# OpenDSU

## Presentation for Trust over IP  Community

**Marco Cuomo**
**Sînică Alboaie**

29 March 2021

## Overview

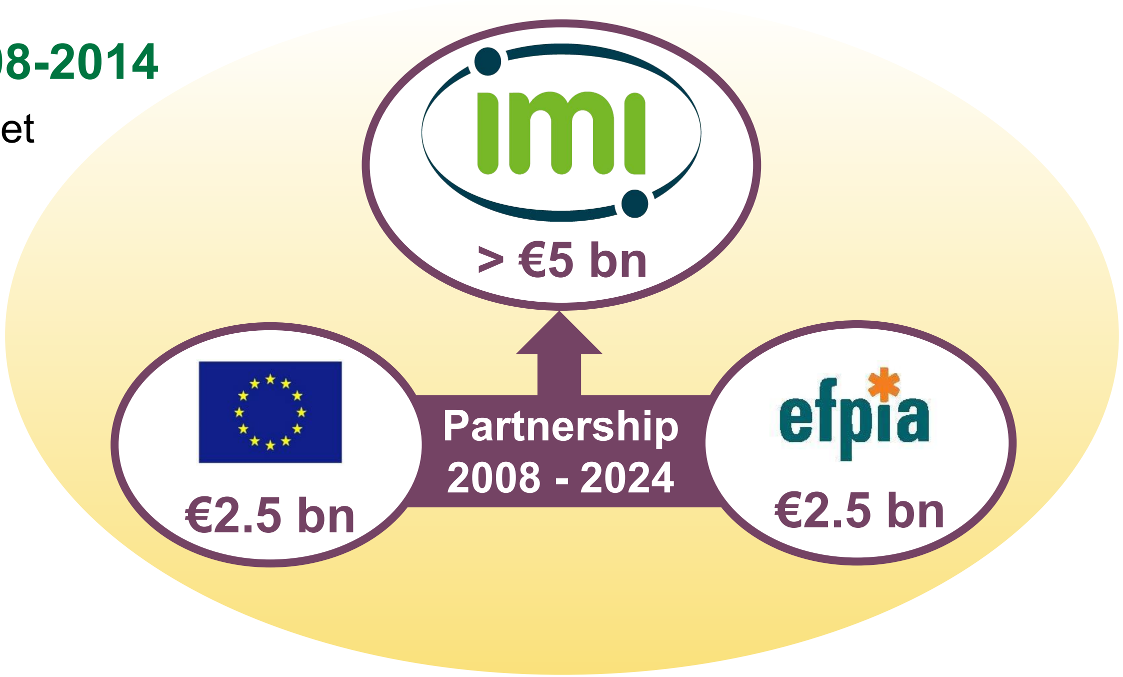
- PharmaLedger Platform Architecture will be based on OpenDSU
- Result of a EU Research Project: PrivateSky ( 2015-2021)
- Towards creating an IDSA  community
- OpenDSU core ideas
  - KeySSIs as generalisation of DIDs
  - Data Sharing Units: signed data and code anchored in blockchains/ledgers
    - approach to solve data sharing and interoperability
    - data validations, provenance, integrity by blockchain anchoring
  - Self Sovereign Applications and the OpenDSU vision of Universal Wallets
- Complementarity but also in disagreement with some DID & VC core beliefs
  - Real use cases require heavy customisations  ( data-models + DSL programmability is weak)
  - Zero Access Infrastructure and avoidance of the Cloud Agents
  - Standardised **verification** looks like a badly leaking abstraction
- OpenDSU specific approach related to validation
  - Custom code based validations

# Innovative Medicines Initiatives (IMI)

*Europe's partnership for health*

**IMI1: 2008-2014**

€2 bn budget
59 projects



**IMI2: 2014-2024**

€3.3 bn budget
More ambitious
More open
Greater scope
60 projects so far

# PharmaLedger in a Nutshell

**Who?** PharmaLedger partners comprises of pharmaceutical companies, hospitals, universities, patient organizations, tech companies... building an ecosystem!

**Why?** To **empower patients**, increase **trust** among healthcare stakeholders, support medicine drug traceability and data privacy, and build a **new culture of collaboration in healthcare**.

**What?** A scalable blockchain **platform validated through reference use cases** in supply chain, clinical trials and health data that will serve trendsetters for the industry, enabling early adopters.

**How?** Pharmaledger will **design**, **validate** and provide **agile delivery** of innovative blockchain-enabled healthcare **applications** across the industry, from manufacturers to patients; while creating an **innovative governance** approach for **sustainability.**

PharmaLedger
BLOCKCHAIN ENABLED HEALTHCARE

- ❖ **Duration**: 3 year, Jan '20 – Dec '22
- ❖ **Consortium:** 29 partners, largest of its kind
- ❖ **Budget**: EUR 22 million
- ❖ **Focus Areas**: Supply Chain, Clinical Trial, and Health Data

# PharmaLedger

PharmaLedger
BLOCKCHAIN ENABLED HEALTHCARE

NOVARTIS
POLITÉCNICA

PDM
RomSoft
Arteevo
TECHNOVATIVE SOLUTIONS
Ekon
CERTH
CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

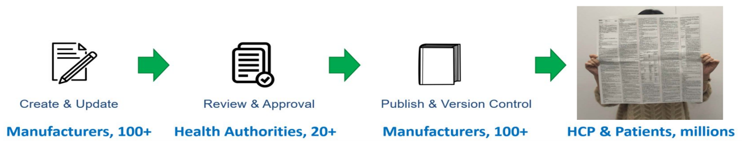
abbvie
AstraZeneca
BAYER
gsk
janssen
MSD
novo nordisk®

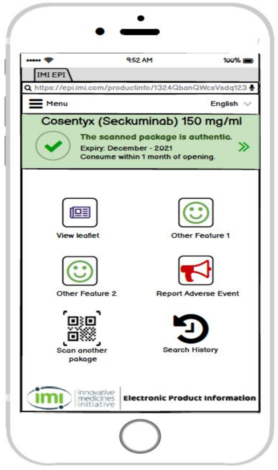
Pfizer
Roche
ucb Inspired by patients. Driven by science.

KU LEUVEN CiTiP
CENTRE FOR IT & IP LAW
Onorach™
ICSI
Uniklinikum Würzburg UK
DEMOCRITUS UNIVERSITY OF THRACE
Hospital Espírito Santo E.P.E.
Bambino Gesù
OSPEDALE PEDIATRICO

INCM
IMPRENSA NACIONAL CASA DA MOEDA
EPF European Patients Forum
EF OCP

# ePI: a concrete use case for OpenDSU





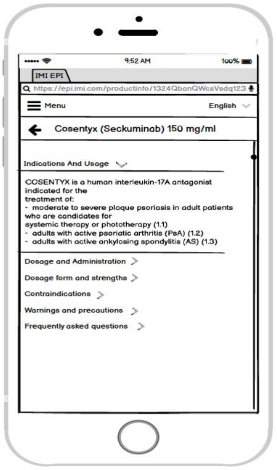
IMI EPI

https://epi.imi.com/productinfo/1324QbanQWcsVsdq123

Menu English

Cosentyx (Seckuminab) 150 mg/ml

The scanned package is authentic.
Expiry: December - 2021
Consume within 1 month of opening.



IMI EPI

https://epi.imi.com/productinfo/1324QbanQWcsVsdq123

Menu English

Cosentyx (Seckuminab) 150 mg/ml

Indications And Usage

COSENTYX is a human interleukin-17A antagonist
indicated for the
treatment of:
- moderate to severe plaque psoriasis in adult patients
who are candidates for
systemic therapy or phototherapy (1.1)
- adults with active psoriatic arthritis (PsA) (1.2)
- adults with active ankylosing spondylitis (AS) (1.3)

Dosage and Administration

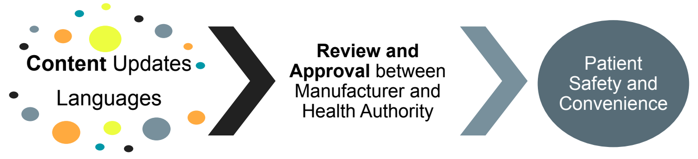
Dosage form and strengths

Contraindications

Warnings and precautions

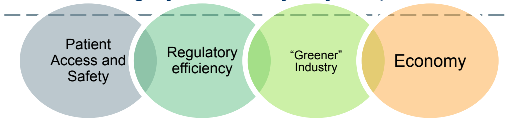
Frequently asked questions

This use case starts with the creation of the **ePI in digital form** by the manufacturer, the **review and approval** of the ePI with the Health Authorities, **updates** to the ePI and **dissemination** of the ePI to the Patient/ Health Care Practitioner/ Provider (HCP).


Content Updates
Languages
Review and Approval between Manufacturer and Health Authority
Patient Safety and Convenience

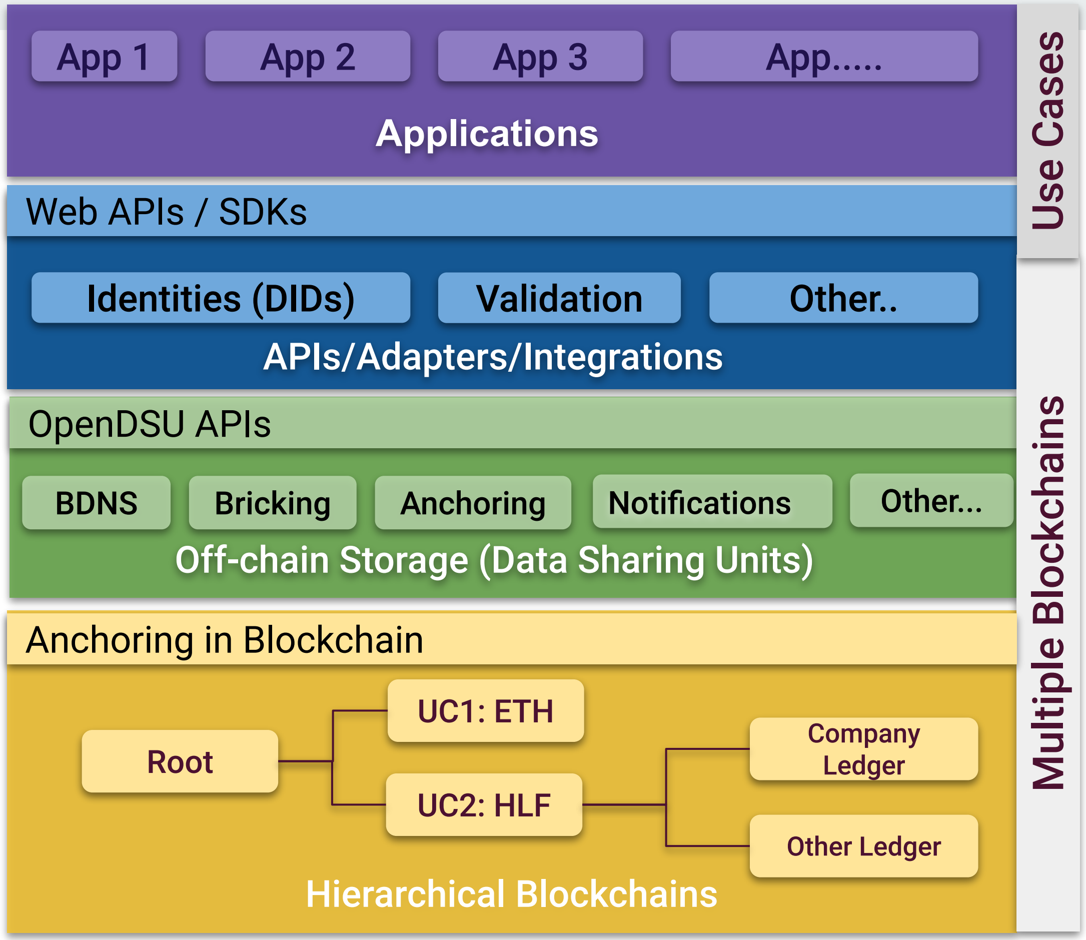
## Blockchain and PharmaLedger Value Proposition

| | |
|---|---|
| Trust | **Transparent and immutable** review and approval transaction records. **Smart contracts** set the transaction rules so only approved eLeaflets are published |
| Interoperability | **Facilitates transactions** between manufacturer systems and multiple health authorities with easy access for Patient with 'One App' |
| Security | **Decentralised system** for storing ePI provides secure platform, instead of central database, providing resilience against cyber attacks |
| Privacy | **Data Self-Sovereignty and Anonymity** are paramount and not negotiable |

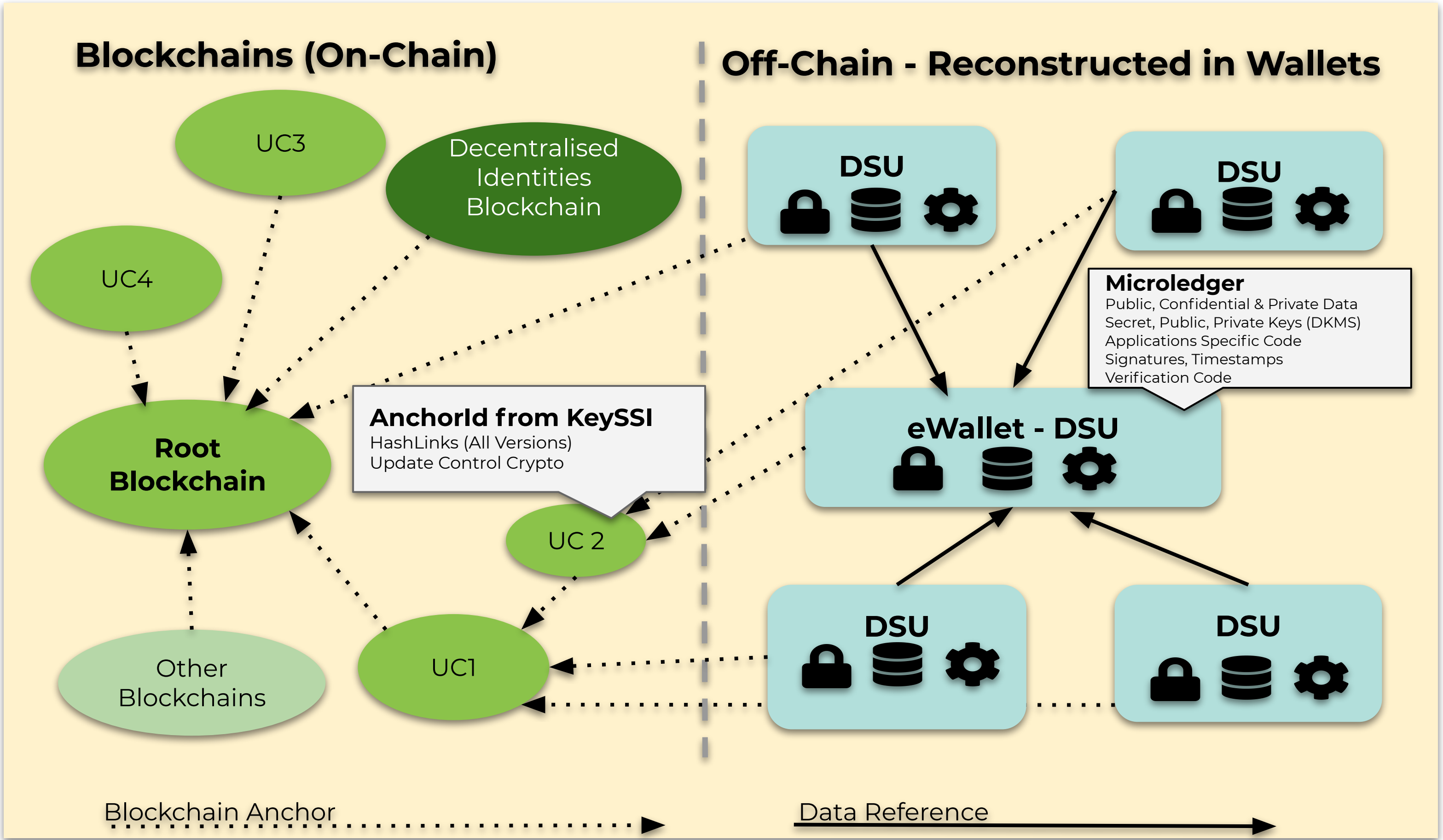


# Layered Architecture

- Blockchain Agnostic
  - Can use any blockchain
  - Integrity & Traceability
- Hierarchical Blockchains
  - Increase security for ledgers with small number of replicas
  - No size fits all
- Off-chain data and computation integrity
  - Confidential Smart Contracts
  - Use Case Specific Optimisations



# DSU Anchoring



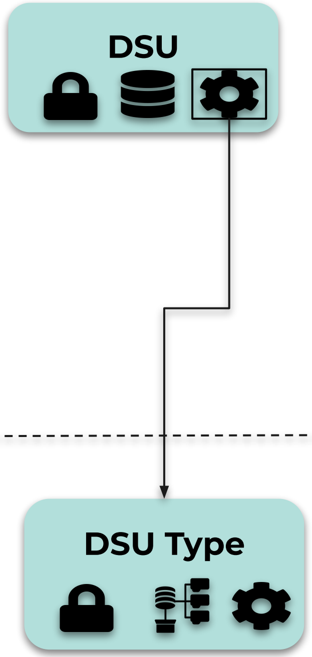
# OpenDSU: DSU-Types Model

**Data Sharing Units (DSU):**

- Encrypted and identified by KeySSIs
- Can store Keys used for encryption and digital signatures

- Data - File Systems or Embedded Databases (key value, indexed, ledgers)

- “Contains” code (Validation & Business Logic) from a DSU-Type
- Instances of DSU Types

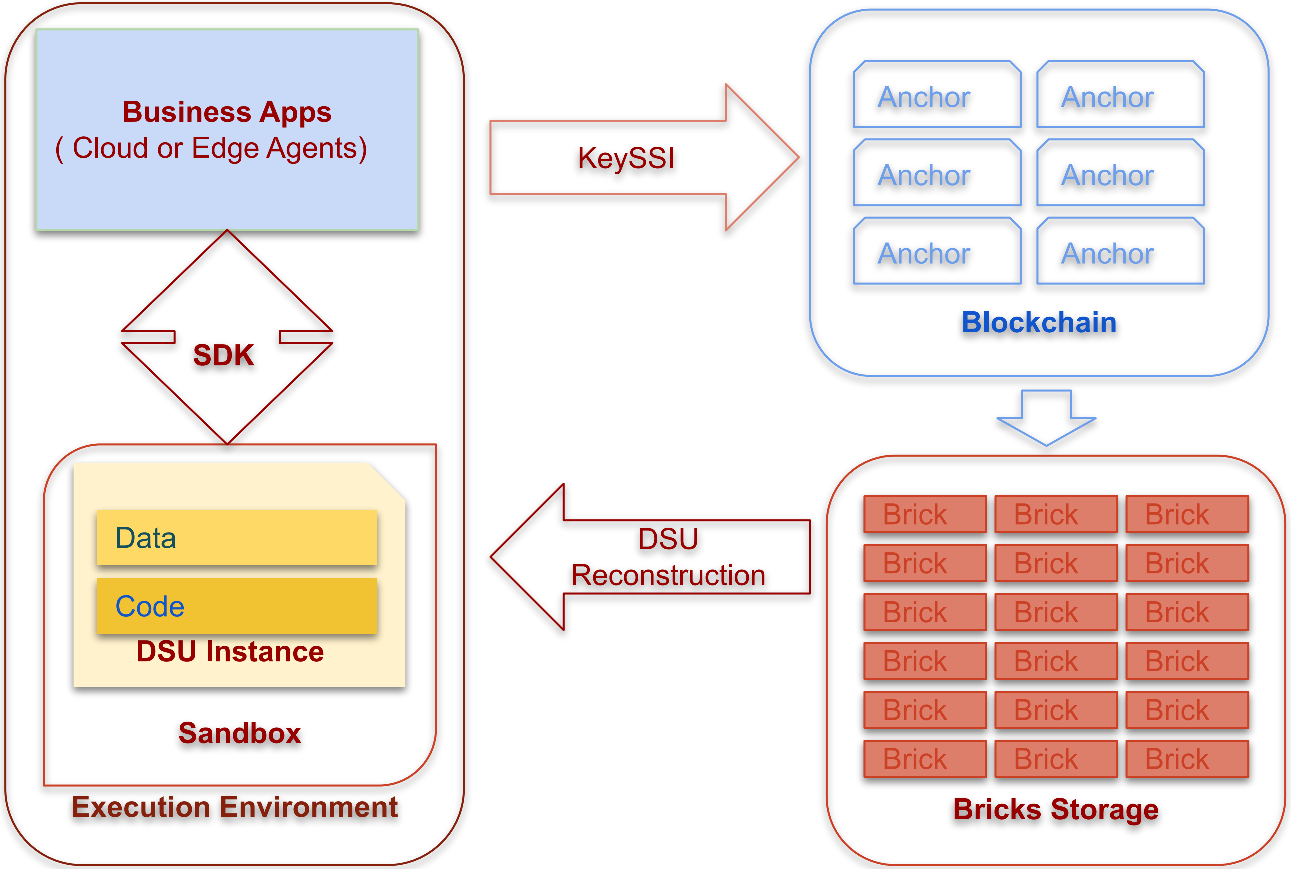


**DSU- Types (named also DSU constitutions)**

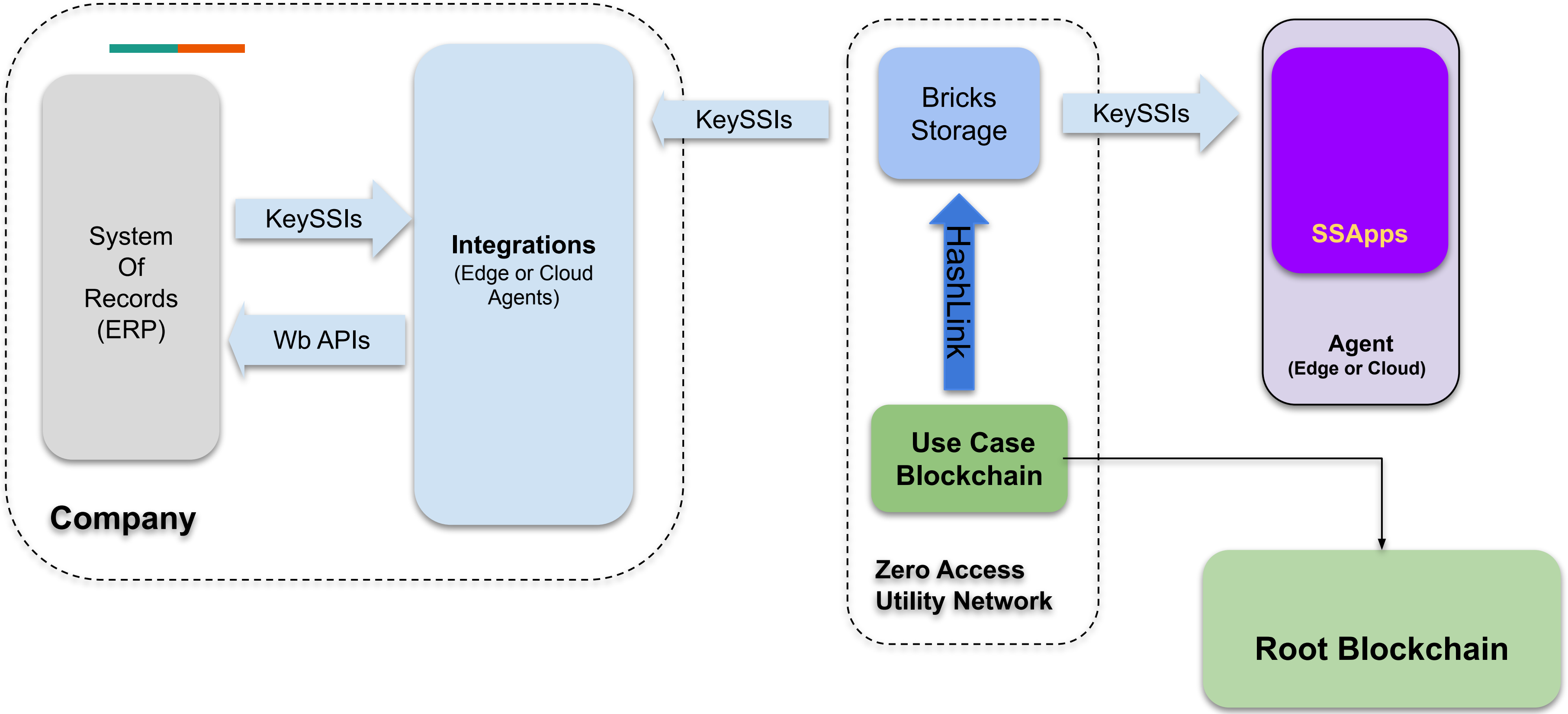
- A DSU storing SIGNED CODE
- Like a class for DSU instances
- DSU APIs (abstract interfaces that hide internal data representations)

# DSU Reconstruction from Bricks

- **Blockchain is for Anchoring**
  **Minimise Data Leakages**
  **Decentralised Access Control using KeySSIs**
  - Identifier
  - Cryptographic Key
- **Client-Side Encryption**
- **The code** of the DSU executed in **a sandbox** (Cloud or Edge Agents)
  **DSUs are lightweight JavaScript containers**



# The OpenDSU Reference Architecture Overview



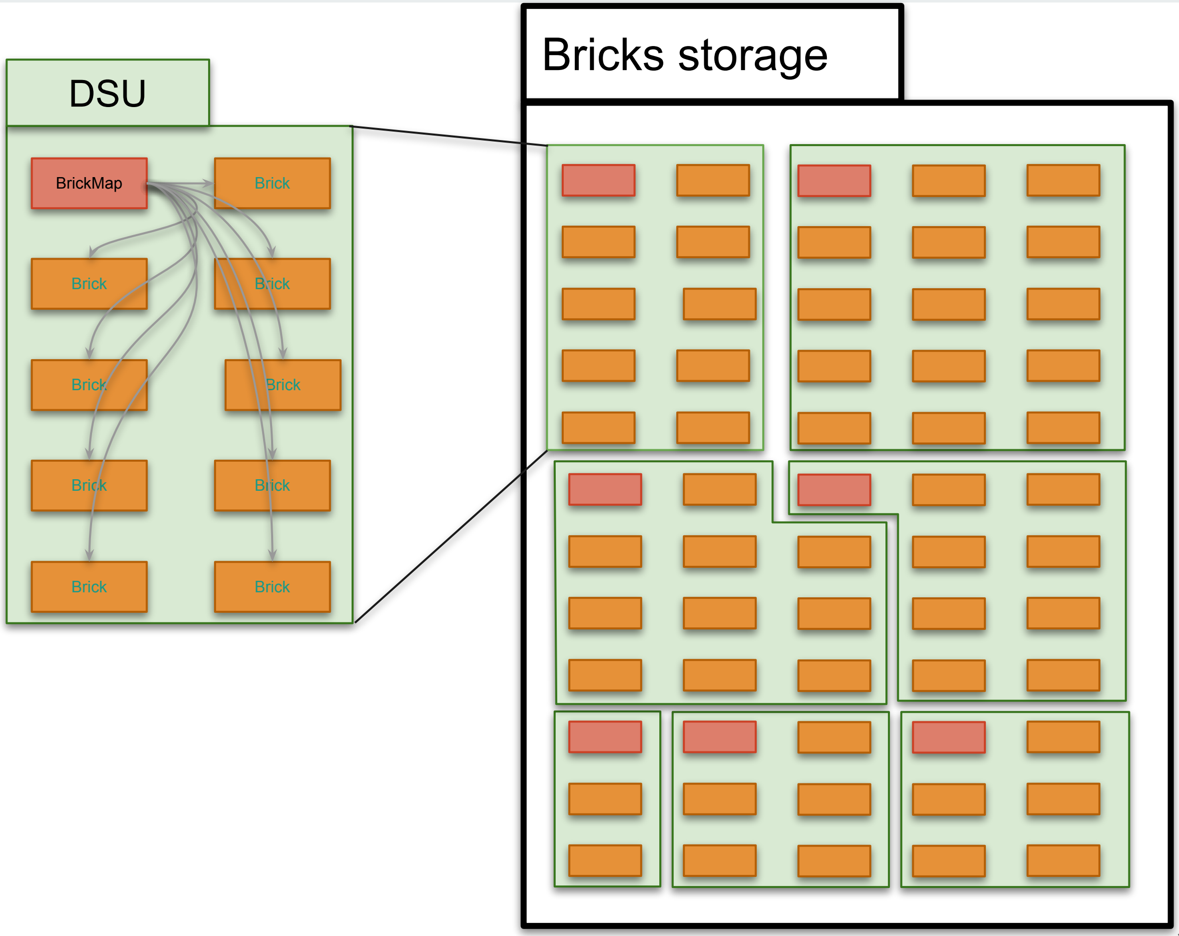
# Bricks Storage - detailed structure

It's a web service that stores and retrieves bricks

A brick once created is **read-only** and HASH identified

A DSU is saved as bunch of Bricks (starting with a BrickMap that keeps references [1] to all other Bricks from the DSU)

If you want to load a DSU the engine loads the first brick (BrickMap), then based on it it loads the rest of the DSU



1 - HashLinkSSI

## BDNS : Decentralisation by segregation

BDNS stands for **Blockchain Domain Naming System**

Resolve names into blockchain network configurations

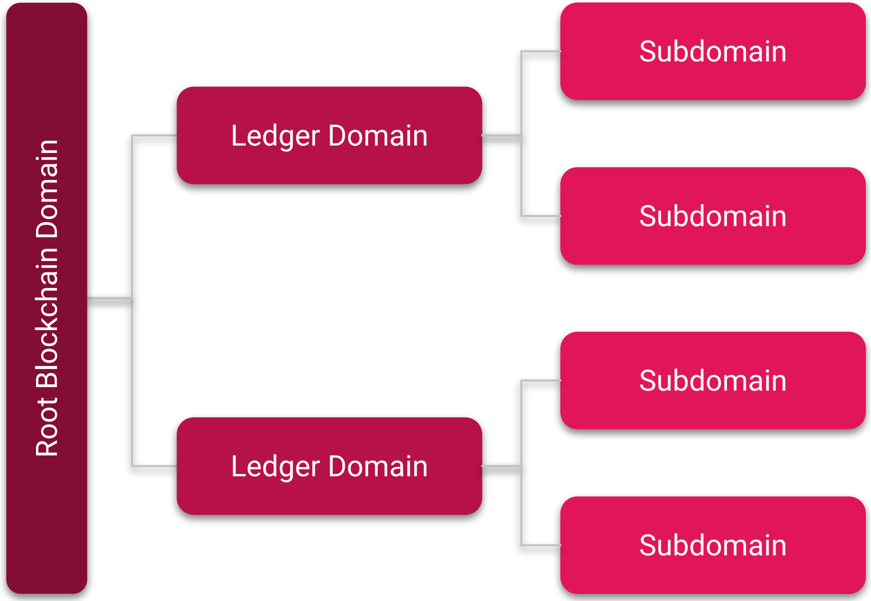
It’s the DNS of OpenDSUs but probably could replace DNS. (DNS was launched in 1985...)

BDNS aims to be a hierarchical and decentralized naming system for blockchains, distributed ledgers, and even for individual DSUs.

BDNS extends/complements the DNS while preserving the existing user experience for the average user.

**Example of a blockchain domains:**

vault.novartis.pharmaledger
epi.novartis.pharmaledger



## KeySSI - Why?

**A KeySSI is both an access token (a key) and an identifier.**

KeySSI is the acronym of “Key Self-Sovereign Identifiers”

- Resolved to DSUs: Identify and locate DSUs (URI)
- Used also for DSU Encryption (decentralised access control)
- DSU Validation and ownership proofs
- Identity and control of the DSU’s anchors ( zero access blockchain anchors)
- could be used to implement DID methods (DID documents obeying the OpenDSU specific decentralised access control rules)

## KeySSI - Syntax

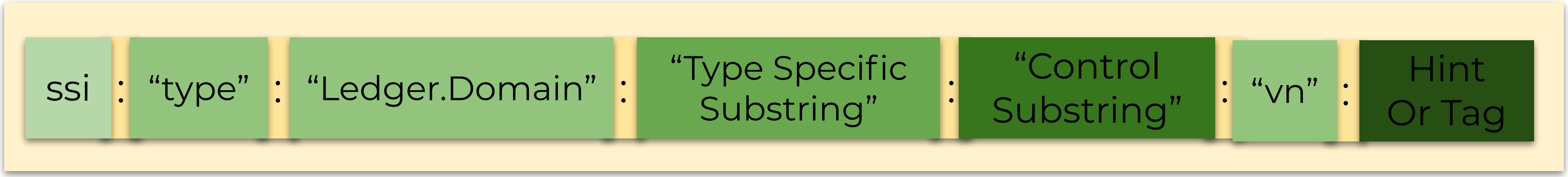


*Diagram: Syntax of KeySSI Identifier*

1. "ssi" just tells us it's a SSI key
2. "type" defines complementary types of KeySSI
3. "Ledger.Domain" represent an ledger/blockchain domain
4. The "Type Specific Substrings" should contain enough random bits for good security.
5. The "Control Substring" used by the anchoring services to validate the requests for a new version of the anchored DSU. The algorithm used for verification is type-specific.
6. The "vn" is a string reflecting the version number of the type. Not be confused with DSU versions.
7. The "hint" part is optional and subtype-specific.

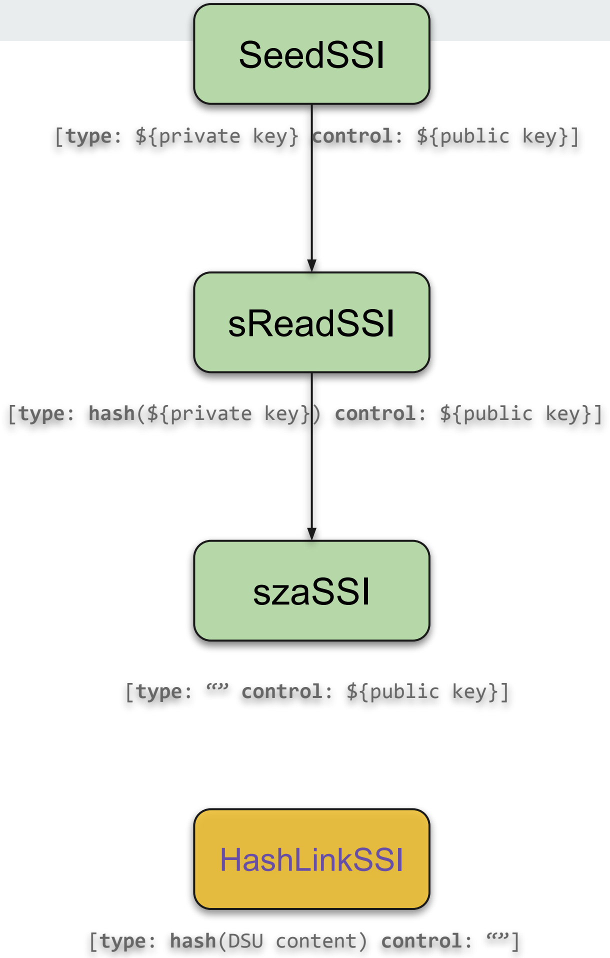
**Examples:**
ssi:seed:default:7PxHBdtYBVxzAQBsDAt9LneGQu5UKSndkg9ngf5d113E::v0
ssi:sza:default:e064764d4701ff563aa9be06c3b301ebdaa262a40c6ecc592254df75dbf097f6::v0

# KeySSI - families example

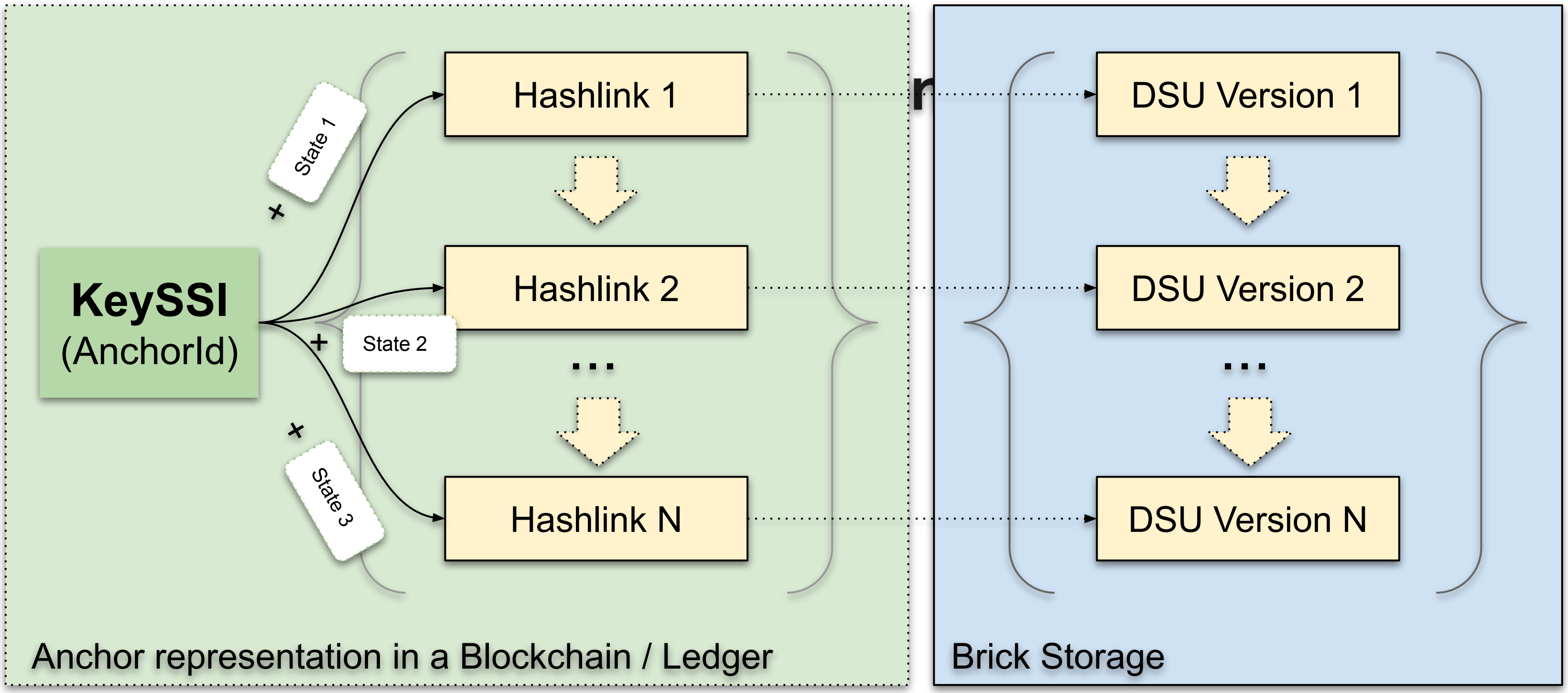
**SeedSSI** - provide write access to the anchored DSU (typically are not shared)

- **sReadSSI** - provides read access to the anchored DSU. The DSU is encrypted with a symmetric key derived from the sReadSSI
- **szaSSI** - provides Zero Access. Having a szaSSI indicates that a SeedSSI exists and the attached DSU has a specified number of versions

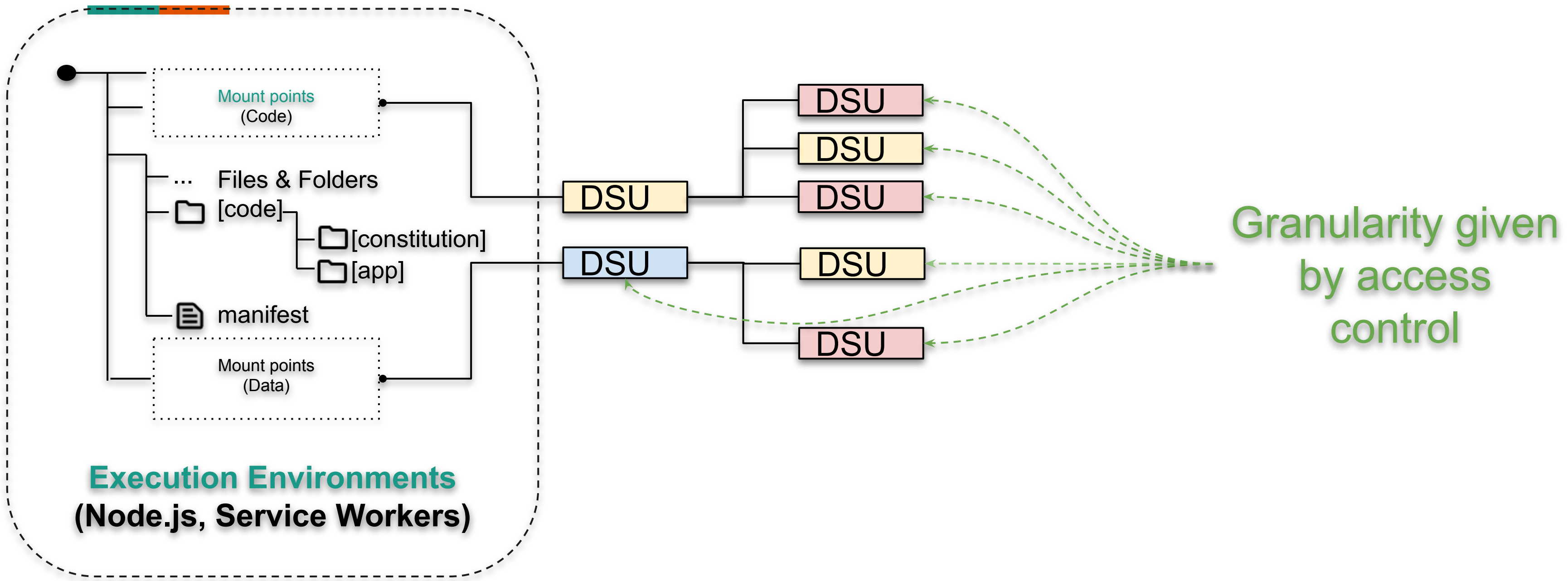
**HashLinkSSI** - used to store references to DSU in brick storage


SeedSSI
[type: ${private key} control: ${public key}]
sReadSSI
[type: hash(${private key}) control: ${public key}]
szaSSI
[type: “” control: ${public key}]
HashLinkSSI
[type: hash(DSU content) control: “”]

# Anchoring using KeySSIs



# DSU structure: Similar to a filesystem (Key/Value database)

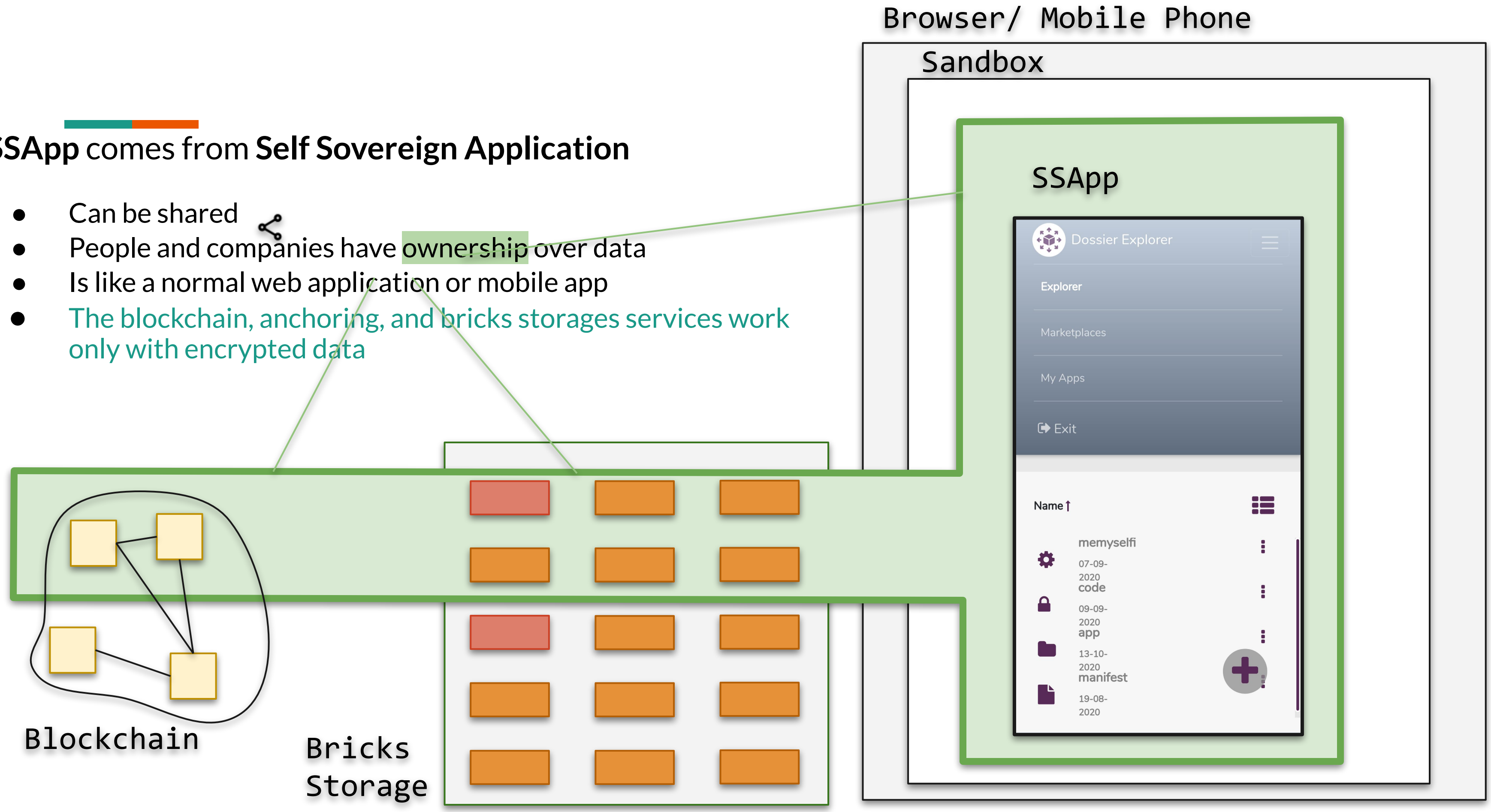


Similar with a docker container but lighter. You can mount DSUs in other DSUs
Can be used to instantiate embedded databases or micro-ledgers(transactions order from anchors)

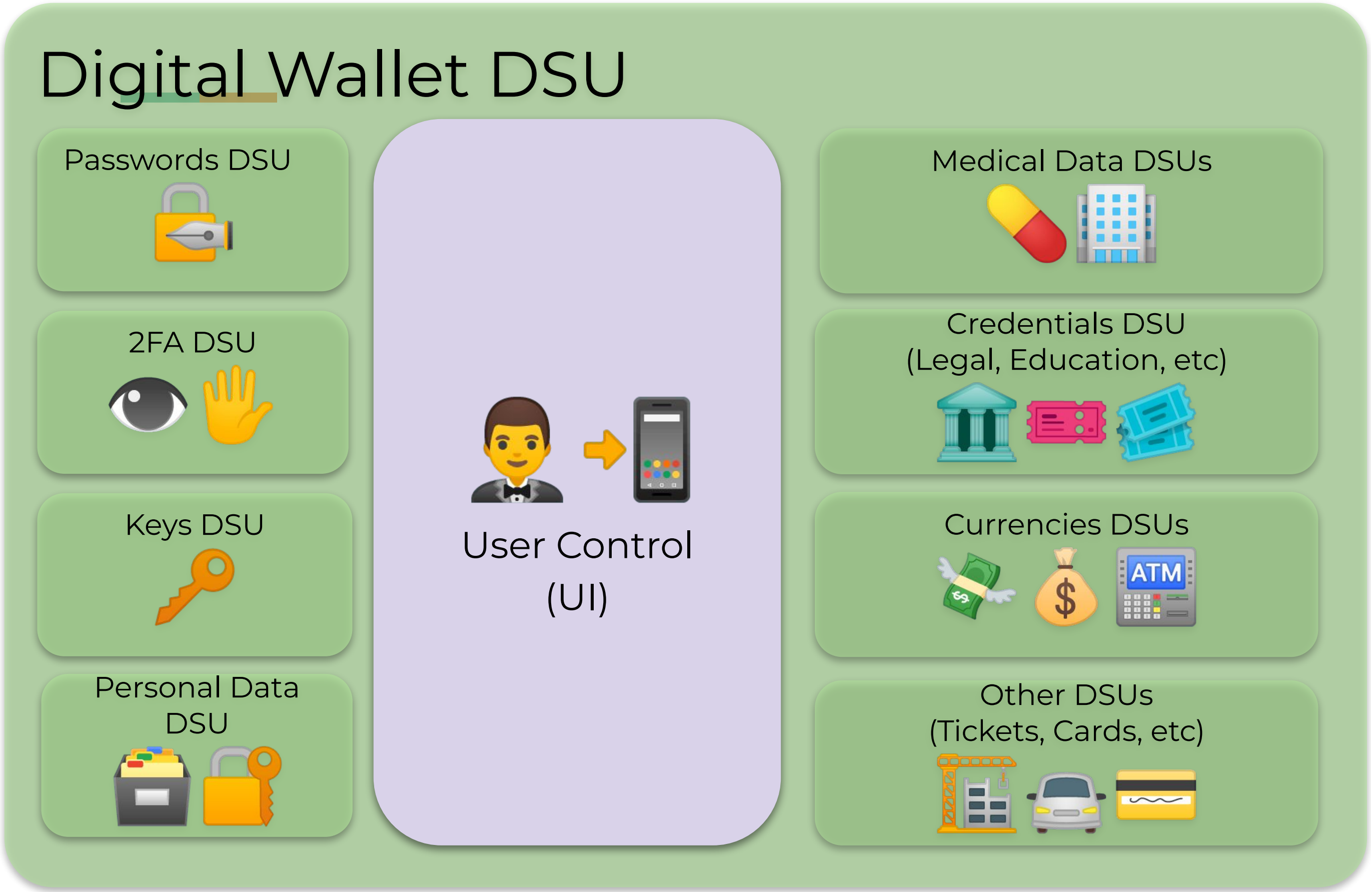
# SSApp - UI in wallets

**SSApp** comes from **Self Sovereign Application**

- Can be shared
- People and companies have ownership over data
- Is like a normal web application or mobile app
- The blockchain, anchoring, and bricks storages services work only with encrypted data



# The vision of the OpenDSU Universal Wallets



**Dynamic, interoperable and portable Digital Wallet**

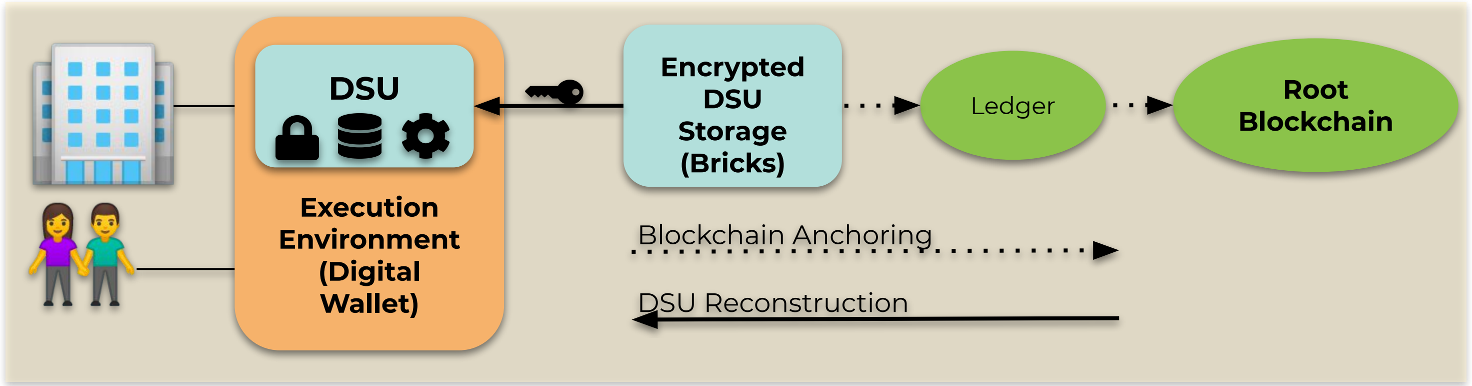
Stored in DSUs

Customizable to the needs of the users (individuals and companies) : Contain DSU embedding SSApps

Supports any device such as smartphones, wearables, browsers

**Will be Digital Wallets the next big applications platform?**

# OpenDSU “Standards”


DSU
Execution Environment (Digital Wallet)
Encrypted DSU Storage (Bricks)
Ledger
Root Blockchain
Blockchain Anchoring
DSU Reconstruction

Defining an **open standard** for handling off-chain storage to achieve **an unified approach** for interoperability: Digital Wallets, Encrypted Data Vaults, Decentralised Key Management Systems, Cloud and Edge Agents

# OpenDSU: Complementarity but also disagreements with some DID & VC core beliefs

**Complementarity and vision alignments**

- We are all aiming for "Digital sovereignty" in identities, data and even applications
- Many great ideas in the DID ecosystems
- OpenDSU can use most W3C DID technologies (but has different opinions on some core beliefs)

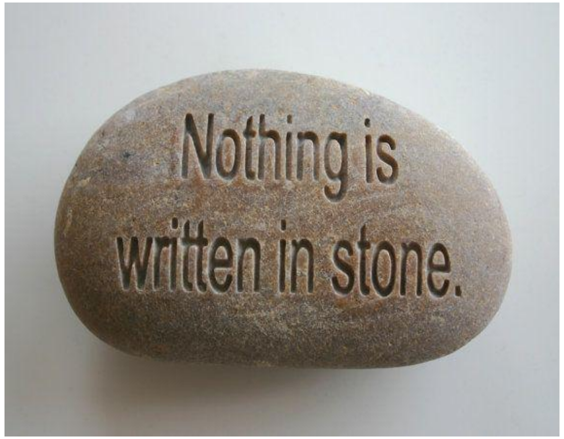
**Different core beliefs (comparing with Hyperledger Aries but even with the core standards)**

- Privacy Enhancing Technologies for enterprise world should be sound from all points of view…
- OpenDSU preference for using "general-purpose languages" and avoidance of too many "standards" on data models and myriads of DSLs (Domain Specific Languages) and protocols.
- Divergent core assumption regarding the importance of "validation" or the role of web services & cloud agents

**Why we diverge?**

- Communication problems with the business stakeholders: they kind of understand digital signatures and validated data by certificates or credentials data but have difficulties explaining "presentations", "selective disclosure" , ZKP or non-corelability in enterprise software.
- Interoperability is about semantics and not about data models. Custom code is unavoidable. Masquerading customisations behind complex configurations and DSLs is just not such a good idea...
- Customisations are always required for UX, performance,etc. Semantics is always expressed in code! Trust is coming from signed code. Glueing together many DSLs exacerbate the leaking abstraction problem. We should not ignore lessons from "semi-failed" technologies like SOAP & WSDL, XML & Semantic Web
- Golden Hammers: exaggerations in the importance of issuer, hoder, verifier triangle or exaggerated role of messaging protocols and of short living choreographies.

## Identifiers that do not identify...

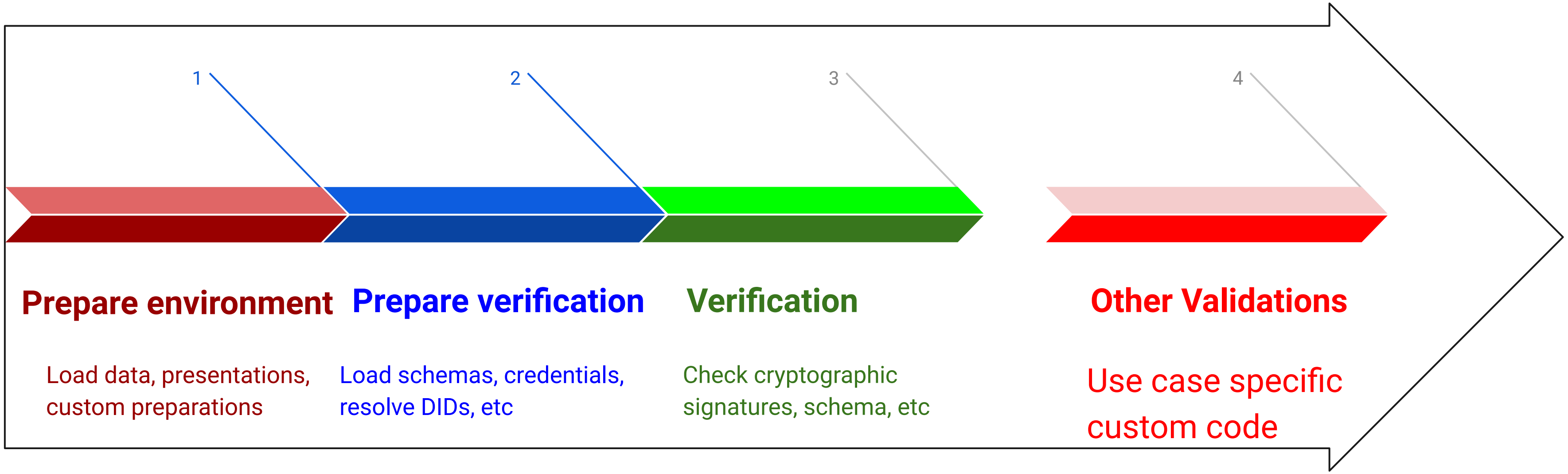


Non-correlating credentials  creates  “**oxymoronic identities**” that are usable only when the transfer of value happens from prover to verifiers and no value is transferred in the opposite sense.  Naturally, the verifiers (or actually the network) have a strong need to check that the prover has enough of that value that it transferred to  verifier.

This seems to severely limiting the use cases (ZKP payments and features for whistleblowers or “trustworthy” fake news creators...). Otherise,  by contradicting the whole thing: it is somehow implied that at least one verifier should be able using external methods to somehow trust enough (or properly identify - correlate) the prover and deliver the services.

Proper identities (with correlating credentials) create  trust for all parties because establish **clear responsibilities** for each participant in a transaction.

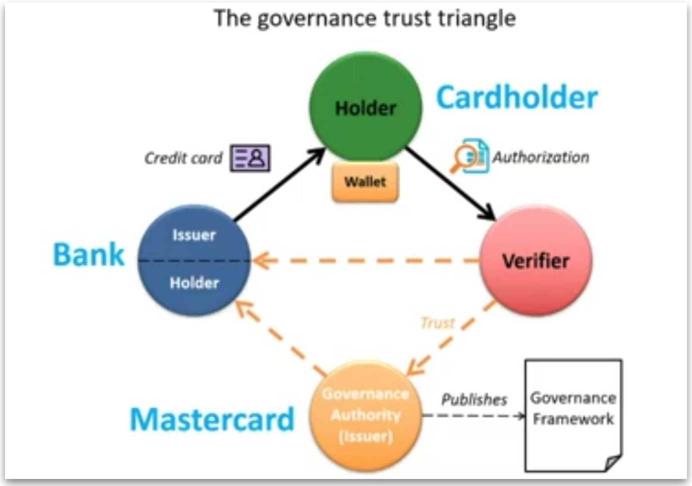
## Validation

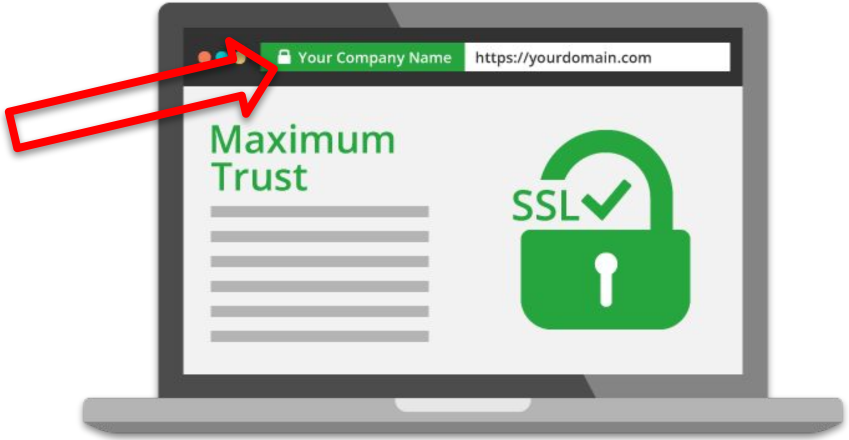

1
2
3
4
Prepare environment
Load data, presentations, custom preparations
Prepare verification
Load schemas, credentials, resolve DIDs, etc
Verification
Check cryptographic signatures, schema, etc
Other Validations
Use case specific custom code

**No silver bullets…** lots of use case specific CODE*

**Schema, Configurations, Credential Definitions, Templates, etc.. are all some sort of DSLs (Domain Specific Languages) -> CODE*

## Trust? Signed code + UI that communicates trust



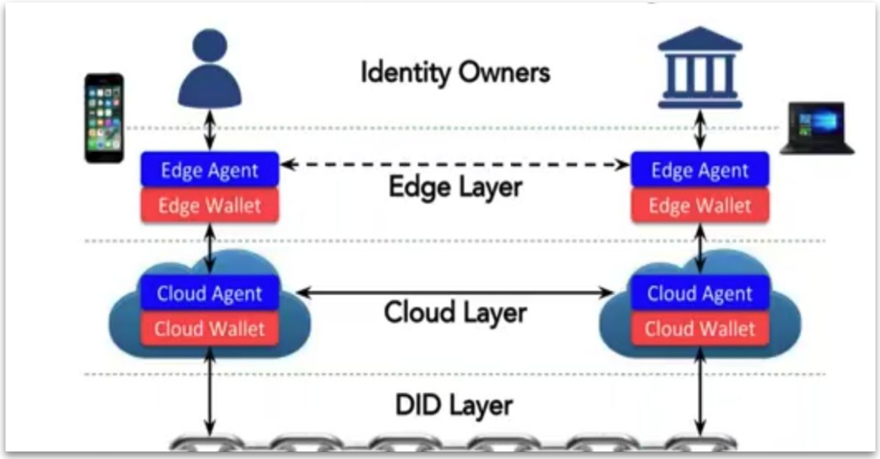


Governance is important but...

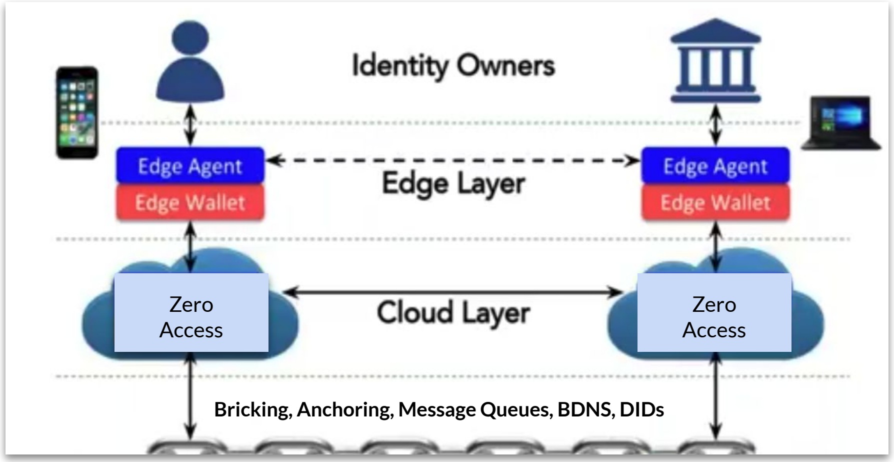
the weakest link is the CODE of the applications

# Towards Zero Access Cloud Infrastructure

## Decentralised Identities Stack (Aries)



## Data Sharing & Identities Stack (OpenDSU)



## Reusable code as Validation Strategies in OpenDSU

- Issue verifiable credentials, X.509 certificates, just sign documents
- Generate specific presentations (when we need presentations)
- Validate proofs on data ( verify signatures, presentations, check other constraints and trust sources)

**Abstracts all the details**

- Data Models & Representations
- Cryptographic primitives

## Conclusions

- **General-purpose languages and not DSLs…**
  - Even for interoperability: migratory code & trust from signed code
- **Wallets should be dynamic and open** (like Web Browsers: capable to load any site)
- **Digital Sovereignty**
  - Starts with decentralised access control (KeySSIs) and Data Sharing
  - Cloud Agents are mostly a plague and a threat for the whole movement...
- **“Validation Strategies” are central**
  - Not a few vague statements in the W3C VC standard
  - Trust and security are such multifaceted problems.
  - Everything is code... Let’s take trust from signing ALL this code!
- **Standards... maybe not so fast**
  - SSIs solutions space is large and open for innovations and darwinian competitions
  - “oxymoronic identities” does not help adoption

## Thank you! Questions?

*"The nice thing about standards is that you have so many to choose from; furthermore, if you do not like any of them, you can just wait for next year's model."*

Andrew Tanenbaum

[www.opendsu.com](www.opendsu.com)
[www.github.com/privatesky](www.github.com/privatesky)