LLMが"脳"になったオフィス — そして都市へ

共生型環境管理システム SOMS

Symbiotic Office Management System

LLM × IoT × 人間経済で実現する自律的空間知能

問題提起: 従来のBMSの限界

従来のBMS (IF-THEN)	SOMS (LLM推論)
CO2 > 1000ppm → 換気ON	CO2 1000ppm + 3人作業中 → タスク生成 + 通知
温度異常 → アラーム	30秒で5度変化 → 改竄の可能性を判断
人感ON → 照明ON	30分同姿勢 → 健康助言を音声で提供
(対応不可)	ホワイトボード汚れ → 報酬付き清掃タスク掲示

決定論的ルールでは **文脈を理解した判断** ができない。
さらに、APIで操作できない **物理タスク** は解決できない。

核心的な問い

> スマートホームAPIで操作できない物理タスクを、
> AIはどう解決するか？

窓を閉める。ホワイトボードを拭く。備品を補充する。
ロボットアームなしに、AIはどうやって物理世界を動かすか？

アンサー: 人間を「高度な汎用アクチュエータ」として、
経済的インセンティブで動かす — これが **共生 (Symbiosis)** の意味。

有機体メタファー

システム全体を一つの **有機体** として設計する。

生物学的機能	SOMS コンポーネント	技術
脳	中央知能 (ReAct推論)	LLM (Qwen2.5 14B)
神経系	メッセージバス	MQTT (Mosquitto)
感覚器	環境センシング	BME680, MH-Z19C, YOLOv11
手足	エッジデバイス	ESP32 (MicroPython / C++)
声	音声合成	日本語TTS
外部協力者	人間	ダッシュボード + タスク経済

LLMが状況を判断 → タスクを生成 → 報酬を提示 → 人間が自由意志で遂行

4層アーキテクチャ

Node-RED / LangChain **不使用** ◦Python + MQTT による純粋なイベント駆動。

ReAct 認知ループ

Think → **Act** → **Observe** を最大5反復。30秒サイクル or イベント駆動。

```
[トリガー] ← MQTT イベント（3秒バッチ） or 30秒定期
    ▼
[THINK]    WorldModel → LLM に状態 + 行動原則を送信
    ▼
[ACT]      LLM が判断 → ツール呼び出し → 安全検証 → 実行
    ▼
[OBSERVE]  結果をフィードバック → 追加行動を判断（最大5反復）
    ▼
[完了]     追加行動不要 → ループ終了
```

- サイクル間隔: 30秒 / LLMタイムアウト: 120秒 / デバイス応答: 10秒
- 正常時は **何もしない** ことを許容する設計

MCP over MQTT

MCP (Model Context Protocol) を MQTT 上に実装。

特性	HTTP	MQTT
通信モデル	リクエスト/レスポンス	パブリッシュ/サブスクライブ
LLM推論とデバイスの時間差	タイムアウトリスク	ブローカーが吸収
マイコン実装	重い	軽量 (最小ヘッダ)
耐障害性	再実装が必要	QoS + LWT 組み込み

```
Brain → mcp/{device_id}/request/call_tool   (JSON-RPC 2.0)
Edge  → mcp/{device_id}/response/{req_id}   (JSON-RPC 2.0)
```

LLMの推論 (秒単位) とデバイスの応答 (ミリ秒〜分) の非対称性を吸収する。

WorldModel: センサーフュージョン

複数センサーを **指数減衰加重平均** で統合。新しい値ほど重みが大きい。

センサー	半減期	設計意図
温度	120秒	緩やかな変化を安定的に追従
CO2	60秒	在室状況の変化に敏感に反応
在室人数	30秒	リアルタイム性を最優先

イベント検知 — 状態変化を検出し、クールダウン付きで発火:

イベント	条件	クールダウン
CO2閾値超過	> 1000ppm	10分
温度急変	3度以上/短時間	—
長時間座位	同姿勢30分以上	1時間

憲法的AI: 言語による行動制約

LLMの行動を **コードのハードコードではなく、言語による原則** で制約する。

1. **自律性 > 自動化** — 目的関数に向けて自ら判断
2. **安全最優先** — 健康・安全に関わる問題は最高優先度
3. **コスト意識** — 報酬は難易度に比例、不必要な依頼をしない
4. **重複回避** — タスク作成前に既存タスクを必ず確認
5. **正常時は介入しない** — 全指標が正常なら何もしない
6. **ローカルファースト** — 全処理オンプレミス、クラウド送信ゼロ

拠点ごとに異なるプロンプト（憲法）を持たせることで、
同じアーキテクチャを異なるドメインに特化できる。

シナリオ: 嵐のプロトコル

```
[T+0s]   気圧急低下 + 天気API「15分後に豪雨」
[T+3s]   Brain: 被害リスクを判断 → 窓の状態を確認
[T+4s]   窓3: スマートアクチュエータ → 自動閉鎖
         窓5: 手動式 → 緊急タスク生成 (報酬: 最大, 緊急度: 最高)
[T+5s]   音声通知 + ダッシュボードに緊急タスク表示
[T+30s]  社員が受諾 → 窓を閉める → 完了報告
```

- **自動化と人間協働のハイブリッド** — 自動化できるものは自動化し、できないものは人間に委託
- 報酬は緊急度に応じて **動的に決定** (通常の5倍に引き上げ)
- LLMが「判断ログ」を記録 → 将来の推論に活用

シナリオ: 都市の呼吸パターン発見

```
Phase 0 (単一オフィス):
  CO2ピーク 9:00, 13:00 — 人数変動と相関

Phase 2 (複数拠点):
  Hub-A (オフィス街): CO2ピーク 9:00, 13:00
  Hub-B (商業施設):   CO2ピーク 11:00, 15:00, 19:00
  Hub-C (住宅街):     CO2ピーク 7:00, 20:00

Phase 3 (都市規模):
  → 人の流れが可視化される:
    住宅街(朝) → オフィス街(日中) → 商業施設(夕方) → 住宅街(夜)
  → 大気質の悪化パターンを検出
  → 都市計画（換気設備・緑地配置）への入力データに
```

単一拠点の環境監視が、都市規模の **知能** になる。

データ主権: 50,000:1 圧縮

層	データ量 (1拠点/日)	外部送信
生信号 (映像 + センサー)	~50 GB	不可
構造化イベント	~500 MB	Hub内保存
City Hub への送信	**~1 MB**	集約統計のみ

50 GB → 1 MB = 50,000:1

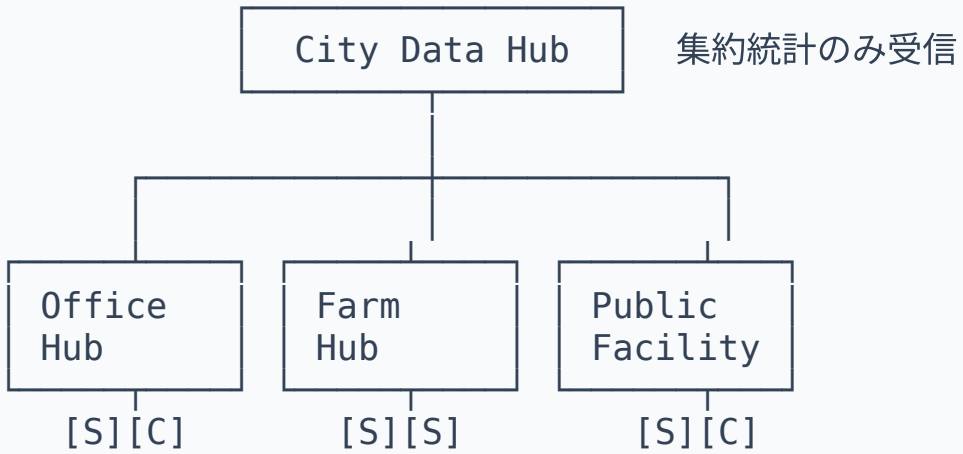
- 映像は RAM 上でのみ処理し、保存しない
- 外部に送るのは1時間集約の統計値のみ
- Core Hub を物理的に撤去 → 全生データ消失 = **物理的データ主権**
- GDPR / 個人情報保護法への最強のコンプライアンス: データを送らない

三層データ処理モデル

```
Layer 0: 物理世界 (Raw Signal)
   │  センサー電圧値、カメラRGBピクセル
   │  → 量が膨大、意味不明、保存不要
   ▼
Layer 1: Core Hub (Local Intelligence)
   │  ローカルLLMがリアルタイム解釈
   │  → 構造化JSON、イベントログ、判断記録
   │  → この層でデータの 99% は破棄 (意味だけ抽出)
   ▼
Layer 2: City Data Hub (Aggregation)
   │  複数 Core Hub の構造化データを集約
   │  → 時空間分析、パターン抽出、都市規模の最適化
   ▼
Layer 3: 洞察と行動 (Insight & Action)
      都市計画への入力、政策提言、資源配分最適化
```

Core Hub アーキテクチャ

SOMSは **Core Hub** の最初の実装。



- 接続するセンサーとシステムプロンプトを変えるだけで **領域特化**
- 各 Hub は **独立して自律動作** — ネットワーク切断時も72時間継続
- Hub 間で共有するのは **集約統計のみ** — 生データは外に出ない

都市展開ロードマップ

Phase	内容	規模
0 (現在)	単一オフィスで E2E フロー実証	1 Hub, 2-3 ノード
1	多ゾーン化、Data Lake 実装	1 Hub, 10+ ノード
2	複数拠点 Core Hub + City Data Hub	2-3 Hub
3	都市内展開、ゼロタッチ配備	10+ Hub

Phase 0 達成済み:
ReAct認知ループ / センサーフュージョン / MCP over MQTT / AI画像認識 / ダッシュボード / 音声合成 / 仮想テスト環境

設計が正しかった部分 (Phase 3 まで変更不要):
MCP over MQTT / ReAct ループ / WorldModel / 憲法的AI / Per-channel テレメトリ / タスク経済

パフォーマンス実測値

AMD RX 9700 (32GB VRAM) + Qwen2.5 14B — GPU サーバー1台で完結。

指標	値
LLM推論速度	~51 tok/s (安定)
正常データ応答	3.3秒
ツール呼び出し応答	6.6秒
エラー率	0% (12リクエスト)
月額クラウド費用	$0

30秒の認知サイクルに対して3-7秒で応答 → 十分に実用的。
全処理ローカル完結。外部依存ゼロ。

技術スタック

層	技術
LLM	Qwen2.5 14B (Ollama, AMD ROCm)
Vision	YOLOv11 (物体検出 + 姿勢推定)
Backend	Python 3.11, FastAPI, SQLAlchemy async
Frontend	React 19, TypeScript, Vite, Tailwind CSS
Messaging	MQTT (Mosquitto), MCP (JSON-RPC 2.0)
Edge	ESP32 (MicroPython / PlatformIO C++)
Database	SQLite (aiosqlite)
Container	Docker Compose
GPU	AMD RX 9700 (RDNA4, 32GB VRAM)

ミドルウェア不使用。Python + MQTT の純粋なイベント駆動設計。

まとめ

SOMS は、LLMを「脳」として物理空間をデータ化する
自律型環境管理システムの実証プラットフォーム。

- **文脈理解** — IF-THENを超えた ReAct 認知ループ
- **物理タスクの解決** — 人間との経済的協働 (共生)
- **データ主権** — 50,000:1 圧縮、クラウド送信ゼロ
- **都市への拡張** — Core Hub が都市の知能インフラになる

> 1つのオフィスから始まる、都市をデータ化するためのアーキテクチャ。

GitHub: `Office_as_AI_ToyBox`