IETF-99

ACME Identifiers and Challenges for VoIP Service Providers

draft-ietf-acme-service-provider-01

mary.ietf.barnes@gmail.com
chris-ietf@chriswendt.net

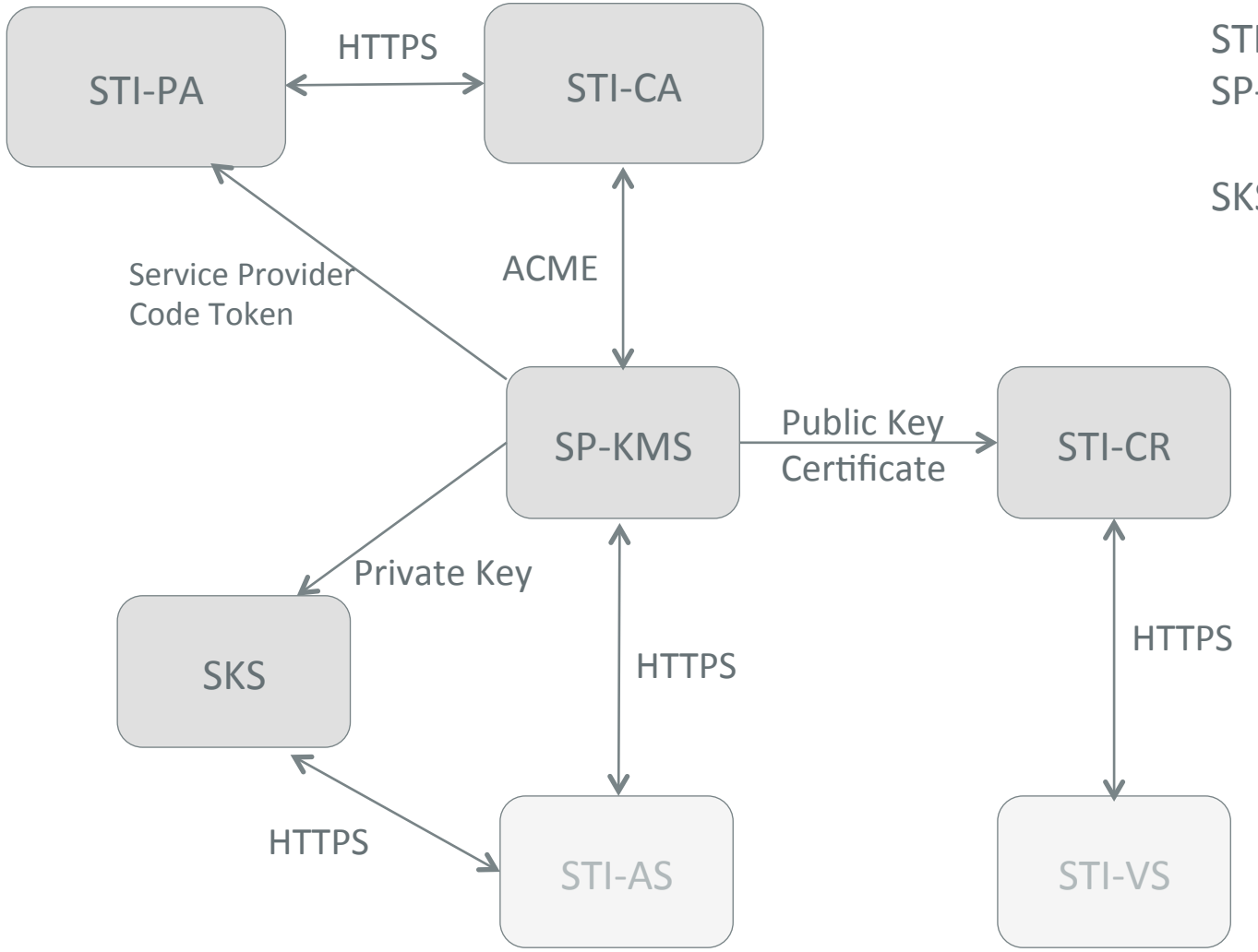
July 21, 2017

Changes since last version

- Added details about the Service Provider Code Token including function in SHAKEN and format
- Changed format of challenge type from:
  - keyAuthorization – Token |.| base64url(JWK_Thumbprint(accountKey))

  to:
  - spcAuthorization: Token |.| spcAuthzToken (which contains the “fingerprint”)
- Described processing of challenge response by the ACME server

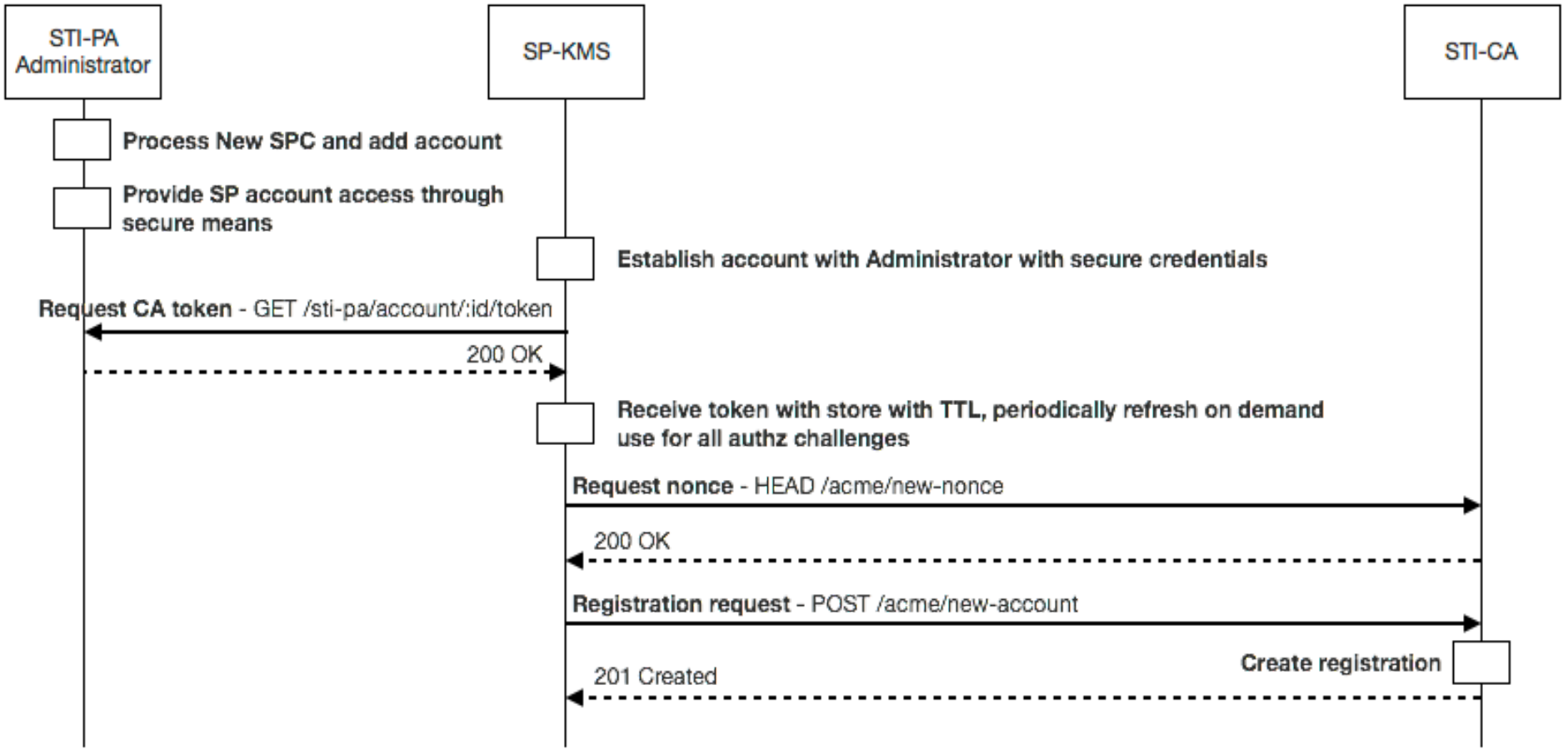
SHAKEN Certificate Management Architecture



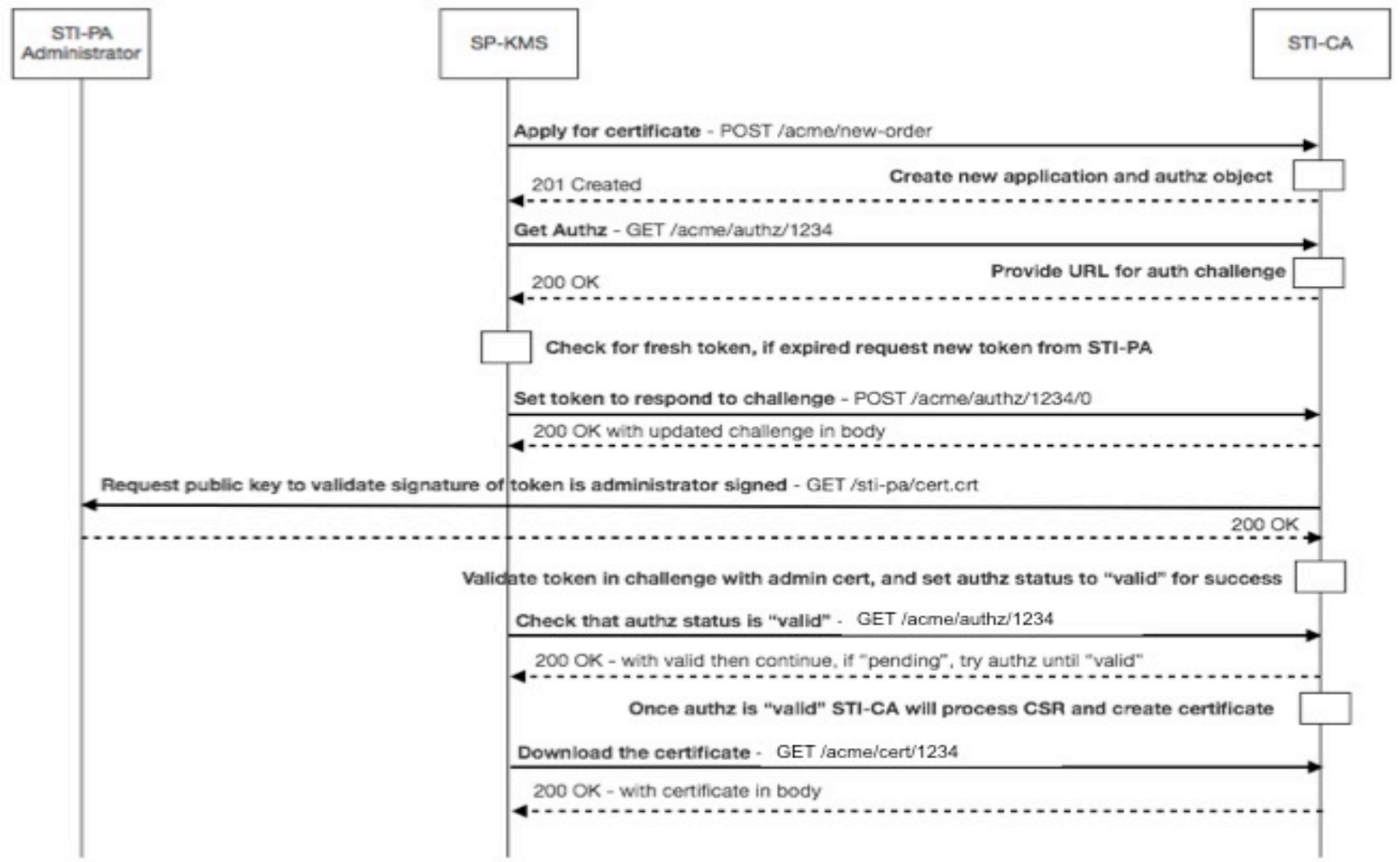
STI-PA: Policy Administrator
SP-KMS: Service Provider
Key Management Server
SKS: Secure Key Store

3

STI-PA Account Setup, SPC Token Acquisition, ACME Acct Registration



Certificate Acquisition



Service Provider Code Token

JWT Header:

- alg: Defines the algorithm used in the signature of the token. For Service Provider Code tokens, the algorithm MUST be "ES256”.
- typ: Set to standard "JWT" value.
- x5u: Defines the URL of the certificate of the STI-PA validating the Service Provider Code.

JWT Payload:

- sub (*): Service Provider Code value being validated in the form of a JSON array of ASCII strings.
- iat: DateTime value of the time and date the token was issued.
- nbf: DateTime value of the starting time and date that the token is valid.
- exp: DateTime value of the ending time and date that the token expires.
- fingerprint: : (Certificate) key fingerprint of the ACME credentials the Service Provider used to create an account with the CA.

“fingerprint” is of the form:

base64url(JWK_Thumbprint(accountKey))

* For ATIS-1000080, only a single Service Provider Code is required in the “sub” field.

Discussion points

1. “Identifier” is specific to STIR TNAuthList (includes both TNs and Service Provider Codes)
   - Draft-ietf-acme-telephone defines a “tn” identifier
     - Could this re-use TNAuthList and define a new challenge type (tn-01)?
       - Depends upon the answer to 2.
2. Challenge type is specific to Service Provider Code Tokens
   - Could we use this as a generic challenge type for tokens of the same form?
     - Possibly *but* could slow down progression of this document