Process automation

The OPC standard

BS UNI studies, Fall semester 2024/2025

Octavian M. Machidon

octavian.machidon@fri.uni-lj.si

Introduction

- **Standard OPC**
  - Previously: Object Linking and Embedding for Process Control (OPC Classic)
  - Today: Open Platform Communication (OPC UA)
- The OPC standard ensures interoperability for secure and reliable data exchange in industrial automation. It is platform-independent and allows data flow between devices from different manufacturers.
- OPC Standard represents the most significant improvement in automation since the IEC 61131 standard (standardized programming languages).
- The development and maintenance of the standard is the responsibility of the OPC Foundation.

Introduction (cont.)

- **Communication Model:**
  - Client – Server
  - Publish - Subscribe (PubSub) in version 1.04
- **OPC Server**
  - Device specifics are hidden behind the OPC server, which:
    - Operates on the same or another device;
    - Is prepared by manufacturers of process devices or independent software companies.
  - An OPC server can simultaneously manage multiple devices of the same type.
  - Multiple OPC servers can run at the same time.
- **OPC Client**
  - Reading and writing process data.
  - Reading and acknowledging alarms.
  - Event monitoring.
  - Retrieving data from historical databases based on various criteria.

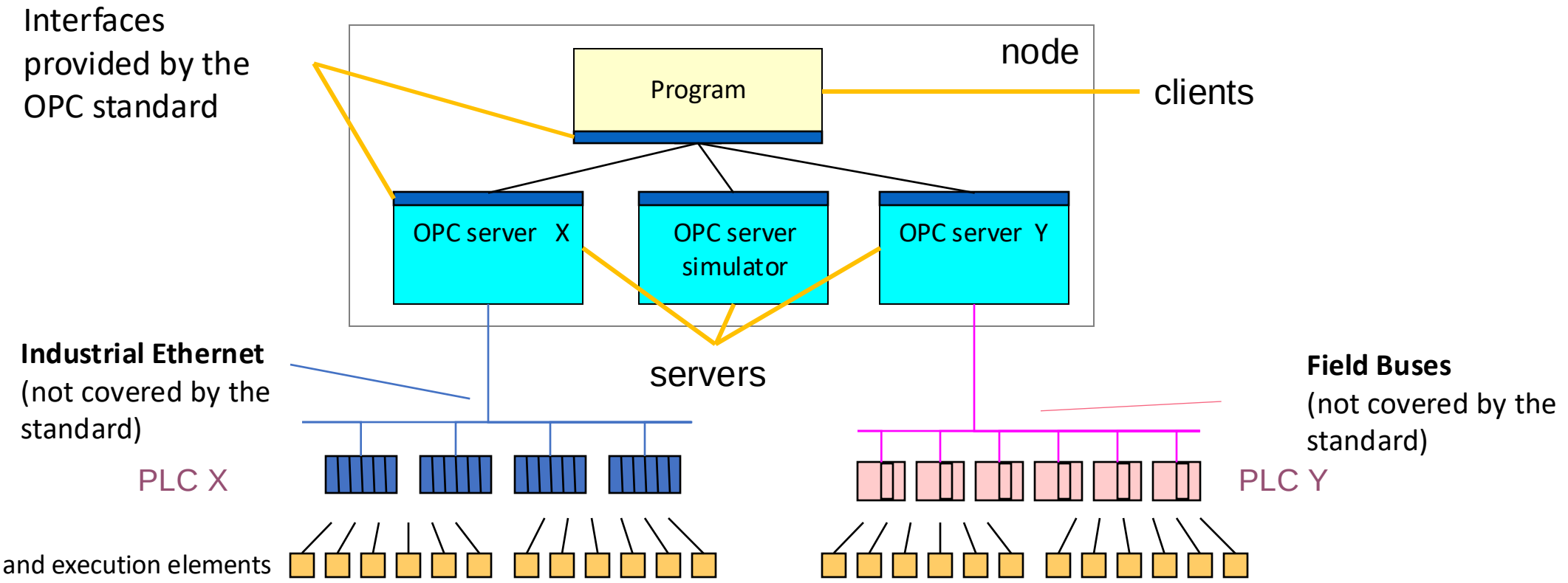
Introduction (cont.)

Benefits for the User

- Transparent access to relevant information:
  - Variables, data types, function blocks, structures, etc.
  - Independent of the system and network.
- Reduces integration time:
  - Controllers, HMI, SCADA, ERP, etc.

Introduction (cont.)

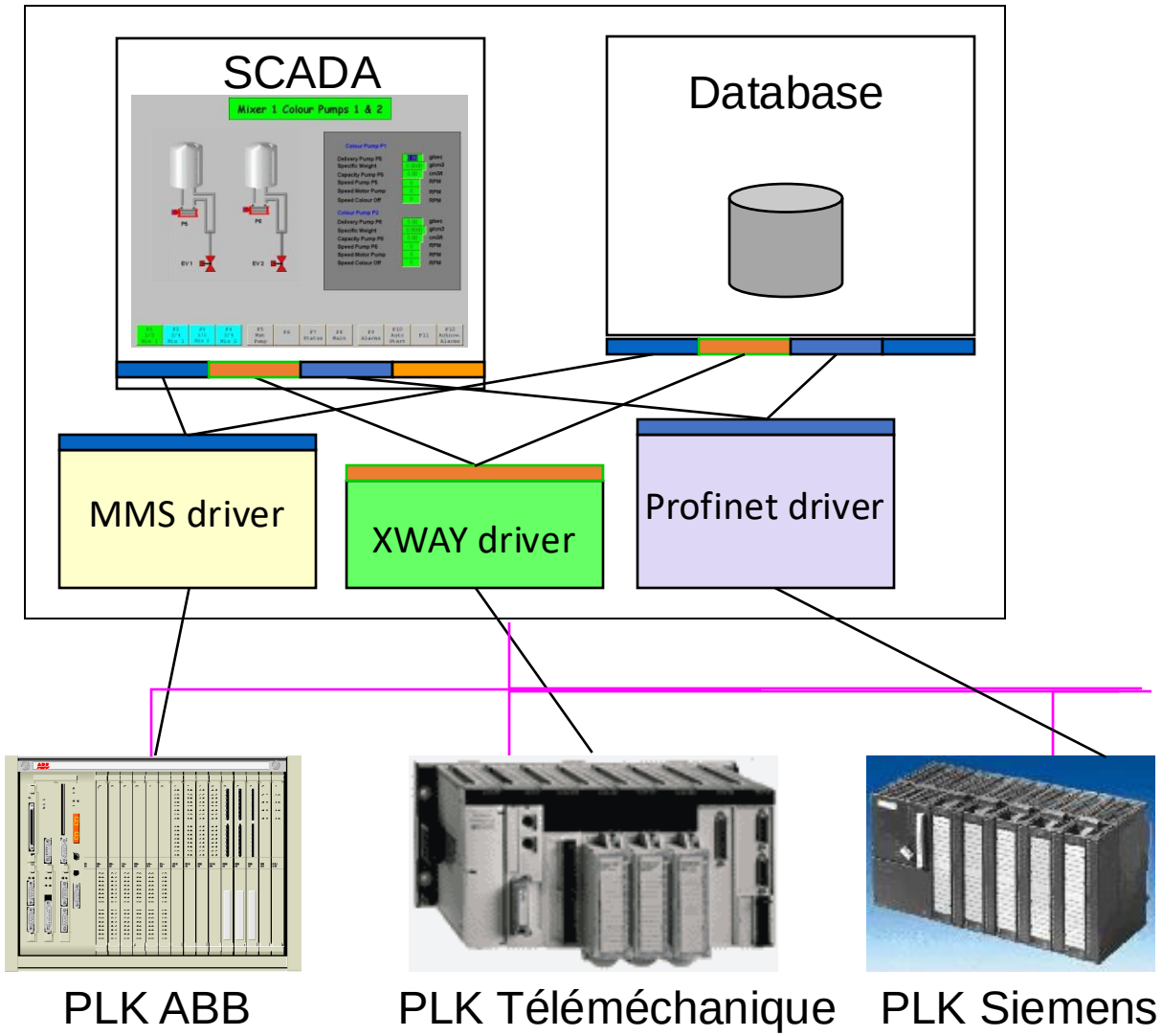
- The standard provides a programming interface (objects and methods) for servers and clients.
- The standard does not cover the connection between servers and automation devices.



Introduction (cont.)

Without the OPC Standard

- Every higher-level control system has a unique interface to access device drivers.

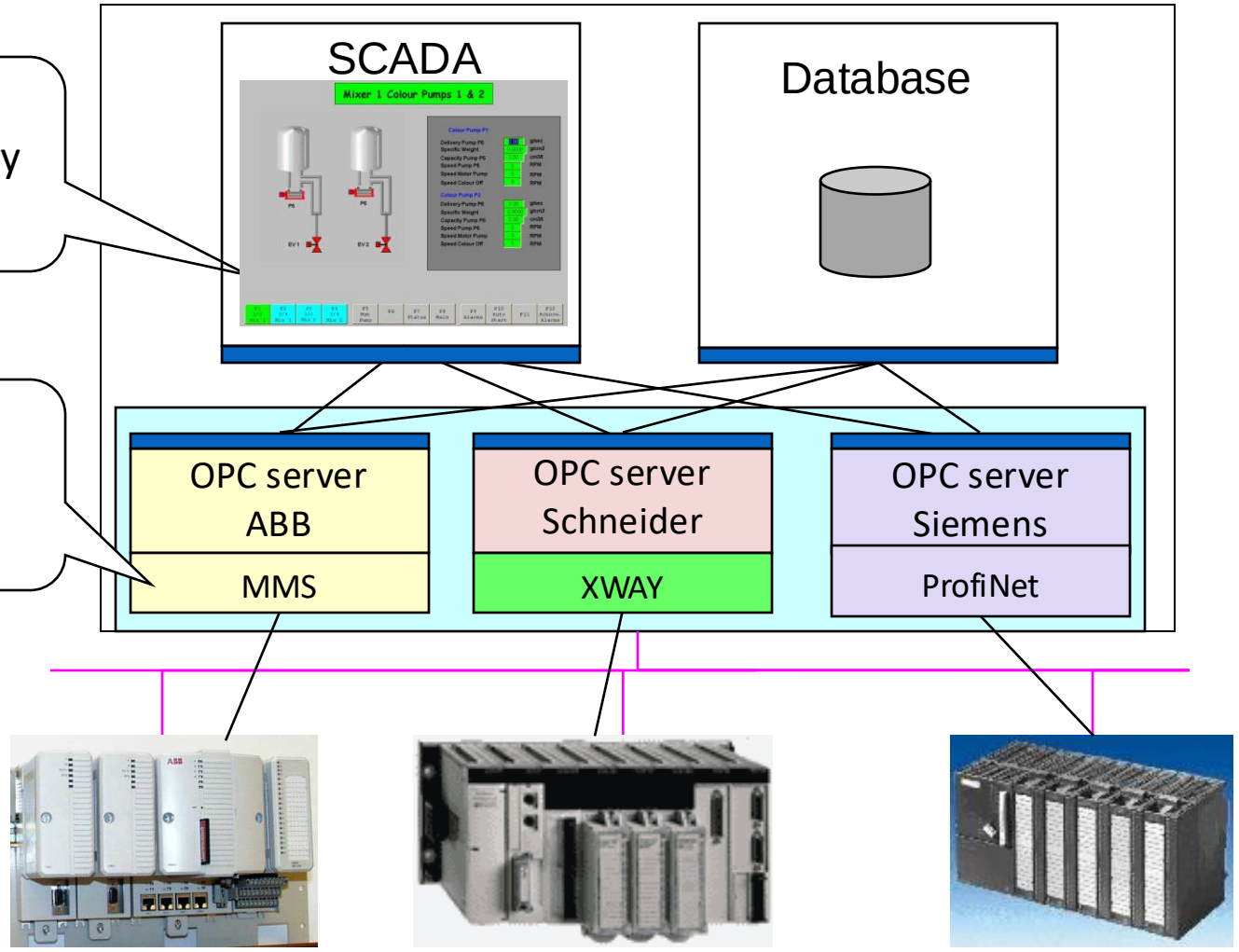


Introduction (cont.)

- Standard OPC

Application software does not require drivers for every type of PLC

Drivers still exist, but they are provided by the device manufacturers



Development

A brief history of the standard's development:

- **1995:** Automation equipment manufacturers (Fischer-Rosemount, Intellution, Opto 22, Rockwell Software) form a group to develop the standard.
- **1996:** The first version of OPC is released, now known as **OPC Classic**, based on Microsoft's technologies: Windows + (D)COM.
- **2004:** Microsoft releases Windows XP Service Pack 2, which significantly restricts the DCOM protocol and consequently limits the OPC standard.
- **2006:** The first version of the new architecture **OPC UA** (Unified Architecture) is released, emphasizing technology independence (OS, transport protocol, computing architecture) and security (certificates).
- **Recently:** In November 2022, the **OPC UA Field eXchange (UAFX)** specification is released, expanding the standard's use to the field-level domain of PLCs (presentation video).

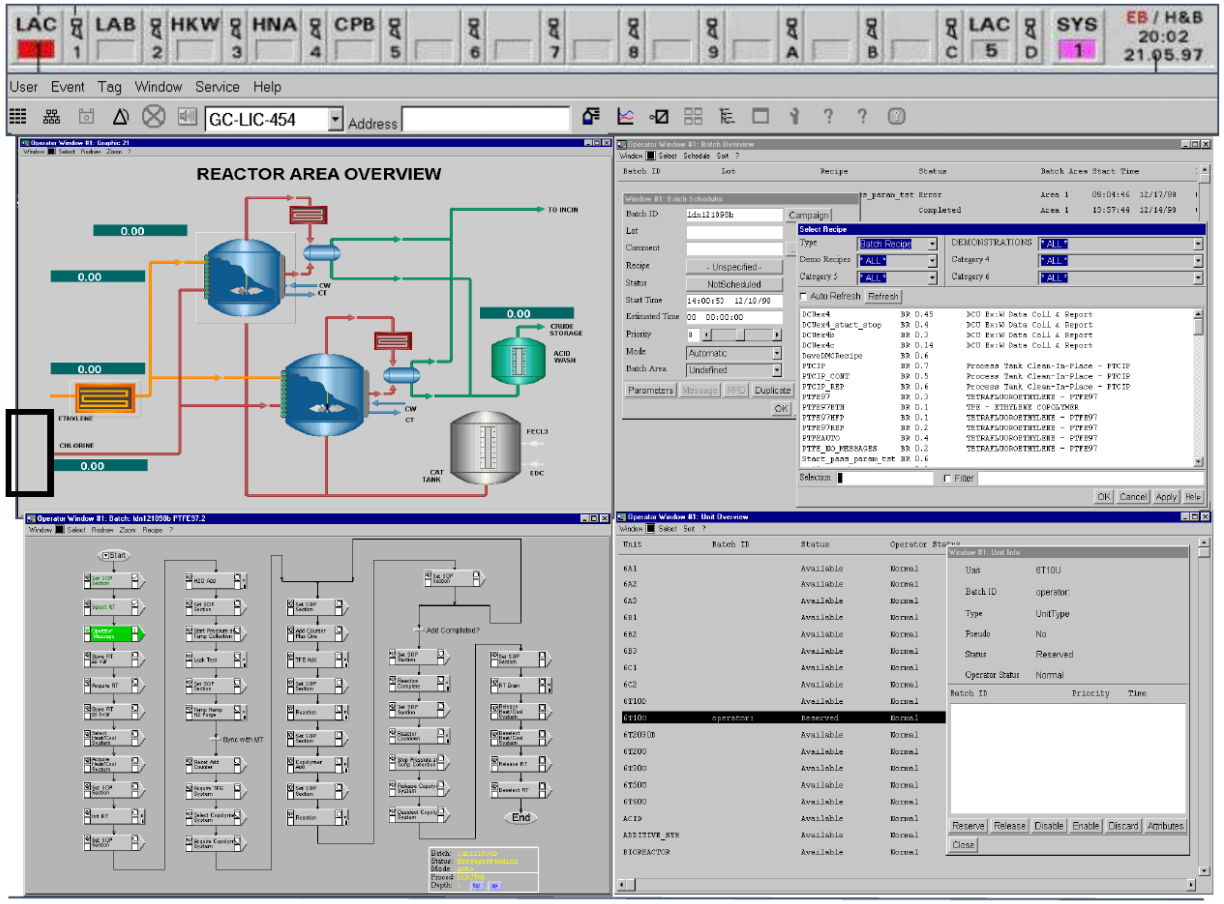
Technological building blocks of the OPC Classic standard (still quite prevalent):

- DA, AE, HDA, Batch, DX, XML-DA

OPC Classic - Components

OPC-DA (Data Access) for accessing data:

- Process variables (points) define the state of the system.
- Variables can be sent:
  - On change.
  - On demand.
  - At specific time intervals.
- The role of OPC-DA is to collect process variables.
- OPC-DA clients are primarily used for:
  - Visualization.
  - Soft control.
  - Integration with higher-level control systems.

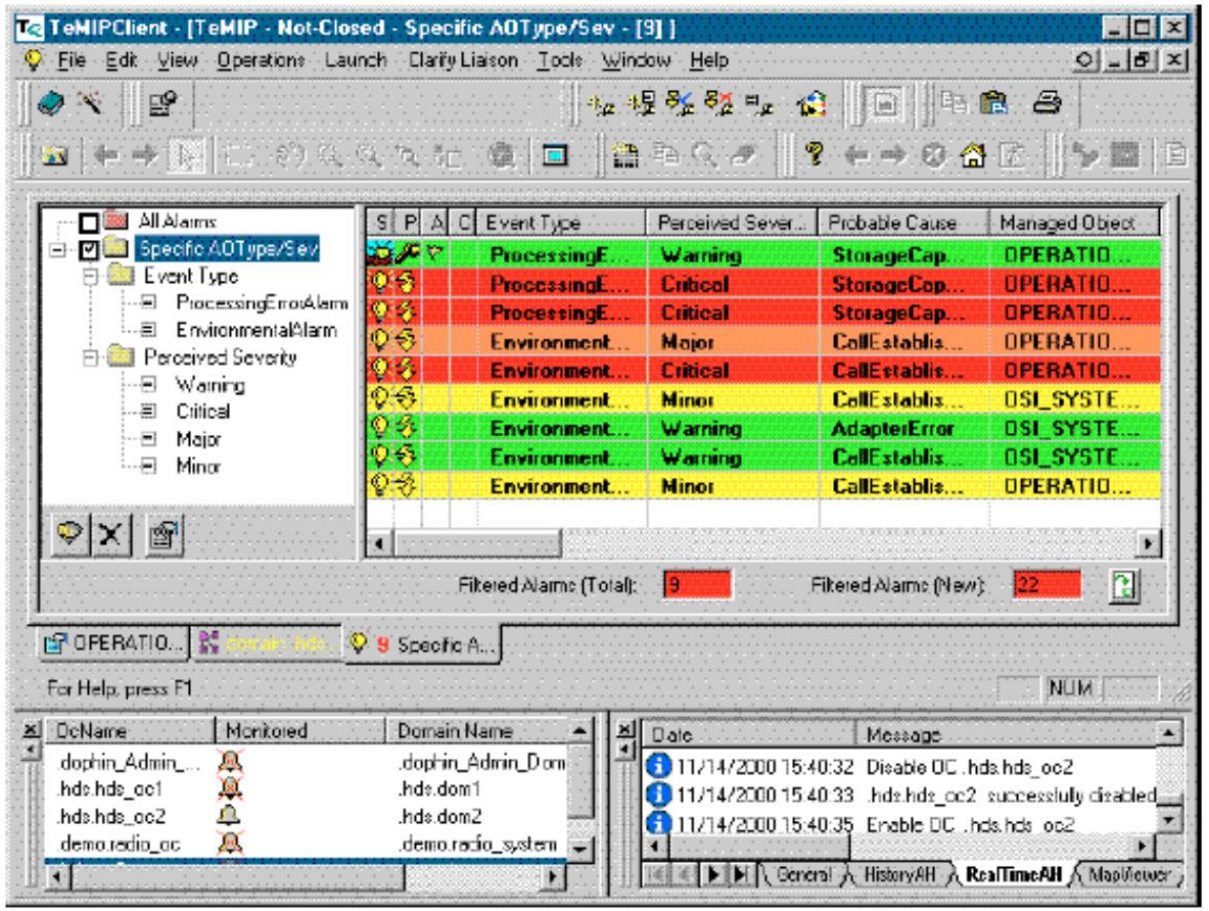


Note: OPC Classic is outdated/obsolete/historical.

OPC Classic – Components (cont.)

OPC-AE (Alarms and Events) for alarms and events:

- **Alarms** are abnormal system conditions requiring operator attention.
  - Example: Low oil pressure.
- **Events** are changes in the process recorded in logs.
  - Example: Start of production.
- **OPC-AE defines:**
  - Collection of alarms and events (event timestamp).
  - Conditions under which they are sent (filtering, priorities).
  - Acknowledgment of alarms.
  - Linking alarms with textual descriptions.
- **OPC-AE clients** are mostly used for alarm and event logs.

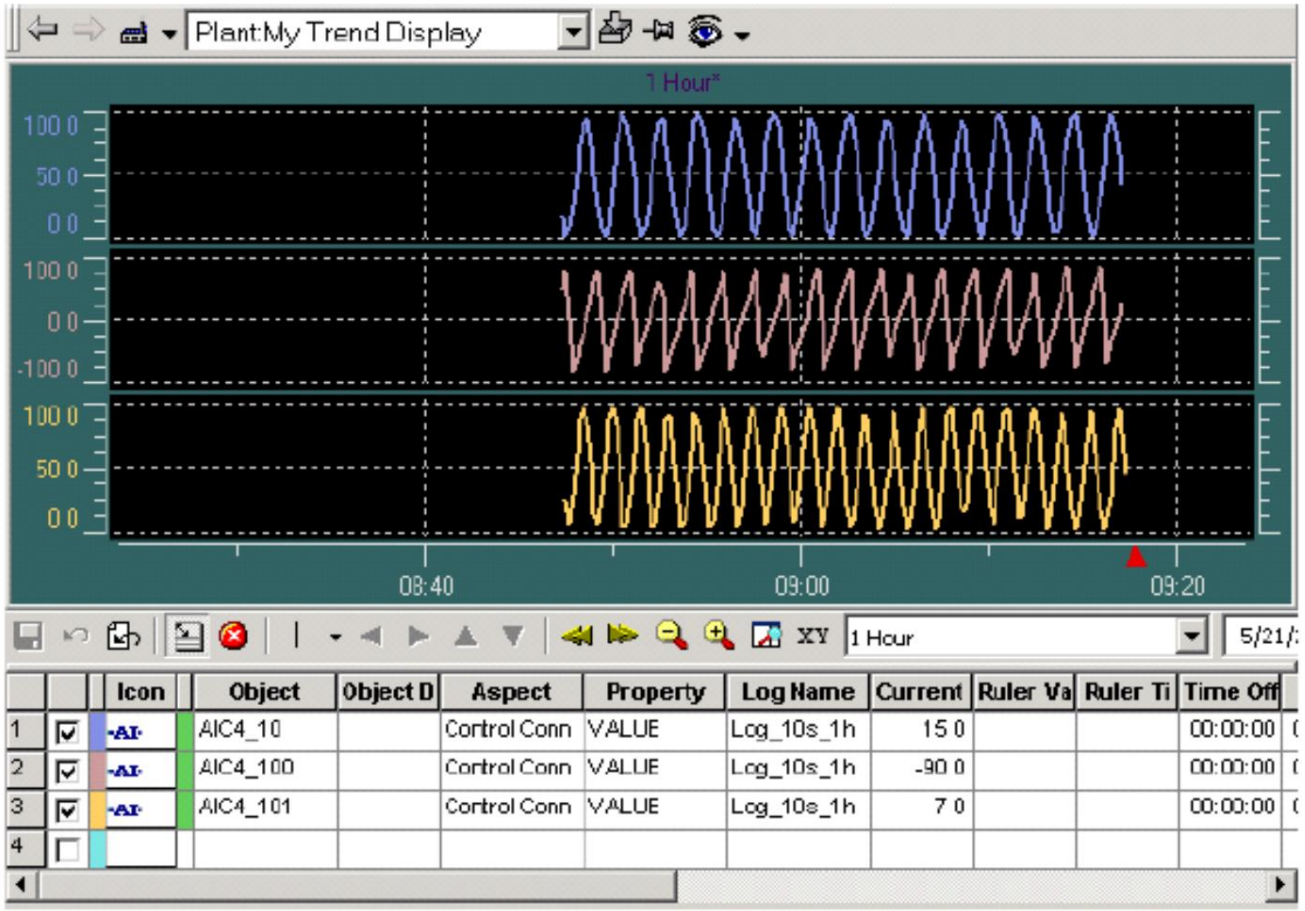


Note: OPC Classic is outdated/obsolete/historical.

OPC Classic – Components (cont.)

OPC-HDA (Historical Data Access) for historical records:

- Historical records represent the state of the system and events, stored in logs for potential later analysis.
  - Example: Process variables, operator actions, recorded alarms.
- **OPC-HDA defines:**
  - How to access historical log records.
  - Filtering procedures.
  - Aggregation (averages, maximums, etc.).
- **OPC-HDA clients** are primarily used for:
  - Displaying trends.
  - Viewing historical records.



Note: OPC Classic is outdated/obsolete/historical.

OPC Classic – Components (cont.)

OPC-Batch for supporting batch production:

- Built on IEC 61512-1.
- Defines interfaces for data exchange between devices:
  - Equipment states.
  - Current operating conditions.
  - Historical records.
  - Recipe contents.

OPC-DX (Data eXchange) for data exchange:

- Standardized interfaces for data exchange between OPC servers.

OPC-XML (eXtended Markup Language) for data exchange (predecessor of OPC UA):

- Data exchange with higher-level systems.
- Adaptation of the standard to Microsoft's .NET framework.
- Usable across various operating systems.
- Specifies the message format for server-client communication using protocols:
  - **XML** (eXtended Markup Language).
  - **SOAP** (Simple Object Access Protocol).
  - **WSDL** (Web Service Definition Language).

Note: OPC Classic is outdated/obsolete/historical.

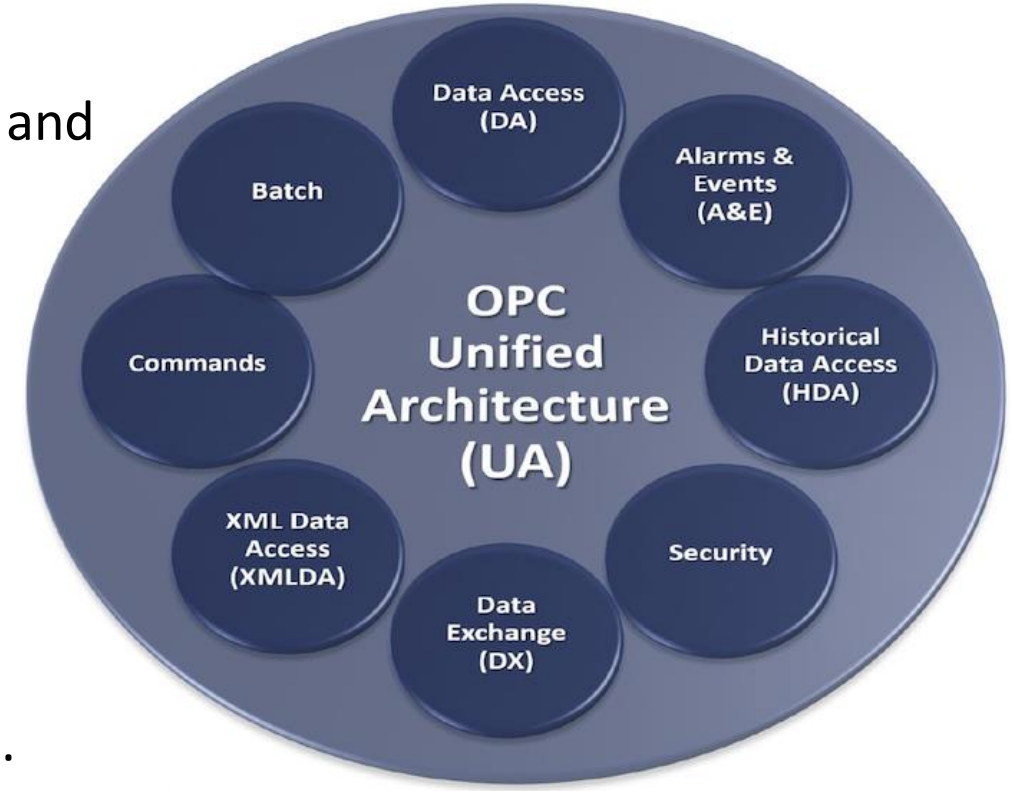
OPC UA

Extends the widely used OPC Classic:

- Enables the connection of PLCs to ERP systems.
- Supports data acquisition, information modeling, and communication.

Utilization of standard technologies:

- Web services (XML, SOAP, WS).
- Simplified configuration and maintenance.
- Increased system visibility (access from multiple device types).
- Broader application areas.
- Reliability.
- Security.
- Speed (OPC binary encoding on the TCP protocol).
- Platform independence (computer architecture, operating system, communication protocol).



OPC UA (cont.)

Scalability:

- **Profiles:**
  - Independent set of functionalities:
    - Methods, data models, security.
  - The server announces what it supports, and the client can verify the required functionalities.
  - Example of a profile: OPC Data Access.

Certificates:

- Certificates are used for secure mutual identification.
- Certificates for servers, clients, and users.
- A list of trusted certificates on each device.

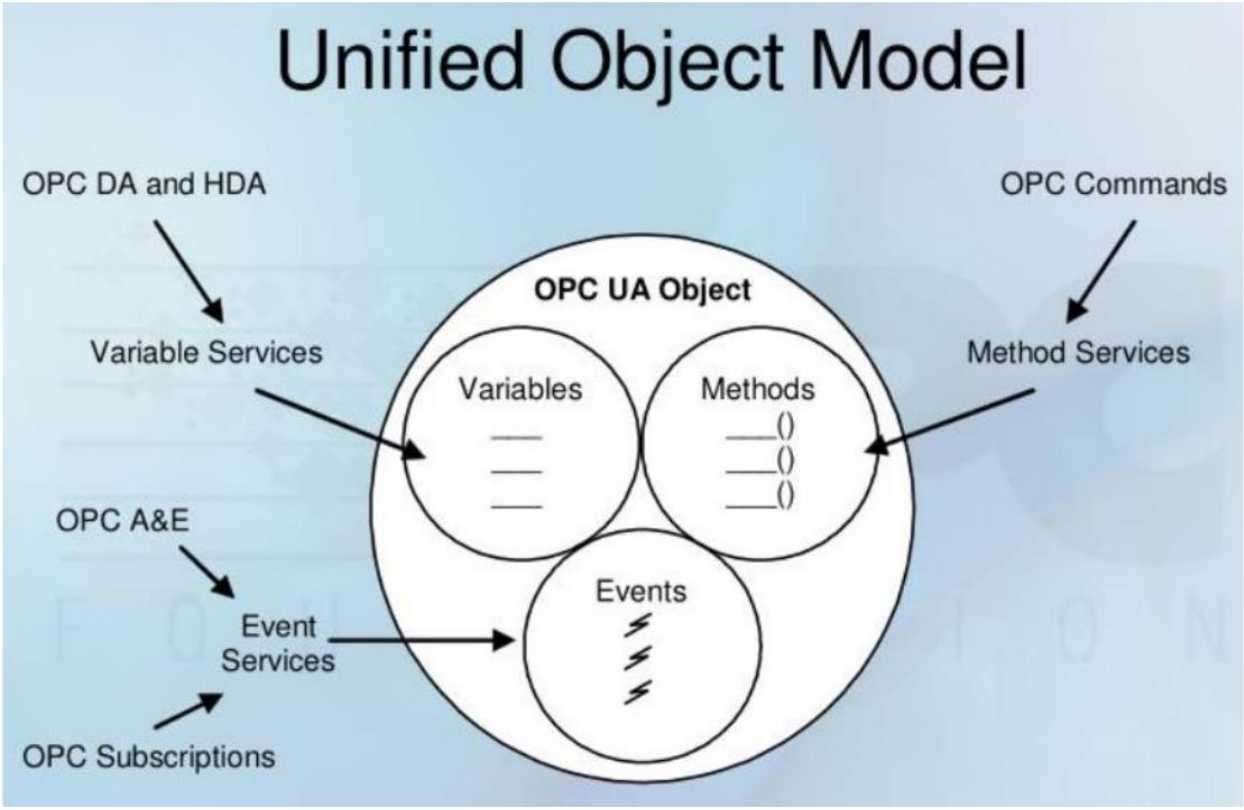
OPC UA (cont.)

Unification of Interfaces

Architecture oriented toward services (Service Oriented Architecture - SOA).

Unified Set of Services

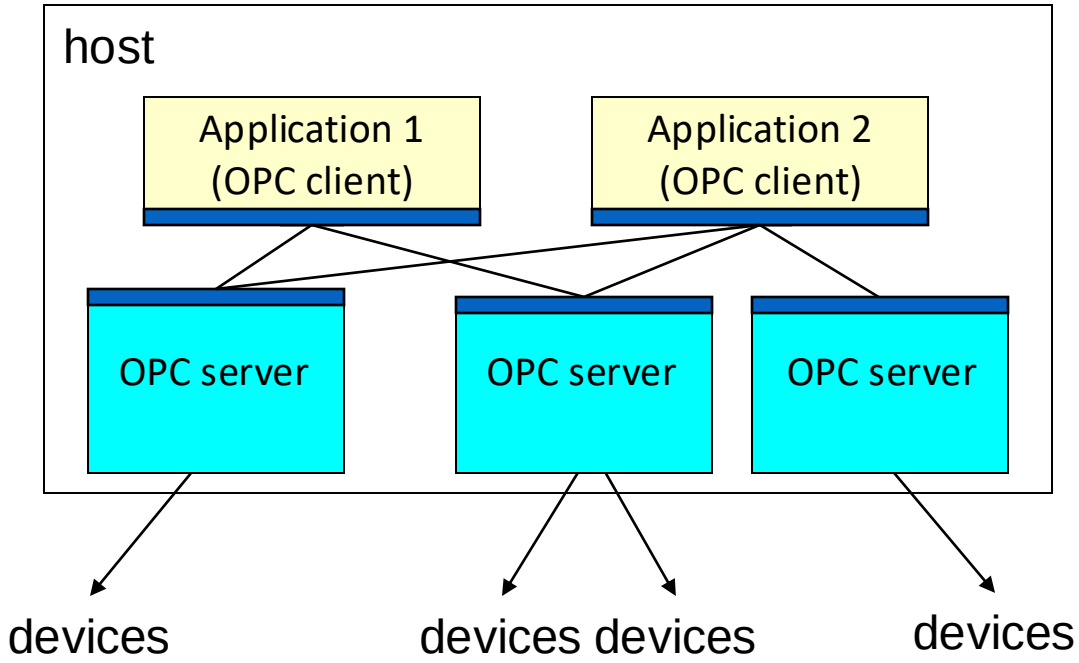
- UA includes OPC Classic functionalities but uses only a single set of services.
- **Services include:**
  - Querying,
  - Reading,
  - Writing,
  - Subscription, etc.
- Named connections between nodes.



Communication

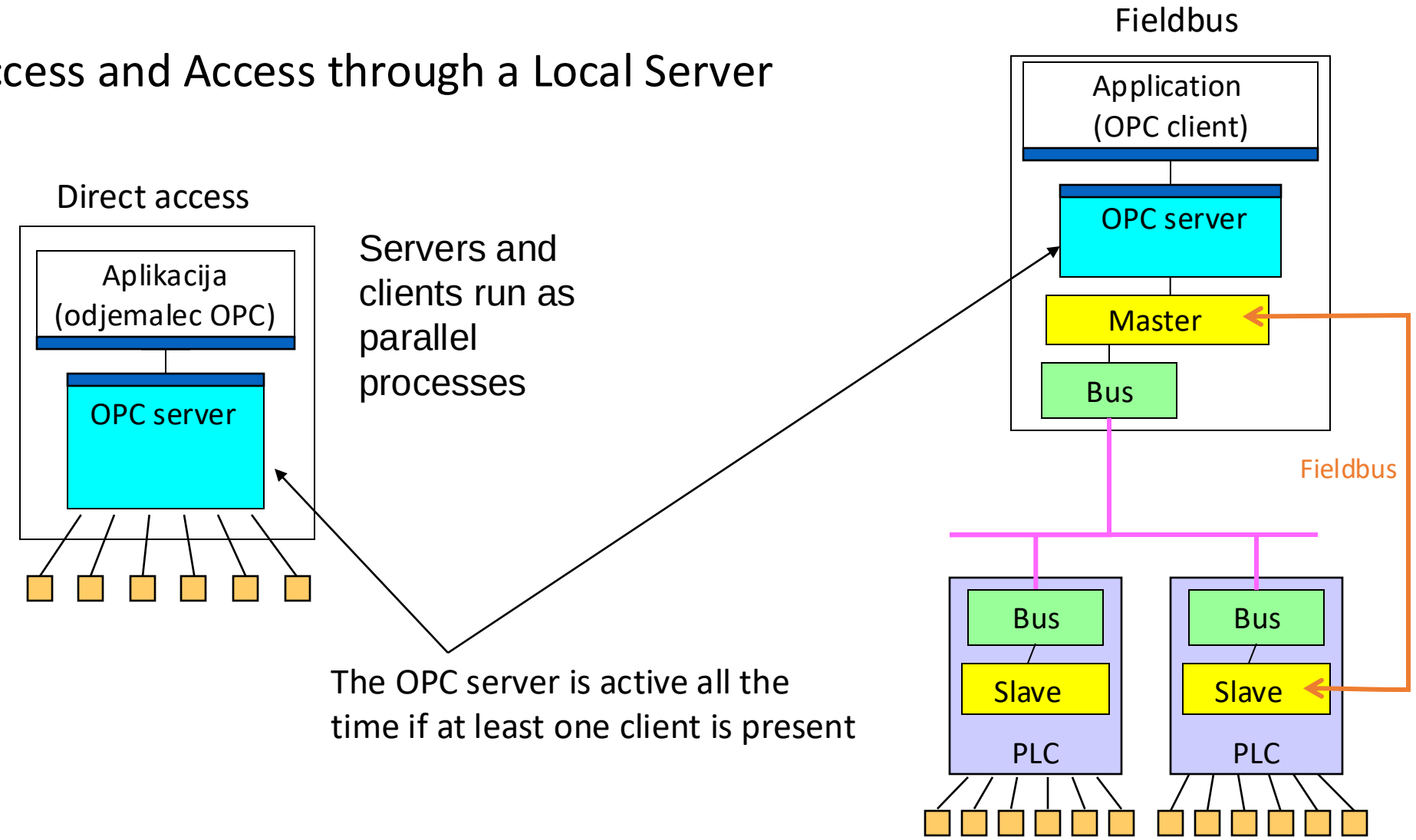
OPC Server and OPC Client on the same computer:

- Servers and clients run as parallel processes.
- The OPC standard provides an interface between the server and client in the form of objects and methods.



Communication (cont.)

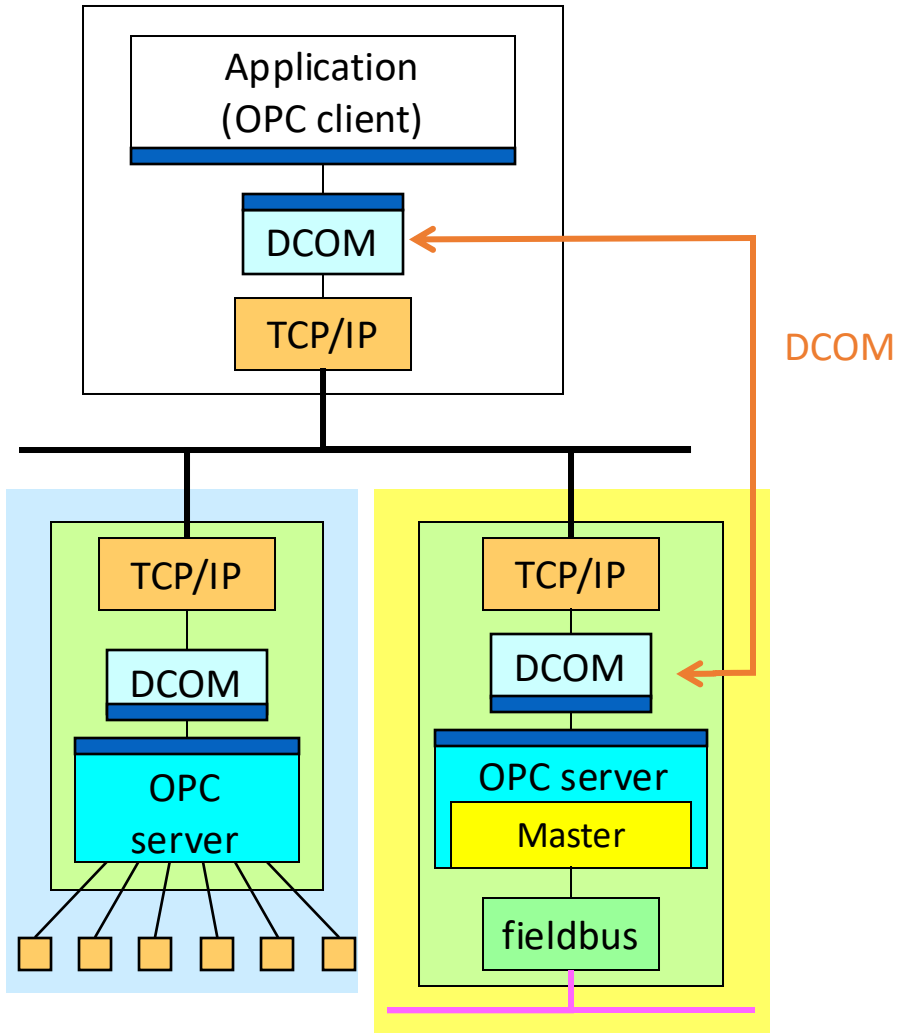
Direct Access and Access through a Local Server



Communication (cont.)

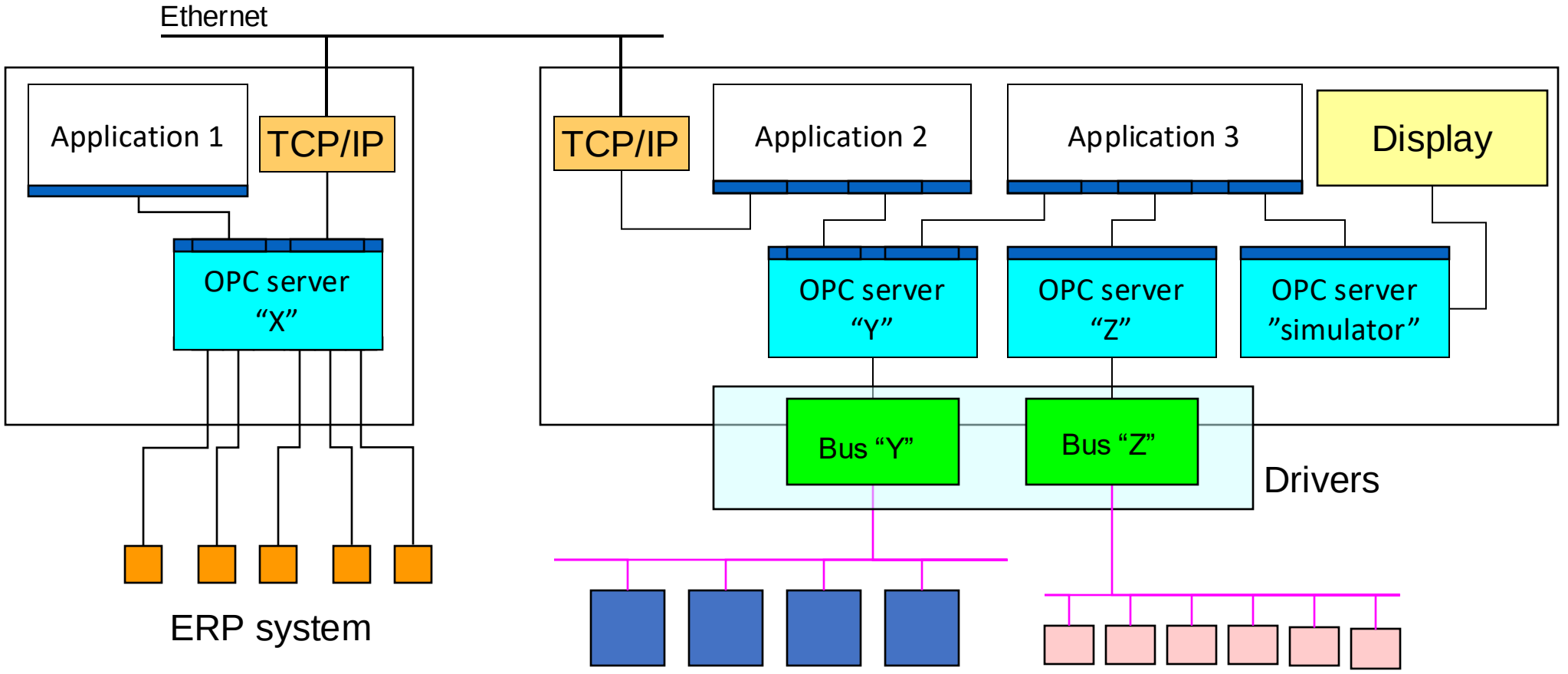
Access to an OPC server on another computer:

- **Problem for OPC Classic:** Firewalls.
- **Solution:** OPC UA.

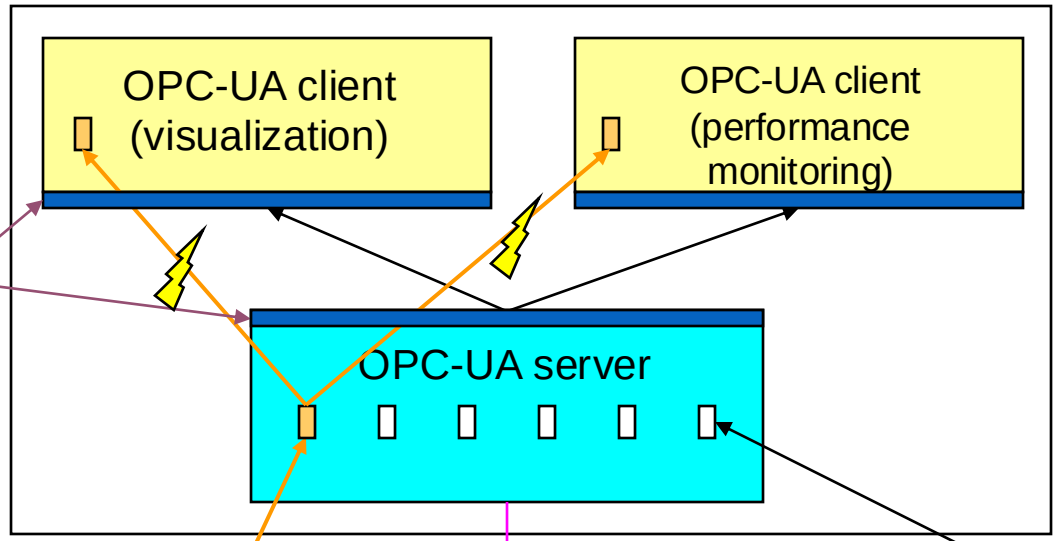


Communication (cont.)

- OPC servers support multiple clients on the same or different computers.



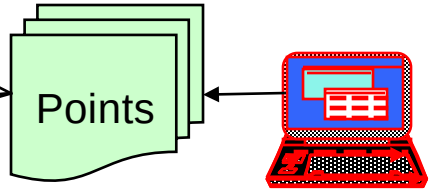
OPC UA: Data Access Specification

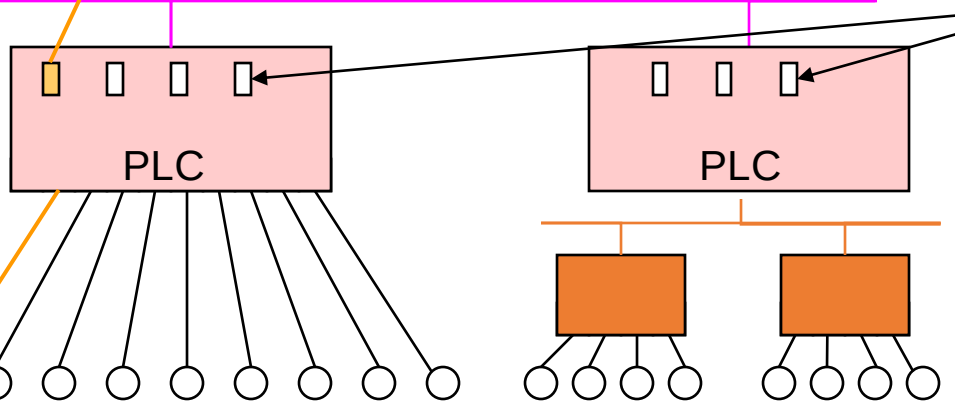


Interface according to the OPC UA standard

The OPC UA server is configured using development tools for controllers and SCADA systems

Variables defined in the PLC are mirrored in the OPC UA server





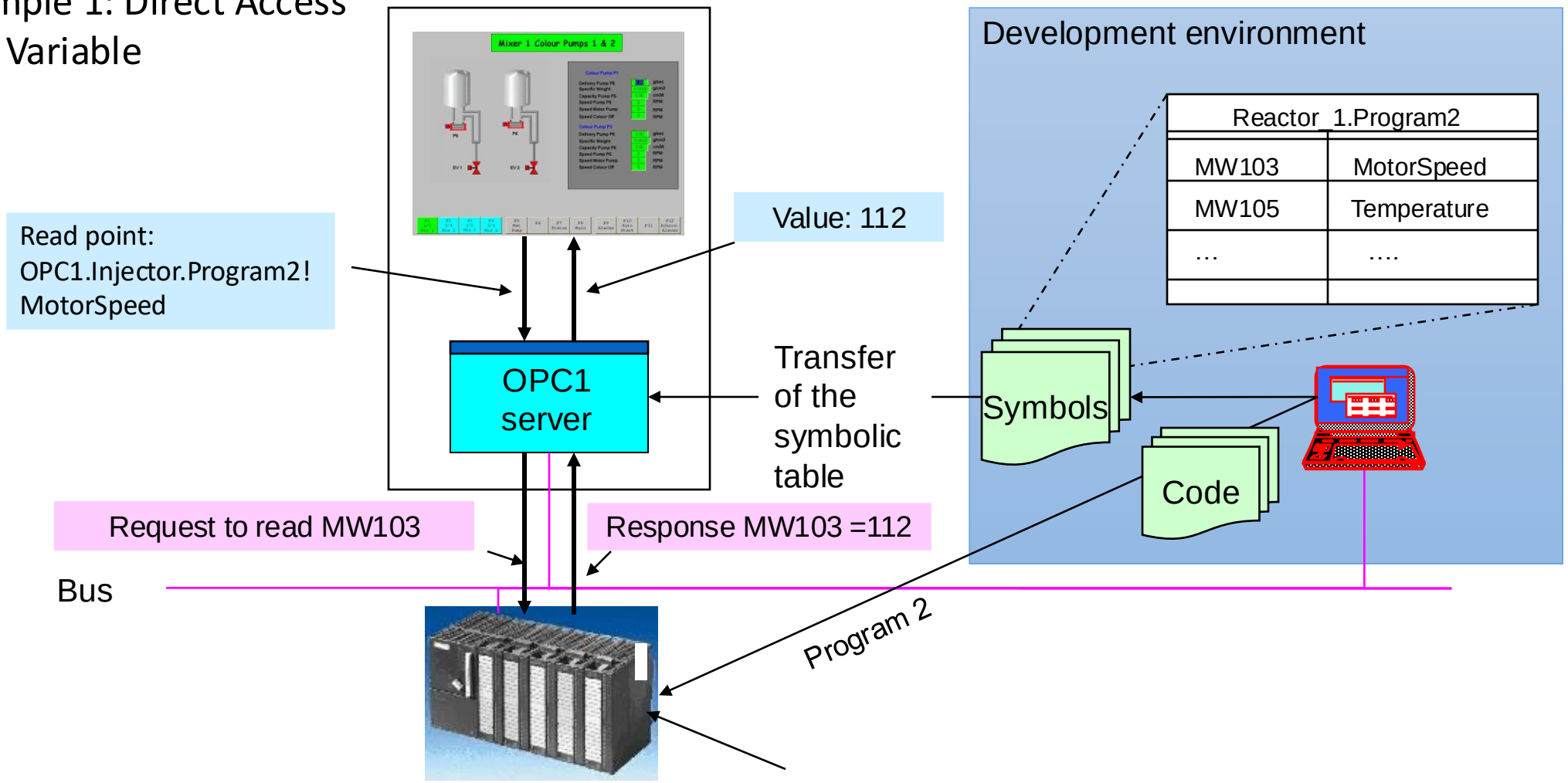
Fieldbus

Field devices

System - sensors

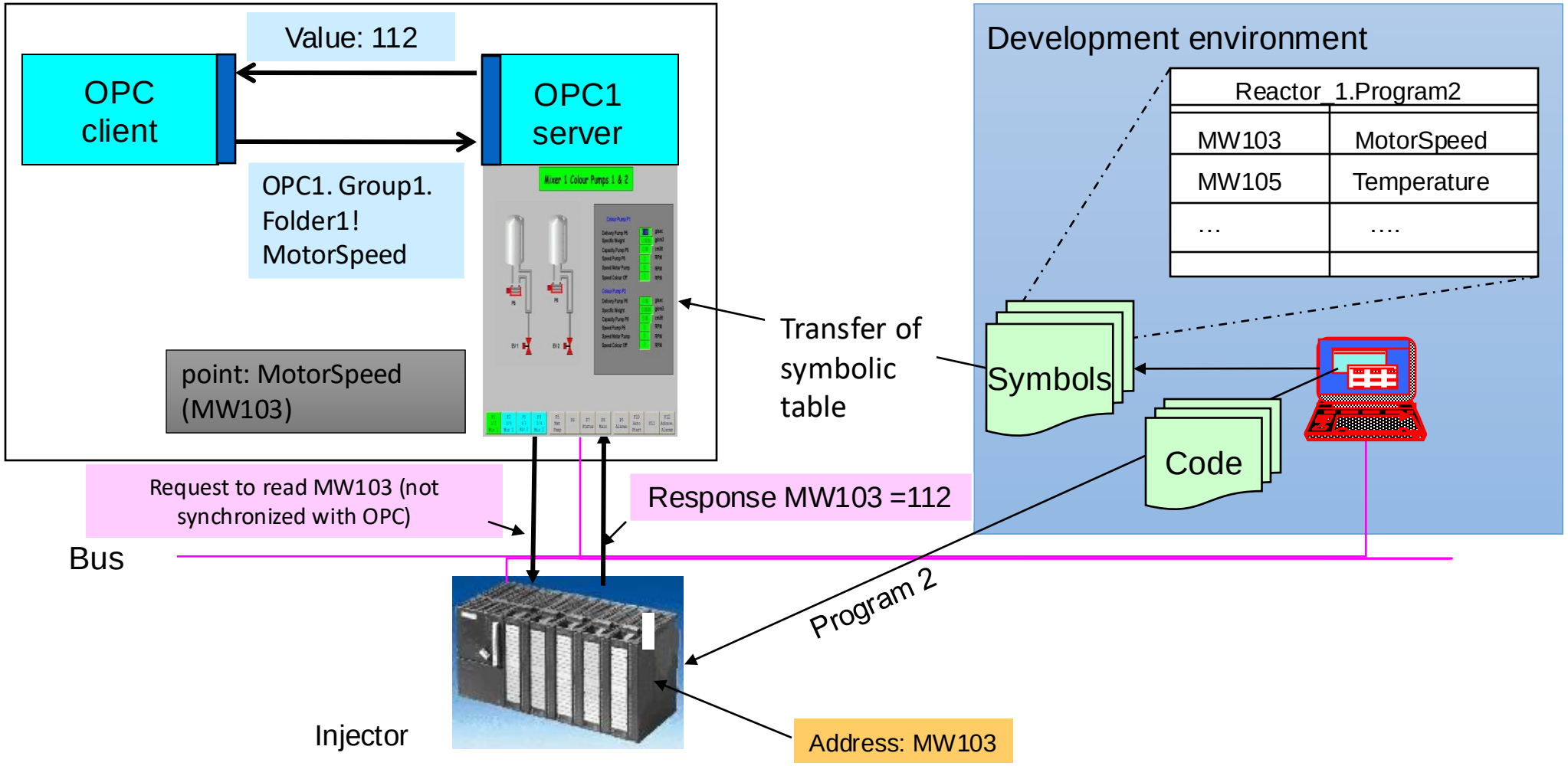
OPC UA: Data Access Specification (cont.)

Example 1: Direct Access to a Variable



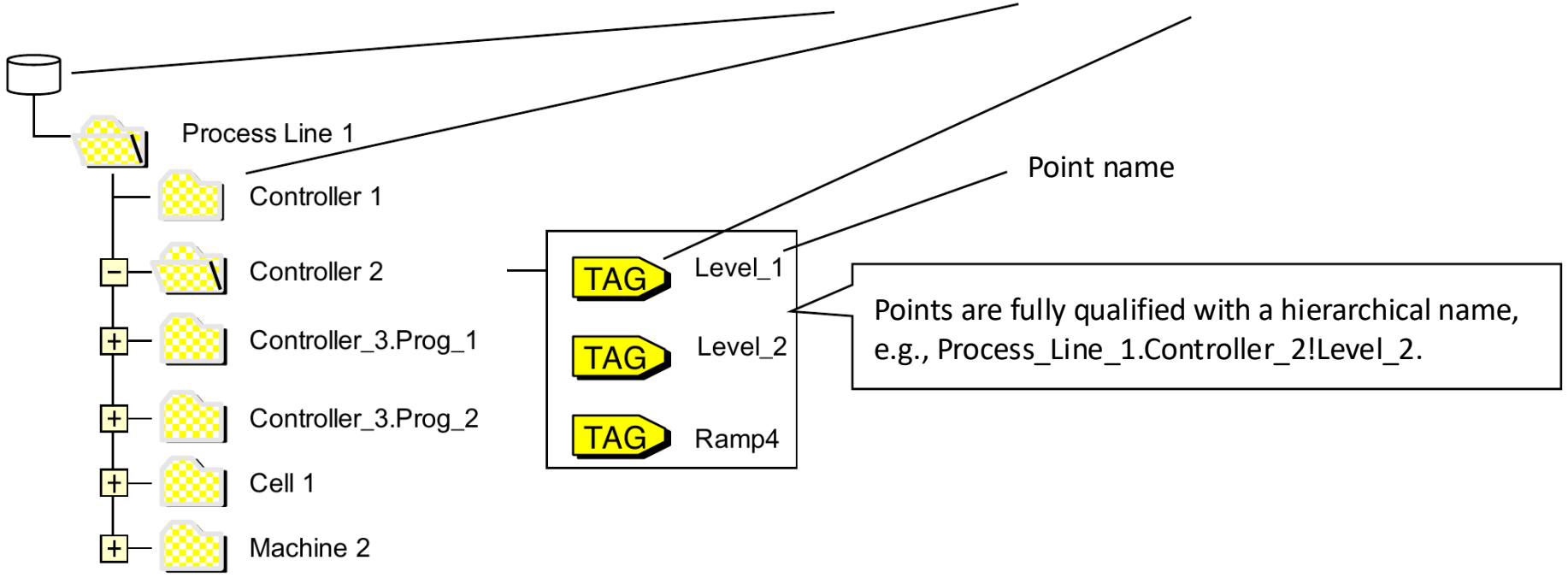
OPC UA: Data Access Specification (cont.)

Example 2: Indirect access to a variable via a SCADA system



Data organization

- OPC Server is organized as a tree with roots, branches, and leaves.



- Branches can contain sub-branches and data points.
- The structure does not have to be strictly hierarchical.
- The structure is established during system configuration.

Data organization (cont.)

- **Data is organized into nodes, forming a tree structure:**
  - The complete set of nodes on a server constitutes the **address space**.
- **Nodes are typed:**
  - Each node has predefined attributes based on its type.
  - These attributes can have dynamic values.
- **Each node has a unique identifier:**
  - Can be a string, integer, or **GUID** (Globally Unique Identifier).
  - The identifier must be unique within its namespace.
  - Namespaces are defined with integers.
- **The most general type of node is an object:**
  - Can contain **variables**.
  - Can contain **methods**.
  - Functions similarly to objects in object-oriented programming.

Server operations

- The client establishes a connection to the server:
  - The server is specified with a URL, e.g., opc.tcp://localhost:4840/.
- Server discovery
- Browsing the server
- Reading and modifying variables
- Subscribing to variable changes and other events:
  - No need to manually check when changes occur.
  - If no changes are detected, the server still periodically reports:
    - Values are not transmitted in this case.
    - The interval is defined by the client.
- Method calls:
  - Objects can have attached methods that the client can invoke.

Data types and properties

Each element has a defined data type:

- The data type of a point on the server can be obtained by browsing the server (special software tools, called browsers, are used for this).
- It is essential to ensure that data types on the server and the client are compatible.

```
Boolean,
Character,
Byte,                          (1 byte)
Word,                          (2 bytes)
Double Word,                   (4 bytes)
Short Integer                  (2 bytes)
Integer                        (4 bytes)
Long Integer
Long Unsigned Integer
Single Float                   (4 bytes)
Double Float                   (8 bytes)
Currency
Date
String
Array of …
```

Data types and properties (cont.)

Points on clients are represented with the following dynamic properties:

- **Value:** Corresponds to previously mentioned data types.
- **Quality:** Indicates the validity of the record (e.g., OK, invalid, questionable).
- **Timestamp:** The time at which the value was transferred from the PLC to the server.
  - Time is provided in **UTC** (Universal Time Coordinated).

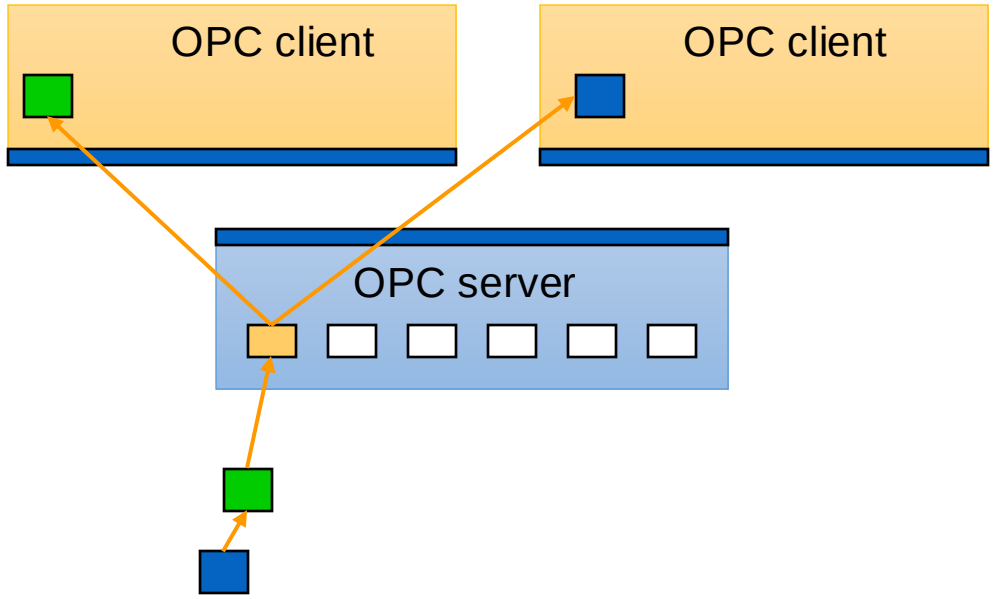
Behavior:

- When reading, all three properties (Value, Quality, Timestamp) are updated.
- When writing, only the **Value** property is used.

Programming

Reading and Writing:

- A new value overwrites the old value; there are no queues or historical records.
- The OPC server does not guarantee that different clients will see the same process state.



Limitations:

- The OPC server does not ensure that all changes to objects are detected.
- If the refresh rate is too slow, some changes might be missed.

Programming (cont.)

Event-Driven Reading:

- A function is called every time an element in a group changes.
- The variable is identified using its address on the client, not its full name.
- For each point, the function provides its value, quality, and timestamp.

Behavior:

- The OPC client can call the function even when a value in any group changes (Global Data Change).
- The application must determine which group and which point the changes belong to.

Programming (cont.)

- **Libraries:**
  - To ensure compliance with the OPC standard, OPC Foundation offers libraries on their website required for accessing OPC servers.
    - Code in C and Java is no longer maintained.
    - .NET Core: https://github.com/OPCFoundation/UA-.NETStandard
  - **Challenges:**
    - Server manufacturers do not always implement all functionalities, which can result in unexpected errors.
- **Beckhoff TwinCAT:**
  - **Resources:**
    - Infosys: TF6100
    - Presentation and Best Practices

Tools, resources

- **OPC Foundation:**
  - https://opcfoundation.org/
  - Standard specifications (members only)
  - Servers, clients, browsers, simulators
- **Python (FreeOpcUa):**
  - **python-opcua** (deprecated): https://github.com/FreeOpcUa/python-opcua
  - **Asynchronous API:** https://github.com/FreeOpcUa/opcua-asyncio
  - **Client with GUI:** https://github.com/FreeOpcUa/opcua-client-gui
- **NodeJS:**
  - https://node-opcua.github.io/
- **UA Expert:**
  - Fully functional client/browser
  - https://www.unified-automation.com/products/development-tools/uaexpert.html

Process automation

The OPC standard

BS UNI studies, Fall semester 2024/2025

Octavian M. Machidon

octavian.machidon@fri.uni-lj.si