ブロックチェーン公開講座 第13回
Dappsの事例（DeFi）

芝野恭平

東京大学大学院工学系研究科技術経営戦略学専攻
ブロックチェーンイノベーション寄付講座
特任研究員
shibano@tmi.t.u-tokyo.ac.jp

Agenda

- DAppsとして向いているアプリ，そうでないアプリ
- 事例紹介
  - Uniswap V2
  - MakerDAO SAI

- 注意：本講義で取り扱う情報は，技術的な情報の共有を目的としており，投資等を推奨するものではありません.

一般的なアプリケーションとブロックチェーンでのアプリケーション（DApps, Decentralized Applications）の違い

一般的な（中央集権型）アプリ

特性
　中央管理者の存在:
　　すべてのデータと処理が中央のサーバーで管理される.
　　管理者はシステムのアップデートやメンテナンスを行う.
　迅速な処理速度:
　　高速なリクエスト・計算処理が可能.
　　一度に多くのユーザーがアクセスしてもスケーラブルに対応できる.
　運営者側のコスト負担:
　　ユーザー側に追加のコスト（例えばガス代）がかからない.
　　サーバーコストは運営者が負担する.
　プライバシーとセキュリティ:
　　データは中央管理者が保管し，アクセスコントロールを行う.
　　セキュリティ対策は運営者に依存する.
制約
　単一障害点:
　　中央サーバーがダウンすると，すべてのサービスが停止する.
　信用（Trust）の問題:
　　ユーザーは中央管理者に対する信用を必要とする.
　　データの改ざんやプライバシー侵害のリスクがある.

ブロックチェーンを用いたアプリ（Dapps）

特性
　非中央集権型:
　　中央管理者が存在しないため，管理者不在のアプリが実現可能.
透明性:
　　トランザクションはブロックチェーン上に公開され，誰でも検証可能.
　　データの改ざんが非常に難しい.
　　スマートコントラクトのコードの検証可能性.
永続性・耐障害性:
　　一度記録された情報は半永久的に保存される.
　　地理的にも分散しており，システムを止められない.
　　ノードの一部がダウンしても，他のノードがシステムを維持する.

制約
　処理速度とスケーラビリティ:
　　一般的にトランザクション処理速度が遅い.
　ガス代:
　　各トランザクションに対してガス代（手数料）が発生する.
　　ユーザーにとってコストが高くなる場合がある.
　　※ 一部のコントラクトウォレットを使うことで，ユーザーには一切のガス代負担をなくすことも可能.
　プライバシーの課題:
　　トランザクションは公開されるため，完全なプライバシーを確保するのが難しい.

オンチェーンの計算能力について

- オンチェーンでの計算性能は，思っている50万分の一くらいです.
  - → MIPS (million instructions per second)を計算.
    - https://ja.wikipedia.org/wiki/MIPS
    - https://en.wikipedia.org/wiki/Instructions_per_second

  - 30M / 3/14 = 0.71 MIPS （参考： https://medium.com/validitylabs/ethereum-smart-contracts-hello-1970s-83c18e3d6398 ）
  - Intel Core i5-11600K (6-core)　　346,350 MIPS at 4.92 GHz

  - 1977年くらいのCPU性能

- 各種スケーリングソリューションの台頭で，できることは少しずつ増えてきてはいるもののやっぱり一般的なアプリケーションの比ではないくらい軽いものしか実現できません.

- 向いていないこと：
  - 写真や動画の処理
  - 大規模なデータの計算

DAppsに向いているものはどういうアプリ？

- 非中央集権型システム：
  - これが不要な場合は一般的なシステムの構築を模索するほうがいいかもしれません.
  - 共通のルールを定義する創造者になる
    - 中央集権型システムでも同じルールをもって運用することは可能
    - しかしながら，現実世界の制約を受けにくい.
      - 誰かに持ち逃げなどされない．（コントラクトにバグがないかは注意）
      - 一部の地域の人にしか使用できない，というようなことがない.
      - スタートアップなどがユーザーの資金を扱ったアプリケーションを作れる.
        - 資金や情報の保全・管理力の乏しい企業が，それをブロックチェーンでカバーするソリューションを作れる.
    - 公開後に自分すらも変えられないルール
      - 変えられないことが価値を生むこと（例：遺産の分配
- 計算能力が不足：
  - 数字の計算やそれだけで価値がもたらされるもの.
  - DeFiやNFT
- 永続性：
  - 消せない，消したくないことを記録すること.
  - Timestampを用いてその時点での存在を証明可能.

DAppsの事例（DeFi，Decentralized Finance）を見てみる

- 今回の講義では，DeFiの事例としてUniswapとMakerDAOの仕組みを見ていく．

- コントラクトの仕組み
  - 具体的な仕組み．

- 非中央集権型でワークするようにどのように設計されているか
  - どんなプレイヤーが想定されているか
  - それぞれのプレイヤーに対して，想定するように動いてもらえるようにどのようにインセンティブが設計されているか

- どのように動くのかをシミュレーション分析を行うことが重要
  - 仮に自分がDeFiをなにか設計することを想定すると・・・
  - 設計通りに動くかどうかシミュレーション分析を行うことが重要
  - 実装はコントラクト（Solidity）で行うが，そうではなく，設計通りに動作をするかシミュレーションプログラムを使って思ったような挙動をするかをチェックする事が重要．
    - Pythonその他の言語は，簡易的で便利な関数が豊富．計算能力も高い自分のPCでのシミュレーション分析が可能．

- 公開されているソースコードを見てみる．
  - ドキュメントだけだと理解が難しいところもある．
  - 細かい挙動の部分を確認するためにソースコードを見てみる．

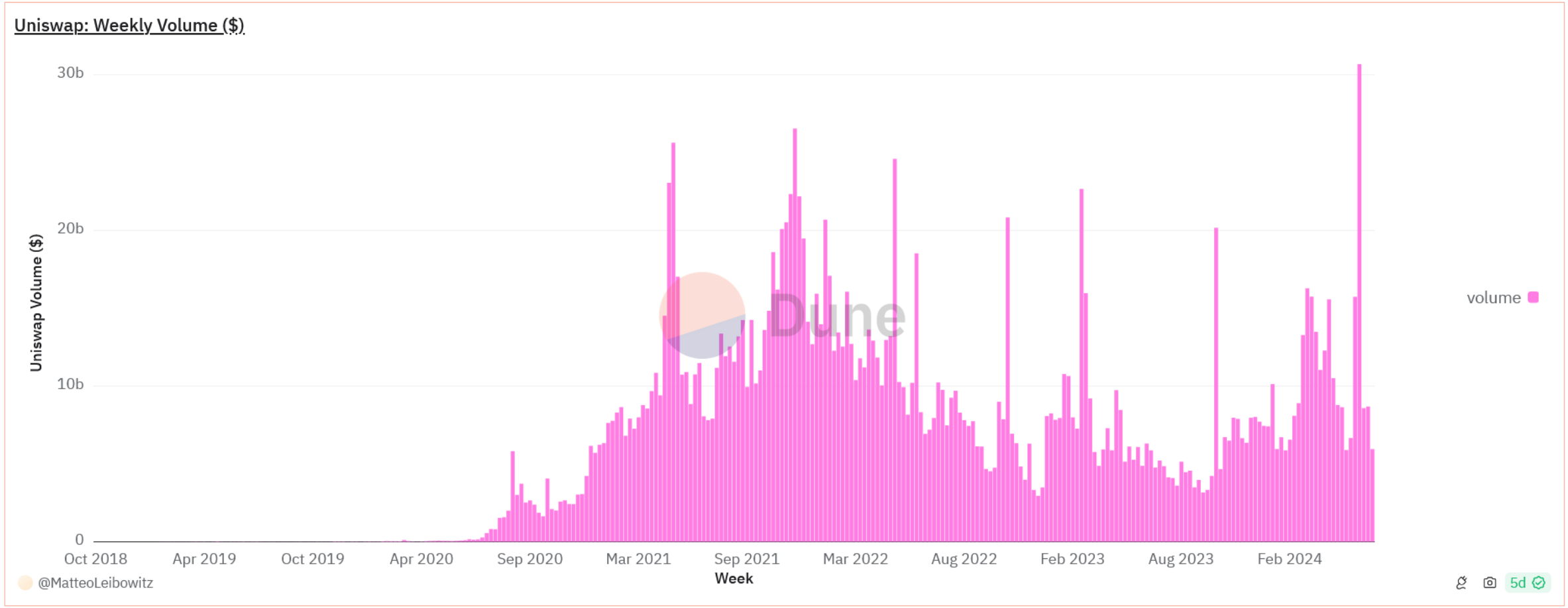
Uniswapとは

- ブロックチェーンを代表するDEX（Decentralized EXchange）
- トークンの交換・売買は，従来型の板取引ではオンチェーンでの実現が難しかったところ，AMM(Automated Market Maker)を実装することによりDEXを実現.

- 2018年にUniswap V1誕生後，2020年にV2，2021年にV3と，機能の拡充を重ねていっている.
  - V1: ETHとの任意のトークンペア
  - V2: 任意のERC20同士のトークンペア
  - V3: 流動性提供の資本効率向上

- 本講義では，取引量が増大するきっかけとな，仕組みが比較的シンプルなV2の紹介を行います.

Uniswapの利用ボリュームの推移

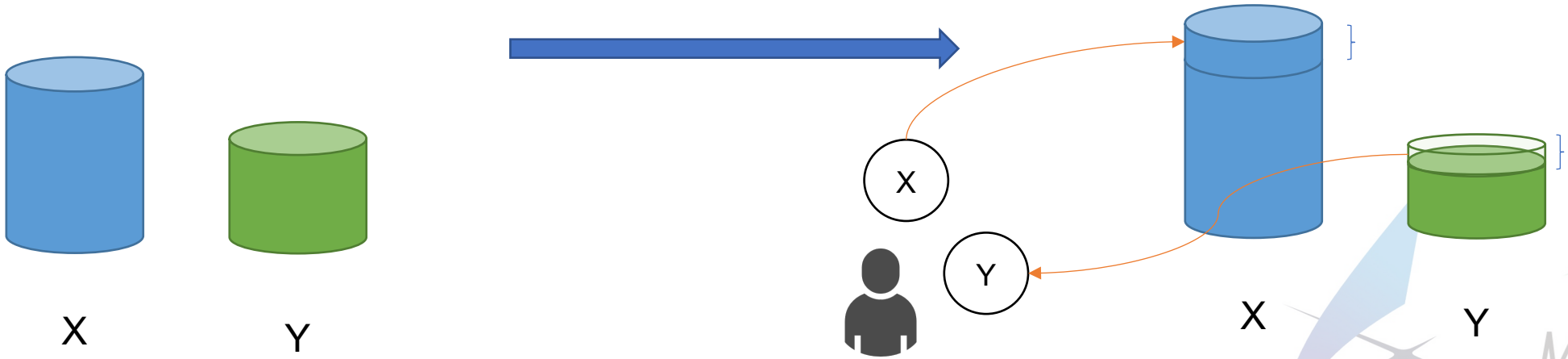


https://dune.com/queries/7867/15678

Uniswapの仕組み

- 売り手と買い手をマッチングさせるのではなく，コントラクトのプールに対して売り買いを行う．
- 任意の2つのトークンを交換できる，トークンペアがそれぞれ存在．
- 交換時には，プール内のそれぞれのトークン残量が適正になるように価格が算出されるようになっている．
- プールにトークンを供給する人は，それに応じた手数料収入が得られる．
  - 流動性を提供する，という．

トークンXとトークンYのペア



トークンXをYにSwap(交換)したい

Uniswapのプレイヤー

- トークンのSwapをしたい人
  - トークンAをもとにトークンBを購入したい
- 流動性提供者

- という2人のプレイヤーを想定して実現.
- この人たちはインセンティブのみで動いており，誰にも強制されていない.

LPトークン

- LPトークン（Liquidity Poolトークン）とは，プールに資産を預け入れるとその代わりに受け取れるトークン.
- LPトークンは，トークンペア（X，Y）ごとに異なるLPトークンとして存在する.
- トークンペアのプールにX, Yトークンを預け入れるとそれに応じたLPトークンを受け取れる.

- LPトークンは，トークンペアに返却すると，プールに入っているトークンX, Yがそれぞれの比率で受け取れる.
  - このとき返却されたLPトークンはバーンされる.
- 例： Xが20トークン，Yが50トークン入っているトークンペアのプールがあり，LPトークンの総発行量は10,000トークンとする．このとき，ある人が手元の300LPトークンをプールに返却したときに受け取れるX, Yトークンそれぞれの量は以下：

  X： 20 * 300/10,000 = 0.6 トークン

  Y： 50 * 300/10,000 = 1.5 トークン

- トークンペアにてスワップするときは手数料がかかり，その手数料はプール内に追加される.
- つまり，スワップされればされるほど1 LPトークンに割り当てられるX, Yトークンの量が増える.

LPトークンの発行（初期プール）

- トークンペアを新しく作るときと，すでに存在しているトークンペアに流動性を提供する際では受け取れるLPトークンの計算式が異なる．
- 新しくプールを作る場合は，以下の式で計算されるLPトークンがもらえる．
  - MINIMUM_LIQUIDITY = 1000は引かれてそのままバーンされる．
  - すなわち一度プールを作ったらどれだけトークンがなくなっても，MINIMUM_LIQUIDITY分の残高は必ず残ることになる．

トークンXとYのペアを作る場合：

x_0 ：トークンXの初期プール量

y_0 ：トークンYの初期プール量

この場合，以下の式で計算されるLPトークンがもらえる：

$$liquidity = \sqrt{x_0 * y_0} - MINIMUM\_LIQUIDITY$$

LPトークンの発行（既存プール）

- 既存プールに追加で流動性提供をする場合は

$$liquidity = \min\left(\frac{x_{deposited}}{x_{reserved}} * totalliquidity, \frac{y_{deposited}}{y_{reserved}} * totalliquidity\right)$$

で計算される量のLPトークンが受け取れる.
X, Y2つのトークンのうち計算して少ない方を採用

$x_{deposited}$ ：追加で入れるXトークン量
$x_{reserved}$ ：元々プールに入っていたXトークン量
$totalliquidity$ ：LPトークンの総発行量

- 追加する各トークンの量は，元のプール内の比率と同じ量を入れる必要がある.

例：X:20トークン，Y: 50トークンのプール，総発行LPトークンの量は10,000を仮定.
このとき，Xを6トークン分流動性提供をしようとしたときには同時に必要なYのトークン量は以下：

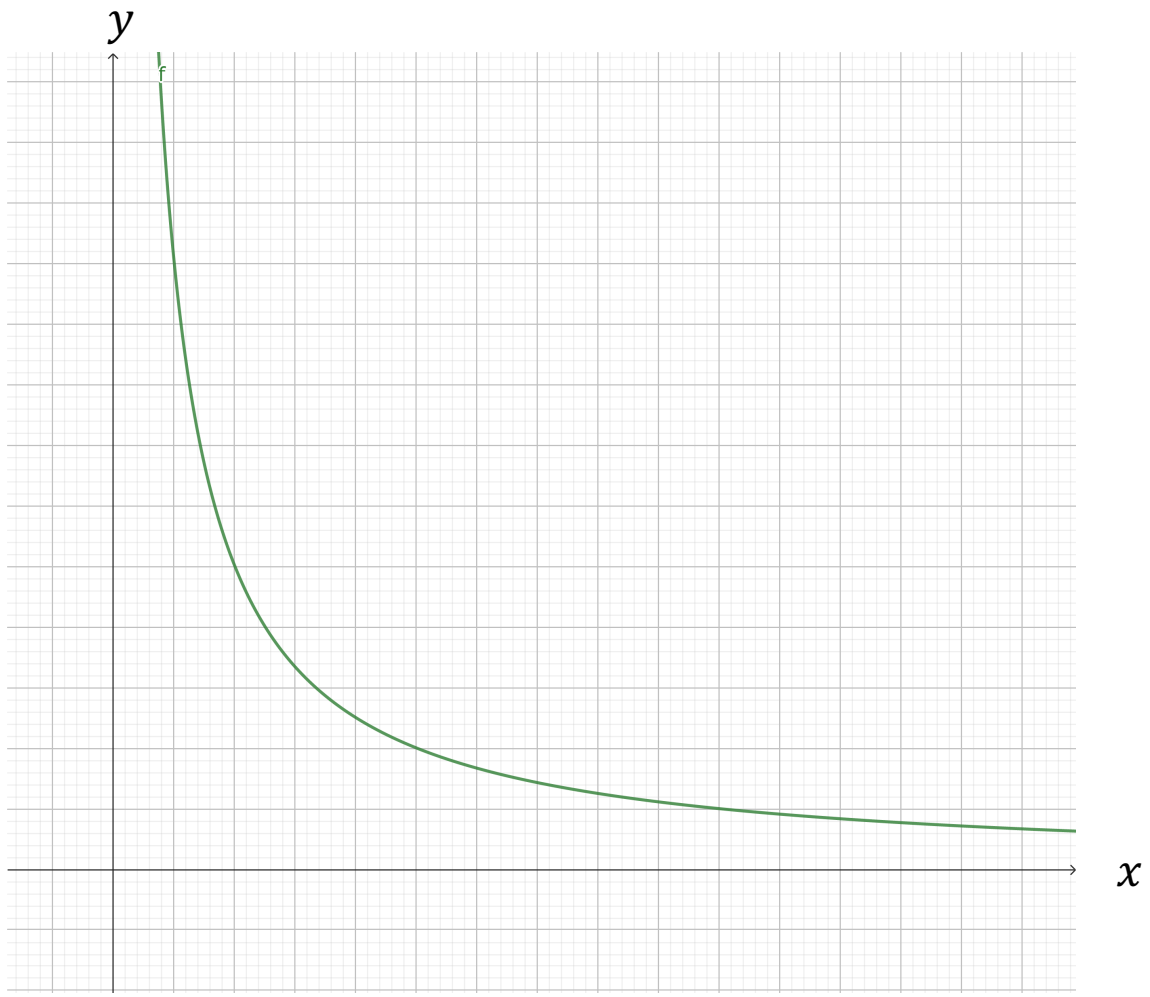
$$6 * \frac{50}{20} = 15$$

このとき受け取れるLPトークンは

$$min\left(\frac{6}{20} * 10000, \frac{15}{50} * 10000\right) = \min(3000,3000) = 3000$$

プール内の残高は，X:26トークン，Y:65トークン，総発行LPトークン：13,000

Swap（交換）の仕組み



https://www.geogebra.org/graphing?lang=ja

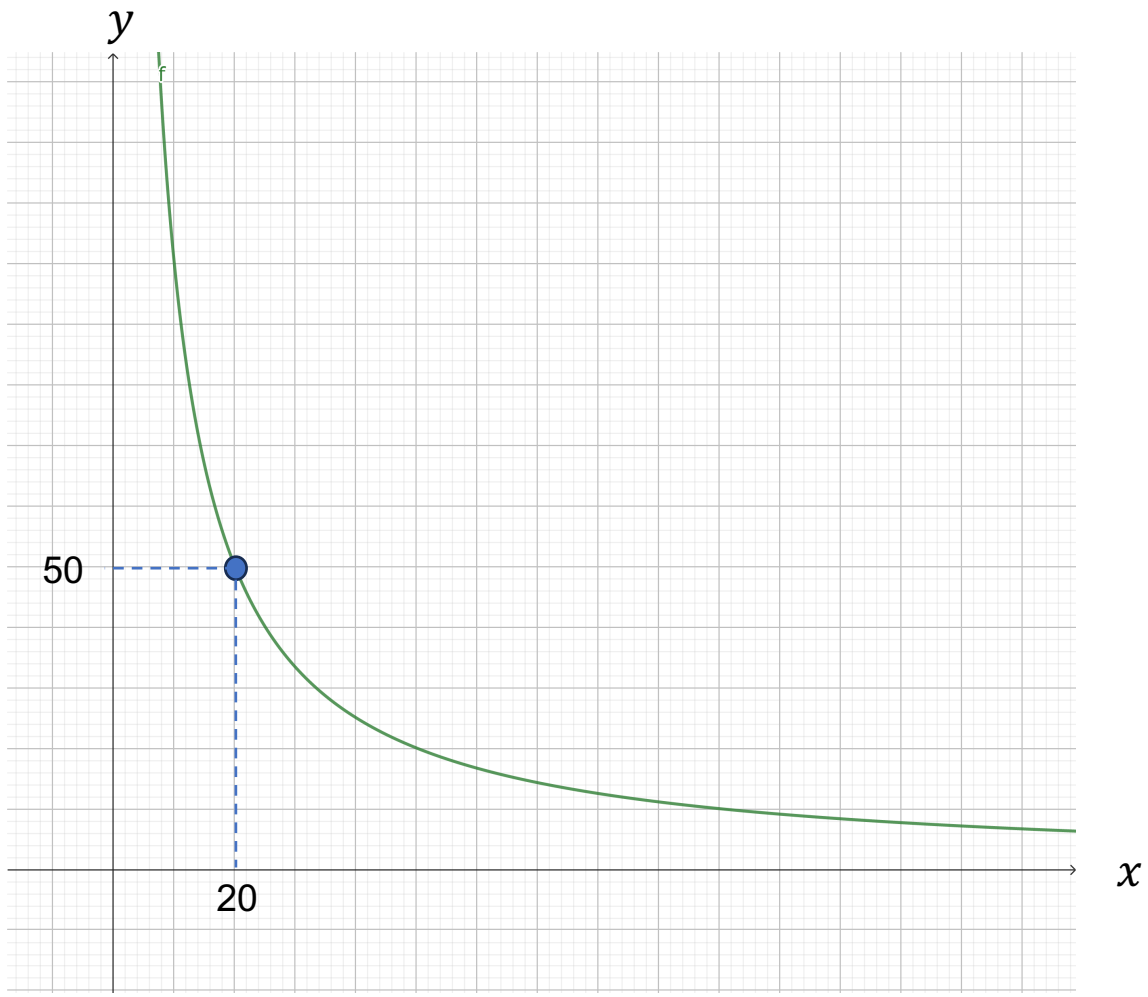
トークンペアX，Yでトークンの交換を行う．
そのプール内のトークン量について以下の式が成り立つようになっている．

$$x * y = (x + \Delta x) * (y + \Delta y) = k = 一定$$

x, y	トークンX, Yそれぞれのプール内の総量
$\Delta x, \Delta y$	Swap時のそれぞれのトークンの交換量．
k	交換時には一定． プール内の各トークン量により変動

※ 例えば，WETHとUSDCの通貨ペアがあったときに，0.5 WETHをUSDCに交換したいとき，Δx=0.5で，Δyが受け取るUSDCの量として上式で計算される．

Swap（交換）の仕組み



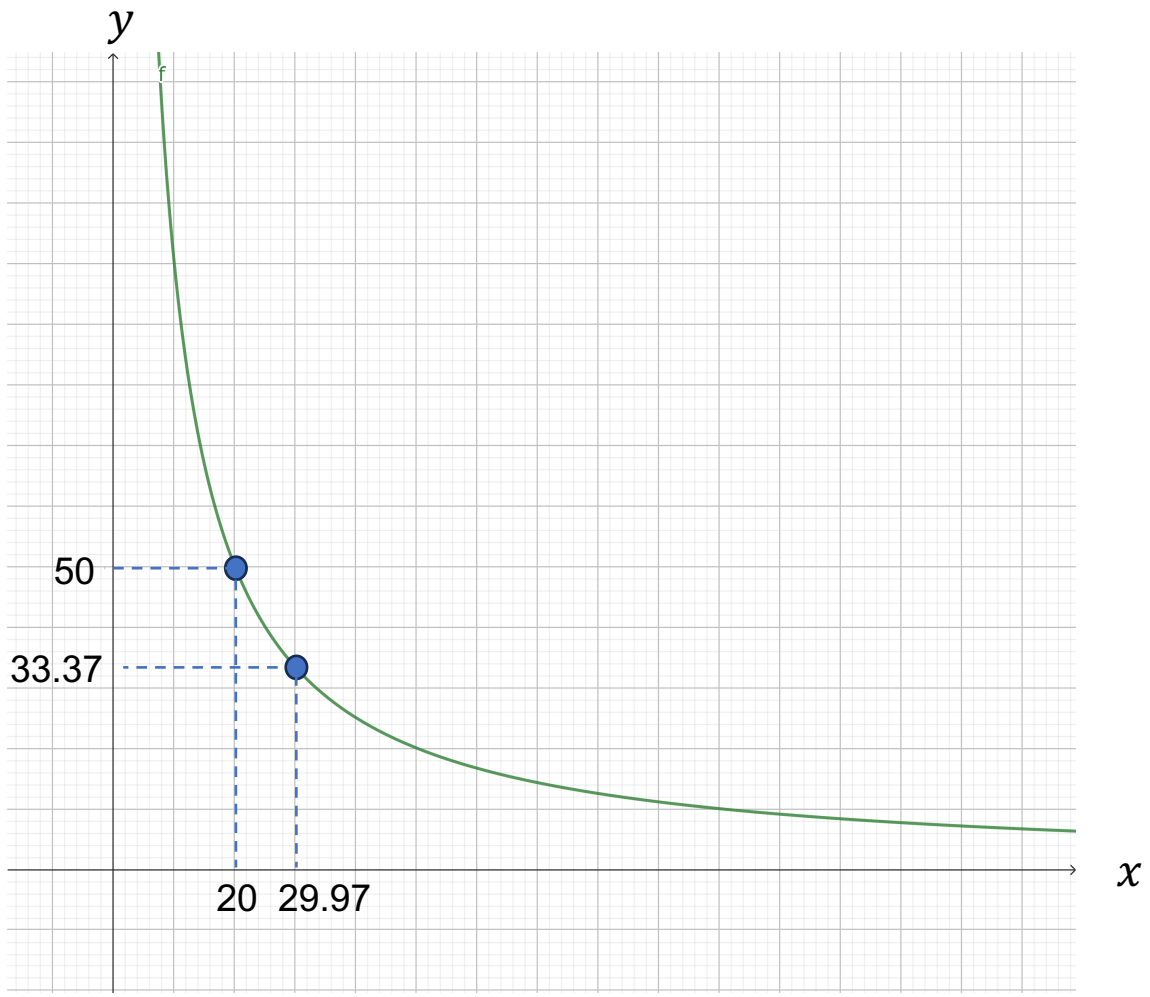
x = 20トークン, y = 50トークンと仮定.
　このときk = 1,000

Xを10トークンこのプールに入れてトークンYにSwap(交換)することを考える.

このときトークンYをどれだけ取得することができるか？

手数料は0.3%

Swap（交換）の仕組み



手数料は
10 * 0.3% = 0.03
で，Xトークンで支払われる．
この0.03トークンはプール内に入る．

Xトークンは手数料が引かれた9.97トークンの対価としてのYトークンを受け取る．
プール内のx,yの積は定数kである必要があるので，xが9.97増加したあとのyの値は曲線上の点で，
y = 1000/ (20+9.97) = 33.37
となる．

もともとはy=50だったので，
50 – 33.37 = 16.63
トークンを受け取ることができる．

Swap（交換）の仕組み

- このとき，コントラクト内のプールは以下の変化がある.
- x = 20 → 30
- y = 50 → 33.37
- k = 20* 50=1000 → 30 * 33.37 = 1001.1

- 手数料分だけプール内の量（k）は増大する.

Uniswap V2のシミュレーション分析

- なんとなく理屈はわかったけど，この仕組みでちゃんと安定して交換がされるかどうかよくわからない．
- シミュレーションによって取引価格が安定しているか，プール量はどれだけあれば安全なのかをシミュレーションで分析してみる．

- Uniswappyを用いて分析を行う．
  - https://github.com/defipy-devs/uniswappy
  - Python
  - Uniswapは有名なだけあって，上記のようにシミュレーション分析用のツールが存在していました．

Uniswap V2のシミュレーション分析

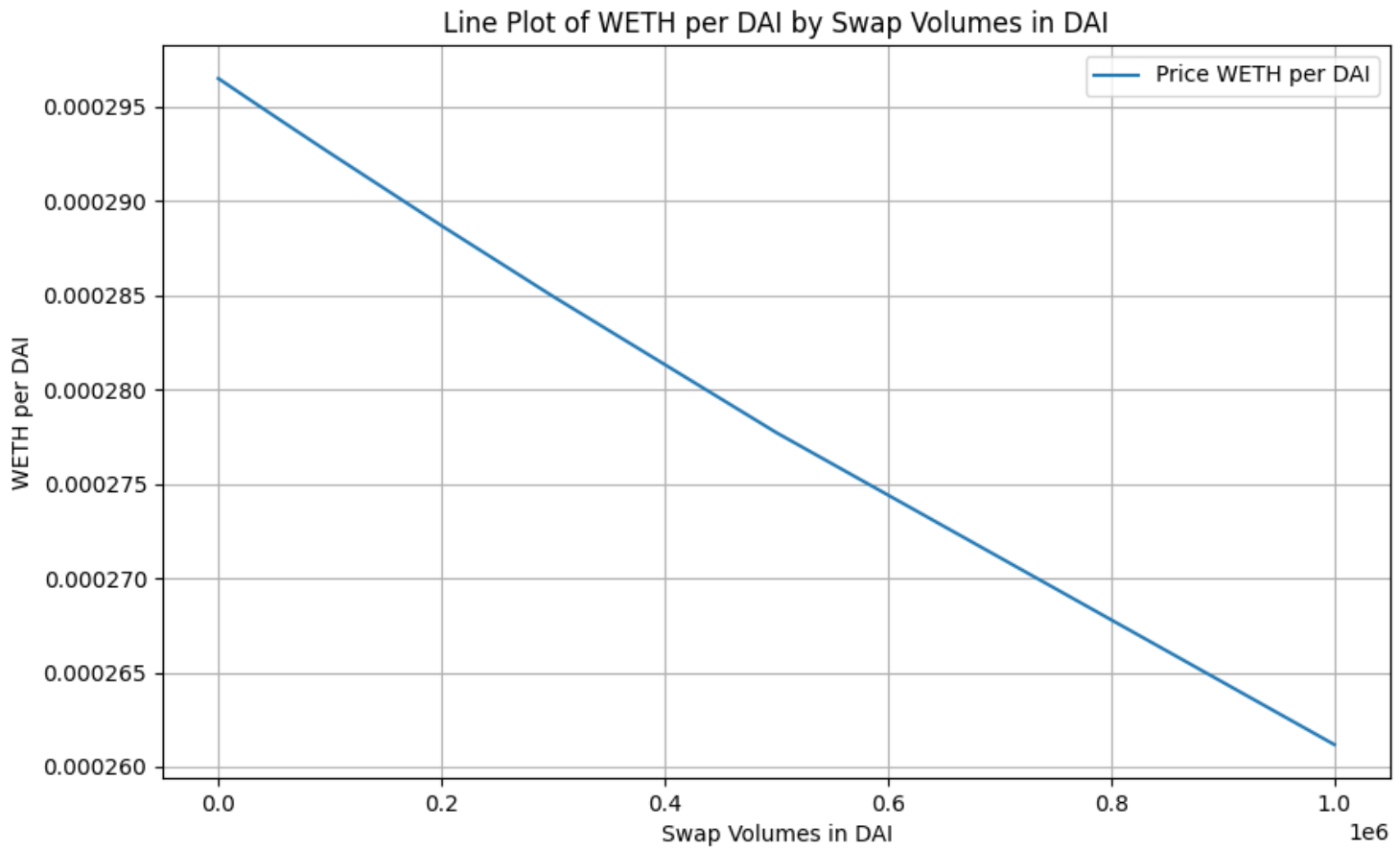
- 3つのケースのシミュレーションを行う.

- プールの量を一定にしたときの取引ボリュームによる価格の変化を調べる.
  - あまりにもたくさんのトークン量を一度にスワップするとプールの量が足らなくなり，適切な価格での取引ができなくなるはず.
  - DAI-WETHの通貨ペアの実際のプールの量を入れてみる.
    - https://etherscan.io/address/0xA478c2975Ab1Ea89e8196811F51A7B7Ade33eB11#code
    - https://www.geckoterminal.com/eth/pools/0xa478c2975ab1ea89e8196811f51a7b7ade33eb11
    - プールされている量は
      - DAI： 7375421.31916，WETH: 2193.257958598

- プールの量によって，通常の取引でどれだけ価格が変化するのかを調べる
  - 1回のSwap量は正規分布からのサンプリングとする
  - プールの量によって価格がどれだけ変動するかを箱ひげ図で図示する.　（サンプル数1,000）

- プールの量によって，LPトークンあたりの価値はどれだけ変化するのかを調べる
  - 上記と同じ分布のTxを仮定.
  - Pトークンの価格がどれだけ変化するかをグラフ化（サンプル数1,000）

Uniswap V2シミュレーション：プールの量を一定にしたときの取引ボリュームによる価格の変化.



プール量：
DAI： 7375421.31916
WETH: 2193.257958598

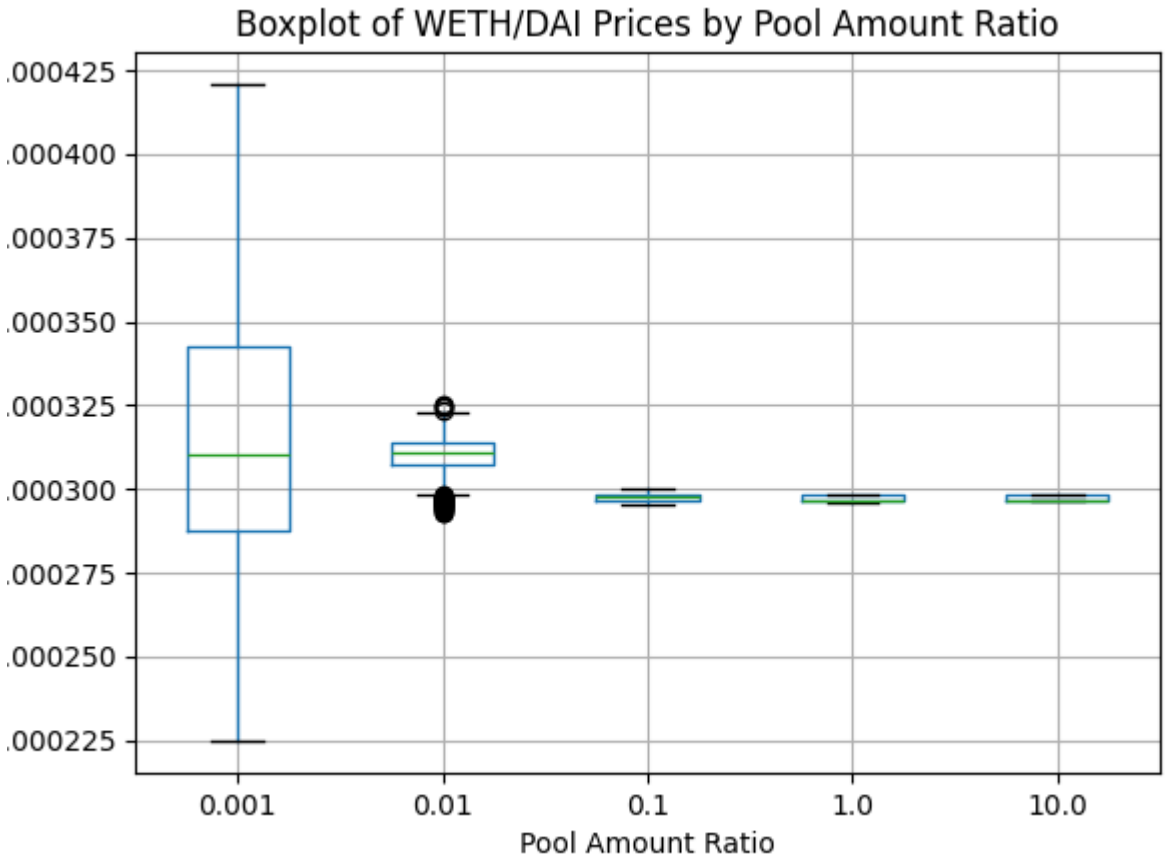
1回のDAI→WETHのSwapを行う.

1 DAIのみをSwapしたときに得られるWETHは約0.000296 WETH.

一方，100M DAIをプールに入れると，1DAIあたり得られるWETHは約0.000261 WETH.

約12%取得できるWETHが減ってしまっている.

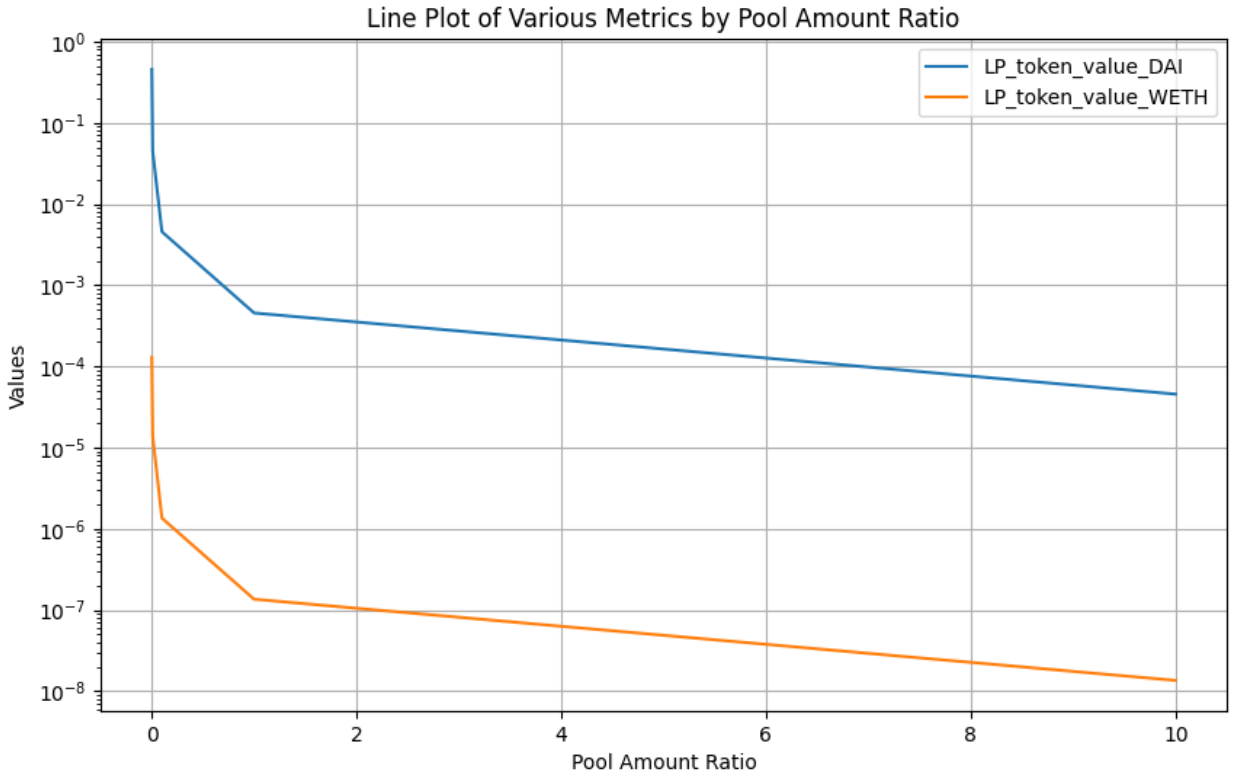
Uniswap V2シミュレーション：プールの量によって，通常の取引でどれだけ価格のブレが生じるのか



プールの量を前ケースと同じWETH/DAIの量を基準として，0.001倍，0.01倍，0.1倍，1.0倍，10倍にした5つのケースを考える．
このとき，Swapが1000回発生したときに価格がどれだけ変動するかを見てみる．
※ SwapはDAI→WETHまたはWETH→DAIが50%の確率で発生．
　取引量は，DAI → WETHの場合は平均100，SD40の正規分布を仮定．（逆の場合は同じレート：平均0.0296, SD0.012）

現在のプール量に対して10分の1くらいまで少なくなったとしてもそこまで価格の変動はないことがわかった．
逆に100分の1，1000分の1くらいまでプール量が減ってしまうとかなり大きく変動してしまい，適正な価格でSwapができないケースが増えてくることがわかる．

Uniswap V2シミュレーション：プールの量によって，LPトークンあたりの価値はどれだけ変化するのか.



前のケースと同様にシミュレーションを行う：
プールの量を，前と同じWETH/DAIの量を基準として，何倍にするかを変化させる.
このとき，Swapが1000回発生したときに価格がどれだけ変動するかを見てみる.
※ SwapはDAI→WETHまたはWETH→DAIが50%の確率で発生.
　取引量は，DAI → WETHの場合は平均100，SD40の正規分布を仮定.　（逆の場合は同じレート：平均0.0296, SD0.012）

最終的な1LPトークンあたりDAI，WETHそれぞれいくらで換金できるのか，をプロット.

プールの量が増えるにつれて，トークン価値が下がっているのが見れる.

運用上の想定される問題シチュエーションとインセンティブ

- プールに対して，取引量が多すぎて価格がブレブレになってしまう
  - → この場合はプール量が少ないため，LPトークンあたりの価値が高い状態.
  - → 取引価格安定のためにはプール量を増やさなければならない，という状況に対して流動性提供者への適切なインセンティブがある.

- プール量は適切だが，市場の価格と乖離があり適切な金額でスワップができない
  - → 市場の価格との差があるということは，その差を利用して利益を出す人がいます．（アービトラージャー）
  - → アービトラージャーが利益がなくなるまでスワップを行うことで，適正価格に落ち着く.
  - → 市場の価格との乖離がある場合に，アービトラージャーに対して適切なインセンティブがある，といえる.

GitHubのソースコードを見てみよう

- Uniswapに代表されるDappsは透明性を求めるためソースコードが公開されているケースが多い.
- https://github.com/Uniswap/v2-core
- https://github.com/Uniswap/v2-periphery

  - ※ Uniswapのリポジトリを見ていてv2-coreだけかな？と思って見ていましたが「How UniswapV2 Swaps Work, Alex R. Mead, Coin Metrics, 2023」にv2-peripheryも使っている旨が記載されておりました.
  - ※ GitHubのサイト上でコードを読むのも簡易的でいいですが，しっかり読む場合はローカルに取得し，コメントを追記しながら理解をしていくのをおすすめします.

- 流動性提供用の関数addLiquidity() (v2-periphery内)を見てみましょう.

Uniswap V2まとめ

- 非中央集権で実現するための各プレイヤーへのインセンティブ
  - 取引する人に加えて，流動性提供者へのインセンティブ

- シミュレーションによる分析
  - Uniswappyというツールを使っての分析が可能

- GitHubリポジトリ
  - ソースコードを見て流動性提供のロジックが確認できた.

ステーブルコインDAI (SAI)の仕組み

MAKER DAI

- 暗号資産担保型ステーブルコインDAI
- MakerDAOによって運営
- 単独担保（WETHに担保が限定）のSAI（昔のDAI）
- 複数担保（WETHだけでなく多くのトークンを担保に可能）のDAI（今のDAI）
- 単独担保型から複数担保型に(2019/11)
  - 単独担保時代のDAIはSAIと名前が変わった.
  - 移行期間を経て移行
    - https://blog.makerdao.com/single-collateral-dai-to-multi-collateral-dai-upgrade-timeline-and-actions/
  - DAI価格：https://www.coingecko.com/ja/%E3%82%B3%E3%82%A4%E3%83%B3/%E3%83%80%E3%82%A4
  - SAI価格：https://www.coingecko.com/ja/%E3%82%B3%E3%82%A4%E3%83%B3/%E3%82%B5%E3%82%A4

- 今回は比較的仕組みがシンプルなSAIの紹介をします.

MakerDAO SAI 概要

- WETH（ETHのERC20版で等価）をコントラクトにロックして，それを担保にステーブルコインSAIを発行.
  - 発行SAIに対応する担保となるWETHのプールを作る.
  - 誰でも自分自身のプールを作れる.
  - ※ 正確にはWETHと交換できるPETHのプールになります.

- SAIはコントラクトに返却すると1USD分のWETHを取得できる.
  - それにより1 SAI = 1 USDをキープしている.
  - ※ 正確には手数料の支払いが上乗せして生じる
  - ※ WETHを取得できるのはプールの作成者のみ.

- WETHは価格変動リスクがあるため，担保の総価値（USD）が発行しているDAIの総価値（USD）を下回らないような仕組みが導入されている.
  - 発行DAIの150%のWETHが必要
  - もし担保資産価値が150%を下回った場合は精算され，そのプールの管理者は損をする.
  - ※ 正確には担保資産はWETHではなくPETHです

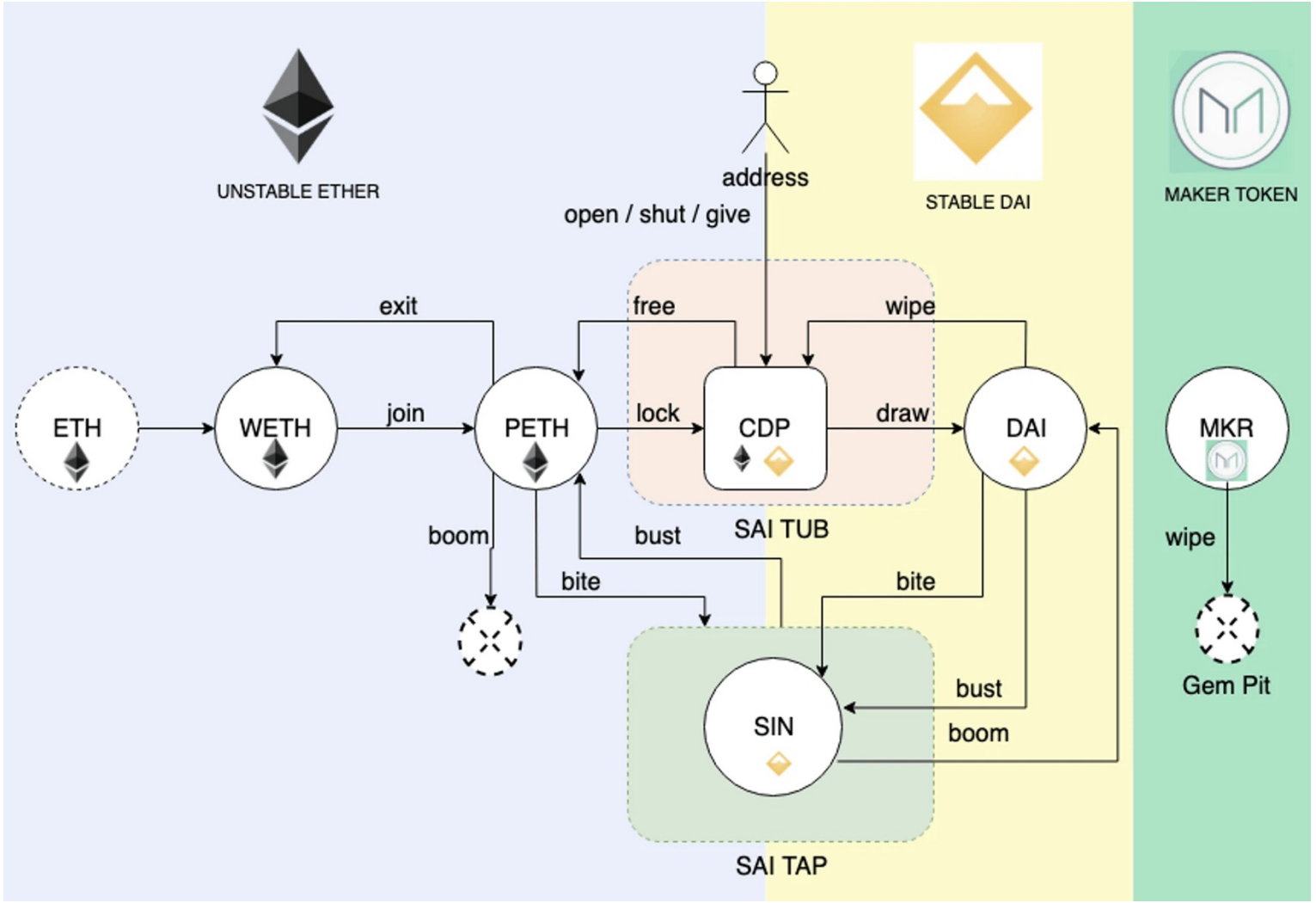
MakerDAO SAIのプレイヤー

- ユーザー
  - SAI発行者・利用者：
  - プール（CDP, Collateralized Debt Positions）をつくるとそのプールに入れたWETHに応じたSAIが取得できる.

- キーパー：
  - 担保資産が150%を切ったCDPを発見し，それを精算する人
  - 精算すると儲けが出るようなインセンティブがある.

- （価格オラクル）
  - WETHの市場価格をオンチェーンに知らせる役割.

全体像



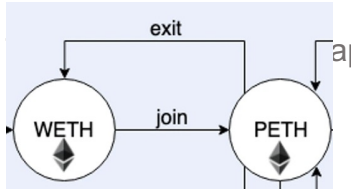
https://medium.com/coinmonks/makerdao-tokens-explained-dai-weth-peth-sin-mkr-part-1-a46a0f687d5e

WETHとPETHの交換

- SAIでは，担保資産として直接ETHを使うのではなく，WETHを経由したPETHを使用します.

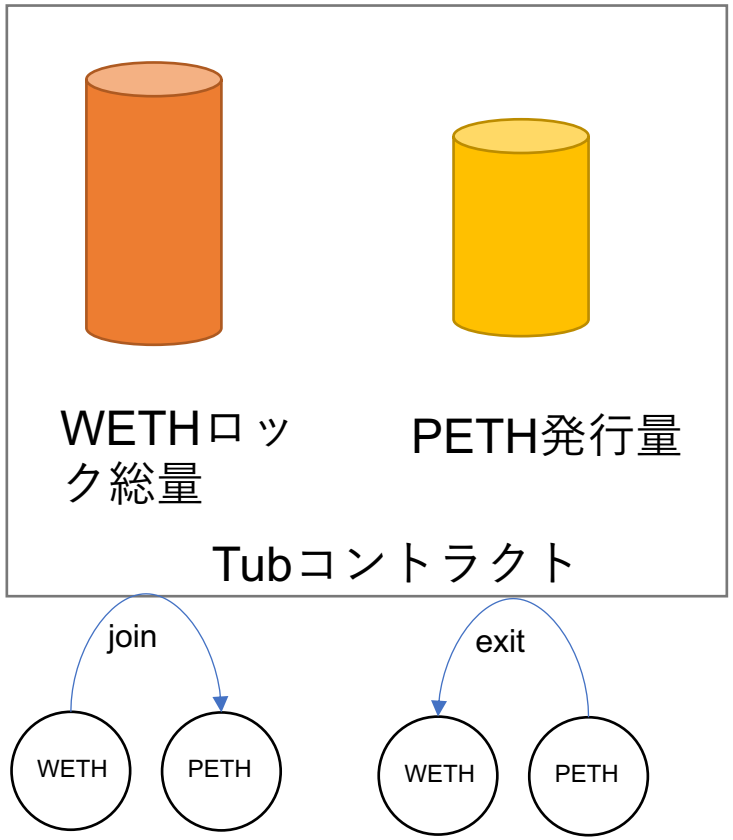
- WETHの役割
  - ERC20互換のETH
  - ETH直接では取り扱いにくい部分があり，ERC20のほうが高機能.

- PETHの役割
  - CDPが破綻したとき，プール内の資産の思わぬ増減が発生する.
  - そのときの緩衝材の役割を果たす.



WETHとPETHの交換: join

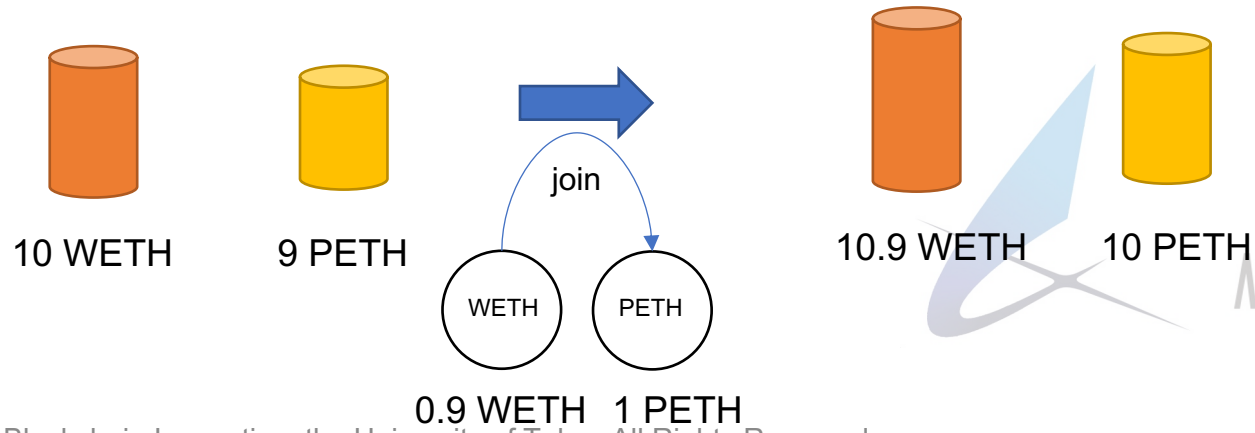
- 基本的に1 WETH = 1 PETHだが，CDPの精算が発生するとPETHの総量が変動する．

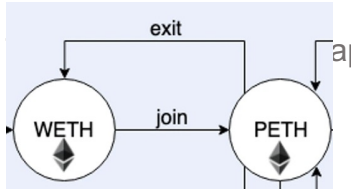


join

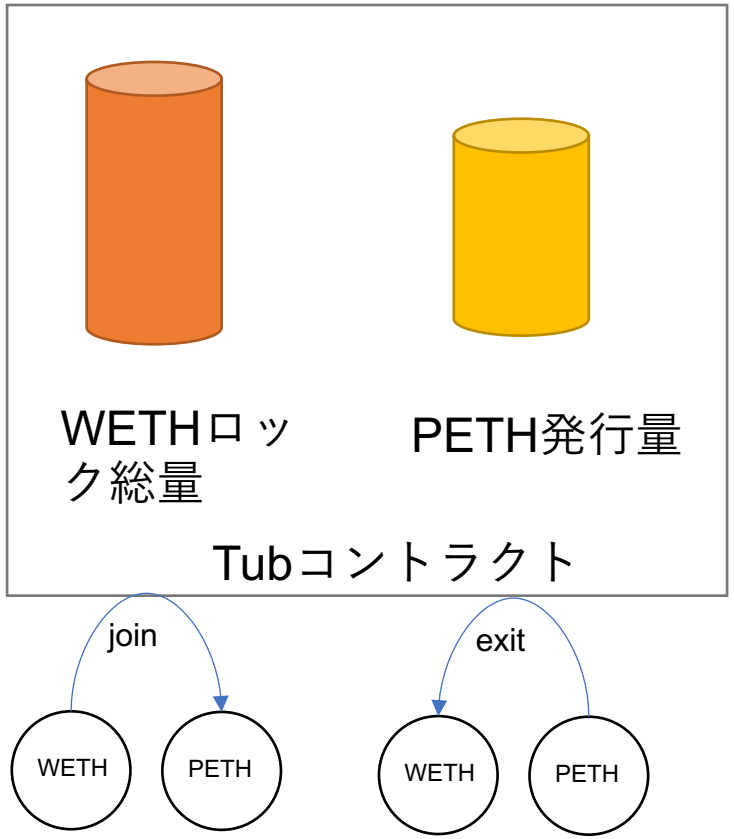
WETHをプールに入れてPETHを受け取れる関数．
このときPETHは新しく発行（ミント）される．
PETH発行量を指定し，WETHロック量とPETH総発行量の比に応じたWETHを新たにロックする必要がある．必要なWETHは，WETH/PETHの比が1になる方向で調整される．

例） WETHのロック総量が10WETH，PETHの総発行量が9WETHのとき：
1 PETH取得するときには 1 * (9/10) = 0.9 WETHロックする必要がある．
join後はWETHロック総量：10.9, PETH総発行量：10 となる．





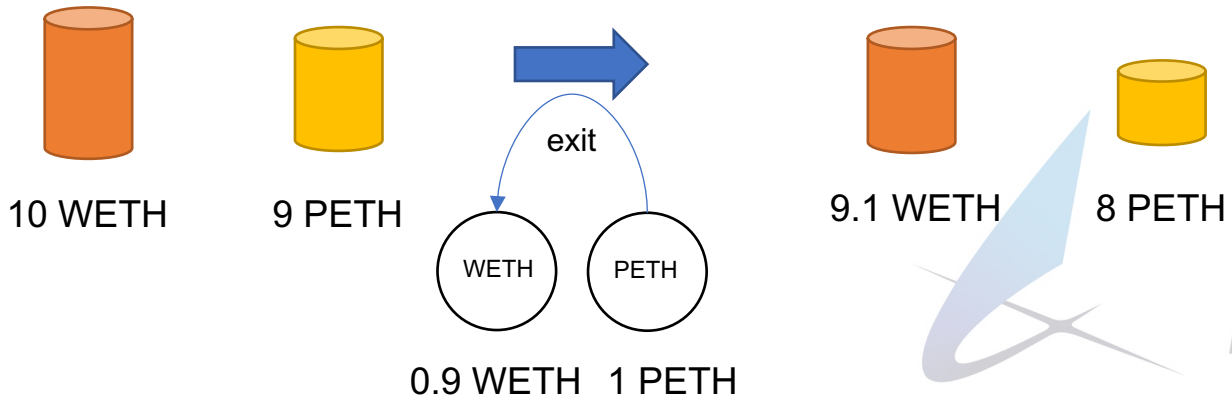
WETHとPETHの交換: exit



exit

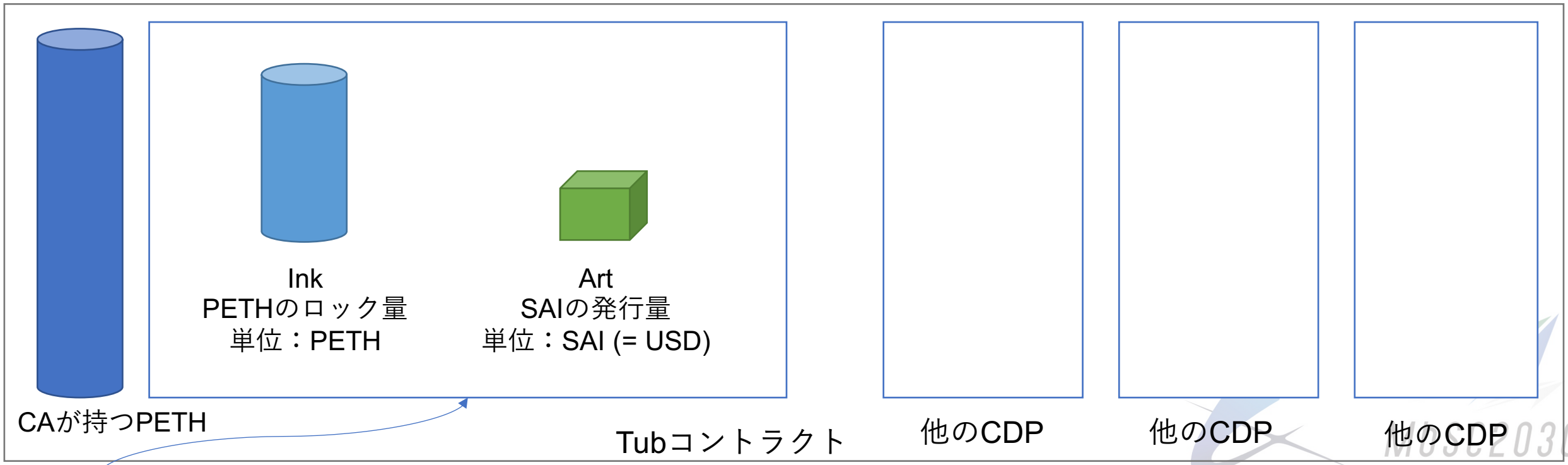
逆にPETHをバーンして，WETHを受け取れる関数.
join時と同じ比率のWETHが取得できる.
PETHは，発行量が少ないときは受け取れるWETHが少なくなるので，バーンする行動にブレーキが掛かる.

例） WETHのロック総量が10WETH，PETHの総発行量が9WETHの場合：
1 PETHバーンすると 1 * (9/10) = 0.9 WETH受け取れる.
exit後はWETHロック総量：9.1, PETH総発行量：8 となる.



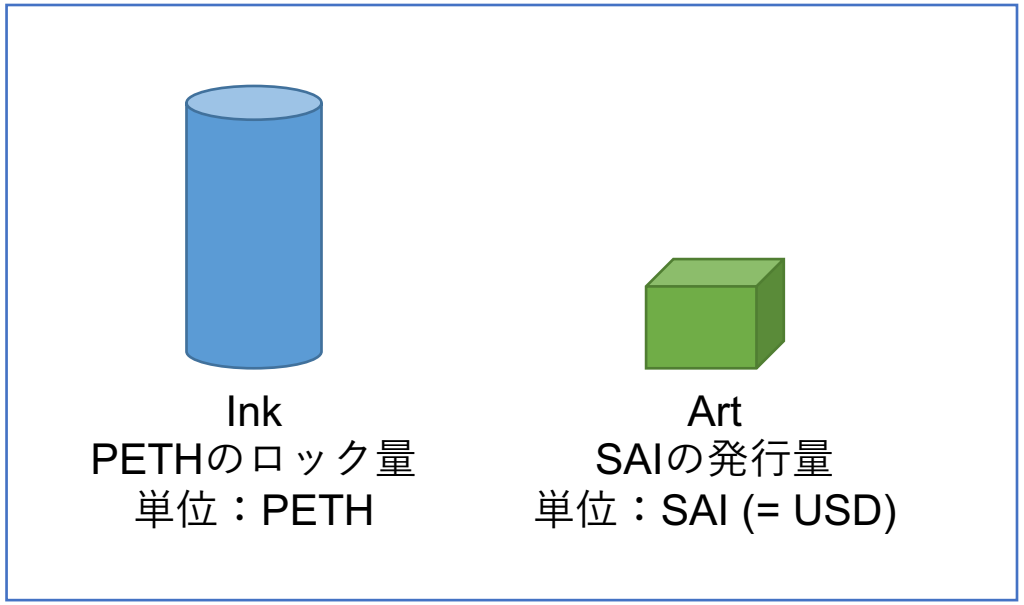
CDPを作る

- CDPは誰でも作ることが可能.
  - PETHを担保として，SAIを借りられるシステム（時間とともに手数料（金利）が増える）
- Tubと呼ばれるコントラクト内の変数としてCDPを作る.
  - 担保資産はこのコントラクト内にすべてのCDPの担保がまとめてロックされる.



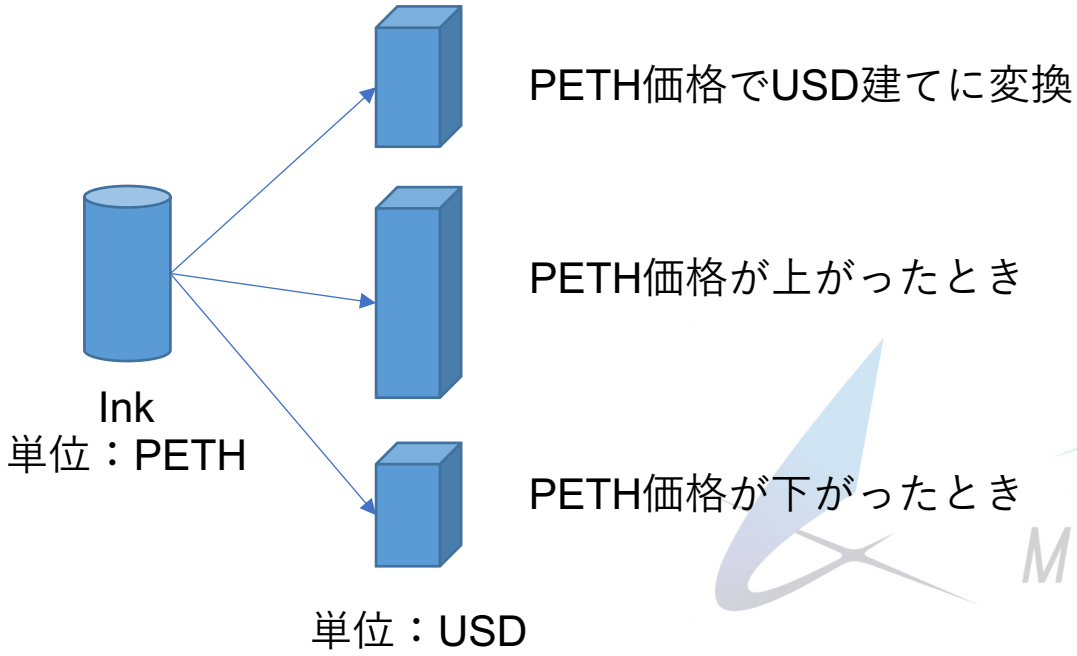
ユーザーは自分のPETHをもとにCPDを自由に作れる

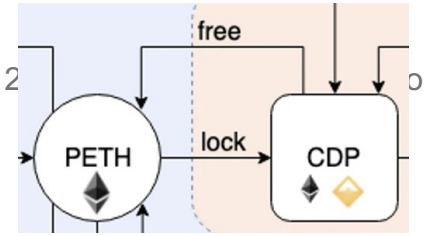
CDP内のInkとArt



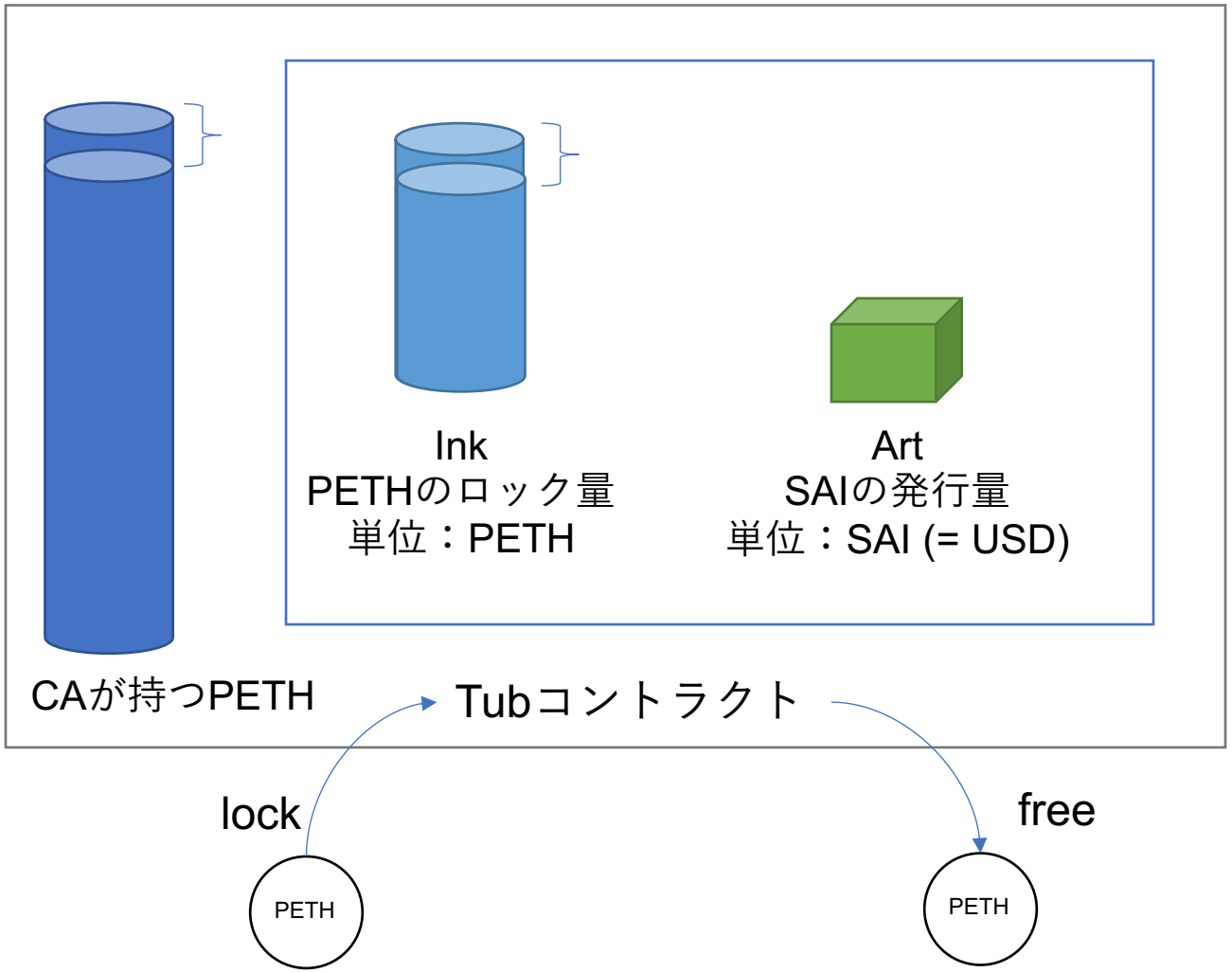
Artの150%以上の価値のInkが必要.
CDPからSAIを取り出すと，その分Artが増える.
また，WETHの価格変動によりPETHの価値も変動する.
※ WETH価格はオラクルにより外部から取得.

USD建てで比較したときにInkはArtの150%以上の価値が必要.





PETHをCDPに入れる，出す: lock, free

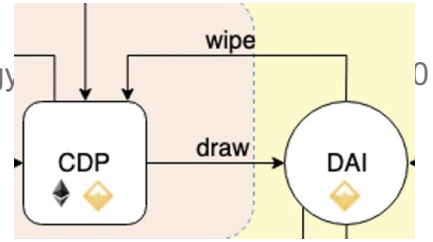


lock

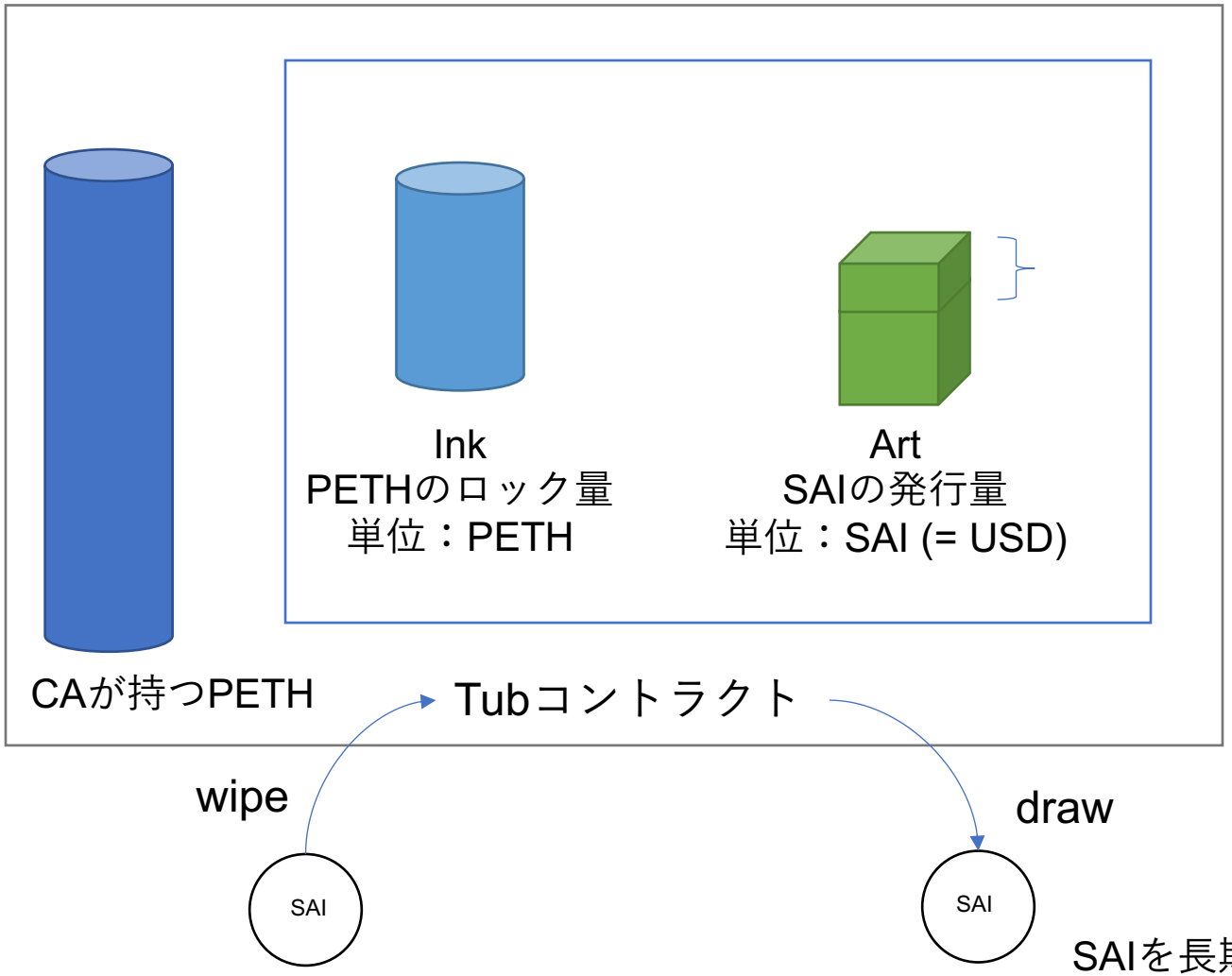
PETHをコントラクトに送金し，該当のCDPのinkを増やす．
lockするのはCDPの持ち主でなくても誰でも可．

free

PETHをCDPから取り出す．
freeはCDPを作った人だけが実行可．
CDPから取り出す際，Inkの総価値がArtの150%を下回ってしまう場合は実行不可．



CDPからSAIを発行・バーン



draw

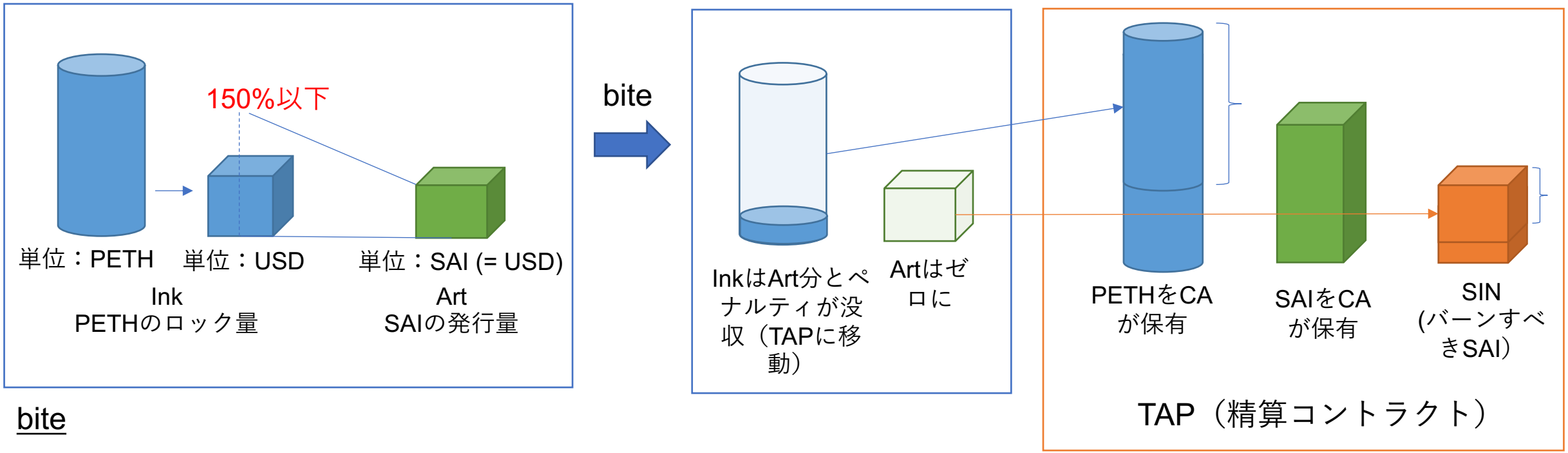
ステーブルコインSAIを受け取る.
このときのSAIは新規発行される.
SAIの発行分，Artを増やす.
このとき担保率が150%を下回るようになる場合は実行不可.
drawはCDPの持ち主しか実行できない.

wipe

SAIをCDPに戻す.
このときArtは減り，戻されたSAIはバーンされる.
戻す際に，SAIに応じた手数料をMKRトークンで支払う.
※ 手数料率はガバナンスによって決定される.
※ 発行している期間に応じて手数料が増える.
※ MKRトークンはMakerDAOのガバナンストークン.
※ 支払われたMKRトークンはバーン用アドレスに.
※ MKRトークン価格は外部（オラクル）から取得.

SAIを長期保有していて手数料が増えたり，CDPをやめたいとき，ETH価格下落により担保割れしそうな場合に，SAIをここに戻すインセンティブが働く.

精算をする: bite

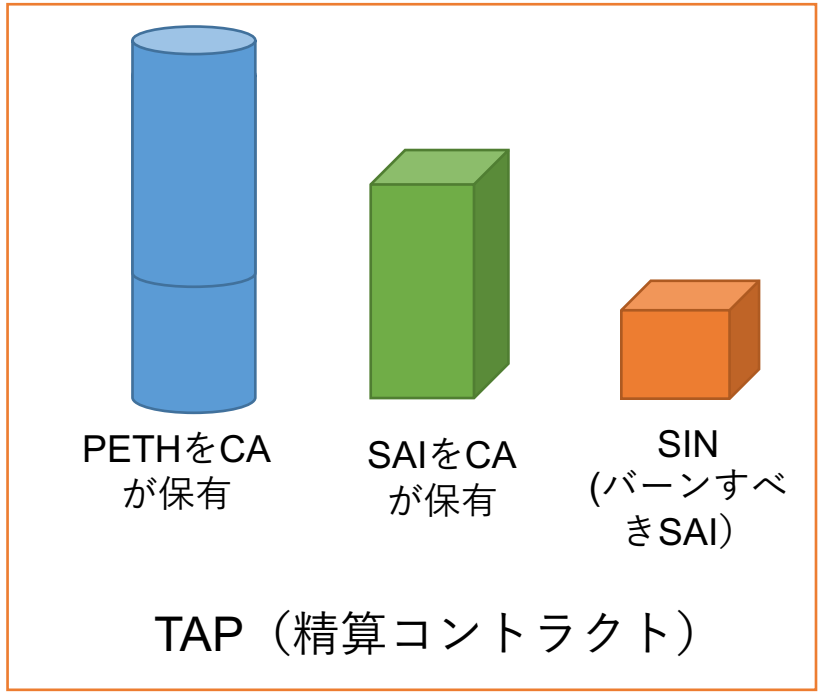


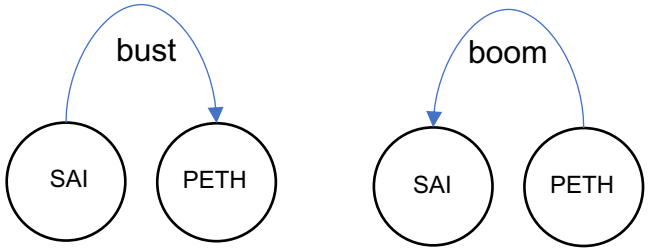
bite

- CDPは，担保割れを起こしているとき精算が可能.
- 精算処理は誰でも実行可能.
- 精算処理が走ると，そのCDP内に残存しているPETHの一部は清算コントラクト（TAP）内に移動される.
  - Artにペナルティ率13%が加算された分のInkが減り，該当のPETHはTAPに移動される.
  - Artはゼロになり，TAP内のSINがその分増やされる.
  - Ink量が足らない場合は全InkのPETHがTAPに移動される.
  - 残存Inkがある場合は，担保割れしていないCDPとして通常運用に戻る.

- 対応するSAIについては市場に出回っているため直接没収はできない.
- 代わりに，TAP内にバーンすべきSAIの量として記録されているSINを追加させる.

TAP（精算コントラクト）について





heal

同量のSINとSAIをバーンする.
どちらか少ない方の総量に合わせてバーンされる.
bustの終了時，boomの開始時に実行される.

bust

市場からSAIを回収して，代わりのPETHを渡す.
SAIをできるだけ回収し，SIN分だけバーンすることが目的.
精算されたCDPが発行したSAIの回収を促すため換金率が良い（+3%)
渡されたSAIに対してPETHが足らない場合はその分ミントされる.
※ この場合，heal後SAIが余るような場合は実行不可.

boom

PETHを受け取ってSAIを渡す.
レートが良く，SINと相殺されてもまだ残っているSAIを市場に流す目的がある.
受け取ったPETHはTAP内に貯まるのではなく，バーンされる.
※ 基本的には使用されない？

CDPを作りSAIを発行するインセンティブ

- 特に手数料収入も得られず，担保率も150%も必要なSAIを発行するためのCDPを開設する人にはどういうメリットがあるのか？

- 発行されるSAIはロックしているWETHに加えて使用可能になる.
  - 例えばETH価格がどんどん上がる，と考えている人はそのETHをロックした上でその66%にあたるSAIを発行．そしてそのSAIを使って市場からETHを購入することで166%のETHを保有・運用することが可能.

- ステーブルコインを取得・使用したい
  - Uniswap等のDEX，CEXでの流動性が低い時代の頃は，CDPを開設しないと必要な量のSAIを取得することはできない.
  - 決済などでステーブルコインのニーズがあり，そして特定事業者へのトラストなしの暗号資産担保型のステーブルコインを使いたいニーズがあった，と考えられる.

担保割れCDPを精算するインセンティブ

- TAPは，換金率がよい．
- 担保割れを起こしているCDPを精算することで，CDP内のPETH（ペナルティを引いた分）がTAPに移行される．
- つまり，その直後に換金率のよいTAPで手元のSAIをPETHに交換可能．

- また，PETHをWETHに交換する際は，ペナルティ分のPETHがバーンされているため，PETHの総量がWETHのロック量より少ない状態になる．
  - exit()ではPETHのバーンを防ぐ方向に交換レートが動くので，WETHへの換金率は下がる．
  - 中長期的に保有していることでPETHとWETHの釣り合いが取れればWETHへの交換をする．
  - join()でWETHをPETHに交換するレートはよくなっているので，その分をさらにWETHを入れてPETHを受取り，釣り合いを取れるようにすることでexit()でWETHを取得するときの換金率は1：1で可能に．

ユーザーがCDPを作りたい，というニーズがあることを前提に，システム上必要な処理「精算」をおこなうためのインセンティブが設計されている．

SAIの市場価格について

- SAIは同額のPETH（で最終的にはWETH）とコントラクトで交換できるとはいえ，手数料がかかる．
  - しかも，SAIを市場で購入した人は自分でCDPを運用していないため，コントラクトから対価としてのPETHを取り出すことはできない．

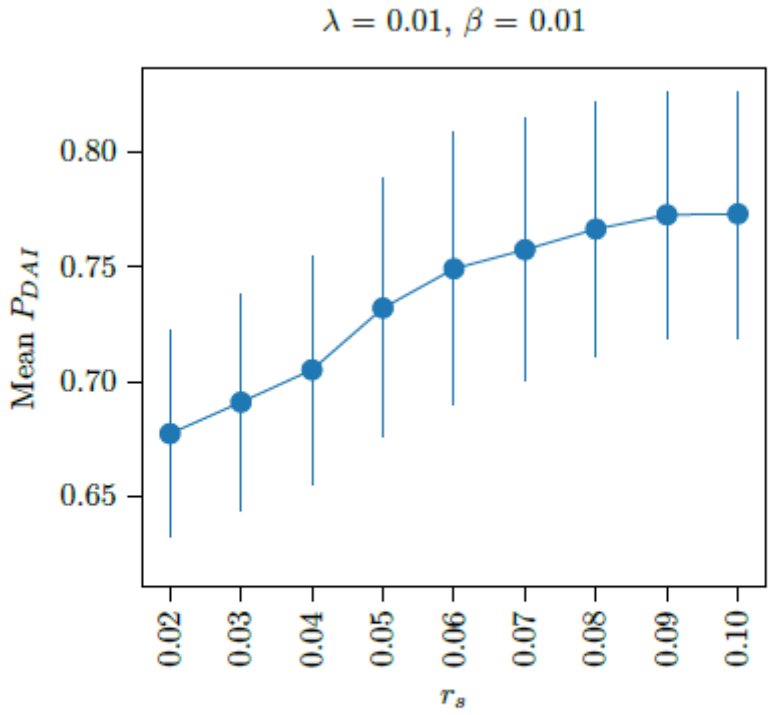
- SAIの市場価格はどのように1ドルになるように調整されているのか？
  - 手数料はガバナンスにより調整できる．
  - SAI価格が1ドルより低いとき：手数料を引き上げる．
    - SAIをコントラクトに戻すインセンティブが働く．
    - つまり，市場におけるSAIの流通量（供給量）が減る．
    - 需要がそのままの場合は，供給量の減少により価格は上がる．
  - 1ドルより高いとき：手数料を引き下げる
    - 上とは逆のことをして，供給量を増やし価格を下げます．

シミュレーション分析

- DAISIMというシミュレーションツールがある.
  - https://github.com/ANRGUSC/DAISIM
  - しかしながらこのツールはうまく動かせなかったので，DAISIMを使って分析を行った論文の一部の結果紹介をします.

- Bhat, Shreyas, et al. "DAISIM: A Computational Simulator for the MakerDAO Stablecoin." *4th International Symposium on Foundations and Applications of Blockchain 2021 (FAB 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
  - https://anrg.usc.edu/www/papers/DAISIM.pdf
  - 投資家は4つの資産を保有（USD，WETH，SAI，PETH）し，自身のリスク嗜好をもとにした行動をシミュレーション，SAIの市場価格などが導ける.
    - 論文中では*USD,ETH,DAI, cETHと表記*
  - 投資家行動はMarkowitz’s Optimal Portfolio Theoryでモデル化.

シミュレーション分析：手数料によるSAI価格の変動，Bhat2021から抜粋



(a) Mean P_{DAI} vs. r_s.

Figure 6 Mean P_{DAI} vs. r_s & β.

P_DAI: DAI（SAI）の市場価格.
r_s: 手数料

手数料が上がるにつれて，SAIの取引価格が上がっていくことが確認できる.
→ 手数料を上げるとSAIの発行量が減る（新しくCDPをつくる人が減る）
→ 市場で流通するSAIの量が減る
→ 需要がそのままで供給量が減ることになるため，価格は上がる.

GitHubのリポジトリ

- https://github.com/makerdao/sai

- draw()を見てみましょう.

SAIのまとめ

- 想定参加者，インセンティブ：
  - ユーザー
  - キーパー

- シミュレーション分析：
  - 実際にシミュレーションの実行はできなかったが，論文の事例を紹介

- GitHubのソースコードを読む：
  - SAIの発行draw()のロジックを概観した

参考URL

- Uniswap
  - https://docs.uniswap.org/contracts/v2/overview
  - https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf
  - https://uniswap.org/whitepaper.pdf
  - https://coinmetrics.io/wp-content/uploads/2023/03/uniswapv2_howSwapsWork-1.pdf
  - https://note.com/prism_crypto/n/n26f6fb4c156c
  - https://zenn.dev/heku/books/77d86a66359561/viewer/6620d6
  - https://zenn.dev/decipher/articles/6e4f33d07bc93a
  - https://github.com/defipy-devs/uniswappy

- MakerDAO
  - https://makerdao.com/whitepaper/Dai-Whitepaper-Dec17-ja.pdf
  - https://makerdao.com/en/whitepaper/sai/#pooled-ether-temporary-mechanism-for-single-collateral-dai
  - https://medium.com/coinmonks/makerdao-tokens-explained-dai-weth-peth-sin-mkr-part-1-a46a0f687d5e
  - https://blog.makerdao.com/ja/year_in_review_2020/
  - https://anrg.usc.edu/www/papers/DAISIM.pdf
  - https://hashhub-research.com/articles/2018-11-13-overview-makerdao-and-dai
  - https://note.com/turingum/n/n60d6c3e5f1e5
  - https://individua1.net/makerdao-pooled-ether-liquidation/
  - https://souta-watatata.medium.com/makerdao-dai%E3%82%AC%E3%82%A4%E3%83%89-10357920337f
  - https://github.com/makerdao/sai

- 本スライドの著作権は，東京大学ブロックチェーンイノベーション寄付講座に帰属しています．自己の学習用途以外の使用，無断転載・改変等は禁止します．