

# Open Platform Communications Unified Architecture **OPCUA**

(released in 2008)

**Majid Aminian**



**01/08/2019**

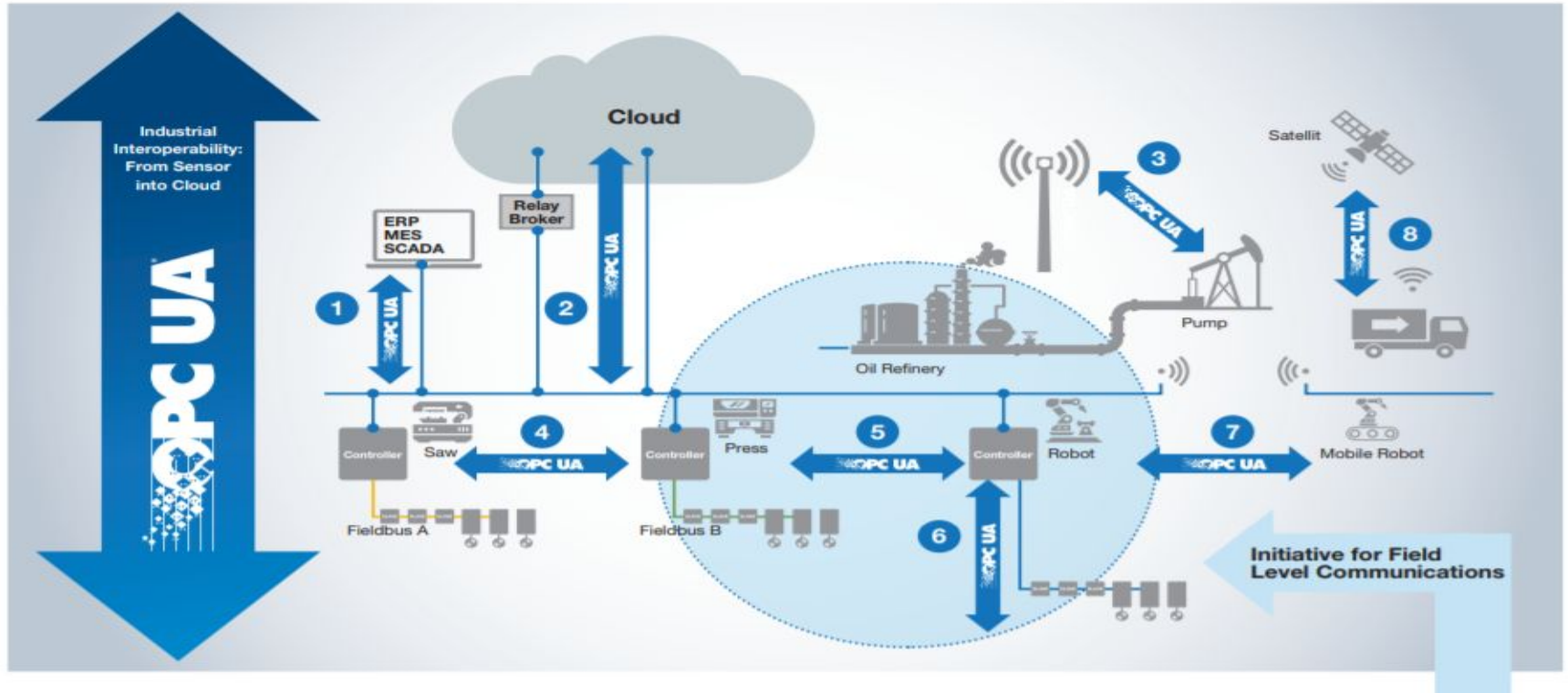
**OPC UA information modeling framework turns data into information and transport of machine-to-machine data and interfaces and the semantics of data**

**OPC UA standardizes industrial connectivity and ensures interoperability between products from different manufacturers**



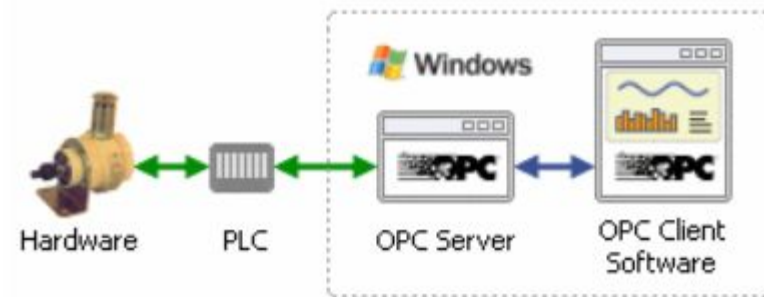
# The Industrial Interoperability Standard

## OPC UA from Sensor to Cloud

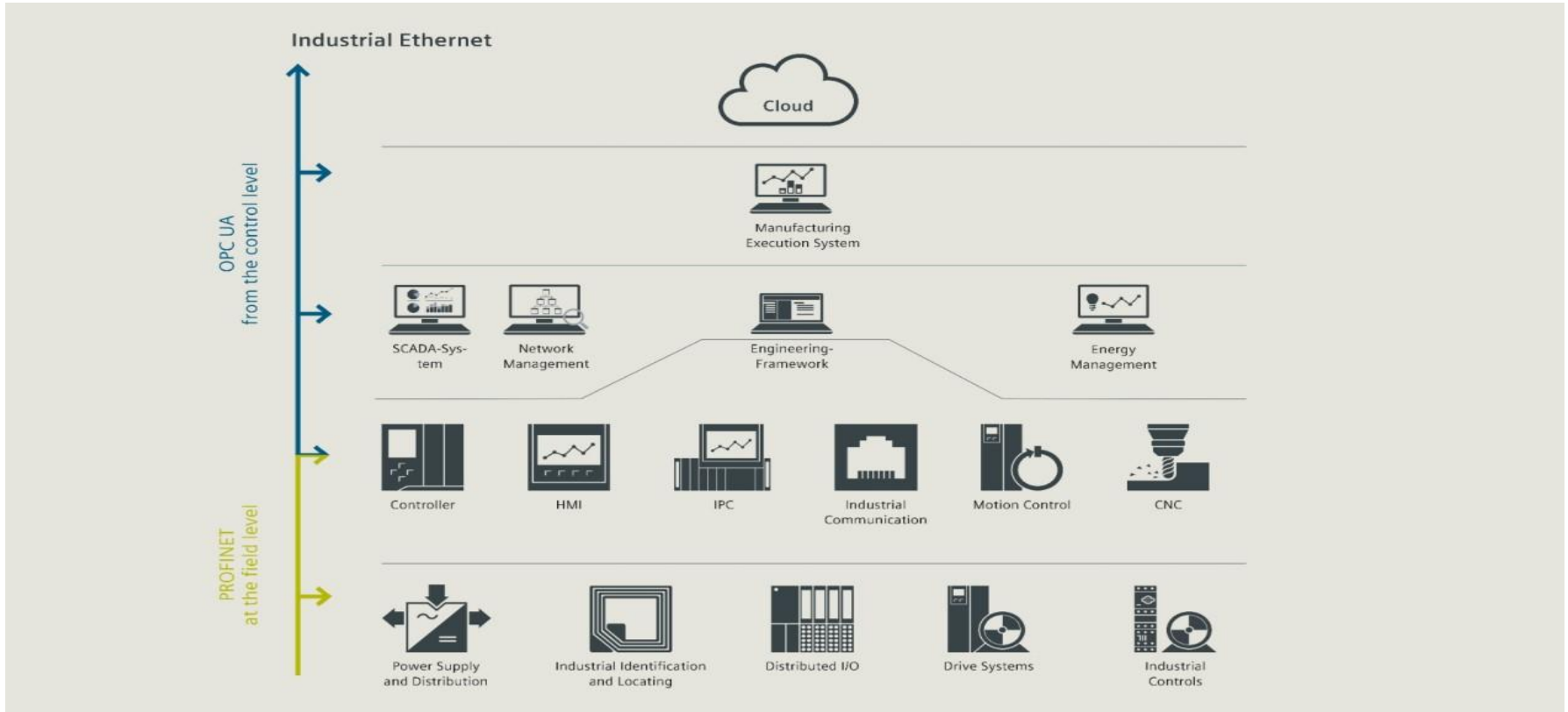




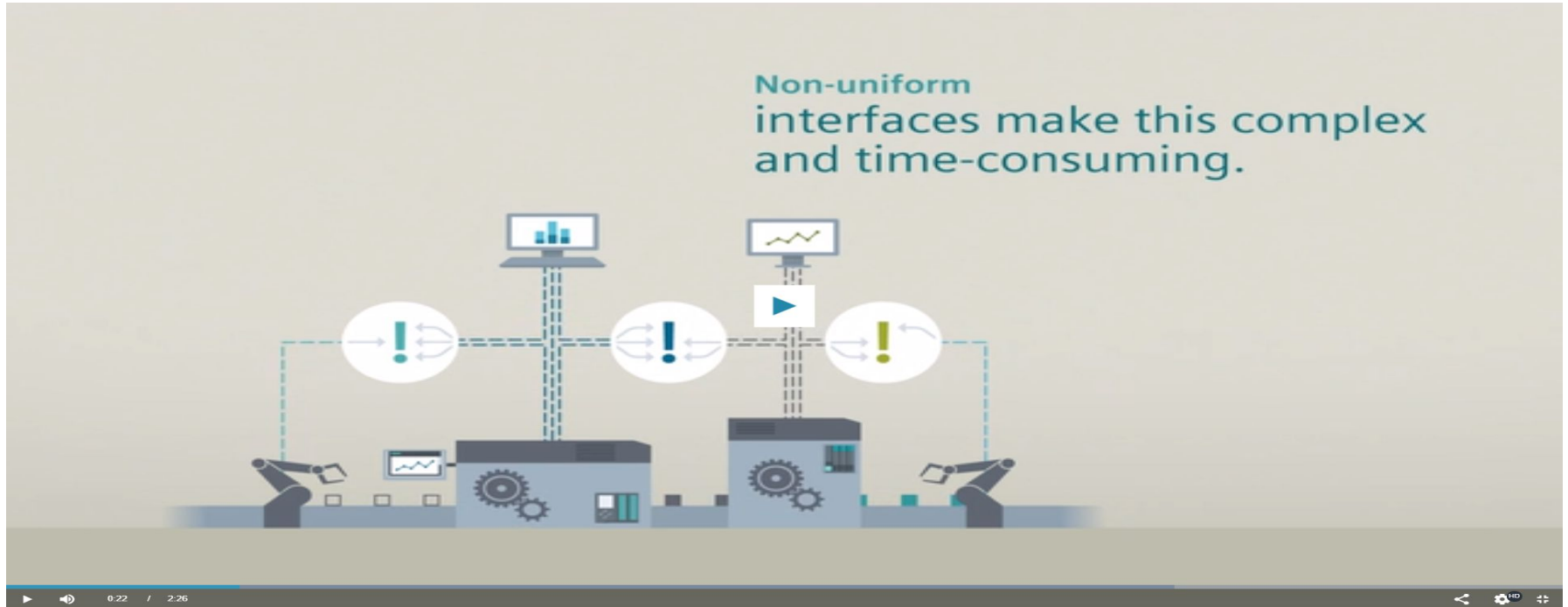
## OPCUA Process



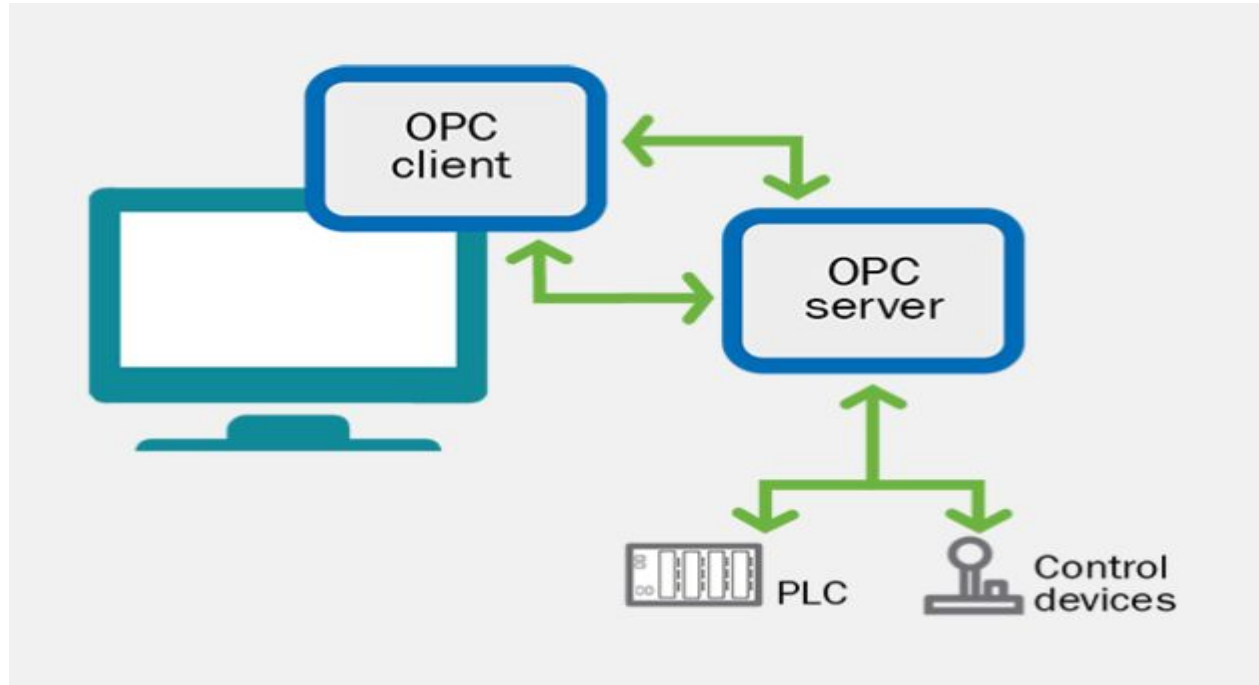
**OPC UA provides a very flexible and adaptable mechanism for moving data between enterprise-type systems and the kinds of controls, monitoring devices and sensors that interact with real world data**



**A Robust Set of Services –OPC UA provides a full suite of services for eventing, alarming, reading, writing, discovery and more.**

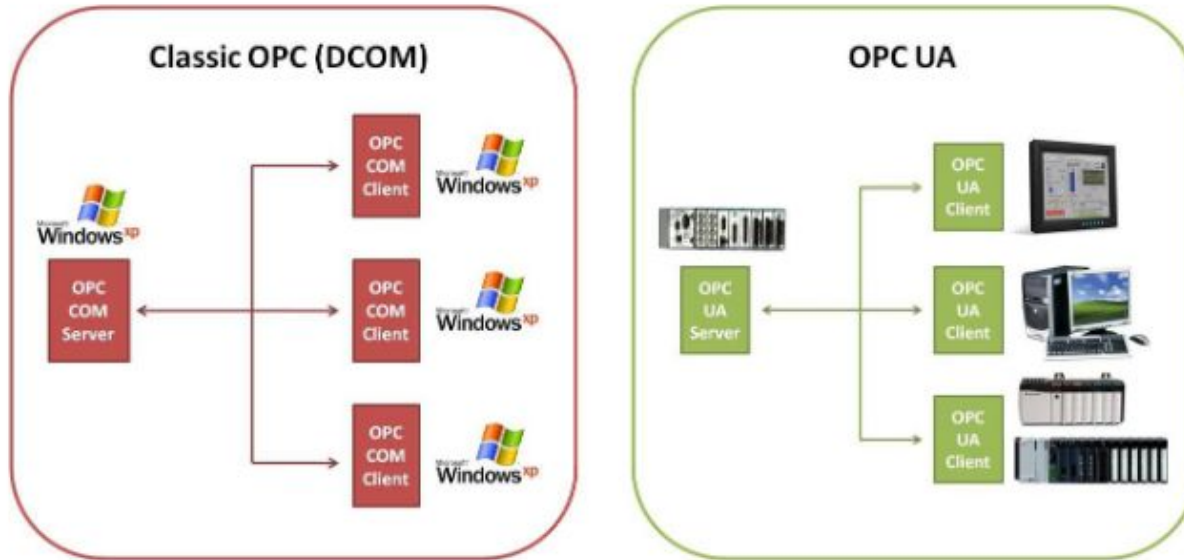


**Internet Capability –OPC UA is fully capable of moving data over the Internet**



available hardware platforms and operating systems:

- **Hardware platforms:** traditional PC hardware, cloud-based servers, PLCs, micro-controllers (ARM etc.)
- **Operating Systems:** Microsoft Windows, Apple OSX, Android, or any distribution of Linux, etc.





## **OPCUA client application:**

In this application we can manage data that was sent by our codes(python codes) and watch streaming data .

Note: we could use different kind of free applications for OPCUA client and server.

In my project i used 'Prosys OPC UA Client' .

# OPCUA Client Environment:

Prosops OPC UA Client Pro Beta

Help

FreeOpcUa Python Server +

Running opc.tcp://192.168.0.1:7000 -- urn:freeopcua:python:server

Search

- Objects
  - MyObject
    - Data\_01
    - Data\_02
    - Data\_03
    - Data\_04
    - Radarmodell0
    - Radarmodell1
    - Radarmodell3
    - Radarmodell4
    - id
    - majid1
    - myvar\_unix\_time
    - sample\_number
    - time\_uncertainty
    - unix\_time\_nsecs
  - Server
- Types
- Views

stream data

Attributes and References +

Attribute	Value
NodeId	ns=2;i=8
NodeClass	Variable
BrowseName	2:Data_01
DisplayName	Data_01
Description	Data_01
WriteMask	NONE (0)
UserWriteMask	NONE (0)
Value	Data_01=0.28141576051712036
Data Type	Int64
ValueRank	Scalar
ArrayDimensions	[]
AccessLevel	CurrentRead, CurrentWrite
UserAccessLevel	CurrentRead, CurrentWrite
MinimumSamplingInterval	0.0
Historizing	false

Filters Browse Direction Forward

ReferenceType	Target
HasTypeDefinition	BaseDataVariableType

Information and description of each data

11:38 25.07.2019

# OPCUA Client Environment:

Prosyp OPC UA Client Pro Beta

Help

FreeOpcUa Python Server +

Running opc.tcp://192.168.0.1:7000 - urn:freeopcua.python.server

Search

Objects

MyObject

- Data\_01
- Data\_02
- Data\_03
- Data\_04
- Radarmodel0
- Radarmodel1
- Radarmodel3
- Radarmodel4
- id
- majid1
- myvar\_unix\_time
- sample\_number
- time\_uncertainty
- unix\_time\_nsecs

Server

- Types
- Views

Attributes and References

Data View X +

Subscription Enabled ☒ Publishing Interval (in milliseconds) 1.000 Subscription Settings...

#	NodeId	DisplayName	Value	DataType	SourceTimestamp	ServerTimestamp	StatusCode	MonitoringMode	Graph
0	ns=2;i=8	Data_01	Data_01=0.9710280895233154	Int64	25.07.2019 11:42:40.936	null	GOOD (0x00000000...)	Reporting	<input checked="" type="checkbox"/>

Show data for (seconds) 60 ☐ Own axis for every signal

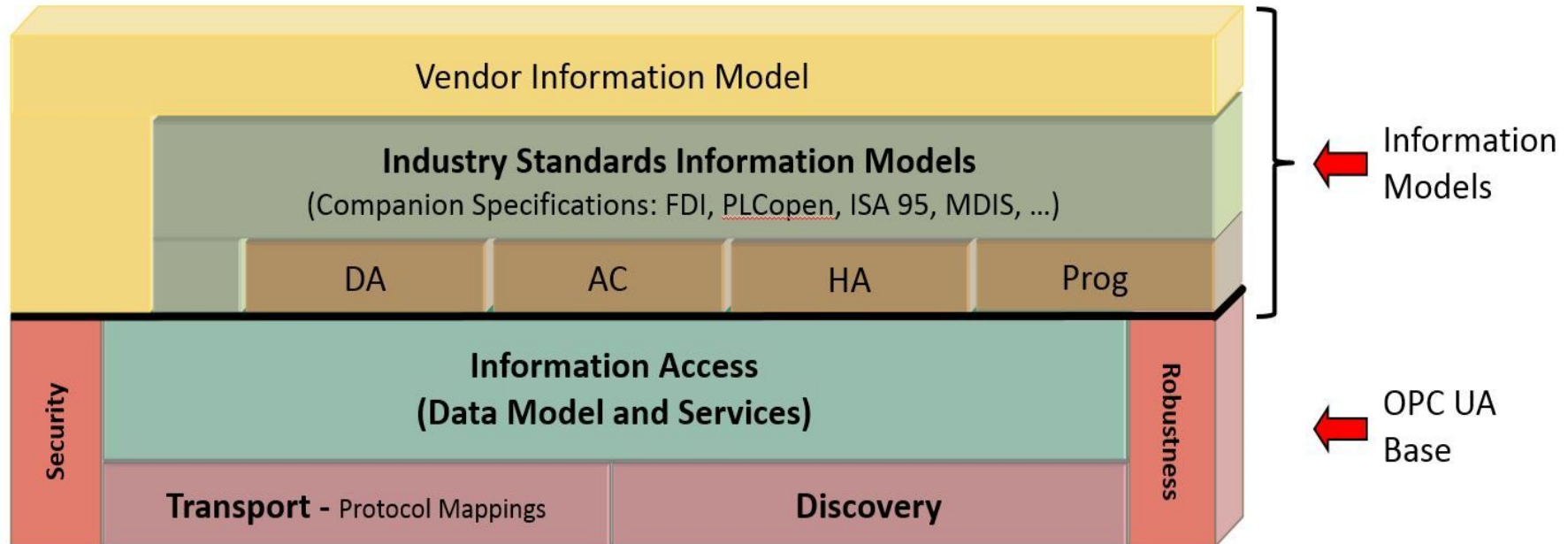
Graph Data\_01

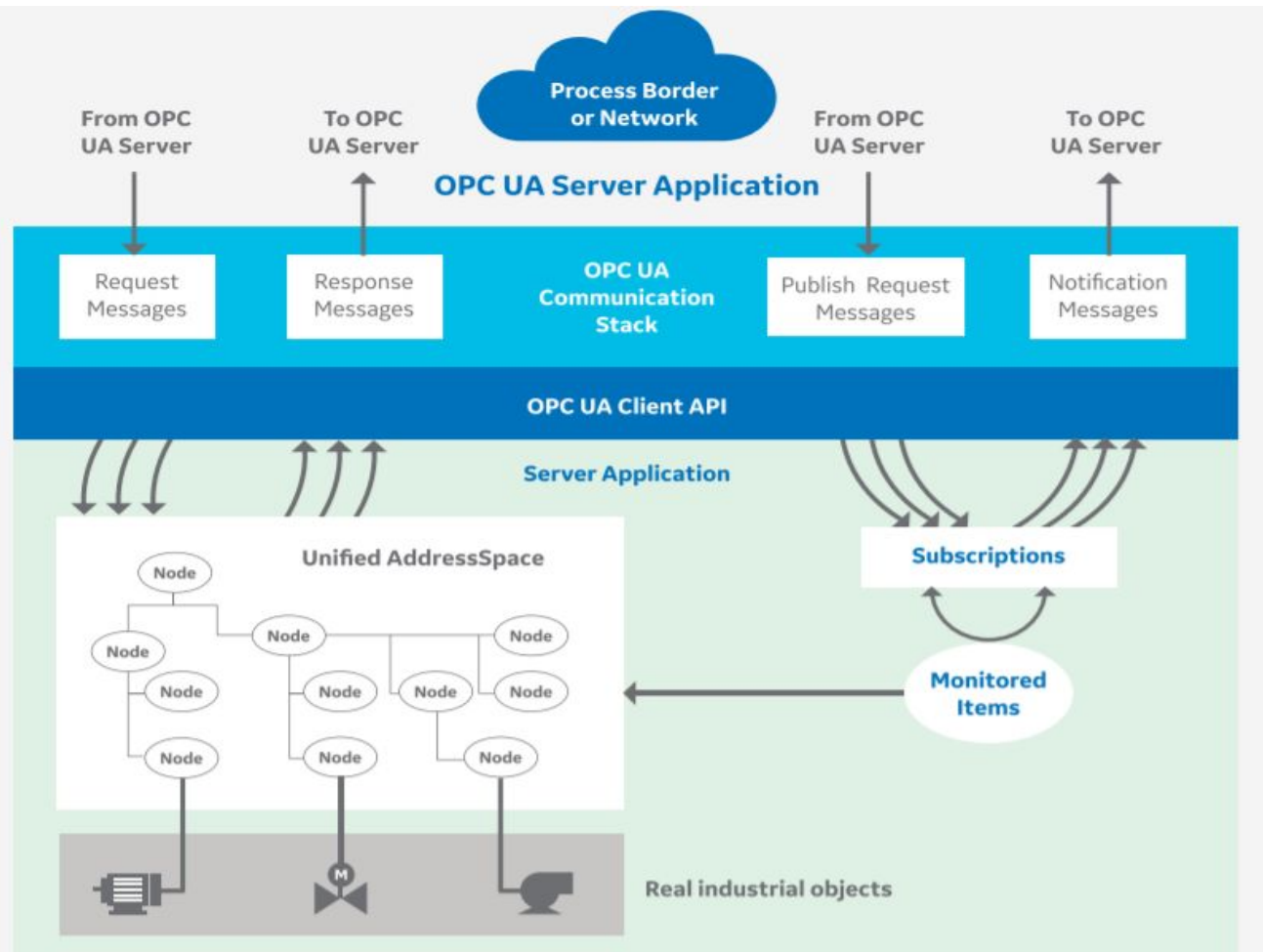
Time	Value	StatusCode
25.07.2019 11:42:26.826	Data_01=0.2747882306575775	GOOD (0x00000000) --
25.07.2019 11:42:27.829	Data_01=0.9575470685958862	GOOD (0x00000000) --
25.07.2019 11:42:28.831	Data_01=0.759941577911377	GOOD (0x00000000) --
25.07.2019 11:42:29.824	Data_01=0.13635072112083435	GOOD (0x00000000) --
25.07.2019 11:42:30.824	Data_01=0.9072827696800232	GOOD (0x00000000) --
25.07.2019 11:42:31.829	Data_01=0.8440632224082947	GOOD (0x00000000) --
25.07.2019 11:42:32.837	Data_01=0.00481584295630455	GOOD (0x00000000) --
25.07.2019 11:42:33.936	Data_01=0.8890573382377625	GOOD (0x00000000) --
25.07.2019 11:42:34.835	Data_01=0.9112909436225891	GOOD (0x00000000) --
25.07.2019 11:42:35.935	Data_01=0.046294305473566055	GOOD (0x00000000) --
25.07.2019 11:42:36.934	Data_01=0.8155558705329895	GOOD (0x00000000) --
25.07.2019 11:42:37.935	Data_01=0.9275877475738525	GOOD (0x00000000) --
25.07.2019 11:42:38.935	Data_01=0.18679971992969513	GOOD (0x00000000) --
25.07.2019 11:42:39.936	Data_01=0.7257310748100281	GOOD (0x00000000) --
25.07.2019 11:42:40.936	Data_01=0.9710280895233154	GOOD (0x00000000) --

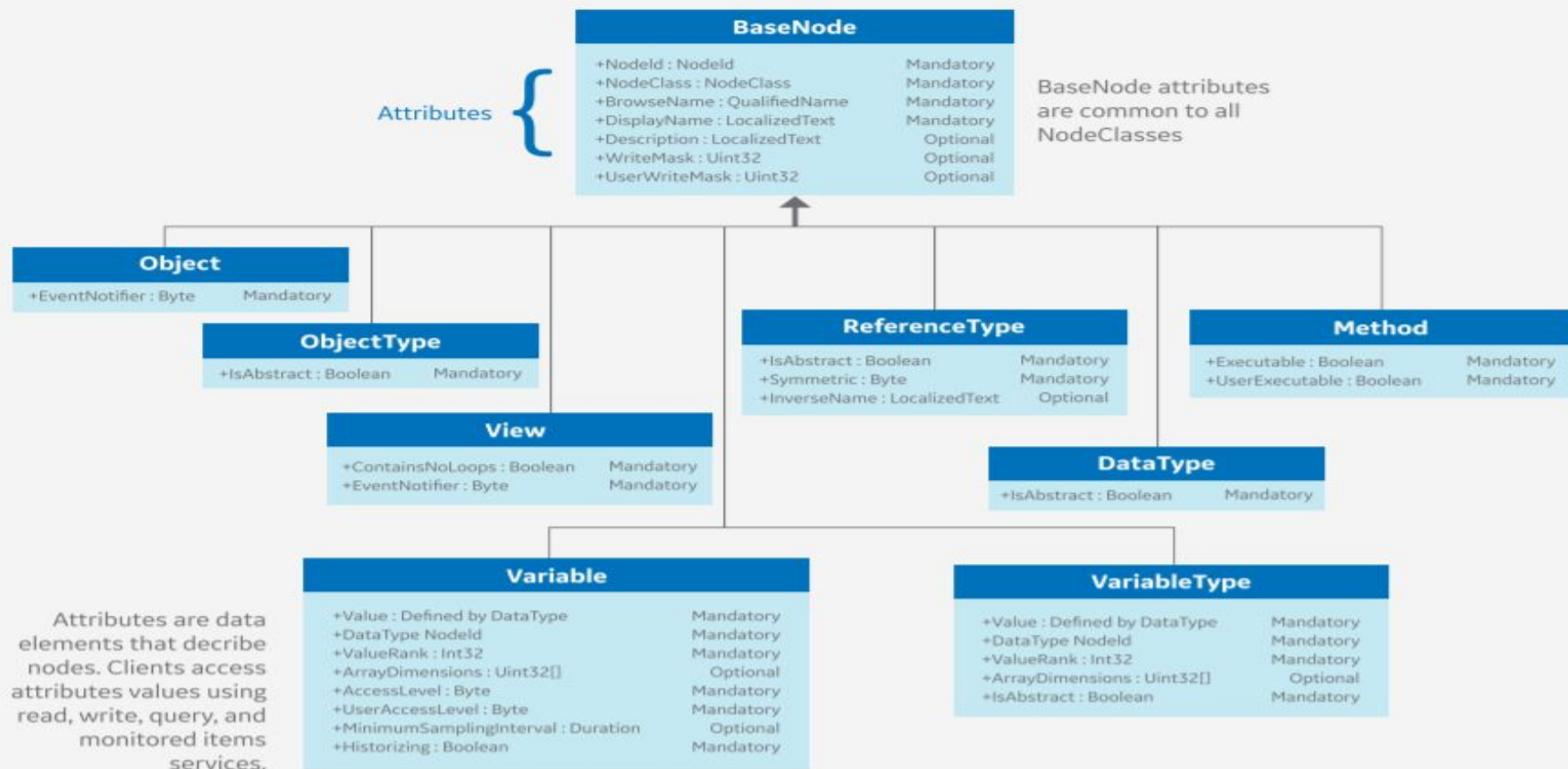
Streaming our data

## Information Modeling:

The OPC UA information Modeling framework turns data into information. With complete object-oriented capabilities, even the most complex multi-level structures can be modeled and extended. Data-types and structures are defined in profiles. For example, the exiting OPC Classic specification were modeled into UA profiles which can also be extended by other organizations:







# Meta data of data(Information Model)

```
33         <Reference ReferenceType="HasComponent">ns=1;i=6005</Reference>
34         <Reference ReferenceType="HasComponent">ns=1;i=6004</Reference>
35     </References>
36 </UAObject>
37 <UAVariable DataType="LocalizedText" NodeId="ns=1;i=6006" BrowseName="1:Data_01" AccessLevel="3">
38     <DisplayName>Data_01</DisplayName>
39     <References>
40         <Reference ReferenceType="HasProperty" IsForward="false">ns=1;i=5002</Reference>
41         <Reference ReferenceType="HasTypeDefinition">i=68</Reference>
42     </References>
43 </UAVariable>
44 <UAVariable DataType="Float" NodeId="ns=1;i=6007" BrowseName="1:Data_2" AccessLevel="3">
45     <DisplayName>Data_2</DisplayName>
46     <References>
47         <Reference ReferenceType="HasTypeDefinition">i=63</Reference>
48         <Reference ReferenceType="HasComponent" IsForward="false">ns=1;i=5002</Reference>
49     </References>
50 </UAVariable>
51 <UAVariable DataType="Float" NodeId="ns=1;i=6008" BrowseName="1:Data_3" AccessLevel="3">
52     <DisplayName>Data_3</DisplayName>
53     <References>
54         <Reference ReferenceType="HasTypeDefinition">i=63</Reference>
55         <Reference ReferenceType="HasComponent" IsForward="false">ns=1;i=5002</Reference>
56     </References>
57 </UAVariable>
58 <UAVariable DataType="Float" NodeId="ns=1;i=6009" BrowseName="1:Data_4" AccessLevel="3">
59     <DisplayName>Data_4</DisplayName>
60     <References>
61         <Reference ReferenceType="HasTypeDefinition">i=63</Reference>
62         <Reference ReferenceType="HasComponent" IsForward="false">ns=1;i=5002</Reference>
63     </References>
64 </UAVariable>
65 <UAVariable DataType="Int64" NodeId="ns=1;i=6001" BrowseName="1:id" AccessLevel="3">
66     <DisplayName>id</DisplayName>
67     <References>
68         <Reference ReferenceType="HasTypeDefinition">i=63</Reference>
69         <Reference ReferenceType="HasComponent" IsForward="false">ns=1;i=5002</Reference>
70     </References>
71     <Value>
72         <uax:Int64>0</uax:Int64>
73     </Value>
74 </UAVariable>
75 <UAVariable DataType="UtcTime" NodeId="ns=1;i=6003" BrowseName="1:myvar_unix_time" AccessLevel="3">
```

## python libraries which we are working with them:

### **1- google protobuf:**

For buffering and getting data on the internet

### **2-OPCUA client and server:**

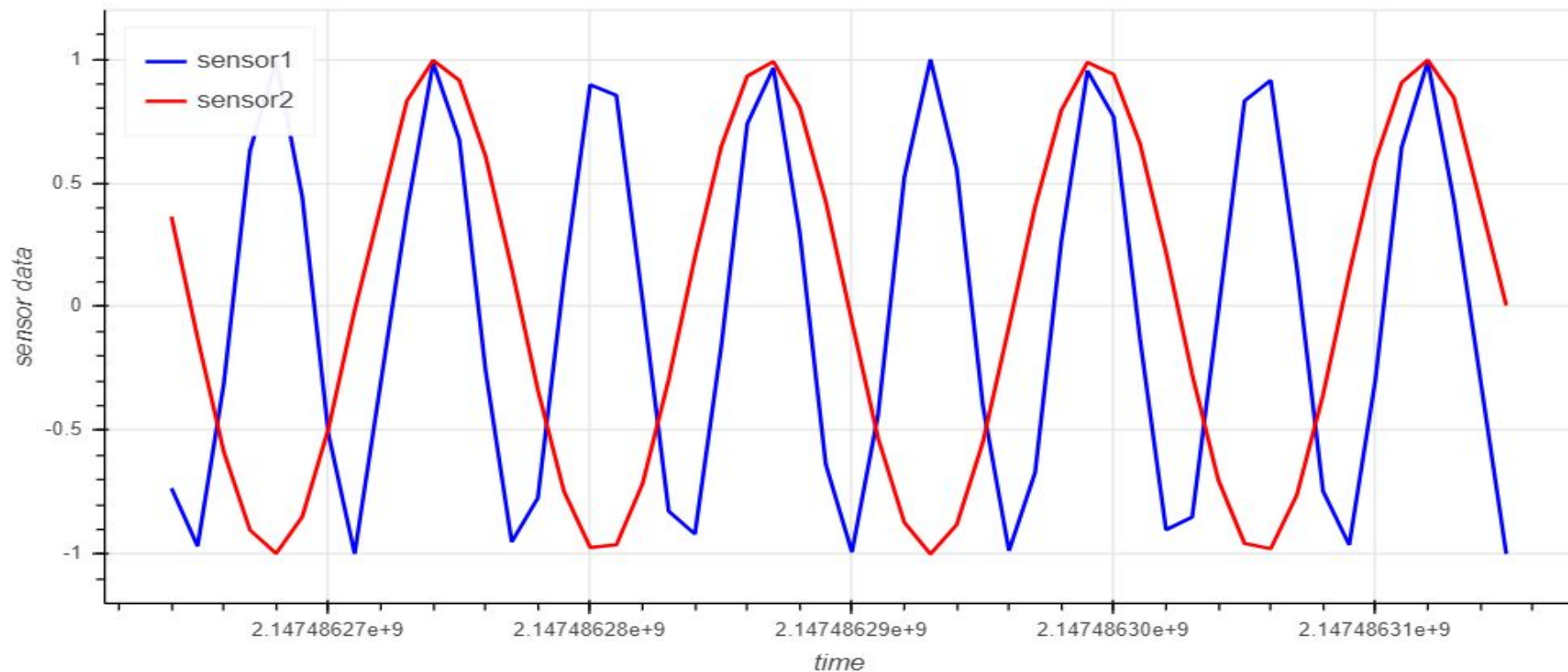
For communicating with opc application

### **3- Streamz and Holoviews:**

for streaming data and plotting them



## A sample of streaming plot:



## Resources for reading more about Information Model:

- <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-5-information-model/>
- <http://www.commsvr.com/UAModelDesigner/Index.aspx>
- [http://www.cas.internetdsl.pl/commserver/P\\_DownloadCenter/P\\_Publications/P-15010101-AddressSpaceInterchangeXML.pdf](http://www.cas.internetdsl.pl/commserver/P_DownloadCenter/P_Publications/P-15010101-AddressSpaceInterchangeXML.pdf)