

OCCIware - A Formal and Tooled Framework for Managing Everything as a Service

Jean Parpaillon, Philippe Merle, Olivier Barais, Marc Dutoo, Fawaz Paraiso

► **To cite this version:**

Jean Parpaillon, Philippe Merle, Olivier Barais, Marc Dutoo, Fawaz Paraiso. OCCIware - A Formal and Tooled Framework for Managing Everything as a Service. CEUR. Projects Showcase @ STAF'15, Jul 2015, L'Aquila, Italy. 1400, pp.18 - 25, Proceedings of the Projects Showcase @ STAF'15. <<http://ceur-ws.org>>. <hal-01188826>

HAL Id: hal-01188826

<https://hal.archives-ouvertes.fr/hal-01188826>

Submitted on 31 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

OCCIware - A Formal and Tooled Framework for Managing Everything as a Service

Jean Parpaillon¹, Philippe Merle², Olivier Barais¹, Marc Dutoo³, and Fawaz Paraiso²

¹ Inria Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France

² Inria Lille - Nord Europe, 40 Avenue Halley, 59650 Villeneuve d'Ascq, France

³ Open Wide, 151 Boulevard Stalingrad, 69100 Villeurbanne, France

Abstract. The OCCIware project aims at building a comprehensive, coherent while modular model-driven toolchain for managing any kinds of cloud resources, especially Data Center as a Service, Deployment as a Service, Big Data as a Service, and Linked Open Data as a Service. Leveraging the Open Cloud Computing Interface (OCCI) and its core model, the OCCIware toolchain applies a model-driven engineering approach based on a formal model of cloud resources and systems. This approach allows for better modularity, clear separation between functional (cloud resources) and non-functional concerns (security, scalability, reliability, etc.). The project brings together ten French partners - academics, SMEs, associations - and is supervised by a Strategic Orientation Committee of eleven top industrial and academic experts. The OCCIware project has been selected by French Ministry of Industry and funded by French Banque Publique d'Investissement (BPI).

1 Project Facts and Figures

Name: OCCIware

Call: French Ministry of Industry, “Investissement d’Avenir” (*Investment for Future*) programme, 4th “Cloud Computing and Big Data” call.

Source of funding: French “Banque Publique d’Investissement” (*Public Bank for Investment*).

Amount of funding: 3,3 M€.

Overall total budget: 5,6 M€.

Man power: 72 men/year.

Project consortium: 10 partners (see Table 1).

Involved people: <http://www.occiware.org/bin/view/About/Contributors>

Project Website: <http://www.occiware.org>

Project duration: 36 months.

Current Status: Started from December 2014.

Standardization Impact: supported by Open Grid Forum (OGF) standard development organization [6].

Support: Five French competitive clusters supporting the project (Systematic, Minalogic, PICOM, Images & Réseaux, Solutions Communicantes Sécurisées).
Strategic Orientation Committee: The complete list of members is available at http://www.occiware.org/bin/view/About/Strategic_Orientation_Committee
Open Source Software Resources: <https://github.com/occiware>

#	Name	Type	Work Package (bold: lead)
1	Open Wide	SME	WP1 , all
2	ActiveEon SA	SME	WP4 , all
4	Institut Mines Télécom / Télécom SudParis	Academic	All
4	Inria	Academic	WP2 , all
5	Linagora GSO	SME	WP5 , all but WP3
6	Obeo	SME	WP3 , all but WP4
7	OW2 Consortium	Association	WP1 and WP6
8	Pôle Numérique	Association	All
9	Scalair	SME	All but WP3
10	Université Josphe Fourier - Grenoble	Academic	All but WP3

Table 1. OCCIware Project Partners

2 Project Overview

2.1 Objectives

While cloud computing has become a reality in most IT domains, migrating existing software to the cloud or developing new innovative added-value cloud resources still require important R&D efforts. Indeed, cloud computing is plagued by heavy partitioning between cloud layers, technical implementations and business domains. For instance, while actual as well as “de facto” market standards have appeared in cloud computing, they are still tied to a particular layer: infrastructure (IaaS), platform (PaaS) or application (SaaS) [22] - actually mostly IaaS, and hardly SaaS.

The aim of the FSN Investissements d’Avenir (Cloud & Big Data 4) OCCIware project is to lower cloud computing adoption costs and break up barriers between its various layers, implementations, domains, by bringing to Open Cloud Computing Interface (OCCI) from Open Grid Forum (OGF) the power of formal methods, model driven engineering (MDE), and Models@run.time, in order to design, model, analyse, simulate, develop, deploy and execute every cloud computing resource as a service.

The OCCIware project aims at providing a comprehensive, coherent while modular model-driven toolchain for managing any kinds of cloud resources, especially Data Center as a Service, Deployment as a Service, Big Data as a Service,

and Linked Open Data as a Service. By using a simple resource-oriented meta-model, the OCCIware toolchain will allow to address any kind of resource-based software, drastically reducing development time by using Models@run.time approach [16] and/or code generation while improving overall quality and non-functional aspects of developed software, thanks to the separation of concerns.

Technically, the OCCIware toolchain is extending the Open Cloud Computing Interface from Open Grid Forum, by turning its core model [24] into a formal resource-oriented meta-model and designing new models addressing different domains. It also provides an Eclipse engineering framework for designing, testing and simulating cloud resources. Finally, the OCCIware toolchain includes a generic runtime for executing such designed cloud resources. In its architecture, the runtime implements the separation of concerns allowed by the meta-model, bringing security, reliability and scalability at no cost to developers of cloud resources.

The OCCIware project will be showcased in four demonstrators targeting Data Center as a Service, Deployment as a Service, Big Data as a Service, and Linked Data as a Service. The OCCIware project will be disseminated through Open Source communities (OW2 Consortium, Eclipse Foundation) and standardization bodies (OGF, DMTF) with help from eleven top international industrial and academic experts of the OCCIware's Strategic Orientation Committee.

2.2 Innovations Beyond the State of the Art

Formal methods have been used successfully in a large variety of domains like processor checking, embedded and critical systems. Aeolus ANR [13] and Mancoosi FP7 [21] projects have delivered the most comprehensive formal model of complex distributed systems. Their project leader is a member of the Strategic Orientation Committee. The OCCIware project aims at describing these models thanks to a single formal meta-model. Nevertheless, to the best of our knowledge, formal methods have not been used in the domain of cloud computing. The OCCIware project aims at proposing **the first formal model for designing and analysing every kind of cloud-based resource-oriented systems**.

This formal model will be based on the first-order relational logic and will be encoded with the Alloy lightweight specification language defined by Pr. Daniel Jackson from MIT [20]. Thanks to Alloy Analyser [14], we will analyse both the OCCIware meta-model and models of cloud resources in order to check their consistency, verify their properties and generate model instances automatically.

Several research projects such as FP7 REMICS [9], FP7 MODAClouds [3], FP7 SeaClouds [10], FP7 PaaSage [8], SINTEF CloudML [1], Eclipse Winery [12], StratusML [11], to cite a few, tackled the provisioning and deployment of multi-cloud applications on existing IaaS and/or PaaS resources through a model-driven engineering approach. These work do not tackle the design and execution of new kinds of cloud resources. Unlike the OCCIware project aims at providing **a model-driven engineering approach to manage every kind of cloud computing resources**.

Several cloud computing standards already exist. The DMTF's Open Virtualization Format (OVF) standard defines a standard packaging format for portable virtual machine images. The DTMF's Cloud Infrastructure Management Interface (CIMI) standard defines a RESTful API for managing IaaS resources only [19]. The OASIS's Cloud Application Management for Platforms (CAMP) standard targets the deployment of cloud applications on top of PaaS resources [4]. The OASIS's Topology and Orchestration Specification for Cloud Applications (TOSCA) standard defines a language to describe and package cloud application artifacts and deploy them on IaaS and PaaS resources [5]. The Eclipse Winery project provides an open source Eclipse-based graphical modelling tool for TOSCA [12] when the OpenTOSCA project provides an open source container for deploying TOSCA-based applications [7] [15]. The FP7 SeaClouds project [10] is based on both OASIS's CAMP and TOSCA standards. The OGF's Open Cloud Computing Interface (OCCI) recommendations [25] propose a generic resource-oriented model [24] for managing any kind of cloud resources, including IaaS, PaaS, and SaaS. Both OVF and OCCI address orthogonal concerns and then are complementary. OCCI is concurrent to CIMI because both address IaaS resource management but OCCI is more general purpose as it can be used also for any kind of PaaS and SaaS resources. CAMP and TOSCA can use OCCI-based IaaS/PaaS resources, so these standards are complementary. The OCCIware project is **based on and extending the OCCI recommendations**.

The FUI CompatibleOne [18] [27] and FP7 Contrail [2] projects have used OCCI recommendations for addressing cloud services interoperability and some partners of the OCCIware project were already involved in these projects. While these two projects have successfully achieved their functional goals, the lack of formal OCCI specifications prevents them to be easily extensible and limits the automation of their implementations. Turning the OCCI core model into a formal meta-model then designing a set of standard models out of it is one objective of the OCCIware project. The OCCIware project will provide **a formal OCCI model supported by a model-driven toolchain facilitating the design, development, and execution of any kind of OCCI-based cloud resources**.

2.3 OCCIware User Story

The OCCIware project outcomes can be illustrated through the following user story. Let a fully resource-oriented application "BeRest". It consumes the following resources through REST web services: compute and storage (IaaS), train and flight timetables (Linked Open Data) and personal calendars (SaaS). It provides the following service as resources: travel booking.

Thanks to our *formal meta-model* and its associated domain-specific language (see Section 3.1), the specifications of resources, including requirements and produced services can be expressed in an homogeneous way and can be verified at design time.

The *OCCIware engineering studio* (see Section 3.2) provides both Eclipse-based *graphical modeler and textual editor* to modelize cloud resources of this application. These tools are then able to expose the application’s cloud resources through different points of view, adapted to each actor:

- *architects* for designing the application,
- *developers* for mapping the design onto implementation,
- *CIO* for evaluating overall foreseen infrastructure cost.

Finally, the *OCCIware runtime* (see Section 3.3) is able to execute the application, *ie* mapping resources onto existing services (*e.g.* Amazon EC2 for infrastructure resources) and exposing “BeRest” services as OCCI resources. OCCIware studio tools will be able to configure the runtime for existing features and/or generating extensions through well-known extension points, for handling new features.

3 Project Organization and Outcomes

The OCCIware project has been split up to six work packages:

- **Transversal activities:** Management (WP1) and Communication and Dissemination (WP6),
- **Technical work packages:** Foundations (WP2), Eclipse toolchain (WP3), and Runtime (WP4),
- **Use Cases and Demonstrators** (WP5).

In addition to internal steering committee, a *Strategic Orientation Committee* has been set up to monitor the adequacy of OCCIware strategy with industrial needs and scientific rapidly evolving state-of-the-art.

3.1 WP2 - Foundations

Theoretical foundations of the project will produce scientific and formal tools, starting from the OCCI Core Specification. The following outcomes are expected.

The global technical architecture of the project results in a precise description of components and interfaces between the components developed in the project. It is planned to update this document with regard to feedback provided when implementing this architecture.

The OCCI formal model is a formalization of OCCI Core Model. The result is a proven meta-model and a set of constraints on this meta-model. This metamodel will be encoded with Alloy.

An OCCI dedicated language will be developed to express both static and dynamic aspects of the OCCIware models. It may be used for describing resources, manipulate them and simulate interactions between them.

Various OCCI resource models will be developed to address all OCCIware use-case requirements, as well as non-functional aspects of the runtime.

3.2 WP3 - Eclipse Toolchain

The Eclipse-based toolchain must help application developers but also CIOs to embrace the resource-oriented paradigm. The Eclipse Modeling Framework (EMF) is particularly suited for producing this kind of tools. The Obeo partner, as a recognized Eclipse expert and active member of the Eclipse community, will lead these tasks.

First, the OCCI meta-model will be translated into an Ecore meta-model. Eclipse tools will be leveraged to produce a text editor for the dedicated OCCI language implemented on top of Eclipse XText⁴. As the toolchain is dedicated not only to developers but also architects, a graphical modeler will be designed and developed on top of Eclipse Sirius⁵. A model-driven simulator will then be developed on top of CloudSim⁶ [17]. The link between the modeling environment and executed applications will be implemented with various code generators or connectors. Generators will generate runtime artifacts like code, configuration files, etc. Connectors will implement the causal link between models and running cloud resources, making OCCIware Models@run.time a reality. Finally, a decision-support tool will be developed to help evaluating the transition from legacy applications to cloud resource-based approach.

3.3 WP4 - Runtime Support

Leveraging the model-driven engineering approach, an execution platform will be able to interpret OCCI models at runtime, providing developers with non-functional aspects in the most transparent and efficient way. While developers can easily model their core business, turning these models into cloud resource-based applications requires a lot more skills due to non-functional aspects: scalable deployment, security, fault-tolerance, etc. Built on top of a kernel able to interpret OCCI models, connectors to existing cloud management interfaces will be developed for monitoring, supervision and distributed deployment. A web-based administration console for OCCI resources is also expected.

3.4 WP5 - Demonstrators and Use-Cases

Four use cases will be developed in the OCCIware project with the objectives of (1) providing requirements to technical work packages, (2) validating the outcomes of the latter and (3) demonstrating the use of the OCCIware toolchain in real industrial environments. The following use cases have been defined:

Datacenter as a Service will demonstrate the use of OCCIware for datacenter management (IaaS) ;

Deploy@OCCIware will offer interoperability layer above existing deployment and monitoring solutions ;

⁴ <https://eclipse.org/Xtext/>

⁵ <https://eclipse.org/sirius/>

⁶ <http://www.cloudbus.org/cloudsim/>

BigData and HPC will use OCCIware to propose scientific applications execution environment as a service ;

LinkedData as a Service will demonstrate the use of OCCIware tools for open linked data based applications.

4 Dissemination and Exploitation

Standardization of methodologies, languages and tools dedicated to resource-oriented software development is the objective of the OCCIware project. Their adoption by targeted audiences will then be a key indicator of the project success. A particular effort has been planned for disseminating technical and scientific results to following targeted communities:

- **Scientific communities** through publications in top journals and conferences. Our precise metamodel for Open Cloud Computing Interface is already published in [23].
- **Industrial communities** will be addressed through industrial events and business clusters.
- All technical outcomes will be published under open source license and then proposed to most appropriate **Open Source communities** (OW2, Eclipse, etc.). Our open source *erocci* generic OCCI Models@run.time is already available at [26].
- Finally, a close relation with OGF **Standards Definition Organization** has been established since the beginning of the project while connections with DMTF and OASIS organizations also exist with the project organization, through OW2 partner and Strategic Orientation Committee members.

Exploitation of the results by partners differs by their really business:

- **Service providers** (Scalair, Pôle Numérique) intend to improve their audience by providing services of high quality, accessible through standard technologies at a lower cost, challenging big actors in their respective market.
- **Integrators and software editors** (Linagora, Open Wide, ActiveEon, Obeo) will benefit from automated development toolchain for integrating at limited cost resource-oriented approach to their existing applications, enabling the access to the huge PaaS and SaaS market.
- **Research institutions** will benefit from the project by establishing their expertise in the first ever formal framework dedicated to the everything-as-a-service paradigm.

5 Conclusion

As the huge majority of software industry is moving toward a fully resource-oriented delivery model, it is time to offer developers a comprehensive toolchain leveraging this convergence to lower development costs and increase the overall

quality of resource-oriented applications. The OCCIware project aims at building this toolchain by bringing together existing technological and scientific tools usually promoted in separated communities: formal methods, model-driven engineering, meta-models, Models@run.time, REST architecture style, devops practices. The added value of the project being measurable through its adoption by software developers and scientifics, a particular effort is put to disseminate the OCCI meta-model, models and associated tooling to scientific and industrial communities, but also open source and standardization organizations.

References

1. CloudML Project Website, <http://cloudml.org>
2. Contrail FP7 Project Web Site, <http://contrail-project.eu>
3. MODAClouds FP7 Project Website, <http://www.modaclouds.eu>
4. OASIS CAMP Website, <https://www.oasis-open.org/committees/camp>
5. OASIS TOSCA Website, <https://www.oasis-open.org/committees/tosca>
6. Open Grid Forum Web Site, <http://ogf.org>
7. OpenTOSCA Project Website, <http://www.iaas.uni-stuttgart.de/OpenTOSCA>
8. PaaSage FP7 Project Website, <http://paasage.eu>
9. REMICS FP7 Project Website, <http://www.remics.eu>
10. SeaClouds FP7 Project Website, <http://www.seaclouds-project.eu>
11. StratusML Project Website, <http://www.stargroup.uwaterloo.ca/~mhamdaqa/stratusml>
12. Winery Eclipse Project Website, <https://projects.eclipse.org/projects/soa.winery>
13. Aeolus ANR Project Web Site, <http://aeolus-project.org>
14. Alloy Analyser Web Site, <http://alloy.mit.edu>
15. Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A., Wagner, S.: OpenTOSCA – A Runtime for TOSCA-based Cloud Applications. In: Service-Oriented Computing, pp. 692–695. Springer (2013)
16. Blair, G., Bencomo, N., France, R.B.: Models@run.time. *Computer* 42(10) (2009)
17. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41(1), 23–50 (2011)
18. CompatibleOne FUI Project Web Site, <http://compatibleone.org>
19. Davis, D., Pilz, G.: Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP. vol. DSP-0263 (May 2012)
20. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. MIT press, Revised edn. (2012)
21. Mancoosi FP7 Project Web Site, <http://mancoosi.org>
22. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. *National Institute of Standards and Technology* 53(6), 50 (2009)
23. Merle, P., Barais, O., Parpaillon, J., Plouzeau, N., Tata, S.: A Precise Metamodel for Open Cloud Computing Interface. In: Proceedings of the 8th IEEE International Conference on Cloud Computing (IEEE CLOUD 2015). IEEE, New York, USA (June 27 - July 2 2015), to appear

24. Nyrén, R., Edmonds, A., Papaspyrou, A., Metsch, T.: Open Cloud Computing Interface – Core. Specification Document GFD-P-R.183, Open Grid Forum, OCCI-WG (Apr 2011), errata update available at <http://redmine.ogf.org/projects/occi-wg/repository/show?rev=core-errata>
25. OCCI-WG: OCCI Working Group Web Site, <http://occi-wg.org/>
26. Parpaillon, J.: Erocci Web Site, <http://erocci.ow2.org/>
27. Yangui, S., Marshall, I.J., Laisne, J.P., Tata, S.: CompatibleOne: The Open Source Cloud Broker. *Journal of Grid Computing* 12(1), 1–17 (2013)