From Cloud to IoT Device Authenticity under Kubernetes Management

George Kornaros, Dimitris Bakoyiannis, Othon Tomoutzoglou and Marcello Coppola

Intelligent Systems and Computer Architecture Lab – ISCA

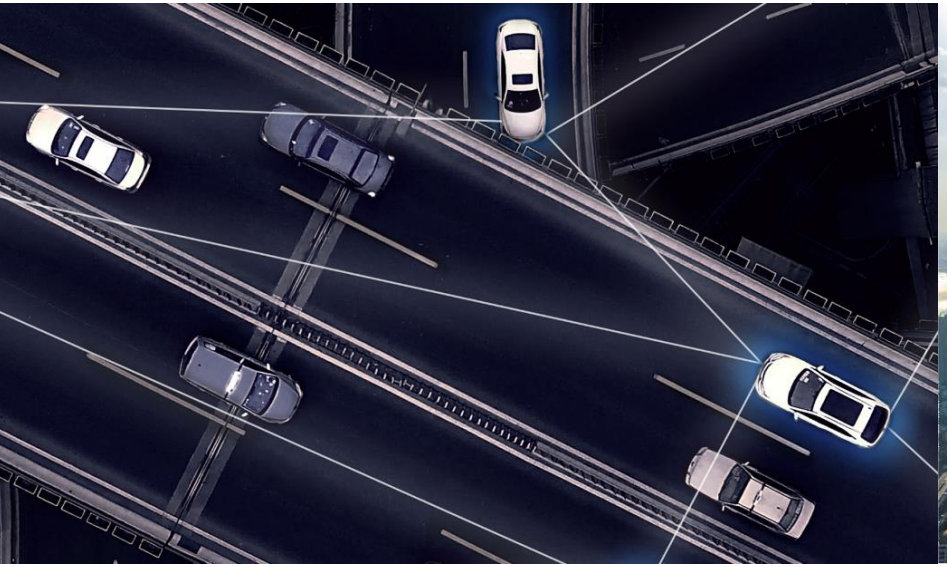
Hellenic Mediterranean University, Heraklion, Crete, GR

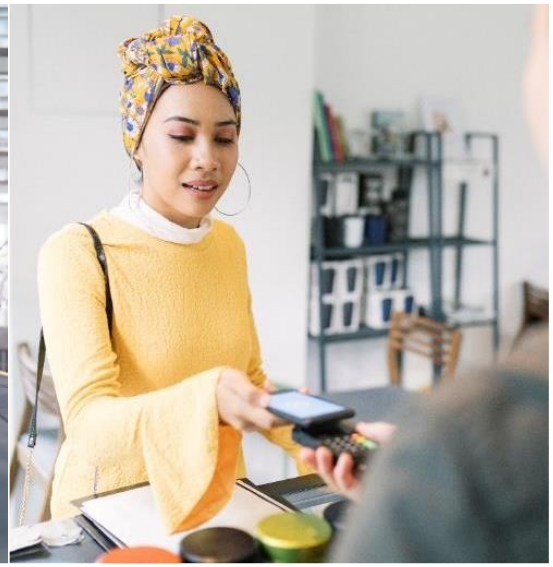
The 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS 2024)
Malmö, Sweden. September 2-5, 2024

Outline

- Challenges and background
- Services security challenges
- SPIFFE/SPIRE technology overview
- FLUIDOS – KubeEdge and SPIFFE integration
- Demonstration
- Conclusion

Billion of Embedded Devices



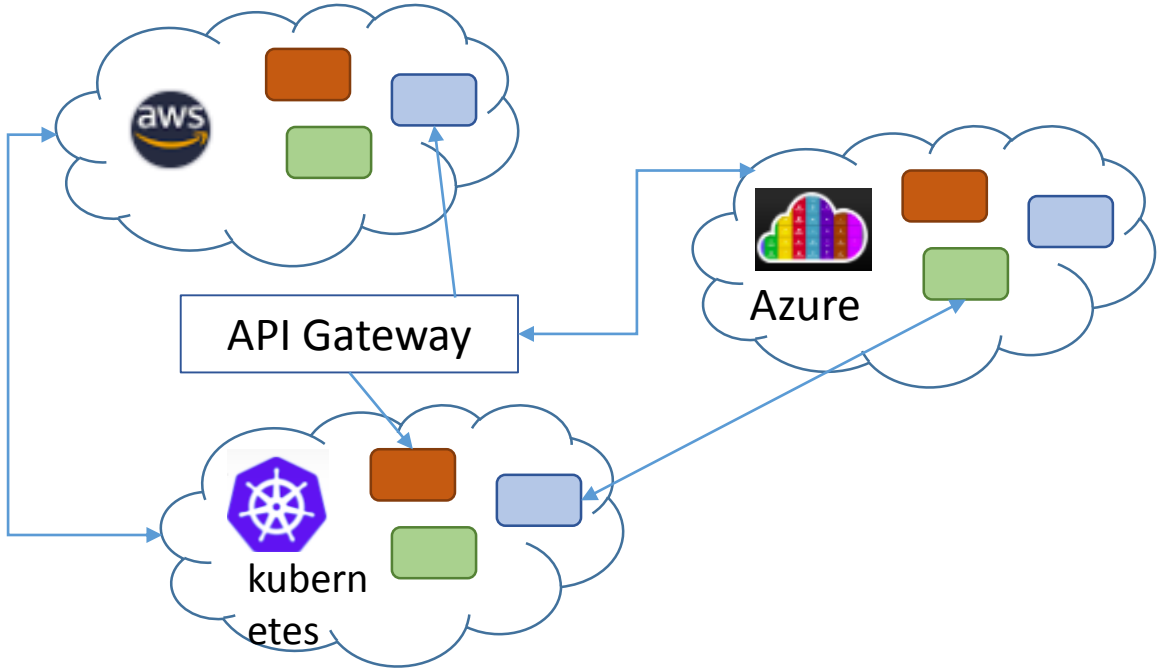
ADAS & Autonomous Vehicles, Cellular-V2X, connected cars

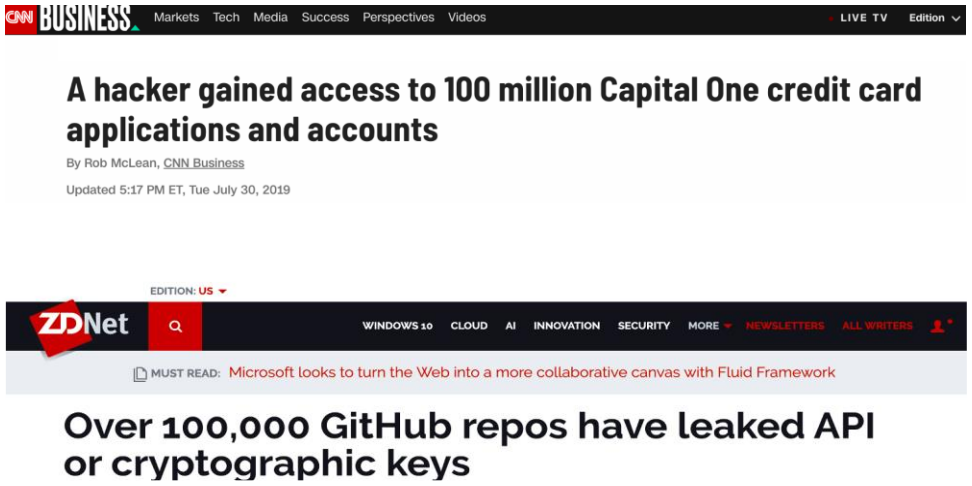
Secure anchors for Mobiles, Industrial, Communications, IoT & Edge nodes

Industrial & IoT Communication with IoT devices
- Trust provisioning
- Secure element communication

Digital signatures
- Secure boot, Industrial & IoT Firmware integrity for IoT devices
- Over-the-air updates, Firmware authentication, smart car access

Exploding Cross-service Communication



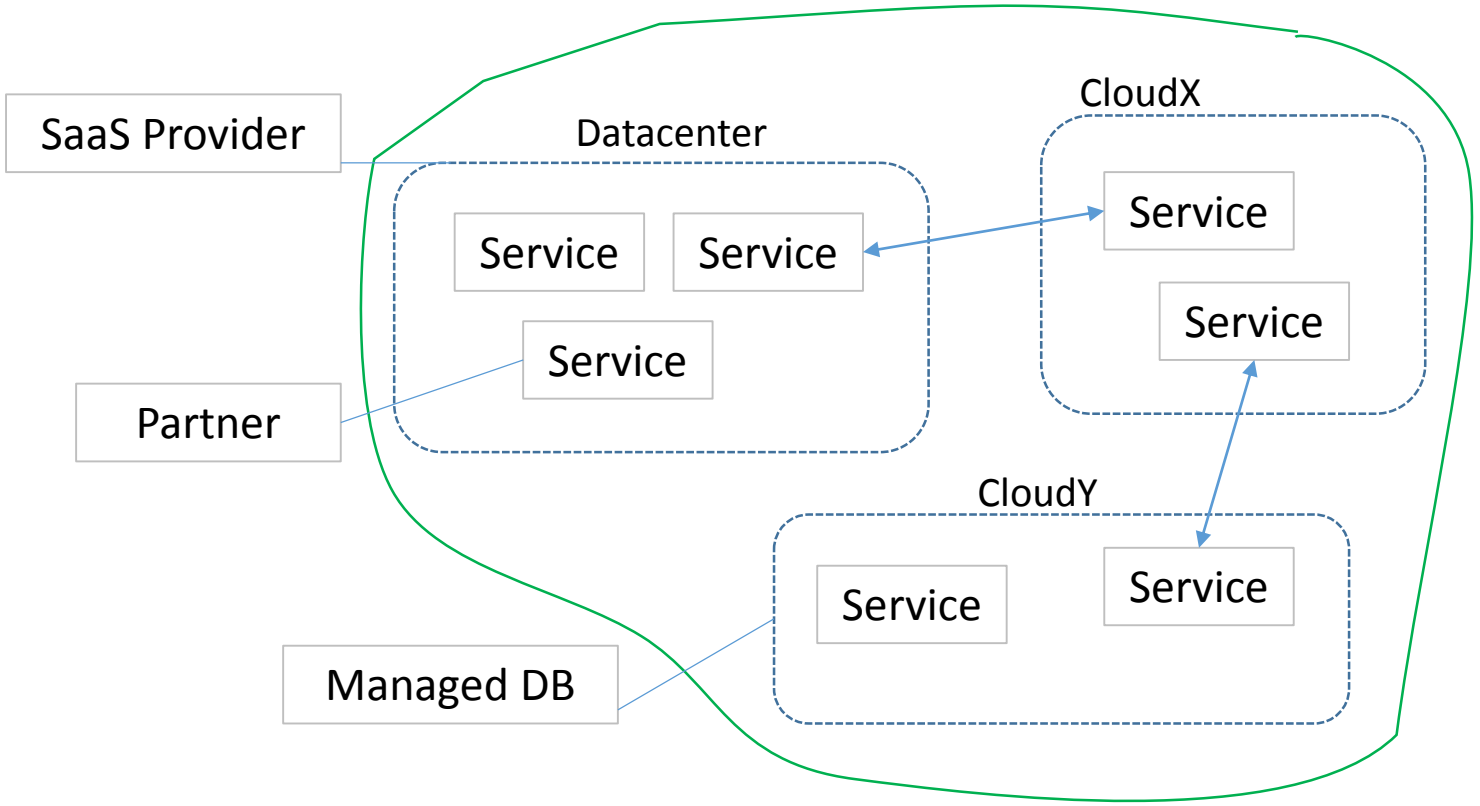


- Increased attack surface & risk of leakage across untrusted networks
- Long-lived service credentials exist across applications, repositories, platforms, and tools, making them ripe for theft.

Workloads Security Risks

- **Misconfigurations**: Improperly configured Kubernetes components can expose the cluster to attacks or unauthorized access.
- **Vulnerable container images**: Deploying containers with known vulnerabilities can lead to security breaches.
- **Insecure communication**: Lack of encryption and mutual authentication between services can result in data leaks and man-in-the-middle attacks.
- **Insider threats**: Malicious or negligent actions by employees or contractors can compromise the security of the container platform.
- **Supply chain attacks**: Infiltration of malicious code or compromised dependencies during the software development lifecycle can lead to compromised applications and infrastructure.

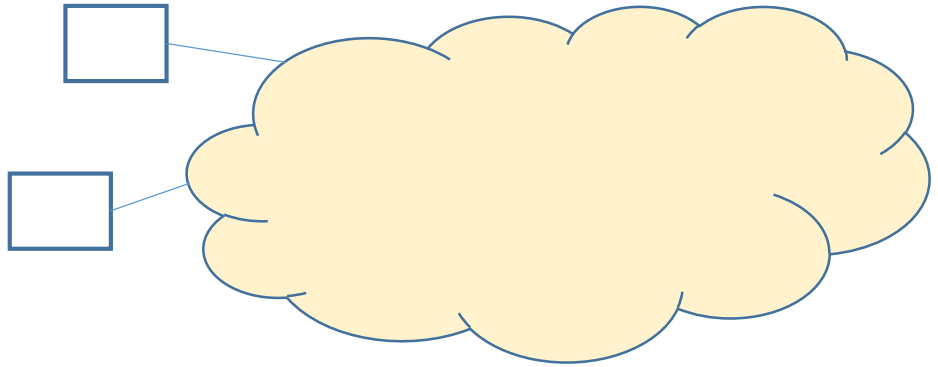
Authentication for Zero-Trust Security Model



Perimeter Security hard to defend when adding: Services, Clouds, Regions

Clouds and Containers Adopting Zero Trust

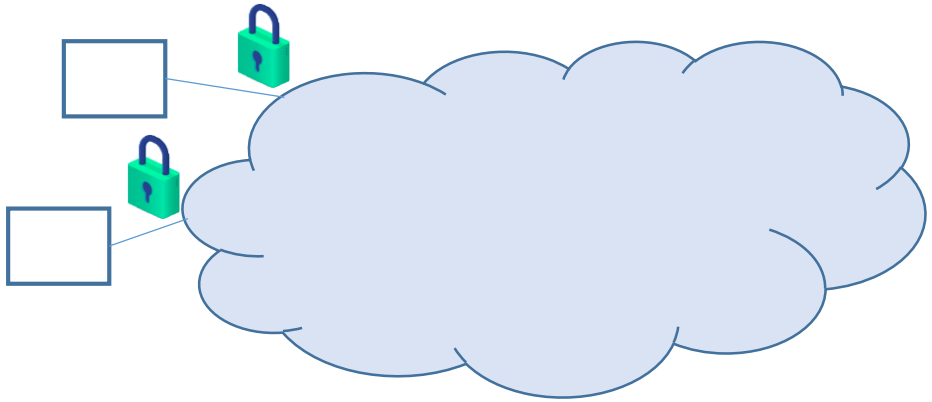
Perimeter - based



Attempts to build a trusted “wall”

- Relies on IP addresses or physical locations
- Difficult to implement for today’s dynamic environments

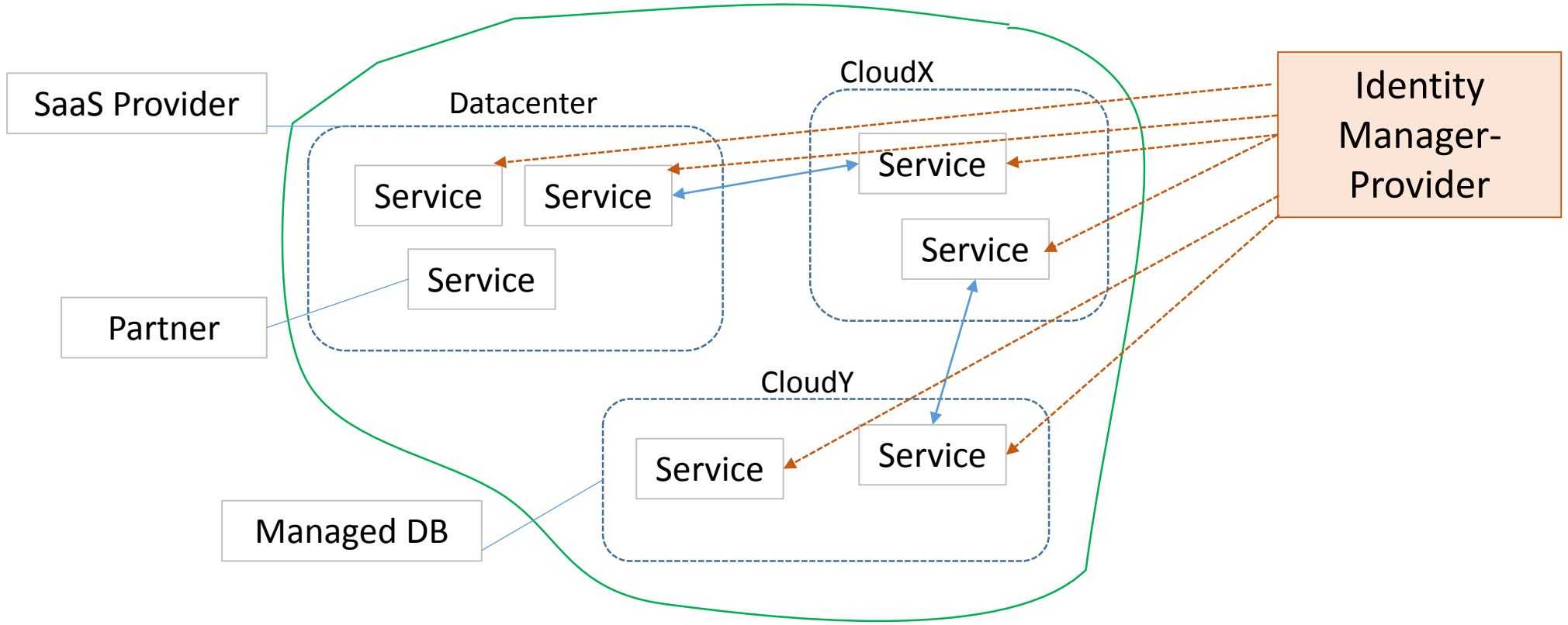
Zero Trust



Assumes “bad guys” are everywhere

- Uses cryptographic identities for authenticating every system/user
- Enables universal enforcement across hybrid infrastructures

SPIFFE Ensures Zero-Trust Security



Guarantee each workload/service will get its own SERVICE identity

- unique
- secure
- provable

SPIFFE
(Secure Production Identity Framework for Everyone)

- Challenge: have an identity, rotating secrets and automated bootstrap for trust and make it available for other systems to authenticate

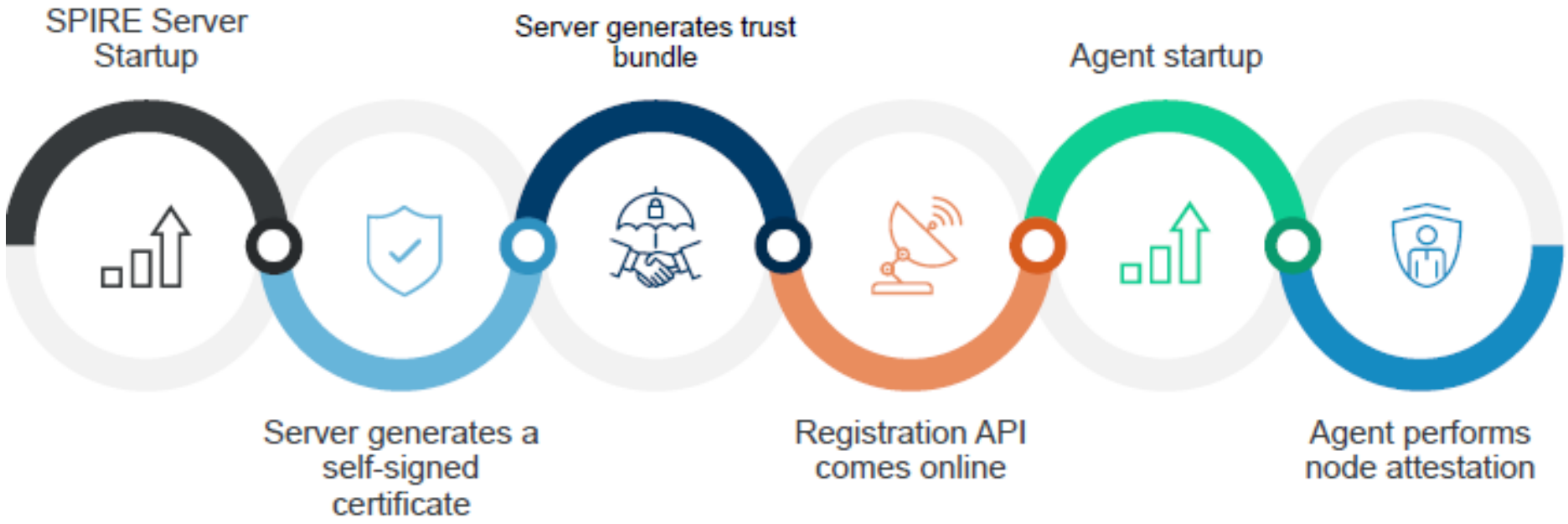
Custom granularity
Can be fine grained as desired. Eg could be a specific process on a node.

Platform Agnostic
Not specific to any platform,
doesn't assume eg k8s.

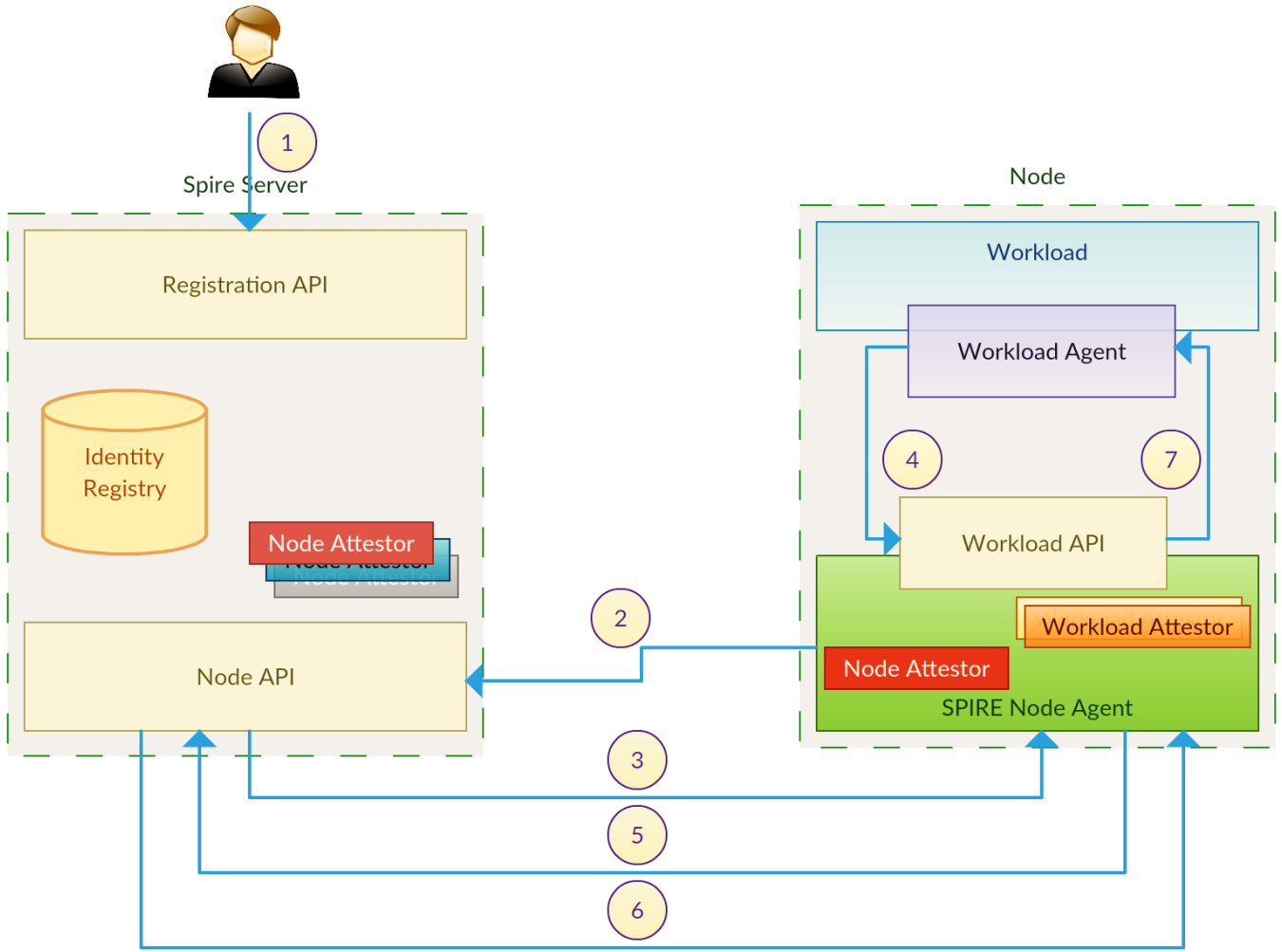
Elastic
A workload can span multiple nodes, each with unique IP addresses

Isolated
Isolated from other workloads such that a malicious workload could not steal the credentials of another

SPIRE Server-Agent Startup

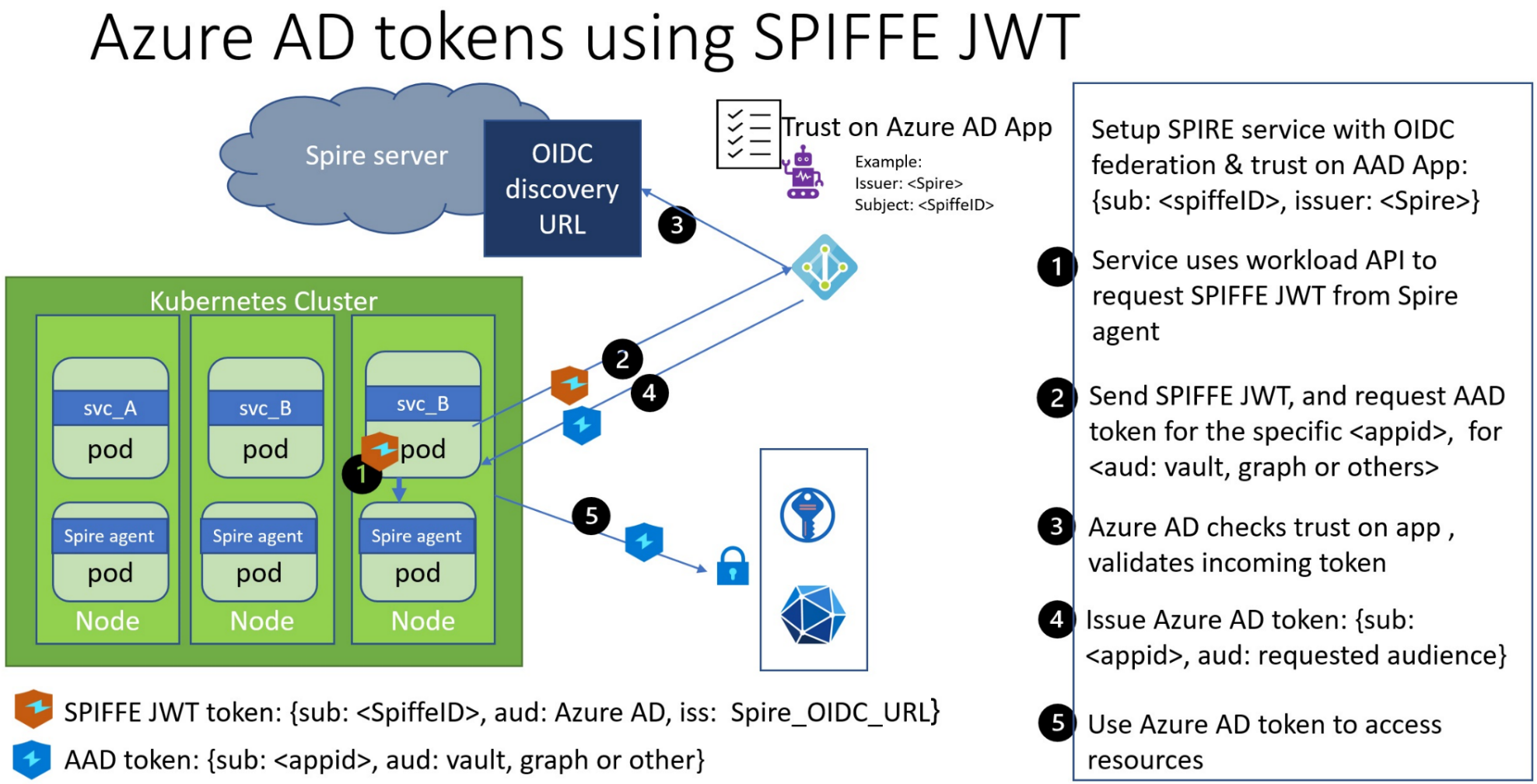


SPIFFE Overview

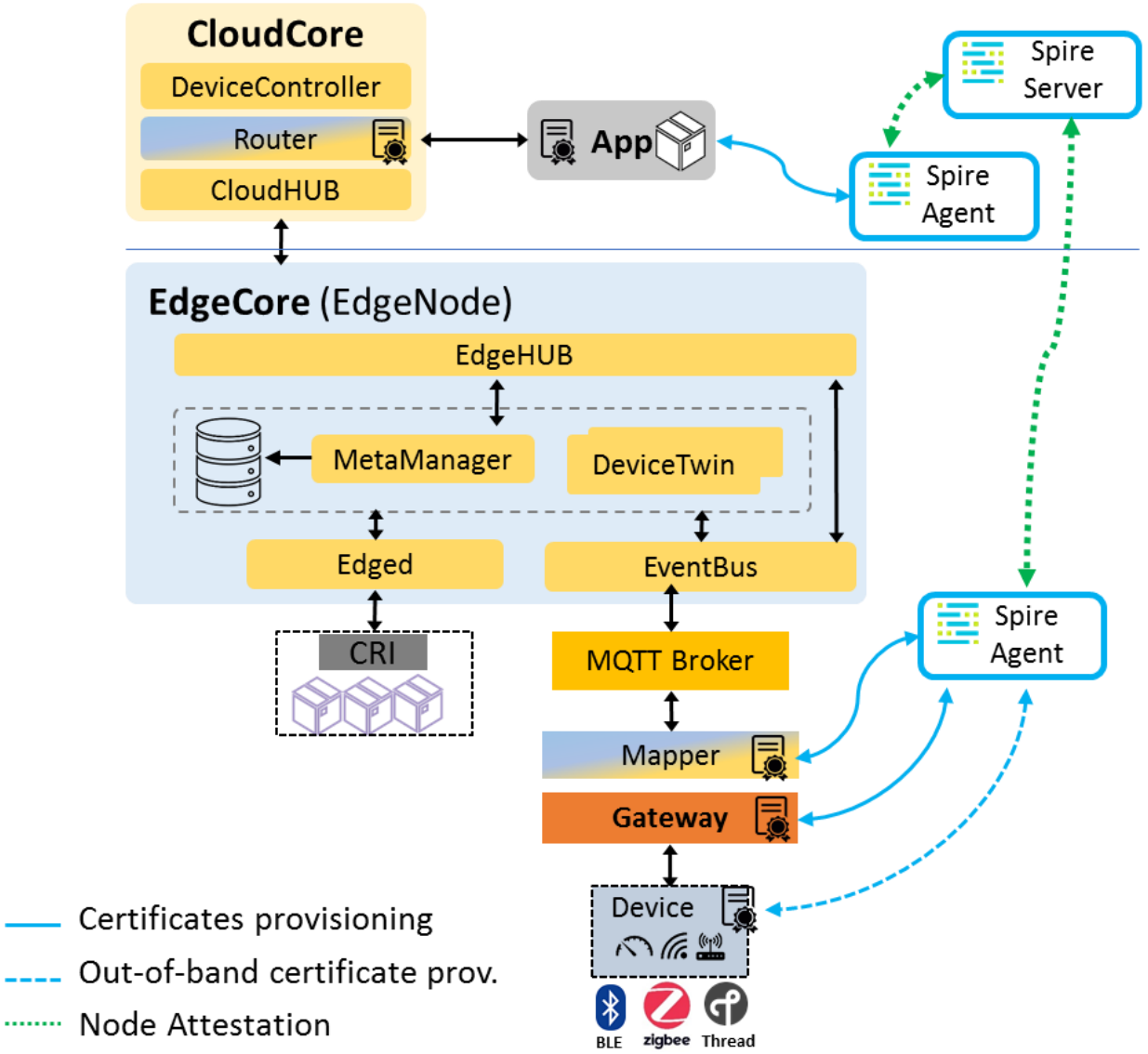


- Registration API is called by either an administrator or a third party application to populate the identity registry with the required SPIFFE IDs and relevant selectors.
- Node agent get authenticated with the SPIRE server using a pre-established cryptographic key pair or based in the infrastructure provider. For example in the case of AWS EC2, node agent will submit the node's Instance Identification Document(IID) issued by AWS.
- Node attestor in the SPIRE server validates the provided identification document based on the used mechanism. If the AWS IID is used, the relevant attestor will validate it with AWS settings. Upon successful validation SPIRE server sends back a set of SPIFFE IDs that can be issued to the node along with their process selector policies.
- When workload start to run in the node, it first make a call to the node agent asking 'who am I?'.
- Based on the process selectors node agent received in the previous step, and using the workload attestors, agent decides on the SPIFFE ID to be given to workload. It generates a key pair based on that and sends the CSR(Certificate Signing Request) to the SPIRE server.
- SPIRE server responds to the node agent with the signed SVID for the workload along with the trust bundles, indicating which other loads can be trusted by this workload.
- Upon receiving the response from SPIRE server, node agent, handover the received SVID, trust bundles the generated private key to the workload. This private key never leave the node it's workload belongs to.

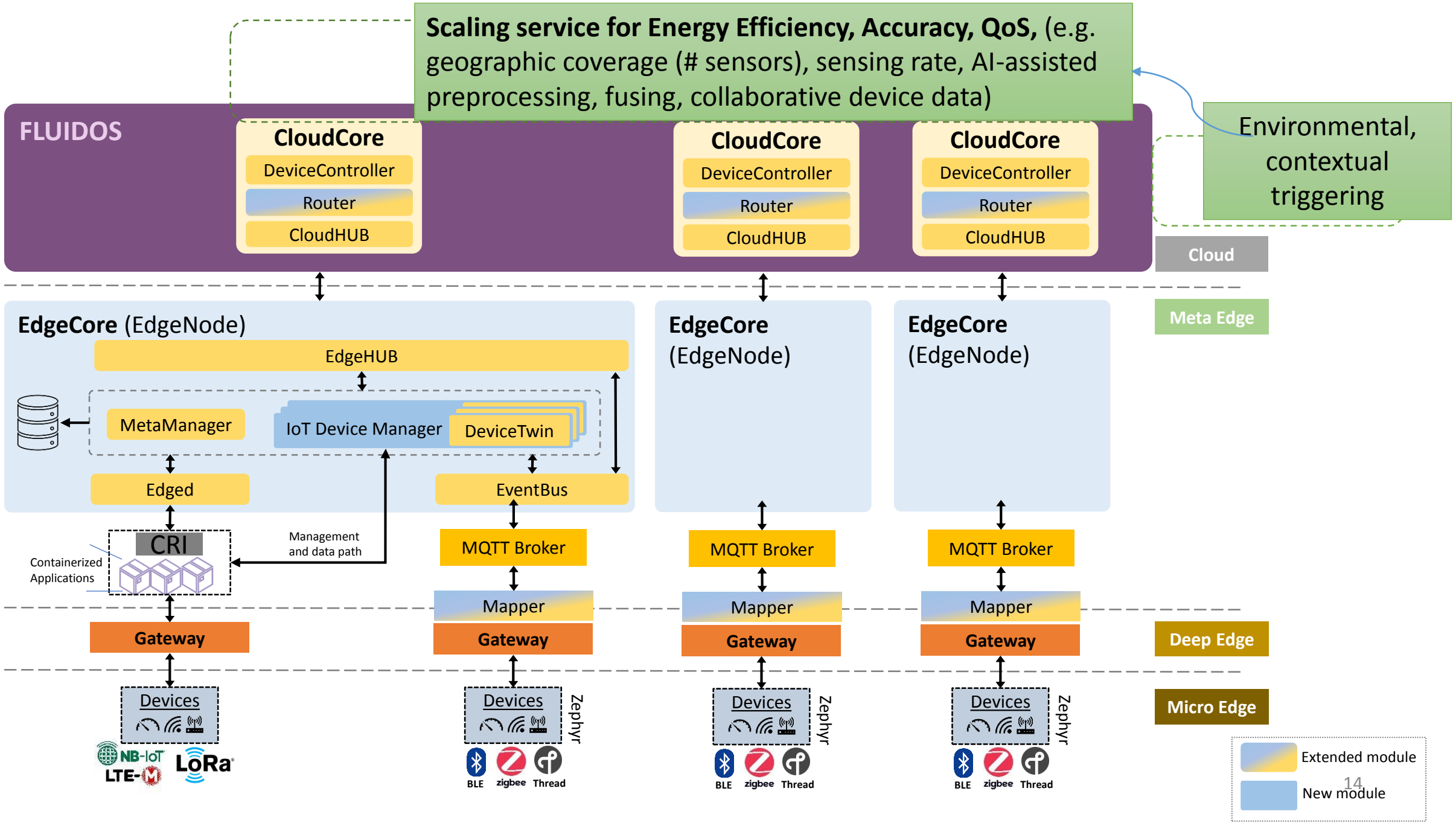
Example: Azure AD workload identity federation with SPIFFE and SPIRE



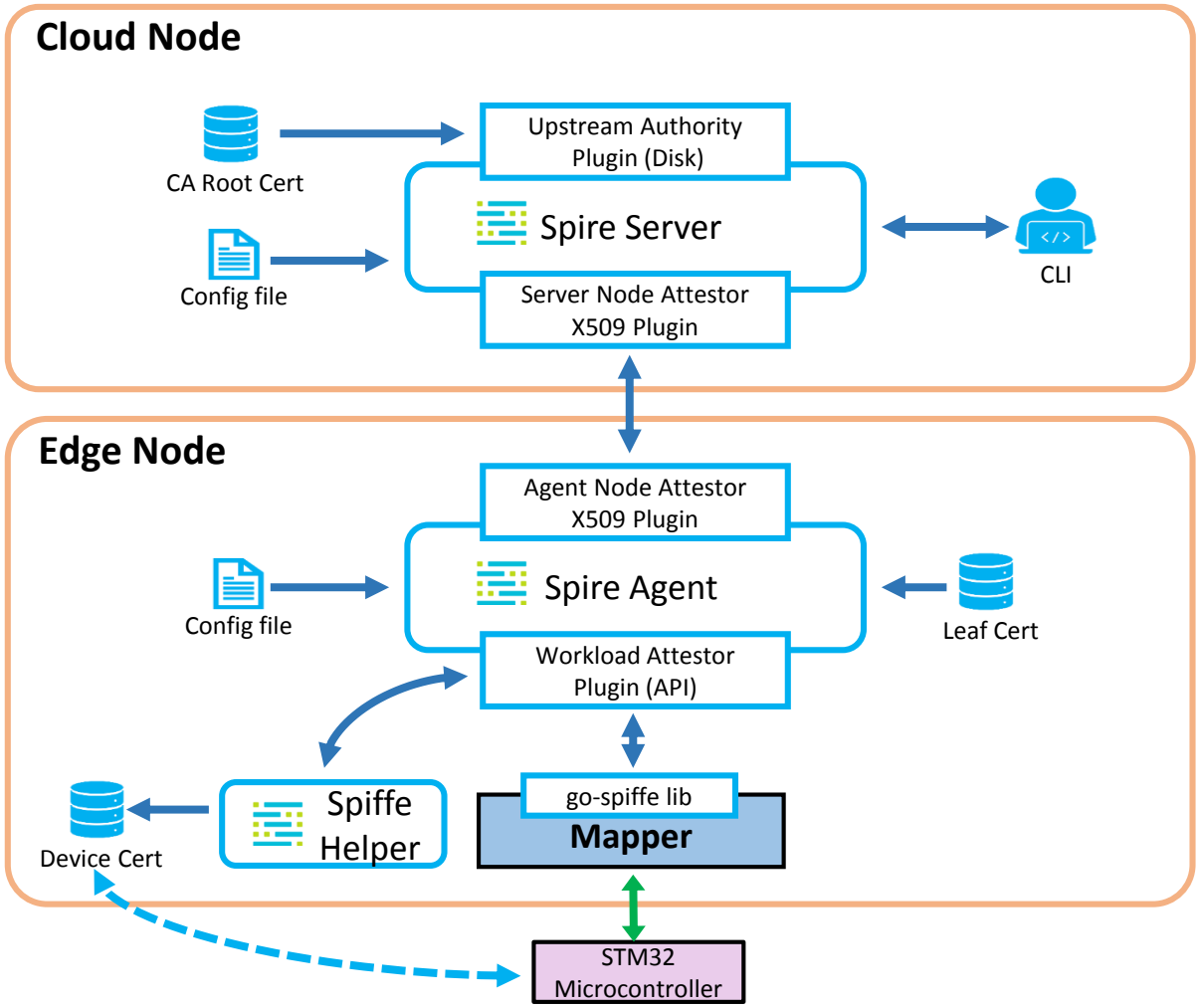
SPIFFE in IoT-Edge Architecture (FLUIDOS)

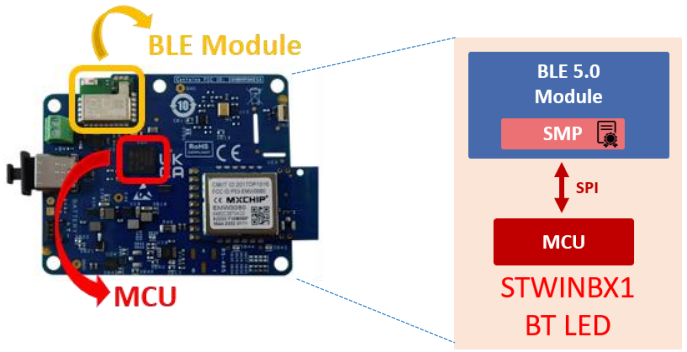


- **Cloudcore-EdgeCore** monitoring resource changes
  - Metadata traffic includes edge node status and application status
- Extended Kubernetes **Custom Resource Definitions** (CRDs) to manage sensor data traffic


Scaling service for Energy Efficiency, Accuracy, QoS, (e.g. geographic coverage (# sensors), sensing rate, AI-assisted preprocessing, fusing, collaborative device data)
Environmental, contextual triggering
FLUIDOS
CloudCore
DeviceController
Router
CloudHUB
CloudCore
DeviceController
Router
CloudHUB
CloudCore
DeviceController
Router
CloudHUB
Cloud
Meta Edge
EdgeCore (EdgeNode)
EdgeHUB
MetaManager
IoT Device Manager
DeviceTwin
Edged
EventBus
EdgeCore (EdgeNode)
EdgeCore (EdgeNode)
CRI
Containerized Applications
Management and data path
MQTT Broker
MQTT Broker
MQTT Broker
Mapper
Mapper
Mapper
Gateway
Gateway
Gateway
Gateway
Deep Edge
Devices
Devices
Devices
Devices
Micro Edge
NB-IoT
LTE-M
LoRa
BLE
zigbee
Thread
Zephyr
BLE
zigbee
Thread
Zephyr
BLE
zigbee
Thread
Zephyr
Extended module
New module

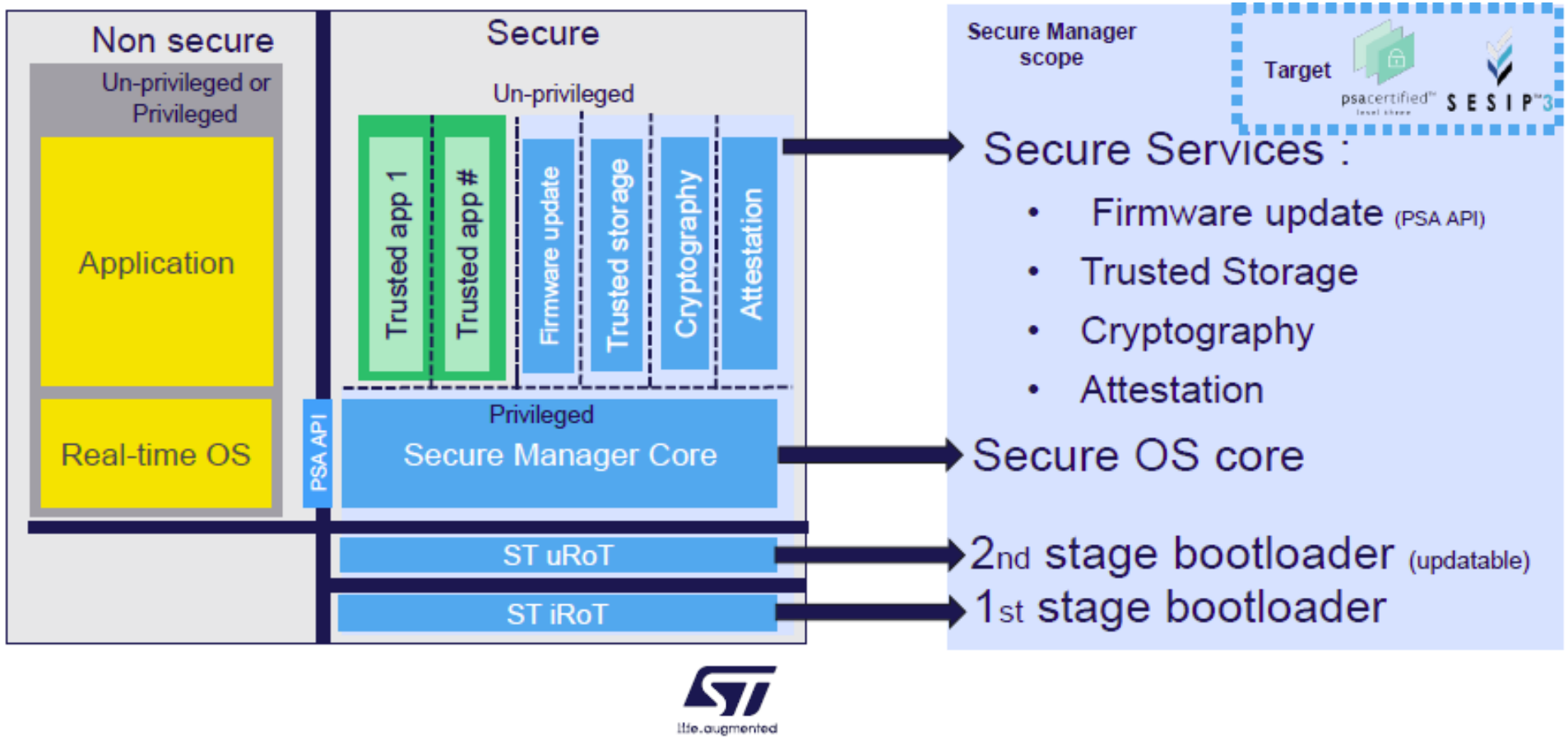
IoT Device Identity Provisioning





Secure Device Management

Trustzone



Securing BT STWINBX1 – Mapper Communication

BLE Stack and Secure Communication:

- **BLE stack integrated - in the GO BLE Library and the BLE Module**
- **BLE stack integrates the Secure Manager (SMP) to enable Secure Connections**
- **Mapper and BLE Module use Diffie-Hellman handshake to secure connection**
- **Diffie-Hellman requires certificates in both sides provided by Spire Agent**

1. Mapper:

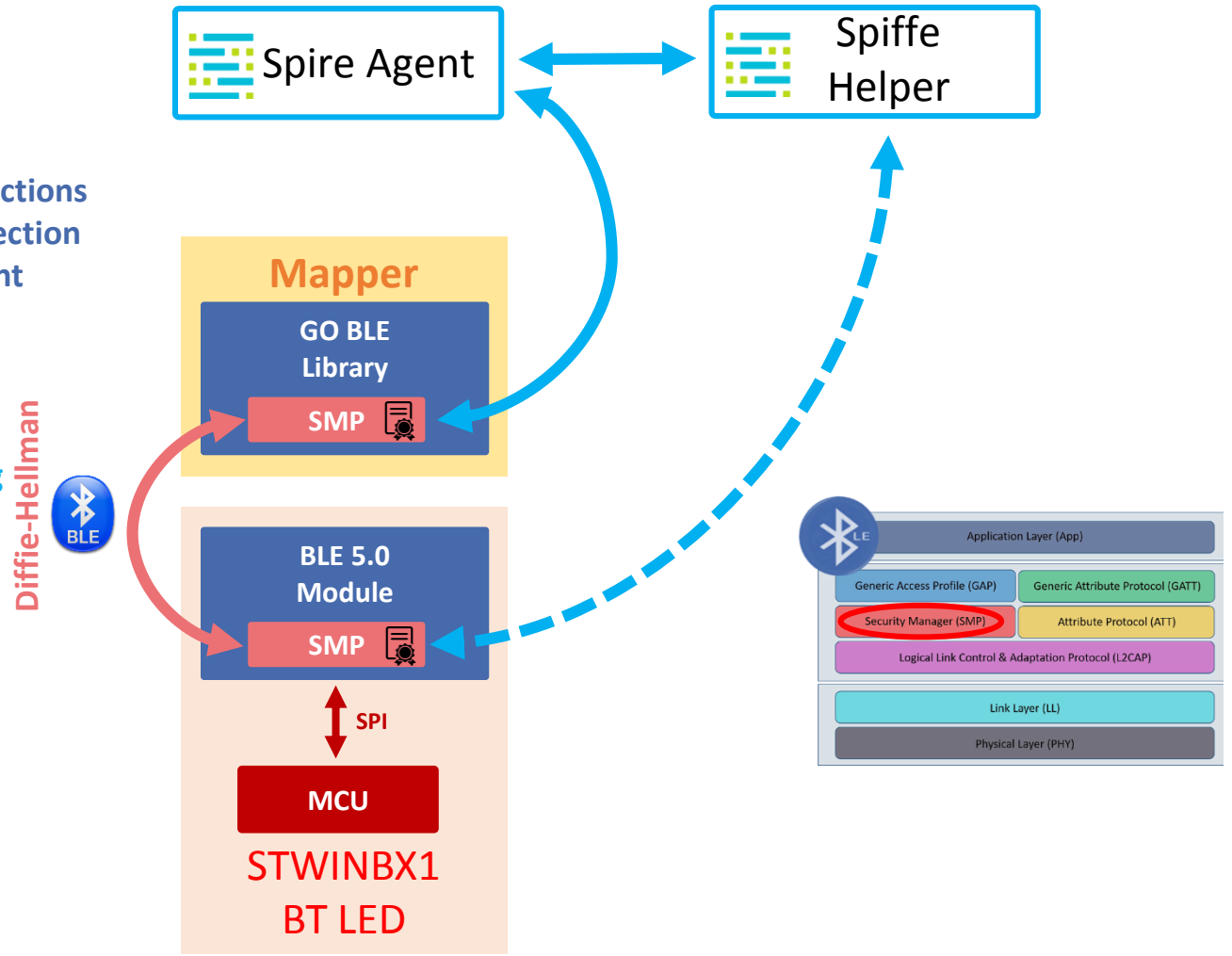
- **Attest to the Spire Agent and receive certificate**
- **Enable BLE Secure Connections (through SMP)**
- **Provide the certificate to the SMP to start Diffie-Hellman secure pairing**

2. Spiffe Helper:

- **Run attestation on behalf of the BT LED**
- **Receive certificate on behalf of the BT LED**
- **Out-of-band distribution of the certificate to the BT LED**

3. STWINBX1 MCU:

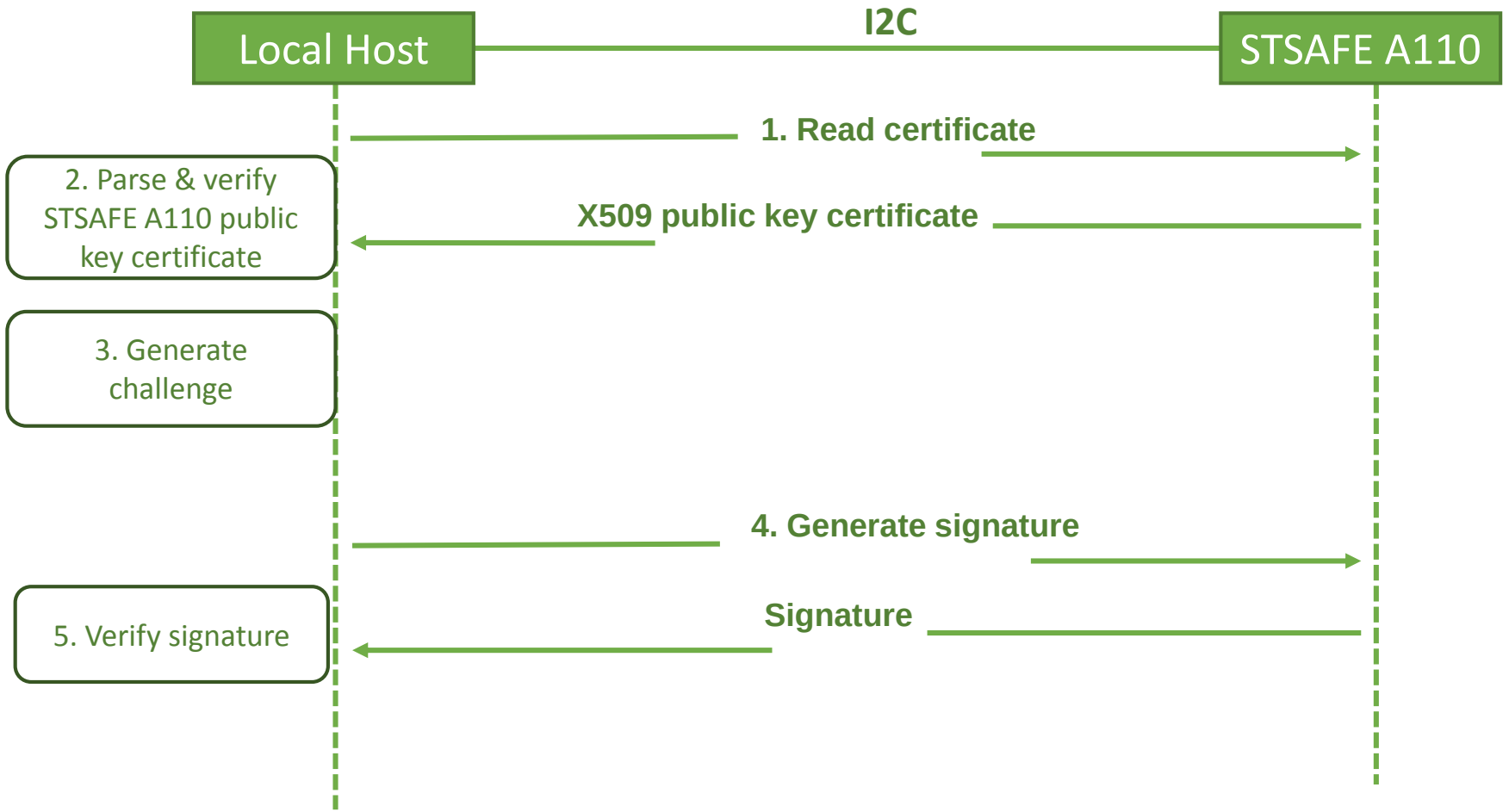
- **Configure the BLE Module to enable BLE Secure Connections (over SPI)**
- **Provide the BT LED certificate to the BLE Module (over SPI)**



Spire certificate distribution/attestation

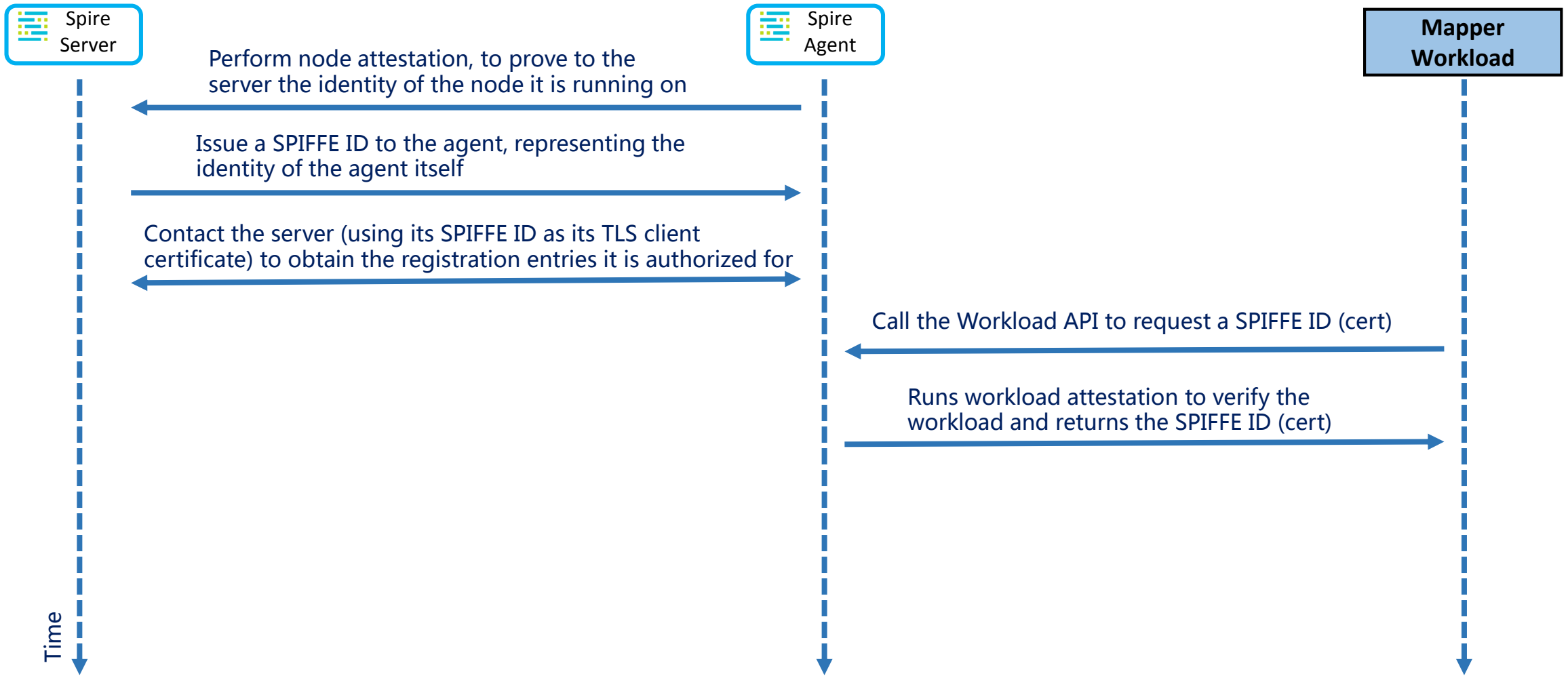
Out-of-band certificate distribution

Secure Device Bootstrapping

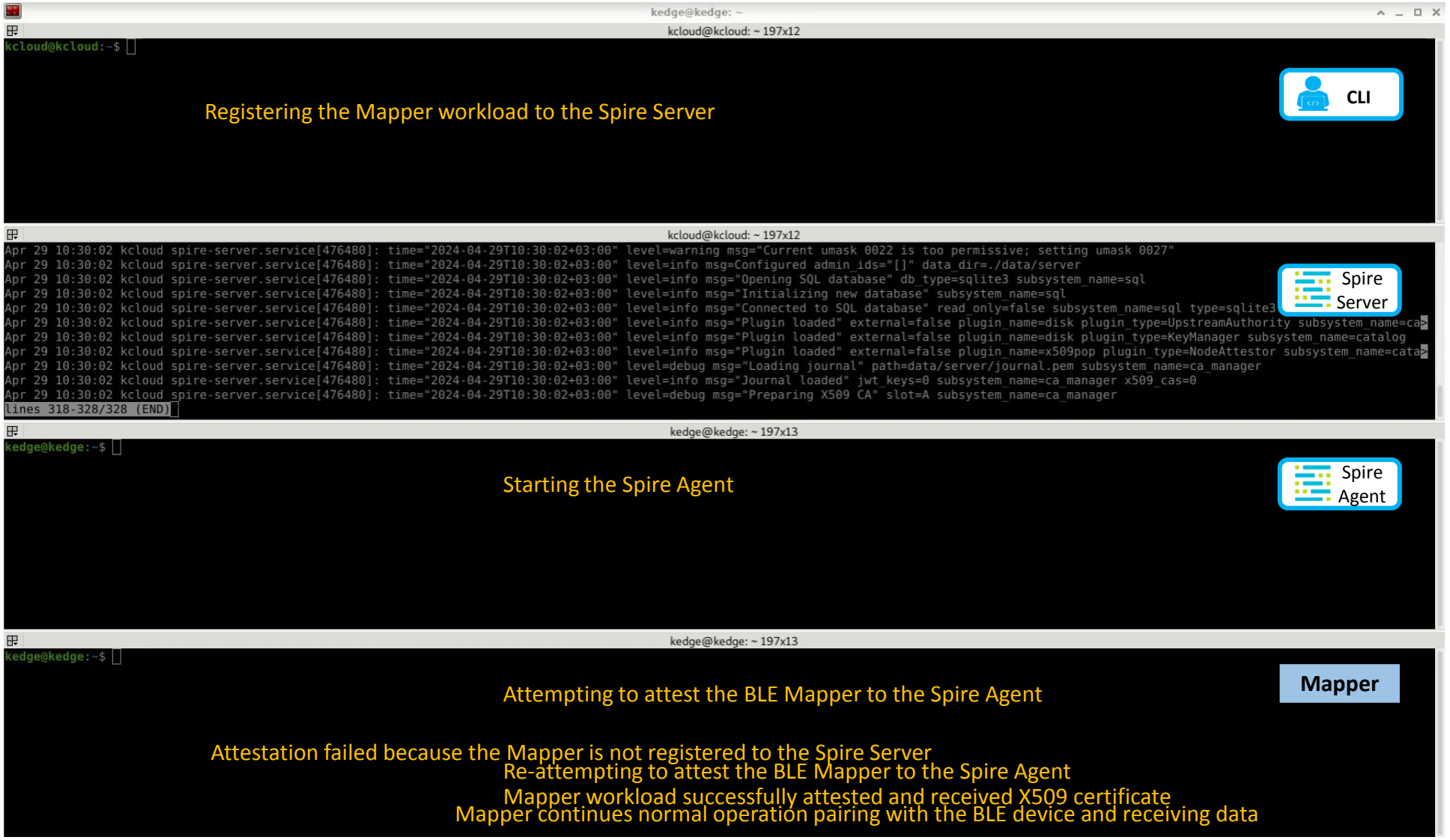


- Threats
  - Device cloning or counterfeiting
  - Device integrity or data corruption
- Countermeasures
  - EAL5+ CC certified secure MCU
  - Secure operating system, secure handling of cryptographic keys
  - Customer secure keys and certificates loading at ST in a security certified environment

Mapper Identity Provisioning



Certificate Distribution to Mapper



Conclusions

- SPIFFE identity management and mTLS based connectivity for trusting Sensor-generated traffic flows in a the Kubernetes-managed cloud-edge setup
- SPIRE's plugin architecture enables diverse workload attestation options beyond the Kubernetes namespace and service account attestation (e.g., offered by Istio)
- Holistic Identity management in IoT-based environment

- SPIFFE and the service mesh ecosystem are the technology I wish the virtualization, networking, and security vendors had built 5-10 years ago…

Thank you for your attention!

EU-H2020 FLUIDOS
https://www.fluidos.eu/

Intelligent Systems and Computer Architecture Lab
https://isca.hmu.gr

George Kornaros
[kornaros@hmu.gr]