2023 ODVA

Industry Conference and 22nd Annual Meeting

Enabling Data Scientist Use Cases with Discoverability and Metadata

Greg Majcher

Rockwell Automation

Data is Valuable

"Data is the new oil"

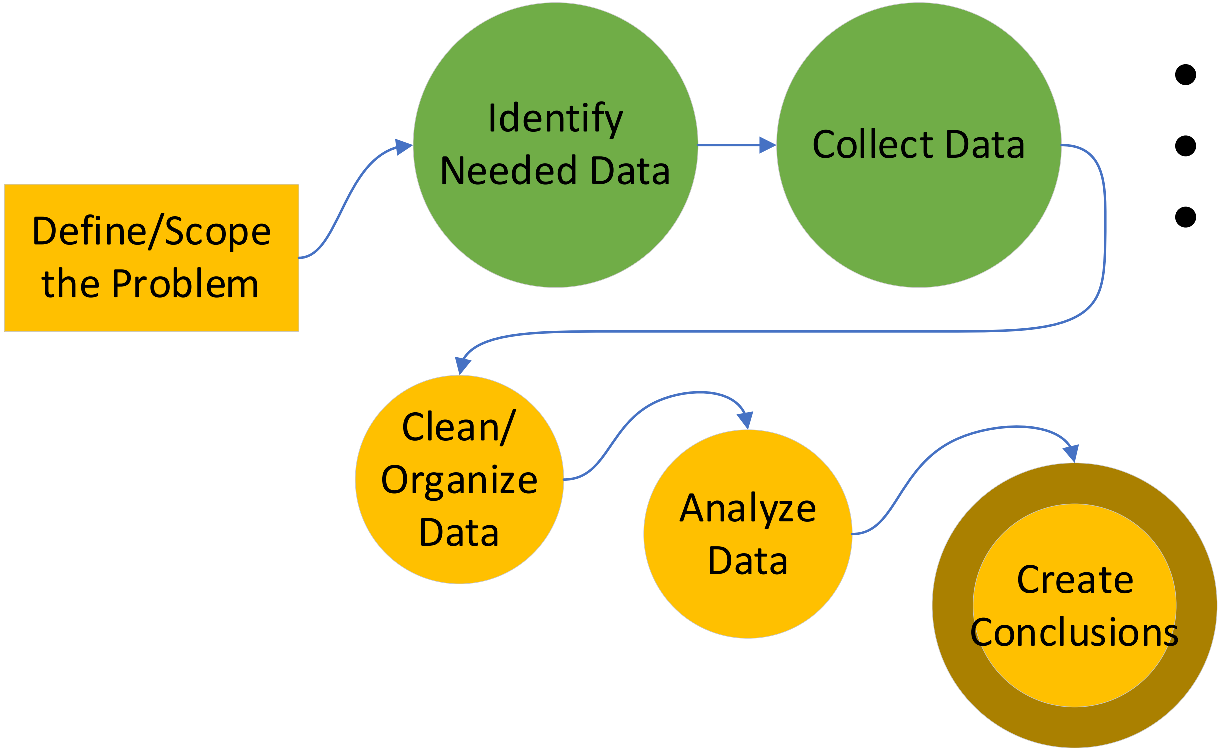
Data Science Introduces New Use Cases

Data Scientists

- What are they looking for? It depends
- How do they want to access data? Don’t care
- How much data? It depends
- How often? It depends

This Photo by Unknown Author is licensed under CC BY-SA-NC

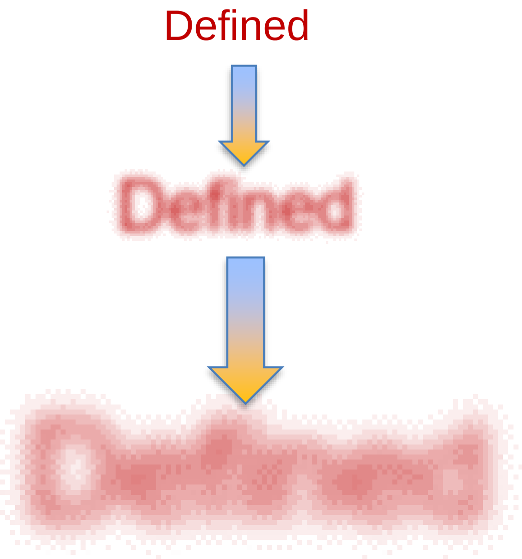
Data Science Workflow



- Discover
- Understand
- Deliver

Example User Stories

- Identify areas for energy savings
- Predict when a component will fail
- Increase the efficiency of a process
- Identify deteriorating quality in a process or a product being produced
- Diagnose performance or quality differences between similar production lines or facilities



Defined

Defined

Defined

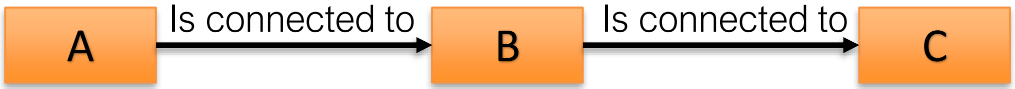
How Can We Help

- Discover
  - Mechanisms to discover devices and the data they possess
- Understand
  - Metadata to help understand the data
- Deliver
  - Efficient delivery mechanisms

🔍 Discovering Devices

Discover Devices

- LLDP



- ListIdentity

B

Is From **Vendor** A, Has **Product Type** B & **Product Code** C
Is at **Revision** D
Has Current **Status** of E
Has a **Serial Number** F and **Product Name** “G”
And is currently in **State** F

Discover Devices

What could we do better?

Learn about device capabilities during discovery

CIP Security Profiles are delimited in the ListIdentity response

We can expand this idea to include other features/capabilities

Discovering Data Online

Discovering Data Online

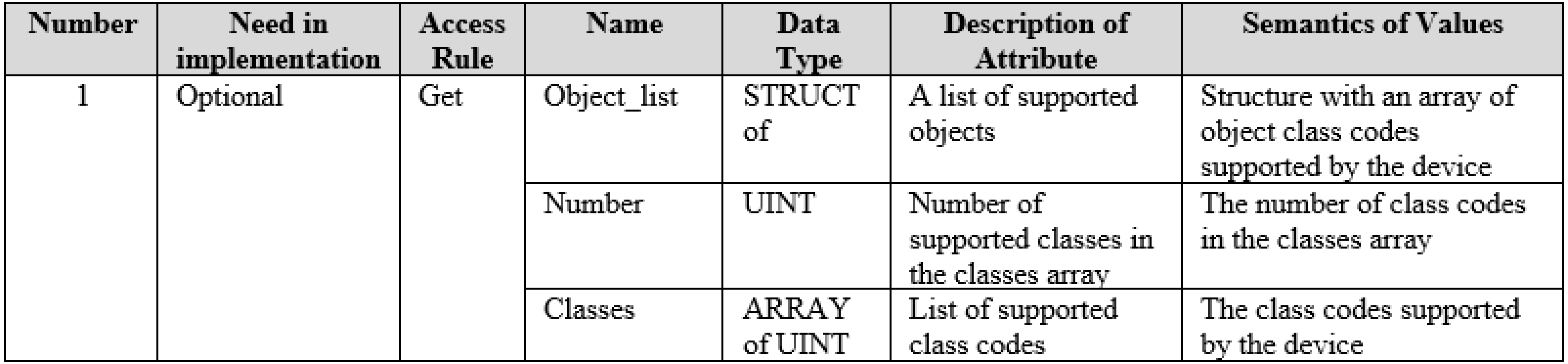
Using a Brute Force Technique

- Send request to all possible classes (65,535)
- Send request to all instances of classes present (4,294,967,295)
- Send request to all possible attributes (65,535)

- 65,535 + (4,294,967,295 x C) + (65,535 x I) = MANY, many messages

Discovering Data Online with Better Implementations

Message Router Instance Attribute 1 eliminates 65,534 messages

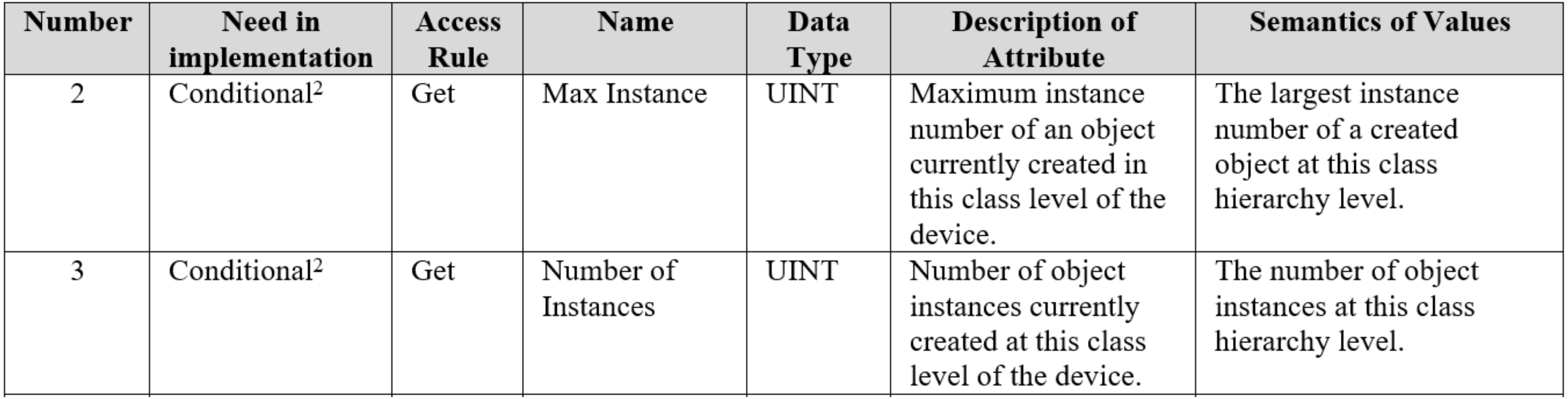


Number	Need in implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
1	Optional	Get	Object_list	STRUCT of	A list of supported objects	Structure with an array of object class codes supported by the device
			Number	UINT	Number of supported classes in the classes array	The number of class codes in the classes array
			Classes	ARRAY of UINT	List of supported class codes	The class codes supported by the device

1 ~~65,535~~ + (4,294,967,295 x C) + (65,535 x I) = MANY messages

Discovering Data Online with Better Implementations

Class Attributes 2 and 3 *could* eliminate over 4 billion messages

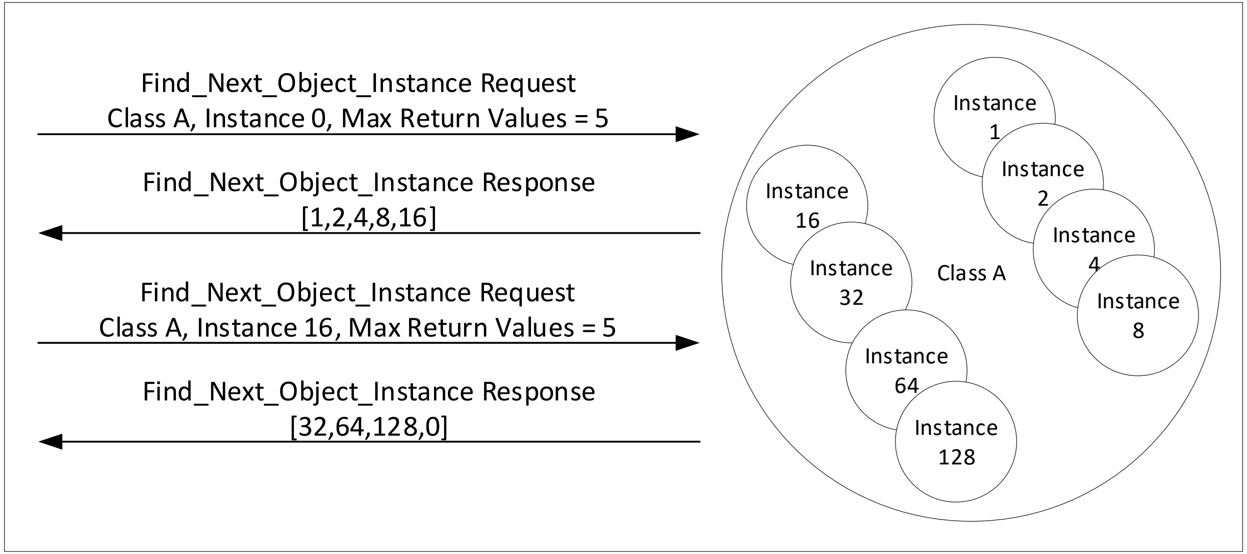


Number	Need in implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
2	Conditional[2]	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.
3	Conditional[2]	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device.	The number of object instances at this class hierarchy level.

1 ~~65,535~~ + (2 ~~4,294,967,295~~ x C) + (65,535 x I) = thousands of messages

Discovering Data Online with Better Implementations

Find_Next_Object_Instance* could be used for sparsely populated lists



Find_Next_Object_Instance Request
Class A, Instance 0, Max Return Values = 5

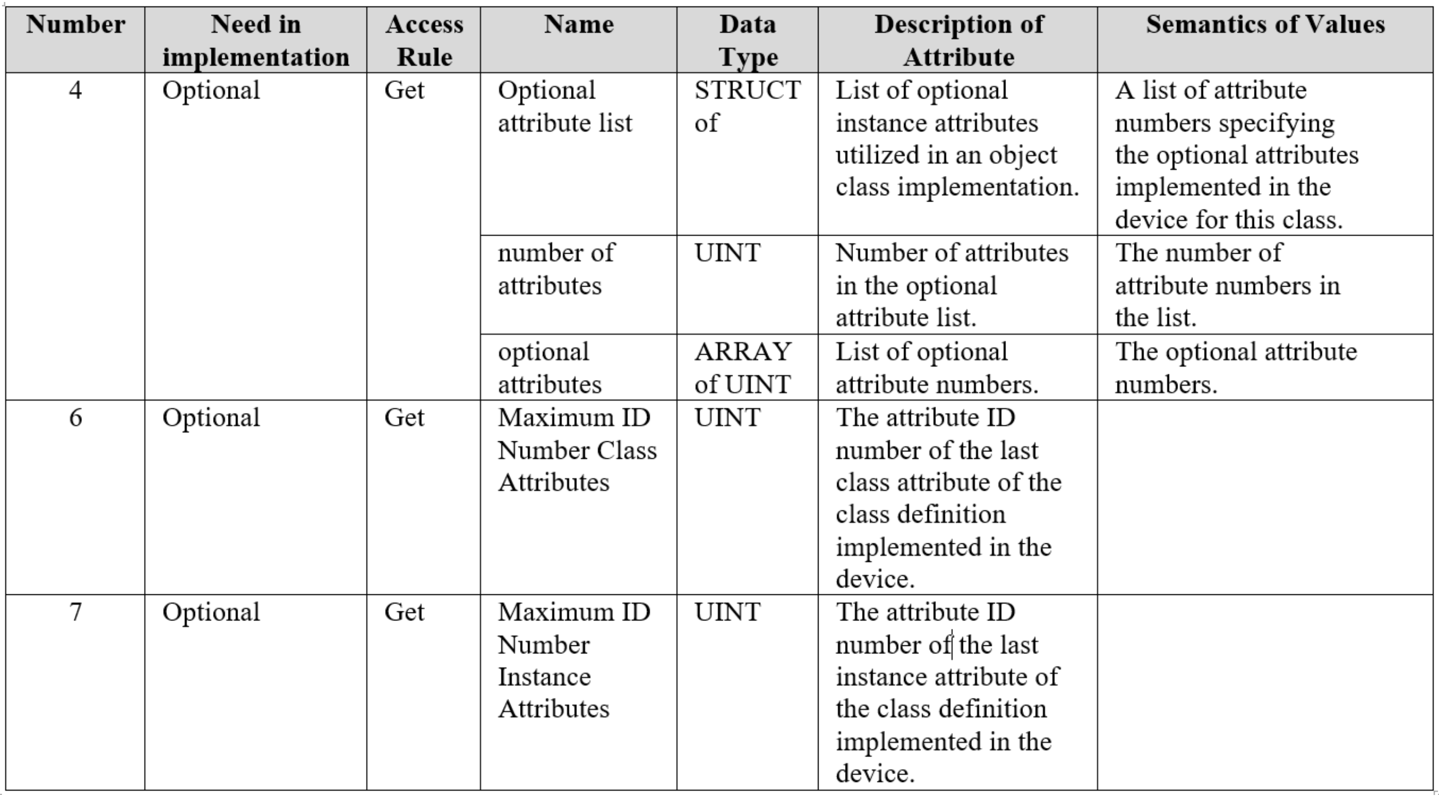
Find_Next_Object_Instance Response
[1,2,4,8,16]

Find_Next_Object_Instance Request
Class A, Instance 16, Max Return Values = 5

Find_Next_Object_Instance Response
[32,64,128,0]

*Note, this service cannot be used for UDINT instances

Discovering Data Online with Better Implementations



Number	Need in implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
4	Optional	Get	Optional attribute list	STRUCT of	List of optional instance attributes utilized in an object class implementation.	A list of attribute numbers specifying the optional attributes implemented in the device for this class.
			number of attributes	UINT	Number of attributes in the optional attribute list.	The number of attribute numbers in the list.
			optional attributes	ARRAY of UINT	List of optional attribute numbers.	The optional attribute numbers.
6	Optional	Get	Maximum ID Number Class Attributes	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device.	
7	Optional	Get	Maximum ID Number Instance Attributes	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device.	

Discover Data Online

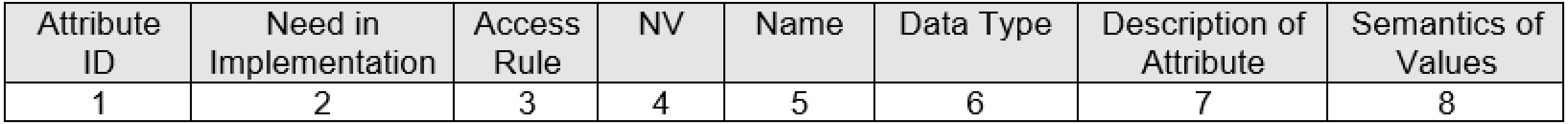
What could we do better?

New class attributes for supported instances and attributes modeled after the Message Router’s Object_List attribute

Understanding Data Online

Understanding Data Online

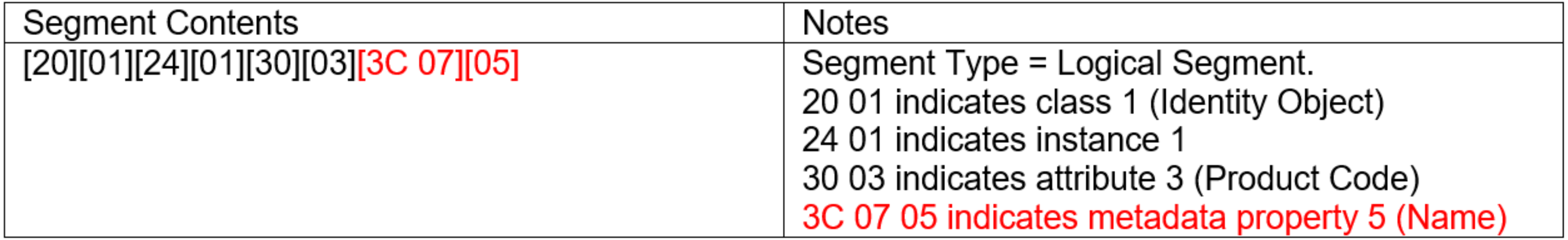
“Metadata” is documented in the specifications, but it is not exposed online



Attribute ID	Need in Implementation	Access Rule	NV	Name	Data Type	Description of Attribute	Semantics of Values
1	2	3	4	5	6	7	8

What if these columns were reported as standardized attribute properties?

We could define logical segments to retrieve them.



Segment Contents	Notes
[20][01][24][01][30][03][3C 07][05]	Segment Type = Logical Segment. 20 01 indicates class 1 (Identity Object) 24 01 indicates instance 1 30 03 indicates attribute 3 (Product Code) 3C 07 05 indicates metadata property 5 (Name)

Discovering Data Offline

Discovering Data Offline

Yes, this is a valid EDS

```
[File]
        DescText = “NotSoSmart LazyBoy Widget EDS File";
        CreateDate = 04-01-1999;       $ Create date
        CreateTime = 17:51:44;
        Revision = 1.1;                $ Revision of EDS
[Device]
        VendCode = 65535;
        VendName = “NotSoSmart, Inc.";
        ProdType = 43;
        ProdTypeStr = “Generic Device”;
        ProdCode = 42;
        MajRev = 1; $ Device Major Revision
        MinRev = 1; $ Device Minor Revision
        ProdName = "LazyBoy Smart-Widget";

[Device Classification]
        Class1 = EtherNetIP;
```

Public Object Class and Vendor Specific Object Class EDS Sections

```
[Discrete Input Class]
        Revision = 2;            $ Revision 2 of the object is implemented
        MaxInst = 8;             $ The highest instance number that exists in the product is 8
        Number_Of_Static_Instances = 8;        $ There are 8 static instances present
        Number_Of_Dynamic_Instances = 0;       $ There are no dynamic instances
        Class_Attributes = 1;                $ Class attribute 1 is supported
        Instance_Attributes = 3, 4, 5, 6;    $ Instance attributes 3, 4, 5, and 6 are supported
        Class_Services = 0x14;               $ Get is supported for class attributes
        Instance_Services = 0x14, 0x10;      $ Get and Set are supported for inst attributes
        Object_Name = "Discrete Input Point Object"
        Object_Class_Code = 0x08;
```

Understanding Data Offline

Understanding Data Offline

```
[Params]
Param1 = 0,
6,"20 06 24 01 30 F0", $ Size, Link Path to Conn Mgr
Object
0x0002,                    $ Descriptor
0xC7, 2,                   $ UINT Data Type, Data Size
"Percent I/O Utilization", $ Name
"%",                       $ Units
"Indicates what percentage of the I/O comms resources are
in use in this device in units of 0.1%", $ Help string
0,1000,0,                  $ Min/Max/Default
,,,,                       $ Unused fields
,,,,
;
```

Delivering Data

Delivering Data

- Request / Response → Unconnected & Class 3
- Small amounts of varied data → Multiple_Service_Packet
- Larger amounts of data → Send_Receive_Fragment
- Lots of data often → New transports

Summary and Call to Action

- There is a lot we can do immediately through vendor adoption
- We can improve some of the existing definitions
- We can define better alternatives

Summary Plan

Near Term (Today)

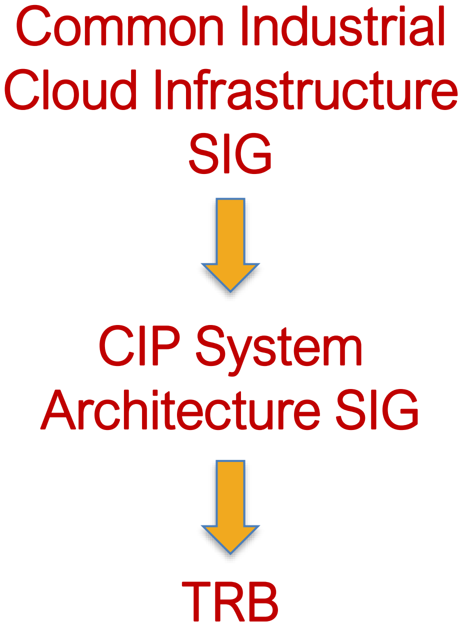
- Fully-defined EDS files

Medium Term (Next Releases)

- Support for Message Router’s Object_List
- Support for Max Instance and Number of Instances
- Support for Param entries to define data

Long Term (Future Publication Cycles)

- Enhance existing definitions
- Define & Support online metadata
- Define & Support new transport protocols where needed



Common Industrial Cloud Infrastructure SIG

CIP System Architecture SIG

TRB

2023 ODVA

Industry Conference and 22nd Annual Meeting

Thank You!

Questions?