
# R7.1: Engineering Services for Digital Preservation Requirements

WP 7 – Industrial Project 1: Engineering Services and Systems for Digital Preservation

Delivery Date: 15/10/2012

Dissemination Level: Restricted (Final version: Public)

TIMBUS is supported by the European Union
under the 7th Framework Programme
for research and technological development and demonstration activities (FP7/2007-2013)
under grant agreement no. 269940

| Deliverable Lead | | |
|---|---|---|
| **Name** | **Organisation** | **e-mail** |
| Ulrich Winkler | SAP | Ulrich.winkler@sap.com |

| Contributors | | |
|---|---|---|
| **Name** | **Organisation** | **e-mail** |
| Gregor Heinrich | iPharro | g.heinrich@ipharro.com |

| Internal Reviewer | | |
|---|---|---|
| **Name** | **Organisation** | **e-mail** |
| Michael Nolan | Intel | michael.nolan@intel.com |
| Wasif Gilani | SAP | wasif.gilani@sap.com |


| Document History | | | |
|---|---|---|---|
| **Version** | **Date** | **Author** | **Changes** |
| V1.0 | 30/08/2012 | Ulrich Winkler, Gregor Heinrich, Rene Cavet | Added content from working documents |
| V1.0 | 12/09/2012 | Mike Nolan, Wasif Gilani | Peer Review |
| V1.1 | | Gregor Heinrich | Merge of review and working notes |
| V1.2 | 20/09/2012 | Gregor Heinrich | Restructured document |
| V1.3 | 23/09/2012 | Gregor Heinrich | Added risks, DP motivations, rewrote scenario, added goals, refined requirements |
| V1.4 | 26/09/2012 | Gregor Heinrich, U. Winkler | Adjustments to requirements |
| V1.5 | 5/10/2012 | Gregor Heinrich, U. Winkler | Adjustments for re-review |
| V1.5 | 10/10/2012 | Mike Nolan, Wasif Gilani | Re-review |
| V1.6 | 12/10/2012 | Gregor Heinrich | Final adjustments |

# Disclaimer

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
| --- | --- |
| AdMon | Advertisement monitoring |
| AAM | Adverse Advertisement monitoring (process) |
| ASP | Application Service Provisioning |
| AVF | Adaptive Video Fingerprinting |
| BCM | Business continuity management |
| BP | Business process |
| CRM | Customer relationship management (also SAP software solution) |
| DP | Digital preservation |
| ERM | Enterprise Risk Management |
| ILM | Information lifecycle management |
| LCM | Legal case management (also SAP software solution) |
| LLM | Legacy lifecycle management |
| PaaS | Platform as a Service |
| PWA | Perpetual Web Archive (fictional company in scenario) |
| SaaS | Software as a Service |
| SDPB | Service-dependent Business Process |
| SLA | Service-level agreement |
| SOA | Service-oriented Architecture |
| TVCM | TV content monitoring |
| WP | Work Package |

# 1 Executive Summary

Software services are at the core of the TIMBUS project as in many scenarios they enable the execution of business processes using IT infrastructure. This document is concerned with the engineering processes and methods that software needs to be "preservable" when executed within service-oriented architecture (SOA). This will pose requirements to both a preservation system and the services themselves.

Starting from a definition of the problem space of software services and SOA, particular aspects are discussed like service-level agreements, the service lifecycle, risk management connected to this as well as particular risks that provision and consumption of software services entail. This treatment is completed by the particular motivations that digital preservation has in connection to the software service risks.

Based on this, a scenario is presented that consists of a inter-organisational network of contract relationships and service dependencies. Technically, the scenario focuses on monitoring of TV broadcasts to detect adverse advertising as well as business processes to properly handle such legal cases. The scenario includes core software services of the partners SAP and iPharro and reflects different business objectives that they deem relevant for exploitation of the TIMBUS results. Besides preservation and resurrection of distributed services in SOA, this includes enterprise risk management, proof of service level fulfilment, software escrow and development support using re-deployment of preserved services.

The scenario is described from a functional perspective and the roles of different organisations described in detail, including relevant details on the methods and service solutions that the partners SAP and iPharro integrate. Beside these companies, three additional organisations are part of the scenario that model real service providers, including one that offers "Preservation as a Service".

Given the scenario and the general considerations about SOA, requirements for "preservable" software services are derived, starting with an identification of stakeholders and use cases in the scenario. These are expected to overlap with a large variety of scenarios in SOA.

For the different use cases, 44 functional and non-functional requirements are identified and presented, separately for the preservation system and for the software services themselves. Many of these requirements are considered generic to all implementations of SOA-based preservation. A part of them, however, are extracted from a preliminary approach to preservation of services, an extension of SOA by a preservation aspect. This method, which is outlined in the document, proposes a message-passing algorithm that allows querying services and their dependencies recursively, dependent on particular preservation parameters like risk level and purpose of the preservation activity.

# 2 Introduction

The TIMBUS project focuses on resilient business processes. It will make the execution context, within which data is processed, analysed, transformed and rendered, accessible over long periods. Furthermore, continued accessibility is often considered as a set of activities carried out in the isolation of a single domain, for example within one organisation or enterprise. TIMBUS, however, considers the dependencies on third-party services, information and capabilities that will be necessary to validate digital information in a future usage context.

This document is concerned with the engineering processes and methods that software services need to enable digital preservation. We are specifically interested in engineering processes and methods related to distributed software services and Service-Oriented Architecture (SOA) in a multi-stakeholder Digital Preservation scenario. We want to study the implications on dependency relations between these services, as they change along their lifecycle, from the negotiation of a service agreement with DP support over the deployment up to the finalisation of the service operation.

Furthermore we want to study how Enterprise Risk Management can be applied in a distributed multi-stakeholder, multi-layered Service Oriented Architecture using Digital Preservation technology and methods.

This document will feed the requirements of the DP architecture and of the features of the software services themselves, as well as their inter-relations and dependencies.

## 2.1 Business processes and software services

Business processes as the target structures of the TIMBUS preservation approach may be described as structured collections of tasks that are performed to achieve a specific objective. Business processes in many cases are supported by IT software that implements particular tasks of the process or completely automates them. Software services as a special form and provisioning scheme of IT software may thus be seen as (partial) implementations of business processes, which may be then called service-dependent business processes (SDBPs).

Preservation of SDBPs will consequently require the preservation of the supporting software services and their dependencies.[1] In effect, preservation of SDBPs contributes to the main objective of TIMBUS to extend the notion of digital preservation from isolated data and information towards explicitly capturing and being able to reproduce the “context within which information can be accessed, properly rendered, validated and transformed into knowledge” (cf. abstract of the TIMBUS DoW): SDBPs are to be captured in connection with their service dependencies in a way that allows re-deployment of the complete actions of the business process with all IT software support.

[1] This corresponds to one of Timbus’s visions, “to preserve the functional and non-functional specifications of services and software, along with their dependencies”.

## 2.2 Goals of this document

This document is a report on the work in process in T7.1, which is being prepared upon request of the review panel 6 months early to the originally planned submission, addressing concerns raised over the progress of the three industrial projects. In this function, it contributes to the following goals that have been synthesised from the DoW and reviewer recommendations:

- Study what SAP and iPharro require to successfully engineer services for preservation (T7.1 objective), taking into account the need for preserving business processes that span over organisational and legislative boundaries (Review recommendation).
- Define the scenario with more detailed information about the business process DP scenarios, data used, and technical (functional and non-functional) requirements, including more technical details that are relevant for the TIMBUS DP system (Review recommendation).

These goals will be applied to a widened scenario compared to the one presented at M12. However, instead of the potential integration of a scenario from Intel, as suggested by the review panel, the extended scenario includes a process to develop software services and more complex service dependencies, e.g., an actual DP service provided within a Software-as-a-Service architecture. This document reporting on work in progress, the scenario will be further adjusted and widened until the official submission of D7.1.

## 2.3 Relation to other documents

In terms of input, this document uses many of the M12 deliverables, specifically concepts from D4.2, the draft scenarios in D4.5, as well as the initial architecture in D5.1 and D5.2. In turn, this document will contribute to a number of documents, among others requirements for software service interfaces in D7.2. Furthermore, this document will contribute to the architecture in D6.1, D6.2 and D6.5. In many respects, the document is similar to D8.1 and D9.1 in that it defines a scenario and extracts requirements for preservation.

## 2.4 Outline

This document is structured as follows:

- Chapter 3 gives an overview of software services, their lifecycle and service-level agreements. It will identify risks associated with provisioning services and will give an overview of how these risks can be dealt with using digital preservation, also considering Enterprise Risk in this context.
- Chapter 4 defines the media monitoring scenario that is being developed, starting with a motivation and overview and then introducing the service network and the different organisations involved. A description of the business processes involved concludes the chapter.
- Chapter 5 identifies goals for preservation connected to the scenario. Based on these goals the requirements for digital preservation are extracted.
- Chapter 5 concludes this report and provides an outlook on future work.

# 3 Software services, their risks and digital preservation

This chapter gives a generic view on software services and the risks associated with service-oriented consumption models. These risks motivate the preservation of services in the context of business processes as their mitigation or avoidance strategies, following the approach of TIMBUS to understand business process preservation as a method to curb risk in the enterprise.

The chapter views preservation of software services generically, while the subsequent chapters exemplify this with a particular scenario that captures a large set of aspects connected to the problem in a business process that may be actually demonstrated with preservation later in the project.

This chapter is structured into three sections: Section 3.1 reviews software services and the Software as a Service paradigm, in particular looking at the contractual side of software services, service level agreements (SLA), and outlining the software service lifecycle. Section 3.2 introduces Enterprise Risk Management (ERM) and Business Continuity Management (BCM) approaches, and Section 3.3 identifies the major risk factors associated with services along their lifecycle that can be managed with the ERM and BCM methods. Finally, Section 3.4 relates these risks to digital preservation, illustrating the need for preserving services along with their process contexts.

## 3.1 Software services and SaaS

Within the traditional enterprise software market, enterprise customers host their solutions themselves. Doing so, they spend money and resources setting up and keeping the complete environment running, taking all the risks arising themselves. Usually, customers buy software installations from the provider and then with an additional contract, the provider regularly sends legal updates to the customer and guarantees support.

**Application Service Provisioning** (ASP) is a complementary delivery model which is still common today. Here, service providers offered completely hosted applications or application suites. The objective was to move the burden of hosting and managing IT solutions away from the customers. However, the success of these approaches in particular for mission-critical enterprise software was limited for three main reasons:

- ASP did not provide sufficient flexibility; this results in long times for setup or change procedures
- Lack of transparency and dependability on the actual service level qualities. This was the result of non-existent or inadequate service level agreement contracts which are static documents and difficult to automatically assess using instrumentation/metrics. Intel and SAP have brought their knowledge of this subject from the SLA@SOI project[2] into TIMBUS.

[2] SLA@SOI (grant agreement number: FP7- 216556), http://sla-at-soi.eu/ .

- Lack of data control; the physical location of data, as well as certainty around its security are an obligation which is not easily outsourced to a third party. Ultimately, the primary data owner rather than the 3rd party provider, is responsible if data is lost or unauthorised access is detected.

**Service-Oriented Architecture** extends this by providing a way for service consumers, such as web-based applications, to become aware of available SOA-based services. By formalising many of the service-related functional and non-functional aspects and defining a well-defined interface to access them, the drawbacks of ASP could be compensated. SOA defines how to integrate widely disparate applications for a Web-based environment via "loose coupling" and uses multiple implementation platforms. Rather than defining APIs, SOA defines the interface in terms of protocols and functionality. Part of this is the definition of service level agreements (SLAs) that allow the consumer to exert control of the features of the service, such as security, performance levels and the functional specifications of the service.

One of the key trends in software markets is the shift to on-demand business based on the **Software-as-a-Service** (SaaS) delivery model. Customers increasingly buy software services that suit their business needs. They demand software that can be consumed in a fast and flexible manner, and where they do not need to take care about the ownership of the required IT resources and data management. Doing so, they rely on the availability and quality of these services for operating their own business. Hence, they require strong guarantees on the quality of service, including archiving and data retention services. Dependable service levels will become a major differentiator in the market of on-demand software solutions and the Internet of Services.

### 3.1.1 Software service level agreements

In a multi-layered SOA environment the exact conditions under which services are to be delivered can be formally specified by Service Level Agreements (SLAs). More information on the legal background of SLAs can be found in deliverable D4.4.

Nowadays, *Service Level Agreements* are elements for

- Specifying and agreeing the conditions under which services are delivered to customers, and
- Managing a service landscape in such a way that resources are efficiently used according to customer needs.

Current SLA models do not take DP of business critical information into account. The SLA@SOI project has, however, resulted in automated extensions to standard SLAs that provide a manageability layer for Platform-as-a-Service (PaaS), among others, offering VMs with defined snapshot intervals and image retention periods. In the context of DP of business processes, this may serve as a basis to develop a control mechanism that allows DP to be included in SLAs and the corresponding service infrastructure.

Looking at the current trend from service-enabled applications to SaaS and Internet of Service scenarios, an enormous pressure for service providers to professionalise and automate the offering and management of their digital preservation services may be foreseen by introducing the notion of machine-readable, precise, and comprehensive DP SLAs in order to be competitive in upcoming service markets.

### 3.1.2 Service lifecycle

Software services generally go through a lifecycle, which reaches from the negotiation and development to their decommissioning. In particular:

- **Service Negotiation** is the capability of the Service Consumer to choose a Service Configuration from the Service Provider.
- **Service Provisioning** is the process of preparing and equipping a service according to Service Configuration/ SLAs to allow it to provide functionality to its users.
- **Service Operation** refers to the phase where the Service Consumer is using the provisioned services. The Service Consumer may obtain support as part of the SLA during this phase, which may be considered a separate process.
- **Service Modification** is the process to modify the parameters of the service, for instance to improve service levels.
- **Service Decommissioning** terminates and decommissions that service.

This "classical" lifecycle may be augmented by specific steps that consider the creation and re-creation of a software service:

- **Service Development**[3] comes into play if the negotiated service needs to be created or adjusted, both before provisioning and configuration adaptation. This is especially the case with custom software services that are "made to measure" to fulfil a negotiated SLA, similar to a requirements specification of conventional software.
- **Software Escrow**[4] comes into play as a means of business continuity management when the service provider cannot maintain and continue its service under the SLA. In an escrow mechanism, the service is deposited with a trusted escrow provider that can release the software to a third party that re-deploys the service given a contractually defined event (cf. deliverable D4.4). Escrow is a special step that goes beyond the service lifecycle because ownership of the services changes and the SLA is transferred to another provider.

An overview of the augmented lifecycle is given in Figure 1. Here the typical process steps are illustrated, given that the service may need to be designed or developed before provisioning it. When the service is operated, it includes support by the service provider. Also, the case of a service interruption due to business reasons is considered, which is resolved by software escrow and leads to an SLA with a new service provider.

[3] This corresponds to the Service Design process group in the ITIL best practices quasi-standard (cf. D4.2).

[4] Software escrow does not correspond to a step in a typical process because after the organisation ceases to function normally, the original process itself is not executed anymore. In Figure 1, this is represented by a backwards arrow for "SLA with new provider".

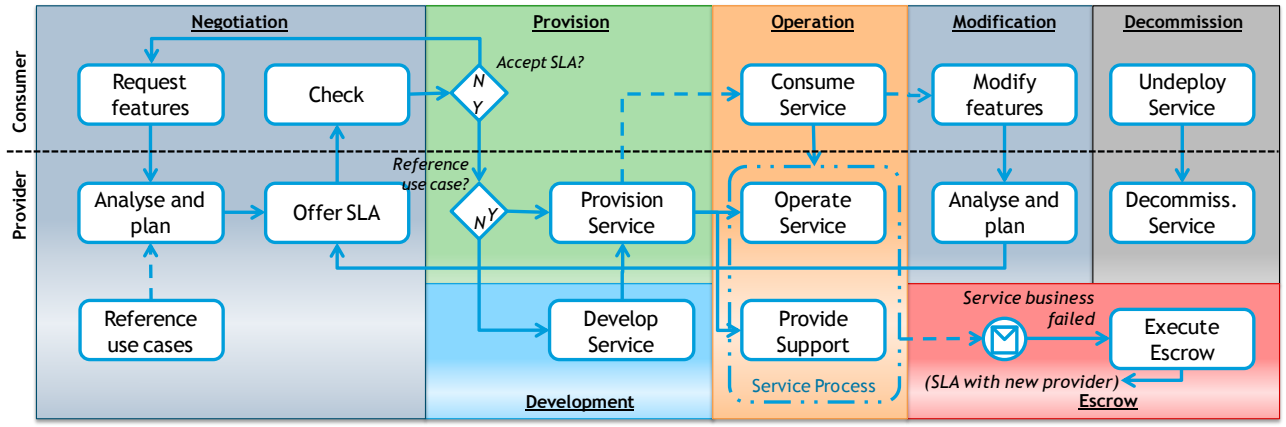


**Figure 1: Augmented service lifecycle.**

Throughout this augmented lifecycle, software services will need to be re-engineered to comply with the requirements of the preservation system and vice versa.

## 3.2 Enterprise Risk Management for Service Oriented Architectures

The focus of the TIMBUS project is the risk-aware digital preservation of business processes. Therefore risk management, and resilient business processes are a core part of the TIMBUS project.

Enterprise Risk Management is a holistic management process that identifies potential threats to an organisation and the impacts to business operations that those threats, if realised, might cause, and which provides a framework for building organisational resilience with the capability for an effective response that safeguards the interests of its key stakeholders, reputation, brand and value-creating activities.

Enterprise Risk Management comprises four groups of activities, which are (1) understanding the organisation, (2) determining risk strategies, (3) developing and implementing a risk response, and (4) exercising, maintaining and reviewing risk arrangements.

All four activities are organised by a fifth activity, the Enterprise Risk Program Management, which initiates risk related projects, assigns responsibilities, observes and manages activities, conducts training, and provides documentation.

### 3.2.1 Understanding the organisation

This activity aims to provide information that enables Risk Expert to (i) identify critical business processes, stakeholders, assets, resources and internal/external dependencies (ii) identify potential threats to critical business processes and (iii) assess and evaluate potential damages or losses that may be caused by a threat to critical business processes. Risk Experts refer to these activities as Business Impact Analysis (BIA) and Dependency Analysis/Risk Analysis (DA/RA).

### 3.2.2 Determine risk strategies

First, the Risk Expert has to understand, estimate, quantify and classify risk probabilities of threats. Second, he has to determine the acceptable minimum level of business process operations to mitigate the business impact. He also has to specify acceptable timeframes in which a normal level of operations has to be restored, such that the organisation can continue to deliver products and services.

### 3.2.3 Developing and implementing a risk response

The Risk Expert can put four responses to a risk in place: to remove a risk, to mitigate the adverse effects of a risk, to transfer the risk responsibility to third parties, or to do nothing at all and simply accept the risk.

To remove or to mitigate the Risk Expert may suggest changes to an existing Service Oriented IT landscape layouts, alter a business process arrangement or develop a recovery plan. The recovery plan details the steps to be taken to maintain or restore business operations to defined levels of operations within given timeframes. The Risk Expert (RE) has to verify that recovery plan is robust and does not depend on resources, which might be unavailable at the time the recovery plan is triggered.

If the RE decides to transfer the risk to third service parties or if a business process depends on external service providers in the first place, the RE needs to Risk related SLAs. The Risk Expert has to detail availability and recovery objectives encoded in a SLA and he has to specify penalties in case the service provider is not able to fulfil the agreements made.

The exact penalties should be derived from the Business Impact Analyses (BIA). The expert needs a methodology to translate business level DP requirements down to individual IT elements risk objectives. For example, he needs to translate the Maximal Tolerable Outage Time of a business process to Return Time Objectives of external services.

### 3.2.4 Exercising, maintaining and reviewing risks arrangements

These activities enable the organisation to demonstrate that risk arrangements are complete, coherent, current and correct. Exercising and reviewing helps Risk Experts to understand the organisation better and gives them opportunities to identify improvements in business recovery plans, continuity strategies and business impact analyses.

## 3.3 Risks associated with software services

In this Section, an overview of risks associated with software service operations is given. Applying digital preservation strategies may mitigate many of these risks.

### 3.3.1 Organisational and economic risks

Organisational and economic risks are related to the organisation providing or using a service and the stakeholders involved. These risks include:

- **R1.1: Loss of knowledge and expertise:** Key employees with fundamental expertise regarding business processes or service operation leave the organisation with insufficient knowledge transfer.
- **R1.2: Failure of service business:** The service provider faces financial shortages. This may affect service quality in various aspects: Payment of dependent service fees and software licenses, reduction of infrastructure quality used, loss of key knowledge in the workforce and finally failure of the whole company.

### 3.3.2 Legal and regulatory risks

Legal risks comprise risks according to legislation and/or contracts, of which SLAs are the most important type for software services. Legal risks include:

- **R2.1: Breach of Service Level Agreement:** The service may be claimed to create results that are deemed insufficient according to the SLA.
- **R2.2: Software license expiration:** Proprietary licenses in software dependencies may become legally problematic after the license expires, for instance if a license has been granted for a given period of time.
- **R2.3: Intellectual property rights infringement:** Intellectual property (IP) rights may be violated when the service uses patented or otherwise protected mechanisms without consent of the IP owner. Examples include IP of a service provider infringed by others (e.g., in competitive offerings) or vice versa IP infringement claims from alleged IP rights owners.
- **R2.4: Violation of legal regulations:** This includes the risk to violate data protection and privacy regulations. In regulated industries (finance, life sciences), this includes incompliance with the respective rules.
- **R2.5: Political risk:** The service may become impossible to operate because local legislation may require actions that are against the SLA terms. For instance, data protection of personal records may be at risk because law enforcement agencies require disclosure. This is especially true if a state becomes corrupted or instable.

### 3.3.3 Operational risks

Operational risks arise from running the service, which might become unavailable or violating the requirements. These risks affect the business of the service provider and consumer alike. Generally, operational risks may result from hardware, runtime environment, software or the people operating the services.

- **R3.1-3: Hardware/runtime environment/software failure:** A hardware or software asset becomes permanently or temporarily unavailable. Hardware/software failure may include the following cases.

  - **R3.1: Hardware/runtime environment/software incompatibility:** The hardware or software asset does not support the requirements of the service any more, for instance after version upgrades that change the behaviour of the asset.
  - **R3.2: Hardware failure:** The hardware ceases to function because of defects. This includes failures of the data center where the service is deployed.
  - **R3.3: Software failure:** The dependent software crashes or produces errors that lead to errors in the service, due to misconfiguration and/or bugs.
- **R3.4: Dependent service failure:** A software service that the BP depends on fails and becomes unavailable, affecting the service itself. In some cases, the dependent service failure may go unnoticed, e.g., as long as the service is not actively polled from the parent process.
- **R3.5: Insufficient service reliability:** The confidence in the result of a service may be insufficient, for instance if particular dependencies cease to work properly or if the service cannot provide the scalability needed to perform its requests.
- **R3.6: Loss of privacy and security breach:** The service may be subject to attacks that breach its security measures, or its data may lose its privacy due to errors or misconfiguration.
- **R3.7: Loss of data:** Data crucial for a service to run is lost, e.g., due to a hardware failure. This loss may be partial, which may result in loss of data integrity.
- **R3.8-11: Obsolescence:** Hardware, runtime environment or software are no longer available but needed for performing the service.
  - **R3.8: Hardware obsolescence:** The software of a service is targeted at specific hardware. For instance, a dependency of the service runs on a specific type of GPU architectures for massively parallel processing. As GPUs quickly evolve in architecture and capabilities, replacing such hardware after a failure may be difficult, and the software cannot run any more.
  - **R3.9: Runtime environment obsolescence:** The software requires a runtime environment that is no longer available. For instance, in a data center, the virtual machine for a web application is upgraded.
  - **R3.10: Software obsolescence:** The same is true for software that for instance works together with a dependent service. If the service is upgraded, the software depending on it may become obsolete. The software service can only continue to operate if it is migrated to work with the new version of the dependency.
  - **R3.11: Data format obsolescence:** The data formats of dependencies may become obsolete. If the service is used in the future, data needs to be migrated.

### 3.3.4 Development risks

Development risks are related to the creation and adaptation of a service according to the requirements of the SLA or their modification:

- **R4.1: Insufficient SLA specification:** The requirements are insufficiently specified in the SLA so the development effort builds the wrong software.
- **R4.2: Inefficient development:** The service may be too costly and/or time-demanding to develop because of technical or organizational issues. If development fails to provide a service in, the service may not be able to be provisioned according to the SLA.
- **R4.3: Ineffective development:** The development fails to achieve the software features specified in the SLA, either in functional (service features) or in non-functional requirements (service quality and scalability). If development fails to provide a service with insufficient features, the service may not be able to be provisioned according to the SLA.
- **R4.4: Inefficient test procedures:** Testing (especially integration) of a complex service may be difficult because it is too costly to deploy all dependencies of the service for simulation. Furthermore, the dependencies may not be realistically simulating the true context of a service.
- **R4.5: Insufficient service quality:** The service is not reliable because of instabilities or inefficiencies in the service implementation.
- **R4.6: Insufficient service documentation:** The service is insufficiently documented, on a user and a development level. Operating the service or reusing it for later development is difficult.

## 3.4 Service preservation as risk mitigation

According to the architecture adopted in TIMBUS (cf. architecture deliverables D5.1, D5.2 and D5.5), risks of the types stated in the Section 3.3 may be seen as drivers of digital preservation activities. In particular, while digital preservation works as a mitigation measure for many of the risks, there's much that can be gained from extending pure data preservation to capturing the context of the services in order to re-deploy the actual BP they implement. Concrete motivations for this context preservation are given by risk type:

- **Organisational risks R1.1-2:** Loss of knowledge due to key staff leaving the company (R1.1) or due to business failure (R1.2) is mitigated by preservation of the service process. Preserving it with its dependent business processes and other context like associated documentation allows new staff to take over more easily. In case of complete discontinuation of the service (e.g., triggering software escrow), the software can be directly re-deployed and the processes associated with service provisioning can be continued from the point where they have been stopped. For instance, service provision may include the provision of support, and process preservation may allow continuation of the support process, i.e., the process with its IT support system and the information how is it used, as well as in its instantiations, i.e., concrete support tickets and their history. Like the service

provision itself, these support process instances extend to both service consumer and provider organisations.

- **Legal risk R2.1:** A central problem in complex software service is to prove SLA fulfilment (R2.1). In many cases, it is not economical to continuously monitor the SLA. Claims of SLA violation can be answered by preserving the business process that was accepted by the service consumer to fulfil the SLA at provisioning time and re-deploy this preserved reference business process for testing with the data in question. By comparison, process/system modifications can be identified, for instance if the customer hosts a service itself and upgrades the database system, which breaks parts of the service and therefore voids the performance guarantee of the service. Also the data may be of a different quality than the originally agreed reference data.

  Other legal risks may in fact be also risks to service preservation, as the prolonged storage and re-deployment may have to deal with expired licenses (R2.2) or the breach of legal regulations (R2.4); cf. deliverable D4.4.

- **Operational risk R3.4:** While unexpected outages of software services due to hardware, runtime environment or software failures (R3.1-3) or data loss (R3.7) are not in scope of business process preservation (they may be easier mitigated or avoided by failover mechanisms and data redundancy), the risk of failing dependent services (R3.4) indeed is in scope of preservation techniques that offer capturing of the dependencies as context of a service. Furthermore, the various scenarios that software, platform or hardware obsolescence (R3.8-11) may occur during the lifetime of a service may be handled using digital preservation (and proper migration) of the service context.

- **Development risks R4.2-4:** Preservation of a software service along with its context translates into some advantages for the service development process. Especially in cases where services are adapted to particular business processes, preservation of context may help reduce development and testing efforts significantly if the context captures the resources needed for development. In particular, if the preserved context includes references to the source code of a service, new service customisation work may start from a reference service and re-use much of its existing re-deployed technology, bootstrapping development and subsequently adapting particular sub-services (mitigating inefficient and ineffective development risks R4.2-3). Using the re-deployed context allows to test the service with its dependencies, which otherwise may be costly (R4.4) and result in insufficient service quality (R4.5).[5] This specifically includes re-testing scenarios where a re-deployed system is tested against new SLA specifications before it is developed further and deployed for service provisioning. In addition, new service components may be tested against a complete set of re-deployed custom services, thus providing unprecedented regression test capabilities, again increasing testing efficiency and mitigating R4.4.

[5] Also, providing second and third-level support on the service is simplified for systems that run in specific environments not accessible to the service support team (at customer site, etc.).

# 4 The media monitoring scenario

Based on the generic view on software services and managing their inherent risks, this section defines an example scenario that is built around a service commonplace in the media industry, media monitoring. The scenario has covers many generic aspects of software services, which allows to establish a good basis to identify the requirements for DP of SOA-based business processes as well as requirements needed from software services themselves. However, as this report is an initial iteration of the respective deliverable, the scenario will be further extended to complete coverage.

In order to be able to focus on the actual aspects of DP of software services, the scenario is a semi-fictional business use case that has a similar structure as its typical real-world counterpart. Using partly fictional components in the scenario is due to three main reasons:

- Like many organisations, SAP and iPharro are not allowed to publish data from customers or partner organisations due to confidentiality and legal constraints. Therefore it is common practice within industrial organisations to develop and research prototypes using fictional scenarios.
- Using a complex service network would depend on additional organisations, such as service providers or particular businesses. For scope and cost reasons, these organisations cannot be made part of the actual scenario.
- Real-world business processes and IT service landscapes tend to be large and compound structures. For example, the documentation of SAP's Accounts Collectable procedure is more than 150 pages long. Using real business process models as underpinning examples would go beyond the limits of this document.

Before describing the scenario in more detail, in Section 3.1 we will review the goals pursued from the perspective of the project partners' businesses, which eventually motivate this study of preservable software services. Subsequently, we will introduce the scenario in Sections 3.2, its service network in Section 3.3 and the associated business processes in Section 3.4.

## 4.1 Motivation and goals of the scenario

From the perspective of the industrial partners in TIMBUS and in line with their exploitation strategies, the use case in WP7 targets at the following goals:

- SAP is the market and technology leader in business management software, solutions, services for improving business processes. A brief introduction into SAP's business offering is given in Section 4.3.2. SAP's business solutions are based on a platform, the SAP NetWeaver platform. SAP NetWeaver provides a module for archiving and preserving business related data and legacy execution environments. This module is called the *Information Lifecycle Management* (ILM) Module. A brief introduction of the ILM module is given in Section 4.3.2.3. The ILM is deeply integrated into the SAP NetWeaver Platform and therefore the ILM is not well suited to preserve

third party applications or remote services. Therefore, SAP is interested to study what is required to manage and execute preservation and resurrection of **distributed services and cross-functional business processes in a service-oriented IT landscape that comprises third-party services**.

- Key industries such as finance, gas, oil have the legal obligation to demonstrate sufficient operational and strategic risk management capabilities [Brown2005, BASEL2005]. Implementing enterprise risk management in a SOA/SaaS setting is not an easy task due distributed nature of a service-oriented architecture. Therefore SAP is interested to study what is required to perform **Enterprise Risk Management for DP in a distributed SOA/SaaS** scenario.
- iPharro provides solutions for TV content and advertisement monitoring and is interested in being able to secure itself against **claims of SLA violation**. iPharro is interested in the requirements necessary to prove service level parameters of its deployed services at a given point in time, mitigating risk R2.1 outlined above.
- iPharro considers a **software escrow** mechanism to increase the trust in the continuity of its service provision. This is a competitive advantage for a small business providing services especially to larger customers, mitigating risk R1.2 outlined above. For this, iPharro needs to understand the requirements necessary to extend its software services for software escrow, especially regarding the service operation and support processes.
- iPharro's solutions work in various IT information system and business process contexts, typically integrated from its product portfolio according to custom requirements. For solution development and improvement of the core components, iPharro wants to deploy its software services against particular IT infrastructure, simplifying complex **integration tests**, and re-test particular solutions from the past with new, mostly functional, service level requirements, thus mitigating **development** risks R4.2-4 outlined above. iPharro would like to study the requirements for such a process re-deployment and re-testing environment.

With respect to the scope, the scenario is designed to cover the most common aspects of software services, including their augmented lifecycle (considering development and escrow) as well as a inter-organisational relations spanned by service agreements (cf. Section 2.2).

## 4.2 Scenario overview

When a company advertises on TV, it is difficult to estimate the impact of this marketing instrument. It is not even easy to check whether the advertisements have been broadcast as agreed. TV advertisement monitoring is the process to control these uncertainties, and until recently, the majority of companies used human test viewers to detect advertisements on TV channels.

However, with the number of simultaneously available TV channels ever-increasing, the process of monitoring advertisements for products and brands is becoming not only more prone to human error but has become cost-prohibitive when done by human test viewers. Advertisement-focused TV media monitoring companies have resolved to look for ways how to automate and optimise advertisement

monitoring workflows in order to maximize gains and overall efficiency. They are increasingly using advanced information systems to support their services.

The scenario presented here features such service with an advanced information system. More specifically, we will introduce a network of services to support business processes across different companies involved in monitoring broadcast advertisements. An overview of the network is presented in Figure 2. As justified in above, we will create the scenario around the project participants SAP and iPharro but will work with fictional third-party companies to make it reasonably realistic.

**Advertisement monitoring.** The scenario considers how a media research company, *Jenson Media Research Inc.*, delivers monitoring services to its clients, which are typically *advertisement customers* of *broadcasters*. Jenson runs an SAP installation to manage its monitoring process, which in turn depends on the provision of automated media monitoring services from *iPharro GmbH*.[6]

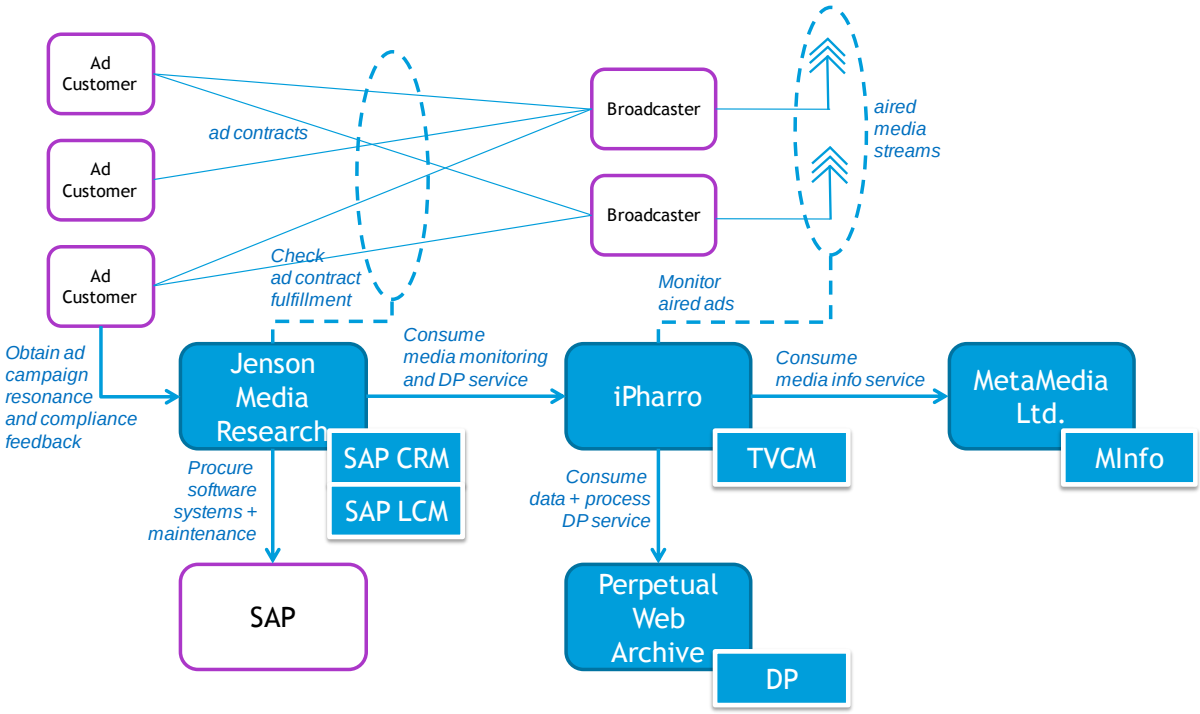


**Figure 2: Overview of the service network.**

In addition to providing statistics, Jenson offers the service of monitoring the advertisement broadcasts for compliance with general regulations and legal issues. In particular, if a client of Jenson is the subject of an adverse commercial, for instance denouncing its reputation, Jenson is entitled to pursue this as a legal case, a separate business process from the actual monitoring process.

[6] This scenario is easily extended to Internet-based advertisements. However, we re-enact a true collaboration between iPharro and the real-world counterpart of Jenson, a large international media research company.

**TVCM service.** The software service that iPharro offers to Jenson, TV content monitoring (TVCM), continuously monitors a large set of TV stations, reporting on all advertisements detected in conjunction with the programs they have occurred in, cf. Figure 3. The service also detects new advertisements and versions of known ones as well as occurrences of particular logos or texts. For instance, in Figure 4 different logos have been detected for an otherwise similar advertisement, which is achieved via comparisons on a subframe level. Such known advertisements are kept on record for later human assessment.

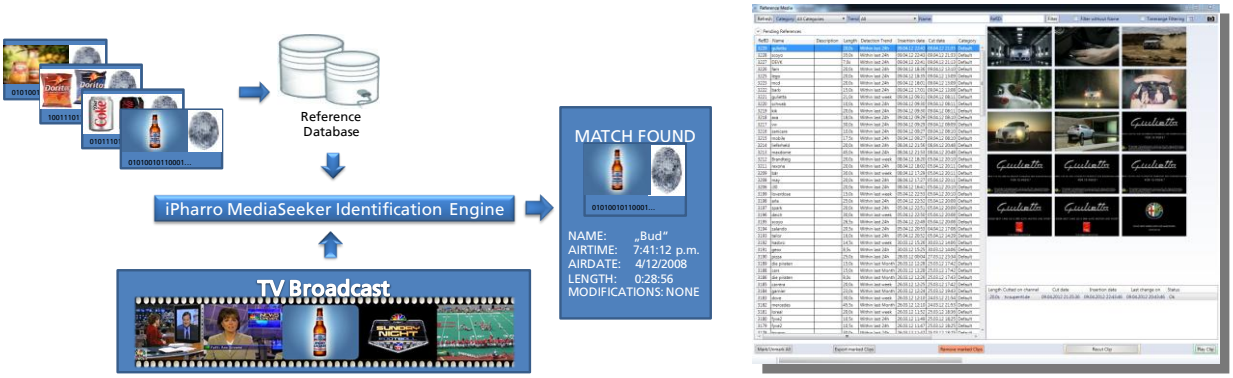


**Figure 3: TVCM system: Principle and user interface.**


**Figure 4: TVCM: Identifying versions via subframe differences.**

**Reporting.** In order to report the advertisement occurrences and their embedding into the TV programme, iPharro makes use of another third-party service provider, *MetaMedia Ltd.*, which provides a software service with structured meta-information on the actual TV programs. iPharro uses this for report generation and to automatically check advertisement placement, e.g., at the start of a football game as agreed between customer and broadcaster. The reports are transmitted to Jenson via a Web service interface. Using MetaMedia, in the scenario we can model an external service provider across corporate boundaries.

**Legal service.** Returning to Jenson and its legal service offering to counteract adverse commercials, the company's staff watches new occurrences of advertisements with the logos or mentionings of clients and checks them for compliance with ethical and legal guidelines. If they are identified to be indeed adverse, a legal case may be opened. In order to support this legal case with evidence, Jenson preserves the process information in the SAP Information Lifecycle Management module deployed at the company, including references to its context, in particular the advertisement in question from iPharro's monitoring service.

**Preserving legal process dependencies.** iPharro offers a DP service contract for its customers in order to "plug in" as a dependency into their own business process preservation solutions. Thus, the SAP system at Jenson can trigger preservation of particular media. To preserve digital objects, iPharro uses digital preservation services of another external provider, *Perpetual Web Archive Inc.* (PWA). PWA may be seen as an organisation that runs the IT solution resulting from the TIMBUS project, and in this scenario the idea of service-based preservation is considered.

**TVCM process preservation.** Also using PWA as DP provider, iPharro itself makes extensive use of digital preservation of its own business processes. This mitigates several risks, including the ones mentioned in Section 4.1: First of all, iPharro protects itself against SLA violation claims. Jenson may assert that iPharro had missed occurrences of advertisements or reported false positives beyond the agreed service level. For this purpose, iPharro preserves the complete distributed system that implements the custom TVCM business process that had been approved by the customer (here: Jenson) and thus is the basis of the SLA. In case Jenson makes assertions as above, the approved preserved reference system can be re-deployed and tested against the data in question.

With this process preservation use case, the actual goal of TIMBUS is explored. Furthermore, the idea of preservation as a service is introduced.

**TVCM escrow.** Preservation may be also used as a basis for software escrow, giving iPharro as a small company the competitive advantage of guaranteeing the availability of the service even after a potential business failure. This often is a prerequisite to get into contract relationships with large customer companies. Notably, in the case of escrow, beside the immediate context of the TVCM business process, the source code, design documents and maintenance processes should be preserved in order to preserve the knowledge associated with the service.

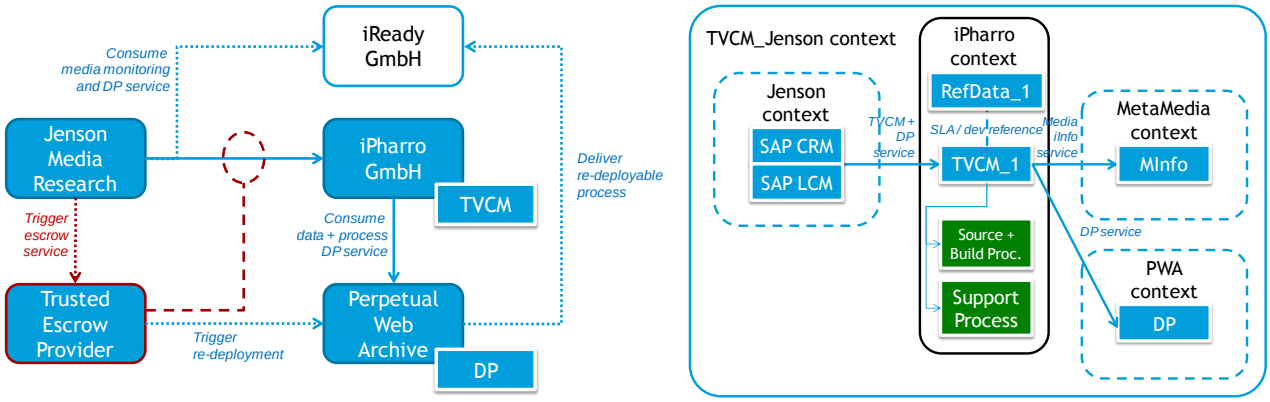


**Figure 5: Overview of the escrow scenario, preserved context.**

In Figure 5 left, the escrow mechanism is illustrated for the scenario at hand (cf. Figure 2). If the escrow conditions are met, e.g., iPharro goes out of business, Jenson draws its contractual option to trigger the escrow service, and the trusted escrow provider named in the contract chooses a third party, in Figure 5 iReady GmbH, to re-deploy the TVCM process and continue the service for Jenson. Re-deployment is triggered in the DP infrastructure at PWA, and the solution delivered to iReady. As a result, Jenson can

continue to consume the media monitoring service, and iReady takes over not only operations but also is enabled to continue software development based on the preserved source code as well as support process, as shown in the example context in Figure 5 right.

An interesting aspect in this context is what transformation the downstream service dependencies need to undergo to "survive" the escrow case. Should the escrow provide for service agreements with new companies in a form of contractual template? Or will the service itself be transferred to another service provider in the escrow case, thus having to migrate the interface and functional differences between the old and new meta-information service provider.

**Preservation for service development and SLA modification.** Another motivation to preserve source code and support is development support, mitigating development risks R4.2-4. As iPharro offers high-precision services (partly 99.9% recall service level for advertisement detection), the solutions are highly customised to particular instances of media analysis, often also integrating particular dependency services, such as meta-data generation as in the current scenario, but also automatic translation, logo detection and optical character recognition and specific interfaces that communicate with the consumer business process. For this, typically iPharro enters a solution development project with its customers that tailors the technology to the particular SLA requirements of the business process at hand. This is in line with the service lifecycle shown in Figure 1 in Section 3.1.2. To allow efficient development and integration testing support for its development team in such a custom solution business, iPharro has an interest to preserve the actual process instances along with the particular scenario they are embedded in, e.g., Jenson's particular business process (or its service interfaces). In a new development, the service may be re-deployed and the necessary changes implemented to comply with new requirements. Furthermore, it is possible to re-deploy the process to study the effect of new data in a preserved business context, or modify parts of the solution for improvements. In effect, such re-deployment or emulation of business process context may be seen as a solution to the integration test problem when new software components need to be tested against existing context. This often is a very expensive step in the software development process.

The preservation for service development has a high overlap with software escrow as it also aims at preserving the developed solution in context and with all source code. On first sight, its requirements may be seen weaker because (1) the knowledge on the particular past solution may be still in the company and (2) the tools and maintenance processes will be still available because the company continues to exist. However, as the evolution of high-performance software like the MediaSeeker platform progresses quickly, digital preservation and its migration and emulation capabilities will be highly relevant, even if the business case is re-used only after few years.

**Enterprise risk management:** By regulation, Jenson is required to run a risk management program. Jenson's business risk expert uses:

- *Business Impact Analysis* to identify critical business processes. One critical business process is the Legal Case Management process. Leal cases often have strict deadlines, for example to file cases or respond to claims.

- *Dependency and Risk Analysis* to identify resources supporting the Legal Case Management process. The risk manager identifies, that PWA is critical resource and uses the risk analysis to identify potential threads to Jenson's business if PWA services become unavailable or if PWA goes out of service.
- *Simulations* to simulate and evaluate risk responses and recovery strategies
- *Business Process recovery* to recover and restore preserved processes within given time frames

## 4.3 Organisations

In the following sections, we will introduce the organisations participating in the scenario in more detail. This also illustrates the background of the processes they implement and technology they use.

### 4.3.1 Jenson Media Research Inc.

The fictional Jenson Media Research Inc. is an independent media agency based in London. Its main business model is dedicated to monitoring advertisement effectiveness across different media channels, including TV and the Web. In short, Jenson's service answers for its clients the question, how much of the advertisement expenses are working and where exactly. This includes monitoring of contract compliance between advertisement customers and broadcasters who air advertisements.

Jenson uses iPharro TV content monitoring services (TVCM, introduced in Section 4.3.3) to monitor aired advertisements. In particular, Jenson is interested if an advertisement has been aired at the right time slot (e.g. when high audience numbers are watching, such as during the Champion's League Football Finals) and in the right quality (e.g. no speed up, no overlays, played in total, played in the right version). In addition, Jenson determines for its customers the reach of advertising campaigns (which consumer segments are actually seeing the ads), identifies current ad spend and channels, and analyses what competitors are doing compared to customers' own brands.

Furthermore, Jenson monitors legally relevant adverse advertisements for its customers. A typical case is if competitors air content on the Web or on TV channels that is objectionable and in some cases even a cause for legal action, such as some forms of comparative advertisement. Monitoring such ads is crucial under some legislation. For instance, if an advertisement in Germany is aired on TV for a given period of time, e.g., 21 days, public consent is assumed for the content, given that it fulfils basic ethical and legal rules. This is the reason why adverse commercials are often broadcasted at night-time or on channels with little audience. Using the TVCM monitoring service, Jenson can protect its clients against such de-facto consent because all new occurrences of advertisements can be easily assessed.

To manage its business processes, Jenson uses SAP software, such as Customer Relationship Management (CRM) and Legal Case Management (LCM). The LCM Process helps Jenson to handle complaints made by clients and to address issues where advertisements have not been aired according to the standards negotiated between advertisement clients and broadcasters.

**Legal Case Management process.** The term Legal Case Management (LCM) refers to law practice management and covers approaches and technologies used by law businesses to leverage knowledge and methodologies for managing the life cycle of a case more effectively. Figure 6, below, illustrates the primary steps in this process.


**Figure 6: Legal case management process.**

If a client files a complaint, a Jenson clerk opens a new legal case and initiates a *case file.* A case file is also created by the clerk if media monitoring reports provided by iPharro indicate potential issues. A case file is an electronic object that contains all relevant information related to a legal case. The case file itself serves also as a documentation or log; all activates related to a case (the process) are recorded in the case file itself.

In a next step, the clerk investigates this case. The clerk uses services provided by the iPharro GmbH to access and view advertisements, video footage and other reports.

In a following step, a Jenson lawyer evaluates the material collected by the clerk. As a result, the case is either rejected or pursued. If the Jenson lawyer has approved a case all stakeholders are informed and legal actions are initiated if required.

Once a legal case has been settled, the Jenson company is required to preserve all case-related material for the course of the legal proceedings and an additional longer term. The material includes video footage and other documents provided by iPharro. However, due to the typical volume of the data involved, it is not economical for Jenson to preserve the video case material provided by iPharro. Instead, Jenson sends a preservation request to iPharro and asks to preserve documents and procedures related to a case for a longer period.

### 4.3.2 SAP AG

In the scenario, SAP AG serves as a software provider to Jenson. This re-enacts the real-world case of SAP's business model: Consulting and integration of customised solutions that are based on its wide portfolio of ERP and other business applications.

The objective of this section is (1) to give overview of SAP ERP solutions, (2) a technology-driven insight on the SAP IT stack including SAP NetWeaver and (3) introducing the SAP Information Lifecycle Management solution. The SAP Information Lifecycle Management (ILM) module is SAP's current solution to digitally

preserve and manage business-relevant data as well as systems. Section 4.3.2.3 discusses the SAP ILM module and Section 4.3.2.4 explains how third party solutions can be integrated into SAP ILM.

#### 4.3.2.1 SAP Business Suite

SAP Business Suite solutions are used to coordinate all the enterprise wide resources, information, and activities in an effective and efficient manner required to complete business processes and to fulfil business targets. Nowadays, a business suite usually consists of a common database and a couple of tightly integrated applications tailored to the needs of various different departments. The common shared database allows every department of an enterprise to store and retrieve business process relevant information. A requirement is that this information should be processed in a real-time, easy to access, secure, reliable and consistent manner.

The SAP Business Suite is a comprehensive family of business applications, providing industry-specific processes for enterprises. SAP Business Suite applications enable enterprises to manage most critical business processes, such as ERP or CRM processes. They form a tightly integrated business application suite that adds value to an organization's value chain.

The SAP Business Suite may contain (but not limited to) the following process modules

- *SAP ERP* – business processes related to procurement, finance, human recourses, and many more.
- *SAP Customer Relationship Management* – business processes related to sales and marketing as well as **Legal Case Management** (described above).
- *SAP Product Lifecycle Management (PLM)* – processes related to product and portfolio management, manufacturing processes, and product data management
- *SAP Supply Chain Management (SCM)* – scheduling and planning processes, demand planning and forecasting processes, order operations, etc.

SAP Business Suite applications are based on the SAP NetWeaver platform – integration and application platform. This reduces total cost of ownership across the entire IT landscape and supports the evolution of SAP Business Suite applications to a services-based architecture.

#### 4.3.2.2 SAP NetWeaver

SAP NetWeaver is SAP's integrated technology computing platform and is the technical foundation for many SAP applications. SAP NetWeaver is a service-oriented application and integration platform. SAP NetWeaver provides the development and runtime environment for SAP applications and can be used for custom development and integration with other applications and systems.

Besides business process modules, such as ERP or CRM, SAP NetWeaver offers various foundation modules to address common aspects used by a plurality of modules. One module, the SAP Information Lifecycle Management modules, is used to preserve data and legacy system.

#### 4.3.2.3 SAP Information Lifecycle Management

The Information Lifecycle Management (ILM) module gives businesses control over what data a business has and where it is located. ILM allows businesses to automate the management of that data so that the business complies with internal, external, and legal requirements for its retention and destruction.

This entails knowing what kind of data a company deals with, where it resides, how long it must be retained, and when it may or even must be destroyed. Once a business knows the data, adequate tools and solutions to manage it are required: move data to the optimal storage location, store it, access it again (for example, during an audit), and destroy it once the data retention period has expired.

The ILM module is used by various business process application, such as ERP or CRM, to handle preservation needs. A large amount of information is created in application software and resides there for some time before it can be moved to archive systems. When data has been moved to storage, the business application – for example the Legal Case Management application - still "owns" and controls the data. So it is essential for applications and storage systems to communicate and be compatible.

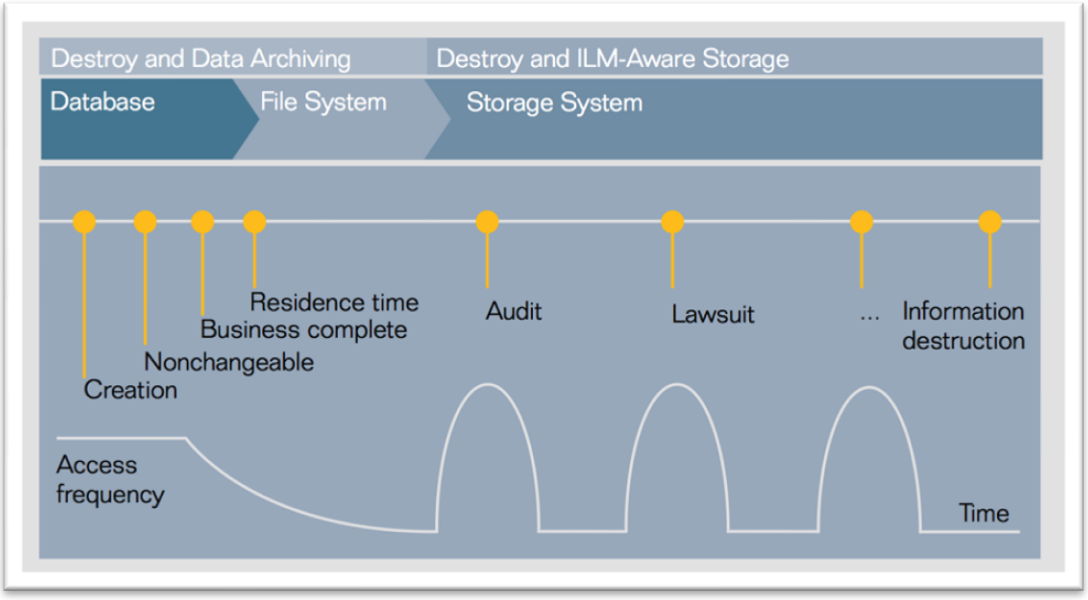


**Figure 7 Information Lifecycle and access frequency.**

Figure 5 depicts the Information Lifecycle and indicates access frequency as well as where data is stored.

ILM address more than just archiving. To this end, the ILM approach at SAP is built upon three cornerstones:

- *Data archiving and data management*, which focuses on data volume management. This is the traditional area of focus for information management and digital preservation.

- *Retention management*, which addresses the management of the lifecycle of data up to the point of destruction. It uses retention policy management and legal holds to help control the retention of data during its entire lifecycle. The ILM supports retention management using a rule driven approach.
- *System and process decommissioning*, which covers the decommissioning of legacy systems by moving data into a central ILM system called the retention warehouse. The retention warehouse is essentially a fully functional (decommissioned) SAP system, which allows to execute legacy processes and to render archived data.

#### 4.3.2.4 Integration of Third-Party Storage Providers into ILM

ILM offers various interfaces to integrate third party storage solutions and databases. One method to integrate third party storage solutions is based on WebDAV which enables the storage of archived data on an external storage system through the WebDAV protocol.

WebDAV, which is an extension to the HTTP protocol, stands for Web Distributed Authoring and Versioning and allows agents to perform remote Web content authoring operations. In addition to these authoring features WebDAV is particularly valuable for its communication protocol, which addresses several limitations of HTTP by providing capabilities, for example, for creating hierarchies (collections) and managing metadata (properties).

#### 4.3.2.5 Limitations of WebDAV in a SOA/SaaS Scenario.

WebDav is not very well suited for DP purposes in a Service Oriented Architecture setting. WebDav requires full read/write access to a (remote) system. A mounted WebDav directory appears just as a normal remote file system. In this scenario, we do not want to give the service consumer (Jenson Media Research) full read/write access to a remote file system provided by third parties. Furthermore, not all process related data are located in one place. As shown in Figure 2 process related data is created and stored at various places in this service-oriented IT landscape.

Moreover, even if Jenson Research is capable of accessing all process related data using WebDav, Jenson Media Research is not necessarily capable of rendering all these information, as Jenson does not possess the required software and computing equipment.

Not being able to access and render the required data (in future) is a potential risk for Jenson and therefore this risk needs to be identified, addressed and managed by the Jensons' Enterpries Risk Management program.

Therefore, SAP is interested to study:

- Engineering of services specifically designed for digital preservation
- Service Interface Standards for Digital Preservations
- Risk Management related to digital preservation in a SoA/SaaS environment

### 4.3.3 iPharro GmbH

iPharro GmbH is a service provider for digital asset management and specifically TV content monitoring (TVCM) based in Darmstadt, Germany. iPharro technology is a scalable content analysis platform that can be used to analyse and identify video clips by comparing them to a database of known video information. This allows iPharro to efficiently monitor a very large number of broadcast television channels in order to detect desired content.

Regarding digital preservation of business processes, the TVCM service is of special interest because it is a large, distributed software system with a service-oriented architecture, providing a good example for preservation of a system with Web service interfaces. In the following sections, we will give an introduction of the technology used for TVCM and will present the architecture of the system.

This section will introduce iPharro's technologies in more detail, giving background on the architecture to be included at the core of the scenario. Section 4.3.3.1 will introduce the idea of fingerprinting technology, Section 4.3.3.2 presents the service architecture that implements it, and Section 4.3.3.3 gives the background on the augmented service lifecycle, including the development processes to integrate iPharro's services into customer business processes.

#### 4.3.3.1 Adaptive video fingerprinting

iPharro's technologies are based on video fingerprinting, a software-based content identification technique used for a variety of tasks in broadcast and media industries. A fingerprint identifies a video when it appears in any media source. By extension, fingerprinting techniques may be as well used for content-based information retrieval (CBIR), then identifying a fingerprint in any object in a given database (the retrieval index), or logo detection, then applying the fingerprint comparison on sub-frame levels or under geometric transformations like perspective or projection.

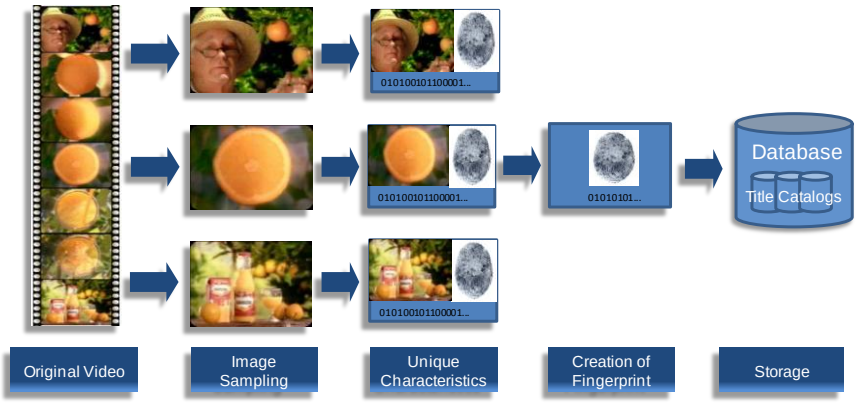


**Figure 8: Video fingerprinting.**

In Figure 8, the principle of determining fingerprints is illustrated. From an original video, images are sampled and unique characteristics extracted to create a concise representation of the content – the actual fingerprint. This fingerprint is stored for later use. Compared to alternative methods like manual annotation or digital watermarking, fingerprinting has the advantage of leaving the original data intact and being

powerful to identify even minor differences in the content. Furthermore, fingerprints may be kept during the complete media life cycle as they can be stored apart from the original content that may subject to additional processing that would for instance change watermarking information.

**MediaSeeker.** Because of its wide scope of applications, solutions to fingerprinting play out their advantages best if they can be adapted to particular requirements at customers' sites. iPharro therefore has developed a modular system that is based on various components and can be applied flexibly in various media environments – Adaptive Video Fingerprinting (AVF). This technology is the basis of the MediaSeeker platform, a suite of technology components that allow implementation of a wide range of media analysis and monitoring solutions.

In the first place, this adaptation allows adjustment of what may be called the "density" of video features in the fingerprint. To be light-weight processing, low density features will be used that are more coarse-grained than individual scenes, e.g., for applications like internet-scope search. Denser features are on frame level, e.g., for applications like TV programme monitoring, finding instances of a particular video in a set of TV programmes, or movie versioning, i.e., finding the locations where different versions of a film differ and how. Figure 9 gives an overview of the different tasks that may be performed using this technology and the different densities they have associated with them. In all of these cases, the AVF fingerprinting and querying technology of the MediaSeeker platform can be applied, for which different modifications are being introduced.

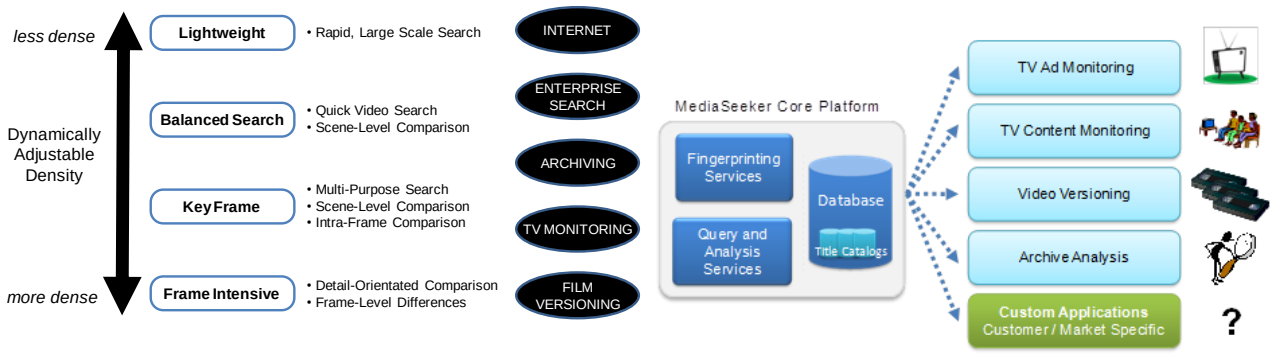


**Figure 9: Analysis tasks and fingerprint density.**

**Generic process embedding.** The AVF fingerprinting and querying approach may be seen as a more generic method to digital asset management. Based on the software services that may be implemented with the MediaSeeker platform, a range of business processes in the media economy may be supported with fingerprinting capabilities. These processes may be set along any step in the media asset lifecycle, from creation over post-production to consumption. This flexibility is shown in Figure 10, where FE = fingerprint extraction and DIS = Database Indexing Server. For instance, fingerprints may be used to identify intellectual property rights of audio and video files. For this, fingerprints of the original content are extracted right after production, and after post-production of the content (e.g., movie clip or news message), billing can be performed based on analysis of the content and direct association with the original content. Tracking the individual pieces of original content is largely simplified. More generally, being able to identify media content by a means independent of the actual media essence allows simplification of

metadata-aware media production and consumption workflows. Metadata can be stored, modified and retrieved at any independent database and at any step in the workflow, as long as the media asset is identified by a fingerprint.

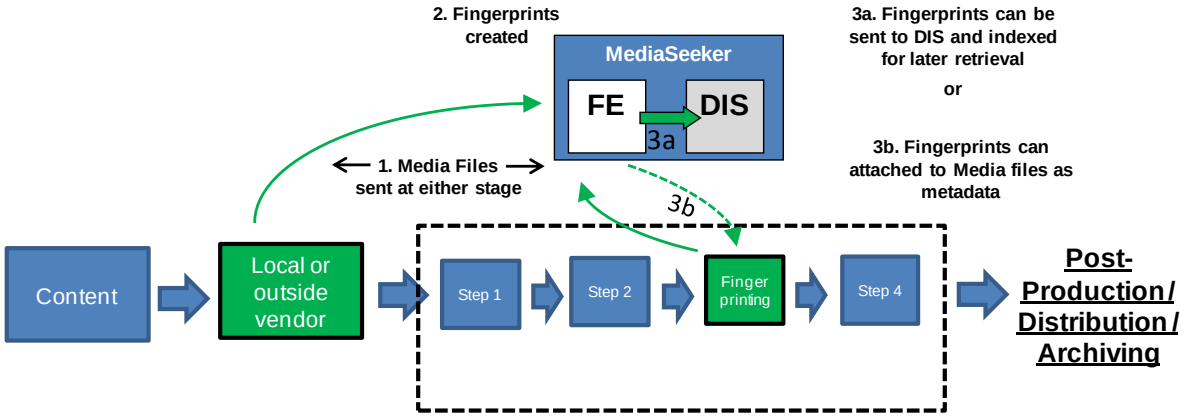


**Figure 10: Generic asset management view of the MediaSeeker platform.**

This generic viewpoint on fingerprints is mentioned in the context of the TIMBUS use case because it shows that the choices made for the use case may be adjusted to more complex business processes at a later stage. One idea here is to have, for instance for a movie, the complete lifecycle stored as a business process. In many cases, it is useful to view the original production process as a structured object. This can help solve problems of provenance of media assets, resolve inefficiencies in media production and may help establish a new form of knowledge management in the media industry. The preservation need in this context is obvious: Movies are watched long times after their production, and the artistic, production and IPR aspects are often lost or their access is difficult because the data is unstructured. Being able to preserve movies as complex objects with multi-perspective structured metadata on them (the context: production, IPR, artistic, content) will simplify keeping them “alive”.

#### 4.3.3.2 MediaSeeker service architecture

The TVCM service used in the scenario is a special form of adaptive video fingerprinting discussed in the previous section. It uses the MediaSeeker platform with internal and external Web service interfaces. Its architecture is shown in Figure 11, including the relation to the contextual services and business processes (arrow direction = “uses” or downstream service dependency, circle connectors = service pins). It consists of three main modules, which can in principle be placed globally distributed and are connected via Web services (also cf. Figure 9):

- *Signal Acquisition and Fingerprinting* performs all pre-processing of the broadcast streams received from satellite or terrestric channels. Here all broadcast stream information is stored temporarily and the fingerprint information extracted. Because the fingerprint information is highly compact compared to the original files, it can be quickly exchanged between system parts. There can be numerous separate units. In this module, hardware is a special dependency because the broadcast

streams require dedicated decoding and often decryption equipment. This will have to be considered for digital preservation.

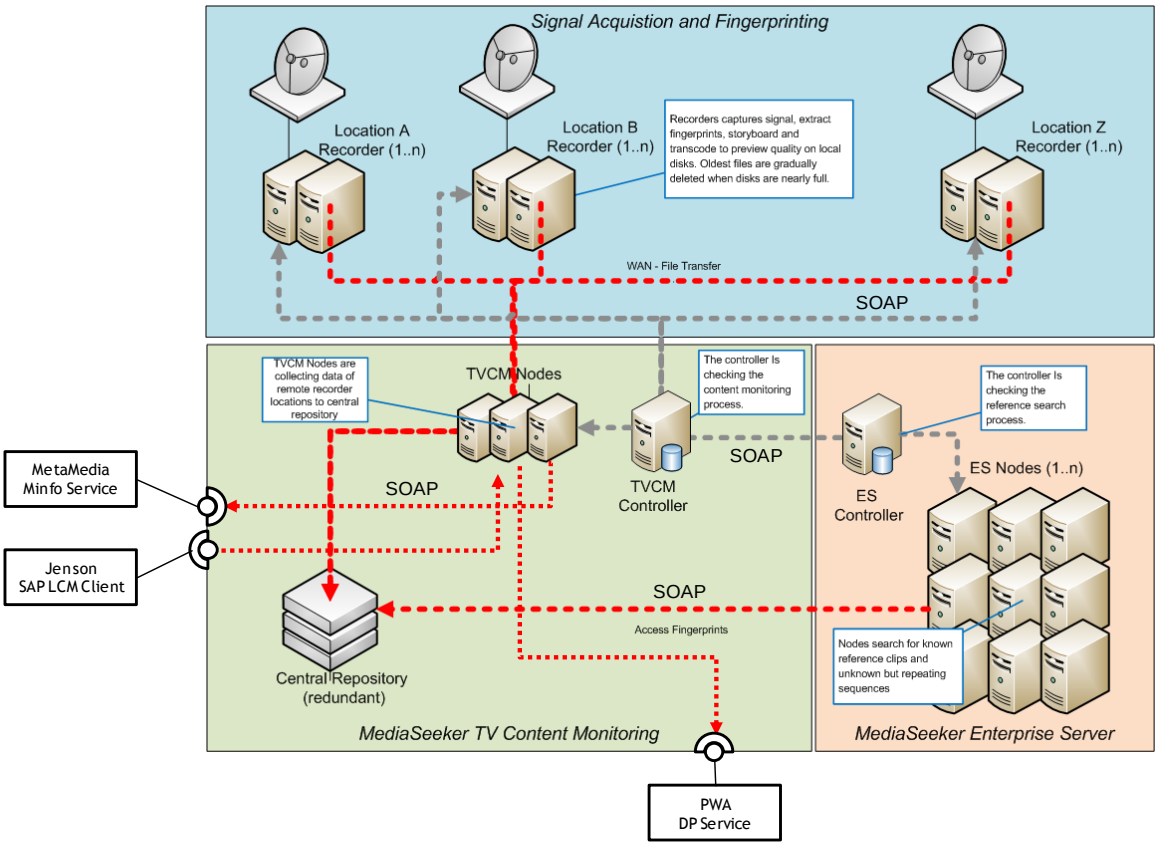


**Figure 11: Architecture of the TVCM process in the media monitoring scenario.**

- *MediaSeeker TV Content Monitoring* controls the fingerprinting process. The TVCM nodes collect data from the remote acquisition modules into a central repository. The TVCM Controller is the master component in the system that drives the actual monitoring process and handles all external service connections. In particular, it provides a service connector for the LCM service with all the necessary interfaces between LCM and the TVCM system. This includes submission of reference data, provision of reports and reference data. The TVCM system also handles the storage of novel advertisement versions for later potential preservation. These preservation requests are received from LCM and are handed over to the preservation service interface provided by PWA.
- *MediaSeeker Enterprise Server* is a cluster of machines that performs the low-level search tasks for known reference clips or unknown clips that may need human attention. The cluster is coordinated by an Enterprise Server Controller. The server module itself only works on fingerprint data, yielding all referential information to the TVCM control. As with the acquisition module, hardware plays a

particular role as many of the algorithms indeed are optimised for a particular target hardware, e.g., graphics processors (GPUs).

**Communication** between the modules is realised using SOAP Web service connections, except bulk video file transfers, which are handled using HTTP transfers between machines and via shared files and folders on the same machine. Such high-volume transfers are only necessary when content is to be transferred to a client or to be preserved, e.g., if media are to be used as detected new advertisement candidates.

Important communication messages between parts of the TVCM service are shown in Table 1. In particular, the actions of creating and querying fingerprints are of importance. The complexity of the service is largely hidden in few service calls.

**Table 1: Messages between modules within the TVCM service (excerpt)**

| ID | Soap Call | Parameters | Return values | Description |
| --- | --- | --- | --- | --- |
| MS1 | TVCMController:: CreateAndIngestFingerprint | MediaFileUri, UseCaseProfileName, Customer ID, … | Fingerprint ID | Create a fingerprint from a given file URI, contextualise by use case and customer |
| MS2 | TVCMNode:: CreateCatalog | CatalogName, … | Catalog ID | Create a new catalog for fingerprints |
| MS3 | TVCMNode:: AddReferenceFingerprintToCatalog | ReferenceFingerprintID, CatalogID ... | Status | Add a fingerprint to a catalog |
| MS4 | TVCMNode:: QueryFingerprint | MediaFPUri, UseCaseName, CatalogID, … | QueryResultSet (XML) | Find fingerprint for video, result set contains for every match: start/end in query and result, media ID, confidence/relevance,... |

Communication with external dependencies of the system works via web service protocols, and the central point of contact is the TVCM Controller that forwards them to the TVCM processor nodes to achieve load balancing. The messages between the TVCM service and upstream and downstream dependencies are shown in Table 2.[7]

**Table 2: Messages between the TVCM service and dependencies (excerpt)**

| ID | Soap Call | Parameters | Return values | Description |
| --- | --- | --- | --- | --- |
| MX1 | LCM→TVCMNode:: SetServiceParameters | Configuration (XML) | Status | Set the parameters of the service, such as interval of reporting, format of reports, channels and times to be monitored, storage of evidence, etc. The service acknowledges with a status message. A corresponding getter method allows checking of the configuration. |
| MX2 | LCM→TVCMNode:: AddReferenceData | ReferenceData (XML) | Status | Add reference data whose fingerprints are to be monitored. This may include for instance logos. |
| MX3 | TVCMController→LCM:: SendMonitoringReport | Report (XML) | Void | Callback to LCM module that sends the aggregated monitoring report for a period set in service parameters. The evidence files of new occurrences of advertisements are stored in the TVCM system and can |

[7] This draft specification had been established as a planning basis.

| | | | | |
|---|---|---|---|---|
| | | | | be accessed via the reported URIs. |
| MX4 | TVCMController→MInfo:: GetBroadcastInfo | Station, TimeInterval | Report (XML) | Get the program information for the station and time interval. This is used by the TVCM controller to create monitoring reports. |
| MX5 | LCM→TVCMNode:: PutPreservedMedia | URI, Metadata | Status | Preserves the media clip. This is part of a legal process preserved by Jenson. |
| MX6 | LCM→TVCMNode:: GetPreservedMedia | URI, format | Status callback | Retrieves the preserved media clip. This is part of a legal process previously preserved by Jenson. The status callback includes the URI to the resurrected media object in the desired format. |
| MX7 | TVCMNode→WPA_DP:: AddPreservedCase | CaseName | CaseID | Add a new case to preserve. |
| MX8 | TVCMNode→WPA_DP:: PreserveMediaClip | MediaUri, CaseID, Metadata, ... | | Add a particular clip to preservation. |
| MX9 | TVCMNode→WPA_DP:: QueryCases | List of Query criteria | List of cases (XML) | Get list of all preserved cases that match criteria. |
| MX10 | TVCMNode→WPA_DP:: GetCasePreservationReport | (List of) CaseID | Report (XML), including URIs | Get a report on the preservation of case, including context graph, URIs and descriptions. |
| MX11 | TVCMNode→WPA_DP:: GetPreservationStatus | (List of) CaseID | Report (XML), including preservation details | Get a report on the status of media preservation, including risk analysis for migration and actions. |
| MX12 | TVCMNode→WPA_DP:: CasePreservationStatus | CaseID, Report (XML) including preservation details and migration recommendations | void | Report on the state of preservation of cases, which is being done to audit changes in the preservation risks (file format deprecation etc.). It is part of iPharro's contract to manage these. |
| MX13 | TVCMNode→WPA_DP:: PreparePreservedItem | URI, target format | Status callback | Trigger retrieval of preserved item in the target format. This may be any digital object, which is prepared at a location reported in the status callback. |

**Preservation aspects.** What is of importance here is how these service connections need to be amended to support digital preservation of the complete system. Special constraints may arise from the fact that all the external service connections of the TVCM service are across corporate boundaries. Moreover, even within the TVCM service itself, corporate boundaries may be crossed. Often in TVCM deployments the Acquisition systems are installed at third-party locations because they need to receive signals for particular regions. Thus the TVCM service itself already implements a business process that spans corporate boundaries. In Section 5.3, a concept is introduced that tries to amend the service concept by "preservability".

Another aspect relevant for preservation is that the Acquisition services in Figure 11 need to be placed in a particular region where they receive their target TV programmes. This raises the question of how to preserve special hardware components that are tailored to the encryption mechanisms that some broadcasters have. Depending on the "depth" of preservation and its goals, different solutions to this may be thought of, from preserving the specification of the functionality via specification of the service software and hardware, up to the (presumably infeasible) task of having to deposit the hardware in a secure space and completing some preparations to re-deploy it in a particular region, all at preservation time. The

notions of migration then become complex as well. The case of code dependent on specific hardware, such as GPUs for fingerprint extraction or querying modules, is related by may be resolved easier, for instance by providing a general-purpose reference implementation that will meet the same functional requirements.

#### 4.3.3.3 Augmented lifecycle for MediaSeeker software solutions

As defined in Section 3.1.2, an augmented service lifecycle includes steps that allow the creation, modification and re-creation of a software service process. All of them require the source code and associated knowledge to build the software, adjust its design and extend the functionalities.

As has been described in Section 4.2, iPharro performs projects for customers that are technologically based on customisation and integration of components of the MediaSeeker platform into their production or analysis processes, which may include specific customisation of a Web service for external consumption by the customer process. The advantage of the customisation approach is that MediaSeeker can reach much better adaptation of the precision and scalability of the service than would be possible using service configuration. The flexibility of the MediaSeeker components with respect to fingerprinting parameters (e.g., density, cf. Section 4.3.3.1) leads to high variability of applications in customers' environments at different points in their respective workflow.

In the following, we will describe the typical approach for such custom development as a basis for preserving custom service solutions, also for development and escrow support.

**1. Service negotiation.** In order to perform a custom solution project, a number of parameters need to be agreed between service customer (i.e., Jenson in the scenario) and provider iPharro. They may be considered an early form of context for the process that the service will be embedded in and are typically fixed during the service negotiation phase (cf. Figure 1). The parameters have been described in more detail in [Hein12a] and include:

- *Fingerprinting use case:* The type of fingerprinting task, cf. Figure 9, e.g., video reel comparison where the locations of change in two versions of a video need to be identified or advertisement monitoring as in the current scenario. TVCM and advertisement monitoring are other examples of fingerprinting use cases requiring changes in the architecture and implementation of a solution. The idea of use cases is to provide the basis for development projects, and variants between solutions of a given use case may be considered sub-use cases, which finally leads to a hierarchy of system structures.
- *Fingerprinting requirements and service levels*: Metrics and target service levels are used as side conditions for the tasks. For the video reel comparison task, a realistic metric is the recall, the number of differences found versus ground truth. In many cases, service levels of of 99.9% recall are offered. Other side conditions include robustness to certain parameters, such as aspect ratio and resolutions of the videos: Changes should be identified between both full HD and anamorphic SD versions of films or different TV input formats.
- *Auxiliary service levels*: The system is specified for instance for a target throughput of data. Scalability of the integrated solution needs to be adjusted to this, for which the fingerprint density

is one parameter, cf. Figure 9, the server pool size another. Other constraints in this category include operation delay (up to near real-time operation) and the information and format of the reports.

- *Reference content*: Typically, the customer provides reference data that is used to test the metrics against in order to validate the system. Such a sample corpus represents typical cases of data encountered during deployment and can be used to establish a reference service level for performance parameters like precision.
- *Runtime environment and service embedding*: The customer has its own workflow and business processes established that the target system needs to be integrated with (cf. Figure 11). This includes the business processes proper as well as the underlying infrastructure.

Part of this information is elicited in questionnaires like that shown in Figure 12, another in a consulting sessions between iPharro and customer teams. As a result of such contexts varying between customers, standard solutions (corresponding to use cases and sub use cases) can be partially re-used.

| | |
|---|---|
| Company Name | |
| Contact Person | |
| Address | |
| Contact Number | |
| Email Address | |
| **General Questions:** | |
| What is the file format? | |
| What codecs are used? | |
| What is the resolution of the content? Example: HD,SD,CIF | |
| How much reference content has to be queried against? | |
| How many queries / day on average? | |
| How long is the average video (reference and query)? | |
| Does the query consist of single, independent files or is it a continuous stream? | |
| Which types of content changes are likely to be present in the compared video material? | |
| Black/White Vs. Color | |
| Cropping | |
| Overlays<br>• Small overlays – Example: TV channel logos<br>• Medium overlays – Example: Subtitles<br>• Large overlays – Example: Information boxes | |
| Large scale overall quality reductions<br>• Camcorder recordings<br>• Very low bit rate video with artifacts | |

| | |
|---|---|
| **Company Name** | |
| **Contact Person** | |
| **Address** | |
| **Contact Number** | |
| **Email Address** | |
| **Hardware Requirements:** | |
| **Number of channels monitored** | |
| **Type of signal** | |
| **Number of locations monitored** | |
| **Name of locations with number of channels monitored at each location** | |
| **Available Internet speed** | |
| **File format and codec of video files (Example: MPEG,WMV)** | |
| **Video standard (Example: NTSC,PAL)** | |
| **Number of reference advertisements** | |
| **For what period of time should the fingerprints be stored?** | |
| **What Capture System, if any, do you currently use?** | |
| **What existing hardware do you have?** | |
| **Additional Comments / Description** | |
| **System Recommendation:** | |
| • **Platform:** .Net Framework 3.5 with Service pack 1<br>• **Operating system:** Microsoft Windows XP | |

**Figure 12: Example questionnaires for requirements elicitation.**

**2. Service development.** iPharro engages in a custom development project. At its start, based on the requirements a specification is produced that takes into account the available technology and experience. The current situation is that this input is distributed across the enterprise: As is customary in software companies, source control systems are used extensively, allowing management of versions of the MediaSeeker software components on a module basis. Furthermore, there exists a body of knowledge on past solutions that are documented with a heterogeneous structure, including system documentation, the

joint versions of the system components working together (so they can be re-compiled), as well as reference data sets. A central factor here is also the experience that exists in iPharro's team in order to work with the existing components and additional information.

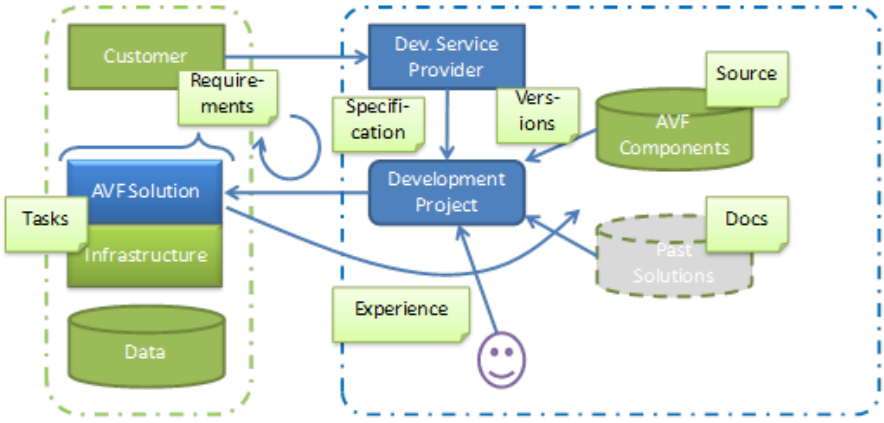


**Figure 13: Development process for AVF/MediaSeeker solutions.**

A graphical overview of the general development process is given in Figure 13. In the project, existing technology (from the repository of components) is re-combined to integrate the novel solution. In many of these projects, also solution-specific components and services are developed and integrated. The project starts with the use case established in the negotiation phase (see above) that has reference solutions from past projects associated with it. Each of these reference solutions uses a set of configurations – which may be changed at runtime or development time. These configurations are component-specific and may be shared between solutions and even use cases. The development projects will therefore start with a configuration of components and then add functionality is needed to fulfil the particular customer needs.

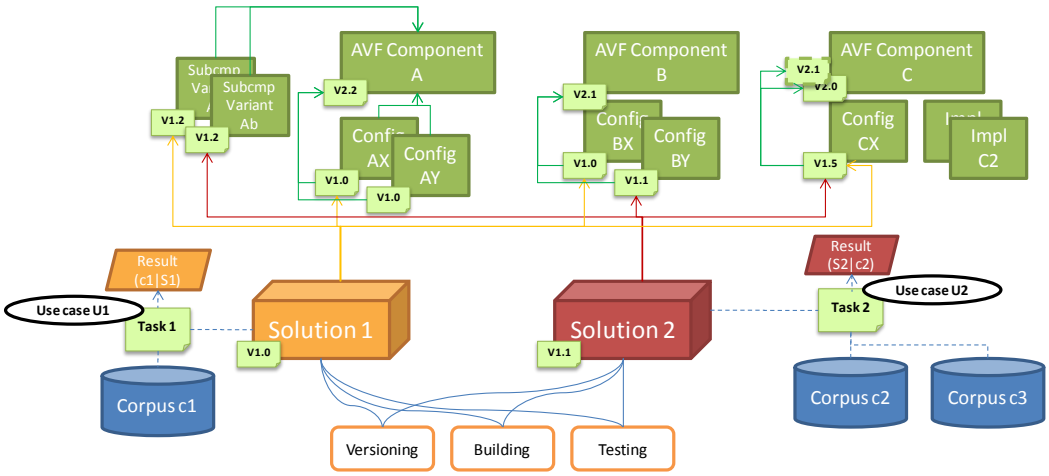


**Figure 14: Sketching the dependencies in AVF solutions.**

**2a. Repository structure and versions.** In order to efficiently provide custom solutions, software components are maintained as described above. The speciality of this maintenance is that components are parts of solutions that fulfil particular performance metrics and other requirements. These requirements

are documented along with the versions. The organisation of the associated repository is outlined in Figure 14 in an exemplary manner.

The crucial aspect here is the set of dependencies of S1 and S2 on different parts maintained in the repository. This includes AVF software components 1 to 3, each of which is contained in both solutions. As an example, despite the relative difference between Reel Comparison and Ad Monitoring, one needs several processing components in common, e.g., a similarity metric for both tasks as well as a feature extractor. Thus, the core implementations of AVF Component A and B may be the same. Configurations may be different (leading to Config AX and AY) and the dependency on these needs to be captured. Furthermore, consider there are small changes to the component that require customisation beyond configuration, i.e., adding a custom heuristic for Solution 2. This may in the future be handled by creating subcomponent variants, Aa and Ab, each slightly modifying the core of Component A. Currently, either there are two branched versions of the component or a case differentiation is done in source code (for instance using #ifdef pre-processor statements in C/C++).

There are other cases where differences between the same component in different solutions may occur: Solutions may use different versions of a component, subcomponent variant or configuration. And they may use different implementations, e.g., one ported to C++ and one in C#, requiring a particular version of the .NET framework.

In other words, identical MediaSeeker components are reusable in multiple solutions and even use cases. But there may be differences in the revisions, configurations and reference corpora.

**2b. Challenges.** While the current development approach is viable, its limitations for larger numbers of projects and a continuing support and maintenance of the resulting solutions and components becomes evident considering two cases of practical relevance (for more details, see the white paper [Hein12a]):

- *Challenge 1: New Solution using existing components:* Development of a new solution based on given proven results is difficult because maintenance of requirements, side-conditions, data etc. is handled independently and therefore complex or requires high experience. The related question is: How can we develop a new solution that uses the optimum versions of components?
- *Challenge 2: New Component for existing solutions:* Maintenance of a set of custom solutions with a common set of components is difficult because it is not clear what happens when updating a particular component in the different systems. The current approach is to use unit tests, but it is difficult to maintain these across a set of projects if systems cannot be preserved as coherent entities. The related question is: How can we preserve a set of interrelated systems on a functional level while updating their components? That is, to maintain and improve the functional and non-functional quality of the systems at different customers' installations while updating them to the latest versions of underlying components to increase their quality, security etc.

An important aspect of the TV monitoring service is its application in customised form within the context of parent business processes. Re-engineering software services to be preservable must consider this augmented lifecycle.

**3-6. Provisioning, operation, modification and decommissioning:** The MediaSeeker solution is evaluated before provisioning it as a service for the customer, first with project-specific metrics at iPharro's site using for instance unit tests and integration tests, and finally in the target runtime environment at the customer's site or in a data center. Both internal and external evaluations (with respect to the development environment) may feed back to Step 2 for optimisation. The solution is deployed and operated for the service consumer, and a customer support process started that executes along the complete lifetime of the service operation. During the lifetime of the service, the solution may later be modified according to the requests of the customer or following some improvements of the core components, for instance a refinement of the core fingerprinting algorithm. This is similar to the development phase and may closely relate to the challenges mentioned in section 2b above. Finally, decommissioning terminates the deployment and support processes, creates final documentation and removes the service if it is not needed for other customers.

**7. Software Escrow.** As a final step in the augmented service lifecycle, software escrow allows the relocation of the service provision in cases the service cannot be maintained or run by iPharro. For software escrow, typically the source code and documentation is deposited at a third party, along with the service executable files. In order to re-run the service process, normally it necessary to transfer a high amount of knowledge, which in many cases is difficult and time-consuming. For this reason, software escrow often cannot prevent service downtime or support issues.

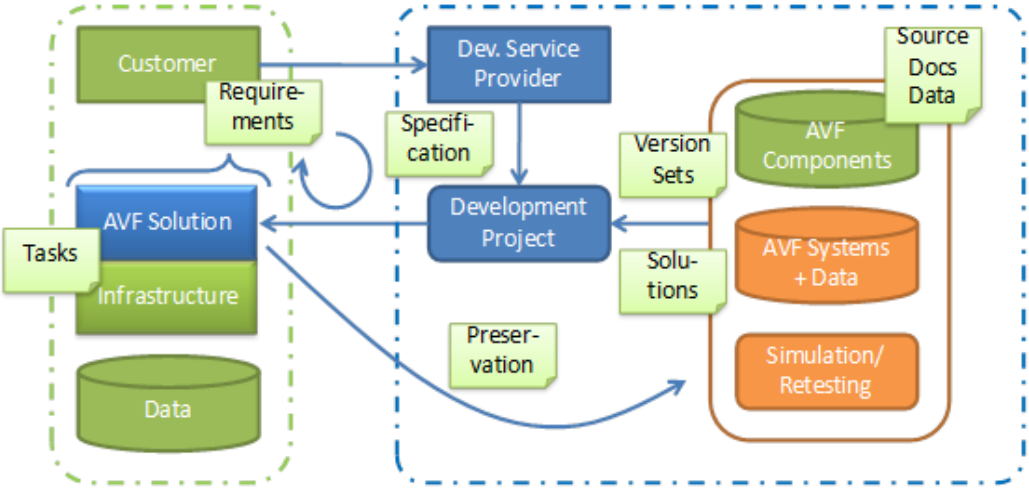


**Figure 15: Solution development with preserved reference solutions.**

**Preservation aspects.** Preservation of MediaSeeker solutions is relevant for several steps in the augmented service lifecycle. In steps 1 and 2, tests of reference solutions with new reference data or on new target metrics may allow much higher efficiency and cost reduction during negotiation (e.g., supporting proofs of concept and pre-sales activities) because the original systems that services have been executed on may be re-deployed as complex objects if adequately preserved. Furthermore, during development integration tests are simplified if partial systems may be re-deployed. Figure 15 illustrates how a DP system may changes the development infrastructure at iPharro. Furthermore, maintenance and customer support

based on heterogeneous solutions is largely simplified if the knowledge of the complex objects can be represented in a structured form and tests can be done using the emulated remote processes.

Regarding the preservation challenges formulated in section 2b above, the development of new solutions corresponds to Challenge 1. What is also of interest is challenge 2 when iPharro's MediaSeeker components are improved. Being able to re-deploy the original preserved processes run at customer's sites results in a huge reduction of testing time, as access to remote systems may be omitted (except for the final deployment).

Finally, software escrow is largely simplified if the service can be re-deployed directly using preservation infrastructure. In the scenario, this infrastructure may be provided by PWA, with iReady taking over the maintenance processes and after some learning time replace the re-deployed components directly with the components built from source code.

### 4.3.4 Perpetual Web Archive Inc.

Perpetual Web Archive Inc. (PWA) is a fictional entity that offers "Digital Preservation as a Service" (DPaaS). PWA offers to preserve business processes, the process execution context, within which data is processed, analysed, transformed and rendered, accessible over long periods.

The majority of Perpetual Web Archive's competitors are focused on the long-term storage of digital content such as documents, images, video, or audio files. Perpetual Web Archive, on the other hand, offers to preserve complete IT infrastructures, which support business processes. This includes the preservation of software and hardware stacks as well as relevant contexts, which together provide the execution layer for executing business processes.

Being able to provide the complete execution environment to support business processes, PWA offers "Business Continuity as a Service" (BCaaS). If the primary IT infrastructure of PWA customers is not in operation anymore, PWA services allow the re-deployment of business processes using previously preserved IT infrastructure. This may for instance be used to mitigate organisational risks (see Section 3.3.1) by supporting software escrow agreements; in our context business continuity has less focus on operational risks (see Section 3.3.3). Furthermore, as in the case of iPharro as a customer, PWA allows preservation of systems for later deployment and re-testing by custom development companies and other customers.

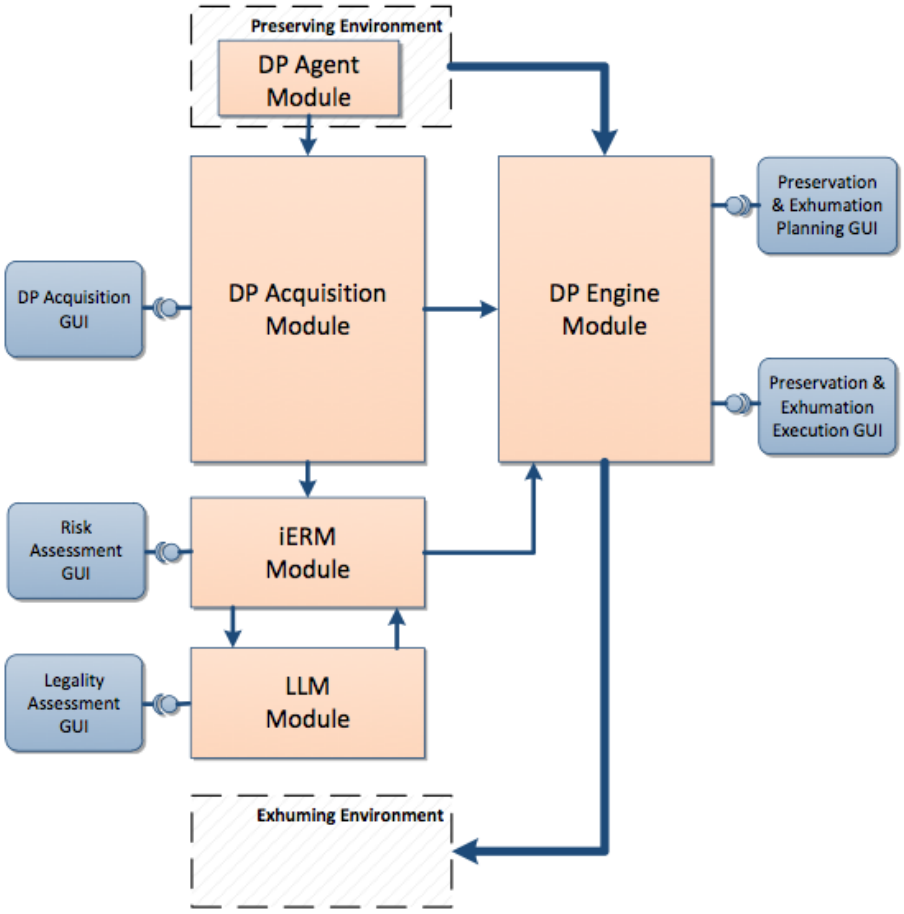


**Figure 16: PWA's DP service architecture.**

Figure 16 gives an overview of the architecture of PWA's service. Its technology is based on the TIMBUS technology stack and reflects the latest TIMBUS revision, published in deliverable D5.5 in September 2012. Full details of its constituent parts can be found there, but for convienience, a brief description of its five primary modules is also provided here:

- **DP Agent** – the DP Agent is running within the Service Consumer environment and captures data required for performing DP.
- **DP Acquisition** – the DP Acquisition component is used for collecting and combining dependencies, contexts and event logs from different DP Agent Modules into the unified model.
- **Intelligent Enterprise Risk Management** – The IERM module is used for assessing risks associated with BPs and dependent resources. It generates a report describing risk levels and cost values associated with different preservation scenarios for the specified subset of BPs.
- **Legality Life-cycle Management** – This module is used for assessing impacts of legalities issues on different preserving scenarios for the specified subset of BPs.

- **DP Engine -** The DP Engine is used to generate preservation & exhumation plans by utilising the risk report and the unified model. The DP Engine also provide mechanisms for verification and testing different stages of preservation & exhumation processes.

**Implementation.** The implementation of this service for the use case will be based on the results output of WP6.

### 4.3.5 MetaMedia Ltd.

MetaMedia Ltd. is a fictional TV metadata provider that offers quality data of content streams publicly offered. The service provides information about what was aired when in a consistent format, making subscribers independent of broadcaster-specific formats of electronic program guides (EPGs). The software service MetaMedia offers is called MInfo and complies with the standard service lifecycle.

It represents a downstream software service in the scenario and allows to model transitive service relationships as well as DP requirements for a partial process offered by a third party.

**Implementation.** The implementation of this service will be based on a custom development by iPharro, using a real database or broadcast information.

## 4.4 Business processes involved

This section describes the major business processes that the stakeholders need to run in the media monitoring scenario. Starting from an overall process, the adverse advertisement monitoring (AAM) process, its dependent processes are described.

**Note:** This section is preliminary. The full business processes need to be presented here.

### 4.4.1 Adverse advertisement monitoring (AAM) and Legal case management (LCM)

The AAM process is the parent process executed at Jenson in order to enforce compliance of the broadcast video with the interests of Jenson's clients. Adverse advertisements are detected and in applicable legal action pursued. This process is an example of a human and software-based process.

The Legal Case Management Process (as described in Section 4.3.1) is implemented in a SAP system and can be (partially) preserved using the ILM module. However, the ILM module is not mend to preserve data generated by the AAM process nor it is supposed to preserve the AAM execution environment.

The LCM process accesses the AAM process four times:

1. A new case file is created if the iPharro monitoring service provides reports that indicate potential intellectual property infringement or adverse content. The receiver of these reports is a Jenson clerk working in the Customer Relationship department. A new case file is also created if a Jenson customer files an official complaint. Reports are generated by the AAM process in an automated fashion by the TVCM service provider iPharro. The assessment is done by a human.

2. Once the clerk starts to collect material to compile a legal case, the iPharro service provides additional material, such as video footage and detail reports.
3. If a legal case has been accepted by a Jenson Lawyer, an automated preservation request to preserve related material, reports and other documents is sent to iPharro. This preservation request details the preservation duration and other information.
4. If a legal case has to be re-opened, the Jenson lawyer may want to investigate archived material. In specific cases (if there is doubt of service level fulfilment), it might be required to evaluate the methods and methodologies used by iPharro to screen TV channels and to analyse the media stream.

### 4.4.2 TV content monitoring process (TVCM)

The TVCM process is a direct dependency of the AAM process. It is a purely service-dependent business process (SDBP). The process monitors a wide range of media channels for occurrences of advertisements, as described above. Figure 17 gives an overview of the fingerprinting and querying processes at the core of the TVCM service. There are two types of customer request: If a video clip is available, the process extracts and stores fingerprints (done by the Fingerprinting module in Figure 11), and if a reference or a fingerprint exists, the request is handled by the query engine (in MediaSeeker Enterprise Server in Figure 11).

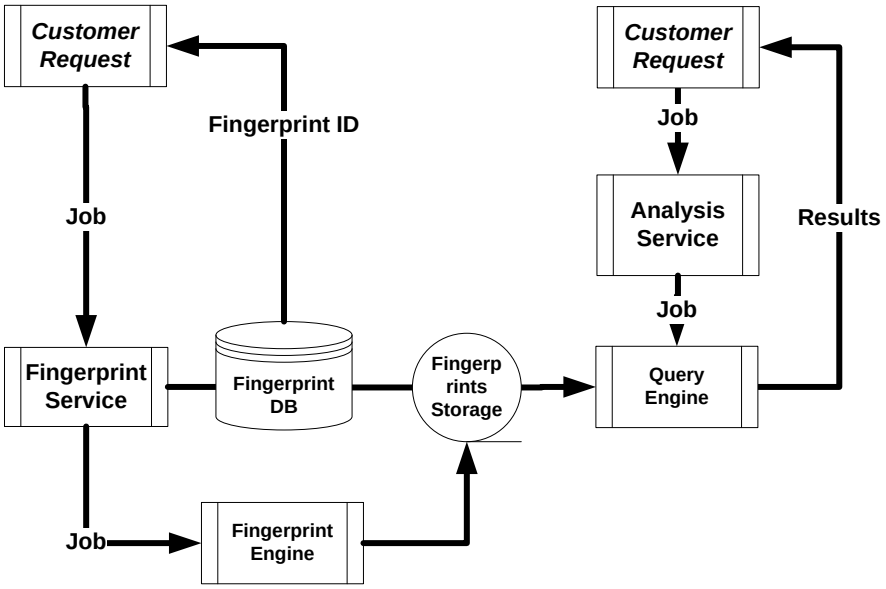


**Figure 17: TVCM fingerprinting and querying, central process.**

A viewpoint from a business-process perspective of iPharro's service is given in Figure 18. Here the dependency on external program metadata is made explicit.

In order to prove the SLA parameters upon SLA breach allegations, a reference TVCM process needs to be re-deployed and tested, along with its dependencies.

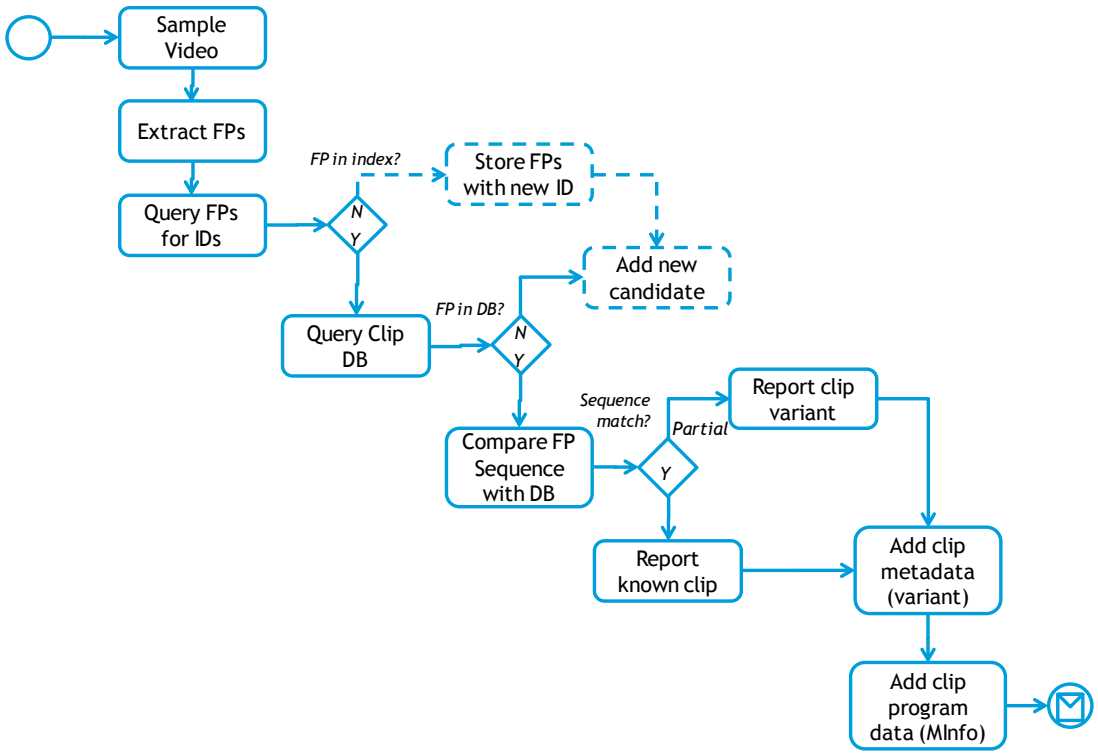


**Figure 18: iPharro service business process with metadata integration (FP = fingerprint; dashed = optional detection of unknown video sequences).**

### 4.4.3 Other processes

There are various other processes in this scenario, such as the development process itself at iPharro, the media meta-information extraction process at MetaMedia and the digital preservation process itself at PWA. Although these processes are of interest, in the current stage of the scenario development we concentrate on the AAM/LCM and TVCM processes because they cover the aspects deemed central to study service preservation, leaving the others for the future.

# 5 Requirements for preservable software services

This section of this report will defined use-cases and will list the high-level requirements for each of the use. The requirements listed in this section are an initial draft and will be further detailed when the full deliverable is submitted in M24 (March 2013).

Similar to the deliverables D8.1 and D9.1, the requirements defined in this section were elicited using a goal-oriented approach. Goals are objectives that the system to be described should achieve. Goal formulations are intended properties to be ensured and bounded by the application domain [Lams01].

To define requirement levels, the key words *must, must not, required, shall, shall not, should, should not, recommended, may* and *optional* in the following requirements list are to be interpreted as described in [RFC2119].

The requirements given will generally be of two types:

- Requirements for the TIMBUS framework (functional, non-functional)
- Requirements for service implementation and interfaces (functional, non-functional)

To obtain some of the requirements, we chose to start from an initial approach to preservation of services that allows clean integration into service-oriented architecture. This concept will be presented in Section 5.3 because D7.2 is due in M24. It results in conditional requirements for “preservable” services that are subject to a preliminary concept developed for D7.2. These requirements will be marked with a * after their identifier.

The rest of this chapter is structured as follows. In Section 5.1, the service lifecycle is analysed for steps where preservation actions occur, and the respective stakeholders in the scenario are identified. Section 5.2 then collects the goals of the system from the perspective of the scenario and extracts use cases. Section 5.3 presents the preliminary concept of an automated service preservation negotiation mechanism that serves as a basis to define requirements, also for development of services. Based on this, Sections 5.4 and 0 list the functional and non-functional requirements for the digital preservation system and for services, respectively.

## 5.1 Preservation in the service lifecycle and stakeholders

Considering the service lifecyle, the places where preservation actions are to be taken may be seen in Figure 19, which is an extension of Figure 1 presented in Section 3.1.2. In particular, these steps are:

- *Preservation of service processes*: When a new solution is developed, its process can be preserved.
- *Preservation of support process*: For escrow, the support process for a given solution is preserved. This process is considered out of scope for the current version of the scenario. It requires continuous additions to the preserved digital object.

- *Migration of preserved items*: After modification of the SLA and the corresponding changes in the process or its software services, the preserved version needs to be migrated to the new structure.
- *Redeployment of reference process*: Given there exist preserved reference processes (solutions), they are redeployed as a basis for development, in an SLA fulfilment assessment, or for software escrow agreements.

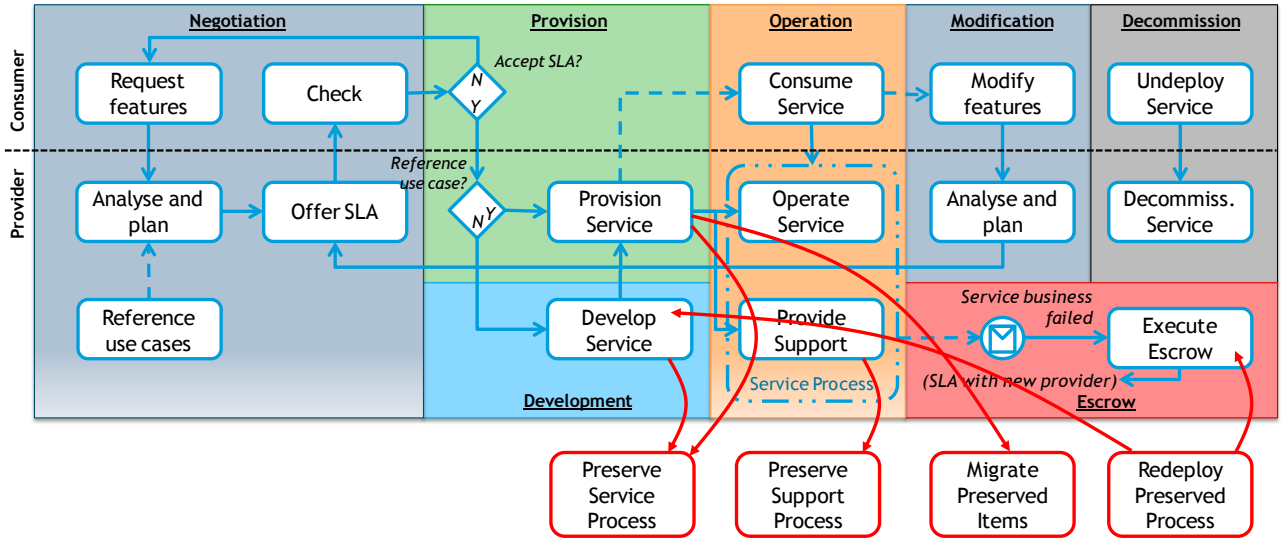


**Figure 19: Digital Preservation in the service lifecycle.**

In terms of particular use cases, the scenario considers five types of stakeholder that are directly involved with the preservation actions:

- *Case Lawyer (Jenson)*: Handles a case for the Jenson company and can choose to preserve a case of adverse advertisement.
- *IT Manager (iPharro)*: Besides handling the service provisioning and operation, she is responsible for preservation of TVCM service installations, both for escrow and for reference purposes. The IT manager also handles re-deployment in case SLA fulfilment is challenged.
- *Solution Developer (iPharro)*: The Solution Developer is concerned with customising and improving the TVCM service. He re-deploys a service process in order to gain insights in a past solution or may use the solution to bootstrap development by creating a proven execution environment for novel functionalities.
- *Escrow Agent*: The Escrow Agent acts as a notary for software escrow. She can trigger a re-deployment of the solution if the contractual pre-conditions are met. The solution is re-deployed within the IT landscape of a new service provder – in the example this was iReady GmbH in Figure 5 in Section 4.2.

- *DP Operator (iPharro or PWA):* The DP operator handles the actual preservation task at technical level. In the scenario he may be either at iPharro or PWA, depending on whether one considers PWA as a pure technology service provider or a preservation consultancy.

## 5.2 Goals and use cases

For the scenario, two goals may be summarised: Digital preservation of services as a specific method to perform risk management and development support, and enterprise risk management in more general terms, providing a complement to preservation-based risk management. In particular, the following high-level goals may be stated:

- G1 Support digital preservation for service oriented architectures
    - G1.1 Preserve service with distributed process context
    - G1.2 Support redeployment of service with different variations
    - G1.5 Support regular service lifecycle
    - G1.3 Support development of preservable services (augmented lifecycle)
    - G1.4 Support service escrow (augmented lifecycle)
- G2 Perform Enterprise Risk Management

**Table 3: Goals for scenario**

| **ID: G1** | **Name**: Support digital preservation of services | **Super-goal**: n/a | **Sub-goals**: G1.1-5 |
| --- | --- | --- | --- |
| **Goal Description**: Support digital preservation for service oriented architectures | | | |
| **Supplementary Information:** The system is to support the preservation of software services in a service-oriented architecture. | | | |
| **Responsible stakeholder:** DP operator (iPharro or PWA) | | **Using stakeholders:** All | |
| **ID: G1.1** | **Name**: Preserve service with distributed context | **Super-goal**: G1 | **Sub-goals**: n/a |
| **Goal Description**: Preserve service with distributed process context | | | |
| **Supplementary Information:** The system should be able to capture distributed context of a service, i.e., dependencies, metadata and other data. Beside methods to manually determine context, automatic means of context negotiation should explicitly be considered. | | | |
| **Responsible stakeholder:** DP operator (iPharro or PWA) | | **Using stakeholders:** IT Manager, Solution Developer | |
| **ID: G1.2** | **Name**: Support service redeployment | **Super-goal**: G1 | **Sub-goals**: n/a |
| **Goal Description**: Support redeployment of service with different variations | | | |

**Supplementary Information:** The system should support the redeployment of a service with its context, but in different types of variation, including replaying it with the same data ("instance preservation"), rerunning it with potentially different data ("process preservation" proper) and with different data and variable service structure ("template preservation").

| **Responsible stakeholder:** DP operator (iPharro or PWA) | | **Using stakeholders:** IT Manager, Solution Developer, Escrow agent | |
|---|---|---|---|
| **ID: G1.3** | **Name**: Support regular service lifecycle | **Super-goal**: G1 | **Sub-goals**: n/a |
| **Goal Description**: Support preservation over the regular service lifecycle | | | |
| **Supplementary Information:** Service preservation should be supported over the complete regular lifecycle of negotiation, provisioning, operation and decommissioning, as well as modifications. Augmented lifecycle steps are considered separately. | | | |
| **Responsible stakeholder:** Solution Developer (iPharro) | | **Using stakeholders:** n/a | |
| **ID: G1.4** | **Name**: Support service development | **Super-goal**: G1 | **Sub-goals**: n/a |
| **Goal Description**: Support development of preservable services | | | |
| **Supplementary Information:** The development of "preservable" services should be supported. Such services for instance may be queried to report their context parameters automatically and may have references to source code and design documentation. | | | |
| **Responsible stakeholder:** Solution Developer (iPharro) | | **Using stakeholders:** n/a | |
| **ID: G1.5** | **Name**: Support service escrow | **Super-goal**: G1 | **Sub-goals**: n/a |
| **Goal Description**: Support service escrow agreements. | | | |
| **Supplementary Information:** As part of the augmented service lifecycle, redeployment of the service should be supported in a different context as a measure of business continuity after business failure of the service provider. The service context should be captured deeply enough to allow a team outside of iPharro to fulfil the SLA, i.e., operate the service and provide support under comparable service levels as iPharro. | | | |
| **Responsible stakeholder:** DP operator (iPharro, PWT) | | **Using stakeholders:** IT manager, Escrow agent | |
| **ID: G2** | **Name**: Support Enterprise Risk Management | **Super-goal**: n/a | **Sub-goals**: n/a |
| **Goal Description**: Support Enterprise Risk Management. | | | |
| **Supplementary Information:** Enterprise risk management provides complementary means to digital preservation as a risk mitigation method. In the service architecture the methods should be considered, including risk analysis and business continuity management supported by existing solutions. This way, the scenario can show more realistic setting of risk management measures available at an enterprise. | | | |

| **Responsible stakeholder:** Case Lawyer (Jenson) | **Using stakeholders:** Case Lawyer (Jenson) |
|---|---|

The use cases connected to these goals and the stakeholders in the previous section are depicted in Figure 20. In Table 4, they are described.

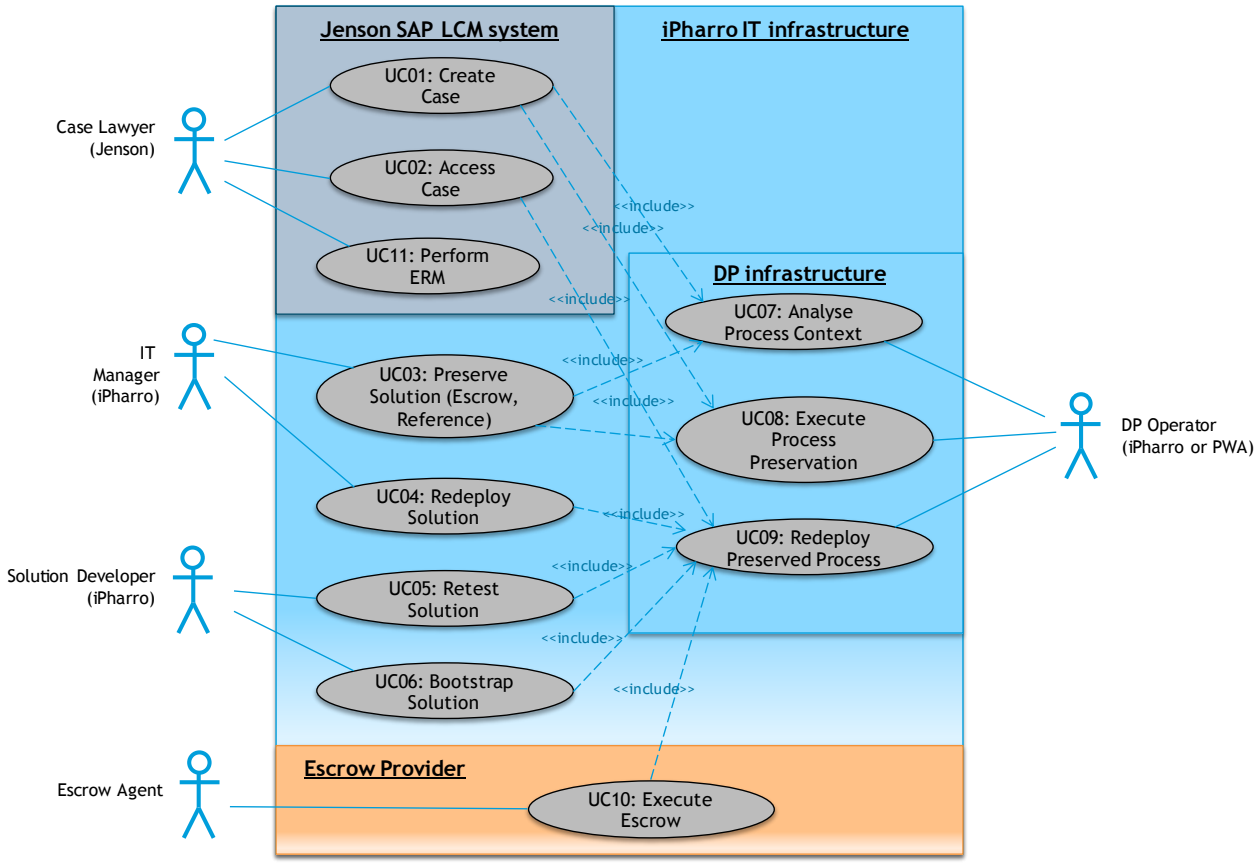


**Figure 20: Digital Preservation and ERM use cases.**

**Table 4: Preservation use cases.**

| Use Case | Description |
|---|---|
| **UC01 – Create Case** | A Jenson lawyer adds a new legal case that under corporate and legal regulations needs to be preserved. This action triggers an analysis of the process (or media) in UC07 context as well as it executes the process preservation action in UC08. |
| **UC02 – Access Case** | A Jenson lawyer retrieves the preserved case process. This action triggers a redeployment request with the DP operator, UC09. The DP infrastructure returns the media in context of the process preserved in the SAP LCM module. The preserved context represents only a sub-context of the case process. |
| **UC03 – Preserve Solution** | An IT manager at iPharro preserves a solution along with its relevant context objects. As the IT manager is not a DP expert, this action is not |

| | |
| --- | --- |
| **UC04 – Redeploy Solution** | An IT manager at iPharro wants to re-deploy the service in order to prove its SLA fulfilment and that the service levels have not changed. The use case triggers a redeployment of the respective process in UC09. Redeployment uses the original reference data but may be run on other data. Re-deployment may be performed as virtualisation or into the production service infrastructure if the service process was migrated properly during its preservation period. |
| **UC05 – Retest Solution** | A solution developer at iPharro re-deploys the service in order to test it against data and partly changed configurations. The use case triggers a redeployment of the respective process in UC09. Retesting re-executes the process at least partially, allowing the developer to quickly test new features or provide support for the customer even if the customer business process or service environment cannot be accessed. |
| **UC06 – Bootstrap Solution** | A solution developer at iPharro re-deploys the service in order to bootstrap a new solution project. The use case triggers a redeployment of the respective process in UC09. Bootstrapping is largely dependent on source code, setting up the development environment for the project and providing access to the re-executed service process. |
| **UC07 – Analyse Process Context** | This use case is executed by the DP operator at iPharro or PWA (if they provide consulting services). The operator analyses the context of the business process to be stored, i.e., the business process that a particular solution is embedded in as well as the dependencies that it has. Furthermore, depending on the purpose of preservation (UC02, 04, 05, 06, 10), the depth of the context is determined, subject to policies that exist in the company and themselves depend for instance on the requirements of SLA and escrow agreements.<br>Preservation of a service will preserve the semantics and interfaces, its documentation and eventually the software components. The TIMBUS project will develop analysis tools for this task. The result of this task is a context graph and a preservation plan or script. Typically, different MediaSeeker use cases will result in template preservation plans that may be adjusted by the DP operator. |
| **UC08 – Execute Process Preservation** | This use case executes the preservation plan prepared in UC08. It can be executed with a large portion of automation, allowing it to be offered as a service by PWA (whereas the context analysis will be better executed by domain experts at iPharro). The execution also contains continuing auditing of integrity and migration issues of the use case. |
| **UC09 – Re-deploy Preserved Process** | The preserved process is re-animated within its execution context. In particular, the environment for re-execution of the process (or service) is checked and adjustments made if there are missing or incompatible items in the context graph, including dependent services such as MInfo in the scenario. |

| | |
| --- | --- |
| **UC10: Execute Escrow** | When the company goes out of business, the Escrow Agent associated with the escrow agreement executes an escrow. For this, similar to UC5 a re-deployment is undertaken. However, here the redeployment environment is the IT or virtualisation infrastructure of a third-party provider. This use case triggers UC09. |
| **UC11: Perform Enterprise Risk Management** | In regulated industries, enterprise risk management is required to support some processes. The case lawyer (or another responsible person) at Jenson runs the risk management procedures necessary to ensure the compliance of the case proceedings with contractual or regulatory rules. This includes risk analysis and business continuity management, which are supported by existing solutions. |

## 5.3 Towards "preservable" software services in TIMBUS[8]

The preservation of software services represents one main objective of TIMBUS. In the presented scenario, the communication of Web services has been addressed, including that between corporate and potentially legislative boundaries. The question is how web services may be augmented to support preservation in this context. This section outlines an initial concept of a solution.

### 5.3.1 Message passing for context discovery

One idea to consider for "preservation-proof" services is a message-passing approach: A query message may be sent to a service to expose all of its dependencies given a particular purpose and risk level, and the service itself constructs these dependencies by querying its direct downstream dependent services and exposing its inherent context information, the messages are passed until the service dependency graph is traversed completely, aggregating the results as response of the original query.

Depending on the two parameters *purpose* and *risk level* (or potentially more), the service may choose to expose different depths of its context, for instance omitting source-code dependencies if the service is to be re-deployed only for reference purposes (as in the SLA proof case in the scenario) but including it for escrow or development support. Each context item will have connected with it a risk level, a URI where it can be reached and a structure that contains important parameters from the context model (cf. D4.2 and D4.3).

**Purpose.** The purpose of preservation is the intended goal that a preservation action needs to fulfil. There are different profiles that may be considered, including risk containment like software escrow or other purposes outlined in Section 3.4. Given different purposes, in turn different preservation policies and rules may be established, for instance regarding the inclusion of source code in the process context.

**Risk level.** The risk level is a type of quality-of-service or SLA parameter for preservation that aggregates the risks of not being able to re-deploy for the actual service for the purpose. It may be a structured numerical value and processing it may be in connection to the formal semantics of the context model.

[8] This section resulted from discussions and analyses in the iPharro team and was intended to form the basis for D7.2.

### 5.3.2 Service provisioning

This querying mechanism in turn will require the service to implement the concepts of preservation purpose and risk level and add the respective logic to it, e.g., reusing a configured component of a TIMBUS "service DP provisioning framework" (or in the scenario the respective framework provided by PWA). When the service is being developed, it needs to be processed by preservation specialists that certify it for particular preservation purposes and usage scenarios. The service consumer may require during SLA negotiation that the service be "preservable" under the purpose given and maximum risk level.

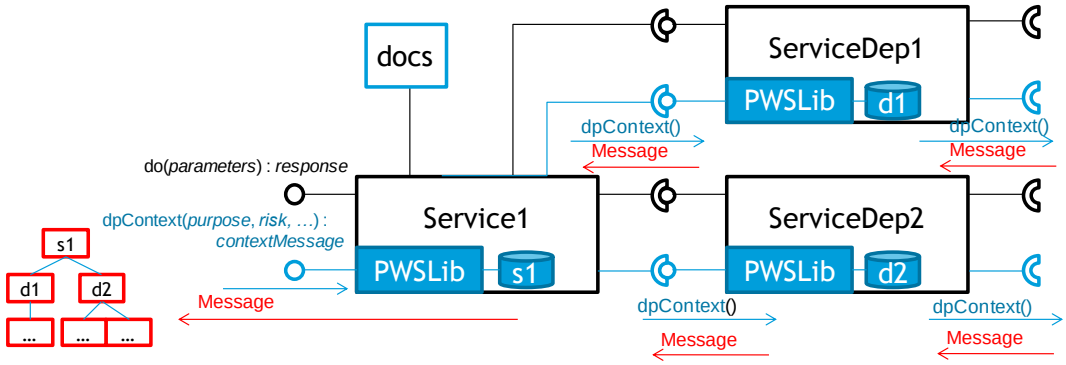


**Figure 21: Message passing for service context discovery.**

At source-code level, introducing preservation methods as part of the software architecture may be realised for instance using Aspect-Oriented Programming, a method that adds generic functionality ("cross-cutting concerns") across implemented objects, separating the service logic from preservation issues.
In Figure 21, the message passing algorithm in an adequately provisioned service is illustrated. Here PWSLib is the "Preservable Web Services" library out of the above mentioned service DP provisioning framework and the database symbols connected to PWSLib represent the data necessary to establish and track context local to each individual service. Reasoning on such context information weighted by risk levels and other confidence information may for instance be achieved by work on probabilistic extensions to description logics, such as log-linear description logics [NNS11].

### 5.3.3 Future directions

This initial work only covers two parameters and a way to automatically detect the context of a given service. The outlined concept is only one possibility but its advantage is that it elegantly can be integrated into a framework of SLA negotiation, potentially even automatic. It furthermore shows that in fact preservable software services may need to undergo a defined provisioning process to be able to expose their context. In the future, the service DP provisioning framework will also have to define the preservation purposes and policies for service certifications. Ideally, it would become the basis for standardisation activities in the context of D7.2 and WP2.

Alternatively, variants to the presented approach may be pursued and update the requirements accordingly.

## 5.4 Preservation System

This section considers requirements posed to the preservation system developed in TIMBUS.

### 5.4.1 Functional requirements for Preservation System

The following table lists functional requirements for the preservation system.

**Table 5: Functional requirements for the preservation system.**

| **ID**: FR-P1 | **Name**: Preservation of distributed service landscape | **Goal**: G1.1 |
| --- | --- | --- |
| **Short description**: The system must provide means to preserve a distributed Service Landscape. | | |
| **Additional information**: The system should provide means to preserve a distributed Service Oriented Architecture. This requires distributed coordination efforts such that all partners in a distributed SoA setting preserve all software systems and software processes which constitute the system to support a business process. The preservation environment itself may or may not be a distributed system. | | |
| **Cross references**: | | |
| **ID**: FR-P2 | **Name**: Context capture | **Goal**: G1.1 |
| **Short description**: Capture of contextual information and metadata. | | |
| **Additional information**: Metadata related to services (e.g. service descriptions) and processes (e.g., process models) | | |
| **Cross references**: | | |
| **ID**: FR-P3 | **Name**: Monitor context | **Goal**: G1.1 |
| **Short description**: The system must be able to monitor and assess the environment where the researcher is executing its analysis | | |
| **Additional information**: The system should provide means to capture process contextual information and metadata related to services. The system needs build in knowledge (e.g. a common model) about what needs to be documented with regard to business processes and the supporting Soa/SaaS software/technology stack such that they can be exhumed and rerun in the future.<br>The environment of a system includes developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.<br>Context monitoring should detect changes in:<br>• SoA/SaaS IT landscape layout<br>• Business process<br>• Execution context | | |
| **Cross references**: | | |

| **ID**: FR-P4 | **Name**: Software Artefacts Dependency Analyser | **Goal**: 1.1 |
|---|---|---|

**Short Description:** The system must provide means to analyse software artefact dependencies

**Supplementary Information:** Dependencies include

- Executable (in different versions)
- Dynamic/static libraries (in different versions)
- Configuration files
- Data files

**Cross references**:

| **ID**: FR-P5 | **Name**: Service Dependency Analyser | **Goal**: 1.1 |
|---|---|---|

**Short Description:** The system must provide means to analyse service dependencies

**Supplementary Information:** The system should provide a tool to automatically identify software service (e.g. Web services, RPC, ...) dependencies. For, example, the tool could use network monitoring to collect, and analyse network traffic. Using the collected data the tool should decide which services communicate with each other.

The tools should also allow to complementary add manual data to the automatically retrieved information.

Compared to the local operating software artefact dependency analyser tool, the service dependency analyser captures information across domains, e.g. form various partners in a SoA/SaaS environment.

Dependencies may include:

- WebServices,
- HTTP/Rest
- Unix IPC,
- Remote Procedure Calls

**Cross references**:

| **ID**: FR-P6 | **Name**: Guided Service Resurrection and Integration Environment | **Goal**: 1.2 |
|---|---|---|

**Short description**: The system should provide a tool that guides the resurrection of services and services landscapes.

**Additional information**: The system should provide a tool to manage the resurrection of single services and distributed service landscapes. The tool should produce guidance to the administrator of what he needs to be resurrected, and what service could be replaced by current (future) systems. The tool should

| | | |
|---|---|---|
| also guide the administrator how the resurrected service landscape should be integrated into existing (future) processes. | | |
| **Cross references**: | | |
| **ID**: FR-P7 | **Name**: Service level for preservation level | **Goal**: 1.3 |
| **Short description**: The system must provide a mechanism to negogiate the parameters under which the DP service guarantees to recover. | | |
| **Additional information**: The system must provide means to restore a business process and process related data within predefined time frames (which have been derived from a Business Impact Analysis, Dependency/Risk Analysis. This may be an automated / semi-automated activity. | | |
| **Cross references**: | | |
| **ID**: FR-P8 | **Name**: Legal Life-Cycle Management | **Goal**: 1.3 |
| **Short description**: The LLM should provide IP management, IT contracts, regulation standards, and user defined obligations. | | |
| **Additional information**: This requirement ensures that the end of the service lifecycle is adequately supported. After a service has been decommissioned, it is necessary to execute a digital preservation process. In order to handle all legal and regulatory issues, a Legacy Lifecycle Management (LLM) is required. Business processes depending on a SoA/SaaS comprised of many interconnected services the legal/regulatory issues become more difficult to maintain and evolve over a long period of time.<br>The LLM module should support:<br>• Intellectual Property Management<br>• IT contracts<br>• Regulation and other legal binding constraints<br>• User defined obligations | | |
| **Cross references**: | | |
| **ID**: FR-P9 | **Name**: Dependency and Risk Analysis Tool | **Goal**: 2 |
| **Short Description:** The system must provide means to perform a Dependency and Risk Analysis | | |
| **Supplementary Information:** The system must provide tools to perform a DA/RA. The DA/RA tool should enable the risk expert to identify critical resources, risks associated to resources, and depended resources. The tool should help the expert to understand.<br>Risk Experts do not only have to understand the effects of an adverse incident on a business, they also have to understand dependencies among business processes, dependent resources and possible root-causes of an adverse incident. | | |

A Dependency Analysis / Risk Analysis (DA/RA) aims to identify all stakeholders, assets, resources and relations to external parties on which a business process depends and analyse the nature, magnitude and probability of adverse events to these dependencies which may disrupts the process.

For example, the analyst needs to understand how a broken air-conditioning unit may affect the datacentre and servers deployed in this datacentre.

**Cross references**:

| **ID**: FR-P10 | **Name**: Business Impact Analysis Tool | **Goal**: 2 |
|---|---|---|

**Short Description:** The system must provide means to perform a Business Impact Analysis

**Supplementary Information:** The system must provide tools to perform a Business Impact Analysis taking process information into account. The BIA should enable the risk expert to identify critical business processes, evaluate potential damage and losses. The BIA should consider financial and non-financial (such as legal) values and indicators a like.

Business Impact Analysis is a crucial tool used by a risk expert to better understand an organisation and helps the risk expert and senior management to make informed decisions. A Business Impact Analysis aims to (a) identify critical business processes (b) and assesses and evaluates potential damages or losses at business level that may be caused by failures of identified in Section 3.3.

Disruptions of business processes may have a financial impact, legal consequences or may cause effects on other business values and indicators, such as reputation, customer satisfaction or customer churn rates. Those values are known as business level Key Performance Indicators (KPIs) and should be considered.

| **ID**: FR-P11 | **Name**: Risk Metrics | **Goal**: 2 |
|---|---|---|

**Short Description:** The system must provide means to develop and set various types risk metrics

**Supplementary Information:** The system must provide tools to develop, derive and set risk metrics for critical business processes.

The risk expert has to specify acceptable timeframes in which a level of operations of a process has to be restored, such that the organisation can continue to deliver products and services.

| **ID**: FR-P12 | **Name**: Risk simulations | **Goal**: 2 |
|---|---|---|

**Short Description:** The system must provide a tool to run risk simulations.

**Supplementary Information:** The system must provide tools to create various scenarios and enable the risk to run simulations to validate preservation strategies. Risk experts do not only have to understand the effects of an adverse incident on a business, they also have to understand dependencies among business processes, dependent resources and possible root-causes of an adverse incident.

The risk manager is required to demonstrate that risk mitigation strategies are complete, coherent, current and correct. Therefore, the risk expert needs tools to run exercise and simulate various scenarios

| | | |
|---|---|---|
| to validate recovery plans and Business Impact Analysis. | | |
| **ID**: FR-P13 | **Name**: Business Process Recovery | **Goal**: 2 |
| **Short Description:** The system must provide means to recover a process | | |
| **Supplementary Information:** The system must provide means to restore a business process and process related data within predefined time frames (which have been derived from a Business Impact Analysis, Dependency/Risk Analysis. This may be an automated / semi-automated activity. | | |
| **ID**: FR-P14* | **Name**: Automatic service context negotiation | **Goal**: 1.1 |
| **Short Description:** The system must provide runtime support for automatic service context negotiation. | | |
| **Supplementary Information:** This conditional requirement is based on the concept outlined in Section 5.3. The DP system should provide for an automatic means of context negotiation and discovery. This may work with message passing as outlined or with another method that automates the process of finding dependencies. | | |
| **ID**: FR-P15* | **Name**: Automatic service context negotiation tools | **Goal**: 1.1 |
| **Short Description:** The system must provide development support for automatic service context negotiation. | | |
| **Supplementary Information:** This conditional requirement is based on the concept outlined in Section 5.3. Given the outlined message passing mechanism, services need to be prepared to be able to respond with context information. This should be supported by the DP framework, possibly in the form of a library that allows to implement the message passing protocol and assemble the context graph of a service. The system may also include an explicit provisioning tool to fill the service with preservation-relevant data. | | |
| **ID**: FR-P16 | **Name**: Development support | **Goal**: 1.4, 1.5 |
| **Short Description:** The system must provide development support for re-deployed systems. | | |
| **Supplementary Information:** When the service is re-deployed for particular purposes (development, escrow), the direct reference to service source code and design documentation should be available to the preservation team. This should include a reference to the source code repository that the service was built from. What may be desired here is the preservation of the build and test infrastructure, allowing to re-run not only the service process itself from the preserved data, but the actual build and test process. By extension, the system may provide means to re-run and continue maintenance and support processes, which are inherent for the development and improvement of the service. | | |

### 5.4.2 Non-functional requirements for Preservation System

The following table lists non-functional requirements for the preservation system.

**Table 6: Non-functional requirements for the preservation system.**

| **ID**: NR-P1 | **Name**: Preservation effort and time | **Goal**: G1.1 |
| --- | --- | --- |
| **Short description**: The preservation effort and time need to fulfil an upper bound. | | |
| **Additional information**: When preserving a service, there needs to be an upper bound on the effort to be put into the actual preservation tasks, including context capture and preservation execution. The upper bound may be determined by context and purpose. | | |
| **Cross references**: FR-P1-2, 5-6, 9 | | |
| **ID**: NR-P2 | **Name**: Monitoring effort | **Goal**: G1.1 |
| **Short description**: The monitoring effort needs to fulfil an upper bound. | | |
| **Additional information**: Similar to NR-P1, monitoring should be possible with limited effort. | | |
| **Cross references**: NR-P1 | | |
| **ID**: NR-P3 | **Name**: Re-deployment effort and time | **Goal**: G1.2 |
| **Short description**: Redeployment effort and time needs to fulfil an upper bound. | | |
| **Additional information**: Similar to NR-P1, re-deployment should be possible with limited effort given a particular re-deployment task. This may be more complex when a service needs to be re-run with variants of the original dependencies. | | |
| **Cross references**: NR-P1 | | |
| **ID**: NR-P4 | **Name**: Security and privacy levels | **Goal**: G1 |
| **Short description**: Preservation needs to preserve the security and privacy of the original service. | | |
| **Additional information**: When a system is preserved, the preservation process or resulting digital object may not be a back door to breaching security or privacy guarantees. A special case arises from the scenario where PWA acts as a preservation service provider. If PWA has no non-disclosure agreement with Jenson, the service and data need to be encrypted in a form to protect their privacy and security. | | |
| **Cross references**: | | |
| **ID**: NR-P5 | **Name**: Preservation across corporate boundaries | **Goal**: G1 |
| **Short description**: The DP system needs to provide functionality to work across corporate boundaries. | | |
| **Additional information**: Services provided by third-parties need to be taken into account for preservation, and the DP system needs to support this with methodology and tools. This may include the partial disclosure of service functionality under escrow or other SLA constraints. | | |

| **Cross references**: | | |
|---|---|---|
| **ID**: NR-P6 | **Name**: Consistency of stored information | **Goal**: G1.2 |
| **Short description**: The information stored for the complex digital object must stay consistent. | | |
| **Additional information**: All information and data need to stay consistent across the preservation duration. This is especially crucial if dependencies have to be migrated. | | |
| **Cross references**: | | |
| **ID**: NR-P7 | **Name**: Human-interpretable context graph | **Goal**: G1 |
| **Short description**: The service context information should be presentable in a human-readable form. | | |
| **Additional information**: In order to be able to adjust preservation context, users should be able to re-enact and interpret the context graph extracted from the service dependencies. This means either tool support to display the context information or provide the context graph in a format that is directly legible. | | |
| **Cross references**: | | |
| **ID**: NR-P8 | **Name**: Context unchanged | **Goal**: G1 |
| **Short description**: Preservation should not compromise the original context of a service. | | |
| **Additional information**: When the service is preserved, its context should not be jeopardised. However, this will not | | |
| **Cross references**: | | |
| **ID**: NR-P9 | **Name**: Permission to preserve parent processes | **Goal**: G1.2, G1.4, G1.5 |
| **Short description**: The preservation system should define a way to negotiate preservation of the parent process of a service. | | |
| **Additional information**: Custom solutions for a given service may have particular patterns of access to the service that may facilitate its operation and re-testing confidence when re-deployed from preservation. Typically a customer of a service will not expose the service context beyond the actual solution project. The preservation system should therefore include templates for permission statements that allow service developers preservation of the parent process context to an extent that it can be reasonably emulated. Here intellectual properties as outlined in D4.4 play a central role. | | |
| **Cross references**: | | |

## 5.5 Preservable services

This section lists requirements for services to be preservable.

### 5.5.1 Functional Requirements for Preservable Services

The following table lists functional requirements for preservable services.

**Table 7: Functional requirements for preservable services.**

| **ID**: FR-S1 | **Name**: Service protocols | **Goal**: G1 |
| --- | --- | --- |
| **Short description**: Service preservation protocols must follow Web service standards. | | |
| **Additional information**: The preservation of services must follow the common standards used for Web service, including SOAP, REST and XML-RPC. | | |
| **Cross references**: | | |
| **ID**: FR-S2 | **Name**: Context structure disclosure | **Goal**: G1.1 |
| **Short description**: Services need to communicate their immediate context structure. | | |
| **Additional information**: Services should be able to tell about their context structure, so the usage of the service paradigm facilitates handling their preservation. The mechanism and format of the local context structure is open, but the mechanism outlined in Section 5.3 may give a partial solution to this. | | |
| **Cross references**: | | |
| **ID**: FR-S3* | **Name**: Context message passing support | **Goal**: G1.1 |
| **Short description**: The services should support the context message passing algorithm. | | |
| **Additional information**: This conditional requirement is based on the concept outlined in Section 5.3. The requirement. When a service is to be preserved, it should support the interface for querying its context. | | |
| **Cross references**: FR-P14*, FR-P15* | | |
| **ID**: FR-S4* | **Name**: Preservation purpose profiles and risk level | **Goal**: G1.1 |
| **Short description**: Services must allow querying of preservation purpose profiles and risk level. | | |
| **Additional information**: This conditional requirement is based on the concept outlined in Section 5.3. It extends the requirement FR-S2*. Services must allow querying of preservation purpose profiles, allowing to determine by an automatic means what preservation purposes the service fulfils and what context this entails. | | |
| **Cross references**: FR-S3* | | |
| **ID**: FR-S5 | **Name**: Service directory support (optional) | **Goal**: G1.1 |
| **Short description**: Services should be findable in a directory service (optional). | | |

| | | |
|---|---|---|
| **Additional information**: Services should be findable in a directory service like UDDI. This may allow resolution of dependency conflicts when a given service cannot be preserved. | | |
| **Cross references**: | | |
| **ID**: FR-S6 | **Name**: Resolution of non-preservable dependencies | **Goal**: G1.1 |
| **Short description**: Services need a means to resolve non-preservable dependencies. | | |
| **Additional information**: If a service is not preservable, there should be a means to preservice its functionality. This requirement generalises FR-S5. | | |
| **Cross references**: FR-S5 | | |
| **ID**: FR-S7* | **Name**: Service context monitoring | **Goal**: G1.1 |
| **Short description**: Services need to support context monitoring. | | |
| **Additional information**: A preserved service should be able to be monitored in terms of its context. For instance, if data formats change or become obsolete, it should be possible to track the dependency over time. This may be implemented by the service preservation interface. | | |
| **Cross references**: | | |
| **ID**: FR-S8 | **Name**: Legal constraints negotiation | **Goal**: G1.1 |
| **Short description**: Services need to resolve legal constraints of their operation. | | |
| **Additional information**: If a service runs under legal constraints, such as dependencies on particular data, it should be possible to query this information in an easy way, so all provisions for preservation can be facilitated. In general, the legal constraints of the preservation of software need to be taken into account, and D4.4 gives an extensive overview of the provisions within the European union and national legislations. | | |
| **Cross references**: | | |
| **ID**: FR-S9 | **Name**: Service context migration | **Goal**: G1.1 |
| **Short description**: Services need to support context migration. | | |
| **Additional information**: When preserved, services should be able to support the migration of their own functionality or their context. For instance, if a dependent service is not available any more, and according to the risk level the preservation system did not require to archive the dependency but only a reference to it, the service needs to be adjusted. | | |
| **Cross references**: | | |
| **ID**: FR-S10 | **Name**: Service security and privacy | **Goal**: G1.2 |
| **Short description**: Services need to keep security and privacy rules in preservation. | | |
| **Additional information**: When preserved and re-deployed, a service must run under the same security | | |

| | | |
|---|---|---|
| and privacy constraints as the original service. This may include the need to re-organise access rights. | | |
| **Cross references**: | | |
| **ID**: FR-S11 | **Name**: Service context re-deployment types | **Goal**: G1.2 |
| **Short description**: Services must support re-deployment of different types. | | |
| **Additional information**: Re-deployment may be in the form of retest, rerun and partial rerun, referring to exactly the same service process with the same data, with different data (re-test), and with a subset of the context graph and new components and the same or novel data (partial re-test).<br>Another requirement is that of re-deployment in a different runtime environment. Services may need to support this, and for the case of preservation this may require specific changes in the service metadata for message passing or another context discovery mechanism. | | |
| **Cross references**: | | |
| **ID**: FR-S12 | **Name**: Service lifecycle support and backwards compatibility | **Goal**: G1.3 |
| **Short description**: Services need to support the regular lifecycle and should be backwards compatible with ordinary services. | | |
| **Additional information**: Preservable services should have the same lifecycle as ordinary services. They should be able to work like ordinary services, the additional interfaces (like the message passing algorithm) should be transparent to regular operation. However, the additional preservation interfaces need to be compatible with the operations on services over the lifecycle, including preservation in service negotiation, adding it when developing the service, provisioning it, keeping the preservability aspect alive when operating the service, and being able to migrate it for service improvement and decommissioning. It needs to be noted that the original service and the preserved services do not need to be in the same step of the lifecycle. | | |
| **Cross references**: | | |
| **ID**: FR-S13 | **Name**: Service development support | **Goal**: G1.4 |
| **Short description**: Services need to support development methods. | | |
| **Additional information**: Preservability of a software service in many cases requires the inclusion of its source code into its direct process context. Otherwise different re-deployment purposes are jeopardised. This means that a preservable software service needs to have a reference mechanism that points to its original implementation. This may be a source code repository with respective revision numbers and branch URLs, along with links to the design documentation. A respective interface may be added to the context graph disclosed when querying the service. | | |
| **Cross references**: | | |
| **ID**: FR-S14 | **Name**: Service maintenance and support processes | **Goal**: G1.4, 1.5 |

| | | |
|---|---|---|
| **Short description**: Services need to allow links to support processes. | | |
| **Additional information**: Support processes for software services include all enabling processes that are needed to operate the service over its lifetime. As a representative case, the actual customer support (first-level, second-level) may be taken. The requirement for this is that there should be a link to the support database and allows to link the improvement and bugfixing process to it. This is especially useful for software escrow where a business ceases to exist and the support process can be continued because all the original support tickets are preserved. | | |
| **Cross references**: | | |
| **ID**: FR-S15 | **Name**: Service escrow support | **Goal**: G1.5 |
| **Short description**: Services need to support software escrow. | | |
| **Additional information**: The service needs to support a mechanism that allows software escrow. This requirement implies that the service needs to provide enough information about itself so any new business can provide the service with a similar quality as the original provider. This requirement may thus include FR-S13 and 14. | | |
| **Cross references**: FR-S13-14 | | |

### 5.5.2 Non-functional Requirements for Preservable Services

The following table lists non-functional requirements for preservable services.

**Table 8: Non-functional requirements for preservable services.**

| | | |
|---|---|---|
| **ID**: NR-S1 | **Name**: Service must be relocatable | **Goal**: 1.2 |
| **Short description**: The service must be able to run on a different system than the original one. | | |
| **Additional information**: If the service is relocated to a new system, it should in principle be working. There are special cases where this is systematically impossible, such as the case of the Acquisition service of iPharro in Section 4.3.3.2 where a receiver may be dependent on (1) hardware and (2) on a specific geographic location. | | |
| **Cross references**: | | |
| **ID**: NR-S2 | **Name**: Service must preserve access rights | **Goal**: 1.2 |
| **Short description**: The service must allow the transfer of access rights | | |
| **Additional information**: The service must preserve the access rights after re-deployment. | | |
| **Cross references**: | | |
| **ID**: NR-S3* | **Name**: Upper bound for context message passing. | **Goal**: 1.1 |

| | | |
| --- | --- | --- |
| **Short description**: The service must allow a maximum time for the context message passing algorithm. | | |
| **Additional information**: The service must reveal its context quickly, so large context dependency graphs may be traversed. The actual time for this is not important, it should only be considered as a generic service level. | | |
| **Cross references**: | | |
| **ID**: NR-S4* | **Name**: Context reliability and security. | **Goal**: 1.1 |
| **Short description**: The service context discovery algorithm must be reliable. | | |
| **Additional information**: When the message passing is used to discover and negotiate context of a service, it must be reliable and secure. That means that a context discovery mechanism needs to (1) provide some error checking method to discover problems of the automatism, that (2) if no error occurred it can be relied on to capture the correct context and (3) that there need to be ways to prevent or audit manipulation of the automatic service context discovery for instance by adverse dependent services. | | |
| **Cross references**: | | |

# 6 Conclusions and Outlook

This document is report on work in progress and an outlook of the deliverable D7.1, which is due in month 24. This report discusses a scenario, goals and requirements to support digital preservation of a distributed Service-Oriented Architecture.

The scenario given in Chapter 3 is a multi-stakeholder Digital Preservation scenario that reflects major requirements associated with typical real-world processes supported by a SOA. In particular, the scenario of a TV content monitoring service is defined where a market research company contracts a media service provider to monitor TV streams in order to create viewing and broadcast statistics as well as to ensure legal and contractual compliance of aired advertisements. This initial scenario already points out a multitude of DP challenges, including preservation across corporate boundaries, preservation for software development and proof of service levels as well as software escrow. Chapter 4 discusses goal and requirements derived from the scenario. We conclude that the preservation of software services is not easy complex task. However, formalised DP service protocols (developed in D7.2) may help simplifying some of the intricacies, such as context discovery for services.

This report is on-going work and as such it allows evaluation of the scenario and its adequacy as a model for more complex service networks. Until completion of the deliverable D7.1 at M24, the following work will be completed:

- *Scenario further extended and refined*: Although the current scenario already covers a broad range of aspects for digital preservation of software services, more cases of service interaction should be considered to cover cases where dependency preservation may prove difficult. This may include more goals pursued to extract requirements.[9]
- *Elaborated definition of the business processes involved*: The actual business processes have been sketched but more formal definitions are input.
- *Refined risks and detailed requirements for Enterprise Risk Management*: The current list of requirements is preliminary and not complete yet. Especially the legal boundaries within the context of a service process are not sufficiently captured yet.
- *Requirements and concepts for service (re-)engineering processes*: Development of preservable services will be elaborated, and the requirements for the TIMBUS framework and the services themselves refined.
- *Synchronising requirements with WP8 and WP9*.

[9] As the partner iPharro will exit the TIMBUS project after this report, the scenario will be rewritten. However, the generic nature of the structure chosen in the media analysis case – company consumes service that themselves have several dependencies across corporate boundaries – is designed to be applicable to a wide range of scenarios and thus expected to be easily transferrable to a new industrial case. The same is expected for the requirements that have been extracted: They are generic across many scenarios.

- *Mapping of requirements to TIMBUS framework components in WP5 and WP6*.

With the refined scenario results, several other tasks in the TIMBUS project will be delivered with information, most importantly T7.2, which will use the requirements information to define service interface standards and may build on the message passing mechanism outlined in Section 5.3. T4.3 and T5.5 will review conceptual and architectural constraints that the scenario imposes, and in T6.1 and T6.2 tools for risk management and DP will be aligned with the specifics of SOA services.

# 7 References

[Lams01] Axel van Lamsweerde. Goal-Oriented Requirements Engineering: A Guided Tour, Proceedings RE'01, 5th IEEE International Symposium on Requirements Engineering, Toronto, August 2001, 249-263.

[Hein12a] Gregor Heinrich, Adaptive video fingerprinting in Timbus: An overview of a scenario, Technical Whitepaper, Timbus project, v1 Sept. 2011, v2 June 2012.

[Hein12b] Gregor Heinrich, Freshmind – a context visualisation tool, Timbus project, WP6 working document, June 2012.

[NNS11] Mathias Niepert, Jan Noessner, Heiner Stuckenschmidt: Log-Linear Description Logics. IJCAI 2011

[Ould96] Ould, M., Strategies for Software Engineering: The Management of Risk and Quality, John Wiley & Sons, San Francisco, 1996

[RFC2119] IETF Network Working Group, "Key words for use in RFCs to Indicate Requirement Levels", IETF Best Current Practice, RFC 2119, 1997.

[Thor11] Grant Thornton, Issues and trends: Assessing and managing SaaS risk, SaaS, Risk Survey, 2011