2
Industrial IoT Dataflow and Security Architecture

"Ensuring that the devices and systems connected to the internet are secure is a key to ensuring the safety and reliability of industrial operations."
– Dr. Richard Soley, Executive Director, Industrial Internet Consortium (IIC)

The sheer scale and complexity of IIoT demands a systematic approach to secure the system architecture. When the degree of complexity is high, decomposing the security paradigm into subdomains helps to manage and mitigate risks. This decomposition is particularly useful to use cases involving several technologies, and spanning across multiple organizational boundaries (a common scenario in IIoT).

Industrial systems last for decades. This further necessitates to plan for protecting industrial IoT systems and assets against both current and future threats.

This chapter presents in-depth insights into IIoT (big) data flows and IIoT reference architectures, and introduces the industrial internet security framework developed by the IIC. These discussions subsequently lead the reader to a simplified four-tier IIoT security model, which decomposes the essential IIoT security measures into four main layers.

But, before getting into those details, we shall present a primer on IIoT attacks, countermeasures, and threat modeling.

The main topics covered in this chapter are as follows:

- A primer on IIoT attacks, countermeasures, and threat models
- Trustworthiness of an IIoT system
- Industrial big data pipeline and architectures
- Building blocks of the industrial IoT security architecture

Primer on IIoT attacks and countermeasures

Understanding the dynamics involved in industrial IoT attacks is crucial to perform security risk analysis and mitigation. Threat modeling is commonly used as a security countermeasure, and has been discussed later in this chapter. Attack and fault trees are two methodologies useful to develop security threat models and to communicate the risk of an attack.

In the real world, most attacks are highly customized to target specific vulnerabilities in IoT products and connectivity. Many attacks target zero-day vulnerabilities. In the case of zero-day vulnerabilities, an exploit already exists and can be easily proliferated through the internet or corporate networks to create a snowball effect. Since IIoT involves significant investment and skills, most attacks involve nation state threat actors, who are motivated to create a major impact.

Some common types of attacks in the IIoT context are as follows:

- Malware-triggered ransomware
- Wired and wireless scanning and mapping attacks
- Network protocol attacks
- Infecting ICS and SCADA intelligence
- Cryptographic algorithm and key management attacks
- Spoofing and masquerading (authentication attacks)
- Unauthorized endpoint control to trigger unintended control flows
- Data corruption attacks
- Operating system and application integrity attacks
- Denial of service and service jamming
- Physical security attacks (for example, tampering or interface exposure)
- Access control attacks (privilege escalation)

More attack types can be added to this list. Today, ransomware attacks are rising steeply. In IIoT, if malware encrypts the data of any control system, it can directly trigger a physical catastrophe. For example, encrypting medical data in a hospital (refer to the WannaCry case study in `Chapter 1`, *An Unprecedented Opportunity at Stake*) could potentially lead to lethal consequences at scale. So, possible attacks in every deployment need to be carefully studied in order to better manage security risks.

Figure 2.1 shows the correlation of vulnerabilities, attacks, and countermeasures:

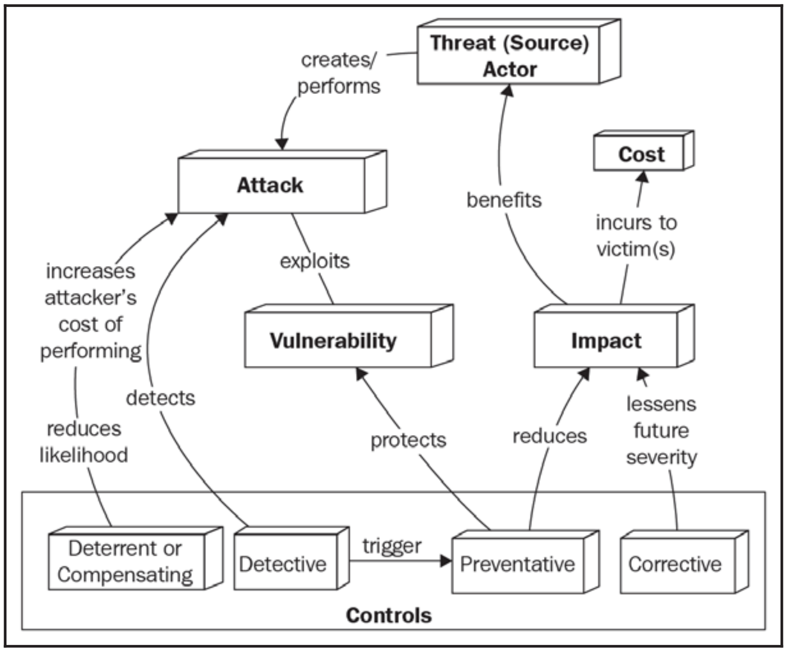


Figure 2.1: Dynamics of attacks and countermeasures; Source: Practical IoT Security, Packt Publishing

Attack surfaces and attack vectors

Industrial security risk was discussed in `Chapter 1`, *An Unprecedented Opportunity at Stake*. To assess the risk of an attack to a system, two commonly used terms are **attack surface** and **attack vector**. Both of these terms are closely tied to the industry the system was designed for, the specific deployment use case, and the associated business objectives.

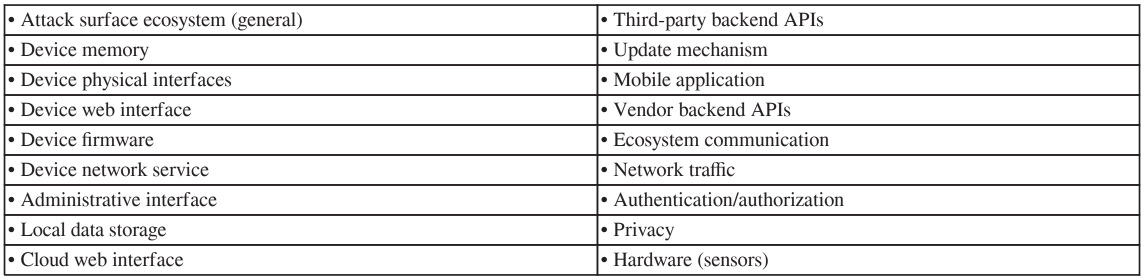
The **attack surface** spans across the system components that can potentially contribute to an attack. For example, in a traditional ICS system connected only to the SCADA network, the attack surface includes exposure to the insider threats, physical threats, vulnerabilities in proprietary SCADA protocols, and so on. However, when an ICS system is connected to a cloud platform, vulnerabilities in the cloud technologies, for example, IP-based WAN connectivity, remote configuration, and device management, and so on. get added to the equation. To sum up, IIoT significantly expands the attack surface of industrial systems and infrastructure.

An **attack vector** includes the tools and technologies that can contribute to an attack. This too is closely tied to the industry and the technologies involved. A threat actor can utilize a variety of mechanisms to launch an attack to compromise a system. So, attack vectors for an IIoT system could be physical, or network-, software-, or supply chain-related. Examples of common cyberattack vectors are phishing campaigns, insecure wireless networks, removable media, mobile devices, malicious web components, viruses, and malware.

Given the cyber-physical nature of the risks involved in IIoT, security practitioners must factor in the physical consequences of threats, attack surfaces, and attack vectors while assessing the overall risk associated with any IIoT deployment.

OWASP IoT attack surfaces

As part of OWASP's IoT Project, a non-exhaustive list of attack surfaces has been identified for IoT systems (OWASP-IoT). The list is included here to provide a basic idea of attack surfaces for IoT systems, and it is applicable to IIoT as well and can be used in attack surface-based analysis. You also can visit the OWASP website, provided in the reference section, for further elaboration:



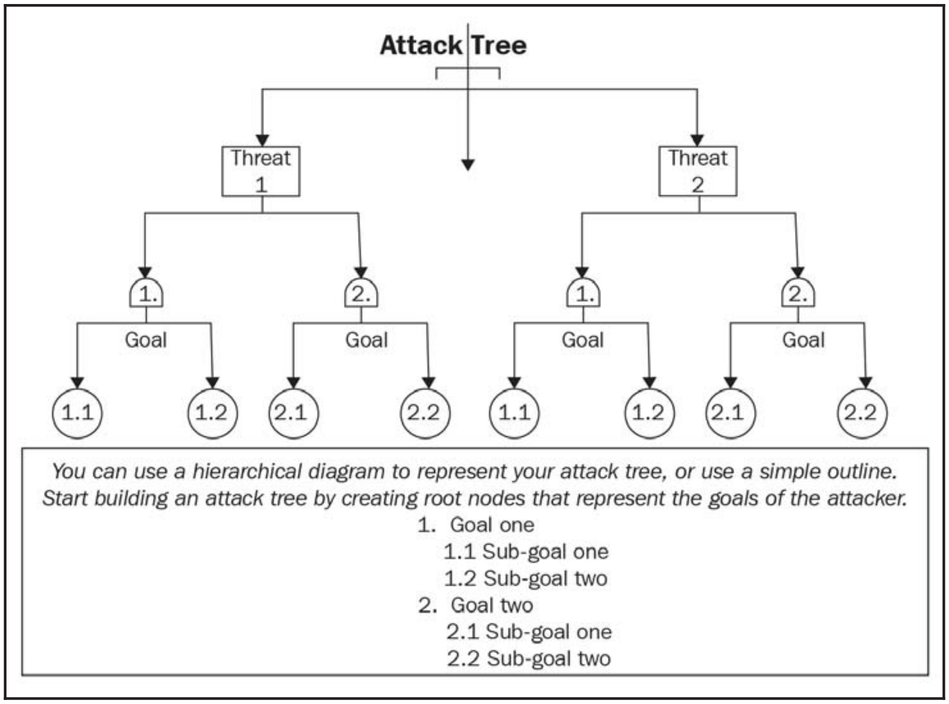
• Attack surface ecosystem (general)	• Third-party backend APIs
• Device memory	• Update mechanism
• Device physical interfaces	• Mobile application
• Device web interface	• Vendor backend APIs
• Device firmware	• Ecosystem communication
• Device network service	• Network traffic
• Administrative interface	• Authentication/authorization
• Local data storage	• Privacy
• Cloud web interface	• Hardware (sensors)

Attack trees

Attack trees provide a structured and hierarchical way to collect and document the potential attacks on a given organization, in order to perform threat analysis. Fundamentally, an attack tree allows us to derive the possible ways in which an asset or target could be attacked.

Attack trees have been used in a variety of industries, especially to analyze threats against tamper-resistant electronic systems, and in digital control systems in power grids. This concept can also be extended and utilized for connected industries.

As shown in *Figure 2.2*, attack trees are multi-level diagrams consisting of one root, and multiple leaf and child nodes. From the bottom up, child nodes are conditions that must be satisfied to make the direct parent node true. Following each path from the bottom up, when the root is satisfied, the attack is complete. Each node may be satisfied only by its direct child nodes:



You can use a hierarchical diagram to represent your attack tree, or use a simple outline. Start building an attack tree by creating root nodes that represent the goals of the attacker.

1. Goal one
   1.1 Sub-goal one
   1.2 Sub-goal two
2. Goal two
   2.1 Sub-goal one
   2.2 Sub-goal two

Figure 2.2: Illustration of an attack tree

Attack trees exploit the power of deduction to cover the entire spectrum of attacks and threats that exist in the wild. The deductions can be integrated with other threat models to create a transparent and direct mode of analysis of attacks and attackers.

In traditional cyber incidents, the goals could be identity theft, data exfiltration, denial of service, and so on. However, for use cases involving cyber-physical systems, the goals could involve physical catastrophe *"ranging from turning off a light bulb to turning off a human heart"* (IOT-SEC). Similarly, new threats and attack flavors for the root nodes also need to be accounted for, due to possible interactions with the physical world.

Fault tree analysis

In the case of IIoT, where attacks are cyber- physical in nature and closely correlates with safety and reliability engineering, fault tree analysis can be used as an effective tool.

IIoT systems and technologies involve a degree of complexity. As a result, a failure at the system level can be the result of faults occurring in any of the subsystems. The likelihood of failure, however, can often be reduced through improved system design. In **fault tree analysis** (**FTA**), logic diagrams are created for the overall system to map the relationship between faults, subsystems, and redundant safety design elements. *Figure 2.3* shows an example of a fault tree diagram:

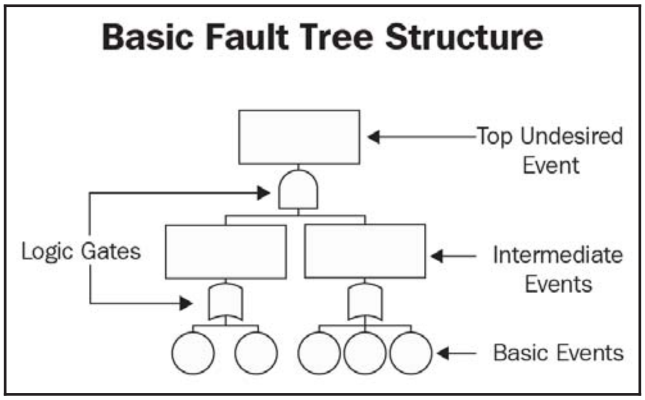


Figure 2.3: Logical structure of a fault tree

Unlike attack trees, FTA is top-down. Here, we analyze by combining a series of lower-level events (involving subsystem failures). Using Boolean logic, these events are combined to analyze an undesirable state of a system. This is also a deductive failure analysis method commonly used in safety and reliability engineering to understand how systems can fail, and hence to find ways to reduce risks of failure.

FTA was first used in the aerospace industry, where safety assurance is mandated at very high levels. For commercial aircraft, the probability of failure is 10^{-9} (one in a billion) (IOT-SEC). Nowadays, in addition to aerospace, FTA is used in many other industries such as nuclear power, chemical engineering, pharmaceuticals, energy grids, and so on. FTA is also used in software engineering, for debugging purposes, and is closely related to the cause elimination technique, used to detect bugs.

Several industry and government standards describe the FTA methodology, such as:

- NUREG–0492 for the nuclear power and aerospace industry
- SAE ARP4761 for civil aerospace
- MIL–HDBK–338 for military systems
- IEC 61025 for cross-industry usage

Threat modeling

It is not possible to eliminate threats. Threats exist regardless of the security measures employed to mitigate the risks of an attack. In real-world deployments, security measures are all about managing risks while acknowledging the existence of threats. However, unless we know the threats for a specific use case, we cannot mitigate them (OWA-TRM).

Threat modeling is a systematic technique to effectively manage and communicate risks. In threat modeling, based on a solid understanding of the architecture and implementation of a system, we identify and rate the threats according to their probability of occurrence. This allows us to mitigate risks in a prioritized order, which can be both cost-effective and efficient (MST-TRM).

Microsoft developed a threat modeling approach for applications, which can also be applied to IIoT systems. So, we shall treat IIoT threat modeling in this section according to Microsoft's approach, which involves the steps shown in *Figure 2.4*:

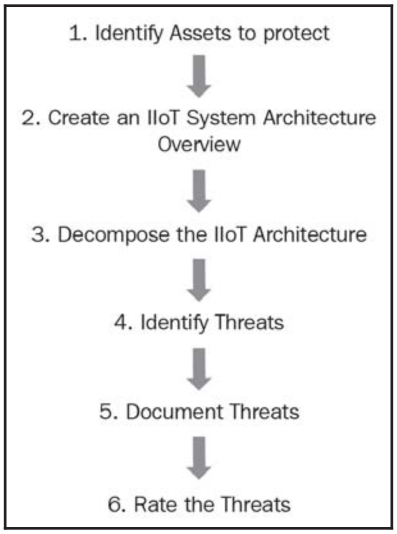

1. Identify Assets to protect
2. Create an IIoT System Architecture Overview
3. Decompose the IIoT Architecture
4. Identify Threats
5. Document Threats
6. Rate the Threats

Figure 2.4: Microsoft threat modeling process applied to IIoT architecture

The steps are explained as follows:

1. **Identify assets:** Identify a list of assets that must be protected.
2. **Create an architecture overview**: Document the overall IIoT system architecture, which includes subsystems, platforms, applications, trust boundaries, control and data flows, and so on.
3. **Decompose the architecture:** Decompose this architecture into system (application, IoT endpoints) and infrastructure (communication protocols, data centers, network protocols) components. Use this to create a security profile for this specific IIoT use case with the goal to uncover vulnerabilities in the design, implementation, or deployment configuration.
4. **Identify the threats**: Based on the attack surfaces and vectors, and by using attack trees and FTA (discussed earlier in the chapter), identify the threats. Two commonly used threat identification techniques are STRIDE and DREAD (discussed in upcoming sections). Both of these techniques were developed by Microsoft, and can be used at this stage.

5. **Document the threats**: Document each threat, using a common threat template that defines a core set of attributes to capture for each threat.
6. **Rate the threats**: Rate each threat and prioritize the threats based on their impact. The rating process weighs on the probability of the threat against the damage that could result from an attack. This allows us to effectively direct investments and resources.
   Rating and ranking of threats can be done using several factors. *Figure 2.5* shows a risk-centric approach that can be applied at a high level for IIoT deployment use cases:

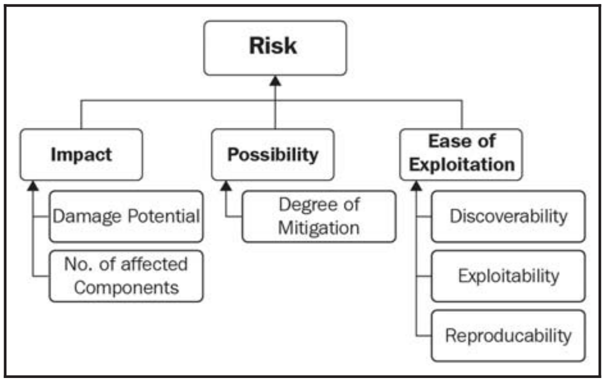


Figure 2.5: Risk-based threat ranking

STRIDE threat model

STRIDE, developed by Microsoft, is a model to identify and classify threats. The STRIDE model has also been extended to include IoT threats (MST-STR), and can be applied to IIoT use cases. The STRIDE acronym represents the following types of threat:

- **Spoofing identity**: A type of threat where a person or device uses another person's credentials, for example, login and password, certificate, and so on, to gain access to an otherwise inaccessible system. A device can use a spoofed device ID.
- **Tampering with data**: Altering the data to mount an attack. The data could be related to a device, protocol fields, unencrypted data in motion, and so on.
- **Repudiation**: When a person or a device is able to refuse to be involved in a particular transaction or event; and when it is not possible to prove otherwise. In the case of a security breach, the inability to trace it to the responsible person or device is in itself a threat.

- **Information disclosure**: Exposure of information to individuals who are not authorized to have access to it. In the IIoT context, this could mean when sensor or operational data is accessible to an adversary planning to launch an attack.
- **Denial of service**: These threats prevent legitimate users or devices from accessing server (compute) or network resources. Exploits that slow down system performance to unacceptable levels can also be considered as a form of denial of service attack.
- **Elevation of privilege**: An unprivileged user penetrates the security defenses to gain a sufficient level of trust and access privileges to compromise or damage the targeted system.

DREAD threat model

After the threats have been identified and classified, it is also important to rank and prioritize them. Higher priority threats must be addressed. The DREAD method is designed to rank the threats (MS-DREAD). Although originally developed for subsystem components (software, firmware, and so on), the DREAD concept can be utilized in threat assessment at various levels of granularity of an IIoT system.

DREAD is an acronym that represents five criteria for threat assessment:

- **Damage**: Assessing the damage that could result if the threat advances to a security attack. In the case of cyber-physical systems, the damage could be data exfiltration, environmental damage, human injury, and so on.
- **Reproducibility**: A measure of how frequently the specific threat would mature into a successful attack. An easily reproducible threat has a higher chance of being exploited.
- **Exploitability**: An assessment of the effort, monetary investment, and expertise required to launch the exploit. Threats requiring low levels of skill and experience are more exploitable than those that require highly skilled personnel and great expense to carry out. In the case of IIoT, the exploits usually involve a high degree of complexity and expertise. If an industrial threat can be remotely exploited, then it is more exploitable than an exploit requiring on-site, physical access and special credentials.
- **Affected users**: The number of users that could be affected by an attack is a measure to prioritize threats. This criteria can also be extended to include the number of devices and assets impacted by the attack.
- **Discoverability**: The likelihood a vulnerability can be taken advantage of.

In the DREAD classification scheme, threats are quantified, compared, and prioritized based on their risk value. The risk value is computed using the following formula:

Threat risk using DREAD = (Damage potential + Reproducibility + Exploitability + Affected Users + Discoverability) / 5

Trustworthiness of an IIoT system

As already noted in this book, the concept of securing cyber-physical systems is a superset of what we normally understand by cybersecurity and information security.

To properly represent the scope of IIoT security, the term **trustworthiness** is used (NIST-CPS) (IIC-IISF). A working definition of trustworthiness for CPS, according to NIST-CPS, is:

> *"Trustworthiness is the demonstrable likelihood that the system performs according to designed behavior under any set of conditions as evidenced by characteristics including, but not limited to, safety, security, privacy, reliability and resilience."*

Trustworthiness of an IIoT system is an important stakeholder expectation. To make an IIoT system trustworthy, security characteristics of both IT and OT domains must be combined (IIC-IISF). As shown in *Figure 2.6*, the key characteristics of a trustworthy IIoT system combine the elements of IT trustworthiness (privacy, security, reliability, and resilience) and OT trustworthiness (safety, reliability, security, and resilience). All references to IIoT security in this book are founded on this concept of IIoT trustworthiness:

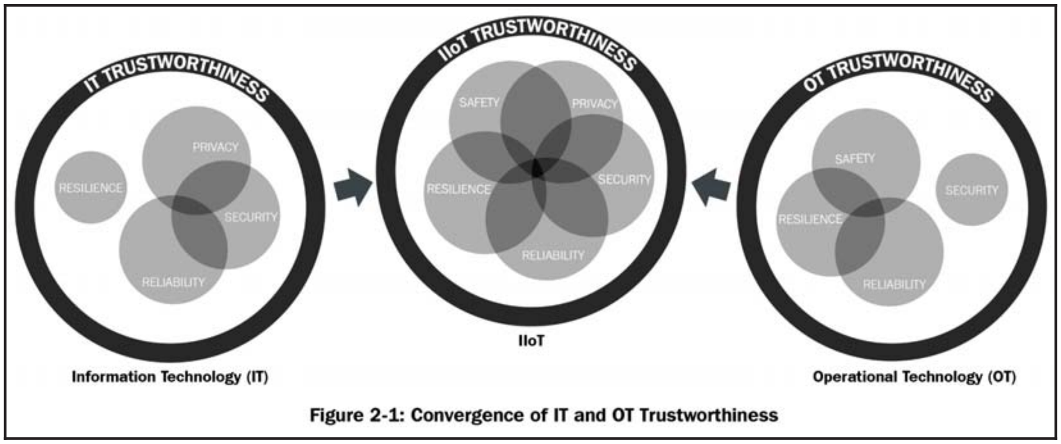


Figure 2.6: IIoT trustworthiness converges IT and OT trustworthiness; Source: IIC-IISF

In an organization, risks are perceived quite differently by the enterprise IT and OT teams. A balanced consideration between OT and IT is needed to ensure the trustworthiness of IIoT systems. The control and data flows, in the case of IIoT, may span across multiple intermediaries. Trust should also permeate across the system life cycle, involving various actors and functional entities, starting from hardware and software component builders, system and platform builders, and the supply chain, all the way to the operational users. `Chapter 7`, *Secure Processes and Governance*, further elaborates on this critical concept.

In the subsequent sections of this chapter, we shall analyze the industrial big data flows, discuss the various IIoT architectural patterns, and subsequently develop a simplified 4-tier security model as a practical foundation for IIoT trustworthiness.

Industrial big data pipeline and architectures

Data is the prime asset in the IIoT value chain. Industrial devices such as sensors, actuators, and controllers generate state and operational data. The information inherent in this industrial big data enables a variety of descriptive, prescriptive, and predictive applications and business insights. This end-to-end flow of data, from the point of ingestion, through information processing using various **extract, transform and load (ETL)** functions, applying AI and machine learning intelligence, up to the point of data visualization and business application, is collectively referred to as the industrial big data pipeline (shown in *Figure 2.7*):

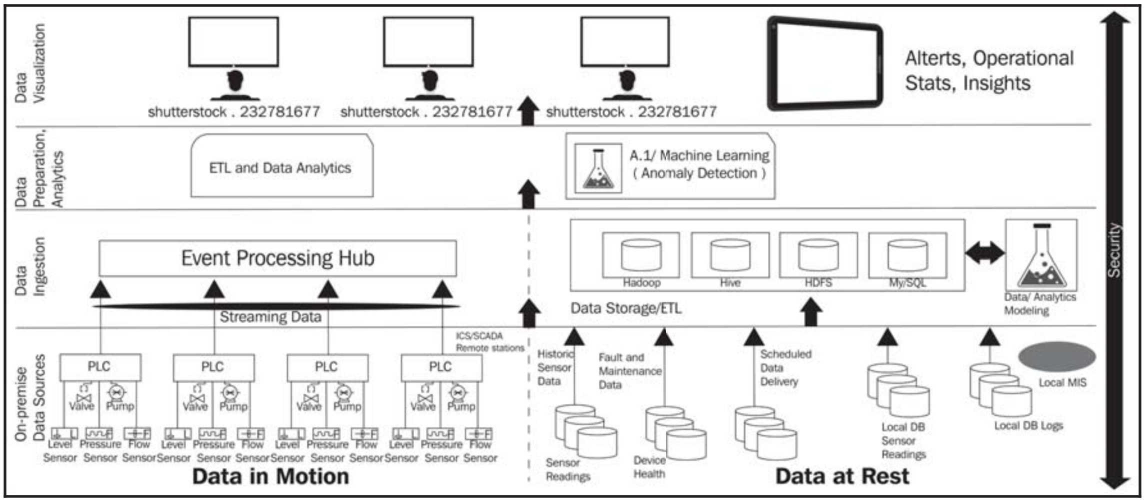


Figure 2.7: Schematic illustration of the stages in Industrial Data flows

The preceding diagram is explained as follows:

- **On-premise data sources**: On-premise data includes usage and activity data – both real-time streaming data (data in motion) and historical/batch data from various data sources. Sensors and controllers embedded in remote sites or plant floors generate big data. This data reflects sensed parameters, controller action, and feedback signal data; from which we can gain granular visibility into real systems. This raw data can be both structured and unstructured, and can be stored in data lakes for future processing or streamed for (near) real-time stream analytics. Data at rest is stored in transient or persistent data stores and includes historical sensor data, fault and maintenance data reflecting device health, and event logs. This data is sent upstream to canonical data stores in platforms, either on-premise or in the cloud, for batch processing.
- **Data ingestion**: Event processing hubs are designed to ingest high data rates and send the data for real-time analytics. In the case of batch data, canonical data stores and computing clusters such as Hadoop/HDFS, Hive, SQL, and so on perform ETL functions and may direct the data to machine learning applications.
- **Data preparation and analytics**: In this stage, feature engineering and ETL can be performed on the data to prepare it for analytics.
- **Stream analytics**: It provides real-time insights based on the sensor data, for example, the device health of a steam turbine. The data can be stored here in long-term storage for more complex, compute-intensive batch analytics. The data can be transformed for consumption by machine learning applications that can predict, for example, the remaining useful life of the steam turbine.
- **Data visualization**: Enterprise-tier applications such as **customer relationship management** (**CRM**), **enterprise resource planning** (**ERP**), and so on consume the data. **Business intelligence** (**BI**) analytics software such as Tableau, Pentaho, and so on can be used to develop data visualization applications to gain a variety of BI insights (for example, performance, remaining useful life, and so on) or create alerts and notifications based on anomalies.

The exact implementation of the big data pipeline and data flows can vary based on specific data governance and data ownership models. The end-to-end pipeline can be fully owned by the industrial organization (for example, a smart windmill) or can leverage private or public cloud infrastructures to leverage application and business domain efficiencies.

In the cases where the assets are dispersed and remote, for example, turbine engines in a wind farm and oil rigs in an oil field, data processing and computational capability may be needed at or near the assets for local analytics and control. This process is further elaborated in the subsequent sections of this chapter.

From an IIoT system trustworthiness perspective, each element of the big data pipeline needs to be designed by integrating data privacy, reliability, and confidentiality controls; and at the same time keeping in view safety, availability, and resilience implications.

Practical mechanisms to integrate security controls such as secure transport, storage and updates, security monitoring, and so on across this industrial data pipeline and data flows are discussed in the subsequent chapters.

Industrial IoT security architecture

In 2015, the IIC released the **Industrial Internet Reference Architecture** (**IIRA**) for IIoT systems (IIC-IIRA). It uses "ISO/IEC/IEEE 42010:2011 Systems and Software Engineering–Architecture Description" for architectural conventions and common practices. IIRA provides an architectural framework to analyze concerns, views, models, and so on with certain degrees of abstraction. The use of reference architectures helps to incorporate security by design. Architects can build use case-specific IIoT architectures on top of these reference architectures.

In this section, the four viewpoints of IIC's reference architecture are briefly discussed. These viewpoints simplify the understanding and decomposition of IIoT architectures. You can find an in-depth treatment of these viewpoints in (IIC-IIRA).

Business viewpoint

The business viewpoint of an IIoT architecture helps to analyze and evaluate business-oriented concerns, such as the business objectives of adopting an IIoT solution and its value, return on investment, lifecycle maintenance costs, and so on. It further identifies how the IIoT system achieves the stated objectives through its mapping to fundamental system capabilities. According to IIC:PUB:G1:V1.80:20170131:

> *"To verify that the resultant system indeed provides the desired capabilities meeting the objectives, they should be characterized by detailed quantifiable attributes such as the degree of safety, security and resilience, benchmarks to measure the success of the system, and the criteria by which the claimed system characteristics can be supported by appropriate evidence."*

Usage viewpoint

The activities and workflows involved in the usage of an IIoT system to achieve the key system and business objectives are analyzed in this viewpoint. An example workflow would be:

1. Register new device to the edge gateway
2. Register the new device in the cloud-based management platform by automatic discovery and querying of all gateways
3. Run remote test procedure appropriate for this device type and verify that values generated are within expected range and consistent with similar devices in the proximity

This analysis maps the usage elements to their functional and implementation counterparts in the overall architecture. Safety is an important trustworthiness factor of IIoT system usage—in addition to data integrity, data confidentiality, and resilience—that needs to be factored in across the usage cycle.

Functional viewpoint

The functional viewpoint provides a basic abstraction to design the important functional components of an IIoT end-to-end architecture. IIoT involves multiple mission-critical functional components with complex structures, mutual interactions, interfaces, and connectivity. These need to be properly designed to ensure safety and resilience.

IIRA decomposes this functional viewpoint into five function domains to better tackle analysis, design, and security integration. These functional domains are applicable across industry verticals. While there can be other ways to decompose function-specific use cases, the following five domains provide a starting point to conceptualize a functional architecture:

- **Control domain**: This focuses on the sensing and actuator functions. Interaction with external physical objects and the environment is the main aspect of this domain, which also deals with environmental safety, resilience, and data protection. Common examples are control units in a wind turbine or autonomous vehicle, or an ICS in an energy grid.

- **Operations domain:** In an industrial internet architecture, traditional industrial controls which are typically focused on one local physical plant, evolves to a higher level. The operations domain includes functions around provisioning, management, monitoring, and optimization across multiple plants, asset types, fleets, or customers. As an example, instead of optimizing one train, IIoT operation domain factors in data combined from multiple fleets owned by different railroads. This can optimize the rail network utilization across an entire country.
- **Information domain**: Represents a collection of functions to gather data from various domains, most significantly from the control domain. This data is then transformed, persisted, and modeled to acquire high-level intelligence about the overall system; which in turn helps us obtain data-driven insights and dynamic optimization. For example, using cost, demand, and logistics, the output of an automated production plant can be dynamically altered. Since these functions mostly belong to the IT domain, proper cybersecurity controls must be integrated in the planning and in design.
- **Application domain**: This includes functions to implement business functionalities, such as application logic and rules, APIs, dashboards, and so on.
- **Business domain**: Functions integrate the IIoT systems with traditional or new business applications such as ERP, CRM, **Product Lifecycle Management** (**PLM**), **Manufacturing Execution System** (**MES**), **Human Resource Management** (**HRM**), asset management, service lifecycle management, billing and payment, work planning and scheduling systems, and so on.

These functional domains cross-cuts multiple system trustworthiness characteristics, as shown in *Figure 2.8*. Depending on the specific use case requirements, these functional domains can be concentrated or dispersed, both logically and physically. For example, the information domain can be provisioned either at the edge of the industrial premises (for faster processing and decisioning), or in remote data centers or with cloud service providers:

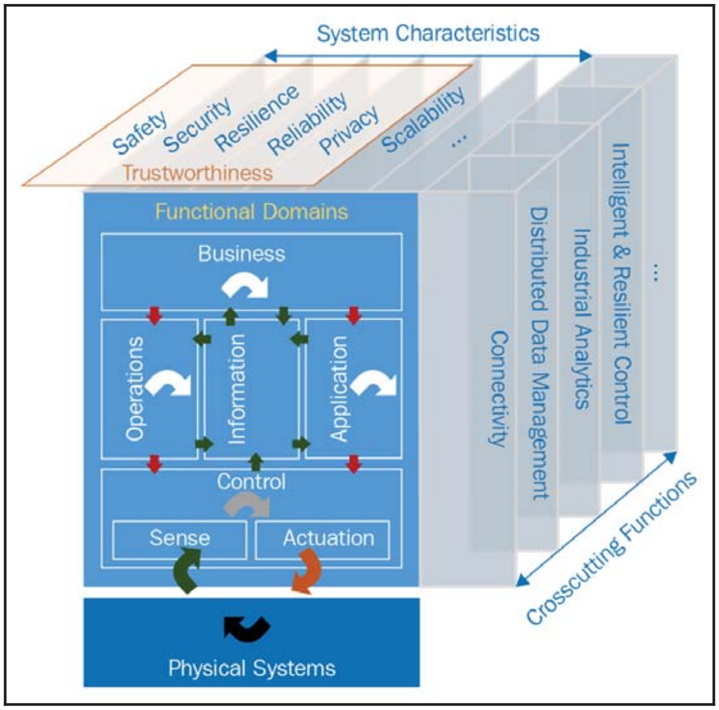


Figure 2.8: Functional domains and cross-cutting IIoT trustworthiness; Source: IIC-IIRA

Implementation viewpoint

The implementation viewpoint is the culmination of the other three viewpoints. This viewpoint needs to factor in the business objectives, such as cost and time-to-market constraints, activities related to product usage, protocols, network topologies, and so on, necessary to meet the functional characteristics.

This is also the viewpoint where security strategies such as security by design, defense in depth, and threat-based risk analysis need to be implemented.

In the next section, we shall review a few common IIoT architectures to establish a baseline understanding before we dissect the security analysis into subcomponents.

IIoT architecture patterns

IIoT deployment includes the various functional domains (control, operations, information, application, and business) discussed in the previous section. Implementation of these domains can result in a variety of architectural patterns (IIC-IIRA). By abstracting the specifics of various IIoT deployments, a few generalized patterns can be derived. We shall discuss two common patterns to help derive a security architectural model.

Pattern 1 – Three-tier architectural model

Three-tier architectures are quite common and involve connectivity, data, and control flows across the following tiers:

1. Edge tier
2. Platform tier
3. Enterprise tier

Figure 2.9 shows a three-tier IIoT architecture:

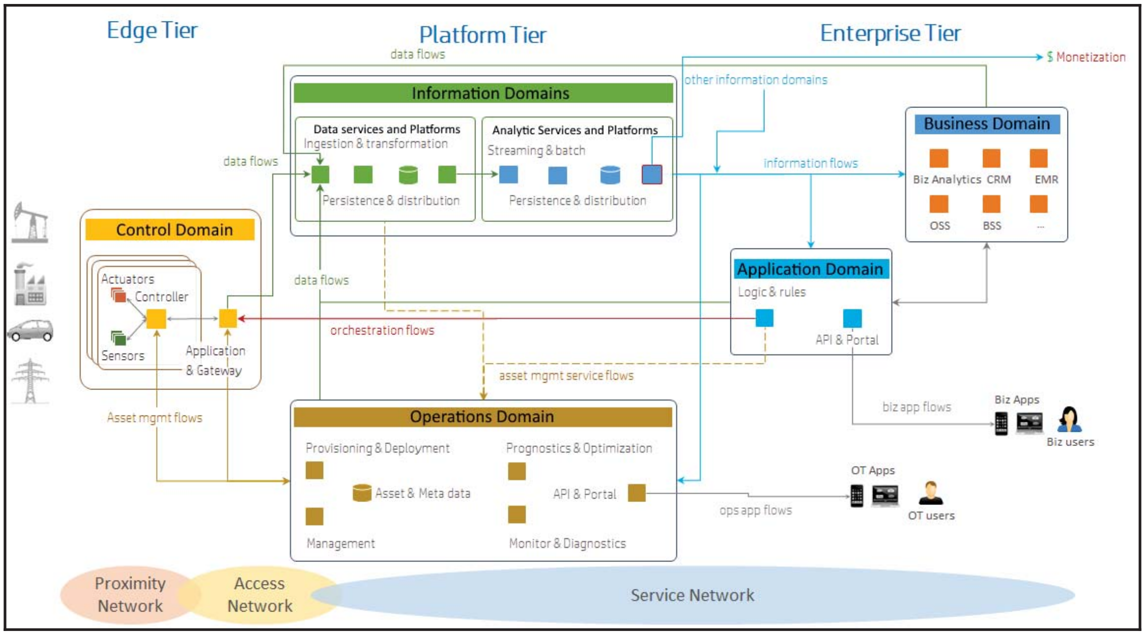


Figure 2.9: Functional domain representation in a three-tier IoT architectural pattern; Source: IIC-IIRA

The three-tier pattern combines the major components of IIoT, such as sensing and control, data processing and transformation, intelligence, communications and connectivity, and also management services and business applications. It also maps to the functional viewpoint. For example, in *Figure 2.8*, the control domain functionality is mapped in the edge tier, information and operations in the platform tier, and application and business in the enterprise tier.
This mapping can vary, depending on the implementation. For example, in some use cases, to enable intelligent edge computing, some functions related to information processing and certain application logic and rules could be implemented in or close to the edge tier.

Connectivity in the edge tier is provided by a proximity network that connects field devices, sensors, actuators, and control systems, also known as edge nodes. Connectivity can be wired or wireless. A proximity network may utilize mesh or LAN network topologies, creating one or multiple clusters, which are then connected to the edge gateway that bridges to WAN or corporate networks. Data is collected from the edge nodes at the edge tier, which can be processed locally or sent via the gateway to cloud-based platforms.

The access network connects the edge and platform tiers. The platform tier consolidates and analyzes data flows originating in the edge tier. The platform tier also forwards management and control management commands from the enterprise to the edge tier. The access network can be a corporate network or a WAN **virtual private network** (**VPN**) over the public internet, or a 3G/4G/5G cellular network.

The enterprise tier is an abstraction of management functionalities. It receives data flows that originate in the edge tier and are processed in the platform tier. This data can be used for visualization or analytics for business decisioning. Operational users in the enterprise tier can also generate control, configuration, and device management commands, which are transported downstream to the edge nodes. The platform and enterprise tiers are connected over the service network. The service network may use a VPN either over the public internet or a private network equipped with enterprise-grade security.

Pattern 2 – Layered databus architecture

A databus is a logical abstraction of connectivity that implements a common set of schemas and a common data model. In a layered databus model, each endpoint in a given layer communicates using that common set of schemas.

The layered databus architecture provides low-latency (real-time), secure, peer-to-peer data communications both within and across the logical layers of an IIoT deployment. This pattern is useful in industrial use cases where control and monitoring are distributed at various operational layers. For example, in a SCADA system in an oil rig, smart machines and controllers deployed in the remote field locations need to directly communicate control and monitoring data, which can also enable faster local analytics.

Supervisory controls, monitoring, and analytics are contained in the supervisory layer.

A separate databus can connect a series of systems for coordinated control, monitoring, and analysis at the next higher level.

In a layered architecture, the databus at various layers may have a different set of schemas or data model. To allow communication across different layers using different data models, a lower-level databus exports only a controlled set of internal data.

To match data models across different layers, databus gateways or adapters may also be used. The adapters may also separate and bridge security domains, or act as interface points for integrating legacy systems or different protocols (IIC-IISF).

The transitions between the layers may filter and reduce data. Since the scope of control and analysis increases at every layer from the bottom up, it is important to reduce the amount of data transmitted across layers to match the increase in scope, latencies, and also level of abstraction.

The data-centric publish-subscribe communication model is very common to data buses, where applications in a given layer simply "subscribe" to data they need as inputs and "publish" information they produce. This publish-subscribe communication model is effective for quickly distributing large quantities of time-critical information, especially when the delivery mechanisms are not very reliable.

Object Management Group's (**OMG**) **data distribution service** (**DDS**) standard utilizes this layered databus model. **Message Queuing Telemetry Transport** (**MQTT**) uses a broker-based publish-subscribe model. DDS and MQTT, and their security capabilities, are discussed in `Chapter 5`, *Securing Connectivity and Communications*.

In *Figure 2.10*, a representation of large SCADA systems used for oil monitoring and operation control is shown as an example implementation of the layered databus architecture:

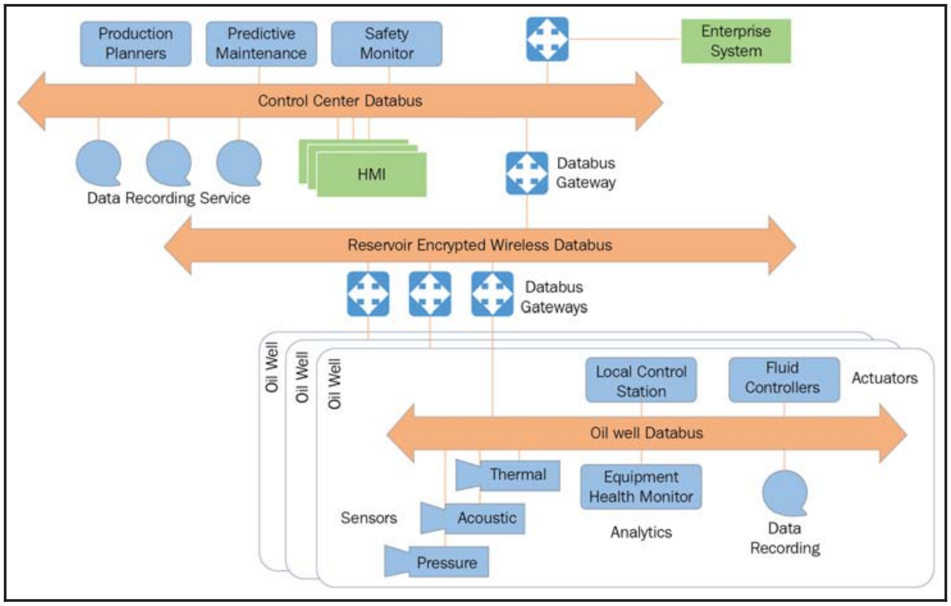


Figure 2.10: Example of layered databus architecture pattern; Source: IIC-IIRA

Building blocks of industrial IoT security architecture

For the three-tier architecture discussed in the previous section, the IIoT security architecture has to span end-to-end across the three tiers – from device endpoints at the edge, through the platform tier, and ultimately to the enterprise tier. In the case of layered databus deployments, the security framework needs to encompass the databus communication and schemas, the endpoints at each layer, and also the interlayer communication through the databus gateways. This proves the pervasive nature of IIoT security. Besides, security can't be bolted on as an afterthought, rather security risks should be evaluated early in the deployment lifecycle; and countermeasures must be built into the design. These security requirements are however, not always easy to implement in real-world industrial IoT deployments, due to some distinguishing characteristics of IIoT, as excerpted below from **IIC's Industrial Internet Security Framework** (**IIC-IISF**) document:

- Since IIoT involves both IT and OT, ideally security and real-time situational awareness should span IT and OT subsystems seamlessly without interfering with any operational business processes.

- Average lifespan of an industrial system is currently 19 years. Greenfield deployments using the most current and secure technologies are not always feasible. Security technology must often be wrapped around an existing set of legacy systems that are difficult to change. In both greenfield and brownfield deployments, all affected parties—manufacturers, systems integrators and equipment owner/operators—must be engaged to create a more secure and reliable IIoT system.
- As there is no single "best way" to implement security and achieve adequately secure behavior, technological building blocks should support a defense-in-depth strategy that maps logical defensive levels to security tools and techniques. Due to the highly segregated nature of industrial systems, security implementation needs to be applied in multiple contexts. Multiple sub-networks and differing functional zones may have different operating technologies and security requirements. Security tools and techniques built for IT environments may not always be well suited for OT environments.
- IIoT systems may have constrained system resources that need to meet various requirements, such as system safety and real-time execution. These factors may not allow implementing all security measures and controls to their fullest extent (as required by the defense-in-depth strategy). The security program implementation considerations should take into account all the required functional and non-functional aspects of the system behavior, including their relative priorities.

Based on the preceding distinguishing characteristics, *Figure 2.11* shows the functional building blocks for a multilayered IIoT security framework from edge to cloud proposed by (IIC-IISF). It maps to the functional viewpoint of IIC's reference architecture:

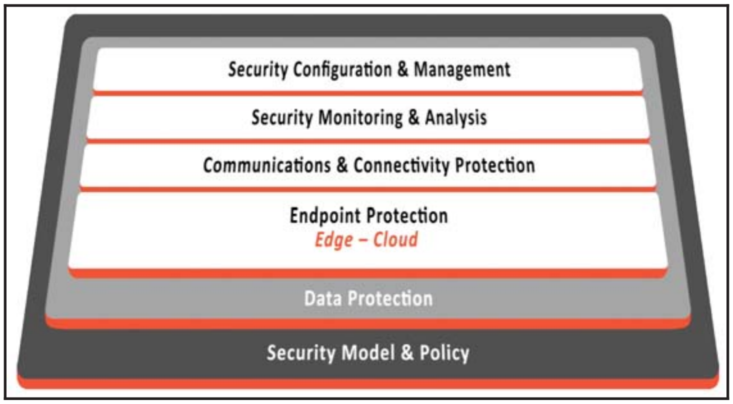

Security Configuration & Management
Security Monitoring & Analysis
Communications & Connectivity Protection
Endpoint Protection
Edge – Cloud
Data Protection
Security Model & Policy

Figure 2.11: Security framework functional building blocks; Source: IIC-IISF

The functional viewpoint of the security framework is composed of six interacting building blocks. These building blocks are organized into three layers. The top layer consists of the four core security functions: endpoint protection, communications and connectivity protection, security monitoring and analysis, and security configuration management.

These four functions are supported by a data protection layer and a system-wide security model and policy layer.

A brief description of each of these layers has been excerpted from (IIC-IISF):

- **Endpoint protection**: This implements defensive capabilities on devices at the edge and in the cloud. Primary concerns include physical security functions, cyber security techniques, and an authoritative identity. Endpoint protection alone is insufficient, as the endpoints must communicate with each other, and communications may be a source of vulnerability.
- **Communications and connectivity protection**: This uses the authoritative identity capability from endpoint protection to implement authentication and authorization of the traffic.
  Cryptographic techniques for integrity and confidentiality, as well as information flow control techniques, protect communications and connectivity.
  Once endpoints are protected and communications secured, the system state must be preserved throughout the operational lifecycle by security monitoring and analysis, and controlled security configuration management for all components of the system.
  These first four building blocks are supported by a common data protection function that extends from data at rest in the endpoints to data in motion in the communications. It also encompasses all the data gathered as part of the monitoring and analysis function and all the system configuration and management data.
- **Security model and policy**: The functional layer governs how security is implemented and the policies that ensure the confidentiality, integrity, and availability of the system throughout its lifecycle. It orchestrates how all the functional elements work together to deliver cohesive end-to-end security.

A four-tier IIoT security model

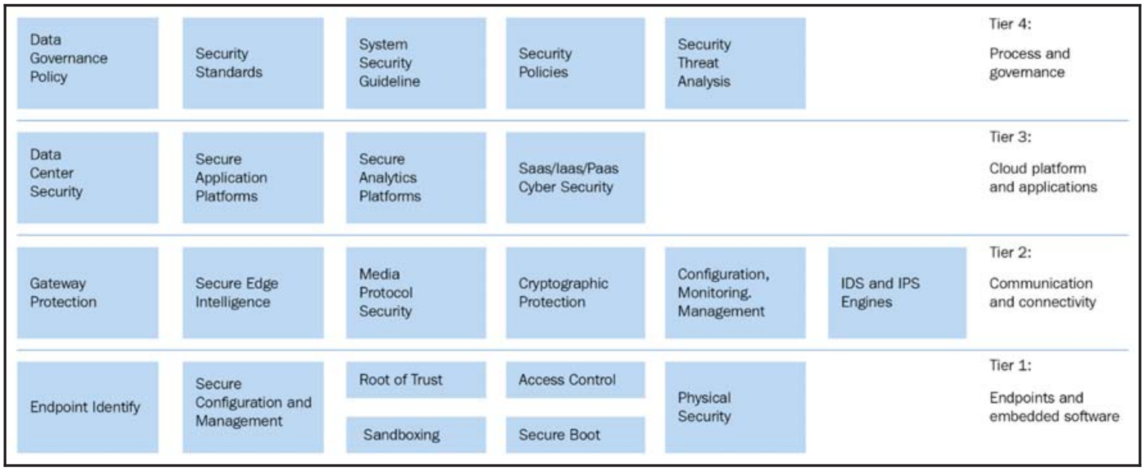
An industrial IoT system is highly complex and involves several moving parts. To simplify the security analysis and implementation, there are multiple ways we can decompose IIoT architecture into constituent components. Since most common deployment models consist of the edge, platform, and enterprise tiers, and security research and development are more aligned with the technology stacks, in this book, to facilitate security analysis, planning, and implementation, we shall dissect the overall architecture in a four-tier security model, with the following tiers:

1. Endpoints and embedded software
2. Communication and connectivity
3. Cloud platform and applications
4. Process and governance

This layering follows the unique security considerations of IIoT as discussed earlier, namely:

- Security integration needs to factor in IT and OT domain specific dynamics
- Security needs to address the industrial lifecycle (which may run into decades) and brownfield deployments (coexistence with older technologies)
- Resource constraints of industrial endpoints and their high availability requirements

This four-tier security model takes into account data protection layer functionality in the IISF (*Figure 2.11*), which encompasses data at rest, in use, and in motion. The functionalities in the top layer of the security framework map to tiers 1-3 of this four-tier security model. The security and policy layer of the security framework maps to the process and governance tier of this model:



Data Governance Policy	Security Standards	System Security Guideline	Security Policies	Security Threat Analysis		Tier 4: Process and governance
Data Center Security	Secure Application Platforms	Secure Analytics Platforms	Saas/Iaas/Paas Cyber Security			Tier 3: Cloud platform and applications
Gateway Protection	Secure Edge Intelligence	Media Protocol Security	Cryptographic Protection	Configuration, Monitoring, Management	IDS and IPS Engines	Tier 2: Communication and connectivity
Endpoint Identify	Secure Configuration and Management	Root of Trust Sandboxing	Access Control Secure Boot	Physical Security		Tier 1: Endpoints and embedded software

Figure 2.12: Four-tier industrial IoT security model

The four-tier model is explained as follows:

- **Tier 1—Endpoints and embedded software**: In IIoT deployments, security must extend from the silicon to the software layers of device endpoints. IIoT endpoints range from resource-constrained field devices to enterprise-grade servers and routers with significant storage and compute capabilities. Many industrial deployments include legacy devices with insecure protocol stacks. This provides a unique environment where security must not be limited to the network perimeter, but extend up to the endpoints. `Chapter 3`, *IIoT Identity and Access Management*, and `Chapter 4`, *Endpoint Security and Trustworthiness*, discuss the challenges involved in IIoT endpoint security, and present various endpoint security methodologies and solutions, such as access and identity management, establishing root of trust and trust chains, secure boot and firmware/software upgrades, partitioning, and more.
- **Tier 2—Communications and connectivity**: This tier focuses on securing data in use and in motion through secured transport, deep packet inspection, intrusion detection and prevention, secured communication protocols, and more. In Chapter 5, *Securing Connectivity and Communications*, the challenges and solutions of securing IIoT connectivity and communication have been dealt with in depth.

- **Tier 3—Cloud platform and applications:** This is the third tier that needs to be secured. Cloud-based IIoT deployments extend the attack surface significantly. IIoT use cases involve mission-critical command and controls with low latency requirements, which presents a unique set of security challenges at this tier. Cloud platform services often extend to the industrial edge, and as such need to factor in special attack vectors and mitigation strategies. Security architectures and methodologies to protect the industrial edge, cloud, and applications are discussed in depth in `Chapter 6`, *Securing IIoT Edge, Cloud, and Apps*.
- **Tier 4—Process and governance**: Practical security management requires a risk-based approach to "right-size" security investments. Security management must cut across the entire lifecycle, from design through operations. IIoT stakeholders must also play their respective roles to secure IIoT deployments.

Every organization that adopts and implements industrial IoT would benefit by having policies and governance guidelines for threat prevention and risk management. This is an essential component of meeting security objectives and business goals with industrial IoT. Security standards developed by industry organizations such as NIST, IEEE, and so on, and also open industry standards, need to be evaluated and suitably adopted at the design and planning phase of any IoT deployment. In addition, use case specific security models and policies need to be developed around configuration and management, data protection, connectivity, endpoint protection, threat analysis, and so on.

`Chapter 7`, *Secure Processes and Governance*, provides more insights into the risk management aspects of industrial IoT. It also reviews existing standards and governance principles to develop a successful security governance model for businesses.

Summary

This chapter presented a primer of attacks, countermeasures, and threat modeling, which lays the foundation for effective risk analysis and mitigation. It also provided the readers with insights into the distinguishing characteristics of trustworthiness for IIoT systems and the functional components of the industrial big data pipeline.

IIoT systems are highly complex; this chapter presented IIoT architectural viewpoints and patterns as developed by the IIC to provide you with a crisp understanding of end-to-end IIoT system components. Based on usage, operations, and functional domains, the IIoT security architecture was decomposed into a four-tier security model, which has been further elaborated in the subsequent chapters of the book.