TBD
Internet-Draft
Intended status: Standards Track
Expires: 22 February 2026

A. Shim
L. J. Han
Hopae Inc.

21 August 2025

Contextual Authentication Presentation Protocol (CAPP)
draft-shim-capp-00

Abstract

CAPP is a decentralized presentation protocol for Verifiable Credentials that enables frictionless, context-triggered, pre-consented credential sharing without requiring interactive challenge-response cycles. It is optimized for physical access, transit, event entry, and other passive authentication scenarios.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 February 2026.

Table of Contents

1. Introduction

The use of Verifiable Credentials (VCs) often requires a verifier-issued challenge, user interaction, and roundtrip communication. CAPP introduces a passive, context-aware presentation flow based on a Consent Profile locally maintained by the Holder.

CAPP is purpose-built for physical and routine authentication scenarios where users should not be required to approve every interaction explicitly. It allows for seamless, automatic credential presentation based on pre-defined conditions (e.g., location, time, trigger signal).

1.1. Example Use Cases

* *Building Access*: An employee walks into the office building and passes through the turnstile without tapping or confirming— CAPP presents a purpose-bound VP to the verifier as the user approaches the gate.

* *Airport Boarding Gate*: A traveler approaches a boarding gate and their flight ticket credential is automatically presented via NFC.

* *Event Entry*: A guest enters via QR or beacon without needing repeated approvals.

* *Subway and Transit Access*: A metro rider walks through the gate using a digital transit pass wallet that pushes the credential passively.

* *Smart Gym/Workspace Entry*: Members are authenticated passively using a pre-agreed consent profile.

* *Healthcare Check-In*: A returning patient’ s insurance credential is passively presented to the hospital kiosk.

These scenarios share a common trait: *the need for high trust, low interaction, and rapid flow*.

2. Terminology

* *Holder*: Entity that owns and controls credentials

* *Verifier*: Entity requesting and verifying a credential

* *CAPP-ready VP*: A VP that has been pre-constructed, bound to a specific context and purpose

* *Consent Profile*: A user-defined policy specifying disclosure conditions

* *Trigger*: QR/NFC/URI or other signal initiating VP flow

3. Protocol Overview

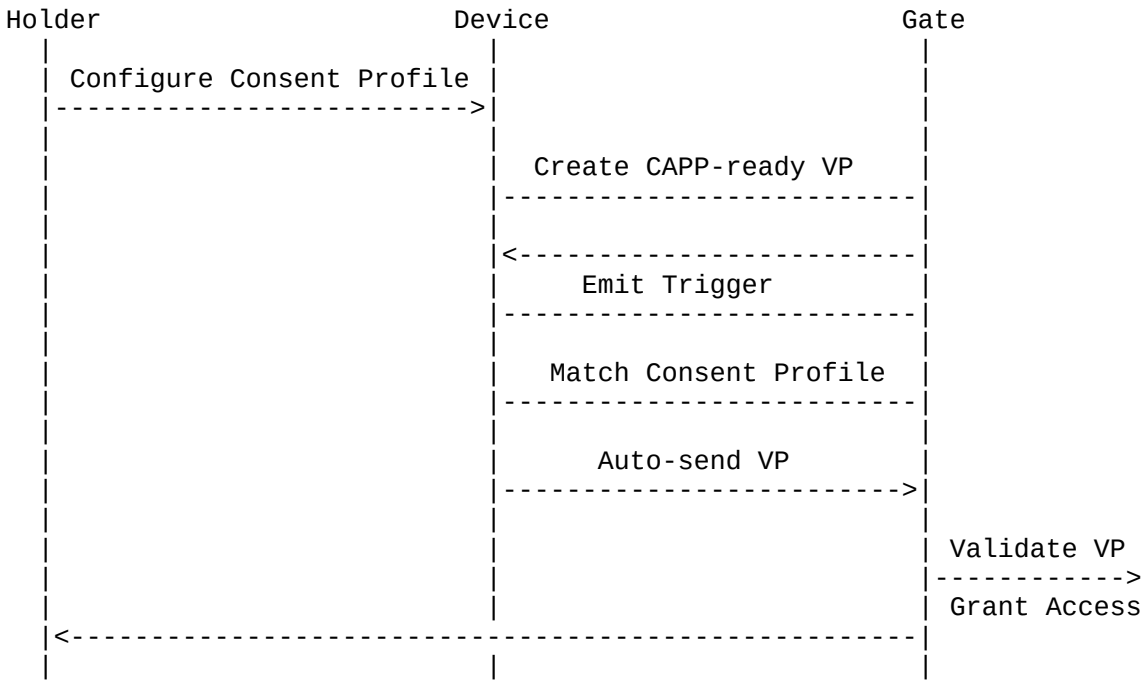
1. *Preparation*: Holder creates VP (with aud, purpose, exp, nonce), configures Consent Profile.

2. *Trigger*: Verifier emits a signed trigger (e.g., capp:// URI).

3. *Consent Profile Matching*: Device checks verifier, time, purpose, context.

4. *Automatic Presentation*: VP sent to verifier endpoint (HTTPS or DIDComm).

4. Flow Diagram (Non-Normative)



5. Examples

5.1. Consent Profile Example

```
{
  "verifier": "did:example:building",
  "purpose": "building-entry",
  "location": "166 Geary St, SF",
  "timeWindow": "08:0018:00",
  "autoPresent": true,
  "disclosurePolicy": "minimal"
}
```

5.2. VP Payload Format

```
{
  "type": ["VerifiablePresentation", "CAPPPresentation"],
  "holder": "did:example:holder123",
  "verifiableCredential": ["<VC or SD-JWT>"],
  "purpose": "building-entry",
  "aud": "did:example:corp-building",
  "exp": "2025-06-12T09:30:00Z",
  "nonce": "f8a8...x3b",
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2025-06-12T08:45:00Z",
    "verificationMethod": "did:holder#key-1",
    "proofPurpose": "authentication"
  }
}
```

6. Security Considerations

* *Replay Mitigation*: Nonce & Expiration REQUIRED; short TTL (<5m); reject replays.

* *Verifier Spoofing*: Triggers MUST be signed; device verifies before presenting.

* *Consent Profile Protection*: Encrypted storage; modifications gated by re-auth.

* *Device Theft*: Require user presence; allow emergency disable.

* *Purpose Binding*: VP MUST include purpose; verifier MUST validate match.

* *Linkability Controls*: Use pairwise DIDs; minimize metadata.

* *Endpoint Security*: TLS required; verifier validates signature, status, audience.

* *Passive Channel*: Short-lived, non-reusable triggers; no direct PII; signed JWT/hash.

* *Auditability*: Devices SHOULD log VP history; allow revocation/ pause.

7. Extensions

* presentation_definition.profile = "capp"

* VC API 2.0 extensions for triggered_presentation

* Secure passive triggers (e.g. ephemeral BLE URIs)

8. Compatibility

* W3C VC Data Model
* SD-JWT / BBS+
* DIDComm v2 / HTTPS POST
* Selective Disclosure JWT

9. IANA Considerations

This document has no IANA actions.

10. References

TBD

Authors' Addresses

Ace Shim
Hopae Inc.
Email: ace@hopae.com

Lukas J. Han
Hopae Inc.
Email: lukas.j.han@gmail.com