Picominer®

(IoT Edge Gateway)

Open source release of a reference implementation of

- ISO/IEC/IEEE 21450:2010(E) TEDS and
- ISO/IEC/IEEE 21451-1:2010(E) NCAP software

- P V S Maruthi Rao
  (maruthi@vidcentum.com)

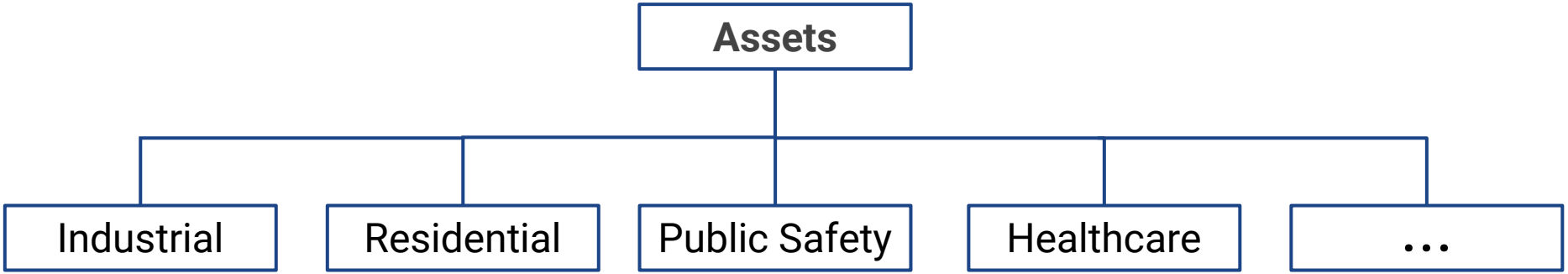
20-Oct-2018

Version 1.0f

Picominer Goal

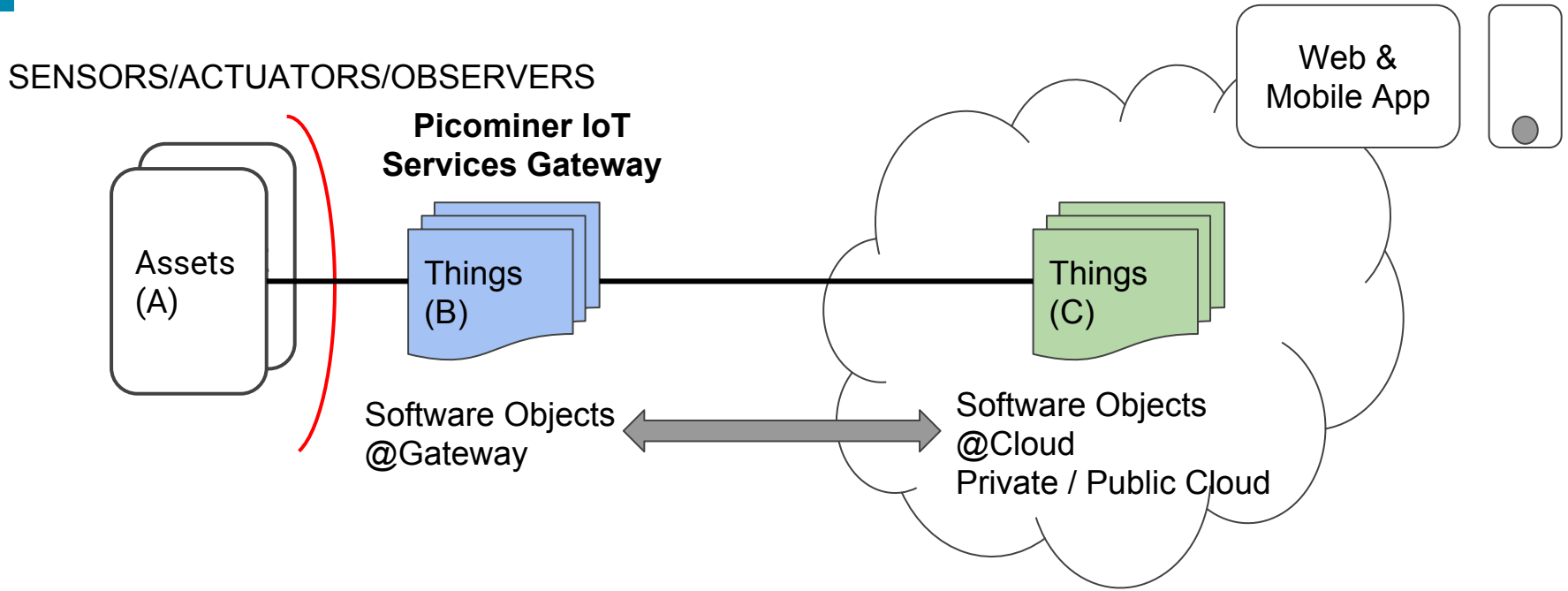
"To make the edge analytics and connected applications super simple."

Assets Create Business **Value**



- Businesses manage “**assets**” and create “**value**”.
- “**Assets**” are “**Things**” in the field of IoT.
- IoT is applicable to all business where “**assets**” are interconnected and interoperated with energy, material and **information**.

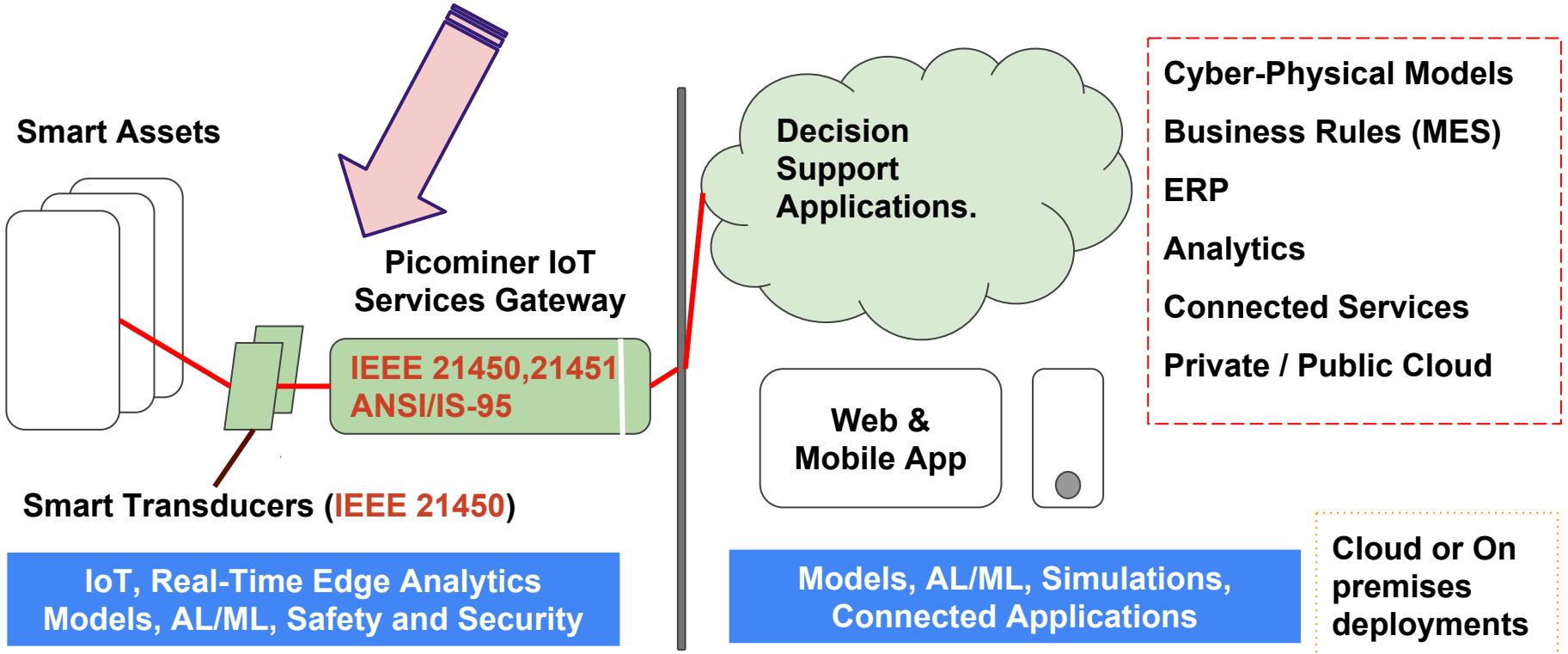
Assets ⇒ Things



- Assets {A} exist as **Things** {B} at the Edge and in the cloud {C}.
- This is the crux of IoT designs with ISO/IEC/IEEE 21450:2010(E) TEDS and ISO/IEC/IEEE 21451-1:2010(E) NCAP.

Picominer: IoT Edge Systems Overview

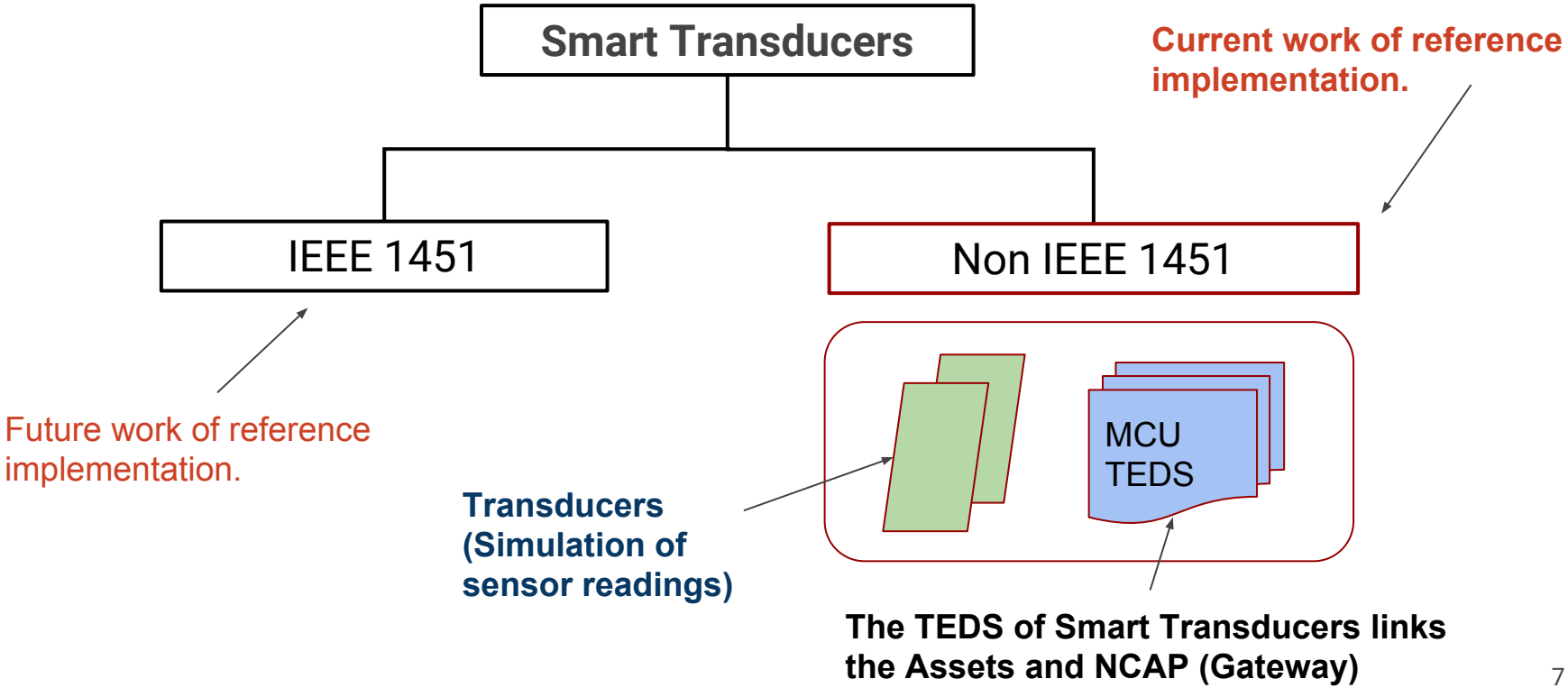
Reference implementation of ISO/IEC/IEEE 21450:2010(E) TEDS and ISO/IEC/IEEE 21451-1:2010(E) NCAP



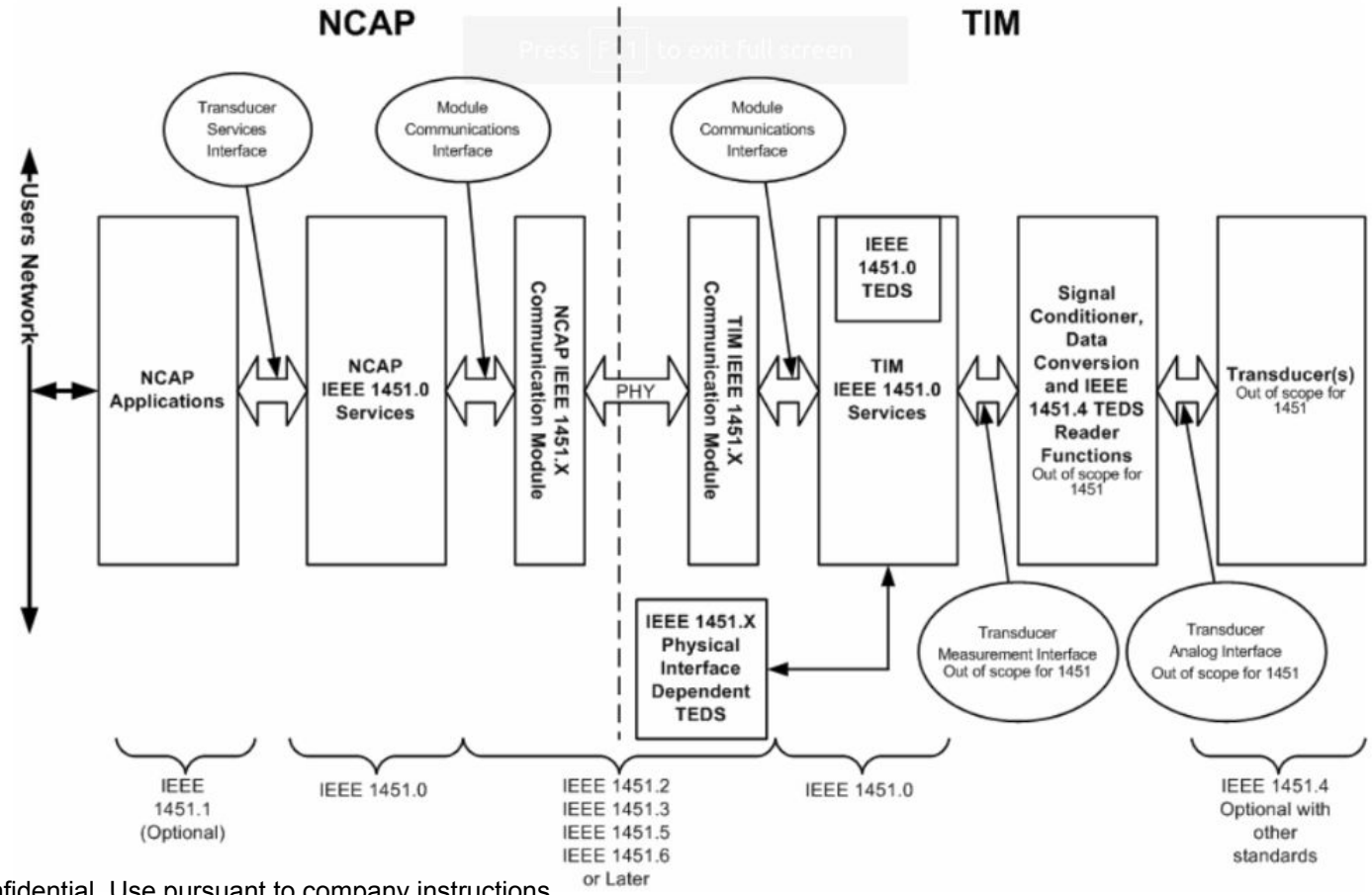
ANSI/IS-95 - for Smart Assets

```
//
// ALL PHYSICAL ASSETS ARE MAPPED IN THE ROW MAJOR ORDER.
// Enterprise / Factory Assets
//   -> Site(s)
//     -> Area(s)
//        -> Work Center(s)
//           -> Unit(s)/Machine(s)
//              -> Equipment Module(s)
//                 -> Control Module(s)
//                    -> Shelf(s)
//                       -> Slot(s)
//
// Slots are numbered from {Top, Left of Site} to
//                         {Bottom, Right of the Site}
//
```

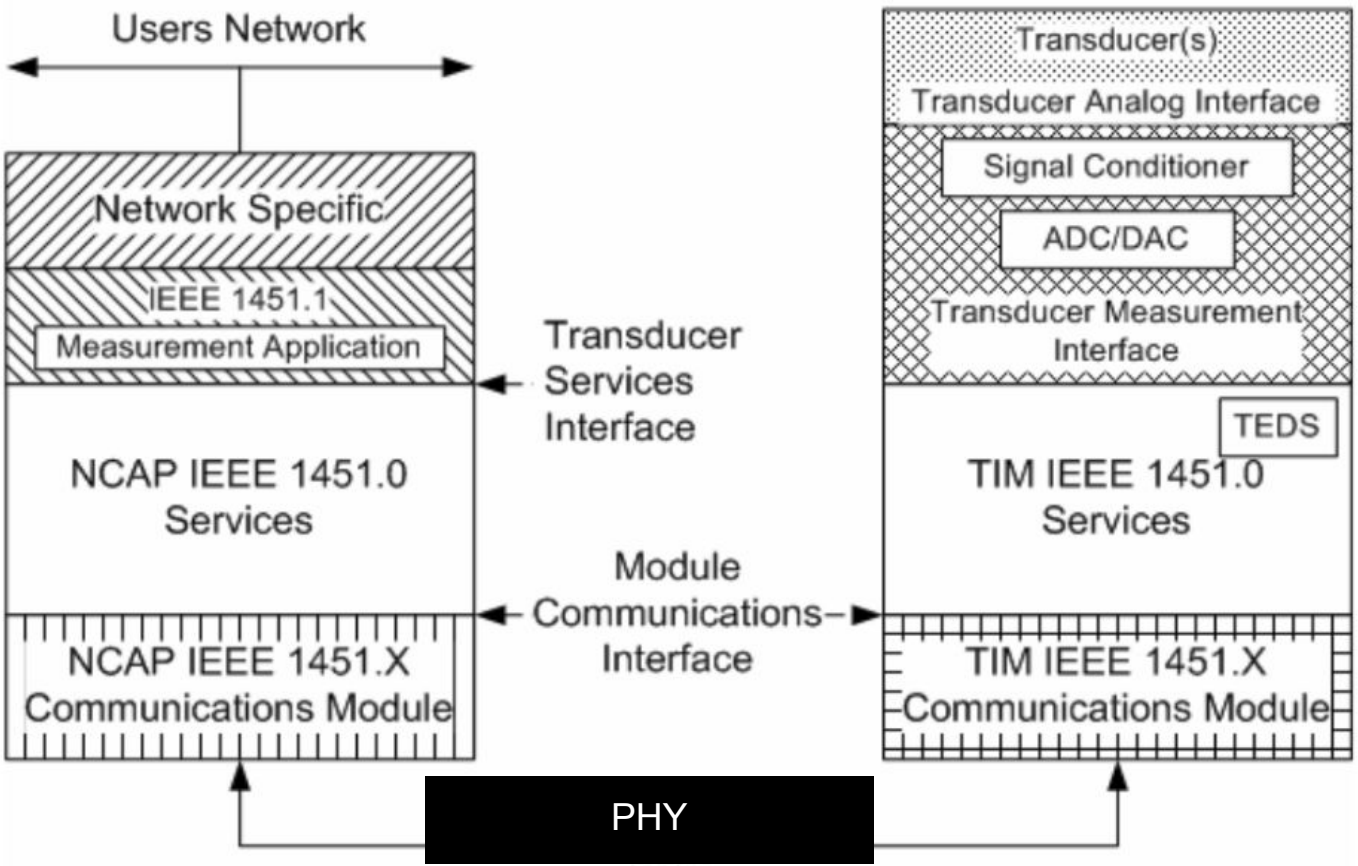
Smart Transducers



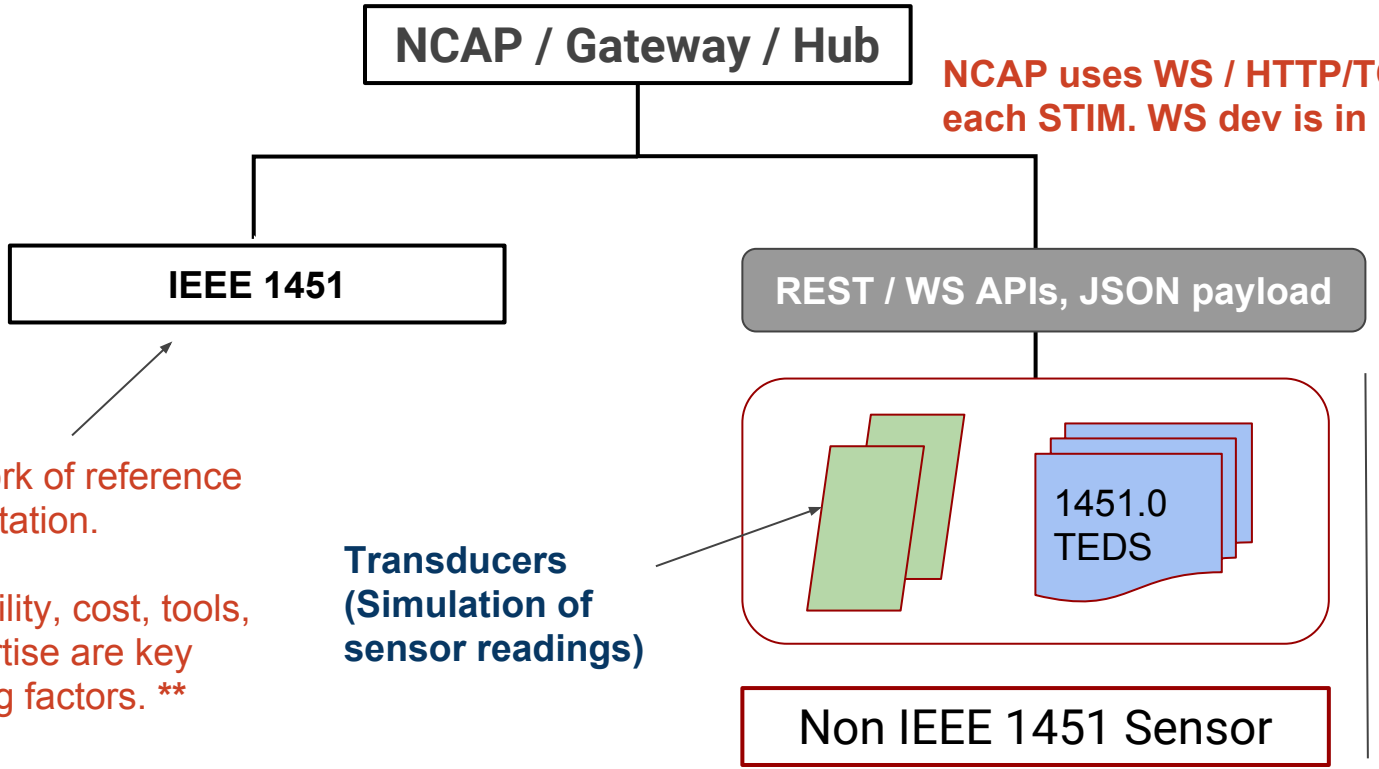
Let’s take a look at the 1451 Standard Interfaces



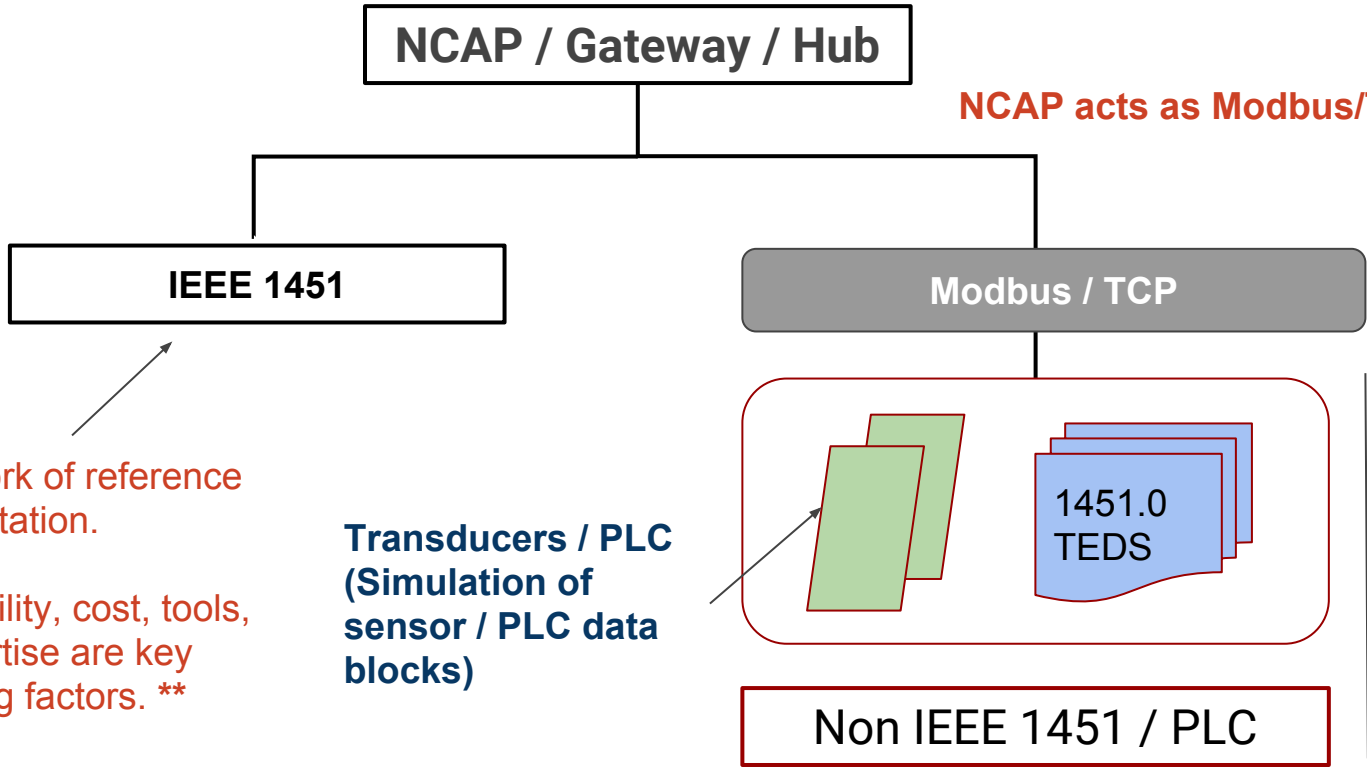
Let’s take a look at the 1451 Standard Interfaces



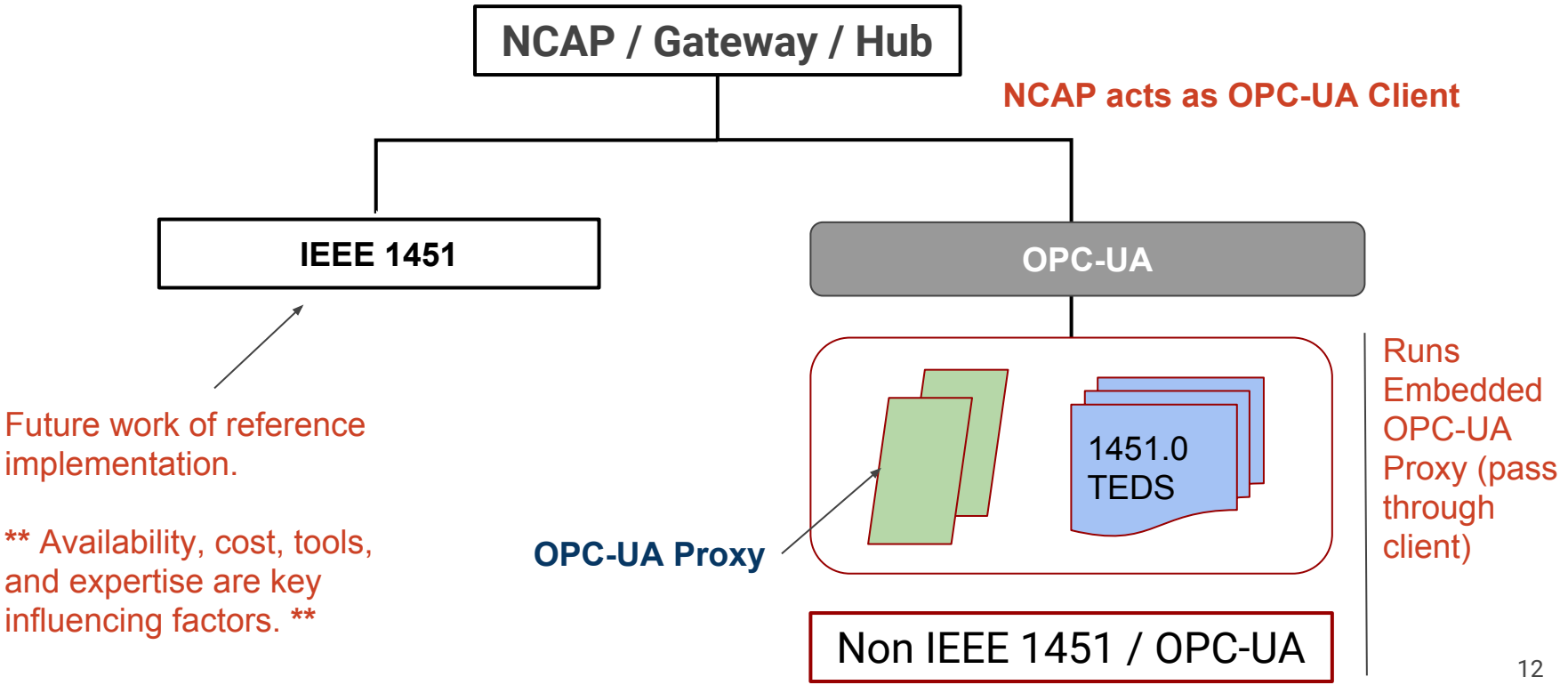
The Interface Architecture NCAP - STIM (WS / HTTP)



The Interface Architecture NCAP - STIM (MODBUS/TCP)



The Interface Architecture NCAP - OPC-UA (Client)



The Interface Architecture NCAP - STIM

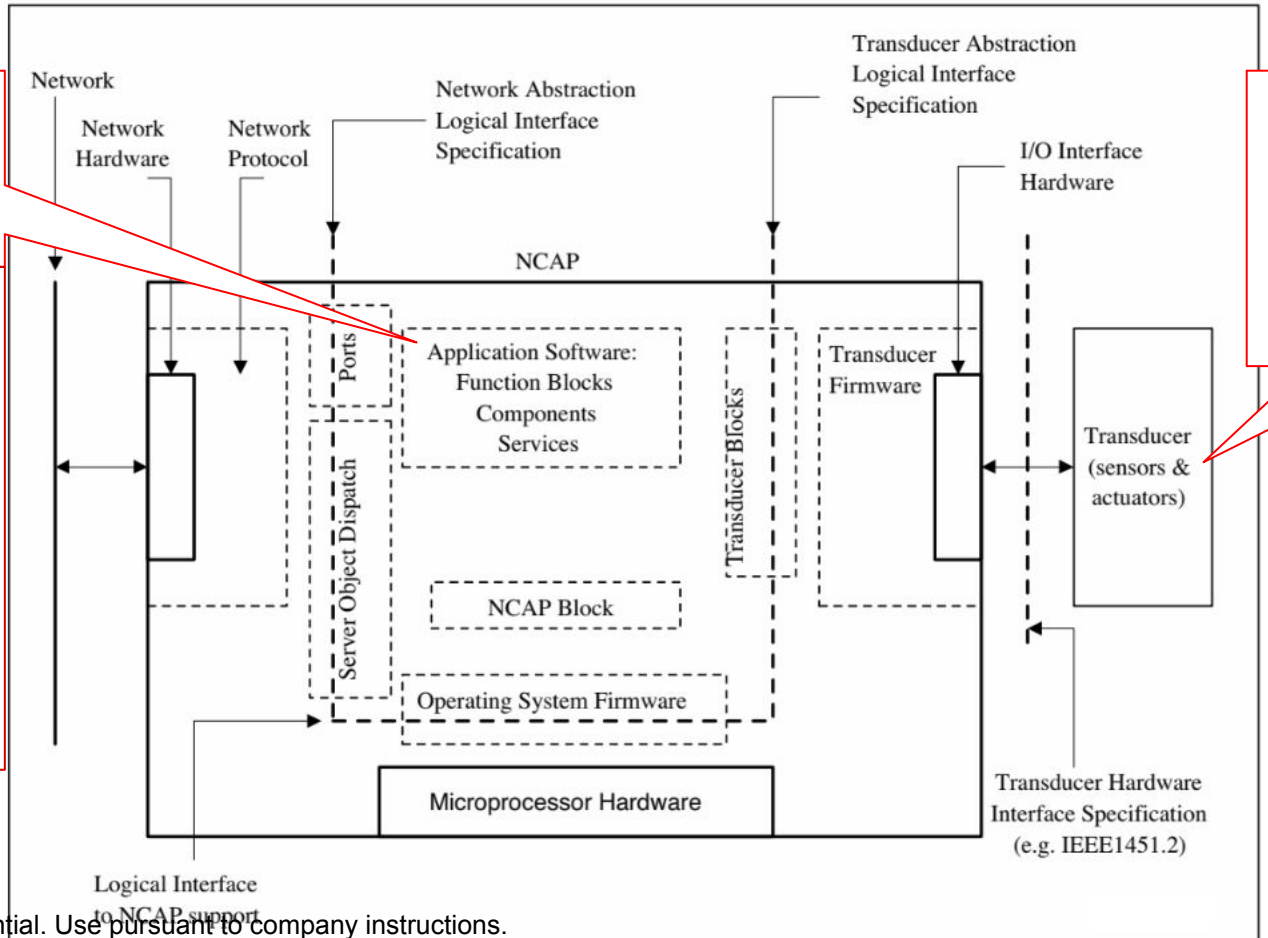
```
{ TIM APIs

    { "site_id", 0 }, { "address", "0.0.0.0" },
    { "port", 8888 }, { "scheme", "http" },

    // APIs exposed by the TimWebServer.
    { "get_and_reload_transducer_defs_api", "get-reload-transducer-defs" },
    { "get_and_reload_product_rules_api", "get-reload-product-rules" },
    { "get_and_reload_assets_api", "get-reload-assets" },
    { "get_and_reload_erp_node_api", "get-reload-erp-node" },
    { "get_and_reload_opc_node_api", "get-reload-opc-node" },
    { "get_and_reload_cyber_physical_config_api",
            "get-reload-cyber-physical-config-api" },
    { "get_iot_api", "iot" },
    { "get_erp_api", "erp" },
    { "get_opc_api", "opc" },
    { "get_default_api", "" },
    ...
}
```

Let's take a look at the 21451 NCAP Interfaces

Things (AI/ML models at the Edge)

1. Realtime Edge Analytics
2. MES at the Edge
3. Safety
4. Security
5. OPC-UA client

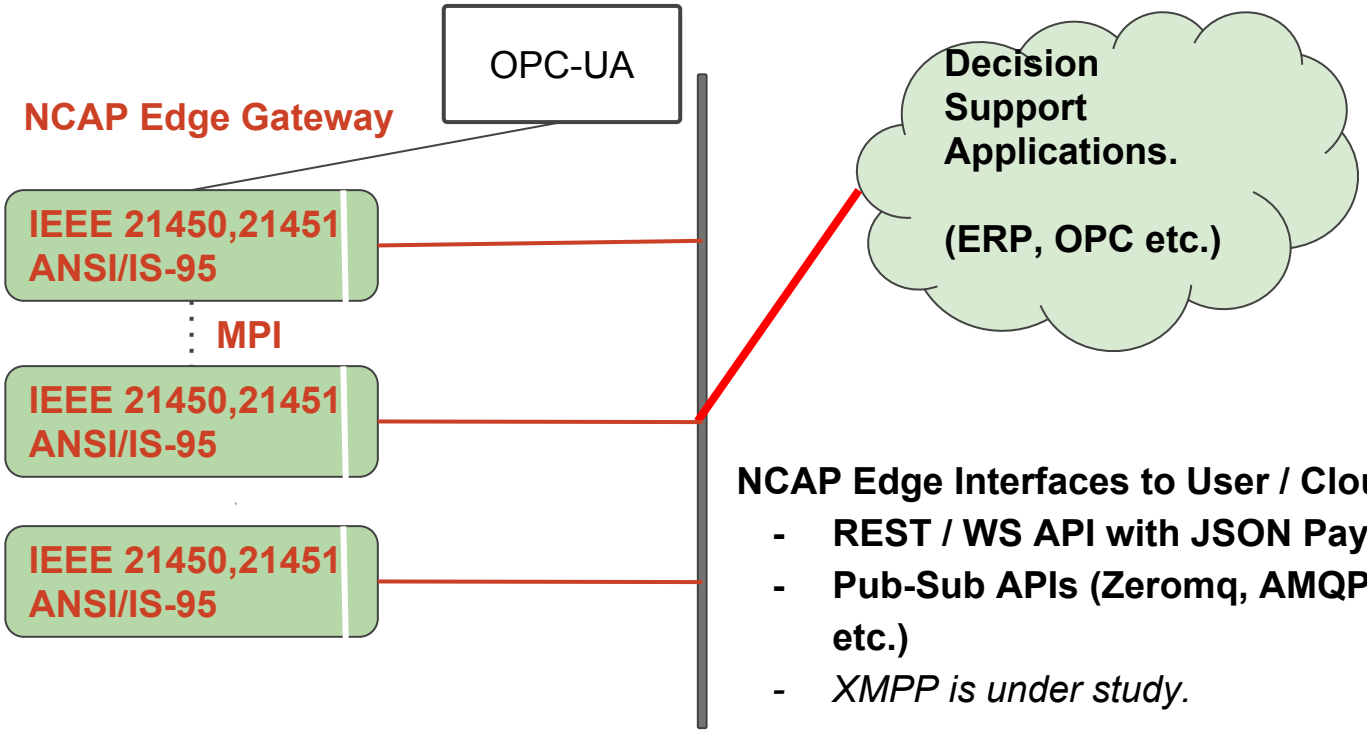
.. etc.



TEDS Transducer Electronic Data Sheets (=> Data Models of Transducers)

Single NCAP Node - Cloud / User Applications

Reference implementation of ISO/IEC/IEEE 21450:2010(E) TEDS and ISO/IEC/IEEE 21451-1:2010(E) NCAP

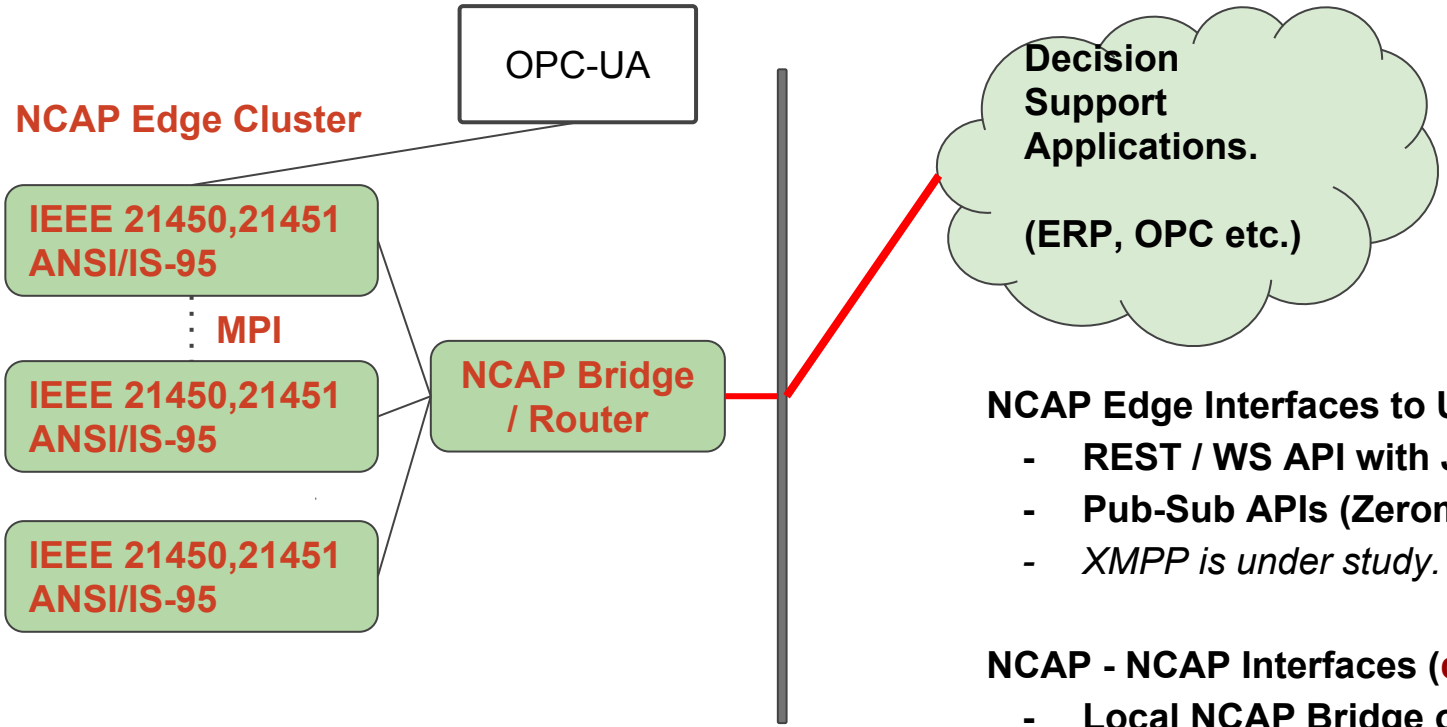


NCAP Edge Interfaces to User / Cloud:

- **REST / WS API with JSON Payload**
- **Pub-Sub APIs (Zeromq, AMQP. XMPP etc.)**
- *XMPP is under study.*

NCAP Cluster - Cloud / User Applications

Reference implementation of ISO/IEC/IEEE 21450:2010(E) TEDS and ISO/IEC/IEEE 21451-1:2010(E) NCAP



NCAP Edge Interfaces to User / Cloud:

- **REST / WS API with JSON Payload**
- **Pub-Sub APIs (Zeromq)**
- *XMPP is under study.*

NCAP - NCAP Interfaces (dev in progress):

- **Local NCAP Bridge or Router**
- **Local MPI infrastructure**

The Interface Architecture NCAP - Cloud / User Applications

{ NCAP APIs

```
    { "site_id", 0 },
    { "address", "127.0.0.1" }, { "port", 7777 }, { "scheme", "http" },
    { "get_config_change_api", "get-config-change" },
    { "get_config_api", "get-config" },
    { "post_config_api", "set-config" },

    // Cyber Physical Configuration
    { "get_cyber_physical_config_api", "get-cyber-physical-config" },
    { "get_new_cyber_physical_config_api", "get-new-cyber-physical-config" },
    { "post_cyber_physical_config_api", "post-cyber-physical-config" },
    { "post_new_cyber_physical_config_key", "cyber-physical-config" },

    // Transducer Definitions
    { "get_transducer_defs_api", "get-transducer-defs" },
    { "get_new_transducer_defs_api", "get-new-transducer-defs" },
    { "post_transducer_defs_api", "post-transducer-defs" },
    { "post_new_transducer_defs_key", "transducer-defs" },
```

...continued to 2 page

The Interface Architecture NCAP - Cloud / User Applications

...continued NCAP APIs from page 1

```
// Product Rule Engine APIs
{ "get_product_rules_api", "get-product-rules" },
{ "get_new_product_rules_api", "get-new-product-rules" },
{ "post_product_rules_api", "post-product-rules" },
{ "post_new_product_rules_key", "product-rules" },
// Asset Management APIs
{ "get_assets_api", "get-assets" },
{ "get_new_assets_api", "get-new-assets" },
{ "post_assets_api", "post-assets" },
{ "post_new_assets_key", "assets" },
// ERP Node API
{ "get_erp_node_api", "get-erp-node" },
{ "get_new_erp_node_api", "get-new-erp-node" },
{ "post_erp_node_api", "post-erp-node" },
{ "post_new_erp_node_key", "erp-node" },
```

...continued to 3 page

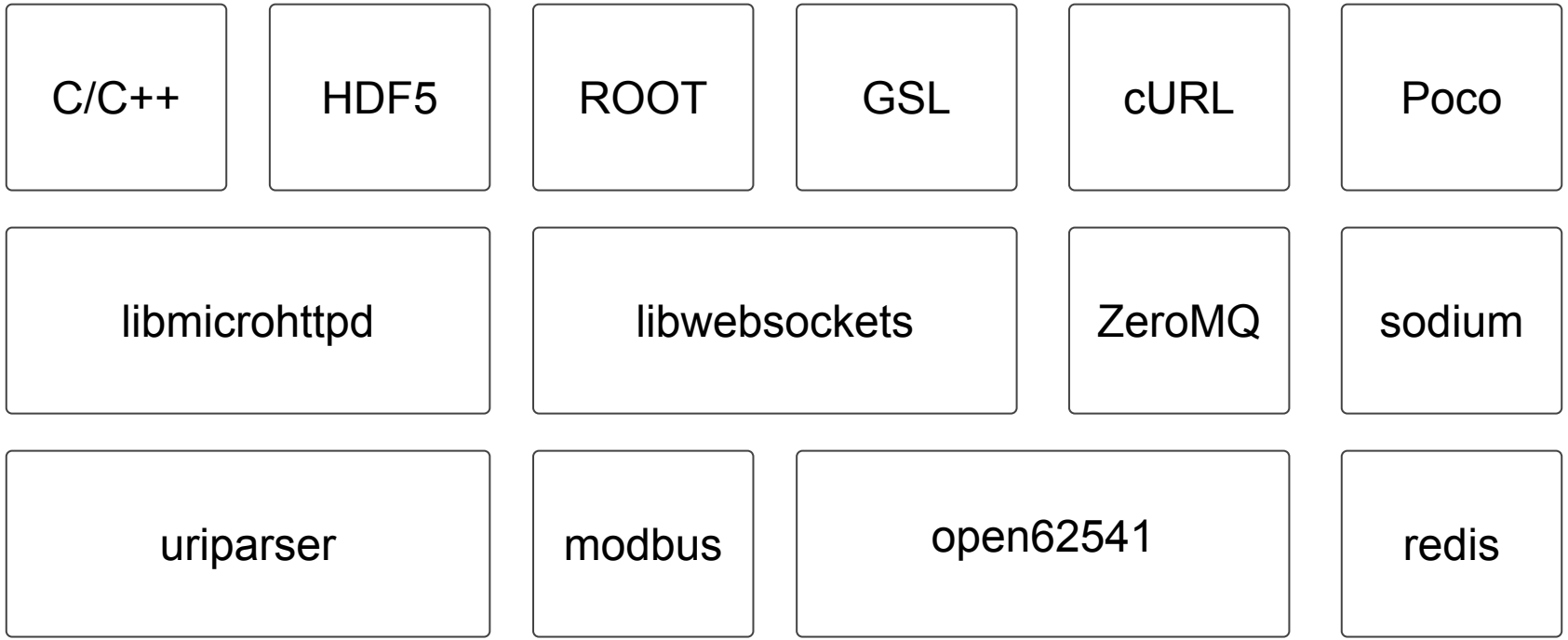
The Interface Architecture NCAP - Cloud / User Applications

...continued NCAP APIs from page 2

```
    // OPC Node API
    { "get_opc_node_api", "get-opc-node" },
    { "get_new_opc_node_api", "get-new-opc-node" },
    { "post_opc_node_api", "post-opc-node" },
    { "post_new_opc_node_key", "opc-node" },

    // Library Management APIs
    { "get_transducer_library_api", "get-transducer-library" },
    { "get_product_library_api", "get-product-library" },
    { "get_assets_library_api", "get-assets-library" },
    { "get_preset_library_api", "get-preset-library" },
    { "get_erp_library_api", "get-erp-library" },
    { "get_opc_library_api", "get-opc-library" },

    ...
}
```

Implementation Notes

```
// NCAP, TEDS software is being developed in C/C++
// Note: Implementation can be simplified further.
```



Implementation Notes

```
// Timestamping, Security
// Note: Implementation is continuously being improved...
```

- NCAP is configured with local time and timezone settings.
- STIM synchronizes with NCAP. NCAP acts as time source to STIM.
- Realtime timestamping is done in the STIM.
- Every packet at NCAP is time-stamped with local time with millisecond resolution.
- User direct access to the NCAP is restricted.
- Only selected NCAP gateways can communicate with the cloud apps.
- SSL, TLS used while communicating with the external applications.
- TEDs are currently ‘hand coded’. A simple TED editor is planned.

Implementation Notes

// **Source Code**

// Note: Implementation can be simplified further.

- Operating Systems: Linux, Android
- Software Developed in C/C++
- Architecture: X86_64 (64 Bit), ARM (32 Bit)
- Hardware Platforms: Single Board Computers such as Raspberry Pi, Beaglebone, Cloud Virtual Machines, Workstations etc.
- **GitHub URL:**
  - https://github.com/picominer/IoT-Gateway
  - https://github.com/picominer/
- **License:** MIT, Vidcentum R&D
- **Released by:** P V S Maruthi Rao, Director Vidcentum R&D Pvt Ltd.
- **Product Line:** The reference implementation is released under Picominer Program (IoT Edge Analytics Gateways) of Vidcentum R&D Pvt Ltd.
  (Picominer: https://picominer.com)

Thank You

Quick Demo:
Supported Browsers: Google Chrome, Opera
Network port to open: 7681 for WSS traffic
https://dashboard.picominer.com/v/2.0
Click on ‘Direct Login’
(no need to register to view demo)

Demo

Q&A