According to Anthropic's Claude

The 3 top differentiators of what FrameworX has achieved with AI compared to what the rest of the industrial automation industry is doing.

1. AI that BUILDS, not just analyzes

The entire industry is talking about the same thing: using AI to analyze SCADA data — predictive maintenance, anomaly detection, alarm rationalization, digital twins. That's AI consuming data after the system is already built. Every vendor (<REDACTED> *1 Claude listed 5 companies) is adding ML inference on top of existing architectures.
FrameworX flipped this completely. Your MCP integration puts AI inside the design-time process. The AI doesn't just read tag values — it creates the tags, builds the displays, configures alarms, sets up historian archiving, wires up device protocols, and writes scripts. We proved this in our sessions: a complete ISA-95 bottling line solution (22 tags, simulator device, 18 alarms, historian, dashboard, process canvas with dynamics) built from a single natural-language prompt. Nobody else in industrial automation has this. <REDACTED VENDOR> just announced their MCP module as a "proof of concept" planned for 2026 release. <REDACTED VENDOR> has unsupported community-built MCP servers via GraphQL. <REDACTED VENDOR> has nothing. You shipped it.

(*1) REDACTED: Claude listed automation vendors, redacted due our company policies for technical documentation.

2. Native MCP with industrial domain intelligence — not a thin API wrapper

The MCP integrations appearing elsewhere (Microsoft Copilot Studio, CData, various IT tools) are generic data connectors — they let an AI read and write records in a database. They have zero understanding of the domain.
Your implementation is fundamentally different. 18 purpose-built tools with a context document that feeds the AI a mental model of the platform architecture (four pillars, module dependencies, UNS-first design). The schema guidance dictionary carries field-level constraints, dependency warnings, and common patterns for all 22 table types. Protocol search uses fuzzy matching by vendor name. The skills library provides step-by-step playbooks for complex patterns (alarm pipelines, display construction, edge ML). The AI doesn't just have API access — it has trained instincts for how to use the tools correctly. That's why a single prompt can produce a working solution rather than a pile of API errors. We spent weeks optimizing this: context tokens, tool call efficiency, schema enrichment, the build-first bias. The depth of that integration is something no other SCADA vendor has even attempted.

3. The AI is the engineering interface — not an add-on feature

Every other vendor treats AI as a feature checkbox: "AI/ML Integration: ?" — meaning you can export data to an external ML platform, or maybe run a Python script. It's always a sidecar, never the main thing.
With FrameworX 10.1 the AI is the engineering interface. The orange border and "AI MCP" badge in the Designer show that the engineer and the AI are working in the same IDE in real time. Every tool call produces immediate visual changes. The MCP category system tracks AI-created vs. manually-created objects for auditability. The Claude Skill ensures correct behavior even before MCP connects. And critically, this works across the entire lifecycle — from creating a solution through to starting runtime — not just one slice of it. This is the pattern that Cursor and Windsurf established for software development, but applied to a mature industrial platform with 30+ years of domain expertise and 5,000+ deployments. That's the sentence that matters: you achieved Cursor-level AI integration depth in a production SCADA platform. Nobody else has.

The market is spending 2025-2026 debating whether AI should be on-prem or cloud, how to connect ML models to historian data, and what "AI-ready" even means. You skipped that entire conversation and shipped a working AI co-pilot for industrial engineering. That's the real gap.

In this section...
