Every IoT project needs a reference architecture that defines what functionality is required, where that functionality will operate, and the key system interfaces. This research provides the Gartner IoT Reference Model to help I&O technical professionals design their IoT architecture.

Overview

Key Findings

- Tiers, layers and interfaces are the fundamental building blocks of an Internet of Things (IoT) architecture. Every IoT architecture blueprint can be built from these components.

- Tiers define where to deploy a component, application or process. The three logical deployment tiers are edge, platform and enterprise.

- Layers define what capabilities an IoT component, function or process must possess. The five key layers are device, communications, data, application and process.

- Interfaces define how data and control flow into, out of and through the system. Interfaces are dependent upon the hardware and software used to implement the layers and tiers.

Recommendations

Infrastructure and operations (I&O) technical professionals focused on cloud and edge computing:
Analysis

Technical professionals must define their IoT architecture blueprint before they can thoughtfully advise key stakeholders who make major investment and operational decisions. The architecture blueprint defines:

- Where that functionality will operate (for example, at the edge)
- What functionality is required (for example, a predictive maintenance algorithm)
- How data and control will flow through interfaces (for example, an API)

In the context of this research, a “blueprint” is a visual representation and documentation of the high-level architectural structure for an IoT system. The blueprint not only contains the technical components, but also addresses governance, management and operational issues. The blueprint must be designed to satisfy the specific technical requirements and business objectives for each IoT project. Conversely, not having a blueprint increases project risk because the architecture remains vague and elusive.

Technical professionals should use their blueprint as a tool for collaboration with the relevant stakeholders, such as technical specialists, business leaders, operational teams and key suppliers. The blueprint exposes important design concerns, including security risks, technology challenges and major investment decisions. The blueprint also serves as a framework that guides technical inquiry, trade-off analysis and vendor selection.
Gartner IoT Reference Model

The Gartner IoT Reference Model is a framework for creating an IoT architecture blueprint. The model consists of three components (see Figure 1):

- **Tiers** define where a component, function or process operates in the IoT architecture.
- **Layers** define what capabilities an IoT component, function or process must possess.
- **Interfaces** define how data and control flow into, out of and through the system.

**Figure 1. Gartner IoT Reference Model**

![Gartner IoT Reference Model Diagram]

This research will describe the three tiers and the five layers, and then provide an example that illustrates all the concepts.
Tiers

A typical IoT solution contains three tiers (see Figure 2). The tiers define where a component, function or process operates in the architecture based on the business objectives and constraints of the specific project. Tiers are logical deployment locations. Note that this figure provides only one of many possible design patterns.

Figure 2. Three Tiers of an IoT Architecture

Three Tiers of An IoT Architecture

The edge tier is where data is sampled and collected from the environment by instrumented “things” or devices. These “things” may include consumer devices (such as thermostats), appliances or industrial systems (such as a robotic arm) that contain sensors to collect data, or configurable parts (such as actuators) to alter the operation of the device. The edge tier may also contain optional IoT gateways that can provide stream processing, data transformation and local storage, as well as integrate other systems to IoT platforms (e.g., data historian).

The platform tier is where the IoT solution performs systemwide actions in collaboration with the edge tier. The platform may perform device management, stream processing, advanced analytics and workload orchestration. It may also invoke enterprise applications. The platform tier is usually implemented on top of a composable set of IoT services developed by one or more commercial providers (e.g., Microsoft Azure IoT, PTC ThingWorx, Hitachi Lumada, Amazon Web Services [AWS] IoT).
Note: Many of the commercial IoT providers refer to their solutions as “IoT platforms.” When we refer to these commercial solutions in this research, we will use the term “commercial IoT services” to distinguish them from the “platform tier.”

The enterprise tier is where IoT integrates with the set of applications, processes and services required to accomplish a business objective. For example, applications such as inventory management, enterprise resource planning and customer relationship management often reside in the enterprise tier. These applications may benefit from the data-driven insights provided by an IoT system. Many IoT platforms include APIs that enterprise applications use to integrate with the IoT platform.

Layers

The Gartner Reference Model decomposes an IoT design into five layers:

1. Device
2. Communications
3. Data
4. Application
5. Process

Each layer defines a logical set of IoT capabilities. Any of the capabilities within a layer may be deployed at one or more tiers. The job of the IoT architect is to determine which capabilities are needed and where (i.e., which tier) they must be deployed. Each of the following sections defines the capabilities for a single layer.

Device Layer

The device layer is where the digital world meets the “real world” (see Figure 3).
The "real world" generates real-time, analog signals, which are sampled (by sensors) and converted to digital information (by devices and other “things”). Such environments may be natural, where seismic or weather-related information will be generated, or they may be man-made environments, such as buildings or factory floors with motion detectors or video cameras.

The device layer includes:

- **Sensors:** The collection of IoT data begins with sensors, which are typically part of an embedded (or real-time) system that samples an analog signal (for example, voltage). The sensor sends the analog signal to an analog-to-digital converter (ADC) where this digitized signal is then sent via a serial bus (for example, Inter-Integrated Circuit [I2C]) to a microcontroller or microprocessor.

- **Things/devices:** Sensors are typically embedded within some type of device or appliance (such as a wind turbine) or other “thing” (such as a robotic arm). This device may contain an embedded OS and perform some amount of basic analytics. Because the sensors may generate extremely high volumes of data, basic analytics can be applied to the raw data to reduce it to a more limited and valuable dataset.

- **Aggregation/gateways:** The aggregation point can act as a gateway by accepting multiple protocols and data models from disparate devices and then translating that data into the protocol, API and data model supported by the platform tier (see the Tiers section). It can also ingest data, perform sophisticated analysis and take control actions. Lastly, the aggregation point can serve as a method to collect data from operational technology that may not have the intelligence or capabilities to integrate and send data to the IoT platform (e.g., data historian).
Layer interfaces: Device layer interfaces typically focus on the physical environment and the devices at the IoT edge. For instance, the physical-environment-to-sensor interface specifies what is being measured (e.g., pressure), measurement frequency (e.g., milliseconds) and measurement precision (e.g., plus or minus 0.01 pounds per square inch). The sensor-to-controller interface specifies the physical connection (e.g., I2C) and message structure. If the IoT design requires an IoT gateway, then the physical interfaces between the gateway and IoT endpoint must be specified.

Security: Device layer security typically focuses on device cataloging, patch management and embedded trust. Device cataloging is a multistep process that includes device detection and device identification. Patch management is a program to assess software and firmware vulnerabilities and to remediate those vulnerabilities by patching as necessary. Embedded trust provides a hardware root of trust (e.g., Trusted Platform Module), which is often a requirement to secure functions at the IoT endpoint.

Communications Layer

The communications layer defines the event broker, network technologies and communications service providers (CSPs) necessary for the IoT system (see Figure 4).

Figure 4. The Communications Layer

IoT solutions include two distinct communications services: local and long range:

Local communications is within the IoT edge tier. The local communications connects IoT endpoints with one or more edge gateway functions and, in some cases, may provide local endpoint-to-endpoint communication. For example, in the case where a manufacturing facility is the IoT edge, an organization may use industrial Ethernet as the local communications networking technology.
Long-range communications is from the edge tier to remote destinations. The long-range communications connects the edge tier to the central IoT platform and, in some cases, may provide edge-to-edge communication. For example, in the case where a remote water well is the IoT edge, an organization may use LoRa as the long-range communications networking technology. In some less common designs, the commercial IoT services may be resident at the edge. In this case, long-range communications may not be necessary.

The communications layer includes:

- **Event broker**: Events carry data that are the “lifeblood” of IoT. Event brokers are essential middleware for a distributed, reliable, manageable IoT implementation because they receive published events and deliver them to interested subscribers. Every IoT platform will provide some form of event broker. The event broker provides minimal coupling between services and allows producing services and consuming services to scale as required. It also enables you to introduce new components that consume or produce existing events without impacting existing services. Common event broker capabilities include publisher/subscriber connection management, reliable event delivery and support for event grouping/delivery (often called a “topic” but also known as subjects, streams or channels). Broker-managed communications can be one-to-many, many-to-one, and many-to-many. The market for event brokers is diverse and complex. It is constantly changing due to combinations of capabilities and features. Refer to Choosing Event Brokers: The Foundation of Your Event-Driven Architecture for more information.

- **Network technology**: Network technologies provide transport services for IoT data, control and management messages. The network technology can provide a personal-area network (e.g., Bluetooth low energy [BLE]), local-area network (e.g., Wi-Fi) or wide-area network (e.g., low-power WAN [LPWAN]). The networks can provide an intratier service (e.g., retail outlet Wi-Fi network within the edge tier) or an intertier service (e.g., internet connection between a manufacturing plant edge and a public cloud platform). For additional information see How to Architect Your Network to Optimize Internet Performance and Reliability.
Communications service providers: Communication service providers and mobile virtual network operators (MVNOs) provide public and private network connectivity service offerings. Additionally, CSPs provide higher-value IoT solutions that include IoT platforms, IoT applications (such as fleet management, smart grid) gateways and multiaccess edge compute (MEC) endpoints, and IoT device management. For example, a CSP can wirelessly connect many IoT things to the platform tier over a cellular network and may also provide SIM card provisioning and monitoring services (e.g., for a cold-chain-monitoring solution on a refrigerated truck). For more information see Architecting a Reference Framework for 5G Private Mobile Networks.

Layer interfaces: The interfaces at this layer focus on event-driven messaging and networking technologies. The messaging protocols (e.g., MQTT) will dictate how client applications connect to the event broker to publish events or receive events. The networking technologies define the local communications technology (e.g., Ethernet), the long-range communications technology (e.g., LoRa) and the physical connections to those technologies (e.g., RJ45).

Security: The security at this layer includes authentication, authorization and encryption mechanisms to protect the movement of information between systems. For example, streaming architectures such as Apache Kafka support secure connections between brokers, producers, consumers and Apache ZooKeeper. The security at this layer may require the support of SSL/TLS and Simple Authentication and Security Layer (SASL) protocols. Additionally, if the design makes use of an IoT platform, then the designer must specify how to integrate the edge tier with the IoT platform identity and access management (IAM) services.

Data Layer

The data layer ensures end-to-end semantic consistency of data throughout the distributed IoT system. It describes how data flows into, out of and through the system, as well as how it is transformed and stored (see Figure 5).
The data layer includes:

- **Data pipeline**: IoT data pipelines are essential components of an IoT system. They move data from multiple sources across the IoT system and make the data available for downstream consumption, either in raw format or as curated data. Data engineers are constantly creating and managing data pipelines to orchestrate the movement, transformation, integration, validation and loading of data throughout organizations. This task is even further exacerbated by IoT solutions because data movement is performed at a greater frequency and velocity and may also be highly distributed. The design and optimization of IoT data pipelines are a collaboration between the IoT architect, infrastructure and operations (I&O) engineer, and data engineer. Refer to *Data Engineering Essentials, Patterns and Best Practices* for further information.
Data models: The data model describes the way you organize and structure your IoT data and logic. It provides a layer of abstraction and consistent representation for all IoT endpoints. This layer of abstraction is used by your algorithms and applications in the application layer. The model may represent sensor data and/or derived, complex event data that the IoT solution must aggregate and process. The model defines the metadata, which is critical for ensuring proper understanding and use of sensor data. For example, if temperature is measured at the edge tier in degrees Celsius, then it must be interpreted at the platform tier using the same units. Technical professionals must determine how they will ensure systemwide metadata consistency across layers and tiers. Lastly, the data model must also define the controls that the edge gateway or platform can impose on the IoT endpoints. For example, if the system measures temperature in a robotic arm servomechanism, then you may want to define a control that takes an action if the temperature exceeds a threshold. You must ensure that you have version control over the models as well as the ability to support more than one model in production at the same time. Refer to Data Modeling to Support End-to-End Data Architectures for additional information.

Data storage: The data layer defines how and where data is stored in the system. The IoT system may process a substantial historical dataset from multiple sources for the purpose of predictive analytics or event classification. This means that an IoT system may not only perform stream processing to determine if any immediate action is required, but also store the data in the platform’s logical data warehouse for historical analytics.

Layer interfaces: The interfaces at this layer focus on the data pipeline, the data model and storage. For instance, the data pipeline interface may include APIs at an edge gateway that perform data transformation. A data model interface may include a description of an IoT device for use by a digital twin application (described in the Application Layer section). Data storage interfaces for an edge gateway may describe the requirement that it supports very fast writes because it must handle high-volume data from IoT endpoints.
Application Layer

The application layer contains the features and intelligence of the IoT system (see Figure 6). It provides the stream processing, data analytics, digital twin and business application integration necessary to satisfy the IoT business requirements.

Figure 6. The Application Layer

One of the key design decisions at the application layer is how best to distribute analytics between the edge and platform tiers. The application layer includes:

- **Security**: Data layer security focuses on topics such as data polymorphism, data poisoning, data theft, data linkage and other inference attacks. Data polymorphism is multiple copies of (raw) data with similar or identical semantics being stored. Security issues arise when not all copies are equally protected. Poisoning is a form of unauthorized access to training data. Data theft frequently involves attacks on the infrastructure, such as exploiting software vulnerabilities and stealing disks. Data linkage and other inference attacks misuse data from different sources to reverse the applied controls with the goal of inferring sensitive data that the attacker otherwise cannot access.
Stream processing: Stream processing enables application components to communicate asynchronously. A stream is an ordered unbounded set of events communicated from a producer to a consumer. Stream processing performs operations not only on discrete events within a stream but also on sets of events within a defined window. A window can be time-bound (e.g., from 01:00 to 01:59) or sliding (e.g., last 30 minutes). For example, an IoT predictive maintenance application may use stream processing to calculate the rolling average of a robotic arm temperature sensor over a 30-minute rolling window. If the average temperature exceeds a preset threshold, it can send an event to an ERP system to communicate that a production line machine needs to be taken offline for predictive maintenance. Refer to *Essential Patterns for Event-Driven and Streaming Architectures* for an in-depth analysis of streaming architectures.

Data analytics: Analytics are an essential feature of IoT systems. They drive business value and operational efficiencies by offering new ways to leverage large amounts of IoT data. Rather than treating analytics as a single component within an overall system, organizations should regard analytics as a feature that can be exploited and deployed throughout the end-to-end IoT architecture. Edge analytics are analytical environments embedded in edge devices and edge gateways. They enable local, real-time analysis of streaming data coming from industrial equipment, vehicles, appliances, systems, manufacturing control systems, devices, and other sensors and actuators. IoT platform analytics are centralized analytical capabilities that use lightweight communication protocols to ingest data from edge devices. Functions performed in the IoT platform are typically associated with data ingestion, advanced analytics and event processing.
Digital twin: A digital twin is enterprise software that implements a virtual representation of an IoT-connected entity used to provide situation awareness that helps improve business decision making and outcomes. Digital twins can be implemented for things such as equipment, processes and organizations, as well as for people. At a minimum, digital twins always monitor things, but digital twin data is often analyzed to predict the future state, and at times is also used to simulate the behavior of physical things, processes or people. A digital twin is a digital representation of real-world people, processes and things. Use cases across many industries are rapidly emerging (see Tool: 50-Plus Digital Twin and IoT Cost Optimization Examples). These include modeling and monitoring of distribution networks in energy and utilities, automation and control of machinery in manufacturing settings (pharmaceuticals and industrial businesses), infrastructure maintenance in the public sector (smart cities and public transport), and connected products and customer intelligence in retail. There is a clear role for data and analytics (D&A) teams in helping their organizations derive value from digital twins, because digital twins are, in essence, D&A capabilities. They are based on analytics, data management and D&A governance capabilities. As such, D&A leaders responsible for building and refining the organization’s D&A strategy need to be aware of digital twin requirements and plan to make use of their capabilities to support digital twin implementation and operation. Refer to What Data and Analytics Leaders Need to Know and Do About Digital Twins for additional information.

Business applications: The IoT system will need to integrate with key business applications (e.g., ERP) and process management systems (e.g., workflow management software). For example, a predictive maintenance IoT use case may need to integrate with an enterprise asset management system. An IoT architecture must define how the IoT solution will interact with existing business systems.

Layer interfaces: The interfaces at this layer define how to integrate IoT software with stream processing, data analytics, digital twins and business applications. If the designer is building proprietary applications, then they will need to define the APIs to those applications. For example, imagine a manufacturing company that builds a digital twin of industrial assets and operations in order to reduce equipment downtime and maintenance costs. The developers will need to integrate with the digital twin interfaces and historical data stores (e.g., data historian) to diagnose conditions and drive maintenance recommendations. Refer to How to Design Great APIs and How to Deliver Sustainable APIs.
Process Layer

The process layer focuses on governance, operations and management processes (see Figure 7).

**Figure 7. The Process Layer**

More so than any other layer, the process layer is about the “why” of your IoT initiative. Technical professionals must have a clear understanding of the business rationale that is driving their IoT initiative to make meaningful decisions about changes to existing, value-delivering processes.

The process layer includes:

- **Security**: Application layer security focuses on authentication, authorization and encryption mechanisms to protect the use of application layer services. This includes protecting gateway software, IoT platform applications and APIs from exploits and abuse. If designers are using an IoT platform and/or a commercial-off-the-shelf IoT gateway, then they must identify the platform and/or gateway protection mechanisms they plan to use. If the designer is building proprietary services, then they will need to define the security mechanisms to protect those services.
**Governance:** Governance encompasses the operational management of IoT devices and the information and services they deliver. Governance ranges from simple technical tasks, such as auditing devices and applying firmware updates, to more complex issues at the intersection of business and technology. One example of the latter is the responsibility to control access to and usage of the data an IoT system generates. Governance is both a technology and a business challenge. Governance issues will become increasingly important as the scale and complexity of IoT grow, and as the number of stakeholders increases. For example, device management, such as applying firmware updates, is a simple form of governance. But even firmware management becomes much more challenging when the management is split between multiple parties, such as the device owner, the device manufacturer and, perhaps, a third party that is operating the device. How and when updates are applied become increasingly important. Further complications include budget ownership where one party owns the budget but another party governs the operations on behalf of the budget owner. Governance is an essential underlying component of cybersecurity because discovering and updating “things” in response to threats are fundamental requirements. Governance issues also emerge in infrastructure management and security. Technical professionals are increasingly being asked to connect IoT equipment to corporate networks and to open firewalls to allow the equipment to communicate outside the organization. Data privacy is also a governance issue. For instance, what is the process to protect the IoT data? Who has access to it? Where is it stored?

**Operations:** Many I&O leaders have yet to consider the impact of IoT on their monitoring, IT service management (ITSM) or automation methodologies. Having a comprehensive strategy for these areas is critically important because IT operations is where IoT initiatives become “operationalized” — or integrated into an organization's standard procedures and day-to-day operations. Just as IoT merges the physical world with the digital world, IT operations will merge the IoT world with the line of business that will rely on the data and operationalization of the “things.” IoT projects that integrate with operational technology (OT), such as industrial control systems, will usually be paid for, and run by, the OT team. The IT team will need to demonstrate to the OT team it can add value to the IoT project before the OT team will allow its participation. This often requires the IT team to become familiar with OT processes, people and technology. In some cases, the IT and OT groups will form a “fusion team” that includes multidisciplinary team members that work together to solve a business problem.
Management: Organizations must define how IoT will integrate with their management systems and processes. Organizations should evaluate central IoT platforms for endpoint management. Hyperscale cloud provider IoT platforms — such as AWS IoT, IBM Watson Internet of Things and Microsoft Azure IoT Suite — currently provide endpoint management services. In addition, vendors such as Software AG, PTC and Hitachi provide industrial IoT platform services. These systems enable you to provision, configure and monitor devices. IoT management will include activities such as:

- IoT device provisioning (including identity assignment, enable/disable device, SIM card management and group device provisioning)
- User provision (e.g., administrators and end users)
- Configuration and change management
- Release management (e.g., firmware updates)
- Incident management
- Password management
- System logging
- Capacity planning

Layer interfaces: The interfaces at this layer define how to integrate with the various systems that provide operations and management services for the IoT system and any human/machine interaction (e.g., IoT dashboard). For example, imagine a manufacturing organization that wants to increase productivity and avoid expensive unplanned equipment downtime events. The IoT developers may need to interface IoT application software with the ITSM system.

Security: Process layer security focuses on security governance, operations and management. Security governance exists to ensure that the strategic requirements of the business are defined and that the security program adequately meets those requirements. This may include discussing and adjudicating between business needs in complex situations. Security operations execute security-related processes relating to current infrastructure on a day-to-day basis. Security management builds and runs the security program to meet these strategic business requirements. This incorporates the various security functions, processes and tactics that make up the security program.
Security Model

IoT security is one of the most important challenges for technical professionals because it can affect the design of every tier, layer and interface in the IoT system. IoT architects must establish their security model early in the process of developing an IoT design. Once established, the security model will facilitate security by design at the very early stages of architecture creation. The creation of a security model is a complex topic and is beyond the scope of this research. We recommend using the following resources to guide the creation of your security model:

- **NIST 8259A (IoT Device Cybersecurity Capability Core Baseline):** This National Institute of Standards and Technology (NIST) report defines a baseline set of IoT device cybersecurity capabilities. Organizations should use this as a vulnerability checklist prior to purchasing or integrating IoT devices.

- **NIST SP 800-82:** This NIST report (SP 800-82 Rev. 2) provides recommendations on how to secure industrial control systems, such as supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), and programmable logic controllers (PLCs).

- **UL 2900:** The UL 2900 series of standards provides cybersecurity product requirements. These include UL 2900-1 (software cybersecurity), UL 2900-2-1 (network-connected healthcare systems), UL 2900-2-2 (industrial control systems), UL 2900-2-3 (security and life-saving signaling system components).

- **Industrial Internet of Things Volume G4: Security Framework:** This report from the Industrial Internet Consortium provides a comprehensive review of industrial IoT (IIoT) security. Part I examines what the key system characteristics are, how they should be assured together to create a trustworthy system, and what makes IIoT systems different from traditional IT systems. Part II reviews security assessment for organizations, architectures and technologies. Part III covers the functional and implementation viewpoints of the IIoT reference architecture.

- **IoT Security Foundation:** The IoT Security Foundation website contains IoT security publications, events, certifications and other resources.

- **Architecting Identity for the Edge of IoT Innovations:** This Gartner research provides an analysis of how authentication and authorization in edge devices are managed and implemented in IoT solutions. It also lays out a framework to assess new IoT solutions.
Predictive Maintenance Example

Consider a fictitious multinational automobile manufacturer that undertook a multiyear manufacturing modernization initiative (see Note 1). One of the organization's business objectives was to improve asset performance management (APM) by using machine learning to determine the probability and timing of asset degradation and failure. The organization wanted to use the sensor data and machine learning to reduce mean time between failures by predicting component failure.

The technical team used the Gartner IoT Reference Architecture to compose its architectural blueprint, layer by layer. The result of the team's analysis was the predictive maintenance architecture blueprint shown in Figure 8. Each layer reflects the decisions of the team. Note how layers span one or more deployment tiers. The interfaces expose the key integration points throughout the architecture.
The team also used this blueprint as the framework for deeper technical analysis and for discussions with business counterparts. It highlighted major areas of financial decisions by highlighting the IoT cloud platform on the blueprint. The commercial IoT services operating in the cloud is a major financial decision because there will be recurring charges that grow as the system scales. Business analysts must therefore analyze how this architectural decision impacts the IoT cost model.

The security organization focused on how to control access to the predictive maintenance algorithm and the data it acts upon. The algorithm is proprietary intellectual property, and the data is confidential information. It decided to encrypt the transport and storage of its IoT data. It also decided to limit access to the algorithm APIs, code and data by using the IAM controls in the commercial IoT service. Lastly, it asked the corporate legal counsel to assess the legal and regulatory impact in the event there was a security breach.
The interface between the IoT system and the technician dispatch system (TDS) is a technical risk primarily because the TDS is an old, somewhat fragile system. Further, the team will need to recruit a data scientist and possibly an algorithm developer to build, test and deploy the predictive maintenance machine learning model and software.

The organization deployed more than 1,000 sensors in six plants across three countries to collect training and inference data. It used the sensor data to create a machine learning model. Ultimately the IoT system helped it to reduce mean time between failures by predicting component failure.

Strengths

This reference model has the following strengths.

**Expedites Creation of an IoT Design**

The Gartner IoT Reference Model expedites the creation of an IoT design. The layers, tiers and interfaces decompose architectural complexity. The blueprint provides a way to visualize an IoT system. The reference model focuses attention on what is most important in an IoT system.

**Provides a Business-, Technology- and Vendor-Agnostic Model**

The Gartner IoT Reference Model makes no assumptions regarding business goals, technology deployment or vendor selection. Technical professionals can use the model to create an architecture blueprint irrespective of which business goals they are trying to satisfy, technology they choose to deploy or vendors they decide to select.

**Facilitates Technical and Business Collaboration**

The Gartner IoT Reference Model provides a visual tool that facilitates technical and business collaboration. Architects can distribute the blueprint as a “master plan” that all IoT teams can use as their target architecture. Technical professionals can also use the blueprints to communicate and collaborate with other stakeholders.

Weaknesses

This reference model has the following weaknesses.

**Does Not Provide IoT Cost Analysis**
This research does not provide IoT cost analysis. Before a business executive will invest money and human capital to deploy IoT, that executive will want to know the ROI. IoT cost analysis helps to determine the ROI and the sensitivity to design alternatives. For example, IoT cloud platform costs grow as the volume of data ingestion and storage requirements increase. A method to offset some of those costs is to deploy an IoT gateway at the edge to perform data aggregation and reduction before the edge forwards data to the platform.

**Does Not Provide Service Quality Analysis**

This reference model does not provide service quality analysis. Considerations such as system latency, availability, disaster recovery, error rate sensitivity and computational scalability will affect the end-to-end experience. The resolution of these technical issues will directly impact the success or failure of an IoT project and must be addressed by the technical team.

**Does Not Provide a Governance Process**

This reference model does not provide a detailed analysis into the organizational structures, technical skills and delivery procedures required for an IoT project. Many organizations have multiple IoT projects underway and therefore must establish a way for project leaders to plan, organize and collaborate.

**Guidance**

IoT solutions unify a diverse set of devices, data streams and processes across operational technology and information technology. Every IoT project needs an architecture blueprint that defines what functionality is required, where that functionality will operate, and what are the key system interfaces. Technical professionals should create their architecture blueprint using the guidance provided below:

- **Analyze each of the tiers and layers.** The three tiers and five layers of the reference model can help you compose your IoT architecture. Review each of the Tiers, Layers and Security Model sections of this research. Pay particular attention to how the capabilities of each tier and layer may help to solve the technical challenges that underlie the business problem you are trying to solve.

- **Design functionality within each layer.** Begin with the device layer. Identify the necessary functionality in each layer, and then move to the next layer until you complete a first pass for every layer. Do not let yourself get stuck on a layer as this process is inherently iterative. Treat the blueprint as a living model and explicitly practice version control with it.
Evaluate where you need to deploy the functionality. Design how each of the functional components fits within one of the three tiers. When doing so, analyze the layer- and tier-specific interfaces and security issues. It is a common practice to have similar functionality in more than one tier (e.g., stream processing at the edge and the platform tiers).

Validate and refine the design. Iteratively review the design with technical experts and other key stakeholders. Use a blueprint as a visual representation of the high-level system architecture.

Note 1. A Fictitious Multiyear Manufacturing Modernization Initiative
This example is based upon discussions with hundreds of organizations. It does not represent a specific company.

Document Revision History
Designing an IoT Reference Architecture - 11 October 2019
Architect IoT Using the Gartner Reference Model - 26 April 2018
Architect Your Internet of Things System by Using the Gartner IoT Reference Model - 20 March 2017

Recommended by the Author
Some documents may not be available as part of your current Gartner subscription.
Choosing Event Brokers: The Foundation of Your Event-Driven Architecture
Architecting a Reference Framework for 5G Private Mobile Networks
How to Architect Your Network to Optimize Internet Performance and Reliability
Data Engineering Essentials, Patterns and Best Practices
Data Modeling to Support End-to-End Data Architectures
Essential Patterns for Event-Driven and Streaming Architectures
Tool: 50-Plus Digital Twin and IoT Cost Optimization Examples
What Data and Analytics Leaders Need to Know and Do About Digital Twins
How to Design Great APIs