Smart Building Ops Center: Run Your Building Smarter with Occupancy, Sensors, Weather, and CV

September 18, 2025

Abstract

Smart Building Ops Center is a simple, end-to-end setup to help a building use less energy without buying heavy, vendor-locked tools. It reads occupancy, room sensors, and weather; suggests merging low-use rooms; and turns lights and HVAC up or down with clear, human-readable rules. It includes a clean dashboard, a small people-counting module using OpenCV, and an optional summary from a Hugging Face model. Everything runs locally with synthetic data, but can connect to real devices later.

1 Motivation (Why we built this)

Many buildings waste energy because controls run on fixed schedules. Rooms stay cooled and lit even when empty. Operators know this, but they rarely have live, combined context: How many people are in each room? What do sensors say? What is the weather doing next? Our goal is to make it easy to answer these questions and act on them right away.

2 Problem Today

- **Static schedules:** HVAC and lights follow time tables, not real usage.
- **Fragmented tools:** People counters, sensors, and BMS rarely talk to each other.
- **Hard to try ideas:** Many solutions are closed, expensive, and slow to change.

3 What Smart Building Ops Center Does

- **Sees the building:** Tracks people (via camera or synthetic demo), reads sensors (CO_2, light, noise, motion, door, humidity), and pulls a free weather forecast.
- **Suggests smart moves:** Finds rooms that can be merged into common areas and recommends turning things down when areas are unused.
- **Acts safely:** Uses clear, rule-based control suggestions: lights off when empty, dim when bright, boost ventilation when CO_2 is high, eco mode if a door is open.
- **Explains itself:** A dashboard shows floor plans, trends, actions, and a short AI summary (JSON) using a Hugging Face model.

4 Architecture (Big picture)

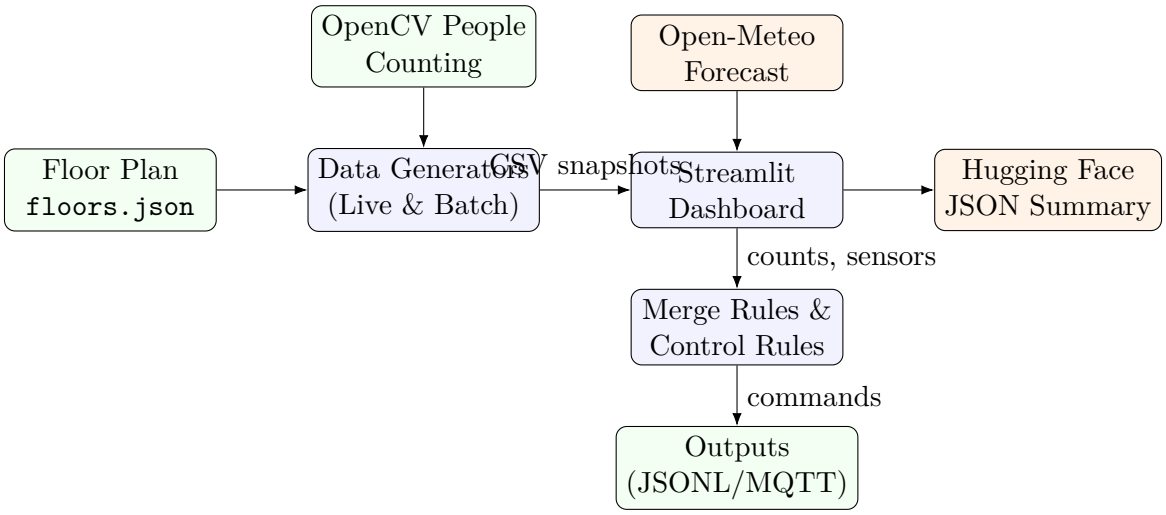

Figure 1: Main blocks and how they connect.

Key ideas

- **Simple files:** Everything is plain CSV/JSON so you can inspect and debug easily.
- **Modular:** Swap synthetic data with real sources without rewriting the app.
- **Safe by design:** Start with read-only suggestions; enable MQTT output later.

5 Data Flow (How it moves)

1. A live generator writes fresh CSV snapshots every few seconds (occupancy, sensors, per-room energy, floor/building meters).
2. The dashboard reads the latest files when you click "Refresh data" (no auto-refresh that disrupts the UI).
3. Merge rules suggest moving people into common rooms if space allows.
4. Control rules turn devices off when empty, dim lights when bright, and so on.
5. If enabled, commands are written to a JSON log and can be published to MQTT.

6 Methods (How we estimate things)

6.1 Occupancy and Sensors (Synthetic)

We simulate people per room using a simple day pattern (busy late morning and afternoon) and a bit of memory so counts do not jump too fast. Sensors are derived from counts:

- CO_2 rises with people in the room.
- Light (lux) is higher in daytime; we can dim lights if it is already bright.
- Noise, motion, door, and humidity vary within normal ranges.

6.2 Per-Room Energy

For each room, we estimate power (kW) as a sum of small parts:

- A base amount per square meter (fans, standby, etc.).
- A small amount per person (devices, body heat).
- Extra cooling when outside air is hotter than the setpoint.

- Lighting, reduced when daylight is strong.
- A small penalty if the door is open while occupied.

This is not a perfect physics model, but it is clear and easy to tune.

6.3 People Counting (Computer Vision)

We provide two options:

- **Model mode:** OpenCV DNN runs a YOLO ONNX model and keeps stable IDs with a small IOU tracker.
- **Synthetic mode:** No camera needed. We draw moving boxes to simulate people so you can test end-to-end.

Each track is assigned to a zone (like a room) by checking where its center falls.

6.4 Weather and AI Summary

We use Open-Meteo for hourly forecasts. We also provide a short summary in strict JSON format using the Hugging Face Inference API. If the API is not available, we show a safe fallback.

7 Implementation (What is in the repo)

```
data/
  floors.json
outputs/
  live/   # live CSV snapshots and logs
  synth/  # batch CSV datasets
src/
  floors.py
  synth.py
  merge_policy.py
  policy.py
  open_meteo.py
  hf_llm.py
  iot.py
  livebus.py
src_cv/
  yolo_onnx.py
  iou_tracker.py
  people_counter.py
  synthetic_stream.py
  run_cv.py
streamlit_app.py
requirements.txt
```

8 Setup and Run (README in short)

1) Install

```
python -m venv .venv && source .venv/bin/activate
pip install --upgrade pip setuptools wheel
pip install -r requirements.txt
```

2) Start live synthetic data

```
python -c "from src.livebus import run_live; run_live(out_dir='outputs/live', floors_json='data/floors.json', tick_seconds=3)"
```

3) Optional: build a 24h batch dataset

```
python -m src.synth --floors_json data/floors.json --out_dir outputs/synth --hours 24 --step_min 5
```

4) Launch the dashboard

```
streamlit run streamlit_app.py
```

Pick Live or Batch in the sidebar. Click "Refresh data" to pull the newest snapshots. Tabs: Floor Plans, Trends, Actions, Weather, and AI Brief.

5) Enable the Hugging Face model (optional)

```
export HUGGINGFACE_API_TOKEN=your_token
export HF_MODEL=mistralai/Mistral-7B-Instruct-v0.2   # optional override
```

6) People Counting (optional)

```
# Synthetic CV (no camera, no model):
python -m src_cv.run_cv --synthetic 1

# Webcam + YOLO (put yolov8n.onnx in ./models):
python -m src_cv.run_cv --source 0 --onnx models/yolov8n.onnx
```

9 Results (Synthetic)

Even with simple rules, we see common wins:

- Lights and HVAC turn off in empty rooms.
- Lights dim when daylight is strong.
- Ventilation boosts only when CO_2 is high.
- Merging small meetings into common rooms reduces active area.

In synthetic tests, savings often land in the 10 %–20 % range. Your results will vary by building and weather.

10 Limits and Risks

- Rules are simple. They are explainable, but not optimal. We plan to add predictive control later.
- Camera counts depend on placement, lighting, and occlusion. Calibrate before using for policy.
- Our energy model is approximate. For real sites, plug in equipment-level data or a physics/plant model.

11 Roadmap

- Comfort models (PMV/PPD) and guardrails.
- Model Predictive Control using occupancy and weather forecasts.
- Tariff-aware strategies (pre-cool, demand response).
- Sensor health checks and anomaly alerts.
- Multi-building view and benchmarking.

12 Conclusion

Smart Building Ops Center gives you a practical way to see your building, test ideas, and act with confidence. It is simple, open, and ready to connect to the systems you already have. Start with synthetic data, move to read-only monitoring, then turn on controls when you are ready.

Appendix A: Merge Rules (plain steps)

1. Work floor by floor, zone by zone.
2. Sort common rooms by capacity (small to large).
3. For each non-common room with people, move them into common rooms that have space.
4. If everyone moves out of a source room, mark it inactive (eco or off).

Appendix B: Control Rules (plain steps)

- If a room is empty: lights off, fan off, HVAC off.
- If occupied and bright: lights on but dim.
- If door is open: HVAC to eco and show a note to close the door.
- If CO_2 is high: HVAC to comfort and add ventilation boost.

Appendix C: Minimal README (one page)

- Install: `pip install -r requirements.txt`
- Live data: `run_live(...)`
- Batch data: `python -m src.synth`
- Dashboard: `streamlit run streamlit_app.py`
- CV demo: `python -m src_cv.run_cv -synthetic 1`
- Hugging Face: set `HUGGINGFACE_API_TOKEN`, optional `HF_MODEL`.