# Module-4
# Chapter-7

# Data and Analytics for IoT



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# An Introduction to Data Analytics for IoT

- In the world of IoT, the **creation of massive amounts of data** from sensors is common and one of the **biggest challenges**—not only from a transport perspective but also from a data management standpoint.

- **For ex** : deluge of data that can be generated by IoT is found in the commercial aviation industry and the sensors that are deployed throughout an aircraft.

- **Modern jet engines** are fitted with thousands of sensors that generate a whopping **10GB of data per second**.

- For example, modern jet engines, similar to the one shown in figure 7.1, may be equipped with around **5000 sensors.**

- Therefore, a twin engine commercial aircraft with these engines operating on average 8 hours a day will generate over 500 TB of data daily, and this is just the data from the engines.

- The potential for a petabyte (PB) of data per day per commercial airplane is not farfetched—and this is just for one airplane.

- Across the world, there are approximately 100,000 commercial flights per day. The amount of IoT data coming just from the commercial airline business is overwhelming.

- This example is but one of many that highlight the big data problem that is being exacerbated(worsened or intensified) by IoT.

- **Analyzing this amount of data in the most efficient manner possible falls under the umbrella of data analytics.**

**Figure 7.1** : Commercial Jet Engine

# Structured Versus Unstructured Data

- **Structured data and unstructured data** are important classifications as they typically require different toolsets from **a data analytics perspective.**

- The figure 7.2 provides a **high-level comparison of structured data and unstructured data.**

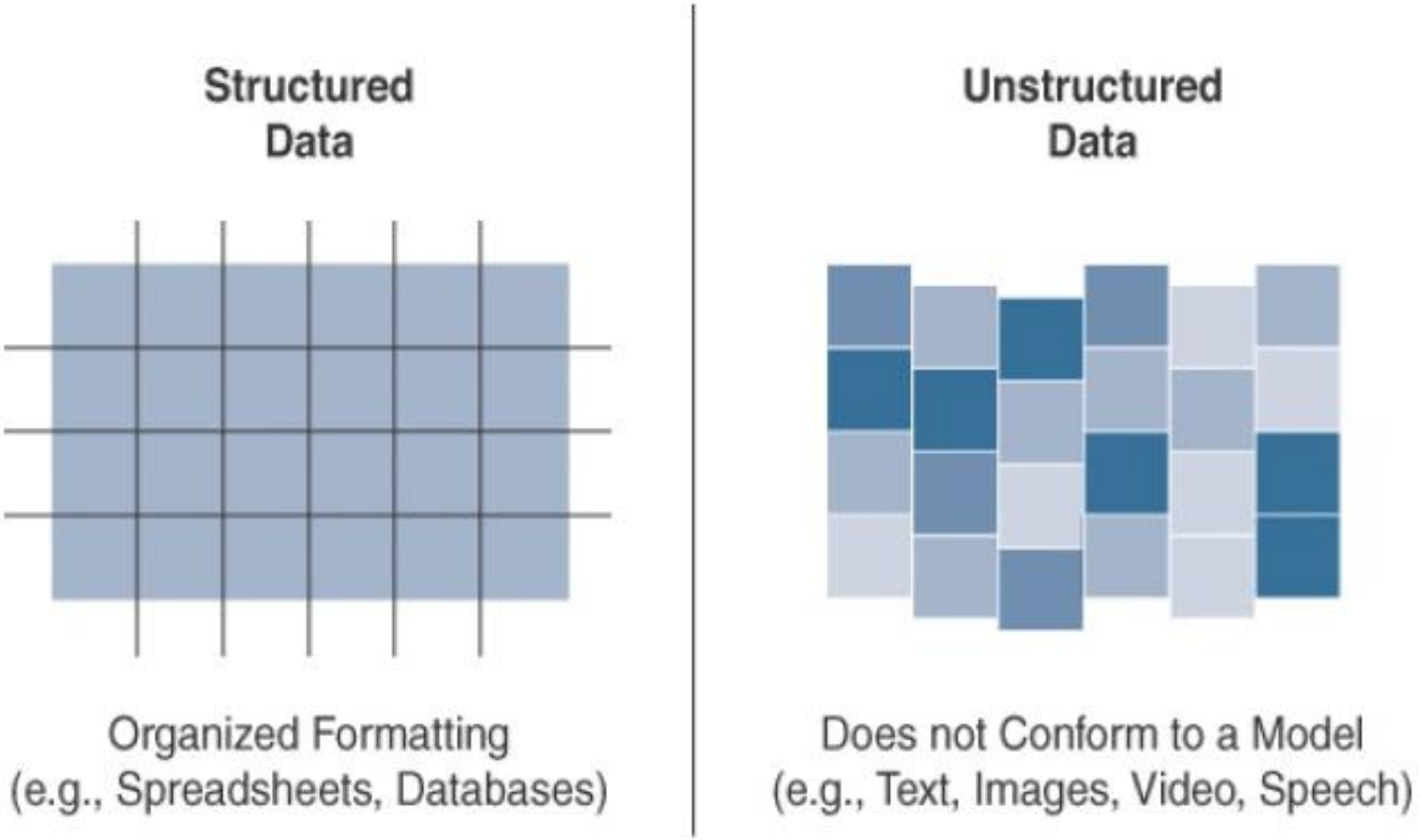


**Figure 7.2** : Comparison Between Structured and Unstructured Data

- **Structured data** means that the data follows a model or schema that defines **how the data is represented or organized**, meaning it fits well with a traditional relational database management system (RDBMS).

- A **structured data** may be in a simple tabular form.

- For ex **: a spreadsheet** where data occupies a specific cell and can be explicitly defined and referenced.

- **Structured data** can be found in most computing systems which includes banking transaction, invoices to computer log files, router configuration etc.

- **IoT sensor data** often uses structured values, such as temperature, pressure, humidity, and so on, which are all sent in a known format.

- **Structured data** is **easily** formatted, stored, queried, and processed; for these reasons, it has been the core type of data used for making business decisions

- Because of the **highly structured data**, a wide array of data analytics tools are available for processing this type of data. For ex : Microsoft Excel and Tableau

- **Unstructured data** lacks a logical schema for understanding and decoding the data through traditional programming means.

- Examples of this data type include text, speech, images, and video.

- As a **general rule, any data that does not fit** neatly into a predefined data model is classified as unstructured data

- According to some estimates, around **80% of a business's data is unstructured.**

- Because of this fact, **data analytics methods** that can be applied to **unstructured data,** such as cognitive computing and machine learning that draw a lot of attention these days.

- A **third data classification, semi-structured data,** is sometimes included **along with structured and unstructured data.**

- A **semi-structured data** is a hybrid of structured and unstructured data and shares characteristics of both.
- **Examples for semi-structured data** included **Email, JSON and XML scripts.**
- **Smart objects in IoT networks** generate both structured and unstructured data.
- Structured data is more easily managed and processed due to its well-defined organization.
- On the other hand, unstructured data can be harder to deal with and typically requires very different analytics tools for processing the data.

# Data in Motion Versus Data at Rest

- As in most networks, **data in IoT networks** is either in transit **("data in motion")** or being held or stored **("data at rest").**
- **Examples** of **data in motion** include traditional client/server exchanges, such as web browsing and file transfers, and email.
- Data saved to a hard drive, storage array, or USB drive is **data at rest.**
- From **an IoT perspective**, the data from smart objects is considered data in motion as it passes through the network en route to its final destination.

- This is often **processed at the edge**, **using fog computing**.
- When data is processed at the edge, it may be **filtered and deleted or forwarded** on for further processing and possible storage at a fog node or in the data center.
- When **data arrives at the data center,** it is possible to process it in real-time, just like at the edge, while it is still in motion.
- **Various tools** such as **Spark, Storm and Flink** are available to this job.

- **Data at rest** in IoT networks can be typically found in IoT brokers or in some sort of storage array at the data center.
- Myriad tools, especially tools for structured data in relational databases, are available from a data analytics perspective.
- The best known of these tools is **Hadoop**.

Module-4
Chapter-7

# Data and Analytics for IoT

## Topic : IoT Data Analytics Overview



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# IoT Data Analytics Overview

- The true relevance of IoT data from **smart objects** is realized only when the analysis of the data leads to actionable business intelligence and insights.

- Data analysis is typically broken down by the types of results that are produced. As shown in the figure 7.3, there are four types of data analysis results:

  i. Descriptive
  ii. Diagnostic
  iii. Predictive
  iv. Prescriptive

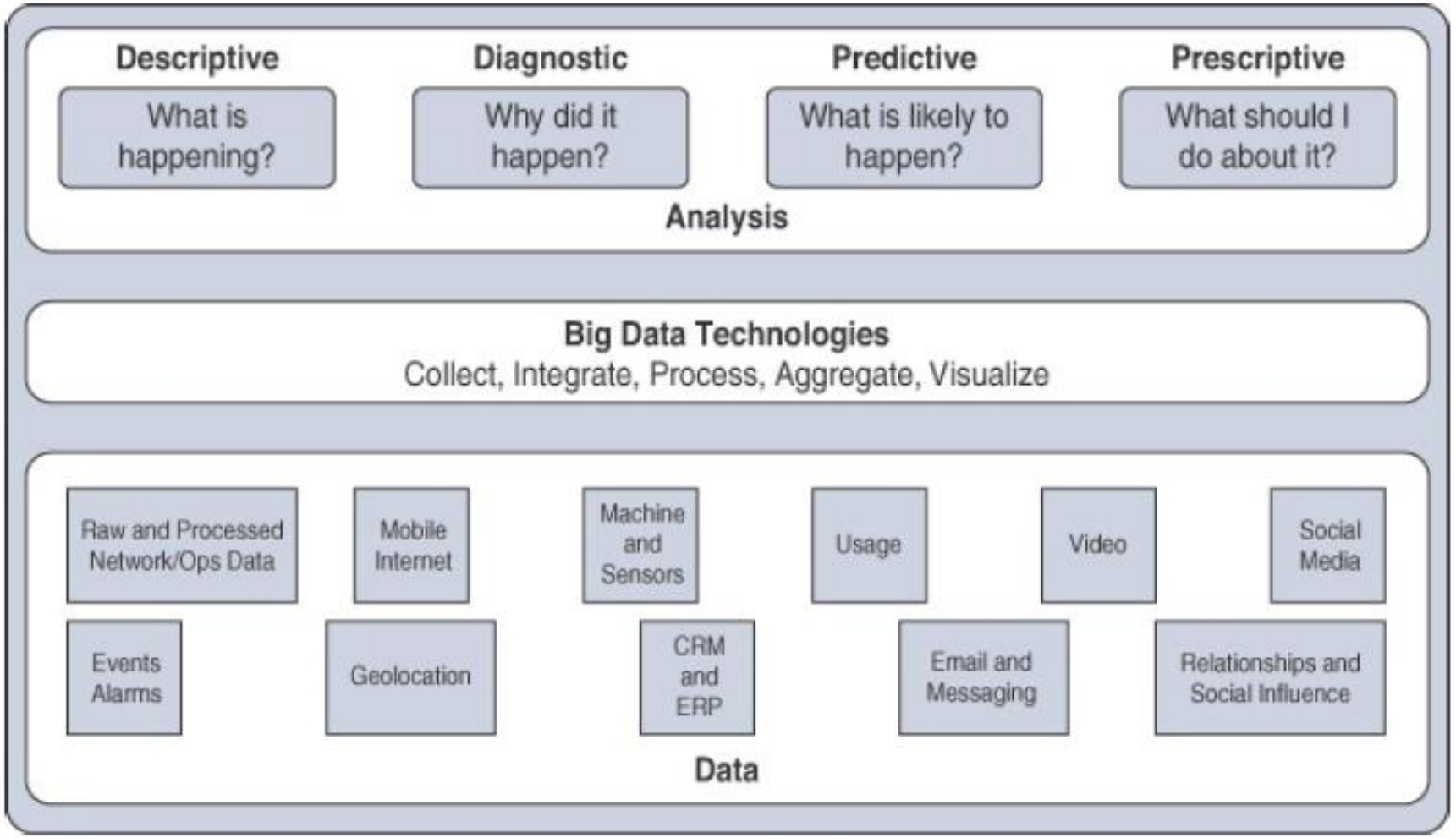


**Figure 7.3** : Types of Data Analysis Results

# Descriptive

- Descriptive data analysis tells you what is happening, either now or in the past.
- For ex : a thermometer in a truck engine reports temperature values every second. From a descriptive analysis perspective, we can pull this data at any moment to gain insight into the current operating condition of the truck engine.
- If the temperature value is too high, then there may be a cooling problem or the engine may be experiencing too much load.

# Diagnostic

- When we are interested in the **"why,"** diagnostic data analysis can provide the answer.
- Continuing with the example of the temperature sensor in the truck engine, we might wonder why the truck engine failed.
- Diagnostic analysis might show that the temperature of the engine was too high, and the engine overheated.
- Applying diagnostic analysis across the data generated by a wide range of smart objects can provide a clear picture of why a problem or an event occurred.

# Predictive

- Predictive analysis aims to foretell problems or issues before they occur.
- For ex : with historical values of temperatures for the truck engine, predictive analysis could provide an estimate on the remaining life of certain components in the engine.
- These components could then be proactively replaced before failure occurs. Or if temperature values of the truck engine start to rise slowly over time, this could indicate the need for an oil change or some other sort of engine cooling maintenance.

# Prescriptive

- Prescriptive analysis goes a step beyond predictive and recommends solutions for upcoming problems.
- A prescriptive analysis of the temperature data from a truck engine might calculate various alternatives to cost-effectively maintain our truck.
- These calculations could range from the cost necessary for more frequent oil changes and cooling maintenance to installing new cooling equipment on the engine or upgrading to a lease on a model with a more powerful engine.

- Both predictive and prescriptive analyses are more resource intensive and increase complexity, but the value they provide is much greater than the value from descriptive and diagnostic analysis.

- We can see that descriptive analysis is the least complex and at the same time offers the least value. On the other end, prescriptive analysis provides the most value but is the most complex to implement.

The figure 7.4 illustrates the four data analysis types and how they rank as complexity and value increase.

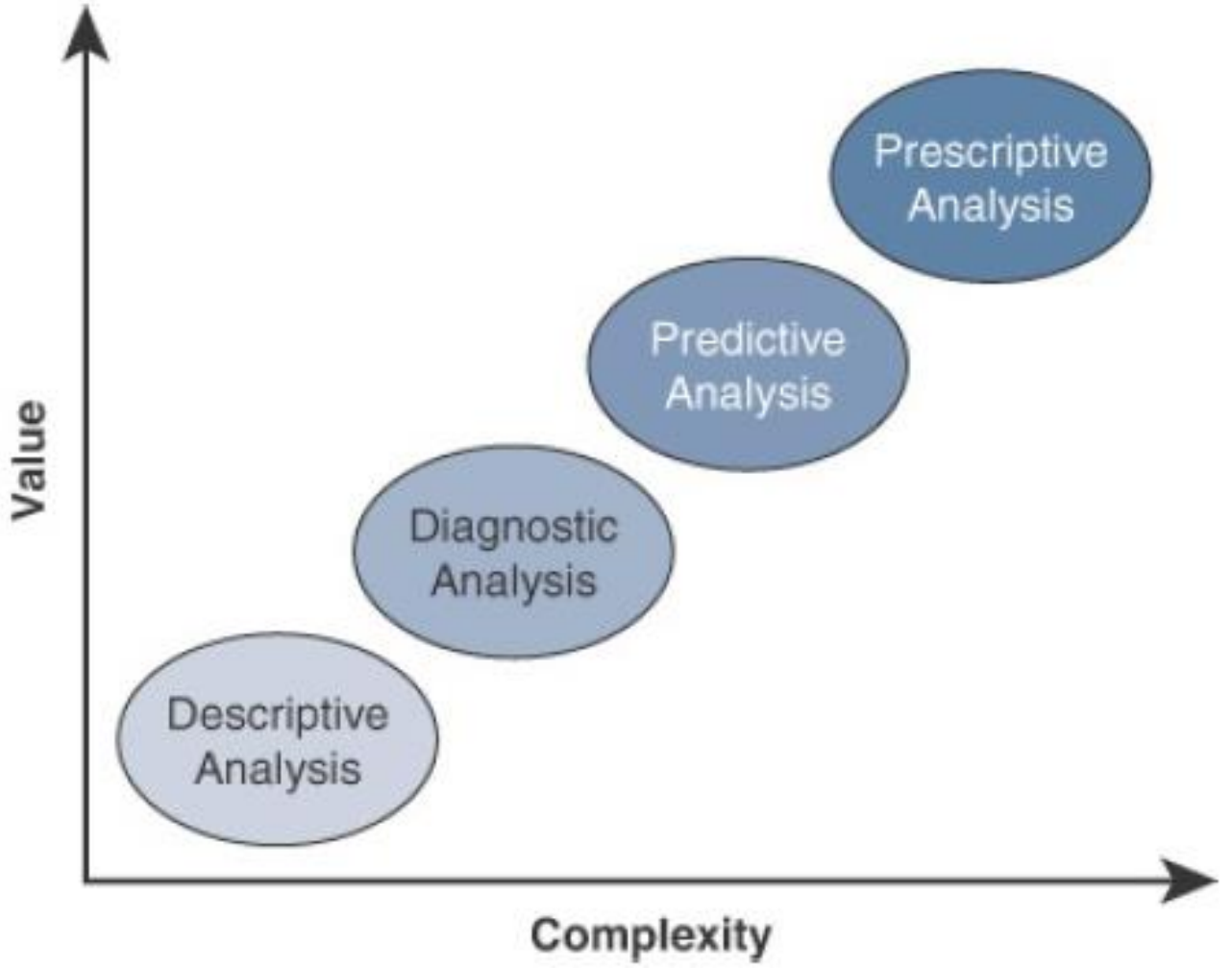


**Figure 7.4 :** Application of Value and Complexity Factors to the Types of Data Analysis

Module-4
Chapter-7

# Data and Analytics for IoT

## Edge Streaming Analytics &Network Analytics



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# Edge Streaming Analytics

- In the world of IoT, vast quantities of data are generated on the fly and often need to be **analyzed** and responded to **immediately.**

- Not only is the **volume of data generated at the edge** immense—meaning the bandwidth requirements to the cloud or data center need to be engineered to match.

- But the data may be so **time sensitive** that it needs immediate attention, and waiting for deep analysis in the cloud simply isn't possible.

## Comparing Big Data and Edge Analytics

- The term ***big data***, usually refers to **unstructured data** that has been collected and stored in the cloud.

- The **data is collected over time** so that it can be analyzed through ***batch-processing tools***, such as an RDBMS, Hadoop, or some other tool, at which point business insights are gained, and value is drawn from the data.

- Tools like Hadoop and MapReduce are great at tackling problems that require deep analytics on a large and complex quantity of unstructured data.

- Due to their distance from the **IoT endpoints** and the **bandwidth** required to bring all the data back to the cloud, they are generally not well suited to real-time analysis of data as it is generated.

- **Streaming analytics** allows you to **continually monitor** and *assess data in real-time* so that you can adjust or fine-tune our predictions as the race progresses.

- In the context of IoT, with **streaming analytics** performed at the **edge** it is possible to process and act on the data in realtime without waiting for the results from a future batch-processing job in the cloud.

- The key values of edge streaming analytics include the following:

- **Reducing data at the edge**
- **Analysis and response at the edge**
- **Time Sensitivity**

## Reducing data at the edge

- The aggregate data generated by IoT devices is generally **in proportion to** the number of devices.
- The **scale** of these devices is likely to be **huge**, and so is ***the quantity*** of data they generate.
- Passing all this data to the cloud is **inefficient** and is unnecessarily **expensive** in terms of **bandwidth and network infrastructure.**

- **Analysis and response at the edge**
  - Some data is useful only at the edge (such as a factory control feedback system).
  - In cases such as this, the *data is best analyzed and acted upon where it is generated.*

- **Time Sensitivity**
  - When timely response to data is required, passing data to the cloud for future processing results in ***unacceptable latency.***
  - **Edge analytics** allows immediate responses to changing conditions.

## Edge Analytics Core Functions

- To perform analytics at the edge, data needs to be viewed as real-time flows. *Streaming analytics* at the edge can be broken down into **three simple stages:**

1. **Raw input data**
2. **Analytics processing unit (APU)**
3. **Output streams**

i. **Raw input data**

- This is **the raw data** coming from the sensors into the analytics processing unit.

ii. **Analytics processing unit (APU)**

- The **APU filters and combines data streams** (or separates the streams, as necessary), **organizes** them by time windows, and performs various analytical functions.

- It is at this point that the results may be acted on by micro services **running in the APU.**

### iii. Output streams

- The data that is output is organized into insightful streams and is used to influence the behavior of smart objects, and passed on for storage and further processing in the cloud.

- Communication with the cloud often happens through a standard **publisher/subscriber messaging protocol, such as MQTT.**

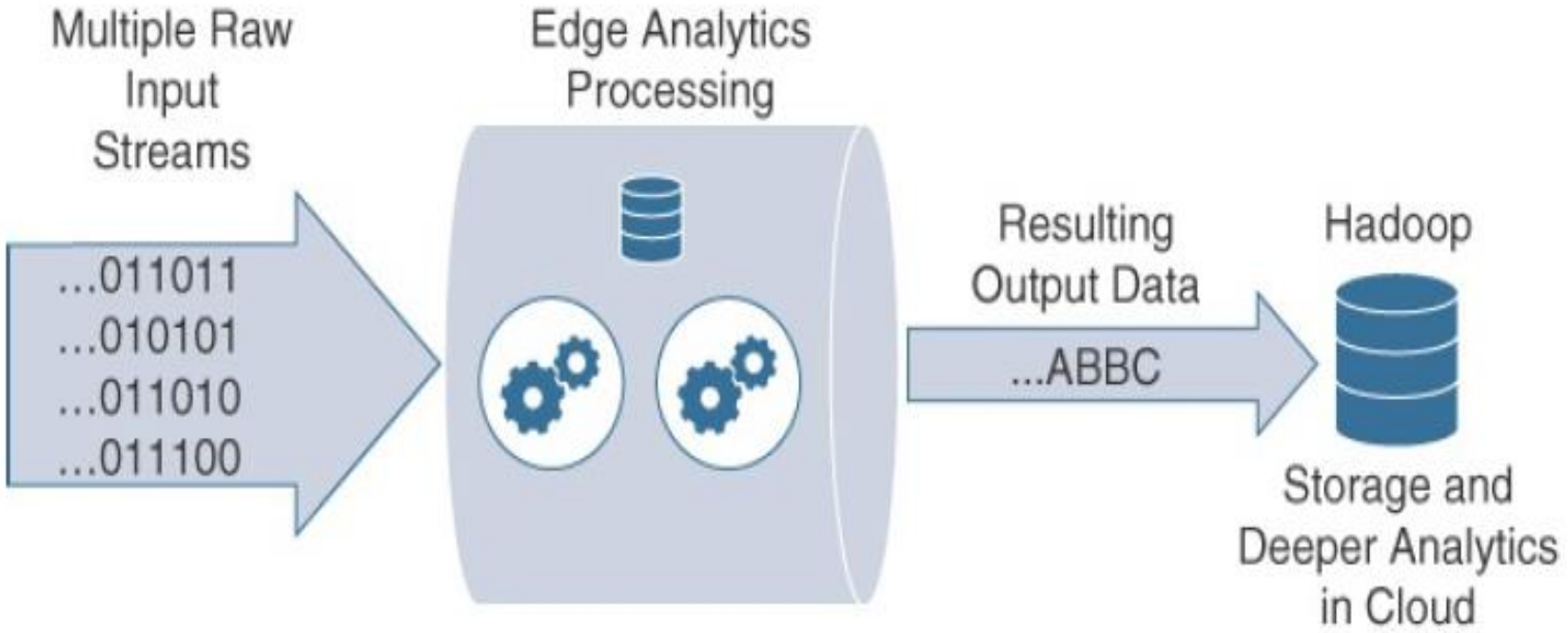


**Figure 7.12** : Edge Analytics Processing Unit

- The figure 7.12 illustrates the stages of data processing in an edge APU.

- In order to perform **analysis in real-time**, the **APU needs** to perform the following **functions**:
  - **Filter**
    - The streaming data generated by IoT endpoints is likely to **be very large,** and most of **it is irrelevant**.
    - For ex : a sensor may simply poll on a regular basis to confirm that it is still reachable.
    - This information is not really relevant and can be mostly ignored. The filtering function identifies the information that is considered important.

## Transform

- In the data warehousing world, **Extract, Transform, and Load (ETL)** operations are used to **manipulate** the **data structure** into a form that can be used for other **purposes.**

- Analogous to data warehouse ETL operations, in **streaming analytics**, once the data is *filtered, it needs to be formatted for processing.*

- **Time**
  - As the real-time streaming data flows, a timing context needs to be established. This could be to correlated average temperature readings from sensors on a minute-by-minute basis.

- **Correlate**
  - **Streaming data analytics** becomes most useful when multiple data streams are combined from different types of sensors.

- These **different types** of data come from different instruments, but when this data is combined and analyzed, it provides an invaluable picture of the health of the patient at any given time.

- Another key aspect is combining and correlating real-time measurements with preexisting, or historical, data.

- Combining *historical data gives the live streaming data a powerful context and promotes* more insights into the current condition of the patient as shown in the figure 7.14.

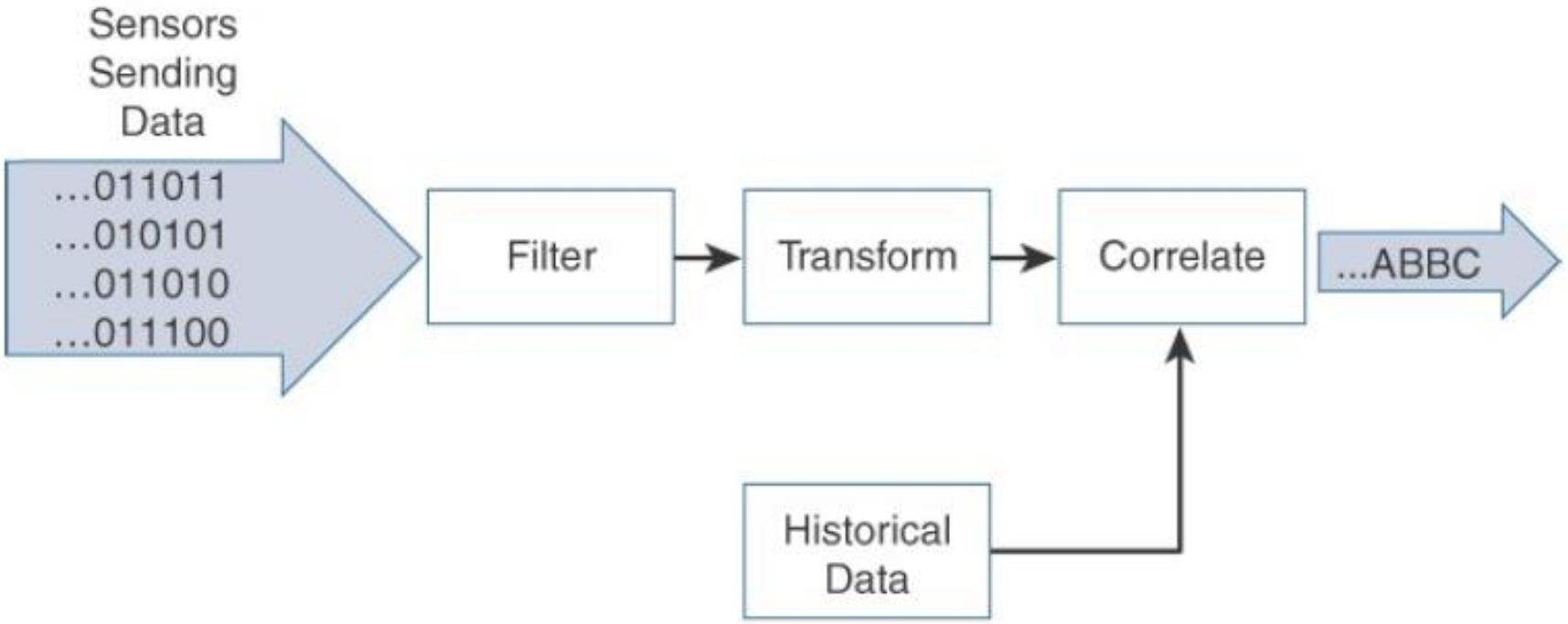


**Figure 7.14** : Correlating Data Streams with Historical Data

- **Match patterns**

- Once the **data streams** are properly cleaned, transformed, and correlated with other live streams as well as historical data sets, pattern matching operations are used to gain deeper insights to the data.

- For ex : say that the APU has been collecting the patient's vitals for some time and has gained an understanding of the expected patterns for each variable being monitored.

- If an unexpected event arises, such as a sudden change in heart rate or respiration, the pattern matching operator recognizes this as out of the ordinary and can take certain actions, such as generating an alarm to the nursing staff.

- The patterns can be **simple relationships**, or they may be **complex**, based on the criteria defined by the application.

- **Improved Business Intelligence**
  - Ultimately, the **value of edge analytics** is in the improvements to **business intelligence** that were not previously available.
  - For ex : conducting edge analytics on patients in a hospital allows staff to respond more quickly to the patient's changing needs and also reduces the volume of unstructured (and not always useful) data sent to the cloud.

# Distributed Stream Analytics

- Depending on the application and network architecture, analytics can happen at any point throughout the IoT system.

- **Streaming analytics may be performed directly at the edge, in the fog, or in the cloud data center**.

- There are no **hard and-fast** rules dictating where analytics should be done, but there are a few guiding principles.

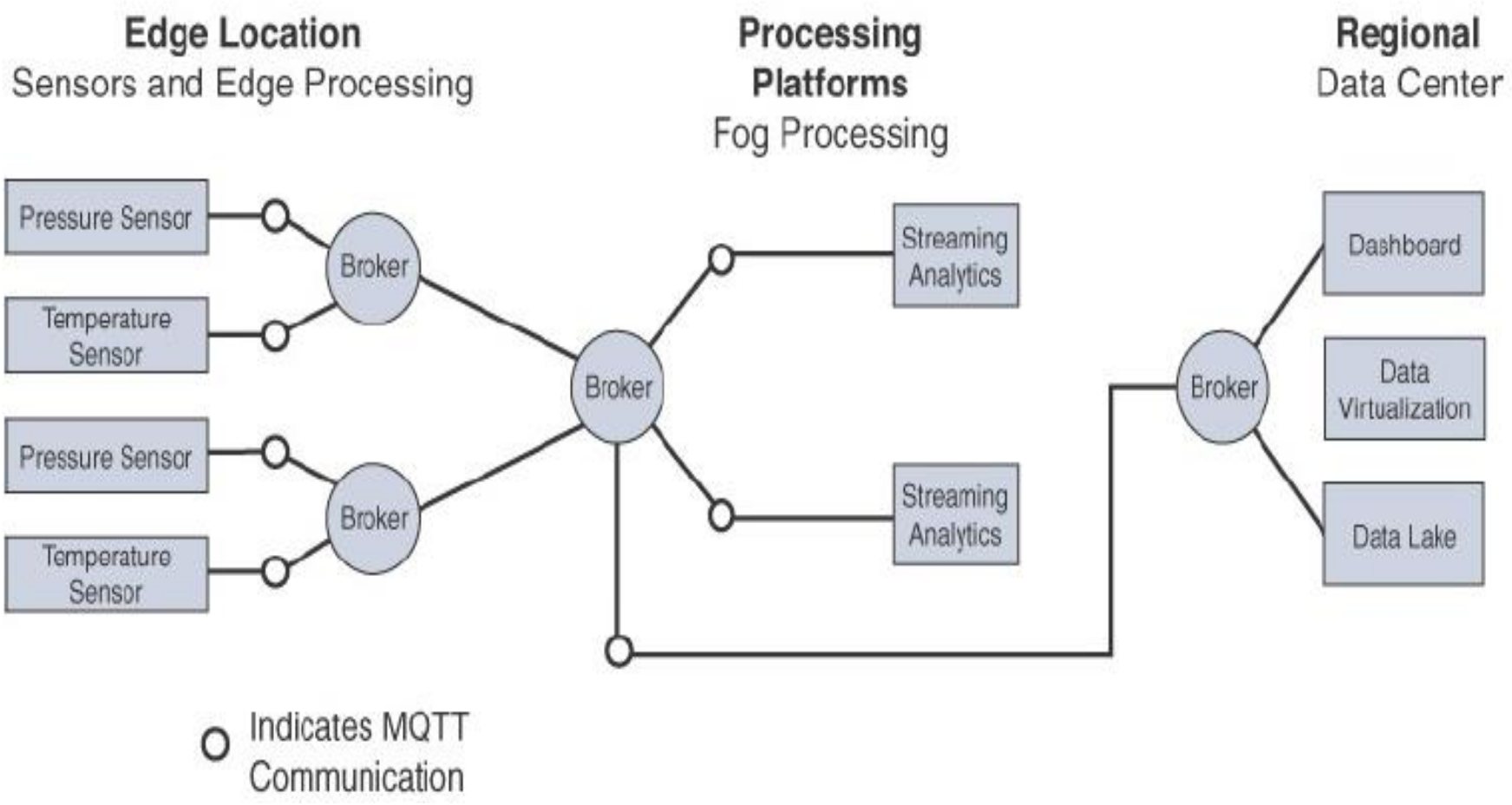


**Figure 7.15** : Distributed Analytics Throughout the IoT System

- The figure 7.15 shows an example of an oil drilling company that is measuring both pressure and temperature on an oil rig.

- While there may be some value in doing analytics directly on the edge, in this example, the sensors communicate via **MQTT** through a message broker to the fog analytics node, allowing a broader data set.

- The fog node is located on the same oil rig and performs **streaming analytics from several edge devices,** giving it better insights due to the expanded data set.

# Network Analytics

- Another form of analytics that is extremely important in managing IoT systems is **network-based analytics**.

- **Data analytics** is concerned with **finding patterns** in the data generated by endpoints whereas network analytics is concerned with **discovering patterns** in the communication flows from a network traffic perspective.

- **Network analytics** has the power to analyze details of communications patterns made by **protocols and correlate** this across the network.

- It allows us to understand what should be considered normal behavior in a network and to **quickly identify** anomalies that suggest network problems due to suboptimal paths, intrusive malware, or excessive congestion.

- The benefits of flow analytics, in addition to other network management services, are as follows:
- **Network traffic monitoring and profiling**
- **Application traffic monitoring and profiling**
- **Capacity planning**
- **Security analysis**
- **Accounting**
- **Data warehousing and data mining**

- **Network traffic monitoring and profiling**
  - *Flow collection from the network layer provides global and distributed near-real-time monitoring capabilities*.
  - **IPv4 and IPv6** network wide traffic volume and pattern analysis helps administrators proactively detect problems and quickly troubleshoot and resolve problems when they occur.

- **Application traffic monitoring and profiling**

  - **Monitoring and profiling** can be used to gain a detailed time-based view of IoT access services, such as the **application-layer protocols**, including MQTT, CoAP, and DNP3, as well as the associated applications that are being used over the network.

- **Capacity planning**

  - **Flow analytics** can be used to track and anticipate IoT traffic growth and help in the planning of upgrades when deploying new locations or services by analyzing captured data over a long period of time.

  - This analysis affords the **opportunity to track and anticipate** IoT network growth on a continual basis.

- **Security analysis**

  - Because most IoT devices typically generate a low volume of traffic and always send their data to the same server(s), any change in network traffic behavior may indicate a cyber security event, such as a denial of service **(DoS)** attack.

## Accounting

- In field area networks, **routers or gateways** are often physically isolated and leverage public cellular services and VPNs for backhaul.
- Deployments may have thousands of gateways connecting thelast-mile IoT infrastructure over a cellular network.
- **Flow monitoring** can thus be leveraged to analyze and optimize the billing, in complement with other dedicated applications, such as Cisco Jasper, with a broader scope than just monitoring data flow.

- **Data warehousing and data mining**
  - **Flow data (or derived information)** can be warehoused for later retrieval and analysis in support of proactive analysis of multiservice IoT infrastructures and applications.

# Flexible NetFlow Architecture

- Flexible NetFlow (FNF) and IETF IPFIX (RFC 5101, RFC 5102) are examples of protocols that are widely used for networks.
- FNF is a **flow technology** developed by **Cisco Systems** that is widely deployed all over the world.
- Key advantages of FNF are as follows:
  - Flexibility, scalability, and aggregation of flow data.
  - Ability to monitor a wide range of packet information and produce new information about network behavior

- Enhanced network anomaly and security detection.
- User-configurable flow information for performing customized traffic identification and ability to focus and monitor specific network behavior.
- Convergence of multiple accounting technologies into one accounting mechanism.

## FNF Components

- FNF has the following main components, as shown in the figure 7.17.

- **FNF Flow Monitor (NetFlow cache)**
- **FNF flow record**
- **FNF Exporter**
- **Flow export timers**
- **NetFlow export format**
- **NetFlow server for collection and reporting**

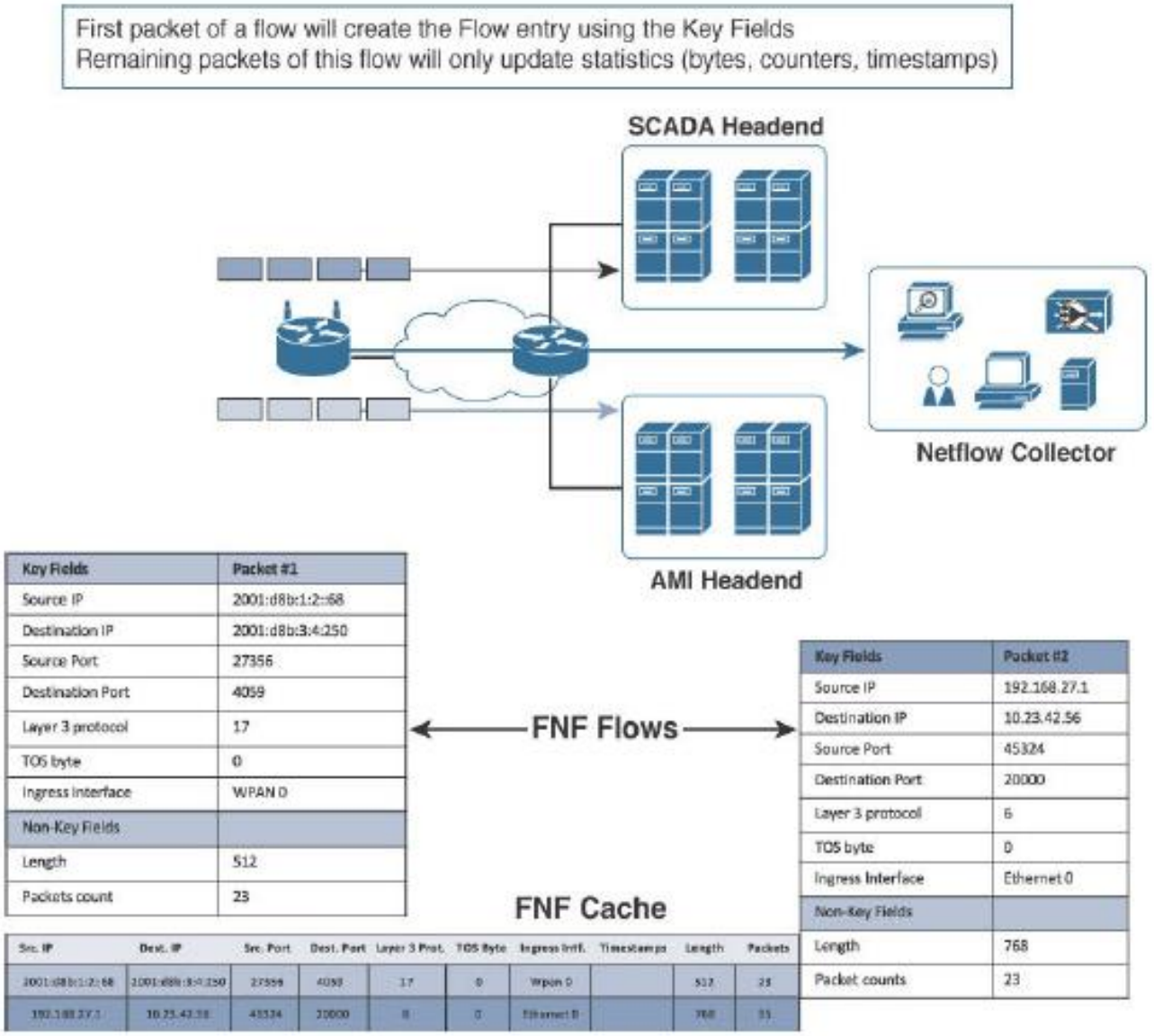


**Figure 7.17** : Flexible NetFlow overview

- **FNF Flow Monitor (NetFlow cache):**
  - The FNF Flow Monitor describes the NetFlow cache or information stored in the cache.
  - The **Flow Monitor** contains the flow record definitions with key fields and non-key fields within the cache.
  - Also, **part of the Flow Monitor** is the Flow Exporter, which contains information about the export of NetFlow information, including the destination address of the NetFlow collector.

- **FNF flow record**
  - A **flow record** is a set of key and non-key NetFlow field values used to characterize flows in the **NetFlow cache.**
  - Flow records may be **predefined** for ease of use or customized and user defined.
  - **User-defined records** allow selections of specific key or non-key fields in the flow record

## FNF Exporter

- There are **two primary methods** for accessing NetFlow data: **Using the show commands** at the command-line interface (CLI), and using an application reporting tool.

- NetFlow Export, unlike SNMP polling, pushes information periodically to the NetFlow reporting collector.

- The **Flexible NetFlow Exporter** allows the user to define where the export can be sent, the type of transport for the export, and properties for the export. Multiple exporters can be configured per Flow Monitor.

- **Flow export timers**
  - Timers indicate how often flows should be **exported to the collection and reporting server.**
- **NetFlow export format**
  - This simply indicates the **type of flow reporting format.**
- **NetFlow server for collection and reporting**
  - This is the **destination of the flow export**. It is often done with an analytics tool that looks for anomalies in the traffic patterns

# Flexible NetFlow in Multiservice IoT Networks

- **FNF** be configured on the routers that **aggregate connections from the last mile's routers.**
- This gives a global view of all services flowing between the core network in the cloud and the IoT last-mile network.
- **Flow analysis** at the gateway is not possible with all IoT systems.

- Some other challenges with deploying flow analytics tools in an IoT network include the following:

  - The distributed nature of fog and edge computing may mean that traffic flows are processed in places that might not support flow analytics, and visibility is thus lost.

  - **IPv4 and IPv6** native interfaces sometimes need to inspect inside VPN tunnels, which may impact the ***router's performance.***

  - Additional network management traffic is generated by **FNF reporting devices.**

Module-4
Chapter-7

# Data and Analytics for IoT

## IoT Data Analytics Challenges & ML



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# IoT Data Analytics Challenges

- As IoT has grown and evolved, it has become clear that traditional data analytics solutions were not always adequate.

- IoT data places **two specific challenges** on a relational database:

- **Scaling problems**
- **Volatility of data**

- **Scaling problems**
  - Due to the large number of smart objects in most IoT networks that continually send data, relational databases can grow incredibly large very quickly.
  - This can **result in performance issues** that can be costly to resolve, often requiring **more hardware and architecture changes.**

- **Volatility of data**
  - IoT data, however, is volatile in the sense that the data model is likely to change and evolve over time.
  - A **dynamic schema** is often required so that data model changes can be made daily or even hourly.

# Dealing with Challenges of IoT Data Analytics

- To deal with **challenges like scaling and data volatility,** a different type of database, known as **NoSQL**, is being used.
- Structured Query Language (SQL) is the computer language **used to communicate with an RDBMS.**
- As the name implies, a NoSQL database is a database that does not use SQL.
- It is not set up in the traditional tabular form of a **relational database.**

- **Other challenges**

- IoT also brings challenges with the live streaming nature of its data and with managing data at the network level.

- Another challenge that IoT brings to analytics is in the area of network data, which is referred to as ***network analytics***.

# Machine Learning

- We often hear the words ***Machine Learning***, ***Deep Learning***, ***Neural Networks*** and ***Convolutional Neural Networks*** in relation to big data and IoT.

- *ML is indeed central to IoT.*

- **Data collected** by smart objects needs to be analysed, and intelligent actions need to be taken based on these analyses.

# Machine Learning Overview

- Machine learning is, in fact, part of a larger set of technologies commonly grouped under the term ***Artificial Intelligence*** **(AI)**.

- **AI** includes any technology that allows a computing system to mimic human intelligence using any technique, from very advanced logic to **basic "if-then else" decision loops.**

- ML is concerned with any process where the computer needs to receive a set of data that is processed to help perform a task with **more efficiency.**

- ML is a vast field but can be simply divided in **two main categories:**

  i. Supervised Learning
  ii. Unsupervised Learning.

# Supervised Learning

- In **supervised learning**, the machine is trained with **input for which there is a known correct answer/label.**
- With **supervised learning techniques**, hundreds or thousands of images are fed into the machine, and each image is **labeled** (human or nonhuman in this case).
- This is called the ***training set***.

- An algorithm is used to determine **common parameters** and **common differences** between the images.
- Each new image is compared to the set of known ***"good images"*** and a deviation is calculated to determine how different the new image is from the average human image.
- This process is called ***classification***.

- **After training,** the **machine** should be able to recognize human shapes.

- Before real field deployments, the machine is usually tested with unlabeled pictures— this is called the ***validation*** or the ***test set***.

- In other cases, the learning process is not about classifying in two or more categories but about **finding a correct value**.

- For ex :
  - The speed of the flow of oil in a pipe is a function of the size of the pipe, the viscosity of the oil, pressure, and a few other factors.
  - When you train the machine with measured values, the machine can predict the speed of the flow for a new, and unmeasured, viscosity.
- This process is called **regression;** regression predicts numeric values, whereas classification predicts **categories.**

# Unsupervised Learning

- In some cases, **supervised learning** is not the best method for a machine to help with a human decision.

- Once data **is recorded**, we can **graph** these elements in relation to one another (for ex: temperature as a function of pressure).

- We can then **input** this data into a computer and **use mathematical functions to find groups.**

- For ex : we may **decide to group** the engines by the sound they make at a given temperature.

- A standard function to operate this grouping, ***K-means clustering***, finds the mean values for a group of engines (for example, mean value for temperature, mean frequency for sound).

- All engines of the same type produce sounds and temperatures in the same range as the other members of the **same group.**

- "There will **occasionally** be an engine in the group that displays unusual characteristics (slightly out of expected temperature or sound range).

- This is the engine that you send **for manual evaluation.**

- The computing process associated with this determination is called ***unsupervised learning***."

- This type of learning is unsupervised because there is not a ***"good"*** or ***"bad"*** answer known in advance.

- The hundreds or thousands of parameters are computed, and small cumulated deviations in multiple dimensions are used to identify the exception.

- The figure 7.5 shows an example of such grouping and deviation identification logic.

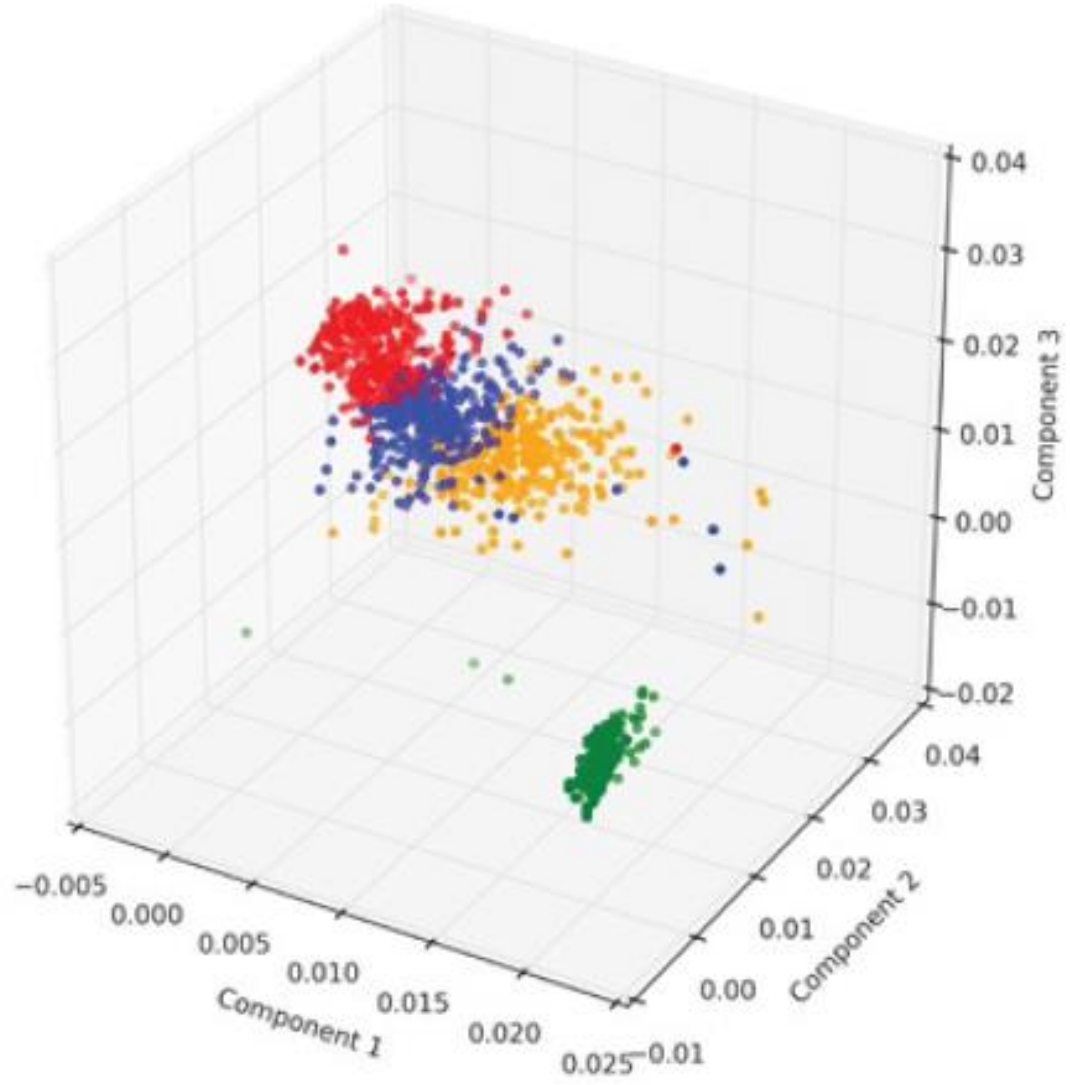


**Figure 7.5 :** Clustering and Deviation Detection Example

Module-4
Chapter-7

# Data and Analytics for IoT

## Neural Networks



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# Neural Networks

- Processing **multiple dimensions/features** requires a lot of computing power.
- Similarly, **supervised learning** is efficient only with a **large training set;**
- **Larger training** sets usually lead to **higher accuracy** in the **prediction**.
- **Training the machines** was often deemed too expensive and complicated.

- Distinguishing a **human from another mammal** and pickup truck from a van are much **more difficult**.

- In order to aid the machine in accomplishing this complex **task neural network** comes into picture.

- ***Neural networks*** are ML methods that mimic the way the human brain works.

- **Neural networks** mimic the **same logic.**

- The **information** goes through **different algorithms** (called units), each of which is in charge of processing an aspect of the information.

- The **resulting value** of one unit computation can be used directly or fed into another unit for further processing to occur.

- The **great efficiency of neural networks** is that each unit processes a simple test, and therefore computation is **quite fast.**

- This **model** is demonstrated in figure 7.6. By contrast, old supervised ML techniques would compare the human figure to potentially hundreds of thousands of images during the training phase, pixel by pixel, making them difficult and expensive to implement (with a lot of training needed) and slow to operate.

- The **neural networks** rely on the idea that information is divided into key components, and **each component** is assigned a **weight.** The **weights** compared together decide the classification of this information (no straight lines + face + smile = human).

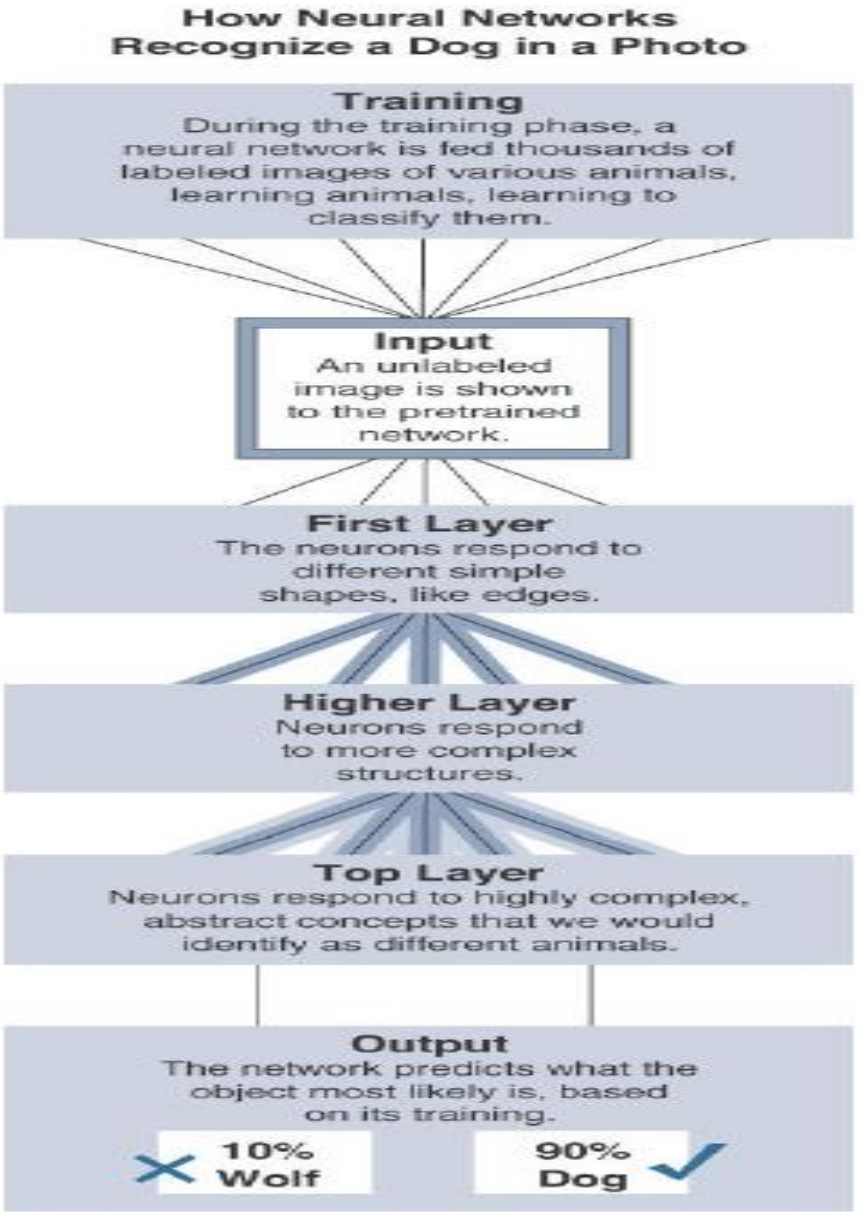


**Figure 7.6:** Neural Network Example

- When the **result of a layer** is fed into **another layer**, the process is called ***deep learning*** ("deep" because the learning process has more than a single layer).

- One **advantage** of deep learning is that having more layers allows for richer intermediate processing and representation of the data.

# Module-4
# Chapter-7

# Data and Analytics for IoT
## Hadoop



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **An Introduction to Data Analytics for IoT**
- **Machine Learning**
- **Big Data Analytics Tools and Technology**
- **Edge Streaming Analytics**
- **Network Analytics**

# Hadoop

- Hadoop is the most recent entrant into the data management market, but it is arguably the most popular choice as a data repository and processing engine.

- Initially, the project had two key elements:

  i. **Hadoop Distributed File System (HDFS)**: A system for storing data across multiple nodes.

  ii. **MapReduce**: A distributed processing engine that splits a large task into smaller ones that can be run in parallel

- Both **MapReduce and HDFS** take advantage of this distributed architecture to store and process massive amounts of data and are thus able to leverage resources from all nodes in the cluster.

- For HDFS, this capability is handled by specialized nodes in the cluster, including ***NameNodes*** and ***DataNodes*** (see figure 7.8):

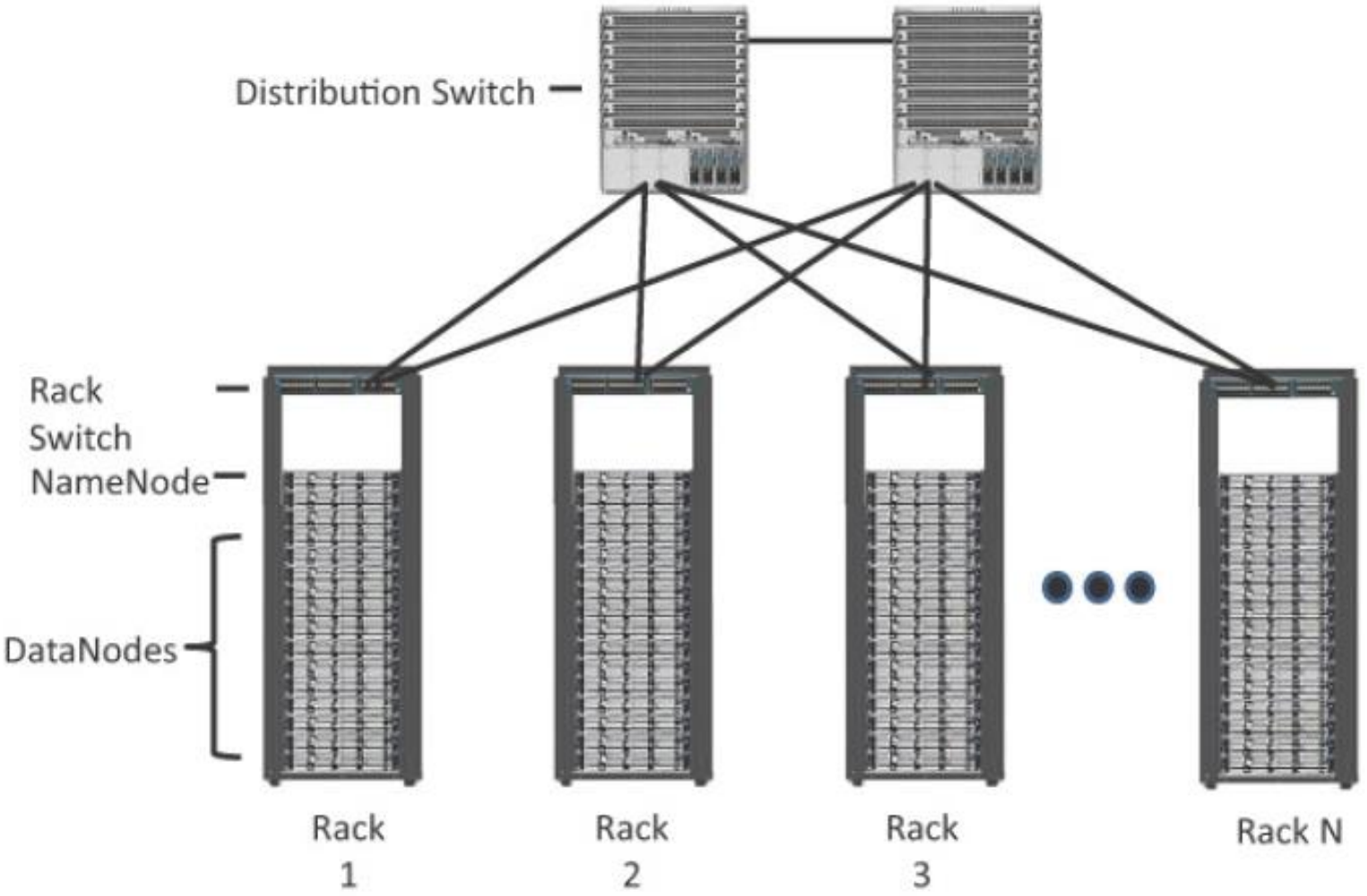


**Figure 7.8 :** Distributed Hadoop Cluster

- **NameNodes :**
  - These are a critical piece in data adds, moves, deletes, and reads on HDFS.
  - They coordinate where the data is stored, and maintain a map of where each block of data is **stored** and where it is **replicated.**
  - All interaction with HDFS is coordinated through the primary (active) **NameNode,** with a secondary (standby) **NameNode** notified of the changes in the event of a failure of the primary.

- The **NameNode** takes write requests from clients and distributes those files across the available nodes in configurable block sizes, usually 64 MB or 128 MB blocks.

- The **NameNode** is also responsible for instructing the **DataNodes** where replication should occur.

## ➢ DataNodes :

- These are the **servers** where the data is stored at the direction of the **NameNode.**
- It is common to have many **DataNodes in a Hadoop cluster** to store the data.
- Data blocks are distributed across several nodes and often are replicated three, four, or more times across nodes for redundancy.
- **Once data is written to one of the DataNodes,** the DataNode selects two (or more) additional nodes, based on replication policies, to ensure data redundancy across the cluster.

- The figure 7.9 shows **the relationship** between **NameNodes and DataNodes** and how data blocks are distributed across the cluster.

- **MapReduce** leverages a similar model to batch process the data stored on the cluster nodes.

- **Batch processing** is the process of running a scheduled or ad hoc query across historical data stored in the HDFS.

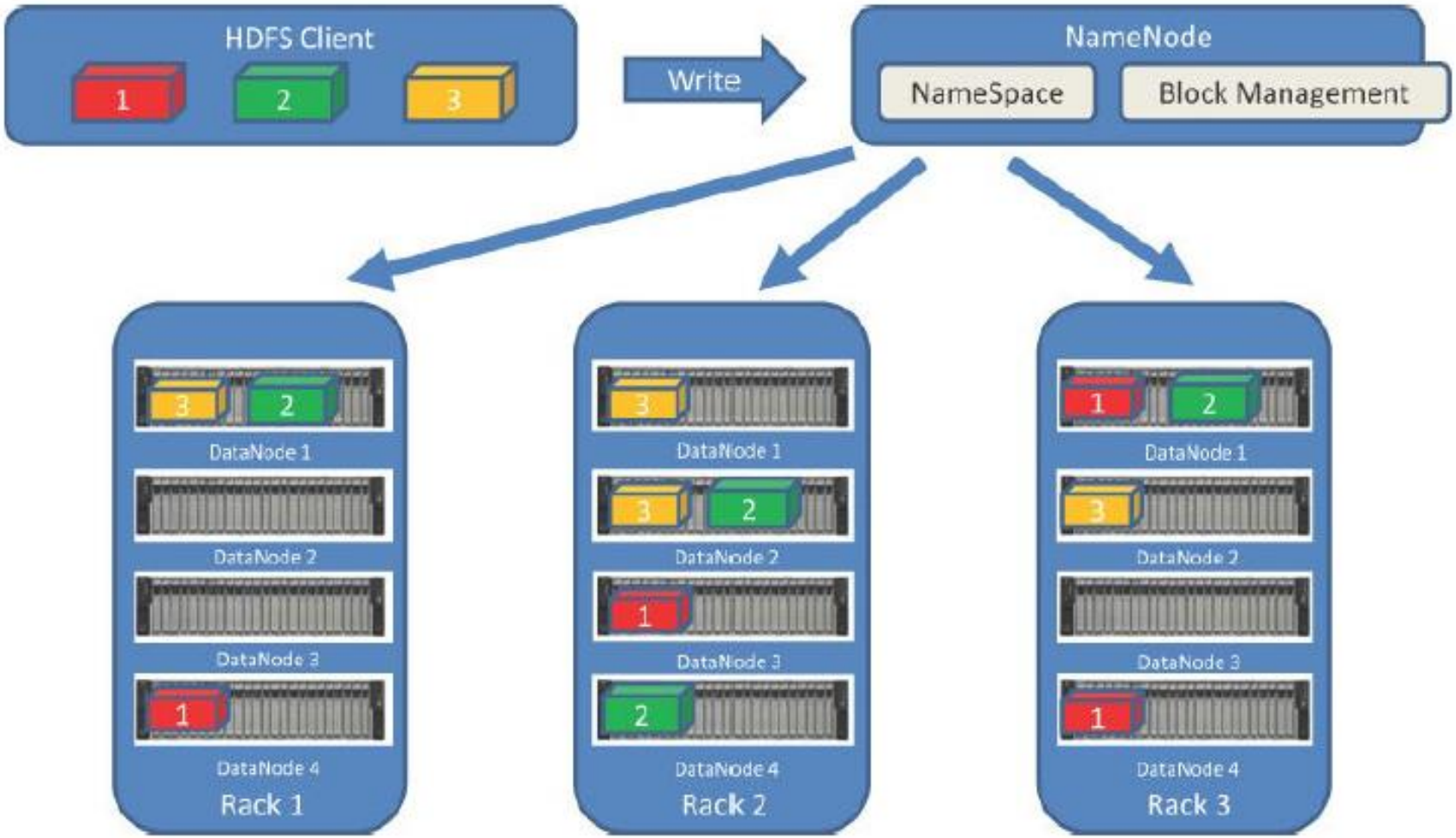


**Figure 7.9 :** Writing a File to HDFS

- A **query** is broken down into **smaller tasks and distributed** across all the nodes running MapReduce in a cluster.

- While this is useful for **understanding patterns and trending** in historical sensor or machine data, it has one **significant drawback: time.**

# YARN

- Introduced with **version 2.0 of Hadoop, YARN (Yet Another Resource Negotiator)** was designed to enhance the functionality of MapReduce.

- With the initial release, MapReduce was responsible for batch data processing and job tracking and resource management across the cluster.

- **YARN** was developed to take over the resource negotiation and job/task tracking, allowing MapReduce to be responsible only for data processing.

- With the development of a dedicated cluster resource scheduler, Hadoop was able to add additional data processing modules to its core feature set, including interactive SQL and real-time processing.

# The Hadoop Ecosystem

- **Hadoop** plays an increasingly big role in the collection, **storage, and processing** of IoT data due to its highly scalable nature and its ability to work with **large volumes of data.**

- Many **organizations** have **adopted Hadoop clusters for storage and processing of data** and have looked for complimentary software packages to add additional functionality to their **distributed Hadoop clusters.**

- Since the initial release of **Hadoop in 2011**, many projects have been developed to add incremental functionality to Hadoop and have collectively become known as the ***Hadoop ecosystem***.

- **Apache Spark**

- **Apache Kafka**

# Apache Spark

- **Apache Spark** is an in-memory distributed data analytics platform designed to **accelerate processes** in the Hadoop ecosystem.

- The ***"in-memory"*** characteristic of Spark is what enables it to **run jobs very quickly.**

- At each stage of a MapReduce operation, the data is read and written back to the disk, which means latency is introduced through each disk operation.

- **Real-time processing** is done by a component of the Apache Spark project called **Spark Streaming**.

- **Spark Streaming** is an extension of Spark Core that is responsible for taking live streamed data from a messaging system, like Kafka, and dividing it into smaller microbatches.

- These microbatches are called ***discretized streams***, or ***DStreams***.

- The **Spark processing** engine is able to operate on these smaller pieces of data, allowing rapid insights into the *data and subsequent actions.*

- Due to this **"instant feedback"** capability, Spark is becoming an important component in many **IoT deployments.**

# Apache Storm and Apache Flink

- **Apache Storm and Apache Flink** are other Hadoop ecosystem projects designed for distributed stream processing and are commonly deployed for IoT use cases.

- **Storm** can pull data from **Kafka** and **process** it in a near-real-time fashion, and so can **Apache Flink.**

# Lambda Architecture

- **Querying** both data in motion (streaming) and data at rest (batch processing) requires a combination of the **Hadoop ecosystem** projects discussed.

- One architecture that is currently being leveraged for this functionality is the **Lambda Architecture.**

- **Lambda** is a **data management system** that consists of *two layers* for ingesting data (Batch and Stream) and one layer for providing the combined data (Serving).

- These layers allow for the packages like Spark and MapReduce, to operate on the data independently, focusing on the key attributes for which they are designed and optimized.

- **Data** is taken from a message broker, commonly **Kafka**, and processed by each layer in parallel, and the resulting data is delivered to a **data store** where additional processing or queries can be run.

- The figure 7. 11 shows this **parallel data** flow through the **Lambda Architecture**.

- The Lambda Architecture is not limited to the **packages in the Hadoop ecosystem**, but due to its breadth and flexibility, many of the packages in the ecosystem fill the requirements of each layer nicely:

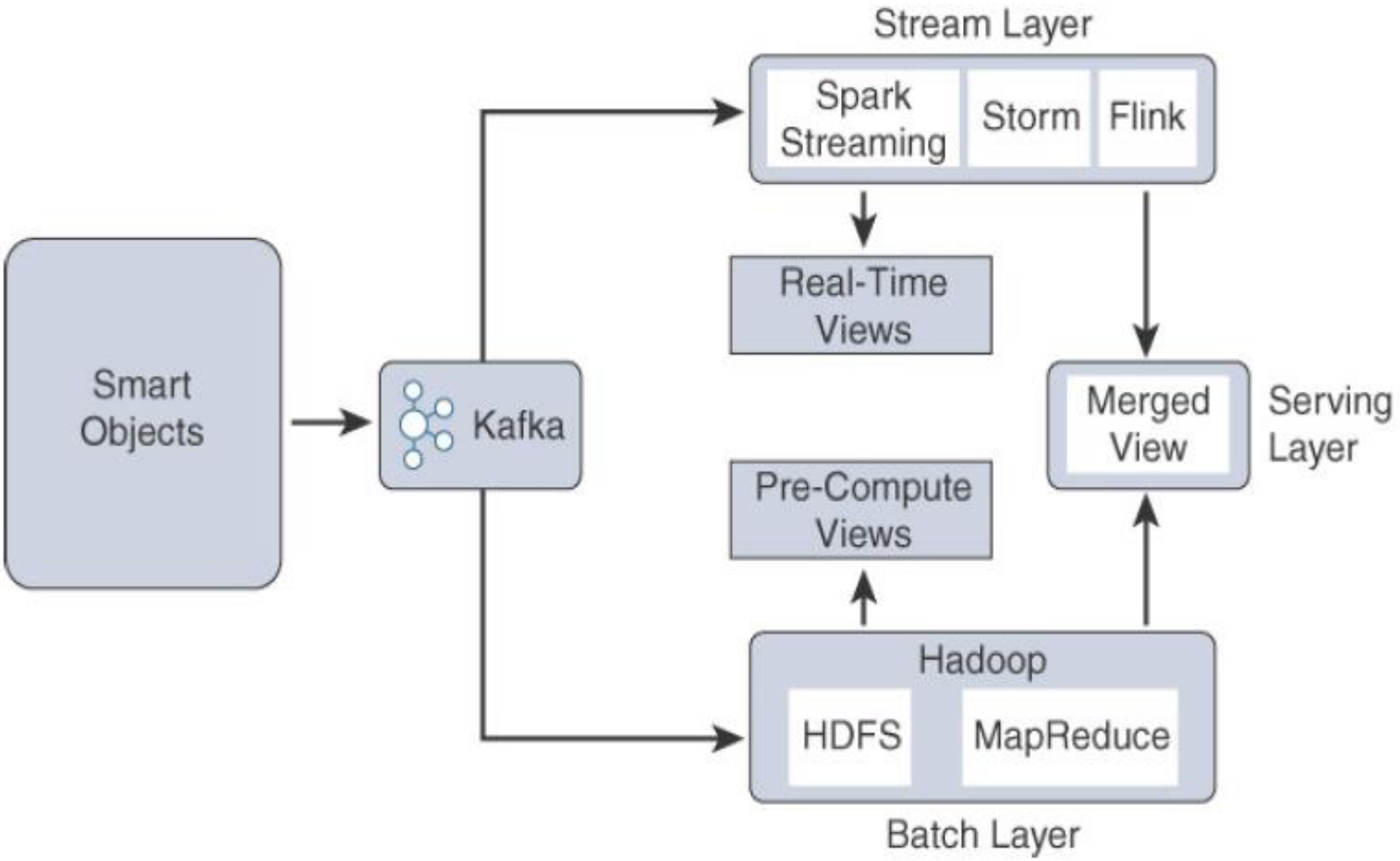


**Figure 7.11 :** Lambda Architecture

- **Stream Layer**
  - This layer is **responsible** for near-real-time processing of events.
  - Technologies such as *Spark Streaming*, *Storm*, *or Flink* are used to quickly *ingest*, *process*, *and analyze* data on this layer.
  - **Alerting and automated actions** can be triggered on **events** that require rapid response or could result in catastrophic outcomes if not handled immediately.

- **Batch Layer**
  - The Batch layer **consists** of a **batch-processing engine and data store.**
  - If an organization is using other parts of the Hadoop ecosystem for the other layers, MapReduce and HDFS can easily fit the bill.
  - Other **database technologies**, such as MPPs, NoSQL, or data warehouses, can also provide what is needed by this layer.

- **Serving Layer**
  - The Serving layer is a **data store and mediator that decides** which of the ingest layers to *query based on the expected result or view into the data.*
  - If an **aggregate or historical view** is requested, it may invoke the **batch layer.**
  - If **real-time analytics is needed**, it may invoke the Stream layer. The Serving layer is often used by the *data consumers to access both layers simultaneously.*

- The **Lambda Architecture** can provide a robust system for *collecting and processing massive amounts of data and the flexibility of being able to analyze that data at different rates.*

- One **limitation** of this type of architecture is its place in the network. Due to the processing and storage requirements of many of these pieces, the vast majority of these deployments are either in data centers or in the cloud.

# Apache Kafka

- A part of processing **real-time events** is to ingest the data generated by the smart objects into a processing engine.

- The **process** of collecting data from a sensor or log file and preparing it to be processed and analyzed is typically handled by **messaging systems.**

- **Messaging systems** are designed to *accept data, or messages*, from where the data is generated and deliver the data to *stream-processing engines* such as **Spark Streaming** or **Storm**

- **Apache Kafka** is a **distributed publisher-subscriber messaging system** that is built to be ***scalable and fast***.

- It is **composed** of topics, or message brokers, where producers write data and consumers read data from these topics.

- The figure 7.10 shows the *data flow from the smart objects (producers), through a topic in Kafka, to the real-time processing engine.*

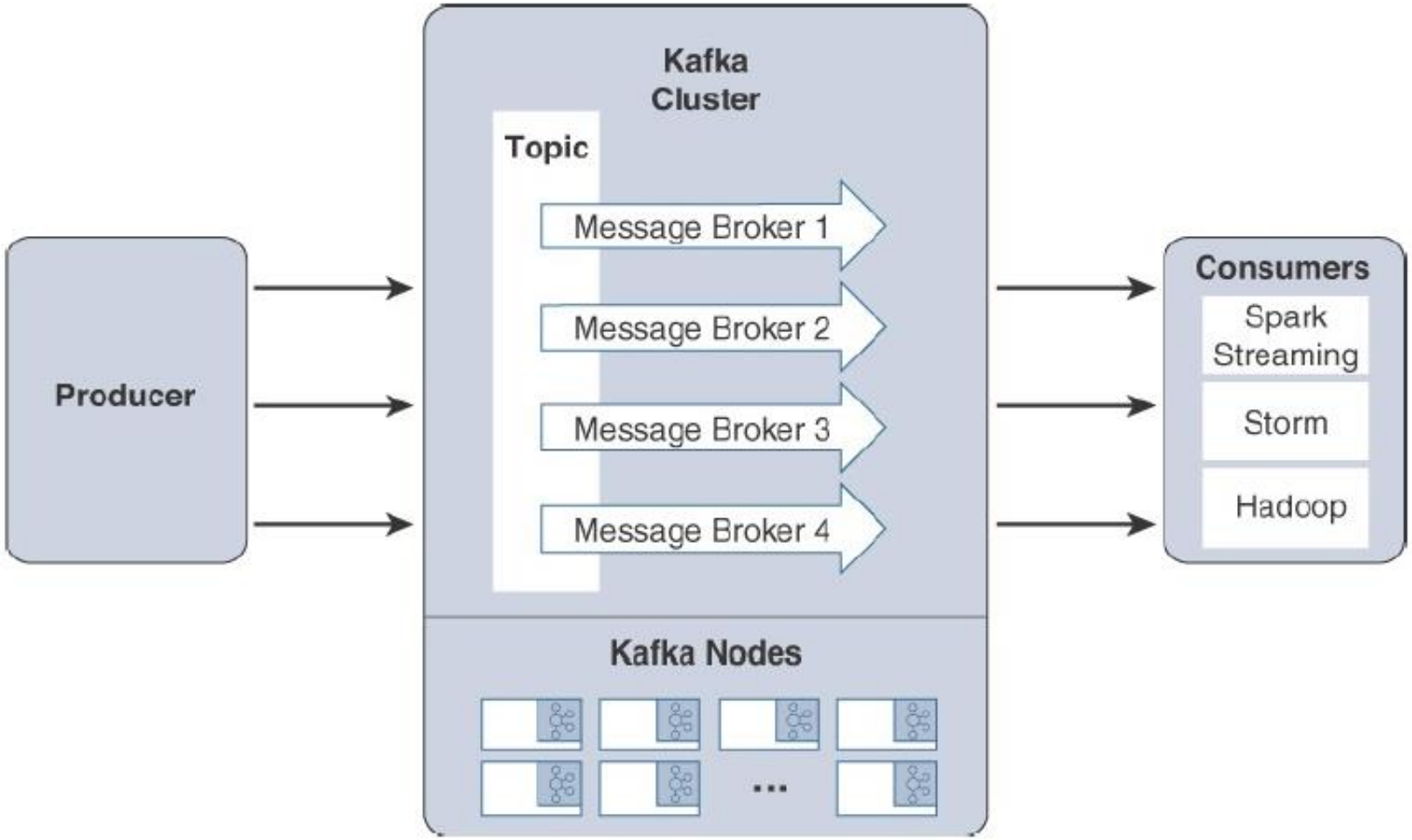


**Figure 7.10 :** Apache Kafka Data Flow

- Due to the **distributed nature of Kafka**, it can run in a **clustered configuration** that can handle *many producers and consumers simultaneously and exchanges information between nodes*, allowing topics to be distributed over multiple nodes.

- The **goal of Kafka** is to provide a *simple way to connect to data sources and allow consumers to connect to that data in the way they would like.*

# Module-4
# Chapter-8

# Securing IoT



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **A Brief History of OT Security**
- **Common Challenges in OT Security**
- **How IT and OT Security Practices and Systems Vary**
- **Formal Risk Analysis Structures: OCTAVE and FAIR**
- **The Phased Application of Security in an Operational Environment**

# A Brief History of OT Security

- In **comparison with other sectors**, cybersecurity incidents in industrial environments can result in physical consequences that can cause **threats** to human lives as well as damage to equipment, infrastructure, and the environment.

- Some of the officially reported cases for the security in the OT environment are as follows.
  - **Stuxnet** malware which damaged uranium enrichment systems in Iran.
  - An event that damaged a furnace in a German smelter.
  - Remotely accessed the sewage control system of Maroochy Shire in Queensland, Australia and released 800,000 liters of sewage into the surrounding waterways(happened in the year 2000).

- **Compounding this problem**, many of the legacy protocols used in IoT environments are many decades old, and there was no thought of **security** when they were **first developed**.

- This means **that attackers with limited or no technical capabilities** now have the potential to launch cyber attacks and pose a threat to the end operators.

- Unlike in **IT-based enterprises**, OT deployed solutions commonly have no reason to change as they are designed to meet specific (and often single-use) functions, and have no requirements or incentives to be upgraded.

- A growing trend whereby OT system vulnerabilities have been exposed and reported. This increase is depicted in the figure 8.1, which shows the history of vulnerability disclosures in industrial control systems (ICSs) since 2010.

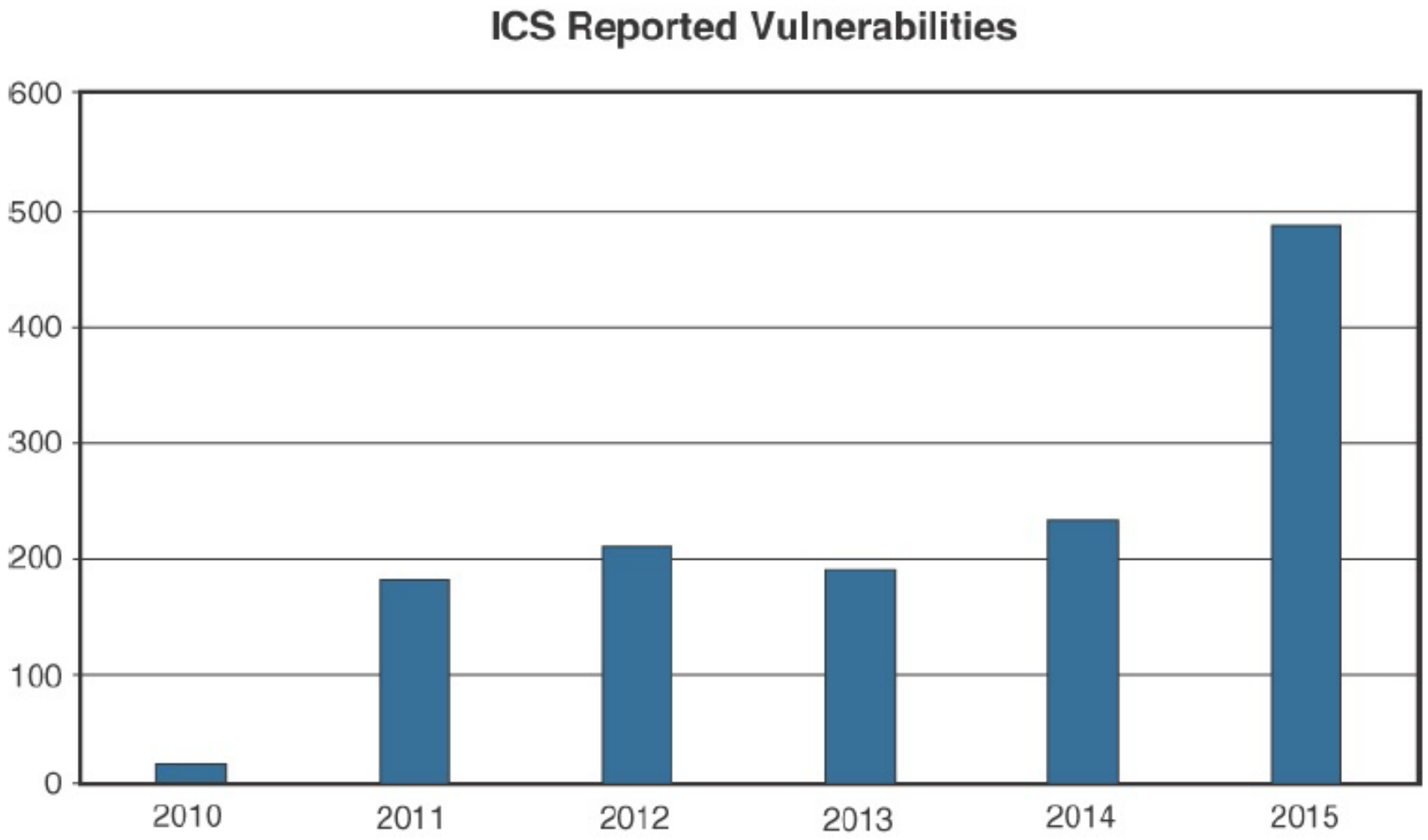


**Figure 8.1** : History of Vulnerability Disclosures in Industrial Control Systems Since 2010

# Common Challenges in OT Security

i. Erosion of Network Architecture

ii. Pervasive Legacy Systems

iii. Insecure Operational Protocols

iv. Device Insecurity

v. Dependence on External Vendors

vi. Security Knowledge

# Erosion of Network Architecture

- Two of the major challenges in securing industrial environments have been **initial design** and **ongoing maintenance**.

- The **initial design challenges** arose from the concept that networks were safe due to physical separation from the enterprise with **minimal or no connectivity** to the outside world.

- The challenge, and the biggest threat to network security, is standards and best practices either being misunderstood or the network being poorly maintained.

- It is more common that, over time, what may have been a solid design to begin with is eroded through ad hoc updates and individual changes to hardware and machinery without consideration for the broader network impact.

- This kind of **organic growth** has led to miscalculations of expanding networks and the introduction of wireless communication in a standalone fashion, without consideration of the impact to the original **security design**.

- These uncontrolled or poorly controlled OT network evolutions have, in many cases, over time led to weak or **inadequate network and systems security.**

# Pervasive Legacy Systems

- Due to the **static nature and long lifecycles** of equipment in industrial environments, many operational systems may be deemed legacy systems.

- For ex : in a power utility environment, it is not uncommon to have racks of old mechanical equipment still operating alongside modern *intelligent electronic devices (IEDs).*

- In many cases, legacy components are not restricted to isolated network segments but have now been consolidated into the IT operational environment.

- From a **security perspective**, this is potentially dangerous as many devices may have historical vulnerabilities or weaknesses that have not been patched and updated.

- Beyond the endpoints, the communication infrastructure and shared centralized compute resources are often not built to comply with modern standards.

- **Insecure Operational Protocols**

- Many industrial **control protocols**, particularly those that are serial based, were designed without inherent strong security requirements.

- Their operation was often within an assumed *secure network and their operational environment* may not have been designed with secured access control in mind.

- Industrial protocols, such as supervisory control and data acquisition **(SCADA)** particularly the older variants, suffer from common security issues.

- Three examples of this are a frequent lack of authentication between communication endpoints, **no means of securing** and **protecting data at rest or in motion**, and insufficient granularity of control to properly specify recipients or avoid default broadcast approaches.

- The **structure and operation** of most of these protocols is often publicly available.

- While they may have been originated by a private firm, for the *sake of interoperability*, they are typically published for others to implement.

- Thus, it becomes a relatively simple matter to compromise the protocols themselves and introduce malicious actors that may use them to compromise control systems.

# Device Insecurity

- Beyond the **communications protocols** that are used and the installation base of legacy systems, control and communication elements themselves have a history of vulnerabilities.

- Prior to 2010, the **security community** paid little attention to industrial compute, and as a result, OT systems have not gone through the same “trial by fire” as IT systems.

- The figure 8.2 shows this graphically by simply overlaying the count of industrial security topics presented at the Black Hat security conference with the number of vulnerabilities reported for industrial control systems.

- The **correlation** between ***presentations on the subject of OT security*** at Black Hat and the number of vulnerabilities discovered is obvious, including the associated slowing of discoveries.

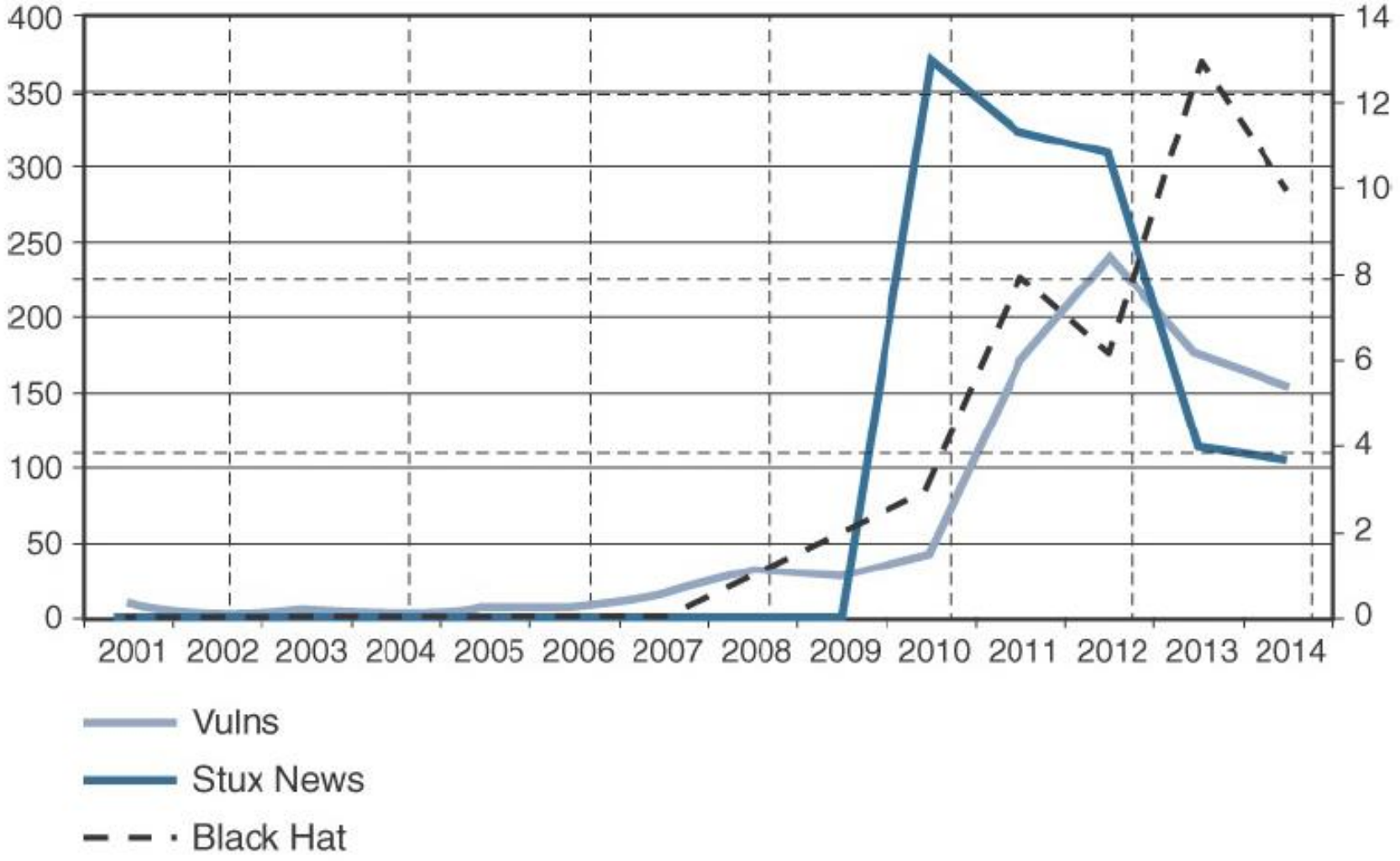


**Figure 8.2** : Correlation of Industrial Black Hat Presentations with Discovered Industrial Vulnerabilities

- Some of the **reasons why ICS(Industrial Control Systems**) are frequently found **vulnerable:**

  - **First,** many of the systems utilize software packages that can be easily downloaded and worked against.

  - **Second,** they operate on common hardware and standard operating systems, such as Microsoft Windows.

  - **Third,** Windows and the components used within those applications are well known to traditionally IT-focused security researchers.

- The ICS vendor community is also lagging behind IT counterparts with regard to security capabilities and practices, as well as cooperation with third-party security researchers.

## Dependence on External Vendors

- While modern IT environments may be outsourcing business operations or relegating certain processing or storage functions to the cloud, it is less common for the original equipment manufacturers of the IT hardware assets to be required to operate the equipment.

- **Direct and on-demand access** to critical systems on the plant floor or in the field are sometimes written directly into contracts or are required for valid product warranties.

- This has clear benefits in many industries as it allows vendors to remotely manage and monitor equipment and to proactively alert the customer if problems are beginning to creep in.

- While contracts may be written to describe equipment monitoring and management requirements with explicit statements of what type of access is required and under what conditions, they generally fail to address questions of shared liability for security breaches.

- Such **vendor dependence** and control are not limited to remote access.

- Onsite management of non-employees that are to be granted compute and network access are also required, but again, control conditions and shared responsibility statements are yet to be observed.

# Security Knowledge

- In the **industrial operations space**, the technical investment is primarily in connectivity and compute. It has seen far less investment in security relative to its IT counterpart.

- Another relevant **challenge** in terms of OT security expertise is the comparatively higher age of the industrial workforce.

- Simultaneously, **new connectivity technologies** are being introduced in OT industrial environments that require up-to-date skills, such as TCP/IP, Ethernet, and wireless that are quickly replacing serial-based legacy technologies.

- The rapid expansion of extended communications networks and the need for an industrial controls aware workforce creates an equally serious gap in security awareness.

- Due to the *importance of security in the industrial space*, all likely attack surfaces are treated as **unsafe.**

- Bringing industrial networks up to the latest and most secure levels is a slow process due to deep historical cultural and philosophical differences between OT and IT environments.

# How IT and OT Security Practices and Systems Vary

- The differences between an **enterprise IT environment** and an **industrial-focused** OT deployment are important to understand because they have a direct impact on the security practice applied to them.

- **The Purdue Model for Control Hierarchy**
- IT information is typically used to make **business decisions,** such as those in process optimization whereas OT information is instead characteristically leveraged to make **physical decisions**, such as closing a valve, increasing pressure, and so on.

- Organizationally, **IT and OT teams** and **tools** have been historically separate, but this has begun to change, and they have started to converge, leading to more traditionally IT centric solutions being introduced to **support operational activities.**

- As the **borders** between traditionally separate OT and IT domains blur, they must align strategies and work more closely together to ensure end-to-end security.

- The **Purdue Model for Control Hierarchy**, is the most widely used framework across industrial environments globally and is used in manufacturing, oil and gas, and many other industries.

- It **segments devices and equipment** by hierarchical function levels and areas as shown in the figure 8.3.

- This model identifies levels of operations and defines each level. The enterprise and operational domains are separated into different zones and kept in strict isolation via an **industrial demilitarized zone (DMZ):**


**Figure 8.3** : The Logical Framework Based on the Purdue Model for Control Hierarchy

i. **Enterprise zone**

- **Level 5: Enterprise network:**
  - Corporate-level applications such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), document management, and services such as **Internet access and VPN entry** from the outside world exist at this level.

- **Level 4: Business planning and logistics network:**
  - The **IT services exist** at this level and may include scheduling systems, material flow applications, optimization and planning systems, and local IT services such as phone, email, printing, and security monitoring.

## ii. Industrial demilitarized zone

- **DMZ**
  - The **DMZ provides a buffer zone** where **services and data** can be shared between the operational and enterprise zones.
  - It also allows for **easy segmentation of organizational control.** By default, no traffic should traverse the DMZ; everything should originate from or terminate on this area.

iii. **Operational zone**

- **Level 3: Operations and control:**
  - This **level** includes the functions involved in managing the workflows to produce the desired end products and for monitoring and controlling the **entire operational system**.
  - This could include *production scheduling, reliability assurance, systemwide control optimization, security management, network management, and potentially* other required IT services, such as DHCP, DNS, and timing.

- **Level 2: Supervisory control:**
  - This **level** includes zone control rooms, controller status, control system network/application administration, and other control-related applications, such as human-machine interface (HMI) and historian.

- **Level 1: Basic control:**
  - At this **level,** controllers and IEDs, dedicated HMIs, and other applications may talk to each other to run part or all of the control function.

- **Level 0: Process:**
  - This is where devices such as sensors and actuators and machines such as drives, motors, and robots communicate with controllers or IEDs.

iv. **Safety zone**

- **Safety-critical:**
  - This level includes devices, sensors, and other equipment used to manage the safety functions of the control system.

- One of the **key advantages** of designing an industrial network in **structured levels,** as with the Purdue model, is that it allows security to be correctly applied at each level and between levels.

- A **DMZ resides between the IT and OT** levels as shown in the figure 8.3.

- Clearly, *to protect the lower industrial layers*, security technologies such as firewalls, proxy servers, and IPSs should be used to ensure that only authorized connections from trusted sources on expected ports are being used.

- The figure 8-4, presents a review of published vulnerabilities associated with industrial security in 2011 shows that the assets at the higher levels of the framework had more detected vulnerabilities.

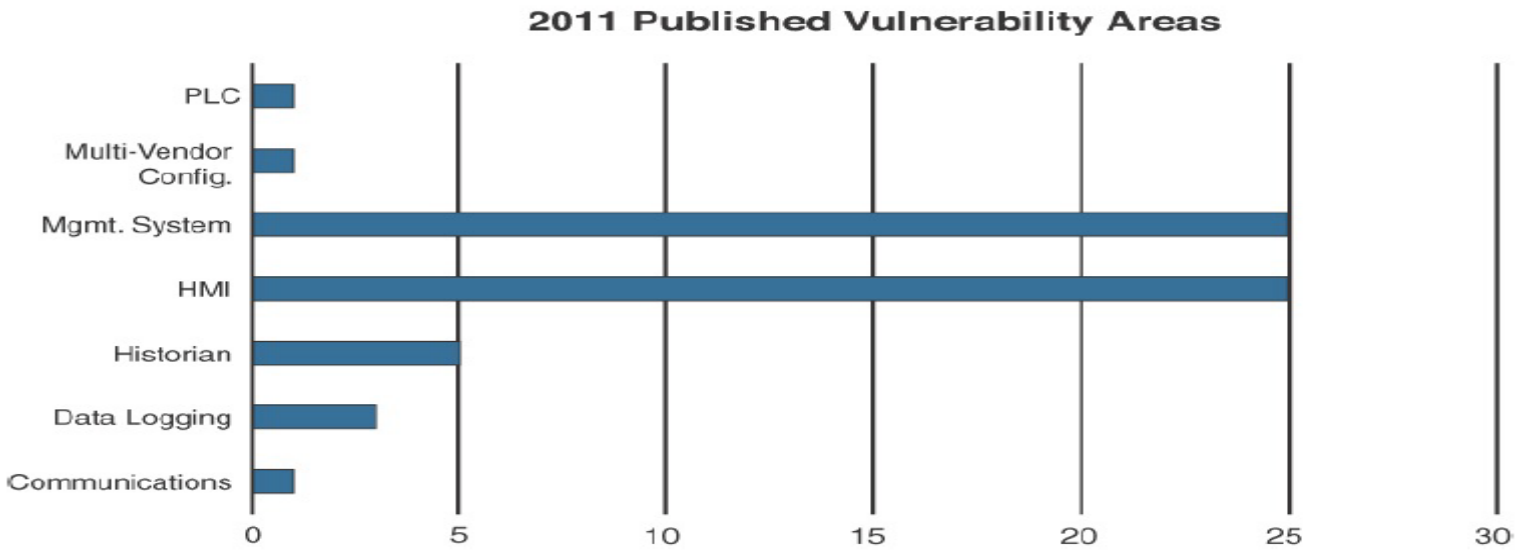


**Figure 8.4** : 2011 Industrial Security Report of Published Vulnerability Areas

## ✓ OT Network Characteristics Impacting Security

- While **IT and OT** networks are beginning to converge, they still maintain many divergent characteristics in terms of how they operate and the traffic they handle.

- These **differences influence** how they are treated in the context of a security strategy.

- For ex : compare the nature of how traffic flows across IT and OT networks-

- **IT Networks**:
  - In an **IT environment**, there are many diverse data flows. The communication data flows that emanate from a typical **IT endpoint travel relatively far.**
  - They frequently traverse the network through **layers of switches** and eventually make their way to a set of local or remote servers, which they may connect to directly.

- **Data** in the form of **email, file transfers, or print services** will likely all make its way to the central data center, where it is responded to, or triggers actions in more local services, such as a printer.

- In the case of **email or web browsing**, the endpoint initiates actions that leave the confines of the enterprise network and potentially travel around the earth.

- **OT networks**
  - By **comparison**, in an OT environment (Levels 0–3), there are typically **two types of operational traffic.**
  - The **first is local traffic** that may be contained within a specific package or area to provide local monitoring and closed-loop control.

- The **second type of traffic** is used for monitoring and control of areas or zones or the overall system.

- **SCADA traffic** is a good example of this, where information about remote devices or summary information from a function is shared at a system level so that operators can understand how the overall system, or parts of it, are operating.

- When **IT endpoints communicate**, it is typically short and frequent conversations with many connections.

- The **nature of the communications** is open, and almost anybody can speak with anybody else, such as with email or browsing.

- Although there are clearly access controls, most of those controls are at the **application level** rather than the **network level.**

## ✓ Security Priorities: Integrity, Availability, and Confidentiality

- **Security priorities** are driven by the nature of the assets in each environment.
- In the **IT business world**, there are legal, regulatory, and commercial obligations to protect data, especially data of individuals who may or may not be employed by the organization.
- **Security priorities** or preferences are diverge based on those differences.

- This emphasis on privacy focuses on the **confidentiality**, **integrity**, and **availability** of the data—not necessarily on a system or a physical asset.

- The impact of losing a compute device is considered minimal compared to the information that it could hold or provide access to.

- In an **operational space**, the safety and continuity of the process participants is considered the most **critical concern.**

## ✓ Security Focus

- **Security focus** is frequently driven by the history of security impacts that an organization has experienced.

- In an **IT environment**, the most painful experiences have typically been intrusion campaigns in which critical **data is extracted or corrupted.**

- In the **OT space,** the history of loss due to external actors has not been as long, even though the potential for harm on a human scale is clearly significantly higher.

- The **result is** that the security events that have been experienced have come more from human error than external attacks.

- **Interest and investment** in industrial security have primarily been in the standard access control layers.

# Formal Risk Analysis Structures: OCTAVE and FAIR

- The **key** for any **industrial environment** is that it needs to address security holistically and not just **focus on technology**.

- It must include **people and processes**, and it should include all the vendor ecosystem components that make up a control system.

- Let us now review two such risk assessment frameworks:
  i. **OCTAVE** (Operationally Critical Threat, Asset and Vulnerability Evaluation) from the Software Engineering Institute at Carnegie Mellon University

  ii. **FAIR** (Factor Analysis of Information Risk) from The Open Group

# OCTAVE

- **OCTAVE** has undergone multiple iterations. Let us focus on OCTAVE Allegro, which is intended to be a lightweight and less burdensome process to implement.

- Allegro assumes that a **robust security** team is not on standby or immediately at the ready to initiate a comprehensive **security review.**

- The figure 8.5 illustrates the **OCTAVE Allegro steps and phases.**

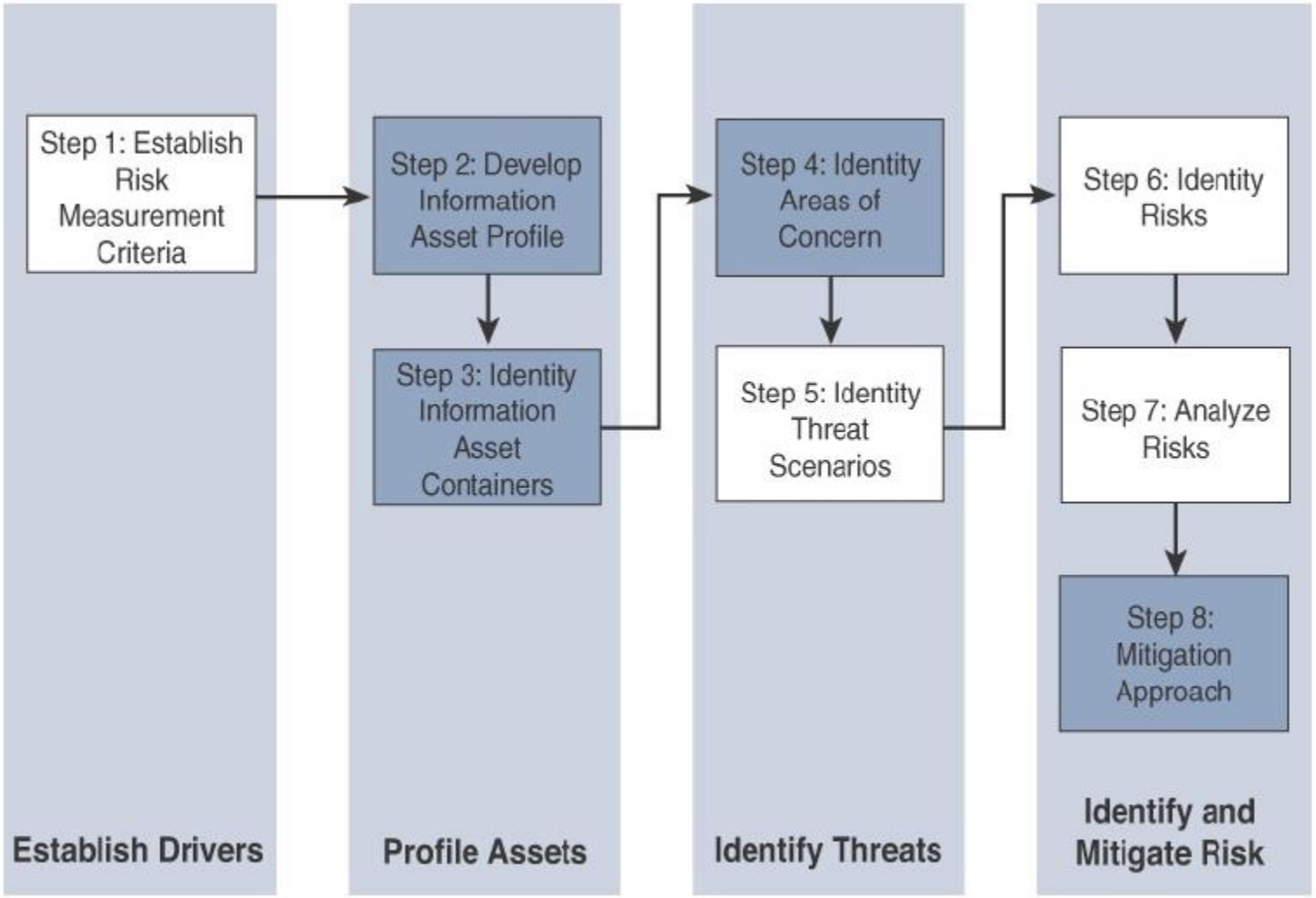


**Figure 8.5** : OCTAVE Allegro Steps and Phases

- **The first step of the OCTAVE Allegro methodology is to establish a risk measurement criterion.**

  - **OCTAVE** provides a fairly simple means of doing this with an emphasis on impact, value, and measurement.

  - The point of having a **risk measurement criterion** is that at any point in the later stages, prioritization can take place against the reference model.

- **The second step is to develop an information asset profile.**
  - This profile is populated with *assets, a prioritization of assets, attributes associated with each asset, including owners, custodians, people, explicit security requirements, and technology assets.*
  - It is **important to stress the importance of process**. Within this asset profile, process are multiple substages that complete the definition of the assets.

- **The third step is to identify information asset containers.**
  - This is the range of transports and possible locations where the information might reside.
  - This **references** the *compute elements and the networks by which they communicate.*
  - However, it can also mean physical manifestations such as hard copy documents or even **the people who know the information.**

- **The fourth step is to identify areas of concern.**
  - At this point, we depart from a data flow, touch, and attribute focus to one where judgments are made through a mapping of security-related attributes to more business-focused use cases.
  - At this stage, the analyst looks to risk profiles and delves into the previously mentioned risk analysis.
  - *History both within and outside the organization can contribute.* References to similar operational use cases and incidents of security failures are reasonable associations.

- **In the closely related fifth step, threat scenarios are identified.**
  - **Threats** are broadly (and properly) identified as potential undesirable events.
  - This definition means that results from both malevolent and accidental causes are viable threats. In the context of operational focus, this is a valuable consideration.
  - It is at this point that an *explicit identification of actors, motives, and outcomes occurs.*

- **At the sixth step risks are identified.**

  - Within **OCTAVE**, risk is the possibility of an undesired outcome. This is extended to focus on how the organization is impacted.

  - For more **focused analysis**, this can be localized, but the potential impact to the organization could extend outside the boundaries of the operation.

- **The seventh step is risk analysis, with the effort placed on qualitative evaluation of the impacts of the risk.**
  - Here the risk measurement criteria defined in the first step are explicitly brought into the process

- **Finally, mitigation is applied at the eighth step.**
  - There are three outputs or decisions to be taken at this stage.
    - **One** may be to **accept a risk and do nothing**, other than document the situation, potential outcomes, and reasons for accepting the risk.
    - The **second** is to mitigate the risk with whatever control effort is required. By walking back through the threat scenarios to asset profiles, a pairing of compensating controls to mitigate those threat/risk pairings should be discoverable and then implemented.
    - The **final** possible action is to defer a decision, meaning risk is neither accepted nor mitigated.

# FAIR

- **FAIR (Factor Analysis of Information Risk)** is a technical standard for risk definition from The Open Group.
- While information security is the focus, much as it is for OCTAVE, **FAIR** has clear applications within operational technology.
- It also **allows** for non-malicious actors as a potential cause for harm, but it goes to greater lengths to emphasize the point

- FAIR places emphasis on both unambiguous definitions and the idea that risk and associated attributes are measurable.

- Measurable, quantifiable metrics are a key area of emphasis, which should lend itself well to an operational world with a richness of operational data.

- At its base, FAIR has a definition of risk as the probable **frequency and probable magnitude of loss.**

- With this definition, a clear hierarchy of sub-elements emerges, with one side of the taxonomy focused on **frequency** and the other on **magnitude**.

- **Loss even frequency** is the result of a threat agent acting on an asset with a resulting loss to the organization.

- This happens with a given frequency called the **threat event frequency (TEF)**, in which a specified time window becomes a probability.

- There are multiple sub-attributes that define **frequency of events,** all of which can be understood with some form of measurable metric. Threat event frequencies are applied to a vulnerability.

- **Vulnerability** here is not necessarily some compute asset weakness, but is more broadly defined as the probability that the targeted asset will fail as a result of the **actions** applied.

- The other side of the risk taxonomy is the **probable loss magnitude (PLM)**, which begins to quantify the impacts, with the emphasis again being on measurable metrics.

- **FAIR defines** six forms of loss, four of them externally focused and two internally focused. Of particular value for operational teams are **productivity** and **replacement loss**.

- **Response loss** is also reasonably measured, with fines and judgments easy to measure but difficult to predict.

# The Phased Application of Security in an Operational Environment

- It is a **security practitioner's goal** to safely secure the environment for which he or she is responsible.

- For an **operational technologist**, this process is different because the priorities and assets to be protected are highly differentiated from the better-known IT environment.

- Let us now look into a **phased approach** to introduce **modern network sec**urity into largely pre-existing legacy industrial networks.

## ✓ Secured Network Infrastructure and Assets

- In a typical IoT or industrial system, the networks, compute, or operational elements would have been in place for many years and given that the physical layout largely **defines** the operational process, this phased approach to introducing modern network security begins with very modest, non-intrusive steps.

- As a **first step,** we need to analyze and secure the basic network design.

- Most automated process systems or even hierarchical energy distribution systems have a **high degree** of correlation between the **network design and the operational design.**

- The figure 8.6 illustrates **inter-level security models and inter-zone** conduits in the process control hierarchy.

- Normal network discovery processes can be highly problematic for **older networking** equipment.

- Given that **condition**, the network discovery process may require manual inspection of physical connections, starting from the highest accessible aggregation point to the last access layer.

- This discovery activity must include a search for wireless access points.

- This kind of **search activity** will yield good results in a small confined environment such as a plant floor.

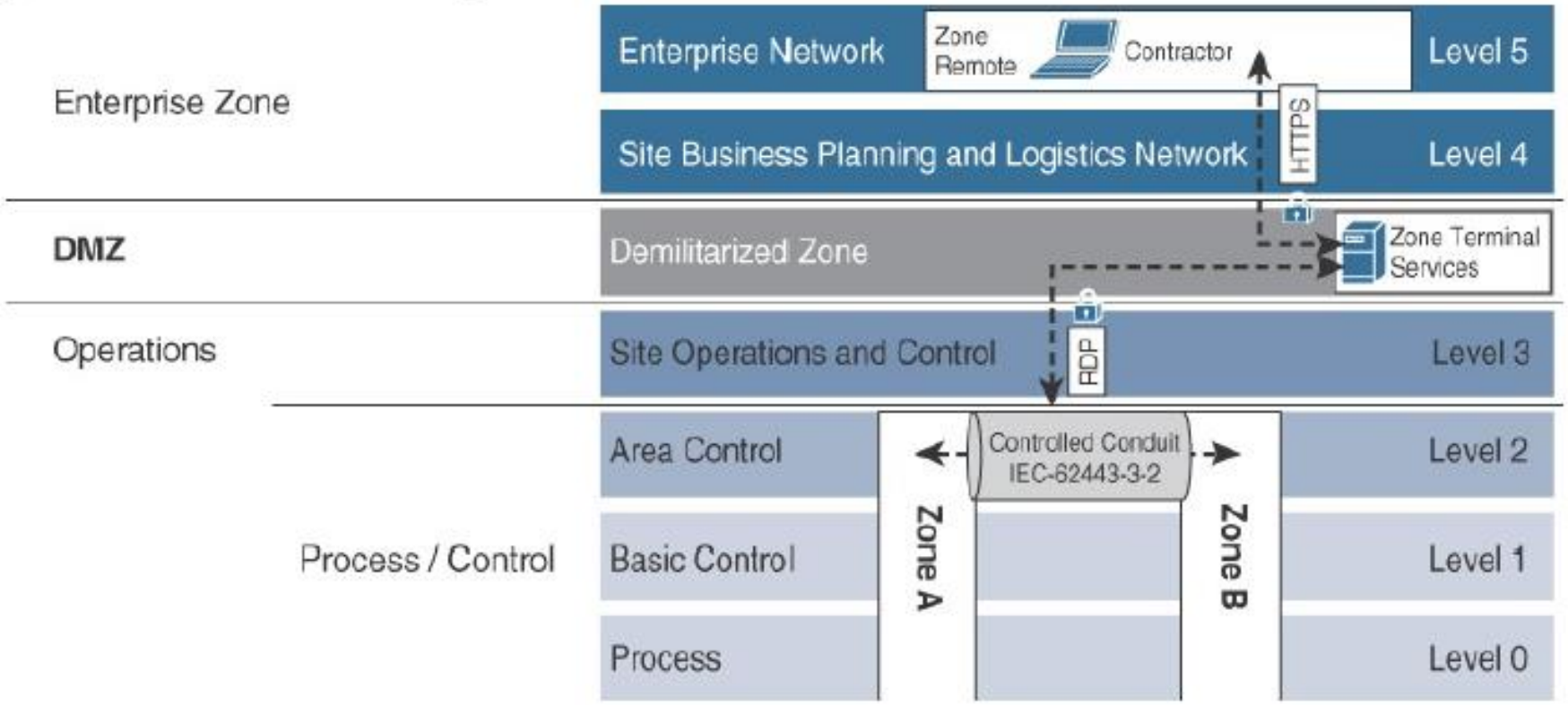


**Figure 8.6** : Security Between Levels and Zones in the Process Control Hierarchy Model

- **Modern networking** equipment offers a rich set of access control and secured communications capabilities.

- Starting at the cell/zone level, it is important to ensure that there is a clear ingress/egress aggregation point for each zone.

- If our **communications patterns are well identified**, we can apply access control policies to manage who and what can enter those physical portions of the process.

- At **upstream levels**, consider traffic controls such as denial of service (DoS) protection, traffic normalization activities, and quality of service (QoS) controls.

- **Network infrastructure** should also provide the ability to secure communications between zones via secured conduits.

- The **next discovery phase** should align with the software and configurations of the assets on the network.

- The next stage is to expand the security footprint with focused security functionality. The goal is to provide visibility, safety, and security for traffic within the network.

- **Visibility provides** an understanding of application and communication behavior.

- The **network elements** can provide *simplified views with connection histories or some kind of flow data and we get a true understanding when we look within the packets on the network.*

- This level of visibility is typically achieved with **deep packet inspection** (DPI) technologies such as intrusion **detection/prevention systems (IDS/IPS).**

## ✓ Deploying Dedicated Security Appliances

- These **technologies** can be used to detect many kinds of traffic of interest, from simply identifying what applications are speaking, to whether communications are being obfuscated or whether exploits are targeting vulnerabilities.

- With the goal of identifying assets, an IDS/IPS can detect what kind of assets are present on the network.

- **Application-specific protocols** are also detectable by IDS/IPS systems. For more IT-like applications, **user agents are of value,** but traditionally, combinations of port numbers and other protocol differentiators can contribute to identification.

- **Visibility and an understanding** of network connectivity uncover the information necessary to initiate access control activity.

- **Access control** is typically achieved with access control lists (ACLs), which are available on practically all modern network equipment.

- For **improved scalability**, however, a dedicated firewall would be preferred. Providing strong segmentation and zone access control is an essential step.

- **Safety** is a particular benefit as application controls can be managed at the **cell/zone edge through an IDS/IPS.**

- **Placement priorities** for dedicated security devices vary according to the security practitioner's perception of risk.
- **Placement** at the operational cell is likely the most fine-grained deployment scenario.
- By **fine-grained** we mean that it is the lowest portion of a network that gives network-based access to the lowest level of operational assets.

## ✓ Higher-Order Policy Convergence and Network Monitoring

- A **security practice** that adds value to a networked industrial space is convergence, which is the adoption and integration of security across operational boundaries.

- This means coordinating security on both the IT and OT sides of the organization.

- Convergence of the **IT and OT** spaces is merging, or at least there **is active coordination** across formerly distinct IT and OT boundaries.

- From a **security perspective**, the value follows the argument that most new **networking and compute technologies** coming to the operations space were previously found and established in the IT space.

- Several areas are more likely to require some kind of coordination across IT and OT environments. Two such areas are **remote access** and **threat detection**.

- For remote access, most large industrial organizations rely upon **backhaul communication** through the IT network.

- There are advanced enterprise-wide practices related to access control, threat detection, and many other security mechanisms that could benefit OT security.

- Using *location information, participant device security stance, user identity, and access target attributes are all standard functions* that modern access policy tools can make use of.

- **Network security monitoring (NSM)** is a process of **finding intruders in a network.**

- It is achieved by collecting and analyzing indicators and **warnings** to prioritize and investigate incidents with the assumption that there is, in fact, an **undesired presence.**

- The practice of NSM is not new, yet it is not implemented often or thoroughly enough even within **reasonably mature and large organizations.**

- There are many reasons for this underutilization, but lack of education and organizational patience are common reasons.

- To simplify the approach, there is a large amount of readily available data that, if reviewed, would expose the activities of an intruder.

# Module-4
# Chapter-8

# Securing IoT



Rukmini B, Dept. of CSE,SMVITM

- This chapter explores the following topics

- **A Brief History of OT Security**
- **Common Challenges in OT Security**
- **How IT and OT Security Practices and Systems Vary**
- **Formal Risk Analysis Structures: OCTAVE and FAIR**
- **The Phased Application of Security in an Operational Environment**

# Formal Risk Analysis Structures: OCTAVE and FAIR

- The **key** for any **industrial environment** is that it needs to address security holistically and not just **focus on technology**.

- It must include **people and processes**, and it should include all the vendor ecosystem components that make up a control system.

- Let us now review two such risk assessment frameworks:
  i. **OCTAVE** (Operationally Critical Threat, Asset and Vulnerability Evaluation) from the Software Engineering Institute at Carnegie Mellon University

  ii. **FAIR** (Factor Analysis of Information Risk) from The Open Group

# OCTAVE

- **OCTAVE** has undergone multiple iterations. Let us focus on OCTAVE Allegro, which is intended to be a lightweight and less burdensome process to implement.

- Allegro assumes that a **robust security** team is not on standby or immediately at the ready to initiate a comprehensive **security review.**

- The figure 8.5 illustrates the **OCTAVE Allegro steps and phases.**

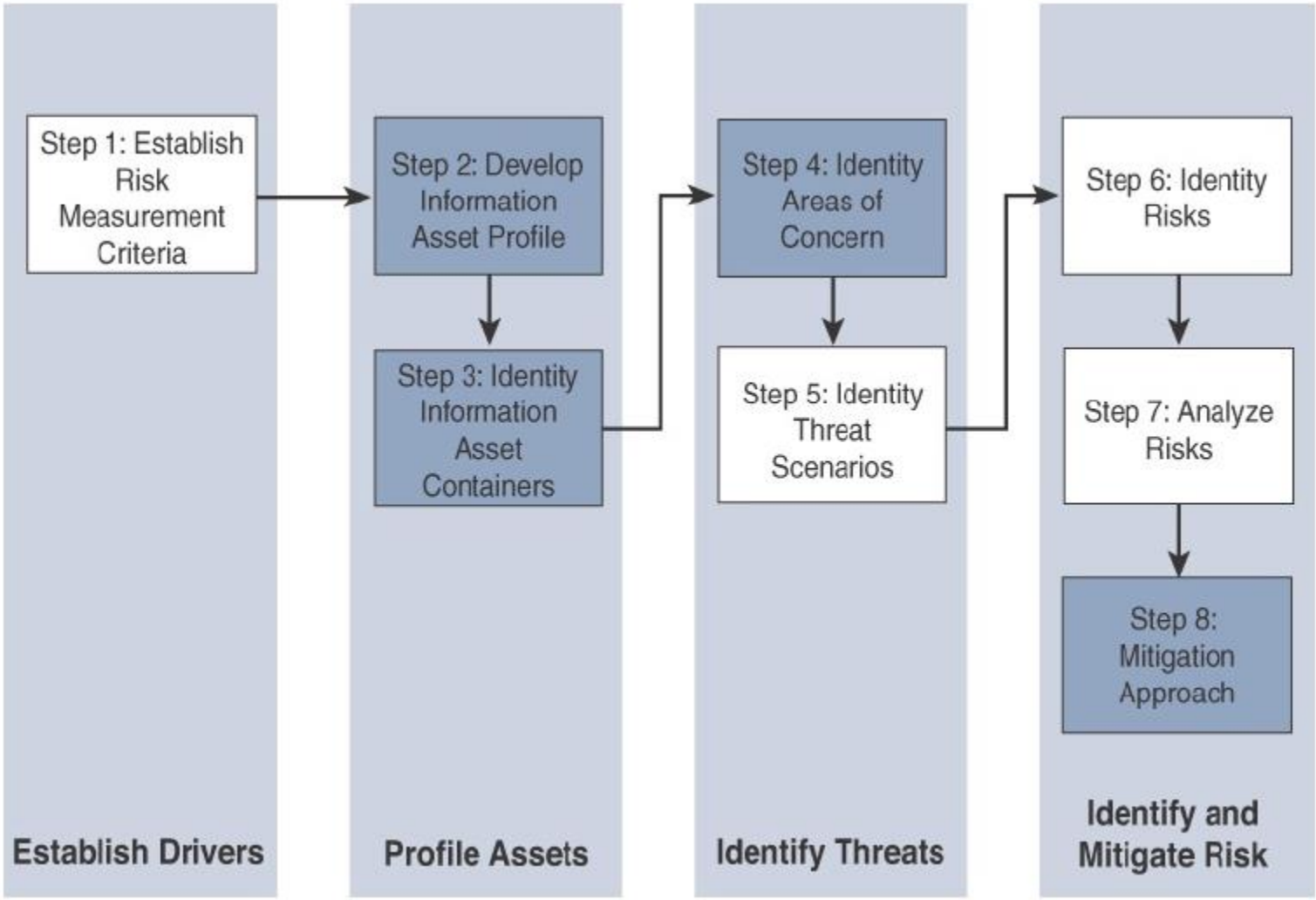


**Figure 8.5** : OCTAVE Allegro Steps and Phases

- **The first step of the OCTAVE Allegro methodology is to establish a risk measurement criterion.**

  - **OCTAVE** provides a fairly simple means of doing this with an emphasis on impact, value, and measurement.

  - The point of having a **risk measurement criterion** is that at any point in the later stages, prioritization can take place against the reference model.

- **The second step is to develop an information asset profile.**
  - This profile is populated with *assets, a prioritization of assets, attributes associated with each asset, including owners, custodians, people, explicit security requirements, and technology assets.*
  - It is **important to stress the importance of process**. Within this asset profile, process are multiple substages that complete the definition of the assets.

- **The third step is to identify information asset containers.**
  - This is the range of transports and possible locations where the information might reside.
  - This **references** the *compute elements and the networks by which they communicate.*
  - However, it can also mean physical manifestations such as hard copy documents or even **the people who know the information.**

- **The fourth step is to identify areas of concern.**
  - At this point, we depart from a data flow, touch, and attribute focus to one where judgments are made through a mapping of security-related attributes to more business-focused use cases.
  - At this stage, the analyst looks to risk profiles and delves into the previously mentioned risk analysis.
  - *History both within and outside the organization can contribute.* References to similar operational use cases and incidents of security failures are reasonable associations.

- **In the closely related fifth step, threat scenarios are identified.**
  - **Threats** are broadly (and properly) identified as potential undesirable events.
  - This definition means that results from both malevolent and accidental causes are viable threats. In the context of operational focus, this is a valuable consideration.
  - It is at this point that an *explicit identification of actors, motives, and outcomes occurs.*

- **At the sixth step risks are identified.**

  - Within **OCTAVE**, risk is the possibility of an undesired outcome. This is extended to focus on how the organization is impacted.

  - For more **focused analysis**, this can be localized, but the potential impact to the organization could extend outside the boundaries of the operation.

- **The seventh step is risk analysis, with the effort placed on qualitative evaluation of the impacts of the risk.**
  - Here the risk measurement criteria defined in the first step are explicitly brought into the process

- **Finally, mitigation is applied at the eighth step.**
  - There are three outputs or decisions to be taken at this stage.
    - **One** may be to **accept a risk and do nothing**, other than document the situation, potential outcomes, and reasons for accepting the risk.
    - The **second** is to mitigate the risk with whatever control effort is required. By walking back through the threat scenarios to asset profiles, a pairing of compensating controls to mitigate those threat/risk pairings should be discoverable and then implemented.
    - The **final** possible action is to defer a decision, meaning risk is neither accepted nor mitigated.

# FAIR

- **FAIR (Factor Analysis of Information Risk)** is a technical standard for risk definition from The Open Group.
- While information security is the focus, much as it is for OCTAVE, **FAIR** has clear applications within operational technology.
- It also **allows** for non-malicious actors as a potential cause for harm, but it goes to greater lengths to emphasize the point

- FAIR places emphasis on both unambiguous definitions and the idea that risk and associated attributes are measurable.

- Measurable, quantifiable metrics are a key area of emphasis, which should lend itself well to an operational world with a richness of operational data.

- At its base, FAIR has a definition of risk as the probable **frequency and probable magnitude of loss.**

- With this definition, a clear hierarchy of sub-elements emerges, with one side of the taxonomy focused on **frequency** and the other on **magnitude**.

- **Loss even frequency** is the result of a threat agent acting on an asset with a resulting loss to the organization.

- This happens with a given frequency called the **threat event frequency (TEF)**, in which a specified time window becomes a probability.

- There are multiple sub-attributes that define **frequency of events,** all of which can be understood with some form of measurable metric. Threat event frequencies are applied to a vulnerability.

- **Vulnerability** here is not necessarily some compute asset weakness, but is more broadly defined as the probability that the targeted asset will fail as a result of the **actions** applied.

- The other side of the risk taxonomy is the **probable loss magnitude (PLM)**, which begins to quantify the impacts, with the emphasis again being on measurable metrics.

- **FAIR defines** six forms of loss, four of them externally focused and two internally focused. Of particular value for operational teams are **productivity** and **replacement loss**.

- **Response loss** is also reasonably measured, with fines and judgments easy to measure but difficult to predict.

# The Phased Application of Security in an Operational Environment

- It is a **security practitioner's goal** to safely secure the environment for which he or she is responsible.

- For an **operational technologist**, this process is different because the priorities and assets to be protected are highly differentiated from the better-known IT environment.

- Let us now look into a **phased approach** to introduce **modern network sec**urity into largely pre-existing legacy industrial networks.

## ✓ Secured Network Infrastructure and Assets

- In a typical IoT or industrial system, the networks, compute, or operational elements would have been in place for many years and given that the physical layout largely **defines** the operational process, this phased approach to introducing modern network security begins with very modest, non-intrusive steps.

- As a **first step,** we need to analyze and secure the basic network design.

- Most automated process systems or even hierarchical energy distribution systems have a **high degree** of correlation between the **network design and the operational design.**

- The figure 8.6 illustrates **inter-level security models and inter-zone** conduits in the process control hierarchy.

- Normal network discovery processes can be highly problematic for **older networking** equipment.

- Given that **condition**, the network discovery process may require manual inspection of physical connections, starting from the highest accessible aggregation point to the last access layer.

- This discovery activity must include a search for wireless access points.

- This kind of **search activity** will yield good results in a small confined environment such as a plant floor.

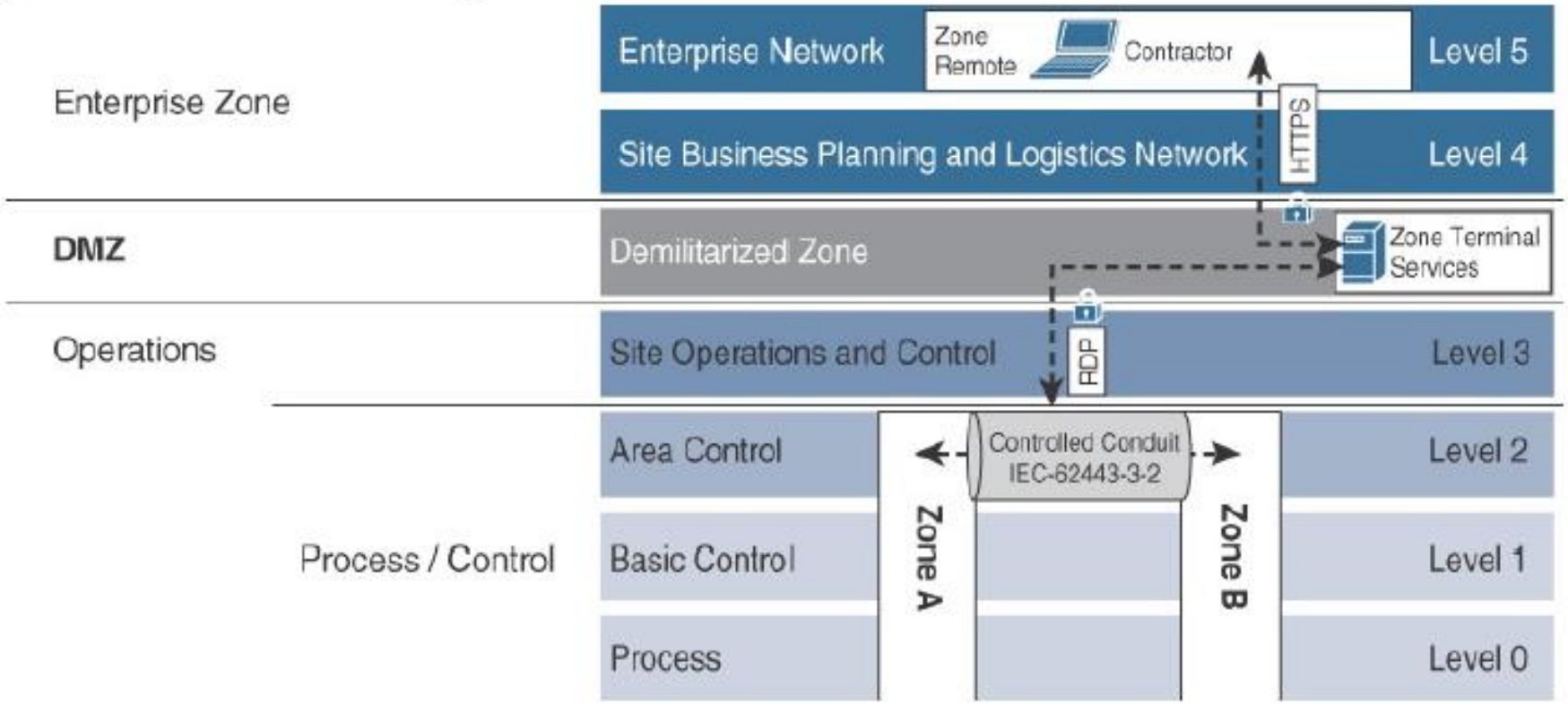


**Figure 8.6** : Security Between Levels and Zones in the Process Control Hierarchy Model

- **Modern networking** equipment offers a rich set of access control and secured communications capabilities.

- Starting at the cell/zone level, it is important to ensure that there is a clear ingress/egress aggregation point for each zone.

- If our **communications patterns are well identified**, we can apply access control policies to manage who and what can enter those physical portions of the process.

- At **upstream levels**, consider traffic controls such as denial of service (DoS) protection, traffic normalization activities, and quality of service (QoS) controls.

- **Network infrastructure** should also provide the ability to secure communications between zones via secured conduits.

- The **next discovery phase** should align with the software and configurations of the assets on the network.

- The next stage is to expand the security footprint with focused security functionality. The goal is to provide visibility, safety, and security for traffic within the network.

- **Visibility provides** an understanding of application and communication behavior.

- The **network elements** can provide *simplified views with connection histories or some kind of flow data and we get a true understanding when we look within the packets on the network.*

- This level of visibility is typically achieved with **deep packet inspection** (DPI) technologies such as intrusion **detection/prevention systems (IDS/IPS).**

## ✓ Deploying Dedicated Security Appliances

- These **technologies** can be used to detect many kinds of traffic of interest, from simply identifying what applications are speaking, to whether communications are being obfuscated or whether exploits are targeting vulnerabilities.

- With the goal of identifying assets, an IDS/IPS can detect what kind of assets are present on the network.

- **Application-specific protocols** are also detectable by IDS/IPS systems. For more IT-like applications, **user agents are of value,** but traditionally, combinations of port numbers and other protocol differentiators can contribute to identification.

- **Visibility and an understanding** of network connectivity uncover the information necessary to initiate access control activity.

- **Access control** is typically achieved with access control lists (ACLs), which are available on practically all modern network equipment.

- For **improved scalability**, however, a dedicated firewall would be preferred. Providing strong segmentation and zone access control is an essential step.

- **Safety** is a particular benefit as application controls can be managed at the **cell/zone edge through an IDS/IPS.**

- **Placement priorities** for dedicated security devices vary according to the security practitioner's perception of risk.
- **Placement** at the operational cell is likely the most fine-grained deployment scenario.
- By **fine-grained** we mean that it is the lowest portion of a network that gives network-based access to the lowest level of operational assets.

## ✓ Higher-Order Policy Convergence and Network Monitoring

- A **security practice** that adds value to a networked industrial space is convergence, which is the adoption and integration of security across operational boundaries.

- This means coordinating security on both the IT and OT sides of the organization.

- Convergence of the **IT and OT** spaces is merging, or at least there **is active coordination** across formerly distinct IT and OT boundaries.

- From a **security perspective**, the value follows the argument that most new **networking and compute technologies** coming to the operations space were previously found and established in the IT space.

- Several areas are more likely to require some kind of coordination across IT and OT environments. Two such areas are **remote access** and **threat detection**.

- For remote access, most large industrial organizations rely upon **backhaul communication** through the IT network.

- There are advanced enterprise-wide practices related to access control, threat detection, and many other security mechanisms that could benefit OT security.

- Using *location information, participant device security stance, user identity, and access target attributes are all standard functions* that modern access policy tools can make use of.

- **Network security monitoring (NSM)** is a process of **finding intruders in a network.**

- It is achieved by collecting and analyzing indicators and **warnings** to prioritize and investigate incidents with the assumption that there is, in fact, an **undesired presence.**

- The practice of NSM is not new, yet it is not implemented often or thoroughly enough even within **reasonably mature and large organizations.**

- There are many reasons for this underutilization, but lack of education and organizational patience are common reasons.

- To simplify the approach, there is a large amount of readily available data that, if reviewed, would expose the activities of an intruder.