

# ■ Cloud-Connected PLC Data Pipeline

## ■ Problem

- No real-time visibility of plant data
- Operators blind to remote production issues
- Delayed decision-making due to batch-based reporting

## ■ Solution

- Node-RED gateway for collecting PLC data
- Streaming via MQTT → Cloud
- Historical logging in SQL Server
- Interactive live dashboards for operators

## ■ Results

- 24/7 remote access to plant data
- Improved uptime and proactive troubleshooting
- Faster decision-making → reduced downtime

## ■ Tech Stack

- Node-RED
- MQTT
- SQL Server

# ■ High-Speed Kafka Streaming for Smart Factory

## ■ Problem

- High-frequency machine data overwhelmed traditional databases
- Data loss during heavy loads
- Lack of real-time anomaly detection

## ■ Solution

- Apache Kafka for distributed streaming
- Partitioned topics for scalability
- AI/ML models consuming Kafka topics for predictive maintenance

## ■ Results

- Handled millions of events per second
- Zero data loss, high reliability
- Early detection of equipment failures → reduced costs

## ■ Tech Stack

- Apache Kafka
- ML Models
- Python Consumers

## ■ Modbus-Enabled Direct Machine Monitoring

### ■ Problem

- Legacy machines lacked smart connectivity
- Operators relied on manual checks
- No central monitoring system

### ■ Solution

- Direct Modbus TCP/IP polling from PLCs
- Real-time dashboard showing machine OEE
- Historical logging in time-series storage

### ■ Results

- Reduced manual intervention
- Improved equipment utilization
- Lower downtime through proactive alerts

### ■ Tech Stack

- Modbus TCP/IP
- SCADA Dashboard
- Time-Series DB

# ■ Python-Based User Management for Smart Factory Apps

## ■ Problem

- Multiple apps had inconsistent logins
- No role-based access control
- Security risks from weak authentication

## ■ Solution

- Centralized Python authentication service
- Role-based access management
- Integration with industrial apps (Batch, Alarm, Reports)

## ■ Results

- Unified login system across apps
- Stronger security & compliance
- Improved user experience

## ■ Tech Stack

- Python Flask
- SQL DB
- JWT Authentication

# ■ Batch Management System with Full Traceability

## ■ Problem

- Paper-based batch recording
- No traceability of raw materials
- Manual approval delays

## ■ Solution

- Digital batch recording system
- Barcode/RFID-based material tracking
- Automated approvals & electronic signatures

## ■ Results

- 100% traceability
- Reduced compliance risk
- Faster batch release times

## ■ Tech Stack

- ■ Web App
- RFID/Barcode
- ■ SQL Server

# ■ Streamlit-Based Plant Performance Dashboard

## ■ Problem

- Operators lacked real-time visibility
- No single source of truth for KPIs
- Manual Excel reporting took hours

## ■ Solution

- Streamlit dashboard with SQL backend
- Real-time KPIs: OEE, downtime, alarms
- Exportable PDF reports for management

## ■ Results

- Instant visibility of KPIs
- Data-driven decision-making
- Reduced reporting time from hours → seconds

## ■ Tech Stack

- Streamlit
- Python
- SQL Server

# ■ Grafana for Industrial IoT Visualization

## ■ Problem

- Complex machine data hard to visualize
- No unified monitoring tool
- Reactive instead of proactive monitoring

## ■ Solution

- Grafana dashboards with IoT data
- Integration with MQTT + SQL
- AI anomaly detection integrated

## ■ Results

- Unified monitoring system
- Intuitive visualization for operators
- Early fault detection

## ■ Tech Stack

- Grafana
- MQTT
- Python AI Models