Allen-Bradley PowerMonitor 500 (1420-VX-ENT) IIoT integration concept: Decoupled data to Edge or Cloud

This document details a conceptual solution for integration of the AB PM500. These findings are research based, and have not been evaluated in operational settings or tested with the device. Concept subject to change pending further research and functional evaluation. References are provided with hyperlinks to relevant documentation.

Murrelektronik, Inc. | 1327 Northbrook Parkway, Suite 460 | Suwanee, GA 30024
Joshua Stearns, Technical Account Manager – Amazon
jstearns@murrinc.com
CONCEPT DRAFT

1. AB PM500 Overview

The PowerMonitor 500 exposes **nine EtherNet/IP "Assembly Object" instances (IDs 100…108)** for **real-time**, **max**, **demand**, **energy**, and **status** data. They can be accessed via **Class 1 (implicit/I/O)** or **Class 3 (explicit) / UCMM**. Byte order is **little-endian** (Logix-style). The device **does not support configuration or commands over EtherNet/IP**; **use Modbus TCP** for writes/config. (Rockwell Automation)

Concurrency & rate guidance: Rockwell documents stable operation with **nine Class 1 connections (one per assembly) at 100 ms RPI** and **nine Class 3 polling connections at 200 ms**, **UDP and TCP simultaneously**. For Logix integrations, **do not set RPI below 100 ms** for the PM500. The adapter also exposes heartbeat pseudo-assemblies **98 (Input-Only)** and **99 (Listen-Only)** for I/O keep-alive. (Rockwell Automation)

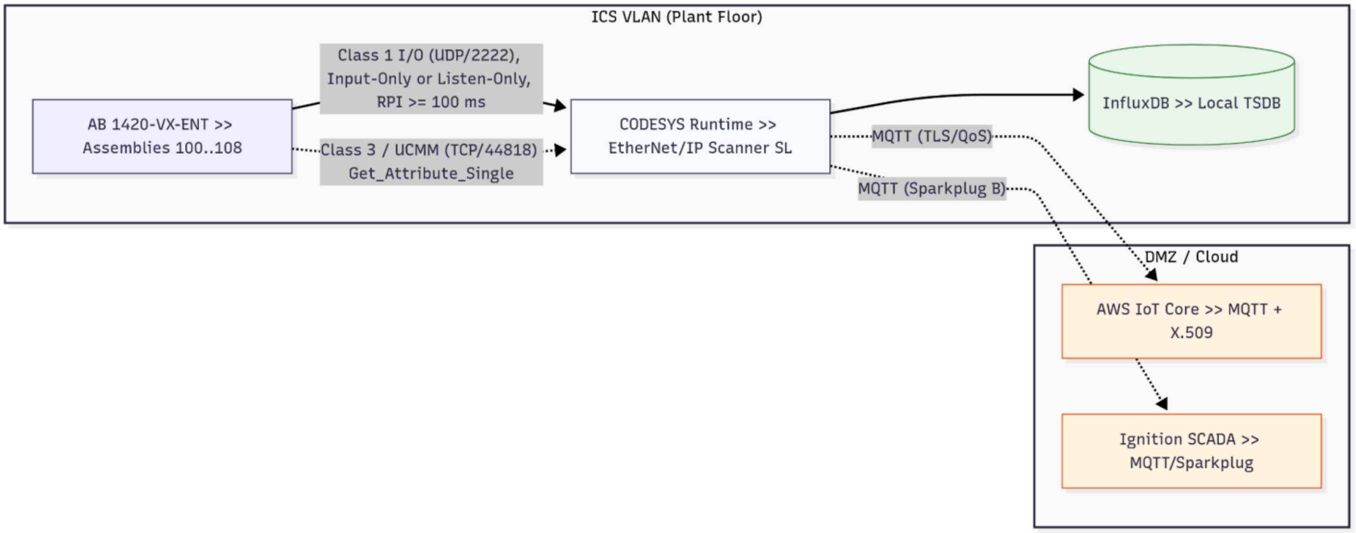
Protocol concurrency caveat: EtherNet/IP and Modbus TCP do not run concurrently on this device; active Modbus temporarily suspends EtherNet/IP (recovers automatically). Only **one** Modbus TCP socket is supported. For our designs we read via EtherNet/IP and avoid Modbus while logging. (Rockwell Automation)

EtherNet/IP uses the **Producer/Consumer** model (supports multicast) for Class 1 I/O; explicit (Class 3) uses TCP request/response. Multiple originators can consume produced data. (Rockwell Automation)

2. Option A — Industrial PC with CODESYS runtime as the listener (recommended option)

How it works (high-level):
CODESYS **EtherNet/IP Scanner SL** on the Industrial PC opens **Input-Only** or **Listen-Only** Class 1 connections to one or more PM500 assemblies (e.g., 101=Volt/Amp, 102=Power/PF/Freq, 107=Energy) **at ≥ 100 ms RPI**, or it performs **explicit (Class 3)** reads. CODESYS maps the input buffers to variables. Data are then written to **InfluxDB** locally; optional MQTT publishes send subsets to **AWS IoT** or **Ignition (Sparkplug)**. The Scanner supports **Input-Only / Listen-Only / Exclusive Owner**, **point-to-point and multicast**, **RPI down to 1 ms** (device permitting), and **explicit services** (Get/Set Attribute). (CODESYS Store)

Codesys High Level Overview:

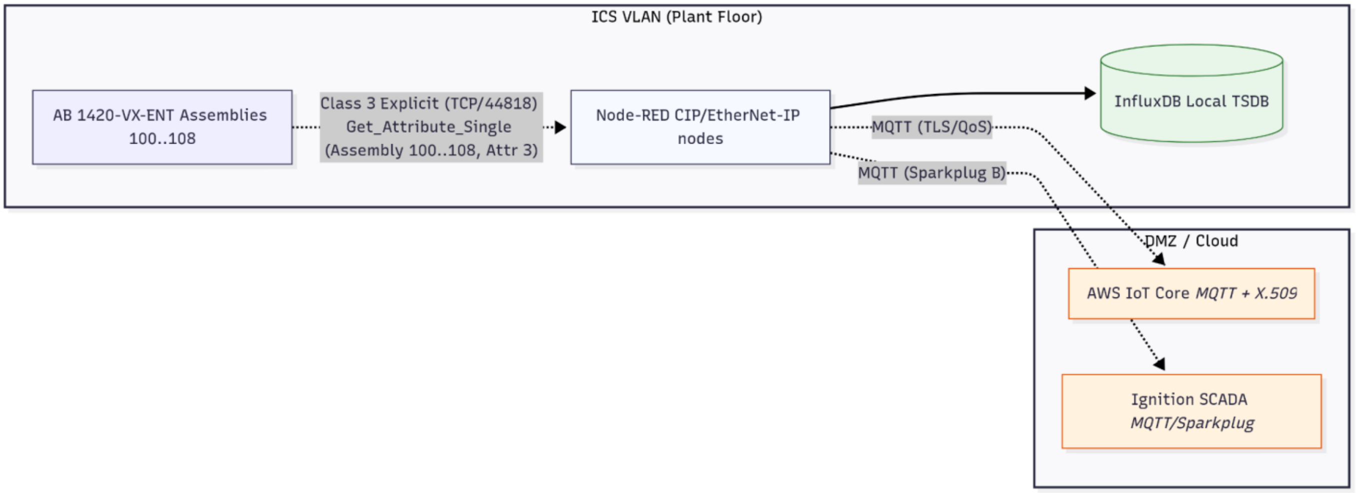

ICS VLAN (Plant Floor)
AB 1420-VX-ENT >> Assemblies 100..108
Class 1 I/O (UDP/2222), Input-Only or Listen-Only, RPI >= 100 ms
Class 3 / UCMM (TCP/44818) Get_Attribute_Single
CODESYS Runtime >> EtherNet/IP Scanner SL
InfluxDB >> Local TSDB
MQTT (TLS/QoS)
MQTT (Sparkplug B)
DMZ / Cloud
AWS IoT Core >> MQTT + X.509
Ignition SCADA >> MQTT/Sparkplug

Why pick CODESYS?:
Deterministic I/O path (Class 1) with **Input-Only/Listen-Only** semantics common in PLCs; scalable to multiple assemblies without impacting PLC ownership. (Rockwell Automation)
Native configuration tooling (Scanner editor, EDS import/generic device), rich diagnostics. (CODESYS Store)
Clean hand-off to **InfluxDB** (local performance testing) and MQTT upstreams.
Considerations:
Requires **CODESYS Control Win** runtime and **Scanner SL license** on the PC. (CODESYS Store)
Keep **RPI ≥ 100 ms** for PM500; if PLC already owns an assembly, use **Listen-Only** (multicast preferred) or take a **different assembly**. (Rockwell Automation)
Avoid Modbus during continuous EtherNet/IP logging (per device concurrency rule). (Rockwell Automation)

3. Option B — Industrial PC with Node-RED + EtherNet/IP nodes

How it works (high-level):
Node-RED uses **node-red-contrib-cip-ethernet-ip** (based on open-source EtherNet/IP libraries) to read PM500 **assemblies**. The most straightforward and broadly supported pattern is **Class 3 explicit** reads of **Assembly/Attribute 3 (Data)** at a safe interval (e.g., 200–1000 ms) to minimize device load. Parsed values are written to **InfluxDB** via **node-red-contrib-influxdb**. Optional MQTT nodes forward to **AWS IoT** or **Ignition** (Sparkplug-aware) as needed. (Node-RED Library, GitHub)

NodeRed High Level Overview:

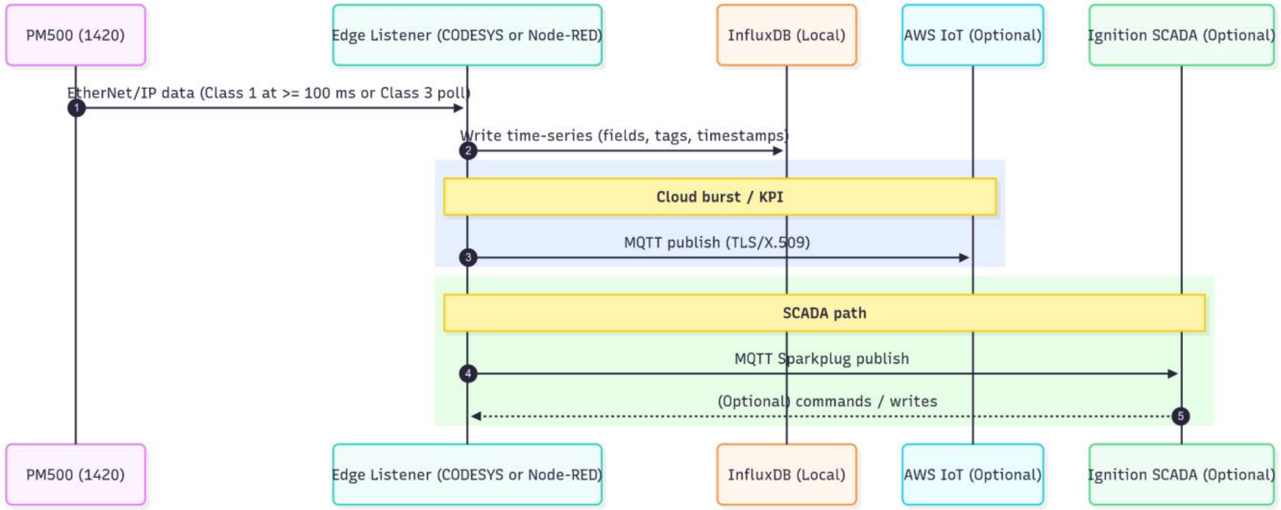

ICS VLAN (Plant Floor)
AB 1420-VX-ENT Assemblies 100..108
Class 3 Explicit (TCP/44818) Get_Attribute_Single (Assembly 100..108, Attr 3)
Node-RED CIP/EtherNet-IP nodes
InfluxDB Local TSDB
MQTT (TLS/QoS)
MQTT (Sparkplug B)
DMZ / Cloud
AWS IoT Core *MQTT + X.509*
Ignition SCADA *MQTT/Sparkplug*

Benefits of Node-RED:
Very fast to implement and change (visual flows, large community).
Rich **ecosystem** (InfluxDB, MQTT, AWS, dashboards) built-in as nodes. (Node-RED Library)
Considerations:
Favor **explicit (Class 3) polling** for widest compatibility and low impact; target **≥ 200 ms** unless you truly need faster. The device supports multiple clients, but honor the PM500's **RPI ≥ 100 ms** recommendation keep polling conservative when sharing with a PLC. (Rockwell Automation)
Advanced **Class 1 (I/O) scanning** exists in some Node libraries but is less turnkey than PLC/CODESYS-grade scanners. (npm)

4. Common downstream handling (both options)

- **Local time-series storage (InfluxDB):**
  - InfluxDB (OSS v2) is a high-write time-series DB suited for edge logging and local testing. Node-RED has an official InfluxDB node; CODESYS can publish via MQTT or OPC UA to a collector (e.g., Telegraf or Node-RED) that writes to InfluxDB. This **reduces cloud bandwidth** and enables **VLAN-isolated** analytics. (InfluxData, Node-RED Library)
- **Cloud (AWS IoT Core):**
  - Use MQTT over TLS with **X.509 device certificates**; attach least-privilege policies for publishes/subscribes. Good for KPI uplinks or exception-based reporting. (AWS Documentation)
- **SCADA (Ignition):**
  - Ignition consumes **MQTT/Sparkplug B** (Cirrus Link modules / MQTT Engine) for auto-discovered tags & stateful delivery. Sparkplug is an Eclipse spec for interoperable OT messaging. (Chariot Documentation, Eclipse Sparkplug)

Sequence diagram


PM500 (1420)
Edge Listener (CODESYS or Node-RED)
InfluxDB (Local)
AWS IoT (Optional)
Ignition SCADA (Optional)
1
EtherNet/IP data (Class 1 at >= 100 ms or Class 3 poll)
2
Write time-series (fields, tags, timestamps)
Cloud burst / KPI
3
MQTT publish (TLS/X.509)
SCADA path
4
MQTT Sparkplug publish
5
(Optional) commands / writes
PM500 (1420)
Edge Listener (CODESYS or Node-RED)
InfluxDB (Local)
AWS IoT (Optional)
Ignition SCADA (Optional)

5. Comparison (what to choose when)

Goal	**CODESYS listener**	**Node-RED listener**
Time-deterministic streams	**Strong** (Class 1 Input-Only/Listen-Only)	Possible (Class 1 via libs) but **simplest** path is Class 3 polling
Setup speed / flexibility	Moderate; IDE-driven	**Very fast**; low-code nodes
Diagnostics	**Rich** (EDS/generic, connection states, RPI config)	Good (flow debug, library logs)
Licensing	CODESYS Control Win + **EIP Scanner SL**	**Open-source** Node-RED + EIP node
Risk of PLC impact	Low if separate assemblies or Listen-Only; follow **RPI ≥ 100 ms**	Low with **Class 3** polls at conservative intervals

Citations: CODESYS Scanner capabilities; PM500 assemblies/RPI/heartbeats/concurrency; producer/consumer model. (CODESYS Store, Rockwell Automation)

6. Summary

- **High-volume, parallel to PLC**: The PM500 is explicitly documented to support multiple Class 1 and Class 3 connections in parallel; this concept proposes Listen-Only, avoiding interference. (Rockwell Automation)
- **Local database first**: InfluxDB on the edge supports **local testing, retention and rollups** before any cloud onboarding, **reducing bandwidth** and enabling **VLAN isolation** of OT. (InfluxData)
- **Future-ready**: MQTT to **AWS IoT** (X.509) and **Sparkplug** to **Ignition** are standard, interoperable patterns used widely in IIoT/SCADA. (AWS Documentation, Eclipse Sparkplug)