CNS/CP
Connection Profiles
System-of-Systems

World of Industries

IBM: World’s 4 trillion dollar challenge (Jan/2010)

Flock of Birds
Relationships in System-of-Systems

- Interface Node
- System-of-Systems (SoS)
- Relationship
- Needs
- Capabilities
- Constituent System (CS)
- Client Node
- SoS
- CS
- Server Node
- Connection Profile (Model)
- Connection Instance(s)
- proto.example.sys
Connection Profile Mechanism

Can **consume** proto.example.sys

Can **serve** proto.example.sys

Properties for: communication, security, location, credentials, trust, commerce, etc.

Client System

Server System

Virtual Node
Non-Twin Use Case

Connection Profile (Model)

Can consume
proto.example.abc

Can serve
proto.example.abc

Properties for: security, location, credentials, trust, commerce, etc.
## Connection Profile Example

<table>
<thead>
<tr>
<th>Name</th>
<th>xyz.ics</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>2</td>
<td>XYZ ICS App Server</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
<td>Visualization Application</td>
</tr>
<tr>
<td>Owner</td>
<td>XYZ Systems, Inc.</td>
<td>It serves visualization data for dashboards and other UI needs for all versions of XYZ's ICS automation system since 2003.</td>
</tr>
<tr>
<td>Title</td>
<td>XYZ Control &amp; Automation System</td>
<td></td>
</tr>
<tr>
<td>Specs</td>
<td><a href="http://www.example.com/sys/ics.html">www.example.com/sys/ics.html</a></td>
<td></td>
</tr>
</tbody>
</table>

**Server**

<table>
<thead>
<tr>
<th>Property</th>
<th>uri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>yes</td>
</tr>
<tr>
<td>Desc</td>
<td>Attribute for URI of the endpoint XYZ ICS server.</td>
</tr>
<tr>
<td>Sample</td>
<td><a href="http://10.0.5.123/dev/sys/api.html?format=json">http://10.0.5.123/dev/sys/api.html?format=json</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>prot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>no</td>
</tr>
<tr>
<td>Desc</td>
<td>Available protocols from this server. Multiple ok. bacnet, json (default), xml, xyzsys, etc.</td>
</tr>
<tr>
<td>Sample</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>no</td>
</tr>
<tr>
<td>Desc</td>
<td>The monthly service cost to provide information.</td>
</tr>
<tr>
<td>Sample</td>
<td>USD 2.50 (default $0)</td>
</tr>
</tbody>
</table>

**Client**

<table>
<thead>
<tr>
<th>Property</th>
<th>api-key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>yes</td>
</tr>
<tr>
<td>Desc</td>
<td>API key required to communicate with the server</td>
</tr>
<tr>
<td>Sample</td>
<td>AsoOs8xiesoOs8xie6qWTG2HZU2HZDSUU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>client-geo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>yes</td>
</tr>
<tr>
<td>Desc</td>
<td>Latitude &amp; longitude of the location of XYZ device.</td>
</tr>
<tr>
<td>Sample</td>
<td>37.751, -97.822</td>
</tr>
</tbody>
</table>

Link to sample profile: xyz.ics
Connectivity Naming System (CNS)
Endpoint vs. Relationship Views

Endpoint-Centric

We focus on Endpoints (a.k.a. silos) as that’s where we see the value is. We see connections mainly as communications between Endpoints.

Relationship-Centric

Endpoints are where computing, people, and assets are. The greater value to focus on are the Relationships between Endpoints.
Current Paradigm: Endpoint Centric

Challenges...

1: Endpoints set, maintain, and synchronize their connection metadata.

2: Each Connection is unique and nothing is normalized!

3: Complex, convoluted, and inefficient.
Relationships Are Complex
A Relationship Made up of Connections

Normalized in two ways...

1: The CNS mechanism to discover/match Nodes using compatible CPs.

2: Connection Profiles (CP) exchange metadata based on agreed rules.

And, it is all dynamically **composable**!
CNS/CP Workflow

<table>
<thead>
<tr>
<th>cp: test.abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
</tr>
<tr>
<td>foo1</td>
</tr>
<tr>
<td>foo2</td>
</tr>
</tbody>
</table>

Startup Node Publishing

Foo (Client)
Registers a Node ID, Context, CPs, Roles, and Properties

Pub/Sub

Orchestrator
Creates Virtual Foo & looks for matching Nodes in the Context given

A Match Starts Subscription

Foo (Client)
Receives an Event when a Connection Instance is created

Pub/Sub

Orchestrator
When matched with Server Far, sends Connection ID & Properties

Ongoing Bi-Directional Subscription

Foo (Client)
Sets Foo Properties
Receives Events when Far Properties change

Orchestrator

Far (Server)
Receives Events when Foo Properties change
Sets Far Properties
Client Node Publishing
{
  "nodeID": "did:example:123456789abcdefgijhjk",
  "context": "context-string",
  "profiles": {
    "cp": "test.abc",
    "role": "client",
    "properties": {
      "foo1": "value for foo1",
      "foo2": "value for foo2"
    }
  }
}

Server Match Found
{
  "nodeID": "did:example:123456789abcdefgijhjk",
  "matchID": "did:example:abcdefgijhjk123456789",
  "connectionID": "NvdmaNDNLPFR9d9dB",
  "cp": "test.abc",
  "properties": {
    "far1": "value for far1",
    "far2": "value for far2"
  }
}

Client>Server Subscription
{
  "connectionID": "NvdmaNDNLPFR9d9dB",
  "properties": {
    "foo1": "value for foo1",
    "foo2": "value for foo2"
  }
}

Server>Client Subscription
{
  "connectionID": "NvdmaNDNLPFR9d9dB",
  "properties": {
    "far1": "value for far1",
    "far2": "value for far2"
  }
}
Systems, Context, and Connections

System 1
- cp:a.x/s
- cp:trust.x/c
- cp:c.x/c
- cp:d.x/c

System 2
- cp:trust.x/c
- cp:e.x/s
- cp:f.x/s

System X
- cp:trust.x/c
- cp:c.x/s

System N
- cp:trust.x/s

Context-1
- cp:c.x/s

Context-2
- cp:trust.x/s

Server
- A
- B
- C
- D

Client
- E
- F

14
CNS/CP and the OSI Stack - For Feedback

1. Physical
- Physical structure
- Coax, Fiber, Wireless, Hubs, Repeaters

2. Data Link
- Frames
- Ethernet, PPP, Switch, Bridge

3. Network
- Packets
- IP, ICMP, IPSec, IGMP

4. Transport
- End-to-end connections
- TCP, UDP

5. Session
- Synch & send to port
- API’s, Sockets, WinSock

6. Presentation
- Syntax layer
- SSL, SSH, IMAP, FTP, MPEG, JPEG

7. Application
- End User layer
- HTTP, FTP, IRC, SSH, DNS

HTTP | CP | CP | HTTP
--- | --- | --- | ---
SSL | CNS | CNS | SSL
Socket | Connection Instance | Connection Instance | Socket
TCP | MQTT, DDS, Sockets, etc. | MQTT, DDS, Sockets, etc. | TCP
IP | Network | Network | IP
WiFi | Data Link | Data Link | Ethernet

“CP” Network
CNS/CP

For more visit cns.cp.io or contact info@padi.io