How can we utilize verifiable credentials, self-sovereign identity and open source hyperledger technologies for a lifelong learning.

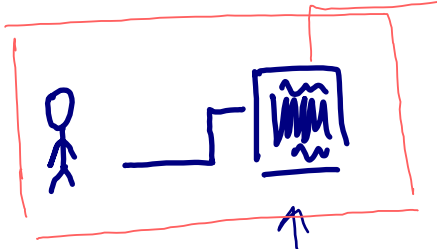
Özgül Küçük
Computer Engineering Department
İstanbul Bilgi University

28/09/21

"In this project we will propose of a software architecture which implements blockchain technology as a secure and efficient solution to keep, verify and digitalize certificates/records within a university, i.e. Bilgi University."

↳ this talk contains unfinished work on this project.

[diploma verification] → general questions



→ legal, signed, diploma OR a certificate } if lost then most of the time not possible to renew.

submits

Job Application

Notary

→ "the person with the ID .... holds a diploma / certificate from ...."

→ is it legal ?

* diploma could be fake
* you may have a certificate but can't find it
* you don't want to reveal all information about your ID.

① ← * avoid creation of fake credentials from a legal institute.

② ← * your qualifications can be attached to your identity

OR

you can create verifiable credentials from your qualifications

↳ (claims) → verifiable credentials

③ ← * verification can/should involve the source

↳ if all stakeholders have digital identities

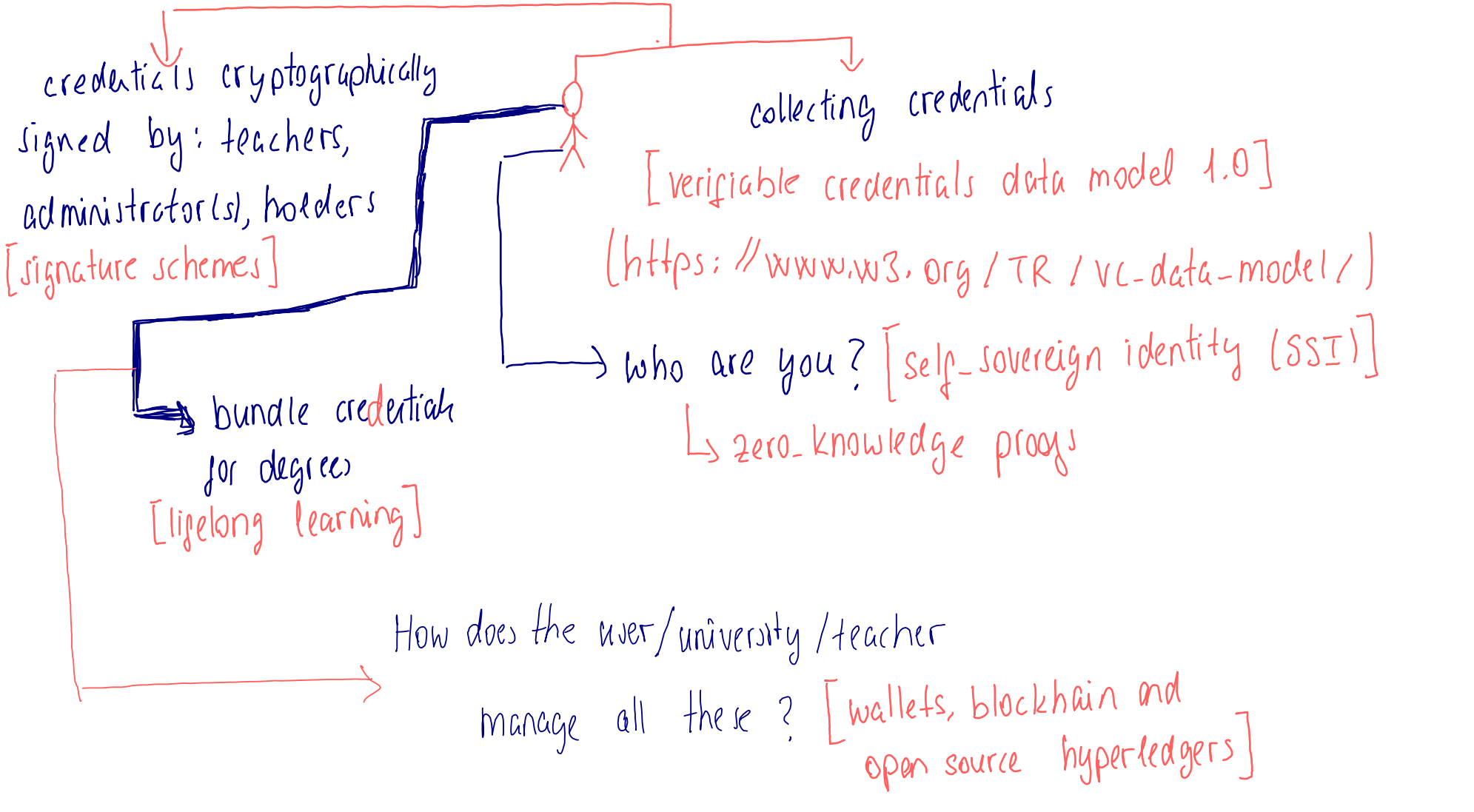
④ ← * intermediaries can be removed, verification can be efficient, secure, private

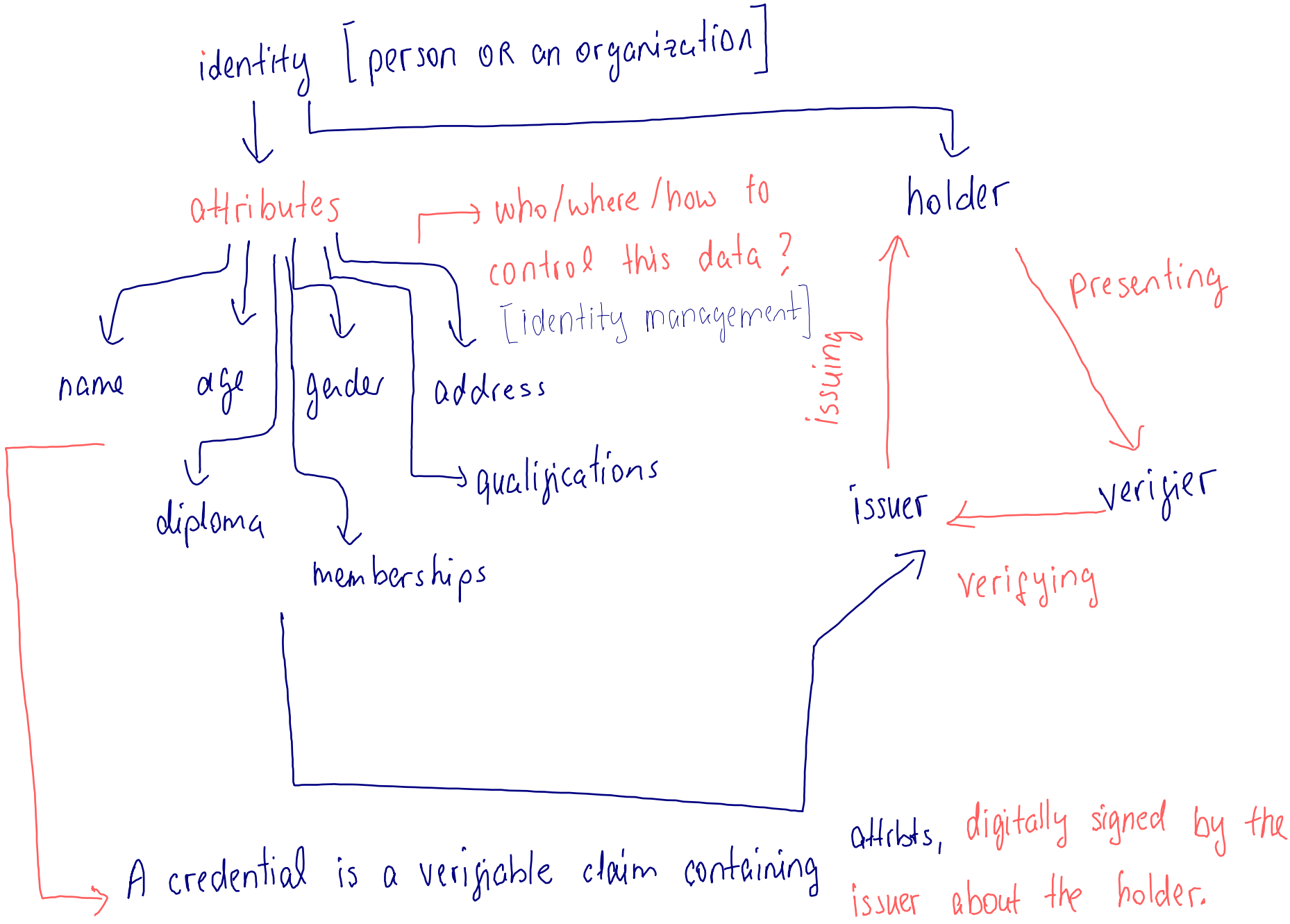
(notary etc, ...)

self-sovereign identity → decentralized, verified with cryptography

5

Digitalize, secure and improve efficiency of academic records keeping, enable lifelong learning.




identity [person OR an organization]
attributes
who/where/how to control this data?
[identity management]
holder
name
age
gender
address
diploma
memberships
qualifications
issuing
presenting
issuer
verifier
verifying
A credential is a verifiable claim containing attribts, digitally signed by the issuer about the holder.

with

self-sovereign identity

↳ holders have the control over their identities

collect signed credentials

present proofs of claims

(without the intermediary)

↳ 12 principles of SSI realized by

Indy, Ursa and Aries

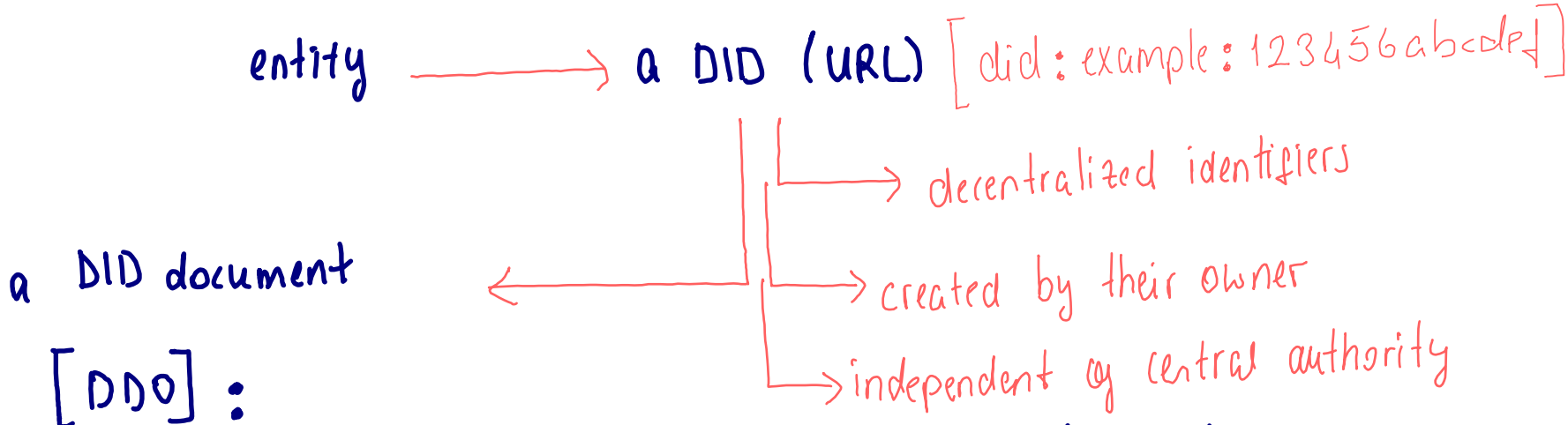
↳ system design

inner components

self-sovereign identity is implemented through decentralized identifiers (DIDs) → a uniform resource locator

↳ as software projects

↳ hyperledger identity projects INDY, URSA and ARIES implements decentralized identity (SSI) and verifiable credentials (VC-data model)



context ⟶ system environment for information exchange between DIDs

id ⟶ DID of this DDO

public key ⟶ (digital signatures, cryptographic operations)

service endpoint ⟶ interactions among DIDs

How does a DID works ?

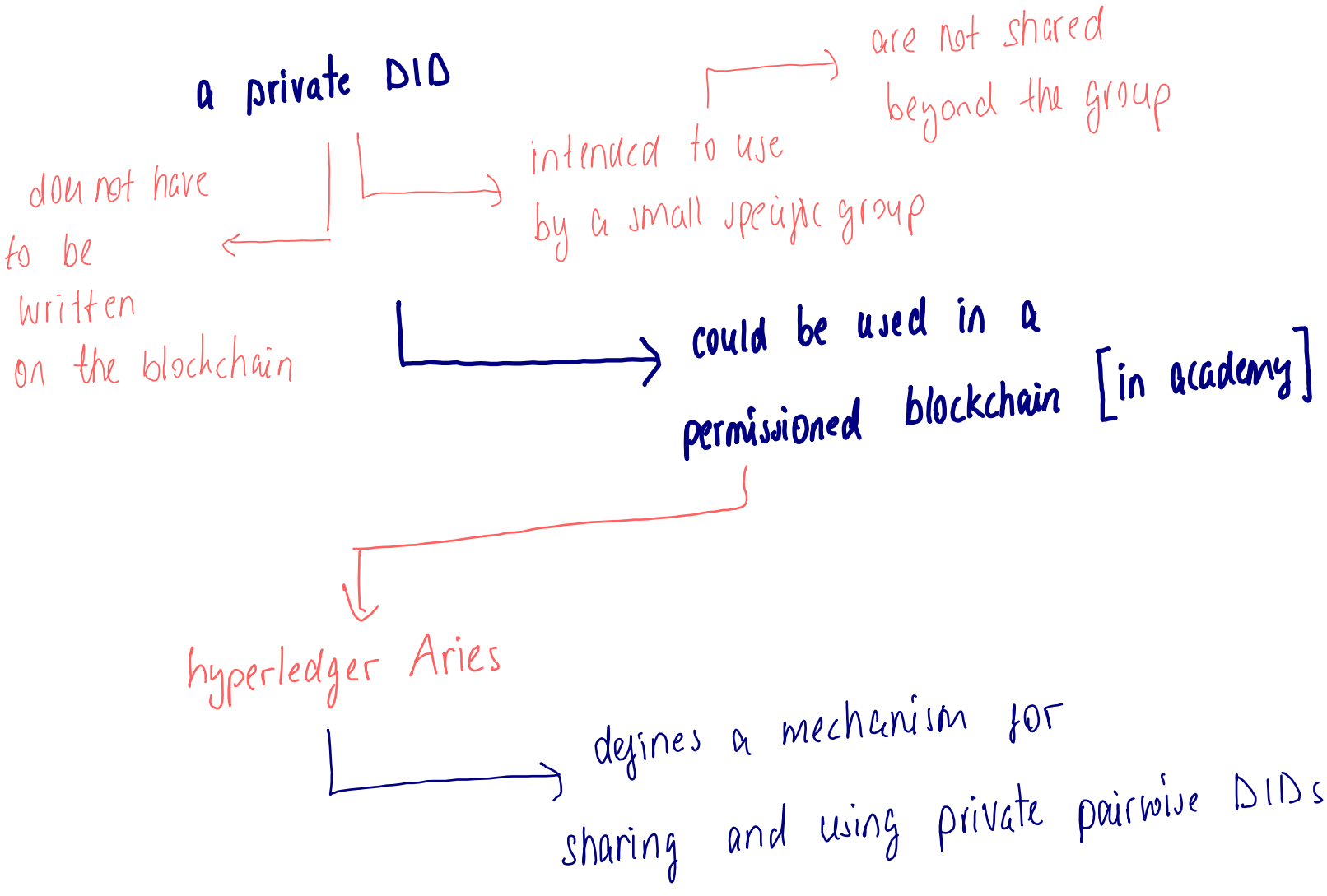
a public DID [of a verifiable credential issuer]

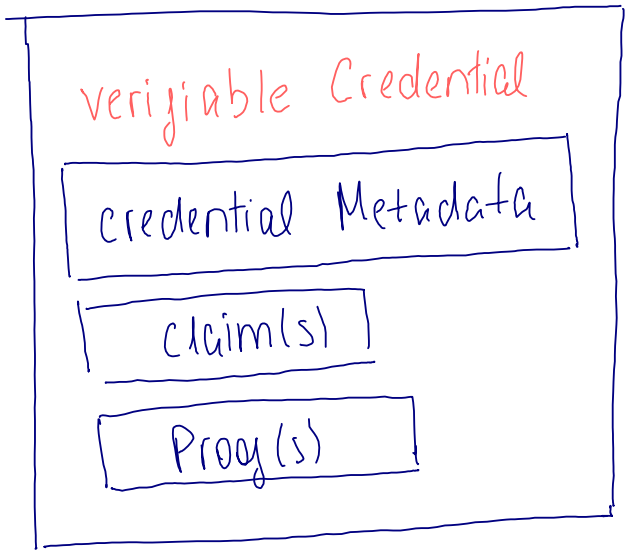
holder Of A Credential

→ verifier [trusts the issuer]

presents the public DID of the issuer (literally to anyone)

→ should be written on a public blockchain [which is costly]

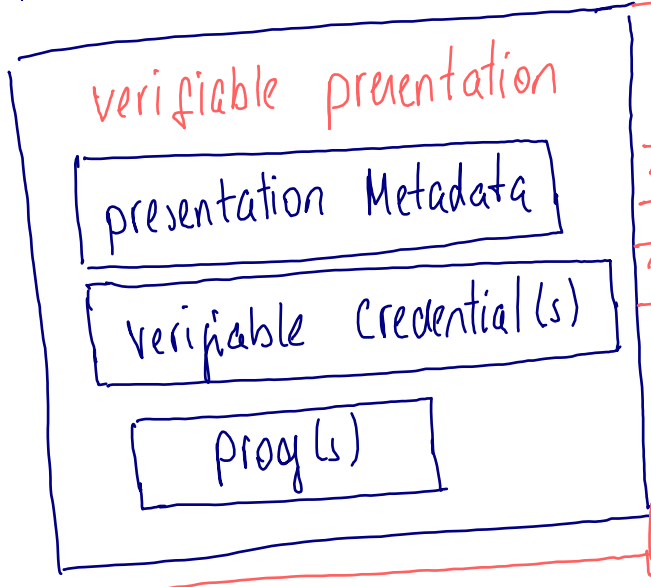

a private DID
intended to use by a small specific group
are not shared beyond the group
does not have to be written on the blockchain
could be used in a permissioned blockchain [in academy]
hyperledger Aries
defines a mechanism for sharing and using private pairwise DIDs



reference:
https://www.w3.org/TR/vc-data-model

credential graph

Reference:
https://www.w3.org/TR/vc-data-model/#claims



lifecycle:

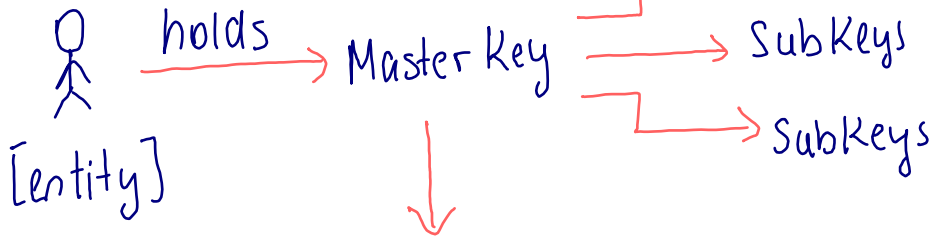
1. issue credentials
2. store verifiable cred. in a credential rep. [digital wallets]
3. compose . . . in a verifiable presentation

4. verification of the verifiable presentation by the verifier

Design Patterns

* blockchain based self-sovereign identity [1]
  * key management
  * DID management
  * credential design
* smart contracts and blockchain applications [3]
* smart contracts and security patterns [2]
  * Checks-Effects-Interaction
  * Emergency Stop
  * Speed Bump
  * Rate Limit
  * Balance Limit
  * Token
  * Authorization
  * Oracle
    * Math
    * Fork check
  * Randomness
  * Poll
  * Time Constraint
  * Termination

Key Management Patterns [1]



use to sign different identities (enrollments)

Master and SubKey generation [1]

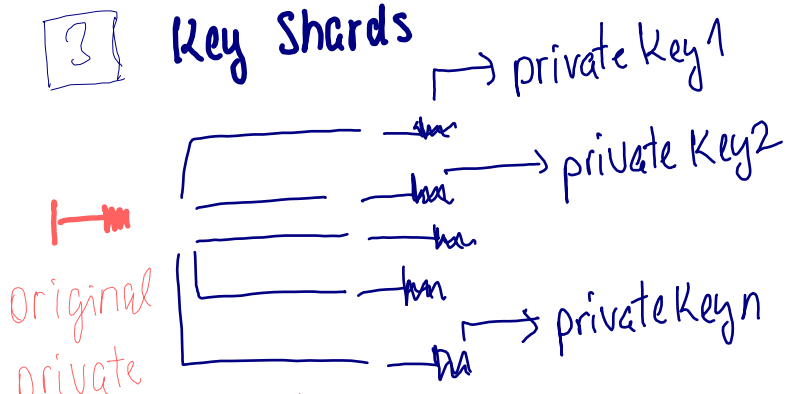
If LOST (?)

[2] Hot and Cold Walled Storage

- online generation and update of keys
  - ↳ online theft
- offline storage, connect to internet to use

→ combine

[3] Key Shards

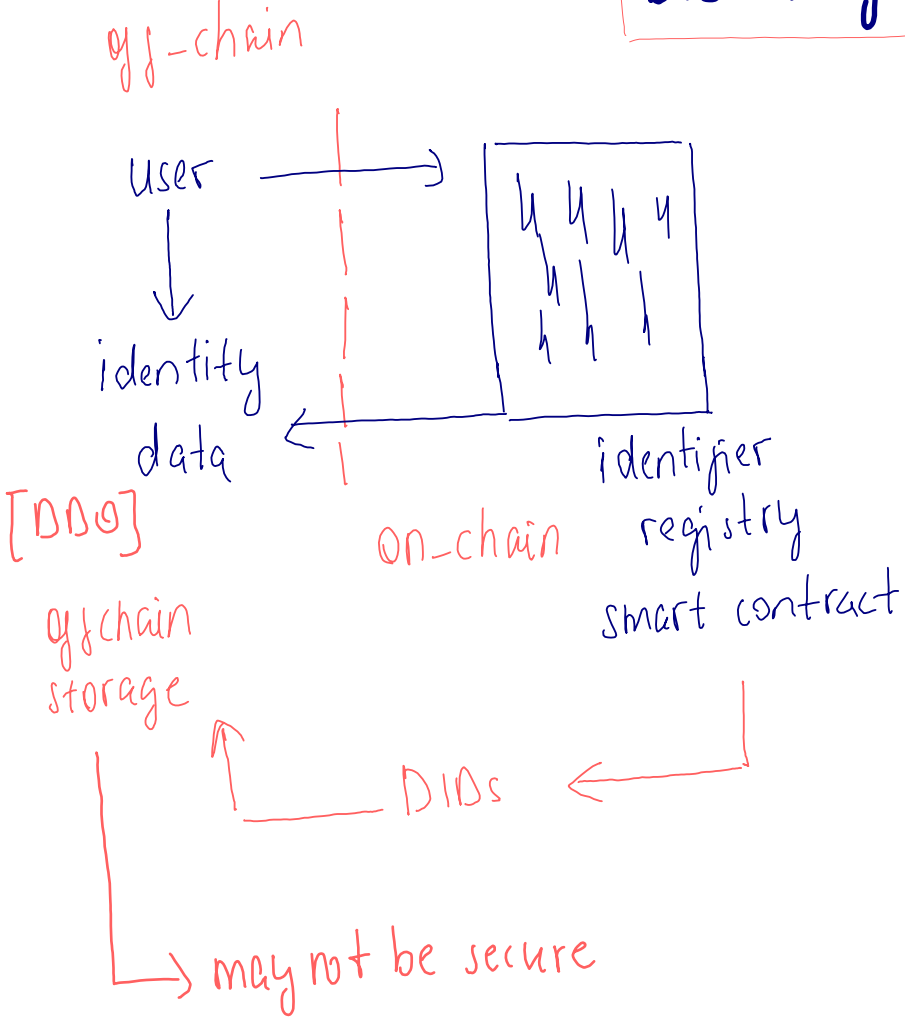


regroup by threshold cryptography

key sharding
- ↳ can be stored in decentralized way
- ↳ secure maintain is a drawback

* Crypto++ → implements Shamir's SS.

DID Management Patterns



identifier Registry

→ ∃ some other design patterns, not relevant (multiple registration, blockchain & social media account pair, ...)

* example: Sovrin uses Sovrin protocol for registration, update, resolution, revocation,

Credential Design Patterns

[1] Selective Content Generation

* Atomic credentials,

  └→ multiple credentials
  [each containing one identity signed individually]

* Selective disclosure signatures

  A general credential

  └→ with special signature schemes ⟶ revealing only necessary information

  (such as Camenisch-Lysyanskaya)

Credential Design Patterns

* Hashed values: A general credential
  → of multiple attributes
  each hashed with
  a different nonce
  → a verifier can validate only those with the actual hash values.

* Zero-knowledge proofs:
  A holder can prove that a credential is within certain values.

Time_Constrained Access

holder

↳ share a link to the credential

valid only certain period of time

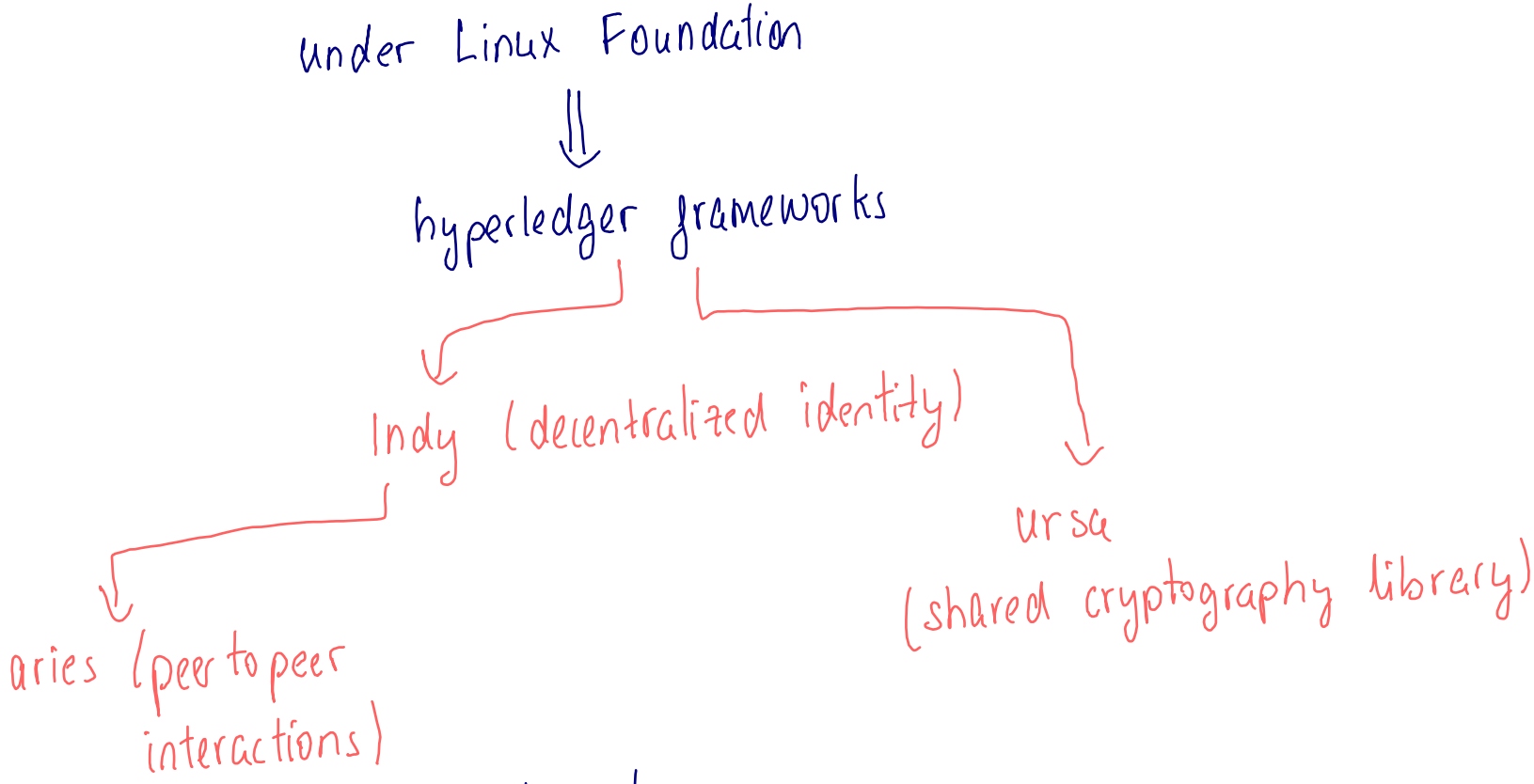
DIDs and verifiable credentials

ARIES ⟶ implementation

[issuers, holders/provers, verifiers] ⟶ secure, point-to-point messaging between agents & participants

Wallets are digital wallets

- ⟶ have secure storage
- ⟶ communicate with each other following Aries protocols

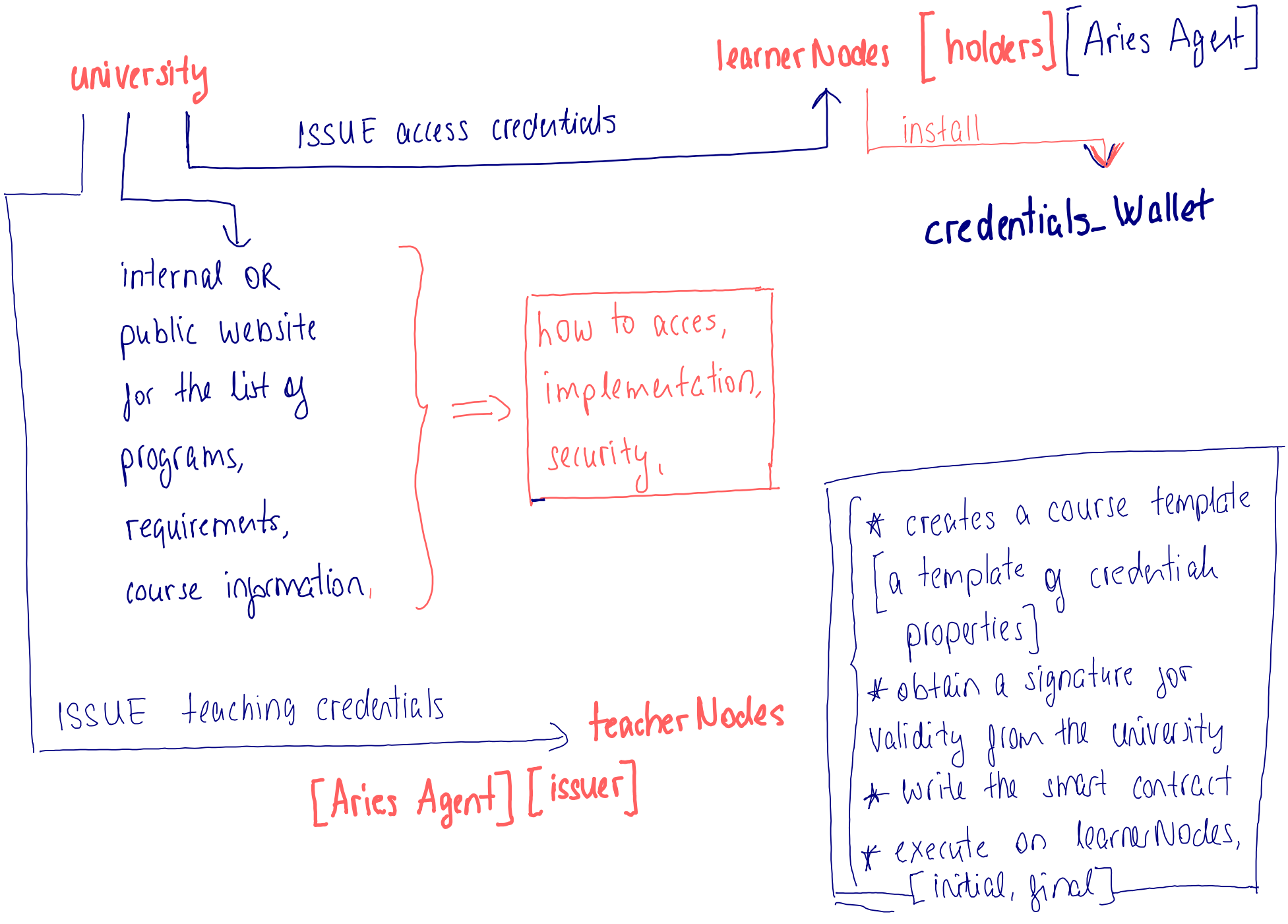


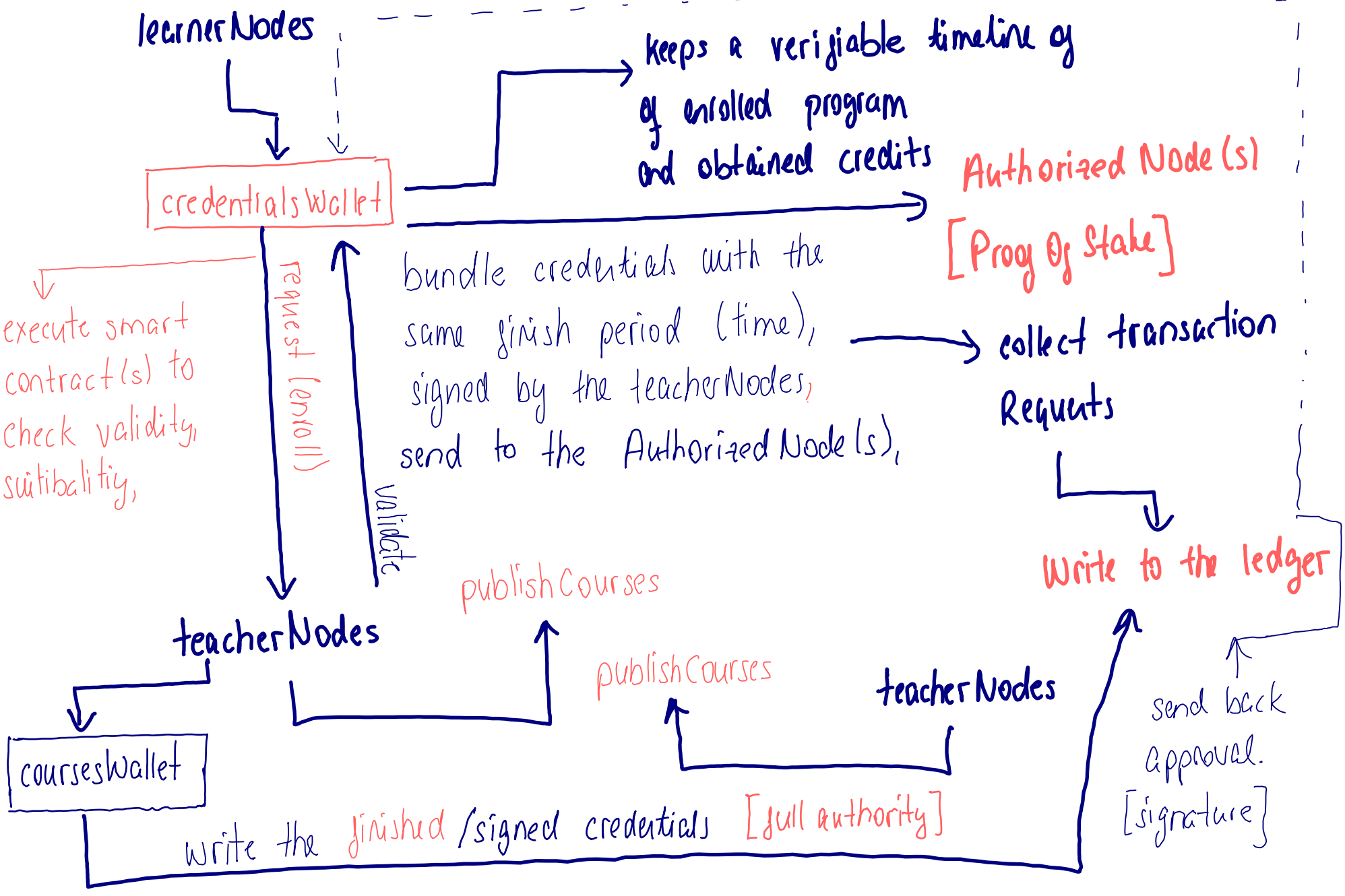
implements

verifiable credentials model,
DIDs,
agents,
zero_knowledge proofs,
selective disclosure

Objectives :

* Students manage / own their education process
  [similar to SSI principles, in the context of higher education]
* Enable to create / follow course (bundles) from different organizations,
  to obtain a degree.
* Transforming course credentials to a verifiable architecture
* Using smart contracts, improving verifiable credentials data model,
  leveraging SSI for identity management, opensource hyperledger technologies.
* privacy and security : public key cryp., zero_knowledge proofs, protocol security
  data security, securing smart contracts.


university
learner Nodes [holders] [Aries Agent]
ISSUE access credentials
install
credentials_Wallet
internal OR
public website
for the list of
programs,
requirements,
course information.
how to acces,
implementation,
security.
* creates a course template
[a template of credential
properties]
* obtain a signature for
validity from the university
* write the smart contract
* execute on learnerNodes,
[initial, final]
ISSUE teaching credentials
teacher Nodes
[Aries Agent] [issuer]


learnerNodes
keeps a verifiable timeline of
of enrolled program
and obtained credits
credentialsWallet
Authorized Node(s)
[Proof Of Stake]
execute smart
contract(s) to
check validity,
suitibality,
request (enroll)
validate
bundle credentials with the
same finish period (time),
signed by the teacherNodes,
send to the Authorized Node(s),
collect transaction
Requests
Write to the ledger
publishCourses
teacherNodes
publishCourses
teacherNodes
send back
approval.
[signature]
coursesWallet
write the finished/signed credentials [full authority]

Nodes [from a learner(s) perspective]

* teacher Nodes → may be from different universities → write to the ledgers of their university OR a government ledger

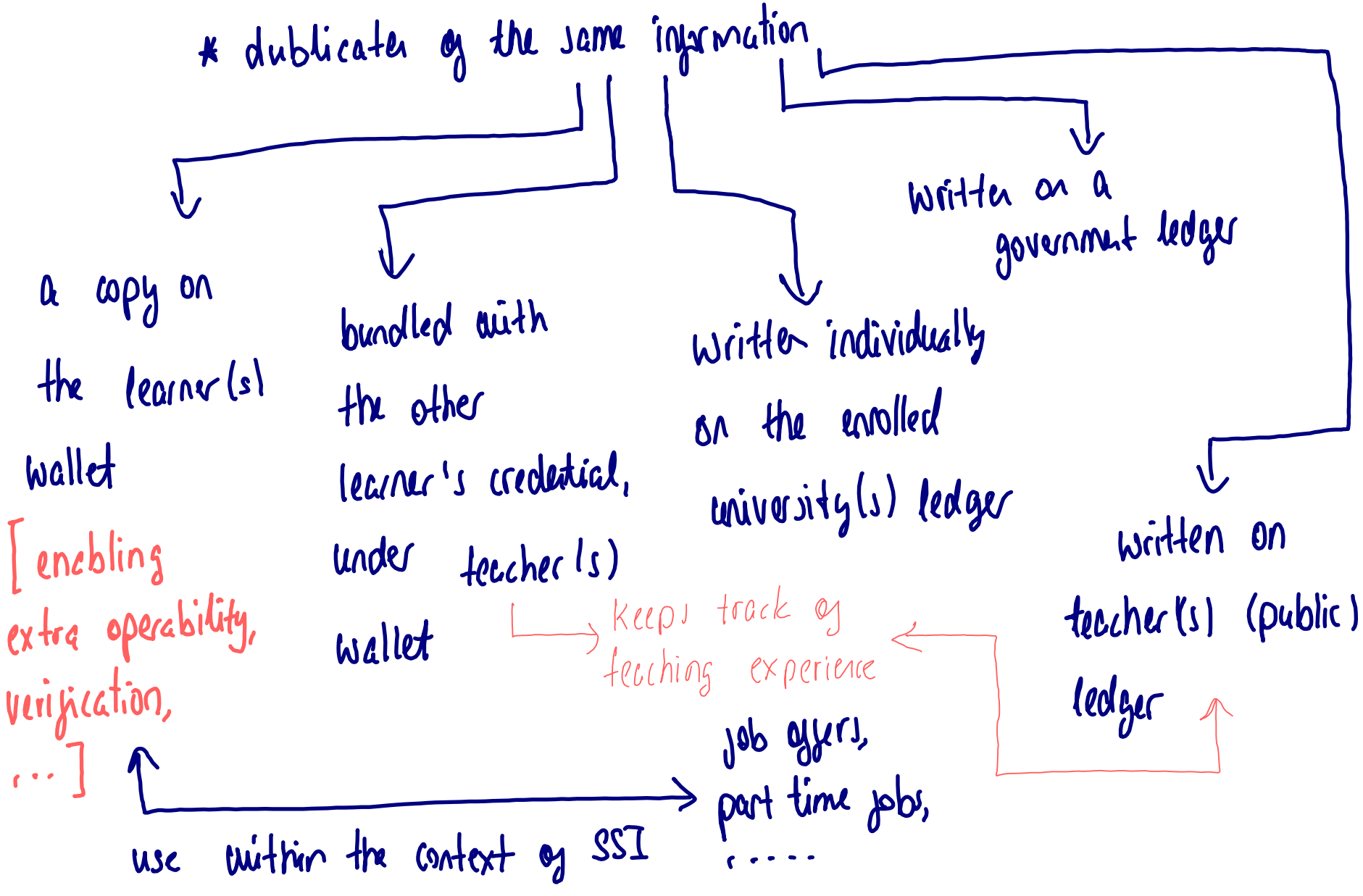
* learner Nodes → send obtained credentials to Authorized Nodes → of their enrolled universities

learner Nodes → independent learners:

keep a copy of earned credentials,

write to a public ledger,

[could be a ledger hosted by the government]


* dublicates of the same information
a copy on the learner(s) wallet
[enabling extra operability, verification, ...]
bundled with the other learner's credential, under teacher(s) wallet
written individually on the enrolled university(s) ledger
written on a government ledger
written on teacher(s) (public) ledger
keeps track of teaching experience
job offers, part time jobs, ...
use within the context of SSI

References

[1] Design Patterns for Blockchain-based Self-Sovereign Identity
Y. Liu, H.Y. Paik, X. Xu and Q. Lu

[2] Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity
M. Wöhrer and U. Zdun

[3] An Emprical Analysis of smart contracts: platforms, applications and design patterns.
M. Bartoletti and L. Pompianu

[4]

smart contract security

issues related to GDPR (general data protection regulation of the EU)

+ other issues related to security and privacy