5G Cellular Telemetry using Zero-Trust Architecture Principles

S. Rangarajan, A. Kogiantis, S. Samtani, G. Di Crescenzo, T. Chen

Peraton | LABS

ITEA Test Instrumentation Workshop

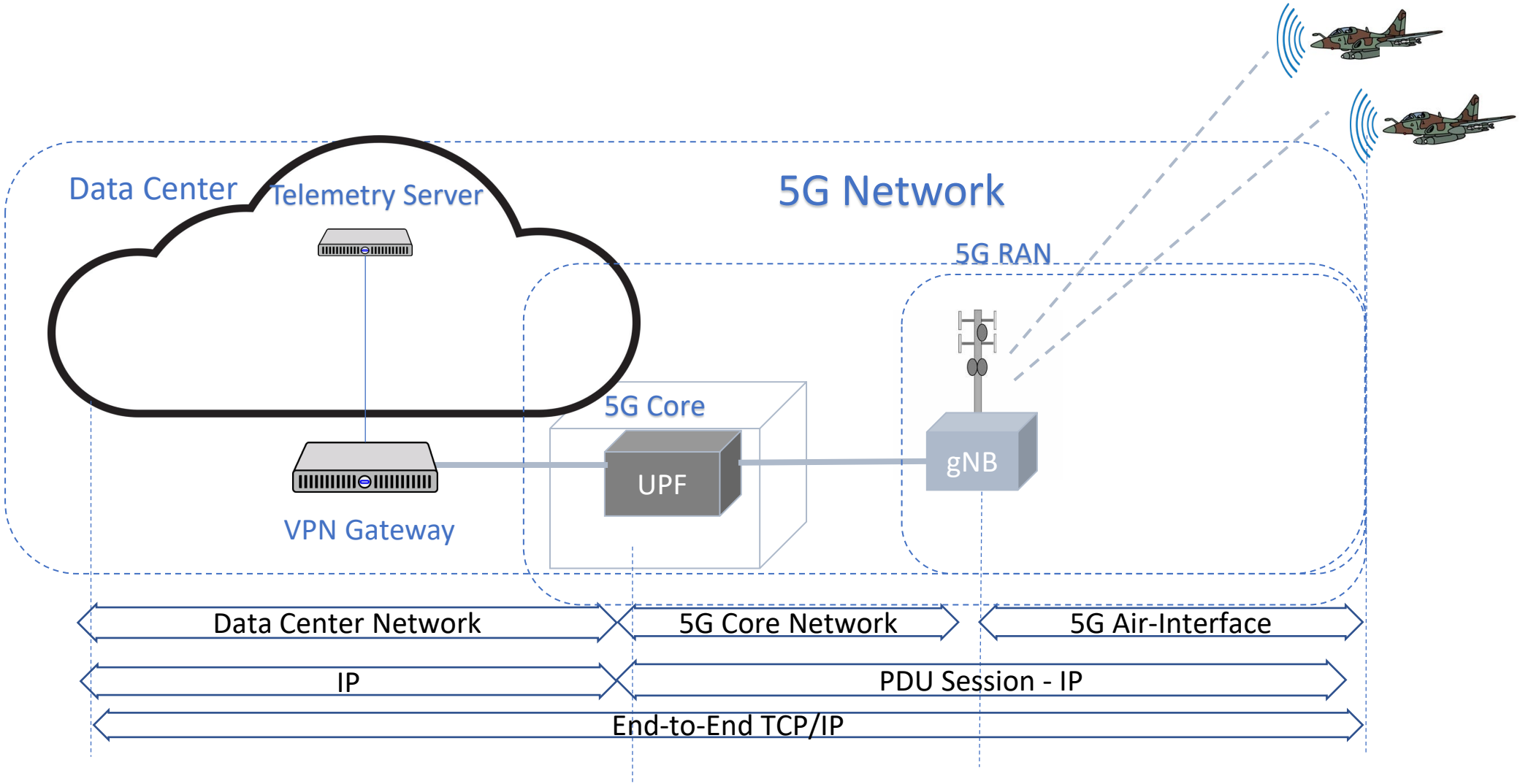
05/16 – 05/19, 2022

Agenda

- Airborne Telemetry over 5G Networks
- End-to-End Zero-trust Security Architecture (ZTA) for a Telemetry Network
- Overview of the security enhancements provided by the ZTA
- How do these enhancements satisfy the ZTA Tenets.
- Details on the four key enhancements
  - Authentication of Telemetry devices
  - Attestation of Telemetry applications
  - Authorization of Telemetry applications to communicate with a Telemetry server
  - End-to-End Hybrid Encryption between the Telemetry devices and the Telemetry Server
- A note on supporting secure memory on the Telemetry devices.

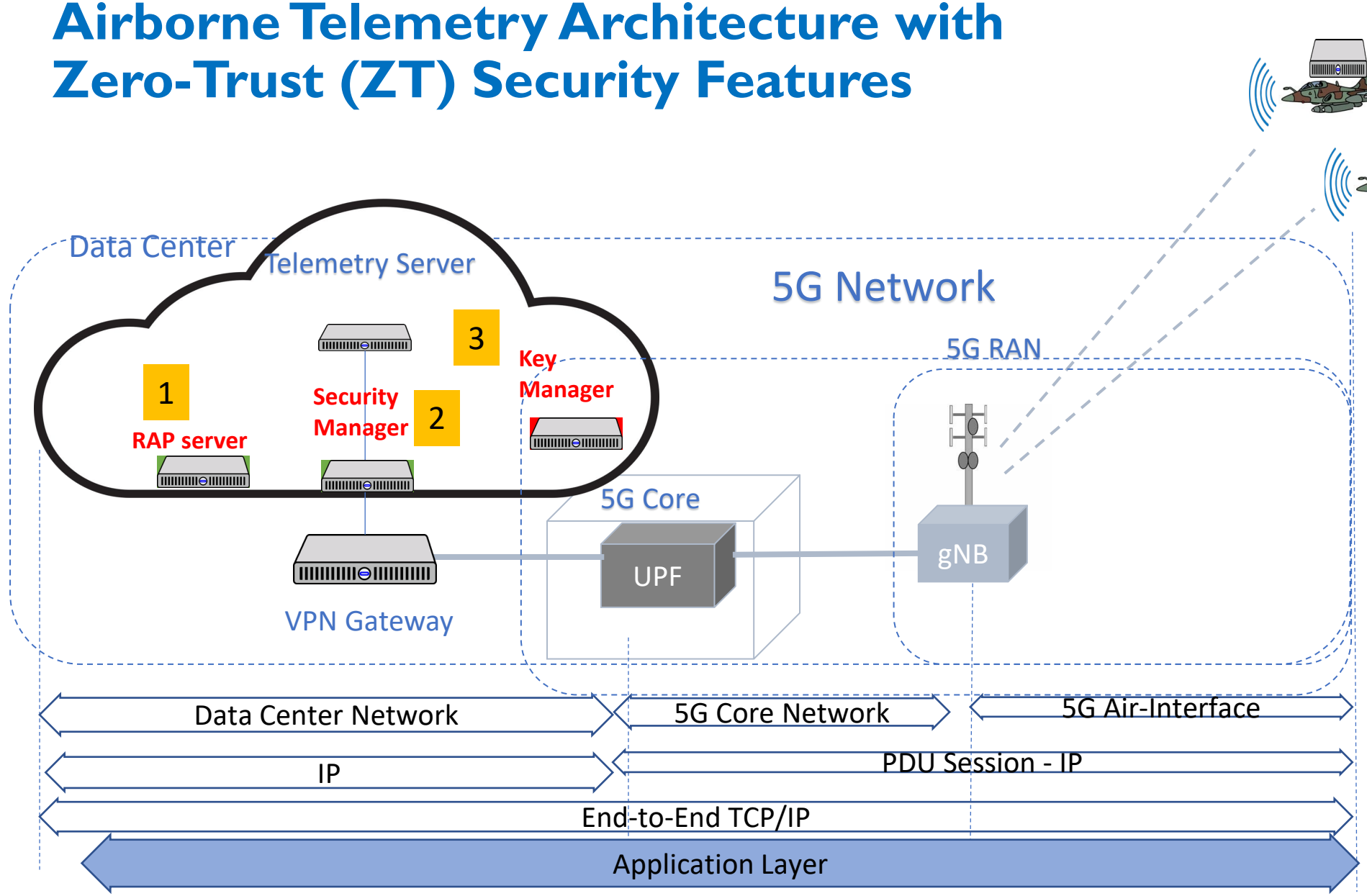
** Concepts discussed in this work have been developed as part of the DoD NIWC Naval Base Coronado (NBC) program NSC-20-2050: Navy Smart Warehouse 5G Network Enhancement Prototype on End-to-End Zero-Trust security for IoT devices

Airborne 5G Cellular Based Telemetry Architecture



Peraton

Airborne Telemetry Architecture with Zero-Trust (ZT) Security Features



1) **RAP (Remote Attestation Protocol) server** – Executes Remote Attestation Protocol, attests software on telemetry devices on the aircrafts, Informs the Security Manager about attested devices to authorize.
2) **Security Manager** –Uses public/private key encryption to receive a shared key from each of the Telemetry devices. Serves as end-point of encryption. Authorizes only attested devices to connect to the Application Servers
3) **Key Manager** - Trusted key manager which creates the private and public keys (similar to a Certificate Authority)
4) **Telemetry devices** on the aircraft that send data to the Telemetry Server on the ground

Peraton

Overview of Security Enhancements

Zero-Trust Security: End-to-end security architecture solution is designed to ensure trust of Telemetry Devices software (on the aircraft) and its communication with the Telemetry Servers on the ground, based zero-trust architecture principles, using the following mechanisms:

- **Authentication:** Only allow authenticated Telemetry devices to enter the network through an authentication step
- **Attestation**: Telemetry Applications that are deployed on the Telemetry devices are attested periodically to ensure they have not been remotely (or physically) modified.
  - **Authorization**, which is an outcome of Attestation where only Attested devices are authorized to access the Application servers
  - Authentication is for validating the Telemetry Device, whereas Attestation is performed on the applications that run on the Telemetry devices; this ensures that the applications have not been tampered with during operation
- **Encryption**: Data from the Telemetry Devices is encrypted end-to-end across the 5G network all-the-way to the Security Manager (which serves as a proxy for the Application servers that consume this data). The link from the Security Manager to the Telemetry Server is expected to be in the Implicitly Trusted Zone.
- **Secure memory**: Provide MPU – Memory Protection Unit (Software-based) or TrustZone (Hardware-based) at the Telemetry Devices, to create a secure part of the memory on the device that Telemetry applications (which run on application memory) cannot access. This is the location where developed security software on the Telemetry Devices reside and are protected.

Main Components of ZT Security Architecture

- **RAP (Remote Attestation Protocol) Server:** Continuously attests that the Telemetry Applications on the Telemetry Devices are is valid and has not been remotely tampered with.
- **Security Manager:** Responsible for authorizing only attested Telemetry Devices to communicate with the Telemetry servers on the ground.
  - Authorization using EAT: An entity attestation token (EAT) is provided to attested devices.
  - An unexpired token is necessary for an IoT device to be presented to the Security Manager to be allowed to communicate with the Application Servers.
- **Key Manager:** The Certificate Authority which creates private/public keys and certificates for the RAP server, the Key Manager and the Telemetry Devices
- **Telemetry Device MCU (not show in the Figure):** The microcontroller unit on the Telemetry Device that implements the security procedures in conjunction with the RAP server and the Security Manager.
  - The Peraton Labs security software on the Telemetry devices run within as secure memory created through either software or hardware mechanisms

- **The Telemetry Devices maintain secure connections with the Security Manager and the RAP server**

Peraton

What is ZTA (Zero-Trust Architecture) Approach satisfies the listed ZTA Tenets

ZTA Tenets

- 1. All data sources and computing services are considered resources.
- 2. All communication is secured regardless of network location.
- 3. Access to individual enterprise resources is granted on a per-session basis.
- 4. Access to resources is determined by dynamic policy—including observable state of client identity, application/service, and requesting asset—and may include other behavioral and environmental attributes.
- 5. Enterprise monitors and measures integrity and security posture of all assets.
- 6. All resource authentication and authorization are dynamic and strictly enforced before access is allowed.

From NIST-207 Document

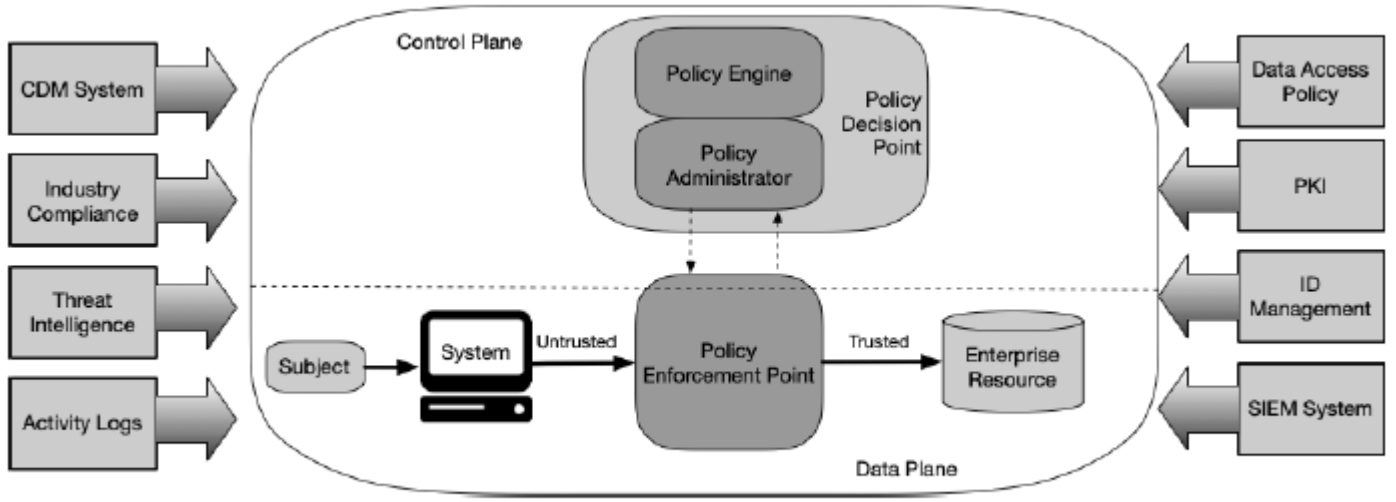
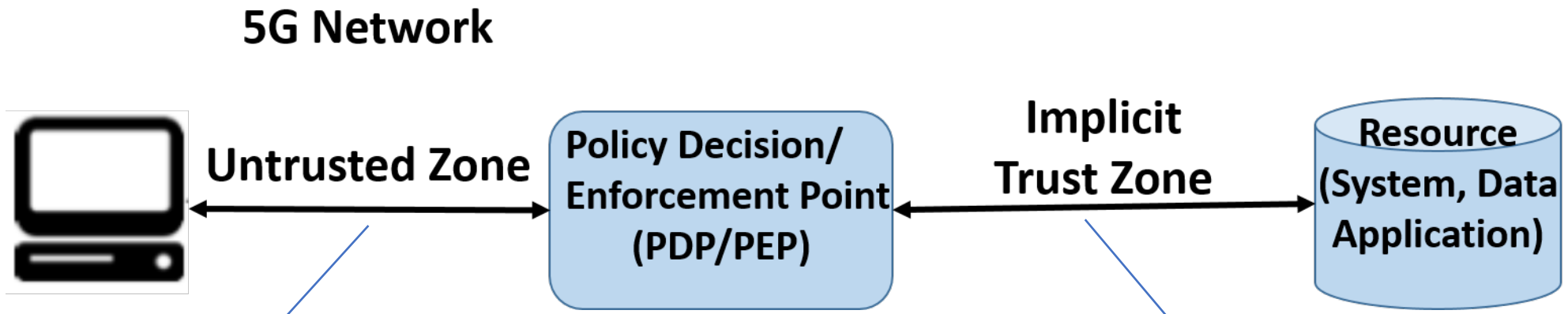


Figure 2: Core Zero Trust Logical Components

- NIST-207 does not provide a solution for ZTA but only provides a framework. We implement a ZTA based on the tenets specified by the framework, using four key mechanisms.
  - Authentication of the Telemetry Device
  - Attestation of the Telemetry Device Application Software
  - Authorization of the Telemetry Applications to communicate with the Telemetry Server
  - End-to-End encryption of the data from the Telemetry Devices to the Security Manager
- In the above architecture, the Policy Decision Point (PDP) is implemented by the RAP Server, and the Policy Enforcement Point (PEP) is implemented by the Security Manager

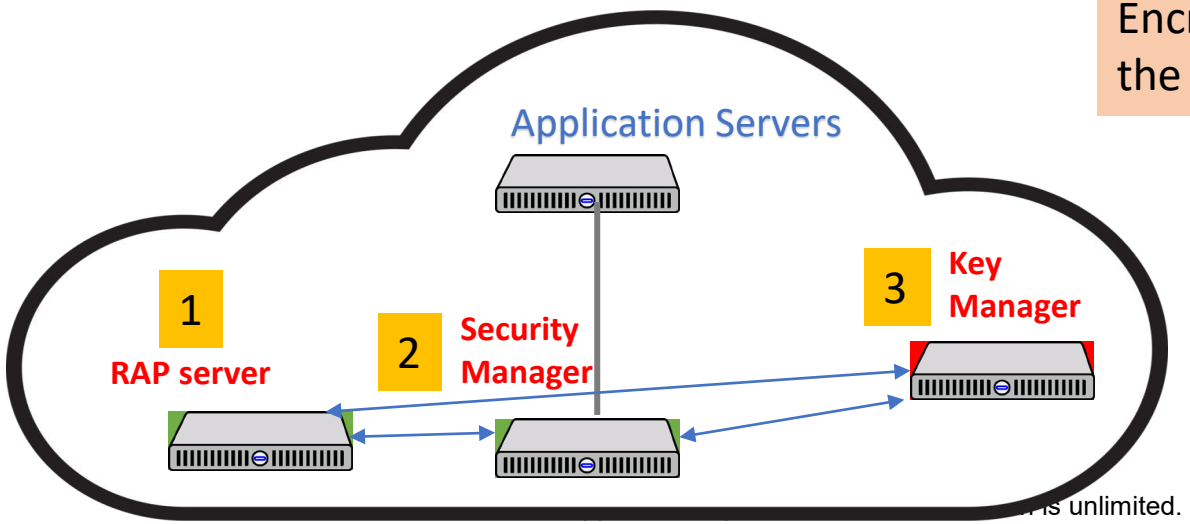
ZTA Architecture from NIST-800-207 and mapping



Enable End-to-End Trust using Authentication and Encryption Attestation and Authorization

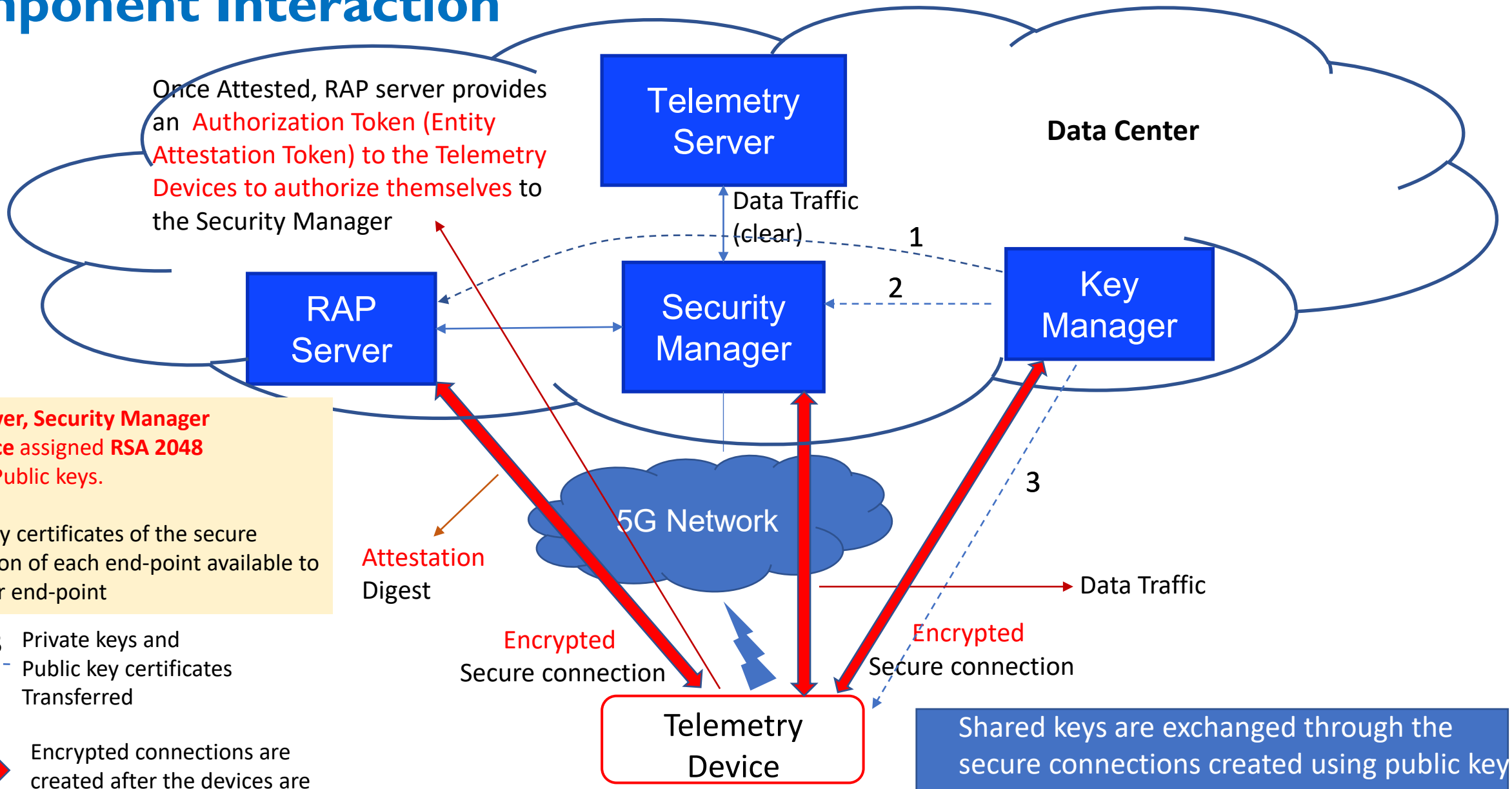
RAP Server – **PDP**
Security Manager – **PEP**
Resource – **Application Servers**

Assume Implicit Trust Zone between Security Manager and Application Servers for now. Encryption can be extended to the Application Servers



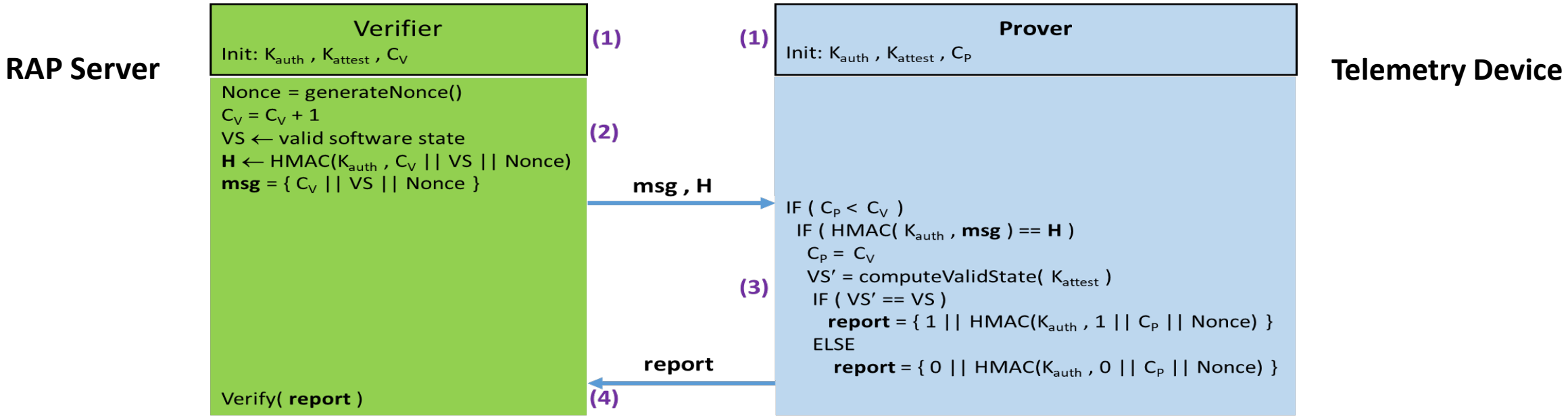
Peraton

Component Interaction



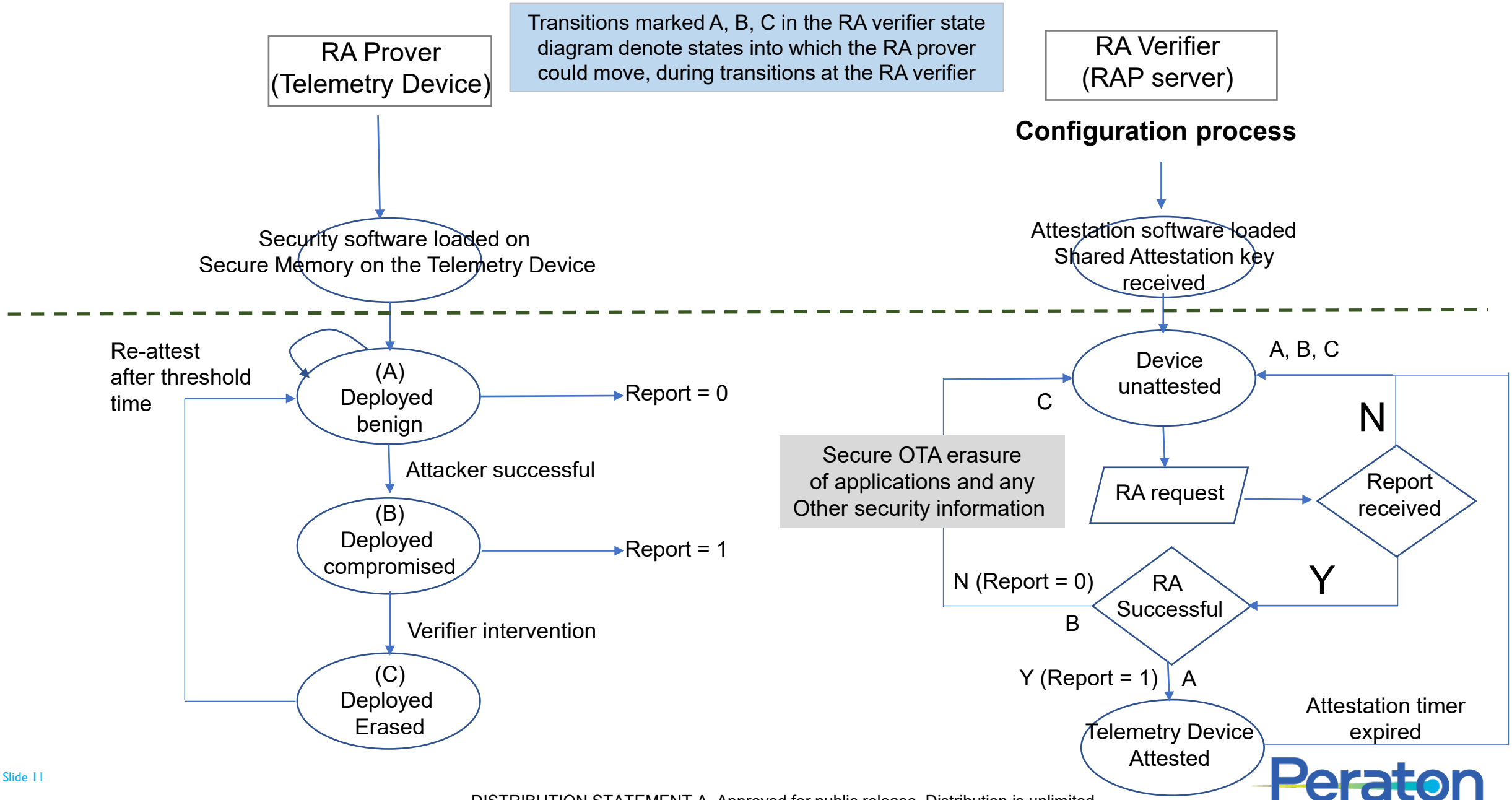
Attestation step is also authenticated

- As part of the Remote Attestation Protocol (RAP), there is a Verifier on the RAP server which uses a Prover on the Telemetry Device (running within the secure memory on the Telemetry Device) to attest the Telemetry Applications.
- Authentication is performed as part of attestation and initiated by the Verifier. Both sides authenticate each other.



- Attestation using HMAC using SHA 256 using the shared attestation key
- Each attestation event will also be authenticated. HMAC is used for such authentication
- The shared attestation key is initially exchanged using public key cryptography

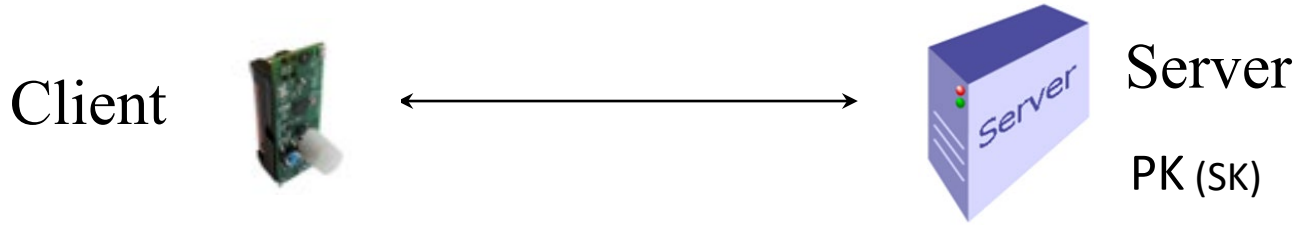
Remote Attestation Flow Graph during Deployment



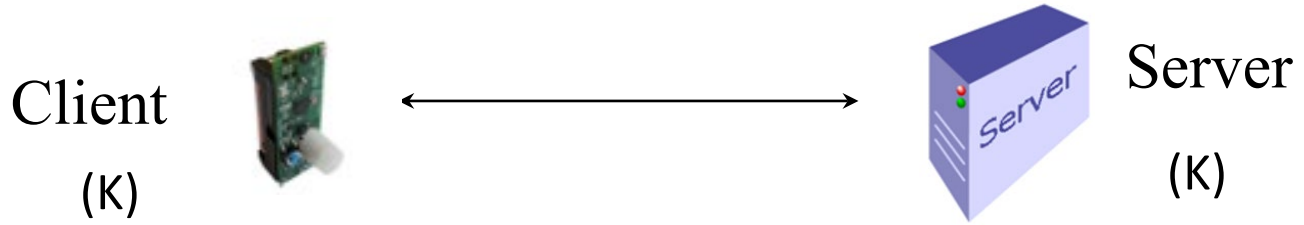
Peraton

What is Hybrid Encryption?

- **Two types of encryption: public-key encryption (aka asymmetric encryption) and private-key encryption (aka symmetric encryption)**
- **Public-key encryption**
  - Client (sender) encrypts using server's (receiver's) public key
  - Server decrypts using a matching secret key
  - Techniques typically include number-theory computations (e.g., RSA exponentiations or ECC multiplications)



- **Private-key encryption**
  - Client (sender) encrypts using a key shared with Server
  - Server (receiver) decrypts using the same shared key
  - Techniques typically include block ciphers (e.g., AES)



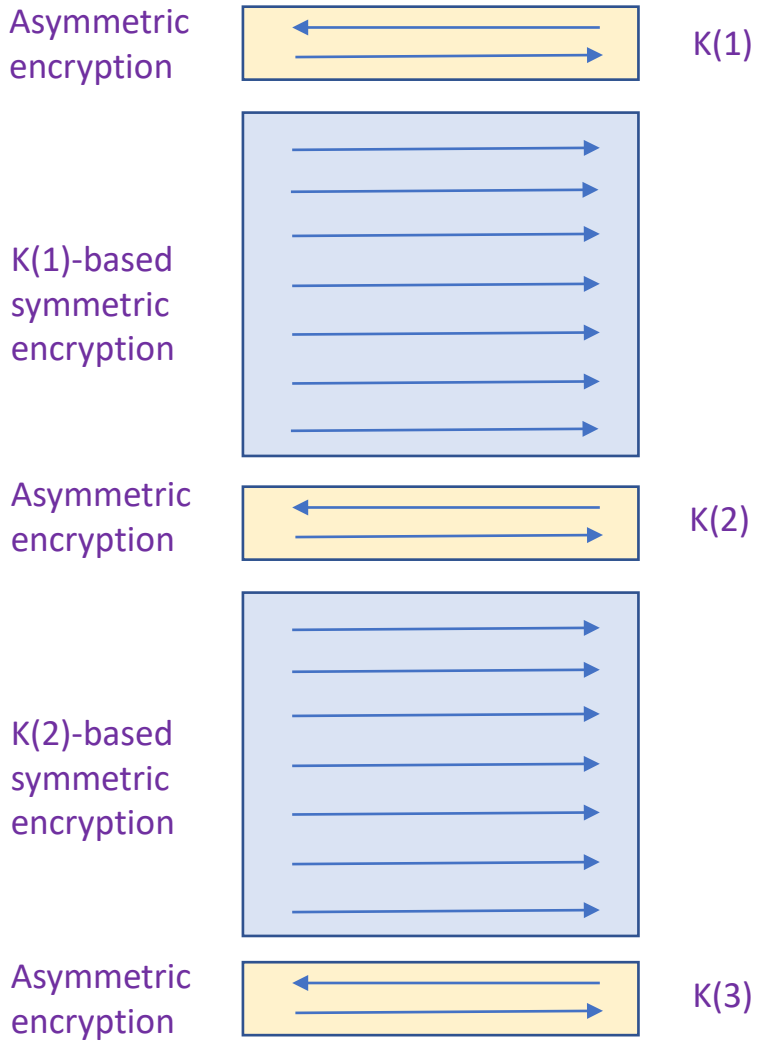
- **Latency of public-key encryption techniques is typically about 3-4 orders of magnitude higher than latency of symmetric encryption**
- **In practice (e.g. TLS, HTTPS) a typical session uses hybrid encryption:**
  - **one run of public-key encryption at the beginning (to encrypt a symmetric key K)**
  - **K-based symmetric encryption (e.g., AES in CBC mode) for rest of the session**

Why do we use Hybrid Encryption?

Telemetry Device to Security Manager communication

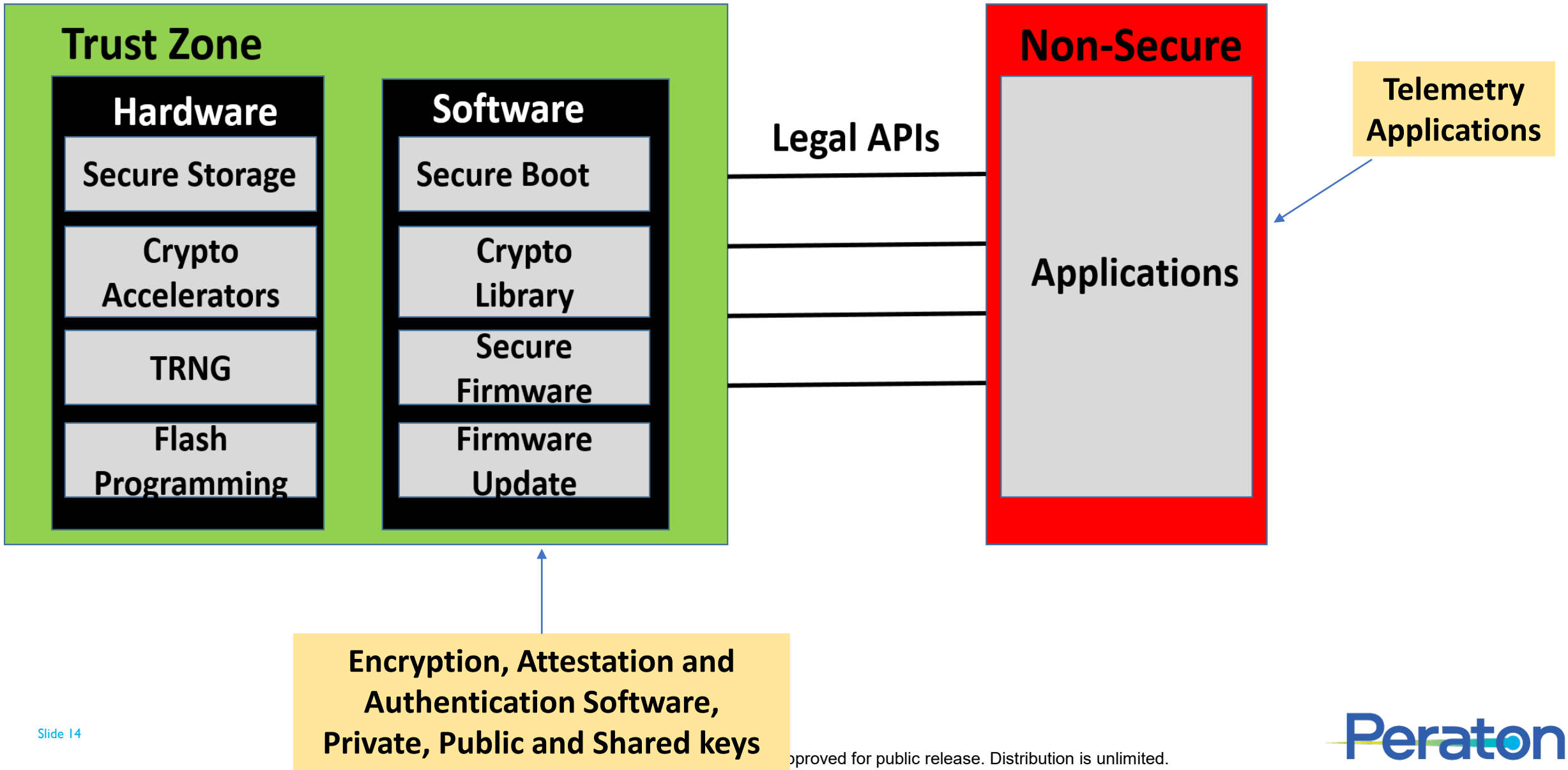
- **Scenario-specific performance/security issues:**
  - A resource-constrained Telemetry device sending encrypted data to a typically not resource constrained server
  - Telemetry device may be unattended for a long time and thus be more subject to undetected compromise

- **Hybrid Encryption for 5G networks, approach:**
  - resource-constrained client should combine
    - rare use of public-key encryption
    - frequent use of block ciphers
  - relative frequency to be set based on a model incorporating a resource-aware combination of
    - security constraints
    - performance constraints



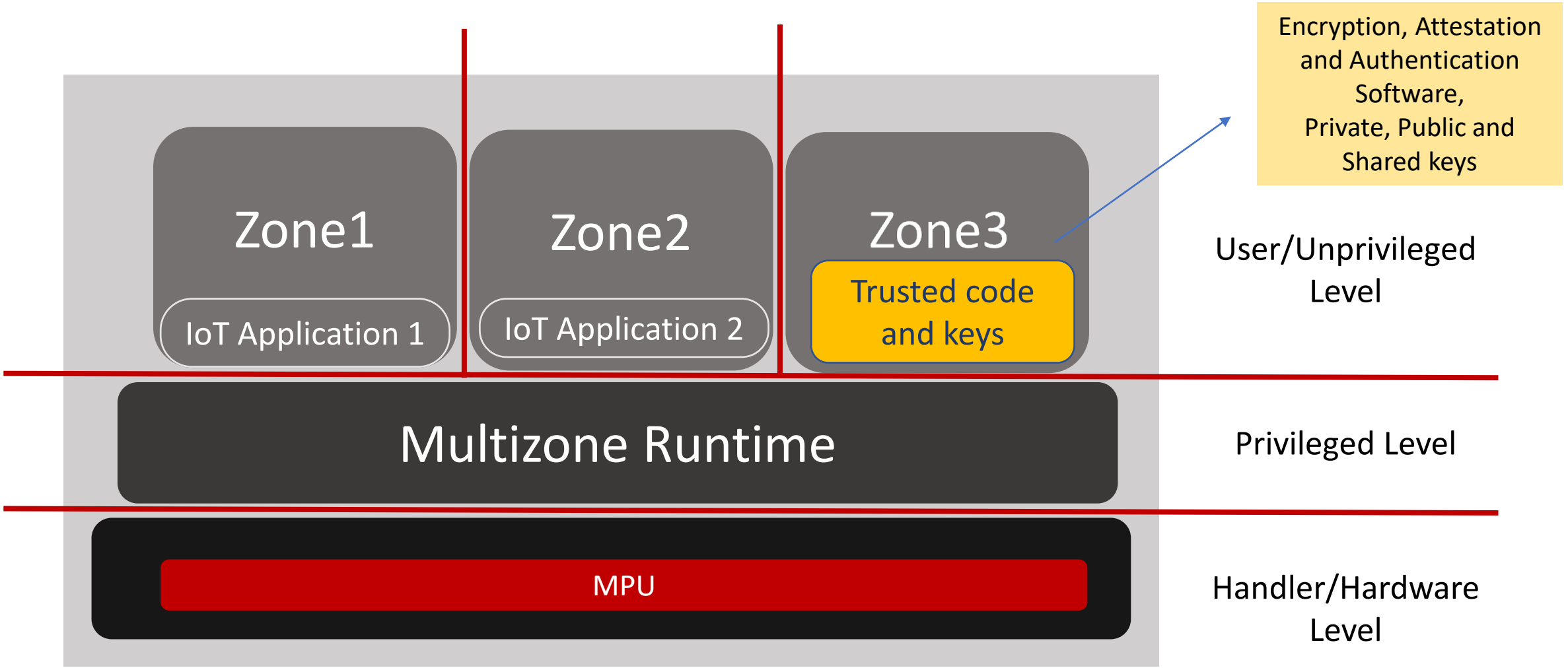
Peraton

How do we implement Secure Memory on the Telemetry Device to store security software? Using TrustZone



Peraton

How do we implement Secure Memory on the Telemetry Device to store security software? Using Memory Protection Unit (MPU)



Peraton

Conclusions

- ZTA based solutions are the basis of modern security architectures
- 5G mobile device flexibility through a ZTA-based security in the testing range opens addressable use cases (IoT, sensing, interchangeable airborne transceiver units)
- Architecture scalable at the Test Range level and across device types
- Automated and centralized device authorization and attestation
- Security Solution integrates seamlessly with a 5G private network in the Test Range