OPC UA
DEVELOPMENT TRAINING

INTRODUCTION IN THE TECHNOLOGY AND HOW IT WORKS

TECHNO software
Innovative Software Solutions in it's Simplicity

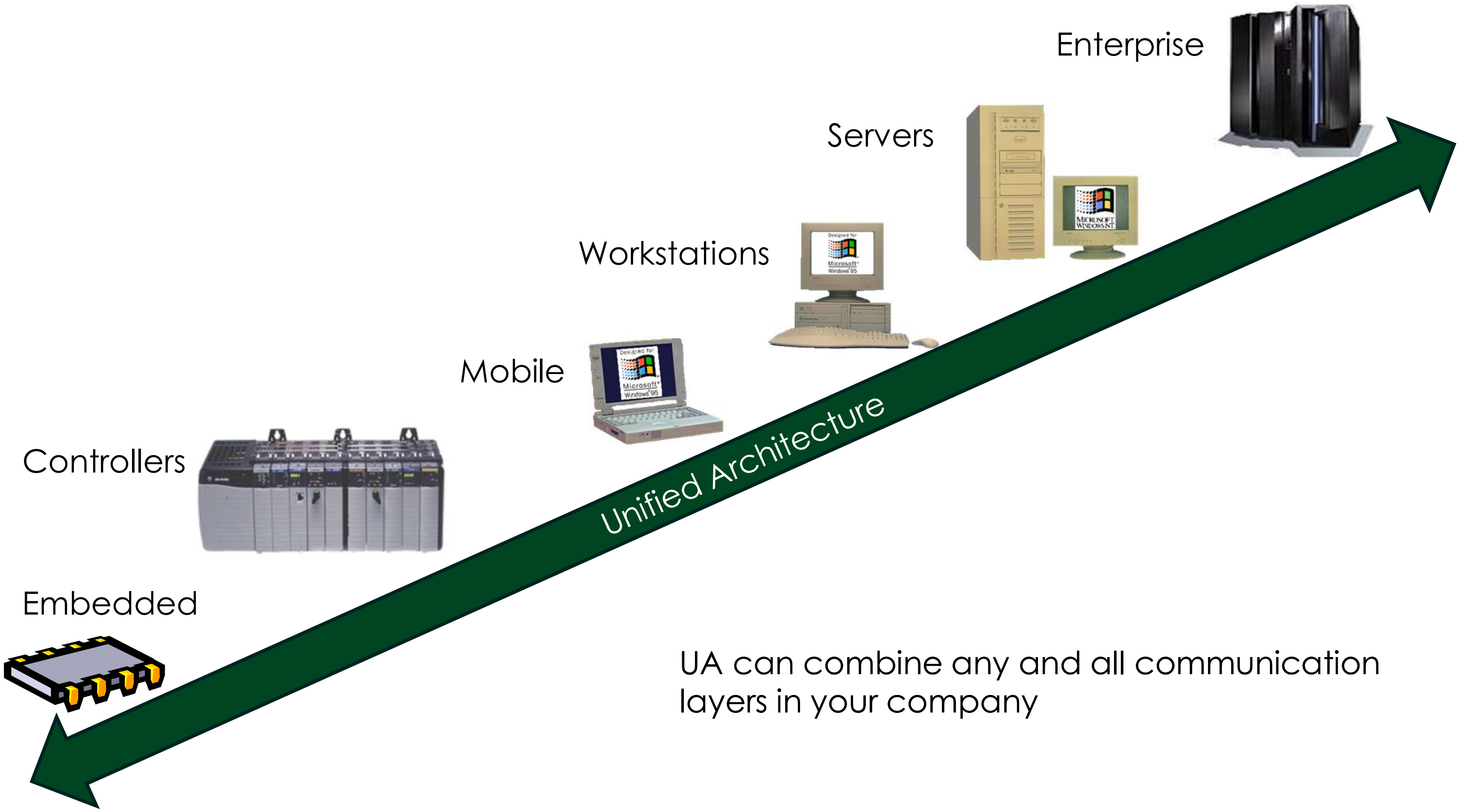
AGENDA

- What is OPC UA? Why replace Classic OPC?
- The OPC UA „Vision“. The goals of OPC UA.
- The OPC UA specification
- Advantages of OPC UA
- Compatibility, Certification
- How OPC UA works

WHAT IS OPC UA?
WHY REPLACE CLASSIC OPC?

- DISCONTINUATION OF COM / DCOM, DCOM LIMITS
- FIREWALL INTER-PROCESS COMMUNICATION
- COMMUNICATION OPPORTUNITIES ON THE INTERNET BECOME INCREASINGLY IMPORTANT
- MORE PROTECTION AGAINST UNAUTHORIZED DATA ACCESS
- EACH CLASSIC OPC SPECIFICATION IS SELF-CONTAINED
- CROSS-PLATFORM INTEGRATION OF OPC
- POWERFUL COMMUNICATION VIA WEB SERVICES
- UNIFIED DATA MODEL
- SUPPORT COMPLEX DATA STRUCTURES
- PROCESS DATA COMMUNICATION WITHOUT DATA LOSS
- SUPPORT OF METHOD CALLS

THE OPC „UA VISION“



Enterprise

Servers

Workstations

Mobile

Controllers

Embedded

Unified Architecture

UA can combine any and all communication layers in your company

THE GOALS OF OPC UA

- „KEEP IT SIMPLE“
- „EVOLUTION“ INSTEAD OF „REVOLUTION“
- PLATFORM INDEPENDENCE AND SCALABILITY
- USER CONTROL
- DATA SECURITY
- PERFORMANCE

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
AddressSpace	The collection of information that an OPC UA *Server* makes visible to its *Clients*. See Part 3 for a description of the contents and structure of the *Server AddressSpace*.
Alarm	A type of *Event* associated with a state condition that typically requires acknowledgement. See Part 9 for a description of *Alarms*.
Attribute	A primitive characteristic of a *Node*. All *Attributes* are defined by OPC UA, and may not be defined by *Clients* or *Servers*. *Attributes* are the only elements in the *AddressSpace* permitted to have data values.
Certificate	A digitally signed data structure that describes capabilities of a *Client* or *Server*.
Client	A software application that sends *Messages* to OPC UA *Servers* conforming to the *Services* specified in this set of specifications.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
Condition	A generic term that is an extension to an *Event*. A *Condition* represents the conditions of a system or one of its components and always exists in some state.
Communication Stack	A layered set of software modules between the application and the hardware that provides various functions to encode, encrypt and format a *Message* for sending, and to decode, decrypt and unpack a *Message* that was received.
Complex Data	Data that is composed of elements or more than one primitive data type, such as a structure.
Discovery	The process by which OPC UA Clients obtain information about OPC UA Servers, including endpoint and security information.
Event	A generic term used to describe an occurrence of some significance within a system or system component.
Event	A special *Attribute* of a *Node* that signifies that a *Client* may subscribe to that particular *Node* to receive *Notifications* of *Event* occurrences.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
Information Model	An organizational framework that defines, characterizes and relates information resources of a given system or set of systems. The core address space model supports the representation of *Information Models* in the *AddressSpace*. See Part 5 for a description of the base OPC UA *Information Model*.
Information Model	The data unit conveyed between *Client* and *Server* that represents a specific *Service* request or response.
Method	A callable software function that is a component of an *Object*.
MonitoredItem	A *Client*-defined entity in the *Server* used to monitor *Attributes* or *EventNotifiers* for new values or *Event* occurrences and generate *Notifications* for them.
Node	The fundamental component of an *AddressSpace*.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
NodeClass	The class of a *Node* in an *AddressSpace*. *NodeClasses* define the metadata for the components of the OPC UA Object Model. They also define constructs, such as *Views,* that are used to organize the *AddressSpace*.
NotificationMessage	A *Message* published from a *Subscription* that contains one or more *Notifications*.
Object	A *Node* that represents a physical or abstract element of a system. *Objects* are modelled using the OPC UA Object Model. Systems, subsystems and devices are examples of *Objects*. An *Object* may be defined as an instance of an *ObjectType*.
Object Instance	A synonym for *Object*. Not all *Objects* are defined by *ObjectTypes*.
ObjectType	A *Node* that represents the type definition for an *Object*.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
Profile	A specific set of capabilities, defined in Part 7, to which a *Server* may claim conformance. Each *Server* may claim conformance to more than one *Profile*.
Program	An executable *Object* that, when invoked, immediately returns a response to indicate that execution has started, and then returns intermediate and final results through *Subscriptions* identified by the *Client* during invocation.
Reference	An explicit relationship (a named pointer) from one *Node* to another. The *Node* that contains the *Reference* is the source *Node*, and the referenced *Node* is the target *Node*. All *References* are defined by *ReferenceTypes*.
ReferenceType	A *Node* that represents the type definition of a *Reference*. The *ReferenceType* specifies the semantics of a *Reference*. The name of a *ReferenceType* identifies how source *Nodes* are related to target *Nodes* and generally reflects an operation between the two, such as “A *Contains* B”.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
RootNode	The beginning or top *Node* of a hierarchy. The *RootNode* of the OPC UA *AddressSpace* is defined in Part 5.
Server	A software application that implements and exposes the *Services* specified in this set of specifications.
Service	A *Client*-callable operation in an OPC UA *Server*. *Services* are defined in Part 4. A *Service* is similar to a method call in a programming language or an operation in a Web services WSDL contract.
Service Set	A group of related *Services*.
Session	A logical long-running connection between a *Client* and a *Server*. A *Session* maintains state information between *Service* calls from the *Client* to the *Server*.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Term	Description
Subscription	A *Client*-defined endpoint in the *Server,* used to return *Notifications* to the *Client*. Generic term that describes a set of *Nodes* selected by the *Client* (1) that the *Server* periodically monitors for the existence of some condition, and (2) for which the *Server* sends *Notifications* to the *Client* when the condition is detected.
Variable	A *Variable* is a *Node* that contains a value.
View	A specific subset of the *AddressSpace* that is of interest to the *Client*.

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Abbreviation	Description
A&E	Alarms and Events
API	Application Programming Interface
COM	Component Object Model
DA	Data Access
DCS	Distributed Control System
DX	Data Exchange
HDA	Historical Data Access
HMI	Human-Machine Interface
LDAP	Lightweight Directory Access Protocol
MES	Manufacturing Execution System
OPC	Manufacturing Execution System
PLC	Programmable Logic Controller
SCADA	Supervisory Control And Data Acquisition

THE OPC UA SPECIFICATION

TERMS, DEFINITIONS AND ABBREVIATIONS

Abbreviation	Description
SOAP	Simple Object Access Protocol
UA	Unified Architecture
UDDI	Universal Description, Discovery and Integration
UDDI	Unified Modelling Language
WSDL	Web Services Definition Language
XML	Extensible Mark-up Language

THE OPC UA SPECIFICATION

- Specification consists of 14 parts with over 700 pages
- The different parts of the specification are grouped in three areas:
  - **Core Specification Parts**
    These core capabilities define the structure of the OPC AddressSpace and the Services that operate on it.
  - **Access Type Specification Parts**
    Apply the core capabilities to specific types of access previously addressed by separate OPC COM specifications, such as Data Access (DA), Alarms and Events (A&E) and Historical Data Access (HDA).
  - **Utility Specification Parts**
    Describes the Discovery mechanisms for OPC UA and ways of aggregating data.

THE OPC UA SPECIFICATION

OPC UA CORE SPECIFICATION

- **PART 1: OVERVIEW AND CONCEPTS**
  PRESENTS THE CONCEPTS AND OVERVIEW OF OPC UA.
- **PART 2: SECURITY MODEL**
  DESCRIBES THE MODEL FOR SECURING INTERACTIONS BETWEEN OPC UA *CLIENTS* AND OPC UA *SERVERS*.
- **PART 3: ADDRESS SPACE MODEL**
  DESCRIBES THE CONTENTS AND STRUCTURE OF THE *SERVER'S* *ADDRESSSPACE*.
- **PART 4: SERVICES**
  SPECIFIES THE *SERVICES* PROVIDED BY OPC UA *SERVERS*.

THE OPC UA SPECIFICATION

OPC UA CORE SPECIFICATION

- **Part 5: Information Model**
  Specifies the types and their relationships defined for OPC UA *Servers*.
- **Part 6: Mappings**
  Specifies the mappings to transport protocols and data encodings supported by OPC UA.
- **Part 7: Profiles**
  Specifies the *Profiles* that are available for OPC *Clients* and *Servers*. These *Profiles* provide groups of *Services* or functionality that can be used for conformance level certification. *Servers* and *Clients* will be tested against the *Profiles*.

THE OPC UA SPECIFICATION

OPC UA ACCESS TYPE SPECIFICATION

- **Part 8: Data Access**
  Specifies the use of OPC UA for data access.
- **Part 9: Alarms and Conditions**
  Specifies the use of OPC UA for access to *Alarms* and *Conditions*. Extends the simple events to include support for *Alarms* and *Conditions*.
- **Part 10: Programs**
  Specifies OPC UA support for access to *Programs*.
- **Part 11: Historical Access**
  Specifies use of OPC UA for historical access. This access includes both historical data and historical *Events*.

THE OPC UA SPECIFICATION

OPC UA ACCESS TYPE SPECIFICATION

- **Part 12: Discovery and Global Services**
  Specifies the interaction with the Discovery Servers.
- **Part 13: Aggregates**
  Specifies the information model associated with Aggregates.
- **Part 14: PubSub**
  Specifies the OPC UA publish subscribe pattern (PubSub) communication model.

THE OPC UA SPECIFICATION

OPC UA CORE SPECIFICATION

Part	Description	Version	Date	State
1	Overview and Concepts	1.04	22.-NOV-2017	Released
2	Security Model	1.04	03-AUG-2018	Released
3	Address Space Model	1.04	22-NOV-2017	Released
4	Services	1.04	22-NOV-2017	Released
5	Information Model	1.04	22-NOV-2017	Released
6	Mappings	1.04	22-NOV-2017	Released
7	Profiles	1.04	01-NOV-2017	Released
8	Data Access	1.04	01-NOV-2017	Released
9	Alarms and Conditions	1.04	22-NOV-2017	Released
10	Programs	1.04	01-NOV-2017	Released
11	Historical Access	1.04	09-JAN-2018	Released
12	Discovery	1.04	07-FEB-2018	Released
13	Aggregates	1.04	22-NOV-2017	Released
14	PubSub	1.04	06-FEB-2018	Released

THE OPC UA SPECIFICATION

OPC UA SPECIFICATION DOWNLOAD

- **OPCFOUNDATION.ORG**
  SPECIFICATIONS ARE NOW AVAILABLE FOR DOWNLOAD FOR EVERYONE (JUST REGISTER) AT

  HTTPS://OPCFOUNDATION.ORG/DEVELOPER-TOOLS/SPECIFICATIONS-UNIFIED-ARCHITECTURE

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- Each OPC Classic Interface has dozens of methods
- 35 UA "Services" grouped in service sets

Discovery *FindServers* *GetEndpoints* *RegisterServer*	**SecureChannel** *CreateSession* *ActivateSession* *CloseSession* *Cancel*	**NodeManagement** *AddNodes* *AddReferences* *DeleteNodes* *DeleteReferences*
View *Browse* *BrowseNext* *TranslateBrowse…* *RegisterNodes* *UnregisterNodes*	**Query** *QueryFirst* *QueryNext*	**Attribute** *Read* *HistoryRead* *Write* *HistoryUpdate*
Method *Call*	**MonitoredItem** *CreateMonitoredItems* *ModifyMonitoredItems* *SetMonitoringMode* *SetTriggering* *DeleteMonitoredItems*	**Subscription** *CreateSubscription* *ModifySubscription* *SetPublishingMode* *Publish* *Republish* *TransferSubscriptions* *DeleteSubscriptions*

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- A few issues with Classic OPC
  - Classic OPC specifications are independent specifications and separated.
  - Each specification requires its own implementation
  - Higher Development efforts, e.g. the «Browser»-Interface for DA is different/separated from the implementation in HDA
- Unification of all Classic OPC specifications
  - One "Node" is used by all specifications
  - Implementation of ONE Interface
  - UA includes all Classic OPC specifications and more …

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

Classic OPC "Spec."	OPC UA "Profile"
Alarms & Events	Alarms & Conditions
Batch	n/a
Commands	Programs
Complex Data	n/a
Data Access	Data Access
Data eXchange	n/a
Historical Data Access	Historical Access
Security	n/a
XML Data Access	n/a

N/A = NOT APPLICABLE

➢ **FUNCTIONALITY IS ALREADY INCLUDED IN THE CORE OF OPC UA**

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- Incorporated transport-level security
- Tools for authentication are directly build in
- Incorporated protection against "Message Spoofing", "Injection" and "Replay"
- Incorporated protection against loss of messages

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- Creating and Using of structured data
- Define your own object types and instances, from simple to complex
- Each OPC UA application can use any data types defined in the UA model

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- No longer only support of Windows!
- Can be ported to any operating system (ANSI C or Java)
- Can be used on embedded devices
- Connects applications for data capture directly to a device

ADVANTAGES OF OPC UA

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- ■ Architecture with multiple layers
- ■ "Upgrade" a layer is possible without influencing other layers
- ■ New technologies can be added to OPC UA without problems

COMPATIBILITY

Simpler Interface

Combines Classic OPC Spec.

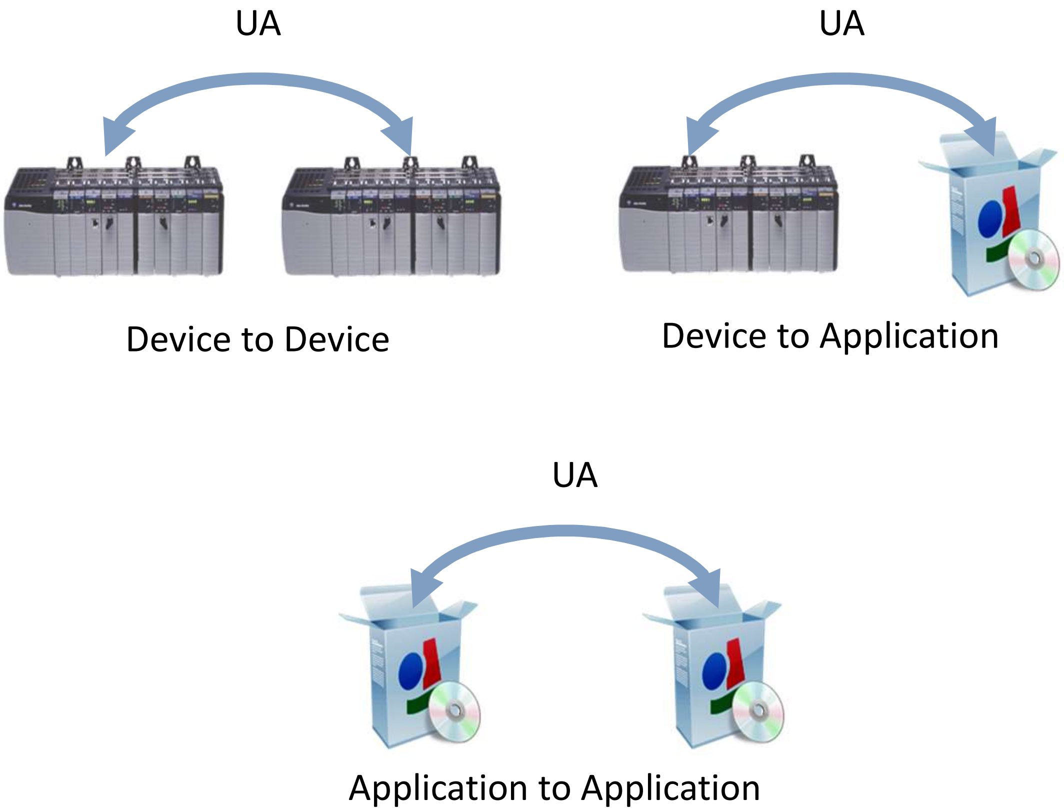
Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility



COMPATIBILITY

Simpler Interface

Combines Classic OPC Spec.

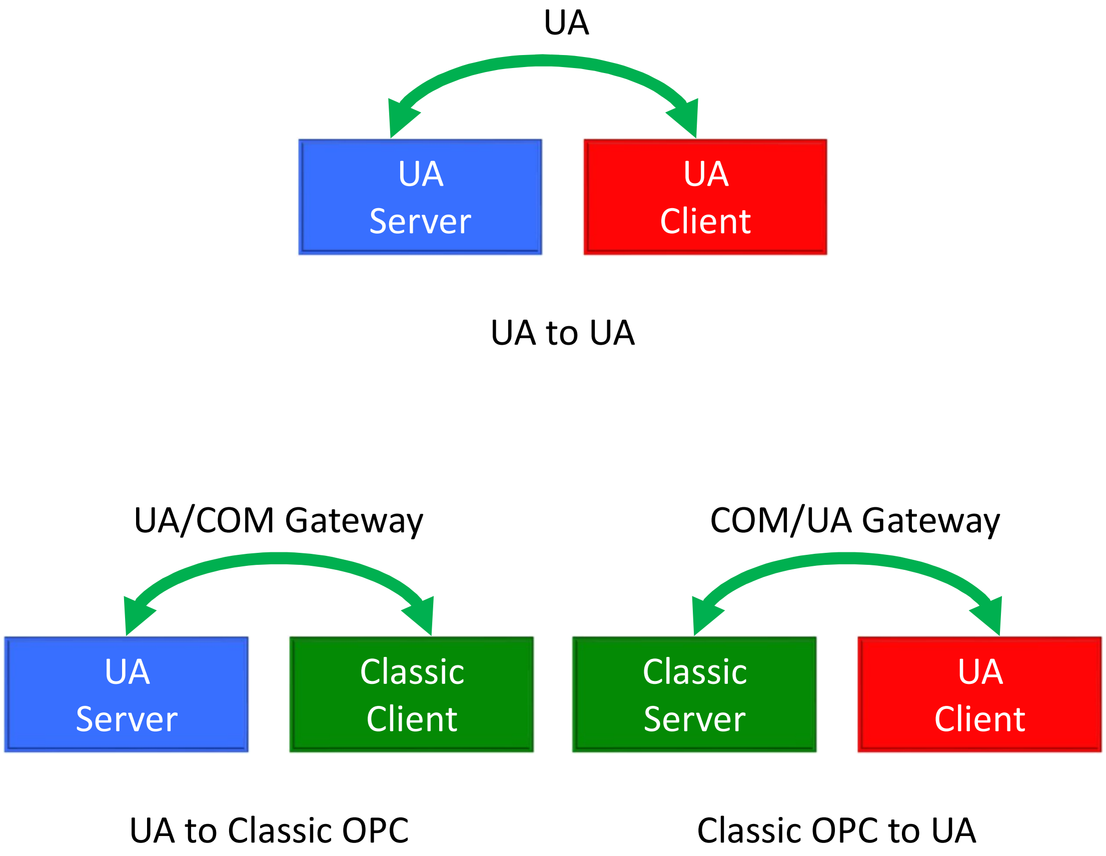
Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility



UA

UA Server

UA Client

UA to UA

UA/COM Gateway

UA Server

Classic Client

UA to Classic OPC

COM/UA Gateway

Classic Server

UA Client

Classic OPC to UA

COMPATIBILITY

DELIVERY FROM OPC FOUNDATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- Specification
  - Part 1 to 13: available free for Download
  - Available for Registered Users

- Redistributables
  - Local Discovery Server (LDS) for Windows
    Allows detection of available OPC UA servers on a system
  - Sample Applications for Windows
    A suite of OPC UA Clients and Servers that demonstrate OPC UA technology and its most popular functionality. All examples are ready to run without any configuration.
  - Available for Registered Users

COMPATIBILITY

DELIVERY FROM OPC FOUNDATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

■ OPC UA Stacks for Developers

- Are the base for development of OPC UA Server and OPC UA Client applications. Available as:
  - .NET Standard Stack https://github.com/OPCFoundation/UA-.NETStandard
  - ANSI C Stack (Legacy) https://github.com/OPCFoundation/UA-AnsiC-Legacy
  - .NET Stack (Legacy) https://github.com/OPCFoundation/UA-.NET-Legacy
  - Java Stack (Legacy) https://github.com/OPCFoundation/UA-Java-Legacy
- Available via GitHub
- Sample Code Only, not full SDKs

COMPATIBILITY

DELIVERY FROM OPC FOUNDATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- ■ Local Discovery Server Source Code
  - Can be used as base for porting to Linux for example.
  - Available via GitHub
    https://github.com/OPCFoundation/UA-LDS
- ■ Global Discovery Server (GDS) Sample
  - Provides the necessary infrastructure to provide enterprise-wide administration of OPC UA Servers.
  - Sample Code Only
  - Available for registered users

COMPATIBILITY
CERTIFICATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- Compliance Test Tool
  - Compliance Test Tool Windows and Linux for Corporate Members
  - A single test tool for testing UA Server and UA Clients
- Interoperability Workshop
  - There are always 3 Interoperability Workshops (IOP-Workshops) per year:
    - USA
    - Europe
    - Japan
  - Test of Client- and Server applications from different vendors to check compatibility between members

COMPATIBILITY

CERTIFICATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

■ OPC Foundation Certification Test Lab

- The OPC Foundation's Certification and Compliance program exists to help members develop and provide high quality products that meet minimum operability requirements.
- OPC Certified products are:
  - **Compliant** with the OPC specifications
  - **Interoperable** with other OPC products from other vendors
  - **Robust**, reliable and able to recover from lost communications, etc.
  - **Usable**, by following universally accepted best-practices
  - **Efficient** in managing resources (CPU, memory, disk space etc.)

COMPATIBILITY
CERTIFICATION

Simpler Interface

Combines Classic OPC Spec.

Security included

Information Modeling

Plattform Independancy

Scalable Architectur

Compatibility

- What is Certifiable?
  - Any OPC UA Client or Server can be certified by a Test Lab.
  - Software Development Kits or "toolkits" cannot be certified directly.
  - Reference implementations (Sample Clients and/or Sample Servers) shipped with the SDK are tested and Certified in a Lab.
  - Applications developed with an SDK that has a certified reference implementation stand a much higher chance of certification, but require their own certification testing in order to become officially certified products.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

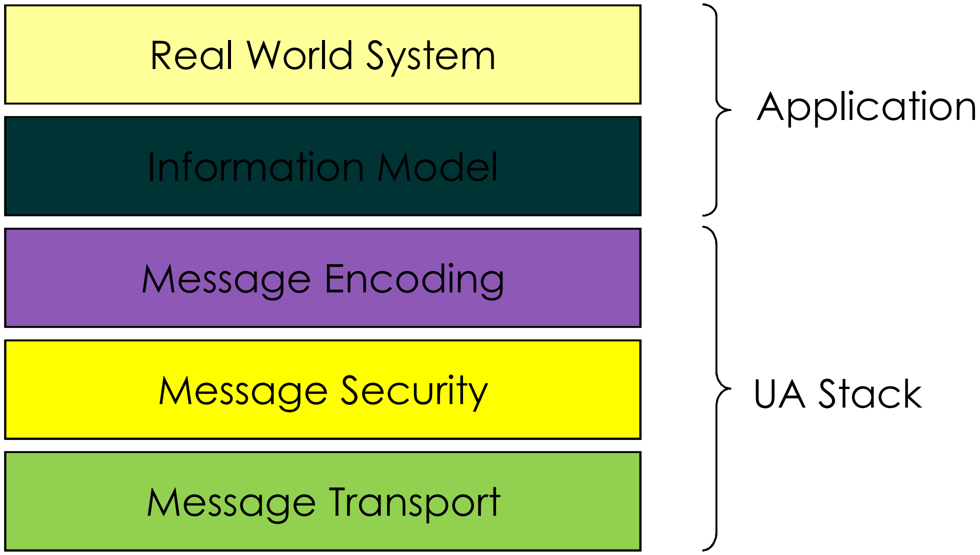
Data Types

Subscriptions

Services

Profiles

- All OPC UA applications use an OPC UA stack from the OPC Foundation!
- The stack includes the UA Services
- The stack handles the connections



Real World System

Information Model

Message Encoding

Message Security

Message Transport

Application

UA Stack

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

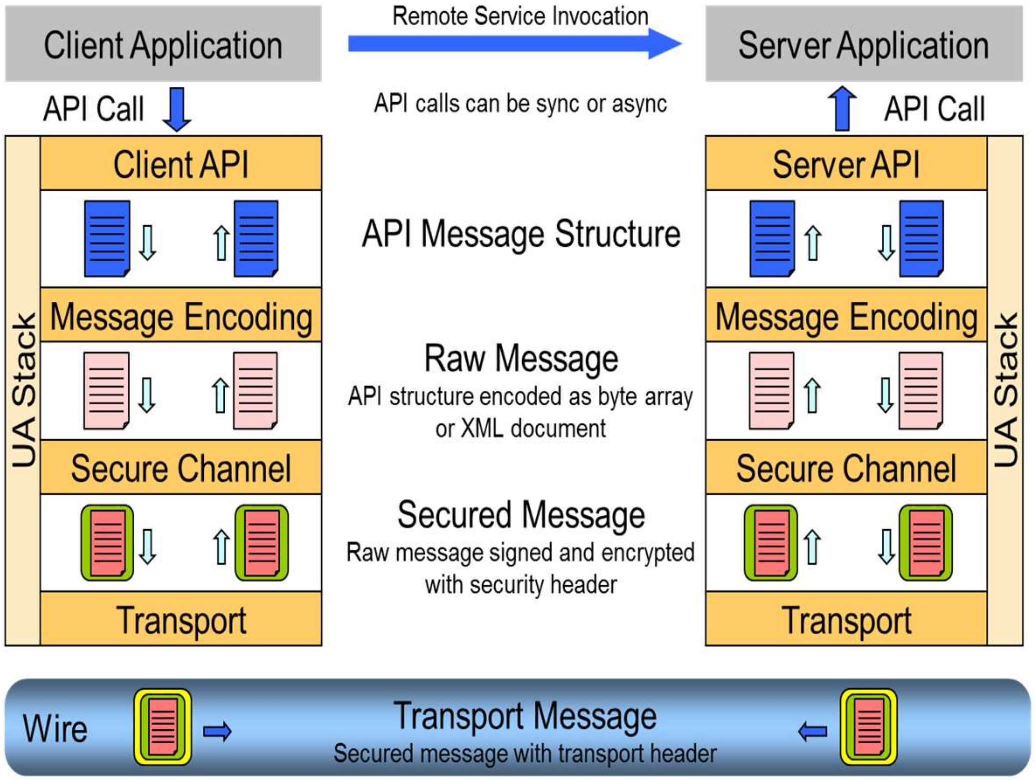
Data Types

Subscriptions

Services

Profiles

- Handling of the message encoding & transport



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

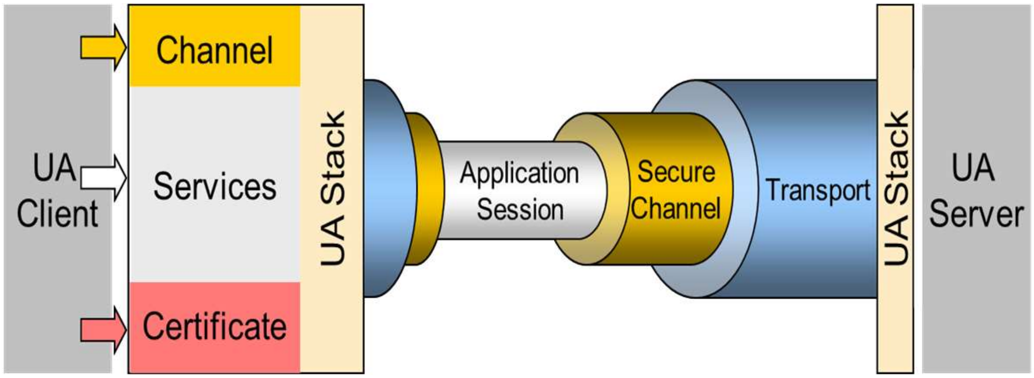
Data Types

Subscriptions

Services

Profiles

- Open a "Channel " to the server:
  - Protocol: SOAP/HTTP or UA Binary
  - Security: Encoding, Options
- Open a "Session" to the Channel:
  - User authentication and settings
  - All service-calls are done via a session



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

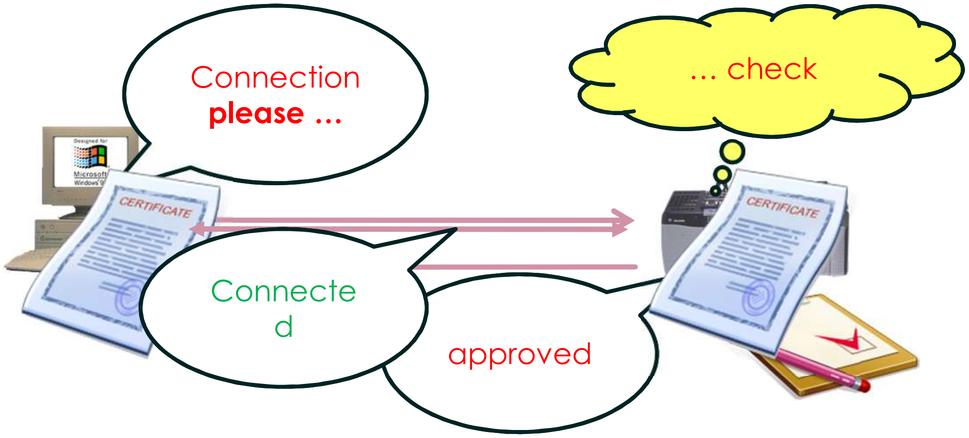
Data Types

Subscriptions

Services

Profiles

- ■ CLIENT PROVIDES A SOFTWARE CERTIFICATE WHICH THE SERVER CAN CHECK
- ■ SERVER PROVIDES A SOFTWARE CERTIFICATE WHICH THE CLIENT CAN CHECK
- ■ CLIENT AND SERVER CAN RESTRICT ACCESS OF APPLICATIONS WITH THESE CERTIFICATE FOR TRUSTABLE APPLICATIONS ONLY.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

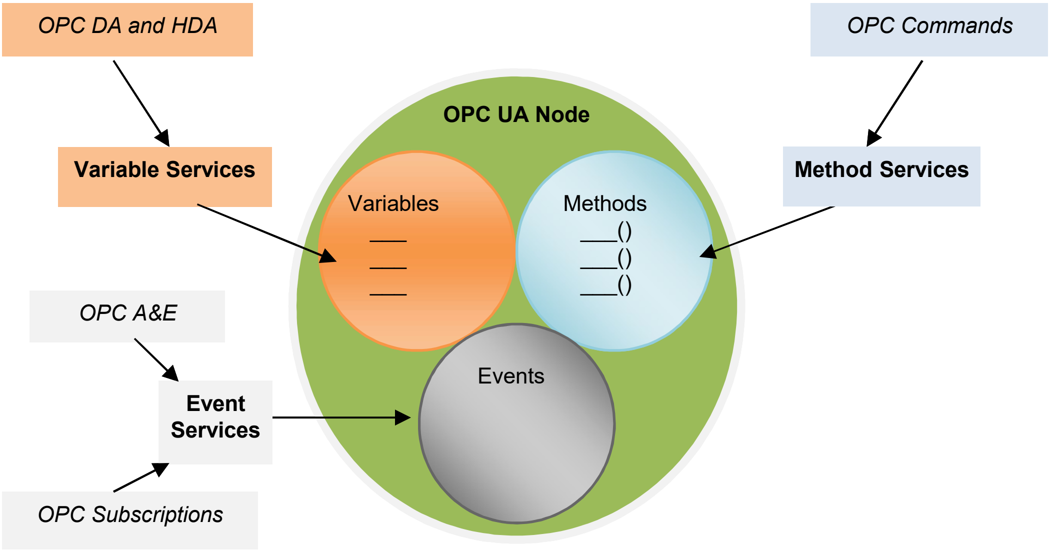
Data Types

Subscriptions

Services

Profiles

- ONE NODE FOR DATA, HISTORY, EVENTS AND METHODS ETC.
- THE FUNCTIONALITY OF NODES CAN BE EXTENDED!



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

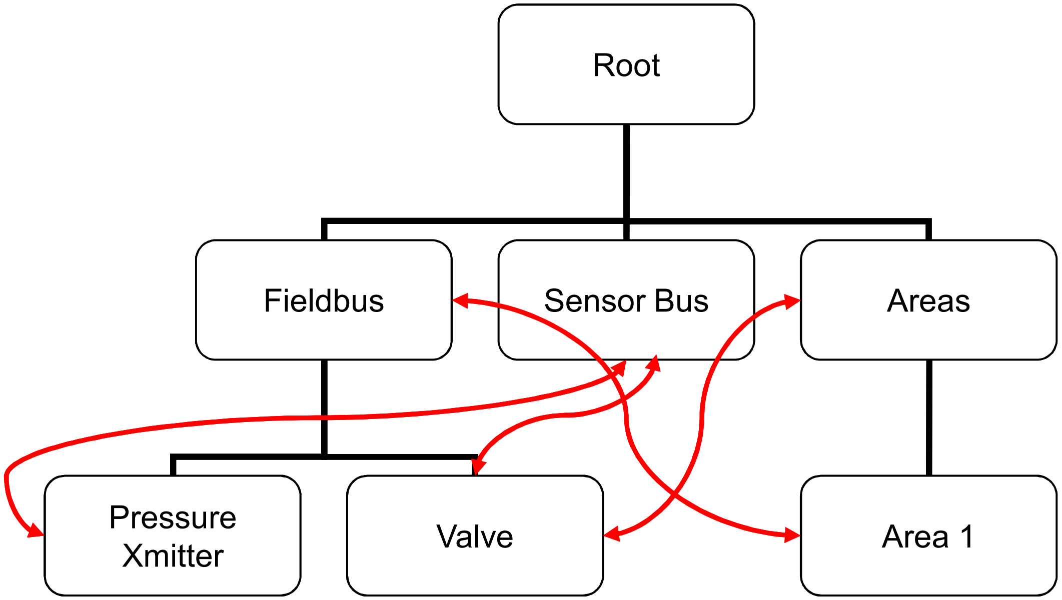
Data Types

Subscriptions

Services

Profiles

- NODES ARE ORGANIZED HIERARCHICALLY
- NODES CAN POINT TO OTHER NODES



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

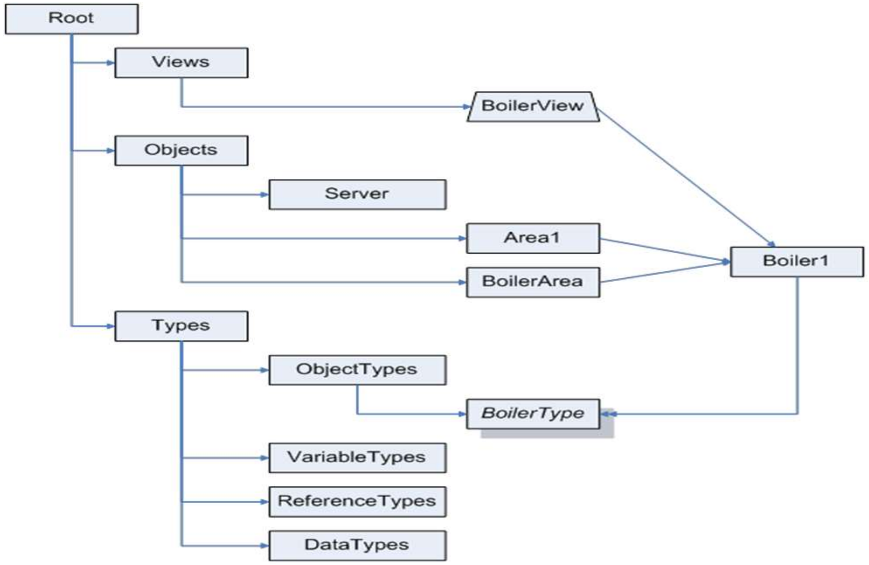
Data Types

Subscriptions

Services

Profiles

- Organization of the address space
- Views (like database views)
- Objects (the Nodes to work with)
- Types (from the Server supported data types)



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- A *View* is a subset of the *AddressSpace*.
- *Views* are used to restrict the *Nodes* that the *Server* makes visible to the *Client*, thus restricting the size of the *AddressSpace* for the *Service* requests submitted by the *Client*.
- The default *View* is the entire *AddressSpace*.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- *SERVERS* MAY OPTIONALLY DEFINE OTHER *VIEWS*.
- *VIEWS* HIDE SOME OF THE *NODES* OR *REFERENCES* IN THE *ADDRESSSPACE*.
- *VIEWS* ARE VISIBLE VIA THE *ADDRESSSPACE* AND *CLIENTS* ARE ABLE TO BROWSE *VIEWS* TO DETERMINE THEIR STRUCTURE.
- *VIEWS* ARE OFTEN HIERARCHIES, WHICH ARE EASIER FOR *CLIENTS* TO NAVIGATE AND REPRESENT IN A TREE.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

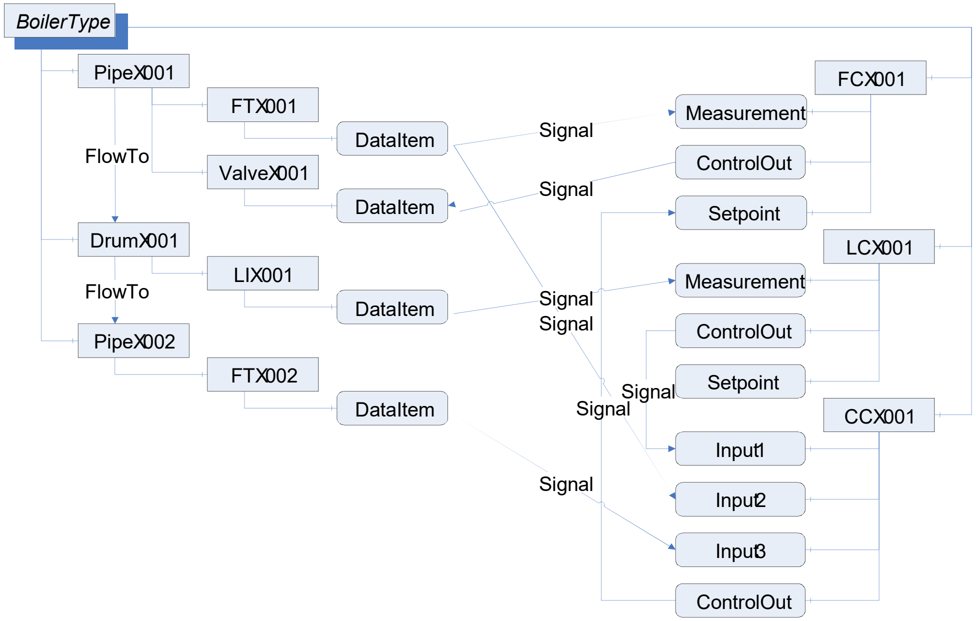
Data Types

Subscriptions

Services

Profiles

- Standard "scalar " data types
- Complex data types are possible
- Definition of object types and instances
- Definition of own data types!



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

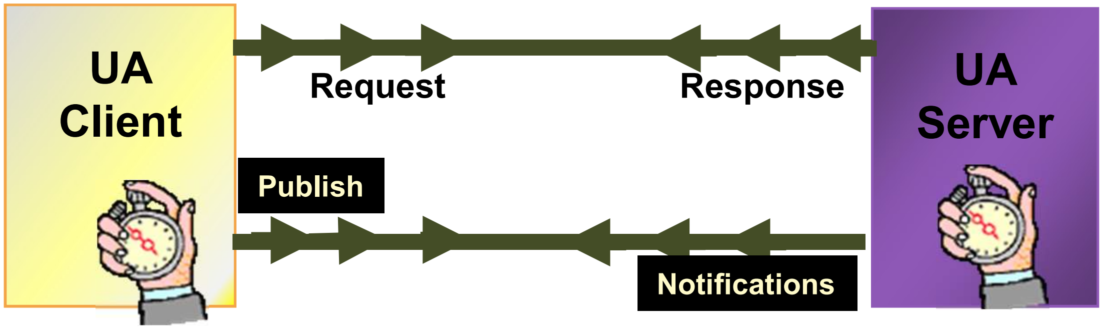
Data Types

Subscriptions

Services

Profiles

- Client defines data points (MonitoredItems) which the server should provide, by using settings like:
  - Time interval, Filter
  - Mode (Disabled, Reporting, Sampling)
  - Buffer size
- Server buffers the “Notifications“
- Client requests Server to send the buffered notifications



UA Client

Request

Response

UA Server

Publish

Notifications

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- All answers to a Publish Request are identified by a sequential ID.
- Clients confirms the ID with the next Publish Request
- Clients use the ID to detect a message loss.
- Clients can request lost Notifications with a RePublish Request.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- The OPC UA *Services* are the collection of abstract Remote Procedure Calls (RPC) that are implemented by OPC UA *Servers* and called by OPC UA *Clients*.
- All interactions between OPC UA *Clients* and *Servers* occur via these *Services*.
- The defined *Services* are considered abstract because no particular RPC mechanism for implementation is defined in this part.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- Part 6 specifies one or more concrete *Mappings* supported for implementation. For example, one mapping in Part 6 is to XML Web Services. In that case the *Services* described in this part appear as the Web service methods in the WSDL contract.
- Not all OPC UA *Servers* will need to implement all of the defined *Services*.
- *Profiles* dictate which Services need to be implemented in order to be compliant with a particular *Profile*.

TECHNO software
Innovative Software Solutions in it's Simplicity

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- *SERVICES* ARE ORGANIZED INTO *SERVICE SETS*.
- EACH *SERVICE SET* DEFINES A SET OF RELATED *SERVICES*.
- THE ORGANIZATION IN *SERVICE SETS* IS A LOGICAL GROUPING USED IN THIS STANDARD AND IS NOT USED IN THE IMPLEMENTATION.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

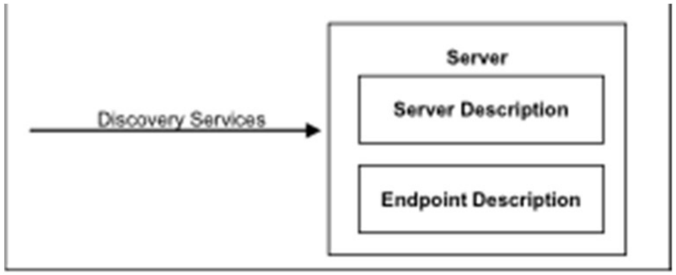
Data Types

Subscriptions

Services

Profiles

- THE *DISCOVERY SERVICE SET* DEFINES *SERVICES* THAT ALLOW A *CLIENT* TO DISCOVER THE *ENDPOINTS* IMPLEMENTED BY A *SERVER* AND TO READ THE SECURITY CONFIGURATION FOR EACH OF THOSE *ENDPOINTS*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

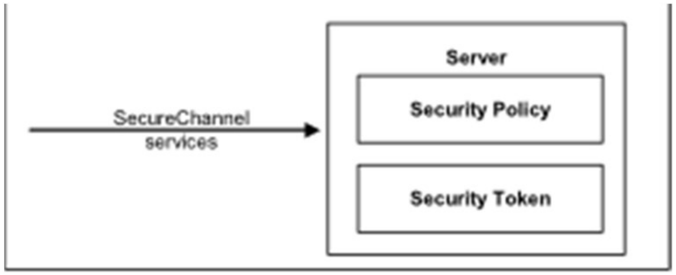
Data Types

Subscriptions

Services

Profiles

- THE *SECURECHANNEL SERVICE SET* DEFINES *SERVICES* THAT ALLOW A *CLIENT* TO ESTABLISH A COMMUNICATION CHANNEL TO ENSURE THE *CONFIDENTIALITY* AND *INTEGRITY* OF *MESSAGES* EXCHANGED WITH THE *SERVER*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

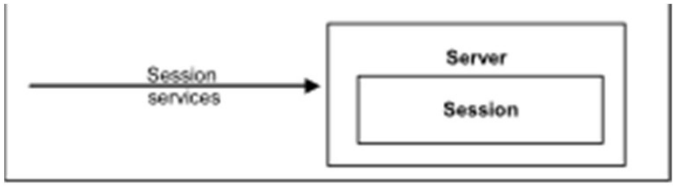
Data Types

Subscriptions

Services

Profiles

- THE *SESSION SERVICE SET* DEFINES *SERVICES* THAT ALLOW THE *CLIENT* TO AUTHENTICATE THE USER ON WHOSE BEHALF IT IS ACTING AND TO MANAGE *SESSIONS*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

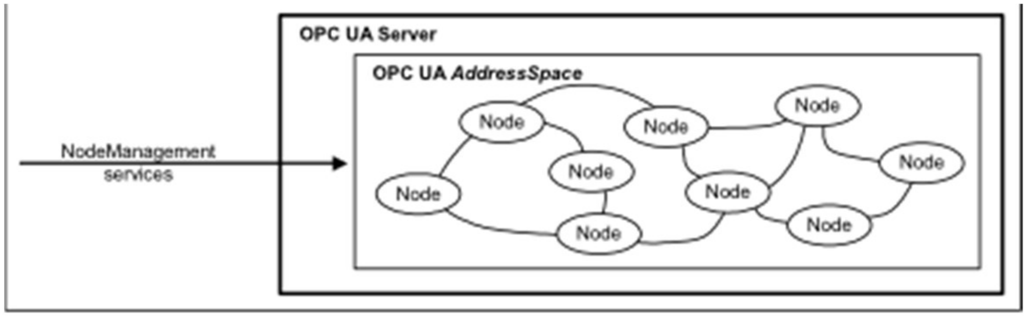
Data Types

Subscriptions

Services

Profiles

- THE *NODEMANAGEMENT SERVICE SET* DEFINES *SERVICES* THAT ALLOW THE *CLIENT* TO ADD, MODIFY AND DELETE *NODES* IN THE *ADDRESSSPACE*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

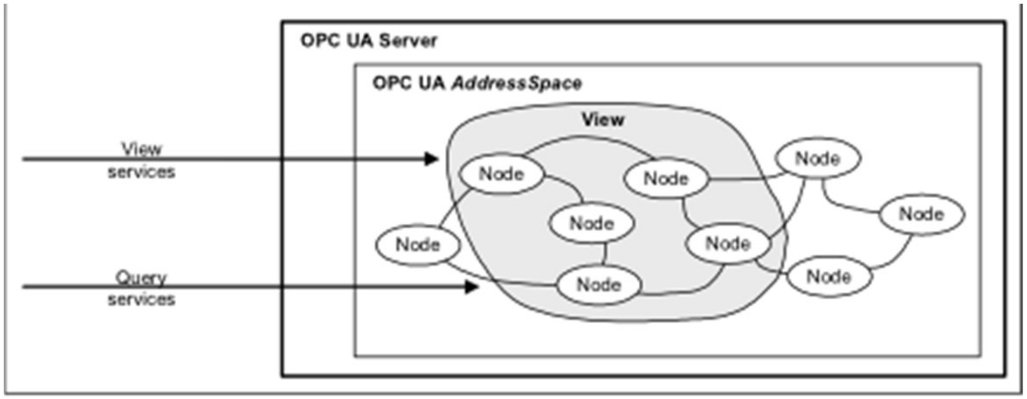
Data Types

Subscriptions

Services

Profiles

- THE *VIEW SERVICE SET* DEFINES *SERVICES* THAT ALLOW *CLIENTS* TO BROWSE THROUGH THE *ADDRESSSPACE* OR SUBSETS OF THE *ADDRESSSPACE* CALLED *VIEWS*. THE *QUERY SERVICE SET* ALLOWS *CLIENTS* TO GET A SUBSET OF DATA FROM THE *ADDRESSSPACE* OR THE *VIEW*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

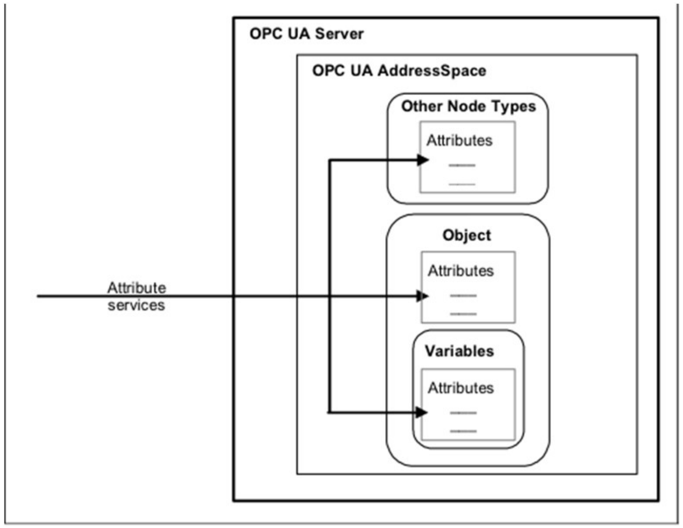
Data Types

Subscriptions

Services

Profiles

- The *Attribute Service Set* defines *Services* that allow *Clients* to read and write *Attributes* of *Nodes*, including their historical values. Since the value of a *Variable* is modelled as an *Attribute*, these *Services* allow *Clients* to read and write the values of *Variables*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

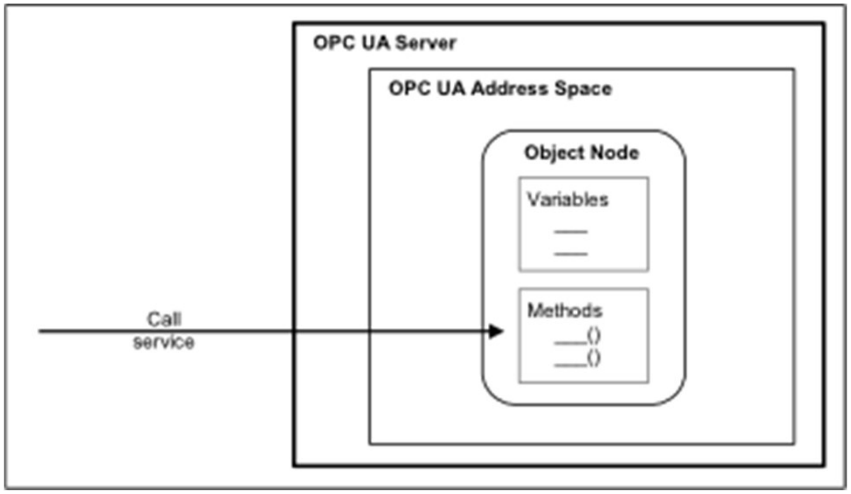
Data Types

Subscriptions

Services

Profiles

- THE *METHOD SERVICE SET* IS ILLUSTRATED IN . IT DEFINES *SERVICES* THAT ALLOW *CLIENTS* TO CALL METHODS. METHODS RUN TO COMPLETION WHEN CALLED. THEY MAY BE CALLED WITH METHOD-SPECIFIC INPUT PARAMETERS AND MAY RETURN METHOD-SPECIFIC OUTPUT PARAMETERS.



TECHNO software
Innovative Software Solutions in it's Simplicity

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

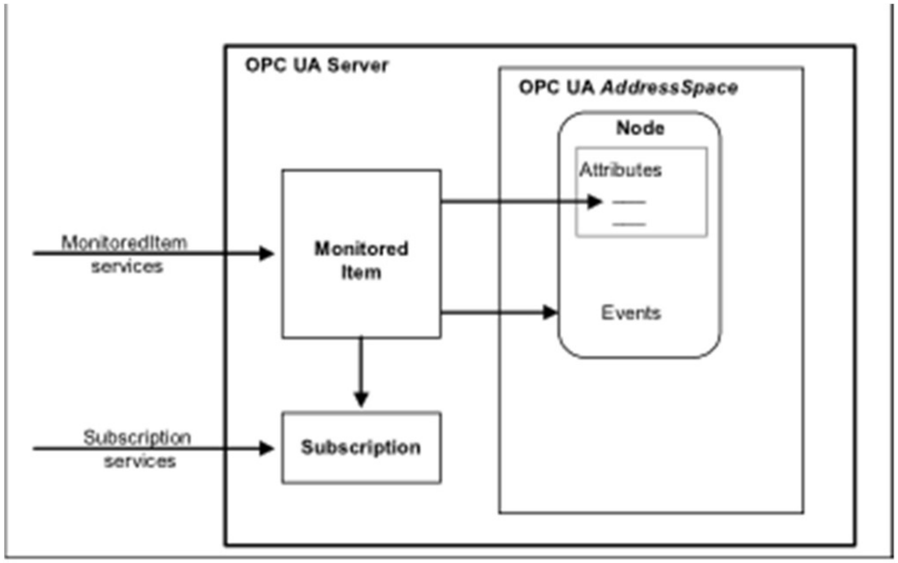
Data Types

Subscriptions

Services

Profiles

- THE *MONITOREDITEM SERVICE SET* AND THE *SUBSCRIPTION SERVICE SET* ARE USED TOGETHER TO SUBSCRIBE TO *NODES* IN THE OPC UA *ADDRESSSPACE*.



HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

Data Types

Subscriptions

Services

Profiles

- The *MonitoredItem Service Set* defines *Services* that allow *Clients* to create, modify, and delete *MonitoredItems* used to monitor *Attributes* for value changes and *Objects* for *Events*.
- These *Notifications* are queued for transfer to the *Client* by *Subscriptions*.
- The *Subscription Service Set* defines *Services* that allow *Clients* to create, modify and delete *Subscriptions*. *Subscriptions* send *Notifications* generated by *MonitoredItems* to the *Client*. *Subscription Services* also provide for *Client* recovery from missed *Messages* and communication failures.

HOW OPC UA WORKS

UA Stack

Connections

Nodes, Browse, Views

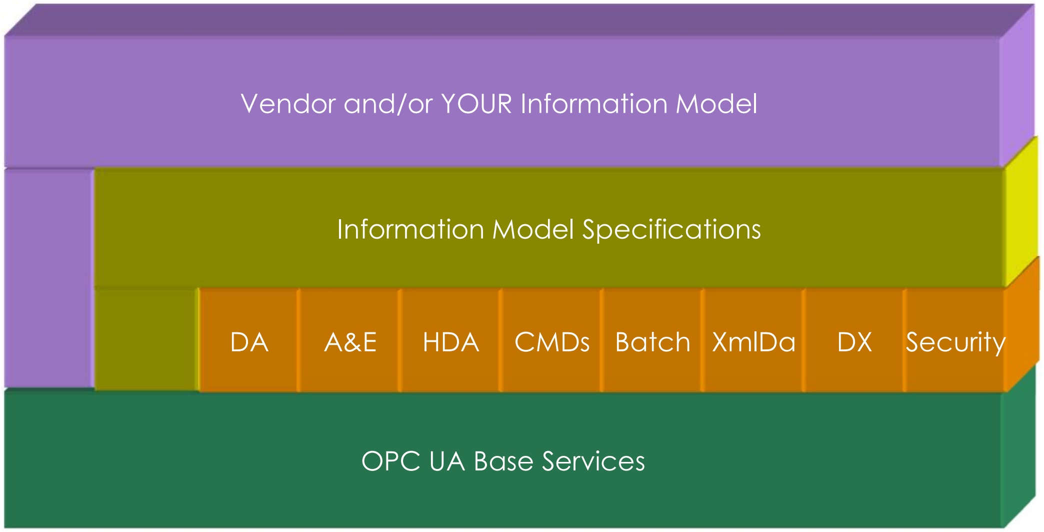
Data Types

Subscriptions

Services

Profiles

- "PROFILES" ARE LOGICAL GROUPED SERVICE CALLS AND DATA TYPES
- A "PROFILE" CORRESPONDS TO A CLASSIC OPC SPECIFICATION
- NEW "PROFILES" CAN BE ADDED OR EXISTING ONES CAN BE USED


Vendor and/or YOUR Information Model
Information Model Specifications
DA
A&E
HDA
CMDs
Batch
XmlDa
DX
Security
OPC UA Base Services