Plasm Network version1.0.

Takumi Yamashita takumi@stake.co.jp

2020 年 3 月 15 日

概要

Plasm Network のミッションはすべての開発者にスケーラブルな分散アプリケーションの開発メソッドを提供することを通して新しい Web のあり方 **Web3.0** を実現することです. このドキュメントでは Plasm Network の目的、ブロックチェーン業界で必要とされる背景と発展に寄与する理由、そして、Plasm Netowrk を通して実現する社会についての概要を記載します. さらに、Plasm Netowrk を達成するためのプロダクトについての設計概要を記載します.

1 Introduction

私達はブロックチェーンを活用することで Web3.0 の実現を目指しています. 既存の社会では権力者による情報や富の独占と、自分達に有利なルールを作ってきた歴史があります [1]. また、公平な仕組みを謳っていてもそこに透明性が無く信頼に依存して成り立っている仕組みと言えるでしょう. そのような既存社会に対してブロックチェーンは、分権的なガバナンスを用いることでプラットフォーマーが管理しない透明性と公平性のある Trustless なシステムを実現します. 何故なら、ブロックチェーンは誰もが閲覧、検証、運用可能なシステム上でフォールトトレラント性と高い改ざん耐性を実現できるシステムだからです [2].

ブロックチェーンのプロトコルを作るとして実際にユーザーにその恩恵を届けるためにはアプリケーションが必要になります. アプリケーションとは一種の OS とユーザーを繋ぐインターフェイスであるからです. ブロックチェーン上のアプリケーションを一般に Decentalized application(Dapps) と言います. 現在、様々なブロックチェーンでスマートコントラクトやチェーンコードといった形で Dapps が開発、デプロイされてユーザーに提供されています. しかしながら、ブロックチェーンの分散冗長化された仕組み上 Dapps の処理性能は決して高いものではありません. 最も規模の大きいスマートコントラクトを搭載したブロックチェーンである Ethereum[3] のトランザクションスループットは秒間 15 トランザクションです [4]. 一方で世界中に多くのユーザーを保有する VISA や Alipay ではそれぞれ秒間 1700 トランザクション [5]、256,000 トランザクションを処理しています [6]. 多くのユーザーが Dapps の恩恵を受けるためには現状の処理性能があまりに不十分であることが分かります. そこでブロックチェーンでは様々なスケーリングソリューションが考え出されました.

ブロックチェーンのスケーリングソリューションはいくつか存在します.

1. **SegWit** : Segwit はブロックサイズを圧縮する手法です [7].
2. **State channel** : State channel はユーザー同士がオフチェーンでいくつかの取引をまとめて行い最終状態のみをブロックチェーンに記述する手法です [8].
3. **Sharding :** シャーディングは複数のノードでトランザクションの分散処理を行います [9].

4. **Plasma**： Plasma は別のチェーンにトランザクション処理を行わせてルートハッシュのみをメインチェーンに保存します [10].

私達はその中でもまず、メインチェーンの外でトランザクションを処理するレイヤー 2 ソリューションに焦点を当てました. レイヤー 1 は Ethereum や Bitcoin と言ったパブリックブロックチェーンのことを指します. それらはトランザクションが飽和している問題を抱えています [11]. このことから、10 年後のブロックチェーンの使用法は今までとは大きく異なり、レイヤー 1 がトラストレイヤー、レイヤー 2 がトランザクションレイヤーとして使用されることになると予想されます.

私達がその中でも初めに Plasma に焦点を置いた理由は、Plasma がメインチェーンの処理性能に最も依存しないスケーリングソリューションだからです. Plasma では単一の Aggregator と呼ばれる運営者がサイドチェーンの運営を行います. つまり、合意形成プロセスの不要な中央集権的管理方法で多くのトランザクションを処理することができます. それは、既存の中央集権的システムで使われているスケーリングソリューションをそのまま転用できることを意味するため分散台帳では不可能な高い処理性能を実現することができるのです. Plasma のアプローチは全ての分散台帳に飛躍的な処理性能の向上を行えるためこの先、必要不可欠な技術になっていくと言えるでしょう.

Plasma が高いスケーラビリティを誇ることは分かりました. しかしながら、Plasma の運用には未だいくつかの問題を抱えています. 一つは Plamsa を使ったアプリケーション (Plapps) で出来ることに制限がある点です. Plasma で出来ることは一階述語論理 (以下、Predicate) で記述可能であることが明らかになりました [12]. もう一つは Plasma は複数のコンポーネントにより構成されている複雑なシステムであるという点です [14]. 単純にスマートコントラクトを記述しデプロイしただけでは Plapps を構築することは出来ません. 具体的には Plapps は親チェーンコントラクト、子チェーン、Aggregator、ユーザーの 4 つのコンポーネントから成り立っておりそれぞれについて処理を記述しなければならなりません. 私達はこれらの問題を Plasm Network を通して解決していきます. **Plasm Network** では Predicate を正しく記述できるように標準規格を設けてライブラリ化します. そして最終的に複数のコンポーネントを簡単にデプロイできるためのクラウドサービスを構築します. Plasm Network が提供するプロダクトを通して開発者が Plapps を開発する過程を手助けします.

また、Plasma を扱う上で私達は OVM[13] という技術を用います. これは、前述した Plasma を記述するために使用するプロトコルです. さらに、OVM は Plasma だけでなくすべての Layer2 Solution を抽象化した存在です. Layer2 Solution には前述した State channel や、Plasma の無限のスケーラビリティとトレードオフに取引履歴の可用性に関する問題を解決した Optimistic Rollup などがあります. OVM を用いることでこれらのような Plasma 以外の Layer2 Solution も扱えるようにする発展性を持ち合わせることが出来ます.

私達はこれらのシステムを Polkadot[15] を軸に展開しようと考えています. Polkadot は複数の異なるブロックチェーンを束ねるプロトコルです. Polkadot によって今まで独立だった複数のブロックチェーンが透明で分権的なガバンスのもと相互運用性を得ることができます. さらに、Substrate[16] と呼ばれるブロックチェーンを作るためのフレームワークがあります. Polkadot が束ねるチェーンや Polkadot 自体も Substrate を使って作られます. 一つの理想的なブロックチェーンですべての用途を賄うことは現実的ではありません. ブロックチェーンを使う理由はユーザーごとに多様であり、すべてのニーズを満たすガバナンスの構築ないし可用性を維持することが難しいからです. それ故に、今まで 900 を超えるパブリックブロックチェーンが作られてきました. Polkadot と Substrate は人々が用途に応じてブロックチェーンを作っていく時代にさらに拍車を掛けていくでしょう. 私達はこの大きな流れに賭けることにしました. 当然、Polkadot においても

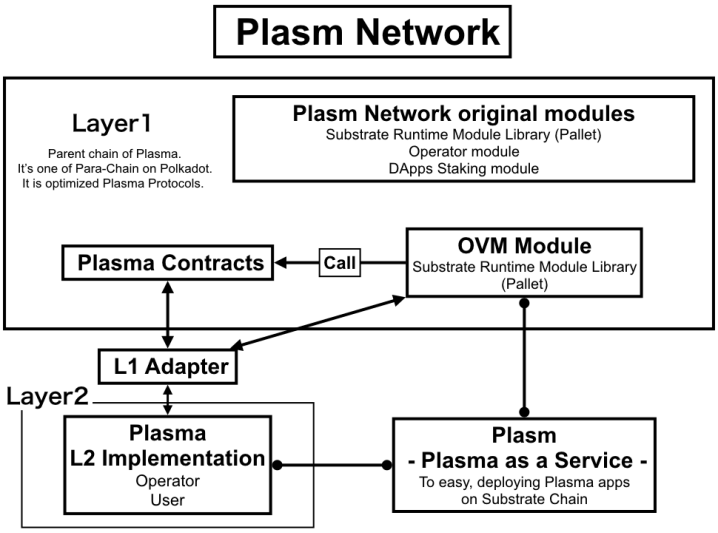

図 1　Plasm Network architecture.

Plasma は必要不可欠な役割です．故に私達は Plasma を展開する先に Polkadot × Substrate を選択しました．

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 はすべての開発者にスケーラブルな分散アプリケーションの開発メソッドを提供するためのプロジェクトです．私達はこれを上記のアーキテクチャにより実現します．

Plasm Network は Substrate を用いて構築された Layer1 のパブリックブロックチェーンです．その上にスケーラブルな Dapps を乗せるための独自の機能と、OVM のモジュール、そして Plasma アプリケーションに必要なスマートコントラクトの標準実装が搭載されています．また、これは Plasma Defauct Standard Chain として機能する Parachain になることを想定しています．一般に Plasma アプリケーションをデプロイする時のデフォルトチェーンとして Plasma Network を選びます．Plasm Network はランタイムに独自の機能を搭載することで Plasma アプリケーションに新たな可能性と快適なユーザビリティを提供します．また、Plasm Network では革新的なトークン発行アルゴリズムを用いて User/Developer ファーストなトークンエコノミーを設計しています．

それをサポートする形で幾つかの Plasma の ブロックチェーンに乗らない 実装と Plasma as a Service が展開されます．Plasma as a Service は Plasma アプリケーションを簡単にデプロイするための Platform as a Service です．メインチェーンを選んで GUI を通してカスタマイズした子チェーンをデプロイ、管理する機能が提供されています．これらは Cryptoecnomics Lab[19] が提供する Plasma Rust Frameworks [**?**] を利用して実装される予定です．

次の各セクションでそれぞれのサービスについての概要と技術詳細を解説していきます．

3 Substrate

Substrate はブロックチェーンを開発するためのフレームワークです．これは Parity Technologies と Web3 Foundation らによって開発されています．Web サイトを作るための Web フレームワークとして Ruby on Rails や Django などがあるように、ブロックチェーンを作るためのフレームワークとして Substrate があります．Subtrate はブロックチェーンを構築するために必要なコアの機能が予め用意しておくことで、ブロックチェーンをゼロからフルスクラッチすることが大変な作業である問題を解決します．Substrate のコア機能にはデータベース、P2P 通信、合意形成アルゴリズム、トランザクションプール、WebAssembly の Executor などブロックチェーンを動かす上で欠かせない様々な要素があります．また、Rust 言語で書かれており、高速、メモリ安全、並列処理、WASM へのコンパイルが容易といった特徴を持ちます．さらに、ブロックチェーンのアップデート時にハードフォークが起こらない、開発目的に応じてプライベート、コンソーシアム、パブリック全てのブロックチェーンを作成可能、将来的には Polkadot との連携により異なるブロックチェーンと相互運用とセキュリティのシェアが可能などのメリットがあります．

4 Polkadot

Polkadot は複数のブロックチェーンを束ねるためのブロックチェーンです．これは **Relaychain** とも呼ばれています．また、Polakdot は前述した Substrate によって作られます．Polkadot の大きな特徴の一つとしてインターオペラビリティとセキュリティの共有があります．これは Polakdot が束ねているチェーンがそれぞれ繋がることができる仕組みとなっています．そして、束ねられているチェーンを **Parachain** と呼びます．Parachain は Relay Chain に接続される異なる独立したブロックチェーンであり、Parachain 同士はインターオペラビリティを得ることができます．また、Parachain は Relaychain からバリデータを借りることができその Parachain 自信に十分なバリデータがいなくても Relaychain のセキュリティを担保することができるようになります．

5 Plasma

Plasma はブロックチェーンにおけるスケールソリューションの一つです．Plasma の基本的なアイディアは、チェーン外でトランザクションをマークルツリーで管理・高速に処理しマークルルートのみをブロックチェーンに刻むというものです．チェーン外の処理を行いブロックチェーンにハッシュを提出する責任者を Plasma の文脈で Aggregator と呼びます．

今回、Plasm でサポートする ”Plasma” は Plasma-Cash をベースにしたものです．これはマークルツリーの葉でトランザクションではなく 1 つの NFT の状態を持ちます．状態の遷移を行うためのルールは後述する OVM によって定義できます．図 1 では状態として所有権を持つ NFT の状態遷移とそれに必要な Transaction の例を示します．

図 3 では状態遷移の説明をしています．状態遷移をするためには 1. ”Owner” の署名があること 2. 新たな状態を output に指定していること 3. 状態がすでに別の方法で遷移されていないこと が必要です．これを OVM を用いて記述します．ここで記述された論理を ”Predicate” と呼びます．これは一階述語論理で記述されます．OVM が受理した Transaction を受け取ったとき、状態を遷移しマークルルートを更新します．

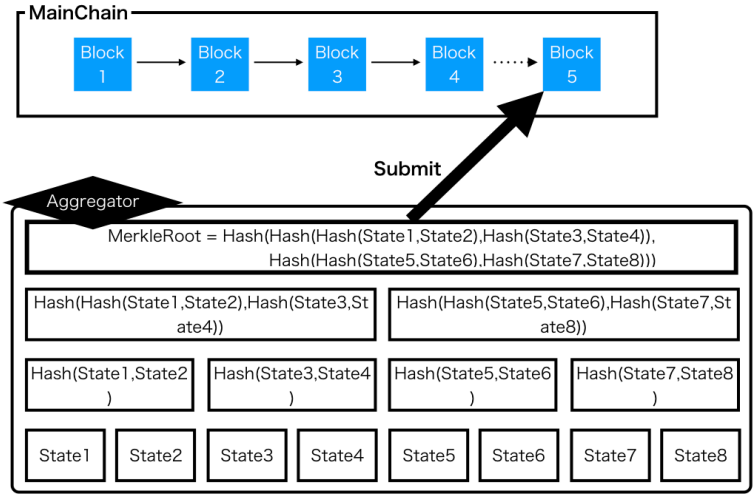

図 2　Plasma state.

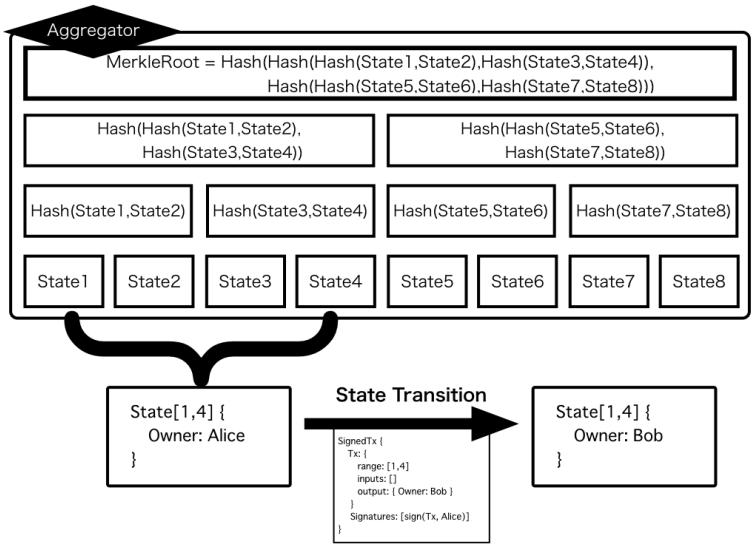

図 3　Plasma state transition.

Plasma において 単一の Aggregator がこれらのトランザクションの処理とマークルルートの提出を行います. 仮に Aggregator が不正を働いた際にユーザの提出したトランザクションが改ざんされる可能性があります. Plasma ではそのような改ざんに対して先述した OVM と Predicate を用いてメインチェーン上でトランザクション、状態の正しさを紛争することができます. これにより Plasma は単一の Aggregator による高速なトランザクション処理能力とブロックチェーンの持つ強固なセキュリティの両方を兼ね備えることができます.

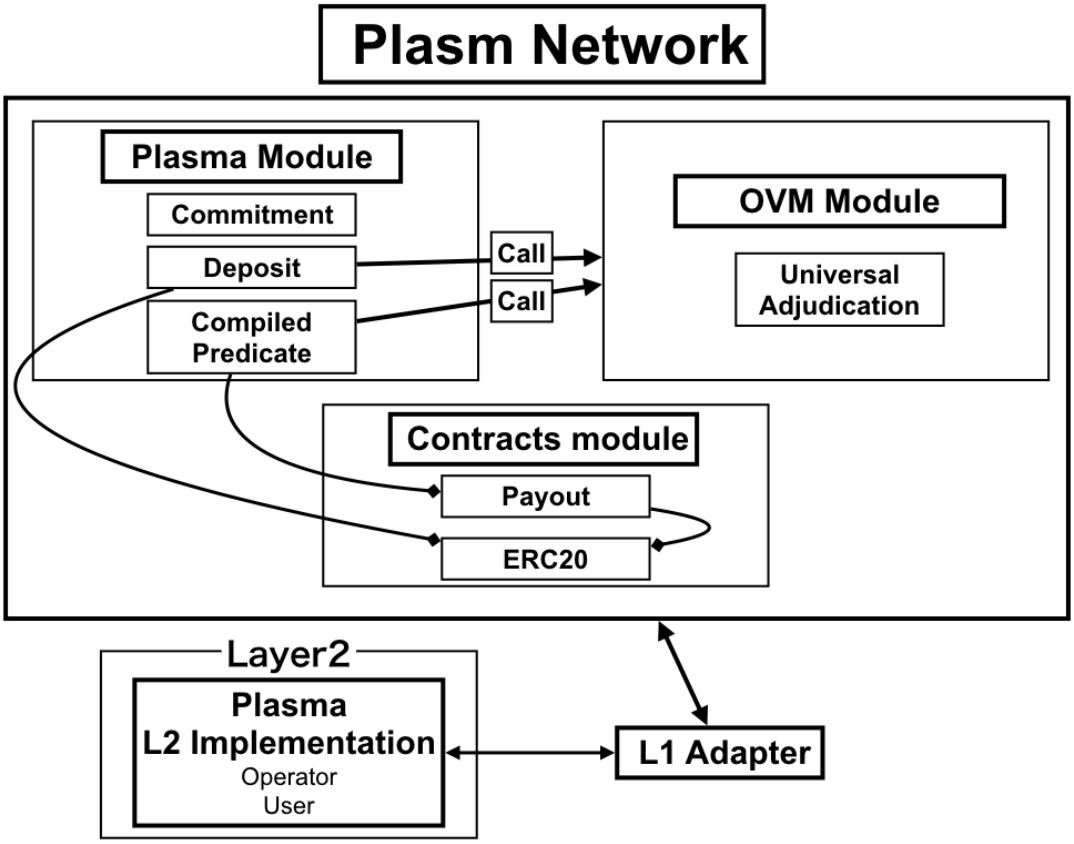

図 4　OVM modules.

6　OVM

OVM はブロックチェーンにおける Layer2 アプリケーションのロジックを統一的に記述するためのプロトコルです. OVM によって記述された内容はそれぞれアプリケーションロジックとして Layer1/2 向けのコードをとしてそれぞれコンパイルされます. Plasm Network では OVM をスマートコントラクトと切り離してモジュールとして用意することでより簡潔で便利に OVM を利用できるようにします.

Plasm Network 内では OVM とその周囲のアーキテクチャは図 4 のようになっています.

専用のクライアントアプリケーションである L1 adapter を介すことで適切に Plasma applications(Plapps) を作成、動作させることが出来ます. Plapps は Plasm Network において OVM, Plaaps, Contracts module によって構成されます. Ethereum における Plasma 開発では、これらのモジュールで提供されているものはすべてスマートコントラクトで管理されていました. しかし、その場合だと複雑なロジックを内包している Plasma アプリケーションを実行する際のガスコストを予測しづらいという問題が発生します. また、複数のコントラクトを組み合わせたアプリケーション構築は開発者に度々混乱を招くことが予想されます. そのため、Plasm Network では 3 つのモジュールに役割を分離することで表面的に簡潔且つわかりやすい構成を考えました. OVM Module ではユーザが Layer1 に上がっている情報に過ちを見つけた際に紛争を起こすための Universal Adjudication と呼ばれる機能が実装されています. Plasma Module は幾つかの Plasma のた

めに必要不可欠なスマートコントラクトの共通実装をサポートしています. そして、各アプリケーションごとに異なるロジックが必要な実装のみを Contracts Module で管理しています.

これらの Plasm Network のロジックは前述した Plasma L2 Implementation によって提供される実装と組み合わせることでアプリケーションを構築することができます.

7 Lockdrop

Lockdrop[21] は機会費用を担保にトークンを発行する仕組みであり多くのユーザが低リスクでトークンを受け取ることができる革新的なトークン発行メカニズムです. Plasm Network ではトークン発行の仕組みとして Lockdrop を利用します. ここでは Plasm Network のトークン発行システムについて解説します. これは Edgeware がトークン発行の仕組みとして考え出した Lockdrop を拡張したものになります. Plasm Network におけるトークンは PLM と表記し "ぷらむ" と呼称します. PLM は小数点以下第 15 位までを扱いそれ以下を切り捨てで計算します. トークンの役割に関する詳細は PLM Token Ecnomics の章 16 にて解説します.

7.1 Lockdrop overview

私達は第一回目の Lockdrop を Ethereum の機会費用を担保に行います. そのため説明では Ethereum による Lockdrop を前提に行います. しかしながら、Lockdrop は TimeLock 機能のあるあらゆるチェーンで代替可能なアルゴリズムであることに注意してください. 図 5 は Plasm Network における Lockdrop の一例を表しています.

Lockdrop は以下の手順で行います.

1. Ethereum のトークンホルダーは Ethereum 上の LockContract に所持している ETH を期間を指定して Lock します.
2. Plasm Network の Genesis Block は、Lockdrop を行ったトークンホルダーに対してそれぞれ Lock された ETH の 総量 × Lock 期間に応じたボーナス $\times\alpha$ だけ PlasmToken を付与します.
3. Plasm Network の Genesis Block は、コミュニティが 初期発行されるトークンの総量 × 15
4. Lockdrop を行ったトークンホルダーは Lock 期間を過ぎると Lock した ETH が使用できるようになります.

Ethereum のトークンホルダーは Lock した Ethereum の量と期間分だけ機会費用を払っています. PlasmToken はこの機会費用を担保にしてトークンを発行することでトークンの価値を担保します. また、Plasm Network の最終的なトークンの発行量は決まっていません. これは Genesis ブロック以降にも Lockdrop によるトークン配布をより公平に行うためです. また、ここで総発行トークンのうち 15% は私達が保有する開発費用です. 具体的には以下のような方法で複数のトークン配布を行います.

8 Multi Lockdrop

Multi-Lockdrop は前述した Lockdrop を複数回行うための仕組みです. Plasm Network では 3 度の Lockdrop を行います. Plasm Network ではトークンの総発行量がジェネシスブロックで決まりません. 3 度

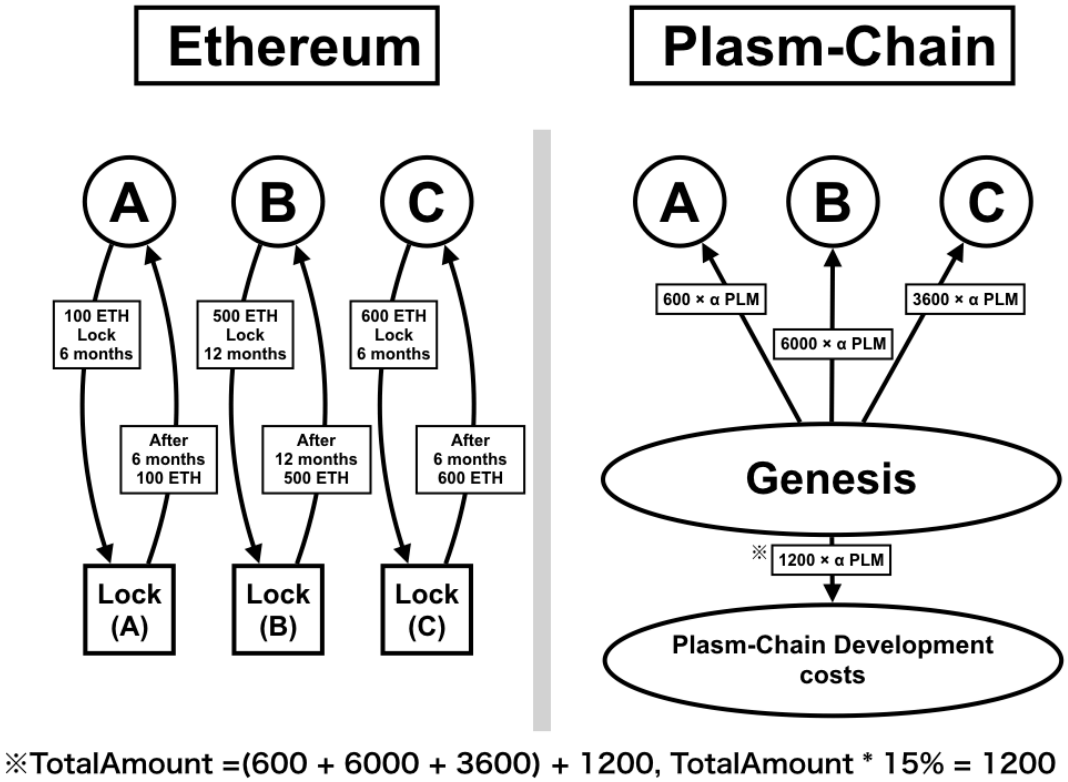

図 5　Lockdrop overview

の Lockdrop によりその都度トークンが発行され、後述する ”Staking” の利回りとしてもトークンが発行され続けます.

Lockdrop を複数回に分ける理由は 2 つあります. 1 つ目は先行者利益が肥大化することを避けるためです. 1 度の Lockdrop で全てのトークンを分配すると、仮に最初の Lockdrop の参加者が少なかった場合にトークンの大半を保有するホルダーが現れるかもしれません. そして、そのようなことが発生した後にロールバックを行うことはチェーンの信頼性を損ないます. ブロックチェーンでは予め提示したルールに基づいてこのような問題を避ける施策を打つ必要があります. この問題を解決するために私達はジェネシスブロックでトークンの総発行量を決定しないアルゴリズムを考案しました. 2 つ目は Plasm Network が安全にスケール/分権化するために序盤はある程度管理可能で実験的な振る舞いが出来るようにするためです. Blockchain の強固なセキュリティシステムは大量のノード参加者と広く分散化されたトークンホルダーが存在することで成り立っています. 起動直後からこのセキュリティを担保することは極めて難しいです. 序盤は信頼できる機関がある程度管理可能な状態であるほうが好ましいでしょう. そこで 3 度の Lockdrop を設けることにより、全てのトークン発行前に不用意なトークン価値の高騰を抑えることで管理コストを下げます. 私達は最終的に Plasm Network を完全なパブリックチェーンとすることを前提として安定して動作するまでの準備期間を設けるために複数回の Lockdrop を行います.

また、Plasm Network では 1-st, 2-nd, 3-rd Lockdrop についてそれぞれ以下のトークンで Lockdrop が可能になります.

- 1st: ETH
- 2nd: ETH, BTC
- 3rd: ETH, BTC
- Polkadot parachain オークション: DOT(これは特別な Lockdrop です. 詳しくは章**??**を参照してください.)

8.1 Defenitions

初めに最初の Lockdrop で配布される PLM の総量 ($TotalPLM^{genesis}$) を以下のように定義します.

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

これらを発行比率 ($IssueRatio$) に応じて Lockdrop に参加したユーザに対して分配します. $IssueRatio$ は Lock したトークンの総量 ($Locked_{token}$) を PLM トークン発行時のドルと Lock したトークンとの変換レート ($DollarRate_{token}$)、そして日数に 1.0005 を日数 ($Days$) 乗したものを掛けたものに近似されます. ここで 1.0005 は Polkadot の金利を参考にしました. Polkadot のデフォルト最大平均年利は 20% と定義されています. [23] これを複利込みの日利に直した際の近似解が 0.05% となるからです.

ユーザ実際には表 8.1 のように Lock する期間を次の 4+1 種類から選ぶことができます. Lock 期間に応じて Lock したトークンの価値をドル換算した値に以下の $LockBonus$ を掛けたものが $IssueRatio$ となります.

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

表 1　Lockdrop bonus table.

※ については DOT での Lockdrop 時にのみ使用可能なオプションです. また、DOT の Lockdrop は特殊であり約 2 年間の Lockdorp にしか対応していません. これに関しては **Polkadot auctions Lockdrop** の章で後述します.

前述の情報を元に $IssueRatio$ を以下のように定義します.

- $Locked_{token}$ は Lockdrop の対象の token を Lock した量です.
- $DolalrRate_{token}$ は $1token$ のドル建て価格です.
- $LockBonus_{days}$ は $days$ 日間 Lock したときの $LockBonus$ です.

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

算出された $IssueRatio$ を元に、配られるトークン数が決定します. トークン配布量は以下のように定義されます.

- n は Lockdrop を行ったユーザ数です.
- $IssueRatio_i$ は ユーザ$_i$ の $IssueRatio$ です.
- PLM_i は ユーザ$_i$ の 得られる PLM の量です. 総発行トークンのうち 15%(17/20) がコミュニティが保有します.

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

つまり、全体の $IssueRatio$ のうち自分の $IssueRatio$ が占める割合だけ PLM が分配されます. この時、開発費用として 3/20 である 75,000,000 PLM が使われます. ここで、$IssueRation$ の総和である $TotalIssueRatio$ を定義します.

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

また、1 回目の Lockdrop における単位 $IssueRatio$ あたりの PLM 発行量をここで α_1 とおきます. これは 2 回目以降の Lockdrop における PLM 発行量を決めるための重要な値となります.

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

2 回目、3 回目の Lockdrop における単位 $issueRatio$ あたりの PLM 発行数を α_2, α_3 と以下の等式を満たすように定義します.

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

上記から 2 回目,3 回目の Lockdrop における ユーザ$_i$ に配られる PLM の量は以下になります.

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

こうすることで、2 回目以降の Lockdrop のおいてユーザは単に $IssueRatio$ に比例した量のトークンを得ることができます. これにより、2 回目以降に Lockdrop を行うユーザが非常に増えた場合、ユーザが取得できる PLM の量が全体の割合に対して過度に少なくなってしまう問題を解決します.

複数回の Lockdrop におけるトークン配布量がどのように変わるかの具体例を以下の図 6 で示します. ここで $DollarRate$ は一定とします.

8.2 なぜ Lockdrop によるトークン発行を行いますか?

- 私達は担保となるユーザの資産を保有しません.
- 低コストでユーザは Lockdrop に参加できます. スマートコントラクトを動作できるすべてのユーザが Lockdrop に参加可能です. すべてのトークン保有者に参加機会があります.
- ユーザは詐欺師による資産の強奪リスクを考慮する必要がありません. 機会費用を担保に PLM を発行します. ユーザが Lock した資産は Lock 期間が過ぎた後に帰ってきます.

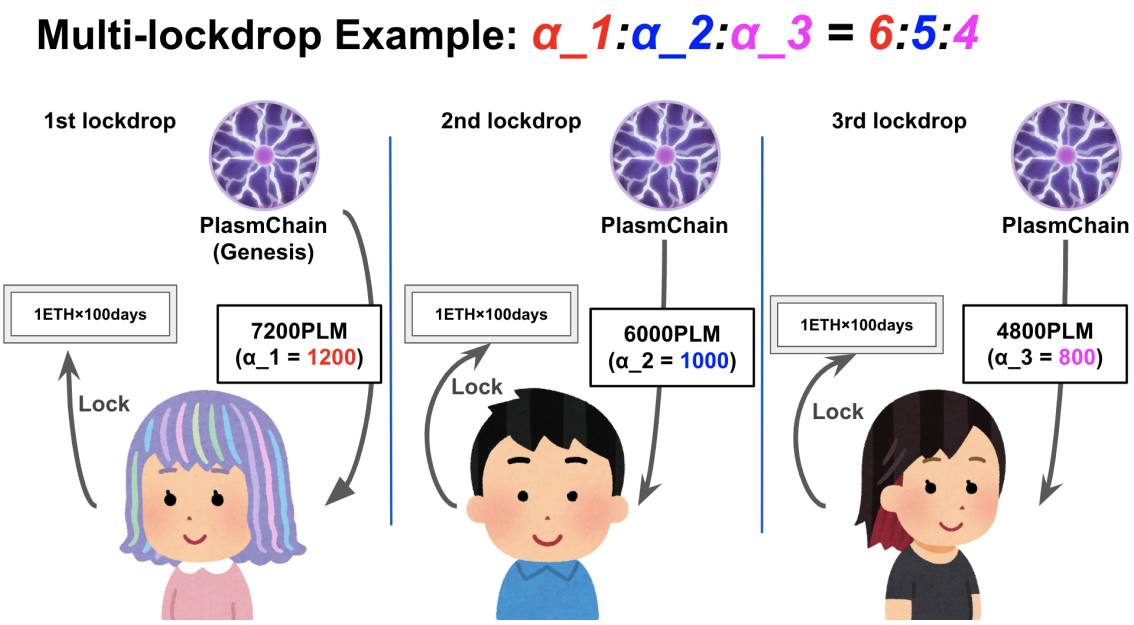

図 6　Multi Lockdrop example.

- Airdrop とは異なり Lockdrop の参加者はコストを支払って PLM を取得します．ゼロでない価値のトークンを発行する事ができます．

8.3　なぜ Lockdrop を複数回に分けますか?

- 初回の Lockdrop ですべての PLM を発行した場合、少数のトークンホルダーが莫大な量の PLM を所有する可能性があります．そうした場合、健全なエコシステムが動作しない危険性があります．過剰な先行者利益を防ぎます．
- 複数回の Lockdrop を行うよりトークンの取得機会を増やすことができます．より大勢に PLM を取得できる機会を設けます．

9　Real time Lockdrop

Real-Time Lockdrop は章 8 で説明した Multi-Lockdrop における 2-nd, 3-rd Lockdrop を行うための仕組みです．1-st Lockdrop では Lockdrop が有効な期間のあと、ジェネシスブロックで一斉にトークンを付与します．Real-Time Lockdrop では Lockdrop 期間中に Lock を行うと即座に PLM トークンを入手することが出来ます．詳細についてはこちらをご覧ください．

10　Polkadot acutions Lockdrop

Polkadot auctions Lockdrop は前章で説明した Lockdrop のうち、DOT を用いた Lockdrop を指します．DOT による Lockdrop は 1-st, 2-nd, 3-rd Lockdrop とは独立しており、特殊な役割を持っています．

Polkadot における DOT には幾つかの役割があります. そのうちの重要な役割として Staking と Parachain deposit があります. Staking はチェーンのセキュリティを担保するための PoS コンセンサス・アルゴリズムに使います. Parachain deposit は Polkadot のあるブロックチェーンを Parachain に入るために一定期間 DOT を預け入れます. Parachain とは、Polkadot と繋がるブロックチェーンのことを意味します. Parachain になることで Polkadot の持つバリデータを借りることができセキュリティを共有することができます. また ICMP[24] を用いて他の Parachain とのインターオペラビリティを得ることができます.

DOT の Lockdrop は Parachain deposit を利用します. Plasm Network を Parachain に入れるための資金として Lock された DOT は使用されます. DOT の Lock 期間は約 2 年であり、その間は Plasm Network は Parachain として活動します. 留意点として、Parachain に入るかどうかの決定はオークションで行うため、この Lockdrop は失敗する可能性があります. もし失敗した際は DOT は Lock されず返却され、PLM は得られません. 成功した場合は DOT は Lock され、PLM トークンを取得することができます. 図 7 は Polkadot における Lockdrop の手順を示します.

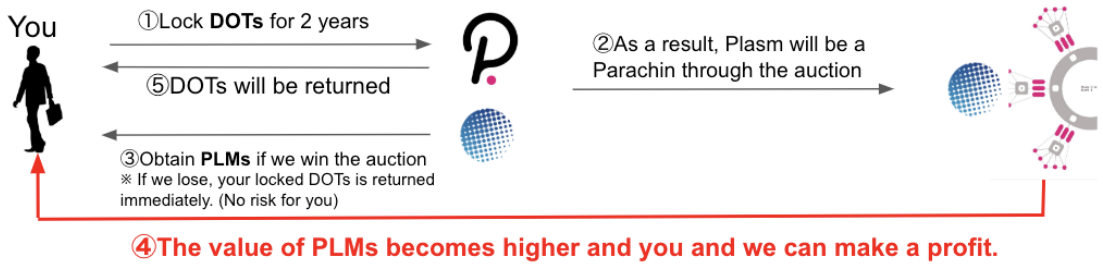

図 7　Polkadot auctions Lockdrop.

このシステムには crowdfund module[25] を使用します. また、実際に DOT が Lock される期間とは別にオークション中の期間も DOT が使用できなくなることに注意してください. このように他の Lockdrop と比較して一定のリスクを伴うため DOT による Lockdrop の LockBonus は最も高くなっています. (Multi-lockdrop の章 8 を参照)

Plasm Network の Lockdrop コスト設計は DOT Lockdrop のコストをベースに作られています. DOT Lockdrop とトレードオフの関係である DOT Staking とのベンチマークを取ることで Lockdrop のコストパフォーマンスを算出することが出来ます.

11　Lockdrop Affiliate Program

Lockdrop Affiliate Program は Plasm Network の認知拡大のために行うプログラムであり、参加者は Lockdrop の際にもらえる PLM トークンを増やすことができます. ここでは 1-st Lockdrop におけるアフィリエイトプログラムについて説明します. アフィリエイトプログラムは以下のルールに従います.

- Lockdrop ユーザは紹介者の Ethereum Address を指定することができます. (紹介者を指定しないこともできます.)
- 有効な紹介者は、紹介した Lockdrop ユーザが貰えるトークンの 1% を追加で獲得できます.
- Lockdrop ユーザが有効な紹介者を指定したとき、そのユーザが貰える PLM トークンを追加で 1% 獲得できます.

この際に追加で発行するボーナストークンは、コミュニティが保有する 15% のトークンから支払われます. 有効な紹介者になるための方法は Plasm Network の Discord にて配信します. また、2-nd, 3-rd Lockdrop では 1-st Lockdrop の参加者が有効な紹介者になります. これについての情報は随時アップロードします.

12 Consensus Algorithm

Plasm Network はセキュリティの維持のために段階的に合意形成アルゴリズムと報酬設計を変更します. 具体的には最初はコミュニティによって選定された Validator のみで運営する Proof of Authority の形式をとります. 次に、Parachain として参加するためにコレイター [?] を中心とした報酬設に変更します. 最終的に Polkadot の Relaychain でも採用されている NPoS に移行します. 具体的には報酬の配布方法は PLM Token Ecosystem の章 16 を参照してください.

12.1 Proof of Authority

Proof of Authority はコミュニティによって選定された Validator のみで運営する合意形成アルゴリズムです. パブリックブロックチェーンはローンチ直後に信頼できるバリデータを十分用意できず脆弱性を持つ可能性があります. そのため、十分なトークンホルダーの分布と潜在的バリデータの存在が確認できるまでは PoA による運営を行います. この時のバリデータにも PoS の時と同様に指定されたパラメータに従って報酬が支払われます.

12.2 Incentives of Collator

Incentives of Collator は Parachain における Collator を運用する報酬設計を与えるものです. Plasm Network は Polakdot の Parachain になることが期待されています. Parachain ではトランザクションを収集しブロックの中のトランザクションを監視し、Relaychain のバリデータにブロックの証明書をに送る Collator が必要です. Collator はフルノードで無くてはいけないため、Collator として Plasm Network に参加するインセンティブが必要です. ここで、Plasm Network の Collator は初め前述の PoA のようにコミュニティによって選定されます. その後、後述する NPoS によって選出されるように変更されます. つまり、バリデータの役割をしていたノードが Plasm Network が Parachain となっている間は Collator として機能します.

12.3 Nominated Proof of Staking

Plasm Network は最終的に Polkadot の RelayChain で採用されているコンセンサスアルゴリズムを使います. このアルゴリズムは大きく 3 ステップに分かれています.

1. NPoS: Nominator は Validator を選出する. [27].
2. BABE: Validator はトランザクションを検証しブロックを生成する [28].
3. GRANDPA: 配布されたブロックをファイナライズする GRANDPA[29].

初期の Validator ノードを限定することで十分な Validator がいない状態において悪意のある Validator が闊歩するリスクを減らすことができます. ブロックの生成報酬は Validator とその支援者である Nominator

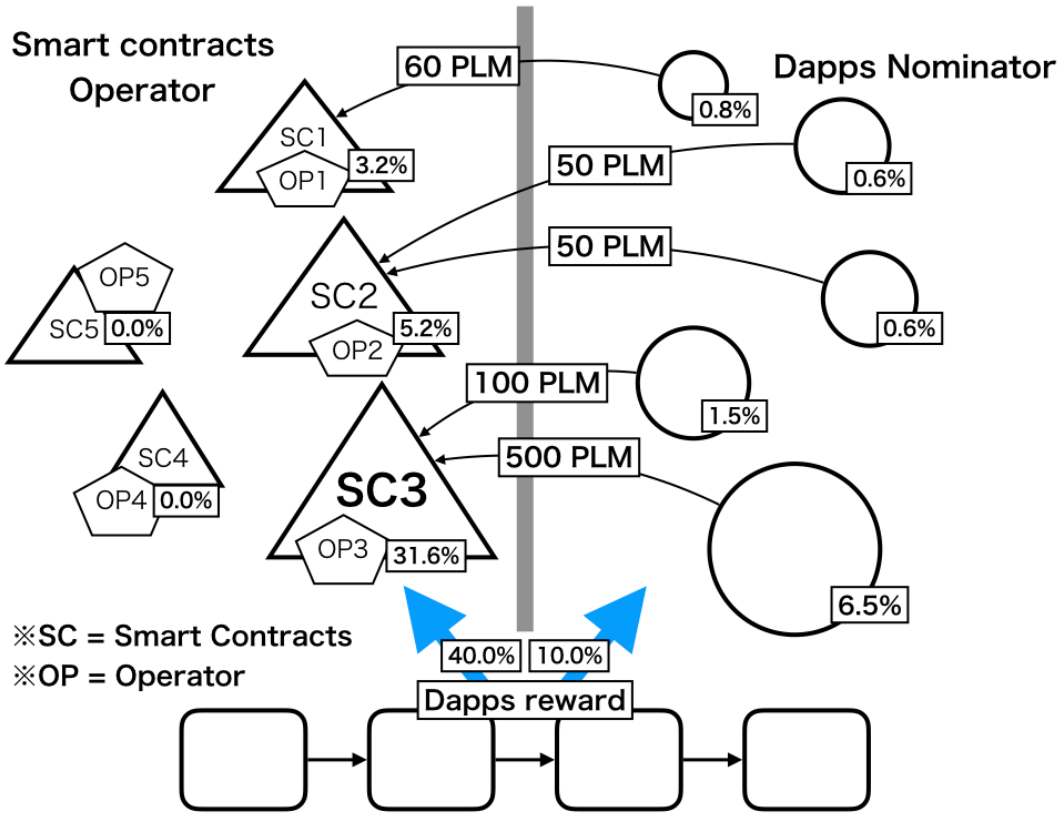

図 8　Dapps Rewrads

に分配されます．また、Plasm Network の価値向上の貢献者にもこの報酬は支払われます．

13　Dapps Rewards

Dapps Rewards はスマートコントラクトの開発者または管理者に継続的な報酬を与える仕組みです．図 8 のように Plasm Network の Staking 報酬の 50% は Plasm Network の価値を向上させたアプリケーション製作者に当てられます．Plasm Network ではスマートコントラクトに対してスマートコントラクトの管理者を割り当てることができ、この管理者のことを ”Operator” と呼びます．ユーザはスマートコントラクトに対しても Staking をすることができます．この行為を Nominate と呼び、それを行う人を Dapps Nominator と呼びます．下図のように、多くの Nominate を受けたスマートコントラクトの Operator は新たに発行された PLM トークンをチェーンから受け取ることができます．

この報酬を Operator と Nominator にそれぞれどのように分配するかを定義していきます．次の変数を定義します．

- $Rewards_{nominate}$：Nominator に割り振られた報酬の合計．
- $Rewards_{contract}$：スマートコントラクトに割り振られた報酬の合計．
- $Rewards_{nominate_{i,j}}$：i 番目のスマートコントラクトに対する j 番目の Nominate で得られる報酬．

- $Rewards_{contract_i}$：i 番目のスマートコントラクトの Operator が得られる報酬.
- n：スマートコントラクトの数.
- m_i：i 番目のに対する Nominate の数.
- $stake_{i,j}$：i 番目のスマートコントラクトに対する j 番目の Nominate により Stake したトークンの量.

この時、この Stake について $Nominate_{i,j}$ により以下の報酬が得られます.

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

Nominator は選んだスマートコントラクトに関わらずスマートコントラクトに対する Stake 総量のうち自信の Stake 量が占める割合に比例した報酬を得ることができます. Stake を受けた $contract_i$ の Operator は以下の報酬が得られます.

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

一方、Operator はスマートコントラクトに対する Stake 総量のうち自信が所有するスマートコントラクトの Stake 量が占める割合に比例した報酬を得ることができます. これにより、Nominator は単純にトークンの価値を向上させると思われるスマートコントラクトに対して Stake するインセンティブが発生します. また、Operator は自信が管理するスマートコントラクトに Stake を受けることで半永続的に報酬を受取ることができます. これはチェーン上のアプリケーション開発者 (管理者) のマネタイズが難しい問題に対する革新的な解決策となることを期待しています. ここで、この Operator/Nominator が報酬を実際に受取るには一定の期間を待つ必要があります.

しかしながら、このシステムは以下の悪いケースを考えることが出来ます.

- **Malicious sock puppet**
  - 価値のない Operator が自作自演で報酬を増やそうとする問題です.
- **A few popular operators are staked from almost nominators**
  - 著名で安定したスマートコントラクトのみに stake が集中する問題です.

それぞれについての対策を以下の記します.

14 Malicious sock puppet

悪意のある Operator または Nominator が価値のないスマートコントラクトに Stake して報酬を得る問題です.

これは図 9 のように、一人一票の Good/Bad Voting システムを導入することで解決します. この投票は、トークンを Staking しているすべてのユーザが各スマートコントラクトごとに一票のみ行うことができます. Operator に対する投票に応じて以下のような罰則を受けます.

- Good の数が Bad の数の 4 倍以上且つそのスマートコントラクトが一定期間以上 Running している場合、そのスマートコントラクトの運用は健全であるとみなし Operator は正常に報酬を受け取ることができます. (i.e $Good \geq Bad \times 4$)

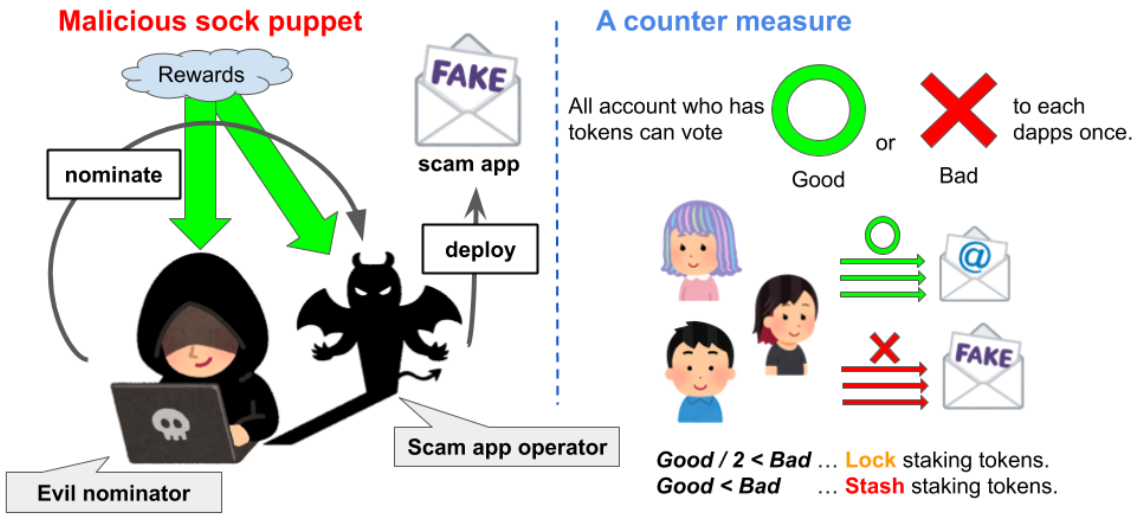

図 9　Malicious sock puppet and its a counter measure.

- Good の数が Bad の数の 4 倍未満またはスマートコントラクトが一定期間以上 Running していない場合、その Operator と Nominate は報酬を受け取ることができません. (i.e $Good < Bad \times 4$)
- **Good の数が Bad の数の 2 倍未満である場合、そのスマートコントラクトに Stake されているトークンはすべて Lock されます. (i.e** $Good < Bad \times 2$**)**
- **Good の数が Bad の数未満である場合、そのスマートコントラクトに Stake されているトークンはすべて Slash されます. (i.e** $Good < Bad$**)**

ここで、Voting するユーザがシビルアタックできることに注意してください. しかしながら、Voting をするにはトークンを保有し Staking する手間がかかります. また、Voting という行為はそれを行うユーザに利益をもたらしません. 故に、十分多くの善良なユーザがいれば、シビルアタックをして投票を独占するコストに対してインセンティブが無いためそのようなケースは問題無いといえます.

例えば、100 万トークンを使い 100 万人に分身して Good を押して自作自演をするよりも 100 万トークンを普通に Stake した方が合理的です. また、実際にはトランザクション手数料も発生するため 100 万人に分身するにはもっとトークンが必要になります.

14.1　A few popular operators are staked from almost nominators

上記のような Slash システムを導入した時、多くのユーザーが既に安定して稼働しているアプリケーションに対して Stake することになるでしょう. そのような問題に対する解決策を示します.

図 10 のように、スマートコントラクトへの Stake に先行者利益を付与します. スマートコントラクトに対する Stake には別途でボーナス報酬を得る権利 (オプション) を加えます. これは Stake による報酬を受け取った $r(\leq 0)$ 日後までの期間に以下の報酬を得る権利を行使することができます. ここで x^k はある時点 k での x を表します. まず、次の変数を導入します.

- $Rewards_{optoion^k_{i,j}}$: ある地点 k における i 番目のスマートコントラクトに対する j 番目の Nominate で得ることができる option を行使したときの利益.

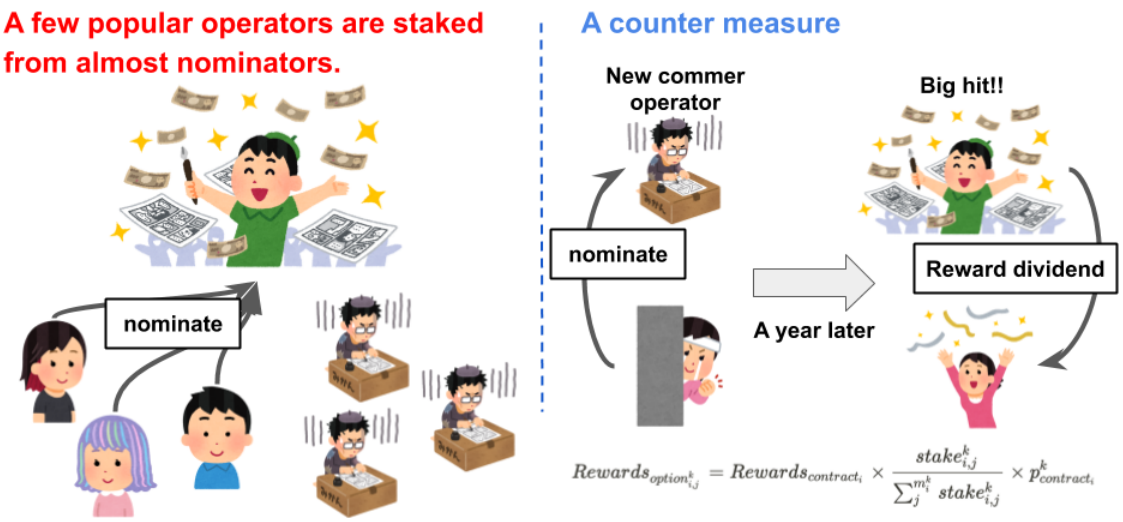

図 10　A few popular operators are staked from almost nominators and its a counter measure.

- $Rewards_{contract_i}$: option 権利行使時点での i 番目のスマートコントラクトの Operator が得られる報酬.
- $stake^k_{i,j}$: ある地点 k における i 番目のスマートコントラクトに対する j 番目の Nominate により Stake したトークンの量.
- m^k_i : ある地点 k における i 番目のスマートコントラクトに対する Nominate の数.
- $p^k_{contract_i}$: ある地点 k における i 番目のスマートコントラクトに Nominate した際に得られるオプション報酬を決定するための係数パラメータ、スマートコントラクトの Operator が定義することができる.

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

ここで $Rewards_{contract_i}$ と $Rewards_{option^k_i}$ が示す Operator が等しいことに注意してください. p は Operator が 0.2 以下の数値を指定することが出来ます.

この権利を行使した時、その時点での Operator の報酬の $p \times 100$

この権利を行使された際は Operator の貰える報酬がその分だけ減少します. 故に Operator は値 p をコントロールすることができます. r, p について Operator が最初に指定することが出来ます.

r, p の制約はこの機能を悪用して不当に Staking 報酬を大量に受け取るだけのアプリが現れるのを防ぐためです.

例えば、0 日目に $p = 0.1$, $r = 200$ の OperatorBob に 合計 100PLM が Stake されているとします. Alice はそのうちの 50 PLM を Stake していました. この時、Alice は r 日目までにその Operator の受け取る報酬の $0.5 \times 0.1 = 0.05$ 倍を受け取ることができる権利を得ます. 100 日目に Operator Bob が 1000PLM の報酬を受け取っていました. Alice は権利を行使して $1000 \times 0.05 = 50$ PLM を得る事ができました.

スマートコントラクトへの Staking は Validator への Staking よりもリスクが高いです. 機械的に Slash の判定がなされる Validator よりも、Voting による Slash が行われるスマートコントラクトのほうが予測がしづ

らいからです. さらに、報酬を得るまでにある程度時間もかかります. 故に上記のような仕組みを加えました. また、スマートコントラクトへの Stake は上級者向けであるため初めに記した $Validator : Operateor = 4 : 1$ の Staking 割合を理想としています.

15 Operator Trading

Operator Trading は Plasma アプリケーションの Operator を売買する仕組みです. 上記の Dapps Rewards の仕組みにより、Operator は恒常的に利益を得ることができます. よりよく Operator の運営を行えるならばその Operator を別の相手に譲ることも一つの案でしょう. 当然、Operator は常時売買されるわけではありません. Operator は自身に対してついた値に対して妥当だと思われる値をつけた相手に対してその権利を譲るでしょう. Operator の権利を譲り受けた対象は Operator としての報酬を受け取ることができます. この時、実際に運用を交替する必要はありません. しかし、Operator を売り渡した側は既にその Oeprator を誠実に運用するインセンティブを失うため必然的に譲渡先の新しい Operator の保有者が Operator の運用を行うことになるでしょう.

16 PLM Token Ecosystem

Plasm Network のトークンエコシステムは Polkadot のトークンエコシステム [23] を踏襲しています. 故に、幾つかの項目では、Polkadot の ecosystem で起用している式や値を転用しています. Plasm Network の Token は PLM と記述し「ぷらむ」と呼びます.

PLM は主に以下の 4 つを役割を担っています.

1. Staking for Consensus, バリデータとノミネータに当たられる報酬.
2. Transaction Fee, 有害な行動を阻止するために使用.
3. Block rewards for Dapps operator, アプリケーションのための持続可能な報酬設計.
4. Good/Bad Voting, Dapps Operator の評価 があります.

PLM は流動性トークンとして使用することを想定しています. 故に、非ゼロ価値の担保と多くの保有者を確保するために複数回の Lockdrop によるトークン発行を行います. 多くの PoS ベースの暗号通貨において Staking の割合は 5 割強を締めていることから、PLM トークンは以下の割合で運用されるように期待されます.

$$1 : 1 = Staking : Liquidity$$

16.1 Inflation Model

Plasm Network のトークンの新規発行を行う際の発行量と分配方法を決定するアルゴリズムを定めます. Plasm Network では全体の新規のトークン発行報酬を NPoS に対する報酬、Dapps Rewards に対する報酬で分け合う構成になっています. すなわち、Staking 行為は Validator に対する Staking(NPoS) とスマートコントラクトに対する Staking(Dapps Rewards) の 2 種類があります. それぞれの Staking によって得られる報酬はどちらも Staking 量に同程度に比例します. またバリデータ/スマートコントラクトに対して

Staking を行うユーザをまとめてノミネータと呼びます. バリデータに対する Staking とスマートコントラクトに対する Staking の割合は次の式を理想とします. ここで $Staking_{validators}$, $Staking_{contracts}$ はそれぞれバリデータに対する Staking, スマートコントラクトに対する Staking とします.

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

また、ここで Dapps Rewards における Operator に支払う報酬について考えます. Operator 報酬は Staking によるインフレ率に比例して増加します. Dapps Rewards の章より理想的な q を満たすときに報酬全体の 50% が Dapps Rewards の報酬に当てられます. またそのときにに Operator の得られる報酬は最大化します. 具体的な報酬の配分を示すために、以下の変数を導入します.

- $Rewards_{operators}$ は Operator が得る報酬の総量です.
- $Rewards_{stakers_{validators}}$ はバリデータに対する Staking によって得る報酬の総量です.
- $Reards_{stakers_{contracts}}$ はスマートコントラクトに対する Staking によって得る報酬の総量です.
- t は Operator が得る報酬の総量がスマートコントラクトに対する Staking によって得る報酬の何倍かを表す係数です.

Dapps Rewards の章**??**より $t = 4$ であり、理想的な q を満たすときの報酬全体の 50% が Dapps Rewards の報酬に当てられます. そのため、理想的な報酬の配分比率を以下のように定めます.

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

また、Staking の割合と報酬の割合は以下のように等しくなります.

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

PLM トークンは Polkadot と同じ NPoS を起用します. このノミネータとバリデータは Staking に対してある年利でトークン運用を行うことができます. また、PLM Dapps operator のノミネータと Operator にも同様にトークン報酬が支払われます. Plasm Network のインフレーションモデルは下記のように定義されます. まず、Polkadot のインフレーションモデルを踏襲して以下のような変数を定義します.

- x は全体の Staking 量をトークンの総発行量で割ったものです.
- X_{ideal} は x の理想的な値です.$Staking : Liquidity = 1:1$ なので,$X_ideal = 0.5$ です.
- q はバリデータへの Staking 量を全体の Staking 量で割った値です.

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

- Q_{ideal} は q の理想的な値です. 上記の式より、q の理想値は $Q_{ideal} = 5/6$ になります.
- $i(x, q)$ は Staker が得られる平均的な年利です. x, q を共に理想値に近づけるために、x, $|Q_{ideal} - q|$(理想的な比率との差分) について単調減少関数です. x, $|Q_{ideal} - q|$ が低い時は Stake 量を増やすインセンティブとして金利を上げます. x, $|Q_{ideal} - q|$ が高い時は Stake 量を減らすインセンティブとして金利を下げます.
- i_{ideal} は x, q が共に理想値であるときの Staker の平均年利 $i(x, q)$ です. 言い換えると $i_{ideal} = i(X_{ideal}, Q_{ideal})$ です.

- $I_{Staking}$ は Staking によるインフレーションレートです. x, q を含む二変数関数である $I_{Staking}$ は三次元の凸関数を描きます. $Staking$ 総量×金利 = インフレ率 と表わせ数式にすると $x \times i(x,q) = I_{Staking}$ と表現できます. また、この値は $i(x,q)$ の報酬設計から x, q が理想値である時に最大化します. 理想状態の式は $X_{ideal} \times i(X_{ideal}, Q_{ideal}) = MaxmiumI_{Staking}$ と表すことが出来ます.
- I_0 はインフレーションレートの下限値です. $x = 1$ or $x = 0$ の時、下限値に収束するようにします. I_0 はバリデータの 運用コストと同値です. なぜなら最低限バリデータの運用インセンティブを確保しなければ、チェーンが途絶えてしまうからです. ここでは $I_0 = 0.025$ を推奨します.
- d はそれぞれ x にかかる調整可能な減衰率です. x が X_{ideal} より d 増加するごとに $I_{Staking}$ が 50% 減少します. 言い換えると $I_{Staking}(X_{ideal} + d, Q_{ideal}) \geq I_{Staking}/2$ です. $d = 0.02$ を推奨します.
- g は q にかかる調整可能な減衰率です. q が Q_{ideal} より g 離れるごとに $I_{Staking}$ が 50% 減少します. 言い換えると $I_{Staking}(X_{ideal}, Q_{ideal} \pm e) \geq I_{Staking}/2$ です. $g = 0.15$ を推奨します.
- $i_{staking}$ はノミネータが Staking により得られる平均的な年利です. これはインフレ率を Staking 比率で割ることで求めることができます. 言い換えると $i_{staking} = \frac{I_{Staking}}{x}$ と表現できます.
- $I_{operators}$ は Operator が得られる報酬によるインフレ率です. これは $I_{Staking}$ における Operator への Staking が占める割合 $(1-q)$ を t 倍したものです. 式 1 と式 2 より、$I_{operator} = t(1-q)I_{Staking}$ と表すことができます.
- $i_{operators}$ は Operator が得られる報酬の平均的な (Stake された量に対する) 利率を表します. 上記の式より $i_{operators} = \frac{I_{operators}}{x(1-q)}$ です.
- I は全体のインフレーション率です. これは Staking に対する報酬と Operator に対する報酬によるインフレ率を足し合わせたもので $I = I_{Staking} + I_{operators}$ です.

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

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

ここで、式 1 から報酬の量の比率とインフレ率の比率は等しい.

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

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

$I_{Staking}$ の式は x について場合分けして以下のようになります.

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

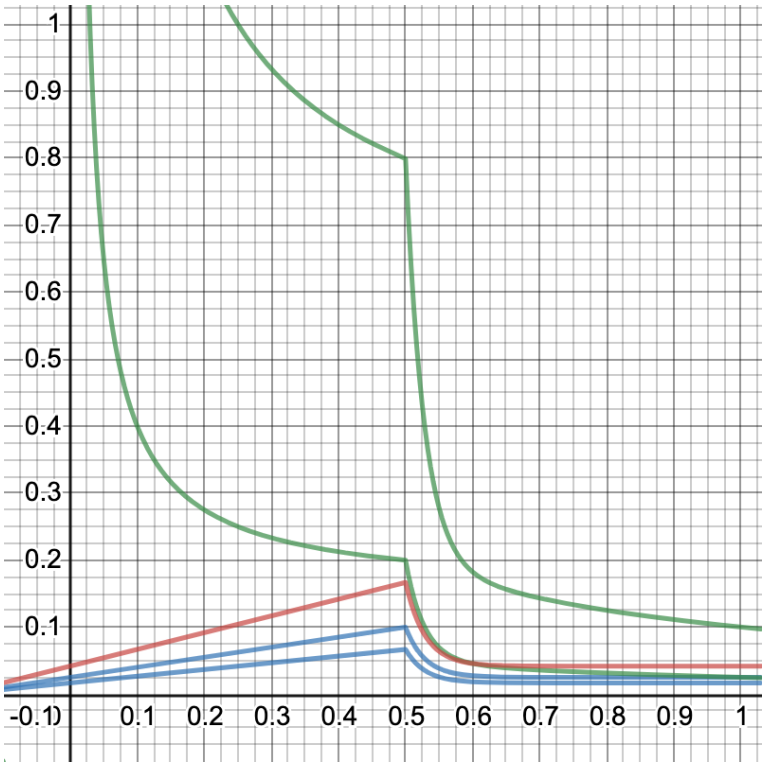

図 11　The inflation graph of $q = Q_{ideal}$.(`https://www.desmos.com/calculator/v8mrxdwbvz`)

図 11 はそれぞれ各パラメータに以下を設定したときのインフレーション率をシミュレーションしたグラフです.

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

図 11 のグラフは $q = Q_{ideal}$ で固定した際のグラフです. ここで、上の緑線が operator の staking 量に対する平均年利 ($i_{operators}$), 下の緑線が staking の平均年利 ($i_{staking}$), 赤い線が全体のインフレ率 (I), 青い上の線が Staking 報酬によるインフレ率 ($I_{Staking}$), 青い下の線が Operator 報酬によるインフレ率 ($I_{Operator}$) を意味します. x , q 共に理想値のときのインフレ率は最大で 0.166...(1/6) となります. 次に $q = 0.2$ であるときのグラフを図 12 に示します. $q = 0.2$ のとき、Staking の割合が $1 : 5 = Staking_{validators} : Staking_{contracts}$ であり、報酬の割合は以下のようになります.

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

図 12 のとき Operator 報酬が占める割合が増えているものの、q が理想値から離れているため operator の

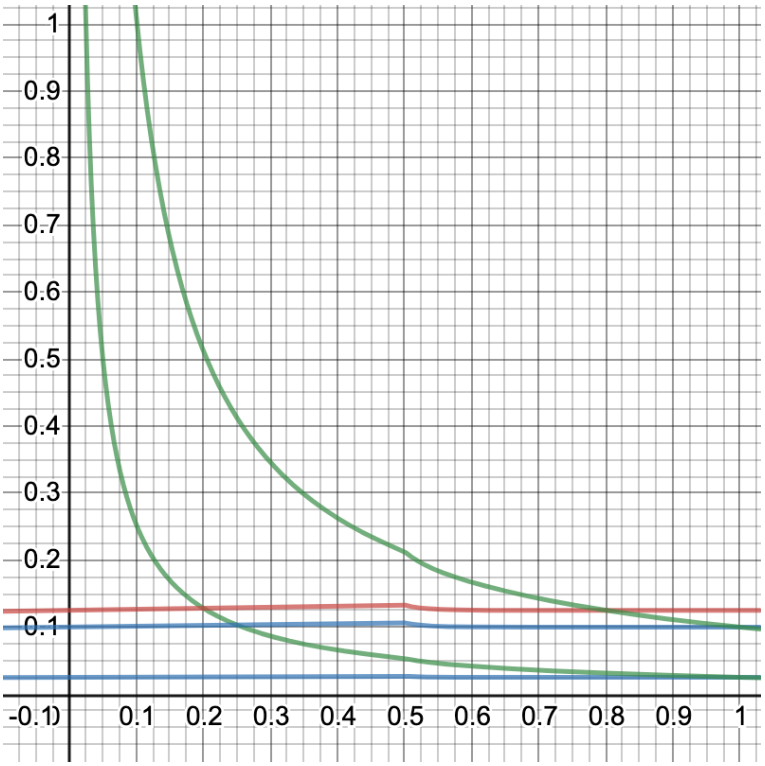

図 12　The inflation graph of $q = 0.2$.

staking 量に対する平均年利を表す上の緑線が低いことが分かります. 結果としてスマートコントラクトに対する Staking の割合が増加したとしても Operator に支払われる報酬は理想状態と大きく変わりません. また、Staking による報酬の平均年利を表す下の緑線も低下するため Staker はバランスを保つためにバリデータに Staking するインセンティブが生まれます. 次に極端な例として、$q = 1.0$ のときのグラフを図 13 に示します. このとき、誰もスマートコントラクトに対する Staking をしていない状態であり、報酬は以下のようになります.

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

図 13 のとき、Opeator が得られる報酬がゼロであり、q が理想値から離れているため Staking に対する報酬の平均年利を表す緑線が理想状態に比べて低下しています. この場合もまた、Staker はバランスを保つためにスマートコントラクトに Staking するインセンティブが生まれます. このとき全体のインフレ率を表す赤い線と Staking 報酬によるインフレ率を表す青い線が重なっているため後者が視認できないことに気をつけてください. また、これらグラフは以下の成約を満たしていることを注目してください.

- 平均年利を表す関数 $i_{staking}, i_{operator}$ は x について単調増加です.
- 平均年利を表す関数 $i_{staking}, i_{operator}$ は q が理想値である時に最大化します.
- $I_{Staking}, I_{Operator}, I$ は x, q が共に理想値である時に最大化します.
- I_0 はインフレーションレートの下限値です.
- q が理想値である時に $Rewards_{staking} : Rewards_{operator} = 5 + 1 : 4 = 6 : 4 = 3 : 2$ を常に満たしま

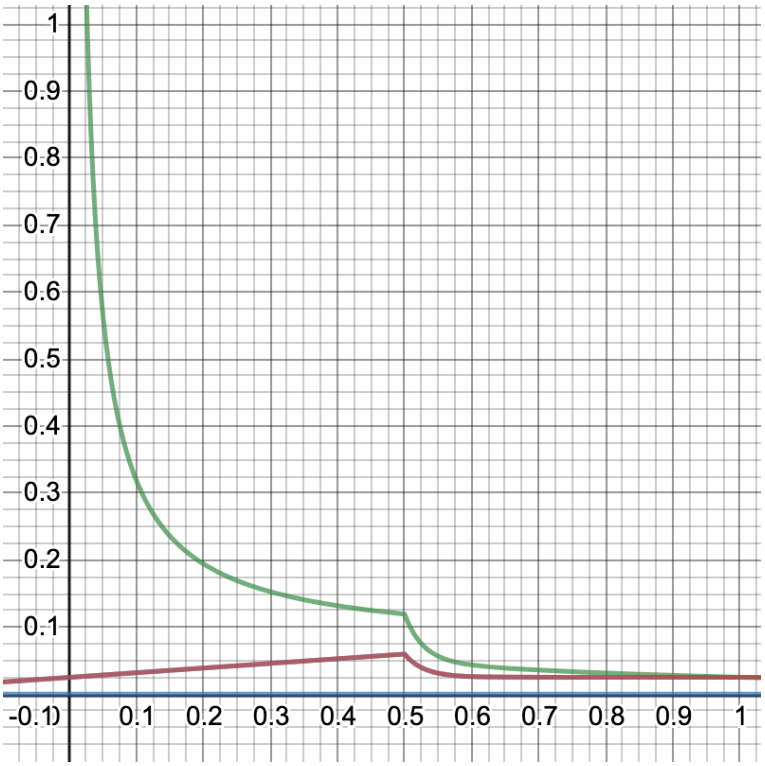

図 13　The inflation graph of $q = 1.0$.

す. 言い換えると q が理想値である時に $I_{staking} : I_{operator} = 3 : 2$ を満たします.

上記のインフレーションモデルを追加することで、Plasm ユーザのインセンティブを調整し Plasm Netowork にとって期待される行動を促します.

16.2　Transaction fee

トランザクション手数料のメカニズムは Polkadot の Transaction fee 調整アルゴリズムを踏襲しています [23].

Plasm Network では Treasury と呼ばれる資金を貯めておくための機関を保有していません. なぜなら Plasm Network は現状ガバナンスによる資金を操作を想定していないからです. Polakdot では Transaction fee の約 20% がバリデータに送金され、残りは Treasury に送金されます. Plasm Network は約 20% をバリデータに送金しますが、残りは Burn されます. これは、租税貨幣論における通貨の価値を担保するために徴収る税金の役割を果たします. Transaction fee の 80% を税金として Burn することでトークンの価値を担保します. また、一定の割合でトークンを Burn することで上記のインフレーションモデルにおける供給インフレを抑える役割も果たします.

16.3 Validator Staking

Validator に対する Staking とセキュリティ担保のメカニズムは Polkadot の NPoS アルゴリズムを踏襲しています [27].

16.4 Collator Staking

コレイターはパラチェーンのトランザクションを収集しブロックの中のトランザクションを監視し、Relaychain のバリデータにブロックの証明書をに送る役割です. Plasm Network が Polkadot の Parachain として参加している間、Plasm Network の Validator は Collator として機能します. この際の、inflation model における変数は Collator の需要に応じて変更されます.

16.5 PoA Staking

PoA は認証されたバリデータのみで合意形成を行うアルゴリズムです. Plasm Network は初め PoA Network としてローンチします. 初め PoA の参加者に平等に報酬が割り振られます. この時にバリデータに支払われる報酬は上記のインフレーションモデルにおける $Rewards_{stakers_{validators}}$ を固定したものとなります. その後、バリデータやユーザは Staking が可能になります. この時の、inflation model における変数は PoA の人数や状況によって変更されます.

Plasm Network は PoA, PoA Staking, Collato, Validator Staking(NPoS) の順でセキュリティ維持のためのアルゴリズムをシームレスに変更していく予定です. これについて Consensus Algorithm の章 12 を御覧ください.

17 Plsm as a Service

Plasm as a Service(以下 PlaaS) は『Plasm』を内部で用いることで Plasma アプリケーションを誰でも簡単にデプロイできるようにするためのサービスです. 図 14 は Plasm as a Service のメイン画面イメージです.

私達が開発している Substrate のライブラリ **Plasm** は Substrate Chain が Plasma を扱えるようにするためのライブラリです. これを用いることで、Plasma アプリケーションをデプロイ可能なパラチェーンを展開することが可能になります. スマートコントラクトでは性能の限界があり実用化に至らないようなケースは非常に多いため Plasma アプリケーションの潜在的需要は非常に高いと言えるでしょう.

しかしながら、Plasma アプリケーションを開発・運用することは容易ではありません.

全体構成図 4 を見て分かる通り Plasma は複数のコンポーネントから成り立っています. これらの構成要素すべてを 1 から実装するのは大変な作業であり通常のコントラクトを実装するより遥かに高い開発コストが発生するでしょう. Plasma 特有の状態管理を踏まえての開発は高度な技術力と Plasma への理解、そして工数を要します. また、Aggregator と呼ばれる Plasma アプリケーションの管理者が必要となり長期的な管理コストも発生します. それらの問題を解決するために我々は GUI を用いて Plasma アプリケーションをより簡単にデプロイできるサービスを提供します.

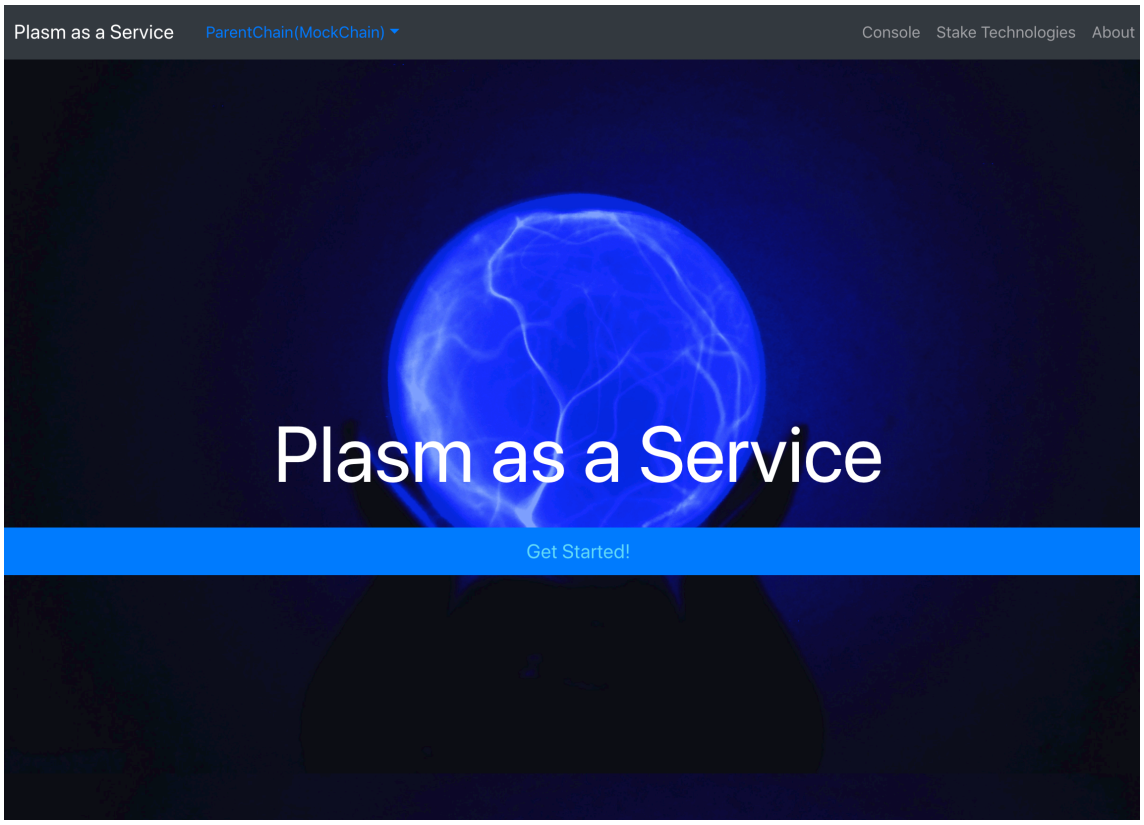

図 14　Plasm as a Service.

17.1　Functions

PlaaS は Plasma アプリケーションを簡単にデプロイ・管理することができます. 具体的には以下のステップで Plasma アプリケーションを作成していきます.

1. **GetStart**: Plasma アプリケーションを作成するスタート画面へとジャンプします.
2. **Generate Plasma App**: Plasma アプリケーションを作成する際に使うテンプレートを選択します. デフォルトでは Payment のみが可能な Plasma Cash が用意されています. また、Customizable な Plasma アプリケーションを作るためのテンプレートも用意されてく予定です.
3. **Settings**: Plasma アプリケーションの設計を記述していきます. 選んだテンプレートに応じて要求される項目を GUI 上で設定していきます.
4. **Deploy a Contract and a Child Chain:** 3 で設計した Plasma アプリケーションを実際にデプロイします. この時に使用される fee などの支払いは PlaaS が自動で行ってくれます. このフェーズでは以下を行います.
    1. 親チェーンに対する Contract のデプロイ.
    2. 2) 子チェーンの DB を持つサーバのデプロイ.
    3. 3) アグリゲータアカウントのデプロイ.
    4. ジェネシスアカウントの作成.

5. **Console**: 4 でデプロイした Plasma アプリケーションの状態や設計を確認・操作することができます. 具体的には以下の操作ができます.
    1. 親チェーンの Contract の状態の確認.
    2. 子チェーンの DB の状態の確認.
    3. アグリゲータアカウントの操作.
    4. 4 で生成したジェネシスアカウントを含む登録したアカウントの操作.

以上の機能により、Plasma のアプリケーション作成者は簡単に Plasma アプリケーションを作成し管理することが可能になります.

17.2 Demo

デモアプリケーションを試して Plasma アプリケーションを数分で作成していく過程を体験してください！(※ 実際にはこのデモでアプリケーションのデプロイはされていません.) https://mock.d3dg769h9ndpcf.amplifyapp.com

18 Conclusion

私達は Plasm Network に以上の仕組みを導入することで Plasma Application の構築に特化したブロックチェーンを構築します. 流動性のあるトークン発行を低コストで行う仕組み、安全に分権化されたブロックチェーンシステムを拡張する仕組み、そして Plasma application の開発者に報酬を与える仕組みの 3 つの仕組みが Plasm Network を繁栄させることでしょう.

Plasm Network を通して私達はすべての開発者にスケーラブルな分散アプリケーションの開発メソッドを Polkadot 上で展開します. 複数の異なるブロックチェーンを束ねる Polkadot に優秀なスケーリングソリューションである Plasma を展開できる恩恵は計り知れません. 私達は典型的な Plasma アプリケーションを PlaaS を使ってものの数分で deploy できるようになります. 複雑な Plasma アプリケーションもカスタマイザブルな Plasm ライブラリを用いることで標準規格に準拠した正しいアプリケーションを開発することができます. また、私達はユーザースペシフィックなユースケースについてサポートするでしょう. Plasm Network ではデプロイされた Plasma アプリケーションは任意で市場に出すことができます. そのアプリケーションが保有するエコノミーを売買することでさながら M&A のようなやり取りが可能になります. 故に、Plasma アプリケーションはもはやただのアプリケーション以上の価値を持っていると言えるでしょう. 私達は Plasma アプリケーションが栄えた新しい Decentalized な世界観を楽しみにしています.

19 Acknowledgment

これらの技術の組み合わせは Plasm ライブラリを支援して頂いた Web3 Foudnation、Substrate/Polkadot の開発元である Parity technologies, Plasma R&D の最先端を担う Plasma Group と Cryptoecnomics Lab の協力なくしては実現できません. 彼らに深い謝辞を申し上げます.

参考文献

[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

[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/