

IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing

IEEE Communications Society

Sponsored by the
Edge, Fog, Cloud Communications with IOT and Big Data Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1934™-2018

IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing

Sponsor

Edge, Fog, Cloud Communications with IOT and Big Data Standards Committee
of the
IEEE Communications Society

Approved 14 June 2018

IEEE-SA Standards Board

Abstract: OpenFog Consortium—OpenFog Reference Architecture for Fog Computing is adopted by this standard. OpenFog Reference Architecture [OPFRA001.020817] is a structural and functional prescription of an open, interoperable, horizontal system architecture for distributing computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum of communicating, computing, sensing and actuating entities. It encompasses various approaches to disperse Information Technology (IT), Communication Technology (CT) and Operational Technology (OT) Services through information messaging infrastructure as well as legacy and emerging multi-access networking technologies.

Keywords: adoption, communication technology IEEE 1934™, information technology, OpenFog™, operational technology

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2018 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 2 August 2018. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-5017-1 STD23185
Print: ISBN 978-1-5044-5018-8 STDPD23185

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Xplore at <http://ieeexplore.ieee.org/> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE standard was completed, the Fog Computing and Networking Architecture Framework Working Group had the following membership:

John K. Zao, *Chair*
Tao Zhang and **Jingyi Zhou**, *Vice Chairs*

Wilbert (Chuck) Adams
Donald Banks
San Bhattarai
Charles (Chuck) Byers
Gaurav Chawla
Robinson Do
Jeff Fedders
Maria A. Gorlatova
Ken Hatano

Alfred Ho
Felix Huang
Hank Ching-Yao Huang
Meiqing Huang
Srikanth Kambhatla
Hong Liu
Tetsushi Matsuda
Jing-Ru Su
Norikatsu Takaura
Frank Chee-Da Tsai

Akihisa Ushirokawa
Hung-Yu Wei
James (Jim) White
Makoto Yasuda
Raymond Yeung
Ron Zahavi
Chi Zhang
Jun Zhou
Mingtuo Zhou

The following members of the entity balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

ARM, Ltd.
BII Group Holdings Ltd.
China Electronics
Standardization Institute
Cisco Systems, Inc.
Dell
East China Normal University
(ECNU)

Hitachi, Ltd.
Huawei Technologies Co., Ltd
Institute for Information Industry
Microsoft Corporation
Mitsubishi Electric Corporation
National Taiwan University
NXP Semiconductors

OpenFog Consortium
Princeton University
Shanghai Research Center for
Wireless Communications
The Chinese University of Hong
Kong
TOSHIBA Corporation
ZTE Corporation

When the IEEE-SA Standards Board approved this standard on 14 June 2018, it had the following membership:

Jean-Philippe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Ted Burse
Guido R. Hiertz
Christel Hunter
Joseph L. Koepfinger*
Thomas Koshy
Hung Ling
Dong Liu

Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Paul Nikolich
Ronald C. Petersen
Annette D. Reilly

Robby Robson
Dorothy Stanley
Mehmet Ulema
Phil Wennblom
Philip Winston
Howard Wolfman
Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1934-2018, IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing.

This IEEE standard adopts OpenFog Consortium™—OpenFog Reference Architecture for Fog Computing [ID: OPFRA001.020817].

That document offers a structural and functional prescription of an open, interoperable, horizontal system architecture for distributing computing, storage, control, and networking functions closer to the users along a cloud-to-thing continuum of communicating, computing, sensing, and actuating entities. It encompasses various approaches to disperse Information Technology (IT), Communication Technology (CT), and Operational Technology (OT) Services through a unifying information messaging infrastructure as well as legacy and emerging multi-access networking technologies.

The adoption of this OpenFog™ document will help to lay down a foundation for the Fog Computing and Networking Architecture Framework, which will be developed in this working group.

IEEE is adopting this document “as is” and in its entirety with the acknowledgement of an error in Line 18 on Page 106 in Section 7.1.3.1. Users should update the sentence as follows: “we push both the ~~both the~~ topology and the corresponding weights to target ~~or~~-classification system.” Additional changes to this standard have been included in Annex A.

Contents

OpenFog Reference Architecture for Fog Computing	1
Annex A (normative) Accepted changes	163



OpenFog Reference Architecture for Fog Computing

Produced by the OpenFog Consortium Architecture Working Group
www.OpenFogConsortium.org

February 2017

Use of this Document

Copyright © 2017 OpenFog Consortium. All rights reserved. Published in the USA.

Published February 2017.

This is an OpenFog Consortium document and is to be used in accordance with the terms and conditions set forth below. The information contained in this document is subject to change without notice.

The information in this publication was developed under the OpenFog Consortium Intellectual Property Rights policy and is provided as is. OpenFog Consortium makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of fitness for a particular purpose. This document contains content that is protected by copyright. Copying or distributing the content from this document without permission is prohibited.

OpenFog Consortium and the OpenFog Consortium logo are registered trademarks of OpenFog Consortium in the United States and other countries. All other trademarks used herein are the property of their respective owners.

Acknowledgements

The OpenFog Reference Architecture is the product of the OpenFog Architecture Workgroup, co-chaired by Charles Byers (Cisco) and Robert Swanson (Intel). It represents the collaborative work of the global membership of the OpenFog Consortium. We wish to thank these organizations for contributing to this work and to the advancement of fog computing technology, research and innovation: Aalto University; ABBALab; Arizona State University; ARM; AT&T; Caltech; Cisco; Dell; FogHorn Systems; Fujitsu; GE Digital; Hitachi; Foxconn; Indian Institute of Technology; Industrial Technology Research Institute; Institute for Information Industry; Institute of Network Coding; The Chinese University of Hong Kong; Intel; Internet Initiative Japan Inc.; ITOCHU techno-Solutions Corporation; Kii; LGS Innovations; MARSEC; Microsoft; Mitsubishi Electric Corporation; National Chiao Tung University; National Taiwan University; Nebbiolo Technologies; NEC Corporation; NGD Systems; NTT Communications; OSIsoft; Princeton University; PrismTech; Real-Time Innovations; relayr; SAKURA Internet; Schneider Electric; Shanghai Institute of Microsystem and Information Technology; ShanghaiTech University; Singapore University of Technology and Design; SRC Inc.; Stichting imec Nederland; The Chinese University of Hong Kong; Toshiba; Technische Universität Dresden; TTTech; University of Colorado Boulder; University of Georgia; University of Pisa; University of Southern California; Vanderbilt University; Wayne State University.

OpenFog Overview

Digital innovation from the Internet of Things (IoT), Artificial Intelligence, Virtual Reality, Tactile Internet and 5G applications is transforming the way we work, commute, shop and play. Data from newly-connected factories, homes, communities, cars, hospitals and more is expected to grow from 1.1 zettabytes (or 89 exabytes) per year in 2016 to 2.3 zettabytes (or 194 exabytes) per year by 2020.¹ Current “cloud-only” architectures cannot keep up with the volume and velocity of this data across the network, thereby reducing the value that can be created and captured from these investments.

Fog computing provides the missing link in the cloud-to-thing continuum. Fog architectures selectively move compute, storage, communication, control, and decision making closer to the network edge where data is being generated in order solve the limitations in current infrastructure to enable mission-critical, data-dense use cases.

Fog computing is a:

A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum.

Fog computing is an extension of the traditional cloud-based computing model where implementations of the architecture can reside in multiple layers of a network’s topology. However, all the benefits of cloud should be preserved with these extensions to fog, including containerization, virtualization, orchestration, manageability, and efficiency. In many cases, fog computing works with cloud. Pillars, which are common themes of the OpenFog reference architecture include security, scalability, openness, autonomy, RAS (reliability, availability and serviceability), agility, hierarchy, and programmability. In addition to the pillars, we describe the roles of each stakeholder in the fog value chain from silicon creator to the Operating System and application developer through a composite architectural description

Fog computing also is often erroneously called edge computing, but there are key differences. Fog works with the cloud, whereas edge is defined by the exclusion of cloud. Fog is hierarchical, where edge tends to be limited to a small number of layers. In addition to computation, fog also addresses networking, storage, control and acceleration.

The OpenFog Consortium was formed on the principle that an open 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 solutions can limit supplier diversity and ecosystems, resulting in a detrimental impact on market adoption, system cost, quality and innovation. It is our intent to ensure the OpenFog reference architecture results in fully interoperable and secure systems, supported by a vibrant supplier ecosystem.

Contents

Use of this Document	2
Acknowledgements	2
OpenFog Overview	3
1 About Fog Computing and the Consortium	9
1.1 OpenFog Reference Architecture Overview	9
2 Areas of Opportunity	10
2.1 OpenFog Reference Architecture Content	10
2.2 The Internet of Things, Cloud and the OpenFog RA	10
2.3 OpenFog and Other Consortia	12
3 Use Cases for Fog	13
3.1 Transportation Scenario: Smart Cars and Traffic Control	14
3.2 Visual Security and Surveillance Scenario	17
3.3 Smart Cities Scenario	19
3.3.1 Smart Buildings	20
3.4 Additional Use Cases	21
4 Pillars of OpenFog RA	22
4.1 Security Pillar	23
4.2 Scalability Pillar	24
4.3 Openness Pillar	26
4.4 Autonomy Pillar	27
4.5 Programmability Pillar	29
4.6 Reliability, Availability, and Serviceability (RAS) Pillar	29
4.7 Agility Pillar	31
4.8 Hierarchy Pillar	31
4.8.1 Hierarchical Fog Deployment Models	34
5 Reference Architecture Overview	38
5.1 Functional Viewpoint	38
5.2 Deployment Viewpoint	39
5.2.1 OpenFog Deployment Types	39
5.2.2 N-Tier Fog Deployment	39

5.3	OpenFog Architecture Description.....	43
5.4	Perspectives (Cross Cutting Concerns).....	46
5.4.1	Performance and Scale Perspective.....	46
5.4.2	Security Perspective.....	47
5.4.3	Manageability Perspective.....	51
5.4.4	Data, Analytics, and Control.....	54
5.4.5	IT Business and Cross-fog Applications.....	56
5.5	Node View	56
5.5.1	Network	58
5.5.2	Accelerators	65
5.5.3	Compute.....	66
5.5.4	Storage	67
5.5.5	OpenFog Node Management.....	68
5.5.6	OpenFog Node Security	69
5.6	System Architecture View	73
5.6.1	Hardware Platform Infrastructure	74
5.6.2	Hardware Virtualization and Containers.....	77
5.7	Software Architecture View	77
5.7.1	Software View Layers.....	77
6	Adherence to OpenFog Reference Architecture.....	89
7	An End-to-End Deployment Use Case	91
7.1	Airport Visual Security	91
7.1.1	Cloud and Edge Approaches	92
7.1.2	Fog Computing Approach	93
7.1.3	Application to Airport Visual Security.....	103
8	Additional Opportunities.....	120
9	Summary and Next Steps	121
10	Appendix – Deeper Security Analysis	122
10.1	Security Aspects	122
10.1.1	Cryptographic Functions	122
10.1.2	Node Security Aspect	127

10.1.3 Network Security Aspect.....	130
10.1.4 Data Security Aspect	137
11 Glossary	143
References	159

Figures

Figure 1 Unfettered Cloud Computing.....	11
Figure 2 OpenFog Consortium and Other Consortia	12
Figure 3 OpenFog Transportation: Smart Car and Traffic Control System ..	15
Figure 4 Opportunities for Smart Cities.....	19
Figure 5 Pillars of OpenFog	22
Figure 6 Layered Architecture View of an IoT System	32
Figure 7 IoT System Deployment Models	34
Figure 8 Fog Hierarchy Example	36
Figure 9 Fog Hierarchical Deployment Model	36
Figure 10 Multi-Tier Deployment	39
Figure 11 Intelligence from data.....	41
Figure 12 Fog Node East/West Communication	43
Figure 13 Architecture Description with Perspectives	44
Figure 14 OpenFog Architecture Perspectives	46
Figure 15 OpenFog Security Layers.....	48
Figure 16 Example Threats and Attacks	49
Figure 17 Management Lifecycle.....	52
Figure 18 Management Layer.....	54
Figure 19 Business Intelligence	55
Figure 20 Cross Fog Application.....	56
Figure 21 Node View.....	57
Figure 22 System Architecture View.....	74
Figure 23 Software Architecture View.....	78
Figure 24 Application Support	82
Figure 25 Containerization for Application Support	82
Figure 26 Application Services Layer	85
Figure 27 Containerization for Application Support	86
Figure 44 OpenFog Technology Ready	89
Figure 43 OpenFog Ready	89
Figure 28 Airport Scenario	91
Figure 29 Key pillars of the OpenFog Architecture	93
Figure 30 OpenFog Architecture Description	94

Figure 31 Node view for Visual Security	95
Figure 32 OpenFog Approach to Visual Security Scenario	104
Figure 33 OpenFog realized for Visual Security	105
Figure 34 Training and Classification system for Machine Vision	106
Figure 35 Airport License Plate Capture	107
Figure 36 Integrated Air Travel Infrastructure - Assumptions.....	108
Figure 37 Garage Gate Entrance.....	109
Figure 38 Central System Analytics.....	110
Figure 39 Terminal Entrance	111
Figure 40 Baggage flow.....	113
Figure 41 Departure Gate.....	116
Figure 42 Arrival Gate.....	117
Figure 50 OpenFog Node Security Architecture	127
Figure 45 OpenFog Security Functional Layers and Operational Planes	131
Figure 46 OpenFog Secure Communication Pathways	132
Figure 47 Protocol Suites for Secure Node-to-Node Communications	133
Figure 48 Protocol Suites for Secure Node-to-Node Communications	133
Figure 49 Protocol Suites for Secure Node-to-Device Communications	135

1 *About Fog Computing and the Consortium*

1.1 OpenFog Reference Architecture Overview

The OpenFog Consortium was founded by ARM, Cisco, Dell, Intel, Microsoft and Princeton University in November 2015. Through its global membership of leading technology & networking players, fog computing entrepreneurs and university researchers, the Consortium is helping to enable game-changing innovation enabled by fog computing through an open architectural framework.

We are guided by the OpenFog Board of Directors and the OpenFog Technical Committee. The technical committee is the technical governing body of all of the working groups of the Consortium. The chair of this group is elected by a vote of the OpenFog Board of Directors and reports directly to the board.

The OpenFog Reference Architecture (OpenFog RA) is intended to help business leaders, software developers, silicon architects, and system designers create and maintain the hardware, software and system elements necessary for fog computing.

Many different technical workgroups in the Consortium are responsible for the different aspects of this reference architecture, including: Communications, Software-Infrastructure and Security. The Architecture Framework workgroup is tasked with the creation and maintenance of this document and other technical publications of the Consortium. All new technical topics requiring investigation are assigned to this group. The charter of each group is managed and approved by the technical committee and Board of Directors.

For further information on these groups or how to participate, please reference www.openfogconsortium.org.

2 Areas of Opportunity

2.1 OpenFog Reference Architecture Content

The OpenFog RA is part of a suite of documents under construction by the OpenFog Consortium and our technical liaisons partners. It is a medium- to high-level view of system architectures for fog nodes and networks. Future documents will provide lower-level details, including formal, enumerated requirements that will form the basis of quantitative testbeds and the specified interoperability of fog elements. Future documents will also refine the use cases described in Chapter 3.

The OpenFog RA is divided into the following chapters:

- **Chapter 2** describes the OpenFog Consortium's mission and plans to accelerate fog computing. It also provides an overview of the OpenFog RA.
- **Chapter 3** presents some use cases where we see fog computing emerging. This list will grow and evolve as the OpenFog RA is refined.
- **Chapter 4** describes the pillars of the OpenFog architecture. These are the guiding principles for the OpenFog RA.
- **Chapter 5** provides an in-depth look at the full OpenFog RA. This section shows the abstract Architectural Description (AD) for the OpenFog RA.
- **Chapter 6** starts the conversation on adherence to the OpenFog architecture. It is our intention to drive standardization across the various interfaces.
- **Chapter 7** shows the abstract OpenFog RA applied to various use cases. This will further clarify each aspect of the OpenFog RA and what needs to be done for a successful implementation.
- **Chapter 8** contains some of the open areas of fog computing and new opportunities for research. OpenFog member companies and academic organizations are continually enhancing and refining the OpenFog RA. This section will help advance the overall architecture over time.

2.2 The Internet of Things, Cloud and the OpenFog RA

The Internet of Things (IoT) is driving business transformation by connecting everyday objects and devices to one another and to cloud-hosted services.

Current deployment models emphasize mandatory cloud connectivity; however, this is not feasible in many real-world situations. These are two of the primary issues with connecting edge devices to the cloud for all services:

- Connected devices are creating data at an exponentially growing rate, which will drive performance and network congestion challenges at the edge of infrastructure.
- There are performance, security, bandwidth, reliability, and many other concerns that make cloud-only solutions impractical for many use cases.

Unfettered cloud-only architectural approaches cannot sustain the projected data velocity and volume requirements of the IoT. To sustain IoT momentum, the OpenFog Consortium is defining an architecture to address infrastructure and connectivity challenges by emphasizing information processing and intelligence at the logical edge. This approach is called fog computing.



Figure 1 Unfettered Cloud Computing

While the cloud itself may play a vital role in many deployments, fog computing represents a shift from traditional closed systems and a reliance on cloud-only focused models. Fog computing is complementary to, and an extension of, traditional cloud-based models.

The fog computing model moves computation from the cloud closer the edge, and potentially right up to the IoT sensors and actuators. The computational, networking, storage and acceleration elements of this new model are known as fog nodes. These are not completely fixed to the physical edge, but should be seen as fluid system of connectivity.

2.3 OpenFog and Other Consortia

The OpenFog Consortium invites open participation from across industry, academia and non-profit organizations that have an interest in the emerging IoT landscape. The mission of the OpenFog Consortium is complementary to other IoT and technology industry alliance groups including the Industrial Internet Consortium (IIC), ETSI-MEC (Mobile Edge Computing), OPC-UA, Open Connectivity Foundation (OCF), OpenNFV, and many others. To avoid duplication of effort and market confusion, the OpenFog Consortium will de-emphasize efforts to optimize for some of these application spaces, and instead focus on optimally serving vertical markets not addressed by other initiatives. In the longer term, through liaisons with these and other bodies, we will drive more convergence across the IoT industry under a common view of edge and fog architectures.

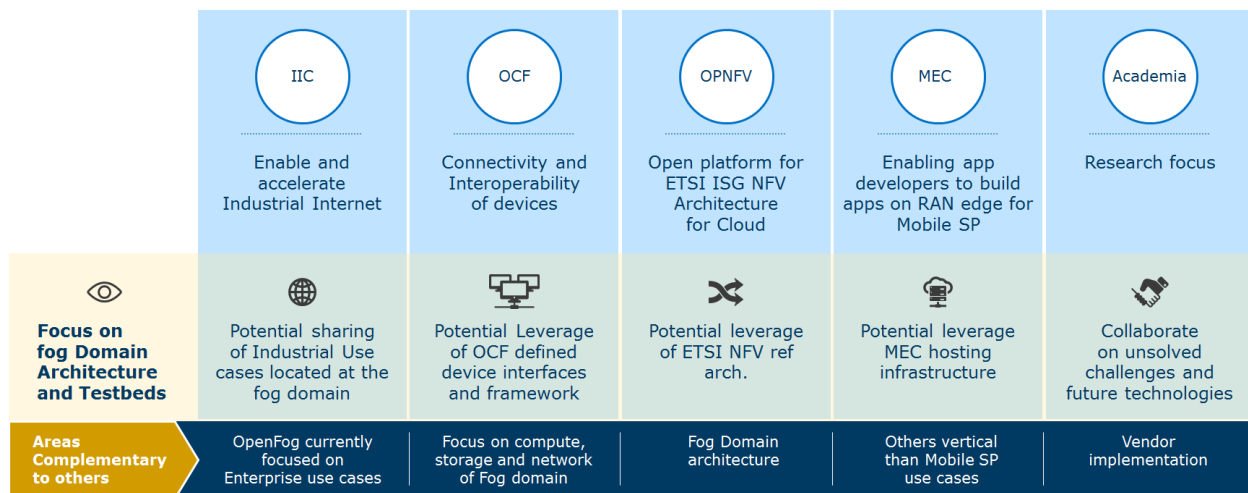


Figure 2 OpenFog Consortium and Other Consortia

3 *Use Cases for Fog*

Fog computing targets cross-cutting concerns like the control of performance, latency and efficiency are also key to the success of fog networks. Cloud and fog computing are on 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 go to fog and what goes to the backend cloud are application specific. This segmentation could be planned, but also change dynamically if the network state changes in areas like processor loads, link bandwidths, storage capacities, fault events, security threats, cost targets, etc.

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

- **Security:** Additional security to ensure safe, trusted transactions
- **Cognition:** awareness of client-centric objectives to enable autonomy
- **Agility:** rapid innovation and affordable scaling under a common infrastructure
- **Latency:** real-time processing and cyber-physical system control
- **Efficiency:** dynamic pooling of local unused resources from participating end-user devices

A quick use case example to illustrate the value of fog: 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) analyze the readings in cloud servers to detect abnormal conditions, and send commands back to adjust the position 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 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. Fog nodes can be highly secure, lessening the hacker threat. Fog nodes can 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.

This document describes a set of high-level attributes of fog computing that we call the pillars (including some of the fog advantages described in the pipeline control scenario). These include security, scalability, openness, autonomy, reliability, agility, hierarchical organization and programmability. We will discuss all of these pillars in detail later in this document.

Platform as a service (PaaS) is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage web applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an application. 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 show how FaaS may be enabled and will be expanded upon in the reference architecture.

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 applications that require real-time decision making, low latency, improved security, and are network-constrained. In this section, we look at a few specific use cases.

3.1 Transportation Scenario: Smart Cars and Traffic Control

In 2016, the average person creates around 650MB of data every day and by 2020, some project that to more than double. However smart autonomous cars will generate multiple terabytes of data every day from the combinations of light detection and ranging (LIDAR), global positioning

systems (GPS), cameras, etc. When the smart car is coupled with intelligent infrastructure, it is clear that a cloud-only model will not work for autonomous transportation, and that a fog computing approach is required. Many of the architecture requirements we describe in smart cars and traffic control also apply to other transportation areas, such as ships/boats, trains, trucks, busses and drones. In this section we will highlight the opportunity for fog computing for smart cars and traffic control, and explain how the requirements are addressed by the OpenFog RA. The figure below is an overview of an intelligent highway application of the OpenFog RA.

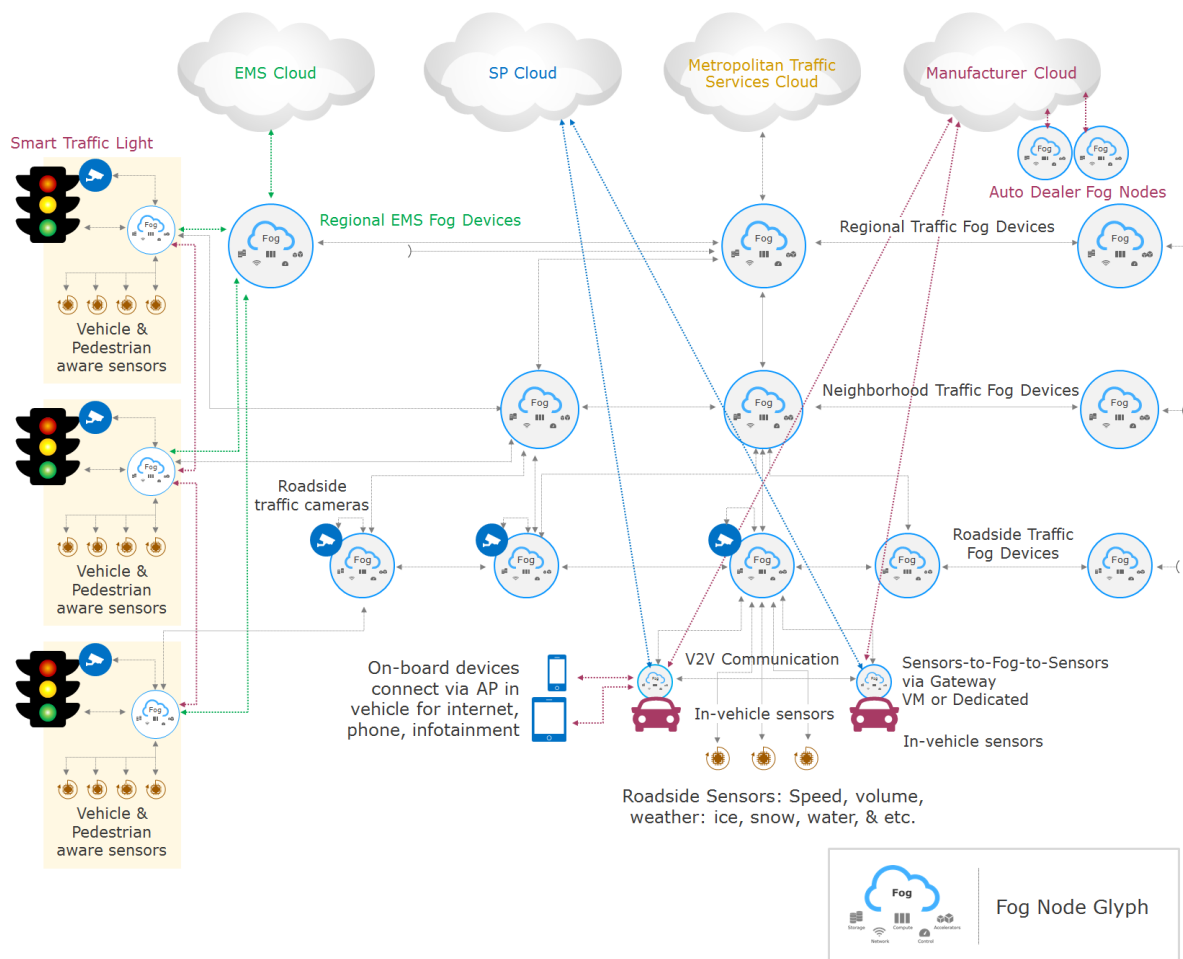


Figure 3 OpenFog Transportation: Smart Car and Traffic Control System

The smart car and traffic control use case presents an opportunity to examine a fog environment containing a rich set of interactions among multiple fog domains as well as multiple cloud domains. Among other things, this use case demonstrates:

- A rich set of interactions among multiple fog domains, as well as multiple cloud domains, including Element Management Systems (EMS), service provider (SP), metro traffic services, and system manufacturer clouds.
- Mobile fog nodes supporting vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-x (V2X) interactions.
- Multiple fog networks owned and operated by different authorities providing similar (and different) functionality
- Multi-tenancy across fog nodes will also be a burgeoning opportunity to consolidate multiple fog networks, improving efficiency.
- Both private and public fog and cloud networks used by a single end point device.

This use case shows also shows the hierarchical and distributed advantages of a fog architecture. As shown in the figure above, the system includes several types of sensors (and actuators) that we refer to as “things.”

Things include roadside sensors (infrastructure) and on-vehicle sensors. These sensors provide data so that the various systems (lights, cars, etc.) can carry out their given functions (e.g. vehicle driving autonomously). Smart transportation systems also manage the actuators that control parts of the infrastructure, such as traffic signals, gates, and digital signs.

The vehicles connect to the cloud and a hierarchy of fog nodes that service the autonomous vehicle or traffic control systems.

Fog computing nodes in the vehicle

In this use case example, the vehicle is a mobile fog node that communicates with other fog nodes as they become available, an example of V2I interactions. The mobile fog node must also be capable of performing all required in-vehicle operations autonomously in circumstances where it cannot connect to other fog nodes or the cloud.

In-vehicle fog nodes provide services including infotainment, advanced driver assistance systems (ADAS), autonomous driving, collision avoidance, navigation, etc. Several different networking technologies, including Dedicated Short Range Communications (DSRC), cellular (e.g. 3G, LTE, 5G, etc.) and other networking technologies securely connect the vehicles to each other and the infrastructure.

The Transportation Fog Network

The Transportation Fog Network is comprised of a three-level hierarchy of fog nodes.

The first level of the hierarchy is the infrastructure fog nodes, or roadside fog nodes. At this level, the roadside fog sensors collect data from other devices such as roadside cameras. The fog nodes perform some local analysis for local action, such as alerting the vehicle about poor road conditions, triggering autonomous response to slow down, and perform some autonomous functions, even if connections to higher layers are unavailable. Data from the first level of interactions is aggregated and sent up the fog hierarchy to the second and third levels of the hierarchy—neighborhood and regional fog nodes—for further analysis and distribution. Some of the data may also be distributed east-west to other infrastructure nodes for their use. Typically, each fog layer in the hierarchy will provide additional processing, storage, and network capabilities in service of the vertical application at their level of the hierarchy. For example, higher level layers provide additional processing to provide data analytics or large storage capacities.

Traffic control systems

Traffic control fog nodes may receive input from other sources, such as smart traffic light systems, municipal managers, and cloud-based systems. Data flows between the traffic control system, infrastructure fog nodes and vehicles in all directions, insuring all levels of the hierarchy have the data and control capabilities they need.

The goal of the OpenFog RA for smart cars and traffic control is to ensure an open, secure, distributed, and scalable architecture that optimizes real time capabilities within a multi-supplier ecosystem. The transportation example shows a complex system of autonomous things and infrastructure generating massive amounts of data. We believe that this use case highlights the need for fog computing to enable safe and effective operations in IoT, 5G, AI and other advanced scenarios.

3.2 Visual Security and Surveillance Scenario

Surveillance and security cameras are being deployed worldwide. These cameras are used to ensure security of materials, people, and places. In addition, these cameras have the ability to generate a massive amount of data, which can exceed terabytes per day for a single camera. Traditional cloud models that were deployed for low-resolution cameras aren't scalable with the 1080p and 4K cameras because of the sheer availability and/or cost of network transport. Additionally, decisions on

security need to be made at the camera or installation location and cannot be made solely in the cloud. Machine vision is also a prime candidate for accelerators and dynamic updating of various algorithms in both hardware and software. These cameras are capturing images of people, places, or things and are tightly coupled to decision-making, which requires a heightened level of security of the camera's software and hardware assets.

Smart cities, smart homes, retail stores, public transportation, manufacturing and enterprises increasingly rely on camera sensors to secure people, identify unauthorized access, and increase safety, reliability and efficiency. The sheer bandwidth of visual (and other sensor) data being collected over a large-scale network makes it impractical to transport all the data to the cloud to obtain real-time insights. A particularly demanding application space is surveillance of high value installations with many people and objects moving through them. Controlling a large network of cameras in an airport is a good example of such an application. This use case will be studied in greater detail later in this document to illustrate applying the architecture to a concrete application.

City-scale deployments that include placing cameras on traffic lights and other camera deployments in remote areas don't have high-bandwidth connectivity to the cloud to upload the collected video, even if the video could fit over the network infrastructure. Real-time monitoring and detection of anomalies (intruders into a building, the fall of an elderly citizen, the misfiring of a piece of manufacturing equipment) pose strict low latency requirements on surveillance systems; timeliness is important from the standpoint of both detection and response.

Additionally, privacy concerns must be addressed when using a camera as a sensor that collects image data so that the images do not reveal a person's identity or reveal confidential contextual information (e.g. intellectual property in a manufacturing plant) to any unauthorized parties. OpenFog RA deployments provide the opportunity to build real-time, latency-sensitive distributed surveillance systems that maintain privacy. It leverages fog nodes to intelligently partition video processing between fog nodes co-located with cameras and the cloud so as to enable real-time tracking, anomaly detection, and insights from data collected over long time intervals. Video analytics algorithms can be located on fog nodes close to the cameras, and take advantage of the heterogeneous processor capability of fog, running parts of the video analytics algorithm on conventional processors or accelerators.

The visual security for airports use case requires all of these fog capabilities to meet its performance, reliability, security and efficiency goals. This includes vehicle detection, people detection, smart retail, and other areas where machine vision via video analytics is important for fog computing. Please see the detailed analysis in chapter 7 for many more details including an application of the OpenFog RA.

3.3 Smart Cities Scenario

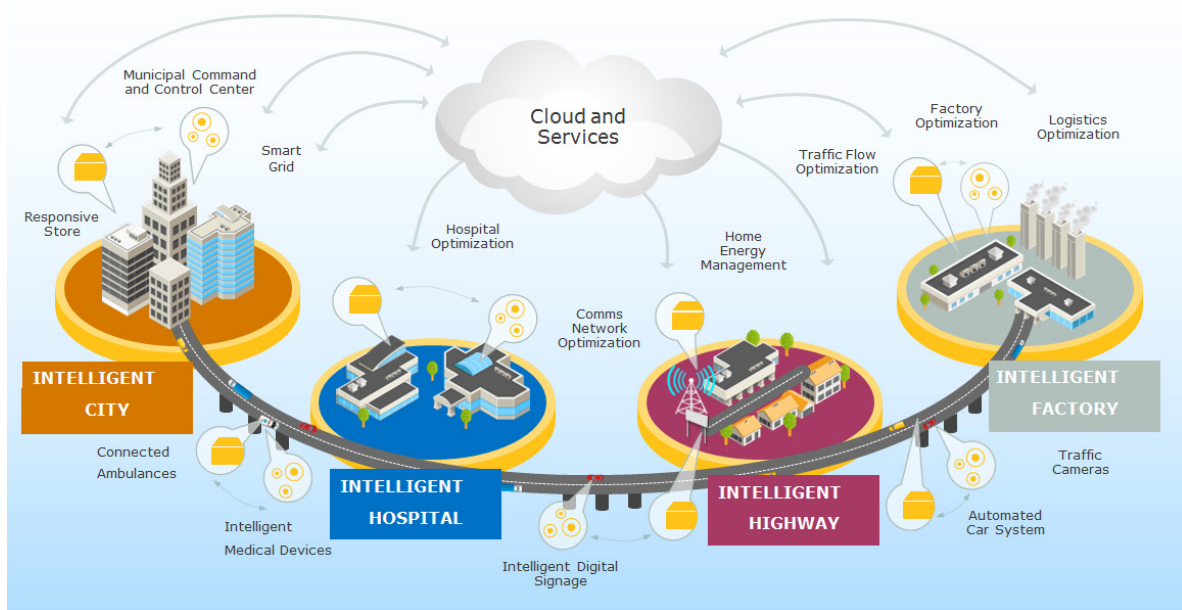


Figure 4 Opportunities for Smart Cities

Smart cities are using technology to deal with many challenges, including traffic congestion, public safety, energy consumption, sanitation, and public internet connectivity.

The OpenFog RA enables greater efficiency and economic realities of smart city operations. The figure above illustrates the various aspects where fog computing can impact smart cities including but not limited to:

- Intelligent city with smart parking, shopping, and infrastructure.
- Intelligent hospitals linking all aspects for greater patient care and healthcare delivery.
- Intelligent highway systems to optimize utilization of infrastructure.
- Intelligent factories and software defined industrial systems.

Connectivity

While most modern cities have one or more cellular networks providing city-wide coverage, these networks often have capacity and peak bandwidth limits that just barely meet the needs of their existing subscribers. This leaves little bandwidth for the advanced municipal services envisioned in a smart city. OpenFog RA deployments coupled with 5G technologies provide an opportunity to address this concern. Fog nodes can provide local processing and storage, and optimize network usage.

Safety and Security

Smart city planning also includes critical public safety and security requirements. For example:

- Municipal networks carry sensitive traffic and citizen data (e.g., police dispatches), and operate life-critical systems (e.g., first responder communications)
- Video security and surveillance systems capture suspicious or unsafe conditions (e.g., utility network problems, unauthorized use of public spaces, etc.)

Note that smart cars and traffic congestion, which is covered as a separate use case, are also top priorities for smart cities.

By providing secure data and distributed analytics, fog computing will play a key role in addressing public safety and security issues for smart cities.

3.3.1 Smart Buildings

Smart buildings may contain thousands of sensors to measure various building operating parameters, including temperature, humidity, occupancy, door open/close, keycard readers, parking space occupancy, security, elevators, and air quality. These sensors capture telemetry data at various intervals and transmit this information to a local storage server. Once this information is processed (analyzed), controller-driven actuators will adjust building conditions as necessary.

Some of this processing and response is extremely time-sensitive. For example, turning on fire suppression systems in response to a fire event or locking down an area if an unauthorized person tries to gain entry. Time-sensitive means real-time response, which requires processing in close proximity to the infrastructure devices.

The OpenFog RA model can be extended into the building's control hierarchy to create a number of smart, connected spaces within each building. Using the hierarchical design of the OpenFog RA, each floor, wing, or even individual room could contain its own fog node.

A fog node could be responsible for:

- Performing emergency monitoring and response functions.
- Performing building security functions.
- Controlling climate and lighting.
- Providing a more robust compute and storage infrastructure for building residents to support smartphones, tablets and desktop computers.

Locally stored operational history can be aggregated and sent to the cloud for large-scale analytics. These analytics can be applied to machine learning to create optimized models, which are then downloaded to the local fog infrastructure for execution.

3.4 Additional Use Cases

Fog computing is relevant to many more use cases in addition to those highlighted here. The OpenFog Board of Directors and Technical Committee will set the most important areas for the Consortium and its partners to focus on in the near and long term. We will continue to expand upon all use cases in further discussions, publications, and testbeds.

4 Pillars of OpenFog RA

The OpenFog RA is driven by a set of core principles called pillars. These pillars form the belief, approach and intent that guided the definition of the reference architecture. They represent the key attributes that a system needs to embody the OpenFog definition of a horizontal, system-level architecture that provides the distribution of computing, storage, control, and networking functions closer to the data source (users, things, et al) along the cloud-to-thing continuum.

The following sections describe each pillar of the architecture.

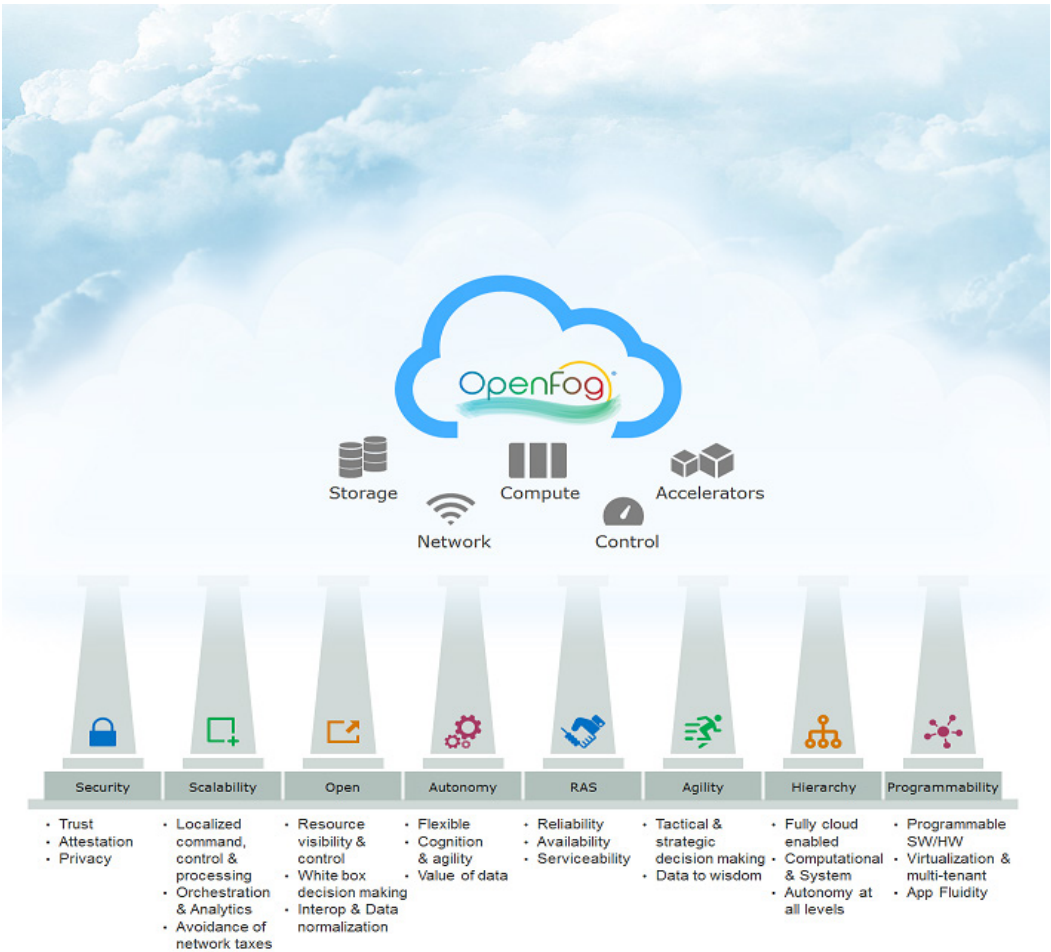


Figure 5 Pillars of OpenFog

4.1 Security Pillar

Many IoT applications supported by the OpenFog RA have privacy critical, mission critical, and even life critical aspects. As such, any security compromise in the fog network can have severe consequences. The OpenFog RA, as it abstracts these technologies, will enable the flexible creation of computing environments that address a broad spectrum of security concerns spanning from IoT devices to cloud and the fog networks in between.

Security in the OpenFog RA is not a one-size-fits-all architecture. Rather, it describes all of the mechanisms that can be applied to make a fog node secure from silicon to software application. Business case, target market, and vertical use case, as well as the location of the node itself, will all create a set of requirements for that node. However, there are certain foundational parts of the architecture, which must be in place in order to build a secure execution environment.

Security implementations have many different descriptions and attributes such as privacy, anonymity, integrity, trust, attestation, verification, and measurement. These are key attributes for the OpenFog RA. Achieving the foundational elements for security requires an approach to discover, attest, and verify all smart and connected “things” before trust can be established.

Conformance to the OpenFog RA requirements will ensure that an OpenFog deployment will be built on a secure end-to-end compute environment. This includes the OpenFog node security, OpenFog network security, and OpenFog management and orchestration security. This will allow architects and designers to focus on the high-value security and privacy problems specific to the types of devices used in their application.

In many applications, particularly for brownfield deployments, or for tiny devices and sensors with little to no security capability, an OpenFog node may act as a device’s first point of secure entry into an OpenFog compute hierarchy and the cloud.

The security pillar of the OpenFog RA starts with a clear definition of base building blocks. All fog nodes must employ a hardware-based immutable root of trust. The Hardware Root of Trust is a trusted hardware component which receives control at power-on. It then extends the chain of trust to other hardware, firmware, and software components. The root of trust should then be attestable by software agents running within and throughout the infrastructure. Because of the proximity to the edge, nodes in fog networks often act as the first node of access control and encryption. This

means they must provide contextual integrity and isolation, and control aggregation of privacy-sensitive data before it leaves the edge.

As more complex topologies are created in FaaS implementations, the attestation continues as a chain of trust from the fog node, to other fog nodes, and to the cloud. Since fog nodes may also be dynamically instantiated or torn down, hardware and software resources should be attestable. Components that are not attestable should not be fully allowed to participate in the fog node or may not be deemed to have fully trustworthy data.

4.2 Scalability Pillar

The scalability pillar addresses the dynamic technical and business needs behind fog deployments. Elastic scalability cuts across all fog computing applications and verticals. The hierarchical properties of fog and its location at logical edges of networks add additional scaling opportunities.

- Individual fog nodes can scale internally, through the addition of hardware or software.
- Fog networks can scale up and out through the addition of fog nodes to assist heavily loaded nodes, either on the same level of the fog hierarchy or in adjacent levels.
- A network of fog nodes can be scaled up or down in a demand-driven elastic environment.
- Storage, network connectivity, and analytics services can scale with the fog infrastructure.

Because of the variability in the use cases for fog computing, the OpenFog RA enables elastic scaling of modest deployments through large mission critical deployments based on demand. This scalability is essential for fog computing implementations to adapt to workload, system cost, performance, and other changing business needs.

Scalability may involve many dimensions in fog networks:

- **Scalable performance** enables growth of fog capabilities in response to application performance demands (e.g., reducing latency between sensor reading and resulting actuator responses).
- **Scalable capacity** allows fog networks to change in size as more applications, endpoints, “things,” users, or objects are added or removed from the network. You can add capacity to individual fog nodes by adding hardware like processors, storage devices, or network

interfaces. You can also add capacity through software and various pay-as-you-grow licensing. The converse is also true.

- **Scalable reliability** permits the inclusion of optional redundant fog capabilities to manage faults or overloads. Redundant fog nodes also ensure a large deployment's integrity and reliability at scale, which is part of the reliability, availability, and serviceability (RAS) pillar. There are hardware and software aspects to scalable reliability. The scalability mechanisms supporting the reliability of fog networks must themselves be highly reliable. Availability (which is a scaling measure closely related to reliability) scales through similar methods.
- **Scalable security** is often achieved through the addition of modules (hardware and software) to a basic fog node as its security needs become more stringent. Capabilities like scalable distribution, rights access, crypto processing capacity, and autonomous security features contribute to scalable security.
- **Scalable hardware** involves the ability to modify the configuration of the internal elements of fog nodes, as well as the numbers of and relationships between fog nodes in networks.
 - Processors scale from modest single core CPUs to specialized accelerator chips with thousands of cores or millions of gates.
 - Networking scales from a single wireless or wire line interface to large arrays of wireless, wire line, and fiber interfaces with aggregate capacities of many Gb/s.
 - Storage can scale from a simple flash chip to large arrays of flash / rotating disks and network attached file systems.

These resources can be configured in initial deployments and retrofit into existing modular fog nodes on an as-needed basis. It is also possible to scale at the network level, by adding arrays of fog nodes at locations in the network where single nodes formally managed the entire load. Hardware scaling can also be in the downward direction, where modules or entire fog nodes that are no longer are needed at a specific location are powered down and/or removed (and perhaps reused elsewhere in a fog network where there is a higher need).

- **Scalable software** is also important and includes applications, infrastructure, and management.
 - The management infrastructure of fog must scale to enable the efficient deployment and ongoing operation of tens of millions of fog nodes in support of billions of smart and connected things.
 - Orchestration must be scalable to manage the partitioning, balance, and allocation of resources across the fog network.
 - Analytics as a capability of fog networks has particularly aggressive scalability targets. This is because analytics

algorithms undergo several orders of magnitude scaling due to increased capacity demands and several additional orders of magnitude due to ever-increasing sophistication of the analytics algorithms.

- Composability and modularity are important aspects of scalability, where individual hardware and software components (perhaps numbering in thousands of types and millions of instantiations) are assembled into a fog network optimized to run the specific applications required.

Scalability enables fog nodes to provide basic support to address the business requirements and enable a pay-as-you-grow model for the FaaS, which is essential to the economics of its initial deployment and long-term success.

4.3 Openness Pillar

Openness is essential for the success of a ubiquitous fog computing ecosystem for IoT platforms and applications. Proprietary or single vendor solutions can result in limited supplier diversity, which can have a negative impact on system cost, quality and innovation. The openness pillar importance is highlighted in our desire for fully interoperable systems, supported by a vibrant supplier ecosystem.

Openness as a foundational principle enables fog nodes to exist anywhere in a network and span networks. This openness enables pooling by discovery, which means that new software-defined fog nodes can be dynamically created to solve a business mission. The security pillar shares a common theme and requirements in openness characteristics

- **Composability** supports portability and fluidity of apps and services at instantiation. Additional emphasis of composability is visible in the programmability pillar.
- **Interoperability** leads to secure discovery of compute, network, and storage and enables fluidity and portability during execution. The marketplace has clearly articulated its desire for a vibrant supplier ecosystem, with reasonable expectations that elements from one supplier can be freely substituted for elements from another supplier. This will be addressed through testbeds, FogFests (plug fest), standardization, and open implementations.
- **Open communication** enables features like pooling of resources near the edge of the network to collect idle processing power, storage capacity, sensing ability, and wireless connectivity. For example, a

compute-intensive application developed in fog architecture can leverage hundreds of gigabytes sitting idle on nearby laptops, systems, and set-top boxes in a household every evening, or among the passengers of a public transit system. The open discovery of these nearby compute resources is critical. Doing the functional work nearest the edge avoids additional network taxes when moving up the stack towards the cloud. We define network taxes as the cost of transmission.

- **Location transparency** of any node instance to ensure that nodes can be deployed anywhere in the hierarchy. Location transparency provides an alternative to network operator control. This means that any IoT device, such as a smart watch, does need its own carrier-owned data plan. Each thing or software entity can observe its local conditions and make decisions on which network to join. Each endpoint in a fog network can optimize its path to the computational, networking and storage resources it needs (no matter if those resources are in the hierarchical layers of the fog, or in the cloud).

4.4 **Autonomy Pillar**

The autonomy pillar enables fog nodes to continue to deliver the designed functionality in the face of the external service failures. In this architecture, autonomy is supported throughout the hierarchy. Decision making will be made at all levels of a deployment's hierarchy including near the device or higher order layers. Centralized decision-making in the cloud is no longer the only option. Autonomy at the network edge means intelligence from local devices and peer data can be used to fulfill the business' mission at the point where it makes the most business sense.

The OpenFog RA supports autonomy for a wide range of functions. It does not rely upon centralized entity for operation (e.g., a backend cloud). Some of the typical areas for autonomy at the edge include:

- **Autonomy of discovery** to enable resource discovery and registration. For example, an IoT device coming online in the field would typically "phone home" first to let the backend cloud know it is alive and its associated functions are available. But when an uplink network to the cloud is unavailable, it can stop the device from going live. An autonomous fog node can potentially act as a proxy for the device registration, which then allows the device to come online without the backend cloud.
- **Autonomy of orchestration and management (O&M)** automates the process of bringing services online and managing them through

the operational lifecycle and decommissioning. Autonomy of O&M entails a number of actions including: instantiation of services; provisioning the environment around the services, such as routing of data flows; and keeping track of the health and status of the resources. All these actions should be as automated as possible through programmability and policies. The architecture includes an autonomous and scalable O&M function that is set up to handle any surge of demand for resources, without real-time reliance on the cloud or the need for significant human labor.

- **Autonomy of security** enables devices and services to come online, authenticate themselves against a minimal set of fog security services, and perform their designed functions. In addition, these security services can store records for future audits. With autonomy, these actions can be performed where they are needed, when they are needed, and regardless of connectivity to the cloud. Fog nodes can autonomously react to evolving security threats, such as updating virus screening algorithms, determination of denial-of-service (DoS) attacks, etc. without administrator involvement.
- **Autonomy of operation** supports localized decision making by IoT systems. Sensors provide data, which is the basis for autonomous actions at the edge. If the cloud or a single place in the system's hierarchy is the only location where decisions can be made, this violates the ability to ensure reliability and as such, the architecture ensure operational autonomy.
- **Cost savings** is a key motivator for autonomy. Connectivity today costs money. The more data that is sent through the network, the higher the costs are for businesses due to network taxes. This drives the need for more processing at the edge of the network, with just-in-time and just-in-need data sent to the cloud as required for additional business insights. For example, when an oil rig generates 30,000 data points a second, not all of the data must be sent through an expensive satellite link. Local and fog domain analytics and pre-processing can autonomously filter out the unimportant data points and extract the more mission critical ones to be delivered to the next layer in the hierarchy.

A key aspect of fog computing is turning data into actionable wisdom. We term this DIKW, which stands for "**D**ata gathered becomes **I**nformation when stored and retrievable becomes **K**nowledge. Knowledge enables **W**isdom for autonomous IoT." This principle is the basis for localized analytics to enable autonomous decision making nearest the edge.

4.5 Programmability Pillar

The programmability pillar enables highly adaptive deployments including support for programming at the software and hardware layers. This means that re-tasking a fog node or cluster of fog nodes for accommodating operational dynamics, can be completely automated. The re-tasking can be done with the help of the fog nodes inherent programmability interfaces which we describe using general purpose compute or accelerator interfaces. Programmability of a fog node includes the following benefits:

- **Adaptive infrastructure** for diverse IoT deployment scenarios and support changing business needs.
- **Resource efficient deployments** maximizing the resources by using a multitude of features including containerization. This increases the portability of components and is a key design goal enabled by programmability.
- **Multi-tenancy** to accommodate multiple tenants in a logically isolated runtime environment.
- **Economical operations** that results adaptive infrastructure to changing requirements.
- **Enhanced security** to automatically apply patches and respond more quickly to evolving threats.

4.6 Reliability, Availability, and Serviceability (RAS) Pillar

Reliability, availability, and serviceability (RAS) is resident throughout successful system architectures and, as such, takes on great importance in the OpenFog RA. Hardware, software, and operations are the three main areas of the RAS pillar.

A reliable deployment will continue to deliver designed functionality under normal and adverse operating conditions. The reliability of the RAS pillar includes but is not limited to the following properties:

- Ensuring reliable operation of the underlying hardware upon which the software is operating, enabling reliable and resilient software and a reliable fog network, which is generally measured in uptime.
- Safeguarding the availability and integrity of data and compute on edge gateways using enhanced hardware, software, and network designs.
- Autonomous predictive and adaptive self-managing capabilities when required by the health of the system to initiate self-healing routines for

hardware and software and upgrade new firmware/application and security patches.

- Increasing customer satisfaction by simplifying support and device self-optimization and healing.
- Initiating requests for preventative maintenance, including new hardware and software patches, network re-routing, etc.
- Testing and validation of system components, including device drivers and diagnostic tools under a variety of environmental conditions.
- Providing alarms, reports, logs, etc.
- Validation of system platforms and architectures through interoperability certification test suites.

Availability ensures continuous management and orchestration, which is usually measured in uptime. The availability of the RAS pillar includes but not limited by the following properties:

- Secure access at all levels of a fog hierarchy for orchestration, manageability, and control, which includes upgradeability, diagnostics and secure firmware modification.
- Fault isolation, fault syndrome detection, and machine learning to help improve Mean Time To Repair (MTTR) of a failed system to achieve higher availability.
- Concept of cloud based back-end support with availability of interfaces throughout the system.
 - Secure remote access from a plurality of devices (not just a single console).
 - Redundant/duplicate device (peer-to-peer) IoT platform.
 - Mesh access capabilities of end-point sensor/peering.
 - Remote boot capabilities of the platform.
 - Modification and control from the lowest level firmware (BIOS) through to the highest software in the hierarchy (cloud).
 - Support for redundant configurations for persistent productivity.

Servicing a fog deployment ensures correct operation. Serviceability of the RAS pillar includes but is not limited by the following properties:

- Highly automated installation, upgrade, and repair to efficiently deploy fog computing at scale.
- Hardware or software can either autonomously heal or be serviced by the various manufacturers.
- Ease of use to accommodate maintenance.
- Serviceability of the system:
 - Hardware, software, applications, networking, and data

- Ease of access/swap-out of the hardware (component interoperability).
- Ease of secure upgradeability of software, BIOS, and applications locally or remotely and in real time.
- Replication of system configuration over cloud on replaced/swap-out systems.
- Support for redundant configurations for persistent productivity.

RAS is especially important for OpenFog RA deployments in harsh environmental conditions and remote locations. This is why aspects from RAS are found throughout the architecture.

4.7 Agility Pillar

The agility pillar addresses business operational decisions for an OpenFog RA deployment. It is not possible for humans alone to analyze the data generated at the scale predicted by IoT as the basis for rapid, sound business and operational decisions. The agility pillar focuses on transforming this volume of data into actionable insights. Agility also deals with the highly dynamic nature of fog deployments and the need to respond quickly to change.

Data generation by sensors and systems in an OpenFog RA deployment are turbulent, bursty, and are often created in huge volumes. Most importantly, data may not have context, which is created only when the data is collated, aggregated, and analyzed. The analysis of data can be executed at the cloud level, but this subjects the data to increasing levels of latency. The ideal approach is to make operational decisions as soon as data can be turned into a meaningful context. The architecture enables the creation of context close to the data generation where it makes the most sense for a given scenario. More strategic, system-wide decisions and policy management can be made further up the layers in the fog hierarchy. This avoids network dependencies we termed as “network taxes” as described in other OpenFog RA pillars.

OpenFog architectural approaches allow IoT system developers to optimize the placement of their applications as decision making components.

4.8 Hierarchy Pillar

Computational and system hierarchy is not required for all OpenFog architectures but it is still expressed in most deployments. The OpenFog architecture is complementary to traditional cloud architectures due in part to the OpenFog hierarchy pillar.

OpenFog RA computing resources can be seen as a logical hierarchy based on the functional requirements of an end-to-end IoT system. Depending on the scale and nature of the scenario being addressed, the hierarchy may be a network of smart and connected partitioned systems arranged in physical or logical layers, or it may collapse into a single physical system (scalability pillar).

Using building automation from smart cities as an example, a company that manages a single office complex may have the entire fog deployment located locally. A large commercial property management company may have distributed fog deployments at local and regional levels feeding information to centralized systems and services. Each fog node is autonomous (autonomy pillar) to ensure uninterrupted operations of the facility it manages.

The figure below shows a logical view of the IoT system from a computational perspective. Each layer in the hierarchy addresses a specific concern of the IoT system.

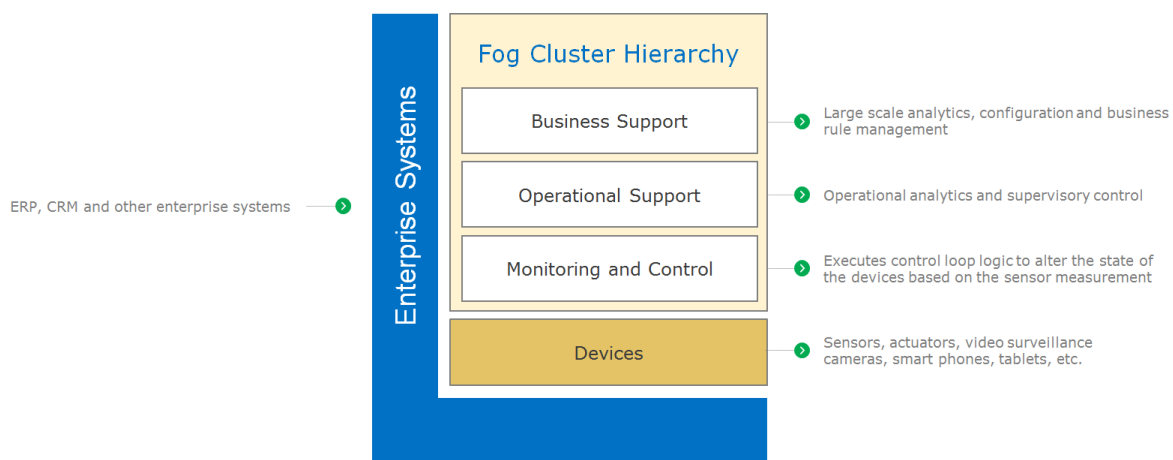


Figure 6 Layered Architecture View of an IoT System

Devices in the hierarchy: Sensors and actuator devices are the physical things and produce telemetry data to be consumed by the monitoring and control layer. This layer analyzes the telemetry and generates actuation commands if the process being monitored deviates from the desired state. Note that the term “process” is used in an abstract sense that it is represented by a set of measured parameters that depend on a set of actuator settings. Depending on the domain problem, some systems may not have any actuators. Similarly, for scenarios like mobile network acceleration, the core function is to accelerate content delivery and not

monitoring and control. On the other hand, systems like building operations may have actuators to change the HVAC and lighting based on occupancy.

Monitoring and control in the hierarchy: Sensors and actuators are connected to microcontrollers that are programmed to monitor and control the state of the processes. A process state is represented by a set of parameters measured by the sensors and modified by the actuators. The main responsibility of this layer is to execute control logic through stateful inspection of the sensor telemetry. This involves computing alarms and generating events, which may trigger workflows through machine-to-machine or human intervention.

Operational support in the hierarchy: The operational support layer is responsible for analyzing streaming telemetry and storing operationally oriented analytics. The analytics may be presented through interfaces like control room dashboards and mobile applications. The scope of the analytics at this layer is narrow; it focuses on the operational aspects of the physical environment for which the system is responsible. This layer combines drill down historical analytics with streaming analytics for a composite picture of real-time operations with some short-term history. The agility pillar is seen in the hierarchy as the implementation of complex event processing on the streaming telemetry data.

Surrogacy in the hierarchy: The computation of a complex operation in the fog nodes may be delegated to the hierarchical nodes to leverage adjacent resources. Consider virtual reality tasks associated with a wearable such as smart glasses. Some of the information retrieval and computation tasks may be carried out on the glasses, while an associated element in the hierarchy (e.g., a smartphone) may handle its storage and connectivity requirements. This hierarchical architecture may leverage all of these devices at the same time, with an intelligent division of labor across them.

Business support in the hierarchy: The primary responsibility of this layer is to store and analyze the entire history of the IoT operations that span multiple systems. This is the system of record for IoT operations as governed by the compliance and record retention policies. Petabyte scale analytics will help in mining insights, business planning, comparing the operational efficiency of processes, operational optimization through training machine learning models, etc. Additionally, metadata and reference data management, business rule management, and the operational health of lower layers are the other aspects of this layer. These are also viewed in the agility pillar.

4.8.1 Hierarchical Fog Deployment Models

The figures below show a subset of the combination of fog and cloud deployed to address various domain scenarios as framed by the layered view of IoT systems. Each fog element may represent a hierarchy of fog clusters fulfilling the same functional responsibilities. Depending on the scenario, multiple fog and cloud elements may collapse into a single physical deployment. Each fog element may also represent a mesh of peer fog nodes in use cases like connected cars, electrical vehicle charging, and closed loop traffic systems. In these use cases, fog nodes may securely discover and communicate with each other for exchanging context-specific intelligence.

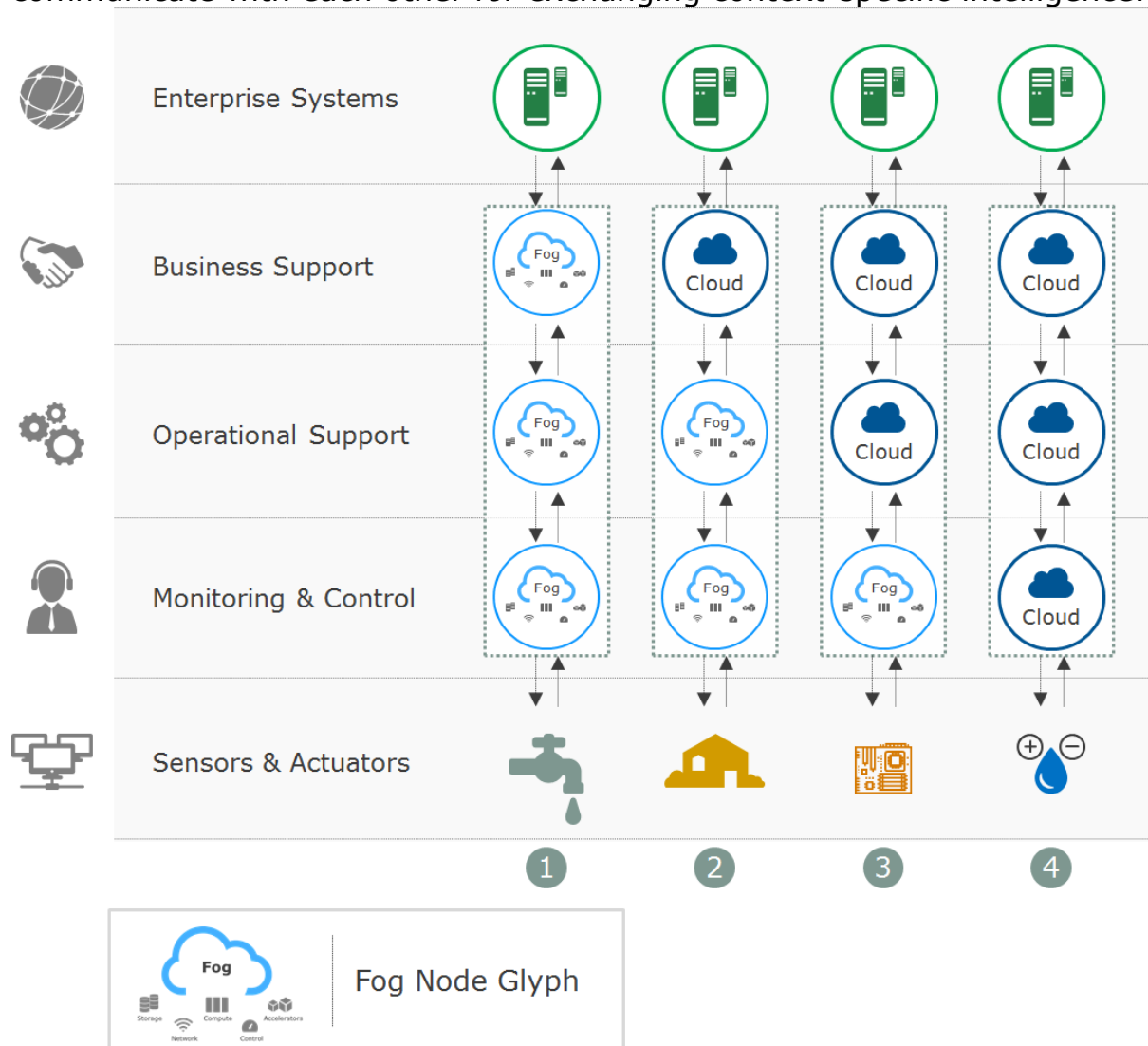


Figure 7 IoT System Deployment Models

Figure 1 shows a fog deployment hierarchy that is independent of the cloud. This model may be applicable for use cases where cloud can't be used for

reasons such as low event to action time window, regulatory compliance, military grade security and privacy, and unavailability of a central cloud in a particular geography. Examples include armed forces combat systems, drone operations, some healthcare systems, hospitals, and ATM banking systems.

Figure 2 shows the cloud used for information processing related to decision making that may have event-to-action time window ranging from hours to days to months. Operation-centric information processing is done by fog deployments located close to the infrastructure/process being managed. Use cases include commercial building management, commercial solar panel monitoring, and retail.

Figure 3 shows the local fog infrastructure used for time-sensitive computation, while the cloud is used for the balance of operational and business-related information processing. Use cases include commercial UPS device monitoring, mobile network acceleration, and content delivery networks (CDNs) for Internet acceleration.

Figure 4 supports use cases like agriculture, connected cars, and remote weather stations. These use cases leverage the cloud for the entire stack due to the constrained environments in which the deployment of fog infrastructure may not be feasible or economical. Fog nodes at the device layer may get some of the monitoring and control function for safety related control. The enterprise systems integrate with cloud for business operations.

Note that the functional boundaries shown in Figures 1-3 are fluid and can be physically deployed in multitude of combinations based on the architecture of the domain-specific solutions. In real world deployments as we discussed earlier, there may be many combinations of physical deployments that involve multi-tenants, fog, and cloud deployments owned by multiple entities. The following diagram illustrates some those models:

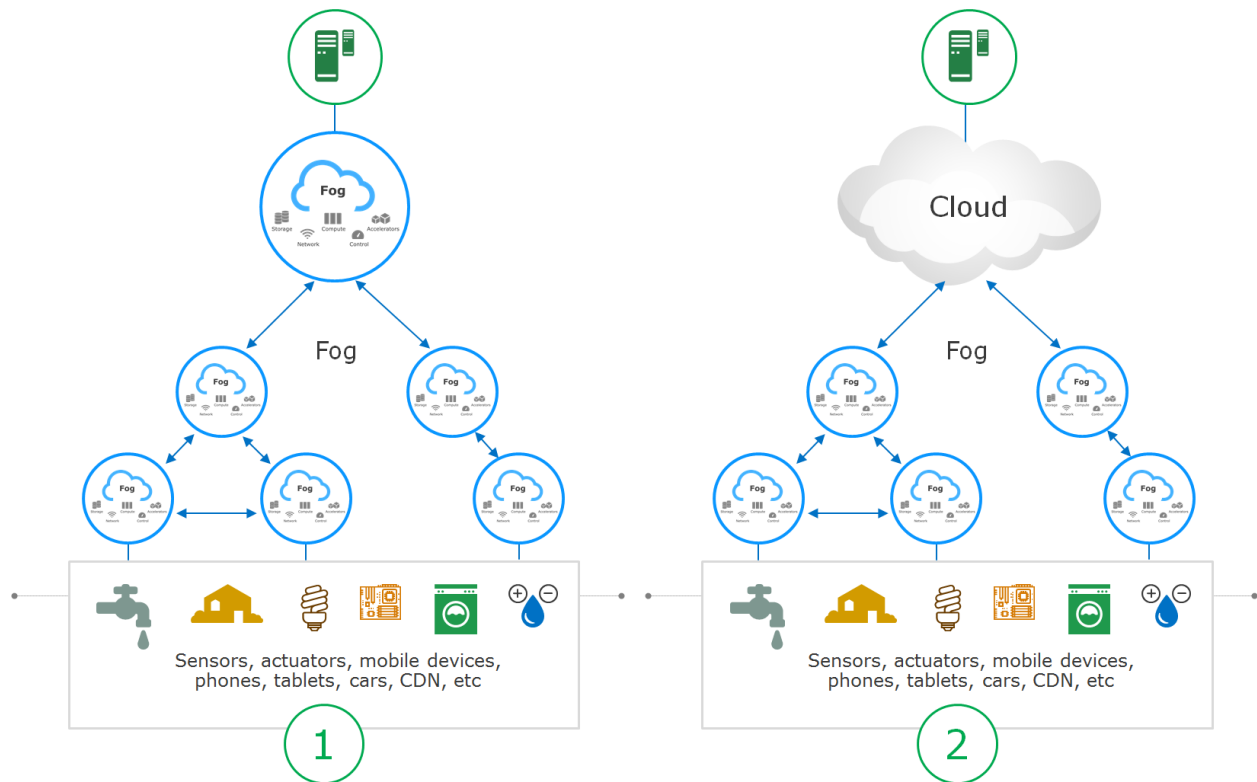


Figure 8 Fog Hierarchy Example

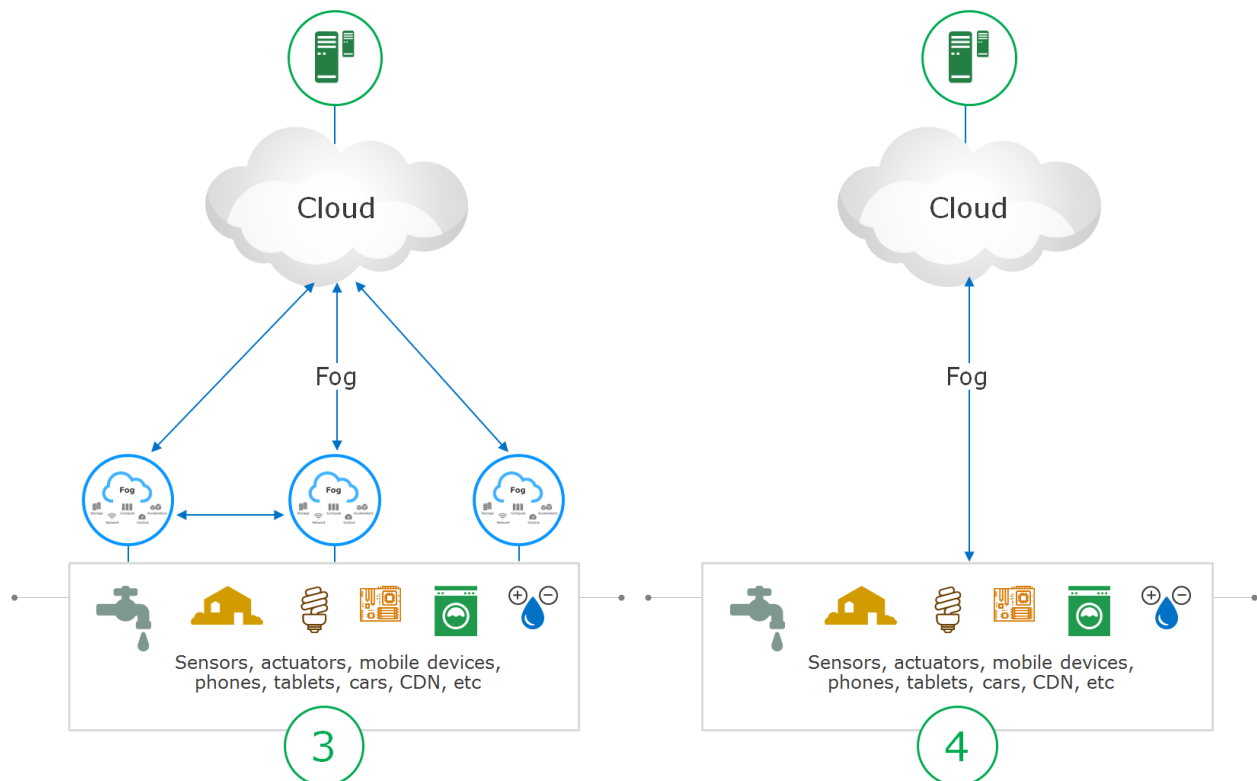


Figure 9 Fog Hierarchical Deployment Model

Many of the usages will occur as represented in Figures 2 and 3. The three-layer fog hierarchies shown here are for illustrative purposes only. Real-world fog deployments may have more or fewer levels. Different vertical application use cases may use a fog hierarchy differently. For example, in a smart city, there may be fog nodes in a region, neighborhood, street corner, and building level. In a smart factory, the hierarchy may be divided by assembly lines, manufacturing cells, and machines.

5 Reference Architecture Overview

The OpenFog reference architecture is based on the eight pillars described in the previous chapter. We use the ISO/IEC/IEEE 42010:2011 international standard as the guideline for describing architecture to stakeholders. The standard enables a common vocabulary across IoT to help support cross-organizational technical collaboration. These are some of the shared vocabulary terms used in the OpenFog RA:

- **OpenFog Architecture Description:** is an abstract representation of an instance of a fog node. It is a composite of multiple views we used to address stakeholders in the fog computing value chain.
- **Viewpoint:** A viewpoint is a way of looking at a system. These included but are not limited to Functional and Deployment viewpoints.
- **View:** A view is a representation of one or more structural aspects of the architecture. In the current revision of the OpenFog RA, the structural aspects are the Software view, System view, and Node view.
- **Perspective:** A perspective is a cross-cutting concern of the architecture.

5.1 Functional Viewpoint

The functional viewpoint of the architecture shows how we apply the OpenFog architectural elements and views to address the various concerns of the stakeholders to satisfy a given scenario. Each scenario chosen will focus on a different aspects and market opportunities for fog computing. We fully expect that the architectural description, views, and perspective may change over time. This change and refinement should be driven from testbeds and application of the OpenFog RA to multiple different markets we see as important to fog computing.

The first end-to-end scenario we will address is the visual security scenario. Our intent will be to define how that scenario works, and then work through various testbeds associated with those aspects to validate or modify the architecture.

5.2 Deployment Viewpoint

5.2.1 OpenFog Deployment Types

How fog software and fog systems are deployed to address a given scenario is also very important. There are many deployment types ranging from embedded to large clustered systems that are fully interconnected. The deployment type(s) chosen are scenario specific, but key aspects of the architecture remain visible regardless of deployment type. However, some may grow or shrink in importance.

5.2.2 N-Tier Fog Deployment

In most fog deployments, there are usually several tiers (N-tiers) of nodes. For this example, we will use a simple food processing plant to help solidify the logical tiers. There is a conveyor belt on which food is processed before moving on to the next level of packaging and shipment.

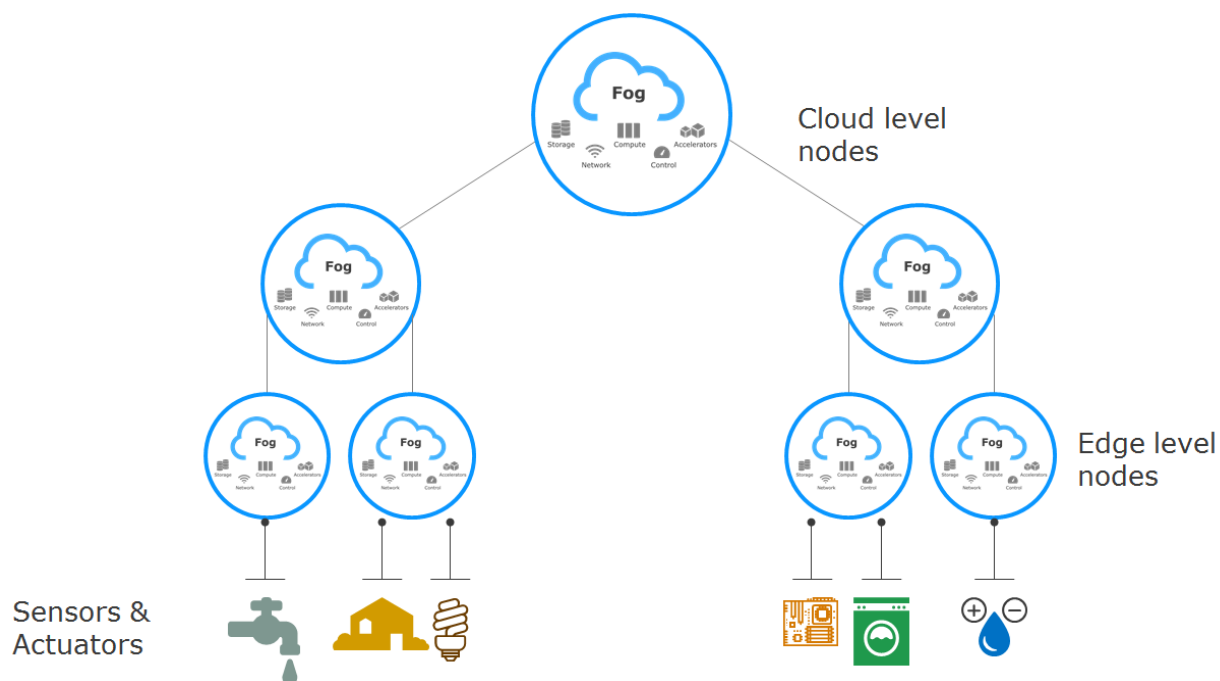


Figure 10 Multi-Tier Deployment

- Nodes at the edge are typically focused on sensor data acquisition/collection, data normalization, and command/control of sensors and actuators.
 - Using our example, the fog nodes nearest the physical technology edge (at the conveyor belt) will need to operate at

millisecond and sub-millisecond granularity from sensing to actuation to avoid product contamination and to ensure safety of operation.

- Nodes in the next higher tier are focused on data filtering, compression, and transformation. They may also provide some edge analytics required for critical real time or near real time processing. As we move away from the true network edge, we see higher level machine and system learning (analytics) capabilities.
 - Using our previous example, we can easily see that this tier needs to operate on a slightly higher level. It should be ensuring that the conveyor belt and others around it are operating more efficiently. This may still be in the latency of milliseconds. The key aspect of OpenFog RA to enable migration of applications, and functionality between the lowest tiers and the middle tiers as required by the scenario.
- Nodes at the higher tiers or nearest the backend cloud are typically focused on aggregating data and turning the data into knowledge. It's important to note that the farther from the true edge, the greater the insights that can be realized.

Note: In some deployment models, some degree of analytics may be located in nodes at the edge of the network (e.g. video analytics on surveillance cameras). This is because the network pipes may not be large enough to cost effectively carry the raw sensor data to higher layer fog nodes for processing. In reality as computational capabilities grow, the analytics functions at the lower tiers will grow. This will enable the overall growth of intelligence of fog deployments over time.

Machine Learning is in the forefront of research today that requires computation for both training models, and inference or scoring those models for close to real time response at the edge. We could use machine learning to optimize operations at a train station in a smart city. In the train station, we could monitor and sense occupancy, movement, and overall system usage and over time adapt our infrastructure to determine how to most efficiently use it.

Using the smart city as a continued example, by including more buildings and having them talk to each other and using another tier above them to gain additional insights, city blocks can operate more efficiently. Learnings from blocks of buildings provide insights on how to make the overall city more efficient and so on and so forth. The key message is that as you move farther away from the true network edge, you can gain operational insights and increase the overall system intelligence. Additionally, migration of data

between these fog nodes in both the horizontal and vertical increases the system performance and operational capabilities.

5.2.2.1 How Use Cases Determine the Number of Tiers

Fog deployments will come large-scale and small, based upon the given scenario being addressed. The number of tiers in a fog deployment will be dictated by the scenario requirements, including:

- Amount and type of work required by each tier
- Number of sensors
- Capabilities of the nodes at each tier
- Latency between nodes and latency between sensors and actuation
- Reliability/availability of nodes

In general, each level of the N-tier environment would be sifting and extracting meaningful data to create more intelligence at each level. Tiers are created in order to deal efficiently with the amount of data that needs to be processed and provide better operational and system intelligence. At the highest level this can be represented by the figure below.

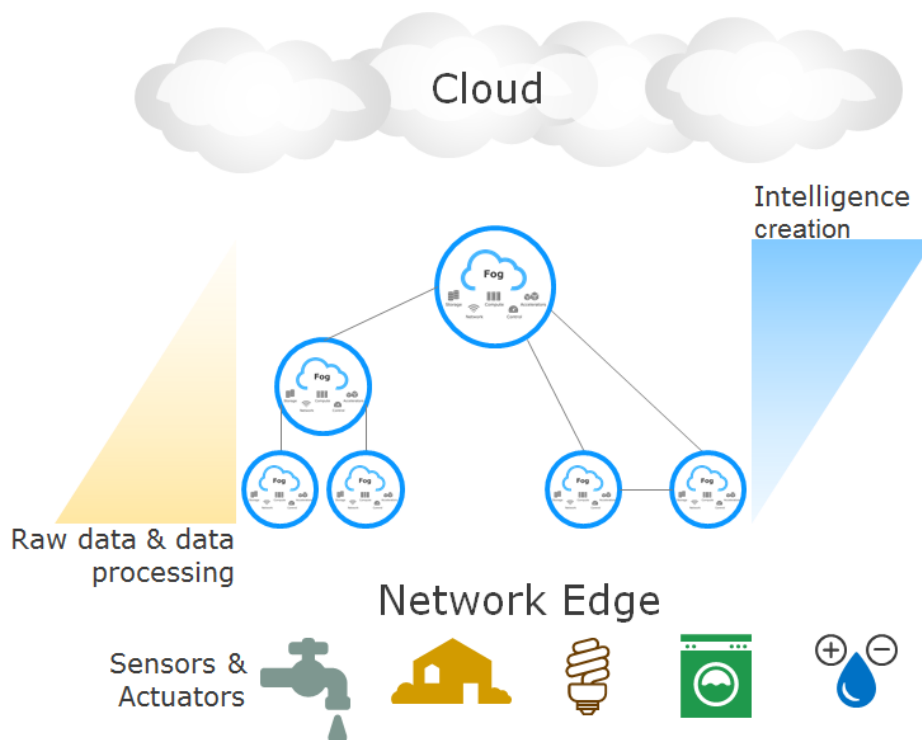


Figure 11 Intelligence from data

Important aspects we will address overtime is that the software running at every level and each node should be enabled to migrate across nodes, span