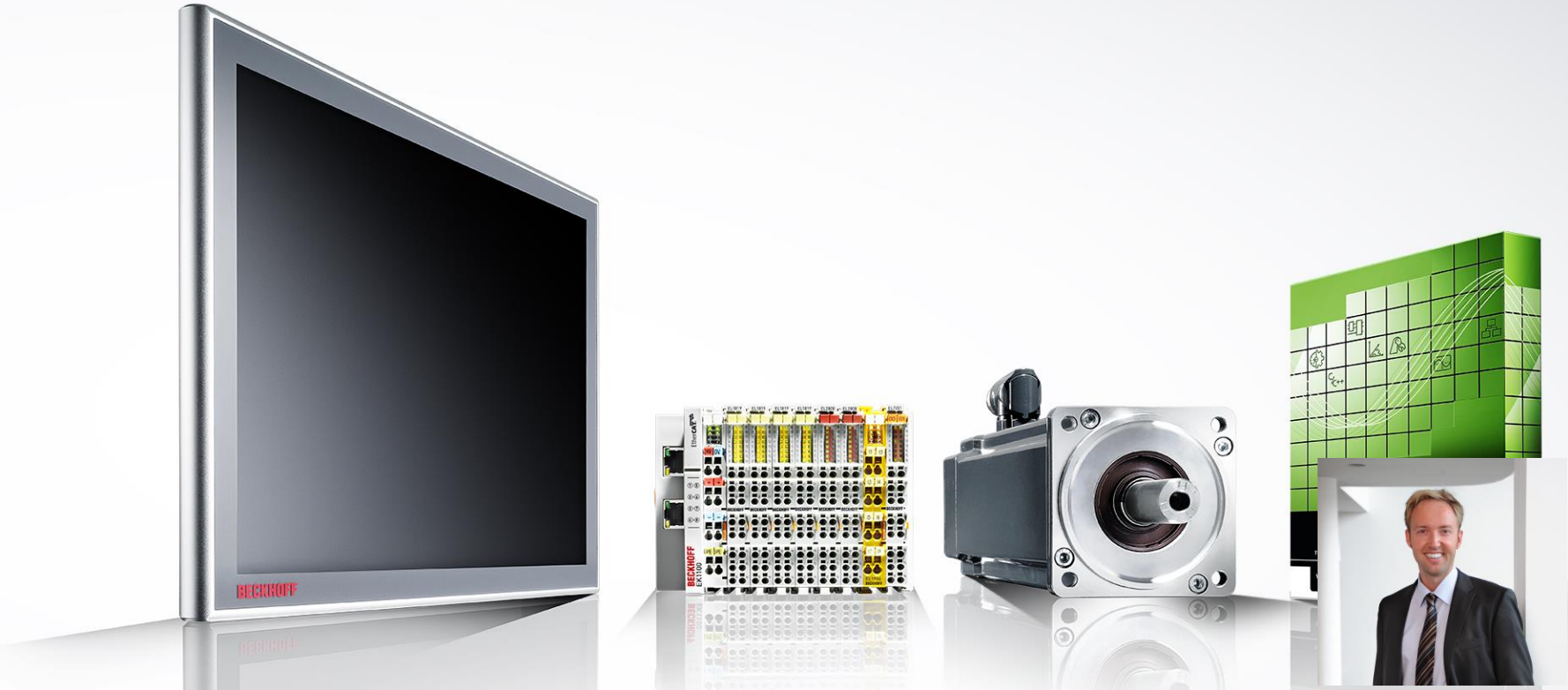
OPC-UA

BECKHOFF



Sven Goldstein
TwinCAT 产品经理
通信 & 自动化接口

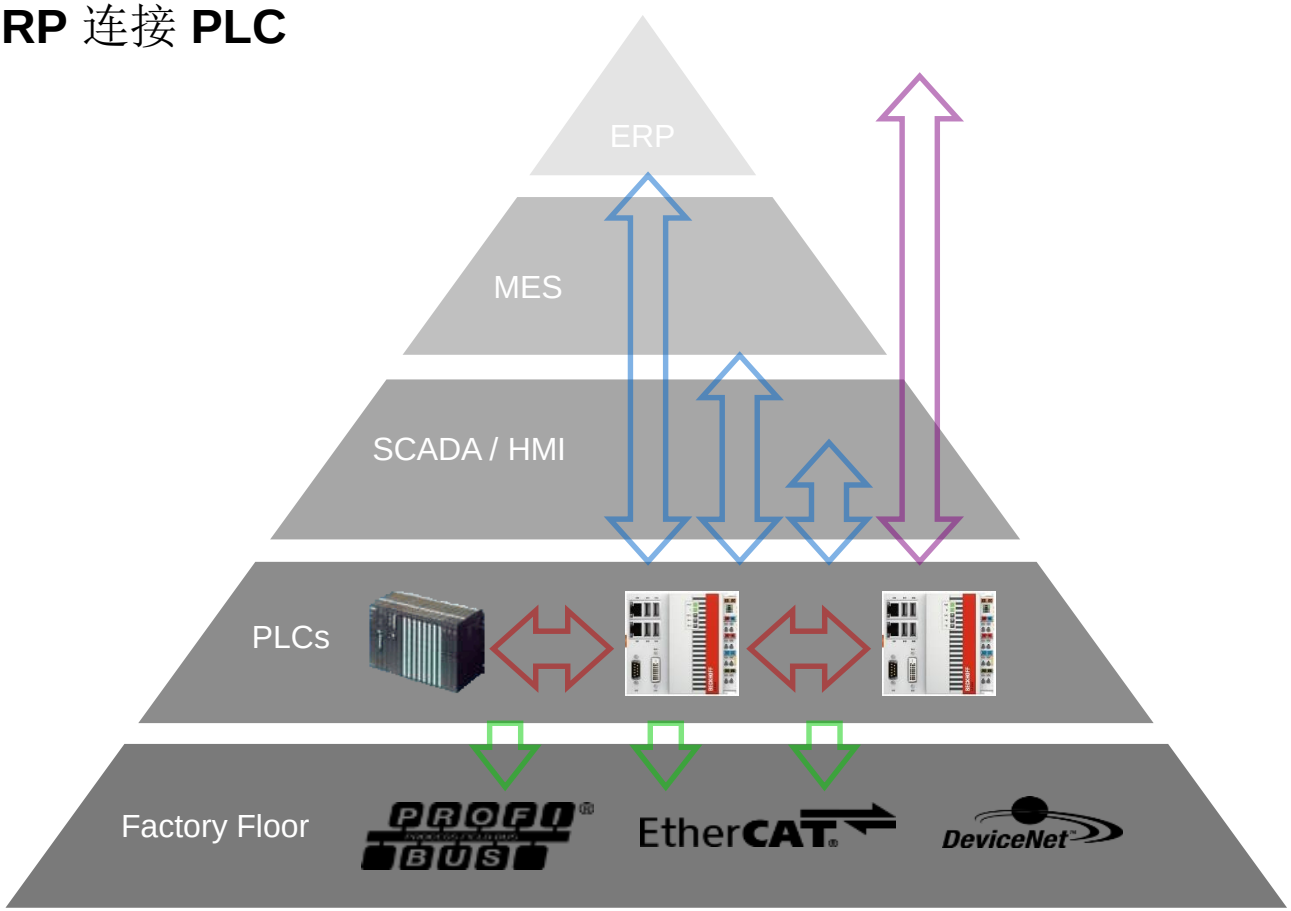
目录

BECKHOFF

当前工厂通讯的结构

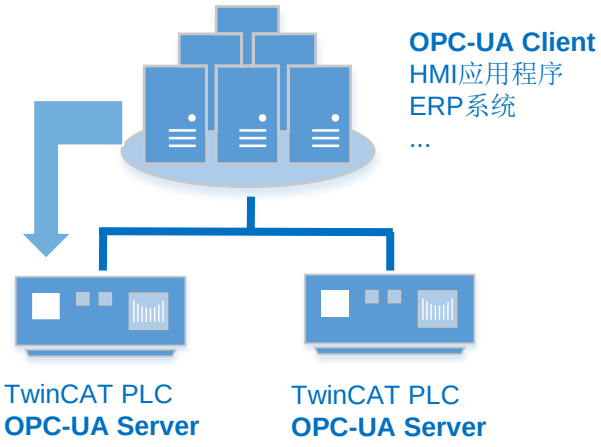
- 纵向: **SCADA / MES / ERP** 连接 **PLC**
  - 访问PLC中的过程数据
- 横向: **PLC** 连接 **PLC**
  - 通过协议进行数据交换
- I/O: **PLC** 连接 **Fieldbus**
  - 访问现场设备的数据
- 云: **PLC** 连接 **Cloud**
  - 访问云的数据记录

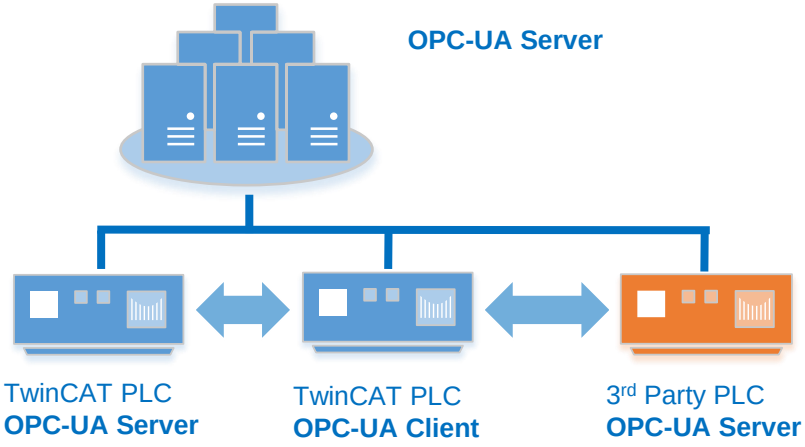
→ 数据传递!!



概述

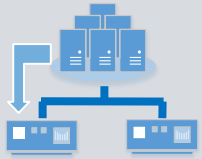
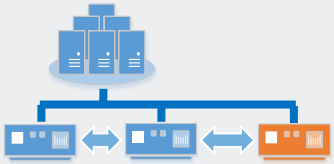
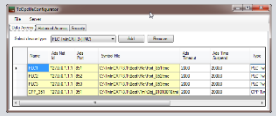
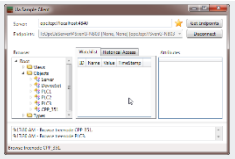
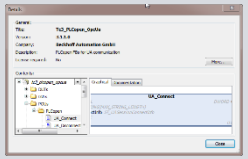
- OPC-UA: 标准化的数据通信协议 (IEC 62541)
- 实现了不同供应商和设备之间的数据通信
- 完备的信息模型和集成安全的特性
- 可以用于多种工业领域





TwinCAT OPC-UA

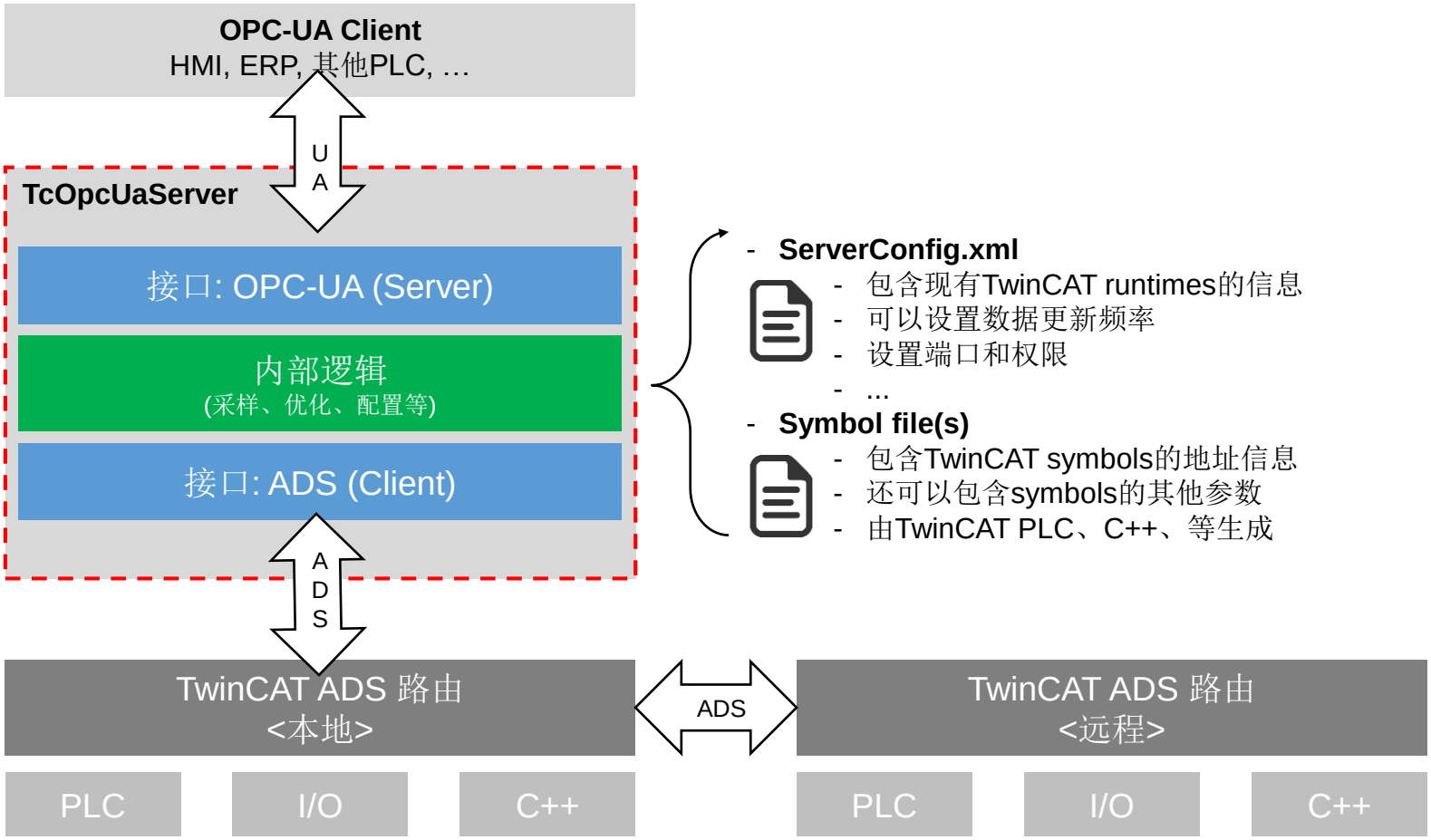
BECKHOFF

产品组件	简介	
OPC-UA Server	可以通过OPC-UA协议来提供TwinCAT PLC、C++、IO-Task中的变量。例如：被HMI应用程序读写。	
OPC-UA Client	可以通过PLC功能块访问本地\远程的OPC-UA Servers 里面的数据	
OPC-UA Configurator	OPC-UA Server的图形化配置和诊断工具	
OPC-UA Sample Client	示例OPC-UA Client，用于快速测试安装和配置	
Tc3_PLCopen_OpcUa	为OPC-UA Client提供相应的库。自动安装在TwinCAT XAE systems中。	

目录

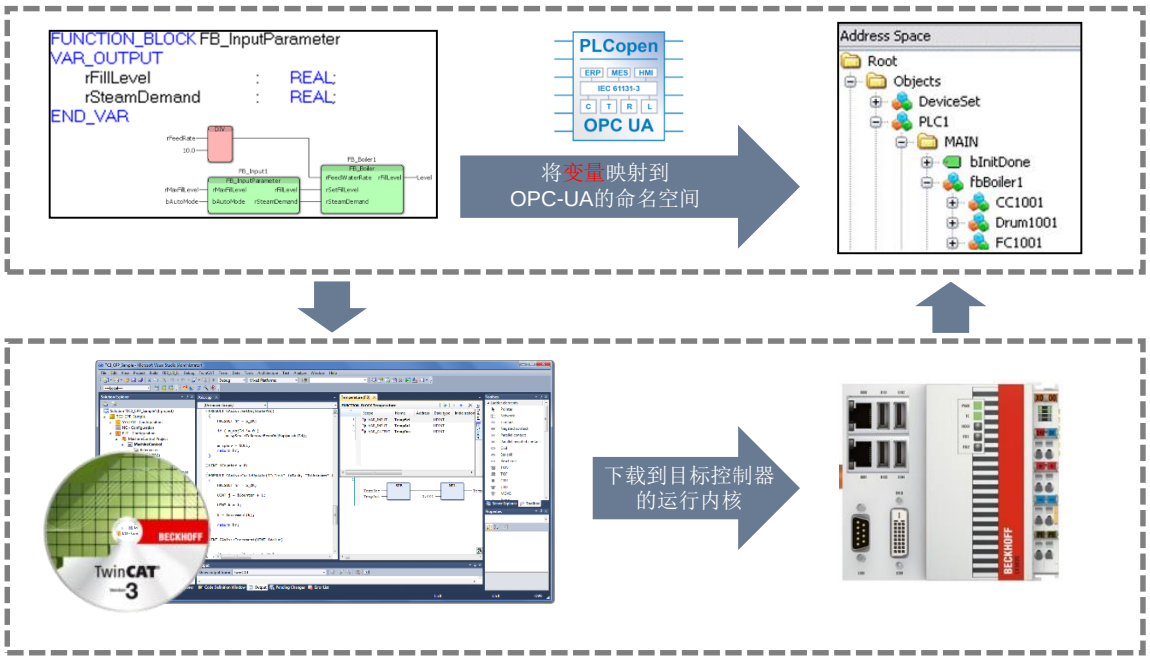
BECKHOFF

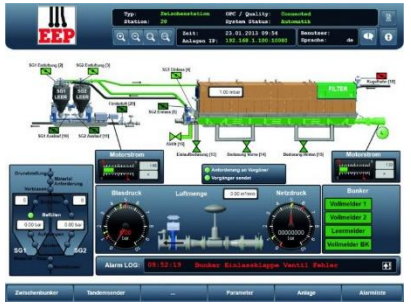
OPC-UA Server: 架构概况



OPC-UA Server: 概述与快速入门

- 默认设置: 先采用默认配置!
- Step 1: 选择访问PLC程序内的变量和使能symbol file下载
- Step 2: 将PLC程序下载到目标控制器的运行内核(TwinCAT runtime)
- Step 3: 通过OPC-UA Client访问PLC程序内的变量。例如：HMI应用

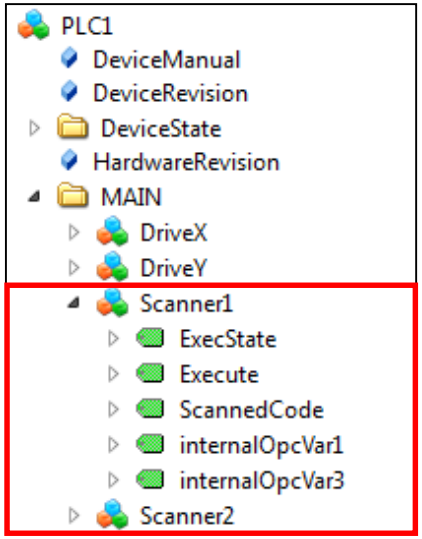




HMI 应用
(OPC-UA Client)

OPC-UA Server: 示例

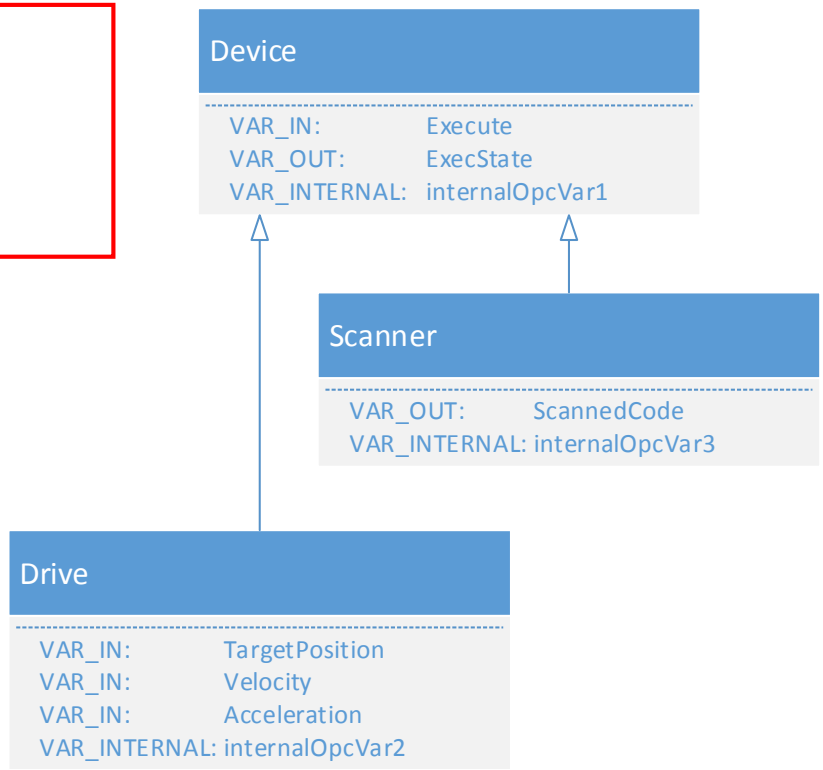
- 小型PLC项目：基于IEC61131-3标准，使用了面向对象中的继承
- UA-Server提供变量的分层信息



Reference	Target DisplayName
HasTypeDefiniti...	
HasInputVar	Execute
HasOutputVar	ExecState
HasLocalVar	internalOpcVar1
HasOutputVar	ScannedCode
HasLocalVar	internalOpcVar3

```
FUNCTION_BLOCK Scanner EXTENDS Device
VAR_INPUT
END_VAR
VAR_OUTPUT
    ScannedCode : STRING;
END_VAR
VAR
    internalOpcVar3 : BOOL;
END_VAR
```

Mapping to UA



OPC-UA Server: Symbol file

- 要求UA-Server从控制器运行内核(TC runtime)中获取symbol信息
- Symbol files包含symbols的地址信息 (变量, 结构, ...)
- 也可以包含更高级的OPC-UA参数(例如：历史数据访问→程序中例2)
- 由TwinCAT XAE生成。例如：当编译PLC程序时生成

Symbol file type	TwinCAT version	Modes	Short descriptions
TPY	TwinCAT 2 and TwinCAT 3	筛选和未筛选	传统格式。没有继承、类型信息和访问TC C++的权限。通常在客户将TC2迁移到TC3时采用。建议在筛选模式下使用PLC comments
TMC	TwinCAT 3	筛选和未筛选	新格式。支持继承、类型信息和访问TC C++的权限。在筛选模式下使用PLC属性。

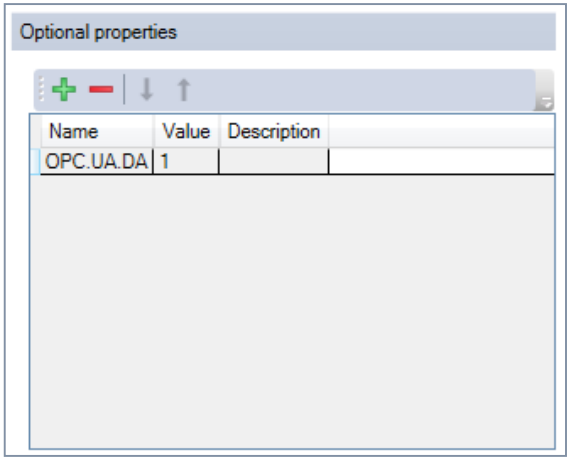
OPC-UA Server: Symbol file

- 在过滤模式下，可以通过OPC-UA明确选择symbols。
- PLC 和 C++都支持使用过滤模式和未过滤模式。
- **未过滤：**所有运行内核中的symbols都可以在OPC-UA 命名空间中找到。
  - 优点：易于安装和维护
  - 缺点：
    - 系统资源占用率高(RAM)
    - 不支持附加的UA特性。例如：历史数据的访问
- **过滤：**可以指定显示在OPC-UA 命名空间中的symbols。
  - 优点：
    - 可以灵活地指定显示在OPC-UA 命名空间中的symbols。
    - 支持附加的UA特性。例如：历史数据的访问
  - 缺点：略微多些开发的工作量

→ OPC-UA Server的默认配置

OPC-UA Server: Symbol file

- 如何使用过滤模式? → UA-Server默认设置
- 在项目开发工程中选择显示在OPC-UA 命名空间中的symbol
  - PLC：使用编译指令
  - C++：使用TMC代码编辑器



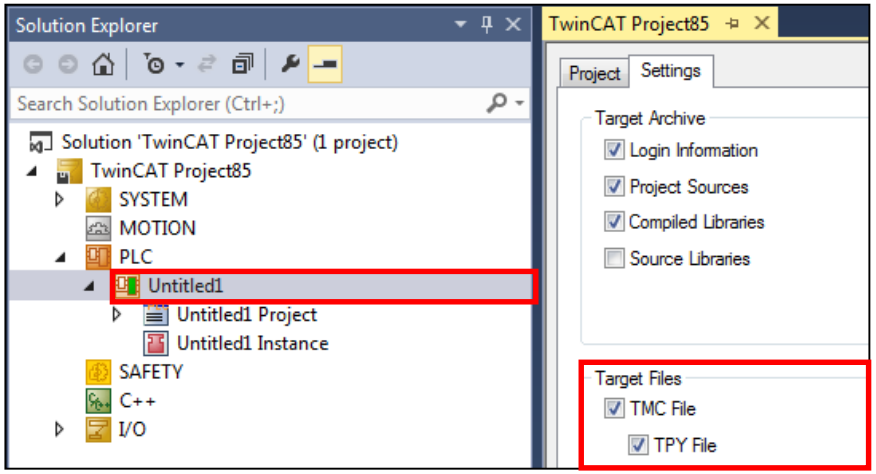
C++: TMC代码编辑器

```
bBool : BOOL;

{attribute 'OPC.UA.DA' := '1'}
Scanner1 : Scanner;

{attribute 'OPC.UA.DA' := '1'}
bBool2 : BOOL;
```

PLC: 将属性放置在symbol前 (TMC)

OPC-UA Server: Symbol file

- 在项目开发工程中由TwinCAT XAE自动生成
- PLC项目选项栏提供进一步的复选框来使能symbol file下载



- 复选框使能后下载symbol file到目标控制器的boot目录下

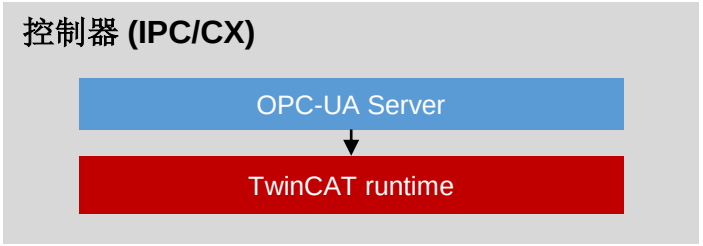
  e.g. *C:\TwinCAT\3.1\Boot\Plc\Port_851.tmc*

  \Hard Disk\TwinCAT\3.1\Boot\Plc\Port_851.tmc

- OPC-UA Server默认情况下导入这个symbol file

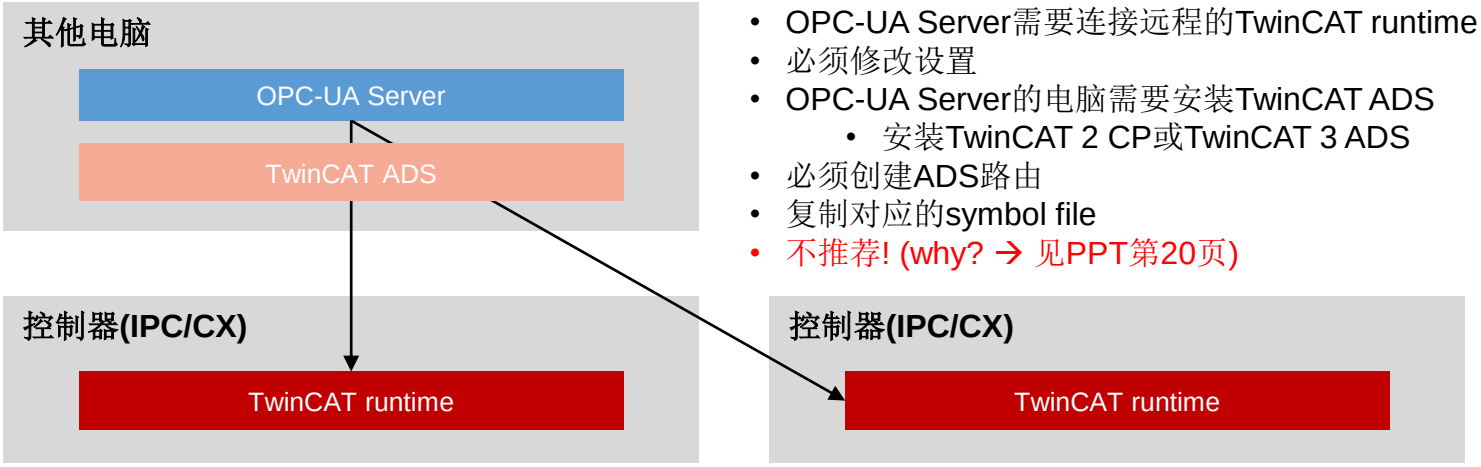
OPC-UA Server: 典型的设置方案

- OPC-UA Server和TwinCAT runtime安装在同一系统中：无需设置



- OPC-UA Server默认连接到第一个本地PLC运行内核中
- 默认的symbol file(Port_851.tmc)被使用
- 无需ADS路由
- 安装TF6100或CAB文件(WinCE) – 搞定!
- 推荐方案! (why? → 见PPT第23页)

- OPC-UA Server和TwinCAT runtime安装在不同的系统中：需要微小的调整

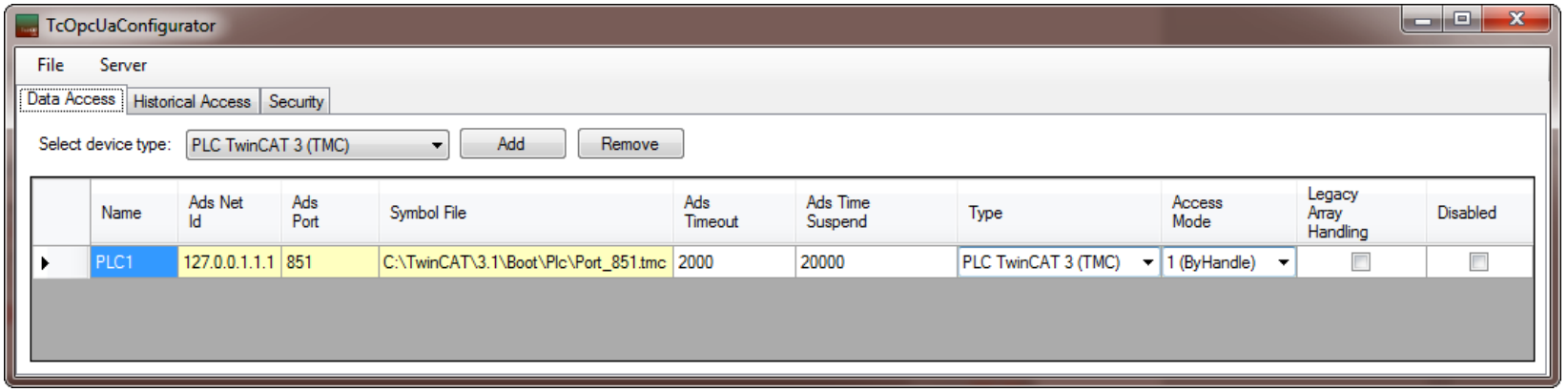


- OPC-UA Server需要连接远程的TwinCAT runtime
- 必须修改设置
- OPC-UA Server的电脑需要安装TwinCAT ADS
  - 安装TwinCAT 2 CP或TwinCAT 3 ADS
- 必须创建ADS路由
- 复制对应的symbol file
- 不推荐! (why? → 见PPT第20页)

OPC-UA Server: 配置

- OPC-UA Server可以配置多个TwinCAT Runtimes – 本地 和/或 远程
- 需要OPC-UA Server计算机和远程系统之间实现ADS访问

  → TwinCAT 2 CP或TwinCAT 3 ADS
- ServerConfig.xml 包含目标控制器的配置 (基于XML)
- 更简单的方法：使用OPC-UA Configurator ! (后面详细介绍)

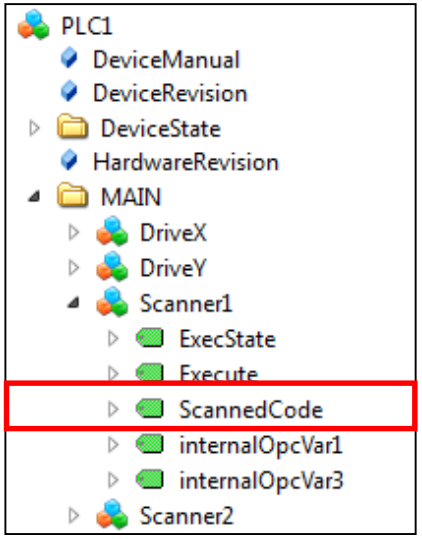


目录

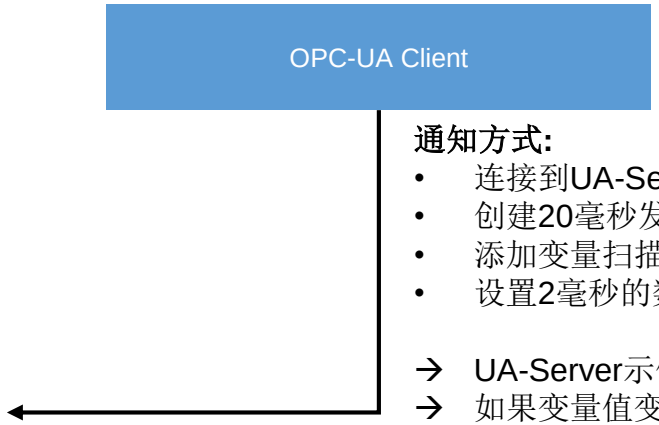
BECKHOFF

OPC-UA Server: 通知 vs. 轮询

- OPC-UA提供两种不同的通信模式：通知和轮询
- OPC-UA Client决定使用哪一种模式和更新频率



Reference	Target DisplayName
HasTypeDefiniti...	
HasInputVar	Execute
HasOutputVar	ExecState
HasLocalVar	internalOpcVar1
HasOutputVar	ScannedCode
HasLocalVar	internalOpcVar3



通知方式:

- 连接到UA-Server
- 创建20毫秒发布间隔的通知
- 添加变量扫描代码到通知
- 设置2毫秒的数据更新频率

→ UA-Server示例变量值(通过ADS)每2毫秒更新一次
→ 如果变量值变化，UA-Client得到UA-Server的通知
→ 通知每20毫秒一次 (发布间隔)

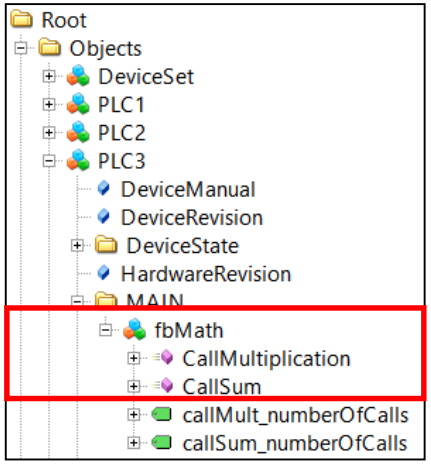
轮询方式:

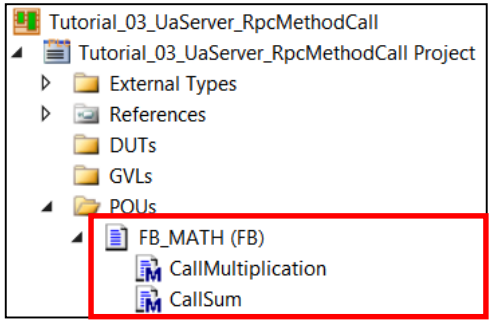
- 连接到UA-Server
- 读取变量的值 (可以周期性)

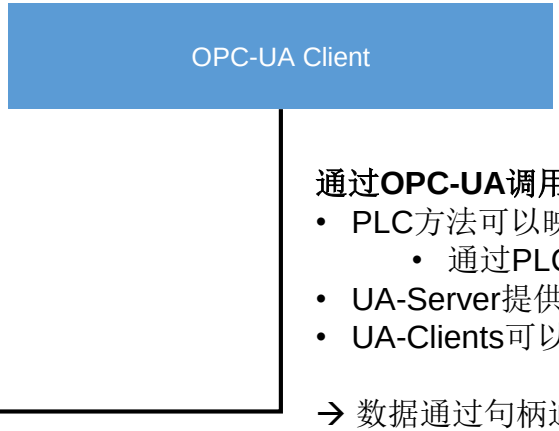
→ 在读请求时，UA-Server读取变量的值(通过ADS)
→ 响应会直接提交到UA-Client

OPC-UA Server: 方法调用

- OPC-UA有调用方法的功能
- TwinCAT 3能够将PLC方法映射到OPC-UA命名空间







通过OPC-UA调用方法:

- PLC方法可以映射到UA命名空间
  - 通过PLC方法中的属性
- UA-Server提供的PLC方法是UA方法
- UA-Clients可以调用这些方法

→ 数据通过句柄通信
→ 输入变量安全传输
→ 代码执行并等待结果
→ 输出结果传输给调用者

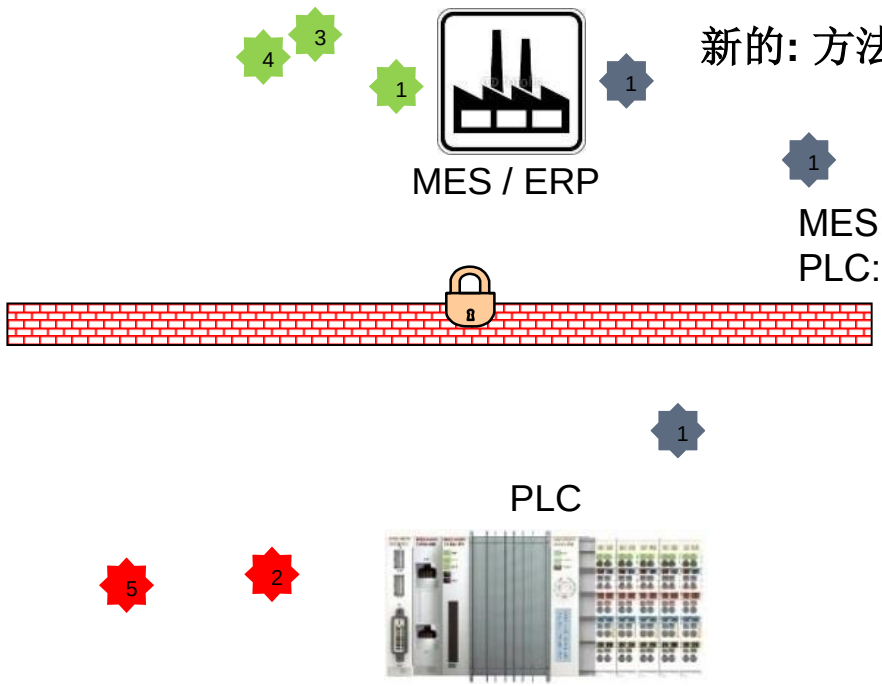
OPC-UA Server: 方法调用

典型应用案例：使握手通信更加有效

传统做法:

1. MES:
   “我想传送配方数据”
2. PLC:
   “可以”
3. MES:
   “这里有配方数据1”
   “这里有配方数据2...”
4. MES:
   “完成！请开始生产”
5. PLC:
   “收到”

耗时握手机制



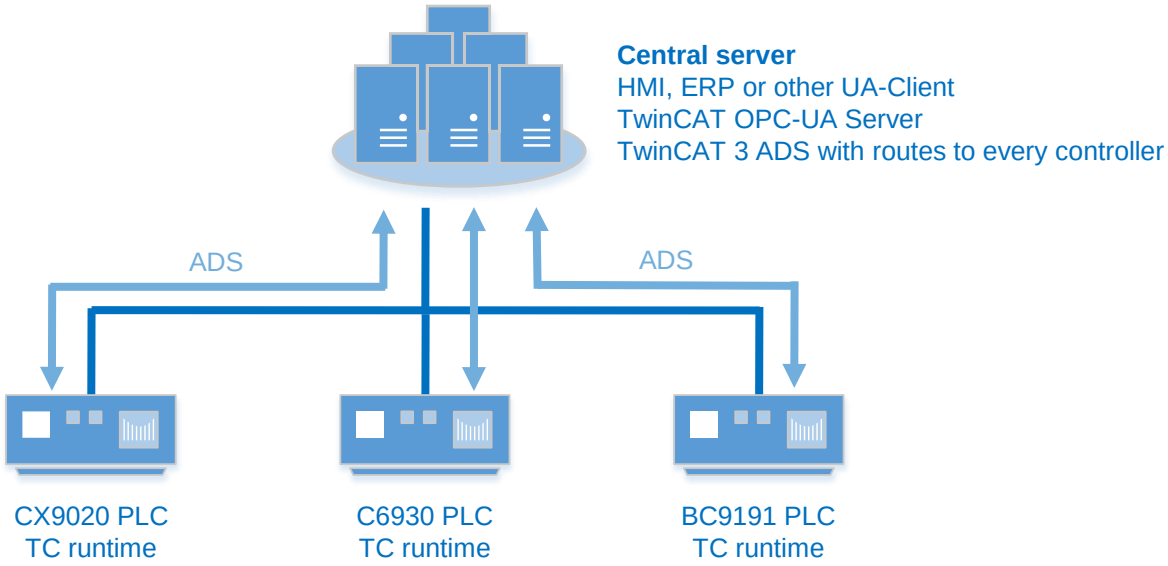
效率提升

新的: 方法调用

处理单一数据通信
- 输入数据安全传输
- 代码执行并等待结果
- 输出结果给调用者

OPC-UA Server: 练习 – 方案1 基本设置

- 传统方法:
  - 中央服务器提供HMI、UA-Server和TwinCAT 3 ADS (原TC2 CP)
  - ADS通过TwinCAT运行内核连接每一个控制器
  - 网络流量主要是ADS通信
  - OPC-UA只在HMI和OPC-Server之间的中央服务器通信
  - 需要和中央服务器交换Symbol files



OPC-UA Server: 练习 – 方案1 的问题

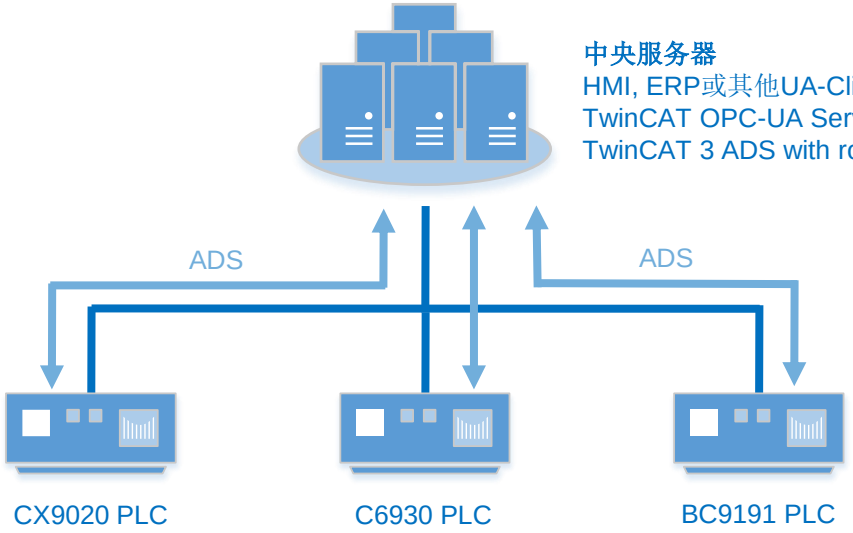
- 该方案的问题：
  - 中央服务器内存消耗高
    - 需要从每一个运行内核中导入symbol files
  - 大量的ADS通信造成网络负载过高 (见下一张PPT)
  - 中央服务器和控制器之间不能实现安全的数据传输
  - 配置步骤繁琐
    - 需要交换symbol files
    - 如果PLC的变量有修改，同时也需要修改中央服务器上的Symbol files

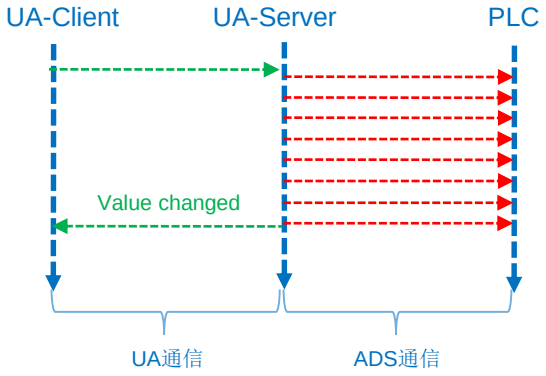
→ 旧的OPC-DA方法通常采用该方案 (因为DCOM的限制)

→ **我们不推荐此方案 (尽管它可行)**

(但是：这种方案节省了许可证成本， 只用安装一个中央TF6100)

OPC-UA Server: 练习 – 方案1的网络负载





- UA-Client创建申请
- UA-Server周期性查询PLC中的值是否变化!
  → ADS采样造成网络负载很高!
- 解决的方法是直接将UA-Server放在控制器上(ADS通信将保持在控制设备上，而不是网络上)

OPC-UA Server: 练习 – 方案2基本设置

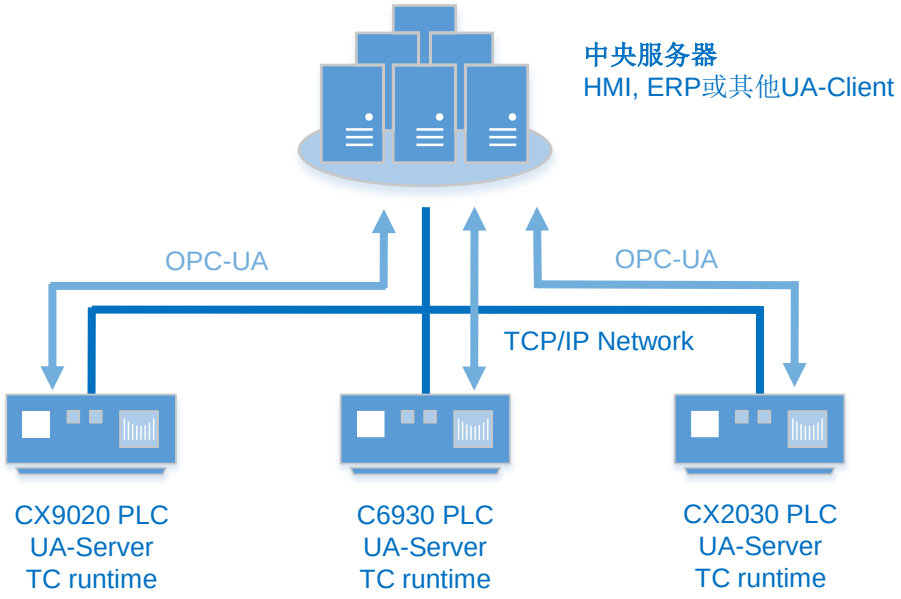
- OPC-UA方式
- 一个中央服务器只有OPC-UA Client
- OPC-UA Server直接安装在控制器上

→ ADS采样只在控制器上执行

→ OPC-UA通过网络通信

- → 优化了UA通信，网络利用率提升
- → UA的安全特性，使得安全性提升
- → 内存消耗分布在多个设备上
- → 不需要交换symbol files

→ 推荐的方案



目录

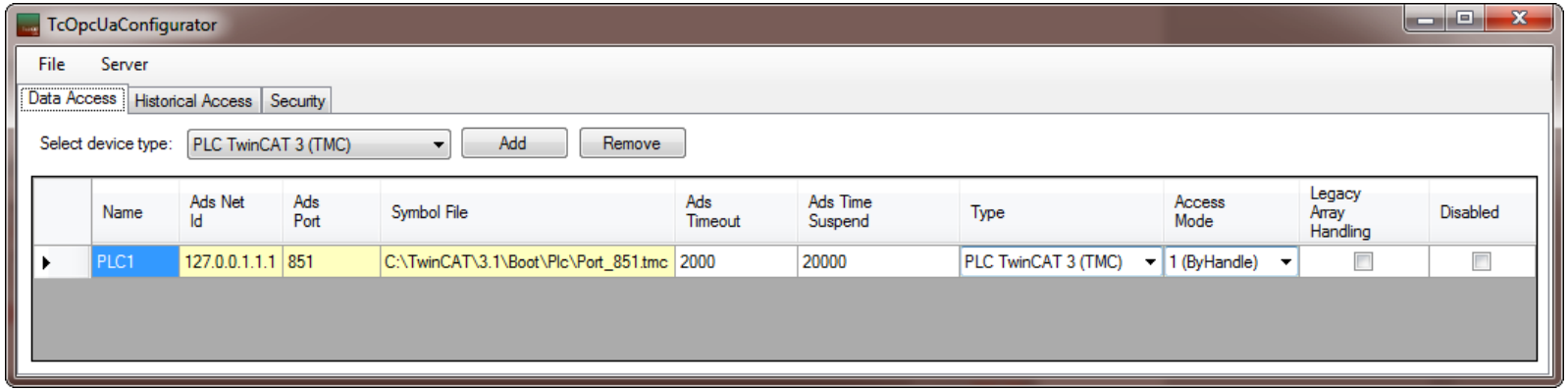
BECKHOFF

OPC-UA Configurator: 概述

- 注意：如果UA-Server和runtime在同一个系统，通常就采用默认配置

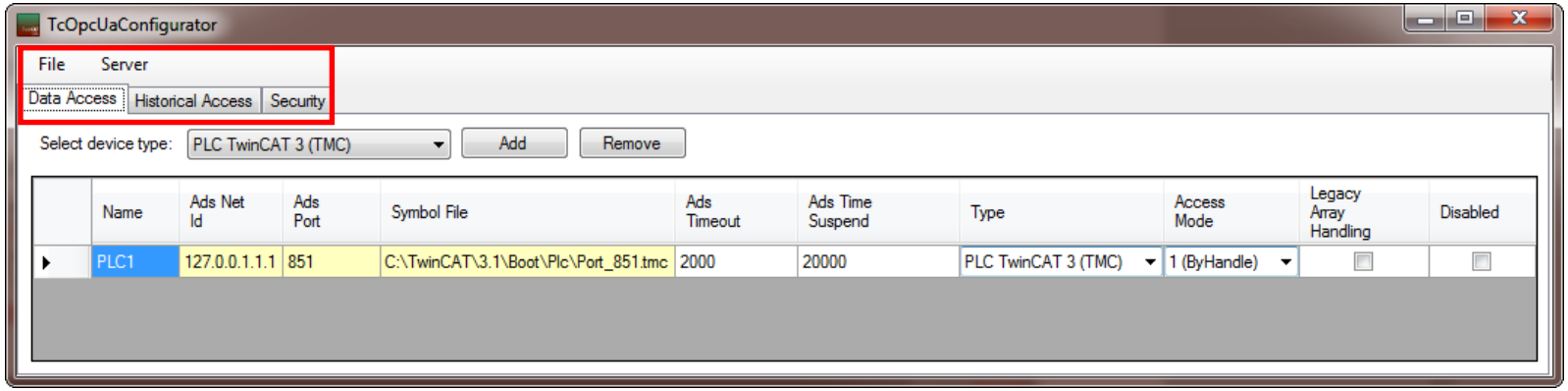
  → 默认情况下UA-Server连接到本地第一个PLC runtime(使用TMC symbol file)

- 高级配置 (例如：多个runtimes) 通过OpcUaConfigurator配置：
  - 在UA的命名空间配置多个TwinCAT runtimes
  - 改变UA-Server的安全特性 (端口，用户凭证，证书管理)
  - 激活\停用UA特性，例如：历史数据的访问
  - 运行在线诊断 → 连接到本地\远程的UA Server并确定其状态



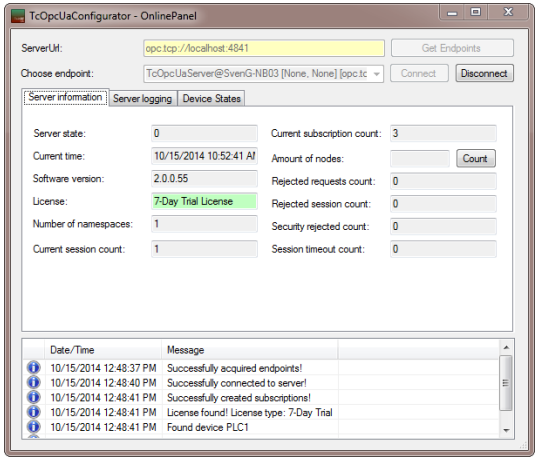
OPC-UA Configurator: 概述

- 通常，TcOpcUaConfigurator包含以下内容：
  - 数据访问
  - 历史数据访问
  - 安全性
  - 工具栏
    - 文件菜单： 打开，保存和激活配置
    - 服务器菜单：在线界面(诊断)，UA-Server重启功能

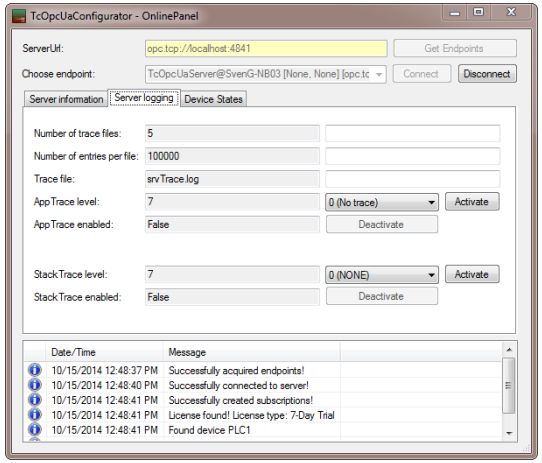


OPC-UA Configurator: 在线界面

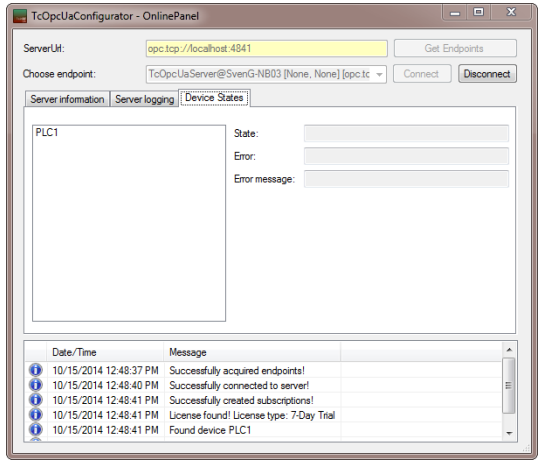
- 在线界面可以访问本地\远程UA-Servers的诊断信息
- 服务器信息选项卡：提供常规信息，例如：版本信息，节点数...
- 服务器日志选项卡：激活\关闭服务器日志记录
- 设备状态选项卡：检查连接TwinCAT runtimes的状态 (查错!)



服务器信息



服务器日志



服务器状态

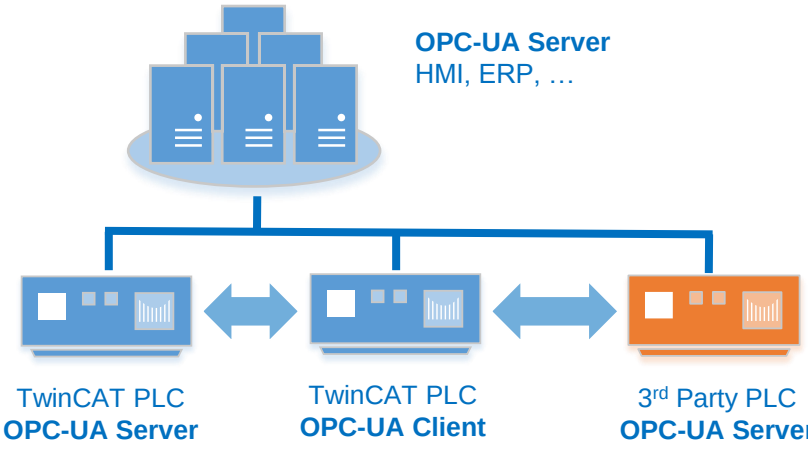
目录

BECKHOFF

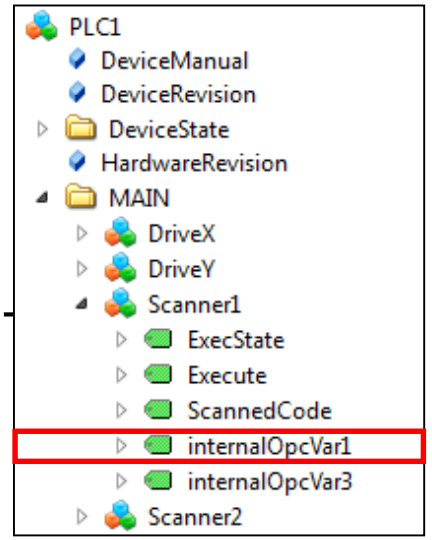
- 概述
- OPC-UA Server
- OPC-UA Server 深入介绍
- OPC-UA Configurator
- OPC-UA Client
- 总结

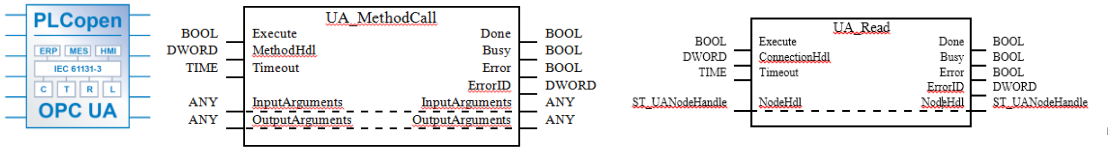
OPC-UA Client: 标准化通信 >控制器之间可以直接通讯<

- 控制器使用OPC-UA标准的PLCopen功能块来操控
- 已有超过10个PLCopen规范的功能块可使用
- 典型应用案例：与第三方设备进行数据交换



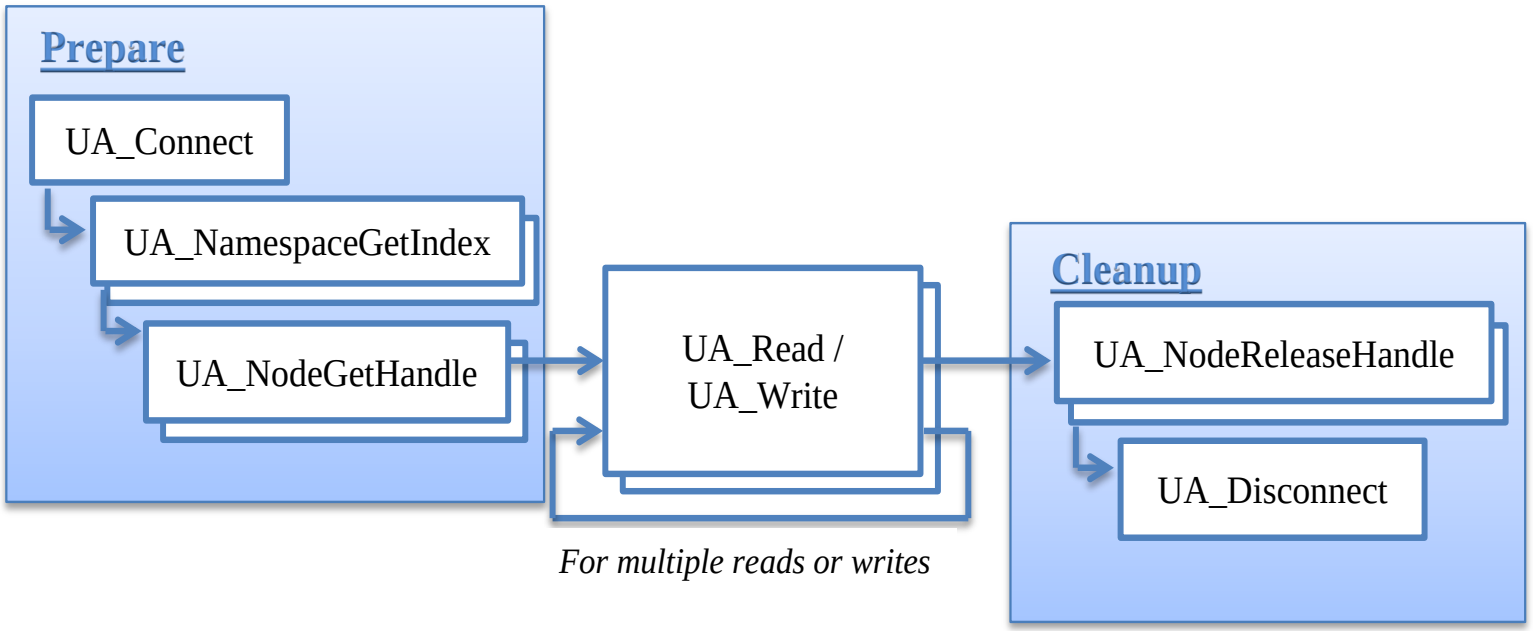
提供UA命名空间





示例：单个节点的读\写过程

- 连接时UA_Connect运行一次
- 访问命名空间时UA_NamespaceGetIndex运行一次
- 读写频率由系统决定



示例：单个节点的读\写过程

- 从UA命名空间中读取变量时需要哪些信息?
  - **UA_Connect**

    → IN: UA-Server的URL连接，例如：opc.tcp://localhost:4840

    → IN: 连接的端点 (有\无安全)

    → OUT: 返回连接句柄
  - **UA_NamespaceGetIndex**

    → IN: 连接句柄

    → IN: 变量所在命名空间的名称，例如：PLC1

    → OUT: 命名空间索引
  - **UA_NodeGetHandle**

    → IN: 连接句柄

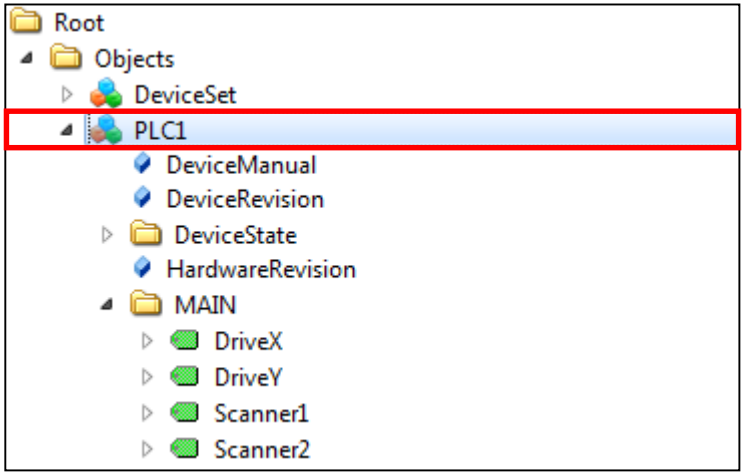
    → IN: 命名空间索引

    → IN: 变量的标识符，例如：MAIN.bBool (TwinCAT中)

    → OUT: 节点句柄

示例：单个节点的读\写过程

- 如何获取命名空间名字和标识符? (使用UA-Expert截屏)
- **命名空间名字:** 变量所在的位置?



- **标识符:** 什么是变量的唯一标识符?

目录

BECKHOFF

- 概述
- OPC-UA Server
- OPC-UA Server 深入介绍
- OPC-UA Configurator
- OPC-UA Client
- 总结

总结:

- TcOpcUaServer:
  - 用来通过OPC-UA将PLC数据发布到HMIs、ERPs...中
  - 使用symbol files获取symbol信息
  - 通过编译指令将PLC symbols映射到OPC-UA上
  - 使用方便：如果安装在运行内核计算机上就不需要配置
  - 可以通过TcOpcUaConfigurator配置更高级的方案
  - 推荐设置方案：安装在运行内核计算机上
    - 降低网络使用率
    - 更好的内存分配
    - 没有symbol file交换
    - …
- TcOpcUaClient:
  - 用来通过OPC-UA与其它设备交换数据
  - 使用PLCopen标准的PLC功能块