# Devcon2　参加報告

2016 年 10 月 12 日
hiron（野畑裕保）

9 月 19 日～21 日に上海で開催された Devcon2 に参加したので、その概要と感想を報告します。

**目次**



## A. Devcon２概観 － 多様な Track theme

1．Devcon2 の発表件数は　78 件！　（21、28、29）
15～25 分/件
プログラムは　agenda

2．どんな発表があったのか
Agenda の各プレゼンの詳細紹介頁の下部には Track theme のタグが付けられている。

Agenda の頁の右上の "Filter by track"で検索、集計すると、

| Track theme | Day1 | Day2 | Day3 | Total |
|---|---|---|---|---|
| casper | 3 | 0 | 0 | 3 |
| clients | 2 | 6 | 2 | 10 |
| dapp development | 3 | 18 | 20 | 41 |
| devcon | 1 | 0 | 0 | 1 |
| development tools | 3 | 12 | 11 | 26 |
| enterprise | 0 | 1 | 5 | 6 |
| ethereum 2.0 | 4 | 1 | 1 | 6 |
| ethereum foundation | 2 | 0 | 0 | 2 |
| evm | 3 | 3 | 0 | 6 |
| go-ethereum | 0 | 1 | 0 | 1 |
| light client | 1 | 2 | 0 | 3 |
| middleware | 1 | 2 | 4 | 7 |
| Mist | 0 | 1 | 0 | 1 |
| off-chain | 3 | 1 | 4 | 8 |
| platform development | 3 | 6 | 8 | 17 |
| prediction market | 0 | 0 | 2 | 2 |
| privacy | 1 | 0 | 0 | 1 |
| research | 12 | 3 | 2 | 17 |
| scalability | 4 | 0 | 1 | 5 |
| security | 5 | 8 | 2 | 15 |
| solidity | 0 | 7 | 1 | 8 |
| swarm | 2 | 1 | 0 | 3 |

4．追加したいTrack theme

Formal verification

Oracle (data-feed)

Identity

IPFS

IoT

Namespace

…

5. 参考になるDevcon2のレポート

(1) Ethereum Devcon2 Conference In Shanghai, Agenda With Slides
プレゼン内容の概要を記述。スライドへのリンクあり。ビデオへのリンクも追加されてきている。EthFans.org（中国語）がオリジナル。
また、他のレポートへのリンクあり。

(2) DEVCON2 report: Day 1 - Session notes & event photos
By David Burela、プレゼン内容について彼のメモあり。
day-2, day-3, および Summit demo day, day-1, day-2などへのリンクは上記頁に掲載されている。

(3) "9 Must-Watch Talks at Ethereum's Big Developer Event" (CoinDesk)もある。

1. Ethereum in 25 Minutes – Vitalik Buterin
2. Swap, Swear and Swindle. Swarm Incentivization – Viktor Trón and Dr Aron Fischer
3. A Correct-by-Construction Asynchronous Casper Protocol – Vlad Zamfir
4. State Channels: Systemic Security Considerations and Solutions – Joseph Poon
5. Panel: Smart Contract Security in Ethereum
6. Formal Verification for Solidity – Dr Christian Reitweissner
7. Imandra Contracts: Formal Verification for Ethereum – Dr Grant Passmore
8. Mist Vision and Demo – Alex Van de Sande
9. Ethereum Blockchain Initiatives at Thomson Reuters – Dr Tim Nugent

プレゼンを逐一紹介するのは困難。そこで

## B.　Devcon2 に参加して感じたこと、思ったこと、考えさせられたこと

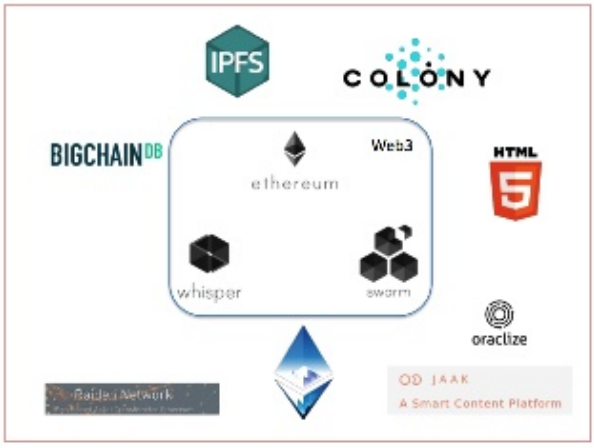
をもとに気になったプレゼンをピックアップします。

"Mauve Revolution" (Vitalik)の発表もあったが、

### 1.　Web3 の強調

"**Blockchain as Web3**"　　("Beyond the Bubble"　2-20　でのキャッチ)

**Web3 と関連技術:**




" Towards Web3 Infrastructure"　から引用

第 4 頁の図に Web3 Base Layer Services (ethereum, swarm, whisper)の区切りをいれた。

**Swarm:**

(1) "Swap, Swear and Swindle. Swarm Incentivisation", Viktor Trón and Dr. Aron Fischer　1-6

最後尾の 1-6 は Day-1 の 6 番目のプレゼンを示す。

(2) "Towards Web3 Infrastructure", Viktor Trón 1-10

・開発ステイタス：α (PoC 0.2)

・仕様書：ethersphere orange paper series

また、Viktor 曰く「Ethereum Foundation では早くから Web3.0 が意識されていた。」
（IBTimes の記事 Ethereum's Viktor Trón talks about Swarm and the skeleton of Web 3.0 ）

**IPFS:**

(3) "IPFS & Ethereum: Projects, Important News, Demos, and More", Juan Benet 3-9

(4) "IPFS Libp2p & Ethereum networking", David Dias and Juan Benet 3-11

・Smart Contract で Web access
世の中（IT）が同じ原理で動かせるならメリット／面白い。

・下記のプロダクト（？）の位置付け、用途の PROS/CONS はなにか？

・swarm vs. IPFS

- swarm または IPFS に適用可 という products がでてきている。
- "IPFS & SWARM" は双方を比較解説している。要参照。

・BigchainDB

・Safe Network (MaidSafe α)

## 2. Off-chain

・off-chain にすることのメリット;
Scalability/Performance の向上、low cost、

(1) "The Raiden Network", Heiko Hees/brainbot technologies 1-8

(2) State Channel 関係
"State Channels and Blockchain Applications", Jeff Coleman/Ledger Labs 1-12

"State Channels: Systemic Security Considerations and Solutions", Joseph Poon/lightning network 1-21

(3) "Mango: Git Completely Decentralized", Alex Beregszaszi 2-27
・GitHub の decentralize 化

(4) "The Golem Project: Ethereum-based market for computing power", Julian Zawistowski 3-17
・Worldwide Super Computer

(5) "Orbit: Distributed, Real-Time Web3 Apps with IPFS and Ethereum", Samuli Poyhtari/Protocol Labs 3-23
・chat app

など。

## 3. Formal Verification

・Writing code correctly is hard.
Easy to test desired behaviour. Hard to check absence of undesired behaviour.

(1) "Formal Verification for Solidity", Dr. Christian Reitwiessner, Dr. Yoichi Hirai 2-5
- ・Formal verification uses techniques to "test" a program on all possible inputs and states.
- ・Key purpose of FV: complexity reduction by probing properties.

(2) "Making Smart Contracts Smarter: Oyente", Loi Luu/NUS 2-19
- ・contract analyzer

(3) "Imandra Contracts: Formal Verification for Ethereum", Dr. Grant Passmore and Evgeny Gokhberg 2-7
- ・cloud-based formal verification system

**その他、** testing, bug pattern、error pattern, attack pattern

(4) "Testing Ethereum Consensus", Dimitry Khoklov 2-22

(5) "Behavioral Types for Smart Contracts", Lucius Greg Meredith 1-17
behavior error, formal verification

下記は "Directions in Smart Contract Research: A Selection", Philip Daian 1-5 で言及された注目すべき主張。（ここまで来たか）

Don't forget traditional SE
Tests, fuzzing, static and dynamic analysis, phased deployment/upgrade, etc.

## 4. Enterprise への適用

8 つのプレゼンが enterprise の tag を持つ。その中から、

(1) "Panel: Ethereum Enterprise"　3-27　での発言でのキーワード;

InterOperability

Privacy

Scalability

Migration plan

Integration

**(2) "Ethereum for Enterprise",** Victor Wong/BlockApps　3-28

Blockchain の enterprise への適用について aggressive な見解を披露。

100 社以上への実装経験。Haskell ベース。

そのプレゼンから、


"Our members are no longer interested in POCs. They want to prototype real systems and push these into production ASAP. We think we will see banks in production end of this year or early 2017.　Tim Grant, Head of R3 Labs"


Two markets とは、

・Simplifying/replacing existing business processes

・Building new applications/platforms as Blockchain apps

番外－1

## 5. Oracle

Sybase や Postgres ではなく、Oracle です。

An oracle is an external actor which can provide information from the real world into the blockchain

(Conditional) Timer Event、為替情報、マーケット情報、

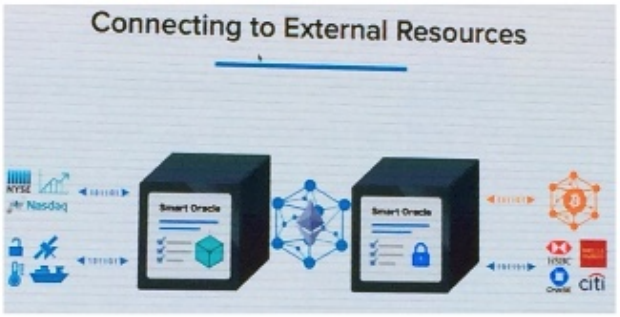
(1) "A Provably Honest Oracle Model: Auditable Offchain Data Gathering & Computations", Thomas Bertani/oraclise.it 3-5

Oraclize is a provably-honest oracle service enabling smart contracts to access the Internet.


PROS:

- Full onchain transparency (both query & results are onchain)
- Direct access to any Web API (no need for them to adapt!)
- No trustline open w/ the oracle
- Doesn't need many oracles in place to provide reliable results (hence working today)

(2) “Smart Oracles”　　/smartcontact.com　3-8




(3) Thomson Reuters の事例:

TR での検討中プロジェクトの紹介。Hyperledger もやっているが、Ethereum に一番注力している。アプリは、

・Customer Identity service: BlockOne ID。KYC service と連携

・ Pricing Oracle Service: ELECKTRON REAL TIME のマーケットデータを Smart Contract から検索可能としている。

“Ethereum Blockchain Initiatives at Thomson Reuters”, Dr. Tim Nugent 3-15　および　“Thomson Reuters Demos New Ethereum Blockchain Use Cases” (CoinDesk)

(4) “Prove It – Blockchain based KYC”　Igor Lillic/ConsenSys　3-22

主なテーマは Identity だが、Oracle を介した KYC にも言及。

## 6. Dapps の開発は　Solidity で十分？

### ・DSL　(Domain Specific Language)

(1) "Designs for the L4 Contract Programming Language Based on Deontic Modal Logic", Dr. Virgil Griffith and Vikram Verma　1-19

"WHY NOT JUST USE SOLIDITY?"

- ○ Many Ethereum programs are "contracts" in the traditional legal sense.
- ○ **SQL** shows us DSLs really do make our lives easier.
- ○ We can make lives easier with a DSL explicitly for Contracts
  - ・E.g., you'll still use Solidity for things like: very novel apps, low-level crypto, ponzi schemes.
  - ・http://dapps.ethercasts.com/
- ○ High-level languages → **less implementation details to think about**.
  - ・Outsources tricky parts to the compiler writer
- ○ Easy mapping to the current legal system
  - ・One day, your contracts will even compile to legally enforceable English!
  - ・Everyday lawyers move to Ethereum for greater reliability?

(2) "Smart_Contracts_as_Parametrization", Henning Diedrich　2-23

当たっていないかもしれない。

(3) "Building Highly Scalable, Optimized, Standardized dApp's (from UI to Contracts)", Nick Dodson　3-25

react-dapp-boilerplate.　Coming soon

これも当たっていないかもしれない。

番外－２

## 7. IoT への取り組み

・Raspberry Pi でも動く Ethereum

Fog/Edge computing

"Ethereum for Resource-Limited Devices" Bob Summerwill 1-18

・Chip の開発 （何も 21.com だけではない）

・Raiden Network 1-8

Parity 1-11, 2-6

iEx.ec: Distributed cloud 3-6 など

・IOTA 関連のプレゼンはありませんでした。

## 8. その他

・Blockchain 3.0

"The Decentralized Collaborative Web", Matan Field/ Backfeed 3-26

こういう整理もある。

**Blockchain-ography**

| Block Chain | Value System | Gist | Decentralized | Example Use cases | Stage |
|---|---|---|---|---|---|
| 1.0 | Hard-coded Objective | Tracking System | Ledger | Trustless cash, diamonds, art, shipments | Early product |
| 2.0 | Soft-coded Objective | Smart Contracts | Execution | Self-executing accounting, banking, games, IoT, contracts | Prototype |
| 3.0 | Emergent Subjective | Collective Intelligence | Governance | Decentralized collab, curation, insurance, investment | Research |

・Devcon2 Identity - Get your Devcon2 Identity Token
https://www.reddit.com/r/ethereum/comments/53gr0y/devcon2_identity_get_your_devcon2_identity_token/ も。

Devcon2 の参加者に Token を発行する activity。参加証明の Token を配布する　だけではなく、
"It's an experiment to see what people do with it and what people build with it." とある ものの。。。

・Censorship resistant
Blockchain の特徴のひとつとして耐検閲性が大きく取り上げられている。
Web2 での個人情報の収集、検閲の懸念がバックグラウンドにある（ようだ）。

## C. まとめの感想

・発表が多く、どれも短い → キーワードを知る → follow-up → 想いを馳せる → なんか XXX 。。。

・Devcon に限らず、継続して参加 → refresh/rewind

この雑文が皆さんの何かの切掛けになれば幸甚です。

Refreshed!

## 番外項目

本項は Devcon2 のプレゼンとは関係がありません

(番外－１)

**ここで** Legacy (BPM-ECM-ERP)**との対比　を考えてみよう。**

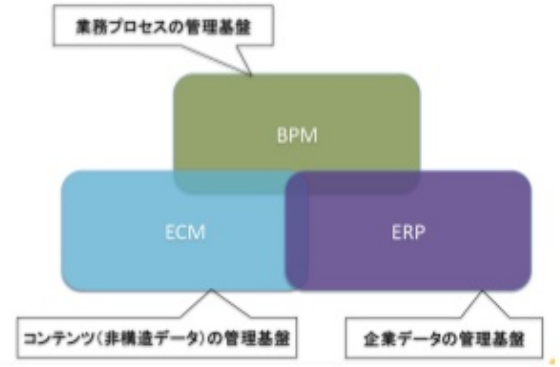
・BPM-ECM-ERP って？

BPM: Business Process Management

ECM: Enterprise Content Management

ERP: Enterprise Resource Planning　(企業の基幹業務をサポート)

他のキーワードも取り上げられる場合もある。CRM、EDM、ERM




(1) 対比：大雑把に言って、

**Blockchain ---　BPM**

(例えば) **Factom ---　ECM**

**Oracle ---　ERP**　(との interrface)

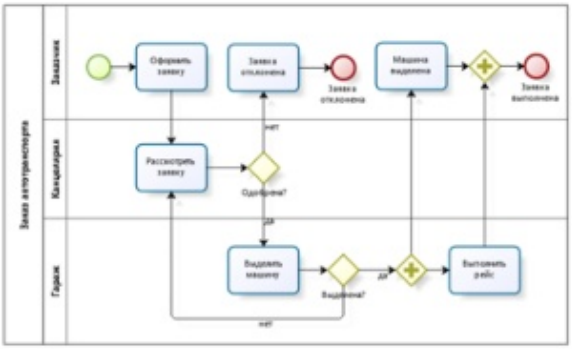
このモデル対比が果たして有効かどうかは 。。。

(番外－２)

再び Legacy との対比の一環として、BPM の開発ツールについて考えてみよう。

(2) **BPMN（Business Process Modeling Notation）を使用**して、ビジネスのスマートコントラクトのワークフローを表現する。

BPMN: ビジネスプロセスをワークフローで描画するグラフィカルな業界標準記法。



BPMN の特長：**すべてのビジネス関係者が容易に、共通の理解をえることができる。**

(3) Blockchain platforms/tools は BPMS (BPM Suite)へと進化する（もしくは既に進化しつつあると期待）

Solidity を使うのもいいが、C（または相当）でプログラミングをしているのに等しい。DSL のツールを使用し、複雑な**ビジネス・プロセスでも、UI の開発を含めて、効率よく、エラー少なく開発出来ないか。**
その解の一つが BPMN/BPMS のアプローチであろう。

BPMS の特長:

- BPM を実現するための実行プラットフォームであり、業務プロセスのライフサイクル・マネジメントが可能
- BPMN のプロセスフローダイアグラムを入力とし、データスキーマの定義、入力フォームのデザイン、条件判定などを NO CODING で行うことができ、実行モジュールを生成する。本番環境でのプロセスモニタリング、デプロイ管理が可能

-eol