

IETF122 – Bangkok ACME

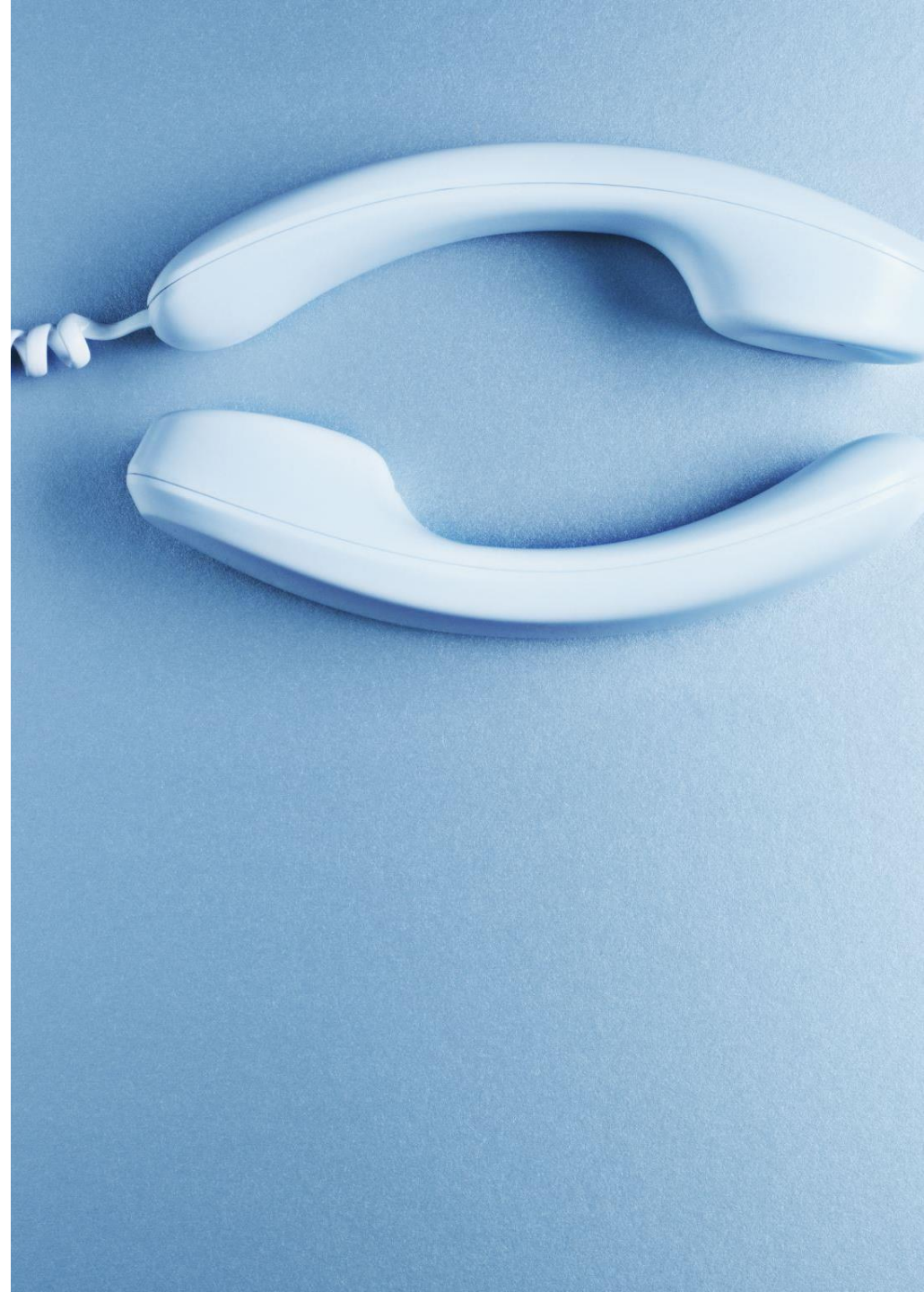
JWTClaimConstraints Authority Token Profile

draft-ietf-wendt-acme-authority-token-jwtclaimcon-00

Chris Wendt, David Hancock

Introduction to JWTClaimConstraints Authority Token Draft

- Defines an **Authority Token Profile** for ACME challenge validation specific to JWTClaimConstraints defined in STIR WG.
- Extends the existing Authority Token validation model [RFC9447] to include JWTClaimConstraints extensions.
- Enables authoritative validation necessary for Secure Telephone Identity (STI) including the ability to exclude not allowed claims and adding integrity constraints for Rich Call Data.





Motivation and Background

- Current ACME **TNAuthList Authority Token Profile** (RFC9448) addresses TNAuthList extension as part of RFC8226
- *JWTClaimConstraints* extension also defined in RFC8226 and extended by EnhancedJWTClaimConstraints in RFC9118 is not addressed.
- *JWTClaimConstraints* define constraints for usage and values of claim for PASSporTs (RFC8225) used in stir authentication.



Key Features of the Draft

- Defines new ACME Identifier Type: `"JWTClaimConstraints"`.
- Provides Authority Token Profile definition for:
 - Format used for Authority Token creation and acquisition from proper issuers of delegate certificates or associated Rich Call Data validated information.
 - ACME challenge specific to *JWTClaimConstraints* validation.



How It Works (High-Level Flow)

1. ACME client constructs a CSR with set of JWTClaimConstraints associated with claims desired.
2. Request to Token Authority for issuing a validated JWTClaimConstraints Authority Token representing client's authority over specified claims.
3. ACME client fulfills token-based challenge with returned token (tkauth-01).
4. ACME server verifies the authority token claims, signature, and account key fingerprint.
5. CA issues STI certificate with verified JWTClaimConstraints extension.



Next Steps

- Seeking WG review and adoption as a working item.
- Was presented at STIR working group and will commit to keep in loop in parallel.
- Aim to rapidly progress toward RFC assuming agreement this is fairly straight-forward.