# OCciware, The Ultimate Framework for Managing Everything as a Service

## Table of Contents

- Abstract 3
- Background 3
- Disclaimer 3
- Introduction 4
- An OCciware Overview 4
- The OCciware Studio 7
- OCCI Extensions 8
- The OCciware Runtime 9
- OCciware Use Case 1: Datacenter as a Service 10
- OCciware Use Case 2: Deploy@OCciware 12
- OCciware Use Case 3: Big Data as a Service 13
- OCciware Use Case 4: Linked Data as a Service 14
- OCciware Outlook 16
- Conclusion 17
- Useful minks 17

## List of Exhibits

- Exhibit 1: Cloud Computing Issues and Challenges 5
- Exhibit 2: OCciware High-Level Architecture 6
- Exhibit 3: OCciware OCCI Studio 7
- Exhibit 4: OCciware Docker Studio 8
- Exhibit 5: OCciware Framework: From Studio to Runtime 9
- Exhibit 6: OCciware Runtime Modular and Extensible Architecture 9
- Exhibit 7: OCciware Runtime Backends 10
- Exhibit 8: Datacenter as a Service and Deploy@occiware Use Cases 11
- Exhibit 9: Big Data as a Service Use Case 13
- Exhibit 10: Linked Data Use Case 14
- Exhibit 11: OCciware Studio Product Line 16
Abstract

This white paper introduces OCCIware, an innovative technology to manage any kind of resources as a service. OCCIware provides a unique and comprehensive framework, from model to code execution, for modeling, designing, verifying, simulating, developing, testing, deploying and running every computing resource as a service. OCCIware leverages the Open Grid Forum (OGF) Open Cloud Computing Interface (OCCI) standard to address the lack of unified cloud computing standard. The OCCIware framework allows programmers to develop cloud applications faster through an integrated tool chain built on the OCCI standard. OCCIware provides a formal, consistent, and comprehensive specification for the OCCI standard thus making it the first cloud computing standard formally specified.

As a main outcome of the project, OCCIware will enable cloud developers and integrators to configure, deploy and manage all layers of their solutions (SaaS, PaaS, IaaS) on the cloud. OCCIware helps cloud consumers to avoid technology and vendor lock-in by providing them, whatever cloud provider they rely on and whichever cloud computing layer positioning they have with a single, unified and standard approach.

This white paper is based on the first results achieved by the project after one year of research and development. It introduces the OCCIware Studio and the OCCIware Runtime, the two main components of the OCCIware framework. It presents also the demonstrators that highlight the benefits of OCCIware in three major areas of cloud computing: IaaS, Big Data and Linked Data.

Background

OCCIware originated as a collaborative project supported financially by the French Fonds National pour la Société Numérique (FSN). The partners of the OCCIware project are: Open Wide (project coordinator), ActiveEon, Inria, Linagora, Obeo, OW2, Pôle Numérique, Université Grenoble Alpes, Scalair, Telecom Sud Paris. The project is endorsed by five Competitive Clusters: Systematic, Minalogic, PICOM, Images & Réseaux, Solutions Communicantes et Sécurisées. This white paper discusses outputs of the first year of work on OCCIware, which have been validated by its sponsors.
Introduction

The OCCIware project develops a framework for modeling, designing, verifying, simulating, developing, testing, deploying and running every computing resource as a service. OCCIware is a tool chain helping to develop cloud applications faster by taking advantage of the Open Grid Forum’s Open Cloud Computing Interface (OCCI) standard. Aligned with OCCI, the OCCIware project integrates and complements existing open source cloud projects such as ProActive Cloud Automation and Roboconf deployment systems, developed by ActiveEon, and Linagora respectively. All results from OCCIware are open source-licensed. OCCIware implements four use cases to validate the technology. They focus on datacenter management (DCaaS) and infrastructure (IaaS), IaaS, PaaS and SaaS deployment and reconfiguration inter-operability, Big Data as a Service (BDaaS) and Linked Data as a Service (LDaaS) through the European platform Ozwillo.com.

An OCCIware Overview

OCCIware was launched with the objective to create a tool chain to enable the management of all cloud computing resources as a service. The chosen approach is to establish everything as a resource thanks to the Open Cloud Computing Interface (OCCI) specifications.

OCCIware addresses today’s cloud computing major challenges in terms of interoperability and standardization: heterogeneity of cloud computing offerings, interoperability between cloud interfaces, integration to build multi-cloud systems, portability of applications and cloud computing users activities.

Exhibit 1, on the following page, illustrates these challenges.

---

OCCI comprises a set of open community-led specifications delivered through the Open Grid Forum. OCCI is a Protocol and API for all kinds of management tasks. OCCI was originally initiated to create a remote management API for IaaS model based services, allowing for the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring. It has since evolved into a flexible API with a strong focus on integration, portability, interoperability and innovation while still offering a high degree of extensibility.

---

(cc by) OW2 2016. This white paper is distributed for purposes of comment and discussion only.
In order to achieve its objective, OCCIware develops scientific and technical tools to considerably lower the cost of entry into the cloud market for existing software vendors. The resulting de-partitioning of cloud computing technologies will facilitate the development of innovative services with great added values.

Exhibit 1: Cloud Computing Partitioning Issues and Challenges

As shown in the architecture schema on Exhibit 2, next page, OCCIware provides the following outcomes and results:

- A formal framework for modeling, designing, verifying, developing, testing, simulating, deploying, executing and managing every kind of resource as a service.

- A dedicated language for managing every kind of resources as a service.

- A studio for engineering every kind of resources as a service. This Studio is based on Eclipse environment.
● A runtime (Model@run.time) support including a resources container, tools for deployment, supervision and administration.

● Four demonstrators in the domain of datacenter management and infrastructure (DCaaS); IaaS, PaaS and SaaS deployment and reconfiguration interoperability (deploy@occiware); Big Data as a Service (BDaaS); and LinkedData as a Service (LdaaS) through the European platform Ozwillo.

Exhibit 2: OCCIware High-Level Architecture
The OCChiware Studio

The OCChiware Studio consists in a factory that is able to produce visually customizable diagram editors for any cloud configuration business domain modeled in OCCI using the OCChiware Extension Studio shown in Exhibit 3.

![Exhibit 3: OCChiware OCCI Studio](image)

This studio, based on Eclipse, includes a text editor, graphic modelers, simulators, generators and connectors. It is up and running and is continuously integrated with other OCChiware components and with the OW2 erocci project. It allows one to design OCCI-compliant extensions and configurations.

The OCChiware Metamodel aims to formally specify the main OCCI concepts. As of today a first EMF metamodel is defined that includes OCCI basic concepts and adds new concepts such as Extension, Configuration, and EDataType that address some limitations of OCCI.
OCCIware, The Ultimate Framework for Managing Everything as a Service

OCCI Extensions

So far, the emerging OCCI standard was mostly used for managing IaaS resources. OCCIware will propose OCCI extensions in order to deliver domain specific designers for environments such as Docker, ProActive Cloud Automation, Roboconf, and more largely any IaaS infrastructure, hypervisor or public clouds.

With the Docker Studio (see Exhibit 4), OCCIware enables the visual design and flexible configuration of a cloud architecture based on Docker.

Exhibit 4: OCCIware Docker Studio
The **OCCIware Runtime**

The OCCIware runtime is based on the OW2 erocci project, a scalable and generic OCCI bus written in Erlang that federates multiple cloud runtimes ("backends") including Linagora’s Roboconf PaaS server and ActiveEon’s ProActive Cloud Automation multi-IaaS connector.

Exhibit 5 shows the interconnection between the OCCIware Studio and the Runtime.

The OCCIware Runtime architecture has been designed to be modular and extensible, by authorizing the combination of different “OCCI Backends” behind a generic “OCCI Frontend”.

Exhibits 6 and 7 illustrate the modular architecture of the OCCIware runtime.

This modular architecture can be easily integrated because it is independent from any programming language.

The OCCIware’s erocci-dbus-java library leverages the API of D-Bus, available on Linux as a standard, which makes possible the extension to any language.

Exhibit 6: OCCIware Runtime Modular and Extensible Architecture
The OCCIware Runtime is compliant with the OCCI standard and is operational, tested and distributed through continuous integration. A first interface is available, and enables interactions with resources within a real-time distributed database (mnesia).

**Interoperable Deployment Services**

OCCIware provides a Java framework allowing for interoperable deployment services. It is the perfect foundation for hybrid Clouds that simultaneously consume several public as well as private Clouds, and not only at infrastructure but also platform and software layers, one being built upon the other. As such, it is a key component of the OCCIware Cloud consumer platform (in opposition to Cloud provider-specific platforms) and its ability to manage and hide complexity of the consumption of underlaying Clouds. It notably enables project partners to link the OCCI runtime and OCCIware use cases. Within this framework, the erocci-dbus-java library facilitates the integration of erocci with adopters’ Java runtimes, but also with either built-in or custom-developed Studio Java connectors. It interfaces with erocci through to the communication technology (IPC) D-Bus.

**OCCIware Use Case 1: Datacenter as a Service**

This use case is led by project partner Scalair. It aims at delivering an elastic Cloud/MultiCloud infrastructure with fast deployment and reconfiguration resources for hosted applications.

Scalair offers a complete platform (e.g., load balancer, fronts, BDD, etc.) that can adapt according to the incoming flow of visitors (load work). The use case offers platform developers a self-scalable system according to two predefined strategies. The first one concerns the management of horizontal and multi-platform scalability, i.e., the multiplication of virtual machines/containers on n-tier architectures with a transparent and autonomous load.
balancing for developers to absorb traffic spikes contingencies. Virtual machines/containers can be spread over several cloud platforms to increase availability level.

The second strategy is about managing vertical scalability (increase in existing resources) to address peak loads on virtual machines/containers.

These strategies are required in several application domains. One of them is carpooling: it can be highly stressed during weekends, making it necessary to have an appropriate and scalable infrastructure. In a typical cloud infrastructure, resources are provisioned in advance and often manually adjusted. During a load increase, resources are adjusted to meet the needs but without any anticipation and often at the explicit request of the client. The carpool use case is developed with data from Ecolutis/SNCF. This use case depends on the development of the elasticity manager that takes inputs from the Monitoring System and Roboconf.
As shown in Exhibit 8, the carpooled application involves specific components to ensure seamless application scalability:

- Elasticity Manager is able to monitor the inflows and the behavior of the different servers. The supervision of these elements enables the platform to anticipate request's overload and make decisions to increase or to add additional resources. The Elasticity Manager makes decisions based on Monitoring Manager outputs.

- Roboconf is able to provide hot (re)configuration of application nodes. For example, when a new node appears or disappears due to elasticity decisions (e.g. load-balancing, etc.), the configuration of dependent application nodes will be updated. For instance, when adding a new application server node, it will be automatically configured to connect to already existing databases, and load balancers will be reconfigured to redirect traffic to this new node. The process is asynchronous, and depends only on the application topology: for example, when autonomic decisions are made by ProActive cloud manager (it decides to add or remove a node), Roboconf performs the adequate application reconfiguration, so the global application still works.

- ProActive Cloud Automation is able to make decisions concerning elasticity (e.g., add or remove nodes), based on criteria like application load, latency, SLA, etc., using a decision-making algorithm. Then, it will be able to make these decisions operational, and change the application topology to adapt to the new situation.

**OCCIware Use Case 2: Deploy@OCCIware**

This is a demonstrator that focuses on deployment, live reconfiguration (with elasticity) and monitoring of complex enterprise applications on several IaaS and PaaS. The Deploy@OCCIware Use Case is also illustrated by Exhibit 8 above, it leverages the ProActive Cloud Automation platform and Roboconf deployment framework.

A typical OCCIware deployment archive contains at least:

- An OCCI description of IaaS resources (possibly PaaS with extended OCCI);

- An "extended OCCI" description of the application (SaaS);

- If necessary, lifecycle management scripts and resources related to the application itself (deployment archives, binaries, configuration files, etc.).

It can also integrate "extended OCCI" descriptors related to various nonfunctional properties, such as elasticity, the SLA or vertical scalability. Deploy@OCCIware runtime can deploy such an archive via OCCI, in an interoperable manner across multiple platforms and IaaS/PaaS targets.
The use-case demonstrates the implementation of such a deployment via OCCI:

- Through Roboconf or ProActive Cloud Automation to deploy on hybrid IaaS or PaaS;
- By Roboconf on ProActive Cloud Automation (seen as an IaaS or PaaS);
- For ProActive Cloud Automation via Roboconf (seen as a deployment system).

The use-case also demonstrates hot reconfiguration, with or without automatic elasticity (e.g., as part of an enterprise web application or e-commerce, web front-end addition / deletion with load balancing to adapt to the load) and monitoring of the platform. It will be based on one or more representative demonstration applications of conventional software architectures (e.g., application servers like JEE or LAMP) and the current use of components in the world of enterprise distributed applications (e.g., Apache, Tomcat, MongoDB, MySQL, Node.JS, ElasticSearch, etc.).

**OCCLware Use Case 3: Big Data as a Service**

This use case led by project partner ActiveEon provides an example of how OCCI can be used to facilitate the configuration and deployment of Big Data and HPC services. ActiveEon proposes a service catalog based on its ProActive Cloud Automation to deploy any Big Data platform. The Big Data platform is deployed using the ProActive Workflows and Scheduling to auto-manage cloud infrastructure resources needed for the platform with the ability to dynamically adjust the amount of compute nodes according to load.

Exhibit 9: Big Data as a Service Use Case

(cc by) OW2 2016. This white paper is distributed for purposes of comment and discussion only.
More specifically, the OCCIware system queries the status of the ProActive platform using the ProActive Scheduling API and to determine, according to predefined rules, whether it is necessary to increase the size of the platform or reduce it (with all relevant configurations at the application level). The OCCI model provides the necessary elements to describe the use cases: initial amount of VMs, the initial VMs configuration, stock triggering rules, scale up and scale down activities, adjusted monitoring. OCCIware makes possible the automation of the Spark platform deployment using the ProActive Workflows and Resource Manager and the execution of multiple Spark streaming applications.

The deployment workflows of ProActive Cloud Automation (PCA) are defined and implemented in the PCA service catalog. The Spark service template is made available and accessible via PCA portal. The OCCI extension of BDaaS resources definition has been started recently. Once the extension is defined, the PCA BDaaS service will be executed by the OCCI BDaaS extension execution.

**OCCIware Use Case 4: Linked Data as a Service**

Led by project partner Pole Numerique, this use case aims at demonstrating the Open Linked Data Broker in data- and IoT-based applications in the field of energy and telecommunications. The work done since 2015 has been showcased in public events on several occasions. It is based on the Ozwillo platform developed by the OASIS project (Openly Accessible Services for an Interactive Society) for the creation of public common reusable data sets through the use of defined set of applications provided to citizens and public organizations.

![Exhibit 10: Linked Data as a Service Use Case](image-url)
The "Linked Data as a Service" demonstrator relies on OC
dIware for configuration modeling
and for the development of tools and interfaces required by service providers to provide
Ozwillo data on demand, in the same way that any other Cloud resource can be provided on
demand.

As illustrated in Exhibit 10, this use case demonstrates OC
dIware ability to expand and
implement the OCCI standard to manage the delivery and consumption of services,
particularly data service, in all its dimensions (type and business specific, read and write
quality of service, authorizations but also tooled configuration and governance of the whole).

The OC
dIware framework applied to OASIS drastically simplifies the process and reduces
the time required for creating new applications. It also enables open provision of OASIS data
and services to all stakeholders.

The next step is for the project to deliver an OCCI-based use case platform and linked-data
resource model, integrated with energy provider's solution and data.
OCChware Outlook

The OCChware studio will be complemented step by step with new connectors to additional cloud platforms, and towards an OCChware product line of domain-specific studios for OCCh, Docker, OpenStack, VMware, Roboconf, etc.

The OCChware innovative technology is the starting point for future business opportunities such as:

- Creation of an Inria's start-up on model-driven cloud computing
- Integration of OCChware Studio into Obeo’s Smart EA product
- Design and development of a “Docker as a Service“ offer
- Design and development of a “Big Data as a Service” offer
- Design and development of a “Linked Data as a Service” offer
Conclusion

Proof of concept has been achieved. The project partners will soon be ready announcing the full and complete OCCIware model-driven tool chain. It will include the development environment, an Eclipse-based “Studio”, dedicated domain-oriented languages, the generators and compilers, and the execution environment (the “erocci” runtime core and its backends).

Industrial project partners are generating and testing the first OCCI experimentations on their use cases, they will be able to validate the OCCI resources model for their needs. The studio is being enhanced with additional generators and connectors, with backend generation capabilities, and with the integration of a simulator and a decision support tool. The work on the OCCIware runtime is focused on completing the eroccidbus-java bridge, and on using it to integrate all backends behind erocci. Use cases will build further on the technologies delivered by OCCIware at the design, development, deployment and monitoring levels.

Thanks to the OCCIware unified approach and using the extensible Open Cloud Computing Interface (OCCI) REST API, developers will be able to configure at once an application in the Cloud with its VMs, its database, its web server, its monitoring and even, for instance, its customer accounts.

All OCCIware components are developed collaboratively by the project partners and are open source licensed. An open source project itself, OCCIware is open to third-party contributions to help the platform grow in functionalities and make the transition to a fully marketable product or service.

Useful Links

To learn more about the OCCIware, please visit: http://www.occiware.org

Access to open-source code:
OCCIware Repository: https://github.com/occiware
OCCIware Studio: https://github.com/occiware/ecore/tree/master/clouddesigner
OCCIware Runtime: http://erocci.ow2.org

Project deliverables: http://www.occiware.org/bin/view/Project_Deliverables/WebHome

Project factsheet: http://www.occiware.org/bin/download/Share_Collateral/OCCIware_FactSheet/OCCIware_FactSheet_20150319.pdf
The OCCIware contributors are

Open Wide

OW2

ActiveEon

Grenoble Alpes University

Inria

Obeo

Scalair

Linagora

Telecom Sud Paris

The OCCIware project is funded by the French Fond National pour la Société Numérique (FSN).