Thermal Grace - VHUB
Comfort-as-a-Service IoT System

Software Architecture Design

Artur Kraskov
Semester 7
ICT & OL, Delta
Fontys 2025-2026

Contents

Contents 1
Introduction 2
- Background 2

Software Architecture 2
- Requirements summary 2
- Stakeholders and Needs 2
- C4 diagram 3
  - Level 1 - Context 3
  - Level 3 - Containers 3
  - Level 3 - Components 4
- System Inventory: Existing vs New/Custom-Built 4
  - Existing systems/services 4
  - New/custom-built systems in this project 4

Quality Attributes 5
- Security 5
  - Threats to address 5
  - Controls / decisions to document 5
- Scalability 5
  - Expected growth vectors 5
  - Scalability decisions 6
- Architecture Constraints and Assumptions 6
- Technology Choices and Rationale 6

Conclusion 7
Reference 7

Introduction

Within the scope of VHUB Comfort-as-a-Service system development a prototype was realized. The current document reports the software architecture design.

Background

Software design is made to satisfy requirements and fit in the context of user stories. Previously the project plan and the requirements document were compiled [1-2]. Hardware architecture was designed [3].

Software Architecture

Requirements summary

The system must be able to accumulate data from various wireless sensors. It must collect user feedback. Weather data must be fetched via an API. It must compute PMV/PPD values and query LLM for analysis and advice. New sensors or access points must be easily integrable and removable - the algorithm must still be capable of providing an accurate analysis, sensing and predicting perceived thermal comfort.

Stakeholders and Needs

Stakeholder group A — Researchers

- **Needs:** validity/traceability of measurements, reproducible computations, export/logging, interpretability, modularity, adaptability.
- **Concerns:** data quality, model transparency, auditability.

Stakeholder group B — Office users/visitors/workers

- **Needs:** easy feedback input, understandable comfort guidance, minimal friction.
- **Concerns:** privacy, data use, clarity, trust.

C4 diagram

Level 1 - Context

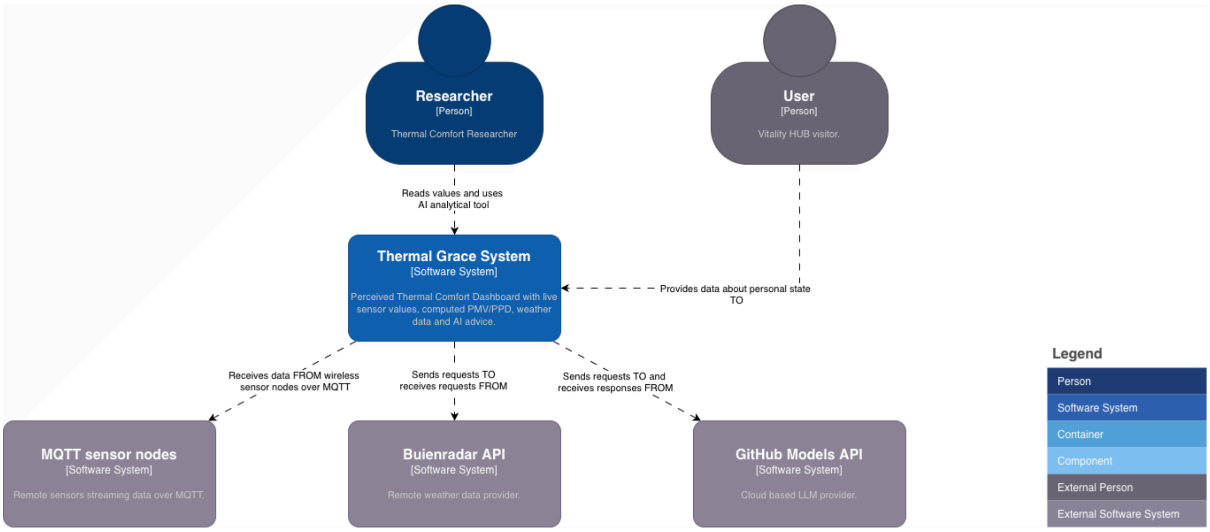

Image 1: C4 diagram level 1 (Context)
High resolution: https://drive.google.com/file/d/1KD9IUZkiTKQc9INoakWdf351ILHdJIYY

Level 3 - Containers

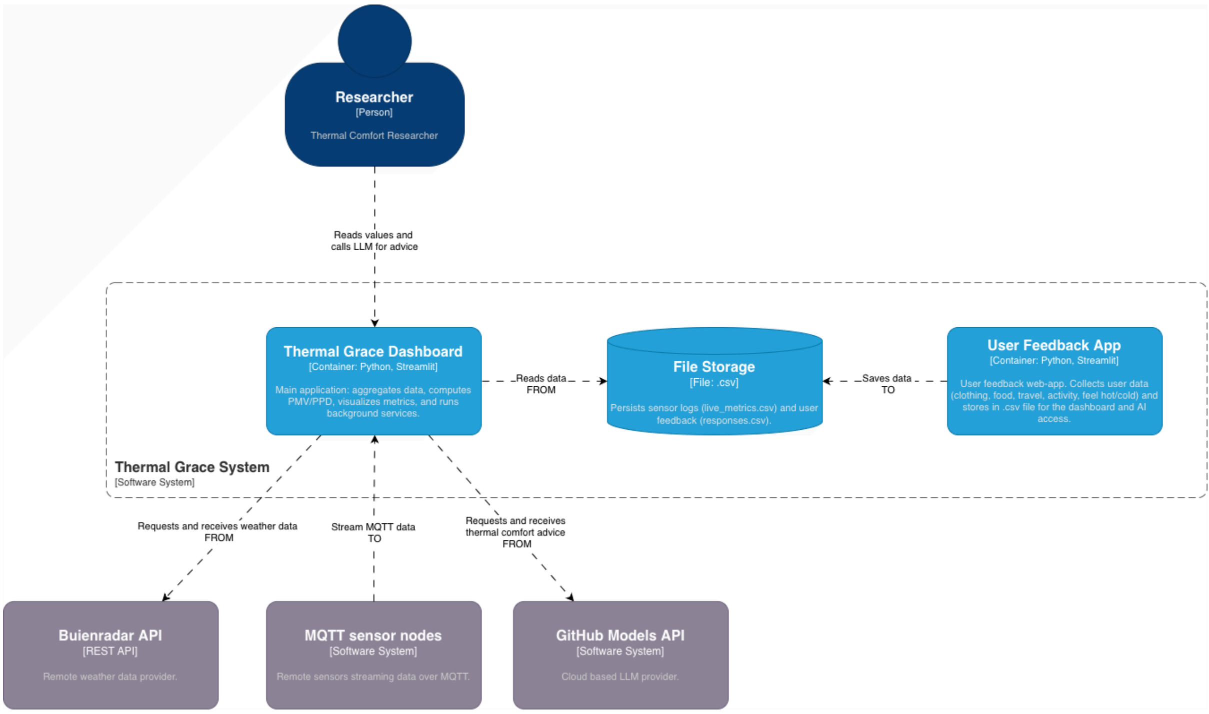

Image 2: C4 diagram level 2 (Containers)
High resolution: https://drive.google.com/file/d/1ZVjtGnIlWeJ-tD-7tcnDyWIDyyZe6cdQ

Level 3 - Components

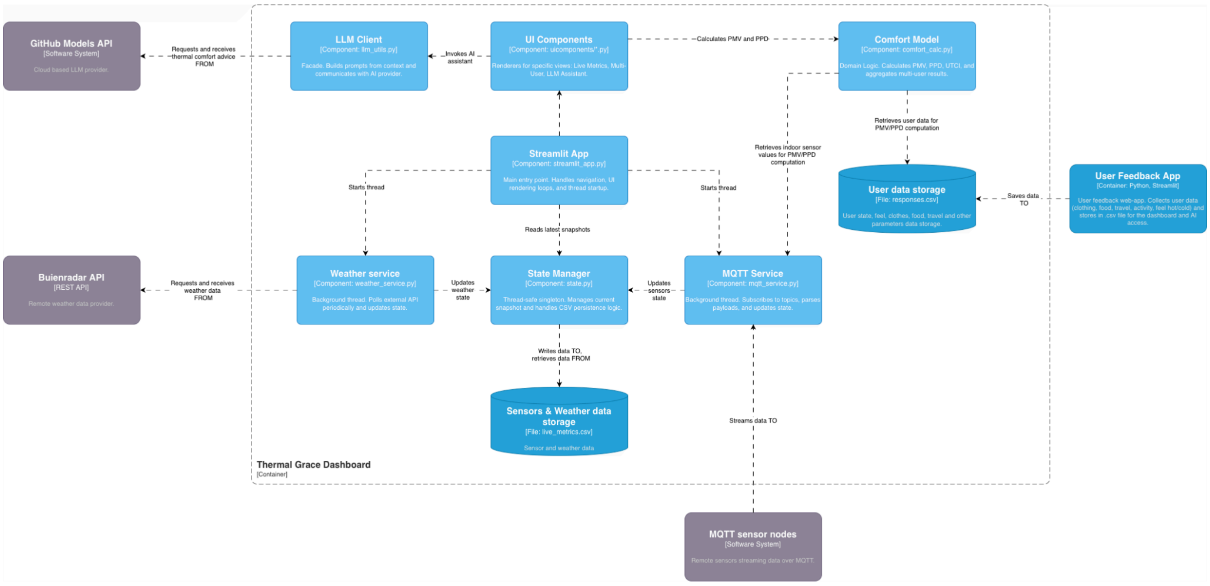

Image 3: C4 diagram level 3 (Components)
High resolution: https://drive.google.com/file/d/1mlP2k4rSQP9jo3I0DwF56UoDrmGmeA2P

The current implementation meets the requirements by allowing for the expansion of API integrations or data points through modular utility functions. New sensors, such as the RD03D mmWave Radar and MTP40F NDIR CO2 nodes, are easily integrable via the MQTT receiver which handles diverse topics and payloads (up to 3 radar targets). Thermal comfort metrics (PMV/PPD) and UTCI are computed using the `PyThermalComfort` module.

All environmental and occupancy data are logged into `.csv` files for long-term analytics and potential AI model training. A key feature is the Adaptive Multi-User Tracking, which aggregates feedback from multiple occupants today to provide group-level comfort metrics and personalized LLM recommendations. Users provide feedback on their current sensation, activity, and clothing without sharing sensitive personal information; each entry is tracked via a unique identifier (UID).

System Inventory: Existing vs New/Custom-Built

Existing systems/services

- **MQTT Broker (Mosquitto / managed broker):** message transport for sensor data.
- **Buienradar API:** external weather provider (REST API, HTTP).
- **GitHub Models API:** external LLM provider (REST API, HTTP).
- **PyThermalComfort - Thermal comfort computation module:** PMV/PPD logic.

New/custom-built systems in this project

- **Thermal Grace Dashboard (Streamlit app):** UI + background services + computations.
- **MQTT ingest logic:** topic parsing, validation, snapshot update.
- **User Feedback App:** Streamlit survey app + response storage (CSV).

- **Sensor node firmware/scripts:** Pico publishers for CO_2 and air/mmWave (if considered part of the overall system).

Quality Attributes

Security

Threats to address

- Unauthorized access to MQTT topics (spoofed sensor data).
- Leakage of API tokens (GitHub Models / weather).
- Exposure of user feedback (personal/contextual data).
- Dashboard endpoint exposure on the network.

Controls / decisions to document

MQTT security

- Broker association with specific client IDs (e.g., `pico-mmwave-air`).
- Use of local network boundaries (192.168.50.x) to prevent external exposure.
- Standardized JSON validation on ingress to prevent malformed data injection.

API credential handling

- Secure injection of `github_models_token` via Streamlit `secrets.toml`.
- Rate-limiting LLM requests (e.g., gated by the "Ask Assistant" button) to prevent API exhaustion.

Network boundaries

- Dashboard access limited to trusted network/VPN; optional reverse proxy with auth.
- Firewall rules: only required ports open (MQTT, Streamlit).

Data protection & privacy

- Minimize Personally Identifiable Information (PII) in feedback; clarify data retention.
- Access control to CSV files; file permissions; backups.

Logging and auditability

- Do not log secrets; sanitize payload logs.
- Record message provenance (topic, timestamp) for traceability.

Secure defaults

- Disable DEBUG logs in production (or redact).
- Input validation for MQTT payloads and feedback fields.

Scalability

Expected growth vectors

- More sensor nodes / more topics.

- Higher message frequency.
- More concurrent dashboard viewers.
- Longer retention / larger CSV files.

Scalability decisions

MQTT extensibility

- Decoupled pub/sub architecture allows adding new specialized nodes (like the MTP40F CO2 sensor) without changing the core MQTT engine.

Storage evolution path

- Current CSV-based storage uses a "Last-N" row fetch strategy to ensure the Streamlit UI remains performant as logs grow.
- Future-proofed for migration to an InfluxDB or TimeScaleDB instance should sub-second querying of historical years be required.

Performance safeguards

- Rate-limit LLM calls; cache recent advice; handle API rate limits/timeouts.
- Avoid full-file CSV reads on every UI refresh (use tail/last N rows).

Multi-user Aggregation

- Logic is optimized to filter unique UIDs from the feedback CSV, ensuring the LLM prompt remains within token limits while providing comprehensive context.

Deployment scaling

- Single instance initially; scale-out path (multiple dashboard instances) requires shared storage + shared state (DB + broker remains central).

Architecture Constraints and Assumptions

- Runs on a single host (initially) with local CSV files.
- External APIs depend on internet availability.
- Sensors rely on Wi-Fi stability and broker uptime.
- Comfort model assumptions - some values are fixed and hardcoded.

Technology Choices and Rationale

- Why MQTT (pub/sub, decoupling, easy sensor onboarding).
- Why Streamlit (fast iteration, research prototype).
- Why CSV (transparent, easy export) + acknowledged limits.
- Why external LLM API (capability vs hosting cost/complexity).

Conclusion

The described software architecture design represents the finalized prototype for Vitality HUB's Perceived Thermal Comfort IoT system. The design is modular, reproducible, and utilizes standard frameworks to ensure extensibility. With the integration of the C4 Level 3 Component architecture and the implementation of multi-user adaptive logic, the system fulfills the project goals for real-time comfort monitoring and AI-driven recommendations. Future research should leverage the collected historical data to further refine the personalized comfort models.

Reference

1. Kraskov A., Kaszuba B., (2025), Vitality HUB Perceived Thermal Comfort | Project Plan, Thermal Comfort-as-a-Service (CaaS) IoT system, FHICT Delta, [Online], Available: https://drive.google.com/file/d/1BXHXID4_SPatpJrU1mDwF0ZKnESI2W_U
2. Kraskov A., Kaszuba B., (2025), Vitality HUB Perceived Thermal Comfort | Requirements, Thermal Comfort-as-a-Service (CaaS) IoT system, FHICT Delta, [Online], Available: https://drive.google.com/file/d/1NooyuNM97BFz3jYOX2aPSRkbFZ7f5yJF
3. Kraskov A., (2025), Thermal Grace - VHUB Comfort-as-a-Service IoT System, Hardware Architecture Design, FHICT Delta, [Online], Available: https://drive.google.com/file/d/1sWcylxS85DdrUbDKQ6F5CqS8aSHxkFRN