ROYAL INSTITUTE
OF TECHNOLOGY

Exercise 5: Interfacing with OPC and IEC 60870-5-104

Nicholas Honeth (nicholash@ics.kth.se)

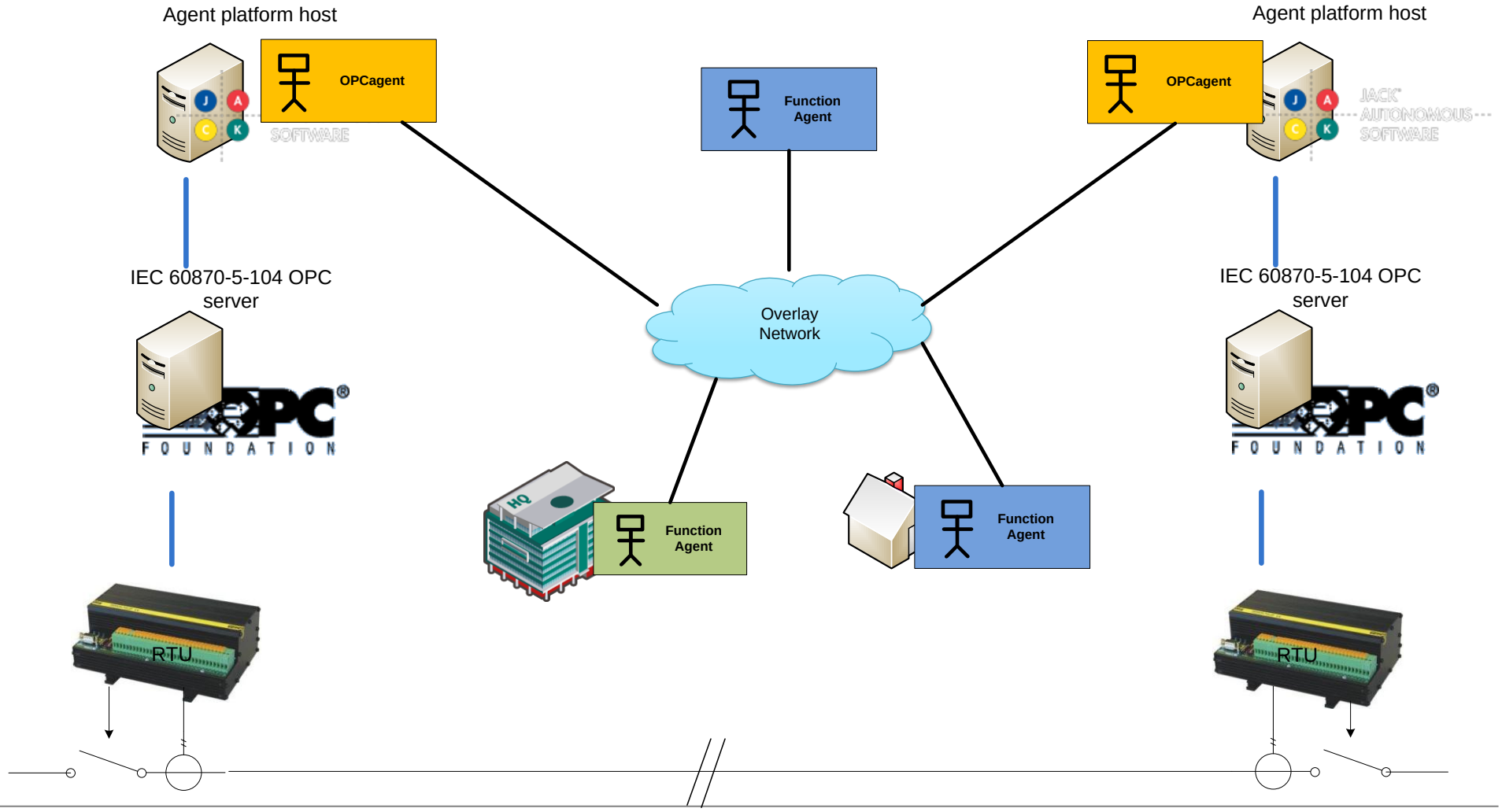
ROYAL INSTITUTE OF TECHNOLOGY

Outline

- Lab architecture
- Recap of last lecture
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client and the OPCagent

Recap:Lab logical architecture

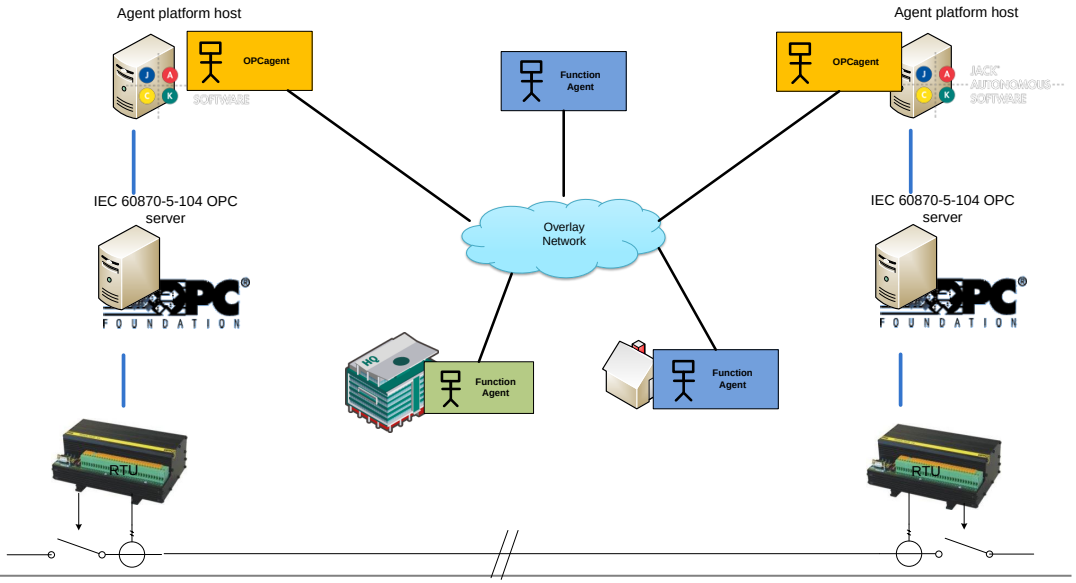
OPCagent: OPC interfacing from JACK



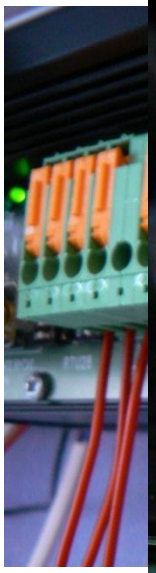
ROYAL INSTITUTE OF TECHNOLOGY

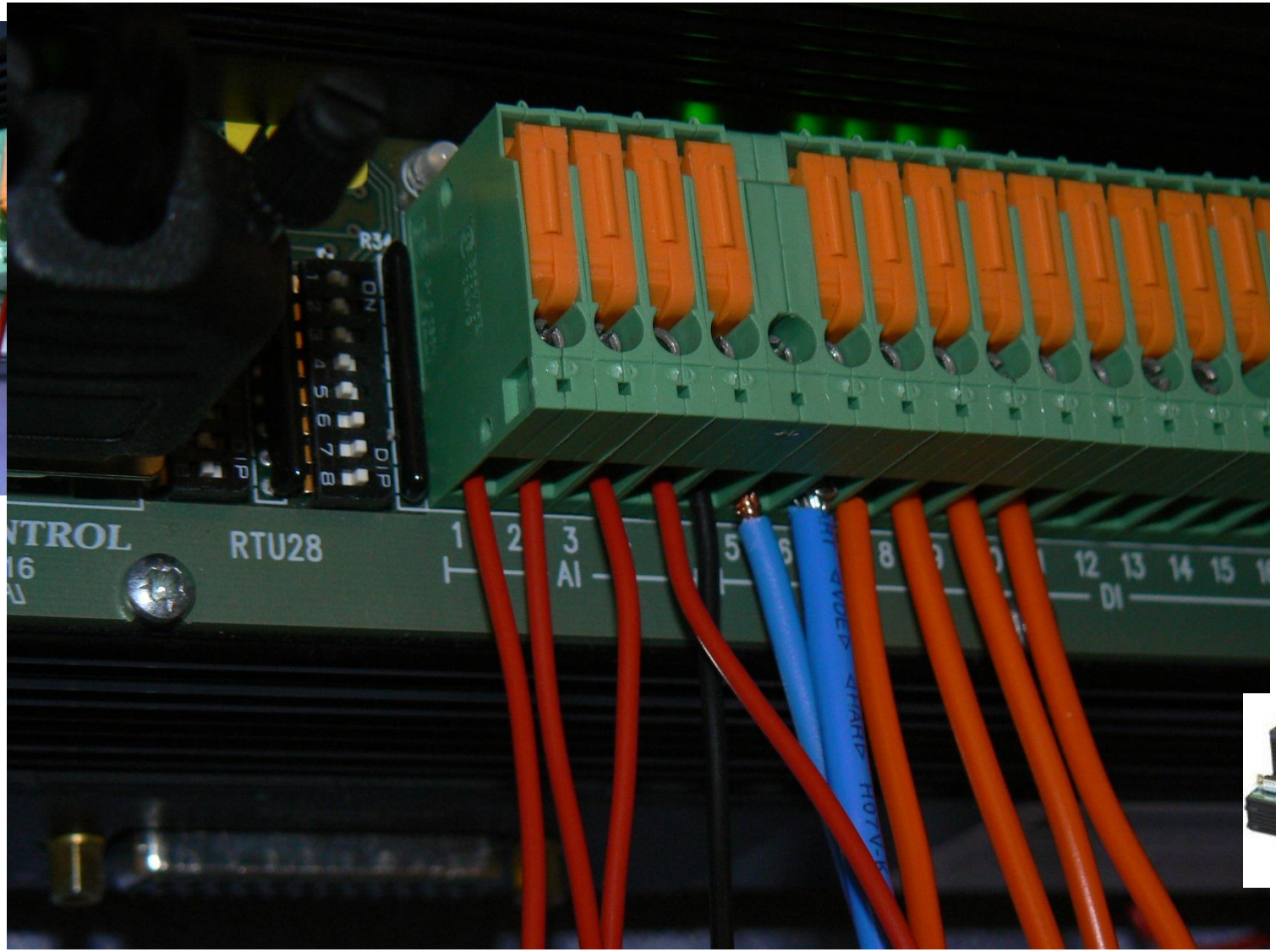
Outline

- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client

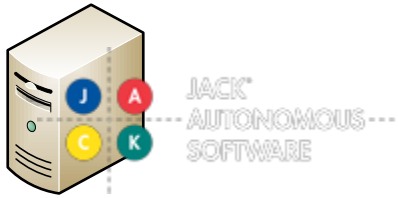


OLE for Process Control (OPC)



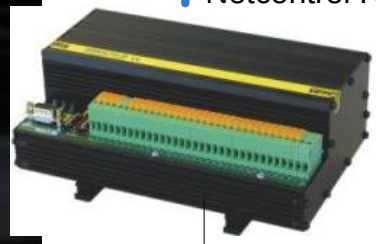


Agent platform host



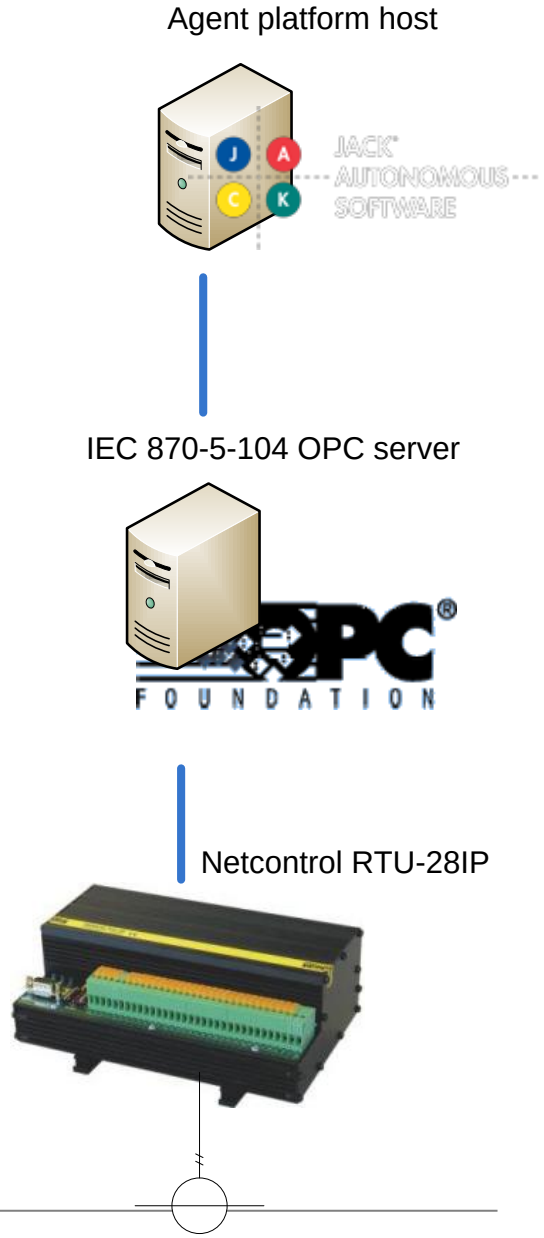
IEC 870-5-104 OPC server

Netcontrol RTU-28IP



OLE for Process Control (OPC)

- Object Linking and Embedding (OLE)
- Originally developed to provide an interface between:
  - Windows software applications and
  - Process control hardware.
- A few specifications (versions):
  - OPC Data Access (DA)
  - OPC Alarms & Events
  - OPC Data eXchange (DX)
  - OPC Commands
  - OPC XML-DA
  - OPC Unified Architecture (UA)



OPC Servers

CybServer870M-10x OPC Servers

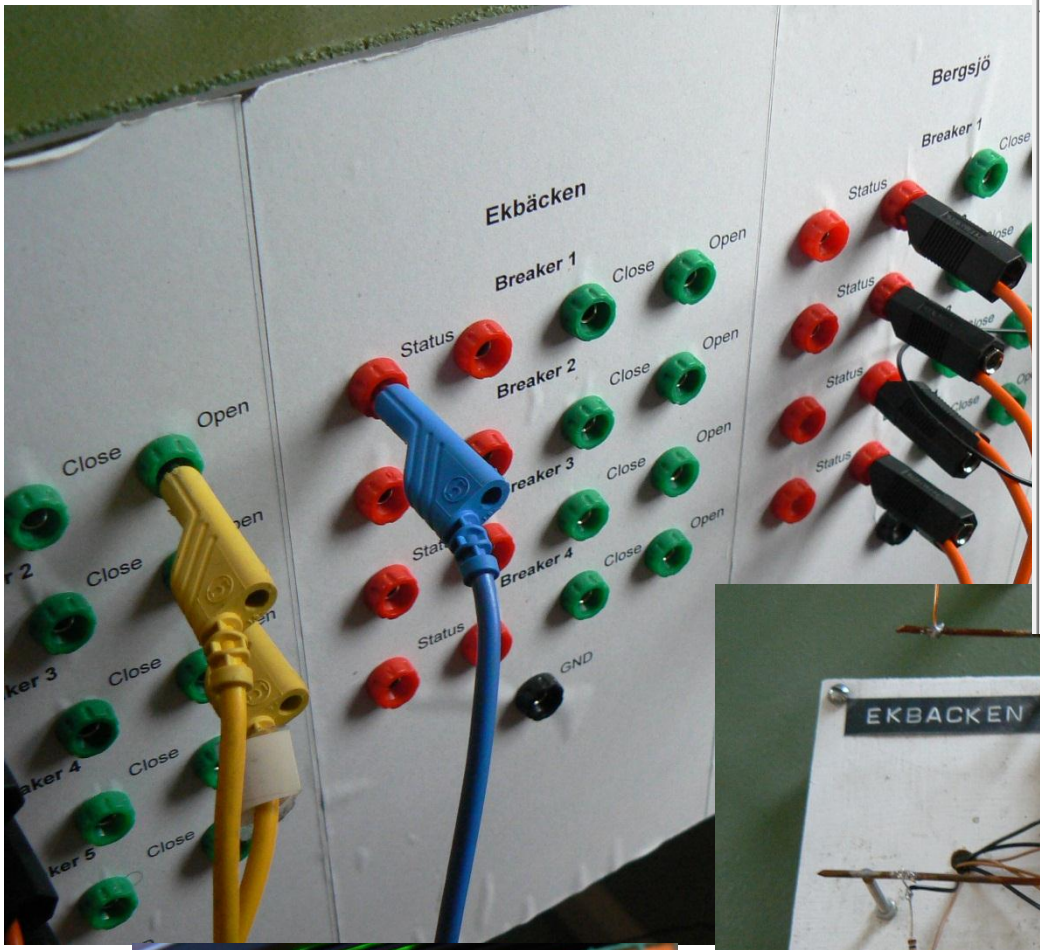
- Client to communicate with:
  - IEC 60870-5-101 (Serial)
  - IEC 60870-5-104 (IP)
- Server for OPC DA
- Makes RTU data accessible:
  - Status
  - Measurements
  - Commands
  - Map tag names to “870 addresses”

IEC 870-5-104 OPC server

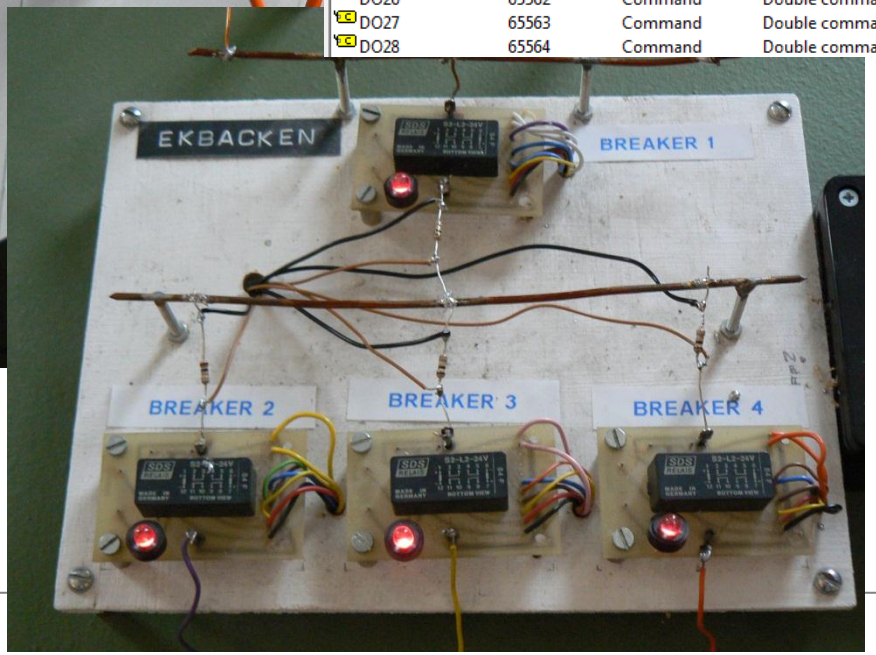
Tag	870 Address	Type	Category
CommStatus	-	System	CommStatus
CommLink	-	System	CommLink
DI5	65541	Acquisition	Boolean
DI6	65542	Acquisition	Boolean
DI7	65543	Acquisition	Boolean
DI8	65544	Acquisition	Boolean
DI9	65545	Acquisition	Boolean
DI10	65546	Acquisition	Boolean
DI11	65547	Acquisition	Boolean
DI12	65548	Acquisition	Boolean
DI13	65549	Acquisition	Boolean
DI14	65550	Acquisition	Boolean
DI15	65551	Acquisition	Boolean
DI16	65552	Acquisition	Boolean
DI17	65553	Acquisition	Boolean
DI18	65554	Acquisition	Boolean
DI19	65555	Acquisition	Boolean
DI20	65556	Acquisition	Boolean
DO21	65557	Command	Double command
DO22	65558	Command	Double command
DO23	65559	Command	Double command
DO24	65560	Command	Double command
DO25	65561	Command	Double command
DO26	65562	Command	Double command
DO27	65563	Command	Double command
DO28	65564	Command	Double command

OLE for Process Control (OPC)

ROYAL INSTITUTE OF TECHNOLOGY



Tag	870 Address	Type	Category
CommStatus	-	System	CommStatus
CommLink	-	System	CommLink
DI5	65541	Acquisition	Boolean
DI6	65542	Acquisition	Boolean
DI7	65543	Acquisition	Boolean
DI8	65544	Acquisition	Boolean
DI9	65545	Acquisition	Boolean
DI10	65546	Acquisition	Boolean
DI11	65547	Acquisition	Boolean
DI12	65548	Acquisition	Boolean
DI13	65549	Acquisition	Boolean
DI14	65550	Acquisition	Boolean
DI15	65551	Acquisition	Boolean
DI16	65552	Acquisition	Boolean
DI17	65553	Acquisition	Boolean
DI18	65554	Acquisition	Boolean
DI19	65555	Acquisition	Boolean
DI20	65556	Acquisition	Boolean
DO21	65557	Command	Double command
DO22	65558	Command	Double command
DO23	65559	Command	Double command
DO24	65560	Command	Double command
DO25	65561	Command	Double command
DO26	65562	Command	Double command
DO27	65563	Command	Double command
DO28	65564	Command	Double command



Agent platform host

JACK AUTONOMOUS SOFTWARE

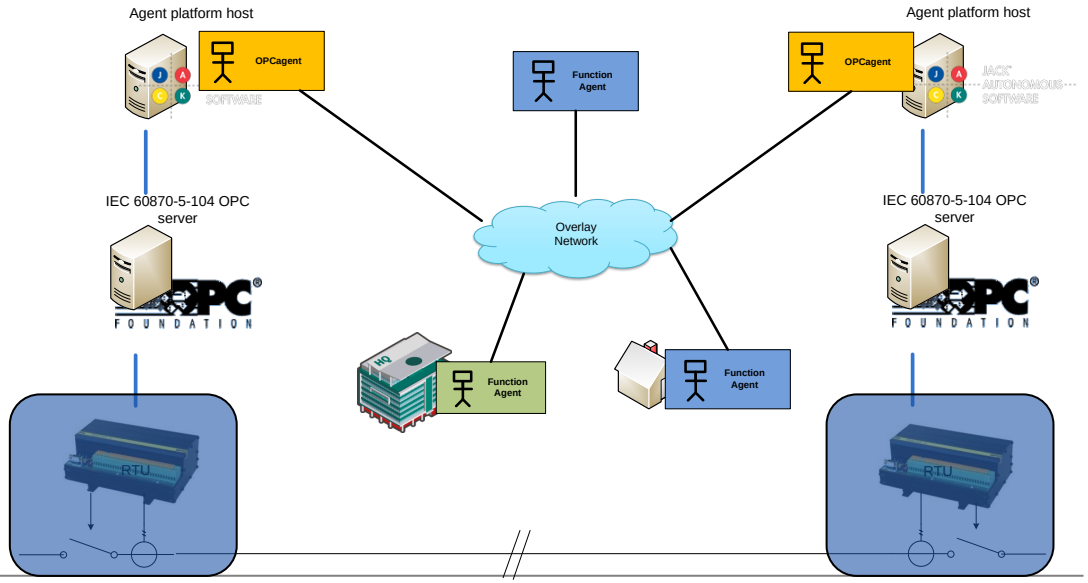
IEC 870-5-104 OPC server

Netcontrol RTU-28IP

ROYAL INSTITUTE OF TECHNOLOGY

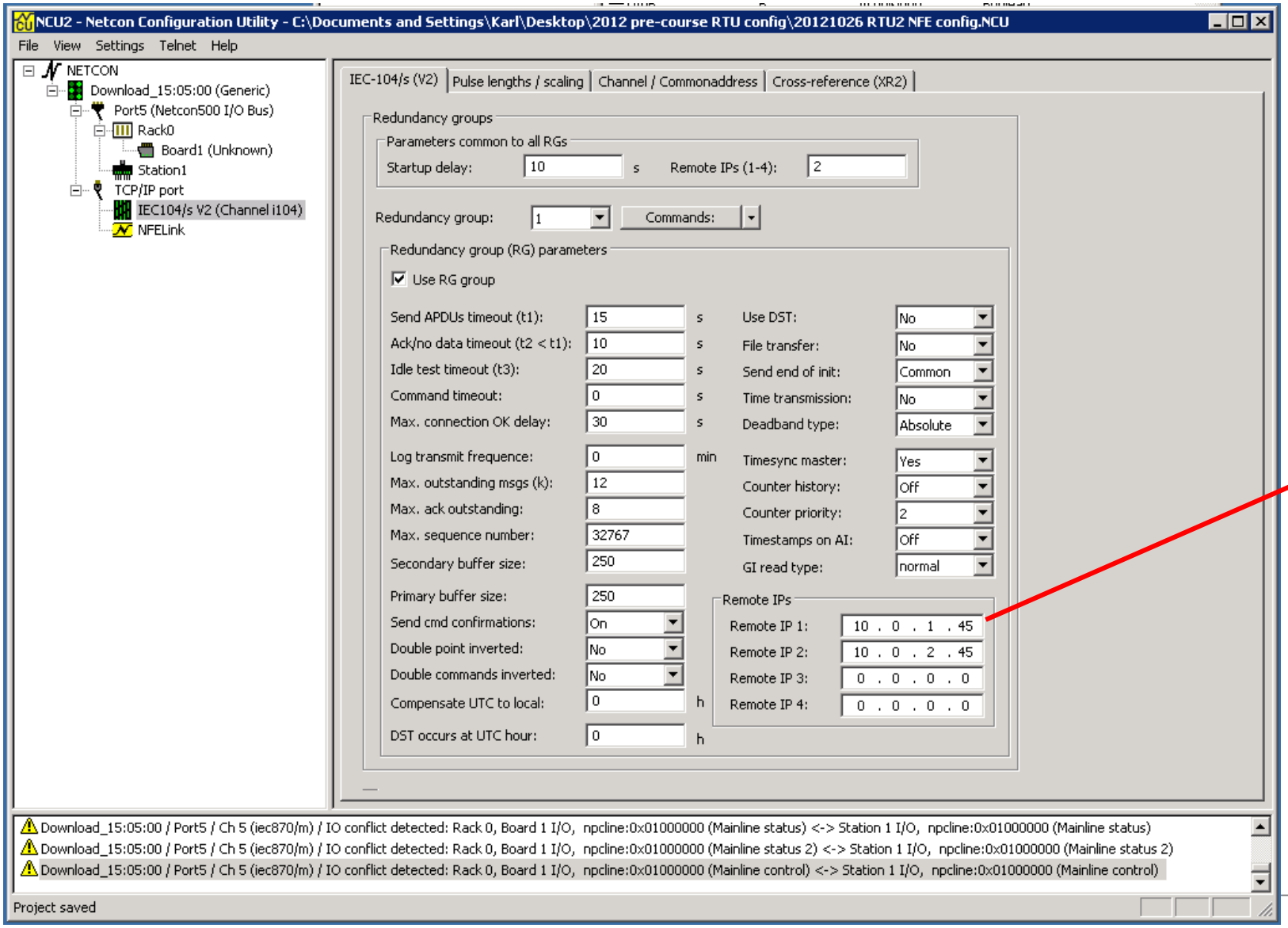
Outline

- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client

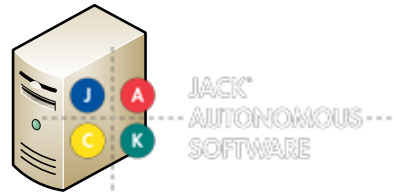


RTU configuration

IEC 60870-5-104 communications config

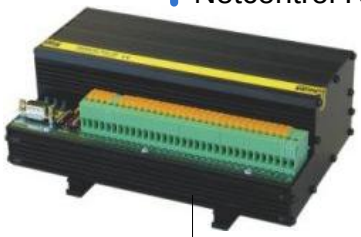


Agent platform host



IEC 870-5-104 OPC server

Netcontrol RTU-28IP



RTU configuration

IEC 60870-5-104 signal mapping

NCU2 - Netcon Configuration Utility - C:\Documents and Settings\Karl\Desktop\2012 pre-course RTU config\20121026 RTU2 NFE config.NCU

File View Settings Telnet Help

- NETCON
  - Download_15:05:00 (Generic)
    - Port5 (Netcon500 I/O Bus)
      - Rack0
        - Board1 (Unknown)
      - Station1
    - TCP/IP port
      - IEC104/s V2 (Channel i104
      - NFELink

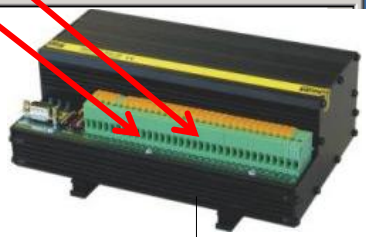
IEC-104/s (V2) | Pulse lengths / scaling | Channel / Commonaddress | Cross-reference (XR2)

Redundancy group (RG): 1

Host address format (IOA): Structured / Unstructured / Hexadecimal

	Use	Host type	Host	Native	Count	Invert	Double
1				Channel 5 (Netcon500 ...			
2	☐	sc (single command)	055:000.000.001	0x01000C00	1	☐	☐
3	☐	dp (double-point indication)	055:000.000.002	0x01000800	1	☐	
4	☑	sp (single-point indication)	055:000.000.100	0x01000000	2	☐	
5	☐	sc (single command)	055:000.000.003	0x01000000	1	☐	☐
6	☐	nv (measurand, normalized)	055:000.000.004	1:000.000.001	1		
7	☐	nv (measurand, normalized)	055:000.000.005	1:000.000.002	1		
8	☐	nv (measurand, normalized)	055:000.000.006	1:000.000.003	1		
9	☐	nv (measurand, normalized)	055:000.000.007	1:000.000.004	1		
10	☐	sp (single-point indication)	055:000.000.008	1:000.000.005	1	☐	
11	☐	sc (single command)	055:000.000.009	1:000.000.021	1	☐	☐
12	☐	sc (single command)	055:000.000.010	0x01FA0000	1	☐	☐

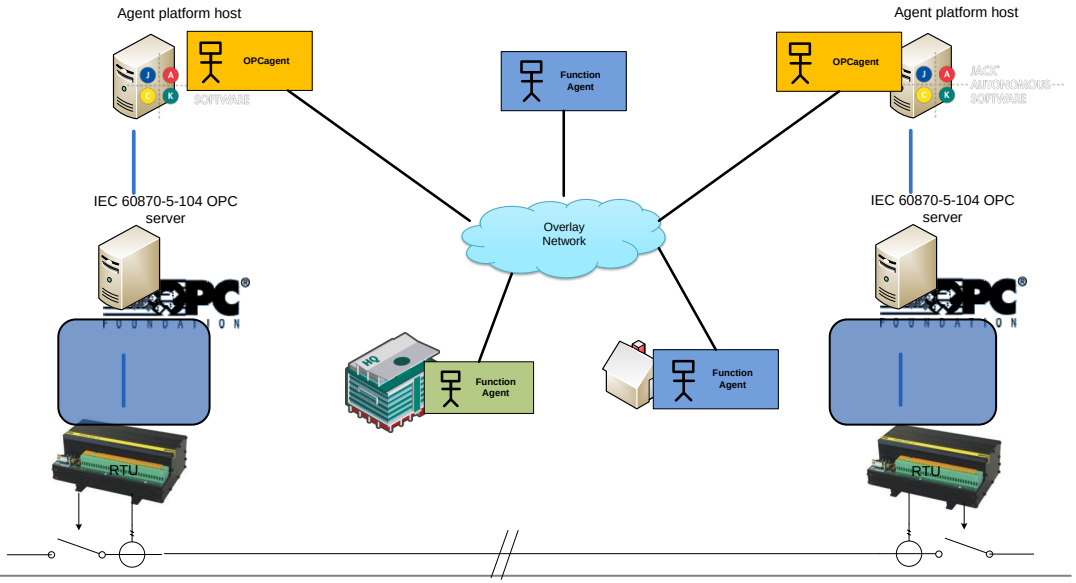
- Channel 5 (Netcon500 I/O Bus
  - Channel I/O
  - Rack 0
    - Board1 (Unknown)
      - Board I/O
      - av (int)
      - si
      - sc



ROYAL INSTITUTE OF TECHNOLOGY

Outline

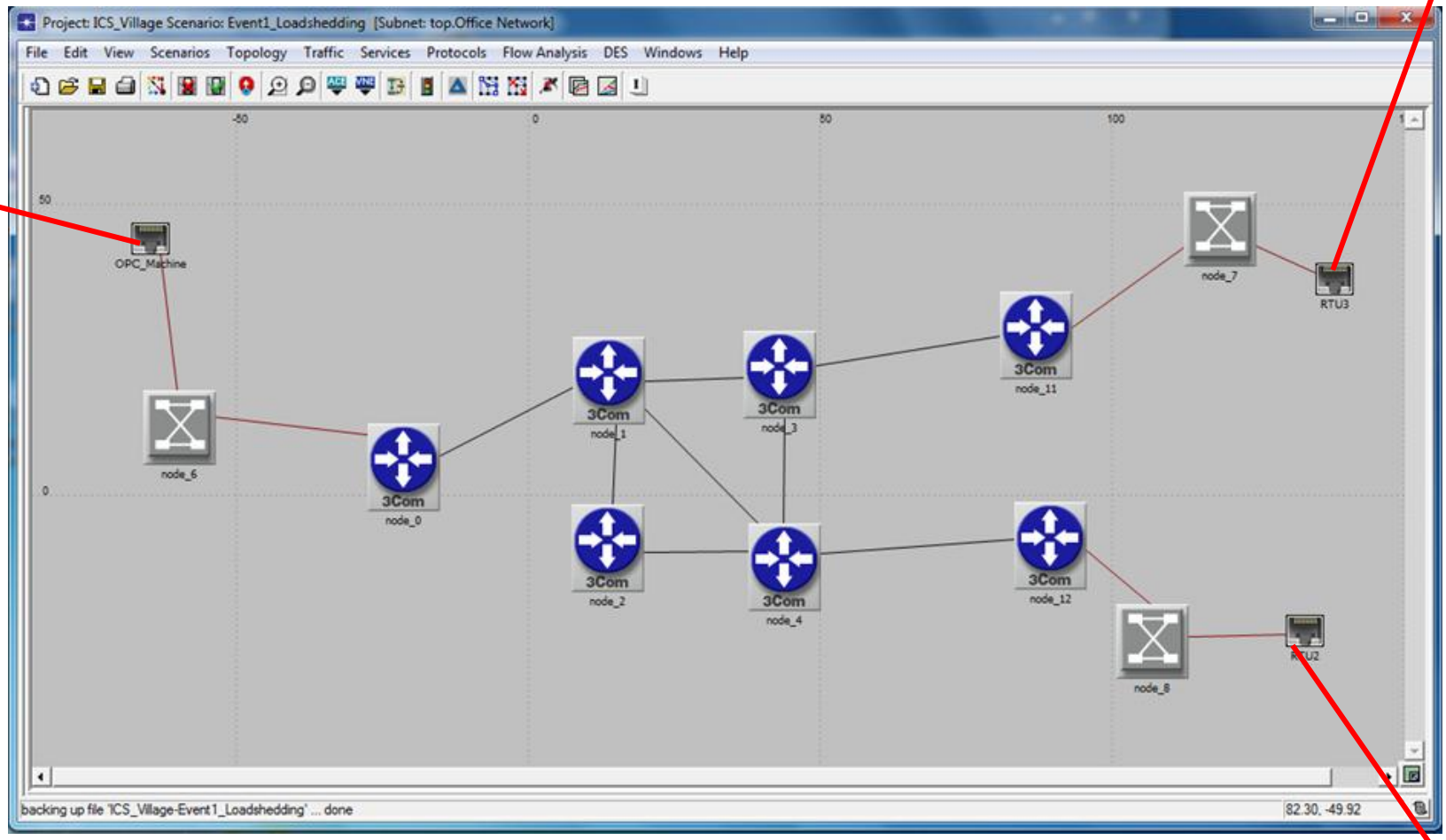
- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client



WAN connectivity

WAN emulation through OPNET

IEC 870-5-104 OPC server

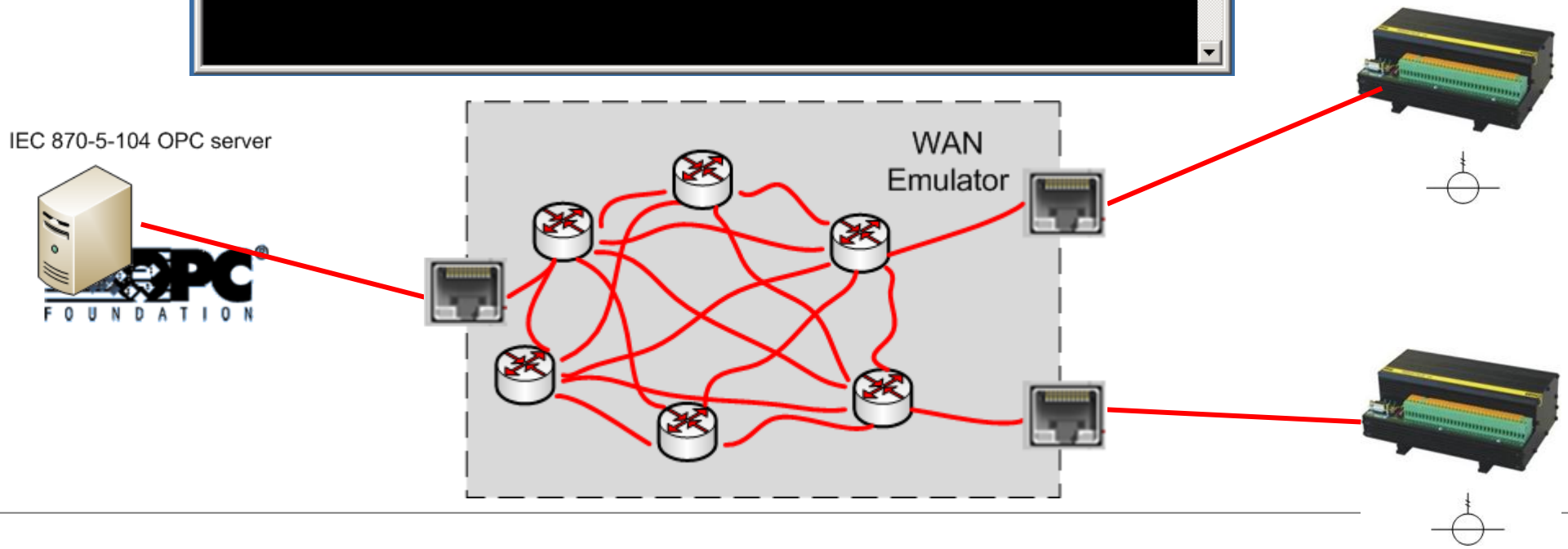


ROYAL INSTITUTE OF TECHNOLOGY

WAN connectivity

Check conectivity over IP – Using ”Ping”

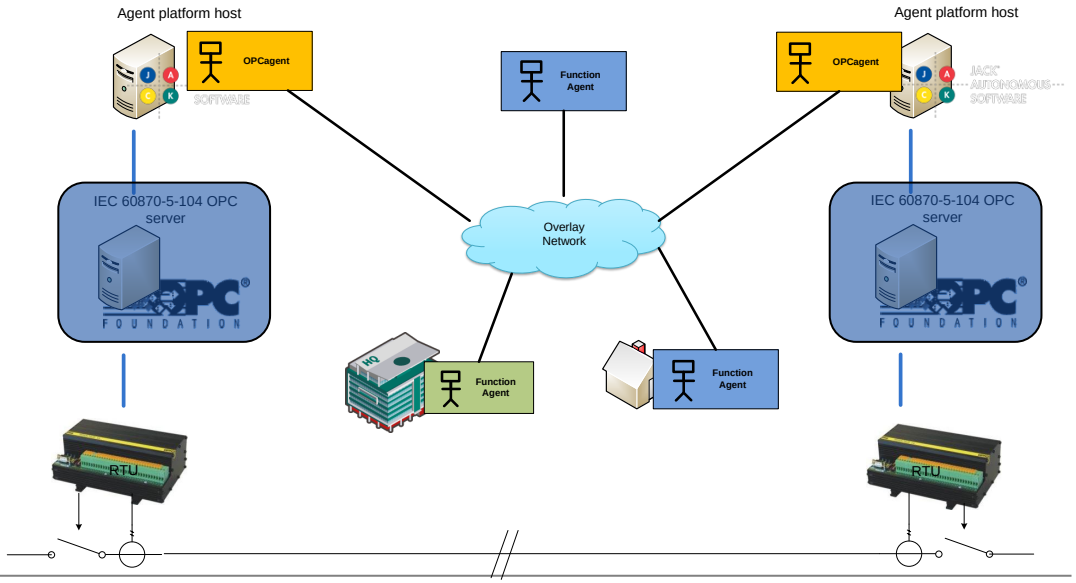
```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Karl>ping 10.0.3.8

Pinging 10.0.3.8 with 32 bytes of data:

Reply from 10.0.3.8: bytes=32 time=120ms TTL=251
Reply from 10.0.3.8: bytes=32 time=56ms TTL=251
Reply from 10.0.3.8: bytes=32 time=56ms TTL=251
Reply from 10.0.3.8: bytes=32 time=41ms TTL=251

Ping statistics for 10.0.3.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 41ms, Maximum = 120ms, Average = 68ms

C:\Documents and Settings\Karl>_
```



ROYAL INSTITUTE OF TECHNOLOGY

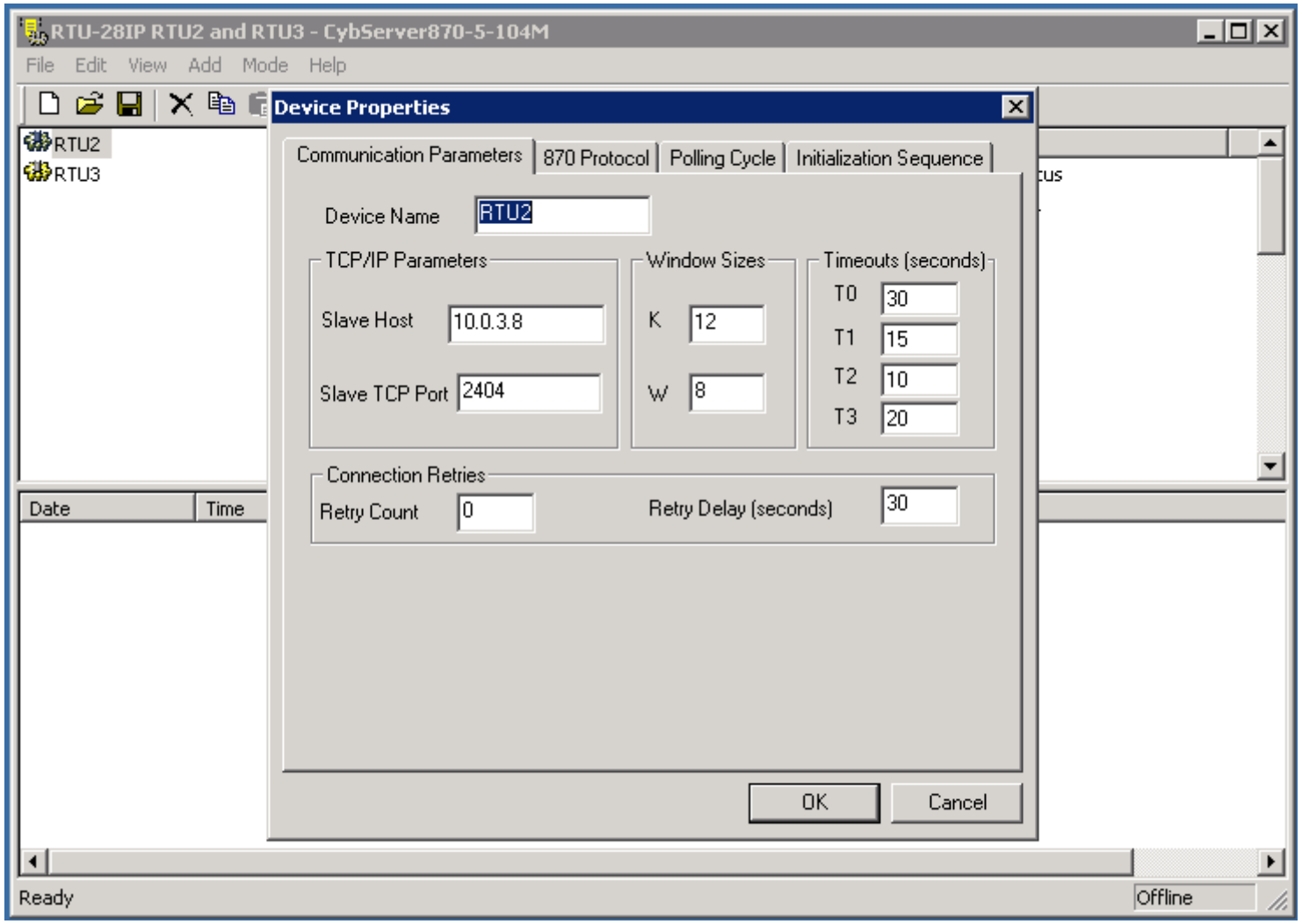
Outline

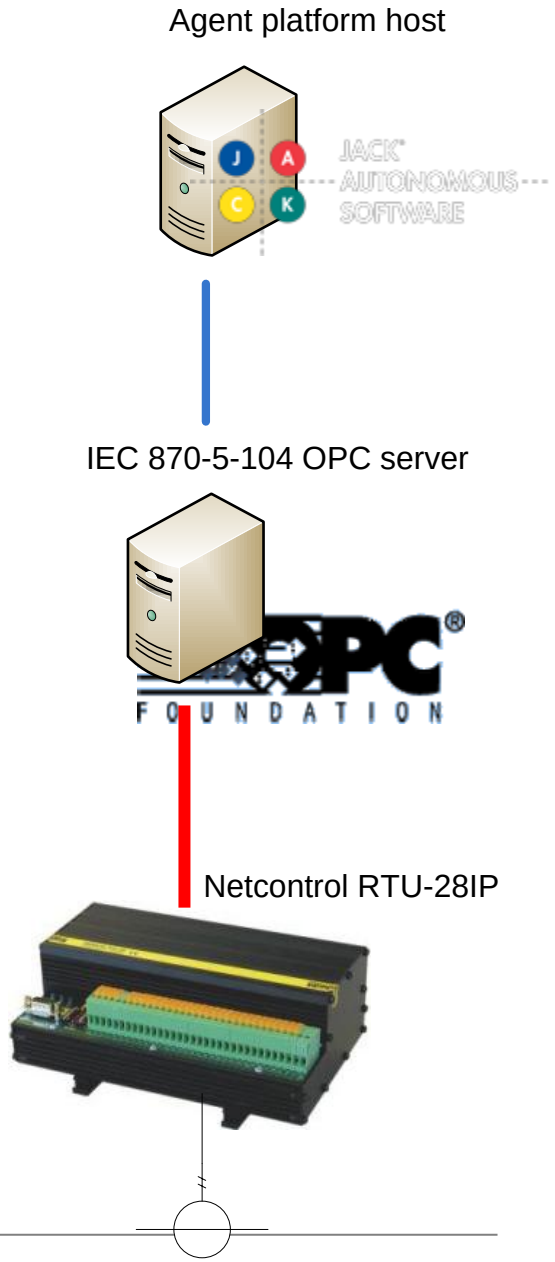
- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client



OPC server configuration

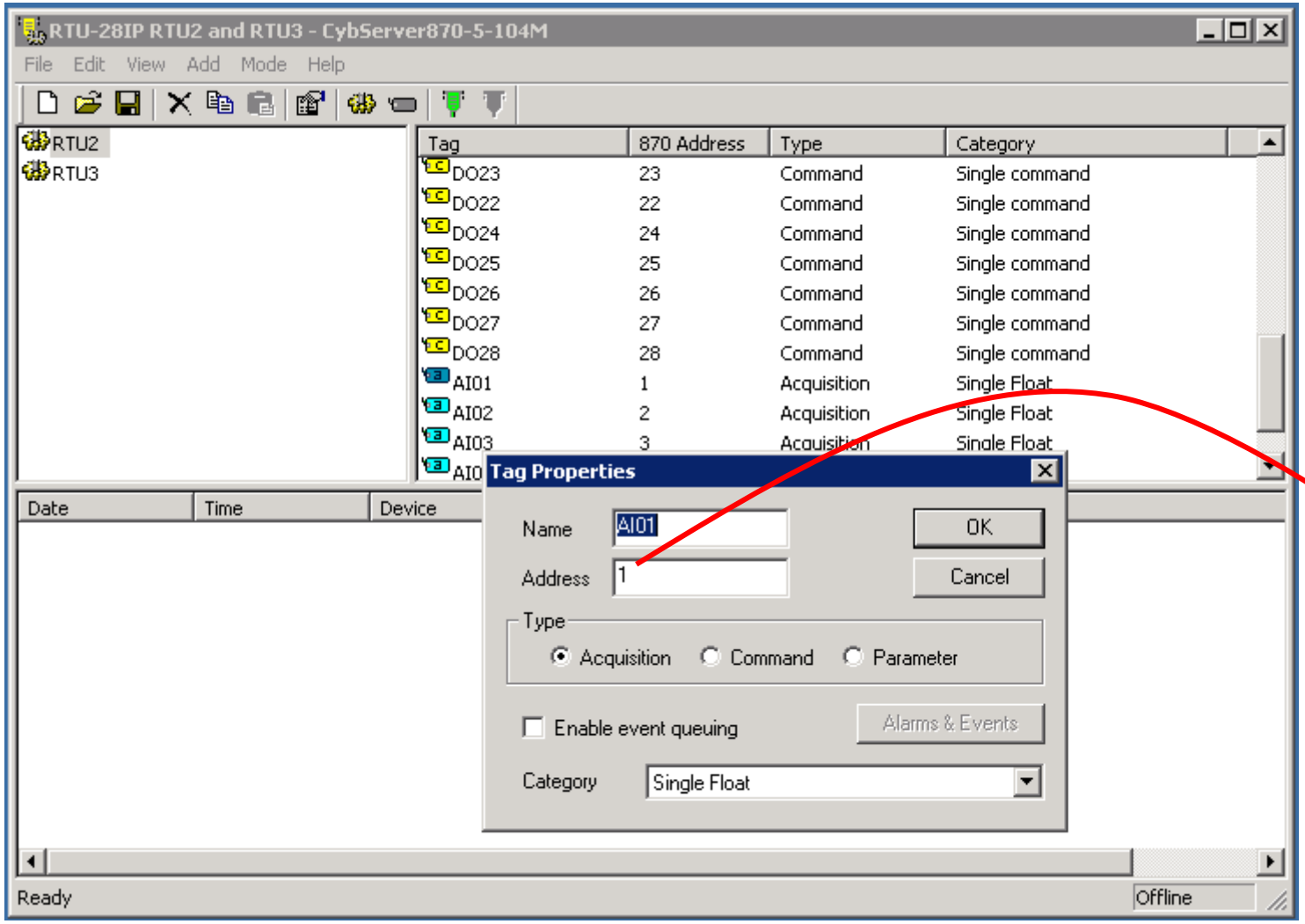
Creating new device profile

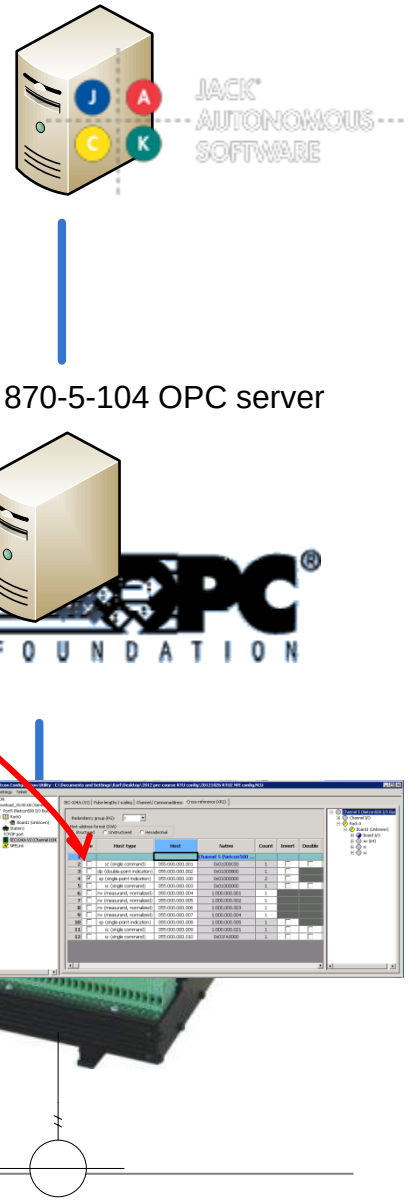




OPC server configuration

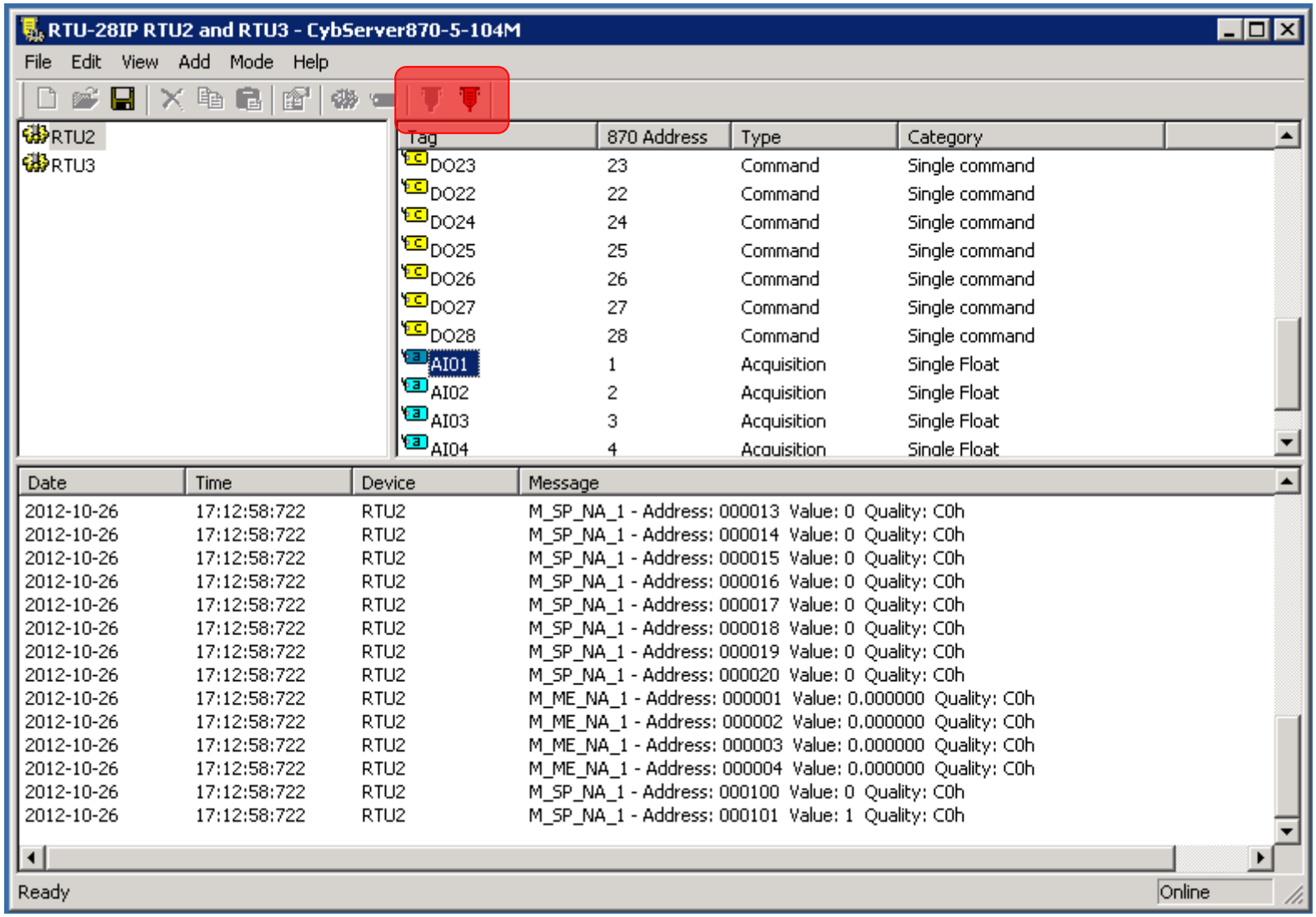
Mapping OPC tag names to IEC 60870-5 addresses

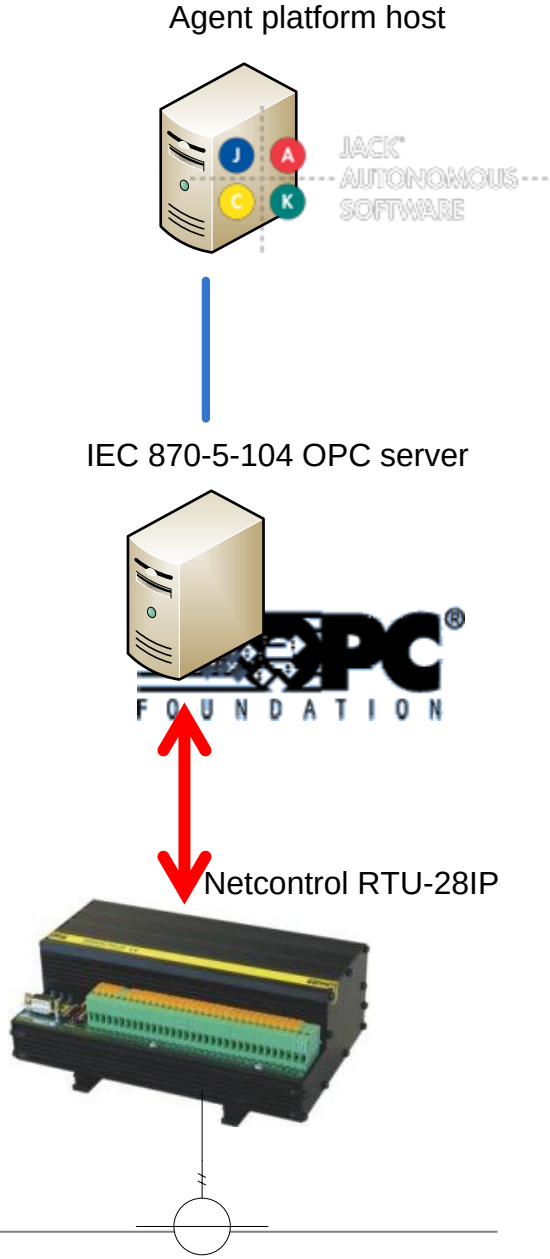




OPC server configuration

Starting the OPC server

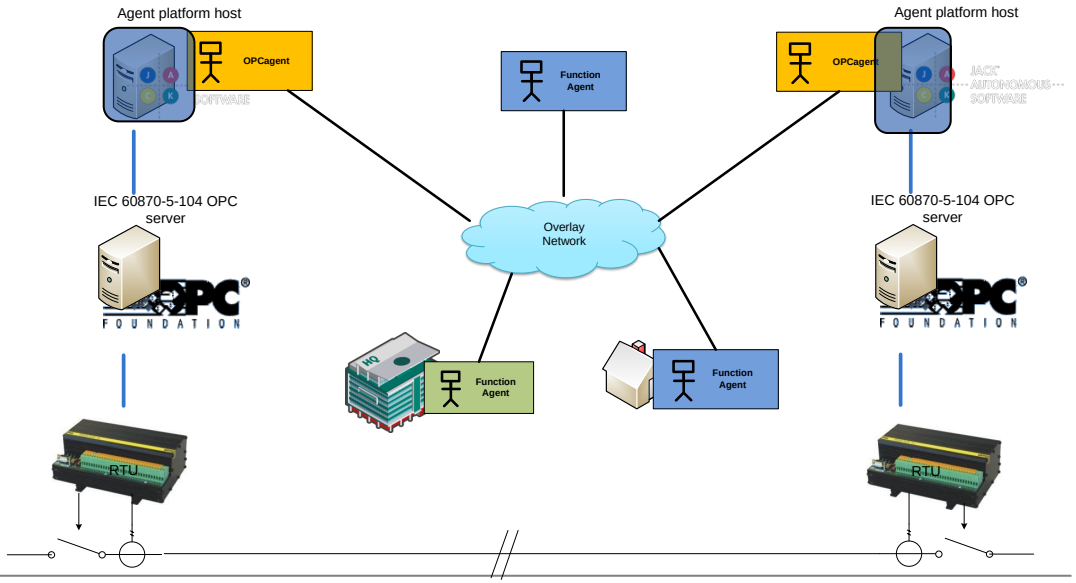




ROYAL INSTITUTE OF TECHNOLOGY

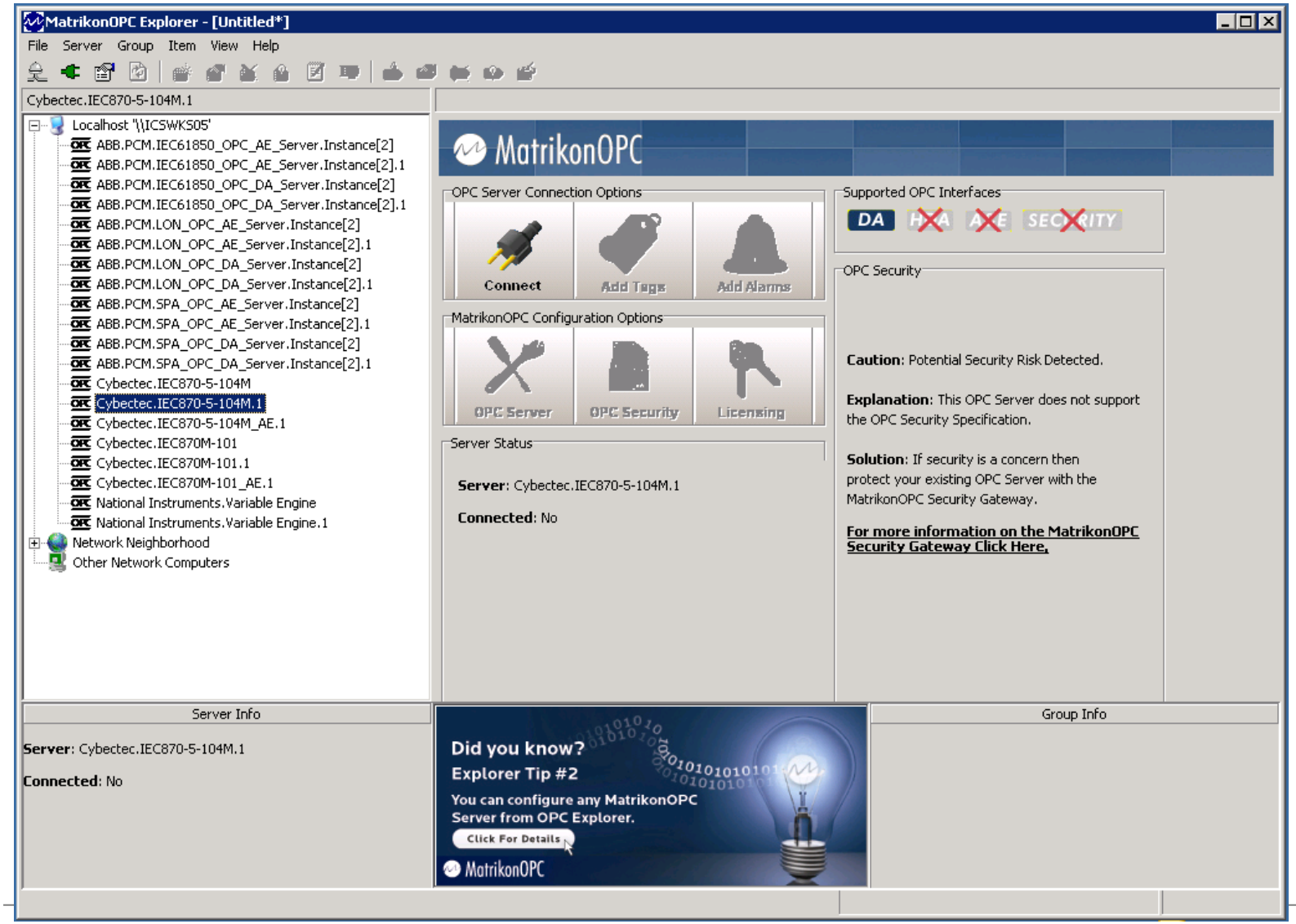
Outline

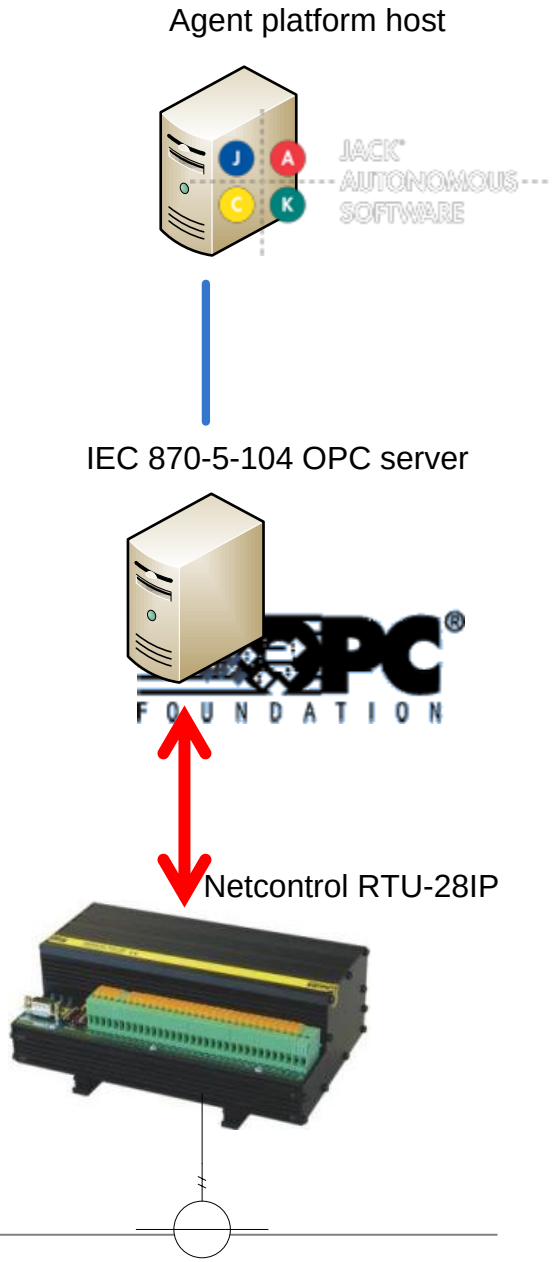
- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client



OPC client connection

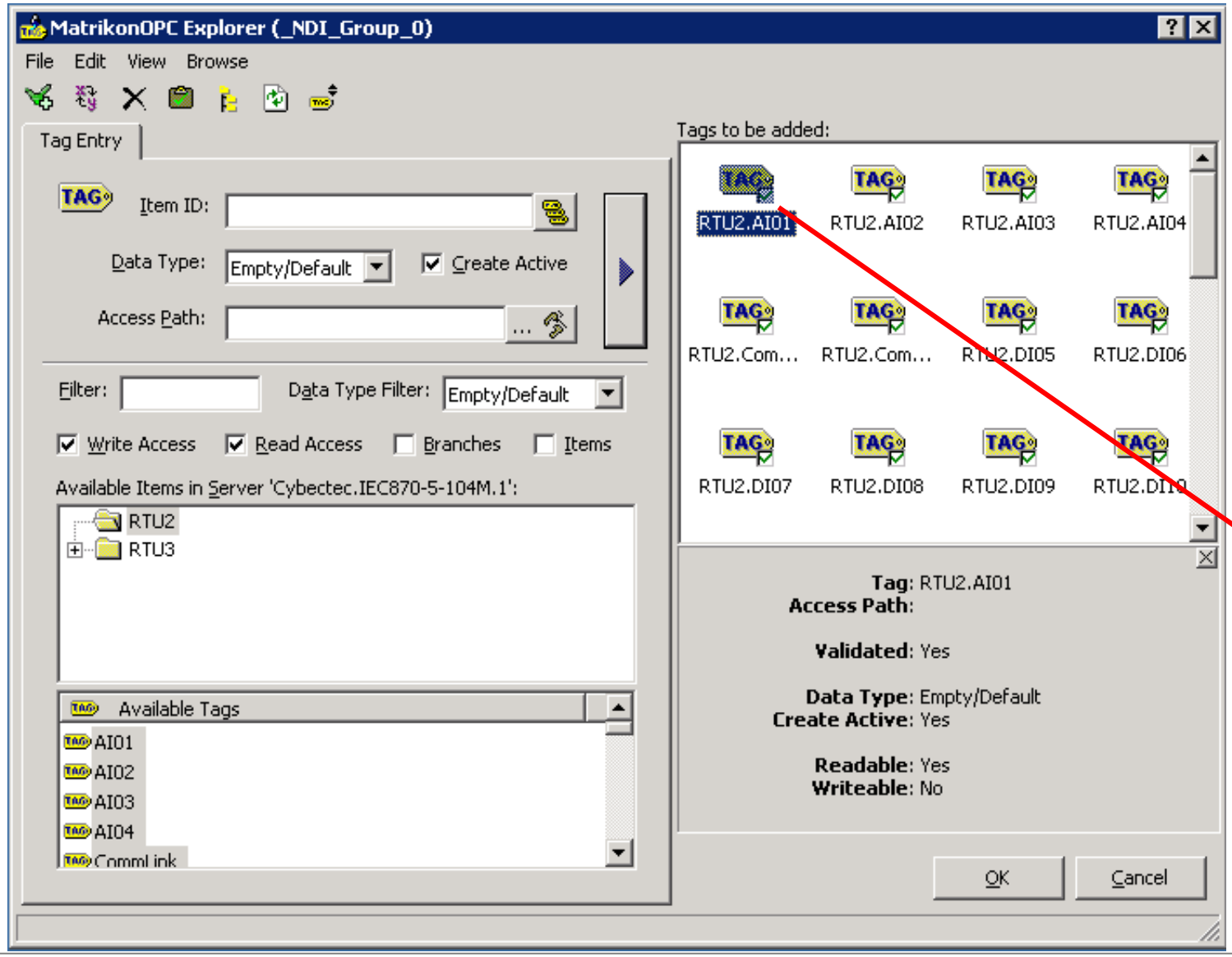
Connecting to your OPC server

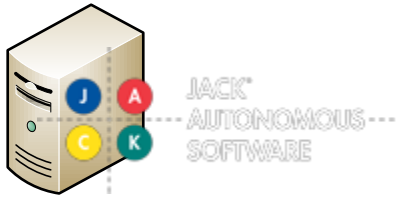




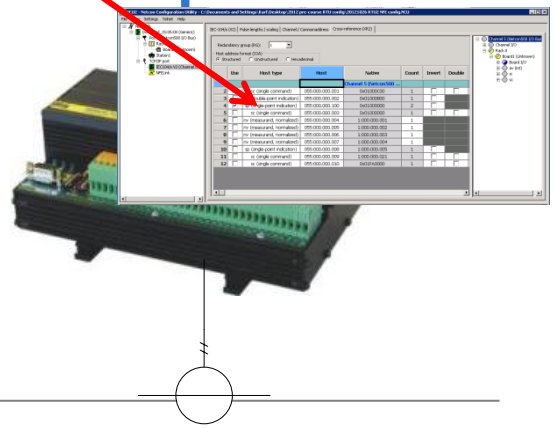
OPC client connection

Connecting to your OPC server





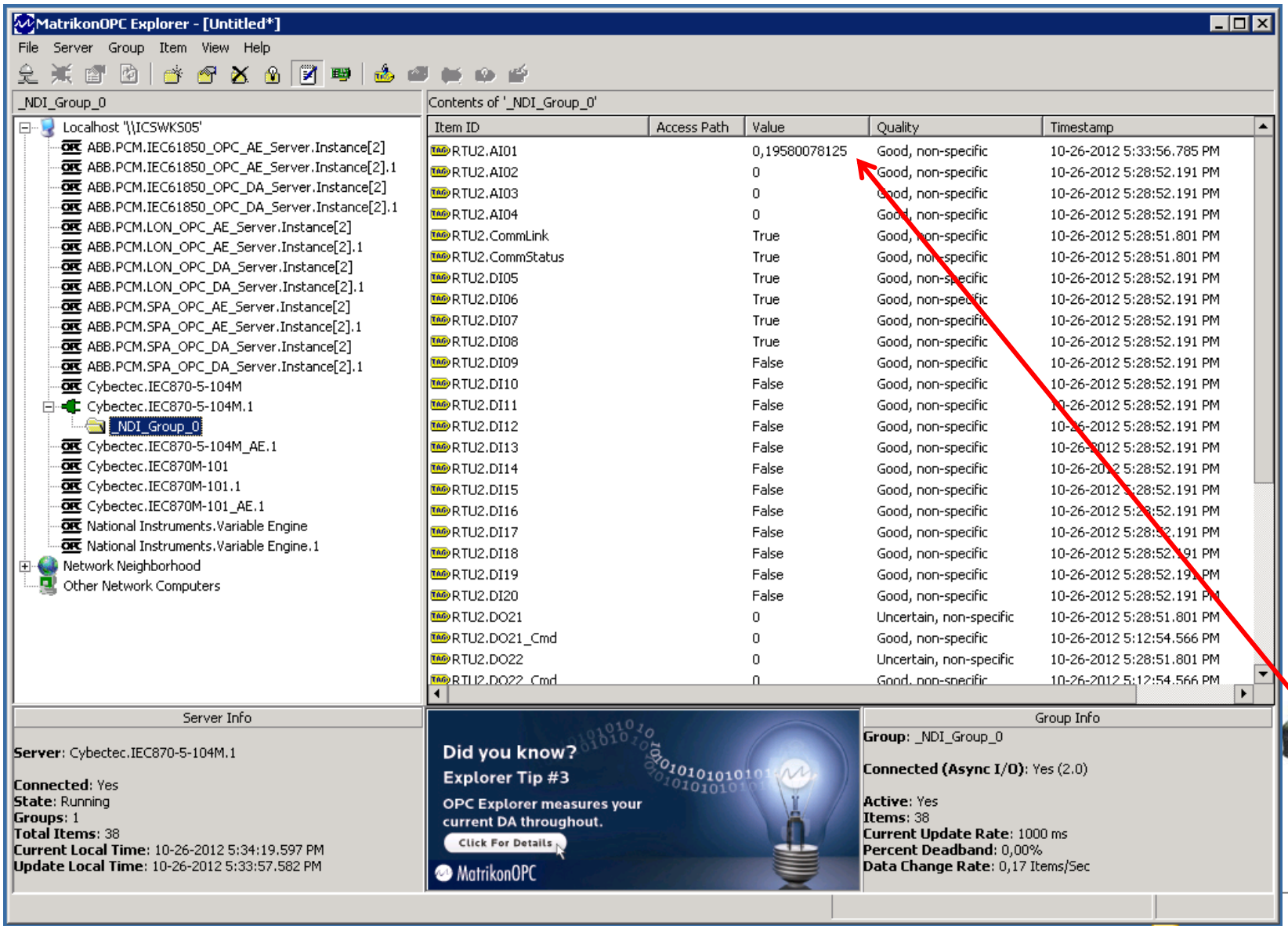
IEC 870-5-104 OPC server



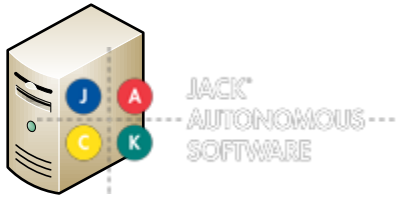
ROYAL INSTITUTE OF TECHNOLOGY

OPC client connection

Connecting to your OPC server

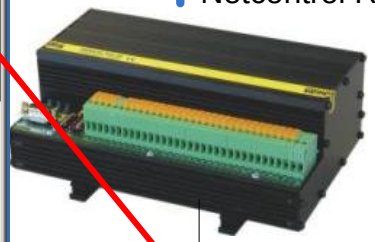


Agent platform host



IEC 870-5-104 OPC server

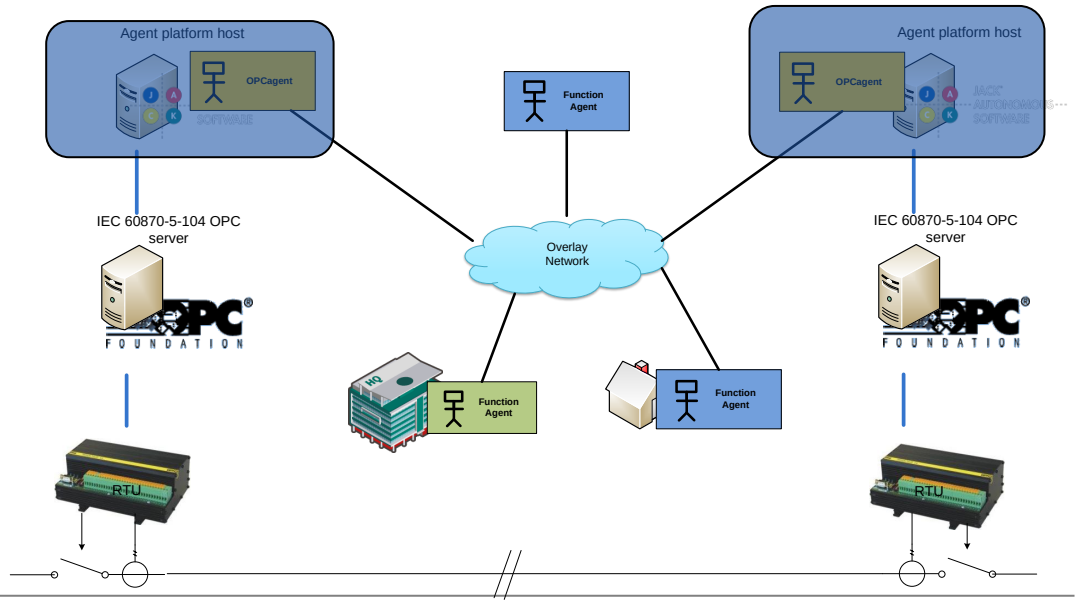
Netcontrol RTU-28IP



ROYAL INSTITUTE OF TECHNOLOGY

Outline

- Lab architecture
- Recap of JACK interfacing with OPC – the OPCagent
- RTU configuration
- WAN connectivity
- OPC server configuration
- OPC client connection
- Java OPC client



Java OPC client

JEasyOPC Java class library

- Open-source java class library
- Interface with OPC servers

```
import javafish.clients.opc.JCustomOpc;
import javafish.clients.opc.JEasyOpc;
import javafish.clients.opc.JOpc;
import javafish.clients.opc.asynch.AsynchEvent;
import javafish.clients.opc.asynch.OpcAsynchGroupListener;
import javafish.clients.opc.browser.JOpcBrowser;
import javafish.clients.opc.component.OpcGroup;
import javafish.clients.opc.component.OpcItem;
```

Initialization:

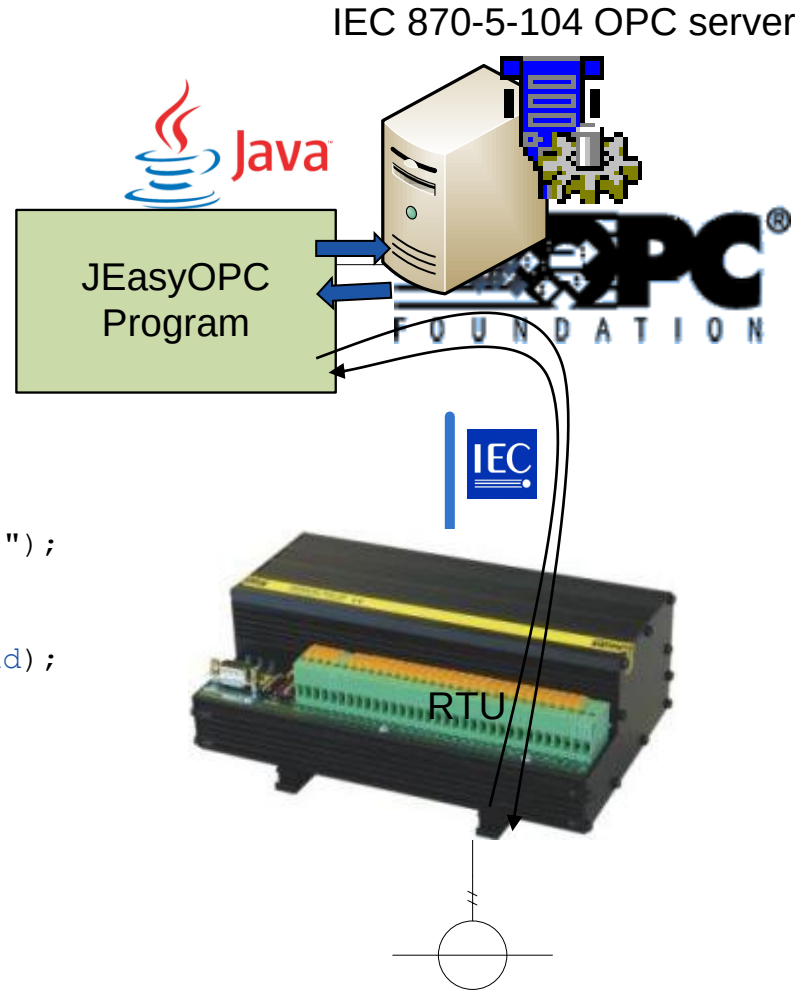
```
jopc_meas = new JEasyOpc(hostName, serverName, groupName + "_meas");
```

Reading:

```
gotItem = jopc_meas.synchReadItem(statusGroup, (OpcItem)ItemToRead);
```

Writing:

```
jopc_command.synchWriteItem(commandGroup, item);
```



ROYAL INSTITUTE OF TECHNOLOGY

Java OPC client

Making Java speak OPC

- Initialise a new JEasyOpc object
- Create groups for different readings

```
jopc_status = new JEasyOpc("localhost", "Matrikon.OPC.Simulation", "JOPC1");

//commandGroup = new OpcGroup("commands", true, 20, 0.0f);
//measGroup = new OpcGroup("meas", true, 20, 0.0f);
statusGroup = new OpcGroup("status", true, 20, 0.0f);
```

ROYAL INSTITUTE OF TECHNOLOGY

Java OPC client

Making Java speak OPC

- Browsing the items

```
JOpcBrowser jbrowser = new JOpcBrowser(hostName, serverName, groupName);
```

ROYAL INSTITUTE OF TECHNOLOGY

Java OPC client

Making Java speak OPC

- Synchronous reading

```
Iterator itemItr = itemVector.iterator();
while(itemItr.hasNext())  {
  try {
      OpcItem gotItem = jopc_status.synchReadItem(statusGroup, (OpcItem)itemItr.next());
      if (gotItem != null) {
        Date timeNow = new Date();
        if (gotItem.getDataType() == 11) System.out.println("Read:" + gotItem.getItemName() + gotItem.ge
         else System.out.println("Read" + gotItem.getItemName() + (float) gotItem.getValue().getFloat()
      System.out.println(gotItem);

  }
  } catch (SynchReadException e) {
      e.printStackTrace();
  }
```

ROYAL INSTITUTE OF TECHNOLOGY

Java OPC client

Making Java speak OPC

- Synchronous writing

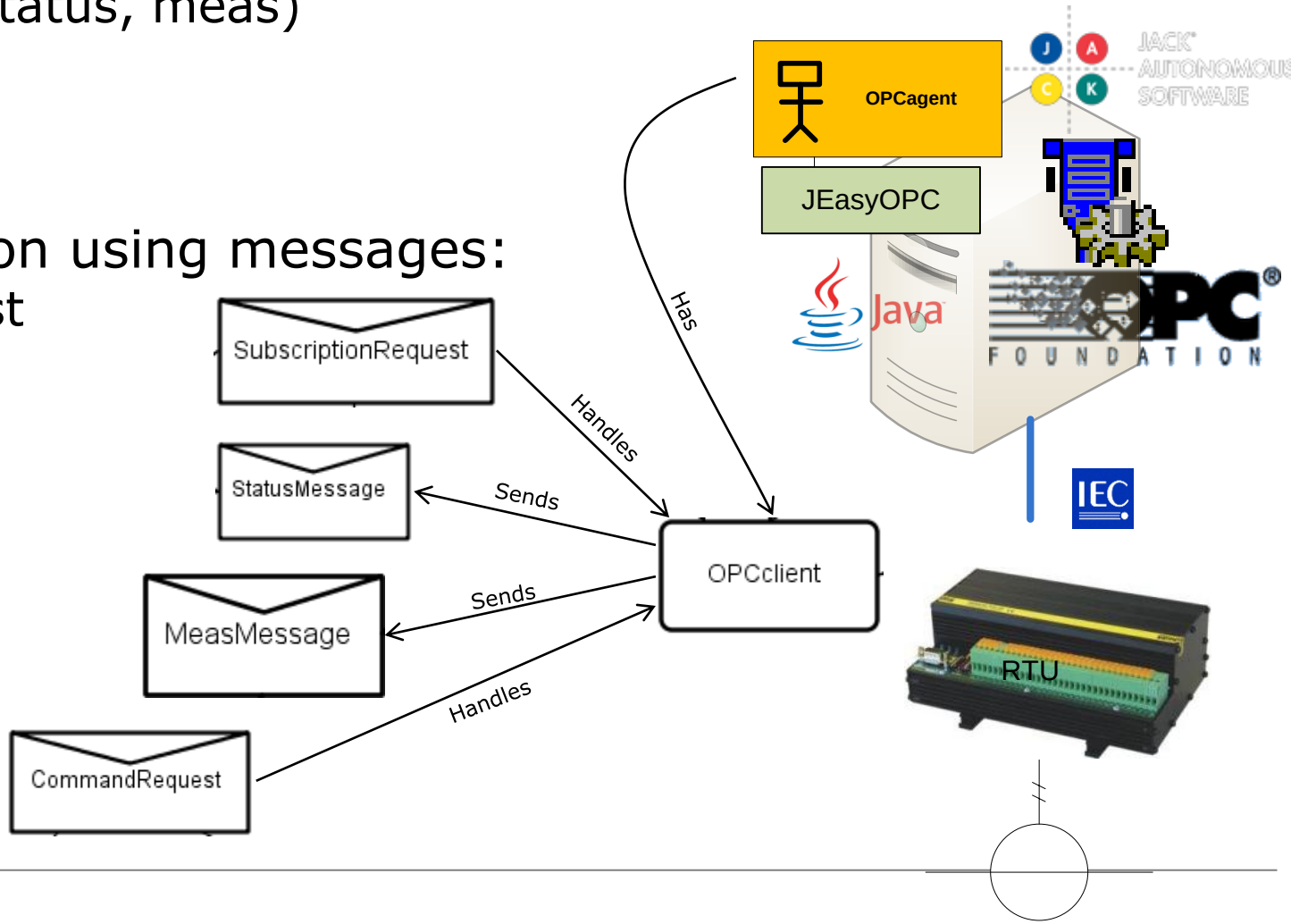
```
jopc_command.synchWriteItem(commandGroup, item);
```

Java OPC client

OPC agent - OPC interfacing from JACK

- JACK agent with **OPC client** capability:
  - Browser OPC server
  - Listen for events (status, meas)
  - Issue commands

- Agent communication using messages:
  - SubscriptionRequest
  - StatusMessage
  - MeasMessage
  - CommandRequest

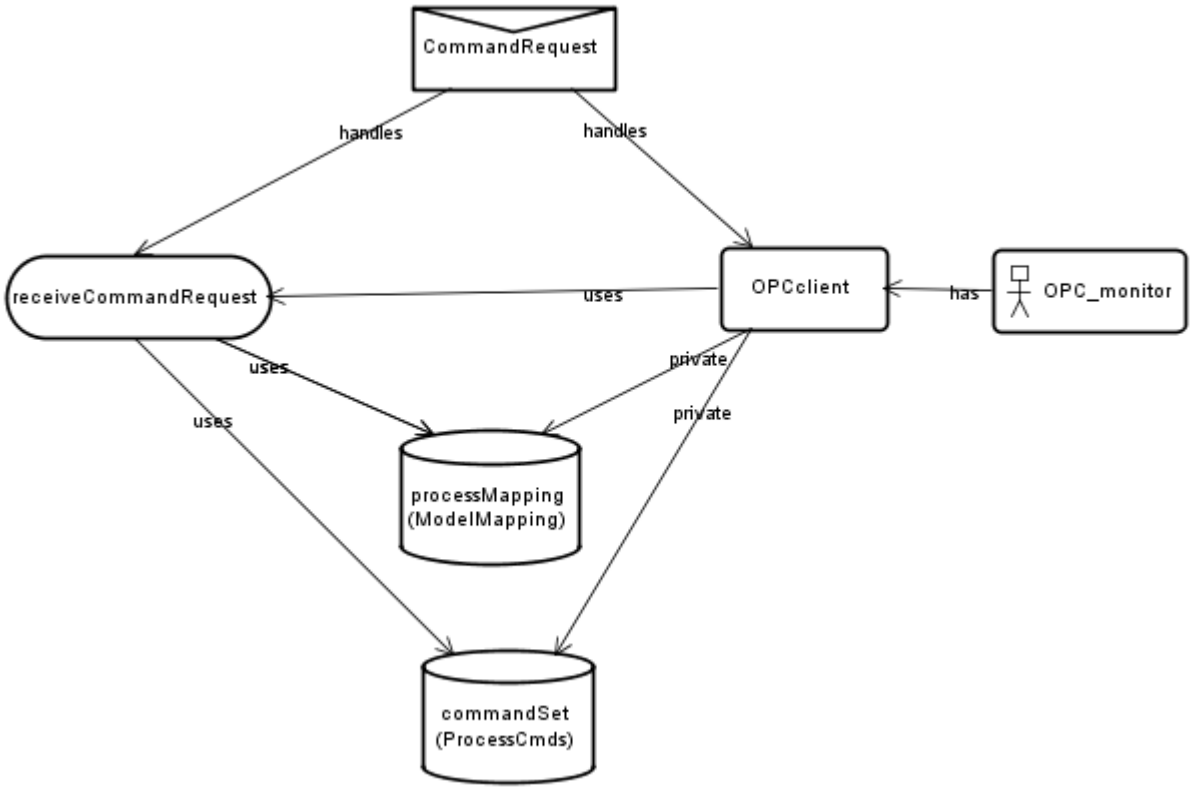


Java OPC client

OPCagent - OPC interfacing from JACK

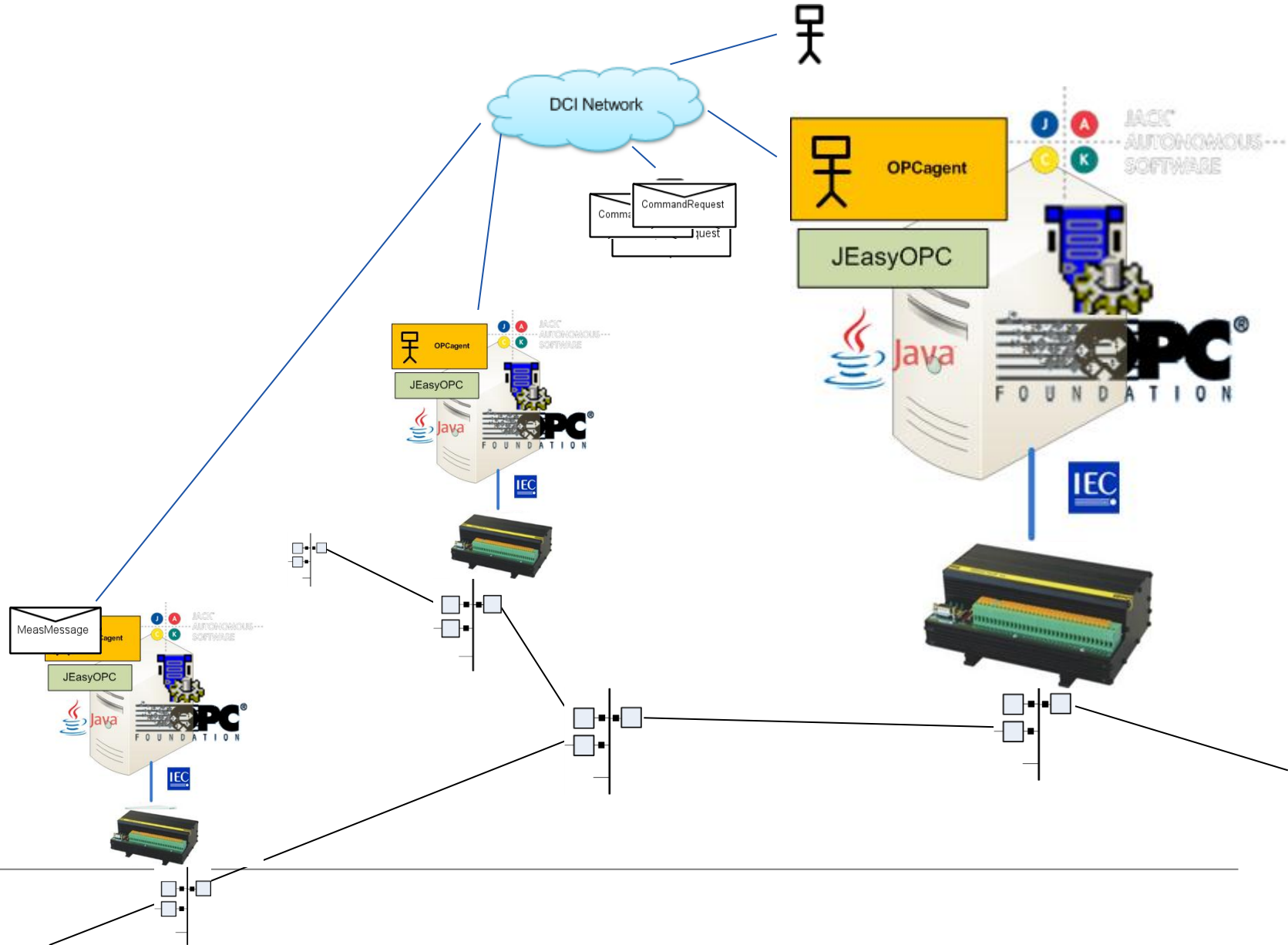
- JACK agent with **OPC client** capability:
  - Browser OPC server
  - Listen for events (status, meas)
  - Issue commands

- Agent communication using
  - SubscriptionRequest
  - StatusMessage
  - MeasMessage
  - CommandRequest



Java OPC client

OPCagent - OPC interfacing from JACK



ROYAL INSTITUTE OF TECHNOLOGY

Summary

- Followed configuration process from RTUs up to application clients.
- Got a better technical understanding of what happens at the OPCagent.
- Exercise lab follows the process of how to browse, read and write to the OPC server.