

CASE STUDY — OEM PARTNERSHIP

IEC 61850 → DNP3 / MODBUS SOFTWARE GATEWAY

PC-based Protocol Conversion Software for
SCADA & Automation System OEMs

200+

Data objects mapped

9

Months to delivery

40%

Integration time saved

3

SCADA platforms supported

SECTOR SCADA / Automation System OEM

DURATION 9 months

PLATFORM Windows & Linux PC

Table of Contents

01	Project at a Glance	3
02	The Challenge	3
03	Engagement Model — Embedded R&D Partnership	4
04	Our Approach	4
05	What We Built — Software Components	5
06	Engineering Challenges Solved	6
07	Outcomes & Business Impact	8
08	Technology & Standards	8

01 Project at a Glance

Client type	SCADA / Automation System OEM
Target market	Transmission & distribution utilities
Product built	IEC 61850 → DNP3 / Modbus PC software
Deployment	Windows 10/11 · Ubuntu 20.04+ · any x86-64 PC
IEC 61850 role	MMS client + GOOSE subscriber
Output protocols	DNP3 Level 2 outstation · Modbus TCP/RTU slave
Data objects	200+ mapped — auto-built from SCD
SCADA platforms	3 supported — GE, Schneider, ABB systems
Engagement model	Embedded OEM product engineering
Duration	9 months
Integration time	40% saved on first field deployment
Reuse	3 substation deployments — single installer

02 The Challenge

An established SCADA and automation system OEM had a proven software platform deployed across dozens of control rooms — but it spoke legacy protocols only: DNP3, Modbus. Utilities rolling out new IEC 61850-native IEDs needed their existing SCADA to see that data immediately, without replacing the control-centre software or deploying dedicated hardware appliances. The OEM needed a PC-installable software component — a Windows and Linux service that runs alongside their existing platform, connects to IEDs over Ethernet, and presents the data as native DNP3 and Modbus to the SCADA layer above.

“We needed a software gateway our customers could install on the SCADA server they already have in the control room — no new hardware, no new rack space, no integration project. Just install, point at the SCD, and it works.”

03 Engagement Model — Embedded R&D Partnership

We worked as a fully embedded extension of the OEM's software R&D team — within their Visual Studio and CMake build environment, their Git workflow, and their QA process. The gateway was built as a standalone Windows service and Linux system service that installs alongside the OEM's existing platform. Division of responsibility was clear from day one:

OUR SCOPE

- IEC 61850 MMS client engine (full)
- GOOSE subscriber for fast status events
- SCD auto-parser, validator & hot-reload
- DNP3 Level 2 outstation (OpenDNP3)
- Modbus TCP + RTU slave
- Data mapping engine — config-file driven
- Windows service + Linux systemd daemon

OEM'S SCOPE

- SCADA platform UI and historian
- EMS / IEC uplink layer
- Customer-facing installer & documentation
- Licensing, pricing, and go-to-market
- Customer IT infrastructure & networking
- Site commissioning & field support
- SCADA operator training

04 Our Approach

A PC software — no hardware dependency

The gateway runs as a Windows service or Linux system service on any standard x86-64 PC — the same SCADA server the OEM's customers already own. No dedicated embedded hardware, no rack unit, no serial console. Installation is a single MSI on Windows or an apt/rpm package on Linux. This eliminated the hardware procurement, lead time, and logistics that make appliance-based gateways slow to deploy.

B SCD-driven auto-configuration — zero manual point entry

The entire DNP3 point database, Modbus register map, tag names, and alarm priorities are generated automatically at startup by parsing the substation SCD file. All IEDs, all logical nodes, all datasets — configured in seconds. When a new IED is added mid-commissioning, the operator drops the updated SCD into a watched folder; the service detects the change, hash-diffs the new version, adds the new points live, and opens a new MMS connection — no restart, no service interruption.

- C Config-file mapping engine — same binary, every substation**
Alldatamapping rules — scalefactors,deadbands, qualityflagtranslations, Modbus word order — live in an XML configuration file, not in compiled code. The OEM ships a single software binary to all customers. Site-specific mapping is handled by the configuration file, which the OEM's commissioning engineers edit without touching the source code. Three substations, three configuration files, one installer.
- D Source timestamp preservation — IED time in the historian**
IEC61850timestampscarrymicrosecondprecision fromtheIED'sPTP-disciplined clock. DNP3 events carry millisecond timestamps. The gateway preserves the IED source timestamp — truncated to milliseconds — in every DNP3 time-tagged event. The SCADA historian receives the original IED event time, not the PC clock receive time. Post-fault sequence-of-events analysis reflects actual field timing across all IEDs.

05 What We Built — Software Components

- 01 IEC 61850 MMS client engine**
ManagessimultaneousMMS connections to all IEDs in the substation. Each connection runs an independent BRCB subscription, sequence-number tracking, and auto-reconnect with event replay. Integrates libIEC61850 as the certified communication foundation — reducing protocol-layer risk and accelerating delivery. Exposed to the mapping engine via an internal async event queue.
- 02 GOOSE subscriber**
ReceivesGOOSE multicast frames directly from the NIC using raw socket capture, bypassing the TCP/IP stack for latency. Status changes from protection IEDs arrive in the DNP3 event buffer within milliseconds of the field event — far faster than MMS reporting. Supervises each GOOSE stream with a configurable timeout and sets COMM_LOST quality flags on stream loss.
- 03 SCD auto-parser & hot-reload**
Parsesthe substationSCDfile at startup to build the full point database automatically. Watches a configurable folder for SCD updates; on change, computes a hash-diff against the previous version, adds new points and connections live, removes stale ones gracefully. Three IEDs were added mid-commissioning on the first deployment without a service restart.

04 DNP3 Level 2 outstation (OpenDNP3)

Serves binary inputs (Group 1/2), analog inputs (Group 30), binary outputs (Group 10/12), and analog outputs (Group 40/41) to the SCADA master. Supports class 0 integrity polls, class 1/2 event polling, unsolicited responses, and DNP3 time synchronisation. Built on the OpenDNP3 library — interoperability validated against GE, Schneider, and ABB SCADA masters.

05 Modbus TCP + RTU slave

Serves holding registers (FC03) for measurements and coils (FC01) for status to legacy RTUs and HMI panels. Supports multiple simultaneous Modbus TCP client connections and serial RTU via USB-to-RS485 adapter. Modbus register map auto-generated from the same SCD configuration file that drives the DNP3 outstation — single source of truth, no spreadsheet drift.

06 Data mapping engine

Translates IEC 61850 data objects to DNP3 and Modbus point types using rules in an XML configuration file: scale factors, offsets, deadbands, quality flag mappings, Modbus word order, and DPS state handling. Validated against the SCD engineering unit at startup — a scale factor inconsistent with the declared unit raises a configuration error rather than silently producing wrong values.

06 Engineering Challenges Solved

01 GOOSE reception on a general-purpose PC NIC

Appliance gateways use dedicated hardware for GOOSE capture. On a standard PC, GOOSE multicast frames compete with TCP/IP traffic on the same NIC queue. We used raw socket capture at the OS network layer — bypassing the TCP stack entirely — and pinned the GOOSE receive thread to a dedicated CPU core with real-time scheduling priority. Measured GOOSE-to-DNP3-event latency on a loaded Windows Server: worst case 8 ms, well within the utility's 50 ms status update requirement.

02 Quality flag translation across three protocol models

IEC61850 has the richest quality model: validity, source, test, operatorBlocked, substituted. DNP3 has a simpler flags byte; Modbus has no quality model at all. We built an explicit, documented mapping: invalid → COMM_LOST, substituted → LOCAL_FORCED, test → CHATTER_FILTER, operatorBlocked → a dedicated binary status point. For Modbus, quality degradation is signalled by writing a configurable sentinel value to the affected register. The OEM's commissioning engineers received a one-page mapping reference at delivery.

03 Modbus register map consistency with runtime configuration

On the first prototype, the Modbus register map was documented in a spreadsheet maintained separately from the mapping configuration file. Within two weeks it had drifted. We resolved this permanently by generating the register map document automatically from the same XML configuration file that drives the runtime — a Python script runs at build time and produces a versioned PDF register map. Single source of truth: the config file is both the runtime and the documentation.

04 Per-point deadband — right from day one

The first release used a global deadband setting for all analog inputs. In the field, frequency required a tight deadband (0.01 Hz), transformer temperature a loose one (1°C), and active power a medium setting. Rather than retrofitting per-point deadband in a follow-up release — as we had done on a prior project — we built it into the XML configuration schema from the start. Each analog point declares its own deadband. Zero follow-up firmware work required after field commissioning.

07 Outcomes & Business Impact

FORTHE OEM

- ✓ IEC 61850 connectivity shipped as a software product — no new hardware SKU
- ✓ Single installer binary deployed across 3 substations — zero re-engineering
- ✓ Per-site onboarding: minutes vs weeks with previous manual approach
- ✓ 40% reduction in integration time on first field deployment
- ✓ Modbus register map auto-generated — documentation always in sync

FOR THE OEM'S UTILITY CUSTOMERS

- ✓ IEC 61850 native — runs on existing SCADA server, no rack space needed
- ✓ SCD hot-reload — new IEDs added live without control room disruption
- ✓ Source timestamps in historian — microsecond SOE precision preserved
- ✓ Multi-vendor GOOSE: GE, Schneider, ABB SCADA interop validated
- ✓ Quality flags documented and consistent across DNP3 and Modbus outputs

08 Technology & Standards

Protocols	IEC 61850-8-1 MMS · GOOSE · DNP3 Level 2 · Modbus TCP/RTU
Platform	Windows 10/11 service · Linux systemd · x86-64 · no hardware dependency
IEC 61850 stack	libIEC61850 (MMS client + GOOSE) · OpenDNP3 · SCD auto-parse
SCL / Config	SCD-driven auto-build · XML mapping rules · hot-reload · auto-generated register map

Need IEC 61850 connectivity in your SCADA software platform?

We build the gateway as a PC-native software component — Windows service or Linux service — so your customers install it on the server they already own. MMS client, GOOSE, DNP3, Modbus, SCD automation — end to end.

contact@ojasq.com | www.ojasq.com