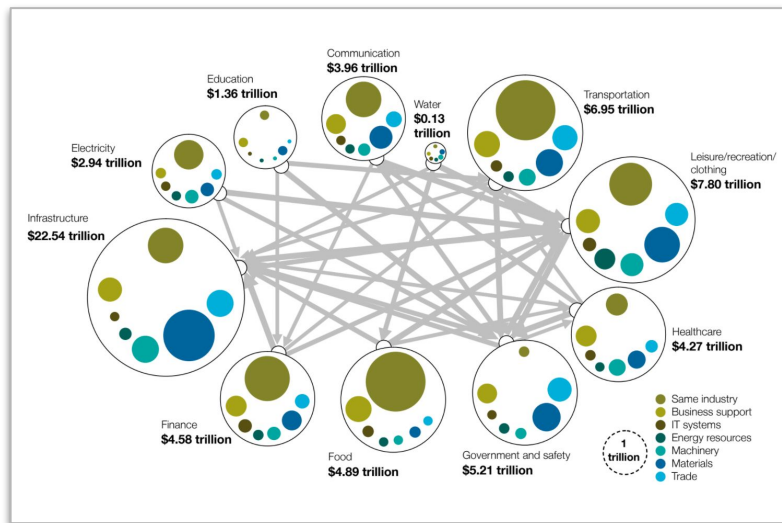# CNS/CP
Connection Profiles

System-of-Systems

World of Industries
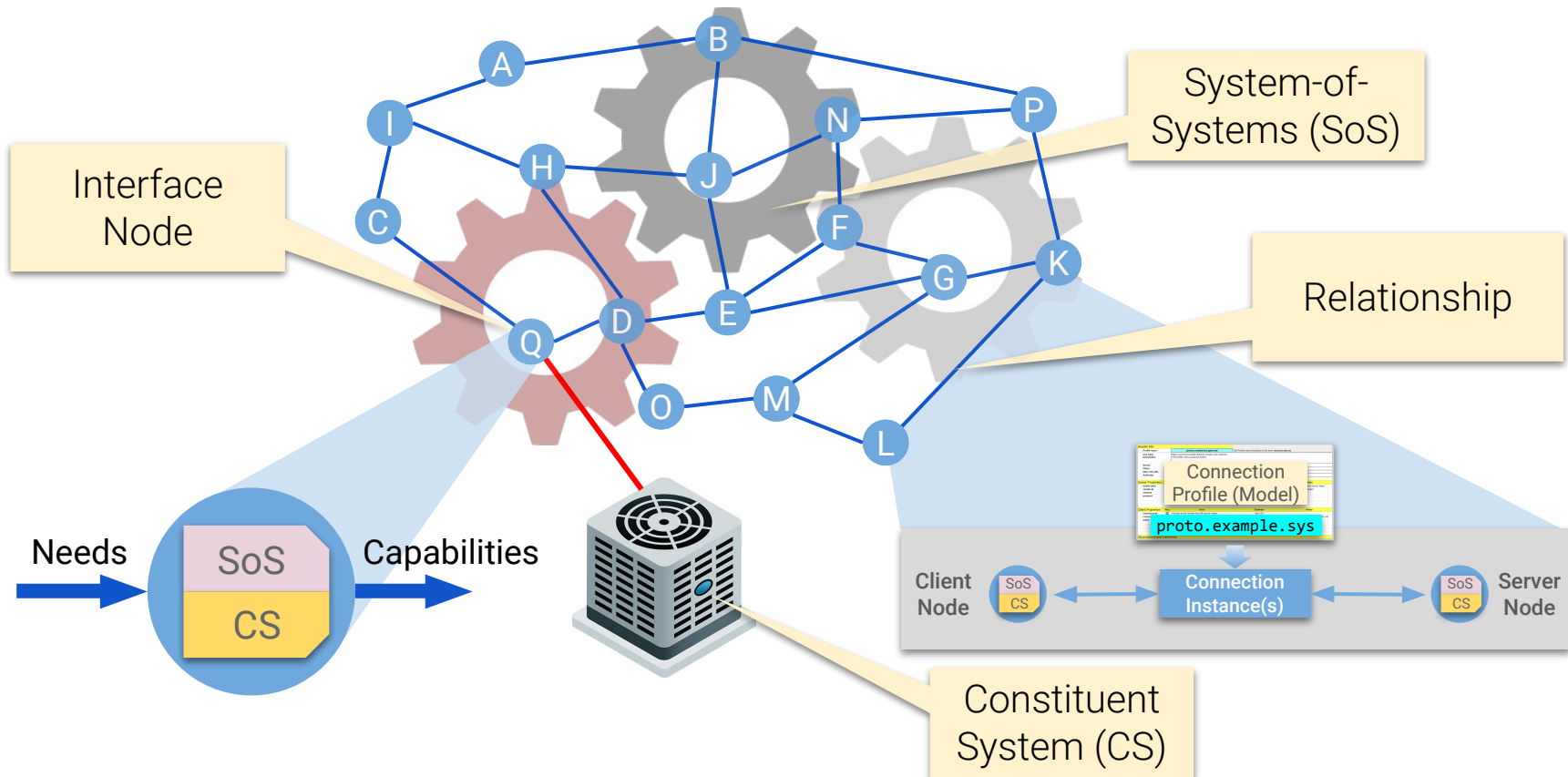
IBM: World's 4 trillion dollar challenge (Jan/2010)

Flock of Birds

Relationships in System-of-Systems

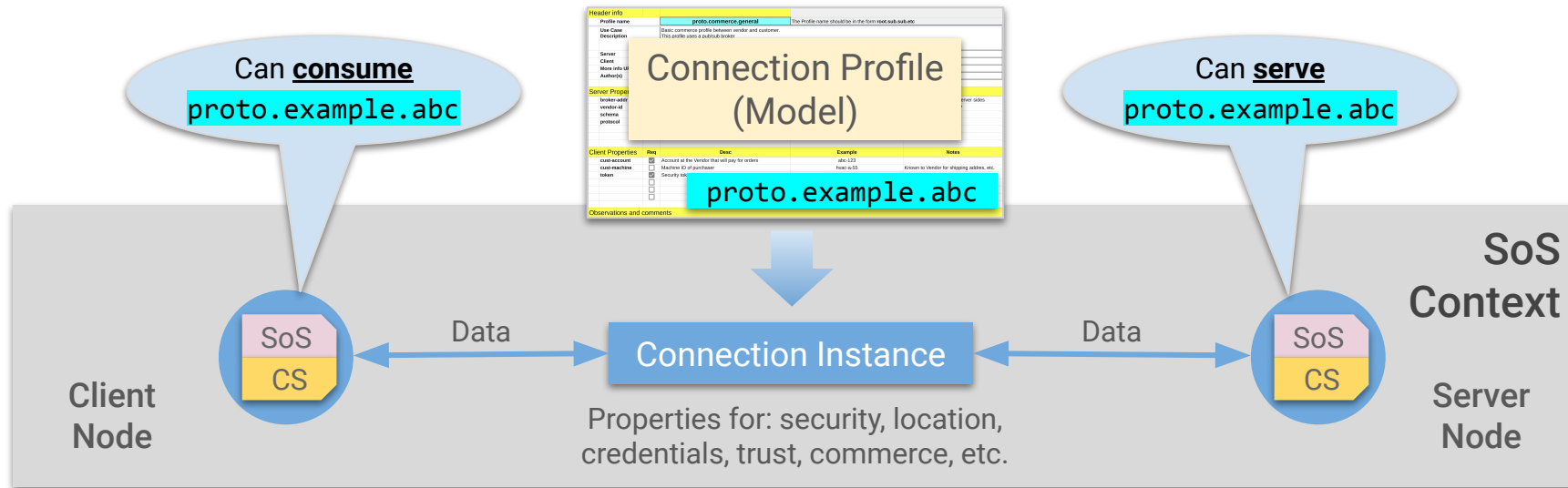Interface Node

System-of-Systems (SoS)

Relationship

Needs → SoS / CS → Capabilities

Connection Profile (Model)

proto.example.sys

Client Node — SoS CS ↔ Connection Instance(s) ↔ SoS CS — Server Node

Constituent System (CS)

3

Non-Twin Use Case

Can **consume**
`proto.example.abc`

Connection Profile
(Model)

`proto.example.abc`

Can **serve**
`proto.example.abc`

SoS
Context

Client
Node

SoS
CS

Data

Connection Instance

Data

SoS
CS

Server
Node

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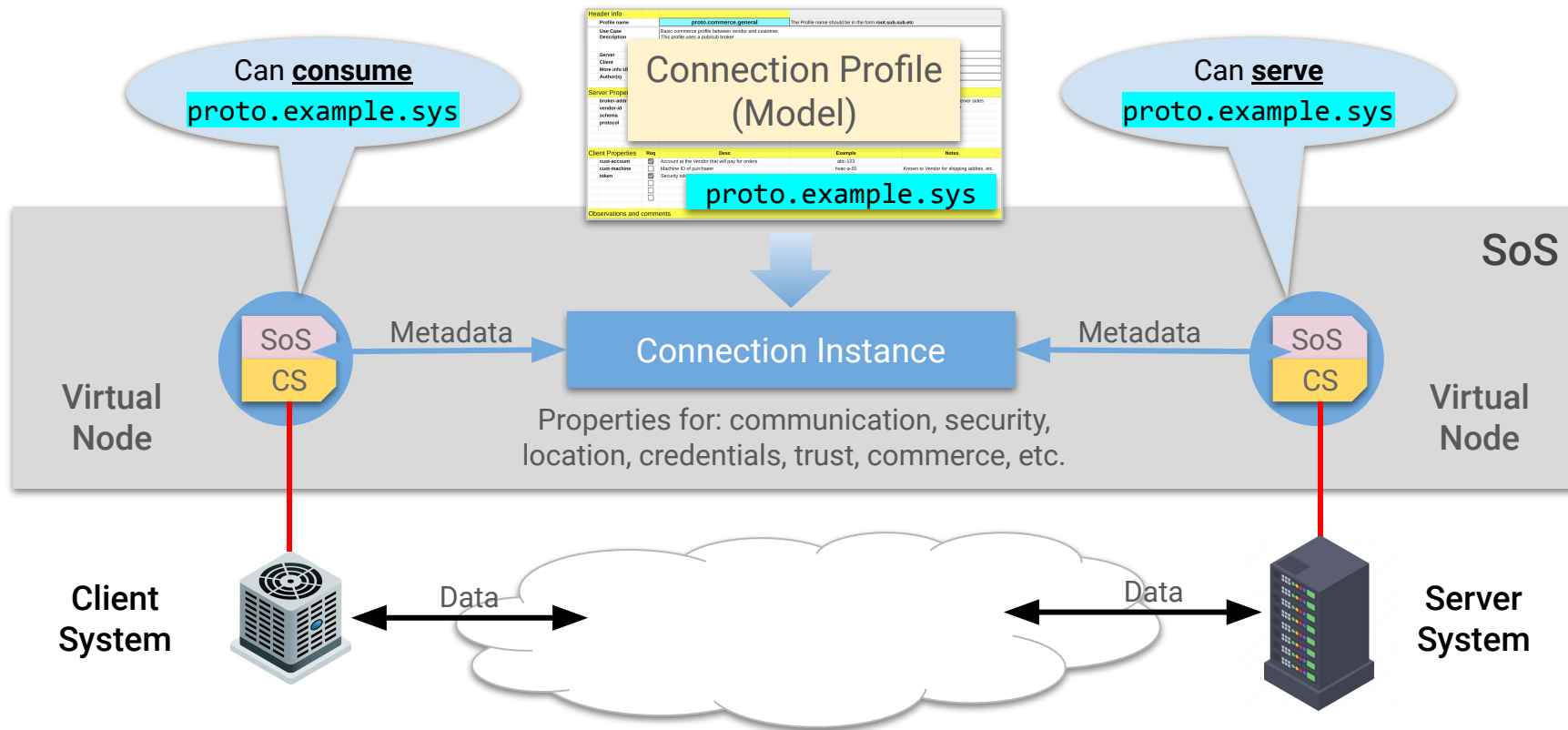
Connection Profile Example

CNS/CP

Name	**xyz.ics**		
Version	**2**	Pub Date	**2020-03-04**
Status	Active		
Owner	XYZ Systems, Inc.		
Title	XYZ Control & Automation System		
Specs	www.example.com/sys/ics.html		

Server	**XYZ ICS App Server**
Client	**Visualization Application**
Desc	Profile for control system for smart buildings. **It serves visualization data for dashboards and other UI needs** for all versions of XYZ's ICS automation system since 2003.

Server

Property	`uri`
Mandatory	**yes**
Desc	Attribute for URI of the endpoint XYZ ICS server.
Sample	http://10.0.5.123/dev/sys/api.html?format=json

Property	`prot`
Mandatory	**no**
Desc	Available protocols from this server. Multiple ok.
Sample	bacnet, json (default), xml, xyzsys, etc.

Property	`cost`
Mandatory	**no**
Desc	The monthly service cost to provide information.
Sample	USD 2.50 (default $0)

Client

Property	`api-key`
Mandatory	**yes**
Desc	API key required to communicate with the server
Sample	AsoOs8xiesoOs8xie6qWTG2HZU2HZDSUU

Property	`client-geo`
Mandatory	**yes**
Desc	Latitude & longitude of the location of XYZ device.
Sample	37.751, -97.822

etc..

Link to sample profile xyz.ics

Connectivity Naming System (CNS)

Connection
CC-BY-ND

`proto.example.sys`
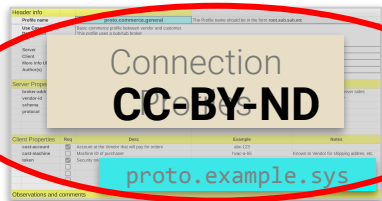
Open Connection
Profile Registry

Directory Interface

Directory Interface

Directory Interface

Client
Nodes

Apache 2.0 License

Server
Nodes

Broker Interface

Broker Interface

Innovation
Happens
Here!

Innovation
Happens
Here!

Broker /
Orchestrator

Innovation Happens Here!

cnscp.io

Endpoint vs. Relationship Views

Endpoint-Centric

Relationship-Centric
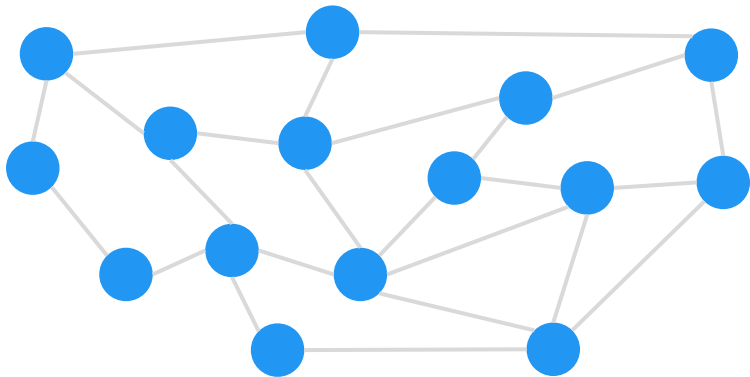
We focus on Endpoints (a.k.a. silos) as that's where we see the value is.

We see connections mainly as communications between Endpoints.
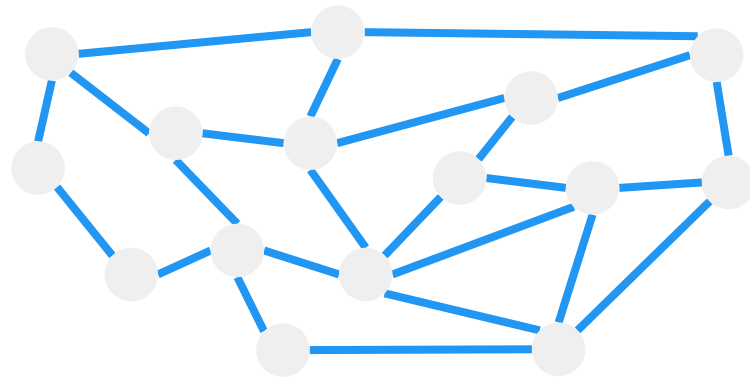
Endpoints are where computing, people, and assets are.

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.

Comms Control		IDs & Secrets
Keys & Certs		URIs & Protocols
Context	Comms Pipe	Trust & Quality
Commerce Info		Business Rules
...		...
Data		**Data**
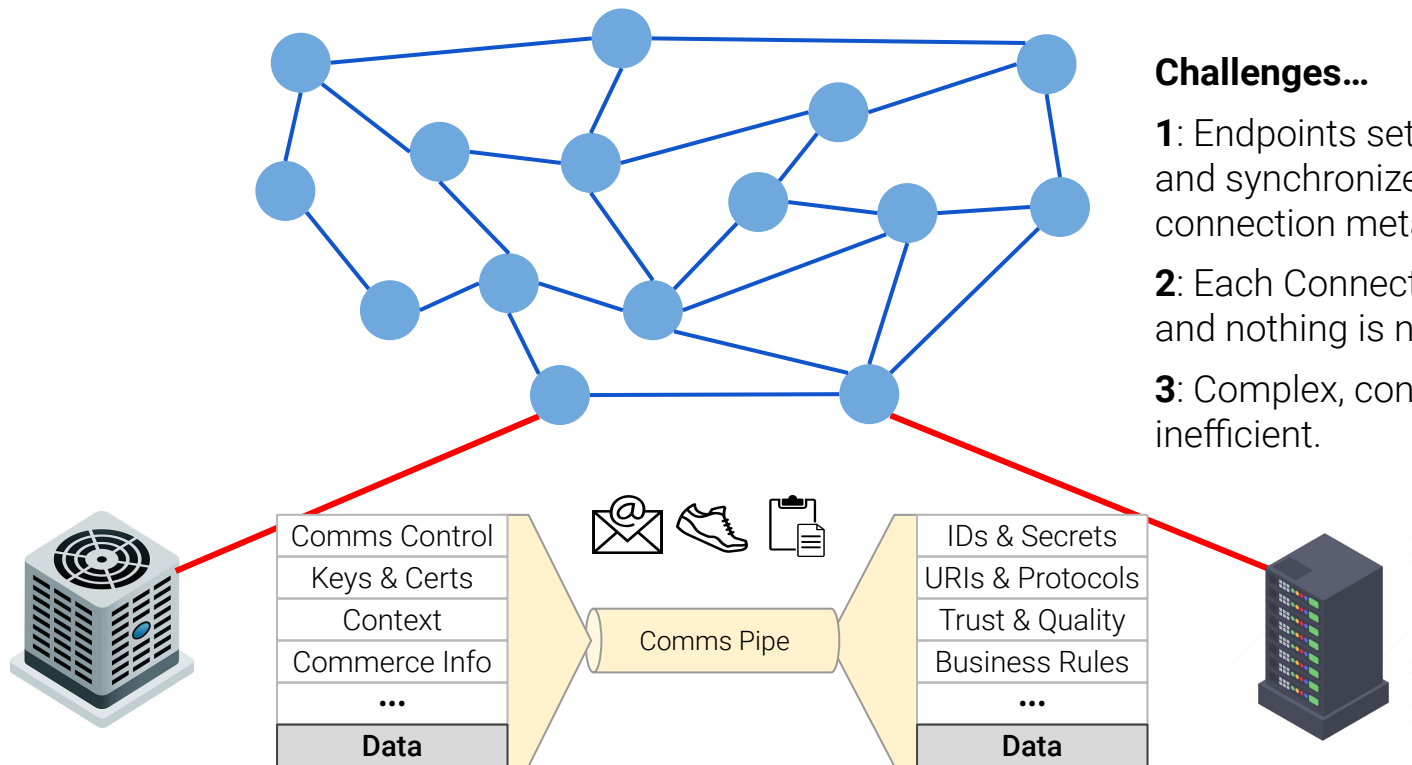
Relationships Are Complex

ID		
Data		
State		
Capabilities		
Trustworthiness		

X	Y
Information	
Context	
Compatibility	
Trust	

ID		
Data		
State		
Capabilities		
Trustworthiness		
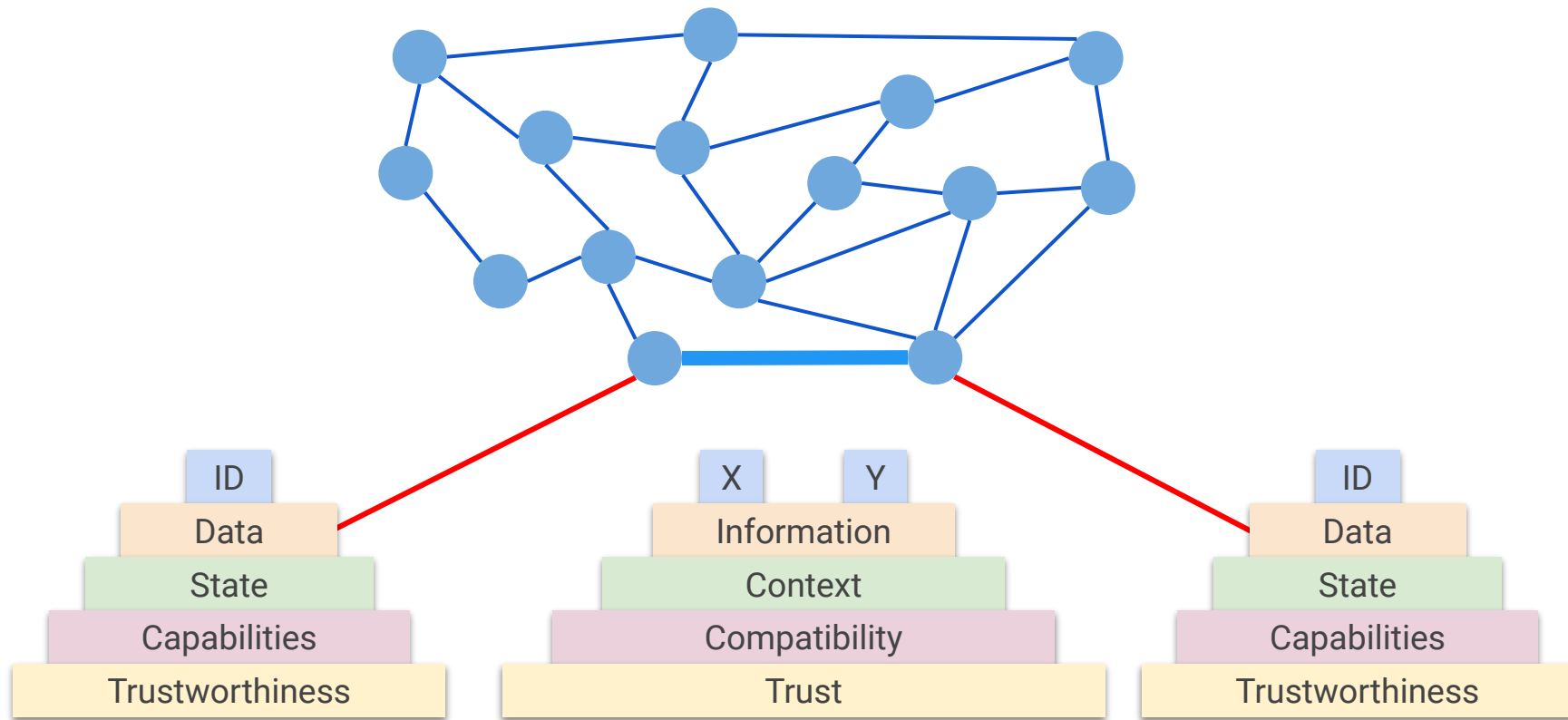
A Relationship Made up of Connections

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**!

Comms Co...					...ecrets
Keys ...	Client	Trust	Server		...otocols
Co...	Client	Comm Params	Server		...Quality
Comme...	Server	Business Rules	Client		...s Rules
...	Client	Other Metadata	Server		...
Metadata	Roles	Connection Profiles	Roles		Metadata

| Data | Comms Pipe (if required) | Data |

CNS/CP Workflow

cp:test.abc	
Client	Server
foo1	far1
foo2	far2

Startup Node Publishing

Foo (Client)	Pub/Sub	Orchestrator
Registers a Node ID, Context, CPs, Roles, and Properties foo1 foo2		foo1 foo2 Creates Virtual **Foo** & looks for matching Nodes in the Context given

A Match Starts Subscription

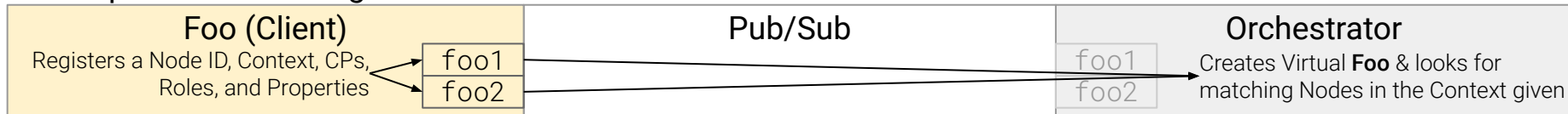| Foo (Client)	Pub/Sub	Orchestrator
Receives an Event when a Connection Instance is created far1 far2		far1 far2 When matched with Server **Far**, sends Connection ID & Properties
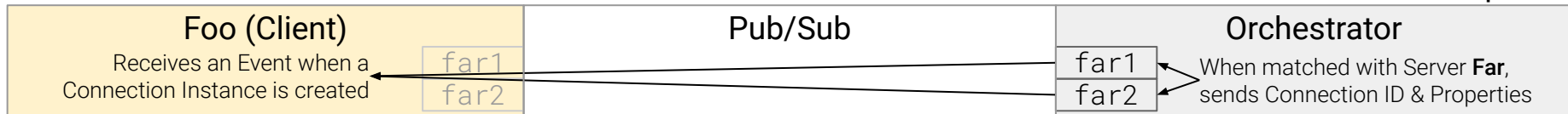
Ongoing Bi-Directional Subscription

Foo (Client)	Orchestrator	Far (Server)
Sets **Foo** Properties foo1 foo2	cp:test.abc	foo1 foo2 Receives Events when **Foo** Properties change
Receives Events when **Far** Properties change far1 far2		far1 far2 Sets **Far** Properties

JSON Example

cp:test.abc	
Client	Server
foo1	far1
foo2	far2
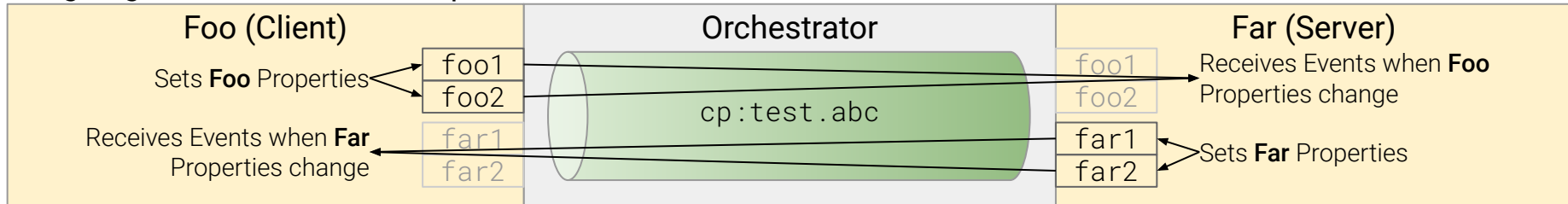
Client Node Publishing {
    "nodeID" : "did:example:123456789abcdefghijk",
    "context" : "context-string",
    "profiles" : {
        "cp" : "test.abc",
        "role" : "client",
        "properties": {
            "foo1" : "value for f
            "foo2" : "value for f
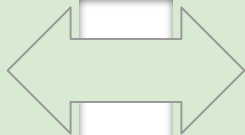   }
    }
}

Server Match Found {
    "nodeID" : "did:example:123456789abcdefghijk",
    "matchID" : "did:example:abcdefghijk123456789",
    "connectionID" : "NvdmaNDNLPFR9d9dB",
    "cp" : "test.abc",
    "properties": {
        "far1" : "value for far1",
        "far2" : "value for far2",

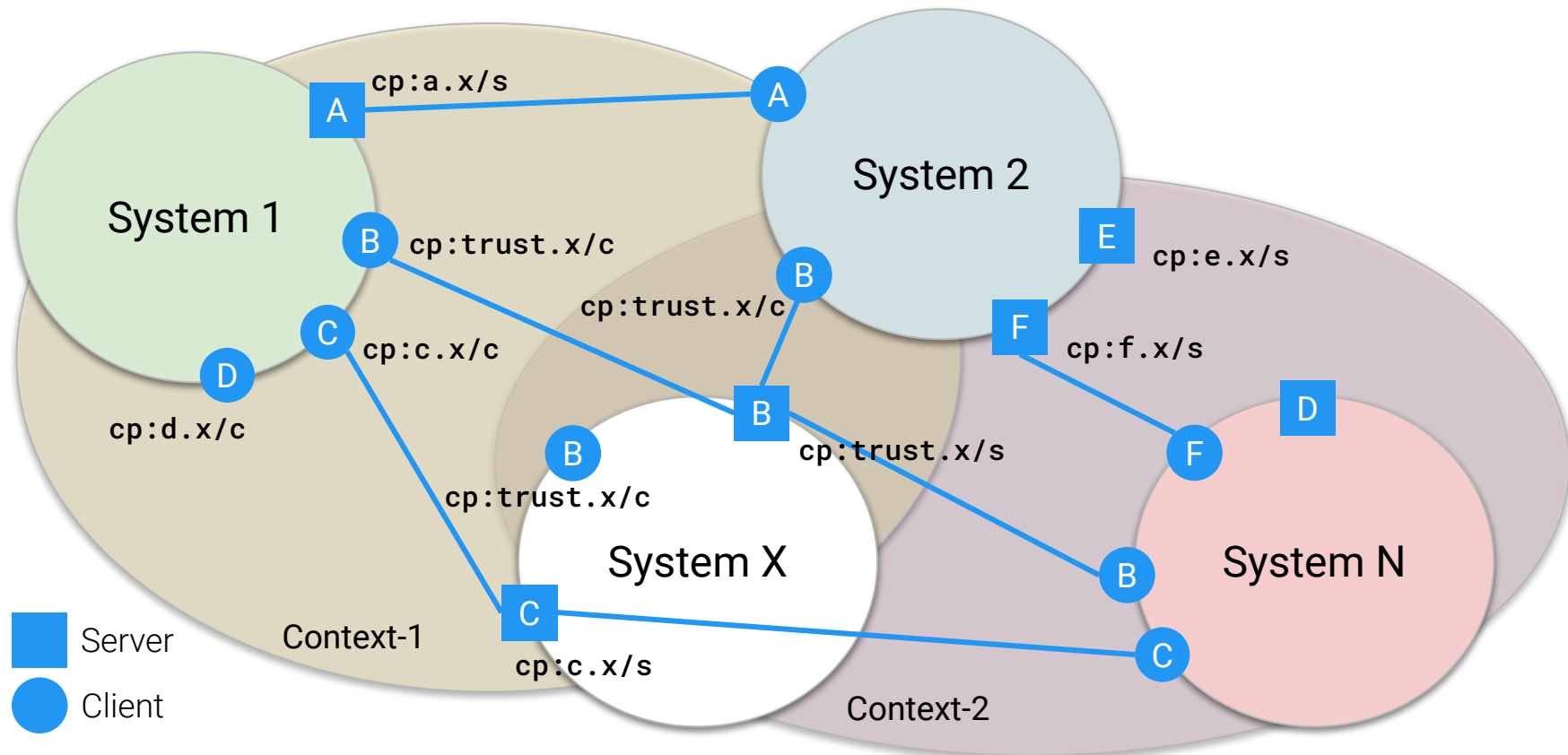**Client>Server Subscription** {
    "connectionID" : "NvdmaNDNLPFR9d9dB"
    "properties": {
        "foo1" : "value for foo1",
        "foo2" : "value for foo2",
    }
}

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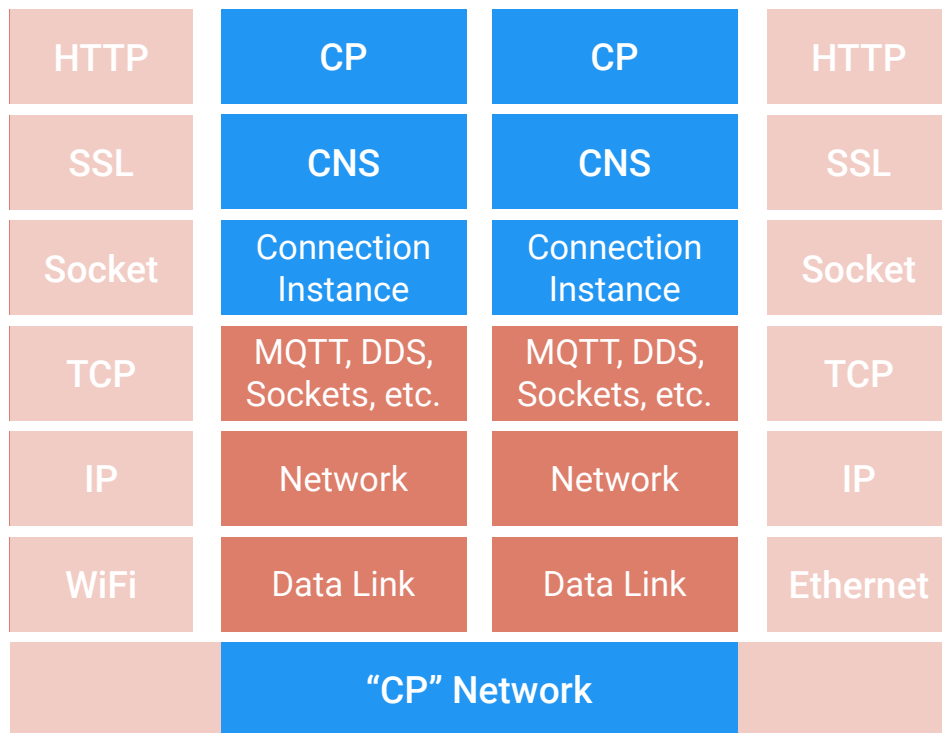   "far2" : "value for far2",
    }
}

Systems, Context, and Connections

CNS/CP and the OSI Stack - For Feedback

	OSI Layer	Description					

			HTTP	CP	CP	HTTP

OSI Stack (left reference):

Layer	Details
7. Application	• End User layer • HTTP, FTP, IRC, SSH, DNS
6. Presentation	• Syntax layer • SSL, SSH, IMAP, FTP, MPEG, JPEG
5. Session	• Synch & send to port • API's, Sockets, WinSock
4. Transport	• End-to-end connections • TCP, UDP
3. Network	• Packets • IP, ICMP, IPSec, IGMP
2. Data Link	• Frames • Ethernet, PPP, Switch, Bridge
1. Physical	• Physical structure • Coax, Fiber, Wireless, Hubs, Repeaters

CNS/CP Mapping:

Col 1	Col 2	Col 3	Col 4
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		

15

CNS/CP

For more
visit cnscp.io or contact info@padi.io