# Designing resource-aware distriubted system based on system level containers

Inti Gonzalez-Herrera, Johann Bourcier, Olivier Barais, François Fouquet

# Designing resource-aware distriubted system based on system level containers

Inti Gonzalez-Herrera[1], Johann Bourcier[1], Olivier Barais[1], and François Fouquet[2]

[1] University of Rennes 1 / IRISA / INRIA firstname.name@irisa.fr
[2] University of Luxembourg / SnT francois.fouquet@uni.lu

**Abstract.** Resource management is critical for application domains where components share their execution environments but belong to different stakeholders, such as smart homes or cloud systems. Yet, current middleware and application containers often hide system-level details needed for dynamic resource management. In particular, they tend to hide resource usage by offering automatic management of these resources (e.g., CPU, memory and I/O). In contrast, system-level containers, such as *Linux Containers* (LXC), allow fine-grain resource management. However, they lack knowledge about the application's structure and its requirements in order to provide fine tuned resource management.
In this tutorial, we will expose Squirrel: a new middleware which aims at combining the benefits from the component based software engineering to design flexible and modular application and the system level containers to manage resources. Squirrel follows an approach where developers specifies contracts on components and connections to describe the expected behavior of their application regarding resource consumption. These high level contracts are then used to automatically configure the system level containers which will hosts the running applications. At the end of this tutorial, applicants will be able to design applications and contracts using Squirrel and run their application inside system level containers to ensure a correct behavior of their application regarding resource consumption.

## 1 Presenter

**Inti Gonzalez-Herrera** is a Ph.D. Student at the University of Rennes 1, where he is a member of the DiverSE INRIA research team. He received his M.Sc. degree in 2010 from the University of Las Villas, Cuba. The topic of his Ph.D. is about providing new mechanisms to guarantee resource reservation in Java-based pervasive middleware. In a broad sense, his research interests include the use of software engineering and system programming to guarantee non-functional properties. Inti Gonzalez-Herrera has co-authored 5 international peer-reviewed conference and workshop papers in the fields of Computer Science and Applied Computing.

**François Fouquet** holds a Master degree in Software Engineering at the University of Rennes 1 in 2009. He obtained his PhD thesis from the Triskell research group in Rennes in 2013. He currently holds a postdoc position at Interdisciplinary Center for Security Reliability and Trust (SnT) in Luxembourg. His topics of interest include software engineering, modeling at design and runtime, and distributed systems. During his PhD work, he notably developed Kevoree. François Fouquet has co-authored 4 international peer-reviewed conference papers and 1 peer-reviewed journal article and several workshop papers.

**Johann Bourcier** is an Associate Professor at the University of Rennes 1, where he is a member of the Triskell INRIA research team. He received his Ph.D. degree in 2008 from the University of Grenoble 1. He is a former member

of the LIG Adele research group (Grenoble) and later the Distributed Software Engineering Section in the Department of Computing of Imperial College London. His research interests include the use of software engineering to simplify the development of highly dynamic pervasive applications. Johann Bourcier has co-authored 18 international peer-reviewed conference, journal and book chapters in venue such as TAAS, GECCO, SEAMS, WICSA, SCC,

**Olivier Barais** (http://goo.gl/fQOF7) is an Associate Professor at the University of Rennes 1, member of the Triskell INRIA research team. He received an engineering degree from the Ecole des Mines de Douai, France in 2002 and a PhD in computer science from the University of Lille 1, France in 2005. After having been a PhD student in the Jacquard INRIA research team, he is currently associate professor at University of Rennes 1 and a member of the Triskell INRIA group. His research interests include Component Based Software Design, Model-Driven Engineering and Aspect Oriented Modeling. Olivier Barais has co-authored 8 journals, 36 international conference papers, 2 book chapters and 26 workshop papers in conferences and journals such as SoSyM, IEEE Computer, ICSE, MoDELS, SPLC and CBSE.

## 2 Duration and Room Requirements

The workshop duration is half a day (4 hours). The type of tutorial will be a hands-on tutorial. This tutorial requires a room with wireless access for participants. We expect between 8 and 20 participants.

## 3 Scope

The intended audience is both software engineers/researchers and PhD students. The attendants must have a laptop with Virtual Box installed and we will provide a dedicated virtual machine to have a working environment. The attendants must be familiar with Java or any Object Oriented Technology. Academics and practitioners alike will benefit from the tutorial. We expect that the presentation is of particular interest for designers of distributed tools and algorithms, both academics and practitioners, who would like to benefit from an abstraction to finely control with resource management.

## 4 Goal and Objectives

The goal of this tutorial is to provide the fundamentals of our Squirel framework which mainly provide a way to perform resource reservation on a modern component based framework named Kevoree [1][3]. Kevoree is a framework language for designing heterogeneous and distributed adaptive applications. Squirel [2] also relies on system level container such as LXC to ensure the resource reservation during the system runtime. The goal of this tutorial is to present Squirel through a practical session to a wide number of Software Engineering practitioners. This event would be a great venue for people to learn-by-example about resource aware development and deployement, in a modern component based framework. This tutorial is also intended to be a privileged moment to collect comments and feedback on our approach and realizations. This overall goal can be broken down into several concrete sub-objectives:

- Component Model. Use the underlying framework (Kevoree) to understand and manipulate the component concepts.
- Deployment on (single & multiple) node. Experiment the simple deployment on single vs multiple execution nodes.
- Develop resource-aware contracts for component based system.
- Run a component-based system with resource reservation feature.

## 5 Overview of the tutorial

The tutorial is divided into 4 main sections

### 5.1 Part 0 - Introduction (20 min talk - 20 min Prerequisites)

**Presentation** This first part introduces Component based software engineering and the Kevoree Framework. After a bit of history, some facts, and examples of realizations, the participants prepare their machines with the necessary tools for the following parts. **Practical session** The practical session of this part ensures that the virtual image is working on each participant's machine.

### 5.2 Part 1 - The basics (20 min talk - 30 min hands-on)

**Presentation** This second part presents the basics on how to design, assemble an deploy an component based application on the Kevoree Framework. **Practical session** The practical session of this part will enables the design and deployment of a system composed of various components.

### 5.3 Part 2 - Design your contract (20 min talk - 30 min hands-on)

**Presentation** This part introduces the Squirell framework and its languages to support the expression of contracts on resources usage. **Practical session** The practical session of this part will involve the design of several contracts to enable resource reservation for each component of the system.

### 5.4 Part 3 - Run your application with resource management (15 min talks - 30 min hands-on)

**Presentation** This final part presents the runtime environment of the Squirell framework to support resource reservation on the running system. **Practical session** The practical session of this part will be to set up the runtime environment of Squirell and deploy the applications with the ressource reservation feature.

## References


1. F. Fouquet, B. Morin, F. Fleurey, O. Barais, N. Plouzeau, and J.-M. Jezequel. A dynamic component model for cyber physical systems. In *Proceedings of the 15th ACM SIGSOFT Symposium on Component Based Software Engineering*, CBSE '12, pages 135–144, New York, NY, USA, 2012. ACM.
2. I. Gonzalez-Herrera, J. Bourcier, E. Daubert, W. Rudametkin, O. Barais, F. Fouquet, and J.-M. Jézéquel. Scapegoat: an Adaptive monitoring framework for Component-based systems. In A. Tang, editor, *Working IEEE/IFIP Conference on Software Architecture, WICSA 2014*, Syndey, Australie, 2014. IEEE/IFIP.
3. B. Morin, O. Barais, G. Nain, and J.-M. Jezequel. Taming dynamically adaptive systems using models and aspects. In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, pages 122–132, Washington, DC, USA, 2009. IEEE Computer Society.