智慧喰 白書 (draft)

水樹素子 sonoko_mizuki

平成 29 年 12 月 19 日

概 要

この世には、様々な分散台帳システムが数多に存在しているが、未だ研究の余地があり課題を抱えている。よってここに、人類を一歩すすめるため新しい分散台帳システム”智慧喰”を提案する。この文書は、”智慧喰”の思想、設計、プロトコル、そして未来についてまとめたものである。

1 Introduction

1.0.1 由来と目的

このシステムの名前を”智慧喰”としたのは以下の目的を満たすことが最終的な到達点だからである。

- 智慧を食べるものとして知識を持った人々が集い、叡智を組み合わせより強固なシステムと成り、人類の礎を築く事を目的とする。
- 情報を食べるものとして異なる台帳が集い、巨大な 1 つの台帳と成り、利用者は情報の位置を知らなくとも参照、変更可能なシステムを築く事を目的とする。

1.1 なせ作るか？

人類が作ってきたインターネットを、一歩先に更新したい。

2 智慧喰い

ここで私の提案するシステム、”智慧喰”について説明する。

2.1 定義

以下の条件を満たすものを**智慧喰システム**と定義する。

1. Command を受け取り、システムの情報を変更する事ができる。
2. Query を受け取り、システムの状態を返す事ができる。
3. 後述するデータモデルを持つ。
4. システムを構成する Peer を任意に増減することができる。

以下の条件を満たすものを**智慧喰システムの完全 Peer** と定義する。

1. 以下の Endpoint を持ち、Peer 間、Client-Peer 間で通信を行う事ができる。
   - API: Command/Query 用 Endpoint
   - Consensus: 合意形成用 Endpoint
   - Internal: 内部通信用 Endpoint
   - Supervise: 監視用 Endpoint
2. 指定された形式でデータを保存することができる
3. 合意形成に参加し、データの整合性を確認することができる

上記の条件から 3 項を抜いたものを **智慧喰システムの準 Peer** と定義する。

2.2 構成要素

智慧喰システムの完全 Peer は以下の装置によって構成されている。なお存在理由を右に記述する。

1. Command/Query を受け取り、受け取った事実をトリガーとして任意のイベントを発生させる装置。受け取った事実を記録するため

2. 受け取った Command/Query の受容可能かを判断する装置。正しい Protocol に基づいた物のみを選別するため

3. Command を下記のデータフローに基づき適切に分配/合意形成を行う装置。システム全体として 1 つの状態を保つため

4. 合意形成後の Command を元にデータを変更する装置。システムの状態を更新するため

5. Peer の管理を行い、システム全体の安定を図る。システムを停止させず、拡張や入れ替えを容易に行うため

智慧喰システムの準 Peer は以下の装置によって構成されている。

1. Command/Query を受け取り、受け取った事実をトリガーとして任意のイベントを発生させる装置。受け取った事実を記録するため

2. 受け取った Command/Query の受容可能かを判断する装置。正しい Protocol に基づいた物のみを選別するため

3. 合意形成後の Command を元にデータを変更する装置。システムの状態を更新するため

4. Peer の管理を行い、システム全体の安定を図る。システムを停止させず、拡張や入れ替えを容易に行うため =

2.3 プロトコル

智慧喰は以下のプロトコル階層を持つ※画像に変える

1. Application Layer

2. Entity Layer

3. Ledger Layer

2.3.1 Application Layer

HTTP だとか GRPC だとか JSON？Protobuf なんか色々、受け入れてく

2.3.2 Entity Layer

どの Domain に属していてどの Account が主体的でどの Asset に対してどんな Command をするのか的な純粋な世界
でかいかな？Domain Layer 提唱？

2.3.3 Ledger Layer

どの Peer が受け取っただとか、どの Peer に投げ込んだらいいのかとかとか
智慧喰内では一番物理に近い

※合意形成とかどこ入れる？Query は合意形成いらないよね？そこんとこいい感じに
※合意形成いらないとか言っちゃった場合のシステムを許容していこう？
※どかーんと Peer が増えた場合、Peer 探す時どうするん

2.4 特徴的な機能

2.4.1 三権分立的権限管理

司法、立法、行政的分権システム。ルールを制定する者はルールを実行する者ではない
ルールを実行する者はルールに則らなければならない
ルールに矛盾が発生することは許されない

仰々しく書いてあるが特定のアカウントが暴走してしまわないよう、できることを分散して複数人の承認なしではでかいことは出来ない、という仕組み。
わかりやすく三権分立を出したが、二分割でも機能する？

2.4.2 移管

管理を信頼のおけるシステムに任せる事ができる。
これは低 Layer のお話
運用者は大変なので自分で運用したくないときは権限を移管して任せてしまおう

この機能がなんで必要？従来のもので良くない？何がしたいの？

3　仕組み

3.0.1　台帳内でのデータ共有

これからうめるよ～

3.0.2　台帳同士の結合

これからうめるよ～

4　FlagshipPeerApplication

ここで長々と描いても結局は机上の空論。実際使ってみないとわからない。
なので個々で FlagshipPeerApplication ”智慧喰 Peer ”について紹介する。

4.1　構成

これからうめるよ～

5　未来

これからうめるよ～