Case Study – SCADA Integration to SAP ERP TSW Module using SAP PI

PART-1: Business Case

Our client has one of the biggest pipeline connectivity network in Gas distribution, (Downstream – Delivery of Natural Gas through Pipeline & Gas Tankers) in the world. They have Decentralized SCADA Servers providing by different vendors like INVENSYS and TELVENT. These will get Pressure/Volume/Temperature/GCV/NCV etc. values from RTU’s (Remote Terminal Units)/Flow Meters attached across pipelines. Based on the values SCADA will control the GAS flow in Flow Meters remotely. Business case was to fetch these values in to SAP –TSW module and schedule the quantity of the Gas to be delivered on particular day. The integration is enabled through SAP PI (Process Integration).

Systems involved in the Landscape

- SAP ERP 6.0 EHP 4 TSW Module.
- SAP PI 7.1 EHP 1.
- SCADA with OPC DA Server (2.0) – Decentralized.
- OPC Gateways – OPC Clients.
SAP TSW (Trader's and Scheduler's Workbench) Module provides functions for stock projection and for planning and scheduling bulk shipments using nominations. TSW provides the relevant master data to model the supply chain.

SCADA (Supervisory Control and Data Acquisition) is not a full control system, but rather focuses on the supervisory level. As such, it is a purely software package that is positioned on top of hardware to which it is interfaced in general via Programmable Logic Controllers (PLCs) or other commercial hardware modules.

- **SCADA with OPC DA Server (2.0) – Decentralized:** As such, it is a purely software package that is positioned on top of hardware to which it is interfaced in general via Programmable Logic Controllers (PLCs) or other commercial hardware modules. There were multiple instances of these SCADA servers which are decentralized. But servers get synchronized with real time data on periodical intervals.

- **Communication with SCADA Server:** SCADA systems include OPC (OLE for Process Control) Servers inside, other than its own Data Sources. OPC Servers are software applications (drivers) that comply with one or more OPC specifications as defined by the OPC Foundation. OPC Servers communicate natively with one or more Data Sources on one side and with OPC Clients on the other. In an OPC Client / OPC Server Architecture, the OPC Server is a Slave while the OPC Client is the Master.

- **OPC Gateways (OPC Clients):** OPC Gateways systems are Client machines that will communicate to OPC Server in SCADA system. These Gateway systems with Windows Operating System and ISS installed & running. Communication between the OPC Client and OPC Server is bi-directional meaning the OPC Clients can read from and write to OPC Servers. An OPC Server can support communications with multiple OPC Clients simultaneously. Latest releases of OPC Servers support HTTP Communication and hence
Web Services that are provided by OPC Servers can be directly consumed by client applications like SAP PI or even SAP ERP directly.

- **Data format in SCADA**: Data in SCADA is available in the form of TAG’s, something like below:
  
  ```
  src./_at_dom_FoxSCADA/object/A1_AQT_03 -- Flow Rate
  src./_at_dom_FoxSCADA/object/A1AAPM_01 -- Pressure
  src./_at_dom_FoxSCADA/object/KT_FC2_VOL -- Volume
  src./_at_dom_FoxSCADA/object/PP_FC1_GCV -- GCV Gross Calorific Value
  src./_at_dom_FoxSCADA/object/PP_FC2_NCV -- NCV Net Calorific Value
  ```

  For each tag corresponding value will be updated to SCADA database by the RTU’s (Remote Terminal Units) connected to Gas Pipeline based on which SCADA will control the flow of Gas through Flow Meters attached to Gas Pipeline. OPC Client machines will request these tag values by specifying the tag name. Client machines got the facility to subscribe the functionality to get updated information of tag values, every second or even shorter time interval.

SAP PI (Process Integration) is the middleware through which the ERP makes a business request that consists of tag names. A tag name in terms of SCADA represents a variable to hold various quantitative measures like GCV (Gross Calorific Value), NCV (Net Calorific Value), Pressure, Volume, and Temperature. Each quantitative measure will have its own tag name. Because all the data stored in SCADA will be in the form of Tags, ERP system should send relevant Tag Names to get the values. As ERP sends a Proxy request this could be manipulated as a SOAP request by using SOAP Message Protocol as XI, which will point to AAE (Advance Adapter Engine). A sample request XML message is shown below.
PART-2: Systems Integration

On the high level SAP ERP system will request Tag values for particular SCADA OPC Server Instance, in response OPC Server will reply with Tag values, Quality of the Tag value, Quality of the Tag, OPC Server Status, and Time Stamp.

- **Deep drive in to the Systems Integration:**
  - As SAP ERP system makes a request and gets the response, so we made this as Client Proxy communication to SAP PI (Middleware System).
  - On the other hand SAP PI system is connected to OPC Gateway (Technically: OPC Client System) across the Firewall. In OPC Gateway system, a standard Web Service is exposed (this is provided by vendors who supply .NET Wrapper software like MatrikonOPC/Advosol). Hence, SAP PI will consume this Web Service at the receiver end.
  - A SOAP request sent by SAP PI will be received by service available in OPC Gateway, now this request has to be passed to OPC Server. In OPC Gateway, the web service request is mapped to a .Net program which will convert HTTP request to COM/DCOM Request and sends the same through ISS to OPC Server.
  - OPC Server will process the request and reply with the necessary Tag Values to the OPC Gateway.
  - OPC Gateway will map the response to SOAP response, which will then reach the SAP ERP through Client proxy response.

This is how synchronous communication happens between the Integrated Systems (SCADA & SAP ERP). Check the below illustration for details.
Deep Dive into the OPC Gateway (INTERNAL to SCADA system only):

- Client’s landscape has 2 OPC Gateway systems. Communication happens with either of the Gateway’s based on the availability.
- Each Gateway is linked to available instances of SCADA. OPC Gateways are manipulated to handle SOAP request coming from SAP PI and a .Net Client Application (Wrapper Software) in OPC Gateway converts the call in to COM/DCOM request and sends to OPCSERVER.
- The .NET applications can access OPC servers only through a software layer that is usually called a .NET Wrapper.
- There are no OPC standard specifications for a .NET interface, so different vendors offer .NET wrapper software with much different interfaces and features.
- One of the .NET wrapper software used was ADVOSOL XMLDA.NET wrapper. This supports access to OPC DA and XML DA servers.
- The Helper classes of wrapper software provide user friendly server access methods for features such as browsing and reading item values.
- **Web Service and Functions:** A SOAP request sent by SAP PI will be received by service available in OPC Gateway. We can send/test the Web Service Request from a web client like SOAPUI testing tool. Below is a Sample Web Service.

![Sample Web Service](image)

Sample Web Service Request: With the available Web Service we can **Read** / **Write** Tag Values available in OPC Gateway.

- Sample function, "**GetProperties**" can be used to get the Tag’s Properties like Quality and Time Stamp etc.
- Sample function, "**GetStatus**" can be used to check the status of the Tag value like ‘Active/In-Active’.

Once the Web Service is available, this can be used in SAP PI on receiving end.

Below is a Sample SOAP UI Request Message.

**PART-3: Building the Interface used for System Integration**

Interface with actual systems involved. I mean Sap ERP TSW, SAP PI, OPC Gateway Systems, and SCADA Servers (OPC Server).

As per the Business Case, the Interface needs to communicate with de-centralized SCADA servers. If there are 10 instances of SCADA Servers in the Landscape, then all the instances will contain same Tags and Values based on the periodic synchronization of the instances.
Our requirement was to fetch the values for specific tags based on the availability of the instances.

- **Interface Overview:**
  - A “Read Interface” has to be executed at a scheduled time every day to get the Tag values and store them in “Z Tables” (User-Defined Tables in SAP ERP).
  - These values are utilized by the module to finalize Allocations, based on which Ticket Creation, Delivery Creation and Billing Creation happens.
  - So to fetch the tag values in ERP, all the Tags relevant to the SCADA server instances are stored in “Z Tables” and used to create a **Proxy Request for Read operation** with the relevant tags.

- **Synchronous Communication (Onward Journey):**
  - In ERP TSW module, a “background program” executes every day at scheduled time and sends a client proxy request containing SCADA TAG’s to SAP PI.
  - In SAP PI, the “Read Interface” is executed which receives the client proxy request and transforms the message using operation mapping.
  - Resultant of this mapping will be mapped as an Operation – Message called **ReadSoapIn** of Web Service provided by OPC GATEWAYS.
  - Using conditional routing in the Read Interface, SAP PI sends to “ReadSoapIn operation” to available OPC GATEWAY using a SOAP request.
  - In OPC GATEWAY, a corresponding Web Service will receive SAP PI’s “ReadSoapIn request”, and sends the request to OPC SERVER in SCADA using .NET program and COM/DCOM technology.

- **Synchronous Communication (Return Journey):**
OPC SERVER processes the request and reads the values related to the requested TAGs, and sends the response back to OPC GATEWAY.
OPC GATEWAY will send the response using Operation – Message ReadSoapOut to SAP PI.
SAP PI receives the response using “ReadSoapOut” Message and this message gets transformed to ERP client proxy response.
ERP client proxy program update relevant table with TAG Values (in SAP ERP) for all OPCSERVERS available.

TAG Values in the ‘Z-table’ in SAP ERP will be used for finalizing allocations, ticket creation, delivery and billing. Below are few of the objects which are needed in SAP PI including Sender and Receiver Message Structures with Mappings in PI.

Sample XML – Request:

```xml
<payload>
  
  <?xml version="1.0" encoding="UTF-8"?>
  <ns0:MT_ReadOperation_OERG xmlns:ns0="urn://InPrim/ReadOperation/">
    <items itemPath="OPCSERVER11" itemName="src://at.dom.FoxSCADA/object://A1_APM101" clientItemHandle=" SAPOPCGW02" MaxAge="1000"/>
    <items itemPath="OPCSERVER11" itemName="src://at.dom.FoxSCADA/object://A1_APM104" clientItemHandle=" SAPOPCGW02" MaxAge="1000"/>
    <items itemPath="OPCSERVER11" itemName="src://at.dom.FoxSCADA/object://A1_AQM_02" clientItemHandle=" SAPOPCGW02" MaxAge="1000"/>
    <items itemPath="OPCSERVER11" itemName="src://at.dom.FoxSCADA/object://A1_AQM101" clientItemHandle=" SAPOPCGW02" MaxAge="1000"/>
    <items itemPath="OPCSERVER11" itemName="src://at.dom.FoxSCADA/object://A1_AQT_03" clientItemHandle=" SAPOPCGW02" MaxAge="1000"/>
  </ns0:MT_ReadOperation_OERG>
</payload>
```

Sample XML – Response:

```xml
  
  <?xml version="1.0" encoding="UTF-8" ?>
  <ns1:MT_ReadOperation_OERG xmlns:ns1="urn://InPrim/ReadOperation/">
    <items ItemName="src://at.dom.FoxSCADA/object://A1_AQT_03" Timestamp="2011-02-15T16:49:12.022+05:30" Value="740.059"/>
  </ns1:MT_ReadOperation_OERG>
```