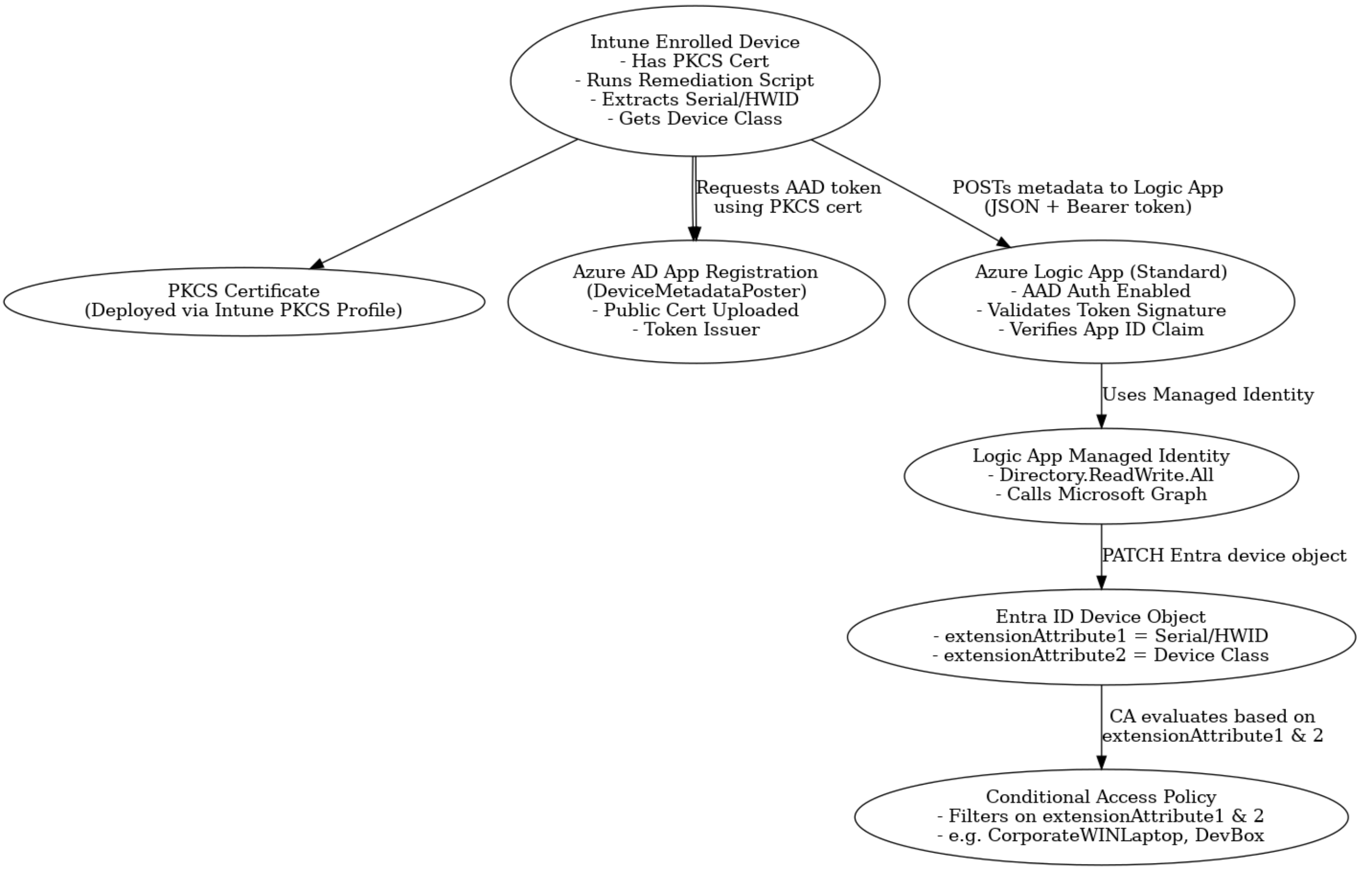
High-Level Design (HLD): Secure Device Metadata Ingestion Using PKCS Certificates and Logic App

Objective:

Objective:

Mitigate the OuttaTune vulnerability by injecting trusted, non-device-writable metadata into the Entra ID device object using a secure Azure Logic App. The metadata supports robust Conditional Access (CA) decisions and cannot be altered from the endpoint without significant privilege or hardware-based tampering.

Architecture Diagram:


Intune Enrolled Device
- Has PKCS Cert
- Runs Remediation Script
- Extracts Serial/HWID
- Gets Device Class
Requests AAD token using PKCS cert
POSTs metadata to Logic App (JSON + Bearer token)
PKCS Certificate (Deployed via Intune PKCS Profile)
Azure AD App Registration (DeviceMetadataPoster)
- Public Cert Uploaded
- Token Issuer
Azure Logic App (Standard)
- AAD Auth Enabled
- Validates Token Signature
- Verifies App ID Claim
Uses Managed Identity
Logic App Managed Identity
- Directory.ReadWrite.All
- Calls Microsoft Graph
PATCH Entra device object
Entra ID Device Object
- extensionAttribute1 = Serial/HWID
- extensionAttribute2 = Device Class
CA evaluates based on extensionAttribute1 & 2
Conditional Access Policy
- Filters on extensionAttribute1 & 2
- e.g. CorporateWINLaptop, DevBox

Flow Summary:

- 1. Intune Device Enrollment:

  - Device is enrolled using Microsoft Intune.

  - A PKCS certificate is issued via Intune's PKCS profile and installed in the device's LocalMachine\My store.

- 2. Remediation Script Execution:

  - Triggered by Intune at enrollment.

  - Collects hardware-bound identifier and device class.
  - Requests Azure AD access token using the PKCS certificate.
- 3. Secure HTTP POST to Logic App:
  - Sends metadata and token to Logic App via HTTPS.
- 4. Logic App Validation & Graph Injection:
  - Validates JWT signature and appid claim.
  - Uses Managed Identity to write to the device's Entra object.
- 5. Conditional Access Enforcement:
  - CA filters use extension attributes set via the Logic App.

Architecture Components:

- PKCS Certificate: Device credential for AAD authentication
- AAD App Registration: Allows only registered apps with cert to get tokens
- Intune Remediation Script: Collects and sends metadata securely
- Logic App (Standard): Validates token and writes metadata to Entra
- Managed Identity: Grants least-privilege Graph access
- Microsoft Graph API: Updates extension attributes
- Entra ID Device Object: Stores metadata not writable by device
- Conditional Access Policies: Evaluate trusted extension attributes

Defence-in-Depth: Risk & Mitigation

- On-device metadata spoofing: Hardware IDs are sourced from TPM, vTPM, UEFI, or BIOS. Spoofing requires hands-on time, specialized tools, or firmware tampering.
- Device-modified CA filters: Extension attributes are only modifiable by trusted cloud roles (e.g., Intune Admin, Global Admin).
- Token theft or reuse: PKCS cert is required for token issuance. Tokens are short-lived and app-specific.
- HTTPS interception (MITM): HTTPS is enforced. Even if intercepted, tokens are unusable without the cert's private key.
- Unauthorized Logic App access: AAD token is validated. Token signature and appid are checked.
- Overprivileged Logic App: Managed Identity is scoped to Directory.ReadWrite.All only.
- Perimeter trust assumptions: No perimeter assumptions. Works across home, roaming, and Autopilot scenarios.
- Replay / API abuse: Tokens are short-lived. Optional throttling and validation layers can be added.

Conditional Access Examples:

- device.extensionAttribute2 -eq "CorporateWINLaptop"
- device.extensionAttribute1 -in ["TPM1234", "UEFI-9876"]
- device.extensionAttribute2 -ne "CorporateWINLaptop"

Summary:

- Remote worker support: Yes
- Autopilot compatibility: Yes
- Tamper-resistant metadata: Yes
- No secrets in logs: Yes
- No on-device write path to CA inputs: Yes
- Token-based authentication via Azure AD: Yes
- Managed Identity for Microsoft Graph access: Yes