# ERC721 規格にワンタイムパスワード機能が付いた
# コンテンツ閲覧チケットトークンの説明図

西沢総合研究所 (NZRI)　西沢克弥



2020 年　7 月 12 日

※ 資料作成の高速化のため　いらすとや　さんの
画像を複数利用しています。
また外部企業の説明図を引用している箇所があります。

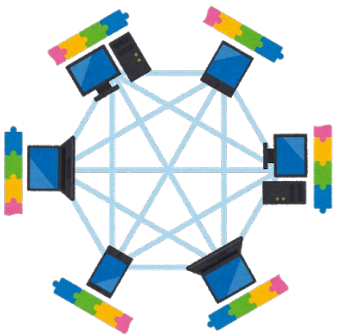
# — ERC-721 規格 —

イーサリアムという仮想通貨を基盤としたネットワーク上で動作する、スマートコントラクト（契約を実行するプログラム）の規格の一つで、非代替なノンファンジブルトークン（ NFT ）の規格です。

ERC20 規格などファンジブルなトークンは 2 号仮想通貨扱いですが、
NFT は仮想通貨には当てはまりませんが暗号資産に該当します。

・イーサリアムのピアツーピアのネットワークでは、ある時間ごとに仮想通貨の取引やスマートコントラクトのプログラムデータを取りまとめてブロック（塊）としてまとめ、前のブロックのハッシュ値と結合して現在のブロックのハッシュ値を求め、それを次のブロックに渡します。

・今回のプロジェクトではある一定時間ごとに塊で区切られることを利用し、時刻同期式のワンタイムパスワードを実現しています。



・イーサリアムの分散台帳の模式図

・ある時間ごとに取引されたブロックが繋がる様子。
左が過去のブロック。
このブロックチェーンはネットワーク上で共有されている

スマートコントラクトを ” 紙 ” に例えると、その紙の書式や規格、
記録事項や紙の振る舞いを決めるのが ERC-721 などの規格です。

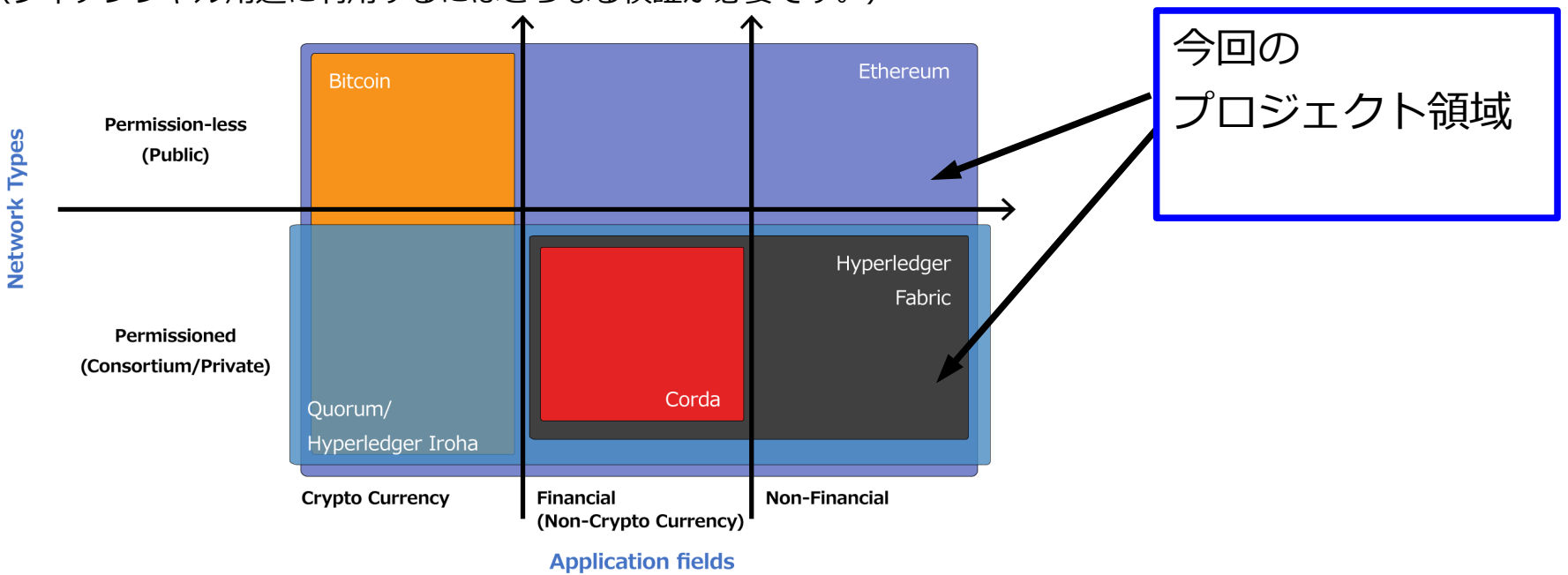
スマートコントラクトはイーサリアムネットワークのブロックに記録されます。
ブロックチェーンを採用してるので一度書き込んでしまうと改ざんできません。
コントラクトのオーナーでも書いてしまったことの修正はできません。

※ 改ざんが困難なのはある時間ごと、例えばイーサリアムでは 30 ～ 15 秒ごとの取引をまとめ、その前のデータのブロックハッシュ値を算出し次のブロックに埋め込んで鎖のようにつなぐことで改ざんを検知できるようにしています。この技術はブロックチェーンといわれます。ハッシュ値とはあるデータの要約で、その値を求める関数をハッシュ関数と呼び複数の種類があります。ハッシュ値から元のデータを推測することは困難な性質を持ち、一方向関数ともいわれます。

以下のページもご覧ください； https://www.nttdata.com/jp/ja/services/blockchain/002/

# イーサリアムやビットコインの立ち位置

・イーサリアムはすべての領域を網羅し、
現在インターネットサーバーとして多用される Linux OS と同じ雰囲気を持っています。
・オープンソースのため、誰でもネットワークの一員になることができ、初心者でもスマートコントラクトを作成しトークンのプログラムを作成する余地があるプラットフォームです。
・今回のプロジェクトの用途は以下の図の領域で表すと、半プライベート型でノンフィナンシャル型です。
（フィナンシャル用途に利用するにはさらなる検証が必要です。）




画像引用元； https://www.nttdata.com/jp/ja/services/blockchain/002/

## ―　ERC-721 規格と本プロジェクトのトークンについて―

http://erc721.org/
https://github.com/OpenZeppelin/openzeppelin-contracts
https://github.com/0xcert/ethereum-erc721

上記の商用利用可能な MIT ライセンスコードを参考に、

こちらで独自にトークンの転送制限機能の追加や任意トークンの除去、
そしてワンタイムパスワード機能の実装を行っています。
トークン発行時のブロック番号を打刻する機能や、
トークンが有効か無効か示すブーリアン変数を備えています。

この機能は 6 カ月以内に有効期限の切れるチケット、
とくに前払式決済手段のうち 6 カ月以内に有効期限が切れるものは届け出がいらないこと踏まえ設計しています。

―　動作の模式図、動作図、アーキテクチャー、運用方法について―

【 NFT 】 ERC-721 にワンタイムパスワードがついたもの。トークン除去機能や譲渡制限がない場合、イーサリアム口座間で自由に流通できる。古本の閲覧権が流通するイメージ。

【ウェブアプリ】あるサーバーにコンテンツとワンタイムパスワードログイン機能が付いたもの。 javascript で動作。

【 Crybon 】ウェブアプリを Electron-builder で実行可能ファイル exe や app 、 dmg ファイルで配布してユーザーのパソコンにコンテテンツを持たせた方式。コンテンツの閲覧にはイーサリアム口座の秘密鍵と、その口座が保有する実行可能ファイルに対応したトークンの ID が必要。
イメージとしては鍵のかかった同じ内容の本が無数に流通し、それを開錠するには鍵のトークンが必要でそのパスワードはワンタイムパスワードを採用している。鍵のトークンは古本の様にネットワーク上で流通する。

● 動作図、アーキテクチャー図

## １．大まかな動作の仕組み

顧客の端末
ウェブブラウザ
秘密鍵 A

ウェブアプリサイト
Web3.js で駆動

サイトの管理

運営者、コントラクトオーナー
ウェブブラウザ
秘密鍵 B

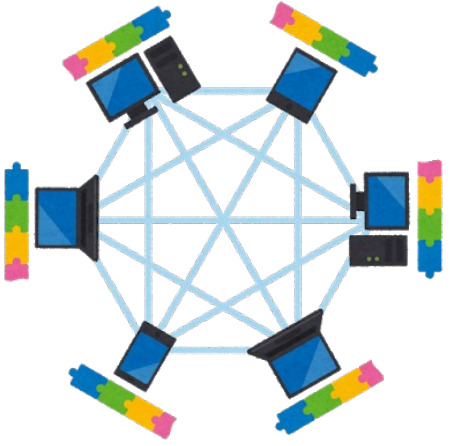
コントラクトから
ワンタイムパスワードを
取得、認証

コントラクトデプロイ、
コントラクトの管理、
内部変数の変更、
トークン発行

分散台帳ネットワーク
例えばネットワーク ID=4 , Rinkeby テストネット

ブロックチェーンノード図で
黄色のブロックにワンタイムパスワード生成器、
緑色のブロックに認証器がプログラムされていると仮定。

ブロック
チェーン
ノード



ネットワーク全体

# ● 動作図、アーキテクチャー図

## 参考　：具体的なアドレスの例

顧客の端末のウェブブラウザに、
顧客の秘密鍵 A = 0x00000000…0000000004479 をメタマスクにインポート
口座アドレス = 0x782C7A8e96D75e6170Fa1C3F0a08a380215eed62
このアドレスを宛先として運営者はチケットトークンの発行を行い、
このアドレス専用のワンタイムパスワードの取得と認証が行われる。

ウェブアプリサイトは javascript と Web3.js で駆動 、
サーバーサイドは php など

ワンタイムパスワードの表示と認証
https://otp-testnet-nzri.herokuapp.com/
（ heroku をサーバーとして利用）

コントラクトから
ワンタイムパスワードを
取得、認証

サイト管理

運営者、コントラクトオーナー
ウェブブラウザ
秘密鍵 B をメタマスクにインポートして所持
運営側口座アドレス **0x0f398803BE4319B98F164cae47589797aC5cF906**

コントラクトの
管理、変数の変更。

トークン発行
発行結果の
台帳への記録。

イーサリアムネットワーク
ネットワーク ID=4 , Rinkeby テストネット

黄色のブロックにワンタイムパスワード生成器
（コントラクトアドレス 0xAa69136eE376cd8Ff1Ba20b97Ba2f728c1ad8121 ）

緑色のブロックに認証器
( コントラクトアドレス 0xA572C78aa375E89887c7381bbf4fecbf3e38A167)
がプログラムされている。

0xA572C78aa375E89887c738
1bbf4fecbf3e38A167

0xAa69136eE376cd8Ff1Ba20b9
7Ba2f728c1ad8121

ブロック
チェーンノード

※ メタマスク無しでもパスワードの取得が可能なウェブアプリも用意可能
https://otp-testnet-nzri-with-pri-key.herokuapp.com/

トランザクション等確認ページ
https://rinkeby.etherscan.io/address/0xAa69136eE376cd8Ff1Ba20b97Ba2f728c1ad8121
https://rinkeby.etherscan.io/address/0xa572c78aa375e89887c7381bbf4fecbf3e38a167

# ２．トークンの発行、チケットの売り出し

分散台帳ネットワーク
例えばネットワーク ID=4 , Rinkeby テストネット

ブロックチェーンノード図で
黄色のブロックにワンタイムパスワード生成器、
緑色のブロックに認証器がプログラムされていると仮定。

顧客の端末
ウェブブラウザ
秘密鍵 A

顧客注文、決済完了、イーサリアムアドレス提示

トークン注文受付サイト

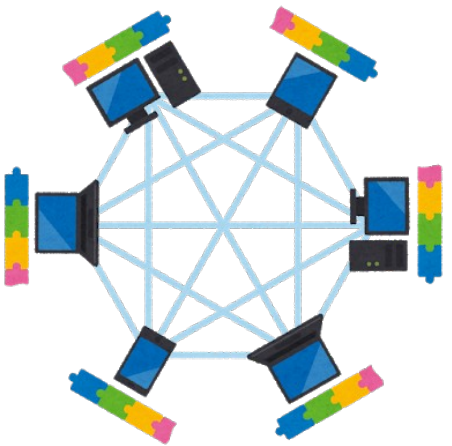
顧客の注文順と
イーササリアムアドレスを取得

運営者、
コントラクトオーナー
ウェブブラウザ
秘密鍵 B

・顧客のアドレス、
トークン ID を用いてトークン発行
トークン発行用自社アプリを使用

トークン
発行

ブロック
チェーン
ノード



ネットワーク全体

※ トークン ID は注文の早い人順に小さな番号を割り当てる。

# ３．トークンの利用【ウェブサイトサービス】

顧客の端末
ウェブブラウザ
秘密鍵 A

・顧客はウェブアプリにアクセスする。
・ Metamask アドオンを使うか、
秘密鍵を直接入力するか選択可能。

・生成器と認証器のコントラクトから
ワンタイムパスワード OTP を取得、認証し
ウェブアプリ上のサービスにログイン

ウェブアプリサイト
Web3.js で分散台帳にアクセス
javascript をベースにする
↓
ログインできた人にサービス提供

【 OTP 認証】

【 OTP 取得】

運営者、コントラクトオーナー
ウェブブラウザ
秘密鍵 B

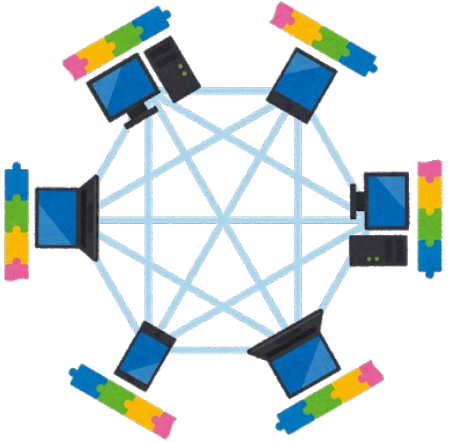
<サービス例>
小説、画像、
マンガ、動画閲覧。

チャット、ライブ配信
会員サイトなども可能

<分散台帳ネットワーク>

ブロックチェーンノード図で
黄色のブロックにワンタイムパスワード生成器、
緑色のブロックに認証器がコントラクトとして
プログラムされている。

ブロック
チェーン
ノード



ネットワーク全体

# 3．トークンの利用【 Crybon( 仮 ) サービス】

顧客の端末
ウェブブラウザ
秘密鍵 A

・顧客は PC で exe ファイルのアプリを開く。
・秘密鍵を直接入力。
・ web3 プロバイダアドレスを直接入力

・生成器と認証器のコントラクトから
ワンタイムパスワード OTP を取得、認証し
exe ファイル内部のコンテンツを閲覧。

＜ Crybon アプリ＞
秘密鍵 A とトークン ID 入力
↓
認証できた人にできた人に、
コンテンツ提供

【 OTP 認証】

【 OTP 取得】

出版社
コントラクトとトークンの発行
exe ファイルの発行

＜サービス例＞
小説、画像、マンガ、動画閲覧。

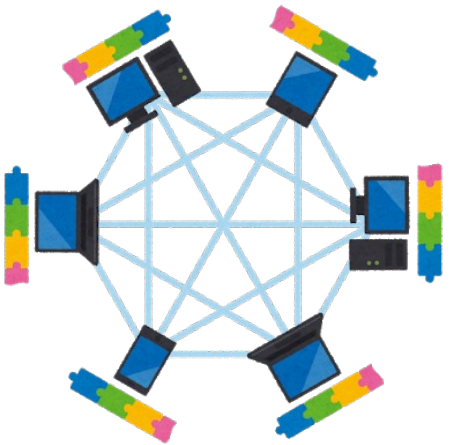
この方式は本のデータを、 exe ファイルと
してユーザーのパソコンで保持できる。

exe ファイルは分散 DB の IPFS などで保
存することも想定。

＜分散台帳ネットワーク＞

ブロックチェーンノード図で
黄色のブロックにワンタイムパスワード生成器、
緑色のブロックに認証器がコントラクトとして
プログラムされている。

ブロック
チェーン
ノード



ネットワーク全体

# 3．トークンの利用【現実世界でのチケット】

・スキャンアプリの試作例（JsQR.js 利用）
https://otp-qr-scan-testnet-nzri.herokuapp.com/

顧客の端末
ウェブブラウザ
秘密鍵 A

QR
コード
表示・印刷

QR コード
スキャン

会場でチケットの
QR コード提示

QR コードスキャン認証ウェブアプリ
カメラを持った端末で、
会場の顧客のチケットを読み取り
↓↓
認証 OK の顧客を入場させる
もしくはサービスを提供する

運営者
ウェブブラウザ
秘密鍵 B

入場者をデータベース化して管理する

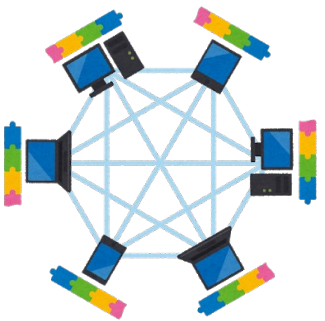
<ウェブアプリサイト>
Web3.js で分散台帳にアクセス
javascript をベースにする
↓↓
OTP を QR コードで端末画面に表示、
紙への印刷も可能

・QR コード表示アプリ（qrcode.js 利用）
https://otp-testnet-nzri.herokuapp.com/

【OTP 取得】

<分散台帳ネットワーク>

【OTP 認証】



ブロックチェーンノード

※ 現在の仕様では、紙に OTP の QR コード印刷する場合は時刻同期型でないパスワードを選択しないといけない。