

# Azure Sphere: MCU, OS & Security Services

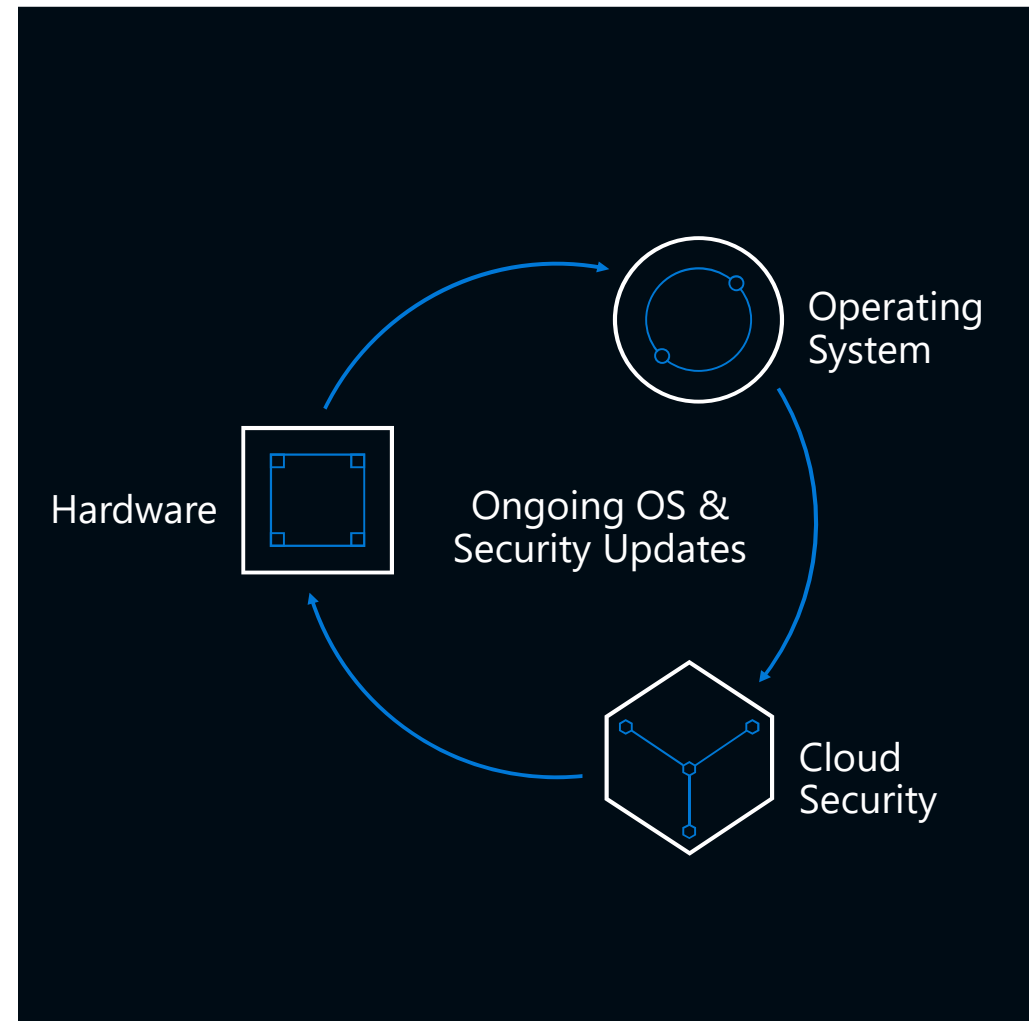
Tan Chee Weng  
Sr Technical Specialist



# Azure Sphere

An end-to-end solution for securely connecting existing equipment and to create new IoT devices with built-in security. Put the power of Microsoft's expertise to work for you everyday.

- Azure Sphere certified chips
- The Azure Sphere Operating System
- The Azure Sphere Security Service
- Ongoing OS and Security Updates

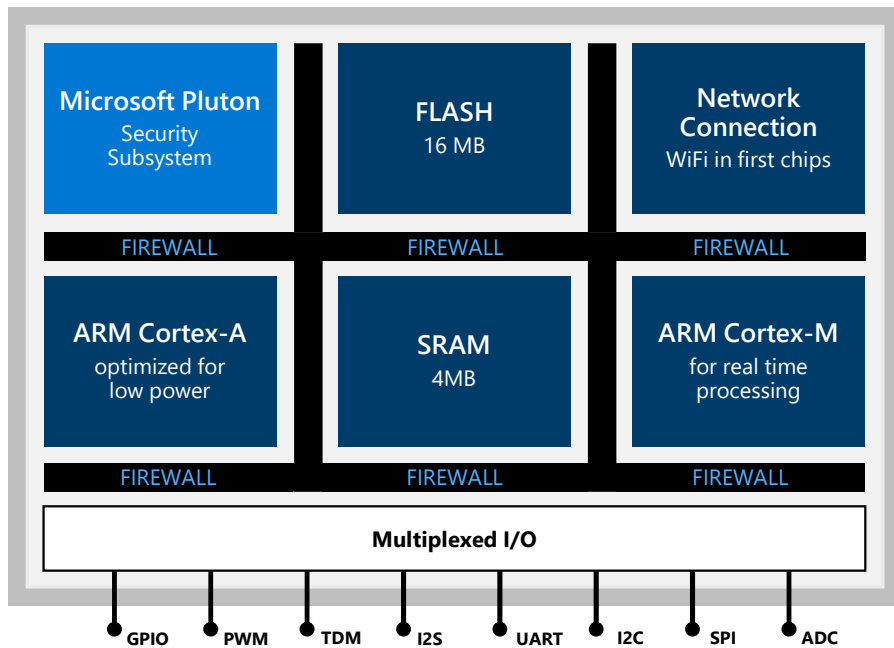


Over 10 years of security and OS updates delivered directly to each device by Microsoft



# Azure Sphere MCU Architecture

# Azure Sphere certified SOCs create a secured root of trust for connected, intelligence edge devices



## Connected

with built-in networking

## Secured

with built-in Microsoft silicon security technology including the Pluton Security Subsystem

## Crossover

Cortex-A processing power brought to MCUs and crossover SOCs for the first time

# Azure Sphere: A Secured, Connected, Crossover, MCU



## Security

- Process Level **Isolation** via Memory Management Unit (MMU)
- Azure Sphere OS: **Specialized IoT OS** running custom Linux with special IoT functionality + Azure Sphere Runtime.
- Client / Server Certificate based **authentication** for cloud communication
- **Authorization** of access to resources via custom capability secured by Pluton

## Portability

- **HW Abstraction** via address space virtualization (via MMU)
- Open Source SW (OSS) libs written against POSIX standards, hence ease **portability** of OSS SW to your application platform

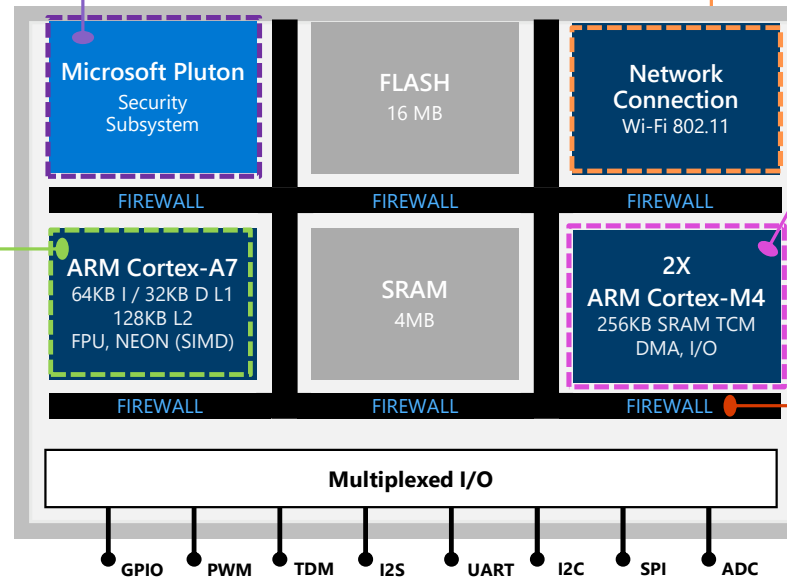
## Extensibility

- Enable post sale monetization with new customer experience with scalable SW update safely, securely
- Headroom for new product features and customer experience.

- HW Root of Trust
- Warrant authenticity of device HW + SW
- HW Anti-rollback protection
- Protect against low entropy attacks
- Firewalls, HW Crypto Accelerator: AES-256, SHA-2, ECC, RSA2K, ECDSA
- e-Fused private and public keys,

Built-in WiFi connectivity 802.11 b/g/n, dual band @ 2.4GHz, 5GHz

- Real time core targeted at real-time workload and interaction with peripherals; secured by Pluton.
- Manufacturers are free to run any Cortex-M runtime. Microsoft will provide a reference M4 runtime.



- Silicon level firewalls that implement the principle of least-privilege. Only grant access to resources that is given explicit permission.
- This principle applies to every resource in the system: RAM, network, flash and peripherals.
- Compromised software cannot access new resources.
- Firewalls are **sticky**. Should the controls to firewall is compromised, it is not possible to reconfigure until the chip is reset.



# Azure Sphere Operating System

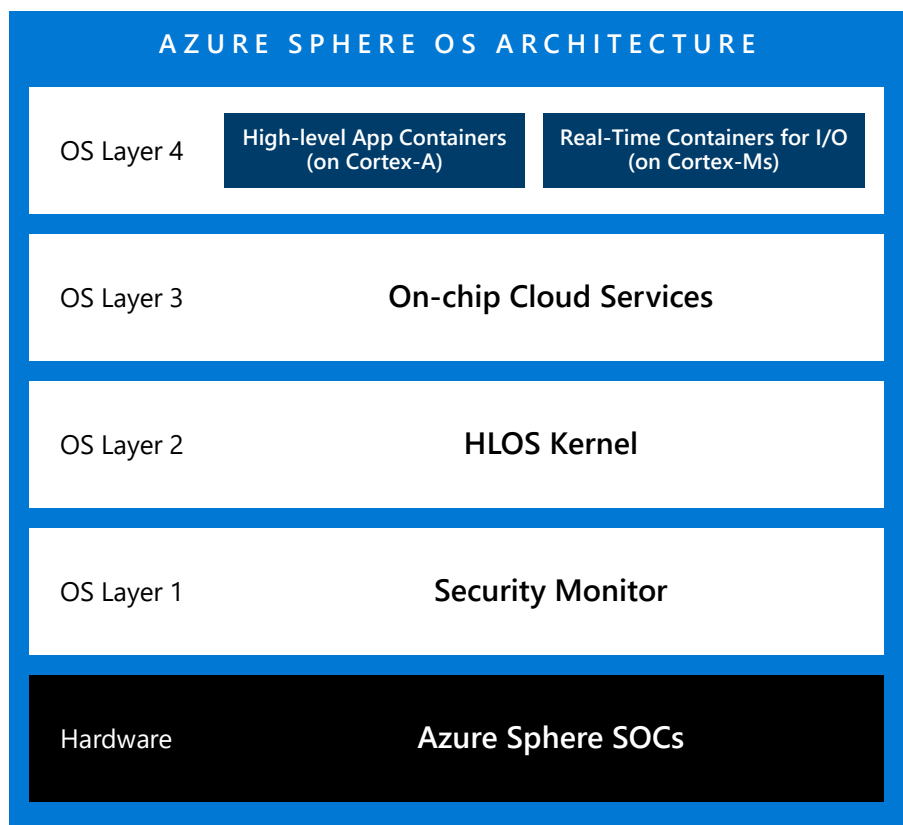
# Azure Sphere OS: Design Goals

## The Azure Sphere OS

- is **secured** by designing in the *7 Properties of Highly Secured Devices*.
- enables **productivity** with an application model for reusable connectivity and I/O solutions.
- creates **opportunity** by optimizing for the memory and flash sizes of Azure Sphere crossover MCUs, and extending to support an agile ecosystem of diverse Azure Sphere silicon, hardware, and software.

# Azure Sphere OS: Architectural Layers

- Layered architecture supports *defense in depth*, *compartmentalization*, and small *trusted computing base*.
- Layers are *independently updatable*.



## Secure Application Containers

Compartmentalize code for agility, robustness & security

## On-chip Cloud Services

Provide update, authentication, and connectivity

## Custom Linux kernel

Empowers agile silicon evolution and reuse of code

## Security Monitor + Pluton Security Subsystem

Guards integrity and access to critical resources

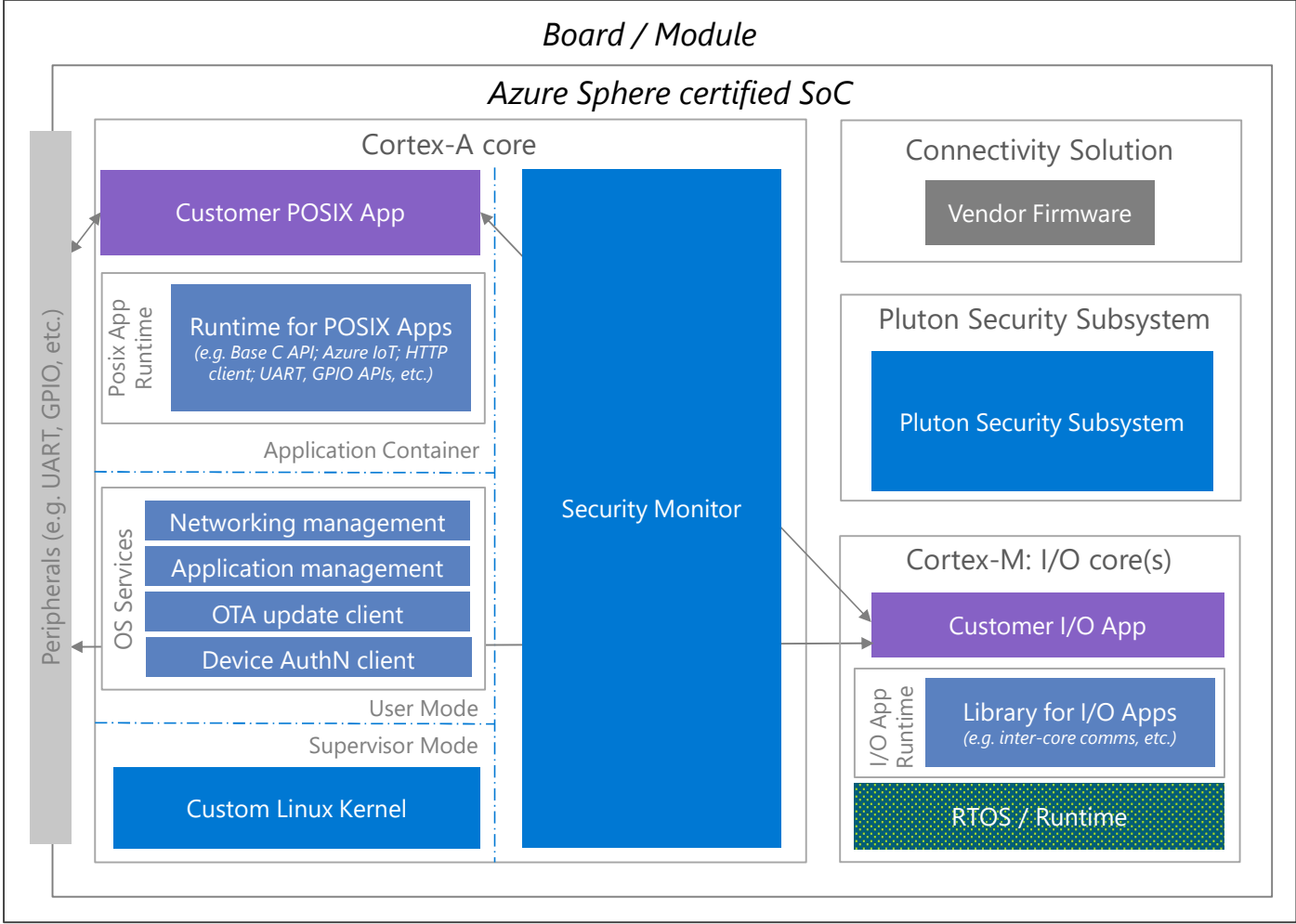
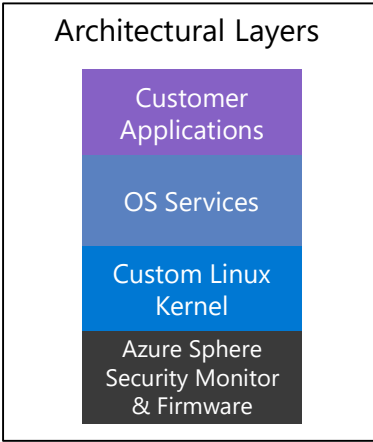
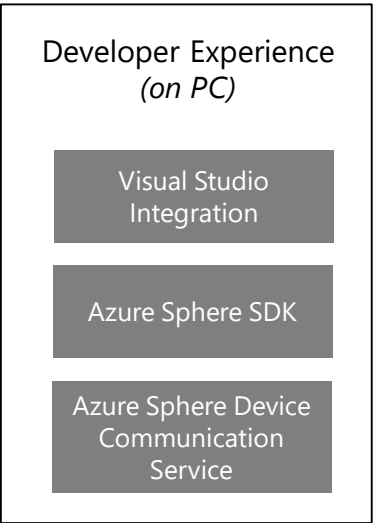


# Azure Sphere OS: Basic Principles - Security

*The Azure Sphere OS is secured by designing in the 7 Properties of Highly Secured Devices.*

- Security Monitor as small trusted computing base.
- Clear separation between apps and OS to compartmentalize and provide defense in depth.
- Modern capability-based app containers that isolate apps from each other and fit within the constrained memory of a crossover MCU.
- No passwords, no user accounts, no logon, no command shell; certificate-based authentication tied to hardware-rooted identity.
- Secure boot sequence and secured, over-the-air (OTA) update process to ensure software authenticity and renewable security.
- OS and apps are independently updatable to maximize agility to respond to security and quality issues on both sides. No need to recompile one to update the other.
- App failure detection to mitigate threats.

# Azure Sphere OS: Basic Architecture Deep Dive



# Trusted Computing Base: Pluton + Security Monitor

- **Uses Pluton to implement Secure Boot, ensuring the integrity and authenticity of the Azure Sphere OS and OEM apps.**
  - Uses Pluton silicon to validate all software loaded on an Azure Sphere MCU is cryptographically signed by Microsoft.
  - OEM applications are signed by uploading to the Azure Sphere Security Service (AS3) using tools in the Azure Sphere SDK.
  - Uses Pluton silicon to attest the Azure Sphere MCU's installed software is genuine and trusted; leveraged by OS services for authentication and remote attestation of device with the AS3
- **Controls access to memory, flash and other shared MCU resources.**
  - Apps cannot arbitrarily scribble over flash; their binaries and static content are mounted for read-only access.
  - Configures and locks down silicon firewalls and memory access to isolate and constrain cores' access to shared chip resources.
  - Brokers and gates access to Pluton Security Subsystem and the hardware root of trust.
  - Controls debug prep and field prep for pre-production development scenarios.

# OS Services + Compartments for POSIX Apps

- **OS Services are responsible for:**
  - Hosting POSIX app compartments.
  - Communicating with the Azure Sphere Security Service (e.g. update, authentication).
  - Managing Wi-Fi authentication; managing network firewall for all outbound traffic.
  - Communicating with a connected PC and debugging apps (developer scenarios only).
- **POSIX app compartments:**
  - Designed for crossover MCUs, but take inspiration from modern app containers present on larger systems (i.e. Docker, mobile apps).
  - Isolate apps from the OS.
  - Constrain apps' access to only the capabilities (e.g. peripheral I/O, network, etc.) they request.

# Compartments for I/O apps

- I/O apps running on Cortex-M cores:
  - Will enable deterministic execution of application code.
  - Are isolated from the rest of the MCU via silicon firewalls.
  - Bare metal support added with 19.05.
- **Currently, a Reference RTOS and Real-time SDK has been released for the Cortex-M4 cores**
  - Customers may optionally adopt this RTOS to accelerate getting started.



# Azure Sphere Security Services (AS3)

# The Azure Sphere Security Service connects and protects every Azure Sphere device

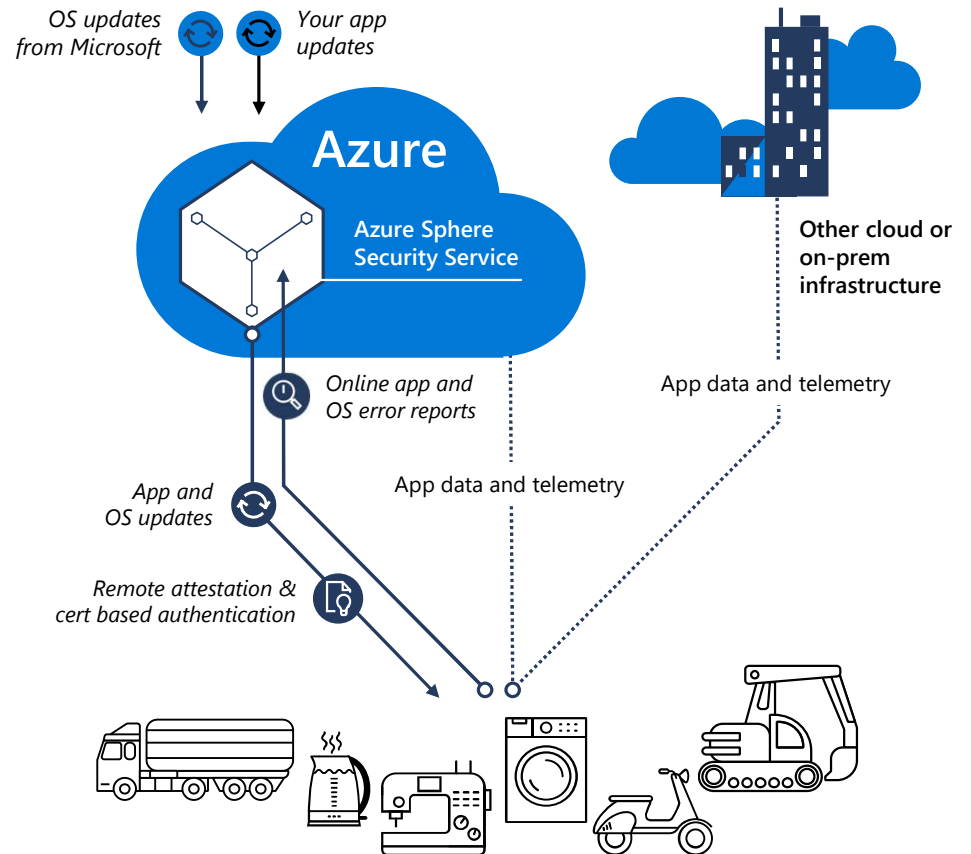
**Protects** your devices and your customers with certificate-based authentication of all communication

**Detects** emerging security threats through automated processing of on-device failures

**Responds** to threats with fully automated on-device updates of OS

**Allows** for easy deployment of software updates to Azure Sphere powered devices

**Cloud choice** for app data and telemetry



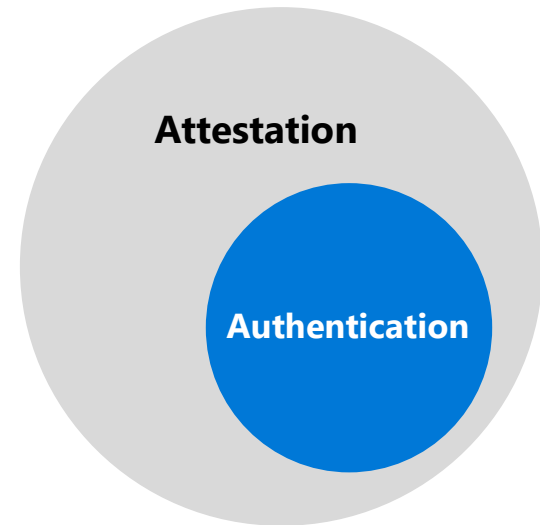
# Authentication vs Attestation

Device authentication:

*Q: Is this my device X?*

Device Attestation:

*Q: What is the state of my device X?*



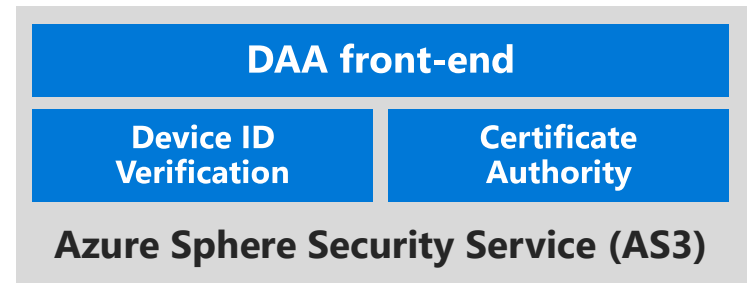


# Device Authentication and Attestation (DAA)

## Main components

DAA front end

Device authentication & attestation

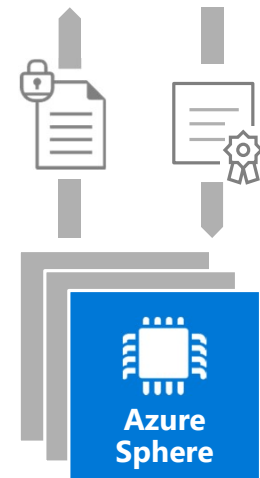


# Device Authentication and Attestation

- The device send a set of signed measurement data (device ID, nonce, software components version, etc) to the DAA service using the private key.
- DAA validate the device signature by recompute the measurement data and ensure the results are the same to ensure it is a trusted device
- Additional checks are done to ensure the version of the software is still trusted.
- Should the device running untrusted software, certificate will not be issued and OS update via OTA will be required
- The absence of a valid certificate will prevent a device from authenticating to a service requiring authentication until the device software is updated.
- If all the checks pass, the device is issued a short-lived X509 certificate.

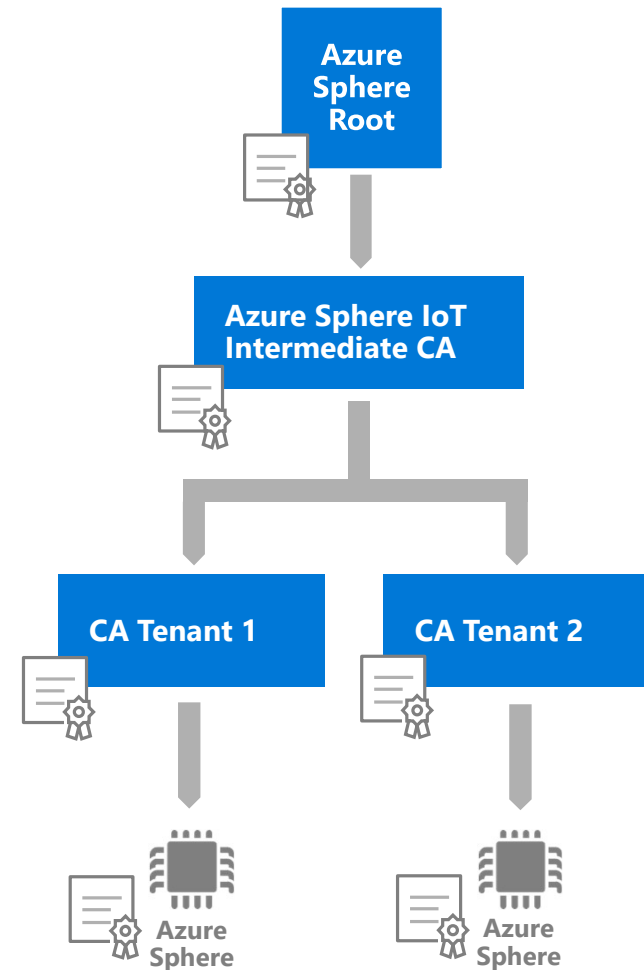
## Device Authentication and Attestation

Is this device genuine?  
Is this device trusted?



# DAA PKI chain-of-trust

- In the Azure Sphere Security Service, an Azure Sphere tenant represents a group of devices.
- The tenant not only contains information about the devices, but also provides an isolation boundary for management of these devices.
- Each device has a unique ID which is “claimed” to a tenant
- The DAA certificate services operate on a per-tenant basis
- Each device that is registered to a tenant receives a certificate that is valid only within the tenant-specific chain.



# X509 device certificate

- **Not customizable**

- They are short lived certificates (24 hours)
- Device authentication certificates have no CRL

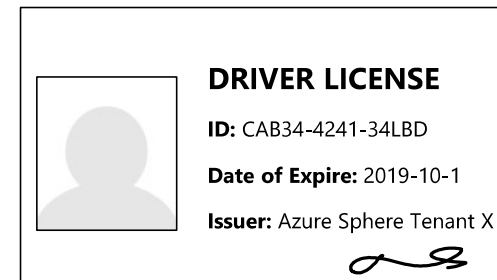
- **Lifetimes:**

- Root CA: 20 years
- Intermediate CA: 10 years
- Tenant CA: 2 years
- Device certificate: 24 hours

- **Must be regular renewed :**

- Renew certificate upon expiration after passing remote attestation process regularly

Field	Value
Subject Name	Device ID
Key Usage	Digital signature
Enhanced Key Usage	Client authentication
OID: 1.3.6.1.4.1.311.98.1	Remote attestation data



# Goals of Azure Sphere renewable security

System Security is NEVER constant and IoT brings a new set of challenges to embedded devices.

Maintain security of all software

Microsoft manages backwards compatibility

Enable OEMs to release software updates

Minimize downtime

OEMs are only responsible for their application

# The Azure Sphere Security Service renews device security

## Secure Over-The-Air (OTA) updates infrastructure

- Cloud infrastructure can deliver updates to Azure Sphere devices around the world.

## Robust application deployment and updates

- Customer-written applications are signed, deployed and updated by the customer using the Azure Sphere cloud.
- Applications can be revoked, removed, and rolled-back to mitigate buggy updates.
- Attestation authorizes only genuine software to execute on device.
- Update software supports automatic rollback to help prevent a bad update from disabling (aka “bricking”) a device.

## Reliable System software updates

- Microsoft automatically manages updating device software to help ensure secure device operation.
- Updates are delivered privately to device creators first to test updates.

# The Azure Sphere Security Service helps identify emerging security threats

## Observation:

- Device software/firmware failures in the field are triggered either by
  - imperfect programming for an extremely rare sequence of events, or
  - attackers probing for attack vectors
- In either case, we want to know and fix the issue.

**On failure in the device software, a report is generated and transmitted to our cloud-based failure analysis system.**

**The failure analysis system correlates error reports on a global scale.**

- For imperfect programming, we gather data to fix problem.
- For attackers, this early warning system often indicates where and how attackers are working, we generate mitigations and fix problems in our software.



# Azure Sphere

## Device Management & SW Lifecycle



# Azure Sphere: Device Management & SW Lifecycle Terminology

- **Product**

- Your appliance

- **Device-Group**

- Logical grouping of devices of a product

- **Applications**

- Your "Application"

- **Images and Image-Packages**

- Image is a version of your Application,
- Image-Package combines image and associated metadata

- **Deployment**

- Targets Image(s) to a device group of a product

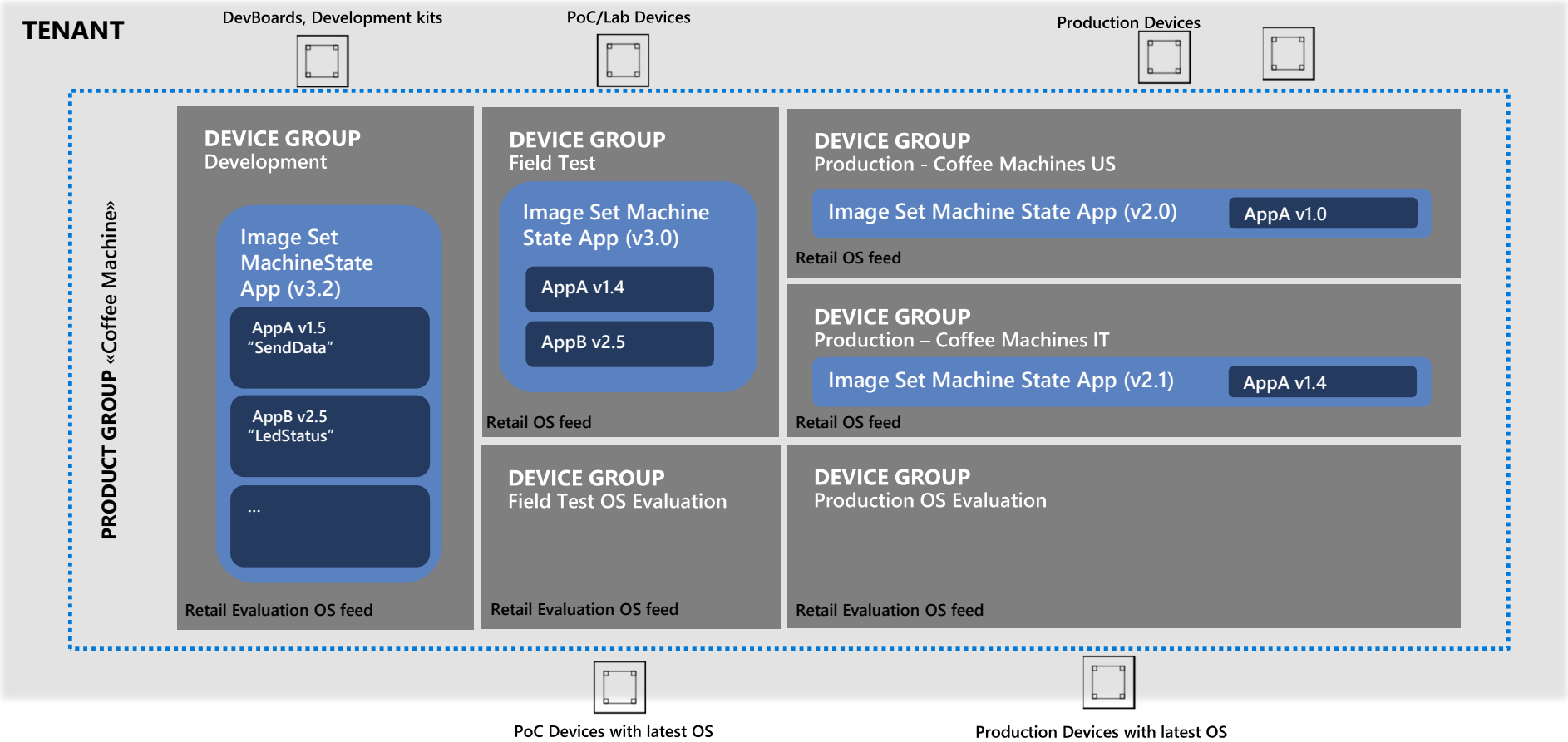
- **Chip SKU & system software**

- An operating system for a particular chip (i.e MT3620)

- **OS Feed**

- "Retail" – Contains officially released SW to be run on all retail devices
- "RetailEval" – A preview of what will soon be in the retail channel and is intended for validation processes

# Azure Sphere: Device Management & SW Lifecycle





# Azure Sphere

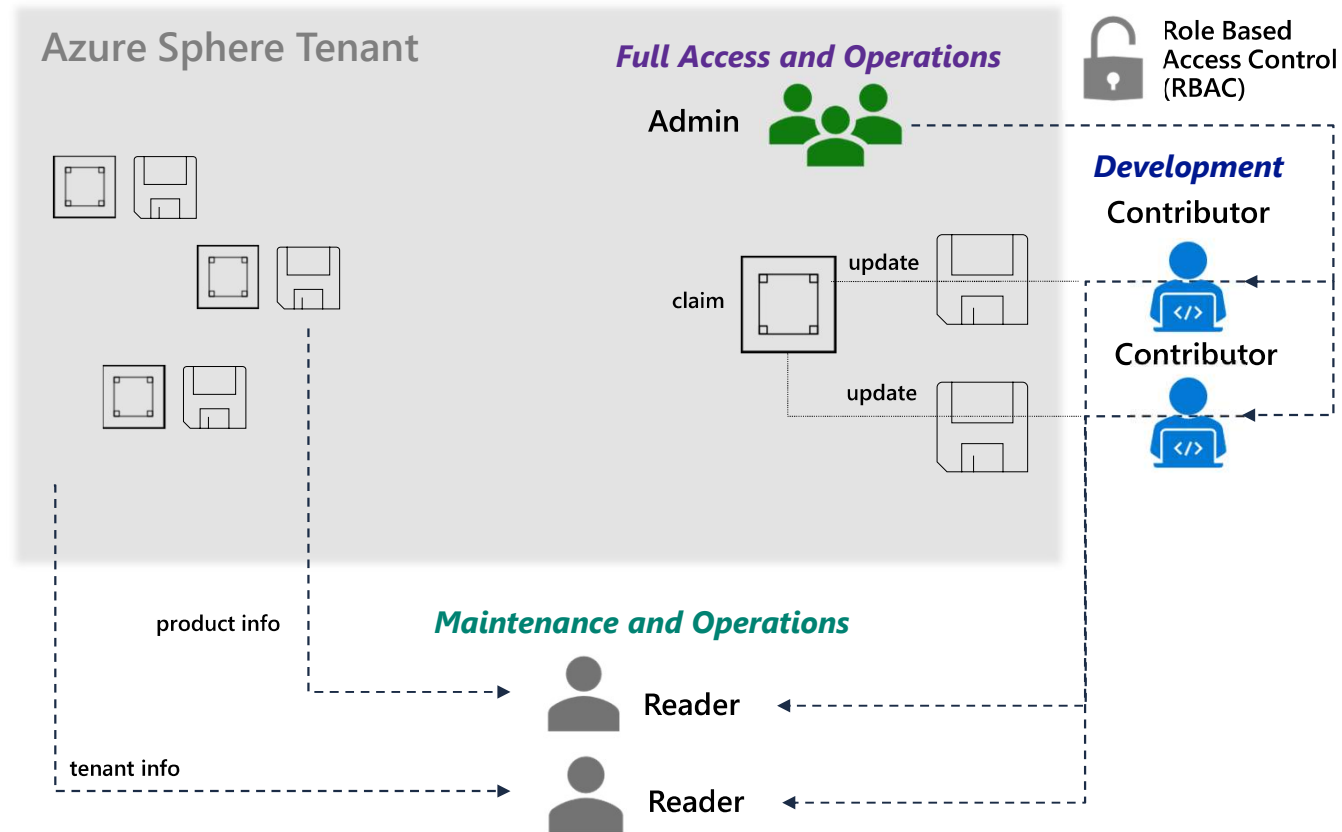
## Role Base Access Control (RBAC)

# Azure Sphere

## Role Base Access Control (RBAC)

### Type of Roles & Responsibilities

- **Administrator**
  - Has the full access to all devices and operations within the tenant, including permission to add or delete other users.
- **Contributor**
  - allow to add devices, create and change deployments.
  - Usually are SW / HW developers who create applications, manage connected devices, and update deployments, but not responsible for managing tenant access.
- **Reader**
  - has access to information about tenant, including claimed devices, deployments, and any error reporting data from devices.
  - This role is for maintenance and operations personnel who are responsible for tracking connected device performance at end-user installations.









# Azure Sphere: Re-cap

# Azure Sphere: a complete solution for creating highly secured devices.



	<p><b>Silicon IP blocks and architecture</b> to create secured &amp; economical MCUs for any device. Built on Pluton security subsystem. Provides tamper-resistant secured hardware root of trust from factory to field. Single-chip package for cost sensitive (MCU-based) devices.</p>	<p>For <b>Silicon Vendors</b> <i>Free IP License</i></p>
	<p><b>Operating System with ongoing servicing</b> to create a dependable root of trust. Multi-layer security architecture with <i>security subsystem</i> to ensure security and recover even under attack. Compartmentalization and defense in depth to ensure "no brick" update. Enhanced <u>Linux kernel</u> enables a robust and agile hardware/software ecosystem. <a href="https://github.com/cwtan7/azsphere-booticamp-master">https://github.com/cwtan7/azsphere-booticamp-master</a></p>	<p>For <b>OEMs</b> <i>Azure Sphere License</i></p>
	<p><b>Azure Sphere Security Service</b> assures ongoing security of Azure Sphere devices. Software update and security renewal of OS and applications. Early-warning detection of device failures (found in emerging security attacks). Certificate-based authentication for device-to-cloud and device-to-device communication.</p>	<p>For <b>OEMs</b> <i>Azure Sphere License</i></p>
	<p><b>Developer SDKs and Tools</b> to make it easy to create secured applications. Professional software development experience in Visual Studio. Application model enables secured reuse of proven software components in devices. Side-loading (field servicing) and over-the-air update of applications.</p>	<p>For <b>OEMs</b> <i>VS License</i></p>