Plasm Network Version1.0

Takumi Yamashita takumi@stake.co.jp
Sota Watanabe sota@stake.co.jp Kim Hoon hoon@stake.co.jp

March 15, 2020

Reminder: This is the second draft. We will polish this draft to make it understandable and professional.

Abstract

The mission for Plasm Network is to provide a scalable decentralized application development method that defines and realize the new form of the web: Web3.0. This paper discusses the purpose of Plasm Network, the technical background information that is necessary to understand the blockchain industry, the reason this industry is important moving forward and what Plasm Network will bring to the industry. Furthermore, this paper will discuss the Plasm Network's core product specifications.

1 Introduction

We aim to realize **Web3.0** via blockchain technology. The traditional social structure allows people with authority to monopolize information, and history has proven that these people will bend the rules to their benefit [1]. Even if people claim that their system is fair, it still lacks transparency, proving that everything is still conceived upon pillars of sand that are called trust. In contrast to this, blockchain provides a system with decentralized governance that does not require a single group or organization to manage everything, effectively removing a single point of failure and making a transparent and trustless system. This is possible because blockchain is a system that allows anyone to view, prove and host with a highly fault-tolerant consensus mechanism [2].

For end-users to fully utilize the strengths of the blockchain protocol, there must be an application that provides an interface to them. Applications that work on top of blockchains are referred to as Decentralized Applications (Dapps). Currently, there are countless Dapps developed in the form of smart contracts and chain codes that are deployed on a blockchain, providing utility for various people. However, due to the decentralized nature of Dapps, the processing speed is far from fast. At the time of this writing, the biggest blockchain is Ethereum [3], and Ethereum has a transaction throughput of 15 transactions per second [4]. In contrast to this, VISA or Alipay can process around 1,700 transactions [5] and 256,000 transactions respectively [6]. It is true that the transaction speed for Dapps is very slow for users to utilize this technology to its full potential. To solve this issue, there have been several blockchain scalability solutions being proposed.

These are some of the well-known blockchain scalability solutions.

1. **SegWit** : Fixing transaction malleability by removing the signature information and storing it outside of the base transaction block [7].

2. **State channel** : Combining off-chain transactions among particular users and only the final state is committed to the main blockchain[8].

3. **Sharding :** Allowing many more transactions to be processed in parallel at the same time by making shards[9].

4. **Plasma** : Storing transactions in separate child chains and only the root hash is stored in the main chain[10].

From here, we want to focus on processing transactions outside of the main chain, called the Layer 2 solution. Layer 1 refers to public blockchains like Ethereum or Bitcoin. In recent years, this layer is suffering from increased transaction capacity, making the blockchain "full" [11]. From this, we can predict that in about a decade, blockchain will have a different usage, where Layer 1 is used as the trust layer, while Layer 2 is the transaction layer.

Among all layer2 solutions, the reason we focus on Plasma is that it is a scaling solution that is the least dependent on the processing performance of the main chain. In Plasma, an operator manages its side-chain without sacrificing decentralization. This means that many transactions can be handled in a centralized way that does not require a consensus process, but all participants on the side chain can safely exit by submitting fraud proofs. The scaling solutions used in the existing centralized system can be used as they are. Hence, it is possible to achieve high processing performance that is not feasible with a native distributed ledger. Plasma should be recognized as an indispensable technology in the future because it can dramatically improve processing performance for all distributed ledgers.

However, Plasma has several drawbacks. First, there is a limitation on what can be done with Plasma Applications (Plapps). All the things that Plasma can is described with The first-order predicate logic as "Predicate"[12].

Second, it is more difficult to make a Plapps compared to making a traditional DApps because writing and deploying smart contracts are not enough to make a Plapps. Plasma is the complicated technical stack that consists of several components [13]. Precisely, Plasma application consists of 4 components, a smart contract on a parent chain, a child chain, an operator, and a user. We solve these two problems through **"Plasm Network"**. This provides a set of standard libraries that enable us to write Predicate. And we provide cloud service to deploy and manage the Plasma components.

With these tools, Plasm developers can build their applications easily.

We use OVM [13] for Plasma. OVM is an abstraction of all Layer2 Solutions, not just Plasma. Layer2 Solutions include the State channel, Optimistic Rollup which solves the problem of transaction history availability with the trade-off of Plasma's infinite scalability, and also. By using OVM, you will be able to use Layer2 Solution other than Plasma like these.

We are planning to build these systems around Polkadot[15]. Polkadot is a heterogeneous multi-chain framework that empowers blockchain networks to work together under the protection of shared security. In addition, there is a framework to create blockchains called Substrate[16]. Currently, Polkadot itself and parachains are created with Substrate. In the future, we think blockchains will be paralleled simply because there is no single perfect blockchain which supports all governance models and customer's needs by itself. Hence, there are more than 900 public blockchains have been built and more and more blockchains are being created. Polkadot and Substrate empower this movement of creating the perfect custom blockchain based on the user's need. Plasma is one of the most promising domains on Polkadot as Dr.Gavin mentioned at Subzero Summit[17]. We decided to make Plasma solutions for Polkadot and Substrate.

2 Plasm Network

Plasm Network is the Project that provides the developing scalable decentralized application methods for the developer. We will achieve this purpose by the figure1 architecture.

Plasm Network is a Layer1 public blockchain built using Substrate. It has some features for deploying scalable Dapps and OVM modules, and standard implementations of smart contracts required for Plasma applications. Plasm chain is supposed to be the first scaling Parachain on Polkadot. In general, Plasm chain will be a default root chain for application developers to connect the application. Plasm Network gives new possibilities and comfortable usability to Plasma applications by incorporating original functions in the runtime. Also, Plasm Network uses the innovative token issuance algorithm to design a User / Developer-first token economy.

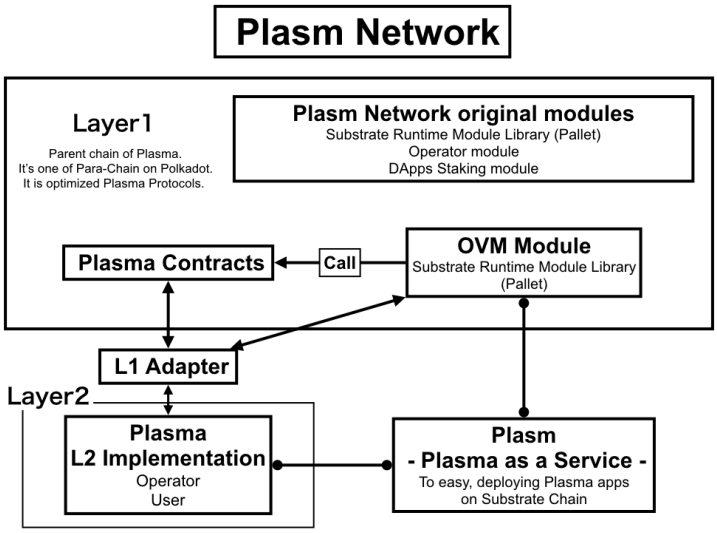

Figure 1: Plasm Network architecture.

Plasma layer2 implementations and Plasma as a Service will be deployed for supporting these. Plasma as a Service is a Platform as a Service for easily deploying Plasma applications. The service provides to select the main chain and deploy and manage customized child chains through the GUI. These will be implemented using Plasma Rust Frameworks [18] provided by Cryptoecnomics Lab [19].

The following sections provide an overview and technical details of each service.

3 Plasma

Plasma is one of the scale solutions in the blockchain. The basic idea of ☒ Plasma is to manage and process transactions in a Merkle tree outside the chain at high speed, and engrave only the Merkle root on the blockchain. The person responsible for performing the off-chain processing and submitting the hash to the blockchain is called an Aggregator in the context of Plasma.

Plasm Network supports "Plasma" which is based on Plasma-Cash. It has one NFT state not a transaction, at the leaves of the Merkle tree. Rules for performing state transition can be defined by OVM as described later. Figure1 shows an example of an NFT state transition that has ownership as a state and the necessary Transaction.

In the figure3 case, in order to make a state transition, 1. It must be signed by "Owner" 2. A new state must be specified for output 3. A state must not have already been transitioned in another way, this is described using OVM. The logic described here is called "Predicate". It is described in first-order logic. When OVM receives the accepted Transaction, it changes the state and updates the Merkle route.

In Plasma, a single Aggregator handles these transactions and submits the Merkle route. If the Aggregator cheats, the transaction submitted by the user may be falsified. Plasma can dispute the correctness of transactions and states on the main chain using OVM and Predicate described above for such tampering. This allows Plasma to combine both the fast transaction processing power by a single Aggregator and the strong security of the blockchain.

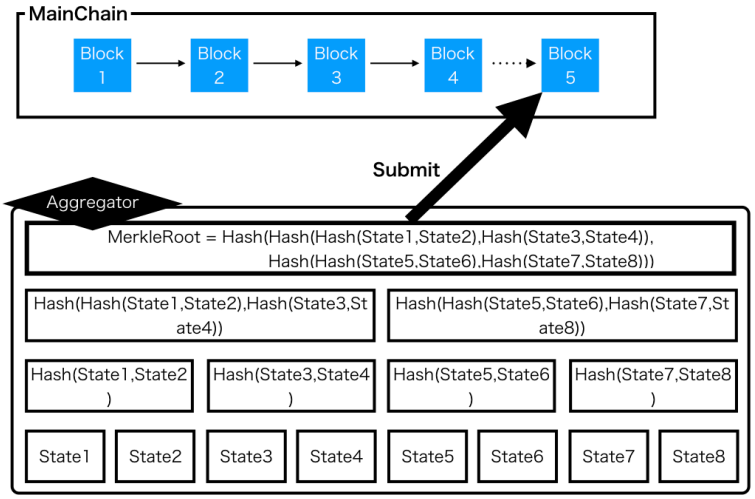

Figure 2: Plasma state.

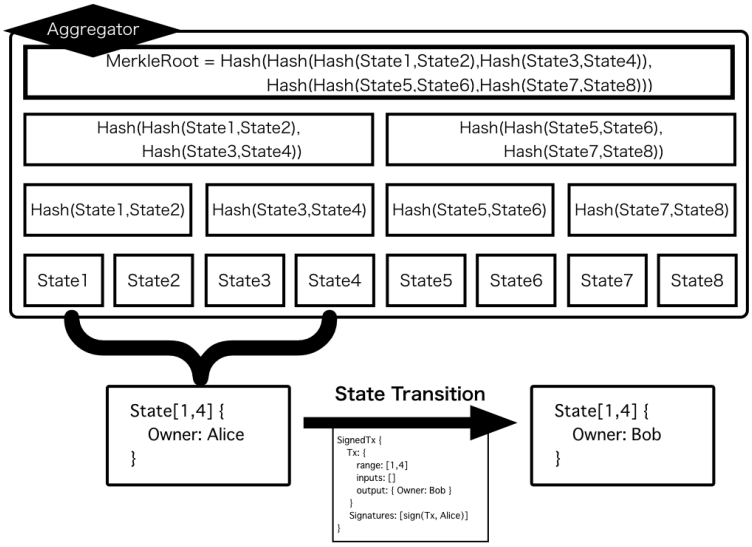

Figure 3: Plasma state transition.

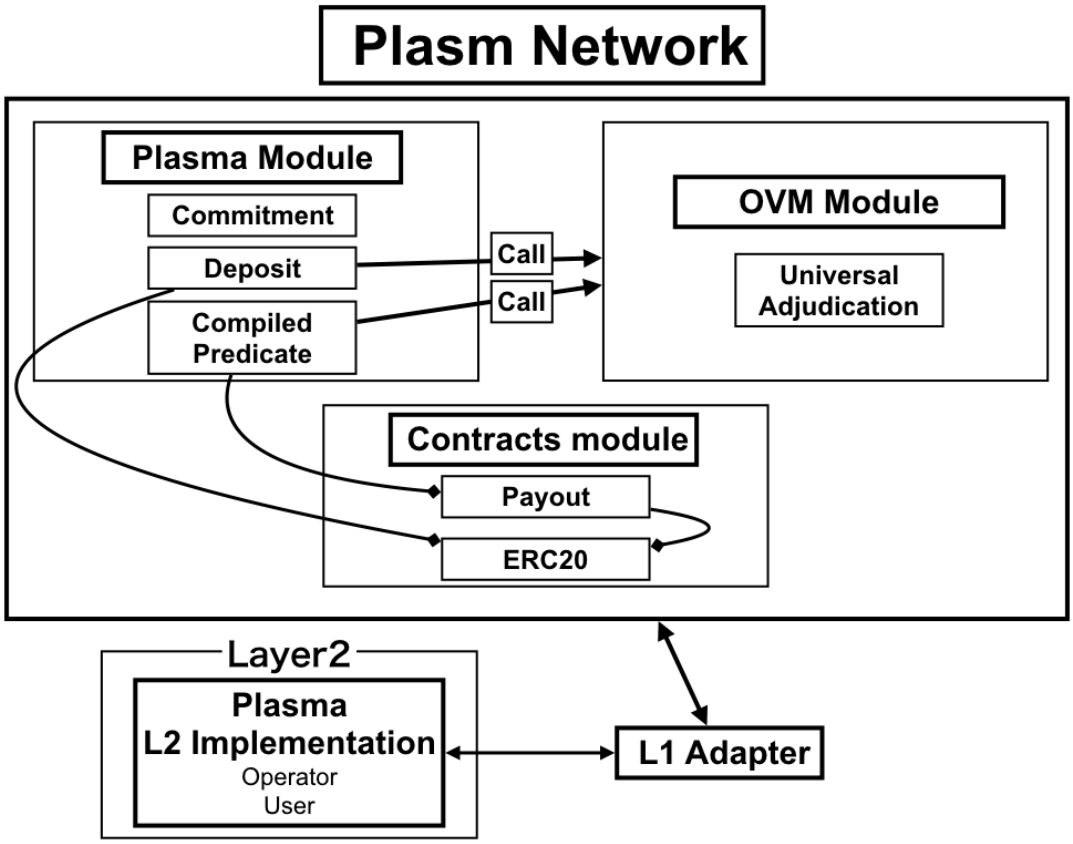

Figure 4: OVM modules.

4 OVM

The OVM(Optimistic Virtual Machine) is a set of standards that streamlines and unifies different Layer2 protocols. We can express complex dispute logics via a single OVM language, and that language consists of Optimistic Game Semantics[13]. For example, we can express Plasma checkpoint and exit claims with 2 simple definitions (we call these "property") by OGS.

Plasm Network divides the OVM from the smart contract and uses smart contract function as a module so that OVM can be used more simply and conveniently.

The OVM and its surrounding architecture are as shown in the figure4.

Plasma applications (Plapps) can be created and execute via the dedicated client application called L1 adapter. On the other hand, Ethereum's Plasma implementation consists of multiple modules that are managed by a set of smart contracts. However, this method makes it is difficult to predict the gas price for a plasma application as these contracts contain several layers of logic that work behind the scene. Furthermore, having multiple smart contracts working in the back makes the implementation process more confusing for developers. In contrast to this, our OVM implementation on Substrate will work differently. Instead of using multiple contracts working interdependently, we abstract these inner workings into three major modules: the OVM module, the Plasma module, and the Contract module.

The OVM module has a function called Universal Adjudication. Users can cause a dispute when they find a malicious attack on the layer1 blockchain. The Plasma module supports a set of smart contracts that are necessary to make a Plasma structure. Lastly, the Contract module manages the implementation of application-specific logic and functions.

The Plasm Network uses the aforementioned modules as building blocks for allowing developers

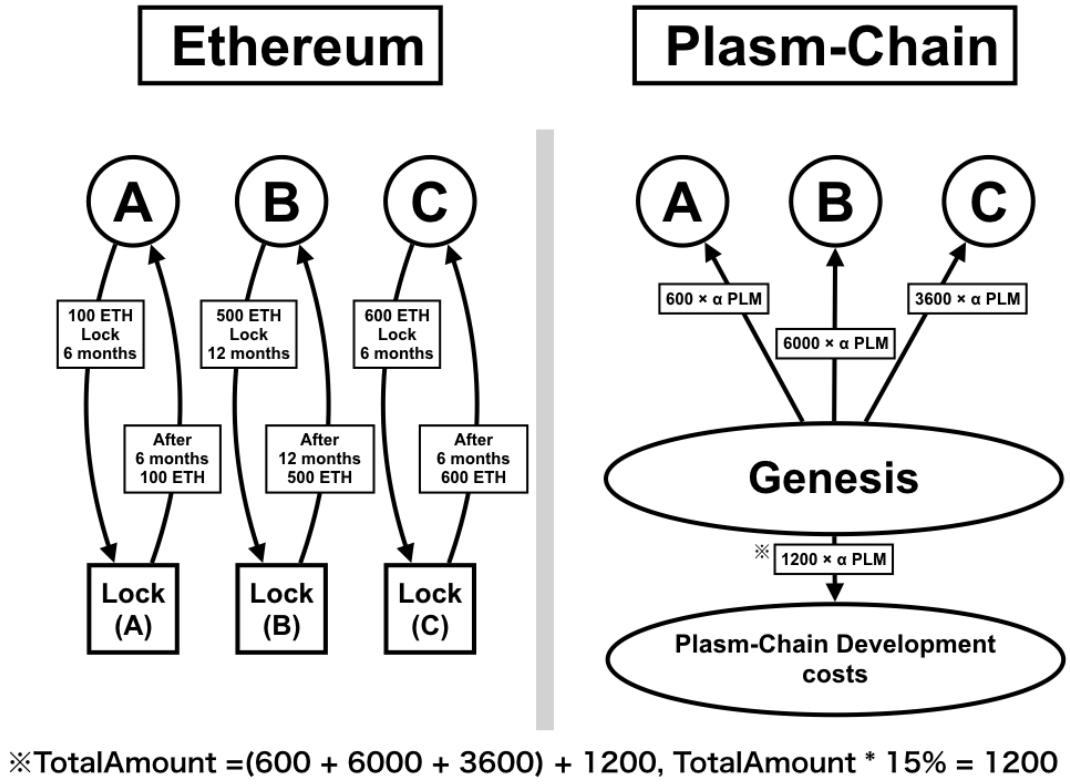

Figure 5: Lockdrop overview

to create applications implementing Plasma L2.

5 Lockdrop

Lockdrop[21] is a new low-risk economic incentivization mechanism, where it uses opportunity costs rather than legal tender (or assets) as collateral. Plasm Network uses this mechanism to issue tokens with monetary value. Throughout this section, we will explain Plasm Network's token issuance mechanism. The concept of a lockdrop was first conceived by Edgeware[22], and the one used for Plasm Network is an expansion of its original mechanism. The native token used in the Plasm Network is written PLM and pronounced as "plum". PLM will only calculate from the 15th decimal place and truncate any numbers below that. For more information regarding the role of the Token, please refer to the PLM Token Economics section14.

5.1 Lockdrop overview

For our first lockdrop, we will be using Ethereum's opportunity cost. Therefore, further sections will make the assumption that the locked token is ETH. However, lockdrop itself is an algorithm that can be implemented to any chains that support TimeLock and is not limited to Ethereum. Figure5 shows an example of how the lockdrop will work on the Plasm Network.

A lockdrop will work by the following process.

1. Ethereum token holder will send the number ETH locking, and the duration of the lock as a transaction to the LockContract that resides within the Ethereum blockchain.

2. For every token holders who participated in the lockdrop, the number of PLM calculated by total locked ETH × Lock bonus per duration $\times\alpha$ will be recorded on the Plasm Network genesis block.

3. The Plasm Team will take total issued amount $\times 15\%$ PlasmTokens from the genesis block.

4. Once the lock duration that the token holder specified has passed, the exact number lock ETH will be returned back to the participant after the lockdrop.

Our assumption is that the Ethereum token holder's opportunity cost is proportional to the number of tokens locked and the duration of the lock. PlasmToken is able to generate value via using those opportunity costs as collateral. Furthermore, the final token supply is not decided. This is to ensure fairness to tokens issued from post-genesis lockdrops. 15% of the total tokens circulating from the lockdrop will go to the Plasm team as part of the development fee. To elaborate, the tokens will be distributed multiple times via the following method.

6 Multi Lockdrop

Multi-Lockdrop is a mechanism in which we repeat the aforementioned lockdrop multiple times. Plasm Network will do this in a total of 3 times. Because of this, Plasm Network's total token supply will not be made concrete at genesis. Tokens will be issued every 3rd lockdrop, and additional tokens will be used via utilizing the "Staking" function, which will be later explained in detail.

There are two main reasons why we have decided to divide the lockdrop into multiple times. First, is to prevent uneven token distribution, as if the number of early bird participants was low, it is possible that there might be someone who holds the majority of the total supply. Furthermore, if we commence a rollback to the previous block state to fix this issue, the integrity of the network itself may be damaged. In a blockchain, it is important to establish a rule before the launch, we must avoid any situation in which we go against the predefined rules. To solve this problem, we have developed an algorithm that does not define the total token supply at genesis. The second reason is to create room for experiments so that the team can ensure that the Plasm Network can scale and be decentralized without any hiccups. The strong security and integrity of a blockchain rely on the distribution of nodes and token holders. It is not desirable to hold the security after the official launch at Stake. With this, repeating the lockdrop three times allows us to understand the distribution of tokens amongst the holders beforehand, which also leads to reducing maintenance costs for fixing these issues and preventing any risks that are followed by such a fix. This aligns with our goal of making Plasm Network a complete public blockchain.

Furthermore, Plasm Network will accept the following tokens for the 1st, 2nd and 3rd lockdrop.

- 1st: ETH
- 2nd: ETH, BTC
- 3rd: ETH, BTC
- Polkadot parachain auction: DOT(Note: That's special lockdrop. The details is section**??**.)

6.1 Defenitions

We define the amount of distributed PLM ($TotalPLM^{genesis}$) from the first lockdrop to be as the following.

$$TotalPLM^{genesis} = 500,000,000$$

The total amount will be distributed to the lockdrop participants in accordance with the token issue rate (IssueRatio). The IssueRatio is proportional to the number of locked tokens,

the exchange rate in dollars ($DollarRate_{token}$) of the locked tokens at the time of the lockdrop and the number of days multiplied by 1.0005 to the power of days ($Days \times 1.0005^{Days}$). The value of 1.0005 is based on Polkadot's interest rate. To elaborate, by default, Polkadot defines its maximum average annual interest rate to be 20% [23]. Converting this into daily interest rates with compound interest gives us an approximate value of 0.05%.

The users have the option to choose the lockdrop duration from the following 4 + 1 options table6.1. The $IssueRatio$ will be determined by the duration fo the lock which comes directly after evaluating the value of the locked tokens in Dollars.

Locked days	LockBonus
30th	×24
100th days	×100
300th days	×360
1000th days	×1600
About 2 years(※DOT only lockdrop)	×2000

Table 1: Lockdrop bonus table.

※The 2 years option is only available for locking DOT tokens. Furthermore, the DOT lockdrops are special in that they are only allowed to lock for 2 years. More information can be found from the **Polkadot auctions Lockdrop** section8.

Based on the aforementioned information, the IssueRatio will be defined as the following.

- $Locked_{token}$ is the number of locked tokens for the lockdrop
- $DolalrRate_{token}$ is the value for 1 token in Dollars
- $LockBonus_{days}$ is the amount of bonus the user will receive according to the locked days

$$IssueRatio = Locked_{token} \times DollarRate_{token} \times LockBonus_{days}(token \in \{ETH, BTC, DOT\})$$

The distributed tokens will be determined based on the IssueRatio. The algorithm for this will be like the following.

- Let n be the number of lockdrop participants (users)
- Let $IssueRatio_i$ be the $IssueRatio$ for user i
- Let PLM_i be the number of PLM user i will receive Considering that the Plasm development team will get 15% (17/20) of the total issued tokens, the receiving number of tokens will be like the following.

$$PLM_i = TotalPLM^{genesis} \times \frac{17}{20} \times \frac{IssueRatio_i}{\sum_{j=0}^{n} IssueRatio_j}$$

In other words, PLM will be distributed by the ratio of your $IssueRatio$ to the total $IssueRatio$. At this time, 75,000,000 PLM, which is 3/20 as development cost, will be used. Here, we define $TotalIssueRatio$, which is the sum of $IssueRation$.

$$TotalIssueRatio = \sum_{j=0}^{n} IssueRatio_j$$

Also, α_1 is the amount of PLM issued per unit $IssueRatio$ in the first Lockdrop. This is an important value to determine the amount of PLM issued in the second and subsequent Lockdrops.

$$\alpha_1 = \frac{PLM_i}{IssueRatio_i} = TotalPLM^{genesis} \times \frac{17}{20} \times \frac{1}{TotalIssueRatio}$$

Define the number of PLM issues per unit $IssueRatio$ for the second and third times to satisfy α_2 and α_3 the following equation.

$$\alpha_1 : \alpha_2 : \alpha_3 = 6 : 5 : 4$$

From the above, the amount of PLM distributed to the second and third user i is as follows.

$$\alpha_j \times IssueRatio_i \quad (j = 2, 3)$$

This allows the user to get the amounts of tokens proportional to $IssueRatio$ on the second and subsequent Lockdrops. This solves the problem that if a large number of users do Lockdrop after the first Lockdrop, the amount of PLM that users can get will be excessively small relative to the overall ratio.

The following figure6 shows an example of how the amounts of tokens distribution changes in multiple Lockdrops. Here, $DollarRate$ is fixed.

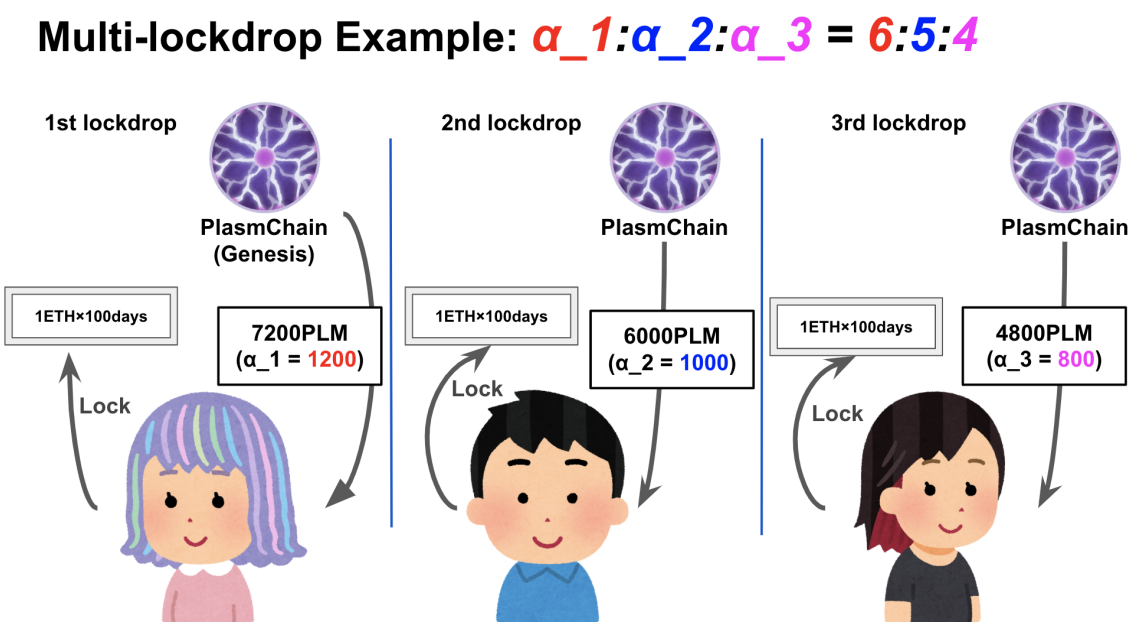

Figure 6: Multi Lockdrop example.

6.2 Why issue Lockdrop tokens?

- We do not hold the assets of the Lockdrop user.
- Users do not need to consider the risk of stealing assets by scammers. Issue a PLM with the opportunity cost as collateral. Assets locked by the user will return after the Lock period has expired.
- Users can join Lockdrop at a low cost. Anyone who can run a smart contract can participate in Lockdrop. All token holders have the opportunity to participate.
- Unlike Airdrop, Lockdrop participants pay a cost to get a PLM. You can issue tokens with a non-zero value.

6.3 Why split Lockdrop multiple times?

- If all PLMs are issued on the first Lockdrop, a small number of token holders may own a huge amount of PLM. If you do, there is a risk that a healthy ecosystem will not work. Prevent excessive first-mover benefits.
- Users can increase their chances of acquiring tokens by performing multiple Lockdrops. Allow more people to earn a PLM.

7 Real time Lockdrop

Real-Time Lockdrop is a mechanism for 2-nd and 3-rd Lockdrop in Multi-Lockdrop described in the previous chapter. In 1-st Lockdrop, after the period, tokens are issued at once in the Genesis block. Real-Time Lockdrop allows you to get a PLM token immediately after you lock during the Lockdrop period. The details are here.

8 Polkadot acutions Lockdrop

Polkadot auctions Lockdrop means the Lockdrop using DOT described in the previous chapter. It is independent and distinct from 1-st, 2-nd, and 3-rd Lockdrop. DOT at Polkadot has some roles. Some of the important roles are staking and Parachain deposit. Staking is used in the NPoS consensus algorithm to secure the chain. Parachain deposit is depositing a DOT for a certain period to join as Parachain on Polkadot. Parachain means a blockchain that connects to Polkadot. By becoming a Parachain it can borrow Polkadot's validator and share security and get Interoperability with other Parachains by using XCMP [24].

Polkadot auctions Lockdrop uses Parachain deposit. Locked DOT will be used to deposit Plasm Network into Parachain. The lock period of DOT expects about two years, during which Plasm Network operates as Parachain. Keep in mind that this Lockdrop can fail because the decision to join Parachain is made at auction. If unsuccessful, DOT will be returned without being locked and PLM will not be got. If successful, DOT is locked and you can get PLM tokens. Figure7 shows the procedure of Lockdrop in Polkadot.

Figure 7: Polkadot auctions Lockdrop.

This system uses the crowdfund module[25]. Also, note that the DOT will not be available during the auction too and it may fail. Because of this, there is a certain risk compared to other Lockdrops, and the LockBonus of Lockdrop by DOT is the highest. Refer to Multi-lockdrop in the previous sections6.

Plasm Network's Lockdrop cost design is based on the cost of DOT Lockdrop. The cost performance of Lockdrop can be calculated by benchmarking DOT Lockdrop and DOT Staking which is a trade-off relationship.

9 Lockdrop Affiliate Program

The Lockdrop Affiliate Program is a program made to incentivize those who have shared any information regarding Plasm Network to their peers. Via the affiliate program, participants of the Lockdrop will be able to receive PLM tokens with a bonus rate. In this section, we will discuss the mechanism of the 1st Lockdrop affiliate program.

The affiliate program mainly has these three rules:

- Any participants can reference their introducer' s Ethereum public address (this is optional for the lockdrop)
- Given that the referenced introducer' s address is valid, the token issuing rate for the address being referenced will gain an additional 1% of the participant' s issuing PLMs.
- Any participants who have referenced a valid introducer will receive another 1% increase in their initial PLMs.

The PLM given as a bonus is allocated in from the tokens held by the community of Plasm Network (15% of the total). The method of becoming a valid introducer is released in the Plasm Network' s Discord server. Furthermore, the valid introducers for the 2nd and 3rd lockdrop will be pooled from the participants of the 1st lockdrop.

10 Consensus Algorithm

The Plasm Network changes consensus algorithms and reward designs step by step to maintain security. Specifically, it initially takes the form of a Proof of Authority that is run solely by Validators selected by the community. Next, we will change to rewards centered on collator [**?**] to participate as Parachain. Eventually, we will move to NPoS, which is also used by Polkadot's Relaychain. Please refer to the PLM Token Ecosystem chapter for details on how to distribute rewards.

10.1 Proof of Authority

Proof of Authority is a consensus-building algorithm that operates only with a validator selected by the community. Public blockchains can be vulnerable when few are launching reliable validators. For this reason, PoA will be operated until a sufficient distribution of token holders and the existence of potential validators can be confirmed. The validator at this time will be paid according to the specified parameters as in the case of PoS.

10.2 Incentives of Collator

Incentives of Collator is an incentive design for operating Collator in Parachain. The Plasm Network is expected to become Parachain of Polkadot. Parachain needs a Collator to collect transactions, monitor the transactions in the block, and send the block certificate to the Relaychain validator. Collators must be a full node, so you need an incentive to join Plasm Network as a collator. At first, the Plasm Network Collator is selected by the community, as in the PoA mentioned above. After that, it will be changed to be elected by NPoS described later. In other words, the node that was acting as a validator function as a Collator while the Plasm Network is a Parachain.

10.3 Nominated Proof of Staking

Plasm Network uses NPoS which is used on the Polkadot relay chain. This consensus algorithm consists of 3 steps.

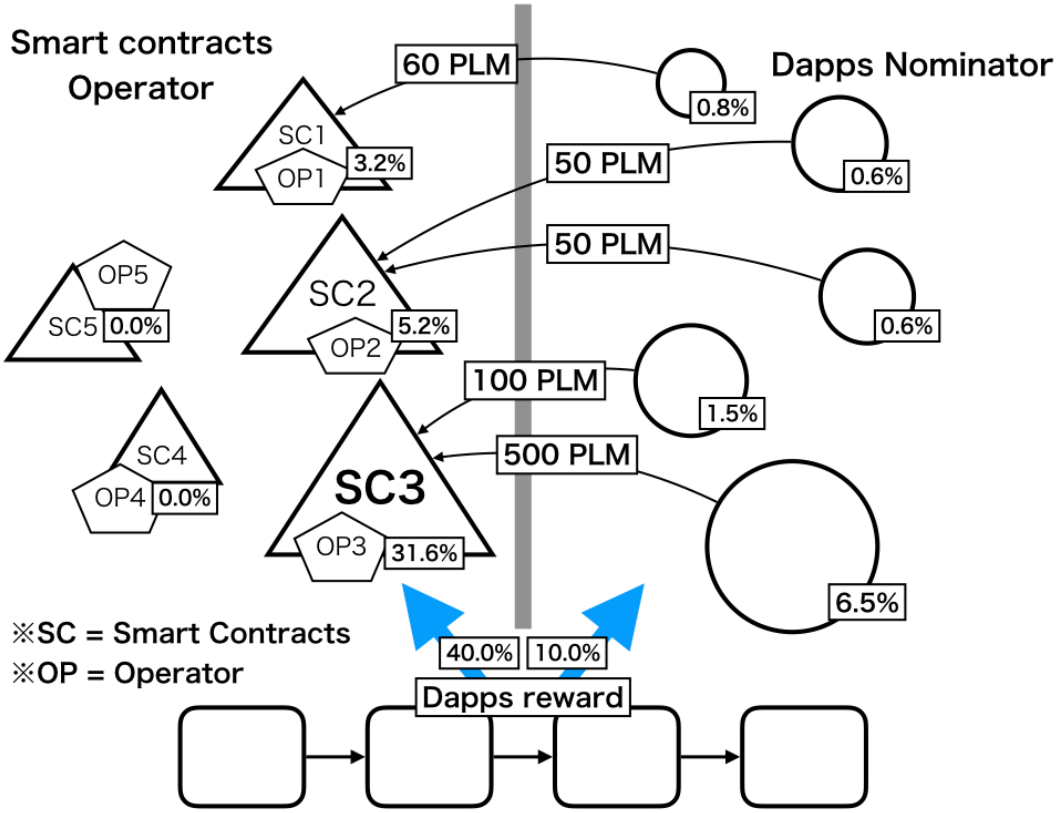

Figure 8: Dapps Rewrads

1. A Nominator selects a validator NPoS[27].
2. A validator verifies transactions and makes a new block BABE[28].
3. Finalize the block that was delivered in the network GRANDPA[29].

The block reward is distributed to the validator who created the block and his Nominator. In addition to that, the reward is also paid to the Plasm Network contributors as follows.

11 Dapps Rewards

Dapps Rewards is a mechanism that rewards developers or administrators of smart contracts on an ongoing basis. The figure8 is a overview of Dapps Rewards. 50% of Plasm Network's Staking reward goes to application developers who have enhanced the value of Plasm Network. The Plasm Network allows you to assign a smart contract administrator to a smart contract, and this administrator is called an "Operator". The user can also take smart contracts. This action is called Nominate, and the person who does it is called Dapps Nominator. As shown below, the operator of the smart contract receiving many nominates can receive the newly issued PLM token from the chain.

We will define how to distribute this reward to Operator and Nominator respectively. Define the following variables:

- $Rewards_{nominate}$: The total rewards allocated to Nominator.
- $Rewards_{contract}$: The total rewards allocated to smart contracts.

- $Rewards_{nominate_{i,j}}$: The rewards allocated to the j-th Nomimate fro the i-th smart contract.
- $Rewards_{contract_i}$: The rewards allocated to the operator of the i-th smart contract.
- n : The number of smart contract.
- m_i : The number of Nominate against the i-th smart contract.
- $stake_{i,j}$: The amount of PLM staked by the j-th Nominate for the i-th smart contract.

Then, $Nominate_{i,j}$ gives the following reward for this stake.

$$Rewards_{nominate_{i,j}} = Rewards_{nominate} \times \frac{\sum_j^{m_i} stake_{i,j}}{\sum_i^n \sum_j^{m_i} stake_{i,j}}$$

The nominator can get a reward proportional to the ratio of your stake amount to the total stake amount for the smart contract regardless of the smart contract selected. The operator of $contract_i$ who received Stake will get the following reward.

$$Rewards_{contract_i} = Rewards_{contract} \times \frac{stake_{i,j}}{\sum_i^n \sum_j^{m_i} stake_{i,j}}$$

On the other hand, the operator can get a reward proportional to the ratio of the stake of the smart contract owned by oneself to the stake of the smart contract. This creates an incentive for the nominator to stake on smart contracts that would simply increase the value of the token. Operators can also receive semi-permanent rewards by receiving stakes on smart contracts managed by themselves. We hope this will be an innovative solution to the difficult problem of monetizing application developers (administrators) on the chain.

Note: The operators and nominators have to wait to receive rewards.

However, this system can consider the following bad cases:

- **Malicious sock puppet**
  - The problem worthless operator stakes to himself and get rewards.
- **A few popular operators are staked from almost nominators**
  - A few popular operators are oligopoly.

The below writes about counter measures about each.

12 Malicious sock puppet

This is the problem hat malicious operators or nominators stake a worthless operator and get rewards.

The figure9 solves to install the voting system that people they are staking can vote a Good or Bad to each operator. Operators are punished below depending on voting results.

- If an operator receives voting that the number of Good is more than and equal to 4 times as the number of Bad and already has been running over a certain period of time, it is better. So it and nominate to it can normally get rewards. (i.e $Good \geq Bad \times 4$)
- If an operator receives voting that the number of Good is less than 4 times as the number of Bad or already has not been running over a certain period of time, it and nominate to it can not get rewards. (i.e $Good < Bad \times 4$)
- **If an operator receives voting that the number of Good is less than 3 times as the number of Bad, the tokens of staking to it are locked. (i.e $Good < Bad \times 2$)**

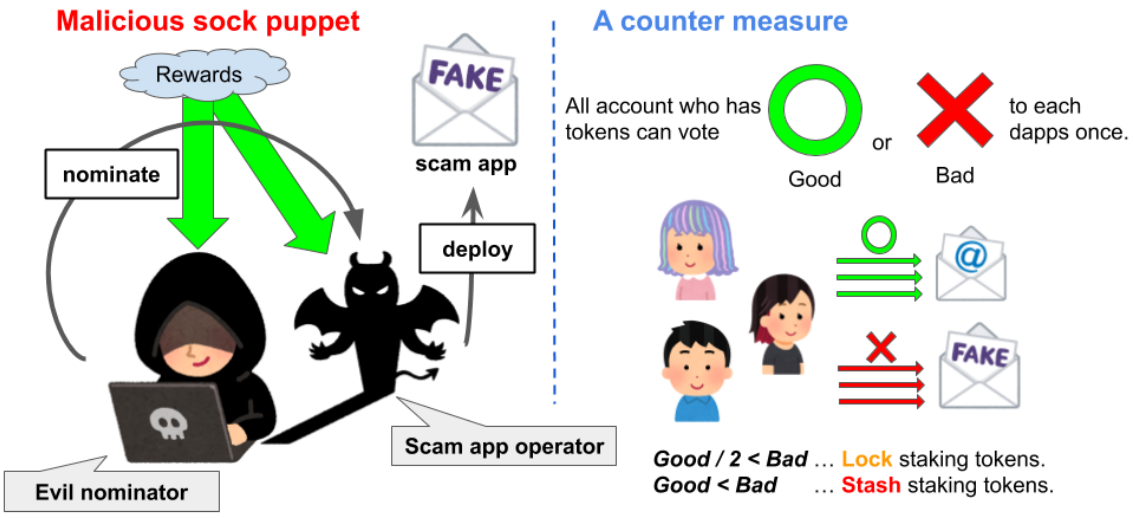

Figure 9: Malicious sock puppet and its a counter measure.

- **If an operator receives voting that the number of Good is less than 3 times as the number of Bad, the tokens of staking to it are slashed. (i.e** $Good < Bad$**)**

Then, you note that a user can vote with Sybil attack. However, a user has to stake his tokens in order to vote. And, voting doesn't make user profits directly. Therefore, while enough honestly users vote, a malicious user can not Sybil attack. Because a malicious user doesn't have incentives.

For example, Alice has 1 million PLM. Alice can make 1 million users and Sybil voting attacks. But, she is better to stake her tokens, but Sybil voting attack. (In actuality, she needs more tokens in order to make 1 million users because of transaction fees.)

12.1 A few popular operators are staked from almost nominators

When Plasmchain installs to the above system, many user stakes to a few operators that they are already stable running. Then the gap between rich and poor is growing. This solution solves the problem.

As shown in the figure10, give smart contracts first-mover benefits. Stakes for smart contracts have the option (optional) to get bonus rewards separately. This allows you to exercise your right to receive the following rewards up to $r(\geq 0)$ days after receiving your Stake reward. Where x^k represents x at some point k. First, introduce the following variables:

- $Rewards_{optoion^k_{i,j}}$: The rewards of getting by exercising an option that can be obtained at the j-th Nominate for the i-th Smart Contract at a certain location k.
- $Rewards_{contract_i}$: The reward of the operator of the i-th smart contract when exercising the option.
- $stake^k_{i,j}$: The amount of PLM staked by the j-th Nominate for the i-th smart contract at time k.
- m^k_i : The number of Nominate against the i-th smart contract at time k.
- $p^k_{contract_i}$: A coefficient parameter for determining the option reward obtained when nominating the i-th smart contract at time k, the operator of the smart contract can be defined.

$$Rewards_{option^k_{i,j}} = Rewards_{contract_i} \times \frac{stake^k_{i,j}}{\sum_j^{m^k_i} stake^k_{i,j}} \times p^k_{contract_i}$$

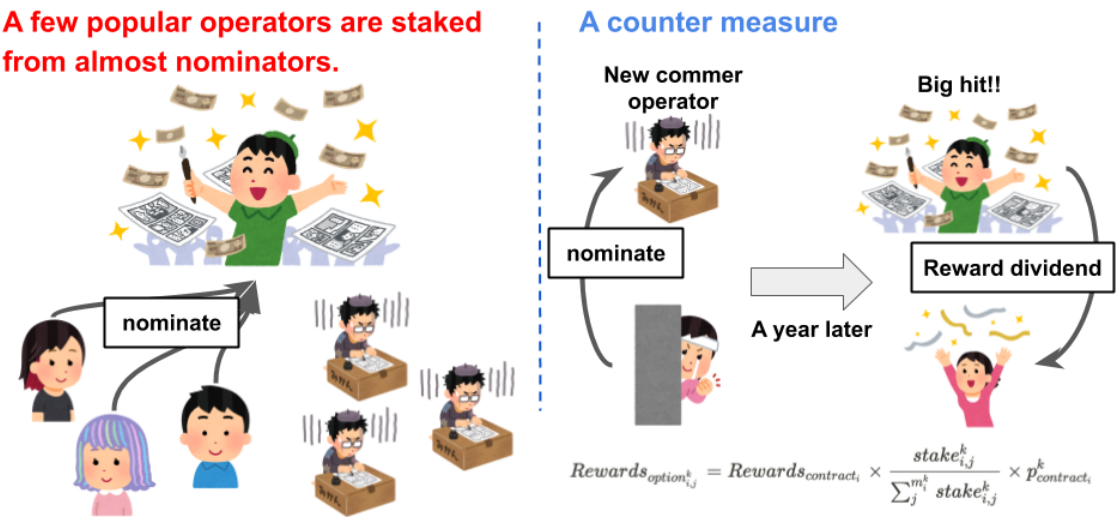

Figure 10: A few popular operators are staked from almost nominators and its a counter measure.

Note that $Rewards_{contract_i}$ and "Operator" indicated by $Rewards_{option_i^k}$ are equal. p allows the operator to specify a value less than 0.2.

When this right is exercised, the operator will be able to distribute the reward from Operator at the time of $p * 100$

When this right is exercised, the reward that the Operator can receive is reduced by that amount. So Operator can control the value p. Operator can specify r, p first.

Note: The limitation about r and p provides that malicious operator or nominator receives many staking rewards by misusing the system.

For example, An operator Bob was staked 100 PLM at 0 days and $p = 0.1$ and $r = 200$. Then Alice staked 50 PLM to Bob. Alice gets a credit that she would get rewards from Bob someday between 100 days after and 200 days after. The amount of the rewards is $0.5 \times 0.1 = 0.05$ times as the amount of Bob's rewards at that moment in time. 100 days after, Bob received 1000 PLM. Then Alice executed the credit and get $1000 \times 0.05 = 50$ PLM.

In general, the risk of staking to operators is higher than one of staking to the validator. Because slashing is difficult to predict compared to slash validators. And getting rewards takes more time. Therefore, we add the above incentive systems for operator staking. We think operator staking is advanced. So the ideal percentage of staking is $Validator : Operateor = 4 : 1$.

13 Operator Trading

Operator Trading is a mechanism to buy and sell Plasma application the operators. Since the right is tradable, this is similar to M&A. With the above Dapps Rewards mechanism, Operators can always benefit. If you can manage the Operator better, it is a good idea to give the Operator to another party. Of course, Operators are not always bought and sold. Operators will give their rights to opponents who give a value that seems reasonable to the value given to them. Those who have been granted the rights of the operators can receive the reward as operators. There is no need to actually switch operations at this time. However, the side that sold the Operator has already lost the incentive to operate the operator in good faith, and the owner of the new Operator to be transferred will inevitably operate the Operator. Through this mechanism, we assume that the new off-chain market will be created.

14 PLM Token Ecosystem

Plasm Network token ecosystem refers to Polkadot. Therefore, this document includes the same formula and values as Polkadot. The token name of Plasm Network is PLM that is called **"PLUM"**.

PLM has four main roles:

1. Staking for Consensus, rewards for validators and nominators.
2. Transaction Fee used to prevent harmful behavior.
3. Block rewards for Dapps operator, Sustainable reward design for applications.
4. Good / Bad Voting, Dapps Operator.

PLM is intended to be used as a liquidity token. Therefore, multiple Lockdrop tokens are issued to secure non-zero value collateral and a large number of holders. PLM tokens are expected to be operated at the ratio of $1:1 = Staking : Liquidity$, as the ratio of Staking is over 50% in many PoS-based cryptocurrencies.

$$1:1 = Staking : Liquidity$$

14.1 Inflation Model

Defines the algorithm that determines the issue amount and distribution method when issuing new Plasm Network tokens. The Plasm Network is structured that the new token issuance fee is shared with Dapps Rewards and a reward for securing the chain. The rewards for securing the chain are expected to become finally NPoS. Thereby, there are two types of Staking actions: Staking (NPoS) for Validator and Staking (Dapps Rewards) for smart contracts. Both rewards from each staking are equally proportional to the amount of staking. Users who take Stakes on validators / smart contracts are collectively called nominators. The ideal ratio of Staking for validators and Staking for smart contracts is: Here, $Staking_{validators}$ and $Staking_{contracts}$ are Staking for validators and Staking for smart contracts, respectively. Then, the below formula is expected the ratio between Staking for validators and Staking for smart contracts.

$$5:1 = Staking_{validators} : Staking_{contracts}$$

We consider the rewards paid to Operators in Dapps Rewards. Operator rewards increase in proportion to the inflation rate due to Staking. Dapps Rewards rewards 50% of the total reward when meeting the ideal q from quote the Dapps Rewards chapter. The rewards obtained by the Operator at that time is maximized. To show the specific reward distribution, we introduce the following variables:

- $Rewards_{operators}$ is the total amount of reward got by the Operator.
- $Rewards_{stakers_{validators}}$ is the total amount of rewards got by staking a validator.
- $Reards_{stakers_{contracts}}$ is the total amount of reward got by staking smart contracts.
- t is a coefficient that represents how many times the total amount of rewards earned by the operator is greater than the rewards earned by taking a smart contract.

$t = 4$ from quote the Dapps Rewards chapter and 50% of the total reward for meeting the ideal q will go to the Dapps Rewards reward. Therefore, the ideal distribution ratio of remuneration is determined as follows.

$$Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} : Rewards_{operators} = 5:1:4t = 4$$

Also, the percentage of Staking and the percentage of reward are equal as follows:

$$Staking_{validators} : Staking_{contracts} = Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}}$$

PLM tokens use the same NPoS as Polkadot. This nominator and validator can operate the token at a certain annual interest rate for Staking. Also, token rewards will be paid to the PLM Dapps operator's Nominator and Operator as well. Plasm Network's inflation model is defined as follows: First, follow the Polkadot inflation model and define the following variables:

- x is the total amount of staking divided by the total amount of tokens issued.
- X_{ideal} is the ideal value of x. $Staking : Liquidity = 1 : 1$, so $X_ideal = 0.5$.
- q is the amount of staking to the validator divided by the total amount of staking.

$$q = \frac{Staking_{validators}}{Staking_{validators} + Staking_{contracts}}$$

- Q_{ideal} is the ideal value of q. From $5 : 1 = Staking_{validators} : Staking_{contracts}$, the ideal value of q is $Q_{ideal} = 5/6$.
- $i(x, q)$ is the average annual interest getting by Staker. It is a monotonically decreasing function of x, $|Q_{ideal} - q|$ (difference from the ideal ratio). To make both x, q close to the ideal value, when x, $|Q_{ideal} - q|$ is low, raise the interest rate as an incentive to increase the amount of stake. When x, $|Q_{ideal} - q|$ is high, lower interest rates as an incentive to reduce stake.
- i_{ideal} is the average annual interest rate of Staker $i(x, q)$ when both x and q are ideal values. in other words, $i_{ideal} = i(X_{ideal}, Q_{ideal})$.
- $I_{Staking}$ is the inflation rate by Staking. $I_{Staking}$, a bivariate function involving x and q, draws a three-dimensional convex function. Expressing Staking total amount x interest rate = inflation rate and expressing it as $x * i(x, q) = I_{Staking}$ Also, this value is maximized when x, q is the ideal value from the reward design of $i(x, q)$. The ideal state equation can be expressed as $X_{ideal} * i(X_{ideal}, Q_{ideal}) = MaxmiumI_{Staking}$.
- I_0 is the lower limit of inflation rate. When $x = 1$ or $x = 0$, converge to the lower limit. I_0 is equivalent to the operating cost of the validator. The reason is that if you do not secure at least the incentive to operate the validator, the chain will break, so $I_0 = 0.025$ is recommended here.
- d is an adjustable decay rate for each x. Each time x is d more than X_{ideal}, $I_{Staking}$ is reduced by 50%. In other words, $I_{Staking}(X_{ideal} + d, Q_{ideal}) \geq I_{Staking}/2$.We recommend $d = 0.02$.
- g is an adjustable decay rate on q. Each time q is g away from Q_{ideal}, $I_{Staking}$ is reduced by 50%. In other words, $I_{Staking}(X_{ideal}, Q_{ideal}\ pme) \geq I_{Staking}/2$. We recommend $g = 0.15$.
- $i_{staking}$ is the average annual interest earned by the nominator through Staking. This can be determined by dividing inflation by the Staking ratio. In other words, $i_{staking} = fracI_{Staking}x$.
- $I_{operators}$ is the inflation rate due to the rewards that the Operator can get. This is t times the ratio $(1 - q)$ of Staking to Operator in $I_{Staking}$. From the below auxiliary formula1 and formula2, $I_{operator} = t(1 - q)I_{Staking}$.
- represents the average (based on the amount staken) interest rate of the operator's reward. From the auxiliary formula, $i_{operators} = fracI_{operators}x(1 - q)$.

- I is the overall inflation rate. This is $I = I_{Staking} + I_{operators}$, which is the sum of the reward for Staking and the inflation rate due to the reward for Operator.

$$
\begin{aligned}
& Staking_{validators} : Staking_{contracts} \\
= \quad & Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} \\
& Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} : Rewards_{operators} \\
= \quad & y : 1 : t \\
& Rewards_{stakers_{validators}} : (Rewards_{stakers_{contract}} + Rewards_{stakers_{validators}}) \\
= \quad & y : (y+1) \\
& Rewards_{staking_{validators}}(y+1) \\
= \quad & (Rewards_{staking_{contract}} + Rewards_{staking_{validators}})y \\
q = \quad & \frac{Staking_{validators}}{Staking_{validators} + Staking_{contracts}} \\
= \quad & \frac{Rewards_{stakers_{validators}}}{Rewards_{stakers_{validators}} + Rewards_{stakers_{contracts}}} \\
q = \quad & y/(y+1) \\
(y+1)q = \quad & y \\
yq + q = \quad & y \\
q + q/y = \quad & 1 \\
q/y = \quad & 1 - q \\
y = \quad & q/(1-q) \\
& Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} : Rewards_{operators} \\
= \quad & q/(1-q) : 1 : t
\end{aligned}
$$

$$
Rewards_{stakers} : Rewards_{operators} = q/(1-q) + 1 : t \tag{1}
$$

Here, the ratio of the amount of reward and the ratio of the inflation rate are equal from formula1.

$$
\begin{aligned}
I_{Staking} : I_{operators} &= q/(1-q) + 1 : t \\
I_{operstors}(q/(1-q)+1) &= I_{Staking} t \\
I_{operators} &= \frac{tI_{Staking}}{\frac{q}{1-q} + 1} \\
&= \frac{tI_{Staking}}{\frac{q}{1-q} + \frac{1-q}{1-q}}
\end{aligned}
$$

$$
\frac{tI_{Staking}}{\frac{1}{1-q}} = t(1-q)I_{Staking} \tag{2}
$$

$I_{Staking}$ is follows the below formula.

$$
I_{Staking} = \begin{cases} I_0 + x(i_{ideal} - \frac{I_0}{X_{ideal}}) \cdot 2^{-|q - Q_{ideal}|/g} & (0 < x \leq X_{ideal}) \\ I_0 + (i_{ideal} \cdot X_{ideal} - I_0) \cdot 2^{(X_{ideal} - x)/d - |q - Q_{ideal}|/g} & (X_{ideal} < x \leq 1) \end{cases}
$$

Figure11 is a graph simulating the inflation rate when each parameter is set as follows.

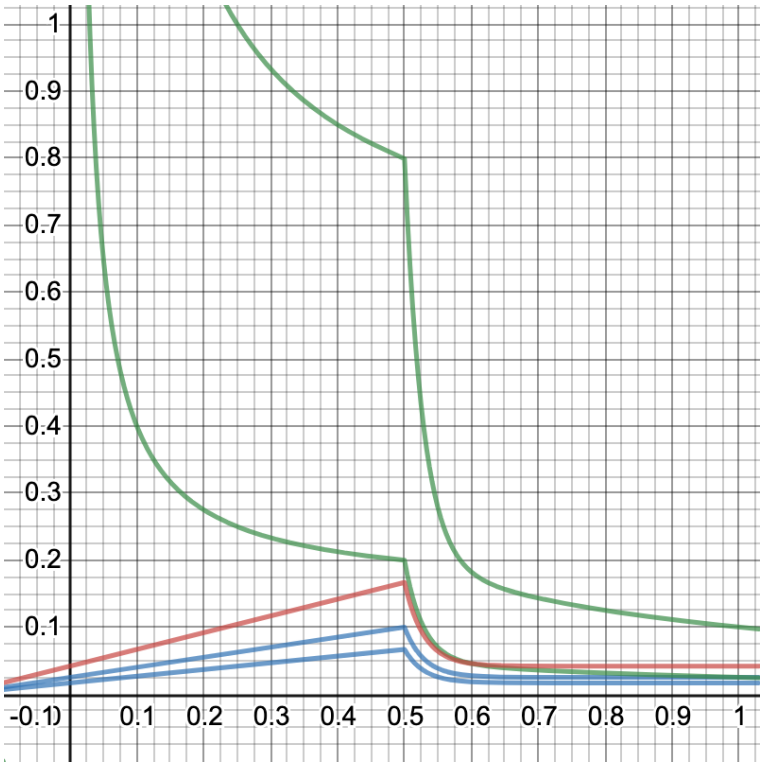

Figure 11: The inflation graph of $q = Q_{ideal}$.(`https://www.desmos.com/calculator/v8mrxdwbvz`)

$$
\begin{aligned}
i_{ideal} &= 0.2 \\
X_{ideal} &= 0.5 \\
Q_{ideal} &= 5/6 \\
I_0 &= 0.025 \\
d &= 0.02 \\
g &= 0.15 \\
t &= 4
\end{aligned}
$$

The figure11 graph is fixed at $q = Q_{ideal}$. Here, the upper green line is the average annual interest rate ($i_{operators}$) for the operator's staking amount, the lower green line is the average annual interest rate of the staking ($i_{staking}$), and the red line is the overall inflation rate (I), the upper blue line indicates the inflation rate due to Staking reward ($I_{Staking}$), and the lower blue line indicates the inflation rate due to Operator reward ($I_{Operator}$). The inflation rate when both x and q are ideal values is 0.166 ... $(1/6)$ at maximum. Next, the graph when $q = 0.2$ is shown in Figure12. When $q = 0.2$, the Staking percentage is $1 : 5 = Staking_{validators} : Staking_{contracts}$, and the reward percentage is as follows:

$$Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} : Rewards_{operators} = 1 : 5 : 20$$

When the figure12, although the ratio of the operator's reward is increasing, the upper green line representing the average annual interest rate for the operator's staking amount is low because q is far from the ideal value. As a result, even if the ratio of Staking to smart contracts increases, the reward paid to Operators is not much different from the ideal state. Also, the lower green line,

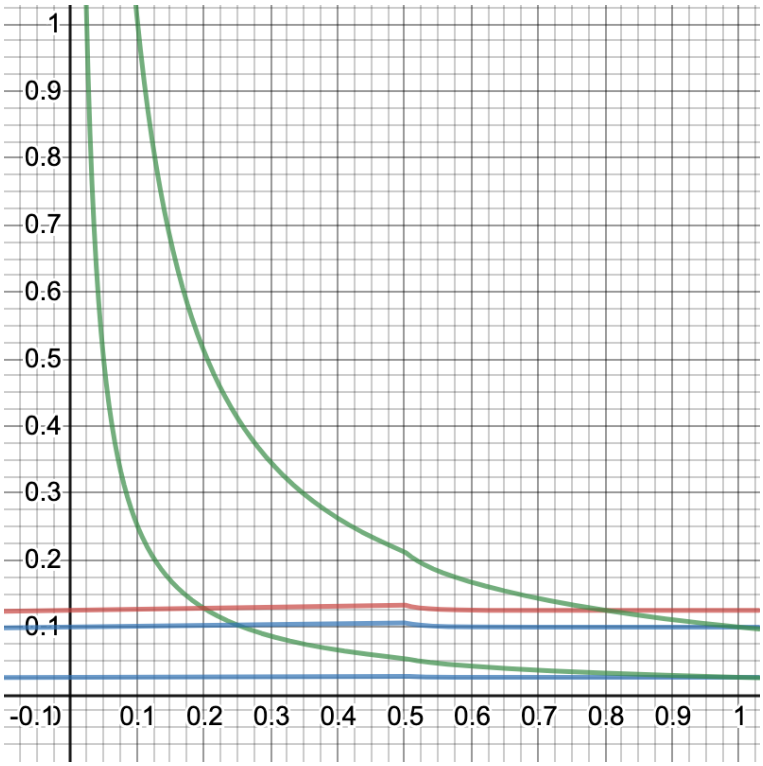

Figure 12: The inflation graph of $q = 0.2$.

which represents the average annual interest rate of Staking rewards, has been reduced, giving the incentive for Staker to take validators to Stake in order to maintain balance. As an extreme example, figure13 shows a graph when $q = 1.0$. At this time, no one has taken Staking for the smart contract, and the reward will be as follows.

$$Rewards_{stakers_{validators}} : Rewards_{stakers_{contracts}} : Rewards_{operators} = 1 : 0 : 0$$

When the figure13, the green line that represents the average annual rate of reward for Staking is lower than ideal because the reward that the Operator gets is zero and q is far from ideal. In this case, too, Staker creates an incentive to take Stakes on smart contracts to maintain balance. Note that the red line that represents the overall inflation rate and the blue line that represents the inflation rate due to the Staking reward overlap, making the latter invisible. Also, note that these graphs meet the following closings:

- The average annual interest functions $i_{staking}, i_{operator}$ are monotonic with respect to x.
- The average annual interest functions $i_{staking}, i_{operator}$ maximize when q is an ideal value.
- $I_{Staking}, I_{Operator}, I$ maximize when $xandq$ are both ideal values.
- I_0 is the lower limit of the inflation rate.
- Always satisfy $Rewards_{staking} : Rewards_{operator} = 5 + 1 : 4 = 6 : 4 = 3 : 2$ when q is the ideal value. In other words, when q is the ideal value, it satisfies $I_{staking} : I_{operator} = 3 : 2$.

By adding the above-inflation model, we will adjust the incentives of Plasm users and encourage the actions expected of Plasm Network.

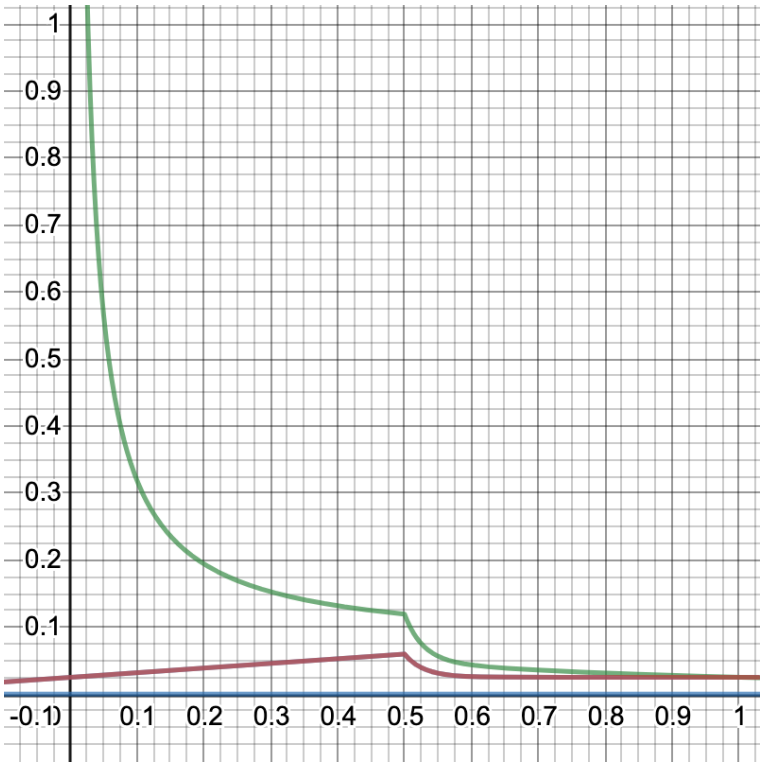

Figure 13: The inflation graph of $q = 1.0$.

14.2 Transaction fee

The transaction fee mechanism follows Polkadot's transaction fee adjustment algorithm[23].

Plasm Network does not have an organization called Treasury to store funds. The Plasm Network does not manipulate governance funds(Alternative, Plasm Network community address exists). At Polkadot, about 20% of the transaction fee goes to the validator and the rest goes to Treasury. Plasm Network remits about 20% to the validator, while the rest is burned. This serves as a tax levy to secure the value of the currency in tax monetary theory. The value of the token is secured by burning 80% of the transaction fee as a tax. Burning tokens at a fixed rate also serves to curb supply inflation in the inflation model described above.

14.3 Validator Staking

The Staking and security mechanisms for the Validator follow Polkadot's NPoS algorithm[27].

14.4 Collator Staking

The collator is responsible for collecting Parachain transactions, monitoring transactions in blocks, and sending block certificates to Relaychain validators. Plasm Network's Validator acts as a Collator while Plasm Network participates as Polkadot's Parachain. At this time, the variables in the inflation model are changed according to the demand of Collator.

14.5 PoA Staking

PoA is an algorithm that forms consensus only with authenticated validators. Plasm Network initially launches as PoA Network. Initially, the rewards are allocated equally to PoA participants. The reward paid to the validator at this time is the fixed value of $Rewards_{stakers_{validators}}$ in the

above-inflation model. After that, validators and users can take Staking. At this time, the variables in the inflation model are changed according to the number of PoA and the situation.

Plasm Network plans to seamlessly change the algorithm for securing the chain in the order of PoA to PoA Staking to Collator to finally Validator Staking (NPoS).

15 Plsm as a Service

Plasm as a Service(PlaaS) is a tool to deploy and manage a Plasma child chain easily under the Plasm parachain on Polkadot. Figure14 shows the main screen of Plasm as a Service.

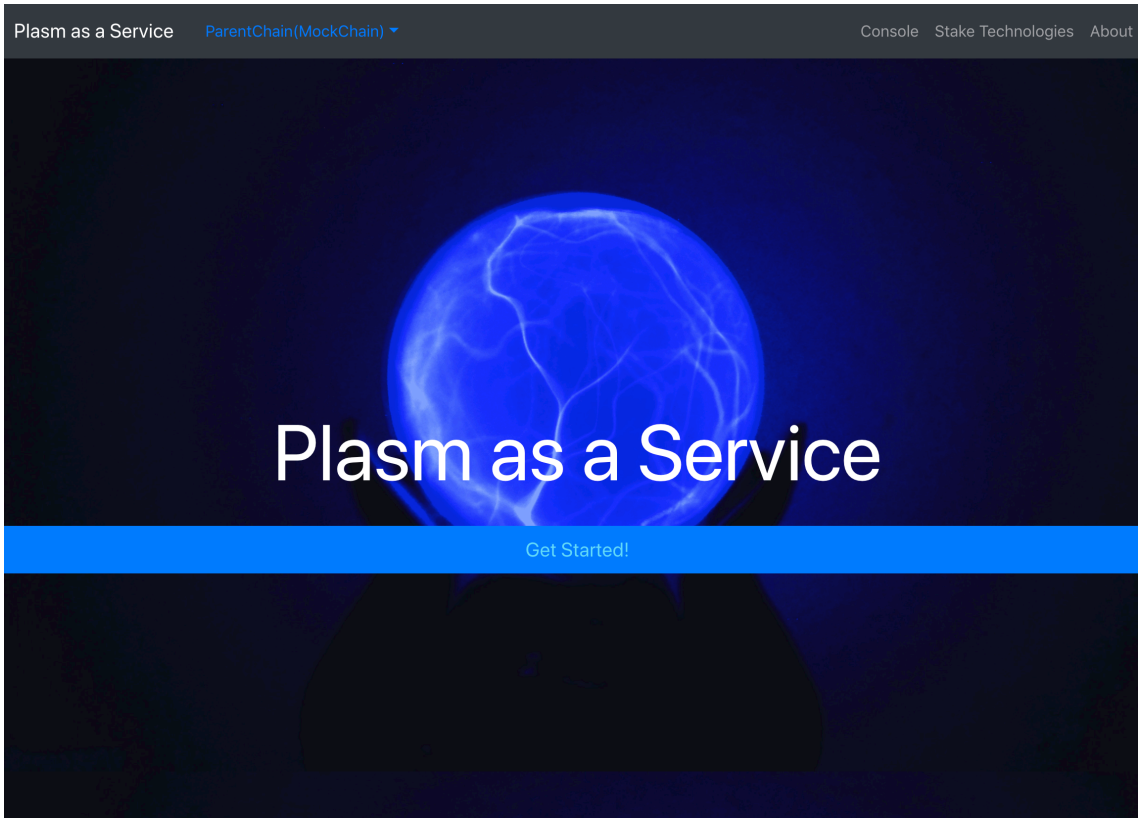

Figure 14: Plasm as a Service.

"Plasm" is a set of Substrate Runtime Module Libraries (SRML) which make it easier to use Plasma functions on Substrate. By using Plasm, any developers can make a child chain and deploy a Plasma application. Plasma applications take a different approach from a traditional approach. The limited capability of a layer 1 smart contract makes many DApps impractical. What Plasma application is doing is that making a set of Plasma components mainly on "off-chain" and connect them to the main chain. Plasma applications are next generation technology which gives developers new possibilities.

However, Deploying and managing a Plasma application is not easy.

As you can see from the figure2, Plasma consists of several components. It is difficult and takes time to implement these components from scratch. Plasma specific knowledge and high technical skills are also needed. To solve these problems, the Plasm team provides a GUI tool which makes it easier to deploy and manage Plapps.

15.1 Functions

PlaaS makes it easy to deploy and manage Plasma applications. More specifically, a user creates an application by the following steps.

1. **GetStart**: Move to the main page to make a Plasma application.
2. **Generate Plasma App**: Choose a template to make a Plasma application. As a default setting, Plasma Cash[20], one of the Plasma standards for payments is prepared. Other templates like Plasma Prime and Plasma ZK-SNARKs???
3. **Settings**: Fill out application information which is required to run a template.
4. **Deploy a Contract and a Child Chain:** Deploy a Contract and a Child Chain: Deploy the application and child chain that were created at the step 3. And,
    1. Deploy a smart contract on the parent chain.
    2. Deploy a server which has the DB of the child chain.
    3. Deploy the operator's account.
    4. Generate the genesis account.
5. **Console**: Manage the status of the Plapps. Specifically,
    1. The status on the parent chain
    2. The status of DB on a child chain
    3. Operator's account
    4. Registered accounts including the genesis account

    are controllable.

Through the functions listed above, developers can deploy and manage Plasma applications easily.

15.2 Demo

You can try a simple PlaaS application.
(※ This demo doesn't deploy an application) https://mock.d3dg769h9ndpcf.amplifyapp.com

16 Conclusion

Through the Plasm Project, we provide all developers with new methods to make decentralized and scalable applications, especially on Polkadot. Since the Polkadot relay chain does not support smart contracts, scalable smart contract platforms are necessary to build applications on top of it. Therefore, Plasma makes the Polkadot network more valuable.

In addition to that, since it is complicated to make Plasma applications, we provide Plasm as a Service to make it much easier. All developers will be able to make customized Plasma applications by using Plasm libraries depending on their needs. In terms of Plasm chain, the participant can buy and sell a particular application and its ecosystem like M&A. The individuals are creating not only a Plasma application but also an economy itself. Hence, a Plasma application is more than an application. We are very excited to see the future of the decentralized Plasma ecosystem.

17 Acknowledgment

Many thanks go out to Web3 Foundation, the creator of Polkadot and Parity Technologies, the creator of Substrate for making innovative protocols. Thanks to Cryptoeconomics Lab for reviewing and cooperating with us.

References

[1] Robert B. Reich, "Saving Capitalism: For the Many, Not the Few" https://www.jstage.jst.go.jp/article/lecgsa/15/0/15_139/_pdf/-char/en

[2] Mastering BitCoin : https://github.com/bitcoinbook/bitcoinbook

[3] Ethereum : https://www.ethereum.org/

[4] Ethereum Transaction Throughput : https://www.coindesk.com/information/will-ethereum-scale

[5] Visa Transaction Throughput: https://hackernoon.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44

[6] ALIPAY Transaction Throughput : https://www.barrons.com/articles/alibaba-records-25-3-billion-in-singles-day-sales-1510538618

[7] Segwit : https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki

[8] StateChannel : Jeff Coleman, Liam Horne, and Li Xuanji "Counterfactual: Generalized State Channels", June 12, 2018, https://l4.ventures/papers/statechannels.pdf

[9] Sharding : Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, Prateek Saxena, "A Secure Sharding Protocol For Open Blockchains", https://www.bubifans.com/ueditor/php/upload/file/20181015/1539597837236127.pdf

[10] Plasma : Joseph Poon, Vitalik Buterin, "Plasma: Scalable Autonomous Smart Contracts" August 11, 2017https://plasma.io/plasma.pdf

[11] Ethereum "almost full" as controversial coin gobbles up capacity : https://www.bloomberg.com/news/articles/2019-08-26/ethereum-almost-full-as-controversial-coin-gobbles-up-capacity

[12] About Predicate : https://docs.plasma.group/projects/spec/en/latest/src/02-contracts/predicate-contract.html

[13] OVM, Ben Jones, Karl Floersh, "Optimistic Game Semantics", January, 2020, https://github.com/plasma-group/website/blob/master/optimistic-game-semantics.pdf

[14] Plasma Components https://docs.plasma.group/projects/spec/en/latest/src/05-client-architecture/introduction.html

[15] Polkadot: DR. GAVIN WOOD, "POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK" https://polkadot.network/PolkaDotPaper.pdf

[16] Substrate https://www.parity.io/substrate/

[17] ink! https://github.com/paritytech/ink/wiki

[18] Plasma Rust Framework https://github.com/cryptoeconomicslab/plasma-rust-framework

[19] Cryptoeconomics Lab https://www.cryptoeconomicslab.com/

[20] Plasma Cash https://ethresear.ch/t/plasma-cash-plasma-with-much-less-per-user-data-checking/1298

[21] Lockdrop https://blog.edgewa.re/full-details-on-the-edgeware-lockdrop

[22] Edgeware https://edgewa.re/

[23] Polkadot Toke Economics : Alfonso Cevallos, Fatemeh Shirazi, 19.11.2019 https://research.web3.foundation/en/latest/polkadot/Token%20Economics.html

[24] XCMP, Rob Habermeier, Fatemeh Shirazi https://research.web3.foundation/en/latest/polkadot/networking/4-xcmp.html

[25] Crowdfund module https://github.com/paritytech/polkadot/blob/master/runtime/common/src/crowdfund.rs

[26] Collator https://wiki.polkadot.network/docs/en/maintain-collator](https://wiki.polkadot.network/docs/en/maintain-collator

[27] NPoS https://research.web3.foundation/en/latest/polkadot/NPoS/

[28] BABE https://research.web3.foundation/en/latest/polkadot/BABE/Babe/

[29] GRANDPA https://research.web3.foundation/en/latest/polkadot/GRANDPA/