ASEC 2024 Conference

The 5th International Electronic Conference on Applied Sciences

04–06 December 2024 | Online

MDPI

Parametric Middleware Routing and Management Services Platform Model for Smart Cities

Volodymyr Nazarenko*, Oksana Zazymko, Andrii Martyn

National University of life and environmental sciences of Ukraine

Kyiv, Ukraine

* volodnz@nubip.edu.ua

INTRODUCTION & AIM

In the 21st century technology has become important part of everyday life and it is used to enhance, assist and improve quality of life. Smart cities is without a doubt can be classified as vital technologies advancement and innovation. A smart city platform leverages technology and data to enhance the quality of life for residents by improving infrastructure, services, and overall urban management. There are many applications and use cases for smart city services - open data available to the public and decision-making support; applications that provide users with real-time data on various city services, transportation, renting and payment; smart girds, smart recycling and waste management, traffic control and city-wide security surveillance among many other services.

Typical Smart City System is made of a number of software and hardware components that communicate and are managed together to collect, transmit, process, and analyze data. This system can be further divided into several layers – end-points devices, such as sensors and actuators; complex IoT devices; network, security and communication devices; cloud and data processing clusters; software services and applications. A large-scale real-time Internet of Things (IoT) platform is a complex multilayered software system designed to manage, process, and analyze data from a vast number of interconnected devices and sensors in real-time. Such platforms can support a wide range of IoT applications, from smart cities and industrial automation to healthcare and logistics. Routing is one of key determinants and important tool for effective and resource efficient smart city service management.

Software-level routing is crucial in the management of complex multilayered software systems, as it solves and assists with the internal-level process communications, acts as part of the system scaling and optimization tasks, besides the software-level routing enables complex systems to be adaptable to new deployment configurations and services orchestration. Traditional approaches to software design and development do not take into account both the high-level and low-level management of complex services, which include IoT devices, real-time applications, and AI-related processing frameworks. Some of the most important components of software system management are internal task routing and services management.

METHOD

Smart city system is developed and implement for real world cities, using real world data. For example, the Kyiv city is the largest in terms of population and total area city in Ukraine. The Kyiv city data had been used as a foundation of the proposed smart city system model. There are several exiting smart city solutions that are used by the local population and administration. Main data is presented in Figure 1, which includes total number of smart city application users, types and quantity of connected smart city hardware (sensors etc.) devices, most commonly used and available network communication devices and other important IoT components.

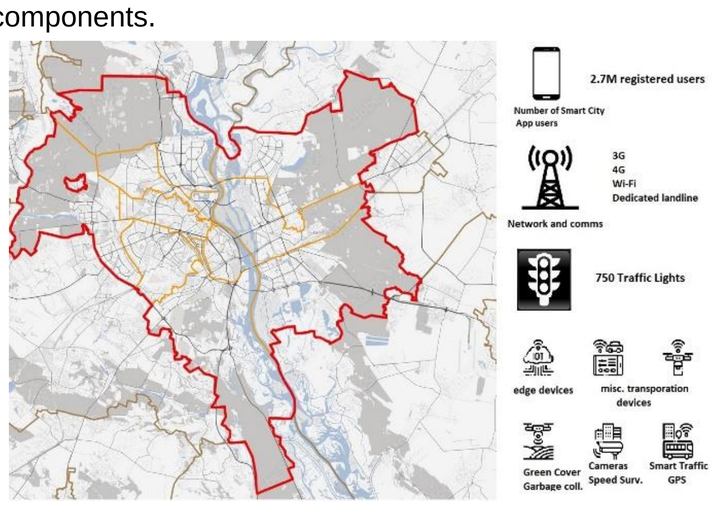

Figure 1. Kyiv city Map and Smart City Profile

A three-layered software architecture model (three-tier architecture) is most common software development model. It is a well-established pattern for organizing software systems, which include smart cities system. The detailed breakdown of the adapted three-layered architecture for the smart city use case is outlined in Table 1: Application layer, Middleware layer and Core-services (Data organizational) layer. Each layer has specific responsibilities and interacts with the other layers with the help of routing and management core systems.

Table 1. Smart city core service layers

Smart City Service layers	Systems Applications	Data structures Important parameters
Application layer	End-user app System manager app City governance Services	User Profile Access and permission Common UI data Number of active connections
Middleware layer	End-point sub-service Smart Communication Prioritization and Routing	Digital and analog data Network packages Tasks and Commands Real time cont. Network traffic and data packets
Core Services layer	Events system manager Data processing and internal bus messaging Core System Service	Table and objects System data Tasks and Commands API and sequential System status

RESULTS & DISCUSSION

The proposed Smart City cloud deployment platform can be conceptualized as a three-layer vertical cloud-service deployment model (Figure 2). This model divides the platform into three separate logical layers, each responsible for specific aspects of the system, from infrastructure to service delivery. On top there is applications suit; middle section consists of main processing, connectors and routing sub-services, while the bottom most layer is responsible for main mathematical, data logical and management operations, as well as smart/AI data processing and prediction applications.

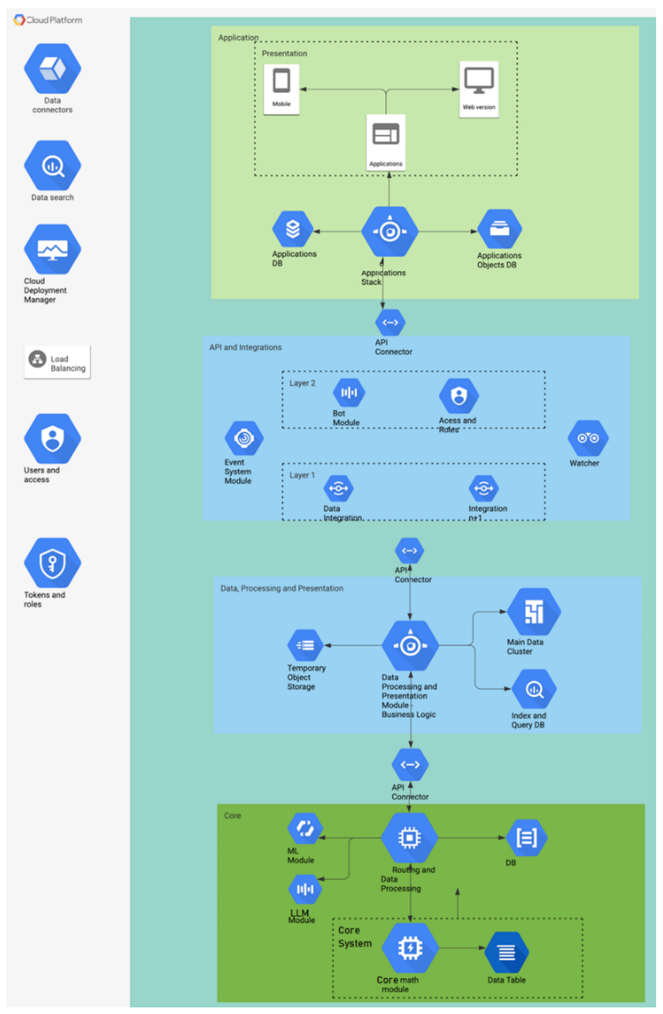

Figure 2. Three-layered Smart City cloud platform sub-services model

Such system can be developed and deployed in any exiting cloud platform provider or under the novel platform, made for the specific smart city systems instance. This cloud deployment model includes many of the critical and typical software components, such as: Virtual machines support; load balancing and security authentication; service hosting and database deployment; analytics and predictions services; smart city specific sub-services – traffic management, energy management, public health and transportation, general smart-city application and others. Being large scale systems, it requires service/system/platform administrator personal to manage and control service states. In case of requirements, this system can connect and communicate with external services via API or any other common software and networking communication protocols. The key requirements in such case, would be security and data integrity.

A real-time events and processing software system involves creating a framework that enable handling of continuous streams of data in real-time or close to real-time. Service real-time routing and communication under the smart events and processing systems happens as a flow of parameters weighted messages. The generalized routing algorithm is illustrated in Figure 3. This algorithm consists of 4 steps, starting from service request, followed by request initialization, requested sub-service enabling internal transactions and communications process, as well as parameters and message delivery to the requester (in case of successful and field event).

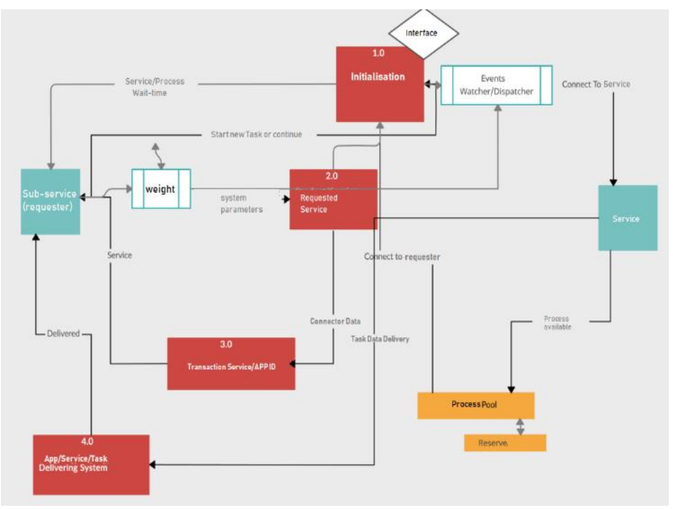

Figure 3. Real-time routing algorithm

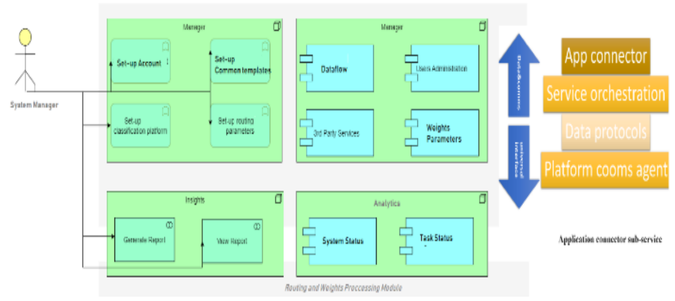

Figure 4. Services Management software model

Table 2. Core management system services

(micro-) Service	Module Service control parameters	Service functionality
Events manager system	Events Watcher System Messages Communications and connectors Event management System Events Dispatcher System	Connect microservices Prioritization Authentication Integration Event processing Traffic routing
Data and information processing system	Request processing Weights processing User/App data and processing Sub-service routing System output/reply	Rules and message classification Data extraction and pre-processing Query processing and prioritization
Data and logic system	Protocol Storage Get/Retrieve Data Connection	Connection Connector Communication Database read/write

CONCLUSION

A large-scale real-time Smart City system is multilayered, complex and scalable system. It must handle various set of tasks and components, ranging from IoT devices and platforms, provide timely real-time responses and manage data. Smart city platform is based on integration of various number of components, each individual in turn consists of its own set of software and hardware services, components. Tasks of sub-services and modules management is as crucial as complex in nature. As it was illustrated this role can be filled in by the special routing and management services, each being platform and deployment agonistic. While routing service is designed following standard protocols, APIs, and middleware layer service; sub-service management systems is low-layer data/process processing and computations first system