WHO and WHAT IS UNI? - Internet of Everything Company

• **UNI Overware** - to connect everything*1
  – Projects the Real world onto the Cyber world utilizing objects (looks like Vertex on graph data)
  – Utilizing enhanced IPv6 addressing space
  – Extends TCP/IP mechanism with Multi-dimensional vertex service
  – Tracks all trail of visited Vertices and Current Vertex as Context*1
  – The Context of vertices is recorded over the network as a UNI Chain*3

• **UNI Underware** (antonym of Overware) - to secure the privacy
  – E2E: White List Gateway and InterVault
  – Confetti: Secret Sharing, distribution by MDVS
  – Fractal Index: Distributed data are indexed with Fractal Compression

*1: Everything: in UNI system “Everything” contains not only devices and services but also human, events and concepts.
*2: Context: Record all transition and strolling oVo history in “white board” as context user behave and stroll among Arbitrary vertices.
*3: UNI Chain: similar to block chain
Map the Real World to the Cyber World

Cyber World
UNI Address($2^{256}$) Address

Real World
IPv4($2^{32}$), IPv6($2^{128}$) Address

A Sensor

B Appliance

Computer

C Service

D Mobile

Events

Community

Concepts

Human
Map the Real World to the Cyber World

Extends TCP/IP mechanism with Multi-Dimensional Vertex Service (SNIF: Social Network InterFace)
oVo: An object which has 32-byte ID will give to not only devices but person and abstracted events, like party, meeting. Representing everything as vertex in graph data.

**Key Technology of UNI - Overware Vertex Object - oVo**

- **Interface to profile data of the oVo**
- **Interface to communicate among oVo utilizing UNI Chain based on P2P protocol**
- **Interface to graph database to discover with set-operation**
- **Interface to AI and recommend engines, like Tensor, word2vec**
- **UNI Tensor calculation**
- **Interface to user, supporting HTML and other game engines, like UNITY**
- **Interface to external system API and Devices**

**[oVoId]** (e.g. 2001:da01::1)
- 32Bytes (IPv6 Address x2)
- Unique ID given by UNI + IPv6 Address
What is oVo

- Overware Vertex Object
- oVo is an object which has 32byte ID and is assigned to every single device, person and abstracted concepts, events, like party or meeting.
  - 16bytes for IPv6 address + 16bytes for UNI unique ID
- oVo is representing everything as vertex in graph data.
- oVo has only data contains URI for each interface, no binary inside oVo
  - All binary split from application to manipulate devices and services are stored in integrated application server.
- oVo has a fixed memory space
  - “White Board” is in memory space to store context
  - Exchanging data in Underware
Customer

• Medical
  – Hospital, Medical University, Nursing care service

• Advertising
  – SSP, DSP, DMP
Overview of associations between UNI and outside UNI

**DEMO Case**

```
UNI System
me
Song A
allplay player
roomA
Other oVo
```

```
Song A
XPIF over dlna
allplay Player
XPIF over allpay
Sensor in Room A
XPIF over GPIO
```

**Other Example : Medical Scenario**

```
UNI System
Mr.A
A’s Vital
A’s Vital 2016/09/19
A’s Vital 2016/09/20
A’s Rad Pix
```

```
A’s Smart Phone
XPIF over BLE
Heart Rate Sensor
XPIF over Zigbee
Blood Pressure Sensor
XPIF over DICOM
```

```
Database
```

```
VPI over HTTP
Smart Phone
XPIF over dlna
Heart Rate Sensor
Blood Pressure Sensor
```

```
Other Example : Medical Scenario
```

```
DEMO Case
```

```
Overview of associations between UNI and outside UNI
```
UNI and WoT
Type of communication to WoT system

Consuming

UNI → WoT Servient → Device

Exposing

Device → UNI → WoT Servient → Device
What UNI can/could contribute to WoT?

- UNI can communicate to external service over WoT interface
  - UNI has interface to external system over HTTP and will implement over WoT Interface.

- UNI could run WoT script on UNI system
  - UNI could support AP runtime
  - UNI Supports C, C++, Java, JavaScripts(node.js)
  - Script can add extra UNI functions as below
    - SNIF (Graph database interface), VPI (User Interface), AIP (AI engine interface), XPIF (binary interface), PROFILE (interface to personal data through SAML, FIDO, X.509 and other protocol)
    - SNIF is using general graph theory with triple to find oVo connecting to “me”

- UNI can manage lifecycle of script, Things Description and binaries
  - UNI oVo (vertex) has mechanism to keep itself up to date
DEMO
Configuration for Demonstration

A Man moves from living room to bed room
Continuous play
Demo Use Case

• Connect Devices to UNI Network with UNI UPnP
  – Discover all devices and songs connecting to network through UNI boot-up process
  – Discover A new Device with UNI UPnP when connect
  – Add new vertices in database dynamically with information from UNI UPnP
  – Get A seed oVo*1 from arbitrary cloud
  – Instantiates A oVo in local host then the oVo connect to the device
  – Then the device become operable on UNI Network
  – Play A song over both allplay and dlna protocol

• Play song in multiple rooms with context
  – Play music in a living room
  – Select and change song agnostic to a protocol of music player
  – Song follows user as moving to other room (since context of A is changed by moving)

*1: Seed oVo: Each type of seed oVo has minimum information to create oVo, and is stored in local or cloud.
What you can see in demo

• Control all devices in home from A device not connected to home network
• Discover all songs and music players over several protocol
  – You can see the activity on graph data monitor and console
  – Relation among vertices will be created automatically
• Instantiate several oVo of device dynamically when connect
  – You can see the activity on graph data monitor and console
  – Relation among vertices will be created automatically
• Play A song over both allplay and dlna protocol
  – Play on mpeg player and allplay player
• Continuous play as changing context
  – Simulates moving by swtich
Next Step

- Implement WoT Interface
  - To consume data from other servers over WoT interface
  - To expose data from UNI system over WoT interface

- Build a joint scenario
UNI Software Deliverables

• UNI Peripherals SDK - Open source, release date: Q1, 2017
  – C, C++, Java, JS(node.js)

• UNI Chain SDK - Open API, release date: TBD
  – C, C++, Java
UNI Devices

• UNI ZMOT
• Home Server with UNI Stack
A Pretty Simple Device user can order from this with one push, and could be a advertising device. Item to shop is configurable.
UNI Home Server

• Functions
  – Host for internal oVo at home
  – Host of UNI Stack
  – Meson to authorize when bind multiple sub-graph
  – Super Node to manage distributed data over Peer-to-Peer protocol
  – Hosts fundamental oVos to get seed oVo from internet