"Cloud Based" Collaboration Platform for Transport & Logistics Business Networks

Prof. Dr. Rod Franklin, Managing Director, Executive Education and Adjunct Professor of Logistics, The Kuehne Logistics University, Hamburg, Germany

Dr. Yagil Engel, Senior Researcher, Microsoft Research, Tel Aviv, Israel (formerly Researcher, IBM Research Labs, Haifa, Israel)
Dr. Marianne Hagaseth, Senior Research Engineer, MARINTEK, Trondheim, Norway
Dr. Asmund Tjora, Research Scientist, MARINTEK, Trondheim, Norway
Dr. Fabiana Fournier, Researcher, IBM Research Labs, Haifa, Israel
Rene Fleischhauer, Researcher, SAP Research, Dresden, Germany
Dr. Clarissa Marquezan, Postdoctoral Research Associate, Paluno (The Ruhr Institute for Software Technology), University of Duisburg-Essen, Germany
Dr. Andreas Metzger, Head of Adaptive Software and Services, Paluno (The Ruhr Institute for Software Technology), University of Duisburg-Essen, Germany
Dr. Michael Stollberg, Senior Researcher, SAP Research, Dresden, Germany
Silvio Tschapke, Masters Student, SAP Research, Dresden, Germany
Dr. Clarissa Marquezan, Post Doctoral Research Associate, Paluno (The Ruhr Institute for Software Technology), University of Duisburg-Essen, Germany

1 Introduction and Motivation

Modern transport and logistics operations are highly distributed inter-organizational activities spanning country boundaries with each business partner acting autonomously with the objective of optimizing their individual supply and production chains without concern for the overall, end-to-end supply chain. Despite significant advances in other application domains, current ICT solutions in the transport and logistics domain generally support this internal myopic focus and provide only limited support for inter-organizational data and process integration. This means that complex inter-organizational integration and coordination activities must be accomplished through ad hoc ICT or manual efforts.

The lack of robust and trustworthy inter-organizational integration and collaboration systems hampers business efficiency and optimization for all parties involved in the planning and execution of international freight movements: customer requirements for end-to-end tracking and tracing must be satisfied through combinations of human inputs and interventions, heterogeneous information from incompatible ICT systems create barriers to interoperability between supply chain partner systems, and the end-to-end coordination of transport planning and execution requires extensive manual effort making transport and logistics operations more costly, opaque and error-prone.

This paper discusses the FInest logistics collaboration platform developed within the EU FI PPP project FInest (www.finest-ppp.eu). The FInest platform is an ICT development based on rapidly emerging technologies offered by the Future Internet [1] including software services, connected sensors and devices, and cloud computing.

The FInest platform offers novel capabilities to the transport and logistics industry that advance the state of the art as follows:

- FInest introduces an artifact-centric approach [2] for defining "collaboration objects" rather than the more traditional interacting or interaction approaches used to model service collaboration. This approach is being employed in the collaboration manager to more effectively manage actual goods flows as they evolve through a transport process.
- FInest provides novel semi-automated e-contracting services that provide digital facilities for managing service levels and establishing performance based contracts between collaboration partners, as well as connection with service market places. This approach extends and adapts strategies and tools for dynamic management of electronic contracts for service-oriented systems [3].
- Finest offers intelligent "sense and respond" event management services that utilize monitoring events along the supply chain in real time providing notification of issues as they arise and, through a learning process, of potential issues that may occur based on the current state of the supply chain environment. It constitutes an extension of complex-event processing solutions to become more proactive and adapted to transport and logistics specifics [4].
- FInest defines real-time planning and re-planning services that augment existing transport and logistics planning solutions to exploit real-world, online event data and forecasting of future situations [5].

After the description of the high-level architecture of the FInest platform in Section 2, this paper elaborates four core modules of the platform that provide the above novel capabilities (Sections 3–6). In addition to describing the technical architecture for those modules, the paper sketches novel, value-added business scenarios that demonstrate how the transport and logistics industry may be transformed by exploiting such cloud based logistics services. In Section 7 we conclude and provide outlook to future research and business opportunities.

2 High-level Architecture of FInest Platform

While the transport and logistics industry has made great strides in attempting to improve its efficiency, limitations in technology, transport infrastructure and regulatory regime incompatibilities have created significant barriers to future improvements. ICT technologies promise to overcome these barriers by allowing organizations to rapidly assemble collaborative logistics networks that can efficiently and effectively execute international trading activities. The Future Internet, with its promise of ubiquitous operation and information access, provides a potential platform for overcoming the limitations of current ICTs.

Building on core capabilities of the Future Internet, currently being developed under the European Union's Future Internet Public Private Partnership program (FI PPP[1]), the FInest platform [5] implements a domain-specific, configurable and extensible set of services for the transport and logistics domain.

[1] Specifically, they are developed within FI-WARE, the European R\&D project establishing a Future Internet core platform (see http://www.fi-ware.eu}

As shown in Figure 1, the FInest platform is structured into three layers.

The FInest platform itself is realized using service-oriented and cloud technology, facilitating interoperability, openness and extensibility through standard interfaces.

In addition, the use of integrated security and privacy management mechanisms ensures the secure and reliable exchange of confidential and business-critical information; including mechanisms such as access and identify management, artifact-based security control, and secure storage and backup.

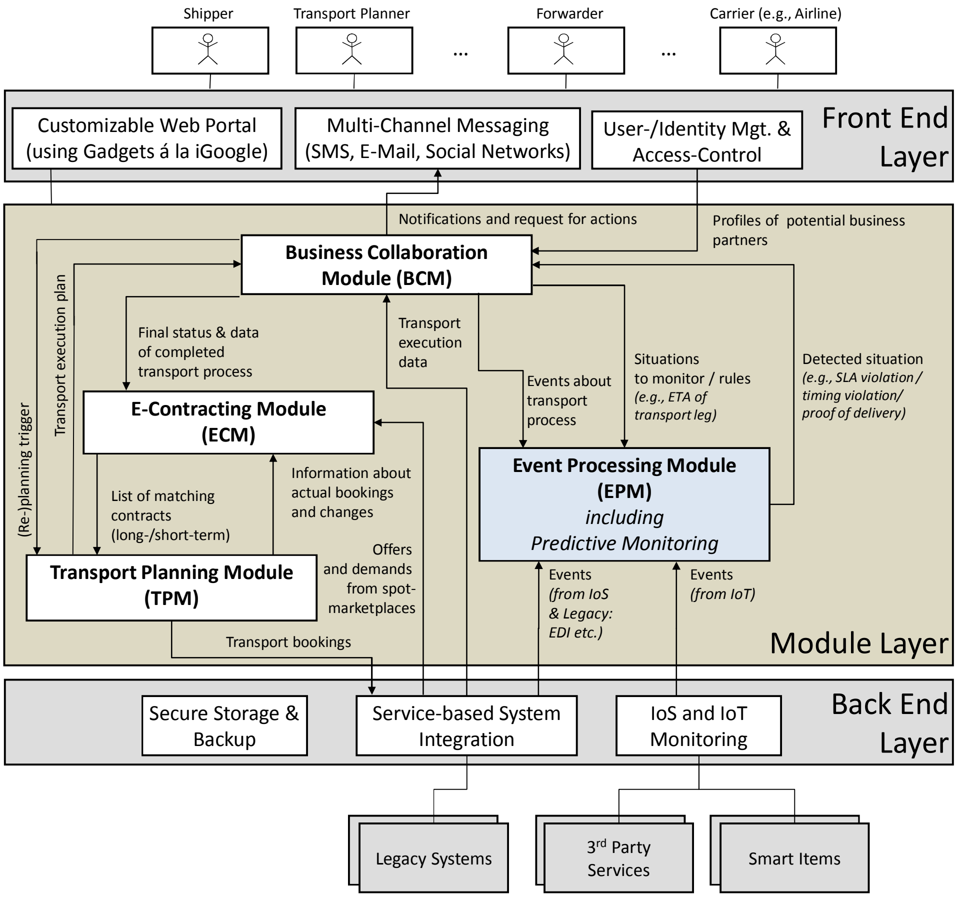

Figure 1: High-level Architecture of the FInest Platform

Below we first outline the three layers and the core modules of the FInest platform. In the remainder of the paper, we then elaborate the details and application scenarios of the core modules.

2.1 Front End Layer

The front end layer of the FInest platform provides users with role specific, secure, ubiquitous access from different devices to information concerning the operation of the transport and logistics network.

The capabilities offered by the FInest platform and its core modules will be offered through a customizable "web" portal. Each user can configure this portal by selecting dedicated gadgets depending on the capabilities needed to perform a user's respective tasks -- quite similar to the iGoogle model. In addition, the front-end will be integrated with messaging systems (such as SMS, E-Mail and Social Networking) so as to notify users and trigger actions.

2.2 Back End Layer

The back end layer of the FInest platform provides access to, and integration with, legacy systems, third-party services (Internet of Services) and Internet of Things (IoT) devices. Specifically, the IoT devices will provide (near) real-time information concerning the transport processes as well as their context, thus allowing systems and users of the FInest platform to quickly and proactively respond to planned and unplanned events. Legacy system integration is facilitated by service-oriented technology, e.g., by exposing features of legacy systems as services, or by offering access to legacy systems via the "Software as a Service" delivery model.

2.3 Modules Layer

The modules layer of the FInest platform allows plugging in targeted transport and logistics service modules. Those modules – in the future – are expected to be offered by third parties based on a business model similar to Apple's App Store. The initial release of the FInest platform will feature four open-source modules (called *Core Modules*) for contracting, planning, monitoring and execution of transport operations:

- **Business Collaboration Module (BCM)** This module supports the inter-organizational (business-2-business) collaboration between transport and logistics network partners by tracking and tracing shipments on the level of business processes and notifying the involved stakeholders in case of deviations or need for action. It may act as an intermediary between these partners and the various cloud based modules selected to manage the efficient flow of goods between the partners. It is detailed in Section 3.
- **E-Contracting Module (ECM)** This module provides computer support for service provider selection, contract management and the provision of contract related service requirements to other modules that utilize this information for ensuring effective and efficient network operation. It is detailed in Section 4.
- **Transport Planning Module (TPM)** This module provides support for dynamic transport planning and re-planning activities, exploiting real-time event data provided through the EPM and with respect to contracts between business partners that are managed within the ECM component. Re-planning of shipments occurs when real-time signals from the EPM indicate that a current transport plan cannot be achieved because of some event that has arisen in the shipment process. Such EPM-events will be analyzed by means of the BCM to understand whether re-planning is feasible at all, or whether other actions need to be taken. It is detailed in Section 5.
- **Event Processing Module (EPM)** This module provides event-processing facilities to determine relevant situations occurring within and in the context of the transport process. Such events include, for instance, delays of transport (notified from BCM), critical weather conditions (from IoT sensors and IoS services), and violation of Cargo 2000 milestones (from legacy systems). It is detailed in Section 6.

3 Business Collaboration Module (BCM)

The Business Collaboration Module (BCM) securely manages end-to-end networks between transport and logistics partners. Its main task is the execution of transport plans created by the Transport Planning Module (TPM) and the integration of information from other FInest modules, external legacy systems (e.g. ERP) and user input. The BCM is the central entry point for users to gain access to information about currently executing transport processes (e.g., the current position of their goods or the state of the process) as well as stored historical data. For this a special modeling approach is used that is able to reflect the interweaving of (static) data and process information within logistics processes. The BCM uses the artifact-centric modeling approach (cf. [1], [2]) and defines so-called Collaboration Objects. Each Collaboration Object encapsulates corresponding data and process information of a particular aspect of a transport process and, therefore, is a manifestation of an artifact from the previously mentioned approach. The BCM provides a specialized meta-model for Collaboration Objects to allow the precise and machine-readable representation of logistics processes. Through the integration of events from the EPM and other sources, the model is kept up-to-date so that all involved stakeholders can be provided with a view of the current transport process and its execution status. Sophisticated security and privacy mechanisms ensure that only authorized users have access to confidential information.

3.1 Conceptual Architecture of BCM

The BCM consists of a set of different components with specialized tasks. Figure 2 gives an overview showing their interaction and the following subsections provide a detailed description for each.

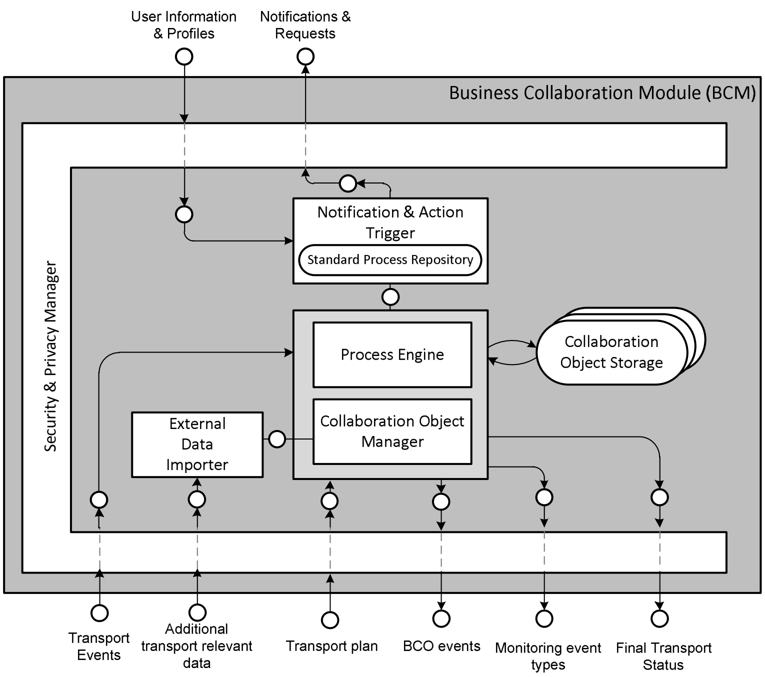

Figure 2: Business Collaboration Module Conceptual Architecture

3.1.1 Collaboration Object Manager

The Collaboration Object Manager (COM) is the central data management unit in the BCM and is concerned with the instantiation of the internal data model based on Collaboration Objects. It uses a specialized meta-model for logistics processes that builds upon existing standards (UBL[2] and GS1[3]). The source for this instantiation is delivered by the Transport Planning Module (TPM), which passes pre-assembled transport plans (static data such as origin, destination, goods information or the set of transport legs) to the BCM. Additionally, other external sources – such as legacy systems – can be used to complement the transport plan. For this, the COM uses the External Data Importer (EDI) to facilitate the import process. As a result all available transport information is stored in the internal data model and, additionally, this model is able to integrate process information in order to reflect the current state of the execution. Furthermore, the COM takes care of the persistent storage and retrieval of this data.

3.1.2 Processing Engine

The Processing Engine (PE) is responsible for keeping track of the execution of previously initialized transport processes and updating the internal data model. Most logistics processes consist of several transport segments, so-called legs. All legs are executed in sequential order. Via the integration of events from the EPM, the PE updates the process information of the corresponding Collaboration Objects based on the status of a transport leg. Through this updating process a specific leg can be marked complete and a subsequent one as started. The constant event monitoring and integration ensures the currency of the internal data model. In real-world scenarios the representation of the process status is much more detailed. This requires the BCM to manage each Collaboration Object using a comprehensive lifecycle model in which process information is integrated. This lifecycle is part or the meta-model and the PE manipulates internal status in accordance to this lifecycle.

3.1.3 Collaboration Object Storage

The Collaboration Object Storage (COS) is the central data store of the BCM. It persistently stores Collaboration Object instances and is able to select and load stored information. This storage is intended to be distributed, indicated by its multiple instances in Figure 2. This means that the COS does not consist of one monolithic data base, but each provider of the BCM can define their own data storage approach. Distribution of the data avoids security and data risks associated with the central storage of confidential information for all participants of a logistics process. The COS works in a cloud-based manner and hides the actual storage technology from potential client systems. These systems simply send store and load requests and the underlying logic delegates these to the appropriate data storage service.

3.1.4 Notification and Action Trigger

The Notification and Action Trigger (NAT) constantly observes ongoing transport processes contained in the COM and updated by the PE. If an update occurs that requires user notification and perhaps an action, the NAT identifies the appropriate users and determines the preferred communication channel for the message type (e.g., SMS in urgent cases and

[2] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

[3] http://www.gs1.org/

email otherwise). For this, user and profile information from the FInest platform are used. The NAT not only works in a reactive manner, but also detects the absence of required events in a transport process. To do 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 company specific processes – and define the milestones for each type. The Standard Process Repository stores these definitions and enables the NAT to load them as necessary. If a process will not reach a certain milestone in time, the NAT is able to detect this and notifies the corresponding user by using the same procedure as that employed for the user notification described earlier.

3.1.5 External Data Importer

The overall purpose of the External Data Importer (EDI) is to eliminate external data format and internal model incompatibilities during the import of data from heterogeneous external sources. To accomplish this, it provides a set of different adapters that map a certain data format to the internal data model of the BCM. For every supported import format the EDI provides a specialized adapter.

3.1.6 Security and Privacy Manager

Logistics processes entail large amounts of confidential information, which must be available for a certain set of stakeholders, but not for others. The Security and Privacy Manager‘s (SPM’s) main task is to ensure the integrity of the stored data and the non-disclosure of confidential information. It checks every in- or outbound communication and uses specialized authorization procedures for the Collaboration Objects-based meta-model. Information about the identity of a certain user (authentication) is provided via the FInest platform itself.

3.2 Implementation Approach for Collaboration Objects

In this section we describe the technical realization of the Collaboration Object concept. This encompasses the definition of a suitable meta-model and the consideration of general implementation approaches with a particular focus on the requirement for the use of industrial strength technologies.

3.2.1 Modeling Approach

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 predefined class of problems. In order to realize the idea of Collaboration Objects it is necessary to develop a meta-model for the transport and logistics domain with respect to the underlying artifact-centric modeling approach. Although meta-models are already used in logistics, they primarily focus on data without regard for typical process structures, or they lead to inflexible, hard to manage model instances. The envisioned solution makes use of existing standards and enhances them with capabilities to realize the concept of artifact-centric modeling. This means that we cover data and process modeling in a similar manner from an abstract level.

The basis for the development of the meta-model for Collaboration Objects is established by existing models for transport process descriptions and standards. The FInest project provides a set of three different use cases and the models are assessed as to how well they are able to represent the use case scenarios. The approach used focuses on the identification of process information for certain entities that is necessary to enable the artifact-centric approach of Collaboration Objects. This investigation results in a set of so-called core types' and relations between them. Each core type can be instantiated in a Collaboration Object and encompasses data and process information of a certain aspect within transport processes (e.g., transport leg or shipping order). Hence, 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 that 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 artifact-centric approach is built on a set of key elements, separated into four explicit, inter-related but separable dimensions [2]. First, each artifact has a data model that contains all the information to ensure business operation. This may include business level information or meta-data as unique identifiers or entity states. Second, the (macro-level) life-cycle defines the stages an artifact passes through during business operations. The third dimension reflects tasks or services. Like SOA, artifact-centric approaches aim at flexible specification of distributed systems [2]. A task or service encapsulates a unit of work meaningful to the whole business process and may be either fully automated or require human interaction. Finally, associations define conditions under which tasks are invoked. Such conditions may include precedence relationships among the services, or constraints between services and events. The Guard-Stage-Milestone (GSM) meta-model, introduced by Hull in [3], is an approach for a declarative specification of life-cycles and interfaces. This meta-model provides a "language" and a graphical notation to specify artifact-centric models. Although based on a conceptual level without any tooling support, the FInest solution design is inspired by its concepts. The outcome is a set of artifact types that behave according to the guard-stage-milestone meta-model and, as a result of the data modeling process, that provide an end-to-end view on the process with access to required information at each specified time.

3.2.2 Technical Realization

The previously described meta-model is designed by means of an artifact-centric modeling approach and is inspired by GSM. However, there is no commercial GSM runtime engine available. Although the ACSI project[4] is currently developing a prototype that directly supports the modeling and execution of GSM based models, it cannot be compared to technologies that are applied in industrial practice. We assume that established technologies provide a higher degree of performance optimization, so that they 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 in an industrial strength environment.

For the artifact 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 high performance 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 aspect of every Collaboration Object is detached and handed to the processing engine separately (see Figure 3). The COM provides the necessary logic to bind the process and data part of each Collaboration Object together. Using this approach,

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

the BCM is able to work with the artifact-centric modeling approach on the logical level and makes this concept available for other modules of the FInest platform. However, technical execution is realized by established technologies, which allows the BCM to be compliant to requirements regarding performance and scalability. In [12] an example implementation is provided based on SAP's Netweaver BPMN engine.

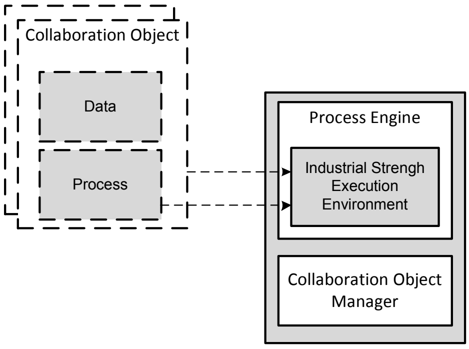

Figure 3: Separate execution of process aspect by execution environment

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

3.3 Demonstration of BCM Capabilities: Resource coordination within a maritime transport scenario

The FInest project has developed a demonstrator to showcase the most important features and capabilities of the BCM. The demonstrator focuses on the coordination of resources for vessel calls (or port calls) within a maritime transport scenario. While maritime transports in general have to deal with a variety of different challenges, two of them are of particular importance for the BCM:

1. Terminal Planning: The distribution of information between involved stakeholders and the terminal is often done in a manual manner. This leads to late notifications and rush work as well as the waste of resources.

2. Resource Coordination: A port is above all a service provider. Services are provided to various customers, such as shipping lines, ship agents or vessel captains, terminals and others. To properly manage the provision of these services 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. However, currently employed systems lack visibility to all the necessary information to perform this task so manual methods are generally employed to distribute this information.

The demonstrator considers three different actors: a Logistics Service Provider: a Shipping Line, Logistics Service Provider: a Port and Logistics Service Provider: Terminal; and outlines a portal that acts as single point of access for each involved stakeholder. The portal uses the BCM as its backend and makes use of its data storage facilities. For each planned transport, the BCM generates the corresponding Collaboration Objects and by the integration of either user generated events or events from external systems, the transport processes are update. The user is informed over the portal about the updates and is able to act accordingly. The novel aspect of this is that all involved users are able to work on one data set, which allows immediate reaction and reduces the amount of manual interventions. External systems can be accessed transparently via the portal. The demonstrator provides an initial impression of how the envisioned infrastructure enables the use of both legacy systems and new cloud based tools, regardless of whether these tools were developed within the project or not.

In Figure 4 a snapshot of the demonstrator can be seen.

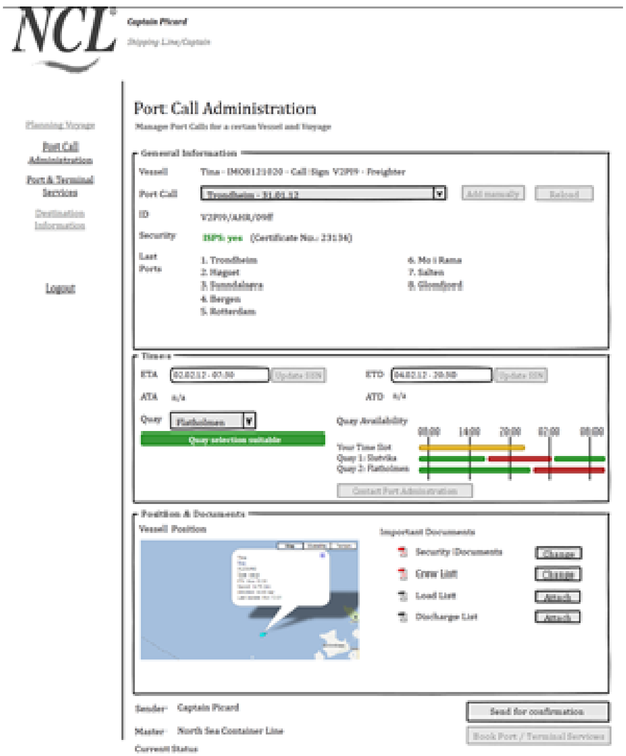

Figure 4: Resource Coordination Demonstrator

4 E-Contracting Module (ECM)

Currently, a transport and logistics (T&L) contract is a paper based document that is not generally available to the individuals who are responsible for executing the service agreed to in the contract. As a consequence, execution of paper-based contracts do not allow automatic, real-time and accurate checks as to whether the Service Level Agreements (SLAs) of such contracts are being maintained, which results in billing problems among parties to a contract and after the fact verification of contract violations.

In addition, the contracting process for complex international movement of goods is currently a manual and time consuming process. The process begins with the identification of a shipment requirement. Needs identification is followed by partner identification and qualification, partner bid development and bidding, bid evaluation and tentative partner selection. Once a tentative partner has been selected a contract is negotiated and agreed between the contracting parties. All these steps until the actual agreement among the parties are typically executed by individuals interacting using email, phone calls, datasheets and/or text documents. There are only a few initiatives trying to automate the contracting process. However, these attempts are still far away from solving the high dependence on manual intervention [16].

There are at least another two other problems associated with the manual contracting process. The first is that in most cases the offers between the negotiating parties are exchanged though datasheets or text documents. This results in a lack of common representation of SLA attributes and requirements in the contracts. The second problem is related to the manual and individual-knowledge-dependent identification of potential partners. Contracts based on "who one knows" generally favor large, established LSPs. This individual-knowledge-dependent process prevents SMEs from having the chance to present their offers or increase their participation in complex international movements of goods.

In order to change the current situation for contract establishment and management we introduce a cloud-based E-Contracting Module (ECM) as a core module of the FInest platform. The major capabilities of the ECM module are:

- Represent in an electronic and online form the attributes of T&L contracts that are most relevant for the execution of T&L services. Storing contracts in an electronic form is not enough. It is also necessary to model this information in such a way that it can be easily accessed and used during the execution of T&L services associated with this contract.
- Detect and signal near-real-time deviations on agreed terms from the contracts. Examples of terms that could suffer deviations are the amount of cargo, time-windows, or discrepancies in location of the goods (source, destination). This capability enables more agility in the T&L business and has the potential to reduce the amount of overcharges because checks on the contracts are now executed in an automated and online manner.
- Signal the need to re-evaluate contracts due to many deviations. In the case of long term relationships among parties, long term contracts are established. These contracts contain general service level requirements (e.g., total amount of items that could be transported, periodicity of reserved capacity, price of transportation, etc.). When a transport service is actually booked under the terms of a long term contract, there may exist discrepancies between what is booked and what has been agreed to in the contract. If these discrepancies occur frequently this may be an indication that the long term agreement is not really satisfying current needs. The ECM is able to identify such situations and signal this information so that corrective actions can be taken.
- Connect to Marketplaces. The ECM has the capacity to connect to different marketplaces, which can include marketplaces inside the FInest platform or external to it.
- Search for service providers and consumers in marketplaces. The objective is to provide automated support for representing both T&L demands and offers. This way, automated and proactive mechanisms of match-making between offers (service provider announcements) and demands (service client's announcements) are developed in the ECM. This search mechanism enables automated and active search for partners.
- Manage the life-cycle of contracts. Examples of phases of a lifecycle are: the creation of the online representation of a contract, notification in case of expired contracts or of contracts near their expiration date.

4.1 High Level Architecture of ECM

Based on the aforementioned capabilities, a high level architecture of the ECM has been defined that is illustrated in Figure 5.

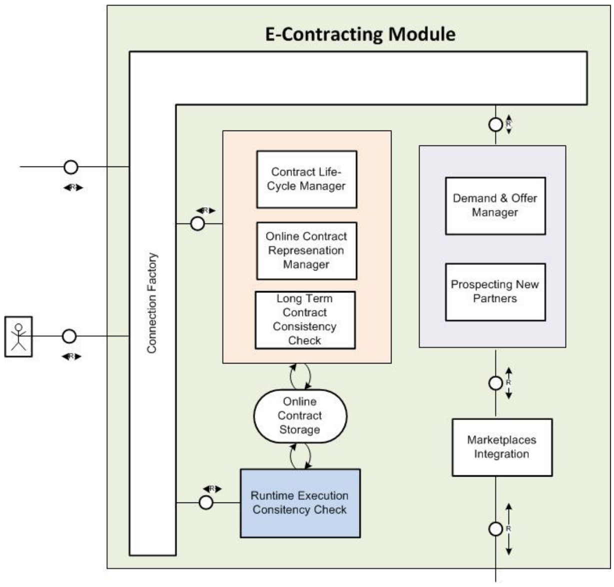

Figure 5: Refinement of High Level Architecture of E-Contracting Module

The key architecture elements supporting the ECM include:

- **Connection factory –** This is a service responsible for providing an interface to the services provided by the ECM module. The interfaces can be accessed directly via users or by other systems using the functionalities of the ECM.
- **Contract Lifecycle Manager** – This service is responsible for managing the creation, update and deletion of contracts. It is also responsible for monitoring and notifying expiration dates of contracts and requirements for changing SLAs due to ongoing failure to perform.
- **Online Contract Representation Manager** – This service is responsible for providing search facilities among the contracts as well as extraction of information from contracts.
- **Long Term Contract Consistency Check** – This service is devoted to checking whether a request for booking capacity associated with a long term contract is actually consistent with the terms of the contract. This checking process is performed before actual execution of the contracted service.
- **Online Contract Storage** – This is a data repository for all established transport and logistics contracts.
- **Runtime Execution Consistency Check** – This service is responsible for checking the actual performance of a service that has completed in order to determine if this performance is consistent with the booked SLAs, which in turn must be consistent with the contracted SLAs.
- **Demand and Offer Manager** – This service provides support for the creation of demands and offers; the search for transport service descriptions; and execution of the requested matchmaking process between offers and demands.

- **Prospecting New Partners** – This service supports the discovery of new partners, either for establishing long term contracts or finding partners in Spot Markets.
- **Marketplaces Integration** – This service is designed to support the integration of information from marketplaces (internal or external to the FInest platform).

The ECM is able to provide the automated and online support for dealing with different challenges in the transport and logistics domain. For example, ECM can support the automation of transport order creation by feeding the planning module with online information about contracted options, as well as provide spot market partner information by interfacing with marketplaces. Another example is support of deviation management through the automating of the process for finding new partners to substitute for the partner who cannot perform the contracted service as planned.

4.2 Demonstration of ECM Capabilities: Late Booking Cancellations

The "Late Booking Cancellation" problem relates to finding solutions to address the cancellation of booked capacity in very short time windows. This is a problem that affects both large and small players in the transport and logistics domain. Late cancellation of booked capacity under long term agreements tends to be covered by penalties agreed to in the contract. This is not true for parties operating with short term or spot contracts.

The absence of long term agreements or established relationships with partners leads organizations to suffer large losses with late booking cancellations. Current systems are not able, or are limited in their capabilities, to assist in the identification of alternatives to cover the cancelled booking and avoid lost revenue and inefficient transport of less than full loads.

A demonstrator was developed as part of the FInest project to test the capabilities of the ECM. The demonstrator used the FInest use case for the shipment of perishable food (fish) from Norway to Brazil. In this use case the shipper, NCL (North Sea Container Line, the shipping operator, offering container feeder services), operates using a spot market model. It receives booking orders that are not associated with long term contracts. Spot contracts are represented by the actual waybills that are transported with the cargo. As a result, there exist no contract between shipper and carrier as long as the cargo does not reach the port terminal; more precisely as long as the cargo is not loaded onto the vessel. This means that shippers are not punished or charged if the bookings are canceled at the last minute or if the cargo never shows up at the port.

This demonstrator shows the envisioned scenario where shippers use the FInest platform to create shipping demands on FInest and/or public marketplaces. Shippers get offers for transport from carriers and can reserve the offered vessel capacity either by prepaying or by simply reserving space. Prepaying will result in confirmed capacity reservations while a normal reservation will only put the cargo on a waiting list that will be used if not enough prepaid reservations have been made for the vessel. Additionally, shippers have the ability to change the amount tendered for a reservation, if required, and carriers have the possibility to cancel non-prepaid reservations.

Carriers can utilize the FInest platform's marketplace, or external marketplaces, to find replacement loads for cancelled shipments using the shipping demands published to these marketplaces. They can react to cancellations either by finding replacements for single cancellations or use a screen to find sufficient shipment tenders to fill the remaining free space on a vessel. Figure 6Figure shows what a carrier, in this case NCL, might see using the FInest platform concerning cancelations and capacity requirements for a ship. Based on the information in the tables, the carrier is able to identify last minute cancelled capacity and has access to support to react to these changes.

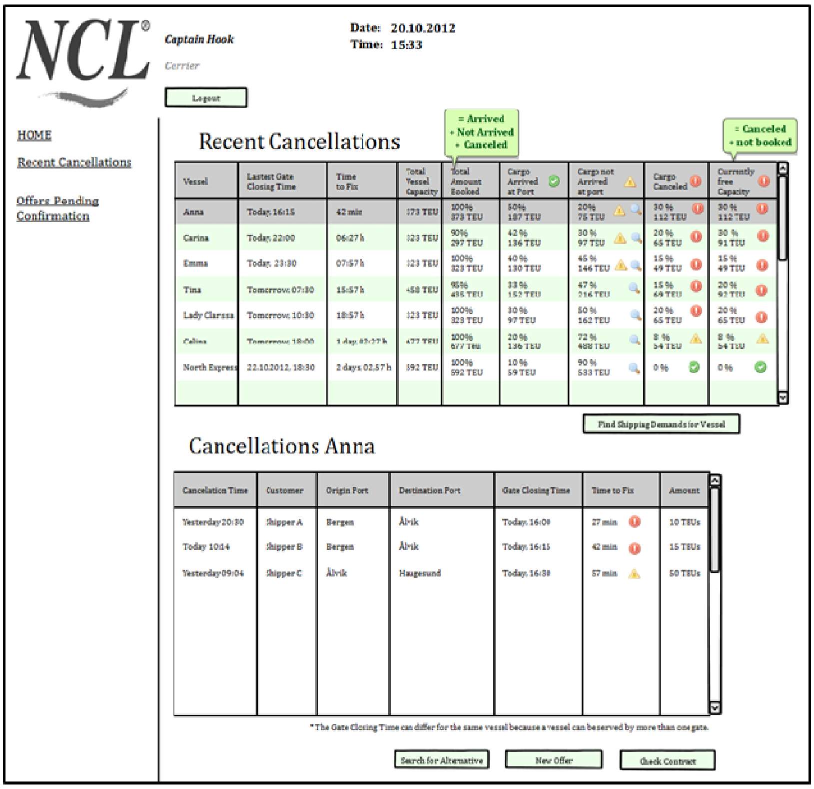

Figure 6: Carrier overview about cancelled booked capacity

One possible action that the carrier can take is to examine shipment demands for ports along the ships sailing route and not simply between the origination and final destination ports. In case of locations along the sailing route, the carrier can directly contact a freight forwarder who would be responsible for organizing the additional transportation route(s) (Figure 7).

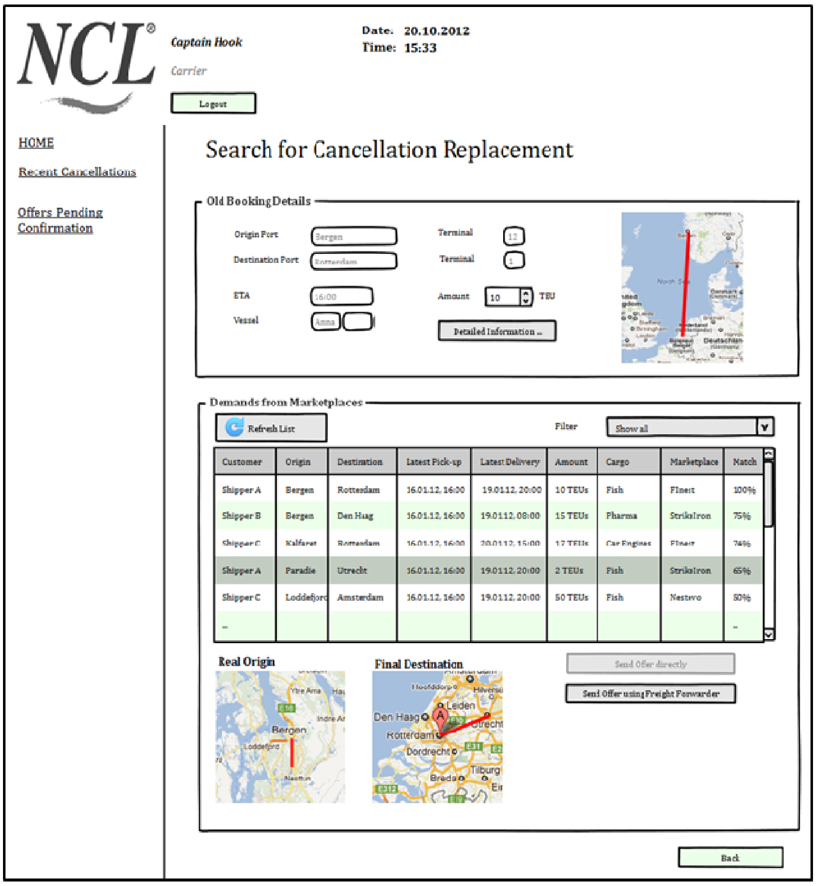

Figure 7: Search for offer matching one specific cancelled booking

A second possible action that the carrier can take is shown in Figure 8. This action focuses on looking at alternative loads for the vessel as a whole. Figure 8 displays, in the graduated blue bars of the chart, the vessel capacity between the ports of its tour and the cancellation rate for each part of such tour. The upper part of Figure 8 displays the shipping demands that can be matched to the voyage of the vessel. By selecting one of these demands the carrier could send an offer to transport such cargo directly to the shipper as an action to try to solve the late booking cancelation of another shipper.

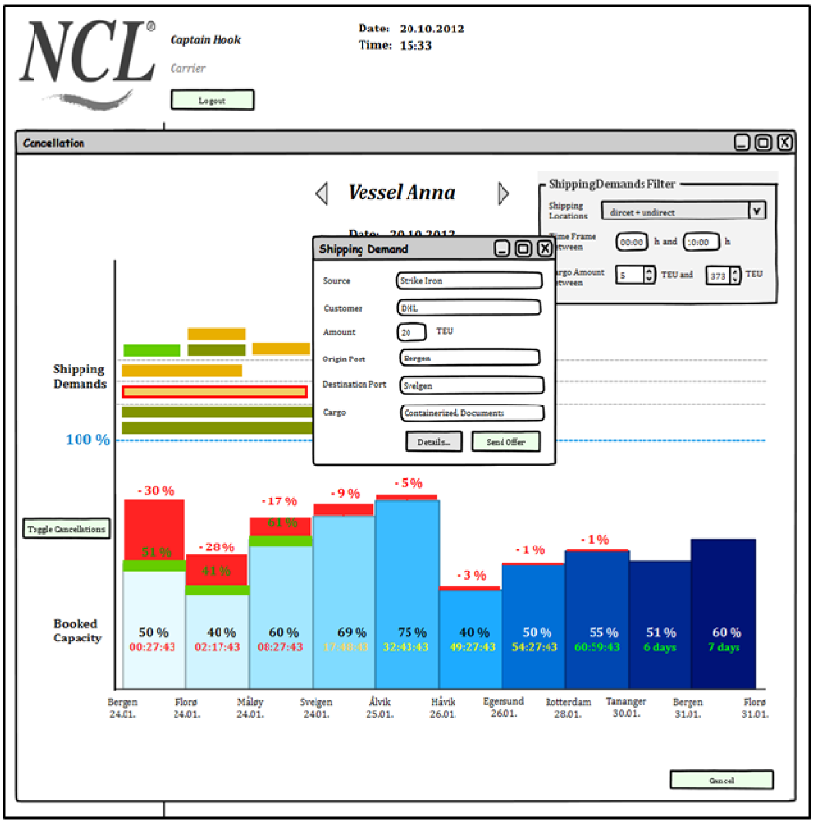

Figure 8: Search for offers matching empty capacity on the voyage of the vessel

The support for the different alternative response approaches demonstrated in this section is based on a combination of the services of the ECM architecture. Specifically, by using the "Demand and Offer Manager", "Prospecting new Partners" and "Marketplaces Integration", the ECM is able provide potential solutions to the late booking cancelation problem. Other modules from the FInest platform are required in order to provide the functionality demonstrated in the examples.

5 Transport Planning Module (TPM)

The Transport Planning and Replanning Module (TPM) is a FInest module intended to help with the planning and replanning of a transport chain, including describing the transport demand, finding and configuring a "best choice" of transport services, quotations and booking of the selected services. A planning process for a shipment will end up with a transport chain plan (TCP) that describes the goods' itinerary through the whole transport chain. Such a planning process is based on information about the cargo to be transported, selection of logistic service providers that will play a role in the transport, as well as schedules and requirements for the different means such that the plan is consistent and easy to track during the transport execution process.

5.1 High Level Architecture of the TPM

Figure 9 shows the conceptual architecture of the TPM containing the following components:

- Interface Manager

- Transport Demand
- Transport Search/Optimization/Selection/Configuration
- Transport Booking
- TSD (Transport Service Description), TEP (Transport Execution Plan) and TCP (Transport chain plan) Maintenance
- Replanning

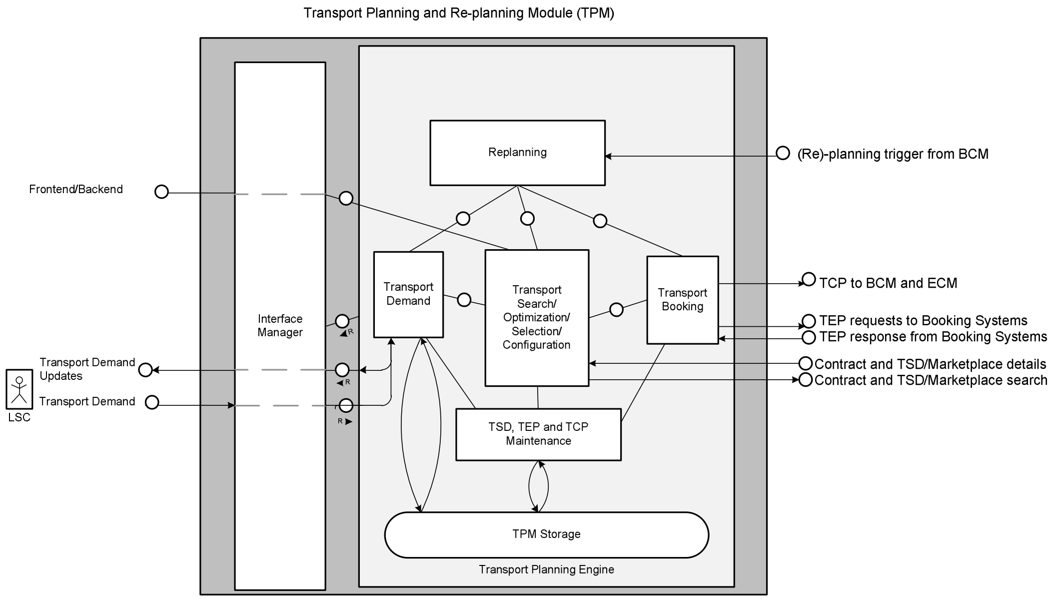

Figure 9: Transport Planning and Replanning Module Conceptual Architecture

5.1.1 Interface Manager

The Interface Manager makes use of functionality from the general FInest platform [5] and provides the necessary logic to adapt Finest platform functionality to the internal TPM components.

5.1.2 Transport Demand

Before planning can start, the users have to describe their demand for transport. For shipments, this includes describing the goods to be transported, the pickup and delivery points, the time interval for pickup and delivery, etc. Note that the demand may include all types of logistics services, not only services directly covering transport, but also services related to terminal and port operations, agent services, document handling services, carrier services, warehousing, etc. As there typically are at least two actors involved in defining the transport demand (the shipper and the consignee of the cargo), it can be useful if several actors are allowed to do updates to the demand, and if the work can be supported by collaboration facilities. This also means that in addition to create, read, update and delete (CRUD) operations on the demand descriptions, services for setting administrative rules and notifications for changing the demand are also necessary.

Third party systems can use the import transport demand service provided by the TPM. If needed, mapping to the TPM transport demand format is done by adapters implemented in the FInest platform.

5.1.3 Transport Search/Optimization/Selection/Configuration

When the demand is described, the system has to find services that can fulfill the demand. First, requests may be sent to the ECM (e-Contracting Module) to check which contracts are relevant for this transport demand. Next, for parts of the demand that are not covered by existing contracts, the TPM may look for services at Logistics Service Providers' (LSP) sites, transport marketplaces (provided through the ECM), and service description publishing hubs and try to find services that wholly or partially matches the demand. Some constraints that can be used to reduce the number of transport services fulfilling a certain transport demand include the LSP's capability to handle the kind of goods/packaging of goods in the demand, the timing of the services (i.e. whether they fit the demand's timing windows), the cost of the services, quality of service, etc.

The Transport Service Description (TSD) format described in the 'Common Framework' [14] is used as the exchange format for transport services. A TSD is an announcement of transport services. A transport service can be the carriage of goods between origins and destinations, but can also be warehousing, port or terminal handling services, document handling, and other services related to the movement of goods.

For the user to have a manageable set of services, the services should be filtered and optimized, i.e. discarding services with poor fit to the demand and configuring the remaining services in a way that suits the demand description in the best possible way. By optimizing, we do not mean traditional network optimizing; however, we mean that the optimal configuration of a solution is found using a fixed route map as given by the logistics service client (LSC). The optimal configuration is decided based on requirements given in the transport demand (cost, time, environmental aspects), based on the given legs and service requests set up by the LSC in the transport demand, and also based on available contracts, services offers from some marketplace(s), and service offers from some booking system(s) that are fetched by the TPM.

When services that fit the transport demand are found, the user can select the services they want to use and may also have the option to manually configure these services. This may also include adding services that the transport service search has not found, e.g., preferred customs agents at the export/import points, parts of services that can be handled by the LSC, etc. During this stage, monitoring requests that are not considered default may also be added to the plan. During this process, it may be useful to have online collaboration with the transport providers and other actors in the process, especially to ensure that configuration changes are acceptable before trying to book a service with these changes. The system, or advanced clients of the system, may also provide some kind of "simulation tool" to try to predict effects of changes in the plan on the different actors. The process may also be asynchronous, related to the marketplace where transport demand and service offers are matched.

5.1.4 Transport Booking

When a service, or a set of services, are selected and configured they have to be booked. The booking process may involve simple bookings through the LSP's systems, or through marketplaces or logistics information hubs with booking facilities. It is the task of the LSP to manage resource and service availabilities and to respond to resource and service bookings. If a booking fails (e.g. the client is not able to get the service they planned for), the system may be configured to go back to an earlier stage, e.g. demand creation (for changing the demand to something that is easier to fulfill), service search (for getting new or updated service descriptions and new possible transport chains), or service selection (for selecting other services or configuration of services). The system may also be configured to try a set of similar services automatically, so a failed booking results in a booking attempt for another service without the client being involved. How this is done in each case is configurable in the TPM. On the LSP side, the TPM may also provide a simple interface for handling booking replies. While the typical booking will be through the LSP's own system or 3rd party systems, a simple reply system will be of help for LSPs not having access to these kinds of systems.

5.1.5 TSD, TEP and TCP Maintenance

This component contains functionality to maintain Transport Chain Plans (TCPs) describing a door-to-door transport. A TCP that is ready for execution contains a set of transport execution plans (TEPs) and this is sent to the BCM for execution together with monitoring information. The Transport Execution Plan (TEP) is a plan established between a LSC and LSP. There is one TEP for each service, and it can be used for arranging other kinds of transport services than pure carriage, such as warehousing storage and terminal operation services. A TEP consists of several classes, including for instance goods item including handling instructions with max and min values for constraints.

5.1.6 Replanning

Most of the functionality used during replanning is the same as that used during planning, e.g. describing and updating demands, searching for new services and booking of services. There is some internal functionality (e.g. not accessed directly by the planner) related to handling of replanning triggers and updating transport chain plans that already has started execution. Figure 10 shows that during replanning, the LSC can re-enter the planning process at several points:

- An alternative transport service can be selected
- The information from previous transport can be reused in to a new transport plan
- Transport service search can be repeated
- Transport demand can be changed.

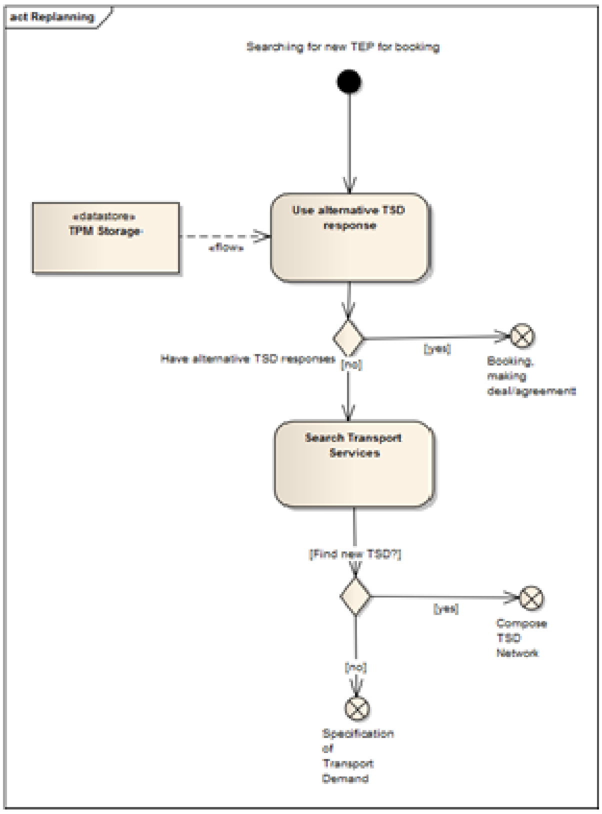

Figure 10: TPM Replanning

5.1.7 TPM Storage

The TPM must store TCPs and all service search and booking related data to be able to perform the planning and any replanning.

5.2 Demonstration of TPM Capabilities

During the work on the TPM, a demonstrator for a transport planner was made in the tool ForeUI [15] in order to demonstrate some of the types of functionality that is envisioned for the module. The business motivation behind the demonstrator is to show how transport order creation can be done when up-to-date information on relevant contracts and transport services are integrated with online booking system capabilities. The scenario shown in the demonstrator is as follows:

1. The supplier starts describing the shipment. The suggested shipment is sent to the FInest transport demand description. This triggers a notification to the planner, and corrections to the shipment and transport plans are done by the planner by updating the demand descriptions. If the supplier needs to change something (or approve changes done by the planner), the system will send them a notification.

2. The planning may start when the demand description is ready; for a well-defined description, this may be done by a click of a button. The planning consist of several stages that are automated:

   a. The planning system starts by fetching the planner's long-term contracts from the ECM. This is used both for prioritizing the search for transport services, and for modifying the terms in the transport service descriptions.
   b. The planning system then fetches information from the parties with long-term contracts, and starts building a set of transport chains from this.
   c. If the set of transport chains cannot be completed by parties having long-term contracts with the planner, the system will search for spot market contracts. The service requests for these are added to the set of transport chains.
   d. An optimization function reduces the set to a set of "best fit" transport chains, and configures these chains to a "best possible" configuration (with regards to timing, cost, quality of services etc.)

3. The resulting transport chains are presented to the planner. The planner may now select and configure plans, including:

   a. Selecting the preferred transport chain and preferred providers for each leg of the chain. Several choices may be made; this is useful both for the cases when one provider is unable to carry the entire demand, and may also be used by an automated booking process to automatically try alternatives if a booking fails.
   b. Configuring the transports. This may include changing the timing, suggestions to new pricing, adding monitoring requests, etc.
   c. Adding new services (i.e. services not included in the automatic transport plan). This may include agent services (customs, etc.)
   d. Enter collaboration with providers. This may be useful for getting approval to transport configurations before booking, and making sure that the described service found by the system is valid (i.e. bookable)

4. When the transport plan is finished, the system tries to book the services in the plan. For a Future Internet system, it is assumed that the booking can be done online using services accessible by the TPM.

At any time, the planner has the possibility to view transport plans at any stage of planning (e.g. during planning, during booking of plan, during execution of plan), including statuses, warnings and deviations.

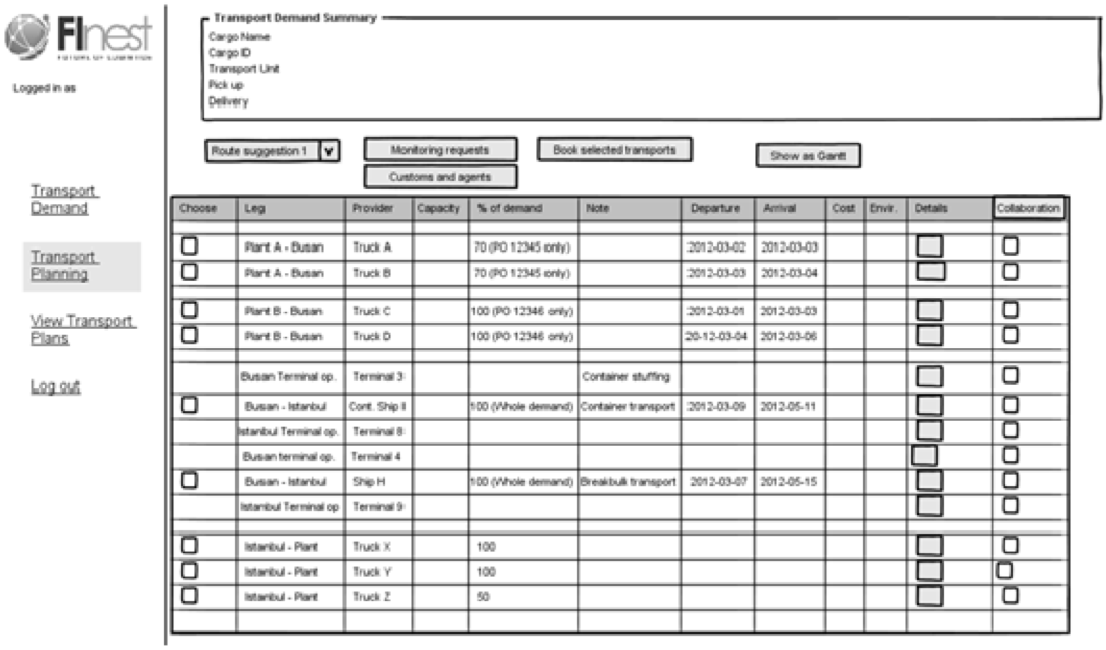

Figure 11: Suggested route based on up-to-date contracts and service information

Figure 11 shows a screen shot where the planner has given a transport demand, and where the TPM has proposed a set of transport services based on contracts and service searches from the ECM and from other 3rd party systems.

5.3 Novelty in Comparison to Standard Transport Planning and Replanning

An important contribution from the Finest project to the domain of operational planning is the possibility to utilize online, up-to-date service provider information on transport services together with online transport demand requests from the logistics service client. This creates new possibilities both for planning with long time horizons (e.g. planning a transport that will be executed weeks or months after the planning starts) and for planning with very short time horizons (e.g. replanning with tight deadlines).

FInest concepts are suitable to support collaboration between different actors (the various providers and clients of different logistics services) during set up and changing of a transport plan (TCP) covering a supply chain. Most important is that the FInest platform is used to make information available to all actors participating in the planning task.

If the LSPs publish updated online description of services and service schedules, this can be used to create a better overview of the transport possibilities; it is envisioned that service description publishing will be common in the future, and the TPM should utilize this potential.

It is also envisioned that booking of transport will be possible through online services, e.g. through the provider's own services or services at a transport marketplace or Logistics Information Hub, and that the TPM can access these booking capabilities. Up-to-date information on the available capacity of the services may improve the quality of the transport plan and may lead to fewer changes for both the LSC (less need for replanning, delivery on time) and the LSP (fewer changes in bookings, better utilization of capacity).

The FInest platform will also make it easier to monitor a transport and based on the received data from a sensor or an observation, it will be easier to calculate if the transport is going as planned, or if there will be deviations that can lead to a replanning. Planning and replanning is all about finding best alternatives to serve the demands within a transport.

6 Event Processing Module (EPM)

The role of the Event Processing Module (EPM) in FInest is to collect events from various sources, and perform what is known as *complex event processing* [6] *(CEP)* operations on them in order to detect situations on which to notify the business collaboration module (BCM) or the frontend of FInest. The availability of such automated response technology will provide FInest users with end-to-end visibility and monitoring of the process, and with the ability to quickly respond to various changes and deviations that often occur in the dynamic and complex logistic processes that FInest is expected to handle. The EPM relies on existing CEP technology; however, as described later, the particular setting of this domain and the specific demands of FInest requires extending the basic event processing architecture in novel directions.

6.1 High Level Architecture of EPM

The Event Processing Module (EPM) includes the following components, which are identified in Figure 12 and briefly described in the following sections:

- Events and Events Sources
- Rules and Rules Definition
- Detected Situations
- Runtime Engine

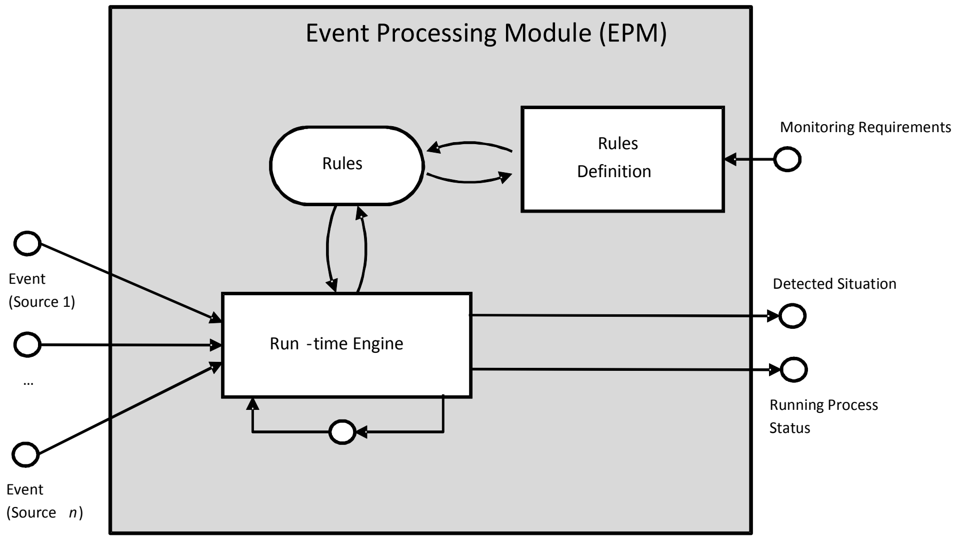

Figure 12: Event Processing Module Conceptual Architecture

6.1.1 Events and Event Sources

FInest EPM will monitor various events that are relevant to the execution of a transport plan, from the time a booking is established. Most event sources and the means of communication with them will be defined in the application. Some of the events are *internal*; for example, as shown in the demonstrator in this deliverable (see Section 6.2), a truck driver may issue a pickup event, or alternatively, submit a booking discrepancy event prior to loading the shipment. Other events are *external*; they are received from various electronic systems such as AIS (Automatic Identification System for radio navigation), Airport, and existing tracking systems employed by freight forwarders. In both cases, we call the entity that produces the event an *event producer* as opposed to a consumer that is the logical complement of the event producer, it accepts event objects from entities in an event processing network and processes them; refer to [7] and to Figure 13 for a general picture of an event processing network.

6.1.2 Rules and Rules Definition

The behavior of the EPM is defined by event processing rules. The two parts of an event processing rule are:

- Event Pattern (the rule *body*): may include a single event, or an operator such as filter, join, aggregation, or trend, defined over a set of event types [8] along with mathematical conditions. For example:
    - A single event, which is a temperature read from RFID, with conditions comparing its value with the allowed range.
    - A trend operator over a series of temperature reads, identifying steady increase.
    - Aggregation operator: sum the gap between actual cargo and expected cargo of flights departing towards a specific airport; a mathematical operation can then forecast the resulting cargo backlog at that airport.
- Derived Event (the rule *head*): if the pattern in the rule body is matched, the rule triggers, and the result is that an event (which is a "complex event", or "high-level event", representing the occurrence of the pattern) is created and emitted. The resulting derived event can be sent to a consumer (for example, notify a user) or serve as an input event to another rule in a chain.

The transport and logistics application is a system that monitors a large number of short-term executions (shipments), both concurrent and sequential. The solution in FInest is a *configurable rules system*, with a structure of three layers:

- Generic event-processing platform, as described above.
- A set of parameterized rules (that can be edited and maintained by designer); these are called *rule templates*. For example: if neither event "pickup" nor event "booking modification" are received within *x* min of scheduled pickup time, change status to "delayed pickup". Here, *x* is a parameter of the rule (in general, multiple parameters are possible).
- Specific logistic scenarios: For each particular logistics scenario the following two configurations are being done: (i) which rules are activated for this scenario, and (ii) the value of the parameters to each rule. In the example above, for a specific TEP (Transport Execution Plan) we might ask this rule to be instantiated, and provide the parameter "30 minutes".

The parameterized set of rules for FInest is created through interaction with domain experts.

6.1.3 Detected Situations

A triggering rule represents the detection of a situation; this is communicated to the rest of the system, as mentioned above, through an event or a derived event. A situation detected by the event processing engine is reported to *event consumers*, which in the case of FInest refer to other technical modules that receive events from the EPM. More specifically, the EPM communicates with the Business Collaboration Module (BCM), in two directions, as detailed below. In addition, EPM may send notifications to the FInest frontend.

- **Input**: *Set of rules to instantiate from the (TEP)*; The TEP provides the configuration discussed under "rules definition" above, by (i) specifying which of the rules and the rules templates should be instantiated to monitor the execution of this plan, and (ii) provides parameters to the relevant rule templates.
- **Output:**
  - *BCM: events related to shipment current status;* Notifications sent to BCM according to the final set of rules employed by the EPM for a specific TEP. These notifications let the BCM update status of all shipment monitoring parameters, and provide each role the alerts and data required by that role (see Figure 14 as an example of alerts pop-ups to end users).
  - *Frontend: notification on future events*; Some notifications are not required by the BCM – these are notifications that do not affect the status of any shipment parameter, but rather provide early warning on an event that is likely to affect a shipment in the future (e.g., a possible forecasted delay). This capability stems from a new paradigm called Proactive event driven computing, in which events (alerts in our case) are not triggered by past events, but by forecasted future events or situations [8].

6.1.4 Runtime Engine

The runtime engine is the piece of software that is responsible to detect during runtime which rules are matched. The engine of FInest will be designed according to the generic event processing network (EPN) architecture summarized by [6] and [7]. The EPN (as illustrated in Figure 13), is the conceptual view of the flow of events to, within, and from the runtime engine. The EPN consists of event producers and consumers, and the runtime engine with *event processing agents* and *channels*.

Event processing agents (EPA) are local processing units with a well-defined logical role. In FInest, an EPA corresponds to a rule in the rule base. Each EPA is responsible for the detection of the event pattern in the rule body, and emits derived events as a result according to the rule head. Because rules are selected and instantiated per execution, the EPAs will be created in runtime, according to the set of rules selected.

Channels are entity whose role is to collect and distribute events between objects in the system. They allow flexible routing of events from event producers to EPAs, between EPAs, and from EPAs to consumers. A channel can be implemented as a multicast protocol, or other message oriented middleware. The main idea is to avoid having each source (producer or EPA) connect to each event destination (EPA or consumer), but have all of them connect to a channel, which contains the routing logic.

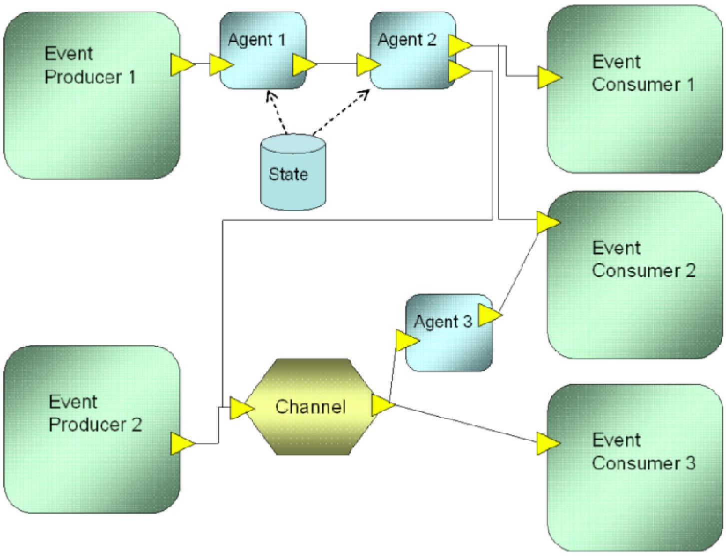

Figure 13: Structure and flow of an example EPN (Event Processing Network)

6.2 Demonstration of EPM Capabilities: Real-time booking update cancellation

In order to demonstrate the event processing module capabilities, the FInest project defined a specific demonstrator titled "real-time booking update cancellation". This demonstrator addresses the challenge of discrepancies between booked shipment and actual shipment which constitute a major source of shipment delays and under or over utilization of resources in today's scenarios of T&L.

The business motivation behind this demonstrator can be summarized as follows: the goal is to reduce the gap between booking and actual; the accomplishment of this goal would allow forwarders and carriers to better know what freight needs to be transported, allow shippers to be more certain of the space reserved to them by forwarders and carriers, and provide better visibility and predictability on the process they run; overall, much better quality of data. With the improved quality of data, the process will have less disturbances, decreased throughput times, and perhaps, most importantly, a better consolidation for the forwarder and increase of the asset utilization by the carriers.

One of the main causes for the gap is found to be the following: *shippers, carriers, and forwarders may not have the same data about a single booking.* Some of the possible reasons for this can be:

- Changes often occur in the delivery specification (volume, weight, number of pieces), and often at the last moment (e.g., initial measurements are not accurate, or items are repacked). Shippers sometimes do not inform on time about changes in the booking, for various reasons; for example:
  - There is no structured process to do that (or, as identified in root cause analysis: shippers are not asked to update changes in their shipping volume or weight when pickup time approaches).
  - Busy operational schedule (hence shippers either forget to update, or do not see this as high priority).
  - Human or technical errors in transferring data between different applications.

The demonstrator shows a solution to prevent the root causes mentioned above. It provides a single cloud location for the order, with the same details available to all parties at the same time, and supports structured information exchange that will allow smooth and collaborative changing of a booking.

There could be many other reasons for the gap, such as risk management of shippers, wrong measurements, and last minute changes. Another part of the demonstrator scenario addresses these cases: we reached the pickup time of the shipment, and the truck driver discovers a discrepancy between the booked and actual shipment. In that case the goal is to reduce the time gap in propagation of capacity information: the earlier all stakeholders in the chain are aware of the change, the better they can adapt. Examples of solutions are: change timing of the transport plan (and potentially postpone pickup), find other carriers, or change priority of different customers in the consolidation of a forwarder.

Figure 14 presents a snapshot of the described scenario in which the same information is viewed and alerts are shown as per user role.

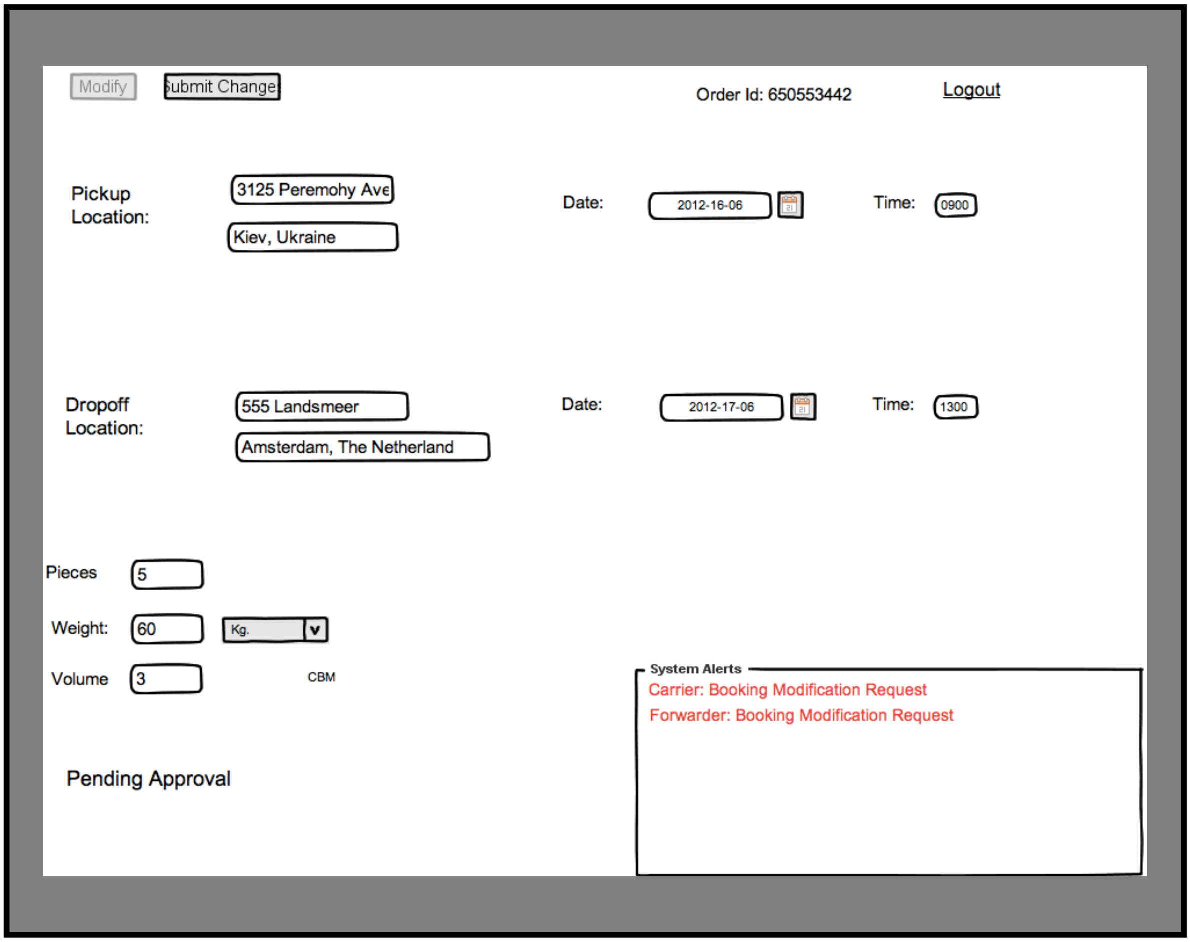

Figure 14: Real-time booking update cancellation screenshot

6.3 Novelty in Comparison to Standard Event-Processing Applications

The FInest project presents interesting challenges to the paradigm of event-processing. Most event driven applications are created on top of the engine and run "forever" (of course with various adaptations once in a while); for example: fraud detection in financial institutions, machinery condition monitoring for heavy industries, and operational management of call centers. The transport and logistics application, in contrast, is a system that monitors a large number of short-term executions (shipments), both concurrent and sequential. This creates two main challenges:

- There is no one specific rule set that is always valid. Each shipment may use a different subset of the rules. Moreover, the values within rules (for example, specific temperature that the shipment should be kept in, the specific time delay that becomes crucial) are specific to shipments. The proposed solution to this challenge in FInest is the 3-layered configurable rules system, detailed in Section 6.1.2.
- In standard event processing applications, the event processing network is static, and specific agents can be initiated when monitored objects are created (such as new customer). In FInest, in contrast, a new TEP requires a whole network of event processing agents to be initiated, supporting a unique set of rules, that is, the system should be able to initiate the set of EPAs for a specific TEP when execution starts, and terminate them when the execution ends.

In addition, proactive notifications popped-up to FInest front-end require extending the conventional event processing architecture to cope with future events and uncertainty (cf. [6].

7 Conclusions and Perspectives

Future Internet technologies and cloud based software delivery models offer a unique opportunity for the logistics profession to move forward in addressing historic issues associated with the international shipment of goods. These technologies and service models provide the ability to improve collaboration in the supply chain by removing barriers to system deployment, increasing transparency, encouraging new on-demand software applications, integrating the Internet of Things with users in real time and encouraging novel business models that capture value in new and innovative ways.

The FInest logistics collaboration platform is one potential approach to leveraging the emerging Future Internet technologies to the benefit of supply chain partners. This platform builds on Future Internet services to create a Web 2.0 business-to-business collaboration platform that can be used to link third party supply chain applications, such as those described as "core modules" above, to create customized service solutions for goods shipments.

The business scenarios that have been described and illustrated by the FInest mockups show that the services proposed by the FInest platform may only be the beginning of a rich eco-system of component services that are developed for the transport and logistics domain and that can be quickly assembled into value-added services in the collaboration manager to create "on demand" supply chain management solutions that are focused on the unique requirements of individual shipments.

While the potential to revolutionize how supply chains are managed exists in Future Internet enabled services such as that shown in the FInest project, significant work still is required to reap the benefits from these services. Prototyping and deployment of the FInest platform will undoubtedly show where additional work is required to achieve the vision that is behind the Future Internet.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 285598 (FInest).

8 Literature

1. Domingue, J.; Galis, A.; Gavras, A.; Zahariadis, T.; Lambert, D.; Cleary, F.; Daras, P.; Krco, S.; Müller, H.; Li, M.-S.; Schaffers, H.; Lotz, V.; Alvarez, F.; Stiller, B.; Karnouskos, S.; Avessta, S.; Nilsson, M. (Eds.) The Future Internet -- Future Internet Assembly 2011: Achievements and Technological Promises, Series: Lecture Notes in Computer Science, Vol. 6656, Subseries: Computer Communication Networks and Telecommunications, 1st Edition., 2011, XVI, 465 p., ISBN 978-3-642-20897-3
2. Bhattacharya, K.; Gerede, C.; Hull, R.; Liu, R.; and Su, J., "Towards Formal Analysis of Artifact-Centric Business Process Models," in G. Alonso, P. Dadam and M. Rosemann (Editors), Proceedings of the 5th International Conference on Business Process Management (BPM 2007), volume 4714 of Lecture Notes in Computer Science. Springer, 2007, pp. 288-304.
3. Resinas, M.; Fernandez, P.; and Corchuelo, R.; „Engineering Automated Negotiations," in J. Riquelme and P. Botella (Editors), XV Jornadas de Ingenieria del Software y Bases de Datos (JISBD 2006), CIMNE, Barcelona, 2006.
4. Schiefer, J. and Seufert, A.; „Management and Controlling of Time-Sensitive Business Processes with Sense & Respond," in Proceeding of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC 2005).
5. R. Franklin, A. Metzger, M. Stollberg, Y. Engel, K. Fjørtoft, R. Fleischhauer, C. Marquezan, and L. S. Ramstad, "Future Internet technology for the future of transport and logistics (invited)," in ServiceWave 2011, Future Internet PPP Track, ser. LNCS, A. Zisman, I. Llorente, M. Surridge, W. Abramowicz, and J. Vayssière, Eds., Springer, 2011.
6. Opher Etzion and Peter Niblett, Event Processing in Action, Manning, 2010.
7. Cathrine Moxey et al., A Conceptual model for Event Processing Systems, An IBM Redguide publication, http://www.redbooks.ibm.com/redpapers/pdfs/redp4642.pdf, 2010.
8. Yagil Engel and Opher Etzion, Towards Proactive Event Driven Computing, in Distributed Event-Based Systems (DEBS), 2011
9. A. Nigam and N. S. Caswell, "Business artifacts: An approach to operational specification," IBM Systems Journal, vol. 42, no. 3, pp. 428-445, 2003.
10. 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.
11. 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.
12. K. Erler, "Design and prototypical Implementation of an Entity-centric Collaboration Environment," Dresden University of Technology, 2011.
13. FI-WARE High-level Description (Product Vision), FI-WARE-11-08-01».
14. A. Vennesland, H. Westerheim, M. Rosén, C. Kjellberg, V. Mary, P. Wiman og J. T. Pedersen, «e-Freight D1.3b: e-Freight Framework - Information Models,» 2010.

15. www.foreui.com

16. Steinmetz, E., J. Collins, S. Jamison, R. Sudaresware, B. Mobasher, and M. Gini, "Bid Evaluation and Selection in the MAGNET Automated Contracting System," Department of Computer Science and Engineering, University of Minnesota Working Paper.