Otto-Friedrich-University of Bamberg

Professorship for Computer Science,

Communication Services, Telecommunication
Systems and Computer Networks

Seminar on

Modern Internet Services in Software Defined and
Information Centric Networks of the next Generation

Topic:

Fog Computing

Submitted by:
Frank Kaiser

Supervisor: Prof. Dr. Udo Krieger

Bamberg, October 9, 2017
Sommersemester 2017

Contents

List of Figures

I. Introduction

Cloud computing is a old paradigm and ideas that went in the same direction exist since the 1990s. Since then many new technologys emerged that first made the cloud possible at all and then forced it to adapt to new standards. From being super centralized we went all the way to a very distributed infrastructure, but now we begin to face new challenges. Challenges about the huge masses of data we generate and that need to be processed, about our privacy and how we can decide what data we want to publish. Challenges of network congestion because you can't send unlimited amounts of data through the pipes. To face and to try to solve these and many more challenges a new mutation of cloud computing is emerging. Fog Computing takes a step back from the super distributed approach and lets data get processed and filtered near the end user so that only relevant data will be pushed upstream. The technical progress in hardware allows even the most basic devices to process data with less latency than sending raw data to the cloud and waiting for it to come back. In this paper I want to give a overview over the chances of Fog Computing, how it integrates within the cloud and how it helps to solve rising challenges in the field of the Internet of Things.

II. Definitions

A. Cloud Computing

"The cloud is just someone else's computer" is a wide-spread joke. And while it can be correct, it is not complete. According to the NIST, 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."[2, p. 2] Additionally the model is defined by five essential characteristics, three service models and four deployment models:

Essential Characteristics:

- On-demand self-service: A consumer can 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 accessed through standard mechanism 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. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
- 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: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type

of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

- Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings.
- Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models:

- Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
- Community cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- Public cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
- Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

([2, p. 2f])

B. Fog Computing

"Fog is an emergent architecture for computing, storage, control and networking that distributed these services closer to the end-users along the cloud-to-things continuum". [3, p. 854]. However, Fog Computing is not a strict architecture that has a clear definition, "instead it represents a notion that supports to push data analyrics towards leaves(i.e. edge nodes)"[1, p. 3]. Chiang and Perera et al. both stress that the important part of the definition is that tasks are not supposed to be completed by the edge nodes only, but on every node on the path to the cloud server as needed by the use case.

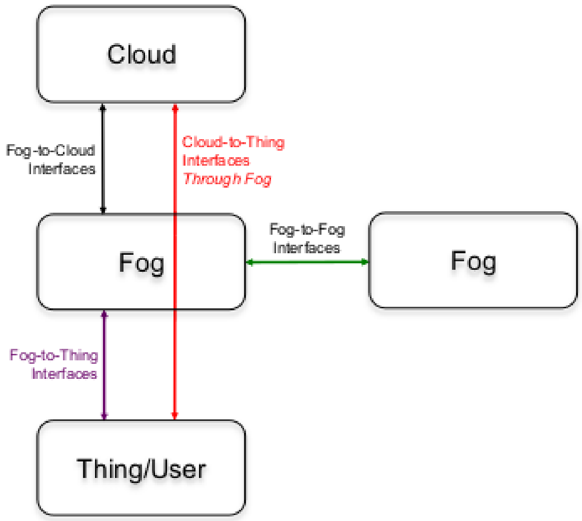

Figure 1: Fog-interfaces[3, p. 862]

As seen in Figure 1 Fog is not designed to replace the cloud, but to play along with it in order to solve rising challenges, especially in the field of IoT. The end point of such a network is called fog node and each one has at least one computing server to which all of its edge devices connect to. The Fog node controller than merges, stores, computes or redistributes data gathered or created by the edge devices. It is also capable of assigning tasks to the computing power of the edge devices. The node itself can either be controlled by another controller node that manages all edge nodes or commits to send selected results directly to a cloud server.

C. Glossary

A Fog Node describes the combination of multiple edge nodes/IoT devices/sensors and a fog node controller which acts as gateway. However Fog Nodes can also inherit multiple other fog nodes as seen in Figure 2. Since a sketch of a fog architecture often looks like a tree, fog nodes at the edge of a network are usually referred as leave nodes.

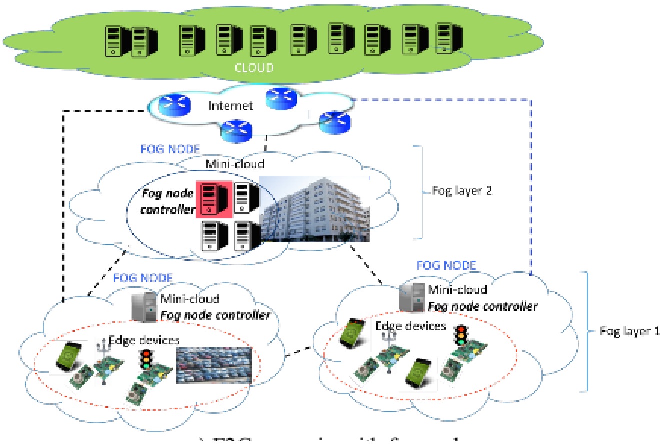

Figure 2: Fog2Cloud Architecture[4, p. 4]

III. Fog and IoT

A. Rising Challenges of IoT

The Internet of Things is a very promising but also very criticised field. Approximately 17 - 30 billion IoT devices are assumed to be online by 2020[1], in cases where it makes sense and where it doesn't. While for investors this can be a huge opportunity, these devices open a very big attack surface and implementing the fog architecture can make things a lot easier. Chiang is naming 5 different main challenges [3, p. 855 ff.]:

A Stringent Latency Requirements: Control systems require latencies between sensor and control node to be within few milliseconds. Some even within tens of a millisecond.

B Network Bandwidth Constraints: Large amount of data generated are generated by sensors(e.g.: up to one gigabyte per second for an autonomous car) requiring high bandwidth if sent to the cloud.

C Resource-Constrained Devices: IoT devices and micro-controllers often have only the resources they need to work. Everyone of them communicating directly with the cloud will cause a lot of overhead.

D Cyber Physical Systems: Cyber-physical systems are often required to be online all the time, especially in safety-critical areas.

E Uninterrupted Services With Intermittent Connectivity to the Cloud: Functionality of a system has to be working even without network connectivity.

F New Security Challenges:

1) Keeping Security Credentials and Software up to Date on Large Number of Devices.
2) Protecting Resource-Constrained Devices.
3) Assessing the Security Status of Large Distributed Systems in Trustworthy Manner.
4) Responding to Security Compromises Without Causing Intolerable Disruptions.

[1] http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

B. How Fog helps to deal with these challenges

The main advantages of the emergent fog architecture are often illustrated as CEAL [3, p. 858] and [5, p. 7]:

- Cognition: A fog node is aware of its environment to react fast to events and if needed redistribute the workload in an optimal manner.
- Efficiency: Dynamically using and redistributing unused resources from end user devices
- Agility: Rapid innovation and affordable scaling.
- Latency: Real-time processing and cyber-physical system control.

1) Dynamic discovery: The Internet of Things is a very heterogeneous field. There is a large variety in sensors, that output different kinds of data over different kinds of communication protocols. That's why a adequate architecture needs to be capable of dynamic discovery and connection of edge devices while providing a high security standard.
The discovering of edge nodes can function in two different ways (see)Figure 3):

- By letting the edge nodes search continuously for fog gateways which wait for a connection request
- By letting the fog gateway search for edge nodes in range and leave the edge devices discoverable

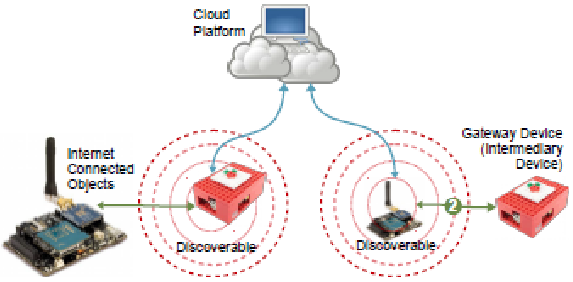
Cloud Platform
Internet Connected Objects
Gateway Device (Intermediary Device)
2
Discoverable
Discoverable

Figure 3: Two ways of discovering edge nodes[1, p. 16]

Both methods usually use low power communication protocols which will be described in detail in section V

2) Latency: Due to evaluating and gathering data closer to the end user, systems can react faster on changing circumstances. This way information does not need to travel to a distant server and back, saving bandwidth and time.

3) Network Bandwidth: Fog controller nodes are capable of distributing the workload along the edge node to cloud continuum in a optimal manner, resulting in less data being sent over the network.

4) Uninterrupted Service: A Fog node is independent of the Internet. It is still able to get data or distribute tasks to its connected edge devices while being disconnected from the rest of the world. This allows local services to stay up even if the node is disconnected.

5) Security: To ensure secure connections from an edge node to the cloud all "things" must "employ a hardware-based immutable root of trust."[5, p.10] i.e. to ensure security the hardware itself needs to be protected from unauthorized manipulation. Then on top of that software agents running throughout the infrastructure must be able to ensure the integrity of the root of trust. Due to the nature of edge nodes in being close to the end user, fog nodes often act as first node of access control and encryption enabling the user to decide what privacy-sensitive data should be aggregated before it leaves the node.[5, p.10]

IV. OpenFog Architecture

A. The OpenFog Consortium

The OpenFog Consortium is a composite of multiple big players founded in 2015 in order to solve challenges the usual and wide-spread, centralized cloud architecture could not handle. The founding members were Arm, Cisco, Dell, Intel, Microsoft and the Princeton University. Right now there are 55 members and the consortium is getting larger as more organizations realize the potential of the Fog Architecture.

B. Fog as a Service

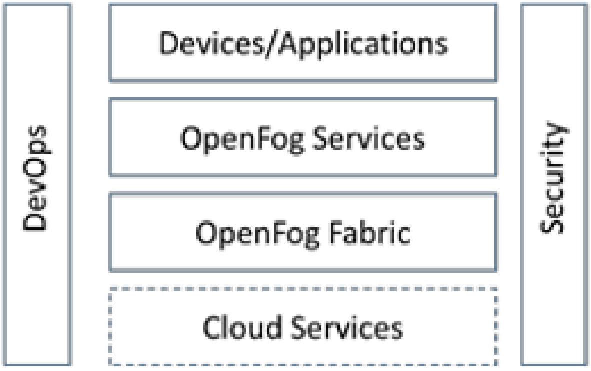

Figure 4: OpenFog Infrastructure View[5, p. 8]

As stated in Section II-B Fog's goal is to extend existing cloud architectures and not to replace them. Such Cloud infrastructures are usually offered as Platform as a Service (PaaS).

OpenFog Fabric consists of nodes or layers and may be either centralized or distributed and may be implemented on dedicated hardware, software or both. No matter what of these possibilities is chosen, such fabric distributes resources across available devices, systems and clouds in order to achieve the goal while fulfilling all requirements.[5, p. 7]

OpenFog Services are built upon the OpenFog fabric infrastructure. Example services are network accerleration, NFV, SDN, content delivery, device management, device topology, complex event processing, video encoding, field gateway, protocol bridging, traffic offloading, crypto, compression, analytics algorithms/libraries etc.[5, p. 8]

Devices/Applications are sensors, microcontrollers, IoT devices running standalone, within a fog deployment or spanning fog deployments.[5, p. 8]

CloudServices are services that use the cloud for computations and/or work with a large data set or pre-processed edge data.[5, p. 8]

Security is important throughout all levels of the architecture as indicated through the vertical pillar in Figure 4. Every unit in every layer needs to participate in state of the art information security practices.[5, p. 8]

DevOps (compund word for 'development' and operations') is a software engineering practice which main characteristic is to strongly advocate automation at all steps of software construction. In the OpenFog architecture an efficient set of standard DevOps processes and frameworks enable automation, providing the agility of software upgrades through controlled integration processes.[5, p. 9]

C. The Pillars of OpenFog

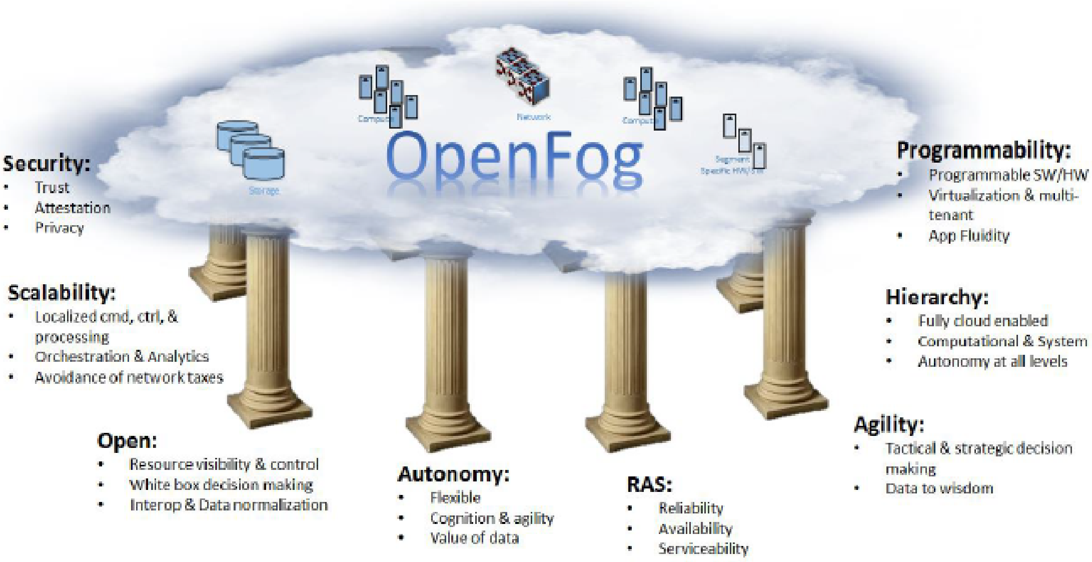

Figure 5: The Pillars of OpenFog[5, p. 9]

The OpenFog consortium defined eight pillars which build the foundation of OpenFog. These consist of: Security, Scalability, Open, Autonomy, RAS, Agility, Hierarchy, Programmability. Some of them were already mentioned in subsection III-B.

1) Scalability: When providing Fog as a Service, you need to be able to scale up and down depending on your clients need. Fog is scalable through five different dimensions:[5, p. 10]

- Scalable performance: Allows to respond to rising application demands
- Scalable capacity: Allows fog networks to grow as more edge devices get connected to the network
- Scalable reliability: Permits ”inclusion of redundant fog capabilities to manage faults or overloads”.
- Scalable Security: While being its own pillar, security needs may be depending on the use case or change over time.
- Scalability of software: By using VMs or infrastructures like docker software on nodes can be scaled up or down as load demands.

2) Open: Proprietary solutions could lead to a walled garden limiting your choice or have negative impact on cost, quality and innovation. That’s why fog is founded on a open architecture.[5, p. 11]

- Composability allows portability of programs and services at instantiation.
- Interoperability ensures secure discovery of compute, network and storage and enables portability at execution time.
- Open communication allows pooling of resources near edge networks in order to collect free processing, storage and sensing resources.
- Location transparency ensures nodes can exist anywhere in the hierarchy.

3) Autonomy: OpenFog relies on the ability to make information based decisions on the edge without waiting for confirmation or other decisions from the cloud.[5, p. 12]

- Autonomy of discovery enables resource discovery
- Autonomy of orchestration and management automates process of bringing services online
- Autonomy of security provides authentication, authorization and accounting, InfoSec etc. allowing services and devices to authenticate themselves against security services.
- Autonomy of operation enables localized decision making.

D. Programmability

Through open APIs, frameworks and runtime containers fog nodes or clusters re-tasking can be done, allowing best use of all available resources.[5, p. 13]

- Adaptive infrastructure for diverse deployment and changing business needs.
- Resource efficient deployments to maximize efficiency on resources through use of container deployment
- Multi-tenancy allowing multiple tenants in logically isolated runtime environments.
- Economical operations resulting from high density and adaptive infrastructure
- Enhanced Security to apply patches and react to upcoming threats.

E. RAS(Reliability, Availability, Serviceability)

The architecture has to be stable and deliver results under normal as well as adverse conditons. To achieve high availability and serviceability hardware, software and networking need to be run reliable alone and in conjunction. Availability enables continuous mangement and orchestration to ensure the task's goal. To achieve serviceability a highly automated installation, upgrade and repair process is essential.[5, p. 13]

F. Agility

Sensors produce a lot of data. The OpenFog architecture allows to process the data at the edge to make informed decisions quickly to maximize performance. Tactical and quick decisions sould be made near the edge, while strategical, system-wide decisions are made higher up in the fog level hierarchy.[5, p. 15]

G. Hierarchy

Depending on the scale of the use case, the hierarchy may consist of a network of smart partitioned systems in either physical or logical layers, or it might be a single physical based system. Usually you will see cloud and fog combined, but there could also be exceptions where only cloud or only fog will be better suited. The cloud provides Infrastructure as a Service(IaaS), Platform as a Service(PaaS) and Software as a Service(SaaS) and all can easily be extend to the fog architecture.[5, p. 16]

V. Communication Protocols

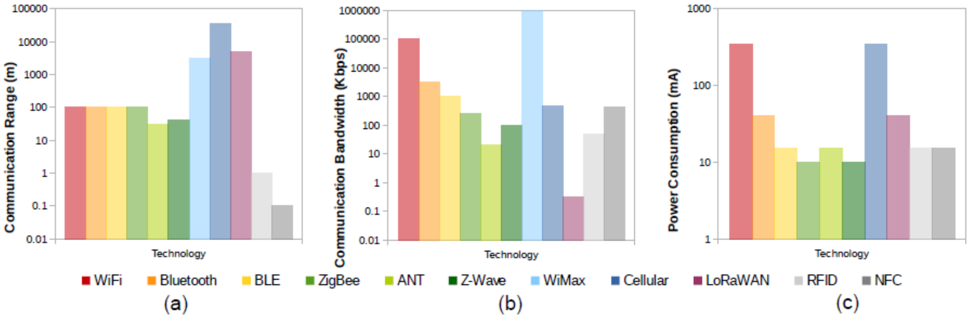

Figure 6: Comparison of communication protocols[1, p. 20]

The needs of a Fog infrastructure depend heavily on the application and luckily there are plenty different communication protocols to choose from with different strengths and weaknesses. When talking about wireless communication in the IoT field you usually consider three main characteristics: Range, Bandwidth and power consumption. As you may notice in Figure 6, most protocols are good in two of those categories but worse in the last one. For example LoRaWAN excels at range and power consumption but has very limited bandwidth.

Wi-Fi: The most common network technology also known as WLAN and a lot of IoT devices use it to connect to the Internet. It provides high bandwidth but as a drawback it need relatively much power. It's best suited for in house coverage due to its range limit of about 20 meters. Wi-Fi is the implementation of the 802.11 IEEE standard[1]

Bluetooth: Bluetooth is also implemented on a lot of devices. It is designed to enable communication over short distances. Bluetooth has 3 different classes. Class 1 providing the shortest range and therefore using the lowest power. Class 3 has the longest range but uses also a lot of power. It is widely used in IoT devices as it allows easy connection to a smart phone or other devices that can be used as a fog gateway. It's standard is the IEE 812.15.1.[1]

Bluetooth Low Energy / Bluetooth Smart: As its name suggests Bluetooth Low Energy (BLE) is a variation of Bluetooth designed to draw very low power and in opposite of standard Bluetooth is not limited on how many devices connect. BLE allows communication with up to 1 Mbps and up to 100m range while consuming less than 10 mA.[1]

Zigbee/Zigbee PRO: Zigbee is a protocol suite covering network, transport and application layer. It is used to build low-cost, low-throughput and low-power wireless mesh networks. The idea of a mesh network is to send data from one node to another until it reaches its destination. Zigbee needs a special application gateway, that needs to connect to the Zigbee network as a node and it needs to support the TCP/IP - Protocols.[1]

ANT: In ANT networks every node can be master and slave, meaning every node can send and/or receive data to enable networking. It is designed for low bit-rate and low power sensor networks and can connect up to 65533 devices. It supports point-to-point, star, tree and mesh topologies. However, the protocol is proprietary.[1]

Z-Wave: is a certification designed to run on low powered battery operated devices. In contrast to ANT, Z-Wave has explicit master and slave nodes. Z-Wave operates in mesh networks and can connect up to 232 nodes.[1]

WiMax: Short for "Worldwide Interoperability for Microwave Access" and describes a set of standards for wireless communication. Its standard is IEEE 802.16. In comparison to Wi-Fi it provides higher bandwidth (up to 1 Gbps) over longer range (3 km) and can connect more nodes. It is a competitor to LTE.[1]

Cellular (GSM/HSPA): High Speed Packet access protocol gives users the opportunity to send data over existing networks over a long range. However, you will have to pay to use it. Depending on the technology, data can be transmitted on different speeds. Ranging from 35-170 kbps (GPRS) to 3-10 Mbps (LTE).[1]

LPWAN (LoRaWAN): Low Power Wide Are Network is designed to, as its name suggests, transmit data over long ranges and using low power. As drawback the bandwidth is quite low. Usually a Star-of-stars topology is used in LPWAN networks. It range and throughput is heavily dependent on the environment. In urban areas 2-5 km can be achieved and in suburban areas 15 km are possible. In perfect conditions (line of sight between nodes) more than 100 kms can be achieved.[1]

RFID: Stands for Radio-frequency identification. It makes use if electromagnetic fields to transmit data. Tags contain electronically stored information. Active RFID can communicate up to 100 m and passive around 100 cm. RFID tags can be found on cards, stickers etc. and are used in IoT applications to enrich non-electronic objects.[1]

NFC: Near field communication enables data transfer between devices in range of 10 cm. It supports peer-to-peer communication. It is most commonly used to enable communication between smart devices.[1]

VI. Use case studies

In this section I will go over several use cases where fog has been or can be applied.

Fog architecture consists of data plane and control plane[3, p. 860]. Data plane describes user driven data while the control plane is the part of the network that carries signaling data and is responsible for routing. Chiang and Zhang name several examples:[3, p. 860]

Data Plane of Fog:

- pooling of clients idle computing/storage/bandwidth resources
- content caching at the edge and bandwidth management at home
- client-driven distributed beam-forming
- client-to-client direct communications
- cloudlets and micro data-centers

Control Plane of Fog:

- over the top (OTT) content management
- fog-RAN: Fog driven RAN
- client-based HetNets control
- session management and signaling load at the edge
- crowd-sensing inference of network states
- edge analytics and real-time stream-mining

A. Data Plane Usecases

Smart Agriculture: In agriculture a lot can be done to optimize the processes. Using sensors on the field to measure nutrition of earth, the level of dryness, monitor farm animals and countless more possible applications and processing them directly on the edge node it is possible to extract relevant information to achieve automation. A fog node could for example gather data about future weather from the cloud and data about current weather and humidity of earth from sensors and control if the field has to be watered.

Smart Health and Well-Being: Arm bracelets measuring your pulse, jackets monitoring your heartbeats, your toilet analyzing your outputs. All these methods can have positive impacts on your well-being. Though the data is very sensitive and could be abused if fallen in false hands. Therefore you would not want to send it directly to a cloud where your doctor or your medical insurance could see it. By gathering the data on a local fog node and preprocessing it, letting a program suggest based on data whether you need medical aid, you are able to stay in control of your data. You decide what kind of information you want to share with your doctor.

Smart Greenhouse Gases Control: Controlling gases in our atmosphere is a task that needs a lot of forces working together. Fog can help in making smart, information-based decisions. In a smart city every unit that emits gases can measure it and store information in local fog nodes. These nodes could then give a warning to its location if needed or suggest measures to improve the situation. By then combining all these different data in a cloud, it is easier to detect the biggest factor in air pollution, allowing the government to make smart, information-based decisions.

B. Control Plan Usecases

Client-Based HetNets Control (in 3GPP Standards): Coexistence is very important in Cellular networks today. Each device is able to check its local conditions and based on them make decisions which network to join. ”The fog-cloud interface allows real-time network configurations be carried out by clients themselves.”[3, p. 860]

”Shread and Spread” Client-Controlled Cloud storage: Instead of sending a whole file to the cloud and let it do analysis, a file could be ’shredded’ into several parts or bytes by the client and distributed to different cloud services. This way it can be ensured the information stays private even if the encryption key gets leaked.[3, p. 861]

Bandwidth Management at Home Gateway: The limited bandwidth capacity in a home gateway is allocated among users and application sessions based on each sessions priority and needs. ”A protoype on a commodity router demonstrates a scalable, economical and accurate control of capacity allocation on the edge”[3, p. 861]

Real-Time Stream Mining for Embedded AI: Considering virtual reality tasks associated with Google glass. Some tasks could be done by the glass (a ”wearable thing”), some on home storage (edge device) and the rest in the cloud. By refining information on each step to the cloud, tasks may be distributed in a intelligent manner.[3, p. 861]

VII. Conclusion

Fog is here. And it is here to stay. Fog is just at its start and as the cloud grew over the last 20 years, fog will follow suit and change over the course of its lifespan. Though the basic mantra to let computation, storage and control occur closer to the end user will stay until new drastic technical progress will be made.

References

[1] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 32:1–32:43, Jun. 2017. [Online]. Available: http://doi.acm.org/10.1145/3057266

[2] P. Mell and T. Grance, "The nist definition of cloud computing," 2011. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[3] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016. [Online]. Available: https://doi.org/10.1109/JIOT.2016.2584538

[4] E. Marín-Tordera, X. Masip-Bruin, J. G. Almiñana, A. Jukan, G. Ren, J. Zhu, and J. Farre, "What is a fog node A tutorial on current concepts towards a common definition," *CoRR*, vol. abs/1611.09193, 2016. [Online]. Available: http://arxiv.org/abs/1611.09193

[5] "Openfog architecture overview." [Online]. Available: https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf