



# Enigma Protocol for Secure & Neutral Computation

川口 将司 / Cho Jang Sa  
as Enigma Collective

## ● この資料について

多かれ少なかれ秘密計算を調べた経験のある方々に向けて構成したスライドです。

秘密計算を知らない方々に向けたスライドも用意してあります：

<https://www.slideshare.net/jangsa7/v011-147749337>

産総研(サイバーフィジカルセキュリティ研究センター)の花岡先生のご厚意により、同組織所属の研究員のみなさまに発表させていただきました。



Enigmaとは



Enigma [ ? ]

- Enigmaとは



# Enigma [ sMPC ]

- Enigmaとは



# Enigma [ TEEs ]

- Enigmaとは

Enigma [ sMPC ]

Enigma [ TEEs ]

Enigma [ ? ]

.....

...

- Enigmaとは



Enigma [ sMPC ]

≠

Enigma [ TEEs ]

- Enigmaとは



# Enigma [ ? ]

## <結論>

### ● Enigmaは秘密計算に中立性をもたらす

1. シークレットコントラクトは公開される
  - シークレットコントラクト=Enigmaネットワークにデプロイ・実行される秘密計算が可能なプログラムのこと
  - 真贋はオンチェーンに刻まれた情報から確かめることができる
2. つまりデータにどのような操作がなされるのかが予め明確になる
  - アウトプットプライバシーを守るアプローチを提供する
3. サービスにバイアスがない証拠を示すことが可能である
  - 不自然な条件文が含まれれば、すぐに検知される
  - 計算ノード側で不正を試みても検知されるし、罰金も受ける

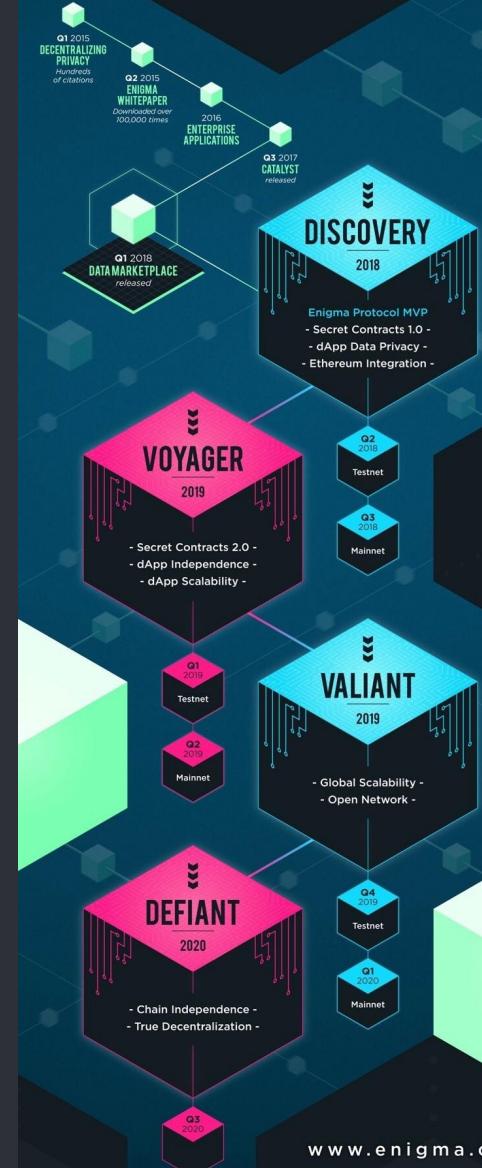
- より概観的なEnigmaの紹介資料(ちょっと前に作成したもの)

<https://www.slideshare.net/jangsa7/v011-147749337>



enigma

ROADMAP  
2018-2020



## ● Enigmaの開発状況ダイジェスト

- 秘密計算技術に”sMPC”を据えて構想されたEnigmaだが、性能問題に直面
- MVP創出のため”TEEs”による実装へ
  - ◆ Intel社とパートナーシップ締結
- 現在はシークレットコントラクトを含めた実装”DISCOVERY”を開発中
  - ◆ 年内リリース？

以降のスライドでは陳腐化しそうな情報に『暫定』をつけている点に注意されたい。

加えてsMPCモードを前提にした情報とTEEsモードを前提にした情報が混在しているため、TEEsモード前提の議論には極力『TEEs』をつけている。

- もくじ

- Enigmaの概要
- シークレットコントラクト
- ユースケース
- (\*) データマーシャルウォレット



● もくじ

- Enigmaの概要
- シークレットコントラクト
- ユースケース
- (\*) データマーシャルウォレット



## <Enigmaの概要>

### Enigmaネットワークにおけるデータとプログラムの配置

#### ★sMPCモードなEnigmaの環境



#### シークレットコントラクト

```
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
    // use the JFrame type until support for  
    // new components is finished  
    JFrame frame = new JFrame("Hello Button");  
    Container pane = frame.getContentPane();  
    pane.add(hello);  
    frame.pack();  
    frame.setVisible(true);  
}  
  
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
    // use the JFrame type until support for  
    // new components is finished  
    JFrame frame = new JFrame("Hello Button");  
    Container pane = frame.getContentPane();  
    pane.add(hello);  
    frame.pack();  
    frame.setVisible(true);  
}  
  
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
    // use the JFrame type until support for  
    // new components is finished  
    JFrame frame = new JFrame("Hello Button");  
    Container pane = frame.getContentPane();  
    pane.add(hello);  
    frame.pack();  
    frame.setVisible(true);  
}  
  
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()
```

許可

許可



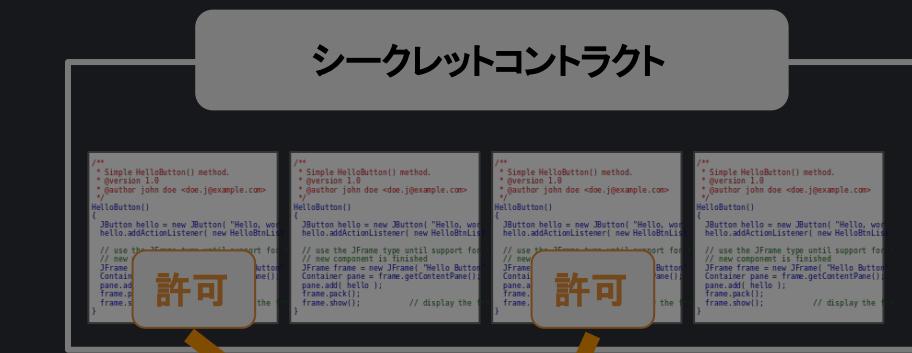
## <Enigmaの概要>

### Enigmaネットワークにおけるデータとプログラムの配置

#### ★sMPCモードなEnigmaの環境



生データ



#### オンチェーン (Layer1)

- ・ブロックチェーン上に刻まれる情報で、改ざんが困難
- ・無対策では刻まれる情報がパブリック
- ・一般的に書き込みが低速である



● <Enigmaの概要>  
Enigmaのオンチェーンに刻まれる情報

シークレットコントラクトの  
Hash値とTaskID  
(今後の実装に依存)



$h(f)$

計算・取引の履歴



署名



計算執行元  
アドレス



手数料

データの所在とアクセス権  
(トラッカー的役割)



計算  
OK

復号  
OK

NG

e.t.c. (実装に依存)



保有残高



難易度

## <Enigmaの概要>

### Enigmaネットワークにおけるデータとプログラムの配置

#### ★sMPCモードなEnigmaの環境

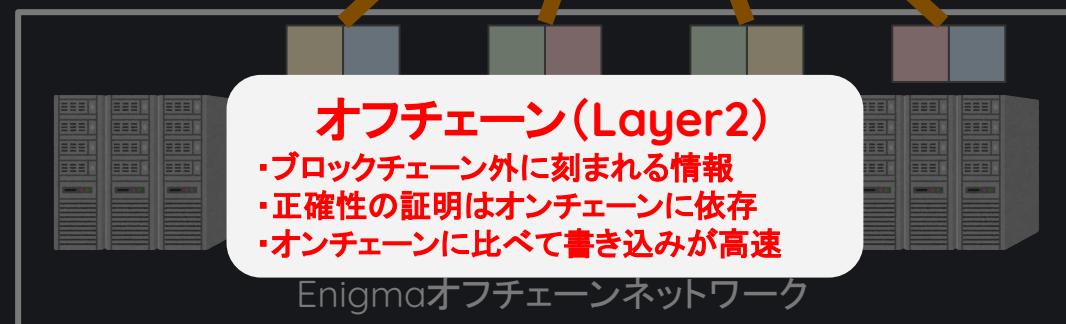


#### シークレットコントラクト

```
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    // use the JFrame type until support for  
    // new component is finished  
    JFrame frame = new JFrame();  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    frame.getContentPane().add(hello);  
    frame.pack();  
    frame.show();  
}  
  
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    // use the JPanel type until support for  
    // new component is finished  
    JPanel pane = new JPanel();  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    pane.add(hello);  
    pane.pack();  
    frame.show();  
}  
  
/** Simple HelloButton() method.  
 * version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    // use the JButton type until support for  
    // new component is finished  
    JButton hello = new JButton("Hello, world");  
    hello.addActionListener(new HelloButtonListener());  
  
    frame.show();  
}
```

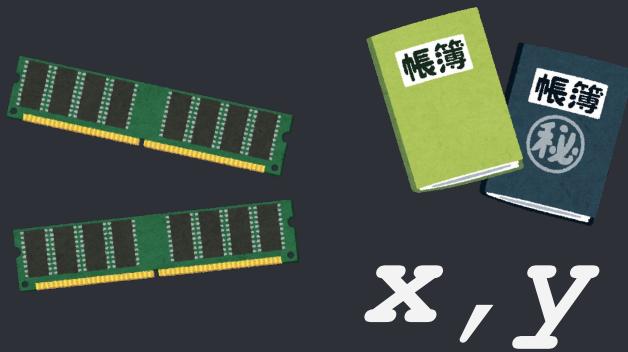
許可

許可



● <Enigmaの概要>  
Enigmaのオフチェーンに刻まれる情報

計算対象データ

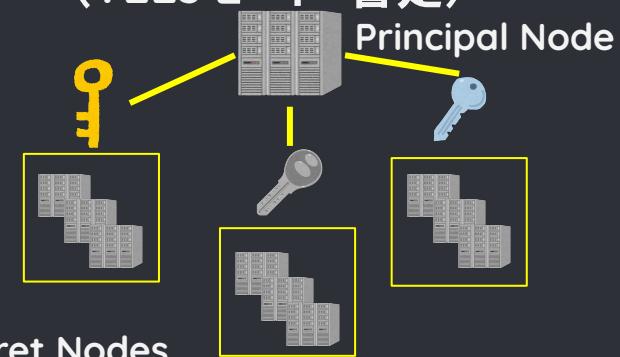


シークレットコントラクト  
(実装に依存)



$f$

グループの共有鍵  
(TEEsモード・暫定)



e.t.c. (一部暫定)



s3/IPFS等の  
リンク情報

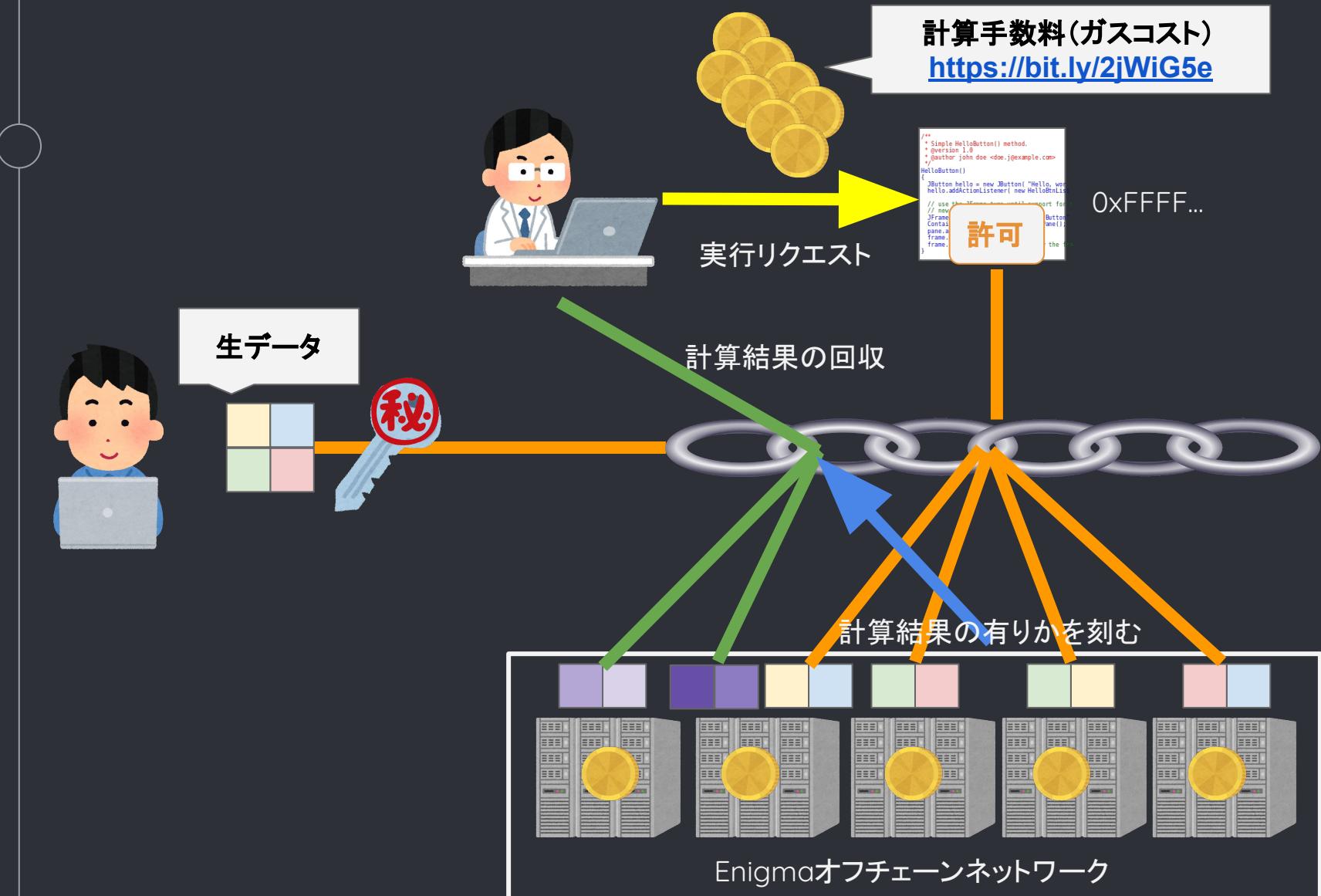


コントラクト毎の  
ワーカー一覧

コントラクトのバイトコード  
あるいはその参照

<Enigmaの概要>

Enigmaネットワーク上の秘密計算、シークレットコントラクトの執行



● <Enigmaの概要>

Enigmaの計算結果が正常に保たれる仕掛け

非中央集権なネットワークでコンセンサスを得るための仕組みは、ネットワークの実装毎に異なる。(例: BitcoinならProof of Work/PoW)

Enigmaでは主にProof of StakeとSlashingに依拠する。

### Proof of Stake (PoS)

ノードを運営するには、一定以上のデポジットが求められる。

デポジットとリソースに応じて、計算の機会(報酬あり)を得やすい。

Stakingされたコイン



### Slashing

背信行為に走ったノードの残高からデポジットを差し押さえる仕組み。

基本的に多数派が勝つため悪玉が多いと善玉が負ける可能性もある。が、現実的には困難でもある。

答:A

答:A

答:B



● <Enigmaの概要>  
Slashingをもっと分かりやすく(1) ~善玉が多数派の場合~



わたしは何に見えますか？

犬！

犬！

猫！



● <Enigmaの概要>  
Slashingをもっと分かりやすく(2) ~悪玉が多数派の場合~



わたしは何に見えますか？

猫！

猫！

犬！



- もくじ

- Enigmaの概要
- シークレットコントラクト
- ユースケース
- (\*) データマーシャルウォレット



● <シークレットコントラクト>

シークレットコントラクトとは

1. Enigmaネットワークにデプロイされたプログラムを指す

- a. Ethereumのスマートコントラクトとの主な相違点は、入力データの秘匿性と完全性を保ちながら計算ができる点
- b. 「アウトプットプライバシーを侵す表現を含んだシークレットコントラクトはデプロイ不可能にする」という実装方針が掲げられている。ただこれを検知可能なパーザが実装可能かは疑問(私見)

2. シークレットコントラクトの内容は公開される

- a. シークレットコントラクトからデータへのアクセス可否は、データ提供者が決めることができる
- b. 手続きが公開されるため、計算結果にバイアスがないことが証明可能である

- <シークレットコントラクト>  
シークレットコントラクトが動くまで  
※DISCOVERY/TEEs実装の暫定版

ここからはシークレットコントラクトの作成から実行終了までの流れを説明する。

現在の実装方針に基づいた情報のため、陳腐化する箇所が多く含まれる点、ご了承いただきたい。

例)

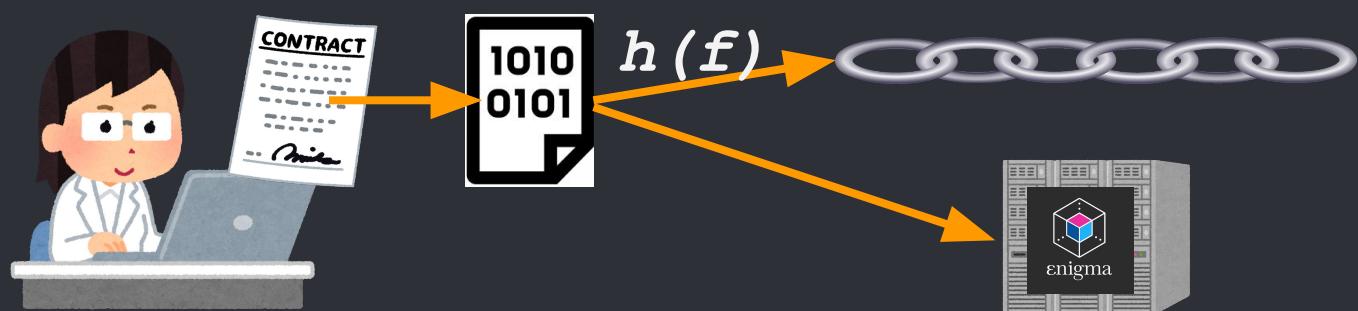
現在Enigmaの合意レイヤーはEthereumのブロックチェーンに依拠しているが、将来的にEnigmaは独自の合意レイヤーを得る予定である

● <シークレットコントラクト>

シークレットコントラクトのデプロイが終わるまで

※DISCOVERY/TEEs実装の暫定版

1. シークレットコントラクトを書く ※Rust言語
  - a. 厳密にはコンパイルしてWASMのバイトコードを生成する
2. シークレットコントラクトのHash値をオンチェーンに刻む
  - a. (2019年7月現在の文脈では)EnigmaはEthereumをオンチェーンとして活用する
  - b. Enigmaは将来的に独自チェーンを持つため時間経過により文脈が変わることもある
3. オフチェーンにシークレットコントラクトをデプロイする

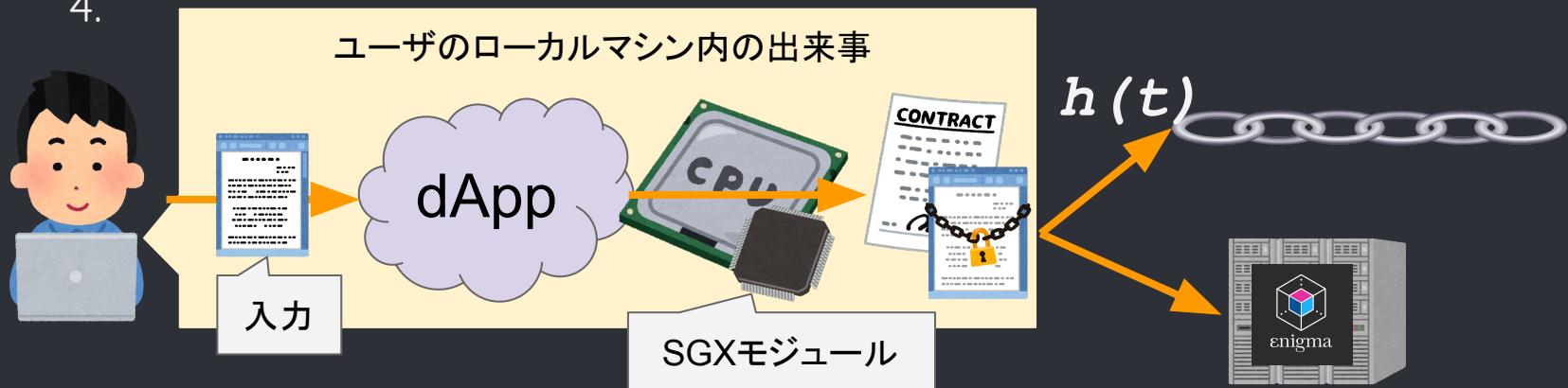


<シークレットコントラクト>

シークレットコントラクトの実行を依頼するまで

※DISCOVERY/TEEs実装の暫定版

- 1. Task(実行対象コントラクトと入力データ他)を生成する
  - a. アプリケーション(dApp)を起動し、入力データを投入する
  - b. このとき入力データに対して、Enigma.jsライブラリによるIntel SGXをつかった命令で暗号化が施される
2. TaskのHash値をオンチェーンに刻む
  - a. このTaskのHash値を”TaskID”と呼ぶ
3. TaskをEnigmaネットワークに送る
- 4.

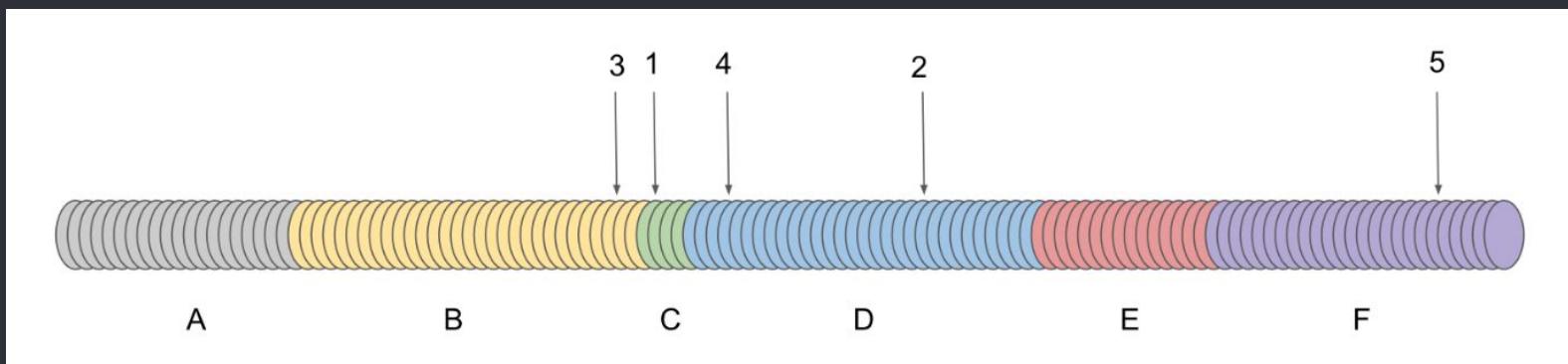


## <シークレットコントラクト>

### シークレットコントラクトの実行が終わるまで(1)

※DISCOVERY/TEEs実装の暫定版

1. シークレットノード群の中から計算担当者(Worker)が選ばれる
  - a. シークレットノード = Enigmaネットワークに参加する秘密計算を担うノードのこと、いわば計算担当者の候補である
  - b. 暫定仕様ではシークレットコントラクト毎にあらかじめ計算担当者がアサインされ、一定期間(Epoch)毎に再アサインされる
  - c. Stakingが多いほど計算担当者として選ばれる可能性が高くなる。つまり計算のお仕事をもらって報酬をもらえる可能性が高い



出典: Kleros whitepaper

<シークレットコントラクト>

シークレットコントラクトの実行が終わるまで(2)

※DISCOVERY/TEEs実装の暫定版

- - 2. 計算担当者(Worker)はEnigmaネットワーク内で計算結果を共有し、計算結果を合意する
  - 3. 所定のEthereumのスマートコントラクトを呼び出す。呼び出された Ethereumのスマートコントラクトは事後処理を担う
    - a. 事後処理はおもに残高の操作
    - b. たとえば計算担当者に対する報酬の支払いなど



<シークレットコントラクト>  
シークレットコントラクトのサンプル



<https://enigma.co/discovery-documentation/SecretContractExamples/>

<Enigmaのまとめ(再掲)>

- Enigmaは秘密計算に**中立性**をもたらす

1. シークレットコントラクトは公開される
  - シークレットコントラクト=Enigmaネットワークにデプロイ・実行される秘密計算が可能なプログラムのこと
  - 真贋はオンチェーンに刻まれた情報から確かめることができる
2. つまりデータにどのような操作がなされるのかが**予め明確**になる
  - アウトプットプライバシーを守るアプローチを提供する
3. サービスに**バイアスがない証拠を示すことが可能**である
  - 不自然な条件文が含まれれば、すぐに検知される
  - 計算ノード側で不正を試みても検知されるし、罰金も受ける

- もくじ

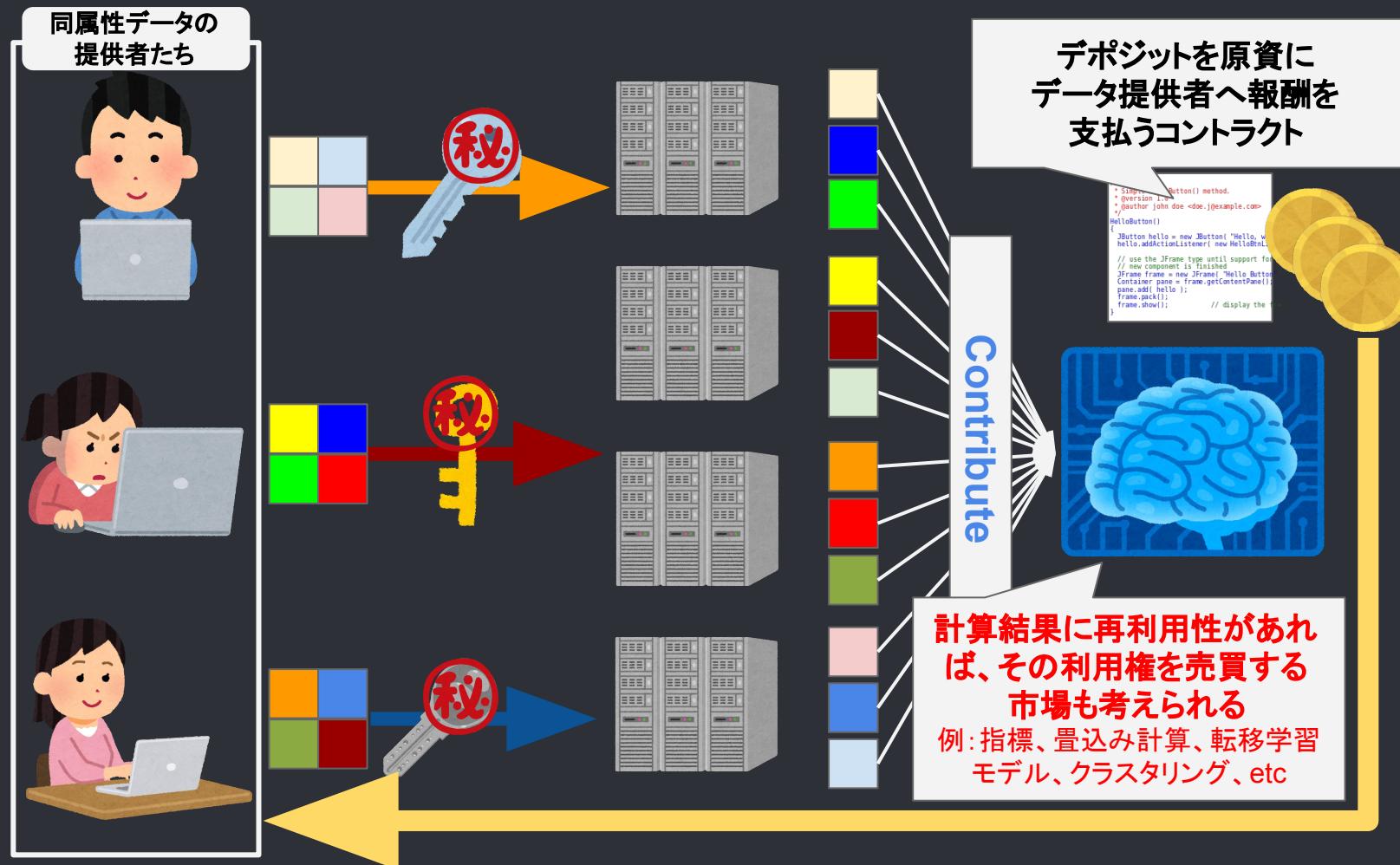
- Enigmaの概要
- シークレットコントラクト
- ユースケース
- (\*) データマーシャルウォレット



<ユースケース>

Enigma Data Marketplace(情報銀行基盤)

「データ提供 -> データ活用 -> 値値の還元」というエコシステムを実現するプラットフォームである。





### <ユースケース>

#### Secret Voting(匿名投票基盤)

投票内容を秘匿した選挙プラットフォームである。

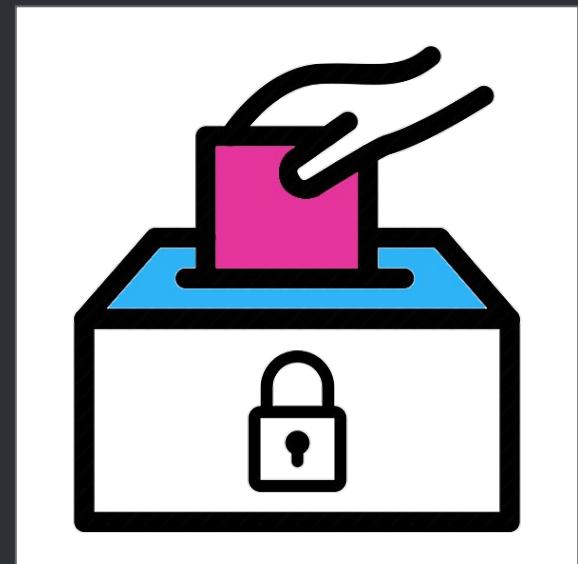
あらかじめ決められたBlock Height(Blockchainにおける時間相当)を迎えると集計プログラムが稼働し、開票結果が明かされる。

ポイントは投票者のプライバシーを守るだけではなく、運営視点でもメリットがある、という点である。

いわゆる選挙違反にあたる「票の買収」を防げるから。

”DAO”など、新しい投票システムの基盤として注目を集めるBlockchainだが、実のところ秘匿性を保つ機構がない限り、公正公平な選挙を開催するには向かない。

やや蛇足だが、そのような脆弱性を突く闇組織のことを“Dark DAO”と呼ぶ。





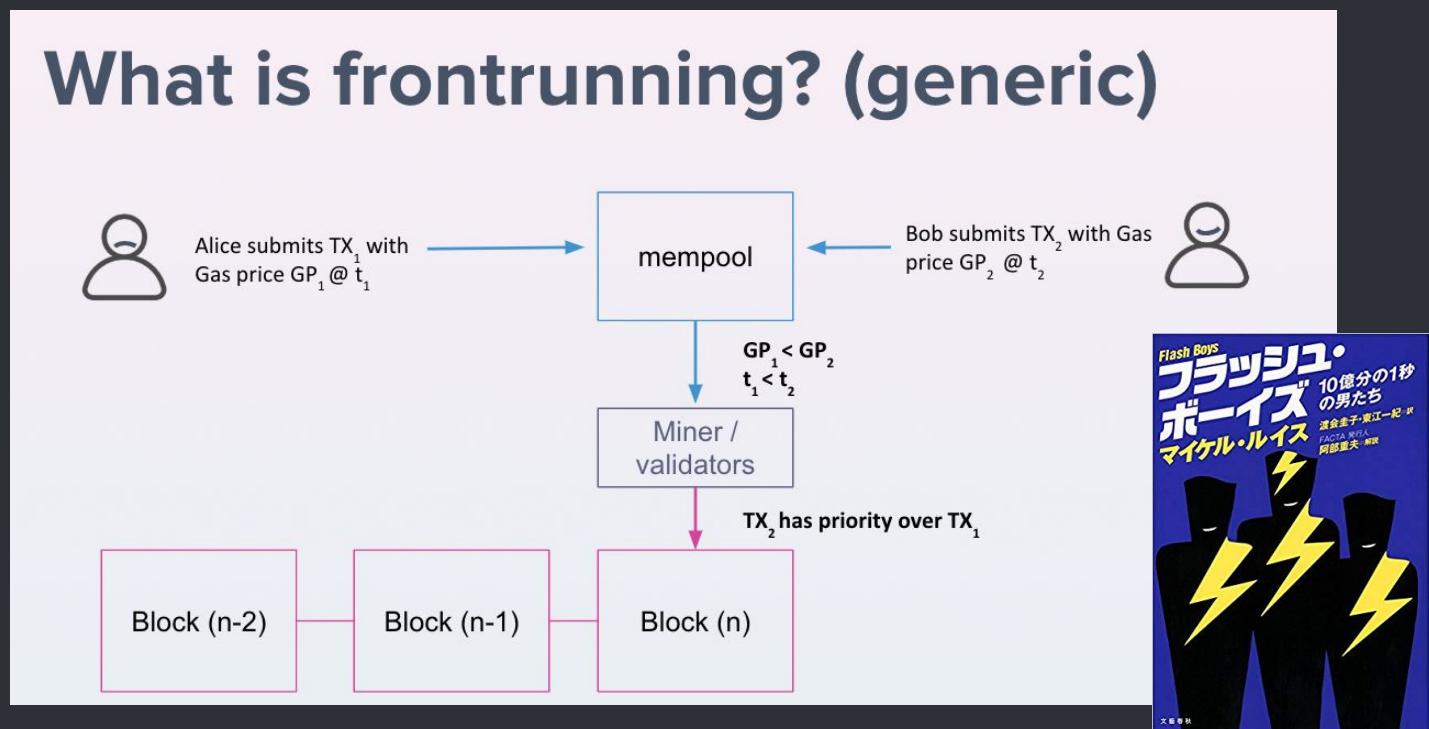
<ユースケース>

## 分散取引所のフロントランニング対策(Anti-FlashBoys2.0)

分散取引所の場合、注文情報をオンチェーンに書き込む速さ(優先度)はガスフィー(ノード運営者に支払う手数料)で稼げるため、従来の中央集権的な取引所よりもフロントランニングを気軽に実行可能である。

<https://arxiv.org/pdf/1904.05234>

Enigmaは注文情報を秘匿するため対策になる: <http://bit.ly/2FKSBjQ>



- もくじ

- Enigmaの概要
- シークレットコントラクト
- ユースケース
- (\*) データマーシャルウォレット



● <消費者視点：データの提供者が負うリスクと権利>

## 企業は消費者のデータを活用して利益を享受している

- データは企業の製品・サービスの魅力向上やマーケティングに活用されるばかりでなく、不正による情報流出の脅威に晒され、政治活動の道具にされることもある
- 多くのサービスは消費者にデータ提供を強制しており、選択の余地を奪われた消費者は自身のデータに対するコントロールを消失している

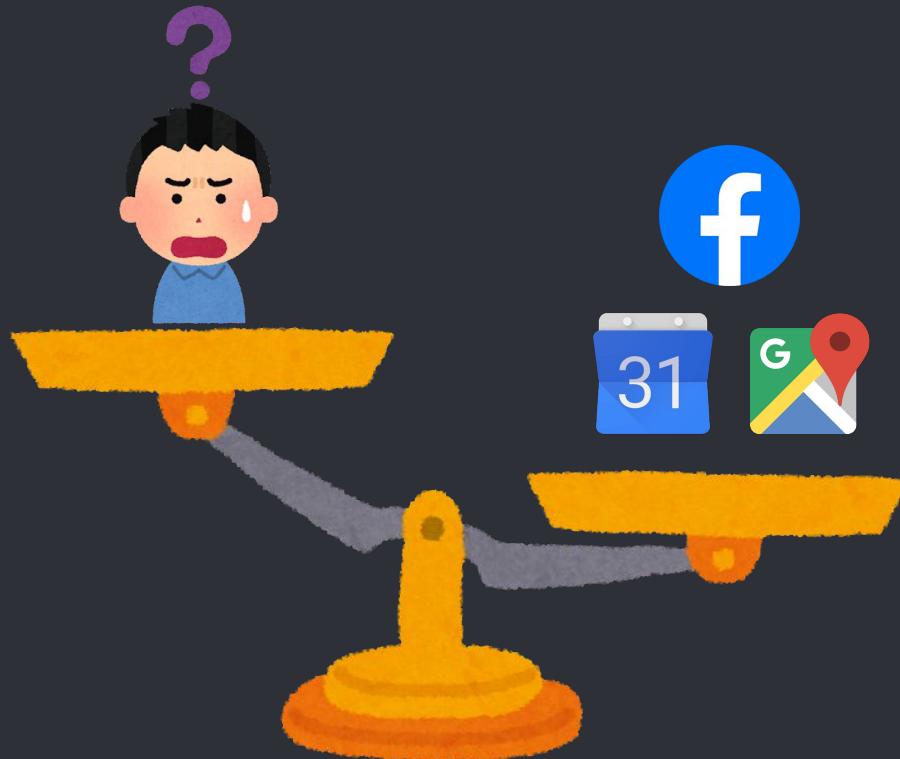




＜補足：企業と消費者のパワーバランス問題＞

『自分より自分に詳しいサービス』が存在しているのは何故か

- 企業と消費者のあいだにデータの保有量および分析能力等のリテラシーに格差があることが原因である（情報の非対称性）

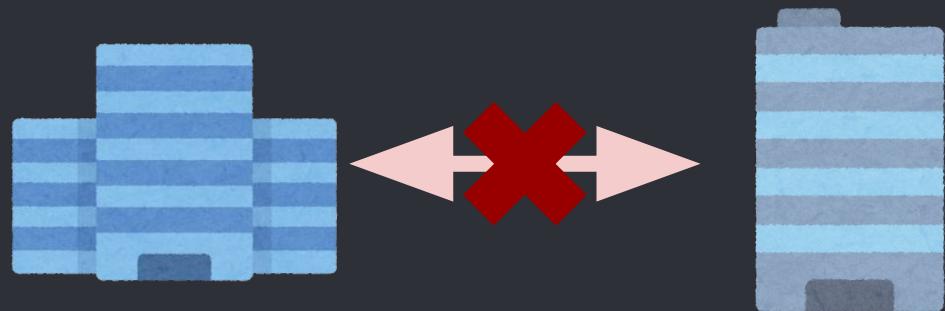


＜企業視点：企業のデータ活用の課題＞

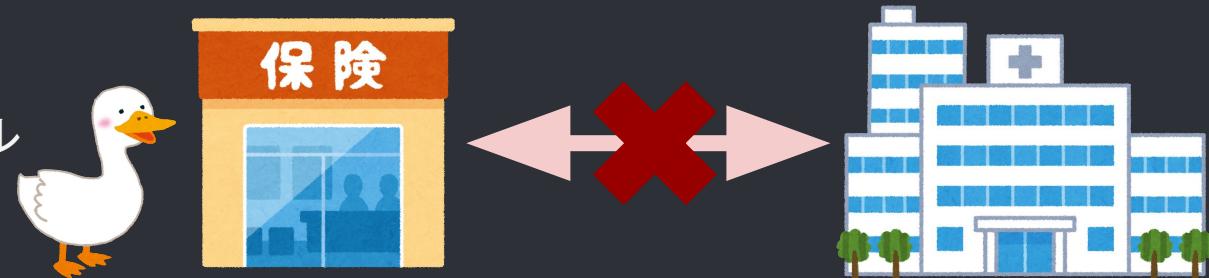
企業同士のデータ共有には様々な障壁が存在する

企業が蓄積したデータのサイロ化は重大な機会損失である

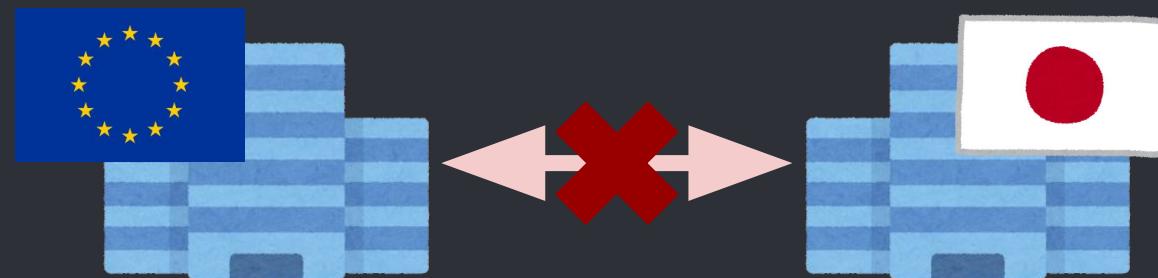
● 消費者の合意  
競合同士の折り合い



組織・業界別のルール



地域毎のルール

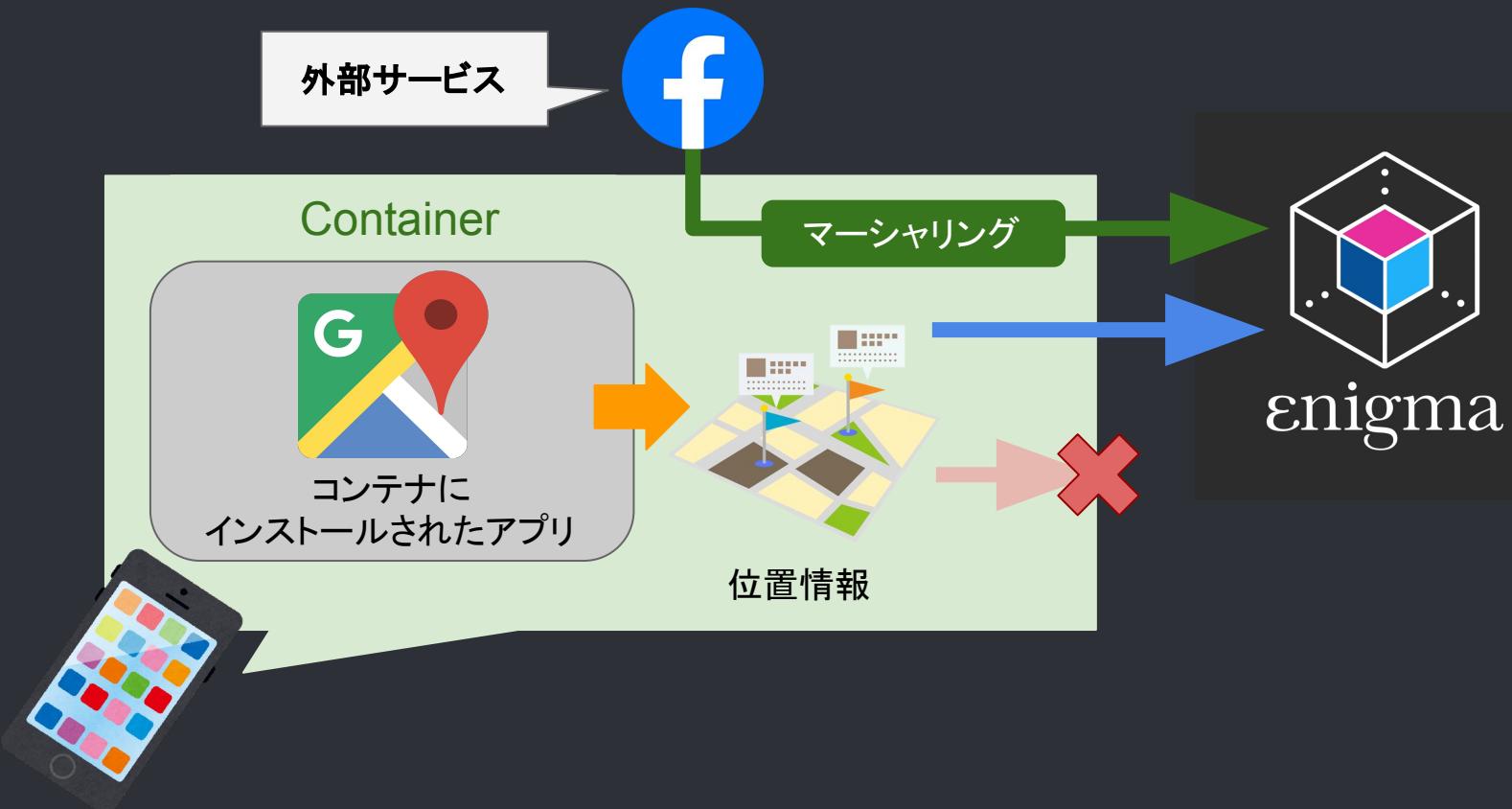


<データマーシャルウォレット>

Data Marshal Walletとは

(ウォレット原案:そらすえさん @SoraSue77、マーシャル構想:ぼく)

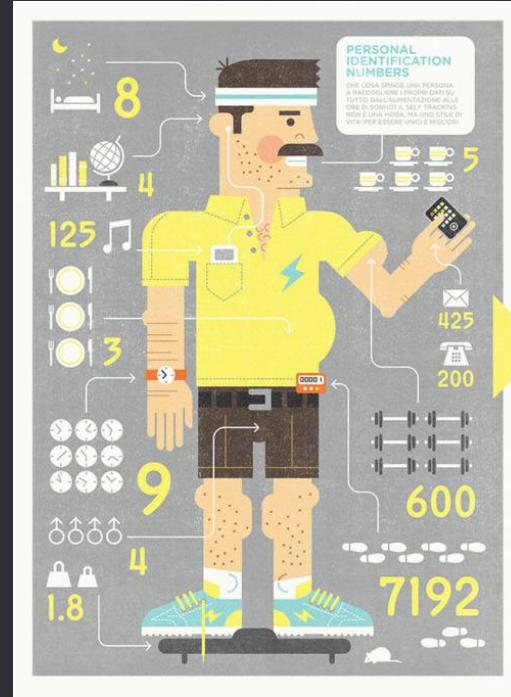
- データのコントロールをサポートするコンテナ環境である
- データ提供に際し、本人の意志を明確に介在させることができる
- また異なるサービスが提供するデータフォーマットを整形・統一(マーシャリング)することで事前処理を容易化する





## ＜データマーシャルウォレットのユースケース＞ 自身を量化する(Quantified Self)仕組みとの相性が良い

- 血圧・心拍数、睡眠時間、食生活、姿勢、位置情報・歩行距離、.....
- 情報共有すべき相手を選んで、デメリットを排除し、メリットを享受する



出典: <https://blog.rescuetime.com/weekly-self-tracking-links-the-quantified-self-edition/>

● <データの価値を取り戻す>

## 収集データを提供者本人と共有する仕組みを最大限に利用する

Googleの各種サービス、Facebook、Snapchat等では、本人に対する収集データの提供がサポートされている。これを活用し、経済的な意味でのデータの価値を部分的に取り戻す。

The image is a composite screenshot of three different service websites demonstrating data download features:

- Facebook:** A modal window titled "Download your information" is shown. It explains that users can download their Facebook data in various formats (HTML, JSON) and provides instructions for creating a password-protected file. It also links to "Access your information at any time".
- Snapchat Support:** A yellow-themed page titled "Snapchat Support" with a cartoon character. It features a search bar and links to "Discover tips and tricks, find answers to common questions, and get help!".
- Google Account Help:** A white-themed page with a navigation bar for "Google Account Help" and "Describe your issue". The main content area is titled "Download your data" and explains how users can export and download their data from Google products like email, calendar, and photos. It includes a note about the process not deleting data from Google's servers and links to account deletion information.



## 出典・参考資料

### 【書籍】

ダイヤモンド社「パーソナルデータの衝撃」 野村総合研究所 城田真琴氏

日経BP社「パーソナルデータの教科書」 野村総合研究所 小林慎太郎氏

### 【論文/WP】

Decentralizing Privacy: Using Blockchain to Protect Personal Data <https://enigma.co/ZNP15.pdf>

Enigma: Decentralized Computation Platform with Guaranteed Privacy [https://enigma.co/enigma\\_full.pdf](https://enigma.co/enigma_full.pdf)

Kleros

<https://assets.ctfassets.net/di6jbl3521d8/4JEvtlKQcbp5ka7OAj7pas/e9e13b47114e28303829de5c48ac443f/whitepaper.pdf>

### 【Web】

Enigma DISCOVERY Documentation <https://enigma.co/discovery-documentation/>

Official Blog of Enigma <https://blog.enigma.co/> "The Developer Quickstart Guide to Enigma"他

GitHub/enigmampc

[https://github.com/enigmampc/enigma-core/tree/develop/examples/eng\\_wasm\\_contracts](https://github.com/enigmampc/enigma-core/tree/develop/examples/eng_wasm_contracts)

Enigma Data Marketplace <https://enigma.co/marketplace/>

Intel社 <https://software.intel.com/en-us/articles/intel-software-guard-extensions-tutorial-part-1-foundation>  
Software Guard Extensions Tutorial Series