Analysis of the State of the Art and Definition of Requirements

D1.1.3 (a)

WP1 – Definition of Requirements, Conceptual Model and Reference Architecture and Integration

Dissemination Level: Public

Lead Editor: Christof Momm

27/07/2012

Status: Final

The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 258862
This is a public deliverable that is provided to the community under a Creative Commons Attribution 3.0 Unported License: http://creativecommons.org/licenses/by/3.0/

You are free:

  to Share — to copy, distribute and transmit the work
  to Remix — to adapt the work

Under the following conditions:

  Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

  Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

  Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

  Other Rights — In no way are any of the following rights affected by the license:

  Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;

  The author's moral rights;

  Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

  Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this Web page.

For a full description of the license legal terms, please refer to:

http://creativecommons.org/licenses/by/3.0/legalcode
Contributors:
Vasilios Andrikopoulos, USUTTT
Andrea Giessmann, SAP
Sergio Garcia Gomez, TID
Miguel Jiménez, UPM
François Exertier, BULL
Frederic Junker, HSG/SAP
Dinh Khoa Nguyen, ERISS
Rouven Krebs, SAP
Benoît Pelletier, BULL
Daniel Scheibli, SAP
Steve Strauch, USTUTT

Internal Reviewer(s):  Sergio Garcia Gomez, TID
                        Nico Kruber, ZIB

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Authors</th>
<th>Sections Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>04/06/12</td>
<td>Christof Momm (SAP)</td>
<td>Initial draft.</td>
</tr>
<tr>
<td>0.2</td>
<td>16/07/12</td>
<td>Christof Momm, Miguel Jiménez,</td>
<td>Added WP2-6 SotA summary + discussion of JElastic,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>François Exertier, Rouven Krebs,</td>
<td>CloudLogic, CompatibleOne</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Benoît Pelletier, Daniel Scheibli, Dinh Khoa Nguyen, Sergio Garcia Gomez</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>16/07/12</td>
<td>Frederic Junker (SAP)</td>
<td>Updated WP3 summary</td>
</tr>
<tr>
<td>0.4</td>
<td>17/07/12</td>
<td>Andrea Giessmann (SAP)</td>
<td>Added discussion on Cordys</td>
</tr>
<tr>
<td>0.5</td>
<td>17/07/12</td>
<td>Christof Momm (SAP)</td>
<td>Added intro + conclusion + executive summary.</td>
</tr>
<tr>
<td>0.6</td>
<td>17/07/12</td>
<td>Steve Strauch (SAP)</td>
<td>Added SotA update for WP7</td>
</tr>
<tr>
<td>0.7</td>
<td>17/07/12</td>
<td>Christof Momm (SAP)</td>
<td>Final editing</td>
</tr>
<tr>
<td>0.8</td>
<td>19/07/12</td>
<td>Sergio Garcia Gomez (TID)</td>
<td>Added details on WP 5 SotA</td>
</tr>
<tr>
<td>0.9</td>
<td>26/07/2012</td>
<td>Christof Momm, Andrea Giessmann</td>
<td>Considered review</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>(SAP)</td>
<td></td>
<td>comments</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>27/07/12</td>
<td>Christof Momm (SAP)</td>
<td>Final editing</td>
</tr>
</tbody>
</table>
Table of Contents

1. Executive Summary.................................................................................................................. 8
2. Update on PaaS State of the Art.............................................................................................. 9
   2.1.1. Evaluation Criteria (Recap) ............................................................................................. 9
   2.1.2. Evaluation Results ......................................................................................................... 10
   2.1.2.1. CompatibleOne ......................................................................................................... 10
   2.1.2.2. Cordys ...................................................................................................................... 12
   2.1.2.3. Jelastic ...................................................................................................................... 14
   2.1.2.4. CloudLogic ............................................................................................................. 16
3. WP-specific Update on State of the Art.................................................................................. 17
   3.1. WP 2 – Service Engineering and Lifecycle Management ............................................. 17
   3.2. WP 3 – Marketplace ......................................................................................................... 18
   3.3. WP 4 – Resource Deployment and Management ......................................................... 19
   3.4. WP 5 – Administration, Accounting, Monitoring and Analytics ............................. 20
   3.5. WP 6 – Native PaaS Technologies ................................................................................. 20
   3.6. WP 7 – Immigrant PaaS Technologies .......................................................................... 20
       3.6.1. Cloud-aware Database Engines ............................................................................... 21
       3.6.2. Cloud-aware Application Servers ........................................................................... 21
       3.6.3. Cloud-aware Composition Frameworks and Engines ........................................... 22
       3.6.4. Cloud-aware Integration Technologies .................................................................. 22
4. Conclusion............................................................................................................................... 23
5. References .............................................................................................................................. 24
List of Figures

Figure 1. Cordys Business Operations Platform .................................................................13
Figure 2: Sample environment design (topology + software) for an application in Jelastic....15
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CaaSt</td>
<td>Building the PaaS Cloud of the future</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure as a Service</td>
</tr>
<tr>
<td>ITIL</td>
<td>IT Infrastructure Library</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>PIC</td>
<td>Product Instance Component</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>REC</td>
<td>Runtime Execution Container</td>
</tr>
<tr>
<td>S&amp;A</td>
<td>Service and Applications (used synonym in this document)</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SotA</td>
<td>State of the Art</td>
</tr>
<tr>
<td>SB</td>
<td>Steering Board</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>WPL</td>
<td>Work Package Leader</td>
</tr>
<tr>
<td>XaaS</td>
<td>All (Native or Immigrant) APIs offered by a PaaS</td>
</tr>
</tbody>
</table>
Executive Summary

The deliverable D.1.1.3 is the third and final iteration of the “Analysis of the State of the Art and Definition of Requirements” document.

Following the approach of the first iteration, the deliverable was again divided into two parts in order to reflect the different nature and target audience of both parts: Part D1.1.3a provides an update on the State of the Art (this document), while the status as well as planning of the user and technical requirements are described in D1.1.3b. This eases the maintenance/evolution of both documents and better addresses the stakeholders for each document. The state-of-the-art part (this deliverable) mainly serves scientific and educational purposes, while we use the requirements part (D1.1.3b) is used to drive the development of features and their evaluation within the use cases.

D1.1.2a [1] already included a very extensive state of the art (SotA) analysis covering around 30 SaaS, PaaS and IaaS solutions. These specific evaluations were based on a comprehensive set of evaluation criteria we defined for each layer. Therefore, this final iteration first provides an update on the PaaS SotA only, which represents the most important related work for 4CaaSt. The analysis covers CompatibleOne, Cordys, Jelastic and CloudLogic.

After this, a brief summary of WP-specific SotA updates is given. Detail can be found in the corresponding scientific reports Dx.1.2. This report concludes with a brief summary of the major insights.
1. Update on PaaS State of the Art

This chapter provides a discussion of selected new PaaS offerings, which are strongly related to 4CaaSt. The evaluation is based on the PaaS evaluation criteria we defined in D1.1.2a [1]. The discussion covers the following PaaS or PaaS-related offerings:

- **CompatibleOne**: is an open source cloud services broker, i.e. a core cloud computing mechanism providing intermediation, aggregation and arbitration of cloud services.
- **Cordys**: A business platform that amongst others comprises application development and business process management PaaS offerings.
- **Jelastic**: A PaaS offering that allows developers to deploy standard java applications in the cloud without any change or adaptation of source code.
- **CloudLogic**: A company that offers consultancy services for migrating software/processes to existing SaaS applications like SalesForce.com, Google Apps etc.

1.1.1. Evaluation Criteria (Recap)

The following table summarises the evaluation criteria for PaaS offerings. For a full description of the criteria please refer to [1].

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock-in grade</td>
<td>The use of open specification and technologies is a key point to take into account when a developer chooses a PaaS solution. 4CaaSt will leverage open APIs and platforms so service developers are not constrained to particular solutions (unless they chose them explicitly). The output of this criterion will be an evaluation of the lock-in risks for the corresponding PaaS platform.</td>
</tr>
<tr>
<td>APIs</td>
<td>Usually, each PaaS solution offers one or several APIs with a set of available functionalities that can be used in a programmatic way. 4CaaSt will offer a set of standard APIs with advanced functionalities to enable elastic service management. The output of this criterion will be a summary of the offered APIs with their relevant functionalities. Another question more related to the previous criterion is if these APIs are standards or not.</td>
</tr>
<tr>
<td>Service Definition Language</td>
<td>The language or format used for the description of a service or application in the PaaS platform. 4CaaSt will provide a complete and easy way to describe a service or application through its blueprint. The output of this criterion will be a short description of the language used and the information included with it for each PaaS platform.</td>
</tr>
<tr>
<td>Service Catalogue</td>
<td>The existence of a service/application catalogue in the platform. Our vision is that Clouds will naturally evolve to become worldwide catalogues/repositories of SaaS applications so 4CaaSt will provide an eMarketPlace in which service providers can expose, share and combine services and applications to create new ones and customers are able to find and consume products based on the offers of the eMarketPlace. The output of this criterion will be “available” or “not available”.</td>
</tr>
<tr>
<td>Service Composition</td>
<td>The service composition capabilities of the PaaS platform. Usually, existing marketplaces focus on the trading of standalone services that do not support service composition. 4CaaSt platform will provide service composition capabilities to allow the reuse of components. The output of this criterion will be “supported” or “unsupported”.</td>
</tr>
<tr>
<td>Scalability and multi-tenancy capabilities</td>
<td>The scalability levels (horizontal/vertical) and multi-tenancy capabilities provided by the PaaS Platform. Multi-tenancy, elasticity and scalability are key functionalities in a 4CaaSt-Cloud environment required to offer the promised automated scalability. The output of this criterion will be a summary of the scalability and multi-tenancy capabilities of the corresponding PaaS platform.</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>The use of specific programming languages inside the PaaS platform. If applicable, the output of this criterion will be the list of specific programming languages used in the PaaS platform.</td>
</tr>
<tr>
<td>Inter-component communication protocols</td>
<td>The protocols used among components within the PaaS platform (e.g. HTTP-based WS or REST, RMI, any, etc.). The output of this criterion will be the list of inter-component communication protocols.</td>
</tr>
<tr>
<td>Networking elements</td>
<td>The possibility of setting up auxiliary runtime artefacts like Load Balancers, Firewalls, Routers, etc. 4CaaSt will transparently create and manage all these kind of artefacts. The output of this criterion will be a list of available networking elements managed by the PaaS platform.</td>
</tr>
<tr>
<td>Cloud Database Service</td>
<td>The existence of a Cloud Database service in the corresponding PaaS platform. 4CaaSt will provide functions to access Cloud Databases and data stores. The output of this criterion will be a brief description of the available Cloud Database service (e.g. SQL or not, structure, etc.) if any.</td>
</tr>
<tr>
<td>Lifecycle Management</td>
<td>The service lifecycle management capabilities provided by the PaaS platform (e.g. staging, support for multiple service versions, etc.). 4CaaSt platform will provide a management lifecycle fitted for composite and multitenant applications. 4CaaSt will completely describe the life of a 4CaaSt service, from design, planning and alignment with the revenue model, through to the delivery of the service. The output of this criterion will be a summary of the lifecycle of a service inside the given platform and the available tools for its management.</td>
</tr>
<tr>
<td>Built in Platform services</td>
<td>The set of available services in a PaaS platform. For instance mail, calendar, SQL, logon services, maps, etc. The output of this criterion will be the list of available platform services.</td>
</tr>
</tbody>
</table>

1.1.2. Evaluation Results

1.1.2.1. CompatibleOne

The CompatibleOne platform [15] is an open source cloud services broker as defined by Gartner, i.e. a core cloud computing mechanism providing intermediation, aggregation and arbitration of cloud services. Currently brokering is available at IaaS level (OpenNebula, OpenStack, Windows Azure …) and a first experimentation is on-going at the PaaS level.

The CompatibleOne platform is aligned with the Cloud Computing Reference Architecture of the National Institute of Standards and Technology. The CompatibleOne approach is based on the open cloud computing interface OCCI.
The CompatibleOne platform provides an automated provisioning and deployment platform independent of any underlying cloud services provider. For example, the CompatibleOne broker could be connected to the 4CaaSt platform by providing a 4CaaSt procci. A “procci” thereby represents an OCCI-based proxy component that enables the integration of an specific cloud service (IaaS, PaaS, XaaS, ...) with the CompatibleOne brokering framework named CORDS (CompatibleOne Resource Description System). The CompatibleOne Manifest document describes the user needs in terms of cloud services. This Manifest could be translated to the 4CaaSt blueprint by the 4CaaSt procci.

For the PaaS services brokering, CompatibleOne currently develops a Proof of Concept including a PaaS agnostic REST API and a full PaaS stack for standard Java EE and OSGi applications (PaaS4Dev module).

CompatibleOne originated as a collaborative project endorsed by the Systematic Competitive Cluster and supported financially by the Paris Region. The partners in the CompatibleOne project include ActiveEon, Bull, City Passenger, Enovance, Eureva, Mandriva, Nexedi, Nuxeo, Prologue, Xwiki, Inria, Institut Telecom and OW2.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock-in grade</td>
<td>Low grade by relying on a standard interface (OCCI), a PaaS agnostic API and Open Source code</td>
</tr>
<tr>
<td>APIs</td>
<td>REST API, OCCI standard based</td>
</tr>
<tr>
<td>Service Definition Language</td>
<td>XML files (named MANIFEST documents) to describe the service and user requirements. The cloud meta-data are expressed as technical and economical criteria and specifications and constraints that are to be taken into consideration. For a deployment in the PaaS, the Manifest describes an application version with its set of modules (e.g. war-files), and for each one the requirements in terms of a multi-tenancy level. Along with that, the Manifest describes the platform template with the configuration and the cardinality of each component (router/container/database) and their relationships.</td>
</tr>
<tr>
<td>Service Catalogue</td>
<td>A component named 'Publisher’ provides publication services or repository services</td>
</tr>
<tr>
<td>Service Composition</td>
<td>Manifest documents can be nested, what permits to provide composition of services</td>
</tr>
<tr>
<td>Scalability and multi-tenancy capabilities</td>
<td>Elasticity is provided by the COES module (CompatibleOne Elasticity Scalability Services) for IaaS resources. For PaaS level, elasticity depends on the underlying PaaS provider capabilities. Same for multi-tenancy support which depends of the underlying PaaS provider accessible from the broker. The PaaS stack provided by CompatibleOne (PaaS4Dev) is already multitenant-capable and aims at providing elasticity in the future</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>Depends on the underlying PaaS provider. The PaaS stack provided by CompatibleOne (PaaS4Dev) is focused on Java EE and OSGi applications</td>
</tr>
</tbody>
</table>

Copyright © SAP and all other members of the 4CaaSt consortium 2012
Inter-component communication protocols | In the PaaS4Dev module, applications are accessible through HTTP REST interfaces.

Networking elements | PaaS4Dev module configures a load balancer for accessing the applications.

Cloud Database Service | PostgreSQL and MySQL support on the roadmap

Lifecycle Management | PaaS4Dev module enables to deploy several instances of the same application version for different environments (test, production, …). The application version instance can be deployed and started independently

Built in Platform services | The PaaS4Dev module includes Java EE services (Java EE 5, Java EE 6 web profile) and Enterprise OSGi services (HTTP, JNDI, Transaction …).

### 1.1.2.2. Cordys

Cordys [16] provides a Business Process Management (BPM) solution that allows the integration and orchestration of various on-premise and on-demand solutions. Cordys can be consumed as a Platform-as-a-Service model in two ways: (1.) Cordys Process Factory and (2.) Cordys Business Operations Platform. While the process factory mainly focuses on developing mashed-up business applications, the Business Operations Platform of Cordys offers a range of integration, process and application design capabilities. Hence the following evaluation will focus on the Cordys Business Operations Platform, since this offering comes closest to the 4CaaSt platform.

Cordys Business Operations Platform supports business process design, execution, monitoring and improvement. Besides the Business Process Management Suite (BPMS), the Platform consists of a Composite Application Framework (CAF) that allows the composition of applications and user interfaces and the Service Oriented Architecture (SOA) Grid. The SOA Grid layer includes a Master and SOA Enterprise Service Bus (ESB), Business Services and Master Data Management (MDM), see Figure 1.
The following table includes the Cordys evaluation results.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lock-in grade</strong></td>
<td>Business processes are defined in BPMN, and mainly WS-* standards are used for service integration. The actual business logic is done in standard java. However, with WS-AppServer Cordys provides a powerful tool to wrap legacy systems, in order to make them available in Cordys Business Operations Platform (BOP).</td>
</tr>
<tr>
<td><strong>APIs</strong></td>
<td>There is only an API for the Metering Service.</td>
</tr>
<tr>
<td><strong>Service Definition Language</strong></td>
<td>Cordys uses the Business Process Model and Notation (BPMN) in order to model the business process. When it comes to adding functionality to the process, so called user interfaces and service calls are used. Services are described by using the Web Services Description Language (WSDL).</td>
</tr>
<tr>
<td><strong>Service Catalogue</strong></td>
<td>A tenant can subscribe to an application by selecting it in a catalogue of applications. All applications are managed by service providers and can be offered via different sales channels where they can be procured, subscribed to, and consumed by the tenants.</td>
</tr>
<tr>
<td><strong>Service Composition</strong></td>
<td>Cordys CAF allows integrating disparate data sources, legacy systems, business software and web content into a personalised, process oriented workplace.</td>
</tr>
<tr>
<td><strong>Scalability and multi-</strong></td>
<td>Cordys BOP scales vertically (scale up/down) as well as horizontally (scale out/in). Scalability is generally accomplished using commodity</td>
</tr>
</tbody>
</table>

Figure 1. Cordys Business Operations Platform
**tenancy capabilities**

Multitenancy is a basic feature of Cordys BOP. All functionality is invoked in the context of a user and the organization of that user. The organization is not really located on a given machine. All service containers - wherever located - can execute the functionality on behalf of a user in the context of that organization.

<table>
<thead>
<tr>
<th>Programming Languages</th>
<th>Java</th>
</tr>
</thead>
</table>

| Inter-component communication protocols | LDAP, SOAP, UDDI, WSDL, XForms, XInclude, XML, XML-DOM, XPath, XSD, XSLT, SAML, WS-I, BPMN, XPDL, JMS, JMX |

| Networking elements | The ESB has pluggable load balancing algorithms that decide which service instance to address. E-mail connector: The e-mail connector enables sending and receiving mails through IMAP or SMTP and POP3. SAP connector: SAP connector provides connectivity to SAP back-ends through RFC and BAPI. Script connector: Script connector allows web services to be consumed in JavaScript. |

| Cloud Database Service | Database connectors like JDBC and OLEDB. Standard Integration Tools for Oracle, Microsoft SQL Server and MySQL |

| Lifecycle Management | Cordys BPMS supports the Discovery, Modelling, Executing, Monitoring and Improvement of business processes. However, they do not provide lifecycle management for the underlying, orchestrated services. |

| Built in Platform services | The Metering Service provides usage tracking of applications and services. In addition Mail, Messaging and SAP Connection Services are available. |

---

1.1.2.3.  **Jelastic**

Jelastic [23] is a PaaS service that allows developers to deploy standard java applications in the cloud without any change or adaptation of source code. It supports a number of well-known development technologies as:

- Application logic: Tomcat6, Tomcat7, Jetty6 or GlasFish3 servers using JDK6 or JDK7.
- Storage:
  - SQL: PostgreSQL 8.4, MySQL 5.5 and MariaDB 5.2.
  - No-SQL: MongoDB 2.0 or CouchDB 1.1.

The developer can get access to the main folders in the nodes (e.g., webapps for Tomcat) to upload application components (e.g., war files), and the configuration files to configure the containers or databases.

The topology for the deployment of applications can be based on load balancers to allow replication of nodes for horizontal scalability and to enable high availability based on session
replication (a mechanism used to replicate the data stored in a session between different instances, which have to be a part of the same cluster). This topology can be changed at any time.

![Environment topology](image)

**Figure 2: Sample environment design (topology + software) for an application in Jelastic**

The elastic rules are based on the KPIs at the infrastructure level (CPU, RAM, Network or HDD) and two possible mechanisms are supported:

- Vertical scalability, by increasing/decreasing resources of the nodes.
- Horizontal scalability, by adding more or removing nodes under a load balancer (in a range of nodes specified by the developer).

The platform also provides a number of guidelines to allow customers to deploy different well-known open source applications and platforms, like Alfresco, Magnolia CMS, Google Web Toolkit, etc.

The pricing of the platform is based on the usage of resources: cloudlets (a measure of a VM capacity), storage, IP addresses per hour and SSL usage per hour.  

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock-in grade</td>
<td>Zero lock-in. Standard java applications are deployed without changes on well known web containers and databases</td>
</tr>
<tr>
<td>APIs</td>
<td>Those of the supported web containers and databases</td>
</tr>
<tr>
<td>Service Definition Language</td>
<td>N/A</td>
</tr>
<tr>
<td>Service Catalogue</td>
<td>Not available. Only own supported platform are available</td>
</tr>
<tr>
<td>Service Composition</td>
<td>Supported, limited to the predefined environments</td>
</tr>
<tr>
<td>Scalability and multi-tenancy capabilities</td>
<td>Not multitenant at the PaaS level. PaaS servers run on reserved IaaS resources. It supports vertical and horizontal scalability</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>Java</td>
</tr>
<tr>
<td>------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Inter-component communication protocols</td>
<td>Protocols supported by Java inside the web containers (e.g., RMI) and required to communicate to supported databases (e.g., JDBC)</td>
</tr>
<tr>
<td>Networking elements</td>
<td>Load Balancers, public IPv4 addresses and SSL support can be configured for the applications</td>
</tr>
<tr>
<td>Cloud Database Service</td>
<td>SQL: PostgreSQL 8.4, MySQL 5.5 and MariaDB 5.2 No-SQL: MongoDB 2.0 or CouchDB 1.1</td>
</tr>
<tr>
<td>Lifecycle Management</td>
<td>Jelastic provides a configuration tool to define environments before deployment and during runtime. Configuration and upload/upgrade of components is done manually</td>
</tr>
<tr>
<td>Built in Platform services</td>
<td>Only those provided by the web containers themselves</td>
</tr>
</tbody>
</table>

1.1.2.4. CloudLogic

CloudLogic [17] a company that specialises on consulting services in the area of cloud computing. It focuses on small and medium organizations and helps them migrating their current software/processes to existing SaaS applications like SalesForce.com, Google Apps etc. Additionally, it acts as a reseller for Google Apps for Business (Gmail, Calendar, Drive/Docs, Sites & Vault). The CloudLogic business model so far is based on fees for the consulting services, which depend on the fees and features of the resold SaaS services (e.g. Google Apps). For instance, the migration of email accounts to Google’s Gmail is 10 GBP per mailbox. Other pricing examples are training fees (400 - 840 GBP per ½ day onsite training) or support (15 GBP for 15 minutes).

Cloud Logic cannot be considered as a XaaS offering, but the offered migration services are highly interesting for every PaaS solution as a value-add service. As confirmed by our PaaS market analysis conducted in WP 9 [11], 4CaaSt customers would appreciate consulting services for choosing the right solution for their needs and for migrating their old solution to the new cloud-based one. Such consulting and migration services could be easily offered via the 4CaaSt marketplace. CloudLogic shows how a corresponding business model could look like, in particular regarding the pricing.
2. **WP-specific Update on State of the Art**

This section for provides a brief update on WP-specific state of the art (SotA). For each work package a brief summary of new SotA discussed since the last iteration of the deliverable along with is given. The detailed version can be found in the corresponding scientific reports for each work package (D[WPx].1.2).

2.1. **WP 2 – Service Engineering and Lifecycle Management**

An up-to-date state-of-the-art analysis of the contemporary approaches for describing, representing, and managing cloud services and resources has been performed during the second iteration of the project and reported in the Deliverable D2.1.2 [3]. It is an extension of the state-of-the-art study presented in D2.1.1 [2] in the first iteration. Basically, we report the following recent approaches:

- A recent application lifecycle management approach on top of an IaaS cloud, in which the OVF is used for the specification of application component packaging and distribution.

- The OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) [27], which also builds on the OVF and plans to enable the interoperable description of application and infrastructure cloud services, the relationship between the service components of a cloud application, and the operational behaviour of these service components (e.g. deploy, patch, undeploy, etc.)

- A declarative configuration framework named SmartFrog [31] for managing the component descriptions of a cloud application.

Our analysis has shown that the recent approaches mainly focus on the specification formats for the packaging and distribution of application components, as well as deploying these components on a federated IaaS cloud. This yields the following shortcomings:

- Recent approaches do not consider the components of a cloud application as independent, loosely-coupled SaaS components that need to be designed for, developed and deployed on the cloud. This is because they merely focus on the IaaS level for the deployment of cloud applications.

- Other approaches (already reported in the first iteration deliverable D.2.1.1) develop cloud applications using distributed SaaS components. However, they usually lead to a vendor lock-in approach where the monolithic SaaS components are predominantly tethered to proprietary platform and infrastructure of a cloud vendor.

- There is no explicit support for specifying end-to-end non-functional requirements for application components and cloud resources.

- Although there has been an OVF-based approach targeting the lifecycle management for a cloud application, it focuses only on the specific points of the lifecycle: definition and deployment configuration for the components, and barely consider runtime monitoring and scaling for dynamically composed and configured applications.

The identified shortcomings help us consolidate the aims and purposes of the Blueprinting Approach as a complete Service Engineering and Lifecycle approach for developing cloud applications. Using the blueprint as the uniform specification format for cross-layered cloud services, i.e. SaaS, PaaS, and IaaS, developers can design and configure their cloud applications as a mixture of SaaS, PaaS, and IaaS blueprints of different vendors.
2.2. WP 3 – Marketplace

A detailed analysis of the State of the Art of electronic marketplaces has been provided during the first review period (month 1 to 14), see [4]. In this analysis, the most prominent cloud service marketplaces (Windows Azure Marketplace, SuiteApp Marketplace, Zoho marketplace, Google Apps Marketplace, Android Marketplace and Force.com AppExchange) have been compared along the main phases involved in market transactions. While the first version of the scientific and technical report of WP3 had a strong focus on the definition of terms and conditions related to electronic marketplaces, as well as scientific and practical state of the art [4], the second version [5] focuses on the major innovations done within WP3. Hence, only state of the art, which is closely related to one of these major innovations, has been investigated. In particular, state of the art related to the following innovations has been in considered:

- **Business-based search, selection and recommendation:** Based on existing service selection algorithms for Web services like for instance Yu and Lin (2005) [40] a business resolution and service selection algorithm for the 4CaaSt marketplace is developed. 4CaaSt’s resolution algorithm evaluates business characteristics (security, availability and price etc.) of the products that fulfill the technical requirements, taking also into consideration the customer’s profile and pricing aspects. The service selection problem is treated as a multi-attribute decision making problem, and hence the approaches of See, Gurnani, and Lewis (2004) [30] as well as Rehman, Hussain, and Hussain (2011) [29] have been taken in account.

- **Social Enhancements.** State of the art social enhancement features include:
  - Socially enhanced user adaptation: Currently established electronic marketplaces individualize their offerings, but thereby do not take into account the social environment of the particular user. Jung and Lee [24] propose a socially enhanced interaction environment for electronic marketplaces.
  - Socially enhanced search: Some socially enhanced search tools exist today (e.g. Google, Facebook profile search). However, those functionalities are optimized for private usage scenarios and do not take into account the characteristics of business applications. Evans and Chi [19] have conducted a survey to assess the importance of social interactions in search events.
  - Socially enhanced market analysis: Current market analysis tools are inaccurate, e.g. due to a lack of high-precision data input for analysis. Borgatti (2005) [13] categorizes different types of interaction between two or more persons on a social network providing a basis for analyzing the market potential of products.
  - Prevention of unfair ratings using social data: Many approaches to tackle unfair ratings exist, however social data is not employed yet to prevent unfair ratings on electronic marketplaces. Zhang and Cohen [41] compare different approaches to prevent unfair rating in electronic marketplaces, and introduce their own approach, which can be leveraged within the 4CaaSt marketplace.

- **Business Model Simulation:** Based on a systematic literature review multiple publications have been extracted that mainly focuses on business model analysis and design methods.
  - Tennent & Friend (2005) [34] give advice to practitioners in their guide book to business modelling.
  - A more general approach, however with similar output, has been published by Osterwalder & Pigneur (2010) [27], based on prior academic research (A Osterwalder & Pigneur, 2002; Alexander Osterwalder, Parent, & Pigneur, 2004).
  - A business model framework is introduced in (Weiner & Weisbecker, 2011) [32] specifically for the internet of services.
The Business Model Analysis Method by Grasl (2009) [20] is an approach to help practitioners improve their business model and derive strategic recommendations based on the formalized business model.

Schief & Buxmann (2012) [33] conceptualize a software industry business model, supporting the description of software firms’ business models in a standardized manner.

The analysis of the state of the art confirms the relevance of the research question and the research gap addresses by 4CaaSt. Different kinds of business model definitions, taxonomies and components have been identified and have been taken into account for further usage in the simulation method, developed in WP3.

- **Price Model Design and Simulation:** Once a service provider has decided upon his business model and corresponding service offerings, the next step is to define the pricing strategy with associated price models. The research done in WP3 takes place in the context of Service Value Network (SVN). A SVN is defined as the flexible, dynamic, delivery of a service, by a business’s coordinated value chains, such that a value-adding, specific, service solution is effectively, and efficiently, delivered to the individual customer, see Hamilton (2004) [21]. The following literature as been identified to be closely related to 4CaaSt price model simulation approach:
  - Service networks built on contribution-based value distribution are investigated by Conte, Satzger, Blau, and Dinther [14].

2.3. **WP 4 – Resource Deployment and Management**

During the second iteration of the project, a state of the art study has been performed for the NaaS (Network as a Service) features, which was missing in the first iteration. The Technical and Scientific Report D4.1.2 [6] presents the full results of this study as well as the achievements beyond the state of the art that have been achieved in the project during the second reporting period. In the following we provide a brief summary of the NaaS-related state of the art analysis. The analysis on the one hand covered approaches already used in data centres, such as IP-based networks, Ethernet-based networks and TRILL-based networks (Transparent Interconnect of Lots of Links) [22]. On the other hand, we analysed approaches not yet employed, in particular L3VPN MPLS (Multi Protocol Label Switching), VPLS (Virtual Private LAN Service) [26] and Software Defined Networking (SDN) [18].

The NaaS concept extends existing 4CaaSt capabilities for instantiating virtualised landscapes by the possibility to handle additional requirements regarding the network. This particularly includes QoS guarantees between the VMs and data center network optimisation.

According to our state of the art analysis, the main drawbacks of the existing approaches are:

- **Security:** VMs running in the same address space are separated by firewalls which naturally add to the delay experienced between the VMs and the end user.

- **Manageability:** IP networks are hard to manage automatically as they use distributed protocols with hard to manage routing.

- **Power consumption:** Today distributed networks do not respect power usage due to the nature of distributed information.
- **QoS**: Pure IP networks are prone to have congestion which could introduce degradation in the user experience or even traffic black holing or packet loss.

- **Scalability**: Using VLANs for user insulation limits the number of possible tenants as there are only 4094 possible VLANs. Spanning tree protocols for Ethernet networks could waist considerable bandwidth thus limiting scalability.

Based on these drawbacks and further considerations the requirements are broken down into the following logical categories: Interfaces (e.g., the interfaces should have modular architecture), Computation (e.g., the NaaS may support PCEP), Interworking (e.g., the NaaS should support BGP for establishing Internetworking) and Multi-Tenancy Support (NaaS should isolate the tenant traffics from each other).

2.4. **WP 5 – Administration, Accounting, Monitoring and Analytics**

A State of the Art study has been performed during the second iteration of the project regarding the Accounting domain, as it had not been provided during the first iteration. The Technical and Scientific Report D5.1.2 [7] presents the results of this study, as well as the achievements beyond the state of the art that have been performed in the project during the second reporting period.

Regarding the Accounting State of the Art study, requirements have been identified to ensure that metrics cover both IaaS, PaaS and SaaS, that accounting model and granularity of metrics are flexible, and that the framework is multi-tenant aware. Grid-SAFE, RESERVOIR and Smart meter frameworks have been evaluated, concluding that the metering approach is based mainly on IaaS level metrics. On the other hand, the billing (and required accounting) strategy of a number of commercial PaaS platforms have been analysed: Heroku, Google App Engine, Amazon RDS and Jelastic PaaS. Apart from Google AppEngine, all the other platforms provide accounting capabilities at the level of IaaS. This analysis reflects that different business models require very different accounting approaches. Thus, a flexible framework must be able to deal with multiple strategies, while there is a gap in the usage of PaaS and SaaS level accounting metrics in existing frameworks and platforms.

2.5. **WP 6 – Native PaaS Technologies**

The update of the State of the Art in WP6 is related to the concept of a Service Enabler, which refers to common interfaces that allow heterogeneous components to act as homogeneous services in a 4CaaSt cloud environment. In this sense, a similar approach is being developed inside the TM Forum, called Software Enabled Services (SES) Management Solution [35]. SES, and specifically the SES Management Interface (SMI) [36], represents the least common denominator contract that any software enabled service has to support so as to be integrated into TM Forum’s Frameworx. SMI shares several objectives with the concept of Service Enablers as defined in 4CaaSt, though our approach differs slightly due to differences in scope and technical implementation constraints. A more detailed set of differences can be found in [8].

2.6. **WP 7 – Immigrant PaaS Technologies**

In this section we briefly provide an update on state of the art concerning the tasks of WP7. Therefore, we focus on the delta compared to the state of the art provided in D7.1.1 [9]. Each of the following subsections summarises this delta for one of the tasks of WP7: T7.1 (Section 2.6.1), T7.2 (Section 2.6.2), T7.3 (Section 2.6.3), and T7.4 (Section 2.6.4). D7.1.2 [10] in this case does not contain any additional SotA analysis!
2.6.1. Cloud-aware Database Engines

To the best of the knowledge of the partners involved in T7.1 there are neither new functionality offered in products or services nor new research topics or directions in the area of cloud-aware database engines. Thus, we consider the state of the art on this domain provided in Section 3.1 of D7.1.1 up to date. In D7.1.2 we therefore focused on identifying distinct features for cloud-aware databases based on this state of the art, most importantly a fully transparent and dynamic scalability.

2.6.2. Cloud-aware Application Servers

Since we introduced performance isolation as a new topic besides multi-tenancy enablement, we structured this section into two subsections investigating the update on state of the art with respect to these two topics.

Multi-tenancy of Application Servers

A couple of initiatives and results partly covering our goals of enabling multi-tenancy appeared during this period. They are briefly described below.

- WSO2 Stratos [38] provides a complete Enterprise Cloud Platform that embraces SOA and modern Enterprise Architecture. It is one of the first enterprise solutions to support multi-tenancy at the platform level. However the application server is focused on the WS-* implementation and doesn't implement the whole Java EE specifications.
- Jelastic [23] is a promising new PaaS solution for Java application, but regarding multi-tenancy so far is limited to level 1 of multi-tenancy, i.e. “one application per container per VM”.

Performance Isolation

The first version of D7.1.1 did not consider performance isolation within multi-tenant environments. In the following, we provide a short summary of the state of the art in this domain. For the full SotA discussion, please refer to our corresponding 4CaaSt publication [25].

Performance isolation is about ensuring that the customers' perceived performance is not affected by other customers on a shared system exceeding their quotas. The enforcement of performance isolation requires additional mechanisms. However, in the context of multi-tenancy most approaches focus on optimising resources, e.g. by dynamically (re-) allocating tenants, and do not address this particular problem.

Besides the isolation mechanism a benchmark and metrics are missing for comparing the level of performance isolation for different SaaS or PaaS offerings in a quantitative way. Different benchmarks and metrics were developed in the past years. Usually these publications focus on single aspects of cloud services like databases. Others discuss metrics for cloud features like elasticity. The most relevant related work for quantifying performance isolation comes from the field of virtualisation, namely metrics for expressing the efficiency of a virtualised environment. The Average Normalised Reduced Throughput (ANRT) for instance reflects the loss of throughput on a per VM basis, when additional VMs are deployed. Nevertheless, they do not set the amount of changed workload in relation to ANRT and use static amount of workload for the VMs. Thus, these metrics are not feasibly to be used for quantifying performance isolation.
2.6.3. Cloud-aware Composition Frameworks and Engines

Since the submission of D7.1.1 a study has been done with the goal to compare existing BPEL and BPMN engines, which have not yet been evaluated in D7.1.1 and identify shortcomings with respect to cloud-awareness. The study includes 6 commercial (IBM Business Process Manager, Oracle BPM Suite 11g, SAP Netweaver BPM Server, etc) and 2 non-commercial (JBoss BPM and Activity) products as well as 5 cloud services (Cordys, BPM by Intalio, etc.) [12].

The following results are particularly interesting. The two most suitable workflow engines according to our criteria and weightings are Interstage BPM by Fujitsu and ActiveVOS Datacenter Edition. Both are also most suitable for Cloud usage. When looking at the SaaS/PaaS offerings the best workflow service is the hosted version of Interstage BPM by Fujitsu.

Most traditional workflow engines perform well in the traditional workflow areas like workflow functionality, integration, communication, extensibility, reliability, security, development support, operation and management support and clustering. But in new areas, which this evaluation focused on, like cloud-awareness, multi-tenancy, capabilities, and quality of service enforcement, most evaluated traditional workflow engines do not meet the new requirements and have therefore significant shortcomings when trying to deploy them “as-is” in the cloud.

2.6.4. Cloud-aware Integration Technologies

To the best of the knowledge of the partners involved in task T7.4 there is no new functionality offered in products or services, but cloud-awareness, especially enabling multi-tenancy for middleware technologies is currently a hot topic addressed in research. For example Walraven et al. propose a multi-tenancy support layer that combines dependency injection with middleware support for tenant data isolation [37]. In contrast to this proposal in the research and work done in T7.4 so far we investigated an extensible and reusable solution to enable tenant-aware communication independent from the communication protocol used and provided a proof of concept implementation based on open source Enterprise Service Bus Apache ServiceMix and the three communication protocols SOAP over HTTP, JMS, and e-mail.
3. Conclusion

In this final iteration of the 4CaaSt project’s State of the Art analysis, we provided an update on the PaaS state of the art covering four new industrial offerings, which are strongly related to the envisioned 4CaaSt platform. All of these platforms represent rather new players in the market. But compared to the established players Microsoft Azure and Google App Engine they offer some interesting new features. In the following we briefly discuss their relation to 4CaaSt:

- CompatibleOne is currently developing an abstraction layer that allows the combination of and migration between different existing PaaS offerings. This would be a very interesting alternative or extension to 4CaaSt, which avoids vendor lock-ins by creating the PaaS layer through dynamic configuration of software stacks on top of existing IaaS offerings.

- Cordys already offers a composite application and BPM platform as a service, which is also addressed by 4CaaSt in terms of the compositions engines. Using 4CaaSt it should be possible to offer a similar service, which is not limited to one BPM engine but instead leaves you the choice between different ones.

- Jelastic has the advantage that it is able to deploy standard java applications in the cloud without any change or adaptation of source code. However, the approach is restricted to a limited set of middleware. 4CaaSt also tries to avoid demanding changes to the original source code and additionally offers flexibility regarding the used middleware components. Like 4Caast, Jelastic offers comprehensive support for scalability, but the KPIs and actions are limited to the IaaS level only.

- Cloud Logic cannot be considered as a XaaS offering, but the offered migration services are highly interesting for every PaaS solution as a value-added service. 4CaaSt should support offering such complementary services via the 4CaaSt marketplace.

Besides this update on new PaaS SotA, the report included a brief update on the WP-specific SotA discussion, which mainly focussed on advanced 4CaaSt features, which have not been addressed by the previous platform versions.
References


