

Technology Stack Research

Compare Database Options (SQLite, PostgreSQL, Timescale DB):

A. SQLite (Local Database for Agent & CMS Prototype)

Perfect for a lightweight, embedded database.

How SQLite Works

SQLite is a serverless relational database embedded inside your application. All data is stored in a single .db file.

Key features:

- Uses SQL syntax
- Fast read/write
- Thread-safe with correct flags

Why SQLite suits your EDR

- **WAL mode** → fast writes for telemetry
- Very stable & efficient
- Zero configuration
- Highly portable
- Small footprint

Use cases

- Agent-side logging
- Telemetry batching
- Policy storage
- Update metadata

Does SQLite Work for Your EDR? → YES (Perfect for local agent storage)

Why It Works

- ✓ Your agent stores local telemetry → SQLite is ideal
- ✓ Tiny footprint (<1MB library)

- ✓ No server required(runs inside agent) → perfect for isolated/offline systems
- ✓ SQLite WAL mode supports high-frequency logging
- ✓ Easy to backup, copy, or rotate

Limitations

- ✗ Not good for high-volume multi-device telemetry on server
- ✗ No clustering or replication
- ✗ Can lock under heavy concurrent writes

Suitability Score: 10/10 for Agent-side Storage

6/10 for CMS(Content Management System) storage (only good for prototypes).

→ **Ideal for the BESS prototype.**

B. PostgreSQL (Full Enterprise Database Server)

A powerful relational database used in production systems.

How PostgreSQL Works

PostgreSQL is a full RDBMS system supporting:

- Multi-user access
- MVCC (Multi-Version Concurrency Control)
- Indexing, triggers, complex queries
- JSONB storage (semi-structured data)
- Extensions

PostgreSQL uses **client-server architecture**.

Why PostgreSQL matters

- Handles large telemetry volumes
- Supports indexing, partitioning
- Strong ACID compliance

Use cases in production EDR

- Central telemetry warehouse
- Policy management

- Device inventory

Does PostgreSQL Work for Your EDR? → YES (For production CMS)

Why It Works

- ✓ CMS receives telemetry from multiple devices → PostgreSQL handles scale
- ✓ Supports JSONB → ideal for storing variable telemetry fields
- ✓ Reliable, secure, open-source
- ✓ Can scale with partitioning
- ✓ Supports time-based querying

Limitations

- ✗ Requires server management
- ✗ Needs more RAM/CPU than SQLite
- ✗ Harder to deploy

Suitability Score: 9/10 for CMS, 3/10 for agent

→ Suitable for scaling BESS beyond academic prototype.

C. TimescaleDB (Time-series Database on PostgreSQL)

How TimescaleDB Works

TimescaleDB is an extension on top of PostgreSQL engineered for time-series data.

It converts tables into hypertables, which automatically partition data by:

- Time
- Device ID
- Telemetry type

Why TimescaleDB is powerful for EDR

- Telemetry is naturally **time-series data** (process events, file-change events, alerts).
- Supports automatic partitioning (hypertables)
- Fast insertion rate
- Efficient querying of historical data

Use cases

- Long-term telemetry storage
- Behavioural analytics
- Historical process patterns

Does TimescaleDB Work for Your EDR? → YES (If you scale to enterprise)

Why It Works

- ✓ Massive write throughput (millions of rows/sec)
- ✓ Built-in retention & compression
- ✓ Fast historical analysis

Limitations

- ✗ Too heavy for academic prototype
- ✗ Requires a full server + resources
- ✗ More complexity in deployment

Suitability Score:

Prototype: **5/10**

Production at scale: **10/10**

→ **Advanced option if scaling to enterprise EDR capabilities.**