Southern Illinois University
CARBONDALE

Department of Computer Science
Southern Illinois University Carbondale

CS 491/531
SECURITY IN CYBER-PHYSICAL SYSTEMS

Lecture 9: Industrial Network Protocols

DR. ABDULLAH AYDEGER

LOCATION: ENGINEERING A 409F

EMAIL: AYDEGER@CS.SIU.EDU

Outline

Industrial Network Protocols

- ICCP
- OPC

Recall: Modbus Operation

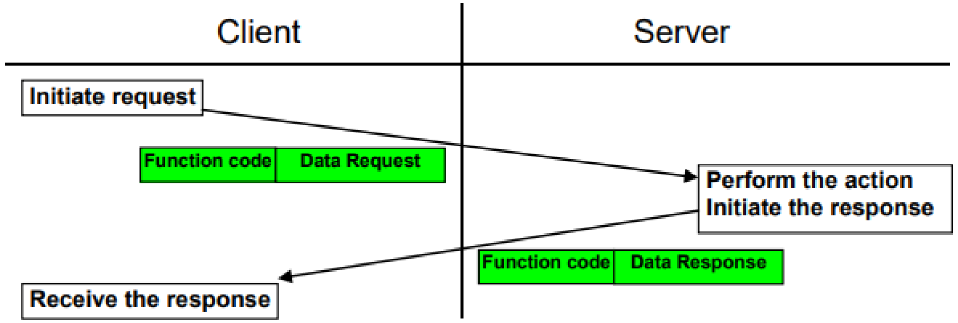
Starts with initial Function Code and a Data Request within a Request PDU

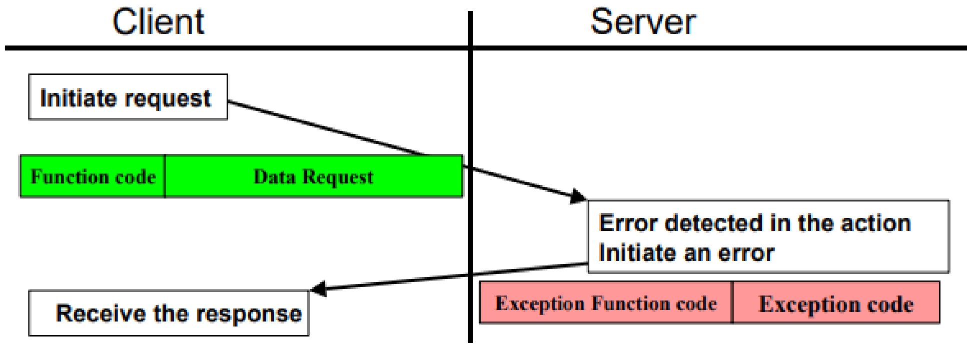
Response either:

- Function Code and Data Response, if no error
- Exception Function Code and Exception Code, if error

Examples of Function Codes and Data Requests:

- Read from an I/O interface
- Write a value to a register (i.e., change the value in register)

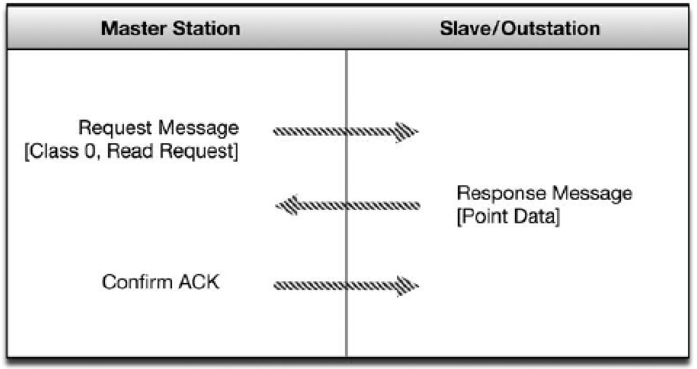


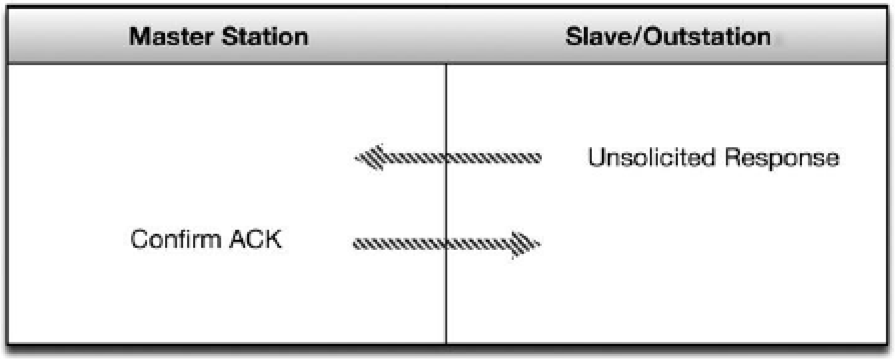


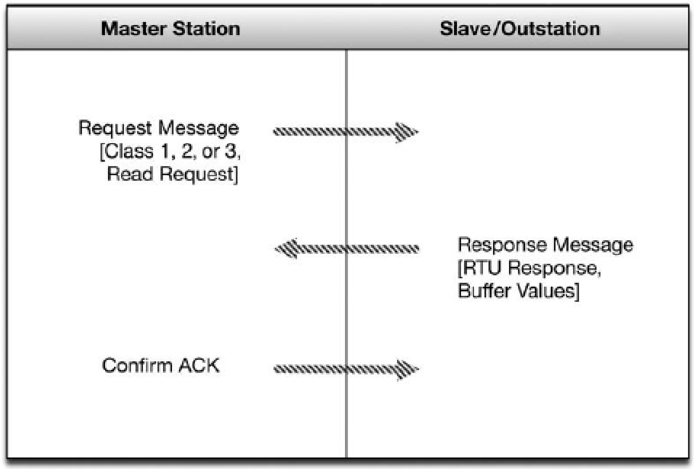
Recall: DNP3 Characteristics

Bidirectional (supporting communications from both Master to Slave and from Slave to Master) and supports exception-based reporting

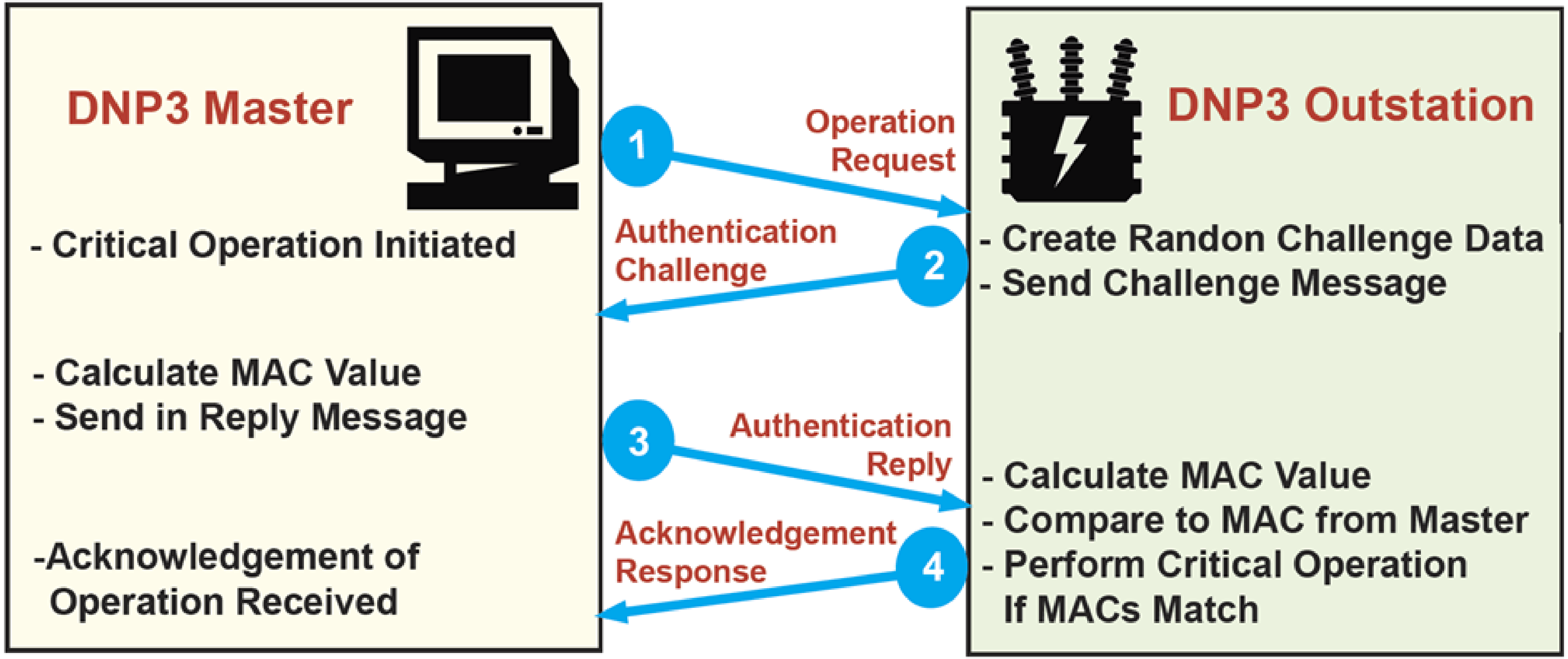
- Possible for a DNP3 outstation to initiate an unsolicited response to notify the Master of an event outside of the normal polling interval
  - Such as an alarm condition


Master Station
Slave/Outstation
Request Message [Class 0, Read Request]
Response Message [Point Data]
Confirm ACK




Master Station
Slave/Outstation
Request Message [Class 1, 2, or 3, Read Request]
Response Message [RTU Response, Buffer Values]
Confirm ACK

Recall: Secure DNP3 Standard



DNP3 Master

- Critical Operation Initiated

- Calculate MAC Value
- Send in Reply Message

-Acknowledgement of
Operation Received

1 Operation Request

2 Authentication Challenge

3 Authentication Reply

4 Acknowledgement Response

DNP3 Outstation

- Create Randon Challenge Data
- Send Challenge Message

- Calculate MAC Value
- Compare to MAC from Master
- Perform Critical Operation If MACs Match

Inter Control Center Protocol (ICCP)

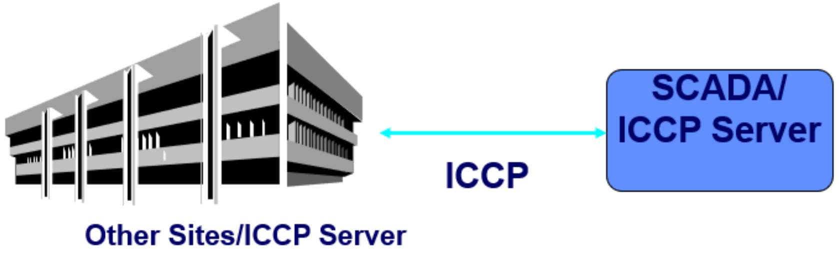
Also known as TASE.2 or IEC 60870-6

Designed for communication between control centers within the energy industry

- Bidirectional Wide Area Network (WAN) communication between a utility control center and other control centers; power plants, substations, and even other utilities

Why is it required?

- To provide standardization for different entities managing regional utilities
- Vendor interoperability over any network



ICCP Functions

Periodic System Data

- Status points, analogue points, quality flags, time stamp, counters, protection events

Device Control

- On/off, trip/close, raise/lower etc. and digital setpoints

Program Control

- Allows an ICCP client to remote control programs executing on an ICCP server

Info transmission

- Scheduling, accounting, outage and plant information
- Historical time series data between a start and end date

ICCP Characteristics

Client/server model

ICCP support is integrated either:

- Directly into a control system
- Provided via a gateway product
- Provided as software running on Unix that can then be installed to perform gateway functions

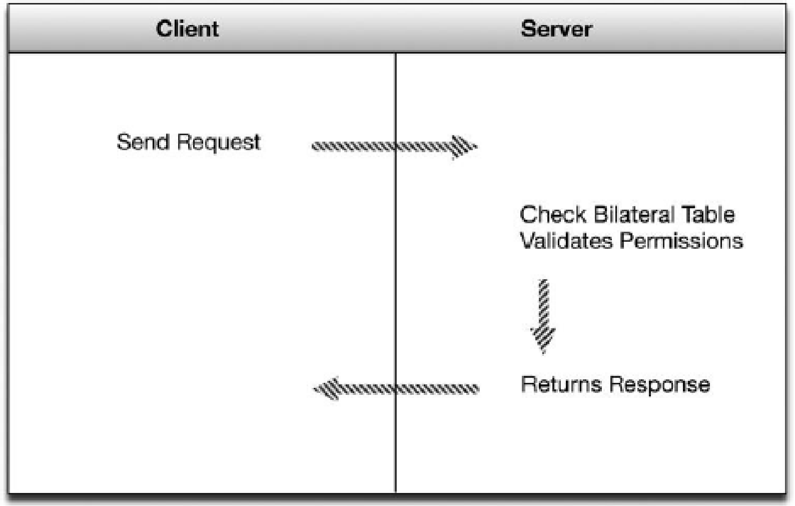
Primarily designed as a unidirectional client/server protocol

- Most modern implementations are bidirectional

ICCP Characteristics

Point-to-point protocol

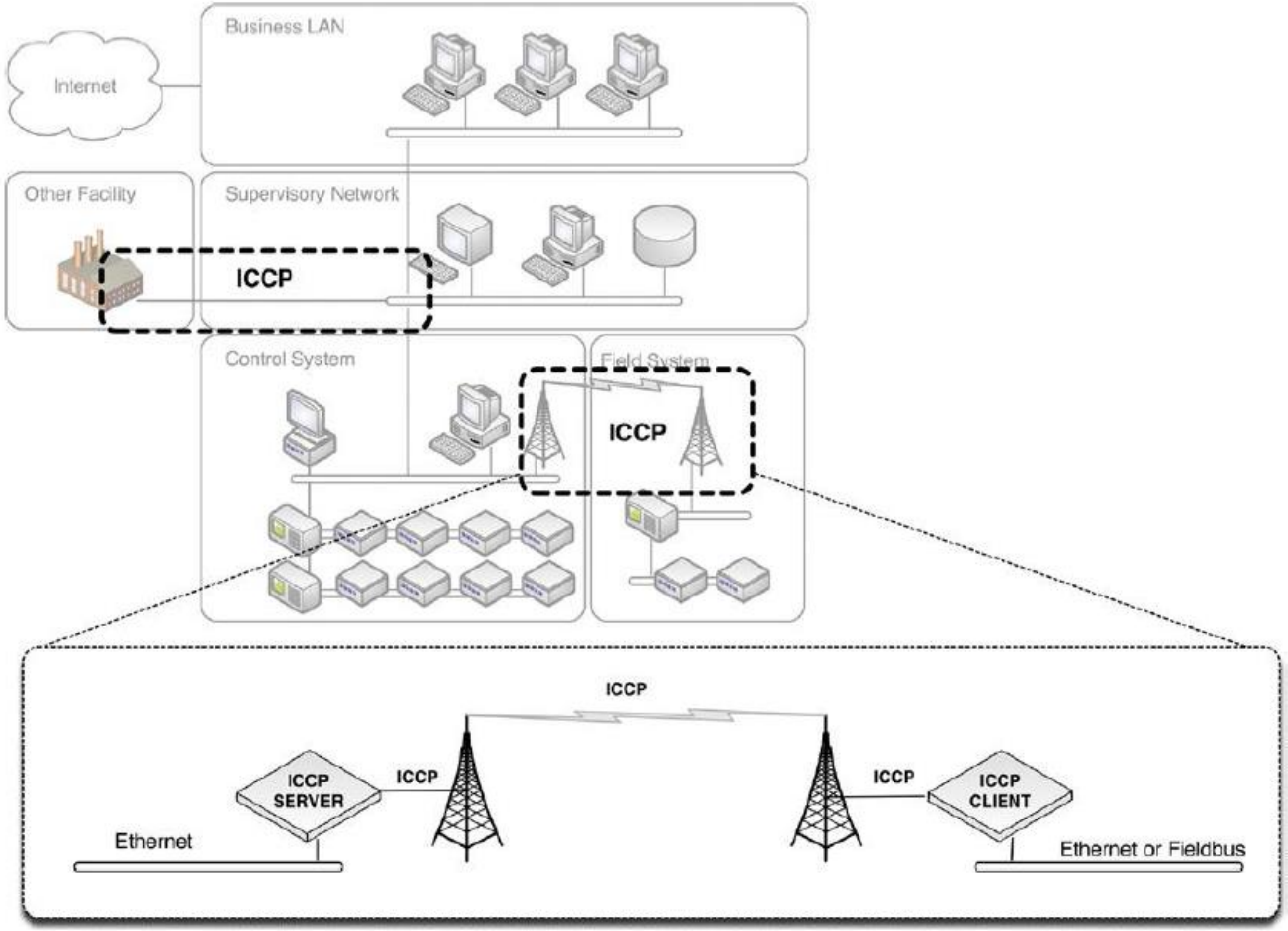
- “Bilateral table” defines an agreement between two control centers connected with an ICCP link
  - Access control list that identifies which data elements a client can access



Where ICCP is used

A few examples:

- Between two electric utilities
- Between two control systems within a single electric utility
- Between a main control center and a number of substations



Security Concerns of ICCP

Lack of authentication and encryption:

- Not mandate authentication or encryption, most often deferring these services to lower protocol layers
  - Although Secure ICCP does exist, it is not ubiquitously deployed

Explicitly defined trust relationships:

- The impact of the exploitation of bilateral tables?

Accessibility:

- WAN and its open to many attacks in the wild

Security Additions of ICCP

Bilateral tables provide basic control

A secure version of ICCP exists that incorporates digital certificate authentication and encryption

Some Security Recommendations for ICCP

WAN protocol; proper penetration testing and patching of ICCP servers and clients

Extreme care in the definition of the bilateral table

- Primary enforcement of policy and permissions between control centers and malicious commands

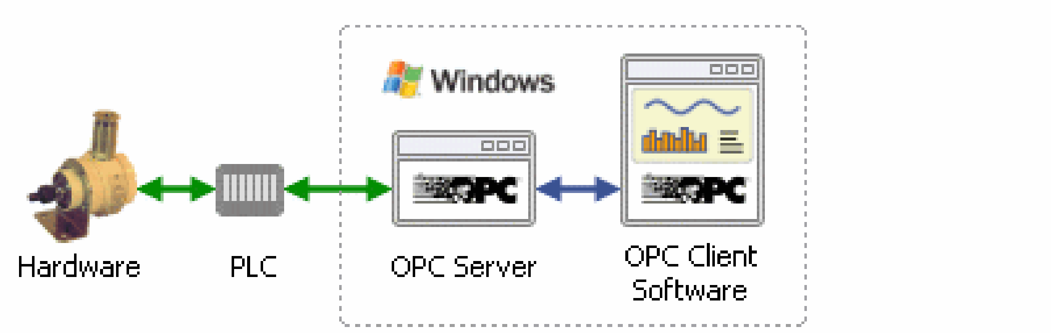
ICCP clients and servers should be isolated into a unique enclave

OLE FOR PROCESS CONTROL (OPC)

Not an INP

- Operational framework for the communication of Windows-based process control systems
- Uses Microsoft's Object Linking and Embedding (OLE) protocol

Suite of protocols that collectively enable process control systems to communicate using some of the underlying networking capabilities of Windows



OPC Characteristics

Server/client pairs

OPC Server is a software program that converts the hardware communication protocol used by a PLC into the OPC protocol

- Client needs to connect to the hardware,
  - Such as an HMI

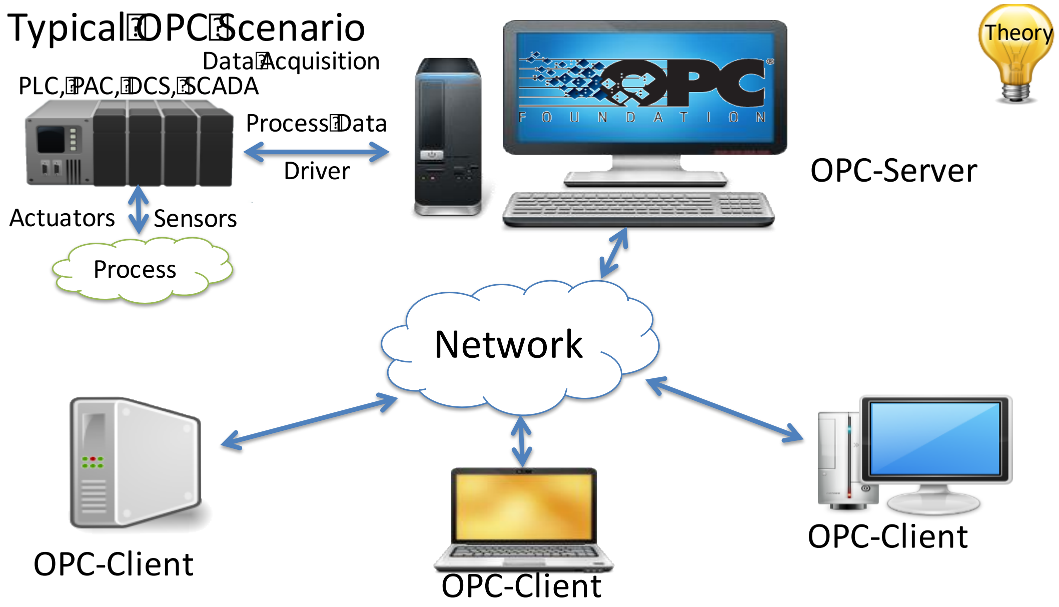
Open source

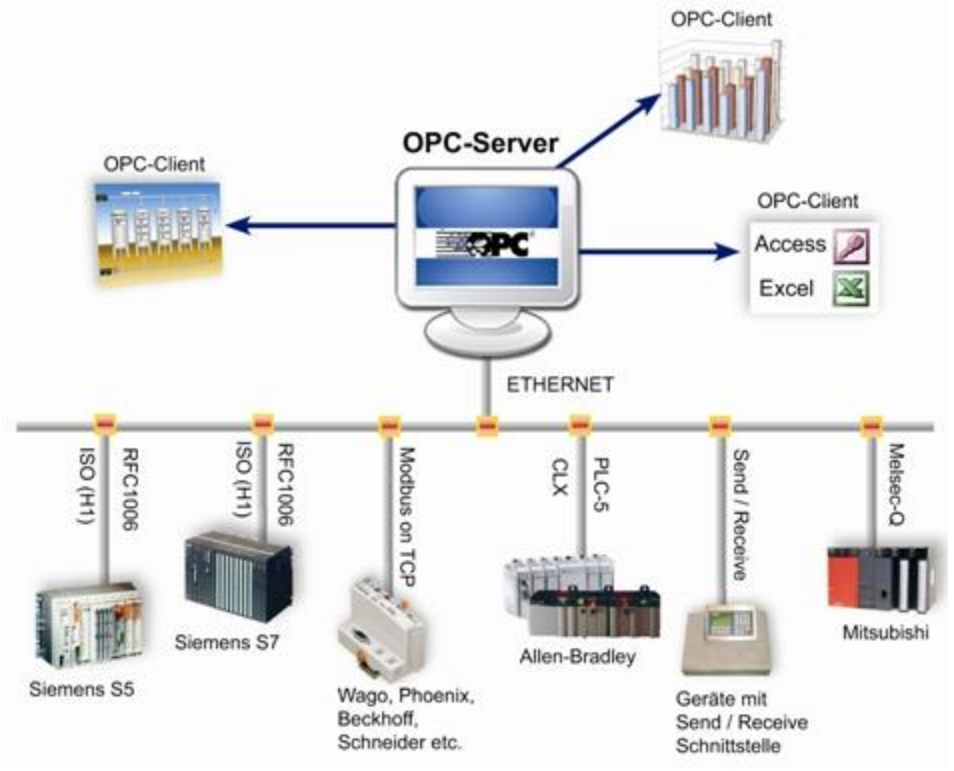
OPC stands for Open Platform Communications

OPC Characteristics

Primary function is to interconnect other distributed control systems with Windows hosts

- Typically connected via an Ethernet TCP/IP network



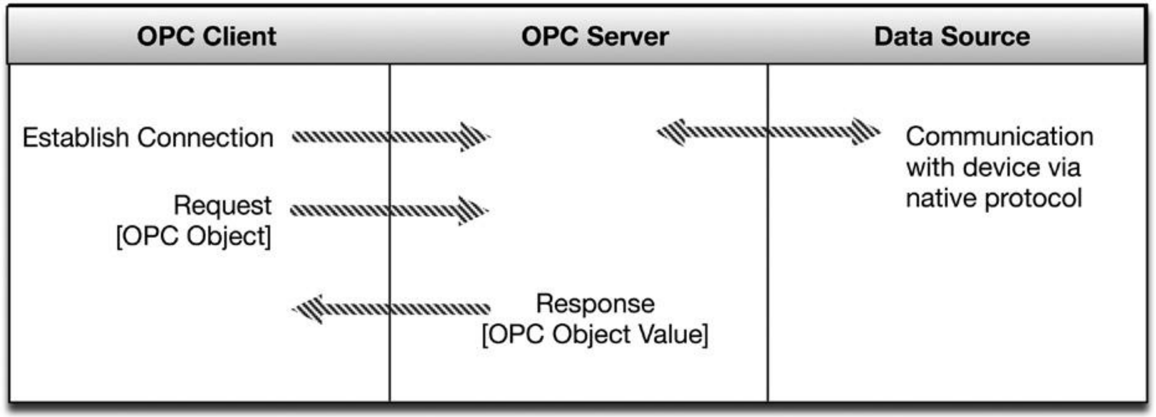


OPC Operation

Client application calls a local process,

- But instead of executing the process using local code, the process is executed on a remote server

1. The process is performed remotely (on the server)
2. Server Remote Procedure Call (RPC) functions then transmit the requested data back to the client computer
3. Finally, the client process receives the data over the network, provides it to the requesting application, and closes the session



OPC Client

OPC Server

Data Source

Establish Connection

Communication with device via native protocol

Request [OPC Object]

Response [OPC Object Value]

OPC Variants:

OPC Unified Architecture (UA)

OPC Express Interface (XI)

OPC Unified Architecture

Platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework

Design specification goals:

- Functional equivalence: all COM OPC Classic specifications are mapped to UA
- Platform independence: from an embedded micro-controller to cloud-based infrastructure
- Secure: encryption, authentication, and auditing
- Extensible: ability to add new features without affecting existing applications
- Comprehensive information modeling: for defining complex information

OPC Express Interface (XI)

OPC .NET (previously OPC Xi) is a communication interface specification designed for secure and reliable access to automation systems

OPC .NET provides a set of methods for accessing real-time data, historical data, events, and alarms

It has been designed for fast local communication, and secure network communication

OPC .NET defines a Service Oriented Architecture (SOA) that is based on MMS (Manufacturing Messaging Service) and WCF (Windows Communication Foundation)

OPC .NET

Standardizes a Windows Communication Foundation (WCF) interface for OPC Classic servers

Standardizes a standard OPC Client application programming interface (API) for accessing both OPC Classic servers via WCF and for accessing OPC UA servers via the UA protocol

Provides a standard OPC .NET wrapper for OPC Classic servers in the form of source code to allow it to be adapted to any .NET platform

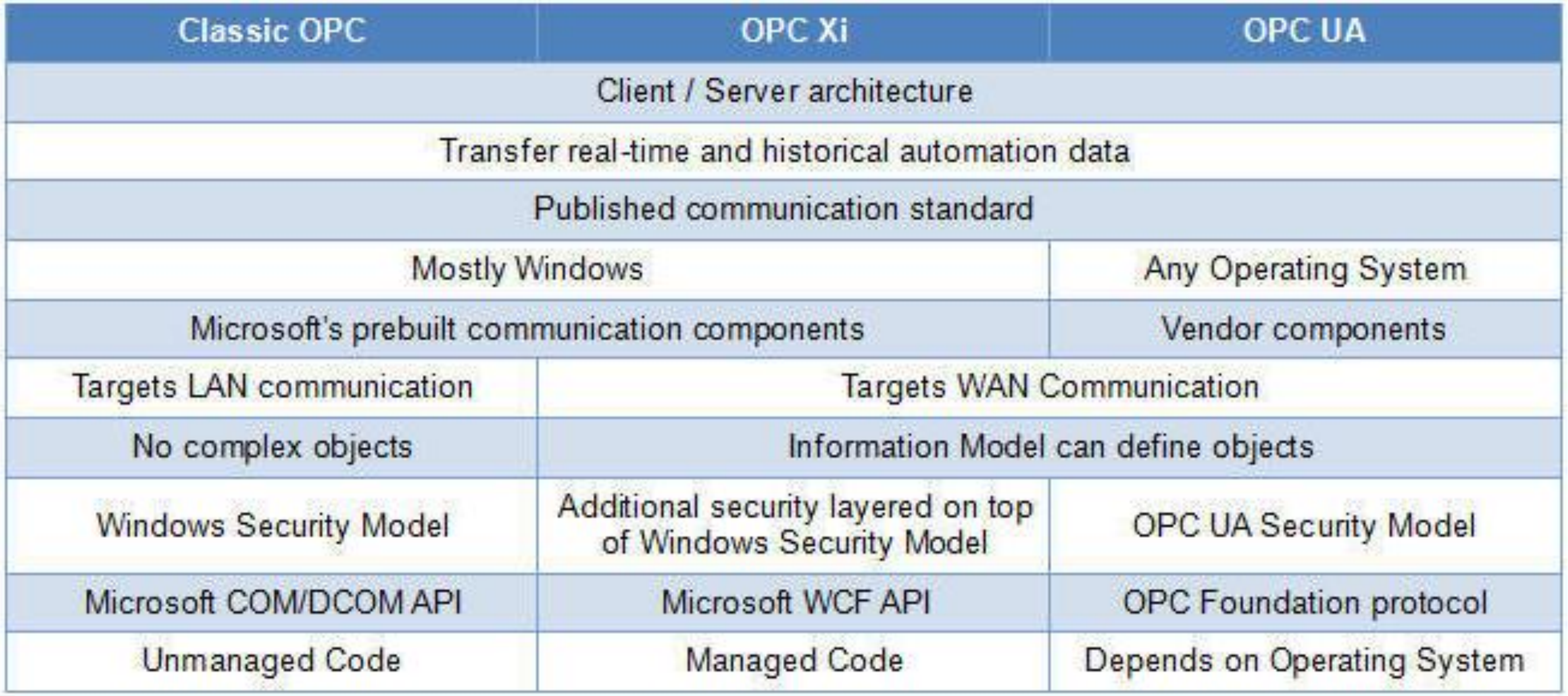
- A standard OPC .NET Client Proxy for WCF that supports the standard OPC Client API for access

OPC Variants

Classic OPC

OPC Xi

OPC UA



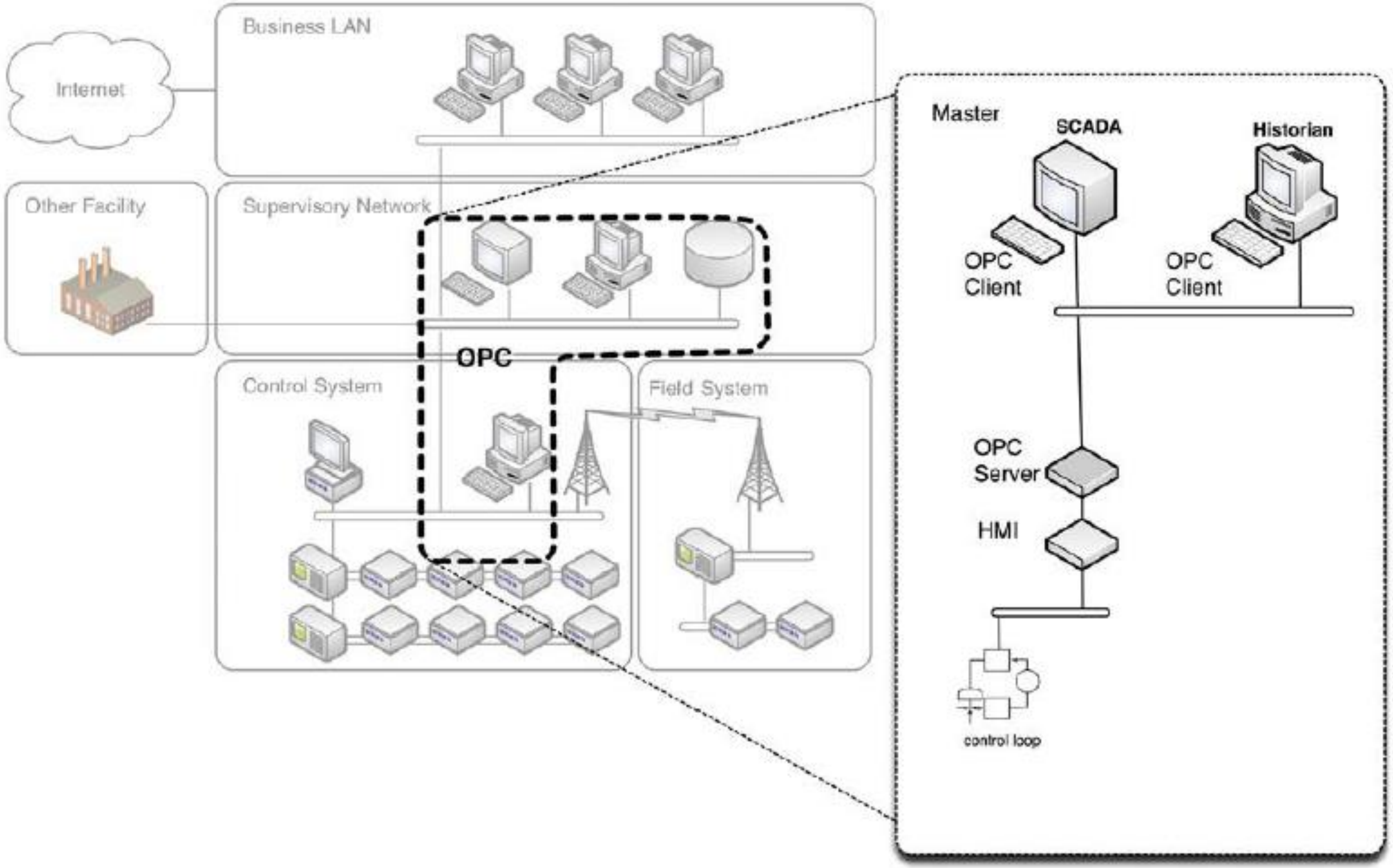
<table>
<tr><th>Classic OPC</th><th>OPC Xi</th><th>OPC UA</th></tr>
<tr><td colspan="3">Client / Server architecture</td></tr>
<tr><td colspan="3">Transfer real-time and historical automation data</td></tr>
<tr><td colspan="3">Published communication standard</td></tr>
<tr><td colspan="2">Mostly Windows</td><td>Any Operating System</td></tr>
<tr><td colspan="2">Microsoft's prebuilt communication components</td><td>Vendor components</td></tr>
<tr><td>Targets LAN communication</td><td colspan="2">Targets WAN Communication</td></tr>
<tr><td>No complex objects</td><td colspan="2">Information Model can define objects</td></tr>
<tr><td>Windows Security Model</td><td>Additional security layered on top of Windows Security Model</td><td>OPC UA Security Model</td></tr>
<tr><td>Microsoft COM/DCOM API</td><td>Microsoft WCF API</td><td>OPC Foundation protocol</td></tr>
<tr><td>Unmanaged Code</td><td>Managed Code</td><td>Depends on Operating System</td></tr>
</table>

Where OPC is used

Data transfer to data historians, data collection within HMIs, and other supervisory controls

- Either between Windows-based devices,
- Or via OPC gateways

Also widely used within an industrial system's business network, including connections to corporate intranets, and even the Internet



OPC Security Concerns

Remote Procedure Call (RPC) makes it highly vulnerable to attack, as it is subject to the same vulnerabilities as the more ubiquitously used OLE

Windows based vulnerabilities

Many systems support additional Windows services that are irrelevant to SCADA systems, resulting in unnecessary processes, which often correspond to open ports

- What then?
- Broader attack surface

OPC Security Concerns

OPC Server Integrity:

- Possible to create a rogue OPC server and to use that server for disruption of service, DoS, information theft through bus snooping, or the injection of malicious code

Legacy authentication services

- Systems within industrial networks are difficult to upgrade

OPC Security Recommendations

OPC servers should be isolated into a unique enclave consisting only of authorized devices

- Standard defense in-depth practices, including a firewall and/or IDS/IPS system that enforces strict control over the type, source, and destination of traffic to and from the OPC enclave

All unnecessary ports and services should be removed or disabled from the OPC server

- Irrelevant applications, and all unused network protocols

A few other INPs

Ethernet/IP

Profibus

EtherCAT

Ethernet Powerlink

SERCOS III