Building the PaaS Cloud of the Future

Use Case - Virtual private Cloud for Large Corporations: Scenario Definition

D 8.3.2

Version 1.0

WP8 – Experimentation

Dissemination Level: Public

Lead Editor: Axel Spriestersbach

15/10/2012

Status: Final Version

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

Seventh Framework Programme

FP7-ICT-2009-5

Service and Software Architectures, Infrastructures and Engineering

Contributors: Axel Spriestersbach (SAP)
Jose Luis Vazquez-Poletti (UCM)
Frederic Junker (HSG/SAP)
Krisztian Sinka (NSN)
Pavlos Kranas (NTUA)

Internal Reviewer(s): Sergio Garcia Gomez (TID)
Stéphane Carrié (FT)

Version History

Version	Date	Authors	Sections Affected
0.1	26/07/2012	Axel Spriestersbach	Import D8.3.1 Resubmission
0.2	06/09/2012	Axel Spriestersbach	Updated TOC from proposal; changed structure to “old” version as agreed during call
0.3/4	1/10/2012	Axel Spriestersbach; Frederic Junker; Jose Luis Vazquez-Poletti, Krisztian Sinka	Updates MP features; Failover fearures, NaaS
0.5	9/10/2012	Axel Spriestersbach	Integration of content into appropriate sections; added section on business value
0.6	12/10/2012	Axel Spriestersbach, Pavlos Kranas	Minor additions und fixes; Internal review version
1.0	15/10/2012	Axel Spriestersbach	Include review comments

Table of Contents

List of Figures

List of Tables

Abbreviations

4CaaSt	Building the PaaS Cloud of the future
ABR	Abstract Resolved Blueprint
AC	Application Component
API	Application Programming Interface
B2B	Business to Business
B2C	Business to Consumer
BAC	Business Advisory Committee
CAPEX	Capital Expenditure
DoW	Description of Work
DSL	Domain Specific Language
EULA	End User License Agreement
GA	General Assembly
IaaS	Infrastructure as a Service
IP	Internet Protocol (project management context: Integrated Project)
ITIL	IT Infrastructure Library
NAT	Network Address Translation
OF	OpenFlow
OVF	Open Virtualization Format
OVS	Open vSwitch
OPEX	Operational Expenditure
P (node)	Provider node. In this context: not the border nodes of the data center
PE (node)	Provider Edge node. In this context the border nodes of the data center
PaaS	Platform as a Service
PC	Project Coordinator
PIC	Product Instance Component
QoS	Quality of Service
REC	Runtime Execution Container
S&A	Service and Applications (used synonym in this document)
SaaS	Software as a Service
SB	Steering Board

SLA	Service Level Agreement
SLO	Service Level Objective
TAC	Technical Advisory Committee
TCC	Technical Coordination Committee
TCO	Total Cost of Ownership
TL	Task Leader
TM	Technical Manager
TOC	Table of Content
VM	Virtual Machine
WP	Work Package
WPC	Work Package Committee
WPL	Work Package Leader
XaaS	Anything as a Service: SaaS, PaaS, IaaS level services, but also Native or Immigrant APIs exposed as a Service

1. Executive Summary

This deliverable is the second 4CaaSt deliverables focusing on scenarios for evaluating the 4CaaSt platform and its features against real life use cases. The initial version D8.3.1 described so called virtual private clouds - services that “use public cloud resources to create their private cloud”.

According to that definition the value of virtual private cloud results from combining public resources and private parts into a cloud landscape, while the private parts maybe completely virtual, i.e. just a set of public cloud resources that are accessible by a limited user group, or very often a combination of a private part within a public cloud (available “on demand”) and a private Cloud running within company boundaries (“on premise”) and connections in between (see Figure 1).

The resources used from the public cloud are usually computing or storage resources. They are used to create a private cloud or extend existing private services in case of resource shortages (see Figure 1(3)). Both scenarios are supported by the 4CaaSt generic features to automatically scale and optimize resource usage allowing expanding or shrinking allocated public resources as required. In addition NaaS helps isolating private parts with the public cloud or to provide the required network links.

Besides the rather standard shared resources described before, 4CaaSt offers a new kind of resource sharing with its trade-able 4CaaSt cloud components. They are described via blueprints offered from a 4CaaSt marketplace so that they are easy to be reused in any kind of private cloud environment - company internal or completely virtual private running on top of a public cloud service.

In contrast to the initial version of this document that focused on an Emergency scenario considered too complex during the Y1 review meeting, the resubmissioned document describes a simplified scenario dealing with records management that maintains all kinds of personal records such as warrant (focus for now),bills, credit cards or bank statements in a single web portal. The issuers of such records, as for example retailers, energy suppliers, or banks, provide access to their records databases including meta-data:the warrant time, product information, service location etc. The public part is a record management portal giving access to all records assigned to a registered user, while the private parts are company specific virtual private deployments for each record provider. To further narrow the scope of the use case and show some of the native 4CaaSt platform features, we focus on warrants as specific record type in some aspects of the scenarios.

This iteration of the report is based on the initial version, with some removals but in its core extensions on the scenario itself to better adopt the 4CaaSt features (described in the experimentation report [9]) or more details on features that have been mentioned in the first report.

2. Introduction

Note: To ease the reading of the document, all modifications to the previous version are marked with ~~strikethrough~~ for removals and gray for additions in the content section 3 to 5 , but not in the sections “Summary“, „Introduction” and “Conclusion”)

The primary goal of the WP8 use cases is validating the 4CaaSt value propositions and provides measurable results for backing up market analysis done in task 9.1. We distinguish 4CaaSt as a generic PaaS platform and 4CaaSt as a service portfolio.

- Generic PaaS platform: 4CaaSt is first of all a framework for managing the lifecycle of an application, from specification to billing including pricing, rating, deploying and monitoring.
- 4CaaSt as a service portfolio: In addition 4CaaSt is a set of predefined components (described by blueprints) that an application/service developer can leverage in his application/service if they want to a) inherit some of the generic 4CaaSt features depending on the component, such as scalability and/or b) benefit by using those components “as a Service” without the need of operation, updates, maintenance etc.

Porting a pre-existing (open-source) application to 4CaaSt would probably have been sufficient for validating the generic PaaS platform, but not the 4CaaSt service portfolio. For this purpose, WP8.1, 8.2 and 8.3 define three scenarios with three prototypes each demonstrating some portfolio services.

The secondary goals of WP8 are

- Provide feedback on 4CaaSt platform usage to WP 1 to 7
    - Bug report
    - Usability report
- Provide material for dissemination

Most notably, the following goals are not in the scope of WP8, although this work package help contributes to their achievement as a side effect of iterative design and development:

- Define 4CaaST value proposition (done in WP1-7)
- Unit test 4CaaSt platform (done in WP 1-7)
- Confront 4CaaSt value proposition to market (done in WP9.1)

Achievements and outlook on extensions

In terms of progress towards the described goals (from the first iteration) the primary goal described above are partially met and partially adopted:

- Generic PaaS platform: A decision was made to focus especially on specification to deployment, monitoring and operation for this use case. Therefore little progress was made for “pricing, rating” and is also not planed.
- 4CaaSt as a service portfolio: some additional features to be used are described in the report, i.e. some details on the planed NaaS usage and a High Availability scenario.

The secondary goals are to provide feedback on 4CaaSt platform usage to WP 1 to 7 in terms of bug reports are a continuous progress during the implementation of the scenario demonstrator while the Usability or friendliness has been reported in the experimentation reports [9]

A source for scenario updates besides the ones anyway planed are described in the experimentation report [9]. The report especially mentions the missing NaaS feature that is added in this document. The current idea is to use the blueprint extension concepts, but introducing NaaS in Blueprints might also result in potentially additional requirements and changes to the blueprinting language. In term of the scenario implementation the reuse of 4CaaSt enabled App Server and Database are still pending but already in implementation. No further extension of the scenario description is required as the plans to use Network as a Service as a WP7 component is already described in the old scenario definition.

To also integrate with business models aspects as recommend we will especially look into using Business Model simulator and Socially Enhanced Market Analysis market for the solution to either the end consumers or records providers should be included into the scenario and is supported during the expert interviews.

Besides that and as this document is the second iteration of the scenario definitions most of the content remains unchanged, with changes highlighted as described above.

3. Virtual private clouds

A 2010 survey from IBM states "Cost savings and time to value are the leading drivers" [3] for cloud adoption. When looking at the survey it becomes clear that cost savings are achieved by lowering the OPEX, while the speed of adoption is due to shorter implementation cycles (and directly lower CAPEX). The study also contains an adoption list for typical public cloud service where already higher level services: Collaboration (Audio/Video/Web Conferencing) tools, and CRM or Sales force automation are topmost, while IaaS level services ("Business continuity/disaster recovery") are "only" the 3rd item listed. The study also shows the importance of hybrid setup and connectivity that are both high rated services.

The virtual private cloud scenario focuses on scenarios where certain services deployed and used by a company, are not maintained and operated by the company itself but by another PaaS provider. Services in this context are often extensions of existing business process. The necessary data exchange, implemented via (web) service interfaces by "on Premise" system, extend onPremise functionality or combine public and private services.

This leads to an architecture often referred to as Hybrid Clouds [5] where services consist of a public and private part. Hybrid Clouds are for example used to scale an internal application, combine internal with public information or access of private information from the public network, i.e. a mobile or HTML5 application hosted in a public cloud, but accessing backend/private information via well-defined and secured APIs. The following diagram depicts three typical cloud setups:

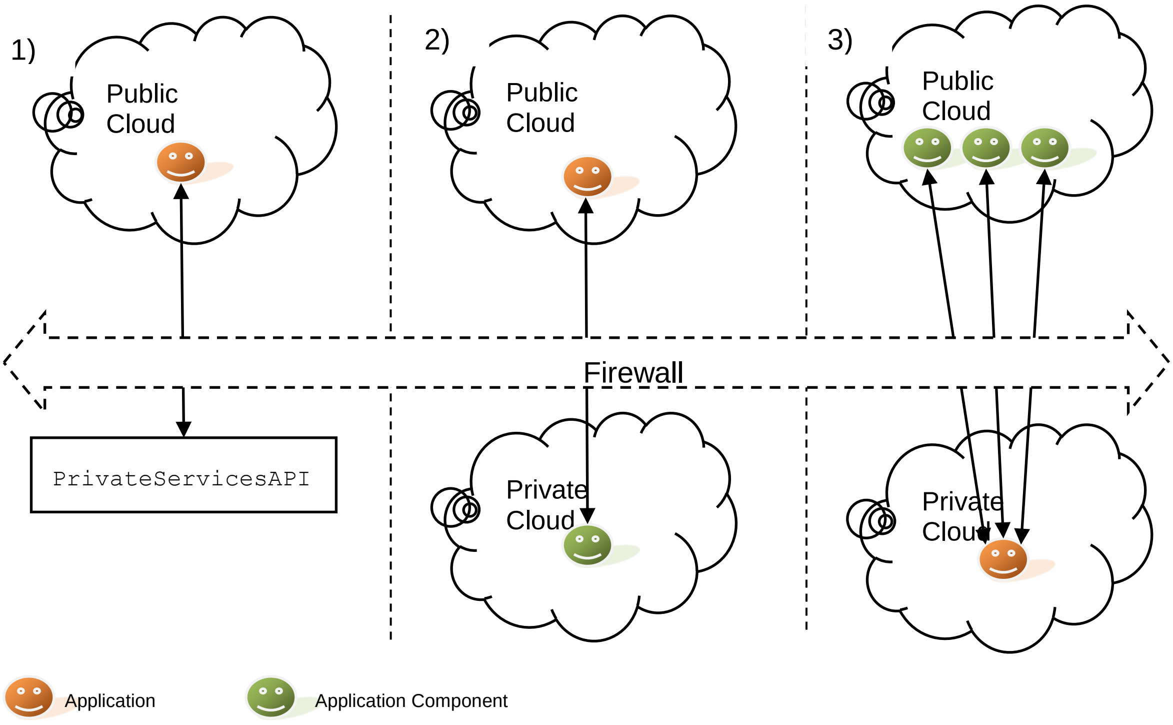


Figure 1. Role of Virtual Private Clouds in hybrid setups

In the first and easiest case, a backend service is accessed via service interfaces (i.e. a Web or RESTful service) from a public cloud. This setup is used in many cross company business processes using well defined (service) interface such as the `PrivateServicesAPI` in Figure 1 as for example a SAP Enterprise Services [4]. The drawback is manual deployment of the backend services done by a company IT department. While for "standard" enterprise services this is unproblematic, it becomes a problem for fine grained and non-standard services required in common case such as mobile applications.

The second case is service operated in the cloud with a backend system that requires additional software or new infrastructures to be installed and operated. Availability of a basic cloud environment such as a 4CaaSt which in principle[1] could run the software is assumed. Typical examples for such systems are services featuring a mobile frontend operated in the cloud requiring information from a backend not exposed via standard APIs (such as in the first case).

The third case is a typical cloud scenario in which a company has a running backend/on-premises system, but seeks for additional computing resources. It is usually for a shorter duration, for lower ramp-up times for service or for a lower initial investment that it is required to use additional resources provided by a cloud provider.

Both the second and third case can also be implemented as virtual private cloud where the private part is running in a public cloud infrastructure that is virtually isolated (in terms of access, network) from other public/private services.

3.1. Related work and deliverables

This deliverable describes one out of three scenarios used to evaluate parts of 4CaaSt. While each scenario evaluates different parts from the 4CaaSt platform some elements are shared among the three and therefore described only once - to avoid repeating - in

- 4CaaSt Value Proposition Whitepaper
- D8.2.1. [2] - The evaluation protocol/approach follows the S-CUBE [7] methodology that is described in D8.2.1. [2] in the sections *"Methodology"* and *"Domain description"*
- D8.3.1.- The initial version and basis for this document.[8]
- D8.3.5 – The experimentation report [9] describing potential updates of the scenario to make better use of 4CaaSt features. Some of them –not all – have been used to update the scenario description

S-Cube as described in D8.2.1.[2] is our general approach for the description of the individual use case. In D8.2.1.[2] gives a framework for the future evolution / next steps to be done by each of the use cases as for example for the evaluation specific evaluation criteria of all 4CaaSt components/features in the context of a specific use case needs be defined for the next iteration of this deliverable.

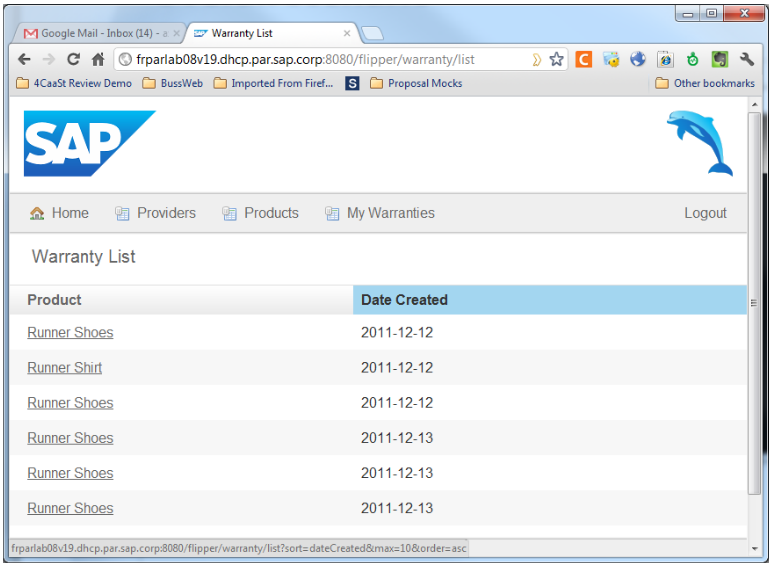
[1] Note: "In principle" is because often the Lifecycles of such parts are beyond a pure technical possibility to run such Application Components in such an environment, but also involve organizational setups, i.e. a testing phase, security checks and signoffs etc. that are ignored here.

4. Scenarios Concept (S-CUBE)

Records management example

The records management application generally manages miscellaneous personal records companies may issue, such as bills, warranties etc. Records issued by multiple companies they collected for individual users, so that they need to access only one central portal to manage their records, instead of individual systems of the issuing companies. This scenario focuses for now on the maintenance of warrants. The application itself consists of two parts: a public web-based frontend for the end user and the private component for each participating company that ether run a database in the virtual private cloud or maintain a connection (with a defined API) to the backend system also hosted in a the virtual private cloud.

The web frontend for the end user basically represents a list of records (see Figure 2) that can be extended by new records, removed, archived or displayed with detailed information. Records also offer certain operations depending on their record type. As our current focus are warranties the application supports adding products to personal list and maintain warrant information, such as warrant provider (usually the producer, but maybe a warrant insurance), warrant time, service addresses etc. Other user interactions with the record store maybe for instance analytics on money spending etc. In addition the system will proactively send reminders via e-mail and/or SMS upon warrants expiration.

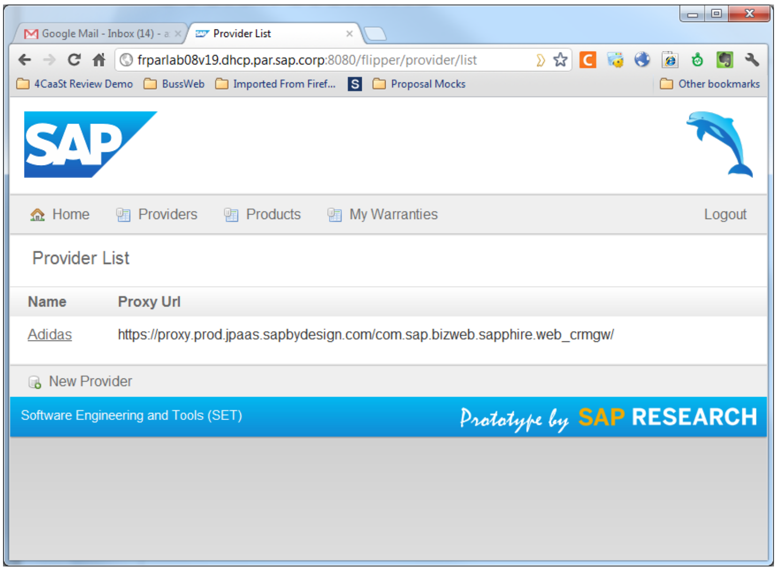

Google Mail - Inbox (14) - a
Warranty List
frparlab08v19.dhcp.par.sap.corp:8080/flipper/warranty/list
4CaaSt Review Demo
BussWeb
Imported From Firef...
Proposal Mocks
Other bookmarks
Home
Providers
Products
My Warranties
Logout
frparlab08v19.dhcp.par.sap.corp:8080/flipper/warranty/list?sort=dateCreated&max=10&order=asc

Warranty List

Product	Date Created
Runner Shoes	2011-12-12
Runner Shirt	2011-12-12
Runner Shoes	2011-12-12
Runner Shoes	2011-12-13
Runner Shoes	2011-12-13
Runner Shoes	2011-12-13

Figure 2. Record List

The web frontend is substituted by a “backend”. This backend stores the meta-data for records such as warrant using either a database hosted in the cloud or by implementing a certain API connecting to a backend of the warrant provider. The link between the product and warrant provider is done via an ID that is entered by an end user in a dialog, by using OCR or by scanning barcodes.


Google Mail - Inbox (14) - a
Provider List
frparlab08v19.dhcp.par.sap.corp:8080/flipper/provider/list
4CaaSt Review Demo
BussWeb
Imported From Firef...
Proposal Mocks
Other bookmarks
SAP
Home
Providers
Products
My Warranties
Logout

Provider List

Name	Proxy Url
Adidas	https://proxy.prod.jpaas.sapbydesign.com/com.sap.bizweb.sapphire.web_crmgw/

New Provider
Software Engineering and Tools (SET)
Prototype by SAP RESEARCH

Figure 3. Warrant Provider administration

In addition to that backend each record/warrant provider needs to register to the public cloud by the service provider[2] (shown in Figure 3). The private enabler components must implement a certain API and are deployed for each record/warrant provider with a connection to the public parts (see Figure 4). The implementation of the private parts may vary: for some this implementation may be only a gateway to their backend system, while for others it may be warrant storage in the private cloud database.

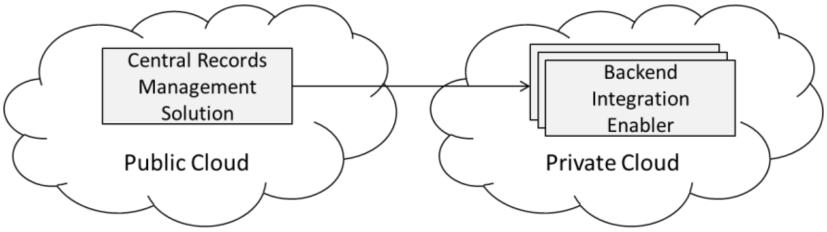

Central Records Management Solution
Public Cloud
Backend Integration Enabler
Private Cloud

Figure 4. Public/Private connection

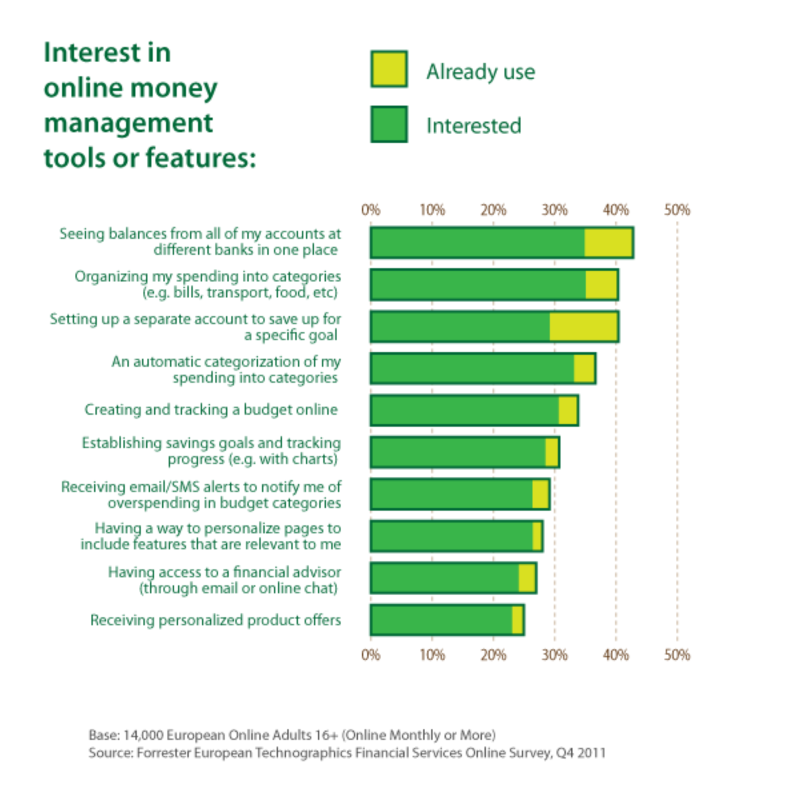
Applications bridging public/private clouds generally suffer from hybrid deployments with different lifecycles. One example requiring synchronized software lifecycles of private components and public components, are cloud web services depending on a specific services from a backend. The backend web service typically is installed and configured using a software lifecycle management system and process. As the public service depends on the availability of the backend web service, it's provisioning should start after the lifecycle management system of the backend signals the availability of the backend web service and also report its configuration, such as logon credential, URL of web service etc. . . The blueprint based approach eases the handling of multiple different lifecycles. For example mobile components are usually on a shorter innovation cycle compared to the backend. If the mobile application requires a certain version of the backend, the lifecycle must ensure that mobile application only gets deployed and instantiated after the backend have been updated.

[2] the one operating the records/warrant management system

Business value of the scenario

At first sight the records management scenario addresses a consumer issue of handling an ever increasing amount of electronic documents in one single place – instead of multiple locations/source, one for each provider – combined with the demand to better understand and manage contracts (i.e. analytic, categorization etc.), spending's etc. However records provider are in transition towards electronic documents motivated by cost savings for the creation and distribution process. In addition records provider expect savings in customer support as one central portal that is known to the consumer that doesn't have to learn many individual portals, including different username/password etc.- In addition the first level support, such as password reset etc., can be handled centrally by the service provider.

The relevance for consumers is supported by a report of Forrester the *“for the past decade, the number of customers using the Web to manage their bank accounts and policies and to research and buy financial products has grown steadily”* indicating the supporting likeliness of people willing to add other (less critical) documents to such services.



Base: 14,000 European Online Adults 16+ (Online Monthly or More)
Source: Forrester European Technographics Financial Services Online Survey, Q4 2011

Figure 5. Forrester: Interest in online money management tools or features[3]

Figure 5 indicated a 45% interest in *“seeing balances from all my accounts in one place”* and a 40% interest in *“organizing my spending's into categories”* and 37% interested in *“automated categorization”*.

[3] Source http://blogs.forrester.com/reineke_reitsma/12-06-22-the_data_digest_interest_in_money_management_tools

To gain more insight on potential solutions in that space, SAP conducted an online end user survey with 132 participants – half IT staff and half non IT participants - in 2012 leading to very similar results and interest in such solutions.

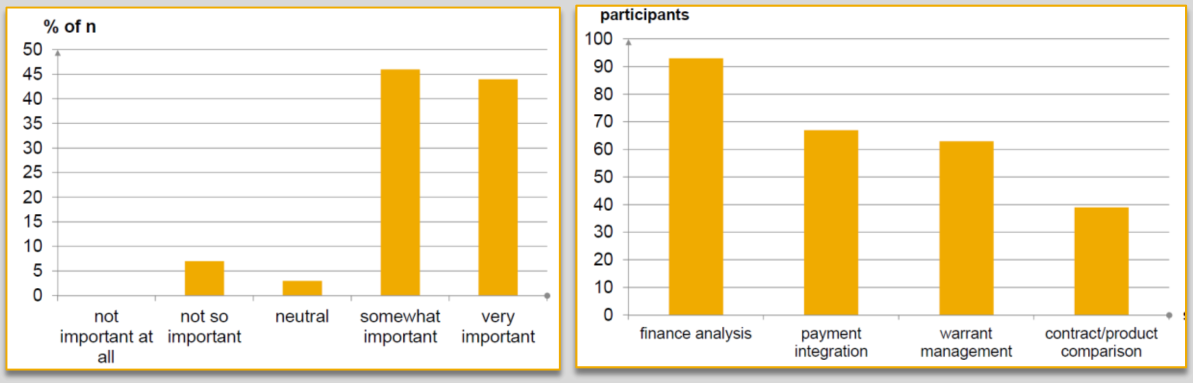


Figure 6. Results End user survey

The right hand side of Figure 6 confirms the Forrester observation of increasing interest in Personal Document Management, whereas the current solutions[4] are considered as time consuming and high manual effort. Users are most interested in financial analysis, payment integration, warrant management (this focus of the scenario implementation) and contact/product comparison. Key features judged as very valuable are

- online contracts/receipts/invoices from one single source
- automated import into a personal management and archiving tool
- central personal management tool for documentation and (e.g. financial) analysis

Whereas the main challenges are

- privacy/trust and
- ease of use.

In terms of the business case it becomes clear that such a service is only partially paid by consumers. While there may be a potential revenue for professional/extended services, the majority of revenue streams comes from the records providers that pay for the service transactional (i.e. per records stored in the system) and/or with a fee. Their benefit is in completely electronic bill, records etc. creation and distribution. The fee might be charged on a regular basis (i.e. yearly) and should be paid when the private services are deployed and configured for the first time from the 4CaaST market place. The transactional fees might also be paid via the 4CaaSt market place, but require additional extensions to monitor and report the records that will not be implemented for the scenario.

[4] without software tools for import or financial analysis

Required/matching features & contributions from 4CaaSt

For the architecture and set of applications or services for virtual private clouds a couple of 4CaaSt features are required and planned to be evaluated for the Use Case:

- Use of Blueprints to ease deployment and configuration. In general a software or service provider benefits from the higher level of abstraction when developing or operating a service. A Service provider that uses off-the-shelf software from 4CaaSt (exposed via blueprints) lowers the development efforts (contributes to CAPEX) and operation and maintenance efforts (contributed to OPEX) as he uses all “lower level” components as a service.
- Users of such applications (companies themselves) might be reluctant to sign up such an application due to data lock-in effects. However the openness of 4CaaSt will lower this effect and allow choosing other deployment options. Interoperable Clouds prevent vendor lock-in and encourage competition between vendors and operators. Once an application gets more important for a company they might be willing to get stronger availability guarantees and faster response times.

In essence the following key 4CaaSt technologies are planned to be addressed by this use case. The list below is taken from the “Description of Work” but reordered according to their respective priorities:

- **Virtualized processing / networking to lower TCO**

  For the Web Frontend (public part) of the records management application operational questions are most challenging as the service is only deployed once but afterwards needs to be scaled with the lowest possible user’s cost as it is a free service for end users. For the private deployments the situation is opposite. Since it is redeployed once for each new warrants provider, the cost of provisioning software (i.e. software updates) and data maintenance (i.e. updating master data) are most critical.

  The use case will validate this feature by describing both parts of the applications using blueprinting technologies and automatically provision them to the public cloud (one time deployment) and private cloud (repeated deployments). Maintenance is out of scope and will not be evaluated.

- **Runtime (RT) elastic scale**

  Elastic scale is one of the main characteristics to be evaluated within Task 8.3. For the records management application scaling the applications is needed especially for the web frontend (public part) for that it is used by many users and should scale accordingly. This is especially necessary as one can expect variations in the load profile due to the fact that for instance bills are usually issued at the end or at the beginning of a month. This requires additional resources (usually computing power) to be quickly added and afterwards released to optimize cost.

  The use case will validate this feature focussing on public web frontend components for which the blueprints are extended with attributes describing the components scaling options. The component is likely to be implemented stateless so stately transitions and the like are not covered.

- **High Availability**

  This section shows the high availability features of 4CaaSt’s lowest architecture layers, composed by Claudia and OpenNebula, and their usage in the scenario. The physical infrastructure counts with two hosts provided by FI-Ware and Red.es.

  The virtual infrastructure is composed by two twin application servers hosting the consumer web application of the scenario and a load balancer. The setup is similar to the landscape for elastic scalability, but application servers will be deployed in a different physical node. Each physical node will be ‘tagged’ differently using the cluster feature from OpenNebula (Example: ‘Cluster 1’ and

‘Cluster 2’). The setup can be shown using the Sunstone portal provided by OpenNebula.

The application of high-availability rules will be defined using the Claudia portal. Additionally, the conversion of OVF to OpenNebula templates will be explained. The **experiment and demo plot planned** for the following fail-over scenario is the following:

1. The service is deployed with an Active-Active configuration for redundancy and load-balancing.
2. The allocation of each service to a different cluster potentially hosted in different locations is shown.
3. One of the clusters crashes (One physical machine will be switched off).
4. We show how the system continues its operation.
5. After a certain time, the infrastructure automatically restarts the services formerly running in the failed cluster in the remaining active cluster.

- **NaaS ~~& Cloud federation~~[5]**

  The setup of the scenario with public and private parts is already a cloud federation. For the use case the private parts will therefore either be installed into a different network segment within Flexiant using their VLAN features or be installed into a “private” partition in the public cloud (described below) ~~cloud hosted by a partner (likely SAP)~~. This also leads to the evaluation of NaaS for this use case that focuses on connecting the public and private parts using a secured connection and potentially some other network SLA such as reserved bandwidth etc.

  Envisioned scenario is shown in Figure 7 below.

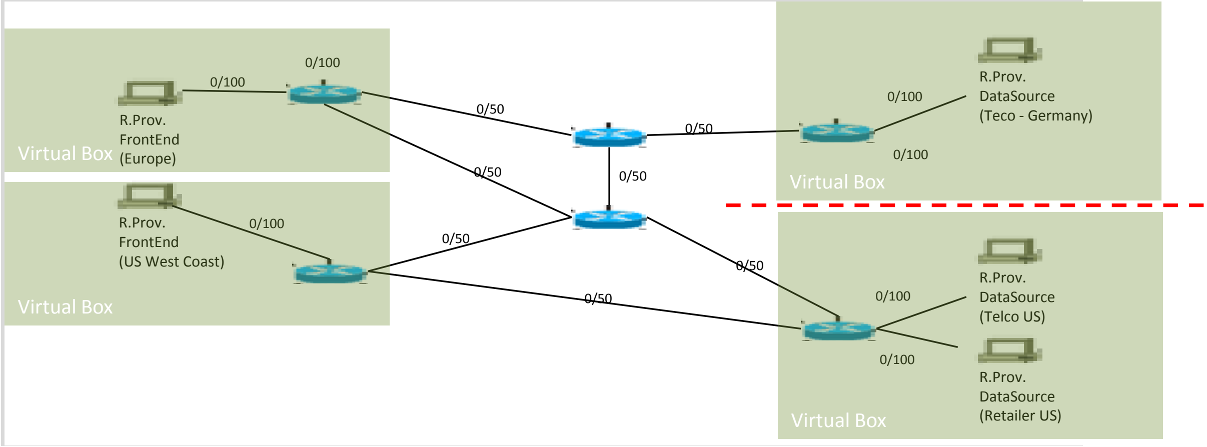


Figure 7. Potential network setup[6]

[5] There’s currently no plan to support NaaS for federated clouds

[6] Please note that the figure serves as example that is likely to be modified to demonstrate the values and features of the 4CaaSt NaaS services.

The figure shows two public web frontend on the left hand side and three records providers at the right hand side. Each of them runs on top of their own appliance with network links/connections with current/maximum bandwidth (i.e. 0 to max 50GBit/sec in the figure) and switches (blue elements in the figure) in-between. The records providers have a certain reserved bandwidth (GBit) while the NaaS tries to guarantee those with given and additional constraints, i.e. a switch failure.

For record providers the scenario distinguished two types: a) Telco providers that usually deliver their bills to the public services at the end of every month and b) a US retailer that has a bills peak end of December. This combination leads to a lack of bandwidth End of December forcing the system to reconfigure the network bandwidth. Also network isolation features are planned to be demonstrated separating (virtual) private deployments (shown as dashed, red line in the figure. Finally network failures (i.e. a network link breaking down

To support the scenario following features are planned to be implemented by the NaaS layer ordered by priority[7]:

- Support for minimum bandwidth guarantee - VMs can request minimum bandwidth for the given segment.

@Scenario: As described above the definition of bandwidth guarantees will be used in the scenario (likely in the Blueprinting or contracting phase) while the NaaS Services try to guarantee them during runtime.

- Routing to external network - Basically Internet connection to/from the VMs. No other direct access to the data center network planned in the given timeframe.

@Scenario: Likely to be used to allow access to external sources (i.e. the real backend)

- Virtual network separation - Tenants' traffic are separated.

@Scenario: Planed to be used for isolation private deployments

- Support for QoS and allocation priorities - Used when the various deployments have priorities and the deployments should take place in the order of preference.

@Scenario: Planed to be used for defining the bandwidth guarantees

- **Cloud Data Store capabilities (DSaaS)**

  While the initial implementation of the Use Case will come with its own database, the version used to test the scale (Out/In) options will replace the database implementation with a DSaaS version provided by 4CaaSt.

- **Collaboration / Coordinating / Communication**

  While the scenario could be extended especially towards collaboration between customers to share their experience on product that a customer bought (or planned to buy), the scenario will initially focus on the integration of SMS (part of the Network Enablers APIs of WP6) for notifications, such as expiration of a warrant.

[7] Note: The focus is on feature planed to be used and their contribution to the scenario. A complete set of NaaS features with details are implementation priorities are part of WP4 deliverables

- **Marketplace**

  For this use case, the marketplace will support the records management application provider in building, deploying, and optimizing the application: first, the application provider will use the marketplace to search for fitting components to realize the application, for instance front end components or data as service offerings.

  Besides technical requirements, the application provider in particular has strong business constraints with respect to costs and service levels provided by the respective components. Because the records management application will be free of charge for the end-users, the provider is interested in a low-cost implementation. The price model simulator as provided by the marketplace will therefore help the provider to analyse and optimize the overall cost of third-party services. Once deployed, the marketplace will furthermore allow the service provider to monitor and analyse the economic performance of the application. By analysing the overall business network of third-party providers of the application provider, the marketplace can also automatically recommend new business relations to increase revenues or lower costs.

  For example, the marketplace can employ the social network to search for technical solutions of other services in order to recommend combinations of products and components which are commonly used. Also, the marketplace offers market analysis functionalities based on social data to the records management application provider. Thereby, the demand from the consumers can be forecasted precisely, allowing for profound decisions to maximize the profitability of the records management application and to determine whether it pays off to add additional features.

  The agents operating on the 4CaaSt marketplace, i.e. buyer and seller agents can communicate via social tools. The data originating from this interaction is called *social data*, which describes the relationship between the agents on the marketplace. Social data is recorded and analyzed to enable *socially enhanced market analysis (SEMA)*. The goal of SEMA is to employ social data to more accurately (1.) forecast demand for given products and functionalities, and (2.) predict the behavior of other market participants in the eMP. Market analysis tools without social enhancement are typically inaccurate, e.g. due to a lack of high-precision data input for analysis. Therefore, social enhancement can support or even replace the statistical estimation approaches currently employed by market analysis tools used by electronic marketplaces. To enable SEMA in the first place, we identify the characteristics of social data emerging in business social environments and define procedures for its acquisition, quality assurance, preparation and analysis. Analysis methods include:

  - measuring the importance of individual nodes in the social graph of agents operating on the 4CaaSt marketplace.
  - approaches from artificial intelligence to perform *plan recognition*, i.e. predicting the actions of agents based on partial knowledge of their environment.

The provider will also benefit from the ability to leverage different income sources via the marketplace, for instance by offering an advertisement service that displays personalized ads to the users. The integrated payment handling of the marketplace will manage all income and spending and therefore allow the application provider to focus on his core business competencies without having to handle standard business functionality himself.[8]

For a private cloud the application, service and platform providers are usually the same persons or companies and therefore benefit from all generic OPEX gains. However additional reductions results from reusing components during the software development phases. Components provided as 4CaaSt compatible software via blueprints and a 4CaaSt Marketplace, results in lower efforts for development and maintenance of applications and/or service on top of them since the provider of the software component takes care of providing and maintaining it. The reduced development efforts contribute to lower CAPEX, while maintenance during operation reduces OPEX.

4.1. 4CaaSt Business Goals

This section outlines a set of Business Goals addresses by 4CaaSt as required by the S-Cube methodology with their use and evaluation within this use case scenario. The section concludes with other business gaols that are only indirectly used and evaluated.

4.1.1. Directly evaluated business goals

Field	**Description**
Unique ID	BG#WP2.BLUEPRINT.001
Short name	Abstract application design
Type	Business goal
Description	**A formalism shall be provided for specifying application software architecture and dependencies toward other components and technology enablers.** The system **shall** enable the software providers to design applications describing the resources (**components and technology enablers)** they need to deploy in an abstract way, providing the functional and non-functional constraints. This way, the software providers will be able to support an abstract application design that: • allows to offer software “service” in a generic way and • “use” software “service” from other providers.

[8] Note: This is described only to show an end to end process: Charging will used/demonstrated by this use case!

	This mechanism should be supported by a composition tooling that will help and drive the developer through the process specifying all the variants that make sense for a given application.
Rationale	Lower cost and length of development and maintenance by enabling software providers to specify the reuse of existing software components (accessible either in an IaaS or SaaS way) and to focus on his part of a solution in which he is the expert.
Involved Stakeholder	Service Provider, Software Provider
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1 (first iteration, focus on functional aspects) RP2 (second iteration, focus on scalability and other non-functional) RP3 (third iteration, focus NaaS)

Table 1 Business Goal - Blueprint: Abstract application design

Field	Description
Unique ID	BG#WP2.BLUEPRINT.002
Short name	Application Lifecycle Management
Type	Business goal
Description	**Platform shall provide full automation of application deployment, configuration, monitoring, healing, optimization using formalization data provided by the software.** The system shall help the software provider in the management of the lifecycle of their applications. Basically, the software provider would concentrate only on the development phase, while the deployment, taking into account the service providers' and the customers' preferences and policies would be handled by the system itself. This includes deciding which concrete services and resources to use for an application both from the functional and non-functional points of view, and both from the technical and the business point of view.
Rationale	Lower cost and length of deployment by enabling the software providers to subcontract the deployment management of their applications to the system. Be able to satisfy a broader spectrum of customers by enabling the service providers and customers to get customized deployments.
Involved Stakeholder	Service Provider, Software Provider, Customer
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1 (first iteration, focus deployment) RP2 (second iteration, focus scalability) RP3 (third iteration, focus NaaS)

Table 2 Business Goal - Application Lifecycle Management

As described above both public as well as private part of the records management deployment will be described using blueprints initially for the sake of deployment and afterwards extended with non-functional properties to a) support proper initial deployment and runtime elastic scale for the public web portal and NaaS features for the private parts. Therefore both business goals BG#WP2.BLUEPRINT.001 and BG#WP2.BLUEPRINT.002 are addressed.

Field	Description
Unique ID	BG#WP3.MARKETPLACE.001
Short name	One Stop Shop
Type	Business goal
Description	4CaaSt platform shall support the trading of any kind of Cloud Service (SaaS,PaaS,IaaS) in a unified way through the abstract description of the applications. This unified marketplace would handle different type of usage models for different types of roles: • A service provider may sell a running service to be used as SaaS in a multi-user way. • A software provider may sell an application to be deployed over concrete resources after customer contract. • A customer may contract the access to an application or the deployment of an application for own usage. • Etc. All the resources of the application, together with the own application, may be provided by different software/service providers and the generated incomes would be distributed among them according to the specified policies.
Rationale	Be able to satisfy a broader spectrum of service providers and customers by providing flexible configuration of cloud (XaaS) services and supporting multiple business models. Allow service providers to benefit from automatic settlement for the usage of their services.
Involved Stakeholder	Service Provider, Software Provider, Customer, End User
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1: publication~~, contracting~~[9] RP2: socially enhanced market analysis~~, settlement~~~~[10]~~ RP3: improved business models

[9] Decided to skip contracting for this use case; contracting will be shown in use case 1 instead

[10] This use case will focus on “socially enhanced market analysis” see section **Error! Reference source not found.**

Table 3 Business Goal - One Stop Shop

In the context of the scenario there are two potential user groups interacting with the marketplace. There is an end user that searches for the records management application for this personal use, and a provider of records that wants to join the service. In the first case a user would search for the application, register and get access to the application that is already running[11]. For the second case a records provider that wants to join the solution contracts with the service provider that provisions a new private service for the records provider[12]. The service provider also markets the solution to user and potential records providers and performs analysis with the means provided by the marketplace to search for new opportunities. The market analysis tools provided by the marketplace, employs social data in order to perform fine-grained analysis on the level of the individual user. Thereby, the demand for a product or a feature is forecasted more accurately than by previous market analysis tools.

Field	**Description**
Unique ID	BG#WP4.DEPLOY.001
Short name	Automated Provisioning and Operation for Heterogeneous Components
Type	Business goal
Description	Application component and its required services and resources shall be able to choose among different platforms to be deployed to and be operated automatically. Based on this selection 4CaaSt will be able to automatically generate suitable deployment designs and automatically provision the corresponding resources. Resource provisioning will include selection of the most appropriate deployment designs considering resource, QoS as well as scalability requirements, the automated technical construction and configuration of virtual machines and their final deployment.
Rationale	Automated deployment reduces the operational and capital invests. An service provider must not care how the underlying are build, deployed and operated and therefore concentrates on the application/service development. Automated operation contributes reduced operational cost; the application provider must not care about all required services; the PIC provider must not care about the REC

[11] Described as “A service provider can sell a running service to be used as SaaS in a multi-user way.” in Business Goal F#WP3.MARKETPLACE.001

[12] Described as “A software provider can sell an application to be deployed over concrete resources after customer contract.” in Business Goal F#WP3.MARKETPLACE.001

Involved Stakeholder	Service provider
Supporting materials	Referred to as • “Resolution process in combination with Blueprints” in the list of WP2 Innovations and • “Automated VM Construction and Provisioning for Heterogeneous Components” on WP4 Innovations[1]
Priority of accomplishment	Must have
Tentative scheduling	RP1, RP2

Table 4 Business Goal - Automated Deployment

Field	**Description**
Unique ID	BG#WP4.ELASTIC.001
Short name	Platform Elasticity / High Availability
Type	Business goal
Description	An application component running on top of 4CaaSt shall be elastic meaning that it (horizontally and vertically) scales automatically while the 4CaaSt platform will use the minimal required resources to achieve the application's required SLA. An application component deployed on a 4CaaSt cloud will scale automatically and with the minimal resources consumption in terms of virtual machines used within given boundaries such as • defined SLAs • defined scale options of the AC (stateful, stateless, etc.) • defined scale/deployment options of the underlying PICs Minimal resource means that the selected deployment will try to use the deployment that uses the least resources to fulfil the defined SLA (i.e. prefer a one low end machine setup to a high-end load balanced setup) One way of minimizing the resource consumption could be for some workload introducing multi-tenancy. High Availability is added to this business goal to show the high availability capabilities provided by 4CaaSt's lowest architecture level. This is put as an extension of these business goals instead of a new business goal as it is based on the same load balances infrastructure.
Rationale	Reduce operational cost by enforcing SLAs while using minimal resources. High Availability is a perfect extension to the scalability architecture as it uses the same infrastructure. The ratio behind High Availability is clear: reduce risks resulting from failures, i.e. hardware breakdowns.[13]
Supporting materials	Referred to as "Elasticity in the PaaS Layer" in the list of WP4 Innovations [1]
Priority of accomplishment	Must have
Tentative scheduling	RP2 Elastic scalability RP3 High Availability

Table 5 Business Goal - Automated (Horizontal / Vertical) Scale

[13] Details describe in section **Error! Reference source not found.**

Field	Description
Unique ID	BG#WP4.NAAS.001
Short name	NaaS Services - Layer 2/3 virtual networks for the deployed applications
Type	Business goal
Description	Applications and services with a defined SLA need to be able to request certain SLAs from the underlying network that connects it's components via network links within or across data centres. For some applications these network links may require certain functional and non-functional qualities, such as secure links, certain network latency or bandwidth guaranties as they affect the high level Application/Service SLAs. For example an application SLA that promises 99.9% availability requires at least 99.9% availability of all network links between the components. Therefore clients must be enabled to customize the defined networks (private or public, IP range, QoS requirements like throughput or latency) and dynamically assign virtual machines to the defined networks. IaaS Clouds provide fundamental computing resources to the consumer including processing, storage and network resources. Thus, logically NaaS is part of the IaaS offer. However, mainstream state-of-the-art IaaS Clouds provides only Layer 3 networking options that will not be sufficient for complex applications, especially in enterprise and telecommunications domains. Many use cases including legacy applications require functionalities like broadcast traffic~~, multicast networking, control over the IP address range used,~~[14] multiple networks accessed by an application, guaranteed quality of service for network connections, etc. These functions are not provided by the IaaS Layer 3 network: the possibility to allow IaaS clients to have Layer 2 networks for their own purposes is needed.
Rationale	Guarantee Application or Services level SLA for distributed applications.
Supporting materials	Referred to as "NaaS" in the list of WP4 Innovations[1].
Priority of accomplishment	Must have
Tentative scheduling	• RP2 (Design / Models) • RP3 (Implementation)

Table 6 Business Goal - NaaS

As described above both public as well as private part of the records management deployment will be described using blueprints initially for the sake of deployment and

[14] Not required by this scenario

afterwards extended with non-functional properties to a) support runtime elastic scale for the public web portal and NaaS features for the private parts. Therefore both business goals BG#WP2.BLUEPRINT.001 and BG#WP2.BLUEPRINT.002 are addressed.

4.1.2. Indirectly evaluated business cases

Monitoring

The domain assumption DA#WP5.MON.001 “Integrated Monitoring” will be evaluated only indirectly by this use case. The focus will be on SLA driven elastic scale that requires event to be fired by a monitoring subsystem. G#WP5.MON.001 “Integrated Accounting of services” will not be evaluated.

Native Services

The records management service will use one of the native services (BG#WP6.SM.001) as an example. The evaluation is planned to use just the software including their management (installation, deployment of components or packages, configuration, etc.), and provide feedback to the work package 6. No specific experiments (i.e. performance, throughput) are planned.

Immigrant PaaS Technologies

The records management service is expected to run on top of the “4CaaSt Immigrant PaaS Technologies” (BG#WP7.SM.001) - especially the application server and the database layer. This is especially true for the public part (the WebUI) while the private part may run on top of another infrastructure. The evaluation is planned to use just the software including their management stack (installation, deployment of components or packages, configuration, etc.), with feedback to the work package (WP7, WP2 and WP4) but no specific experiments are planned.

5. Use Cases (S-CUBE)

5.1.1. Provide a Software to 4CaaSt

This first use case describes the process of providing software to 4CaaSt. Most of the work that needs to be done is standard software provisioning for 4CaaSt as indicated by the grey coloured parts in the UML diagram below. Specific for this scenario and its use cases is definition of private parts and their configuration.

Field	**Description**
Unique ID	UC.8-3.001
Short name	Provide a Software to 4CaaSt
Related to	Primary: BG#WP2.BLUEPRINT.001, BG#WP2.BLUEPRINT.002 Secondary: BG#WP3.MARKETPLACE.001, BG#WP4.NAAS.001, BG#WP4.DEPLOY.001
Involved actors	• Software Provider • Service Provider
Detailed Operational Description	**Software provider perspective:** This 1st Use Case is concerned with the steps required to register a software binary to the 4CaaSt platform. As the software development process is not supported by the 4CaaSt platform, which focuses on provisioning and operation of binary software artefacts, software development steps are out of scope for this use case. Anyway one can assume that before starting development a software provider accesses the Marketplace of the 4CaaSt platform to ***“Access and select existing Blueprints”*** to a) find out what is available and supported and b) later on define the Abstract Blueprints. Therefore the software developer will use the search features provided by the 4CaaSt marketplace. Afterwards the software provider starts ***”Creating the new Application”*** and for the records management scenario develop the public and private software the Servlet/Java based artefacts. When the components are ready the software developer ***“Uploads the Artefact”*** to the 4CaaSt marketplace and ***“Defines the Abstract Blueprint”*** for both artefacts. They express a) offerings and dependencies to other 4CaaSt components (AC, PIC etc.), ***“configuration properties”***, definition of resource requirement hints for proper initial deployment, driven from the combination of various parameters (i.e. estimated workload), a flag expressing the ***“public and private parts”*** and the “one to many” connections of the public(1) and private(many) parts with required “***network SLA”***. **Service provider perspective:** After the software becomes available to the 4CaaSt platform it can be used to define a service with the software. The service provider therefore “***selects the Blueprint***” of the software described before and upon customer request triggers the ***“Deployment & Run of the software”*** including all operations (see UC.8-3.003)
Problems	The scenario deals with the questions of how difficult it is to develop or port an

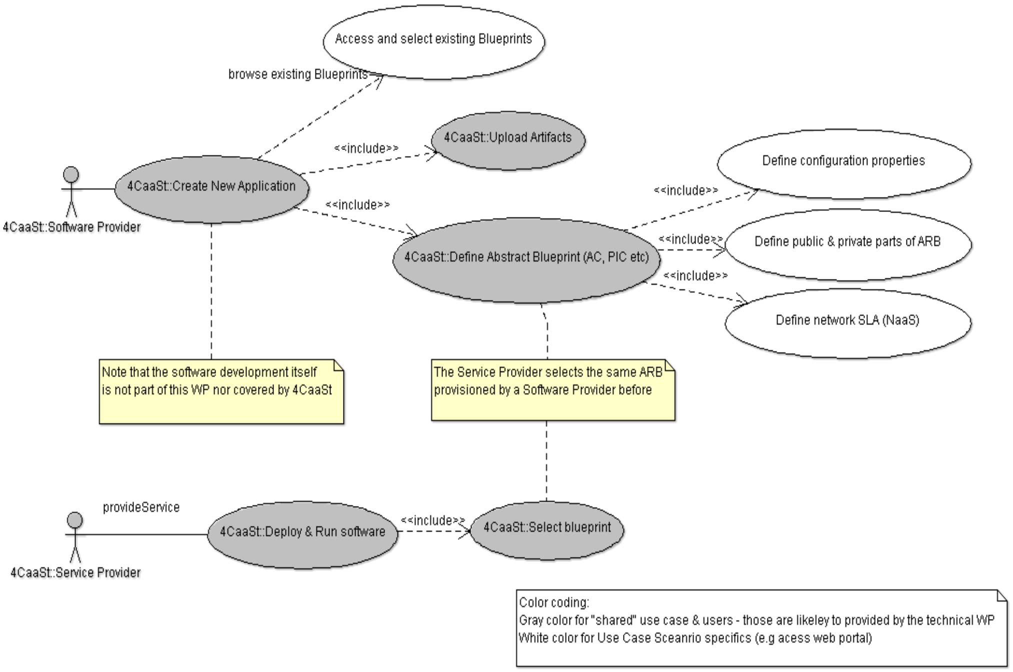
and challenges	application and make it available to the 4CaaSt platform.[15] The use case therefore challenges - effort to describe a software consisting of more than one interdependent components that themselves require other platform components using the blueprint concept and all related/require steps, such as searching for other required blueprints, uploading artefacts etc. - the scenario requires some additional configurations and properties for the blueprint descriptions such as network SLA, the dependency of private software and services to a public web interface including additional configurations that are ideally automated with the deployment
Additional material	Sub Use Case UML diagram:  Access and select existing Blueprints browse existing Blueprints 4CaaSt::Upload Artifacts <<include>> 4CaaSt::Create New Application 4CaaSt::Software Provider <<include>> Define configuration properties <<include>> 4CaaSt::Define Abstract Blueprint (AC, PIC etc) <<include>> Define public & private parts of ARB <<include>> Define network SLA (NaaS) provideService 4CaaSt::Deploy & Run software <<include>> 4CaaSt::Select blueprint 4CaaSt::Service Provider Note that the software development itself is not part of this WP nor covered by 4CaaSt The Service Provider selects the same ARB provisioned by a Software Provider before Color coding: Gray color for "shared" use case & users - those are likeley to provided by the technical WP White color for Use Case Sceanrio specifics (e.g acess web portal)

Table 7 Use Case - UC.8-3.001 Provide a Software to 4CaaSt

[15] F#WP2.01, F#WP2.02

5.1.2. Create Service from Software

Field	Description
Unique ID	UC. 8-3.002
Short name	Create Service from Software
Related to	BG#WP2.BLUEPRINT.001, BG#WP3.MARKETPLACE.001
Involved actors	• Service Provider • Service Customer
Detailed Operational Description	This scenario naturally extends the previous use case (UC. 8-3.001) from the service customer perspective. **Service provider perspective:** After the software becomes available to the 4CaaSt platform it can be used to define a service with the software. The service provider therefore "***selects the Blueprint***" of the software described before and upon customer request triggers the ***"Deployment & Run of the software"*** including all operations (see UC.8-3.003). In addition the service provider must extend the technical blueprint with business information such as price plans in the marketplace for the service to become a real product (***"Product definition and customisation"***) In addition to the previous a Service Provider may ***"Perform an analysis in the marketplace for opportunities using social networking tools"***[16]. For example, the Service Provider can: a) analyse and forecast consumer demand, requirements and satisfaction with the records management system and other similar products, and b) search for companies registered to 4CaaSt that might be interested to be added to the records management system. **Service Customer perspective:** A Service Customer (focussed using the public records management portal) uses the normal 4CaaSt contracting mechanism to get register and get access to the service. He therefore searches for the blueprint in the marketplace and ***"Contracts the Service form the Marketplace"*** that is offered from him for free and ***"Contract the Application"***[17] and ***"Configure SLA"*** by agreeing to the applications terms and conditions. As the service consumer is especially interested in elastic scalability, he focuses on defining the necessary Rules and KPI. A Service Customer that acts as a new warrant provider and therefore focuses on the private component also uses the normal 4CaaSt contracting mechanism to register and get access to the service, but in contrast maybe allowed to

[16] F#WP3.08

[17] Note: These steps "Contracts the Service form the Marketplace" and "Contract the Application"are enablers to define SLA adding automated scalbility and NaaS which is the focus for this use case. Therefore subject/focus for the use case and scenario evaluation is defining scalablity SLAs.

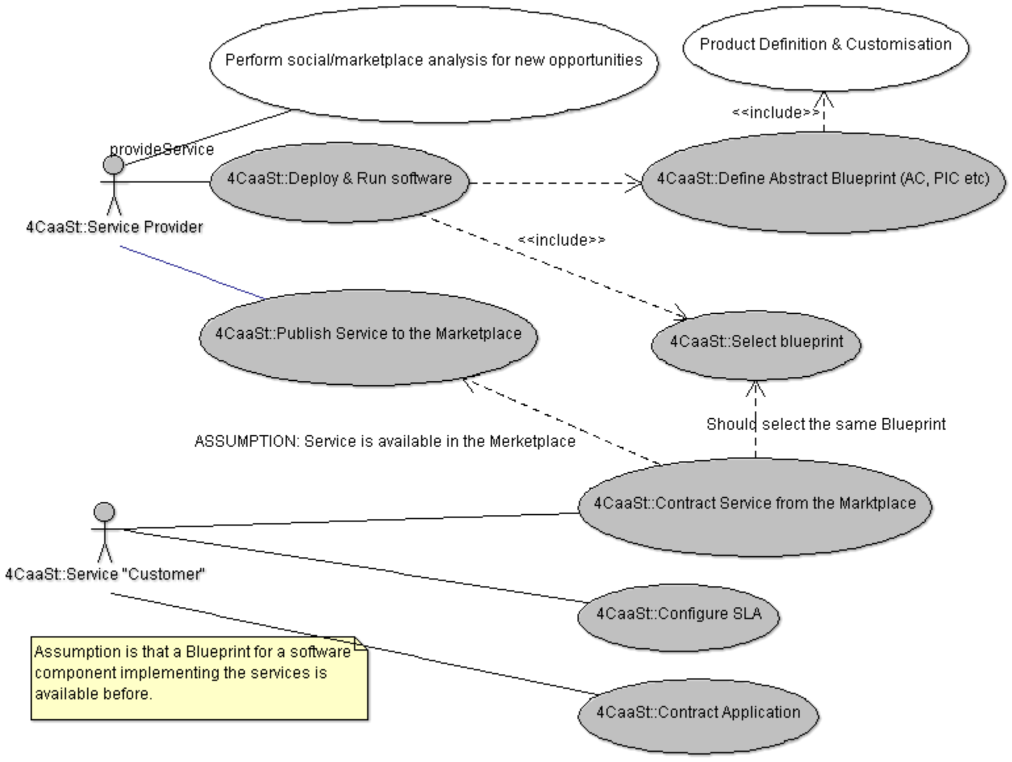
	“Configure SLA” as this service is provided individually to him.
Problems and challenges	This use case deals with the specifics of transforming software made available as binary artefacts and described using a technical 4CaaSt blueprint, into a SaaS product available via an end user marketplace, where it can be searched for and where it is advertised and targeted to certain set of users. The specific challenges are therefore • Marketplace social data analysis and recommendations for combinations of products and components which are commonly used jointly and forecast of consumers demand, allowing for profound decisions to maximize the profitability of the records management application and to determine whether it pays off to add additional features.[18] • Required action to transform software into a product (as a Service) such as adding SLA for the public Web Interfaces (→ to support scaling), price plans and customisations.[19]
Additional material	 Perform social/marketplace analysis for new opportunities Product Definition & Customisation <<include>> provideService 4CaaSt::Deploy & Run software 4CaaSt::Define Abstract Blueprint (AC, PIC etc) 4CaaSt::Service Provider <<include>> 4CaaSt::Publish Service to the Marketplace 4CaaSt::Select blueprint Should select the same Blueprint ASSUMPTION: Service is available in the Merketplace 4CaaSt::Contract Service from the Marktplace 4CaaSt::Service "Customer" 4CaaSt::Configure SLA Assumption is that a Blueprint for a software component implementing the services is available before. 4CaaSt::Contract Application UML diagrams supporting the understanding of the scenario.

Table 8 Use Case - UC.8-3.002 Create Service from Software

[18] F#WP3.05

[19] F#WP3.01, F#WP3.10, F#WP3.11

5.1.3. Solution Provisioning & Operation

Field	Description
Unique ID	UC.8-3.003
Short name	Solution Provisioning & Operation for private/public components
Related to	- BG#WP4.DEPLOY.001, BG#WP4.ELASTIC.001
Involved actors	- Service User ○ Private Service User - Service Customer - System User
Detailed Operational Description	As the majority of work load can be expected for the End User Web Frontend of the records management application these (public deployment) must scale automatically while using the least possible resource, as this service is expected to cause most of the cost, while being free to the end user. The cost is because end users will use the web interfaces frequently, while the private deployment is rather seldom. Therefore optimisation is much more useful for the public web interface that will use and implement 4CaaSt elastic scalability features. As the Web Frontend is using a typical web application server / database setup that especially well-suited for horizontal scalability, this setup will be the preferred scalability option for this scenario. The private components, which are (initially) deployed as individual deployments for each participant, multi-tenancy is a more suitable way of resource optimisation as it is expected that those deployment are very homogenous and their resource demand is rather low and constant over time. Therefore the private parts will be extended from an initial VM based deployment towards a multi-tenant deployment. Both the public and private components are ***“Provisioned and configured”*** base on the standard 4CaaSt ***“Deploy & Run software”*** as well as “***Configure Application”*** features. While this is done initially for all end users of the public service (Records Management Web UI) by the Service provider; the “***Provisioning and configuration of private components”*** are is done individually and repeated for new record/warrant providers aiming at participating in the service. The elasticity of the public components relies on standard 4CaaSt features. First the Service provider needs to ***“Configure the SLA”*** required for the public component implementing the Records Management Web UI. Based on that ***“Monitoring KPIs”*** should be configured (semi-)automatically and when the 4CaaSt platform ***“Detects a SLA risk”*** the 4CaaSt platform should trigger the appropriate horizontal ***“Scale”*** operations in order the keep the SLA (if

	technically possible). In addition to normal scale operation, it is planned to add High Availability features to the use case as **High Availability** easily extends the scalability architecture using the same load balancing infrastructure.[20]
Problems and challenges	This use case deals with service provisioning and efficient operation of software on top of and based on 4CaaSt features. The specific challenges are therefore • Although evaluation of the monitoring infrastructure is not in scope of this use case, we have to; in order to use the elastic scale features of 4CaaSt configure the monitoring infrastructure to provide the required data. • Evolving public/private parts from a single instance deployment towards ◦ an elastic operation[21] and deployment[22] of public components within defined SLAs[23] o a tenant-aware applications and multi-tenant deployment (private components) including all changed to the application, the deployment and the 4CaaSt infrastructure or landscape[24] • Finally the hybrid setup requires special network configuration to a) separate/isolate the private deployment from the public cloud (→ virtual private cloud) with b) specific allowed connections to ether the public web interface (→ intercloud) or some private backend (→ virtual private cloud). Other network QoS properties may be needed.[25]

[20] Details describe in section **Error! Reference source not found.**

[21] F#WP7.10, F#WP7.19

[22] F#WP7.11

[23] F#WP4.02

[24] F#WP7.17

[25] F#WP4.07

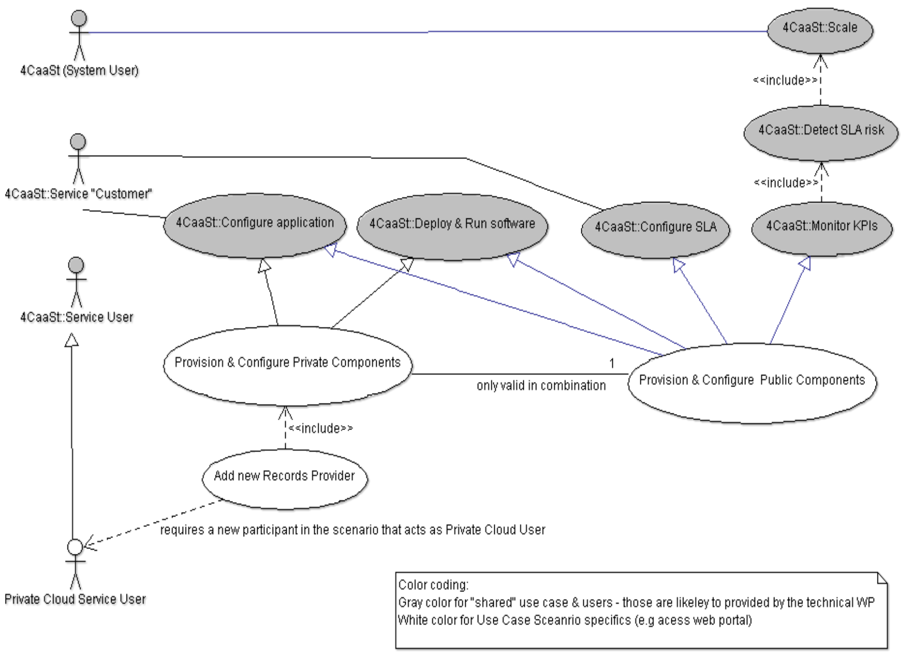
Additional material	 4CaaSt (System User) 4CaaSt::Scale <<include>> 4CaaSt::Detect SLA risk <<include>> 4CaaSt::Service "Customer" 4CaaSt::Configure application 4CaaSt::Deploy & Run software 4CaaSt::Configure SLA 4CaaSt::Monitor KPIs 4CaaSt::Service User Provision & Configure Private Components 1 only valid in combination Provision & Configure Public Components <<include>> Add new Records Provider requires a new participant in the scenario that acts as Private Cloud User Private Cloud Service User Color coding: Gray color for "shared" use case & users - those are likeley to provided by the technical WP White color for Use Case Sceanrio specifics (e.g acess web portal) UML diagrams supporting the understanding of the scenario.

Table 9 Use Case - UC.8-3.003 Solution Provisioning & Operation

5.1.4. Use Records Management Solution

Field	**Description**
Unique ID	UC. 8-3.004
Short name	Use Records Management Solution (private and public parts)
Related to	BG#WP2.BLUEPRINT.001, BG#WP4.DEPLOY.001 → Refers to the deployment of the private cloud for the records/warrant management provider. The use of the service itself, does not contribute to any of the 4CaaSt business goals.
Involved actors	• Service User ○ End User ○ Private Service User • Software Provider • System User
Detailed Operational Description	This use case describes the ***"Use of the (records*** **management) software"** of an end user and warrant/record provider with the system. **End user perspective:** A (end) user accesses the system via the ***"Records Management Web UI"*** portal. After ***"Login"*** into the system the user sees a list of the registered records (***"My Records"***) containing all kinds of records. The user can select a record from the list and perform all kinds of ***"CRUD operations on all records"***: • Read the details of the records • Updated the details of the records • Delete the record Certainly new records can be added. For records of type warrant (that are the main focus for this use case) the user just enters an ID code (read from the product or scanned as 2D barcode). The system will then retrieve warrant relevant data automatically from (any registered) warrant provider. Prior to expiring warrants the user a notification ***"SMS for expiring warrants"*** is send by the System User. **Warrant/record provider (Private Cloud Service User) perspective :** A warrant provider aiming to participate in the Records Management System performs two major tasks. First he must be registered and added to the system and second maintain his private data - warrants in this case. The registered to the Records Management System, the Warrant/record provider needs to contract potential private components (options warrant DB or Backend connection, details below). The provisioning of the private components is described in UC.8-3.003. After the provisioning is finished the Service Provider (of the records management solution) needs to ***"Register the new records provider"*** including the ***"Configuration of private components"*** so that the private components become aware of the records management system as well (i.e. the URL of the records management system). Data maintenance of this private deployment is done via the ***"Management Web Portal"*** by the warrant/record provider. Its purpose is to maintain meta data for product warrants, such as

	• the product ID (from with a 2D barcode maybe generated) • the guarantee period • the product name (optional) • addresses for local support (optional) • etc. For that purpose two potential implementations are possible. The first variant relies on a warrant database with a***” CRUD web interface for editing warrants”***, while the second variant basically provides a connection to some backend system that stores the warrant meta-data. In this case the warrant provider needs ***“Configure the connection to the backend system”***
Problems and challenges	This use case outlines the interaction with the records management service and therefore uses only a few of the 4CaaSt platform features, as deployment and operational issues are already described in the previous use cases UC. 8-3.001-003. The use case therefore only uses Integrated Telco services of 4CaaSt native service to send an SMS.[26]

[26] F#WP6.11

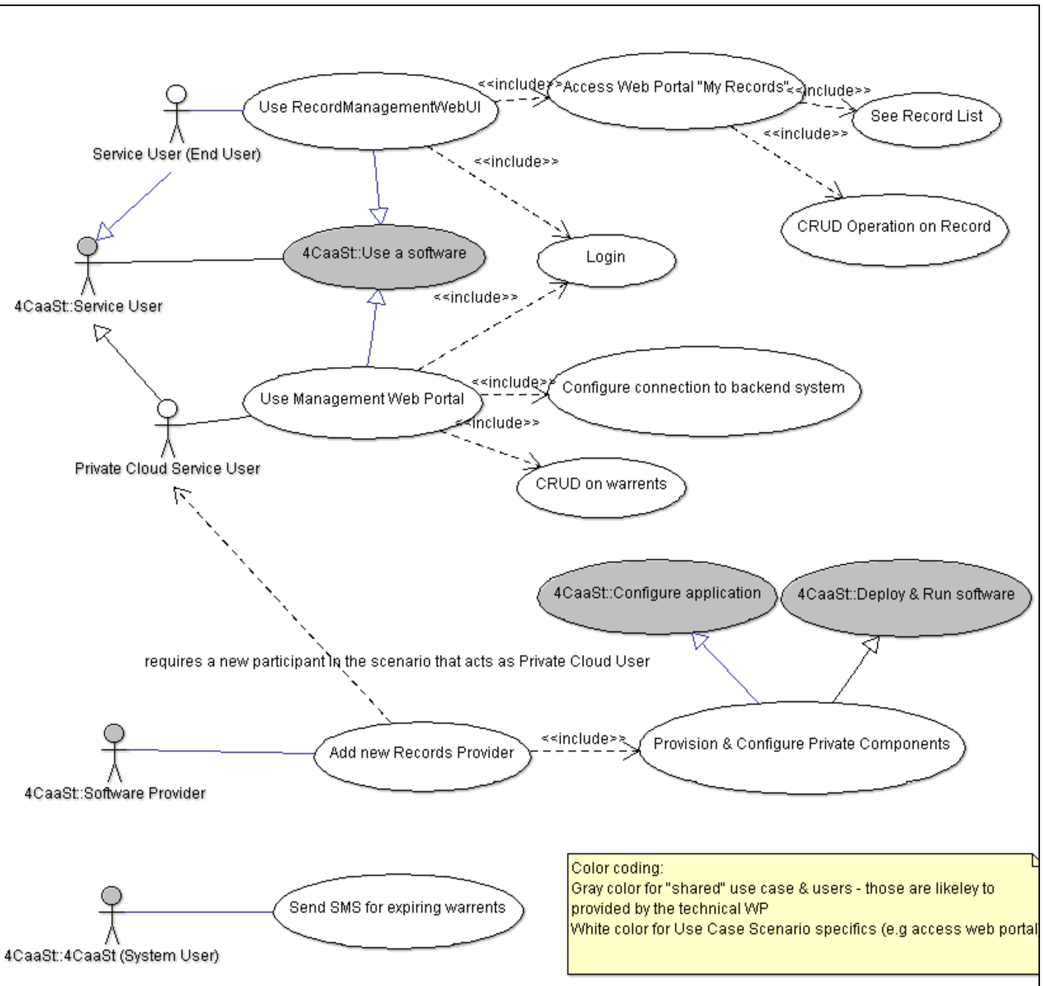
Additional material	 Service User (End User) Use RecordManagementWebUI <<include>> Access Web Portal "My Records" <<include>> See Record List <<include>> CRUD Operation on Record <<include>> Login 4CaaSt::Service User 4CaaSt::Use a software <<include>> Private Cloud Service User Use Management Web Portal <<include>> Configure connection to backend system <<include>> CRUD on warrents 4CaaSt::Configure application 4CaaSt::Deploy & Run software requires a new participant in the scenario that acts as Private Cloud User 4CaaSt::Software Provider Add new Records Provider <<include>> Provision & Configure Private Components 4CaaSt::4CaaSt (System User) Send SMS for expiring warrents Color coding: Gray color for "shared" use case & users - those are likeley to provided by the technical WP White color for Use Case Scenario specifics (e.g access web portal UML diagrams supporting the understanding of the scenario.

Table 10 Use Case - UC.8-3.004 Use Records Management Solution

6. Conclusion

This document is second documents describing the scenarios planned to be implemented on top of 4CaaSt in order to evaluate some of the business value envisioned by the project. All scenario documents follow the S-Cube methodology and structure described in D8.2.1 [2]. This document describes general setup scenario for private clouds in combination with public clouds, outlines the business value for such setups and a storyline for the records management use case.

The focus is on a records management system consisting of two parts: one end user portal deployed into the public cloud and several private deployments for each company participating is the solution. The evaluation of individual features of the 4CaaSt platform is planned for specific parts of the scenarios: automated deployment of the private and private part; scalability of the public part of the scenario. The complete solution will be offered in the marketplace. The blueprinting as the enabler will be used to describe to describe the complete solutions as well as individual parts. Building features or 4CaaSt services used are the application server and database support, as described in the experimentation report. NaaS features are currently under development used by the scenario when they become available – here no changes to the current demonstrator should be required. The network enablers require an extension of the demonstrator implementation for sending SMS as well as additional requirements in the blueprint.

Concerning the next steps described in the previous deliverable D8.3.1, the following status has been reached

- For the implementation of the scenario with results from the technical work packages, a demonstrator has been developed and ported to 4CaaSt using some of the artefacts from the technical work packages[27]
- The evaluation method, criteria and features used so far, have been described in the experimentation report.

For this scenario we plan to finally base the scenario deployment on WP6 and WP7 components and focus on operational features especially elastic scale but also high availability. Finally initial steps include NaaS and Network enabler interface are planned.

[27] Details in the experimentation report

7. References

[1] 4CaaSt FP7 Project, ”4CaaSt Value Proposition Whitepaper”, http://4caast.morfeo-project.org/publications/4caast-publications, 2011.

[2] 4CaaSt D8.2.1 “Use Case Applications eMarketPlace for the Mass Market: Scenario Definition, Deliverable D8.2.1, 2011.

[3] IBM Corporation, “Dispelling the vapor around cloud computing”, ftp://public.dhe.ibm.com/common/ssi/sa/wh/n/ciw03062usen/CIW03062USEN.PDF, 2010.

[4] SAP Enterprise Services, http://www.sdn.sap.com/irj/sdn/soa, 2011.

[5] http://www.internetnews.com/infra/article.php/3924036/VMware+Makes+Cloud+Partner+Product+News.htm, 2011.

[6] searchcloudcomputing, http://searchcloudcomputing.techtarget.com/definition/cloud-computing, 2011.

[7] E. Di Nitto, P. Plebani “Describing Case Studies: the S-Cube approach”, http://www.s-cube-network.eu/pdfs/EC-document-20100713.pdf, 2010.

[8] 4CaaSt D8.3.1 “Use Case - Virtual private Cloud for Large Corporations: Scenario Definition”, Deliverable D8.3.1, 2011.

[9] 4CaaSt D8.3.5 “Use Case Virtual Private Cloud for Large Corporations: Report on Experimentation”, Deliverable D8.3.5, 2012.

8. Annex

Annex A. Business Goals

List of all Business Goals

Wp2

Field	**Description**
Unique ID	BG#WP2.BLUEPRINT.001
Short name	Abstract application design
Type	Business goal
Description	Formalism shall be provided for specifying application software architecture and dependencies toward other components and technology enablers. The system **shall** enable the software providers to design applications describing the resources (**components and technology enablers)** they need to deploy in an abstract way, providing the functional and non-functional constraints. This way, the software providers will be able to support an abstract application design that: • allows to offer software "service" in a generic way and • "use" software "service" from other providers. This mechanism should be supported by a composition tooling that will help and drive the developer through the process specifying all the variants that make sense for a given application.
Rationale	Lower cost and length of development and maintenance by enabling software providers to specify the reuse of existing software components (accessible either in an IaaS or SaaS way) and to focus on his part of a solution in which he is the expert.
Involved Stakeholder	Service Provider, Software Provider
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1 (first iteration, focus on functional aspects) RP2 (second iteration, focus on scalability and other non-functional) RP3 (third iteration, focus NaaS)

Field	Description
Unique ID	BG#WP2.BLUEPRINT.002
Short name	Application Lifecycle Management
Type	Business goal
Description	Platform shall provide full automation of application deployment, configuration, monitoring, healing, optimization using formalization data provided by the software. The system shall help the software provider in the management of the lifecycle of their applications. Basically, the software provider would concentrate only on the development phase, while the deployment, taking into account the service providers' and the customers' preferences and policies would be handled by the system itself. This includes deciding which concrete services and resources to use for an application both from the functional and non-functional points of view, and both from the technical and the business point of view.
Rationale	Lower cost and length of deployment by enabling the software providers to subcontract the deployment management of their applications to the system. Be able to satisfy a broader spectrum of customers by enabling the service providers and customers to get customized deployments.
Involved Stakeholder	Service Provider, Software Provider, Customer
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1 (first iteration, focus deployment) RP2 (second iteration, focus scalability) RP3 (third iteration, focus NaaS)

WP3

Field	Description
Unique ID	BG#WP3.MARKETPLACE.001
Short name	One Stop Shop
Type	Business goal
Description	4CaaSt platform shall support the trading of any kind of Cloud Service (SaaS,PaaS,IaaS) in a unified way through the abstract description of the applications. This unified marketplace would handle different type of usage models for different types of roles: • A service provider may sell a running service to be used as SaaS in a multi-user way. • A software provider may sell an application to be deployed over concrete resources after customer contract. • A customer may contract the access to an application or the deployment of an application for own usage. • Etc. All the resources of the application, together with the own application, may be provided by different software/service providers and the generated incomes would be distributed among them according to the specified policies.
Rationale	Be able to satisfy a broader spectrum of service providers and customers by providing flexible configuration of cloud (XaaS) services and supporting multiple business models. Allow service providers to benefit from automatic settlement for the usage of their services.
Involved Stakeholder	Service Provider, Software Provider, Customer, End User
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1: publication, contracting RP2: social enhancement, settlement RP3: improved business models

Field	Description
Unique ID	BG#WP3.MARKETPLACE.002
Short name	Business customization
Type	Business goal
Description	Customers shall be able to configure how the applications they have contracted are going to be deployed or provisioned. Customers will define

	several non-functional business related options that will be used to decide which of the possible solutions to deploy fulfils better their preferences.
Rationale	Be able to satisfy a broader spectrum of customers by enabling them to customize their contracted applications.
Involved Stakeholder	Service Provider, Customer
Supporting materials	
Priority of accomplishment	Must have
Tentative scheduling	RP1: basic preferences RP2: role based customization RP3: dynamic roles based on history

WP4

Field	**Description**
Unique ID	BG#WP4.DEPLOY.001
Short name	Automated Provisioning and Operation for Heterogeneous Components
Type	Business goal
Description	Application component and its required services and resources shall be able to choose among different platforms to be deployed to and be operated automatically. Based on this selection 4CaaSt will be able to automatically generate suitable deployment designs and automatically provision the corresponding resources. Resource provisioning will include selection of the most appropriate deployment designs considering resource, QoS as well as scalability requirements, the automated technical construction and configuration of virtual machines and their final deployment.
Rationale	Automated deployment reduces the operational and capital invests. An service provider must not care how the underlying are build, deployed and operated and therefore concentrates on the application/service development. Automated operation contributes reduced operational cost; the application provider must not care about all required services; the PIC provider must not care about the REC
Involved Stakeholder	Service provider
Supporting materials	Referred to as • "Resolution process in combination with Blueprints" in the list of WP2 Innovations and • "Automated VM Construction and Provisioning for Heterogeneous Components" on WP4 Innovations
Priority of accomplishment	Must have
Tentative scheduling	RP1, RP2

Field	**Description**
Unique ID	BG#WP4.ELASTIC.001
Short name	Platform Elasticity
Type	Business goal
Description	An application component running on top of 4CaaSt shall be elastic meaning that it (horizontally and vertically) scales automatically while the 4CaaSt platform will use the minimal required resources to achieve the application's required SLA. An application component deployed on a 4CaaSt cloud will scale automatically and with the minimal resources consumption in terms of virtual machines used within given boundaries such as • defined SLAs • defined scale options of the AC (stateful, stateless, etc.) • defined scale/deployment options of the underlying PICs Minimal resource means that the selected deployment will try to use the deployment that uses the least resources to fulfil the defined SLA (i.e. prefer a one low end machine setup to a high-end load balanced setup) One way of minimizing the resource consumption could be for some workload introducing multi-tenancy.
Rationale	Reduce operational cost by enforcing SLAs while using minimal resources.
Supporting materials	Referred to as "Elasticity in the PaaS Layer" in the list of WP4 Innovations
Priority of accomplishment	Must have
Tentative scheduling	RP2 (stateless), RP3

Field	Description
Unique ID	BG#WP4.NAAS.001
Short name	NaaS Services - Layer 2/3 virtual networks for the deployed applications
Type	Business goal
Description	Applications and services with a defined SLA need to be able to request certain SLAs from the underlying network that connects it's components via network links within or across data centers. For some applications these network links may require certain functional and non-functional qualities, such as secure links, certain network latency or bandwidth guaranties as they affect the high level Application/Service SLAs. For example an application SLA that promises 99.9% availability requires at least 99.9% availability of all network links between the components. Therefore clients **must** be enabled to customize the defined networks (private or public, IP range, QoS requirements like throughput or latency) and dynamically assign virtual machines to the defined networks. IaaS Clouds provide fundamental computing resources to the consumer including processing, storage and network resources. Thus, logically NaaS is part of the IaaS offer. However, mainstream state-of-the-art IaaS Clouds provides only Layer 3 networking options that will not be sufficient for complex applications, especially in enterprise and telecommunications domains. Many use cases including legacy applications require functionalities like broadcast traffic, multicast networking, control over the IP address range used, multiple networks accessed by an application, guaranteed quality of service for network connections, etc. These functions are not provided by the IaaS Layer 3 network: the possibility to allow IaaS clients to have Layer 2 networks for their own purposes is needed.
Rationale	Guarantee Application or Services level SLA for distributed applications.
Supporting materials	Referred to as *"NaaS"* in the list of WP4 Innovations
Priority of accomplishment	Must have
Tentative scheduling	• RP2 (Design / Models) • RP3 (Implementation)

WP5

Field	Description
Unique ID	DA#WP5.MON.001
Short name	Integrated Monitoring
Type	Domain Assumption
Description	The system shall collect custom and built-in KPIs for any resource

	deployed and managed. This means that the system shall support capturing and aggregating specific KPIs for the different layers of the cloud stack (S/P/I/aaS), in order to enable advanced management capabilities, like scaling. In order to help developers to deploy monitorable applications, a set of tools will be provided to specify, trigger and generate the appropriate monitoring information
Rationale	Enable manageability of cloud applications and the resources they use.
Involved Stakeholder	Service Provider, Software Providers, Platform Provider
Supporting materials	None
Priority of accomplishment	Must have
Tentative scheduling	Iterative increments throughout the project.

Field	**Description**
Unique ID	BG#WP5.MON.001
Short name	Integrated Accounting of services
Type	Business Goal
Description	The system shall provide accounting for custom and built-in counters involving usage of the platform. This will allow for example detailed rating and bills to be generated. Different types of accounting capabilities can be provided so that developers and providers can make use of them and define their price models accordingly.
Rationale	Allow customers to have lower bills by enabling pay-per-use of services
Involved Stakeholder	Service Provider, Software Provider, Customer.
Supporting materials	None
Priority of accomplishment	Must have
Tentative scheduling	RP2

WP6

Field	**Description**
Unique ID	BG#WP6.SM.001
Short name	Integration of native services
Type	Business goal
Description	The software providers shall have the possibility of using external existing services and APIs in an easy way. A representative portfolio of services is needed, together with the mechanisms to easily integrate them in the applications. Software provider shall not have to manage those services nor their lifecycle.
Rationale	Lower cost and length of development and maintenance by enabling developers to integrate existing services without cost.
Involved Stakeholder	Software provider
Supporting materials	None
Priority of accomplishment	Must have
Tentative scheduling	RP2

WP7

Field	**Description**
Unique ID	BG#WP7.SM.001
Short name	Cloud aware software platforms
Type	Business goal
Description	**The system shall provide a set of well-known platform technologies available for use by software providers using basic configuration.** Therefore, software providers will not have to worry about the management of those technologies (installation, deployment of components or packages, configuration, etc.). Moreover, the software providers shall benefit from scaling and multi-tenancy features of those platforms. 4CaaSt platform shall provide facilities to integrate application servers, orchestration engines, relational databases and integration technologies like ESBs.
Rationale	Lower cost and length of development and maintenance by providing a set of cloud-aware well known technologies to support software development.
Involved Stakeholder	Software Provider
Supporting materials	Description of Work

Priority of accomplishment	Should have
Tentative scheduling	Several examples of technologies must have integrated by the end of the project.

Annex B. Glossary

Application

An Application is the software that has been developed based on a given Platform and, therefore, can be deployed and executed (hosted) on any 4CaaSt-compliant Cloud that supports such a Platform. An Application is composed by Application Components.

Blueprint

A Blueprint is the descriptor of an application component/artifact (i.e. AC, PIC…) available for a 4CaaSt-compliant Cloud. The Blueprint contains information about the component/artifact itself (e.g. its name) its abstract offerings (e.g. a servlet container) and its requirements (i.e. a Java runtime.

Abstract Resolved Blueprint / Application Blueprint

An Abstract Resolved Blueprint / Application Blueprint is the descriptor of an application stack hosted in a 4CaaSt-compliant Cloud. It's build from individual Blueprints, their offerings and requirements that are matched and results in a set of potential deployments.

Application Component

An Application Component is one of the building blocks in which the software of a given application is structured. For example, web pages, EJBs, servlets, workflow programs, etc.

Application Component Instance

An Application Component is running on top of a specific Product Instance.

***Enabl*er**

An Enabler provides/supports the offered Platform Service, allowing Application Providers to access some functionality (e.g. DBRepository, Telco Services, etc.). The Enabler facilitates the Product Instance/s to be used. Each Enabler must provide the corresponding API.

Mashup

A mashup uses and combines data, presentation or functionality from two or more sources/services to create new services. Examples maybe Apps (Android/Apple) that combines services available from the Internet to provides value added services.

Network Virtualization

In computing, Network Virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization involves platform virtualization, often combined with resource virtualization.
(Source: http://en.wikipedia.org/wiki/Network_virtualization)

Platform

A Platform is a suite of Platform Technologies.

Platform Technology

A technology that can be used to develop Application Components. A J2EE container, a BPEL container, an elastic data store, a Cloud RDBMS or an SMS/MMS service are examples of Platform Technologies. A Platform Technology is linked to a well-defined set of APIs used to program Application Components. Note that Platform Technologies are not linked to particular products (e.g., Oracle BPEL engine, Apache Tomcat or PostgreSQL) but well defined APIs. Some of these technologies can be considered “native” to the Cloud, since they were developed and flourished upon the advent of this new conception of computing. Other, “immigrant”, technologies are well-established technologies available before the age of Cloud computing (e.g. application servers, database systems, composition engines, ESBs), which need to be adapted to suit the requirements imposed in this new environment and become first-class Cloud technologies.

Product

A Product implements a given Platform Technology (E.g. Jonas, JBoss, Apache Web Server, etc.).

Product (eMarketplace)

A Product in the eMarketplace is a commercial offer that results from adding specific commercialization conditions to the actual functionality (service or application).

Platform Instance (PI)

A combination of Platform Instance Components (PIC) which materialize all technologies linked to a given Platform.

***Product Instance Component (PIC*)**

A product which on its own, or combined with other Product Instances, implements a given Technology. Examples of Product Instances would be Jonas, JBoss, Apache Web Server, Apache Tomcat, PostgreSQL, RedHat Linux, etc.

Virtual Platform

A combination of Product Instances which materialize all technologies linked to a given Platform.

***Run*time Execution Container (REC)**

It represents the VM, together with the runtime images associated to a set of Product Instances, properly configured to support the execution of a number of Application Components Instances.

Runtime Execution Container (REC) Manager

An entity in charge of the management of existing RECs in the Cloud.

ScaleOut-ScaleIn

ScaleOut means adding more computing nodes to a system and distribute software application with potentially some added resource such as a Load Balancer. The opposite of ScaleOut is ScaleIn.

ScaleUp-ScaleDown

ScaleUp means adding resources to a single node in a system, such as more CPUs, more memory or more network bandwidth. The opposite of ScaleUp is ScaleDown

Scale horizontally,

Scale vertically

See ScaleOut-ScaleIn (Scale horizontally) or ScaleUp-ScaleDown (Scale vertically)

Service

Piece of software that offers functionality ready for integration into an application. A service is to be used by developers.

Storage Object

It is the abstraction of any storage resource accessible through its URI. A container of data regardless the type or format of the data it contains.

Technology

See Platform Technology.

User & User Roles

See D1.1.1(b) User and User roles

Virtual Machine

An instance of a virtual computer, hosting a functional operating system.

Virtual Network

A virtual network is a computer network that consists, at least in part, of virtual network links. A virtual network link is a link that does not consist of a physical (wired or wireless) connection between two computing devices but is implemented using methods of network virtualization. (Source http://en.wikipedia.org/wiki/Virtual_network)

Virtual Private Network (VPN)

A virtual private network (VPN) is a computer network that uses a public telecommunication infrastructure such as the Internet to provide remote offices or individual users with secure access to their organization's network. It encapsulates data transfers between two or more networked devices which are not on the same private network so as to keep the transferred data private from other devices on one or more intervening local or wide area networks. (Source http://en.wikipedia.org/wiki/Virtual_network)

Virtual Platform

The Virtual Platform represents the instantiation of a resolved Platform. It is composed by Virtual Products which execute the Application Components. The Virtual Platform provides a uniform view of the execution context of an Application.

Virtual Product

A Virtual Product is a proxy interface to the Product Instance(s) that executes the Application Components. It allows configuring and deploying Application Components.