EM Tools Input Normalization Strategy

🎯Purpose

This document outlines the approach used by EM Tools to support multiple input sources (e.g., CBECC, IESVE, Pollination) through a unified intermediate file format called `normalized_model.json`. This enables consistent QA, reporting, and transformation workflows regardless of the originating simulation platform.

🎙️ Supported Input Sources

Tool / Platform	File Types Used	Normalization Strategy
CBECC (Res/Com)	`ab.xml`, `ap.xml`, CSV	Parsed into zone, envelope, and load summaries
IESVE	`.inp`, `.csv`	Geometry and energy report parsing
Pollination	`.hbjson`	Geometry + program tagging via Honeybee
Rhino + Honeybee	`.hbjson`	Same as Pollination
EnergyPlus (planned)	`.idf`	Reverse-map loads and geometry (TBD)
OpenStudio (planned)	`.osm`	SDK-driven data extraction (TBD)

Why Use Normalized JSON?

- 🔁Unifies data across formats
- 🈂️Enables multi-format output (IDF, ECON-1, LEED, CSV)
- 🛄Powers QA dashboards and test runners
- 🛅Decouples input parsing from output generation

This approach reduces complexity and maximizes flexibility.

🧩Structure Overview: `normalized_model.json`

```
{
  "project_name": "Example Project",
  "zones": [ ... ],
  "envelope": { ... },
  "lca": { ... },
  "ghg_tags": [...],
  "source_file": "input.ab.xml"
}
```

⛳Usage in EM Tools

- All modules from `v0.4` forward assume normalized inputs
- `v0.5` (Manual J) and `v0.6` (HBJSON) depend on this intermediary format
- `idf_generator.py` and `econ1_report_generator.py` are being updated to use it

Key Benefit

All future parsers (e.g. `hbjson_translator.py`, `cbecc_parser.py`) output the **same structure**, making them interchangeable upstream tools in the pipeline.

🎭Future Work

- Add support for `.idf` and `.osm`
- Improve schema validation with optional Pydantic or JSON Schema
- Create CLI tool to preview `normalized_model.json` content