Real-Time HazMat Environmental Information System: A micro-service based architecture

Ghyzlane Cherradi, Adil El Bouziri, Azedine Boulmakoul, Karine Zeitouni

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 982–987

Procedia Computer Science

www.elsevier.com/locate/procedia

The 7th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS2017)

Real-Time HazMat Environmental Information System: A micro-service based architecture

Ghyzlane CHERRADI[a]*, Adil EL BOUZIRI[a], Azedine BOULMAKOUL[a], Karine ZEITOUNI[b]

[a]Innovative Open Systems Research Group (IOS), LIM Laboratory
Computer Science Dept., Faculty of Sciences and Technology of Mohammedia (FSTM),
University Hassan II of Casablanca, Morocco
[b]DAVID Université de Versailles Saint-Quentin-en-Yvelines, France

Abstract

The dangerous goods are important to businesses and the life of a city. Thousands of tons of oil, toxic, chemical, corrosive, flammable and radioactive materials are transported each day. However, an accident involving hazardous materials could entail serious consequences for road users, infrastructure, and the environment. In order to build a planning aid system to reduce the risks of transporting hazardous, we adopt a microservices-based architecture on a cloud environment. This type of architecture consists of a set of loosely coupled and independently deployable services for more scalable applications. Due to distributed nature of microservices, the system will be developed as a suite of small services aligned with risk management process; each service will be running in its own logical machine or container technology such as Docker.

Keywords: *Transportation of dangerous goods, environmental information system, microservices based architecture, data collector;*

* Corresponding author. Tel.: 00 + 212 + 6 33 78 03 07.
E-mail address: cherradighyzlane@gmail.com

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.457

1. Introduction

The dangerous goods are used in several industrial sectors, including all branches of human activities. However, hazmat accidents can pose a significant security threat. Such accidents may occur in industrial units where hazardous substances are produced, handled and stored; or during their transportation[1]. Since the transport routes of these substances almost pass through urban areas, the increasing risks, which are created, affect all resident and non-resident populations close to these industrial units. To ensure the safe and reliable transportation of hazardous materials, there is a real need to detect all threat scenarios. Thus, we propose in this work a real-time environmental information system, which can track in real-time hazmat vehicles. It provides up-to-date information, knowledge, and services that help decision makers and urban planners to supervise hazmat transportation and better reduce the risk of moving hazardous materials. In order to enhance overall performances, the architecture of the proposed system is based on a microservice concept to develop and deploy applications, which are composed of independent, modular and self-contained units. By definition, microservices are small software components, specialized in one task and work together to achieve a higher-level task[2]. At the same time, microservices encourage a level of fine granularity that supports evolving parts of a system in relatively isolation. Since each microservice is autonomous and independent, it is easy to monitor and replace a faulty service without affecting any other. In this paper, we focus on microservices and overall architecture. The latter provides cooperating microservices, in which each microservice runs a unique process and communicates through a well-defined, lightweight mechanism (API orchestrator) to serve a functionality.

2. System functionalities

As previously mentioned, the global purpose of the proposed real-time environmental information system is to provide a management platform of dangerous goods transport risk. Therefore, the functionalities provided by this system should be aligned with risk management process. The latter usually starts with the step that focuses attention on specific scenarios, in order to identify risk sources since hazards are recognized through trend monitoring and investigation of safety occurrences[4,5]. The incidents and accidents are clear indicators and should be therefore investigated to determine the hazards that play the main role in a threat[6]. Hence, we use a historical data with a data collection module[21]. The historical data is based on accident and incident investigation and analysis in order to discover all facts pertinent of a past incident or accident; and thus, identify opportunities for improvements which help to avoid future, similar accidents[13]. Each hazard that is identified must be evaluated and classified[4,5,7]. This process requires a documentation and regulation database, which facilitates the recovery of characteristics and regulations of these dangerous goods. Moreover, there is a need to use a GIS technology to manage, maintain and retrieve geographical data. The Geographic information systems (GIS) are widely used in risk analysis in the transportation of hazardous materials. The GIS models help to quantify and provide better accuracy by analyzing the risk related to various parameters[8]. Furthermore, to be able to respond effectively in case of accidents involving hazardous materials, we have to develop different scenarios, in order to anticipate and identify the affected area, by the use of a monitoring module based on GPS and GIS technology in combination with atmospheric dispersion models[22]. The last step of risk management process is to analyze simulation results for the sake of mitigation plan; that is to identify the changes in terms of risk assessment or risk ranking, and the mitigation possibilities in terms of actions[4,5]

3. Software Architecture

By definition, the main goal of a software architecture is to identify the requirements that affect the structure of the application. It is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. In this section, we will describe the different technical and operational requirements that must be offered by our system; afterwards we will discuss and justify our proposed architecture.

3.1. Technical and operational requirements

Building a real-time environmental system of guaranteed safety of hazmat transport, raises challenging technological problems. Actual techniques and technology are of a little help to provide a global framework. The development of such system must take into account a variety of requirements:

- Integration of a wide range of information on an affected area, for increasing the accuracy of scenarios in order to produce an effective planning.
- The system should be accessible from a number of different locations or even device types,
- Fast evolving environment with rich dynamics,
- The system have to offer a user-friendly interfaces, higher reliability and better performance,
- The failure of a component should not affect system performance, security, safety and availability,
- Being open for future extensions,
- Allow interaction between people and environment by responding to various information demands and providing support through various outputs.

These requirements are achieved in two ways: i) software architecture choice should give the ability to start with the minimum bare. Then we can add more components to expand the system as needed, without affecting other components. ii) The technical architecture ensures a quality system design, which provides good system performance, an effective human-machine interface, optimal operational cost, and flexibility for future changes.

3.2. Microservice based architecture

Driven by the idea that the combination and integration of efficient and scalable components help to build a scalable, reliable and resilient system, it is necessary to use an open architecture that divides large software project into loosely coupled components, which communicate with each other through simple APIs.

A microservices architecture is an approach that divides the solution into a microservice component parts and treats them as separate development efforts; which increase the speed of development and go down the cost of making changes. The microservice component is structured to achieve exactly one identifiable purpose[13,14]. This type of architecture has many features that focus on various aspects. Among its features:

- Faster and simpler deployment and rollback with smaller services: Taking advantage of the divide and conquer paradigm in software delivery and maintenance.
- Eliminate long-term commitment to single stack technology: selecting the right tool, language and technology per service, without having to conform to a homogeneous environment being dictated by shared infrastructure.
- Guarantee evolutionary design, by giving the possibility to incremental change in an architecture as a first principle.

An important feature of the microservice architecture is the isolated and individual deployment of each service. It is important that every instance of each microservice would have complete autonomy over the environment. The Docker technology is improved on the benefits of virtualization by avoiding the full cost of an operating system and sharing some of host services that would be unnecessarily duplicated in each virtual machine. In our proposed architecture, we use Docker containers as units of deployment for microservices.

3.3. Overall Open Architecture

2. The proposed architecture provides cooperating microservices, in which each microservice runs a unique process and communicates through a well-defined, lightweight mechanism (API orchestrator) to serve a business goal. In this architecture, each microservice includes its own persistence; it can be independently deployed, upgraded, and replaced. The distributed and modular nature of microservice architecture needs that every instance of each microservice would have complete autonomy over the environment[3]. Besides, the container technology such as Docker can be useful in microservice deployment. The Docker technology gives the possibility to embedded microservice in a cloud-based, continuously integrated environment.

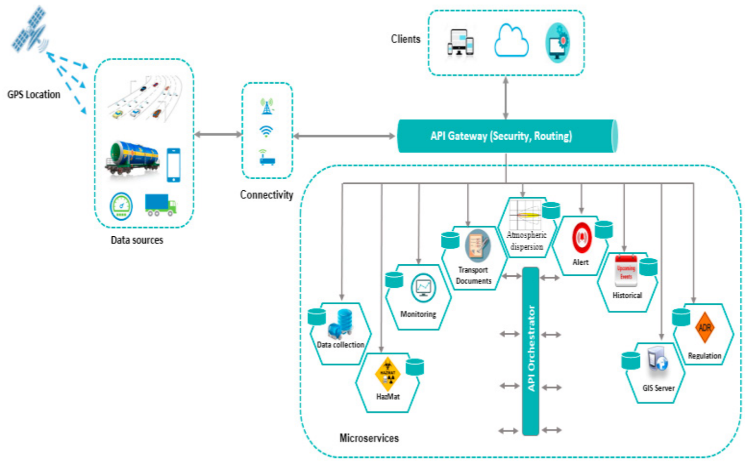

Fig. 1. Generic microservices-based architecture.

Figure. 1 illustrates the proposed architecture of our environmental system. Using sensors and GPS technology, the real-world data (location, vehicle information, hazmat information, real-time environmental information, etc...) are transmitted to the data collection microservice, his goal is to store and transmit data on request to other microservices. The monitoring microservice allows real spatio-temporal data to be captured from data collection microservice and simulate the trajectories of the different connected vehicles. To show the current location on the map, a GIS microservice is used to integrate spatial data set in order to generate map and GIS tools to perform spatial operations[8]. All transport of goods, regulated by ADR (European Agreement concerning the International Carriage of Dangerous Goods by Road) must be accompanied by documentation[12]. Therefore, to facilitate the viewing of transport documents, we call the transport documents microservice, which will make it possible to download the various documents required from the registration number of the vehicle. Then, we use the regulation microservice for provide extensive regulatory information. This can increase the level of speed, accuracy and reliability of road transport regulatory authority's decision. The hazmat information, such as product identification, the physical and chemical properties, the nature of danger etc, will be provided by the hazmat microservice. The latter takes the ONU number as filtering attribute. This hazmat information represents the input data for atmospheric dispersion microservice, whose the key features is to generate a special variety of output scenarios and evaluate the different types of risk. To prevent Hazmat transportation risk, the manager needs a system that could give him an early warning when it detects an intrusion. This is guaranteed by the use of alert generator microservice. This microservice also allows the user to be alerted via his phone in the case of a nearby accident involving hazardous materials. The following sections detail the technical architecture of some microservices such as data collection, Monitoring, and GIS microservices.

4. Data collection microservice

Currently, we are able to share data from connected devices on the internet due to IoT technologies. In this section, we propose a data collection microservice based on MQTT (Message Queue Telemetry Transport) protocol in order to build a rapid scalable IoT platform, which can collect and transport data.

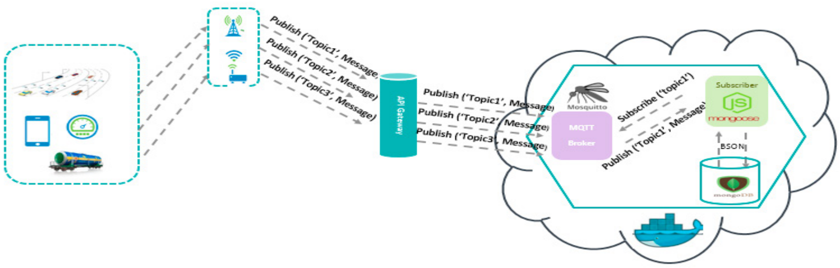

Fig. 2. Data collection microservice architecture

The IoT devices represent the sensor layer in an IoT infrastructure, which is responsible for obtaining the various types of static/dynamic information of the real world through various types of sensors; and to share with Internet access[16], through the various connectivity such as mobile communication networks, wireless local area network (WiFi) and satellite network, etc. The MQTT protocol transports device data and communicates it to the cloud IoT broker server. It is a publish /subscribe messaging transport, that is an adequate protocol to connect small device to constrained networks[17]. It is a many to many communication protocol for passing messages between multiple clients through a central broker, and it uses less power to maintain an open connection, to receive and send messages to them. The Cloud IoT broker server represents the service layer in an IoT infrastructure. It consists of an MQTT broker (Mosquitto broker), MQTT client (Node server), and a data server (Mongodb). Furthermore, embedded into a container docker and based on cloud computing technology, our cloud-IoT broker server can offer two basic services as infrastructures for storage and to deliver information to other microservices[17, 18].

5. Monitoring and GIS microservice

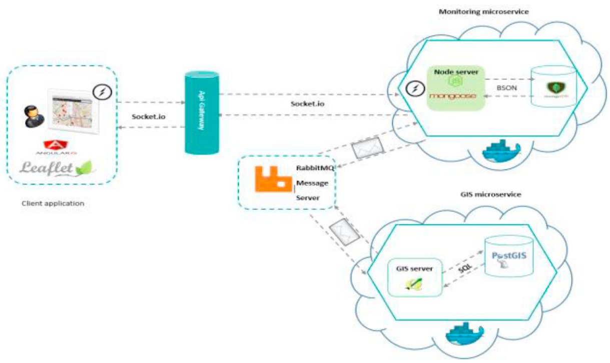

Fig. 3. Monitoring and GIS microservice architecture

First, the monitoring microservice socket is listening to connection, then from the client's side, the number and the vehicle to truck is chosen. After that, every some seconds the data collection microservice sends vehicle location to the monitoring microservice, which is connected to a database for storing the user information. Moreover, in order to produce a map that combines the location information of vehicle with other data that shows road characteristics displayed in "layers", we use a GIS microservice based on Qgis server. The later also allows the read and the store of a geographical data. For spatial data storage, we use the PostgreSQL and PostGIS extension. The client application was developed using leaflet API to building a mapping javascript application combined with AngularJs framework. This monitoring microservice can be useful for seeing which dangerous materials exist in a given territory and to have a real-time knowledge of the movements of transported materials that can enable possible accident[20].

6. Conclusion and future works

In this paper, we propose a global microservice based architecture, to have a real-time information system that has highly scalable applications on cloud environment. This can help managers to produce effective plans to reduce the risks of transporting hazardous materials in urban areas. The microservice architecture is an architectural style that provides a number of benefits by adopting a divide and conquer approach to software design and deployment. This style of architecture allows our system to be a distributed business-driven system, by exposing their functionalities as business microservices separately deployed into a cloud environment deployed within our work by docker technology. Our future work will be aimed at implementing the overall proposed architecture, and obtain an experimental results and quantifiable comparison between the traditional data collection and the proposed platform.

Acknowledgement

This work was partially funded by the CNRST project in the priority areas of scientific research and technological development "Spatio-temporal data warehouse and strategic transport of dangerous goods»

References

1. Belotti J, Transport international de marchandises, Vuibert 2015.
2. David G, Developing Microservices with Node.js. Packt Publishing; 1ed, April 26, 2016.
3. Babak M, Microservice Architecture Building microservices with JBoss EAP 7. Version 1.0, June 2016.
4. Chan H. K., Wang X. Fuzzy Hierarchical Model for Risk Assessment, London: Springer, vol. 10, pp. 978-979, 2013.
5. Hyatt N Guidelines for process hazards analysis (PHA, HAZOP), hazards identification, and risk analysis, CRC press, 2003.
6. Tomasoni, A. M. , Models and methods of risk assessment and control in dangerous goods transportation (DGT) systems, using innovative information and communication technologies, Ecole Nationale Supérieure des Mines de Paris; Università degli studi di Genova-Italie, 2010.
7. Aven T, Quantitative risk assessment: the scientific platform, Cambridge University Press, 2011.
8. Biass S., Frischknecht C. & Bonadonna C. A fast GIS-based risk assessment for tephra fallout: the example of Cotopaxi volcano, Ecuador -Part II: vulnerability and risk assessment, Natural hazards, vol. 64, no. 1, pp. 615-639, 2012.
9. Centrone G., Pesenti R., Ukovich W., Hazardous materials transportation: a literature review and an annotated bibliography. In C. Bersani, A. Boulmakoul, E. Garbolino, R. Sacile (editors), Advanced Technologies and Methodologies for Risk Management in the Global Transport of Dangerous Goods, IOS NATO Science Series Book, Amsterdam, 2008.
10. Drouin C. & Leroux, D, Transport et Environnement : Analyse des risques associés au transport des matières dangereuses en milieu urbaine, Congrès de l'Association des transports du Canada, 2004.
11. Boulmakoul A., Karim L., Laarabi M. H., Sacile R. & Garbolino, E, Mongodb hadoop distributed and scalable framework for spatio-temporal hazardous materials data warehousing, in 7th International Congress on Environmental Modelling and Software: Bold Visions for Environmental Modeling, iEMSs 2014.
12. Diernhofer F., Kohl B. & Hörhan R, Risk Assessment of Transport for the Dangerous Goods in Austrian Road Tunnels, in 4th International Symposium on Tunnel Safety and Security, Frankfurt am Main, Germany, 2010.
13. Viktor F, Automating the Continuous Deployment Pipeline with Containerized Microservices, The DevOps 2.0 Toolkit, 2016.
14. Bob F, Microservices, IoT and Azure: Leveraging DevOps and Microservice Architecture to deliver SaaS Solutions, Apress; 1st ed., 2015
15. Karagiannis V, Chatzimisios P, Vazquez-Gallego F, Alonso-Zarate J, A Survey on Application Layer Protocols for the Internet of Things, Transaction on IoT and Cloud Computing, 2015.
16. Foster A., A Comparison between DDS, AMQP, MQTT, JMS, REST and CoAP, Version 1.4, January 2014.
17. Cherradi G, El Bouziri A, Boulmakoul A, Smart Data Collection Based on IoT Protocols, JDSI'16, ISSN 2509-2103, 2016.
18. Gubbi J, Buyya R., Marusic S, and Palaniswa mi M, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, vol. 29, pp. 1645-1660, 2013.
19. Amirian P, Winstanley A. & Basiri A, NoSQL storage and management of geo-spatial data with emphasis on serving geospatial data using standard geospatial web services, 2013.
20. Holmes S, Getting MEAN with Mongo, Express, Angular, and Node, Manning Publications, 2015.
21. Cherradi G, El Bouziri A, Boulmakoul A, Environmental Information System for HazMat Transportation and Risk Assessment , INTIS 2016, ISBN 978-9954-34-378-4, ISSN 2351-9215, 2016.
22. Aloha (Areal location of hazardous atmospheres), technical documentation 2013.