FInest – **F**uture **In**ternet **e**nabled optimi**s**ation of **t**ransport and logistics networks

D5.2

Conceptual Design of Collaboration Manager Component

Project Acronym	FInest	
Project Title	Future Internet enabled optimisation of transport and logistics networks	
Project Number	285598	
Workpackage	WP5 Business Network Collaboration	
Lead Beneficiary	SAP	
Editor(s)	René Fleischhauer	SAP
Contributors(s)	Silvio Tschapke	SAP
Reviewer	Alias, Cyril	UDE
	Agathe Rialland	MRTK
	Marianne Hagaseth	MRTK
Dissemination Level	Public	
Contractual Delivery Date	30th March 2012	
Actual Delivery Date	30th March 2012	
Version	V1.0	

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 285598

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 285598

Abstract

This document contains the second deliverable of Work Package 5. The work package is responsible for the Business Collaboration Module (BCM), which aims at the introduction of an infrastructure to securely manage the end-to-end networks of transport and logistics partners. It integrates information from different external sources as well as other modules of the FInest platform and makes this available for end-users of the system. In order to ensure the non-disclosure of confidential data, the BCM enables user and access management, which provides users with specific views on the data accordingly to their individual disclosure level.

In this document we focus on the refinement of previously elaborated conceptual architecture of the BCM and the previously conducted requirements analysis. Additionally, we describe a demonstrator, which was developed by joint efforts between domain and ICT partners of this work package. It considers the effective coordination of resources by the provisioning of a central portal. The demonstrator was the source for a variety of insights in the demands of the T&L domain and with this, builds an important pillar within the conducted refinement process. Furthermore, this deliverable describes the preliminary approach to realize the concept of Collaboration Objects and thereby, moves towards the technical specification of this FInest module.

Consequently, we primarily address two tasks, defined by the Description of Work for Work Package 5: T5.1 – Requirements Analysis and Selection of Technology Baseline and T5.2 – Conceptual Design and Technical Specification. Task T5.3 – Technological Alignment with the FI PPP Core Platform is not addressed again, because there is no new input from the FIWARE project that has to be considered. Additionally, we give a preview on task T5.3 – Initial technical specification of collaboration manager component.

Document History

Version	Date	Comments
V0.1	08-02-2012	Definition of document structure
V0.5	14-03-2012	Finished demonstrator description, updated conceptual architecture and updated requirements
V0.9	16-03-2012	Ready for internal review
V1.0	30-03-2012	Document ready for submission

Table of Contents

List of Tables

List of Figures

Acronyms

Acronym	Explanation
ARH	Port of Alesund (FInest partner)
ATA	Actual Time of Arrival
ATD	Actual Time of Departure
BCM	Business Collaboration Module
CO	Collaboration Object
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
IMO	International Maritime Organization
LSC	Logistics Service Client
LSP	Logistics Service Provider
MOF	Meta Object Facility
MRTK	Marintek (FInest partner)
NCL	North Sea Container Line (FInest partner)
T&F	Tyrholm & Farstad (FInest partner)
SOA	Service-oriented Architecture
OMG	Object Management Group
AIS	Automatic Identification System
GSM	Guard-Stage-Milestone

1. Introduction

In the previous deliverable we described the Business Collaboration Module (short: BCM) as the central unit within the FInest platform that securely manage end-to-end networks between transport and logistics partners. Its main task is the execution of a transport plan created by the Transport Planning Module (TPM) and the integration of information from FInest modules, external legacy systems (e.g. ERP) and user input. With this, the BCM is the central entry point for users to get access to information about currently executed transport processes (e.g., the current position of the goods or the state of the process) as well as stored historical data. In order to manage all this information the BCM uses a special modeling concept, called Collaboration Objects. Each Collaboration Object encapsulates a special aspect of a transport process (e.g., a certain leg or maybe a particular service such as warehousing[1]) and the whole process is described through a combination is described by a combination of several of these objects. Logistics processes have a tight interweaving of (static) data and process information. For example, a certain transport process has a maritime leg or service, which ships the goods on a container vessel. For this leg or service it is important to know the container id, the IMO number of the vessel, ETD, ETD, etc. All this information is static and primarily relevant for this leg. However, there is also process information, which occurs during the execution of the leg that must be reflected in the internal data model in order to represent the current stage of the process. This process information encompasses the current state of the leg or service (e.g. delayed, in time) and data collected during the execution, like ATA or ATD. In order to reflect this in the internal data model, an entity-centric modeling approach is followed, which combines static data and process information in a single unit – the Collaboration Objects.

In the previous deliverable we introduced this vision of the BCM and gave an overview about currently applied technologies for collaboration in the T&L domain in a first section. Subsequently to this, we elaborated the BCM's conceptual design in a first version. This architecture was the basis for an initial analysis for functional, technical and non-functional requirements, which are derived under consideration of the domain requirements. Based on these requirements we conducted a State-of-the-Art analysis. We primarily investigated approaches for data modeling and cloud-hosting. This revealed a suitable realization of the Collaboration Object concept by the usage of an entity-centric approach, as described above. The examination of cloud-hosting technologies and research projects showed potential approaches to realize a distributed, transparent and trustworthy cloud-based storage for Collaboration Objects. In a last section we investigated the proposed Generic Enablers from FIWARE and submitted additional request to realize the BCM's vision.

Formatted: Space After: 0 pt, Line spacing: single

[1] Please note that the modelling of Collaboration Objects is not finished yet. This is part of the detailed technical specification of the BCM, which will be finally delivered in Deliverable D5.5. The mentioned examples just reflect current ideas and approaches and are not necessarily represented in the final specification.

In this deliverable, we will build upon these results. Thereby, the refinement of the initial conceptual architecture in order to come to a stable version was our primary target. In combination with the work on conceptual architecture, we also focused on the refinement of previously defined requirements based on increased domain knowledge, better understanding of the collaboration of the FInest modules and experience exchange with domain partners. The latter was conducted with a joint venture between the WP5 ICT and domain partners. The scope was Use case 1, more precisely the challenge of 'resource coordination'. In detail, this means the exchange of resource availability, bookings and responses to them between stakeholders of this use case in a more efficient manner. The creation of the demonstrator led us to very valuable insights in the information exchange between logistic partners and influenced the update of the conceptual architecture. A specialized component was introduced by us, which aims at the more effective transmission of messages. During different meetings with the project partners of Use case 1 (in detail ARH, NCL and MRTK) we gained additional domain knowledge as well as clarification about the functionality of the FInest platform, which influenced the refinement process in a more indirect manner. In a last work stream we worked towards the realization of the Collaborations Object concept and gave with that a preview in the BCM's technical specification.

The remainder of this document is structured as follows: In a first chapter we will describe the refinement of the conceptual architecture and use this as basis for the update on the requirements in the following. Following to this we will provide a preliminary description of the approach to realize the vision of the Collaboration Objects and with this, we will give a preview on the first aspect of the technical specification. In the last chapter we will present the demonstrator for the resource coordination challenge in Use case 1. We describe the current as-is situation, introduce the demonstrator approach and derive the envisioned to-be scenario. In a subsequent chapter we

2. Conceptual Architecture Refinement

In this section we cover the description of the BCM's revised conceptual architecture. We updated it based on the refined conceptual architecture of the FInest platform (*cf.* Deliverable D3.2) in order to provide the necessary functionality, and integrated it in the overall infrastructure as desired. For a better understanding of the implemented changes we present in Figure 2 the refined architecture as well as include with Figure 1 the first draft (*cf.* Deliverable D5.1). In the following subsection we describe the changes in detail (descriptions of unchanged parts are omitted; *cf.* Deliverable D5.1 – Section 3). Afterwards, we investigate if and how the refined conceptual architecture is able to meet the general domain and use case requirements identified by WP1 and WP2.

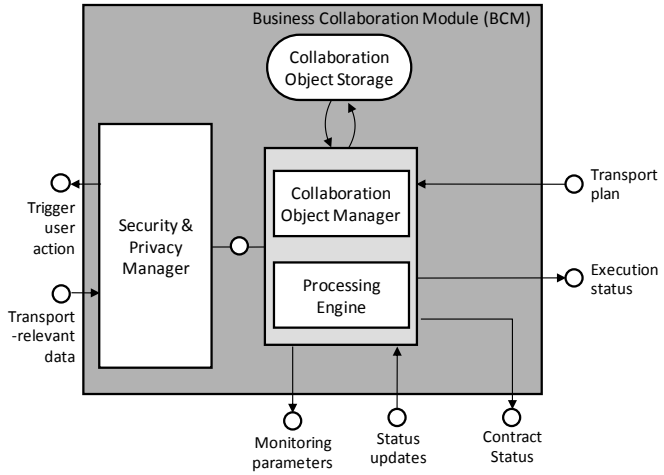


Figure 1 - First draft of conceptual architecture (*cf.* Deliverable D5.1)

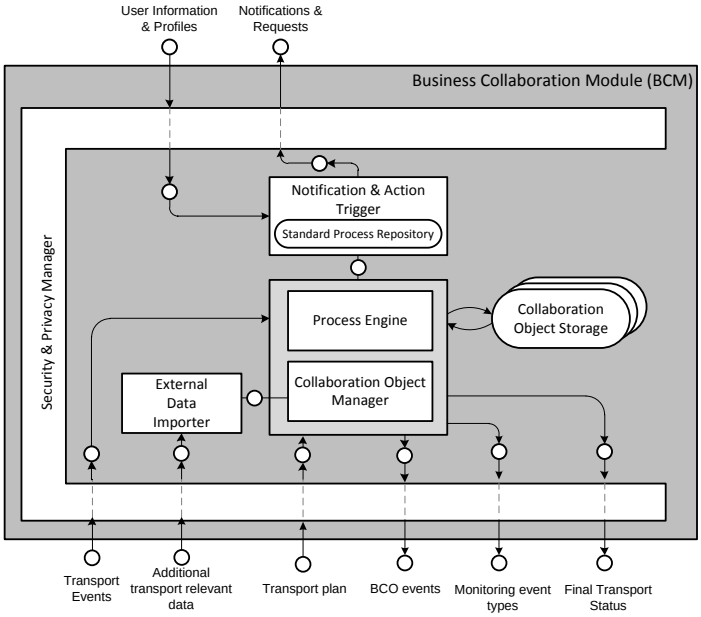


Figure 2 – Updated conceptual architecture

2.1. Refinement of Conceptual Architecture

In this section we present the refinements of the BCM's conceptual architecture. Each updated component will be described in a particular subsection.

2.1.1. Notification & Action Trigger

One of the central new active components (agent) of the BCM's architecture is the *Notification & Action Trigger* (short: Notification Trigger). It constantly observes ongoing transport processes, contained in the *Collaboration Object Manager* and updated by the *Processing Engine*. If an update occurs that requires notification and maybe an action, the Notification Trigger detects the corresponding users and determines the preferred communication channel for the message type. For this, the Notification Service uses *user & profile information* contained in the FInest platform and initiates the sending of the information.

For example, in a transport process organized by a forwarder it is necessary to inform the forwarder about every important process step and eventually, carriers, which are responsible for legs subsequent to the current one, but not the previous ones. The Notification Trigger determines the necessary users to inform and sends the event messages to them; thereby it is able to choose different communication channels for different message types. Error notification maybe send as SMS directly to the mobile phones of the users in order to ensure a short reaction time or even directly to the LSP's or LSC's backend system in order to inform the client about changes via their familiar system.

Additionally, the Notification Trigger is not just working in a reactive manner, but also detects the absence of required events of a transport process and therewith, is working in a proactive manner. For this, it verifies their current execution against predefined process definitions. These definitions are specific for special transport types – e.g., land, sea or air transport, but also the transport of dangerous goods or even to describe company specific processes – and define the milestones for each type. The *Standard Process Repository* stores these definitions and enables the Notification Trigger to load them if necessary. If a process will not reach a certain milestone in time, the Notification Trigger is able to detect this and notifies the corresponding user by using the same procedure as described above.

As an example for this behavior we consider a maritime transport of goods via a container vessel. The previously process definition for this transport type encompasses a rule which requires a port call latest 24 hours before the vessel reaches the port. If the processing engine will not receive a notification within this time box, the Notification Trigger will detect this and inform the ship captain or the shipping line's operator about the missing port call to trigger the necessary action to fix this. Examples from the other use cases are the notifications on certain points in time (noon reporting) in Use case 3 or the notification at certain waypoints in Use case 2.

2.1.2. External Data Importer

The other new agent within the BCM's conceptual architecture is the *External Data Importer*. Its overall purpose is to eliminate data format and model incompatibilities during the import

from heterogeneous external sources. For this, it provides a set of different adapters which map a certain data format to the internal data model of BCM. Hence, for every supported import format a specialized adapter has to be provided.

2.1.3. Security & Privacy Manager

The provided features of the *Security & Privacy Manager* remain the same compared to these described in the previous Deliverable D5.1. However, in the current conceptual architecture the in- or outbound communication have to go through this agent. The previous approach considered platform internal communication in general as safe and does not uses the security & privacy manager to verify the data. However, the envisioned deployment scenarios of the FInest platform also encompass a distributed application environment. In this case, but also in others, a general rule for security in distributed systems should to be applied: clients cannot be trusted. However, this rule usually refers to input data of clients; in case of the BCM it is also necessary to validate the receiver of information due to its high degree of confidentiality (vital business data).

2.1.4. Collaboration Object Storage

As the agent described in the previous section, the features and responsibilities of the *Collaboration Object Storage* remain the same. In addition to this, the current version of the conceptual architecture indicates that different storages can be combined in order to enable a distributed approach as demanded by requirement R403.

2.2. Gratification of Domain & Use Case Requirements by the BCM

In the section above we present the refinement of the BCM's conceptual architecture. In this section we investigate if this new architecture is able to gratify the domain and use case requirements identified by the Deliverables D1.3 (domain requirements) and D2.3 (use case requirements). This ensures that the technical realization of the BCM is able to meet these requirements and therewith, provides a solution for the identified problems of the domain.

Section 3.2 of Deliverable D3.2 presents an alignment of the domain and use case requirements to the technical work packages of the FInest project. Each of these work packages is represented by a so-called core module of the FInest platform. We investigated this alignment in order to validate if the current BCM architecture is able to provide the demanded features for this core module. We can conclude that the features, which are aligned to the BCM, are supported by the refined conceptual architecture and hence, a technical realization based on this concept is able to gratify these requirements. In detail we present this verification in Appendix B (on page 45). For this, we used the same structure and descriptions as D3.2 in Section 3.2. The provided table shows for each identified requirement a description, which component of the BCM's conceptual architecture is responsible for the demanded functionality.

In the next chapter, we use this conceptual architecture in order to identify requirements for the particular components of the conceptual architecture. This investigation is done under the consideration the general domain and use case requirements.

3. Refinement of Requirement Analysis

In this section we cover the refinement of requirements initially determined in Deliverable D5.1. These requirements are validated under consideration of better domain knowledge and understanding of demanded features. In particular, we used information from three different sources for this:

- Business requirements for future transport and logistics ICT solutions (*cf.* Deliverable D1.3)
- Detailed specification, identification of main challenges and proposal of potential solution ideas for the use case scenarios (*cf.* Deliverable D2.3)
- Influence factors from the updated conceptual architecture (cf. Section 2)

The first two bullet points are the manifestation of our better domain knowledge that we were able to gain during the last six months and the last point indicates that we set this knowledge into the context of the BCM.

As in Deliverable D5.1, we provide a subsection for each component of the BCM's conceptual architecture and present the requirements in three different categories: functional, technical and non-functional. New or updated requirements are tagged (with *updated* and *new*) and the position of the tag indicates what has been changed. Deleted requirements are explicitly mentioned within each subsection, together with the exact reason for their deletion.

3.1. Security & Privacy Manager

In Deliverable D5.1 we described the Security & Privacy Manager as the central unit that ensures the non-disclosure of confidential information. For this it uses authentication and authorization technologies as well as uses encryption to protect the stored data. However, security and privacy are cross cutting concerns over the whole FInest platform and therefore, it has been decided to provide the basic security functionality within the general FInest platform (*cf.* Deliverable D3.2 for details). The Security & Privacy Manager makes use of this functionality and provides the necessary logic to adapt it to the internal module architecture and requirements (it applies authentication and authorization at the level of *Collaboration Objects*). Hence, the basic requirements for this module remain the same, but the descriptions are updated to emphasize the usage and adaptation of external security functionality.

We deleted no requirements for the Security & Privacy Manager.

Table 1 - Refined Requirements Security & Privacy Manager

Functional Requirements

Identifier	Name	Description
R101	Access Control	Use security functionality provided by the FInest platform to enable a precise access control mechanism. *(Updated)*
R102	Restricted Access	Ensure that only authorized users and third party systems can access the stored data by the usage of security functionality provided by the FInest platform. *(Updated)*
R103	Secure Transfer	Use provided security functionality to ensure secure and reliable information transfer over the different communication channels. *(Updated)*
Technical Requirements		
Identifier	**Name**	**Description**
Rt101	Adapt Authorization Techniques *(Updated)*	Adapt provided authorization techniques to the internal data model of the BCM. Enable authorization at the level of Collaboration Objects and their features. *(Updated)*
Rt102	Adapt Authentication Techniques *(Updated)*	Adapt provided authentication techniques to the internal data model of the BCM. Enable authorization at the level of Collaboration Objects and their features. *(Updated)*
Rt103	End-to-End Encryption	Use appropriate protocols provided by the FInest platform to provide end-to-end encryption if data is delivered or received. *(Updated)*
Non-Functional Requirements		
Identifier	**Name**	**Description**
Rn101	Efficiency	Applied security mechanisms must not harm the systems performance or lead to high system loads.

3.2. Collaboration Object Manager

As we described in Deliverable D5.1, the Collaboration Manager is the central data managing unit of the BCM. It is concerned with the definition and implementation of a suitable data model (on the basis of the entity-centric approach as shown in Section 5 of D5.1), the instantiation of the model for concrete transport processes as well as the further management of the instances (loading, storing). We have updated the previously examined requirements in order to reflect this core functionality more precisely.

Our investigations have shown that there is a significant difference between the adaption of the data model during the design (adaptation) and the change of a model instance during the execution of the a transport process (agility). In order to reflect this change an additional requirement was introduced.

Additionally, a new requirement demands the use of industrial strength technologies to implement the vision of selected (entity-centric) modeling approach in a scalable and performant manner. Only technologies shall be used, which are applied in the industrial practice and provide the necessary performance within a large-scale business environment.

We deleted two different requirements due to the following reasons:

- R205 (Standardized Data Formats): This requirement is externalized to the new External Data Importer component because it is responsible for the alignment of heterogeneous data sources to the internal data model
- Rt201 (Modeling Approach): This requirement has no technical aspect. The demanded functionality is already covered by R201 and no further technical counterpart is needed.
- Rt202 (Flexible Model Definition): This requirement has no technical aspect. The demanded functionality is already covered by R203 and no further technical counterpart is needed.

Table 2 - Refined Requirements Collaboration Object Manager

Functional Requirements		
Identifier	**Name**	**Description**
R201	Appropriate Modeling Approach *(updated)*	Use a suitable modeling approach to enable the vision of Collaboration Objects. *(updated)*
R202	Completeness	Gather all (necessary) information related to a logistics process.
R203	Adaptability *(updated)*	Ensure easy adaptability (design time) of the used data model in order to ensure the compliance to different manifestations of logistics processes, e.g. for different

		companies. *(updated)*
R204	Data Management *(updated)*	Initialize the data model for one particular transport process; load and store information when it is necessary. *(updated)*
R206 *(new)*	Agility *(new)*	Provide an agile data model (runtime) that is able to integrate changes on logistics processes even at the execution (e.g., for unforeseen changes). *(new)*
R207 *(new)*	Ensure Data Quality *(new)*	The plausibility of data is an important aspect in logistics processes. This requirement ensures that the Collaboration Manager conducts basic validity checks during the instantiation of a model for a certain transport process.
Technical Requirements		
Identifier	**Name**	**Description**
Rt203	Industrial Strength *(updated)*	Use industrial strength technologies to enable an scalable and performant implementation of the modeling approach *(updated)*
Non-Functional Requirements		
Identifier	**Name**	**Description**
Rn201 *(new)*	Scalability & Performance *(new)*	Provide necessary scalability and performance to support large scale data sets *(new)*

3.3. Processing Engine

In contrast to the Collaboration Manager, the Processing Engine is not concerned with the definition of a (meta-)model for logistics processes, its instantiation and the further management of these instances. Its primary focus is the integration of external events (from other components, especially the EPM, or other potential sources), their interpretation and the integration of this information into the model instance for a logistics process. With this, the internal model instances are kept up-to-date.

Although we gave this description already in Deliverable D5.1, there were cross cutting concerns with the Collaboration Object Manager, which we deleted now:

- R302 (Process Completeness): Even by choosing an entity-centric modeling approach, which combines process and data in encapsulated entities, the Processing Engine is only concerned with updating the process information. The definition of the process model is covered by the Collaboration Object Manager.
- R303 (Process Change Support): We deleted this requirement for the same reason. The adaptation of the model is handled by Collaboration Object Manager.
- Rt301 (Process Definition and Execution): This is also conducted by the Collaboration Object Manager.

In addition to this, we introduced a requirement for industrial strength technologies in order to ensure performance and scalability of this component. See Section 3.2 for further details.

Table 3 – Refined Requirements Processing Engine

Functional Requirements		
Identifier	**Name**	**Description**
R301	Event Integration *(updated)*	Receive events from other FInest modules, especially the EPM, or external source, interpret these and update the internal data model accordingly in order to reflect the current state of logistics processes in the CO-based model of the BCM. *(updated)*
Technical Requirements		
Identifier	**Name**	**Description**
Rt302 *(new)*	Industrial Strength Implementation *(new)*	Use industrial strength technologies in order to provide a saleable and performant implementation. *(new)*
Non-Functional Requirements		
Identifier	**Name**	**Description**
Rn301 *(new)*	Scalability & Performance *(new)*	Provide necessary scalability and performance to support large scale data sets *(new)*

3.4. Collaboration Object Storage

As we outlined in Section 2.1.4, there were no major changes for the Collaboration Object Storage. This also can be seen within the different requirements, which remain basically the same. We introduced the requirement for the use of industrial strength technologies (Rt403) in order to ensure performance and scalability for this component, similar to the Collaboration Object Storage and the Processing Engine (see Section 3.2 for details). Additionally we deleted two requirements, however mainly due to duplications of concerns:

- R401 (Performance and Scalability): This requirement is also represented by Rt403 (Industrial Strength Technology). Due to the technical nature of the concern, we deleted this functional requirement.
- R405 (Suitable for CO-based Model): Same reason as for the deletion of R401.

Table 4 - Refined Requirements Collaboration Object Storage

Functional Requirements		
Identifier	**Name**	**Description**
R402	Integrated Security Management	Safe storage of obtained information by the provision of integrated security mechanism
R403	Distributed Storage	Implement the data storage in a distributed system. Enable the support for cloud-based storage.
R404	Transparency & Trust	Use of transparent and trustable storage infrastructure in order to enable user to determine how their probably confidential data is stored and managed.
Technical Requirements		
Identifier	**Name**	**Description**
Rt401	Encryption	Use encryption to safely store the gathered information, even in a distributed environment.
Rt402	Transport Security	Uses appropriate technologies to ensure the security and integrity of data within the distributed storage, especially during the data transfer.
Rt403	Industrial Strength Technology *(updated)*	Use industrial strength technologies that natively support the nature of the chosen modeling approach in order to enable sufficient scalability and performance. *(updated)*

Non-Functional Requirements		
Identifier	**Name**	**Description**
Rn401	Reliability	Failure safe system
Rn402	Data Integrity	Ensure consistency and integrity of stored data
Rn403	Scalability & Performance *(updated)*	Provide sufficiently scalability and performance (especially with regard to response and latency times) in order to support real-world business scenarios. *(updated)*

3.5. Notification & Action Trigger

The Notification Service is one of the new components within the conceptual architecture of the BCM. As we described in Section 2.1.1, it is basically concerned with two different but interrelated functions of the BCM: It analyses the execution of logistics processes and validates the current stage against information of an internal repository of standard process definitions. Deviates a process from the predefined definitions, it can be there is something going wrong and a responsible user of the FInest platform has to be informed. Additionally, it produces notifications at certain time intervals or that certain checkpoints within the execution of a transport process. Furthermore, this component selects the right communication channel for the notification message depending on user defined preferences.

Hence, these three different aspects are described by three different functional requirements. A technical requirement is determined due to the BCM will use the provided user management facilities of the FInest platform (*cf.* Deliverable D3.2).

Table 5 - Requirements Notification Service

Functional Requirements		
Identifier	**Name**	**Description**
R501	Process Analytics	Analyze model instances of logistics processes to determine the current process status. Check this status against predefined rule (stored in the Standard Process Repository) and trigger user notification if necessary.
R502	User Notification	Identify corresponding users for a detected situation (delay, arrival notification, etc.) and initialize the message transmission. Use the FInest user management facility for the user detection.

R503	Communication Channel Selection	Select the appropriate communication channel in consideration of the message/notification type (error, warning, information, etc.) and the user defined preferences. Use the FInest user management facility for this.
Technical Requirements		
Identifier	**Name**	**Description**
Rt501	User Management Integration	Integration of user management facilities of the FInest platform in order to gather all necessary profile information of a certain user.

3.6. External Data Importer

The External Data Importer is the second new component we introduced in Section 2. It is concerned with the alignment of data from external and probably heterogeneous sources to the internal data model of BCM. This can be achieved by a set of specialized adapters which map a certain data format to the internal data model.

Because the functionality of this component was externalized from the Collaboration Object Manager it also ‘inherits’ the requirement, which concerns the integration of external data. Further requirements are not defined.

Functional Requirements		
Identifier	**Name**	**Description**
R601	Support Standardized Data Formats	Currently, various standardized data formats are in charge in the T&L domain for data interchange. The most important of them have to be supported in order to enable communication of the BCM with existing legacy systems. Examples for this are EDIFACT or Cargo2000 but also existing ERP, SCM or CRM systems.

4. Towards Technical Design – Initial Draft of the Implementation Approach for Collaboration Objects

In this section we describe the realization of the Collaboration Object concept from a technical perspective. This encompasses the definition of a suitable meta-model and the consideration of

general implementation approaches under particular consideration of the requirement for the use of industrial strength technologies.

The representation of dynamic environments with changing scenarios requires a set of rules, according to which valid models can be created. Meta-modeling is about the development of such rules. In fact, it concerns the analysis and construction of schemas or types for a pre-defined class of problems. In order to realize the idea of collaboration objects it is necessary to develop such a meta-model for the transport and logistics domain with respect to the underlying entity-centric modeling approach. Although meta-models are already used in logistics, they mostly focus on the data, without regarding typical process structures, or they lead to inflexible, hardly manageable model instances. The envisioned solution makes use of these standards and enhances them with capabilities to realize the concept of entity-centric modeling and meet previously defined requirements visibility (enable global view), flexibility and scalability. This means, that we cover data and process modeling similar on an abstract level.

The basis for the development of the meta-model is established by existing models for transport process descriptions, standards and the requirements from the FInest use cases. We use the use cases also to verify the created meta-model to ensure it is suitable to describe different logistics processes. Subsequently, core types and their relations, which state the basic elements for the meta-model, are derived. The result is a (template) library of designated core types that is used for the creation of door-to-door transport models. These loosely coupled core types support variation of the global process and at the same time encapsulate behavior, which does not vary for different scenarios. This is mainly established by a componentized design and by constraints, which enforce the creation of valid process models.

The Object Management Group (OMG) standardized a meta-modeling architecture called Meta Object Facility (MOF)[2]. According to this, the conceptual solution design is located on the MOF M2 layer, whereby the representation of specific transport plans is located on the lower M1 layer. The benefits of meta-modeling in general are an improved understanding of the problem domain, efficient information management, and data integration. Furthermore, a flexible meta-model allows the creation of different scenarios, without being restricted to pre-defined process templates. This is intended with flexibility and re-usability of process models and therefore is a main characteristic of the solution approach. Additionally, the design provides adaption for customer demands.

Meta-models can be designed by use of different approaches, according to whether the focus is on processes or on the data. In entity-centric modeling, the focus is on the business relevant artifacts. As already described in Deliverable D5.1, entity-centric modeling is a promising approach and seems to be the golden-mean between process-centric modeling and pure data-centric modeling. Hence, this approach is utilized, but it is necessary to develop a specialized meta-model that incorporate the particular features and aspects of entity-centric modeling.

The entity-centric mindset is built of key elements, separated into four explicit, inter-related but separable dimensions [1]. First, each business entity has a data model, which contains all the information to ensure business operation. This may include business level information or meta-

Formatted: Space After: 0 pt, Line spacing: single

[2] http://www.omg.org/mof/

data as unique identifiers or entity states. Second, the (macro-level) life-cycle defines the stages a business entity passes during business operations. The third dimension reflects tasks or services. Like SOA, entity-centric approaches aim at flexible specification of distributed systems [1].A task or service encapsulates a unit of work meaningful to the whole business process and may be either fully automated or requires human interaction. Finally, associations define conditions under which tasks are invoked. Those constraints may include precedence relationships among the services, or constraints between services and events. The Guard-Stage-Milestone (GSM) meta-model, introduced by Hull in [2], is an approach for a declarative

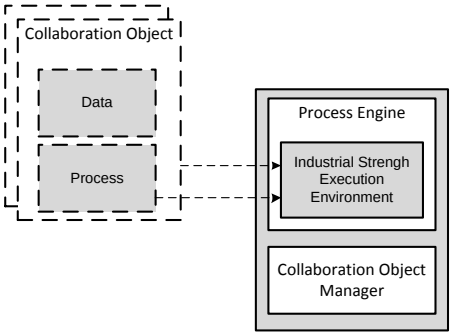


Figure 3 - Separate execution of process aspect by industrial strength execution environment

specification of life-cycles and interfaces. This meta-model provides a "language" and a graphical notation to specify business entity models. Although based on a conceptual level without any tooling support, the solution design is inspired by its concepts. The outcome is a set of entity types, which behave according to the guard-stage-milestone meta-model; and due to the incorporated data modeling it provides an end-to-end view on the process with access to required information at each time.

Previously, the development of entity core types, their encapsulated data and life-cycles, and the subsequent instantiation process was explained. This makes the major part of the solution approach. Now, the envisioned solution with focus on model execution is described. The meta-model is designed by means of an entity-centric modeling approach and is inspired by GSM. However, there is no "battle-proven" GSM runtime engine available. Although, the ACSI project[3] is currently developing with Barcelona a prototype that directly supports the modeling and execution of GSM based models, it cannot be compared to technologies that are applied in the industrial practice for years. We assume that established technologies provide a higher degree of performance optimization, so that are able to handle large data sets as expected for the BCM. Therefore, the model must be based on existing technologies that allow its execution on an industrial strength environment.

For the entity life-cycle this has the following impact. Its design must be modular and flexible, because the internal processes have to be delegated to a scalable and performant execution environment. This execution environment is managed by the *Processing Engine* of the BCM and enables the use of established industrial strength technologies. This requires that the process

[3] http://www.acsi-project.eu/

Formatted: Space After: 0 pt, Line spacing: single

aspect of every *Collaboration Object* is detached and handed to processing engine separately (see Figure 3). The *Collaboration Object Manager* provides the necessary logic to bind process and data part of each collaboration object together. By this, the BCM is able to work with the entity-centric modeling approach on the logical level and make this concept available for other modules of the FInest platform. However, the technical execution is realized by established technologies, which allows us to be compliant to the requirements regarding the performance and scalability. We mentioned this concept already in Deliverable D5.1 and refer again to [3], which provides an example implementation for it based on SAP's Netweaver BPMN engine as industrial strength execution environment.

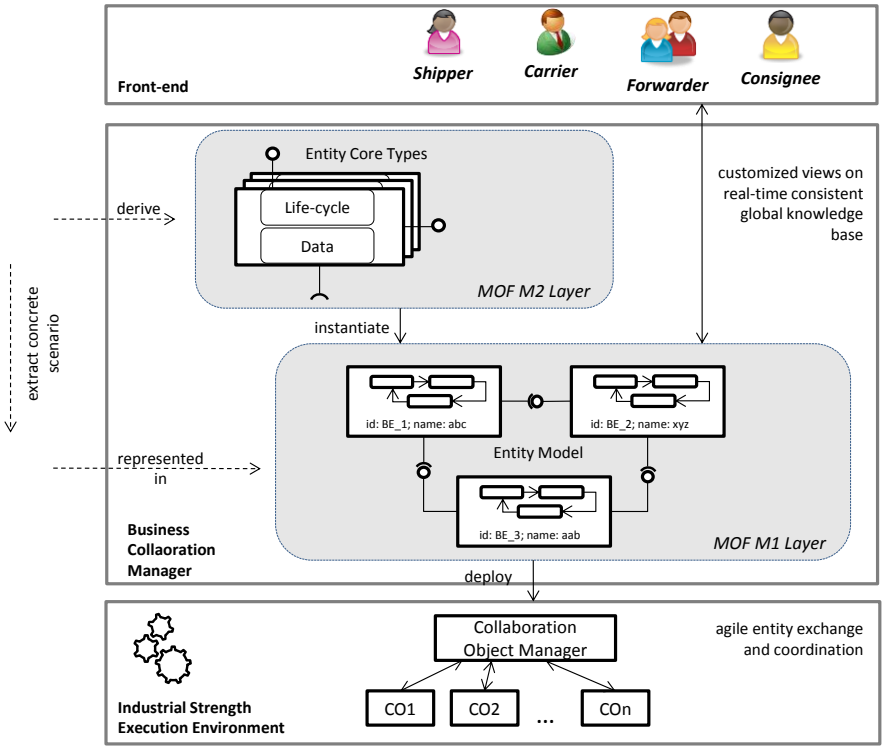


Figure 4 - Modeling, instantiation and execution of collaboration objects

Although internal life-cycles are encapsulated within a business entity and delegated for execution, they influence each other by their states and events. Moreover, the involved stakeholders require a global view on the process and might request certain entities at the same time. The encapsulated life-cycles therefore must be put together to a holistic process. Besides entity coordination, monitoring and serialization is provided during runtime. In the end, the underlying loosely coupled architecture simplifies entity exchange in case of derivations at runtime.

Figure 4 summarizes the presented concept, which allows us to provide an entity-centric based approach for meta-modeling in combination with industrial strength technologies ensures agility, flexibility, and scalability.

5. Resource Coordination Demonstrator

We present in this section the demonstrator that addresses the resource coordination challenge of Use case 1. This demonstrator was elaborated by joint efforts between domain, especially ARH, and ICT partners of WP5. The primary aim of it is the presentation of an approach to ease the coordination of resources in a maritime transportation scenario (as Use case 1) by optimize the visibility of services (port and terminal services to a ship calling at the port) and the necessary communication for booking such services. Thereby, we focus on the results of this specific demonstrator. For a more general introduction to the demonstrator, their goals and used creation procedure please refer Deliverable D3.2.

In the first section we describe the addressed challenges and the scope of the demonstrator more in detail. Based on this, we illustrate these challenges by an as-is scenario description (current practices) and present the approach of the demonstrator afterwards. Together with this approach we present the envisioned to-be scenario. A detailed description of the particular screens of the demonstrator is provided in Appendix A (on page 31).

Note: This demonstrator is, as the demonstrators implemented by WP6-8, online available under the following URL:

http://proto.finest-ppp.eu:8080/eu.finest-ppp.project.mockup/

However, this area is password protected and the login credentials will not be provided in this document (public document). Please approach the main author of this document directly if you need access.

5.1. Addressed Challenges and Scope

As described by Deliverable D3.2 the FInest team elaborated four different demonstrators to discover sweet spots for new applications enabled by the project. They give a first impression how the provided infrastructure enables the combination of existing legacy and new tools, regardless of whether developed within the project or not. Each of these demonstrators is based on identified use case challenges determined by WP2. The Resource Coordination Demonstrator addresses two challenges from Use case 1 (described in detail in D2.3):

- **Terminal Planning**: As in many logistics processes, also the distribution of information between other stakeholders and the terminal is done in a manual manner (despite the use of EDI messages, much manual work remains to register information). This often leads to late notification and therewith to rush work as well as the waste of resources. For example, the workload for T&F depends on the number of vessels which has to be charged and discharged, the quantity of cargo, but most importantly the moment of arrival of the ship and the time available to conduct the cargo handling operations. In some cases T&F have to adjust their manpower to the changing requirements hire additional day-workers if necessary. However, the employment of these has to be well planned and a late notification about a changed ship schedule and late communication of discharge / load list cause a lot of trouble for the terminal.
- **Resource Coordination**: A port is above all a service provider. Services are provided to various customers, such as shipping lines, ship agent or vessel captains (e.g. quay

allocation, mooring/unmooring services, fresh water, electric supply or waste disposal for vessels), terminals (e.g., use of the port infrastructure such as quays) or others (e.g. pilots or captain need current weather information). For this it is necessary to publish information on capacity and services delivered, as well as available resources, in order to enable the port users to book services (and in the case of the ship, prepare its voyage and port call). However, currently applied systems lack this visibility and usually depend on manual information distribution. For example, ARH receives the majority of port calls (service requests) via email/fax/phone, which is the only point in time when the customer gets the opportunity to check for the availability of a quay or tug boat during the booking process.

These two challenges were combined to the challenge "Coordination of Resources at Port and Terminal", addressed in the *to-be scenario 2* described in D2.3. The demonstrator is addressing this challenge at hands of a scenario based on vessel calls. It encompasses three different stakeholders, an agent of a shipping line (NCL), a port administration (ARH) and a company operation at the terminal (T&F). Every stakeholder is represented by a domain partner of the FInest project.

5.2. As-is Description

This section focuses - in contrast to the previous one - not on the general description of addressed challenges but rather at an as-is description of the demonstrator's underlying scenario.

The demonstrator considers three different actors (*cf.* Deliverable D2.3 for more details): *Logistics Service Provider: Shipping Line* (role executed by the Port Control), *Logistics Service Provider: Port* and *Logistics Service Provider*[4]*: Terminal* (role executed by the Terminal Operator). The following describes the different responsibilities of each role within the conduction of a vessel call and the subsequent booking of port and terminal services. For each role a separated subsection is provided.

5.2.1. LSP: Shipping Line

The underlying scenario of the demonstrator assumes a shipping line that owns a set of ships which are conducting regular voyages based on predefined routes. NCL, a domain partner of the FInest project, operates in this fashion. However, this assumption does not limit the demonstrator, since the crucial point is that a shipping line sales space on their ships and this model is valid for the majority of sea-based shipping companies.

As a first step in the scenario, the shipping line has to create and publish schedules for their ships. Customers of the shipping line, e.g. shippers or agents, can use these schedules to plan their shipments and book capacities on certain vessel voyages. Nowadays, it is common to publish the current schedules on the website of the shipping line, but there are also printed

[4] LSP

Formatted: Space After: 0 pt, Line spacing: single

registers for ship arrivals and departures in certain ports. But in any case the customers have to inspect the schedules manually.

After a customer has found a suitable vessel and voyage, he/she sends a booking request to the shipping line. In most of the cases these requests are transmitted by email, fax or phone and have to be integrated manually in the system of the shipping line. The scenario assumes that the shipping line is using a specialized tool suite in order to manage their vessels, bookings and ongoing voyages. These systems are mostly closed entities, but can also provide an external interface which allows privileged partners to transmit data. An example for such a system is Softship[5] used by the domain partner NCL. It provides an EDI interface, which allows agents of the shipping line to make bookings for customers, for example, KN is working as such an agent for NCL.

The next step is that the shipping line is in charge to inspect these and send a confirmation or rejection back to the customer. This information exchange is also done mostly via email, fax or phone and therefore, has to be done manually. When the shipping line closes the booking period for a certain ship and voyage, the various documents necessary for the different involved parties during the voyage are completed. The creation of these documents is in most cases assisted by the used tool suite (e.g. Softship) and the information provided by these systems. The documents encompass operation plan (for one ship / voyage), stowage plans (built in collaboration between the ship and the shipping line), load and discharge lists for every quay/terminal of the voyage. Information related to as the vessel (like crew lists and security certificates) are handled by the ship itself. When this planning phase for ship voyages is completed, the shipping line has to take care of a proper execution of the transport process. For this it is necessary that for every port a ship will visit during its voyage, a port call (vessel call) will be announced. This port call encompasses information about the ship visiting the port, data and schedule, but also mandatory security-related information like last ports visited, information about dangerous goods, crew list, etc., and finally information about services that the ship needs from the port and at the terminal. A port call provides vital information for the port, which is needed to plan the resources (e.g. quay or tug boats). But also for the terminal this information is necessary because it provides an updated time frame of the ship arrival and maybe forces an updated service plan (e.g. new crane usage plan for the day or hiring of more workers for loading and discharging). Nowadays, especially the late notification of such port calls causes a lot of problems due to it can lead to rush work or waste of resources.

5.2.2. LSP: Terminal

The second role in the scenario is represented by the terminal. Its services are booked by the shipping line to implement the discharging and loading of cargo at a certain quay in a port. Furthermore, the terminal also offers storage capacities which can be used to temporarily store cargo under defined conditions (e.g. under -18°C) (storage services booked by cargo owner or cargo agent). Also the necessary interaction with oversea shipping lines, which own the containers, is realized at the terminal.

Formatted: Space After: 0 pt, Line spacing: single

[5] http://www.softship.com/

Currently, the terminal describes their services on their website, where potential customers can check for this information and book certain services. Special bookings are mostly transferred via fax, email or phone. However, it is not necessary to book the discharge of every container manually. This relies on the previously created discharge and load list and these can be transferred in an electronically way. For example, T&F receives EDI messages from Softship system used by NCL about cargo to be discharged and loaded (COPRAR[6]).

The biggest challenge in this process is the preparation for unloading / loading operations and the reception of precise information, which is directly depending on receiving the correct and final discharge and load list in due time. Due to the lack of visibility during the transport process the terminal often have to react quickly. Due to mostly manual propagation of information the consequence is that the terminal gets only late notifications of changed bookings or delayed ships. This leads to unnecessary high resource usage and maybe incomplete loading.

5.2.3. LSP: Port

The last role within the demonstrator scenario is the port. It is the local authority who must control all traffic at the port, the owner of the quay infrastructure and it leases this infrastructure to different customers (e.g. the terminal services provider). This infrastructure usually encompasses a sensor network, the AIS system, which allows the port to determine ship positions and this is combined with other information (e.g., weather information) and shared with customers. Hence, the port also offers information services to its customers.

As already mentioned, the biggest problem for the port currently is the information propagation about the availability of its services. Most of the service bookings, usually done via port calls, reach the port in an unstructured and not machine processable way (e.g., email, fax or phone). The received information is manually entered in specialized systems for the port administration. An example for such a system is PortWin[7], used by ARH. Obviously, the port shares the drawback of manual communication handling with the other user roles within the scenario, but this is not the main problem.

The SafeSeaNet[8] network allows the transmission of port call information already today and more and more port calls are done by using this system. However, the main problem for the port remains: this information is locked into a local data silo. The currently used system is not able to publish to the received information, so that customers can get an instant overview about the provided services (by the port) at a certain point in time and not all ships use SSN (it is only mandatory for certain types or sizes). Furthermore, SSN focuses on security related information, but does not cover commercial information. This also denies the check for free capacities and therewith hampers an effective planning.

Formatted: Space After: 0 pt, Line spacing: single

[6] http://www.unece.org/trade/untdid/d01a/trmd/coprar_c.htm

[7] http://www.seamless.no/index.php?lang=19http://www.seamless.no/index.php?lang=19

[8] http://www.emsa.europa.eu/operations/maritime-surveillance/safeseanet.html

5.2.4. Demonstrator Approach & To-be Scenario Description

In the previous section we described the three different user roles in the demonstrator's scenario and how they currently implementing the process of booking port and terminal services for ships. In this section we will describe the approach of the demonstrator to address the described problems.

The basic idea of the demonstrator is to provide a portal as single point of access for every stakeholder of a logistics process. This portal integrates information necessary for the conduction of logistics processes and presents it to the right stakeholder. It also works as an information interchange platform, which allows each user of the portal to directly communicate with other users in real-time. This communication can be realized by sending text messages, comparable to emails, but also by sending structured information which can directly be integrated in the interface of other portal users. An example for the latter case could be a port service booking request, which can be processed by the portal and integrated in the UI of the receiver (e.g., the port LSP) without any further manual intervene. By the integration of existing legacy systems (e.g., Softship or PortWin) continuous use of these systems is allowed. The portal makes vital information from these systems available for all other portal users and therewith, unlocks existing data silos. The integration of information originating from the portal back to the legacy systems allows the transparent use of the platform.

Furthermore, the portal enables logistics service providers to describe their services in a structured and machine processable manner and makes the descriptions available over the platform. This allows subsequent service search by customers in order to ease the planning of transport processes. By the integration of up-to-date information in such service description the providers are able to publish their current level of resource usage and (resource) availability. The integration can be done via a connection to backend systems of the provider, so that no manual effort is necessary to maintain the information. By this the portal enables its users to implement effective resource coordination.

By this approach a much better visibility is achieved. Information that is available for one user of the portal can easily be shared with other users in real-time. This enables a much better planning for all involved stakeholders (e.g., the terminal) since unforeseen changes in the process can be communicated more efficiently. For example, a vessel delay can be indicated by the shipping line or the captain and instantly transmitted to the port and terminal or by AIS information. A service provider uses the portal to increase the visibility of its services and uses real-time information to publish the current state of its services (e.g., cancellation of one service makes it instantly available for other customers). This increases the effectiveness of resource coordination. Additionally, the amount of manual information processing and distribution is reduced which is beneficial for the overall effectiveness within logistics processes.

In Appendix A – Detailed Demonstrator Description (on page 31) we provide a detailed explanation of the demonstrator[9]. We describe, based on the demonstrator description template

Formatted: Space After: 0 pt, Line spacing: single

[9] The envisioned operations and interactions among various parties enabled by this portal are also described in the to-be scenario 2 presented in D2.3

and at hands of screenshots, the functionality and the envisioned features of the previously described portal for effective resource coordination and (terminal) service planning.

6. Conclusion

In this document we provide the refinement of the BCM's conceptual architecture and the functional, technical and non-functional requirements for this FInest module. Both were initially elaborated by Deliverable D5.1 and with the conducted refinement, these parts reached a stable version and can be used as basis for the subsequent technical specification of the BCM. We also described the demonstrator that was created in cooperation with domain partners of this work package. Additionally, we presented a preliminary approach to realize the concept of Collaboration Objects as a first step towards the technical specification of the BCM.

After the introduction, in which we summarized the results of the last deliverable and described the activities since the last milestone, in Chapter 1, we presented the revised version of the BCM's conceptual architecture in Chapter 2. In consideration of this architecture and our improved domain knowledge, we conducted the refinement of the requirements for the BCM in Chapter 3. Afterwards, in Chapter 4 we presented our initial approach to realize the vision of Collaboration Objects. In the last chapter we presented the 'Resource Coordination Demonstrator'.

With the results presented in this document, we will be able to continue our work on the technical specification of the BCM. As mentioned earlier, the refined conceptual architecture and the refined requirements will work as the basis for this. Until the next milestone we have to elaborate a suitable meta-model for the Collaboration Objects (core types, *cf.* Section 4), provide a concrete approach to implement the vision of these object and define an execution environment based on applied technologies. Furthermore, we have to clarify the data models, which are exchanged between the FInest modules, which leads to a precise definition of the BCM's interfaces. Last but not least, it is necessary that we develop a mechanism that allows us to subscribe to special event types in receive updates for these events in order to update the internal data model.

7. References

[1] R. Hull, "Artifact-centric business process models: Brief survey of research results and challenges," *On the Move to Meaningful Internet Systems: OTM 2008*, pp. 1152–1163, 2008.

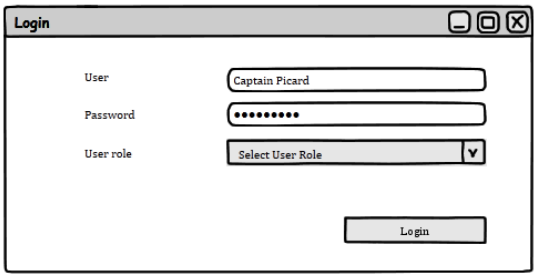
[2] R. Hull et al., "Introducing the guard-stage-milestone approach for specifying business entity lifecycles," *Web Services and Formal Methods*, no. 257593, pp. 1–24, 2010.

[3] K. Erler, "Design and prototypical Implementation of an Entity-centric Collaboration Environment," Dresden University of Technology, 2011.

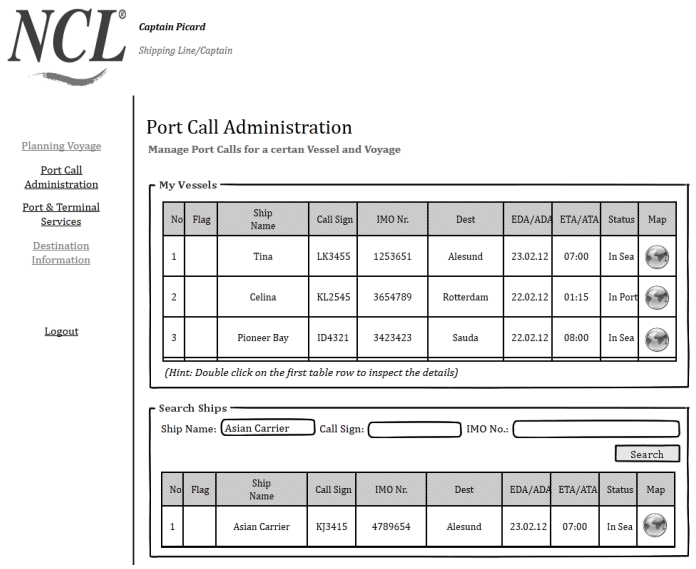
8. Appendix A – Detailed Demonstrator Description

General Information	
Element	*Description*
Title	Demonstrator 2
As-is scenario	Use case 1 describes the shipment of dried/frozen fish from Norway to Brazil. When looking at the feeder part of the shipment, this process involves different stakeholders: the fish producer, the shipping line, the terminal company and the port. The current coordination between these different parties currently requires a lot of manual effort due to the fact that mainly fax/phone/email is used for communication. Even when modern telecommunications systems are available (e.g. SafeSeaNet), the information is locked into local data silos where it cannot be accessed by other involved parties. In detail, Use case 1 describes a scenario in which a ship has to make a port call and book services from the port and from the terminal company. Due to delays in this process it is often hard to plan the available resources and service precisely. The fact that only the directly addressed party receives real time information reinforces the problem. For example, a shipping agent of a ship makes a port call for a certain port. Currently, this information is locked at the port and is not propagated efficiently enough to the subsequent terminal operator, so that he/she cannot update the service schedule.
Addressed main challenge	Resource coordination (between port, terminal and shipping line or ship captain in order to organize a port call – *cf.* Usecase 1 for further details)
To-be scenario	Envisioned in the demonstrator is that the involved parties are able to share information in real-time. Each partner reports the necessary information to the Finest portal outlined by the demonstrator and the portal distributes it to all other stakeholders which depend on this. For example, a port call will no longer be transmitted directly / exclusively to the port, but rather to the FInest portal which send this information also to the terminal and the authorities in order to make precise and more reliable service schedules. Additionally, the portal idea allows a better communication between the parties by providing and uniform channel and established new sales channels for services (promote and book services directly through the portal)

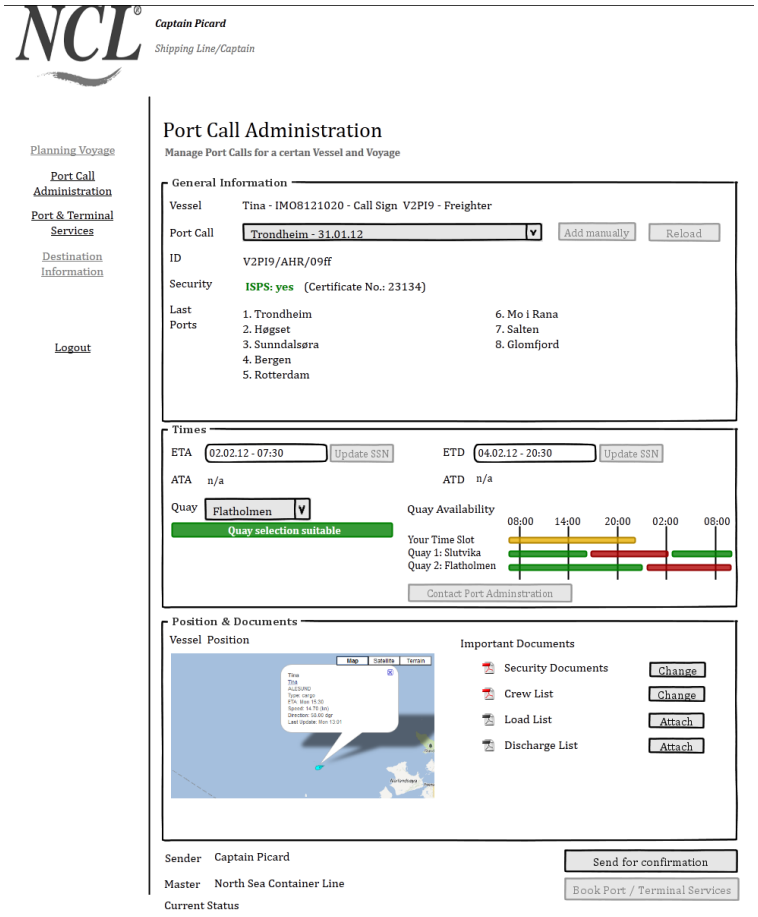
Demonstrator approach	The demonstrator outlines a 'Resource Hub' that is able to coordinate the available resource and the necessary services for a certain ship. It propagates information as soon it is available to all involved parties and therewith facilitates the effective use of resources and the accurate planning of services. Additionally, the demonstrator utilizes the idea of a commination portal that allows the partners to interact more quickly.
Involved modules	BCM, EPM, TPM
Lead	René Fleischhauer (SAP)
Involved partners	ARH, MRTK, NCL, T&F, SAP

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Login
Screenshot	
Screen purpose	This screen does not provide real authentication functionality, but rather lets the user select a certain role for the demonstrator. User name and password are ignored. The screen allows the selection of different roles to see the cooperation of the different stakeholder in the scenario.

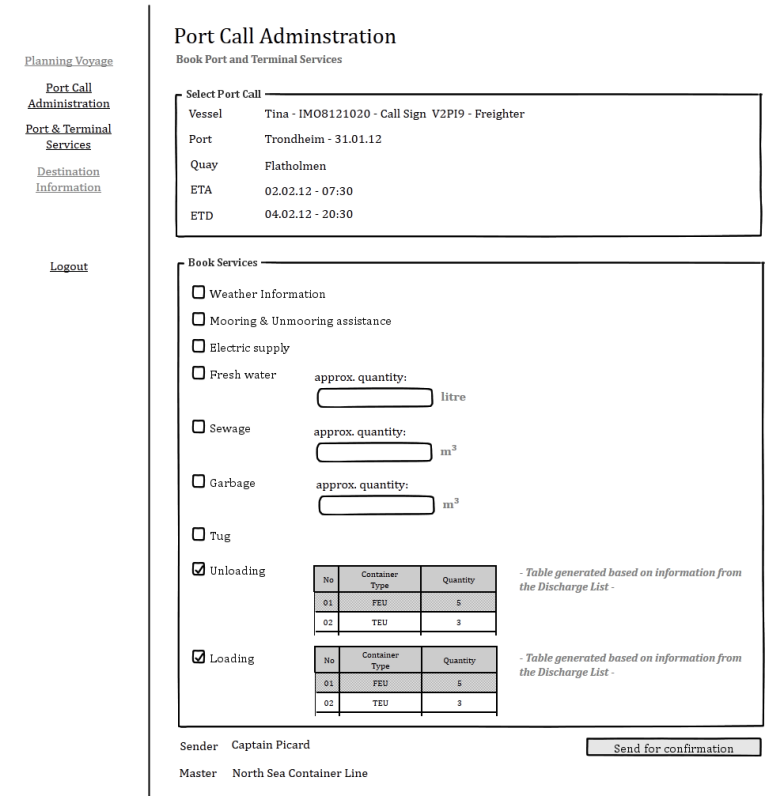
Detailed description	• User: - *ignored* -
	• Password: - *ignored* –
	• User role: Let the user select one of the three different user roles for the demonstrator
	  ○ Logistics Service Client (Carrier/Ship)
	  ○ Logistics Service Provider (Terminal)
	  ○ Logistics Service Provider (Port)
	• Login: Switches to the first screen of the selected user role

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Client (Carrier/Ship) -> Port Call Administration (1)
Screenshot	
Screen purpose	This is the first screen of the port call administration facility of the demonstrator. It allows shipping line operator or a vessel captain to select a ship, which is currently on a voyage, see its route and submit port calls for the different ports of the voyage. Additionally, the 'Search Ships' group allows the user to search within a ship register in order to get detailed information about foreign ships.

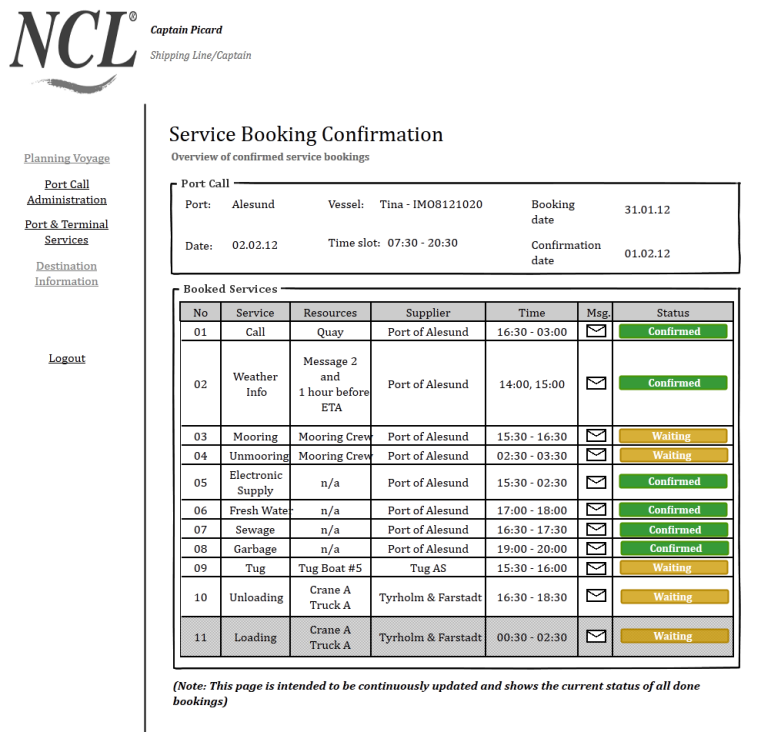
Detailed description	• My Vessels: Displays all ships of the line, which are currently on a voyage in a table
	  ○ Hovering the map symbols displays the current ship position
	  ○ **Double click on the first row will jump to the next page**
	• Search Ships: Allows the user to search for foreign ships (not animated!)

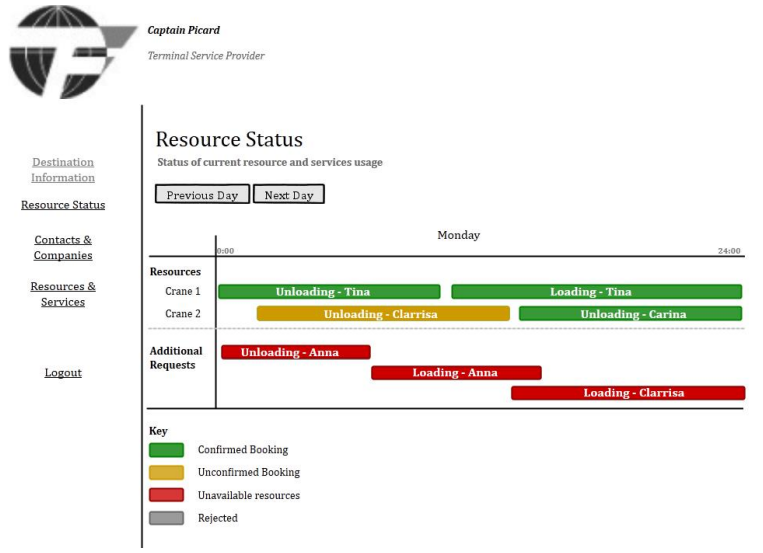
Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Client (Carrier/Ship) -> Port Call Administration (2)
Screenshot	
Screen purpose	Make a port call for specific vessel on a voyage. This information is transmitted to the FInest portal and available for all other stakeholders involved in the process.

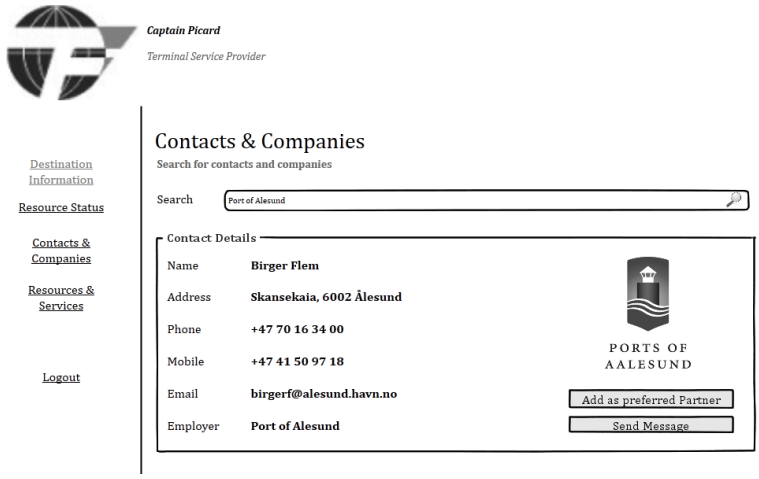
Detailed description	• Vessel: Name, IMO, call sign and type of the ship • Port Call: Combo box to select a particular port of the ship's voyage to create a port call; the demonstrator indicates the loading of this information from existing backend systems. • ID: Internal ship ID • Security: Availability of ISPS certifications and its number • Last ports: the 10 last ports of the vessel • ETA, ETD: Originally calculated arrival and departure times loaded from the backend or SSN; option to update these times manually and send the information back to SafeSeaNet • ATA, ATD: Real-time information about the actual arrival and departure times • Quay: All quays available in the selected port; a Gantt chart on the right beside it, displays the current schedule of the quay and the own time frame; a message indicates whether the quay selection is suitable or not. • Vessel Position: Displays the current position of the vessel within a map • Cargo: detail for the cargo on the ship which should be discharged at the port • Important Documents: List of important documents for the vessel, which are needed within the port (discharge and load lists, crew lists and other security documents); provides the option to upload the documents to make them directly available • Send for confirmation: Sends the previously entered data to the FInest platform, which forwards it to responsible user in order to process the request • Book Port / Terminal Services: After sending the port call to the system this button will become available; it directs to a new screen which allows the booking of services for the vessel within the port and terminal

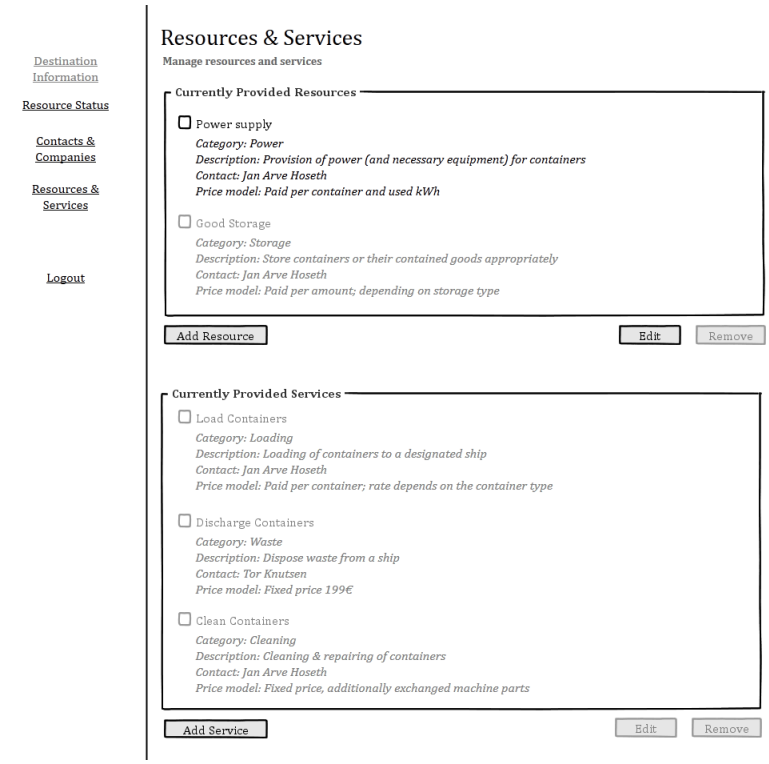
Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Client (Carrier/Ship) -> Port Call Administration (3)
Screenshot	
Screen purpose	This screen allows the booking of certain services at the port and terminal, after the users sent the port call request and his/her quay request was confirmed. The necessary data is provided by information port and terminal and is submitted beforehand (see *Logistics Service Client (Terminal) -> Resources & Services*). However, this intended functionality is only outlined and not implemented.
Detailed description	• Select a Port Call: Summarizes the relevant data of the previously submitted port call • Book Services: Allows the user to select desired services in the target port • Send for confirmation: Sends the booking request to the respective

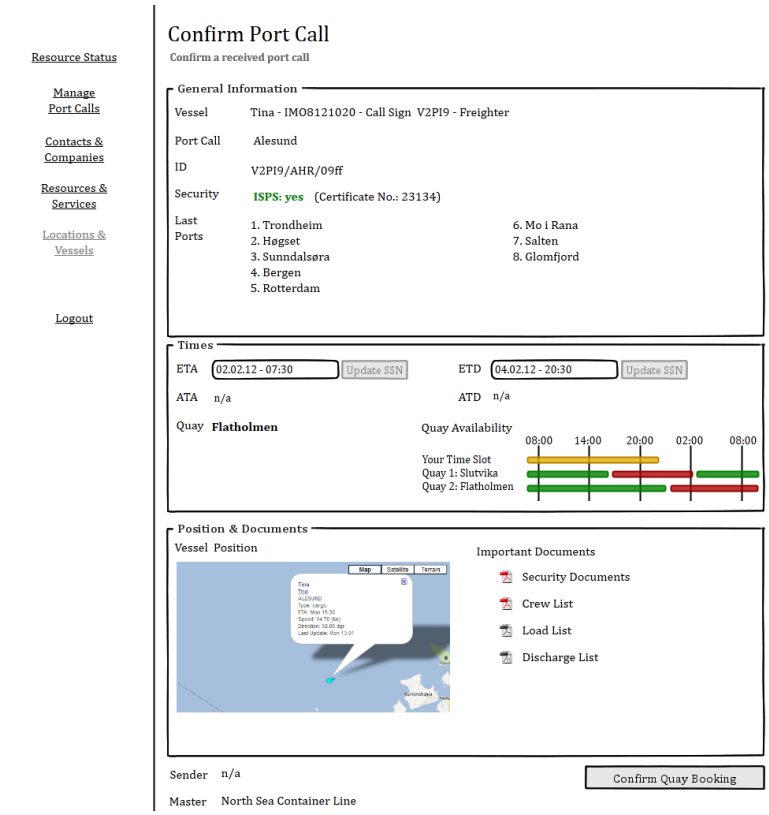
	service providers for verification.

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Client (Carrier/Ship) -> Service Booking Confirmation
Screenshot	
Screen purpose	After the user submitted the service booking requests, he/she will get forwarded to this screen. It summarizes the status of the different services bookings. A status can be 'Waiting', 'Confirmed' or 'Canceled'. This screen is also available through the menu entry 'Port & Terminal Service' in order to enable user to check the status of the booking at a later point in time.

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Provider (Terminal) -> Resource Status
Screenshot	
Screen purpose	This screen presents the incoming service requests and gives the ability to assign these requests to provided resources of the user's company. In the demonstrator, the terminal will get loading or discharging service requests and has to determine the availability of cranes within the terminal. The screen uses a Gantt-diagram to provide a suitable visualization of the current resource consumption and allows the user to accept or decline service requests.
Detailed description	Accept/Decline request: Hover with the mouse the bars of the chart to get a popup window to accept or decline the service request.

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Provider (Terminal) -> Contracts & Companies
Screenshot	
Screen purpose	This screen allows the user to search for specific partners in the outlined demonstrator portal. By this, the user is able to get additional information about other potential business partners and can directly contact them.

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Provider (Terminal) -> Resources & Services
Screenshot	
Screen purpose	This screen allows the terminal (or port) to add resources and services that can be booked by logistics service users and acts as data basis for the booking of terminal and port services in the demonstrator scenario (see *Shipping Line/Captain -> Port Call Administration (3)*)
Detailed description	Add Resource: Allows the user to a certain resource Edit (resource): Allows the user to edit a previously created resource Add Service: Allows the user to add a certain service Edit (service): Allows the user to edit a previously created service

Detailed Information	
Element	*Description*
Demonstrator title	Demonstrator 2
Screen name	Logistics Service Provider (Port) -> Confirm Port Call
Screenshot	
Screen purpose	*Note:* The port and the terminal are both service provides and hence, they require similar functionalities within the demonstrator. The screens *'Resource Status'*, *'Contracts & Companies'* and *'Resource Services'* are in most parts identical to those of the *Terminal Service Provider*. The most important resource for the port is its quays and their current usage. Thus, the demonstrator shows this within the '*Resources & Services*' screen of the LSP: Port. However the functionality is identically to the role LSP: Terminal and hence, we omit a detailed description here. The Port Call Confirmation screen provides the port with the ability to inspect incoming port calls and quay booking request. The presented information is identical to *Shipping Line/Captain -> Port Call*

	Administration (2) with the difference that the port only has to confirm (or decline) the quay booking. This is the counterpart to the 'Send for Confirmation' button in the *Shipping Line/Captain -> Port Call Administration* (2) screen.
Detailed description	Confirm Quay Booking: Confirm the quay booking for the selected port call.

9. Appendix B – Gratification of Domain & Use Case Requirements by the BCM

Table 6 - Gratification of Domain & Use Case Requirements by the BCM (detailed overview)

No	Demanded ICT Support (copied from D3.2 Section 3.2.2)	Demanded BCM Functionality (copied from D3.2 Section 3.2.2)	Responsible BCM Components
Requirements Fulfillment Analysis – General			
17	**Single information source** for all parties, **less one-to-one communication** *(UC2-2) (UC3-2) (UC1-3)*	1. Defines 'expected process' and controls its execution, sends notification on status updates, sends 'reminders' when inputs / actions are expected 2. Provide links / copies of transport documents	*1.* Notification & Action Trigger and Standard Process Repository (cf. Section 2.1): The repository stores description of the 'expected processes' and compares the current execution against it. In case of deviations, the user will be notified. *2.* Collaboration Object Manager and Collaboration Object Storage (cf. Section 2.4): Transport documents are static data and as such encapsulated by the Collaboration Objects. The corresponding storage persists this data and enables access if necessary.
Requirements Fulfilment Analysis – Planning Category			
5	A "**booking" portal for spot buying**: linked to market place, enable quotations / benchmarking, reservation of capacity, sending booking requests to service providers, sending booking confirmation to customer, and monitoring of	1. Contract usage -> ECM (informed via BCM)	1. Process Engine, Collaboration Object Manager and Collaboration Object Storage: These components observe the execution of the transport process and update the internal data model. The final status of each transport process is transmitted to the ECM after its end and is used to determine the contract usage (*cf.* Interface "Final Transport Status – Section 2)

	booking status. *(UC1-1) (UC3-3)*		
21	A "**deviation management**" system for **automatic updates in plan** based on reception of deviation signals configured as triggers for re-planning (event-triggered re-planning) *(UC2-4) (UC3-4)*	1. Re-planning triggered by either a. By events / alerts (**EPM -> BCM -> Front-End**) b. Status updates & 'proposal for re-planning' in **BCM**	1. Notification & Action Trigger (*cf.* Section 2.1): This component observes the execution of the transport process. a. The BCM informs the TPM about the necessity of the a re-planning via the "BCO events" interface, if user actions or EPM indicate that a re-planning shall occur b. If the incoming events indicate that a re-planning is recommended, the component sends notifications to the user for a final judgment.
Requirements Fulfillment Analysis – Collaboration Category			
5	A "**booking" portal for spot buying**: linked to market place, enable quotations / benchmarking, reservation of capacity, sending booking requests to service providers, sending booking confirmation to customer, and monitoring of booking status. *(UC1-1) (UC3-3)*	1. Contract usage -> ECM (informed via BCM)	1. Process Engine, Collaboration Object Manager and Collaboration Object Storage: These components observe the execution of the transport process and update the internal data model. The final status of each transport process is transmitted to the ECM after its end and is used to determine the contract usage (*cf.* Interface "Final Transport Status – Section 2)
7	A system for **automated contact-based control**: calculate KPIs for comparing agreed performance (criteria from contracts) with actual	1. Enablement: information for this is provided in FInest Platform (BCM for closed logistics processes)	1. Collaboration Object Storage (*cf.* Section 2.4): The storage persists the data of transport processes and establishes the bases for the analysis of historical data. Subsequent business routines can analyze the saved data of previous transport processes to calculate the necessary

	performance for the transport plan), and historic data recording. *(UC3-3)*		KPIs.
11	Collaboration platform for **fast communication** regarding planning and re-planning and new input for the event manager *(UC2-4) (UC3-4) (UC1-1) (UC1-2)*	1. Notify involved users + trigger actions in Front-End	1. Notification & Action Trigger (*cf.* Section 2.1): The component uses user information and profiles from the FInest platform and notifies the involved user depending on the stored notifications preferences stored in the profile. The user can afterwards trigger the action.
12	A **single platform for booking** of services for carriers at **transshipment points** (airport, port, terminals) *(UC1-3) (UC1-2)*	1. Integrates features of all FInest Modules and Components, incl.: information kept in **BCM**, resource management information from legacy & external systems (**Back-End**), etc.	2. Collaboration Object Storage (*cf.* Section 2.4): Stores data of history and up-to-date information of current transport processes. Collaboration Object Manager loads this data and makes it available for other modules of the FInest platform or even external systems.
15	The **right information** and documentation **on time** with **Alerts for delays** in document / data flow *(UC2-2) (UC3-2)*	1. Defines 'expected process' and controls its execution, sends notification on status updates, sends 'reminders' when inputs / actions are expected	1. Notification & Action Trigger and Standard Process Repository (cf. Section 2.1): The repository stores description of the 'expected processes' and the Notification & Action Trigger observes the current execution of transport process by the use of the Collaboration Object Manager. If the comparison of a process against its respective 'expected process' the component sends notifications to user of the platform by using the user and profile information.
Requirements Fulfillment Analysis - Monitoring & Visibility Category			
6	A "**monitoring**" portal with (semi) automated input from the tracking systems, showing status of the shipment/cargo (or event) *(UC2-3) (UC3-1)*	1. Status Updates on Shipments	1. Process Engine and Collaboration Object Manager (*cf.* Section 2): All data of a certain transport process is stored in the Collaboration Objects. The Process Engine receives events from the EPM and integrates this information into the respective Collaboration Objects by the use of

			the Collaboration Object Manager. With this the BCM is able to reflect updates on shipments.	
18	A **booking intelligence** application which combines booking information from different supply chain partners and gives a when trigger inconsistencies occur. *(UC2-3)*	1. Timestamp & status updates coming as events from **EPM**	1. Process Engine and Collaboration Object Manager (*cf.* Section 2): The Processing Engine integrates the events from the EPM into the Collaboration Object by the means of the Collaboration Object Manager. For this it also records the timestamp of the updates.	
20	"**Deviation management**" system for **detecting early signals**, automatic recording of changes, generating real-time alerts according to predefined points of interests, early indicators / alert rules and expected performance criteria *(UC2-4) (UC3-4)*	1. Knowledge about expected / planned logistics process 2. Actions to be done 3. Whom to notify	Notification & Action Trigger, Standard Process Repository and Collaboration Object Manager (*cf.* Section 2 and Section 2.1): 1. The Collaboration Object Manager initialize a transport plan of a certain type; the type refers to standard process of the Standard Process Repository; 2. The Notification & Action Trigger observes the execution of the process; based on the information from the Standard Process Repository it is able to determine which actions have to be performed at which process steps. 3. Notification & Action Trigger uses user & profile information from the FInest platform in order to determine responsible users for certain actions and these will be notified.	
Requirements Fulfillment Analysis - Resource Management Category				
10	A "**resource coordination**" portal for **transshipment** points, providing up-to-date	1. Current Status of overall logistics process	1. Process Engine and Collaboration Object Manager (*cf.* Section 2): Each transport process is represented by a set of Collaboration Objects managed by the Collaboration Object	

	information to suppliers about arrival/departure and all resources requested / booked. *(UC1-2) (UC1-3)*		Manager. These object models the planned transport and by the integration of Events (from the EPM) by the Processing Engine this model is kept up-to-date. By this means, information about the current status of the overall logistics process can be provided by the BCM.