Fog Use Case Scenarios

Use Case: Aerial Drones

Vertical: Supply Chain Delivery

An OpenFog Consortium Architectural Use Case

1 Snapshot: Aerial Drones

WHY FOG

Why is fog the best architecture for this use case?

Commercial drones operate in many environments, from aerial to subterranean. They are becoming an important tool for supply chain delivery in many industries. Fog enables near real-time adjustments and collaboration in response to anomalies, operational changes or threats. Fog computing enables drones, as self-aware individual fog nodes, to interoperate and cooperate as a dynamic community.

WHICH FOG PILLAR

Which fog pillar best describes this use case?

Drones with onboard fog nodes incorporate the autonomic functions defined by the fog infrastructure. This gives drones the ability to make appropriate operational decisions if there is an interruption in communications with the cloud or data center resources or in the case of real time operational control events.

VALUE

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

Fog computing creates an interoperable infrastructure that makes it practical for drones from many different vendors to share the skies safely without adding layers of proprietary communications, interfaces and networks. Without fog interoperability, complex interfaces would introduce dangerous communications latency, increased hardware footprint, expense, compromised safety, and other problems.

CLOUD & EDGE

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

Fog computing provides a hierarchical infrastructure across: mobile fog nodes (onboard drones); fog-enabled ground support; and higher-level fog nodes for additional resources and functions (such as aggregation, consolidation, virtualization and containers). A fog infrastructure enables architects to efficiently distribute services across compute, storage, networking, security, and other functions.

2 Table of Contents

3 Introduction

Note: The preamble section of this document (pages 4 through 11) is common to all OpenFog use cases. It provides descriptions and reference points for fog architectural attributes and properties. The Aerial Drones use case begins on page 12.

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.

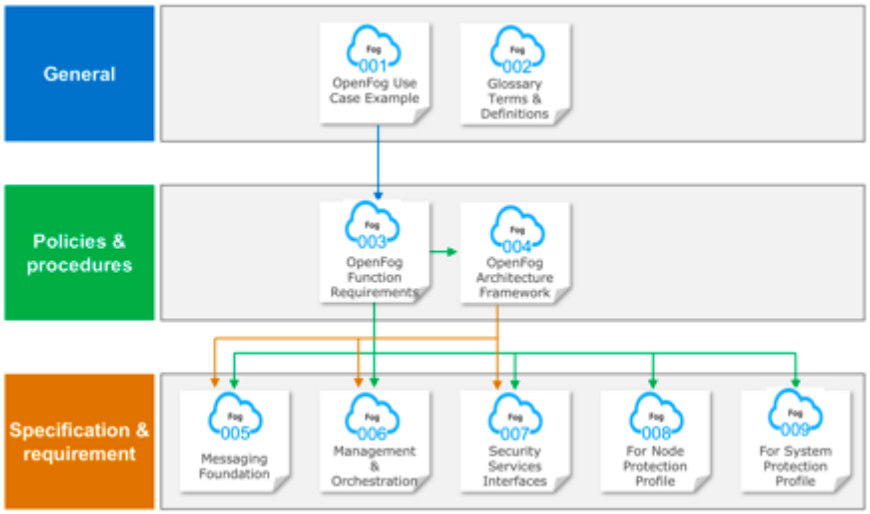

Figure 1. Hierarchy of OpenFog Consortium specification documentation

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: Aerial Drones

Use Case: Aerial Drones

Application: Supply Chain Delivery

Use Case Overview

Unmanned Aerial Vehicles (UAVs), commonly called aerial drones, are becoming a mainstream business tool for a wide range of applications. Supply chain delivery is one of the fastest growing of these commercial use cases. Given the high costs associated with almost all forms of delivery—shipping, trucking and aviation—it's not surprising that industries are exploring how drones can augment traditional delivery methods to reduce costs.

Beyond cost reductions, aerial drones can provide the basis for innovative new services that simply aren't as practical today. They can be the basis for new business models and sources of revenue.

In order to realize this potential—and as yet unimagined applications—drones must address concerns about safety, security, reliability, performance, and other challenges. All of these issues can be addressed with a fog-enabled infrastructure that supports drone operations and augments an edge-to-cloud only computing model.

Aerial drones can be designed for single use or reconfigured and reprogrammed to support many applications. The programmability and multi-tenant advances created by fog, combined with the community behavior, also enabled by fog, make it ideal for supporting multiple

drone fleet operators that may share a common ground support infrastructure.

Fog computing distributes compute, storage, networking, security, and other functions across a hierarchy of fog nodes.

Fog-enabled drones. One of the architectural pillars of fog computing is autonomy, which encompasses situational awareness, analysis, and sub-millisecond response to changing conditions. Drones require sufficient on-board intelligence to act with autonomy when needed. With their autonomy capabilities, drones with on-board fog nodes can also act as a community, working collectively with other drones to execute a common mission.

Fog-enabled ground support infrastructure. Fog nodes embedded in or adjacent to ground infrastructure devices communicate with fog nodes on drones. This hierarchical, distributed model enables complex, split-second coordination of landings, takeoffs, flight path management, loading, unloading, maintenance and a myriad of other tasks associate with high-volume drone operations.

Regional fog nodes. Regional fog nodes provide additional resources and functions, such as data aggregation, consolidation, virtualization and containers.

While aerial drones represent the most common type of drones deployed, there are other operational models as well: aquatic, underwater, terrestrial and subterranean drones. This use case focuses on the use of aerial drones in supply chain delivery. It also addresses mixed operational models, in which a supply chain delivery route requires the cooperation of drones designed for different operational models.

 **Challenges**	• Safety concerns over drone crashes and rogue (malicious or malfunctioning) drones. • Concerns over hijacking of drones and their cargo. • Concern that a disruption in network connectivity, or insufficent bandwidth, will affect the predictability of drone performance. • Consolidated fleet management, even when the drones are owned and operated by different vendors. • Regulatory concerns over non-line-of-sight drone operations.
 **Solution**	• Fog computing nodes located on the drone to support autonomous awareness, analysis and sub-millisecond response to changing conditions in flight. • Fog computing nodes on the drone provide a full complement of security measures, from downloading and installing software patches to perimeter defense. • Fog computing controllers on the ground communicate with fog nodes on drones to manage the complex, split-second coordination of landings, takeoffs, loading, unloading and maintenance. • Inter-fog node communication enables close proximity operations while ensuring collision avoidance. • Fog computing provides the onboard computing, networking, programmability and other features to enable drones to work in hives and swarms on a collective task, with each drone in the collective still maintaining its ability to act autonomously. • The multi-tenant properties of fog, combined with fog-enabled community behavior, makes it ideal for supporting multiple drone fleet operators sharing a common ground support fog infrastructure.

Technology

- The hierarchical fog architecture shares and processes data as close to the source as possible for the greatest performance and efficiency (for example, low latency, bandwidth efficiency, and reliability).
- Fog-based drones enable redundancy, sensing systems, and coordination in order to handle close proximity navigation, changes in weather conditions, and other in-flight adjustments.
- Fog-enabled ground support makes interaction with sensor arrays, radio access points, postal boxes, landing platforms and other devices, part of a richly interconnected network of distributed intelligence.

Degrees of Autonomy

Aerial drones, also known as unmanned aerial vehicles (UAVs), are already being used in a variety of industries. Drones can augment traditional package delivery methods and open the doors (or the skies) to services that weren't practical before, such as rapid delivery of supplies to disaster areas and remote regions.

Drones may not replace traditional supply chain delivery methods, such as shipping, trucking and commercial air transportation. However, aerial drone fleets can offer compelling benefits to complement these delivery channels, including:

- Making package delivery more practical for hard-to-reach areas (from crowded cities to remote communities)
- Reducing labor costs or dependence on human labor
- Greatly reducing the time between placing an order and receiving the delivery
- Reducing traffic congestion
- Reducing some of the burden of infrastructure maintenance

- Enabling new package delivery models

The Society of Automotive Engineers created a scale for autonomous vehicles to show the degrees of automation and autonomy. A similar delineation applies to aerial drones, as shown in Figure 2.

Fully autonomic drones will share the skies with human-driven and semi-autonomic transport aircraft. Fog computing will also provide the interoperability architecture elements for cooperation as the use cases for drones evolve.

Fog techniques can be valuable for all these levels but are essential for levels 3-5, as shown in Figure 2. Currently, most commercial drones support level 2-3 operations. In most jurisdictions, regulations restrict operation to line of sight, with the full attention of a pilot, which is generally applicable for levels 2 and below. However, with time, technology and regulatory relief should allow the higher levels and the fully autonomous operation that is the focus of the rest of this use case.

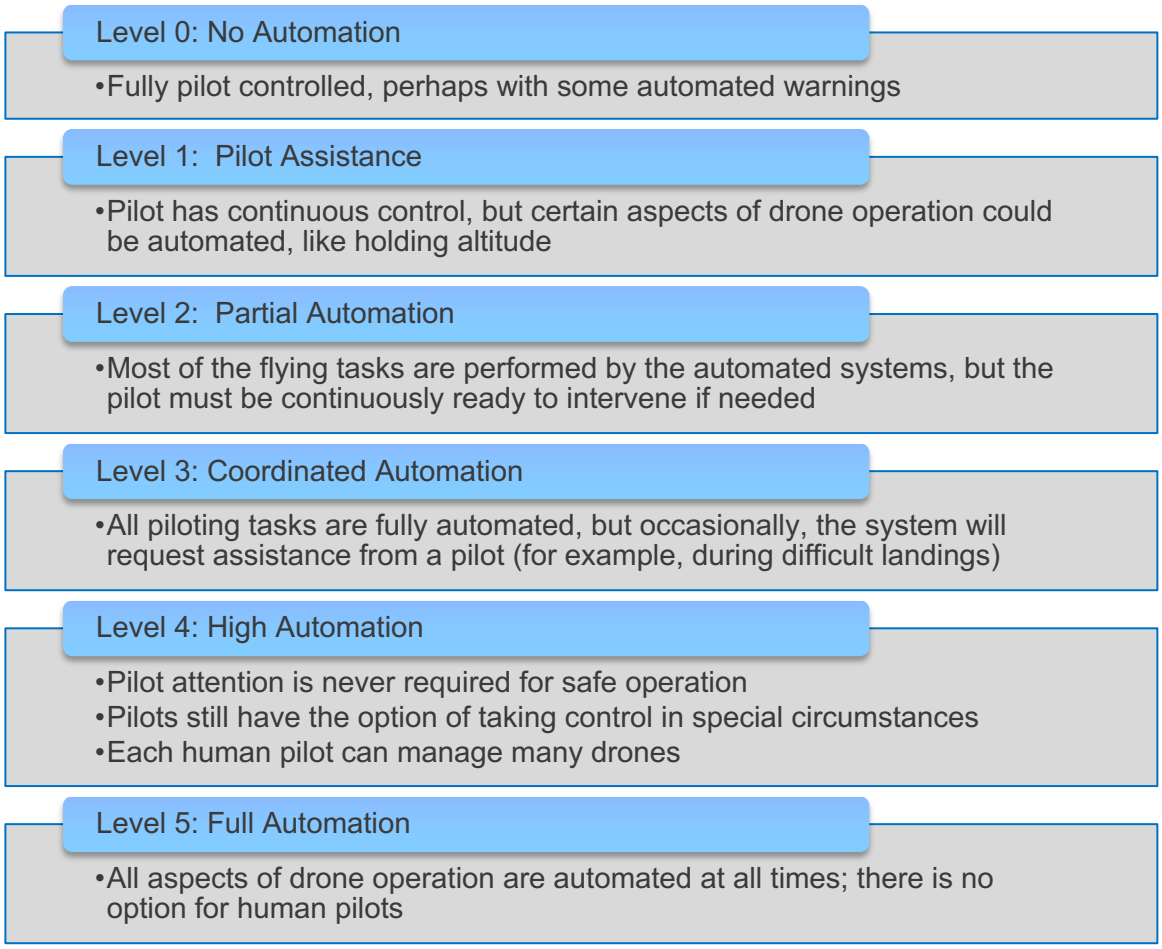

Figure 2. Degrees of automation and autonomy for drones can be characterized using the same scale as the one used by the Society of Automotive Engineers for autonomous vehicles.

The following section reviews several challenges that must be addressed in order to realize large-scale use of aerial drones for commercial package delivery.

Safety in the Air

Shared airspace is the only way to arrive at high-scale aerial drone delivery services. Consider the split-second timing required to

coordinate the loading, takeoff, landing and maintenance of commercial drone fleets.

A drone could be traveling at 100 miles per hour on final approach—or 147 feet per second. Consider that the best cloud round trip latency is around 80 milliseconds. During that time, the drone flies about 12 feet between round-trip cloud messages. Delays introduced by routing all of that information through the cloud can make it impossible to achieve near-time response, and this will result in unsafe landings, collisions and various other safety concerns.

Essentially operating as a mobile fog node, a drone has the ability to make split second decisions about its own operational safety as well as the other drones or environment around it. It can have the ability to alter its own programming, in order to facilitate adaptive models to best fit the immediate situation.

Fog controllers on the ground provide the proximity required to shorten the communications loop between the drone and the "control tower." Latency can be reduced to such a degree that a drone will only travel two inches before the next update is delivered. If that same communications were to go through the cloud, the drone will have traveled 12 feet. That difference is very significant for safe drone operations.

Autonomy in the air also means that the aerial drone can run through self-check procedures to ensure that all systems are operating properly. In the event that the drone detects a problem with its own operational condition, the fog computing node on the drone can take the appropriate action to correct or compensate for the problem—or return to the hub for maintenance.

A subset of this on-the-ground information can also be routed to higher fog layers or the cloud for analytics. Complete logs of the communication can be sent to the cloud for long-term archiving.

The Effect of Disruption in Network Connectivity on Drone Safety

Drones require sufficient bandwidth and reliable network connectivity to operate in a consistent and safe manner. However, environment, weather and even jamming may interfere with the network or signal. Bandwidth bottlenecks can occur when central management systems communicate with multiple drones at the same time. This delay can result in dangerous delays in control transmission and drone tracking.

Fog-enabled drones can make autonomous decisions based on analysis of a situation; this autonomy provides continuity and ensures safe operations even when there is a disruption in connectivity.

In remote areas, access to the cloud may require expensive satellite links, making drone supply chain delivery cost-prohibitive. In these situations, fog-enabled drone autonomy—or access to regional fog nodes for updates on airspace and other environmental conditions—can make supply chain delivery with drones a practical reality in any region.

Managing Multi-vendor Drone Fleets

As large-scale drone package delivery takes off, it will become impractical for every vendor to have a fleet that operates independently of every other fleet on the planet. In order to scale aerial drone use, stakeholders in supply chain delivery will need to develop standardized drone hubs that can coordinate the flights of many types of droves from companies, much like airports do. This

coordination will require standardization in the messaging, data and communication mechanisms to ensure that drones can interoperate in shared airspace.

A multi-tenancy architecture is required in order for multi-vendor drone fleets to share a common infrastructure. However, drone fleets present a far more complicated hierarchy of tenants.

Fog computing handles this complexity by extending the concept of multi-tenancy. Here are two examples:

- Fog combines multi-tenancy with community behavior, which is the ability of an individual drone to be part of a collective and still take autonomous action.
- Fog enables architects to design flexible, even universal Ground Control Stations (GCS) that don't have to dedicate space for drones or payloads; yet isolation is enforced between drones from different owners.

Regulatory Requirements

Aerial drones will need to operate in a complex regulatory environment. The U.S. Federal Aviation Administration (and their counterparts worldwide) has regulated the operation of drones. Some of these regulations limit drone flights to within line-of-sight of the operator and prohibit operation in many flight areas and conditions. Fog computing techniques can help manage the risks that created the regulatory environment and may one day help relax it. Future rulemaking will weigh in on opening up operations beyond vision line-of-sight when enabled and secured through advanced technologies.

A fog infrastructure helps address the technical elements that support regulatory requirements. Fog computing offers the sub-millisecond latency, compute capacity, network bandwidth, as well as the safety, privacy, reliability, efficiency, multi-tenancy, hierarchical control, distributed storage, and security required for drone operations.

8 Business Case

Much has been written about the benefits of using aerial drones to transform supply chain delivery, including:

- Faster, more cost-effective delivery to remote locations, including congested inner cities to rural locations
- The ability to cover inhospitable terrain easily
- Less congested roadways
- Reduced labor costs
- Greater energy efficiency
- Cleaner air

When drones are enhanced with fog nodes, the business case is strengthened by the standardization and interoperability that fog computing provides. The business benefits include:

- More choices in suppliers of components, products, and services, which should facilitate innovation and keep costs down
- Make it faster and simpler to scale drone operations
- Create new applications and services without adding new layers of proprietary networks that can add costly, inefficient management overhead

Fog computing will also make it practical to routinely mix different types of drones for supply chain delivery. For example, while this use case focuses on aerial drones, drones are designed for operation in other environments, as shown in Figure 3.

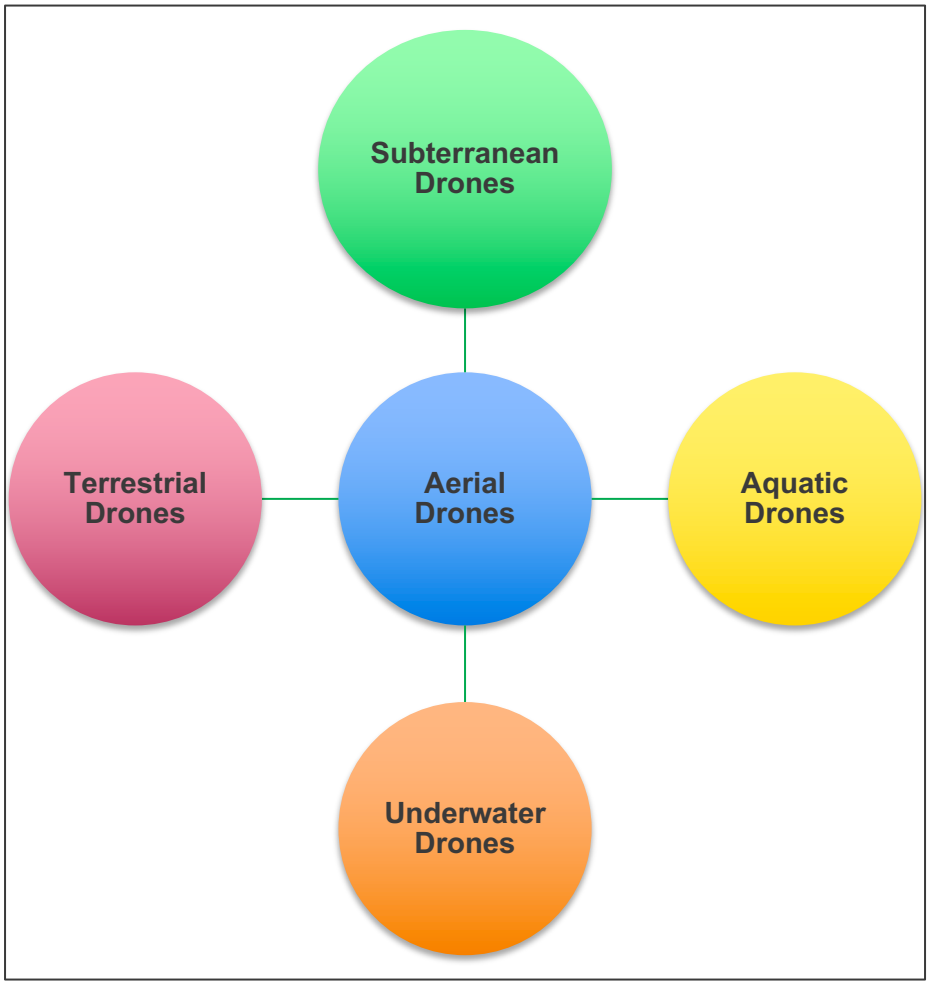

Figure 3. Drones in all of these operational models are now used in commercial applications. Fog computing enables all of these drone types (even from different vendors) to collaborate and interoperate as part of a cohesive and also adaptable supply chain delivery process.

The reason that it's useful to mention these other operational models is because different types of drones might be used cooperatively in a single delivery. This enables architects to construct more complex and dynamic applications, such as transferring goods from an underwater drone to an aerial drone to a terrestrial drone in order to cover a delivery route.

9 Vertical Applications and Services

There are many other examples of how drones can be used for horizontal applications and services across industries, as well as industry-specific applications. Table 1 shows a matrix of just some of the possibilities of aerial drones in industry.

Table 1. The Matrix of Possibilities for Aerial Drones in Industry

Drone Offerings	Single or special-purpose drone	Multi-purpose, modular drone	Drone fleets: single vendor	Drone fleets: Drones-as-a-Service
Examples of Vertical Industries		**Examples of Vertical Applications and Services**		
Agriculture		• Animal tracking • Monitoring weather and crops • Targeted treatment of pests and diseases • Equipment inspection		
Conservation and Environment		• Animal tracking (tags and visual checks) • Surveillance (poachers and polluters)		
Entertainment		• Crowd surveillance • Aerial live video broadcasting • 3D aerial entertainment and billboards		
Healthcare		• Delivery of medical supplies		
Law enforcement/first responders		• Surveillance • Locating missing persons • Support emergency response		
Transportation		• Traffic monitoring and congestion • Package delivery • Last mile delivery of data		
Utilities		• Inspections and Maintenance • Surveillance • Emergency management		

This use case focuses on fog computing for the use of aerial drones in for supply chain delivery. Following are two applications that will benefit from—and even require—fog computing.

- Individual drones used to carry packages
- Individual drones working together in a hive or swarm to carry a heavy package

Individual Drone for Package Delivery

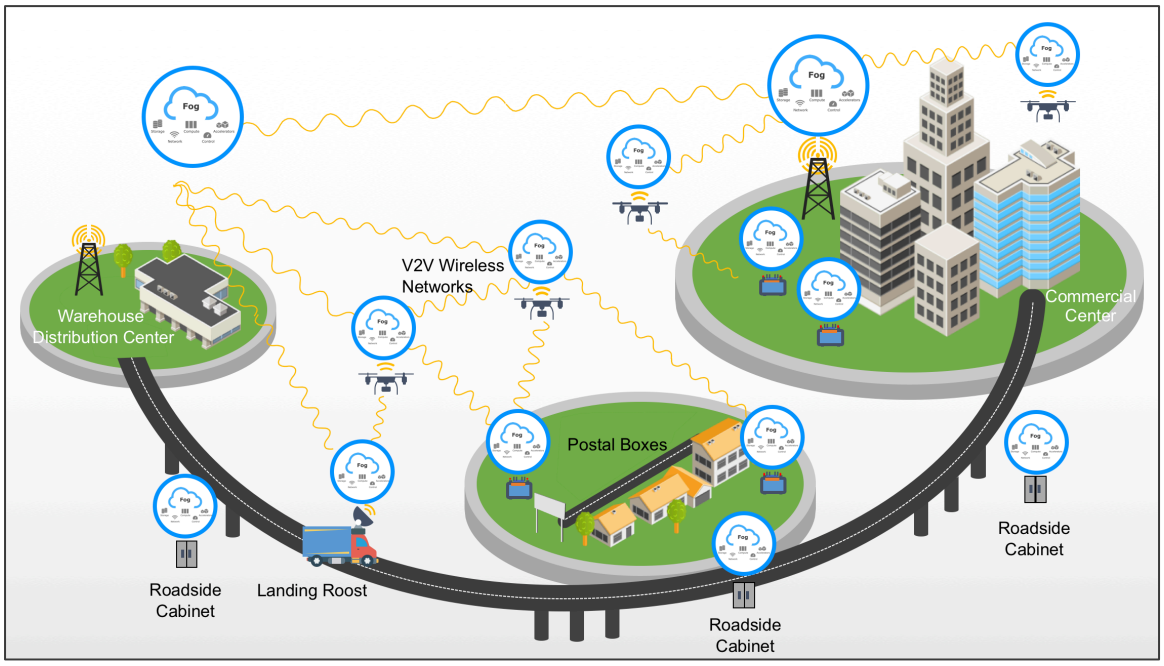

Figure 4. For local deliveries, individual fog-enabled drones can department from warehouses or trucks equipped with landing platforms. There is a constant handoff of information along the supply chain route between drones and the ground support infrastructure.

In commercial supply chain delivery, there are least three layers in the fog hierarchy, as shown in Figure 4. The fog-enabled aerial drone is at the low level of the fog hierarchy. This simply means that the onboard

fog node is designed with the most basic fog capabilities, which are usually compute, network and storage.

Each drone needs the equivalent of a runway, gate and hanger, a way to take on and offload cargo, refueling, inspection and maintenance, and a myriad of other services. One big difference with drones as opposed to traditional deliveries is the speed at which the same services are executed: it may be literally a matter of seconds.

Fog computing makes it possible for architects to design landing platforms of virtually any size with any level of programmable functionality. For example, a larger landing platform can support a fleet of drones. The drone is guided into an individual perch; an array of perches can make up a roost. A fog-enabled smaller landing platform can be integrated into a homeowner's postal box.

Landing platforms can also be stationery or mobile. Figure 5 shows a landing platform for a fleet of drones on top of a delivery truck. This enables the truck to be loaded with hundreds of packages at a central warehouse, drive with its drone fleet to a location near all the delivery addresses and support the fleet of drones as they make multiple short flights to complete the deliveries.

A landing platform—depending on where it's located and how many drones it supports—can be equipped with a highly programmable fog node. This node can provide numerous controller functions, such as:

- Downloading flight logs
- Uploading mission instructions
- Conducting pre-flight checks
- Air traffic control for the airspace around the platform (by collecting and analyzing data from a local surveillance cameras and other sensors)

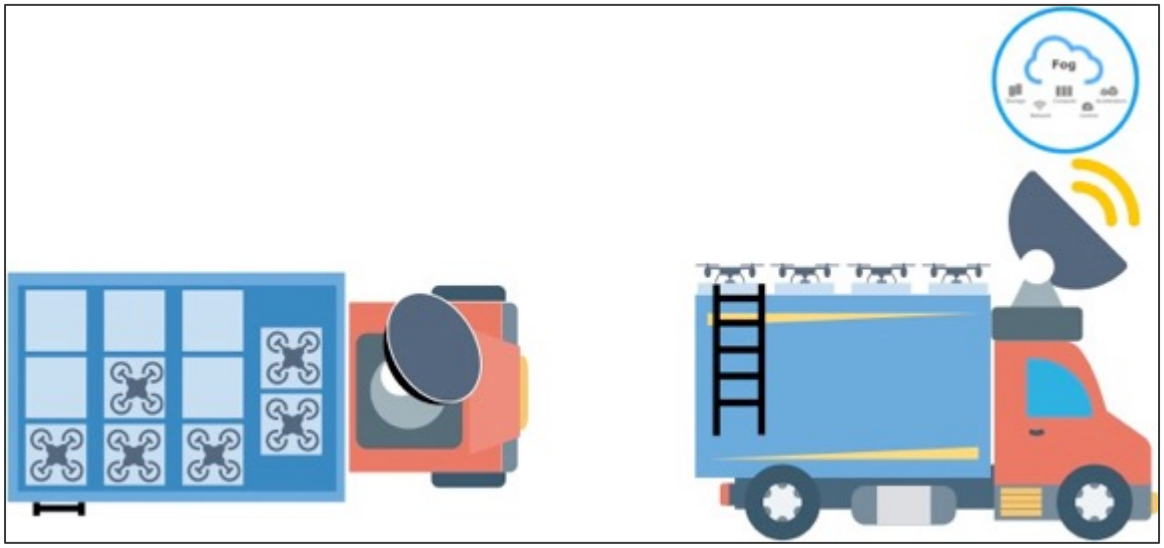

Figure 5. For local deliveries, operators can outfit delivery trucks with landing platforms, creating a mobile landing platform. The trucks themselves are actually fog-enabled nodes, providing similar capabilities (such as air traffic control) as stationery landing platforms.

Drones Working in a Hive or Swarm to Carry Heavier Packages

What happens when a carrier has large packages to deliver? Certainly, large drones will be built for this purpose. However, such drones are covered under a different set of regulations, require a different infrastructure, etc.

Another, more flexible option is trying to designate and preprogram multiple drones to work together to carry a single package. But it can be difficult to predict—and design scenarios for—all the problems that can beset a group of drones working together.

Fog computing architects recognize that there are limitations to relying on preprogramming or remotely controlling cooperative behavior. The solution is to program "community" behavior among drones.

A community is possible because fog nodes are self-aware, peer-aware and self-organizing. While drones are working cooperatively, they maintain their individual autonomy. This means that individual drones can sense and act intelligently to ensure the success of the shared task.

The concept is similar to a colony of bees, where each bee has its own role but can act in unison with other bees dynamically (such as protecting the hive if it comes under attack).

Let's say that four drones are working collectively to carry a heavy package, as shown in Figure 6. When one of the drones begins to develop anomalous behavior, such as flying lower because its battery is running low, the other three drones can compensate for the problem. Note that the other drones don't have to know the reason for anomalous behavior, only the common goal. In this example, one or more drones operating as a hive or swarm can increase their power output to compensate for the faltering drone in order to maintain altitude and speed and complete the mission.

These are examples of other service modification or remediation reprogramming measures, such as:

- Changing the frequency at which the hive or swarm communicates, so that a rogue or malfunctioning drone can't affect the behavior of any other drone
- Sending a warning to the other drones in the collective not to trust a drone or share information with a drone

Changing the flight pattern to keep the rogue or malfunctioning drone from interfering in the flight pattern or mission of the hive or swarm.

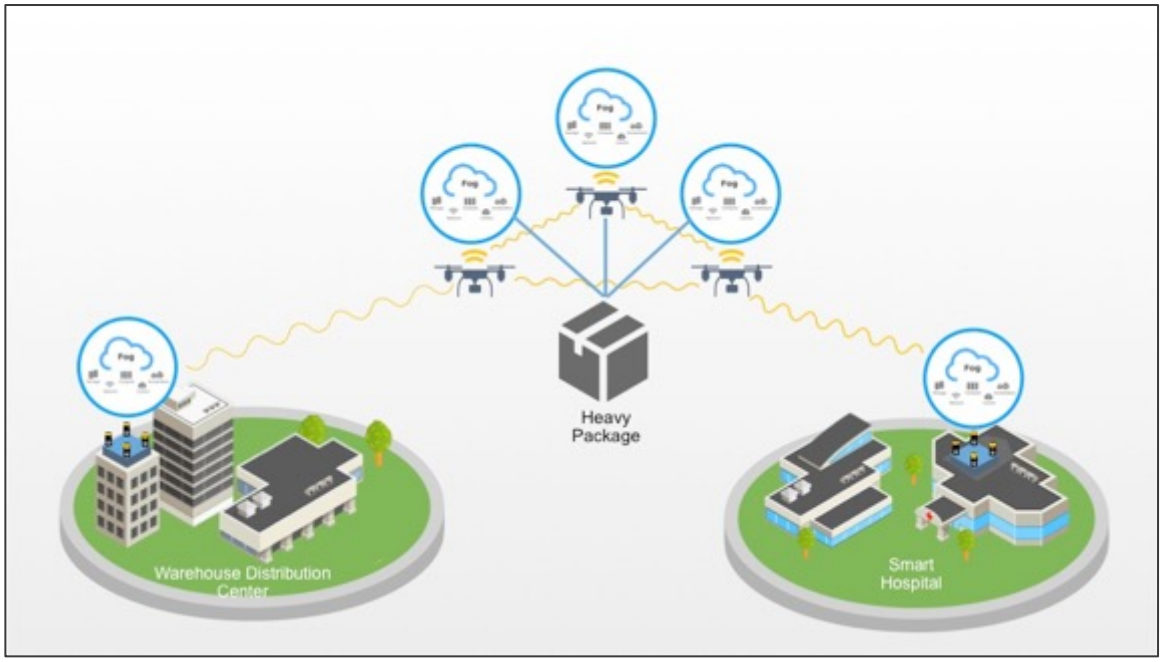

Figure 6. Fog enables operators and owners to use drone resources more efficiently (e.g., repurposing idle drones) and achieve a greater degree of mission flexibility. With fog-enabled community behavior, multiple drones can work together to accomplish a specific task. In this example, multiple drones can form a hive or swarm to carry a heavy package. The members of the hive or swarm retain their individual autonomy, which gives them the ability to adapt in near real time to issues in transit. There are also multiple forms of collectives to fit structured or dynamic supply chain requirements.

10 Advantages of the Fog Computing Architecture

The fog computing approach provides the interoperability, messaging, and interface standards to create a cooperative community of drone fog nodes, while greatly reducing latency, network bandwidth and availability constraints. As shown in Figure 7, the architecture for fog computing is based on eight pillars as identified by the OpenFog Consortium.

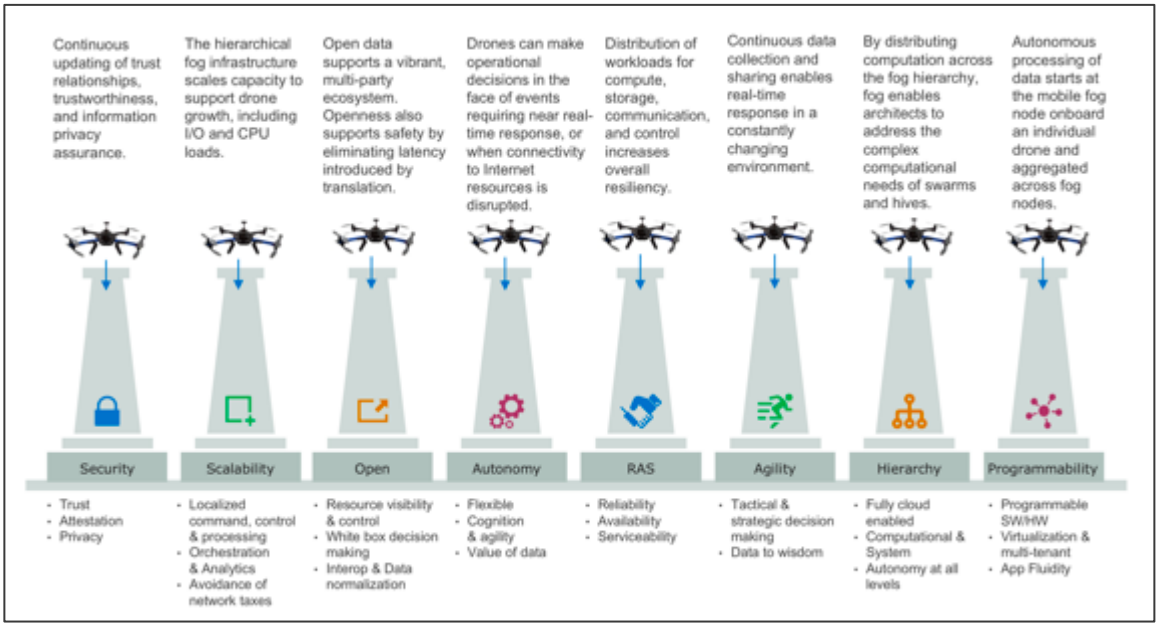

Figure 7. OpenFog Reference Architecture pillars mapped to the Aerial Drones for the Supply Chain Delivery Use Case.

- **Security** –Aerial drone systems can pose a threat if their operation is compromised by a malfunction or malicious tampering. The fog computing hierarchy enables architects to distribute and layer security mechanisms across the infrastructure. Fog-enabled drones can detect and isolate anomolies in other fog-enabled drones. behaviors to isolate drones computing adds layers of security and on-board intelligence or self-awareness to rapidly identify platform

and system level anomalies so security systems can engage and repair the affected systems. Fog also facilitates privacy, so critical data like mission logs, video streams, and cargo manifests won't fall into unintended hands. Drones with onboard fog nodes can also take individual authentication and hardware root-of-trust and apply these functions to a community of drones to enhance overall security capabilities.

- **Scalability** – It is important for aerial drone networks to have scalable performance and capacity in order to handle a growing amount of work. Fog computing provides the interoperability between the aerial drones and the ground base systems. For consistent performance, communication and maintenance of service level agreements, it is important for the drone fleet and ground based systems to scale the infrastructure equally according to the the load. A load may be I/O-based (for example, dealing with rate of transmission, reducing latency or managing high priority data download), or CPU–based (for example, managing an increase in computational requirements due to local analytics or data procesing). As drone services grow, the fog infrastructure scales its capacity as well.

- **Open** – Interoperability is key to rapid innovation in drone-based applications and services. Fog computing establishes open standards that provide the services critical to scale and interoperate, as many different stakeholders, vendors and suppliers will want to provide hardware and software for aerial drones and associated ground support systems.

- **Autonomy** –Drones equipped with fog nodes incorporate autonomic functions defined by the fog infrastructure, in order to manage, orchestrate, and ensure proper operation. This autonomy is critical when connectivity to the cloud or the ground support

network is limited, intermittent or lost, or in the event of operational control events requiring near real-time response.

- **Reliability/Availability/Serviceability (RAS)** – Aerial drone support systems are often mission critical, with stringent availability requirements. This means the hardware and software must be highly reliable, and the support systems that configure and maintain them must be very efficient. Fog computing infrastructure provides higher levels of RAS in order to protect data integrity, ensure that unit or system level anomalies are rapidly detected, isolated, and repaired.

- **Agility** – Similar to automated driving and robotics, aerial drones with onboard fog elements have the ability to automatically control and configure themselves and potentially other nearby or community fog nodes. In order to support high-scale aerial drone operations, both aerial and ground systems must adapt in the face of rapidly-changing requirements and applications.

- **Hierarchy** – Fog networks support a hierarchial system, representing local, neighborhood, and regional levels, efficiently divides computational and other tasks. This enables autonomy at all levels. It also enables architects to address the more complex computational needs of paired, swarms and hive drones.

- **Programmability** – Ground support networks as well as other nearby drones add much of their value through services. These services are specified and deployed through application programs and software. The fog nodes providing the support must be programmable to allow for continuous software innovation, updates and enhancements via interfaces from ground systems as well as through near field communication with other drones during coordinated operation.

11 Architectural Considerations

Drones can be designed and programmed for an endless variety of tasks. But in order to create a fog infrastructure that enables aerial drones to execute supply chain delivery tasks, it is important to understand how drones work alone and in concert with each other. The variability of actions and interactions can be daunting from a traditional design and programming perspective.

Fog computing provides an infrastructure that is optimized for variability and even unpredictability. Table 2 describes drone behaviors that apply to all drone types.

Table 2. Behavior, Social, and Organizational Models

Behavior Models: describe the way that drones interact with the environment and each other	
Solo	How a drone operates alone and on its own merit
Paired	How two drones operate together, coordinating work, services or to fulfill a singular purpose
Swarm	How multiple drones operate together to deliver a collective or coordinated behavior
Hive	How drones which may be collocated but do not have the same purpose; a hive may contain multiple solo, paired, and swarming drones
Social Models: describe the way that drones observe and associate with the environment and with each other	
Solitary	Describes a drone or drones that operate in isolation from other drones. These type drones will ensure that their solitary operation will continue and may coordinate with other drones for safety, space, speed and other constraints.

Social	Describes a drone or drones, which operate in a way that allow their incorporation into a community for coordinated operations.
Isolated	Describes a drone or drones that operate in a solitary fashion, completely isolated and unaware of other drones. These type drones will not join a community or group for coordinated operations
Accessible	Describes a drone or drones which are accessible for coordination or direction from other drones
Restricted	Describes a drone or drones that operate with impunity and complete isolation, without any communication or acknowledge of others.
Organizational Models:	
Homogeneous	Describes an application in which the operational and behavioral models are similar or identical. In a supply chain, an example would be a hive of aerial drones of the same type (owned by a single company) delivering food to customers.
Heterogeneous	Describes an application in which the operational and behavioral models are a mix of types. In a supply chain, an example would be a package delivery that requires some part of the delivery route to use an aquatic drone, which then hands off the package to an aerial drone for the final leg of the journey.

Functions of Individual Fog Nodes

A fog node can be as small as a chip or as large as a server. It can be stationary or mobile. It can be on the ground (or even below ground) or in the air. It can be built for harsh conditions, such as extreme temperatures, corrosion, vibration, extreme humidity, etc.

Each aerial drone is defined as a fog node, which means it has all of the basic functionality of an individual fog node. These functions can include:

- Compute, network and storage (these are generally the minimum capabilities in a fog node).
- A larger fog node (which could be in a larger drone or in a ground support infrastructure device) can also include functions like communications, control, security, acceleration, cognition, and analytics.
- Properties provided by these functions are low latency, security, scalable performance, reliability, programmability, agility, and autonomy.
- Modularity to support scalability and agility of functions by adding or upgrading hardware and software modules.

Topology

As shown in Figure 8, a fog-based infrastructure is hierarchical. This hierarchical approach is critical for achieving real-time decision making.

Usually, this is a three-level hierarchy of low, middle and upper level fog nodes. Note: This hierarchy doesn't reflect the physical locations of the nodes, but loosely correlates to the logical proximity to the cloud, e.g., the fog node on a drone is a low-level node.

- Mobile fog nodes on drones communicate with local fog nodes, which can be co-located or embedded in physical infrastructure

devices such as drone landing platforms and roadside infrastructure elements or street-level fog nodes).

- Local fog nodes communicate with regional fog nodes, which can be located in towers across supply chain delivery routes; in addition to tracking drones from segment-to-segment.
- Regional fog nodes on towers can also provide sensor fusion on data collected from other sensor arrays for weather, video, radar and other edge devices.
- Regional fog nodes also aggregate data to forward to—and download data from—multiple clouds, which may be owned by fleet operators, private vendors, municipalities, air traffic control agencies, and other stakeholders.

Figure 8: illustrates a fog computing infrastructure for supply chain delivery in a smart city. The hierarchy helps ensure dynamic flight plans, air traffic control database updates, and up-to-the-second weather reports while the aerial drones are in transit.

Ground Support Infrastructure

In a fog hierarchy, the ground support infrastructure is defined as any fog-enabled structure, device or system with which the fog-enabled drone might interact. In essence, fog turns these ground support elements into a network of richly interconnected distributed intelligence. This provides enhanced situational awareness and coordinates operations across many drones and geographies.

Locations: Ground support infrastructure fog nodes can be anywhere, including: co-located with or embedded in devices on street corners and rooftops in smart cities and towers along flight routes.

Types and functions: The types and functions of fog-enabled ground support elements are numerous and varied, including sensor arrays, radio access points, roosts and postal boxes.

Ground fog nodes are the platform for all the sensors that monitor the airways and the radio transceiver ground stations for the Vehicle-to-Infrastructure (V2I) networks.

Ground control stations (GCS) can have many different functions. One is air traffic control. GCS capabilities can be integrated into drone landing platforms to direct drones landing and taking off from perches and roosts. Or a GCS can be a dedicated regional fog node that provides air traffic control for all drones traveling through a given airspace.

Design considerations: Ground support fog nodes do not have as many constraints as fog nodes on board drones—Size, Weight and Power (SWaP). Ground-based fog nodes can be much larger, heavier, and support higher power ratings, and are capable of higher capacity and performance.

Community Behavior

Community behavior is one of the great strengths of having a fog infrastructure supporting drone operations.

As long as an individual fog node is a member of the same management domain, it can also become part of a collective, as shown in Table 3. Note: a management domain has the top-level authority for a group of fog-enabled drones.

There are also different types of collectives enabled by fog, which provides greater versatility for delivery models (and more efficient use of drone assets).jj

Table 3. Drone Community Behavior

	Individual Autonomous Drones	Persistent Hive (members are always the same drones)	Dynamic Hive (members can add or remove drones)	Swarm
Management Domain #1	Drone 7	Drone 7	Drone 7	Can include members from one or multiple hives, as long as the hives are controlled under the same management domain.
	Drone 15	Drone 15	Drone 15	
	Drone 20	Drone 20	Drone 20	
	Drone 45		Drone 45	

The OpenFog RA recommends that every fog node that functions as a drone should also have the ability to be part of a hive or swarm. When it is part of this kind of collective, the OpenFog RA recommends incorporating "service modification and remediation." This means that

the drones can use the programmability (one of the pillars of the RA) to reprogram themselves in order to isolate or exclude the rogue or malfunctioning drone.

It isn't necessary for any drone know what is causing anomalous behavior in another drone in the collective; what is important is that the drones are capable of *inflight reprogramming* in near real time. Because time may be of the essence with fast-moving drones, this inflight reprogramming must be autonomous—that is, without requiring control or orchestration from the ground station.

Hives or swarms can be created by design or aggregate dynamically for a specific purpose. Members in this collective may be co-located or be brought in from different areas. They do not have to have a single function. For example, a drone designed for food delivery can be "commissioned" to be part of a hive or swarm to deliver medical supplies. This flexibility enables operators of drone fleets to utilize drone resources more efficiently and better respond to varying loads and unusual flight conditions.

Multi-tenancy for Drone Fleets

Fog nodes onboard drones and fog nodes that are part of the ground support infrastructure have the capability to support multi-tenancy. This means that fog nodes may be owned by a specific entity (such as a delivery company, government authority, service provider, etc.), and application software on those fog nodes may be owned by many other stakeholders on a time-sharing basis.

The owner of the fog nodes is referred to as the landlord. The various stakeholders sharing the capabilities of those fog nodes are called tenants. Typically, many different tenants can simultaneously share

the computational, networking and storage resources of a landlord's fog node.

The application software running on all fog nodes in the hierarchy serves stakeholders with widely differing concerns. This means that the requests for resources from the shared infrastructure are highly diverse. Multi-tenancy in fog manages this variability. For example:

- Tenants must be assured that they receive an agreed-to share of the fog node's resources; fog ensures that no single tenant gets more than their negotiated share of the fog resources
- The data model in multi-tenant fog is complex, because some data is to be held in strict privacy by a single tenant, while other data needs to be selectively shared with other tenant processes that have pre-authorized permission to use it; fog protects all tenant software and data from unauthorized modification or cross-disclosure, such as security attacks that spread across tenants or security gaps that allow data leaks between tenants

Table 4 shows how multiple stakeholders become multiple tenants on a fog node, and examples of the applications each tenant might run.

Table 4. Examples of Multi-tenancy Applications for Drone Fleets

The mobile fog node on the drone is typically owned by a drone fleet operator, who is contracted by landlord. Following are examples of the tenants that might be sharing applications on the fog node onboard the drones and what those applications are for:	
Drone manufacturer	• Managing the flight control system • Recording telemetry streams • Diagnosing problems
Fleet operator	• Tracking their assets • Billing

E-commerce company	• Coordinate fleet, swarm and hive operations
Owner(s) of packages in transit	• Track progress • Monitoring condition and handling of cargo • Notify the sender and recipient of delivery progress
The owner(s) of the ground support infrastructure	• Maintain V2I links • Providing identification and authentication of drones in flight
Air traffic control authorities	• Monitor drone proximity to ensure collision avoidance • Manage and update flight plans
Public safety authorities	• Certify that drones and cargo are safe and have not been hijacked or diverted in flight
Package recipients	• Authorize and accept packages as required
The ground support infrastructure is typically a hierarchy of fog nodes, owned by a municipality's smart city division or a service provider, with a similar multi-tenant plan. Following are examples of the tenants that might be sharing applications on the fog-enabled ground infrastructure and what those applications are for:	
Manufacturer of ground fog nodes	• Monitor status • Maintain recording logs • Manage hardware or software problems
Owner of the ground fog nodes	• Manage capacity • Maintain V2I links • Set security policies • Perform sensor fusion

Communications network operator	• Manage the fog node interconnect with the Internet
Drone fleet operator	• Perform ground coordination of the fleet
E-commerce company	• Track and coordinate all delivery missions in the airspace
Air traffic control authorities	• Perform precision location of all drones in the airspace • Predict and avoid collisions • Verify all drones are following their flight plans
Public safety / homeland security	• Monitor for unsafe conditions • Verifying the cargo on each drone matches its manifest • Monitor for suspicious drone behavior or intrusion into controlled airspace • Take countermeasures to disable rogue drones

Security

Adding security features (such as encryption and anti-cloning chips on multiple sensors) will increase the cost of the drone. Downloading security credentials, patches and updates from the cloud to the drone can consume valuable bandwidth and take time. These approaches may result in making security compromises or delaying security response.

One of the major concerns about aerial drones is that they can be hacked or hijacked. For example:

- How do you protect the secure operation and onboard data of an aerial drone?
- How do you identify a compromised drone?
- How do you immobilize or rapidly recover a drone that is compromised or under control of an unauthorized entity?
- How do you ensure that drones cannot be hacked and rerouted to steal the drone and/or its cargo?
- How do you check drones to make sure that they haven't been loaded with unauthorized, potentially dangerous cargo?
- How do you stop a drone from going rogue (either because of malfunction or malicious intent) and causing damage in the air or on the ground?

At the base functional level, a fog node on the any type of drone can handle security, without adding complexity, size or cost to any other drone parts. Fog computing will ensure not only the security of the data throughout its lifecycle, but also the transmission, receipt, acknowledgement, provenance and chain of custody of the data. With these functions covered, a drone will have the confidence to act upon an event with the assurance that the information is valid and the outcome is predictable.

A fog node on board an aerial vehicle can take care of security updates even in mid-flight. Community-based access to other fog-enabled drones can also enhance security through shared authentication. Aerial drones can also work as a hive or swarm to provide a perimeter defense against hackers.

Screening on the Ground

A drone can be physically inspected using several different screening mechanisms. This provides assurances that a drone is carrying the

expected cargo, and not carrying anything dangerous. The screening process to validate this is similar to an airport checkpoint.

Fog computing enables architects to design screening solutions that keep data local. This is important for two reasons:

- Securing screening data: Screening captures extremely sensitive information. Keeping this data local helps ensure that the data isn't seen by unauthorized agents or intercepted for malicious purposes.
- Performance: Drone screening must be performed at very high speeds which requires low latency processes. Fog computing keeps these processes local in order to support deep drone scanning involving multiple screening devices—including analytics and sensor fusion—at high speeds.

Following is a sample scenario using a device called a fog-enabled screening perch.

The drone, with its cargo, is placed or lands on some type of conveyer belt where it is locked in place. It is moved through a number of screening devices, such as:

- A rotary x-ray scanner
- A rotary millimeter wave scanner for radio screening
- Weighing scales (to verify the weight is as expected)
- Software validation (to verify the software running on the drone isn't compromised)

Once the drone moves through all the screening sensors, the local fog node implemented in a screening device performs analytics and sensor fusion functions to determine if the drone is carrying what its manifest

says it is, and that it isn't carrying any unauthorized or dangerous payloads.

If the drone clears all screening checks, it is released from the conveyor belt and allowed to launch.

The Air Traffic Control authorities and fog-enabled GCS are also updated with new screening credentials that authorize this drone's safe passage along the next segment of its mission route.

Should something dangerous or unauthorized be detected during the screening, the conveyer belt could continue to retain the drone, and move it to a place of isolation for further investigation, or even into a bunker if the payload is identified as dangerous (e.g., chemicals or explosives).

In addition to this level of security, the fog computing hierarchy can be used to create virtual boundaries between regions of higher and lower security criticality, as discussed in the next section.

Screening in the Air

The sensors and analytics systems that provide screening for drones in transit have low latency and high reliability requirements. Without a fog hierarchy, screening drones at high scale would be extremely difficult to achieve.

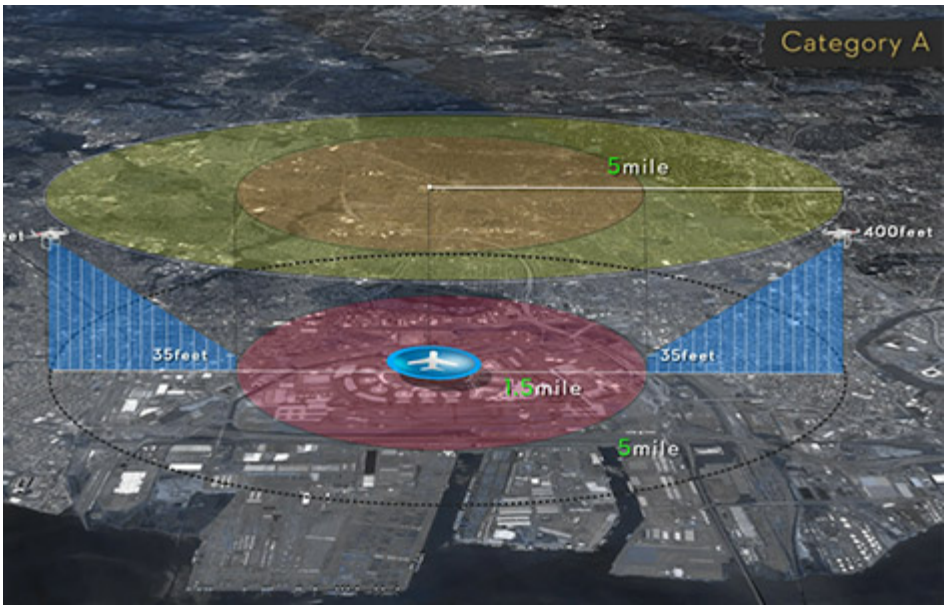

Figure 9: Without fog computing, it would be impractical if not impossible to screen a high volume of drones moving at high velocity through airspace requiring different levels of security clearance.

Figure 9 shows a fly zone with several concentric virtual boundaries of increasingly tight security. The concentric rings graduate from yellow to blue to red (where red requires extreme security or credentials to enter). These boundaries are architected with networks of sensors and fog nodes that can have the following functions:

- Screening checkpoints
- Detection of unauthorized drone crossings for each virtual boundary
- Countermeasures to neutralize unauthorized drones

If a drone attempts to fly past its authorized boundary, the fog-based sensor networks would immediately detect it, and countermeasures can be activated to neutralize and/or capture the drone.

Note: Earlier, we described how a truck could be used as a mobile fog node for carrying many drones for local and regional deliveries. Similarly, a military vehicle could be turned into a mobile fog node for dispatching screening checkpoints wherever they are needed. With the fog computing infrastructure in place, screening checkpoints can be set up dynamically, and communications with drones, base stations, and Air Traffic Control authorities established in near real time.

Communications

Near-Field and Long-Distance Communications

In fog computing for drones, there are both near-field and long-distance communications considerations (Table 5). Communications with satellites and base stations are examples of long-distance communications. Communications between drones in proximity to one another is an example of near-field communications.

The OpenFog Reference Architecture recommends that long-distance communications are always used as a backup to near-field communications. The proximity of drones to each other should never be controlled by long-distance communications, as the latency is too great.

Table 5. Communication Modalities

Wireless ground stations	3G / 4G / 5G Cellular Wi-Fi Microwave backhaul Aircraft radio standards GPS / GLONASS / Differential GPS Free Space Optical Data Links
Sensor arrays combining many modes	Video (Visible, IR, Thermal, LIDAR) Radar Weather / atmospheric

	Acoustic (microphones, seismic) CBRNE detectors
Fog computing resources	Sensor fusion Analytics Computation accelerators (FPGAs, GPUs, DSPs) Local decision making Hierarchical storage High performance / high reliability

Data and Analytics

Data at Rest/Data in Motion/Data in Use

In any infrastructure model, including edge-to-cloud, drone systems must have the ability to collect and generate data. The advantage in fog computing is that compute and analytics capabilities are distributed across the fog hierarchy. This enables architects to design supply chain delivery applications that require the autonomy, programmability and other capabilities discussed in this use case. Following are examples of how data might be handled in a fog computing hierarchy.

There are two types of data: data at rest and data in motion. Data is exchanged drone-drone, drone-infrastructure, and up and down the hierarchical levels of the fog-based ground support infrastructure.

Data at rest can include things like mission instructions, performance logs, video archives and "black box" recordings. In an edge-to-cloud model, data at rest is probably stored in the cloud, or in off-line storage associated with the cloud.

With fog, architects can also store data at rest in:

- The memory of a drone's mobile fog node
- The main memory or storage array of an element of the ground support fog hierarchy

The ability to store data at rest in layers of the fog hierarchy has several advantages: first, of course, is that it can make retrieval faster. It can also strengthen data security and privacy. Security can be enhanced with the ability of fog nodes to provide strong encryption before the data is stored.

Data in motion is being actively transported between destinations. Examples common in supply chain delivery include streaming video, drone position traces, and radio propagation measurements.

Let's look at video data, which can be used by:

- Fog-enabled control and telemetry systems between the ground
- Fog-enabled drones to monitor their situation and performance and provide control inputs to operate them

In order to support near real-time decision making based on streaming video, the on-board fog node provides some degree of analytics, as well as pre-processing to compress the data and forward it to the ground support fog nodes for more detailed analysis.

Nodes lower in the fog hierarchy may perform the simpler pre-processing functions on the video stream (for example, contrast enhancement or feature extraction), while fog nodes in the upper level of the hierarchy may perform the more difficult analytics algorithms (like pattern matching and object recognition).

The OpenFog Reference Architecture recommends that both data at rest and data in motion meet basic requirements, such as:

- Appropriate security / cryptography processing must protect the data
- The fog hierarchy must have adequate bandwidth in its links
- Main memory and mass storage devices must accommodate the expected load; the storage must be adequately reliable, perhaps using RAID and similar redundancy techniques, as data corruption or loss can have strong consequences (regulators will require minimum data retention and data custodial discipline, especially for data associated with accidents or other abnormal events)

Processing Algorithms and Analytics

One of the advantages of fog computing is that processing algorithms and analytics are distributed throughout the fog hierarchy. In an edge-to-cloud model, depending on cloud-based analytics can be costly and add unacceptable latency in decision making. Fog puts the computation, networking and storage capabilities at optimal levels of the hierarchy, helping to achieve the best possible performance, security and efficiency.

One example is video analytics. Video signals from both the drone-mounted cameras and the surveillance cameras that are part of the ground support infrastructure are processed by powerful CPUs or GPUs. These analytics can provide near real-time information such as:

- Estimating a drone's speed and heading
- Determining a drone's position relative to other drones or obstacles

- Detecting objects
- Aligning the drone to targets for docking or landing
- Detecting the spatial relationships between multiple drones in paired, swarm or hive operations

Many different types of analytics operations can be applied all kinds of sensor data streams. This is just a sampling of what's possible:

- Microphone sensor data can be processed by sound classification algorithms to locate drones and determine their operational status by their sound
- Radar / LIDAR sensor data can be processed by analytics algorithms that can determine the state of an airspace
- Data from radio sensors can determine the frequency and signal strength of a drone's transmission to improve the performance of the radio network
- Analytics on the drone's flight computer can process its inertial navigation system and positioning system data, and pass these on to higher-level analytics in the drone's fog node for additional processing
- Data aggregated from all of the sensors in a drone fleet (a fleet of drones is owned by a single operator) can be analyzed by higher-level fog nodes; these analytics can help coordinate the collective operations of a fleet and improve its efficiency.
- Individual sensor readings can be unreliable—due to noise, interference, intermittent network connections, calibration errors, etc. By fusing the readings of multiple sensors, perhaps of different types, a fog-enabled GCS for air traffic control has a much more trustworthy picture of the situation in the air.

12 Testbed Considerations

OpenFog testbeds are designed to verify specific use cases and to test aspects of the fog architecture. The use case content in this document can be a reference and the architecture and communication chapters can provide a guide for design and implementation of fog systems.

The composition 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.

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.

13 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 through adherence to the architectural principles.

14 Next Steps

The OpenFog Reference Architecture 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, and 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 that 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 Reference Architecture.

We welcome comments on this document and our work. To submit commentary or for more information, please contact info@OpenFogconsortium.org.

15 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.

16 Authors and Contributors List

Authors	Contributors
Chuck Byers, Cisco Systems	Evan Birkhead, OpenFog Consortium
Katalin Bartfai-Walcott, Intel	Judith Kelley, OpenFog Consortium
	Ryan Gentry, Intel

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

17 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/