都市のAI化 — 分散型ローカルLLMアーキテクチャの技術的実証

SOMS: Symbiotic Office Management System

Core Hub Phase 0 — 1つのオフィスから都市全体へ

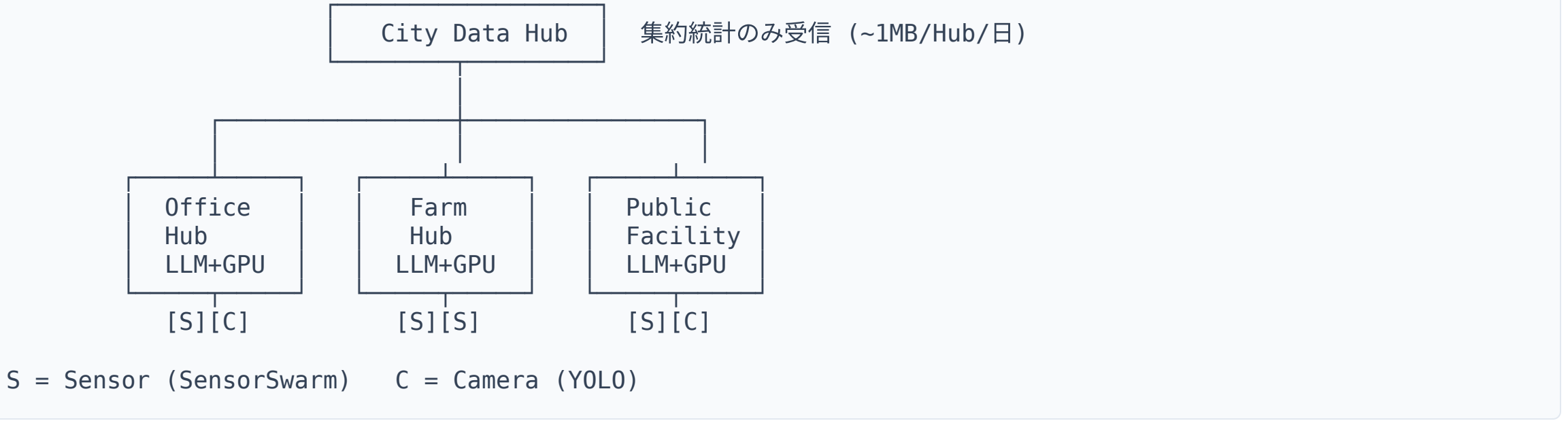
問題提起: 都市の情報化の構造的欠陥

現在のスマートシティ	Core Hub アーキテクチャ
データ → クラウドに吸い上げ	データ → **ローカルLLMが即座に処理**
API料金を払って自分のデータにアクセス	**自身の計算資源で完結**
ネット切断 → 全機能停止	**孤立状態でも自律動作継続**
カメラ映像が外部サーバーに保存	**RAM上で処理・即時破棄**
数百ms〜秒のクラウド遅延	**3-7秒でローカルLLM応答**

核心的問い: データが生成される場所（建物）で処理されないのは、なぜか？

→ 答え: ローカルでLLM推論できるGPUが安価になったのは最近だから。**今なら可能。**

Core Hub: 建物レベルの自律AI



各 Core Hub は **同一アーキテクチャ** — プロンプト（憲法）とセンサーの変更だけで領域特化。

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
```

50 GB → 1 MB = 50,000:1 — これがデータ主権の技術的保証。

SOMS: Core Hub Phase 0 の有機体メタファー

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	実装状況
脳	中央知能 (ReAct推論)	LLM (Qwen2.5 14B)	5ツール, 3層安全機構
神経系	メッセージバス	MQTT + MCP (JSON-RPC 2.0)	ESP32実通信済み
感覚器	環境センシング	BME680, MH-Z19C, YOLOv11	3モニター, 4層姿勢分析
手足	エッジデバイス	SensorSwarm (Hub+Leaf)	4種トランスポート
声	音声合成	VOICEVOX + LLMテキスト生成	拒否ストック100件事前生成
経済	タスク経済	複式簿記 + デマレッジ	PWAウォレットアプリ
外部協力者	人間	ダッシュボード (キオスク)	2段階重複検知

ReAct 認知ループ + 多層安全機構

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
```

3層安全機構 (Phase 0 で実装済み):

- **Layer 3**: 重複検知フィルタ + speak レート制限 (1回/サイクル)
- **Layer 4**: サイクルレート制限 (最小25秒間隔)
- **Layer 5**: 行動履歴追跡 (30分窓、反復行動を抑止)

MCP over MQTT: AI-デバイス通信プロトコル

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

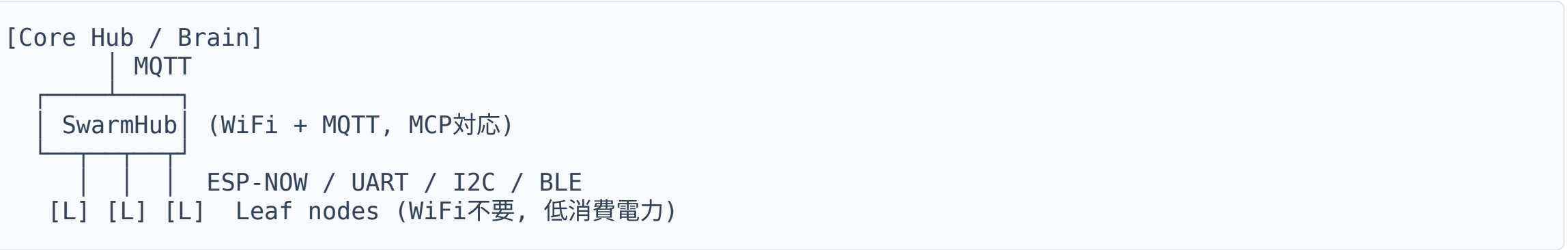
特性	HTTP	MQTT
通信モデル	リクエスト/レスポンス	パブリッシュ/サブスクライブ
LLM推論とデバイスの時間差	タイムアウトリスク	ブローカーが吸収
マイコン実装	重い	軽量 (最小ヘッダ)
耐障害性	再実装が必要	QoS + LWT 組み込み

```
Brain → mcp/{device_id}/request/call_tool   (JSON-RPC 2.0)
Edge  → mcp/{device_id}/response/{req_id}   (JSON-RPC 2.0)
Telemetry → office/{zone}/sensor/{device_id}/{channel} → {"value": X}
```

Phase 3 (都市展開) まで **プロトコル変更不要** — 設計の正しさが実証された。

SensorSwarm: スケーラブルなエッジ層

都市AI化の精度は末端センシングの密度に比例する。



特性	Hub	Leaf
WiFi	あり (MQTT接続)	**不要** (Hub経由)
プロトコル	MCP (JSON-RPC 2.0)	バイナリ (5-245B, XOR checksum)
デバイスID	`swarm_hub_01`	`swarm_hub_01.leaf_env_01` (ドット表記)
トランスポート	WiFi	ESP-NOW / UART / I2C / BLE (4種)
消費電力	通常	**極低** (WiFiスタック不要)

Hub経由でBrainからは各Leafが独立センサーとして見える → **建物全体を安価に高密度カバー**

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

Perception: YOLOv11 + 4層活動分析

Tier 1: 高速検出 (5秒間隔, QVGA)
→ YOLO11s.pt で人物検出 (在室カウント)

Tier 2: 姿勢推定 (人物検出時のみ)
→ YOLO11s-pose.pt でスケルトン抽出

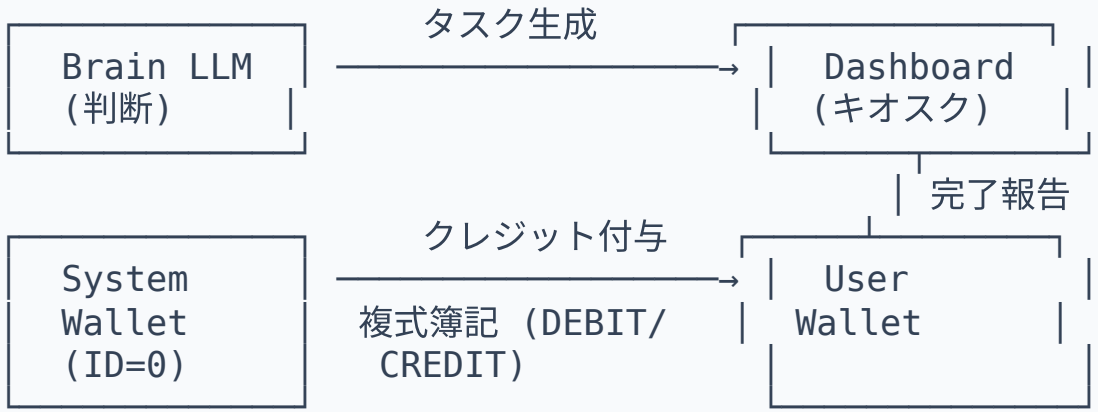
4層時間バッファ (活動パターンの長期追跡):

層	時間解像度	保持期間	用途
Tier 0	30-50ms	数秒	瞬間姿勢
Tier 1	100-500ms	数分	動作パターン
Tier 2	1-4s	数十分	中期傾向
Tier 3	~1h	4時間	長時間追跡

カメラ自動検出: ネットワーク ping sweep + YOLO 検証 → プラグアンドプレイ

共生経済: 複式簿記 + デマレッジ

物理タスクをAIが生成し、人間が遂行し、経済が回る。



機能	実装
台帳	複式簿記 (PostgreSQL, トランザクションID付き冪等性保証)
タスク報酬	500〜5000ポイント (LLMが動的決定)
デバイスXP	運用貢献度 → 動的乗数 (1.0x〜3.0x)
デフレ機構	手数料5%焼却 + デマレッジ2%/日
ウォレットPWA	残高確認 / QRスキャン / P2P送金 / 履歴

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 → MCP経由で自動閉鎖
         窓5: 手動式 → 緊急タスク生成 (報酬: 5000, 緊急度: 最高)
[T+5s]   VOICEVOX音声通知 (alertトーン) + ダッシュボード表示
[T+30s]  人間が受諾 → 窓を閉める → 完了報告 → クレジット付与
```

- **自動化と人間協働のハイブリッド**
- 報酬は緊急度に応じてLLMが **動的に決定** (通常の5倍)
- 2段階重複検知でタスクの二重発行を防止

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
```

各Hubの **1時間平均CO2値のみ** で都市知能が創発する。

データ主権: 物理的保証

層	データ量 (1 Core Hub/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

- 映像は RAM 上でのみ処理し、ディスクに保存しない
- 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: データを送らないことが最強のコンプライアンス
- Hub間通信: mTLS (相互認証) + TLS 1.3

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
1	多ゾーン化、Data Lake + Data Mart 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 (変更不要な設計):
MCP over MQTT / ReAct ループ + 3層安全機構 / WorldModel + センサーフュージョン / SensorSwarm (Hub-Leaf) / 憲法的AI / Per-channel テレメトリ / タスク経済 + 複式簿記

Phase 1 で追加: Event Store (TimescaleDB) / Data Mart 集約パイプライン / OTA更新

パフォーマンス実測値

AMD RX 9700 (RDNA4) + 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)
Dockerサービス数	11
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。

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, 4層バッファ)
Backend	Python 3.11, FastAPI, SQLAlchemy async
Frontend	React 19, TypeScript, Vite 7, Tailwind CSS 4
Messaging	MQTT (Mosquitto), MCP (JSON-RPC 2.0)
Edge	SensorSwarm (ESP32 Hub + Leaf, 4種トランスポート)
Economy	複式簿記 (PostgreSQL), デマレッジ, PWAウォレット
Database	PostgreSQL 16 (asyncpg) / SQLite fallback
Voice	VOICEVOX (日本語TTS, 拒否ストック事前生成)
Container	Docker Compose (11 services)
GPU	AMD RX 9700 (RDNA4)

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、分散型ローカルLLMで物理空間をAI化する
Core Hub アーキテクチャの Phase 0 実証プラットフォーム。

- **都市の情報化** — データが生まれた場所で処理する、本当のスマートシティ
- **自律的AI** — IF-THENを超えた ReAct 認知ループ + 多層安全機構
- **物理タスクの解決** — 人間との経済的協働 (共生) + 複式簿記経済
- **データ主権** — 50,000:1 圧縮、クラウド送信ゼロ、物理的保証
- **スケーラブルエッジ** — SensorSwarm で安価に高密度カバー
- **都市への拡張** — Core Hub が都市の知能インフラになる

> クラウドにデータを送ることは、都市の情報化ではない。
> 都市が自ら考えることが、本当の情報化だ。

GitHub: `Office_as_AI_ToyBox`