**Fully Cloud-Based IEC 104 Solution – Best Approach**

If you need **all IEC 104 data in the cloud**, the best approach is to **replace the Raspberry Pi with a cloud-hosted IEC 104 gateway** while ensuring **low-latency communication, high availability, and security**. Here’s the optimal setup:

**1. Best Cloud Architecture for IEC 104**

The most efficient setup involves a **dedicated cloud-hosted IEC 104 gateway** running on a **Virtual Machine (VM) or a containerized service (Docker/Kubernetes)**.

**A. Direct Cloud Gateway Approach**

* **IEC 104 data is sent directly to the cloud** from Energinet.
* Requires a **static public IP and secure VPN/tunnel** to comply with Energinet’s security requirements.
* A cloud VM running a **SCADA-compliant IEC 104 gateway software** is necessary.

**B. Cloud-Edge Hybrid Approach (Recommended)**

* Use an **on-premises relay node** (like a lightweight Raspberry Pi or industrial IoT gateway) to **capture IEC 104 data**.
* The relay node **forwards all IEC 104 messages to the cloud** using **MQTT, WebSockets, or secure VPN**.
* Ensures **local fallback in case of cloud failure**.

**2. How to Implement a Fully Cloud-Based IEC 104 Gateway**

**Step 1: Choose a Cloud Provider**

* **AWS (EC2 + IoT Core)**
* **Azure (Virtual Machines + IoT Hub)**
* **Google Cloud (Compute Engine + Pub/Sub)**

A VM-based setup provides **full control over IEC 104 traffic**.

**Step 2: Deploy an IEC 104 Gateway in the Cloud**

Use one of the following options:

1. **OpenMUC j60870 (Java-based)** – Deploy in a **Docker container on the cloud VM**.
2. **IEC 104 to MQTT Bridge** – Converts IEC 104 messages into MQTT format and sends them to the cloud.

**Step 3: Secure Network & Communication**

Since IEC 104 is **not natively designed for the cloud**, a secure **VPN or private connection** is required:

* **Option A: VPN or Direct Connect** (Preferred for Security)
* Use **AWS Direct Connect / Azure ExpressRoute / Google Cloud Interconnect** for a **dedicated link**.
* Or set up a **secure OpenVPN or WireGuard tunnel**.
* **Option B: TLS-Encrypted IEC 104 Over Public Internet**
* Use **IEC 104 over TLS** (if supported).
* Use **port forwarding on cloud firewall** (**allow TCP 2404** for IEC 104).

**Step 4: Cloud Data Processing & Storage**

* **Real-Time Streaming**: Use **Apache Kafka, AWS Kinesis, or Google Pub/Sub** to process IEC 104 data.
* **Storage Options**:
* **Time-Series Database**: InfluxDB, TimescaleDB (for SCADA-like data storage).
* **Cloud Storage**: AWS S3, Azure Blob, or Google Cloud Storage.
* **SQL for Structured Data**: PostgreSQL, MySQL.
* **Analytics & Visualization**:
* Use **Grafana, Power BI, or AWS QuickSight** for monitoring.

**3. Pros & Cons of a Full Cloud-Based IEC 104 Solution**

| **Aspect** | **Cloud-Based IEC 104** | **On-Prem Raspberry Pi** |
| --- | --- | --- |
| **Latency** | 20-100ms (higher due to WAN) | 3-15ms (local network) |
| **Security** | Requires VPN/Firewall | Local, easier to control |
| **Reliability** | Dependent on internet/cloud uptime | Works even offline |
| **Scalability** | Easily scalable in the cloud | Limited by Raspberry Pi’s resources |
| **Cost** | Cloud VM costs + data transfer fees | One-time hardware cost |

**4. Conclusion**

1. Deploy a cloud-hosted IEC 104 gateway (Docker/VM)
2. Secure traffic via VPN (AWS Direct Connect, OpenVPN, or WireGuard)
3. Use MQTT/Kafka for real-time data ingestion
4. Store data in a time-series or SQL database
5. Monitor using Grafana, Power BI, or cloud dashboards