Booking apartments using crosschain payment methods

1st Roman Páleník
dept. name of organization (of Aff.)
name of organization (of Aff.)
Bratislava, Slovakia
xpalenikr1@gmail.com

[1] ***Abstract*—Blockchains are closed networks, and they can not communicate with each other. This fact makes blockchain a technology limited by the possibilities of the current using chain and impossible to share assets with other networks. If we wanted to use another chain, we had to sell our assets and buy them on another chain, which was not only not practical, but also made blockchain less independent. That is why it is now a discussed topic of cross-chain sharing assets. Cross-chain solutions bring possibilities to share assets between chains and have access to all possibilities of all chains with one account and one set of assets. Although some cross-chain solutions or networks focus specifically on cross-chain sharing like Polkadot, the use in real life in the form of applications is practically insufficient.**

We present Crossbnb, a web3 application run on Polkadot Moonbeam chain, a parachain in the Kusama/Polkadot ecosystem, with the possibility of cross-chain sharing assets and payment methods and as it seems that it works like classic Airbnb in terms of renting an apartment. We represent apartments as NFT, and we can store them on every EVM-based chain. Using the NFT bridge, we can change the network on which is NFT stored. Payments methods are also cross-chain, and we can pay by any ERC20 tokens we have on any Ethereum based chain.

***Index Terms*—NFT, crosschain, sharing assets, bridge, dapp**

I. Introduction

In 2009 Satoshi Nakamoto published the Bitcoin white paper and introduced a new way of electronic payment. This technology was unique, with no need of having a centralized financial institution.

Over the years, many blockchains have been created: Bitcoin, Litecoin, Ethereum, and Polkadot. We have created smart contracts, fungible tokens, non-fungible tokens, and decentralized applications. Blockchain has started becoming capable of processing a lot more than transactions and attracted the attention of companies and developers as an alternative to centralized applications. [1] Nowadays, we talk about lending money, being able to prove ownership of an asset in real life, or creating a DAO. That is a decentralized alternative to existing real-life companies, but instead of having board members, Decentralized Autonomous organizations are run by a government composed of community members holding tokens.

On the other hand, blockchain networks are fragmented and isolated. We cannot share information peer-to-peer between chains. This isolation makes blockchain more challenging to use, and some use cases are not even possible to implement without third parties, which deprives decentralization of blockchain. [2]

Therefore developers have started to work on cross-chain asset solutions. By using cross-chain technologies like bridges, we can send coins, tokens, or NFT from one chain to another. We have explicitly created blockchains to support cross-chain, like Polkadot or Cosmos. [4] These networks has own protocol of cross-chain sharing and also support cross-chain with other networks via bridges. Although several cross-chain solutions have emerged in recent years, this approach is not widely used and implemented in decentralized applications.

We see an opportunity with cross-chain sharing in the sphere of decentralized apps. The application can be on one chain, using every advantage of the network such as low fee or networks protocol like Polkadot's cross-chain protocol. However, users can interact with it and use assets from another chain without manually retrieving assets from their network through auction. Therefore we decided to create a cross-chain version of Airbnb built on cross-chain sharing assets such as ERC20 tokens and ERC721 tokens by using bridges.

The following section will discuss some existing cross-chain sharing asset solutions and decentralized forms of Airbnb. Section III describes the design architecture of our decentralized app using bridges and NFTs. Section VI sums up the article with results.

II. Related work

In this section, we analyze existing solutions in cross-chain sharing assets. The second subsection discusses the decentralized form of Airbnb and looks at its functioning.

A. Xclaim

Xclaim is a cross-blockchain protocol for transferring an asset from one chain to another. It is trustless, and its primary functioning is based on cryptocurrency-backed assets [3]. The atomic swap protocol inspires it, but it tries to overcome it and repair some of the atomic swap's problems. For example, the authors point out that all parties to change must be online during the atomic swap process. Xclaim achieves this using smart contracts and chain relays. This method can be used for multiple applications like cross-chain payment, N-way atomic swap, or temporary transaction offloading.

[1]My supervisor: Ing. Viktor Valaštín, PhD student at FIIT STU

1) Working: Sending assets from one chain to other authors refers to the Central claim. This claim offers four protocols on which it is easy to demonstrate how Xclaim works. The first one is called Issue. It consists of 3 steps. The first requester checks if the Smart contract on issuing blockchain is available and checks the vault. The requester then locks the asset with the vault and specifies when the asset is sent on the issuing blockchain. When the vault validates this step, announce that the smart contract and intelligent contract send the correct amount of assets to the wallet on another chain.

Another protocol is transfer. The sender notifies the smart contract, which changes the asset owner. The vault witnessed this transfer and changed owner on the backed asset he holds. Of course, only that amount of asset with was transferred. If someone changes an asset to an issuing chain asset rather than changing half of it, he can take back only half of the asset on the requested chain. The swap protocol is very similar, but the two sides swap assets instead of sending them from one place to another. Also, the vault must witness the operation.

The last protocol is redeeming. It is the process of burning issuing chain assets and getting the back-backed asset. We lock issuing chain assets with smart contracts, then send a request to the vault. Vault witness our locking and release-backed asset. Again, how many assets do we lock on issuing chain? That many assets will be redeemed from the vault. As the final step, the vault confirmed that the asset was redeemed, and the smart contract burned the locked asset.

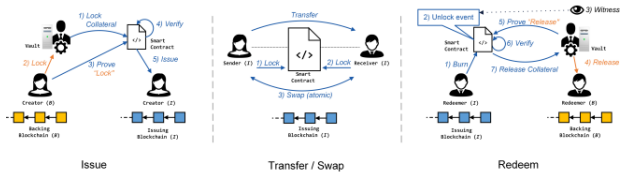

Fig. 1. High-level overview of the Issue, Swap and Redeem protocols in XCLAIM's (under successful execution) [3]

The central claim is functioning protocol but does not meet the requirements for Xclaim such as e Consistency, Redeemability, and Liveness. [3]. The vault in Centralclaim monitors the chain for new crated-backed assets and notifies the smart contract. However, it can steal the backed asset and violate Consistency if it fails. Alternatively, it must redeem the amount of backed asset to redeem less, and the other asset is stolen. It is against Redeemability. Also, the vault must always be online, which is against Liveness.

2) Chain relays: Chain relays are mechanisms how to achieve o provide data from the backing blockchain B to the iSC on the issuing blockchain I. [3] It is a component of the smart contract, and it can inform of backed asset state. Xclaim uses a relay chain to upgrade CentralClaim.

Chain relay as another proving mechanism for solving Consistency—requester proving to chainRelay that funds were sent to vault rightly. The smart contract has proof and information on how many assets were backed. Therefore when the vault sends how many asset-backed, it has to be the same amount. Similarly, it solves redeem problem. Vault, after redemption, has to prove that assets were given to the redeemer. To the chainReley is sending information about this transaction. If the smart contract does not receive proof from chainRelay, the vault is punished.

Instead of trusting a vault, a smart contract has proof from the user using chainRelay. [3] Because transfer and swap protocols need only a smart contract to work, Xclaim achieves Liveness.

Next, we modify CentralClaim by bringing collateral to it. Vault needs to lock collateral. Therefore in case of error or trying to make fraud in Redeem, the smart contract can compensate the redeemer and pay a punishment fee. Naturally, the amount of collateral must be higher than the number of backed assets.

B. Snowbrige

Snow bridge is trustless Snowbridge is a trustless system for bridging arbitrary data between Polkadot and Ethereum. [7] Snowbridge has layered architecture consists from 3 layers: bridge functionality(low level), trust functionality(mid-level) and application functionality(high level). The mechanism consists of some general concepts. A bridge is using a remote procedure call or RPC. The messages they send between chains consist of data to identify the source and recipient and payload for the recipient application. A channel is a means by which users send messages to each other. The channel is made up of sender and recipient, and no one can send a message to the other without submitting it to the channel. A bridge is two channels in opposite directions.

1) Core components: The core components of the bridge are Ethereum-to-Polkadot Direction and reverse. Ethereum smart contract can create a message that it wants to send to Polkadot by creating Polkadot RPC. RPC message is represented as a data structure. It is a one-way call and does not require a reply. For processing incoming messages are Polkadot Parachain Pallets. The payload can contain anything that is understood by receiving pallet. For accepting and verifying Ethereum, messages are responsible palettes called Ethereum Light client verifier. It has a mechanism to detect if the transaction was valid and if it emitted an event.

From another direction, we also have Ethereum RPC, a data structure that creates a Polkadot pallet to call Ethereum smart contract. The payload, in this case, must contain ABI for the receiver contract. There is also a Polkadot light client verifier in the form of a smart contract on Ethereum.

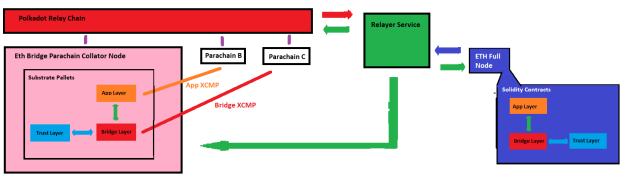

Fig. 2. High-level overview of snowbridge architecture [7]

C. ChainBridge

ChainBridge is a modular multi-directional blockchain bridge built by the company ChainSafe. [8] The goal is to connect the platform Moonbeam to Ethereum, but it has only been implemented on test nets Moonbase alpha and Kovan/Rinkeby. It is a message-passing protocol with three main roles. The listener constructs the message on base from what he gets from the source chain. The router creates a link between listener and writer interprets the message to the destination chain. The logic of the bridge is written in smart contracts on both sides. The bridge contract starts the whole cross-chain process by calling the handler contract. Handler contract validates props from bridge contracts. The last type of contract is the target contract, which interacts with each side of the bridge.

1) General Workflow: The whole process starts with the call deposit function on the source chain. We must set up target chain, resource ID, and call data, which are .... and send to handler contract to validate. After validation, the source contract emits a deposit event function to relayers, and this is called a proposal. Relayers pick up the event and begin voting on the proposal. Relayers vote on the proposal. If they do, is the event emitted by the bridge contract on-chain B. Suppose the threshold is met, relayers do not vote for the proposal but execute via bridge contract. The bridge contract calls the handler contract for control data and executes the proposal.

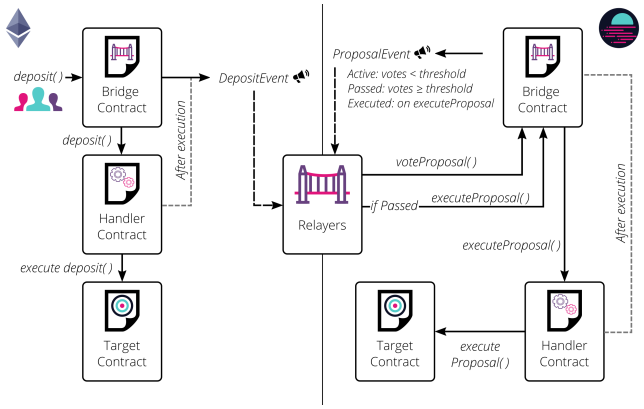

Fig. 3. Diagram summarizing general workflow of Chainbridge [8]

D. Cross-Consensus Message Format

Cross-Consensus Messaging format, XCM for short, is messaging format released on Polkadot for cross-chain interoperability. Polkadot has three systems that implement XCM. UMP, DMP, and XCMP. UMP(Upward Message Passing) serves as a message from parachain to its relay chain. DMP(Downward Message Passing) works opposite UMP. The relay chain sends a message to parachain using UMP, and XCMP allows parachains to communicate with each other. XCM is not the only provider of this type of communication but also guarantees for the future. More precisely, it is a guarantee that parachains will be able to communicate with each other even if their specifications change. The goal of XCM is to be future-proof, efficient, and forward-compatible.

1) Core properties: CM's core is XCVM - Cross-Consensus Virtual Machine. It is an ultra-high level non-Turing-complete computer. It is register-based and works with instructions and datatypes.

Locations in XCM are defined by MultiLocation data type. In the basics is the singleton global data structure. Location is in relative format rather than using root because of changing or merging two blockchains. In this case, the root would be changed and the location invalid. For fees, has XCM special register called Holding register. In this register is sent assets, part of the asset will be used as a fee, and other will be sent to the correct location. Users also have to buy computing time for XCM by fees.

For holding a broad spectrum of assets is data type Multi-Asset. MultiAsset secures total spectrum assets with data types such as MultiAssets, WildMultiAsset, and MultiAssetFilter. MultiAsset datatype is a structure with two variables in it AssetId, Fungibility, which identifies the asset. In XCM is AssetId, either concrete or abstract. The abstract is the name of the asset(which is not much in use because of not knowing all names of case insensitive), and concrete is type multilocation, which is a specific address to an asset like it has been mentioned before. In the case of fungibility is, the amount of token, or in the case of a non-fungible token, is the only name of the token class.

The essence of cross-blockchain is not fees or play with locations but sending assets between chains. In Polkadot, there are two ways. The first one is for chains that trust each other, and it is called teleporting. It is a method when the asset is burned on one chain and mint in another. We are talking about trust because the receiving chain receives the message and the minting of the coin. Therefore, they must trust the sending chain that the coin was successfully destroyed and only then sent the XCM.

The other method is called reserves, and it is more complicated than teleporting. We use this method in case the chains do not trust each other. The third side is called reserves. If we are want to send assets from one chain to another, we must send them to the third chain, and that chain sends them to the second chain. Naturally, three sides cost more fees and fuel than two sides.

E. DTravel

DTravel si a decentralized blockchain-based competitor to Airbnb created by blockchain company Binance. The application's primary goal is to bring platform-oriented rending rooms. A second goal, but also important, is DAO. [5] The chain has its token TRVL, which is also a governance token. Token holders are part of the decisions and routing of the application, and they decide its internal rules. It is hosted by a travel company named Travala, a centralized booking system with a cryptocurrency payment option.

We can create a rent offer on this website and rent someone's house like classic Airbnb. Application is working on

top of Ethereum, and smart contracts manage its logic. As we mentioned before, Dtravel has its TRVL tokens. TRVL is an ERC20 token and has been deployed on Ethereum. We can earn tokens by hosting a home or buying them as any other token. Dtravel also connects with the Binance blockchain via AnySwap, and we can transfer these tokens between chains. They also provide a premium account in the form of NFT. [5]

Although they are crypto-oriented and provide some level of integration with any swap, the application is more focused on the DAO aspect and has no cross-chain payment methods.

F. Bitbook

Bitbook is a booking platform that offers booking hotels, cars, or flights. They say that today's system has high fees for renting a room, which unnecessarily increases the rental price because hotels and rental services depend on working with corporations with high rental fees. Therefore they provide their booking service. Their main point of using Bitbook is a reward system in Bitbook tokens. We can earn tokens by booking some service, inviting others, or creating content on their website.

Bitbook accepts Fiat currencies as well as assets on crypto wallets. They currently work on a payment method that includes Bitbook tokens. By rewarding users for using their services, Bitbook is trying to create a community and entice new potential clients.

G. A Decentralised Sharing App

The app is designed to rent homes and all other things in general. The system architecture consists of smart contracts, Ethereum clients to interact with them, and web UI. The logic of a smart contract is designed with four primary functions, create a smart contract, register an object, rent a device and reclaim it. Functions are entirely deterministic, and they are executed on Ethereum with no need for third trusted parties.

Creating a smart contract is straightforward by executing a transaction. The sender of the transaction becomes the owner of the contract.

Every smart contract has a map data structure objects. Calling method registerObject is creating a new record in the map consisting of a key, which is the object's ID and value with information about the object such as description or deposit due. Object id is represented as a QR code in the user interface to improve user experience and usage of the dapp.

If the object is free(no one is actively renting it), it can be rented by executing a transaction with an equal or higher object deposit. Deposit is one of the object properties defined in registerObject.

The reclaimObject gives the rented object back to the owner. Then it calculates and sends a rental fee to the owner, and the renter gets the remaining deposit. [9]

III. Polkadot

Polkadot is a multi-chain platform created to improve chains interoperability. The creator is Gavin Wood, the co-founder of Ethereum. It is written in Rust but also supports JavaScript

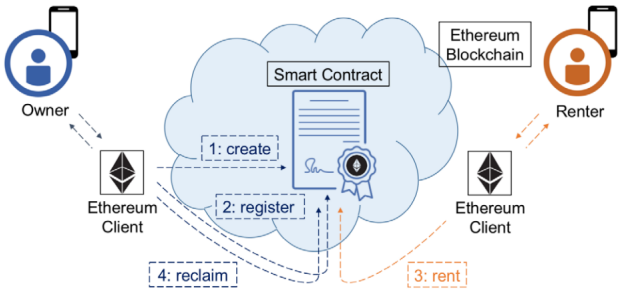

Fig. 4. Schematic procedure of the DAPP with the smart contract [9]

API. Polkadot contains two main components relay chain and parachain. The relay chain is the center of Polkadot and works as a hub for many chains. This chain takes place through governance mechanism, parachain auctions, or participating in the consensus mechanism. The primary responsibility is to coordinate the system, including parachains[8] Parachains are the actual blockchains connected to the network. The relay chain has ports to which para chains connect. It is also very promising for developers because security is the responsibility of the relay chain, and Polkadot itself is robust. Similar to parachain is a para thread. The difference is in purpose and connection to the relay chain. Imagine we have some feature that does not need to be connected to parachain all the time and has only time limitations. On this purpose serves para thread, when we can be connected to Polkadot for a short time if we need to.

There exist two tools that help Polkadot be very responsive in the context of blockchain, treasury, and Kusama. Kusama is Polkadot's test network but with real stakes. The better expression would be practice network. Kusama is supposed to test how chains are suitable to communicate with each other, how they work, not if they work. Therefore is a real stake chain to prevent spamming and hacking on-chain and only test serious chains and protocols. Polkadot use nominated proof of stake. In addition to validators and nominators, there are also collators. They collect parachain transactions and produce state transition proofs for the validators on the Relay Chain.[8] All validators validate on the relay chain in cryptocurrency DOT.

A. Moonbeam

Moonbeam is a parachain in the Polkadot ecosystem. Its uniqueness and usability are that it supports Ethereum smart contracts, which means that smart contracts codes from Ethereum are compatible with the Moonbeam network with no or minimal changes. Native token is called Glimmer, and as consensus algorithm is delegated proof of stake. If a smart contract or any other Ethereum mechanism is expected, proof of work, such as a pooling mechanism, will not work correctly on Moonbeam. Another difference is that Moonbeam uses governance mechanisms on-chain. However, on Moonbeam are available to smart contracts in Solidity, ecosystem tools such as block explorers, and developers' tools such as Truffle or Remix.

Similar to Ethereum, there are two types of accounts, user account, and smart contract. They are token holders, have a token balance, and can be controlled with the private key. However, Moonbeam has an additional third type of account called a proxy account. This type of account can perform a limited number of actions on behalf of another user. [10] Users can use a proxy account for keeping it in cold storage but can do actions with the weight of accounts tokens. Moonbeam also splits balances into five categories free - balance can be used from Substrate API, reducible can be used through Ethereum API, reserved misc frozen, and fee frozen.

The main difference in terms of consensus is that Moonbeam uses Delegated proof of stake. This brings the possibility of a better user experience than in Ethereum. We can check transactions using both Ethereum JSON RPC along with Polkadot JSON RPC with these steps:

1) We make a call to the network, and it gives us the hash of the latest finalized block
2) We fetch the block number from the given hash
3) We compare the retrieved block number with the block number of our transaction. It passed if our transaction was added in the previous block
4) Last check is retrieved block by number and check if transaction hash is included in the block

IV. Design

Based on the previous sections, we can see cross-chain sharing between Polkadot and Ethereum and between other networks. Nevertheless, an application that would act as Airbnb and provide the possibility of communication between Polkadot and Ethereum is missing. We chose the Polkadot MoonBeam network as the implemented environment. MoonBeam is an Ethereum based blockchain connected to Polkadot that allows us to perform solid, smart contracts. Ethereum smart contract compatibility allows us to have one code base and, at the same time, implement functionality for all Ethereum based blockchains that can process smart contracts.

Any user with Metamask and Ethereum or MoonBeam wallet can register on our website his real estate. Real estate is represented as a non-fungible token. This token we can send to MoonBeam or Ethereum as we like. The cross-chain transfer option gives users the freedom to have their resources on their preferred network. There may be several reasons for this, such as avoiding high fees on the current network or taking advantage of other benefits that the network brings. This step makes the application cross-chain also in terms of functioning and not only in terms of payment methods.

For cross-chain sharing tokens, we use two bridges: the chain bridge and the MyNFT bridge. A chain bridge is a bridge between MonnBeam and Ethereum and can transfer ERC20 tokens. Another bridge is the MyNFT bridge, a project in progress, but they already offer transfering NFT between MoonBeam and Ethereum for ERC721 tokens. We represent real estate by NFT. NFT is a blockchain solution to claiming ownership of a certain thing, so it cannot happen that an offer for real estate is created by someone to whom the property does not belong. This way, we can avoid potential fraud, and the NFT will not lose value outside of the application because it still represents real estate in the real world.

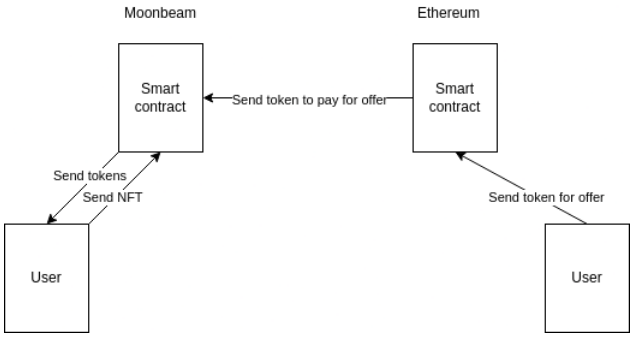

Fig. 5. Diagram of token flow in our design

The first scenario is a whole lending process. First, we must register real estate. Then we can set on which blockchain we would like to mint NFT. After confirmation mint in Metamask, we receive NFT. With NFT, we create an offer and set which real estate is lending, how long, and which types of tokens we accept. NFT is then locked by a smart contract for the duration of the offer. Another user can see listed offers. We choose one that fits his requirements. To accept the offer, we must choose a wallet and chain from which we will pay. He confirms token Transfer in Metamask, and the website creates a temporary record that this user owns the NFT. Therefore he grants access to the offering real estate for an agreed time. If there is no accepted offer, the user who created it can turn it off and unlock his NFT.

The second scenario is like we mentioned before for transferring NFT between chains. We can choose on which network we will store NFT. We can avoid a fee for cross-chain sharing in acceptance offers or use any advantage of holding NFT on any chain, like the Polkadot cross-chain sharing protocol. We open a list of our real estate and change for the current fee chain, which is NFT holding.

V. Summary and future work

Our primary focus was to create a peer-to-peer Airbnb-like decentralized application using cross-chain solutions. We were able to use bridges to provide cross-chain payment methods in the form of ERC20 tokens from Ethereum on Polkadot. We have also created some level of security using NFT as represents of real estate. We also provided usage of NFT outside the application ecosystem by doing this because it is a full-fledged representation of real estate in the form of NFT. Adding an NFT bridge to an application opens up NFT storage capabilities on any Ethereum based platform that the application will support. Using Polkadot MoonBeam allows us to write smart contracts in Solidity language, which means that the application's codebase applies to all Ethereum based networks. Therefore there is eventual theoretical support for all Ethereum based networks.

We want to add a payable method using native coins of supported networks in the near future. Our design only supports tokens, which can discourage some users because they do not have or want to use the native network currency. Another improvement would be implemented using a unified bridge that could carry both tokens and native currencies.

REFERENCES

[1] A. Zohar, "Bitcoin: Under the hood," Commun. ACM, vol. 58, no. 9, pp. 104–113, 2015.

[2] Pascal Lafourcade and Marius Lombard-Platet. "About blockchain interop- erability". In: Information Processing Letters 161 (2020), p. 105976. issn: 0020-0190.

[3] Alexei Zamyatin et al. "Xclaim: Trustless, interoperable, cryptocurrency-backed assets". In: 2019 IEEE Symposium on Security and Privacy (SP).

[4] Wood, Gavin. "Polkadot: Vision for a heterogeneous multi-chain framework." White Paper 21 (2016): 2327-4662.

[5] Dtravel whitepaper. url: https : / / whitepaper . dtravel . com / whitepaper - 1 / technology

[6] Qin Wang et al. "Non-fungible token (NFT): Overview, evaluation, oppor- tunities and challenges". In: arXiv preprint arXiv:2105.07447 (2021).

[7] Snowbridge documentation. url: https://snowbridge-docs.snowfork.com/

[8] Moonbeam documentation. url: https://docs.moonbeam.network/

[9] Bogner, Andreas, Mathieu Chanson, and Arne Meeuw. "A decentralised sharing app running a smart contract on the ethereum blockchain." Proceedings of the 6th International Conference on the Internet of Things. 2016.

[10] Polkadot documentation. url:https://wiki.polkadot.network/docs/learn-proxies