ARCHITECTURAL CASE STUDY

Device Identity & Trust Architecture

A lifecycle for secure onboarding, ownership, credentials, access decisions, and auditability.

Onboard | **Identify** | **Authorize** | **Transfer** | **Retire**

Abdul Rehman

Solution Architect | Principal .NET Engineer

Architecture, engineering decisions, and delivery scope.

Beyond the first connection

Connected devices need a consistent basis for identity, ownership, and authorization throughout their lifetime.

Common architecture gaps

- Shared or hard-coded credentials make individual devices difficult to distinguish.
- Ownership and tenant assignment depend on manual records.
- Credential changes become difficult after deployment.
- Technicians retain broad permissions beyond a support window.
- Access decisions, risk signals, and audit records remain disconnected.
- Transfer and retirement are not treated as security events.

Architectural objective

Separate trust concerns from product applications, with a shared control plane that existing systems can adopt incrementally.

Identity	A durable device record independent of network location.
Ownership	An explicit tenant relationship with controlled transfer.
Authorization	A decision based on identity, context, and requested action.
Auditability	A history of security and lifecycle events.

02 / LOGICAL ARCHITECTURE

A shared trust control plane

Explicit responsibilities connect devices, people, and existing applications to a common trust model.

Device
Firmware / gateway

Technician
Installer / support

Customer app
Portal / mobile

Existing cloud
Product integration

API gateway | Device authentication | Tenant context | Rate limiting

Onboarding
Claims, enrollment, activation

Identity registry
Device records and lifecycle

Credentials
Issuance, expiry, revocation

Policy
Access and trust decisions

Ownership
Assignment and transfer

Risk
Signals and device posture

Audit
Security events and evidence

Integration
APIs, events, and adapters

Identity store | Certificate authority / KMS | Policy store | Event bus | Audit log

Conceptual target architecture. Boxes represent logical responsibilities; some lifecycle capabilities remain planned.

Trust continues after onboarding

The design carries device identity through operational changes, ownership transfer, and retirement.

01 Register
Create the device record and ownership claim.

02 Onboard
Verify enrollment or bootstrap evidence.

03 Issue identity
Establish unique identity and a scoped credential.

04 Authorize
Evaluate identity, tenant, user, and resource context.

05 Monitor
Observe health, credential age, and risk signals.

06 Rotate
Renew credentials before expiry or after a security event.

07 Transfer
End prior ownership and establish the new relationship.

08 Retire
Disable access, revoke credentials, and preserve history.

Scope: onboarding was prototyped. The complete lifecycle is designed; certificate rotation and risk scoring are planned extensions.

Identity, authorization, and risk

A valid credential is one input. Access also depends on current ownership, policy, and operational context.

Identity evidence

- Unique device identifier
- Certificate or signed credential
- Issuer, expiry, and revocation
- Device model and firmware metadata

Access context

- Tenant and ownership relationship
- Requested resource and operation
- Device lifecycle state
- Technician role and support window

Risk and history

- Credential age and failure signals
- Device health and posture
- Ownership or configuration changes
- Quarantine and audit history

ILLUSTRATIVE POLICY

A registered device may upload telemetry when its credential is valid, it belongs to the active tenant, its risk is within policy, and the requested scope is permitted.

Decision: allow, deny, or require additional verification.

Adopt trust controls incrementally

Integration follows device capability and the existing system landscape.

Direct device API

A capable device uses secure onboarding and token endpoints.

Trusted gateway

A gateway represents constrained devices and carries their identity.

Existing cloud adapter

A product cloud exchanges identity, ownership, and lifecycle events.

SDK or middleware

An application validates identity evidence and requests policy decisions.

Engineering considerations

- A shared trust service becomes a critical dependency: define availability and failure behavior.
- Keep device identity stable while credentials and ownership relationships change.
- Define what happens when policy context is missing, stale, or inconsistent.

Representative settings: connected buildings, industrial equipment, consumer devices, and operational fleets.

Architecture through engineering

My role: Solution Architect and Principal .NET Engineer.

Architecture defined

- Control-plane responsibilities and domain boundaries
- Device and ownership models
- Onboarding and credential lifecycle flows
- Tenant-aware authorization and integration patterns

Engineering implemented

- .NET services and REST APIs
- Clean architecture and domain-driven design
- Tenant-aware data access
- Audit and lifecycle event handling

Prototype and next steps

- Device onboarding workflows prototyped
- Ownership transfer and technician access designed
- Certificate rotation planned
- Risk scoring planned

Core technologies and concepts

.NET | ASP.NET Core | REST APIs | X.509 / PKI | Multi-tenancy | Policy-based authorization | PostgreSQL | Audit logging