges such as usability hamper adoption

- Groups like the OpenFog Consortium develop reference architectures and work towards standardization

Questions?