Fog Use Case Scenarios

Use Case: Process Manufacturing – Beverage Industry

Vertical: Smart Factories

An OpenFog Consortium Architectural Use Case

1 Snapshot: Process Manufacturing – Beverage Industry

WHY FOG

Why is fog the best architecture for this use case?

Fog enables systems in smart factories to connect with other external systems, while alleviating concerns about security, confidentiality and interconnectivity. Fog nodes provide functions of encryption, anonymization and translation – even for legacy and less powerful devices. Fog nodes are also suitable for handling dynamic connections between networks that differ in Quality of Service (QoS) and traffic rate. Fog can support real-time control functions.

WHICH FOG PILLAR

Which fog pillar best describes this use case?

The Autonomy and RAS (Reliability, Availability, Serviceability) pillars of the OpenFog Reference Architecture are most crucial to the beer brewing use case. The Autonomy pillar provides proactive detection and repair to avoid production line stoppages; The RAS pillar improves availability of critical functions.

VALUE

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

Fog can help process manufacturers to realize business value in three areas: 1) Domain-specific brewing secrets can be replicated through fog-based digital twins; 2) Fog-enabled remote maintenance makes more efficient use of operational expenditures; and 3) Fog-based resource sharing enables brewers to respond to sudden fluctuations in demand and enables resources to be provided to partner businesses and multi-tenant control system models.

CLOUD & EDGE

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

Fog augments cloud and edge by enabling low-latency cooperation between on-premise devices in the smart factory. For example, fog nodes can aggregate and compress big data onsite, then send to the cloud to carry out analysis. The cloud can provide intensive site monitoring using the fog network. When an anomaly is detected, fog nodes can automatically send relevant logs to the equipment vendors.

2 Table of Contents

3 Introduction

Note: The preamble section of this document (pages 3 through 11) is common across all OpenFog use cases. It provides descriptions and reference points for fog architectural attributes and properties. The specifics of this smart factory use case begin 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.

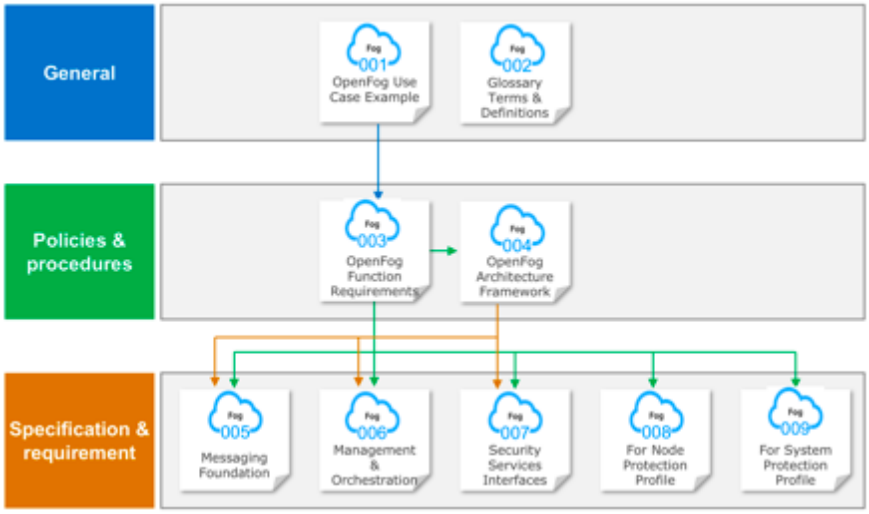

Figure 1. Hierarchy of OpenFog Consortium specification documentation

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: Process Manufacturing – Beverage Industry

Use Case: Process Manufacturing – Beer Brewing

Vertical: Smart Factories

Executive Summary

Increasingly, consumers want to buy high-quality products at a low cost. Manufacturers are making factory floor investments to meet the demand, keep the quality high, and produce a positive business outcome for the manufacturer. Producing food and beverage products presents particular manufacturing challenges. Take the case of craft beers, one of the fastest-growing alcoholic beverage segments. Craft beers appeal to consumers who are interested in trying new types of beers that they don't associate with the mass-produced beers. Demand fluctuates with the predictable spikes (such as the Super Bowl) and unpredictable (unseasonably hot weather).

In order to respond to consumer demand, craft beer production must be extremely flexible and adaptable. Yet quality control must be consistent. Craft beer manufacturing typically takes place in a small/intermediate volume factory and requires a high mix of different product types and recipe variants.

Producing craft beer involves dealing with natural ingredients such as malt, hops, and brewer's yeast. The manufacturer must deal with the idiosyncrasies of nature in how the ingredients act independently and with one another. On the shop floor, skillful adjustment and precise (yet flexible) production controls are required to create a unique product with little variance. In addition to ensuring the uniformity of the beverage, the manufacturer's processes must incorporate the technique of the crafter.

Smart factories that utilize IoT and fog computing is the way to ensure a consistent supply and reliable quality for craft beer production. IoT has several advantages in a fog environment:

- IoT technologies enable the techniques of crafters to be digitized and therefore reproduced with a consistent level of quality and uniformity. Large numbers of sensors and actuators throughout the brewery keep the production processes under tight control.
- IoT enables greater flexibility and adaptability in production processes, enabling a single factory to produce many more varieties of beer without massive changes to the production line, which can drive up costs.
- On the maintenance side, IoT sensors and applications can monitor and detect signs of product quality degradation in real time, facilitating a quick response.

Fog computing provides the foundation for the IoT-based smart factory. A fog-based infrastructure is capable of collecting the wealth of sensor data distributed among brewery equipment and aging vessels. In a fog environment, even the techniques used by experienced beer crafters can be digitally recorded and captured as data, to be repeated exactly in subsequent batches. Fog nodes can also store proprietary manufacturing recipes and control methods.

Challenges

- Maintain quality and consistency working with natural ingredients (whose behavior is idiosyncratic by nature).
- Reduce variances that can be introduced by manufacturing equipment/machinery and processes.
- Improve factory productivity /avoid work stoppage by discovering signs of mechanical problems before they affect production.
- Optimize factory investments for a growth business that is characterized by turbulent fluctuations in demand and high product mix.
- Minimize the impact of resource surpluses and shortages.

Solution

- The fog-based IoT smart factory reliably creates high-quality, highly-consistent products with minimal involvement from humans.
- By capturing the quality of products currently being made by manufacturing equipment/machinery in real time, and fine-tuning production parameter settings for the subsequent processes, the quality of products is improved in real time throughout the entire production process.
- A connection between manufacturing equipment and the systems of equipment manufacturers and maintenance providers allows signs of failure to be analyzed and the stock status of replacement parts to be checked. As a result, schedules can be adjusted quickly, and problematic parts can be replaced without interrupting the production process.
- By allowing selected production resources to be shared across factories and corporations, shortages in production capacity that arise due to sudden surges in demand can be alleviated by using the resources of other factories or organizations to reduce the risk of lost business opportunities.
- The development of new services to share surplus internal production resources caused by fluctuations in demand with other factories or organizations will generate manufacturing service models.
- The fog-enabled smart factory can efficiently integrate the supply chain, ordering ingredients and supplies just-in-time, and keeping distributors apprised of the progress of each production run.

Technology

- Fog nodes that capture and analyze actions of expert brewers to create digital twins.
- Autonomous systems that include intelligence from multiple fog nodes, involving sensors and actuators at all phases of production.
- Simple, flexible and secure interconnections between equipment and maintenance systems using fog nodes.
- Construction of autonomously-distributed resource pools using virtualization technology.

Use Case Overview

The following is an example of a conceptual implementation of fog computing for producing beer in a smart factory. The purpose of presenting this use case is to promote more architectural conversations about fog computing use cases for the manufacturing industry.

Introduction

Unlike industrial production that uses inorganic resources, the food and beverage industry has to deal with the products of nature. In the case of craft beer brewing, this involves ingredients such as malt, hops and brewer's yeast. Creating a uniform product is more difficult when you're working with organic material.

That challenge is magnified by the fact that brewing craft beer involves brewing many different varieties in small quantities. This makes it difficult to select optimum parameters for different varieties in a range of different conditions, and dynamically adjust for process variations.

Various elements, for example the temperature and moisture in upstream and maturing processes or the state of yeast and hops, must be adjusted alongside other elements such as ingredient quantities, proportions, process time and temperatures, maturation periods, and so on.

This delicate process depends on the ability to apply the experience of crafters to scale to higher volume production without sacrificing quality, to work with manufacturing partners to deal with fluctuating demand, and to make business and operational improvements. As a result, brewers cannot simply flip the switch to speed up production.

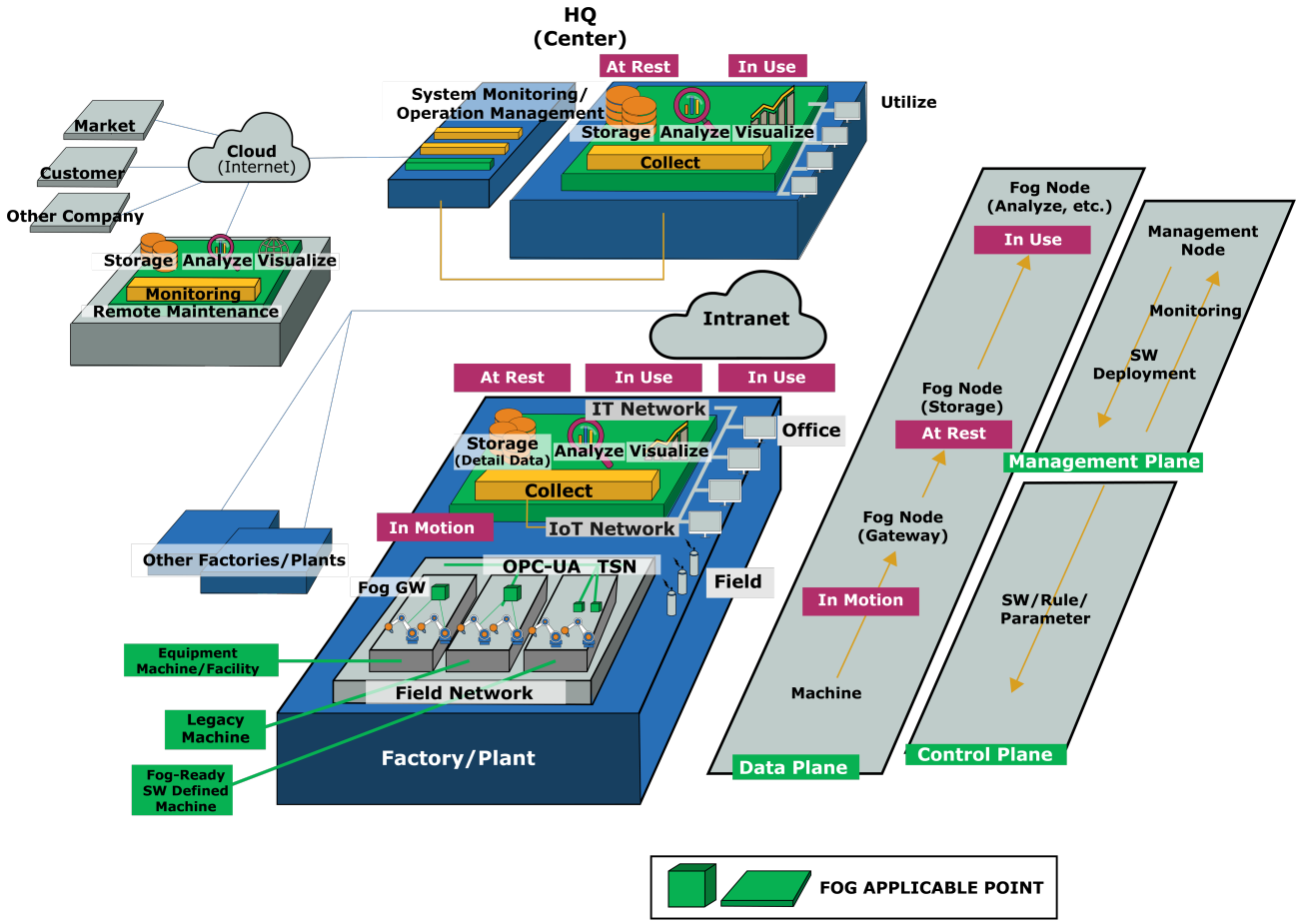

Figure 2. A typical implementation of fog computing across a smart factory spans production processes as well as business activities.

Incorporating advanced fog-based IoT technology into craft beer manufacturing plants creates smart factories that can accomplish the following:

- Reproduce the skills of a crafter through the creation of a "digital twin," using data collected from sensors on vessels and other production tools. (A digital twin is a digital replica of physical assets, processes and systems that provides both the elements and the dynamics of how an IoT device operates throughout its life cycle.)

- Create a stable inventory supply by securing materials based on demand forecasts, optimized inventory management, and coordination of the supply chain and production resources.
- Achieve real-time maintenance response based on automated, predictive detection of manufacturing equipment failures and inventory/order management of maintenance parts.
- Ensure quality monitoring and low-latency response to quality degradation, reducing losses caused by defective products.

The local intelligence of fog computing supports smart manufacturing on the shop floor by enabling materials to be secured based on demand forecasts, optimized inventory management, and advanced detection of production equipment problems for preventive repairs.

Interoperability

In the fog-architected beer brewing scenario, interoperability plays an important role. Here are a few examples:

1. Remote maintenance can be a key factor in improving yield rates and production efficiency. Remote equipment and systems using different protocols and data formats (including legacy production machines) can be connected easily, thanks to the interoperability enabled by OpenFog Reference Architecture.

2. To meet fluctuations in demand, beer brewers can dynamically create virtual groups of fog nodes across multiple fog networks. This enables multiple factories to share production equipment. Interoperability is required to create these virtual groups. Platform-level, modular interoperability enables legacy systems' resources and properties to be exposed as sharable smart objects to the information and device models in virtualized environments. This includes common interaction procedures.

Business Case

Consumer lifestyles and diversified preferences mean that, in order to meet market demand, food and beverage producers must be able to cope with small-quantity, large-variety products, along with product lifecycles with large fluctuations in demand periods and quantities. This is particularly true for craft beer. Batch-to-batch product consistency is especially important in this market.

In the manufacturing industry, dealing with these trends is a major management and investment strategy challenge. In order to operate profitably and meet aggressive growth targets, manufacturers must be able to:

- Secure materials based on short-term demand forecasts
- Maintain consistent, high-quality manufacturing operations
- Optimize inventory management
- Maximize human resources and hire/retain highly-skilled personnel
- Invest in production facilities with long depreciable lives while considering fluctuating demand
- Prevent unplanned work stoppages due to unexpected mechanical failures.

The inability to hit every one of these targets consistently can mean business failure - even if you have a fantastic product. There are numerous examples of the immediate deep financial losses caused by the production of defective products, and the even deeper loss of future business if the manufacturer's reputation doesn't recover.

Factories may create a small volume of expensive products or a large volume of products in a short period of time. If, for example, a factory produces 1,000 units of an expensive item every minute, producing

defective products for even one minute could result in tens of millions of dollars of financial damage. What's more, as the number of personnel required for inspections increase, labor costs and other related expenses will also rise. To minimize losses and to manage costs, a fog-based IoT framework is required.

Fog and IoT also help craft brewers solve the problem of replicating and automating the knowledge and techniques of their crafters. This is not only a productivity requirement, but it helps secure the techniques as company-owned intellectual property or proprietary assets. This level of automation enables the factory to rapidly and consistently scale its production, in case demand is seasonal or surges for a successful product.

Applications

Fog is a key enabling factor in the software-defined smart factory.

Factories have high levels of confidential information – beer recipes, production processes and failure logs, just to name a few. Further, many manufacturers want to store the data on their own premises, not in the cloud. In these cases, data analytics needs to take place on the device side without sending data to a remote cloud. Moreover, analytics on the fog node at the local site alleviates the increased demand of the bandwidth to the cloud. Analytics on fog nodes onsite also solve time constraint problems, because the round-trip time lag between production line sensors, to the cloud, and back to production line actuators isn't needed – and the local fog node can react and make decisions in real-time, or, in challenging cases, within tens of milliseconds.

Digital twins presents an ideal case to utilize human resources efficiently, from a time and place perspective. In the digital twin scenario, one operator can monitor and manage multiple devices in

multiple places through digital twin simulation. This enables the fog-based infrastructure to learn from past history, provide real-time control of current production processes, and extrapolate the production parameters that may be needed for future production runs, all in a single, integrated system.

Fog can orchestrate multiple business processes, such as beer brewing and logistics. The hierarchical structure of fog nodes forms dynamic groups to exchange the needed information for efficient collaboration. For example, a company can fill a temporary shortage of production capacity by tapping into another company's idle equipment. Intellectual property such as special recipes can be kept secure. The products which need to be moved from one to the other company can be delivered by a logistics company in near real time, which improves the time required to ship final products.

Following are examples of how fog computing impacts architectural considerations for creating Smart Factory applications that drive business and technical benefits in craft beer brewing:

Creating a Digital Twin of the Crafter

It is well-known that wines produced in the same region undergo huge changes in their characteristics depending on the year that they were produced. The same is true in the world of craft beer; there are innumerable variations. Natural ingredients such as barley and hops can have wildly differing production areas, harvest periods, flavor and texture characteristics, as well as variables such as the environmental temperature and humidity in the brewing process.

What ratios should be used? How long should it be fermented? When should it be stirred? The answers vary with every batch.

The "crafter" determines the answers to such questions. However, not every small to medium-sized craft beer breweries have a sufficient number of crafters, so they may not be able to scale to brew the quantities and varieties that they would like. A digital twin can help.

The multitude of sensors across the fog-based production area are constantly generating data. Some of this data could replicate the senses of experienced brew masters. For example, sensors could record the carbon dioxide production rate of the fermentation processes, noting the color of intermediate products along the line or performing chemical analysis similar to the brewer's sense of smell. The amount of the data generated by the large numbers and types of sensors in the factories is huge, and it's difficult to collect and store all the data and send them out to the cloud layer. The brewer needs to choose which data needs to be collected, stored and sent in order to save the storage capacity and bandwidth.

As shown in Figure 2, fog nodes are used to collect and aggregate this data. The on-board storage and compute capabilities of the fog node analyze and compare conditions to the parameters defined by the crafters locally, and can instantly control actuators that influence process parameters like temperature, pressure, flow, time, etc.

Furthermore, analysis of big data collected through this process is also useful in creating new products. It can be converted into new recipes and rules for brewing new beer flavors . As the data itself becomes an intellectual asset , the fog infrastructure will provide robust security to block external leakage and keep potentially valuable production details secret.

There is another business advantage to creating the digital twin: As routine production tasks become largely automated, skilled crafters have time to devote themselves to creating new products.

Improving Yield Rates

Smart factories use many different types of production equipment. If this equipment could operate autonomously (individually and as part of a process) to adapt to the production situation in real time, it could improve yield rates and batch-to-batch consistency.

There are many ways that a fog environment could improve yield rates in a brewery. For example, as the new beer is boiling, it can be monitored by local fog nodes in real-time for color, temperature, acidity, specific gravity, sugar content, and so on. The kettle temperature and agitation can be automatically adjusted to keep the process within the brew master's control limits, ensuring that no batches are wasted. In this way, fog enables the brewery to realize just-in-time production.

Another example is to adjust the temperature of the fermentation vessels in response to CO2 production rates. This ensures the yeast action is on target, regardless of the ambient temperature or variations in yeast strain genetics. Again, by controlling quality, no time, energy, or product is wasted.

Beer packaging is an important factor that impacts overall business operations efficiency. After quality testing, beer has to be moved from the brewing vats to the bottling and canning assembly lines, and then labeled, boxed and shipped. This involves complex robots and discrete manufacturing processes.

Let's look at two in-depth examples where fog nodes harness local compute power to gauge and improve the performance of robotics machinery during the discrete portion of the beer production scenario:

Example #1 demonstrates how a single piece of machinery can execute a sequence of actions autonomously:

1. A piece of equipment drills a tapped hole

2. The position of the hole is examined by capturing it as an image

3. If the image shows that the position of the hole has deviated from the blueprint by a certain amount, the drill is automatically repositioned to the original value specified in the blueprint.

This is a fairly simple example of how autonomous assessment and correction of a mistake can prevent defective products. This particular action only involves a few steps, but it requires precision. It must be performed efficiently, but some latency in the process is not a critical concern. However, this changes if robotics equipment is used. Robotics require a more complex sequence of movements, even for a single task. Control of these movements must be performed within milliseconds, or, in the case of a motor drive control, microseconds. That's where fog nodes provide the compute power needed to ensure precision.

Example #2 requires coordination between machines. Assume that the machine that drilled the tapped hole is Machine 1. A machine that subsequently inserts a screw is Machine 2.

If the hole made by Machine 1 deviates from the specified position by more than a certain amount, the ID of the product concerned and information regarding the deviation is sent to Machine 2. Having received the information, Machine 2 adjusts to insert the screw in the deviated hole, preventing a defective product with an improperly inserted screw from being produced.

If Machine 1 makes an adjustment that affects the actions of Machine 2, we've introduced information sharing between the two systems. Now Machine 2 must act autonomously to act on this information, essentially learning and adapting.

Fog nodes on each piece of equipment collect real-time information from sensors. By keeping these types of actions local, it makes autonomous action more efficient, because the information doesn't have to be aggregated and sent to a data center for analysis and response.

A filtered version of the information is sent up through the fog hierarchy to the data center at headquarters for analysis. This supports future planning and longer-term improvements, as shown in Figure 2 on page 16.

Remote Maintenance

The brewing of craft beer involves linking many pieces of equipment throughout the preparation, fermenting, filtering, and bottling processes. Consequently, a failure in one piece of the equipment can shut down the whole production line. It can take a long time to locate and analyze the cause of the failure and repair the part or update the software.

The fog-based smart factory addresses this through predictive maintenance. Fog nodes collect data from equipment sensors as well as the connections between the sensors and systems. Although there are many data formats and protocols that sensors and systems must handle, because fog nodes normalize data, the sensors and systems can be easily connected without awareness of the different access methods of each system. Small deviations from normal sensor readings could indicate an impending failure (such as a failing pump

seal, bad bearing, or bottle-handling actuator problem), and enable the system to repair it before a catastrophic failure.

Additionally, fog nodes automatically select the most appropriate communication routes to equipment manufacturers and maintenance providers and securely send data to the maintenance system in real time. If the maintenance system detects a sign of failure once it analyzes the received data, the inventory is immediately checked for replacement parts and schedules are adjusted with little impact on production.

To monitor factory equipment remotely, and perform maintenance and repair, factory systems must be interconnected with those of the maintenance provider. However, due to information confidentiality and the complexity of connecting various systems (which utilize different protocols and communication methods), system interconnectivity has not advanced as much as one would expect.

As shown in Figure 3, fog computing provides both interoperability and security for data sharing. It also has local compute, storage and analysis to send more useful information to maintenance systems, which conserves bandwidth.

If there is a large amount of sensor data, it can consume bandwidth and slow production processes. Fog nodes contain compute, storage, analysis and control capabilities, which allows processing to be executed as close to the equipment as possible. Instead of raw sensor data, more useful information can be sent to the fog nodes on maintenance systems.

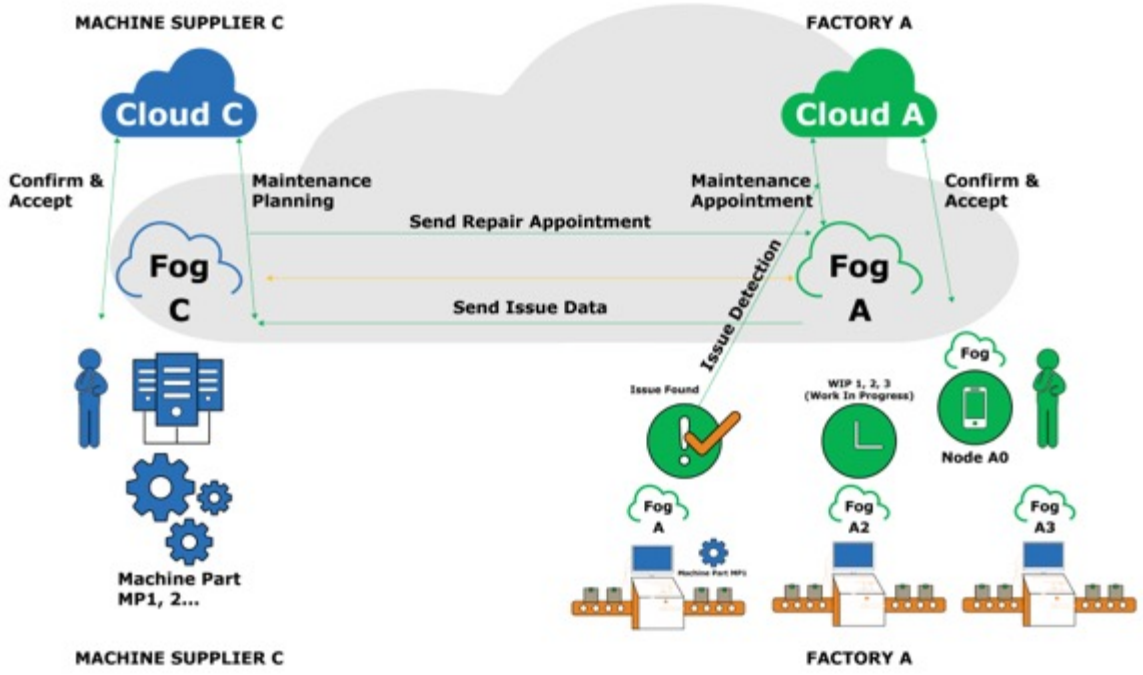

Figure 3. Remote monitoring, maintenance and repair of factory equipment requires factory systems to be interconnected with those of the maintenance provider.

Meet Fluctuations in Demand

As mentioned previously, craft beer is produced in small volumes and in a large number of varieties, which makes forecasting demand for craft beer quite tricky. Consequently, if there are resource shortages (personnel, equipment, facilities, etc.), craft beer brewers face lost business opportunities. The brewer can also lose money by overinvesting in these resources in order to manage infrequent demand surges.

Therefore, production resources and diverse customer needs (flavor, aroma, etc.) should be divided into the respective categories of (1) production capacity for the points of difference in beer, which are accomplished with the brew master's unique techniques and (2)

production capacity to fulfill general brewing demands (standardized common specifications, labeling, packaging, etc.).

A brewer can reduce overinvestment risk by using a distributed fog-based virtualization platform that can collect and analyze information on general production capacity. For example, fog analytics might conclude that large fluctuations in demand can be handled by supplementing the production capacity of an under-capacity brewery with that of a brewery with excess capacity.

Further, a distributed virtualized platform makes it easier to dynamically construct a virtual production line based on the brewery's resources. This requires the ability to share real-time data on production facilities and capacity, yield rates, processes, current production volumes, supply and demand forecasts, and so on. Large numbers of sensors distributed across the production line, combined with the capabilities of a hierarchy of fog nodes, accomplish this.

In each factory, fog nodes collect, aggregate and analyze the necessary data related to general production capacity. The virtual platform dynamically interconnects fog nodes between different factories. Essentially, the platform creates a virtual group of fog nodes that all try to produce consistent product, regardless of the specific factory they are working in. Then data collected by the fog nodes is sent back to the virtual platform for decision making on resource utilization across factories.

The interoperability required to create a virtual group of fog nodes across multiple fog networks takes place through the OpenFog Reference Architecture's standardized network communications, application frameworks, which leverages standardized APIs. The OpenFog RA also guarantees interoperability for information models for the production resource data and supply & demand information shared between fog nodes, allowing production resource information to be traded dynamically.

Much of the data in smart factories is highly confidential. Factory owners and supervisors have valid concerns about allowing data to be shared outside of the factory. But it's some data must be shared in order to work with partners in the supply chain. Fortunately, all fog nodes have functions to manage privacy and security. Fog nodes autonomously determine the content of data and send the appropriate data to only the required systems. The use of data classification (such as Open or Closed), encryption and VPNs can provide even more secure communication. Using OpenFog's advanced security capabilities, any brew master's recipe can be implemented on any factory's production line, without fear of unintended cross-disclosure of proprietary information in either direction.

It's worth noting that once the fog computing infrastructure is in place, it's easy to generate new income streams by creating a new application based on business scenarios for trading production resources.

Architectural Considerations

In the following section, we'll go into more details on architectural considerations related to maintaining quality and uptime in the smart factory using fog computing.

Proactive Maintenance and Repairs

Key Technical Challenges. Malfunctions in production line equipment need to be detected as soon as possible to prevent or minimize factory downtime. Detecting possible failures depends on systems internal to a piece of equipment. This may require communication between machines, which is even more challenging if one of those systems is older, legacy manufacturing equipment.

The ability to incorporate fog computing with legacy and new equipment is an important benefit to smart factories, which can continue to leverage their expensive investments in existing equipment.

Architectural Considerations. Many open and proprietary protocols are used for various legacy production machines or sensors. Fog nodes can work as interpreters or protocol gateways for these legacy systems. [The OpenFog-defined Application Programming Interfaces (APIs) help define this interoperability. – please reference the OpenFog Reference Architecture document.]

Sensors deployed on legacy machines collect the data and send it out to the fog nodes lower or higher in the fog hierarchy, depending on the action required. These can range from inexpensive temperature or flow sensors up to very sophisticated analytical instruments such as gas chromatographs, all integrated into a single automated production flow.

For example, you might design a fog node at the lower level of the hierarchy (perhaps located on an individual machine, collecting data from all of the equipment sensors) with an anomaly detection engine. If the fog node is also connected to an actuator, it could analyze the data, interpret the anomaly, and then could autonomously react and compensate for the problem or fix the issue. Or it can send the appropriate requests for service higher up the fog hierarchy (north-south) to the maintenance service provider. The appropriate security mechanisms can be provided at this stage, before the request is transmitted to third-party provider.

If this situation requires real-time decision making (for example, the need to shut down equipment before damage occurs, or adjust critical process parameters), fog nodes can provide millisecond-level latency analysis and action. The manufacturer doesn't have to route this real-time decision making through the cloud data center, avoiding potential

latency, queue delays, or network/server downtime which would result in industrial accidents or poor product quality.

Fog nodes higher up in the hierarchy can add more functionality, such as visualization of production line operation, monitoring the status of malfunctioning machines, modification of production planning, ordering ingredients from suppliers, and sending alerts to the appropriate people.

Communications Considerations. When retrofitting a brown field manufacturing plant, the communications protocol is provided in the OpenFog RA. In green field sites, fog-ready software-defined machines support time sensitive networking (TSN) or other broadly-used protocols, and can connect to an IoT network natively.

If the Operational Technology area is strictly protected from the outside world using physical firewalls, data doesn't have to be encrypted. But in systems that must interact with external entities, the hardware root of trust capabilities of fog nodes can ensure highly-secure communications and trustworthy processing and storage capabilities. Ideally, the security path needs to be established end-to-end.

Anomaly detection can occur at the fog node level. The detection engine is usually created at the upper layer, like the data center in the factory or the business center, by using the collected data from fog nodes. The created engine is then deployed to the fog node on the production line. Yet not all the data is sent to the upper layer because of the bandwidth. In some cases, the data is stored for a certain period of time, and this stored data is retrieved by the upper layer once an anomaly is detected.

The highest level of the fog network might be located at headquarters, connected to the factory via an intranet. If a fog node needs to communicate with other companies, it must be equipped with highly

protected interfaces.

Remote Maintenance

Key Technical Challenges. To remotely monitor equipment in the smart factory and to perform maintenance and repair, factory systems must be interconnected with those of the maintenance provider or equipment supplier. However, due to information confidentiality and the complexity of connecting systems which utilize different protocols and communication methods, system interconnectedness has not advanced as much as one would expect.

Fortunately, fog computing provides both interoperability and security for data sharing. It also has onboard compute, storage and analysis to send more useful information to maintenance systems, which conserves bandwidth. The multi-tenant capabilities of fog permit a single local fog node to run the mainstream production software, plus independent, isolated software provided by the suppliers doing local monitoring and predictive maintenance.

Architectural Considerations. A connection between production equipment and the systems of equipment manufacturers and maintenance providers allows signs of failure to be analyzed and the stock status of replacement parts to be checked. As a result, schedules can be adjusted quickly, and problematic parts can be replaced without interrupting the production process.

Simple, flexible, and secure interconnections between equipment and maintenance systems can be obtained using fog nodes.

Although there are many data formats and protocols that sensors and systems must handle, fog nodes convert that data to ensure interoperability. In a fog environment, sensors and systems can be

easily connected without awareness of the different access methods of each system.

Additionally, fog nodes automatically select the most appropriate communication routes to equipment manufacturers and maintenance providers and send data to the maintenance system concerned in real time. If the maintenance system detects a sign of failure once it analyzes the received data, the inventory is immediately checked for replacement parts, and schedules are adjusted with little impact on production.

Communications Considerations. When diagnosing equipment in remote locations or sending rich data for analysis to a remote maintenance system when a malfunction occurs, prioritizing bandwidth allows important data to be delivered in real time. The best way to prioritize bandwidth is to process and analyze more data at the lowest possible level of the fog hierarchy.

- High-load processing, such as machine learning based on rich data, would be performed at the data center.
- The administrator determines the destination of data for analysis and responsibility for action.
- Fog networks can also provide security for data that doesn't have to pass through the cloud.

Resource Sharing

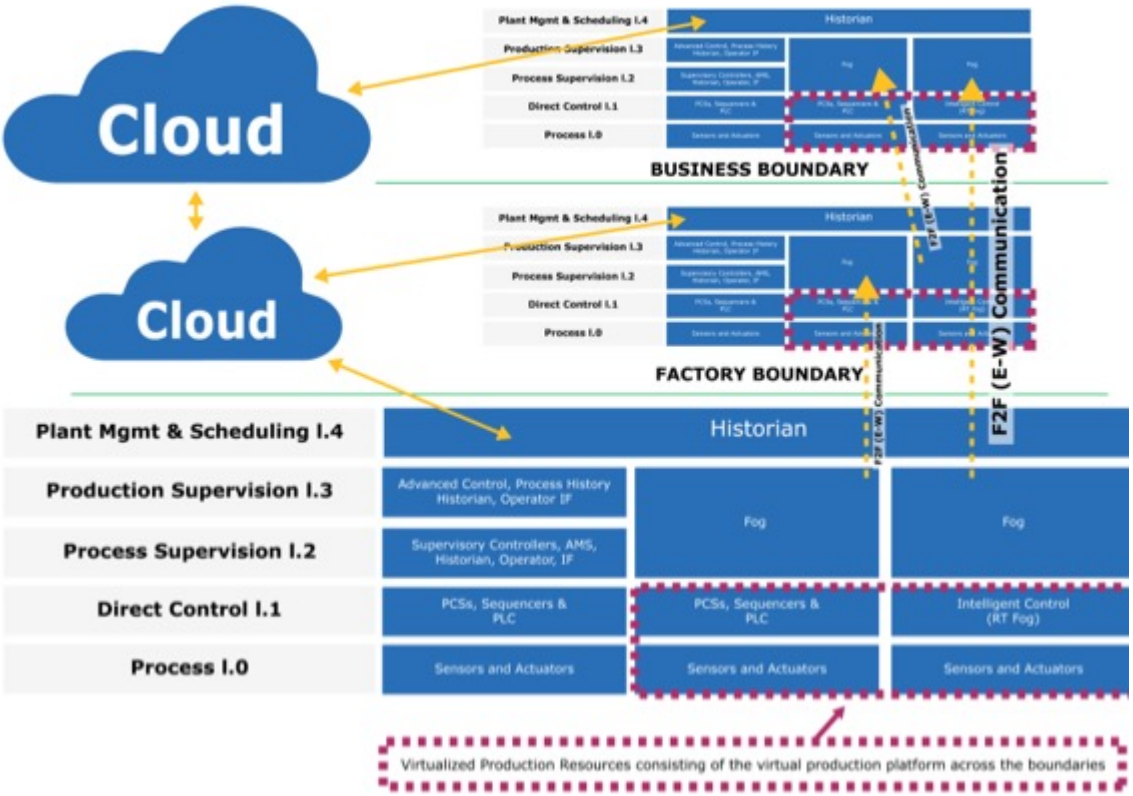

Figure 4: The fog computing architecture enables factory automation systems to be software defined and to communicate with each other in order to share resources across business boundaries.

Key Technical Challenges. Production resource sharing can enhance CAPEX efficiency for resource-limited, growing businesses. It allows them to share (sell/buy or subcontract) production resources in a dynamic and timely manner.

By transforming industrial systems from purpose-built and discrete systems to software defined and modularized systems, fog can enable OPEX improvements. Fog helps manufacturers to deploy high compute nodes for intelligent monitoring and control, as well as factory-wide orchestration:

- The risk of exposing potentially sensitive information for factory-to-factory transactions via east-west communications requires deploying rigorous security mechanisms, and process isolation between multiple tenants on the same fog nodes.

- Data formats and information models of production systems vary from one factory to another. Different business systems need to be able to interoperate and share resources. The OpenFog architecture can unify these.

- Resource sharing must not impact other parts of production operations.

The fog infrastructure has to closely communicate with systems at the plant management and scheduling layers. The functions managed by fog have to be highly modularized and orchestrated across the boundaries. In order to optimize the use of shared resources, sensors, actuators and controls, they need to utilize real-time intelligence for real-time monitoring, control and remote adjustment.

Architectural Considerations. Interoperability is the key to enabling resource sharing in manufacturing plants. At the platform level, given a wide variety of field bus systems, sensor/control data structures, security requirements, safety requirements, operation process controls, etc., it is necessary to abstract legacy systems resources/properties and expose them as interoperable information model (Smart Objects), with a common interaction procedure to access the information and the device models on virtualized environment. This data abstraction is a key function of fog.

This platform-level interoperability scales east-to-west. If more capacity is needed, more fog nodes can be added as peers at one level of the fog hierarchy. For vertical level interoperability, modular design and interfaces between the modules within the fog node need to be defined. In order to secure highly optimized (customized) vertical

information flow and control, the definitions of the modules and the interfaces need to allow flexible deployments for each production line. This interoperability can span from smart sensors and actuators at the lowest levels, up through several layers of fog hierarchy, and into the cloud.

Communications Considerations

In this section, we will drill down into the OT-level fog topology requirements in detail. At the platform level, Purdue levels L1 (or L0)–L3 of ISA-95 can be converged into a software-defined fog system. L2/L3 functionality starts leveraging more advanced analytics (including training/learning), closer to the fog nodes, optimizing factory operations. This enables sharing of the most up-to-date production resources with other factories.

As L0/L1 deployments shift towards fog deployments that are optimized more on inference and control, the needs for real-time analytics/network/IO become critical.

The fog platform may be deployed at L3 and the layers beneath so that complex enterprise business logic on L4 and above doesn't need to be recreated. Fog topology in L1-L3 in a factory will likely be logical topology where modularized functions can be flexibly composed to dynamically change production requests. L0 systems/devices can be augmented by L1 fog systems.

While the topologies are logically and flexibly constructed, physical attributes such as real-time requirements and safety requirements need to be considered. Fog-to-fog orchestration management between factories can be point-to-point, but may require orchestration management of dedicated fog nodes when the scope of resource sharing becomes large scale and complicated.

In order for automation systems to be software defined in a brownfield factory, a foundation of consolidated supervisory systems is required. This can be L2/L3 modular designed software-defined systems. L1 systems can be software-defined control systems deployed on industrial grade PCs and servers. Then, real-time control and analytics are enabled at L0 devices.

These migrations to fog-capable systems don't need to be applied to entire systems. In fact, deployments should expand only to the scope where they make sense. Since fog is scalable, modular, and hierarchical it can expand to the scope and upgrade the capabilities later as needed.

When implementing fog in environments with constrained resources on the systems and devices, security and messaging implementations have to be defined in minimum viable manner (i.e. MVIs).

In manufacturing factory environments, 10 μsec order latency may be required for cases where real-time control of servo motors with analytics is involved. In craft brewing, for example, this is found on a high-speed bottling machine. Otherwise, IP packet latency in general may be managed in μsec order, even for most time-critical systems - especially in discrete manufacturing environments.

In those cases, time to take processing data into the system will be more critical than network latency if the network latency is properly managed by a protocol such as Time-Sensitive Networking (TSN). When production resources are shared or remotely managed from the fog in another factory, these latency controls have to be managed autonomously, as fog-to-fog communication between factories cannot meet μsec packet transfer and data processing requirements.

In fog-based manufacturing environments, data at rest will physically sit on distributed memory/storage systems in a secure manner. However, even data at rest will have to consider how hot the data is.

Hot data will likely be retrieved and stored in a frequent manner, so it may be wise to store the data in a fog node near the system where data is generated and used, and leave it in main memory instead of persistent storage.

For data in motion, as previously discussed, time deterministic data transfer is essential, and it has to be on fully secure network. Given the dynamic nature of fog topology, establishing secure communication on a transaction-by-transaction basis may require a new paradigm of security design and management. For data in use, it's also critical to keep data integrity secure in order to ensure that Smart Objects are managed consistently.

For processing algorithms and analytics, higher-level fog nodes must be relatively large scale to execute complex problem solving. Resources must be shared without introducing downtime or lowering production efficiency of the rest of the systems. This starts with rules-based algorithms on a very small scale, but quickly requires advanced analytics such as artificial intelligence (AI).

Fog node deployments can range from edge computing platforms such as IoT gateways with minimum level of capabilities/performances to VM/containers that make the system software defined. Then, it can grow as performance/capability demands increase with, for example, industrial PCs, industrial servers and on premise data centers.

Mapping the Use Case to the 8 Pillars of OpenFog

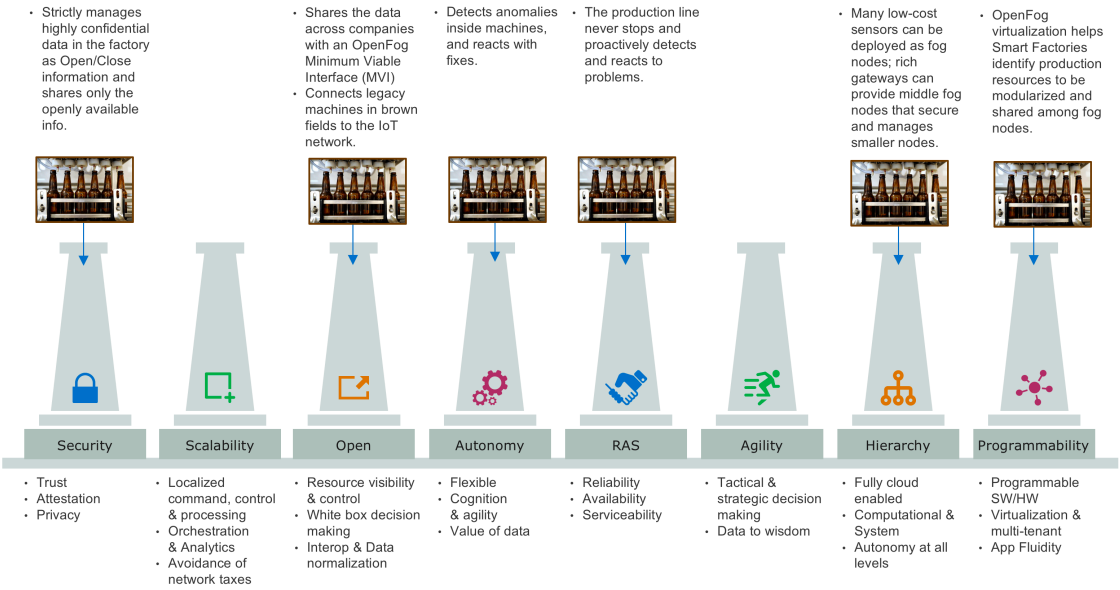

Figure 5. In software-defined production deployments, these OpenFog pillars play a critical role.

- **Security**. The OpenFog architecture enables production systems to be built upon a secure end-to-end computing environment. Things-to-Fog (T2F), Fog-to-Fog (F2F) and Fog-to-Cloud (F2C) connections can be established dynamically even across various applications. Because fog computing enables on-site data processing, sensitive information can stay closed inside the manufacturing plant. The OpenFog architecture is defined to enable secure remote maintenance.

- **Scalability**. By sending only the analysis result, fog computing can reduce the amount of data needed to transmit from a factory to the cloud. This improves the scalability of smart factory resources and third-party providers with regard to communication bandwidth. Computational capacity, network bandwidth and storage size of fog networks can be dynamically scaled to match demand.

- **Open**. The interoperability provided by the OpenFog architecture enables transparent resource discovery and sharing via open APIs. APIs also enable production equipment in the factory to connect to remote maintenance service providers and other partners.

- **Autonomy.** The autonomy provided by fog computing allows the manufacturer to perform designated actions even when there is limited or no communication with the data center or cloud—including resource sharing with other factories. Fog computing enables the brewery to reduce the number of stoppages with early detection of possible failures or predictive maintenance.

- **Reliability/Availability/Serviceability (RAS)**. The OpenFog requirements are defined to ensure high reliability, availability and serviceability in a mission-critical production environment. The architecture also provides remote maintenance and predictive maintenance and contributes to RAS.

- **Agility**. Agility enables localized and intelligent decision making within a fog system. A minor failure in production equipment in the factory can be detected and addressed immediately. Agility can enable manufacturing lines to rapidly adapt to changing customers needs. Low-mid volume / high mix factory production schedules are easily accomidated using fog, and changing between many different processes on the same line is simple. Agility also enables predictive maintenance, which reduces factory downtime.

- **Hierarchy**. The OpenFog architecture allows various T2F, F2F and F2C deployment models off and on the manufacturing premises. It also allows that hybrid and multiple services can be run on fog node and cloud. Monitoring & control, operational support and business support for manufacturing can be

implemented in a dynamic and flexible hierarchy of multiple layers of fog nodes.

- **Programmability.** Resources can be repurposed based on business needs such as resource sharing with other businesses. By optimizing resource usage, the manufacturer can become more efficient. Particularly in manufacturing, OpenFog programmability can enable dynamic change of production lines and factory equipment while maintaining overall production efficiency. It can create dynamic value chains. Additionally, the programmability provided by OpenFog computing also allows data analytics to be conducted in a factory.

Testbed Considerations

There are increasing demands for suppliers, manufacturers, logistics and traders to connect in real time to improve productivity and efficiency with the smart factory capabilities described above. In order to construct such a new production system without vendor lock-in, it is required to be able to use various fog nodes connected vertically and horizontally. The OpenFog Consortium will create testbeds to validate attributes characteristic of smart factories such as real-time nature, information confidentiality, and interoperability between multi-vendor fog nodes.

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 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
Akira Shimizu, ARM	Chuck Byers, Cisco Systems
Tadashi Takahashi, Dell	Evan Birkhead, OpenFog Consortium
Toshimichi Fukuda, Akiko Murakami, Fujitsu Limited	Judith Kelley, OpenFog Consortium
Norikatsu Takaura, Hitachi, Ltd.	
Masahiro Shimohori, Intel Corp.	
Tetsushi Matsuda, Mitsubishi Electric Corp.	
Taiga Yoshida, NTT Communications Corp.	
Masahito Yashiro, Jerome Lefebvre, OSIsoft LLC	
Shunsuke Kikuchi, Sakura Internet.	
Kimihiro Nakamura, Ken Hatano (Lead Author), Toshiba Digital Solutions 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/