SILICON VALLEY CODE CAMP

DDS Connector: The Industrial Internet of Things Platform in Node.js

Gianpiero Napoli
Senior Software Engineer - RTI
gianpiero@rti.com / @magopieri

SILICON VALLEY CODE CAMP

DDS Connector: The Industrial Internet of Things Platform, now in Node.js

Gianpiero Napoli
Senior Software Engineer - RTI
gianpiero@rti.com / @magopieri

Agenda

DDS Connector: The Industrial Internet of Things Platform, now in Node.js

- Who am I
- What is **DDS**
- **What is the Connector**
  - How does it work in **node.js**
  - **Hands On ← ← ←**
- What is the **IIoT**
  - Protocols for the IIoT

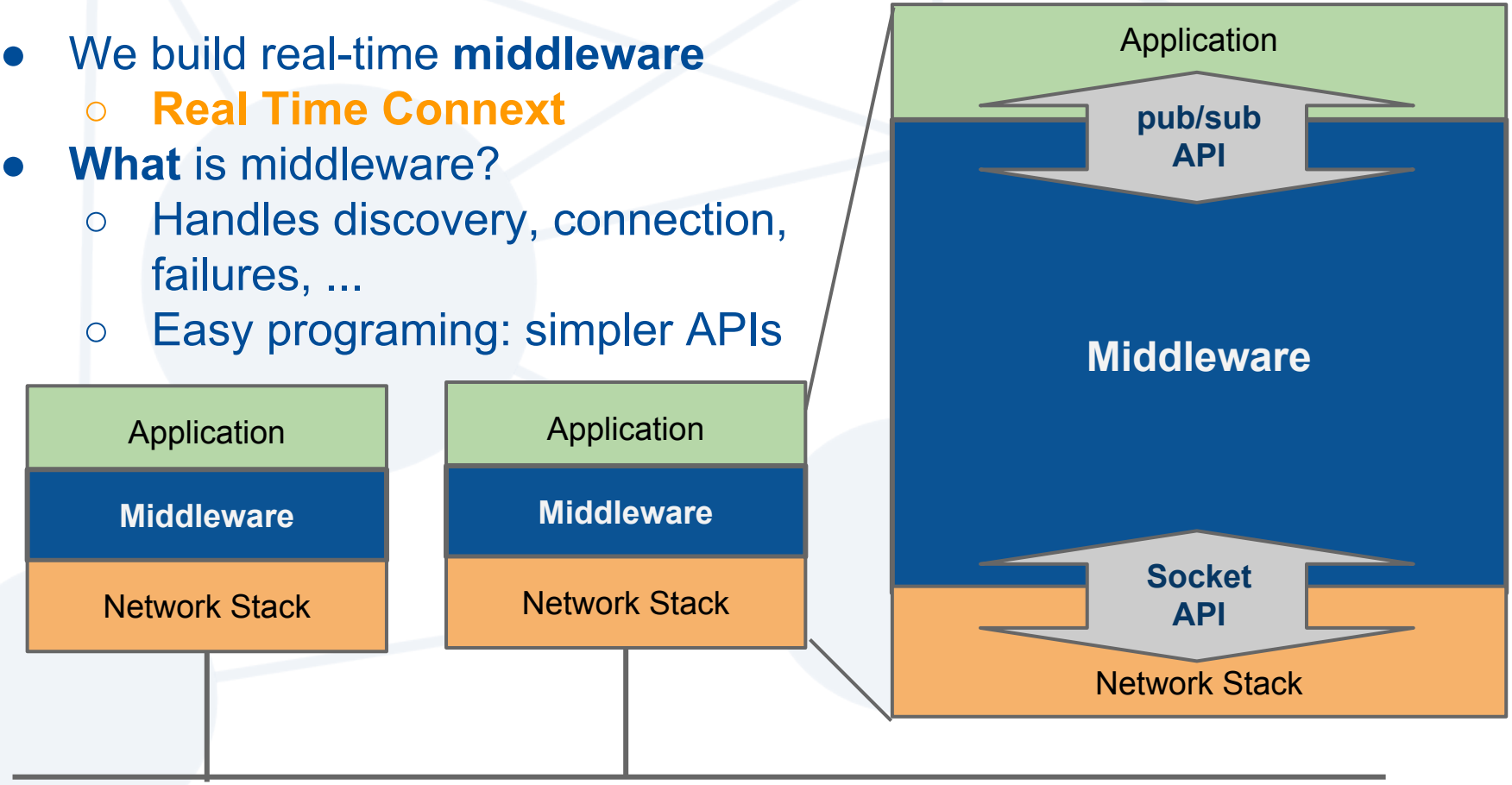
Who am I?

and what do I do

Who?

What we do

- We build real-time **middleware**
  - **Real Time Connext**
- **What** is middleware?
  - Handles discovery, connection, failures, ...
  - Easy programing: simpler APIs



What is DDS

https://en.wikipedia.org/wiki/Dentist#/media/File:US_Navy_030124-N-1328C-510_Navy_dentist_treats_patients_aboard_ship.jpg

Data Distribution Service

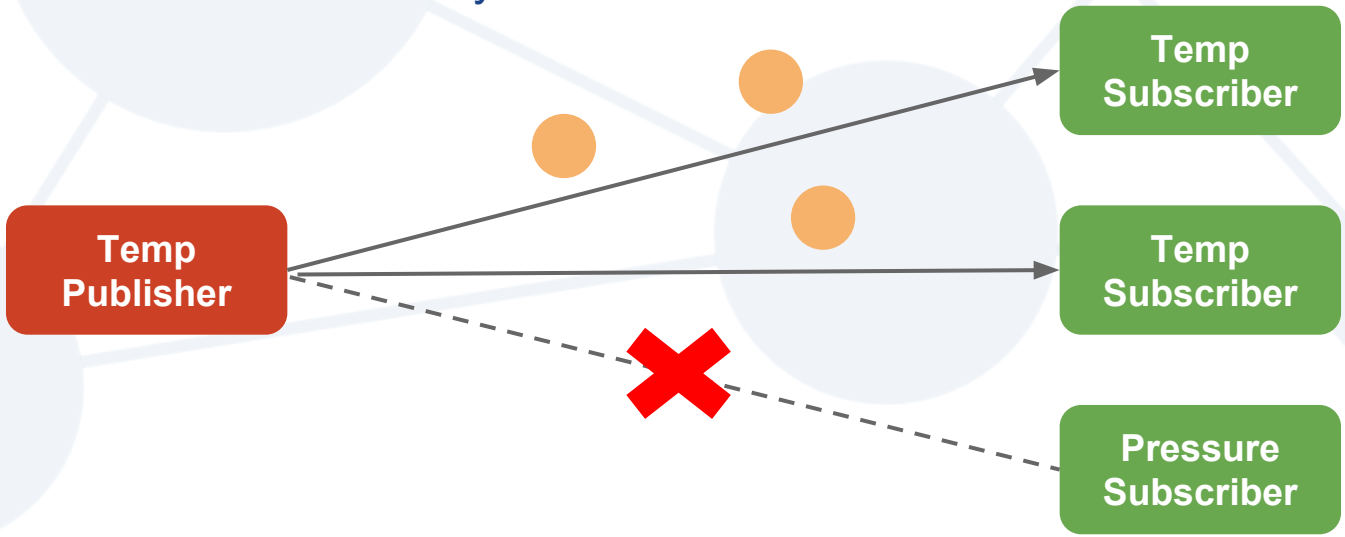
a **real time** communication technology
standard for the **Internet of Things**

The DDS Family

- Object Management Group Standards
- Data Distribution Service (DDS)
  - API
  - QoS
- Real-Time Publish Subscribe (RTPS)
  - Data encoding
  - Interaction Protocol
  - On the Wire Format
- Extensions:
  - XTypes
  - Security

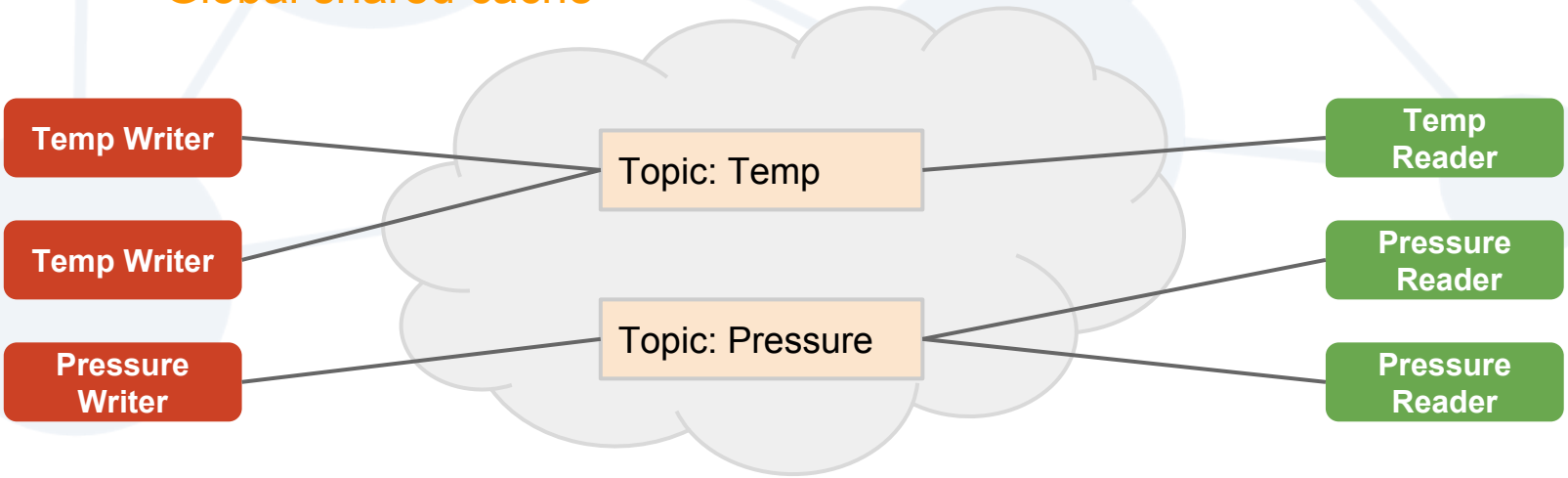
Publish/Subscribe

- Paradigm shift:
  - From "**give me your information**" to "**send me your data have when you have more**"
- Applications specify what can provide and what are they interested in
  - Middleware handles sending, reception and conversion
  - e.g. "I offer temperature data", "I'm interested in pressure data"
- Applications are matched by interests:



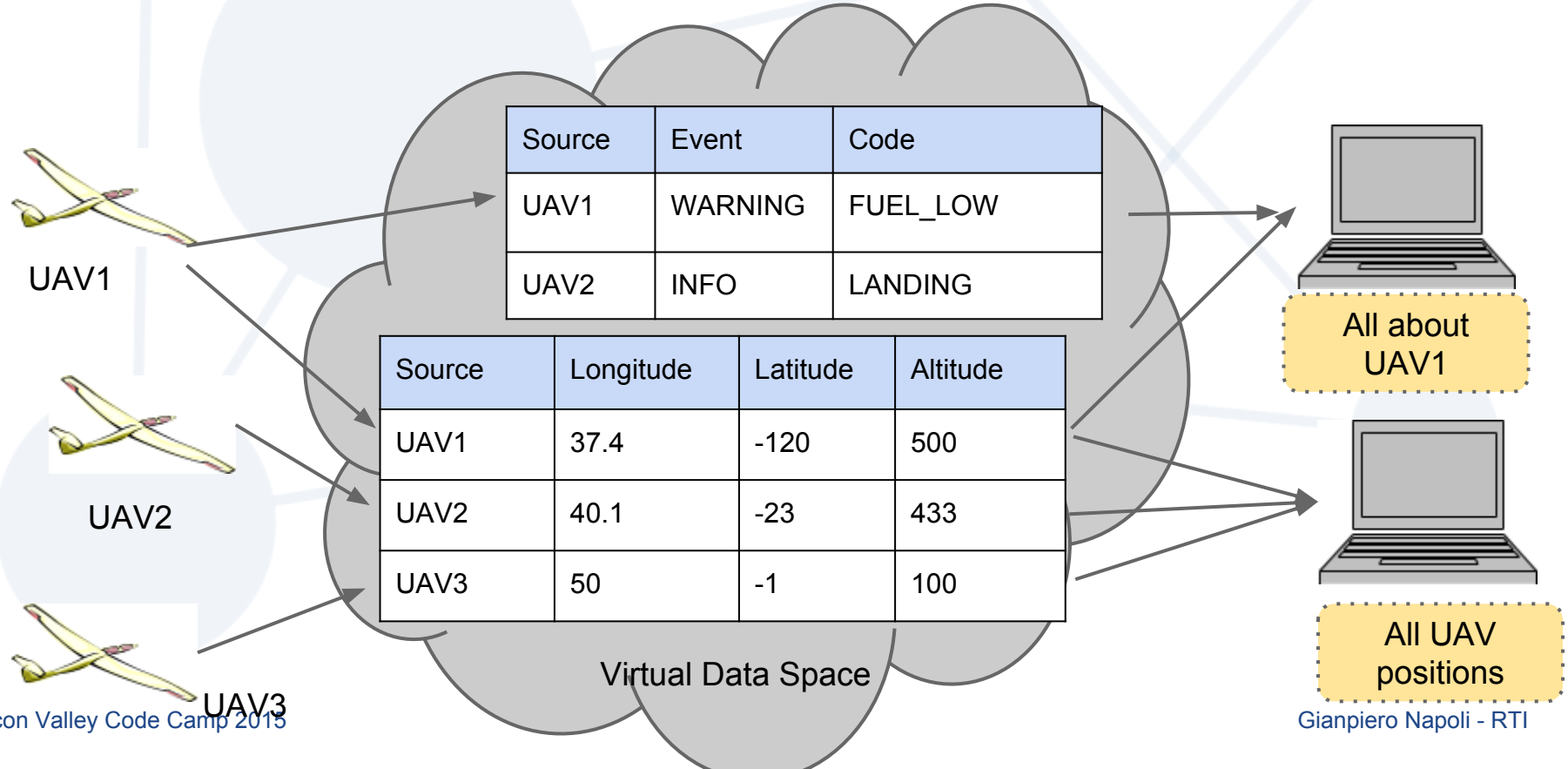
Data Centric Model

- Data drive the communication
  - Data type and content define the interactions
  - e.g. Temperature data
- Topic is the exchange unit
  - Name + **Type**
  - Samples are univocally identified by keys (like in DB)
  - QoS per publication: matched vs. offered
- Global Data Space
  - Applications publish topics to a global data space
  - Global shared cache

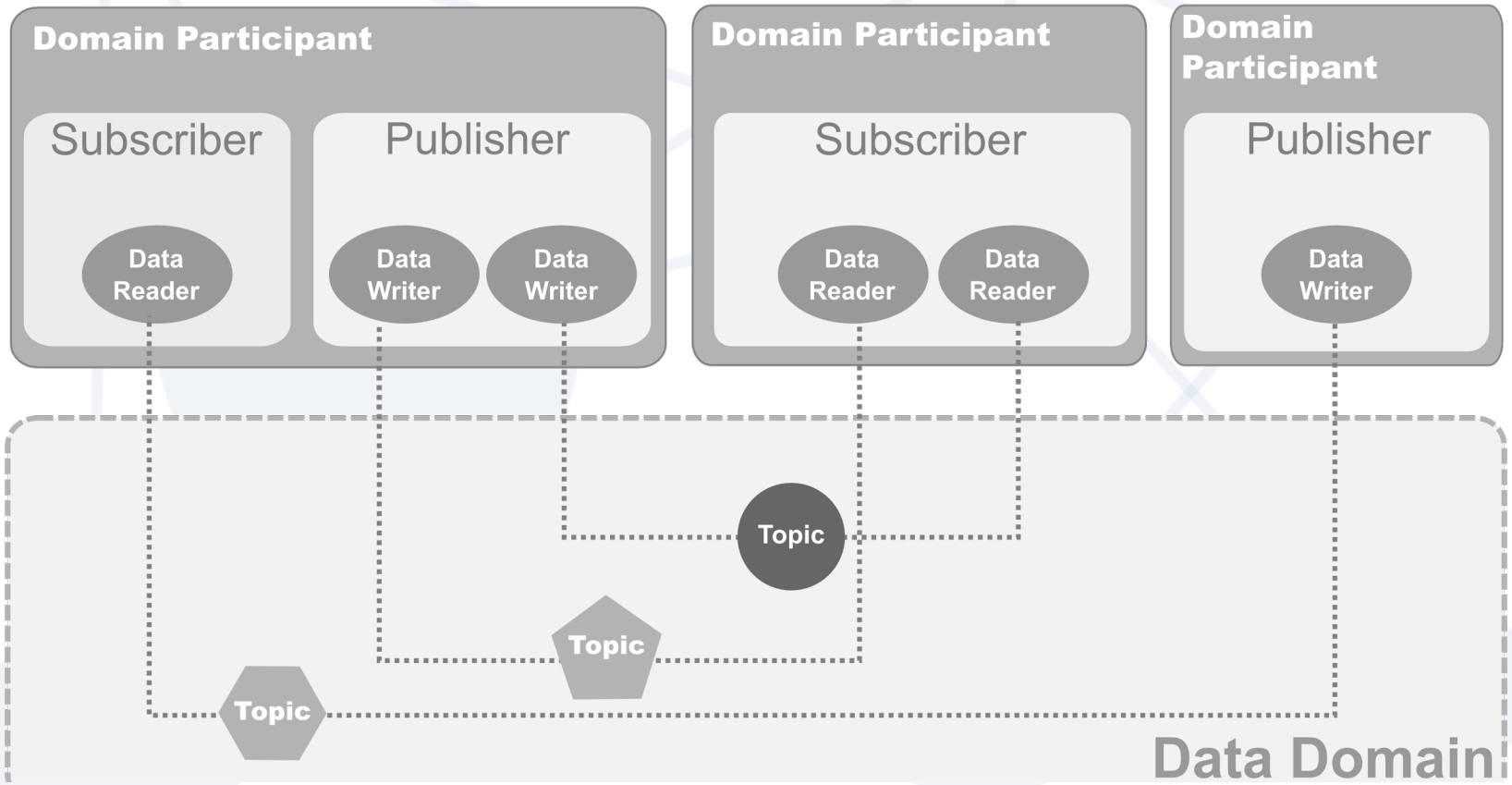


Data Centric Model

- Decentralized
- Acts as a distributed database/cache
- No servers involved



DDS Architecture



Quality of Service

choose your ingredients and you are ready to go..

https://commons.wikimedia.org/wiki/File:Salad_bar-02.jpg

Quality of Service (QoS)

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	OWNERSHIP
RELIABILITY	OWNERSHIP STRENGTH
TIME BASED FILTER	LIVELINESS
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User

Presentation

Redundancy

Transport

Example: Reliable Alarm/Events

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	OWNERSHIP
RELIABILITY	OWNERSHIP STRENGTH
TIME BASED FILTER	**LIVELINESS**
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User

Presentation

Redundancy

Transport

Example: Data Redundancy

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	**OWNERSHIP**
RELIABILITY	**OWNERSHIP STRENGTH**
TIME BASED FILTER	**LIVELINESS**
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User

Presentation

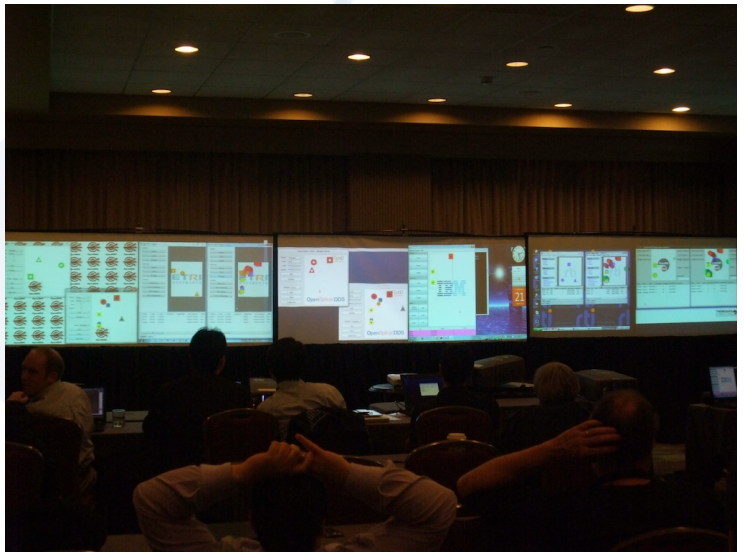
Redundancy

Transport

The DDS Standard

- OMG!
- Interoperability
- Standards family
  - DDS 1.2 (API)
  - RTPS 2.0 (Wire prototocol)
  - X-Types (Extensibility)
  - WEB-DDS
  - UML Profile
  - PSM (C++, Java)
  - Security (WIP)
  - RPC (WIP)

OBJECT MANAGEMENT GROUP



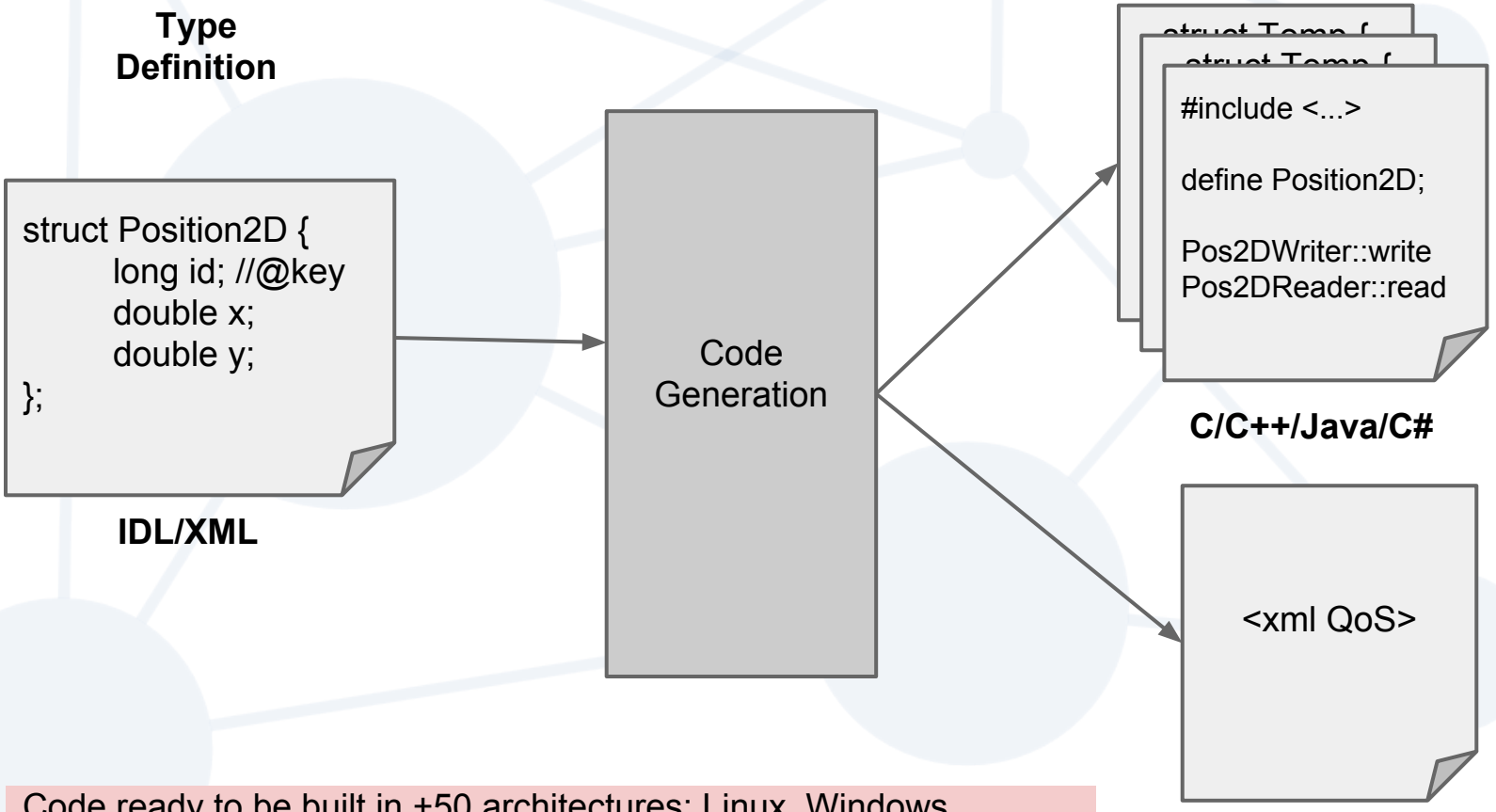
http://portals.omg.org/dds/

What is the DDS Connector?

What is Connector?

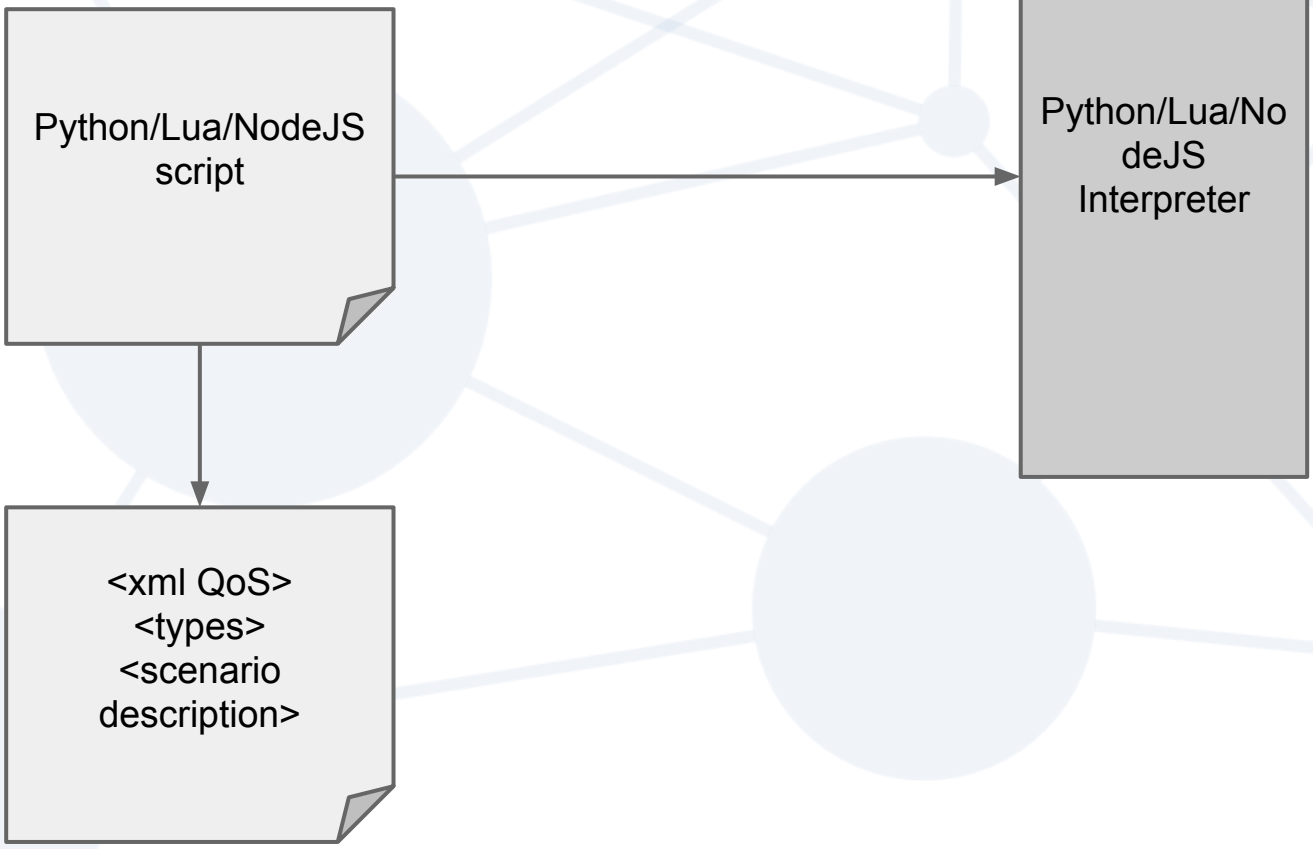
- Simple API for Data-Centric Publish/Subscribe
  - Built on top of DDS
  - Very few methods
  - Experimental
- Scripting language bindings
  - **node.js**, python, nodered
- Prototyping and Testing
  - Entities and QoS configured by XML

Classical DDS Workflow

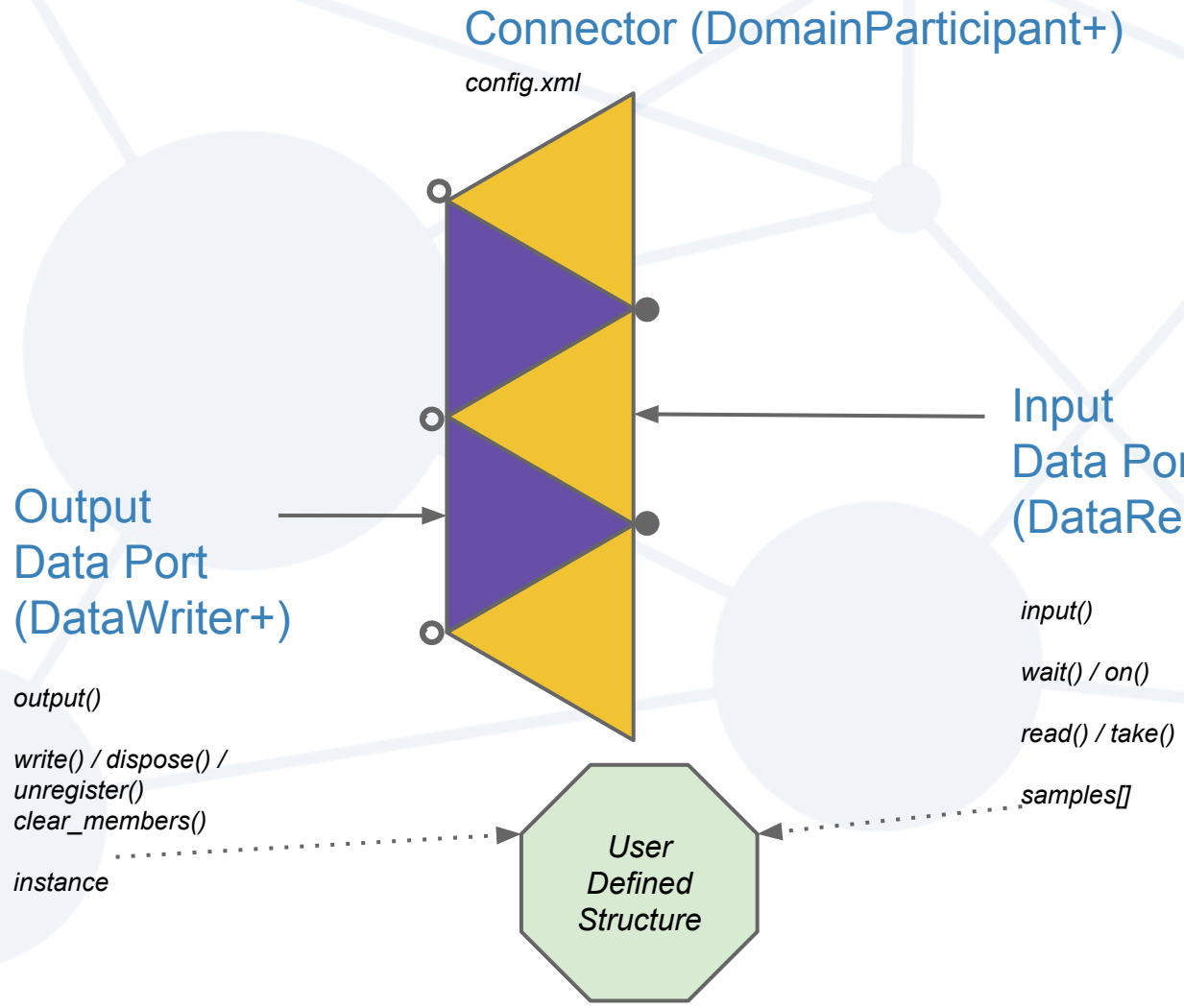


Code ready to be built in +50 architectures: Linux, Windows, VxWorks, Integrity….
This process can be simplified even more: **RTI Prototyper/Connector**

Connector Workflow



DDS Connector Input & Output ports



Anatomy of a Publisher in Connector

```
var rti    = require('rticonnextdds-connector');
1.   var connector = new rti.Connector("MyParticipantLibrary::Zero","ShapeExample.xml");
2.   var output = connector.getOutput("MyPublisher::MySquareWriter");
3.   output.instance.setNumber("shapesize",30);
4.   output.instance.setString("color", "BLUE");
5.   output.write();
```

1. Create a connector
2. Get the datawriter
3. Set the instance values
4. Set the instance values
5. Write the value

Anatomy of a Subscriber in Connector

```
var rti    = require('rticonnextdds-connector');
1.   var connector = new rti.Connector("MyParticipantLibrary::Zero",”ShapeExample.xml");
2.   var input = connector.getInput("MySubscriber::MySquareReader");
3.   connector.on('on_data_available',function() {
4.   input.take();
     for (i=1; i <= input.samples.getLength(); i++) {
         if (input.infos.isValid(i)) {
             console.log(JSON.stringify(input.samples.getJSON(i)));
         }
     }
});
```

1. Create a connector
2. Get the datareader
3. Register a function to be called when data are available
4. Read them
    a. For each sample
        i. check if valid
        ii. print :)

About the examples

- Temperature sensor scenario
  - Sensors (publishers)
    - Publish temperature data every second
  - Console (suscribers)
    - Shows a table with data being published
- Configured by XML
  - Tutorial.xml
    - Define the DDS entities and their QoS settings

```
struct Sensor {
    string id; //@key
    long value;
    long timestamp;
};
```

Anatomy of the XML

1. **QoS Library**
   a. **QoS Settings**
2. Types
   a. Type Definition
3. Domain Library
   a. Domains and Topics
4. Participant Library
   a. DDS Entities hierarchy

```
<!-- Qos Library -->
<qos_library name="QosLibrary">
  <qos_profile name="DefaultProfile"
is_default_qos="true">
    <participant_qos>
      <transport_builtin>
        <!-- <mask>UDPV4 | SHMEM</mask>-->
        <mask> SHMEM</mask>
      </transport_builtin>
    </participant_qos>

    <datareader_qos>
   <!-- Modify reader values here -->
    </datareader_qos>

  </qos_profile>
</qos_library>
```

Anatomy of the XML

1. QoS Library
   a. QoS Settings
2. **Types**
   a. **Type Definition**
3. Domain Library
   a. Domains and Topics
4. Participant Library
   a. DDS Entities hierarchy

```
<types>
  <struct name="Sensor"     extensibility="
extensible">
   <member name="id" stringMaxLength="128" id="0"
type="string" key="true"/>
   <member name="value" id="1" type="long"/>
   <member name="timestamp" id="2" type="long"/>
 </struct>
</types>
```

Anatomy of the XML

1. QoS Library
   a. QoS Settings
2. Types
   a. Type Definition
3. **Domain Library**
   a. **Domains and Topics**
4. Participant Library
   a. DDS Entities hierarchy

```
<!-- Domain Library →
<domain_library name="MyDomainLibrary">
  <domain name="MyDomain" domain_id="0">
   <register_type name="Sensor" kind="dynamicData"
type_ref="Sensor"/>
   <topic name="Temperature" register_type_ref="
Sensor"/>
 </domain>
</domain_library>
```

Anatomy of the XML

1. QoS Library
   a. QoS Settings
2. Types
   a. Type Definition
3. Domain Library
   a. Domains and Topics
4. **Participant Library**
   a. **DDS Entities hierarchy**

```
<!-- Participant library -->
<participant_library name="MyParticipantLibrary">
 <domain_participant name="Console" domain_ref="
MyDomainLibrary::MyDomain">
   <subscriber name="TempSubscriber">
     <data_reader name="TempReader" topic_ref="
Temperature"/>
   </subscriber>
  </domain_participant>
  <domain_participant name="Sensor" domain_ref="
MyDomainLibrary::MyDomain">
  <publisher name="TempPublisher">
    <data_writer name="TempWriter" topic_ref="
Temperature"/>
    </publisher>
  </domain_participant>

</participant_library>
```

Hands On

https://github.com/gianpiero/dds-firststeps

Example: Basic pub/sub

- Objetive
  - In this example we show how to publish data
  - Learn differences between read/take
  - Learn how to change QoS settings in Connector
- Documentation:
  - https://community.rti.com/rti-doc/510/ndds.5.1.0/doc/pdf/RTI_CoreLibrariesAndUtilities_QoS_Reference_Guide.pdf

Example: Filtering

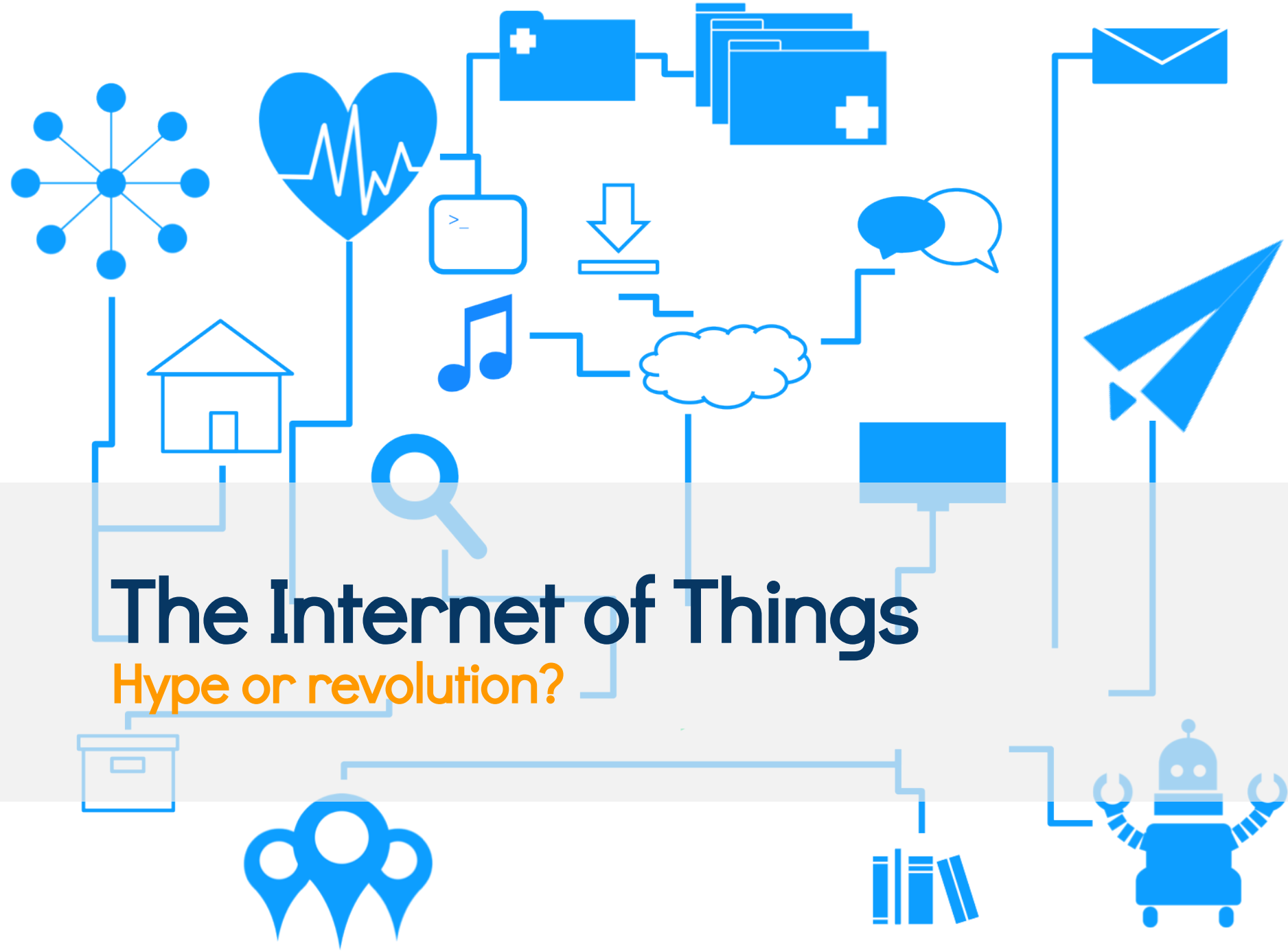
- Objetive
  - Learn how to filter data per subscriber
- Description
  - The console application will only receive the data matching a certain criteria
- Two types of Filtering:
  - Content
    - Use a filter with SQL Expression
  - Time Based
    - time_based_filter QoS

Example: Durability

- Objetive
  - Learn how to provide recent history to late joiners
- Description
  - A console application will receive the recent history published before it was started
- QoS:
  - Durability QoS

Example: Integration with node ecosystem

- Objetive
  - Show how to integrate DDS Connector with other available technology
- Description
  - A nodejs app will receive DDS Data and push them into a websocket



The Internet of Things

Hype or revolution?

Multiple Definitions

Cisco "Internet of Everything"

…the latest wave of the Internet -- connecting physical objects…to provide better safety, comfort, and efficiency

IBM "Internet of Things"

…a completely new world-wide web, one comprised of the messages that digitally empowered devices would send to one another. It is the same Internet, but not the same Web.

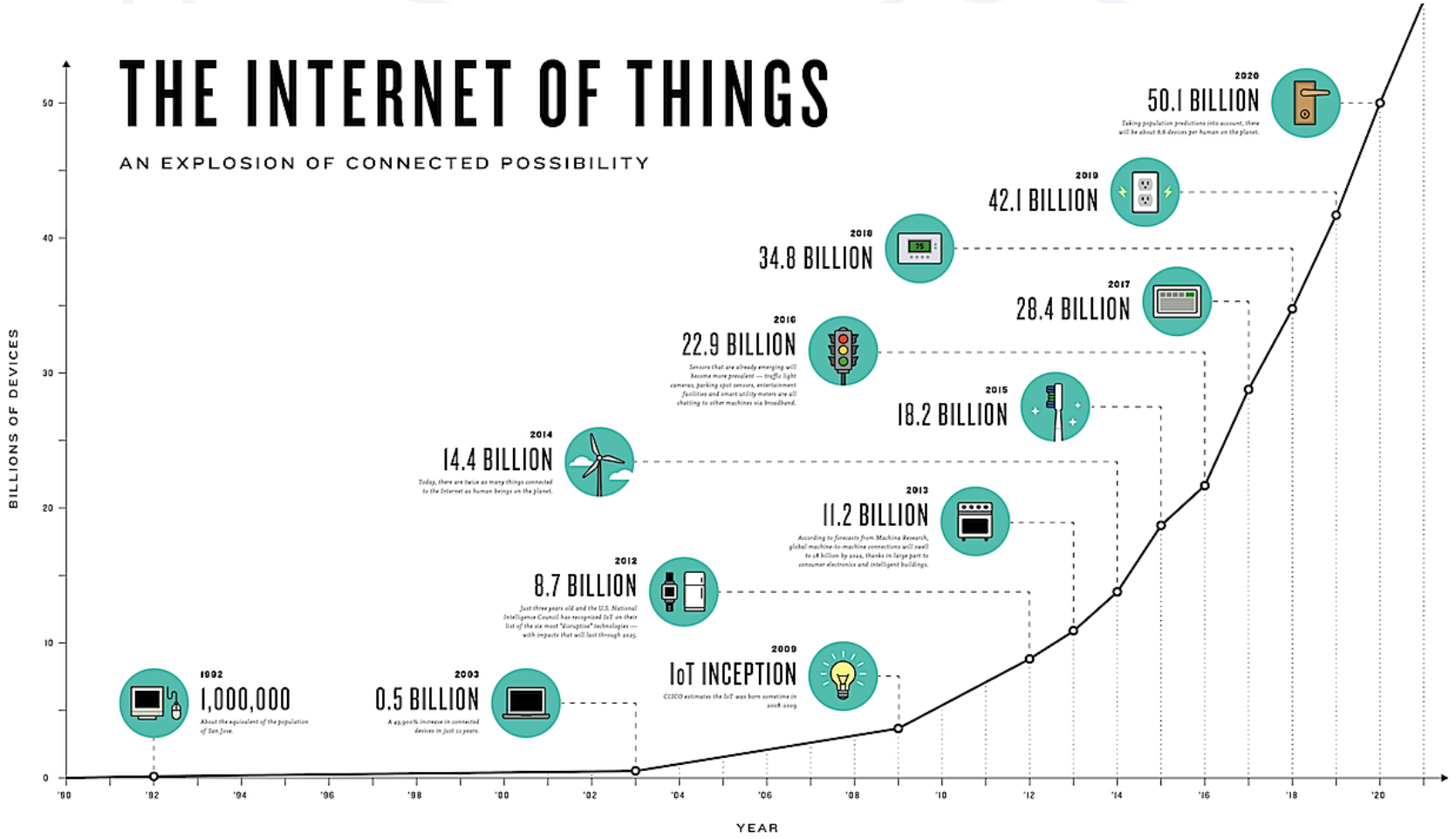
GE "Industrial Internet"

…convergence of machine and intelligent data…to create brilliant machines

RTI "Your Systems. Working as One."

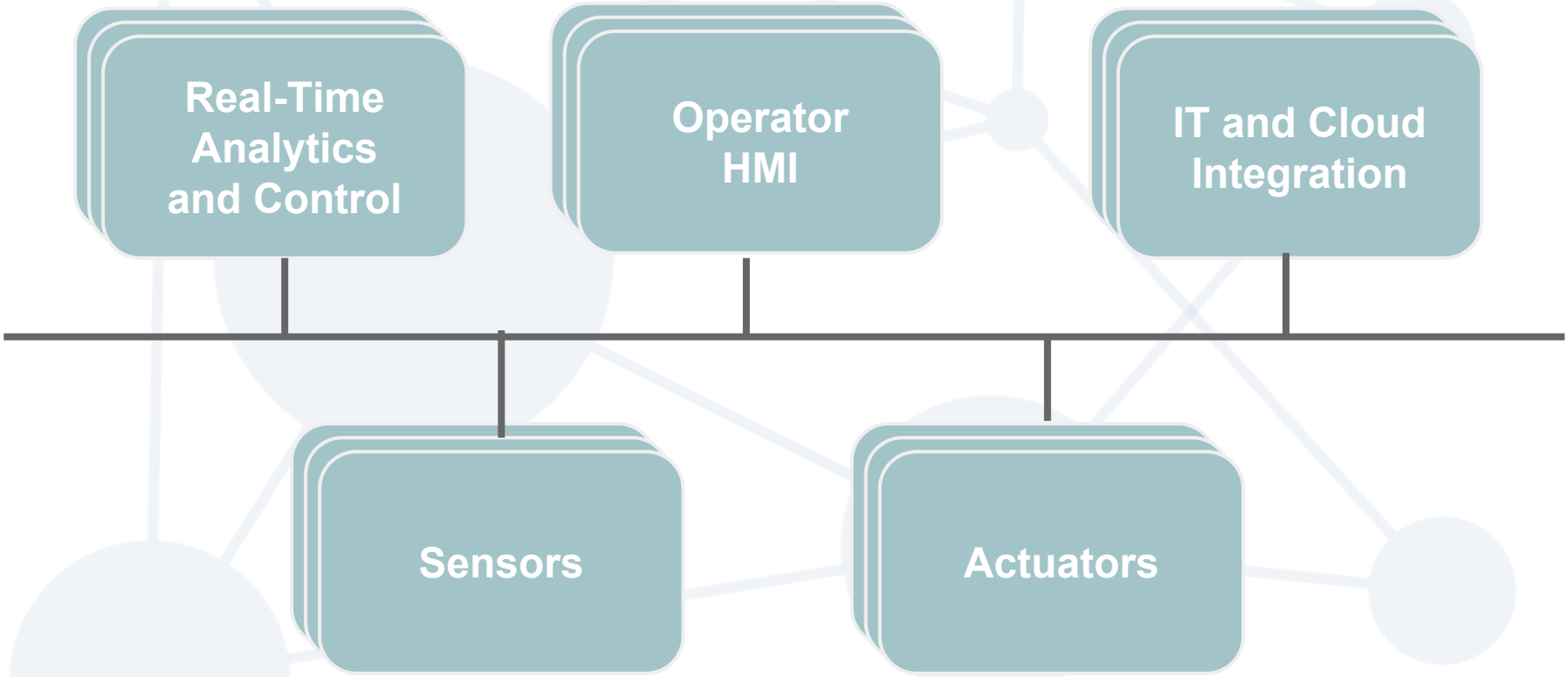
…an entirely new utility. As profound as the cell network, GPS, or the Internet itself. The Internet of Things and the Intelligent Systems it enables will fundamentally change our world.

The Next Internet



Internet of Things?

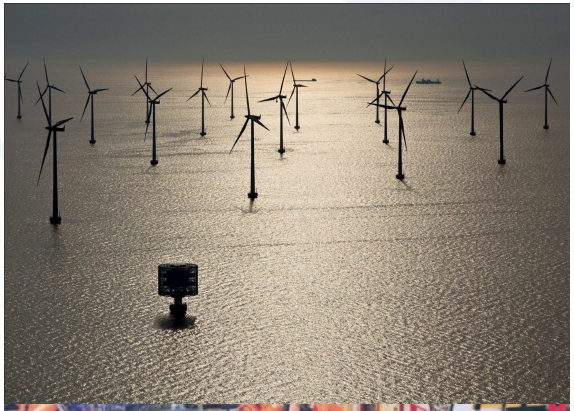
The future: Smart Systems

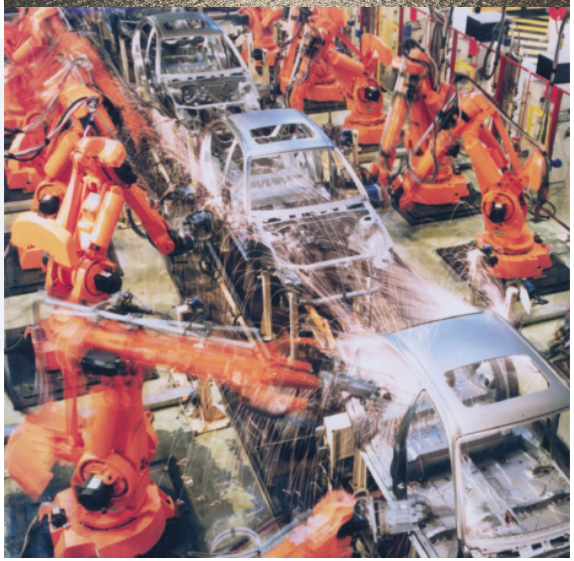


DATA DISTRIBUTION

IIoT Scenarios

- Manufacturing
- Transportation
- Energy
- Agriculture
- Healthcare
- Public infrastructure





IIoT Challenges

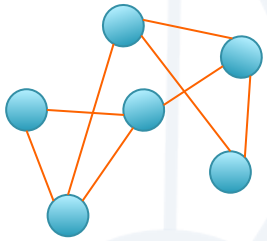
- Scalability
- Security
- Interoperability
- Complexity
- Fault Tolerance
- Mobility/Ubiquity
- Automatic Discovery

Protocols for the IIOT

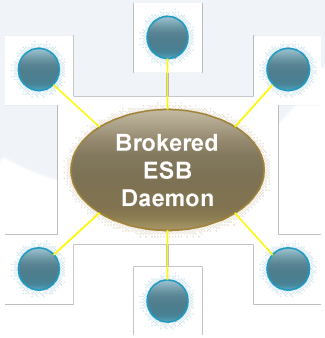
DDS, AMQP, MQTT, ZEROMQ, CoAP...

Types of Flow

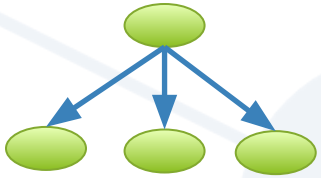
Point-to-Point



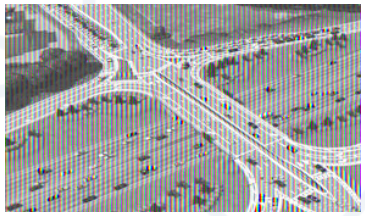
Client/Server

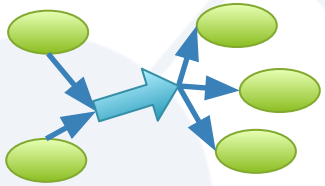


Publish/Subscribe

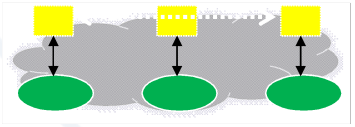


Queuing





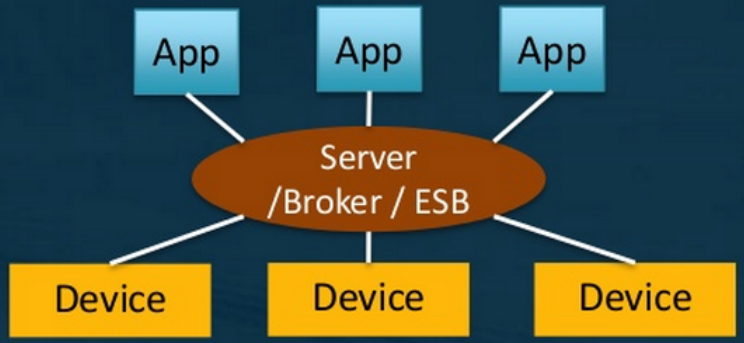
Data-Centric



Data-Centric
Publish/Subscribe (DCPS)
DataBus

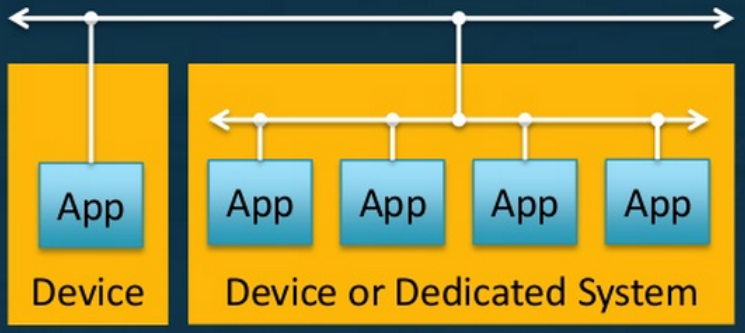
Types of Architectures

AMQP, MQTT, CoAP



- Centralized Hub and spoke
  - Comms route through server
  - Web server, broker or ESB
- Server queues messages with simple QoS (e.g. TCP)
- Simplified discovery
- Server challenges: failover, latency under load, error recovery

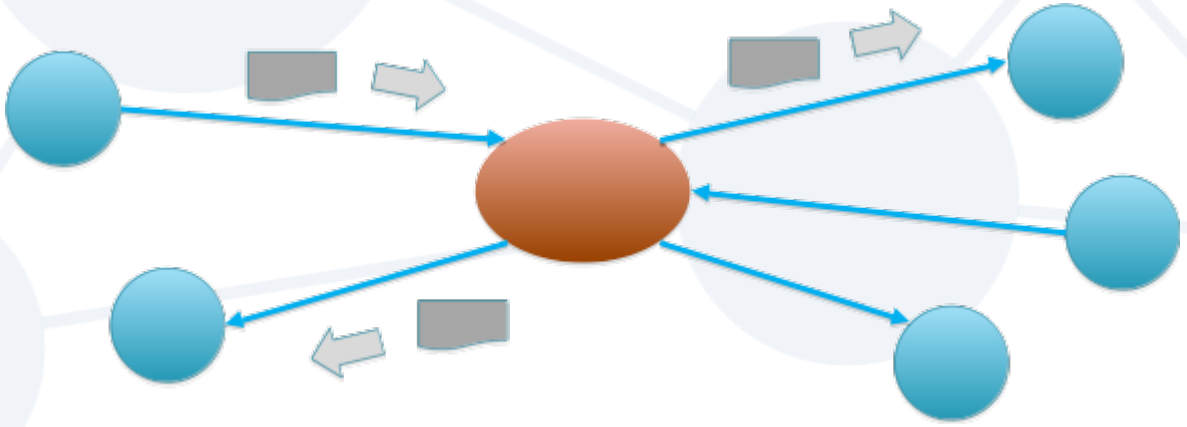
DDS



- Decentralized Peer-to-peer
  - Fully embeddable
  - No external sftw (e.g., broker)
- Bus shares data with complex QoS
- Discovery challenge
- No single point of failure, bottleneck, or service interruptions

Message Centricity

- Traditional middleware exchanges messages
- Infrastructure is unaware of the content (opaque)
- Developers write applications that send messages between participants



Data Centricity

- Data-centric middleware maintains state
- Infrastructure manages the content
- Developers write applications that read and update a virtual global data space

Source (Key)	Wind	Power	Phase
WPT1	37.4	122.0	-12.20
WPT2	10.7	74.0	-12.23
WPTN	50.2	150.1	-11.98

Persistence Service

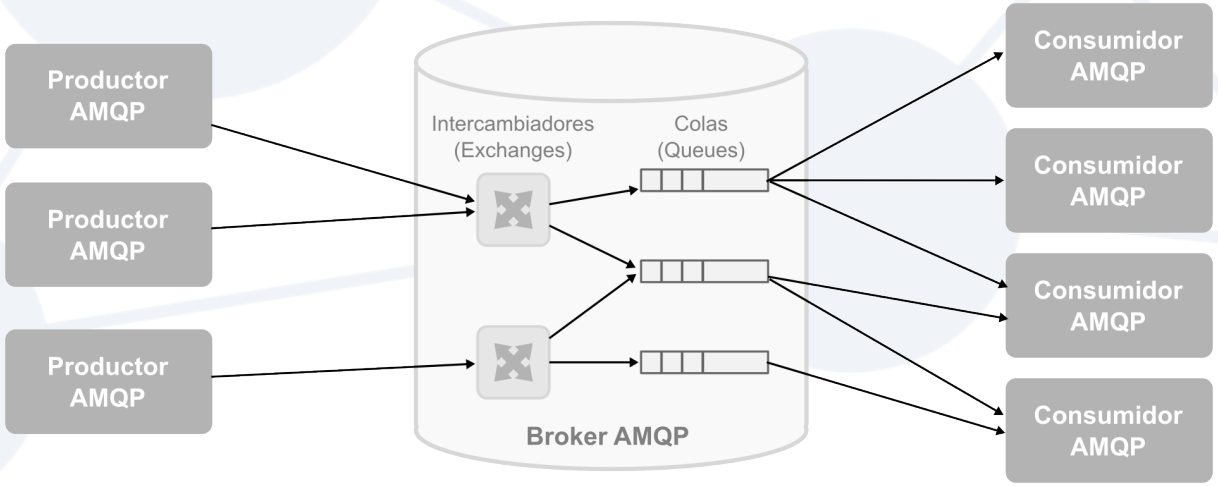
Recording Service

CoAP

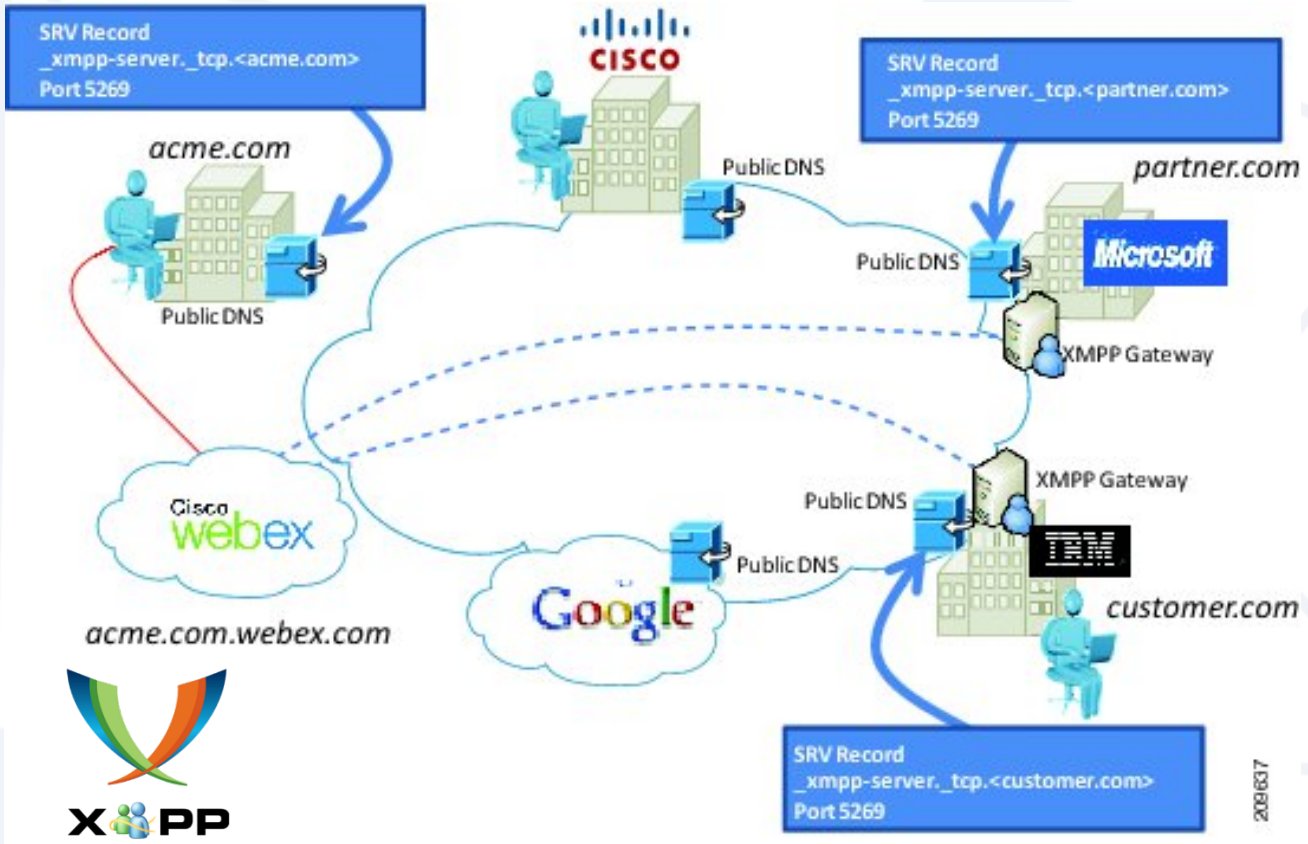
- COnstrained Application Protocol
- Web of Things
  - REST model for small devices
- Pull model
- RFC7252
- http://coap.technology/

AMQP

- Advanced Message Queuing Protocol
- Wire-level protocol
- One to One and One To Many
- Broker based
- message centric
- Implementations:
  - RabbitMQ, ActiveMQ

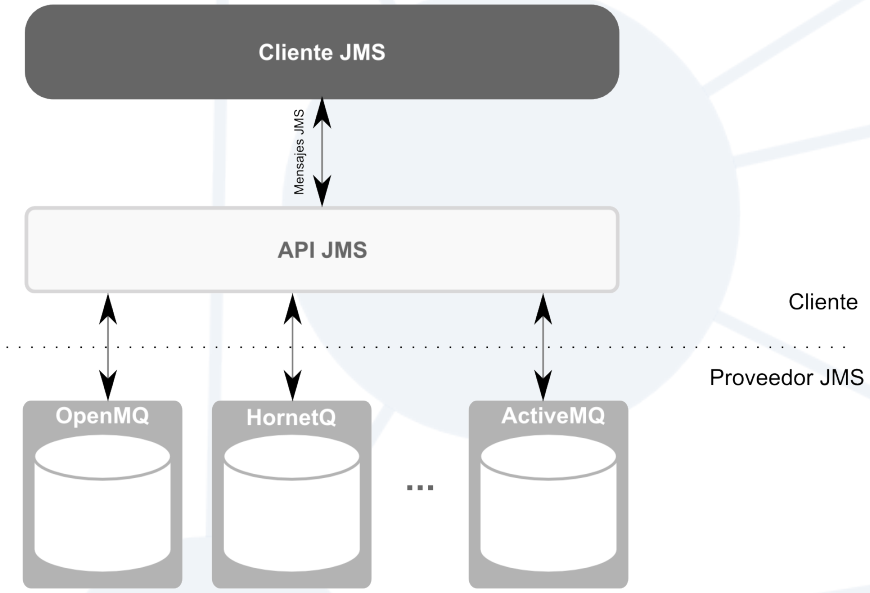


XMPP: Access Device Data



Extensible Messaging and Presence Protocol (XMPP)

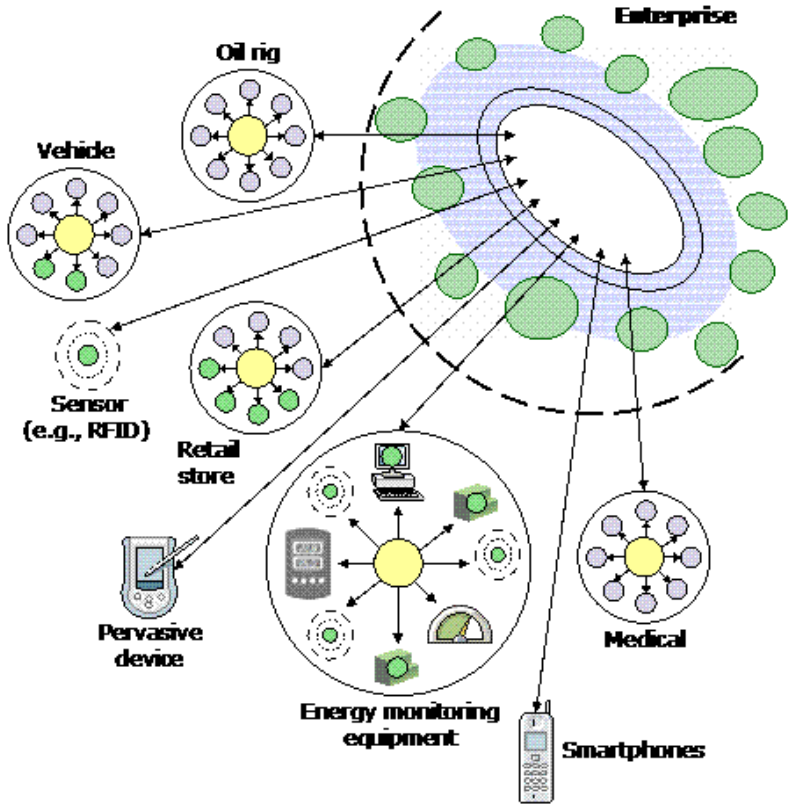
JMS



- Java Messaging Protocol
- API
- Broker based
- Message centric
- Implementations:
  - Websphere, ActiveMQ,...

MQTT: Collect Device Data

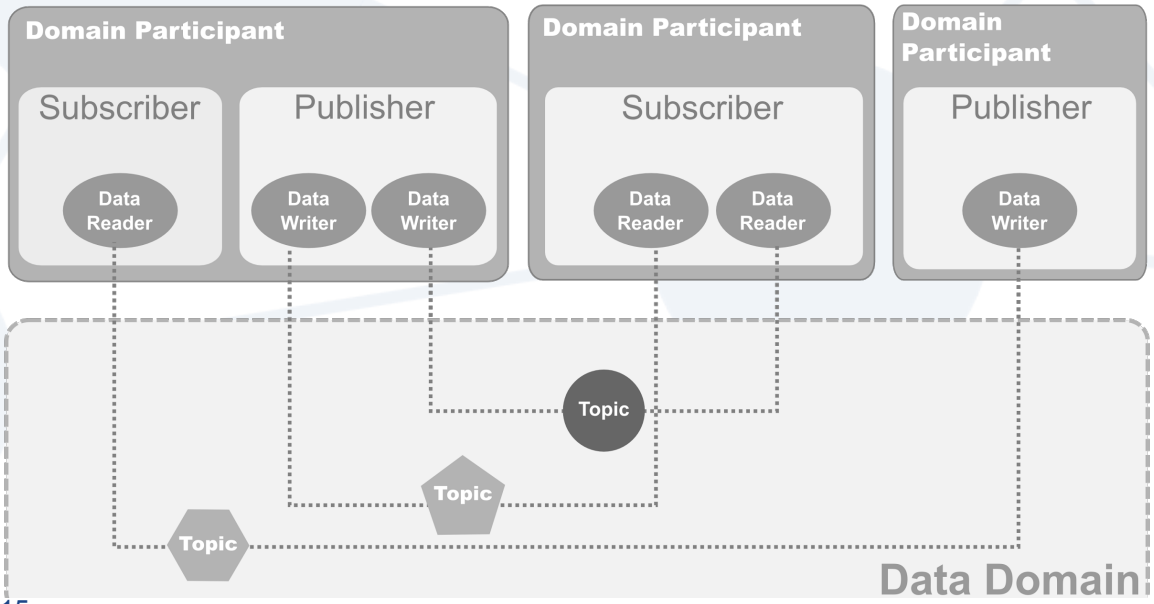
- Brokered
- Lightweight
- Wire protocol
- Implementations:
  - paho, mosquitto, ...



Message Queuing Telemetry Transport (MQTT)

Data Distribution Service

- Data Centric Approach
- Fully distributed Architecture
- Advanced QoS
- Stantard
- Implementations:
  - RTI Connext, Vortex, OpenDDS, CoreDX



Pub/Sub Protocol Comparison

	AMQP	JMS	MQTT	DDS
Architecture	Broker	Broker	Broker	Decentralized
Type	Topic	Topic	Topic	Content/Type
Standard API	N	Y	N	Y
Standard Wire	Y	N	Y	Y
Transport	TCP	TCP	TCP	UDP or TCP or SHMEM...
QoS	Y(3)	Y(4)	Y(3)	Y (20*)
Standard Payload Format	N	N	N	Y: CDR
Filtering	Content	Content	N	Content/Time

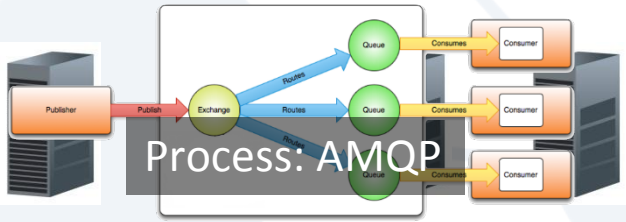
...so which one to use?

IT DEPENDS!

Choose AMQP If…

- Distributing work, not information?
- Just send A to B?
- Speed & CPU use not important?
- Can't lose anything?

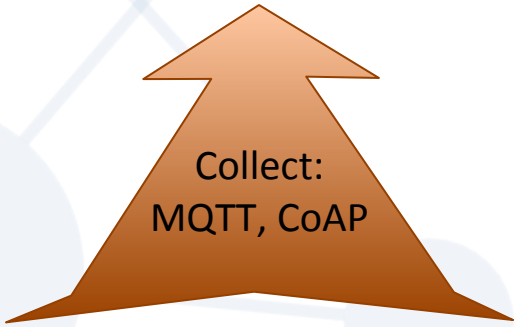
- 3 or 4 => AMQP



Choose MQTT If…

- ✓ Think of it as collection?
- ✓ Little device-device communications?
- ✓ Large number of devices?
- ✓ Very small devices?

- 3 or 4 => MQTT



Choose XMPP/REST If…

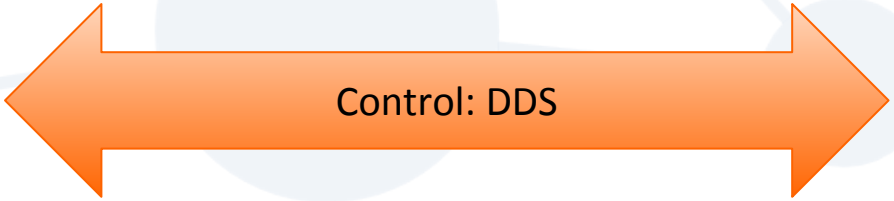
- Use the word “my”?
- Few connectivity points in large space?
- Speed & CPU use not important?
- “Always” connected?

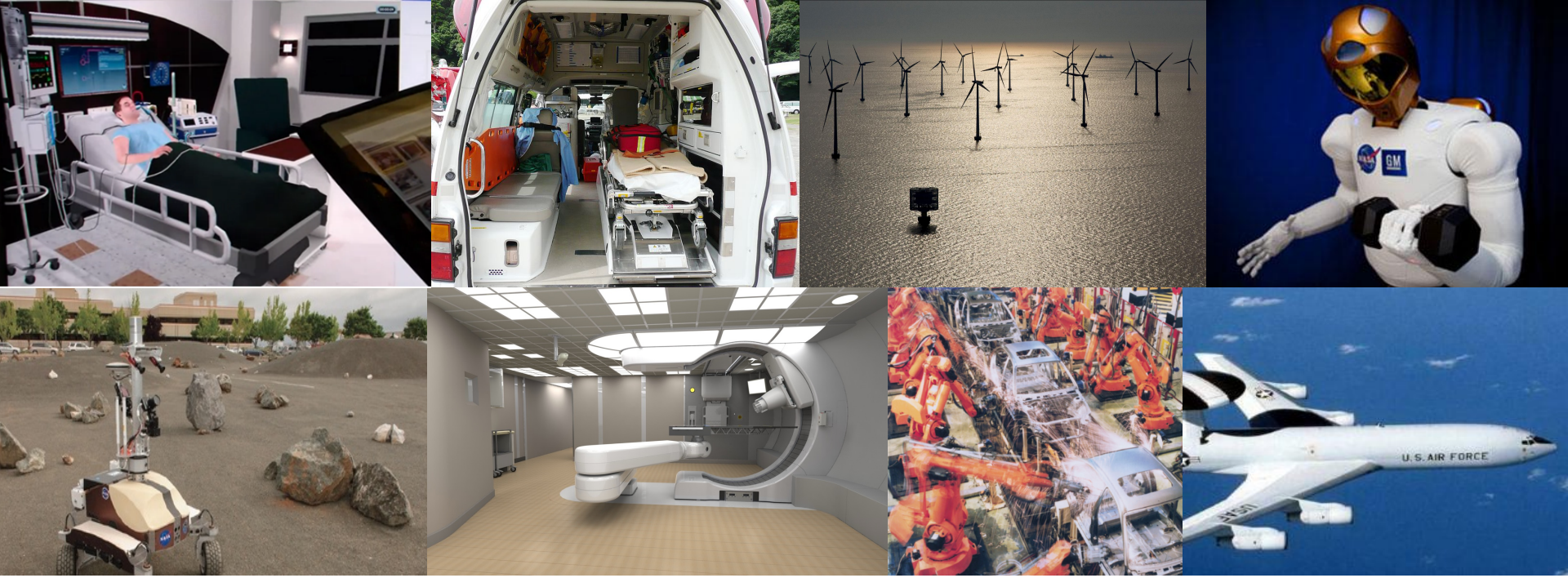
- 3 or 4 => XMPP or REST

Choose DDS If…

- Disaster if offline for 5 minutes?
- Measure performance in ms or us? Or scale >100+ applications? Or 10k+ data values?
- Code actively developed for >3 yrs?

- 2 or 3 => DDS



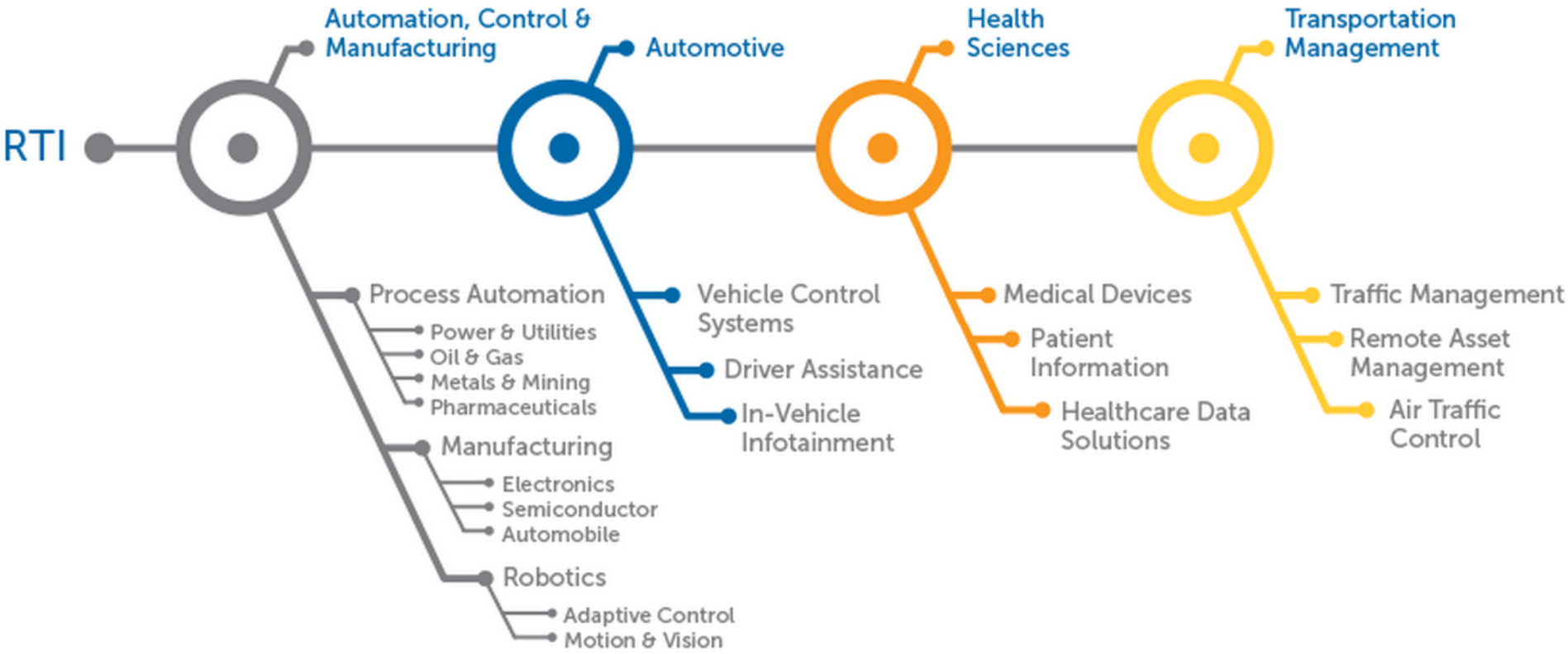


DDS in the IIoT

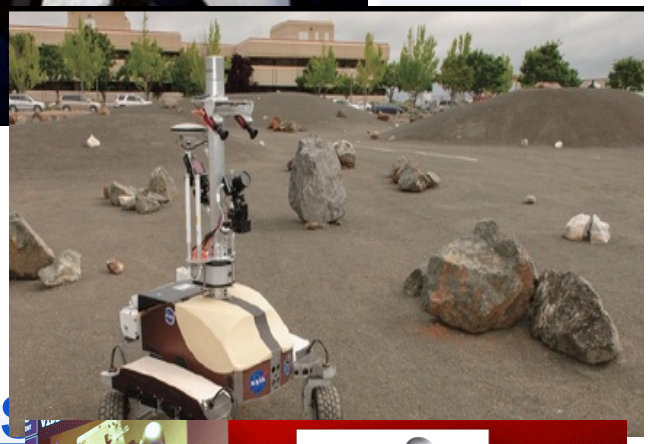
Real world successful examples

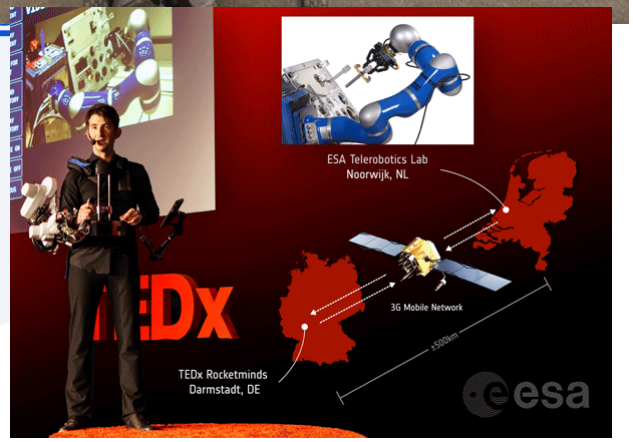
http://www.rti.com/industries/iot.html

IIoT from different fields



Robotics





- NASA/ESA
  - Share data over disadvantaged links
  - Control Room Supervisory
  - Remote control from satellite
  - Coordination between sens and actuators [Video][TedX] [Demo]
- Unmanned vehicles
  - SAR

Energy

- Siemens Wind Power
  - IT integration for maintenance
- LocalGrid
  - Monitor and control
- Grand Coulee Dam
  - Largest Electricity Producer
  - in US
- Green Energy IIC Testbed

Industrial Applications

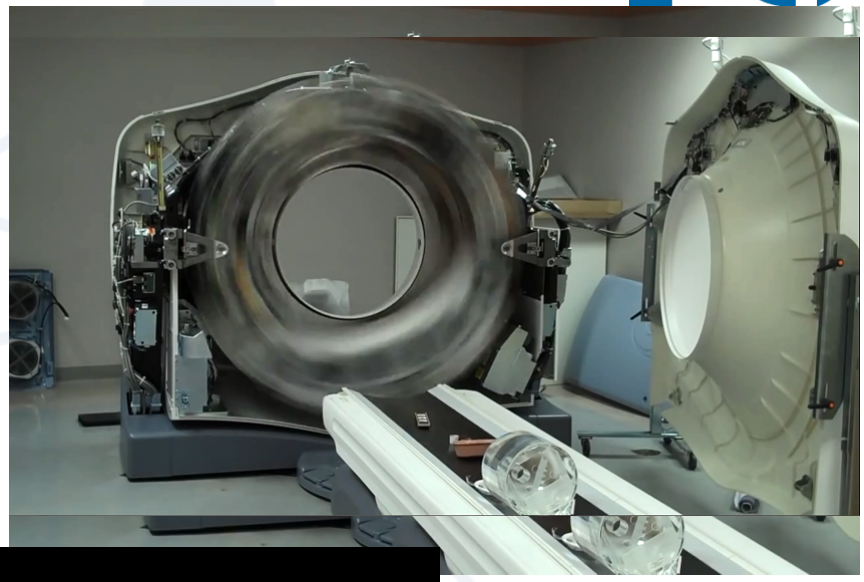
- Control
  - DDS connects the controller, the GUI and the historian
- Integration
  - DDS communicates powerful computers and embedded devices

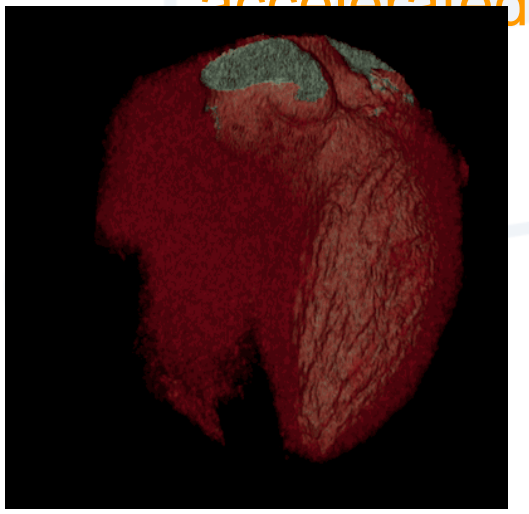
Medical Applications

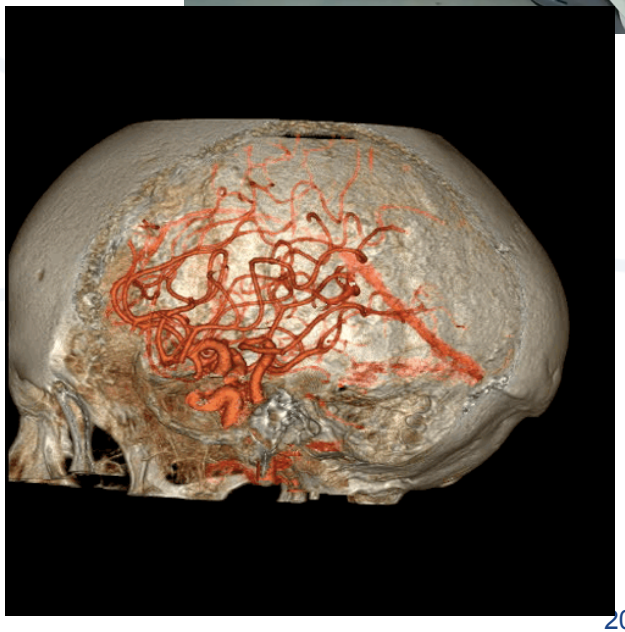
- Hospital management
  - Continuous treatment
  - Preventing 50.000 deaths/year (article)
- Patient monitoring
  - DDS connects devices with decision engines
  - 60% emergency vehicles!
- Interoperability
  - ICE standard
- Reliability
  - Mevion's Proton-Beam Radiation Therapy system zaps tumors with accelerated protons
  - Continuous treatment

Medical: Imaging and treatment

- Imaging
  - GE Healtcare (link)
- Treatment
  - Mevion's Proton-Beam Radiation Therapy system zaps tumors with accelerated protons







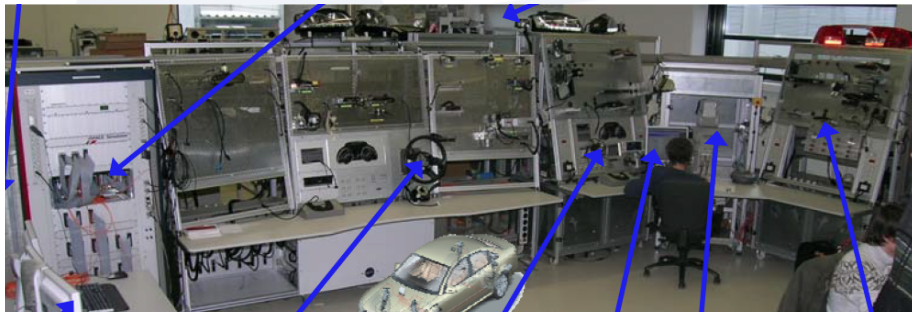
Transportation: Smart cars

- Volkswagen
  - VW Driver Assistance and Integrated Safety
    - radars, laser range finders, and video integration
  - Avoid obstacles, detect lane departures, track eye activity, and safely negotiate bends
- Audi
  - (video)
- Bridge high speed networking to the CAN bus

Transportation: Distributed testing

Audi

- A modern car contains 100+ CPUs
- Audi uses hardware-in-the-loop simulation to feed realistic data to components in a lab for testing
- RTI middleware enables a modular test environment that scales to work with hundreds of devices



Transportation

- Rail Tracking
- Flight Training Simulators
- Air Traffic Control
  - NAV Canada: 2nd ANSP in the world
  - 24x7 Operation

Thanks for your attention!

Any questions?

gianpiero@rti.com / @magopieri

References:

- Download connector here: https://github.com/rticommunity/rticonnextdds-connector
- More info on connector here: https://community.rti.com/downloads/experimental/rticonnextdds-connector
- Examples (both node and python): https://github.com/gianpiero/dds-firststeps
- For any question contact me or write on our forum: https://community.rti.com/forums/technical-questions