

A Public Key Service Provider for Verification in Multiple Issuers and Verifiers

An optional public key service for issuer key registration and discovery, enabling verifiers to obtain current verification key material on demand in multi-domain environments.

Source: [draft-wang-spice-public-key-service-provider-02](#)

DongHui Wang,
Faye Liu,
Lun Li,
Yuning Jiang,
— Huawei

Key Problem

Digital credential ecosystems become hard to operate when verifiers need to validate credentials from many issuers whose keys can rotate or be revoked.

Why This Matters

Current Approaches

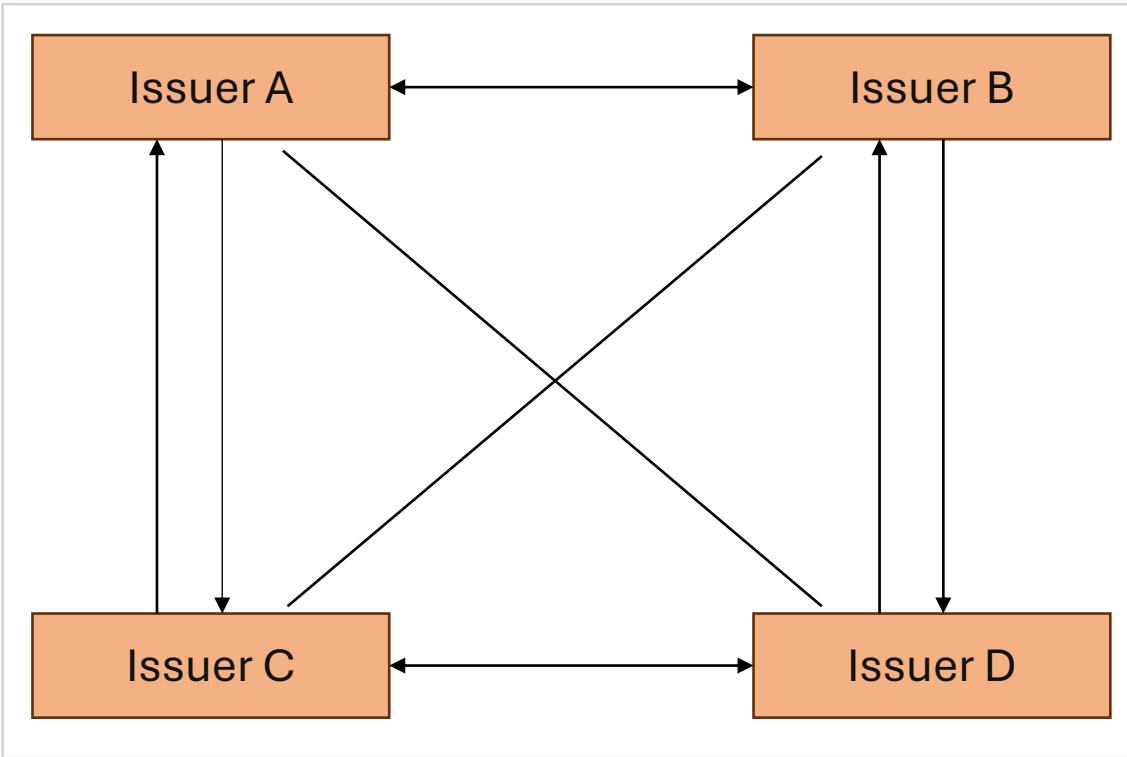
- Verifiers maintain local trust stores where administrators manually provision issuer certificates and keys.
- A mutually trusted third party acts as a root to bridge trust between different issuers.
- Issuers establish direct trust by mutually signing and certifying each other's public keys.

Limitations of Current Approaches

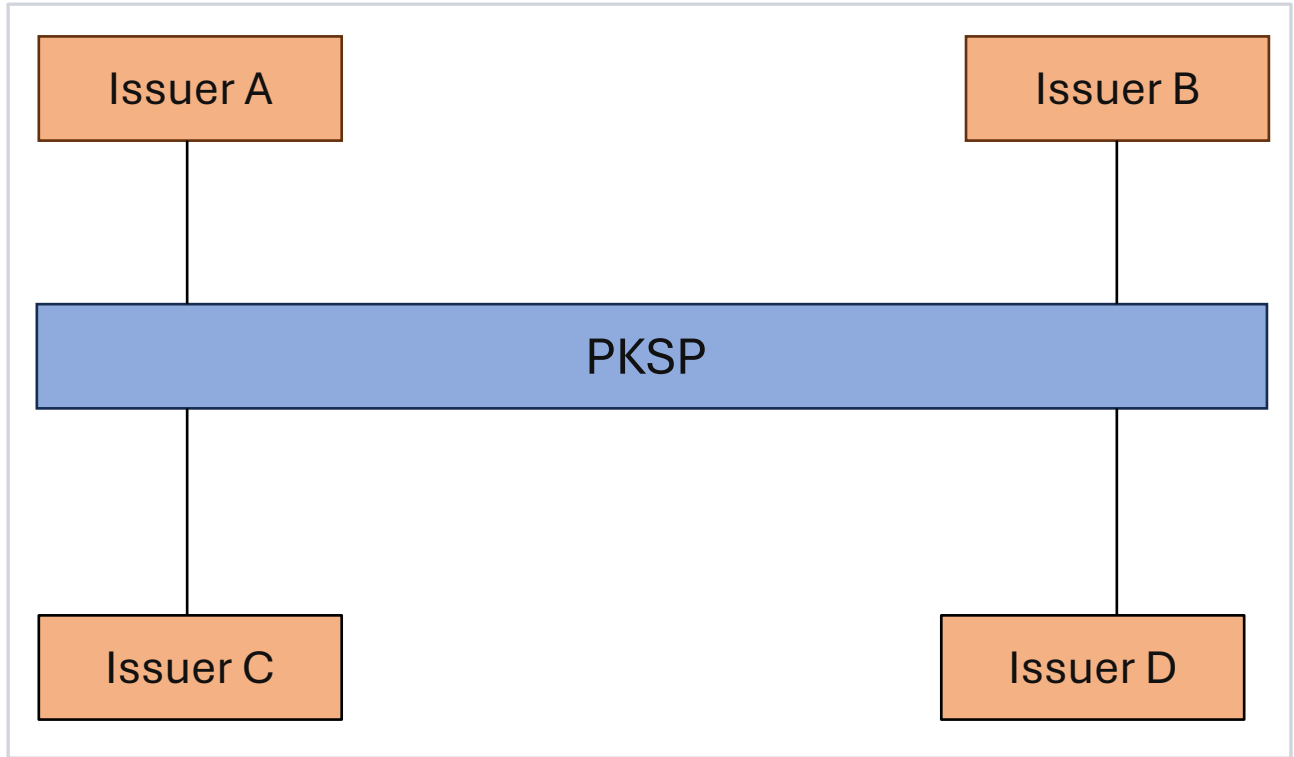
- Manual configuration does not scale well in constrained or dynamic environments.
- Difficult to manage frequent key rotations and revocations.
- A common trust anchor may not match real-world trust relationships across domains.
- Cross-certification becomes complex in many-to-many settings.

PKSP reduces operational complexity by supporting issuer public key registration and discovery, including key status information needed for verification.

Architecture Overview

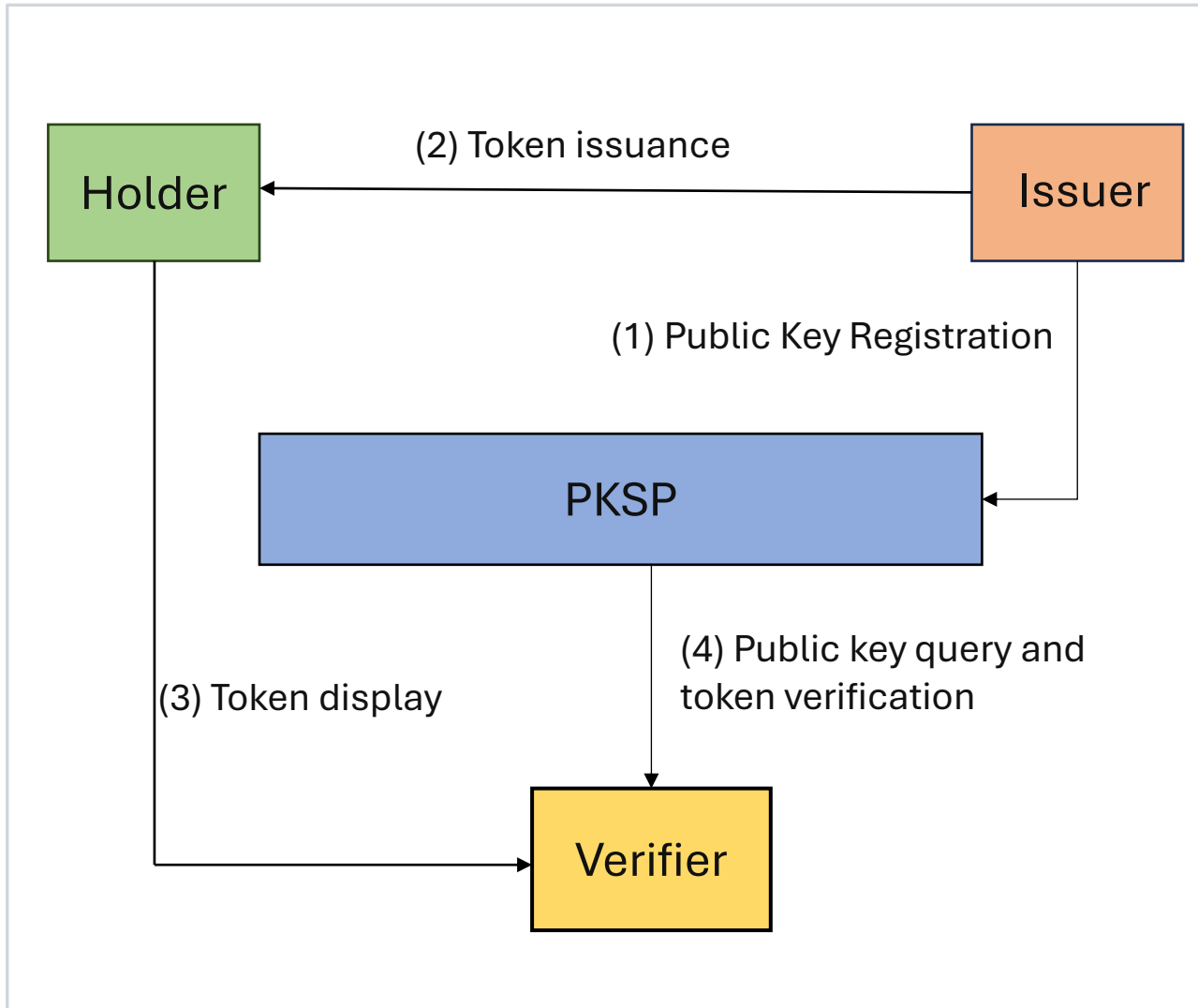


Traditional approach:
Issuers establish pairwise mutual trust. Workload increases further when public key updates and revocations occur.



PKSP-based approach:
Issuers publish registration and revocation messages to PKSP, which can then be accessed by other issuers or verifiers during verification.

PKSP for Verification



Role of PKSP

- Accepts public key registration from issuers.
- Stores registered public keys securely and in a non-tamperable manner.
- Returns the corresponding public key when a verifier queries for a key.

Verification inputs

- Queries can use issuer identification information from the token, such as the “iss” field.
- Response metadata may include the public key expiration date and key type.

Registration Flow

1 Submission by the Issuer

The issuer sends public keys, declared purposes, and descriptive information such as name, identifier, expiration time, and token-claim declaration.

2 Validation by the PKSP

PKSP validates the issuer public key and examines the descriptive information for accuracy and completeness, potentially cross-referencing existing sources.

3 Response from the PKSP

If the information is valid and compliant, PKSP returns a successful registration response. Otherwise, it returns a rejection notice with the reason.

4 Completion of Registration

Once the issuer receives a positive response, the registration is complete. PKSP updates its records and supports future public key queries.

Verification Flow

1	Token Submission	The Holder sends the token to the Verifier.
2	Public Key Request	Verifier identifies the Issuer from the token and requests their public key from the PKSP
3	PKSP Processing	PKSP authenticates the Verifier and returns the requested public key along with relevant metadata (e.g., expiration)
4	Signature Validation	Verifier uses the retrieved key to validate the token's digital signature.
5	Outcome Handling	<ul style="list-style-type: none">• Success: Verifier proceeds with business logic (e.g., granting access).• Failure: Verifier rejects the request and logs the event for security auditing.

PKSP supports token verification by making issuer public key material available on demand.

Continue the Conversation, and Thank You!

What PKSP contributes

- An optional architecture for issuer public key registration and discovery in multi-domain environments.
- A registration flow between issuer and PKSP, and a verification flow between verifier and PKSP.
- A permissioned distributed ledger as one enabling technology direction.

Future works

- Operation of distributed ledger.
- Detailed protocol.
- Security considerations.

Feedback is welcome on deployment assumptions, registration and query procedures, and the role of PKSP in SPICE token validation workflows.