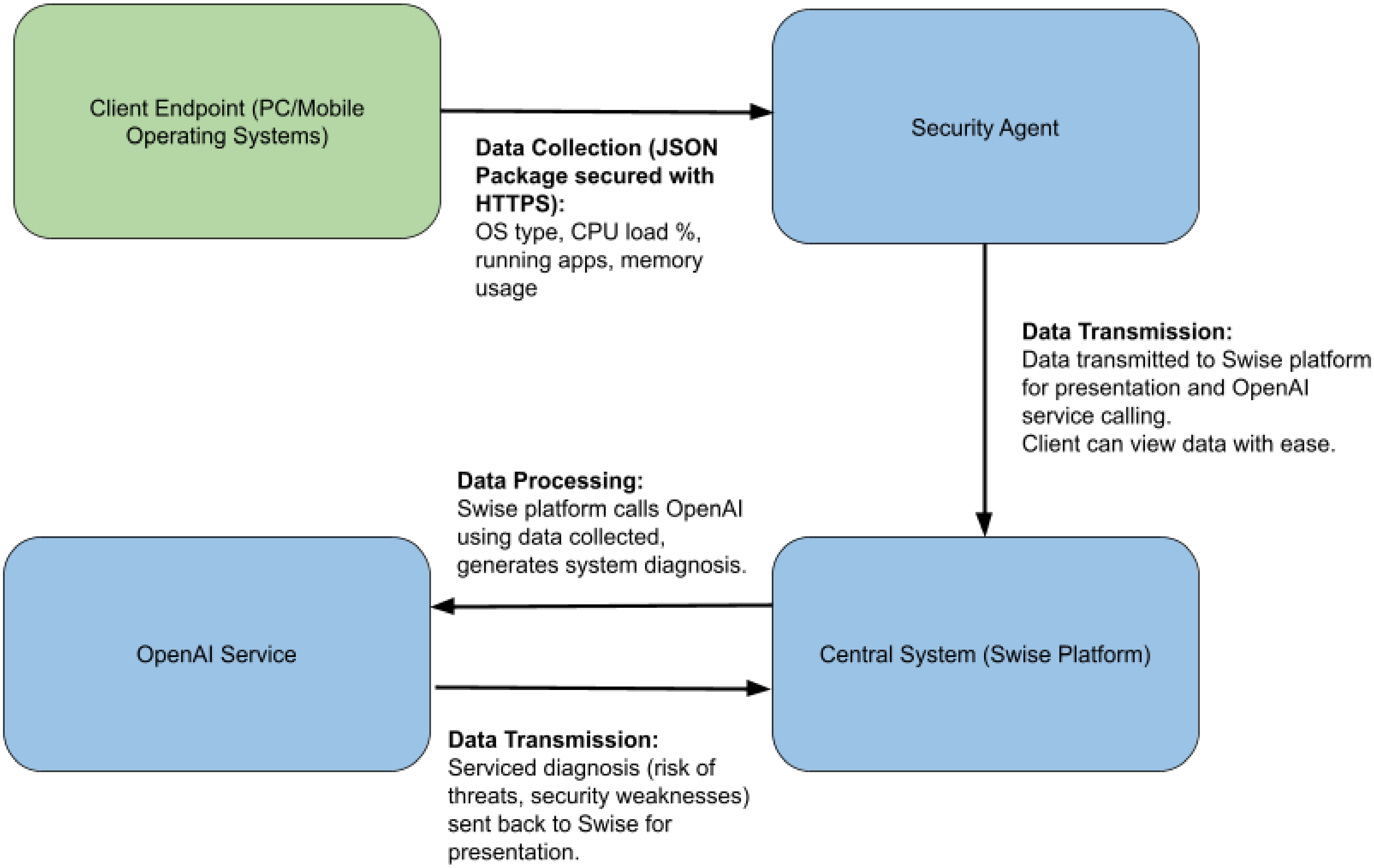
PART ONE

Central Destination Points:

1. Security Agent (Used in client's device)
2. Central AI System (Swise.ai platform)
3. OpenAI (LLM model)


Client Endpoint (PC/Mobile Operating Systems)
Security Agent
OpenAI Service
Central System (Swise Platform)

Data Collection (JSON Package secured with HTTPS):
OS type, CPU load %, running apps, memory usage

Data Transmission:
Data transmitted to Swise platform for presentation and OpenAI service calling.
Client can view data with ease.

Data Processing:
Swise platform calls OpenAI using data collected, generates system diagnosis.

Data Transmission:
Serviced diagnosis (risk of threats, security weaknesses) sent back to Swise for presentation.

Client Endpoint Agent Core Components:

1. Collection Module
    a. Collects generic system data
        i. OS version
        ii. CPU + Memory usage
        iii. Running processes (high-level)
        iv. Installed programs
        v. Device metadata (non-PII (personably identified information)

2. Communication Module
    a. Networking component that sends data back to Central AI System
        i. Securely send telemetry to Central AI System
        ii. Handle retries and exponential backoff if network fails
        iii. Use TLS encryption for all communication
        iv. Attach authentication token or certificate
3. Scheduler
    a. When the agent collects and transmits data
        i. Polling Mode (continually runs every few moments)
        ii. Interrupt/Event-based Mode (runs on a condition: CPU performance influx, memory usage influx)

Low Impact Performance Strategy:

1. Use Event-Based instead of Polling Scheduler
    a. **Polling** uses significantly more computing power due to continuously running even when nothing of significance is occurring
    b. Event-based only triggers on an important event (interrupt: high CPU usage, memory usage), using less computing power as agent only collects or transmits data during significant intervals

2. Utilise Data Compression
    a. Reduce size of collected data before transmission
    b. Critical for mobile end-points due to 4G/5G bandwidth considerations
    c. Less storage and processing power needed by the Central AI System for transmitted data

Security Best Practices:

1. Agent-to-Central-System Communication
    a. Enforce Encryption
        i. All communication between the agent and the central system must use **HTTPS (HyperText Transfer Protocol Secure)**, ensuring protection
    b. Mutual Authentication
        i. The server and the agent both verify each other's identity using certificates.
2. Use of the OpenAI API Key
    a. Key Never Stored on the Agent
        i. Key only utilised by Central System for safety
3. Data Privacy
    a. Data Minimisation
        i. Achieve minimum data collection for what is required
    b. PII Censoring
        i. Produce system that automatically prevents sending PII

Future Considerations (Cross-Platform Integration):

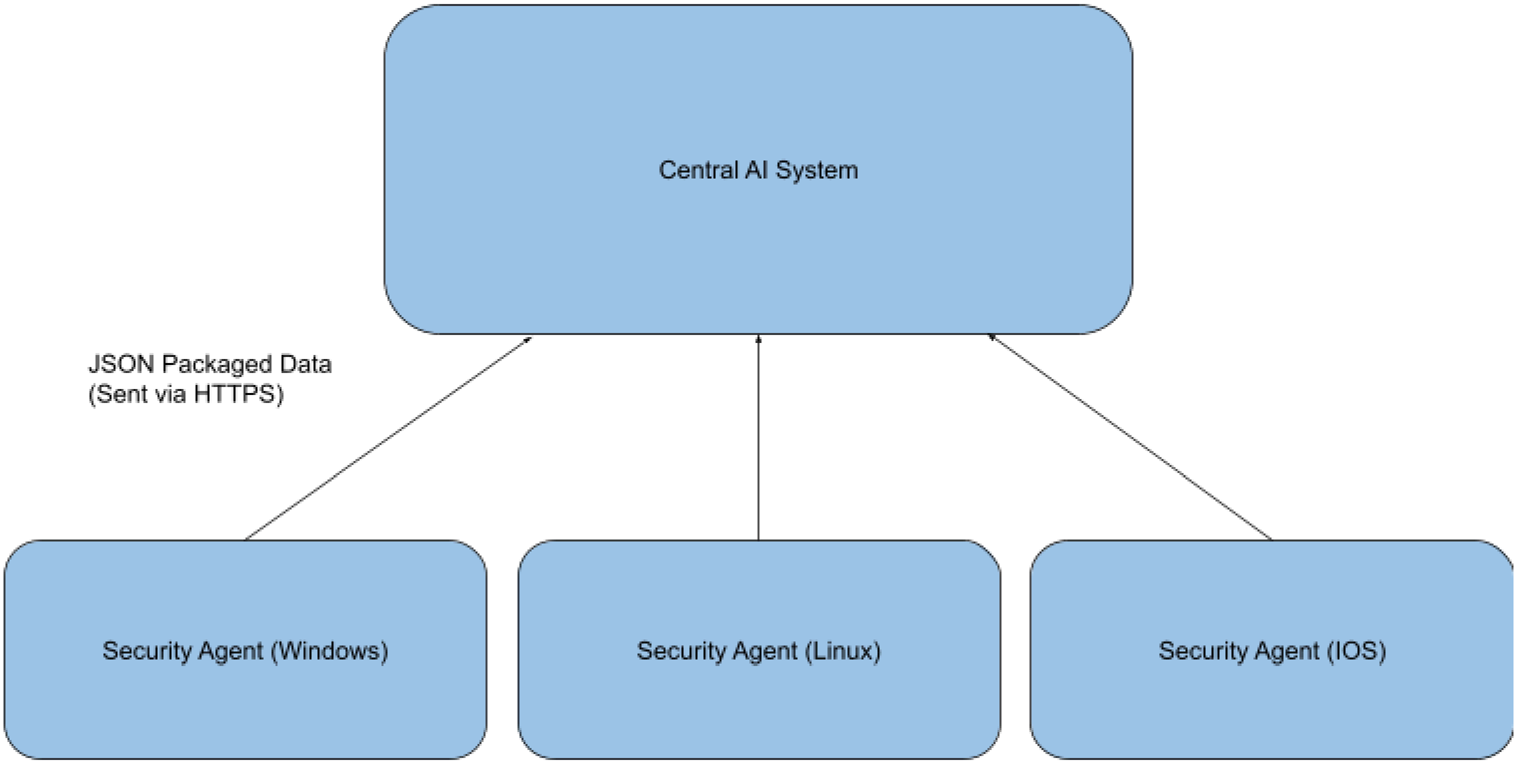
*Security Agent should be installed on a device locally in order for data extraction to occur.

Linux (Ubuntu) Integration:

1. Similar to Windows OS, use shell commands to access data

IOS Integration:

1. Utilise ‘Mobile Device Management (MDM)’ to access to installed apps, last user + timestamp and OS update statuses
2. Cannot use shell commands to access information


Central AI System
JSON Packaged Data
(Sent via HTTPS)
Security Agent (Windows)
Security Agent (Linux)
Security Agent (IOS)