

Industry	Building Information Management, Smart Home, Smart Factory
Use Case Title	Smart Home Data Acquisition and Ingestion Pipeline for a Data Fusion Platform Phase 1
About the Customer	The Client is a Consumer Electronics Taiwanese company. We worked on this project as a Technology Partner to NTU, Singapore & the Client
Business Problem	<p>The Client had a plan to build an Energy Optimization platform for Smart Homes and Smart Building. It however required a deep expertise in Sensor Data Fusion, Data Pipeline and AI. They were looking for an efficient and scalable solution to acquire and ingest data from these smart homes into a centralized data lake. The existing challenges include:</p> <ul style="list-style-type: none"> <li>● Data Variety: Smart homes generate a diverse range of data, including sensor data, video feeds, and plug data.</li> <li>● Real-Time Streaming: Smart home data is time-sensitive and requires real-time streaming capabilities.</li> <li>● Data Volume and Scalability: As the number of smart homes increases, so does the volume of data. The solution must be scalable enough to handle data from a large number of smart homes simultaneously.</li> <li>● Data Security and Privacy: The sensitive nature of smart home data necessitates secure communication mechanisms and the protection of user privacy</li> <li>● Lack of sufficient data for AI Model training</li> </ul>
Solution	<p>Scalable data pipeline was developed to acquire and ingest data from redundant sensors and high-resolution cameras installed at the smart home.</p> <p><b>Stats</b></p> <ul style="list-style-type: none"> <li>▪ Pipelines for around 150 home were created for testing.</li> <li>▪ Data was generated synthetically for each home at granularity of 1 minute</li> </ul> <p><b>Deployment</b></p> <ul style="list-style-type: none"> <li>▪ Data Acquisition component /Edge utility was deployed on Raspberry PI edge device.</li> <li>▪ Data ingestion pipeline were deployed on AWS cloud.</li> </ul> <p><b>Approach &amp; Implementation</b></p> <ul style="list-style-type: none"> <li>▪ We developed streaming data pipelines &amp; Apache Kafka request response mechanism to create end to end pipeline for acquisition &amp; ingestion.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ <b>Streaming Pipelines:</b> Scalable streaming pipelines were built using Kafka and PySpark to acquire and process sensor data, video feeds, and plug data from smart homes.</li> <li>▪ <b>Edge Utility Modules:</b> Edge utility modules, including an IP camera reader, IoT utility, and edge filtering modules, were developed to acquire data from smart home devices at the edge.</li> <li>▪ <b>Kafka Request-Response Mechanism:</b> A request-response mechanism using Kafka was implemented to establish secure and efficient communication with edge devices without relying on public IP addresses.</li> <li>▪ <b>Data Storage and Management:</b> Acquired data is stored in MongoDB, providing a NoSQL database for efficient storage and easy retrieval. AWS S3 is used for storing large volumes of data, ensuring scalability and durability.</li> </ul>
Outcome	<ul style="list-style-type: none"> <li>▪ Successfully acquired and ingested data from multiple smart homes into the centralized database/data lake.</li> <li>▪ Real-time access to sensor data, video feeds, and plug data for monitoring and analysis purposes.</li> <li>▪ Enabled efficient data processing and analysis for insights and actionable information in the smart home domain.</li> <li>▪ Improved scalability and performance of the data acquisition and ingestion platform, allowing for easy expansion and handling of increasing data volumes.</li> <li>▪ Enhanced security and privacy through the use of edge utility modules and secure communication mechanisms.</li> <li>▪ By implementing the Smart Home Data Acquisition and Ingestion Platform, we have successfully created a scalable and efficient system for acquiring and processing data from multiple smart homes, providing valuable insights and enabling improved smart home management and monitoring capabilities.</li> </ul>