Global University Blockchain Workshop
Start your genesis block
Join Ontology Today, Be a Part of the Future

- Raise questions and discuss during the course
- Get tech updates at first time
- Meet developers worldwide
- Reach Ontology core developers anytime
THE RUNDOWN

01 Final web
  The future

02 Way to Final Web
  Way to the future

03 What can we do
  Ontology’s solution to the future

04 What can you do
  Be a Part of the Future

05 Q & A
FINAL WEB
Let’s start from the Web
“The Web as I envisaged it, we have not seen it yet. The future is still so much bigger than the past.” (2009)

– Founder of World Wide Web
Sir Tim Berners-Lee

Who is this handsome gentleman?

*Picture resource: Internet
What is Web 1.0/2.0?

1.0  Web Portal
2.0  Social Network

What are the problems in Web 1.0/2.0?

• The truth
• Private data
• Top market
• Etc.
What might Web 3.0 be?
“Using the WWW infrastructure to create a global, decentralized, weblike mesh of machine-processable knowledge.”

– Founder of World Wide Web
Sir Tim Berners-Lee
WHY? Decentralized Knowledge
Distribution of the network

Centralised (A)  Decentralised (B)  Distributed (C)

Final web // Decentralized
Centralized Web

- Centralized
- Unclear data ownership
- Abuse of resources without permission
- Moral trust
- Trust process based on third parties
- Unquantifiable

Decentralized Web*

- Decentralized
- Digitized data ownership
- Self-sovereign resources
- Technical / legal / social trust
- Independent trust
- Quantifiable

* Decentralized Web (TRUST Network)
WHY? Decentralized Knowledge
Only Valuable Info is Knowledge

Linked Entropy Reducing Valuable
WHAT?

Web of Society

- Self-sovereign identity & data
- Data privacy & protection
- Web-like linked data
- Transparency on the path of properties-class
HOW? Decentralized Knowledge
<table>
<thead>
<tr>
<th>Cloud</th>
<th>Blockchain + Ontology</th>
<th>Web of Society</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dynamic rental service</td>
<td>• User becomes a stakeholder</td>
<td>• Self-sovereign identity &amp; data</td>
</tr>
<tr>
<td></td>
<td>• Trust + Stability</td>
<td>• Data privacy &amp; protection</td>
</tr>
<tr>
<td></td>
<td>• Benign competitive business environment</td>
<td>• Web-like linked data</td>
</tr>
<tr>
<td></td>
<td>• Traceability &amp; Transparency</td>
<td>• Transparency on the path of properties-class</td>
</tr>
</tbody>
</table>
Web 3.0

Final Web
WAY TO FINAL WEB
TRUST? Infrastructure
Traditional Web

- Centralized trust
- Unclear data ownership
- Moral trust
- Trust process based on third parties
- Abuse of resources without permission
- Unquantifiable

TRUST Network

- Decentralized trust
- Digitized data ownership
- Technical / legal / social trust
- Independent trust
- Self-sovereign resources
- Quantifiable
How to TRUST?
<table>
<thead>
<tr>
<th>Network Components</th>
<th>TRUST Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decentralized network</td>
<td>Decentralized trust</td>
</tr>
<tr>
<td>Blockchain infrastructure</td>
<td>Digitized data ownership</td>
</tr>
<tr>
<td>Smart contract mechanism (code is law)</td>
<td>Technical / legal / social trust</td>
</tr>
<tr>
<td>Self-sovereign identity</td>
<td>Independent trust</td>
</tr>
<tr>
<td>Self-sovereign data token</td>
<td>Self-sovereign resources</td>
</tr>
<tr>
<td>Distributed data exchange framework</td>
<td>Quantifiable</td>
</tr>
</tbody>
</table>
Three dimensions of trust infrastructure

NETWORK
Trust the decentralized network for consensus algorithm with certain hypothesis

DATA PROCESSING AND MANAGEMENT
Trust the decentralized applications for the traceability of actions and permission control on self-sovereign data

IDENTITY
Trust the dApp users for their clear immutable action log with verifiable claims/signatures of their self-sovereign identities
A decentralized network infrastructure
How to design a blockchain project?
Trilemma

Consensus algorithm
- Safety
- Fault Tolerance
- Liveness

Blockchain project
- Scalability
- Performance
- Security
Economics

Data = Credential + Rights

Cross-border of Traditional Business

Token Economics

Token For Trust Endorsement

Asset-based Exchange

Data Assetization

Trust Asset

Linkable

Exchangeable
BUSINESS

The Market

INFRASTRUCTURE

Blockchain

Way to final web // Blockchain project
Way to final web // Blockchain project

EOS  Ethereum  COSMOS

*Picture resource: Internet
Ontology

Way to final web

Blockchain project
Knowledge – Next Phase of Ontology
What can we do
What is Ontology?

New Generation of Multi-Chain Infrastructure

A Distributed Collaboration Platform
<table>
<thead>
<tr>
<th>Performance issues</th>
<th>Different business requirements</th>
<th>Digital economy</th>
<th>Real economy</th>
</tr>
</thead>
</table>

// Ontology
200+ Core Members
900+ Technical Contributors
19 Languages
2,000,000+ Community Contributors
TRUST Redefined
## Our Output

<table>
<thead>
<tr>
<th>ONT ID</th>
<th>DDXF</th>
<th>Sharding</th>
<th>Cross-chain</th>
<th>MainNet</th>
</tr>
</thead>
</table>

// Our output
ONT ID
Trust the dApp users for their clear immutable action log with verifiable claims/signatures of their self-sovereign identities

DECENTRALIZED IDENTITY
Framework integrates multi-dimensional trust to provide secure, credible and fine-grained identity management.

• Self-sovereign Identity
• Qualification and verification of identities
Decentralized Data Exchange Framework (DDXF)

Supports cross-system data interoperability, data-based value assessment and data processing tracking.

**SELF-SOVEREIGN DATA**
- Off-chain
- Data access tokenization
- Data processing and transaction attestation

**DATA INTEROPERABILITY CROSS SYSTEMS**
- Resource exchange and data interaction through data pattern change, token transfer and transaction
Way to final web // DDXF
Sharding

Ontology Sharding named “Multi-Layered”
Root Chain

SHARD 1
- Contract
  - Contract range 0 ... 20
    - Contract #11

SHARD 2
- Contract
  - Contract range 21 ... 40
    - Contract #32

SHARD 3
- Contract
  - Contract range 41 ... 60
    - Contract #54

SHARD 4
- Contract
  - Contract range 61 ... 80
    - Contract #68

SHARD 5
- Contract
  - Contract range 81 ... 100
    - Contract #95
Support both scale-up and scale-out
- Multi-layer cross-shard service

3-dimens of sharding
- Network sharding
- State sharding
- Transaction sharding
Why is Ontology Sharding named “Multi-Layered”? 
Sharding

- For scalability
- From one blockchain to multiple parallel blockchains
- Be as safe as one blockchain
Sharding – Evolution of Multi-Processor

• For scalability
• From single thread to multiple threads
• The difficulty is from design of memory system
Sharding – Evolution of Multi-Processor

- For scalability
- From single blockchain to multiple parallel shard chains
- The difficulty is from design of memory system
Blockchain Sharding

How the memory system support parallel shard chains?
Sharding Performance

For performance
- maximize parallel part
- minimize serial part

Sharding by smart-contract
- smart-contract
- local state
- transactions to shard contracts

\[
\text{Speedup}(N) = \frac{1}{(1-P) + \frac{P}{N}}
\]

Serial part of job = \(1 \text{ (100\%)}\) - Parallel part

Parallel part is divided up by \(N\) workers
Blockchain Shard `memory system`

- Defines how blockchain shards interact with each other.
- Blockchain is durable computing system
  Blocks record history, and the history must be able to be replayed.
- Consensus algorithm defines how to generate the chain of blocks.

Consensus

- Safety
  if (protocol-following) nodes do decide on values, then they will decide on values that do not conflict.
- Liveness
  any (protocol-following) node will eventually make a decision that accepts or rejects any given value.
Properties of Blockchain Consensus

For each of these protocols:

We will define:

- **Protocol messages**
  So protocol states and their transitions are defined (sets of messages with no more than \( t \) faults)

- **An estimator**
  So that we can relate protocol states to values of the consensus

Because that’s all we need to satisfy the cbc framework

CBC: Correct-by-Construction
Safety of CBC Estimator

Estimate Safety:
• A proposition $p$ is safe at protocol state $\sigma$, if it holds for the estimator for all future protocol states $\sigma'$.

Safety Oracle
• If:
  – There is an e-clique in $\sigma$ of non-Equivocating nodes with total weight $W_e$
  – The total weight of all validators is $W_t$

Then:
• Clique e is safe in $\sigma$ for a protocol with fault tolerance $t < 2 \times W_e - W_t$
• The estimator will agree with the largest majority of latest messages. (e.g. GHOST)
Messages among Shards

- Ordering of xshard messages, to keep blockchain replay-able
- Validation of cross-shard messages
- Reasonable message passing latency
Safety of CBC Shard Block Estimator

Protocol messages:

Very similar to the blockchain, but:
- There’s a chain for every every shard
- They can share “merge blocks”

`Merge blocks` represents atomic-syncing-blocks among shard chains.
Ontology Multi-Layered Sharding

- Child shards follow their parent.
- Parent shard is contained in child shards.
- `Merge block` is in the parent’s fork choice.
  `Merge block` is atomic
- Child shards are verifiable-randomly ordered by their parent.
`Merge Blocks` in Ontology Sharding

- Parent shard is contained by its all child shards
- Child shard block must bind with one parent shard block

Estimator Safety:
A proposition $p$ is safe at protocol state $\sigma$, if it holds for the estimator for all future protocol states $\sigma'$. 
Why is Ontology Sharding named “Multi-Layered”? 

It’s a MULTI-LAYERED consensus system or Hierarchical consensus system.
More about Ontology sharding?

- XShard messaging
- XShard Atomic transaction processing
- XShard Governance (Fee Settlements)
- ...

Way to final web // Sharding
Cross-Chain

- Infinite expansion with chain peers
- Improving TPS bottleneck and reducing storage pressure
- Value anchoring of gas fee throughout chain network
- Secure cross-chain transactions
- Well defined of relayers' incentive mechanism
- Broad business scenarios
DEVELOP WITH ONTOLOGY
Decentralized Blockchain Application (dApp)

A dApp is a ‘blockchain enabled’ website, that runs on a peer-to-peer network of computers rather than a single server, where the Smart Contract is what allows it to connect to the blockchain. It contains both front-end and back-end and run independently on all nodes.
Key Features of a dApp

- **Decentralized**
  No one controlling power.

- **Protocol / Algorithm**
  Open protocol to everyone.

- **Incentive**
  Crypto tokens are used to reward participants.

- **Blockchain**
  Code / Data stored in a blockchain.
How to develop a dApp

01 Know your project
  • Design

02 Setup the project
  • Choose Platform, Framework
  • Design UI, Protocol, APIS

03 Code the application
  • Smart-Contract, Front-End, Back-End

04 Deploy and Test Your dApp
  • Make sure fully tested
Before we start coding …
How dApp run on Ontology blockchain?

How Ontology blockchain run?

Which part of dApp run on Ontology blockchain?

How smart-contract work?

How to interact with smart-contracts on Ontology blockchain?
How Ontology blockchain run?

- Decentralized P2P network
- Messages are broadcasted to all nodes
- Blocks are relayed in network
- Blocks are processed and verified by all nodes
How smart contract work?

- **Transaction (Message)**
  - contract address
  - input arguments
- **Code**
  - executed by virtual machine
- **Data**

---

**Transaction from network**

**World state \( \sigma_t \)

**User Account**

**Virtual Machine**

**World state \( \sigma_{t+1} \)

**Gas supply**

**Message**

**Refund**

**Gas**

**Updated**
How to interact with smart contracts on Ontology?

- **Front-End**
  - App
- **Back-End**
  - SDK
- **Protocol**
  - Smart Contract ABI
    (Application Binary Interface)
- **Blockchain**
  - Virtual Machine
All you need to Develop with Ontology
Develop with Ontology

Smart Contract and VM

<table>
<thead>
<tr>
<th></th>
<th>VM / Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light contract</td>
<td>Solidity / EVM; Python / NEO VM</td>
</tr>
<tr>
<td>Complex contract</td>
<td>WASM (EOS; Ontology; ETH 2.0)</td>
</tr>
<tr>
<td>Optimize for Finance</td>
<td>Move (Facebook)</td>
</tr>
</tbody>
</table>

Ontology multi-VM solution

<table>
<thead>
<tr>
<th>VM</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEO VM</td>
<td>Python</td>
</tr>
<tr>
<td>WASM VM</td>
<td>C++; Rust</td>
</tr>
</tbody>
</table>
Develop with Ontology

- **dApp Development Framework (Punica)**
  - [http://punica.ont.io/docs/punica/](http://punica.ont.io/docs/punica/)
  - [https://punica.gitbook.io/docs/](https://punica.gitbook.io/docs/)
  - [http://punica.ont.io/tutorials/](http://punica.ont.io/tutorials/)

- **Online Smart-Contract IDE**
  - [https://smartx.ont.io/#/](https://smartx.ont.io/#/)

- **Chrome Plugin Wallet (Cyano)**
  - [https://github.com/ontio-cyano](https://github.com/ontio-cyano)

- **SDKs**
  - [https://ontio.github.io/documentation/ontology_overview_sdks_en.html](https://ontio.github.io/documentation/ontology_overview_sdks_en.html)
  - [https://github.com/ontio/ontology-java-sdk](https://github.com/ontio/ontology-java-sdk)
  - [https://github.com/ontio/ontology-python-sdk](https://github.com/ontio/ontology-python-sdk)
Demo

Smart contract development using Punica and SmartX – The Basics

Prerequisites

- Punica – Ontology dApp development framework
- SmartX – Ontology’s online smart contract IDE and debugger
- Cyano wallet – Google Chrome extension
- Explorer – Web based public tool used to track blockchain activity and transactions in general

1. Project setup
2. Launching SmartX
3. Start writing the program logic
4. Deployment and testing
Have a try now!
Step 1

Setup your Project with Punica

http://punica.ont.io/docs/punica/
Step 2

Code your Smart-Contract with SmartX

https://smartx.ont.io/
Step 3

Code, Compile, Deploy, Test

1. Install Cyano
2. Apply Test Ong from https://developer.ont.io/applyOng
3. ABI
4. Deploy https://explorer.ont.io/testnet
5. Test
Step 4

Build dApp: Interact with Blockchain

http://punica.ont.io/docs/punica/
Try ONT ID now!
Develop with Ontology

Infra

Governance model
Token economics model
Network hypothesis

Design

Smart contract mechanism
Governance contract
Staking / gas generation & circulation
Consensus
Network Infra

Engineering

Biz app(s)

dApp(s) Other dApp(s) FT/NFT/PFT

// ONT ID
Develop with Ontology

Get ready for business

Business

Application

Claim

Proof of the capability to own the digital asset

ONT ID

Token & DToke n

Digital asset

Assetization

Tokenization

Data

Storage

ONTology
WHAT CAN YOU DO?
Welcome to Blockchain World, Seed Developers!

- Get your ONE & ONLY Certificate
- Scan QR code
- Active it on chain with ONT Auth
- Pass to Klein Bottle Challenge
- Join Ontology
- Start your genesis block
Ontology Authenticator

Self-sovereign Identity & Personal Claims Tool
Ontology Klein Bottle Challenge

- Technical capability and marketing resources
- Full-scale incubation support and global community support
- Investment from Ontology and be a core part of the Ontology ecosystem

$20,000 ONG Rewards Available to Each University

To find out more, please visit ont.io
To be part of future // Join Ontology

Join Ontology today, be a part of the future

Part-time

Tech Contributor

Community Coordinator

Full-time
Scan and join Ontology
Ask anything about **Ontology** now
TRUST Redefined

ont.io

contact@ont.io

OntologyNetwork

ONTnetwork

Ontology

https://discord.gg/pQRHtbD

Scan to download ONT AUTH