Prople: The Personalization Platform

Abstract

This paper introduces a *Prople* project, it's a *Personalization Platform*. It combines experiences from Web2 and Web3 designed specifically for the user's personal assets and activities. It manages three user's core domains:

- Identity
- Social
- Finance

For the `Identity`, this project implements the `Decentralized Identity (DID)` standard and derives the functionality of *SSI (Self Sovereign Identity) Agent* to its own computation agent called *Prople Vessel*. The *Vessel* is a combination of functionalities of *SSI Agent* and also *PDS (Personal Data Storage)*, which is used to save and compute data. It's like a *Pod* from Solid Project.

To support wide adoption, it will also provide *Vessel Service Provider (VSP)*, which is an open and decentralized network where people can help to participate to form the *PAIR (Prople Agent Identity Registry)* decentralized network and a cluster of *Vessel*, so a user with the non-technical background will able to use this platform without the need to set up any technical things.

For the `Social` activities, it will adopt current existing standards for the decentralized social network, *ActivityPub*, which has already been implemented in the current well-known decentralized network, *Mastodon*.

For *Finance*, it will adopt cryptocurrency to help the user manage their own financial assets in the decentralized and P2P networks. Specifically for this core domain, it will focus on **Chain Abstractions** and **Network Interoperability** rather than creating or managing its own crypto network.

For its *base network*, it will be using NEAR Protocol, because this network protocol has the same mission and vision as the **Chain Abstraction** and **Network Interoperability** including already providing base technology components so we can focus on our domains, this network protocol also already has a good performance and security standard as the crypto network.

Other networks to support *interoperability* like:

- ZetaChain
- Omni Network
- Axellar Network

The core foundation domain for this platform is the *Identity*. As long as a user or an actor is able to maintain and has *true ownership* of their identities, they'll be able to own and manage anything that important to them. The mission and vision of this project is to bring back the power and control back to the people which is not controlled by any single entity, organization, company, or institution.

Introduction

Today, the software is almost anything, everything, and everywhere, the software is eating the world. Almost all human activities today cannot be separated from software. We have software to deliver our food, help us write a personal blog post, pay for anything, and even detect and monitor our heartbeats. Today's software is designed to penetrate more deeply into our lives as individuals.

Today's human interactions are also mediated by software, like real-time messaging apps, dating apps, and social networks. We as a human interact with others through some digital intermediaries where people gathered there in the some digital platforms.

For the business and all financial activities, almost everything is done through the digital intermediaries too. We have a digital bank and digital payment gateways. Almost all of our financial transactions happen through some software belonging to some institutions, no matter it's an old institution or a new digital institution.

Social, business, and finance activities will always be attached to an actor, whether it's a user, organization, or even just a bot (the other software), which means, we must know who the activity or the asset belongs to whom. The answer is, that we need a *unique identity*. Without this *unique identity*, we will not know the activity's actors or the asset's owners. Imagine there is a gold bar saved in some bank's vault without any identities to show its owners, Does it mean that everyone will have the right to own that gold?

Identity is the core foundation of human activities. Without it, we don't know when we are talking or communicating with whom. Without it, there is no ownership of something in this world. Without it, everything will be ambiguous, we need a *unique identifier* to eliminate this ambiguity.

All of these components have already been provided by current institutions or tech companies. Their software already provides solutions for social, business, and finance activities and assets, by providing their own *Identity Management*.

If a user wants to interact with some digital platforms or networks, for whatever reasons, either for social or business activities, the first thing they have to do is to create their *unique identity* on those platforms or networks. All of the user's activities and assets will belong to their *identity* registered on that specific platform. If a user needs to access two or more platforms or networks, it means they have to create their *identity* on those platforms, one by one. This is the reason why we have open standards for *identity management* like *OpenID*.

The real problem is not the *interoperability* or the standards, but who owns and controls the identity.

Users' identities will be saved on the platform or network databases, a storage that a user will not have any access to it. User will only be able to use their identity through the platform's application or software. This platform or organization will own it, they own, control, and manage it. This institution or organization will have enough power and rights to do anything to their user's identity, like edit or even remove it. It's because the data is saved on their platforms. Once user's or actor's identities are owned by some intermediaries or third parties, all of their activities and assets are controlled by these organizations.

There is no true ownership, no privacy, and no control for the individual over their own identities, activities, and assets.

Then came an era when *Bitcoin* was born, it's the first decentralized and P2P cryptocurrency network. A network where people have their own *unique addresses* and are able to maintain their *crypto assets* is called *BTC*. This *unique address* indicates the user's *unique identity* on that network, and it's secured cryptographically. At this network, user are able to send their asset (*BTC*) through their unique addresses. And because it's P2P and decentralized, there are no centralized organizations or institutions that own or control them.

Prople is an *open personalization platform* that aims to help *actors* to manage three important domains:

- *Identity*
- *Social*
- *Finance*

It has a *Prople Vessel* which is an *Actor Agent* its functionalities are derived from the concept of *SSI (Self-Sovereign Identity) Agent*. The *Vessel* is used as a *digital representation* of an *actor*. It should be able to be online as long as 7x24x365 and help an *actor* manage their online activities and assets.

Prople Vessel is a software that has functionalities that have a combination of these three things:

- *SSI Agent*, used to manage identities
- *Social Agent*, used to manage any social activities such as real-time communication
- *Crypto Wallet*, used to manage financial activities and assets

Prople has a mission and vision to give back true ownership and control back to the people.

Background

The Evolution Of Web: Read - Write - Own

Web1 - Read

The era of *Web1* it's like the beginning adoption of the *Internet*. The *Internet* is a result of interconnected computer networks. *The Internet Protocol* is a set of rules used to communicate between networks and devices on the network.

When the Internet was introduced for the first time, there was a problem raised which was *sharing information*, like what already been described by *The Father of the web: Sir Tim Berners-Lee*:

In those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you have to learn a different program on each computer. Often it was just easier to go and ask people when they were having coffee…

Short story, Tim introduced a new technology at that time, called *hypertext*. The technology became what we know today as *HTML (Hyper Text Markup Language)* and the beginning of the *Web1* era.

A those times, the *web* provided a limited and static user experience, where a user was only able to *read* the information, Tim tried to solve the problem at the time this technology was introduced, which was, sharing information.

With the technology, users are able to read anything on the Internet in this era.

Web2 - Write

The problem evolved, from limited and static user experiences, becomes the real-time interactions, where a user is able to share anything on the Internet. They're not just reading the information, but also able to create the information and get the feedback in real-time manners, it produces more interactive user experiences, where users are able to interact with others.

Many tech companies start to raise at this time:

- - MySpace
- - Facebook
- - YouTube

Web 2.0 is about creation and sharing of information delivered via the web.

Web3 - Own

There is some ambiguity between the `Web3` and `Web 3.0`. The `Web 3.0` main concept is about "the semantic web", which tries to make structured data available on the Internet. But in this paper, we will use the term of *Web3*.

Taken from The Web3 Foundation:

A decentralized and fair internet where users control their own data, identity, and destiny. Users own their own data, not corporations. Global digital transactions are secure. Online exchanges of information and value are decentralized.

The *Prople* inspired and designed on top of these missions and spirits, where *Prople* provides a solution to three core domain problems:

- *Decentralized Identity*
- *Decentralized Social*
- *Decentralized Finance*

The only mission of *Prople* is to give back ownership and control back to the people.

Related Projects - Decentralized Movements

SolidProject

The objectives of this project are :

Your data, your choice. Solid is an open standard for structuring data, digital identities, and applications on the Web. Solid aims to support the creation of the Web as Sir Tim Berners-Lee originally envisioned it when he invented the Web at CERN in 1989. Tim sometimes refers to Solid as “the web - take 3" — or Web3.0 — because Solid integrates a new layer of standards into the Web we already have. The goal of Solid is for people to have more agency over their data.

This project was initiated by *The Father of the Web: Sir Tim Berners-Lee*, and the objective was to make *Web 3.0* become a reality.

Ref: https://solidproject.org/about

AT Protocol & BlueSky

The tagline of this project is: *The Social Internet*

The AT Protocol is a networking technology created to power the next generation of social applications

BlueSky itself is a social application that provides a global-scale microblogging experience. This project introduces a decentralized social network protocol, in which the network is not a private network owned by a single corporation, but uses a *federated network*.

This platform *identity management* already implements the `DID (Decentralized Identity)` standard.

Ref: https://atproto.com/docs

Mastodon

Mastodon is a well-known project that also introduces *decentralized social media*.

Social networking that's not for sale. Mastodon provides you with a unique possibility of managing your audience without middlemen.

The decentralized social networking protocol used at this platform is `ActivityPub`.

Ref: https://joinmastodon.org/

Lens Protocol

This social network runs on top of the *Polgyon POS* network protocol and will have its own network secured by *Ethereum* built on the ZK Stacks.

Lens is an open social network that allows users to own their content and connections

This project introduces an interesting concept of *Profile NFT*.

The Profile NFT, a key element of the Lens Protocol, grants you control over your social graph and content. An address can own multiple Profile NFTs, each maintaining a record of all posts, quotes, mirrors, comments, and other content created.

Ref: https://www.lens.xyz/docs

Evaluation

The Open Standards

There already existed protocol standards for the `Identity` and also `Social Network`, which are:

- *Decentralized Identifiers (DIDs)*
- *ActivityPub*

Both of these projects have already been official W3C recommended standards:

- W3C: ActivityPub
- W3C: Decentralized Identifiers

The Chain Abstractions & Network Interoperability

Today, we have so many L1 / L2 networks in the cryptocurrency ecosystem, and the problem of these multiple chain networks is about *interoperability*, because each of network will be designed and built using different standards.

The need for *chain abstraction* is really important in the cryptocurrency ecosystem. Too many networks available will make the user use multiple wallets to manage their multiple identities and assets.

There are several interesting projects that designed to solve this *interoperability* and *fragmented* issues. Some of them are:

- NEAR Protocol
- Zetachain
- Axellar Network

Prople Solutions

Unlike *AT Protocol* which tries to create a new protocol, *Prople* will leverage current existing standards. To solve three core domain problems, *Prople* will use these standards:

- - *Decentralized Identity* : implement *W3C DID* & *SSI (Self Sovereign Identity)* management
- - *Decentralized Social* : implement *ActivityPub* and *IPLD*
- - *Decentralized Finance*: will focus on *chain abstractions* and *interoperability* networks, which in the beginning will run on top of *NEAR Protocol* and use other interoperability tools and networks like *Zetachain* and *Axellar*. Specifically for financial technology, *Prople* will only focus on off-chain computation and tools.

The *Prople Vessel* is heavily inspired by the *Solid Pod*. The difference with it is about the functionalities, which `Vessel` offers more functionalities. It's not just for *personal data storage*, but a combination of *Crypto Wallet* and *SSI Agent*.

The *Prople Vessel* is designed to be a *container* that is deployed on server environments, owned and controlled by a single *actor*. A user may have multiple *vessel agents* that deployed either in the same or different servers and even users able to deploy it on their *localhost*. User will be able to be *connected* with others through their *vessel agents*. This single *container* will have functionalities to manage all three core domain problems.

Unlike the *Solid Pod*, where users will required to set up their *Pods*, there is a concept of *VSP (Vessel Service Provider)*. This *service provider* is like an operator that wants to set up *VSP* on their environments and be connected with other *service operators*. These *operators* will interconnect with each other to form a *decentralized and P2P network* of *PAIR (Prople Agent Identity Registry)*. The *VSP* operator is like a *node operator* or *validator* in public blockchain networks which need to run a *node* that connects to each other to form *decentralized & P2P cryptocurrency networks*. By providing this decentralized network, any non-technical users, will be able to adopt this technology and set up their own *Prople Vessel Agent* through this network.

The Personalization Platform

Overview

There are evolutions in Web technologies, from the `Web1` to `Web3`. The evolution flows become like this:

Read -> Write -> Own

There are also movements about our digital behaviors:

- From the *account balances* to a *personal wallet*
- From the secured digital organizations to the cryptographically secured
- From the *username* to the *public key* or *public address*
- From the *password* to the *private key* or *seed phrase*
- From the *centralized social network* to the *decentralized social network*

Prople is about the *Personalization Platform*. Which an *actor*, has a *true ownership* of their *assets*. They'll be able to maintain and manage their digital life activities.

The Rise of Cryptography

The *personalization movement* doesn't need to eliminate any organization or institution. The difference is through this movement, an *actor* will host, own, and manage their *identities* including all their assets. There are still possibilities for organizations/institutions to work and integrate side by side with any *actors* without the need to control or own the *actor's identity* or *actor's assets*. One of the key reasons behind the organization or institution must host and control their users' identities and assets is because of the *verification process*. By hosting their user's identities and assets, the organization or institution will able to *verify* any of their user's activities including their assets.

Since the rise of *Bitcoin*, the first decentralized and P2P cryptocurrency network, and then *Ethereum* as the first *decentralized shared computation*, at the same time, it also the rise of cryptography.

Cryptography is all about securing information and communication. It's like writing in a secret code that only authorized people can understand.

Through the cryptography algorithms, there are two important concepts:

- Encryption / Decryption
- Digital Signature

Bitcoin, *Ethereum*, *Solana*, *NEAR Protocol*, all of these blockchain networks depend on cryptography algorithms, especially for *asymmetric encryption*. Encryption is the core of cryptography techniques, including *digital signatures*. By using cryptography, we're able to verify the actor's identities and assets, through its digital signatures, without the need to host the actor's data in some organization or institution database vault.

There is no need anymore, for any *actors* to create their *credentials* like *usernames* and *passwords* to many organizations, what we need now is just to share the actor's *public key*. Through this key, an organization or institution will be able to verify any *digital messages* including *digital assets* and *digital activities*.

The *message exchanges* between an *actor* and the *organization* also able to be encrypted, it provides a more secure communication channel between parties.

The Rise of Decentralized and P2P Networks

Bitcoin is not just the first *public permissionless* blockchain network, it also proves that important transactions like sending a *value* between two parties is able to be executed without any intermediaries or third-party entities. There are no *centralized entity* or *exclusive intermediaries*, it is just a network which not controlled and owned by any single organization, and even it's more secure than any *centralized entity*.

In software engineering, there is a term called *SPOF (Single Point of Failure)*, it's a concept where there is a single part or component in the system that if this part/component fails, it will stop the entire system. That's the risk of a *centralized entity*. *Bitcoin*, *Ethereum,* and other several public blockchain networks are all secured by *decentralized networks*. As long as there is a *node operator* that stays alive, the network is alive.

There are no SPOF in decentralized networks

As has already been explained above that there are no *centralized organizations* own *Bitcoin*. If there are no organizations or institutions, how do they manage their networks? It's *easy* to say when we are building a *private / enterprise network* like *Microsoft*, *Google* or *Facebook* to maintain coordination, maintain deployment. How to do all of that in the *decentralized networks*? How to maintain coordination? And the most important thing is, how to maintain *trust*.

The answer is *peer-to-peer networks*. There are only *node operators* and *validators* exist in the *Bitcoin* networks or other public blockchain networks. Each of the available *peers* communicates with each other through some defined *communication protocol*, which is a set of rules that have already been defined. All of these peers also *watch* each other, they'll *verify* all incoming transactions through some defined *consensus algorithms*. If a single *node* fails, there are still hundreds, or even thousands of them still available online, running independently. Each of the *peers* will also store the same data, which means, that if there is a single *node* fail, the data is still available in the next hundreds or thousands of other nodes. The data will be *distributed* and *replicated* to all available nodes.

Prople: The Personalization Platforms

First things first, *Prople* is not a *public blockchain network*. *Prople* is about the *decentralized network*, the *platform,* and the *ecosystem*, which is designed specifically to give back freedom, control, and *ownership* to *actors*, whatever *actor* means, it can be an individual or person, an organization or institution, a device, or even a *bot*.

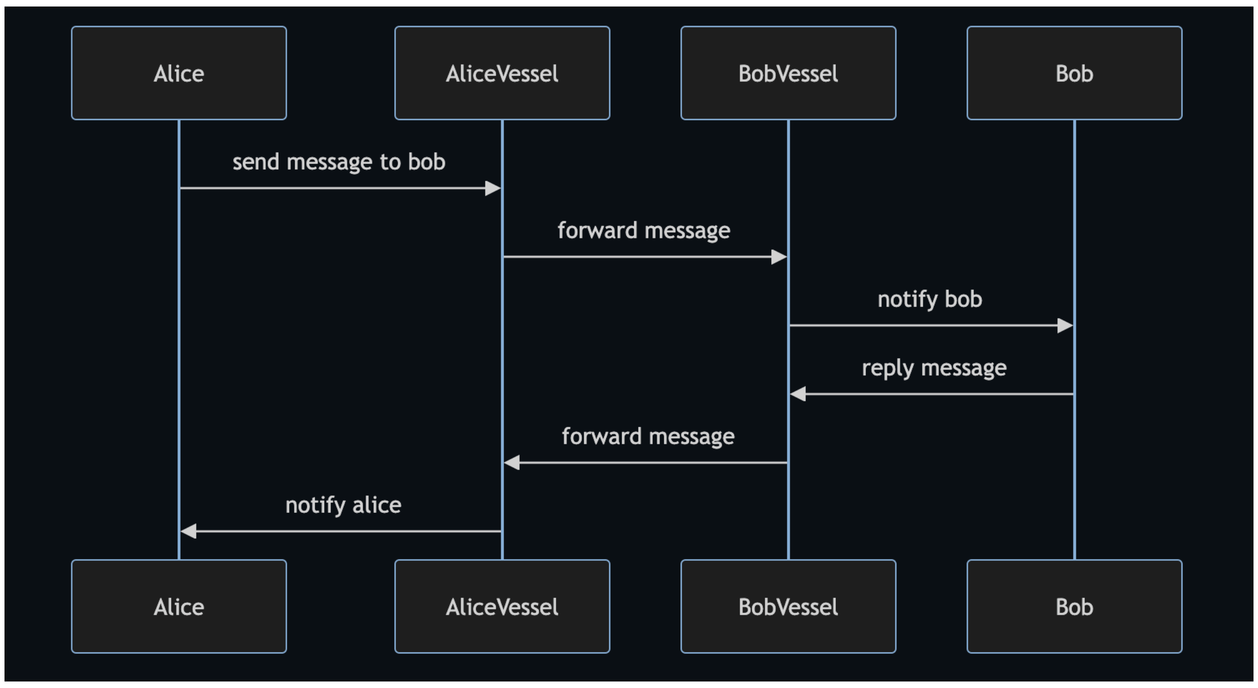
Prople will provide an *ecosystem*, where an *actor* will able to integrate with an organization or institution in a *fair relationship*. If as a person, when we join some *social network*, they (the provider/organization), have the right to use our profiles and activities as the data and sell it to advertisers without notifying us and *sharing* nothing with us. They've got money from our data, is it *fair*?

Totally eliminating organizations or institutions is also not the answer, the right answer is to fix the connection between the organization or institution with the *actor*. The *ecosystem* in the *Prople*, is designed to have this *good*, *healthy,* and *fair* connection between both parties. An *actor*, like a person, still needs the intermediaries or organization, and vice versa, but the connection and relation between both must be based on *fair relation* and *fair connection*. The *actors* own and manage their *identities*, activities, and assets, and the organization doesn't need to host their actor data in their vaults just for verification, it can be done through cryptography algorithms.

There is a software called *Prople Vessel*, which is a *container* or an *agent* used as a *digital representation* of an *actor*. What *actors* need to do is to deploy this *agent* into their environment, a cloud or bare metal server, and they'll be able to connect to their agents through a *controller* which is a desktop/web client. This software (vessel), is used to connect to other *actors*, it can be a user, an organization, or even a *bot*.

An actor may have multiple vessels, but a single vessel can only be owned by a single actor.

The message flows between users can be visualized in the diagram below:



All the *communication* between *actors* will always go through their *vessels*.

The *Prople Vessel* will have functionalities to manage these three domain problems:

- *DID (Decentralized Identity)*
- *Decentralized Social*
- *Decentralized Finance*

All of these three core domains are designed specifically to help an *actor* to own and control their activities and assets, including their integration with some entities, organizations, or institutions, inside the *Prople Ecosystem*.

Core Domains

This section will give out a high-level overview of all the available standards and protocols used inside the *Prople*.

Identity

It is the core foundation of *Prople*. The other domains will be built on top of this primary domain. There are already many *identity management solutions* out there, including the *OpenID*, which try to solve the *interoperability* problem. The problem that try to solve here is about *ownership* and *self-managed* identity management, which focuses on *personalization identity platform*.

Prople does not bring new solutions or standards to solve these issues, but implements existing standards:

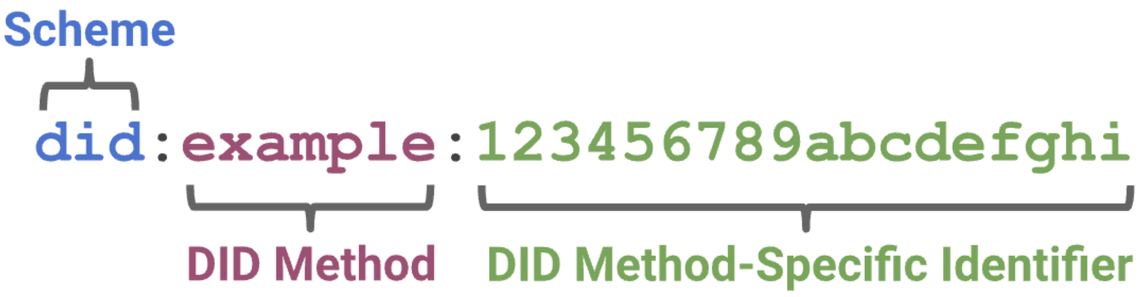
- *DID (Decentralized Identity)*
- *SSI (Self Sovereign Identity)*

DID (Decentralized Identity)

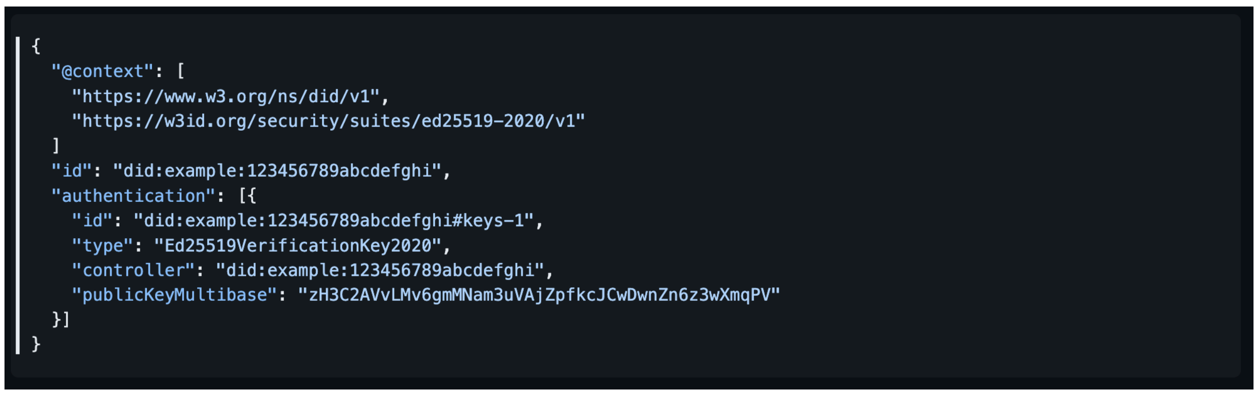
> *Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity*

The most interesting of this concept is this data model designed to be separated from any *centralized registries*. A *DID* syntax is a *URI* that associates a *DID Subject* with a *DID Document*, and the *DID Document* itself will contain cryptographic materials.

DID format:

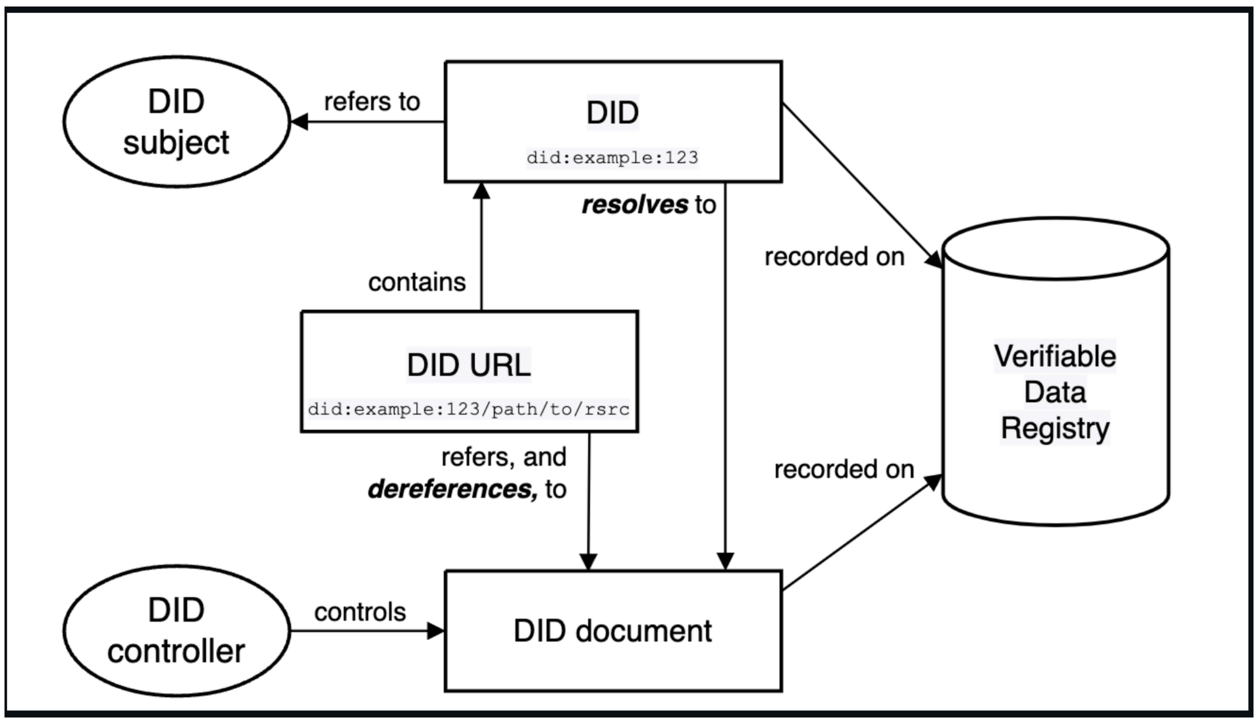

Scheme
did:example:123456789abcdefghi
DID Method
DID Method-Specific Identifier

Example of *DID Document*:



```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }]
}
```

The architecture overview of `DID`:

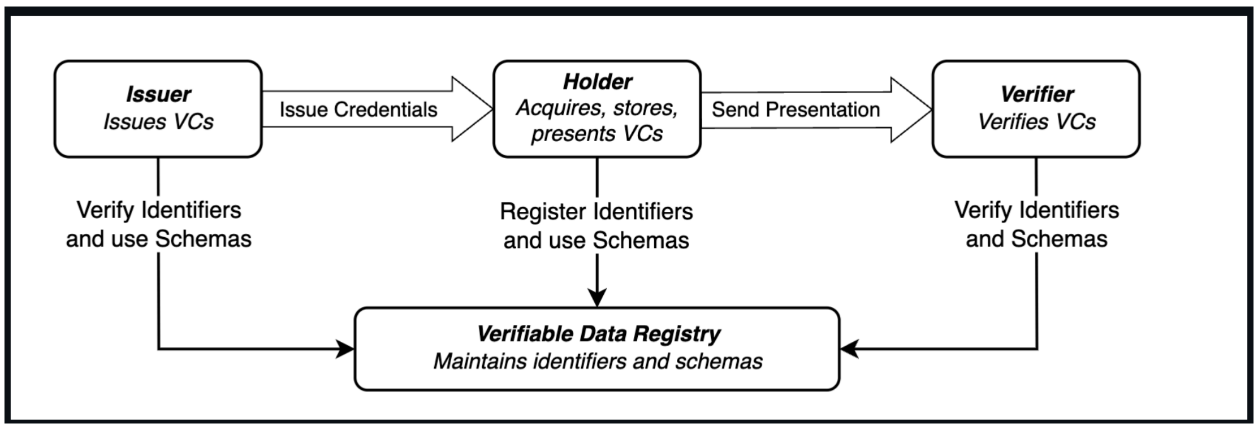


Components:

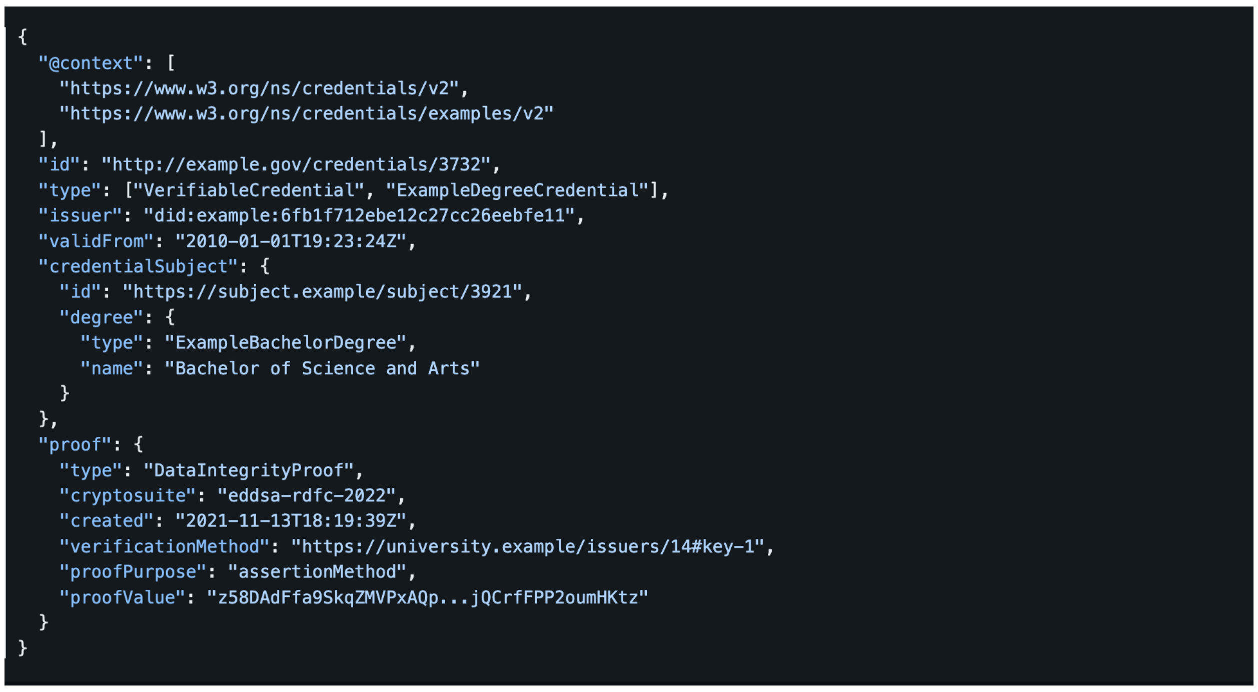
- DID: It's the main identifier which is composed of three parts, the scheme did, the *method identifier*, and a unique identifier
- DID Subject: Every DID will always refer to a DID Subject which is an *entity* or *actor* identified by the DID, and it might also the DID Controller
- DID Controller: It's an *entity* or an *actor* that has capability to modify the DID Document
- DID Document: It contains information associated with the DID. This document will contain any cryptographic public keys to express the *verification method*
- Verifiable Data Registry (VDR): To resolve the DID Document, the DIDs will be recorded in the underlying system or network. Any system or vault used to store the DID Document and the DIDs will be called a VDR, regardless of whatever the technology used

To manage *credentials*, there is a concept called DID Verifiable Credential (DID VC), which is used to express *credentials* that are cryptographically secure, privacy-respecting, and machine-verifiable.[^2] A *holder* of VC is able to generate the Verifiable Presentation (VP) and share this VP with the *verifiers* to prove the *credentials*.

The overview message flows:


Issuer
Issues VCs
Issue Credentials
Holder
Acquires, stores, presents VCs
Send Presentation
Verifier
Verifies VCs
Verify Identifiers and use Schemas
Register Identifiers and use Schemas
Verify Identifiers and Schemas
Verifiable Data Registry
Maintains identifiers and schemas

An example of VC documents with embedded cryptographic proofs:



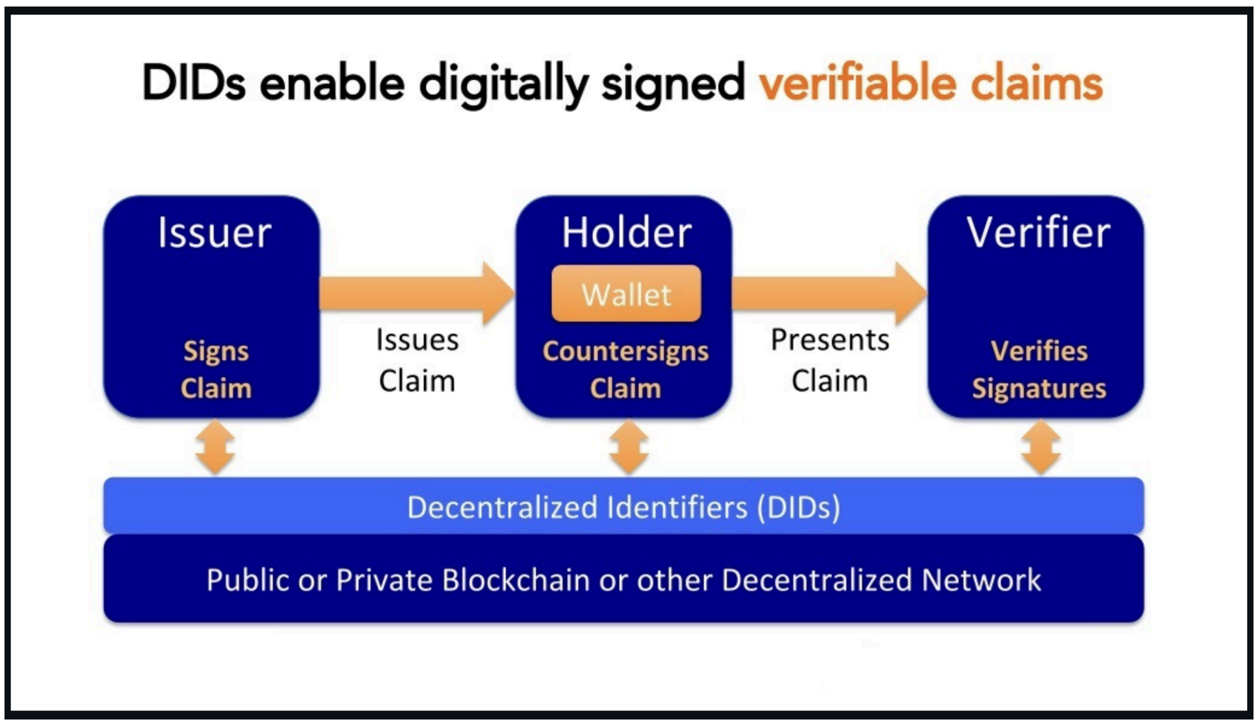
```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.w3.org/ns/credentials/examples/v2"
  ],
  "id": "http://example.gov/credentials/3732",
  "type": ["VerifiableCredential", "ExampleDegreeCredential"],
  "issuer": "did:example:6fb1f712ebe12c27cc26eebfe11",
  "validFrom": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "https://subject.example/subject/3921",
    "degree": {
      "type": "ExampleBachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
  "proof": {
    "type": "DataIntegrityProof",
    "cryptosuite": "eddsa-rdfc-2022",
    "created": "2021-11-13T18:19:39Z",
    "verificationMethod": "https://university.example/issuers/14#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z58DAdFfa9SkqZMVPxAQp...jQCrfFPP2oumHKtz"
  }
}
```

The `verification method` defined above, will refer to the defined keys in the `DID Document`. It means, that to verify this `VC Document`, we need to solve the `DID` first to get the `DID Document` which contains a set of *public keys*. This *public key* is used to verify the digital signature at `proofValue`.

SSI (Self Sovereign Identity)

> Self-sovereign identity (SSI) is an approach to digital identity that gives individuals control over the information they use to prove who they are to websites, services, and applications across the web

The overview architecture enables individuals to own and control their identities:



DIDs enable digitally signed verifiable claims

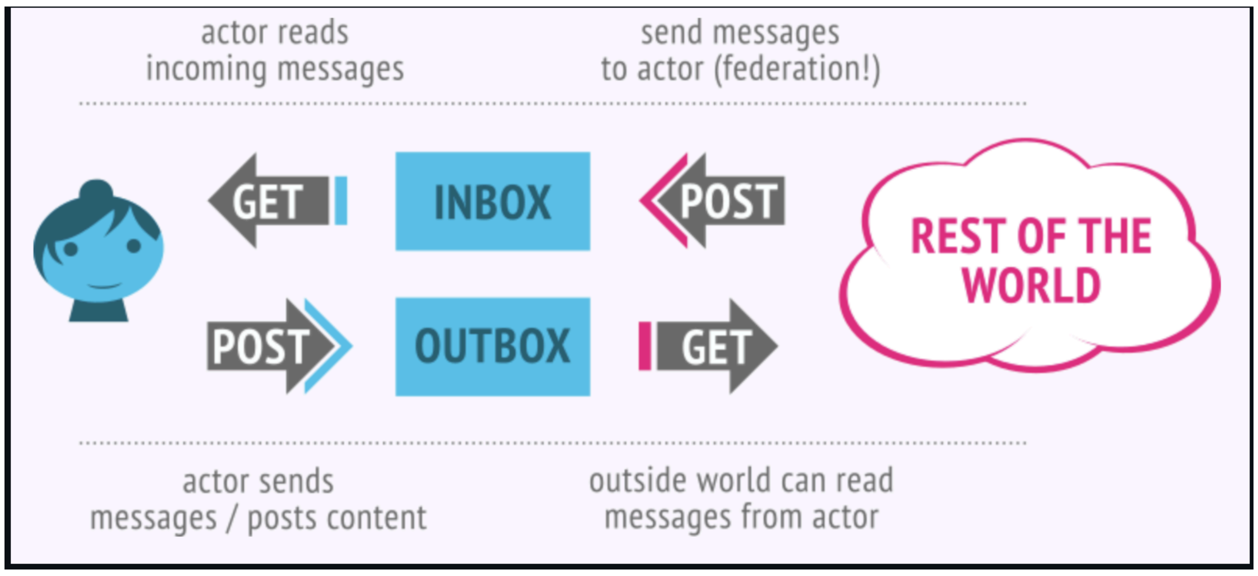
The `SSI` is the higher concept on top of `DID`. The `DID` itself is a low-level implementation of the *sovereign identity*.

Social

ActivityPub

The main objective of *social activities* defined in `Prople` ecosystem is about the *decentralized social network*. Instead of building a new algorithm or new protocols, `Prople` will implement the currently existing standard, called `ActivityPub`.

> Enter ActivityPub! ActivityPub is a decentralized social networking protocol based on the ActivityStreams 2.0 data format



This standard protocol for the *decentralized social network* is aligned with the `Prople` vision that *actors* should have the freedom and rights to control their social activities, and this standard looks like able to reach the objective.

This standard has also already been recommended by the `W3C`:

> The ActivityPub protocol is a decentralized social networking protocol based upon the [ActivityStreams] 2.0 data format. It provides a client-to-server API for creating, updating, and deleting content, as well as a federated server-to-server API for delivering notifications and content

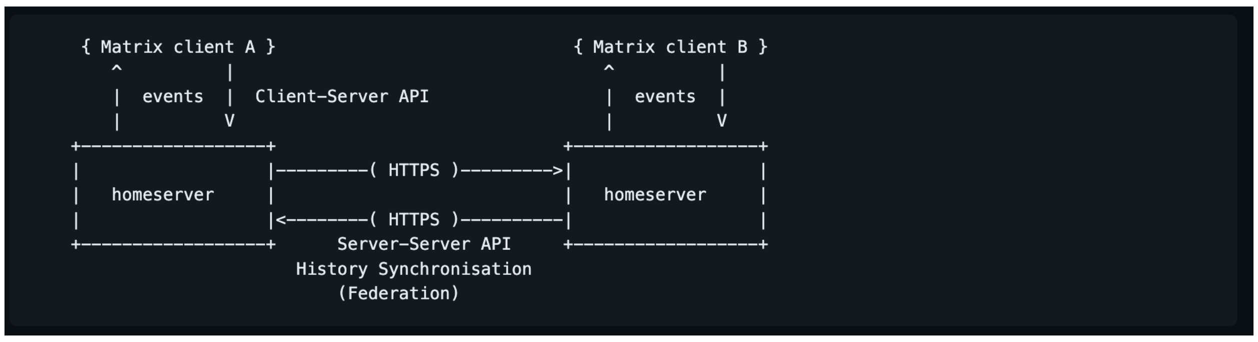
Matrix Protocol

> An open network for secure, decentralized communication

To enable real-time messaging communication, `Prople` also will implement this standard protocol, especially for its `Client-Server API`.

> The client-server API allows clients to send messages, control rooms, and synchronize conversation history. It is designed to support both lightweight clients which store no state and lazy-load data from the server as required - as well as heavyweight clients which maintain a full local persistent copy of server state

Data flows architecture



```
  { Matrix client A }                             { Matrix client B }
      ^          |                                    ^          |
      |  events  |  Client-Server API                 |  events  |
      |          V                                    |          V
  +------------------+                            +------------------+
  |                  |---------( HTTPS )--------->|                  |
  |   homeserver     |                            |   homeserver     |
  |                  |<--------( HTTPS )----------|                  |
  +------------------+      Server-Server API     +------------------+
                         History Synchronisation
                             (Federation)
```

Finance

Public Blockchain Network

`Prople` will be integrated with some of the cryptocurrency networks, and for the base network or as the foundation it will be integrated with the `NEAR Protocol`. Because there are so many networks out there, the main concerns to be taken here are about the *chain abstraction* and *interoperability*, and because of these two things, `NEAR` is chosen, rather than other networks.

The next implementation will probably support other networks, through some *abstraction* projects, such as:

- `Zetachain`
- `Axellar Network`

Although the base foundation network will use `NEAR`, it doesn't mean to limit users' transactions. The main objective of *chain abstraction* is that the user should be able to send their asset to any network out there, *easily*.

For the first phase, *actors* will only be able to send the assets that are already supported or exist at the `NEAR Network`, next phase, they should be able to send or receive assets across the networks.

Sovereign Crypto Wallet

As with common cryptocurrency networks, `Prople Vessel` also will be a *sovereign wallet* which means will maintain the *actor* keypairs containing the *public* and *private* keys, secured with the `SSS (Shamir Secret Sharing)` algorithm. By implementing this algorithm, the *actor's private keys* will be separated into multiple shares, and one of them will be downloaded to the *actor's* local environment, and all of the available shares will be generated encrypted. So even, if the *actor's vessel* is attacked by some hackers, they'll not be able to use the existing keys because the most important share has already been downloaded into different locations.

The definition of this wallet is not like a common web3 wallet that is also used to connect to some *apps*. The first implementation of the *wallet* will designed to manage the *actor* keypairs or as a *key management*, and *transaction management*.

The `Prople Crypto Wallet`, will not act like a common wallet when connecting to the *dapps*. The `Prople Ecosystem` will provide a different way to maintain the connection between the application and the *actor wallet*, which will also take leverage from the `DID` account management above.

Architecture

Principles

- `ADAP (As Decentralized As Possible)`
- `ASAP (As Simple As Possible)`

`Prople` has a mission and vision to provide an open and decentralized network where anyone can participate to join the network. It is different from other private networks which are owned and maintained by a single entity like a company.

Unlike current *blockchain networks* which *sometimes* it will require high resources to run their nodes, `Prople` designed to be *as simple as possible*, because if it's getting easier to run a *node* by anyone, it means we have high confidence that our network will be decentralized enough, to make sure it always *as decentralized as possible*.

Components

Vessel: Personal Self Sovereign Agent & Wallet

`Prople Vessel` is a software which is like a *container* that is deployed in some server's environments, it can be a cloud or bare metal environment, or even deployed in *localhost* and use some tools, maybe like *ngrok* to make it reachable through the Internet.

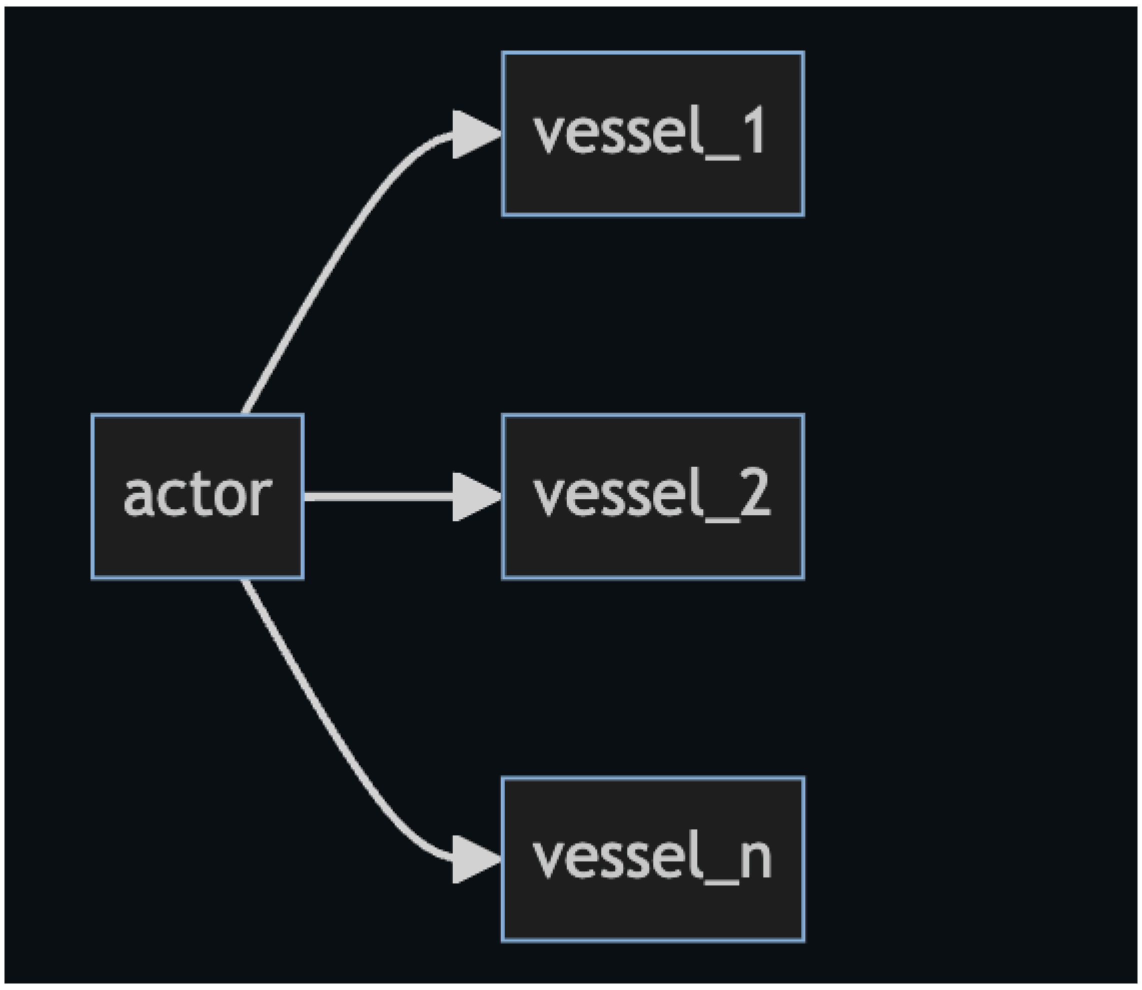
> A `Vessel` should be able to deployed to any environments which able to reached through the Internet. It should not be like a *blockchain node* or *validator* that need a high level computation or expensive resource computation.

The `Vessel Daemon` that need to be deployed and run, will open a single port used to maintain `HTTP JSON-RPC API`. The `API` itself will have a functionalities to maintain:

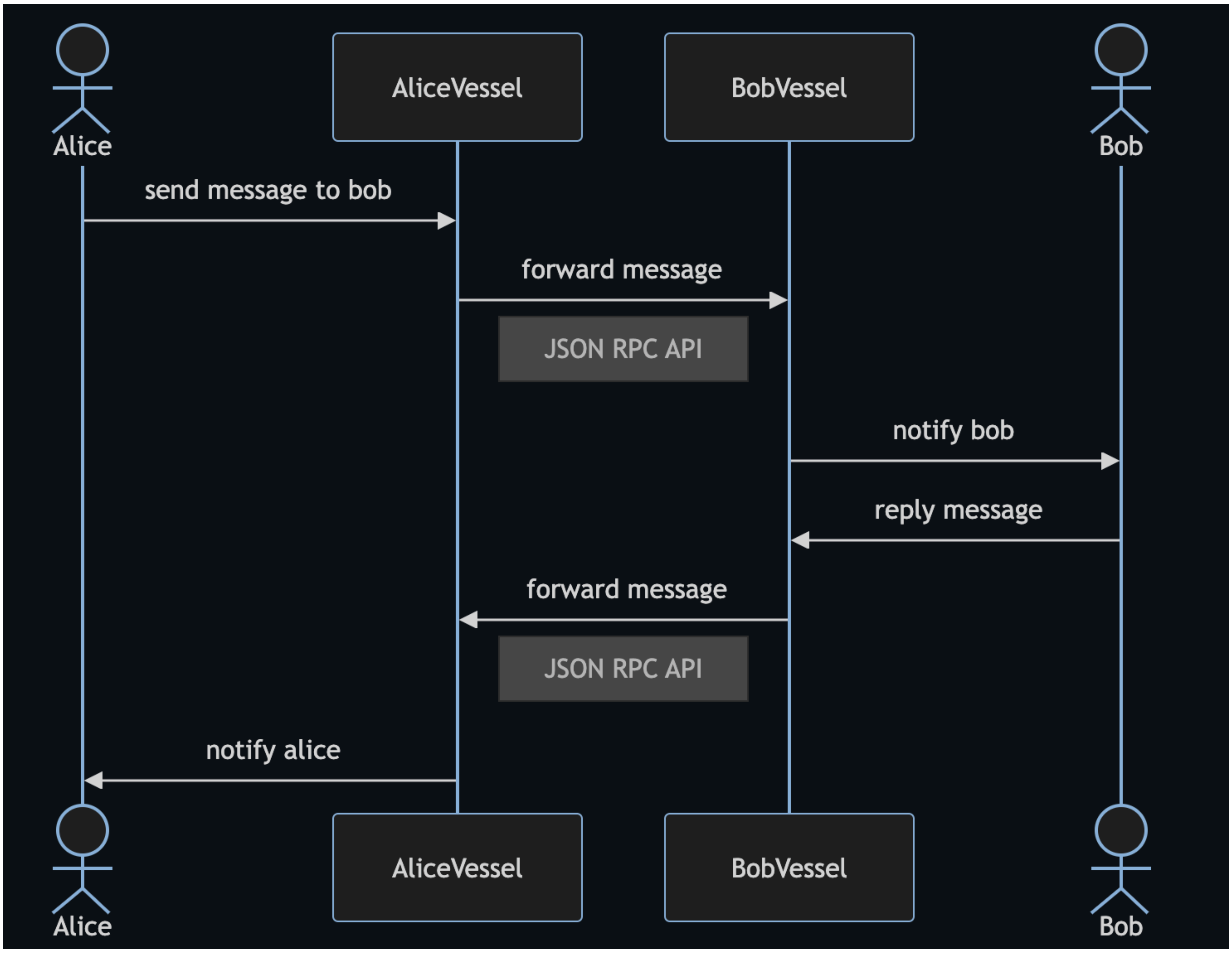
- `Identity`
- `Connection`
- `Social`
- `Finance`

It choose the `JSON-RPC` as main protocol communication becaus its simplicty, no need to maintain multiple endpoints and its handlers, just a single enpdpoint that able to receive multiple commands.

Each of an *actor* will may have multiple *vessels*, but a single *vessel* will only be owned by a single *actor*.



All communication between *actors* will have through their *vessels*, and the communication protocol used between *vessels* is via `JSON-RPC API`.



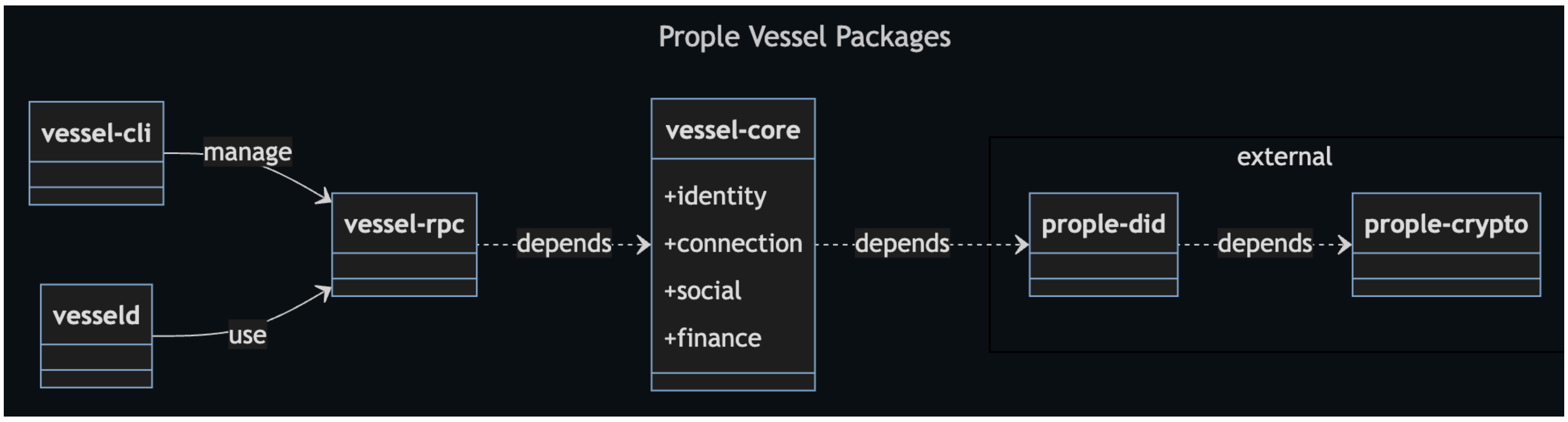
`Prople Vessel` designed to be deployed and run as a single instance. There is no need to make a *replica* of it, because its usages only for a personal activities, including its interaction with other *vessels*. The communication between *vessels* will always happened *one-to-one* not *many-to-one*.

The analogy is, when we are building a *centralized entity*, there will be a lot of incoming connections and requests. This condition give us needs to make our system must be *scalable* enough to handle the incoming requests, this condition called as *many-to-one*.

For the personal activities, it will be very rare to face those kind of condition. The communication between *vessels* should be happened through *one-to-one* connections.

Packages

The `Vessel` codebase separated into multiple `Rust Packages`:



Internal packages:

- `vessel-core`: It's a core domain abstractions which provides all domain logics
- `vessel-rpc` : It's an RPC handlers and also persistent storage
- `vessel-cli` : It's a command-line utility used to communicate with the *actor vessel*
- `vesseld` : It's a *vessel daemon* or binary application that will be deployed and executed in some environments

External packages:

- `prople/did` : It's a package build to modeling the `DID`
- `prople/crypto` : It's a package used to manage cryptography algorithms

Cryptography

There are two important needs for the *cryptography* :

- Key exchanges
- Digital signatures

The cryptography at the `prople/crypto` designed specifically only for `Prople Ecosystem`. There are multiple algorithms used:

- `ECDH`: Used to generate *keypairs* (public and private keys), to establish a *shared secret key*
- `EdDSA`: It's a digital signature scheme, which will be used as the *unique identifier*
- `Blake3`: Used for a *hash function*
- `Chacha20-Poly1305`: It's a `AEAD (Authenticated Encryption with Additional Data)` algorithm
- `Base58`: Used for encoding format

The *key exchanges* will be used to create a secure channel of communication between *actors*.

There are two kind of *connections* :

- Public connection
- Private connection

For an *actor* to be able to connected and share *private messages*, they have to be *connected*. This special connection, means they need to exchange their *ECDH public keys*. These keys will be used to generate the *shared secret key*, which is a combination between private keys and public keys from others. Once this *secret keys* generated, their next communication will be formed in the *encrypted messages*.

This kind of algorithms used through the `DIDComm` standard.

> DIDComm uses DIDs (Decentralized Identifiers) to establish confidential, ongoing connections, without the need for usernames and passwords.
>
> DIDComm protocols enable trusted interactions between parties

The `EdDSA` algorithm used for two cases:

- Message digital signatures
- Identity unique identifiers

The algorithm behind the *unique identifiers*:

eddsa::KeyPair -> get publicKey -> hash: SHA3 -> hash: Blake3 -> multibase: Base58Btc

This algorithm actually follow what `Bitcoin` when generate the *address*, but using different hashing algorithms.

Bitcoin	**Prople**
`SHA256` & `RIPEMD`	`SHA3` & `BLAKE3`

VSP (Vessel Service Provider)

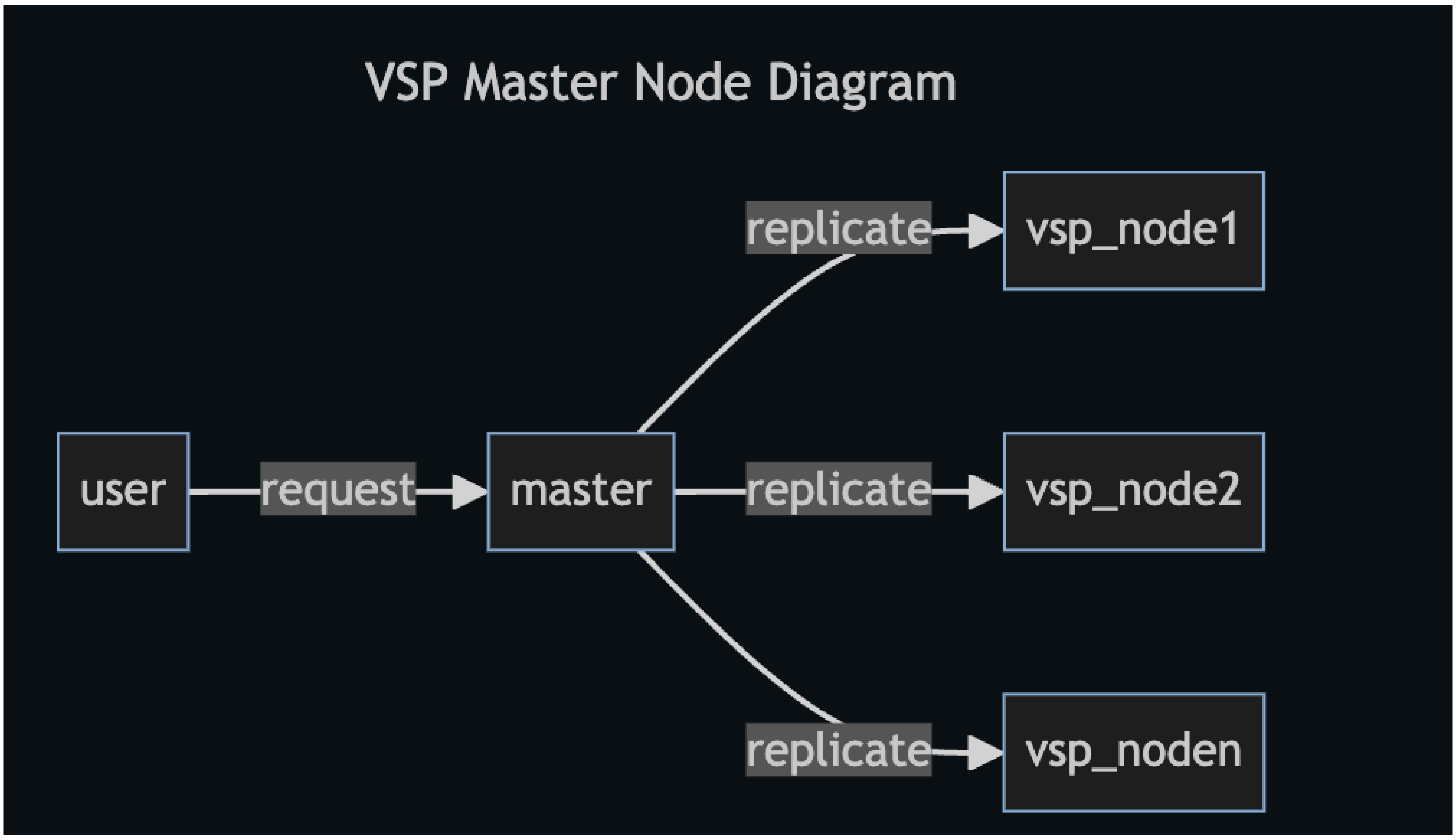
`Prople` designed by a people for the people. The objective is building the personalization platform, and help each others

The `Prople Vessel` is really a good fit to maintain a personal needs. But the disadvantages is, it need a technical skills to setup. An user need to deploy and execute the binary in their chosen environments, it will be an easy task for technical persons, but for the non-technical person it will be really hard.

The `Prople` designed to be an *open and decentralized networks*, which means, anyone can participate to join and build the network. A technical users will able to participate and join the network by become the *service providers*, providing help to other non-technical users, as a `VSP (Vessel Service Provider)`.

The `VSP` will maintain `Prople Vessel` as a *logical views* to handle multi tenants. The difference with the single running daemon of *vessel* is if a single daemon running the engine in some environments, the *VSP* will run as a cluster engine used to maintain multiple *vessel abstractions*. This concept is more like a *database replication system*, where a software engineer setup their database which contain multiple nodes, and each of these nodes connected each others and replicate the data to all of available nodes. All the functionalities will be same like a single *vessel instance*.

For the highlevel overview of `VSP` will be like this:



Each of data will be replicated to other available nodes using the `Raft Procotol` as a base protocol standard to replicate the data including for *master/leader election*. Any user requests will always comes through the *master node*.

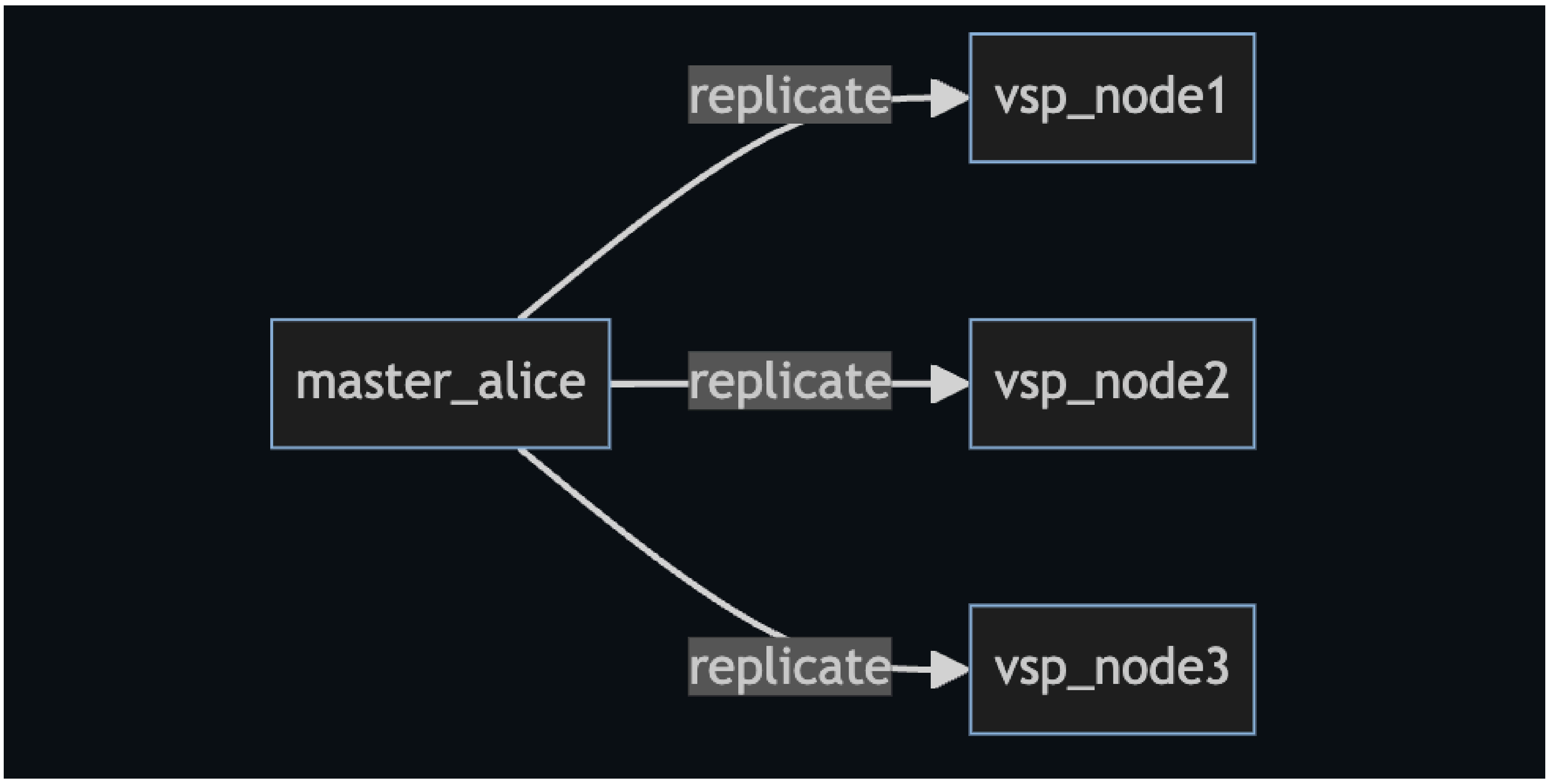
The internal communication between nodes including for its *master/leader* will using *libp2p network stacks* to maintain peer-to-peer network communications

This *libp2p* will also be used for the *master-to-master* communication protocol between *providers*

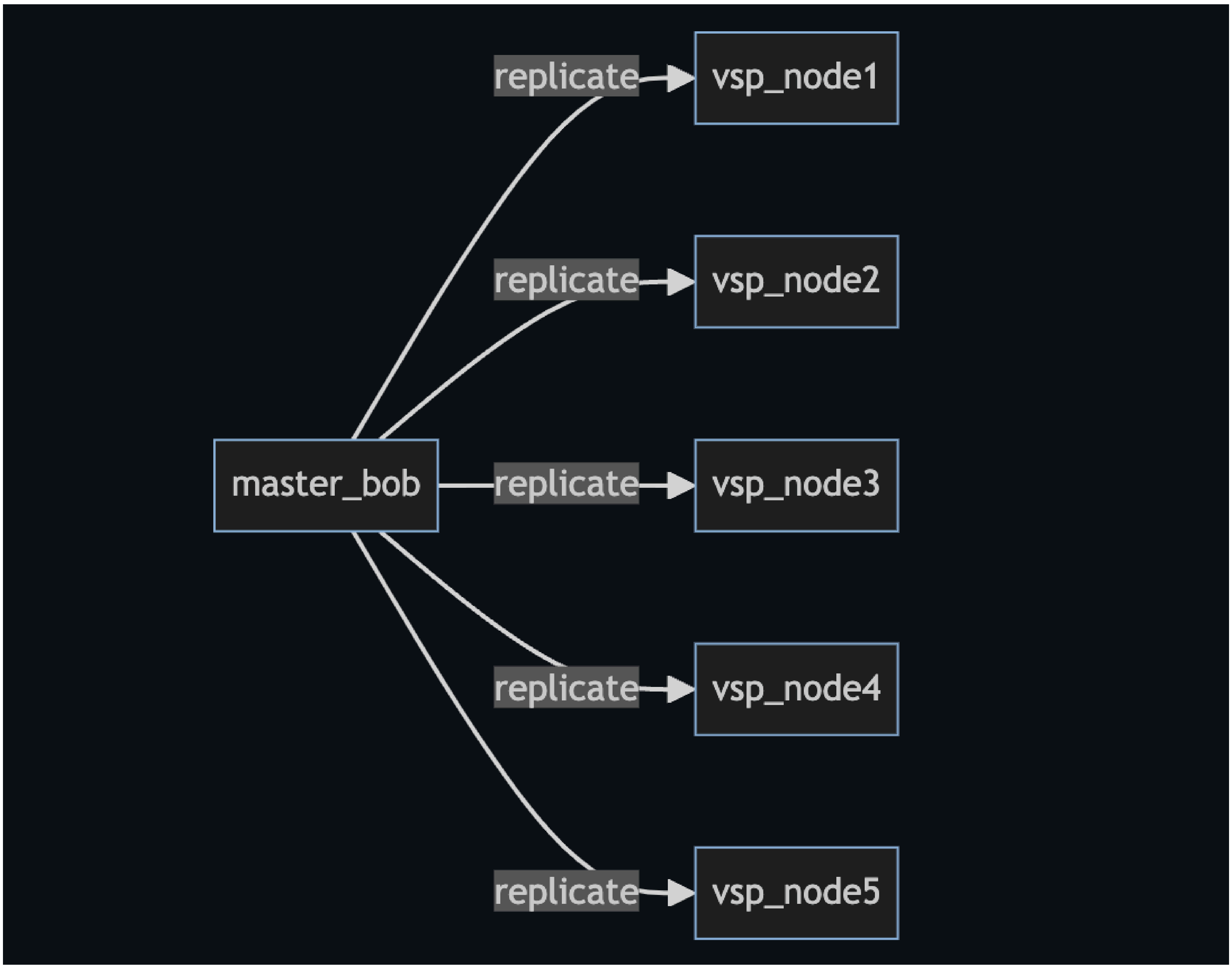
Example use cases

Each of *providers* may be connected to each others. Let's say there are two *service providers*, `Alice` and `Bob`. Both of them as a *provider*, already setup their *vsp networks*.

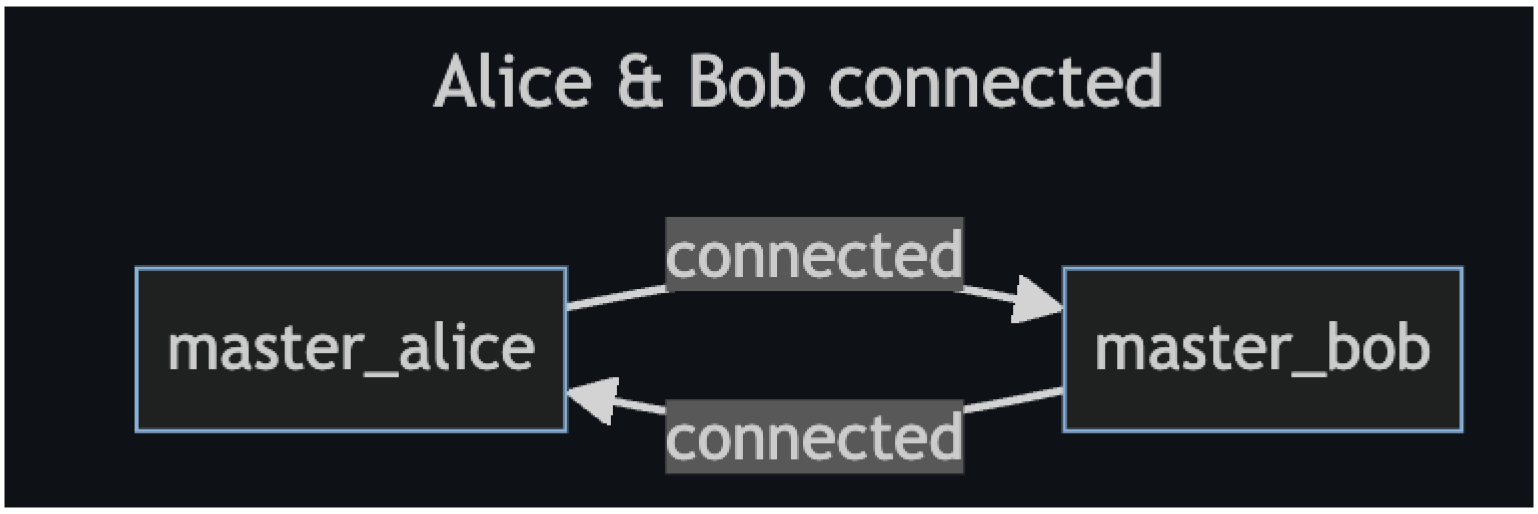
Alice networks:



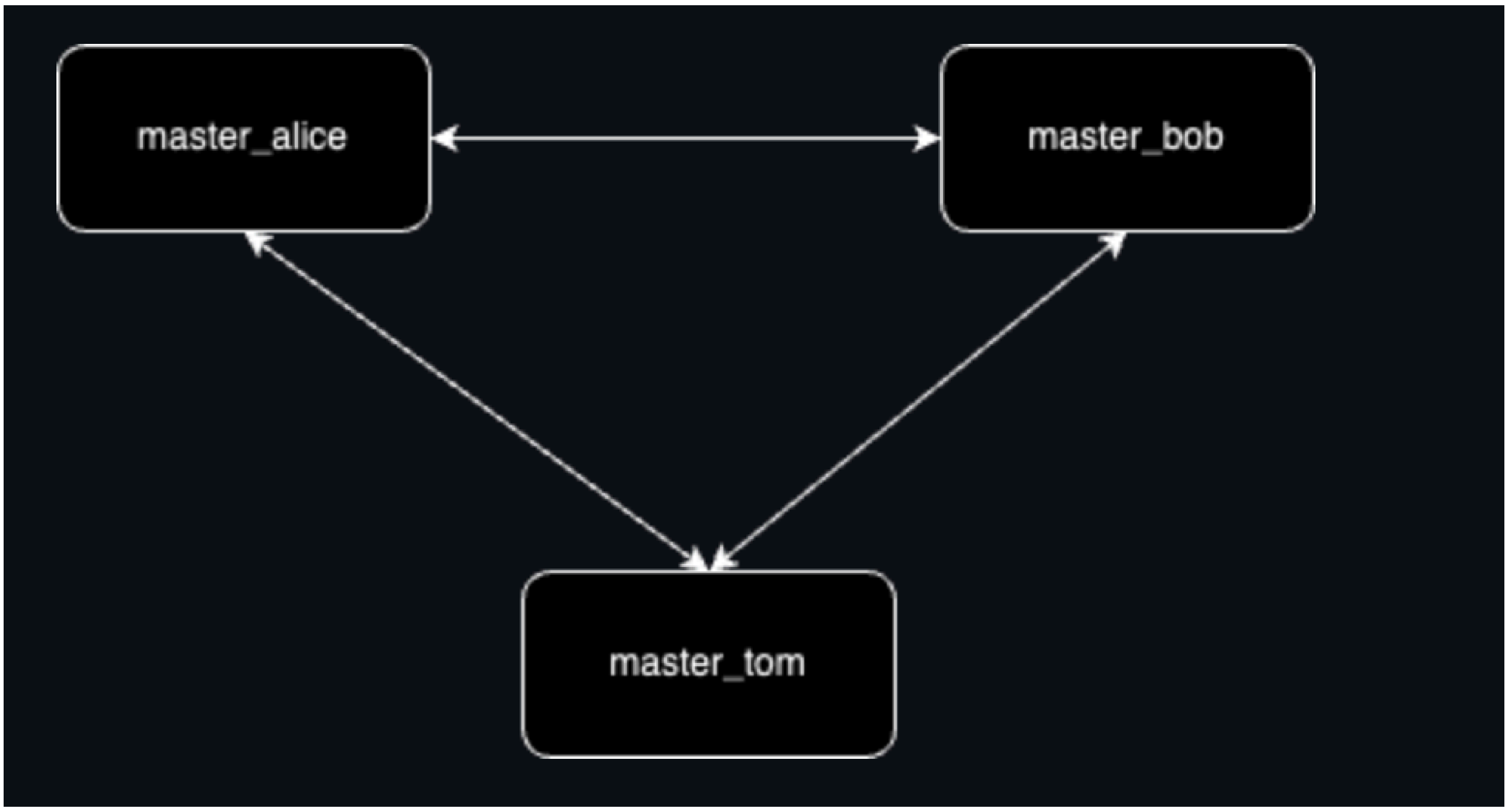
Bob networks:



They may have to connected each others:



If there are third party involved, let's say, `Tom`, the network diagram will be like this:



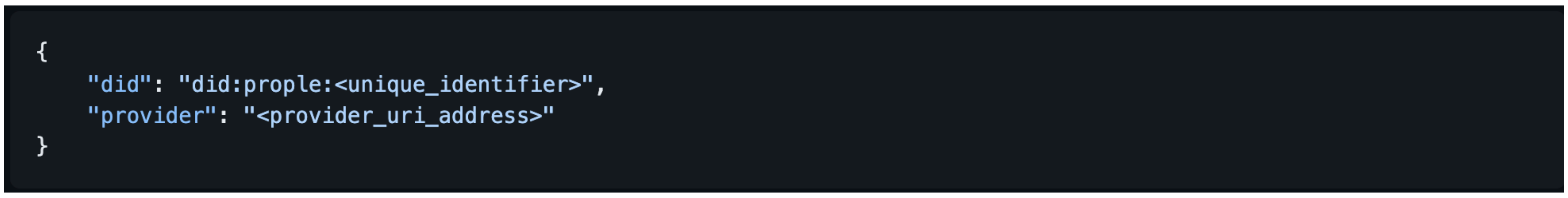
The communication protocol between *master nodes* will be formed through the `JSON-RPC API`. It will have additional API methods compared with the `Prople Vessel` single instance.

Unlike *blockchain node operator*, the primary data submitted by user will be saved only in a single *provider*. For an example, an *UserA* submit their identities through the `Alice Provider`. The *UserA* data will be saved only at the `Alice Provider`, and will only be replicated inside the `Alice VSP Nodes`, the *primary data* will not be shared with other connected *provider master nodes*.

The reason of this approach is to make sure that once an user decide to remove their personal data *forever*, they only need to remove it from a single *node*, other nodes that asking the value will be affected. It's different concept with the *blockchain node* where the data will always saved in the *immutable database*.

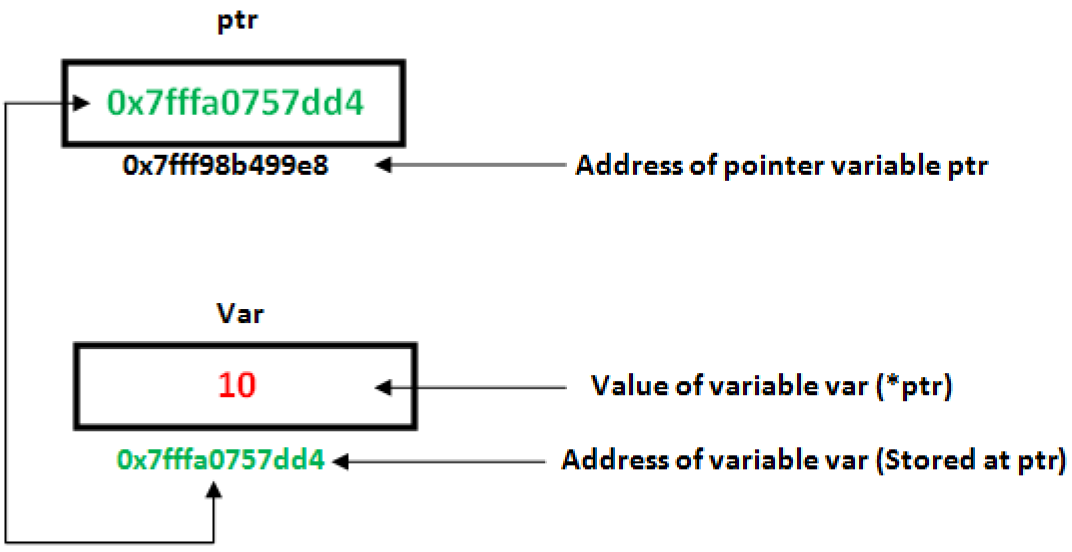
To make sure that an user has a true ownership and control for their data, especially for their identities, they have full rights to modify and delete their own data

The data that will be shared to other connected *master nodes* is just the `DID Account` with the *provider uri*.



```
{
    "did": "did:prople:<unique_identifier>",
    "provider": "<provider_uri_address>"
}
```

It following the concept of *pointer references* in the software engineering.

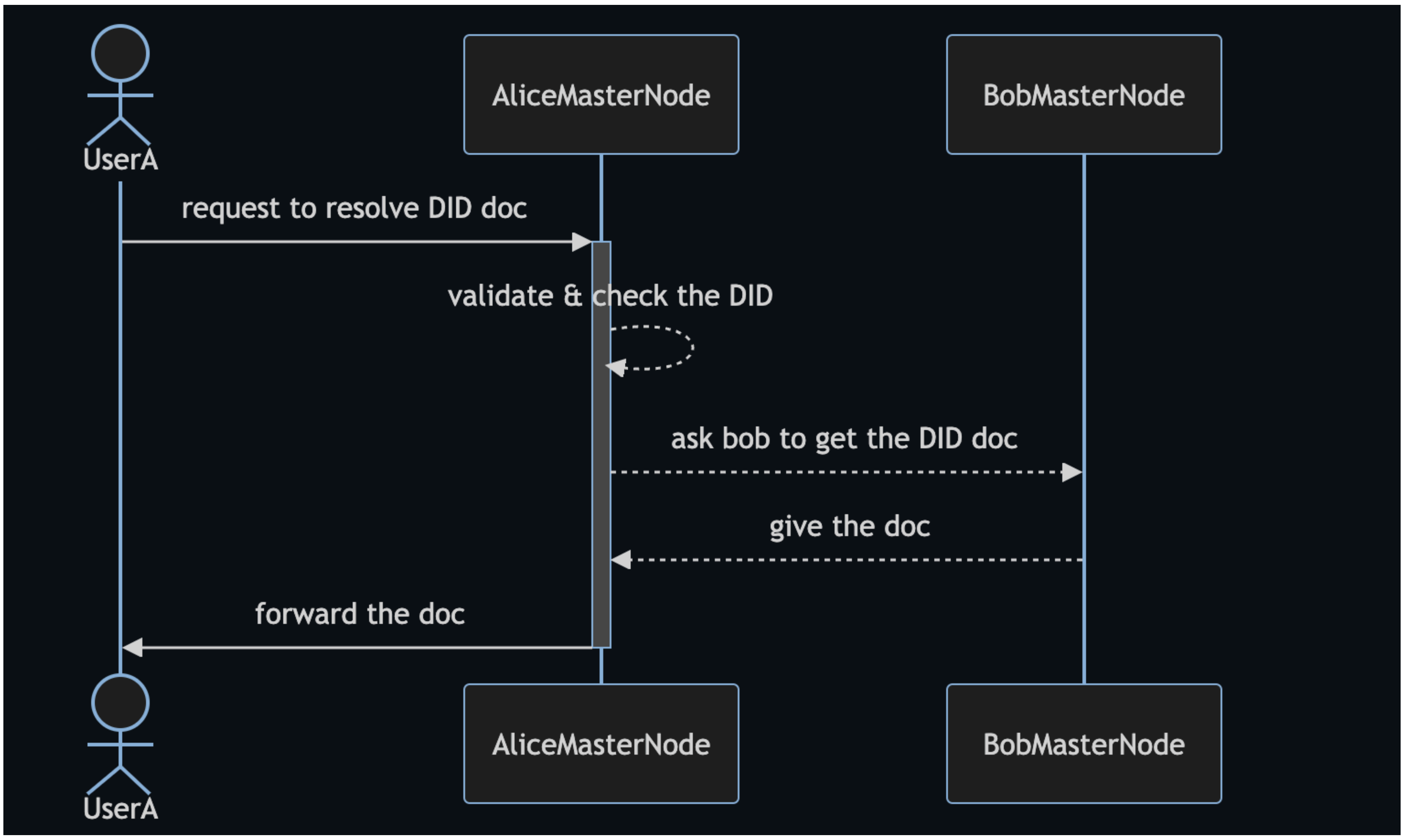


ptr
0x7fffa0757dd4
0x7fff98b499e8
Address of pointer variable ptr
Var
10
Value of variable var (*ptr)
0x7fffa0757dd4
Address of variable var (Stored at ptr)

Source: https://www.geeksforgeeks.org/cpp-pointers/

Any users will able to connected to any available *providers* and able to *resolve DID document* to any nodes. But if the requested DID is not belongs to them, each of node need to make a *call* to the *DID provider* to get the full *DID Doc*.

Example:



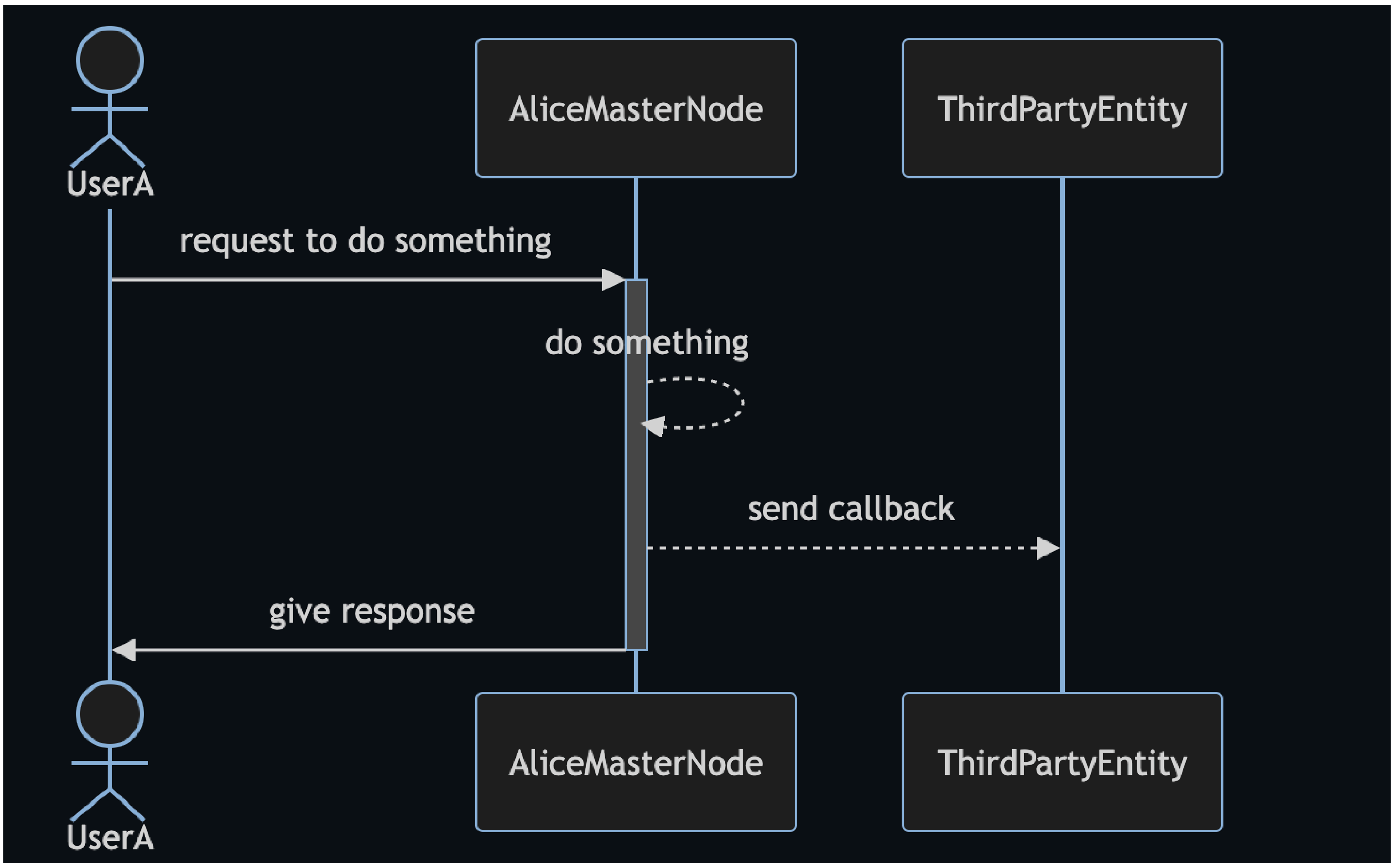
If the requested `DID Account` already belongs to `Alice` then it doesn't need to make any calls to other provider's nodes.

> The `Prople` decentralized network designed to provides *decentralized computation resources* for all users. It's not just used to manage their identities, but also other domains such as for the *social* and *finance*

Vessel Webhooks

Whatever user's choices, either it's *personal vessel single instance* or through the *VSP*, each of user's vessel must have a *Webhook API*.

Each of internal activities inside the *Vessel* will have an *event*, and when an *event* triggered it will send an request to *registered endpoint as callback*. All of the callback endpoints must be registered first to got the callback request calls.



If some of you think, "why webhook API? why not websocket?", the answer is back to our first principles, which is, *ASAP (As Simple As Possible)*, using Webhook is much simpler than managing and maintaining the WebSocket.

Ecosystem

For now, the `Prople Ecosystem` will contains:

- `Prople Platform`
- `Prople Apps`
- `Prople Community`
- `Prople OSS Center`

Prople Platform

`Prople Platform` provides two core components:

- `Prople Vessel`
- `Prople VSP Network`

`Prople Vessel` is a *single instance* of user's *vessel agent*. An user must setup their own *vessel agent* in some environments that able to reaches through the Internet. It can be a cloud environment, a bare metal server, or even in their *localhost* and connected through *ngrok*.

`Prople VSP Network` is an open, decentralized and P2P network. Anyone can join and participate to this network. An user also able to use this network, if they won't to setup their own *vessel agent*. This network provides same features of `Prople Vessel`, the difference is, user doesn't need to setup their own *vessel agent*, what they need are:

- Connect their *vessel controllers* or *vessel client* to this network
- Reserve their own *vessel agent*

There are will be a fee to use this network to reserve the *vessel agent*. This fee will be paid to the *vessel providers*

Prople Apps

The `Prople Apps` is a place where developers and engineers contribute by building the application on top of the `Prople Platform`.

Developers and engineers are free to promote their application at the *community hub*. They're also allowed to gain profitability from their consumers.

There are no limitations of technology stacks to start building an application on top of our platform. They should be free to use any technology stacks they're like. There are no limitations of the technology, it means there are many possibilities of the applications:

- The desktop app
- The mobile app
- The custom network

The differences between the application built on top of the `Prople Platform` with the *common* Web2 applications are:

- It must not force their users to create an identity that will be *sealead* inside their applications
- The connection between the application and their users must be through user's `Prople Vessel Agent`, either user use a single instance or through the `VSP Networks`

The experience building the application on top of `Prople Platrom` is almost like the Web3 experience. If in Web3 has a concept of *wallet*, then there is a new concept of *vessel agent* in the `Prople`. The connection between the application and their users must be through user *vessel agent*, even their application must be run it's own *vessel* to be connected with their users.

Once users have been *connected* with the application, they're start to use the application's features.

Prople OSS (Open Source Software) Center

The `OSS Center` is a *central hub* maintained by the *foundation* which provides:

- Core technology platform
- Tooling or any supported software

All the provided technologies (including software, tools or library) will always be in *open source licenses*. The supported licenses are:

- AGPL v3.0
- GPL v3.0 or later
- Apache License 2.0
- MIT License

For the documentation and collaboration management:

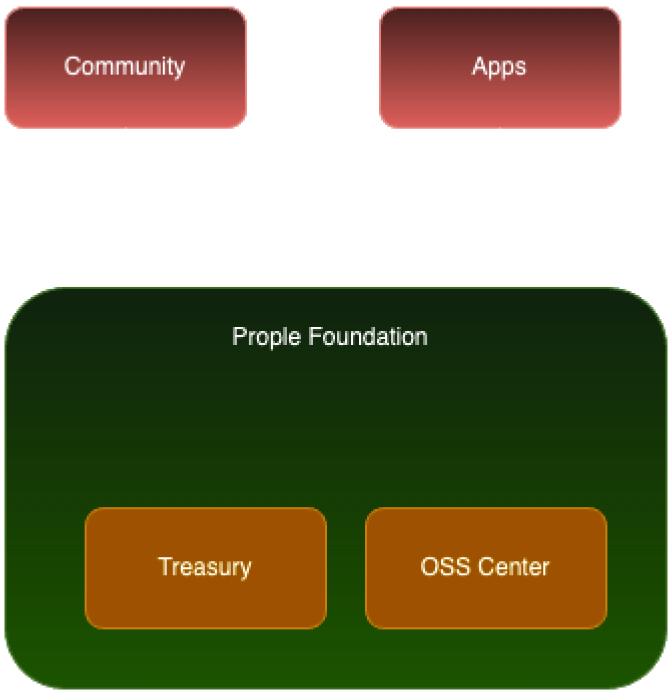
- `Prople Paper`
- `Prople OSS RFC`
- `Prople OSS ADR (Architecture Decision Record)`

The main repository of the `Prople OSS Center`: https://github.com/prople

This repository will be used for all the software, library, sdk, including for its documentation (paper, rfc and adr)

Governance

There is a `Prople Foundation` as the *core foundation*. The high-level diagram visualize the relations:



The `Prople Foundation` will only maintains two core domains:

- `Treasury`
- `OSS (Open Source Software) Center`

The *community* and also the *apps* will be outside of the *foundation*, which means, the foundation should not put too many powers or influence to control them, and also to make sure for the *opennes* and rich growing community and *adoptability* of applications.

The `Prople Foundation` itself will be managed as *non-profit organizations* which provides the `OSS` and also it's `Treasury` to support the ecosystem.

Treasury

This term taken from the `Web3 DAO` concepts. The *treasury* is a place where the *foundation* accept any donations from outside or external entities. The *treasury* can be managed with two ways:

- Fiat currency
- Crypto currency

All the *treasury* deposits including for its expenses will always be reported *publicly*.

The treasury deposits will be used for the core development and the ecosystem needs.

OSS Center

`Prople Foundation` will provides the core technology *without* any supports for now except only for the papers and documentation, because for now there are still a lot of things that need to be done, so focus on the core development is important.

The `OSS Center` for now will only focus on these components:

- `Prople Vessel`
- `Prople VSP`

Including for any libraries, packages, and modules that will support those components.

Since the `Prople Foundation` is not like common startup or tech company, it's a *non profit organization*, anyone can be a *contributor*, but, to be a *maintainer* it will be chosen by the core team, and the one of conditions to be a *maintainer*, they should be a *contributor* first. The active, honest, and effective *contributor* will be recommended to join *maintainer* teams.

As a *maintainer*, they should be rewarded using the currency deposited at the *treasury*, to reward them for their time and resources help this project running.

Maintainer is not a technical person that will doing *coding*, but it may also be a person that willing to help maintains:

- Community
- Documentation
- Social media

Conclusion

In this paper we've introduced the `Prople` and it's concept of `Personalization Platform`. While there is already existed the `Web3` technology, including blockchain, cryptocurrency, user already been introduced with the concept of *wallet* where they can manage their own crypto assets. `Prople` introduce more broader concepts.

`Prople` as *personalization platform* introduce new experience between `Web2` and `Web3` combined. This platform will push the *personalization* more than current *crypto wallet* can do. It's not eliminating the `Web2` or `Web3` experiences, it try to enrich both of user's experiences.

`Prople` not try to introduce a new standard or protocols, it designed and build on top of existing standard and protocols:

- `Self Sovereign Identity`
- `DID (Decentralized Identity)` by `W3C`
- `ActivityPub` and `Matrix Protocol`
- Existing *crytocurrency* networks, either it EVM (L1/L2) or non-EVM networks, and it has a chance someday will be integrated with the `Bitcoin` ecosystem through its L2 networks, such as `Stacks`

`Prople` provides the software, tools and decentralized networks through its `VSP P2P Networks` to help users maintain their *personalization* activities and assets through their *vessel*. Both of these components will introduce a new experiences to users to use the Internet without sacrifice their *ownership*, *control* and *privacy*.