

あらゆるアセットを管理するビジネスロジックを兼ね備えた 汎用型分散台帳基盤の開発 — Proskenion —

1. 背景

ブロックチェーン技術はビットコイン開発の過程で生まれ、ビットコインの取引を記録する分散型台帳を実現するための技術であった。分散型台帳技術はデータベースを一つのサーバで中央集権的に管理するのではなく複数のサーバで分散管理することで、単一点障害耐性を持ったデータベースである。ブロックチェーンはそれに加え「ブロック」というデータの単位を生成し、チェーンのように連結していくことによりデータを保管する。このブロックは一つ前のブロックのハッシュ値を持っているため、ブロックに書き込まれたデータを改ざんするにはその場所から先頭までの全てのブロックを改ざんする必要がある。このため、ブロックチェーンでは改ざんに対する強い耐性を持っているとされている。分散台帳では複数のサーバが同一のデータを保持するために様々な合意形成アルゴリズムが考案されている。昨今では様々なブロックチェーンプラットフォームが開発されているが、未だに技術的課題がいくつも残されており、どれも実用に耐えうる代物ではなかった。代表的な問題としてスケーラビリティ、情報透過性、などがある。また、仕様変更に伴って発生するハードフォークにも問題があり、合意形成アルゴリズムの移行は非常に大変な作業であった。さらに、予め定義された報酬設計に欠陥が見つかった場合に後になって変更することが出来ず、将来的な展望に不安を持つ要素が多々ある。

2. 目的

本プロダクトProskenionは一般的なブロックチェーン特有の機能である非中央集権的、非改ざん性、単一障害点耐性に加え、ブロックチェーンの合意形成と報酬設計をハードフォーク無しに変更できる仕組みとプリミティブなコマンドの組み合わせで高い表現力のオブジェクトを操作できる仕組みを導入したシステム開発を目的とした。これによりハードフォークの問題を解決すると共にマイニングの主体となる母体をより自由に定義できるようにした。また、チェーンが過渡期を迎えて初期の報酬設計が機能しなくなった際に設計し直すことも可能となる。

3. 開発の内容

3.1. システム構成

ブロックチェーンに必要な構成を、マイクロサービスアーキテクチャのようにいくつかの独立した機構に分けて考えた (Figure 1)。ソフトウェアアーキテクチャとしては変更強いクリーン・アーキテクチャを意識しつつ、ディレクトリ構造的にはEthereumのプロジェクトを参考にした。また開発言語にはGo Languageを採用し

た。これらはOSSで開発する上で参入障壁と機能変更に対する柔軟性を担保するためである。

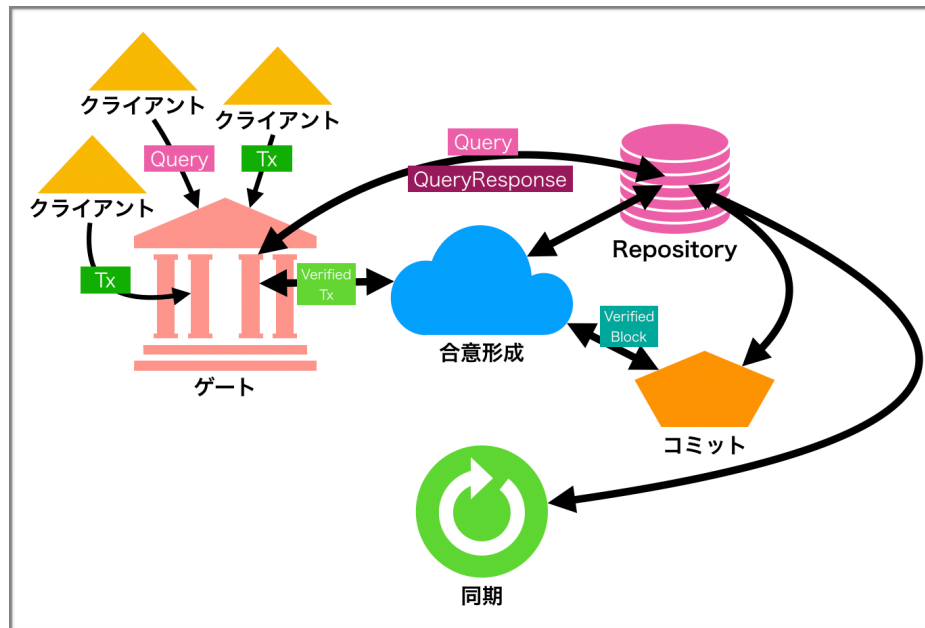


Figure 1. Blockchain Architecture

3.2.機能

Proskenionはパブリックブロックチェーンの共通機能である①非中央集権的な分散ノード管理、②改ざんが困難、③単一障害点耐性の3点を実現した。それに加えてProskenion特有の機能として④プリミティブな命令セットの組み合わせによる高い表現力、⑤独自DSLであるProskenion Domain Specific Language (ProSL) によるインセンティブ／合意形成のカスタマイズの容易化、⑥ハードフォーク無しで変更可能なインセンティブ／合意形成の仕組みの3点を実現した。

3.3.特徴的技術

Proskenionは④⑤⑥の実現のために合意形成を設計するためのDSL、“ProSL”が組み込まれている。ProSLは“Consensus algorithm”, “Incentive algorithm”, “Update algorithm”そして“Genesis Block”の4種類がある。これらはそれぞれ異なるタイミングで動作する。Consensus Algorithmはブロックの生成者を選出する仕組みを設計する。Incentive Algorithmはインセンティブ付与の仕組みを設計する。Update AlgorithmはConsensus, Incentive, Update algorithmを変更するための条件を設計する。Genesis Blockは高さ0のブロックに刻まれる命令群を設計する。

4. 従来の技術（または機能）との相違

従来のブロックチェーンプラットフォームとの最も大きな相違点として、合意形成／報酬設計の柔軟性がある。また、他のプラットフォームが金融向けの用途に特化

する傾向があるところを、Proskenionはエンタメ／社会実験の領域に有用であるように差別化している。

5. 期待される効果

自由に合意形成と報酬設計を定義できるということは、マイナーとなる主体とマイニングの行為の代替を定義できると言い換えることができる。これを利用して、コンテンツの配信者（以下「クリエイタ」）が良いコンテンツを頒布する行為をマイニングとすることで、クリエイタ自身が運営主体となる新しいブロックチェーンを作り出すことができる（Figure 2）。

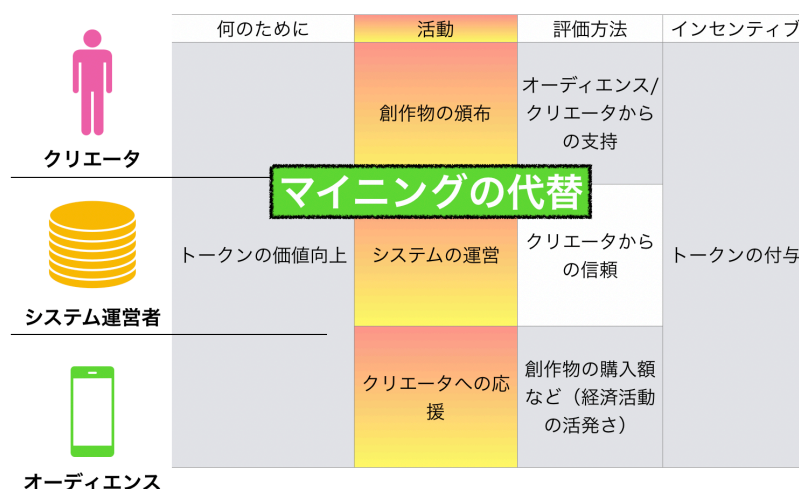


Figure 2. Mining for creators.

また、動的に合意形成と報酬設計を定義できる特性を活かして社会実験で利用する形も検討できる。「こういった報酬設計」が「どのような結果をもたらすか」の実験を中～小規模で行うことができるため、大規模に適用したい社会実装を小さな集団の中でシミュレーションできるようにする効果が期待できる。

6. 普及（または活用）の見通し

Proskenionの応用先として考えられる例は以下のものがある。

・ オンライン同人即売会

オンライン上でコミックマーケットのような仕組みを模擬する。Proskenionを用いることで自治コミュニティが運営する同人即売サービスが実現できる。クリエイタを同人作家と定義して作品の販売をマイニングと位置づける。評価は他のクリエイタからの信頼度と販売利益と定義する（Figure 3）。これによりシステム管理者により制限を受けない自分たちだけの販売所を作ることができる。

・ ミスコン

システムのテスト運用という意味でミスコンのような短期的な人気投票システムを行うのも面白い。ミスコンのエントリー者をクリエイタと定義して彼らのインセン

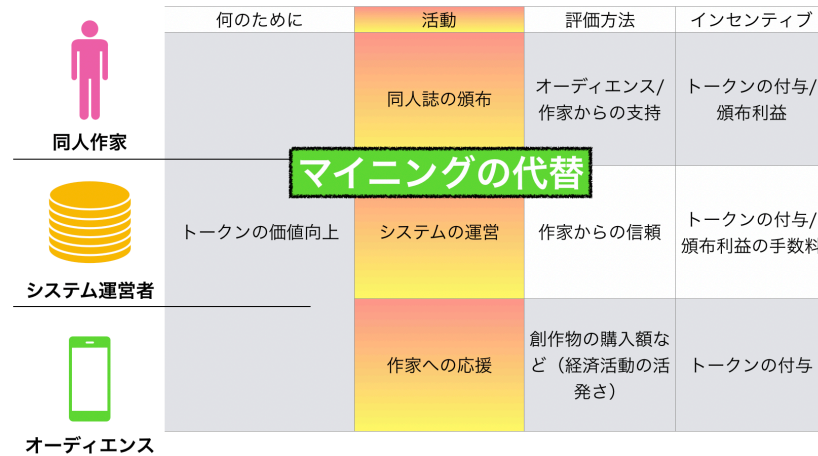


Figure 3. Mining for Online comic market



Figure 4. Mining for Ms.Mr. Contest

ティブ設計が直接評価方法となり実証実験的なコンテストが開催可能であると思われる (Figure 4)。

7. クリエータ名 (所属)

山下 琢巳 (東京大学 大学院情報理工学系研究科 創造情報学専攻)

(参考) 関連URL

1. [proskenion] <https://github.com/proskenion/proskenion>
本プロダクトのメインソースコード
2. [satellitex/protobuf] <https://github.com/satellitex/protobuf>
本プロダクト用のmapのMarshalを決定的にしたprotobufライブラリ
3. [satellitex/pagerank] <https://github.com/satellitex/pagerank>
本プロダクトで使う用にカスタマイズしたpagerankアルゴリズムライブラリ
4. [API Documents] <https://proskenion.github.io/docs>
本プロダクトを操作するための API Documents。