Automated Certificate Management Environment Working Group F. Geng
Internet-Draft P. Wu
Intended status: Standards Track Huawei Technologies
Expires: 20 November 2026 X. Chen
TrustAsia
19 May 2026

Automated Certificate Management Environment (ACME) Identity-Controlled Validation Extension
draft-geng-acme-idp-00

Abstract

This document defines an Identity Control Validation (ICV) framework for the ACME protocol [RFC8555], introducing a new ACME identifier type "idp" and a new challenge type "idp-01". The ICV framework allows ACME servers to delegate trusted Identity Providers (IdPs) to verify certificate applicants' control over the claimed identities.

This document also defines an optional X.509 certificate extension, the Trust-Domain-Restricted Certificate Extension, which explicitly indicates that certificates issued via the ICV framework are backed by a specific IdP trust domain and whose trustworthiness depends on the current status and policies of that IdP, preventing trust leakage into contexts outside the IdP's trust domain.

This extension is parallel to Domain Control Validation (DCV) [RFC8555]; either can independently verify an applicant's control over identities/resources and request certificates. Proof-of-Possession (PoP) [I-D.geng-acme-public-key] serves as an auxiliary enhancement framework. Based on this architecture, two standardized certificate enrollment models are formed: the Domain Validation model (DCV + optional PoP) and the Identity Validation model (ICV + optional PoP). This document focuses on the latter.

This document defines three deployment models: the IdP-Operated Certificate Model, the Intra-PKI Domain Mutual-Trust Model, and the CA-Integrated IdP Model. It supports multiple authentication protocols via the extensible idp_method parameter, covering both device and account identities.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 20 November 2026.

Copyright Notice

Table of Contents

1. Introduction 4
  1.1. Verification Dimensions in ACME 5
  1.2. Review of Prior Work and Scope of This Document 5
  1.3. Design Principles 7
  1.4. Solution Overview 7
  1.5. Relationship to Existing Work 8
2. Terminology and Definitions 9
3. ICV Framework Architecture 10
  3.1. Three Validation Dimensions and Three Major Protocol Frameworks 10
  3.2. Hierarchical Trust Model 11
  3.3. Two Standard Certificate Enrollment Modes 12
  3.4. Identity Identifier Types and Coverage Scope 13
  3.5. Synergy Matrix of ICV and PoP 13
4. Deployment Modes 14
  4.1. Mode 1: IdP-Operated Certificate Model 14
    4.1.1. Trust Establishment 15
    4.1.2. ACME-IDP Client 16
    4.1.3. Subsequent Identity Validation 17
  4.2. Mode 2: PKI Intra-Domain Mutual-Trust Model 17

4.3. Mode 3: CA-Integrated IdP Model 18
4.4. Mode Selection Guidance 18
5. ICV Framework Protocol Flow 19
5.1. Overall Interaction Model 19
5.2. ACME-ICV Client Behavior 21
6. Protocol Details 21
6.1. The "idp-01" Challenge Object 21
6.1.1. Trigger Conditions 22
6.1.2. Challenge Object Fields 22
6.2. IdP Authentication Methods 23
6.2.1. Account Identity Authentication Methods 23
6.2.2. Device Identity Authentication Methods 23
6.3. Format of the "acmeIdpToken" 24
6.4. Challenge Response and Server Validation 26
6.5. Trust-Domain-Restricted Certificate Extension 27
6.5.1. Extension Definition 27
6.5.2. Critical Marking Policy 28
6.5.3. Boundary with the RFC5280 certificatePolicies Extension 28
6.5.4. CA Writing Requirements 29
6.6. Error Types 29
6.7. IdP Operational Certificate Format and Interoperability 29
6.7.1. Certificate Format Requirements 30
6.7.2. Interoperability with Existing IdPs 30
7. Security Considerations 30
7.1. Trust Anchor Levels and Threat Model 31
7.2. Layered PoP Verification 31
7.3. Certificate Lifecycle and Authentication Strength 32
7.4. Timeliness 32
7.5. Privacy 32
7.6. Comparison with the SSO/OAuth Security Model 33
7.7. Comparison of Security Models between SSO/OAuth and idp-01 34
7.8. Comparison with Stand-Alone Use of RFC9447 36
7.9. Order Binding 37
7.10. Relying-Party Validation Logic for Trust-Domain-Restricted Certificates 38
7.10.1. Handling of Critical and Non-Critical Extensions . . 39
7.10.2. Relationship with Defense-in-Depth 39
8. IANA Considerations 39
9. Appendix A. Same-Domain Trust — Example of the IdP Operational Certificate Mode (Mode1, Stand-Alone ICV + CSR) 41
9.1. Phase 1: IdP Requests an Operational Certificate 41
9.2. Phase 2: User Requests an Identity Certificate 42
10. Appendix B. Same-Domain Trust — Example of CA-Integrated IdP Mode (Mode3) 42
10.1. Prerequisites 43

10.2. Workflow 43
10.3. Scenario Value 44
11. Appendix C. Same-Domain / Cross-Domain Trust — Enterprise PKI Bridging Scenario (Mode2) 44
11.1. Scenario 1: Same-Domain Trust 44
11.2. Scenario 2: Cross-Domain Trust 45
12. Appendix D. Cross-Domain Trust — E2EE Meeting Certificate Provisioning (Mode1 + Mode2) 45
12.1. Case 1: Pre-established Root-Certificate Mutual Trust Between Both Enterprises (Mode2) 46
12.2. Case 2: No Pre-established Root-Certificate Mutual Trust Between Enterprises — Guest Mode (Mode1) 46
13. Appendix E. Cross-Domain Trust — Automated C2PA Certificate Issuance (Mode1 / Mode2 / Mode3) 47
13.1. Requirements Description 48
13.2. Applicability of the Three Deployment Modes 48
13.3. Deployment Recommendations 48
14. Appendix F. Cross-Domain Trust — Federated Trust Scenarios (Mode2 + Federated Trust Model) 49
14.1. Scenario Description 49
14.2. Workflow 50
15. Appendix G. Stand-Alone ICV Trust with idp-PoP — Temporary Certificates for Public Device Scenarios 50
15.1. Participants and Prerequisites 50
15.2. Workflow 51
16. Appendix H. Enterprise Intranet Device Proxy Scenario (ACME-ICV Proxy Mode) 52
16.1. Scenario Description 52
16.2. Proxy-Mode Architecture 52
16.3. Proxy Workflow 53
16.4. Security Considerations and Limitations 54
17. Appendix I. Cross-Domain Trust — sigstore Code-Signing Scenario (Mode1 + OIDC-Bridged IdP) 54
17.1. Motivation 55
17.2. Alignment with This Draft 55
17.3. Simplified Workflow 56
17.4. Value 57
18. Acknowledgements 57
19. References 58
19.1. Normative References 58
19.2. Informative References 59
Authors' Addresses 60

1. Introduction

1.1. Verification Dimensions in ACME

ACME [RFC8555] models certificate enrollment as an "Order", which contains a set of "identifiers". Each identifier corresponds to an "authorization" holding a set of "challenges". Existing challenges such as "dns-01" and "http-01" address Domain Control Validation (DCV). However, business requirements for certificate issuance also include Proof-of-Possession (PoP) and Identity Control Validation (ICV). These three dimensions are mutually independent and shall be standardized as separate frameworks to allow flexible combination by CAs.

In non-DCV scenarios such as S/MIME, code signing, device certificates, and cross-enterprise trust-domain mutual recognition, the core identifiers of certificates shift from domain names to email addresses, accounts, device serial numbers, or user identity IDs. Accordingly, the security foundation must change from "control of communication resources" to "control of the claimed identity". The current ACME ecosystem lacks a general-purpose identity validation framework under autonomous CA control. The ICV framework defined in this document aims to fill this gap.

From a PKI architectural perspective, this gap also reflects a deeper structural issue: the lack of standardization of Registration Authority (RA) functions. In traditional PKI deployments, the RA verifies the identity of certificate applicants and forwards their requests to the CA; however, RA implementations across different CAs are not interoperable with one another. By formalizing the role of an IdP as "an entity holding an X.509 operational certificate with a specific EKU" (see Section4.1 for details), this document transforms the RA process previously relying on proprietary interfaces or manual workflows into a standardized authorized entity that can be automatically managed via the standard ACME protocol and validated through standard X.509 path validation.

1.2. Review of Prior Work and Scope of This Document

The ACME Working Group has undertaken several exploratory efforts in the area of non-DCV identity validation. This section briefly reviews the core contributions and remaining issues of these works to clarify the scope and positioning of this document.

(a) [I-D.biggs-acme-sso]:

This draft proposes the "sso-01" challenge and introduces the concept of involving third-party identity providers in ACME validation. This document inherits the "third-party IdP integration" concept from that draft, fully returns trust-anchor control to the CA (see Section7.6 for details), and extends it into a more general ICV framework.

(b) [RFC8823]:

This specification defines the "email" identifier type and the "email-reply-00" challenge to support automated issuance of end-user S/MIME certificates. This document does not modify the protocol behavior of [RFC8823]; "idp-01" and "email-reply-00" may coexist and are selected by the CA according to policy — idp-01 elevates the security foundation to "proof of control over an identity".

(c) [I-D.ietf-acme-client]:

This draft pioneered the identification of automation requirements for end-user, device, and code-signing certificates, describing applicable use cases for ACME in these scenarios. Building upon those scenarios, this document further provides a standardized identity validation mechanism.

(d) [I-D.ietf-acme-telephone]:

This draft explored validating telephone number identifiers via external authorities, extending ACME's applicability to non-DNS identifiers. This document further expands the validation scope to more general identity control scenarios.

(e) [I-D.ietf-acme-device-attest]:

This draft leverages hardware trust roots such as TPM and WebAuthn to verify device identities, targeting device trust-root and hardware attestation scenarios. Complementary to it, this document covers two dimensions respectively: "what the device is" and "who controls the device".

(f) [RFC9447] ("tkauth-01") and [I-D.ietf-acme-openid-federation]:

The former establishes a general-purpose authoritative token challenge framework, while the latter introduces the multi-layer trust chain mechanism of OpenID Federation. Both provide important references for the design of acmeIdpToken and cross-domain trust propagation in this document. [RFC9447] does not independently address identity validation issues; the boundary between this document and [RFC9447] is detailed in Section 7.8.

In summary, existing efforts have advanced ACME adoption in non-DCV scenarios from perspectives including identifier extension, challenge mechanisms, and trust-chain collaboration. This document focuses on defining a general-purpose, independently deployable identity control validation framework with CA-led trust anchors, forming a parallel, complementary, and extensible relationship with existing work.

1.3. Design Principles

* Separation of Concerns: ICV solely validates identity control. DCV and PoP are handled by existing frameworks.

* Prioritize closed-loop trust within the PKI ecosystem: Reuse the existing X.509 and ACME standards stack.

* CA Trust Anchor Supremacy over IdP: The CA proactively authorizes IdPs and retains final decision-making authority over the certificate lifecycle.

* Pluggable Authentication Mechanisms: Different authentication protocols are adapted via the idp_method parameter. This document defines a unified ACME challenge flow, while specific authentication implementations are selected by the IdP according to scenarios.

1.4. Solution Overview

This document defines the core components of the ICV framework: the "idp" identifier type and the "idp-01" challenge type. ICV and DCV are peer-level mechanisms; each can complete validation independently and request certificates via CSR. PoP serves as an auxiliary enhancement framework and may be used in combination with ICV.

The three deployment modes correspond to different relationships between CAs and IdPs respectively. Mode 1 (IdP Operational Certificate Model) applies to non-CA IdPs; Mode 2 (Intra-PKI Domain Mutual-Recognition Model) applies to scenarios where both parties operate private CAs; Mode 3 (CA-Integrated IdP Model) applies to scenarios where the CA directly manages user/device identities itself. The external protocol interactions remain consistent across all three modes, with differences only in internal trust-establishment paths. See Section 4 for details.

The ICV framework involves two specialized variants of ACME clients:

* ACME-ICV Client: Deployed on end-user/device side, it is responsible for registering identity certificates for users or devices using the "idp-01" challenge.

* ACME-IDP Client: Deployed on the IdP side and used only in Mode 1, it encapsulates the logic for enrollment, renewal, and revocation of operational certificates.

The detailed behaviors of the two types of clients are defined in Sections 4.1 and 5.2, respectively.

This document also defines an optional X.509 certificate extension, trust_domain_restriction (see Section 6.5). CAs *MAY* include this extension when issuing certificates generated via the ICV path to explicitly indicate that the trustworthiness of such certificates relies on a specific IdP trust domain. The extension is defined as non-critical by default to ensure deployment compatibility with the existing TLS/S/MIME ecosystem; CAs *MAY* optionally mark it as critical based on deployment context. Design rationale is provided in Section 3.2, and relying-party validation logic is specified in Section 7.10.

By default, the acmeIdpToken includes a bound_to_order claim that cryptographically binds the token to the byte-level hash of the current "newOrder", aligning with the order-binding semantics of the "pk-01" challenge defined in [I-D.geng-acme-public-key]. See Sections 6.3 and 7.9 for details.

1.5. Relationship to Existing Work

The table below summarizes the relationship between this document and existing ACME work; detailed comparisons are provided in Sections 7.6 through 7.8.

Existing Documents	Relationship
RFC 8555	Parallel (DCV vs. ICV)
RFC 8823 (email-reply-00)	Enhanced security, coexistent
draft-ietf-acme-client	Requirements vs. Implementation
draft-biggs-acme-sso (Expired)	Inherited principles
RFC 9447 (tkauth-01)	Complementary: Identity vs. Authorization
draft-ietf-acme-openid-federation	Hierarchical Collaboration
draft-ietf-acme-device-attest	Complementary: Hardware vs. Subject
draft-ietf-acme-profiles	Template selection, complementary
draft-geng-acme-public-key (pk-01)	Auxiliary PoP Enhancement

This document has no overlap or conflict with existing work in terms of motivation, security model, or protocol behavior.

Note: All references to [I-D.draft-geng-acme-public-key] within this document refer to the unsubmitted version 07.

2. Terminology and Definitions

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*", "*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*", "*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

* ACME Server (AS): A CA that implements the ACME protocol and issues certificates.

* Identity Provider (IdP): An independent service trusted by the AS that verifies a requester’ s control over a claimed identity. In Mode3, the IdP functionality is performed by the CA itself. In Mode1, the IdP acts as a formally authorized RA of the CA by holding an operational certificate issued by the CA.

* ACME-ICV Client: A variant of the ACME client deployed on the EE side, responsible for registering identity certificates using the "idp-01" challenge.

* ACME-IDP Client: A variant of the ACME client deployed on the IdP side, responsible for automated full-lifecycle management of IdP operational certificates. Used only in Mode1.

* IdP Identifier: An identity identifier declared in "newOrder". Its value field is a string that identifies the claimed identity, formatted as a URI to ensure namespace isolation across different identity types (e.g., mailto:user@example.org (https://mailto:user@example.org), urn:vin:1HGCM82633A123456, urn:sn:DEV-XYZ-001).

* idp-01 Challenge: An ACME challenge type that delegates identity control validation to the IdP.

* acmeIdpToken: A JWT [RFC7519] issued by the IdP to prove completion of identity authentication. It is transmitted under the field name "acmeIdpToken" in the ACME challenge response message. It may optionally include a bound_to_order claim to cryptographically bind the token to the current newOrder (see Sections 6.3 and 7.9).。

* idp_method: Identifier of the authentication protocol used by the IdP.

* IdP Operational Certificate: An X.509 certificate issued by the CA to the IdP, with its EKU containing id-kp-acmeIdpTokenSigning. It serves as the core mechanism to formally authorize the IdP as a CA-endorsed RA. This certificate is used only in Mode1; its format and extension definitions are specified in Section6.7. The CA may require the IdP to apply for operational certificates of different validation levels (e.g., DV or OV) per security policies.

* Trust-Domain-Restricted Certificate: An X.509 certificate containing the trust_domain_restriction extension defined in this document (see Section6.5). This extension explicitly indicates that the trustworthiness of the certificate relies on the specified IdP trust domain, and relying parties shall validate it in conjunction with IdP status and policies.

* Certificate Public Key: The public key bound to the finally issued certificate. In the ICV + PoP combined branch, this public key is derived from the value field of the "pk" identifier in newOrder (i.e., the base64url-encoded SubjectPublicKeyInfo [RFC5480], in accordance with [I-D.geng-acme-public-key]); in the standalone ICV trust-idp-PoP branch, it is taken from the confirmed_public_key claim of acmeIdpToken; in the standalone ICV + CSR branch, it comes from the PKCS#10 CSR provided during the finalize phase.

3. ICV Framework Architecture

3.1. Three Validation Dimensions and Three Major Protocol Frameworks

The ACME protocol suite comprises three independent validation dimensions, each corresponding to one protocol framework.

DCV and ICV are in a parallel relationship; both can independently complete identity validation and request certificates. PoP is an auxiliary enhancement framework.

Dimension	Framework	Core Question
Domain Control Validation (DCV)	RFC 8555	Do you control this domain name?
Identity Control Validation (ICV)	idp-01	Do you control this identity?
Proof-of-Possession (PoP)	pk-01	Do you possess this private key?

Figure 1: Three Validation Dimensions and Three Major Protocol Frameworks

The relationship between identifier types and frameworks is as follows:

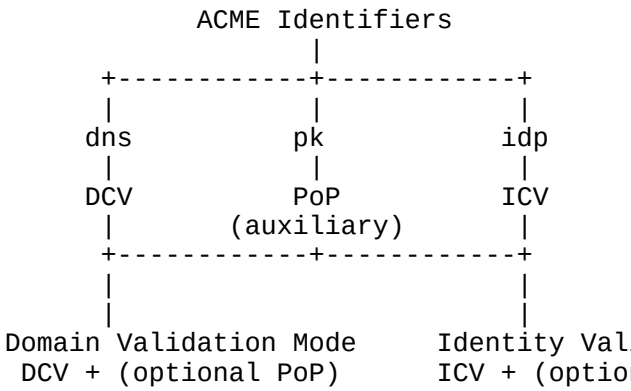


Figure 2: Relationship Between Identifier Types and Frameworks

3.2. Hierarchical Trust Model

The ICV framework introduces a hierarchical trust structure into ACME. Unlike traditional ACME (where the CA performs end-to-end notarization directly for domain names), this model can be summarized with three roles:

* IdP: Performs substantive validation of an applicant’ s identity within its trust domain and outputs signed identity assertions in the form of "acmeIdpToken".

* CA: Converts identity assertions from the IdP into final certificates complying with the X.509 specification and manages certificate lifecycle. The relationship between the CA and the IdP varies by deployment mode: Mode1 establishes authorization by issuing IdP operational certificates; Mode2 enables mutual trust via pre-configured root certificates; Mode3 has the CA act as the IdP itself.

* Relying Party: When using a certificate, in addition to performing standard PKI path validation, it MAY recognize the trust_domain_restriction extension (see Section6.5) to determine whether to rely on the current status and policies of the IdP.

The core implication of this model is: when a certificate includes the trust_domain_restriction extension, trust is jointly underpinned by the CA and the IdP — the CA ensures compliance of the issuance process, while the IdP guarantees the authenticity of identity assertions. Failure of either party (e.g., revocation of the IdP's operational certificate, disabling of an account via IdP policy) *SHOULD* affect a relying party's acceptance of the certificate.

This model does not require relying parties to perform IdP status checks for all ICV-issued certificates. Whether such checks are conducted is determined jointly by the presence of the trust_domain_restriction extension in the certificate and the relying party's local policy. Detailed relying-party validation logic is provided in Section7.10.

The hierarchical trust described in this section does not conflict with the design principle of "CA trust anchor takes precedence over IdP" in Section1.3, as they operate at different layers:

* Framework Layer (§1.3): The CA acts as the trust anchor of the ICV framework itself — it determines which IdPs may participate (by issuing or refusing operational certificates), and governs final certificate issuance and revocation. IdPs cannot obtain authorization autonomously within the framework.

* Content Layer (this section): For identity attributes and the assertion that "the applicant controls the claimed identity", the IdP is the substantive source. The CA converts the IdP's assertions into X.509 format without re-validating the identity content itself.

The statement in Section7.1 that "the CA trust anchor takes precedence over the IdP" adopts the semantics of the framework layer and is not inconsistent with this section.

3.3. Two Standard Certificate Enrollment Modes

DCV and ICV verify the requester's control over domain names and identities, respectively. Since a single certificate cannot be both a domain-validated certificate and an identity-validated certificate, DCV and ICV are mutually exclusive in practice. Based on this orthogonality, two standard certificate enrollment modes exist.

* *Domain Validation Mode*: DCV + optional PoP. Typical combinations: {dns-01} or {dns-01, pk-01}.

* *Identity Validation Mode*: ICV + optional PoP. Typical combinations: {idp-01}, {idp-01, pk-01}, or the standalone ICV-trusted idp-PoP branch (not combined with pk-01, where the IdP already provides PoP for the certificate public key during authentication).

The identity validation mode is subdivided into three branches, whose differences are summarized collectively in the synergy matrix in Section3.5.

3.4. Identity Identifier Types and Coverage Scope

This document defines the "idp" identifier type, whose value field identifies the claimed identity. The ICV framework covers two broad categories of identities:

* Physical Device Identity: Device serial numbers, vehicle VINs, etc. Attestation protocols are typically provided by chip vendors or device manufacturers. Such identity identifiers do not inherently imply a public key. When the IdP establishes PoP of the device private key during identity validation (e.g., TPM endorsement signatures, WebAuthn assertions), the standalone ICV-trusted idp-PoP branch applies (see Appendix H); otherwise, the standalone ICV + CSR branch applies.

* Subject (Human or AI) Account Identity: User email addresses, employee IDs, AI agent identifiers, and asserted identities indirectly held by other subjects upon human authorization. Attestation protocols are provided by the IdP, and any branch may be used.

3.5. Synergy Matrix of ICV and PoP

The identity validation mode is subdivided into three branches, whose differences are summarized below in a consolidated manner.

Branch	newOrder identifiers	Certificate Public Key Source	finalize body
Standalone ICV + CSR	[{idp}]	SPKI from the CSR submitted during the finalize phase	{"csr":...}
Standalone ICV (IdP-PoP Trusted)	[{idp}]	acmeIdpToken.confirmed_public_key	{} empty object
ICV + PoP	[{idp}, {pk}]	SPKI decoded from the value of the "pk" identifier	{} empty object

Figure 3: Synergy Matrix of the Three Branches of Identity Validation Mode

Applicability Notes:

* Standalone ICV + CSR applies to scenarios where identity authentication and certificate keys are decoupled, e.g., multiple-purpose certificates associated with the same IdP identity (see Appendix A).

* Standalone ICV with IdP-PoP trust applies to scenarios where the identity key and certificate key are consolidated, or deployments such as shared public devices where an additional pk-01 challenge is impractical (see Appendices G and H).

* ICV + PoP applies to high-security scenarios requiring direct CA validation of certificate private-key ownership (acme-PoP), or post-quantum KEM algorithms for which CSRs are not feasible.

4. Deployment Modes

The external protocol interactions remain consistent across the three deployment modes; they differ only in how the CA establishes trust in IdPs and how trust anchors are configured. This section describes each mode individually, followed by selection guidance provided in Section4.4.

4.1. Mode 1: IdP-Operated Certificate Model

This mode applies to non-CA IdPs, including enterprise internal LDAP/ OAuth services, content platform user systems, social media identity systems, and others. The CA establishes trust by issuing dedicated operational certificates to the IdP, enabling the IdP to participate in the ICV framework without operating its own CA.

An IdP-operated certificate is a standardized RA credential for an IdP to join the ICV framework; its EKU includes id-kp-acmeIdpTokenSigning (see Section6.7 for details). This mechanism replaces vendor-specific RA-CA integration interfaces and manual authorization processes used in traditional PKI with a single ACME certificate enrollment. Any ACME-capable IdP only needs to apply for one operational certificate to obtain RA qualification.

4.1.1. Trust Establishment

The IdP applies to the CA for a dedicated operational certificate (EKU id-kp-acmeIdpTokenSigning). The CA verifies the IdP's eligibility per policy. Depending on the risk level, verification may be categorized as follows:

* DV Level: Verifies the IdP's control over its domain via the "dns-01" or "http-01" challenge. Applicable to low-risk scenarios.

* OV/EV Level: Requires the IdP to complete Organization Validation (OV) or even Extended Validation (EV). OV-level validation can be automated via the following approaches:

  (1) Pre-validation plus EAB mechanism: The IdP first submits organizational identity materials to the CA via out-of-band means (or initial manual review). After the CA completes verification, it issues an External Account Binding (EAB) credential for the IdP. Subsequent ACME-IDP clients present the EAB when registering accounts or requesting operational certificates, allowing the CA to automatically associate the account with the pre-validated organization.

  (2) Manual administrator approval: For on-premises deployments, CA administrators may confirm the IdP's organizational identity through a manual approval workflow.

Upon successful validation, the CA issues the operational certificate. It is RECOMMENDED that the operational certificate have a validity period of no more than one year and support ACME renewal. The CA *SHOULD* explicitly state the validation-level requirements for such operational certificates in its Certificate Practice Statement (CPS).

4.1.2. ACME-IDP Client

The ACME-IDP client is a specialized variant of the ACME client deployed on the IdP side. It encapsulates logic for requesting, renewing, and revoking operational certificates on top of standard ACME client behavior (per [RFC8555]).

The ACME-IDP client SHOULD support the following functions:

* Initial Enrollment: Upon initial IdP deployment, automatically create an ACME account (or use a pre-provisioned account key), submit a "newOrder" (declaring the IdP identity identifier with profile [I-D.ietf-acme-profiles]: "idp-op-cert"), complete CA-required validation challenges, submit a CSR, and obtain the operational certificate. The client *SHOULD* support the EAB mechanism to enable automated issuance of OV certificates.

* Automated Renewal: Automatically renew the operational certificate prior to its expiration. Renewal is *RECOMMENDED* to be triggered when 30% of the certificate validity period remains.

* Automated Revocation: Automatically submit a revocation request to the CA when the IdP no longer acts as an IdP.

* Challenge Handling: *SHOULD* support automated processing of the "dns-01" or "http-01" challenges.

The private key of the operational certificate for the ACME-IDP client *MUST NOT* be persistently stored in plaintext. Use of an HSM, TEE, or OS-level secure key storage mechanism is RECOMMENDED. In case of private-key compromise or suspected compromise, the current operational certificate *MUST* be revoked immediately and a new one re-enrolled.

The ACME-IDP client does not participate in online interactions for end-user identity authentication; it only manages operational certificates on the IdP side.

Client Type	Deployment Location	Certificate Type	Core Functionality
Standard ACME Client	EE-side (Domain Scenarios)	Domain Certificate Certificate Type	DCV + Certificate Enrollment
ACME-IDP Client	IdP-side	IdP Operational Certificate	Operational Certificate Lifecycle Management
ACME-ICV Client	EE-side (Identity Scenarios)	Identity Certificate	ICV + Identity Certificate Enrollment

Figure 4: Client Roles in the ICV Framework

4.1.3. Subsequent Identity Validation

After the IdP obtains an operational certificate, it uses the corresponding certificate private key to sign the acmeIdpToken in subsequent "idp-01" challenges. When validating the token, the CA verifies the operational certificate via standard X.509 certificate path validation. The validation level (DV / OV / EV) of the operational certificate determines the CA's trust in the IdP, which further constrains the permitted types of end-entity certificates issued.

4.2. Mode 2: PKI Intra-Domain Mutual-Trust Model

This mode is used when both the CA and the IdP operate private CAs. Deploying private CAs is standard practice in enterprise IT environments (e.g., Windows Server AD CS, EJBCA). When end-to-end secure communication is required between two organizations, each party most likely runs its own private CA.

In this mode, participants exchange root certificates based on a bilateral mutual-trust agreement, and the ACME server directly configures the peer CA's root or subordinate CA certificate as a trust anchor. The peer CA directly signs the acmeIdpToken using its existing CA certificate. The issuing CA retains final authority over the lifecycle of end-entity certificates. This mode can be cross-domain (where both parties belong to separate organizations) or intra-domain (between CAs of different departments or tiers within a single organization).

In cross-domain deployments, the CA *SHOULD* include the trust_domain_restriction extension in issued certificates, where the idpIdentifier points to the peer-organization IdP participating in mutual trust, enabling relying parties to correctly interpret the certificate's trust scope.

4.3. Mode 3: CA-Integrated IdP Model

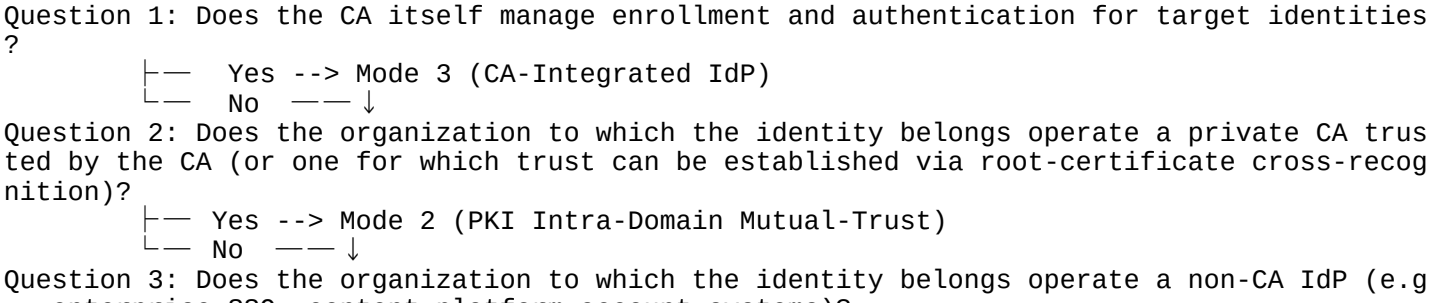
This mode applies when the CA itself directly manages identity enrollment and authentication for users or devices. The CA concurrently acts as the IdP—performing identity registration and control validation, and issuing certificates accordingly. The idp_url in the "idp-01" challenge points to the CA's own authentication endpoint, or the CA may directly complete challenge validation based on internal authentication results.

This mode has seen extensive product-grade industry adoption, including integrations such as Windows Server AD CS with Active Directory, HashiCorp Vault PKI (ACME-enabled since version 1.14), Foxpass Cloud PKI with Microsoft Entra ID / Google Workspace, and GlobalSign Auto Enrollment Gateway with AD.

In this mode, the CA functions as a unified identity and certificate management platform with no need for an external IdP. External protocol interactions for "idp-01" remain unchanged to ensure interoperability across all three modes. Since the trust domains of the CA and IdP overlap, the CA generally does not need to include the trust_domain_restriction extension in issued certificates, unless it explicitly intends to mark the authentication source for consistency with certificates issued under Mode 1 and Mode 2.

4.4. Mode Selection Guidance

The following decision criteria help deployers select the appropriate mode:



In practice, all three modes may coexist within a single CA: the CA may use Mode3 for its own organization, Mode2 for federation partners, and Mode1 for external IdPs.

Dimension	Mode 1	Mode 2	Mode 3
CA-IdP Relationship	Separated (Non-CA IdP)	Separated (Both are CAs)	Integrated (CA = IdP)
Trust Anchor	Operational certificate issued by the CA	Mutual root certificates	CA itself
Trust Domain	Intra-domain	Cross-domain/intra-domain	Intra-domain
IdP Client	ACME-IDP Client	Not required	Not required
Typical Use	Enterprise IdP + External CA	Inter-enterprise PKI mutual-trust	AD CS + AD, Vault PKI, Foxpass
Privacy	Low to Medium	Medium to High	Low to Medium

Figure 5: Comparison of the Three Deployment Modes

5. ICV Framework Protocol Flow

This section describes the overall behavior of the ICV framework from client- and process-oriented perspectives. Section5.1 presents the conceptual timeline of three-party interactions, while Section5.2 enumerates the specific responsibilities of the ACME-ICV client. Field formats, triggering conditions, validation rules, and extension definitions on the server/IdP side are specified in Section6.

5.1. Overall Interaction Model

The ICV framework follows the identifier-authorization-challenge architecture of the ACME protocol. External interactions for the "idp-01" challenge remain consistent regardless of the deployment mode.

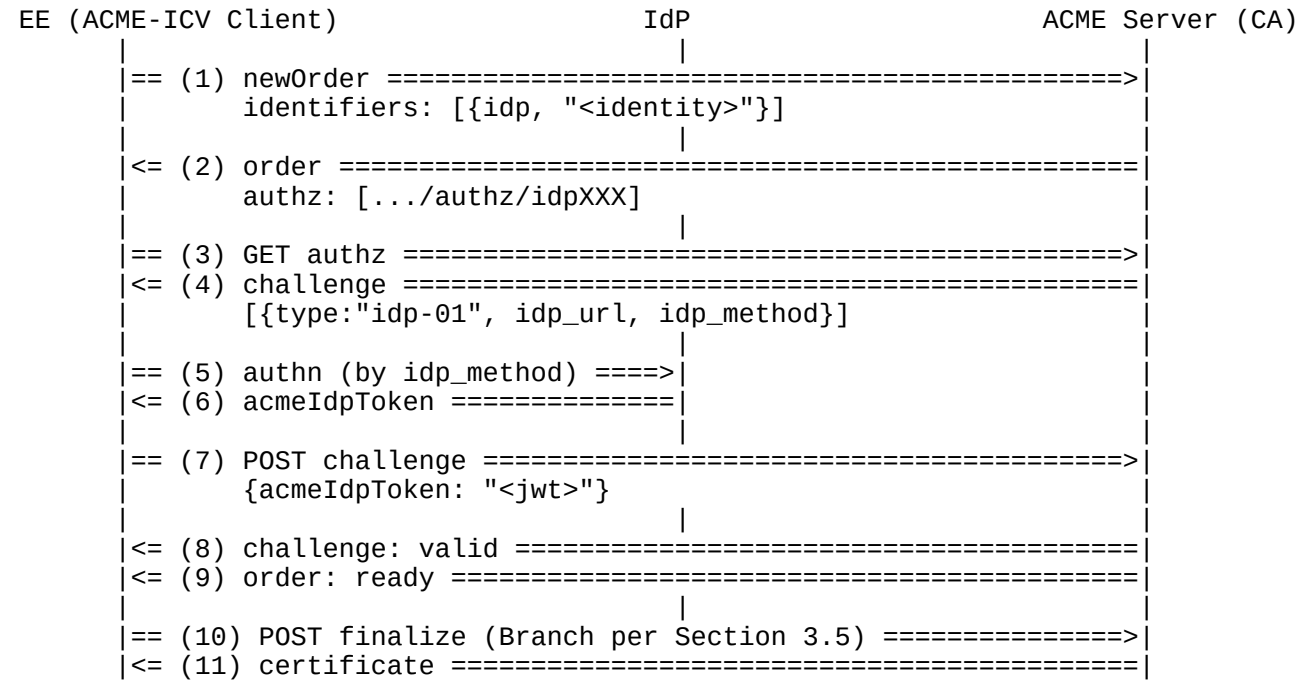


Figure 6: Overall Interaction Model of the ICV Framework (Conceptual View)

Notes:

1. In Mode3, the IdP and CA are operated by the same entity. Steps (5)(6) are performed internally. The CA may still optionally issue an acmeIdpToken externally to maintain interoperability. Regardless of the deployment mode, acmeIdpToken by default includes the bound_to_order claim to bind the token to the current "newOrder" (see Section6.3).

2. The difference between the standalone ICV+CSR branch and the standalone ICV trusting idp-PoP branch occurs only at Step(10). The combined ICV+PoP branch additionally declares the "pk" identifier in the "newOrder" and independently completes the "pk-01" challenge (see [I-D.geng-acme-public-key]).

5.2. ACME-ICV Client Behavior

The ACME-ICV client is a variant of the ACME client deployed on the EE side, used to request identity-bound certificates via the "idp-01" challenge. Building on standard ACME client behavior (per [RFC8555]), it encapsulates interactions with the IdP as well as logic for obtaining and submitting the acmeIdpToken. Regardless of the deployment mode adopted, the core behavior of the ACME-ICV client remains consistent: the client only needs to know the idp_url and idp_method, and does not need to perceive whether the IdP is an external standalone entity or integrated within the CA itself.

The ACME-ICV client *SHOULD* support the following behaviors:

* Identifier Declaration: Declare the "idp" identifier in the "newOrder" request to trigger the "idp-01" challenge (trigger conditions specified in Section6.1.1), where its value represents the identity of the user or device.

* raw_newOrder Caching: The client *MUST* retain the exact byte sequence of the JWS protected payload of the "newOrder" request locally, and compute SHA-256(raw_newOrder) using the identical byte sequence during the "idp-01" challenge phase. The client *MUST NOT* re-serialize parsed JSON for use as raw_newOrder (see Section6.3).。

* "idp-01" Challenge Handling: Identify the "idp-01" challenge within the authorization object, parse the idp_url and idp_method fields, interact with the IdP via the specified authentication protocol to complete identity authentication. When requesting the IdP to issue an acmeIdpToken, the client shall pass base64url(SHA-256(raw_newOrder)) to the IdP as the bound_to_order hint (the specific delivery mechanism is defined per individual idp_method), and finally obtain the acmeIdpToken and submit it to the challenge URL.

* Finalization-phase Processing: Submit the CSR, an empty request, or coordinate with the validation result of the "pk-01" challenge according to the selected branch defined in the coordination matrix in Section3.5.

6. Protocol Details

6.1. The "idp-01" Challenge Object

6.1.1. Trigger Conditions

When the identifiers array in a "newOrder" request contains an identifier of type "idp", the ACME server *MUST* create an authorization object for this identifier, and the challenges array of this authorization object *MUST* contain exactly one challenge of type "idp-01".

The server *SHOULD NOT* return an "idp-01" challenge for identifiers of non-"idp" types.

The server *MAY* offer other future-defined compatible challenge types side-by-side within the same authorization object; the client *MAY* select any one of them to complete validation per local policy (per [RFC8555] § 7.1.4).

6.1.2. Challenge Object Fields

The challenge object extends the standard fields defined in [RFC8555] § 7.1.5 and § 8 (such as type, url, status, validated, and error) with the following additional fields:

Field	Req.	Description
idpIdentifier	Yes	Unique identifier generated by the server for this challenge
idp_url	Yes	IdP validation endpoint URL
idp_method	Yes	Authentication protocol (see Section6.2)
idp_cert_fingerprint	No	Fingerprint of the IdP-operated certificate (Mode1)
deployment_mode	No	"idp-op-cert", "pki-intra", or "ca-idp"

Figure 7: Fields of the "idp-01

The idpIdentifier is independently generated by the server for each challenge with no less than 128 bits of entropy. It serves as a unique challenge identifier and anti-replay mechanism, ensuring that an acmeIdpToken issued for one "idp-01" challenge cannot be re-consumed by the server. Byte-level order binding is enforced by the bound_to_order claim within the acmeIdpToken (see Section6.3).

The two mechanisms are complementary: idpIdentifier prevents replay at the challenge level, while bound_to_order prevents cross-order replay of tokens.

6.2. IdP Authentication Methods

The CA selects the authentication protocol adopted by the IdP via the idp_method field. Values of idp_method are managed jointly by the initial set defined in Sections6.2.1 and 6.2.2 of this document and the IANA registry (Section8), covering both account identities and device identities.

6.2.1. Account Identity Authentication Methods

* pkic: The IdP holds a certificate trusted by the AS. If the identity exists in the form of a public key, the IdP verifies that the applicant possesses the private key bound to the identity via idp-PoP (e.g., requiring the applicant to sign the idpIdentifier, with the IdP verifying the signature using the public key from its trusted certificate).

* opaque [RFC9807]: Password-Authenticated Key Exchange, verifying possession of the private key bound to the identity during the AKE phase.

* webauthn [WebAuthn]: Hardware-bound two-factor authentication.

* internal: Used for Mode3 (CA-integrated IdP). This method indicates that IdP functionality is performed by the CA itself, which directly authenticates the applicant's identity using its internal identity store. No external JWT token is required for identity verification; for protocol interoperability, the CA may optionally issue an acmeIdpToken.

6.2.2. Device Identity Authentication Methods

Device Identity Authentication Methods are used to verify a device's control over its claimed physical identity identifier. Trust anchors are typically third-party entities independent of the CA (e.g., chip vendors, device manufacturers).

* device-puf: Based on Physical Unclonable Functions.

* device-tpm: Based on TPM Endorsement Keys.

* device-vin: Based on Vehicle Identification Numbers.

* device-sn: Based on device serial numbers.

Identifier	Type	Description	Reference
pkic	Account	PKI certificate chain validation	This document
opaque	Account	OPAQUE protocol	[RFC9807]
webauthn	Account	WebAuthn Level 2	[WebAuthn]
internal	Account	Internal CA authentication	This document
device-puf	Device	PUF challenge-response validation	This document
device-tpm	Device	TPM EK validation	This document
device-vin	Device	Vehicle VIN validation	This document
device-sn	Device	Device serial number validation	This document

Figure 8: Initial Set of IdP Authentication Methods (Extensible for Future Use)

6.3. Format of the "acmeIdpToken"

The JWT [RFC7519] issued by the IdP (encoded as JWS [RFC7515]) contains the following claims:

Claim	Req.	Description
iss	Yes	IdP identifier
sub	Yes	Applicant identity identifier (may be pairwise pseudonym)
aud	Yes	AS identifier
iat	Yes	Issued-at time
exp	Yes	Expiration time (recommended 5 minutes)
jti	Yes	Unique token identifier (one-time use)
idpIdentifier	Yes	Matches the value in the challenge object
idp_method	Yes	Authentication method used
bound_to_order	Yes*	base64url(SHA-256(raw_newOrder)). The client informs the IdP of the byte-level hash of the current newOrder during IdP authentication; the IdP echoes this value in this claim to cryptographically bind the token to the current order (see Section7.9).
nbf	No	Standard JWT [RFC7519] claim; if present, the CA MUST validate per RFC7519 §4.1.5.
confirmed_public_key	No	Present only when the IdP uses a public-key authentication protocol; carries the base64url-encoded SPKI of the certificate public key

Figure 9: ‘acmeIdpToken‘ JWT Claims

*Note** bound_to_order: Required by default (*SHOULD*). The CA SHOULD require this claim to be present and match during validation. For specific backward-compatibility scenarios, the CA policy MAY allow its absence; however, the CA *MUST* explicitly document the security consequences of accepting weak binding in its CPS (see Section7.9).

The exact byte definition of raw_newOrder is consistent with §10.1.1 in [I-D.geng-acme-public-key]: the raw byte sequence of the JWS protected payload after BASE64URL-DECODE and before JSON parsing. The client *MUST NOT* re-serialize the parsed JSON for use as raw_newOrder.

6.4. Challenge Response and Server Validation

ACME-ICV clients submit to the challenge URL via a standard authenticated ACME POST request:

```
{"acmeIdpToken": "<jwt>"}
```

Upon receiving the response, the server performs validation following these steps:

1. Determine the trust anchor according to the deployment mode (operational certificate path / cross-recognized root certificate / internal identity system).

2. For Mode1 / Mode2, verify the JWS signature of the acmeIdpToken and validate the IdP certificate path per [RFC5280]. For Mode3, the CA's internal identity system directly provides the validation result.

3. Validate token claims according to the following rules:

   a. iss and aud *MUST* match the expected values from the challenge object;

   b. exp *MUST* be later than the current time. If the token includes nbf, it *MUST* be earlier than or equal to the current time (per [RFC7519] §4.1.5);

   c. jti *MUST NOT* have been consumed within the jti replay window (see Section7.4);

   d. idpIdentifier and idp_method *MUST* match those in the challenge object.

4. If the token contains the bound_to_order claim, the server *MUST* perform a byte-level comparison against the SHA-256 value computed server-side from the persisted raw_newOrder (or equivalent storage). Return urn:ietf:params:acme:error:badIdpToken on mismatch. If CA policy requires this claim to be present but it is missing from the token, the server *SHOULD* reject the request likewise.

5. Establish the source of the certificate public key per the selected branch (see Section3.5).

6.5. Trust-Domain-Restricted Certificate Extension

This section defines a new X.509 certificate extension trust_domain_restriction, which a CA *MAY* include when issuing an ICV certificate to explicitly indicate that the certificate's trustworthiness depends on the specified IdP trust domain.

6.5.1. Extension Definition

Proposed OID: id-pe-acme-idp-trust-domain, assigned under the PKIX private extension arc (final value to be allocated by IANA, see Section8). A certificate *SHALL* contain at most one trust_domain_restriction extension.

ASN.1 Syntax:

```
id-pe-acme-idp-trust-domain OBJECT IDENTIFIER ::= { TBD }

TrustDomainRestriction ::= SEQUENCE {
    idpIdentifier         UTF8String,
    idpPolicyURI          IA5String     OPTIONAL,
    idpAssertedAttributes SEQUENCE OF AssertedAttribute
                                        OPTIONAL
}

AssertedAttribute ::= SEQUENCE {
    type     OBJECT IDENTIFIER,
    value    UTF8String  -- Length MUST be  256 octets
}
```

Field Description:

* idpIdentifier: The URI of the IdP (matching idp_url in the challenge object), or the Subject DN of the IdP's operational certificate.

* idpPolicyURI: Optional. A URI pointing to the IdP's current certificate policy statement, enabling relying parties to retrieve the latest policy on demand.

* idpAssertedAttributes: Optional. A subset of attributes asserted by the IdP in the acmeIdpToken. Attribute types *MUST* be selected from the IANA-registered allowed set (see Section8); the CA *MUST NOT* pass unregistered-type attributes into this field.

6.5.2. Critical Marking Policy

This extension *SHOULD* be issued as non-critical by default, ensuring that existing relying parties such as TLS libraries and S/MIME clients that do not recognize this extension can still read standard certificate fields. This is a deployment-compatibility choice for this version.

A CA *MAY* mark this extension as critical under the following conditions:

* The CA knows that all target relying-party implementations support this extension; or

* Deployment within a closed ecosystem (e.g., enterprise intranets, industry-specific PKI), where progress of relying-party upgrades can be guaranteed.

Marking this extension as critical forces relying parties that do not recognize the extension to reject the certificate, providing fail-closed security guarantees; however, this comes at the cost of blocking unupgraded relying parties. A CA *SHOULD* clearly document the timeline for critical-flag upgrades and the affected deployment contexts in its Certificate Practice Statement (CPS).

6.5.3. Boundary with the RFC5280 certificatePolicies Extension

[RFC5280] §4.2.1.4 defines the certificatePolicies extension for declaring policy OIDs that a certificate adheres to. The trust_domain_restriction extension differs in the following aspects:

* certificatePolicies defaults to non-critical and carries semantics oriented toward a "policy compliance statement"; trust_domain_restriction carries runtime-resolvable idpIdentifier and an optional idpPolicyURI, enabling relying parties to actively verify the current state of the IdP.

* certificatePolicies cannot convey the set of attributes asserted by the IdP at issuance time (idpAssertedAttributes), which serve as part of audit and policy inputs within the ICV framework.

* certificatePolicies is typically used by relying parties to “identify whether a certificate conforms to an expected policy category” and is not employed for real-time validation logic.

A CA *MAY* use both extensions concurrently: certificatePolicies to mark the policy OID associated with an ICV certificate, and trust_domain_restriction to carry information required for real-time IdP validation. The two extensions do not conflict with each other.

6.5.4. CA Writing Requirements

When a CA decides to issue a certificate containing the trust_domain_restriction extension:

* idpIdentifier *MUST* match the iss claim within the acmeIdpToken and the idp_url in the idp-01 challenge object;

* The content of idpAssertedAttributes shall be taken directly from the corresponding field in the acmeIdpToken, and attribute types *MUST* be IANA-registered (see § 8);

* A CA *SHOULD NOT* map attributes asserted by the IdP into Subject Directory Attributes or non-standard extensions, to prevent semantic proliferation.

6.6. Error Types

This section lists the error types defined by this extension. All errors use the ACME error URN namespace prefix "urn:ietf:params:acme:error:" (per [RFC8555] § 6.7):

Error URN	Trigger Condition
urn:ietf:params:acme:error:badIdpToken	Invalid token signature, reused jti or mismatched bound_to_order
urn:ietf:params:acme:error:rejectedIdpId	Mismatched identifier or iss/aud claims
urn:ietf:params:acme:error:idpTimeout	Expired token
urn:ietf:params:acme:error: trustDomainRestrictionViolation	CA policy requires issuance of a certificate with the trust_domain_restriction extension, but the acmeIdpToken content fails to satisfy population requirements for this extension

Figure 10: Error Types

6.7. IdP Operational Certificate Format and Interoperability

The IdP operational certificate is the core mechanism for formalizing an IdP as a CA-authorized Registration Authority (RA).

6.7.1. Certificate Format Requirements

An IdP operational certificate *SHOULD* follow the X.509 v3 certificate format specification ([RFC5280]) and shall meet the following specific requirements:

* Certificate type: X.509 v3 end-entity certificate.

* Extended Key Usage (EKU): *MUST* include id-kp-acmeIdpTokenSigning (see IANA registration in Section8).

* Key Usage: *MUST* include digitalSignature.

* Basic Constraints: The CA flag *MUST* be FALSE.

* Subject DN: *SHOULD* at least contain CN = IdP domain name or organization identifier; OV-level certificates *SHOULD* additionally include the O and C fields.

* Subject Alternative Name (SAN): *SHOULD* contain the dNSName of the IdP validation endpoint, matching the domain in the idp_url field.

* CRL Distribution Points: *MAY* be included.

* Validity period: Controlled by CA policy; recommended not to exceed one year, with support for ACME renewal.

6.7.2. Interoperability with Existing IdPs

For existing deployed IdP systems, integration can be achieved via the following approaches:

* LDAP/AD-based IdPs: Deploy an ACME-IDP client alongside existing AD domain controllers or LDAP servers. Use enterprise-native LDAP/AD connectors for user authentication, while the ACME-IDP client solely handles operational certificate management and acmeIdpToken issuance.

* SAML/OIDC-based IdPs (e.g., Shibboleth, Keycloak): The ACME-IDP client provides protocol translation from SAML/OIDC to acmeIdpToken (JWT), enabling existing IdPs to participate in the ICV framework without modifying their core authentication logic.

7. Security Considerations

7.1. Trust Anchor Levels and Threat Model

The CA is the issuer of the final certificate and holds a higher trust-anchor level than the IdP. The IdP assists the CA in proving the binding between an identity and a public key. When a CA adopts the standalone ICV (trust-idp-PoP) branch, the CA fully relies on the IdP's validation results for identity-public-key PoP — the security boundary of this mode is tied to the security strength of the IdP's authentication protocol.

Major threats and mitigations are as follows:

* Compromise of the operational certificate private key (Mode1): If the IdP operational certificate private key is compromised, an attacker may forge tokens. The CA mitigates this risk via certificate validity periods, revocation mechanisms, and key-strength requirements. Compromise of OV-level operational certificates has more severe consequences; IdPs are recommended to use HSMs and shorten certificate validity periods.

* CA issuance review: The CA *MUST* verify the IdP identity, with policies defined in the CPS.

* Root certificate management (Mode2): Root certificate exchange *MUST* occur over secure channels. Compromise or mis-issuance of either party's root certificate impacts the entire cross-trust system.

* Forged identity control verification following IdP compromise: Defense-in-depth can be provided via "pk-01".

* Auditing of internal identity authentication (Mode3): When the CA itself acts as the IdP, the CA's identity authentication logs and certificate issuance decisions *SHOULD* be covered by audits defined in the CPS.

7.2. Layered PoP Verification

Within the ICV framework, PoP verification can occur at two layers:

* idp-PoP: When an IdP's authentication method uses public-key authentication, the IdP verifies during authentication that the applicant holds the private key bound to the identity.

* acme-PoP: The CA directly verifies via the "pk-01" challenge that the applicant holds the private key corresponding to the certificate public key, entirely independent of the IdP.

Based on the type of the IdP's authentication method and its own policies, the CA decides whether to trust idp-PoP and whether to additionally require acme-PoP.

7.3. Certificate Lifecycle and Authentication Strength

The certificate lifecycle *SHOULD* match the PoP security strength of the IdP authentication method:

* Low-entropy credentials (e.g., OPAQUE passphrases): Issue only short-lived certificates (hours to days), and *SHOULD* require additional acme-PoP.

* Hardware-bound two-factor authentication (e.g., WebAuthn): Longer-lived certificates (months to one year) may be issued, with acme-PoP optional.

* Existing PKI certificate chains: PoP security strength matches that of the source CA; long-lived certificates are recommended.

* Internal identity authentication (e.g., LDAP, AD integration): Security strength depends on the internal authentication protocol strength of the CA.

Specific policies are defined in the CA's CP/CPS.

7.4. Timeliness

The token validity period *MUST NOT* exceed 5 minutes, and the jti shall be single-use only. The exact duration of the jti replay window is chosen by the CA implementation, but it *MUST* cover the maximum possible token validity period.

Even in the extreme scenario where an attacker bypasses the jti replay window, the bound_to_order claim (see Sections 6.3 and 7.9) restricts the token to the "newOrder" for which it was generated, preventing attackers from redirecting it to other orders. This provides defense-in-depth for timeliness protection.

7.5. Privacy

The IdP learns the user identity during authentication. The sub claim *SHOULD* use pairwise pseudonyms (distinct values per aud), and the AS shall use sub only for the minimum scope required for certificate issuance. In cross-domain deployments where the CA and IdP belong to different trust domains, cross-domain transmission of identity information introduces additional privacy risks. Under Mode3 where the CA acts as the IdP itself, identity information is

transmitted within a single trust domain, substantially reducing privacy risks compared with cross-domain deployments.

Using distinct pairwise pseudonyms for different aud values mitigates correlation attacks (linking the same user across multiple IdPs via a shared sub): a sub value for one aud *SHOULD NOT* be reused across other aud values, so as to preserve unlinkability among relying parties.

The idpAssertedAttributes field within the trust_domain_restriction extension persists IdP-asserted attributes into the X.509 certificate, exposing them to all relying-party processing paths (including CT logs). The CA *SHOULD* assess privacy consequences when selecting which attributes to embed:

* Embed only attributes that are operationally necessary and consented to by the applicant;

* Attributes involving sensitive categories (e.g., health, religion, internal organizational roles) *SHOULD* be excluded by default unless explicitly required by the relying-party context;

* Prefer the “pairwise pseudonym + role” pattern for expressing access rights over “real-name + detailed attributes” when feasible.

Before making authorization decisions using idpAssertedAttributes, relying parties *SHOULD* constrain the set of acceptable attribute types in local policy (see Step4 in Section7.10) to prevent privilege escalation triggered by unexpected attributes.

7.6. Comparison with the SSO/OAuth Security Model

The three-party interaction model of "idp-01" is structurally similar to SSO/OAuth, yet fundamentally different in trust-anchor architecture: SSO/OAuth adopts a single-trust-anchor structure (the IdP is the sole trust anchor), whereas "idp-01" features a dual-trust-anchor structure (the CA ranks higher than the IdP, with the CA granting active trust via operational certificates or pre-provisioned root certificates).

Dimension	SSO/OAuth	idp-01
Trust Anchor Structure	Single trust anchor (IdP only)	Dual trust anchor (CA superior to IdP)
Token Semantics	Authorizes access to online resources	One-time credential triggering long-term certificate issuance
Security Impact of Output	Authorization ends on token invalidation	Certificate persists, impact lasts months/years
PoP Verification	Single-layer performed by IdP	Layered verification: idp-PoP + optional acme-PoP
Certificate Lifecycle Control	N/A	Governed by authentication strength
Trust Anchor Establishment	Passively retrieved via OIDC Discovery	Pre-provisioned via operational or root certificates, with active trust granted by CA

Figure 11: Comparison of Security Models between SSO/OAuth and idp-01

7.7. Comparison of Security Models between SSO/OAuth and idp-01

Although IdP operational certificates are formally similar to subordinate CA certificates (both issued by a CA and used to authorize external entities), they differ fundamentally in security role, authorization scope, and trust-transfer mechanism.

* Role: A subordinate CA acts as a "certificate issuer", with X.509 certificates as its authorization output. An IdP (holding an operational certificate) acts as an "identity verifier", with one-time JWT tokens as its authorization output.

* Authorization scope: Authorization granted to a subordinate CA is permanent and broad. Authorization granted to an IdP operational certificate is temporary and fine-grained (each token includes idpIdentifier, idp_method, and one-use constraints via jti).

* Compromise impact: Compromise of a subordinate CA private key equals partial leakage of root CA authority, allowing attackers to issue arbitrary end-entity certificates. If an IdP operational

certificate private key is compromised, attackers may only forge JWT tokens and cannot directly issue X.509 certificates. Furthermore, the CA performs secondary validation during each token check (combining identity identifiers and challenge nonces from orders) and can immediately revoke the operational certificate to block authorization.

Key Security Recommendations:

* When issuing an IdP operational certificate, the CA *MUST* explicitly define its intended usage in the certificate: the cA flag in Basic Constraints *MUST* be FALSE, Key Usage *MUST* include digitalSignature, and Extended Key Usage (EKU) *MUST* include id-kp-acmeIdpTokenSigning. The IdP operational certificate *SHOULD NOT* include any EKUs intended for certificate issuance (e.g., id-kp-serverAuth or anyExtendedKeyUsage).

* Relying parties *SHOULD NOT* treat IdP operational certificates as subordinate CA certificates; they *SHOULD* only accept them for verifying the JWT signature of acmeIdpToken.

* CA operators *SHOULD* clearly specify in the CPS the differences between IdP operational certificates and subordinate CA certificates in terms of authorization scope, audit requirements, and security incident response procedures.

The table below summarizes the differences in key security attributes:

Attribute	IdP Operational Certificate (Mode1)	Subordinate CA Certificate
Core Function	Issue identity-verification JWT tokens	Issue X.509 certificates
Basic Constraints	cA=FALSE	cA=TRUE
Key EKU	id-kp-acmeIdpToken-Signing	serverAuth, clientAuth or anyExtendedKeyUsageE
Authorized Output	JWT (short-lived, one-time)	X.509 certificate (long-lived, chain-transferable)
May Issue End-Entity Certificates?	No	Yes
Private Key Compromise Impact	Attackers may forge identity tokens but cannot issue certificates; limited impact scope	Attackers may issue arbitrary trusted certificates; broad impact and high recovery cost (CRL/OCSP)
Typical Lifespan	Recommended 1year with automated ACME renewal	Typically 1-10years with manual management

Figure 12: Key Security Attribute Differences Between IdP Operational Certificates and Subordinate CA Certificates

7.8. Comparison with Stand-Alone Use of RFC9447

There is a clear boundary between the validation objectives of RFC9447 [RFC9447] and "idp-01":

* RFC9447 validates authorization of a specific identifier by an external authority — for example, a telephone carrier’ s right to allocate a given number.

* "idp-01" validates binding of an applicant’ s identity to a public

key by an IdP — focusing on _who the applicant is_ and _which key they hold_, rather than the source of authorization for a given identifier.

The two may be combined within a single order: first validate the identity-public-key binding via "idp-01", then validate identifier authorization via tkauth-01.

Dimension	RFC 9447 (tkauth-01)	idp-01
Validation Target	Identifier authorization source	Identity-public-key binding
Trust Anchor Source	Identifier-specific external authority (e.g., TNAuthList)	Operational certificates, root certificates, or the CA itself (corresponding to the three deployment modes)
Public Key Handling	Not applicable	Works with pk-01; supports idp-PoP / acme-PoP
Deployment Mode	Single (external authority)	Three deployment modes
Relationship with PoP	Orthogonal	Directly associated via the synergy matrix (see Section 3.5)

Figure 13: Comparison between RFC 9447 and idp-01

The semantics of tkauth-01 are that possession of a token issued by an external authority constitutes possession of the corresponding identifier. It does not natively support standardized authorization for IdP identities (operational certificates), branched public-key sources (CSR / idp-PoP / pk-01), or deployment models where the CA itself acts as the authority. This document therefore defines the ICV framework as a standalone challenge type, independent of RFC9447 in terms of validation objectives and deployment modes.

7.9. Order Binding

"pk-01" [I-D.geng-acme-public-key] cryptographically binds proof-of-possession of the private key to all bytes of the current newOrder by including SHA-256(raw_newOrder) in the PoP input. To align with the security model of "pk-01", this document requires the IdP to carry back the same hash in the form of the bound_to_order claim when issuing an acmeIdpToken (see Section6.3), which the CA verifies in Step4 of § 6.4.

Multiple defensive layers are provided:

* bound_to_order claim: Cryptographically binds the token to all bytes of the "newOrder". Any modification to "newOrder" fields

(including the identifiers array) will cause token validation to fail;

* One-time consumption of idpIdentifier: The server generates an idpIdentifier with no less than 128 bits of entropy for each challenge and marks it as consumed immediately after the challenge reaches a final state, preventing repeated submission of the same token;

* One-time use of jti + exp 5minutes: Restricts token lifetime, serving as defense-in-depth in addition to bound_to_order and one-time consumption of idpIdentifier.

The specific protocol binding for how clients transmit the raw_newOrder hash to the IdP is defined per idp_method; a typical implementation submits the hash as an additional parameter to the IdP's authentication endpoint. The bound_to_order claim supports two categories of IdP implementations:

* ACME-context-aware IdPs: Populate the claim directly with the hash provided by the client;

* Non-ACME-context-aware IdPs: The CA may accept weak binding (relying solely on one-time jti + idpIdentifier) under policy, but *SHOULD* explicitly document the security consequences of such weakening in its CPS.

7.10. Relying-Party Validation Logic for Trust-Domain-Restricted Certificates

When a relying party receives a certificate containing the trust_domain_restriction extension, it *SHOULD* perform the following additional checks in addition to regular PKI path validation:

1. Whether the IdP indicated by idpIdentifier in the certificate remains in the set trusted by local policy. If trust in the IdP has been revoked by local policy, the relying party *SHOULD* reject the certificate.

2. For certificates issued under Mode1, the relying party *MAY* check the revocation status of the underlying IdP operational certificate via OCSP/CRL. If the operational certificate has been revoked, the relying party *SHOULD* reject downstream identity certificates issued while the IdP was active.

3. If the certificate contains an idpPolicyURI, the relying party *MAY* retrieve the IdP's current policy statement per local policy and verify whether attributes asserted in the certificate remain within the IdP's current authorization scope.

4. If the certificate contains idpAssertedAttributes, the relying party *MUST* apply a whitelist filter on attribute types according to local policy before making authorization decisions using these attributes. Any attribute type not in the locally permitted set *SHOULD* be ignored.

The extended checks defined in this section *do not replace* standard PKI path validation; the two operate cumulatively.

7.10.1. Handling of Critical and Non-Critical Extensions

This document specifies that the trust_domain_restriction extension is issued as non-critical by default (see § 6.5.2). This choice has two security implications:

* Relying parties that do not recognize the extension will ignore it and treat the certificate as a regular one — this is the compatibility trade-off with the existing TLS/S/MIME ecosystem. By choosing the non-critical setting, the CA accepts the risk that relying parties may skip trust-domain checks, and *SHOULD* mitigate this risk via other controls (e.g., restricting certificate validity periods, monitoring CT logs, limiting the EKU set).

* Relying parties that recognize the extension *SHOULD* validate it in accordance with the rules in this section.

When a CA issues this extension as critical, relying parties that do not support the extension will reject the certificate, providing fail-closed security guarantees; however, the CA *MUST* verify that the target relying-party ecosystem supports this extension.

7.10.2. Relationship with Defense-in-Depth

Even if a relying party does not recognize the trust_domain_restriction extension, the device-level private-key possession proof (acme-PoP) provided by the "pk-01" synergy branch still serves as an independent key-binding guarantee: an attacker who compromises the IdP operational certificate private key cannot replace the certificate public key. CAs shall take this into account when evaluating which scenarios require combined ICV + PoP deployment.

8. IANA Considerations

This document requests IANA to complete the following registrations:

* ACME Identifier Type: Register idp.

* ACME Validation Method: Register idp-01.

* EKU OID: Register id-kp-acmeIdpTokenSigning (OID to be assigned).

* X.509 PKIX Private Extension OID: Register id-pe-acme-idp-trust-domain (OID to be assigned, expected under the PKIX private extension arc; the exact value shall be determined by IANA in the SMI Security for PKIX Module Identifier registry).

* ACME Message Fields: Register idpIdentifier, idp_url, idp_method, deployment_mode, idp_cert_fingerprint, acmeIdpToken.

* acmeIdpToken JWT Claim Names: Register bound_to_order, idpIdentifier, idp_method, confirmed_public_key. The semantics of the bound_to_order claim are defined in Sections 6.3, 6.4, and 7.9.

* Error Types (Full URNs):

  - urn:ietf:params:acme:error:badIdpToken

  - urn:ietf:params:acme:error:rejectedIdpId

  - urn:ietf:params:acme:error:idpTimeout

  - urn:ietf:params:acme:error:trustDomainRestrictionViolation

* IdP Authentication Method Registry: Initial values are pkic, opaque, webauthn, internal, device-puf, device-tpm, device-vin, device-sn. New entries shall be added under the Specification Required policy ([RFC8126]).

* ACME idpAssertedAttributes Attribute Type Registry: For the AssertedAttribute.type field of the trust_domain_restriction extension. No entries are pre-populated in the initial version; new entries shall be added under the Specification Required policy, and each entry *MUST* provide the attribute OID, semantic description, character set, and maximum length. CAs *MUST NOT* write attributes of unregistered types into certificates.

9. Appendix A. Same-Domain Trust — Example of the IdP Operational Certificate Mode (Mode1, Stand-Alone ICV + CSR)

This appendix addresses the following problem: When an organization has deployed an identity management system (e.g., enterprise LDAP, content-platform user system) that is not a CA, how to issue identity certificates automatically for members of the organization in a standardized manner.

This scenario falls under a trust chain type referred to as same-domain trust, applicable to Mode1 (the IdP operational certificate model) and the stand-alone ICV + CSR branch. The involved parties are the CA, the IdP (the organization's identity management system, not a CA), and end users.

9.1. Phase 1: IdP Requests an Operational Certificate

The IdP automatically completes operational certificate enrollment and lifecycle management via an ACME-IDP client. Depending on the trust level required by the CA, the operational certificate may be of DV or OV level. The automated enrollment flow is illustrated below using the OV level as an example:

1. An administrator of the IdP organization first completes organization validation (pre-validation for OV) on the CA's enterprise portal. Upon CA approval, EAB credentials are generated for this IdP.

2. The IdP deploys the ACME-IDP client, configures the EAB credentials, and creates an ACME account.

3. Submit a "newOrder" declaring a DNS identifier with profile: "idp-op-cert-ov" (or "idp-op-cert-dv").

4. The CA returns required challenges according to the profile requirements; these are automatically completed by the ACME-IDP client.

5. Submit a CSR, and the CA issues the operational certificate (with EKU including id-kp-acmeIdpTokenSigning).

For IdPs requiring only a DV-level operational certificate, the pre-validation in Step1 may be skipped.

9.2. Phase 2: User Requests an Identity Certificate

1. The user submits a "newOrder" using an ACME-ICV client on the device, declaring an idp identifier (e.g., an email address) and omitting the "pk" identifier (the certificate public key will be submitted via CSR).

2. The CA returns an "idp-01" challenge with deployment_mode: "idp-op-cert".

3. The ACME-ICV client interacts with the IdP to complete identity authentication, and the IdP issues an acmeIdpToken using the private key of its operational certificate.

4. The ACME-ICV client submits the token; upon successful validation by the CA, the challenge status becomes valid.

5. The ACME-ICV client submits a standard PKCS#10 CSR during the finalize phase. The CA extracts the certificate public key from the CSR and issues an identity certificate (e.g., an S/MIME certificate).

The certificate public key may differ from the user’ s identity public key maintained on the IdP side — the IdP verifies control over the identity, while the certificate public key is independently generated by the user and submitted via the CSR, preserving the user’ s flexibility to use distinct key pairs for different purposes.

10. Appendix B. Same-Domain Trust — Example of CA-Integrated IdP Mode (Mode3)

This appendix addresses the following problem: When a CA itself has built-in user identity management capabilities, how to implement a fully automated closed-loop workflow of “identity registration → identity authentication → certificate issuance” on a unified platform, avoiding separate deployment and management of two independent systems for the CA and IdP.

This scenario falls under a trust chain type referred to as *same-domain trust*, applicable to Mode3 (the CA-integrated IdP model) and the stand-alone ICV + CSR branch (see Section3.5). The involved parties are an internal enterprise CA (acting simultaneously as the IdP) and enterprise employees.

10.1. Prerequisites

* The enterprise has deployed an internal CA (e.g., Windows Server AD CS, HashiCorp Vault PKI, Smallstep CA, etc.) that also manages user identity registration and authentication.

* The CA has configured its built-in IdP validation endpoint (where idp_url points to the CA's own identity authentication service).

* Employees have registered their identities in the user directory managed by the CA.

* The ACME-ICV client is installed on employee devices.

10.2. Workflow

1. Create a "newOrder": An employee submits a "newOrder" to the CA via the ACME-ICV client, declaring an "idp" identifier (the employee's enterprise identity such as employee ID or email address) and omitting the "pk" identifier (the certificate public key will be submitted via CSR).

2. CA returns an "idp-01" challenge: The CA recognizes that the identifier belongs to a locally managed user and returns an "idp-01" challenge with deployment_mode: "ca-idp" and idp_method: "internal".

3. Complete identity authentication: The ACME-ICV client parses the idp_url (pointing to the CA's own validation endpoint) and completes authentication using the internal method.

4. CA issues a token: After verifying the employee's identity, the CA issues an acmeIdpToken (or directly completes challenge validation based on internal state).

5. Submit the token and CSR: The ACME-ICV client submits the token; upon successful validation by the CA, the challenge status becomes valid. The employee submits a standard PKCS#10 CSR during the finalize phase, from which the CA extracts the certificate public key and issues an identity certificate.

10.3. Scenario Value

This scenario demonstrates the simplicity and efficiency of the ICV framework under the deployment model where the CA and IdP are unified. Compared with traditional solutions requiring separate deployment and management of CA and IdP, Mode3 significantly reduces the infrastructure complexity of enterprise identity certificate management. This model aligns with industry practices of products such as Windows Server AD CS, HashiCorp Vault, and Foxpass Cloud PKI.

11. Appendix C. Same-Domain / Cross-Domain Trust — Enterprise PKI Bridging Scenario (Mode2)

This appendix addresses the problem: when two organizations each operate their own private CAs and need to automatically issue certificates for members of the other organization to enable secure communication interoperability, how to leverage existing PKI infrastructure for automated cross-organizational certificate provisioning.

This scenario falls under Type-1 or Type-2 trust chains (same-domain or cross-domain trust) and applies to Mode2 (the PKI internal-domain mutual-recognition model).

11.1. Scenario 1: Same-Domain Trust

Boundary Note: When an organization operates a single internal CA that also performs identity validation functions, the “CA acting as IdP” case actually falls under Mode3 (CA-integrated IdP), whose full workflow is provided in AppendixB. This section references it as a boundary case for Mode2 to illustrate that the cross-domain bridging approach described in this appendix degenerates into Mode3 for strictly single-CA organizations with embedded identity services.

The degenerated workflow is as follows: An employee submits a "newOrder" via the ACME-ICV client, the CA returns an "idp-01" challenge (with idp_method set to internal), and after successful validation, the employee submits the certificate public key via a CSR during the finalize phase. The entire workflow is completed within the enterprise boundary.

If an organization operates multiple internal CAs (e.g., a root CA with subordinate CAs, or private CAs for different departments), this constitutes the genuine same-domain variant of Mode2 described later in this section. Handling is identical to the cross-domain case (C.2), except that secure channels for pre-configuring trust anchors are established internally within the organization without out-of-band coordination.

11.2. Scenario 2: Cross-Domain Trust

Enterprises A and B engage in joint research and development, and Enterprise A needs to issue short-lived client certificates for Bob, an employee of Enterprise B. Both parties have exchanged root certificates via out-of-band channels.

1. Bob submits a "newOrder" via the ACME-ICV client (including enterprise identifiers; this scenario uses the combined ICV + PoP approach and also declares a "pk" identifier).

2. The CA of Enterprise A returns an "idp-01" challenge with deployment_mode: "pki-intra", where idp_url points to the CA of Enterprise B.

3. The ACME-ICV client assists Bob in signing a submission to Enterprise B's CA using the private key of his enterprise identity certificate. After validation, Enterprise B's CA issues an acmeIdpToken (with a pairwise pseudonym as the sub claim).

4. Enterprise A's CA validates the token using Enterprise B's root certificate, verifies proof-of-possession for the certificate public key per "pk-01", and finally issues the short-lived client certificate.

Multi-party federations may implement automated cross-issuance by exchanging root certificates and configuring trust anchors.

12. Appendix D. Cross-Domain Trust — E2EE Meeting Certificate Provisioning (Mode1 + Mode2)

This appendix addresses the following problem: In cross-organizational end-to-end encrypted meeting scenarios, external participants need short-lived admission certificates issued by the host CA to join group key agreement. The organizations of external participants may have pre-established PKI mutual trust with the host, or may share no trust relationship at all.

This scenario falls under Type-2 trust chains (cross-domain trust). Mode2 and Mode1 are adopted respectively depending on whether pre-configured root-certificate mutual trust exists between the host and the organizations of external participants.

12.1. Case 1: Pre-established Root-Certificate Mutual Trust Between Both Enterprises (Mode2)

The CAs of Enterprise A and Enterprise B have exchanged root certificates via out-of-band channels. Employees of Enterprise B hold identity certificates issued by Enterprise B's CA. This case adopts the combined ICV+PoP branch (see Section3.5): external participants complete the "idp-01" challenge using identity certificates issued by Enterprise B's CA, and perform "pk-01" validation via the "pk" identifier declared within the same order. The CA of Enterprise A validates the token using the pre-configured root certificate of Enterprise B and issues short-lived meeting admission certificates. The remaining steps of the workflow are consistent with AppendixC.2.

12.2. Case 2: No Pre-established Root-Certificate Mutual Trust Between Enterprises — Guest Mode (Mode1)

No pre-configured CA root-certificate trust relationship exists between Enterprise A and Enterprise B. An employee of Enterprise B is invited as a guest to temporarily join an external meeting hosted by Enterprise A. This scenario demonstrates the deployment flexibility of the ICV framework in the absence of pre-established PKI trust relationships, adopting the stand-alone ICV trust with the idp-PoP branch (see Section3.5): Enterprise A does not need to pre-trust Enterprise B's CA; it only needs to pre-register a temporary guest account on its own IdP to automatically issue a short-lived certificate valid for this meeting to the guest.

As a concrete implementation example, identity authentication between the guest and Enterprise A's IdP may use the OPAQUE protocol ([RFC9807]):

* Enterprise A's IdP supports user registration and authentication via the OPAQUE protocol.

* Temporary guest accounts created by the meeting organizer are registered using OPAQUE; the guest's temporary account credential (passphrase) is delivered over an out-of-band secure channel.

* Enterprise A's IdP specifies idp_method: opaque in the idp-01 challenge.

* The guest's ACME-ICV client performs the OPAQUE protocol with the IdP to complete authentication. During the AKE phase, the IdP confirms the guest holds the registered private key (idp-PoP) and issues an acmeIdpToken.

Key steps:

1. Pre-registration of temporary account by meeting organizer: The temporary account is valid only for the duration of this meeting and expires automatically upon meeting conclusion. The temporary login passphrase is sent to the guest via an out-of-band channel such as encrypted email.

2. Initial guest access: The guest connects to Enterprise A's meeting server using temporary account credentials via the ACME-ICV client, with access restricted only to the meeting waiting area at this stage.

3. CA returns "idp-01" challenge: Enterprise A's CA returns an "idp-01" challenge with deployment_mode: "idp-op-cert" and idp_method: "opaque".

4. Guest performs OPAQUE authentication: After verifying the guest holds the registered private key during the AKE phase, the IdP issues an acmeIdpToken (with a pairwise pseudonym as the sub claim) using its operational certificate private key.

5. Certificate issuance and meeting admission: The CA issues a short-lived admission certificate after validating the token. The guest reconnects to the meeting using the new certificate and completes E2EE group key agreement.

The entire process has no dependencies on Enterprise B's infrastructure. The ACME-ICV and ACME-IDP clients operate independently to deliver full automation for cross-domain guest scenarios. The OPAQUE implementation is particularly suited to use cases requiring high passphrase security: even if the IdP is compromised, attackers cannot obtain guest passphrases for credential-stuffing attacks.

13. Appendix E. Cross-Domain Trust — Automated C2PA Certificate Issuance (Mode1 / Mode2 / Mode3)

This appendix addresses the problem that within the C2PA (Coalition for Content Provenance and Authenticity) content-credentialing ecosystem, individual creators and organizational entities require signing certificates issued by CAs listed in the C2PA Trust List. Current certificate issuance workflows rely on manual identity vetting and cannot be automated for large-scale content production deployments.

Depending on the creator's identity type and the PKI deployment status of their affiliated organization, one of the three deployment modes may be applied to this scenario.

13.1. Requirements Description

A C2PA Manifest binds content to the signer's identity via digital signatures. Verifiers confirm content provenance by checking whether the signing certificate is issued by a CA included in the C2PA Trust List.

Applicants fall into two categories: individual creators (e.g., photographers, journalists) whose identities are typically verified on content platforms; and organizational entities (e.g., news agencies, content platforms, brand owners) that usually hold organizational certificates issued by enterprise CAs.

13.2. Applicability of the Three Deployment Modes

* Mode1 (IdP Operational Certificate Model): Applicable to individual creator scenarios. A content platform obtains an operational certificate from a C2PA-compliant CA to act as an IdP via an ACME-IDP client. Creators complete authentication with the platform IdP using an ACME-ICV client to obtain a token, after which the CA issues a C2PA certificate.

* Mode2 (PKI Intra-Domain Mutual-Recognition Model): Applicable to organizational entity scenarios. The enterprise CA root certificate is added as a trust anchor to the C2PA-compliant CA. Employees complete the "idp-01" challenge using their enterprise certificates.

* Mode3 (CA-Integrated IdP Model): Applicable to scenarios where the CA platform itself manages creator identities and certificate lifecycles.

CAs embed certificate policy OIDs in compliance with C2PA specifications. Bulk issuance can be fully automated via enterprise CA endpoints.

13.3. Deployment Recommendations

Scenario	Mode	IdP Role
News agency for its journalists	Mode2 or3	Agency enterprise CA
Photo agency for contracted photographers	Mode1 or3	Agency authentication system
Social platform for verified users	Mode1 or3	Platform identity system
Government agency for civil servants	Mode2 or3	Government PKI

Figure 14: Deployment Recommendations for C2PA Scenarios

14. Appendix F. Cross-Domain Trust — Federated Trust Scenarios (Mode2 + Federated Trust Model)

This appendix addresses the problem of how to leverage federated trust infrastructure to reduce coordination costs when multiple independent organizations require automated cross-organizational identity certificate provisioning via a multilateral trust framework, rather than signing bilateral mutual-recognition agreements one-by-one.

This scenario falls under Type-3 trust chains (federated trust). Built upon Mode2 (the PKI intra-domain mutual-recognition model), it incorporates a federated trust model and adopts the combined ICV+PoP branch to guarantee ownership of the certificate public key during cross-domain certificate issuance (see Section3.5).

14.1. Scenario Description

Multiple research institutions form a joint consortium via the eduGAIN federation. eduGAIN is an international service connecting global research communities and higher-education identity federations, covering more than 80 participating federations and linking over 8000 identity and service providers. The consortium requires CAs of all member institutions to issue short-lived access certificates for researchers from other member institutions.

The federated trust chain is established under OpenID Federation1.0. Layered collaboration is formed between "idp-01" (the validation action layer) and OpenID Federation (the trust model layer): the federated trust chain addresses “how to trust an IdP”, while "idp-01" addresses “how to perform identity-to-public-key binding validation”.

14.2. Workflow

1. Researcher Alice (from Institution A) submits a "newOrder" to the CA of Institution B via the ACME-ICV client.

2. The CA of Institution B verifies the identity of Institution A’s IdP through the OpenID Federation trust chain and retrieves the token-signing public key.

3. The CA returns an "idp-01" challenge with idp_url pointing to Institution A’s IdP.

4. The ACME-ICV client assists Alice in completing authentication to obtain a token (with a pairwise pseudonym as the sub claim).

5. The CA of Institution B validates the token and issues a short-lived access certificate.

15. Appendix G. Stand-Alone ICV Trust with idp-PoP — Temporary Certificates for Public Device Scenarios

This appendix addresses the problem of how users can securely and conveniently obtain short-lived identity certificates for untrusted public devices (e.g., conference room terminals, shared workstations) when participating in secure communications, without pre-provisioning any key material or certificates, while minimizing required user interactions.

This appendix presents a representative application of the stand-alone ICV (CA-trusted idp-PoP) branch. It falls under Type-1 trust chains (same-domain trust) and adopts Mode1 (the IdP operational certificate model).

15.1. Participants and Prerequisites

Participants: a CA, an IdP (holding an operational certificate via an ACME-IDP client), an end-user, and a public device (with an ACME-ICV client installed).

Prerequisites: The user has registered an OPAQUE account with the IdP and securely deposited an ephemeral public-private key pair encrypted at the IdP. The IdP has obtained an operational certificate from the CA, and the CA trusts the IdP's authentication results for OPAQUE accounts as well as the idp-PoP proof.

15.2. Workflow

1. User logs into the public device and retrieves the ephemeral key pair: The user enters their OPAQUE account passphrase via the ACME-ICV client on the public device. The client performs the OPAQUE protocol with the IdP to complete authentication, securely retrieves the encrypted ephemeral public-private key pair deposited with the IdP, and decrypts it using the passphrase to obtain the ephemeral private key.

2. Create a "newOrder": The client submits a "newOrder" to the CA, declaring only the idp identifier (set to the user's OPAQUE identity) and omitting the "pk" identifier.

3. CA returns an "idp-01" challenge with deployment_mode: "idp-op-cert" and idp_method: "opaque".

4. Complete identity authentication: The client forwards challenge information to the IdP. The IdP issues an acmeIdpToken using the private key of its operational certificate. The token contains a confirmed_public_key claim whose value is the base64url-encoded SPKI of the ephemeral public key — a public key known to the IdP from OPAQUE registration, for which the IdP verifies user possession of the corresponding private key during this authentication.

5. Submit the token: The client POSTs the acmeIdpToken to the challenge URL.

6. CA validates and issues the certificate: The CA validates the token signature and content. As the CA is configured to trust OPAQUE-based idp-PoP and no "pk" identifier is present in the order, the CA applies the stand-alone ICV (idp-PoP-trusted) branch. It directly extracts the ephemeral public key from confirmed_public_key as the certificate public key, issues a short-lived device certificate, and sets the subject to the pairwise pseudonym contained in the token.

7. Use the temporary certificate: The public device now holds the ephemeral private key and its corresponding short-lived device certificate, enabling access to the enterprise E2EE video conferencing system. Upon meeting conclusion, the certificate expires naturally, and the ephemeral key pair on the public device can be securely erased.

The entire workflow requires the user to remember only a single passphrase.

16. Appendix H. Enterprise Intranet Device Proxy Scenario (ACME-ICV Proxy Mode)

This appendix addresses the problem of enabling automated identity certificate issuance using the ICV framework when enterprise endpoint devices reside on an intranet behind a firewall and cannot directly access external public CAs. This represents a typical scenario where an IdP acting as a Registration Authority (RA) fosters new business models within the PKI/CA ecosystem — the IdP may serve as both an identity-verification gateway and a certificate enrollment proxy.

16.1. Scenario Description

An enterprise internal network hosts numerous devices (e.g., workstations, servers, IoT devices) located behind firewalls or NAT gateways. The enterprise IdP acts as an identity management gateway deployed at the enterprise network perimeter (e.g., in the DMZ), with connectivity to both the intranet and the public internet.

The IdP deploys an ACME-ICV proxy client that serves as a unified certificate enrollment representative for enterprise intranet devices. The proxy client itself has obtained RA authorization from the CA via an IdP operational certificate. Intranet devices initiate local certificate requests to the IdP proxy. After verifying device identities, the IdP proxy completes the idp-01 challenge with the external CA on behalf of the devices.

16.2. Proxy-Mode Architecture

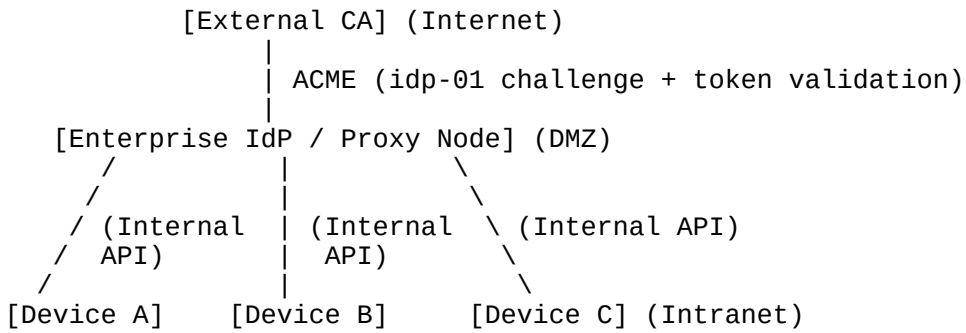
Involved components:

* External CA: A public or externally trusted CA.

* Enterprise IdP (proxy node): Deployed in the DMZ, holding an operational certificate issued by the CA, and running an ACME-ICV proxy client.

* Intranet devices: End-user devices within the enterprise intranet that cannot directly access the external CA.

The architectural relationships are as follows:



16.3. Proxy Workflow

Taking an intranet IoT sensor (Device X) requesting an identity certificate as an example:

1. Device X initiates a local request: It generates a public-private key pair, constructs a local certificate request containing a device identifier (e.g., serial number SN:XYZ123) and the public-key SPKI, and sends the request to the IdP proxy node via an internal API.

2. The IdP proxy verifies the device identity: Validation is performed in accordance with enterprise policies (based on pre-shared keys, MAC-address binding, or device registration records in the enterprise CMDB).

3. The IdP proxy executes the external ACME-ICV workflow:

   a. As an ACME-ICV client, it submits a "newOrder" to the external CA, declaring the idp identifier set to DeviceX's pairwise pseudonym.

   b. The CA returns an "idp-01" challenge with deployment_mode: "idp-op-cert", where idp_method is selected according to the device type (e.g., device-sn).

   c. The IdP proxy encapsulates DeviceX's authentication result into an acmeIdpToken and signs it using the private key of its operational certificate. The token contains a confirmed_public_key claim whose value is the base64url-encoded SPKI of DeviceX's public key.

d. The IdP proxy POSTs the token to the CA challenge URL.

4. The CA validates and issues the certificate: The CA validates the acmeIdpToken (chaining back to the IdP operational certificate) and trusts the IdP's device authentication result. Since no "pk" identifier is declared in the order, the CA applies the stand-alone ICV (idp-PoP-trusted) branch, directly extracts DeviceX's public key from confirmed_public_key, and issues an end-entity certificate.

5. Certificate delivered to the intranet device: The CA returns the certificate to the IdP proxy, which forwards it to DeviceX.

16.4. Security Considerations and Limitations

* The IdP proxy must implement strong authentication mechanisms to prevent adversaries from impersonating intranet devices and requesting certificates from the proxy. Mutual authentication using pre-shared client certificates or API keys is recommended between intranet devices and the proxy.

* The private key for the IdP proxy's operational certificate *MUST* be strictly protected (use of an HSM or secrets management service is recommended).

* The proxy mode introduces an additional trust boundary: the CA trusts the device authentication results from the IdP proxy, while the IdP proxy trusts identity claims from intranet devices. Enterprises *SHOULD* enforce rigorous intranet device registration and validation policies, and regularly audit proxy logs.

* This mode is suitable for enterprise environments requiring centralized certificate lifecycle management, especially for large-scale IoT/OT deployments where devices cannot directly access the public internet.

17. Appendix I. Cross-Domain Trust — sigstore Code-Signing Scenario (Mode1 + OIDC-Bridged IdP)

This appendix addresses the following problem: In modern software supply chains, manual issuance, long-term private-key custody, and revocation workflows for code-signing certificates severely restrict automated deployment. Most developers already have verifiable identities established via public OIDC providers such as GitHub, Google, and Microsoft, yet lack a standardized path to convert such identities into X.509 certificates acceptable within the code-signing ecosystem.

This scenario falls under Type-2 trust chains (cross-domain trust) and adopts the OIDC-bridged variant of Mode1 (the IdP operational certificate model). This appendix aligns with sigstore [SIGSTORE]'s "keyless signing" workflow in terms of motivation and role partitioning; the detailed interaction flow is consistent with Section5.1 and AppendixD.2 and is therefore not elaborated further here.

17.1. Motivation

Pain points of traditional code-signing deployments:

* Certificate issuance relies on manual organizational validation and cannot be self-serviced within CI/CD pipelines;

* Signing private keys require long-term secure storage (HSM/smart cards), imposing heavy management overhead;

* Developer identities across projects and organizations are difficult to reuse in traditional PKI.

Industry practice with sigstore has demonstrated that converting developers' OIDC identities into short-lived code-signing certificates, combined with transparency logs recording certificate validity at signing time, enables a "zero long-term key management" code-signing workflow for large-scale software supply chain scenarios. This approach has been adopted by major ecosystems including Kubernetes, npm, and PyPI. The goal of this appendix is to incorporate this industry practice into the ACME standardization path, so that automated issuance of code-signing certificates can be achieved under the unified ICV framework, rather than relying on proprietary interfaces of specific implementations.

17.2. Alignment with This Draft

The existing sigstore architecture naturally aligns with the ICV framework of this draft in terms of roles and protocol interfaces:

* Fulcio (the CA issuing short-lived code-signing certificates) = ACME Server (CA);

* Public OIDC providers (GitHub, Google, Microsoft, etc.) = third-party IdPs;

* Signing tools such as cosign = ACME-ICV clients;

* Developer OIDC identities (the combination of iss + sub) map to the value field of the "idp" identifier in URI form.

Since public OIDC providers do not hold operational certificates issued by the CA (lacking the id-kp-acmeIdpTokenSigning EKU), this scenario adopts the OIDC-bridged IdP deployment pattern: the CA itself or a trusted operator deploys an ACME-IDP client together with an OIDC bridge. The bridge holds an operational certificate issued by the CA and re-signs using an acmeIdpToken after validating upstream OIDC ID Tokens. The recommended value for idp_method is "oidc" (pending IANA registration), and the bound_to_order claim is carried via the standard OIDC nonce parameter (see Section7.9). This pattern is consistent with the “SAML/OIDC-based IdP integration” described in Section6.7.2.

The certificate lifecycle is recommended to follow sigstore’s short-lived certificate practice (ranging from minutes to hours) and be deployed in conjunction with upstream transparency logs (e.g., sigstore Rekor). Transparency logs are a complementary mechanism outside the scope of this draft and are orthogonal to the trust_domain_restriction extension: the former addresses evidence preservation for “certificate validity at signing time”, while the latter addresses runtime recognition of the IdP trust domain by relying parties.

17.3. Simplified Workflow

Participants: a CA, an OIDC-bridged IdP (deployed by the CA itself or a trusted operator, holding an operational certificate issued by the CA), public OIDC providers (GitHub, Google, Microsoft, etc.), developers, and their CI pipelines. This scenario applies the stand-alone ICV + CSR branch (see Section3.5).

1. CI-triggered signing: A pipeline (e.g., GitHub Actions, GitLab CI) initiates a signing task. An ACME-ICV client (a cosign-style signing tool) retrieves the upstream OIDC ID Token from the runtime environment.

2. Create a "newOrder": The client generates an ephemeral signing key pair and submits a "newOrder" to the CA, declaring the "idp" identifier with a value formatted as a URI: oidc:<issuer>#<subject>.

3. CA returns a challenge: with deployment_mode: "idp-op-cert", idp_method: "oidc", and idp_url pointing to the OIDC-bridged IdP endpoint.

4. Bridged IdP validates identity: The client submits the ID Token to the bridged IdP, where the nonce field carries base64url(SHA-256(raw_newOrder)). After validating the upstream token’s signature and the iss/aud/exp/nonce claims via the OIDC

JWKS, the bridged IdP issues an acmeIdpToken using the private key of its operational certificate, with a pairwise pseudonym recommended for the sub claim.

5. Token submission and certificate issuance: The client POSTs the acmeIdpToken to the challenge URL. Upon successful validation by the CA, the client submits a PKCS#10 CSR during the finalize phase, and the CA issues a short-lived code-signing certificate.

6. (Optional) Transparency log registration: The CA or client submits the certificate and signing event to an external transparency log (e.g., sigstore Rekor) to preserve evidence of certificate validity at signing time.

The entire workflow is transparent to developers: developers only need an OIDC identity and do not manage any long-term signing private keys. Ephemeral key pairs are generated per CI job and naturally expire alongside the certificate.

17.4. Value

This appendix demonstrates the capability of the ICV framework to build a standardized bridge between public OIDC identities and enterprise-grade PKI: developers do not need additional credentials, upstream OIDC IdPs require no modifications to their authentication protocols, and CAs retain final issuance authority. Aligned with real-world deployment experience of sigstore, this mode provides a standardized path for integrating “zero long-term key management” code-signing workflows into the ACME ecosystem, and together with Appendix E (C2PA), achieves full coverage of both content-signing and code-signing scenarios.

18. Acknowledgements

The motivation and design of this document have benefited from exploratory work by numerous experts in the ACME Working Group on non-DCV authentication approaches over time.

The authors thank [I-D.biggs-acme-sso] for pioneering the concept of third-party identity provider participation in ACME validation. Thanks are extended to [RFC8823] for contributions to defining email identifiers and enabling automated issuance of end-user S/MIME certificates. We acknowledge [I-D.ietf-acme-client] for first identifying automation requirements for end-user, device, and code-signing certificates. Appreciation goes to [I-D.ietf-acme-telephone] for exploring the applicability of non-domain identifiers (such as telephone numbers) validated via external authoritative entities. We thank

[I-D.ietf-acme-device-attest] for leveraging hardware roots of trust to verify device identities, providing key references for device authentication scenarios. Finally, gratitude is given to [RFC9447] and [I-D.ietf-acme-openid-federation] for serving as core references in the design of the generic authoritative token challenge framework and multi-layered trust-chain mechanisms.

The authors also thank all members of the ACME Working Group for their valuable feedback during discussions on ACME identity-related proposals.

19. References

19.1. Normative References

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8823] Melnikov, A., "Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates", RFC 8823, DOI 10.17487/RFC8823, April 2021, <https://www.rfc-editor.org/info/rfc8823>.

[RFC9447] Peterson, J., Barnes, M., Hancock, D., and C. Wendt, "Automated Certificate Management Environment (ACME) Challenges Using an Authority Token", RFC 9447, DOI 10.17487/RFC9447, September 2023, <https://www.rfc-editor.org/info/rfc9447>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <https://www.rfc-editor.org/info/rfc5480>.

[RFC9807] Bourdrez, D., Krawczyk, H., Lewi, K., and C. A. Wood, "The OPAQUE Augmented Password-Authenticated Key Exchange (aPAKE) Protocol", RFC 9807, DOI 10.17487/RFC9807, July 2025, <https://www.rfc-editor.org/info/rfc9807>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

19.2. Informative References

[I-D.geng-acme-public-key]
Geng, F., Wu, P., Xia, L., and C. , "Automated Certificate Management Environment (ACME) Extension for Public Key Challenges", Work in Progress, Internet-Draft, draft-geng-acme-public-key-06, 17 April 2026, <https://datatracker.ietf.org/doc/html/draft-geng-acme-public-key-06>.

[I-D.ietf-acme-telephone]
Peterson, J. and R. Barnes, "ACME Identifiers and Challenges for Telephone Numbers", Work in Progress, Internet-Draft, draft-ietf-acme-telephone-01, 30 October 2017, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-telephone-01>.

[I-D.biggs-acme-sso]
Biggs, A., Barnes, R., and R. Moynihan, "Automated Certificate Management Environment (ACME) Extension for Single Sign On Challenges", Work in Progress, Internet-Draft, draft-biggs-acme-sso-01, 8 April 2021, <https://datatracker.ietf.org/doc/html/draft-biggs-acme-sso-01>.

[I-D.ietf-acme-client]
Moriarty, K., "ACME End User Client and Code Signing Certificates", Work in Progress, Internet-Draft, draft-

ietf-acme-client-14, 11 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-client-14>.

[I-D.ietf-acme-openid-federation]
De Marco, G., Pitman, B., Geoghegan, T., Cook, D., and J. Jones, "Automatic Certificate Management Environment (ACME) with OpenID Federation 1.0", Work in Progress, Internet-Draft, draft-ietf-acme-openid-federation-00, 16 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-openid-federation-00>.

[I-D.ietf-acme-device-attest]
Weeks, B., Mallaya, G., Rajala, S., Bonnell, C., and R. Hurst, "Automated Certificate Management Environment (ACME) Device Attestation Extension", Work in Progress, Internet-Draft, draft-ietf-acme-device-attest-04, 5 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-device-attest-04>.

[I-D.ietf-acme-profiles]
Gable, A., "Automated Certificate Management Environment (ACME) Profiles Extension", Work in Progress, Internet-Draft, draft-ietf-acme-profiles-01, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-profiles-01>.

[SIGSTORE] Sigstore Project, "sigstore: A new standard for signing, verifying and protecting software", October 2021.

[WebAuthn] Balfanz, D. and et. al., "Web Authentication: An API for accessing Public Key Credentials Level 2", April 2021.

Authors' Addresses

Feng Geng
Huawei Technologies
Email: gengfeng@huawei.com

Panyu Wu
Huawei Technologies
Email: wupanyu3@huawei.com

Xin Chen
TrustAsia
Email: palos.chen@trustasia.com