PaaSage

Model Based Cloud Platform Upperware

Deliverable D6.1.2

Final Requirements

Version: 1.0

PROJECT DELIVERABLE

Name, title and organisation of the scientific representative of the project's coordinator:

Mr. Philippe Rohou, GEIE ERCIM, Tel: +33 (0)4 97 15 53 06, Fax: +33 (0)4 92 38 78 22

E-mail: philippe.rohou@ercim.eu

Project website address: http://paasage.eu

Project	
Grant Agreement number	317715
Project acronym:	PaaSage
Project title:	Model Based Cloud Platform Upperware
Funding Scheme:	Integrated Project
Date of latest version of Annex I against which the assessment will be made:	3rd July 2014
Document	
Period covered:	M6-24
Deliverable number:	D6.1.2
Deliverable title	WP6 Final Requirements
Contractual Date of Delivery:	30/09/2014
Actual Date of Delivery:	30/09/2014
Editor (s):	Stefan Spahr (LSY), Dirk Muthig (LSY)
Author (s):	Maciej Malawski (AGH), Dariusz Krol (AGH), Alexander-Frederic Walser (ASCS), Franky Vanraes (BEWAN), Philippe Massonet (CETIC), Frode Finnes Larsen (EVRY), Eirik Røthe (EVRY), Morten Skylstad (EVRY), Geir Martin Elnan (EVRY), Anthony Sulistio (HLRS), Dirk Muthig (LSY), Stefan Spahr (LSY), Achilleas Achilleos (UCY), Christos Mettouris (UCY), Avgoustinos Constantinides (IBSAC)
Reviewer (s):	Michaël Van de Borne (CETIC), Edwin Yagub (GWDG), Kuan Lu (GWDG)
Participant(s):	AGH, ASCS, BEWAN, CETIC, ERCIM, EVRY, FLEX, GWDG, IBSAC, LSY, SINTEF, STFC, UCY, USTUTT-HLRS
Work package no.:	WP6
Work package title:	Requirements, Integration and Testing
Work package leader:	Stephane Mouton (CETIC)
Distribution:	PU
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	115

DISCLAIMER

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

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

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

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

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

Document History

Date	Version	Author	Description
18/3/2014	0.1	S. Spahr	Initial structure.
19/6/2014	0.2	S. Spahr	Edited traceability chapters, added mapping tables.
27/8/2014	0.3	S. Spahr	Compilation of partner documents.
02/09/2014	0.4	S. Spahr	Further integration of use cases.
03/09/2014	0.5	S. Spahr	Further integration of use cases.
09/09/2014	0.6	S. Spahr	Further integration of use cases.
23/09/2014	0.7	S. Spahr	Include the changes after the 1st review.
25/09/2014	0.8	S. Spahr	Include feedback from the use case partners.
30/09/2014	1.0	S. Spahr	Final version (1.0).

Table of Contents

List of Figures

1 Introduction

This deliverable describes the final objectives and requirements of the system resulting from the analysis iterations of the project. The deliverable will provide full traceability of the requirements with respect to the PaaSage use case domains as there are industrial cloud, eScience, and the public sector.

The traceability with respect to PaaSage components will also be specified along with the rationale for assigning the requirements to the architectural components.

Finally the deliverable will contain traceability to the integration tests that were used to verify that the integrated components met the system requirements.

The requirements described in this document are based on deliverable D6.1.1 (initial requirements) and extend the description for every scenario of each application domain. The application domains used for this document are:

- Flight scheduling (industrial sector, provided by LSY)
- Industrial Enterprise Resource Planning (industrial sector, provided by BEWAN)
- Financial services (industrial sector, provided by UCY and IBSAC)
- Complex scientific applications/data farming (eScience sector, provided by AGH)
- Resource intensive simulations including the automotive domain (eScience sector, provided by HLRS/ASCS)
- Human milk bank (public sector, provided by EVRY)

For details of these use cases see also deliverable D6.1.1, chapters 3 to 6.

The main goal of this document is to capture the requirements that different potential adopters of cloud technology will have about the way to deploy new applications or migrate existing applications onto a cloud. To reach this goal, this deliverable will describe the final scenarios for future usage of the PaaSage method and tools at a very detailed level.

This detailed description of each scenario enables all use case stakeholders to define and execute the integration tests to verify the integrated components of PaaSage.

Furthermore, the scenarios described here and the final requirements gathered are the foundation for the realisation of the demonstrators developed by WP7. Following the aforementioned sector-related structure, the demonstrators will show the applicability of the PaaSage system. Depending on the use case demonstrated, different key feature, like application optimization or process interaction, are in the focus of a particular demonstrator.

Note this deliverable is not about the detailed requirements specifications of components to be developed within technical PaaSage tasks. Such work will be carried within the PaaSage work packages WP2-5 in compliance with the architectural guidelines defined by WP1.

The deliverable will detail each of the use cases listed above. It is structured as follows:

- The general template structure used for each use case is described in Section 2.
- Sections 3 to 8 detail each use case listed using this template.
- Section 9 gives a synthesis consolidating, structuring, and highlighting common requirements across the cases.
- Section 10 summarizes with a conclusion and an outlook.

2 Introduction of the Structure

We describe here the general template that will be followed by each use case.

2.1 Objectives (revised)

This section aims at describing what each company is doing in general; what are the classes of products or processes which can be improved by using cloud computing in general and especially by using the PaaSage method. The meaning of this chapter is to recap the use case background and to highlight alignments with respect to what was described in deliverable D6.1.1

This description is refined in the subchapters:

- Selection of the use case scenario
- Overview over the prototype
- Motivation for the cloud

2.2 Scenario Description

This chapter describes the scenarios, following the standard scheme (SEI ATAM).

Below you can find some definitions e.g. the types of scalability, deployment models and other terminology which are used by the use case scenarios.

2.2.1 Definitions

2.2.1.1 Scalability

The following scaling operations are relevant for the use case.

- Scale up:
  Increases e.g. the number of cores within one node or the available main memory (RAM). There might be a restart of the application necessary or the added resources are only available for application services started after the scale-up operation.
- Scale down:
  The counterpart of scale-up, to free resources for other applications. Freeing resources might also require a restart or a shutdown of the affected application services.
- Scale out:
  This means adding more computational resources to the existing infrastructure of the system. A common use case is cloud bursting, in case of a high demand of computational power for a limited period of time.
- Scale in:
  Reduce the resources if there is no need. Saves electrical power and money and/or frees these resources for other cloud applications.

2.2.1.2 Deployment models

We distinguish the following deployment models[1]:

[1] See also http://en.wikipedia.org/wiki/Cloud_computing#Deployment_models

- Private cloud:
  A private cloud infrastructure is solely operated by a single organization. This can be done in-house or out-sourced.
- Public cloud:
  A cloud infrastructure is called public, when the services are only available over public networks (Internet); there is no direct connectivity. The services might be also publicly available.
- Hybrid cloud:
  The combination of at least two cloud infrastructures – a private cloud and a public cloud. Possible use cases include providing service to a public audience together with e.g., private computation etc.
  Another use case is cloud bursting, to extend private cloud resources during spikes in processing demands.
- Distributed (single) cloud:
  The cloud services are provided by a distributed set of machines, running at different locations while still connected to a single network (i.e. a single cloud).

More than for private cloud deployments the following aspects came to the fore for hybrid-, public or distributed cloud deployments:

<u>Data partitioning</u>

The reasons for data partitioning can be due to:

- Data confidentially restrictions or regulations:
  Company restrictions and/or legal constraints may force the location of data storage or processing.
- Near-edge location:
  Locate services and/or data near by the user to minimize the response time or reduce the network traffic.

<u>Security</u>

- Data privacy (see also data partitioning)
- Network security (use of SSH, virtual private networks etc.)
- Access control

<u>Multi tenancy:</u>

This can be treated like a virtual cloud environment, i.e. for the sake of simplicity we can reduce it to this.

2.2.1.3 Key Performance Indicators (KPIs)

In some of the use case scenarios there are references to KPIs. This is done there in an informal way. A formal KPI definition is out of scope for this deliverable and can be defined within the project in a later stage.

2.2.2 Detailed Scenarios

This chapter is filled with the complete list of scenarios covered by each use case.

Subchapters can cluster several scenarios together and the following use case template can be used to describe the quality attributes of the group:

Use Case	Use case identifier and short description.
Description	Goal to be achieved by this use case – from the point of view of the actor that initiates this use case.
Prerequisites (Dependencies) & Assumptions	References and citations relevant to the use case. Conditions that must be true for use case to be possible or to terminate it successfully.
Steps	Interactions between actors and system those are necessary to achieve the goal.
Variations (optional)	Any variations in the steps of a use case.
Quality Attributes	Quality attributes that apply to this use case. This is a short description what the actor using the target system expects.
Issues	This section describes typical issues/problems that occur; things in the current system that should be avoided in the future.

The scenario descriptions follow a standard scheme (SEI ATAM):

Scenario Id	A unique identifier to refer to the scenario easily, e.g. a combination of an abbreviation of the scenario type and a number like SO-1 (scale out) etc.
Scenario Name	A short, descriptive name.
Scenario Type	Categorization, might be refined e.g. in quality attributes, like: • Scalability • Multi customer (different time zones, load distribution etc.) • Security
Artefact	The artefact treated by the scenario.
Context	The status of the artefact and the environment (e.g. in terms of conditions) in which the stimulus arrives.
Stimulus	The event or condition arriving at the artefact, description of the source (might be internal or external), e.g.: • Latency (time interval between the stimulation and the response) • Expected load (like intensive process has finished; heavy calculation is planned soon, ...)
Response	The expected reaction or behaviour of the artefact to the stimulus. Formulate expectations with respect to process steps in

	PaaSage workflow or specific for subject system, e.g. model: • Software • Technology stack • Origin and desired target status • Stimulus condition and rules as precise as possible related to models
Response Measure	A measurable description of the response, allowing the decision whether the scenario is fulfilled. Response measure, covers: • System properties: Cost, performance, infrastructure utilization • Process properties: Cost of change, time, quality/risk • Security violations: Provider, locations

Example:

Scenario Id	**SO-LAT-1**
Scenario Name	**Scale out due to network latency**
Scenario Type	Latency reduction
Artefact	WEB-UI in conjunction with the FleetManager component
Context	The (network) latency time is higher as a configurable limit since a specified amount of time.
Stimulus	Network monitoring signals significant latency problem over the last minutes.
Response	The system moves the service onto a cloud environment with lower latency. This might be a different provider or a cloud provider near by the user of the service.
Response Measure	The average latency reported by the network monitoring facility drops below a specified limit.

2.3 Scenario Grouping

This section groups several scenarios together for easier reference in the following sections. The groups not necessarily need to be disjunctive. Below you find the structure and an example.

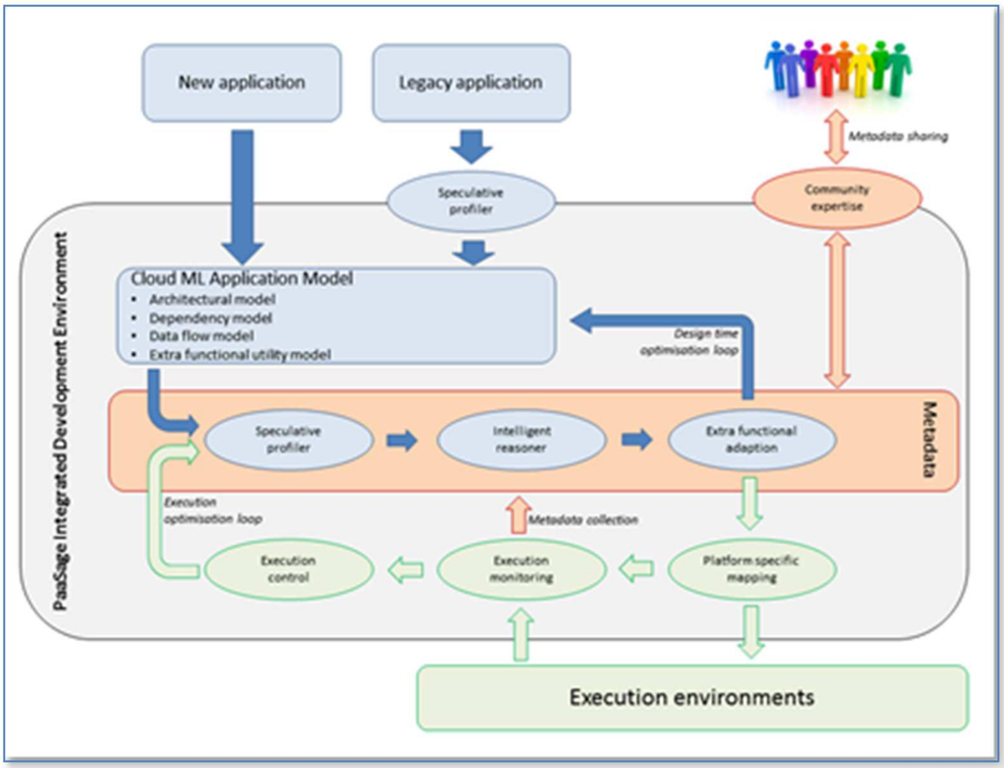
Scenario Group Id	Scenario Ids	Description

Example:

Scenario Group Id	Scenario Ids	Description
SOG-LAT-1	SO-LAT-1, SO-LAT-2, …, SO-LAT-42	Scale out scenarios triggered by latency related stimuli.
SOG-MEM-1	…	…

2.4 Traceability with respect to PaaSage components

This section describes the mapping between the classes (groups) of different scenarios and the PaaSage components.



The following table describes how each scenario group maps to the PaaSage components.

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB

Example:

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
SOG-LAT-1	Latency specification for network related components. Defined in WS-Agreement ….	The Reasoner uses the monitoring data to ….. Uses the MDDB to find a better provider which still satisfies …	Executionware measures the latency of the network related components … Components are moved to …	Data about network latency, costs and the locations serviced by different cloud providers

2.5 Traceability to the integration tests

The following table maps several Scenario groups into one integration test group. The aim of this chapter is to describe the connection of the scenario groups(s) and the different integration test scenarios.

Integration test scenario group	Scenario group Id	Description

Example:

Integration test scenario group	Scenario group Id	Description
ITG-1	SOG-LAT-1, SOG-LAT-2, SOG-FOO-1, SIG-BAR-2	Integration tests which focus on latency related scale-out/ scale-in scenarios.

3 Industrial Sector Case – Flight Scheduling

This case is supported by Lufthansa Systems (LSY).

3.1 Objectives (revised)

3.1.1 Selection of the use case scenario

From the wide variety of airline applications Lufthansa Systems offers, we selected an application from the NetLine product suite, which is used for airline schedule planning, called NetLine/Sched.

Today's airlines need to permanently revise their schedule plans in response to competitor actions, or to follow updated sales and marketing plans, while constantly maintaining operational integrity. This makes schedule management a very complex process. These challenges call for a state-of-the-art scheduling system which optimally supports the development, management and implementation of alternative network strategies. NetLine/Sched supports all aspects of schedule development and schedule management. It offers powerful and easy to use schedule visualization and modification, supports alternative network strategies and schedule scenarios and measures the profitability impacts of alternative scheduling scenarios. The system is used every day by more than 45 airlines around the globe, ranging from small to large carriers and using different business models.

3.1.2 Overview over the prototype

For the scope of the PaaSage project, LSY provides a prototype with reduced (business) functionality compared to the existing NetLine/Sched system. This is because of different reasons:

- The NetLine/Sched software is closed source.
- Current version of NetLine/Sched is not build as an application which could gain from a cloud system in a way we want to focus in PaaSage.
- With this prototype LSY will focus on demonstrating new architectural styles and technologies and not on specific business functionality.

Therefore, LSY started the development of a prototype together with another project partner, the AGH University in Krakow (done as a Master thesis). The key aspects of the prototype are:

- Provide a minimal flight scheduling service, with:
  - Legs and (aircraft) rotations
  - Some exemplary schedule validity checks
  - Some exemplary reports
- Use a simple domain with minimal business functionality on schedules, legs etc.
- Put the focus on the architecture and on cloud specific attributes

The prototype is developed in Scala (i.e. it runs on the JVM) and uses the Actor pattern (using the Akka[2] framework), Event sourcing[3], a distributed publish-subscribe event bus etc.

The architecture also makes use of the CQRS[4] pattern and is developed with the DDD[5] approach. The goal is, to build a highly scalable and distributed application, to proof the added value of the PaaSage framework and its tools.

The following architecture diagram shows the relationships between the command/query interfaces, implemented as RESTful service components and the write model as well as the one to many read model(s).

According to the CQRS pattern, the event sourcing functionality of the write model publishes so-called domain events continuously to the connected read model(s).

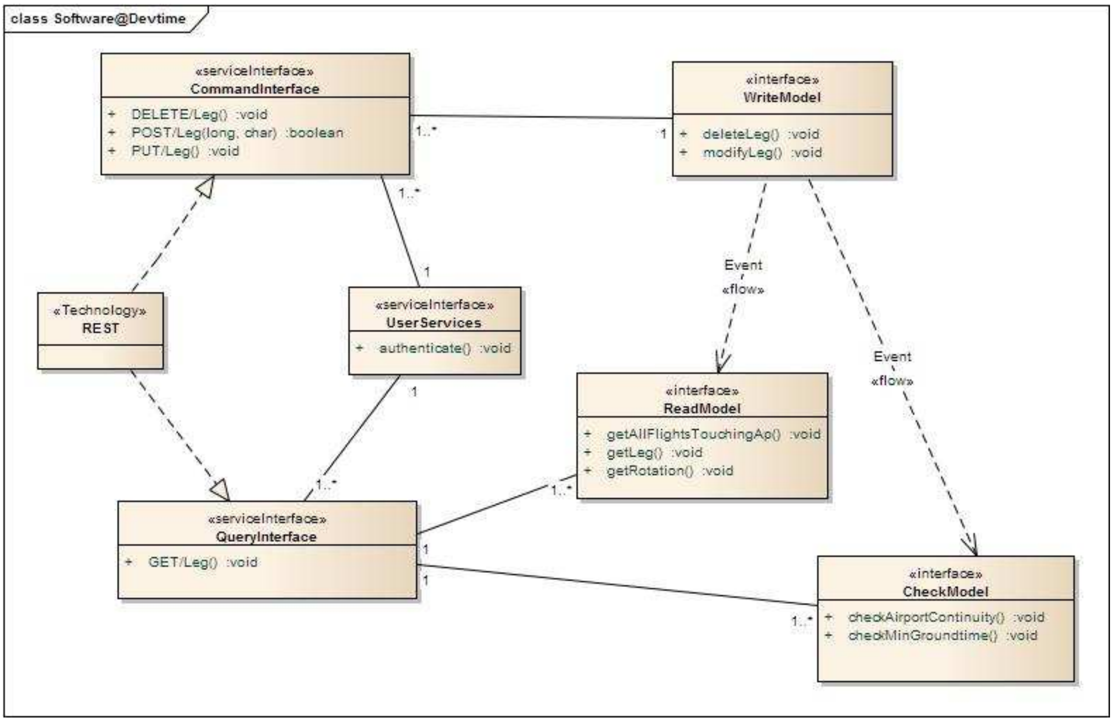


Figure 3.1: Overview of the Flight Scheduling prototype architecture

During runtime, the various deployment artefacts can be deployed into different environments. The following diagram shows only one example, using 3 nodes together with a supporting company LDAP server for the user authentication.

[2] Akka framework, http://akka.io
[3] E.g. Martin Fowler: http://martinfowler.com/eaaDev/EventSourcing.html
[4] E.g. Martin Fowler: http://martinfowler.com/bliki/CQRS.html
[5] http://en.wikipedia.org/wiki/Domain-driven_design

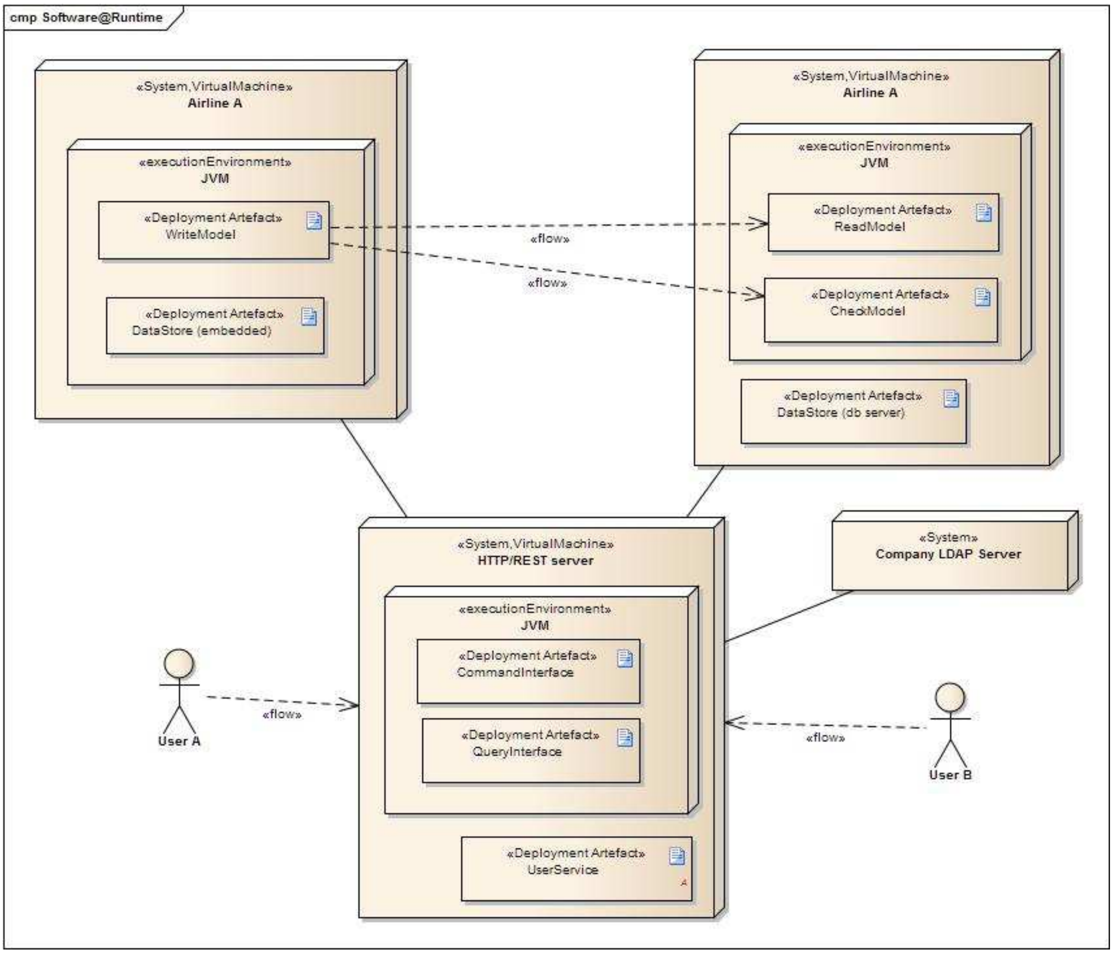


Figure 3.2: Example deployment of the Flight Scheduling prototype

3.1.3 Motivation for the Cloud

From year to year, the airline industry has the challenge of working more and more cost effectively. Cooperation's and mergers happen to use the synergetic effects and to establish the necessary market power.

To meet these challenges, the airline companies' need, amongst other things, an IT infrastructure, application landscape and system operation with high flexibility and usability. The applications must support different kinds of collaboration models, better than today.

To support such strategic alliances of individual airlines (i.e. former competitors) the companies need the aforementioned flexible infrastructure and application software. These environments must be able to perfectly scale vertically and horizontally. Therefore, besides the infrastructure, the used application software must be designed to scale and to efficiently use the given resources.

Cloud computing will be one of the key factors to achieve this flexibility. A company which develops application software to run in a cloud environment needs abstraction from specific cloud service providers to prevent a vendor lock-in, to allow shorter development cycles for new products and to gain additional benefit for the application user by providing advanced system management features.

The following chapters highlight the aspects of system operation and application development in more detail.

3.1.3.1 System Operation

As previously mentioned, Lufthansa Systems offers its customers the entire range of IT-services, including consulting, development and implementation of industry solutions, as well as operations.

From operational point of view, the use of cloud services for significant reduction of costs (as it can be realized by pure virtualization) is a major issue. This reduction will be implemented through a homogenous infrastructure by using cloud platform standards. Using the PaaSage method enables us to realize these factors also across different cloud infrastructures. Supporting deployment into hybrid clouds easily (build up on customer and provider cloud infrastructures) is another key benefit of the PaaSage method.

This homogenization of the infrastructure might be the basis for a homogeneous application landscape. This in turn improves consolidated processes around.

Lowering the heterogeneity of the infrastructure and the application landscape as well as the process diversity has a direct impact on the staff structure. There is less special qualification for people needed and due to automated control of the operation there is even less personnel needed at all.

3.1.3.2 Application development

Application development is a huge part of the Lufthansa Systems portfolio. The offered software products are flexible and highly customizable. They share data with other products whenever it makes sense.

Developing applications which are designed to run in a cloud environment will benefit from at least these topics:

- Reduced complexity
- Improved quality
- Reduced development time / reduced cost

Reduced complexity

Modularization enables us to develop in a feature-based approach. Subsystems and services are then more decoupled and well documented and therefore the demand to know every part of the system is lower than today.

Operational aspects are hidden by the cloud architecture. Standardized persistence models can be offered by the cloud environment and used by a service. Scalability is inherently supported by the cloud infrastructure if the application service is designed according to the cloud design patterns.

Improved quality

Test scenarios gains from a better modularization as well as from the cloud infrastructure itself.

Modularized systems might be tested in a down-scaled test scenario (before the integration test is executed). Only the changed services need to be tested by the developer and/or the test team.

Provisioning of an adequate test environment should be considerably easier in a cloud infrastructure than configuration of a non-virtualized, conventional environment.

Reduced development time / reduced cost

The aforementioned change of test execution of modularized systems is also reflected by the development process. A more iterative process model is supported by such service oriented architecture. The feedback loop between requirements analysis, prototyping and the customer is much more agile than before.

This will result in shorter development cycles and therefore the project can be finished with reduced cost.

3.2 Scenario Description

This chapter describes the scenarios, following the standard scheme (SEI ATAM).

It introduces first the different types of metrics and stimuli and the collection of metrics as used for the LSY use case and referenced by the following scenarios.

3.2.1 Metrics / Stimuli

This section summarizes the metric sources and metric collection which can act as stimuli e.g. for a scaling operation.

3.2.1.1 Metric sources

System

- Memory consumption:
  Free / used main memory (RAM)
- Network latency:
  Latency describes the time interval between stimulation and the corresponding response. For network connections the round-trip-latency is mostly meant. The network latency is very important for a responsive system design (e.g. a Web UI). The latency is measured in milliseconds [ms]
- Network bandwidth:
  The throughput of the network connection, measured in e.g. Mbit/s. A high network bandwidth is necessary to transfer huge amounts of data in a reasonable time (e.g. huge data sets to display on the UI, file downloads, HD media streaming etc.)
- CPU load (system or user load):
  The run-queue length of the computational processing unit (CPU) of the node in a time period (e.g. 1 minute, 5 minutes etc.). A high run-queue length indicates a high number of waiting processes to get on one of the CPU cores.

Application

<u>Execution environment / Java GC metrics</u>

Memory:

- Heap memory usage:
  The used and committed heap.
- Non-heap memory pool usage:
  The used code cache and used CMS (Concurrent Mark Sweep) permanent generation space.
- Garbage collection:
  The garbage collection CPU time.

- Class count:
  The loaded and unloaded class count for the JVM.
- Thread count:
  The current number of active threads in the JVM.
- Thread pool:
  The active and idle thread count for the pool.

<u>Computation (</u>Actor related)

- Message processing time:
  Per actor time to process one message.
- Time waiting in (queue-) mailbox:
  Message waiting time in the in-queue (aka mailbox) of one actor.
- Mailbox (queue) size:
  Number of messages waiting in the in-queue of an actor.

<u>REST / WebUI</u>

- Processing time for incoming requests:
  Time to process an incoming REST or UI request. For actor based HTTP frameworks (like akka-http), this metric relates to the actor message processing time metric.

Services

<u>Database</u>

- TX count
  Number of transactions (writes) per second.
- Query rate
  Number of executed database queries per second.

<u>Storage</u>

- Free space
- Transfer rates (similar to the bandwidth definition above).
- Response time (similar to the latency definition).

3.2.1.2 Metric collection and evaluation

The above metrics can be collected and evaluated in different ways. They can be taken as an:

- absolute value
- a trend or a moving average
- as histograms or buckets

The last metric collection can be evaluated by using percentiles. Percentile metrics can be interpreted as follows: the 95th percentile point indicates that 95% of measured values were less than the metric value. It provides a sense of how the values are distributed.

Metric collection and evaluation should be separate in the way that the collection does not make presumptions on the evaluation. This is necessary to have as much as possible data points available to switch between different evaluations or to introduce even new evaluation algorithms and run them on historic metric collections.

3.2.1.3 Components and used metrics

For the description of the scenarios, we use only a subset of the above metrics. The reason is that several metrics are subsets of others (e.g. a problematic heap memory usage of the execution engine might also lead into a main memory issue) or different metrics are handled in the same way as described in the following scenarios.

The following abbreviations are used in this table:

- 'H' for REST HTTP handler actor
- 'W' for the write model actor system
- 'C' for the read model for checks
- 'R' for the read model(s) for report

	Scale-up	**Scale-down**	**Scale-out**	**Scale-In**
System				
CPU load (user/system) [trend]	H,W,C,R	H,W,C,R	H,W,C,R	H,W,C,R
Memory consumption [trend]	H,W,C,R	H,W,C,R	H,W,C,R	H,W,C,R
Network latency [histogram/percentiles]			H,W,C,R	H,W,C,R
Network bandwidth [histogram/percentiles]			H,W,C,R	H,W,C,R
REST				
Processing time for incoming requests [histogram/percentiles]	H,W	H,W	H,W	H,W
Computation				
Mailbox size [histogram/percentiles]	W	W	W	W
Database				
Query rate			R	R
Storage				
Free space	W,R			

3.2.2 Detailed Scenarios

3.2.2.1 System CPU related

Use Case	Scaling due to CPU related issues
Description	Scale up/down or out/in due to computing resource (CPU) reasons.
Prerequisites (Dependencies) & Assumptions	To support scale up/down, the operating system and/or the application must be capable to integrate or to remove CPU resources without a restart. For scale out/in the application must be capable to distribute themselves over several nodes and also to reverse this process (aka capability to 'breathe').
Steps	(see chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	Scale out over the boundary of the (current) cloud environment, e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Elasticity of the application due to changing demands of compute power. Fulfils the SLAs with minimum costs.
Issues	High costs due to upfront allocation of computing power, still insufficient compute power for peaks and waste of computing resources for dead seasons.

Scenario Id	**SU-CPU-1**
Scenario Name	**Scale-up due to a CPU bound application component**
Scenario Type	CPU load adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The CPU load of the VM is higher than 70% since a specified amount of time.
Stimulus	VM monitoring signals a CPU bounded application-component. The trend of the CPU load indicates a steady rise or even a constant high load.
Response	The system increases the number of cores associated to the VM. If this is not possible a scale-out action needs to be triggered.
Response Measure	The CPU load reported by the VM monitoring facility drops below 70%.

Scenario Id	**SD-CPU-1**
Scenario Name	**Scale-down due to an idle or less active application component.**

Scenario Type	CPU load adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The CPU load of the VM is lower than 40% since a specified amount of time.
Stimulus	VM monitoring signals a less active or inactive application-component. The trend of the CPU load indicates a steady decrease or even a constant low load.
Response	The system decreases the number of cores associated to the VM. If this is not possible a scale-in action needs to be triggered.
Response Measure	The CPU load reported by the VM monitoring facility constantly rises above 40% and below 70%

Scenario Id	**SO-CPU-1**
Scenario Name	**Scale-out due to a CPU bound application component**
Scenario Type	CPU load adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	Relates to SU-CPU-1.
Stimulus	SU-CPU-1 triggers scale-out.
Response	The system starts a new service instance within the cluster in the same cloud or moves the cluster to a different cloud environment with more powerful CPU resources. This might also be a different cloud provider.
Response Measure	The load balancer uses the new instance and the CPU load reported by the VM monitoring facility drops below 70%.

Scenario Id	**SI-CPU-1**
Scenario Name	**Scale-in due to an idle or less active application component**
Scenario Type	CPU load adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	Relates to SD-CPU-1
Stimulus	SD-CPU-1 triggers a scale-in.
Response	The system shuts down the service instance running on a cloud provider with the highest cost (maybe the one with high number of used cores).
Response Measure	The CPU load reported by the VM monitoring facility constantly rises above 40% and below 70%

3.2.2.2 System memory related

Use Case	Scaling due to memory related issues
Description	Scale up/down or out/in due to main memory (RAM) related issues.
Prerequisites (Dependencies) & Assumptions	To support scale up/down the operating system and/or the application must be capable to integrate or to remove RAM resources without a restart. For scale out/in the application must be capable to distribute themselves over several nodes and also to reverse this process (aka capability to 'breathe').
Steps	(see chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	Scale out over the boundary of the (current) cloud environment, e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Elasticity of the application due to changing demands of main memory. Fulfils the SLAs with minimum costs.
Issues	High costs due to upfront allocation of huge amounts of main memory, still insufficient main memory for peaks and waste of main memory resources for dead seasons.

Scenario Id	**SU-MEM-1**
Scenario Name	**Scale-up due to a memory bound application component**
Scenario Type	Memory adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The memory consumption of the VM is quite high (i.e. the system starts paging/swapping etc.).
Stimulus	VM monitoring signals a memory bounded application-component. The trend of the memory consumption indicates a steady rise or even a constant high value.
Response	The system increases the amount of assigned main memory to the VM. If this is not possible a scale out action needs to be triggered.
Response Measure	The main memory usage reported by the VM monitoring facility drops below a configurable border (respectively the system stops paging/swapping).

Scenario Id	**SD-MEM-1**
Scenario Name	**Scale-down due to application components with very low memory usage.**
Scenario Type	Memory adaptation

Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The memory consumption is quite low.
Stimulus	VM monitoring signals a low memory consumption of the VM. The trend of the memory consumption indicates a steady decrease or even a constant low value.
Response	The system revokes a significant amount of assigned main memory from the VM.
Response Measure	The main memory usage reported by the VM monitoring facility indicates a better resource usage but a configurable percentage of assigned main memory is still left unused.

Scenario Id	**SO-MEM-1**
Scenario Name	**Scale-out due to a memory bounded application component.**
Scenario Type	Memory adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	Relates to SU-MEM-1.
Stimulus	Monitoring as described under SU-MEM-1 shows high memory usage, but actions defined for SU-MEM-1 cannot be implemented (e.g. due to lack of available main memory).
Response	The system starts a new service instance within the cluster in the same cloud or moves the cluster to a different cloud environment with more main memory resources. This might also be a different cloud provider.
Response Measure	The load balancer uses the new instance and the main memory usage reported by the VM monitoring facility indicates a much better resource usage.

Scenario Id	**SI-MEM-1**
Scenario Name	**Scale-in due to application components with very low memory usage.**
Scenario Type	Memory adaptation
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	Relates to SD-MEM-1
Stimulus	Monitoring as described under SD-MEM-2 shows a very low memory usage for some of the cluster nodes.
Response	The system shuts down the service instance running on a cloud provider with the highest costs.
Response Measure	The memory usage reported by the remaining VM monitoring facilities indicates a much better resource usage

3.2.2.3 System network related

Use Case	Scaling due to network related issues
Description	Scale out/in due to network latency and/or network bandwidth reasons.
Prerequisites (Dependencies) & Assumptions	To support scale out/in, the application must be capable to distribute themselves over several nodes and also to reverse this process (aka capability to 'breathe').
Steps	(see chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	Scale out over the boundary of the (current) cloud environment, e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Responsiveness of the application due to changing loads. Reacts to user inputs or M2M communication in a timely manner. Fulfils the SLAs with minimum costs.
Issues	High costs due to upfront allocation of nodes with reasonable network connection. Still insufficient network latency or bandwidth due to unavailable near-edge locations (no own data centre available etc.).

Scenario Id	**SO-LAT-1**
Scenario Name	**Scale-out due to high network latency**
Scenario Type	Latency reduction
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The (network) latency time is higher as configurable limit since a specified time. The usage of histograms and percentiles can improve the metric evaluation.
Stimulus	The Network monitoring facility indicates a significant high latency over the last (configurable) period of time.
Response	The system moves the service onto a cloud environment with lower latency. This might be a different provider. For components with extreme demands to responsibility (REST/HTTP service), the selected cloud environment should be located near to the user of the service (near-edge relocation).
Response Measure	The average latency (e.g. for the 95% percentile) reported by the network monitoring facility drops below a specified limit.

Scenario Id	**SI-LAT-1**
Scenario Name	**Scale-in due to a considerably low network latency together with high costs (for the cloud provider in use).**

Scenario Type	Latency & SLA/Cost optimization
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The (network) latency time is considerably low but at the same time the application is scattered over different cloud environments with higher costs than necessary. This can be the result of a scale-out, happened before (e.g. SO-LAT-1), together with a changed application usage profile meanwhile.
Stimulus	The Network monitoring facility indicates a considerably low latency over the last (configurable) period of time.
Response	The system shuts down the service instance running on a cloud provider with the highest cost. If the component runs into a cluster, the load balancer redirects the requests to the remaining nodes. If the service instance is the last remaining instance, the system spawns a new instance on a cheaper cloud environment.
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

Scenario Id	**SO-NBW-1**
Scenario Name	**Scale-out due to low network bandwidth**
Scenario Type	Performance improvement
Artefact	REST/HTTP service, write-model, read-model, check-model
Context	The network bandwidth is lower as configurable limit since a specified time. The usage of histograms and percentiles can improve the metric evaluation.
Stimulus	The Network monitoring facility indicates a significant low bandwidth over the last (configurable) period of time.
Response	The system moves the service onto a cloud environment with a better network connection. This might be a different provider. For components with extreme demands to responsibility together with higher amount of data to be transferred (read models) the selected cloud environment should be located near to the user of the service (near-edge relocation).
Response Measure	The average bandwidth (e.g. for the 95% percentile) reported by the network monitoring facility is higher as the specified limit.

Scenario Id	**SI-NBW-1**
Scenario Name	**Scale-in due to a considerably too high network bandwidth together with high costs (for the cloud provider in use).**
Scenario Type	Performance improvement & SLA/Cost optimization
Artefact	REST/HTTP service, write-model, read-model, check-model

Context	The network bandwidth is considerably high but this capacity is not used by the application. The application might be scattered over different cloud environments with higher costs than necessary. This can be the result of a scale-out, happened before (e.g. SO-LAT-1, SO-NBW-1), together with a changed application usage profile meanwhile.
Stimulus	The Network monitoring facility indicates a considerably high network bandwidth not used by the application over the last (configurable) period of time.
Response	The system shuts down the service instance running on a cloud provider with the highest cost. If the component runs into a cluster, the load balancer redirects the requests to the remaining nodes. If the service instance is the last remaining instance, the system spawns a new instance on a cheaper cloud environment.
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

3.2.2.4 REST/HTPP Service related

Use Case	Scaling due to processing time for incoming requests.
Description	Scale up/down or out/in due to processing time reasons. In principle, the scale out/ scenarios relates to the computing resource (CPU) scenarios. The difference is the focus on tuning of the existing parallel execution capabilities (e.g. thread pools etc.).
Prerequisites (Dependencies) & Assumptions	To support scale up/down, the operating system and/or the application must be capable to integrate or to remove changed resources (e.g. thread pool resizing) without a restart. For scale out/in, the application must be capable to distribute themselves over several nodes and also to reverse this process (aka capability to 'breathe').
Steps	(see chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	Scale out over the boundary of the (current) cloud environment, e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Elasticity of the application due to changing demands of compute power to stay responsive. Fulfils the SLAs with minimum costs.
Issues	Wasting existing compute power by misconfiguration of the application. High costs due to upfront allocation of computing power, still insufficient compute power for peaks and waste of computing resources for dead seasons.

Scenario Id	**SU-RPT-1**
Scenario Name	**Scale-up due to high processing time for incoming requests**
Scenario Type	Performance improvement
Artefact	REST/HTTP service
Context	The processing time for incoming requests is higher than a configurable limit. The usage of histograms and percentiles can improve the metric evaluation.
Stimulus	The application monitoring (e.g. using KAMON, Akka monitoring, etc.) signals a high processing time for incoming REST requests.
Response	The system tries to tune the thread pool for Akka or it increases the number of cores associated to the VM. If this is not possible a scale-out action needs to be triggered.
Response Measure	The processing time drops below the configurable limit.

Scenario Id	**SD-RPT-1**
Scenario Name	**Scale-down due to considerably low processing time together with high costs (for the cloud provider in use).**
Scenario Type	Cost reduction
Artefact	REST/HTTP service
Context	The processing for incoming requests is much faster than necessary (by the given SLAs). The usage of histograms and percentiles can improve the metric evaluation.
Stimulus	The application monitoring (e.g. using KAMON, Akka monitoring, etc.) signals a low processing time for incoming REST requests.
Response	The system tunes the thread pool for Akka (decrease no. of threads) or it decreases the number of cores associated to the VM. If this is not possible a scale-in action needs to be triggered.
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

Scenario Id	**SO-RPT-1**
Scenario Name	**Scale-out due to high processing time for incoming requests**
Scenario Type	Performance improvement
Artefact	REST/HTTP service
Context	Relates to SU-RPT-1
Stimulus	SU-RPT-1 triggers scale-out

Response	The system starts a new service instance within the cluster in the same cloud or moves the cluster to a different cloud environment with more powerful processing capabilities (CPU resources). This might also be a different cloud provider.
Response Measure	The load balancer uses the new instance and the processing time drops below the configurable limit.

Scenario Id	**SI-RPT-1**
Scenario Name	**Scale-in due to a considerably low processing time together with high costs (for the cloud provider in use).**
Scenario Type	SLA/Cost optimization
Artefact	REST/HTTP service
Context	Relates to SD-RPT-1
Stimulus	SD-RPT-1 triggers a scale-in
Response	The system shuts down the service instance running on a cloud provider with the highest cost. If the component runs in a cluster, the load balancer redirects the requests to the remaining nodes. If the service instance is the last remaining instance, the system spawns a new instance on a cheaper cloud environment.
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

3.2.2.5 Computation (write model) related

Use Case	Scaling due to application computation/processing time.
Description	Scale up/down or out/in due to computation time reasons in the write-model. In principle, the scale out/ scenarios relates to the computing resource (CPU) scenarios. The difference is the focus on tuning of the dispatcher and the existing parallel execution capabilities (e.g. thread pools etc.).
Prerequisites (Dependencies) & Assumptions	To support scale up/down, the operating system and/or the application must be capable to integrate or to remove changed resources (e.g. thread pool resizing) without a restart. For scale out/in, the application must be capable to distribute themselves over several nodes and also to reverse this process (aka capability to ‘breathe’).
Steps	Upsizing: 1. Tune the dispatcher configuration and the parallel execution capabilities (e.g. thread pool) 2. Increase the number of cores 3. Scale out

	Downsizing: 1. Scale in 2. Decrease the number of cores 3. Tune the dispatcher configuration and the parallel execution capabilities (e.g. thread pool) (see also chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	Scale out over the boundary of the (current) cloud environment, e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Elasticity of the application due to changing demands of compute power to stay responsive. Fulfils the SLAs with minimum costs.
Issues	Wasting existing compute power by misconfiguration of the application. High costs due to upfront allocation of computing power, still insufficient compute power for peaks and waste of computing resources for dead seasons.

Scenario Id	**SU-MBX-1**
Scenario Name	**Scale-up due to high (actor) mailbox size**
Scenario Type	Performance improvement
Artefact	Write-model
Context	The mailbox size of an actor is perpetually increasing in size.
Stimulus	The application monitoring signals increasing size of mailbox (queue) length.
Response	The system retunes the thread pool of the corresponding dispatcher within the limits of the used VM (no. of available cores). If the limits are reached, a different scale-up action (SU-MBX-2) needs to be triggered.
Response Measure	The mailbox size goes down to a reasonable size.

Scenario Id	**SU-MBX-2**
Scenario Name	**Scale-up due to high (actor) mailbox size w/ fully utilized thread pool**
Scenario Type	Performance improvement
Artefact	Write-model
Context	The mailbox size of an actor is perpetually increasing.
Stimulus	The application monitoring signals increasing size of mailbox (queue) length together with a fully utilized thread pool

	configuration.
Response	The system increases the number of cores associated to the VM and tunes the thread pool configuration accordingly. If this is not possible a scale-out action needs to be triggered.
Response Measure	The mailbox size goes down to a reasonable size.

Scenario Id	**SD-MBX-1**
Scenario Name	**Scale-down due to very small average (actor) mailbox size (underutilized thread pool)**
Scenario Type	Cost optimization
Artefact	Write-model
Context	The mailbox size of an actor is constantly low or even decreasing together with a corresponding thread pool configuration which causes an underutilized thread pool.
Stimulus	The application monitoring signals constantly small mailbox sizes.
Response	The system retunes the thread pool of the corresponding dispatcher (within given lower limits) of the used VM. If these limits are reached, a different scale-down action (SD-MBX-2) needs to be triggered.
Response Measure	The mailbox size may increase but stays still below the configurable limit.

Scenario Id	**SD-MBX-2**
Scenario Name	**Scale-down due to very small average (actor) mailbox size (underutilized VM)**
Scenario Type	Cost optimization
Artefact	Write-model
Context	The mailbox size of an actor is constantly low together with a corresponding thread pool configuration already adjusted to the lower limits.
Stimulus	The application monitoring signals constantly small mailbox sizes.
Response	The system decreases the number of cores associated to the VM and retunes the thread pool accordingly. If this is not possible a scale-in action needs to be triggered.
Response Measure	The mailbox size may increase but stays still below the configurable limit.

Scenario Id	**SO-MBX-1**
Scenario Name	**Scale-out due to high (actor) mailbox size**
Scenario Type	Performance optimization
Artefact	Write-model
Context	Relates to SU-MBX-2
Stimulus	SU-MBX-2 triggers a scale-out
Response	The system starts a new service instance within the cluster in the same cloud or moves the cluster to a different cloud environment with more powerful processing capabilities (CPU resources). This might also be a different cloud provider.
Response Measure	The load balancer uses the new instance and the mailbox size falls below the configurable limit.

Scenario Id	**SI-MBX-1**
Scenario Name	**Scale-in due to a very small average (actor) mailbox size (cost optimization)**
Scenario Type	Cost optimization
Artefact	Write-model
Context	Relates to SD-MBX-2
Stimulus	SD-MBX-2 triggers a scale-in
Response	The system shuts down the service instance running on a cloud provider with the highest cost. If the component runs in a cluster, the load balancer redirects the requests to the remaining nodes. If the service instance is the last remaining instance, the system spawns a new instance on a cheaper cloud environment.
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

3.2.2.6 Database query (read-model) related

Use Case	Scaling due to response time
Description	Scale out/in due to query response time reasons in the read-model.
Prerequisites (Dependencies) & Assumptions	The used database system of the read-model must support scaling, e.g. using sharding techniques etc.
Steps	(see chapter ‘3.4 Traceability with respect to PaaSage components’)
Variations	Scale out over the boundary of the (current) cloud environment,

(optional)	e.g. bursting from a private cloud into a public cloud.
Quality Attributes	Fulfils the SLAs with minimum costs. Minimize license costs.
Issues	High costs due to upfront allocation of database servers and licenses, still insufficient query response time for peaks and waste of database resources and licenses for dead seasons.

Scenario Id	**SO-DBQ-1**
Scenario Name	**Scale-out due to descending / low query response time**
Scenario Type	Performance
Artefact	Read-model
Context	The query response time of the read-model database (e.g. a MongoDB instance) is descending or constantly low which influences the reporting capabilities of the application.
Stimulus	The read-model database monitoring indicates a query bottleneck.
Response	The system starts an additional cluster node for database. This can be a replication, a sharding node etc., depending on the database technology used.
Response Measure	The query response time improves and drops below the configurable limit.

Scenario Id	**SI-DBQ-1**
Scenario Name	**Scale-in due to very low query rate / query response time is considerable low**
Scenario Type	Cost optimization
Artefact	Read-model
Context	The query rate is fairly low together with a considerably low query response time. The database is at least split into one replica or uses at least one sharding instance.
Stimulus	The database monitoring reports the low query rate in conjunction with a low query response time.
Response	The system removes on replica / shard and shuts down the VM on the node (if no longer needed).
Response Measure	Costs decrease by a substantial amount but still the SLAs are not violated.

3.2.2.7 Storage related

Use Case	Scale out due to storage memory
Description	Scale up due to lack of disk space
Prerequisites (Dependencies) & Assumptions	
Steps	(see chapter '3.4 Traceability with respect to PaaSage components')
Variations (optional)	
Quality Attributes	Data security, access speed.
Issues	High costs due to upfront allocation of storage capacity.

Scenario Id	**SU-STO-1**
Scenario Name	**Scale-up due to lack of free storage**
Scenario Type	SLA / operation
Artefact	Write-model, read-model, check-model
Context	Running short of storage capacity is foreseeable.
Stimulus	Database / storage subsystem monitoring indicates low free space on the (virtual) device.
Response	Add additional storage capacity of the same type (i.e. technology/transfer rate etc.) to the existing storage.
Response Measure	Storage capacity returns to good shape.

3.3 Scenario Grouping

Scenario Group Id	Scenario Ids	Description
SG-CPU	SU-CPU-1, SD-CPU-1, SO-CPU-1, SI-CPU-1	Scaling caused by CPU load related stimuli.
SG-MEM	SU-MEM-1, SD-MEM-1, SO-MEM-1, SI-MEM-1	Scaling caused by memory consumption related stimuli.
SG-LAT	SO-LAT-1, SI-LAT-1	Scaling caused by network latency related stimuli.
SG-NBW	SO-NBW-1, SI-NBW-1	Scaling caused by network bandwidth related stimuli.

SG-RPT	SU-RPT-1, SD-RPT-1, SO-RPT-1, SI-RPT-1	Scaling caused by processing time related stimuli.
SG-MBX	SU-MBX-1, SU-MBX-2, SD-MBX-1, SD-MBX-2, SO-MBX-1, SI-MBX-1,	Scaling caused by (actor) mailbox size related stimuli.
SG-DBQ	SO-DBQ-1, SI-DBQ-1,	Scaling caused by database query rate/query response time related stimuli.
SG-STO	SU-STO-1	Scaling caused by storage related stimuli.

3.4 Traceability with respect to PaaSage components

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
SG-CPU	The CPU related specifications are defined in CAMEL (e.g. the percentage values used in this scenario group, the metric evaluation method etc.).	The Profiler analyses the CAMEL model and provides a list of providers that matches the defined criteria's. The Reasoner uses the metrics from the Executionware monitoring facility to select the best matching provider which satisfies the cost parameters, the SLA definitions etc. For modifications of the deployment, the Adapter queries the MDDB to find a different solution which still satisfies the defined criteria's.	The monitoring facility collects all CPU related metrics. If necessary, the Executionware (adaptation and control) relocates components to different nodes or a different cloud provider.	Records summarized data sets about CPU related metric for each provider from previous runs and from the (external) PaaSage community.
SG-MEM	The memory related specifications are defined in CAMEL.	See SG-CPU.	The monitoring facility collects all main memory related metrics. If necessary, the Executionware (adaptation and control) relocates components to different nodes or a different cloud provider.	Records summarized data sets about main memory related metrics for each provider from previous runs and from the (external) PaaSage community.

SG-LAT	Network latency related specification is defined in CAMEL.	See SG-CPU	The monitoring facility collects all network related metrics. If necessary, the Executionware (adaptation and control) relocates components to different nodes or a different cloud provider.	Records summarized data sets about network related metrics for each provider from previous runs and from the (external) PaaSage community.
SG-NBW	Network bandwidth related specifications are defined in CAMEL	See SG-CPU	See SG-LAT	See SG-LAT
SG-RPT	Request processing time related specification is defined in CAMEL	See SG-CPU	The monitoring facility collects all processing time metrics from the application monitoring. If necessary, the Executionware (adaptation and control) relocates components to different nodes or a different cloud provider.	Records summarized data sets about processing time related metrics for each provider from previous runs and from the (external) PaaSage community. These metrics correlates to the SG-CPU ones.
SG-MBX	Mailbox (actor queue) sizing related specification is defined in CAMEL	See SG-CPU	The monitoring facility collects all actor-mailbox sizing related metrics from the application monitoring. Together with thread pool and CPU metrics. If necessary, the Executionware (adaptation and control) relocates components to different nodes or a different cloud provider.	Records summarized data sets about mailbox and thread pool/CPU sizing related metrics for each provider from previous runs and from the (external) PaaSage community. These metrics correlates to the SG-CPU ones.

SG-DBQ	Database query response time requirements are specified in CAMEL.	See SG-CPU	The monitoring facility collects database related metrics from the database monitoring. If necessary, the Executionware (adaptation and control) extends/shrinks replication sets or shards.	Records summarized data sets about database query response time related metrics for each provider and database engine from previous runs and from the (external) PaaSage community.
SG-STO	Storage system related specifications is defined in CAMEL	See SG-CPU	The monitoring facility collects storage related metrics from the storage monitoring facility (SAN, NAS etc.). If necessary, the Executionware (adaptation and control) extends the assigned storage space.	./.

3.5 Traceability to the integration tests

Integration test scenario group	Scenario group Id	Description
INT-SRV	SG-CPU, SG-MEM, SG-RPT, SG-MBX	Related to server infrastructure
INT-NW	SG-LAT, SG-NBW	Related to network infrastructure
INT-STO	SG-STO	Related to storage
INT-DB	SG-DBQ	Related to database technologies used

4 Industrial Sector Case – Industrial ERP

This case is supported by the BEWAN partner.

4.1 *Objectives (revised)*

BEWAN is an IT Service company located in Belgium, delivering products and services in the domain of IT infrastructure, software development and consultancy.

In the software development branch, BEWAN focalizes on ERP software (Enterprise Resource Planning). BEWAN advises, sells, customizes, delivers, implements, gives training & support in the domain of CRM, Sales & Purchase, Manufacturing, Warehouse, HR, Finance, Business Intelligence, e-Commerce, Property Management and Collaborative applications for SME's and departments of large organizations and multinationals. All of BEWAN's applications have been developed in-house and can be easily adapted in order to fulfil specific requirements from its customers. However, most of those (licensed-) applications are not cloud-ready, not SaaS-ready and therefore run on private machines.

BEWAN is in a process of redeveloping its standard applications and the strategy for the future is to offer SaaS – Multi Tenant software solutions to its customers. In order to fulfil SLAs, BEWAN needs a technology that permits appropriate deployment selection, control over the execution and automatic scaling. Thanks to the PaaSage platform, this need will be covered. "Develop once, Deploy many" is the base line of the PaaSage project and that appeals strongly to BEWAN. The involvement of BEWAN in the PaaSage project is a real opportunity to put their software and service offer one step ahead of its competitors.

4.1.1 Selection of the use case scenario

ERP is a broad domain and consists of many modules helping companies to run their businesses. One of the modules of an ERP system that can be very well designed as a cloud application is the 'after sales service' module. BEWAN will sell this application, under a SaaS model, to its customers. The actors in the 'after sales process' will be provided with an application that is accessible from different locations, using different devices, different client apps, however, sharing the same data & services. Availability from anywhere at any time and back office integration are key differentiators compared to existing manual or on premise systems.

In an after sales module, there are different actors (human or machine). The following story board illustrates the different actors and activities in such a system.

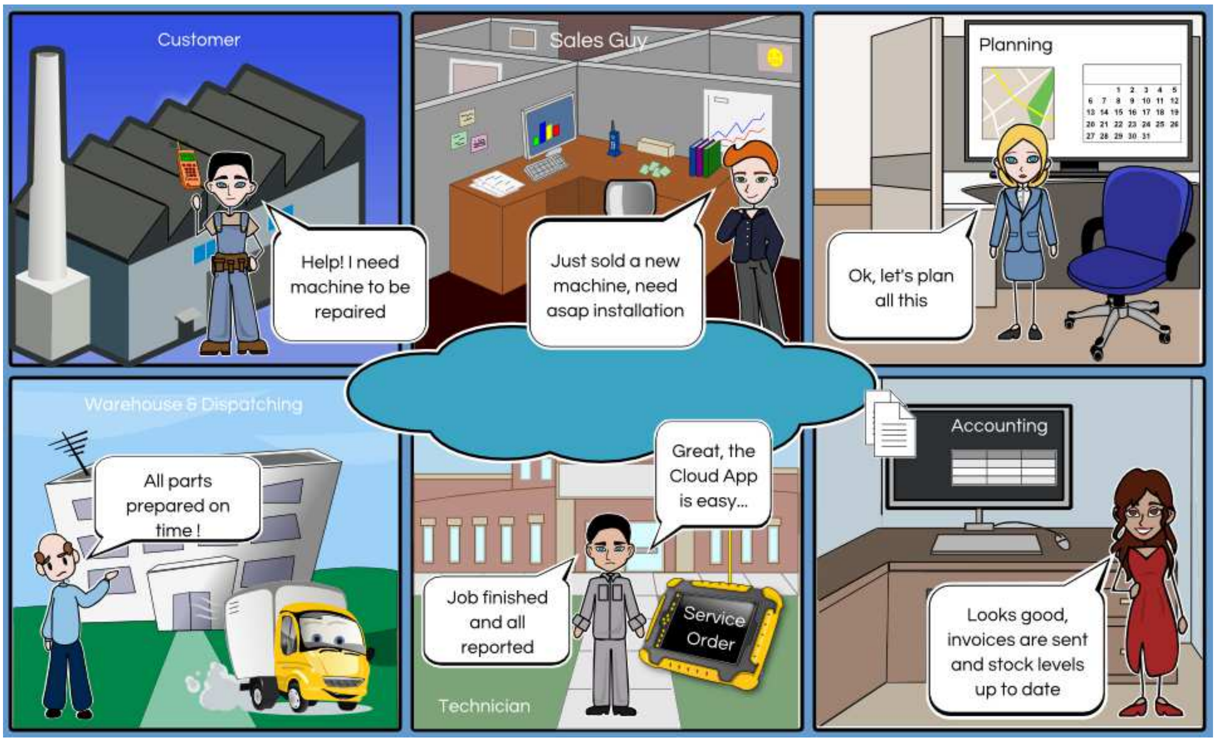


Figure 4.1: Illustration of Actors and Activities in an After Sales Process

High level overview of Actors and Activities:

- the customer requesting a service (repair, maintenance) and tracking the status
- the sales application, requesting a new installation
- the contract application, requesting recurrent maintenance tasks
- the machine itself, pushing data to the cloud (Internet of Things)
- the planner, planning the execution of tasks
- the warehouse, preparing the necessary material and spare parts
- the field technician, travelling from the dispatching to the customers and executing the tasks and reporting spent time, used parts, etc.
- the accounting application, updating stock levels, producing the invoices, etc... once the job is finished

The actors need different applications running on, or accessible from different devices. The customer will probably use his PC/web browser to connect to the customer portal in order to request a service, or he will call the service desk to request a service. The planner will use a back office client application in order to plan the tasks and pass information to the warehouse (prepare material). The technician will use a mobile application (connected to the cloud) to receive his tasks, inquire information about customers, contacts, machine history etc. and to enter data concerning the service task. After that, financial application services are needed to produce invoices based on time spent and material used, or to produce manufacturer claims in case of repairs covered by warranties.

In a traditional system, many of the activities in the process of the after sales department are done by phone, email and paper. An integrated system, highly available in the cloud and accessible anytime and from everywhere, will be far more efficient and will save a lot of work. Instead of calling, emailing, writing service reports on paper, re-entering data in the back office application, etc. everything can be done by using the cloud application. Spare parts can be ordered by using web services of manufacturers. Agendas can be shared. Technical documentation can be queried and Instant Messaging can be easily implemented. In case of unavailability of an internet connection, the technician can work offline and the application can synchronize once the connection is back. In the near future, more and more machines will be connected directly to the cloud and will report in real time status, utilization, defects, etc. (Internet of Things). Without cloud ready applications, monitoring and analysing this kind of data will be impossible.

For BEWAN, such an application can be offered in a multi-tenant SaaS model or in a private cloud model, and PaaSage will be of great value as far as the deployment and execution decisions are concerned.

4.1.2 Overview over the prototype

BEWAN has been developing a prototype of the application which will be used as a proof of concept (PoC) for the PaaSage project. The PoC does not contain all the business functionality of the final integrated ERP application; neither does it provide full multi-tenant support. On the other hand, the PoC is stable and possible PaaSage-related changes will not interfere with the ongoing development of the full ERP system, and vice versa.

The following high level diagram illustrates the business process from a functional point of view. Remark that for this PoC, not all of the activities are covered by the cloud application. Some activities remain in the back office systems, while others will be executed using the cloud application.

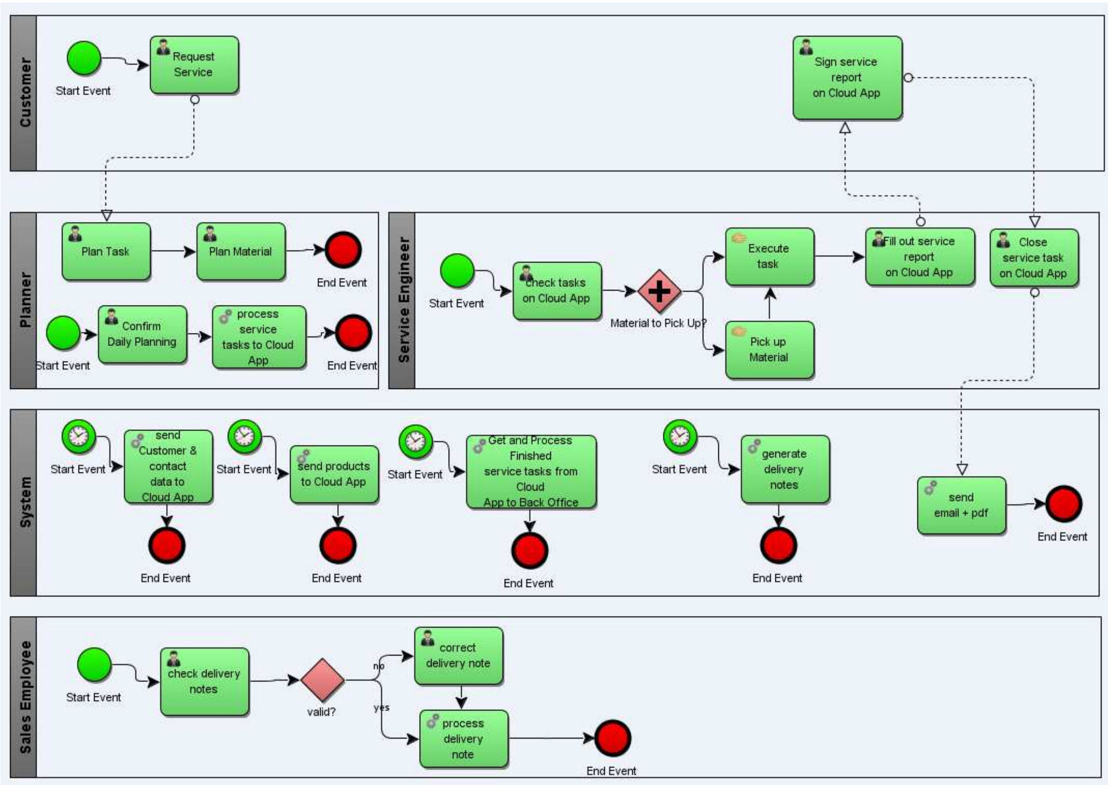


Figure 4.2: High level after sales process diagram

From an architectural point of view, and for this PoC, each tenant (each of BEWAN's customers using this application) will run the application in an isolated environment. The different tenants will connect to different URLs to access their version of the application. Each tenant will run one, or more (based on the load) instances of the application. Each environment will need an Apache webserver, a MySQL database server, an email server, the PHP binary, and of course, the application code. For the purpose of this PoC, the administrative management of the tenants will not be implemented.

In the final version of the full multi-tenant ERP application, this will be different. Except for the platform's metadata (i.e. tenants, SLAs, users, access control, database connections, application workflow customizations, extensions, plug-ins, metering & billing data) the tenant's business data will still be kept in isolated environments for each tenant. However, according to the load, one or more identical (stateless) instances of the application services will run on one or more servers to serve all the tenants. In the same way, one or more database servers will be deployed to support many databases.

Concerning security, both the PoC and the final applications will use a centralized security system (managed by the application itself).

4.1.3 Motivation for the Cloud

The SaaS scenario leads BEWAN automatically to the cloud. Depending on the availability, usage, load, or specific SLAs, BEWANs objective is to be able to deploy applications to its own cloud infrastructure but also to high performance and more scalable clouds when needed. Cloud computing offers important advantages to the user compared to classical client-server on premise execution, especially when we think of availability and redundancy, elasticity (scale up and down processors, memory, and storage), worry-free exploitation, zero infrastructure-maintenance and the accessibility through the internet from anywhere and any device.

The changing business landscape, the changing IT and applications usage, the transition from large upfront capital investment projects (Capex) to flexible, pay-per-use models (Opex), the new business development (new modern applications for new markets, wider distribution over the internet, new target customers) are for BEWAN the drivers for choosing SaaS and the cloud, and to use the PaaSage technology for supporting the deployment selection and execution control.

4.2 Scenario Description

Use Case	Scale Up / Down (SUD)
Description	Scale up or down an application deployed on the BEWAN cloud infrastructure or another initially chosen infrastructure in order to maximize performance, given the SLA, for the lowest cost
Prerequisites (Dependencies) & Assumptions	The infrastructure provides required Scale Up / Down possibilities
Steps	1. Performance KPI's (as defined in SLA) pass the thresholds 2. PaaSage detects the problem 3. PaaSage platform requests Scale Up/Down 4. PaaSage monitors and gives feedback to the provider of the application
Variations (optional)	
Quality Attributes	• Scale up/down according to increasing/decreasing MEM usage during a certain time (SUD-MEM-X scenarios) • Scale up/down according to increasing/decreasing CPU usage during a certain time (SUD-CPU-X scenarios) • Scale up/down according to increasing/decreasing I/O operations during a certain time (SUD-I/O-X scenarios) • Preventive Scale up/down according to known peak periods (SUD-PREVU-X scenarios)
Issues	

Scenario Id	**SUD-CPU-1**
Scenario Name	**Scale Up or Down due to CPU utilization**
Scenario Type	Scalability
Artefact	Instance of the application services
Context	Average CPU utilization during a configurable lapse of time exceeds a percentage (+ or -) of the configured CPU allocation
Stimulus	Monitoring of the CPU utilization, lowest cost
Response	The system allocates more or less CPUs
Response Measure	The average CPU usage in percentage of the configured CPU allocation

Scenario Id	**SUD-MEM-1**
Scenario Name	**Scale Up or Down due to memory utilization**
Scenario Type	Scalability
Artefact	Instance of the application services
Context	Average memory utilization during a configurable lapse of time exceeds a percentage (+ or -) of the configured memory allocation
Stimulus	Monitoring of the memory utilization, lowest cost
Response	The system allocates more or less memory
Response Measure	The average memory usage in percentage of the configured memory allocation

Scenario Id	**SUD-I/O-1**
Scenario Name	**Scale Up or Down due to I/O**
Scenario Type	Scalability
Artefact	Instance of the data storage services
Context	Average I/O during a configurable lapse of time is higher or lower than a configured number of I/O requests
Stimulus	Monitoring of the I/O requests, lowest cost
Response	The system starts a configurable number of additional storage services
Response Measure	The average I/O requests on running instances

Scenario Id	**SUD-PREVU-1**
Scenario Name	**Preventive Scale Up or Down due to known peak periods**
Scenario Type	Scalability
Artefact	Application services
Context	Configurable peak periods need the system to scale up or down
Stimulus	Peak periods arrived or past, lowest cost
Response	The system starts a configurable number of additional application services
Response Measure	The running instances

Use Case	Scale Out / In (SOI)
Description	Scale Out or In an application from the BEWAN cloud infrastructure or another initially chosen infrastructure to another provider (or vice versa) in order to maximize performance, given the SLA, for the lowest cost
Prerequisites (Dependencies) & Assumptions	Scale Up is not possible anymore on the original infrastructure. The new target infrastructure provides the needed resources.
Steps	1. Performance KPIs (as defined in SLA) pass the thresholds and boundaries of the current infrastructure 2. PaaSage detects the problem 3. PaaSage searches a new target infrastructure and performs a scale OUT/IN – instantiating new images on a public cloud 4. PaaSage monitors and gives feedback to the provider of the application
Variations (optional)	
Quality Attributes	• Scale Out/In when scaling up is impossible due to MEM limitations during a certain time (SOI-MEM-X scenarios) • Scale Out/In when scaling up is impossible due to CPU limitations during a certain time (SOI-CPU-X scenarios) • Scale Out/In when scaling up is impossible due to I/O limitations during a certain time (SOI-I/O-X scenarios) • Scale Out/In due to an availability issue, i.e. infrastructure up or down (SOI-AVAILABLE-X) • Scale Out/In due to network latency (SOI-NWLAT-X) • Scale Out/In due to bandwidth problem (SOI-BANDW-X) • Preventive Scale Out/In when scaling up is impossible according to known peak periods (SOI-PREVU-X scenarios)
Issues	

Scenario Id	**SOI-CPU-1**
Scenario Name	**Scale Out or In due to CPU usage**
Scenario Type	Scalability
Artefact	Instance of the application services
Context	Average CPU utilization during a configurable lapse of time exceeds a percentage (+ or -) of the configured CPU allocation and allocating more CPU is not possible on the private cloud
Stimulus	Monitoring of the CPU utilization and limits, lowest cost
Response	The system starts / stops instances on a public cloud
Response Measure	The average CPU usage in percentage of the configured CPU allocation

Scenario Id	**SOI-MEM-1**
Scenario Name	**Scale Up or Down due to memory utilization**
Scenario Type	Scalability
Artefact	Instance of the application services
Context	Average memory utilization during a configurable lapse of time exceeds a percentage (+ or -) of the configured memory allocation and scaling up is not possible
Stimulus	Monitoring of the memory utilization, lowest cost
Response	The system starts / stops instances on a public cloud
Response Measure	The average memory usage in percentage of the configured memory allocation

Scenario Id	**SOI-I/O-1**
Scenario Name	**Scale Up or Down due to I/O**
Scenario Type	Scalability
Artefact	Instance of the data storage services
Context	Average I/O during a configurable lapse of time is higher or lower than a configured number of I/O requests and scale up is not possible
Stimulus	Monitoring of the I/O requests, lowest cost
Response	The system starts / stops an instance of the storage services on a public cloud
Response Measure	The average I/O requests on running instances

Scenario Id	**SOI-AVAILABLE-1**
Scenario Name	**Scale out / in based on availability of the private cloud**
Scenario Type	Scalability
Artefact	Application services
Context	The private cloud is unavailable (or returns to available)
Stimulus	The private cloud is available or not, lowest cost
Response	The system starts a configurable number of instances on a public cloud or restarts the instances on the private cloud when available again
Response Measure	Availability / Unavailability

Scenario Id	**SOI-NWLAT-1**
Scenario Name	**Scale out / in according to network latency**
Scenario Type	Scalability
Artefact	Application services – User Experience
Context	The average network latency during an amount of time is higher / lower than configured
Stimulus	Monitoring of the network, lowest cost
Response	The system starts / stops new instances on public cloud(s) closer to the users
Response Measure	The average network latency / response time

Scenario Id	**SOI-BANDW-1**
Scenario Name	**Scale out / in according to available bandwidth**
Scenario Type	Scalability
Artefact	Application services – User Experience
Context	The average available bandwidth during an amount of time is lower / higher than configured
Stimulus	Monitoring of the network, lowest cost
Response	The system starts / stops new instances on public cloud(s) closer to the users and/or the data storage
Response Measure	The average network latency

Scenario Id	**SOI-PREVU-1**
Scenario Name	**Preventive Scale Up or Down due to known peak periods**
Scenario Type	Scalability
Artefact	Application services
Context	Configurable peak periods need the system to scale out or in
Stimulus	Peak periods arrived or past, lowest cost
Response	The system starts a configurable number of instances
Response Measure	The running instances

Use Case	Processing and Data Location (LOC)
Description	Legal restrictions or company policies can oblige the physical location of data storage and/or processing, for example data must be stored in the country where the company (BEWAN's customer) is a legal entity, or data must be stored inside the company's datacentre, or even application must run and data must reside on the company's cloud infrastructure.
Prerequisites (Dependencies) & Assumptions	PaaSage supports private/on premise clouds
Steps	1. PaaSage examines the SLA for a requested deployment and identifies constraints on deployment/execution location and/or data location 2. PaaSage searches the target infrastructure that meets the SLA and performs a deployment or instantiates accordingly 3. PaaSage takes into account this requirement when scaling out (see other Use Cases) 4. PaaSage gives feedback about the deployment (e.g. what has been deployed where etc.).
Variations (optional)	
Quality Attributes	• SLA defined location of data (LOC-1) • SLA defined location of processing (LOC-2)
Issues	

Scenario Id	**LOC-1**
Scenario Name	**Deployment and execution issue due to processing localization restrictions**
Scenario Type	Processing Location
Artefact	Application services
Context	SLA’s require that application services are running on machines in a given region, country, area or on a private cloud
Stimulus	SLA requirement, lowest cost
Response	The system instantiates application services on an SLA compatible cloud
Response Measure	Location of the running application services

Scenario Id	**LOC-2**
Scenario Name	**Data storage issue due to storage location restrictions**
Scenario Type	Data Location
Artefact	Storage services
Context	SLAs require that data is stored in machines in a given region, country, area or in a private cloud
Stimulus	SLA requirement, lowest cost
Response	The system instantiates storage services on an SLA compatible cloud
Response Measure	Location of the data

4.3 Scenario Grouping

Scenario Group Id	Scenario Ids	Description
SUD	SUD-MEM SUD-CPU SUD-I/O SUD-PREVU	Scaling Up/Down scenarios triggered by performance KPIs, preventive scenarios and cost minimisation on the BEWAN cloud or on another cloud provider
SOI	SOI-MEM SOI-CPU SOI-I/O SOI-AVAILABLE SOI-NWLAT SOI-BANDW SOI-PREVU	Scaling Out/In scenarios triggered by performance KPIs, preventive scenarios, availability, network KPIs, availability and cost minimisation
LOC	LOC	Deployment and execution restricted by data and/or processing location restrictions and cost considerations

4.4 Traceability with respect to PaaSage components

The following table describes how each scenario group maps to the PaaSage components.

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
SUD	Scaling and cost criteria defined in the application model	The Profiler analyses the application model and produces a list of providers that satisfy the requirements. The Reasoner uses the list of providers, monitoring data, scaling & cost, rules to select a provider to deploy new instances. The Adaptor generates commands for (re)deployment.	Executionware executes the commands generated by the adapter on the provider selected by the Reasoner. Executionware monitors the execution. Application services are instantiated or stopped, infrastructure resources are re-allocated.	Storing monitoring data, provider data, cost of usage, etc.
SOI	Scaling and cost criteria	The Profiler analyses the application model	Executionware executes the commands generated	Storing monitoring

	defined in the application model	and produces a list of providers that satisfy the requirements. The Reasoner uses the list of providers, monitoring data, scaling & cost, rules to select a provider to deploy new instances. The Adaptor generates commands for (re)deployment	by the adapter on the provider selected by the Reasoner. Executionware monitors the execution. Application services are instantiated or removed.	data, provider data, cost of usage, etc.
LOC	Data and processing location specified in the application model	The Profiler analyses the application model and produces a list of providers that satisfy the SLA. The Reasoner uses the list of providers, location data, and cost rules to select a provider to deploy new instances. The Adaptor generates commands for (re)deployment.	Executionware collects information about data location and usage of running instances.	Storing information about data location and usage.

4.5 *Traceability to the integration tests*

The following table maps several scenario groups into one integration test group. The aim of this chapter is to describe the connection of the scenario group(s) and the different integration test scenarios.

Integration test scenario group	Scenario group Id	Description
ITG-SCALE	SUD SOI	Integration testing covering scale up and down, Integration testing concerning sale out and back in
ITG-LOC	LOC	Integration testing covering location requirements

5 Industrial sector - Financial services

This case is supported by the UCY and IBSAC Intelligent Business Solutions Ltd partners.

5.1 *Objectives (revised)*

5.1.1 Selection of the use case scenario

Cyprus is one of the largest financial centres in Europe and the Middle East. In particular, Cyprus has a large financial and auditing services sector with large firms including the Big Four – Deloitte, PWC, Ernst & Young, KPMG and several local firms. Figure 5.1 shows that the financial and insurance sector has the biggest percentage in terms of economic activities in 2010. Thus, today's financial firms and internal accounting departments need Corporate Administration software in order to work faster and effectively with the vast amount of companies that are managing, while constantly maintaining operational integrity and full compliance with international standards. These challenges call for a state-of-the-art corporate administration solution which optimally supports all the necessary functionalities.

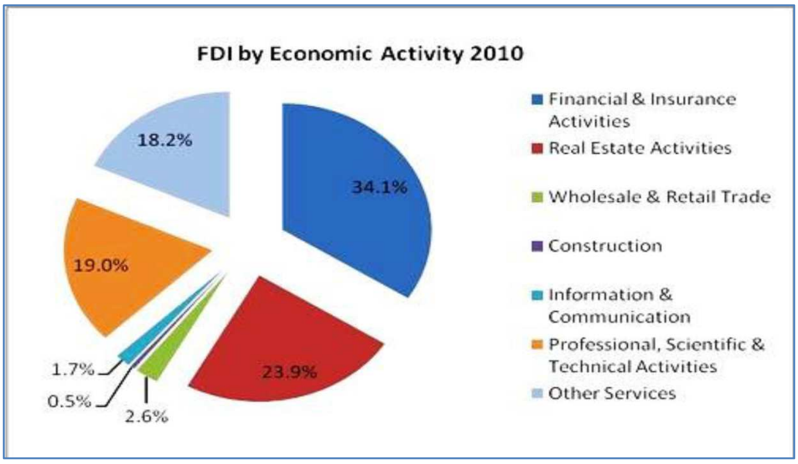


Figure 5.1: Foreign Direct Investment (FDI) By Economic Activity 2010[6]

Yet no dedicated cloud-based corporate solution exists, but rather proprietary solutions for each type of corporate administration functions. This provides a large and diverse market based on the diverse cloud platforms currently employed by these firms for other company technological activities. From a wide variety of business applications and services that IBSAC Intelligent Business Solutions Ltd offers and supports, an application from the financial sector, namely Infoscreen Quorum, was selected. The application is used for the preparation of all legal and government papers and also generation of several financial statements in full compliance with the International Financial Reporting Standards.

5.1.2 Overview over the prototype

The financial application is used principally by accountant and law firms throughout the island of Cyprus. The aforementioned application works on the basis of a

[6] Source: Cyprus Promotion Investment Agency http://www.cipa.org.cy/easyconsole.cfm/id/131

client/server architecture. In particular, all clients need to install the software as a Windows application, which is used to communicate with the application server. Based on user requests, the application server retrieves the data from the database server and the Windows application displays the data on a user-friendly interface in a simple format that the user is able to easily comprehend and interact with.

Currently, the application server and the database server are located inside of the company's internal network using the on premise application. However, the final outcome with PaaSage will be to provide the ability and option for the clients of IBSAC to host application and database servers in the cloud. Moreover, PaaSage will provide the capability for IBSAC clients, for instance to select their cloud provider (e.g., based on costs) that will permanently host the financial application, un-deploy from private cloud and deploy to a public cloud that offers more resources required in a limited period of time (e.g., corporate levy payment period).

Hence, using PaaSage the financial application will be able to be totally or partially hosted in the cloud based on IBSAC clients' preferences. For instance, application and database server could be located in a public cloud, instead of being in-house on a private cloud, based on the requirements of the customer. In other cases, since several customers are reluctant to use the public cloud due to confidentiality reasons, the database server could stay in-house on the private cloud. Even though with this scenario, companies will be able to have savings on the purchase and maintenance of the application server. In fact using PaaSage it will be straightforward, almost "with a touch of a button", to transfer their existing application server in their preferred cloud provider. However, for organizations that do not have any issues with confidentiality, they will be able to host both application and database server in the cloud in order to get all the benefits of cloud hosting.

5.2 Scenario Description

In this section the different scenarios currently envisioned for the financial use case are described in detail using the SEI ATAM Quality Attribute Scenario template.

Use Case	**Scale** Out
Description	Scale Out the financial application from the UCY private cloud infrastructure to a public cloud provider infrastructure or to a hybrid cloud infrastructure.
Prerequisites (Dependencies) & Assumptions	- Scale Up is not possible on the UCY private cloud due to insufficient resources, so scale out to a new public cloud that provides the required resources. - Scale Out from the UCY private cloud to a public cloud provider to reduce costs. - Scale Out from the UCY private cloud to a hybrid cloud to reduce latency and preserve data confidentiality.
Steps	1. The financial application is deployed using PaaSage. 2. A Quality Attribute is violated, which is detected by the PaaSage platform. 3. The PaaSage platform applies the scalability rule

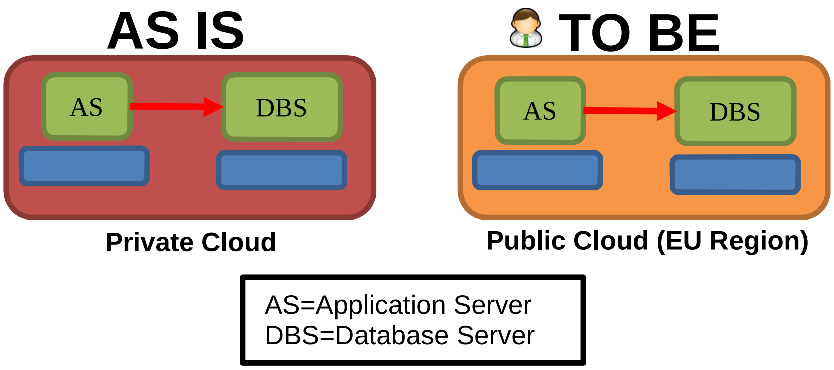
	associated with the violated metric. 4. The PaaSage platform performs a scale out operation. 5. The PaaSage platform keeps monitoring the relevant Quality Attribute.
Variations (optional)	
Quality Attributes	• Basic Private to Public Cloud Un-deploy/Redeploy to reduce *operational costs* • Scale out due to *insufficient resources* (i.e., CPU and RAM) • Scale out due to *network latency*
Issues	

Quality Attribute Scenario 1:

Move private to public permanently: A financial firm decides to change their technological strategy in order to cut-down costs by moving the financial application and database server from the in-house private datacentre to the public cloud.

Scenario Id	**PTP-CO-1 (Private to Public – Cost – 1)**
Scenario Name	**Basic private to public cloud un-deploy/redeploy to reduce operational costs**
Scenario Type	Cost reduction
Artefact	Application server and database server
Context	There is a need to reduce operational costs by moving away from the private datacentre to the public cloud.
Stimulus	**Static:** Avoiding requirement to purchase new servers and upgrade the company's software. Moving from in-house datacentre to the public cloud is a more economical solution.
Response	The system un-deploys the financial service from the in-house private datacentre and redeploys to the selected public cloud provider based on the company's preferences.
Response Measure	The costs are reduced due to the migration to the public cloud.

Graphical Representation Scenario 1:



<u>Quality Attribute Scenario 2:</u>

Move private to public temporarily: A company due to temporary workload needs to move application and database server to the cloud (i.e., scale out) in order to get more resources.

Scenario Id	**PTP-LO-1 (Private to Public – Load – 1)**
Scenario Name	**Scale out due to insufficient resources (i.e., CPU and RAM)**
Scenario Type	Requirement for additional resources
Artefact	Application server and database server
Context	The workload is higher from a configurable limit (e.g., load of CPU and RAM is more than 70%).
Stimulus	**Dynamic:** Load monitoring reports an insufficient resources problem over the last minutes.
Response	The system moves the financial service to a cloud environment with more resources.
Response Measure	The resources usage reported by the load monitoring component drops below a specified limit.

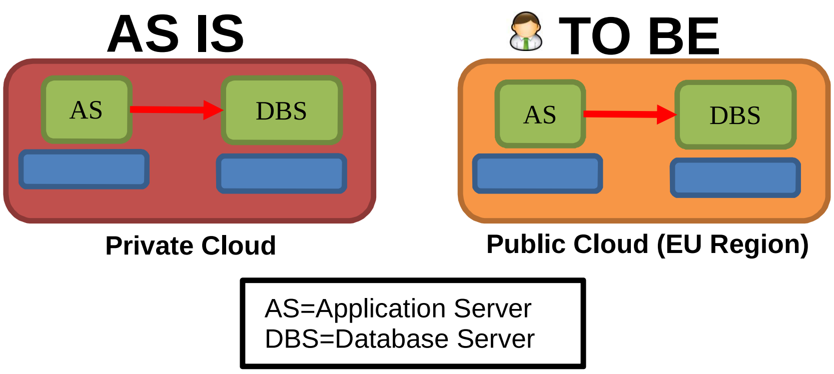
Example: Companies in Cyprus pay once a year a corporate levy. Organizations using the financial software require more resources 2-3 weeks before the deadline. In this case, their private cloud cannot support the additional load during these 2-3 weeks which leads to delays of payments and penalty on late payments (additional costs for their clients and for their payroll).

Example procedure:

1. 50 users are working on the software throughout the year.
2. During the corporate levy period, the number of users that need to work to meet deadlines rises to 250.
3. System monitors the load of the in-house server.

4. If the load of CPU and RAM is more than 70%, an alert will be send to the administrator that migrations will start during the night.
5. Migrations start during the night.
6. An alert will be send to the admin on completion.
7. Admin needs to test that everything is OK.

Graphical Representation Scenario 2:

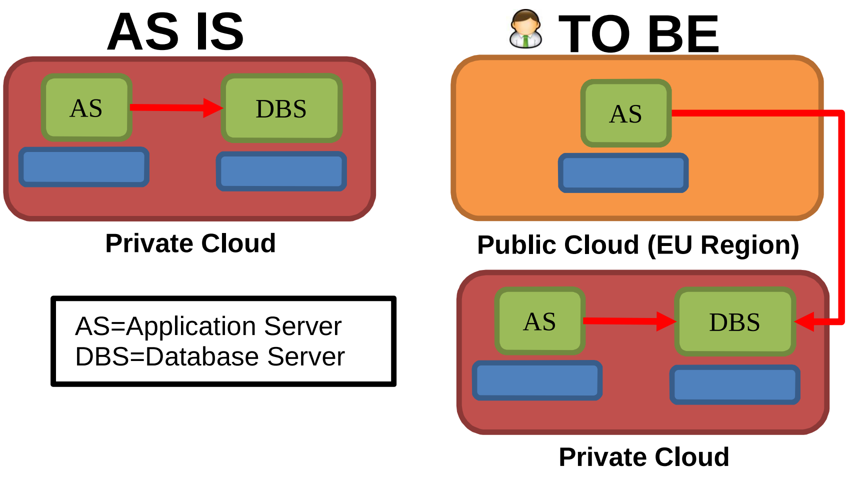


Quality Attribute Scenario 3:

Create a copy of the application server temporarily to a public cloud keeping database server private to preserve confidentiality: A company has several employees (e.g. 10 employees) travelling to an international financial company branch overseas, while they need to use the financial application to perform their work. Due to the use of the application from overseas, network latency is experienced. Network latency is monitored which triggers and configures the VM instance while a copy of the application server is deployed on the public cloud. For data confidentiality reasons the database server remains in the private cloud. Note that the existing application server remains in the private cloud for the Cyprus company branch needs.

Scenario Id	**SO-LAT-1 (Scale Out – Latency – 1)**
Scenario Name	**Scale out due to network latency**
Scenario Type	Latency reduction
Artefact	Application server and database server
Context	The (network) latency time is higher than a configurable limit since a specified amount of time.
Stimulus	Network monitoring signals significant latency problem over the last minutes.
Response	The system moves the service onto a cloud environment with lower latency. This might be a different provider or a cloud provider nearby the user of the service.
Response Measure	The average latency reported by the network monitoring facility drops below a specified limit.

Graphical Representation Scenario 3:



5.3 *Scenario Grouping*

This section groups several scenarios together for easier reference in the following sections.

Scenario Group Id	Scenario Ids	Description
PTPG-CO-1	PTP-CO-1	Moving to public cloud scenarios triggered by cost reduction.
PTPG-LO-1	PTP-LO-1	Moving to public cloud scenarios triggered by insufficient resources.
SOG-LAT-1	SO-LAT-1	Scale out scenarios triggered by latency related stimuli.

5.4 *Traceability with respect to PaaSage components*

The following table describes how each scenario group maps to the PaaSage components.

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
PTP-CO-1	The criteria about cost parameters and objectives are defined in	Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy	Executionware measures usage and license costs of running applications.	Storing the data about usage and license costs from different providers.

	Saloon and WS-Agreement.	the SLA. Reasoner uses the monitoring data to choose a provider that satisfies the cost parameters and objectives. Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Components are moved to the other suitable provider.	
PTP-LO-1	Resources specification (i.e., CPU, RAM) for computing related components. Defined in WS-Agreement.	Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. Reasoner uses the monitoring data to choose a provider that satisfies the load requirements. Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware measures the load percentages of computing-related components. Components are moved to the other suitable provider.	Storing the data about resources load and the locations serviced from different providers.
SO-LAT-1	Latency specification for network related components. Defined in WS-Agreement.	Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. Reasoner uses the monitoring data to choose a provider with the low network latency in the last few minutes. Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware measures the latency of network-related components. Components are moved to the other suitable provider.	Storing the data about network latency and the locations serviced from different providers.

5.5 Traceability to the integration tests

The following table maps several Scenario groups into one integration test group. The aim of this chapter is to describe the connection of the scenario group(s) and the different integration test scenarios.

Integration test scenario group	Scenario group Id	Description
ITG-1	PTPG-CO-1, PTPG-LO-1	Integration tests which focus on (static) cost related and on (dynamic) resources full un-deploy and redeploy scenarios.
ITG-2	SOG-LAT-1	Integration tests which focus on (dynamic) latency related scale-out/ scale-in scenarios.

6 eScience sector - complex scientific applications

This case is supported by the AGH partner.

6.1 Objectives (revised)

AGH University of Science and Technology is one of the Polish technical universities. The **Department of Computer Science** employs teaching and research staff of over 80 people, devoting their research efforts to various IT directions, including scalable distributed systems, cross-domain computations in loosely coupled environments, knowledge management and support for life sciences.

Within the scope of research projects, AGH collaborates closely with researchers and application users from the eScience domain, both local and international. The interesting use cases for PaaSage are those that require either large-scale workflow or data farming processing. AGH is either involved directly in supporting these applications on grids and clouds or develops tools that enable and facilitate execution of them on these infrastructures.

Local eScience applications and tools are related mostly to the PL-Grid project users and include:

- Bioinformatics applications, in collaboration with the Jagiellonian University Medical College. They include genetic data analysis (sequence alignment, similarity search) as well as proteomic experiments: protein folding and structural comparison. The infrastructures used for these experiments are clusters, grids and clouds [1][2].
- Investigating potential benefits of data farming application to study complex metallurgical processes including generation of Statistically Similar Representative Volume Element and Digital Material Representation. This research is conducted by Faculty of Metals Engineering and Industrial Computer Science AGH [7][8].

International collaborations in eScience domain include:

- Virtual Physiological Human initiative, where the scientific workflows are deployed on the cloud in the scope of VPH-Share project [3]. The workflows mainly use the Taverna [16] engine for orchestrating the Atomic Services and a specific plugin for Taverna is developed to dynamically create service instances on the cloud using the Atmosphere [14] cloud platform developed by AGH. Other large-scale workflows that are under development use DataFluo workflow engine [15] developed by University of Amsterdam.
- Multiscale applications from fusion domain developed using workflow tools and MAPPER framework [4]. The MAPPER project provides tools for running multiscale applications on distributed computing infrastructures. The application from the fusion domain used Kepler [17] workflow system to orchestrate its tasks.
- Collaboration with Pegasus team from University of Southern California for support of scientific workflows on cloud infrastructures [5][6]. This collaboration resulted in algorithms for scheduling and provisioning for workflow ensembles on clouds and cost optimization of applications on cloud infrastructures [18]. One of the important benefits of this collaboration is the

experience with scientific workflows that use Pegasus [19] workflow management system and the workflow gallery that contains real and synthetic workflows [20].

- Mission planning support in military applications with data farming within the EDA EUSAS project. In the scope of the project, a novel approach to military training was developed, based on behaviour modelling and multi-agent simulations. At first, soldiers' behaviour was captured during a series of training sessions and transformed into a set of rules, which was then used during highly realistic agent-based simulations of military missions [9]. The aim of the data farming in the process was to develop a better understanding of soldiers' behaviour and identify potential vulnerabilities. During data farming experiments, numerous agent-based simulations were executed, each with different environmental conditions, e.g. emotional state of civilians involved in a mission. Data generated during the simulations was collected and analysed to find cases when the selected strategy was wrong, e.g. there were too many casualties. The underlying infrastructure for executing the simulations included private clusters, grids and clouds [10][11].

6.1.1 Selection of the use case scenario

The two main tools that are developed by AGH to support these applications are:

- HyperFlow workflow execution engine that is based on hypermedia paradigm and supports flexible processing models such as data flow, control flow, and includes the support for large-scale scientific workflows which can be described as directed acyclic graphs of tasks [13].
- Scalarm is a massively self-scalable platform for data farming, which supports phases of data farming experiments, starting from parameter space generation, through simulation execution on heterogeneous computational infrastructure, to data collection and exploration [12].

While these eScience applications and supporting tools are in various stages of development and maturity, none of them uses the model based approach for development and deployment on clouds that is proposed within PaaSage. Therefore, all of them can benefit from the PaaSage platform.

6.1.2 Overview over the HyperFlow prototype

In HyperFlow, a workflow is simply a set of processes connected through ports and exchanging signals. The basic abstraction for workflows, a process, is defined by:

- Input ports and associated signals which arrive at the process.
- Output ports and associated signals which are emitted by the process.
- Function invoked from the process which transforms input signals to output signals.
- Type of the process which determines its general behaviour. For example, a *dataflow* process waits for *all* data inputs, invokes the function, and emits all data outputs. A *parallel-foreach* process, in turn, waits for *any* data input, invokes the function, and emits the respective data output.

The prototype of a cloud-based workflow execution in HyperFlow is implemented as shown in Figure 6.1:. The user needs to provide the workflow description using the

DSL, which is a DAG in JSON format. This includes all tasks with the dependencies, as well as input and output files in the form of URLs. The whole workflow application consists of:

- Master which includes a Workflow Engine (HyperFlow) together with Redis Database and RabbitMQ server for communication,
- Shared storage (e.g. NFS server) that requires a separate VM for data exchange between workers,
- Worker that includes a part of executor (a generic component managing task execution) and application-specific binaries (e.g. for Montage application). Worker may be executed on multiple VMs, i.e. scaled out (horizontally) for parallel execution.

  For PaaSage, two scale-out variants are planned:
- Dynamic auto-scaling, where the platform uses dynamic information about the application and infrastructure, such as resource utilization, queue length or virtual machine load.
- Auto-scaling based on the scheduling plan, where the HyperFlow planner prepares a task scheduling plan and VM provisioning plan and these plans are used to trigger auto-scaling decisions at runtime. This variant can be useful when e.g. the workflow consists of several stages and the estimates of resource requirements are known in advance.

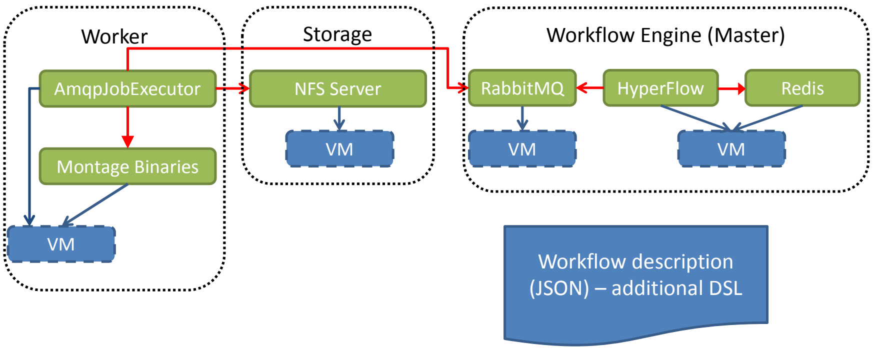


Figure 6.1: HyperFlow application deployed on the cloud

6.1.3 Motivation for the Cloud in HyperFlow

Cloud infrastructure is a natural solution for high throughput computing (HTC) workflows. The function of a process within the workflow sends a job specification to a remote message queue, while local Executors residing on Virtual Machines deployed in a cloud fetch the jobs from the queue, invoke the appropriate application components, and send the results back to the queue. Since multiple VMs and execution engines could be deployed and connected to the same queue, it acts not only as a communication medium, but also as a load balancing mechanism.

Such loosely-coupled architecture provides not only the distributed computing capability, but also allows taking advantage of scaling out the application worker components in response to the changes in the infrastructure or due to application specific events. These events can result e.g. from the application reaching a specific stage of the workflow, which has different resource demands than the previous stage. Thanks to the dynamic architecture of the cloud and the capabilities of the PaaSage platform, it will be thus possible to adjust the number of computing resources to the needs of the application and achieve such high-level objectives as completing the workflow within a given deadline or minimization of the resource cost.

6.2 Scenario Description for HyperFlow

The scenarios of HyperFlow usage are related to horizontal scaling of worker VMs. Since scale-out and scale-in operations are symmetric, we group them together (scale-out/in – SOI) for the sake of clarity of presentation.

Use Case	Deployment and scale out/in of HyperFlow application
Description	The HyperFlow engine and worker VMs form a virtual cluster that can be dynamically scaled out and in, in accordance with the demand.
Prerequisites (Dependencies) & Assumptions	The scaling out makes sense for the workflows that are large enough to benefit from parallel execution. This means that the number and size of tasks that are executed in parallel outweighs the costs and overheads of VM start-up.
Steps	1. The HyperFlow application is deployed using PaaSage 2. A KPI relevant to HyperFlow execution is violated 3. The PaaSage platform applies the scalability rule associated with the violated metric. 4. The PaaSage platform Executionware performs a scale out/in operation. 5. The PaaSage platform monitors the relevant KPI
Variations (optional)	An important variation of the horizontal scaling is the cooperation of the PaaSage platform with the workflow scheduler of HyperFlow. In this case, the workflow planner in cooperation with the Reasoner prepares a plan that finds the optimal number of VMs for each of the stages of the workflow. The plan has the form of scalability rules triggered by application-specific metrics (such as workflow stage) that trigger the appropriate scaling actions of PaaSage Executionware.
Quality Attributes	• Utilization of the virtual cluster • Average number of jobs in the queue. • Workflow stage
Issues	The issues will be the cooperation of HyperFlow workflow scheduler with the PaaSage platform in order to prepare and enforce the scheduling plan.

Scenario Id	**HF-DEPL-1**
Scenario Name	**Deployment of HyperFlow application on cloud infrastructure**
Scenario Type	Deployment
Artefact	HyperFlow Engine, RabbitMQ, Worker node (executor +

	application binaries)
Context	The initial deployment of all the components needed to execute scientific workflows need to be deployed on demand when researchers need them.
Stimulus	Request from the end-user of infrastructure administrator.
Response	The system spawns the whole cluster in its initial configuration.
Response Measure	The HyperFlow engine is available for accepting workflows submitted by the users.

Scenario Id	**HF-SOI-UTL-1**
Scenario Name	**Scale out/in due to resource utilization**
Scenario Type	Scalability
Artefact	Worker node (executor + application binaries)
Context	The resource utilization of the cluster of worker VM is out of desired range within a set time window, i.e. lower than 30% or higher than 80%.
Stimulus	Resource utilization monitoring reports the utilization of the cluster is out of the desired range.
Response	The system spawns/terminates VMs, respectively. The newly deployed VMs are of the same flavour, to avoid heterogeneity of the cluster.
Response Measure	The average utilization of the cluster reported by the monitoring service returns to the desired range.

Scenario Id	**HF-SOI-JBQ-1**
Scenario Name	**Scale out/in due to increase/decrease of the number of jobs in the queue**
Scenario Type	Scalability
Artefact	Worker node (executor + application binaries)
Context	The number of ready tasks in the AMQP queue is increasing/decreasing within a set time window.
Stimulus	The monitoring sensor deployed within the AMQP broker reports the increase/decrease of the number of tasks in the queue.
Response	The system spawns/terminates VMs, respectively. The newly deployed VMs are of the same flavour, to avoid heterogeneity of the cluster.
Response Measure	The average number of jobs in the queue reported by the monitoring service returns to the steady state, i.e. it remains

	constant within a specified range.

Scenario Id	**HF-SOI-STG-1**
Scenario Name	**Scale out/in due to reaching the specific stage by the workflow**
Scenario Type	Scalability
Artefact	Worker node (executor + application binaries)
Context	The workflow stage (the application-specific metric reported by the HyperFlow engine to the Executionware) changes to a new value, e.g. the workflow completes stage 2 and begins stage 3.
Stimulus	The monitoring sensor deployed within HyperFlow reports the new value of the workflow stage metric.
Response	The system spawns/terminates VMs, respectively. The newly deployed VMs are of the same flavour, to avoid heterogeneity of the cluster.
Response Measure	The number of VMs reaches the value desired for the current stage, as defined in the workflow scheduling plan.

Scenario Id	**HF-SOI-TERM-1**
Scenario Name	**Terminate the cluster of worker VMs on workflow completion**
Scenario Type	Scalability
Artefact	Worker node (executor + application binaries)
Context	The workflow engine reports to the monitoring system that the execution of workflow is complete.
Stimulus	The monitoring sensor deployed within HyperFlow reports that the “running” flag is set to 0 (false).
Response	The system terminates all the worker VM
Response Measure	The number of worker VMs is 0.

6.3 *Scenario Grouping for HyperFlow*

This section groups several scenarios together for easier reference.

Scenario Group Id	Scenario Ids	Description
HF-DYN	HF-DEPL-1, HF-SOI-UTL-1, HF-SOI-JBQ-1	HyperFlow scalability scenarios requiring only dynamic information on the workflow execution (resource utilization,

		number of jobs in the queue)
HF-PLAN	HF-DEPL-1, HF-SOI-STG-1, HF-SOI-TERM-1	HyperFlow scalability scenarios requiring the scalability rules generated in advance by the workflow planner

6.4 Traceability with respect to PaaSage components for HyperFlow

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
HF-DYN	Resource utilization and job queue length specification for HyperFlow applications, defined in Saloon.	The Upperware connected to MDDB finds the best deployment and configuration of HyperFlow application. The scalability rules for utilization/queue length are passed to the Executionware.	Executionware deploys the HyperFlow cluster. At runtime it measures the desired metrics and reacts when they are out of range. The scale-out/in actions are triggered and the cluster reaches the desired state.	Data about resource utilization, queue length are stored in the time series database of MDDB and available for querying and historical analysis.
HF-PLAN	Workflow stage and termination flag metrics specification for HyperFlow applications, defined in Saloon.	The workflow planner within HyperFlow cooperates with the Reasoner of PaaSage to prepare a scheduling plan for workflow consisting of multiple stages. Upperware connected to MDDB finds the best deployment and configuration of HyperFlow application. The scalability rules for workflow stages and termination are passed to the Executionware.	Executionware deploys the HyperFlow cluster. At runtime it measures the desired metrics and reacts when they are triggered. The scale-out/in actions are triggered and the cluster reaches the desired state.	Data about workflow stages and termination times are stored in the time series database of MDDB and available for querying and historical analysis.

6.5 Traceability to the integration tests for HyperFlow

The following table maps several Scenario groups into one integration test group. The aim of this chapter is to describe the connection of the scenario groups(s) and the different integration test scenarios.

Integration test scenario group	Scenario group Id	Description
ITG-HF-1	HF-DYN	Integration tests which focus on scaling in/out HyperFlow workers based on dynamic information.
ITG-HF-2	HF-PLAN	Integration tests which focus on scaling in/out HyperFlow workers based on information from the workflow planner. Cooperation between workflow planner and PaaSage Reasoner is required.

6.5.1 Overview over the Scalarm prototype

Scalarm is a massively self-scalable platform for data farming, which supports phases of data farming experiments, starting from parameter space generation, through simulation execution on heterogeneous computational infrastructure, to data collection and exploration.

Massive self-scalability is the main non-functional requirement which has to be supported by Scalarm in order to conduct data farming experiments at a large-scale efficiently. Activities performed in different phases of a data farming experiment impose that the used software and infrastructure is elastic and can be scaled automatically on demand, e.g.:

- during "Input space specification" multiple time consuming design of experiment (DoE) methods can be executed to explore possibilities of input space size reduction,
- "Simulation execution" usually requires numerous simulation to be executed in parallel in a HTC manner,
- "Output data exploration" often involves executing computationally intensive data mining methods on large data sets to extract knowledge from simulations output.

Scalarm architecture is depicted in Figure 6.2. Scalarm consists of loosely coupled services responsible for managing experiments, storage, and simulations. Also, there is a dedicated service, called Information Service, which implements the Service Locator pattern. Besides Information Service, each Scalarm service can be instantiated multiple times to scale out in order to attain massive scalability. The Scalarm architecture follows the master-worker design pattern, where the master part includes Experiment Manager, Storage Manager and Information Service, while the worker part includes Simulation Manager. Each instance of a service can be run on a separate infrastructure to provide fault tolerance and increase overall performance by exploiting services locality.

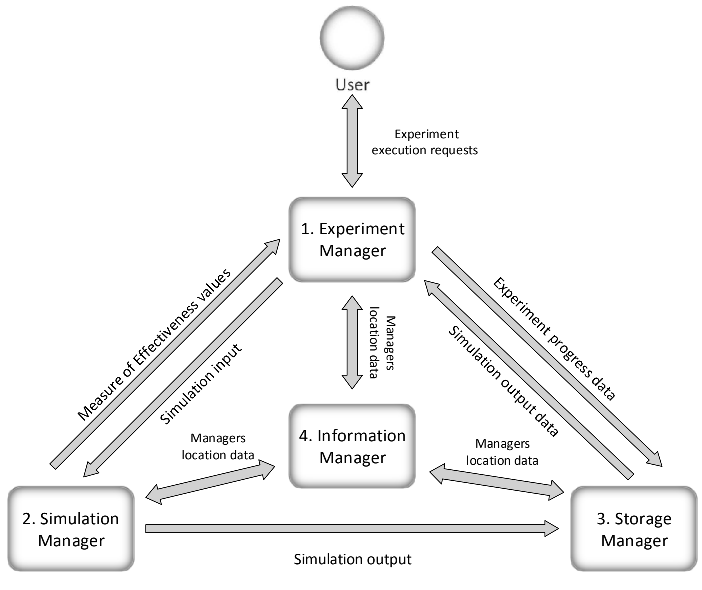


Figure 6.2: Overview of the Scalarm architecture.

6.5.2 Motivation for the Cloud in Scalarm

The basic motivation for conducting data farming experiments with Scalarm on clouds is the requirement for an elastic access to potentially huge amount of computational resources. Commonly, a scientist has access to a small infrastructure locally, e.g. an institutional cluster exposed as a private cloud, which is sufficient for small-scale data farming experiments. However, to conduct large-scale experiments, the scientist needs to use another, larger pool of resources. More and more often, a public cloud, e.g. Amazon Elastic Compute Cloud (EC2), is used as another virtually unlimited pool of resources. Large-scale experiments typically involve thousands of simulation runs to execute. Conducting such experiments requires more resources for running more instances of Simulation Managers, which generates more workload on the master part of Scalarm. Hence, it is desirable to use either a single cloud or a combination of resources from multiple clouds (both private and public) as a single, elastic environment in such experiments. In addition, as the offer of available public clouds is still expanding, the software coordinating data farming experiments should be capable to embrace different clouds with minimal modifications required.

6.6 Scenario Description for Scalarm

Scaling scenarios presented in this section describe when, why and how each Scalarm service should be scaled out and in. They correspond to use cases covering scaling scenarios of each Scalarm service. As each service can be scaled out and in based on the actual workload, we identified four "scaling out and in" scenario pairs.

The first described service is Experiment Manager. It is mainly a CPU-bound application, which exposes a Graphical User Interface and a REST API. Hence, its scalability behaviour is related to the response time performance metric.

Use Case	Experiment Manager scaling
Description	Scale out or in of the Experiment Manager service
Prerequisites (Dependencies) & Assumptions	The Experiment Manager service is run in multiple instances on different servers connected with a private network
Steps	1. PaaSage monitoring collects workload information about running Experiment Manager instances 2. Specified performance KPI is violated and an appropriate scaling procedure is triggered 3. The PaaSage platform decides whether to start a new instance or stop an already running one. 4. The PaaSage platform performs an appropriate scaling action 5. The PaaSage platform notifies Scalarm Information Service where the new instances is started
Variations	Experiment Manager can be scaled out or in depending on the

(optional)	actual workload
Quality Attributes	• Experiment Manager response time • Workload of a server hosting an Experiment Manager instance
Issues	-

Scenario Id	**SCAL-1-EM-SO**
Scenario Name	**Experiment Manager scale out due to poor response time**
Scenario Type	Performance increase
Artefact	Experiment Manager
Context	The service response time is higher than the configured value on average within a specified time window.
Stimulus	Monitoring signals that the service response time is higher than expected over the specified time window.
Response	The system starts a new instance of Experiment Manager in a cloud close to other Scalarm services.
Response Measure	The measured response time of Experiment Manager drops below the specified threshold.

Scenario Id	**SCAL-2-EM-SI**
Scenario Name	**Experiment Manager scale in due to low workload**
Scenario Type	Utilized resources reduction
Artefact	Experiment Manager
Context	The service response time is higher than the configured value on average within a specified time window.
Stimulus	Monitoring signals very low CPU utilization of resources utilized by Experiment Manager instances.
Response	The system stops the Experiment Manager instances with the least CPU utilization level.
Response Measure	The amount of utilized resources is reduced, without significant performance loss.

The second scalable Scalarm service is the Storage Manager. It manages a sharded cluster of a NoSQL database and provides a dedicated REST API to store results of the executed simulation runs. Two metrics are crucial for efficient execution of this service: IO utilization and hard disk space.

Use Case	Storage Manager scaling
Description	Scale out or in of the Storage Manager service
Prerequisites (Dependencies) & Assumptions	The Storage Manager service is run in multiple instances on different servers connected with a private network
Steps	1. PaaSage monitoring collects workload information about running Storage Manager instances 2. Specified performance KPI is violated and an appropriate scaling procedure is triggered 3. The PaaSage platform decides whether to start a new instance or stop an already running one. 4. The PaaSage platform performs an appropriate scaling action 5. The PaaSage platform notifies Scalarm Information Service where the new instances is started
Variations (optional)	Storage Manager can be scaled out or in depending on the actual workload
Quality Attributes	• Number of IO requests performed by Storage Manager • Disk space available on a server hosting Storage Manager • Workload of a server hosting a Storage Manager instance
Issues	-

Scenario Id	**SCAL-3-SM-SO**
Scenario Name	**Storage Manager scale out due to high IO utilization**
Scenario Type	Performance increase
Artefact	Storage Manager
Context	The service IO utilization is higher than the configured value on average within a specified time window.
Stimulus	Monitoring signals more than expected IO requests on resources utilized by Storage Manager instances.
Response	The system starts a new instance of Storage Manager in a cloud close to other Scalarm services.
Response Measure	The average IO request count on resources used by Storage Manager instances drops below the configured threshold.

Scenario Id	**SCAL-4-SM-SI**

Scenario Name	**Storage Manager scale in due to low workload**
Scenario Type	Utilized resources reduction
Artefact	Storage Manager
Context	The average utilization level of resources used by Storage Manager instances is below the configured threshold within a specified time window.
Stimulus	Monitoring signals very low number of IO requests on resources utilized by Storage Manager instances.
Response	The system stops the least utilized instance of Storage Manager.
Response Measure	The amount of utilized resources is reduced, without significant performance loss.

Scenario Id	**SCAL-5-SM-SO-2**
Scenario Name	**Storage Manager scale out due to insufficient disk space**
Scenario Type	Performance increase
Artefact	Storage Manager
Context	The storage capacity requirement exceeds the available disk space on resources used be Storage Manager instances.
Stimulus	Monitoring signals low available disk space.
Response	The system starts a new instance of Storage Manager in a cloud close to other Scalarm services.
Response Measure	The average available disk space on resources used by Storage Manager instances increases above the configured minimum.

Scenario Id	**SCAL-6-SM-SI-2**
Scenario Name	**Storage Manager scale in due to low workload**
Scenario Type	Utilized resources reduction
Artefact	Storage Manager
Context	The average utilization level of disk space on resources used by Storage Manager instances is below the configured threshold.
Stimulus	Monitoring signals very low disk space utilization on resources utilized by Storage Manager instances.
Response	The system stops the least utilized instance of Storage Manager.
Response Measure	The amount of utilized resources is reduced, without significant performance loss.

The last Scalarm service, which should be scaled, is the Simulation Manager. It constitutes the worker part of the platform; hence the number of its instances influences the progress rate of data farming experiments conducting with Scalarm. On the other hand, running too many Simulation Manager Instances may lead to undesired costs.

Use Case	Simulation Manager scaling
Description	Adding more instances of Simulation Manager or stopping the running ones
Prerequisites (Dependencies) & Assumptions	The Simulation Manager service is run in multiple instances on different servers connected with a connection to other Scalarm Services
Steps	1. PaaSage monitoring collects workload information about running Simulation Manager instances 2. Specified performance KPI is violated and an appropriate scaling procedure is triggered 3. The PaaSage platform decides whether to start a new instance or stop an already running one. 4. The PaaSage platform performs an appropriate scaling action
Variations (optional)	Simulation Manager can be scaled out or in depending on the progress of conducting data farming experiments
Quality Attributes	• Conducted data farming experiments' progress rate • Cost of running computations
Issues	-

Scenario Id	**SCAL-7-SiM-SO**
Scenario Name	**Simulation Manager scale out due to insufficient disk space**
Scenario Type	Increase data farming experiment progress rate
Artefact	Simulation Manager
Context	The progress rate of a conducted data experiment is lower than expected and the risk of not finishing the experiment in time is too high.
Stimulus	Monitoring signals there is too few simulations finished in a specified time window.
Response	The system starts a new instance of Simulation Manager in the cheapest cloud, which fulfils Simulation Manager requirements.

Response Measure	The experiment's progress rate increases above the expected value.

Scenario Id	**SCAL-8-SiM-SI**
Scenario Name	**Simulation Manager scale in due to too high cost**
Scenario Type	Execution cost reduction
Artefact	Simulation Manager
Context	The number of Simulation Manager instances is too costly, while the progress rate of the running experiment is above the expected level.
Stimulus	Monitoring signals too high cost of running Simulation Manager instances.
Response	The system stops a random instance of Simulation Manager.
Response Measure	The amount of utilized resources is reduced, without significant loose in experiment progress rate.

6.7 *Scenario Grouping for Scalarm*

This section groups several scenarios together for easier reference.

Scenario Group Id	Scenario Ids	Description
SCAL-SO	SCAL-1-EM-SO, SCAL-3-SM-SO, SCAL-4-SM-SO-2, SCAL-7-SiM-SO	Scalarm scaling out scenarios
SCAL-IN	SCAL-2-EM-SI, SCAL-4-SM-SI, SCAL-6-SM-SI-2, SCAL-8-SiM-SI	Scalarm scaling in scenarios

6.8 *Traceability with respect to PaaSage components for Scalarm*

The following table describes how each Scalarm scaling scenario group maps to the PaaSage components.

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
SCAL-OUT	The scaling out criteria about workload,	PaaSage Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy	PaaSage Executionware executes the commands prepared	Storing the data about current workload of Scalarm

	experiment's progress rate, available disk space, and computation's cost are defined in Saloon.	deployment requirements of Scalarm services. PaaSage Reasoner uses the list of providers selected by Profiler, scaling rules specification, and monitoring data to generate a list of possible deployment configurations of new instances of Scalarm services. PaaSage Adapter generates commands for new Scalarm service instances deployment and monitoring.	by PaaSage Adapter on the cloud provider selected by PaaSage Reasoner. Scalarm service instances are instantiated and notifications are triggered.	services and hosting servers.
SCAL-IN	The scaling in criteria about workload, experiment's progress rate, available disk space and computation's cost are defined in Saloon.	PaaSage Reasoner uses the monitoring data and current Scalarm deployment model to select the least utilized Scalarm service instance. PaaSage Adapter generates commands for stopping the selected Scalarm service instance.	PaaSage Executionware executes the commands prepared by PaaSage Adapter against the selected Scalarm service instance. The selected Scalarm service instance is stopped.	-

6.9 Traceability to the integration tests for Scalarm

The following table maps several Scenario groups into one integration test group. The aim of this chapter is to describe the connection of the scenario group(s) and the different integration test scenarios.

Integration test scenario group	Scenario group Id	Description
ITG-SCAL-1	SCAL-OUT	Integration tests which focus on scaling out different Scalarm services
ITG-SCAL-2	SCAL-IN	Integration tests which focus on scaling in different Scalarm services

6.10 References

[1] M. Malawski, J. Meizner, M. Bubak, and P. Gepner. Component approach to computational applications on clouds. *Procedia Computer Science*, 4:432-441, May 2011.

[2] T. Jadczyk, M. Malawski, M. Bubak, and I. Roterman, "Examining protein folding process simulation and searching for common structure motifs in a protein family as experiments in the Gridspace2 virtual laboratory," in *PL-Grid*, Springer-Verlag, 2012, pp. 252–264.

[3] M. Bubak, M. Kasztelnik, M. Malawski, J. Meizner, P. Nowakowski, and S. Varma, "Evaluation of Cloud Providers for VPH Applications," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, pp. 200–201.

[4] M. Ben Belgacem, B. Chopard, J. Borgdorff, M. Mamoński, K. Rycerz, and D. Harezlak, "Distributed Multiscale Computations Using the MAPPER Framework," *Procedia Comput. Sci.*, vol. 18, pp. 1106–1115, 2013.

[5] E. Deelman, G. Juve, M. Malawski, and J. Nabrzyski, "Hosted Science: Managing Computational Workflows in the Cloud," *Parallel Process. Lett.*, (accepted), 2013.

[6] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Cost- and Deadline-Constrained Provisioning for Scientific Workflow Ensembles in IaaS Clouds," in *SC '12 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2012.

[7] D. Król, L. Dutka, J. Kitowski, "Towards efficient virtual experiments in metallurgy with data farming on heterogeneous computational infrastructure", in Cracow'13 Grid Workshop: November 2013, Krakow, Poland, pp. XX–XX, 2013.

[8] M. Ambrozinski, K. Bzowski, L. Rauch, M. Pietrzyk, Application of statistically similar representative volume element in numerical simulations of crash box stamping, Archives of Civil and Mechanical Engineering / Polish Academy of Sciences. Wrocław Branch, Wrocław University of Technology, 2012, vol. 12(2), pp. 126–132.

[9] Hluchy L., Kvassay M., Dlugolinsky S., Schneider B., Bracker H., Kryza B., Kitowski J.: Handling internal complexity in highly realistic agent-based models of human behaviour. In Proceedings of 6th IEEE International Symposium on Ap- plied Computational Intelligence and Informatics (SACI 2011), pp. 11–16, 2011.

[10] B. Kryza, D. Krol, M. Wrzeszcz, L. Dutka, and J. Kitowski, "Interactive cloud data farming environment for military mission planning support," Computer Science Journal AGH, vol. 13, no. 3, pp. 89–100, 2012.

[11] D. Krol, B.Kryza, M. Wrzeszcz, L. Dutka, and J.Kitowski, "Elastic Infrastructure for Interactive Data Farming Experiments," Procedia Computer Science, vol. 9, no. 0, pp. 206 – 215, 2012. Proceedings of the International Conference on Computational Science, ICCS 2012.

[12] D. Krol, M. Wrzeszcz, B. Kryza, L. Dutka, and J. Kitowski, “Massively Scalable Platform for Data Farming Supporting Heterogeneous Infrastructure,” in The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, IARIA Cloud Computing 2013, Valencia, Spain, pp. 144–149, 2013.

[13] B. Balis, B. Hypermedia workflow: a new approach to data-driven scientific workflows. In High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion, pp. 100-107, IEEE 2012.

[14] P. Nowakowski, T. Bartynski, T. Gubala, D. Harezlak, M. Kasztelnik, M. Malawski, J. Meizner, and M. Bubak, “Cloud Platform for Medical Applications,” in *eScience 2012*, 2012.

[15] R. Cushing, S. Koulouzis, A. Belloum, and M. Bubak, “Dynamic Handling for Cooperating Scientific Web Services,” in *2011 IEEE Seventh International Conference on eScience*, 2011, pp. 232–239.

[16] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, “Taverna: lessons in creating a workflow environment for the life sciences,” *Concurr. Comput. Pract. Exp.*, vol. 18, no. 10, pp. 1067–1100, 2006.

[17] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, “Scientific workflow management and the Kepler system,” *Concurr. Comput. Pract. Exp.*, vol. 18, no. 10, pp. 1039–1065, 2006.

[18] M. Malawski, K. Figiela, and J. Nabrzyski, “Cost minimization for computational applications on hybrid cloud infrastructures,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1786–1794, Jan. 2013.

[19] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, B. Berriman, J. Good, A. Laity, J. Jacob, and D. Katz, “Pegasus: A framework for mapping complex scientific workflows onto distributed systems,” *Sci. Program.*, vol. 13, no. 3, pp. 219–237, 2005.

[20] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, “Characterizing and profiling scientific workflows,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 3, pp. 682–692, 2013.

7 eScience sector – resource intensive simulations

This case is supported by the UULM / ASCS partner.

7.1 Objectives (revised)

The High Performance Computing Centre (HLRS) is a research and service institution affiliated to the University of Stuttgart, and offering HPC resources to academic users and industry. HLRS also provides consultancy services, and training for industry and academia to program large-scale systems and converts existing applications or algorithms into large-scale use cases for performing scientific experiments.

Collaborative research with automotive industry is done together with the Automotive Simulation Centre Stuttgart (ASCS), where ASCS fosters application-oriented research in the field of automotive engineering by the use of information and communication technologies. It also promotes and accelerates the transfer of the latest results of scientific research on numerical simulation. The goal of the ASCS is to provide industry with HPC simulation methods which satisfy high scientific standards and also fulfil ambitious industrial demands.

Scientific computing requires an ever-increasing number of heterogeneous resources to deliver results for growing problem sizes in a reasonable timeframe. With the current business procedure of HPC centre, it is quite difficult for users to configure and manage the execution of their resource-intensive applications. Many characteristics need to be defined or estimated in advance, such as how many cores are actually needed, how much memory is required for computation, how should the machines be configured, expected execution time, etc. There is no general strategy to assess the configuration, as it depends on the specific requirements of the application and its input/output data. Overestimating the aforementioned parameters will occupy unnecessary computing resources, thus, leading to unnecessary cost; whereas underestimation will lead to unnecessary delays and even loss of results. The second problem is that (a) if a user wants to rent dedicated resources, a large number of machines need to be reserved in order to reduce the overall execution time. This would require that at any time a certain number of machines are available for usage - which means the machines have to be reserved in advance and the number of nodes is fixed. This is not only costly, but also very inflexible, leading to a lower resource utilisation load. (b) If the application is deployed on a publicly accessible HPC centre, the jobs have to be put in a job queue. In this case, the user competes for the resources and has to wait for uncertain time before his/her application can be executed.

To address the above issues, the main objective is to connect multiple HPC systems via cloud for the parallel execution of eScience simulations or applications. Thus, this approach helps to achieve the desired higher resource utilization level, better resource usability, and reduce the administrative overhead for the users by providing a *“Simulation as a Service”* in the cloud.

7.1.1 Selection of the use case scenario

Molecular Dynamics (MD) or Computational Fluid Dynamics (CFD) simulations are highly representative for modern eScience research tasks. This kind of calculations provides information about how a given substance behaves under a given set of physical conditions, e.g. to predict material behaviour for industrial purposes. The same calculations have to be executed multiple times by sweeping the parameter values through the parameter range of each boundary condition. The process contains usually several iterations of execution, e.g. the first iteration performs a coarse granular simulation over selected points in the parameter space, while the second iteration runs fine granular simulations around the point that showed remarkable phenomenon in the first iteration. Simulations in different granularity have also different requirements on the capability of the resources.

In the automotive industry, a remarkable shift from design processes based on physical prototypes to a computationally-aided development process based on virtual prototypes is recognizable in the last couple of years. Especially in the concept phase, the most concept relevant decisions are made on the basis of simulation results. For the virtual prototype or the Computer Aided Engineering (CAE) case, simulations for the CFD and structural mechanical (Computational Structural Mechanics - CSM) design of the vehicles are carried out intensively in the early development phase.

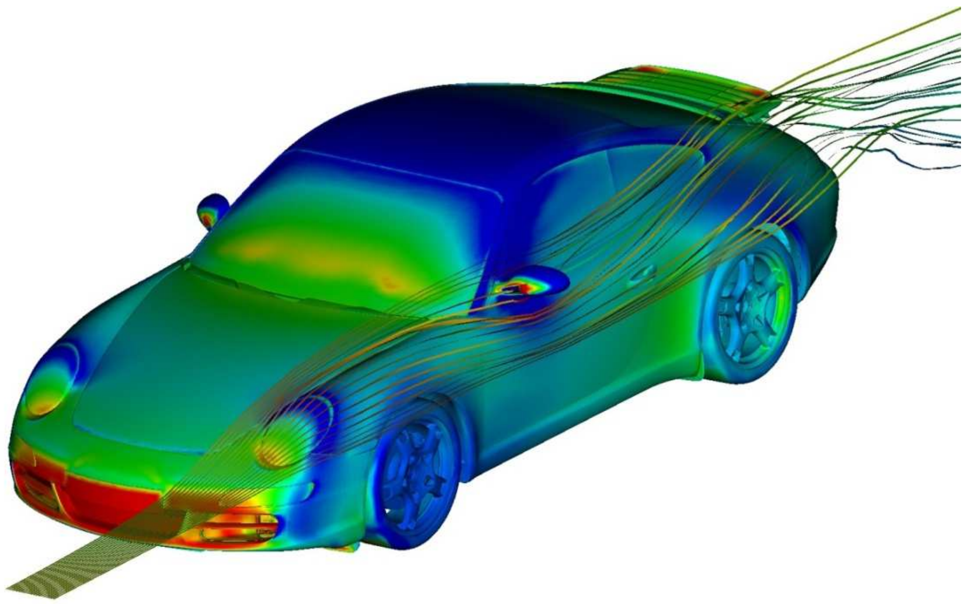


Figure 7.1: CFD simulation of a car side mirror.

An exemplary use case is the development of a side mirror, as shown in Figure 7.1. The current mirror development process combines both experimental techniques and simulation methods. Various areas of development are involved, such as styling, engineering, testing, simulation and approval. Basically, the three criteria: styling, field of view, and flow behaviour (including impact on fuel consumption and noise emission) need to be taken into account.

The process is as follows. In the early concept phase, several styling designs (e.g. 5 – 10) are created, either as plasticine models or virtually. In the very first step of deciding whether a mirror design "stays in the race" or not, corporate philosophy plays a much more important role than the field of view or flow conditions surrounding the mirror. Only the approved design proposals pass through the next stages, namely the field of view and flow analyses. The field of view can be verified with a relatively simple process. On the one hand, various mirror geometries are physically attached to the vehicle and then analysed and evaluated stationary and during driving. Obviously this is time-consuming and costly. On the other hand, more and more car manufacturers make use of modern virtual methods. Diverse virtual mirror geometries are instantaneously installed on a virtual driver's seat to perform studies of the content in the mirror and evaluate the visibility. Again some of the mirror designs might be discarded, while others undergo the most expensive or complex part of the development process, i.e. the CFD analysis. The traditional way is to perform wind tunnel experiments which require the use of full vehicles. Faster, cheaper and much more flexible is again the virtual counterpart, i.e. flow simulations on HPC machines, as shown in Figure 7.1. The investigated mirror designs are calculated, evaluated, compared to each other or wind tunnel results, and optimized from the flow and pollution point of view. The whole design process is iterative, and most often the best compromise between styling and functionality.

To summarize, the current side mirror development process includes experimental as well as virtual methods and still a lot of manual work regarding the simulations and evaluations of results which could be automated in the future.

7.1.2 Overview over the prototype

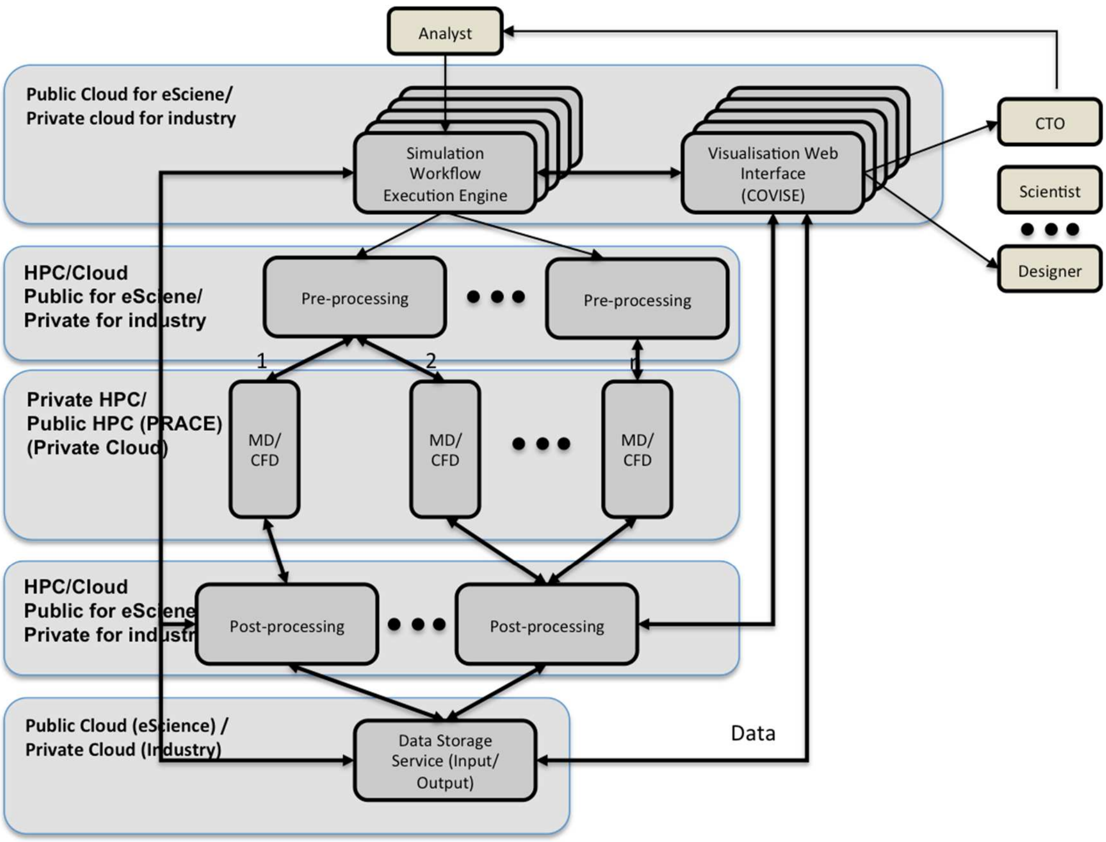


Figure 7.2: Architecture of target application across HPC and cloud.

In order to enable the execution of such simulation applications on various resource environments, PaaSage has to be able to support workflow-like applications such as depicted in Figure 7.2. An approach may involve the following main modules:

- **Workflow engine** is responsible for the configuration, instantiation, execution, monitoring and control of distributed tasks across cloud and HPC. This includes the necessary access rights, data conversion, scaling behaviour, implicit adaptation to the infrastructure and identification of appropriate distributed resources.
- **Visualization web application** is responsible for visualising the complex three-dimensional structure of the datasets in real-time. It enables users to analyse their datasets intuitively in a fully immersive environment through state of the art visualization techniques including volume rendering and fast sphere rendering. It is a module developed by HLRS within the COVISE project[7].
- **Pre-processing module** is responsible for preparing the input data together with the corresponding values for the initial and boundary conditions. The required data must meet precise requirements that strongly depend on the considered numerical method.
- **MD/CFD instances** are compute and communication intensive, and usually run using OpenMPI[8] and OpenMP[9]. The instantiation of one the simulations are dynamic during the execution of the workflow and the number of instances is depends on the application configuration and output of individual parameters probe.
- **Post-processing module** is responsible for analysing and preparing the output data for visualisation to end-user. It also allows dynamic update of simulation configurations like boundary conditions to run several iterations before desired result is found.
- **Centralised data storage** is mainly used to access persistent input/output files of the applications. The final result of entire parameter sweeps will be aggregated on the centralized storage.

Since users interact directly with the system through the interfaces of the workflow engine and/or the visualization service, they need to be deployed in a public or at least in a shared cloud environment for e-Science users, and in a private cloud environment for industry users, so that the users can access the application from anywhere and at any time. Taking advantages of the cloud could also ensure the availability and scalability of these modules. The simulation applications might involve different user groups e.g. academia, institute or car manufacturer. In order to realise multi-tenancy for serving those multiple user groups (tenants), separate software instances have to be set up. In addition, real-time requirements will necessitate low response time of the according services.

Due to the performance issues, the MD/CFD simulation modules have to be deployed to HPC or private cloud that provides compatible performance. The simulation modules will be instantiated at run time and the application code together with input

[7] http://www.hlrs.de/organization/av/vis/covise/
[8] http://www.open-mpi.org/
[9] http://www.openmp.org/

data are to be staged in to allocated compute resources. It has to be noted that, as shown in Figure 7.2, the simulation module itself is a sub-workflow and contains different computation steps that can be roughly categorized into three groups: pre-processing, simulation and post-processing. These steps can be similarly treated as individual logical blocks or modules with individual scaling behaviour.

The pre- and post-processing modules could be deployed to HPC or cloud depending on a specific use case, e.g. the requirements of concrete tools/algorithms based on the capability of the resources. Different instances of pre- or post-processing with different configurations are required for different number of simulations. They should also scale out/in together with the simulation modules to improve the performance.

Regarding the data storage service, strong consistent cloud storage is required due to the parallel read/write and there is large-volume data transfer (up to several GBs depending on problem size) between the cloud storage and other modules. Depending on the specific case, the results may be shared publicly, in which case the data storage service may be hosted in a public cloud. However, industrial use cases will insist on private deployment and maximum security.

7.1.3 Motivation for the Cloud

HPC plays an incomparable role in industrial areas and academic researches, particularly for compute-intensive applications. However, scientific computing requires an ever-increasing number of heterogeneous resources to deliver results for growing problem sizes in a reasonable timeframe. With the current business procedure of HPC, it is difficult for users to access and manage the execution of such applications. With the recent cloud hype, there has been a growing interest from the e-Science and HPC community to exploit cloud infrastructure, as they seem to offer just the capabilities required by the researchers because of its well-known advantages:

- Strong computing resources (scalability)
- “on-demand resources” (elasticity)
- High availability
- High reliability,
- Large data scope
- Reduced capital expenditure (cheap).

Cloud appeals to the scientists that need resources immediately and temporarily. Scientific applications with minimal communication and I/O are also best suited for clouds. Thus, the HPC community would benefit mostly from a combination of the strength of the two environments.

7.2 Scenario Description

The below scenario relates to scaling in the application due to job completion and/or cost issue.

Use Case	Scale in of an application
Description	Scale in or reducing the number of running VMs
Prerequisites (Dependencies) & Assumptions	None
Steps	1. The PaaSage platform monitors the relevant KPIs, such as CPU load of VMs and the current running cost 2. The PaaSage platform detects job completion or the running cost is close to the threshold 3. The PaaSage platform performs a scale in by shutting down idle and unused VMs 4. The PaaSage platform monitors the above KPIs
Variations (optional)	
Quality Attributes	• Scale in due to job completion is defined in RIS-SI-RUN-1 • Scale in due to running cost is described in RIS-SI-CST-1
Issues	Job completion and running cost

Scenario Id	**RIS-SI-RUN-1**
Scenario Name	**Scale in due to the job completion**
Scenario Type	Shutting down unused or idle VMs
Artefact	Pre-processing and post-processing modules along with MD/CFD instances
Context	Shutting down unused or idle VMs due to job completion
Stimulus	The running job signals its completion or the running VMs have low or zero CPU loads over the last minutes.
Response	The system shutdowns unused or idle VMs.
Response Measure	The low or zero CPU load reported by the monitoring facility exceeds a specified time limit.

Scenario Id	**RIS-SI-CST-1**
Scenario Name	**Scale in due to higher running cost**

Scenario Type	Shutting down idle and/or running VMs
Artefact	Pre-processing and post-processing modules along with MD/CFD instances
Context	Shutting down idle and/or running VMs in order to reduce total running costs
Stimulus	The system detects an issue regarding to higher running cost than estimated.
Response	The system shutdowns idle and/or running VMs.
Response Measure	The running cost detected by the system exceeds the pre-defined cost estimation and/or the total cost.

The below scenario relates to scaling out or migrating the application to a different provider due to waiting time in the job queue.

Use Case	Scale out or migration of an application
Description	Scale out or migrate the application to a different provider
Prerequisites (Dependencies) & Assumptions	None
Steps	1. Performance KPI is violated 2. The PaaSage platform seeks and finds an alternative solution 3. The PaaSage platform performs a scale out 4. The PaaSage platform monitors the relevant KPI
Variations (optional)	
Quality Attributes	• Scale out due to job queue issues are defined in RIS-SO-JBQ-1 and RIS-SO-JBQ-2 • Scale out due to network latency issues are described in RIS-SO-NET-1, RIS-SO-NET-2, RIS-SO-NET-3 and RIS-SO-NET-4 • Scale out due to cost minimisation objective is listed in RIS-SO-CST-1 • Scale out due to licensing issue is mentioned in RIS-SO-LCS1
Issues	Job queue, network latency, cost minimisation and licensing issue

Scenario Id	**RIS-SO-JBQ-1**
Scenario Name	**Scale out due to long waiting time in the job queue**
Scenario Type	Reduction of the waiting time in the job queue
Artefact	Pre-processing and post-processing modules along with MD/CFD instances
Context	The queue waiting time is higher as a configurable limit since a specified amount of time.
Stimulus	Queue monitoring signals a significant problem over the last minutes about the long waiting period.
Response	The system moves the application onto a different testbed. This might be a different HPC or cloud provider that is located nearby.
Response Measure	The waiting time reported by the monitoring facility exceeds a specified time limit.

Scenario Id	**RIS-SO-JBQ-2**
Scenario Name	**Scale out due to high number of jobs in the queue**
Scenario Type	Reduction of the waiting time in the job queue
Artefact	Pre-processing and post-processing modules along with MD/CFD instances
Context	The number of jobs that are in front of the queue is higher than a pre-defined threshold or limit.
Stimulus	Queue monitoring signals a significant problem over the last minutes about the high number of jobs exceeding the threshold.
Response	The system moves the application onto a different testbed. This might be a different HPC or cloud provider that is located nearby.
Response Measure	The number of jobs reported by the monitoring facility exceeds a pre-defined threshold or limit.

The below scenario relates to scaling out or migrating the application to a different provider due to network issues.

Scenario Id	**RIS-SO-NET-1**
Scenario Name	**Scale out due to network latency**
Scenario Type	Network latency reduction
Artefact	Pre-processing and post-processing modules, MD/CFD instances, and visualisation web component.
Context	The (network) latency time is higher as a configurable limit since a specified amount of time.

Stimulus	Network monitoring signals significant latency problem over the last minutes.
Response	The system moves the application onto a different testbed. This might be a different HPC or cloud provider that is located nearby.
Response Measure	The average latency reported by the network monitoring facility drops below a specified limit.

Scenario Id	**RIS-SO-NET-2**
Scenario Name	**Scale out due to low network bandwidth**
Scenario Type	Network bandwidth consumption
Artefact	Workflow engine and visualisation web component.
Context	The network bandwidth is lower as a configurable limit since a specified amount of time.
Stimulus	Network monitoring signals significant bandwidth problem over the last minutes.
Response	The system moves the affected artefacts closer to the centralised data storage.
Response Measure	The average bandwidth by the network monitoring facility drops below a specified limit.

Scenario Id	**RIS-SO-NET-3**
Scenario Name	**Scale out due to Network File System (NFS) issues**
Scenario Type	Problems with NFS with respect to response time and high latency
Artefact	MD/CFD instances and post-processing module
Context	NFS is not responsive or has a high latency after a period of time
Stimulus	Network monitoring signals significant NFS problem over the last minutes.
Response	The system moves the affected artefacts onto a different testbed. This might be a different HPC or cloud provider that is located nearby.
Response Measure	The average response time by the network monitoring facility increases above a specified limit.

Scenario Id	**RIS-SO-NET-4**
Scenario Name	**Scale out due to missing connectivity to the license server**
Scenario Type	Problems with the authorisation of the license key on external license server due to network error

Artefact	MD/CFD instances, and pre-processing and post-processing modules
Context	No connection to license server
Stimulus	Connection error to the license server
Response	The system moves the application onto a different testbed with a successful connectivity to the license server.
Response Measure	Connection error to the license server

The below scenario relates to scaling out or migrating the application to a different provider due to relevant costs like usage and license fees.

Scenario Id	**RIS-SO-CST-1**
Scenario Name	**Scale out due to cost minimisation**
Scenario Type	Minimising total cost
Artefact	Pre- and post-processing modules and MD/CFD instances
Context	Higher cost than other testbeds or providers
Stimulus	The estimated / initial cost calculation is higher than a defined threshold
Response	The system runs the user application onto a cheaper testbed.
Response Measure	The estimated / initial cost calculation is below the defined threshold

Scenario Id	**RIS-SO-LCS-1**
Scenario Name	**Scale out due to licensing issue**
Scenario Type	Licensing issue
Artefact	Pre- and post-processing modules and MD/CFD instances
Context	The licensing issue prohibits the use of particular libraries or programs outside the specified premises
Stimulus	License restriction imposed by the system administrator or owner
Response	The system runs the user application into a testbed that has the proper license.
Response Measure	License restriction imposed by the system administrator or owner

The below scenario relates to handling confidential data.

Use Case	Handling confidential data

Description	Confidential data need to be stored and run in trusted providers
Prerequisites (Dependencies) & Assumptions	None
Steps	1. Performance KPI is defined in Service Level Agreement (SLA) 2. The PaaSage platform seeks and finds a solution that satisfies the SLA 3. The PaaSage platform executes the application 4. The PaaSage platform monitors the relevant KPI
Variations (optional)	
Quality Attributes	Described in more detail in the quality attribute scenarios: RIS-DT-LOC-1 and RIS-DT-LOC-2
Issues	The location of storing and running confidential data

Scenario Id	**RIS-DT-LOC-1**
Scenario Name	**Storing confidential data**
Scenario Type	Data location problem
Artefact	Centralised data storage
Context	Confidential data need to be stored locally (private cloud) or a trusted HPC centre.
Stimulus	SLA specifically defines where to store the data.
Response	The system runs the application on a private cloud or a trusted HPC centre.
Response Measure	The location of running application.

Scenario Id	**RIS-DT-LOC-2**
Scenario Name	**Running confidential data**
Scenario Type	Data location problem
Artefact	Pre- and post-processing modules and MD/CFD instances
Context	Applications that use confidential data need to be run locally (private cloud) or on a trusted HPC centre.
Stimulus	SLA specifically defines where to run the applications.
Response	The system runs the application on a private cloud or a trusted HPC centre.

Response Measure	The location of running applications.

The below scenario relates to authorisation issues.

Use Case	Authorisation issues
Description	Managing user roles and permissions as well as user groups
Prerequisites (Dependencies) & Assumptions	None
Steps	1. The PaaSage platform enables system administrators to create and define several user roles and permissions as well as user groups 2. The system administrators create users and allocate them to the appropriate roles, permissions and groups 3. The PaaSage platform seeks and finds a solution that satisfies the SLA 4. The PaaSage platform gives system and data access only to the authorised users
Variations (optional)	
Quality Attributes	Described in more detail in the quality attribute scenario: RIS-AT-MGT-1 and RIS-AT-MGT-2.
Issues	User access to system and data

Scenario Id	**RIS-AT-MGT-1**
Scenario Name	**Managing user roles and permissions**
Scenario Type	User management
Artefact	Workflow engine, centralised data storage and visualization web application
Context	Access to data is regulated, thus, only authorized users are allowed.
Stimulus	Each user account is associated with one or more roles / permissions.
Response	Users are given access to the system according to their roles and permissions.
Response Measure	None

Scenario Id	**RIS-AT-MGT-2**
Scenario Name	**Managing user groups**
Scenario Type	User management
Artefact	Centralised data storage and visualization web application
Context	Users can belong to one or more collaboration groups for a limited time only, i.e. during the project duration.
Stimulus	Each user account is associated with one or more roles / permissions.
Response	The system denies access to the data for non-authorized users.
Response Measure	None

The below scenario relates to security issues.

Use Case	Security
Description	Security-related issues when running applications
Prerequisites (Dependencies) & Assumptions	None
Steps	1. Users defined security requirements related to the network communication and single tenancy mode 2. The PaaSage platform provides a secure access to the testbed and experiment results via HTTPS, SSH and/or VPN 3. The PaaSage platform seeks providers that can provide a secure access and able to run applications in a single tenancy mode 4. The PaaSage platform executes the application 5. The PaaSage platform monitors network access and single tenancy mode
Variations (optional)	
Quality Attributes	Described in more detail in the quality attribute scenarios: RIS-SC-ENC-1 and RIS-SC-TCY-1
Issues	Network communication and single tenancy

Scenario Id	**RIS-SC-ENC-1**
Scenario Name	**Using secure network communication**

Scenario Type	Secure network communication
Artefact	Workflow engine, centralised data storage and visualization web application
Context	Network communication among the artefacts shall be encrypted
Stimulus	The use of HTTPS, SSH and/or VPN to access the testbed and experiment results
Response	The system choses a provider that offers secure network communication
Response Measure	None

Scenario Id	**RIS-SC-TCY-1**
Scenario Name	**Host all running VMs to be executed on the same physical servers (single tenancy mode).**
Scenario Type	Single tenancy mode
Artefact	Pre- and post-processing modules and MD/CFD instances
Context	When executing the application on the cloud, all running VMs shall be executed on the same physical servers.
Stimulus	SLA defines this requirement
Response	The system allocates dedicated physical servers to run all the VMs
Response Measure	None

7.3 *Scenario Grouping*

This section groups several scenarios together for easier reference.

Scenario Group Id	Scenario Ids	Description
RIS-SIG-JOB	RIS-SI-RUN-1, RIS-SI-CST-1	Scaling in scenarios triggered by job related stimuli
RIS-SOG-JBQ	RIS-SO-JBQ-1, RIS-SO-JBQ-2	Scaling out scenarios triggered by job queue related stimuli
RIS-SOG-NET	RIS-SO-NET-1, RIS-SO-NET-2, RIS-SO-NET-3, RIS-SO-NET-4	Scaling out scenarios triggered by network latency related stimuli
RIS-SOG-CST	RIS-SO-CST-1, RIS-SO-LCS-1	Scaling out scenarios triggered by cost related stimuli
RIS-DTG-LOC	RIS-DT-LOC-1, RIS-DT-LOC-2	Scaling out scenarios triggered by data related stimuli
RIS-ATG-	RIS-AT-MGT-1, RIS-AT-MGT-2	Scaling out scenarios triggered by

MGT		authorisation related stimuli
RIS-SCG-ENC	RIS-SC-ENC-1, RIS-SC-TCY-1	Scaling out scenarios triggered by security related stimuli

7.4 Traceability with respect to PaaSage components

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
RIS-SIG-JOB	The criteria about the CPU-related metrics and cost are defined in CAMEL.	The Profiler analyses the CAMEL model and provides a list of providers that matches the defined criteria. The Reasoner uses the metrics from the Executionware monitoring facility to select the best matching provider which satisfies the cost parameters, the SLA definitions etc. For modifications of the deployment, the Adapter queries the MDDB to find a different solution which still satisfies the defined criteria.	The monitoring facility collects all CPU-related metrics and running costs. If necessary, the Executionware (adaptation and control) shutdowns idle and/or running VMs.	Records summarized data sets about CPU-related metric and cost for each provider from previous runs and from the (external) PaaSage community.
RIS-SOG-JBQ	The criteria about job queue are defined in Saloon.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider with the min. average waiting time in the last few minutes. The Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware measures the average waiting time in the job queue. Components are moved to the other suitable provider.	Storing the data about average waiting time in the job queue from different providers.
RIS-SOG-NET	Latency specification for network related components. Defined in WS-Agreement.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider with the low network latency in the last few minutes. The Adapter queries MDDB to find a better provider which	Executionware measures the latency of network-related components. Components are moved to the other suitable provider.	Storing the data about network latency and the locations serviced from different providers.

		still satisfies the SLA, and generates a reconfiguration plan.		
RIS-SOG-CST	The criteria about cost parameters and objectives are defined in Saloon and WS-Agreement.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider that satisfies the cost parameters and objectives. The Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware measures usage and license costs of running applications. Components are moved to the other suitable provider.	Storing the data about usage and license costs from different providers.
RIS-DTG-LOC	The criteria about data location and their usage are defined in the following DSLs: CloudML, Saloon and Security.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider that satisfies the models. The Adapter queries MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware provides information regarding to the location of data and running applications. Components are moved to the other suitable provider.	Storing the information about the location of data and running applications.
RIS-ATG-MGT	The criteria about user and group managements are defined in CERIF and Security.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider that satisfies the models. The Adapter performs high-level application management, which involves monitoring components and usage on multiple cloud providers.	Executionware sets the appropriate permissions to users, groups, data and running applications.	Storing the information about user roles, groups, and permissions.
RIS-SCG-ENC	The criteria about security and running VMs are defined in Saloon, CloudML and Security DSLs.	The Profiler analyses the CAMEL configuration model & produces a list of providers that satisfy the SLA. The Reasoner uses the monitoring data to choose a provider that satisfies the models. The Adapter performs high-level application management, which involves monitoring components and usage on multiple cloud providers.	Executionware uses secure communication and runs the VMs as specified in the model.	Storing the information about security features and functionalities, and the location of running VMs.

7.5 *Traceability to the integration tests*

Integration test scenario group	Scenario group Id	Description
RIS-ITG-1	RIS-DTG-LOC, RIS-ATG-MGT, RIS-SCG-ENC	Integration test which focuses on the data privacy and security issues.
RIS-ITG-2	RIS-SIG-JOB, RIS-SOG-JBQ, RIS-SOG-NET, RIS-SOG-CST	Integration test which focuses on scale-in and scale-out scenarios.

8 Public sector – human milk bank

This case is supported by the EVRY partner. The description is preliminary, due to the substitution of this use case.

8.1 *Objectives (revised)*

EVRY is the largest IT Company in Norway and the second largest IT services company in the Nordic region. With 10.000 employees, EVRY delivers daily IT services from 50 Nordic towns and cities for more than 14.000 public and private sector customers. EVRY provides very extensive deliveries to Norwegian and Nordic companies, financial institutions, national public sector entities, municipalities and health authorities. EVRY is the force behind a whole range of innovations that have transformed and simplified the way people access services across society. Around 1 million Norwegians use services delivered by EVRY each day. According to EVRY's estimates virtually the entire Norwegian population has used IT services delivered by EVRY over the course of each week.

EVRY has a large business group focusing on public sector. EVRY’s ERP and case management solutions for the public sector alone support 70% of all citizens in Norway. Its strategy is to maintain and develop its strong position in public sector, creating value for its customers to the benefit of society.

The Norwegian and European public sectors are under pressure to develop more efficient ways of providing services for the inhabitants and businesses. In the next ten years, the demographics of Norway will go through a significant shift where a large proportion of the population will transfer from being of working age into retirement. This will give two effects on the public sector: the demand for public services will increase significantly; and there will be a reduction in the total size of the workforce. ICT and cloud will be a significant driver to reduce the negative sides of the demographic change.

There are currently 428 municipalities, 19 regional mid-level governmental districts and a numerous of central government units including regional health authorities which owns all public hospitals. For many of the public services, integration across public units and across different technologies are required. The result is an enormous architectural challenge in order to modernise public sector.

8.1.1 Selection of the use case scenario

Cloud computing will be an important enabler in the digitalization of public sector.

Information security and control on data is important for our customers and public sector. Therefore, our main strategy is to deliver cloud service from EVRY’s data centre – typical a “private cloud”. But since there are so many different systems and vendors, including on-premise legacy applications – we need to handle “multiple hybrid cloud”.

EVRY has selected the “Human Milk Bank” project as a use case and pilot for PaaSage. This project is a part of our Health Care initiative in business group Public Sector.

The milk bank project will simplify, increase traceability, increase security and increase collaboration for the human milk banks. The milk bank collects milk from mothers that has the ability to produce more than needed in order to give to mothers that not are able to produce enough or at all. It is a market between mothers and the milk banks (a hospital) and between milk banks (hospitals). Human milk bank and breast milk has a lot of focus in other parts of the world, including Europe, both for the obvious health reasons and for economic reasons (is cheaper).

The pilot developed in Norway can potentially be rolled out to hundreds of milk banks other parts of the world. In Norway we have today 12 milk banks. In US and Canada together there are only 19(!) One of the reasons for this low number is security concerns and complexity. The milk bank use case wants to solve this. This solution could also potentially be used to donate other kinds of fluids.

Figure 8.1: Active vs. planned milk banks in Europe

From a technical point of view we are talking about private or community cloud, UI for the laboratory workers, data register, data exchange and integration. Hand held units and potentially self-service on mobile. Also easy rollout to other regions. The application is based on Microsoft. This alone will have a value for PaaSage since a true multi-Cloud deployment platform must support the Microsoft stack – like it or not.

The milk bank solution will be built as a true cloud application with support for multiple languages. Phase 1 will be to deploy on one milk bank pilot. If this succeeds, it can be deployed nationally and last prepared for global deployment.

The first version of the pilot does not necessarily need the cloud or PaaSage. It can be implemented as a traditional web application. But rolling out this application to a larger scale of milk banks and still keep the operational and maintenance cost down, cloud is essential. And since the customers have their own preferences on cloud and are location sensitive, we are talking about deployment to multiple clouds. This is the sweet spot of PaaSage.

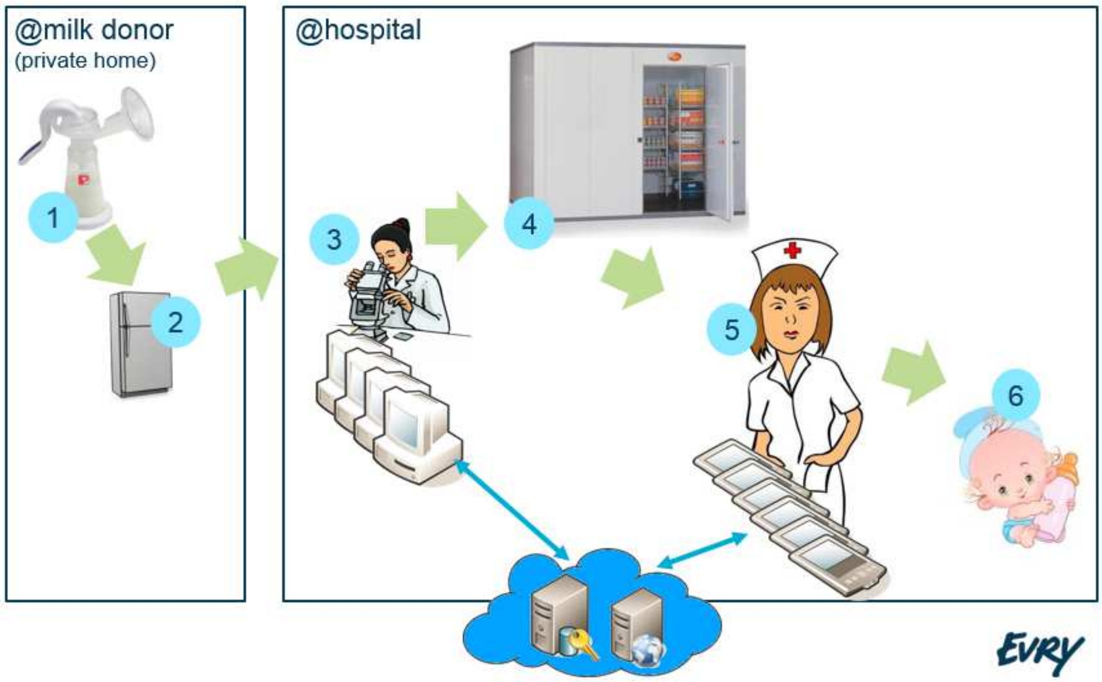
The Public Sector Milk bank case will therefore demonstrate two scenarios:

1. *The deployment and use of Microsoft technology in PaaSage*
2. *Modelling and deployment of an application in multiple clouds.*

8.1.2 Overview of the prototype

EVRY is working on a pilot of the milk bank solution. This pilot will be deployed for a single milk bank. The ambition is to roll this out national if the pilot is a success.

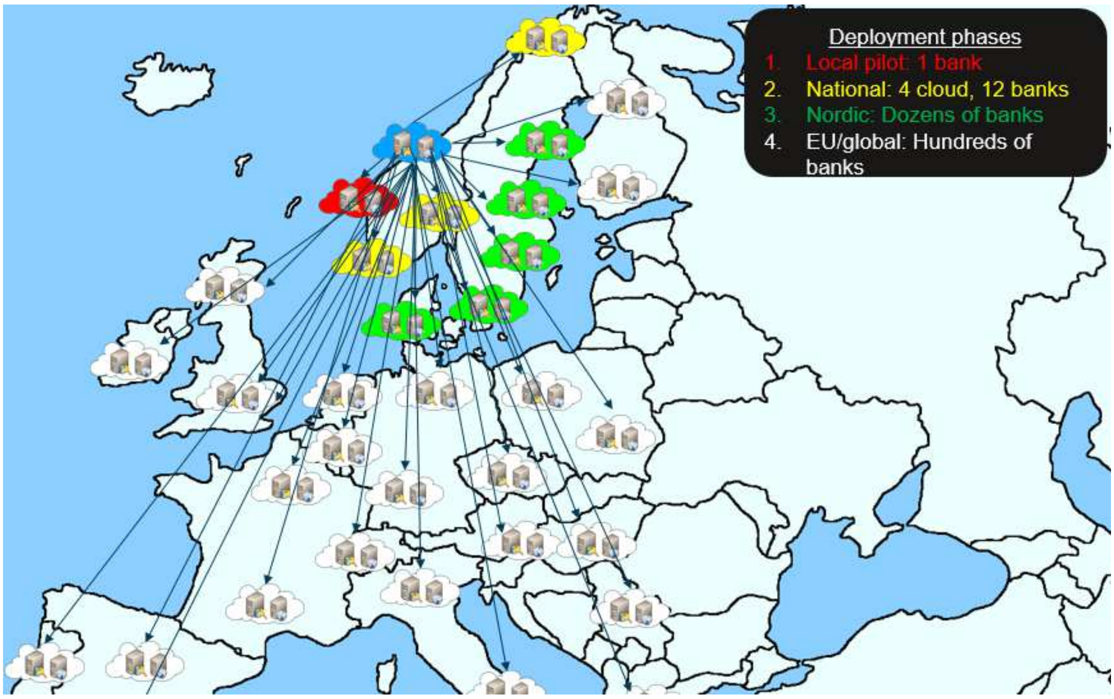
The figure below shows the workflow and architectural overview for the stand-alone pilot. This pilot will be developed as a pure cloud application with multi language support. The milk is delivered frozen at the hospital (step 2) and is analyzed, controlled and registered in the lab (step 3) before it is transferred to the milk bank stock (step 4). The nurse brings the milk to the children in the hospital. She uses a hand held device to scan the milk and child in order to secure traceability.



After a successful pilot deployment (phase 1), the ambition is to roll the solution out nationally in Norway (phase 2). The four regional health authorities in Norway has in total 12 milk banks. The regional health authority organizes and handles ICT separately and potentially can have their own cloud preferences.

Given a successful launch in Norway, EVRY can deliver milk bank solutions to the Nordic countries (phase 3) and the rest of the world (phase 4). Different nations have different politics and ways of handle ICT and solutions within health care. In order to deliver a cost efficient cloud solution in that landscape we need to handle multiple clouds in an efficient manner.

With PaaSage, we should be able to model the total architecture of our managed service for the milk banks. Helping us to develop the application once and deploy it cross-cloud. Managing the solution like it was deployed to one single cloud.



The figure above illustrates the potential roll out phases and shows the complexity of multi cloud, and how we would like to build “one single virtual cloud”.

The figure below show a few GUI mockups for the milk bank solutions.

8.1.3 Motivation for the Cloud

There are currently 428 municipalities, 19 regional mid-level governmental districts and many central government institutions in Norway today. The hospitals are grouped into four regional health authorities. All of these units are more or less autonomous regarding ICT services (there are some national regulations). Simplification, standardization, reuse and scalability are important driver's in the future public ICT architecture. In order to meet future challenges and be a catalyst for innovation and digitalization of public sector, EVRY must continue to master the complexity of the cloud and develop new sustainable business models.

Despite simplification, standardization and reuse, multiple clouds are unavoidable due to regulations and autonomous public unites. The need for multiple clouds will be enhanced approaching other countries. Many of the public ICT services/applications are small and not necessary commercial attractive in such small scale. In order to make it sustainable for commercial companies, the services/applications must be delivered in a large volume. But as stated above, large volume can lead to multiple clouds which can be costly to manage. In a large global deployment it is also important to constantly be able to easily change cloud providers according to the customer preferences and needs. Typical new regulations force the public unit to change the underlying cloud provider or move the application/data.

The human milk bank is an example of a relatively simple application which, isolated to one bank/installation, may have a limited commercial potential. This again can make it challenging for one single milk bank to carry the cost. In order to get volume, multiple clouds are required due to regulations and autonomous public units.

To make the cost as low as possible for customers in the public sector, EVRY would like to manage multiple clouds as one "virtual cloud". To do so we need a technology like PaaSage. For customers that are not that sensitive to location or cloud vendor, EVRY can use PaaSage to select optimal cloud vendor according to their requirements.

8.2 Scenario Description

Use Case	Handle Microsoft application in multiple clouds
Description	Manage a Microsoft application in multiple heterogeneous clouds as one single "virtual cloud"
Prerequisites (Dependencies) & Assumptions	PaaSage must support Microsoft technology
Steps	Change in global milk bank architecture needed a. New milk bank deployment b. Upgrade of milk banks c. Emergency patch of single bank d. Termination of milk bank

	e. Move milk bank Depending on scenario, the deployment is triggered by the cloud deployment manager manually according to the deployment scenario modelled in PaaSage.
Variations (optional)	
Quality Attributes	
Issues	Cost optimization, heterogeneous multi cloud orchestration and management

Scenario Id	PUBL-NEW
Scenario Name	New milk bank deployment
Scenario Type	Multi-customer
Artefact	Web application, database and related artefacts
Context	New planned customer (milk bank) introduced in the global architecture. New deployment model created in PaaSage and will be triggered manually. Target scripts/adapters are ready.
Stimulus	Planned deployment as a result of a new customer agreement
Response	PaaSage will interpret the deployment model and act accordingly
Response Measure	Cost and complexity reduced. The Human Milk bank application can be managed as one virtual cloud application even though it is running on a heterogeneous multi-cloud architecture.

Scenario Id	PUB-UPGR
Scenario Name	Planned upgrade/release of all applications in the deployment architecture
Scenario Type	Multi-customer
Artefact	Web application, database and related artefacts
Context	A planned new release of all deployed applications modelled in PaaSage.
Stimulus	Planned deployment as a result of a planned release
Response	PaaSage will interpret the deployment model and act accordingly
Response Measure	Cost, complexity and risk reduced. The Human Milk bank application can be managed as one virtual cloud application even though it is running on a heterogeneous multi-cloud architecture.

Scenario Id	PUB-FIX
Scenario Name	Quick fix or patch of a single milk bank application due to a local error situation. To be followed by PUB-UPGR.
Scenario Type	Multi-customer
Artefact	Web application, database and related artefacts
Context	Error situation at a single milk bank. Error identified and fixed but will only be rolled out to the affected cloud. The fix will be upgraded in the next release that will be rolled out globally ASAP according to the PUB-UPGR scenario.
Stimulus	Planned single deployment as a result of an error fix
Response	PaaSage will interpret the deployment model and act accordingly
Response Measure	Cost, complexity and risk reduced. The Human Milk bank application can be managed as one virtual cloud application even though it is running on a heterogeneous multi-cloud architecture.

Scenario Id	PUB-TERM
Scenario Name	Termination of an existing milk bank
Scenario Type	Multi-customer
Artefact	Web application, database and related artefacts
Context	And exiting milk bank agreement is terminated and removed from the deployment architecture
Stimulus	Planned clean-up and removal of an application as a result of a terminated agreement
Response	PaaSage will interpret the deployment model and act accordingly
Response Measure	Cost and complexity reduced. The Human Milk bank application can be managed as one virtual cloud application even though it is running on a heterogeneous multi-cloud architecture.

8.3 Scenario Grouping

Scenario Group Id	Scenario Ids	Description
PUBG-DEPL	PUB-DEPL, PUB-UPGR, PUB-FIX	Deployment scenarios
PUBG-TERM	PUB-TERM	Clean-up scenario
PUBG-MOVE	PUB-DEPL, PUB-TERM	Move an application to new cloud NOTE: The DB is handled manually in this scenario

8.4 Traceability with respect to PaaSage components

Scenario Group Id	CAMEL (Appl. Model)	Metadata (Profiler, Reasoner, Adapter)	Executionware (control, monitoring, adaptation)	Community/ MDDB
PUBG-DEPL	Global milk bank architecture defined in CAMEL	The Profiler analyses the CAMEL configuration model and verify the SLA. The Reasoner uses the monitoring data to choose a provider that satisfies the models (if requested). The Adapter performs high-level application management.	Executionware provides information regarding to the location of data and running applications. It provides an overall picture of the health and trends of the global deployment architecture.	Storing the information about the location of data and running applications. Storing the information about user roles, groups, and permissions.
PUBG-TERM	Respective application removed from CAMEL	The Profiler analyses the instructions in CAMEL and the Adapter scripts the local clean-up procedure.	Executionware provides information regarding to the location of data and running applications. Secure traceability and audit of data and the clean-up.	
PUBG-MOVE	Instructions of move and clean-up defined in CAMEL	The Adapter can query the MDDB to find a better provider which still satisfies the SLA, and generates a reconfiguration plan.	Executionware provides information regarding to the location of data and running applications. Secure traceability and audit of data.	

8.5 Traceability to the integration tests

Integration test scenario group	Scenario group Id	Description
PUB-ITG-1	PUBG-DEPL, PUBG-MOVE	Integration test which focuses on deployment
PUB-ITG-2	PUBG-TERM	Integration test which focuses on clean-up

9 Generic Requirements on the PaaSage Platform

This section describes some generic use cases for the PaaSage platform. The use cases are based on discussions with case study partners and technical partners.

9.1 Basic Multi-Cloud Deploy Un-deploy and Redeploy

Use Case	**Basic Multi-cloud deploy/un-deploy/redeploy**
Description	Company A un-deploys n-tier application from private cloud and redeploys on public cloud: • Database contains personal data (Privacy) • Data must stay in country (Data location)
Prerequisites (Dependencies) & Assumptions	It is assumed that the deployment is multi-cloud, i.e. that it can involve several cloud providers.
Steps	1. The system administrator has deployed a multi cloud application in a private cloud with the PaaSage platform 2. The system administrator requests the PaaSage platform to un-deploy the multi cloud application. The Executionware stops the VM and saves system images. 3. The PaaSage Reasoner calculates the best possible deployment that meets the applications requirements, and presents it to the system administrator. 4. The system administrator confirms the proposed deployment 5. The Reasoner then passes the deployment to the Adapter and the Executionware to deploy it on a target cloud
Variations (optional)	
Quality Attributes	
Issues	

Figure 9.1: Un-deploy, redeploy scenario

The above figure illustrates a scenario of this use case. Company A has a private cloud and is running an n-tier application, along with many other applications. This is described in the "As Is" state. The IT administrator must perform some maintenance on the machines of the n-tier application. Unfortunately there are no more available machines in the private cloud. Using the PaaSage platform the IT administrator un-deploys the n-tier application from the private cloud and redeploys it on a public cloud in Europe to meet data location and privacy constraints. The resulting state is described as the "To Be" state where we see that the private cloud initially deployed components are now deployed in the public cloud.

9.2 Hybrid Cloud with Scale out to Public Cloud

Use Case	**Hybrid cloud with scale-out to public cloud**
Description	Storyboard Company A deploys n-tier application on hybrid cloud (private with scale out to public cloud) • Database contains personal data (Privacy) • Data must stay in country (Data location)
Prerequisites (Dependencies) & Assumptions	The application has been designed to be deployed in a cloud.
Steps	1. The system administrator submits a deployment request for a n-tier application. 2. The Reasoner examines the application requirements and proposes to deploy the application components on a hybrid cloud, i.e. some of the components are deployed in the private cloud, and other components are deployed in the public cloud. The deployment is constrained by the requirements on personal data and the restriction on

	data location. 3. The system administrator accepts the deployment proposed by the Reasoner. 4. The Executionware deploys the components in the private cloud. 5. The Executionware deploys the components in the public cloud. This requires that the PaaSage platform have the credentials to access the public cloud. 6. The Executionware starts the application and starts monitoring its execution.
Variations (optional)	
Quality Attributes	• Privacy • Confidentiality • Data location
Issues	

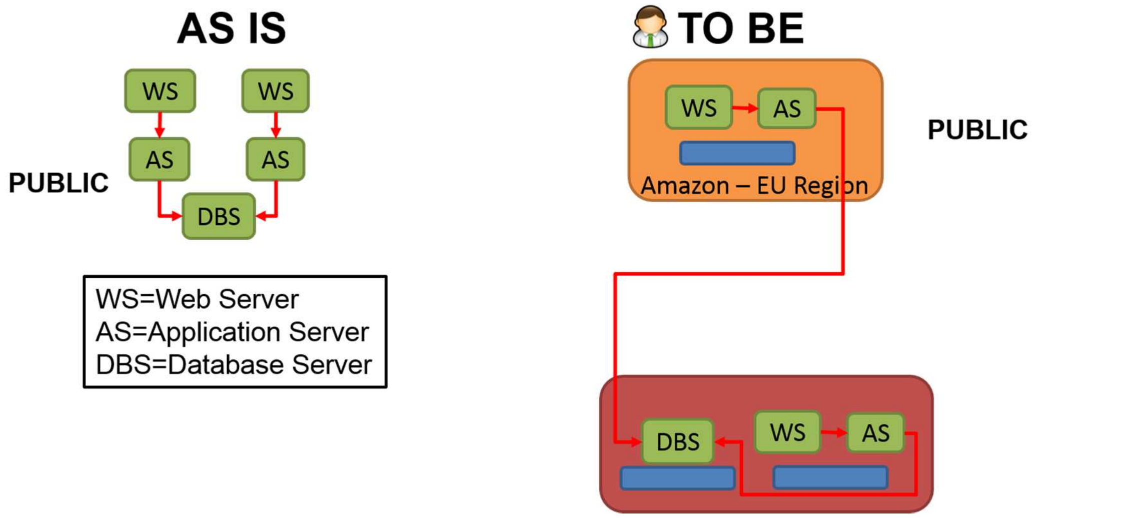


Figure 9.2: Hybrid Cloud with Scale out to Public Cloud

The above figure shows a scenario for the use case. Company A has a private cloud and decides to create a hybrid cloud with Amazon in the EU region. The initial state "As IS" is shown in the left part of the diagram: we can see that the n-tier application has two instances of application servers and two instances of web servers. The system administrator uses PaaSage to deploy its application components across the hybrid cloud. Since the database contains personal data, it is decided to keep the database within the data controller's, i.e. company A, domain. Furthermore since the data has to remain within the country it is kept in the private cloud. Some processing is done in Amazon, but within the Amazon EU region. The result of the deployment on the hybrid cloud is shown in the right part of the figure "To Be".

9.3 Cross Cloud Deployment to Optimize Cost

<table>
<tr><th>Use Case</th><th>Cross-cloud deployment to optimize cost</th></tr>
<tr><td>Description</td><td>Move to multiple clouds (cross clouds) to optimize the cost with data location constraint.</td></tr>
<tr><td>Prerequisites (Dependencies) & Assumptions</td><td></td></tr>
<tr><td>Steps</td><td>The system administrator requests PaaSage to deploy a n-tier application. The administrator provides an objective function for the deployment. The objective function specifies that cost of the deployment must be minimised.
The Reasoner retrieves the list of public clouds that meet the requirements. It finds a deployment that minimises cost while satisfying constraints such as data location, privacy or response time. The deployment is presented to the system administrator.
The system administrator evaluates the cost of the proposed deployment. He either accepts the deployment or requests a new deployment.
If the deployment is rejected, he can edit the application requirements such as the objective functions or constraints. The Reasoner goes back to step 2 and finds another deployment that minimises cost.
If the deployment is accepted, it is passed to the Adapter and Executionware to be deployed and monitored.</td></tr>
<tr><td>Variations (optional)</td><td></td></tr>
<tr><td>Quality Attributes</td><td>Cost
Data location</td></tr>
<tr><td>Issues</td><td></td></tr>
</table>

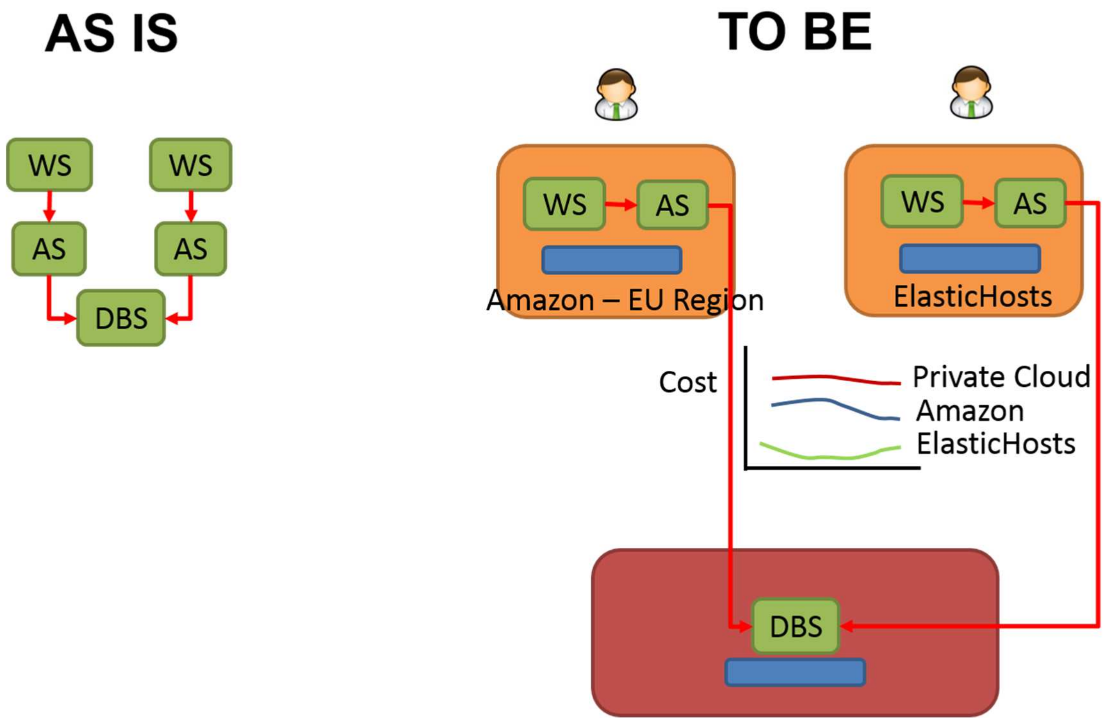


Figure 9.3: Cross Cloud Deployment to Optimize Cost

The above figure describes a scenario of this case study. Company A decides to use PaaSage to optimize cost when deploying its application components in the cloud. It also decides to work with public cloud provider to be able to reduce costs further. The PaaSage platform knows about the costs models of the private cloud and the two public clouds to optimize the deployments. The left part of the figure "As Is" shows the components of the n-tier application to be deployed. The left part of the figure "To Be" shows one deployment of the application components that minimises deployment cost. The deployment cost in time is shown for each provider: the cost of the private cloud is higher than the cost of the public cloud. The database server is the only component that is kept in the private cloud due to the personal data that needs to be protected.

9.4 Cross Cloud Scalability under Cost Objective

Use Case	**Cross-cloud scalability under cost objective**
Description	User load varies and the application must scale consistently across the different clouds. • Cost objective function must be respected. • Performance must be improved
Prerequisites (Dependencies) & Assumptions	The multi-cloud application is deployed across several cloud providers
Steps	1. The system administrator submits an application to be deployed. 2. The Reasoner proposes a deployment involving several cloud providers.

	3. The system administrator selects the proposed deployment; the Adapter and Executionware deploy it. 4. The Executionware monitors the deployment, and the monitoring data is sent to the Adapter. 5. Due to increasing user load, the adapter detects the need to scale out. 6. The Adapter sends scale out commands to the different clouds so that they scale in a coherent manner.
Variations (optional)	
Quality Attributes	• Cost • Performance • Data location
Issues	

Figure 9.4: illustrates the use case with a scenario. As the application becomes a success, more and more customers are using it. The application has been deployed on three different clouds: the database server has been deployed in the private cloud because it contains personal data, and web servers/applications servers have been deployed in public clouds near customer locations. The system load increases and the response time increases. PaaSage is used to manage the scalability of the application components across the private cloud and the two public clouds. Scalability must be taken into account by the Reasoner to ensure that the cost of the deployment is still being minimised. There have been some issues with response time, and customers complained. The Adapter monitors response time and scales across the different cloud providers. The left side of Figure 9.4: shows the different components of the n-tier application to be deployed. The right side of Figure 9.4: shows the application that has been scaled out on the two public clouds: new instances of the web server and application server have been created on each of the public clouds to meet the extra user load. The database server running in the private cloud has not been scaled out because it can handle the extra user load.

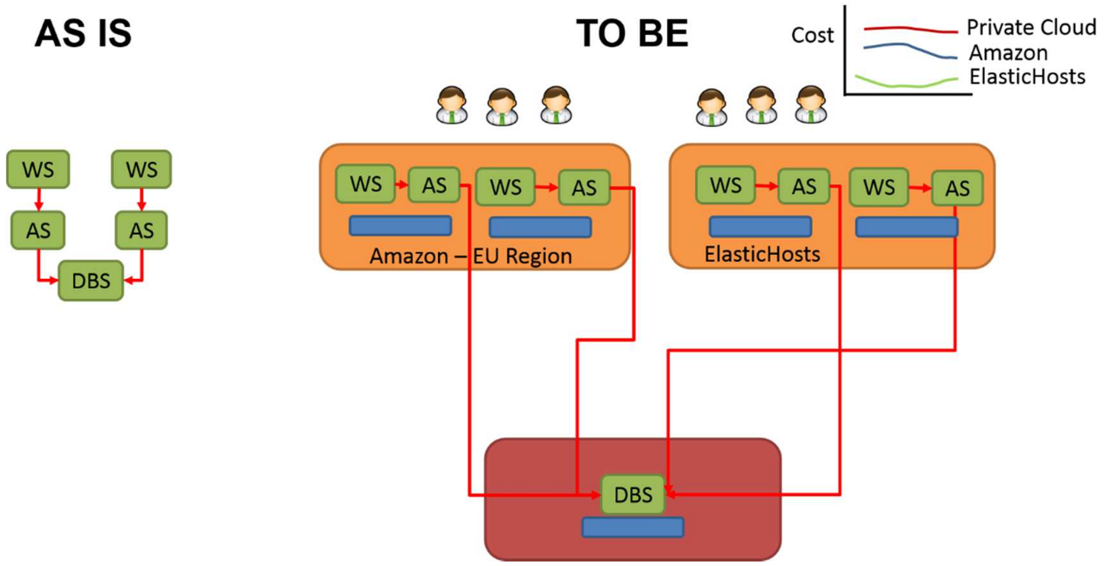


Figure 9.4: Cross Cloud Scalability under Cost objective

9.5 New Market - Add Cloud Provider / with Data Partitioning

Use Case	New Market - Add US cloud provider to serve clients in another country – with data partitioning
Description	Company opens new market in US. A US based public cloud provider. • Data partitioning is added to improve performance • Cost still to be minimised
Prerequisites (Dependencies) & Assumptions	A company that has a customer base in the US decides to provide services in a new country.
Steps	1. A company decides to provide its services and extend its current cloud deployment with a provider located in a new country. 2. The system administrator decides to partition the data: he decides to have one database server per country, and to integrate the country databases in a global database in the company's private cloud. The country databases contain personal data and will have to be secured appropriately. 3. The system administrator updates the CAMEL model accordingly and requests the PaaSage platform to update the current deployment. 4. The Reasoner proposes a new deployment of the multi-cloud application. 5. The system administrator accepts the proposed deployment. 6. The Reasoner requests the Adapter to un-deploy the current deployment and save the state of the system images. 7. The Reasoner then sends the new deployment model to the Adapter. 8. The Adapter and the Executionware execute the new deployment with database servers in each country.
Variations (optional)	
Quality Attributes	• Data security • Data location • Performance
Issues	

The figure below illustrates the use case. Company A decides to expand into the US market. It decides to deploy its application in Amazon US East region. With the growing size of the database and issues with response time, it is decided to partition the database into two: one for the EU region, and one for the US region. Integration of data from both of these databases is done in the private cloud. Because the data contains personal data, it is decided to deploy the database in the Amazon virtual private cloud (VPC) of each region and to use a VPN connection. These new constraints must be taken into account by the Reasoner to continue minimizing cost of the deployment. The right part of the figure shows the components to be deployed. As can be seen there are three database servers: one for each of the two countries, and one global integrated database. The right part of the figure shows the resulting deployment. As it can be seen the data has been partitioned, and the local database servers that contain personal data have been secured by deploying them in a virtualized private cloud.

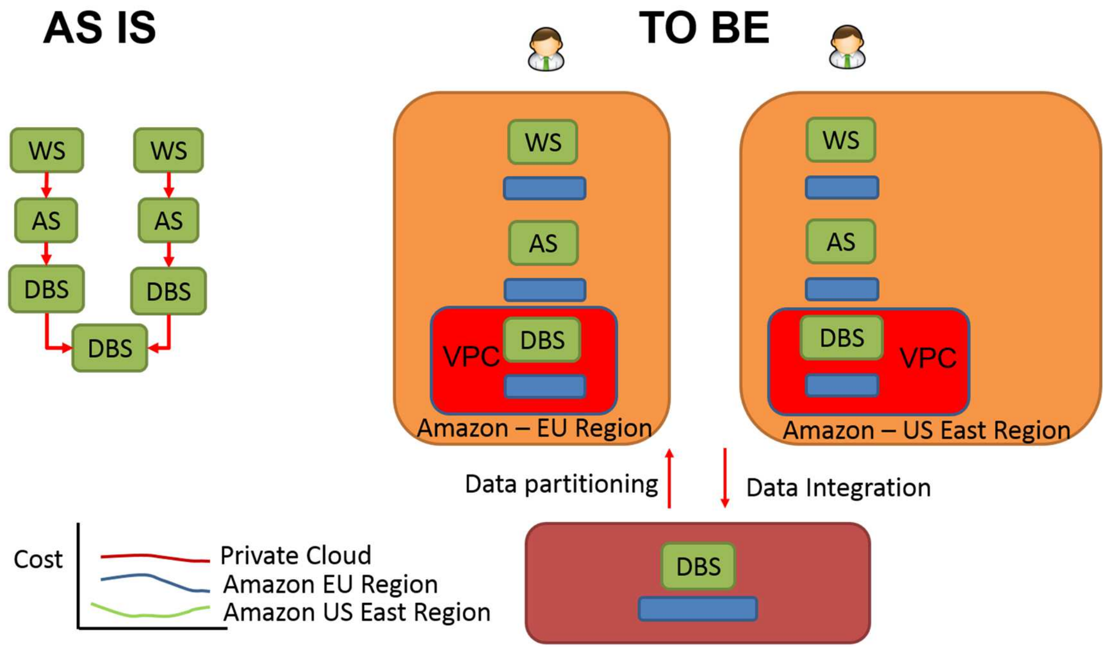


Figure 9.5: Add Cloud Provider with Data Partitioning

9.6 Company Collaboration via Public Cloud

Use Case	**Company collaboration (or merger) via public cloud**
Description	Company A collaborates with a new company, e.g. in a supply chain, via a database in a shared public cloud
Prerequisites (Dependencies) & Assumptions	Two companies are involved in this use case. Each company has its own IT infrastructure. The two companies decide to collaborate in the context of a supply chain. It is decided that the best way to collaborate is to create a common database links the data from both companies about product and services. It is decided that the shared database will be deployed in a public cloud. A database schema is defined by both companies to support the supply chain. Both companies also agree how their internal IT systems will update and read data from the database.
Steps	1. The system administrators of the two companies update their IT systems running in the internal cloud to work

	with the new common database. 2. A system administrator defines a CAMEL model for the database server and submits it to the PaaSage platform. 3. The Reasoner proposes a deployment model 4. The system administrator accepts the proposed deployment. 5. The Reasoner sends the deployment model to the adapter. 6. The Executionware deploys the database in the selected public cloud. 7. The Executionware then connects the database to the two enterprise IT systems.
Variations (optional)	
Quality Attributes	• Data security • Performance
Issues	

The figure below illustrates the above use case with a scenario. Company A merges (or decides to collaborate?) with company B to form an integrated supply chain. To support the merger it is decided to quickly create a shared database in a public cloud. Since the data is confidential it is deployed in the VPC of the Amazon EU region. The application must now serve many new users and must scale accordingly across the different databases, private and public clouds. The right side of the figure shows the components of the two applications and the shared database. The right side of the figure shows the resulting deployment. The common database is deployed in a virtual private cloud to ensure that data is protected. The data is only accessible by the IT systems from the two companies that have merged.

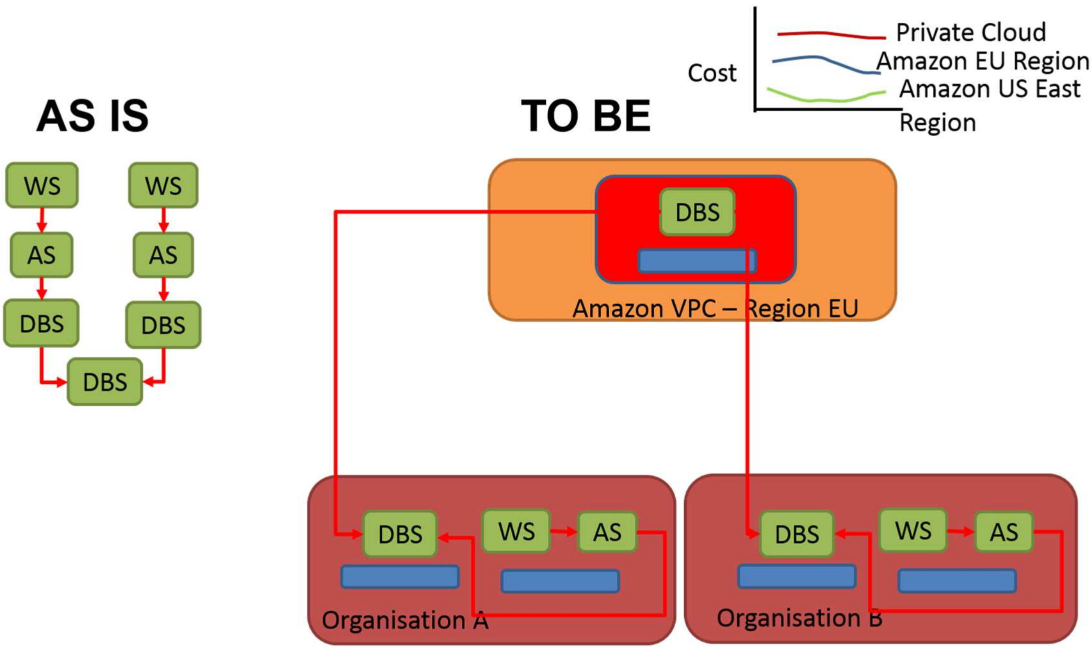


Figure 9.6: Company Collaboration via Public Cloud

10 Conclusion and Outlook

While the initial requirements deliverable (D6.1.1) has focused on the description of different case studies and provided initial requirements on cloud deployment scenarios and the PaaSage platform, this deliverable (D6.1.2) has extended the initial requirements with more precise requirements for the PaaSage platform.

Through a better understanding of the PaaSage platform, the cloud strategy reflecting the requirements of each partner has been revised and more precisely described.

Based on the cloud modelling language (CAMEL), the workflow and the role of each component of PaaSage, the case study partners are able to understand how to match and integrate their use cases into the PaaSage platform. Besides, the case study partners have defined their detailed use cases by describing required usage scenarios, which have been grouped and linked to the individual components. Through this exercise, the expected component behaviour is better identified. The use cases from the different case studies have been complemented by generic use cases. These generic use cases explore some potential behaviour of the PaaSage platform based on a consensus between case study partners and technical partners. Furthermore, they have explored different functional and non-functional requirements for the PaaSage platform e.g., by defining deployment plans in order to minimize the deployment costs under constraints of availability or security. Some areas of the PaaSage workflow such as run-time adaptation and design-time adaptation of deployment plans are well identified; But for sake of brevity, only briefly covered here by the generic use cases.

As the first version of PaaSage platform becomes available, the case study partners will be able to experiment with and deploy their case studies into the PaaSage platform. This will, in turn, lead to a refined understanding of the expected behaviour and requirements. These requirements will be managed internally within the project, even though this deliverable is the final requirements deliverable.