Architecting Access Control for Business Processes in the Cloud

Khaled Gaaloul[1], Sami Yangui[2], Samir Tata[2] and Hend*erik* A. Proper[1]
[1]Public Research Centre Henri Tudor, Luxembourg.
{Khaled.Gaaloul,Erik.Proper}@tudor.lu
[2]Institut Mines-Telecom, Telecom SudParis, CNRS UMR Samovar, Evry, France.
{Sami.Yangui,Samir.Tata}@telecom-sudparis.eu

Abstract

Enterprise architecture (EA) aims to provide management with appropriate indicators and controls to steer and model service-oriented enterprises. Nevertheless, common enterprises architecture frameworks lack of access control mechanisms supporting security requirements within organizations. Moreover, the rapid permeation of information technology motivates new computing paradigms such as cloud computing. In this paper, we propose an approach for modeling access control requirements in enterprise architecture, and supporting its provisioning in Cloud providers. The idea is to leverage EA paradigms to ensure business-IT alignment when modeling access control, and deploying access control mechanisms in the Cloud. Our approach is illustrated through the handling of an e-Government scenario, in which EA modeling and appropriate Cloud resources provisioning are motivated.

1. Introduction

Enterprise architecture (EA) is generally considered to provide a good steering instrument to analyze the current state of the enterprise and guard the cohesion and alignment between the different aspects of an enterprise such as business processes and their ICT (Information and Communications Technology) support. EA aims to provide management with appropriate indicators and controls to steer and model service-oriented enterprises [14, 10].

The architecture modeling languages aim to support EA specification and description of enterprises components and their relationships; thereby ensuring an overall picture of the enterprise design and deployment. A prime example is the ArchiMate[1] modeling standard for EA [9]. This means that architects can use ArchiMate to model, amongst others, an organization's products and services, how these products and services are realized by business processes, and how in turn these processes are supported by information systems and their underlying IT infrastructure [9]. However, such techniques and languages do not address security issues in a satisfactory way [18, 3, 4]. For instance, access control artefacts are simply represented on the IT level without taking into accounts the modeling (i.e. process level) and enforcement (i.e. application level) of access control policies.

With the rapid permeation of information technology into the physical world and human society, new computing paradigms such as Cloud Computing emerge rapidly. The National Institute of Standards and Technology (NIST for short) related to the U.S Department of Commerce defines Cloud Computing as a new model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage resources, applications, services, etc.). These resources should be swiftly provisioned and released with minimal management effort and service providers interaction [11].

Promoted by the governments and industries, together with the effort of universities and research institutes around the world, a new revolution of information technology is on the eve [23]. In this ongoing revolution, security has become the leitmotif for politics. Moreover, security is nowadays considered, by the industry, as a major concern that has gained increasing focus to research for new solutions. An example of such effort are the recent calls for cyber security projects research under the scope of the digital agenda for Europe, by the initiative Horizon 2020[2].

For organizations focusing more on technology architecture, cloud computing could indeed presents a tendency solution. But for businesses that want to successfully adopt cloud computing in a way that aligns to their business strategy, enterprise architecture is imperative. The contribution of this paper is twofold: (1) modeling access control for

[1]http://www.opengroup.org/archimate/

[2]http://ec.europa.eu/digital-agenda/en/cybersecurity

978-1-4799-4299-2/14 $31.00 © 2014 IEEE
DOI 10.1109/IWAISE.2014.9

business processes in EA, and (2) deploying access control policies in Cloud providers. The goal is to focus on the most important features and aspects of access control in EA, express its Cloud readiness and provide appropriate mechanisms to provision them in a Cloud context.

As for the first contribution, we introduce the role-based access control (RBAC) standard [16], and experiment its relevance to ArchiMate. Then, we identify, through a case study, the relevant access control's components to be migrated to the Cloud. The second contribution relates our defined approach to provision the identified access control's components in existing Cloud providers. By provisioning, we mean the allocation of all necessary provider resources to host and run an application/service to deploy.

The remainder of this paper is structured as follows. Section 2 introduces the extension of ArchiMate with the RBAC concepts. Section 3 presents the case study and identify the access control components to be deployed in the Cloud. Section 4 details and comment our defined approach to perform this deployment. Section 5 discuss related work. Finally, section 6 concludes and outlines future work.

2 Access Control in ArchiMate

In this section, we identify the organizational needs as well as the security requirements in EA models. ArchiMate lacks specific guidelines for modeling an enterprise from a security perspective [4]. Dealing with that, we present the RBAC access control model, and identify its relevant concepts and relationships to be mapped to ArchiMate.

2.1 The ArchiMate language

ArchiMate is an Open Group standard [19] for the modeling of enterprise architectures[3], emphasizing a holistic view of the enterprise. This means that architects can use ArchiMate to model, amongst others, an organization's products and services, how these products and services are realized and delivered by business processes, and how in turn these processes are supported by information systems and their underlying IT infrastructure [10].

The ArchiMate language defines three main layers [9]: (1) The Business layer offers products and services to external customers, which are realized in the organization by business processes; (2) The Application layer supports the business layer with application services which are realized by (software) application components; (3)m The Technology layer offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication devices and system software.

[3]http://www.opengroup.org/archimate/

Table 1. ArchiMate business layer concepts, obtained from [6, 10]

ArchiMate concept	**Definition**
Business actor	Individual persons, but also groups of people within an organization.
Business role	A role that an actor fulfills in an organization. This role is usually defined as the work carried out by an actor.
Organizational service	The following concepts realize a service [6]: *Business processes, business functions, business interactions*.
Business event	A business event is something that happens (externally) and may influence business processes, functions or interactions.
Business object	An entity that is manipulated by behavior such as business processes or functions.

The scope of this paper remains at the organizational level (i.e., roles, actor, business process, etc.). Hence, we focus on ArchiMate business layer meta model. As implied by name, the business layer focuses on an organization's business concepts such as products, (commercial) services, and business processes. A description of the main concepts are defined in Table 1.

2.2 Access control model

Organizations use access control mechanisms to mitigate the risks of unauthorized access to their data, resources, and systems. An access to a resource is determined based on the relationship between the requester and the organization or owner in control of the resource. In other words, the requester's role will determine whether access will be granted or denied. Several access control models exist to address changes in organizational structures, technologies, organizational needs, technical capabilities, and organizational relationships.

The RBAC model is a widely implemented mechanism for protecting system resources standardized by the American National Standard for Information Technology (ANSI). The RBAC model needs only to be made to role assignments, which are significantly fewer than individual assignments [16]. The RBAC model relies on user authentication, which in turn relies on identity management and defines relationships between the main concepts of Users, Roles and

Table 2. RBAC concepts, obtained from [1]

Concept	Definition
User (U)	A Person often human, but can also be systems.
Role (R)	R is a responsibility defined for an organization.
Role Hierarchy (RH)	It defines a partially ordered role hierarchy in an organization.
User Assignment (UA)	Each U has a set of associated R.
Permission (P)	An approval of a mode of access (i.e. operation) to a resource (i.e. business object).
Session (S)	S is a mapping between U and possibly many R.
Permission Assignment (PA)	Each R has a set of associated P.
Constraint	Constraints restrict permissions such as a separation of duties (SoD).

Permissions. RBAC's constraints restrict permissions depending on contextual information such as separation of duties (SoD) [2]. The RBAC model is presented in figure 1 and the core concepts are textually described in Table 2 based on the foundational work of [1].

2.3 Business layer and access control

Figure 1 describes the mapping between the ArchiMate business layer model and the RBAC model. Note that we use only an excerpt of the ArchiMate business layer meta model to focus on these concepts and relations relevant for the access control model RBAC. Concepts and relations mapping have been facilitated by the application of existing approaches for ontology mapping [13, 24]. They are explained as follows:

- The concept *Business actor* defines an individual persons (e.g., customers or employees), but also groups of people (e.g., departments or business units) within the organization. In RBAC, we define a *User* as a specialization of a *Business actor*.

- The concept *Business role*: A role that an actor fulfils in an organization. Importantly, this role is usually defined as the work carried out by an actor. In RBAC, we define an organizational *Role* as a specialization of a *Business role*.

- The concept *Business object*: An entity manipulated by behavior such as business processes or functions. In RBAC, we define a *Resource* as a specialization of a *Business object*.

- The RBAC relations: *User assignment* and *Permission assignment*, to manipulate resources, are respectively defined in ArchiMate as *assigned to* and *accesses* relations.

3 Illustrative Example

In this section, we present a real world scenario from an e-Government case study [15] i.e. Mutual Legal Assistance (MLA) process to well illustrate the meta model integration. This system defines a a decentralized scenario involving Eurojust, a national authorities of two European countries, regarding the execution of measures for protection of a witness in a criminal proceeding. We focus on the MLA process and their underlying ICT using EA modeling language ArchiMate. Then, RBAC constraints are integrated within ArchiMate based on our mapping efforts.

3.1 MLA architecture

Here we describe the MLA process cross Eurojust organizations A and B. At the business level, we define the main actors: Prosecutor A and Judicial Authority Officer (JAO) B. The work consists of granting access to an external role Prosecutor when issuing an MLA request (see the business interaction 'Send MLA Request' on the top of Figure 2). The analyze of the request is done by the actor JAO B who will give access to the specified files of the business object 'MLA documents' (see the internal business process 'Process MLA' in Figure 2). The reason of the business interaction is that the organizational service MLA requires two roles (Prosecutor and JAO) to be executed.

At the application level, Eurojust integrates services such as MLA service and CMS (Case Management Service) to process data on the individual cases on which Eurojust national members are working (see application services, components and data objects in Figure 2).

3.2 The access control framework

The application CMS is defined to support data access and processing. At an architectural level, this application is defined at the application layer of ArchiMate. We develop an RBAC solution supporting CMS and its 'Access Control Component'. We present an access control framework (ACF) to support authorization policies within enterprise architecture (see Figure 3). ACF is defined as a set of software components which accept requests to access resources, analyze these against policies representing actual access rights to resources, and return a response based on this analysis.

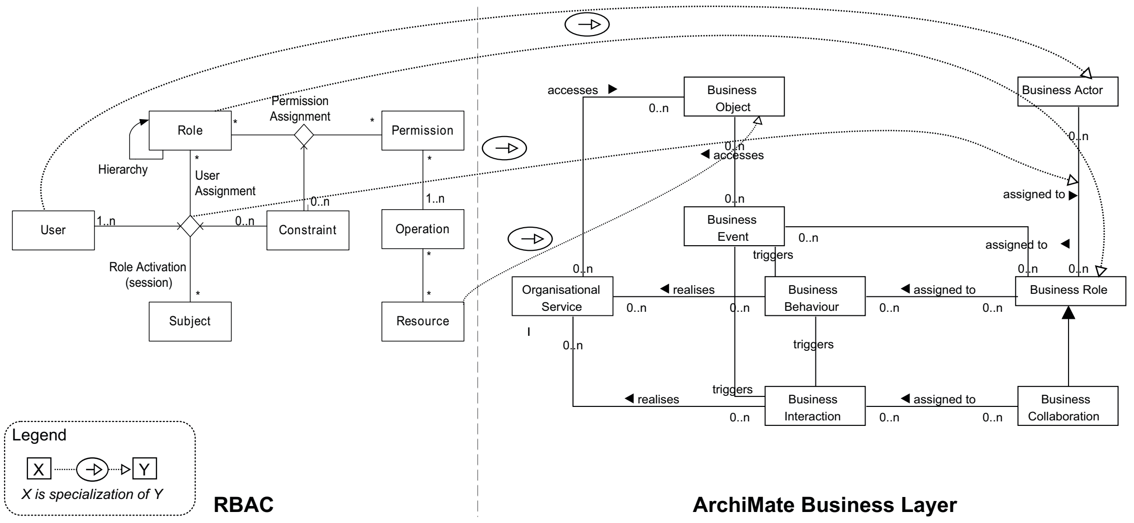

Figure 1. Mapping of ArchiMate business layer with RBAC concepts

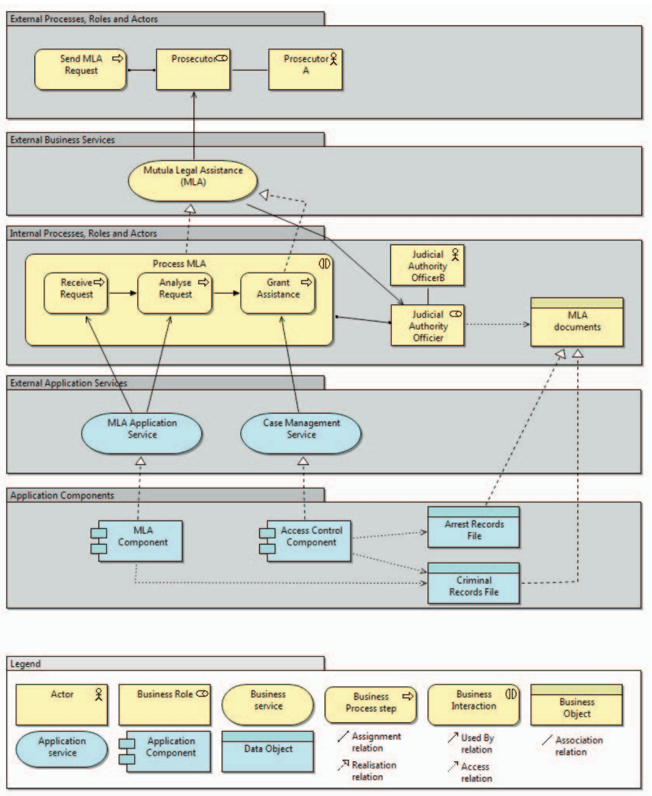

Figure 2. MLA model using ArchiMate extensions with RBAC mapping

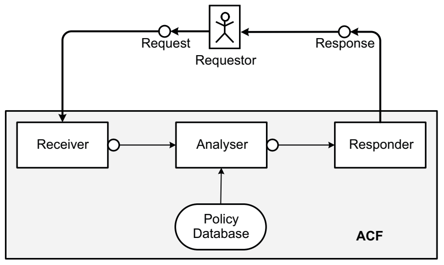

Figure 3. Access control framework

To illustrate the original architecture of an ACF, a request is issued by the requester, which is received by the *Receiver* component in ACF. This is then sent to the *Analyzer* component that queries policies stored in a policy database. A response is generated by the *Responder* component, which defines a decision (permit, deny, or not applicable) that is sent back to the requester. The application components are: the *Receiver*, the *Analyzer* and the *Responder*, where the policy database refers to data object in the ArchiMate application layer.

Based on this, we highlighted that the MLA process defines decentralized collaboration between different actors, where a Cloud environment could be suitable for specific services such as secure data exchange. Thence, we decide to provision its correspondent ACF in Cloud platforms in order to take advantages brought by a Cloud context such as the efficient and secured management of a highly distributed and dynamic environment and the pay as-you-go model [5].

4 Provisioning ACF components in Cloud providers

In this section, we detail our defined approach to provision in a target Cloud provider ACF components of a given process. This approach is based and extends our already performed service-micro containers [21, 12]. This choice was motivated by the nature of the services packaging process for micro-containers building which consists to embed only one service within. This coincides perfectly with ACF architecture since we consider each ACF component as an autonomous service interacting with the others through appropriate communication messages. We introduce our already performed service micro-containers in Section 4.1. Then, we detail the performed process to package the ACF components in micro-containers in Section 4.2. After that, we describe how we deploy them in a target Cloud provider in Section 4.3.

4.1 Service micro-containers

In a previous work, we demonstrated that the use of classical service containers (e.g. Apache Tuscany, Apache ODE, Apache Axis, etc.) is not suitable in Cloud Computing context [21, 12]. In fact, we highlighted that classical service containers are neither scalable nor elastic. To address these drawbacks, we proposed a novel prototype of service containers that we called micro-containers. The origin of this name comes from the fact that micro-containers provide the minimal functionalities to manage hosted service lifecycle. These basic functionalities ensure the minimal main process of our micro-container (e.g. services hosting, interaction with clients, etc.). For example, we failed to incorporate a safety module for managing access since it is a prototype and management competitions module as we are assuming a single service per container.

We thought of designing a system composed of two main parts:

1. The service micro-container,
2. The generic packaging platform that build the micro container and package the service to deploy on it.

Since we consider several types of services (languages, bindings, etc.), we are able to generate dynamically the correspondent micro-container from the packaging deployment platform for each service to be packaged. An overview of the main components of the packaging framework and architecture of generated micro-containers are detailed in Figure 4.

4.2 ACF components packaging

To package an ACF component and build the appropriate micro-container for it, one must mainly provide for the deployment framework two elements:

1. The component to package with all its artefacts (code, resources, etc.),
2. A deployment descriptor that specifies the container options.

Note that the packaging framework process ACF components to package as autonomous and elementary services. The *Processor module* analyzes the deployment descriptor to determine if there is non-functional properties to enclose (e.g. mobility, monitoring) and then, process the service source code, detects the service binding types, instantiates an appropriate *Communication module* implementing these bindings from the *Communication Generic Package* and associates it the service sources. The same principle is also followed for the selection of the *invocation module* to run the service once packaged in the micro-container. Based on the service implementation programming language, the appropriate *invocation module* is instantiated from the *Invocation Generic Package*.

The resulting code represents the generated micro-container code. It is composed only of the necessary modules for the deployed service, no more, no less. Generated micro-container hosts the service and implements its bindings regardless its communication protocol support as long as they are included in the *Generic Communication Package* and regardless the programming language as long as they are included in the *Invocation Generic Package*. Adding new communication protocols or programming languages support consists in adding the correspondent components in the packaging framework generic packages.

Henceforth, each generated service micro-container consists at least of three modules:

- *Communication module* to establish communication and to support connection protocols,
- *Invocation module* to process ingoing and outgoing data into and out of the server (packing and unpacking data),
- Service module to store and invoke the packaged service and its contract (service descriptor).

Subjecting ACF services to the packaging framework allow us to package each one of them in a specific micro-container. The obtained set of the micro-containers are standalone applications (i.e. Java ARchive files) that can be run and interact to ensure provide the business functionality of the initial ACF introduced in Section 3.2.

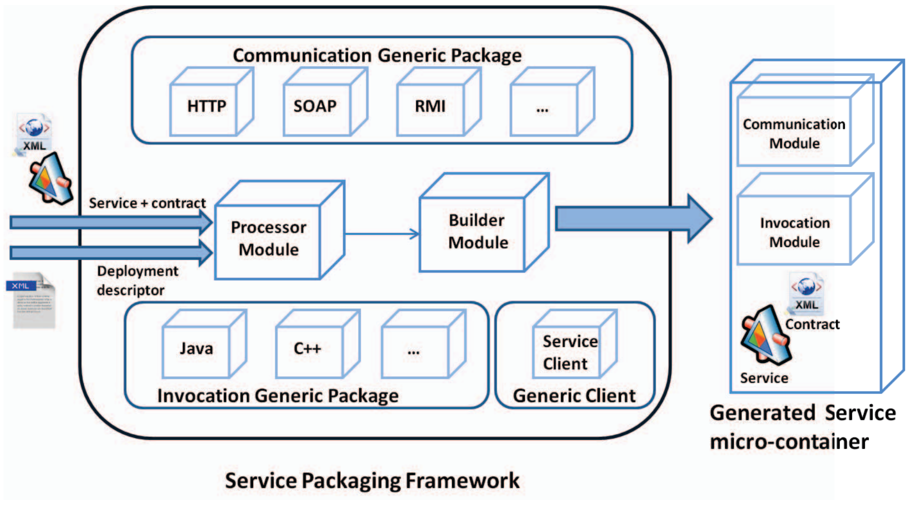

Figure 4. Service micro-container packaging framework

4.3 ACF components deployment

There are two ways to deploy the obtained micro-containers. The first one consists of deploying them as standalone applications in an IaaS. To perform this, we upload and run the micro-containers in virtual machines instantiated from an infrastructure manager solution such as OpenNabula or OpenStack.

The second way to deploy the service-micro containers consists on the use of our already performed generic PaaS application provisioning and management API (called COAPS API) [17, 20]. The use of COAPS API ensures the real independence of our approach regarding the target platform until the end of the process. We have implemented COAPS API to allow human and/or software agents to provision and manage PaaS applications. This API exposes a set of generic HTTP operations (i.e. GET, POST, PUT and DELETE) for Cloud applications management and provisioning regarding the target PaaS. it provides an abstraction layer for existing PaaS allowing PaaS applications provisioning in a unified manner.

5 Related work

There exist several IT Governance frameworks that have some focus on enterprise security. One of the most known frameworks is the Control Objectives for Information and related Technology (COBIT) [8] which is already in the version 5 and has specific internal IT related goals with security (e.g., security of information, processing infrastructure and applications). One standard that focused on IT security is the ISO/IEC 2700 [7] which has a practice guide addressing access control issues. Our work focused on the application area, where we have architected and deployed access control mechanisms in the cloud. The primer results provide access control concepts as well as services in PaaS applications.

The Zachman framework [22] is an enterprise architecture framework for enterprise architecture, which provides a formal and highly structured way of viewing and defining an enterprise. The framework defines six different perspectives (Scope, Business model, Information system model, Technology model, Detailed description and Actual system) describing the information which is considered essential in an enterprise architecture. These perspectives should be described in six different ways (Data, Function, Network, People, Time and Purpose). The Open Group Architecture Framework (TOGAF) is a framework for enterprise architecture which provides a comprehensive approach for designing, planning, implementing, and governing an enterprise information architecture [19]. TOGAF contains an architecture development method (ADM) that describes which steps should be taken to develop an enterprise architecture that has the four architectural domains (Business, Data, Application and Technology). Nevertheless, neither Zachman nor TOGAF provide specific solutions to express access control requirements from the strategy to the cloud migration.

6 Conclusion and Future Work

In this paper, we have proposed an approach modeling access control in enterprise architecture, and deploying its mechanisms in the Cloud. The idea is to leverage EA paradigms to ensure business-IT alignment when modeling access control and then, provisioning access control mechanisms in existing Cloud providers. We have used the mapping techniques to extend ArchiMate with RBAC model, which helped us to express the security needs of the MLA scenario from business strategy to application. Moreover, we have identified which components should be provisioned in the Cloud i.e. the access control framework component. To perform this deployment, we use our already developed service-micro containers to package these service before deploying them in target IaaS and/or PaaS providers.

In the near future, we plan to extend the packaging framework to support access control policies expressed by Cloud end users and include them to generated micro-containers. To that end, a solution consists in extending the deployment descriptor schema allowing end users to express required policies to packages within the service in the micro-container.

References

[1] G.-J. Ahn and R. S. Sandhu. The RSL99 language for role-based separation of duty constraints. In *RBAC '99: Proceedings of the fourth ACM workshop on Role-based access control*, pages 43–54, New York, NY, USA, 1999. ACM.

[2] R. A. Botha and J. H. P. Eloff. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal*, 40(3):666–682, 2001.

[3] C. Feltus, E. Dubois, H. A. Proper, I. Band, and M. Petit. Enhancing the Archimate standard with a responsibility modeling language for access rights management. In *Proceedings of the Fifth International Conference on Security of Information and Networks*, pages 12–19, New York, NY, USA, 2012. ACM.

[4] K. Gaaloul and H. A. Proper. An Access Control Model for Organisational Management in Enterprise Architecture. In IEEE, editor, *The 9th International Conference on Semantics, Knowledge & Grids*, pages 135–149, Beijing, China, 2013.

[5] R. Grossman. The Case for Cloud Computing. *IT Professional*, 11(2):23–27, March 2009.

[6] M.-E. Iacob, H. Jonkers, M. M. Lankhorst, H. A. Proper, and D. Quartel. *ArchiMate 2.0 Specification*. The Open Group, 2012.

[7] ISO/IEC. ISO/IEC 27002: Information technology Security techniques Code of practice for information security management, 2005.

[8] I. G. I. ITGI. *COBIT 4.1*. ISA, 2007.

[9] M. M. Lankhorst. *Enterprise Architecture at Work - Modelling, Communication and Analysis (4. ed.)*. The Enterprise Engineering Series. Springer, 2013.

[10] M. M. Lankhorst, H. A. Proper, and H. Jonkers. The Architecture of the ArchiMate Language. *Enterprise, Business-Process and Information Systems Modeling*, pages 367–380, 2009.

[11] P. Mell and T. Grance. The NIST Definition of Cloud Computing. Technical Report Special Publication 800 - 145, 2011.

[12] M. Mohamed, S. Yangui, S. Moalla, and S. Tata. Web Service Micro-Container for Service-based Applications in Cloud Environments. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011 20th IEEE International Workshops on*, pages 61–66, June 2011.

[13] N. Noy and M. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[14] M. Op 't Land, H. Proper, M. Waage, J. Cloo, and C. Steghuis. *Enterprise Architecture – Creating Value by Informed Governance*. Springer, Berlin, Germany, 2008.

[15] T. A. R4eGov. Towards e-Administration in the large, March 2006. `http://www.r4egov.eu/`.

[16] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[17] M. Sellami, S. Yangui, M. Mohamed, and S. Tata. PaaS-Independent Provisioning and Management of Applications in the Cloud. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 693–700, June 2013.

[18] M. W. A. Steen, M. E. Iacob, M. M. Lankhorst, H. Jonkers, M. Zoet, W. Engelsman, J. Versendaal, H. A. Proper, and K. Gaaloul. Service Modelling. 103:59–94, 2012.

[19] *The Open Group – TOGAF Version 9*. Van Haren Publishing, Zaltbommel, The Netherlands, 2009.

[20] S. Yangui, I.-J. Marshall, J.-P. Laisne, and S. Tata. CompatibleOne: The Open Source Cloud Broker. *Journal of Grid Computing*, 12(1):93–109, 2014.

[21] S. Yangui, M. Mohamed, S. Tata, and S. Moalla. Scalable Service Containers. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 348–356, Nov 2011.

[22] J. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.

[23] H. Zhuge and B. Xu. Basic operations, completeness and dynamicity of cyber physical socio semantic link network CPSocio-SLN. *Concurrency and Computation: Practice and Experience*, 1532-0634:n/a, 2010.

[24] S. Zivkovic, H. Kühn, and D. Karagiannis. Facilitate Modelling Using Method Integration: An Approach Using Mappings and Integration Rules. In *Proceedings of the Fifteenth European Conference on Information Systems, ECIS 2007, St. Gallen, Switzerland, 2007*, pages 2038–2049. University of St. Gallen, 2007.