FST Network Workshop

FST Network Team 2019

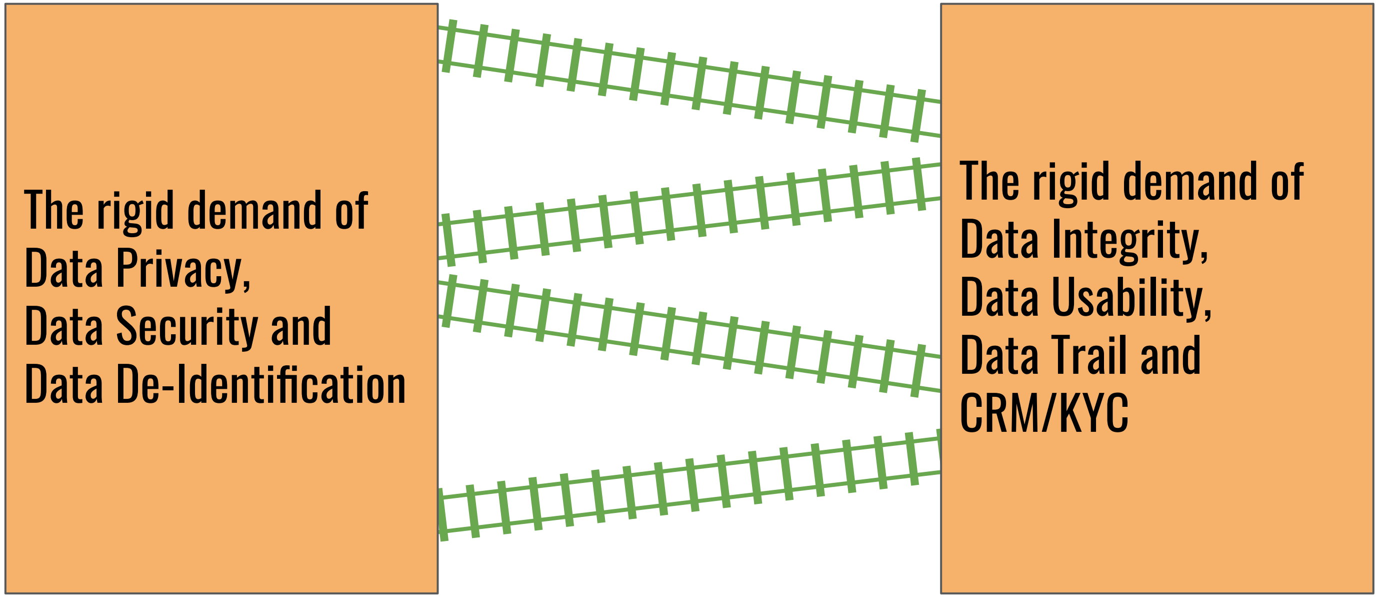
Accountable Data Network
(FST DataRail)

Two Dilemma

The rigid demand of
Data Privacy,
Data Security and
Data De-Identification

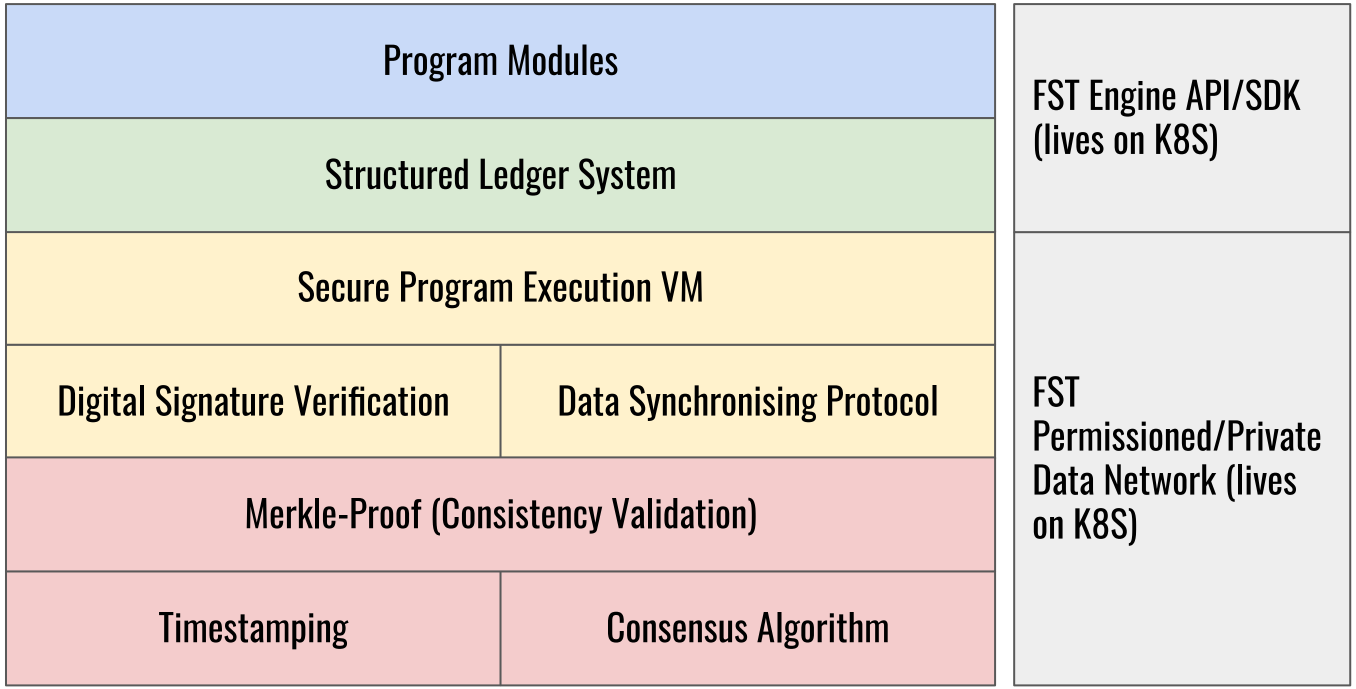
The rigid demand of
Data Integrity,
Data Usability,
Data Trail and
CRM/KYC

Two Dilemma, One Rail



The rigid demand of Data Privacy, Data Security and Data De-Identification
The rigid demand of Data Integrity, Data Usability, Data Trail and CRM/KYC

Accountable Data Network (ADN) arch

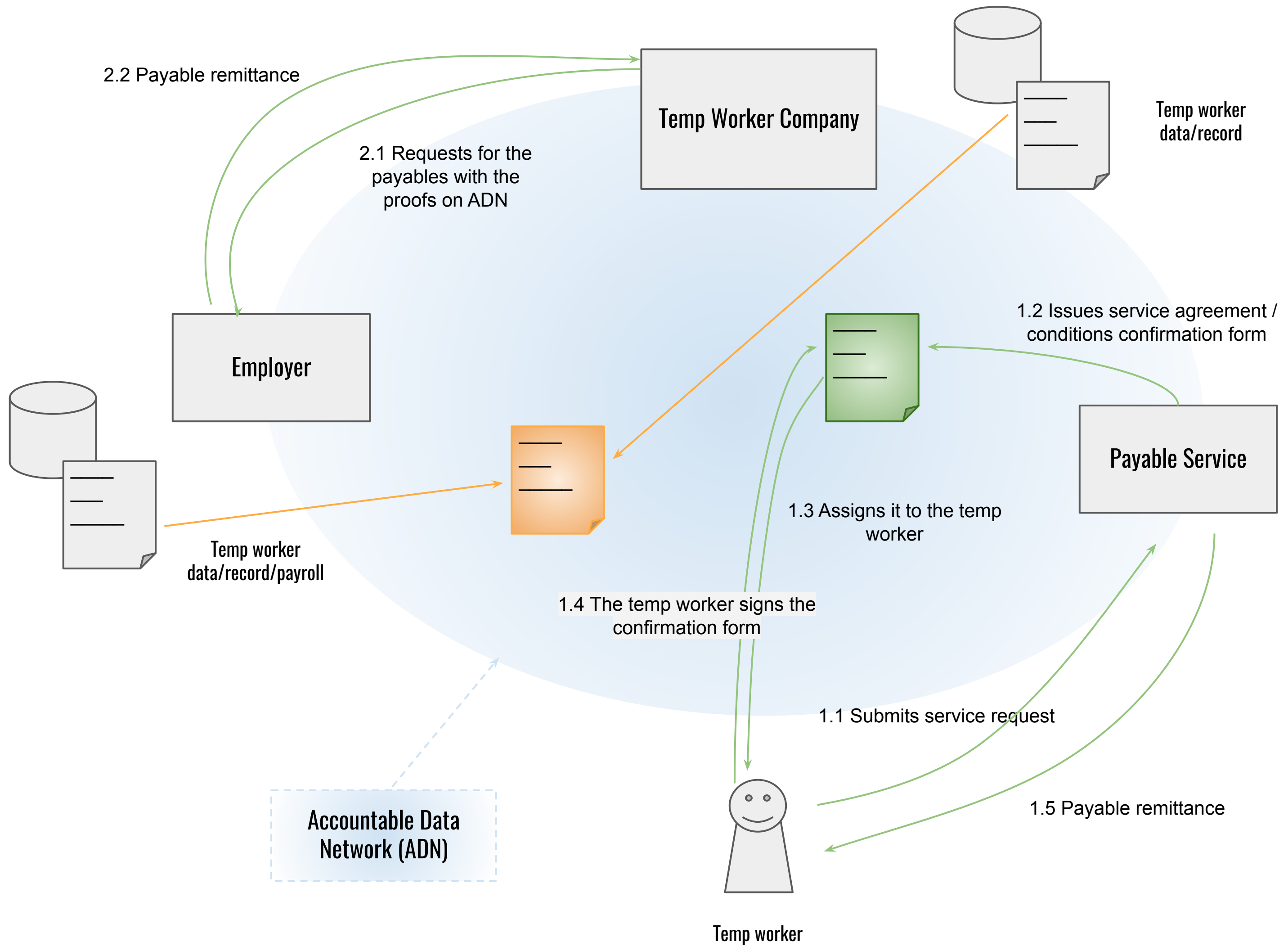


Program Modules
Structured Ledger System
Secure Program Execution VM
Digital Signature Verification
Data Synchronising Protocol
Merkle-Proof (Consistency Validation)
Timestamping
Consensus Algorithm
FST Engine API/SDK (lives on K8S)
FST Permissioned/Private Data Network (lives on K8S)

IAM / ADM

- Accountable Data Management
  - Availability, Usability, Consistency
  - Data Integrity
  - Data Security
  - Data Access across multiple departments/companies

- Digital Signature
- Timestamping

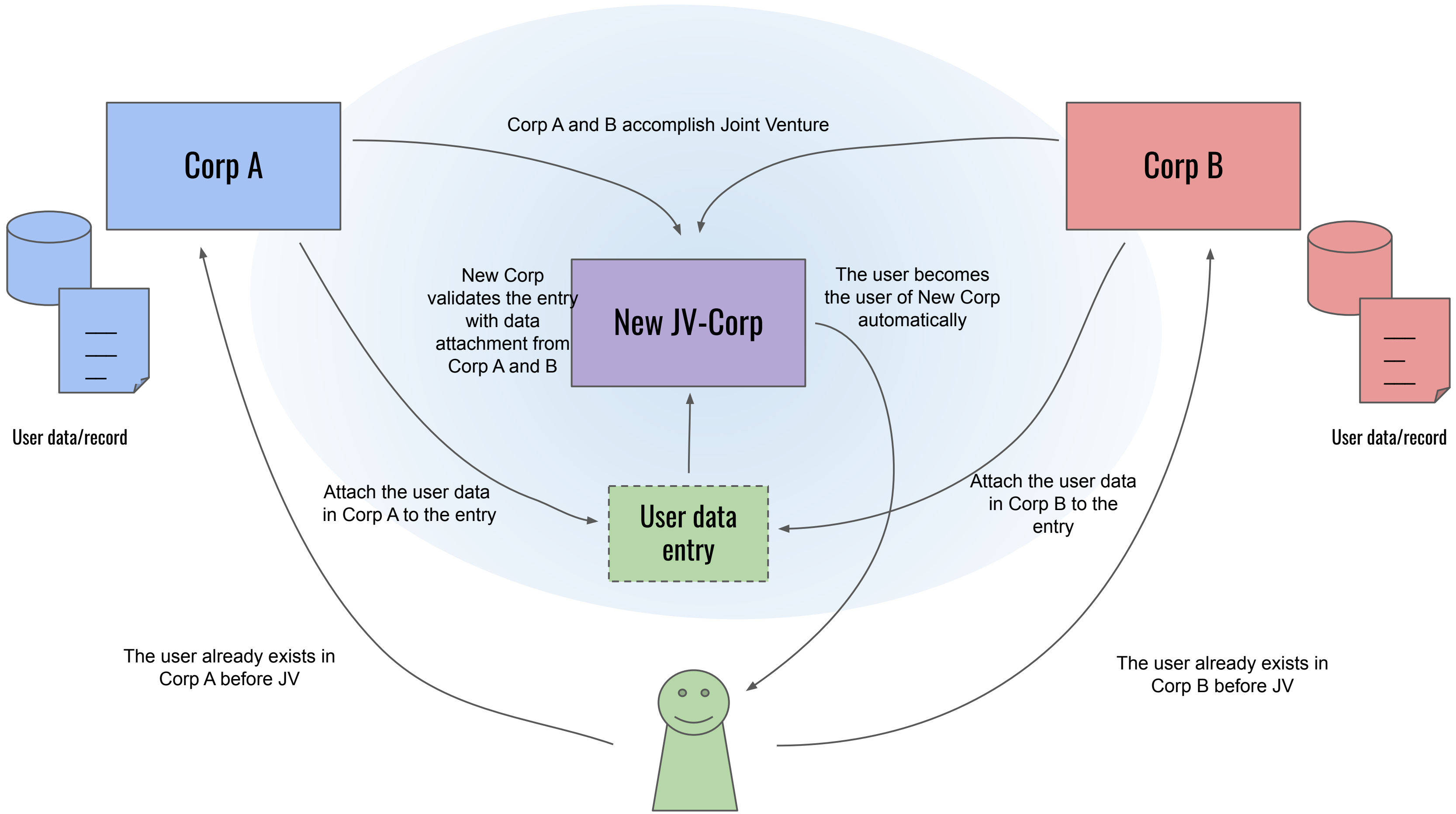

2.2 Payable remittance
Temp Worker Company
Temp worker data/record
2.1 Requests for the payables with the proofs on ADN
Employer
1.2 Issues service agreement / conditions confirmation form
Payable Service
Temp worker data/record/payroll
1.3 Assigns it to the temp worker
1.4 The temp worker signs the confirmation form
1.1 Submits service request
1.5 Payable remittance
Accountable Data Network (ADN)
Temp worker

IAM / CRM

- Customer Relationship Management

- KYC
- Data Privacy
- Compliance

- Natural Identity / Given Identity


Corp A
Corp A and B accomplish Joint Venture
Corp B
User data/record
New Corp validates the entry with data attachment from Corp A and B
New JV-Corp
The user becomes the user of New Corp automatically
User data/record
Attach the user data in Corp A to the entry
User data entry
Attach the user data in Corp B to the entry
The user already exists in Corp A before JV
The user already exists in Corp B before JV

Data Privacy

- An long existing problem of data privacy in
  - cross-department and cross-company data interaction
  - software development, data engineering
  - auditing or proofs of sensitive data usage (GDPR)

- The next sessions will cover:
  - Continuous-Compliance can be done smoothly by highly-consistent and fully-traceable data trail
  - De-Identification of the sensitive data on Accountable Data Network

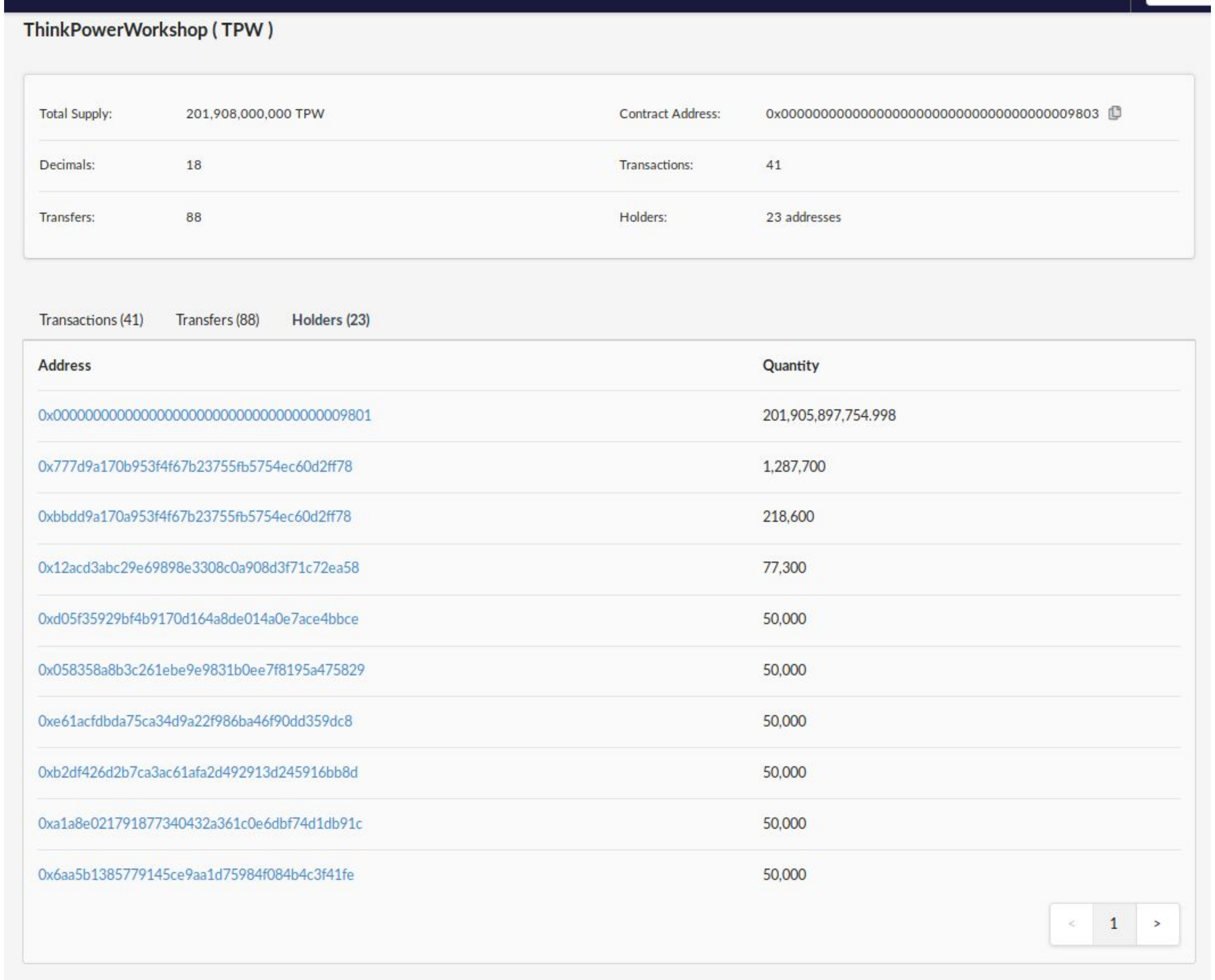
Structured Ledger System

Ledgers

- A Ledger is a Whitelist which is able to:
  - protect every data trail of itself
  - uphold its own authentication and authorisation functionalities
  - record the `ownership:data` key-value and the trail of the key-values' changes
  - manage and utilize the data (key-values) as certificates or volumes of resource usages
  - assure all instructions in itself are highly-consistent and fully-traceable

- Two common ledger storage (mapping) types
  - ownership -> value
  - ownership -> id of the object

- Two common ledger operation types
  - transfer (from, to, value)
  - transfer (from, to, value, call_data)



ThinkPowerWorkshop (TPW)

Total Supply:	201,908,000,000 TPW	Contract Address:	0x0000000000000000000000000000000000009803
Decimals:	18	Transactions:	41
Transfers:	88	Holders:	23 addresses

Transactions (41) Transfers (88) **Holders (23)**

Address	Quantity
0x0000000000000000000000000000000000009801	201,905,897,754.998
0x777d9a170b953f4f67b23755fb5754ec60d2ff78	1,287,700
0xbbdd9a170a953f4f67b23755fb5754ec60d2ff78	218,600
0x12acd3abc29e69898e3308c0a908d3f71c72ea58	77,300
0xd05f35929bf4b9170d164a8de014a0e7ace4bbce	50,000
0x058358a8b3c261ebe9e9831b0ee7f8195a475829	50,000
0xe61acfdbda75ca34d9a22f986ba46f90dd359dc8	50,000
0xb2df426d2b7ca3ac61afa2d492913d245916bb8d	50,000
0xa1a8e021791877340432a361c0e6dbf74d1db91c	50,000
0x6aa5b1385779145ce9aa1d75984f084b4c3f41fe	50,000

< 1 >

Token / Voucher structure

- There are 6 kinds of data types supported in Accountable Data Network:
    - Binary
    - Ordinal
    - Count
    - Value
    - Categorical (Nominal)
    - Time or Interval
- Token Ledger is suit for
    - Value / Categorical (Weak) types
- Voucher Ledger is suit for
    - Binary / Ordinal / Count / Categorical (Strong) / Time or Interval (expiry time set in Voucher) types
- One or multiple Voucher(s) can be published under one Token
    - Structured binding for various data types usages for complex product/service digitalisation

Certificate / Identity / Point System

- Certificate / Identity
  - Binary / Ordinal / Count / Categorical / Time or Interval -> Voucher

- Point
  - More traits in Value -> Token
  - More traits in Ordinal / Count / Categorical -> Voucher

- Certificate and Point type can further be used as Attributes (Tags), and the Attributes need to be De-Identified for Data Privacy

De-Identification

- De-Identification != Anonymisation
  - If a data is de-identified, Human or even AI must not be able to identify the data
  - The methodology of hashing data with salt is quite common

- General De-Identification procedure in Accountable Data Network:
  - Take the unique identity/index of a data or dataset
  - Hash the unique identity/index and take first 20 bytes as the unique de-identified address (entry) on data network
  - Transfer Token/Voucher to the de-identified address as Attributes Attachment
  - The metadata of Token/Voucher can also be de-identified to increase the level of De-Identification

De-Identification with Accountable Data Network

- Actual benefits
  - Accountability Enabled
    - The de-identified data on Accountable Data Network can be fully-traced by the digital-signature and the timestamps of the data creator, uploader and user
  - High Consistency
    - The consistency of the de-identified data can be validated anytime/anywhere without knowing its actual content or intention (since human and AI <u>cannot directly deal with the data which is totally de-identified</u>)
  - Availability/Usability
    - Even the data is leaked from Accountable Data Network, there is no way to identify and utilize the de-identified data. But still, the de-identified data can be used with ADN’s API and SDK with secure authentication and authorisation, and the data leakage can be proved by the trail on ADN
- Problems
  - How to make the procedure of data de-identification manageable and smooth ?
  - How to perform secure data de-identification ?
  - How can the multiple parties cooperate data interaction based on the de-identification ?

Attribute-based Access Control (ABAC)

- The general way to check an user’s permissions in ABAC:
  - Fetch all the attributes needed under the user
  - Any data record or history or any data in another system can be attributes
  - Do if/else or rule-engine to check if the attributes under the user satisfy requested resources and actions’ configuration
  - For a new service or for an unknown situation, ABAC can still work well since it only reacts to the attributes it needs

- v.s. Role-based Access Control (RBAC)
  - Any kind or combination of the permissions needs to bind to one role
  - An user must have at least one role
  - If an user’s permissions need specific adjustments, then the original role might be affected or the user needs to be assigned one specific role

ABAC with Accountable Data Network

- Actual benefits
  - Accountability Enabled
    - The attributes generated/attached by various companies can be fully-traced by the digital-signature and the timestamps
  - High Consistency
    - The consistency of the attributes can be validated anytime/anywhere no matter how numerous or complicated of the attributes
  - Availability/Usability
    - To utilize ABAC in complicated or unknown environment, the nature of data integrity/security/accessibility must automatically take effect

- Problems
  - What if the data entry (e.g. users, goods) is sensitive ?
  - What if the attributes of the data entries are sensitive ?
  - Does the High Data Consistency conflict with Data Privacy ?

FST Engine API/SDK

Glossary

- FST DataRail / Accountable Data Network
  - An unique network mechanism that insures the consistency, insertion order and existence of any data
- Address
  - An entry on Accountable Data Network, each address can actually belong to one private key, but it is cryptographically difficult to be reverse-attacked, since the address is the hash of the public key, and there are 16^40 addresses available
- Contract / Smart Contract
  - A set of program modules deployed on accountable data network with their business-logics and storages are computed/executed exactly under their pre-configurations and constraints. A Smart Contract has its own address.
- Ledger
  - A kind of Smart Contract that stores the Accountable Data and verifies the authentication/authorisation of any operation on it.
- ERC-20
  - An interface for basic ledger operations and storage
- ERC-721
  - An interface for basic non-fungible ledger operations and storage

Glossary

- ERC-1376
    - An interface for advanced ledger operations and storage with optimisations and high-extensibility, compatible to ERC-20 and ERC-721
- Token / Smart Token / Fungible Token
    - An ERC-1376 implementation with fungible, 18-decimals ledger (1 Token = 1 * (10^18) value stored in Token Ledger).
- Voucher / Smart Voucher / Fungible Voucher
    - An ERC-1376 implementation with fungible, 0-decimal ledger (1 Voucher = 1 value stored in Voucher Ledger). Each Voucher ledger can be set its own expiry (UNIX Epoch time format).
- EthereumKey / Wallet File / Key File / Key Storage
    - A JSON that contains encrypted ECDSA private key, the passphrase hashing is used with Scrypt/PBKDF2
- Gas / Fuel
    - A protocol-level and service-level gas that protects the entire Data Network from the DDOS attack, since any instruction on a Data Network must consume gas. In FST PPB/Engine, the protocol-gas is called Ether, and the service-gas is Master Service Gas
- Transaction
    - A data set contains digital-signature, timestamping, target address, gas amount and the bytecode to the Smart Contract
- Transfer
    - A balance change or data transferring in a ledger, it contains ***from*** (originator), ***to*** (receiver) and the ***value*** (data amount)

Glossary

- Password
  - The credential for the authentication in sign-in
- Passphrase
  - The credential for decrypting the EthereumKey (Wallet File / Key File) to use private key for transaction signing

- Master
  - The user that manages and control the whole white-labeled platform (FST Engine)
- Issuer
  - The user that is authorised to issue token and publish voucher (ledger creation). And only Master can authorise Issuers to the create ledger (issuing token) by transferring IL (Initialisation License for Ledger Creation) to Issuers.
- End-User
  - The common user that can use and control its own wallet

FST Engine API Categories

- For Master
- For Issuers
- For End-Users or Anyone

- API Doc

APIs for Master

- Create Issuer/End-User
  - Master can create issuers or end-users on its own, the permission of the issuers or end-users will be set during the creation. Please don’t forget to distribute enough protocol-gas (Ether) to newly created Issuer or End-User.
- Get Issuers/End-Users
- Transfer Master Token (needs signing)
  - Master can transfer its Master Token to Issuers who needs to manage smart contract modules on FST Engine, the usage of smart contract modules consumes Master Service Gas (Master Service Gas can be converted from Master Token, it is for protecting the ADN and FST Engine). In other words, Master Token is a certificate/authorisation to use smart contract module
- Mint IL (Initialisation Lisence for Ledger Creation) (needs signing)
  - Master can transfer IL to Issuers to allow them to issue their own main ledger (Token)
- Purchase Master Service Gas (needs signing)
  - Master can also directly convert its own Master Token to Master Service Gas and fill Master Service Gas to Issuers.

APIs for Issuers

- Create End-User
  - Issuer can create end-users on its own, the permission of the end-users will be set during the creation. Please don’t forget to distribute enough protocol-gas (Ether) to newly created End-User.
- Get End-Users
- Issue Token (needs signing)
  - Issuer must create its main ledger (Token) before using more smart contract modules in FST Engine. This action will consume 1 IL, and one issuer can only issue one token in life time.
- Purchase Master Service Gas (needs signing)
  - Issuer can convert its own Master Token to Master Service Gas and fill Master Service Gas to itself. The balance of Master Service Gas can be checked via Blockchain Explorer API
- Publish Fungible Voucher (needs signing)
  - Issuer can create sub ledgers (Vouchers) whenever its Master Service Gas balance is enough, each Voucher publishing will consume 600+ Master Service Gas (every single voucher per available day consumes 0.00003 Master Service Gas)

APIs for End-Users or Anyone

- Blockchain Explorer APIs
  - Every operations, ledgers or accounts can be fetched via Blockchain Explorer API
- Get own Ethereum Key
  - The requester of this API can get and only get its own Ethereum Key (encrypted wallet file)
- Transfer Ether (needs signing)
  - The requester of this API can send its Ether.
- Transfer Fungible Token (needs signing)
  - The requester of this API can send its Token.
- Transfer Fungible Voucher (needs signing)
  - The requester of this API can send its Voucher.

FST Engine Authentication/Authorisation

- Password Authentication
    - Anyone can perform sign-in to get its own JWT to use FST Engine API

- RSA/ECDSA Authentication
    - For imported End-Users, the Master or Issuer can help End-Users generate JWT (with Master or Issuer’s digital signature) to perform Cross-Domain Authentication and Authorisation

- Authorisation
    - Master
    - Issuer
    - End-User

Key Management Best Practice

- DOs:
  - Do let the end-user set the passphrase of its wallet file (key file)
  - Do compute and generatethe wallet file (key file) on client’s side
  - Do store the wallet file (key file) on server (cloud) side safely as backup
  - Do store the private key in KMS on cloud if needed

- DO NOTs:
  - DO NOT store the passphrase of the wallet file (key file) on server’s (cloud) side
  - DO NOT store the private key decrypted from the wallet file on server’s (cloud) side

APIs Usage & Modeling

Prerequisite

- Google Chrome / Firefox / Opera (at least one)
- Node LTS (v10, node-gyp installed)

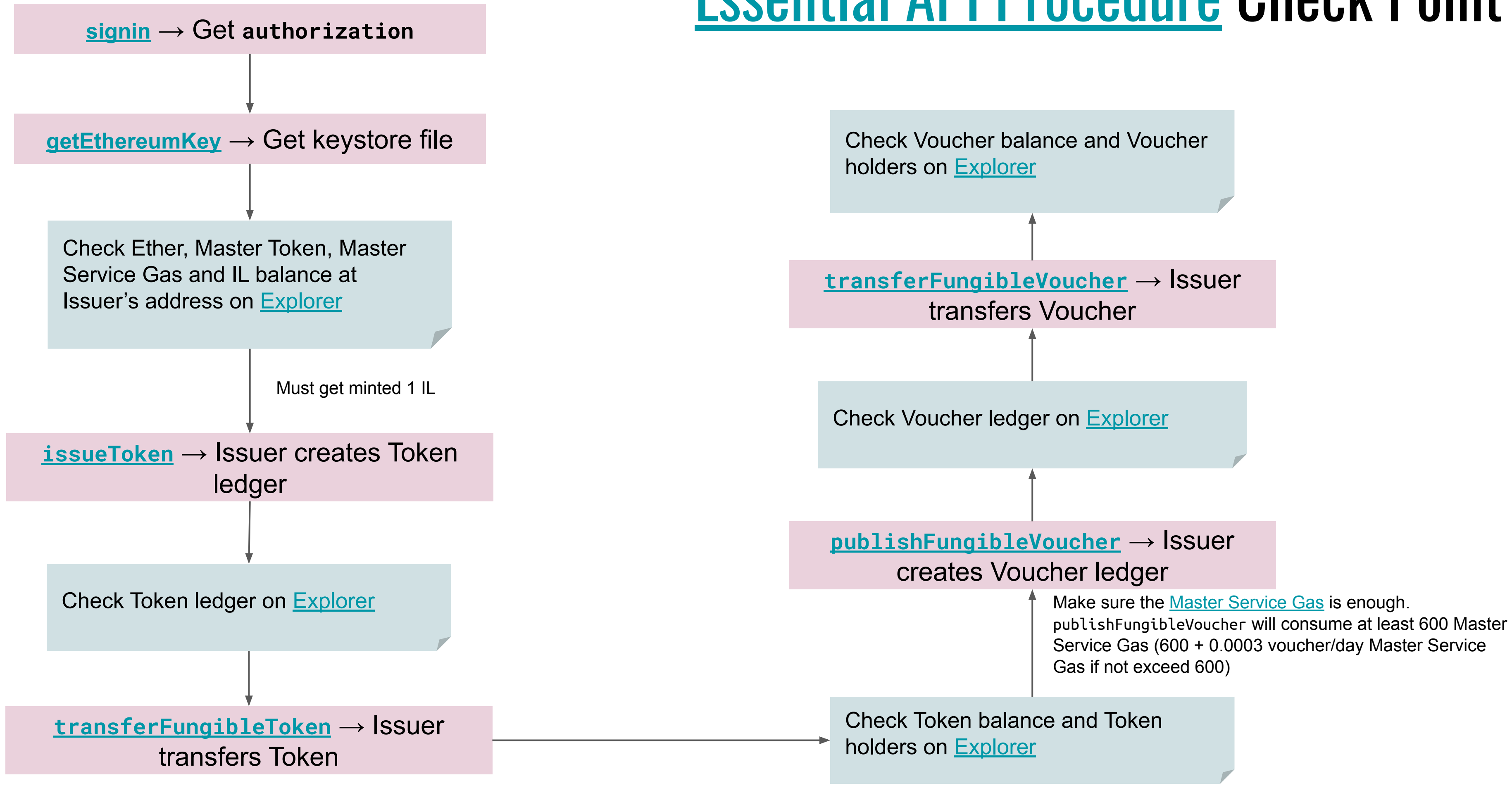
Tools

- Workshop API Doc
- FST Engine API GraphQL Playground
- Blockchain Explorer
- Simple FST Engine interaction toolkit

Overall structure of Data Modeling in FST DataRail

1. Decide what to track and audit
2. Assign Data Types to those to be tracked and audited
3. Design ownerships (who should be able to control those) with other Data (Attributes)
4. Decide the parameters for De-Identification
5. Create the authority/manager of the Ledgers (the issuer)
6. Create the Ledgers (de-identify the metadata of the Ledgers if needed)
7. Distribute/Attach the data in Ledger to the receivers, targets or any data entry (de-identify if needed)
8. Trace/Fetch the data trail from Data Network Explorer
9. Re-Identify those metadata/indices in data trail safely with original data storage

Essential API Procedure Check Point



signin → Get `authorization`

getEthereumKey → Get keystore file

Check Ether, Master Token, Master Service Gas and IL balance at Issuer's address on Explorer

Must get minted 1 IL

`issueToken` → Issuer creates Token ledger

Check Token ledger on Explorer

`transferFungibleToken` → Issuer transfers Token

Check Token balance and Token holders on Explorer

Make sure the Master Service Gas is enough. `publishFungibleVoucher` will consume at least 600 Master Service Gas (600 + 0.0003 voucher/day Master Service Gas if not exceed 600)

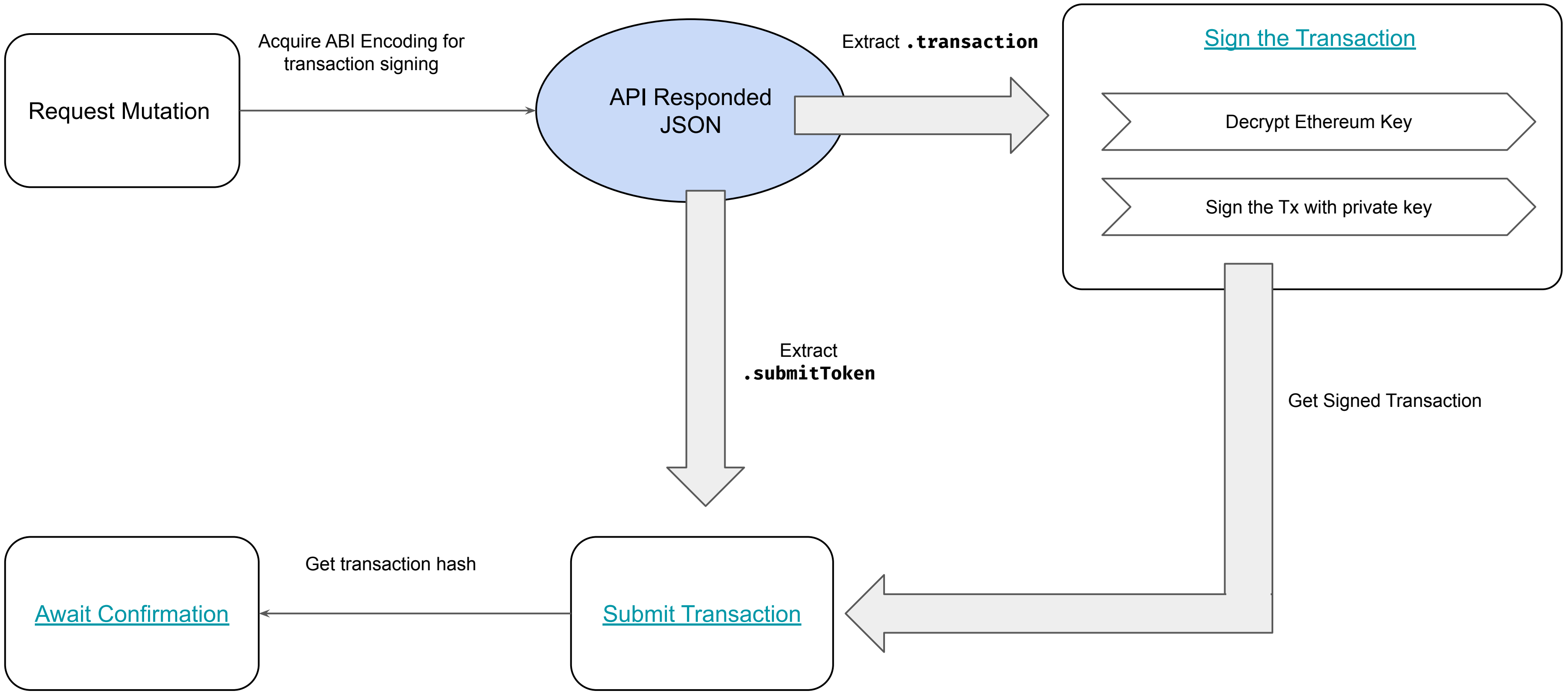
`publishFungibleVoucher` → Issuer creates Voucher ledger

Check Voucher ledger on Explorer

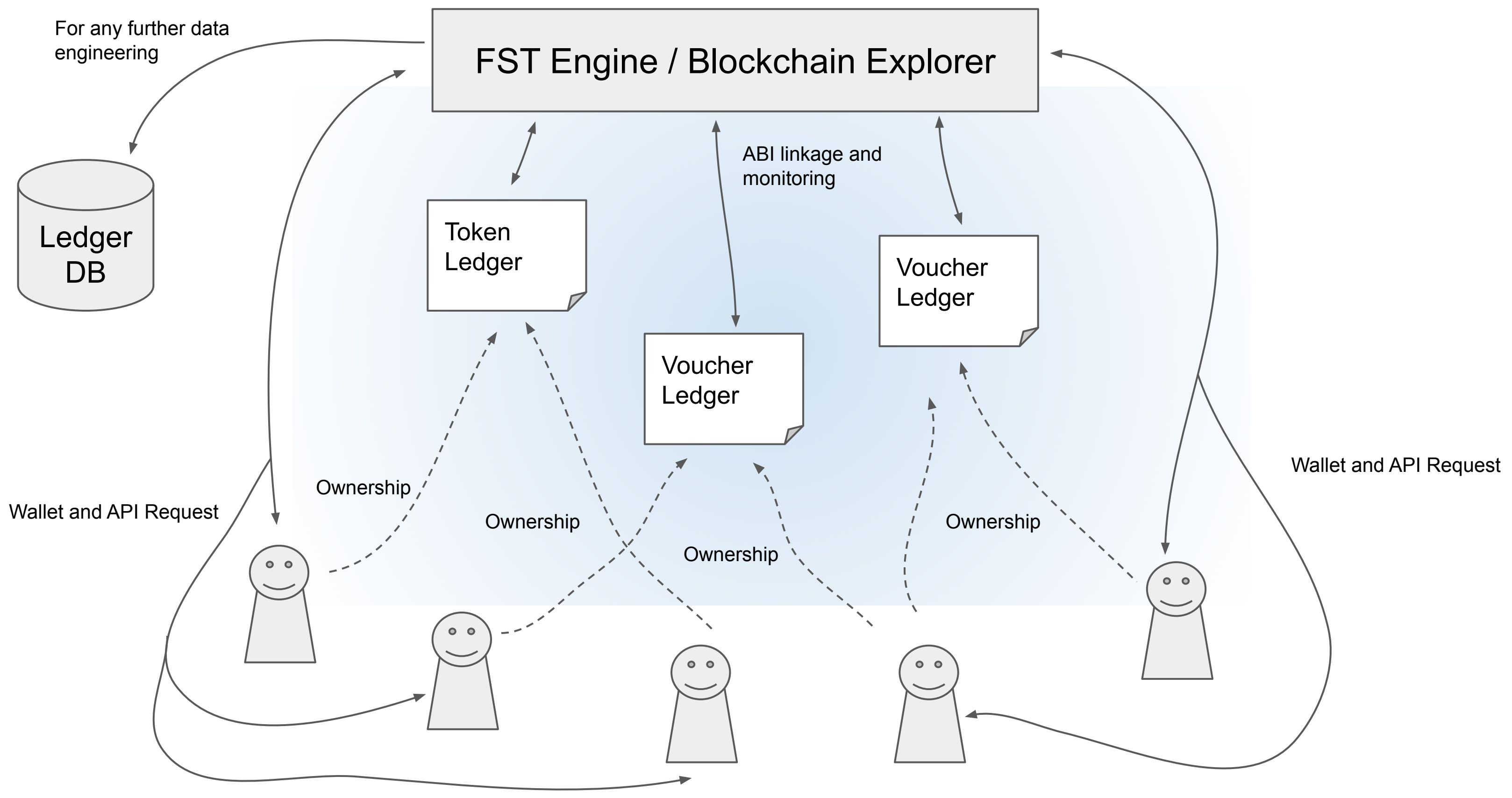
`transferFungibleVoucher` → Issuer transfers Voucher

Check Voucher balance and Voucher holders on Explorer

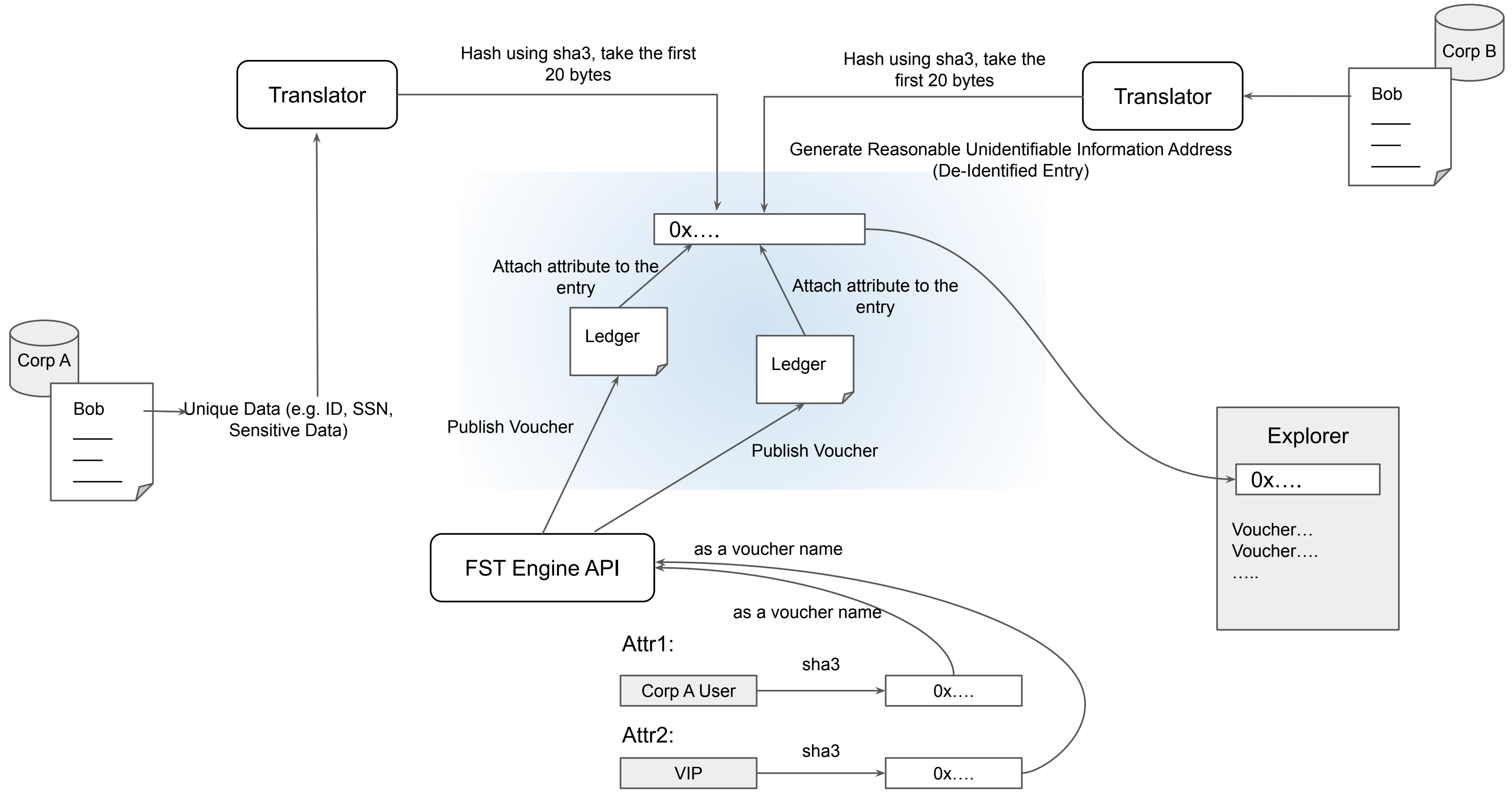
Procedure for signing/submitting the transaction



Certificate / Point System



De-Identified Data Interaction System


Corp B
Bob
Translator
Hash using sha3, take the first 20 bytes
Translator
Hash using sha3, take the first 20 bytes
Generate Reasonable Unidentifiable Information Address (De-Identified Entry)
0x….
Attach attribute to the entry
Attach attribute to the entry
Ledger
Ledger
Corp A
Bob
Unique Data (e.g. ID, SSN, Sensitive Data)
Publish Voucher
Publish Voucher
Explorer
0x….
Voucher…
Voucher….
…..
FST Engine API
as a voucher name
as a voucher name
Attr1:
Corp A User
sha3
0x….
Attr2:
VIP
sha3
0x….