

Build Your AI Data Layer

In 7 stappen - van losse PLC-data naar een fabriek die AI begrijpt

"You're missing one layer - and AI can't fix what it can't understand."

Voor PLC/SCADA engineers die hun fabriek AI-ready willen maken
zonder vendor-lock en zonder €40K aan licenties

Lees-tijd: 15-20 min · Doe-tijd stap 1: 30 min

[TECHFLOW24.COM](https://techflow24.com)

© 2026 TechFlow Industrial

Build Your AI Data Layer

In 7 stappen - van losse PLC-data naar een fabriek die AI begrijpt

"You're missing one layer - and AI can't fix what it can't understand."

Deze gids is voor jou als PLC/SCADA engineer of OT-architect die ziet dat AI in jouw fabriek geen voet aan de grond krijgt - en wil weten hoe je dat oplost. Niet door €40.000 per jaar aan AVEVA Connect te betalen. Niet door een nieuwe historian te kopen. Door zelf, in 7 stappen, de laag te bouwen die ontbreekt.

Lees-tijd: 15-20 min. Doe-tijd voor stap 1: 30 min.

Het probleem kort samengevat

Je fabriek genereert terabytes data uit PLCs, SCADA, historians en ERP - en kan toch geen drie beslissingen noemen die deze week op basis daarvan genomen zijn.

Dat is **de missing layer**: niet meer dashboards of tools, maar **de semantische laag die raw machine-data omzet in informatie waarop een mens of AI kan handelen - binnen het tijdsvenster waarin het er nog toe doet.**

Deze gids laat zien hoe die laag eruitziet. In 7 stappen.

Voor je begint - de Solve-toets

De grootste fout in industriële data-projecten: beginnen bij de data, hopen dat actie er ooit uit volgt. Doe het andersom. Begin bij **wat moet er fysiek veranderen in de fabriek?** - dan bouw je terug.

We noemen dat de Solve-toets. Voordat je gaat bouwen, schrijf één concreet event op in dit sjabloon:

"Ik wil [WIE] in staat stellen om [WAT TE BESLUITEN OF DOEN] binnen [HOE SNEL] na [TRIGGER]."

Voorbeeld uit BakeryWorks Utrecht (de fictieve bakkerij die we doorlopen):

"Ik wil de shift-leider in staat stellen om de oven-temperatuur correctie te valideren binnen 4 minuten na: tunnel-oven zone 3 daalt onder 195°C voor >30 sec terwijl er product in zone 3 is."

De 5 criteria om elk Solve-event mee te toetsen:

1. **Closed loop** - gaat de actie terug het systeem in zodat je achteraf weet wat er gebeurde?
2. **Owner identified** - is er één specifieke persoon (of systeem) verantwoordelijk?
3. **Time-bound** - heeft de actie een venster waarin het er nog toe doet?
4. **Measurable outcome** - kun je achteraf vaststellen dat er fysiek iets is veranderd?
5. **Traceable to source** - kun je vanaf de actie terug naar de raw machine-data?

Als één van de vijf "nee" is, dan bouw je een data-museum. Niet een data-layer.

Belangrijk: Solve is **geen 8e stap**. Solve is de toets die je op de hele 7-stappen build legt. Het is wat de build af maakt.

Goed. Met je Solve-event op papier, hier zijn de 7 stappen.

Stap 1 - Connect

Raw signalen uit PLC/SCADA/sensoren ophalen - geen interpretatie, geen modellering. Alleen: data uit zijn silo.

Open-source toolset: OPC-UA voor moderne PLCs, MQTT via **MonsterMQ** voor edge, Modbus TCP voor legacy.

Valkuil: *"we verbinden alles voor het geval dat". Verbind alleen wat je Solve-event nodig heeft.
Drie tot vijf tags, niet tweehonderd.*

Stap 2 - Condition

Raw data fatsoeneren: deduplicatie, smoothing, time-align (UTC + ISO 8601), engineering unit conversion.

Raw PLC-data is rauw - een sensor stuurt elke seconde, ook als er niets verandert. Een DCS publiceert in lokale tijd zonder DST-correctie. Conditie is geen doel; het is een filter dat downstream beslist wat zinvol is.

Valkuil: *"standaardiseer alles". Conditie is een filter, geen rapport.*

Stap 3 - Model

Dit is het hart van de hele build. Raw tags krijgen betekenis door ze in een **Unified Namespace (UNS)** te plaatsen - een hiërarchische taal die de hele fabriek deelt. Onder de motorkap: ISA-95 als ontologie (Enterprise → Site → Area → Work Center → Equipment → Tag).

Een tag heet niet meer `plc-oven-a/db100/dbw14 = 2400` . Hij heet:

```
bakery-works-utrecht/line-a/baking/tunnel-oven-01/zone-3/temperature = 240.0°C
```

Een mens snapt dat direct. Een AI-agent ook. Zonder Model-stap heb je nog steeds bytes - alleen schoner.

Valkuil: *ISA-95 als compliance-papier behandelen. Het is een werkende taal, niet een rapport.*

Stap 4 - Store

Opslaan: time-series voor metingen + event-store voor Solve-events.

MongoDB voor flexibele documents - werkt prima als beide tegelijk, één database. Retention policy per topic; niet alles 7 jaar bewaren.

Valkuil: *alles in raw PLC-poll-frequency bewaren. Niemand heeft 1Hz data uit 2018 nodig.*

Stap 5 - Orchestrate

Event-driven workflows. Een sensor verandert → een trigger gaat af → een actie (of een beslissing-aanmens) wordt uitgevoerd.

Hier wordt data **informatie**. Data-driven (verzamel alles, analyseer later) is niet hetzelfde als event-driven (definieer wat moet gebeuren, verzamel wat daarvoor nodig is). Event-driven verslaat data-driven omdat het Solve-events bouwt, niet rapporten.

Toolset: **N8N** voor workflows. Niet voor data-routing - dat doet MonsterMQ.

Valkuil: *Node-RED. Voor productie-OT is N8N's flow-versioning, error-handling en UI volwassener.*

Stap 6 - Visualize

Dashboards die per persona zijn afgestemd. Operator, shift-leider, planner, plant-manager - elk een eigen view.

Grafana voor time-series + drilldowns. Eén dashboard per persona, geen "alles-in-één". Threshold-overlays direct in de grafiek (Solve-trigger lines). Live events feed als tabel onderaan.

Valkuil: *PowerBI als visualisatie-laag in OT - te traag voor real-time, te zwaar voor branche-engineer onderhoud, te vendor-lock.*

Een dashboard zonder beslissing is wallpaper. Eén mét beslissing is een control-room.

Stap 7 - Distribute

API's voor downstream consumers - andere apps, BI-tools, AI-agents, mobile apps voor operators.

FastAPI voor REST endpoints. WebSocket voor live-streaming. Webhooks voor outgoing notifications.

Deze endpoints zijn de **producten** die je data-layer levert. Daarop bouw je Teams-bots, mobile apps, AI-agents.

Valkuil: *alle endpoints publiceren voor "later". Publiceer alleen wat een echte consumer aanvraagt - anders bouw je technical debt.*

Je data-layer is alleen waardevol als hij gebruikt wordt. Een endpoint zonder consumer is dead code.

Solve - de toets terug

Je hebt nu een werkende data-layer op een fictieve bakkerij. Maar - is het Solve? Loop het door:

CRITERIUM	BAKERYWORKS SOLVE-A	STATUS
Closed loop	Shift-leider ack-t in Teams, response landt in MongoDB	✓
Owner identified	Dienstdoende shift-leider uit ERP shift-rotatie	✓
Time-bound	4 min response, daarna escalatie plant-manager	✓
Measurable outcome	Batches gered per kwartaal × €batch-waarde	✓
Traceable to source	Action-log → MES → MQTT → PLC tag	✓

Vijf vinkjes. Werkende data-layer.

Onthoud: een fabriek met perfecte ISA-95 modellen en schitterende dashboards maar zonder Solve-events heeft geen data-layer. Die heeft een **data-museum**.

Wat nu?

Drie wegen vooruit:

OPTIE 1 - BOUW HET ZELF IN 6 MAANDEN

Alle code is open source. De stack draait op €8/mnd VPS. Je hebt 15 jaar OT-ervaring - je kunt dit. Wat het van je vergt: - 14 weken vol-time of 6 maanden parttime - Geen review op je architectuur (je leert door fouten) - Geen community van peer-engineers die hetzelfde doen - Zelf de Python/Docker/REST API skills aanleren

OPTIE 2 - DOE DE LIVE WORKSHOP

60-min live online workshop. Volgende sessie: juni 2026. Live demo van de IDP-stack (Connect → Distribute), je vult ter plekke een 16-item AI-readiness scorecard in, en je verlaat de sessie met een gepersonaliseerde gap-analyse voor jouw plant.

OPTIE 3 - HET ONLINE PROGRAMMA - BUILD YOUR AI DATA LAYER

14 weken, één module per week: - Wekelijkse 60-min review-call van Johannes op JOUW architectuur - Community van 20-25 PLC/SCADA engineers op zelfde reis - ISA-95 templates per branche (brouwerij, chemie, metaal, food, pharma) - 30+ N8N workflow templates - Skills-bridge module: Python, Docker, C#, REST, XML - in OT-context - Capstone: één Solve-event van begin tot eind op JOUW plant

Founding cohort prijs: €49/mnd lifetime lock-in (eerste 25 plekken). Daarna €97/mnd.

Vul de AI-readiness scorecard in →

16 vragen (5 min). Je krijgt direct een gepersonaliseerde gap-analyse over je skills (Python, Docker, MQTT, OPC-UA, vendor playbooks) en AI-affiniteit. Eerste stap richting het online programma.

Snelle links

Open-source stack De werkende Docker-stack (MonsterMQ + MongoDB + Grafana + Traefik) uit deze gids: github.com/cftservices/idp-os

Community Facebook groep "Industrial Data Platform" - engineers die dit samen bouwen: facebook.com/groups/1282445730514883

Waitlist Reserveer een plek in de founding cohort (€49/mnd lock-in, eerste 25): techflow24.com/nl/contact

Over TechFlow

TechFlow Industrial bouwt **the missing AI data layer** voor industriële fabrieken - een open-source data-architectuur (MonsterMQ + MongoDB + Grafana + Docker) die PLC, SCADA, MES en ERP verbindt en raw machine-data omzet in AI-ready informatie. Gedreven door 15 jaar shop-floor ervaring in fabrieken (Siemens PLC/SCADA, historians, DCS). **Build Your AI Data Layer** is het flagship-programma voor PLC/SCADA engineers die de stap willen maken naar Industrial Data Architect.

techflow24.com

"You're missing one layer - and AI can't fix what it can't understand."