EDGE–CLOUD ARCHITECTURE DESIGN FOR INTELLIGENT PC ASSISTANT

Chen Khoon Seah

chenkhoon@yahoo.com

Contents

Intelligent PC Assistant – Architecture Overview 2

1. Introduction 2

2. Core Components 2

2.1 Telemetry Collection (Edge) 2

2.2 Local Model (Edge) 3

2.3 Edge-Cloud Services 3

2.4 Config File (config.yml) 3

3. Data & Control Flow 3

Setup Phase: 3

Telemetry Ingestion: 4

Local Inference: 4

Cloud Interaction (if enabled): 4

System Response: 4

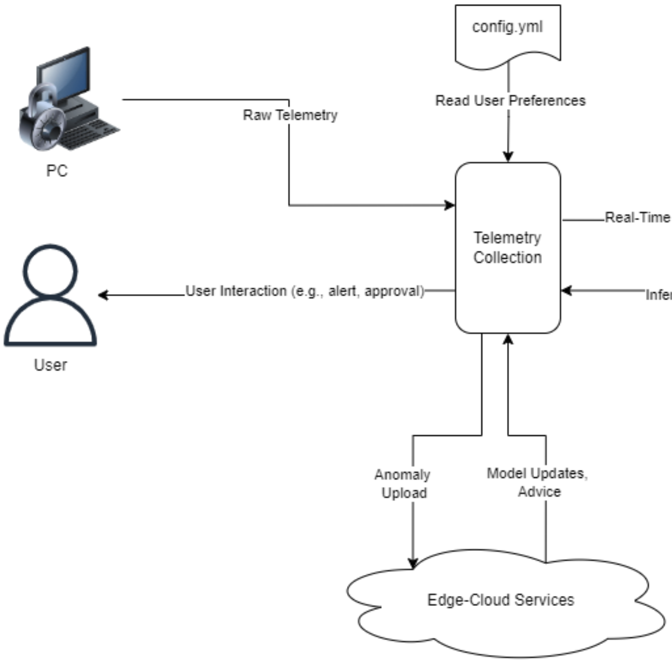
Intelligent PC Assistant – Architecture Overview

1. Introduction

This architecture outlines the design of a hybrid **Edge–Cloud AI system** for a proactive support assistant on client devices. The system leverages **local inference for responsiveness and privacy**, while utilizing **cloud intelligence** for deeper insights, updates, and scalability. Configuration and behaviour are governed by user-defined preferences at initial setup.

---

2. Core Components



Edge–Cloud Intelligent Assistant Workflow

2.1 Telemetry Collection (Edge)

- Acts as the central coordinator on the device.
- Collects **raw system data** (e.g., CPU usage, thermal readings).
- Routes data to both the **Local Model** for immediate analysis and to the **Edge-Cloud** for further processing (based on user consent).
- Manages event triggers, user notifications, and edge-cloud communication.
- Reads user preferences from a configuration file (config.yml) at startup.

2.2 Local Model (Edge)

- Performs **lightweight, real-time inference** on incoming telemetry data.
- Examples include anomaly detection (e.g., overheating) or command interpretation.
- Operates **offline** with low latency and respects privacy by processing data locally.
- Returns actionable results (e.g., "thermal anomaly detected") to the Telemetry Collection module.

2.3 Edge-Cloud Services

- Performs **heavy computation** using cloud resources.
- Handles:
  - Large-scale telemetry aggregation
  - Fleet-wide pattern detection
  - Generative support responses (via LLMs)
  - Recommendation engines
- Sends back:
  - **Cloud-generated insights** to enhance user support
  - **Model updates** to improve on-device inference capabilities

2.4 Config File (config.yml)

- Stores user-defined settings including:
  - Whether telemetry is collected
  - Whether data is sent to the cloud
  - Whether cloud model updates are allowed
- Ensures **compliance with privacy regulations** (e.g., GDPR, PDPA).

---

3. Data & Control Flow

Setup Phase:

- System reads config.yml to determine telemetry, cloud, and model usage settings.

Telemetry Ingestion:

- Device generates system data → captured by Telemetry Collection.

Local Inference:

- Telemetry data is passed to Local Model for real-time analysis.
- Inference results (e.g., detected anomaly) are returned.

Cloud Interaction (if enabled):

- Telemetry data is sent periodically or upon trigger to Edge-Cloud services.
- Cloud returns:
  - **Contextual insights** (e.g., recommended user actions)
  - **Model updates** (e.g., improved detection logic)

System Response:

- Assistant takes appropriate action:
  - Alert user
  - Apply local optimization
  - Escalate to support (cloud integration)

---

4. Key Benefits

Feature	Benefit
Edge Inference	Low-latency decisions, offline operation, preserves privacy
Cloud Intelligence	Scalable insights, support automation, model retraining
User Configurability	Transparent setup, customizable privacy & control options
Modular Design	Extensible, maintainable architecture for real-world use

---

5. Conclusion

This hybrid architecture balances the **responsiveness and privacy of edge computing** with the **depth and scalability of cloud-based AI**. It is suitable for deployment on millions of client devices, ensuring proactive support experiences without compromising user trust or compliance requirements.