FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

A Small Contribution Would Support Us.

Dear Viewer,

Future Vision BIE is a free service and so that any Student/Research Personal **Can Access Free of Cost**.

If you would like to say **thanks**, you can make a **small contribution** to the author of this site.

Contribute whatever you feel this is worth to you. This gives **us support** & to bring **Latest Study Material** to you. After the Contribution Fill out this Form (https://forms.gle/tw3T3bUVpLXL8omX7). To Receive a **Paid E-Course for Free**, from our End within 7 Working Days.

Regards

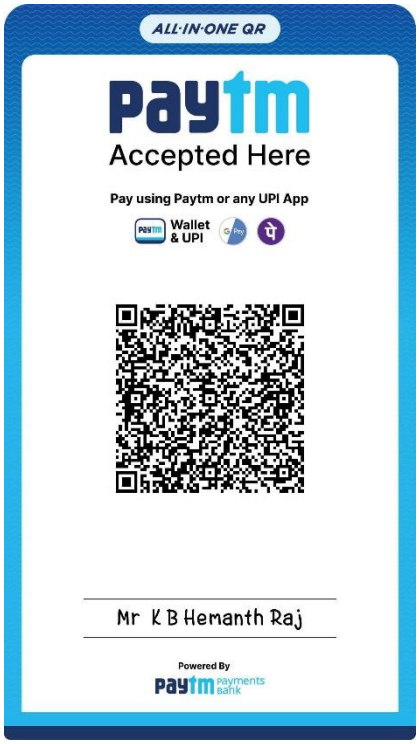
- K B Hemanth Raj (Admin)

Contribution Methods

UPI ID

1. futurevisionbie@oksbi
2. futurevisionbie@paytm

Scan & Pay



Account Transfer

Account Holder's Name: K B Hemanth Raj

Account Number: 39979402438

IFSC Code: SBIN0003982

MICR Code: 560002017

More Info: https://hemanthrajhemu.github.io/Contribution/

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering & MORE...

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE

Module – 4 Data and Analytics for IoT

Topics covered:

An Introduction to Data Analytics for IoT:

- **This section introduces the subject of analytics for IoT and discusses the differences between structured and unstructured data. It also discusses how analytics relates to IoT data.**

Machine Learning:

- **This section delves into the major types of machine learning that are used to gain business insights from IoT data.**

Module – 4 Data and Analytics for IoT

Topics covered:

Big Data Analytics Tools and Technology: Big data is one of the most:

- **This section examines some of the most common technologies used in big data today, including Hadoop, NoSQL, MapReduce, and MPP.**

Edge Streaming Analytics:

- **IoT requires that data be processed and analyzed as close to the endpoint as possible, in real-time. This section explores how streaming analytics can be used for such processing and analysis.**

Module – 4 Data and Analytics for IoT

Topics covered:

Network Analytics:

- **The final section of this chapter investigates the concept of network flow analytics using Flexible NetFlow in IoT systems.**
- **NetFlow can help you better understand the function of the overall system and heighten security in an IoT network.**

Module – 4 Data and Analytics for IoT

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.
- A great example of the 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.

Module – 4 Data and Analytics for IoT

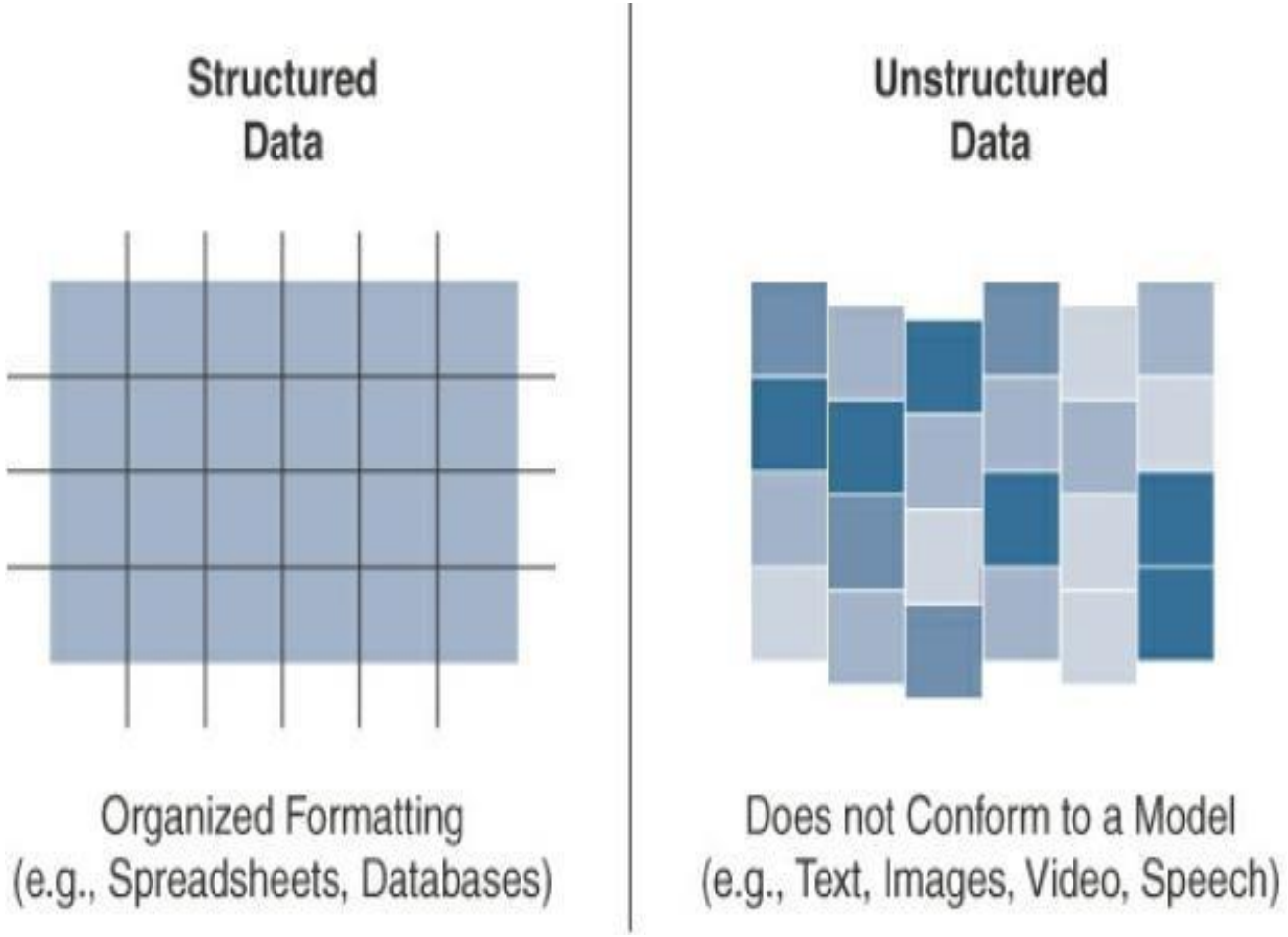
An Introduction to Data Analytics for IoT:

- **Modern jet engines are fitted with thousands of sensors that generate a whopping 10GB of data per second.**
- **modern jet engines, may be equipped with around 5000 sensors.**
- **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!**
- **Aircraft today have thousands of other sensors connected to the airframe and other systems. A single wing of a modern jumbo jet is equipped with 10,000 sensors.**

Module – 4 Data and Analytics for IoT

Structured Versus Unstructured Data:

- **Structured data and unstructured data are important classifications as they typically require different toolsets from a data analytics perspective.**
- **Figure provides a high-level comparison of structured data and unstructured data.**



Module – 4 Data and Analytics for IoT

Structured Versus 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).
- In many cases we will find structured data in a simple tabular form—for example, a spreadsheet where data occupies a specific cell and can be explicitly defined and referenced.
- IoT sensor data often uses structured values, such as temperature, pressure, humidity, and so on, which are all sent in a known format.

Structured Versus Unstructured Data:

- **Structured data is easily formatted, stored, queried, and processed.**
- **From custom scripts to commercial software like Microsoft Excel and Tableau, most people are familiar and comfortable with working with structured data.**

Module – 4 Data and Analytics for IoT

Structured Versus Unstructured Data:

- 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.

Structured Versus Unstructured Data:

- Because of this fact, data analytics methods that can be applied to unstructured data, such as cognitive computing and machine learning, are deservedly garnering a lot of attention.
- With machine learning applications, such as natural language processing (NLP), we can decode speech.
- With image/facial recognition applications, we can extract critical information from still images and video.
- Smart objects in IoT networks generate both structured and unstructured data

Module – 4 Data and Analytics for IoT

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

Module – 4 Data and Analytics for IoT

Data in Motion Versus Data at Rest:

- 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.
- Data does not come to rest at the edge.
- 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.
- Tools with this sort of capability, such as Spark, Storm, and Flink, are relatively nascent compared to the tools for analyzing stored data.

Module – 4 Data and Analytics for IoT

Data in Motion Versus Data at Rest:

- **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.**
- **Hadoop not only helps with data processing but also data storage.**

Module – 4 Data and Analytics for IoT

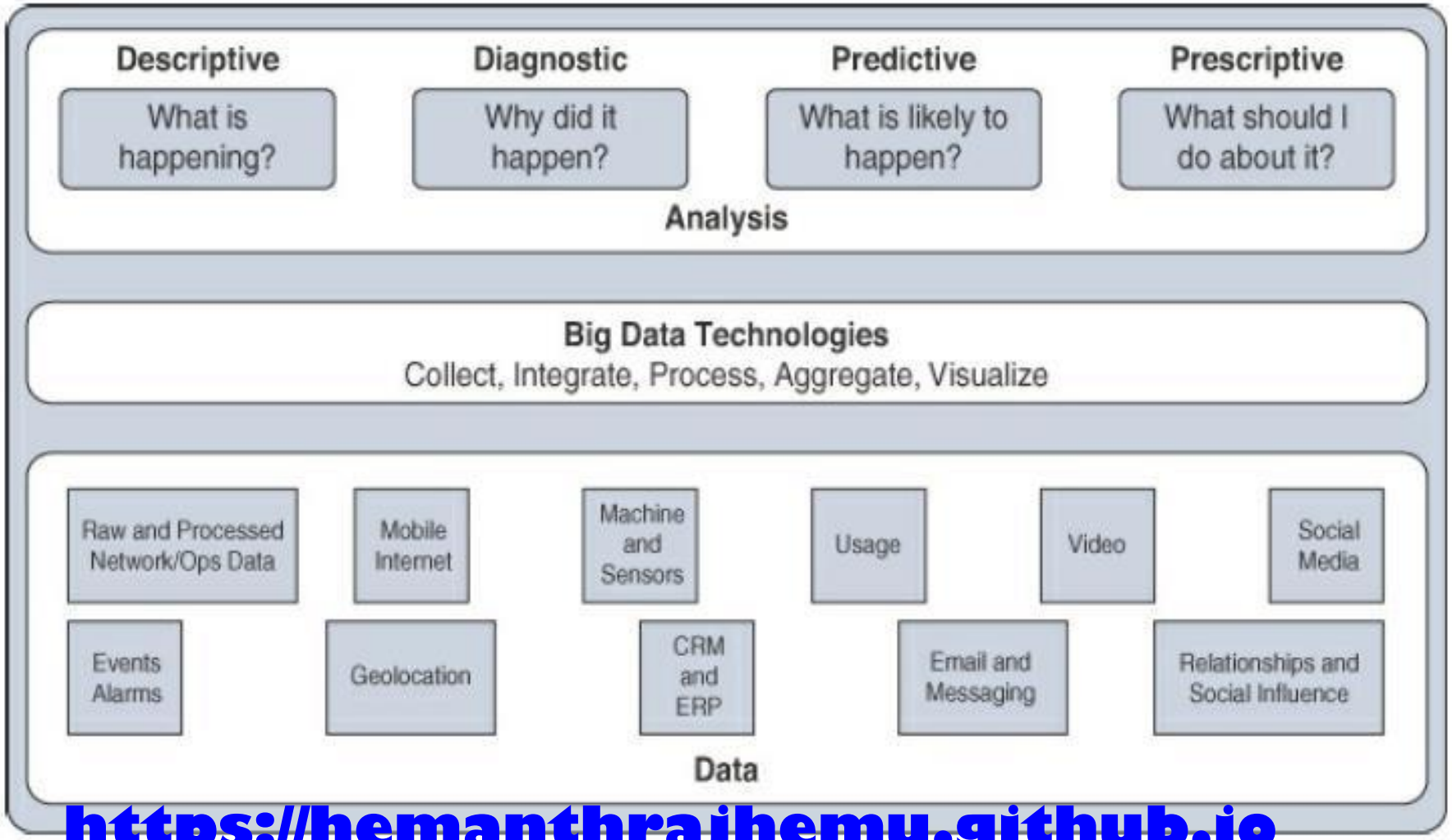
IoT Data Analytics Overview:

- The true importance 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 Figure, there are four types of data analysis results:

1 . Descriptive 2. Diagnostic 3. Predictive 4. Prescriptive

Module – 4 Data and Analytics for IoT

IoT Data Analytics Overview:



Types of Data Analysis Results

https://hemanthrajhemu.github.io

Module – 4 Data and Analytics for IoT

IoT Data Analytics Overview:

1 . Descriptive:

- **Descriptive data analysis tells us what is happening, either now or in the past.**
- **For example, 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**

Module – 4 Data and Analytics for IoT

IoT Data Analytics Overview:

2 . 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.**

Module – 4 Data and Analytics for IoT

IoT Data Analytics Overview:

3 . Predictive:

- **Predictive analysis aims to foretell problems or issues before they occur.**
- **For example, 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 perhaps 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.**

IoT Data Analytics Overview:

4 . 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.**

Module – 4 Data and Analytics for IoT

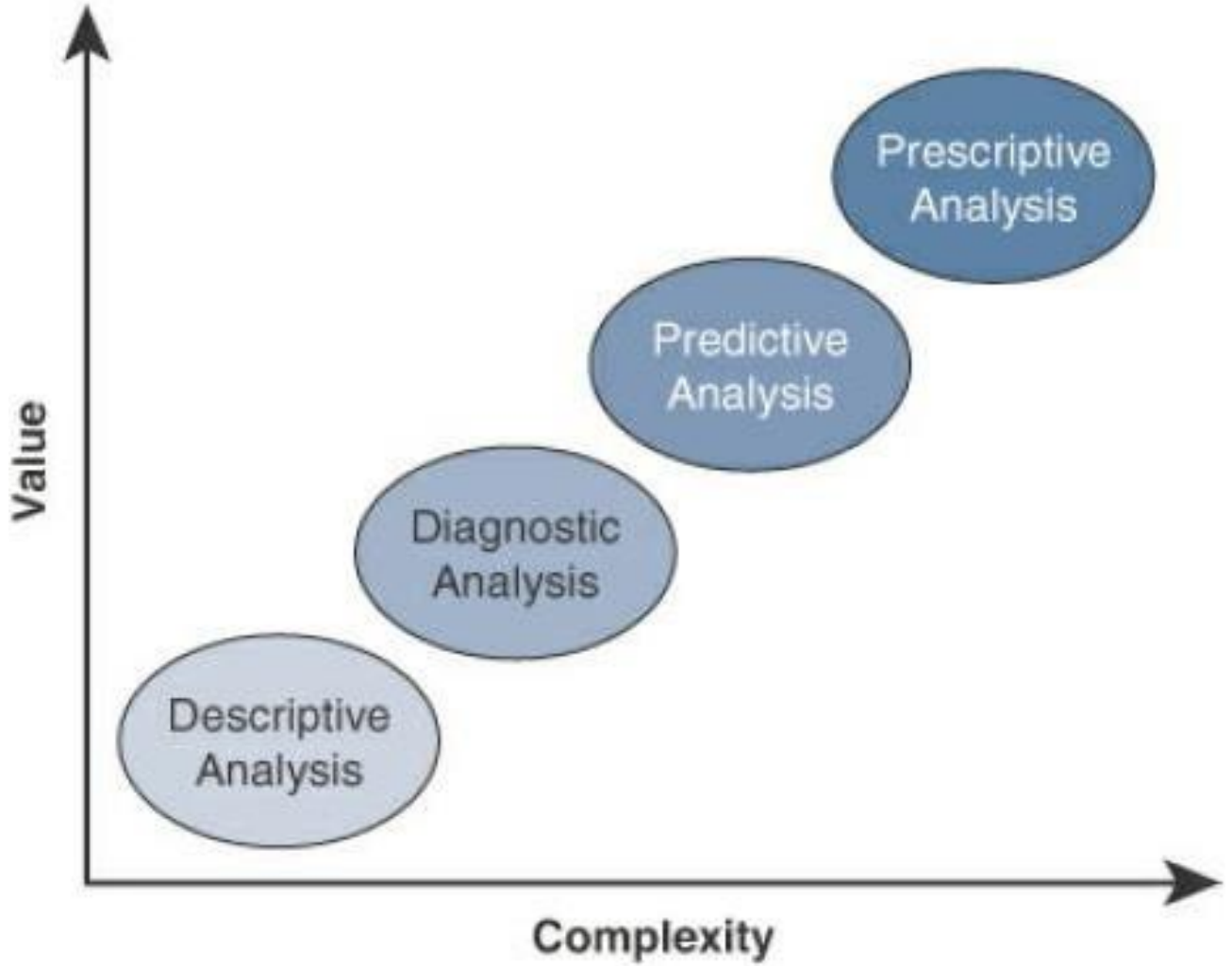
IoT Data Analytics Overview:

- 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.
- Figure illustrates the four data analysis types and how they rank as complexity and value increase.
- 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

Module – 4 Data and Analytics for IoT

IoT Data Analytics Overview:

- **Figure illustrates the four data analysis types and how they rank as complexity and value increase.**



Application of Value and Complexity Factors to the Types of Data Analysis

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

- While relational databases are still used for certain data types and applications, they often struggle with the nature of IoT data.
- IoT data places two specific challenges on a relational database:
- 1. 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.

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

2. Volatility of data:

- With relational databases, it is critical that the schema be designed correctly from the beginning.
- Changing it later can slow or stop the database from operating.
- Due to the lack of flexibility, revisions to the schema must be kept at a minimum.
- IoT data, however, is volatile , 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.

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

- 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.
- NoSQL database is a database that does not use SQL.
- It is not set up in the traditional tabular form of a relational database.

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

- NoSQL databases do not enforce a strict schema, and they support a complex, evolving data model and are also inherently much more scalable.
- In addition to the relational database challenges that IoT imposes, with its high volume of smart object data that frequently changes, IoT also brings challenges with the live streaming nature of its data and with managing data at the network level.
- Streaming data, which is generated as smart objects transmit data, is challenging because it is usually of a very high volume, and it is valuable only if it is possible to analyze and respond to it in real-time.

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

- The market for analyzing streaming data in real-time is growing fast.
- Major cloud analytics providers, such as Google, Microsoft, and IBM, have streaming analytics offerings, and various other applications can be used in house.
- Another challenge that IoT brings to analytics is in the area of network data, which is referred to as network analytics.

Module – 4 Data and Analytics for IoT

IoT Data Analytics Challenges:

- With the large numbers of smart objects in IoT networks that are communicating and streaming data, it can be challenging to ensure that these data flows are effectively managed, monitored, and secure.
- Network analytics tools such as Flexible NetFlow and IPFIX provide the capability to detect irregular patterns or other problems in the flow of IoT data through a network.

Module – 4 Data and Analytics for IoT

Machine Learning:

- **Machine learning, deep learning, neural networks, and convolutional networks are heard in relation to big data and IoT.**
- **ML is central to IoT.**
- **Data collected by smart objects needs to be analyzed, and intelligent actions need to be taken based on these analyses.**
- **Performing this kind of operation manually is almost impossible (or very, very slow and inefficient).**

Module – 4 Data and Analytics for IoT

Machine Learning:

- Machines are needed to process information fast and react instantly when thresholds are met.
- For example, every time a new advance is made in the field of self-driving vehicles, abnormal pattern recognition in a crowd, or any other automated intelligent and machine-assisted decision system, ML is named as the tool that made the advance possible.

Module – 4 Data and Analytics for IoT

Machine Learning Overview:

- Machine learning is 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.
- Any computer that uses rules to make decisions belongs to this realm.

Module – 4 Data and Analytics for IoT

Machine Learning Overview:

- A simple example is an app that can help us find your parked car.
- A GPS reading of our position at regular intervals calculates your speed.
- A basic threshold system determines whether we are driving (for example, "if speed > 20 mph or 30 kmh, then start calculating speed").
- A typical example is a dictation program that runs on a computer.
- The program is configured to recognize the audio pattern of each word in a dictionary, but it does not know your voice's specifics—your accent, tone, speed, and so on.

Module – 4 Data and Analytics for IoT

Machine Learning Overview:

- You need to record a set of predetermined sentences to help the tool match well-known words to the sounds you make when you say the words.
- This process is called machine learning.
- 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:

supervised and unsupervised learning.

Supervised Learning:

- In supervised learning, the machine is trained with input for which there is a known correct answer.
- For example, suppose that you are training a system to recognize when there is a human in a mine tunnel.
- A sensor equipped with a basic camera can capture shapes and return them to a computing system that is responsible for determining whether the shape is a human or something else (such as a vehicle, a pile of ore, a rock, a piece of wood, and so on.).

Supervised Learning:

- 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.
- The comparison is usually done at the scale of the entire image, or pixel by pixel.

Module – 4 Data and Analytics for IoT

Supervised Learning:

- Images are resized to have the same characteristics (resolution, color depth, position of the central figure, and so on), and each point is analyzed.
- Human images have certain types of shapes and pixels in certain locations (which correspond to the position of the face, legs, mouth, and so on).
- 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 and, therefore, the probability that what is shown is a human figure.
- This process is called classification.

Supervised Learning:

- 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, depending on the ML system used— to verify that the recognition level is at acceptable thresholds.
- If the machine does not reach the level of success expected, more training is needed.

Supervised Learning:

- In other cases, the learning process is not about classifying in two or more categories but about finding a correct value.
- For example, 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, where as classification predicts categories

Module – 4 Data and Analytics for IoT

Unsupervised Learning:

- In some cases, supervised learning is not the best method for a machine to help with a human decision.
- Suppose that you are processing IoT data from a factory manufacturing small engines.
- You know that about 0.1% of the produced engines on average need adjustments to prevent later defects, and your task is to identify them before they get mounted into machines and shipped away from the factory.

Module – 4 Data and Analytics for IoT

Unsupervised Learning:

- With hundreds of parts, it may be very difficult to detect the potential defects, and it is almost impossible to train a machine to recognize issues that may not be visible.
- we can test each engine and record multiple parameters, such as sound, pressure, temperature of key parts, and so on.
- Once data is recorded,we can graph these elements in relation to one another (for example, temperature as a function of pressure, sound versus rotating speed over time).
- We can then input this data into a computer and use mathematical functions to find groups.

Unsupervised Learning:

- For example, 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).
- Grouping the engines this way can quickly reveal several types of engines that all belong to the same category (for example, small engine of chainsaw type, medium engine of lawnmower type).

Unsupervised Learning:

- 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

Module – 4 Data and Analytics for IoT

Unsupervised Learning:

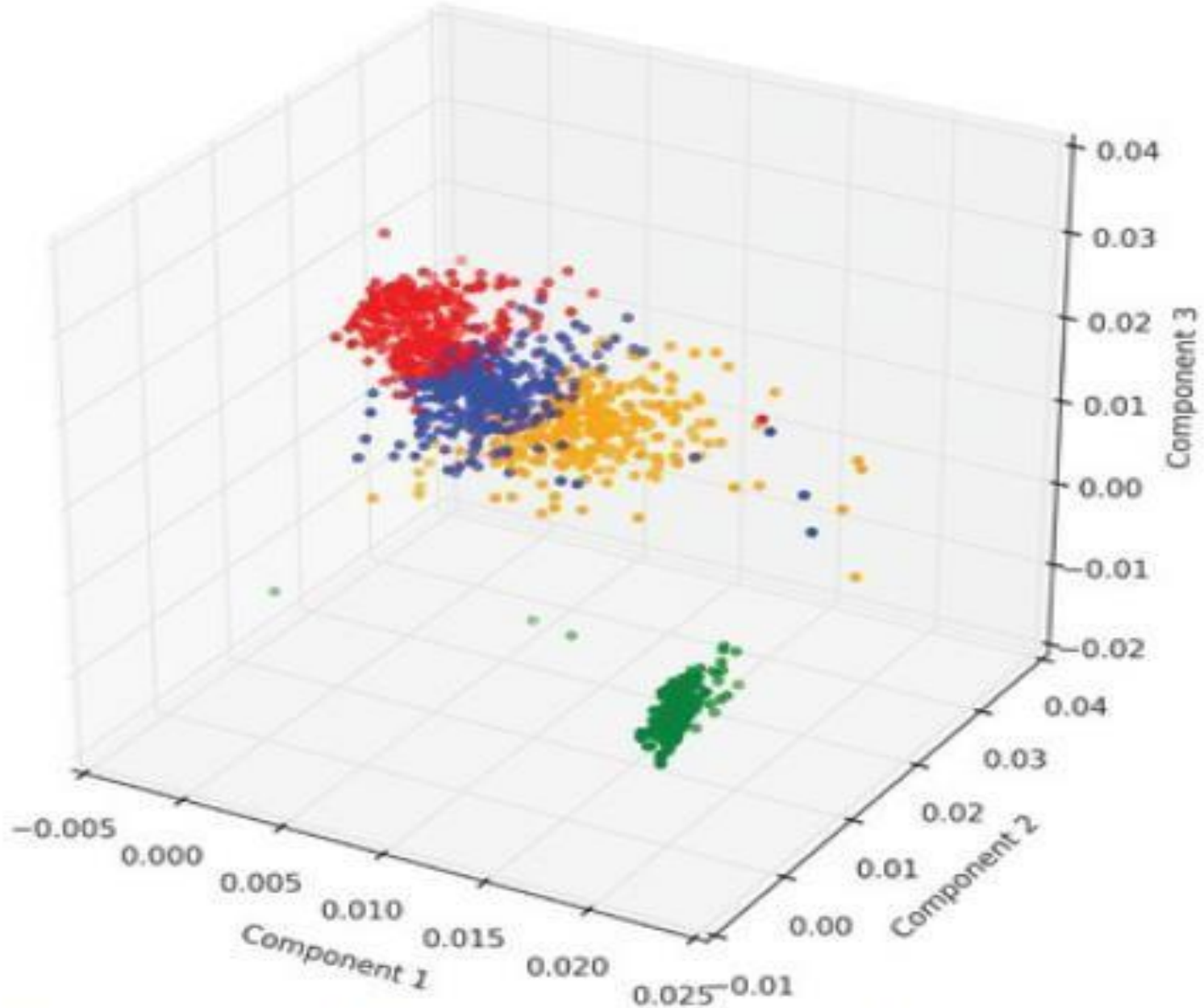
- The type of learning is unsupervised because there is not a "good" or "bad" answer known in advance.
- It is the variation from a group behavior that allows the computer to learn that something is different.
- The example of engines is, of course, very simple.
- In most cases, parameters are multidimensional.
- In other words, hundreds or thousands of parameters are computed, and small cumulated deviations in multiple dimensions are used to identify the exception.

Unsupervised Learning:

- Figure shows an example of such grouping and deviation identification logic.
- Three parameters are graphed (components 1, 2, and 3), and four distinct groups (clusters) are found.
- we can see some points that are far from the respective groups.
- Individual devices that display such "out of cluster" characteristics should be examined more closely individually.

Module – 4 Dat

Unsupervised Learning:



Clustering and Deviation Detection Example

https://hemanthrajhemu.github.io

Neural Networks:

- Processing multiple dimensions requires a lot of computing power.
- It is also difficult to determine what parameters to input and what combined variations should raise red flags.
- 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.

Module – 4 Data and Analytics for IoT

Neural Networks:

- Neural networks are ML methods that mimic the way the human brain works.
- When we look at a human figure, multiple zones of our brain are activated to recognize colors, movements, facial expressions, and so on.
- our brain combines these elements to conclude that the shape we are seeing is human.
- 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

Neural Networks:

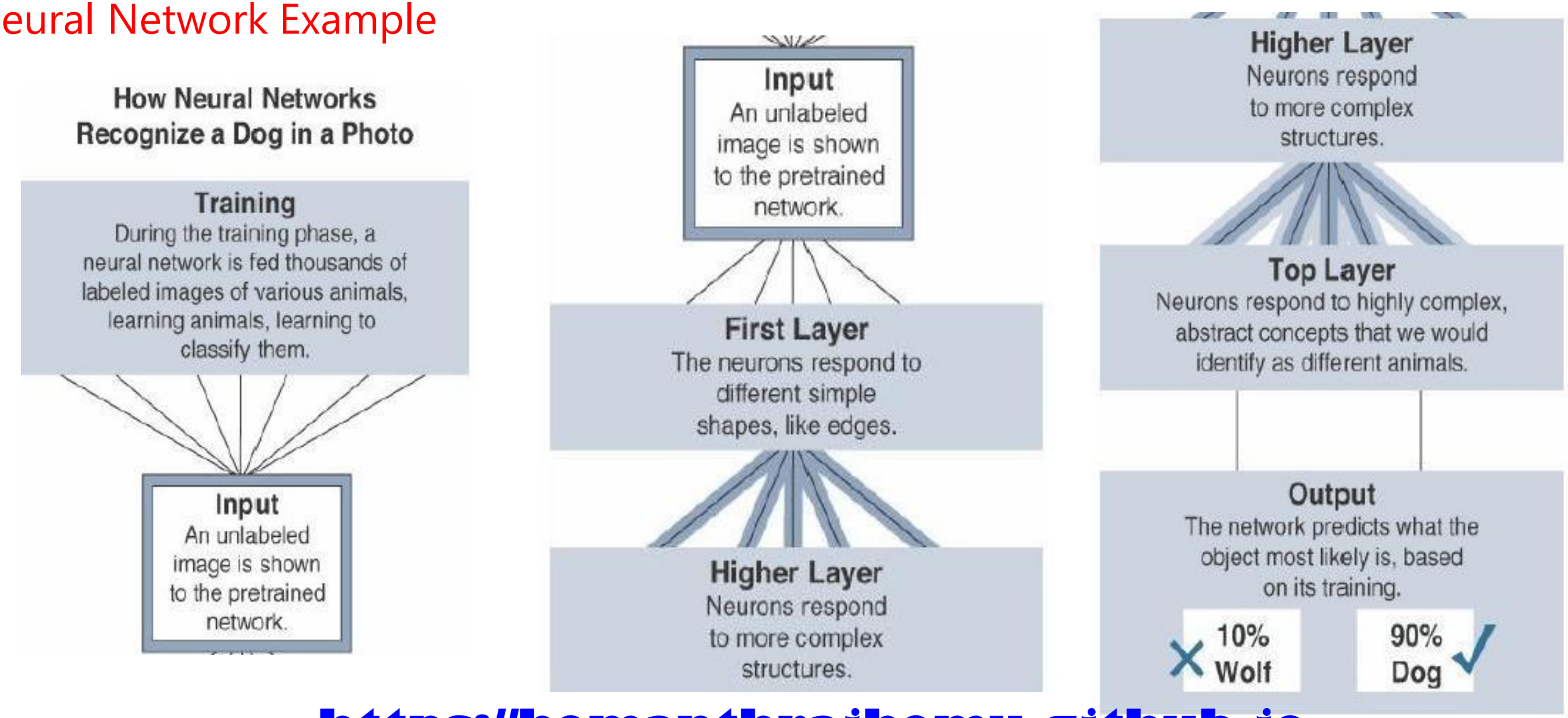
- The resulting value of one unit computation can be used directly or fed into another unit for further processing to occur.
- In this case, the neural network is said to have several layers.
- For example, a neural network processing human image recognition may have two units in a first layer that determines whether the image has straight lines and sharp angles—because vehicles commonly have straight lines and sharp angles, and human figures do not.

Neural Networks:

- If the image passes the first layer successfully (because there are no or only a small percentage of sharp angles and straight lines), a second layer may look for different features (presence of face, arms, and so on), and then a third layer might compare the image to images of various animals and conclude that the shape is a human (or not).
- The great efficiency of neural networks is that each unit processes a simple test, and therefore computation is quite fast.

Module – 4 Data and Analytics for IoT

Neural Network Example



Neural Networks:

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

Neural Networks:

- One advantage of deep learning is that having more layers allows for richer intermediate processing and representation of the data.
- At each layer, the data can be formatted to be better utilized by the next layer.
- This process increases the efficiency of the overall result.

Machine Learning and Getting Intelligence from Big Data:

- When the principles of machine learning are clear, the application to IoT becomes obvious.
- The difficulty resides in determining the right algorithm and the right learning model for each use case.
- Such an analysis goes beyond the scope but it can be useful to organize ML operations into two broad subgroups:

  1. Local learning 2. Remote learning

Machine Learning and Getting Intelligence from Big Data:

- **1. Local learning**
  - In this group, data is collected and processed locally, either in the sensor itself (the edge node) or in the gateway (the fog node).
- **2. Remote learning**
  - In this group, data is collected and sent to a central computing unit (typically the data center in a specific location or in the cloud), where it is processed.

Machine Learning and Getting Intelligence from Big Data:

- Regardless of the location where (and, therefore, the scale at which) data is processed, common applications of ML for IoT revolve around four major domains:

1. Monitoring
2. Behavior control
3. Operations optimization
4. Self-healing, self-optimizing

Machine Learning and Getting Intelligence from Big Data:

1. Monitoring:

- Smart objects monitor the environment where they operate.
- Data is processed to better understand the conditions of operations.
- These conditions can refer to external factors, such as air temperature, humidity, or presence of carbon dioxide in a mine, or to operational internal factors, such as the pressure of a pump, the viscosity of oil flowing in a pipe, and so on.

Machine Learning and Getting Intelligence from Big Data:

1. Monitoring:

- ML can be used with monitoring to detect early failure conditions (for example, K-means deviations showing out of- range behavior) or to better evaluate the environment (such as shape recognition for a robot automatically sorting material or picking goods in a warehouse or a supply chain).

Machine Learning and Getting Intelligence from Big Data:

2. Behavior control:

- **Monitoring commonly works in conjunction with behavior control.**
- **When a given set of parameters reach a target threshold —defined in advance (that is, supervised) or learned dynamically through deviation from mean values (that is, unsupervised)—monitoring functions generate an alarm.**
- **This alarm can be relayed to a human, but a more efficient and more advanced system would trigger a corrective action, such as increasing the flow of fresh air in the mine tunnel, turning the robot arm, or reducing the oil pressure in the pipe.**

Machine Learning and Getting Intelligence from Big Data:

3. Operations optimization:

- Behavior control typically aims at taking corrective actions based on thresholds.
- However, analyzing data can also lead to changes that improve the overall process.
- For example, a water purification plant in a smart city can implement a system to monitor the efficiency of the purification process based on which chemical (from company A or company B) is used, at what temperature, and associated to what stirring mechanism (stirring speed and depth).

Machine Learning and Getting Intelligence from Big Data:

3. Operations optimization:

- Neural networks can combine multiples of such units, in one or several layers, to estimate the best chemical and stirring mix for a target air temperature.
- This intelligence can help the plant reduce its consumption of chemicals while still operating at the same purification efficiency level.
- As a result of the learning, behavior control results in different machine actions.
- The objective is not merely to pilot the operations but to improve the efficiency and the result of these operations.

Machine Learning and Getting Intelligence from Big Data:

4. Self-healing, self-optimizing:

- A fast-developing aspect of deep learning is the closed loop.
- ML-based monitoring triggers changes in machine behavior (the change is monitored by humans), and operations optimizations.
- In turn, the ML engine can be programmed to dynamically monitor and combine new parameters (randomly or semi-randomly) and automatically deduce and implement new optimizations when the results demonstrate a possible gain.

Machine Learning and Getting Intelligence from Big Data:

4. Self-healing, self-optimizing:

- The system becomes self-learning and self optimizing.
- It also detects new K-means deviations that result in predetection of new potential defects, allowing the system to self-heal.
- The healing is not literal, as external factors (typically human operators) have to intervene, but the diagnosis is automated.

Machine Learning and Getting Intelligence from Big Data:

4. Self-healing, self-optimizing:

- In many cases, the system can also automatically order a piece of equipment that is detected as being close to failure or automatically take corrective actions to avoid the failure (for example, slow down operations, modify a machine's movement to avoid fatigue on a weak link).
  - For all these operations, a specific aspect of ML for IoT is the scale.
  - A weather sensor mounted on a light pole in a street can provide information about the local pollution level.

Predictive Analytics:

- When data from multiple systems is combined and analyzed together, predictions can be made about the state of the system.
- For example, "Transportation," examines the case of sensors deployed on locomotives.
- Multiple smart objects measure the pull between carriages, the weight on each wheel, and multiple other parameters to offer a form of cruise control optimization for the driver.

Module – 4 Data and Analytics for IoT

Predictive Analytics:

- At the same time, cameras observe the state of the tracks ahead, audio sensors analyze the sound of each wheel on the tracks, and multiple engine parameters are measured and analyzed.
- All this data can be returned to a data processing center in the cloud that can re-create a virtual twin of each locomotive.
- Modeling the state of each locomotive and combining with anticipated travel and with the states (and detected failures) of all other locomotives of the entire city or country allows the analytics platform to make very accurate predictions on whatissue is likely to affect each train and each locomotive.

Module – 4 Data and Analytics for IoT

Big Data Analytics Tools and Technology:

- Hadoop is at the core of many of today's big data implementations.
- Big data analytics can consist of many different software pieces that together collect, store, manipulate, and analyze all different data types.
- It helps to better understand the landscape by defining what big data is and what it is not.
- Generally, the industry looks to the "three Vs" to categorize big data:

  1. Velocity 2. Variety 3. Volume

Big Data Analytics Tools and Technology:

1. Velocity:

- Velocity refers to how quickly data is being collected and analyzed.
- Hadoop Distributed File System is designed to ingest and process data very quickly.
- Smart objects can generate machine and sensor data at a very fast rate and require database or file systems capable of equally fast ingest functions.

Big Data Analytics Tools and Technology:

2. Variety:

- Variety refers to different types of data and data is categorized as structured, semi-structured, or unstructured.
- Different database technologies may only be capable of accepting one of these types.
- Hadoop is able to collect and store all three types.

Big Data Analytics Tools and Technology:

3. Volume:

- **Volume refers to the scale of the data.**
- **Typically, this is measured from gigabytes on the very low end to petabytes or even exabytes of data on the other extreme.**
- **Generally, big data implementations scale beyond what is available on locally attached storage disks on a single node.**
- **It is common to see clusters of servers that consist of dozens, hundreds, or even thousands of nodes for some large deployments.**

Big Data Analytics Tools and Technology:

- The characteristics of big data can be defined by the sources and types of data.
- First is machine data, which is generated by IoT devices and is typically unstructured data.
- Second is transactional data, which is from sources that produce data from transactions on these systems, and, have high volume and structured.
- Third is social data sources, which are typically high volume and structured.
- Fourth is enterprise data, which is data that is lower in volume and very structured. Hence big data consists of data from all these separate sources.

Big Data Analytics Tools and Technology:

- Data ingest is the layer that connects data sources to storage.
- It's the layer that preprocesses, validates, extracts, and stores data temporarily for further processing.
- There are several patterns to consider for data ingest.
- First is multisource ingestion, which connects multiple data sources to ingest systems.

Module – 4 Data and Analytics for IoT

Big Data Analytics Tools and Technology:

- In multisource ingestion pattern, ingest nodes receive streams of data from multiple sources and do processing before passing the data to intermediate nodes and to final store nodes.
- This pattern is typically implemented in batch systems and in real-time systems.

Big Data Analytics Tools and Technology:

- Many industrial automation and control systems feed data into two distinct database types, relational databases and historians.
- Relational databases, such as Oracle and Microsoft SQL, are good for transactional, or process, data.
- Their benefit is being able to analyze complex data relationships on data that arrives over a period of time.

Big Data Analytics Tools and Technology:

- On the other hand, historians are optimized for time-series data from systems and processes.
- They are built with speed of storage and retrieval of data at their core, recording each data point in a series with the pertinent information about the system being logged.
- This data may consist of a sensor reading, the quantity of a material, a temperature reading, or flow data.

Big Data Analytics Tools and Technology:

- Relational databases and historians are mature technologies that have been with us for many years, but new technologies and techniques in the data management market have opened up new possibilities for sensor and machine data.
- These database technologies broadly fit into a few categories that each have strengths and potential drawbacks when used in an IoT context.
- The three most popular of these categories are massively parallel processing systems, NoSQL, and Hadoop.

Massively Parallel Processing Databases:

- Enterprises have used relational databases for storing structured, row and column style data types for decades.
- Relational databases are often grouped into a broad data storage category called data warehouses.
- Though they are the centerpiece of most data architectures, they are often used for longer-term archiving and data queries that can often take minutes or hours.
- Depending on the number of items in the database and the complexity of the question being asked, the response could be slow to return.

Massively Parallel Processing Databases:

- Massively parallel processing (MPP) databases were built on the concept of the relational data warehouses but are designed to be much faster, to be efficient, and to support reduced query times.
- To accomplish this, MPP databases take advantage of multiple nodes (computers) designed in a scale out architecture such that both data and processing are distributed across multiple systems.

Massively Parallel Processing Databases:

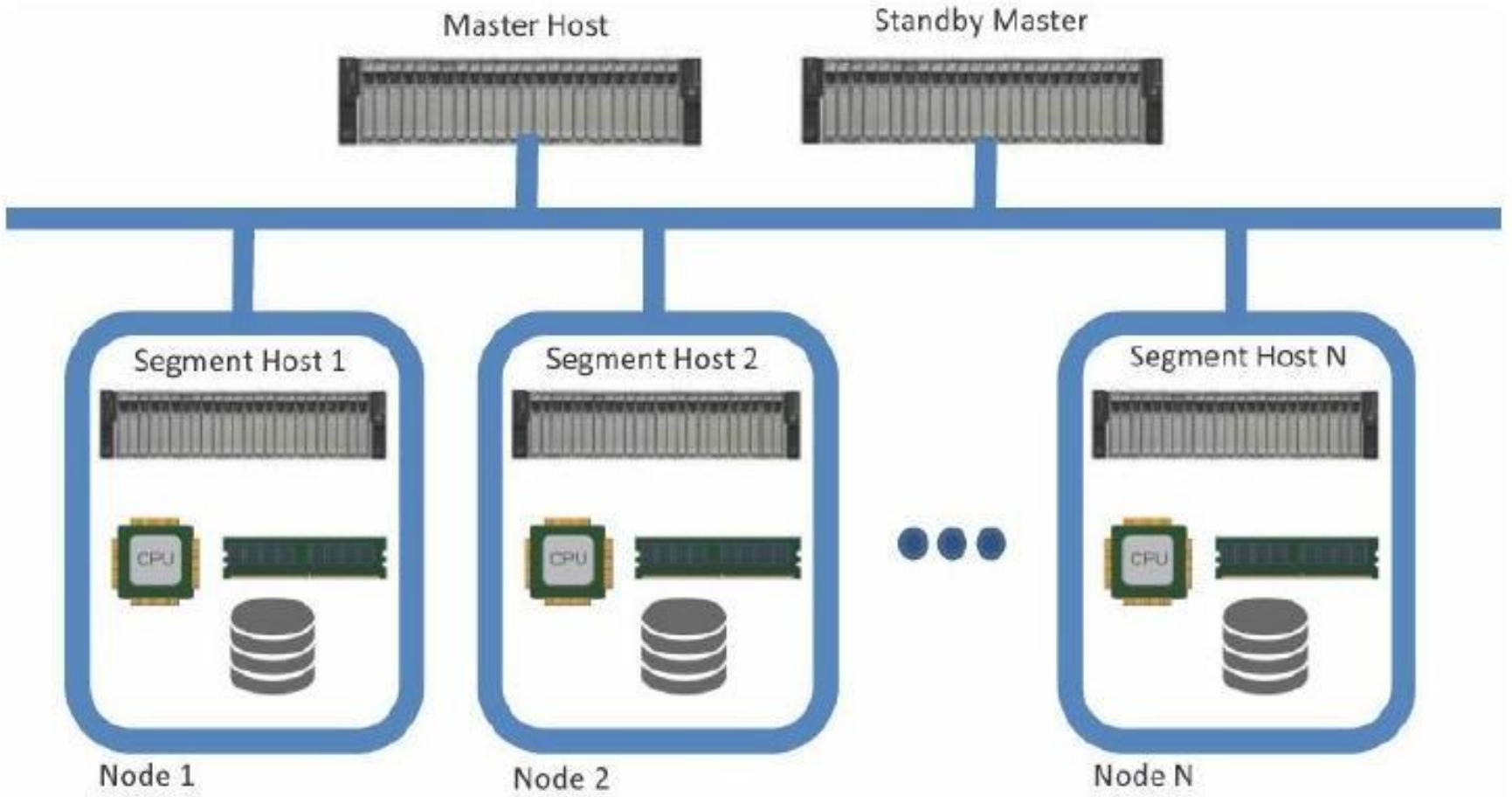
- MPPs are sometimes referred to as analytic databases because they are designed to allow for fast query processing and often have built-in analytic functions.
- As the name implies, these database types process massive data sets in parallel across many processors and nodes.
- An MPP architecture typically contains a single master node that is responsible for the coordination of all the data storage and processing across the cluster.
- It operates in a "shared-nothing" fashion, with each node containing local processing, memory, and storage and operating independently

Massively Parallel Processing Databases:

- Data storage is optimized across the nodes in a structured SQL-like format that allows data analysts to work with the data using common SQL tools and applications.
- The earlier example of a complex SQL query could be distributed and optimized, resulting in a signifycantly faster response.
- The sources and types of data may vary, requiring a database that is more flexible than relational databases allow.

Module – 4 Data and Analytics for IoT

Massively Parallel Processing Databases:



MPP Shared-Nothing Architecture

https://hemanthrajhemu.github.io

NoSQL Databases:

- NoSQL ("not only SQL") is a class of databases that support semi-structured and unstructured data, in addition to the structured data handled by data warehouses and MPPs.

- NoSQL is not a specific database technology; rather, it is an umbrella term that encompasses several different types of databases, including the following:

1. Document stores
2. Key-value stores
3. Wide-column stores
4. Graph stores

NoSQL Databases:

1. Document stores:

- This type of database stores semi-structured data, such as XML or JSON.
- Document stores generally have query engines and indexing features that allow for many optimized queries.

2. Key-value stores:

- This type of database stores associative arrays where a key is paired with an associated value.
- These databases are easy to build easy to scale

Module – 4 Data and Analytics for IoT

NoSQL Databases:

1. Document stores:

- This type of database stores semi-structured data, such as XML or JSON.
- Document stores generally have query engines and indexing features that allow for many optimized queries.

2. Key-value stores:

- This type of database stores associative arrays where a key is paired with an associated value.
- These databases are easy to build and easy to scale

NoSQL Databases:

3. Wide-column stores:

- This type of database stores similar to a key value store, but the formatting of the values can vary from row to row, even in the same table.

4. Graph stores:

- This type of database is organized based on the relationships between elements.
- Graph stores are commonly used for social media or natural language processing,

NoSQL Databases:

- NoSQL was developed to support the high-velocity, urgent data requirements of modern web applications that typically do not require much repeated use.
- The original intent was to quickly ingest rapidly changing server logs and clickstream data generated by web-scale applications that did not neatly fit into the rows and columns required by relational databases.
- NoSQL is built to scale horizontally, allowing the database to span multiple hosts, and can even be distributed geographically.

NoSQL Databases:

- NoSQL key-value stores are capable of handling indexing and persistence simultaneously at a high rate.
- This makes it a great choice for time series data sets, which record a value at a given interval of time, such as a temperature or pressure reading from a sensor.
- Many NoSQL databases provide additional capabilities, such as being able to query and analyze data within the database itself, eliminating the need to move and process it elsewhere.
- They also provide a variety of ways to query the database through an API, making it easy to integrate them with others data

Module – 4 Data and Analytics for IoT

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.
- Hadoop was originally developed as a result of projects at Google and Yahoo!, and the original intent for Hadoop was to index millions of websites and quickly return search results for open source search engines.

Hadoop:

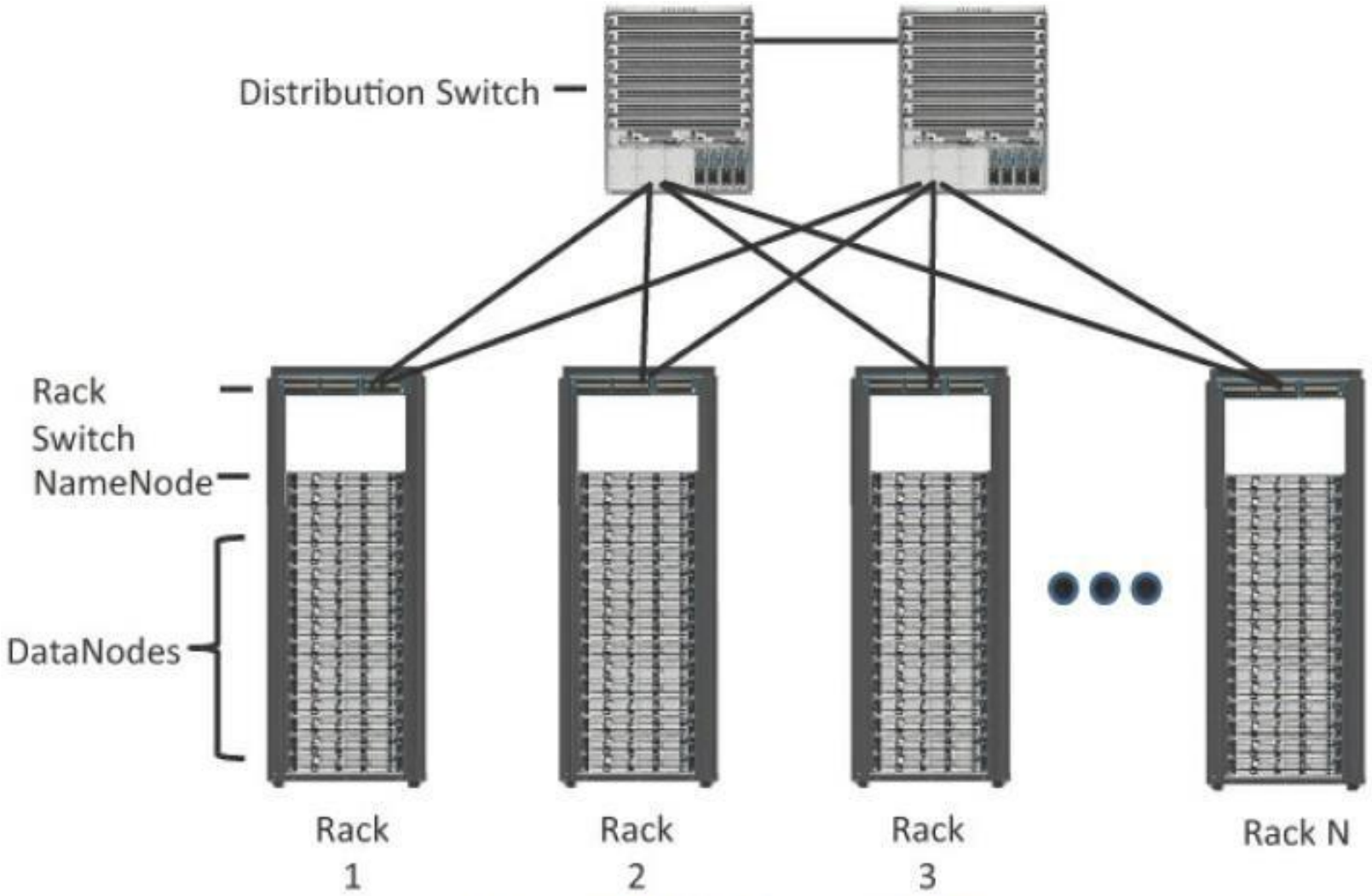
- Initially, the project had two key elements:

1. **Hadoop Distributed File System (HDFS):**
   - A system for storing data across multiple nodes
2. **MapReduce:**
   - A distributed processing engine that splits a large task into smaller ones that can be run in parallel.

- Both of these elements are still present in current Hadoop distributions and provide the foundation for other projects

Module – 4 Data and Analytics for IoT

Hadoop:



https://hemanthrajhemu.github.io

Module – 4 Data and Analytics for IoT

Hadoop:

- Much like the MPP and NoSQL systems, Hadoop relies on a scale-out architecture that leverages local processing, memory, and storage to distribute tasks and provide a scalable storage system for data.
- 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

Hadoop:

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.

Hadoop:

NameNodes:

- 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.

Hadoop:

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.

Hadoop:

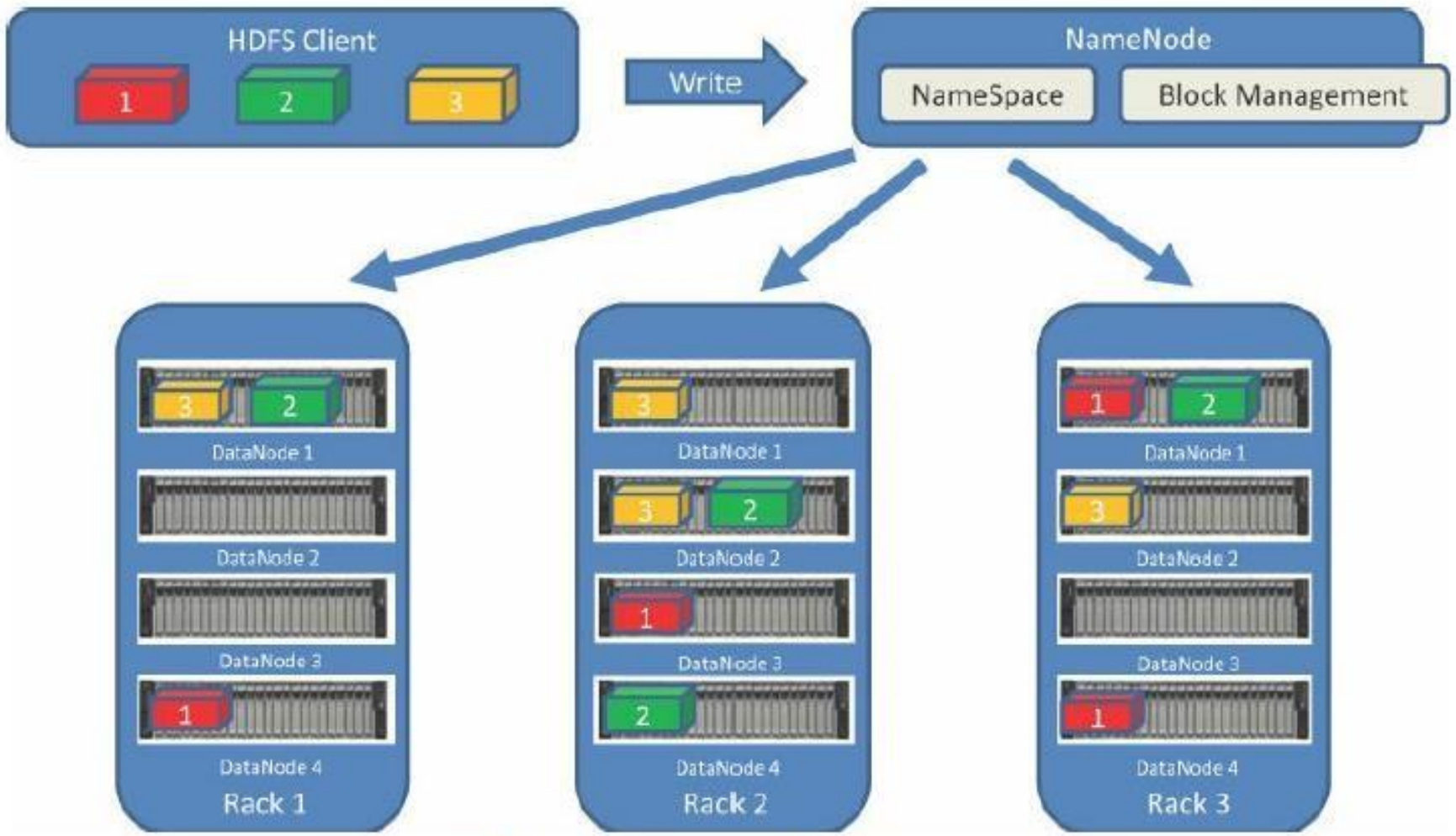
DataNodes:

- 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.
- Disk redundancy techniques such as Redundant Array of Independent Disks (RAID) are generally not used for HDFS because the NameNodes and DataNodes coordinate block-level redundancy with this replication technique.

Module – 4 Data and Analytics for IoT

Hadoop:

Figure shows the relationship between NameNodes and DataNodes and how data blocks are distributed across the cluster.



Writing a File to HDFS

https://hemanthrajhemu.github.io

Hadoop:

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

Module – 4 Data and Analytics for IoT

Hadoop:

- **While this is useful for understanding patterns and trending in historical sensor or machine data, it has one significant drawback: time.**
- **Depending on how much data is being queried and the complexity of the query, the result could take seconds or minutes to return.**
- **If you have a real-time process running where you need a result at a moment's notice, MapReduce is not the right data processing engine for that.**

Hadoop:

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.

Module – 4 Data and Analytics for IoT

Hadoop:

YARN:

- 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, in addition to batch processing using MapReduce.

The Hadoop Ecosystem:

- Hadoop now comprises more than 100 software projects under the Hadoop umbrella, capable of nearly every element in the data lifecycle, from collection, to storage, to processing, to analysis and visualization.

Hadoop:

The Hadoop Ecosystem - Apache Kafka:

- 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.

Module – 4 Data and Analytics for IoT

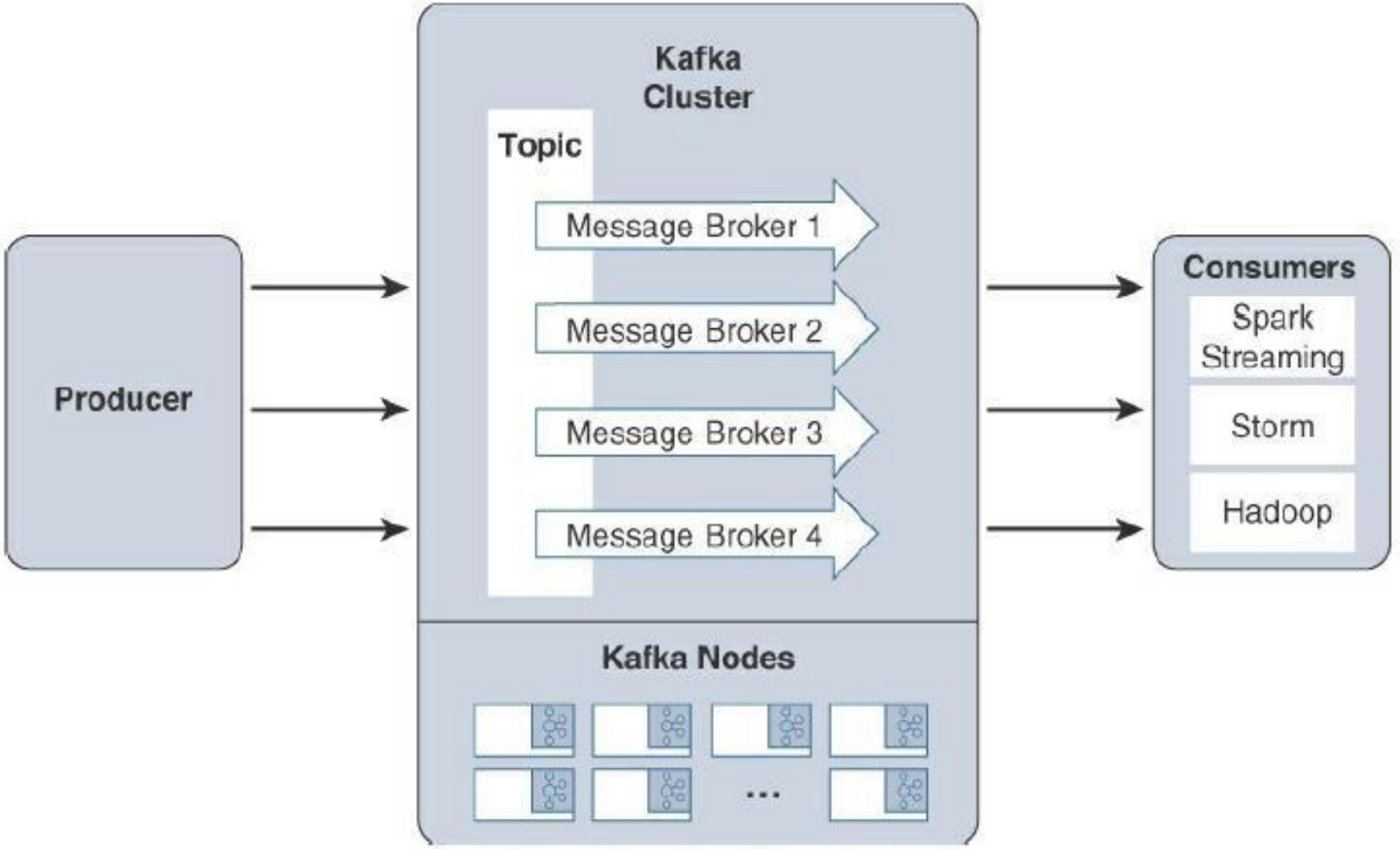
Hadoop: The Hadoop Ecosystem - Apache Kafka:

- It is composed of topics, or message brokers, where producers write data and consumers read data from these topics.
- Figure shows the data flow from the smart objects (producers), through a topic in Kafka, to the real-time processing engine.
- 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 [illegible] so that data in the way they would like.

Module – 4 Data and Analytics for IoT

Hadoop: The Hadoop Ecosystem -

Apache Kafka:



Module – 4 Data and Analytics for IoT

Hadoop: The Hadoop Ecosystem - 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.
- However, with Spark, the processing of this data is moved into high-speed memory, which has significantly lower latency.
- This speeds the batch processing jobs and also allows for near-real-time processing of events.

Hadoop: The Hadoop Ecosystem - Apache Spark:

- 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.

Hadoop: The Hadoop Ecosystem - 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.**

Hadoop: The Hadoop Ecosystem - Lambda Architecture:

- Ultimately the key elements of a data infrastructure to support many IoT use cases involves the collection, processing, and storage of data using multiple technologies.
- 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.

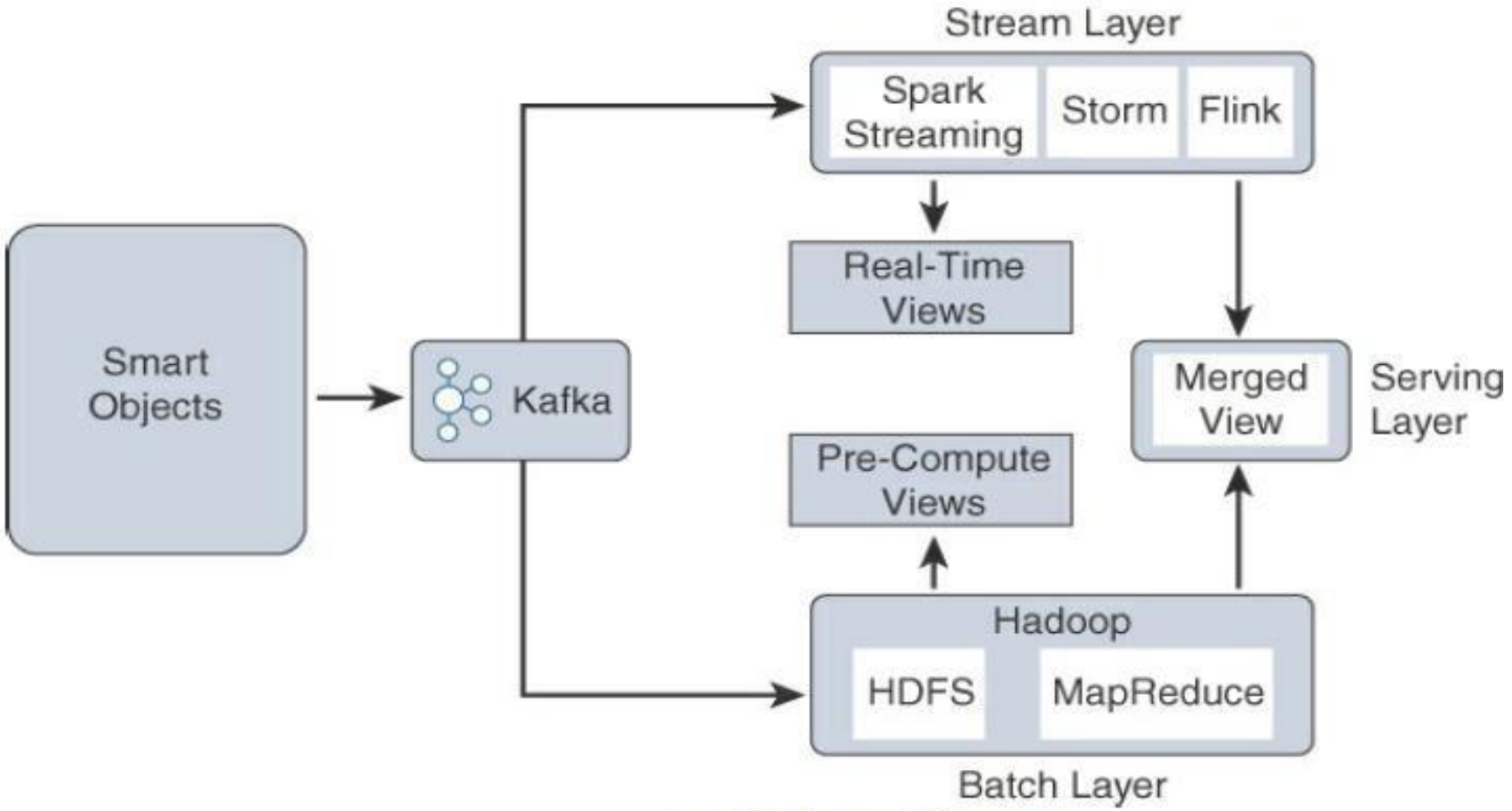
Module – 4 Data and Analytics for IoT

Hadoop: The Hadoop Ecosystem - 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.

Module – 4 Data and Analytics for IoT

Hadoop: The Hadoop Ecosystem - Lambda Architecture:



Lambda Architecture

Module – 4 Data and Analytics for IoT

Hadoop: The Hadoop Ecosystem - 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:

1. **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 [illegible]

Hadoop: The Hadoop Ecosystem - Lambda Architecture:

2. 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.

Hadoop: The Hadoop Ecosystem - Lambda Architecture:

3. 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.

Hadoop: The Hadoop Ecosystem - Lambda Architecture:

- 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.

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 key values of edge streaming analytics include the following:

1. 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.**

Comparing Big Data and Edge Analytics:

The key values of edge streaming analytics include the following:

2. 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.**

3. Time sensitivity:

- **When timely response to data is required, passing data to the cloud for future processing results in unacceptable latency.**

Comparing Big Data and Edge Analytics:

Edge Analytics Core Functions:

- **To perform analytics at the edge, data needs to be viewed as real-time flows.**
- **Whereas big data analytics is focused on large quantities of data at rest, edge analytics continually processes streaming flows of data in motion.**
- **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:

Comparing Big Data and Edge Analytics:

Edge Analytics Core Functions:

1. Raw input data:

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

2. 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.**

Comparing Big Data and Edge Analytics:

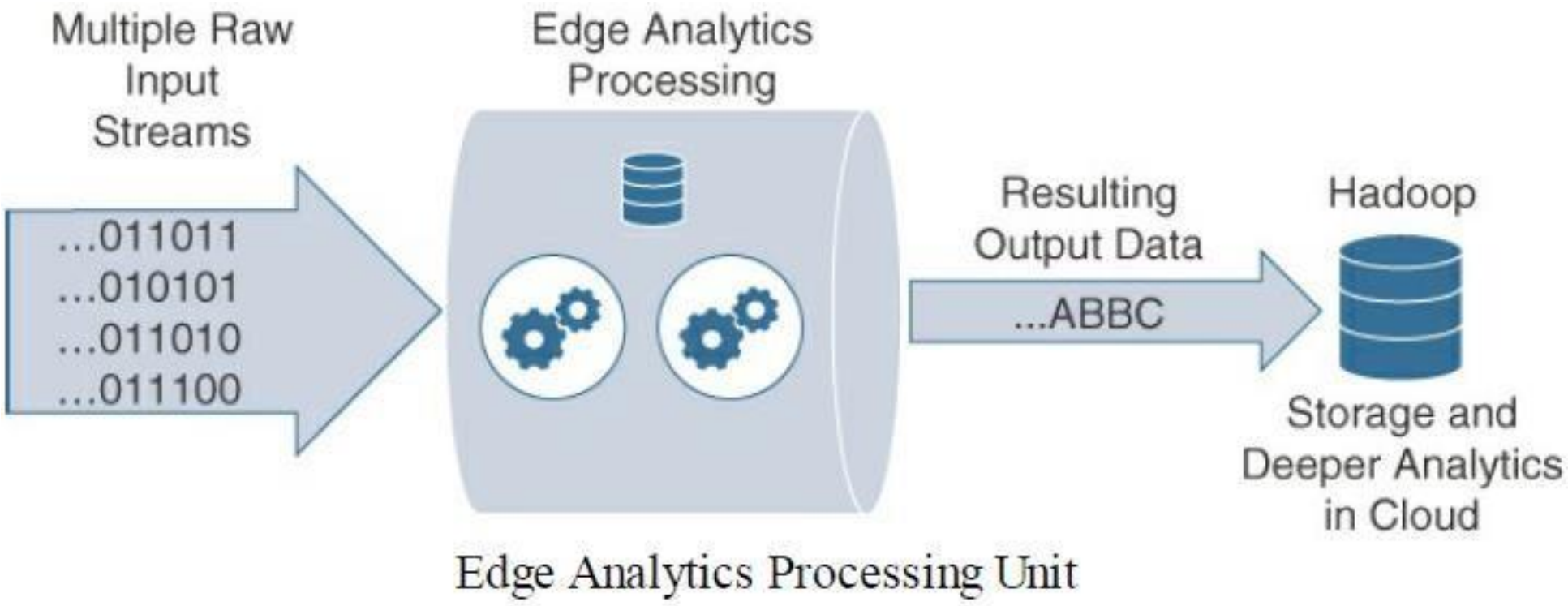
Edge Analytics Core Functions:

3. 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.**

Comparing Big Data and Edge Analytics:

Figure illustrates the stages of data processing in an edge APU.



Comparing Big Data and Edge Analytics:

Edge Analytics Core Functions:

In order to perform analysis in real-time, the APU needs to perform the following functions:

1. **Filter:**

- **The streaming data generated by IoT endpoints is likely to be very large, and most of it is irrelevant. For example, a sensor may simply poll on a regular basis to confirm that it is still reachable.**

- **The filtering function identifies the information that is considered important.**

Comparing Big Data and Edge Analytics:

Edge Analytics Core Functions:

2. 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.**

Module – 4 Data and Analytics for IoT

Comparing Big Data and Edge Analytics:

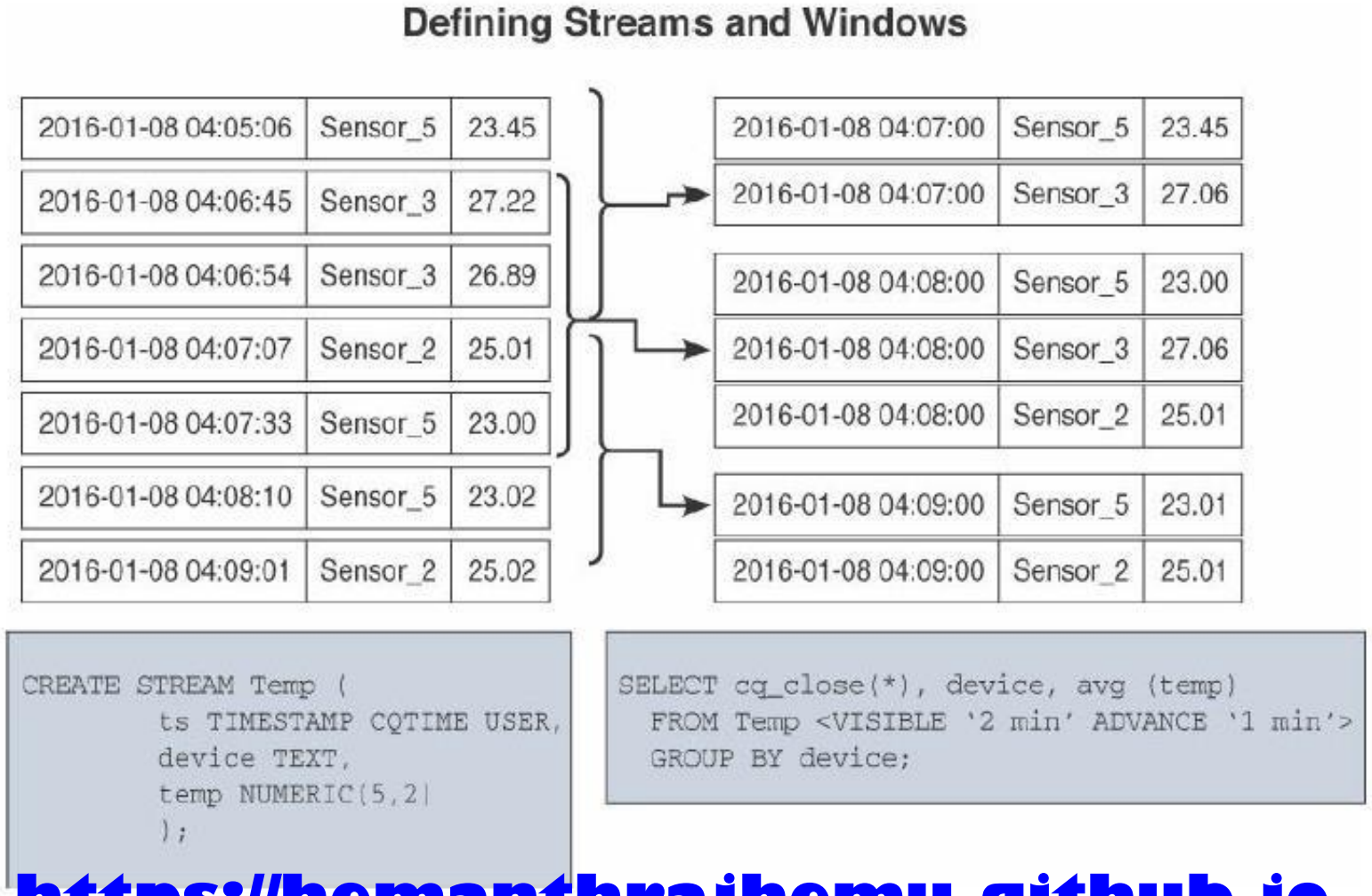
Edge Analytics Core Functions:

3. 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.**
- **For example, Figure shows an APU that takes input data from multiple sensors reporting temperature fluctuations. In this case, the APU is programmed to report the average temperature every minute from the sensors, based on an average of the past two minutes.**

Module – 4 Data and Analytics for IoT

Comparing Big Data and Edge Analytics:



https://hemanthrajhemu.github.io

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

4. Correlate:

- **Streaming data analytics becomes most useful when multiple data streams are combined from different types of sensors.**
- **For example, in a hospital, several vital signs are measured for patients, including body temperature, blood pressure, heart rate, and respiratory rate.**
- **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**

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

4. Correlate:

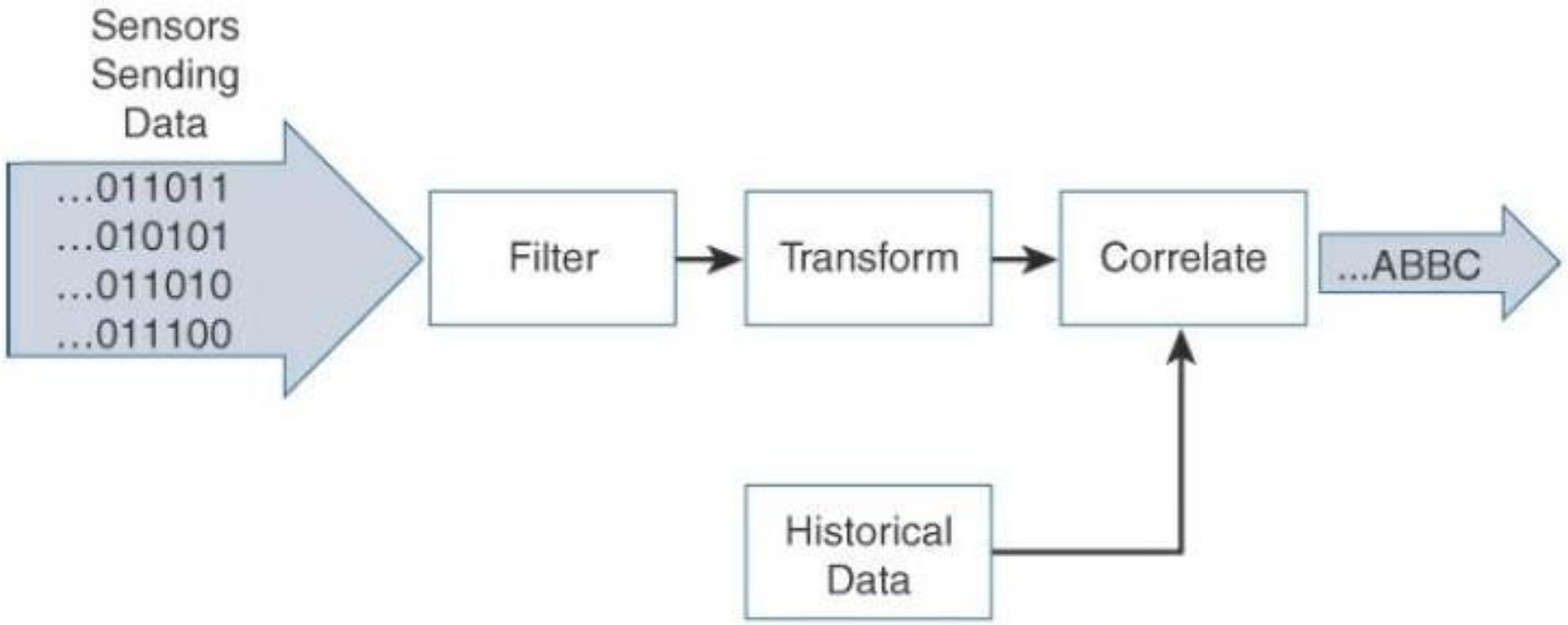
- **Streaming data analytics becomes most useful when multiple data streams are combined from different types of sensors.**
- **For example, in a hospital, several vital signs are measured for patients, including body temperature, blood pressure, heart rate, and respiratory rate.**
- **These different 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**

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

4. Correlate:

- **For example, historical data may include the patient's past medical history, such as blood test results.**
- **Combining historical data gives the live streaming data a powerful context and promotes more insights into the current condition of the patient (see Figure).**

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:



Correlating Data Streams with Historical Data

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

5. 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 example, 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.**

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

5. Match patterns:

- 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.

Comparing Big Data and Edge Analytics: Edge Analytics Core Functions:

6. Improve business intelligence:

- **Ultimately, the value of edge analytics is in the improvements to business intelligence that were not previously available.**
- **For example, 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.**

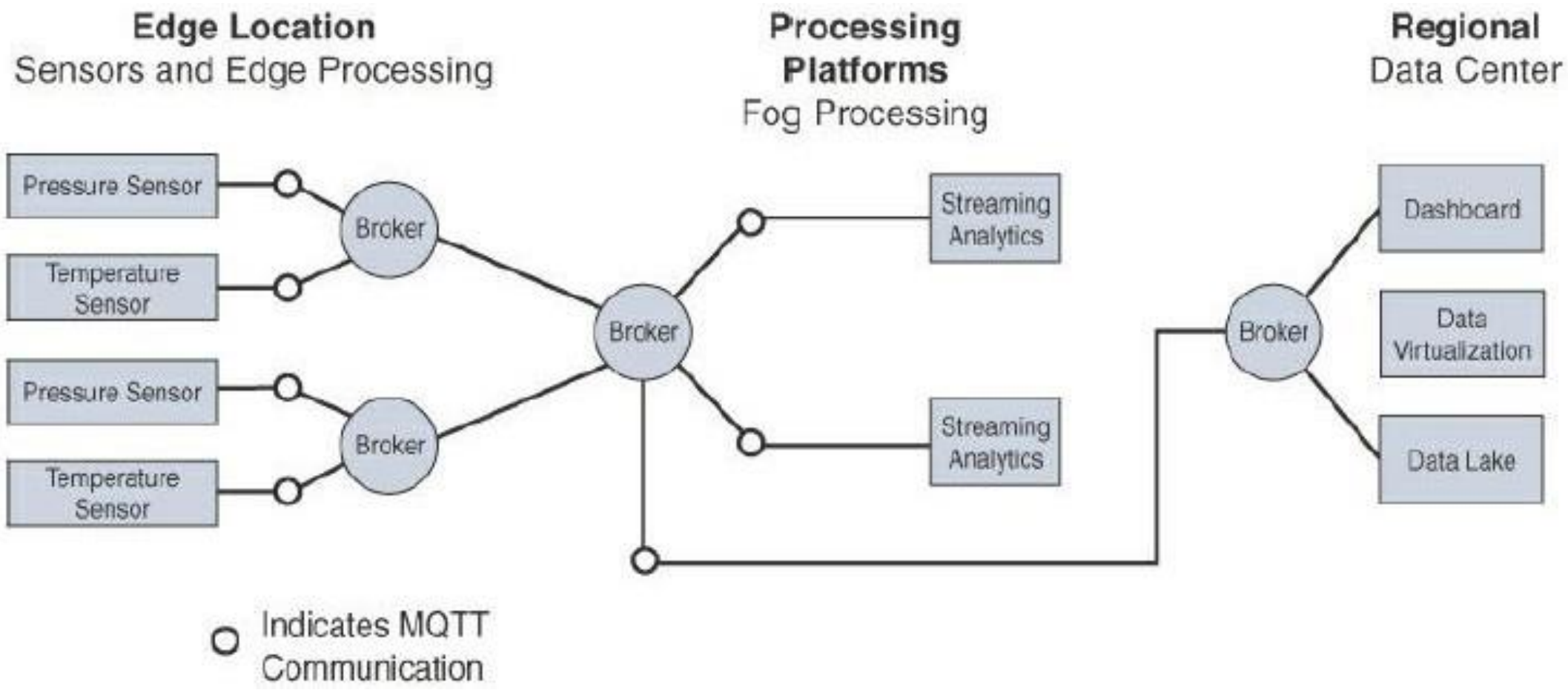
Module – 4 Data and Analytics for IoT

Distributed Analytics Systems:

- **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.**
- **Fog analytics allows, to see beyond one device, giving visibility into an aggregation of edge nodes and allowing to correlate data from a wider set.**
- **Figure shows an example of an oil drilling company that is measuring both pressure and temperature on an oil rig.**

Module – 4 Data and Analytics for IoT

Distributed Analytics Systems:



Distributed Analytics Throughout the IoT System

Module – 4 Data and Analytics for IoT

Distributed Analytics Systems:

- **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.**
- **It may not be able to respond to an event as quickly as analytics performed directly on the edge device, but it is still close to responding in real-time as events occur.**
- **Once the fog node is finished with the data, it communicates the results to the cloud (again through a message broker via MQTT) for deeper historical analysis through big data analytics tools.**

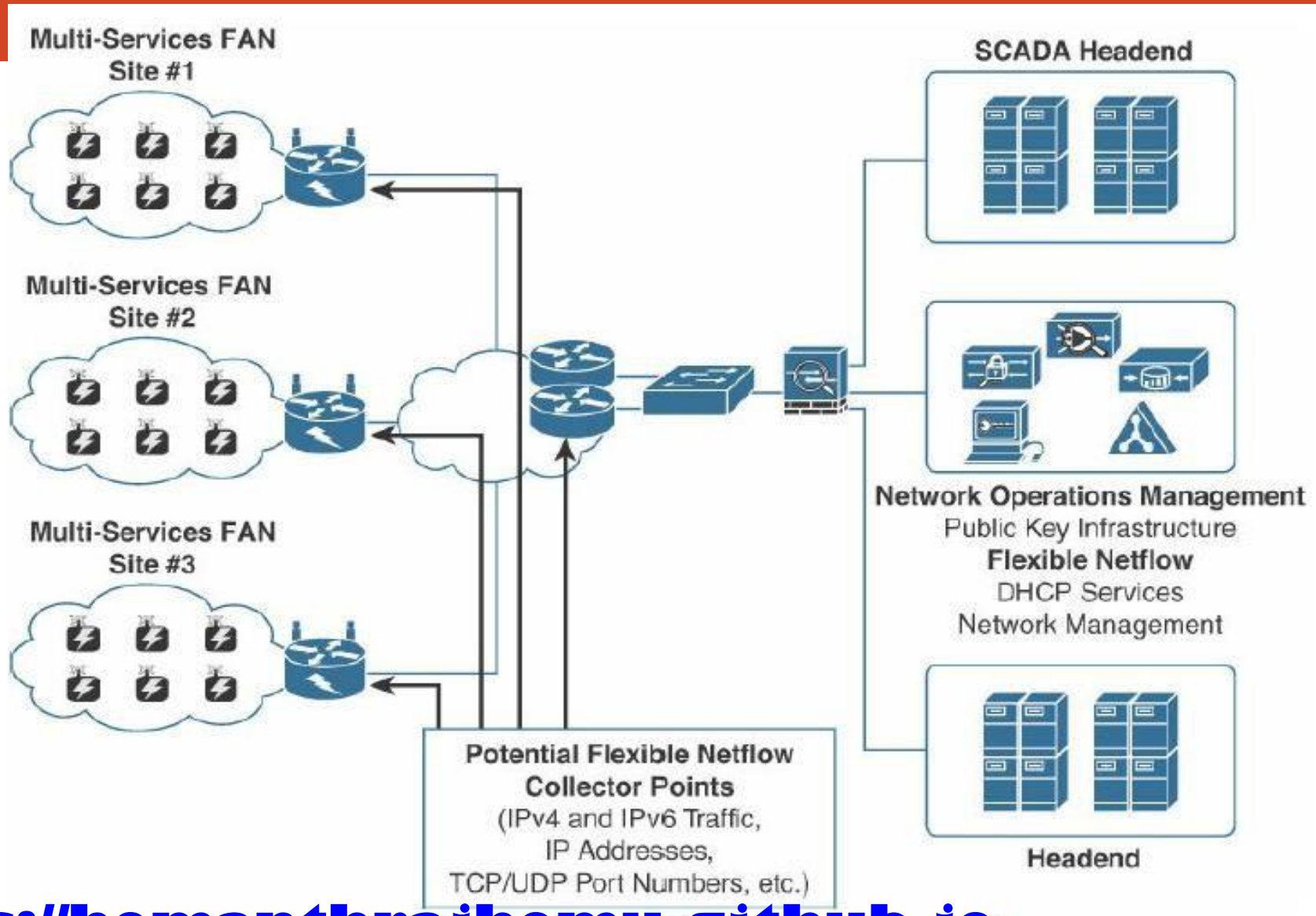
Network Analytics:

- Data analytics systems that are concerned with finding patterns in the data generated by endpoints, 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 quickly identifies anomalies that suggest network problems due to sub optimal paths, intrusive malware, or excessive congestion.

Module – 4 Data and Analytics for IoT

Network Analytics:

- Figure shows field area network (FAN) traffic analytics performed on the aggregation router in a smart grid.



https://hemanthrajhemu.github.io

Module – 4 Data and Analytics for IoT

Network Analytics:

- Network analytics offer capabilities to cope with capacity planning for scalable IoT deployment as well as security monitoring in order to detect abnormal traffic volume and patterns (such as an unusual traffic spike for a normally quiet protocol) for both centralized or distributed architectures, such as fog computing.
- Consider that an IoT device sends its traffic to specific servers, either directly to an application or an IoT broker with the data payload encapsulated in a given protocol.
- This represents a pair of source and destination addresses, as well as application layer–dependent TCP or UDP port numbers, which can be used for network analytics.

Module – 4 Data and Analytics for IoT

Network Analytics:

- **One of the drivers of the adoption of an IP architectural framework for IoT is to leverage tools and processes largely known and deployed by Internet service providers (ISPs) as well as private corporate enterprise networks.**
- **To monitor network infrastructure, de facto industry standards and protocols allow pervasive characterization of IP traffic flows, including identification of source and/or destination addresses, data timing and volume, and application types within a network infrastructure .**

Network Analytics:

- For example, centralized routers or switches that aggregate subnetworks as well as nodes that are highly distributed and connect the last mile of the infrastructure can be used to collect flow information.
- After data is collected in a known format, it can be sent to an external network analytics tools that delivers unique services to network managers, like security and performance monitoring and capacity planning.

Network Analytics:

Benefits of flow analytics, in addition to other network management services, are as follows:

1. 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.**

Network Analytics:

Benefits of flow analytics, in addition to other network management services, are as follows:

2. 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.**

Network Analytics:

Benefits of flow analytics, in addition to other network management services, are as follows:

3. 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.**

Module – 4 Data and Analytics for IoT

Network Analytics:

4. 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.**
- **Security can be enforced by ensuring that no traffic is sent outside the scope of the IoT domain.**
- **For example, with a LoRaWAN gateway, there should be no reason to see traffic sent or received outside the LoRaWAN network server and network management system.**

Network Analytics:

5. 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 the last-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. .

Module – 4 Data and Analytics for IoT

Network Analytics:

6. 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.

Module – 4 Data and Analytics for IoT

Flexible NetFlow Architecture:

Key advantages of FNF are as follows:

1. **Flexibility, scalability, and aggregation of flow data**
2. **Ability to monitor a wide range of packet information and produce new information about network behavior**
3. **Enhanced network anomaly and security detection**
4. **User-configurable flow information for performing customized traffic identification and ability to focus and monitor specific network behavior**
5. **Convergence of multiple accounting technologies into one accounting mechanism**

Module – 4 Data

Flexible

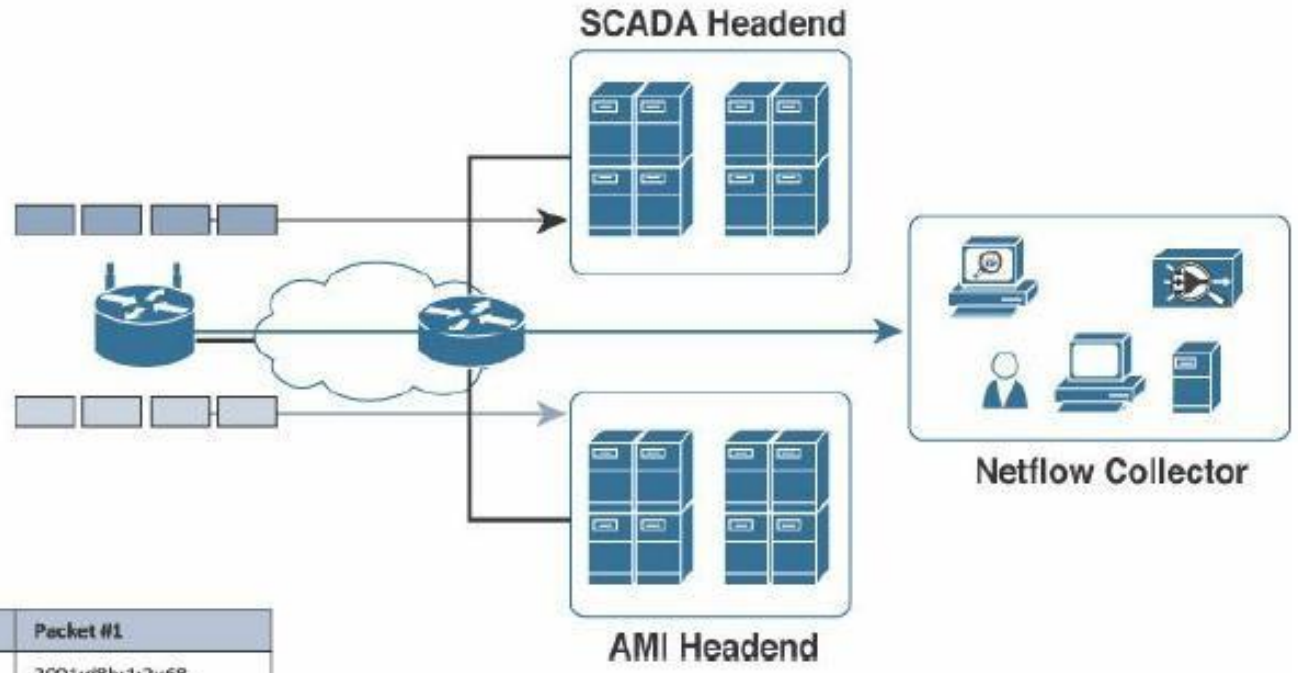
NetFlow

Architecture:

FNF Components:

FNF has the following main components, as shown in Figure :

First packet of a flow will create the Flow entry using the Key Fields
Remaining packets of this flow will only update statistics (bytes, counters, timestamps)



Key Fields	Packet #1
Source IP	2001:d8b:1:2::68
Destination IP	2001:d8b:3:4:250
Source Port	27356
Destination Port	4059
Layer 3 protocol	17
TOS byte	0
Ingress Interface	WPAN 0
Non-Key Fields	
Length	512
Packets count	23

FNF Flows

Key Fields	Packet #2
Source IP	192.168.27.1
Destination IP	10.23.42.56
Source Port	45324
Destination Port	20000
Layer 3 protocol	6
TOS byte	0
Ingress Interface	Ethernet 0
Non-Key Fields	
Length	768
Packet counts	23

FNF Cache

Src. IP	Dest. IP	Src. Port	Dest. Port	Layer 3 Prot.	TOS Byte	Ingress Intf.	Timestamps	Length	Packets
2001:d8b:1:2::68	2001:d8b:3:4:250	27356	4059	17	0	Wpan 0		512	23

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 (used to create a flow, unique per flow record: match statement) and non-key fields (collected with the flow as attributes or characteristics of a flow) within the cache.

FNF Flow Monitor (NetFlow 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.
- The Flow Monitor includes various cache characteristics, including timers for exporting, the size of the cache, and, if required, the packet sampling rate.

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.
- A typical predefined record aggregates flow data and allows users to target common applications for NetFlow.

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:

- In the context of multiservice IoT networks, it is recommended that 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 (although not between IoT devices).
- FNF can also be configured on the last-mile gateway or fog nodes to provide more granular visibility.

Module – 4 Data and Analytics for IoT

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.**
- **The added cost of increasing bandwidth thus needs to be reviewed, especially if the backhaul network uses cellular or satellite communications.**

Module – 4 Securing IoT

A Brief History of OT Security:

- **More than in most 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.**
- **While there are certainly traditional IT-related security threats in industrial environments, it is the physical manifestations and impacts of the OT security incidents that capture media attention and elicit broad-based public concern.**

Module – 4 Securing IoT

A Brief History of OT Security:

- **One example of a reported incident where physical damage was caused by a cybersecurity attack is the Stuxnet malware that damaged uranium enrichment systems in Iran.**
- **Another example is an event that damaged a furnace in a German smelter.**
- **In both incidents, multiple steps led to the undesirable outcomes.**
- **Many of the security policies and mitigation procedures that were in place went unheeded.**

Module – 4 Securing IoT

A Brief History of OT Security:

- **In addition to physical damage, operational interruptions have occurred in OT environments due to cybersecurity incidents.**
- **For example, in 2000, the sewage control system of Maroochy Shire in Queensland, Australia, was accessed remotely, and it released 800,000 liters of sewage into the surrounding waterways.**
- **In 2015, the control systems of the Ukrainian power distribution operator Kyiv Oblenergo were remotely accessed by attackers, causing an outage that lasted several hours and resulted in days of degraded service for thousands of customers.**

Module – 4 Securing IoT

A Brief History of OT Security:

- **Historically, attackers were skilled individuals with deep knowledge of technology and the systems they were attacking.**
- **However, as technology has advanced, tools have been created to make attacks much easier to carry out.**
- **To further complicate matters, these tools have become more broadly available and more easily obtainable.**

Module – 4 Securing IoT

A Brief History of OT Security:

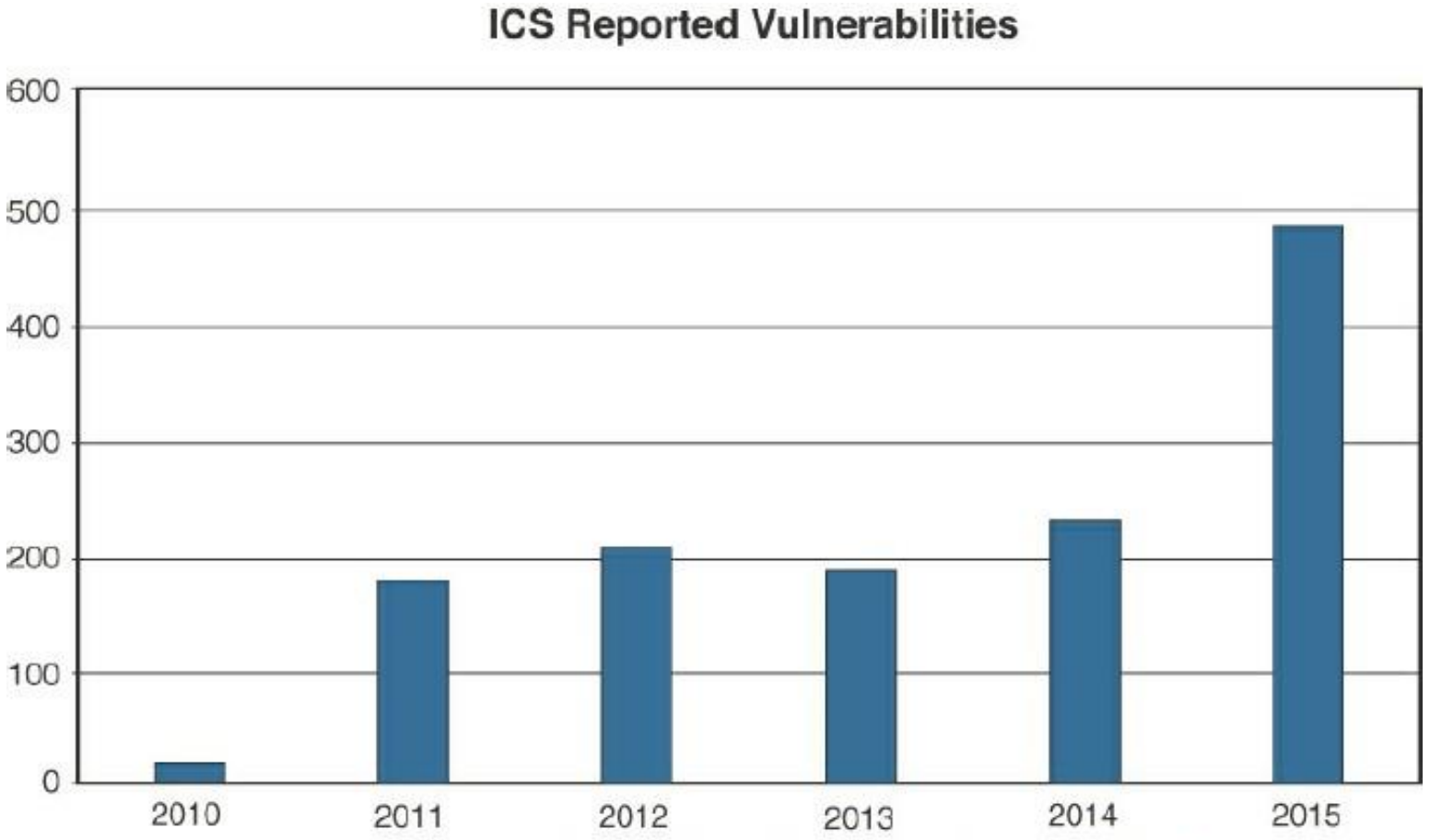
- **Compounding this problem, many of the legacy protocols used in IoT environments are many decades old, and there was no thought of security when they were first developed.**
- **This means that attackers with limited or no technical capabilities now have the potential to launch cyber attacks, greatly increasing the frequency of attacks and the overall threat to end operators.**
- **The isolation between industrial networks and the traditional IT business networks has been referred to as an "air gap" suggesting that there are no links between the two.**

A Brief History of OT Security:

- **Most of the industrial control systems deployed today, their components, and the limited associated security elements were designed when adherence to published and open standards were rare.**
- **The proprietary nature of these systems meant that threats from the outside world were unlikely to occur and were rarely addressed.**

Module – 4 Securing IoT

A Brief History of OT Security:



History of Vulnerability Disclosures in Industrial Control Systems Since 2010 (US Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) https://ics-cert.us-cert.gov).

https://hemanthrajhemu.github.io

Module – 4 Securing IoT

Common Challenges in OT Security: The security challenges faced in IoT are by no means new and are not limited to specific industrial environments.

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, and the assumption that attackers lacked sufficient knowledge to carry out security attacks**

Module – 4 Securing IoT

Common Challenges in OT Security: Erosion of Network Architecture:

- **The challenge, and the biggest threat to network security, is standards and best practices either being misunderstood or the network being poorly maintained.**
- **In fact, from a security design perspective, it is better to know that communication paths are insecure than to not know the actual communication paths.**
- **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.**

Module – 4 Securing IoT

Common Challenges in OT Security: Erosion of Network Architecture:

- These uncontrolled or poorly controlled OT network evolutions have, in many cases, over time led to weak or inadequate network and systems security.
- In many industries, the control systems consist of packages, skids, or components that are self-contained and may be integrated as semi-autonomous portions of the network.
- These packages may not be as fully or tightly integrated into the overall control system, network management tools, or security applications, resulting in potential risk.

Module – 4 Securing IoT

Common Challenges in OT 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 example, in a power utility environment, it is not uncommon to have racks of old mechanical equipment still operating alongside modern intelligent electronic devices (IEDs).**
- **From a security perspective, this is potentially dangerous as many devices may have historical vulnerabilities or weaknesses that have not been patched and updated, or it may be that patches are not even available due to the age of the equipment.**

Module – 4 Securing IoT

Common Challenges in OT Security: Pervasive Legacy Systems:

- **communication methods and protocols may be generations old and must be interoperable with the oldest operating entity in the communications path.**
- **This includes switches, routers, firewalls, wireless access points, servers, remote access systems, patch management, and network management tools.**
- **All of these may have exploitable vulnerabilities and must be protected.**

Common Challenges in OT Security: Insecure Operational Protocols:

- **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.**

Module – 4 Securing IoT

Common Challenges in OT Security: Insecure Operational Protocols:

Modbus:

- **Modbus is commonly found in many industries, such as utilities and manufacturing environments, and has multiple variants (for example, serial,TCP/IP).**
- **It was created by the first programmable logic controller (PLC) vendor, Modicon, and has been in use since the 1970s.**
- **It is one of the most widely used protocols in industrial deployments, and its development is governed by the Modbus Organization.**

Module – 4 Securing IoT

Common Challenges in OT Security: Insecure Operational Protocols:

Modbus:

- **Authentication of communicating endpoints was not a default operation because it would allow an inappropriate source to send improper commands to the recipient.**
- **For example, for a message to reach its destination, nothing more than the proper Modbus address and function call (code) is necessary.**
- **Some older and serial-based versions of Modbus communicate via broadcast.**
- **The ability to curb the broadcast function does not exist in some versions.**

Common Challenges in OT Security: Insecure Operational Protocols:

Modbus:

- **Validation of the Modbus message content is also not performed by the initiating application.**
- **Instead, Modbus depends on the network stack to perform this function.**
- **This could open up the potential for protocol abuse in the system.**

Common Challenges in OT Security: Insecure Operational Protocols:

DNP3 (Distributed Network Protocol):

- **DNP3 is found in multiple deployment scenarios and industries.**
- **It is common in utilities and is also found in discrete and continuous process systems. Like many other ICS/SCADA protocols, it was intended for serial communication between controllers and simple IEDs.**
- **In the case of DNP3, participants allow for unsolicited responses, which could trigger an undesired response.**

Common Challenges in OT Security: Insecure Operational Protocols:

DNP3 (Distributed Network Protocol):

- **The missing security element here is the ability to establish trust in the system's state and thus the ability to trust the veracity of the information being presented.**
- **This is akin to the security flaws presented by Gratuitous ARP messages in Ethernet networks, which has been addressed by Dynamic ARP Inspection (DAI) in modern Ethernet switches.**

Common Challenges in OT Security: Insecure Operational Protocols:

ICCP (Inter-Control Center Communications Protocol):

- **ICCP is a common control protocol in utilities across North America that is frequently used to communicate between utilities.**
- **Given that it must traverse the boundaries between different networks, it holds an extra level of exposure and risk that could expose a utility to cyber attack.**

Common Challenges in OT Security: Insecure Operational Protocols:

ICCP (Inter-Control Center Communications Protocol):

- **Initial versions of ICCP had several significant gaps in the area of security.**
- **One key vulnerability is that the system did not require authentication for communication.**
- **Second, encryption across the protocol was not enabled as a default condition, thus exposing connections to man-in-the-middle (MITM) and replay attacks.**

Common Challenges in OT Security: Insecure Operational Protocols:

OPC (OLE for Process Control):

- **OPC is based on the Microsoft interoperability methodology Object Linking and Embedding (OLE).**
- **This is an example where an IT standard used within the IT domain and personal computers has been leveraged for use as a control protocol across an industrial network.**

Module – 4 Securing IoT

Common Challenges in OT Security: Insecure Operational Protocols:

OPC (OLE for Process Control):

- **In industrial control networks, OPC is limited to operation at the higher levels of the control space, with a dependence on Windows-based platforms.**
- **Concerns around OPC begin with the operating system on which it operates.**
- **Many of the Windows devices in the operational space are old, not fully patched and at risk due to a plethora of well-known vulnerabilities.**

Common Challenges in OT Security: Insecure Operational Protocols:

OPC (OLE for Process Control):

- **Particular concern with OPC is the dependence on the Remote Procedure Call (RPC) protocol, which creates two classes of exposure.**
- **The first requires to clearly understand the many vulnerabilities associated with RPC, and the second requires to identify the level of risk these vulnerabilities bring to a specific network.**

Module – 4 Securing IoT

Common Challenges in OT Security: Insecure Operational Protocols:

International Electrotechnical Commission (IEC) Protocols:

- **The IEC 61850 standard was created to allow vendor-agnostic engineering of power utility systems, which would, in turn, allow interoperability between vendors and standardized communication protocols.**
- **Three message types were initially defined:**
- **MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Event), and SV (Sampled Values).**
- **Web services was a fourth protocol that was added later**

Common Challenges in OT Security: Insecure Operational Protocols:

International Electrotechnical Commission (IEC) Protocols:

MMS (61850-8.1):

- **MMS is a client/server protocol that leverages TCP/IP and operates at Layer 3.**
- **It provides the same functionality as other SCADA protocols, such as IEC 60870 and Modbus.**

Common Challenges in OT Security: Insecure Operational Protocols:

International Electrotechnical Commission (IEC) Protocols:

GOOSE (61850-8.1):

- **GOOSE is a Layer 2 protocol that operates via multicast over Ethernet.**
- **It allows IEDs to exchange data "horizontally," between bays and between substations, especially for interlocking, measurement, and tripping signals.**

Common Challenges in OT Security: Insecure Operational Protocols:

International Electrotechnical Commission (IEC) Protocols:

SV (61850-9-2):

- **SV is a Layer 2 protocol that operates via multicast over Ethernet.**
- **It carries voltage and current samples, typically on the process bus, but it can also flow over the station bus.**

Both GOOSE and SV operate via a publisher/subscriber model, with no reliability mechanism to ensure that data has been received.

Module – 4 Securing IoT

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

Risk assessment frameworks:

OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) from the Software Engineering Institute at Carnegie Mellon University.

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

Module – 4 Securing IoT

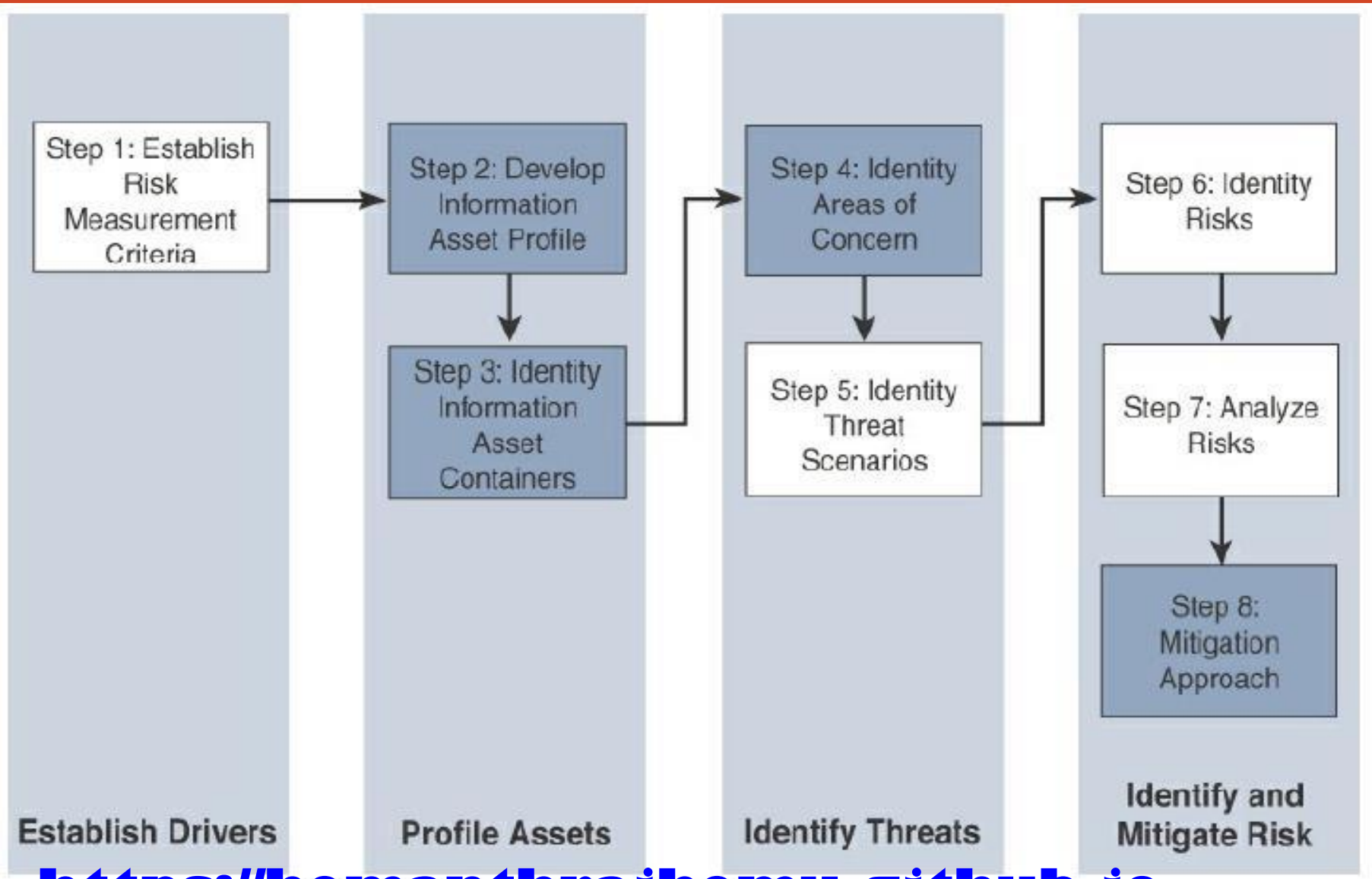
Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **OCTAVE has undergone multiple iterations.**
- **The version focuses on is 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.**
- **This approach and the assumptions it makes are quite appropriate, given that many operational technology areas are similarly lacking in security-focused human assets.**

Module – 4 Securing IoT

OCTAVE:



Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **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.**

Module – 4 Securing IoT

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **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.**
- **Within this asset profile, process are multiple substages that complete the definition of the assets**

Module – 4 Securing IoT

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **There are judgment-based attributes such as prioritization.**
- **Rather than simply assigning an arbitrary ranking, the system calls for a justification of the prioritization.**
- **With an understanding of the asset attributes, particularly the technical components, appropriate threat mitigation methods can be applied.**
- **With the application of risk assessment, the level of security investment can be aligned with that individual asset.**

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **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.**

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- The fourth step is to identify areas of concern.
- 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.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **Closely related is the fifth step, where 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.**

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- **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.**

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- 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.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

OCTAVE:

- 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.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

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.
- Like OCTAVE, it also allows for non-malicious actors as a potential cause for harm, but it goes to greater lengths to emphasize the point.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

FAIR:

- 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.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

FAIR:

- **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.**

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

FAIR:

- 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.

Common Challenges in OT Security: Formal Risk Analysis Structures: OCTAVE and FAIR:

FAIR:

- 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.
- Finally, competitive advantage and reputation are the least measurable.

The Phased Application of Security in an Operational Environment:

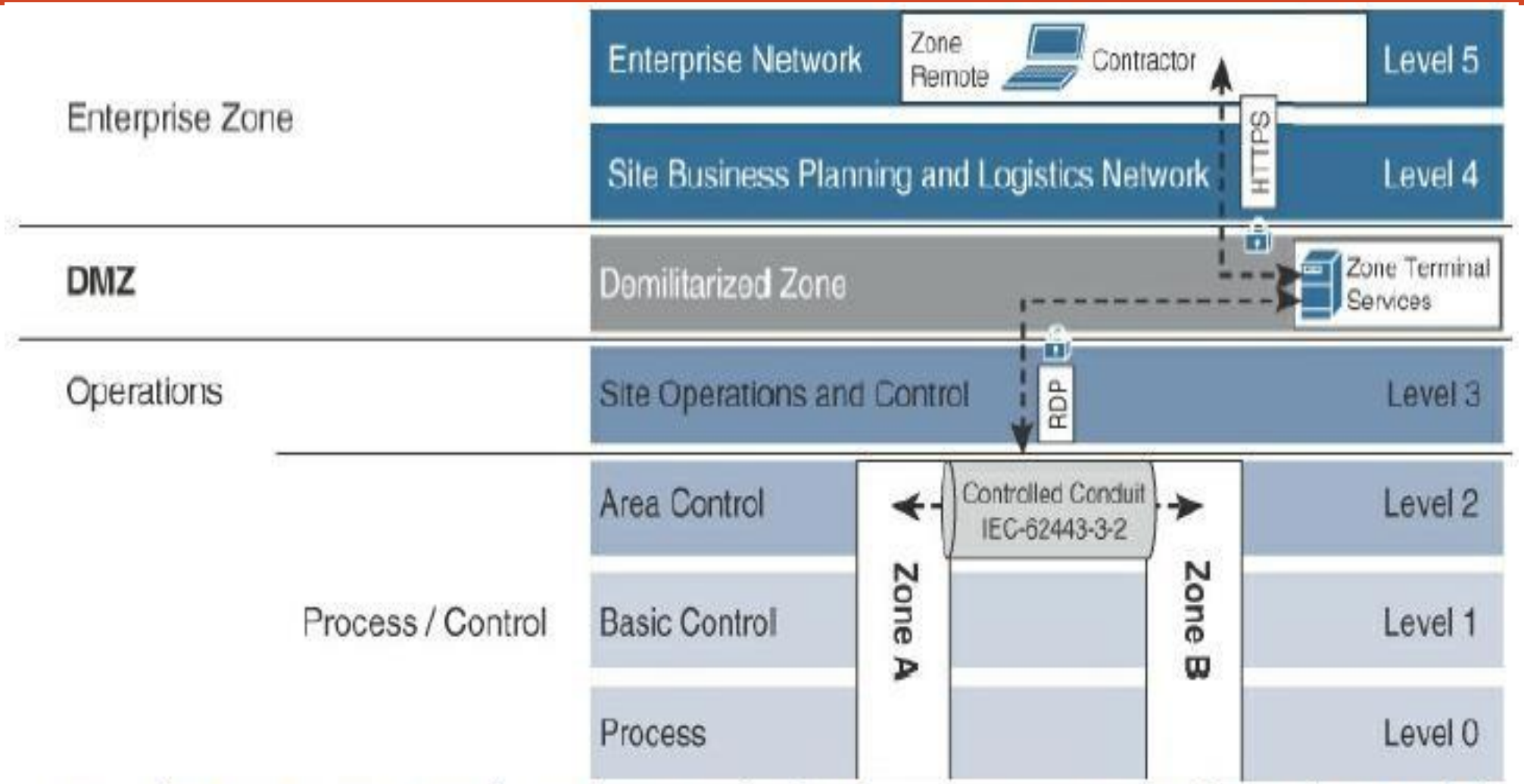
Secured Network Infrastructure and Assets:

- 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.
- It is a basic tenet of ISA99 and IEC 62443 that functions should be segmented into zones (cells) and that communication crossing the boundaries of those zones should be secured and controlled through the concept of conduits.

Module – 4 Securing IoT

The Phased Application of Security in an Operational Environment:

Secured Network Infrastructure and Assets:



Security Between Levels and Zones in the Process Control Hierarchy Model

https://hemanthrajhemu.github.io

Module – 4 Securing IoT

Secured Network Infrastructure and Assets:

- Normal network discovery processes can be highly problematic for older networking equipment.
- In fact, the discovery process in pursuit of improved safety, security, and operational state can result in degradation of all three.
- Given that condition, the network discovery process may require manual inspection of physical connections, starting from the highest accessible aggregation point and working all the way down to the last access layer.
- This discovery activity must include a search for wireless access points.

Secured Network Infrastructure and Assets:

- Normally, in an IT environment, the very first stage of discovery is focused on assets connected to the network.
- Assets remain critical, but from an efficiency and criticality perspective, it is generally recommended to find data paths into and between zones (cells) rather than the serial links between devices within a zone.

Module – 4 Securing IoT

Secured Network Infrastructure and Assets:

- Once the network is physically mapped, the next step is to perform a connectivity analysis through the switch and router ARP tables and DHCP requests within the network infrastructure.
- This should help further illuminate connectivity, good or bad, that has occurred.
- Firewall and network infrastructure data can contribute to understanding what devices are talking to other devices and the traffic paths over which this is done.

Secured Network Infrastructure and Assets:

- Network infrastructure should also provide the ability to secure communications between zones via secured conduits (see Figure).
- The primary method is encrypted communications in the form of virtual private networks (VPNs).
- VPNs can come in multiple forms, such as site-to-site, which would be appropriate between a utility substation and a control center, or perhaps in cell-to-cell communications.

FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

Quick Links for Faster Access.

CSE 8th Semester - https://hemanthrajhemu.github.io/CSE8/

ISE 8th Semester - https://hemanthrajhemu.github.io/ISE8/

ECE 8th Semester - https://hemanthrajhemu.github.io/ECE8/

8th Semester CSE - TEXTBOOK - NOTES - QP - SCANNER & MORE

17CS81 IOT - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS81/

17CS82 BDA - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS82/

17CS832 UID - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS832/

17CS834 SMS - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS834/

8th Semester Computer Science & Engineering (CSE)

8th Semester CSE Text Books: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Text-Book.html

8th Semester CSE Notes: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Notes.html

8th Semester CSE Question Paper: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Paper.html

8th Semester CSE Scanner: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Scanner.html

8th Semester CSE Question Bank: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Bank.html

8th Semester CSE Answer Script: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Answer-Script.html

Contribution Link:

https://hemanthrajhemu.github.io/Contribution/

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates:
https://telegram.me/joinchat/AAAAAFTtp8kuvCHALxuMaQ

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/