Fog Use Case Scenarios

Use Case: Live Video Broadcasting
Vertical: Entertainment Industry

An OpenFog Consortium Architectural Use Case

1 Snapshot: Live Video Broadcasting

Today's sporting events need to broadcast live video from all corners of the arena or race course with zero latency. Hierarchical fog nodes shorten video latency and decrease backhaul bandwidth. The OpenFog architecture delivers the agility to manage video services and video algorithms. These algorithms distribute the video process services in different layers from camera to cloud, according to their different performance requirements.

The Hierarchy pillar of the OpenFog Reference Architecture is the most crucial to video streaming. The live video network is divided into several interoperable layers, allowing real-time and non-real-time services to be deployed in different layers. Real-time video services and algorithms are deployed in low fog nodes close to cameras and users; non-real-time services are deployed in upper fog nodes; and video data from cameras are aggregated to high fog nodes or the cloud.

By greatly reducing video latency, the OpenFog architecture provides significant savings on network backhaul bandwidth. This leads to an excellent user experience for onsite attendees and more efficient operations. OpenFog supports many kinds of vertical applications, enabling video data to be shared between the fog and several clouds, or between fog nodes.

By virtue of its support for hierarchical layers, the OpenFog architecture augments edge-only and cloud-only solutions. Fog computing provides a distributed and layered architecture, which connects and interoperates with edge and cloud applications and resources.

2 Table of Contents

3 Introduction

Note: The preamble section of this document (pages 3 through 10) is common to all OpenFog use cases. It provides descriptions and reference points for fog architectural attributes and properties. The Live Video Broadcasting 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.

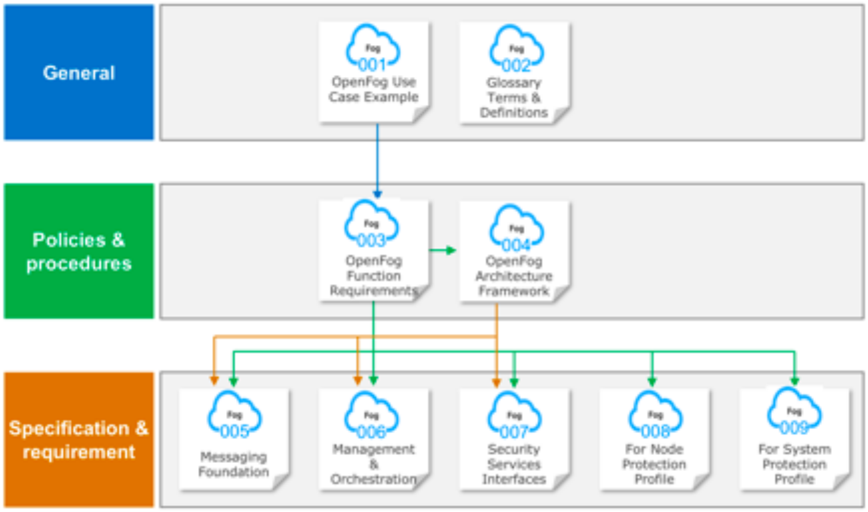

Figure 1. Hierarchy of OpenFog Consortium specification documentation

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: Live Video Broadcasting

Use Case: Live Video Broadcasting

Vertical: Entertainment Industry

Executive Summary

At today's sporting events, fans are demanding high-quality, real-time video on their smartphones and tablets. Events can take place in a stadium, across multiple venues (like the Olympics) or outdoors. Even outdoor events vary—from a fixed track to races that start in one location and end many kilometers - even days - away.

Outdoor races that span miles are particularly challenging for broadcast video. Bicycle racing is a perfect example: the courses can be hundreds of kilometers. Race routes can take athletes through areas that are inaccessible to fans. Radio transmission environments are often challenging.

Throughout these areas, broadcasters use High-Definition (HD) cameras to live stream coverage. These cameras generate massive amounts of video data. For example, one HD camera with a 1080P (2 million pixels) video format and a bit rate of 4MB per second per channel can generate about 1.8GB per hour. In a race such as the Tour de France, there are hundreds of cameras along the course, including mobile camera operators on bikes. All of this data has to be processed and analyzed. With a cloud-to-edge solution, this happens in the cloud. But transmitting massive amounts of video data from the cameras to the cloud requires expensive backhaul bandwidth. It also introduces unacceptable latency and congestion that not only impacts the video quality, but also slows other applications on the network.

Fog computing keeps the video processing local, eliminating latency and congestion issues and eliminating the need to provision expensive backhaul links. With fog, processed and filtered video can be sent to the cloud for long-term storage. In the cloud, video data can also be shared with other vertical applications, such as safeguards. In this bicycle race use case, we'll examine how the OpenFog Reference Architecture provides a layered, hierarchical framework to support high-bandwidth real-time video:

Challenges	• Bike racing fans want to watch races in real time on their smart devices. • HD cameras capture massive amounts of video data that has to be processed, analyzed and broadcast in real time. • Traditional cloud-to-edge solutions puts processes in the cloud, which requires expensive backhaul. • Insufficient bandwidth can cause congestion, which affects the broadcast experience and may affect other applications sharing this bandwidth.
Solution	• Video servers are deployed on fog nodes located close to the cameras and the viewing audience. • Video is transmitted in near real-time from cameras to local fog servers and from fog servers to handheld devices over 4G/5G networks. • Using fog computing with 4G/5G networks saves on provisioning larger backhaul circuits to support unfiltered video traffic. • The fog architecture enables sharing of video data between adjacent fog nodes on the same level, which makes it easier to process, converge and consume video data from many different cameras on the fly. • Video data can also be aggregated at higher-level fog nodes and transmitted to the cloud for storage.

Technology

- Hierarchical fog architecture shares and processes data as close to the source as possible for the greatest efficiency (low latency, bandwidth efficiency, etc.).
- A hierarchical fog network might be set up around the course as follows:
    - Low fog nodes are located near the camera and audience for low latency video processing and transmission.
    - Middle fog nodes aggregate video data from multiple low fog nodes in different locations around the course, and can stitch together video from multiple nearby cameras. They can also perform video analytics, extracting features and tracking objects and riders.
    - High fog nodes filter and further aggregate video data and forward to the cloud for storage or more detailed analysis. They can also compute race statistics.
- Fog computing makes it easy to scale (add, delete, update) fog nodes to support races characterized by different distances, durations and terrains.
- Fog nodes can incorporate all the functionality of a video server, increasing operational efficiency.

Use Case Overview

The following is an example of an implementation of fog computing for live video broadcasting at a sporting event – in this case, for a long-distance bike race. The purpose of presenting this use case is to promote more architectural conversations about fog computing use cases for horizontal video services applications in industries such as entertainment or smart cities.

Business Case

The demand for video services is growing across the board – for business, consumer and in particular, for the entertainment industry. Consumers want to stream the action in real time on their personal devices, as well as view it on large screens throughout the venue. This is already the case in the entertainment and broadcasting industries, especially for indoor sports events in large arenas and outdoor sports events, such as skiing and bike racing competitions.

Bike racing competitions are particularly challenging. They can involve dozens or hundreds of competitors. The events can last hours or days. And the courses can range from dozen to thousands of kilometers through city streets and mountainous terrain.

These events are big revenue generators for sports organizations, advertisers and sponsors, the athletes, the producers and broadcasters and even the municipalities where these events are held. A lot of people have a stake in capturing large audiences for these events. Spectators who attend these events now expect to watch the whole race on their smart devices, with excellent HD / UHD video quality.

To provide this video experience, broadcasters (producers of the video content) must have a way to capture the video data from dozens or hundreds of HD cameras installed around the course. The data from these cameras has to be captured, processed and transmitted in virtually real time. This presents tremendous challenges in computation, storage, and communications.

In addition to using this video collection and analysis for event viewing, the data can also be used for public safety:

- Police can also use the video feeds to monitor these sprawling events without having to physically cover every kilometer with law enforcement or emergency personnel.

- Medical personnel can quickly locate accidents and injured athletes.
- Emergency responders and event officials can communicate safety information to spectators.
- Race officials can use the video to monitor the athletes' compliance with rules and replay any questionable events.

In a cloud-only model, video data is sent to the cloud for processing. This can increase latency, making it difficult for broadcasters to deliver a good video experience to paying viewers. Traditional cloud-based solutions also require a significant investment in backhaul equipment, fiber networks, satellite links, etc.

Fog computing provides the distributed and layered architecture to build real-time video services effectively. A key benefit is that fog nodes can incorporate all the functionality of a video server locally, increasing operational efficiency.

Other services, such as image analysis and security protection, can be deployed on the same platform without creating a new network that has to be managed separately. With this configuration, the same fog network that serves one type of sporting event can support many other events. And a fog network is designed with interoperability as a core architectural principal, making it easy to add new services to the same network. The multi-tenancy property of the OpenFog architecture permits several stakeholders to each run their own software on the same fog nodes (for example, race officials, police, concession vendors and broadcast networks can all use the same fog node processors).

The Fog-Based Configuration

This is the sequence for delivering live video broadcasting covering a long distance bicycle racing:

1. The technology to support live video broadcasting is evolving quickly, and HD cameras are already deployed all along bike courses around the world.
    a. Sometimes multiple cameras are located at strategic locations to capture different angles.
    b. Cameras are often located in areas where wireless coverage can be an issue: narrow city streets, heavily wooded areas, and mountainous terrain.
2. Local fog-based video servers can be positioned along the route to collect and consolidate data from multiple cameras.
    a. Video servers can be located in low, medium or high fog nodes, depending on the operator's demand.
    b. Video servers can process video data from cameras, for example, encoding, transcoding or caching.
3. With fog, when a smart device requests video on demand or live video, the fog node sets up the connection to the user equipment (UE).
4. People along the course can watch the racers in their line of sight, and use their smart devices to watch real-time video broadcasting of the race from start to finish. In large events such as marathons, individual users may be able to select their own cameras and angles from a menu of hundreds, zeroing in on the parts of the race or athletes of most interest to them.
5. At the same time, viewers at home can watch the event in real time.

The HD cameras produce massive amounts of data. In an edge-to-cloud computing model, this data must be transmitted to the cloud for video processing. This results in very expensive backhaul bandwidth. If the bandwidth is insufficient, the congestion will affect video quality. It could also affect other applications sharing the backhaul link.

A fog architecture is based on a hierarchy of low, middle, and high fog nodes. Low fog nodes — depending on how much functionality is required — can be located inside or adjacent to the HD cameras or local video servers. As shown in Figure 2, the video can be pre-processed (encoding/decoding and transcoding) locally by video servers equipped with low fog nodes (or adjacent to fog nodes).

Mid level fog nodes can perform aggregation functions, for example merging views from multiple cameras and performing video switching to the most interesting camera views. Analytics algorithms in mid-level fog nodes could detect features in the video streams – such as an athlete's bib number - and share the video, precise time and location, with real-time tracking and lap timing applications.

High fog nodes can cover a bigger area than low fog nodes. High fog nodes converge data from low fog nodes. High fog nodes are deployed with even more powerful video servers than the servers located near cameras. Some system and router decision services can be executed at the high fog node level of the hierarchy. For example, when a user requests a video live from a low fog node on a local video server, these services located at the higher levels of the hierarchy can find and dynamically build the low-latency route from the original video server to the user's smart device.

Fog also supports data sharing between fog nodes on the same and higher levels and with one or more clouds. The fog architecture uses strong security and digital rights management, ensuring video is only transmitted to recipients who have permission to view it.

There is a smart decision element in fog, which can dynamically distribute the video analysis's algorithms or services to different fog nodes according to their latency requirement. Real-time video processing can be executed in fog nodes inside or adjacent to video servers located near cameras, which can shorten the processing time of video data. With fog, video on demand services can be deployed closer to users.

Fog computing virtually eliminates video latency and saves expensive network backhaul bandwidth. During the race competition, if there is an emergency, fog nodes can also perform real-time analytics to provide specific types of information to first responders.

The fog nodes handle communications to 4G/5G networks. Fog nodes can be deployed in any type of terrain where interference might be a problem. If the wireless quality degrades, the video server equipped

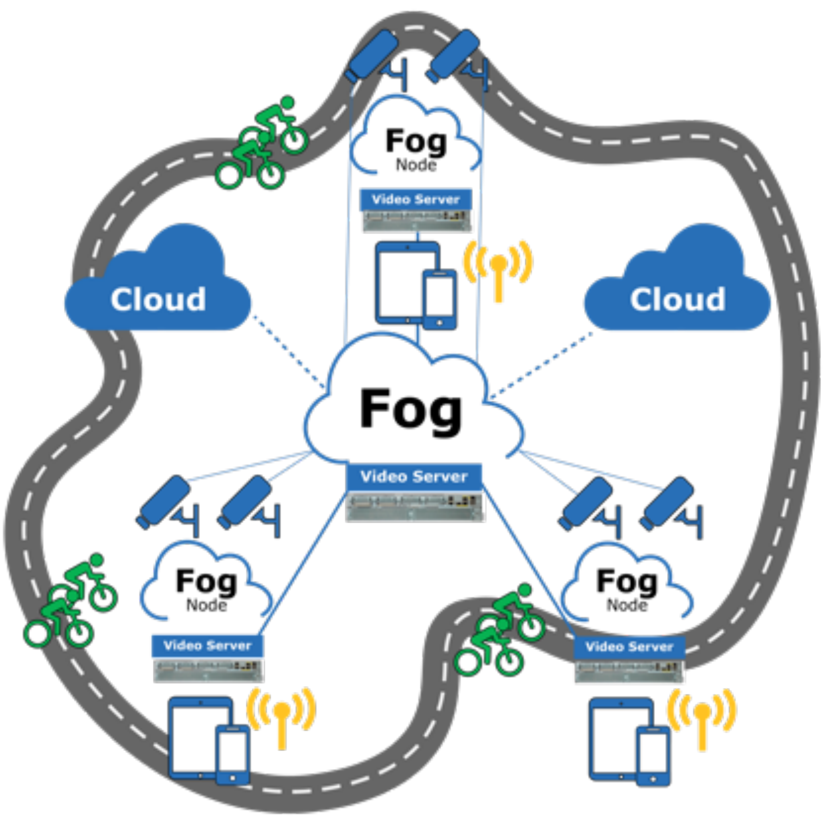

Figure 2. Live video is deployed in a layered fog configuration during a bicycle race.

with onboard or adjacent fog nodes can adjust the video coding rate. 4G/5G fog nodes can share infrastructure resources, such as hardware and software resources, which can also improve the efficiency of the video servers.

Fog networks also support agile load balancing and high reliability. Fog systems can adjust the resources of workloads according to the load of the fog node, moving workload from high fog nodes to low fog nodes. Loads can also be balanced between fog nodes on the same level of the fog hierarchy, sending traffic east-west between congested fog nodes and over to adjacent nodes with resources to spare. If a fog node is down or unreachable for any reason, the workload on it can be moved to an adjacent node, providing high reliability to vertical applications.

Pillars of Fog Computing

The OpenFog Consortium has identified eight pillars of the fog computing architecture. Live video broadcasting maps to the eight pillars as described below:

- **Security:** Fog nodes protect video data during transport and processing, from the camera to the cloud. Video data is processed and stored in local fog nodes. Also, security of 4G/5G networks can help to avoid illegal user access to fog nodes or the video data they are processing.
- **Scalability:** Fog can easily adjust quantities of storage resources and computer resources between fog nodes. When the video workload is too high for a fog node, the fog nodes can dynamically move storage resource and compute resources from the busy node to one that is idle.
- **Open:** The fog architecture supports several operators in the system including different wireless operators, hardware operators, software operators, video service providers and different vertical

clouds. Fog is not limited to a special operator or special video service producers, which can improve the efficiency of developing fog systems. Fog supports several vertical clouds, which can provide more business opportunities for operators.

- **Autonomy:** With fog computing, live video systems can be more autonomous by keeping more computing, storage, analytics, and other services local.
- **Programmability:** Video services can be programmed directly in some accelerators, such as FPGAs or GPUs, to shorten the latency of processing video data. Fog nodes and video services can be easily added or deleted without having to redesign the whole system.
- **Reliability/Availability/Serviceability (RAS):** System-wide decisions and policy management in high fog nodes can provide RAS in the live video scenario. This improves service availability.
- **Agility:** Live video data is processed and transferred in fog, eliminating the need to transfer the data to the cloud to be processed. Video data is processed in fog nodes in real-time. Fog decision-making helps source fog node transfer-encoded video data to the target fog node nearest to the viewers. All of the algorithms running on the fog hierarchy can be dynamically updated during an event.
- **Hierarchy:** The fog-based live video system is divided into several layers to deploy real-time services and non-real-time services. Video server and real-time video services and algorithms are deployed in low fog nodes close to cameras and viewers. Non-real-time services are deployed in high fog nodes. Video data from cameras can be aggregated to high fog nodes to forward to the cloud for fog-cloud collaboration.
- **Distribution:** When co-deployed, 4G/5G and fog systems can share the distributed infrastructure, including hardware and software resources, which can improve the efficiency of servers.
- **Load Balancing:** When high fog nodes manage low fog nodes, they can control their load balancing. If some fog nodes are overloaded, their workloads can be moved dynamically to idle nodes.

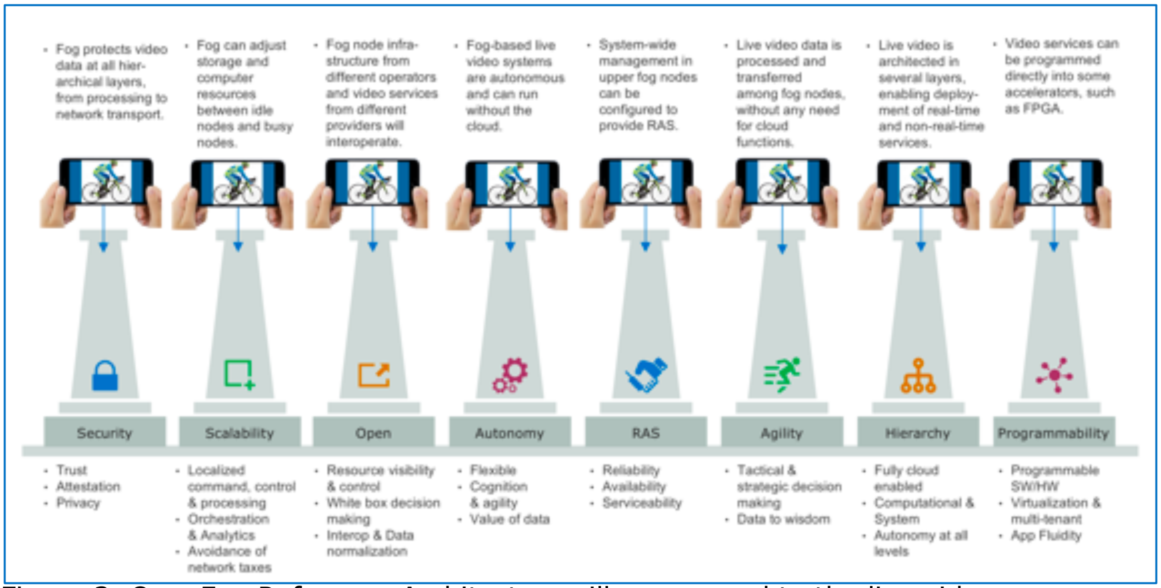

Figure 3. OpenFog Reference Architecture pillars mapped to the live video use case.

Architectural Design

The fog architecture for live video is designed according to Figure 4.

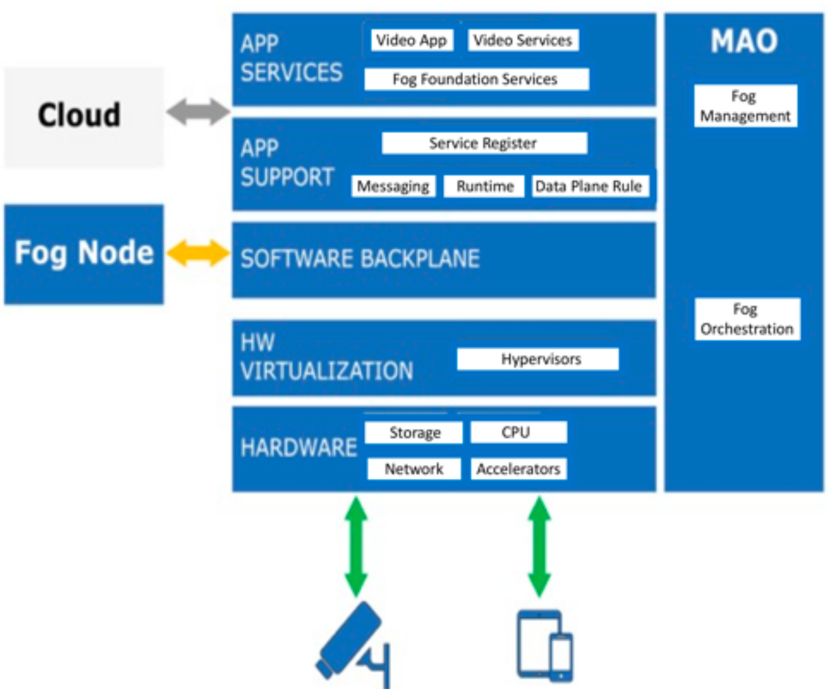

Figure 4. Recommended fog architecture for live video.

The App Services Layer. Video applications and video services are deployed at this layer. Video services are divided into two categories: fog foundation services and video field services.

Fog foundation services include the base services that are supplied by the platform, such as device management, data and analysis service, command and notify service, etc. Video field services include the services related to the video application itself, such as encode/decode services, compress/decompress services, transcode video services, etc.

Video services or video applications are responsible for managing the video data, such as managing video users, command processing, or data transfer.

The App Support Layer. This layer supplies the basic software to support video services, such as:

- Storage – Fog provides video data management, storage and persistence capability to support durable persistent and in-memory caches.
- Service register - Fog manages the register/unregister/search of video services and other base services as part of its orchestration capability.
- Data plane management – Fog manages the traffic rules of video data, and data movement within fog nodes and across fog networks.
- Application management – Fog manages the lifecycle of services.
- Runtime engines – Fog provides the execution environments for applications and micro services, including virtual machines, containers, etc.
- Message communications infrastructure – Fog provides infrastructure-to-message communications among micro services.
- Application servers – Fog supports host video servers.

Backplane Layer. The software backplane provides the fog node resource (compute, storage, and networking resources) to software, including the operating system, software containers, and file system software. The most basic cloud-service model is Infrastructure as a Service (IaaS). IaaS providers can offer computing infrastructure, such as virtual machines and other resources, reused in a fog backplane. The backplane layer is a key to interoperability in OpenFog software.

Hardware and Hardware Virtualization Layers. The hardware layer includes computer, storage, network and accelerator. To reduce the latency of video processing, a high-performance accelerator, such as an FPGA or GPU, can be used. Hardware virtualization includes hypervisors and containers.

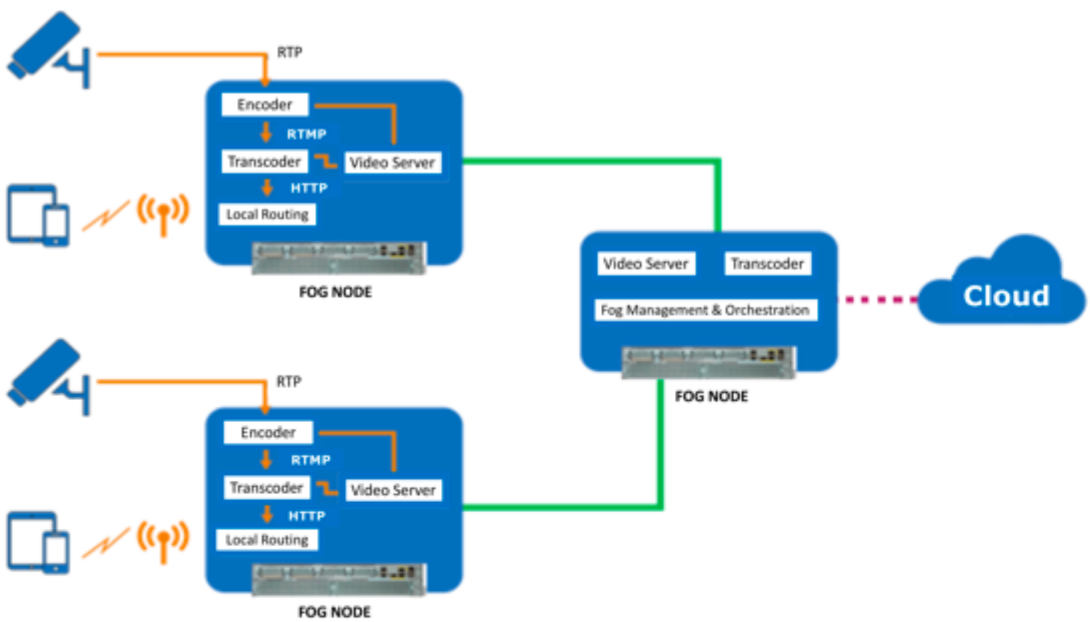

Figure 5. Configuration of fog node hierarchy for the live video use case.

As shown in Figure 5, at a high level the low-latency video fog architecture is divided into two layers: low fog nodes and high fog nodes. Management and orchestration functions are deployed in the upper fog nodes, which may connect to clouds.

Low fog nodes. Low fog nodes may be deployed with 4G/5G base stations. Low fog nodes are close to the cameras and the user's smart devices. Cameras nearby are connected to low fog nodes through cable. Real-time video services, including encoding, decoding and transcoding video data services, are deployed into every low fog node. Video servers are also deployed here, as described on page 15 in the fog-based race course configuration. It can store video data locally and manage video services and applications.

High fog nodes. Many lower fog nodes can connect to one or more high fog nodes. A more powerful video server is deployed at this level to help route requested video data between different low fog nodes and between the high fog node and cloud. Sophisticated fog networks may also have one or more intermediate layers, managing video analytics and other mid-level functions.

Management and orchestration. Management and orchestration are deployed in the high fog node. The high fog node is responsible for managing all fog resources and orchestrating video services into suitable fog nodes.

Implementation Considerations

There are a variety of options for implementing fog networks for live video broadcasting:

- Fog nodes can be co-deployed with a 4G/5G network entity. In this instance, the fog system can use wireless services to help shorten the latency of video services.
- Video servers can be deployed in both low fog nodes and high fog nodes.

- Video data and real-time video services can be deployed in low fog nodes.
- Mid level fog nodes can host optional capabilities like video analytics, object recognition, augmented reality, etc.
- Existing video protocols can be leveraged, such as NFV, MEC, MANO, etc.
- Popular open-source technologies can be leveraged, such as OpenStack, OPNFV, etc.

Here are some recommended deployments for low, middle and high fog nodes:

Low fog nodes: In a 4G/5G configuration, these fog nodes are deployed within the cell site and with multiple instances.

Middle fog nodes: These fog nodes can be deployed in middle locations, such as at the CU/CRAN, and have stronger capability than low fog nodes. For example, they provide more compute and storage resources than low fog nodes.

High fog nodes: These fog nodes are deployed with CN and are typically based on more powerful servers. They have more capabilities than lower level fog nodes.

Video Processing Algorithms and Analytics

The video processing algorithms and analytics used in the bike racing implementation are as follows:

1. Video data encoding and decoding algorithms. Because the encoding protocols that video servers use may be different than

the protocols used by original video data captured by cameras, a video data encoding service is needed to encapsulate the original video data from the camera into the target protocol of the video server. It is also needed to decode the data.

2. Video data transcoding algorithms. Because the protocols the video server uses may be different than the protocols used by the viewer's smart devices, a video data transfer service is needed. The service transfers the video data into the device's target protocol, for example, HTTP-FLV.

3. Optimizing video dynamic encoding based on wireless QoS from a 4G/5G network. The video encode rate can be adjusted by the wireless quality of radio channel conditions. For example, the 4G/5G network can provide wireless radio channel conditions to the video server. When the wireless quality of the radio channel is very good, a high encode rate is adopted. If the wireless quality is not good, a lower encode rate is adopted. With this optimization, the fog system decreases the loss of video packets over the radio channel, improving the user experience.

4. Optimizing video dynamic encoding based on TCP packets provided by 4G/5G network. The video encode rate can be adjusted by loss of a TCP packet. In this case, the 4G/5G network will provide the video packet loss to the video server. When the TPC packet loss is very small, a high encode rate is adopted. If it's large, a lower encode rate is adopted. With this optimization, fog systems can decrease the loss of video packets in a 4G/5G network, improving the user experience. These adjustments can be made in <1 second timescales.

5. Video AI algorithm. When the video data is located in fog nodes, the associated video analysis algorithms, or AI, can be deployed in the video server. For example, in the event of an emergency, image analysis based on video data can help first responders. When co-deployed with Hadoop, for example, the algorithm can

be optimized by analyzing historical data. Image recognition algorithms can identify objects and extract features from the video (for example the bib numbers of athletes) and make this data available to other applications in the fog or cloud.

6. <u>Flexible storage.</u> Fog supports multiple storage solutions, tailored to the application requirement. In this use case, real-time video data is stored in low fog nodes, which is close to the data source and wireless devices. This can provide low-latency video service and also strengthen protection by local security teams. Video data can be stored in a high fog node to forward filtered or pre-processed video data to the cloud.

<u>Communications</u>

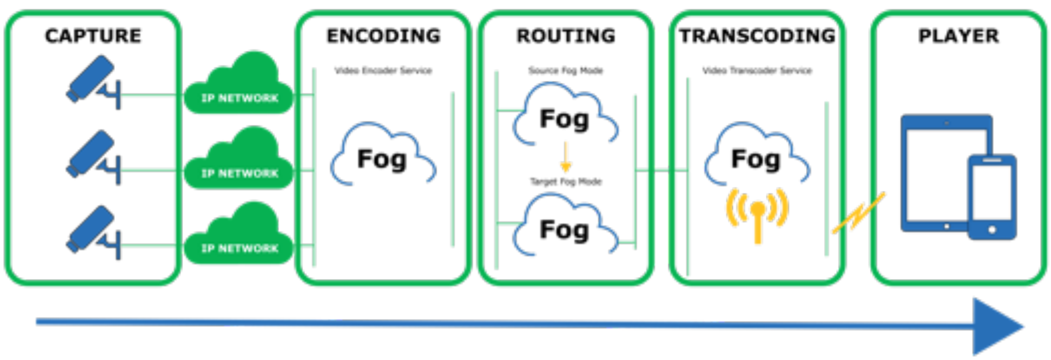

Figure 6. Data in motion of live video in a fog environment.

This section describes the communications setup procedures for enabling low-latency live video in the bike racing use case.

<u>Step 1: Capture.</u> Cameras are connected with near low fog nodes by IP cable. After a camera captures video data, it transfers the video data to the low fog node to be processed. The camera may be very high performance, for example, 1920x1080p and may support H.264

Baseline/Main/High profile. The original video data may be in several popular protocols, such as the RTP protocol.

Step 2: Encoding. Video servers may support different protocols than the original data protocol, for example, the RTMP protocol. The video encode service is deployed in low fog nodes to help encode the original video data into target protocols. Once the original video data are sent from camera to low fog nodes, the video data is sent to this service to be encoded. (This service can be located in the camera if the camera supports it.) The encoded data is sent to the routing service to be processed.

Step 3: Routing. A wireless smart devices can be identified as User Equipment (UE). When a UE requests a live video, the UE will access a wireless base station nearby and send the video request to the video server near the wireless network.

Because the low fog node is co-deployed with a wireless base station and the video server is deployed in it, the video request message is sent to the video server and the access low fog nodes.

Next, the video server looks for the requested video data. If the video server finds the requested video address in its own node, the service will setup a connection between the UE and the fog node. Otherwise, the video server will send the request to the upper video server in the high fog node. These requests could be for live streams which are forwarded in real-time, or for archived replays of previous events, which are retrieved from the fog's hierarichal storage, or from the cloud.

The upper video server will look up the video address in the low fog nodes. When the right fog node is found, the upper video server will build the bridge between the target fog node and the access fog node, and also set up a connection between the UE and the access fog node.

Once the encoded data is sent to video server, the server will send it to the transcode service in the same fog node, or the access fog node of the UE using the same connection setup.

Step 4: Transcoding. The viewer's smart video player may need different video protocols, resolutions, frame rates or data bandwidth than typically used in encoded video data, such as the HTTP protocol. The video transcode service is deployed in low fog nodes to transcode the video data into target protocols of players in smart terminals.

Step 5: Play. The UE receives requested video data from the access fog node. The video server sends the transcoded video data to the wireless smart devices.

If a user requests the video from the internet, the video request message may be sent from the cloud to the video server's upper fog node. The requested video can be transferred from the fog node running the video to the cloud, and from the cloud to UEs. If local copies of the requested videos are stored in nearby fog-based caches, those copies are used to reduce network bandwidth and cloud load.

4G/5G and Fog

4G/5G networks can collaborate with fog networks to provide an excellent video experience to users. Following is a breakdown of key interoperability features:

- The 4G/5G network can route the data between 3GPP wireless networks and low-tier fog nodes, which can shorten the latency of transportation of video data.
- A 4G/5G network is a naturally distributed network. Distributed fog nodes of different layers can be co-deployed with distributed

4G/5G network entities.

- Fog systems can use the wireless service to optimize video experience. The 4G/5G network can provide some wireless services to fog, such as bandwidth optimization services, wireless quality services, location services, etc. The video service in fog node can use these services to optimize processing and transportation of video data.

- 4G/5G network and fog system are based on some common infrastructure technology, such as hardware virtualization and software virtualization. Some protocols or open source technology can be shared with them, such as NFV and OpenStack. All of those can improve the efficiency of servers.

Messaging

As shown in Figure 7, fog provides messaging communication for services in intra-nodes/inter-nodes, and also for services between fog nodes and cloud. Standard message communication protocols can be used, such as RabbitMQ, ZeroMQ, MQTT, and REST API.

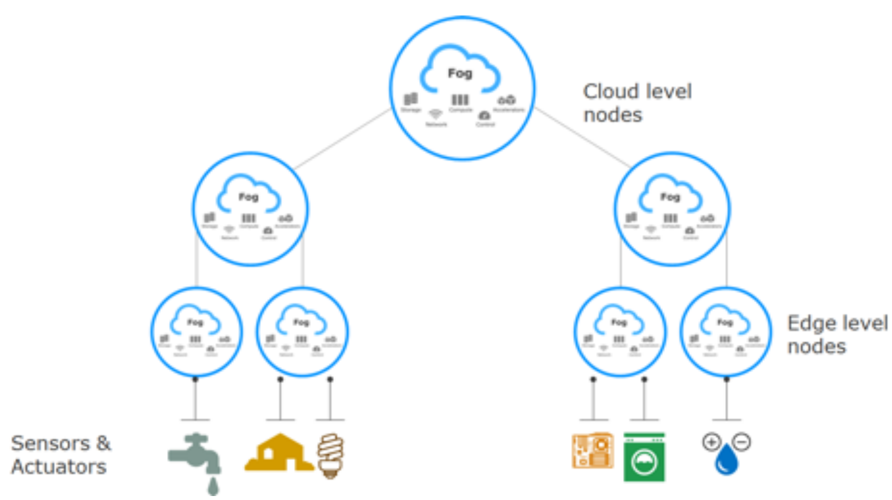

Figure 7. Messaging communication services within the OpenFog architecture.

Testbed Considerations

Testbeds are created to verify the use case and fog architecture. The live video broadcasting use case content in this document is intended to be a reference for the development of related testbeds, and the architecture and communication chapters can provide a guide for design and implementation of fog systems.

The composition for the testbed can be simplified. Cloud is not necessary in an initial testbed deployment. To simply the testbed, fog management and fog orchestration can be ignored at first.

Several wireless technologies can be considered, such as LTE, 5G, etc. Because the fog system is open, any video data analysis service or algorithms not mentioned above can be added, deleted or modified.

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 sponsors 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. 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 intends to partner with standards development organizations and provide 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 application programming interfaces (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.

For more information, please refer to the documentation and resources on the OpenFog website or 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, our 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.

11 Authors and Contributors List

Authors	Contributors
Meiqing Huang, ZTE Corp.	Chuck Byers, Cisco Systems
	Evan Birkhead, OpenFog Consortium
	Judith Kelley, OpenFog Consortium
	Jingyi Zhou, ZTE Corp.
	Nancy Liu, ZTE Corp.

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/