DOI 10.12694/scpe.v14i4.928
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 14, Number 4, pp. 201–213. http://www.scpe.org

A MESSAGE ORIENTED MIDDLEWARE FOR CLOUD COMPUTING TO IMPROVE EFFICIENCY IN RISK MANAGEMENT SYSTEMS*

MARIA FAZIO, ANTONIO CELESTI, ANTONIO PULIAFITO, AND MASSIMO VILLARI †

Abstract. Transportation of Dangerous Goods represents a sensitive problem due its congenital high potential risk of causing disaster if an accident occurs. Transportation of Dangerous Goods Risk Management systems reduce the possibility of both accidental disasters and terrorist attacks detecting unusual events and blocking possible threats. Cloud computing can facilitate the development of such kinds of systems thanks to new emerging paradigms and technologies. In this paper, we discuss the design of a new Message-Oriented Cloud Middleware for Cloud, that can be used to develop a Cloud-based Transportation of Dangerous Goods Risk Management system. More specifically, we investigate issues on Transportation of Dangerous Goods, in order to focus the attention on the requirements of the Risk Management system. Then, we describe how to use the Message-Oriented Cloud Middleware for Cloud architecture and the necessary utilities in particular here for supporting Transportation of Dangerous Goods.

Key words: message oriented middleware, cloud computing, federation, service provisioning, planetary system model.

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction. Risk Management Systems are very complex distributed systems in which different heterogeneous infrastructures and resources need to be properly integrated and managed. In particular, the risks involved in the Transportation of Dangerous Goods (TDGs) over multi-modal ways (e.g., freeways, railways, air and sea routes) have been attracting great interest in the recent years. In fact, dangerous goods can cause terrible disaster if an accident occurs, producing uncontrollable effects in highly populated areas or during popular events. Moreover, the risk becomes more concrete if we consider that dangerous goods can potentially be an objective of terrorist attacks. TDGs is a very complex problem, involving economical, legislative and technological aspects. The complexity raises due to the fact that for reducing as much as possible all risks, the previous aspects need to be addressed all together.

Nowadays, advanced technologies in the field of ICT (Information and communications technology) promise a way to track in real time the entity of such transportations and efficiently manage the exposure to related risks. Innovative technologies can actively support goods tracking and provide valuable added value services to provide legally requested information and also to minimize risk in case of failures and accidents. Nevertheless, the development of a TDG Risk Management System is not easy at all due to the number of utilities that need to be integrated and coordinated (e.g., sensing, high performance computing, storage, security, etc).

Cloud computing has reached a high level of complexity embracing many application fields. Indeed, the Cloud-like technologies allow the development of next generation versatile systems in which different types of technologies and hardware/software solutions can be integrated.

In this paper, we present a novel Message-Oriented Middleware (MOM4C), that can be usefully adopted for the development of a TDG Risk Management System. MOM4C allows to set up Cloud facilities aggregating different Cloud utilities coming from different enterprises, organizations, and governments in a federated environment. According to the MOM4C terminology, a "Cloud Facility" is a mash-up Cloud service composed integrating one or more Cloud utilities, instead, a "Cloud utility" is a specific Cloud service (e.g., virtualization, storage, network, computation, security, sensing, data analytics, etc). In simple words, the aim of the middleware is to acts as a liaison among utilities in order to support the deployment of advanced, flexible, and differentiated Cloud facilities [17]. In such a versatile scenario enterprises, organizations, and governments become, at the same time, customers and providers. MOM4C provides flexibility, efficiency, and elasticity for the setup of Cloud facility to Cloud providers, seamlessly integrating the utilities belonging to different heterogeneous environments or administrative domains. It allows to expand existing Cloud systems and to integrate several virtual and physical resources. Its ability of collecting heterogeneous utilities and abstracting their functionalities to high level Cloud facilitates is very useful for the development of advanced applications

*This work was partially supported by Projects SIMONE and SIGMA, Italian National Operative Program (PON) 2007-2013.
†DICIEAMA, University of Messina, Contrada Di Dio, 98166 Sant'Agata - Messina (mfazio(acelesti,apuliafito,mvillari).unime.it).

for Internet of Things (IoTs). MOM4C has been designed according to the Message-Oriented model. This model has already been used for the designing of Cloud middleware such as IBM WebSphere MQ (MQSeries), TIBCO Rendezvous, and RabbitMQ. In comparison with them, MOM4C allows to develop services fitting the requirements of Cloud computing.

Due to its features, MOM4C can offer a solid support to TDG scenarios. A TDG Risk Management System mainly requires: i) a monitoring system able to localize and track dangerous goods even analyzing their states according to different types of information (e.g, temperature, pressure, gas detection, etc); ii) a data collection and elaboration system able to correlate the different pieces of information coming from the monitoring system; iii) an intelligence transportation system able to provide: transport mode optimization and traffic management through a "smarter" use of transport networks; iv) an informative system able to disseminate alerts to the population in case of disaster providing pieces of information that potentially can save lives.

In order to satisfy such requirements, we analyze the possibility of arranging Cloud facilities for TDG Risk Management Systems (TDG Cloud Facilities) combining several Cloud utilities, in particular we gathered the utilities we develop next, in four main branches: sensing, virtualization, big data management, and trusted computing.

The rest of the paper is organized as follows. In Sect. 2, we discuss the main concerns regarding the TDG. In Sect., 3, we present the MOM4C computing model, discussing a few architectural aspects in Sect. 4. An example of TDG Cloud facility arranged be means of MOM4C is discussed in Sect. 5. A possible combination of both hardware/software solutions and technologies for the implementation of TDG Cloud facilities is discussed in Sect. 6. In Sect. 7, we provide an overview regarding other available Cloud middleware, highlighting how they differ from MOM4C. Conclusions and remarks are summarized in Sect. 8.

2. TDG Concerns. TDG risk management systems able to reduce the risk of both accidental disasters and terrorist attacks make extensive use of sensing infrastructures to assess the risk itself and to detect unusual events. TDG risk management systems asks for a continuous monitoring of activities related to transportation. It is necessary not only to track the position of the vehicle and the status of the cargo, but it is also important to understand how the environment interacts during the transportation of dangerous goods. Automatic vehicle identification techniques relying on Radio Frequency Identification (RFID) permit to electronically gather shipment information. Route planning can reduce the probability of disaster. It can be time-independent or reactive. In particular, route planning is reactive if real-time pieces of information about the conditions of the transport network are periodically updated in the management system. Such pieces of information are gathered by sensor networks and made available in real-time databases. In addition, Geographic information System (GIS) will permit geospatial data management for decision making processes.

2.1. The State of the Art on TDG. The TDG problem has been gathering great attention from both research community and business companies. The main goal is to develop a TDG risk management systems able to prevent disasters. In ICT fields, several initiatives appeared, each one addressing specific requirements.

MITRA [2] is a research project funded by the European Commission with the objective to prototype a new operational system based on regional responsibilities for the monitoring of dangerous goods transportation in Europe. It provides a real-time knowledge of position and contents of dangerous goods through the European Geostationary Navigation Overlay Service (EGNOS), that is a satellite based augmentation system developed by the European Space Agency, the European Commission and EUROCONTROL. In case of dangerous situations, GSM communications allow to alert the Security Control Centre, which is responsible to prevent accidents, manage crisis and enable quick intervention.

SMARTFREIGHT [3] is a European research project, partly funded by the European Commission under the 7^{th} Framework Program (7FP). The overall objective of SMARTFREIGHT is to address new traffic management measures towards individual freight vehicles by using open ICT services, with an emphasis on the interoperability between traffic management and freight distribution systems, and an integrated heterogeneous wireless communication infrastructure within the framework of CALM (Communication Access for Land Mobiles)

In [18], the authors propose a complete monitoring and tracking solution for truck fleets. The system exploits battery-powered environmental sensors (temperature, humidity, pressure, gas concentration and ionizing radiation levels), connected by a ZigBee-based Wireless Sensor Network. Collected data is then sent from the

vehicle to a remote server via a GPRS link. The GPS positioning system is integrated by the use of an Inertial Navigation System, which guarantees a precise estimate of the position also when the GPS signal is weak or temporarily lost.

The solution proposed in [20] aims to improve the security of maritime container transport of dangerous goods by the real-time monitoring of container state. This system uses micro-sensor technologies and radio frequency communication technology to obtain the dangerous goods condition inside containers, as well as automatic positioning in the cargo hold. Information on the state of dangerous goods are transmitted to the shore monitoring center on land through INMARSAT stations.

By comparing the different solutions for dangerous goods transportation, we have identified the following common goals: 1) localization and tracking means of freight transportation, 2) monitoring of goods according to several types of information (temperature, pressure, gas detection,...), 3) data collection and elaboration, 4) definition of policies for disaster prevention, 5) definition of policies for emergency management. However, the existing solutions exploit heterogeneous systems, hardly to be integrated. Indeed, they differ a lot in terms of sensor technologies, communication infrastructures, design of the system organization and software support. Here, our idea is to setup an environment able to harmonize these heterogeneous systems.

2.2. Open Issues. Companies operating in the monitoring of dangerous goods have to use specific technologies that depend on several factors: the type of dangerous goods that are tracked, their geographical position and route, means of transport, legislation of the country and so on. International Regulations define standard procedures for the treatment of dangerous goods. However, from a technological point of view, they do not provide any specification with reference to the monitoring infrastructure installation. The result is that actually there is no compatibility between different monitoring systems managed by organizations or companies, both in terms of hardware and software.

Another important issue is related to the transportation of the adopted solution. Each solution focuses on a specific method of transportation (such as ship, truck, airplane or railways) and the concept of multi-modal service is not faced at all. However, the aggregation of information from multi-modal ways can be extremely useful to predict terrorist attacks. Furthermore, in case of attacks, the management of different types of way out from the disaster area can save human lives.

A world wide standardized solution is still missing. Recent events have shown the importance of collaboration among different countries to fight against terrorism. So, we imagine a future transportation system where efforts will integrate activities along the roads, highways, railways, harbors and airports at once. The integration will also include activities provided by different operators inside the same country and among different countries.

3. The MOM4C Computing Model. Currently, many pieces of Cloud middleware have been appearing on the market. As highlight on the state of the art analysis discussed in Sect. 7, the available solutions are related to specific scenarios. On the contrary, the TDG risk management system requires to address a versatile scenario in which different utilities have to be integrated (e.g., sensing, virtualization, big data management, trusted computing, etc). For such a reason, in this paper we present a solution based on the MOM4C, a solution that in our opinion, well fits the requirements of TDG risk management systems. Differently from other available pieces of middleware, analyzed in Sect. 7, MOM4C abstracts the type of offered services, providing a framework able to integrate both the current and future Cloud solutions, offering to the customers the possibility to customize their Cloud facilities.

3.1. The Need of a Middleware for Emerging Cloud-Based Systems. Analyzing the trend of the Cloud computing market, we can highlight, on one hand, a growing number of providers that are investing in Cloud-based services and infrastructures and, on the other hand, the interest of companies in long-term, customizable and complex business solutions, which must be easy to be set up, reliable, and accessible through the Internet. MOM4C has been design to fill up this gap, integrating existing infrastructures and resources in form of Cloud utilities into one efficient, scalable, reactive and secure distributed system. Its deployment can be strategic for many different stakeholders, as shown in Fig. 3.1. MOM4C enables third-party enterprises and developers to implement Cloud facilities in an easy way, integrating different Cloud utilities (e.g., storage, network, computation, security, sensing, data analytics, etc) according to a mash-up development model. In

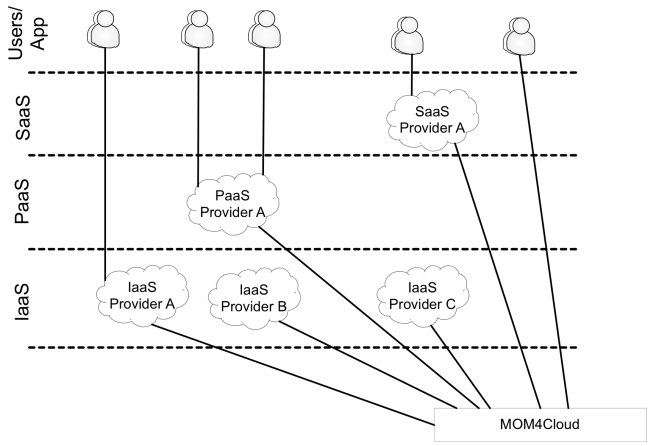

FIG. 3.1. *All stakeholders and cloud layers involved in MOM4C reference scenarios.*

this way, enterprises, organizations, and governments can quickly Cloud facilities integrating different Cloud utilities.

MOM4C enables Cloud providers to abstract the service level. Typically, Cloud providers can deliver three main service levels, i.e., Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). According to such a classification, MOM4C allows to develop Cloud facilities in form of IaaS, PaaS, and SaaS instances. It is important to notice that also Cloud utilities themselves can be hardware/software functionality delivered in form of IaaS, PaaS, and SaaS.

IaaS Providers deliver computers and devices (i.e., physical and/or virtual) and other resources. Typically, a Virtual Infrastructure Manager (VIM) controls one or more hypervisors each one running several Virtual Machines (VMs) as guests. A VIM allows to manage a large numbers of VMs (e.g., preparing disk images, setting up networking, starting, suspending, stopping VM, etc) and to scale services up/down according to customers' requirements. An example is represented by a provider that offers to end-users on-demand VMs execution. PaaS providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Software developers can implement and run their solutions on a Cloud platform without the cost of buying and managing the underlying hardware and software layers. Typically, the underlying computer and storage resources automatically scale up/down to match application demand. Another example is represented by a provider that offers a platform that collects data coming from one or more sensor networks and that offer Application Program Interfaces (APIs) for data processing, hence enabling developers to implement intelligent sensing applications. SaaS providers, typically deliver on-demand pieces of software via Web 2.0 that are usually priced on a pay-per-use basis. Providers install and manage applications in the Cloud and users access these ones from software clients, generally web browsers. A case in which a provider that offers via Web 2.0 interface an office automation software suite such as Google Drive to manage documents. Furthermore, a Cloud facilities built through MOM4C will be able to integrate Cloud utilities even belonging to different administrative domains in a federated system. In a federation, each entity is independent and can not be conditioned by a "central government" in its activities. The components of a federation are in some sense "sovereign" with a certain degree of autonomy from the "central government": this is why a federation can be intended more than a mere loose alliance of independent entities. Moreover, the treatment of all the data and information transferred through MOM4C is performed according to secure policies able to assure: data confidentiality, data integrity, data authenticity, non-repudiation of the sender, non-repudiation of the receiver.

3.2. MOM4C: a Planetary System Model. The MOM4C computing model was inspired by a planetary system model. Due to its native ability in integrating heterogeneous infrastructures and resources in form of Cloud utilities, MOM4C can potentially offer a wide plethora of Cloud Facilities able to provide complex, customizable and differentiated mash-up services.

A monolithic design of the proposed system is inconceivable, since it implies a heavy effort in management

of all the available components, low scalability and useless service availability for clients. On the contrary, to guarantee the maximum flexibility, we have conceived MOM4C as a very modular architecture, in which every client can customize Cloud facilities according to their business requirements. From the client point of view, we can schematize MOM4C as well as a planetary system, as shown in Fig. 3.2. The planetary system is composed

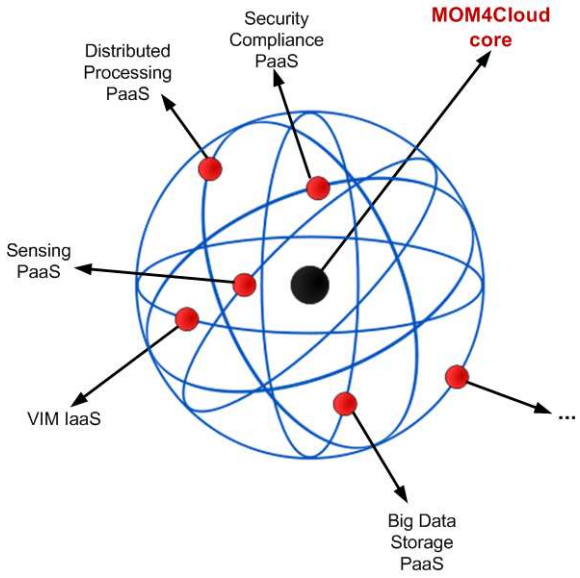

FIG. 3.2. *Planetary system model for service provisioning through MOM4C.*

by one or more planets that orbit around a central star. According to our abstraction, Planets identify available utilities. For example, utilities can be: i) VIM, for on-demand VM provisioning; ii) Sensing PaaS, collecting data by different sensing environments; iii) Distributed Processing PaaS, providing high computational power; iv) Big Data Storage PaaS, providing distribute storage for huge amount of data, and so on. The core of MOM4C is the star of the planetary system. It provides all the basic functionalities necessary for the life of planets. Specifically, it includes a scalable messaging and presence system, security mechanisms for data integrity, confidentiality and non-repudiation, federation management and other specific communication features for the management and integration of heterogeneous utilities.

All the possible combinations of planets specialize the behavior of the planetary system. According to our similitude, a specific planetary system configuration, including target planets defines the Cloud facility. In fact, according to our definition, the Cloud facility has to be customizable from clients in order to fit specific business scenarios.

4. MOM4C Architecture. MOM4C is designed according to the message-oriented paradigm, in order to provide an efficient communication system among different distributed components. From the message-oriented paradigm, MOM4C inherits a primary benefit, that is loosing coupling between participants in a system due to their asynchronous interaction. It results in a highly cohesive, decoupled system deployment. It also decouples the performance of the subsystems from each other. Subsystems can be independently scaled, with little or no disruption of performance into the other subsystems. With reference to the management of unpredictable activity overloads in a subsystem, the message-oriented model allows to accept a message when it is ready, rather than being forced to accept it. MOM4C adds important features, that are strategic for business in Cloud. Its major benefits includes:

- **Modularity**: the middleware can be quickly extended using different modules characterizing different available utilities. It can be easily customized in order to suit a specific Cloud scenario.
- **Polymorphism**: each distributed entity in the system can play different roles according to the system requirements. Different rules includes both the core management tasks and the utility-related tasks.

- **Security**: an indispensable requirement for the large-scale adoption Cloud computing is security, especially in business scenarios. Security has to be natively addressed at any level of communication (intra-module, inter-module, and inter-domain), providing guarantees in terms of data confidentiality and data integrity.
- **Federation**: it is a strategic approach to promote collaboration among cooperating Cloud providers.

4.1. Cluster and Execution Layers. As depicted in Fig. 4.1, MOM4C is based on a distributed architecture, organized in two layers, that are the Cluster Layer (CL) and the Execution Layer (EL). The Cluster

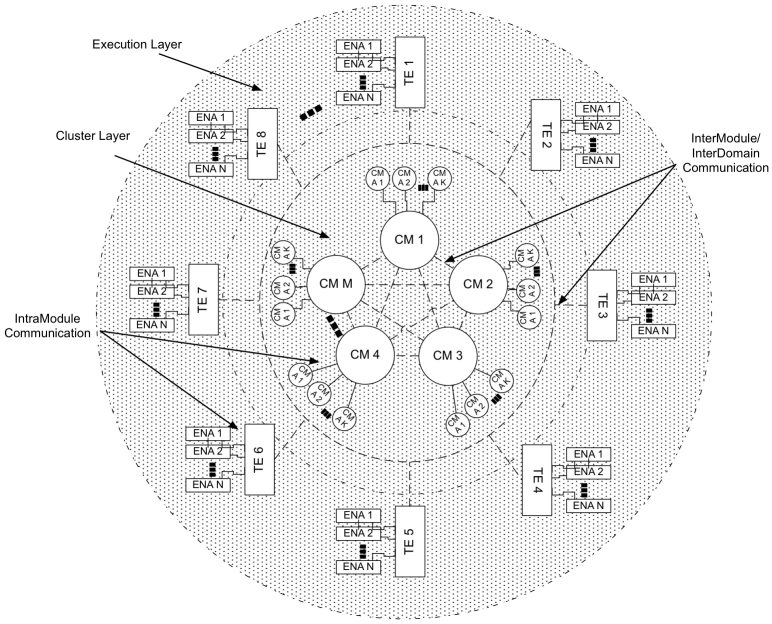

FIG. 4.1. *MOM4C basic scheme.*

Layer represent the "core" of MOM4C. It consists of an overlay network of decentralized Cluster Manager (CM) nodes. Each CM is responsible for the working activities of the Task Executor (TE) nodes belonging to the cluster. The EL is composed of TEs, which are intended to perform operative tasks. TEs can be trained to perform a specific task. It means that they do not instantiate all the services and utilities available in MOM4C, but they download code, initialize and configure services, launch software agents whenever they receives instructions from the CM. An appropriate utility module configuration into TEs allows to specialize MOM4C services. According to the specific code in execution at TEs, we have different characterizations of the EL.

To perform different types of tasks (e.g., VM execution and sensing data gathering), we set up specialized ELs, which independently works according to the CL specifications. Such an organization of roles and activities carries out high modularity to the MOM4C system. Building around the Cluster Layer many TE layers at the same time characterizes the MOM4C behavior. Thus, an ad-hoc layers configuration is designed to support a specific scenario. With reference to the planetary system model, the star includes all the functionalities of the Cluster Layer, which sustains the whole system. Any orbit represents a specific Execution Layer and the planet is the utility offered by TEs belonging to the related Execution Layer.

Another important feature of MOM4C is the polymorphic nature of nodes. At different times, each physical node can serve as CM or TE. However, only a node in a cluster is elected as CM and actively works for managing the whole cluster. Some other node are elected as "passive CMs", which are redundant CMs that can quickly replace the active CM if it fails. This approach improves the fault tolerance of the CL. The size of the cluster depends on the system workload and it can dynamically change according to the specific elasticity requirements of the system. About TEs, they can belong to one or more ELs, hence they work at different Cloud utilities. Such a concept is better explained in Fig. 4.2. For example, TE 1, 2, 3, 4, 5, 6 are hypervisor servers working

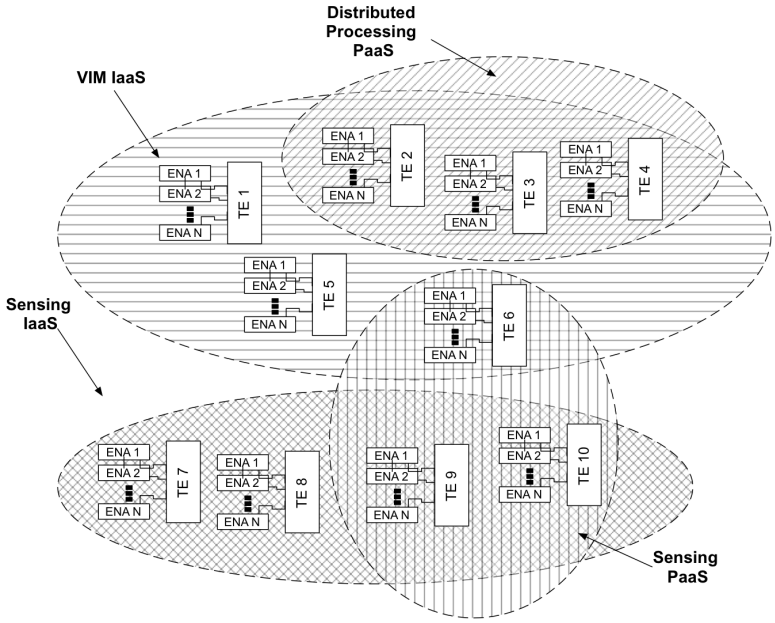

FIG. 4.2. *Hybrid Executor Node Layer composition.*

to provide a VIM IaaS. At the same time, TE 2, TE 3, and TE 4 work also to provide a Distributed Processing PaaS, since software agents running on TEs are independent active processes. Following the example in Fig. 4.2, TE 7, 8, 9, and 10 work as embedded devices for Sensing IaaS provisioning, whereas TE 6, 9, and 10 works for a Sensing PaaS, for example collecting sensing data from TE 9 and 10 and providing services through the AJAX Web APIs of Web application deployed in TE 6.

4.2. All Turn Around the Communication System. The strong point of MOM4C is represented by its communication system. In fact, the middleware supports three types of communications:

- **IntraModule Communication**: it characterizes information exchange inside each node of the architecture, both CMs and TEs. It guarantees a seamless way for allowing their internal software modules to communicate each other.
- **InterModule Communication**: it governs communications between CMs and TEs and vice-versa.
- **InterDomain Communication**: is specific for communications among CMs belonging to different administrative domains, hence enabling InterCloud or Cloud federation scenarios.

In order to ensure as much as possible the middleware modularity, the tasks running on each node are mapped on different processes within the Operating System, which communicate each other by means of an Inter Process Communication (IPC) or InterModule communication. According to the message-oriented design of MOM4C, InterModule communications are based on an Instant Messaging and Presence (IMP) protocol. A presence system allows participants to subscribe to each other and to be notified about changes in their state. On the other hand, Instant messaging is defined as the exchange of content between a set of participants in near real time. InterDomain communications among different administrative domains are managed considering the federation agreements among the domains. Federation allows Cloud providers to "lend" and "borrow" resources. Thus, a CM of a domain is able to control one or more TEs belonging to other domains.

5. A Cloud Facility for TDG Risk Management Systems. In this Sect., we firstly discuss what the requirements are for a TDG Risk Management System, and than, we present and example of Cloud facility for TDG Risk Management System (TDG Cloud facility) combining four Cloud utilities, i.e., sensing, virtualization, big data management, and trusted computing.

5.1. Functional and Non-Functional Requirements. By comparing the different available initiatives in the field of TDG, analyzed in Sect. 2.1, we have identified the following functional requirements:

1. monitoring, localizing and tracking of dangerous goods even analyzing their states according to different types of information (e.g, temperature, pressure, gas detection, etc);
2. collect, analyze, and correlate the different pieces of information coming from different monitoring activities
3. transport mode optimization and traffic management through a "smarter" use of transport networks
4. disseminate alerts to the population in case of disaster providing pieces of informations that potentially can save lives.

Cloud computing can offers several benefits in carrying out all these activities and we are going to explain how by means of MOM4C.

Considering the monitoring related to the transportation of dangerous goods, existing solutions differ a lot in terms of sensor technologies, communication infrastructures, design of the system organization and software support. Companies operating in the monitoring of dangerous goods have to use specific technologies that depend on several factors: the type of dangerous goods that are tracked, their geographical position and route, mode of transport, legislation of the country and so on. International Regulations define standard procedures for the treatment of dangerous goods. However, from a technological point of view, they do not provide any specification with reference to the monitoring infrastructure installation. The result is that, usually, there is not compatibility between different monitoring systems managed by different organizations or companies, both in terms of hardware and software. Another important point is related to the adopted transportation solution. Each solution focuses on a specific method of transportation (such as ship, truck, airplane, or railways) and the concept of *concurrent* service is not faced at all. However, the aggregation of pieces of information from *concurrent* ways can be extremely useful to predict terrorist attacks. Furthermore, in case of attack, the management of different types of way out from the disaster area can save human lives.

Regarding non-Functional requirements, the TDG Cloud facility has to abstract the underlying infrastructures and resources through Cloud utilities. In fact, the MOM4C, represents the *GLUE* to integrate and homogenize such heterogeneous infrastructures and resources. By using the concept of virtualization, the MOM4C can abstract hardware and software resources and, thus, guarantee an high level of interoperability among different physical infrastructures involved in the intelligent transportation activities.

The monitoring activity causes a massive collection of data, which need to be organized and processed in a transparent way, in order to provide an integrated knowledge of the context. The context knowledge is the base to build up strategies at the National Security level. High amount of data means more efficient services, but implies high requirements in terms of processing power and storage space. However, the demands of resources significantly vary depending on several parameters, such as the geographical area, traffic, and so on. The distributed nature of Cloud computing guarantees high availability of computational and storage resources as services, which can be dynamically adapted to specific needs of the system. Concurrent transport of dangerous goods is characterized by specific constrains, which need guarantees on the quality of the informative services (e.g., reaction time to an event occurrence, synchronization of activities, trust in using third party support, etc). The high flexibility of the Cloud in dynamic configuration, management and optimization of resources and services allows to effectively respond to the quality of service requirements of the system.

Smart services supporting the transport requires to correlate pieces of information regarding the environment, goods, carriers and freight operators, and determine the best routes for goods transfer. MOM4C offers a very innovative approach to develop TDG risk management system through a distributed system where resources and context information are accessible through well-defined interfaces. This approach allows to implement new services without any knowledge of physical infrastructures and software architecture, making the TDG easier and flexible. Another important feature offered by MOM4C is its ability to manage a federation of several cloud providers (i.e., enterprises, organizations, and governments). Thus, we consider the Cloud environment as a constellation of hundreds of independent, heterogeneous, private/hybrid Clouds able to interact each other, maintaining separated their own administration domains accomplishing *inter-cloud* scenarios. This requirement is particularly important in the transport management, because actors that interact to improve their services do not intend to disclose their informative systems. Another important requirement in case of terroristic attack, is

the security of the TDG risk management system. In fact, in order to avoid "man in the middle" attacks, causing potential data corruption or unauthorized information disclosure, both communications among the different components and the access to these latter have to be trusted.

5.2. MOM4C Utility Composition. Thanks to its modularity, MOM4C allows to instantiate different types Cloud facilities. As previously discussed, as well as a planetary system is composed by a star with several planets that turn around it along their orbits, in MOM4C, a Cloud facility is built around the MOM4C core (i.e., the central star) and several Cloud utilities (i.e., planets). From an architectural point of view, we remark that the MOM4C core consist of an overlay network of decentralized CM nodes, whereas a each Cloud utility consists of an overlay network of TE nodes that offer a particular service.

In order to better explain the planetary system model at the basis of the MOM4C design, let us consider the possible utility composition to support the TDG risk management scenario. Considering both the functional and non-functional requirements discussed in Sect. 5.1, in our opinion a possible TDG Cloud facilities arranged with the MOM4C should four main Cloud utilities: sensing, virtualization, big data management, and trusted computing.

The *sensing* utility allows to virtualize different types of sensing infrastructures, adding new capabilities in data abstraction. It gathers sensing information from a peripheral decision-maker, called Virtual Pervasive Element (VPE), able to interact with smart sensing devices or sensing environments [10]. The *VIM* utility allows to aggregate heterogeneous computing infrastructures, providing suitable interfaces at the high-level management layer for enabling the integration of high-level features, such as public Cloud interfaces, contextualization, security [6] [12] and dynamic resources provisioning [8]. The *big data management* utility allows to storage a huge amount of data an to perform an efficient retrieval of them adopting, for example, the map/reduce approach. The *Trusted Computing utility* allows interact with the Trusted Platform Module (TPM) on the physical host [7] by means of a software agent. The TPM is a hardware micro-controller that allows to combine hardware and software components by building a chains of trust. In addition by means of the remote and deep attestation protocols, the utility is able to verify the configuration of physical hosts and VMs.

Figure 5.1 depicts an example of TDG Cloud facility combining seven Cloud utilities. The utilities are orchestrated by the MOM4C core with which communicate in a secure way through the MOM4C communication system. Utility 1 collects sensing data coming from several sensor networks monitoring different transport ways (i.e., freeways, railways, air and sea routes). Utility 2, add to the Cloud facility the ability to virtualize the physical datacenter, by means of a VIM, in order to arrange different scalable virtual environments. In addition, this utility allows the Cloud facility to scale up/down the virtual infrastructure asking external resources when it is required (e.g., when the physical resources are run out the Cloud facility can ask for resources to external providers). Utility 3 enables big data management through a system able to store huge amount of pieces of data and to efficiently retrieve and process them using map-reduce mechanisms. The utility is built into a virtual infrastructure that can be scaled up/down when required thanks to utility 2. Finally, utility 4 adds to the Cloud facility trusted computing capabilities enforcing remote attestation in both physical and virtual servers respectively considering physical and virtual TPMs. In this way, if a physical or virtual machine is corrupted the utility will be able to immediately detect the attempt of attack and block it.

Regarding the secure communication between the MOM4C core and the various utilities, the middleware natively involve secure communication by means of digital signing and message encryption mechanisms. From an architectural point of view, this means that both CM and TE nodes communicate each other through secure channels. Such a feature is enforce by MOM4C independently from any specific type of Cloud facility.

6. Solutions and Technologies for the TDG Cloud Facility.

6.1. Communication System. According to the design specifications of MOM4C, the InterModule and InterDomain communication systems have been implemented using a well known MIPO, that is XMPP. The XMPP (RFC 3920 and RFC 3921), also called Jabber, is becoming more and more popular due to its flexibility to suit different scenarios where a high level of re-activeness is strongly required. Despite it was born for human interaction via chat room it can be used to develop the communication of whatever distributed system well fitting the requirements of Cloud computing. XMPP is an XML-based protocol used for near-real-time, extensible instant messaging and presence information. XMPP remains the core protocol of the Jabber Instant

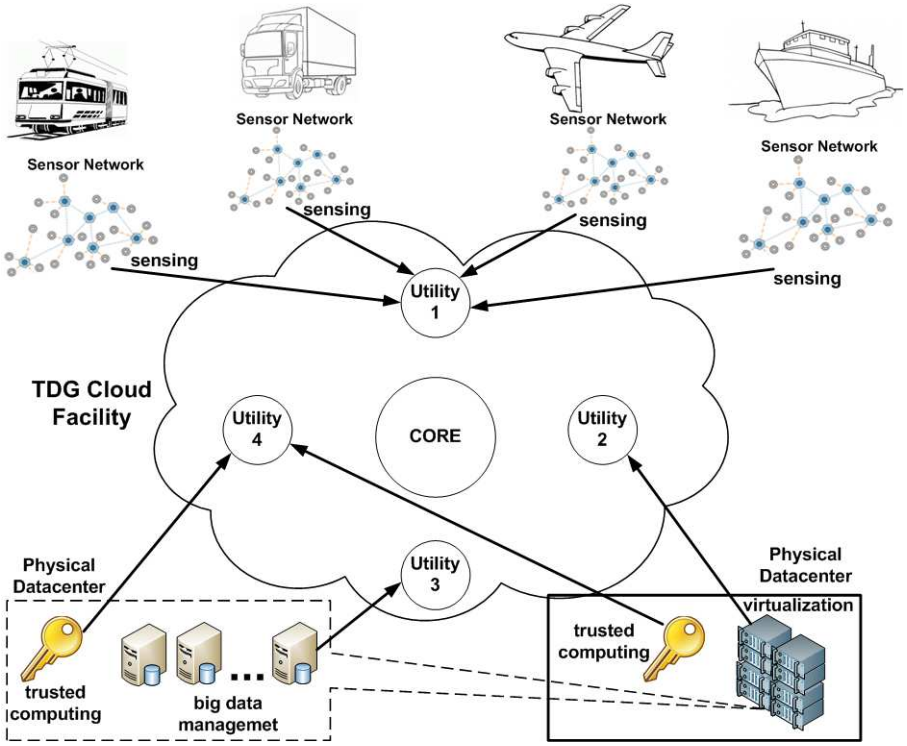

FIG. 5.1. *Example of TDG Cloud facility with MOM4C.*

Messaging and Presence technology. The "Jabber" technology leverages open standards to provide a highly scalable architecture that supports the aggregation of presence information across different devices, users and applications. As the client uses HTTP, most firewalls allow users to fetch and post messages without hindrance. Thus, if the TCP port used by XMPP is blocked, a server can listen on the normal HTTP port and the traffic should pass without problems. Custom functionality can be built on top of XMPP, and common extensions are managed by the XMPP Software Foundation. This make the protocol flexible to be extended with QoS and security features.

6.2. Cluster Manager Node. The CM Coordinator is the core of a CM.In order to communicate with other nodes of the system, the CM Coordinator exploits three different interfaces: IntraModule Interface, used to interact with specific modules, InterDomain Interface, used for interconnecting the CMs belonging to different administrative domain, TE Interface, for communicating with all the TE nodes. Apart from the IntraModule Interface that works by means of the DBUS communication system, the InterModule and InterDomain interfaces are connected to different XMPP rooms, in order to separate different communications. It performs high level operations assigning tasks to different TEs, taking into account the system workload and features of each node. Moreover, through InterDomain Interface, the CM Coordinator provides information about the Cluster state, collected through its Interfaces.

6.3. Task Executor Node. The TE Coordinator is the main component of the TE node. It is responsible to execute the command sent by a CM. In addition, it monitors resources and the Operating System in order to optimize their usage. For example, a real time information on CPU, RAM, storage and network utilization can be acquired. The main activities of a TE Coordinator are: 1) Streaming the collected information; 2) providing the collected information on demand; 3) sending a specific notification (alert) when a pre-determined condition is verified. All the TE Coordinators have to interact exploiting the persistent XMPP connection made available through the CM Coordinator Interfaces. The other nodes, in order to perform temporary peer-to-peer communications, can attend an ephemeral XMPP session connecting themselves to an "utility room".

6.4. Possible Solutions for Utility Implementation. In Sect. 1, for arranging Cloud facilities useful for TDG Risk Management Systems, we recognized four types of utility: sensing, virtualization, big data management, and trusted computing. The implementation of these Cloud utilities can be achieved using different software tools and frameworks. Here below we describe what we selected for accomplishing our cloud environment.

The sensing utility, has to be developed considering the Sensor Web Enablement (SWE) standard which enables developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and useable via the Web. Further standards that have to be considered includes

- **Observations & Measurements (O&M)**. Standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time.
- **Sensor Model Language (SensorML)**. Standard models and XML Schema for describing sensors systems and processes associated with sensor observations; provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing of task-able properties, as well as supports on-demand processing of sensor observations.
- **Transducer Model Language (TransducerML or TML)**. The conceptual model and XML Schema for describing transducers and supporting real-time streaming of data to and from sensor systems.
- **Sensor Observations Service (SOS)**. Standard web service interface for requesting, filtering, and retrieving observations and sensor system information. This is the intermediary between a client and an observation repository or near real-time sensor channel.
- **Sensor Planning Service (SPS)**. Standard web service interface for requesting user-driven acquisitions and observations. This is the intermediary between a client and a sensor collection management environment.
- **Sensor Alert Service (SAS)**. Standard web service interface for publishing and subscribing to alerts from sensors.
- **Web Notification Services (WNS)**. Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services and other elements of service workflows.

There are several Open Source, Free-Ware, and Commercial Off-the-Shelf (COTS) activities committed to the development of Sensor Web Enablement (SWE) oriented software. This includes software to support servers, middleware, and clients, as well as tools for creating and validating SWE encodings. Interesting tools include 52 North, MapServer, OOSTethys, Space Time Toolkit, SWE Common Library, Process Execution Engine Library,

The virtualization utility can be developed using different hypevisors and frameworks (e.g., KVM/QEMU, XEM, VMware, VirtualBox, libvirt, etc). Regarding the VIM, developers have to consider the possibility either to develop a customize solution or using existing solution including, for example, OpenStack, Open Nebula, Clever, Nimbus, Eucalyptus, etc. In addition the Open Virtualization Format (OVF) standard has to be considered.

The big data management utility can be developed considering both a distributed file system and a parallel processing system able to fast retrieve and process pieces of data. To this regard, a possible solution is represented by Apache Hadoop. It is a popular framework providing both a distributed file system (HDFS), and a processing environment adopting the map-reduce paradigm. Further valuable alternative solutions include Nutch, Cloudera, Hypertable, HBase, Apache Mahout, and Apache Cassandra.

Regarding the trusted computing utility, the Institute for Applied Information Processing and Communication (IAIK) of the Graz University of Technology (AT) have been developing many software libraries and tools. The Trusted Computing Group (TCG) has defined a Trusted Software Stack (TSS) to simplify the access from software modules to TPM. In particular TSS defines an Application Programming Interface to operating systems and applications. Furthermore for supporting the development of trusted applications, the TGG has defined TCG Device Driver Library (TDDL). Further details are available in [1]

7. Related Works. Some works in literature deal with the need of Cloud middleware, addressing specific issues and exploiting different technologies. To support application execution in the Cloud, in [13], authors present CloudScale. It is a piece of middleware for building Cloud applications like regular Java programs and easily deploy them into IaaS Clouds. It implements a declarative deployment model, in which application developers specify the scaling requirements and policies of their applications using the Aspect-Oriented Pro-

gramming (AOP) model. A different approach is proposed in [19]. The authors present a low latency fault tolerance middleware to support distributed applications deployment within a Cloud environment. It is based on the leader/follower replication approach for maintaining strong replica consistency of the replica states. If a fault occurs, the reconfiguration/recovery mechanisms implemented in the middleware ensure that a backup replica obtains all the information it needs to reproduce the actions of the application. The middleware presented in [5] has been designed aiming mission assurance for critical Cloud applications across hybrid Clouds. It is centered on policy-based event monitoring and dynamic reactions to guarantee the accomplishment of "end-to-end" and "cross-layer" security, dependability, and timeliness. In [9], the authors present a piece of middleware for enabling media-centered cooperation among home networks. It allows users to join their home equipments through a Cloud, providing a new content distribution model that simplifies the discovery, classification, and access to commercial contents within a home network. In [14], the authors focus their work on the integration of different types of computational environments. In fact, they propose a lightweight component-based middleware intended to simplify the transition from clusters, to Grids and Clouds and/or a mixture of them. The key points of this middleware are a modular infrastructure, that can adapt its behavior to the running environment, and application connectivity requirements. The problem of integrating multi-tenancy into the Cloud is addressed in [4]. The authors propose a Cloud architecture for achieving multi-tenancy at the SOA level by virtualizing the middleware servers running the Service Oriented Architecture (SOA) artifacts and allowing a single instance to be securely shared between tenants or different customers. The key idea of the work is that the combination between virtualization, elasticity and multi-tenancy makes it possible an optimal usage of data center resources (i.e., CPU, memory, and network). A piece of middleware designed for monitoring Cloud resources is proposed in [16]. The presented architecture is based on a scalable data-centric publish/subscribe paradigm to disseminate data in multi-tenant Cloud scenarios. Furthermore, it allows to customize both granularity and frequency of received monitored data according to specific service and tenant requirements. The work proposed in [11] aims to support mobile applications with processing power and storage space, moving resource-intensive activities into the Cloud. It abstracts the API of multiple Cloud vendors, thus providing a unique JSON-based interface that responds according to the REST-based Cloud services. The current framework considers the APIs from Amazon EC2, S3, Google and some open source Cloud projects like Eucalyptus. In [15], the authors present a piece of middleware to support fast system implementation and ICT cost reduction by making use of private Clouds. The system includes application servers that run a Java Runtime Environment (JRE) and additional modules for service management and information integration, designed according to a SOA.

8. Conclusion and Remarks. In this paper, we have discussed the design of a Cloud-based Risk Management System for the Transportation of Dangerous Goods (TDG). A TDG Risk Management System requires the integration of different heterogeneous sensing infrastructures and different ICT assets regarding for example monitoring, processing, storage, etc.

MOM4C enables software architects to seamlessly design such a kind of distributed system thanks to a message oriented approach. In fact, MOM4C, according to a planetary system model, allows software architects to arrange distributed systems as Cloud facilities combining different utilities. In addition, the middleware allows to different enterprises, organizations, and governments to cooperate in a federated Cloud environments in a transparent way.

More specifically, an example of TDG Cloud facility has been described combining four main Cloud utilities, i.e., sensing, virtualization, big data management, and trusted computing. An interesting aspect of MOM4C is its ability to adapt the Cloud facilities to the system requirements even in a heterogeneous environment. As we have demostrated, such a feature makes the middleware, a valuable solution for the development of next generation versatile systems in the field of TDG.

REFERENCES

[1] Trusted Computing Grouop (TCG): http://www.trustedcomputinggroup.org/.
[2] 2004-2006. MITRA: Monitoring and intervention for the transportation of dangerous goods. http://www.mitraproject.info/.
[3] 2009. SMARTFREIGHT project, FP7-216353. http://www.smartfreight.info//.

[4] A. Azeez, S. Perera, D. Gamage, R. Linton, P. Siriwardana, D. Leelaratne, S. Weerawarana, and P. Fremantle, *Multi-tenant SOA Middleware for Cloud Computing*, in IEEE CLOUD'10), 2010, pp. 458–465.
[5] R. Campbell, M. Montanari, and R. Farivar, *A middleware for assured clouds*, Journal of Internet Services and Applications, 3 (2012), pp. 87–94.
[6] A. Celesti, M. Fazio, M. Villari, and A. Puliafito, *Se clever: A secure message oriented middleware for cloud federation.*, in IEEE Symposium on Computers and Communications (ISCC'13), ISCC '12, 2013.
[7] A. Celesti, M. Fazio, M. Villari, A. Puliafito, and D. Mulfari, *Remote and deep attestations to mitigate threats in cloud mash-up services*, in World Congress on Computer and Information Technologies (WCCIT'13), Sousse, Tunisia, 2013.
[8] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, *Integration of clever clouds with third party software systems through a rest web service interface*, in IEEE Symposium on Computers and Communications (ISCC'12), ISCC '12, 2012, pp. 827–832.
[9] D. Diaz-Sanchez, F. Almenarez, A. Marin, D. Proserpio, and P. Arias Cabarcos, *Media Cloud: an open cloud computing middleware for content management*, IEEE Transactions on Consumer Electronics, 57 (2011), pp. 970–978.
[10] M. Fazio, M. Paone, A. Puliafito, and M. Villari, *Huge amount of heterogeneous sensed data needs the cloud*, in SSD'12, 2012.
[11] H. Flores and S. N. Srirama, *Dynamic Re-configuration of Mobile Cloud Middleware based on Traffic*, in IEEE MASS'12), October 8-1 2012.
[12] A. Juels and A. Oprea, *New approaches to security and availability for cloud data*, Communication of the ACM, 56 (2013), pp. 64–73.
[13] P. Leitner, B. Satzger, W. Hummer, C. Inzinger, and S. Dustdar, *Cloudscale: a novel middleware for building transparently scaling cloud applications*, in SAC'12, 2012, pp. 434–440.
[14] E. Manias and F. Baude, *A component-based middleware for hybrid grid/cloud computing platforms*, Concurrency and Computation: Practice and Experience, 24 (2012), pp. 1461–1477.
[15] H. Nagakura and A. Sakurai, *Middleware for creating private clouds*, Fujitsu Scientific & Technical Journal (FSTJ), 47 (2011), pp. 263–269.
[16] J. Povedano-Molina, J. M. Lopez-Vega, J. M. Lopez-Soler, A. Corradi, and L. Foschini, *Dargos: A highly adaptable and scalable monitoring architecture for multi-tenant clouds*, Future Generation Computer Systems, May (2013).
[17] A. Ranabahu and M. Maximilien, *A Best Practice Model for Cloud Middleware Systems*, in Best Practices in Cloud Computing: Designing for the Cloud, 2009.
[18] F. Valente, G. Zacheo, P. Losito, and P. Camarda, *A telecommunications framework for real-time monitoring of dangerous goods transport*, in Intelligent Transport Systems Telecommunications,(ITST),2009 9th International Conference on, October 2009, pp. 13 –18.
[19] Z. Wenbing, P. Melliar-Smith, and L. Moser, *Fault Tolerance Middleware for Cloud Computing*, in IEEE 3rd CLOUD'10, July 2010, pp. 67–74.
[20] Z. Yingjun, X. Shengwei, X. Peng, and W. Xinquan, *Shipping containers of dangerous goods condition monitoring system based on wireless sensor network*, in Networked Computing (INC), 2010 6th International Conference on, may 2010, pp. 1 –3.

Edited by: Maria Fazio and Nik Bessis
Received: Nov 2, 2013
Accepted: Jan 10, 2014