



PaaSage

Model Based Cloud Platform Upperware

Deliverable D1.7.3.

User Interface Design and Development

Version: 1.0

PROJECT DELIVERABLE

Name, title and organisation of the scientific representative of the project's coordinator: Mr Philippe Rohou Tel: +33 4 9715 5306 Fax: +33 4 92385011 E-mail: <u>philippe.rohou@ercim.eu</u>

Project website address: <u>http://www.paasage.eu</u>

Project	
Grant Agreement number	317715
Project acronym:	PaaSage
Project title:	Model Based Cloud Platform Upperware
Funding Scheme:	Integrated Project
Date of latest version of Annex I against which the assessment will be made:	03-07-2014
Document	
Period covered:	M25-36
Deliverable number:	D1.7.3.
Deliverable title	WP1 PaaSage User Interface Design and Development
Contractual Date of Delivery:	30 September 2015
Actual Date of Delivery:	02 October 2015
Editor (s):	Tom Kirkham (STFC), Franky Vanraes (be.wan)
Author (s):	Tom Kirkham (STFC), Franky Vanraes (be.wan)
Reviewer (s):	Kamil Figiela (AGH), Philippe Massonet (CETIC)
Participant(s):	
Work package no.:	WP1
Work package title:	Technical Foundation
Work package leader:	STFC
Distribution:	PU
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	22

DISCLAIMER

This document contains description of the PaaSage project work and findings.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the PaaSage consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 27 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu.int/)



PaaSage is a project funded in part by the European Union.

History

Date	Version	Author	Description			
16/3/2015	0	Franky Vanraes	Setup structure & template			
17/4/2015	1	Franky Vanraes	Franky Vanraes Initial Inventory			
19/8/2015	2	Tom Kirkham Added policy editor				
21/8/2015	3	Franky Vanraes	Added Executionware GUI			
22/9/2015	4	Tom Kirkham	TK Final Edits before Internal Review			
24/9/2015	5	Franky Vanraes	Final Checks before Internal Review			
02/10/2015	6	Tom Kirkham	Changes following internal review			

Table of contents

1	Exe	ecutive Summary	6
2	Intr	oduction	6
	2.1	PaaSage Lifecycle	6
	2.2	PaaSage End Users	7
	2.3	CAMEL	8
3	Cor	nfiguration Interfaces	9
	3.1	CAMEL tree-based editor	9
	3.2	CAMEL textual editor	.10
	3.3	Policy Editor	.11
4	Dep	ployment Interfaces	.12
	4.1	Social Network	.13
5	Exe	ecution Interfaces	.18
	5.1	Colosseum Execution Management	.18
6	Sur	nmary / Discussion	22

1 Executive Summary

PaaSage presents business users with the ability to specify business requirements for the deployment of applications on multiple Cloud platforms. In order to deliver this simplicity the project uses a model driven approach where technical models are designed by specialist users for modification by all classes of end user. The creation of specialist user interfaces enable users to specify, monitor and re-configure Cloud application deployment requirements.

This document describes these interfaces and how they are applied to support various users in the PaaSage lifecycle. Evaluation of the interfaces is conducted in the evaluation deliverable at the end of the project with focus here being given to the interfaces position in the wider PaaSage technical architecture. At the time of writing at the end of year 3 the selection of user interfaces present low level interfaces to specific model design through to high level business orientated design tools.

2 Introduction

This deliverable will describe each user interface developed for the project and its position in the PaaSage architecture. Specific focus will be given to its core function and implementation. In this first section we set the context that these descriptions are best viewed in, particularly with respect to the PaaSage lifecycle and end user groups.

2.1 PaaSage Lifecycle

The PaaSage project presents a lifecycle to Cloud deployments. Central to this is the configuration, deployment and execution of models. User interfaces at each stage of the lifecycle aid the end user in the utilisation of the PaaSage platform to best suit his or her business needs.



Figure 1 PaaSage Lifecycle

Each component in the PaaSage lifecycle processes models defined for use in the PaaSage platform. As the models pass through components they evolve and this process is the PaaSage lifecycle the stages of which are illustrated in Figure 1. These models are expressed in CAMEL and this standard is the focus for the initial user interfaces dedicated to configuring application requirements.

2.2 PaaSage End Users

In the wider vision of PaaSage we expect the platform to be deployed to suit a variety of business contexts. These include singe private instances within an organisation to cross organisation instances covering a specific application domain. Within all deployments the platform will provide the same core user interfaces described in this deliverable. End user groups using these interfaces have been split into three categories.

Systems Administrators: These end users are responsible for the host network(s) where the PaaSage installation resides. Key concerns for system administrators are the integration of PaaSage with existing systems for core business function such as identity management. The Systems Administrator will also through his or her role have a good idea of organisation policy and can use this to add detail to CAMEL models relevant to the organisation.

DevOp: The DevOp user's main role is to specialise on the PaaSage platform. He or she will have the responsibility in ensuring that relevant CAMEL models are developed and presented to the end user organisation. Also the DevOp will be responsible for the core performance and stability of the PaaSage deployment. As a result they will monitor specific components during execution with the goal to supply specific support in case of error.

Business User: This user fulfils the role of the non-technical user of the platform. PaaSage is designed to enable business users to embrace the potential of Cloud computing technology with minimal skills or knowledge needed on how to deploy and manage applications in the Cloud. It is therefore expected that the Business User will have minimal interaction with the detailed technical aspects of PaaSage which will be achieved by the Dev Op and Systems Administrator user. However, it is expected that the business user will set business goals which will form service level objectives for the platform and be the person who starts and owns the deployment.

User interfaces that the users interact will be explained in the next section of the document, but as an overview the diagram below shows the main PaaSage interfaces and their likely end users.



Figure 2: Main UIs and End Users

As figure 2 illustrates the main technical functions related to CAMEL, Policy and Execution configuration are left to the Systems Administrator and DevOp user groups. The Business User is only expected to interact with the Social Network as we expect most people to be familiar with common Social Network functions.

2.3 CAMEL

Finally, before we go into detail of each user interface at this point it is worth recapping over the role of CAMEL in PaaSage. Cloud Application Modelling and Execution Language (CAMEL) is a family of domain-specific languages (DSLs) to define models for the PaaSage platform. The current CAMEL DSLs family includes CloudML, Saloon, Scalability Rules Language (SRL) and CERIF. The use of DSL enables CAMEL to embrace existing communities for the specification of deployment and specific details including constraints associated with them. This community engagement (as in the case of XACML) yields existing tools and user interfaces that have been incorporated in the PaaSage project for use.

The CAMEL model presents a hierarchical form of the DSLs, with the branches of each DSL to specify the characteristics of requirements or constraints. In the context of the PaaSage process, the user defines the CAMEL model in order to specify the rules and constraints about the deployment and execution of the application. The CAMEL model specification is achieved through the CAMEL Editor, which is an implementation based on the Eclipse platform. Thus, the CAMEL Editor is the

starting point at which the user engages with the PaaSage process and where the exploration of the PaaSage user interfaces begins.

After CAMEL models are created they are then deployed (by the Business User) and monitored in terms of execution by the more technical end users. CAMEL provides the glue in the project which not only evolves during the PaaSage lifecycle phase but also is the means by which most data captured and expressed in the user interfaces is contained.

3 Configuration Interfaces

As discussed this document describes each set of interface per PaaSage lifecycle phase. Configuration is the first PaaSage phase and is the point at which the process of CAMEL model creation or modification happens to suit the end users specific application deployment requirements. These requirements are expressed in CAMEL and WP2 is responsible for the CAMEL modelling user interfaces. Currently, there are two user interfaces addressing CAMEL editing:

- The tree-based editor
- The textual editor

Configuration is also handled by the Social Network via the Policy Editor. Using the Policy Editor user can specify security constraints that are applied to existing models before deployment.

3.1 CAMEL tree-based editor

3.1.1 Purpose

Supporting and enabling users to capture their application model in CAMEL. While not as convenient as text-based editor, this editor was created at early stage of CAMEL development to enable early adopters to test and validate CAMEL functionality.

3.1.2 Type of User Interface

The CAMEL tree-based editor runs into the Eclipse IDE. Eclipse is a common open source <u>desktop based</u> Integrated Development Environment. Eclipse offers a <u>graphical</u> <u>user interface</u> for modelling applications and integrates the EMF (Eclipse Modelling Framework) facilitating the definition of DSLs (Domain Specific Language) based on the Ecore metamodel. The CAMEL metamodel was build using EMF and provides the CAMEL Ecore model which is then imported in Eclipse in order to model specific applications.

The CAMEL tree-based editor is compatible with Windows, OS-X and GNU-Linux versions of the Eclipse IDE.

3.1.3 For which role/type of actor the CAMEL tree-based editor is intended

The CAMEL tree-based editor is intended for :

• DevOps with specific knowledge of the end application to be executed on the PaaSage platform.

3.1.4 PaaSage Functions provided by the CAMEL tree-based editor

The editor enables the end user to specify models for deployment on the PaaSage platform

3.1.5 How to reach the CAMEL tree-based editor

The CAMEL tree-based editor is reached from a locally installed Eclipse IDE, after importing the Ecore metamodel.

Documentation, tutorial and installation guidelines can be found in the CAMEL Git repository of PaaSage hosted by OW2.

3.1.6 Current - Future status

The tree-based editor is currently in a stable prototype status.

3.1.7 Examples – Screenshots

Java EE - C:\Bewan\PaaSage\Work	Workspaces\CAMEL_Luna\be.wan.camel.ecore\model\BeWAN2.xm	ii - Eclipse	↔ <u> </u>
File Edit Navigate Search Proj	ect Sample Reflective Editor Run Window Help		
	N 3. 9. R = 7 4 - 0 - 4 - 7 - 6 -	109 (c) x? + (Q) 22 (5 + N) + N	i ← + → +
			Quick Access
Project Explorer 20 🖷 🗈 E 😒 🗊 🌣	BeWAN2.xmi BeWAN		Quick Access
	Name	ReWAN Application	
	Owner	Cloud Provider BeWAN	
	Version	** v1.0	
Selected Object: Application BeWAN A	Application		

Figure 3 CAMEL Tree Based Editor

3.2 CAMEL textual editor

3.2.1 Purpose

This interface to CAMEL was developed to provide another means to specify and edit CAMEL models. In this editor specification of CAMEL models can be done using concrete textual syntax, and their serialisation as XMI files. The interface is suited to the technical user with a knowledge of CAMEL such as the DevOp.

3.2.2 Type of User Interface

The CAMEL textual editor runs under Eclipse-Xtext. Xtext is a framework for development of programming languages and domain specific languages. Xtext provides among other capabilities, full text editing, syntax coloring, content assistance, validation and fixing. Eclipse is an Integrated Development Environment running on a local desktop. (see also 4.1.2).

The CAMEL textual editor is compatible with Windows, OS-X and GNU-Linux versions of the Eclipse IDE.

3.2.3 For which role/type of actor the CAMEL textual editor is intended

• DevOps with specific knowledge of the end application to be executed on the PaaSage platform.

3.2.4 PaaSage Functions provided by the CAMEL textual editor

The editor enables the end user to specify models for deployment on the PaaSage platform

3.2.5 How to reach the CAMEL textual editor

The CAMEL textual editor is reached from a locally installed Eclipse IDE with the Xtext components installed.

Documentation, tutorial and installation guidelines can be found in the PaaSage Git repository hosted by OW2 under the CAMEL module.

3.2.6 Current - Future status

A first prototype is available on the projects OW2 GitHub.

3.3 Policy Editor

Purpose

The Social Network allows users to select pre-configured models for deployment. In addition to this it supports a security policy editor that can be used to edit existing models to configure additional security constraints. This is of significant use for the Systems Administrator user who can use this interface to ensure data and applications are executed using CAMEL models that take into account organisational constraints expressed as policy.

3.3.1 Type of User Interface

The policy editor is a web application

3.3.2 For which role/type of actor the Policy editor is intended

System Administrators will add their organization knowledge with respect to security and privacy of application and data access into the Policy Editor for inclusion in the CAMEL models

3.3.3 PaaSage Functions provided by the Policy editor

The policy editor enables XACML policies to be created. These link specific user roles to data objects or application endpoints. Hence, they can be used to restrict access to specific functionality and data. The editor links to a library that adds this XACML into CAMEL as a series of constraints for the model.

3.3.4 How to reach the CAMEL textual editor

Current demo implementation is here <u>http://kirkhac3.miniserver.com/ponte-</u> site/index.php

3.3.5 Current - Future status

The policy editor was created for the EU Framework 7 project Ponte. We have modified it to suit PaaSage user requirements and also integrated the policy creation to enable CAMEL modification / addition of new constraints at the same time. The interface to add more complexity and as CAMEL develops we expect additional constraint functionality to be developed which we can add to the UI. It can be found in the projects OW2 Git Hub under Identity Management.

3.3.6 Examples – Screenshots

Home	Users	Actions	Available actions	Roles	Assign Role	СТР	> Log
Home » /	Assign Rol	<u>es</u> » Manage	e				
Man	age	Assig	n Roles				
You may	ontionall	v enter a co	mparison operator (< <= >	>= <> or =) at	the hea	eninning of each of your search values to specify how the comparison sho
be done.	optional	y enter a ce	inpunson operator (, 4 01 <i>f</i> at	the beg	symming of each of your search values to specify new the companion shoe
	Usernam	e Select a	an username 🔻				
	Del						
	KO	e Select a	arole 🔻				
	Ctpi	d Select a	CTP V				
		Search					

Figure 4 Policy Editor

4 Deployment Interfaces

Within PaaSage deployment is the process where a model is passed to components which analyse its contents to find the most optimal infrastructure to which it can be ran on. At this phase the user requires support in selecting the right model for his / her application if no configuration is required or the ability to select their defined model and deploy it to the PaaSage platform in a simple way. Feedback of the deployment process is also needed to ensure the user is happy with the process that can be repeated if this is not the case.

To support the user in simple deployment and analysis of deployment results PaaSage uses the Social Network as its user interface for model deployments. Instances of the Social Network can be limited to individual companies, departments or across industries. The benefit of the social network as a point of deployment is that it is a point at which knowledge can be shared regarding experiences with the PaaSage platform.

4.1 Social Network

4.1.1 Purpose of the User Interface

The Social interface provides a simple way in which non-technical users can select a model and deploy it. If a user has questions regarding the deployment he or she can seek support either from other users or captured and shared knowledge present on the social network.

For technical users the Social Network enables the same type of knowledge sharing but on a deeper technical level. As a centre point for the project the Social Network can support single sign on and provide a hub focal point for the provision of formal and informal documentation on the use of PaaSage.

4.1.2 Type of User Interface

As a web based application the Social Network embraced familiar features associated with web based social network interfaces. To make the deployment simple we have created a feature termed the Big Green Button. Here the user selects a model and presses the big green button to deploy.

4.1.3 For which role/type of actor the UI is intended

As the name 'social network' suggests, the UI is intended for a wide audience. Users can have dedicated roles, but can also be anyone interested in cloud application modelling and execution.

The Social Network is the point at which all end users converge. It is a point by which high level data can be viewed by technical users on specific model availability and the point at which application execution can be started and controlled by the business user.

4.1.4 PaaSage Functions provided by the UI

The functionalities of the Social Network UI are not directly interacting with other components of PaaSage (modelling, profiler, reasoning, adapter, execution and monitoring). The information entered via the UI is stored into the MDDB, and the information retrieved via the UI is also provided by the MDDB. The other PaaSage components are reading the MDDB and also feeding the MDDB with for instance execution history of an application or information about cloud providers.

After login the UI provides as main functionality :

- Access to the user profile, personal messages and settings
 - Personal dashboard with
 - o Activity on users models & components
 - o Feeds
 - o Personal information
 - o Discussion groups
 - o Models
 - o Overview of the application models currently loaded in the MDDB
 - Possibility to "use" the model

- Possibility to "run" the model
- Mark the model for "follow"
- Filter the overview on criteria of interest, requirements
- "used" models are added to the shopping cart
- Components
 - Overview of the component models currently available in the MDDB
 - Select a category (apps, DB servers, OS, Web Servers, ...)
 - Possibility to "use" the component
 - Filter the overview on the modelling framework
 - o "used" components are added to the shopping cart
- Community
 - Overview of the existing Groups
 - Possibility to join a group
 - Overview of the feeds and discussions per group
 - Possibility to create a new group and invite members
- Shopping cart
 - After selecting existing models and/or components the user can open big shortning part and generate the initial CAMEL model in 4 stores
 - his shopping cart and generate the initial CAMEL model in 4 steps
 - Select the components
 - Select where each component will be deployed and the type of VM
 - Select the connections between components
 - Select a name for the model
 - The new model is than added to the model overview, from where the initial model (xmi) file can be downloaded
- My area
 - An alternative to view the users configs, active runs, models, components, credentials, followed apps

4.1.5 How to reach the UI

From a web browser : <u>http://socialnetwork.paasage.eu/</u>

Users must login or signup for a new login.

4.1.6 Current - Future status of the UI

The prototype of the user interface (and the social network infrastructure) is relatively stable. The final version is under development and is expected by Month 36 in the PaaSage project plan. Work on the responsive design is also to be done in order to be able to use the UI on mobile devices.

4.1.7 Examples – Screenshots

Parales A Models Components Community	vanraesf]	Log in
	Sign up		
Duild application models that can yup an multiple Cloud	Display name		
platforms	Email address		
Build or migrate old applications. View application execution statistics. Manage resources and run applications on the Cloud.	Username		
Collaborate with a powerful community	Password		
Create your network. Find experts. Rate, review and discuss applications.	Password (again for verification)		
	Register		
Build and run application models			
Find analications or components, view their execution statistics and deploy them directly or u	ise them to create your own a	polication model	
Join the community			
Find groups and users, connect with them and participate in discussions			

Figure 5 Social Network: Front Page

PAAGE A Models Components	Community search		📜 🚠 🌲 🖾 🎂 🗠 🤸 Log out
SUMMARY OF ACTIVITY FOR YOUR MODELS	S AND COMPONENTS		Profile Completeness
Top Rated	Most Run	Statistics Overview	Your Profile is 70% complete
E be wan Web Blog Application	be.wan Web Blog Application	0 FRIENDS	
average rating, urb (u reviews)	runs. u	2 MODELS	What is your role in your Organization?
Service Management Application average rating: 5.00/5 (1 reviews)	Service Management Application runs: 0	0 COMPONENTS	
		0 QUESTIONS	Skills View All (5)
		0 REPLIES	analysis
			Business Intelligence
			Add a skill Add
Live Feed			Areas Of Interest View All (7)
[4]]			usability
be wan Web Blog Application			user experience
Al Innes (FLEX) is now a friend with Eve.	Austin (FLEX) 9 days ago		Add an Interest Add
Al Innes (FLEX) is now a friend with Darre	en 9 days ago		Popular Groups View All
			3 Scalability topics
Al Innes (FLEX) is now a friend with nhing	nins, 9 days ano		15 users
	g		
Al Innes (FLEX) joined the group Scalabil	lity topics 9 days ago		Java Developers
Al Innes (FLEX) joined the group Java De	evelopers 9 days ago		

Figure 6 Social Network: Dashboard

PARACE A Models Components I	Community	1) 🖉 🚠 🌲 🕷 🌾 🖻	🛛 🔸 Log out
Models					
Recommended for you					<u>Expand</u> ¥
Be. Wan Web Blog Application By Tany vames Create 2 Bays ago be wan Web Blog Application	MaSS dry negaric Created 15 days ago Manipulate and Stress System	Multi-Ther Ghost Blog by Jog Domatcha Create: 77 Bays ago A distributed blog application based on ghost and postgres as it will be used for teaching.	EBIORI: HIPLACHE: FALLED TO LOND HIML, ARTITy_Lacker:13 TestApp0 Arest application with 100 esecutions A test application with 100 esecutions	FILTERS Keyword Status © Deployed © May yed deployed	٩
0 uses 0 runs 0 watches ***	1 uses 0 runs 0 watches ***	3 uses 0 runs 3 watches ***	1 uses 100 runs 2 watches	Clouds @ All	
Milk Bank By Frode Finnes Larsen Created: 149 days apo The milk bank will simplify, increase traceability.	Flight Scheduling By Stefan Spathr (LSY) Created: 149 days ago NetLine/Sched supports all aspects of schedule	Quorum Corporate And Financial Management By processions Created: 150 days ago	Molecular Dynamics By Atmony Satello Created: 150 days ago Molecular Dynamics (MD) is a method used to	Amazon Flexiant Microsoft Azure Google Compute Cloud	
increase security and increase collaboration for the human milk banks. The milk bank collects	development and schedule management. It offers powerful and easy to use schedule visualization and	Quorum is a powerful software solution, ideal for organisations wishing to upscale and streamline their entity management and company secretarial	predict physical properties of chemical and biological substances. One field of current studies are	Geography All North America	
5 uses 0 runs 2 watches	7 uses 0 runs 2 watches	4 uses 0 runs 1 watches	2 uses 0 runs 0 watches	South America	
Service Management Application	S.00 2 144.7 2 20 ms JEnterprise By Antonia Papakaanoou Created: 150 days ago	SensApp By Alessandro Rossin Created: 150 days apo	0 (2.37 (2.20 ms) Scalarm By Michail Orzechowski Created: 154 days ago	🗎 Asia 🗎 Africa Australia	
by many varies channel channel to only a po This application parmits field-workers (i.e. technicians, maintenance engineers,) to receive service tasks on their mobile devices. The users	The SPEC/Enterprise/2010 benchmark is a full system benchmark that allows performance measurement and characterization of Java EE 5.0 servers and	SensApp is an open-source, service-oriented application for storing and exploiting large data sets collected from sensors and devices. It is designed	The Massively Seld-Scalable Platform for Data Faming: With Idusport for all phases of a data farming experiment; from design through	Tags If All Health Care	View All(2)
3 uses 0 runs 0 watches	12 uses 14 runs 1 watches	7 uses 0 runs 1 watches ***	7 uses 6 runs 6 watches	Arch	
☆ 4.50 ② 20 ms				Ацау	

Figure 7 Social Network: Example Models



Figure 8 Social Network: Components

PAATAGE 🕈 Models Components Co	mmunity search				jä 🐨 🛉 🖩 🤻 🖂	✤ Log out
Components > WebServers				Filters		
Category "WebServers" has 37 components						
				Keyword		Q
Yaws Created: 173 days ago Cicudity, Cher	Zend Created: 173 days ago CibodML, Chef	Zend-Server Created: 173 days ago Cloudlu, Cher	Zotonic Creater 173 days apo Cloudill, Chef	Modelli All Chef	ing Frameworks	
Hooder eta .	The second secon	TEDUCTED.	The second	CloudM		
400 uses 4K watches	400 uses 4K watches	400 uses 4K watches	400 uses 4K watches			
	☆ 4.5 99% 20 ms	☆ 4.5 99% 20 ms				
Vertx Created: 173 days ago CiccolfL, Chef.	Tomcat6apr Created: 173 days ago CloudML, Chef	Php-Fpm Created 173 days apo CloudML, Chef	Unicom-Ng Created: 173 days apo CloudML, Chef			
WebServers	WebServers	WebServers	WebServers			
400 uses 4K watches	400 uses 4K watches	400 uses 4K watches	400 uses 4K watches			
Puma Created: 173 days ago ChoutML Cher	Uwsgi Create: 173 deys ago ChootM. Chef	Railsbox Created: 173 days ago ChouML. Chef	Ssi Created: 173 days ago CloudML. Chef			
WebServers	WebServers	WebServers	WebServers			
400 uses 4K watches	400 uses 4K watches	400 uses 4K watches	400 uses 4K watches			
		us 1 2 3 4 Next -				

Figure 9 Social Network: Component view of Web Servers

A Models Components Community	🐚 🛃 🌲 📾 🍓 🖾 🤸 Log out
CONFIGURATIONS ACTIVE RUAS MY MODELS MY COMPONENTS CREDENTIALS APPS WATCHED	CREATE Application Model
Drafts	
Finals	
Take wan Veb Blog Application Tables Management Application average rating: 05 (0 reviews) average rating: 5005 (1 reviews)	

Figure 10 Social Network: My Area

🔊 👘 Models Components Commun	nity	search			ў 🖁 🕯	- -	🔸 Log ou
New Application Model							
Step 0 S	Step 1	Step 2	Step 3	Step 4			
You can create a new deployable app	plication model, either:				_		
From external resources, or							
Starting from an baseline model cre	eated automatically from	a list of components	5.				
Currently you have 4 component(s) in	n your cart.						
Note: You can extend the baseline mo	odel using EMF editor						
Import Model Create baseline model							

Figure 11 Social Network: Create a New Model

5 Execution Interfaces

Once deployment is complete the CAMEL model has evolved to contain details of infrastructure to which the application will be deployed on. If the business user is happy with this he or she can then execute the application via the Social Network. In this step the social network invokes specific execution interfaces which provide a real time reporting interface to the state of the application. This enables the technical end users to abort and redeploy applications based on low level execution monitoring.

Execution interfaces are standalone web based interfaces linked to the tools used to execute the application.

5.1 Colosseum Execution Management

5.1.1 Purpose

Using this interface end users can monitor the live execution of the deployed application in the Cloud. It is of particular use to Dev Ops to ensure that the platform is operating correctly and System Administrators to ensure the application is being executed in line with organisational goals. A link between the interface and the Social Network is something being considered for year 4 in order to give the Business User a view on Execution.

5.1.2 Type of User Interface

The Executionware UI is designed as a Web UI

5.1.3 For which role/type of actor the UI is intended

The Executionware UI is intended for the technical users (DevOp and System Admins). The metrics reported and also information related to the application state will be of a technical nature and outside of the expected knowledge of the business user.

5.1.4 PaaSage Functions provided by the UI

The UI acts as the front-end :

- To the executionware Frontend API, providing a CRUD to manage VM, image, application,... (see more on : https://github.com/dbaur/execWareFrontend/blob/master/documentation/api/R EADME.md);
- To the SRLEngine, providing a CRUD to manage Monitoring (composedMonitors, rawMonitors and monitorInstances);
- To KairosDB, displaying results (charts) from monitoring sensors (cpu_usage, memory_usage,...).

5.1.5 How to reach the UI

The prototype is accessible here :

http://154.52.32.50/executionwareUI

Login : john.doe@example.com

Password : admin

(refresh the url to <u>http://154.52.32.50/executionwareUI</u> after login error)

5.1.6 Current - Future status of the UI

Prototype developed and installed during year 3 of the project. At year 4 the interface is set to be tested and developed further.

5.1.7 Examples – Screenshots

	Login
9 you are not logged	
Email address	
john.doe@example.com	
Password	
Submit	

Figure 12: Colosseum: Front Page

				I	Logout
ip address	appl	ication			
	id	name		+	
application	2	my application	्	C.	×
applicationComponent	3	aaa	ত্	C.	×
instance	4	application	୍	C	×
component					

Figure 13Colosseum: Application View

ip address	component										
	id	name	init	preinstall	install	postinstall	pre Start	start	startDetection	stopDetection	ро
application	1	web application			install.sh			start.sh			
applicationComponent	2	database			install.sh			start.sh			
instance	3	mysqlProxy			install.sh			start.sh			
component	4	LoadBalancer			install.sh			start.sh			
cloud -											

Figure 14 Colosseum: Component View

						l	_ogout
ip address	cl	oud					
application 👻	id	nama	andpoint	ani		÷.	
application	1	amazon	https://amazon.de:9696	1	Q	ľ	×
applicationComponent	2	openstack	https://openstack.org:8774	2	Q	Ø	×
instance							
component							
cloud 🗸							
cloud							
hardware							
image							

Figure 15 Colosseum: Cloud View

									I	_ogout
ip address	V	irtualN	Mach	ine						
application -	id	name	cloud	image	hardware	location	cloudUuid		+	
cloud 🗸	1	VM One	1	3	3	4		Q	Ø	x
communication										
user -										
virtualMachine 👻										
virtualMachine										
virtualMachineTemplate										

Figure 16 Colosseum: VM View

virtualMachine : 2

cloud : 1

cloudHardware : 1

cloudImage : 2 cloudLocation : 1

name : frank_srl_couchbase_1

3 : 192.168.3.80

4 : 134.60.64.49



Figure 17 Colosseum: VM Execution View

list update delete

6 Summary / Discussion

This document has given a quick tour of the main user interfaces available on the PaaSage platform at the end of year 3. This has been set in the context of the PaaSage lifecycle see Figure 18 and main user groups.



Figure 18: User Interfaces Present in the PaaSage Lifecycle

Central to the delivery of PaaSage is the CAMEL format which in itself is highly technical and hard for non-technical users to embrace. Thus, a key focus of interface development in PaaSage is the Social Network that sits in the middle of the lifecycle integrating with all phases.

A key benefit of the Social Network is that it presents an interface using common Social Network functions and is therefore suitable for all classes of users. The expansion of the Social Network and its interfaces into the configuration of CAMEL and monitoring of application execution is a key focus in the final year of PaaSage.

As mentioned at the top of the document and evaluation of user interfaces will be conducted in the evaluation deliverable within WP1 due at the end of the project. The success of the projects continued development of interfaces is core to the establishment of the project vision of specify once and deploy anywhere for Clouds.