CCIE and CCDE Evolving Technologies Study Guide

Brad Edgeworth, CCIE No. 31547
Jason Gooley, CCIE No. 38759
Ramiro Garza Rios, CCIE No. 15469

Cisco Press

CCIE and CCDE Evolving Technologies Study Guide

Brad Edgeworth, CCIE No. 31547
Jason Gooley, CCIE No. 38759
Ramiro Garza Rios, CCIE No. 15469

Cisco Press

CCIE and CCDE Evolving Technologies Study Guide

Brad Edgeworth, Jason Gooley, Ramiro Garza Rios

Published by:
Cisco Press

01 18

ISBN-13: 978-0-7897-5972-6
ISBN-10: 0-7897-5972-1

Warning and Disclaimer

This book is designed to provide information about Evolving Technologies in the CCIE and CCDE written certification exam. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

About the Authors

Brad Edgeworth, CCIE No. 31574 (R&S & SP), is a Systems Engineer at Cisco Systems. Brad is a distinguished speaker at Cisco Live, where he has presented on various topics. Before joining Cisco, Brad worked as a network architect and consultant for various Fortune 500 companies. Brad's expertise is based on Enterprise and Service Provider environments with an emphasis on architectural and operational simplicity. Brad holds a Bachelor of Arts degree in Computer Systems Management from St. Edward's University in Austin, Texas. Brad can be found on Twitter as @BradEdgeworth.

Jason Gooley, CCIE No. 38759 (R&S & SP), is a very enthusiastic and spontaneous person who has over 20 years of experience in the industry. Currently, Jason works as a Technical Solutions Architect for the Worldwide Enterprise Networking Sales team at Cisco Systems. Jason is very passionate about helping others in the industry succeed. In addition to being a Cisco Press author, Jason also contributes to the development of CCIE exams, provides training for Learning@Cisco, is an active CCIE mentor, a committee member for the Cisco Continuing Education Program (CE), and also a Program Committee member of the Chicago Network Operators Group (CHI-NOG, www.chinog.org).

Ramiro Garza Rios, CCIE No. 15469 (R&S, SP, and Security), is a Solutions Integration Architect with Cisco Advanced Services.

His expertise is on Enterprise and Service Provider network environments with a focus on evolving architectures and next-generation technologies. He is also a Cisco Live distinguished speaker. Ramiro is currently working on a multiyear Cisco Application Centric Infrastructure (ACI) project for one of the top three Tier 1 ISPs in the United States.

Before joining Cisco Systems in 2005, he was a network consulting and presales engineer for a Cisco Gold Partner in Mexico, where he planned, designed, and implemented both Enterprise and Service Provider networks.

About the Technical Reviewer

David Hanes, CCIE No. 3491, is a Principal Engineer in Cisco System's Cloud Support Technical Assistance Center (TAC). Specializing in the Internet of Things (IoT) and Collaboration technologies, he assists in escalated customer issues and the incubation of new products and solutions. David has authored various industry publications in his areas of expertise, including the Cisco Press books *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things* and *Fax, Modem, and Text for IP Telephony*. He has spoken at industry conferences around the world and is a Cisco Live Hall of Fame Speaker. He has worked on various standardization efforts, including leading and participating in working groups with the SIP Forum and authoring and contributing to RFCs in the IETF. David also has over a dozen patents issued and pending related to IoT, Collaboration, and other computer networking technologies. He holds a B.S. in Electrical Engineering from North Carolina State University.

Dedications

This book is dedicated to the memories of my father, David Edgeworth. While you are no longer present, I still feel your impact every day. Thank you for everything that you have given me. May you rest in peace.

—Brad Edgeworth

I would like to dedicate this book to my family. To my wife Jamie for supporting me through this process (again), even though you are currently pregnant with our son Jaxon. Thank you for letting me jump behind the keyboard once again! To my daughter Kaleigh, who is now almost 4 years old. You are growing so fast. Never give up on what you want. If at first you don't succeed, try and try again. I love you more than anything and I can't wait to finish this dedication so I can spend more time with you! To my son Jaxon, I love you so much and you aren't even here yet! In order to be great, one must make great sacrifices. To my father and brother, thank you for always encouraging me to just jump in and do it. To my late mother, you are still the guiding light that keeps me on the right path. To the rest of my family, I love you!

—Jason Gooley

I would like to dedicate this book to my wife Mariana, who was extremely supportive throughout this process and for providing constructive criticism on my artwork. I would also like to dedicate this book to my four kids, Ramiro, Frinee, Felix, and Lucy, for putting up with me while I was physically present but mentally absent writing this book. And last but not least, to my parents and my in-laws for their ongoing love and support.

—Ramiro Garza Rios

Acknowledgments

Brad Edgeworth:

Jason and Ramiro, thanks for helping me out on this project. I am privileged enough to know you, let alone work with you.

This is the part of the book that you look at to see if you have been recognized. Well, many people have provided feedback, suggestions, and support to make this a great book. Thanks to all who have helped in the process, or even in educating me, especially Brett Bartow, Dan Wiggins, Dan Wasson, Carlos Rojas, Darryl McCartney, Dan Behrens, and my managers.

Jason Gooley:

First, thank you to Brad and Ramiro. I had a blast working on this project with you! Thank you Brett Bartow and the rest of the Cisco Press team for all of the support during the creation of this book. It was a pleasure to have the chance to work with you all again!

I would like to thank the entire GSD team for supporting me during this process. In no particular order, thank you Andre Laurent, Tyler Creek, Walt Sacharok, David Prall, Nicole Wajer, Dax Mickelson, Dmitry Figol and Stephanie Anderson. This team is a big part of my family and I love you all!

A special thanks to Jim Cook for being my huckleberry, calm voice of reason, and helping me stay on course. To Luke Kaelin for always being there to keep me sane and to keep me laughing. To my friend Vince Baldocchi for all the kind words and support. I can't thank you all enough for always believing in me.

Lastly, I would like to thank my wife Jamie again. Without you, none of this would be possible. You mean the world to me. I love you!

Ramiro Garza Rios:

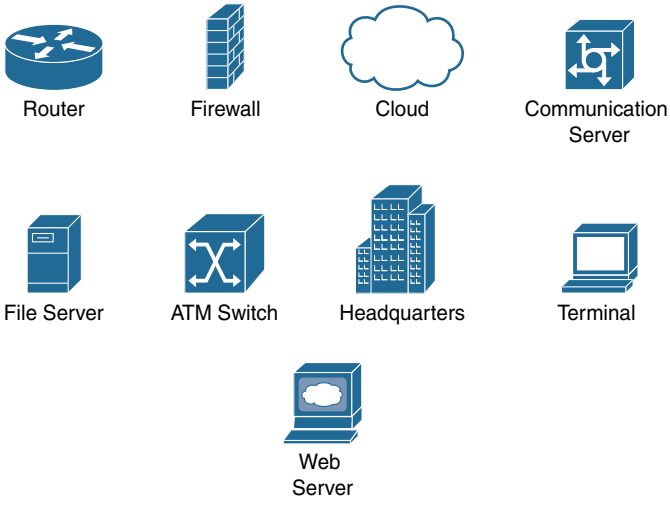
Brad, thank you for inviting me to participate in this exciting project and for the chance to work with you once again.

A big thank you to the Cisco Press team for your ongoing support, and a special thanks to Brett Bartow for guiding us through the creation of this book and for helping us stay focused and on track.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Credits

Final Version of NIST Cloud Computing Definition Published, NIST, October 25, 2011.

Table 2-1 The Treacherous 12 – Top Threats to Cloud Computing in 2016.

Table 2-2 Cisco Systems, Inc

Figure Credits

Figure 1-2 Cisco Systems, Inc

Figure 1-7 Screenshot of Manufacturing cells © Cisco Systems, Inc

Figure 2-16 Cisco Systems, Inc

Figure 2-33 Cisco Systems, Inc

Figure 3-1 Screenshot of CLI Cisco © Cisco Systems, Inc

Figure 3-3 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-4 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-5 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-6 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-7 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-8 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-9 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-10 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-11 Screenshot of Google Postman dashboard © 2018 Postdot Technologies, Inc.

Figure 3-14 Screenshot of DEVNET main page © Copyright 2018 Cisco DevNet

Figure 3-15 Screenshot of DEVNET main page © Copyright 2018 Cisco DevNet

Figure 3-16 Screenshot of DEVNET main page © Copyright 2018 Cisco DevNet

Figure 3-17 Screenshot of DEVNET main page © Copyright 2018 Cisco DevNet

Figure 3-21 Screenshot of GitHub main webpage © 2018 GitHub, Inc.

Figure 3-22 Screenshot of GitHub main webpage © 2018 GitHub, Inc.

Figure 3-23 Screenshot of GitHub main webpage © 2018 GitHub, Inc.

Figure 3-24 Screenshot of GitHub main webpage © 2018 GitHub, Inc.

Figure 3-27 Screenshot of YAML Lint example © YAML

Figure 3-28 Screenshot of YAML Lint example © YAML

Figure 3-29 Screenshot of YAML Lint example © YAML

Introduction

Cisco is once more leading the way in building a workforce capable of moving with technological changes through the evolution of its certification programs. Changes to the Expert-Level (CCIE/CCDE) programs will enable candidates to bridge their core technology expertise with knowledge of the evolving technologies that organizations are adopting at an accelerated pace, such as cloud, IoT, and network programmability.

Combining this book with the other Cisco Press certification books that are written for a specific track will provide a complete source of knowledge to help CCIE and CCDE candidates succeed on their written exams.

Goals and Methods

The most important and somewhat obvious goal of this book is to help you pass the written CCIE and CCDE exams. One key methodology used in this book is to help you discover the exam topics that you need to review in more depth, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. This book does not try to help you pass by memorization, but helps you truly learn and understand the topics.

Who Should Read This Book?

This book is not designed to be a general networking topics book, although it can be used for that purpose. This book is intended to tremendously increase your chances of passing the evolving technologies components of the CCIE and CCDP written exams. Although other objectives can be achieved from using this book, the book is written with one goal in mind: to help you pass the exam.

How This Book Is Organized

Although this book could be read cover to cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover just the material that you need more work with.

The chapters, Chapters 1 through 3, cover the following topics:

- **Chapter 1, "Internet of Things"—**This chapter discusses the Internet of Things (IoT) from a perspective of business transformations, connectivity, and methods of securing it. Most IoT networks share similarities with Enterprise Networks, but are different in behavior and operational aspects.
- **Chapter 2, "Cloud Fundamentals"—**This chapter provides a holistic overview of cloud environments using virtual machines (VMs) or containers in a public, private, or hybrid model. Topics include cloud service models, connectivity, security, scalability, and high availability designs.

- **Chapter 3, "Foundational Network Programmability Methods"—**This chapter covers modern programmability and automation methods that can be used to interact with different applications and devices through the use of APIs. This chapter also focuses on the Cisco DevNet developer community as well as other important tools to help readers on their programmatic journey.

Certification Exam Topics and This Book

The questions for each certification exam are a closely guarded secret. However, we do know which topics you must know to *successfully* complete the evolving technologies portion of all CCIE/CCDE-level written exams. Cisco publishes them as an exam blueprint for CCIE/CCDE Evolving Technologies. Table I-1 lists each exam topic listed in the blueprint along with a reference to the book chapter that covers the topic. These are the same topics you should be proficient in when working with Cisco wireless LANs in the real world.

Table I-1 *CCIE/CCDE Evolving Technologies Topics and Chapter References*

CCIE/CCDE Evolving Technology Exam Topic		Chapter(s) in Which Topic Is Covered
1.0	**Cloud**	
1.1	*Compare and contrast public, private, hybrid, and multicloud design considerations*	2
1.1.a	Infrastructure, Platform, and Software as a Service (XaaS)	2
1.1.b	Performance, scalability, and high availability	2
1.1.c	Security implications, compliance, and policy	2
1.1.d	Workload migration	2
1.2	*Describe cloud infrastructure and operations*	2
1.2.a	Compute virtualization (containers and virtual machines)	2
1.2.b	Connectivity (virtual switches, SD-WAN, and SD-Access)	2
1.2.c	Virtualization functions (NFVi, VNF, and L4/L1)	2
1.2.d	Automation and orchestration tools (CloudCenter, DNA Center, and Kubernetes)	2
2.0	**Network Programmability**	
2.1	*Describe architectural and operational considerations for a programmable network*	3
2.1.a	Data models and structures (YANG, JSON, and XML)	3
2.1.b	Device programmability (gRPC, NETCONF, and RESTCONF)	3

CCIE/CCDE Evolving Technology Exam Topic		Chapter(s) in Which Topic Is Covered
2.1.c	Controller-based network design (policy-driven configuration and northbound/ southbound APIs)	3
2.1.d	Configuration management tools (agent and agent-less) and version control systems (Git and SVN)	3
3.0	**Internet of Things**	
3.1	*Describe architectural framework and deployment considerations for Internet of Things (IoT)*	1
3.1.a	IoT technology stack (IoT network hierarchy, data acquisition, and flow)	1
3.1.b	IoT standards and protocols (characteristics within the IT and OT environment)	1
3.1.c	IoT security (network segmentation, device profiling, and secure remote access)	1
3.1.d	IoT edge and fog computing (data aggregation and edge intelligence)	1

Chapter 1

Internet of Things

This chapter covers the following topics:

- Business Transformation and Digitization
- IoT Fundamentals
- Securing IoT

The *Internet of Things,* also known as *IoT*, has become the latest industry buzzword. So what exactly does it mean? At its simplest, IoT is a network of *things* (or devices) that traditionally are not a part of a computer network (printers, laptops, servers, or cell phones). With all these things connected to the network, data can be collected off these *things* and extrapolated in ways that were almost impossible to imagine before. This information can be used to change the ways people have done things in the past and improve the way we live, work, play, and learn.

Business Transformation and Digitization

Over the past years, specific companies have dominated their respective markets. The barrier to entry has been high to enter a market, thus making it difficult for new companies.

Entrepreneurs realized that starting a new business in these markets required a new business model to overcome these barriers. By incorporating digitization—the act of transforming information into a digital format—new business models can be developed and thus change the paradigm. Digitization allows new companies to provide new benefits to customers while allowing them to compete with large, well-established businesses.

For example, ride-share companies like **Uber** and **Lyft** have revolutionized and transformed the taxi industry with the use of smartphones. Customers request a ride from their phone, are picked up, and are then transported to their destination seamlessly.

The process of hailing a taxicab or trying to find a taxicab company (which varies from city to city) is simplified. Drivers can choose to work a number of hours based on their availability, making it easy for them to maintain work-life balance—and making it more desirable employment than working for a cab company. Digitization offers all these benefits to customers and employees while providing an experience that is better, faster, and cheaper than taking a traditional taxi.

Other companies like **Airbnb** and **VRBO** have disrupted the hotel markets. In fact, most companies have realized that they need to embrace the digitization process and look at other business models so that they can maintain their existing customer base while acquiring new customers through different business use cases.

Digitization is occurring in all market verticals, including banking, healthcare, manufacturing, utilities, real estate, mining, and even government municipalities. In all these markets, organizations are collecting more data, analyzing the data in real time, and taking action based on the real-time analysis of that data.

IoT technologies are a component of a company's digitization architecture. The following market segments are already deploying IoT technologies:

- Manufacturing
- Mining
- Oil and gas
- Utilities
- Smart buildings
- Health and medical
- Retail
- Hospitality
- Transportation
- Connected cities and emergency services

IoT Fundamentals

The first computer network derives from the mainframe architecture. Mainframe computers contained massive amounts of processing power, storage, and memory. Mainframes were very expensive to acquire and operate, but they could run multiple programs simultaneously, making them cheaper than other computer systems at the time. Direct console access to the mainframe was limited, and users were force to connect to the mainframe from "thin" (dumb) terminal clients via the first computer network.

As processors became more powerful and easier to manufacture, personal computers provided a lot of functionality at a smaller cost, allowing smaller companies to have access to the benefits of a computer without the costs associating with owning and

maintaining a mainframe. Computer networks evolved as they no longer connected thin terminal clients in a small network, instead connecting multiple devices spread across large geographical boundaries.

The protocols to communicate between devices evolved and became more complicated, and they could no longer be managed by the mainframe team. Today, most corporate computer networks are managed by the *information technology (IT)* departments.

Just as the design of these IT networks evolved as more and more computers were connected to them, the network architecture for IoT networks must evolve too. Traditional IT networks follow the common Access-Distribution-Core model that can connect thousands of devices.

An IoT network is often associated with *operational technology (OT)*, which is defined as the hardware/software that influences physical processes through direct monitoring and/or control. OT networks exist in manufacturing facilities. For example, a programmable logic controller (PLC) is responsible for controlling the movement of motors in robots or conveyors based on feedback from sensors. The OT network connects the PLCs, robots, and sensors together.

An IoT network must be able to scale to handle hundreds of thousands of devices or possibly more. IoT networks become more complicated because they can be large scale (hundreds of thousands) and consist of different devices that communicate with different protocols.

Note OT and IoT networks are often maintained by engineers who are not from IT departments and hence might not be familiar with traditional IT best practices.

IoT Architecture Models

IoT networks are similar to the first computer networks, which ran multiple network protocols like IPX/SPX, NetBEUI, and TCP/IP. IoT networks must support a range of devices that use a variety of protocols. This is because the hardware refresh cycle is longer than corporate IT systems. As new systems are installed in some parts of a facility, the older systems using different protocols are not updated. Connecting devices that use different protocols requires some skill, and the architecture must also be able to scale to thousands of things while providing security.

Some common IoT architecture models are discussed next.

Machine-to-Machine (M2M) IoT Architecture

In 2013, the European Telecommunications Standards Institute (ETSI), 13 of the founding members, launched an initiative to generate a reference M2M IoT architecture. A key concept was the use of stacks, similar to that of the Open Systems Interconnection (OSI) model used in networking. A key benefit to using stacks is that a component in a stack can be swapped out without impacting the components of the stack next to it.

This provides benefits in the IoT space, considering that multiple things made by different manufacturers often must communicate using different protocols. The M2M IoT architecture focuses on IoT services, applications, and networks by providing interoperability through a variety of application programming interfaces (APIs):

- The **IoT M2M application layer** is focused on providing connectivity of things to the applications. A key component is that the application layer is responsible for tying into other business intelligence (BI) systems.
- The **services layer** refers to the logical components of the network and the management protocol they use.
- The **network layer** refers to the realm in which the things and other devices communicate. It includes the physical network that links them together.

Figure 1-1 demonstrates the machine-to-machine IoT reference architecture.

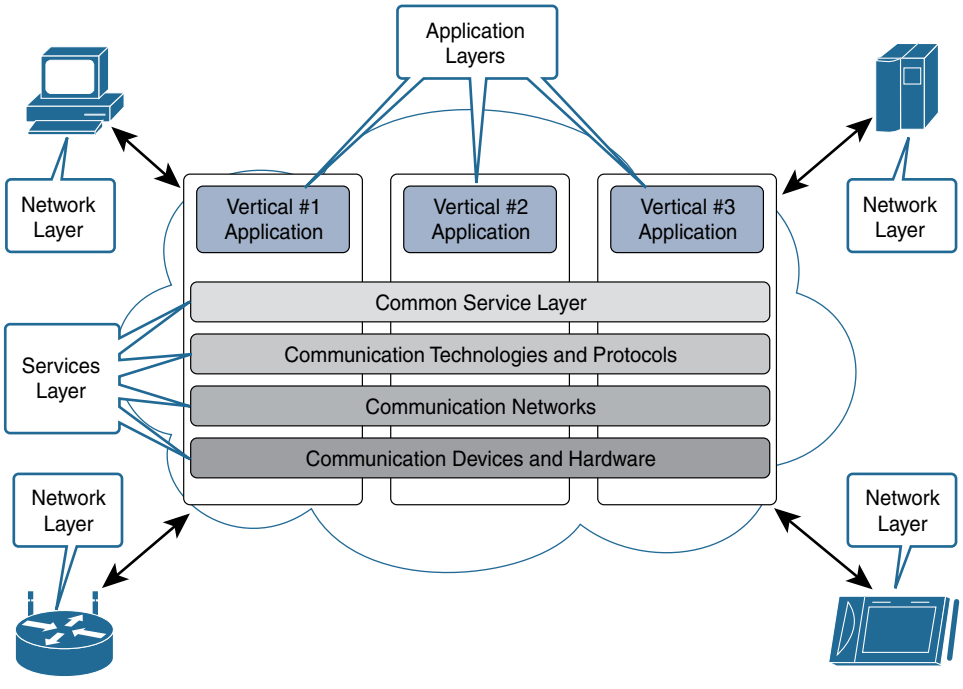


Figure 1-1 *Machine-to-Machine IoT Architecture*

The IoT World Forum (IoTWF) Architecture

In 2014, while in Chicago, the IoTWF created a seven-layer architectural reference model. This model provides clarity and simplicity through visualization of all the

IoT components: **sensors**, **network**, **computing**, and **storage**. The model consists of the following seven layers:

- **Physical Devices (Layer 1):** The bottom layer, which contains things (devices, sensors, and so on).
- **Connectivity (Layer 2):** This layer provides connectivity among things in Layer 1, Layer 2, and Layer 3. Communications and connectivity are concentrated in this one level.
- **Edge Computing (Layer 3):** The functions in this level are determined by the need to convert data into information that is ready for processing at a higher level.
- **Data Accumulation (Layer 4):** This layer is responsible for storing data that was traditionally transmitted live across the wire. The storage of the data allows for analysis or computation at a later time.
- **Data Abstraction (Layer 5):** This layer is responsible for rendering data and its storage in ways that enable the development of faster or simpler applications. It is responsible for reconciling multiple data formats from different sources, assuring consistent semantics, and confirming that the data is complete.
- **Application (Layer 6):** This layer is where information is analyzed and interpreted.
- **Collaboration and Processes (Layer 7):** This layer encompasses people and processes. In essence, this layer is responsible for providing people the right data, with the right analysis, at the right time so that they can engage the correct process.

Figure 1-2 demonstrates the IoT World Forum Reference Architecture.

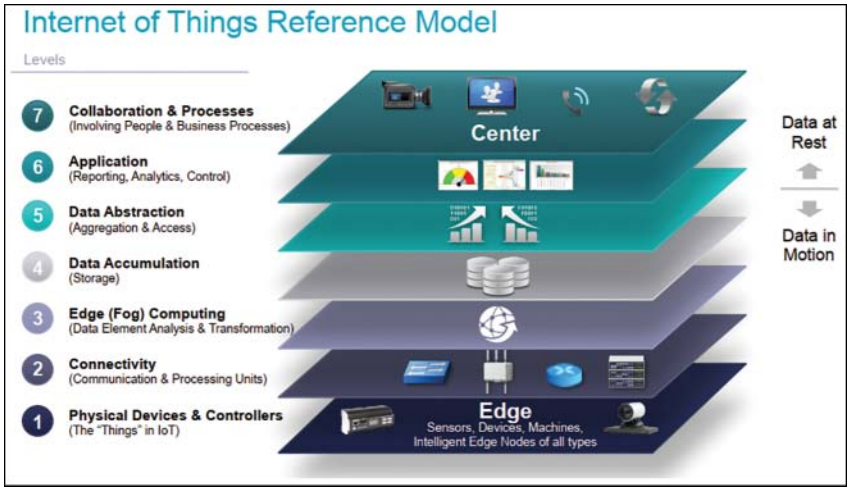


Figure 1-2 *IoT World Forum Reference Model*

Common IoT model

Multiple other IoT reference models have continued to evolve over the years. Some of the models have minor differences between them, whereas others have major differences in how they interact. A simplified IoT Architecture consists of the following:

- Things
- Communications
- IoT applications

Figure 1-3 demonstrates the components of the Common IoT model that will be discussed in the following sections.

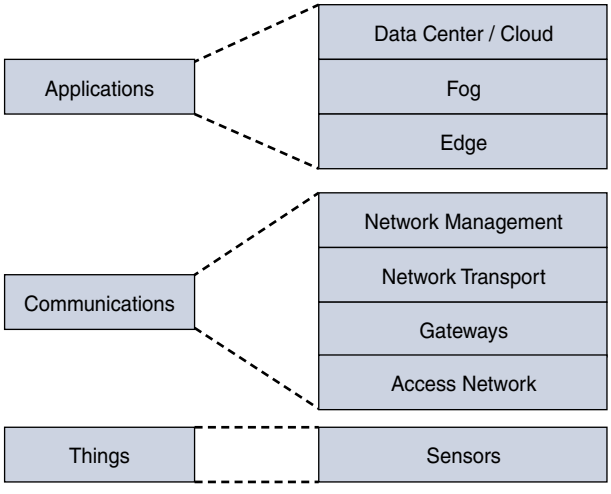


Figure 1-3 *Common IoT Model*

Things

As a simple definition, a thing is a device that provides relevant information. The intelligence on a thing can vary from device to device. Communication between things can occur in the following manners:

- **Unidirectional:** A temperature sensor collects the current temperature from its local thermometer and sends the temperature to a management device for analysis.
- **Local:** A manual thermostat can collect the current temperature for its internal thermometer and then turn on or off an HVAC (heating, ventilation, and air conditioning) unit.
- **Bidirectional:** An electronic thermostat can collect the current temperature from its internal thermometer and then send the data to a management device for analysis. The management device could send a signal back to the thermostat to turn on or off the HVAC unit.

A **smart thing** (also known as a smart object) can take relevant information from the data it receives and take appropriate action after analysis of that information. Some of the newer wireless electronic thermostats are examples of a smart things because they have the capability to be programmed remotely.

Some electronic thermostats can poll a remote temperature sensor and average the temperature of its local sensor to identify the average temperature before turning the HVAC unit on or off. To some people that might not seem very "smart." However, what if the electronic thermostat could check two different types of sensors? The electronic thermostat could check a temperature sensor and a sensor attached to a door to see if the door is open. If the door is open, turning on the HVAC system would result in a waste of electricity, and now the thermostat has relevant information about whether the cooled air would just flow out through the open door.

Here are some of the common sensors used in an IoT infrastructure:

- **Temperature**: These sensors collect temperature and report temperature at set intervals.
- **Vibration**: These sensors collect movement across one, two, or three axes. Vibration sensor is a generic term for proximity sensor, displacement sensor, and accelerometer. The frequency of the vibration and application depends on which sensor is used.
- **Pressure**: These sensors can report back liquid or air pressure.
- **Air quality**: These sensors can report air quality, or the chemical breakdown in a specific area. They can be used to detect chemicals that are harmful to humans without protective gear, an improper balance of chemicals for a specific manufacturing process, or the release of toxins into the air that do not meet governmental standards so that the general public can be informed.
- **Water quality**: These sensors report water quality, or the chemical breakdown in the water. Water is a vital component for manufacturing. It can be used for cooling of equipment, for washing of material as a phase of manufacturing, and for the cleanup process. Generally, manufacturers try to reuse their water supply when they can, but certain processes require the water to have an acceptable level of nonstandard components, which could result in the water needing further treatment before reuse.
- **Energy consumption**: Energy consumption is observed and reported as electricity is consumed from the perspective of a wire, electrical component (such as a circuit breaker), or the motor itself.
- **Location**: Location sensors could be GPS sensors that provide specific latitude and longitude, or they could be enabled with a unique wireless signature (Wi-Fi, Bluetooth, and so on) where they can be detected via antennas by an existing wireless infrastructure, and the location can then be calculated using triangulation based on signal strength at multiple radio towers.
- **Occupancy**: Occupancy sensors can track if an area is being occupied or the number of occupants in an area. This information can be correlated with occupancy

in an area at a specific time, which is helpful for identifying resource utilization. Some occupancy sensors have mechanisms to identify one person from another to track repetitive occupancy/utilization.

- **Video:** Video can be used as a livestream camera or can be stored for analysis or replay at a specific point-in-time. Video sensors can be used for a variety of use cases, from physical security to occupancy sensors, for assisting with the visual inspection of manufacturing processes, and for locating a specific person for law enforcement reasons.

Note Vibration, energy consumption, and location sensors are used in some of the most common business use cases.

Vibration: Placing vibration and torque values on motors and machines allows computers to detect when a failure will occur on a device (motor, engine, truck, robot, and so on). This allows maintenance to be scheduled in advance and repairs to be made without affecting other processes. For example, if a rock truck breaks down in a mine, it blocks the tunnel, preventing other rock trucks from passing in the tunnel. The broken-down rock truck must be towed out of the tunnel to allow operations to be restored. During this time, the mine cannot haul raw product.

Energy consumption: Users and businesses are now capable of tracking the amount of energy being consumed in real time. This may result to changes in how a business uses a building or machine in an attempt to lower its cost. However, by connecting these sensors with the utility companies, businesses can now create smart grids. This allows companies to detect power outages and pinpoint failure locations quicker so that service can be restored. Another benefit of using a smart grid is that it allows consumers to use green technologies like solar power to run their homes, or return power to the grid.

Location: Location sensors allow for real-time location services (RTLS) such as the following:

- Enabling the transportation industry to track its vehicles and make that information available to its customers.
- Locating specific resources (forklifts, torque tools, laptops, and so on) quickly. These sensors can also be used for inventory management for manufacturers.
- Locating employees. Generally, this is done to locate an employee with a specific skillset, but could also be used for man-down safety reasons in chemical/refinery environments.
- Restricting or prohibiting certain things or devices from entering a specific area. When an area is defined in the RTLS system, when a tag crosses the barrier, it creates an alarm. This is known as **geo-fencing**.

Table 1-1 provides a high-level overview of typical IoT sensors used in various market verticals.

Table 1-1 *Things and Market Segment Consumption*

	Manufacturing	Mining	Oil and Gas	Utilities	Smart Buildings	Health and Medical	Retail	Hospitality	Transportation	Connected Roadways	Emergency Services
Temperature	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Humidity	✓	✓	✓		✓				✓	✓	✓
Vibrations	✓	✓	✓						✓		✓
Pressure	✓	✓	✓	✓					✓		✓
Occupancy					✓	✓	✓	✓	✓		✓
Air Quality	✓	✓	✓	✓		✓		✓	✓	✓	✓
Water Quality	✓	✓	✓	✓				✓			✓
Energy Consumption	✓			✓	✓		✓	✓	✓	✓	
Location	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓

A variety of variables and factors need to be taken into account concerning the type of smart thing that should be used. Specifically, the type of data and business use cases should always dictate which sensors are to be used. The selection of things used should also take into account the following items:

- **Power source:** Is the power source wired or battery driven? If battery driven, how long can the battery last, and how often does it need to be replaced or recharged?
- **Ability to move:** Is the thing physically attached to an immovable object (steel beam, pole, and so on) or is it mobile via a wearable, attached to an animal or a cart?
- **Frequency that data is transmitted:** How often is data expected to be transmitted from the thing? The amount of data and frequency can directly correlate to the life of the thing before a temporary power source must be recharged or replaced.
- **Connectivity method (wired or wireless):** How does the thing communicate with other devices? Does it use a wired technology (serial, Ethernet, and so on) or does it use a wireless technology? The transmission rate for that media and distance to the next upstream network device should be accounted for, too.
- **Density of things:** How many things are needed in an area to effectively capture data? Can data be transmitted through other things like a mesh network?
- **Processing capability:** Not all IoT devices have the same amount of processing power or storage. These components vary based on the function and size of the device and directly correlate to the cost.

Communications

This layer provides a mechanism to communicate with other things or external sensors. This layer contains the following constructs:

- Access network (medium)
- Gateways
- Network protocols
- Data exchange protocols

Access Network (Medium): This sublayer refers to the physical medium or wireless technology that is used to provide connectivity between things. Specifically, it is connectivity for thing-to-thing communication, or it provides a centralized aggregation on the network. Physical cables provide consistent availability and bandwidth but are not always available because of cost (initial procurement and/or installation) or not feasible for things that are mobile.

In these circumstances, wireless technologies are commonly deployed. As a general rule of thumb, bandwidth decreases as the wireless range increases. Figure 1-4 demonstrates some common wireless technologies that correlate with a technology's ability to cover

bandwidth against the range available. It is important to take into consideration power consumption and cost when selecting a technology. In general, power consumption and cost increase with high throughput and higher distances.

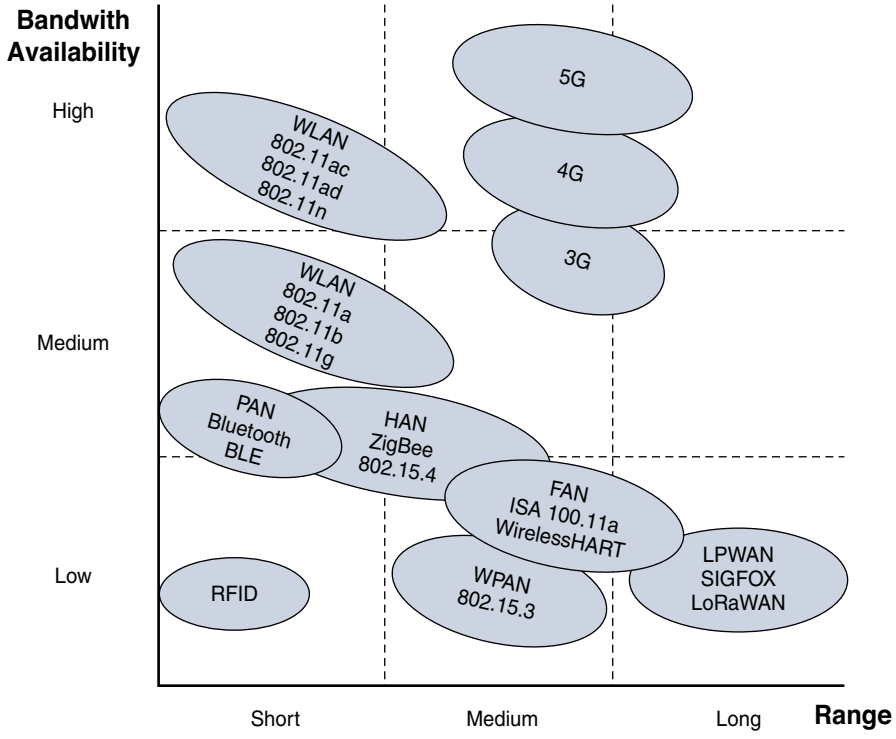


Figure 1-4 *The Effect of Wireless Range on Bandwidth*

This sublayer consists of networks that group things as follows:

- **Personal area network (PAN):** The network around a person. Bluetooth is a common technology used in PANs.
- **Home area network (HAN):** The network around a home. This includes Bluetooth Low Energy (BLE), ZigBee, and 802.11 Wireless.
- **Local area network (LAN):** Network around local devices. Most IoT environments prefer to use Category 5/6 cabling because most electricians can repair replace this cabling. Category 5/6 cables are restricted to 100m for signal integrity. Fibre optic cable can be used to exceed this restriction but is not as commonly used in a LAN environment because of the limited workforce that can service this medium.
- **Field area network (FAN):** Network around a large field area. In general, the FAN is a wide-open area that covers less than 4km.

- **Wide area network (WAN):** Network that provides connectivity between sites kilometers apart.
- **Low-power wide area network (LPWAN):** Wireless network that is designed for long-range communications using low power because devices are powered by battery. LoRaWAN and SIGFOX are common LPWAN technologies.

Gateways: As a simple definition, gateways provide a method of connecting different technologies together through translation at the highest portion of the networking stack. Gateways map semantics between two different things and perform translation between devices. It is important to note that gateways do not just provide protocol translation, but that they can merge different architectures.

From the connectivity context, the gateway connects PANs, HANs, LANs, FANs, LANs, LPWANs to the backbone network. Gateways connect to a backbone network to provide a high-speed and hierarchical architecture.

Gateways can provide a method of aggregation of things and for processing data at the edge of the network. This concept is explained further in the "Data Transportation and Computation" section.

Note There is a direct correlation between the increase in cost and operational complexity and the number of technologies and protocols that are merged into a design.

Network Protocols: The network protocol is an essential component for all network transmissions. It is responsible for locating things on the network and controlling the flow of data. The transport layer collects all the data and prepares it into a format that can be presented to an application.

The network protocol should be an open standard considering that most IoT networks come from devices and things from different manufacturers. Proprietary protocols would require translation and introduce complexity when things communicate with proprietary protocols with other manufacturers' things that do not have access to them. The Internet Protocol (IP) has been well established and has become one of the de facto network protocols for IoT.

Note Network protocols include the IPv6 protocol too. The use of IPv6 is encouraged because the address space does not have the limitation of the IPv4 protocol. In fact, multiple IoT specifications such as 6LoWPAN and RPL do not include IPv4.

Data Exchange Protocols: The flow of data between things and systems must either be initiated or requested. Some sensors will push data at regular intervals, whereas other sensors require polling of information (also known as a pull). Table 1-2 displays multiple data exchange protocols and some key factors used for selecting a data exchange protocol.

Table 1-2 *Well-Known Data Exchange Protocols Consumption*

Protocol	Transport	Format	Discovery Mechanism	Function
Constrained Application Protocol (CoAP)	UDP	Binary	Negotiation and well-known URI.	REST resource manipulation. Resource tagging with attributes. Resource discovery.
Extensible Message and Presence Protocol (XMPP)	TCP/HTTP	XML	Server with Presence announcement.	Manage presence. Establish sessions. Data transfer.
Message Queuing Telemetry Transport (MQTT)	TCP	Binary	Things connect to a message broker.	Publisher/subscriber.

IoT Applications

Now that all of these smart things can communicate with each other, the true power of IoT can be realized with the usage of applications. IoT applications are classified as analytic and control applications.

IoT Analytic Application: This type of application collects information from multiple things and organizes it in a manner such that calculations can be made easily. Analytic applications can provide historical data, trends, and suggestions based on the results calculated.

An example of an IoT analytic application could be one that collects the amount of energy consumed in a manufacturing plant on a thing-by-thing basis. The analytic application could provide reports that contain power consumption on a daily basis that could be correlated with the types of machines and product yield for that day. The plant planner could make changes to his schedule or process based on this information to improve efficiency.

IoT Control Application: This application controls smart things. A key concept is that the IoT control application contains the ability to process logic that the smart thing does not contain, or to provide a level of orchestration across multiple things.

An example of an IoT control application comes from the oil and gas segment, where oil needs to be moved from one location to another. The application would open the necessary valves in the pipeline and then start the pumps so that oil can flow from one location to another. In this example, the valves would not have any knowledge of the pumps, or vice versa, and the control application would orchestrate turning the pumps on/off or making the valves opened/closed.

It is important to note that some IoT applications can perform both control and analytic functions.

Note Smart services are IoT applications that increase overall efficiency. For example, a program on a thermostat that can detect if a door is open and overrides the thermostat from turning on the HVAC system can be considered a smart service.

The design of an IoT application must take into account the data structure, frequency, size, and volume of the data transmitted. All of this information directly correlates to the volume of the data transmitted from things to the IoT applications:

- **Structure:** Data exists in either a structured or unstructured format when it is transmitted and stored. Structured data is organized in a logical manner and provides a consistent form of data entry. Unstructured data comes from human language. The same data that is sent in structured format could be sent in unstructured format, but would require context to understand where that data maps.
- **Frequency**: The timeframe in which a thing will transmit data to another thing.
- **Size:** The size of the data correlates to whether the data is structured or unstructured. If the data is structured, then every field should be examined to identify whether or not it truly is necessary, because every field will contribute to the size of the update sent from thing to thing.
- **Volume:** The volume directly correlates with the type of data, frequency of transmission, size of the data, and the number of devices. If a thing transmits 10KB of data every 5 seconds, this might not seem like a large amount of data. However, if there are 1,000 such things, then the total amount of daily data is 172.8GB.

On the other hand, if a thing transmits 100KB of data every minute and there are 1,000 such things sending data; then the daily data amounts to only 144GB. So even if a thing sends more data, if the frequency is lower, then the overall amount of data is lower.

Data Transportation and Computation

Normal corporate networks use a hierarchical network model that divides the network into three layers: **core**, **distribution**, and **access**. Most end-user devices connect to the access layer, and the data center (DC) connects to the core devices (or a server switch block that connects to the core). Most of the data is generated, stored, and processed in the data center.

However, IoT networks generate almost all of their data at the access layer. IoT networks are generally very large, and they must scale appropriately to handle the data from all of the smart things. The data structure and number of smart things directly correlate to the volume of traffic that is analyzed.

A large IoT network with a million sensors that transmit 15KB of data every 30 seconds can result in 1.8 terabytes (1,800,000MB) of data on an hourly basis (~16 petabytes of

data annually). IoT networks of this size must take into account the amount of available bandwidth the access layer has. Depending on the physical access medium, an update could saturate the link.

Figure 1-5 demonstrates four things that transmit via radio towers that can only support 150KB of network traffic. The size of an update is 45KB, which can easily pass across a 150KB link. However, when Thing 1's packet is combined with the updates for Things 2, 3, and 4, the total data exceeds the supported speed of the radio towers. There is not enough bandwidth for all four things' updates.

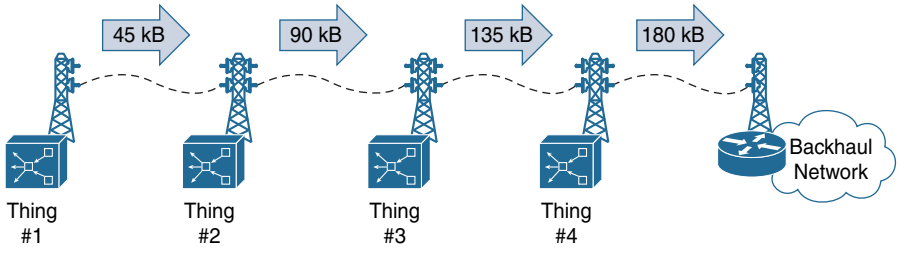


Figure 1-5 *Link Saturation*

Connected vehicles can generate large amounts of data that exceed 20GB per day. The transfer of this data to the manufacturer on a daily basis is not necessary because most of the relevant processing should be performed within the vehicle. Only a small subset of the data generated is needed by the manufacturer to understand how the vehicle can be improved based on its usage.

Data Center and Cloud

Data centers are the heart of delivering IT services by providing storage, compute resources, communications, and networking to the devices, users, and business processes and applications. Data centers are designed with proper power and cooling requirements to sustain large amounts of servers for the processing of data. Recently, data analytics have added value and growth in data centers by touching on aspects of an enterprise and its processes.

The growth of cloud-based services is due to an increased focus on business agility and cost optimization. The adoption of cloud services and applications enables faster delivery of services with improved operational efficiencies. Connecting to public-based cloud services can provide additional cost reductions with improved accessibility, with the assumption that a mechanism exists to encrypt/protect the data in transit.

Latency can become an issue with sending all the data from sensors to the cloud or DC. Depending on the amount of time for a packet to transmit from a thing to a DC, analysis of the data at the DC with the proper response sent back to the thing could exceed an

acceptable amount of time. For example, suppose a camera takes a photo of a manufacturing line as parts leave a stamping press for visual inspection as part of the quality control process. Inferior parts should be discarded and removed from the conveyor belt as they are moved to the next station. If there is too much latency, a failed part could move down the conveyor belt. One solution could be to lower the speed of the conveyor belt, but that might reduce the performance of that manufacturing facility.

Placing compute resources in a hierarchical fashion can result in reducing latency, conserving network bandwidth, and increasing local efficiency.