FOG COMPUTING: PRINCIPLES, ARCHITECTURES, AND APPLICATIONS

Prepared By,
SHELLY SHIJU GEORGE
ASSISTANT PROFESSOR

INTRODUCTION

- Internet of Things (IoT) environments consist of loosely connected devices that are connected through heterogeneous networks.
- In general, the purpose of building such environments is to collect and process data from IoT devices in order to mine and detect patterns, or perform predictive analysis or optimization, and finally make smarter decisions in a timely manner.

Data in such environments can be classified into two categories:

- **Little Data or Big Stream**: transient data that is captured constantly from IoT smart devices
- **Big Data**: persistent data and knowledge that is stored and archived in centralized cloud storage

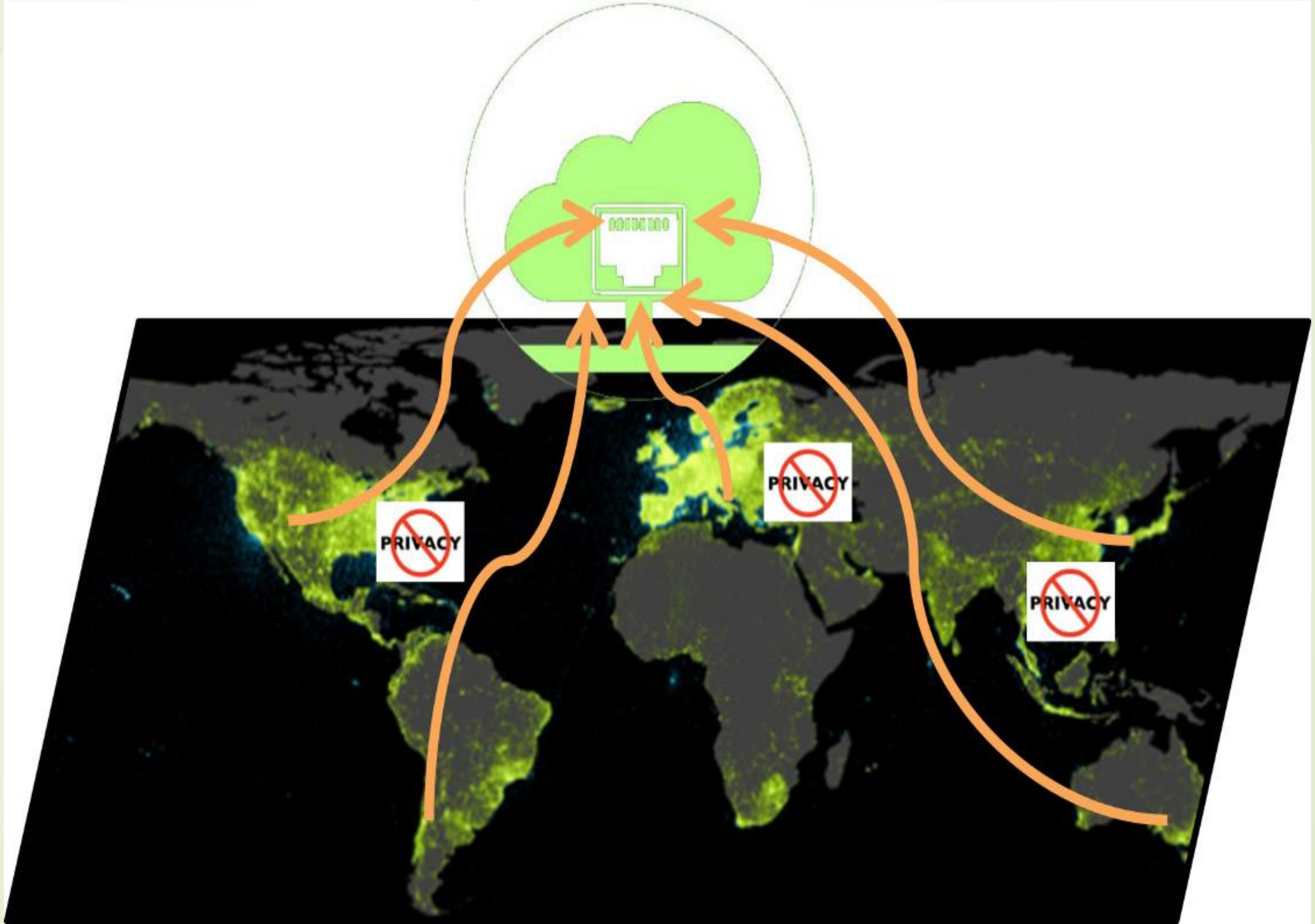
- IoT environments, including smart cities and infrastructures, need both Big Stream and Big Data for effective real-time analytics and decision making.
- This can enable real-time cities that are capable of real-time analysis of city infrastructure and life, and provides new approaches for governance.
- At the moment, data is collected and aggregated from IoT networks that consist of smart devices, and is sent uplink to cloud servers, where it is stored and processed.

- Cloud computing offers a solution at the infrastructure level that supports Big Data Processing.
- It enables highly scalable computing platforms that can be configured on demand to meet constant changes of application requirements in a pay-per-use mode, reducing the investment necessary to build the desired analytics application.
- This perfectly matches requirements of Big Data processing when data is stored in centralized cloud storage.
- In such a case, processing of a large magnitude of data volume is enabled by on-demand scalability of Clouds.
- However, when data sources are distributed across multiple locations and low latency is indispensable, in-cloud data processing fails to meet the requirements.

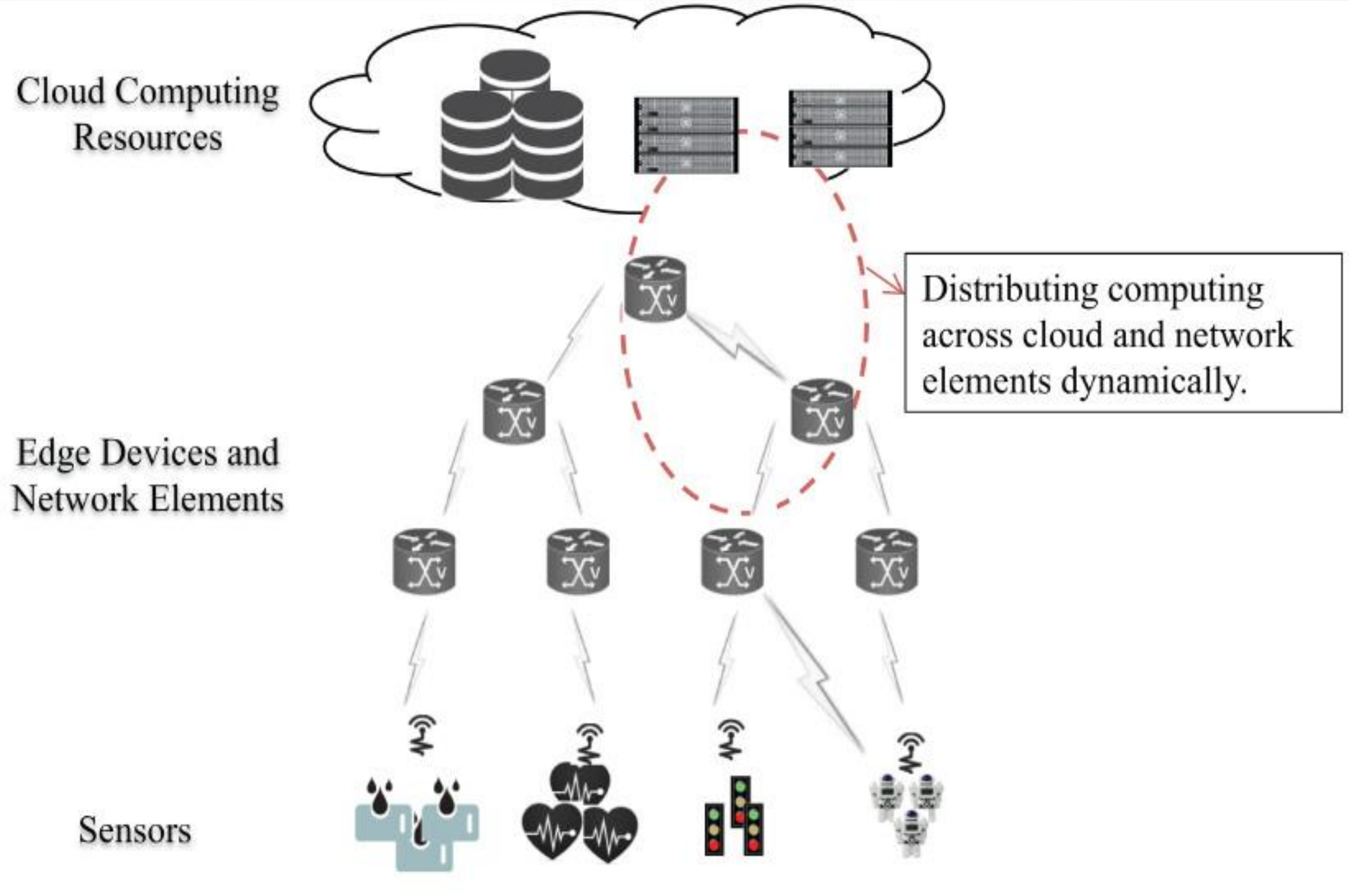
- A recent analysis of Endomondo application, a popular sport-activity tracking application, has revealed a number of remarkable observations.
- The study shows that a single workout generates 170 GPS tuples, and the total number of GPS tuples can reach 6.3 million in a month’s time.
- With 30 million users, the study shows that generated data flows of Endomondo can reach up to 25,000 tuples per second.
- Therefore, one can expect that data flows in real-time cities with many times more data sources—GPS sensors in cars to air- and noise-pollution sensors—can easily reach millions of tuples per second.
- Centralized cloud servers cannot deal with flows with such velocity in real time.

- In addition, a considerable numbers of users, due to privacy concerns, are not comfortable to transfer and store activity-track-data into the cloud, even if they require a statistical report on their activities.
- This motivates the need for an alternative paradigm that is capable of bringing the computation to more computationally capable devices that are geographically closer to the sensors than to the clouds, and that have connectivity to the Internet.
- Such devices, which are at the **edge of the network** and therefore referred to as **edge devices**, can build local views of data flows and can aggregate data to be sent to the cloud for further offline analysis.
- To this end, **Fog computing** has emerged.

Endomondo has 30 Million Users Around the Globe, Generating 25,000 Records per Second



Fog Computing is a Distributed Computing Paradigm That Extends the Cloud Services to the Edge of the Network



- It facilitates management and programming of compute, networking, and storage services between data centers and end devices.
- **Fog computing essentially involves components of an application running both in the cloud as well as in edge devices between sensors and the cloud, that is, in smart gateways, routers, or dedicated fog devices.**
- Fog computing supports **mobility**, **computing resources**, **communication protocols**, **interface heterogeneity**, **cloud integration**, **and distributed data analytics** to address requirements of applications that need low latency with a wide and dense geographical distribution.

Advantages associated with Fog computing including the following:

- Reduction of network traffic
- Suitable for IoT tasks and queries
- Low-latency requirement
- Scalability

1. Reduction of network traffic

- Cisco estimates that there are currently 25 billion connected devices worldwide, a number that could jump to 50 billion by 2020.
- The billions of mobile devices such as smart phones and tablets already being used to generate, receive, and send data make a case for putting the computing capabilities closer to where devices are located, rather than having all data sent over networks to central data centers.
- Depending on the configured frequency, sensors may collect data every few seconds.
- Therefore, it is neither efficient nor sensible to send all of this raw data to the cloud.
- Hence, fog computing benefits here by providing a platform for filter and analysis of the data generated by these devices close to the edge, and for generation of local data views.
- This drastically reduces the traffic being sent to the cloud.

2. Suitable for IoT tasks and queries

- With the increasing number of smart devices, most of the requests pertain to the surroundings of the device.
- Hence, such requests can be served without the help of the global information present at the cloud.
- Because of the local nature of the typical requests made by this application, it makes sense that the requests are processed in fog rather than cloud infrastructure.
- Another example can be a smart-connected vehicle which needs to capture events only about a hundred meters from it.
- Fog computing makes the communication distance closer to the physical distance by bringing the processing closer to the edge of the network.

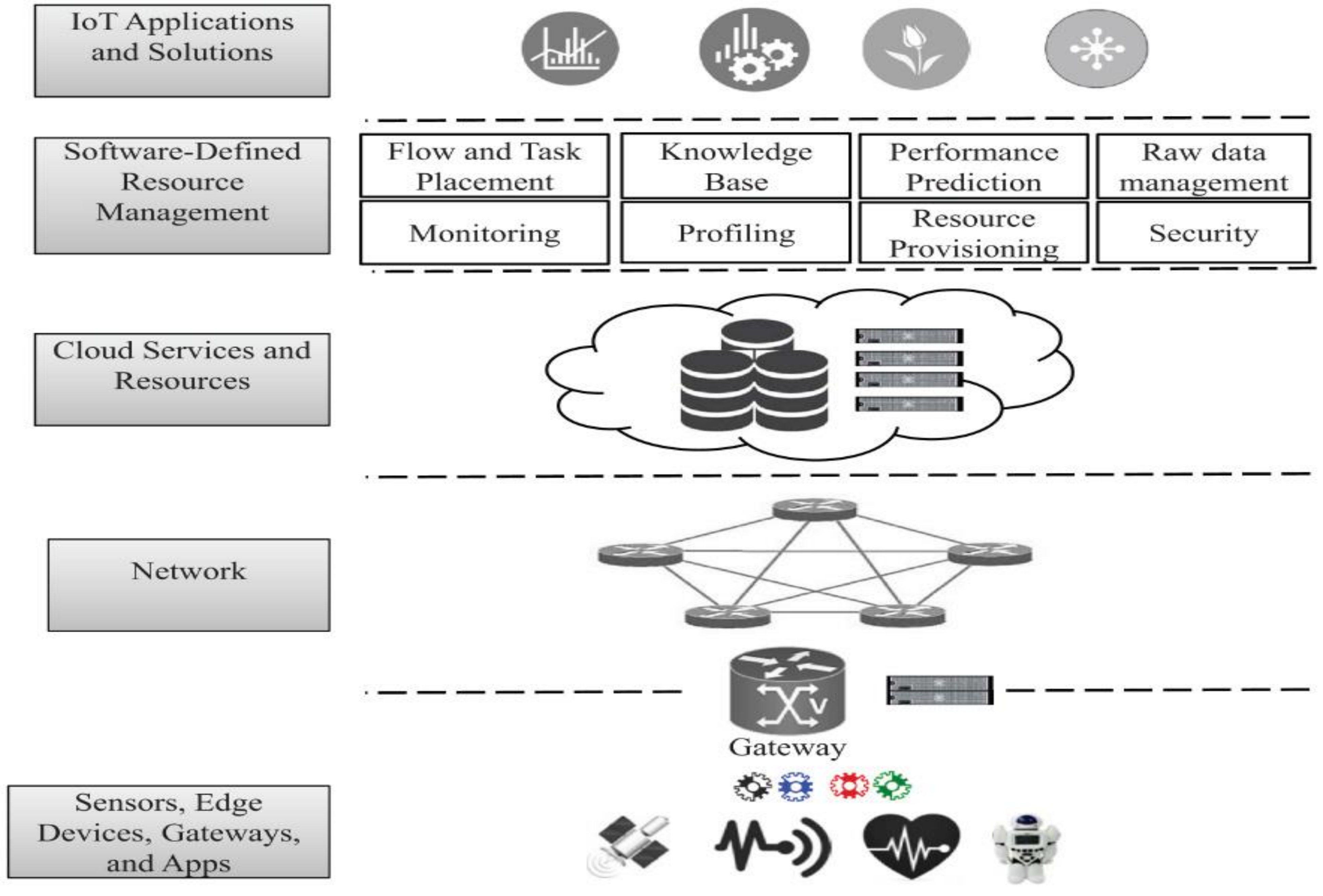
3. Low-latency requirement

- Mission-critical applications require real-time data processing.
- Some of the best examples of such applications are cloud robotics, control of fly-by-wire aircraft, or antilock brakes on a vehicle.
- For a robot, motion control depends on the data collected by the sensors and the feedback of the control system.
- Having the control system running on the cloud may make the sense-process-actuate loop slow or unavailable as a result of communication failures.
- This is where fog computing helps, by performing the processing required for the control system very close to the robots—thus making real-time response possible.

4. Scalability

- Even with virtually infinite resources, the cloud may become the bottleneck if all the raw data generated by end devices is continually sent to it.
- Since fog computing aims at processing incoming data closer to the data source itself, it reduces the burden of that processing on the cloud, thus addressing the scalability issues arising out of the increasing number of endpoints.

Fog Computing Reference Architecture



- In the bottommost layer lays the end devices (sensors), as well as edge devices and gateways.
- This layer also includes apps that can be installed in the end devices to enhance their functionality.
- Elements from this layer use the next layer, the network, for communicating among themselves, and between them and the cloud.
- The next layer contains the cloud services and resources that support resource management and processing of IoT tasks that reach the cloud.
- On top of the cloud layer lays the resource management software that manages the whole infrastructure and enables quality of Service to Fog Computing applications.
- Finally, the topmost layer contains the applications that leverage fog computing to deliver innovative and intelligent applications to end users.

- Looking inside the Software-Defined Resource Management layer, it implements many middleware like **services to optimize the use of the cloud and Fog resources on behalf of the applications**.
- The goal of these services is to **reduce the cost of using the cloud at the same time that performance of applications reach acceptable levels of latency**, by pushing task execution to Fog nodes.

This is achieved with a number of services working together, as follows.

- Flow and task placement
- Knowledge Base
- Performance Prediction
- Raw-Data Management
- Monitoring
- Profiling
- Resource Provisioning
- Security

1. Flow and task placement

- This component keeps track of the **state of available cloud, Fog, and network resources** (information provided by the Monitoring service) to identify the best candidates to hold incoming tasks and flows for execution.
- This component communicates with the Resource-Provisioning service **to indicate the current number of flows and tasks**, which may trigger new rounds of allocations if deemed too high.

2. Knowledge Base

- This component stores **historical information about application demands and resource demands** that can be leveraged by other services to support their decision-making process.

3. Performance Prediction

- This service utilizes information of the Knowledge- Base service **to estimate the performance of available cloud resources**.
- This information is used by the Resource-Provisioning service **to decide the amount of resources to be provisioned**, in times where there are a large number of tasks and flows in use or when performance is not satisfactory.

4. Raw-Data Management

- This service has **<u>direct access to the data sources</u>** and provides views from the data for other services.
- Sometimes, these views can be obtained by simple querying (eg, SQL or NOSQL REST APIs), whereas other times more complex processing may be required (eg, MapReduce).
- Nevertheless, the particular method for generation of the view is abstracted away from other services.

5. Monitoring

- This service **keeps track of the performance and status of applications and services**, and supplies this information to other services as required.

6. Profiling

- This service **builds resource- and application-profiles** based on information obtained from the Knowledge Base and Monitoring services.

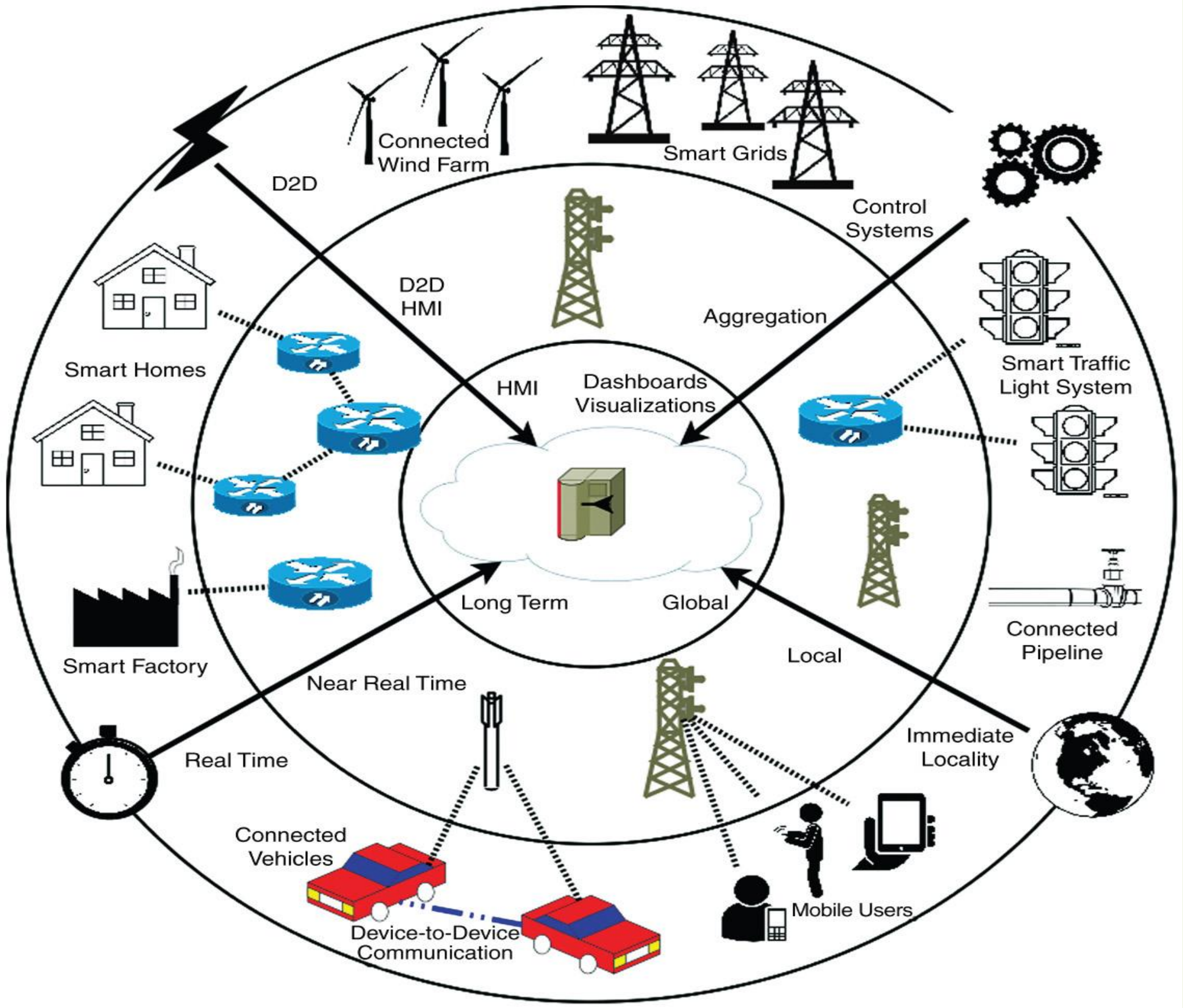
7. Resource Provisioning

- This service is **<u>responsible for acquiring cloud, Fog, and network resources for hosting the applications</u>**.
- This allocation is dynamic, as the requirements of applications, as well as the number of hosted applications, changes over time.
- The decision on the number of resources is made with the use of information provided by other services (such as Profiling, Performance Prediction, and Monitoring), and user requirements on latency, as well as credentials managed by the Security service.
- For example, the component pushes tasks with low-latency requirements to edge of network as soon as free resources are available.

8. Security

- This service **supplies authentication, authorization, and cryptography**, as required by services and applications.

Range of Applications Benefiting From Fog Computing



- There is a variety of applications benefiting from the Fog-computing paradigm.

We discuss the major applications first, and then we elaborate more on enablers and related work in the area.

- HEALTHCARE
- AUGMENTED REALITY
- CACHING AND PREPROCESSING

1. HEALTHCARE

- A set of fall-detection algorithms, including algorithms based on acceleration measurements and time-series analysis methods, as well as filtering techniques to facilitate the fall-detection process.
- A real-time fall-detection system based on fog computing that divides the fall-detection task between edge devices and the cloud.
- A three-tier architecture for a smart-healthcare infrastructure, comprised of a role model, layered-cloud architecture, and a fog-computing layer, in order to provide an efficient architecture for healthcare and elderly-care applications.
- The fog layer improves the architecture by providing low latency, mobility support, location awareness, and security measures.
- The process flow of the healthcare application is modeled using Business Process Model and Notation (BPMN) and is then mapped to devices via a service-oriented approach.

2. AUGMENTED REALITY

- Augmented reality applications are highly latency-intolerant, as even very small delays in response can damage the user experience.
- Hence, fog computing has the potential to become a major player in the augmented reality domain.
- Brain-state classification is among the most computationally heavy signal-processing tasks, but this needs to be carried out in real time.
- The system employs both fog and cloud servers, a combination that enables the system to perform continuous real-time brain-state classification at the fog servers, while the classification models are tuned regularly in the cloud servers, based on the EEG readings collected by the sensors.

- A Wearable Cognitive Assistance system based on Google Glass devices that assist people with reduced mental acuity.
- Because of the nature of cognitive devices with constrained resources, the compute-intensive workloads of this application need to be offloaded to an external server.
- However, this offloading must provide crisp, real-time responses; failing to do so would be detrimental to the user experience.
- Offloading the compute-intensive tasks to the cloud incurs a considerable latency, thus the authors make use of nearby devices.

- These devices may communicate with the cloud for delay-tolerant jobs like error reporting and logging.
- The aforementioned works are typical applications of fog computing, in that they perform latency-critical analysis at the very edge and latency-tolerant computation at the cloud—thus portraying fog as an extension of cloud.

3. CACHING AND PREPROCESSING

- Users connect to the Internet through fog boxes, hence each HTTP request made by a user goes through a fog device.
- The fog device performs a number of optimizations that reduces the amount of time the user has to wait for the requested webpage to load.
- Apart from generic optimizations like caching HTML components, reorganizing webpage composition, and reducing the size of web objects, edge devices also perform optimizations that take user behavior and network conditions into account.
- For example, in case of network congestion, the edge device may provide low resolution graphics to the user in order to reach acceptable response times.
- Furthermore, the edge device can also monitor the performance of the client machines, and, depending on the browser rendering times, send graphics of an appropriate resolution.

- One of the major advantages of fog computing is linking IoT and cloud computing.
- This integration is not trivial and involves several challenges. One of the most important challenges is data trimming.
- This trimming or pre-processing of data before sending it to the cloud will be a necessity in IoT environments because of the huge amount of data generated by these environments.
- Sending huge volumes of raw data to the cloud will lead to both core-network and data-center congestion.

- Data generated by IoT devices is sent to the smart gateway, either directly (one-hop) or through sink nodes (multi-hop).
- The smart gateway handles the pre-processing required before sending the data to the cloud.
- In the architecture proposed by the authors, the smart gateway is assisted by fog-computing services for operations on IoT data in a latency-sensitive and context-aware manner.
- Such a communication approach paves the way for the creation of a richer and better user experience for IoT applications.