MCMC MTSFB TC G047:2024

TECHNICAL CODE

SMART CITY - DATA INTEGRATION PLATFORM

Developed by

Registered by

Registered date: 8 August 2024

MCMC MTSFB TC G047:2024

Development of technical codes

The Communications and Multimedia Act 1998 (Laws of Malaysia Act 588) ('the Act') provides for a Technical Standards Forum designated under Section 184 of the Act or the Malaysian Communications and Multimedia Commission ('the Commission') to prepare a technical code. The technical code prepared pursuant to Section 185 of the Act shall consist of, at least, the requirements for network interoperability and the promotion of safety of network facilities.

Section 96 of the Act also provides for the Commission to determine a technical code in accordance with Section 55 of the Act if the technical code is not developed under an applicable provision of the Act and it is unlikely to be developed by the Technical Standards Forum within a reasonable time.

In exercise of the power conferred by Section 184 of the Act, the Commission has designated the Malaysian Technical Standards Forum Bhd ('MTSFB') as a Technical Standards Forum which is obligated, among others, to prepare the technical code under Section 185 of the Act.

A technical code prepared in accordance with Section 185 shall not be effective until it is registered by the Commission pursuant to Section 95 of the Act.

For further information on the technical code, please contact:

Malaysian Communications and Multimedia Commission (MCMC)
MCMC Tower 1
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel : +60 3 8688 8000
Fax : +60 3 8688 1000
Email : stpd@mcmc.gov.my
Website: www.mcmc.gov.my

OR

Malaysian Technical Standards Forum Bhd (MTSFB)
MCMC Tower 2
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel : +60 3 8680 9950
Fax : +60 3 8680 9940
Email : support@mtsfb.org.my
Website: www.mtsfb.org.my

Contents

Page

Committee representation iii
Foreword iv

0. Introduction 1
1. Scope 1
2. Normative references 1
3. Abbreviations 2
4. Terms and definitions 2
   4.1 Analytical tools 2
   4.2 Administrator 2
   4.3 Application Programming Interface (API) documentation 2
   4.4 Batch data processing 3
   4.5 Complex events 3
   4.6 Data cataloguing 3
   4.7 Data security and governance 3
   4.8 Data variety 3
   4.9 Data velocity 3
   4.10 Data veracity 3
   4.11 Data volume 3
   4.12 Data warehouses 3
   4.13 Error handling 3
   4.14 Historical data 3
   4.15 Internet of Things (IoT) devices 3
   4.16 Operational metadata 3
   4.17 Personal data 4
   4.18 Processing pipelines 4
   4.19 Real-time data processing 4
   4.20 Scalability 4
   4.21 Security and privacy 4
   4.22 Semi-structured data 4
   4.23 Structured data 4
   4.24 Technical metadata 4
   4.25 Unstructured data 4
5. Data Integration Platform 4
   5.1 Overview 4

5.2 Platform architecture 5
5.3 Stakeholders 6
5.4 Platform components 6
6. General specifications 19

Committee representation

This Technical Code was developed by the Internet of Things and Smart Sustainable Cities Working Group of the Malaysian Technical Standards Forum Bhd (MTSFB), which consists of representatives from the following organisations:

CelcomDigi Berhad

Cyberview Sdn Bhd

Favoriot Sdn Bhd

Kiwitech Sdn Bhd

Maxis Broadband Sdn Bhd

SIRIM Berhad

Sunway University College Sdn Bhd

TM Technology Services Sdn Bhd

UCSI Education Sdn Bhd

Universiti Malaya

Universiti Putra Malaysia

Universiti Sains Islam Malaysia

Universiti Teknologi MARA

Foreword

This Technical Code for Smart City - Data Integration Platform ('Technical Code') was developed pursuant to Section 185 of the Communications and Multimedia Act 1998 (Laws of Malaysia Act 588) by the Internet of Things and Smart Sustainable Cities Working Group of the Malaysian Technical Standards Forum Bhd (MTSFB).

This Technical Code shall continue to be valid and effective from the date of its registration until it is replaced or revoked.

SMART CITY - DATA INTEGRATION PLATFORM

0. Introduction

The smart city data integration platform is hailed as an instrumental innovation that significantly impacts urban living. This platform denotes the seamless integration of digital technology into urban infrastructure and services. The applications of smart city integration can span from the basic and cost-effective, such as real-time traffic monitoring, to the more complex and high-end, like integrated emergency response systems. Hence, the use cases, implementations, and applications of smart city integration are immensely varied.

This entails the use of digital technology to streamline urban mobility, ensuring timely and efficient transit. For instance, smart traffic lights can adapt to real-time traffic conditions, helping to mitigate congestion, while smart parking systems direct drivers to available parking spots, saving time and fuel. However, there is a rising concern that the adoption of smart city solutions increases the risk of silo deployments, poses challenges of security and privacy, and leads to non-standard solution deployments.

In light of this, there is a pressing requirement for the establishment of a baseline smart city data integration platform standard, which can serve as the cornerstone and guidance for the security prerequisites in smart city applications and services.

1. Scope

This Technical Code establishes the framework for the systems and processes for the development of a smart city application which requires data integration among multiple systems, data sources, and interfaces. The framework aims to streamline data collection, processing, and analysis across various applications developed by multiple vendors for smart city initiatives.

The primary objective is to create a standardised, efficient, and scalable framework for integrating data from different smart city applications. The framework addresses real-time and batch data processing and emphasises security, privacy, interoperability, and compliance.

2. Normative references

The following normative references are indispensable for the application of this Technical Code. For dated references, only the edition cited applies. For undated references, the latest edition of the normative references (including any amendments) applies.

Act 709*, Personal Data Protection Act 2010*

MCMC MTSFB TC G039, *Industrial Internet of Things (IIoT) Connectivity and Communications Framework*

MS ISO/IEC 27001, *Information technology - Security techniques - Information security management systems - Requirements*

ISO/IEC 24039, *Information technology - Smart city digital platform reference architecture - Data and service*

Malaysia Smart City Framework, 2018, Ministry of Housing and Local Government

3. Abbreviations

For this Technical Code, the following abbreviations apply.

AI	Artificial Intelligence
API	Application Programming Interface
BI	Business Intelligence
CCTV	Closed-Circuit Television
CSV	Comma-Separated Values
ELT	Extract, Load and Transform
ETL	Extract, Transform and Load
GIS	Geographic Information System
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MQTT	Message Queuing Telemetry Transport
NoSQL	Not Only SQL
ONVIF	Open Network Video Interface Forum
PDPA	Personal Data Protection Act
REST	Representational State Transfer
RTSP	Real-Time Streaming Protocol
URL	Uniform Resource Locator
XML	Extensible Markup Language

4. Terms and definitions

For this Technical Code, the following definitions apply.

4.1 Analytical tools

A range of tools and techniques such as statistical analysis, predictive modelling, machine learning algorithms, and data mining capabilities provided by the platform.

4.2 Administrator

A user who has the highest privilege level possible for a user of the device, which can mean they can change any configuration related to the intended functionality. The user who owns or who purchased the device.

4.3 Application Programming Interface (API) documentation

Well-documented instructions on how to use the Application Programming Interface (API), including headers, parameters, and return data format.

4.4 Batch data processing

Collecting and processing data at regular intervals before storage.

4.5 Complex events

Patterns and correlations in incoming data streams that trigger an action.

4.6 Data cataloguing

Organising and categorising data assets with metadata about each data asset.

4.7 Data security and governance

Implementing security measures like access controls, encryption, and auditing in data lakes.

4.8 Data variety

Refers to the different types of data, including structured, unstructured, and semi-structured data.

4.9 Data velocity

This denotes to the speed at which new data is generated and collected.

4.10 Data veracity

This relates to the trustworthiness or quality of the data collected.

4.11 Data volume

This refers to the amount of data generated or collected.

4.12 Data warehouses

Centralised repositories of data that store data in a structured format.

4.13 Error handling

Providing clear, useful error messages for better troubleshooting.

4.14 Historical data

The data that is stored over a long term to provide a historical view.

4.15 Internet of Things (IoT) devices

Non-standard computing hardware such as sensors, actuators or appliances that connect to a network and can transmit data.

4.16 Operational metadata

Information about the permitted or expected operations performed on the data, such as data collection time and usage permissions.

4.17 Personal data

Any information relating to an identified or identifiable natural person.

4.18 Processing pipelines

Systems for continuously undertaking actions in a sequential method on the real-time or batch data processing system and optimising it as necessary.

4.19 Real-time data processing

The immediate collection and processing of data after ingestion and metadata processing and before storage.

4.20 Scalability

The ability of the platform to handle increasing data volumes and complexity.

4.21 Security and privacy

Measures to protect sensitive data, including user authentication, data encryption, and access controls.

4.22 Semi-structured data

Data that has some form of organisation but is not as rigidly structured as relational data, is commonly found in Extensible Markup Language (XML), JavaScript Object Notation (JSON), or Not Only SQL (NoSQL) databases.

4.23 Structured data

Data is organised in a predefined manner, like in relational databases or Comma-Separated Values (CSV) files.

4.24 Technical metadata

Information about the data formats, size, and data type.

4.25 Unstructured data

Data without a pre-defined model or organisation, examples include Closed-Circuit Television (CCTV) footage and text files.

5. Data Integration Platform

5.1 Overview

A data integration platform is a platform that provides smooth data exchange, ensuring consistent city operations. To support these processes, this Technical Code focuses on the following key areas.

a) Data acquisition

Gathering information from various sources and devices.

b) Metadata management

Link and store metadata repositories with the data-originating process.

c) Data ingestion

Organised control of stored data, enabling easy access and usage.

d) Data analysis (process)

Providing insights from the processed data for actionable data-driven decision-making. This includes real-time data processing and batch data processing.

e) Data storage

Safe and effective retention of data. The data storage medium includes the data lakes and the data warehouses.

f) Data visualisation

Processes facilitate information amalgamation and scrutiny, providing the visualisation or dissemination of data to relevant stakeholders and systems. These processes result in a data-driven decision-making approach to help city management.

5.2 Platform architecture

Figure 1 below provides an overview of the data integration lifecycle, from the data ingestion to processing and data dissemination (left to right in the diagram). The processes indicated in the architecture and the relevant processes are described in detail in the rest of this Technical Code.

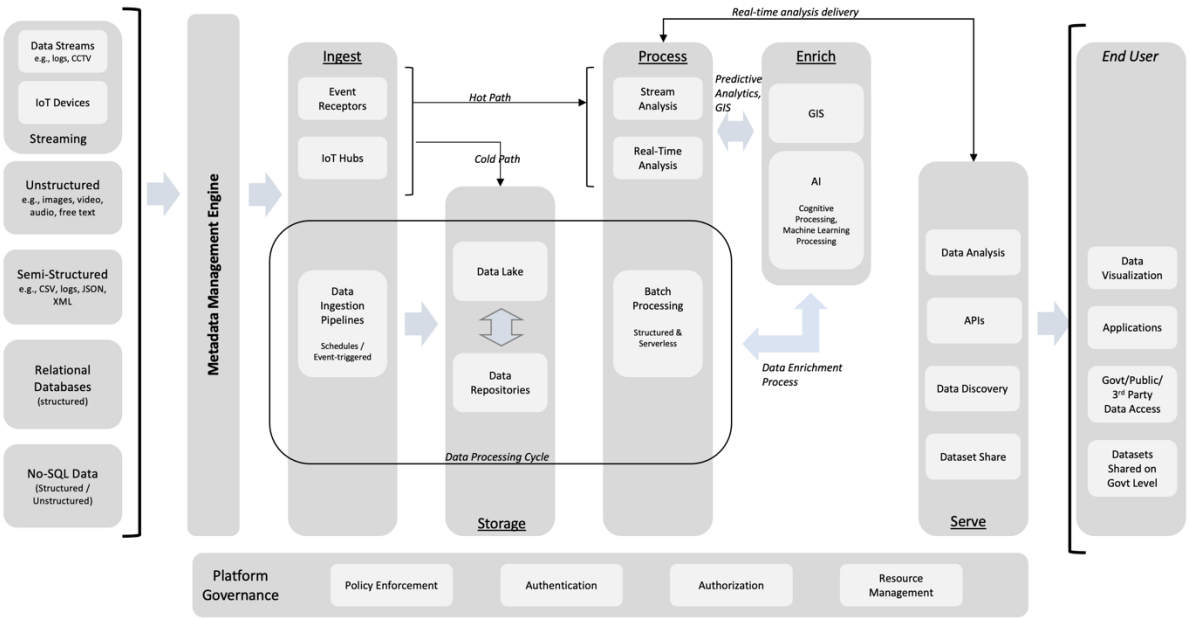

Figure 1. Smart city data integration, processing and management architecture

5.3 Stakeholders

A smart city data integration platform involves multiple stakeholders that are directly or indirectly engaged or use the applications or solutions associated with the platform.

These stakeholders may include the following.

a) Smart city planners.

b) Technology or solution providers.

c) Data providers.

d) Planners and developers.

e) Smart city solution architects.

f) Smart city consultants.

g) Academia.

h) Emergency services.

5.4 Platform components

The smart city data integration platform consists of multiple individual components that work together to provide a stable, coherent, and seamless data management for smart city operations.

5.4.1 Data ingestion

The data acquisition and collection process are very wide and versatile area that depends on;

a) Nature of data - ascertain, if the data is an IoT or other stream of data, or batch transactional data, or any other data type,

b) Data sources - ascertain the systems or processes that generate data e.g., IoT devices, existing datasets, database connections, or data exports.

c) Data acquisition medium - ascertain at how the data is acquired e.g. API, log dumps, file export, or database connections.

To design and configure data ingestion processes, the data needs to be classified into the following categories.

5.4.1.1 Data ingestion categories

The data being ingested falls into the following broader categories as below.

a) IoT and Data streams

The devices such as sensors and smart devices generate a vast amount of real-time data. Also, audio and video devices generate large amount of data that is acquired as data streams. The data streams are a continuous flow of data generated by various sources. They can be collected using stream processing tools of the data integration platform.

b) Structured and unstructured data

Structured data is data that is organised in a predefined manner, such as data in relational databases or CSV files. Unstructured data, on the other hand, lacks a specific form of a model, such as text files or social media posts. The collection method depends on the source and type of data.

c) Relational and NoSQL databases

Data can also be collected from various types of databases. Relational databases store data in a structured format, usually in tables, while NoSQL databases are used for storing unstructured or semi-structured data.

5.4.1.2 Data acquisition process

For the above data categories and the nature of data sources, there are data acquisition processes that follow the following steps.

a) Identify and catalogue data sources

The first step is to identify and catalogue the sources of the data. This could include data from IoT devices, data streams, databases, data dumps source files, etc.

b) Establish data collection method

Establish appropriate data collection methods depending on the source and type of data. This could include setting up APIs for IoT devices, stream processing tools for data streams, database queries for relational and NoSQL databases, or data file imports in case of data dumps or exported files.

c) Data collection

Once the methods are established, the data collection processes can be executed, and the data transfer takes place.

d) Data validation

Ensure that the data being transferred is completely executed, received, and stored. Create processes for failed APIs, broken data streams failed data queries, and fallback processes.

e) Data transfer security

The data being transferred is ensured to take place securely and efficiently.

At each step of data acquisition, it is important to document and communicate the data ingestion plan with the data owners or stakeholders.

5.4.2 Metadata management

Metadata management plays an important role in the overall process of data integration. It occurs after the data collection and before any data storage or pre-processing, in the data lifecycle. The metadata helps organise the data, making it easier to find, access, and use. It also helps in ensuring data quality, data accessibility, sensitivity, ownership, and permitted use duration, if any.

The metadata management stores and provides information about when and how the data was collected, what would be the permitted uses, and any compliances required e.g., Personal Data Protection Act (PDPA), or the data schema and sensitivity.

5.4.2.1 Metadata types

For a smart city data integration platform, there are two primary types of metadata which are as follows.

a) Technical metadata

This includes information about the data formats, size, data type, etc.

b) Operational metadata

This includes information about the permitted or expected operations performed on the data, such as when it was collected, last updated, last accessed by whom, or the context of data usage permission, if the data owner is a third-party stakeholder.

5.4.2.2 Metadata management process

The data integration platform involves the metadata management system to undertake the following processes to manage the metadata.

a) Metadata creation

This process involves generating metadata at the time of data ingestion or data collection.

b) Metadata storage

Once created, the metadata needs to be stored in a way that it can be easily accessed and used. The metadata storage may involve using the metadata repository, which is a type of database designed specifically for storing metadata.

c) Metadata usage

Metadata is used to help find, understand, and manage data.

d) Metadata maintenance

Metadata may need to be updated or fixed over a period of time with new data entities being ingested and existing ones updated or deleted. This process needs to be in place so that the metadata remains accurate, reliable, and up to date.

5.4.3 Data processing

The data processing forms an integral and most critical part of a smart city data integration platform, the data processing is responsible for converting the raw data into the insights, forecasts, analysis, etc.

The data processing function falls into two broader categories, real-time data processing, and batch data processing.

5.4.3.1 Realtime data processing

Real-time data ingestion or processing plays a crucial role in the overall process of data integration. It occurs at the very beginning of the data lifecycle, right after data ingestion and metadata processing and before data storage.

It is required so that the data is collected and processed as soon as it is generated, processed, and stored to facilitate the most up-to-date information for decision-making and analysis. It is important to treat real-time data ingestion in a separate category and not confuse and treat real-time or stream data

as batch data processing, as this could cause delays in data collection and processing, which could lead to outdated or inaccurate data being stored and processed.

5.4.3.1.1 Data types for real-time processing

Although there may be multiple sources that generate the stream or real-time data, the following are the most common types of stream or real-time data in the context of smart cities that the smart city data integration platform should be ready to cater for but not limited to below.

a) IoT devices

These devices, such as sensors and smart meters, generate a vast amount of real-time data. The data from these devices is typically collected via APIs or direct data transmission to the data ingestion component for the smart city data integration platform.

b) Data streams

Data streams are a continuous flow of data generated by various sources. This could be video stream from CCTV feeds via Real-Time Streaming Protocol (RTSP) or Open Network Video Interface Forum (ONVIF) integration, system logs or other similar data sources.

5.4.3.1.2 Realtime data process components

As real-time data is ingested, several basic data components work together to ensure data consistency and reliability. In the real-time data context, these components shall be highly optimised and capable of scaling with the scale of data being ingested, these operations are as follows.

Real-time data processing components requires the data integration platform to be able to undertake the following functions and specifications.

a) Data pre-processing

This component undertakes data pre-processing operations such as data cleaning, formatting, transformation, and aggregation of the real-time data as the data is ingested.

b) Real-time processing capabilities

The platform shall deploy tools that can be used to process the data in real-time as it is being ingested.

c) Complex event processing

The event processing, or, event processing hub, detects the patterns and correlations in the incoming data stream. It triggers an event as the event hub detects the anomaly outside of the predefined ranges.

d) Data filtering

The unnecessary data shall be removed from the data being ingested. Given the volume of real-time data, not all incoming data might be relevant. Filtering ensures only useful data is processed further. In general terms, it is called to remove the data noise.

e) Data transformation

The data being ingested (e.g. stream data, Message Queuing Telemetry Transport (MQTT), etc.) might not be in the correct format required for storage or further analysis. Data transformation involves converting the raw data into a format suitable for analysis. This could involve operations like parsing dates, converting data types, or normalising text data.

f) Data aggregation

Aggregate and summarise the data as it is ingested. It may use data transformation to create new dimensions in the current data or extend the data ingested. For example, there might be a requirement to calculate the average temperature from a stream of temperature sensor data.

5.4.3.2 Batch data processing

Batch data ingestion holds a key role in the overall process of data integration. Batch data processing happens at the same stage similar to real-time data processing, which is before the data storage. The batch data needs to be collected and processed at regular intervals, providing a consistent flow of data storage and consumption. Lack of proper batch data ingestion processes, there could be delays in data collection and processing, which could lead to outdated or inaccurate data being used for decision-making and analysis.

5.4.3.2.1 Data types for batch data processing

Most of the data sources that a smart city platform may encounter are in the form of batch data at regular intervals or following key events. The following are the most common types of data in the context of smart cities that the smart city data integration platform should be ready to cater the following.

a) Structured data

Data that is organised in a predefined manner, such as data in relational databases or CSV files.

b) Semi-structured data

This is data that does not conform to the formal structure of data models but contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data. Examples include XML, JSON, or NoSQL databases.

c) Unstructured data

This is data that does not have a pre-defined model or is not organised in a pre-defined manner. Examples include CCTV footage, text files, social media posts, emails, etc.

5.4.3.2.2 Batch data processing components

As the batch data is ingested, several basic data operations by different components shall be undertaken to ensure data consistency and reliability. For batch data, these operations undertaken by these components shall be highly systematic, scalable, and sustainable for data errors or temporary data losses.

The batch data processing components requires the data integration platform to be able to undertake the following functions and specifications.

The most basic of these operations are as follows.

a) Data cleaning

Once the data is ingested, the data shall go through a data cleaning process to ensure that the quality of data is maintained. Data cleaning should involve at the least removing duplicates, correcting errors, and dealing with missing values, wherever applicable.

b) Data transformation

The cleaned data might need to be transformed to a specific required format. The platform shall deploy the tools to transform the data suitable for further processing. This could involve operations like normalising data (bringing all data to a common scale), encoding categorical data (converting categories into numerical values), feature scaling (changing the range of values), and more. The transformed data is then ready for further processing.

c) Data aggregation

The platform shall deploy the required tools and capabilities to be able to process data aggregations. Data aggregation involves summarising and grouping data. This could involve operations like calculating averages, sums, counts, etc., over each group of data.

d) Data storage

The platform can store the data in suitable data stores. This could involve storing the data in a data lake, a data warehouse, or another type of storage system. The stored data can then be accessed and used for further analysis, processing, reporting, or decision-making.

e) Manage data processing pipelines

It is important to continuously monitor the performance of the batch data processing system and optimise it as necessary. The platform needs to deploy suitable tools and capabilities to monitor and optimise the data ingestion pipelines. This could involve tuning the parameters of the data processing operations, scaling up the system to handle larger volumes of data, or making other adjustments to improve the efficiency and effectiveness of the system.

5.4.3.3 Data enrichment

Data enrichment is the process of enabling raw or processed data to improve its quality and usefulness by applying domain-specific advanced processing. The data enrichment process applies to real-time data processing, as well as batch data processing. The goal is to enable more sophisticated analytics and decision-making. Among other, in the smart city context, major areas to enrich data includes data enrichment by Geographic Information System (GIS) data processing and applying Artificial Intelligence processes.

5.4.4 Data storage

The data storage for a smart city data centre, primarily categories into two major areas, data lakes and data warehouses, below are description of these two categories.

5.4.4.1 Data lakes

Data lakes serves as a central repository for all the data collected by the smart city data integration platform, allowing for flexible and versatile analysis. It contains the native data format to provide maximum flexibility, governance, and processing while optimising the storage. The data lakes are commonly supported by big data systems to fulfil the requirements expected from the data lakes.

5.4.4.1.1 Data Lake Capabilities

In general, any data storage medium may serve as the data lake, however, some certain capabilities and specifications are required for a storage medium to act as a data lake in the smart city integration platform.

These specifications include the following.

a) Scalable storage

A data lake can scale to store petabytes of data, making it suitable for storing the vast amounts of data generated in a smart city. A large amount of data access and query is supported by parallel processing to support data management.

b) Data variety

A data lake can store a wide variety of data, from structured data to unstructured data, making it versatile for different types of analyses.

c) Real-time ingestion

A data lake shall be able to ingest data in real-time, allowing for up-to-date analyses.

5.4.4.1.2 Data lake components

The data storage in the data lakes takes place after the pre-processing of the data and before any data analysis is done. The data is stored in data lakes in native format. However, the data lakes are required to be able to undertake the following functions and specifications.

a) Data ingestion integration

The data lake should be able to connect and integrate with the data ingestion processes and pipelines. The data is ingested in its native format without the need for initial processing or transformation.

b) Data storage

The data lake should be able to store data in a scalable format. The storage system should support both horizontal scaling as well as vertical scaling. The data being stored remains in its native format.

c) Data cataloguing

For the platform to make the data in the data lake findable and usable, it is important to catalogue the data. This involves creating metadata for the data, which describes the data and provides information about its source, format, and other attributes. This metadata is then used to organise the data within the data lake.

d) Store and handle large amounts of data

Data lakes are designed to handle large volumes of data, making them suitable for scenarios where data from multiple sources may be stored, scale the data storage, and can be accessed by multiple processes engaged in analysis, sharing, artificial analysis, etc.

e) Data format flexibility

Data lakes store data in its native format, which allows for maximum flexibility when it comes to processing and analysis.

f) Four-V’s characteristics

Data lakes are compatible and able to manage the Four-V's of data management, i.e. data volume, data velocity, data veracity and data variety.

g) Data security and governance

The platform shall ensure the security of the data in the data lake. This involves implementing security measures like access controls, encryption, and auditing. Additionally, data governance policies should be in place to manage the quality, consistency, usability, and security of the data.

5.4.4.2 Data warehouses

A data warehouse is a large, centralised repository of data that is used to guide business decisions. Unlike a data lake, a data warehouse stores data in a structured format, typically organised in a schema that has been designed for easy reporting and analysis. Data warehouses are typically used to store historical data and enable complex queries and analyses. There may be more than one data warehouse in a smart city data integration platform. The data warehouses are generally deployed as structured or relational databases.

5.4.4.2.1 Data warehouse capabilities

The data warehouses are deployed as fast data access mediums that support quick data querying and processing. The data warehouses, in a smart city data integration platform context, shall deploy the following capabilities and specifications.

a) Support structured storage

A data warehouse shall store data in a structured format, making it suitable for structured data and enabling complex queries and analysis.

b) Historical data

A data warehouse needs to store historical data, providing a long-term view of data over time.

c) Support common querying languages

The data warehouse shall support complex querying, aggregation, sorting, and grouping of the data within the context of the data query.

d) Integration with tools

A data warehouse should be able to integrate with various tools, such as Business Intelligence (BI) or Artificial Intelligence (AI), allowing for easy reporting and analysis.

e) Scalable storage

A data warehouse can scale to store large volumes of data, making it suitable for storing the vast amounts of data generated in a smart city.

f) Structured data

A data warehouse stores data in a structured format, making it suitable for structured and semi-structured data.

g) Batch loading

A data warehouse should be able to load data in batches at regular intervals.

h) ETL and ELT Data Processing Support

The data warehouses are the sources of information for the analysis and decision-making processes. These processes require the latest data to be available and processed. The platform shall be able to update the data and provide a single point of truth. The data update and processing are facilitated by the processes Extract, Transform and Load (ETL) or Extract, Load and Transform (ELT).

5.4.4.2.2 Data warehouse components

The data warehouse is required to perform certain key functions, and to support these functions, the platform must deploy the following primary components.

a) Data lake integration

This is the process of querying and importing data from various sources. In the context of a smart city data integration platform, this could involve collecting data from IoT devices, sensors, databases, and more.

b) Data cleaning and transformation

Before the data is loaded into the data warehouse, it is cleaned and transformed into a format that fits the schema of the data warehouse. This process is known as ETL. It involves removing duplicates, handling missing values, transforming data into the required format, and more.

c) Data loading

Once the data is cleaned and transformed, it is loaded into the Data Warehouse. The data is typically loaded in batches at regular intervals.

d) Data management

After the data is loaded, it needs to be managed effectively. This involves ensuring data quality, implementing data security measures, and managing metadata.

e) Scheduled processing

The ETL or ELT pipelines can run on schedule and provide a mechanism to run the pipelines in a predefined manner at predefined intervals.

f) Safe failover

The ETL or ELT processes shall allow the safe failover mechanism, where the pipeline processing fails due to inconsistent, inaccurate, or invalid data, the safe failover process shall log the inconsistencies and not break the pipeline processing for the rest of the data.

5.4.5 Data dissemination

Data dissemination or data serving is the process of making processed and stored data available to end-users or stakeholders in a useful and accessible format. This involves serving data through various methods such as data analysis tools, API connectivity, data discovery modules, and dataset sharing.

The smart city data integration platform shall deploy the data serving components that can integrate with the data processing and storage components. For example, the data analysis tools might pull data

from the data warehouse, analyse it, and then present the results to the user. The API would provide a way for third-party applications to access the data in the data warehouse. The data discovery modules would provide an interface for users to explore and interact with the data. The dataset sharing component would provide mechanisms for sharing data with other users or organisations.

5.4.5.1 Data dissemination capabilities

The data dissemination component in a smart city data integration platform should be able to demonstrate the following specifications.

a) Handling large volumes of data

The smart city data integration platform needs to handle large volumes of data that are continuously being generated from various sources such as IoT devices, sensors, social media feeds, and more. This requires robust and scalable data-serving components that can efficiently process and serve this data to end users. The platform needs to handle peak data loads without performance degradation, ensuring that users can access the data they need when they need it.

b) Real-time or near-real-time data access

In many smart city applications, stakeholders need access to real-time or near-real-time data. For example, traffic management, CCTV analysis, etc. Therefore, the data-serving components of the platform should be designed to provide real-time or near-real-time data access. This involves using technologies like stream processing, in-memory databases, and real-time analytics.

c) Data security and privacy

Data security and privacy are critical considerations in the smart city data integration platform. The platform should implement robust security measures to protect the data from unauthorised access and breaches. This involves using encryption, secure APIs, and secure data-sharing mechanisms. Additionally, the platform should comply with data privacy regulations, ensuring that personal data is handled appropriately and that users have control over their data.

5.4.5.2 Data dissemination methods

5.4.5.2.1 Data analysis

Data dissemination via data analysis would be one of the major ways to provide insights to the end users and help them make data-driven decisions. The data analysis component requires the following capabilities.

a) Data processing capabilities

The platform is required to be able to handle both structured and unstructured data, and be capable of processing large volumes of data quickly and efficiently.

b) Analytical tools

The platform should provide a wide range of analytical tools and techniques, such as statistical analysis, predictive modelling, machine learning algorithms, and data mining capabilities. These tools should be flexible and customisable to meet the specific needs of the analysis.

c) Visualisation features

Data visualisation is a crucial aspect of data analysis. The platform should provide robust visualisation tools that allow users to create charts, graphs, and other visual representations of data to better understand patterns, trends, and correlations.

d) Scalability

The platform needs to be scalable to handle increasing data volumes and complexity. As the amount of data grows, the platform should be able to maintain performance and provide consistent results.

e) User-friendly interface

The platform should have an intuitive, user-friendly interface that makes it easy for users of all skill levels to navigate and use. This includes clear navigation, easy-to-use tools, and a clean, uncluttered design.

f) Collaboration features

The platform shall support collaboration, allowing multiple users to work on the same datasets or projects simultaneously. This could include features like shared workspaces, version control, and commenting.

g) Security and privacy

The platform shall deploy robust security measures in place to protect sensitive data. This includes user authentication, data encryption, and access controls. Additionally, the platform should comply with relevant data privacy regulations.

h) Integration capabilities

The platform should be able to integrate with other systems and platforms, such as databases, data warehouses, and BI tools, to allow for seamless data flow and enhance the platform's versatility.

i) Customisability

The platform should be customisable to meet the specific needs of the smart city. This could include custom workflows, custom reporting, and the ability to add or modify features.

5.4.5.2.2 Application Programming Interface (API) connectivity

APIs are the most common way to share data between systems. The APIs bring flexibility to data transfer with data transformation, schema definitions, and API discovery and documentation systems. To reduce the risk of APIs being unable to be published or consumed among systems, a smart city data integration platform shall standardise the API components and deploy the following capabilities.

a) Standardised data formats

APIs should support standardised data formats such as JSON or XML. This ensures that the data can be easily consumed by a wide variety of applications and platforms. Also, the API documentation shall clearly define the content type and return data type from any API.

b) Security

The APIs shall implement robust security measures to protect the data they expose. This could include authentication (ensuring that only authenticate users can access the data), encryption (protecting data in transit), and rate limiting (preventing abuse of the API).

c) Versioning

The API component in the smart city data integration platform should support the versioning of APIs to allow for changes and improvements over time without breaking existing integrations. This can be achieved by including a version number in the API's Uniform Resource Locator (URL) or request headers.

d) Documentation

All the APIs exposed by the data integration platform shall be well-documented, with clear instructions on how to use the API, what data is available, and what each API endpoint does. This includes the headers of the API, parameters required, and return data format and schema.

e) Error handling

The APIs should provide clear, useful error messages when something goes wrong. This helps developers to troubleshoot issues and understand how to use the API correctly.

f) Scalability

The platform should be designed to handle a large number of requests and scale as demand increases. This might involve techniques like load balancing or caching.

g) Performance

The API components should be optimised for performance, with fast response times, e.g. few kilobytes of payload returned should be under two seconds, etc. This might involve techniques like efficient data querying, compression, asynchronous processing for long running queries, and pagination.

h) Interoperability

The APIs should be designed with interoperability in mind, ensuring they can work effectively with a range of different systems, technologies, and standards.

5.4.5.2.3 Data discovery module

Data discovery is a critical component of the smart city data integration platform, allowing users to find and understand the data they need. The data discovery module in the smart city data integration platform shall provide users with easy access to the data they need, in a way that is secure and easy to understand. The data discovery module shall adhere to the following specifications.

a) Search functionality

The data discovery component needs to provide robust search functionality, allowing users to easily find the related data. This could include keyword search, filtering options, and advanced search capabilities.

b) Data cataloguing

The platform shall provide a data catalogue that organises and categorises data assets, making it easier for users to find and understand the data. The catalogue should include metadata about each data asset, such as its source, format, and when it was last updated, etc.

c) Data profiling

Data profiling involves examining the data and collecting statistics and information about that data. The data discovery component should provide data profiling capabilities, giving users insights into the data's quality, structure, and content.

d) Data lineage

Data lineage provides information about the data's origins and where it moves over time. This can help users understand how data is transformed as it moves through different processes and systems.

e) Data preview

The intended users should be able to preview data before they decide to use it. This could involve displaying a sample of the data or providing a summary of the data's contents.

f) User-friendly interface

The data discovery component should have an intuitive, user-friendly interface. Users should be able to easily navigate the data, with clear labels and instructions.

g) Integration with other components

The data discovery component should be well-integrated with the rest of the data integration platform.

h) Security and access controls

Access to data needs to be controlled based on user roles and permissions. Sensitive data should be protected, and users should only be able to access the data they are authorised to see.

5.4.5.2.4 Dataset sharing

Dataset sharing is an essential part of the smart city data integration platform, enabling the platform and admins to share the data with other platforms, systems, or devices, in a controlled and secure manner. The dataset sharing module shall adhere to the following specifications.

a) Access control

The platform needs to deploy robust access control mechanisms to ensure that only authorised users, or systems can access shared datasets. This could include user authentication, role-based access control, and the ability to set permissions for individual datasets.

b) Data formats

The platform shall support a variety of data formats for sharing, such as CSV, JSON, XML, etc. This ensures that the shared data can be easily consumed by a wide variety of applications and platforms.

c) Version control

The platform should provide version control for shared datasets. This allows users to track changes to the data over time and ensures that users are always working with the most up-to-date version of the data.

d) Data anonymisation

If the shared data contains sensitive information, the platform needs to provide data anonymisation features to protect privacy. This could involve techniques like data masking, pseudonymisation, or differential privacy.

e) Metadata sharing

Along with the data itself, the platform should also allow for the sharing of metadata. This helps users understand the context of the data, including its source, when it was last updated, and any transformations it has undergone.

f) Security

The platform shall implement robust security measures to protect the data being shared. This includes encryption of data in transit and at rest and regular security audits.

6. General specifications

Besides the specifications and characteristics for each component of the data integration platform, there are general specifications that the platform shall comply with and deploy these features for seamless, secure, and efficient operations. These features are as follows.

a) Interoperability

The smart city data integration platform needs to be designed to support interoperability standards and protocols, ensuring that it can effectively communicate with a wide range of external systems and applications.

b) Data quality management

The platform should have mechanisms in place to ensure the quality of the data being disseminated. This could involve data validation checks, anomaly detection, and error-handling mechanisms. Ensuring data quality is crucial for maintaining the trust of the users of the platform.

c) Scalability

The platform should be scalable, and able to handle increasing volumes of data as the smart city system grows and evolves. This involves not only the ability to store large amounts of data but also the ability to process and serve this data efficiently.

d) Flexibility

The platform should be flexible, and able to adapt to changing data sources, formats, and user needs. This could involve supporting a wide range of data formats, providing flexible data serving options, and allowing for customisation by users.

e) Data governance

Data governance involves the overall management of the availability, usability, integrity, and security of the data employed in an enterprise. A sound data governance program includes a governing body or council, a defined set of procedures, and a plan to execute those procedures.

f) Performance

The platform should provide high performance, ensuring that users can access and interact with the data quickly and smoothly. This involves optimising data processing, storage, and serving mechanisms to minimise latency and maximise throughput.

g) Reliability

The platform should be reliable, providing consistent and accurate data to users. This involves ensuring the integrity of the data, as well as the availability of the platform and its services.

Acknowledgements

Members of the Internet of Things (IoT) and Smart Sustainable Cities Working Group

Dr Gopinath Rao Sinniah (Chair)	Favoriot Sdn Bhd
Mr Mohd Amin Mohd Din (Vice Chair)	Maxis Broadband Sdn Bhd
Dr Teng Kah Hou (Secretary)	UCSI Education Sdn Bhd
Mr Abbas Ali (Draft Lead)	Kiwitech Sdn Bhd
Mr Mohamad Norzamir Mat Taib (Secretariat)	Malaysian Technical Standards Forum Bhd
Mr Low Kien Yap	CelcomDigi Berhad
Mr Ang Kah Heng/ Ms Athirah Tan Abdullah	Cyberview Sdn Bhd
Mr Meng Keen Wai/ Mr Wong Chup Woh	Maxis Broadband Sdn Bhd
Ms Norhanisah Mohd Basri	SIRIM Berhad
Professor Ts Dr Lau Sian Lun	Sunway University College Sdn Bhd
Mr Mohd Zakir Hussin Baharuddin	TM Technology Services Sdn Bhd
Dr Mohd Yamani Idna Idris	Universiti Malaya
Professor Dr Borhanuddin Mohd Ali	Universiti Putra Malaysia
Dr Hafizal Mohamad	Universiti Sains Islam Malaysia
Associate Professor Ir Dr Yusnani Mohd Yussoff	Universiti Teknologi MARA

By invitation

Mr Surentharan Ramadas	Bahagian Kerajaan Tempatan Pulau Pinang
Ms Nur Amilin Mohd Khazani/ Ts Md Farabi Yussoff Md Yussoff	Jabatan Perancangan Bandar dan Desa
Mr Mohamed Shajahan Mohamed Iqbal	Three-OPP (M) Sdn Bhd