NOC Water & Gas Meters

Proof of Concept

Network Operations Center for Smart Metering

CrateDB | **EMQX MQTT** | **FastAPI** | **Vue.js + Quasar** | **Grafana** | **AI/ML** | **Leaflet GIS**

City of Zagreb -- Smart Utility Infrastructure

15,724 Households | 31,448 Meters | Real-Time IoT Analytics

Version 1.0 -- June 2026

1. Executive Summary

This Proof of Concept demonstrates a complete Network Operations Center (NOC) for real-time monitoring and analytics of water and gas metering infrastructure across the City of Zagreb. The system simulates 15,724 households (31,448 meters) generating continuous IoT telemetry, processed through a modern cloud-native stack centered on CrateDB as the real-time data engine.

15,724	31,448	2.5M+	254K
Households	Smart Meters	Readings	Leak Alerts

The PoC validates end-to-end feasibility of: real-time telemetry ingestion at scale, anomaly and leak detection via AI, predictive maintenance scheduling, GIS-based network visualization, and operational dashboards -- all powered by CrateDB's distributed SQL analytics engine.

2. PoC Scope & Objectives

2.1 Capabilities Demonstrated

Capability	Description	Status
Real-Time Ingestion	MQTT telemetry from 31K meters via EMQX	Validated
Time-Series Storage	CrateDB with auto-partitioning & compression	Validated
Real-Time Analytics	Sub-second SQL on millions of readings	Validated
AI Anomaly Detection	Isolation Forest per-sensor models	Validated
AI Leak Detection	Sequential pattern analysis for pipe leaks	Validated
Predictive Maintenance	2-year replacement cycle + health scoring	Validated
GIS Visualization	Leaflet map with clusters + heatmap overlay	Validated
Digital Twin	SVG schematic of water/gas network	Validated
Alarm Management	Threshold + AI-driven alerting	Validated
Multi-Language UI	Croatian, English, German (i18n)	Validated
Load Testing	Locust.io at 720x real-time speed	Validated

2.2 Business Objectives

- Non-Revenue Water Reduction -- AI leak detection identifies continuous low-flow anomalies
- Operational Efficiency -- Centralized NOC replaces fragmented manual processes
- Asset Lifecycle Management -- Predictive scheduling reduces emergency replacements 30-40%
- Regulatory Compliance -- Complete audit trail of readings, alarms, and maintenance
- Customer Transparency -- Real-time consumption data for consumer portals

3. System Architecture

3.1 Six-Layer Architecture

```
VISUALIZATION LAYER
  Vue.js + Quasar (SCADA)  |  Grafana (Analytics)  |  Leaflet GIS  |  Digital Twin (SVG)
```

```
APPLICATION LAYER
  FastAPI (REST + WebSocket)  |  AI Services (Python/NumPy)
  Alarm Engine  |  Leak Detector  |  Predictive Maintenance  |  Socket.IO
```

```
DATA & ANALYTICS LAYER
  CrateDB (Distributed SQL + Time-Series)
  4 tables | Daily partitioning | Bulk HTTP API | PostgreSQL wire protocol
```

```
MESSAGING LAYER
  EMQX 5.8 (MQTT Broker) | 31K+ connections | Topic: noc/{district}/{id}/reading
```

```
SIMULATION LAYER
  Locust.io | 35 zone users | paho-mqtt | 720x speedup | 15,724 households
```

3.2 Technology Stack

Layer	Technology	Role
Data Store	CrateDB 5.9	Distributed SQL + time-series, PG wire compatible
Message Broker	EMQX 5.8	MQTT broker, 31K+ concurrent connections
Backend API	FastAPI + Uvicorn	Async REST/WebSocket, buffered batch ingestion
Frontend	Vue 3 + Quasar 2	SCADA dashboard, Leaflet GIS, i18n
Analytics	Grafana 11.4	14-panel dashboard, CrateDB PG datasource
Simulation	Locust.io	Distributed load gen, realistic meter models
AI/ML	scikit-learn, NumPy	Isolation Forest, trend analysis, leak detection
Infrastructure	Docker Compose	Single-command deployment, 7 services

4. CrateDB -- Key Benefits

CrateDB was selected as the central data platform because it uniquely combines distributed SQL, native time-series support, and PostgreSQL compatibility in a single engine -- eliminating the need for separate databases for operational and analytical workloads.

1

Unified OLTP + OLAP Engine

Handles both high-throughput ingestion (2,500+ readings/sec via bulk HTTP API) and complex analytical queries (aggregations across 2.5M+ rows in sub-second latency) on the same cluster. No ETL pipeline or separate data warehouse needed.

2

Native Time-Series Partitioning

Automatic daily partitions via PARTITIONED BY (date_trunc). Enables efficient time-range queries (partition pruning) and simple data lifecycle management (drop old partitions for retention).

3

PostgreSQL Wire Protocol Compatibility

Grafana connects via standard PostgreSQL driver -- zero custom plugins. Existing tooling (psycopg2, DBeaver, pgAdmin) works out of the box, reducing integration effort and vendor lock-in.

4

High-Throughput Bulk Ingestion

Bulk HTTP API (/_sql with bulk_args) ingests 200+ readings per batch in a single round-trip. Sustained 2,500+ msg/sec demonstrated with buffered batch flushing.

5

Columnar Storage & Compression

Automatic columnar compression for time-series data. Significantly higher storage efficiency than row-oriented databases -- critical for 5-10 year meter data retention in utility applications.

6

Distributed SQL at Scale

Tables sharded across nodes (CLUSTERED INTO N SHARDS). Horizontal scaling by adding nodes without application changes. PoC runs single-node; production scales linearly.

7

Real-Time Views & Refresh Control

Configurable refresh_interval (3-10s) per table balances data freshness vs query performance. Views provide instant access to current operational state (e.g., active alarms).

8

Schema Flexibility (OBJECT/DYNAMIC)

Dynamic object columns allow AI services to store variable JSON structures without predefined schema. Accommodates evolving ML model outputs without DDL migrations.

4.1 CrateDB vs Alternatives

Criterion	CrateDB	InfluxDB	TimescaleDB	MongoDB
SQL Compatibility	Full SQL + PG wire	Flux/InfluxQL	PostgreSQL	NoSQL
Distributed Default	Native sharding	Enterprise only	Manual setup	Replica sets
Time-Series Native	PARTITIONED BY	Purpose-built	Hypertables	Generic
Grafana Integration	PG driver	Native plugin	PG driver	Plugin needed
Bulk Ingestion	HTTP bulk_args	Line protocol	PG COPY	Bulk write
Ad-hoc SQL Analytics	Full SQL	Limited	Full SQL	Agg pipeline

5. Data Model & Scale

5.1 CrateDB Tables

Table	Purpose	Partitioning
households	Consumer registry (15,724 records)	None (reference)
household_readings	Meter telemetry (2.5M+ records)	Daily
alarm_events	Alarm log (257K+ records)	None
maintenance_predictions	AI predictions (1.38M+ records)	None

5.2 MQTT Message Format

```
{
  "household_id": "HH-00001", "district": "Donji Grad",
  "timestamp": 1717858320000, "lat": 45.815, "lng": 15.982,
  "address": "Ilica 42",
  "water": { "id": "ZG-00001", "model": "Elster V200",
             "reading_m3": 1523.456, "delta_m3": 0.042 },
  "gas":   { "id": "ZG-17501", "model": "G4 BK",
             "reading_m3": 8234.789, "delta_m3": 0.315,
             "temperature": 21.3 }
}
```

6. AI & Predictive Analytics

6.1 Anomaly Detection (Isolation Forest)

- Algorithm: scikit-learn IsolationForest (unsupervised, no labeled data required)
- Per-sensor models: Individual detector for each of 31,448 meters
- Features: 7 statistical features from sliding window (mean, std, trend, IQR)
- Training: After 20 samples (~40s), auto-retrain every 60 samples
- Output: Anomaly score 0-1, threshold-based classification

6.2 Leak Detection

- Logic: 3+ consecutive readings with small delta (0.05-0.3 m3) = suspected leak
- Memory-efficient: deque(maxlen=5) per household, auto-pruning
- Result: 254,000+ leak alerts -- demonstrates Non-Revenue Water detection

6.3 Predictive Maintenance

- Cycle: 2-year mandatory replacement for water and gas meters
- Health scoring: Age factor + drift rate + volatility + baseline degradation
- Distribution: 60% OK / 25% Due Soon (90 days) / 15% Overdue
- Output: health_score (0-1), estimated_days_remaining, risk_factors

7. Visualization & Dashboards

7.1 SCADA Dashboard Pages

Page	Route	Key Features
KPI Dashboard	/	8 stat cards, alarm feed, district grid
Grid Map	/map	Leaflet clusters, heatmap overlay, filters
Digital Twin	/digital-twin	SVG network schematic, animated flow
Assets	/assets	Household table, search, dual meter view
Maintenance	/maintenance	Progress bars, days remaining, badges
Alarms	/alarms	Severity/type filters, real-time updates

7.2 Grafana Integration

14-panel auto-provisioned dashboard connected to CrateDB via PostgreSQL driver: stat panels (readings, consumption, temperature, alarms, overdue meters), time series (consumption by district), histogram (gas temperature), bar chart (top districts), state timeline (alarms), anomaly table, and pie chart.

7.3 Internationalization

Full UI localization in Croatian, English, and German via vue-i18n. Language switcher persists selection in localStorage.

8. PoC Results & Performance

8.1 System Performance

Metric	Result	Target	Status
Sustained ingestion	2,500+ msg/sec	1,000 msg/sec	PASS
API response (p95)	< 50ms	< 200ms	PASS
Query latency (2.5M rows)	< 100ms	< 500ms	PASS
Frontend page load	< 2s	< 3s	PASS
Socket.IO latency	< 100ms	< 500ms	PASS
AI anomaly detection	< 1ms/reading	< 10ms	PASS
Map render (15K markers)	< 3s	< 5s	PASS
Uptime during operation	100%	99%	PASS

8.2 Infrastructure Resources

Service	CPU	Memory	Disk
CrateDB	~15%	1 GB heap	~2 GB
EMQX	~10%	~300 MB	~100 MB
FastAPI Backend	~20%	~200 MB	--
Vue.js Frontend	~5%	~150 MB	--
Locust Simulator	~10%	~200 MB	--
Grafana	~5%	~200 MB	~50 MB

9. Production Readiness Roadmap

Phase	Items	Effort
Phase 1: Hardening	EMQX TLS+ACL, CrateDB multi-node, Retention, Backup	2-3 weeks
Phase 2: Integration	Real meter protocols (OMS/M-Bus), CIS/billing, SSO	4-6 weeks
Phase 3: Advanced AI	Multi-sensor correlation, RUL estimation, Forecasting	6-8 weeks
Phase 4: Scale	City-wide 200K+ meters, Kafka streaming, CDN, Monitoring	8-12 weeks

NOC Water & Gas Meters PoC -- Documentation generated June 2026
Built with CrateDB | EMQX | FastAPI | Vue.js | Grafana | Leaflet | scikit-learn
For the City of Zagreb Smart Utility Infrastructure Initiative