Fog Use Case Scenarios

Use Case: Robots Simultaneous Localization And Mapping (SLAM)
Vertical: Smart Robotics

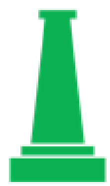
An OpenFog Consortium Architectural Use Case

1 Snapshot: Fog-Enabled Smart Robots

WHY FOG

Why is fog the best architecture for this use case?

By leveraging key principles of fog computing that enable processing to take place in close proximity to the robots, robots simultaneous localization and mapping (SLAM) is enabled by high-performance real-time edge processing, optimized analytics, and heterogeneous applications.



WHICH FOG PILLAR

Which fog pillar best describes this use case?

While all 8 pillars are represented in the robots SLAM scenario, the Scalability pillar enables the most significant advancements in performance and efficiency. For example, the administrator can distribute tasks for compute, storage, and communication to the robot itself, to fog nodes, or to the cloud, depending on QoS or latency requirements.

VALUE

What are the business advantages of building this use case with fog?

The SLAM use case speeds up the time to process vast amounts of data required in life-or-death situations such as firefighting or rescue operations. Tasks can be completed in real-time despite the limited onboard processing capabilities of robots SLAM. It also adds efficient, cost-effective technical processes to applications such as surveillance and scientific exploration.

CLOUD & EDGE

How does this use case augment or supersede cloud and edge architectures?

Years ago, the cloud enabled multi-robot systems to share information and learn from each other. Those same systems can now benefit from the fog architecture, which provides real-time, close proximity processing services for which the cloud is too remote.

2 Table of Contents

3 Introduction

Note: The preamble section of this document (pages 3 through 11) is common across all OpenFog use cases. It provides descriptions and reference points for fog architectural attributes and properties. The Smart Robot use case begins on page 11.

The OpenFog Consortium is defining applications and architectures for fog computing. The Consortium defines fog computing as: ***A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum***.

The first step in this architectural process is understanding the spectrum of vertical markets and applications that we expect fog computing technologies may serve. This document focuses on a representative use case that we believe spans many aspects of fog computing and therefore serves to define the functions we hope fog architecture, fog implementations, and fog deployments will provide.

It is important to understand how this use case fits into the overall process the Consortium uses to define interoperable and certifiable architectures. As shown in Figure 1, the use case described in detail in this document is a starting point for the suite of OpenFog technical documentation. When taken together, OpenFog use cases cover the basic fog functions of approximately 80% of the comprehensive set of IoT network applications we have identified for fog.

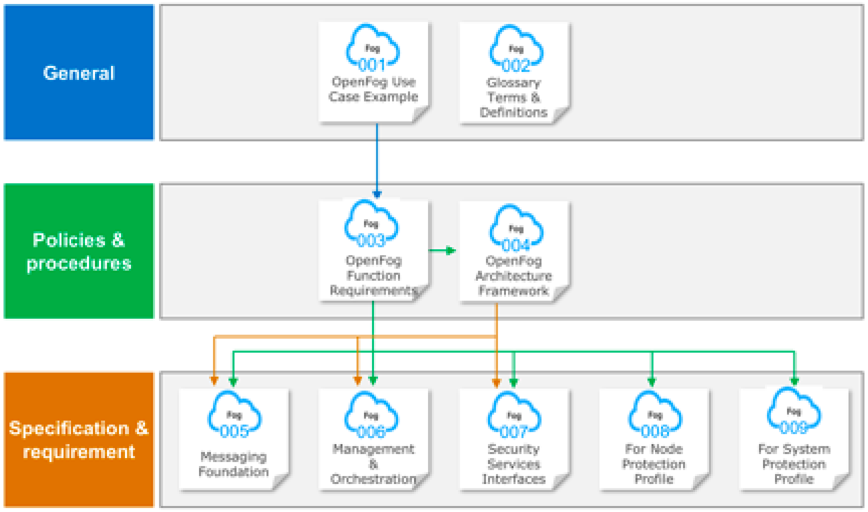


Figure 1. Hierarchy of OpenFog Consortium specification documentation

The composite of all use cases outlines a problem statement for OpenFog, describing the essential functions for all fog elements and networks. The Consortium extracts requirements from these use cases, and distills and correlates them to produce a detailed *Fog Requirements Document*. These requirements serve three important purposes:

1. To drive the OpenFog Reference Architecture;
2. To guide the development of OpenFog testbeds for testing and validation purposes; and
3. To provide guidance to implementers of fog nodes and networks.

The *Architecture Framework Document* is a compendium document that describes the key functional components of OpenFog as well as the interfaces between these components.

The Consortium also publishes additional documents, which describe details in areas such as security, management and orchestration, and messaging. Implementers may use the compendium as a guide for the conceptual planning and architecture design for their fog-based systems, and as implementation best practices for OpenFog elements and networks that will interoperate and can be certified as OpenFog compliant.

OpenFog Consortium workgroups reviewed and discussed hundreds of potential fog use cases spanning more than a dozen vertical markets related to IoT. The Consortium carefully selected a set of use cases that we believe spans a representative set of potential fog applications.

These use cases will highlight one or more representative attributes of fog such as latency, network bandwidth, reliability, security, programmability, scalability. The derived requirements from the use cases we include will cover an illustrative sample.

As mentioned, OpenFog technical requirements comprise a platform that covers approximately 80% of common fog functions. The remaining 20% of requirements needed to support specific use cases which are application dependent and won't be defined by the Consortium.

Readers should pay detailed attention to the subset of use cases that most closely match their areas of interest. We encourage you to browse additional use cases, as they may highlight less obvious aspects of fog that could prove valuable, and give insight into the rationale of the OpenFog requirements.

Readers are also encouraged to collect additional use cases and submit them to OpenFog for requirements extraction and potential inclusion in future use case documents.

4 Fog Computing Overview

Fog computing provides the missing link in the cloud-to-thing continuum. It is a critical architecture for today's connected world as it enables low latency, reliable operation, and removes the requirement for persistent cloud connectivity to address emerging use cases in Internet of Things (IoT), 5G, Artificial Intelligence (AI), Virtual Reality and Tactile Internet applications.

Fog architectures selectively move compute, storage, communication, control, and decision making closer to the network edge where data is being generated and used. This solves the limitations in current infrastructure to enable mission-critical, data-dense use cases.

Fog computing is an extension of the traditional cloud-based computing model where implementations of the architecture reside in multiple layers of a network's hierarchy. These extensions to the fog architecture may retain all the benefits of cloud computing, such as containerization, virtualization, orchestration, manageability, and efficiency.

The fog computing model provides the ability to move computation and storage from the cloud closer the edge, based on the needs of the data and the service requirements. These functions can potentially reside right next to the IoT sensors and actuators. The computational, networking, storage and acceleration elements of this new model are known as fog nodes. These nodes may also reside in the cloud, as they comprise a fluid system of connectivity and don't have to be fixed to the physical edge.

5 The OpenFog Reference Architecture

The OpenFog Consortium was founded on the principle that an open and interoperable fog computing architecture is necessary in today's increasingly connected world. Through an independently-run open membership ecosystem of industry, end users and universities, we can apply a broad coalition of knowledge to these technical and market challenges. We believe that proprietary or single vendor fog solutions are of limited value, as they can limit supplier diversity and ecosystems, resulting in a detrimental impact on market adoption, system efficiency, quality and innovation.

The OpenFog Reference Architecture (RA) is a medium- to high-level view of system architectures for fog nodes and networks. It is the result of a broad collaborative effort of the OpenFog ecosystem of industry, technology and university/research leaders. It was created to help business leaders, software developers, silicon architects and system designers create and maintain the hardware, software and system elements necessary for fog computing, as well as design, architect and develop solutions that enable fog-cloud, fog-thing and fog-fog interfaces.

6 Benefits of Fog

Fog computing targets cross-cutting concerns such as the control of performance, latency and efficiency, which are also key to the success of fog networks. Cloud and fog computing are on path to a mutually beneficial, inter-dependent continuum.

Certain functions are naturally more advantageous to carry out in fog nodes, while others are better suited to cloud. The traditional backend cloud will continue to remain an important part of computing systems as fog computing emerges. The segmentation of what tasks and single purpose functions go to fog and what goes to the backend cloud, are application and implementation/use case specific.

This segmentation can be planned and static, but can also change dynamically if the network state changes in areas such as processor loads, link bandwidths, storage capacities, fault events, security threats, energy availability, cost targets, and so on.

The OpenFog RA enables fog-cloud and fog-fog interfaces. OpenFog architectures offer several unique advantages over other approaches, which we term SCALE:

- **S**ecurity: Additional security to ensure safe, trusted transactions
- **C**ognition: Awareness of client-centric objectives to enable autonomy
- **A**gility: Rapid innovation and affordable scaling under a common infrastructure
- **L**atency: Real-time processing and cyber-physical system control
- **E**fficiency: Dynamic pooling of local unused resources from participating end-user devices

To illustrate this concept, let's look at a quick use case example: Consider an oil pipeline with pressure and flow sensors and control valves. One could transport all those sensor readings to the cloud (perhaps using expensive satellite links) to analyze the readings in cloud servers to detect abnormal conditions, and send commands back to adjust the positon of the valves.

There are several problems with this scenario: The bandwidth to transport the sensor and actuator data to and from the cloud could cost many thousands of dollars per month; those connections could be susceptible to hackers; it may take several hundred milliseconds to react to an abnormal sensor reading (during which time a major leak could spill significant oil); and if the connection to the cloud is down, or the cloud is overloaded, control is delayed or, in the worst case, completely lost.

Now, consider placing a hierarchy of local fog nodes near the pipeline. They can connect to sensors and actuators with inexpensive local networking facilities. These fog nodes immediately establish a community which provides the ability to collaborate. They can be highly secure, lessening the hacker threat. Fog nodes can also be given the authority to react to abnormal conditions in milliseconds, quickly closing valves to greatly reduce the severity of spills.

Local control in the fog nodes produces a more robust control system. Moving most of the decision-making functions of this control system to the fog – and only contacting the cloud occasionally to report status or receive commands – creates a superior control system.

Fog computing includes a set of high-level attributes of fog computing that we call the pillars; these include some of the fog advantages described in the pipeline control scenario. There are 8 pillars in total: security, scalability, openness, autonomy, reliability, agility, hierarchical organization and programmability. We will discuss all of these pillars in detail later in this document.

The OpenFog RA defines the required infrastructure to enable building Fog as a Service (FaaS) to address certain classes of business challenges. FaaS includes Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and many service constructs specific to fog. The infrastructure and architecture building blocks below illustrate how FaaS may be enabled; this will be expanded upon in the reference architecture document.

The OpenFog RA describes a generic fog platform that is designed to be applicable to any vertical market or application. This architecture is applicable across many different markets including, but not limited to, transportation, agriculture, smart cities, smart buildings, healthcare, hospitality, financial services, and more, providing business value for IoT, 5G and AI applications that require real-time decision making, low latency, improved security, privacy protection and are network-constrained.

7 Use Case Scenario: Smart Robots

Use Case: Robots Simultaneous Localization And Mapping (SLAM)

Vertical: Smart Robotics

Executive Summary

Robotics systems and their missions are becoming more complex. As a result, it is important for robotics applications and modules to work in parallel and share information. This helps them achieve difficult tasks more efficiently.

One such task is SLAM - Simultaneous Localization And Mapping. In SLAM, a robot is placed in a previously unknown environment and builds a map of the environment. It also simultanously locates itself within the map.

SLAM is useful in a variety of mission-based applications, including rescue operations, fire fighting, underwater exploration, space exploration and surveillance. The SLAM task is challenging because it involves processing large amounts of data from different heterogenous sensors such as cameras, inertial measurement units (IMUs), and laser rangers. Moreover, particularly in life-or-death applications, the task must be completed in real-time despite the limited onboard processing capabilities of SLAM robots.

In recent years, in an effort to overcome these challenges, we have seen the introduction of distributed robot systems, cloud-based robot systems and other autonomous machines. In this architectural use case, we propose a fog-based robotic solution for these advanced SLAM systems. By leveraging key principles of fog computing, robots SLAM will be enabled by high-performance real-time edge processing, optimized analytics, and heterogeneous applications. With fog, each of these can deployed in close proximity to a multi-robot system. As shown in Figure 2, the fog solution for SLAM can be conceptualized as a three-layer hierarchy:

1. The lower layer is the device layer, which is composed of robots.

2. The middle layer is fog-net layer, which is composed of distributed fog nodes. These fog nodes reside close to the robots to provide a real-time messaged processing mechanism for robot applications, to gather data from the device layer up to the cloud layer, and to apply a set of publish-subscribe brokers that relay information to cloud services.

3. The fog-net layer constructs a distributed stream processing framework (DSPF). When coupled with batch processing engines in the cloud or fog node layer, the data can be processed in the cloud and the information can be returned to the robots. Applications execute data analytics by DSPF, enabling the framework to process streaming data in real-time.

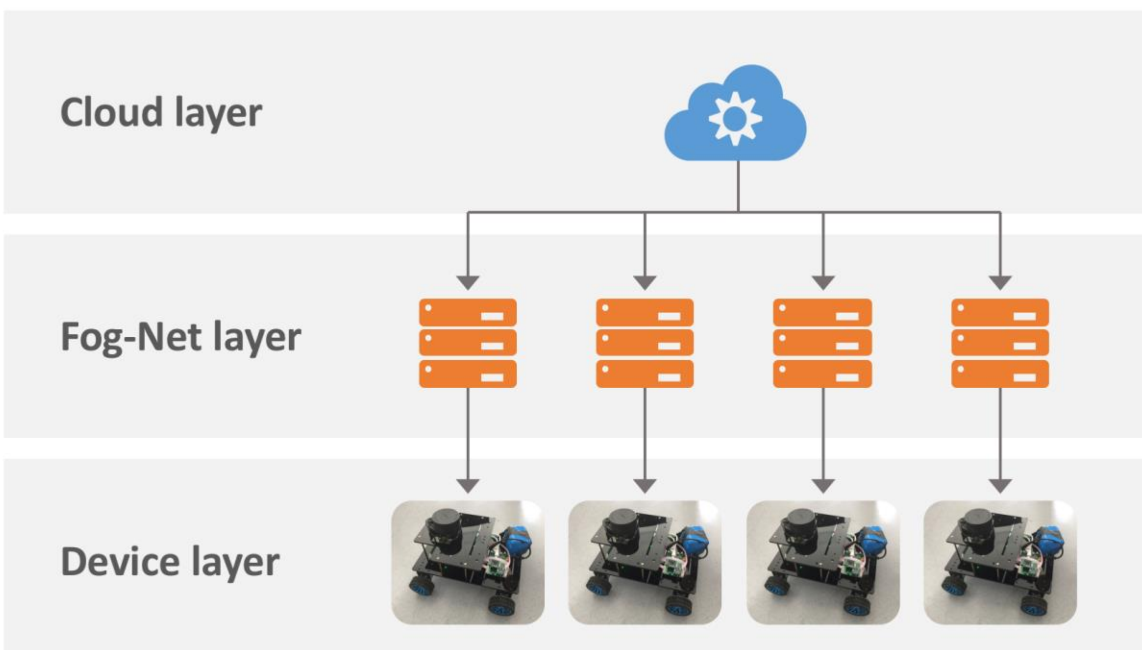


Figure 2. Hierarchy of the fog-based robot use case.

Challenges	• Robots SLAM applications are typically data intensive and involve processing of large amounts of data from different heterogenous sensors such as cameras, inertial measurement units (IMUs), and laser rangers. • Many robots SLAM applications are mission critical situations, such as firefighting and rescue, which require real-time data processing.
Solution	• A three-layer fog architecture built around fog nodes that are in close proximity to the robots. • Fog nodes provide a real-time messaged processing mechanism that gather data from the device layer, and applies a set of publish-subscribe brokers that relay information to cloud services. • SLAM data is processed in the fog-net and the information can be returned to the robots. • The cloud-layer merges and storage maps, and enables overall optimized robots locating and route planning.
Technology	• By leveraging key principles of fog computing, robots SLAM is enabled by high-performance real-time edge processing, optimized analytics, and heterogeneous applications. • The fog-net layer constructs a distributed stream processing framework (DSPF) that works with batch processing engines in the fog-net layer. • Applications execute data analytics by DSPF, enabling the framework to process streaming data in real-time.

Introduction

With the continued development of industrial robots, programmed robots have reached advanced levels of performance in real-time applications. There have been significant breakthroughs in accuracy, robustness and compatibility. However, in many real-world cases, pre-programmed robots cannot meet application requirements. For example, when facing extreme environmental situtations, such as earthquakes, exploring unknown space or unfamiliar conditions, robots struggle to operate fast, react quickly, or accurately grasp the problem at hand.

As network technology evolved during the latter part of the 1990s, researchers developed and improved the control of robotic network interfaces and their robustness, and the field of networked robotics emerged. A robotic network refers to a group of robots connected through a wired or wireless communication network. An individual robot in networked robotics is called a "node." With sensing data and information shared among nodes, operators can transmit command data remotely and receive measurement feedback, ensuring that a specific operation is carried out.

The development of networked robotics has enabled robots to be utilized in a variety of useful applications, such as long-distance medical surgery, disaster relief and other specialized cases. However, like the single robot, robotics networks also face some inherent physical limitations.

For example, due to the limitations of the robot's size and related factors, there are obvious limitations to the computing and storage capacity of individual robots. This in turn leads to a limited capacity of networked robotics, particularly when facing highly-complex processing tasks. The corresponding performance improvement of individual robots brings other limitations as well.

With the development of cloud computing, big data and other leading-edge technologies, the integration of cloud technology and multi-robot systems now allows for the design of multi-robot systems with high performance and high complexity. Today we have the capability for cloud-based multi-robot systems to allow robots to share information, learn from each other and update the database to achieve a virtuous closed loop.

This has now paved the way for SLAM. In 2015, Mohanarajah et al. presented a cloud-based collaborative visual SLAM system consisting of low-cost robots and remote Amazon servers [1]. The SLAM robots offered real-time map estimation through parallel computing. They implemented a highly sophisticated system using low-cost components, which lowered the technological threshold for further research.

Additionally, Aguero et al., through their participation at an open competition, presented a software framework for cloud-hosted robot simulation aimed at advancing the technology behind robotic systems that deal with disaster response[2].

With the introduction of cloud technologies, the availability of computation distributions and communication modes increased. These elements can be applied in different SLAM scenarios and in fact are critical for meeting performance requirements. However, this increased capability brings a new set of challenges.

For example, extracting relevant patterns from data in the cloud is a typical requirement in many SLAM applications. This poses the challenge of data format conversion when dealing with data uploading and downloading.

Another challenge is cloud security, especially the storage of important data in the cloud, which increases the requirements on various aspects of the robotic systems.

Finally, it is challenging to ensure real-time performance, to choose the service quality guarantee methods, and to determine the corresponding effect analyses. These are areas where fog computing can bridge the gap between device and cloud and yield high-performance systems that meet advanced requirements.

The SLAM Process

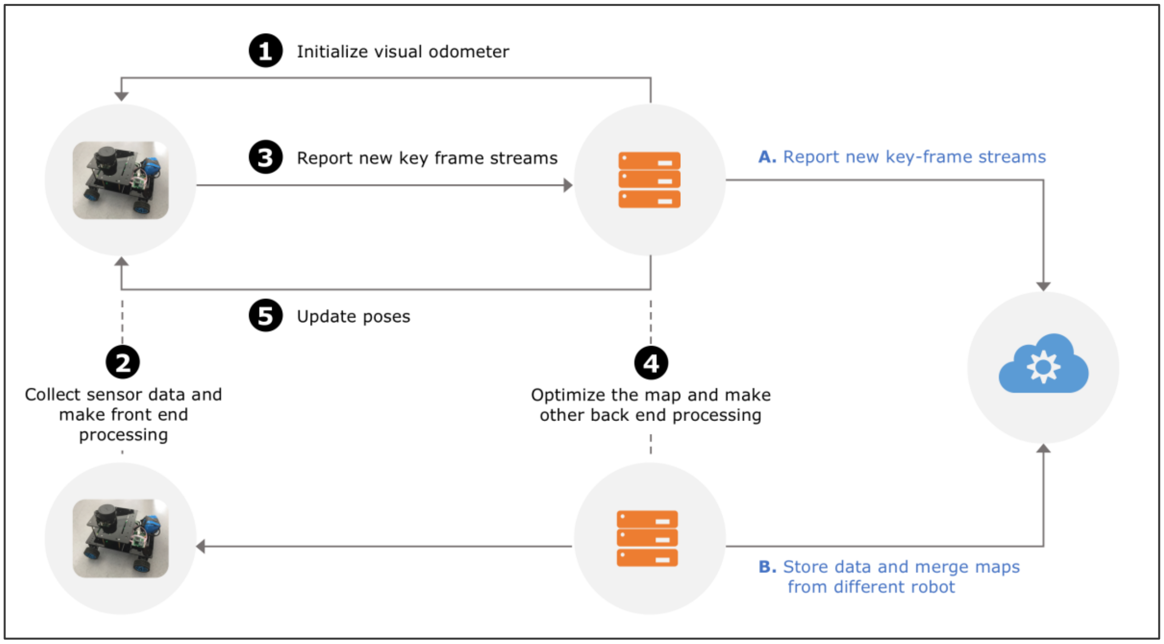


Figure 3. Fog-based SLAM.

To understand how fog can impact SLAM operation and performance, let's first dissect the SLAM processes. As shown in Figure 3, the SLAM processes includes sensor data acquisition, tracking, local mapping, loop closing, nonlinear optimization, and mapping.

The computational complexity of modules such as nonlinear optimization is very large, and the robot's local processing is often limited by the small footprint and limited computational power of the robot. With the help of fog's compute, storage and transmission capabilities, large, computationally intensive and storage demanding tasks can be efficiently uploaded to the cloud, improving mapping accuracy and speed. This capability also enables the adoption of concurrent, low-cost multi-robot operation, which further improves the speed of map building. A map constructed by a single robot can also be merged in the cloud, to build a larger map quickly.

A fog-based SLAM process is shown in Figure 3. The front-end processing of SLAM is, as before, deployed on robots that are in charge of tasks such as collecting sensor data and visual odometry. Fog nodes are in charge of back-end processing, including map optimization and loop closing.

When fog nodes receive new key frames from robots via wireless communication, they add key frames to local databases and transfer them to to the cloud. They can then optimize the map and set updated poses for the robots. Transmitting new key frames and updated poses between robots and fog nodes requires minimal bandwidth resources. The map merging task can be deployed in the cloud. The cloud can receive all the new key frames from all the fog nodes, and the fog nodes don't need to exchange data with each other.

In short, with the help of fog nodes, SLAM robots can efficiently leverage cloud computing, storage and communication capabilities. Multiple robots can collaborate to generate maps and obtain accurate pose information in real-time. In this way, fog enables collaborative SLAM robot application scenarios that support more complex and diverse applications.

Key Technical Challenges

The key issues and challenges facing the implantation and deployment of fog-based SLAM robots are as follows:

1. *Resource Allocation & Scheduling.* The ability to upload computational tasks with high complexity to the fog network is one of the most important characteristics of fog-based robotics. Unlike cloud-based solutions, fog-based robotics can finish these tasks in real time. While there are advantages to this, it also opens up new challenges. Considering different working equipment, interface settings, and network environments for a given computational task. The choice of uploading, self-processing or assigning tasks to the nearest fog node has an important impact on overall performance and efficiency.

   As with inter-machine communications, the amount of computations that take place in the fog make the emergence of latencies more likely. New algorithms and techniques are needed to counter changes in network delays that might impact real-time processing, such as communicating over 5G, or with New Radio's flexible short frame structure in uRLLC (Ultra Reliable &

Low Latency Communication).

Besides real time communication technology, a mechanism for dynamic allocation of computing tasks should be activated, thereby reducing the delay time. Considering the large size of a data stream in navigation and other SLAM applications, the most critical aspect is the computation-communication tradeoff. In any case, carefully chosen computation offloading schemes will usually outperform local computational capabilities. These techniques can only be accomplished in fog-based networks due to their close physical proximity to the robots, which is not possible with cloud.

2. *Data Interaction Between Robot and Fog Platform.* Devices and sensors from different manufacturers may output data with different structures. In fact, even different models of a product from the same manufacturer may result in considerable differences in the output data's structure. The diversity of data structures puts a strain on the compatibility requirement for the cloud input interface.

   To solve this problem, current mainstream cloud platforms often provide multiple interfaces for various types of data formats. However, due to the limited number of interfaces, the data to be uploaded must be properly preprocessed. The robustness and real-time performance of the data exchange is greatly affected by data formats requiring translation from one form to another. Fortunately, this processing can be handled by the fog-based solution because of the data's local processing advantage.

3. *Security.* The introduction of cloud technology has greatly expanded the potential of multi-robot operations. At the same time, it also introduced complex new technical challenges for privacy and security. These hidden dangers can also affect data generated by computing devices and sensors used in cloud-based robotics.

   Commercial science and technology solutions have suffered from serious data leakage incidents, especially during the upload of photos and video to the cloud. In scientific research and industrial practice, there are vulnerabilities in transmitting and storing key data in the cloud. In fog-based solutions, large amounts of data is processed directly in the lower network layer, which can decrease the risk of being hacked.

4. *Service Quality Guarantee Methods and Effects Analysis.* In order to meet the needs of users with different service quality requirements, the network is expected to provide different levels of quality. Real-time requirements require high priority data processing, while tasks with less than real-time requirements are granted lower priority.

   Using a correspondingly differentiated service model similar to that for QoS, higher bandwidth priority is given to tasks with higher real-time requirements, according to the characteristics of the SLAM system.

   The fog-based robot solution can provide smart resource allocation, task scheduling and quick response. Using message brokers and streaming workflow services, the processing tasks at the fog nodes can provide services in parallel with the units of message and stream. DSPF deployed on the fog network can process mass data at the network edge and doesn't need to transmit to cloud end. In this scenario, the cloud layer is responsible for processing the huge compute and non real-time tasks that require only low data transmission rates.

Architectural Considerations

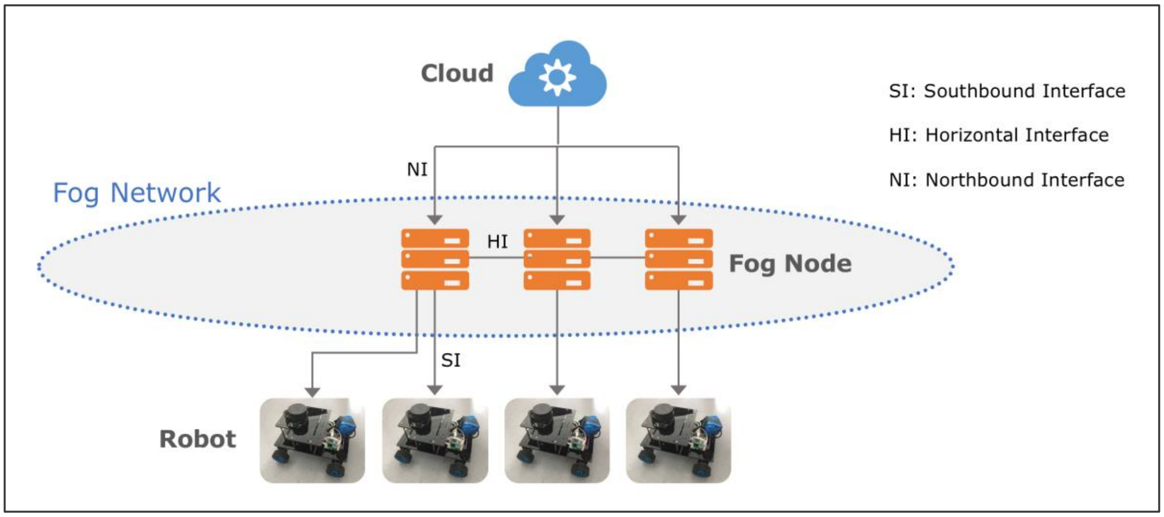


Figure 4. Functional view of a fog-based robot solution.

As shown in Figure 4, the fog-based robot solution includes three main network entities: robots, fog nodes, and cloud. Robots are connected to fog nodes via wireless interface such as 4G, 5G, or WiFi. The functional interface between robots and fog nodes is called the southbound interface. This defines information of the data panel and control panel. The data panel carries key frames and presents information which is processed by robots or fog nodes.

The control panel carries communications which manage all work flows such as initializing, stopping, getting key frames and setting robot poses. Fog nodes can form a network which might be a star network, bus network, or other. Fog nodes can communicate with each other via a horizontal interface. Multiple fog nodes can collaborate and realize CoMP (Coordinated Multiple Points), distributed storage system, or distributed computing systems, enabling them to access more powerful capabilities, such as communication, storage and computing. The horizontal interface carries the control and data information, which supports coordinating functions.

Because fog nodes and fog networks are much closer to the robots, the latency of robot processing can be reduced to 10 millisecond or less. Fog nodes can communicate with the cloud via a northbound interface. Tasks that are not time sensitive, such as huge compute or storage capabilities, can be deployed on the cloud.

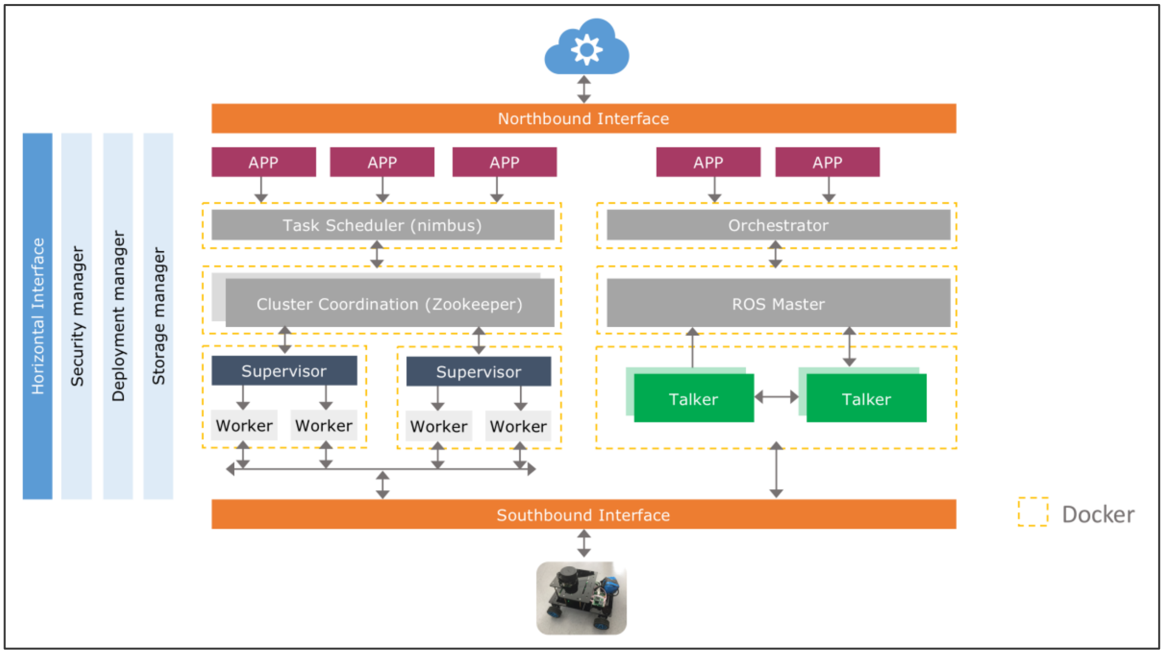


Figure 5. Functional view of the fog-based robot use case.

As shown in Figure 5, the functional modules of the fog-based robot solution can be divided into task process, communications, storage, deployment, security, and interfaces. When faced with a low-latency requirement and stream-oriented processing, we can apply a real-time streaming processing framework such as Storm or Spark. With Storm for example, the task process could be composed of application, task scheduler, cluster coordination, supervisor and worker.

To be compatible with legacy mechanisms, we can adopt ROS using the deployment manager. The storage manager can support distributed storage systems and centralized databases. Distributed storage systems are suitable for delay-insensitive applications that demand large storage. Centralized databases are suitable delay-sensitive systems. Using the deployment manager, we can select different storage schemes. The security manager can manage different security functions such as authentication, encryption, privacy protection, and so on.

The southbound interface provides information exchange of control panel, data panel and management panel between fog nodes and robots. The horizontal interface provides information exchange between fog nodes to support fog-net-based collaborative functions. The northbound interface provides information exchange between fog nodes and the cloud to support delay-insensitive, compute- intensive and huge storage tasks.

The performance targets depend on application requirement as follows:

- A 1-millisecond level application can operate directly by the robot.
- A 10-millisecond level application can be uploaded to fog nodes.
- Applications greater than 100-milliseconds can be uploaded to the cloud.
- Real-time data that is normally less than 1GB can be stored in the robot.
- Semi-real-time data that is normally more than 10GB can be stored in fog nodes.
- Non real-time mass data can be stored in the cloud for big data post analysis and processing.

Advantages of the Fog Computing Approach

The fog computing approach greatly reduces the latency, network bandwidth and availability constraints of robot SLAM applications. The architecture for fog computing is based on eight pillars, as identified by the OpenFog Consortium. In the robot SLAM use case, the following pillars are addressed:

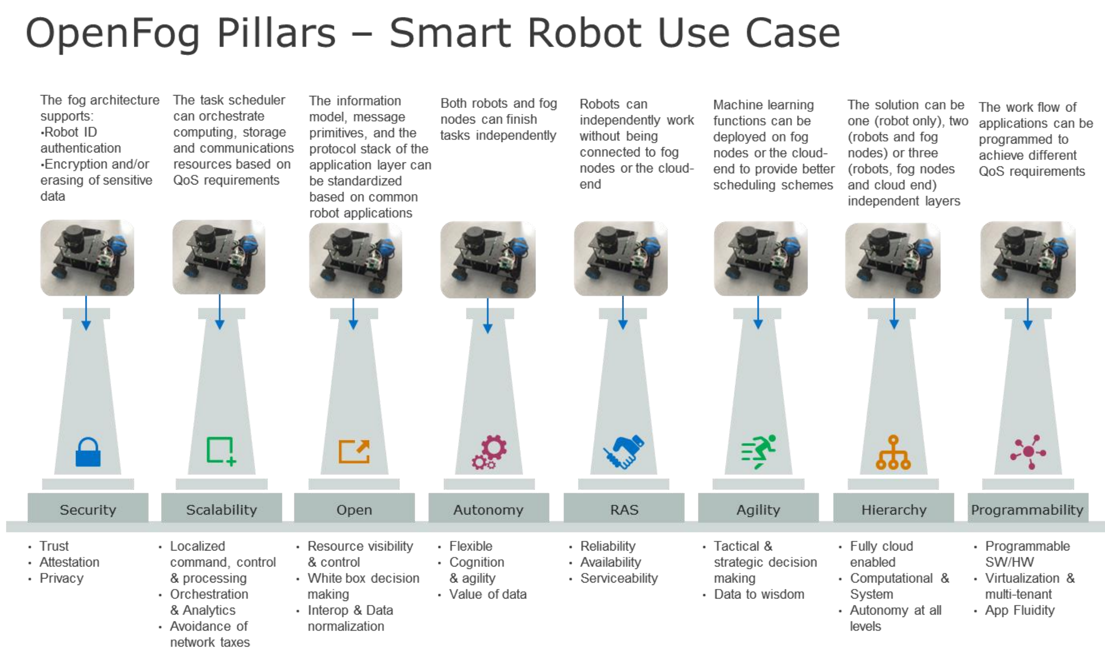


Figure 6. The OpenFog Reference Architecture pillars mapped to the Smart Robot Use Case. The table below shows the function-to-pillar mapping descriptions.

Pillar	Functions
Security	• Robots ID authentication is supported on the SI. • Data information encryption is supported on the SI, NI, and HI. • Sensitive information included in key frames and map data is encrypted or erased before it is stored into databases.
Scalability	• The task scheduler can orchestrate computing, storage, and communications resources based on QoS requirements or the status of robots, fog nodes and cloud. • For delay-sensitive tasks, the task scheduler can assign them to either robots or neighboring fog nodes. For delay insensitive tasks, the task scheduler can assign them to the cloud. • The task scheduler considers compute-communication tradeoffs when making decisions. • Fog nodes can form a collaborative network to support more powerful communication, computing and storage systems.
Open	• The information model, message primitives, and protocol stack of application layer on SI, NI and HI can be standardized based on the common robot applications. • The policies of task scheduling, resources orchestration and modules deployment can be open and defined by users.

	• Any kind of resource can be visualized by the application interface.
Autonomy	• The network topology is flexible to support different application scenarios, such as standalone robots, networks of robot and fog nodes, or networks of robots, fog nodes and cloud. • Both robots and fog nodes can finish tasks independently.
RAS	• Robots can independently work without being connected to fog nodes or the cloud. • Fog nodes can finish work without being connected to the cloud. • Fog nodes can organize a collaborative network to support much more reliable communications, more powerful communications, andmassive storage.
Agility	• Machine learning functions can be deployed on fog nodes or the cloud to provide better scheduling schemes. • Robots can gain more experience and knowledge from big history databases located in fog nodes or the cloud.
Hierarchy	• The solution can be one (robots only), two (robots and fog nodes) or three (robots, fog nodes and cloud end) layers. • Each level can work independently. • Fog nodes can provide distributed computing and storage solutions for higher QoS applications.
Programmability	• The workflow of applications can be programmed to achieve different QoS requirements. • The policies of task scheduling, resources allocation, and information processing mechanisms can be programmed by users.

Communications Considerations

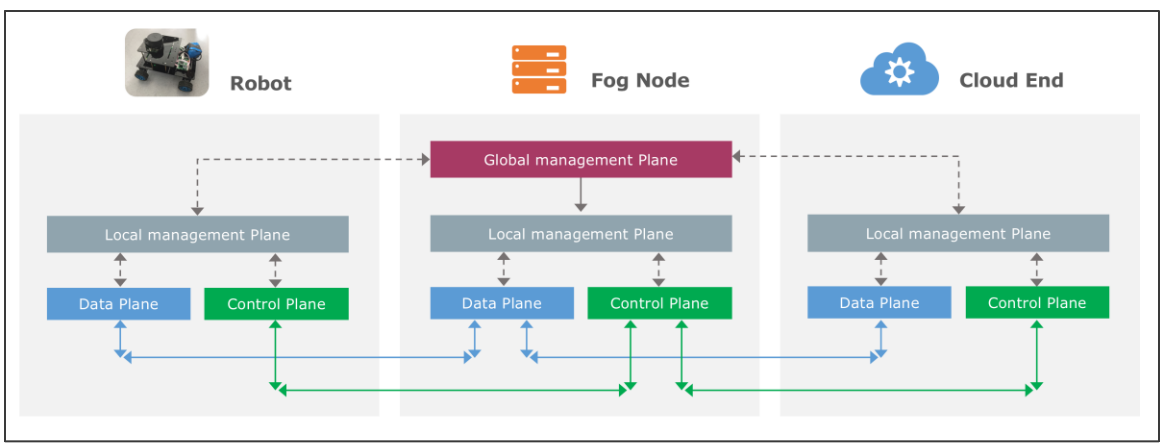


Figure 7. Diagram of Data, Control and Management Plane deployment scheme.

One panel deployment scheme is shown in Figure 7. Considering requirements for robot autonomy and low-delay closed loops, fog can provide a flexible plane deployment scheme. Robots can locally management, data, and control plane even without connecting to fog and cloud. Because fog nodes are closer to robots than the cloud, they can have a global management plane that defines better policy by making global computing-communication tradeoff decisions.

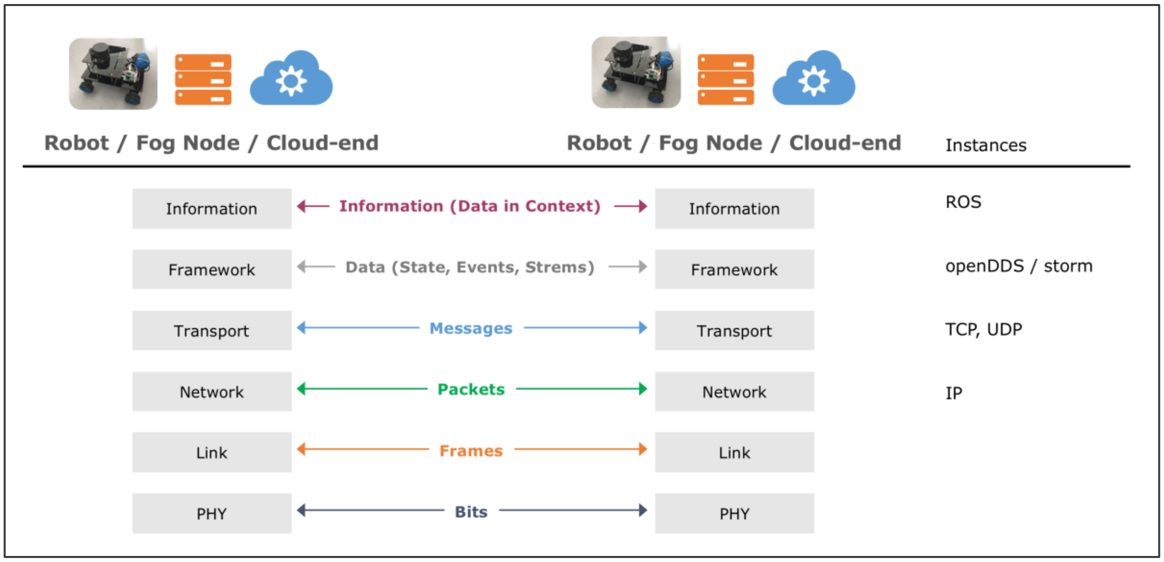


Figure 10. The information model and communication protocol stack.

As shown in the information model in Figure 10, the communication protocol stack can be divided into six layers. In theory, one robot can communicate with the other robots directly, but here, we emphasize the topology of robot-fog node, fog node-cloud end, robot-fog node-cloud combinations.

Because applications or tasks can be deployed flexibly on these kinds of topologies, robots, fog nodes and cloud can be regarded as peer-to-peer from the point of view of the information model. The communication standard for the lower four layers will evolve to 5G for high data rate and low latency. At the present stage, we can utilize 4G or WiFi.

The entities bearing scheduling tasks and resource allocation are fog nodes to meet the requirements of easily collecting information and quick response. Fog nodes should also be assingned the responsibility of robot management. Management of fog nodes can be deployed on one fog node or on the cloud, according to user-defined policy.

The workflow of robot applications can be programmed by scripts to achieve different performance targets. Thus the control signaling for the applications can be programmed to meet quickly changing user requirements.

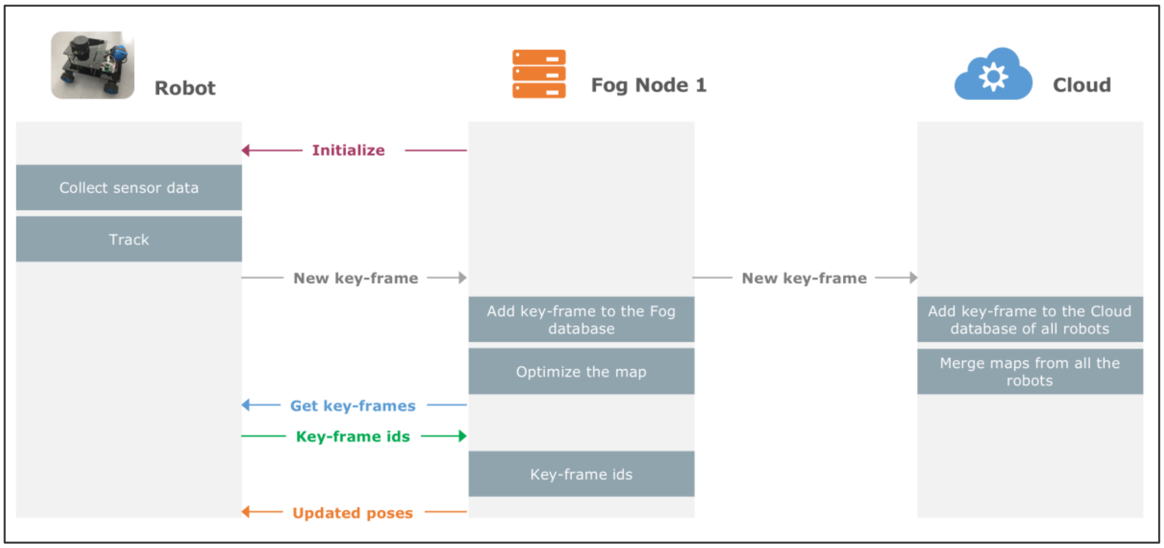


Figure 11. Message interaction flow of SLAM within the three-layer architecture.

The message interaction flow for the SLAM workflow is shown in Figure 11. The information in data plane forms real-time stream flow. Administrators have the flexiblility to adopt real-time stream processing framework such as Storm and Spark.

<u>Fast Track to OpenFog Testbeds</u>

There is a large, growing ecosystem of robot manufacturers and suppliers. OpenFog members are exploring a testbed comprised of robots, fog nodes and the cloud. The testbed platform will be staged in three steps:

1. The first step is to construct the basic data, control and management framework with a mature technical framework

such as ROS, Storm, Spark, LTE, and WiFi. This will enable us to run the typical robotic workflow, such as SLAM.

2. The second step is to accommodate the collaborative enhancements for SLAM, such as multi-robot map merging.

3. The third step is to build out the testbed architecture while complying with the best practices defined in the OpenFog Reference Architecture.

The OpenFog Consortium will create testbeds for this ecosystem where the common interfaces developed for interoperability in fog will be standardized and certified. These unique spaces will foster cooperation among suppliers, and enable amazing applications to be developed using an architecture designed for low-latency, high-bandwidth, heterogeneous operations.

The hierarchy of OpenFog testbeds will be structured as follows:

1. Many small, research-oriented locations that OpenFog members are able to access will focus on proving the high-level OpenFog architectural requirements and satisfying the minimum interoperability requirements via their Proof-Of-Technology (POT) Testbeds. The outcome of these Proof-Of-Technology testbeds could be open source code or a research publication available to OpenFog members.

2. Medium-sized, Interoperability Operation Model (IOM) testbeds will focus on overall solutions and end-to-end applications, with at least three OpenFog sponsoring members participating to promote usage of diverse OpenFog Ready Solutions. They will demonstrate adherence to the OpenFog Reference Architecture and component-level interoperability and compatibility.

3. Large, regional testbeds will test pre-productization devices for application to the co-located OpenFog Certification Lab that will be developed in the future. After the OpenFog Certification Lab

validates a product, members will be able to release it as an OpenFog Certified product. We expect many verticals, use cases, and individual applications will have specific requirements for interoperability and preferences for certain types of testbeds, and the Consortium intends to adapt to their needs.

8 Adherence to the OpenFog Reference Architecture

The OpenFog Consortium is partnering with standards development organizations, providing detailed requirements to facilitate a deeper level of interoperability. This will take time, as establishing new standards is a lengthy process. Prior to finalization of these detailed standards, the Consortium is laying the groundwork for component level interoperability and certification. Testbeds will prove the validity of the OpenFog Reference Architecture (RA) through adherence to the architectural principles.

9 Next Steps

The OpenFog Reference Architecture (RA) is the first step in creating industry standards for fog computing. It represents an industry commitment toward cooperative, open and interoperable fog systems to accelerate advanced deployments in smart cities, smart energy, smart transportation, smart healthcare, smart manufacturing and more. Its eight pillars imply requirements to every part of the fog supply chain: component manufacturers, system vendors, software providers, application developers.

Looking forward, the OpenFog Consortium will publish additional details and guidance on this architecture, specify APIs for key interfaces, and work with standards organizations such as IEEE on recommended standards. The OpenFog technical community is working on a suite of follow-on specifications, testbeds which prove the architecture, lists of requirements, and new use cases to enable component-level interoperability. Eventually, this work will lead to certification of interoperable elements and systems, based on compliance to the OpenFog RA.

We welcome public commentary on this use case and on other work by the OpenFog Consortium. To comment or for additional information on this material, please contact info@openfogconsortium.org.

10 About the OpenFog Consortium

The OpenFog Consortium was founded to accelerate the adoption of fog computing and address bandwidth, latency and communications challenges associated with IoT, 5G and AI applications. Committed to creating open technologies, its mission is to create and validate a framework for secure and efficient information processing between clouds, endpoints, and services. OpenFog was founded in November 2015 and today represents the leading researchers and innovators in fog computing.

For more information, visit http://www.openfogconsortium.org/; Twitter @openfog; and LinkedIn /company/openfog-consortium.

Reference:

[1] Gajamohan Mohanarajah, et al., 'Cloud-Based Collaborative 3D Mapping in Real-Time With Low-Cost Robots', IEEE TRANS. AUTOMATION SCIENCE AND ENGINEERING, VOL. 12, NO. 2, APRIL 2015, pp 423-430

[2] Carlos E. Agüero, et al., 'Inside the Virtual Robotics Challenge: Simulating Real-Time Robotic Disaster Response', IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 12, NO. 2, APRIL 2015, pp. 494-506

11 Authors and Contributors List

Authors	Contributors
Kai Li, ShanghaiTech University and the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences.	Evan Birkhead, OpenFog Consortium
Ming-Tuo Zhou, ShanghaiTech University and the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences.	
Jian Li, ShanghaiTech University and the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences.	
Haidong Xu, ShanghaiTech University and the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences.	

Note: All publicly available use cases are reviewed and approved by the OpenFog Technical Committee.

12 Copyright / Disclaimer

This reference document is designed to provide a foundation for extracting requirements when developing fog-based architectures. It is a compendium document to the OpenFog Reference Architecture. https://www.openfogconsortium.org/ra/