



Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca



Italiadomani  
PIANO NAZIONALE  
DI RIPRESA E RESILIENZA



SERICS  
SECURITY AND RIGHTS IN THE CYBERSPACE



# CrypTO

## CONFERENCE



Politecnico  
di Torino



Telsy

A TIM  
ENTERPRISE  
BRAND

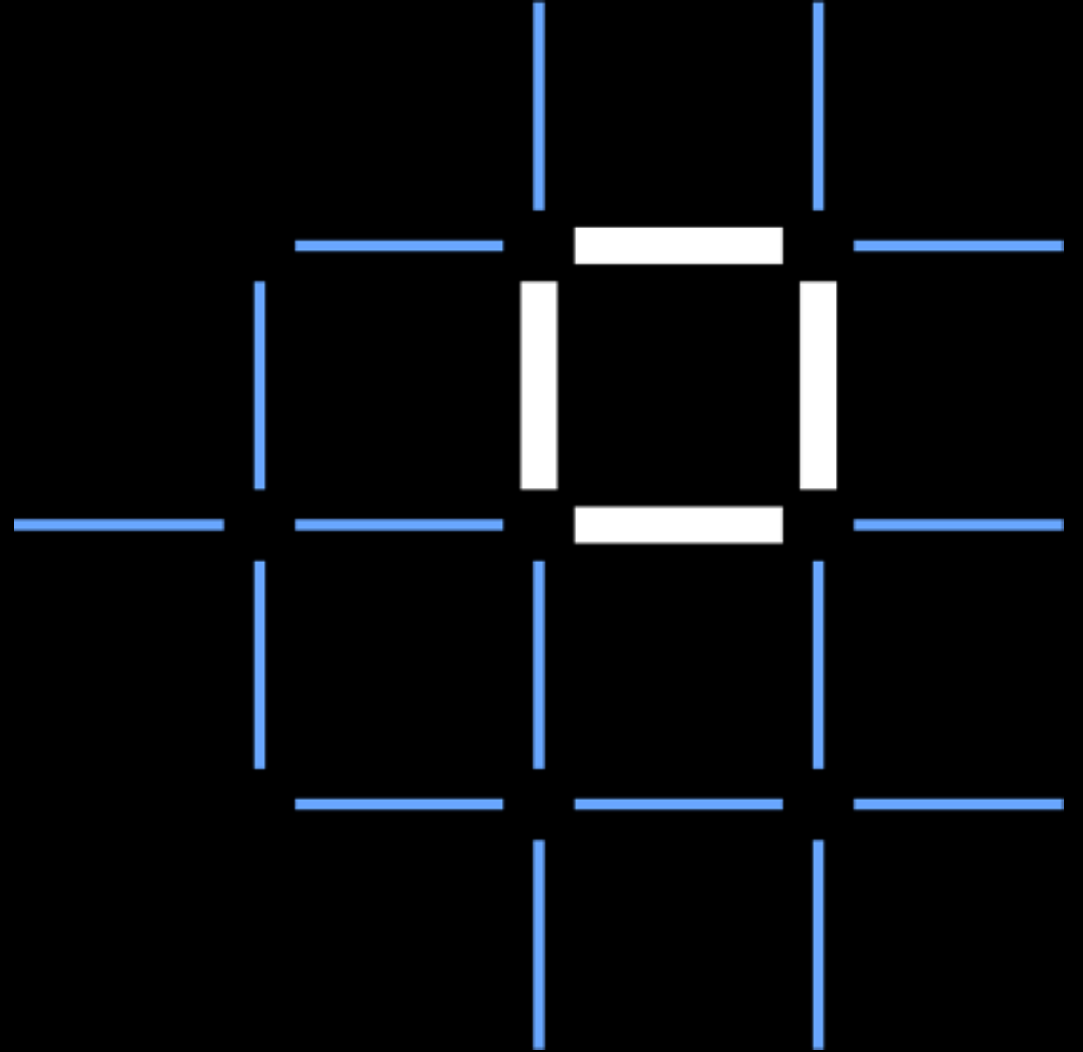


# Cryptographic protocols for identity management in digital asset systems

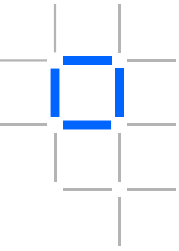
---

Alessandro Sorniotti

23 May, 2025

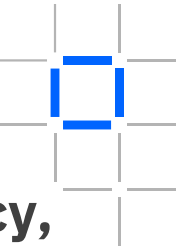


# A couple of words about myself



- Principal Research Scientist at IBM Zurich
  - System security
  - Applied cryptography
  - Distributed systems
- Previously at SAP Research
- PhD at EURECOM
- BEng/MEng from polito

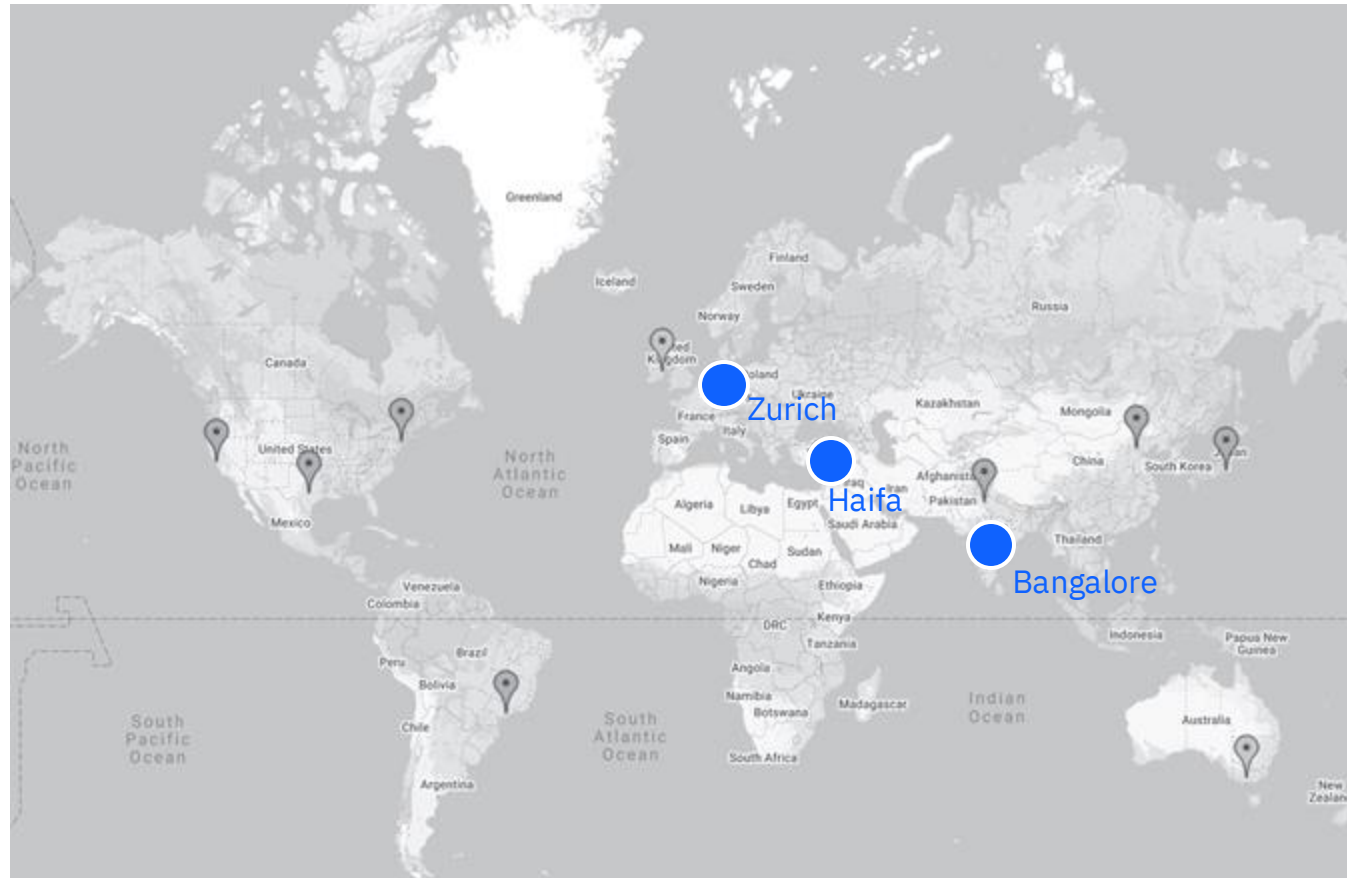
# Decentralised Trust @IBM Research



- Strong expertise in **Security & Privacy, Cryptography, DLT/Consensus algorithms, Distributed Systems**

- Key contributor of:
  - Token Exchange Frameworks
  - DLT Interoperability Frameworks
  - HPL Fabric architecture 1.0 & Development team
  - **Security, Privacy & Consensus** mechanisms HPL Fabric 0.5, 1.0+
  - Self Sovereign (**privacy-preserving**) identity cryptographic blocks

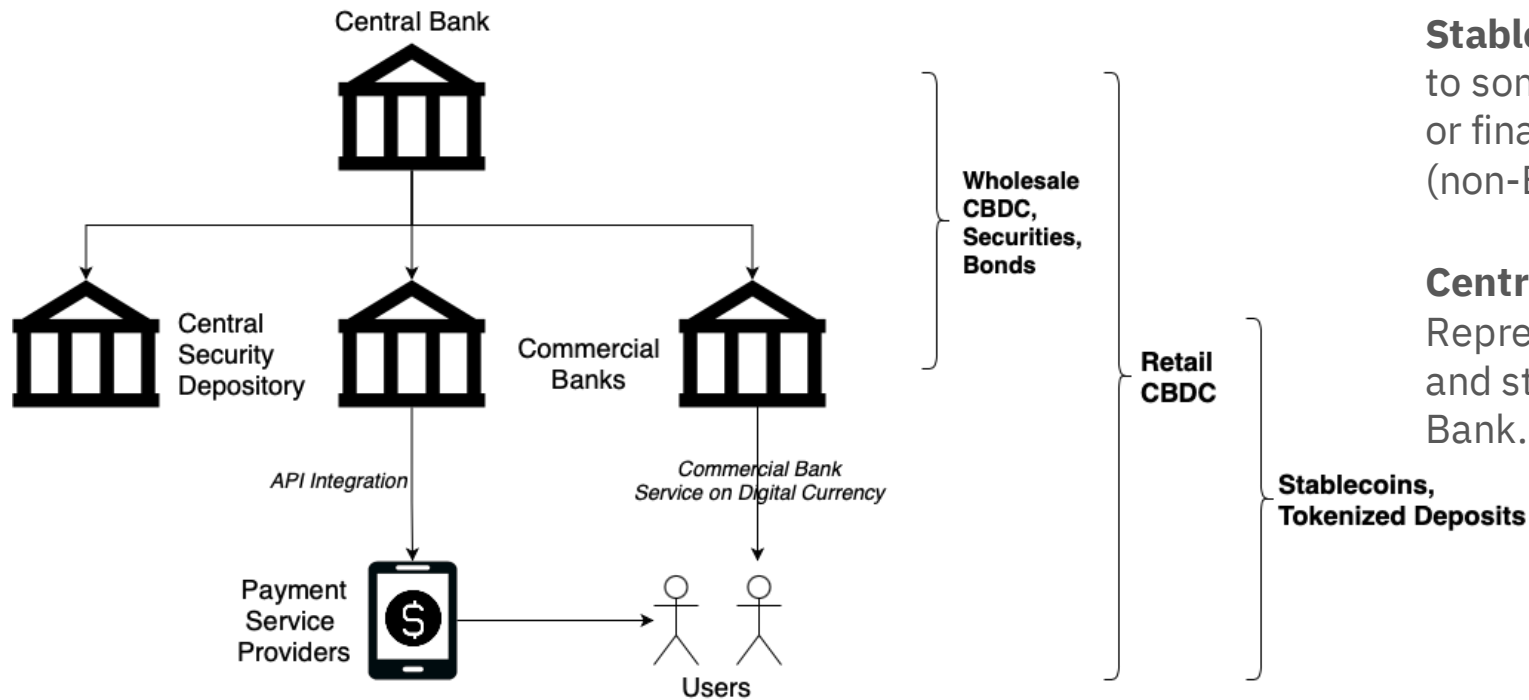
- Involvement in **client engagements** to route client requirements to **differentiating** reusable application components, or prominent platform extensions



Digital Assets and Identity Management

# Digital Assets in the wild

Tokenization value-add: Inclusion, Transparency, Interoperation, Innovation, Resilience



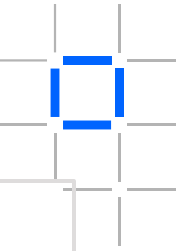
**Tokenized Securities:** Representations of securities bonds in a transparent and interoperable digital system.

**Tokenized Deposits:** Representations of commercial bank (fractional) deposits in a transparent and interoperable digital system issued.

**Stablecoins:** Digital assets that peg their market value to some external reference (e.g., currency, commodity or financial instrument). Frequently issued by private (non-Bank) entities on permissionless systems.

**Central Bank Digital Currency (CBDC):** Digital Representation of today's fiat currency for payments and storage of value with direct liability to the Central Bank. It comes in retail and wholesale versions.

# The three pillars of our work



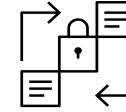
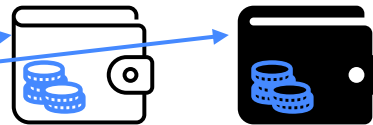
## User Identity:

- Unique identifier per user
- Public key associated to the individual's financial activity
- A digital credential (SSI) binding the user identifier to the public key



## Token (UTXO model):

- Digital representation of an asset
- Consists of a type, a value, and the asset's owner
- With various privacy levels



## DLT for settlement:

- Decentralisation: can sustain failing or compromised regions/nodes
- Ledger of transactions
- Value-add: resilience, transparency, extensibility, provenance



Identity for Digital Assets



# Self-sovereign identity (SSI) is a viable approach for Digital Assets

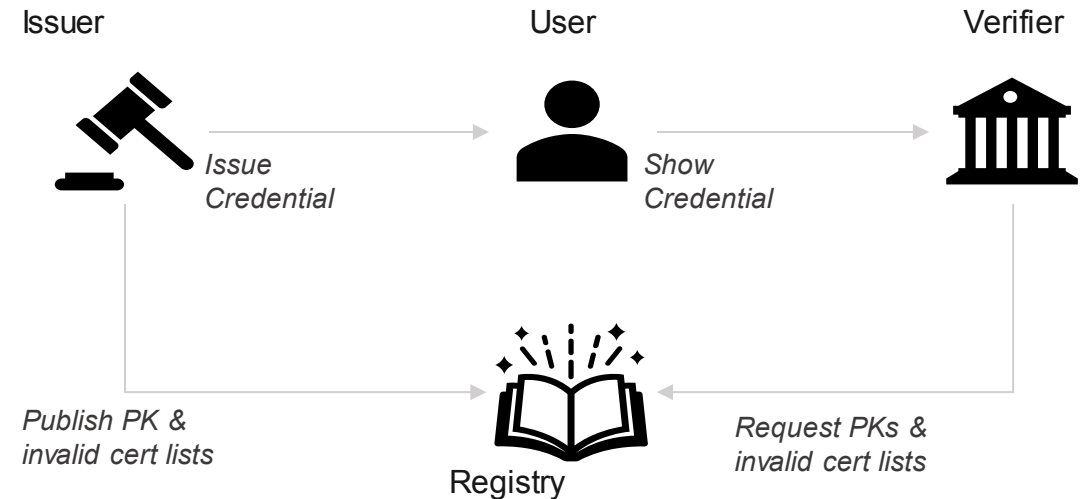


## Why do we even need identity?

- Permissioned system
- Strict regulations (KYC, AML...)

## SSI Value-add:

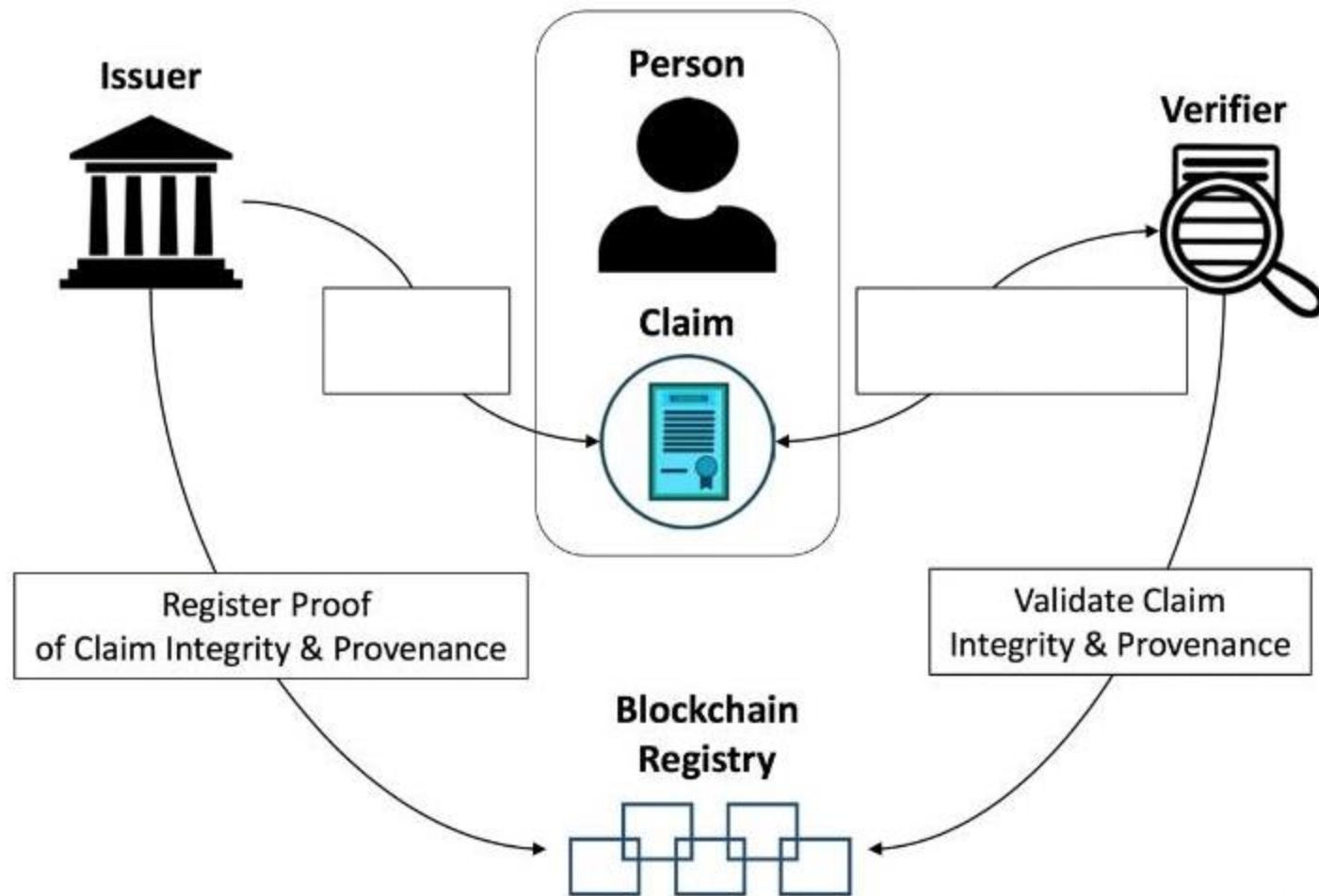
- Privacy
- Security
- Scalability
- Interoperation
- Cost-effective

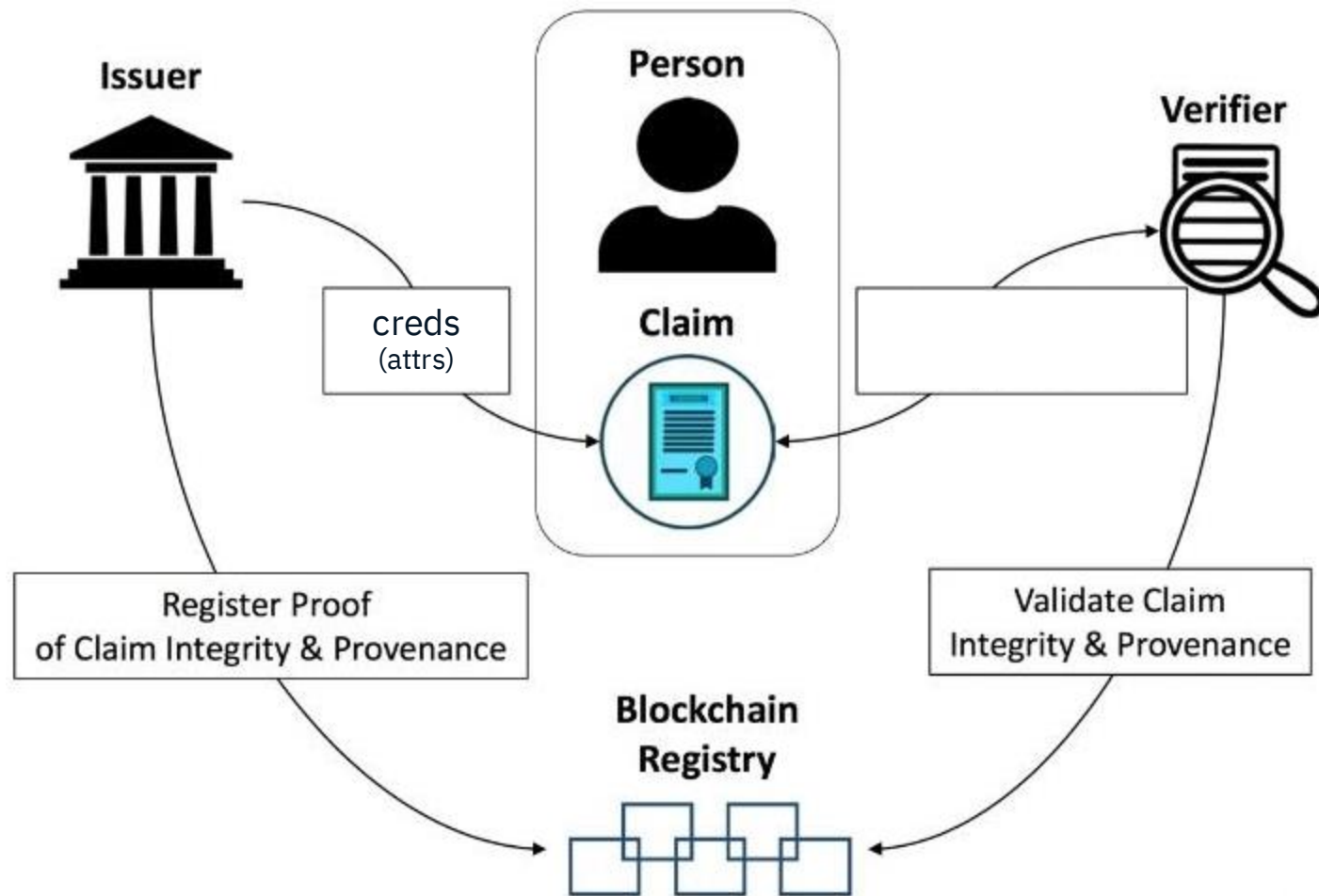


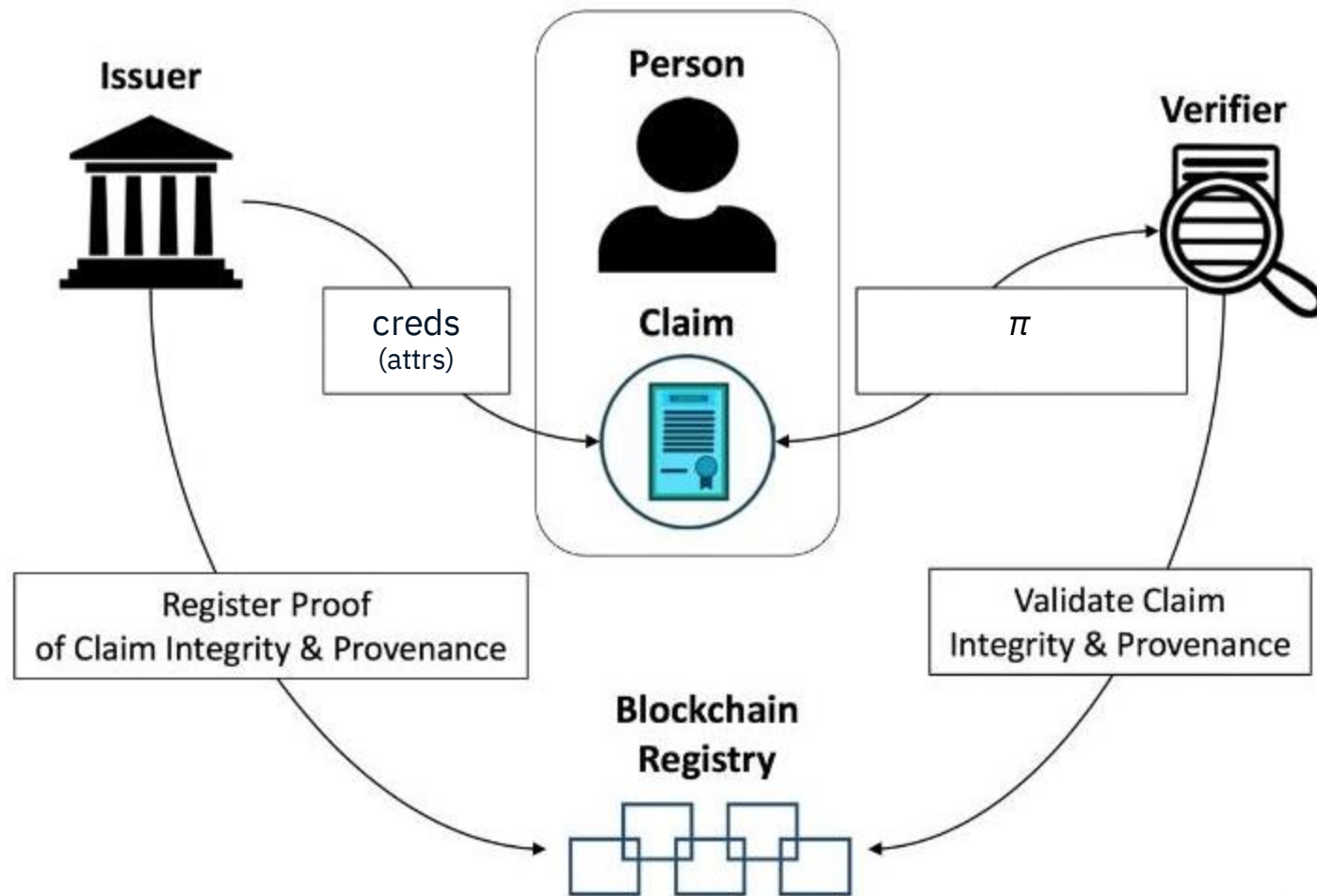
**Anonymous credentials:** means of authenticating to 3<sup>rd</sup> parties while achieving privacy and data minimisation

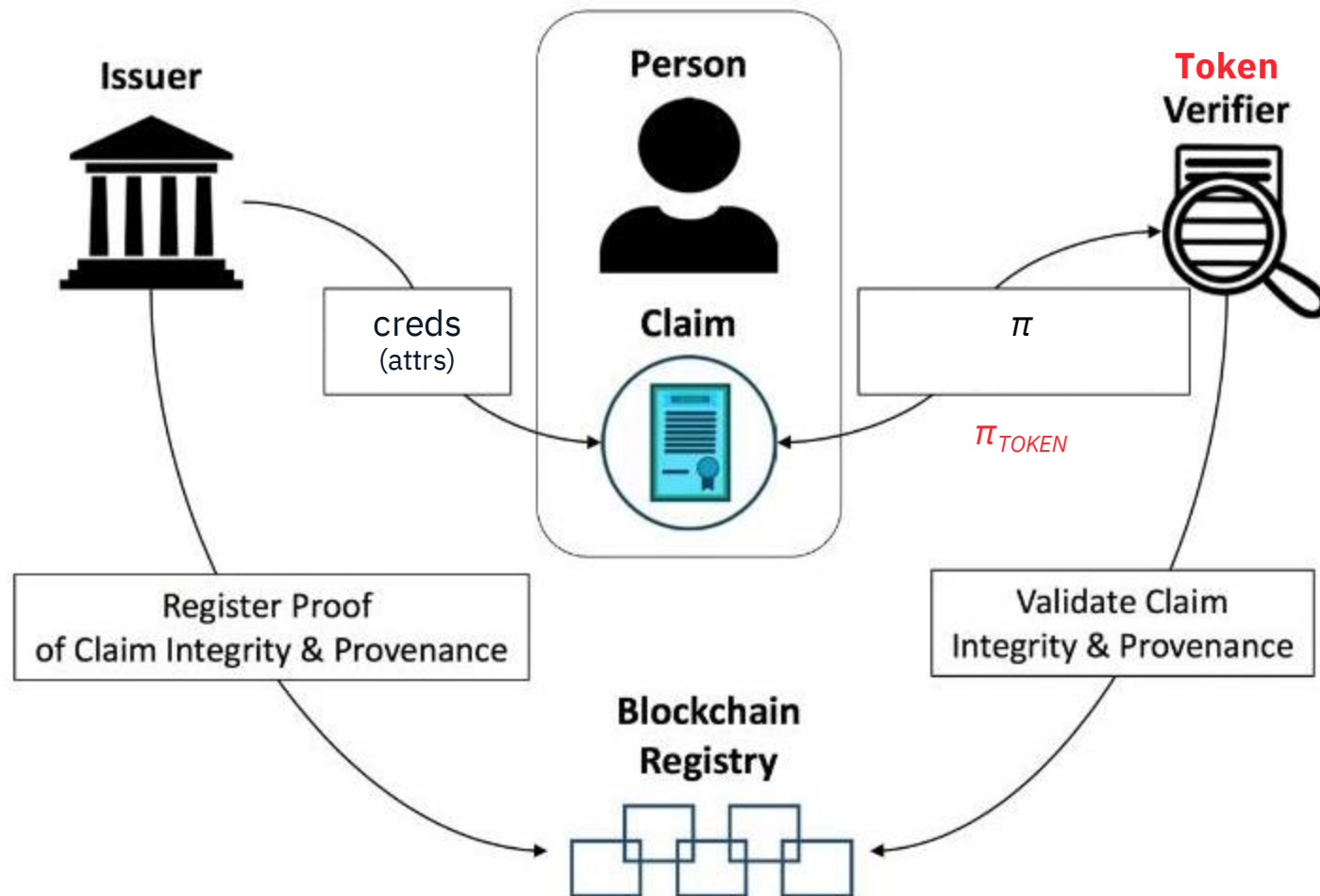
Based on

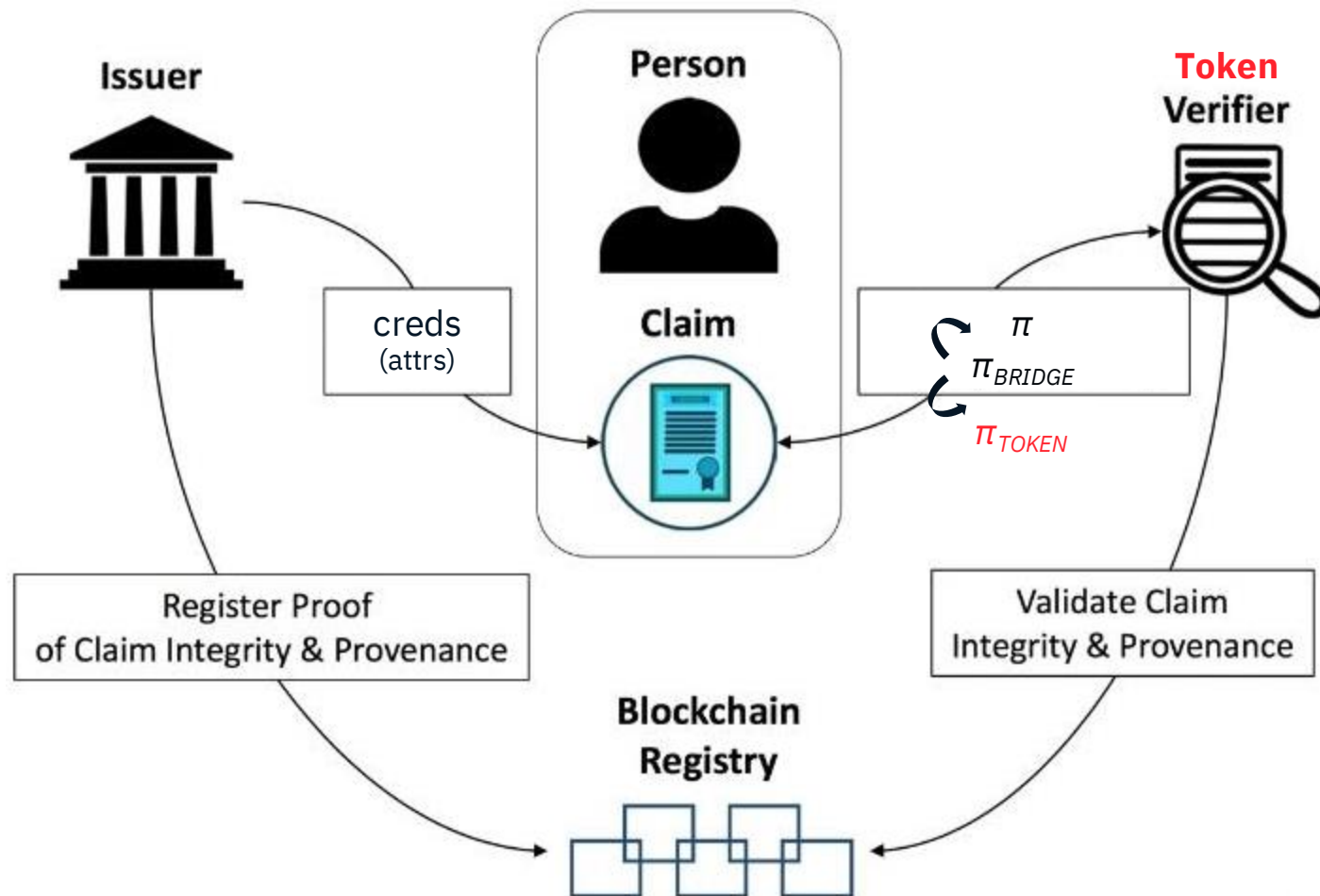
- Signature schemes
- Zero-knowledge proofs

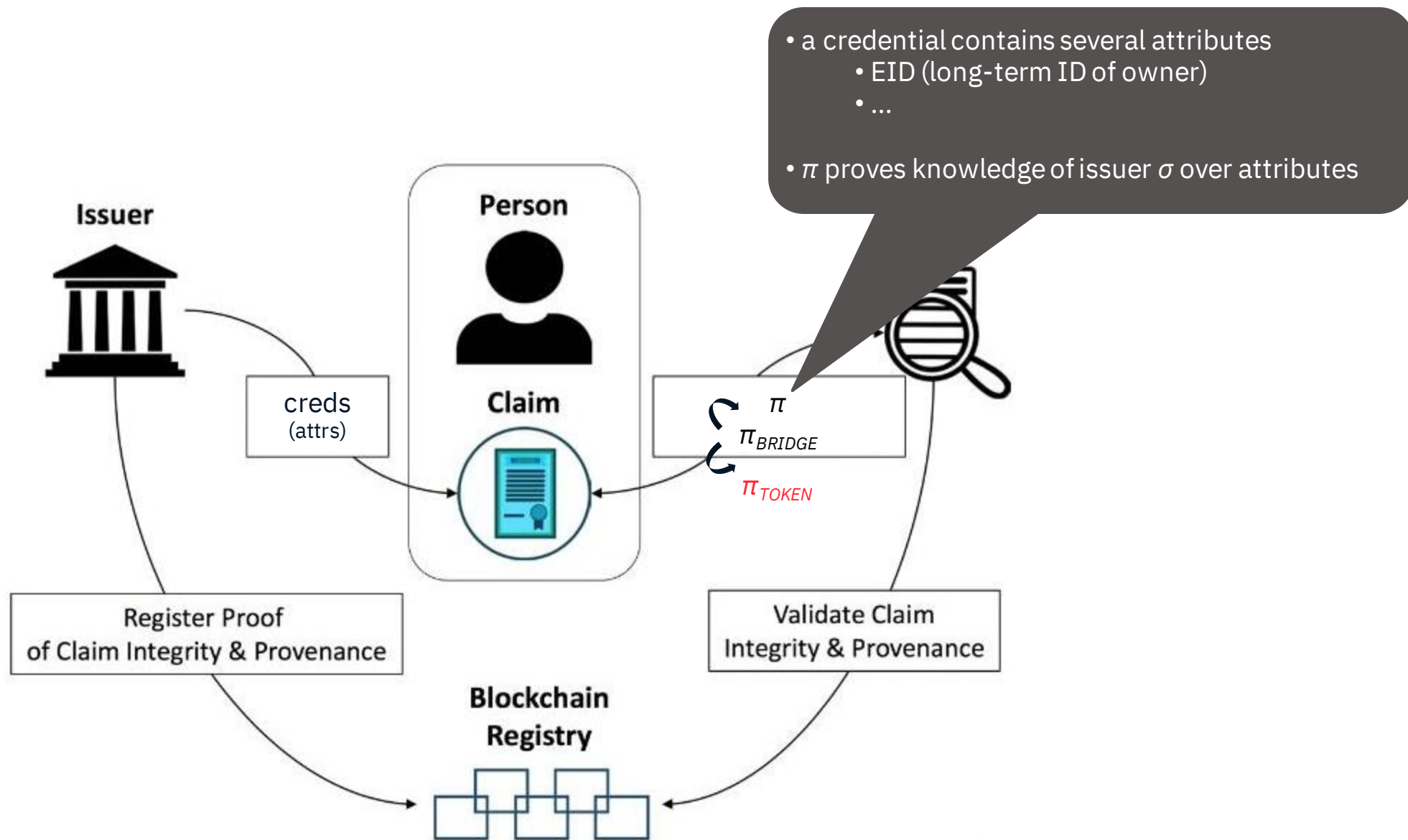


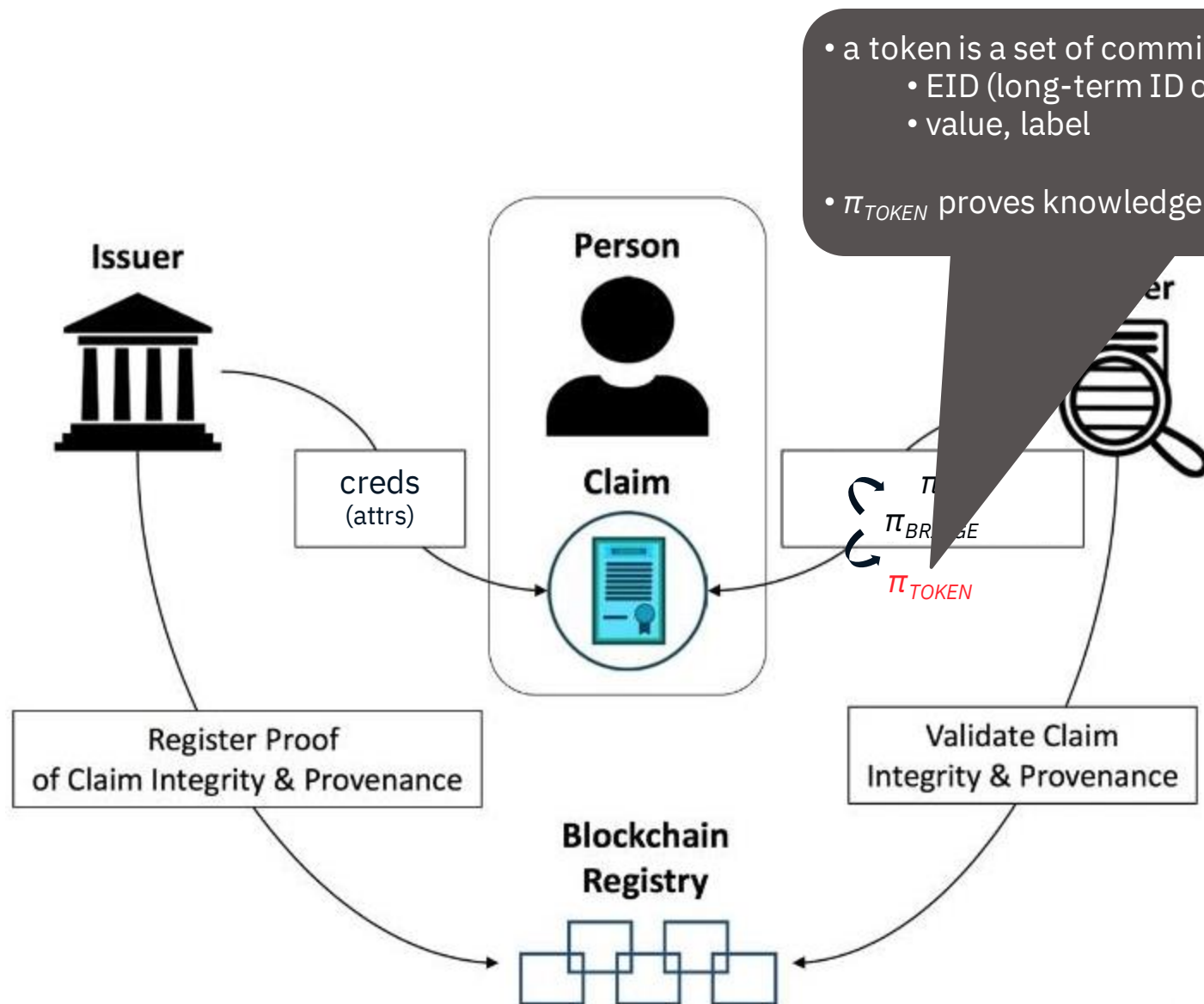






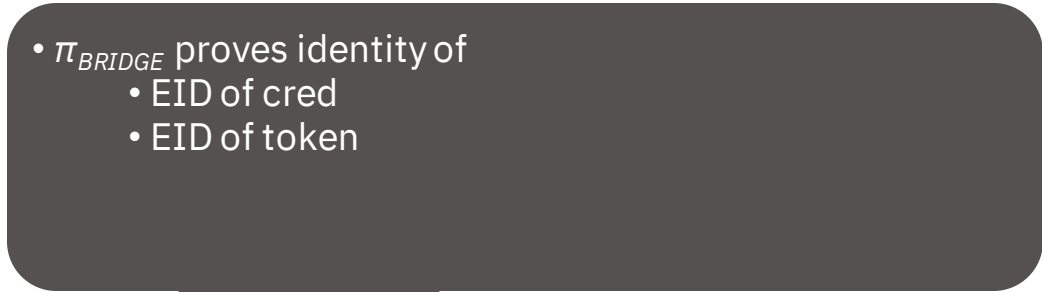


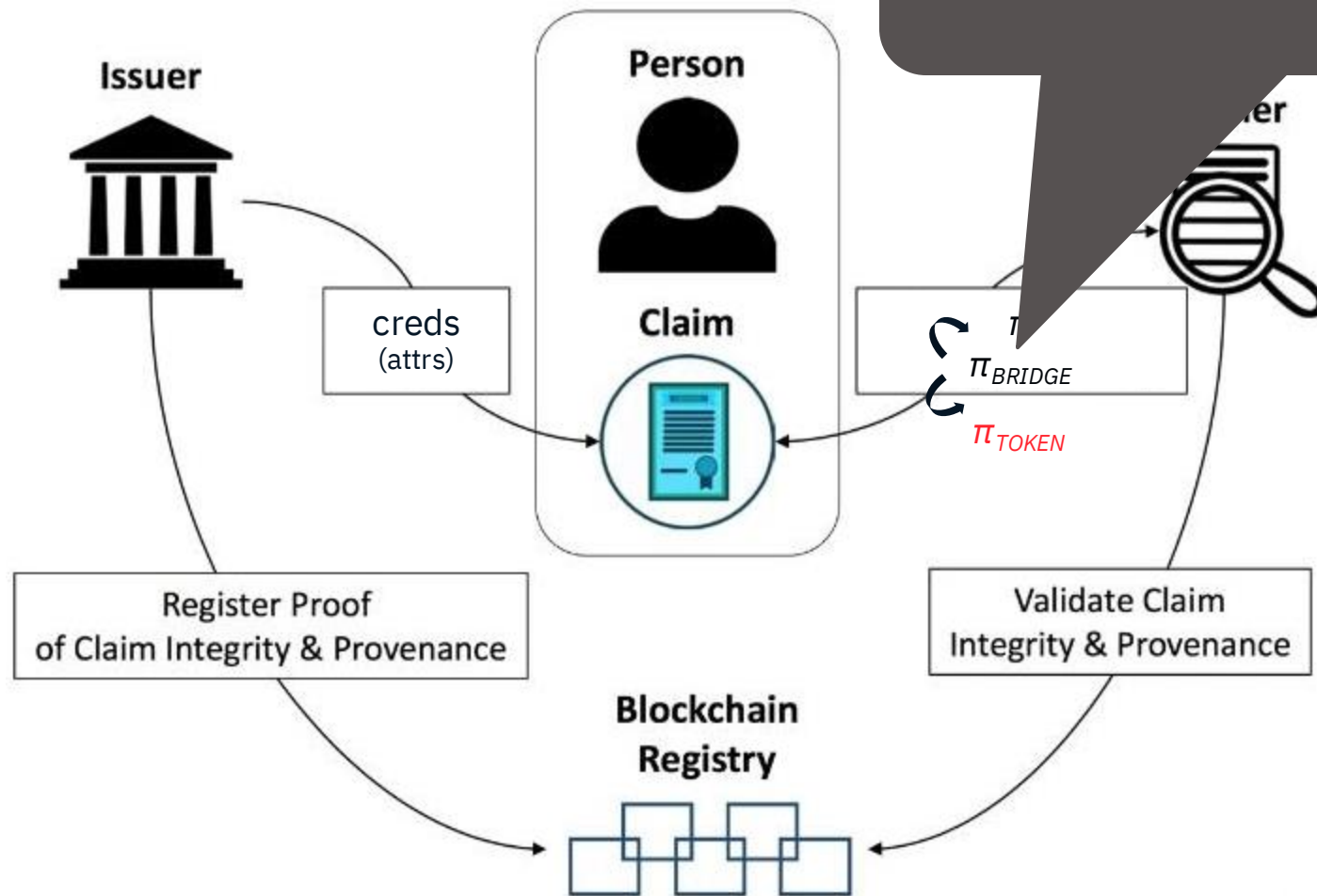




- a token is a set of commitments
  - EID (long-term ID of owner)
  - value, label
- $\pi_{TOKEN}$  proves knowledge of openings (and other stuff...)







- $\pi_{BRIDGE}$  proves identity of
  - EID of cred
  - EID of token

Hardware security for payments

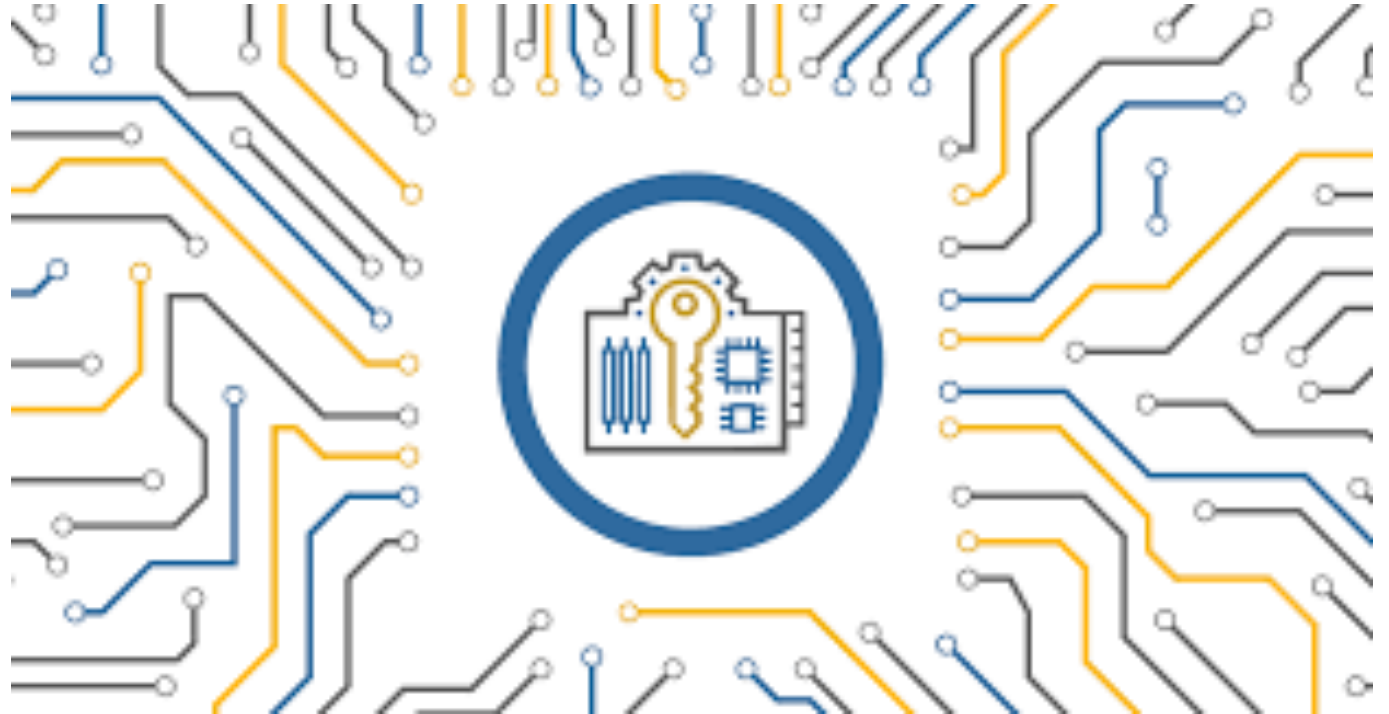
# Self-sovereign identity (SSI) is a viable approach for Digital Assets

## Where to store secrets?

- Secure payment system requires
  - high security for secrets
  - against both user and hackers

## Degrees of security

- Software
  - flexible
  - easy to hack
- Secure Enclaves
  - higher security
  - less programmable/extensible
- HSMs
  - highest security
  - least programmable
  - severely resource-constrained



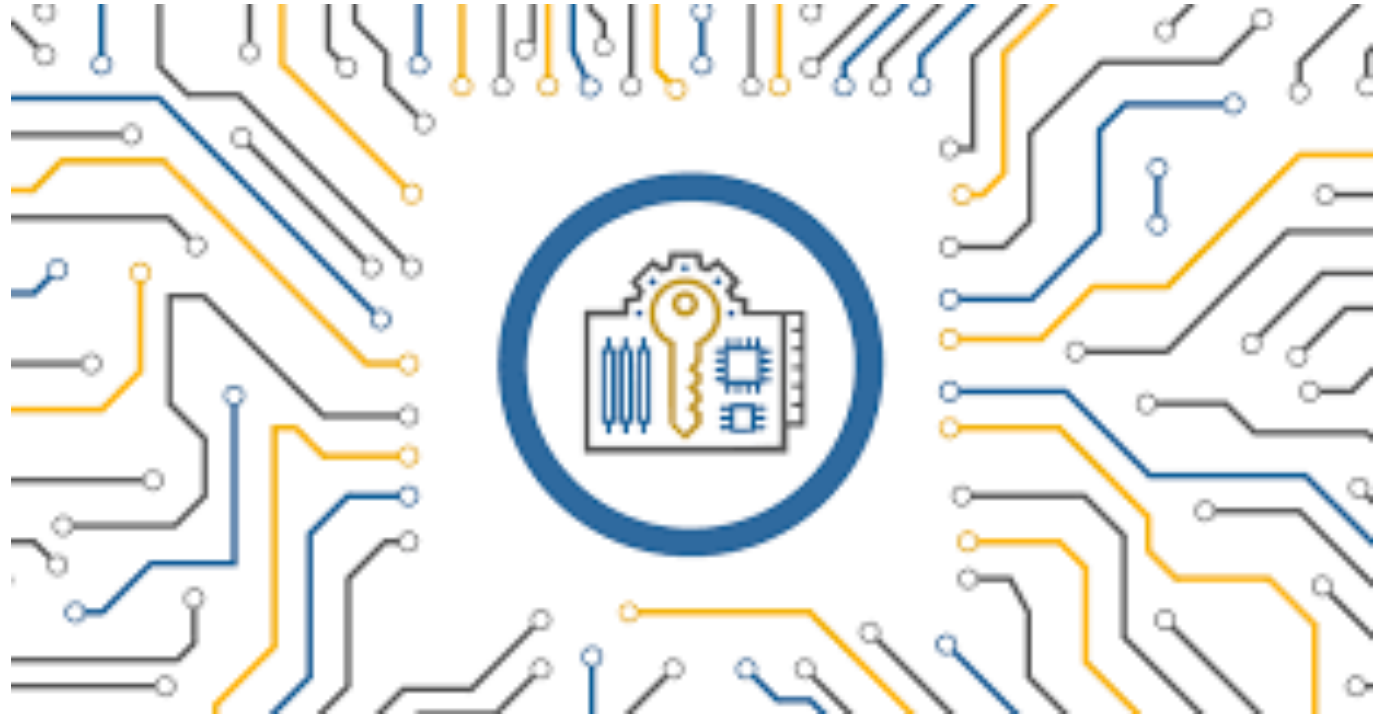
# Self-sovereign identity (SSI) is a viable approach for Digital Assets

## Where to store secrets?

- Secure payment system requires
  - high security for secrets
  - against both user and hackers

## Degrees of security

- Software
  - flexible
  - easy to hack
- Secure Enclaves
  - higher security
  - less programmable/extensible
- **HSMs**
  - **highest security**
  - **least programmable**
  - **severely resource-constrained**



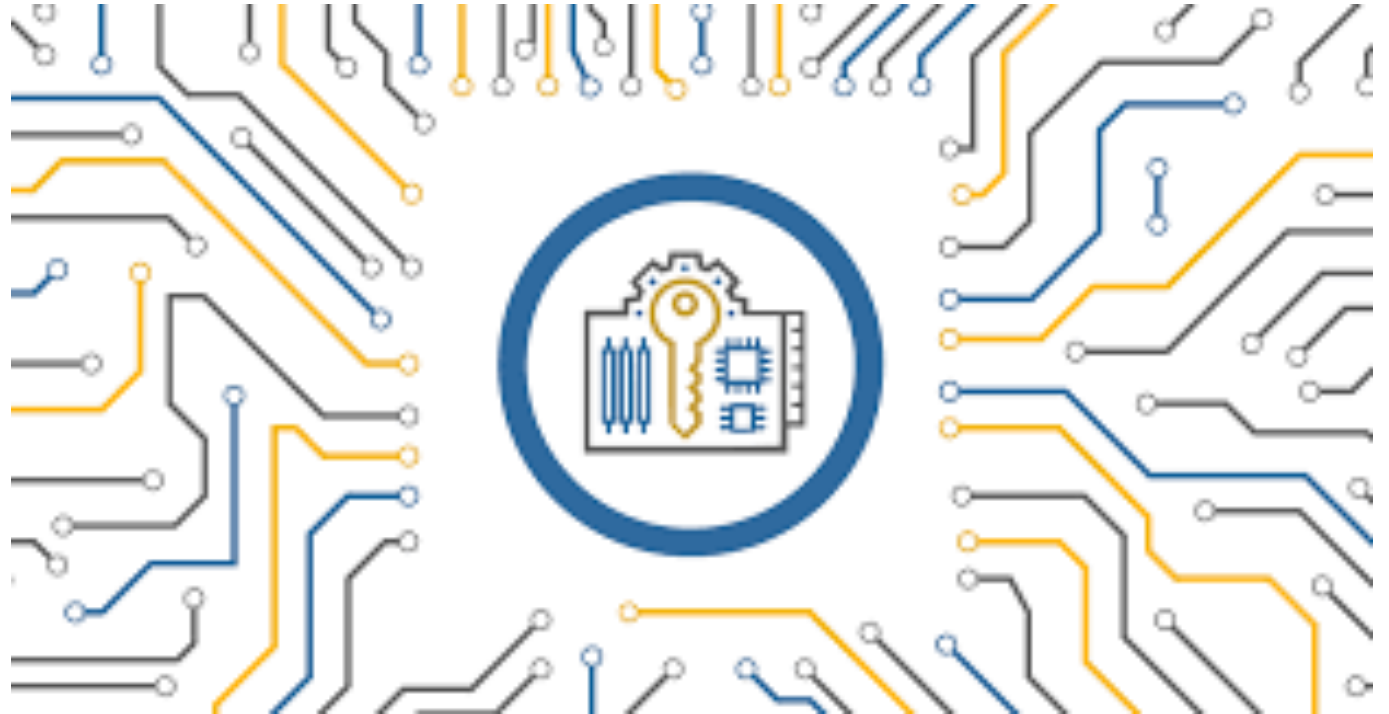
# Self-sovereign identity (SSI) is a viable approach for Digital Assets

## Where to store secrets?

- Secure payment system requires
  - high security for secrets
  - against both user and hackers

## Degrees of security

- Software
  - flexible
  - easy to hack
- Secure Enclaves
  - higher security
  - less programmable/extensible
- **HSMs**
  - **highest security**
  - least programmable
  - **severely resource-constrained**



# POK for BBS+ Signature [CDL16]

## Inputs

- **Prover's Inputs:**  $m_1, \dots, m_\ell, (A, e, s)$  satisfying:  $A = (g_1 h_0^s \prod_{i=1}^{\ell} h_i^{m_i})^{1/(e+x)}$ .
- **Verification Predicate:**  $m_i = v_i$  for  $i \in V \subseteq \{2, \dots, \ell\}$ .

## Proof Generation

- **Compute Statement:**

1.  $r_1 \leftarrow \mathbb{Z}_p, r_2 \leftarrow \mathbb{Z}_p, r_3 = r_1^{-1}, b = g_1 h_0^s \prod_{i=1}^{\ell} h_i^{m_i}, s' = s - r_2 r_3.$
2.  $A' = A^{r_1}, d = b^{r_1} h_0^{-r_2}, \bar{A} = A'^{-e} b^{r_1}.$

- **Compute Proof:**

$$\pi := \text{PK} \left\{ \begin{array}{l} (\{m_i\}_{i \in H}, e, s', r_2, r_3) : \\ A'^{-e} h_0^{r_2} = \bar{A}/d \wedge \\ d^{-r_3} h_0^{s'} \prod_{i \notin V} h_i^{m_i} = g_1^{-1} \prod_{i \in V} h_i^{-m_i} \end{array} \right\}$$

## Verification

Verify  $\pi$  and check  $e(A', w) = e(\bar{A}, g_2).$

# POK for BBS+ Signature [CDL16]

# Joint POK for BBS+ Signatures

## Inputs

- **Prover's Inputs:**  $m_1, \dots, m_\ell, (A, e, s)$  satisfying:  $A = (g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i})^{1/(e+x)}$ .
- **Verification Predicate:**  $m_i = v_i$  for  $i \in V \subseteq \{2, \dots, \ell\}$ .

## Inputs

- **Card's Inputs:**  $m_1$ .
- **Holder's Inputs:**  $m_2, \dots, m_\ell, (A, e, s)$ , satisfying  $A = (g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i})^{1/(e+x)}$ ,  $Q = h_1^{m_1}$ .
- **Verification Predicate:**  $m_i = v_i$  for  $i \in V \subseteq \{2, \dots, \ell\}$ .
- **Common Randomness:**  $r \leftarrow \mathbb{Z}_p$ .

## Proof Generation

- **Compute Statement:**
  1.  $r_1 \leftarrow \mathbb{Z}_p, r_2 \leftarrow \mathbb{Z}_p, r_3 = r_1^{-1}, b = g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i}, s' = s - r_2 r_3$ .
  2.  $A' = A^{r_1}, d = b^{r_1} h_0^{-r_2}, \bar{A} = A'^{-e} b^{r_1}$ .

- **Compute Proof:**

$$\pi := \text{PK} \left\{ \begin{array}{l} (\{m_i\}_{i \in H}, e, s', r_2, r_3) : \\ A'^{-e} h_0^{r_2} = \bar{A}/d \wedge \\ d^{-r_3} h_0^{s'} \prod_{i \notin V} h_i^{m_i} = g_1^{-1} \prod_{i \in V} h_i^{-m_i} \end{array} \right\}$$

## Proof Generation

- **Compute Statement:**
  1. (Holder):  $r_1 \leftarrow \mathbb{Z}_p, r_2 \leftarrow \mathbb{Z}_p, r_3 = r_1^{-1}, b = g_1 h_0^s \cdot Q \cdot \prod_{i=2}^\ell h_i^{m_i}, s' = s - r_2 r_3 - r$ .
  2. (Holder):  $A' = A^{r_1}, d = b^{r_1} h_0^{-r_2}, \bar{A} = A'^{-e} b^{r_1}$ .
  3. (Card):  $B = h_1^{m_1} h_0^r$ .

- **Compute Proof:**

$$\pi := \text{PK} \{ (m_1, r) : h_1^{m_1} h_0^r = B \}$$
$$\pi' := \text{PK} \left\{ \begin{array}{l} (\{m_i : 1 \neq i \notin V\}, e, s', r_2, r_3) : \\ A'^{-e} h_0^{r_2} = \bar{A}/d \wedge \\ d^{-r_3} h_0^{s'} \prod_{1 \neq i \notin V} h_i^{m_i} = g_1^{-1} B^{-1} \prod_{i \in V} h_i^{-m_i} \end{array} \right\}$$

## Verification

Verify  $\pi$  and check  $e(A', w) = e(\bar{A}, g_2)$ .

## Verification

Verify  $\pi, \pi'$  and check  $e(A', w) = e(\bar{A}, g_2)$ .

We modify the PoK for BBS+ signatures from [CDL16](left) to joint PoK (right), where the first attribute is known only to the card, while holder just knows commitment to it.

We use common randomness to essentially decompose the original proof as two separate proofs supplied by card and holder.



# POK for BBS+ Signature [CDL16]

# Joint POK for BBS+ Signatures

## Inputs

- **Prover's Inputs:**  $m_1, \dots, m_\ell, (A, e, s)$  satisfying:  $A = (g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i})^{1/(e+x)}$ .
- **Verification Predicate:**  $m_i = v_i$  for  $i \in V \subseteq \{2, \dots, \ell\}$ .

## Inputs

- **Card's Inputs:**  $m_1$ .
- **Holder's Inputs:**  $m_2, \dots, m_\ell, (A, e, s)$ , satisfying  $A = (g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i})^{1/(e+x)}$ ,  $Q = h_1^{m_1}$ .
- **Verification Predicate:**  $m_i = v_i$  for  $i \in V \subseteq \{2, \dots, \ell\}$ .
- **Common Randomness:**  $r \leftarrow \mathbb{Z}_p$ .

## Proof Generation

- **Compute Statement:**
  1.  $r_1 \leftarrow \mathbb{Z}_p, r_2 \leftarrow \mathbb{Z}_p, r_3 = r_1^{-1}, b = g_1 h_0^s \prod_{i=1}^\ell h_i^{m_i}, s' = s - r_2 r_3$ .
  2.  $A' = A^{r_1}, d = b^{r_1} h_0^{-r_2}, \bar{A} = A'^{-e} b^{r_1}$ .

- **Compute Proof:**

$$\pi := \text{PK} \left\{ \begin{array}{l} (\{m_i\}_{i \in H}, e, s', r_2, r_3) : \\ A'^{-e} h_0^{r_2} = \bar{A}/d \wedge \\ d^{-r_3} h_0^{s'} \prod_{i \notin V} h_i^{m_i} = g_1^{-1} \prod_{i \in V} h_i^{-m_i} \end{array} \right\}$$

## Proof Generation

- **Compute Statement:**
  1. (Holder):  $r_1 \leftarrow \mathbb{Z}_p, r_2 \leftarrow \mathbb{Z}_p, r_3 = r_1^{-1}, b = g_1 h_0^s \cdot Q \cdot \prod_{i=2}^\ell h_i^{m_i}, s' = s - r_2 r_3 - r$ .
  2. (Holder):  $A' = A^{r_1}, d = b^{r_1} h_0^{-r_2}, \bar{A} = A'^{-e} b^{r_1}$ .
  3. (Card):  $B = h_1^{m_1} h_0^r$ .

- **Compute Proof:**

$$\pi := \text{PK} \{(m_1, r) : h_1^{m_1} h_0^r = B\}$$
$$\pi' := \text{PK} \left\{ \begin{array}{l} (\{m_i : 1 \neq i \notin V\}, e, s', r_2, r_3) : \\ A'^{-e} h_0^{r_2} = \bar{A}/d \wedge \\ d^{-r_3} h_0^{s'} \prod_{1 \neq i \notin V} h_i^{m_i} = g_1^{-1} B^{-1} \prod_{i \in V} h_i^{-m_i} \end{array} \right\}$$

## Verification

Verify  $\pi$  and check  $e(A', w) = e(\bar{A}, g_2)$ .

## Verification

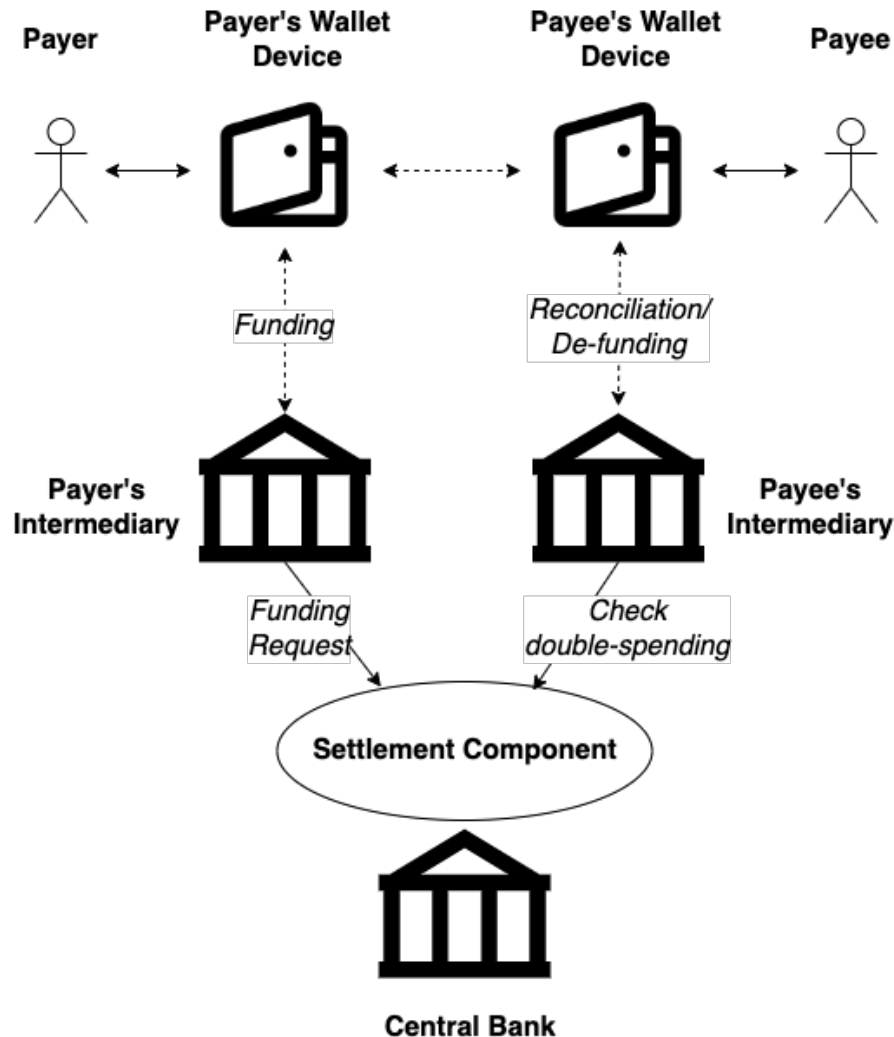
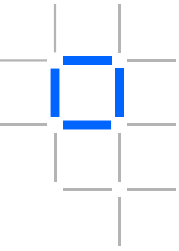
Verify  $\pi, \pi'$  and check  $e(A', w) = e(\bar{A}, g_2)$ .

We modify the PoK for BBS+ signatures from [CDL16](left) to joint PoK (right), where the first attribute is known only to the card, while holder just knows commitment to it.

We use common randomness to essentially decompose the original proof as two separate proofs supplied by card and holder.

Offline payments

# Requirements



**Security:** payments proceed as instructed by the token owners; one cannot spend more than what they own without being detected; one cannot transfer more than a certain amount in a given payment (or deposit).

**Accountability:** users should not be able to repudiate their payments if they are found to double-spend.

**(Offline) Immediate Finality:** offline payments are final without going through an online reconciliation.

**Privacy:** confidentiality of (honest) payments w.r.t. the central bank and intermediaries, and non-involved users.

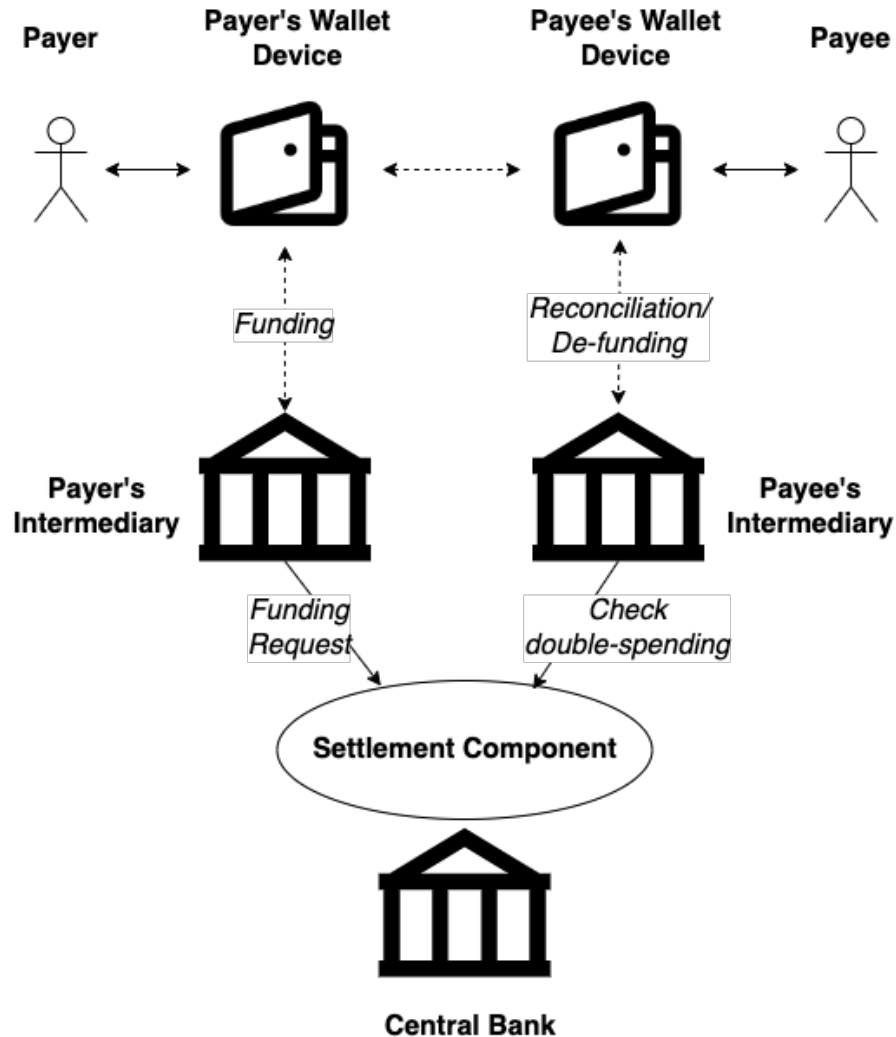
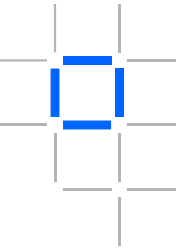
# Threat Model

## Potential Security Attacks

- A payer may try to:
  - 1) Forge coins into a payment
  - 2) Forge payment amount
  - 3) Overspend
- A payee may try to trick a payer into spending more than intended.
- Payers and payees may collude to be successful.
- Intermediaries and Central Bank trusted to:
  - grant user/device identities honestly & revoke them as per policy
  - handle funding/reconciliation requests correctly



# Our Approach: Combining Cryptography & Secure Hardware

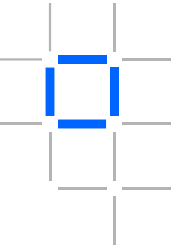


Leveraging **cryptography** to

- **authenticate users/token owners** in transactions
- detect double-spendings & **identify double-spenders**
- protect the **privacy** of individual payments
- enforce **sanctions/audits on users** when needed

Exploring **optimal combination of HW devices** to **increase the cost of a double-spending attack**

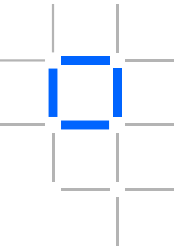
# Overview






- All users enroll with the offline service for their wallet , provisioned with identity credentials `cred` and secrets `sk`

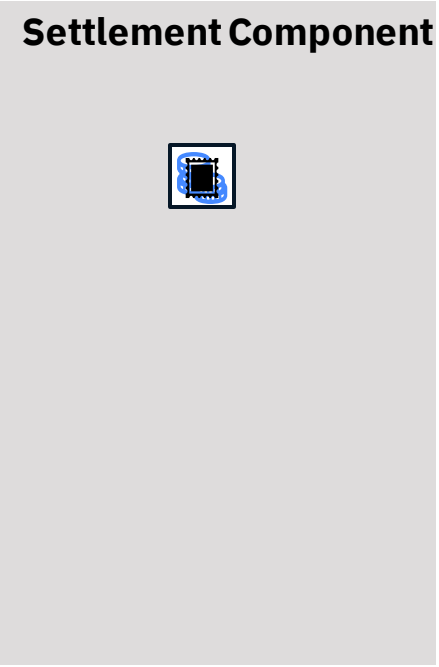
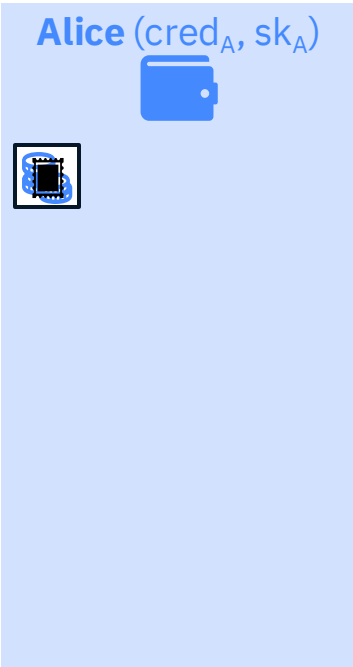
## Settlement Component

# Overview

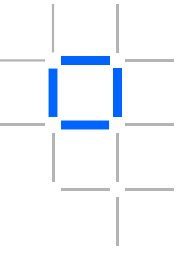







- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 

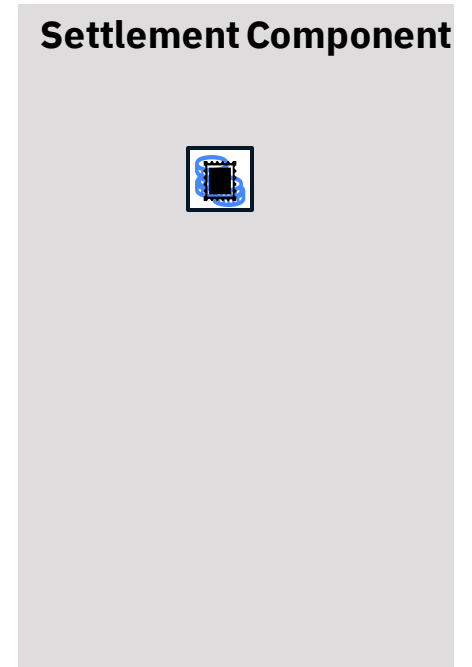
- **Issuer** to **Alice**:



# Overview

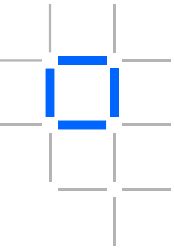







- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 

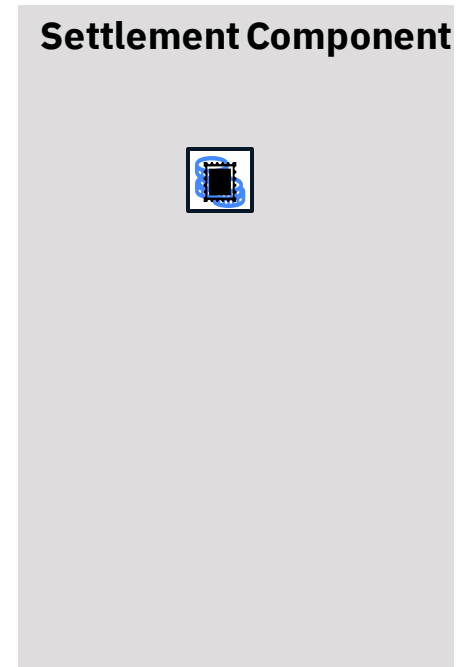
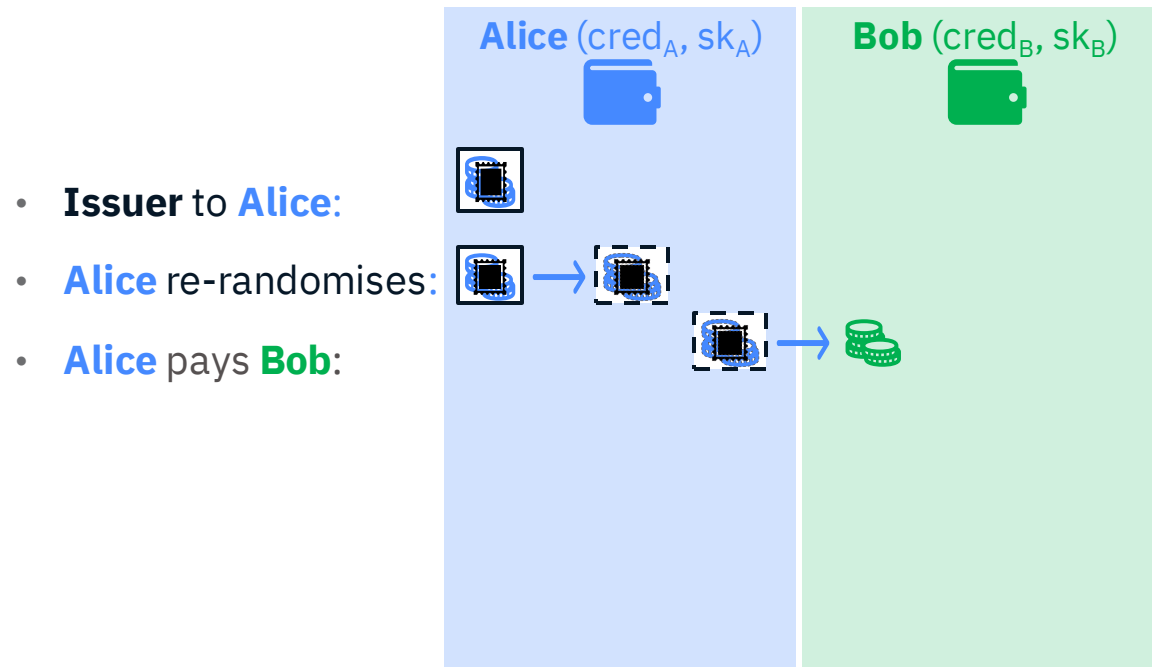




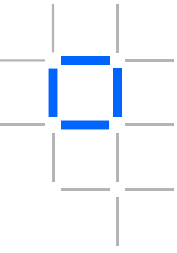
# Overview








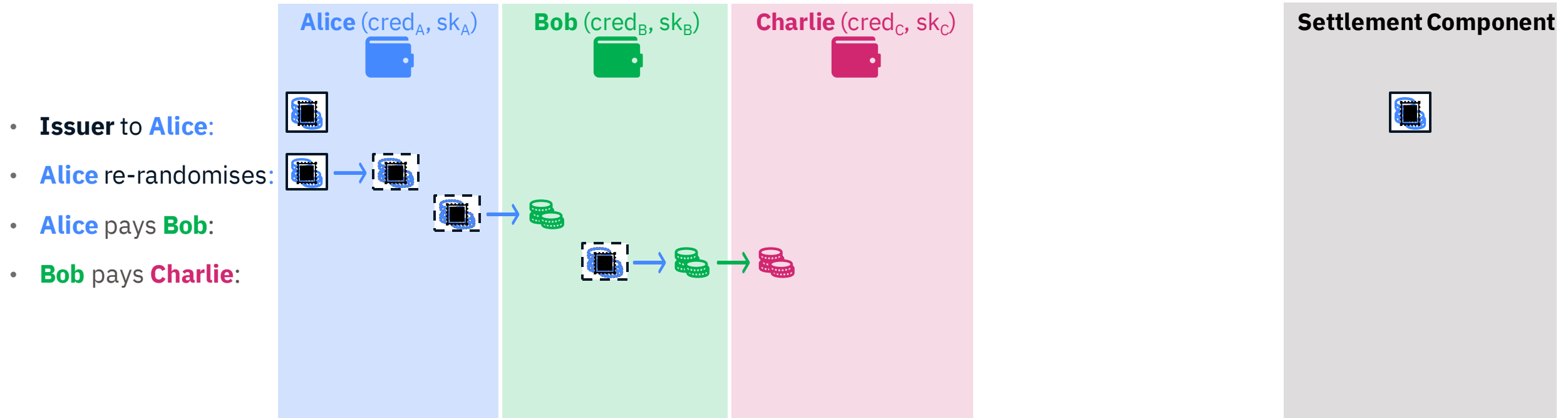
- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 



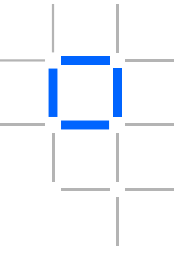
# Overview








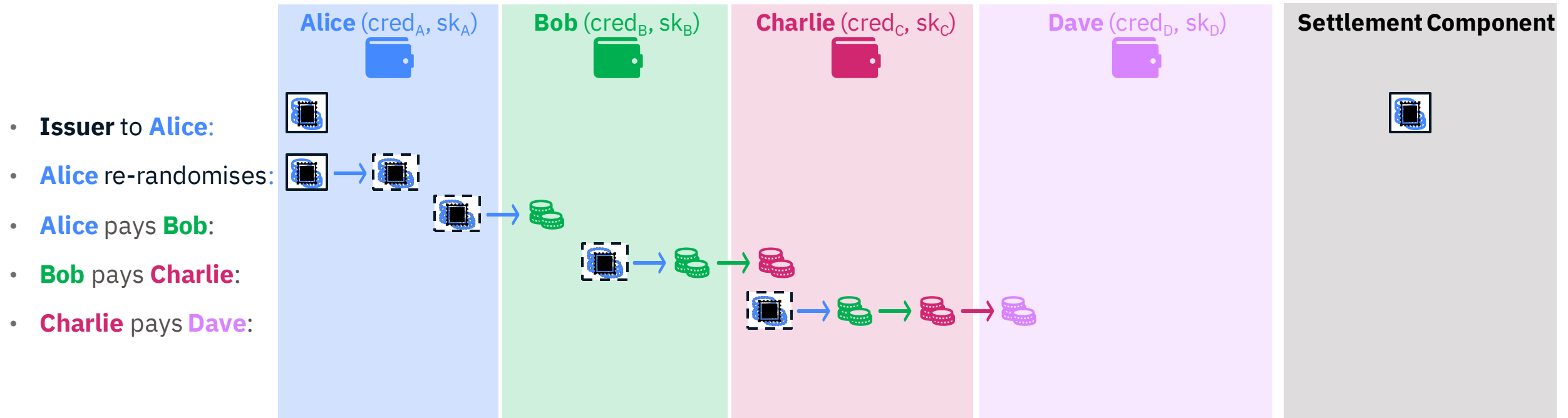
- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 



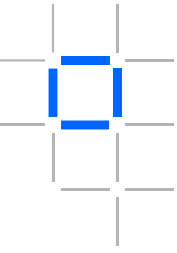
# Overview








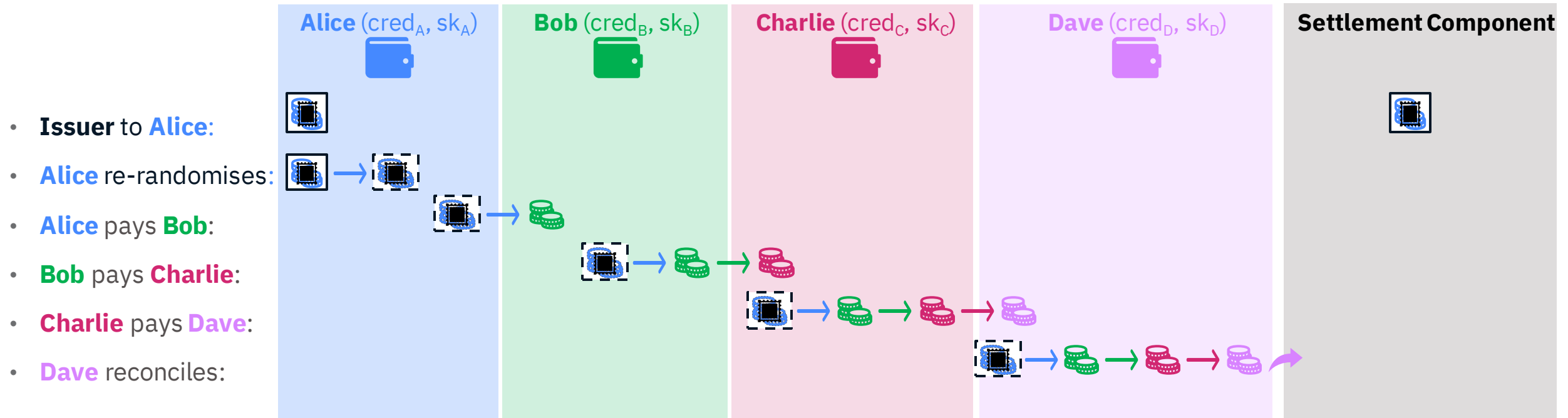
- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 



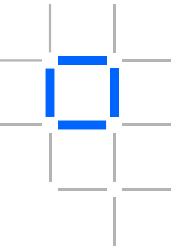
# Overview








- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 

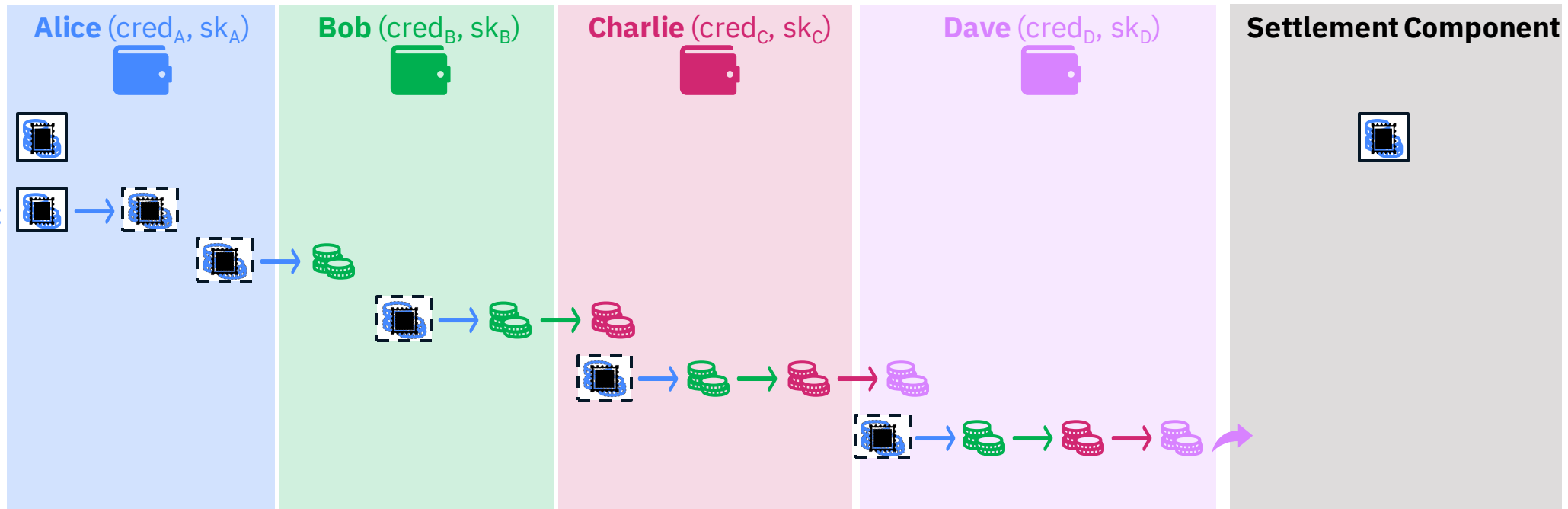


# Overview



- All users enroll with the offline service for their wallet , provisioned with identity credentials  $cred$  and secrets  $sk$
- **Alice** withdraws token of fixed value **val** from Central Bank and stores the tokens  to their wallet 
- **Alice** converts  to an unlinkable but equivalent form to spend it 

- **Issuer** to **Alice**:
- **Alice** re-randomises:
- **Alice** pays **Bob**:
- **Bob** pays **Charlie**:
- **Charlie** pays **Dave**:
- **Dave** reconciles:



Elli Androulaki, Angelo De Caro, Kaoutar Elkhayaoui, Romain Gay, Rebekah Mercer, Alessandro Sorniotti:  
Secure and Privacy-preserving CBDC Offline Payments using a Secure Element. IACR Cryptol. ePrint Arch.  
2024: 1746 (2024)

