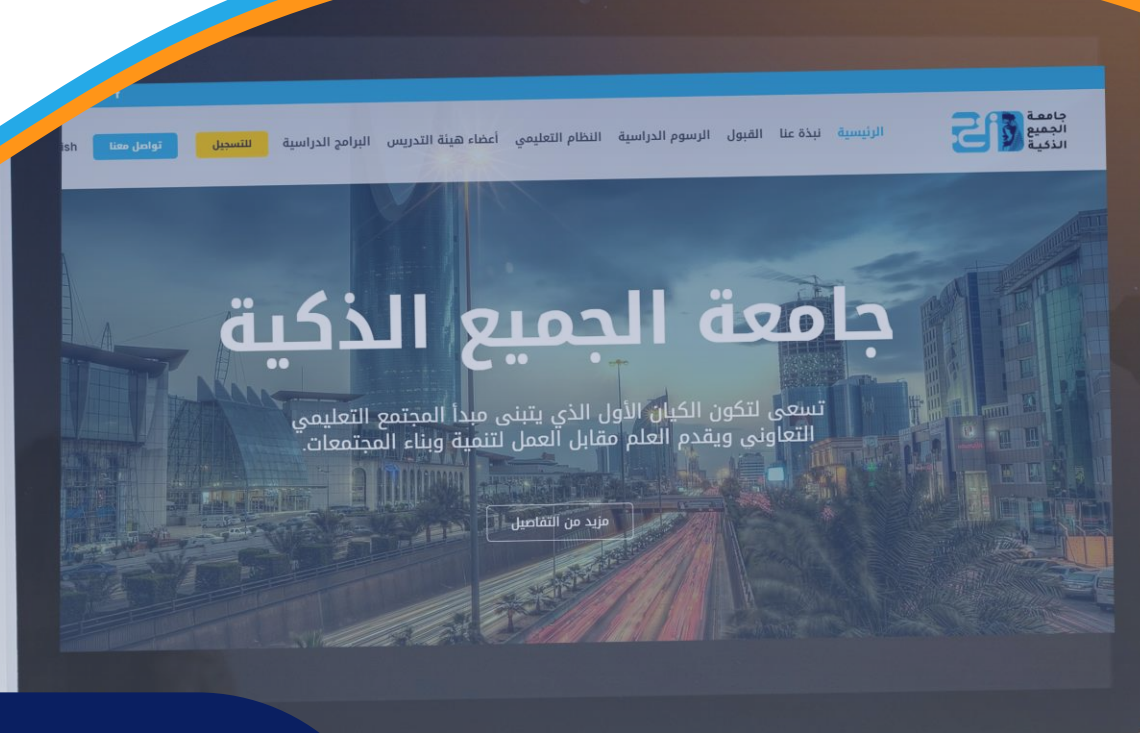




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



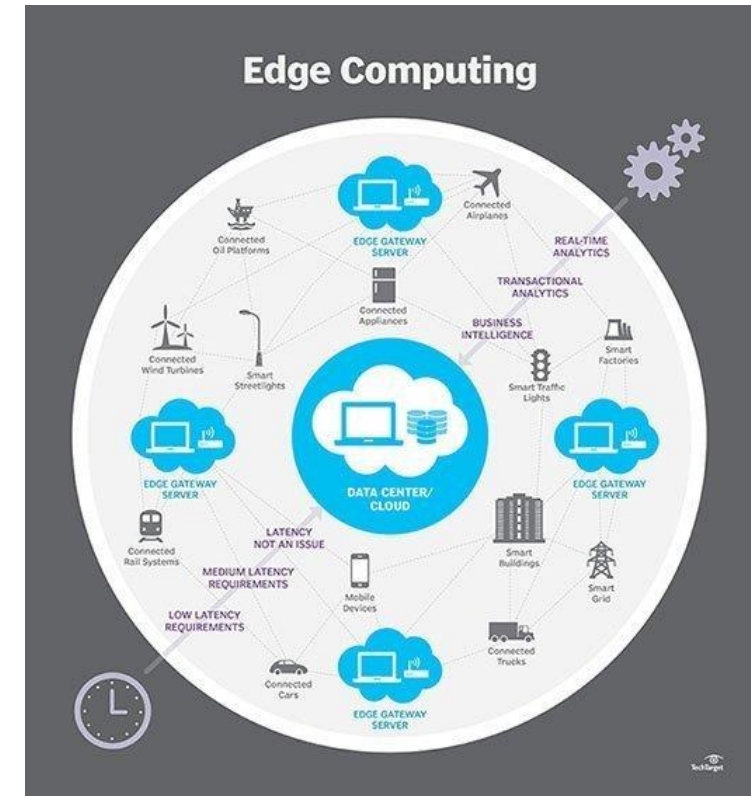
IoT Security
Prepared by Dr. Abdelrhman Ahmed
Lecture 6

Security of IoT Edge Network

Chapter 06

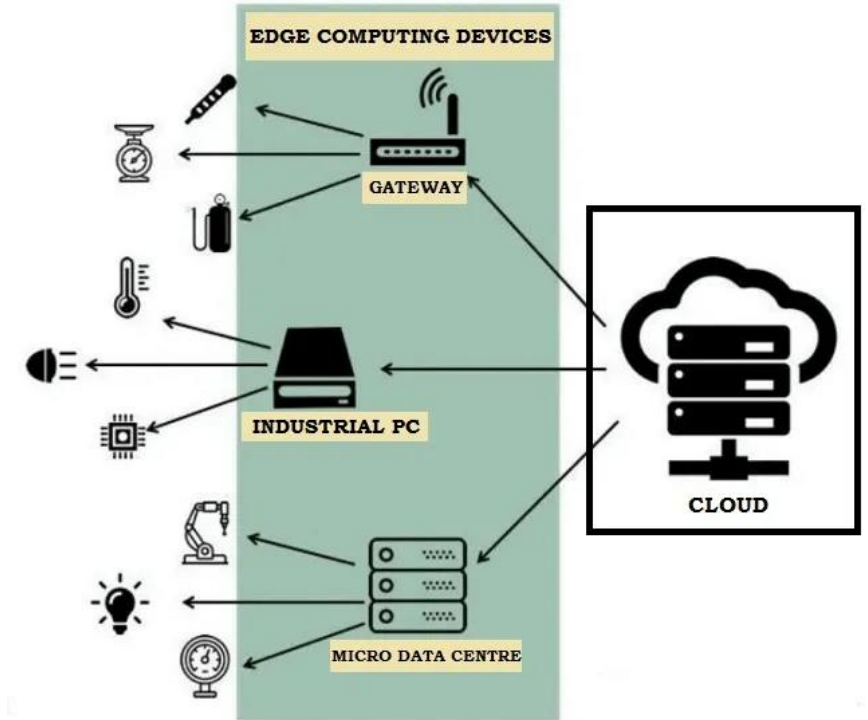
Edge Computing

Edge computing takes place at or near the physical location of either the user or the source of the data. By placing computing services closer to these locations, users benefit from faster, more reliable services with better user experiences, while companies benefit by being better able to support latency-sensitive applications, identify trends, and offer better products and services. Edge computing is one way that a company can use and distribute a common pool of resources across a large number of locations to help scale centralized infrastructure to meet the needs of increasing numbers of devices and data.



The Internet of Things (IoT) is made up of smart devices connected to a network—sending and receiving large amounts of data to and from other devices—which produces a large amount of data to be processed and analyzed.

Edge computing, a strategy for computing on the location where data is collected or used, allows IoT data to be gathered and processed at the edge, rather than sending the data back to a data center or cloud. Together, IoT and edge computing are powerful ways to rapidly analyze data in real time.



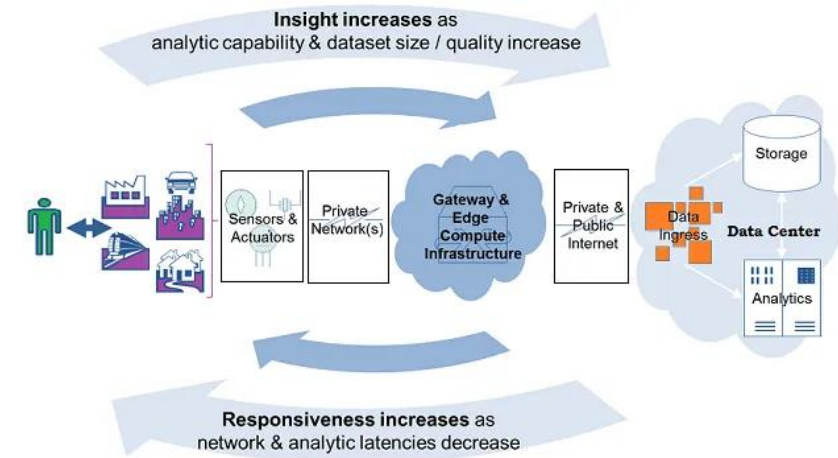
Edge devices are physical hardware located in remote locations at the edge of the network with enough memory, processing power, and computing resources to collect data, process that data, and execute it in almost real time with limited help from other parts of the network.

An IoT device is a physical object that has been connected to the internet and is the source of the data. An edge device is where the data is collected and processed.

Edge devices can be considered part of the IoT when the object has enough storage and compute to make low-latency decisions and process data in milliseconds. The terms IoT device and edge device are sometimes used interchangeably.

IoT benefits from having compute power closer to where a physical device or data source actually exists. In order for the data produced by IoT devices to react faster or mitigate issues, it needs to be analyzed at the edge, rather than traveling back to a central site before that analysis can take place. Edge computing is a local source of processing and storage for the data and computing needs of IoT devices.

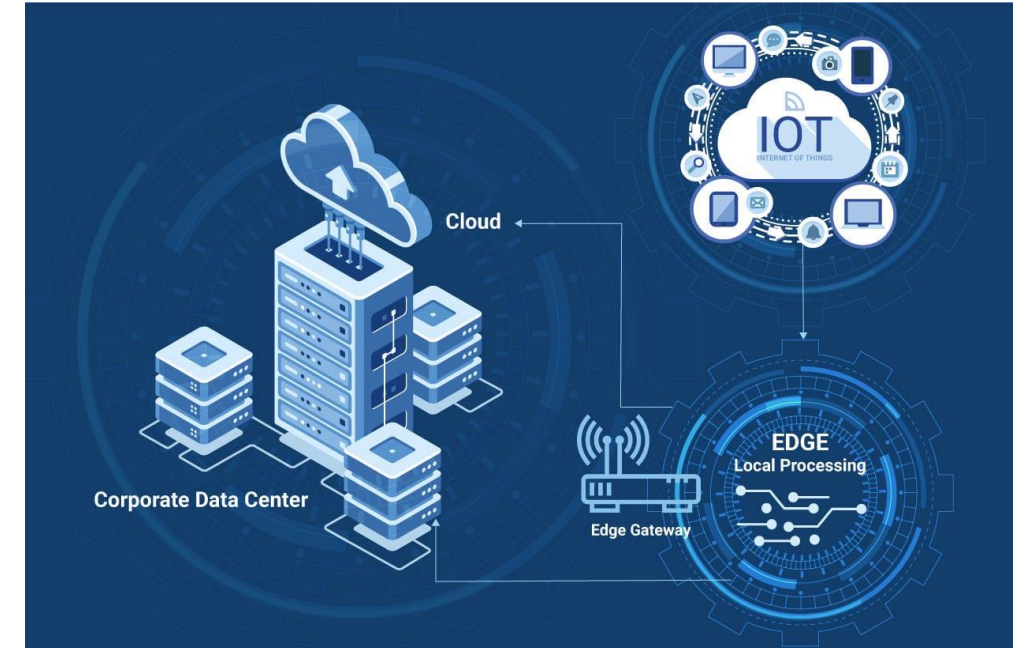
An IoT gateway can send data from the edge back to the cloud or centralized datacenter, or to the edge systems to be processed locally.



The Internet-of-Things (IoT) edge is where sensors and devices communicate real-time data to a network. IoT edge computing solves latency issues associated with the cloud, as data is processed closer to its point of origin. Along with reduced latency, IoT edge architecture brings enhanced safety and a smoother end-user experience.

On a high-throughput network, such as 5G, IoT edge can be used to process large amounts of data nearly instantaneously, creating a more immersive, comprehensive experience for the user. At the same time, even when relatively small sums of data are transmitted, IoT edge can make machines and other devices that impact human safety work faster, keeping operators and others safer.

- Reduced latency of communication between IoT devices and the central IT networks.
- Faster response times and increased operational efficiency.
- Improved network bandwidth.
- Continued systems operation offline when a network connection is lost.
- Local data processing, aggregation, and rapid decision-making via analytics algorithms and machine learning.



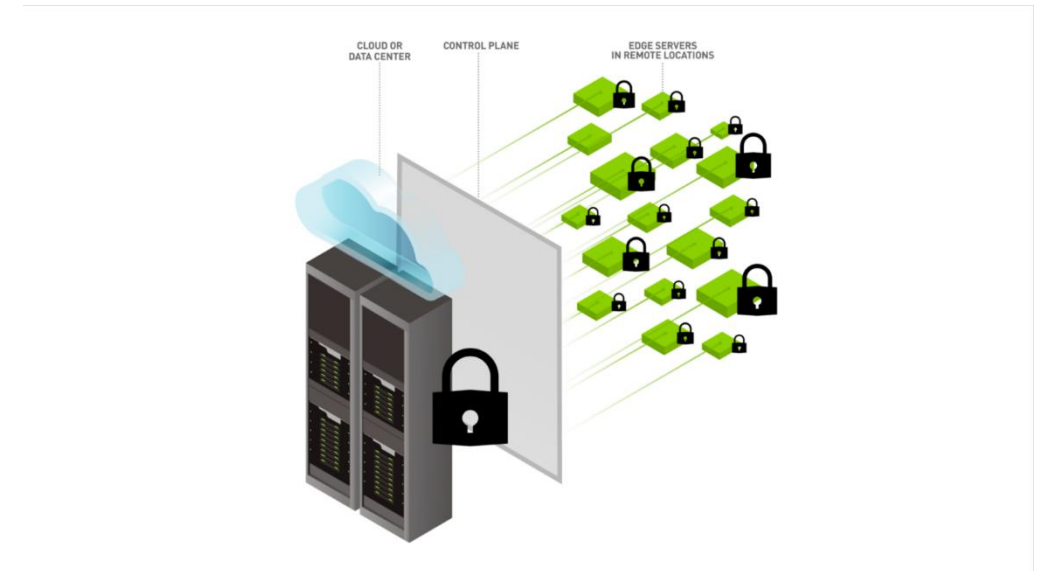
There are many security challenges related to IoT devices that are not as frequent as other types of devices. Some of these challenges are due to the low-cost and high-volume nature of these devices, and some are due to the processes and quickly changing technologies that are used to build them. Key challenges include:

- **Not all device manufacturers follow secure development practices.**
- **Not all devices allow the installation of software updates, which results in obsolete software running on them.**
- **The dynamic and rapid nature of the market results in the inconsistent installation of security patches, especially for retired or older versions of products.**
- **There are a few IoT security standards that are tailored to and implemented by IoT devices.**

Moreover, IoT devices at the edge face other challenges that apply only to these environments. Network connectivity at the edge is limited and sporadic, which makes it particularly hard to keep these devices up to date. At the humanitarian edge in which first responders and other emergency personnel operate, IoT devices may need to be deployed with little planning and in unsecured areas, which makes it easy for bad actors to interfere with their operation. At the tactical edge where military personnel execute missions, there will likely be malicious parties trying to gain access to these devices, which may be deployed over a large physical area with no direct supervision. The challenges inherent to IoT devices make these situations even more susceptible to attacks.

There are various ways to try to address these challenges:-

1. One option is to create or extend standards to improve the security of IoT devices, especially at the edge. This option would require IoT devices to implement these standards.
2. Another option is to assume that off-the-shelf IoT devices may be vulnerable or untrusted and to perform runtime monitoring and enforcement of security policies for access to these devices.



There are currently no widely accepted standards for authentication and authorization for IoT devices. A current standards proposal is Authentication and Authorization for Constrained Environments (ACE), which is a protocol being developed by a working group in the Internet Engineering Task Force (IETF). This group is adapting the existing OAuth 2.0 protocol, which is widely used by industry, to work with constrained devices with limited resources such as memory and processing power, as is the case with IoT devices. However, ACE does not take into consideration the challenges of humanitarian and tactical edge scenarios. SEI project, which we call “**Authentication and Authorization for IoT Devices in Disadvantaged Environments**” (AAIoT) focuses on addressing two gaps in the ACE protocol:

- (1) bootstrapping client and device credentials and
- (2) authorization revocation for compromised devices.

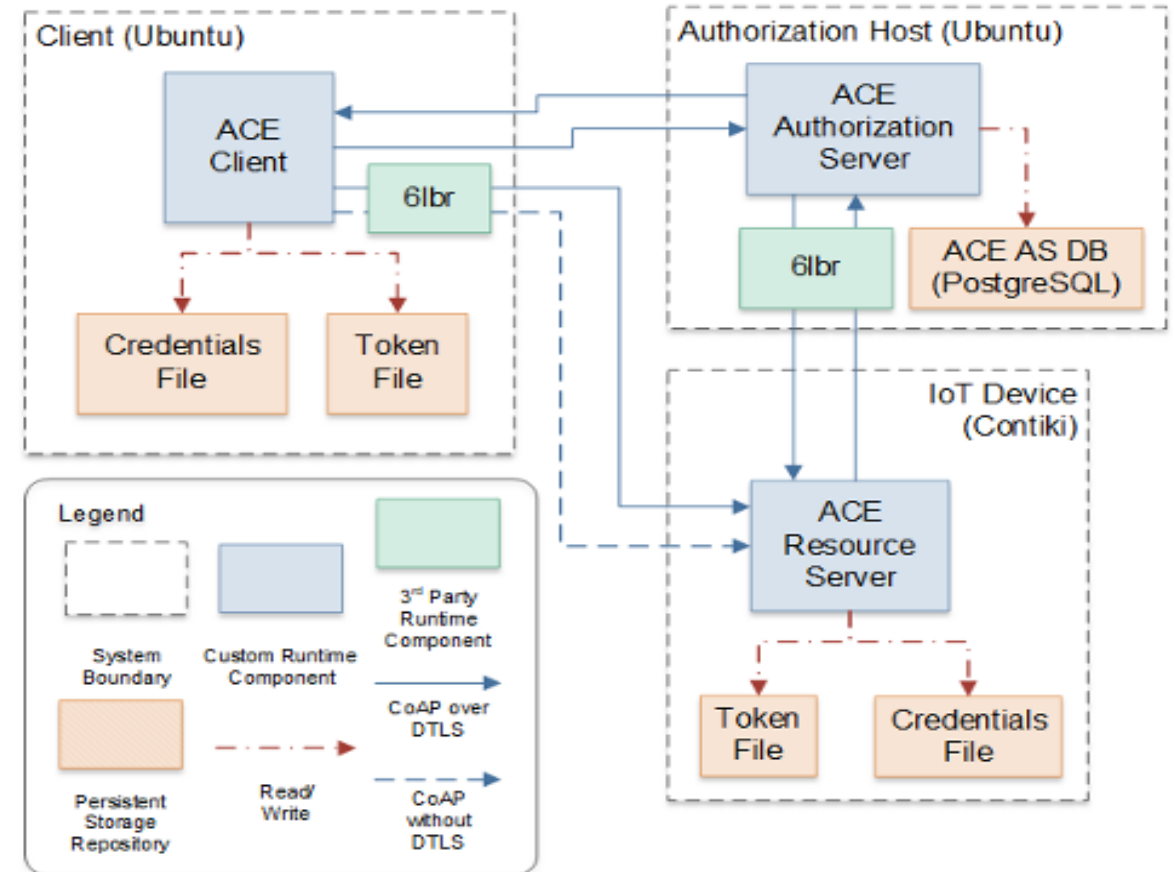
Bootstrapping of Credentials: By definition, bootstrapping of credentials is out of scope for ACE because of the heterogeneity of IoT devices. However, in disadvantaged environments, not including bootstrapping—exchanging credentials used to set up secure channels to communicate between devices—as an integral part of the process is risky because client and device capture and impersonation are likely and of high impact. In our solution, we defined a process where a QR code physically associated to a device contains a pre-shared key (PSK) that would be scanned during the pairing procedure between a client and a device, to securely generate and exchange keys. This process allows for pairing in the field but requires the IoT device to have the capability of receiving and storing new credentials.

Authorization Revocation: The ACE protocol assumes a stable connection between an IoT device and an authorization server, which authorizes third parties to access the IoT device by providing them with an access token that expires after a set time. In disadvantaged environments devices may be disconnected from an authorization server for longer periods of time. As a result, expiration times would need to be longer than usual to guarantee continued access. If an IoT device is compromised, however, it is critical to let all parties know that they should no longer have access to resources on that IoT device. We therefore extended ACE by defining a process for token revocation (which is not currently supported by ACE) in which the different parties can contact the authorization server to check if a token is still valid. This work has led to a new proposed extension to the ACE standard for token revocation.

Chapter 06

Standards-Based IoT Security Approach

One disadvantage of this approach is that an IoT device must implement the ACE protocol, and the extensions we defined, to take advantage of the solutions described. ACE is not yet an approved protocol, and even when it is, it could take a while for it to be widely adopted. Thus, other types of solutions that can work with commodity devices should also be considered. Validation for the ACE protocol extension through the AAIoT architecture.



When integrating commodity IoT devices into existing networks, there is a high likelihood that some of these devices may not implement any security protocols or may have unpatched vulnerabilities. It is even possible for one of these devices to be compromised at manufacturing time, and thus already have malicious code on it (i.e., a supply-chain risk). However, being able to use commodity IoT devices is an advantage at both the humanitarian and tactical edge for quick response to changing missions and environments. To protect the devices from external attacks and the networks from potential attacks from these commodity devices, a solution is needed that does not require changing the software on the device itself. The Kalki platform is one such solution.

Kalki is a software-defined IoT security platform that moves security enforcement to the network, thereby enabling the integration of commodity IoT devices, even if they are not fully trusted or configurable. Kalki leverages software-defined networking (SDN) concepts to act as a flexible intermediary between these devices and the network they are connecting to, ensuring that both are protected. This protection is done through the definition of a policy model for each device type, which can ensure that protections are customized to cover each device's specific vulnerabilities and shortcomings. The Kalki system also allows the user to easily change these policies if new vulnerabilities are found in a device model, or if the environment they must connect to requires specific policies.



Kalki uses information from the network traffic to and from a device, as well as from sensor data collected by a device, to detect both cyber and physical threats. This approach allows the system to detect potential tampering with an IoT device, as well as network-based attacks to or from a device. The security policies for a device can be combined to monitor for different types of attacks or unexpected states and react to stop an attacker.

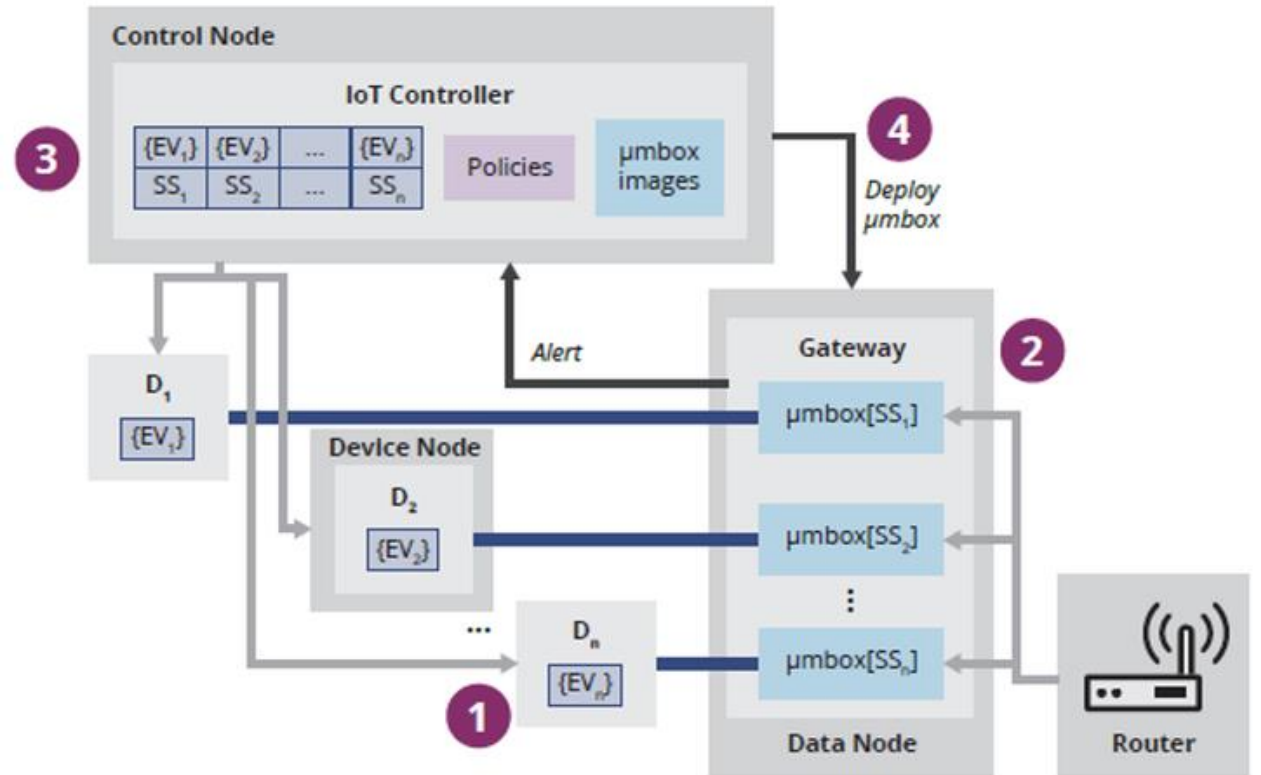
The network monitoring and security measures of the Kalki platform are handled by μ mboxes (pronounced “micro-m-boxes”), which are small software modules that implement network function virtualization (NFV) functionality. NFV allows software implementation of functions traditionally performed by dedicated hardware, such as a firewall or an intrusion-detection system (IDS). Moreover, NFV allows easy isolation and modularization of different types of network monitoring and reactions, which we encapsulate in μ mboxes.

In the Kalki platform, μ mboxes are implemented as containers that can be easily chained together to monitor for different threats in different ways or to protect a device or a network from different types of attacks. All traffic to and from a device goes through a set of μ mboxes deployed on a Kalki node called the data node. This set of μ mboxes can be different for each device, depending on its specifications.

Chapter 06

Components and Steps in Kalki Platform

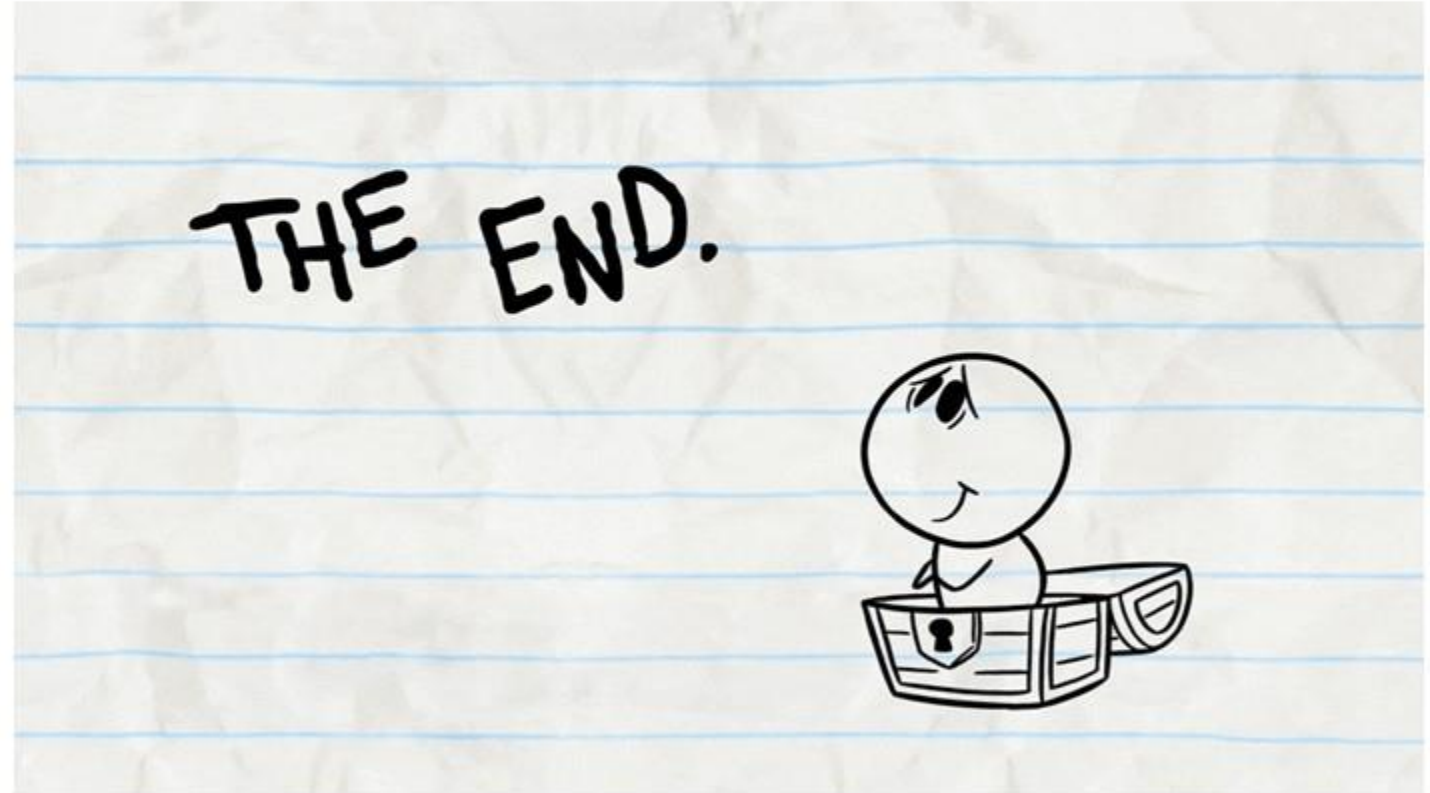
- (1) sensor data from IoT devices is monitored
- (2) network traffic is tunneled and monitored through μ boxes
- (3) control node maintains security state for each device and reacts through policies
- (4) control node modifies security postures in data node through μ box deployment changes when needed.



Besides having different μ mbboxes and general policies for each device, a Kalki node called the control node also maintains a security state for each device. By default, this state can be normal, suspicious, or under attack. Security policies can be associated to each security state for each device, so that a different set of μ mbboxes is deployed for each security state. The control node collects all information from μ mbboxes and sensor data and can trigger changes in the security state based on the configured policies. The control node sends commands to the data node to set up the proper μ mbboxes and sets of network rules based on the new security state.

The End

Please feel free to contact me anytime if you have any questions



جامعة القادسية
EVERYONE'S SMART UNIVERSITY