Comprehensive Architectural Guide to Configuring Network Device Enrollment Service (NDES) with Azure Active Directory Application Proxy for Intune SCEP Deployment

1. Executive Summary

In the contemporary landscape of enterprise mobility and unified endpoint management, the secure distribution of digital identities is a cornerstone of Zero Trust security architectures. As organizations migrate from traditional on-premises management to cloud-native frameworks spearheaded by Microsoft Intune, the requirement to bridge legacy Public Key Infrastructure (PKI) with modern cloud control planes becomes paramount. The Simple Certificate Enrollment Protocol (SCEP), facilitated by the Network Device Enrollment Service (NDES), serves as this critical bridge, enabling devices to request and obtain user or device certificates from an on-premises Active Directory Certificate Services (AD CS) environment.

Historically, exposing the NDES interface to the public internet—a requirement for managing mobile devices that operate outside the corporate perimeter—necessitated complex Demilitarized Zone (DMZ) configurations, inbound firewall ports, and the deployment of Web Application Proxy (WAP) servers. This traditional approach introduced significant attack surface risks and operational overhead. The advent of Azure Active Directory (Microsoft Entra ID) Application Proxy has revolutionized this architecture. By leveraging a secure, outbound-only reverse proxy mechanism, organizations can publish internal NDES endpoints to the internet without opening inbound ports, thereby significantly hardening the security posture of the critical PKI infrastructure.

This comprehensive research report provides an exhaustive, step-by-step technical analysis of designing, deploying, and maintaining an Intune SCEP environment using NDES behind Azure AD Application Proxy. It delves into the intricate details of server role configuration, Internet Information Services (IIS) hardening, connector deployment, and the nuanced troubleshooting methodologies required to maintain a high-availability identity issuance service. The report is structured to guide enterprise architects and senior systems engineers through every phase of the lifecycle, from initial prerequisite planning to advanced operational diagnostics.

2. Architectural Framework and Zero Trust Principles

2.1 The Role of SCEP in Modern Device Management

The Simple Certificate Enrollment Protocol (SCEP) was originally conceived by Cisco Systems to facilitate the scalable distribution of certificates to network devices such as routers and switches. In the context of Microsoft Intune and modern endpoint management, SCEP has been adapted to provision identity certificates to a diverse fleet of managed devices, including iOS, Android, Windows, and macOS endpoints. These certificates serve as the bedrock for various authentication scenarios, including certificate-based authentication (CBA) for Wi-Fi (802.1x), Virtual Private Networks (VPNs), and mutual TLS (mTLS) authenticated web services.

The operational workflow represents a hybrid model where the "source of truth" for identity issuance remains the on-premises PKI—controlled by the organization—while the "authority to request" is delegated to the cloud-based Mobile Device Management (MDM) platform. This separation of duties is enforced through a rigorous challenge-response mechanism. When Intune generates a SCEP profile for a targeted device, it creates a unique, time-bound, and randomized challenge string. The device, upon receiving this policy, generates a key pair and presents the public key along with the challenge string to the NDES server. The NDES server, augmented with a specialized policy module (the Microsoft Intune Certificate Connector), validates this challenge against the Intune cloud service before instructing the underlying Certification Authority (CA) to issue the certificate. This ensures that only managed, compliant devices authorized by Intune can obtain a valid certificate from the corporate CA.

2.2 Architectural Evolution: From Web Application Proxy (WAP) to Azure AD Application Proxy

For years, the standard recommendation for publishing NDES to the internet involved the use of Microsoft Web Application Proxy (WAP) or third-party reverse proxies (e.g., F5 Big-IP, Citrix ADC) located in the perimeter network (DMZ). While functionally valid, the WAP architecture imposed specific security and operational burdens:

- **Inbound Attack Surface:** WAP required allowing inbound HTTPS traffic (TCP port 443) from the entire internet to the DMZ, and subsequently from the DMZ to the internal NDES server. This exposure increased the risk of Distributed Denial of Service (DDoS) attacks and exploitation of vulnerabilities in the listening service.[1]
- **Infrastructure Complexity:** Maintaining a WAP infrastructure required dedicated servers, load balancers, and complex firewall rule sets, adding to the total cost of ownership.
- **DMZ Management:** The servers in the DMZ required patching, monitoring, and management, often creating friction between security and infrastructure teams.

The paradigm shift toward Azure AD Application Proxy fundamentally alters this traffic flow, aligning it with Zero Trust principles. The Azure AD Application Proxy utilizes a lightweight agent—the Application Proxy Connector—installed inside the corporate network. This

connector establishes a persistent, outbound-only connection to the Azure cloud service.

Architectural Advantages of Azure AD Application Proxy:

1. **Zero Inbound Ports:** The solution requires no inbound ports to be opened on the perimeter firewall. All communication is initiated outbound from the connector to the Azure service on standard HTTPS ports.[2]
2. **Cloud-Scale Protection:** Microsoft Entra ID acts as the front door, absorbing the initial connection attempts. This provides inherent protection against volumetric DDoS attacks, as malicious traffic is filtered at the Microsoft edge before it ever reaches the on-premises infrastructure.
3. **Simplified Topology:** The removal of the DMZ requirement for this specific service simplifies the network topology. The connector can reside on the NDES server itself or on adjacent member servers, collapsing the infrastructure footprint.
4. **Global Availability:** The Azure Application Proxy service is globally distributed, ensuring that device requests are routed to the nearest operational entry point, improving latency and reliability for a geographically dispersed mobile workforce.

2.3 Traffic Flow Analysis and Mechanism of Action

Understanding the precise flow of traffic is essential for configuration and troubleshooting. In the Azure AD Application Proxy scenario, the flow operates as follows:

1. **Policy Delivery:** The device receives a SCEP profile from Microsoft Intune. This profile contains the external URL of the SCEP service and the dynamic challenge string.
2. **DNS Resolution:** The device attempts to contact the SCEP service. It performs a DNS lookup for the external URL (e.g., scep.contoso.com or tenant.msappproxy.net). This resolves to a public IP address owned by the Microsoft Azure Application Proxy service.
3. **Request Initiation:** The device sends an HTTPS GET request to the Azure endpoint. The URL typically contains the operation GetCACert or PKIOperation.
4. **Proxy Traversal:** The Azure service identifies the tenant and application associated with the request. It determines that the application is configured for "Passthrough" authentication (a critical requirement discussed later).
5. **Tunnel Routing:** Azure routes the request down the established secure tunnel to the on-premises Application Proxy Connector.
6. **Internal Forwarding:** The connector receives the request and forwards it to the internal NDES server URL (e.g., https://ndes.corp.contoso.com/certsrv/mscep/mscep.dll).
7. **Processing and Validation:** The NDES server's IIS process receives the request. The Intune Certificate Connector (Policy Module) validates the challenge with the Intune service.
8. **Issuance:** Upon validation, NDES requests the certificate from the CA.
9. **Response:** The certificate is returned to NDES, then to the Connector, up the tunnel to Azure, and finally to the device.

3. Infrastructure Prerequisites and Planning

Before any software installation occurs, the environment must be rigorously prepared. The interdependencies between Active Directory, the Certificate Authority, and the NDES server are complex, and deviations from supported configurations can lead to instability.

3.1 Active Directory Certificate Services (AD CS) Design

The backend Public Key Infrastructure must be robust. A "Standalone" CA is strictly unsupported for this integration; the environment requires an **Enterprise Certification Authority** running on Windows Server.

- **Operating System Requirements:** The CA must run on an Enterprise edition of Windows Server. While Windows Server 2008 R2 SP1 is the absolute minimum floor (requiring hotfix KB2483564), modern deployments should utilize Windows Server 2016, 2019, or 2022 to ensure long-term supportability and security compliance.[3]
- **CA Hierarchy:** The NDES server interacts with an Issuing CA. This CA must be online and accessible. If the organization utilizes an offline Root CA (a best practice), the Issuing CA must have a valid certificate chain and an accessible Certificate Revocation List (CRL).
- **Template Support:** The integration relies heavily on Certificate Templates published in Active Directory. The Enterprise CA must have permissions to read these templates and issue certificates based on them.

3.2 NDES Server Specifications and Placement

The Network Device Enrollment Service role requires careful placement. It acts as a Registration Authority (RA), a sensitive role that can request certificates on behalf of others.

- **Dedicated Infrastructure:** It is a critical security requirement **not** to install the NDES role on the same server that hosts the Enterprise Certification Authority. Co-locating these roles creates a single point of failure and exposes the CA directly to web-based traffic and potential compromise, which is considered an unsupported configuration by Microsoft.[3]
- **Domain Membership:** The NDES server must be a domain-joined member server. It must reside in the same forest as the Enterprise CA to facilitate Kerberos authentication and RPC communication.
- **Operating System:** Windows Server 2012 R2 is the minimum, but Windows Server 2019 or 2022 is strongly recommended for better TLS support and IIS management features.
- **Virtualization:** NDES is a lightweight service primarily bound by network I/O and CPU for SSL termination. It is an ideal candidate for virtualization.
- **Hardening:** As a web-facing server (even via proxy), it should be hardened. However, care must be taken not to disable protocols required for internal communication with the CA.

3.3 Network Topology and Firewall Considerations

Implementing Azure AD Application Proxy alters the firewall requirements significantly compared to legacy WAP deployments.

Outbound Traffic (Connector Server):
The server hosting the Azure AD Application Proxy Connector (which may be the NDES server or a separate server) requires outbound access to the Microsoft Cloud.

- **Ports:** TCP 80 and TCP 443.
- **Destinations:** *.msappproxy.net, *.servicebus.windows.net, and various login endpoints (login.microsoftonline.com).
- **Inspection:** If the environment uses deep packet inspection (DPI) or SSL bridging on the outbound firewall, exemptions may be required. The connector utilizes long-lived WebSocket connections and mutual authentication that can be disrupted by aggressive proxy inspection.

Internal Traffic (NDES Server to Infrastructure):
The NDES server must communicate with internal infrastructure components.

- **To Certification Authority:** RPC (TCP 135) and dynamic high ports (49152-65535) for DCOM communication. This allows the NDES service to request certificates.
- **To Domain Controllers:** LDAP (TCP/UDP 389), LDAP SSL (TCP 636), Kerberos (TCP/UDP 88), and DNS (TCP/UDP 53).
- **To CRL Distribution Points:** HTTP (TCP 80) access to wherever the CA publishes its revocation lists. Failure to check revocation status is a common cause of NDES startup failures.[3]

3.4 Service Account Strategy and Least Privilege

The architecture necessitates the use of several distinct identities. Using a single "super account" for all functions is a security violation and hinders auditing.

Account Role	Type	Context	Permissions Required
NDES Service Account	Domain User	Runs the IIS App Pool	• Member of local IIS_IUSRS on NDES server. • Log on as a batch job rights.

			• **Read** and **Enroll** permissions on the SCEP Certificate Template. • **Request Certificates** permission on the CA.
Intune Connector Account	System or Domain User	Runs the Connector Service	• Local Administrator on NDES server (for installation). • Log on as a service rights. • If handling revocation: **Issue and Manage Certificates** on the CA.[3]
Connector Installation Account	User (Temporary)	Installation Context	• Global Admin or Application Admin in Entra ID. • Local Admin on the server.
Enterprise Admin	User (Temporary)	NDES Config Wizard	• Required only once to run the AD CS Configuration Wizard to install the RA certificates.

Important Note on Service Principal Names (SPN):
If the NDES application pool is running as a domain user, it is best practice to register a Service Principal Name (SPN) for the HTTP service, even if Kerberos is not the primary authentication method for the external SCEP request. This ensures that internal management tools or future authentication changes operate correctly.

- Command: setspn -s http/ndes.fqdn domain\serviceAccount.[3]

4. Implementation: Active Directory Certificate Services

The configuration of the PKI itself is the foundation of the SCEP process. The NDES server does not decide *what* kind of certificate to issue; it relies on a Certificate Template configured on the CA.

4.1 Certificate Template Configuration

Administrators must create a specific certificate template that NDES will use. This is typically done by duplicating an existing template, such as "User".

Step-by-Step Template Configuration:

1. Open the **Certificate Templates Console** (certtmpl.msc) on the CA.
2. Right-click the **User** template and select **Duplicate Template**.
3. **General Tab:**
    - **Template Display Name:** Give it a clear name, e.g., "Intune SCEP User".
    - **Validity Period:** Define the lifespan of the certificates (e.g., 1 year). This must not exceed the validity period of the Root CA certificate.
    - **Publish certificate in Active Directory:** Ensure this is checked.
4. **Request Handling Tab:**
    - **Purpose:** Signature and encryption.
    - **Allow private key to be exported:** Generally unchecked for mobile devices to ensure non-repudiation, though some specific S/MIME scenarios might require it.
5. **Subject Name Tab:**
    - **Supply in the request:** This is the most critical setting. Select this radio button.
    - *Reasoning:* Intune constructs the Subject Name (SN) and Subject Alternative Name (SAN) dynamically based on the enrolled user's attributes (e.g., UPN, Email). NDES passes this information in the request. If the template is set to "Build from this Active Directory information," the certificate would be issued with the identity of the NDES service account, not the end user.
6. **Extensions Tab:**
    - **Application Policies:** Ensure **Client Authentication** is present. If the certificate is also used for iOS Wi-Fi, ensure the proper EKUs are listed.
    - **Key Usage:** Digital Signature, Key Encipherment.
7. **Security Tab:**
    - Add the **NDES Service Account** (created in Section 3.4).
    - Grant **Read** and **Enroll** permissions.
    - *Note:* The actual end-users do *not* need permissions on this template, because NDES requests the certificate on their behalf.

4.2 Issuance Policy and Security Descriptors

Once the template is created, it must be added to the CA's issuance list.

1. Open the **Certification Authority** snap-in (certsrv.msc).
2. Right-click **Certificate Templates** > **New** > **Certificate Template to Issue**.
3. Select the newly created "Intune SCEP User" template.

5. Implementation: Network Device Enrollment Service (NDES)

The installation and configuration of the NDES role is a multi-stage process involving Windows role installation, post-deployment configuration, and significant IIS tuning.

5.1 Role Installation and Dependencies

On the designated NDES server, the role is installed via Server Manager or PowerShell.

Dependencies:
The NDES role depends on Internet Information Services (IIS). The installation wizard will automatically select IIS, but administrators must verify that specific sub-components are included. The Intune Connector requires specific.NET framework support.

- **Web Server (IIS) > Application Development:**
  - **ASP.NET 3.5** (requires.NET Framework 3.5 features installed).
  - **ASP.NET 4.7.2** (or the latest version available on the OS).
  - **ISAPI Extensions** & **ISAPI Filters**: Essential for the mscep.dll operation.
- **Management Tools:**
  - **IIS 6 Metabase Compatibility**: This legacy component is often required for the connector's installation scripts to correctly interact with the IIS configuration.[3]

5.2 Service Configuration and RA Certificates

After the binary files are installed, the service must be configured to integrate with the CA.

1. Log in as an **Enterprise Administrator**.
2. Launch the **AD CS Configuration** wizard from Server Manager.
3. Select the **Network Device Enrollment Service** role service.
4. **Service Account:** Specify the **NDES Service Account** (domain user).
5. **RA Information:** Input organization details.
6. **Cryptography:** Accept the default provider and key length (typically 2048-bit).
7. **CA Selection:** Browse and select the Enterprise CA configured in Section 4.

The "RA Certificates" Concept:
During this wizard, the NDES server requests two special certificates from the CA: the Exchange Enrollment Agent certificate and the CEP Encryption certificate.

- *Enrollment Agent:* Used by NDES to sign the request sent to the CA.
- CEP Encryption: Used to encrypt the challenge password and other sensitive data during the initial handshake.
  These certificates are stored in the local computer's personal store on the NDES server. If these certificates expire or are revoked, NDES will cease to function immediately.

5.3 IIS Hardening and Optimization (Request Filtering)

The default IIS configuration is insufficient for Intune SCEP requests. SCEP requests initiated by Intune often contain extremely long query strings and URL paths because they include the full certificate chain and a signed data blob.

The "Long URL" Problem:
Standard IIS settings limit the maximum URL length to approximately 4KB. Intune SCEP requests can easily exceed this, sometimes reaching 20KB or more depending on the key size and chain depth. If IIS is not tuned, the NDES server will reject the request with an HTTP 404.14 or 404.15 error, which is often masked as a generic "500 Internal Server Error" or "Bad Gateway" by the Application Proxy.5
Configuration Steps:

1. Open **IIS Manager**.
2. Select the **Default Web Site** (or the site hosting NDES).
3. Double-click **Request Filtering** in the central pane.
4. In the Actions pane (right side), click **Edit Feature Settings**.
5. Adjust the following values:
   - **Maximum URL length (Bytes):** Set to **65534**.
   - **Maximum query string (Bytes):** Set to **65534**.
   - **Allow double escaping:** Enable this if necessary (though usually not required for standard SCEP, some documentation suggests it for specific client edge cases).[4]
6. Click OK and restart IIS (iisreset).

Verification:
These settings are written to the web.config file in the root of the website or the applicationHost.config. Administrators should verify that the system.webServer/security/requestFiltering section reflects these changes.
Application Pool Identity:
Verify that the SCEP application pool in IIS is running under the identity of the NDES Service Account.

- Navigate to **Application Pools**.
- Right-click the **SCEP** pool > **Advanced Settings**.
- Ensure **Identity** matches the configured service account.
- Ensure **Load User Profile** is set to **True** (this helps with accessing the user's certificate store).

5.4 Registry Configuration for Template Mapping

Unlike the UI-driven setup for many Microsoft services, the mapping between NDES and the specific Certificate Template created in Section 4 is done via the Windows Registry.

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MSCEP

Keys to Modify:
There are three specific registry values that determine which template NDES requests from the CA based on the "purpose" flag in the incoming SCEP packet.

1. **EncryptionTemplate**
2. **GeneralPurposeTemplate**
3. **SignatureTemplate**

Action:
Edit each of these string values and replace the default "IPSECIntermediateOffline" (or similar) with the Template Name (the distinct object name, not the display name) of the Intune SCEP template created earlier (e.g., "IntuneSCEPUser").

- *Insight:* In most Intune deployments, the same template is used for all three purposes. This simplifies configuration. Failing to update these keys will result in NDES attempting to request a default template that likely doesn't exist or for which it has no permissions, causing immediate failure.[3]

Restart Requirement:
After modifying these registry keys, the IIS service must be restarted for NDES to reload the configuration.

6. Implementation: Microsoft Intune Certificate Connector

The Microsoft Intune Certificate Connector is the logic glue that binds the generic NDES service to the specific requirements of the Intune cloud. Technically, it functions as a "Policy Module" for NDES.

6.1 Connector Architecture and Policy Modules

In a standard SCEP implementation, the "challenge password" is a static shared secret. This is insecure for massive device deployments. Intune improves this by generating a *dynamic* challenge for every single request.

When NDES receives a request, it passes the challenge to its configured Policy Module.

- **Standard Windows Module:** Checks against a static local value.
- **Intune Module:** Intercepts the challenge, establishes an outbound HTTPS connection to the Intune Service, and asks: "Is this challenge valid, and is it associated with this specific

device ID?"
- If Intune returns "Yes," the module tells NDES to proceed. If "No," it instructs NDES to reject the request.

6.2 Installation and Registration

Prerequisite Check:
Before installing, ensure Internet Explorer Enhanced Security Configuration (IE ESC) is disabled on the NDES server for both Administrators and Users. The connector sign-in process utilizes modern authentication libraries that can be blocked by IE ESC, leading to registration failures (blank login screens or script errors).3
Installation Steps:

1. Log in to the **Microsoft Intune admin center**.
2. Navigate to **Tenant administration** > **Connectors and tokens** > **Certificate connectors**.
3. Download the **Certificate Connector for Microsoft Intune**.
4. Run the installer on the NDES server with **Administrator** privileges.
5. **Features Selection:** Select **SCEP**. If this connector will also be responsible for revocation (importing CRL information to Intune), select that option as well.
6. **Service Account:** Select a user account to run the connector service. This can be the **SYSTEM** account or a domain user. The account needs permission to write to the event log and communicate with Intune.
7. **Proxy Configuration:** If the server sits behind a corporate web proxy, provide the details here. The connector supports unauthenticated and authenticated proxies.
8. **NDES Detection:** The installer will attempt to detect the NDES role. If the IIS prerequisites (ASP.NET) are missing, it will halt here.
9. **Registration:** The installer will prompt for Azure AD credentials. Sign in with a **Global Administrator** or **Intune Administrator** account with a valid Intune license. This registers the connector instance with your tenant.

6.3 Post-Installation Validation

Once installed, verify the health of the connector:

- **Services Console:** Ensure the **Microsoft Intune Certificate Connector** service is running.
- **Intune Portal:** Check the connector status in the Intune admin center. It should report "Active".
- **Registry Check:** Verify that the policy module registry key at HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MSCEP\Modules has been updated to point to the Intune connector DLLs rather than the default Windows module.

7. Implementation: Azure AD Application Proxy

This section details the publication of the internal NDES service to the internet. This is the pivotal step that replaces the legacy Web Application Proxy.

7.1 Connector Group Strategy and High Availability

The **Azure AD Application Proxy Connector** is the agent that facilitates the outbound tunnel.

Placement Strategy:

- **Co-location:** It is fully supported to install the App Proxy Connector directly on the NDES server. This is the simplest configuration and reduces network hops.[2]
- **Dedicated Server:** For larger environments, connectors are often placed on dedicated "connector servers" to separate duties.

High Availability:
A single connector is a single point of failure. It is best practice to install connectors on at least two servers (e.g., two NDES servers, each with a connector).

- **Connector Groups:** In the Azure portal, assign these connectors to a specific **Connector Group** (e.g., "SCEP-Connectors"). This allows you to isolate traffic for SCEP from other published applications and ensures that if one server goes offline, the Azure service automatically routes traffic to the remaining active connectors in the group.[8]

7.2 Enterprise Application Configuration

With the connectors active, you must configure the "Enterprise Application" in Microsoft Entra ID.

Creation Steps:

1. Navigate to **Microsoft Entra admin center** > **Enterprise applications** > **Application proxy**.
2. Click **+ New application** > **Add an on-premises application**.

Core Settings:

- **Name:** Enter a descriptive name, e.g., "Intune NDES SCEP".
- **Internal URL:** This is the URL the connector uses to talk to NDES. It must match the internal FQDN of the NDES server, e.g., https://ndes.corp.contoso.com/.
    - *Requirement:* The NDES server must have a valid internal SSL certificate bound to port 443 in IIS that matches this FQDN. The App Proxy connector will validate this certificate. If the internal cert is self-signed or untrusted, the connector will refuse the connection.[2]
- **External URL:** Select the public-facing URL.
    - *Option A (Default):* https://ndes-tenantname.msappproxy.net/
    - *Option B (Custom):* https://scep.contoso.com/ (See Section 7.3).

- **Pre Authentication: Passthrough**.
  - *Critical Architecture Decision:* You **must** select Passthrough. You cannot use "Azure Active Directory" pre-authentication. The SCEP protocol (as implemented by native OS clients on iOS/Android) does not support interactive logins (MFA, redirects, OAuth flows) during the certificate request process. The device simply expects to post a binary blob to an endpoint. Securing the endpoint is done via the SCEP challenge validation, not the identity of the connection.[2]
- **Connector Group:** Select the group created in 7.1.

Additional Settings:

- **Backend Application Timeout:** Set to **Default**. Only increase to **Long** if you experience specific timeouts, but be aware that long timeouts can mask underlying performance issues with the CA.[9]
- **Translate URLs in Headers: Yes**.
- **Translate URLs in Application Body: No**.
  - *Reasoning:* SCEP request/response bodies are binary and often encrypted. Attempting to parse and translate URLs within the body can corrupt the payload, causing verification failures on the device.[7]

7.3 DNS Strategy: Custom Domains vs. Default Suffixes

Organizations have a choice regarding the external URL.

Default Suffix (msappproxy.net):

- *Pros:* Zero configuration. Instant availability.
- *Cons:* Long, unsightly URLs. If you change your tenant name, the URL might change, breaking existing SCEP profiles.
- *Tenant Name Caveat:* If you have renamed your tenant, you might encounter issues where the App Proxy expects the *old* tenant name in the URL. This is a known quirk in the App Proxy service logic regarding fallback domains.[10]

Custom Domain (scep.contoso.com):

- *Pros:* Professional, persistent URL. Decouples the service from the tenant name.
- *Configuration:*
  1. Upload a wildcard certificate (*.contoso.com) or specific certificate (scep.contoso.com) PFX to the Application Proxy configuration.
  2. Create a **CNAME** record in public DNS: scep.contoso.com -> ndes-tenantname.msappproxy.net.
- *Recommendation:* Always use a custom domain if possible. It provides flexibility to migrate the backend (e.g., to a different tenant or a different proxy technology) without reissuing SCEP profiles to thousands of devices.[11]

7.4 Handling Pre-Authentication and Protocol Constraints

It is vital to reiterate why **Passthrough** is the only supported mode. When an administrator selects "Azure Active Directory" pre-authentication, the App Proxy expects the client to present an authentication token or to interactively sign in via a web interface.

- The **iOS Keychain** and **Android Keystore** processes that perform SCEP enrollment run at a system level. They do not have a user interface to display a Microsoft login prompt.
- They do not have access to the user's Azure AD OAuth tokens.
- Therefore, any setting other than Passthrough results in the device receiving a login page (HTML) instead of a SCEP response (binary), causing the enrollment to fail with a generic error.
  Security is not compromised because the NDES server will only issue a certificate if the device presents a challenge blob that Intune has explicitly generated for it. A random attacker hitting the URL cannot get a certificate because they lack a valid challenge.2

8. Intune Policy Configuration

With the infrastructure pipeline established, the final configuration occurs within the Intune management plane.

8.1 Establishing Trust: Trusted Certificate Profiles

Before a device can trust the certificate it receives from NDES, it must trust the chain of authority that signed it. If a device receives a certificate signed by "Contoso Issuing CA," but it does not have that CA's public key in its Trust Store, it will reject the certificate.

Steps:

1. **Export Certificates:** Export the public keys (.cer) of your Root CA and your Issuing CA (and any intermediates). Do not export private keys.
2. **Create Profiles:**
   - In Intune, go to **Devices** > **Configuration profiles** > **Create profile**.
   - Select the platform (e.g., iOS/iPadOS) and type **Trusted certificate**.
   - Upload the Root CA .cer.
   - Repeat for the Issuing CA/Intermediate CA.
3. **Assignment:** Assign these profiles to the *same* user/device groups that will receive the SCEP profile. The trust must land on the device before or simultaneously with the SCEP request.[3]

8.2 Defining the Request: SCEP Certificate Profiles

The SCEP profile tells the device *how* to ask for a certificate.

Key Configuration Parameters:

- **Certificate type:** User or Device.
- **Subject name format:** This is where you leverage Intune variables. E.g., CN={{UserName}}, E={{EmailAddress}}.
    - *Insight:* This allows a single profile to serve thousands of users, with each certificate uniquely identifying the holder.
- **Key usage:** Digital Signature, Key Encipherment.
- **Key size:** 2048 (standard) or 4096 (high security).
- **Root Certificate:** Select the *Trusted Certificate* profile created in 8.1 that corresponds to the Root CA.

SCEP Server URLs:
This is the most common point of misconfiguration.

- You must enter the **External URL** configured in the Azure AD App Proxy.
- **Crucial Formatting:** You cannot just enter the domain. You must append the full path to the SCEP endpoint.
- **Format:** https://<External-URL>/certsrv/mscep/mscep.dll
    - *Example:* https://scep.contoso.com/certsrv/mscep/mscep.dll
    - *Example:* https://ndes-tenant.msappproxy.net/certsrv/mscep/mscep.dll
- Many administrators forget the .dll extension or the /certsrv/mscep/ path, leading to 404 errors on the device.[13]

9. Operational Maintenance and Security Hygiene

A "set and forget" mentality leads to PKI outages. The NDES environment requires active lifecycle management.

9.1 Certificate Rotation and Lifecycle Management

The NDES server relies on several certificates, each with different expiration timelines:

Certificate	Purpose	Expiration Risk	Remediation
Server SSL	IIS Binding	Internal communication fails. App Proxy connector returns "Bad Gateway" due to trust error.	Renew via internal CA. Update IIS Bindings. No restart usually needed.
Enrollment Agent	Signing SCEP Requests	NDES stops issuing certificates. Logs show "Signature"	Renew in the User Certificate store of the NDES Service

		errors.	Account. NDES usually auto-picks the new one, but restart IIS to be sure.
CEP Encryption	Decrypting Challenges	NDES fails to decrypt incoming requests.	Renew in the User Certificate store of the NDES Service Account.
App Proxy Cert	External SSL	Devices cannot connect. SSL errors on client.	If using a custom domain, upload the new PFX to the Azure portal before expiration.

Monitoring Strategy:
Implement monitoring on the NDES service account's personal certificate store ("My" store). Alert on expiration < 30 days.

9.2 Patch Management and Connector Updates

- **OS Patching:** NDES servers should be patched regularly. However, treat them as critical infrastructure. Rebooting NDES during a mass-enrollment event (e.g., start of a school year) will cause enrollment failures.
- **Connector Updates:** The Intune Certificate Connector is updated frequently by Microsoft to support new features or fix bugs. While it supports auto-update, administrators should periodically verify the version in the Intune console to ensure the auto-update mechanism is functioning.

10. Advanced Troubleshooting and Diagnostics

When SCEP fails, it can be difficult to pinpoint the break in the chain. Is it the device? The network? Azure? NDES? The CA?

10.1 Interpreting HTTP Status Codes (The 403 Paradox)

A common confusion arises during testing. Administrators often try to validate the URL by pasting https://scep.contoso.com/certsrv/mscep/mscep.dll into a web browser.

- **Result:** They receive an **HTTP 403 Forbidden** error.
- **Interpretation:** This is **GOOD**. It indicates success.[15]

    - *Why?* The mscep.dll ISAPI extension expects a specific SCEP payload (GET parameters or POST body). When accessed via a browser with no payload, the server correctly denies access.
    - *Diagnosis:* If you see the IIS 403 page, it proves:
        1. DNS is correct.
        2. Azure App Proxy is working.
        3. The Connector is working.
        4. Authentication (Passthrough) is working.
        5. IIS is running.
- **True Failure:** If you see a generic browser "Can't reach this page" or an "HTTP 504 Gateway Timeout," that indicates a connectivity break.

10.2 Analyzing Event Logs and Tracing

On the Device (Windows):

- **Event Viewer:** Applications and Services Logs > Microsoft > Windows > DeviceManagement-Enterprise-Diagnostics-Provider > Admin.
- **Success:** Event ID **36** ("SCEP: Certificate request generated successfully").
- **Failure:** Look for errors indicating URL reachability or "Bad Request".

On the NDES Server:

- **Application Log:** Look for events from the **NetworkDeviceEnrollmentService** source.
- **IIS Logs:** located in C:\inetpub\logs\LogFiles. Check for the incoming requests.
    - *Tip:* If you see the request in IIS logs with a 200 OK, the network path is fine. If you don't see the request, it's blocked at the App Proxy or Firewall.
- **Intune Connector Logs:** Located in C:\Program Files\Microsoft Intune\NDESConnectorSVC\Logs. These logs are verbose and show the communication with the Intune cloud. Look here for "Challenge Validation Failed" errors.

On the App Proxy Connector Server:

- **Event Viewer:** Applications and Services Logs > Microsoft > Microsoft Entra private network > Connector > Admin.
- **Common Errors:**
    - *Backend unreachable:* Check the internal URL and SSL trust.
    - *Kerberos errors:* Check SPNs (though less relevant for Passthrough).

10.3 Common Configuration Pitfalls

1. **Mismatched Template Names:** The registry keys on the NDES server must match the *Schema Name* of the template, not the Display Name. These often differ.
2. **Service Account Permissions:** The NDES service account is frequently forgotten when assigning permissions to the Certificate Template or the CA itself.
3. **App Proxy Pre-Auth:** Accidentally leaving Pre-Authentication set to "Azure Active

Directory" is the #1 cause of SCEP failure in this architecture.

4. **Date/Time Skew:** Ensure the NDES server, Connector server, and Azure services are time-synced. Kerberos and TLS are time-sensitive.

Conclusion

The integration of on-premises NDES with Azure AD Application Proxy represents the mature, secure standard for hybrid identity issuance. By adhering to the detailed architectural guidelines presented in this report—specifically the rigorous hardening of IIS, the precise configuration of the Intune Connector policy module, and the correct deployment of the Azure App Proxy in Passthrough mode—organizations can achieve a seamless, invisible certificate provisioning experience. This architecture not only meets the functional requirement of delivering certificates to internet-facing devices but does so while strictly adhering to modern security principles, eliminating the need for perilous inbound firewall configurations and reducing the enterprise attack surface.

Works cited

1. Integrate NDES, SCEP with Web Application Proxy - Microsoft Q&A, accessed December 12, 2025, https://learn.microsoft.com/en-us/answers/questions/1185094/integrate-ndes-scep-with-web-application-proxy
2. Use Microsoft Entra application proxy with a Network Device ..., accessed December 12, 2025, https://learn.microsoft.com/en-us/entra/identity/app-proxy/app-proxy-protect-ndes
3. Configure infrastructure to support SCEP certificate profiles with ..., accessed December 12, 2025, https://learn.microsoft.com/en-us/intune/intune-service/protect/certificates-scep-configure
4. SSO to domain resources from Azure AD Joined Devices - The MEGA Series - Part 5 - Configure the Network Device Enrollment Service (NDES) - MSEndpointMgr, accessed December 12, 2025, https://msendpointmgr.com/2022/02/23/sso-to-domain-resources-from-azure-ad-joined-devices-the-mega-series-part-5-configure-the-network-device-enrollment-service-ndes/
5. Troubleshooting device to NDES server communication for SCEP certificate profiles in Microsoft Intune, accessed December 12, 2025, https://learn.microsoft.com/en-us/troubleshoot/mem/intune/certificates/troubleshoot-scep-certificate-device-to-ndes
6. Support Tip - How to configure NDES for SCEP certificate deployments in Intune, accessed December 12, 2025, https://techcommunity.microsoft.com/blog/intunecustomersuccess/support-tip---how-to-configure-ndes-for-scep-certificate-deployments-in-intune/455125
7. Use Certificates to enable SSO for Microsoft Entra join devices, accessed

December 12, 2025, https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-hybrid-aadj-sso-cert
8. Windows Hello For Business | PDF - Scribd, accessed December 12, 2025, https://www.scribd.com/document/500229282/Windows-Hello-for-Business
9. Add an on-premises application for remote access through application proxy in Microsoft Entra ID., accessed December 12, 2025, https://learn.microsoft.com/en-us/entra/identity/app-proxy/application-proxy-add-on-premises-application
10. Cannot deploy On Premise Application (NDES/App-Proxy) : r/entra - Reddit, accessed December 12, 2025, https://www.reddit.com/r/entra/comments/1e1bmiz/cannot_deploy_on_premise_application_ndesappproxy/
11. Configure Microsoft Intune – Certificates – Part 5: CNAME and Application Proxy, accessed December 12, 2025, https://albertneef.tech/2018/09/06/configure-microsoft-intune-certificates-part-5-cname-and-application-proxy/
12. Configure custom domains with Microsoft Entra application proxy, accessed December 12, 2025, https://learn.microsoft.com/en-us/entra/identity/app-proxy/how-to-configure-custom-domain
13. Intune SCEP Certificate Workflow Made Easy With Joy - Part 4 - Anoop C Nair, accessed December 12, 2025, https://www.anoopcnair.com/intune-scep-certificate-made-easy-with-joy-4/
14. Prepare your environment for SCEP Certificate Enrollment with Microsoft Intune, accessed December 12, 2025, https://msendpointmgr.com/2016/04/12/prepare-your-environment-for-scep-certificate-enrollment-with-microsoft-intune/
15. Azure AD Application Proxy and NDES/SCEP with Intune - Microsoft Learn, accessed December 12, 2025, https://learn.microsoft.com/en-us/answers/questions/570464/azure-ad-application-proxy-and-ndes-scep-with-intu