**ARTIST**

**FP7-317859**



***Advanced software-based seRvice provisioning and migraTIon of legacy Software***

**Working Document**

**Script for ARTIST Review 1**

|  |  |
| --- | --- |
| **Editor(s):** | Leire Orue-Echevarria |
| **Responsible Partner:** | TECNALIA |
| **Status-Version:** | Final – v1.0 |
| **Date:** | 18/10/2013 |
| **Distribution level (CO, PU):** | CO |

|  |  |
| --- | --- |
| **Project Number:** | FP7-317859 |
| **Project Title:** | ARTIST |

|  |  |
| --- | --- |
| **Title of Deliverable:** |  |
| **Due Date of Delivery to the EC:** |  |

|  |  |
| --- | --- |
| **Workpackage responsible for the Deliverable:** |  |
| **Editor(s):** | Leire Orue-Echevarria (TECNALIA) |
| **Contributor(s):** | All partners |
| **Reviewer(s):** |  |
| **Approved by:** |  |
| **Recommended/mandatory readers:** | All partners |

|  |  |
| --- | --- |
| **Abstract:** |  |
| **Keyword List:** |  |
| **Licensing information:** |  |

**Document Description**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Date* | *Modifications**Introduced* | |
| *Modification Reason* | *Modified by* |
| V0.1 | 18.10.13 | First draft version | TECNALIA |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Table of Contents 4](#_Toc370197755)

[Table of Figures 4](#_Toc370197756)

[Table of Tables 4](#_Toc370197757)

[Terms and abbreviations 5](#_Toc370197758)

[1 Java PetStore Scenario 7](#_Toc370197759)

[1.1 Business case definition motivation 7](#_Toc370197760)

[1.2 PetStore Assessment Process 8](#_Toc370197761)

[1.3 Design and Implementation Overview 8](#_Toc370197762)

[1.4 Pre-migration phase 8](#_Toc370197763)

[1.5 Migration phase 9](#_Toc370197764)

[1.5.1.1 Model discovery and understanding (reverse engineering) 9](#_Toc370197765)

[1.5.1.2 Target environment specification 9](#_Toc370197766)

[1.5.1.3 Modernization-Technical Tasks 11](#_Toc370197767)

[1.5.1.4 Modernization-Process Related Tasks 13](#_Toc370197768)

[1.5.1.5 Modernization-Business Model Related Tasks 14](#_Toc370197769)

[1.6 Post-migration Tasks 14](#_Toc370197770)

[1.6.1.1 Evolution Related Tasks 16](#_Toc370197771)

[1.7 Summary of videos and responsible partner 17](#_Toc370197772)

[1.8 References 18](#_Toc370197773)

# Table of Figures

[Figure 1: Target Environment Specification Actions 10](#_Toc369871206)

[Figure 2: PetStore Post-migration Tasks 15](#_Toc369871207)

[Figure 3: PetStore Post-migration Tasks 15](#_Toc369871208)

[Figure 4: PetStore Verification Test Results 16](#_Toc369871209)

[Figure 5: PetStore Verification Test Results 16](#_Toc369871210)

# Table of Tables

**No se encuentran elementos de tabla de ilustraciones.**

# Terms and abbreviations

|  |  |
| --- | --- |
| BFT | Business Feasibility Tool |
| BPMN | Business Process Model and Notation |
| BT | Benchmarking Tool |
| CAML | Cloud Application Modelling Language |
| CEO | Chief Executive Officer |
| DAO | Data Access Object |
| DT | Deployment Tool |
| EPF | Eclipse Process Framework |
| EUbBET | End-user based Behavioural Equivalence Tool |
| HRM | Human Resource Manager |
| IDE | Integrated Development Environment |
| M2M | Model to Model |
| M2T | Model to Text |
| MAT | Maturity Assessment Tool |
| MDFE | Model Driven Forward Engineering |
| MDRE | Model Driven Reverse Engineering |
| MDT | Model Discovery Tool |
| MPT | Methodology Process Tool |
| MUT | Model Understanding Tool |
| NFR | Non-functional Requirement |
| NFRVT | NFR Verification Tool |
| OMG | Object Management Group |
| ORM | Object Role Modelling |
| OT | Optimization Tool |
| PDM | Platform Domain Models |
| PIM | Platform Independent Model |
| PSCT | Performance Stereotype Classification Tool |
| PSM | Platform Specific Model |
| PT | Profiling Tool |
| RDBMS | Relational Database Management System |
| RST | Requirements Specification Tool |
| SbSp | Service based Software providers |
| SbSpCT | SbSp Certification Tool |
| SotA | State of the Art |
| SPEM | Software Process Engineering Metamodel |
| SQL | Structured Query Language |
| TCbBET | Test-case based Behavioural Equivalence Tool |
| TFT | Technical Feasibility Tool |
| TGT | Target Generation Tool |
| UML | Unified Modelling Language |
| V&V | Validation & Verification |
| WP | Work Package |

# Java PetStore Scenario

The Java Pet Store is a popular J2EE example application in the [J2EE Blueprints series](http://java.sun.com/blueprints/enterprise/index.html), created by Sun Microsystems [1]. It models an e-commerce application where customers can purchase pets online using a Web browser. The purpose of this application is to demonstrate the capabilities of the J2EE platform and it is written for learning purposes (for example to illustrate the many different design patterns on various parts of J2EE).

PetStore is being used in ARTIST as a first simple use case to perform initial proof of concepts of the tools, techniques and approaches being developed in the context of the project.

## Business case definition motivation

The Pet Store example is a set of source code and documentation that can be downloaded from 02]. ARTIST aims to tackle the problem of software modernization under two perspectives, the technical one and the business one. That is the reason why **we need to define the business case of the Pet Store in order to evaluate and test all the tools under both perspectives**.

***PetStore Business Case***

SME-Software is an English company producing and marketing software products targeting SME Enterprises. Their customers include Hotels, Restaurants, Stores, Chartered Accounting Firms, Employment Law specialists.

Its PetStore application started as a management system or ERP for stores selling pets, and evolved as a web portal for selling pets over Internet.

Nowadays SME-Software sells its application by licenses (to be renewed each year), and offers the possibility to customize it through consultancy services via ad-hoc projects. This customization includes the purchase of the new infrastructure (Application server, database server, etc.), installation of the adequate application server and database server software versions to fit the PetStore requirements, and customization of the interface layer.

The customers may contract these services to companies other than SME-Software.

Optionally, customers can contract maintenance support.

With the emergence of cloud computing and the SaaS business model, SME-Software managers have started thinking about offering their Software as a Service. Many customers have shown their willingness to contract the service instead of buying the product and the infrastructure required to install it.

SME-Software´s CIO is aware of the emergence of Cloud computing and he has participated in some talks, where the advantages of Cloud Computing have been exposed. He notices that their customers are demanding changes in the way they sell and offer their product but he is still reluctant to change because:

* He doesn’t know which implications the change from SaaP (Software as a Product) to SaaS can have.
* He doesn’t know how much time, and costs will imply a potential migration.
* He wants to maintain selling licenses as the same time.
* He doesn’t know which IaaS provider to choose, and which implications this could have, although Google App Engine seems to be his first option.
* He doesn´t know which risks (market and technical) are associated to this change.
* He doesn´t know which business processes are affected and how to adapt them to the new selling model

## PetStore Assessment Process

The Methodology Process Tool (MPT) tool will personalize the ARTIST methodology for each migration project relying on the results obtained in the pre-migration phase by the different tools (MAT, TFT and BFT). From MAT, migration goals and high level recommendations will be used. From TFT, the list of migration tasks and affected components will be used. Finally, from BFT, the selected business scenario, an initial cost-benefit analysis and the set of processes that may be impacted in his organization. Combining all this information MPT will personalize the general ARTIST Methodology for each specific migration project.

## Design and Implementation Overview

Once the results coming from the pre-migration phase are available, MPT will personalize the ARTIST Methodology. The overall ARTIST Methodology serves as a template for the instantiated methodology for a concrete migration project.

For the proof of concept and considering the results from the pre-migration phase the ARTIST methodology for the Java PetStore migration process is described in the following sections.

## Pre-migration phase

Video 1: WP6 – Generic Methododology Video (EPF Plugin)

Pre-migration phase is the trigger for starting (or neglecting) the migration. For that reason, following ARTIST approach, **the pre-migration phase will be always performed entirely as it is defined in the ARTIST general methodology**.

In the Java PetStore Scenario all the pre-migration phase tasks have been performed used the ARTIST tools (MAT, TFT and BFT) and obtaining the results that will conduct the methodology personalization process.

**TECHEVA.Petstore, BUSEVA.PetStore**: Answer online questionnaires through MAT to get the positioning of the PetStore and the list of Migration Goals, and high level recommendations.

Video 2: WP5 - MAT (TECNALIA)

**TECHFEAS.PetStore**: Use Eclipse TFT perspective and load the PetStore application, which will be inspected by TFT, getting the component model and the results of the technical feasibility analysis consisting on the component complexity list and the breakdown of migration tasks with their corresponding effort estimation.

Video 3: WP5 - TFT DEMO VIDEO (ATOS and TECNALIA Components)

**BUSFEAS.PetStore:** Use BFT to explore different possible alternative business scenarios that may emerge from the choices made by actors (including SME Software Co) in the model.

Video 4: WP5 - BFT DEMO VIDEO (ENG and TECNALIA Components)

**MIGRAT.PetStore**: With all the information compiled through the initial evaluation and the feasibility analysis SME Software Co has to decide whether to proceed with the migration project or not.

In this exercise we assume that SME Software Co decides to go ahead with the migration and to Google App engine.

## Migration phase

#### Model discovery and understanding (reverse engineering)

The migration phase starts with the model discovery and understanding of the application. More specifically:

**MODELDISCO.PetStore.Webapp**: Use the components (i.e., model discoverers) from the ARTIST Model Discovery Toolbox to produce the required initial models out of the different artefacts composing the PetStore application. These artefacts include the Java source code of the application of course, but also relevant configuration files or other related assets (XML-based). The generated models are both low-level technical models (Java code models, SharePoint Data models, KDM Code models or XML models) or directly obtained UML models (base Class Diagram models for describing the structural aspects or Activity Diagram/Sequence Diagram models for the behavioural ones, UML Profile model for representing an API/library).

Video 5: Java code model (INRIA)

Video 6: Generic UML class diagram (structural) – direct and via KDM (INRIA)

Video 7: Generic UML Class Diagram Discoverer (structural part of the code) from C# source code or assembly – direct (SPARX)

**MODELUNDERS.PetStore**: Use the components (e.g. model transformation chains) from the ARTIST Model Understanding Toolbox to produce the needed derived models from the previously discovered base models. These base models are thus deeper processed/analysed in order to extract the needed views for the remainder of the process (notably the forward engineering activities). The generated views (i.e. derived models) are for instance a UML Component Diagram showing the different layers of the applications (e.g. according to the MVC pattern), a stereotyped UML Class Diagram model annotated with persistence information or more-specific JEE views (for instance to identify the actual Java Entity Beans in the application).

Video 8: Domain Model Slicer (TUWIEN)

#### Target environment specification

The specification of target environments may aid in various phases of the ARTIST Methodology. Inclusion of cost aspects in the providers and services descriptions is necessary in order to be used for example in the pre-migration assessments, regarding the business feasibility of a migrated app. Furthermore, different approaches may be investigated based on the available services, their characteristics and nature of usage and the associated costs.

The following actions may be performed with relation to target specification:

* Acquisition of cost elements of services during pre-migration phase to compare different offerings and their anticipated cost of usage during application operation on the Cloud.
* Acquisition of benchmarking data regarding the specific application type for various types of services to compare the offered services performance (Benchmarking is performed offline through suitable tools offered by WP7).
* Investigation of metrics combining cost and performance.
* Investigation of service offerings terms (such as SLA guarantees, offering limitations etc.) through the provided descriptions.

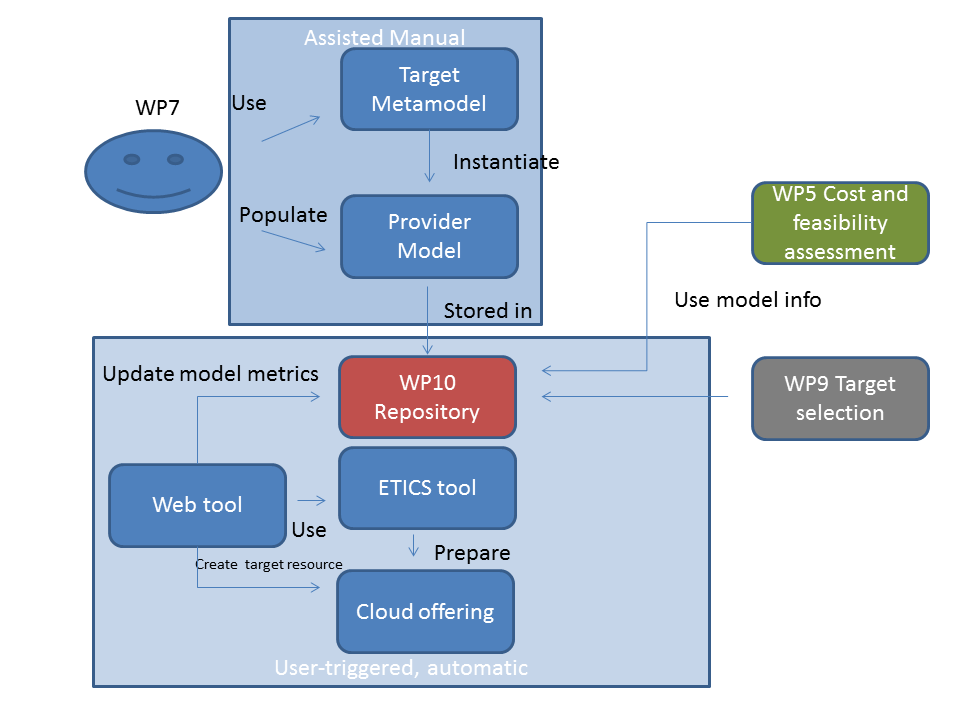


Figure 1: Target Environment Specification Actions

In the specific case, since GAE was preselected, the conceptual steps are the following:

* **TEBENCH.GoogleAppEngine** (offline): this refers to benchmarking of GAE instance types in order to investigate different performance metrics and their relation to their offered cost. Benchmarking will be provided for a wide range of application types (such as databases, web servers, stream servers, scientific calculations etc.) and is performed offline in a repetitive way. Results are stored and used to populate the next task (TEMPOP.GoogleAppEngine) along with various other information based on the model template.
* **TEMPOP.GoogleAppEngine** (offline): this refers to using the model definition in the previous figure to create the descriptions for the GoogleAppEngine environment. This includes for example listing of the available types but also creation of more specific descriptions regarding SLA guarantees, cost aspects per usage, discount aspects etc. This information may also be used by SLAMGTPRO and PROVIDERMGTPRO, since it will also contain information on supporting services features such as available monitoring, billing etc.

Video 9: CloudML@ARTIST (TECNALIA)

* **APACLAS.Petstore**: in this task a suitable application category for the specific app will be selected, given that the developer has knowledge about the internal behaviour of the app code in terms of performance. The application categories correspond to the different benchmarks used in the TEBENCH task. Following the selection, a comparison may be made between various available services based on the benchmarking metrics. A more automated process may be combined with the following task.
* **APAPUP.Petstore**: in this task it will be investigated whether the performance behaviour of the specific application may be automatically classified in the aforementioned application types. This will be helpful in cases where the developer may not have knowledge of the application behaviour (mainly in cases where there are combined types and the bottleneck is difficult to investigate) or external libraries are used. This task feeds the base information to the APACLAS.Petstore, for the case that the latter is not user-defined but based on artificial intelligence techniques.
* Video 10: Benchmark tool -- installer + video highlighting the PetStore profiling (ENG and ICCS)

#### Modernization-Technical Tasks

The PetStore modernization includes the following technical tasks:

**EMREQ.PetStore:** Annotate PIMs with GAE requirements. In order to start the forward engineering activities, the requirements that should be fulfilled have to be defined precisely on the technical level. These technical requirements may be derived by revising and refining the business goals by breaking them down to concrete metrics and measurements. For the task of specifying the requirements, a specific language is needed that is able to annotate UML models discovered in the reverse engineering process. Thus, a UML profile has to be defined in this respect that fits into the general methodology of the *Cloud Application Modelling Language* (CAML). In particular, if component-based views are used such annotations may be beneficial to guide further architecture restructuring or clustering components and bind them to specific Cloud environment.

For instance, such a requirement annotation for a component could be to annotate the back-end component of the PetStore with the requirement stereotype «highly scalable database». Subsequently, this stereotype application could be further exploited in the forward engineering process to modernize towards a NoSQL database.

The tool support should provide a UML modelling environment with specific support for the *Cloud* *Requirements Profile* and specific application and validation support in order to check that the annotations are properly defined. This latter aspect is of particular importance to end up with high quality systems after the code generation phase.

Video 11: CAML (TUWIEN)

**OPTAPPFEA.PetStore:** After having specified the requirements to be fulfilled by the application to be migrated, transformations may take care of optimizing the system in order to meet these stated requirements. While some of these transformations may operate on a more platform independent level, others may consider also platform specific information by transforming the application to certain Cloud environments for actually realizing the desired functionality with the required quality.

For this task, first a set of transformations is needed to explore the deployment space by finding appropriate Cloud providers that seem promising to fulfil the stated requirements, and second, transformations are needed as well to adapt the system in order to fully operational in the Cloud environments. The latter step is needed due to certain restrictions of Cloud environments for gaining benefits such as high scalability.

For instance, for the PetStore example we have assumed in the previous step to have the requirement of having a highly scalable database. Thus, at least the following steps have to be supported by the set of transformations:

* Data schema refactoring and optimization to target NoSQL persistence frameworks
* Data dump transformation to fit the data into the target NoSQL persistence frameworks
* Persistence layer adaptation based on the selected target NoSQL persistence frameworks

The tool support should provide a collection of transformations that are automatically selected and trigged by a transformation orchestration engine that has to decide based on the stated requirements and model structures how to proceed in the forward engineering step. Furthermore, tracing between the platform independent models and produced platform specific models has to be supported out-of-the-box.

Video 12: M2M Transformations - Objectify transformation (ATOS)

**GSCODE.PetStore:** Having the optimized versions in terms of platform specific models is the basis for the code generation that is concerned with producing the code to be executed in the Cloud environments. Thus, the supported programming languages of the Cloud environment have to be used as generation target as well as available libraries and APIs should be also exploited in the code generation process as good as possible. While some details of the produced code may be produced by following conventions, other features may be configured in the platform specific models in order to fully exploit the potential of Cloud environments.

This task has to be supported by a set of code generation templates implemented as Model-to-Text transformations. By these transformations, the best practices how to realize certain functionality in Cloud environments is formalized and reused to produce high-quality code by following certain patterns and code styles in a consistent manner.

An example in the PetStore scenario would be to generate Java code out of the platform specific models as well as XML configuration files, the database definitions as well as a database dump by using a serialization format that can be imported in the Cloud databases.

The tool support should provide a collection of code generation templates for a selected set of Cloud providers out-of-the-box and extension points for producing new templates for emerging Cloud providers. Furthermore, core code generation templates should be provided that can be shared between different Cloud providers, e.g., that use Java as programming language. However, these templates should be easily customizable to be specialized for the peculiarities of specific Cloud providers, e.g. to exploit certain APIs for optimized caching.

Video 13: M2T Transformation (TUWIEN)

**GDEPL.PetStore:** Having all artefacts required to run the migrated applications in Cloud environments, the next step is to bundle and transfer them to the Cloud environments. As a prerequisite, the Cloud environment has to be set up to run the application, e.g. set up the computation nodes, databases, firewalls and so in case one is going for Infrastructure as a Service (IaaS) providers. This activity is normally referred to as provisioning and has to be achieved before the application can be actually executed in the target environments.

This task has to be supported by a provisioning engine and by a deployment engine that are inspired by current build process management tools. A deployment language as well as a provisioning language have to be developed that allow to describe the needed steps to get the application running in a certain Cloud environment. These specifications have to be fully executable by the deployment and provisioning engines. Finally, from a given platform specific model, a default deployment and provisioning specification has to be derivable by a transformation that infer a default deployment and provisioning strategy based on the given structures and dependencies in the models.

An example in the PetStore scenario where this support is need is the deployment of the generated artefacts in the Amazon platform. Before the application can be executed, the needed execution environment has to be set up and configured properly. Finally, the artefacts have to be deployed in the execution environments and the data needed for running the application in the new context has to be injected.

The tool support has to provide modelling editors to describe deployments and provisions in the to-be-developed modelling language. Furthermore, transformations have to be provided to produce default specifications for these two concerns out of platform specific models. Finally, the deployment and provisioning engines have to be integrated in this tool to automatize the deployment and provisioning tasks.

#### Modernization-Process Related Tasks

The PetStore modernization includes the following process related tasks:

**DEVPRO.PetStore:** As the SME Company is changing the software they sell (it will be sold as a service) they have to change the process to develop the software. The new application shall fulfil specific characteristics such as multitenancy, high availability and high scalability. For this purpose, Agile Software methodologies should be used (i.e. SCRUM).

**UPPRO.PetStore**: Following the new software model, frequent releases shall be introduced in the software update process.

**INCIMGTPRO.PetStore: PetStore** will be offered as a service, and due to this an incident management process has to be established. Incidence Management includes any event which disrupts, or which could disrupt, a service.

**SLAMGTPRO.PetStore**: Every service shall be accompanied by an SLA. SME Software shall define a proper process to establish an SLA management process including the definition and monitoring of those SLAs.

**HELPDESKPRO.PetStore**: As a service provider, SME Software has to define a unique point of contact between the company and the users of the service (PetStores). This will be determined by the HelpDesk process.

**MKTPRO.PetStore**: The asset that SME Software is going to sell will change from a product to a service so the marketing process will also change as the targeted market and market strategy will change.

**ACCOUNTPRO.PetStore**: SME Software will have to adapt the accountability process to a new service. The accountability of a service based on Cloud Computing is complex and several new key aspects have to be considered (incorporating new costs models and new pricing models).

**PROVIDERMNG.PetStore:** SME Software will have a new provider, the Cloud provider, so the provider’s management process shall be adapted including aspects specific to Cloud Providers.

**ROLESALIG.PetStore**: As new processes have been defined in SME Software, also new roles have to be included in the organisational chart.

Spoken, as part of the presentation

#### Modernization-Business Model Related Tasks

The PetStore modernization includes the following Business Model related tasks:

**MARKSEG.PetStore**: Market to be addressed by SME Software has to be redefined according to its new offer.

**VALPRO.PetStore**: The value proposition for the PetStore service has to be reformulated as the offering, the problems that the solution solves and the customers’ needs have changed.

**CHAN.PetStore**: The channels that SME Software uses for approaching the customers will change and should be redefined.

**CUSTREL.PetStore:** SME Software will not sell only a product; it will also offer a service, so the relationships with the customers will change. This business process should also be redefined.

**REVE**.**PetStore**: The price for the new service offering by Software Company should be defined. In this case SME Software will determine it by the use. For that, a monitoring component shall be incorporated to the application.

**KEYRES**.**PetStore**: As the value proposition of SME Software has incorporated a new asset, new resources (such as an IaaS provider) are required.

**KEYASSOC**.**PetStore**: The determination of new partners is required for the offering of the new PetStore service.

**COSTSTRUCT**.**PetStore**: New costs will be incorporated to the SME Software accountability, such as those related to the IaaS provider. This analysis should be performed in the scope of COSTRUCT task.

Spoken, as part of the presentation

## Post-migration Tasks

The post-migration tasks are split can be split into two categories: the validation of the migration goals, functional and non-functional, as well as the certification process.



Figure 2: PetStore Post-migration Tasks

**VERBETC.PetStore: Verification of the behavioural equivalence with test cases**

The first approach to validate behavioural equivalence uses test cases. If test cases are present, they can be migrated from the legacy PetStore application to the Amazon platform by using the *same tool chain* as the legacy PetStore application itself. If no test cases are present, they can be generated from the platform-specific model (*PSM*) and the platform-independent model (*PIM*) using an adapted version of the tool *FShell*.

Either way, the resulting test cases are executed in the development environment of the migrated system. Each test case yields a verdict: *pass* in case of success or *fail* otherwise. If any execution fails, a detailed report is sent back to the migration step to correct the respective error. Otherwise the next task is executed. For the user, a simple overview is generated, similar to the one in JUnit (see Figure 3).



Figure 3: PetStore Post-migration Tasks

**VERBEUBT.PetStore: Verification of the behavioural equivalence with end user-based tests**

After the behavioural equivalence has been validated using test cases without failing, end user-based tests are used. For this, both the legacy PetStore application as well as the migrated PetStore application need to run in parallel. Interactions of a real user with the migrated PetStore are captured and replayed on the legacy application using the tracing information from the complete migration tool chain, e.g., PIM, PSM, and model transformations. A dedicated capture/replay tool has to be implemented.

In case a mismatch between the behaviour of the migrated system and the legacy system is found, a report detailing the error is sent back to the migration tasks. This report can also be presented to the user.

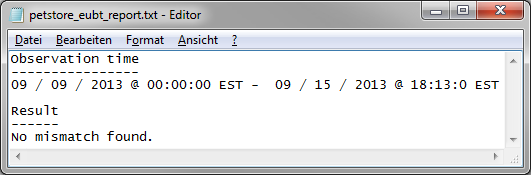


Figure 4: PetStore Verification Test Results

**VENOFR.PetStore: Verification of non-functional requirements**

The validation of the non-functional requirements take place when the validation of the functional requirements has been successful. The non-functional requirements are collected in a *goal model* produced during the migration assessment. Each requirement/goal defines a non-functional property (e.g., utilization of any component), a workload (e.g., 1000 simultaneous user login requests), and an achievement condition (e.g., < 0.75) and can be validated using a model-level approach and a code-level approach.

The model-level approach takes the *PIM* and the selected cloud provider and simulates the execution of the stated workload using our fUML-based analyser. The code-level approach monitors the migrated and deployed system and simulates the execution of the stated workload using an external tool such as Apache *JMeter*. The results are compared with the goal achievement condition set by the user and presented in HTML format.



Figure 5: PetStore Verification Test Results

Video 14: Verification of NFR (TUWIEN)

**CERTMIPROD.PetStore: Certification of the migrated product**

The certification of the migrated product is optional and is aimed at service providers such as the SME-Software company is now.

As the certification of the migrated product is independent from the actual migration and is done independently of the other tools. The certification will evaluate the business, the process, and the technology. A company can get certified by doing a self-evaluation, through submitting certification evidence to the certification authority and through an on-site evaluation by the certification authority. Three different certification levels can be achieved: bronze, silver, and gold.

Video 15: Certification model (TECNALIA) – TBD as it is not yet a prototype but an Excel-based application

#### Evolution Related Tasks

Spoken, as part of the presentation WP10 slides

## Summary of videos and responsible partner

Each video has to be maximum 2 minutes long and has to include text (Arial 18) with explanations. It must be playable by Windows Media Player (.avi format).

|  |  |  |
| --- | --- | --- |
| Video No. | Tool | Responsible partners |
| 1 | Generic Methodology Video (EPF Plugin) | ICCS |
| 2 | MAT | TECNALIA |
| 3 | TFT | ATOS and TECNALIA |
| 4 | BFT | ENG and TECNALIA |
| 5 | Java to code model | INRIA |
| 6 | Generic UML class diagram (structural) – direct and via KDM | INRIA |
| 7 | Generic UML Class Diagram Discoverer (structural part of the code) from C# source code or assembly – direct | SPARX |
| 8 | Domain Model Slicer | TUWien |
| 9 | CloudML@ARTIST | TECNALIA |
| 10 | Benchmark tool -- installer + video highlighting the PetStore profiling | ENG and ICCS |
| 11 | CAML | TUWien |
| 12 | M2M Transformation - Objectify transformation | ATOS |
| 13 | M2T Transformation | TUWien |
| 14 | Verification of NFR | TUWien |
| 15 | Certification model (TBD as it is not yet a prototype but an Excel-based application) | TECNALIA |

## References

1. Java Pet Store J2EE Experimental Application, <http://www.oracle.com/technetwork/articles/javaee/petstore-137013.html>

[2] PetStore J2EE Experimental Application: <http://www.mia-software.com/html/miaStudio/download/modisco/examples/javapetstore-2.0-ea5.zip>