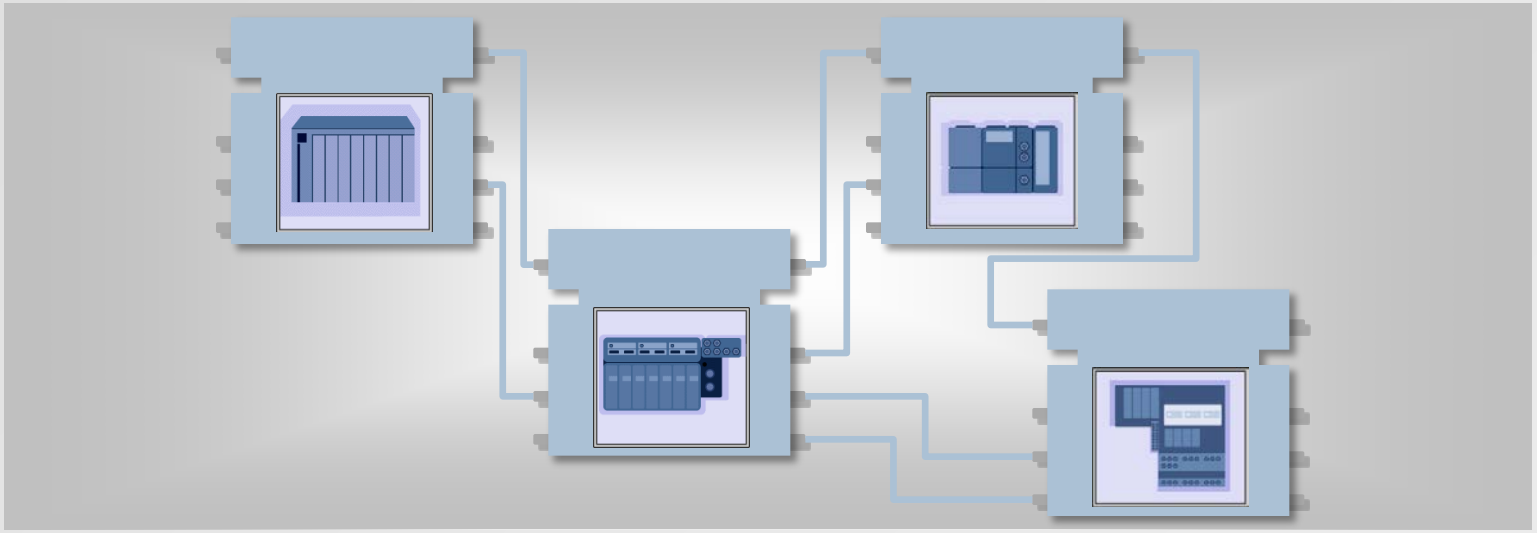
Architecture for Services Composition in OPC UA Servers using FORTE



Federico Pérez, Marga Marcos, Darío Orive

Contents

Introduction

- **Cyber-Physical Production Systems (CPPS):**
  - Computation and process for production systems
  - Collaborative entities communicating in factory automation environments
  - Industrial communications
    - Complex
    - Different solutions at the different layers
  - Middleware solutions
    - OPC UA: OPC Unified Architecture
  - Trends:
    - Open software and hardware
    - Assorted communication technologies
    - Miniaturization of the hardware (Single Board Computer – SBC)
    - Reduction of cost

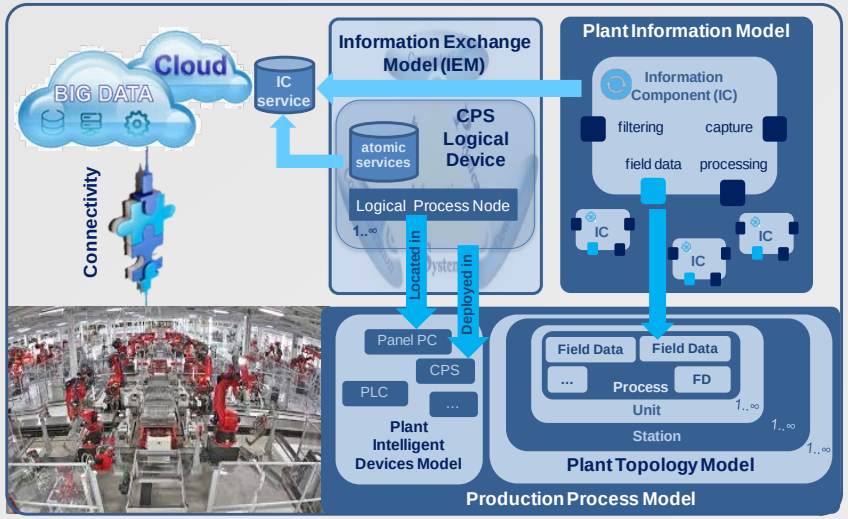
Introduction

OPC UA (Unified Architecture) is a set of specifications trying to cover real-time requirements to exchange information and use commands in industrial control.

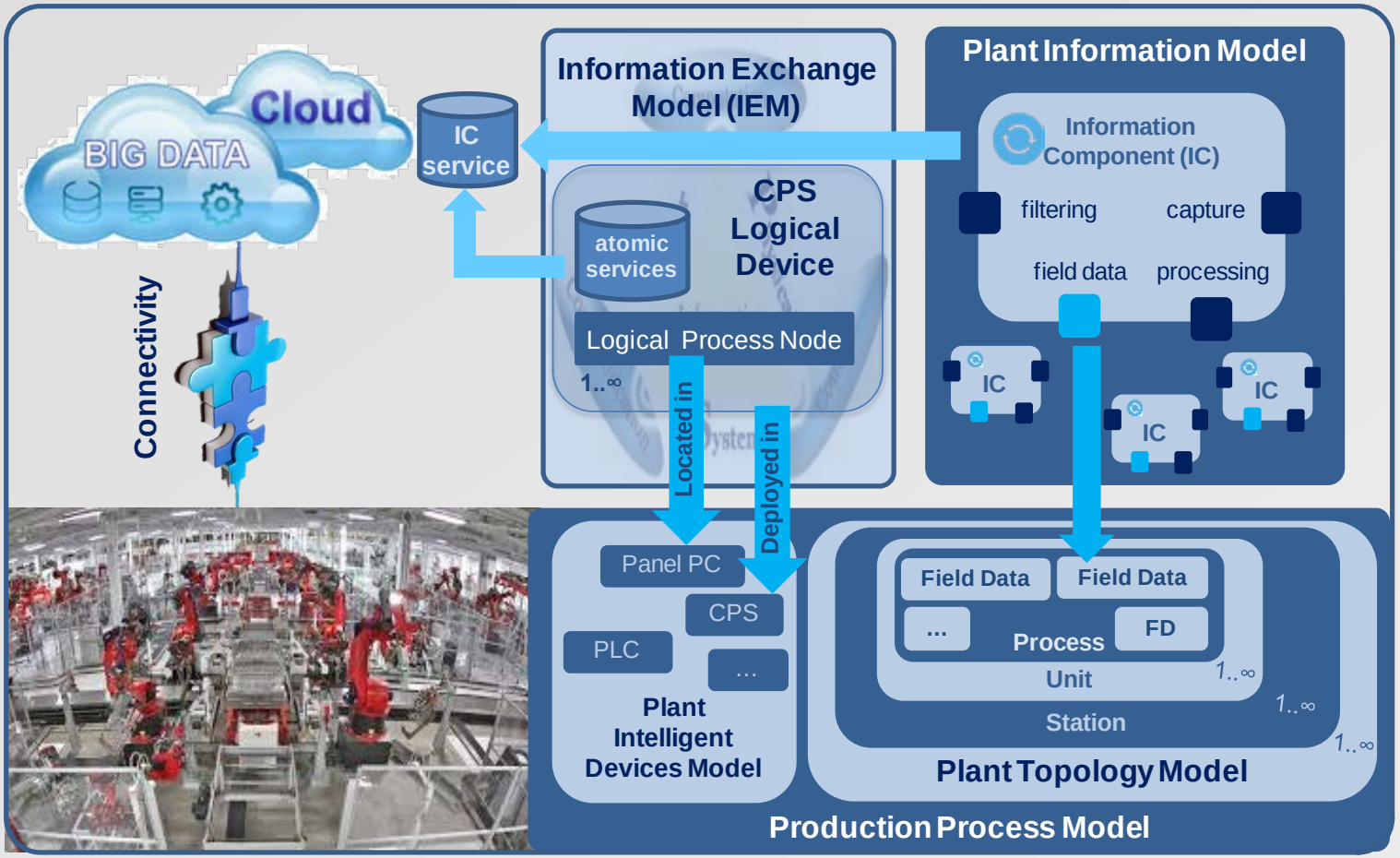
OPC UA promoted by OPC Foundation and standardized as IEC 62541

CPPS Architecture

- **Production Process Model**
  - Plant Topology Model
  - Plant Intelligent Device Model
- **Information Exchange Model**
  - Atomic Services
  - Logical Process Nodes
  - CPS Logical Devices
- **Plant Information Model**

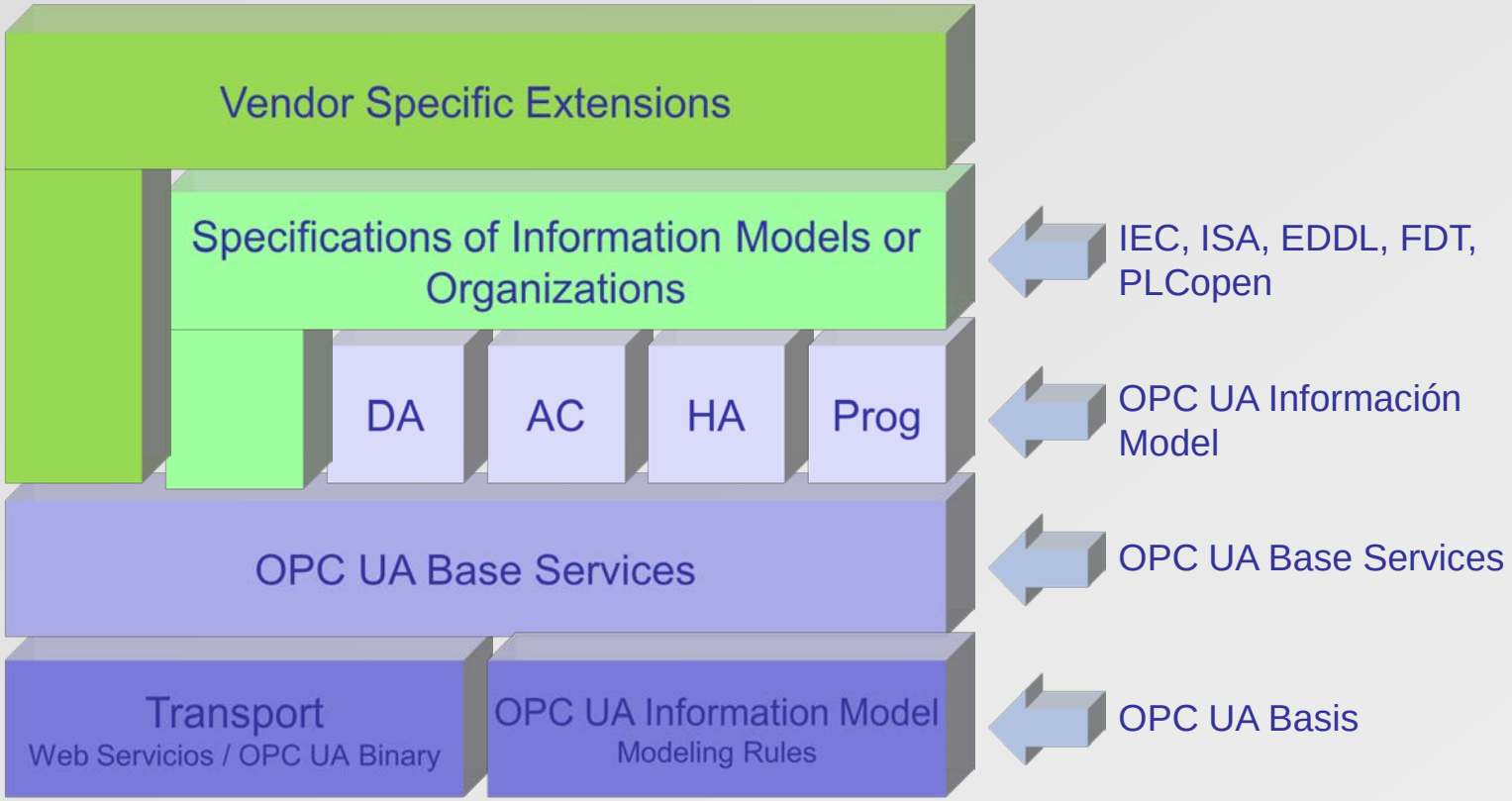


CPPS Architecture



CPPS Architecture in OPC UA

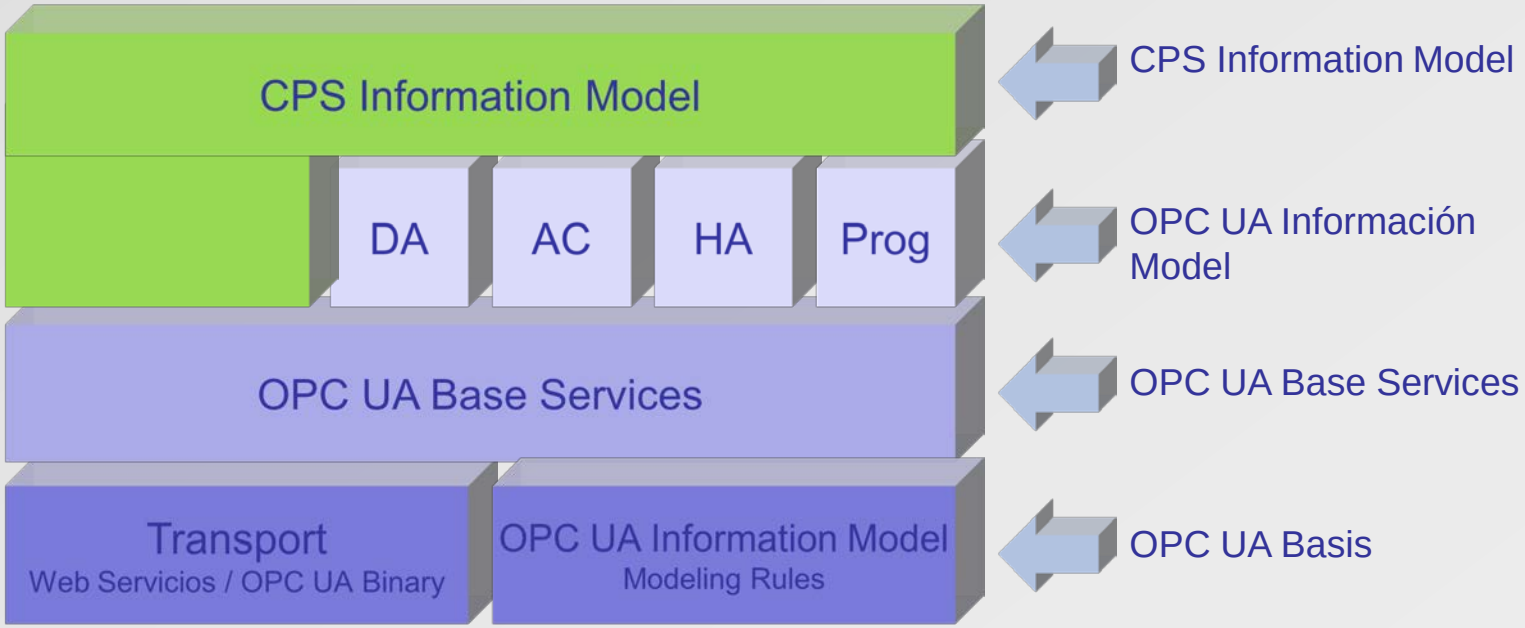
❑ CPPS model included as an OPC UA specific layer



CPPS Architecture in OPC UA

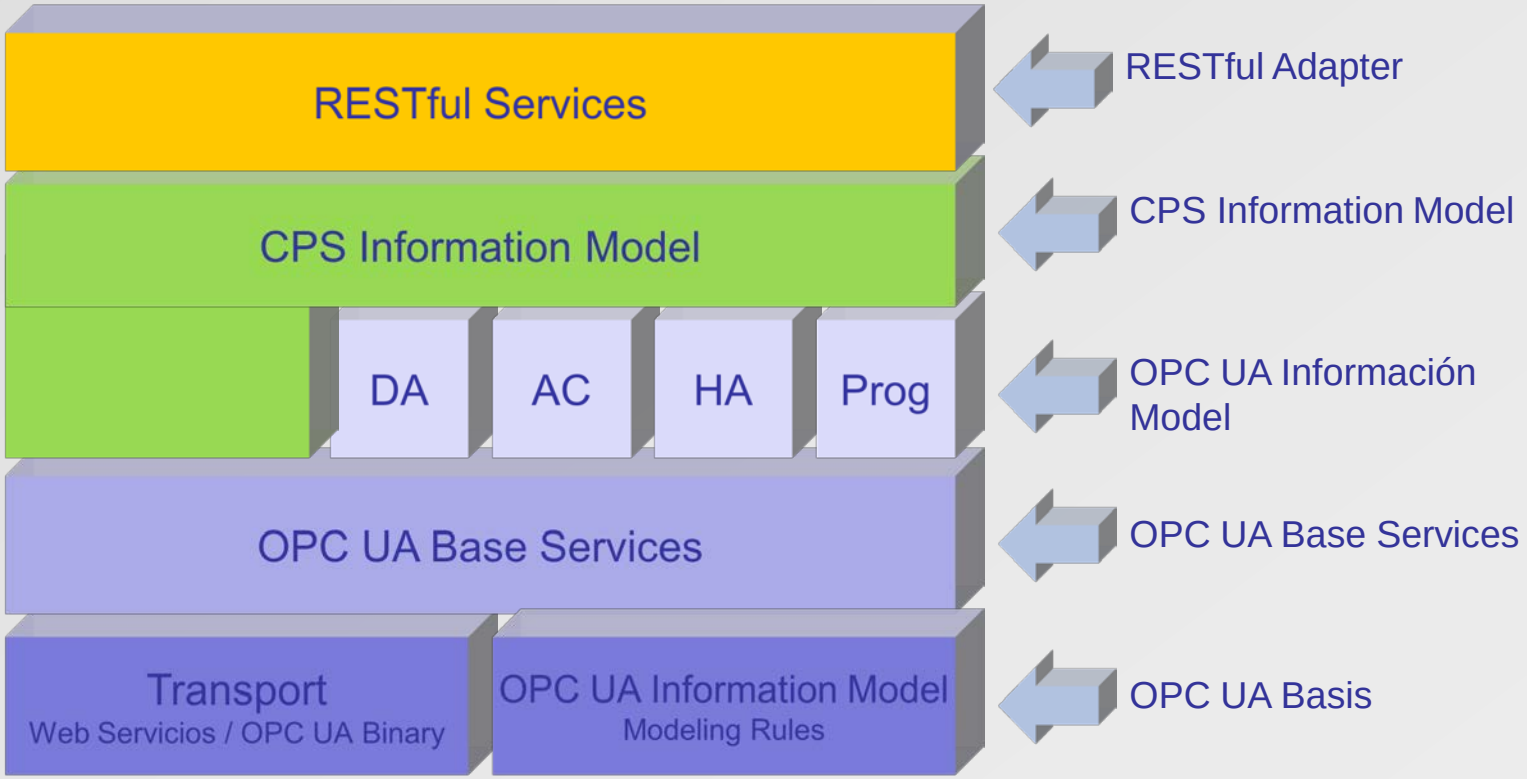
- Introduction
- CPPS Architecture
- CPPS Connectivity
- OPCUA Static
- OPCUA Dynamic
- Summary

❑ CPPS model included as an OPC UA specific layer

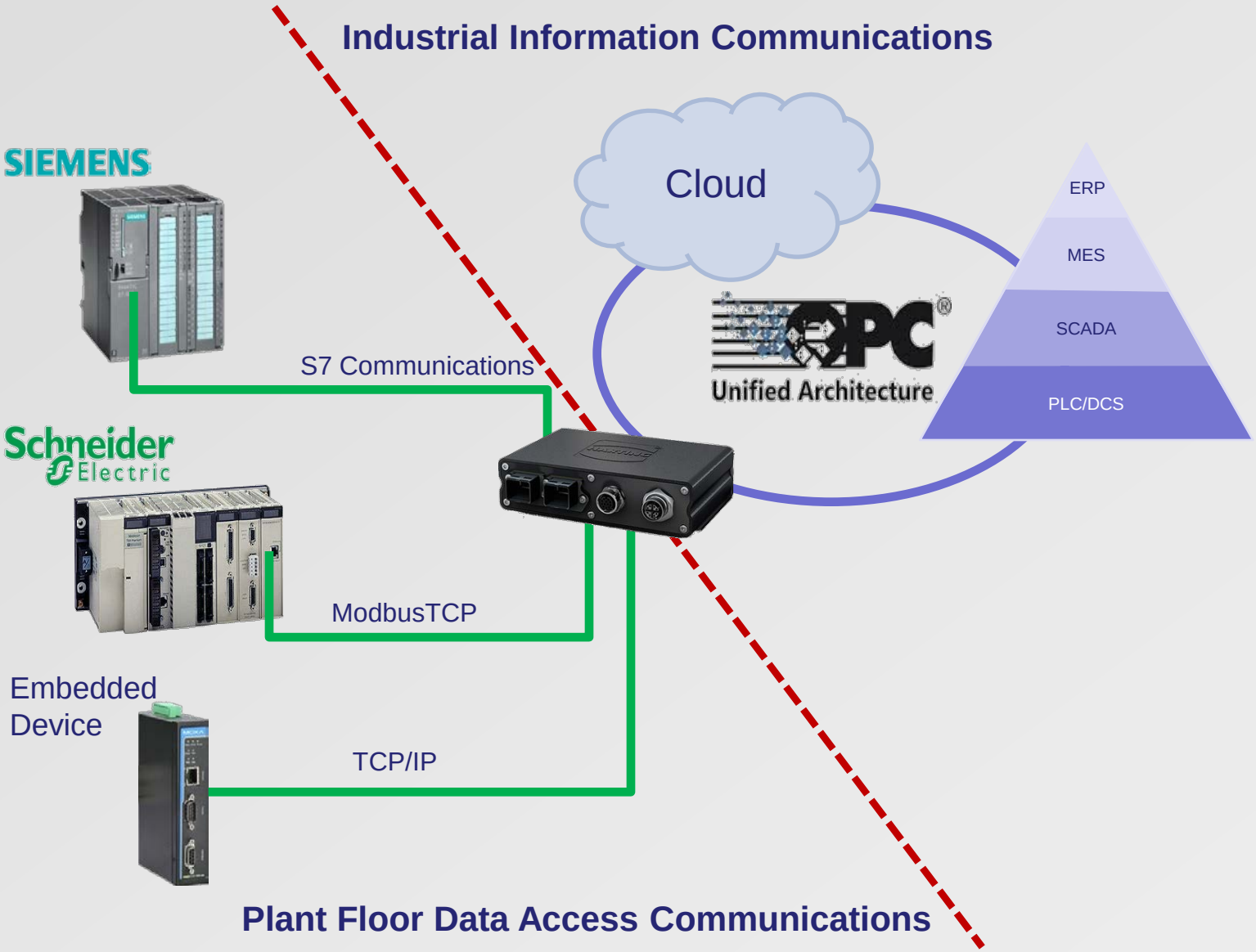


CPPS Architecture in OPC UA

❑ CPPS model included as an OPC UA specific layer



CPPS Connectivity

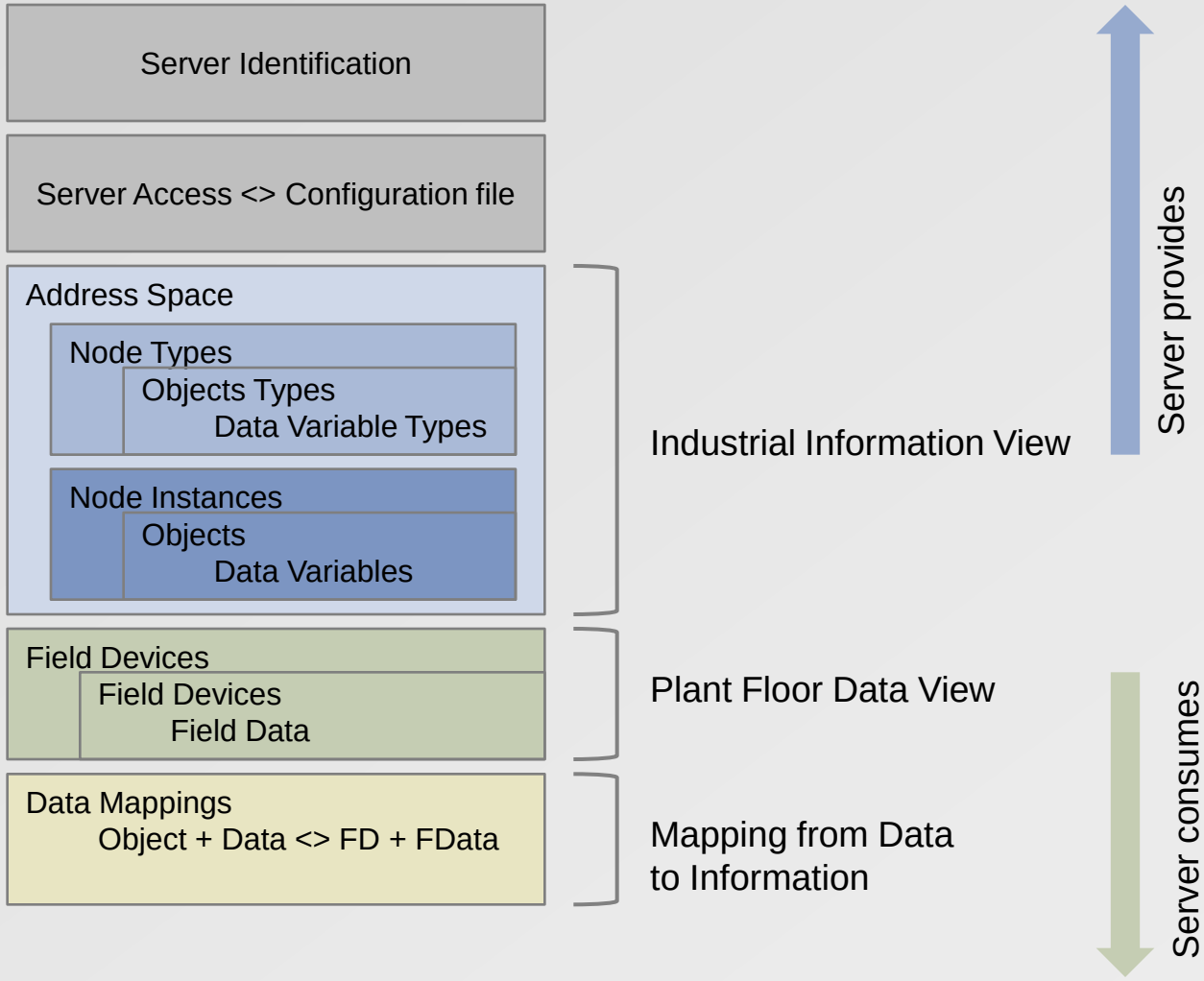


OPC UA Server Library

- **Multiplatform (Windows, Linux, Linaro ARM)**
  - Unified Automation SDK
- **XML Configuration**
  - Address Space (PLC likeness)
  - Field Data Access
    - S7 Communications (Siemens)
    - ModbusTCP (Schneider)
    - TCP/IP (Embedded Devices)
    - PiFace Digital (RaspberryPi)
- **Tested for own and third-party clients**

OPC UA Server Implementation

OPC UA Server Application Configuration XML File



OPC UA Server Implementation

OPCUA Server Application Configuration XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<OPCUAServerConfig Comment="OPCUAmdk Test Server Configuration - Arduino">
    <Identification ApplicationDomain="ThinkingFactory" Description="OPCUA server configuration model" />
    <VersionInfo Organization="GCIS DISA ETSI" Version="0.0" Author="FPG" Date="2016-07-05" Remarks="OPCUAmdk test server" />
    <Server Name="UATFServer" URL="opc.tcp://[NodeName]:48010" URI="urn:[NodeName]:ThinkinfFactory:UATFServer" ConfigFile="OPCServerConfig.xml"/>
    <NodeTypes>
        <ObjectType Name="ArduinoIODevType" NodeId="ArduinoIODev_Type">
            <DataVariableType Name="I0PINType" NodeId="I0PIN_Type">
                <Value Type="BOOL" Default="FALSE" />
                <AccessLevel>READ</AccessLevel>
                <ModellingRuleId>OPTIONAL</ModellingRuleId>
            </DataVariableType>
            <DataVariableType Name="O2PINType" NodeId="O2PIN_Type">
                <Value Type="BOOL" Default="FALSE" />
                <AccessLevel>READWRITE</AccessLevel>
                <ModellingRuleId>OPTIONAL</ModellingRuleId>
            </DataVariableType>
            <DataVariableType Name="I1REGType" NodeId="I1REG_Type">
                <Value Type="WORD" Default="0" />
                <AccessLevel>READ</AccessLevel>
                <ModellingRuleId>OPTIONAL</ModellingRuleId>
            </DataVariableType>
            <DataVariableType Name="O3REGType" NodeId="O3REG_Type">
                <Value Type="WORD" Default="0" />
                <AccessLevel>READWRITE</AccessLevel>
                <ModellingRuleId>OPTIONAL</ModellingRuleId>
            </DataVariableType>
        </ObjectType>
    </NodeTypes>
    <NodeInstances>
        <Object Name="ArduinoTk" NodeTypeId="ArduinoIODev_Type">
            <DataVariable Name="Button" NodeTypeId="I0PIN_Type"/>
            <DataVariable Name="Potentiometer" NodeTypeId="I1REG_Type"/>
            <DataVariable Name="GreenLed" NodeTypeId="O2PIN_Type"/>
            <DataVariable Name="RedLed" NodeTypeId="O3REG_Type"/>
        </Object>
    </NodeInstances>
```

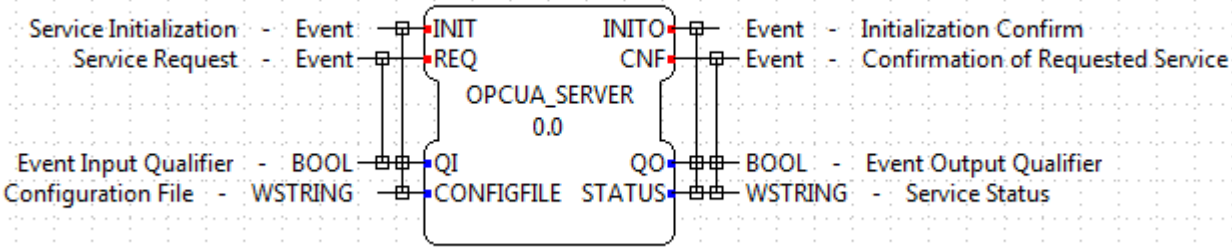
OPC UA Server Implementation

OPCUA Server Application Configuration XML File

```
    <FieldDevices>
        <FieldDevice Name="ArduinoTCP133" Type="ModbusTCP">
            <UpdateTime>50</UpdateTime>
            <IPAddress>192.168.0.133</IPAddress>
            <FieldData Name="IButton" Type="BOOL" AccessLevel="READ" Address="%I0" />
            <FieldData Name="IPotentiometer" Type="WORD" AccessLevel="READ" Address="%IW1" />
            <FieldData Name="OGreenLed" Type="BOOL" AccessLevel="READWRITE" Address="%Q2" />
            <FieldData Name="ORedLed" Type="WORD" AccessLevel="READWRITE" Address="%QW3" />
        </FieldDevice>
    </FieldDevices>
    <DataMappings>
        <DataMapping Object="ArduinoTk" DataVariable="Button" FieldDevice="ArduinoTCP133" FieldData="IButton" />
        <DataMapping Object="ArduinoTk" DataVariable="Potentiometer" FieldDevice="ArduinoTCP133" FieldData="IPotentiometer" />
        <DataMapping Object="ArduinoTk" DataVariable="GreenLed" FieldDevice="ArduinoTCP133" FieldData="OGreenLed" />
        <DataMapping Object="ArduinoTk" DataVariable="RedLed" FieldDevice="ArduinoTCP133" FieldData="ORedLed" />
    </DataMappings>
</OPCUAServerConfig>
```

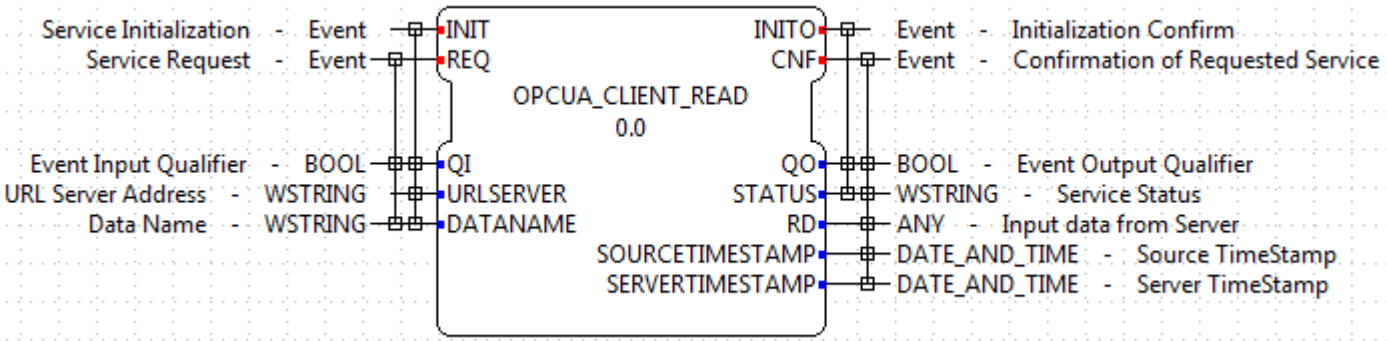
4DIAC-FORTE Implementation

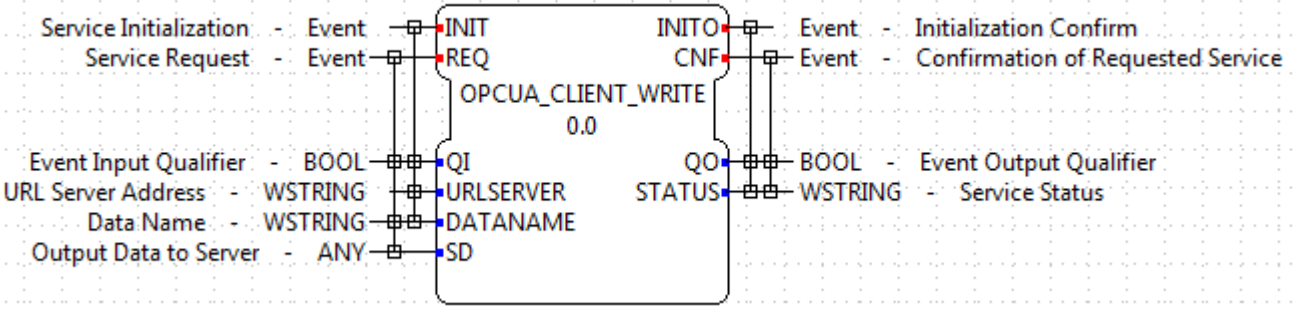
OPCUA Server SIFB



4DIAC-FORTE Implementation

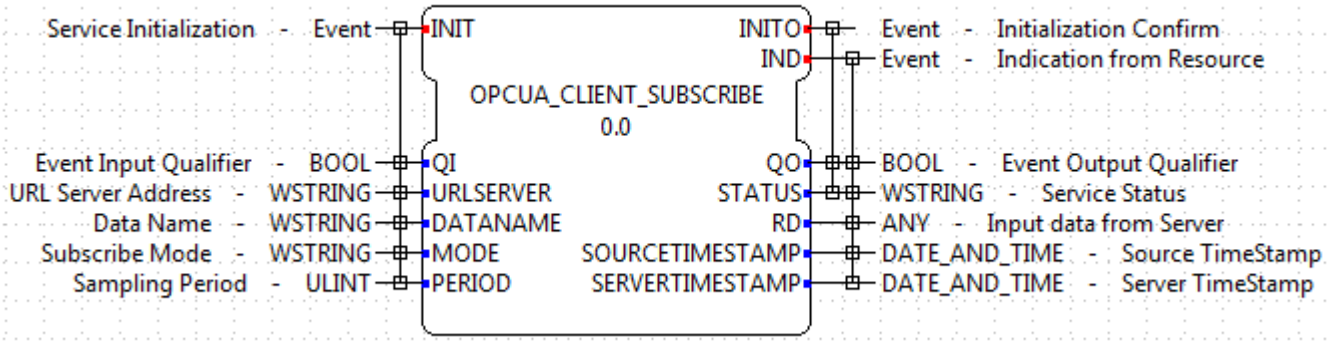
OPCUA Client SIFBs





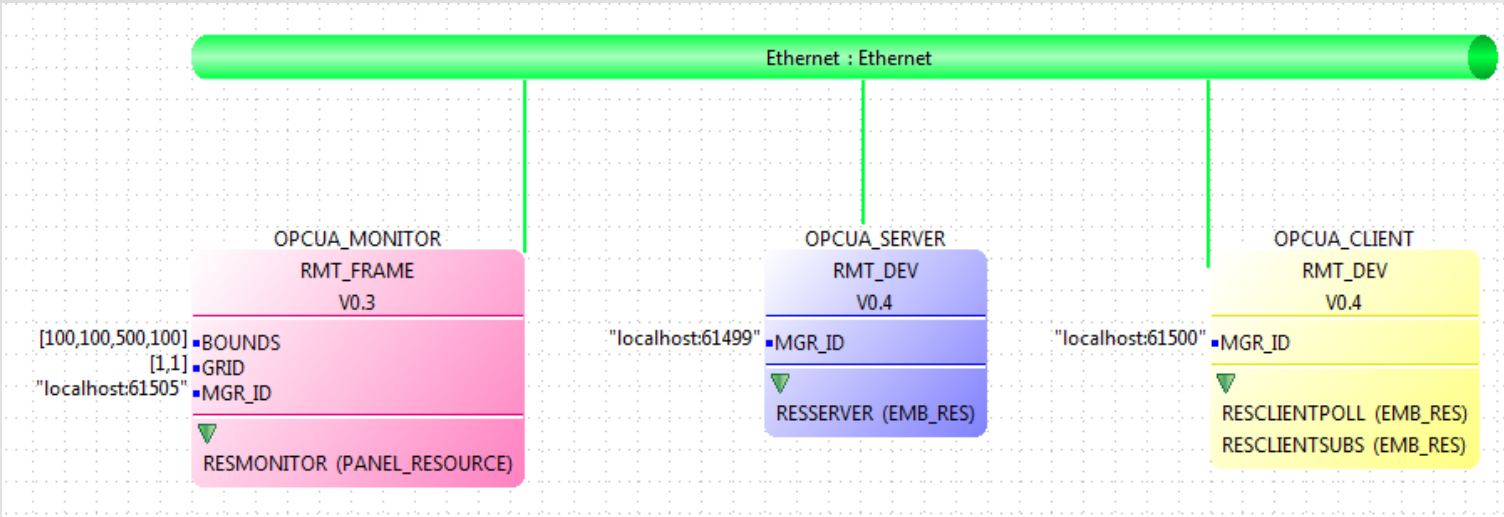
4DIAC-FORTE Implementation

OPCUA Client-Subscription SIFB



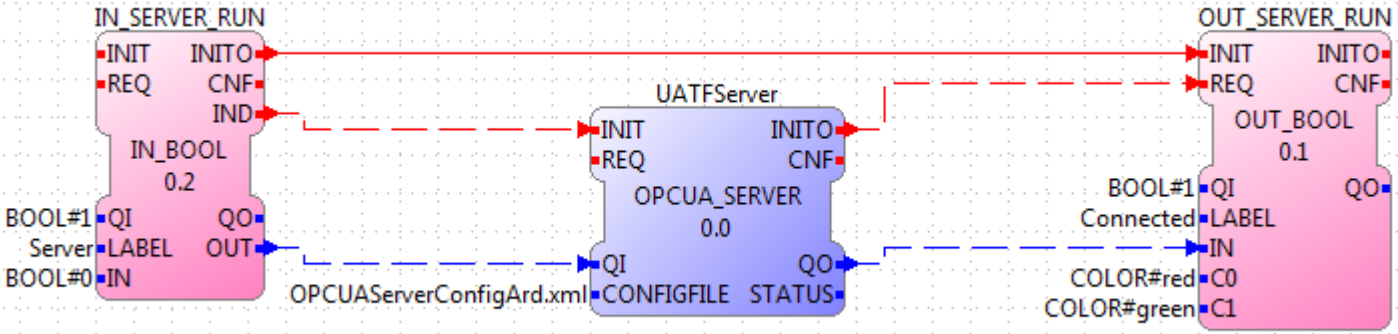
4DIAC-FORTE Implementation

OPCUA Test – IEC 61499 System

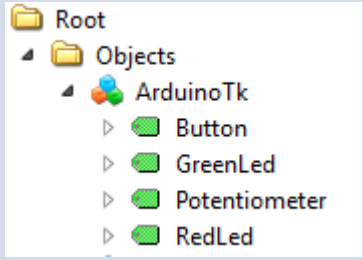


4DIAC-FORTE Implementation

OPCUA Test 1 – Server Application

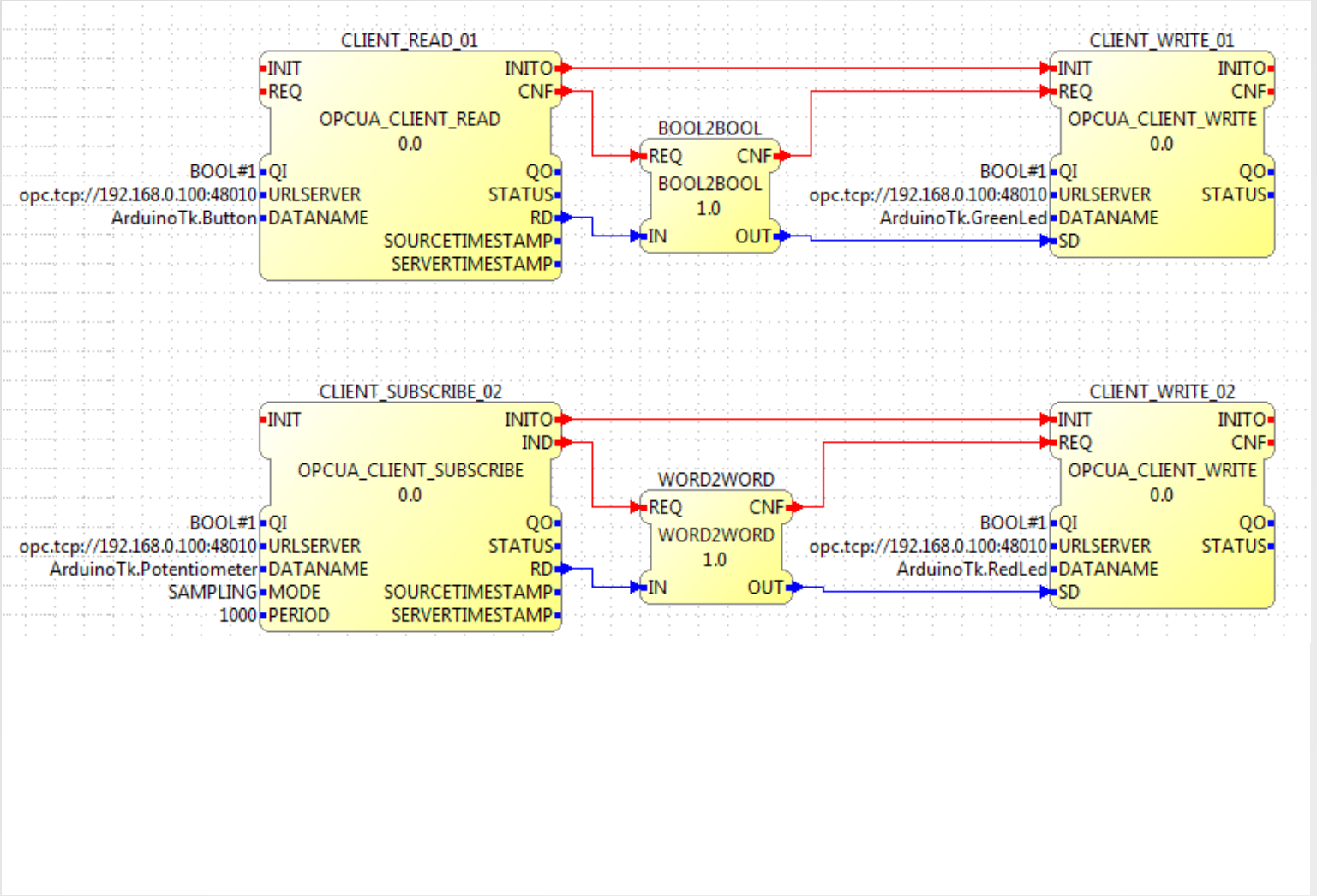


Server Address Space



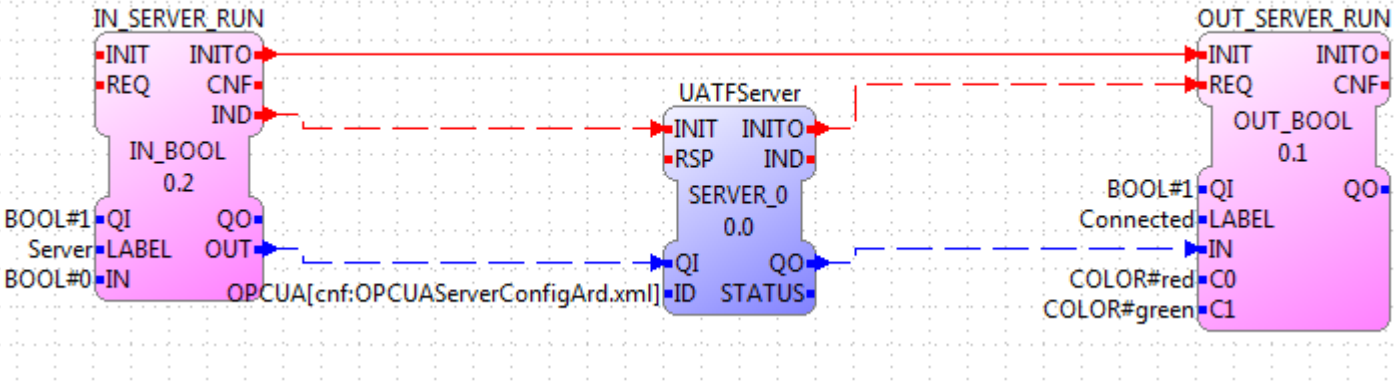
4DIAC-FORTE Implementation

OPCUA Test 1 – Client Application

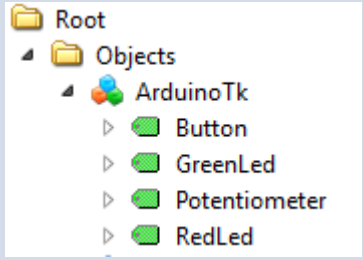


4DIAC-FORTE Implementation

OPCUA Test 2 – Server Application – FORTE ComLayer

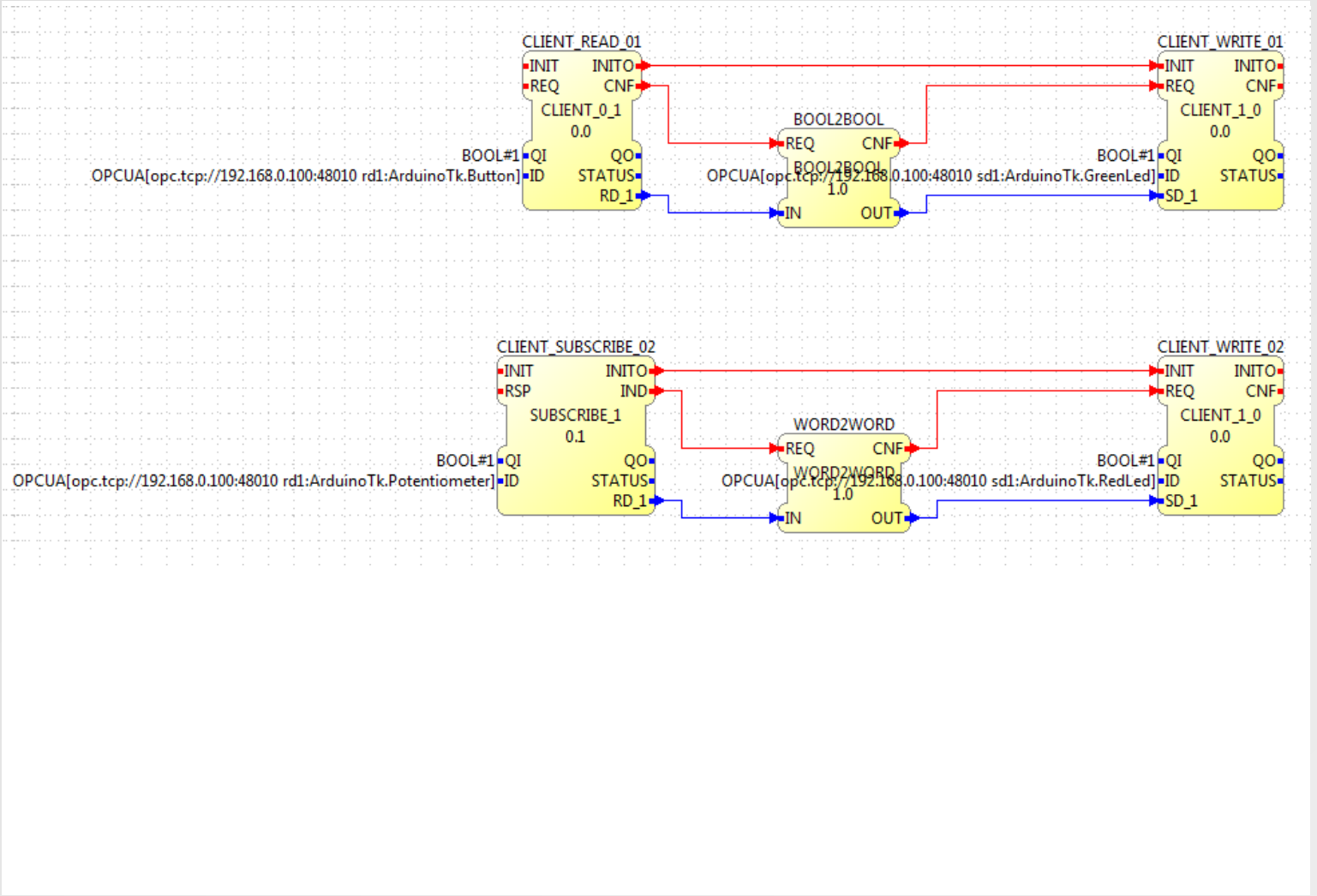


Server Address Space



4DIAC-FORTE Implementation

OPCUA Test 2 – Server Application – FORTE ComLayer



OPC UA Server Dynamic Configuration

TAGs: TagName <ObjectName.VariableName>

- **Memory Tags**
  - Local Memory Access Driver
    - LocalMemory

`<TagName>,<Type>,<AccessLevel>`

- **Process Tags**
  - Field Data Access Drivers
    - S7 (Siemens devices)
    - ModbusTCP (Schneider devices)
    - TCPDataLink (Embedded devices)
    - PiFaceDigital (RaspberryPi – PiFace Digital)

`<TagName>,<Type>,<AccessLevel>,<FDname>,<FDAddress>`

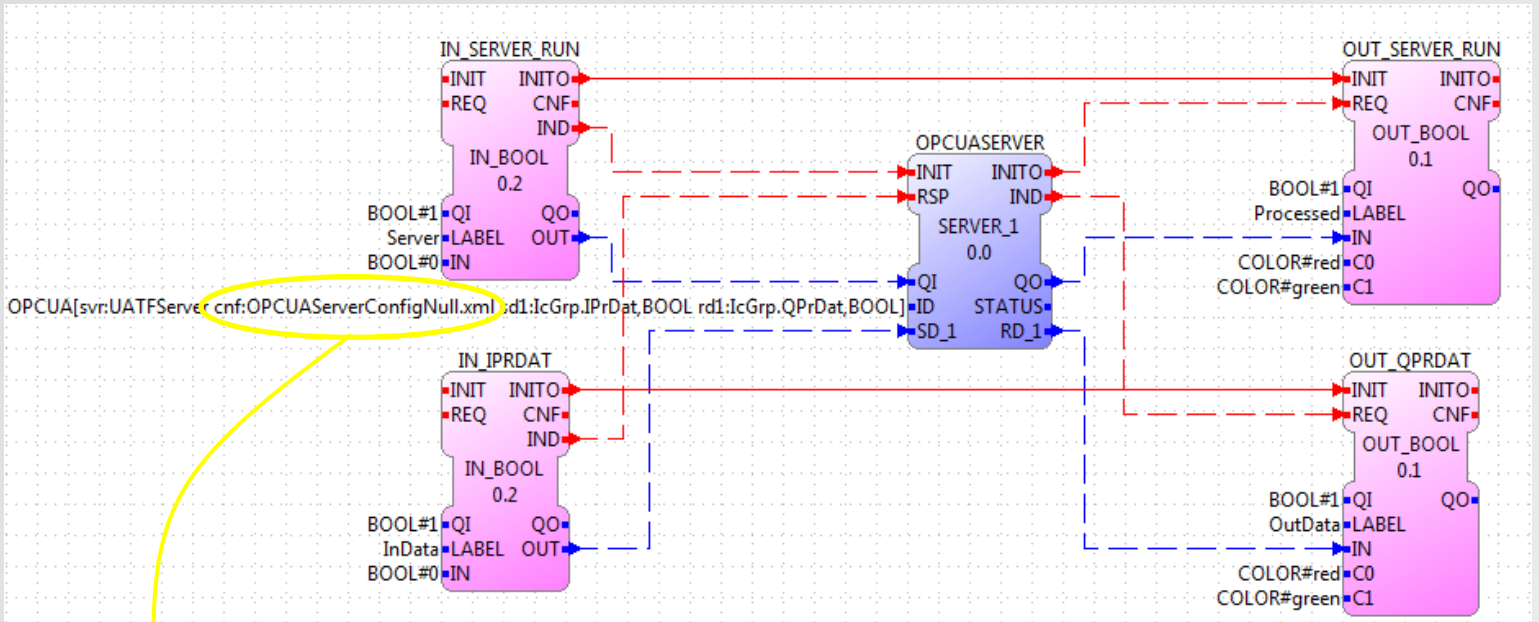
OPC UA Server Dynamic Configuration

OPCUA[<params>]

- **Server Parameters**
  - Server name [o] – `svr:<servername>`
  - Configuration file [m] – `cnf:<configfile>`
  - Field device [o] – `fd:<fdname,driver,params,...>`
  - Process Tag [o] – `tg[num]:<proctag>`
  - SD parameter (memory tag) [o] – `sd[num]:<memtag>`
  - RD parameter (memory tag) [o] – `rd[num]:<memtag>`
- **Client Parameters**
  - Client name [o] – `cln:<clientname>`
  - Server name [o] – `svr:<servername>` `(memory access)`
  - Server URL [o] – `opc.tcp://<URL>` `(opc ua access)`
  - SD parameter [o] – `sd[num]:<tagname>`
  - RD parameter [o] – `rd[num]:<tagname>`

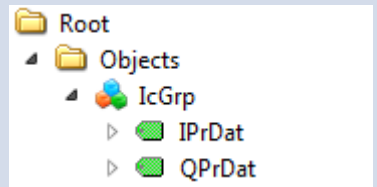
4DIAC-FORTE Implementation

OPCUA Test 3 – Dynamic Server Application – FORTE ComLayer



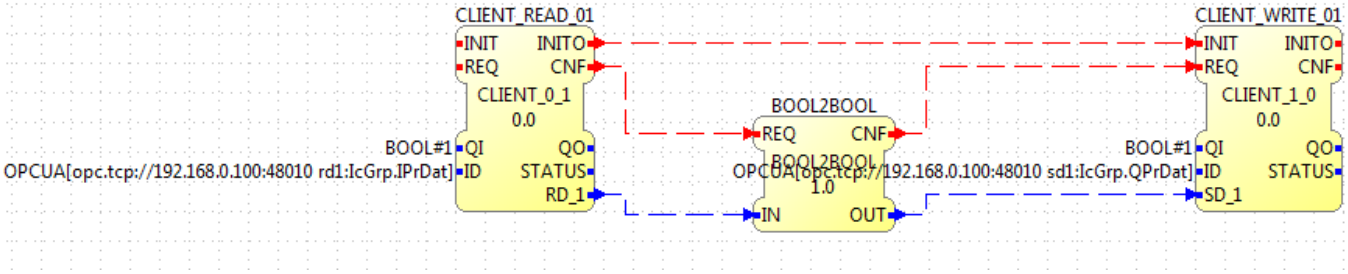

```
<?xml version="1.0" encoding="UTF-8"?>
<OPCUAServerConfig Comment="OPCUAmdk Test Server Configuration - Null">
    <Identification ApplicationDomain="ThinkingFactory" Description="OPCUA server configuration model" />
    <VersionInfo Organization="GCIS DISA ETSI" Version="0.0" Author="FPG" Date="2016-07-05" Remarks="OPCUAmdk test server" />
    <Server Name="UATFServer" URL="opc.tcp://[NodeName]:48010" URI="urn:[NodeName]:ThinkinfFactory:UATFServer" ConfigFile="OPCServerConfig.xml"/>
</OPCUAServerConfig>
```

Server Address Space



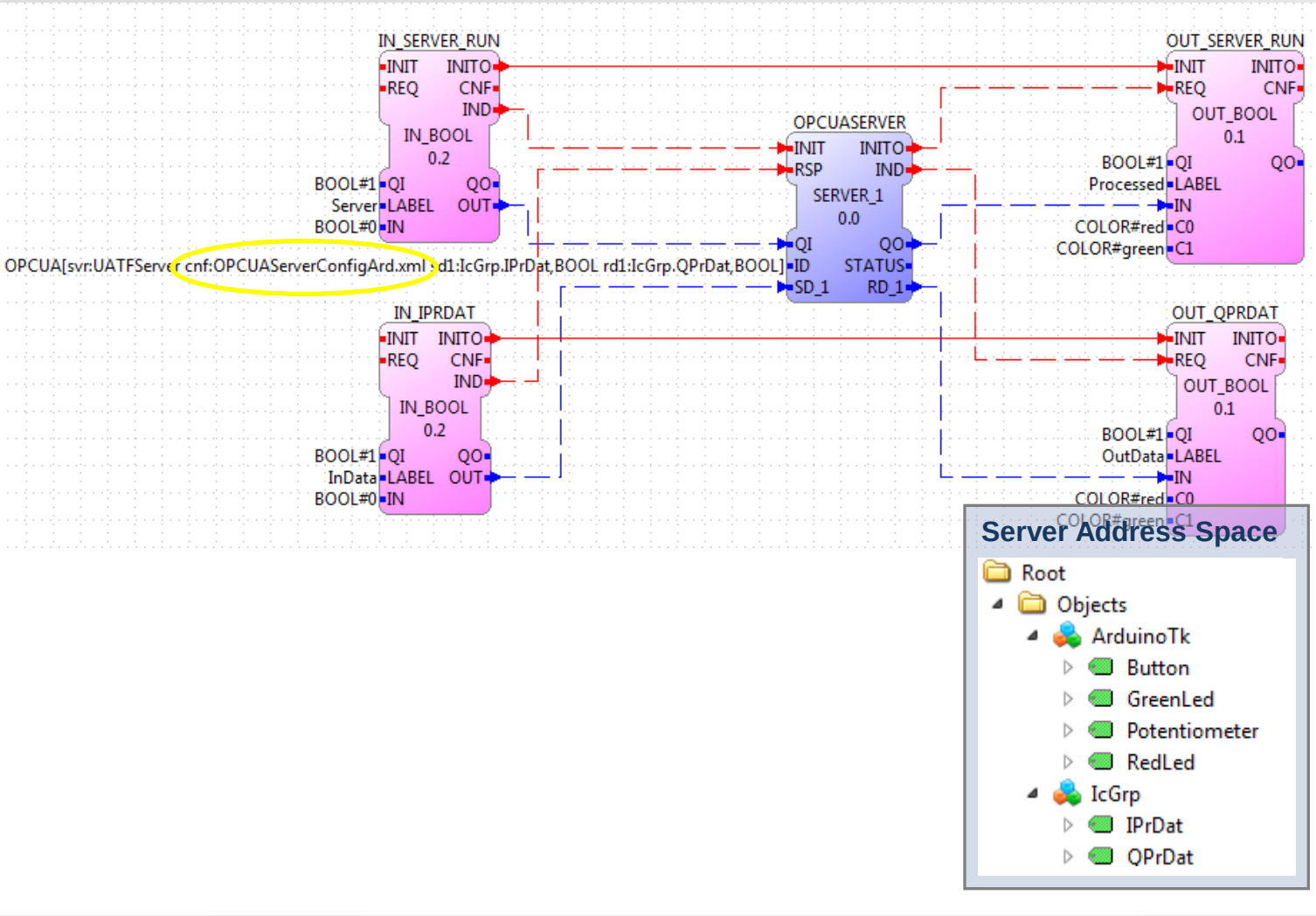
4DIAC-FORTE Implementation

OPCUA Test 3 – Dynamic Client Application – FORTE ComLayer



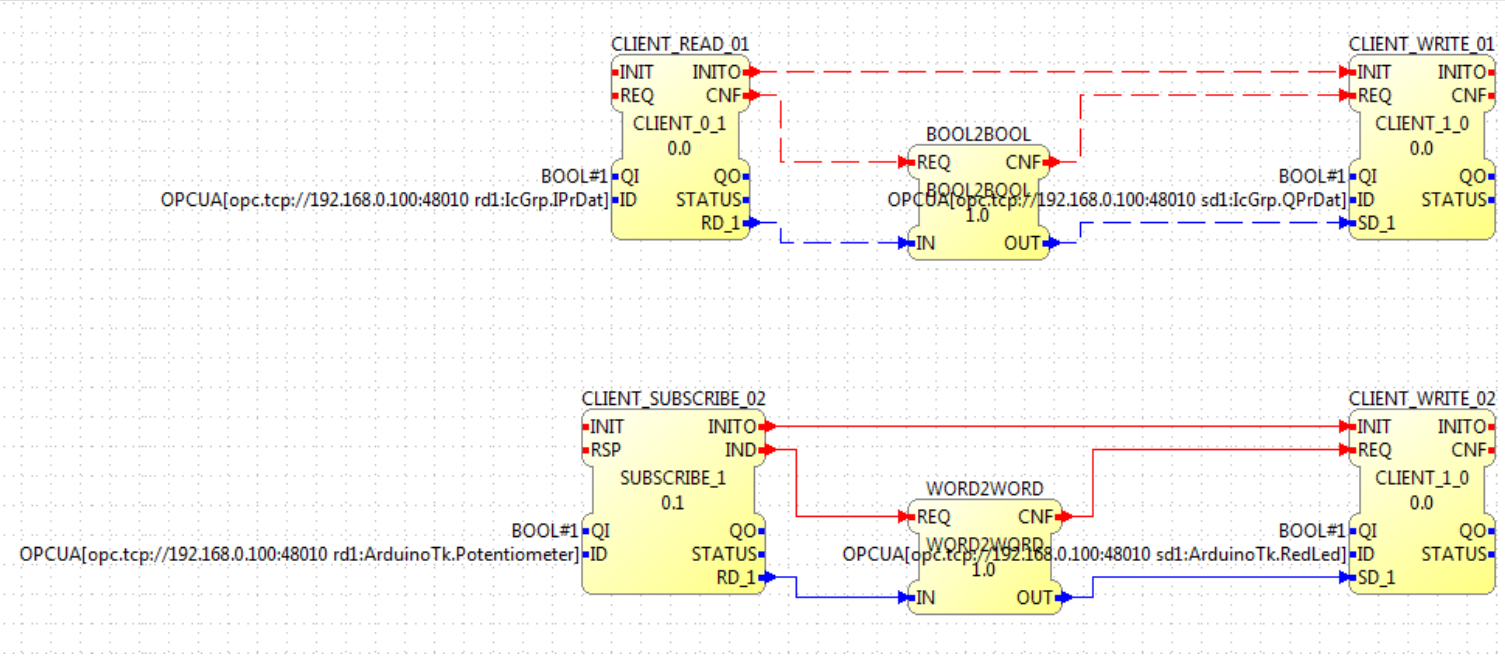
4DIAC-FORTE Implementation

OPCUA Test 4 – Dynamic Server Application – FORTE ComLayer



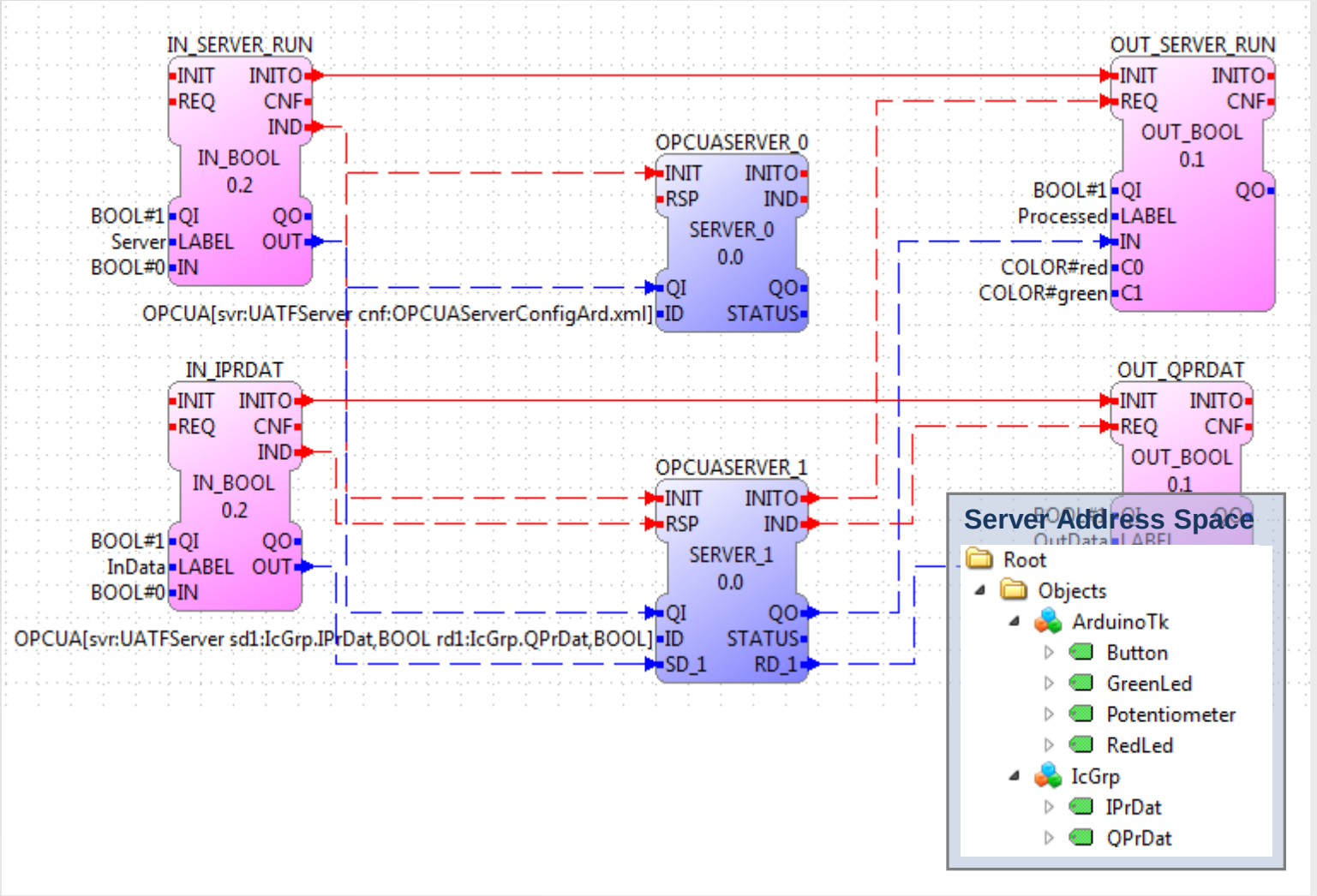
4DIAC-FORTE Implementation

OPCUA Test 4 – Dynamic Client Application – FORTE ComLayer



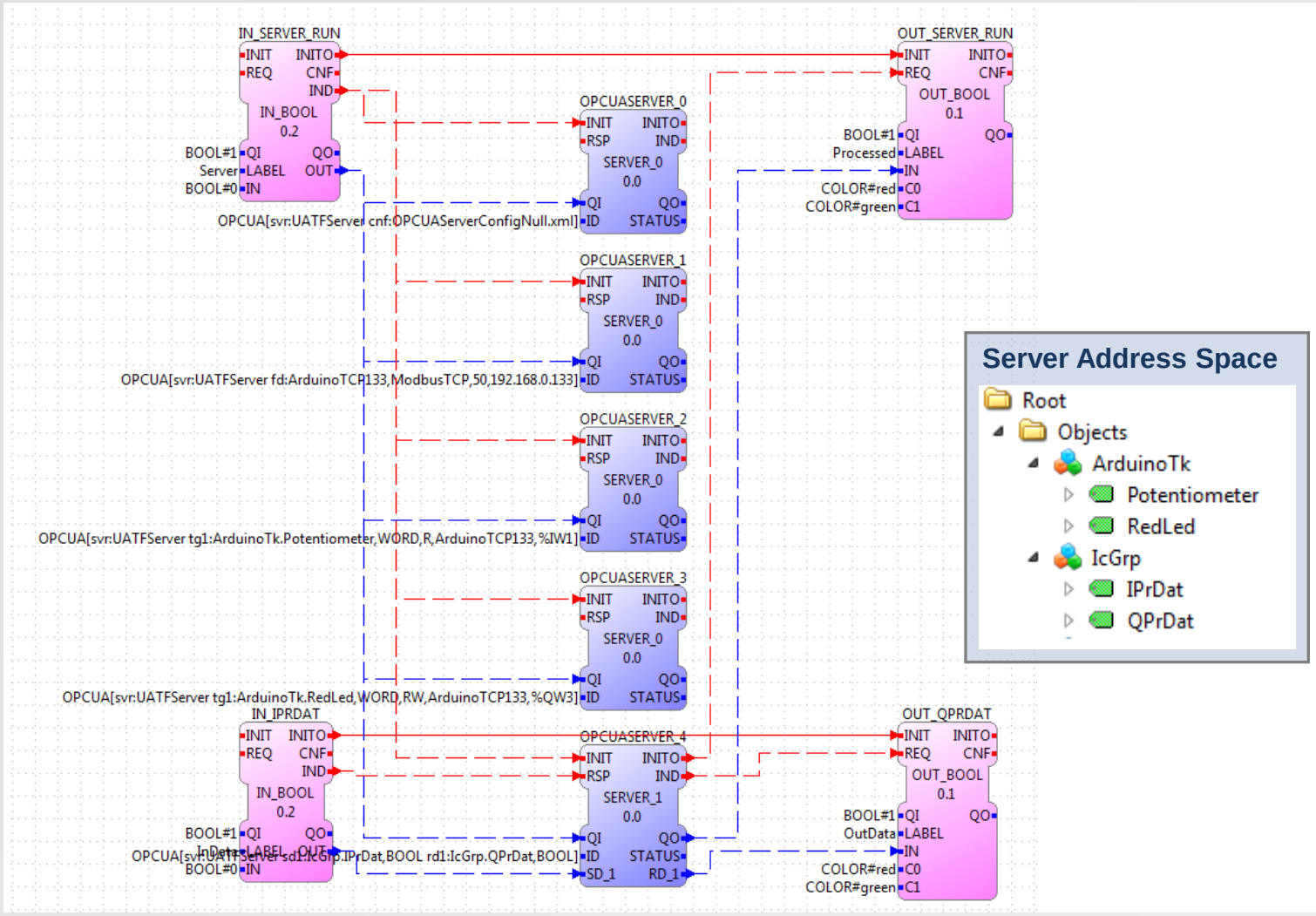
4DIAC-FORTE Implementation

OPCUA Test 5 – Dynamic Server Application – FORTE ComLayer



4DIAC-FORTE Implementation

OPCUA Test 6 – Dynamic Server Application – FORTE ComLayer



Conclusions

- **SIFB architecture for services composition**
  - Model-based architecture
  - IEC 61499 compliant – FORTE ComLayer
  - Making use of well-established standards
  - Seamless integration within Industry 4.0 contexts
- **Future work …**
  - RESTful access for cloud
  - FORTE integration in OPC UA server using methods

Questions