BUILDING TRUST AT THE EDGE: LESSONS LEARNED

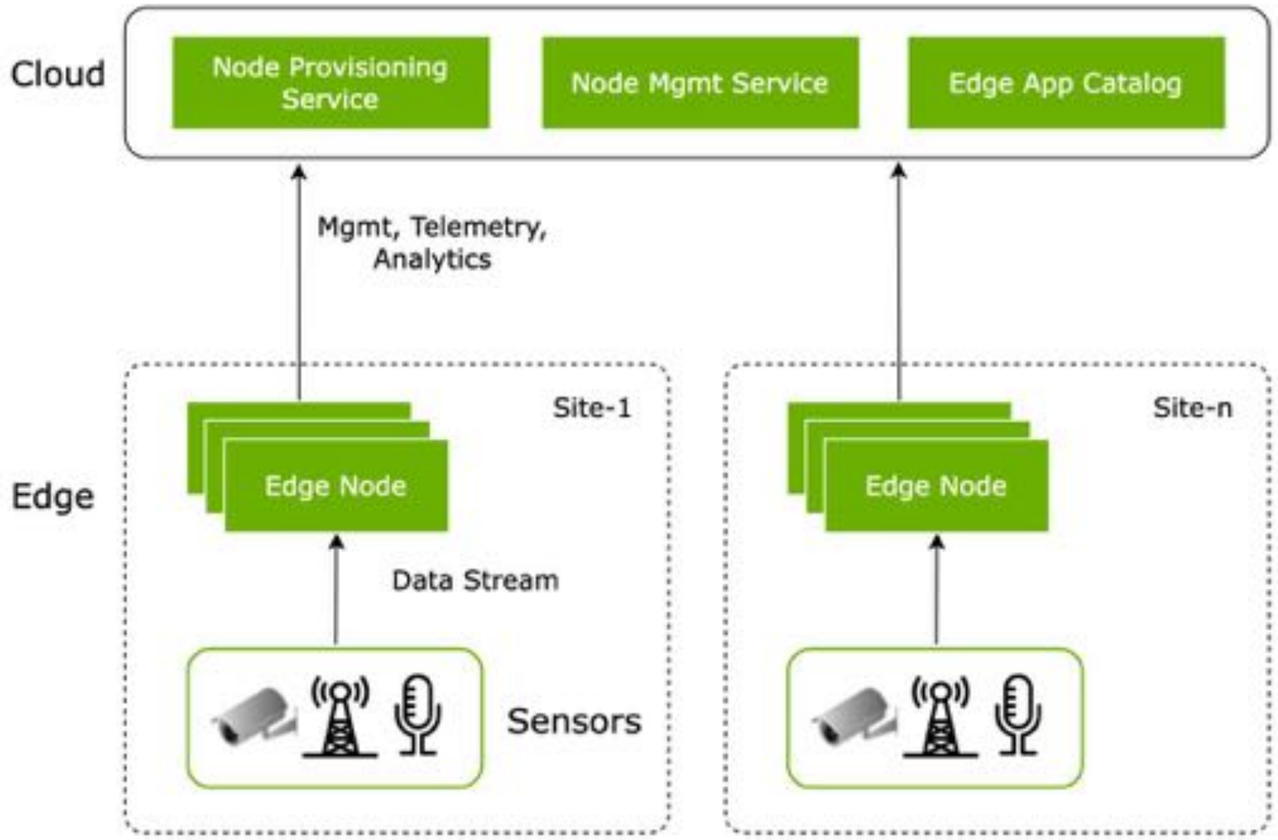
BINU RAMAKRISHNAN
github.com/prbinu

AGENDA

- Edge Computing Overview
- Attack Models & Motivations
- Edge Security Design - Building Blocks
- Orchestrating Security at the Edge
- Challenges

EDGE COMPUTING

- Edge computing brings compute and storage closer to the sources of data for applications that require low latency, high scalability and high throughput.
- Enables applications such as AI inference to process data in real-time - conserving network bandwidth and response time



EDGE COMPUTING ACROSS INDUSTRIES

RETAIL

MANUFACTURING

TELECOM

SMART CITIES

HEALTHCARE

EDGE COMPUTING MODELS

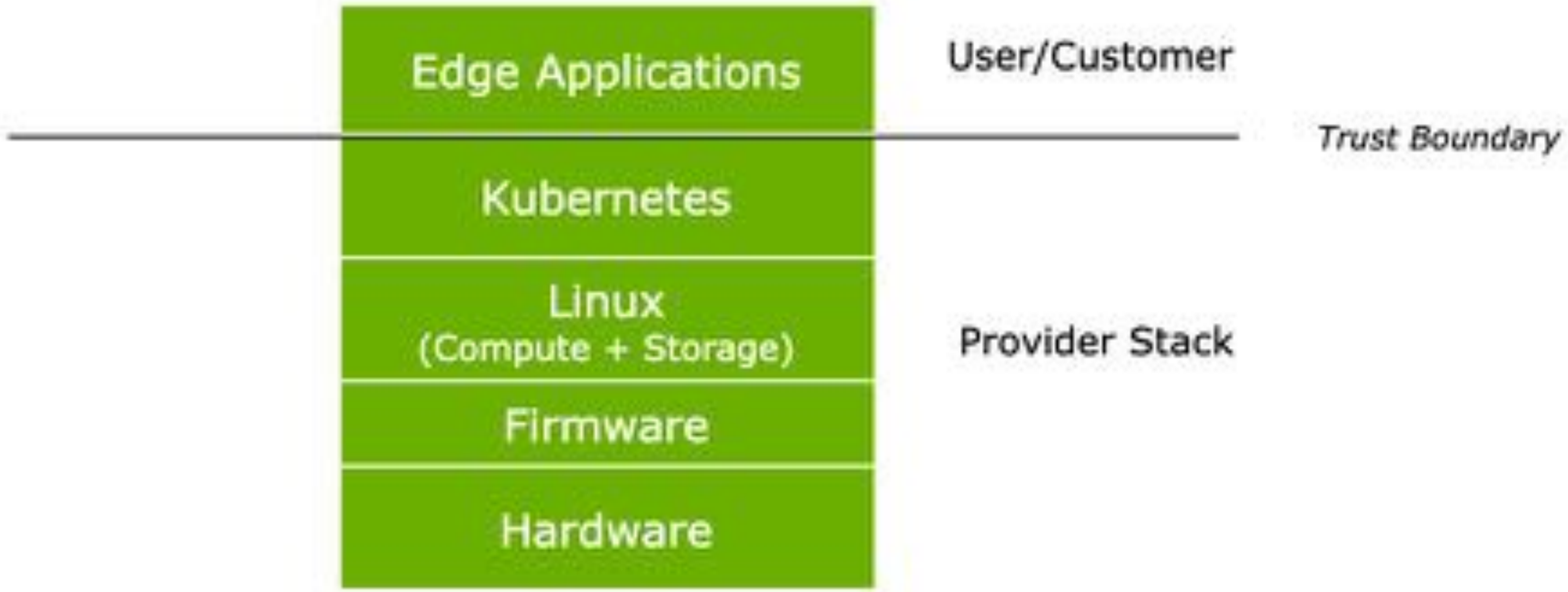
Edge Appliance

- Provides an end-to-end experience to users
- Capable of doing pre-defined set of functions
- Data collected are processed by trusted applications
- Purpose-built
- Example: Smart thermostat.

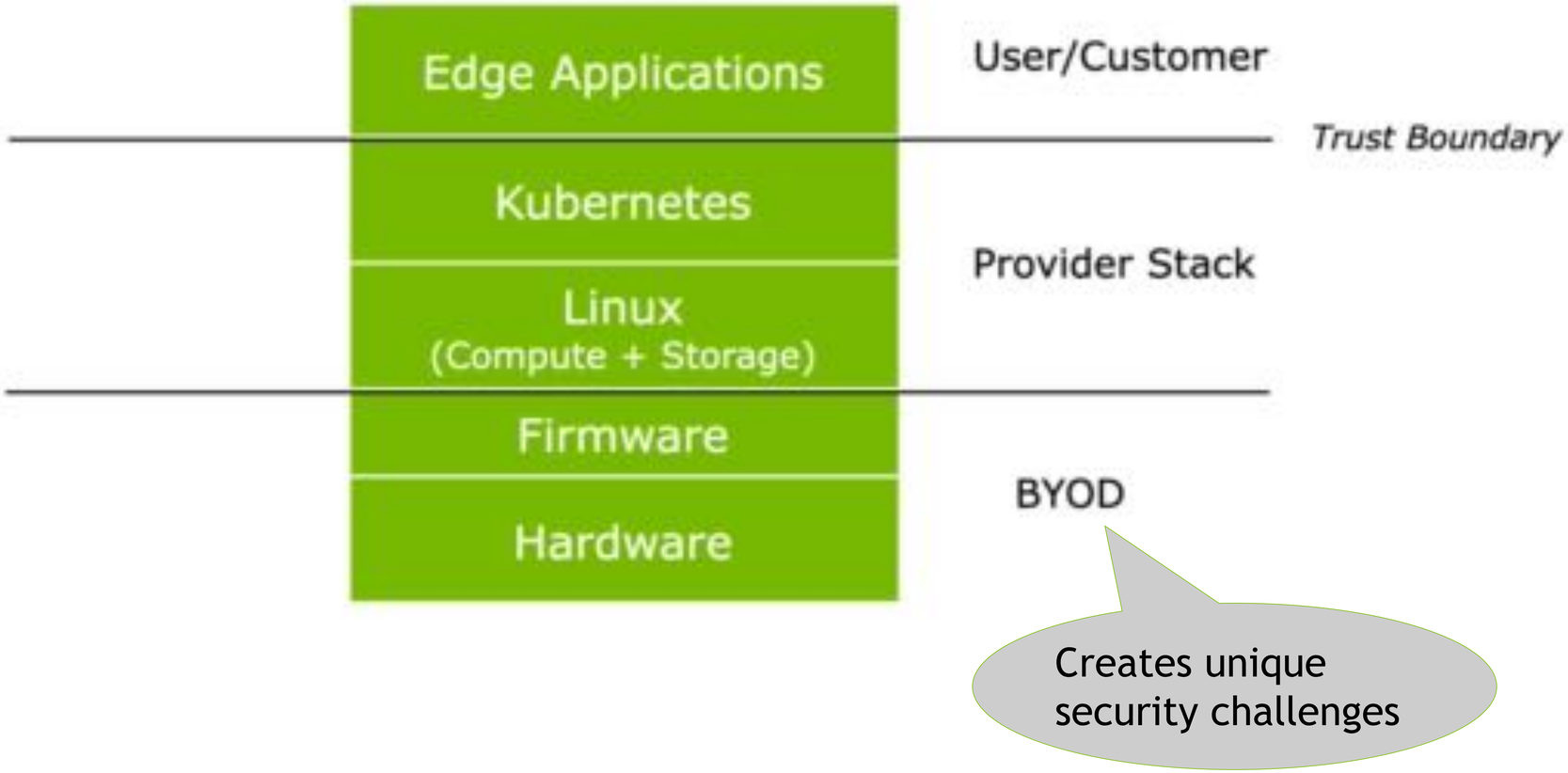
Edge Platform (focus of this session)

- A platform to run customer/user applications
- Flexible - generic platform that supports wide range of applications such as healthcare, retail and manufacturing
- Relatively powerful computers, capable of running applications like AI Inference.
- Data is processed by customer deployed arbitrary (untrusted) software

EDGE NODE STACK



EDGE NODE STACK



ASSUMPTIONS & SCOPE

Assumptions

- A cloud managed egde platform service

Scope

- The edge nodes deployed in the customer premises
- Securing edge nodes from a service provider perspective

Not in scope

- Cloud Security, Security of IoT sensors

EDGE SECURITY

What level of security is desired?

ATTACK MODELS & MOTIVATIONS

THREAT ACTORS

- Nation-state actors
- Organized groups - APT & Cybercrime-as-a-service groups
- Disgruntled employees
- Hacktivists

ATTACK MODELS

- Physical access (local) and remote attacks
- Exploiting vulnerable endpoints and unauthorized access
- Hardware and software supply-chains

MOTIVATIONS

- Data & Intellectual property (eg AI models)
- Ransomware
- Steal compute resources
- Denial of service

IMPACT

- Risk to life (e.g. Medical), disclosure of PII and other sensitive/confidential data
- Financial loss and losing market to your competitor
- Brand
- Disrupt services causing chaos in the operations (e.g. Smartcity env operated by govt/quasi-govt agencies)

EDGE SECURITY: ADDITIONAL THREATS TO CONSIDER

Physical accessibility

- Unauthorized physical access to edges devices
- With physical access, an attacker can mount side-channel attacks such as timing attacks, power consumption analysis, cold-boot attacks etc.
- Modify firmware or exploiting firmware vulnerabilities

Heterogeneous hardware (BYOD)

- Increased threat surface area due to multiple hardware vendors and capabilities
- Lack of standardization in hardware and preboot environment - example - Vendor specific Firmware update process and tools

Shared responsibility of Edge nodes (BYOD)

- Human errors and process gaps - leading to unlocked UEFI/firmware, BMCs
- Are customers patching firmware and other components in a timely manner?

EDGE SECURITY DESIGN: BUILDING BLOCKS

Hardware Root-of-trust (RoT)	Trusted Computing Base	Preboot/Firmware Security
Identity, Authentication & Authorization	Secure Edge Node Enrollment	Encryption & Data Protection
Remote Attestation	Boot-time & Run-time Integrity	Threat Detection & Failure Reporting
Supply-chain Security	Vulnerability Mgmt & OTA Updates	Defense in Depth

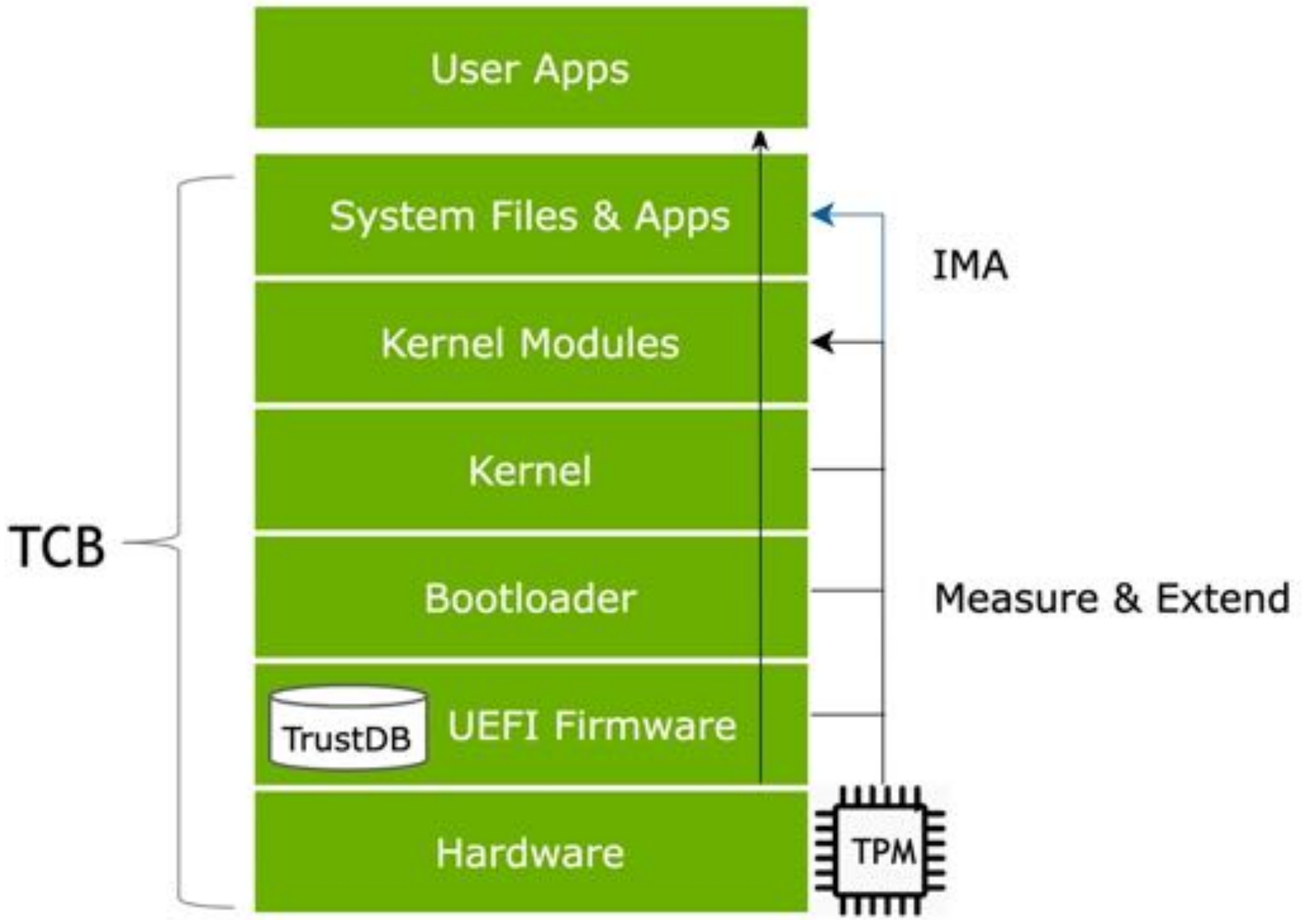
ROOT OF TRUST (ROT) & TRUSTED COMPUTING BASE (TCB)

UEFI & Secure Boot - ensure that the node boots only using software trusted by service provider. If any of the software in the boot chain is compromised, the signature would not match, and the node would not boot the image.

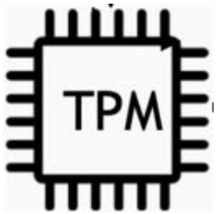
Trusted Platform Module (TPM) & Measured Boot - maintains a cryptographic measurement (hash) of each software layer in the boot chain.

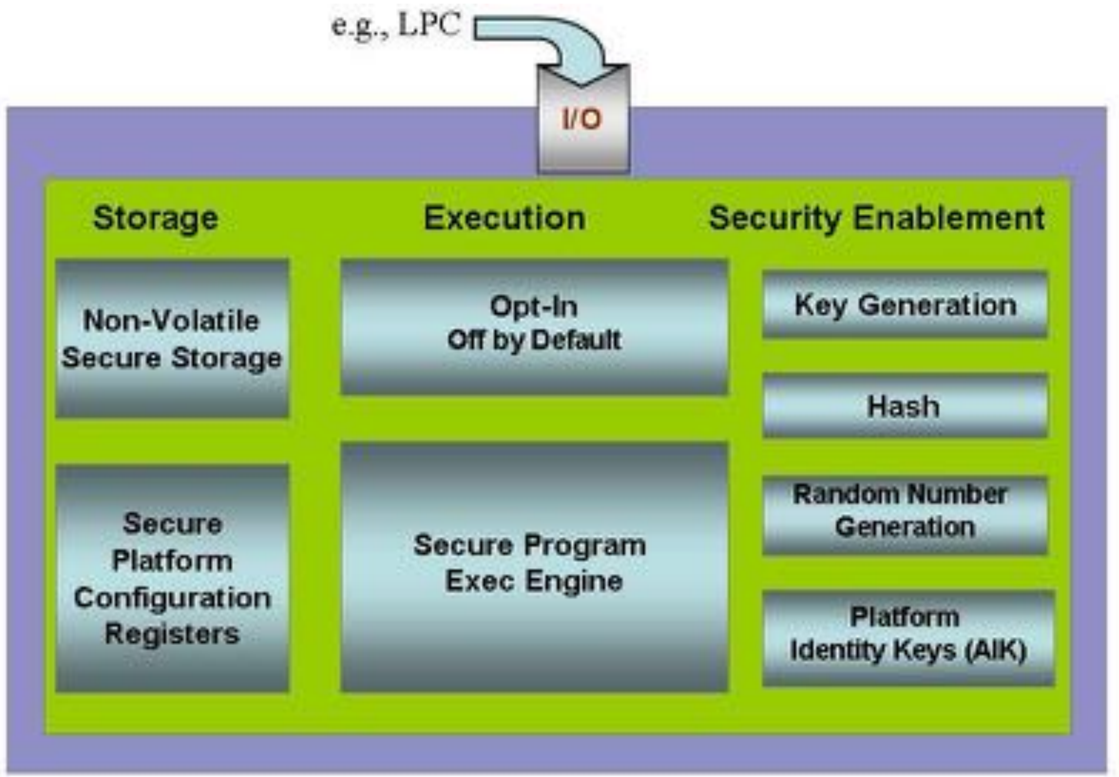
Integrity Measurement Architecture (IMA) - Extend measureboot to enforce run-time file integrity.

TCB - the minimum set of hardened/well-tested hardware, firmware, and software components that are critical to the security of the node



HARDWARE ROT: TRUSTED PLATFORM MODULE (TPM)





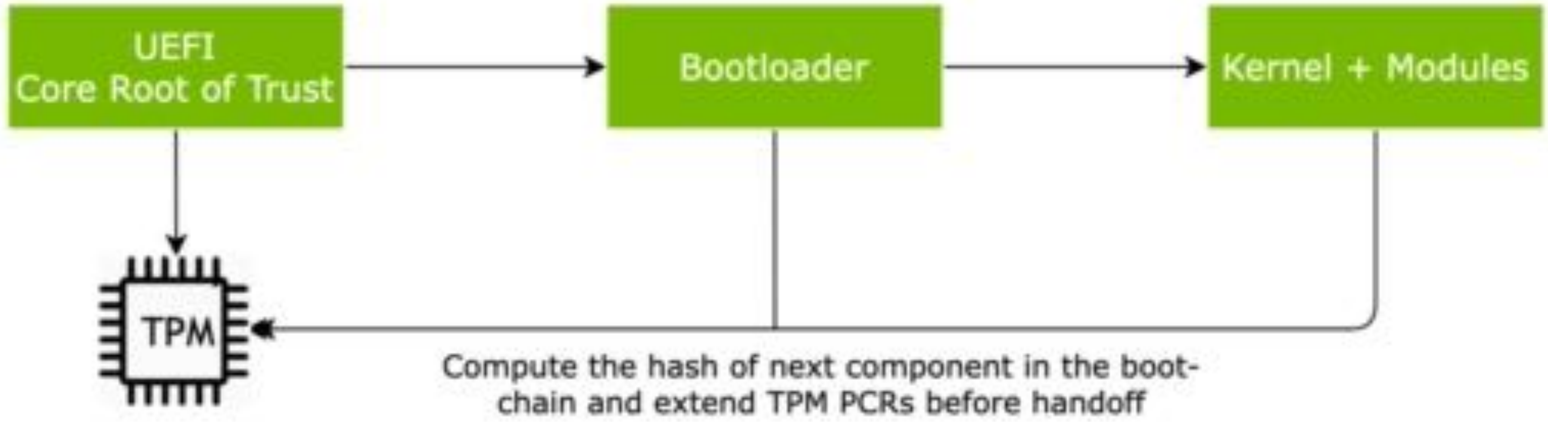
A TPM is coprocessor on the motherboard, which is capable of:

- generating random-numbers, keys and hashes, storing keys,
- Performing cryptographic operations
- Provides Platform Configuration Registers (PCR) to store hashes - used for performing attestation.

Image source https://trustedcomputinggroup.org

SECURE BOOT

MEASURED BOOT



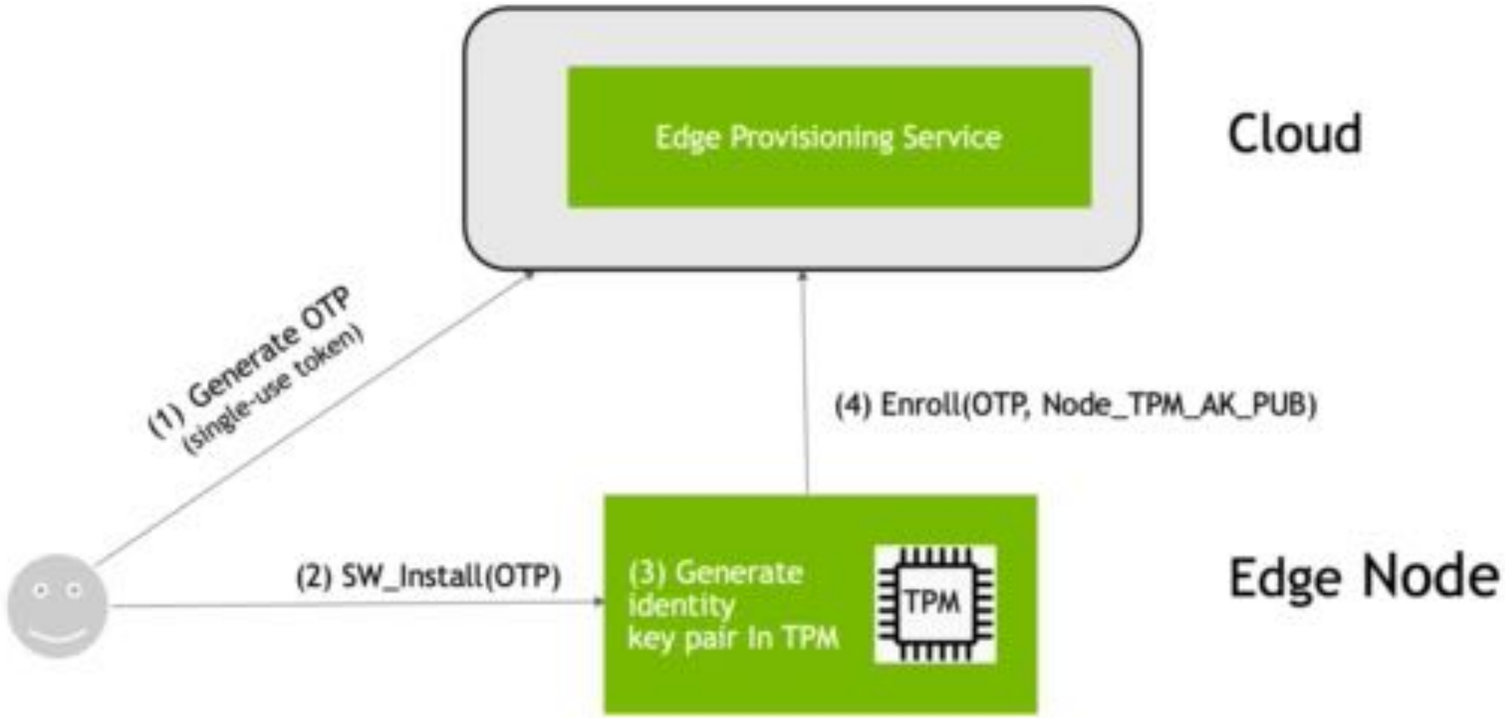
- RoT of Secure Boot and Measured Boot are different
- Secure Boot signature failures abort the boot process
- Measured Boot is passive and doesn’t block the boot process. However it needs an external service to attest the measurements

EDGE NODE ENROLLMENT

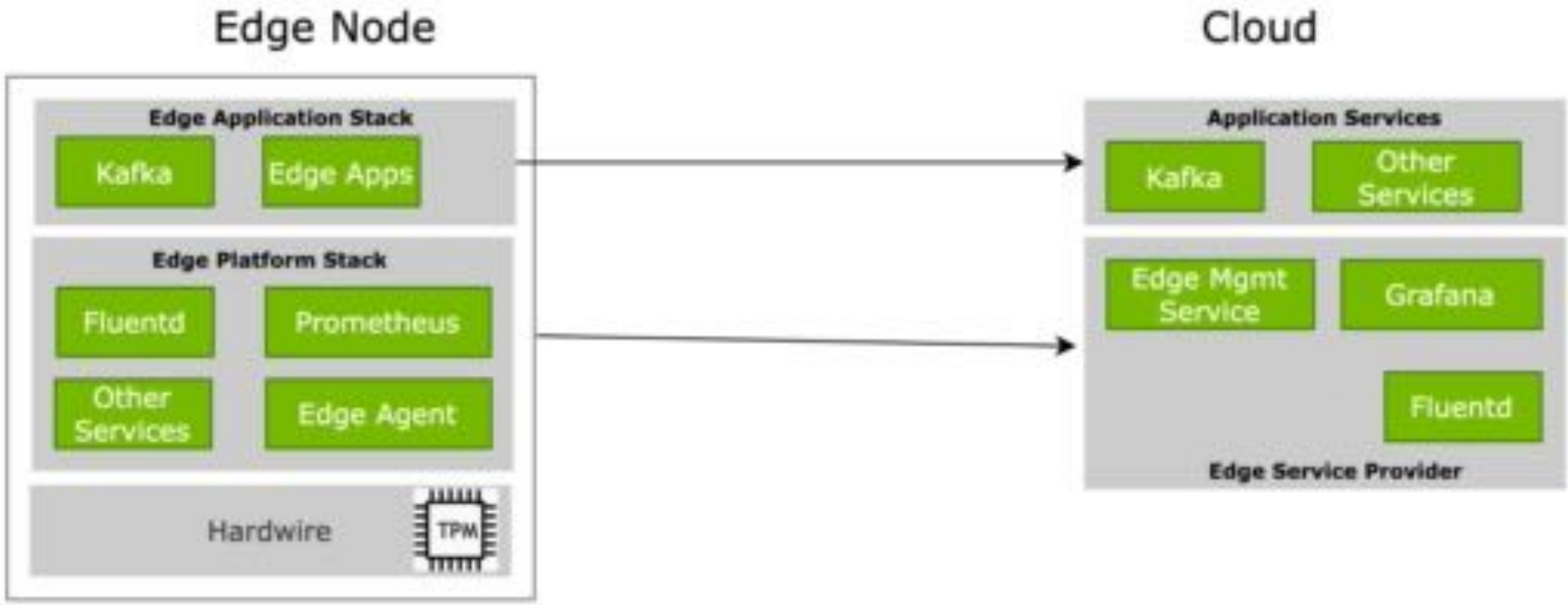
Node enrollment is the process through which new edges node are securely introduced to the provider’s cloud platform.

This process uses a short-lived one-time code generated by cloud provisioning service that binds TPM generated identity key with the newly enrolled node. Subsequently this key is used for edge node authentication with cloud services.

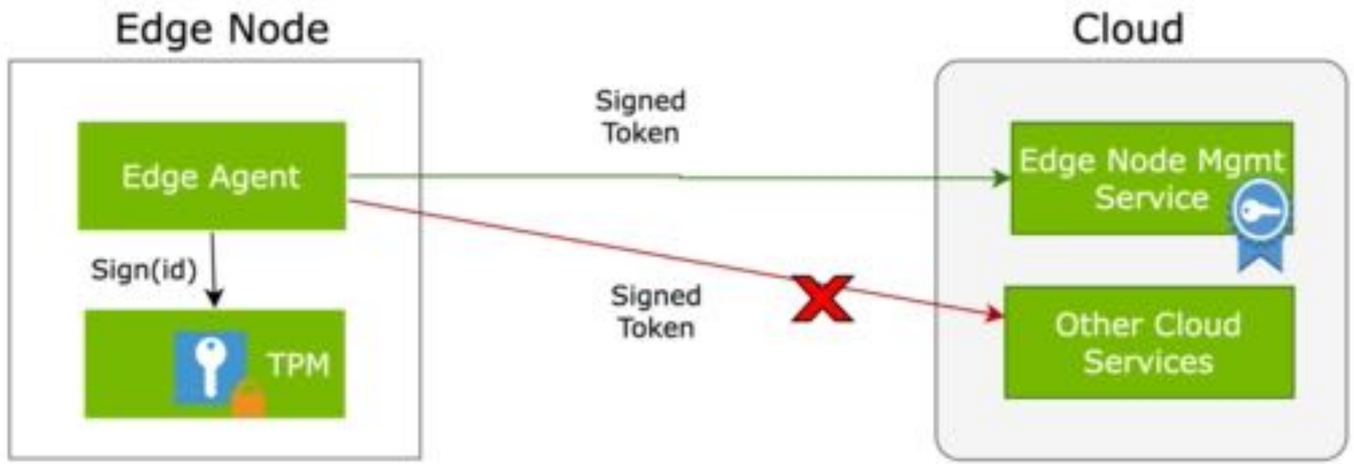
The TPM protects cryptographic keys and operations, and prevents attackers stealing keys, and device cloning.



AUTHN & AUTHZ: USE CASES



EDGE NODE IDENTITY

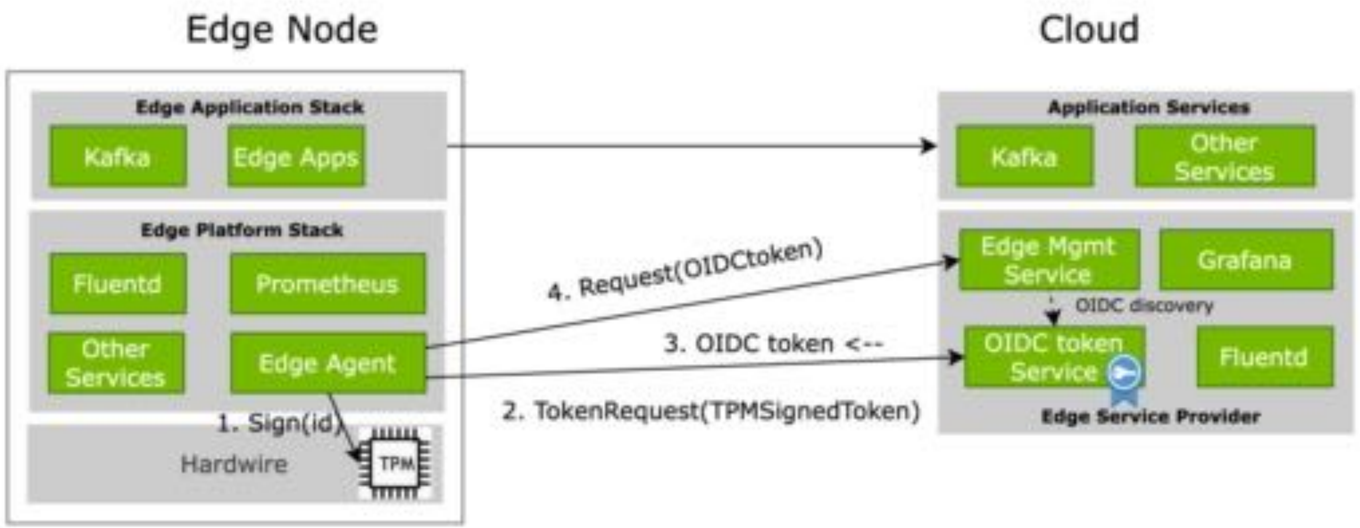


The Edge Node Mgmt Service already has the TPM public key (registered as part of node enrollment), hence it can verify the edge TPM signed requests.

However, this would not work with other services because those services don't have the edge TPM public key for validating the token.

To remedy this issue, introduced a cloud OIDC token service that issues ephemeral tokens to use by any OIDC compliant services.

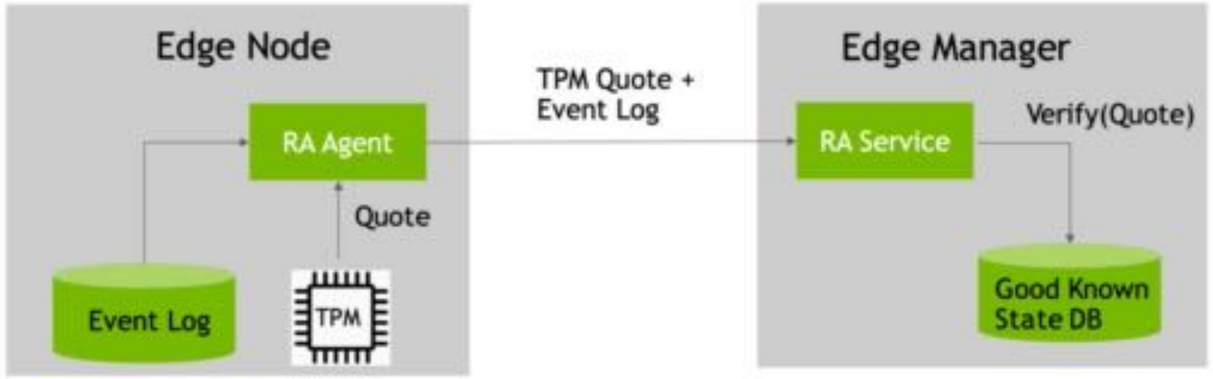
IDENTITY, AUTHN & AUTHZ: EXAMPLE



1. Edge Agent requests TPM to sign token (with a nonce provided by the server, not shown in this diagram)
2. Edge Agent makes a request to OIDC token service.
3. OIDC token service validate the TPM signed token, and if valid, it issues an ephemeral OIDC token to Edge Agent
4. All subsequent communications by Edge Agent with Edge Mgmt Service are now authenticated by this token

Alternatively, step 1 & 2 can be replaced with mutual TLS with TPM-backed key using PKCS#11 interface.

REMOTE ATTESTATION (RA)



RA is a mechanism to verify the correctness of the software and hardware configurations of a remote system. It ensures that the connecting device is a legitimate edge node and is in known valid state before delivering service.

Compute the initial reference hash, aka the golden PCR hash, and upload it to the edge manager

When the edge node is booted, it sends a Quote — a PCR measurement signed by TPM Attestation Key (AK). The value is compared with a previous known secure state. A change in boot configuration modifies the PCR values.

As a run-time extension to attest IMA file integrity measurements, RA can be performed every time the Edge Agent requests new OIDC token from Edge Manager.

DATA SECURITY

- Data at rest
  - dm-crypt, keys stored in the TPM
- Data in transit
  - TLS/HTTPS
- Secure data deletion (node repurpose/retire)

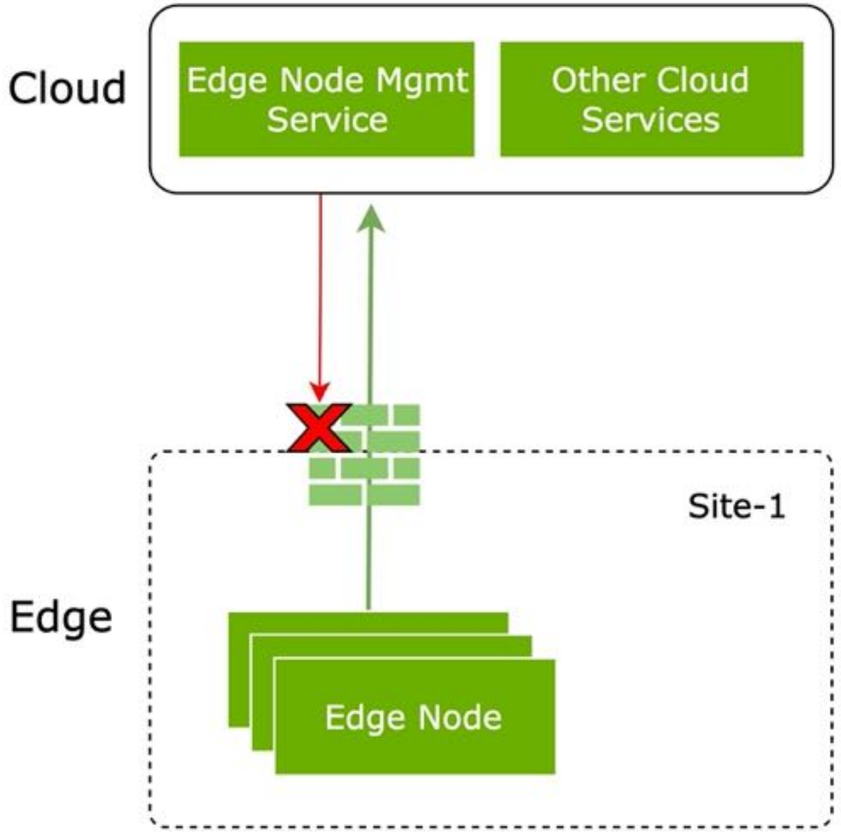
TLS between edge and external/cloud services is critical to protect against man-in-the-middle attacks.

NETWORK

Punching holes into customer Firewalls to allow inbound connections to Edge node listeners are easier said than done.

Recommendation:

- Since Firewall policies generally allow outbound traffic (say to TCP port 443), always initiate connections from Edge nodes.
- Also it is a security best practice not have listeners on Edge nodes



RUN-TIME INTEGRITY & THREAT DETECTION

Defines the appropriate activities to identify the occurrence of a security event

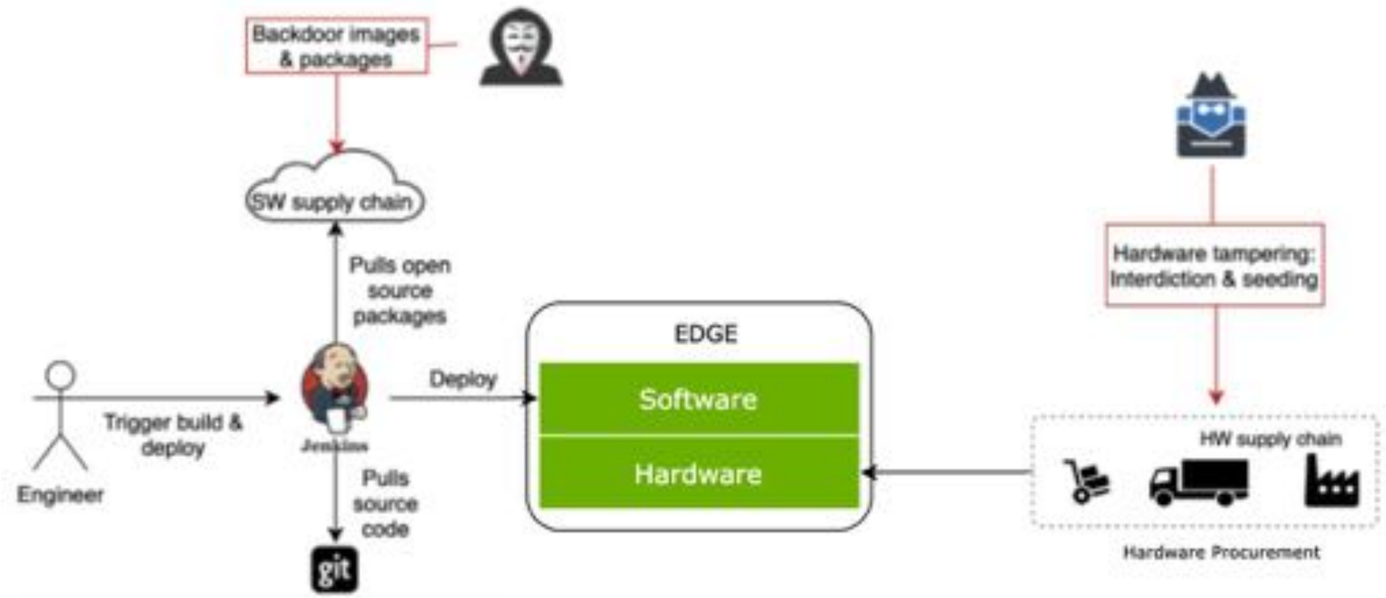
Examples:

- Reverse Shell - An attacker spawns a reverse shell
- Reverse DOS
- Bash script exec - detect an unauthorized script execution
- Detect CVE exploitation (e.g. log4j / LogShell vulnerability)
- Brute-force SSH - repeated failed login attempts
- Cryptomining - lookup for mining domains and IP addresses, Stratum protocol
- Data exfiltration / Unauthorized file access (e.g. secret files)

Technologies:

- Falco
- eBPF/KRSI

SUPPLY-CHAIN SECURITY



Best practices and technologies to consider

- Source cautiously and avoid adding external dependencies if there are other options.
- Scan for vulnerabilities in open source software - CVEs, Malware etc.
- Secure distribution of software (over TLS), publish checksums
- Artifact/image signing. Sigstore/Cosign (sigstore.dev) offers tools for signing, verifying and protecting software
- Build supply chain provenance. In-toto (in-tot.io) and SLSA (slsa.dev) are two popular options in this space.

VULNERABILITY MANAGEMENT & OVER-THE-AIR (OTA) UPDATES

Vulnerability Management is an continuous function to identify and patch vulnerable packages in a timely manner. OTA acts as a vehicle to patch edge systems in a secure and reliable fashion.

OTA update is initiated in the event of:

- Remote Attestation failures that trigger a system update process to rollback to a known secure state
- Open source vulnerability disclosures that trigger OTA security patch update
- Security attacks, and recovery

Note:

OTA may alter the TPM PCR measurements, hence we need to recompute and update the Golden measurements in a reliable manner.

DEFENSE IN DEPTH

Auth

Use of ephemeral tokens and certificates

RBAC for customer, remote console for device administrators

Isolation

Host and application: Kubernetes Pod/Docker, use of non-privileged containers for application workloads

Between applications: Kubernetes Namespaces

Hardened OS (CISCAT):

Read-Only filesystems for system software partitions

Separate encrypted data, /tmp and /var partitions

Stripped down OS image to reduce attack surface area

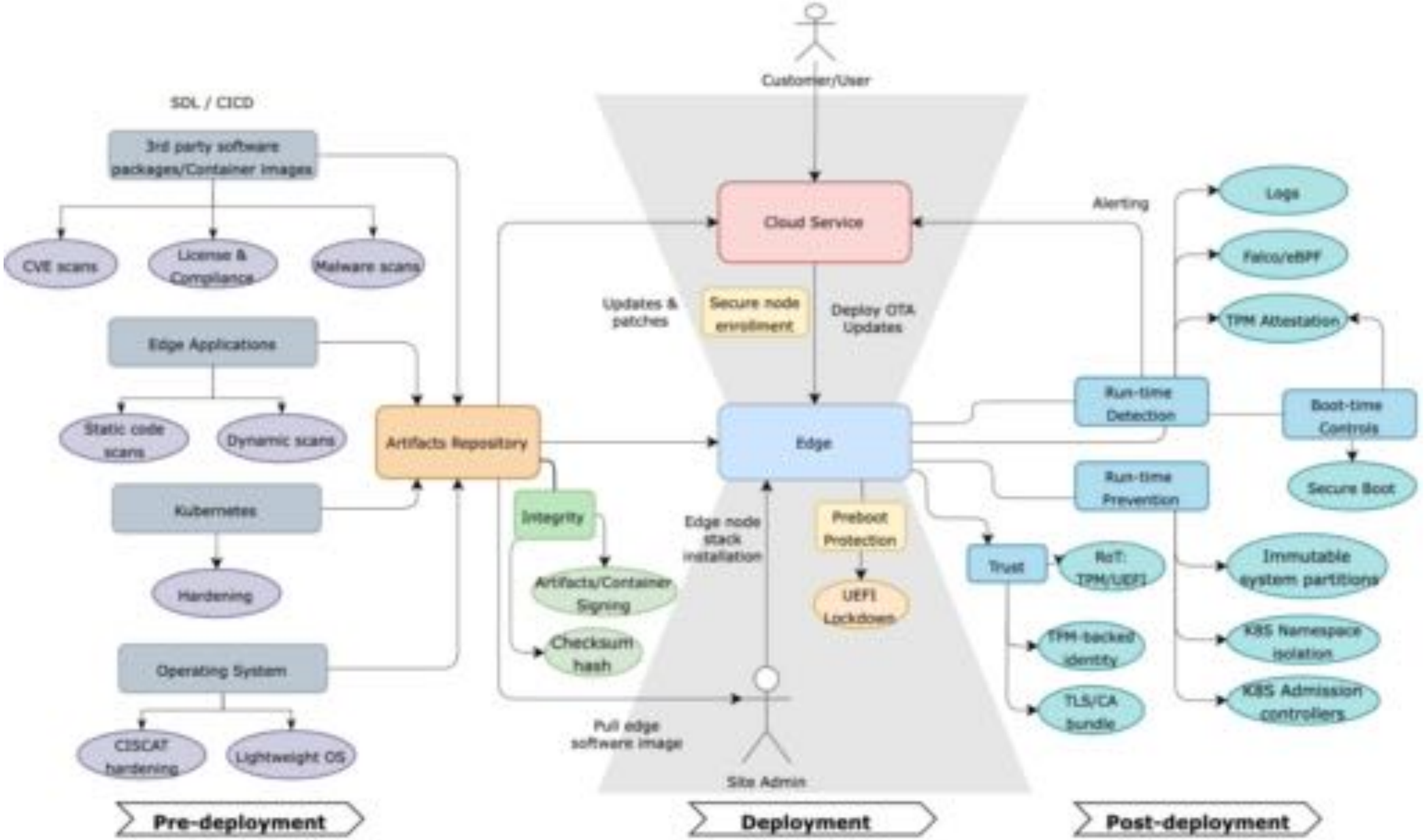
Hardened Kubernetes stack (CISCAT)

RBAC policies

Admission Controllers to enforce user application policies (e.g. run as unprivileged user, port exposure policies, Apparmor etc)

Dedicated namespaces for customer workloads

SECURITY ORCHESTRATION - A HIGH LEVEL RECAP



CHALLENGES

Building trust on disparate pieces of hardware and software provided by a variety of vendors is an uphill task.

Here are my observation on few issues we came across as part our journey

- Secure Boot - If you have custom Kernel Modules or dependent on 3rd party modules, then making Secure Boot work would be hard without updating UEFI key database, but updating it needs physical presence.
- Stability of TPM PCR banks are not consistent across different OEM hardware
- Lack of security standardization in Firmware / preboot environment
- Supply chain attestation is in its early stages, and not ready for a wide adoption
- Artifact signing - package checksum works, container signing is doable, Signing and verification of other artifacts types are not trivial.

ACKNOWLEDGEMENTS


NVIDIA