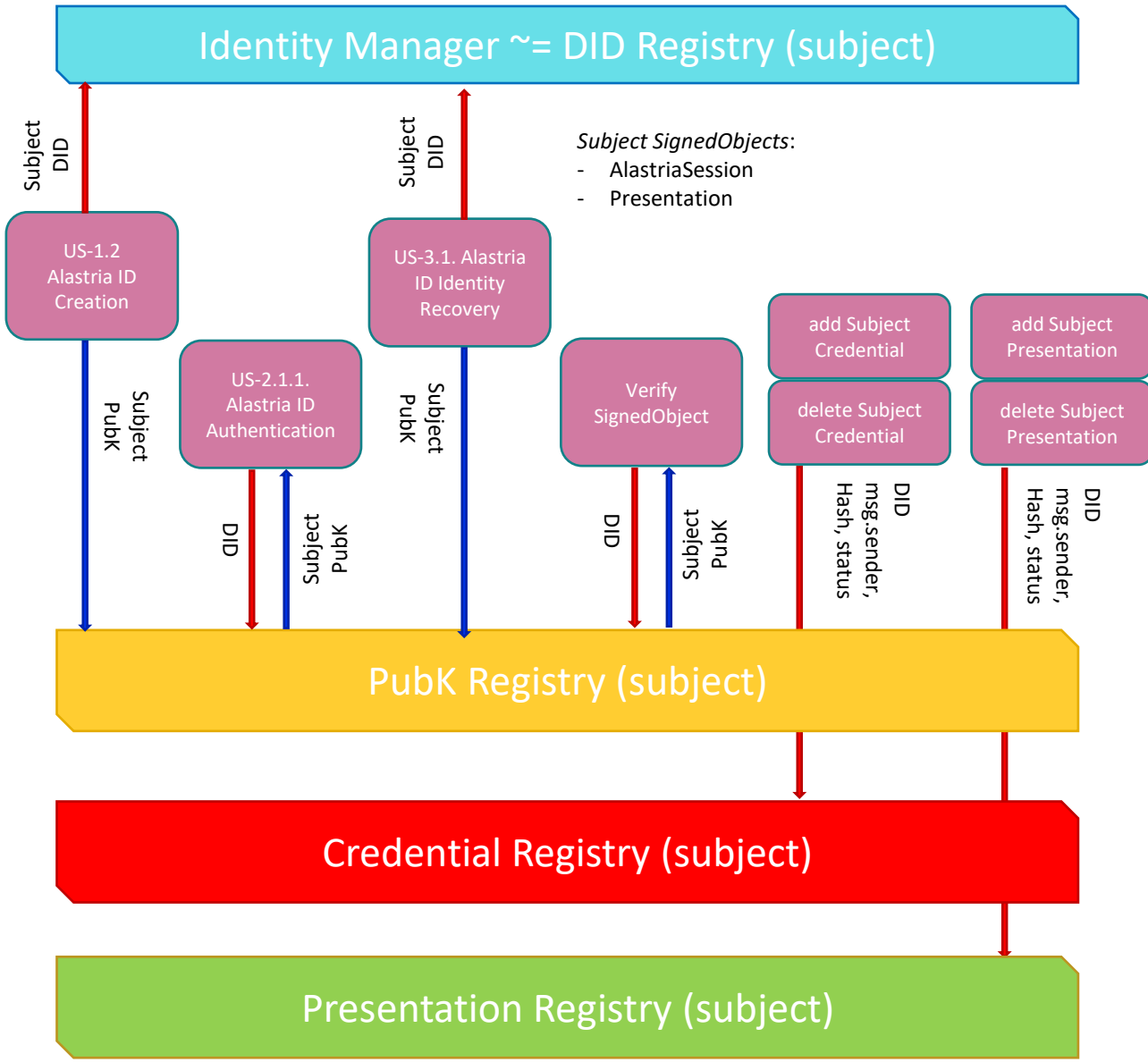
Blockchain

Alastria EPIC

Rationale

GDPR problems at Alastria ID and other Blockchain based SSI

- Alastria ID SSI model relies on registering relevant information regarding the different actors (Issuers, Subjects, Service Providers) and the identity activity (Credential Registry, Presentation Registry).
- Albeit initially no personal data is registered (credentials and presentations are handled off-chain) the identifiers (DID) act as pseudonymous of the users.
- That's why they can be considered protected data, under GDPR. Writing them in an immutable system seems not a good idea.



Rationale

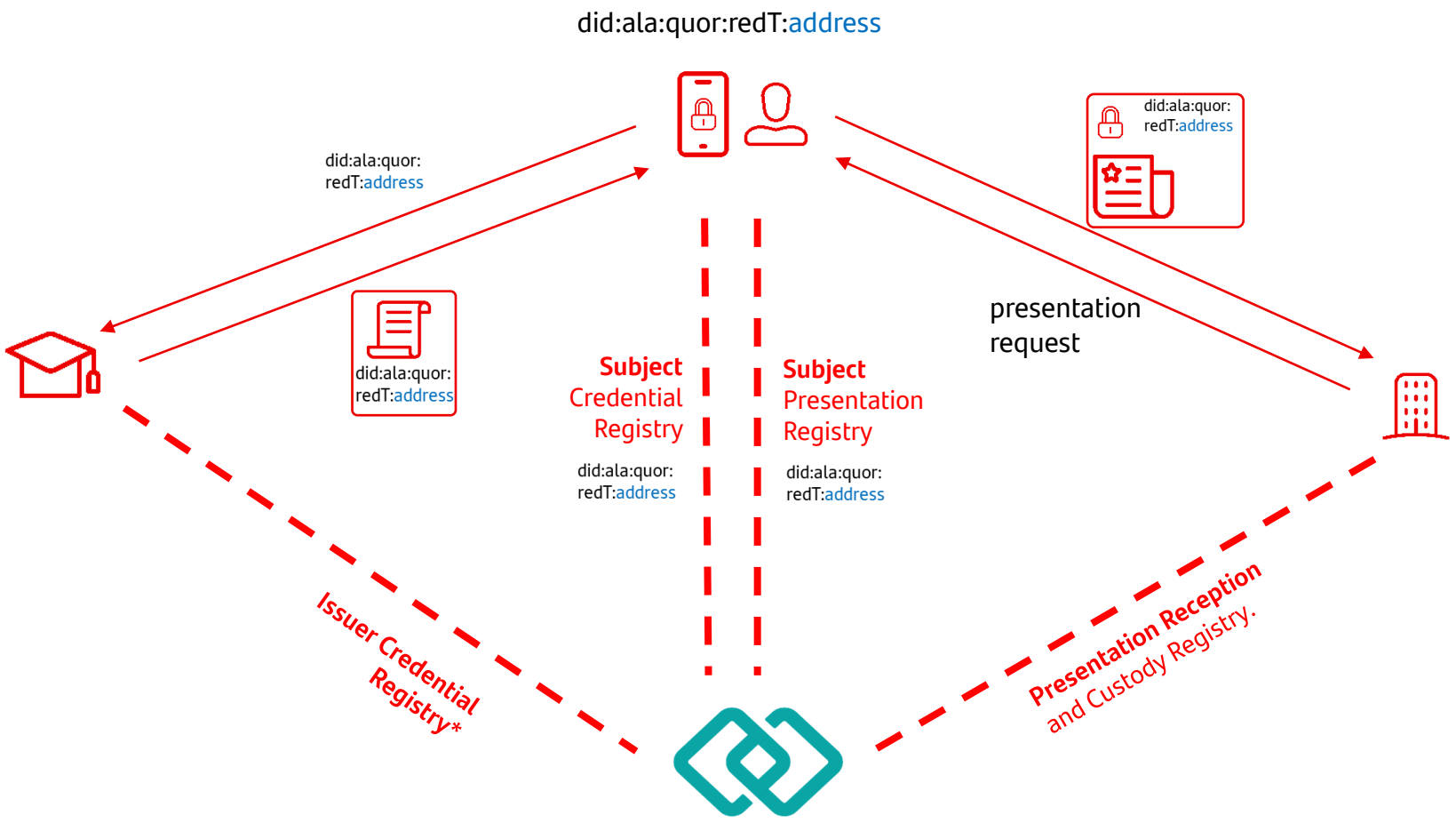
SSI vs GDPR

The only user personal information suitable to be **recorded** in a Blockchain network are **revocations** as the benefits exceed the drawbacks of personal information being recorded.

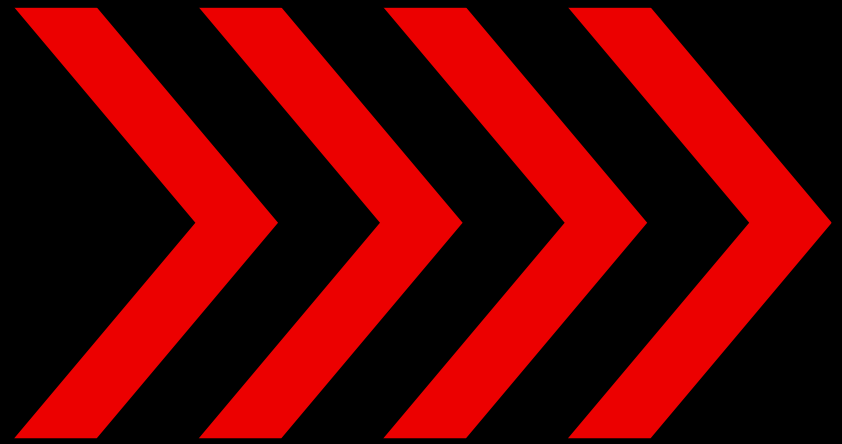
Not recording Public Keys requires whole new approach to Credential verification.

It would be also advisable for the user to have **multiple identifiers** so he can kind-of obfuscate his information.

Credential & presentations (currently)



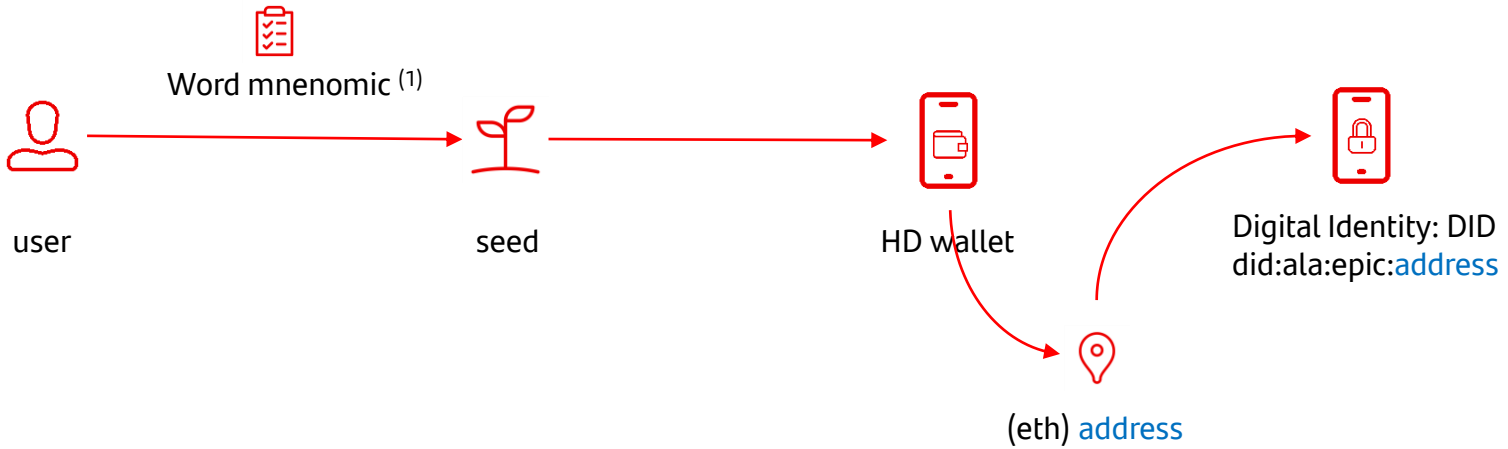
New Cryptography



Wallet and DID generation

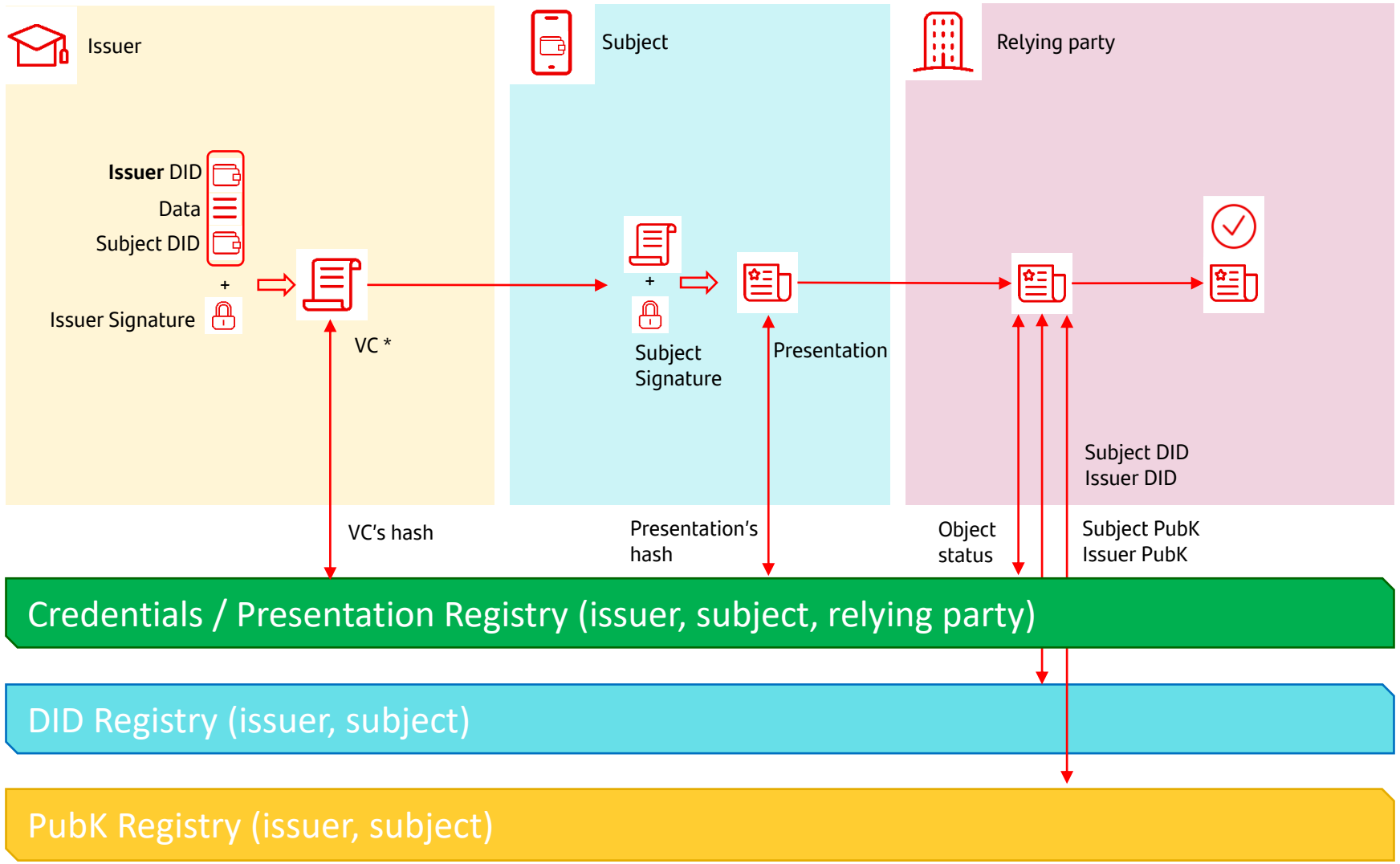
The DID is the central user identifier, therefore an alias of the user and it never should leak into Blockchain.

In the other hand it could be the root of all the other derived identifiers if HD Wallets used.



(1) https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

Object validation (original)

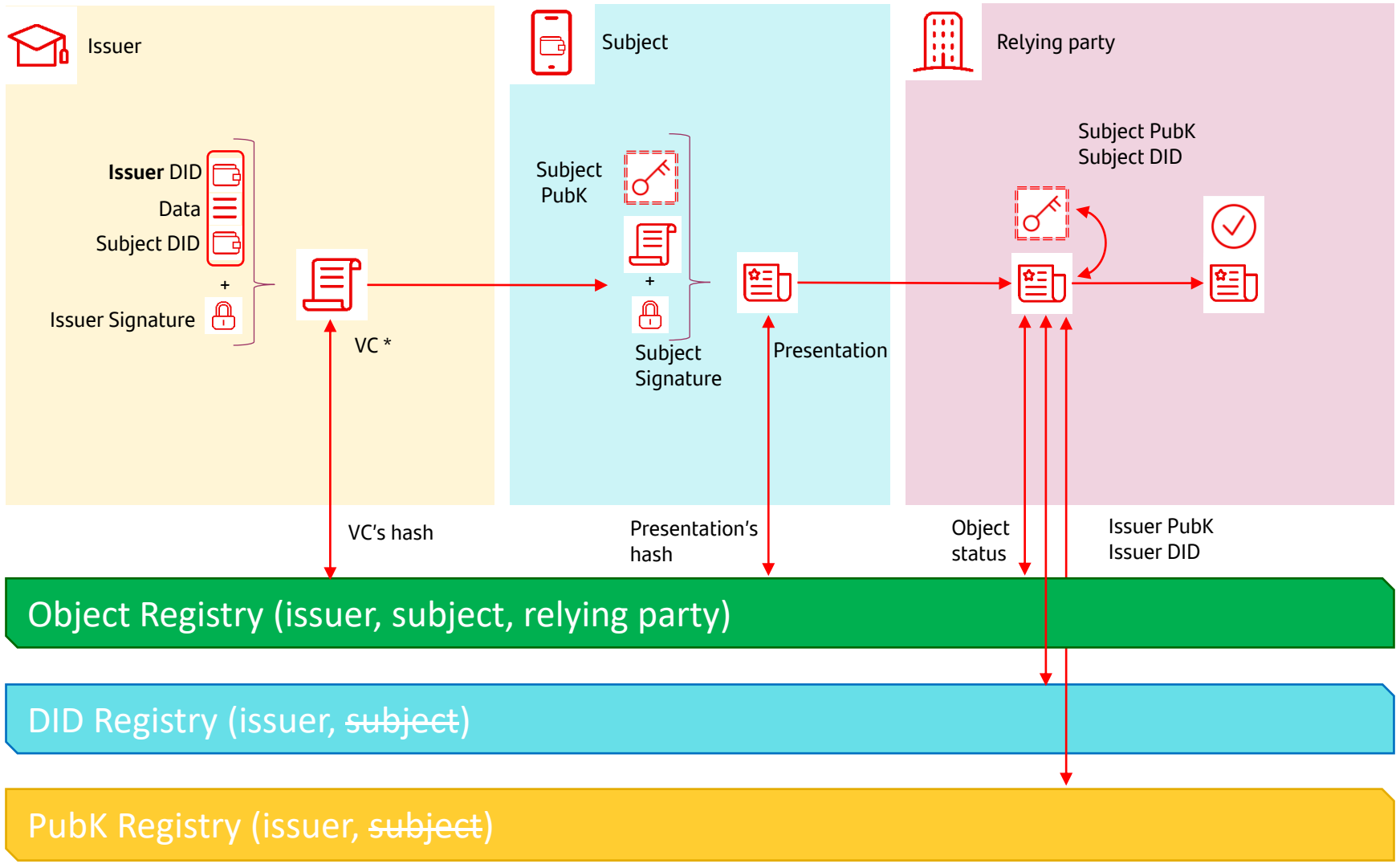


(*) Agnostic to VC schema/data model

PRIVACY

Block›chain

Object validation without subject PubK registry



(*) Agnostic to VC schema/data model

Here comes the derivations

What if form a Primordial Secret we could create some other Secrets that could be controlled by themselves and the Primordial Secret?

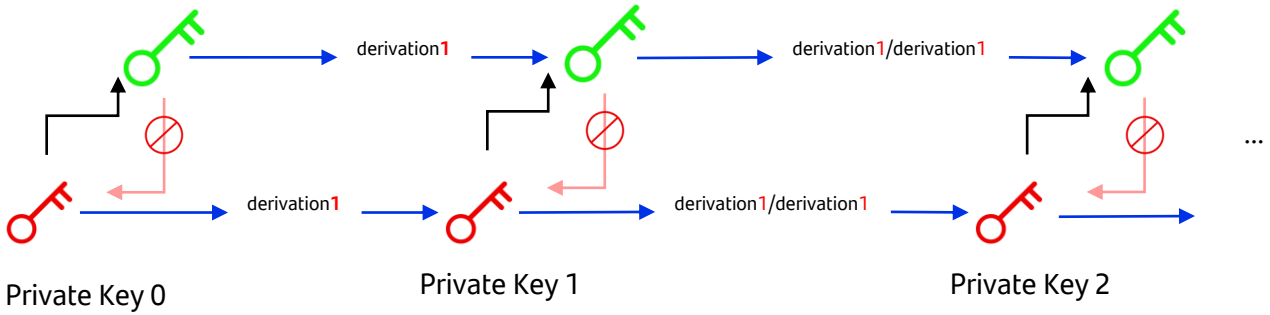
Those are the hierarchical deterministic (HD) wallets as described in BIP-32:

https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

We can generate a Master Private Key, from it create Derived Private Keys and Derived Public Keys.

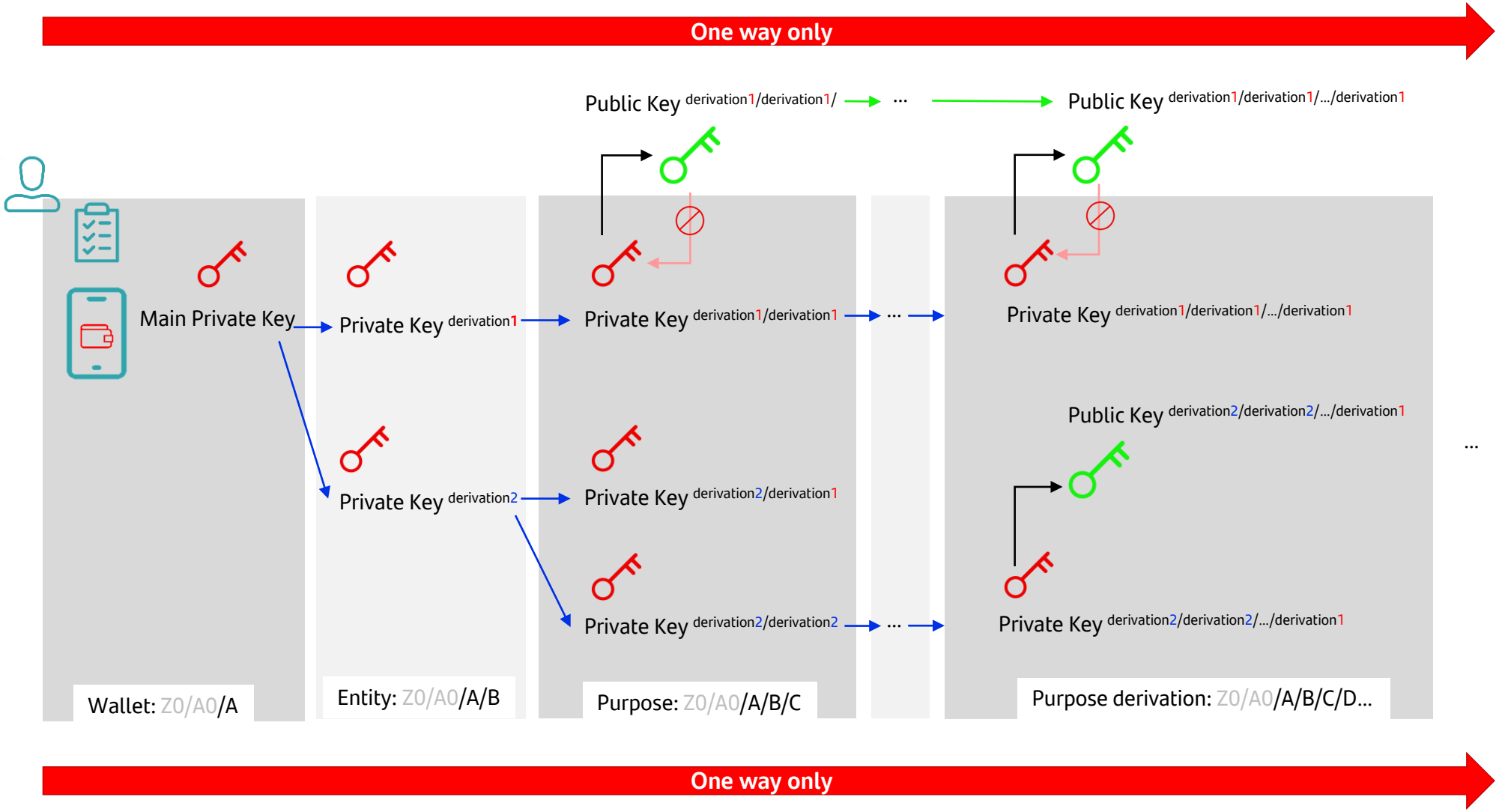
Using only “Normal” derivations where $i<2^{31}$ that allows both Private Key and Public Key derivations

HD Wallets in action



One way only

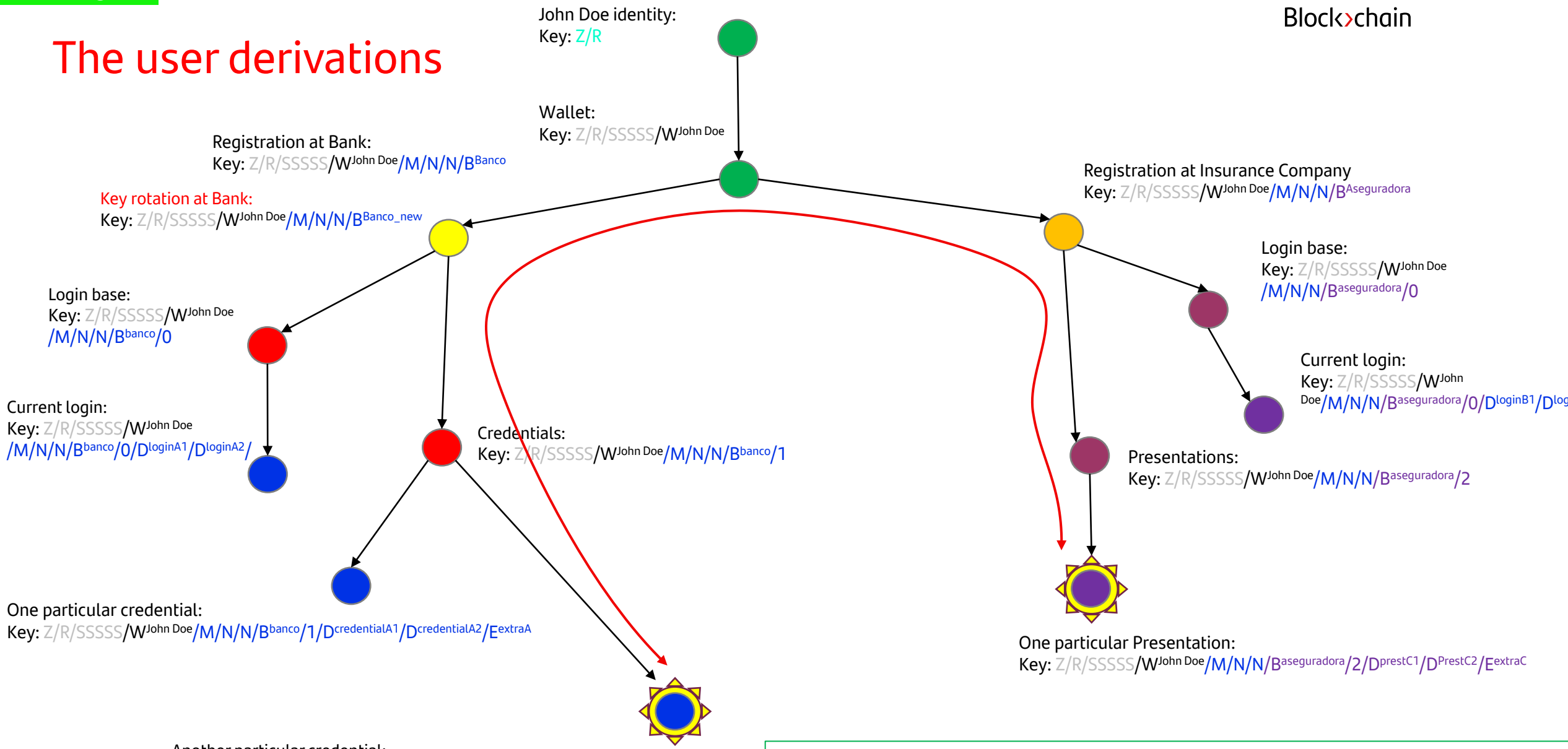
The keys


One way only
Public Key derivation1/derivation1/ ··· Public Key derivation1/derivation1/.../derivation1
Main Private Key
Private Key derivation1
Private Key derivation1/derivation1
Private Key derivation1/derivation1/.../derivation1
Private Key derivation2
Private Key derivation2/derivation1
Public Key derivation2/derivation2/.../derivation1
Private Key derivation2/derivation2
Private Key derivation2/derivation2/.../derivation1
Wallet: Z0/A0/A
Entity: Z0/A0/A/B
Purpose: Z0/A0/A/B/C
Purpose derivation: Z0/A0/A/B/C/D...
One way only

PRIVACY

Block›chain

The user derivations



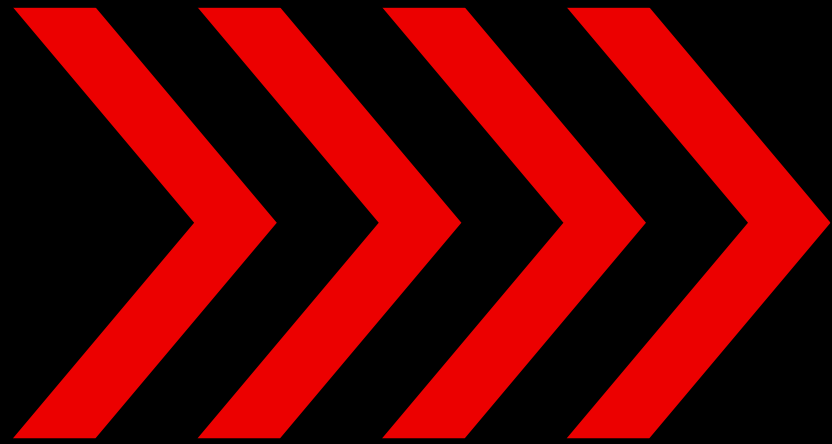
Presentation to Aseguradora:

signed by: Key: Z/R/SSSSS/W^John Doe/M/N/N/B^aseguradora/2/D^prestC1/D^PrestC2/E^extraC

issued to: Key: Z/R/SSSSS/W^John Doe/M/N/N/B^banco/1/D^credentialB1/D^credentialB2/E^extraB

proof: Key: Z/R/SSSSS/W^John Doe + /M/N/N/B^banco/1/D^credentialB1/D^credentialB2/E^extraB / + /M/N/N/B^as

Normalizing derivations



How many derivations?

Security is based on randomness, given by the **words mnemonic**, not the derivations and the **secrecy** of the public keys

Anonymity is given by **variety** of the addresses and **un-linkabiltiy** between one Key (derived) and its ancestors

Derivations can help **organize** the **purpose** of each different Key

Derivation patterns

As there are some parts in the derivations that are variable in length it is necessary to exchange the derivation patterns when a derivation is sent.

Example:

A user is about to receive a credential. The Issuer of the credential requests the user a DID that will be recipient of the credential, and the Issuer stablishes the pattern "/E/E/E" and sends "167873/57659/43172".

The user himself will use a "/D/D" derivation for his part of the credential derivation, setting "98765/364".

Therefore the Issuer has to use the following derivation to calculate the DID of the user from the Extended Public Key that the User sent the Entity during the on-boarding:

"/131071/0407/100111001/375351/1/98765/364/167873/57659/43172"

That matches the following pattern:

"/M/T/N/B/C/D/D/E/E/E" where "M/T/N" are fixed for this network, "B" is selected by the user for his communications with this Issuer, "C" is fixed to "1" regarding credential issuance

Santander

Suggested subject derivation paths schema

Block›chain

Alias	Pattern	Derivation	Derivation meaning	Notes
	Z	m/1037171 m/44' ...	General Purpose: (ie Identity, Crypto)	Add more in case of other use cases https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki#purpose
	R	/([0-9])*	Holder's Wallet Identity RECOVERY	0 - 2^31, randomness required
	SSSSS	/([0-9])*	SECURITY by isolation	0 - 2^31, randomness required per S level, 5 derivation levels at least
Main Identity	W	/([0-9])*	Holder's WALLET Identity derivation (isolation purposes)	0 - 2^31, randomness required
	M	/131071 ...	METHOD/main entity (ie Alastria) derivation	Free slot at https://github.com/satoshilabs/slips/blob/master/slip-0044.md, suggested registration (131071 is a Mersenne prime)
	T	/0407 /7487 ...	Network TECHNICAL (ie "quor", "fabr")	As in https://github.com/alastria/alastria-identity/wiki/Alastria-DID-Method-Specification#21-alastria-did-scheme
Network DID	N	/100111001 /112212211 /1311131 /1411141 ...	NETWORK name ("specific-idstring" Alastria, redT, redB, redQ, redH)	From palindromic primes list http://oeis.org/A002385/b002385.txt as a nerd joke/ validation of the derivation path
Interacting DID	B	/([0-9])*	Relationship with Other Party derivation (ie Issuer)	0 - 2^31, randomness required, DO NOT USE A PUBLIC LIST
	C	/0 /1 /2 /3 /1000 ...	Identity purpose : (ie Authentication, Credential issuance, Presentation signature, Presentation request Delegation ...)	
	D*	(/([0-9])*)*	Holder generated derivations: (In general at least TWO levels of purpose derivations, different values PER credential to avoid pre-image attacks against privacy)	0 - 2^31, randomness required, DO NOT USE A PUBLIC LIST
	E*	(/([0-9])*)*	Other Party generated derivations	0 - 2^31, randomness required, DO NOT USE A PUBLIC LIST

Suggested entity derivation paths schema

Block›chain

Alias	Pattern	Derivation	Derivation meaning	Notes
	Z	m/1037171 m/44' ...	General Purpose: (ie Identity, Crypto)	Add more in case of other use cases https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki#purpose
	R	/([0-9])*	Holder's Wallet Identity RECOVERY	0 - 2^31, randomness required
	SSSSS	/([0-9])*	SECURITY by isolation	0 - 2^31, randomness required, 5 derivations at least
Main Identity	W	/([0-9])*	Holder's WALLET Identity derivation (isolation purposes)	0 - 2^31, randomness required
	M	/131071 ...	METHOD/main entity (ie Alastria) derivation	Free slot at https://github.com/satoshilabs/slips/blob/master/slip-0044.md, suggested registration
	T	/0407 /7487 ...	Network TECHNICAL (ie "quor", "fabr")	As in https://github.com/alastria/alastria-identity/wiki/Alastria-DID-Method-Specification#21-alastria-did-scheme
Network DID	N	/100111001 /112212211 /1311131 /1411141 ...	NETWORK name ("specific-idstring" Alastria, redT, redB, redQ, redH)	From palindromic primes list http://oeis.org/A002385/b002385.txt as a nerd joke/ validation of the derivation path
Interacting DID	DELETED: Actually B derivations are not needed for Entitites, they use a single identity for all its interactions UNTIL they rotate their identity			
	C	/0 /1 /2 /3 /1000 ...	Identity purpose : (ie Authentication, Credential issuance, Presentation signature, Presentation request Delegation ...)	
	D*	(/([0-9])*)*	Holder generated derivations: (In general at least TWO levels of purpose derivations, different values PER credential to avoid pre-image attacks against privacy)	0 - 2^31, randomness required, DO NOT USE A PUBLIC LIST
	E*	(/([0-9])*)*	Other Party generated derivations	0 - 2^31, randomness required, DO NOT USE A PUBLIC LIST

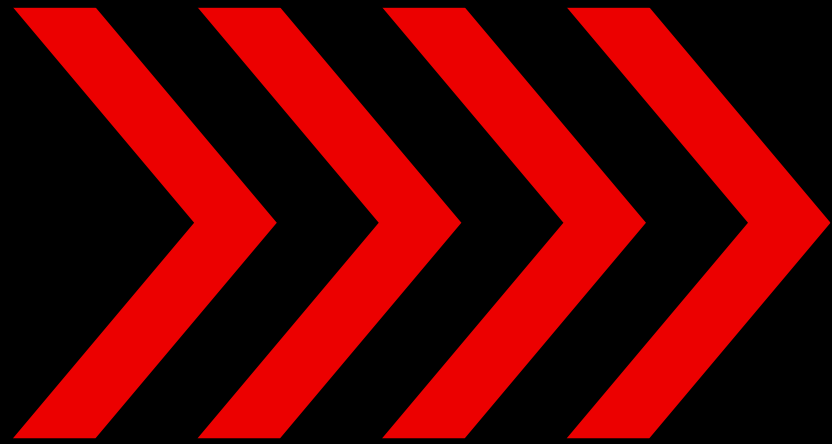
https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final

Table 4: Security strength time frames

Security Strength		Through 2030	2031 and Beyond
< 112	Applying protection	Disallowed	
	Processing	Legacy use	
112	Applying protection	Acceptable	Disallowed
	Processing		Legacy use
128	Applying protection and processing information that is already protected	Acceptable	Acceptable
192		Acceptable	Acceptable
256		Acceptable	Acceptable

Derivations where anonymity is part of the purpose whould have >128 bits, ergo at least 5 levels at derivation S

New procedures



Block

Modified 2022/10/07

Sample derivation paths

Z	R	SSSS	W	M	T	N	B	C	DD	EEE
m/1037171	/11235813	/13547/2414753/5463/4860124	/81552345	/131071	/0407	/100111001	/375351	/1	/98765/364	/167873/57659/43172

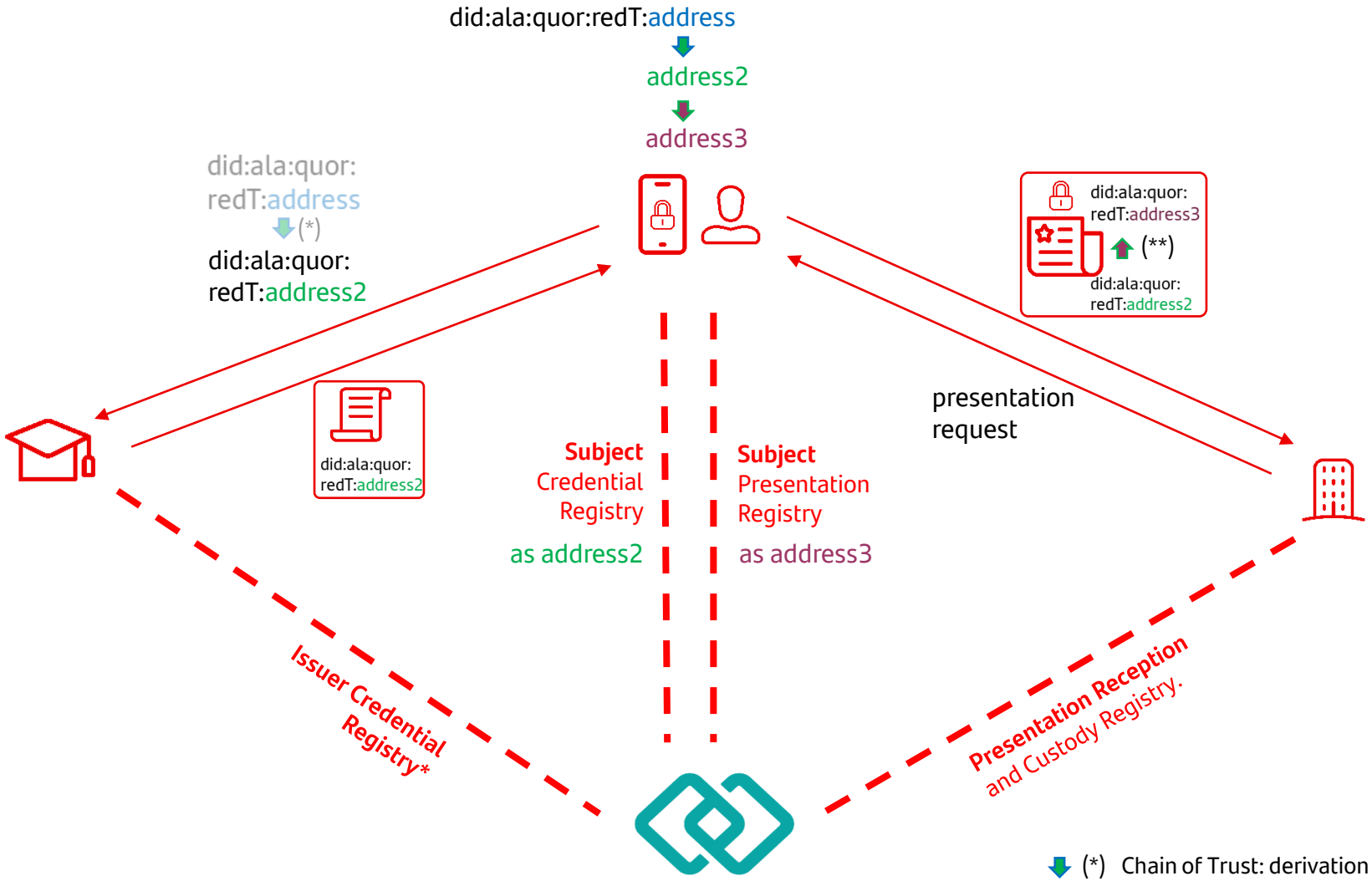
"Main Identity" calculated from "12-words seed" + derivation
m/1037171/11235813/13547/2414753/5463/4860124/81552345

At a "Given network DID" is calculated from "Main Identity" + derivation **/131071/0407/100111001**

"Interacting DID" with an Issuer Entity the user shares the DID calculated from "Given network DID" + derivation "/375351"

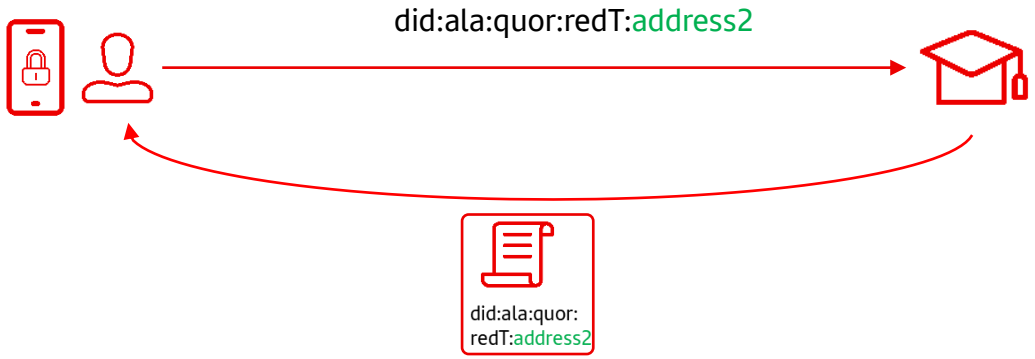
When requesting a credential the Issuer Entity requires the derivation: /167873/57659/43172 using the pattern "/E/E/E" for that same credential the user sets the derivation /98765/364 using the pattern /D/D. So the Entity takes the given "Interacting DID" and calculates "/1/98765/364/167873/57659/43172" where "/1" is fixed for credential issuance purpose (derivation level C)

Credential & presentations (EPIC)

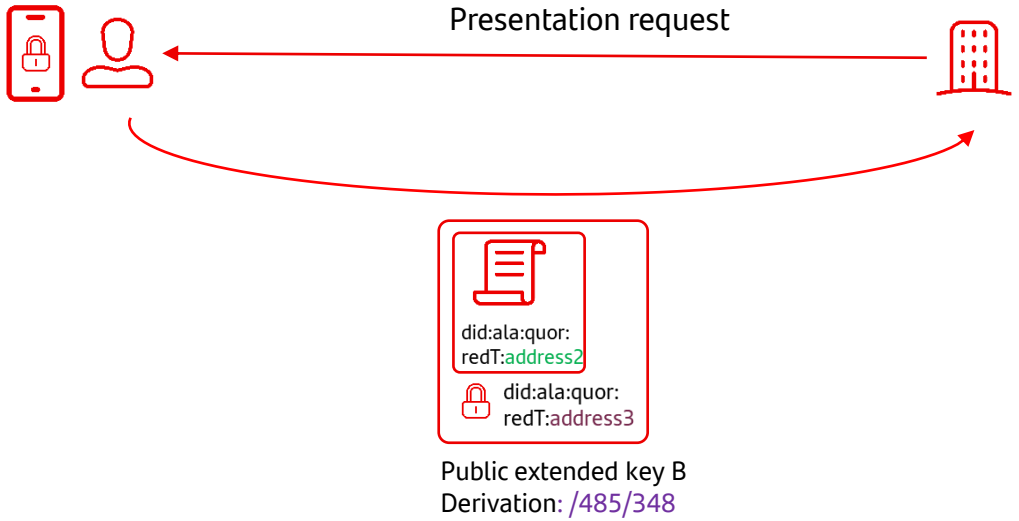


(*) Chain of Trust: derivation from address to address2, optional
(**) Chain of Trust: derivation from address2 to address3

Chain of Trust



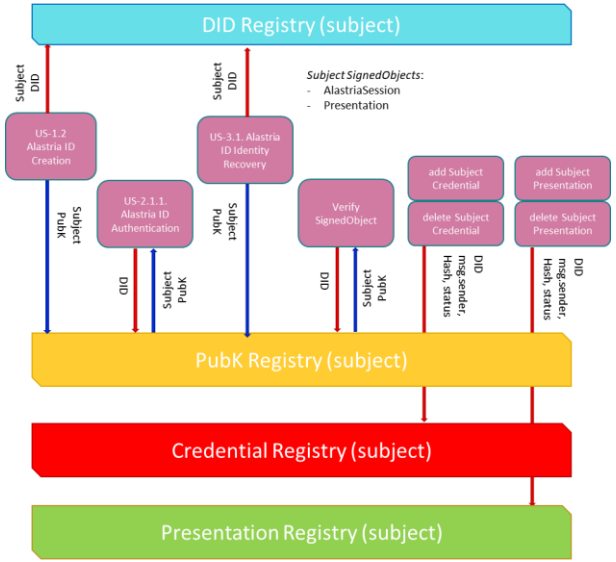
For the best privacy option instead of using “address” to receive a Credential it would be better to use “address2” that comes from the sample derivation “m/190/7/834/49/340853/36”. The subject can create a different “address2” per issuer or even per credential, just storing the derivation along with the credential links “address2” to the credential.



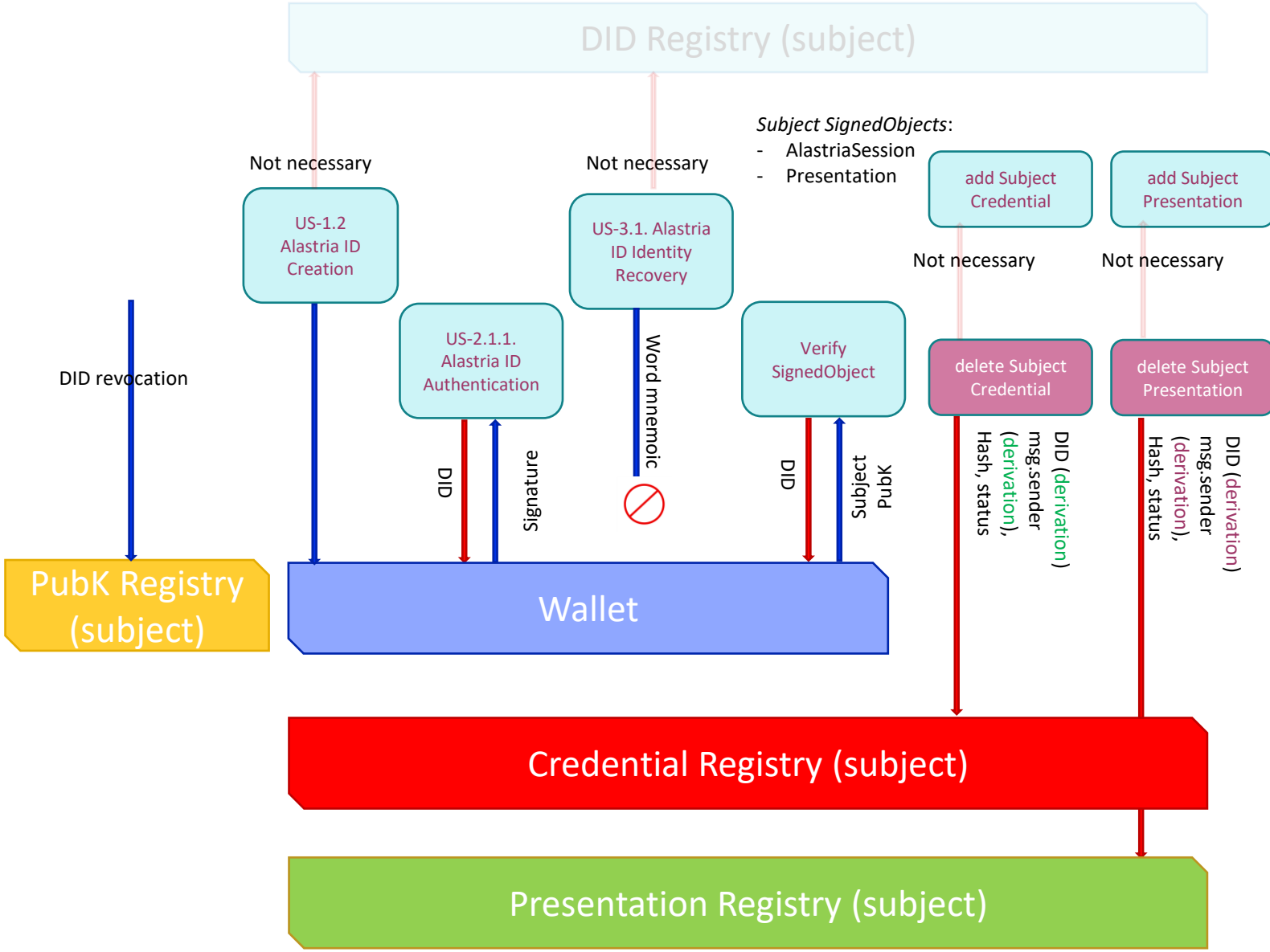
1. **R**elying **P**arty receives a Presentation, signed by “address 3”
2. RP can check the signature validity
3. Inside the Presentation there’s a credential emmitted to “address 2”
    1. RP can confirm that the public key B is associated to “address 2”
    2. RP can calculate public key C from public key B + derivation /485/348
    3. RP can confirm that the “address 3” is associated to public Key C
4. Therefore the subject is in control of Public Key B and Public key C, and Public Key C is derived from B, so the presentation is signed by the same subject of the credential.

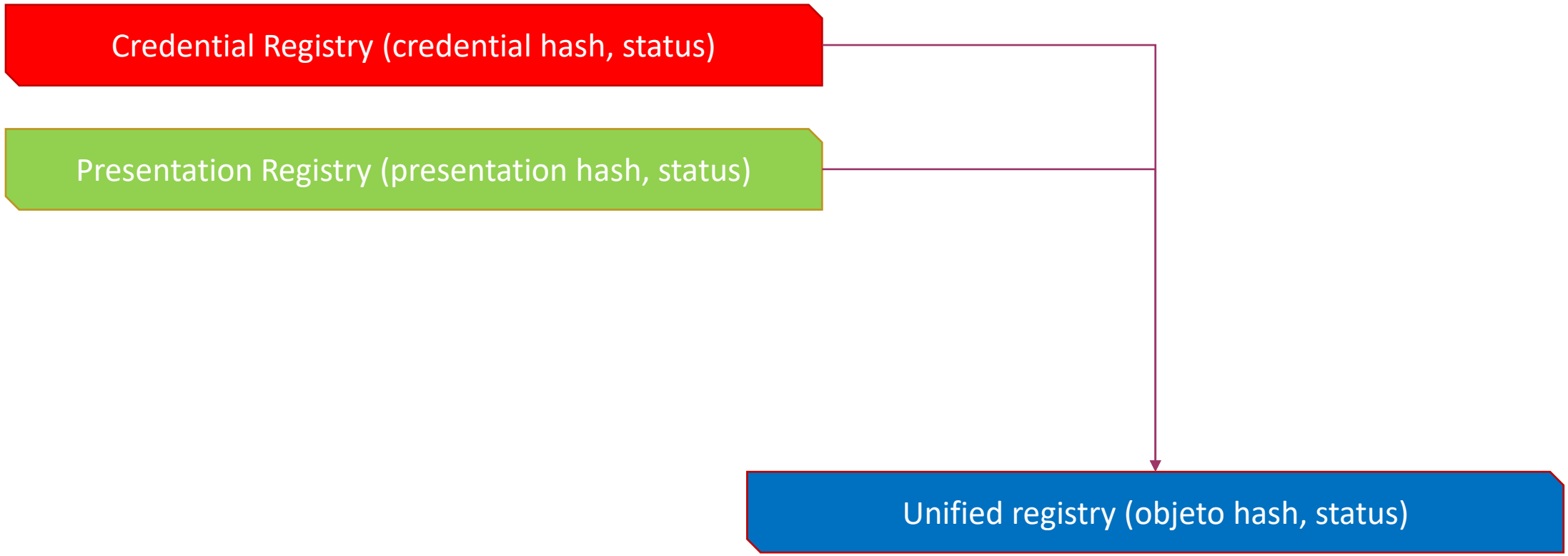
Updated model

NO GDPR problems at Alastria ID?



Block›chain




Credential Registry (credential hash, status)
Presentation Registry (presentation hash, status)
Unified registry (objeto hash, status)

Adoption plan

Alastria Architecture

Demo Wallet

Uses library integrated

Demo Entity

Uses Swagger defined services

Service API

Library

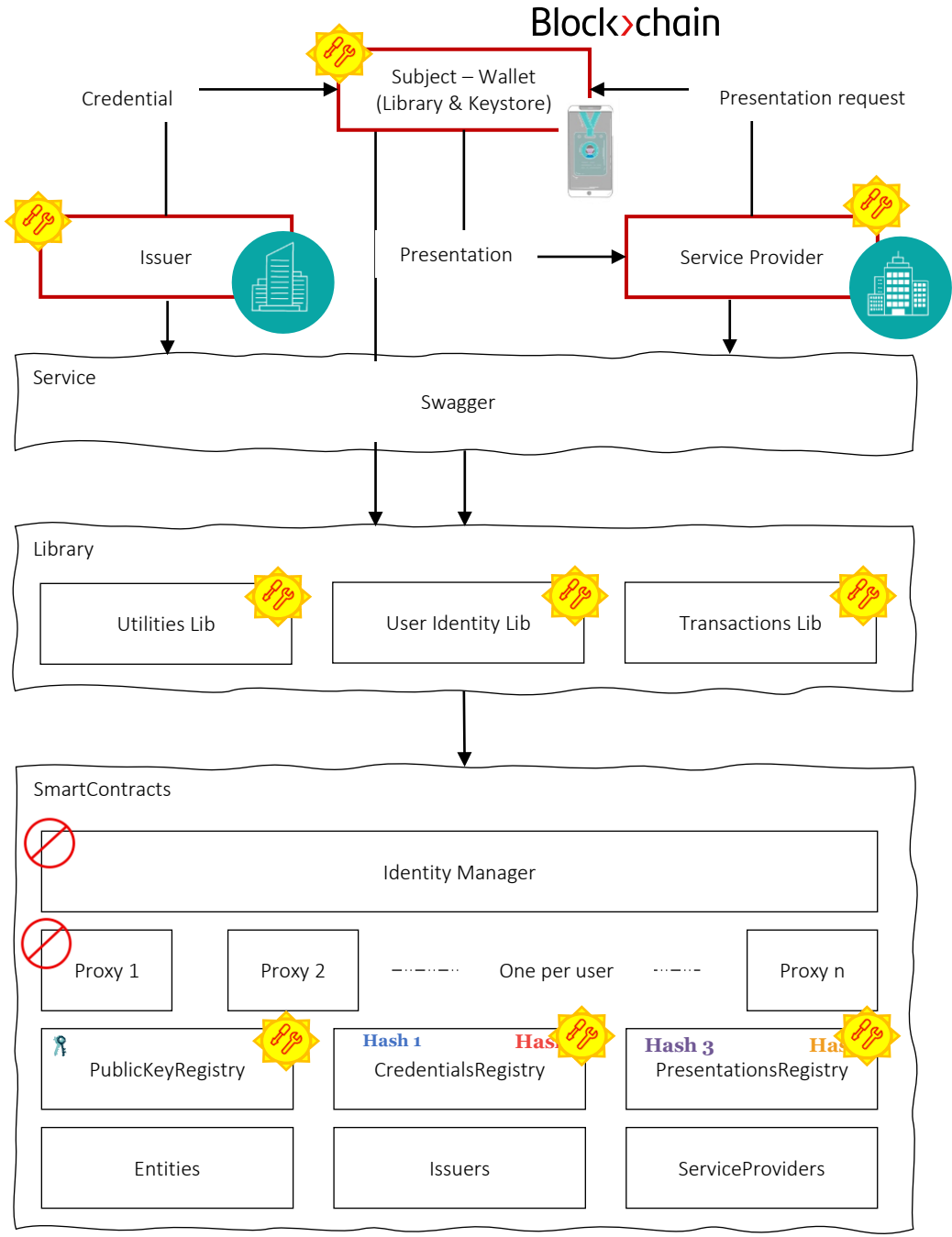
Strongly recommended to ensure interoperability

Smart Contracts

Mandatory to ensure intra-operability

DID method specification

The very definition of **Alastria ID**



Adoption plan

		Notes
Crypto model	HD Wallets	Cryptography review
	Derivations	Derivation definition
	Objects	Schemas and verifications
	VC verification	
	Anti-cracking protection	
DID method	US stories adaptations	New Recovery
Wallet		Whole new implementation of Key generation, storage, etc
Service Provider		Whole new implementation of Key management, storage, etc
Libraries	Libraries	New libraries
	Examples	New examples
SmartContracts		Review
Documentation		

Comparision

Derivation based DIDs

Credentials

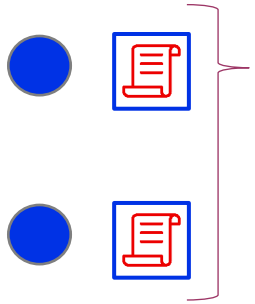
Presentation

Credential 1:
Key: Z/R/SSSSS/W^John Doe/M/N/N/B^banco/1/D^credentialA1/D^credentialA2/E^extraA

Credential 2:
Key: Z/R/SSSSS/W^John Doe/M/N/N/B^banco/1/D^credentialB1/D^credentialB2/E^extraB

Presentation:
Key: Z/R/SSSSS/W^John Doe/M/N/N/B^aseguradora/2/D^prestC1/D^PrestC2/E^extraC

Wallet:
Key: Z/R/SSSSS/W^John Doe



Signature of Presentation using **Presentation Key**
(at derivation: Z/R/SSSSS/W^John Doe/M/N/N/B^aseguradora/2/D^prestC1/D^PrestC2/E^extraC)

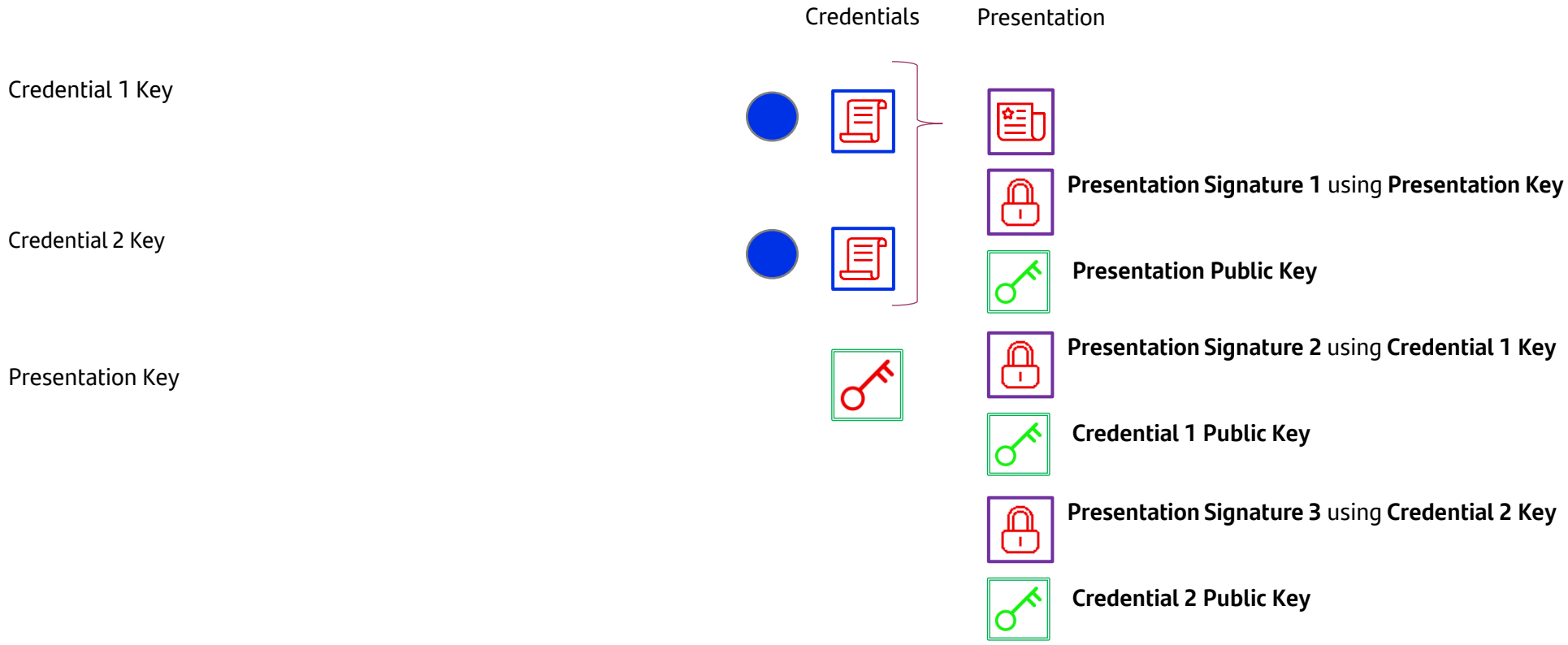
Wallet public key (at derivation: Z/R/SSSSS/W^John Doe)

- Z/R/SSSSS/W^John Doe/M/N/N/B^banco/1/D^credentialA1/D^credentialA2/E^extraA : **Credential 1 derivation**
- Z/R/SSSSS/W^John Doe/M/N/N/B^banco/1/D^credentialB1/D^credentialB2/E^extraB : **Credential 2 derivation**
- Z/R/SSSSS/W^John Doe/M/N/N/B^aseguradora/2/D^prestC1/D^PrestC2/E^extraC : **Presentation derivation**

Procedure:
1. Calculate **Presentation Key** using **Wallet public key** + Presentation derivation
2. Verify that **Wallet public key** + **Presentation derivation** produces address used in Presentation DID.
3. Verify **Signature of presentation** with calculated **Presentation Derivation**
4. Verify that Wallet public key + **Credential 1 derivation** produces address used in Credential 1 DID.
5. Verify that Wallet public key + **Credential 2 derivation** produces address used in Credential 2 DID.

EVERYTHING comes Wallet Public Key using different derivations!

Several unrelated DIDs



Procedure:

1. Verify **Presentation Signature 1** using **Presentation Public Key**
2. Verify **Presentation Signature 2** using **Credential 1 Public Key**
3. Verify **Presentation Signature 3** using **Credential 2 Public Key**

These steps only proves user is in control of Presentation, Credential 1 and Credential 2 keys at one moment NOT that they are related to him or in fully in control always (keys can be shared)