A Reference Architecture for Integration Platforms

Quan Wang

September 2016

UNIVERSITY OF TWENTE.

Development of a Reference Architecture for Integration Platforms

Author: Quan Wang
Supervisors: Marten Van Sinderen, Associate Professor
Prince Singh, PhD candidate
Maria-Eugenia Iacob, Adjunct Hoogleraar/Professor
Luís Ferreira Pires, Associate Professor
Department: EEMCS, University of Twente
Email: q.wang0122@hotmail.com
q.wang@student.utwente.nl

ABSTRACT

With the introduction of more and more distributed applications in enterprises, various integration platform products are designed and created to increase the interoperability between enterprise applications systems and devices. For example, different middleware technologies can facilitate communication and data management between distributed applications and devices. There are various types of middleware technologies, such as message oriented middleware (MOM), object request broker (ORB), remote procedure control (RPC), remote database access (RDA) and so on. As one of the most popular integration solutions, enterprise service bus (ESB) has gained an increasing popularity in this domain. It can not only integrate application systems, but is also extending to integrate various devices.

In order to facilitate the future development of integration platforms, a reference architecture of integration platforms is required. A reference architecture can provide guideline and design principles for developers to design system architectures. However, a general and systematic approach on designing integration platforms in literature is not existing yet. In our research, a systematic literature review was carried out to help develop a reference architecture for integration platforms. By using the reference architecture, developers of integration platforms no longer need to design from scratch. The reference architecture can be used as a template to design integration platforms. Besides, a case study in the domain of logistics was carried out to validate the correctness of the reference architecture.

This thesis started with a systematic literature review (SLR), which resulted in 31 research papers concerning integration platforms in different domains. After that, an insight into terms of "integration platforms" and "reference architectures" was respectively presented. The definition, characteristics and several instances of integration platforms were firstly described. Then, the definition, decomposition and abstraction level of reference architectures were discussed. To construct the reference architecture of integration platforms, components of integration platforms in research papers from the SLR result were extracted and collected altogether. Then all these components were classified into various categories based on their functionality described in the 31 research papers. Based on those categories, a conceptual model of integration platforms was designed. Besides, a design pattern tree was also created in the reference architecture to help designers choose appropriate design patterns of integration platforms. Finally, a case study in the domain of logistics was studied to validate the correctness of the reference architecture.

Keywords*—Integration platforms; Reference architecture; Data integration; Enterprise application integration; Architecture*

Table of Contents

Figure List

Table List

Textbox List

1. INTRODUCTION

1.1. Background

With the increased applicability of “internet of things”, more and more physical objects are being connected together to exchange and share data and to cooperate for a specific task. Similarly, enterprise information systems are also being integrated together to exchange and share data and to collaborate for a common business goal. For instance, the German Indonesian Tsunami Early Warning System (GITEWS) [1] integrated sensors, sensor systems, web services and other application systems altogether to monitoring tsunami. Data and information provided by each object can be used by others. Services provided by each object can be combined with other services to reach a common goal. In this case, the task of monitoring tsunami can not be accomplished by any one object. It has to be realized by the combination of all objects which are connected by the integration platform. Hence, integration of application systems and other devices has become more and more prevalent and important.

To achieve communication and collaboration between different objects in enterprises, the idea of “integration platforms” is a general solution. Integration platform (IP) is defined as a computer software which integrates different applications and services, whose interfaces are integrated so that interoperability can be achieved among an enterprise or across different enterprises. An IP can be a small adapter when there are only a few objects to be integrated. In this case, one object is connected to another one by an adapter. Besides, IPs can be realized by various middleware, such as message oriented middleware (MOM), remote procedure control (RPC) middleware and object request brokers (ORBs) middleware. Database access services such as JDBC and ODBC also belong to middleware. As a popular and prominent integration product, enterprise service bus (ESB) is provided and used for many enterprises because of its abundant functions. Except the function of integration, an ESB product normally has some extra functions such as data format transformation, security, service orchestration and so on. In the example of GITEWS, various sensors, sensor systems, web services and other application systems are connected by the tsunami service bus. They can exchange data and combine business processes through the service bus. As a result, enterprise devices, application systems and other platforms are integrated together.

One of the vital advantages of IPs is that data and services can be integrated and shared between different departments within an enterprise or across enterprises. This benefit is able to reduce much labor consumption and time consumption in some cases. For Booly IP [2], data can be shared and exchanged by medical services from a data warehouse. Moreover, an IP can be designed based on enterprise business process and rules according to enterprise business goals and requirements, which can avoid “island” of different applications and increase efficiency of enterprise performance. For example, the IP in [3] uses an ESB to integrate and orchestrate external services to reach a common business goal. The component of “service orchestration” or “business process management” is the core logic to organize and control business processes. Hence, the introduction of IPs in enterprises and other domains becomes a necessary and important step.

1.2. Problem statements

Problem Statement 1: how to apply the method of designing IPs in domain "A" to IPs in domain "B"?

IPs nowadays can be found everywhere in different domains, such as healthcare, environment, education, transportation and etc. For instance, Jinxing Xu and Qi Ma [4] designed an IP in the domain of transportation by combing GIS and SOA technologies to integrate geography related information; P.Y.Wang and J.Shen [5] described a procurement IP in the domain of government which is designed by using cloud services to integrate government procurement related services and information; Xiaolong Fu and Qixin Liu [6] designed a resource sharing IP in the domain of education to solve the problem of informatization in higher education institutions; Zhexi Yang and Ying Zhao [7] presented a steel circulation IP in the domain of supply chain to share resources. As these IPs are built for different business goals and reside in different domains, they may have different business flows, data structures and implementation languages. Hence, it is inapplicable to apply the method of designing IPs in domain "A" to IPs in domain "B".

Problem Statement 2: how to deal with compatibility between IPs and various kinds of external objects?

Another problem is that an IP usually needs to integrate various objects, such as application systems, local web services, cloud services and other devices. These objects usually have different data formats, computing ability, APIs and so on. Integrating different objects with an IP should use different methods and technologies. Besides, the compatibility between IPs and new developed applications has to be taken into account in advance for some cases. How to deal with compatibility between the IP and external objects needs to be solved.

Problem Statement 3: how to choose the most appropriate design pattern or technologies for designing IPs?

Moreover, there are different methods, technologies and products which can be used and combined to design IPs, such as service oriented architecture (SOA), EDA, IPaaS and so on. As mentioned before, even an adapter can be treated as a small IP which can connect two objects together. For more objects, an enterprise service bus (ESB) can be introduced. Each object is connected with ESB by means of an interface. The ESB is responsible for exchanging data and information between these objects. Hence, the third problem of the research is that there are many options for designing IPs. How to choose the most appropriate designing method and how to figure out the relationship between designing patterns need to be dealt with.

The three research problems discussed above make it difficult to apply the architecture of IPs in one domain to design IPs in another domain. Hence, developers with tasks of designing new IPs in a specific domain need to design from scratch when there is no available template of IPs in that domain. As a result, the reusability of designing experience is limited. Hence, architects who need to design a new architecture of IPS are at most times forced to design from scratch, without applying previous design experiences or relevant rules, which will cost more time and labor resources.

1.3. Goal

The goal of the research is to develop a reference architecture (RA) for IPs to enhance the efficiency and reusability of designing IPs in any domain. The RA should be able to solve problems discussed above. The RA should provide guidelines and principles of building IPs in different domains in the future. It should contain common elements for all IPs and some essential elements for some specific IPs. The RA should be used as a foundation for developers who are tasked with development of IPs. As an application of the designed RA, we will apply the RA to develop an IP based on a case study.

1.4. Research questions

Based on problem statements proposed in section 1.2, the goal of the research was described in section 1.3 to solve these problems. To reach the research goal, the following four research questions (RQ) are proposed.

RQ 1: What are the existing IPs in different domains in literature?

In order to build up the RA of IPs based on existing IPs, it is necessary to study existing IPs in literature. This research question will be addressed in Chapter 3 via a systematic literature review (SLR). The result of the SLR consists of a list of research papers which presents a specific IP.

RQ 2: What are the common elements of IPs?

As aforementioned, there might be some common elements of IPs in different domains. These common elements comprise design patterns and building blocks or components of IPs. Design patterns of IPs are solutions and approaches on implementing them. Building blocks are components which comprise an IP. Common elements of IPs are essential inputs for the RA. Hence, identifying common elements of IPs in each research paper is very crucial. This research question will be dealt with in Chapter 6.

RQ 3: What would be the appropriate method to design the RA of IPs based on those commonalities?

The most crucial task in this research is to build up a RA for all IPs. To build up the RA, common elements collected from RQ 2 will be used. These common elements will be filtered and organized as an input for the RA. The standards and methods used for filtering and organizing these common elements become an essential problem. This research question will also be addressed in Chapter 6.

RQ 4: Does the IP in the case study align with the RA?

To validate the correctness of the RA, a study case in the domain of logistics will be adopted. As there are various enterprises, actors, applications and devices involved in logistics, IPs in this domain also encounter problem statements described in section 1.3. This research question will be dealt with in Chapter 7.

1.5. Structure

The research paper is structured as follows. Chapter 2 will describe the research methodology applied for designing the RA of IPs. In Chapter 3, a SLR will be carried out to collect IPs in literature for designing the RA, resulting a list of research papers. In Chapter 4, an insight to the concept of IPs will be presented while the concept of RAs will be presented in Chapter 5. In Chapter 6, the construction of the RA will be done in detail. In Chapter 7, the validation of the RA will be given by using a logistics study case. In Chapter 8, we will discuss about our work. In Chapter 9, we will give the conclusion of our work.

As the following words appear in the research paper for many times, they will be used in the format of abbreviation. These abbreviations are shown in Textbox 1.

Textbox 1. List of abbreviations

IP: Integration platform

RA: Reference architecture

SOA: Service oriented architecture

SLR: Systematic literature review

ESB: Enterprise service bus

2. RESEARCH METHODOLOGY

To address the four research questions proposed in section 1.4, three research steps are used throughout the whole research, which is shown in Figure 1. Firstly, a systematic literature review (SLR) will be carried out. The SLR method is based on Andy Siddaway [8]. It produces a list of IPs from literature which covers different domains, such as medication, transportation, supply chain, government, environment, and so on. These IPs are used as data sources for developing the RA in step 2.

Secondly, the method of developing RAs based on Vanessa Strickera and Kim Lauenrotha [9] will be introduced to extract common elements from data sources. These common elements include building blocks of IPs which will be collected and classified based on their functionalities. They also include design patterns which are used for implementing these IPs. The output of the second step is an initial RA for IPs. The initial RA for IPs will be used as a template for validation in step 3.

Thirdly, the validation of the initial RA will be performed, with a final RA as the output. The validation is carried out by a study case (see Appendix D) in the domain of logistics. Some parts of the initial RA for IPs will be altered and some elements will be added or deleted after validation. The final RA for IPs should be able to provide guidelines and principles for developers to design IPs in all domains.

Each research step shown in Figure 1 addresses one or two research questions. The SLR deals with research question 1: "*What are the existing IPs in different domains in literature?"* The SLR will produce a list of existing IPs in literature. The development of the RA for IPs deals with research question 2& 3: "*What are the common elements of IPs?"&" What would be the appropriate method to design the RA of IPs based on those commonalities?"* The development of the RA produces common elements of existing IPs as well as an initial RA for IPs. The validation of the RA deals with research question 4:" *Does the IP in the case study align with the RA?*" The validation of the RA produces a final RA for IPs.

In order to apply the methodology in the research correctly, each research step is described and illustrated in detail in the following sections.

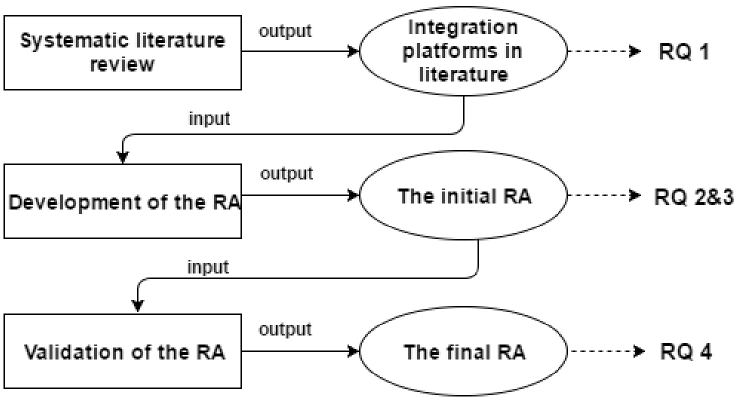


Fig. 1. Research methodology throughout the research

2.1. Systematic literature review

A SLR is "a review of the evidence on a clearly formulated question that uses systematic and explicit methods to identify, select and critically appraise relevant primary research, and to extract and analyze data from the studies that are included in the review [10]." To carry out a SLR, the following steps in Figure 2 are adopted based on the method from [Andy Siddaway].

Firstly, a systematic literature review research question will be formulated. The SLR research question should focus on problems that the research will deal with. Secondly, search terms will be extracted from the SLR research question. Normally, some key words or key terms are required when searching online or from libraries. These key words or key terms are the most essential words from the SLR research question. Thirdly, inclusion and exclusion criteria are formulated and will be applied in the last step. These criteria help remove duplicated and useless results and extract useful results. Fourthly, the searching process will be carried out from several databases, with a list of research papers as an initial searching result. Lastly, aforementioned inclusion and exclusion criteria will be introduced, with a fewer list of research papers as the final result.

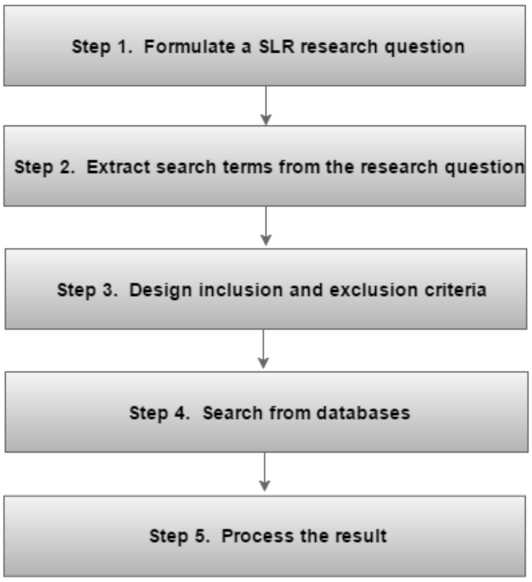


Fig. 2. The systematic literature review process

2.2. Development of the RA

The method used to develop the RA is a pattern-based RA [Vanessa Stricker]. The method is the most appropriate approach of designing RAs for our research after taking some other approaches into consideration. Before deciding to apply this method, some other papers were also consulted. For instance, Andre Vasconcelos [11] presented a RA for the Portuguese health sector; Paper [12] described a RA for the government applications; Ulrich Meisse [13] proposed a RA for Early Warning Systems (EWS). Although these RAs all provide templates for designing and building architectures of information systems, they each only focus on one specific domain. The RA in domain "A" can not be adopted for designing a RA in domain "B".

Moreover, they are not pattern-based. According to the research question 3 proposed in section 1.4, we should figure out the most appropriate design pattern or technologies. These RA methods deviate from our problem statement 3. Hence, the idea of building the RA for IPs is derived from [Vanessa Sticker], which can be seen in Figure 3. It consists of three components: guidelines and principles, the reference model and the reference specification. Each component of the RA are described in detail below.

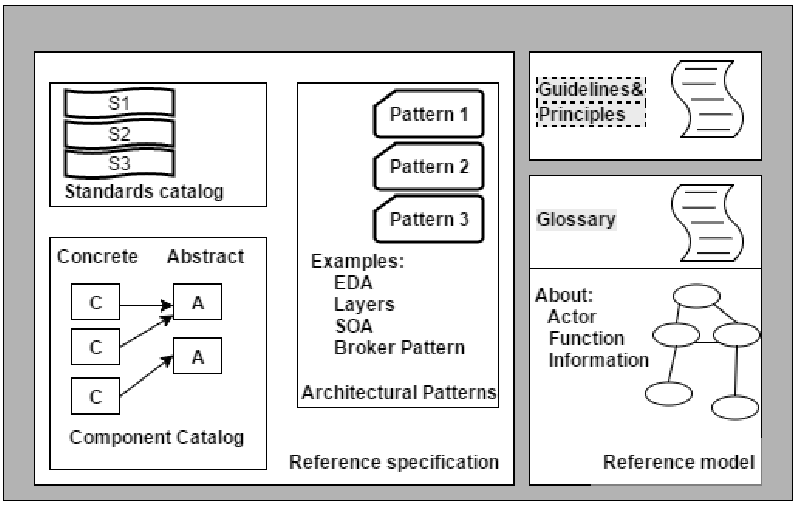


Fig. 3. Structure of the pattern-based reference architecture [9]

The guidelines and principles: This part of the RA captures two components. Firstly, it captures the guidelines of building up a specific architecture based on its requirements. As the RA provide a generic architectural solution, it should be able to present a decision tree for developers to choose an appropriate solution based on their knowledge and design requirements. The guidelines of the reference architecture perform an essential role in being a decision tree. Secondly, it captures the principles of constructing the framework as well as the properties related to each component in the reference architecture. The principles are principles of choosing appropriate design patterns. They help developers choose the most appropriate technologies and products based on their situation and requirements.

The reference architecture model: The reference architecture model contains two components: glossary and the conceptual model. The glossary defines essential terms which will be partly used in constructing or describing the RA. The conceptual model presents important entities that constitute the IP as well as the interactions between these entities. In our research, the conceptual model describes the IP from different views: structure, behavior and functionality [14]. The structure shows the internal components of IPs. The behavior shows the relationship and interaction between these components. The functionality is referred to each component.

The reference architecture specification: This part contains three components: standards catalog, component catalog and the architectural design patterns. The standards catalog provides the standards which refer to the guidelines and principles. It defines the rules of choosing components of integration platforms. The components catalog provides description of both concrete and abstract products. Each concrete product will be mapped into an abstract product. Besides, each description refers to standards, concepts and behavioral characteristics. The architectural design patterns present several approaches of designing the systems, which is the core science of the reference architecture. For example, SOA, EDA, Layered design pattern are all popular design patterns involved for integration.

2.3. Validation of the RA

In order to validate the RA, a study case in the domain of logistics [Janssen Logistics b.v] will be applied in Chapter 7. The method used for validation is presented in Figure 4.

Firstly, a baseline architecture of Janssen Logistics b.v will be presented. The baseline architecture will be discussed from the perspective of actors, business processes, applications and data storage. Secondly, some problems are identified and proposed in Janssen Logistics b.v based on the baseline architecture. To solve these problems, the RA developed in Chapter 6 should be used. Thirdly, an IP for Janssen Logistics b.v will be designed based on the RA. Fourthly, we will check whether the problems proposed in the second step has been solved by the IP.

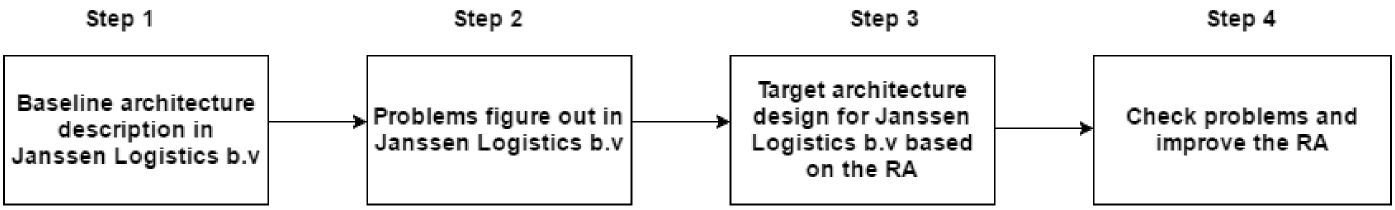


Fig. 4. The validation process for the RA

3. SYSTEMATIC LITERATURE REVIEW

In this chapter, a SLR is illustrated to deal with the first research question mentioned in Chapter 1: *What are the existing integration platforms in different domains in literature?* The systematic literature review is based on the approach from [8], which has been described in Chapter 2. In this Chapter, we will carry out the detailed process, operations and results.

3.1. The systematic literature review process

Step 1. Formulate a SLR research question.

As the SLR is used to deal with the research question 1 proposed in section 1.4, we can use it as the SLR research question. Hence, the SLR research question becomes: *"What are the existing IPs in different domains in literature?"* The answer of the SLR research question is a list of research papers concerning IPs in different domains.

Step 2. Extract search terms from the SLR research question.

According to the SLR research question, "integration" and "platforms" are extracted as the search terms as they related to what we need to figure out. In order to increase the relevance of the searching result, "integration platforms" is used as the exact phrase to search the relevant papers. Besides, some synonyms like "collaborative platforms" or "cooperative systems" are discarded as they give a huge number of results. Besides, they sometimes also give irrelevant results. For example, they may give many results in the area of biochemistry model or physical structure.

Step 3. Design inclusion and exclusion criteria

To discard irrelevant papers and obtain the searching result in high relevance, the following three inclusion and exclusion criteria are formulated to filter the original searching result.

(1) Exclusion 1. Duplicated papers extracted from different sources are merged together and only one is remained.

(2) Inclusion. The paper is about integration platforms in the area of information systems.

(3) Exclusion 2. Papers that do not have sufficient content discussing architectures of IPs are removed. For instance, some papers that only discuss techniques of building up an IP architecture are discarded.

Step 4. Search from databases

There are many online databases which are specific for information and communication technologies, among which the following four electronic databases are chosen: IEEE xplore, Web of Science, Scopus and Springer link.

When searching on different databases, some rules are applied. Firstly, the search term is applied to search in title only. Secondly, the filter of "written language in English" is applied. Thirdly, the filter of "publication year between 2010 and 2016" is applied to avoid obsolete information and technologies.

Step 5. Inspect results

The total number of result from all databases was 341, among which 70 from IEEE xplore, 95 from Web of Science, 172 from Scopus and 4 from Springer Link. After removing duplicated papers, the amount decreased to 188. After applying the inclusion criteria, the amount decreased to 72. After applying the second exclusion criteria, the amount decreased to 31. The removed 41 papers had no detailed description of architectures of IPs.

By applying inclusion and exclusion criteria, the result has been decreased in each round. The searching result after each process of inclusion and exclusion can be seen in Figure 5. The final result after inclusion and exclusion criteria is given in Appendix A.

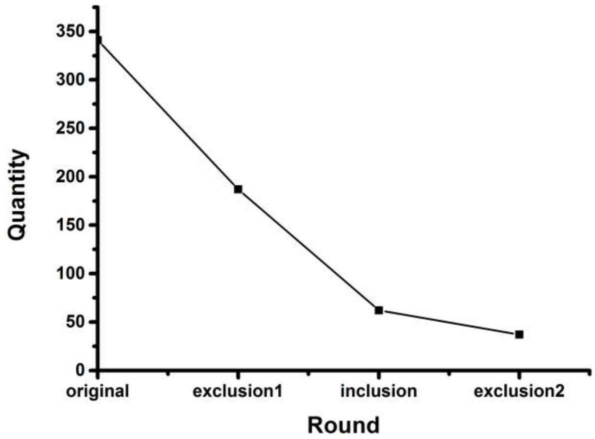


Fig. 5. The process of inspecting the searching result

3.2. Classification of the SLR result

As mentioned in previous, our RA should be feasible for designing IPs in all domains. In order to find commonalities of IPs between different domains, it is efficient to classify these research papers based on domains. The classification can provide an easier way of comparison of IPs between the covered domains. For example, IPs extracted from the SLR are from various domains, such as transportation, medication, government and so on. Researchers can analyze the relationship of IPs between different domains based on the classification result. The classification of the SLR result based on domains is shown in TABLE I. The first column presents the category number while the second column shows categories based on domains. The third column lists relevant paper number corresponding to categories while the fourth column presents the number of papers in each domain.

As we can see in the table, IPs from the SLR involve various domains. Among them, there are 7 domains covering at least 2 papers: general enterprise systems, supply chain, government, medication, campus, biology and transportation. Besides, there are 9 papers which presents integration platforms in other domains, such as smart cars, music applications, manufacturing and so on.

TABLE I. CLASSIFICATION OF INTEGRATION PLATFORMS BASED ON DOMAINS

CategoryNo.	Category	Paper No.	Total papers
1	Enterprise systems (5)	3, 5, 6, 11, 21	5
2	Supply chain (4)	15, 24, 28, 29	4
3	Government (2)	12, 26	2
4	Medication (3)	17, 18, 19	3
5	Campus information(2)	14, 30	2
6	Biology (4)	10, 16, 25, 31	4
7	Transportation (2)	4, 9	2
8	Other (9)	1, 2, 7, 8, 13, 20, 22, 23, 27	9

In **Category 1**, all 5 papers are related to integration of enterprise application systems. Among them, paper 3 proposed an extensible platform architecture for portable cloud service integration, with implementation of access control policies and mechanisms for sharing resources. Paper 5 described an information integration based on workflows (IIBW) platform which offered an integrated access to heterogeneous resources. Paper 6 presented a XML/ Agent based IP. Each industry process operation subsystem is encapsulated as an agent. Based on business goals, different agents are integrated together using agent technologies. Paper 11 developed a business processes oriented heterogeneous IP. In paper 21, the combination of SOA and web services was applied to design an application IP, which solved the problems of poor information sharing capabilities and business adaptabilities.

In **Category 2**, there are four papers dealing with IPs in the domain of supply chains. Paper 15 proposed a data exchange platform based on EAI to solve the problem of data exchange between steel circulation supply chain applications. Paper 24 discussed a supply chain IP for remanufacturing. Paper 28 presented an agent based IP for supply chain logistics. Paper 29 described a supply chain IP in textile industry to solve the problems of high transaction costs, long transaction process and irrational resource collection.

In **Category 3**, both papers of 12 and 26 are about government information and resources IPs. Paper 12 provided a cloud based IP for government procurement to solve the problems of procurement information management. The IP included three layers: physical resources, virtual resources control and cloud services. Paper 26 presented a SOA based IP for government. It was realized by an ESB and web services.

In **Category 4**, all three papers of 17, 18 and 19 discuss about medical information integration. Paper 17 proposed a healthcare IP to tackle the issue of exchanging and integrating information from different healthcare information systems. Paper 18 presented a healthcare IP to solve the problem of lack of generality and lack of focus on wireless components. It also solved the problem of difficulty to integrate with preexisting research.

In **Category 5**, both papers of 14 and 30 are about integration of campus information systems. Paper 14 discussed the design of information system IP at the university level. Paper 30 proposed an IP to aggregate and manage campus information.

In **Category 6**, all four papers of 10, 16, 25 and 31 handle problems of biology information integration. Paper 10 presented a data IP which featured a flexible data model to offer powerful comparative analysis. Paper 16 developed a data IP supporting collaborative research projects. Paper 25 proposed a reference model for designing data IPs in biomedical research. Paper 31 presented a middleware based IP to integrate bioinformatics services.

In In **Category 7**, both two papers of 4 and 9 are about transportation. Paper 4 deals with IP for GIService-based transportation while paper 9 handles IPs for warship information systems.

Besides, a classification of the searching result based on relevance of usage is carried out. Three categories are identified: “Most relevant”, “Medium relevant” and “Least relevant”. The classification result and the classification criteria are given in Appendix B.

4. INTEGRATION PLATFORMS

Integration platforms have appeared in our daily life for a long time in different domains. However, it means different things in different domains. For instance, the IP presented by Felix Dreher and Thomas Kreitler [15] is composed of Foswiki, Solr/ Lucene and some other helper applications for storing and sharing data; S. Krishnamoorthy and A. Agrawala [16] described an IP which is made up of three layers: service tier, interface tier and context tier; Wu Deng and Huimin Zhao [17] showed an IP which is composed of RFID devices, SOA-based components and other application systems. These IPs are designed using different technologies and for different purposes. To have a better understanding of the concept of IPs, an insight to some existing IPs is needed.

In this section, the definition of IPs from different sources will be given in section 4.1, together with main characteristics of IPs in section 4.2. Besides, three existing architectures of IPs will be analyzed to demonstrate these characteristics in section 4.3.

4.1. Definition of integration platforms

To explain the term of IPs in more detail, some definitions of IPs from different sources are given here. For example**,** an IP can be defined as a computer software which integrates different applications and services [18]. An IP can be created from components or directly purchased as a pre-built product ready for installation or procured from an integration Platform as a Service (iPaaS) offering. In this sub-section, IPs from Informatica, TIBCOand Cisco are selected to give a clear exhibition of IPs.

Informatica: Informatica [19] provides a product of Integration Platform as a Service (iPaaS), which is a cloud IP for both application integration and data integration. As presented in Figure 6, Informatica cloud IP provides the best connectivity to SaaS (Software as a Service) and SaaS applications. Developers can use different data formats, such as JSON, OData, XML or SOAP web services protocol to realize applications. Besides, it encompasses batch and real-time integration services. Informatica cloud IP can help to deliver business data to the right device at right time and operate business processes over different applications at real time. Moreover, it provides a strong API framework for developers and end users to add new functionalities to applications in a less complex way. Finally, it provides robust cloud security and administration.

TIBCO: The TIBCO Integration Platform [20] is another famous IP which acts as a foundation of TIBCO's Fast Data solution. The TIBCO IP is illustrated in Figure 7. As can be seen from the figure, the TIBCO IP provides abundant connectivity to people, processes and systems. It is composed of several internal components: event processing, analytics, business process management, master data management, enterprise service bus, API management and data distribution. The data distribution module consists of managed file transfer, business-to-business, enterprise messaging and web messaging. The entities that can be integrated by the integration platform include people, cloud services and IT systems and applications.

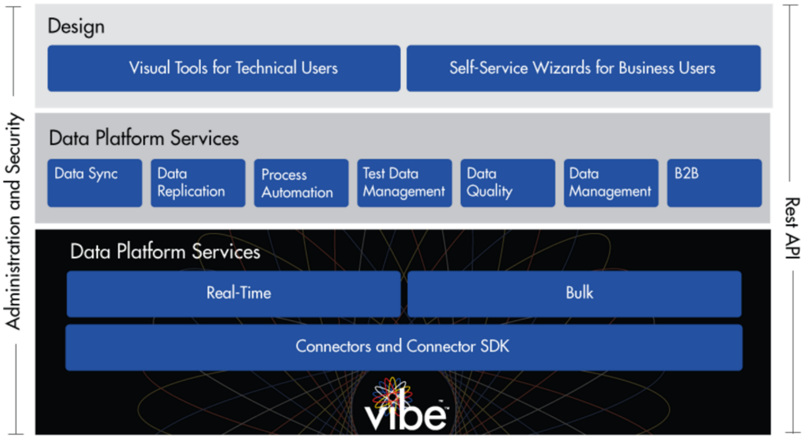


Fig. 6. Informatica iPaaS integration platform [19]

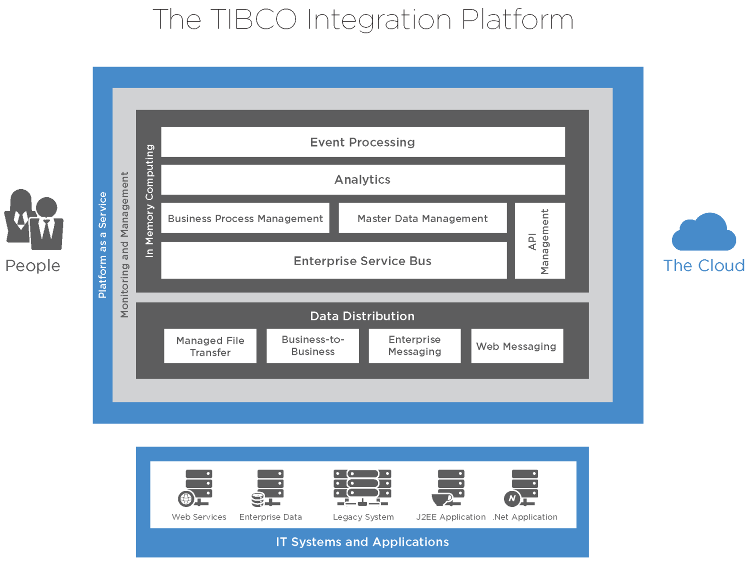


Fig. 7. TIBCO integration platform [20]

Cisco: Cisco Integration Platform [21] facilitates the business processes that are operated by distributed applications and the business data that should be shared by different objects. These functionalities are achieved through a lightweight service bus and end-to-end API management. The advantages of the Cisco IP comprises three aspects. Firstly, it decreases the platform deployment period. It has over 150 prebuilt connectors, graphical design tools and mixed-model deployment support to integrate any application or deploy a service as an API. The Cisco IP provides two deployment options: software only and private cloud. Secondly, the lightweight service bus and end-to-end API management ensure the rapid delivery of business outcomes and enhance business agility. Thirdly, the Cisco IP takes advantages of legacy and new designed applications, which enables business process more simplified.

4.2. Characteristics of integration platforms

An IP may comprise different components, such as data sources, computation ability or decision making capabilities. Normally, they are distributed or shared over distributed systems. For instance, among the healthcare IP proposed by Jerzy Brzezinski [22], there are sources of medical information, mediator, authorization, registry and index which can be introduced by different applications. These components in an IP can communicate and cooperate in a faster and more efficient way, which can improve the interoperability of systems involved and fulfill business needs. Three main characteristics of IPs are listed as follows.

1. Data integration. The integration of data is realized by collecting data from external objects, such as cloud services, application systems and sensors and other devices. These cloud services, application systems, sensors and devices are integrated by IPs. External objects send requests to IPs while IPs return back data to them. By collecting data from external object, each object that is integrated by IPs can share common data, which helps store, manage and process datasets and increases the performs of information systems.

2. Support business process. For service integration, there is often a service orchestration module in IPs. The service orchestration module or business process management module is responsible for orchestrating various services as a business flow to reach a common business goal. Service orchestration can on one hand increase business efficiency, it can on another hand save much labor consumption. As a result, different services, applications and devices are bound and collaborate together to support workflows or business processes within or outside an enterprise.

3. A common façade for integrated objects. For application integration, it is normal to realize integration at the level of interface. One common interactive interface is provided to replace other interfaces of applications or devices, with the same functionality remained. This characteristic can help to reduce complexity and manage the IP.

In order to better illustrate and prove these three characteristics of IPs, the following three existing IPs in TABLE II are separately analyzed. The first column represents the paper number, which is corresponded to the SLR result in Appendix A. The second column presents the title of the paper. The third column shows the characteristic that the paper proves. Each paper is used to illustrate one specific characteristic of integration platforms. Paper 17 proves the ability of data integration; Paper 5 proves the ability of business process management and paper 7 proves the ability of building common façade of integration platforms.

4.3. Integration platforms in existing literatures

As mentioned above, three papers: *"Healthcare integration platform", "A new information integration platform based on workflows" and "An approach for developing a mobile accessed music search integration platform"* were selected from the SLR result to explain the characteristics of IPs respectively. Paper 17 discusses a healthcare IP which aims at tackling the issue of exchanging and integrating medical information originating from different healthcare information systems. Paper 5 deals with a new information IP based on workflows. Paper 5 discusses an IP for Android users to search music from different music searchers. In the following, three research papers are respectively discussed to show how they prove the each of the characteristic of IPs proposed in last section.

TABLE II. EXISTING INTEGRATION PLATFORMS TO PROVE CHARACTERISTICS (SEE APPENDIX A)

No.	Title	Characteristic
17	Healthcare Integration Platform	Data integration
5	A new Information Integration platform Based on Workflows	Business process management
7	An approach for Developing a Mobile Accessed Music Search Integration Platform	Common facade

4.3.1. Healthcare integration platform

In order to prove the first characteristic of IPs mentioned in last section, a healthcare IP solution proposed by Jerzy Brzezinski and Stanislaw Czajka [22] is introduced. The "Healthcare integration platform" is aimed to provide patients with access to their medical records, which are extremely important information for them. Two standards are adopted to represent the medical data and communication separately. The first standard is HL7 CDA, which is a Clinical Documentation Architecture designed by HL7 [23] to enable the exchange of medical messages in XML format. The second standard is based on concepts from IHE (Integration Healthcare Enterprise) [24], which is an organization aimed to provide rules for the application of different medical standards.

The healthcare IP uses web services to achieve different functions related to medical information. The detailed services structure is based on IHE XDS (Cross-Enterprise Document Sharing), which is an architecture depicted in Figure 8 for the exchange of medical records. The healthcare IP is mainly made up of several web services: *source service, index service, registery service, authorization service and meidatior service.* To well illustrate the working process of medical information exchange, store and share, the detailed service functions and interactions between services are explained.

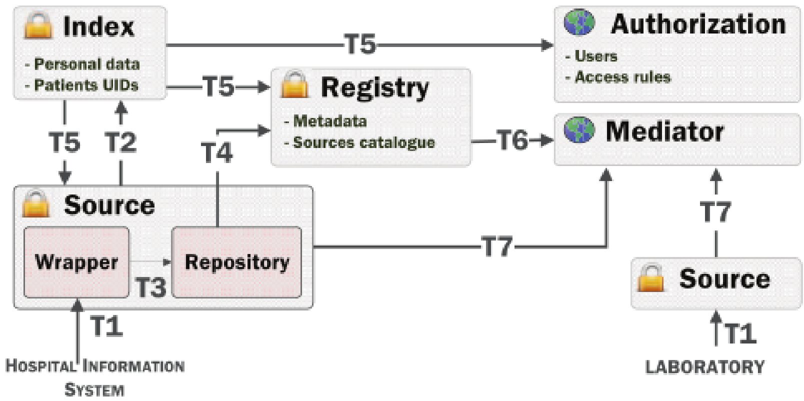


Fig. 8. Services arrangement and interactions based on the IHE XDS [22]

Source: The *Source Service* mainly has three functions in operating medical data and information: (1) obtaining medical data from differnet healthcare units; (2) subimiting perosnal information to *Index Service* to faciliate identifying from different healthcare units; (3) enabling documents able to be retrieved. To obtain medical data from different medical units, a wrapper is applied in order to convert healthcare data in different format into a document in a standard format. To submit personal medical information to *Index Service*, the communication adopts HL7 ADT (Admission, Discharge, Transfer), which guarantees interoperability within information systems. In order to enable documents able to be retrieved, *Source Service* acts as a document repositroy or a mediator in retrieving medical information from different healthcare units.

Index: The *Index Service* is aimed to indentify patients by assigning each patient a UID (Unique identifier). The *Index Service* can faciliate users to retrieve their personal information from the healthcare IP. In order to ensure that information of a patient in different facilities can be binded, it is necessary to cross-reference patients' identity, which is done based on IHE PIX profile [25].

Registry: The *Registry Service* is aimed to organize doucments' metadata. As medical servcies are provided by different medical facilities, the medical data is in different format. The *Registry Service* gathers information about the location of documents.

Authorization: The *Authorization Service* is aimed to secure the data by providing different access rights to different users. The access right to certain data can be in a temporary time with selected data format.

Mediator: The *Mediator Service* is a bridge connecting data shared by the platform and outside applications.

From the "Healthcare Integration Platform", we can see that data integration is achieved. Different users have different access rights to different healthcare units, which ensures the security. Besides, index and registry are utilized to organize medical document and metadata.

4.3.2. A new information integration platform based on workflows

To demonstrate the second characteristic of IPs- supporting business process, the paper of "A new information integration platform based on workflows" from Hu Zhiping and Zhang Yi [26] is introduced. In this paper, a new information IP based on workflows is presented. As mentioned in the paper, the information integration based on workflows (IIBW) platform provides integrated access to multiple distributed heterogeneous information sources, thus giving the illusion of a centralized, homogeneous information system.

In order to explain how IIBM platform supports business processes or workflows, the architecture of the IP is shown in Figure 9. The platform mainly consists of two parts: data fusion and workflows. The former one gives a uniform data format while the latter plans and executes a set of queries.

Data fusion system provides data schemas as unified format to upper applications. The data schemas semantic rules that not only present description to upper application in natural language, but also provide data sources with metadata descriptions as easily as possible.

The workflow system plans and executes a set of queries from applications. Applications send a request to the IIBW Platform. The request consists of information from applications as well as information needs to be retrieved by the platform. The planning system automatically constructs an execution plan for retrieving required information by combing information sources and input/output rules.

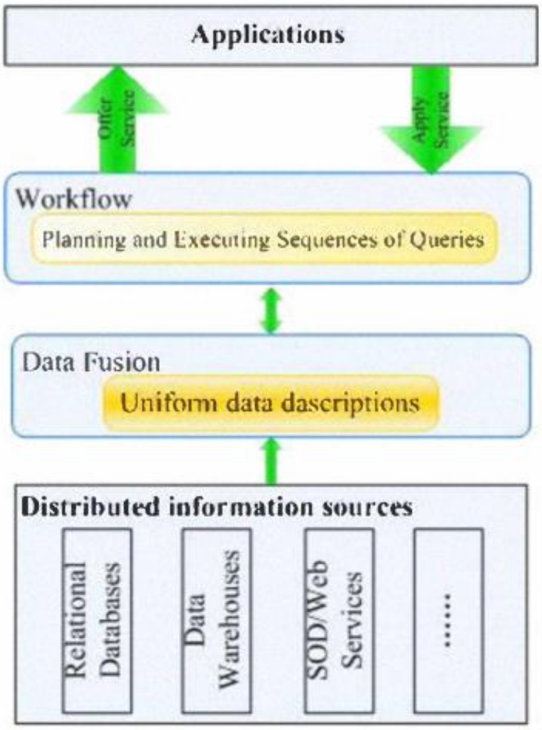


Fig. 9. Architecture of IIBW platform [26]

4.3.3. An Approach for Developing a Mobile Accessed Music Search Integration Platform

This paper of “An approach for developing a mobile accessed music search integration platform” from Marina Purgina and Andrey Kuznetsov [27] can be used to prove the third characteristic of integration platforms- a common façade for integrated objects. In this research paper, a mobile accessed application is developed as the IP to enable Android users to access several music searchers including *Musipedia, Music Ngram Viewer*, and *FolkTuneFinder*. As different music searchers provide different input query formats, the IP supports different query styles. These query styles are transformed into an appropriate one through a common interface.

To have a better understanding of the music access IP, the structure of the IP is presented in Figure 10. As can be seen in the Figure, the query input style can be chosen from the interface provided by the activity of *InputStyleSelection*. Based on the query input style returned from the previous activity, another activity (*MelodyContour*, *MusicScore*, *VirtualPiano*, and *Rhythm Tapper*) is activated and another interface is provided. Then the user input is transferred to three music searchers: *Musipedia*, *Music Ngram Viewer*, and *FolkTuneFinder*. They transform the user input an appropriate format that they support.

From the IP we can see that a common façade is realized to integrate different objects. The activity of *InputStyleSelection* provides common interface, which acts as a common facade for users to choose their music query input styles. Until now, the third characteristic of the integration platform is well proved.

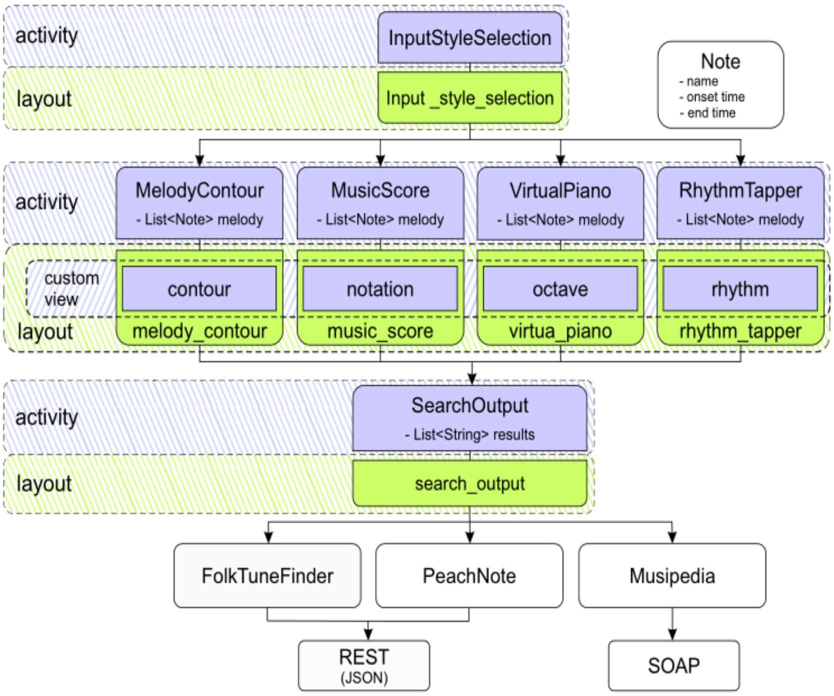


Fig. 10. Architecture of Music access integration platform [27]

4.4. Learning from integration platforms

In this Chapter, we learned a lot about IPs. On one hand, we can see that an IP can be defined as a software which integrates application and devices. It can be used for both application integration and data integration. On the other hand, we learned three important characteristics of IPs. Firstly, IPs can facilitate data sharing and exchanging between applications, devices and services by providing a data flow bridge. Secondly, IPs also realize business process management by combining and orchestrating each business process provided by applications or devices. Thirdly, IPs provide a common façade for all integrated objects by providing a common user interface and APIs for connecting with objects.

To learn more about IPs, the study of SLR results provided in Appendix A can be done. Moreover, a RA for IPs will be developed in Chapter 6 by studying the SLR results. The RA will give more knowledge about IPs, such as design patterns, building blocks, reference model and so on.

5. REFERENCE ARCHITECTURE

As the goal of our research is to build up a RA for IPs, it is important for us to understand the term as well as properties of it. In this section, we first collect definitions of RA from different sources, which can give a clear view and understanding of it. Then, decomposition of a RA will be presented based on a specific instance. Finally, the abstraction level of a RA will be presented.

5.1. Definition

RAs are used in various domains. As our research focuses on the area of IPs for information systems, the concept of RA is addressed from the perspective of information systems. Different sources give different definitions and explanations of it. Sun Computer [28] states that:

> *Sun's Reference Architectures has been designed, tested, tuned and documented, so you can reduce the complexity, costs, and risks of deploying new technology in your enterprise. Before choosing to implement a Reference Architecture, you can test a proof-of-concept system at a Sun Solution Centre. Sun's Reference Architectures combine:*

- *A documented multi-tiered architecture*
- *Recommended technology products from Sun and other vendors*
- *Architecture, sizing, and implementation guides*

The Open Group [29] created a SOA based RA for the Open Group. The SOA based RA consists of nine layers: *Operational Systems Layer, Service Component Layer, Services Layer, Business Process Layer, Consumer Layer, Integration Layer, Quality of Service Layer, Information Layer and Governance Layer*. Each layer assumes that the following three aspects need to be supported by the SOA based RA: requirements, logical and physical. The requirements include all functions that the layer enables and all of its capabilities. The logical includes all the architectural building blocks, design decisions, options, key performance indicators and so on. The physical aspect includes the realization of each logical aspect using technology, standards, and products.

BEA Systems [30] provides a document which includes "a worked design of an enterprise-wide service oriented architecture (SOA) implementation, with detailed architecture diagram patterns, opinions about standards, patterns on regulation compliance, standards templates, and potential code assets from members".

5.2. Decomposition of a reference architecture

As mentioned by Robert Cloutier [31] in the paper of "*The Concept of Reference Architectures*", a RA should address three aspects: technical architecture, business architecture and customer context. The relationship between technical architecture, business architecture and customer context is still depicted in Figure 11 in order to show the complete decomposition of a reference architecture. The technical architecture focuses on technique aspect. It captures design patterns in building a specific architecture of information systems. The business architecture focuses on business model and life cycle, which provide a guideline in the technical aspect. Similarly, the customer enterprises and users captured in the customer context also provide guideline in the technical aspect. To have a clear guideline from a RA to derive a concrete architecture, it is crucial to understand relations and interactions between these three aspects. However, business architecture and customer context are often missing (see Rosen [2007]), which leads to an incomplete solution for a reference architecture. In our research, business architecture and customer context are omitted as they have no effect on the result of the RA. Hence, only the technical architecture is remained and taken into consideration in our research.

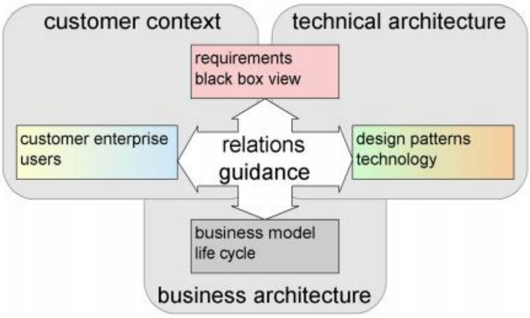


Fig. 11. Business architecture, technical architecture and customer context[31]

5.3. Abstraction level of a reference architecture

A RA is normally abstract. In order to transform a RA into an implemented application system, it is necessary to undergo several steps, which can be seen in Figure 12 [31]. As can been seen in the Figure, actuals systems can be derived from a reference architecture. Meanwhile, a reference architecture can also be mined from actual systems. In the first case, system architectures should be first derived from the reference architecture. These system architectures belong to a class of architectures. Then these system architectures are used for design and engineer for engineering documentation, which guides how the actual system can be assembled. In the second case, field feedback from actual systems is first used to rephrase engineering documentation. Then, constraints and opportunities are added to create system architectures. Finally, a reference architecture is mined from existing system architectures. In our research, we put our focus on designing a RA for IPs by mining from existing system architectures. In the future, the RA for IPs can be used to derive actual systems.

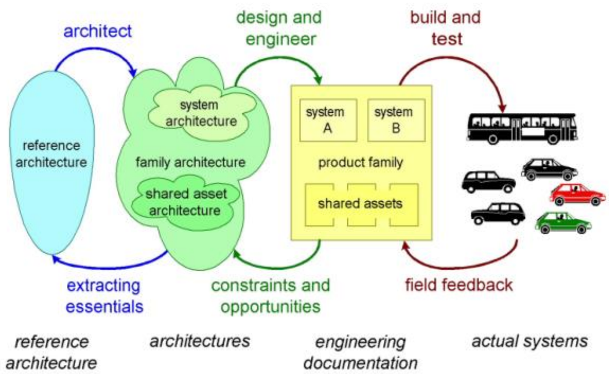


Fig. 12. Process to transform a reference archietcure into a real system[31]

6. DEVELOPMENT OF THE REFERENCE ARCHITECTURE

In this Chapter, the RA for IPs will be designed based on the method derived from [9], which has already been described in Chapter 2. The RA for IPs comprises three parts: *reference specification, reference model, guidelines and principles*, which will be identified and designed by extracting and analyzing elements from existing literature results.

6.1. Reference specification

The reference specification contains building blocks, design patterns and standards catalog. Building blocks are used as components to construct IPs. Design patterns are approaches or ways to design IPs. Standards catalogue provide design options for developers based on requirements. First of all, we will extract design patterns from the SLR results; then a further research on the SLR results will be gone through to collect building blocks for the RA; finally some standards catalog are listed by studying the SLR results.

(1) Design patterns

Based on the study from the SLR results, we can see there are various essential technologies and approaches which can be used for developing IPs. For example, *service oriented architecture (SOA), web services and cloud computing* are the most popular technologies used by developers. As many people may get confused about the relationship between them, here the association between them is given on Figure 13 [32]. Cloud computing is totally encapsulated by web services because cloud computing uses web services for connections. However, there are also other situations where web services are used other than cloud computing. For instance, web services can be implemented in SOA. Moreover, there are some situations where SOA is used other than web services.

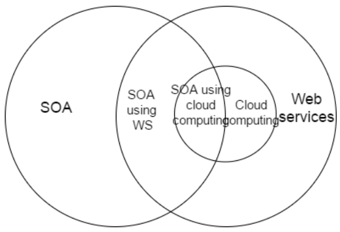


Fig. 13. The relationship between SOA, web services and cloud computing [32]

To help researchers have a flexible choice of design patterns that will be used for developing an IP, a general view of design patterns from all the existing 31 IPs is listed in Table III. The first column represents the paper number, which is corresponding to the SLR result in Appendix A. The second column represents authors of each paper. The third column shows the design pattern or technologies applied in the paper. If there are no specific design patterns or technologies that are mentioned in the paper, it is marked as “no specific”. As the technology of ESB is popular in the area of integration, a fourth column is given to show whether the IP applies ESB or not. If the ESB technology is used, it is marked with “ √ ”. If it is not used or not mentioned in the paper, it is marked with “X”. The fifth column shows whether the IP has already been implemented or not. If the IP has already been implemented, it is marked with “√”. If it is not implemented yet or is just a proposed architecture for IPs, it is marked with “X”.

TABLE III. DESIGN PATTERNS OF INTEGRATION PLATFORMS

Paper	Authors	Design patterns	ESB adoption	Implementation
1	Teng, Ju-Ting	Web services, ESB	√	X
2	Xu, Yi	Cloud computing (Web services)	X	X
3	Pinho, Eduardo	Cloud computing (Web services)	X	X
4	Hu, Jinxing	SOA, web services	X	X
5	Hu, zhiping	No specific	X	X
6	X.Li	No specific	X	X
7	Purgina, Marina	Web services	X	√
8	Dreher, Felix	SOA, Web services, ESB	√	√
9	Xiong, Ying	No specific	X	X
10	Dong, Long H.	Web services	X	√
11	Li, Qing	SOA, Web services, ESB	√	X
12	Wang, P.Y.	SOA, Cloud computing (Web services), ESB	√	√
13	S. Krishnamoorthy	Cloud computing (Web services)	X	√
14	Fu, Xiaolong	SOA, Web services, ESB	√	X
15	Z.Yang	Web services	X	X
16	Dreher, Felix	Web services	X	√
17	J. Brzezi	Web services	X	X
18	J. Woodbridge	No specific	X	X
19	G. Landolfi	Web services	X	X
20	Butakov, Sergey	Web services	X	√
21	Wei, N.	SOA, Web services, ESB	√	X
22	Deng, Wu	SOA, Web services	X	X
23	M. Fang	SOA	X	X
24	Q.Yun	No specific	X	X
25	Ganzinger, M.	SOA, Web services	X	X
26	Liu, Ying	SOA, Web services, ESB	√	√
27	Lu, S. P.	Web services	X	√
28	Li, Ming	SOA, Web services	X	X
29	G.Feipeng	No specific	X	√
30	Shen, jiquan	SOA, Web services	X	X
31	Llambias,Guzman	Web services, ESB	√	X

From Table III, we can see there are 24 IPs using web services, 12 IPs using SOA and 4 IPs using cloud computing for integration. Besides, there are 11 IPs combining SOA and web services for integration and 1 IP combining SOA, web services and cloud computing for integration. As cloud computing is encapsulated by web services, the combination of SOA, web services and cloud computing equals to the combination of SOA and cloud computing. In order to give a direct view of the popularity for each design pattern mentioned above, the usage ratio for each design pattern among the 31 IPs is shown in Figure 14. The design patterns of *web services*, *SOA*, *cloud computing*, *SOA+ web services*, *SOA+ web services+ cloud computing* separately occupy 77.4%, 38.7%, 12.9%, 35.3% and 3.2%.

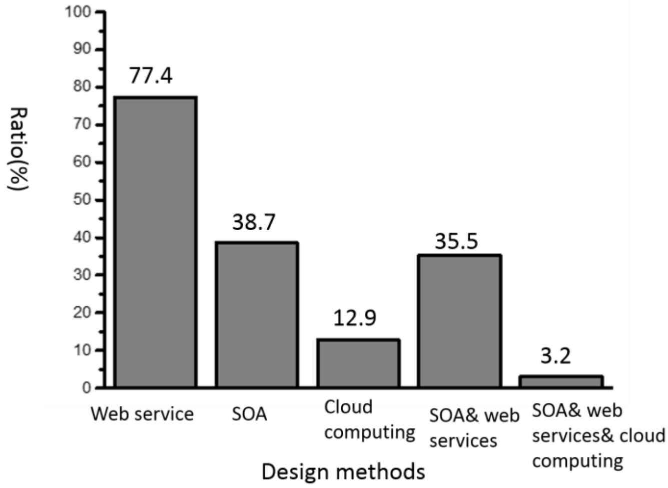


Fig. 14. Design method ratio among the SLR result

Besides, we can see there are 8 IPs using ESB and 23 IPs not using ESB. It has to be noted that IPs that have no declaration about the usage of ESB are treated as they do not use ESB. Also, we present the usage ratio of ESB in Figure 15. There are 25.8% IPs directly declaring that they use ESB as an enabling tool for integrating different objects. The rest 74.2% IPs have no declare that they adopt ESB as the enabling tool. As mentioned in [33], ESB acts as the middleware glue infrastructure that holds SOA parts together and integrates and manages the communication between different web services, applications and data sources. Currently, there are various ESB products in the market, which can be applied directly or redesigned according to enterprise business requirements. ESB not only integrates application systems and web services, but also extends to integrate other devices, which helps implement *Internet of Things*. Hence, ESB is a popular tool for integration platforms.

Moreover, we can see that only a few of IPs from the SLR results have already been implemented as real systems. Most of them are only proposed architectures of IPs.

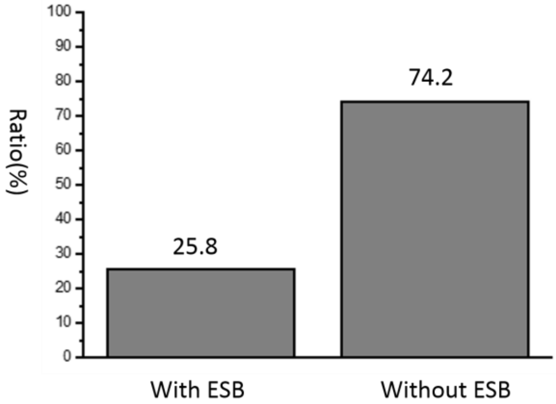


Fig. 15. ESB adoption ratio among the SLR result

(2) Building blocks

One of the most important part of the RA for IPs are building blocks, which are used to construct IPs. Building blocks of IPs can show the level of granularity of internal components. For example, both "*data filtering module*" and "*data processing module*" can become building blocks of IPs. However, "*data processing module*" comprises *"data filtering module", "data transformation module", "data grouping module", "data compression and decompression module"* and so on. As IPs in different papers have a different abstract level description of their architectures, it is necessary to map every building block of an IP into a standard building block concept. For example, Yi Xu and Jun Yan [34] presented an IP for smart cars which is deeply abstracted into physical entities, such as door locks, sensors, GPS and mobile phones. However, the supply chain IP proposed by Qiaoyun Yun [35] is only abstracted into application systems which are distributed to different locations or departments involved in the supply chain. To map every building block of an IP into a standard building block concept, each paper has been gone through.

To map each building block of IPs from the SLR result, architectures of these IPs are decomposed into several building blocks. Then, each building block is mapped to one or several standard building block concepts based on their functionalities. These building block concepts are extracted by experience from Quan Wang and Prince Singh. The final mapping result of each paper is shown in Appendix C. It presents building blocks of each IP, functionalities of each building block and standard building block concepts. The final standard building block concepts comprise "user interface", "service interface", "application API", "device API", "data exchange", "data processing", "service orchestration", "alias services", "QoS", "central database", "registry", "other files" and "external resources". In order to have a better understanding of each building block concept, a brief description of each term is given in Textbox 2. Besides, the abbreviation of each standard building block concept is given.

TABLE IV. STANDARD BUILDING BLOCK CONCEPTS IN EACH LAYER

Layer	Building block concepts
Presentation layer	UI, AIA, SII, DIA
Business logic layer	SO, DP, DE, QoS, AS
Data layer	CD, RE, OF

Textbox 2. Description of standard building block concepts

User interface (UI): *an end-user interface for interaction between users and IPs;*

Application integration API (AIA): *a connector for connecting outside application systems and IPs;*

Service integration interface (SII): *a connector for connecting outside web services and IPs;*

Device integration API (DIA): *a connector for connecting outside devices and IPs;*

Service orchestration (SO): *a module for organizing business processes by service combination;*

Data processing (DP): *a module for processing incoming or outgoing data, such as data grouping, data compression and decompression, data transformation and data filtering;*

Data exchange (DE): *a module for receiving and sending messages, as well as message routing and invocation;*

Quality of Service (QoS): *a module for ensuring the quality of services, such as stability, reliability and so on;*

Alias services (AS): *internal services which add extra functionality of integration platforms, such as database operation;*

Central database (CD): *the central database which stores core data of integration platforms;*

Registry (RE): *a module which registries outside web services or applications for facilitating locating;*

Other files (OF): *a module which stores other information, such as metadata of the integration platforms or outside applications and web services;*

To facilitate in designing conceptual model of the RA, these standard building block concepts are classified into three layers. These three layer are composed of *"interface layer"*, "*core logic layer"* and *"storage layer"*. Standard building block concepts are classified and given based on layers in Table IV. If the specific IP architecture has a building block which has one of the functionalities of a specific standard building block concept, the building block concept in that paper will be marked with "√". Otherwise, it is marked with "X".

The first layer is the interface layer, which is responsible for the connection between IPs and external resources. In this layer, four standard building block concepts were identified, which include "user interface", "service integration interface", "application integration API" and "device integration API". The mapping result of the interface layer is shown in Table IV.

The second layer is core business logic layer, which is responsible for dealing with business logic. In this layer, five standard building block concepts were identified, which include "service orchestration", "data exchange", "data processing", "QoS" and "alias services". The mapping result of the business logic layer is shown in Table V.

The third layer is data layer, which is responsible for collecting data from various external resources and providing a common data center for sharing. In this layer, three standard building block concepts were identified, which include "central database", "registry" and "other files". The mapping result of the data layer is shown in Table VI.

TABLE V. BUILDING BLOCKS MAPPING IN PRESENTATION LAYER

No	Application integration API	Service integration interface	User interface	Device integration API
1	X	√	X	X
2	X	√	X	X
3	X	√	√	X
4	X	√	√	X
5	X	X	X	X
6	√	X	√	X
7	X	√	√	X
8	X	√	X	√
9	√	X	X	X
10	√	√	√	X
11	X	√	X	X
12	X	X	X	X
13	√	√	√	X
14	√	√	√	X
15	√	√	X	X
16	X	X	√	X
17	√	X	√	X
18	√	X	X	√
19	X	√	√	X
20	X	X	X	X
21	X	X	√	X
22	√	X	X	√
23	X	X	X	X
24	√	X	X	X
25	X	X	√	X
26	√	X	√	X
27	X	X	X	X
28	√	X	√	X
29	√	X	X	X
30	X	X	X	X
31	X	X	√	X

TABLE VI. BUILDING BLOCKS MAPPING IN BUSINESS LOGIC LAYER

No	Service orchestration(BPM)	Data processing	Data exchange	QoS	Alias services
1	√	√	√	√	√
2	X	√	√	X	X
3	√	√	X	X	√
4	X	√	√	√	√
5	√	√	X	X	√
6	√	X	√	√	√
7	X	√	√	X	X
8	√	X	X	X	√
9	√	√	√	√	√
10	X	X	X	√	√
11	√	X	X	√	√
12	X	X	X	√	√
13	X	√	X	X	√
14	X	√	√	√	√
15	X	√	√	√	√
16	X	√	X	√	√
17	X	√	√	√	X
18	X	√	√	√	X
19	X	X	X	X	√
20	X	√	√	√	√
21	√	√	√	√	√
22	X	√	X	X	√
23	X	√	√	√	√
24	√	X	√	X	X
25	X	√	X	√	√
26	√	√	√	√	√
27	√	X	X	√	√
28	√	√	√	√	√
29	√	√	√	√	√
30	√	√	√	X	X
31	X	√	√	√	X

TABLE VII. BUILDING BLOCKS MAPPING IN DATA LAYER

No	Central database	Registry	Other files
1	X	X	X
2	√	X	X
3	√	X	√
4	√	X	√
5	X	X	X
6	√	X	X
7	X	X	X
8	X	√	X
9	√	X	X
10	√	X	X
11	X	√	X
12	X	X	X
13	√	X	X
14	√	√	X
15	X	X	X
16	√	X	X
17	X	√	X
18	√	X	X
19	√	X	X
20	X	X	X
21	√	X	X
22	X	√	X
23	√	X	X
24	√	X	X
25	X	X	X
26	X	√	X
27	√	X	X
28	X	X	X
29	X	X	X
30	√	√	X
31	√	X	X

By analyzing building block concepts in the interface layer of IPs, we can classify IPs into three categories: application integration, web service integration and device integration. Application integration needs application integration APIs to integrate with IPs. Service integration needs service integration interfaces to integrate with IPs. Device integration needs device integration API to integrate with IPs. Here the web service interface also encapsulates cloud service interface. In order to have a clear view of each integration category among the SLR result, the ratio of each integration category appearing in the SLR is depicted in Figure 16. From the figure, we can see there are respectively 41.9%, 38.7% and 9.7% IPs involving application integration, service integration and device integration. Among them, web service integration is the most popular integration solution as most enterprise application integration is implemented by using web services. Before the introduction of web services, application integration is realized by some EAI products. Tibco, Webmethods and See Beyond were the leaders in this space [36]. Besides, 12.9% IPs have both application integration and service integration; 6.4% IPs have both application integration and device integration; 3.2% IPs have both service integration and device integration. It has to be noted there are no IP involving all these three integration categories. To realize the presentation layer, adapters are often used to connect with external applications, services and devices. Besides, a user interface can be programmed to interact with external users.

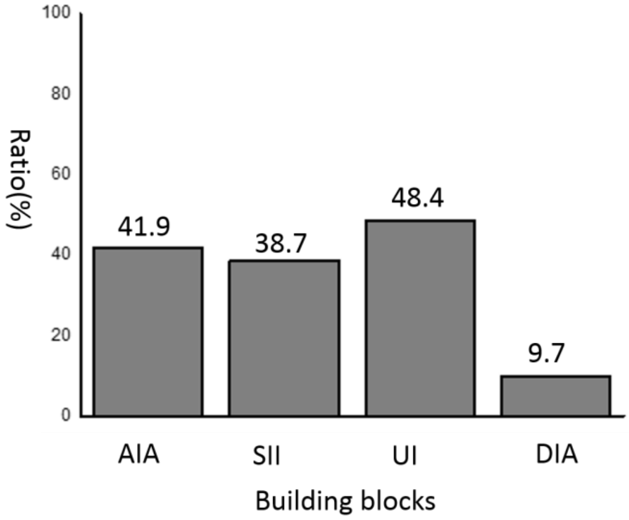


Fig. 16. Building block ratio in the presentation layer

Building block concepts in the business logic layer have most abundant functions of IPs. Normally, building block concepts in this layer can be encapsulated by a customized ESB. For example, service deployment, administration and monitoring can be embed inside an ESB. Besides, it can also have some business related functions, such as business process management, business process monitoring and data management. In order to have a clear view of building block concepts in the layer of business logic among the SLR result, the ratio of each building block concept in this layer is depicted in Figure 17. From the figure, we can see there are 45.2%, 74.2%, 61.3%, 67.7% and 77.4% IPs involving service orchestration (business process management), data processing, data exchange, QoS and alias services. Service orchestration or business process management is a module which combines several services or applications to realize a common business goal. Data processing module includes data filtering, data format transformation, data compression and decompression, and data grouping. Data exchange includes data receiving and sending, data exchange, message routing and invocation. QoS module includes problems of reliability, stability and some other properties of the integration platform itself. Besides, there should be some other building block concepts for IPs. For example, a database operation module realizes the connection of the business logic layer and databases. It can be implemented by JDBC, ODBC or some other technologies.

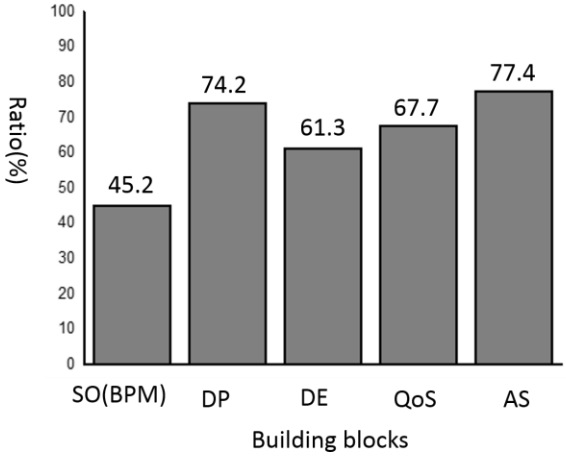


Fig. 17. Building block ratio in the business logic layer

Data storage also performs as an essential part of integration platforms, especially for data IPs. In this layer, three storage building block concepts are identified: central database, registry and other files. In order to have a clear view of building block concepts in the layer of data storage among the SLR result, the ratio of each building block concept in this layer is depicted in Figure 18. From the figure, we can see there are 54.8%, 22.6% and 6.5% IPs involving central database, registry and other files. Central database stores the core information of the integration platform, as well as information from external integrated partners in some cases. Registry module is responsible for registering outside entities, such as web services, applications or devices. It eases the process of message routing. Other files module may comprise metadata of the integration platform, some local files and so on. Besides, some distributed databases may be integrated with IPs. They may reside on external application systems. These databases can be synchronized to the local database to realize data integration in some cases. Remote database access middleware can be applied as a tool to interact with distributed databases.

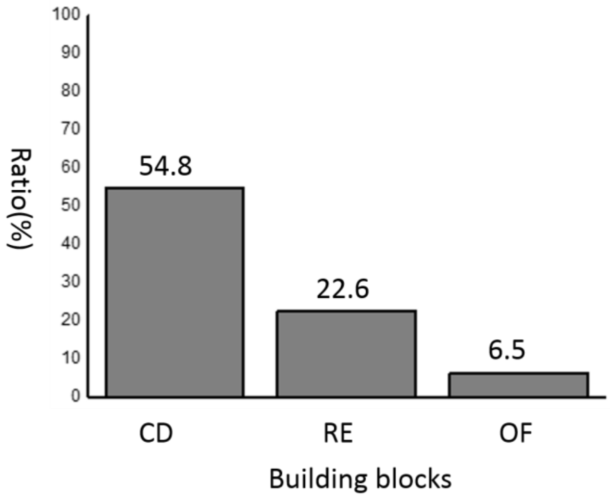


Fig. 18. Building block ratio in the data layer

(3) Building block concept usage ratio in different integration types

To see how relevant each standard building block concept is mapped to application integration, service integration and device integration respectively, the ratio of each building block concept appear in each case is separately represented in Figure 19 , Figure 20 and Figure 21.

For the application integration case, we can see in total that each standard building block concept appears in application integration platforms more or less. The top four standard building block concepts that exist in application integration platforms for most times are "data processing", "data exchange", "QoS" and "alias services", which all appear 76.9% times. "User interface" and "central database" both appear 53.8% times. "Service orchestration" and "registry" both appear for 30.8% times. "Other files" has no appearance in application integration category.

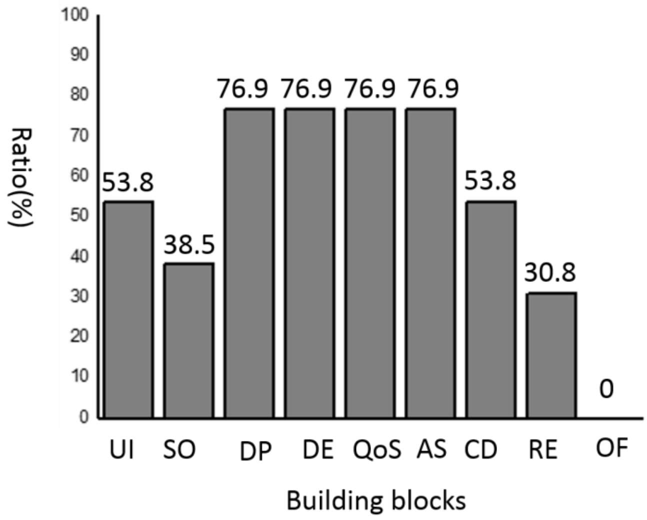


Fig. 19. Building block ratio in application integration category

For the service integration case, we can also see in total that each standard building block concept appears in service integration platforms more or less. The top five standard building block concepts that exist in service integration platforms for most times are “alias services”, “user interface”, “data processing”, “central database” and “QoS”, which separately appear 75%%, 66.7%, 66.7%, 58.3% and 50% times. They all appear in service integration at least half times in total. “Data processing”, “service orchestration” and “registry” separately appear for 41.7%, 33.3% and 25% times. “Other files” appear for least times, which is 16.7%.

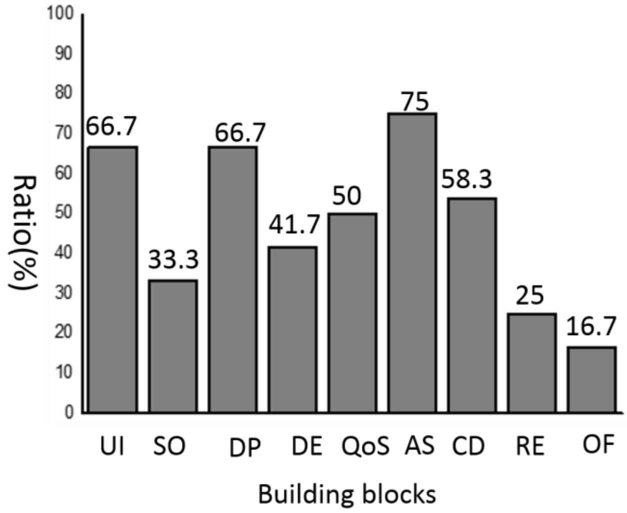


Fig. 20. Building block ratio in service integration category

For the device integration case, we can see in total that not all standard building block concepts appear in device integration platforms. “User interface” and “other files” appear 0 times in device integration. The standard building block concept that exists in device integration platforms for most times is “registry”. It appear 100% times. “Data processing” and “alias services” both appear for 66.7% times. “Service orchestration”, “data exchange”, “QoS” and “central database” all appear for 33.3%% times.

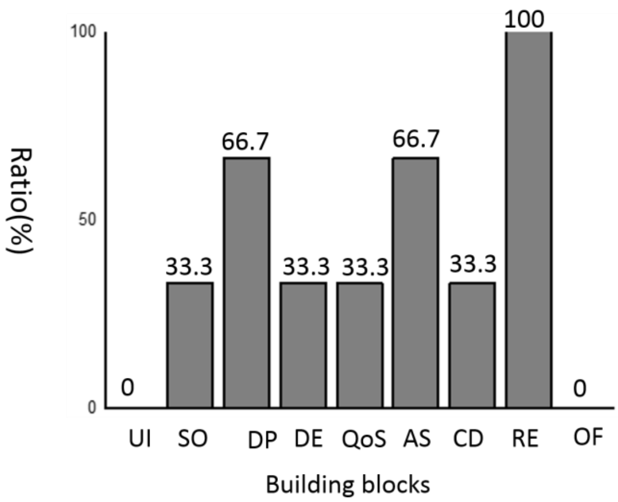


Fig. 21. Building block ratio in device integration category

(4) Building block concept usage ratio in different domains

As integration platforms existing in the SLR result are involved in various domains, it is rational to analyze how relevant each standard building block concept is mapped in different domains. In our research, there are 7 domains which comprise more than one integration platforms. These domains include “enterprise systems”, “supply chain”, “government”, “medication”, “campus”, “biology” and “transportation”. The ratio of each standard building block concept which appears in a specific domain will be separately given.

The case of integration platforms in the domain of “enterprise systems” is given in Figure 22. As can be seen from the figure, all standard building block concepts ever appear more or less times except “device integration interface”. The top standard concepts which appear in this domain for most times are “service orchestration” and “alias services”, which both appear for 100% times. “User interface”, “QoS”, “central database” and “data processing” all appear in this domain for 60% times. ”Service integration interface” and “data exchange” both appear for 40% times. “Application integration AIP”, “registry” and “other files” all appear in this domain for 20% times.

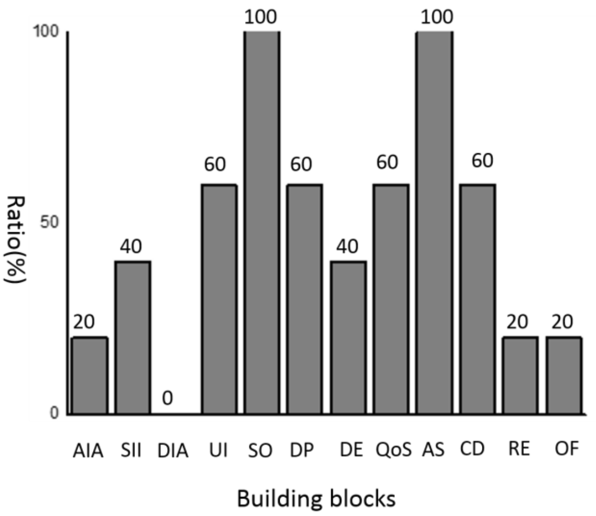


Fig. 22. The ratio of standard building block concepts appearance in enterprise systems

The case of integration platforms in the domain of “government” is given in Figure 23. As can be seen from the figure, all standard building block concepts ever appear more or less times except “device integration interface”, “registry” and “other files” as well. The top standard concepts which appear in this domain for most times are “application integration API” and “data exchange”, which appear for 100% times. “Service orchestration”, “data processing”, “QoS” and “alias services” all appear for 75% times. The rest 3 concepts: “service integration interface”, “user interface” and “central database” all appear in this domain for 25% times.

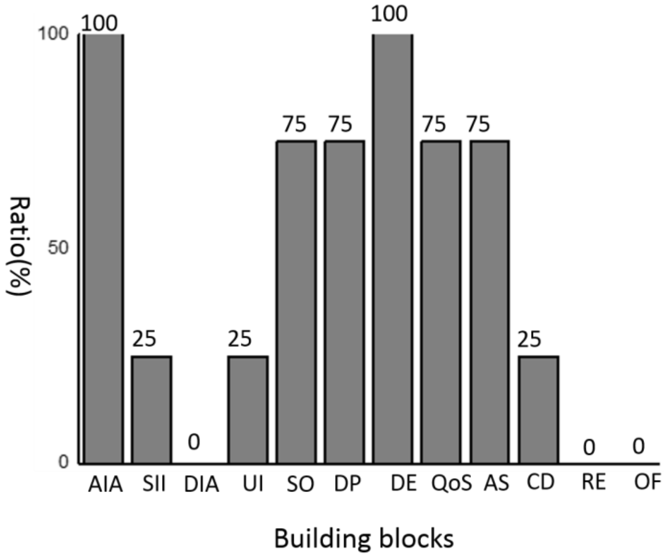


Fig. 23. The ratio of standard building block concepts appearance in government

The case of integration platforms in the domain of "supply chain" is given in Figure 24. As can be seen from the figure, not all standard building block concepts ever appear more or less times. "Application integration API", "service integration interface", "device integration API", "service orchestration", "data exchange", "registry" and "other files" never exist in supply chain. The top standard concepts which appear in this domain for most times are "QoS" and "alias services", which appear for 100% times. "User interface", "data processing" and "central database" all appear in this domain for 50% times.

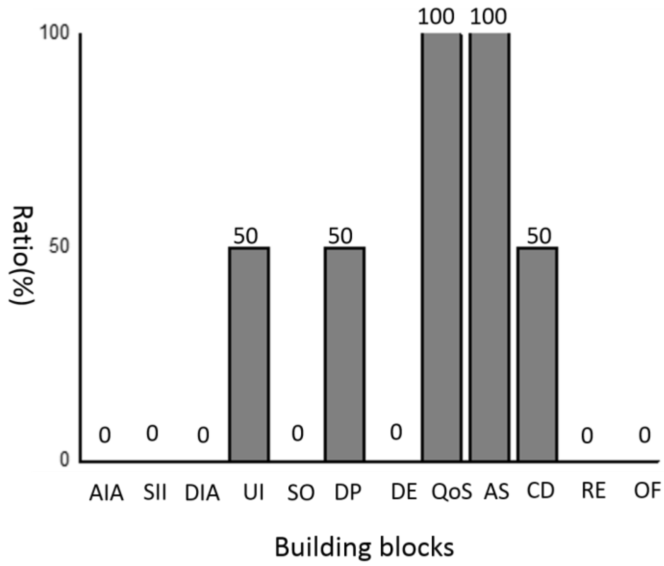


Fig. 24. The ratio of standard building block concepts appearance in supply chain

The case of integration platforms in the domain of "medication" is given in Figure 25. As can be seen from the figure, all standard building block concepts ever appear more or less times except "service orchestration" and "other files". The top standard concepts which appear in this domain for most times are "application integration API", "user interface", "data processing", "data exchange", "QoS" and "central database", which reach for 66.7% times. "Device integration API", "service integration interface", "alias services" and "registry" only appear in this domain for 33.3% times.

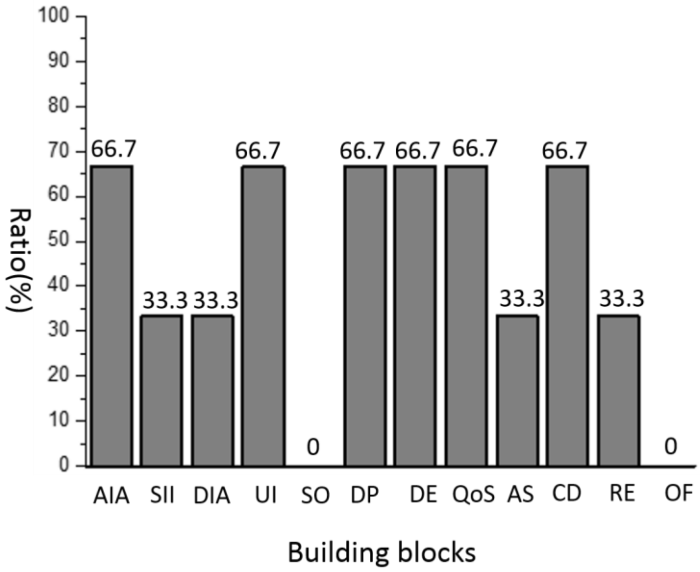


Fig. 25. The ratio of standard building block concepts appearance in medication

The case of integration platforms in the domain of "campus" is given in Figure 26. As can be seen from the figure, all standard building block concepts ever appear more or less times except "device integration interface" and "other files". The top standard concepts which appear in this domain for most times are "data exchange", "data processing", "central database" and "registry", which reach for 100% times. The rest building block concepts all appear in this domain for 50% times.

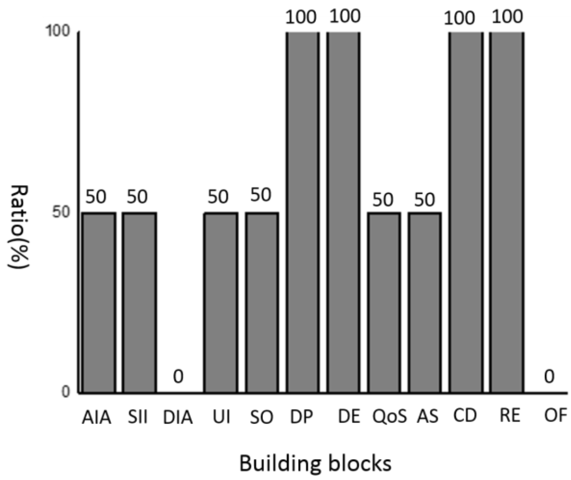


Fig. 26. The ratio of standard building block concepts appearance in campus

The case of integration platforms in the domain of “biology” is given in Figure 27. As can be seen from the figure, all standard building block concepts ever appear more or less times except “device integration interface”, “service orchestration”, ”registry” and “other files”. The top standard concepts which appear in this domain for most times are “user interface” and “QoS”, which reach for 100% times. ”Alias services”, “data processing”, and “central database” all appear in this domain for 75% times. “Application integration API” and “service integration interface” only appears in this domain for 25% times.

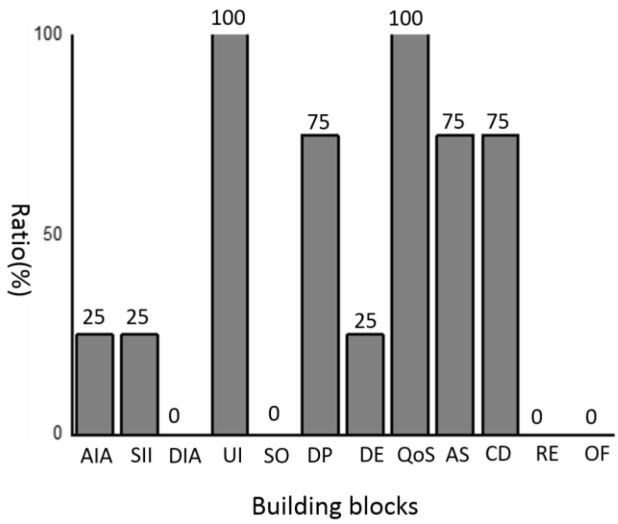


Fig. 27. The ratio of building block concepts appearance in biology

The case of integration platforms in the domain of “transportation” is given in Figure 28. As can be seen from the figure, all standard building block concepts ever appear more or less times except “device integration interface” and “registry”. The top standard concepts which appear in this domain for most times are “data processing”, “data exchange”, “QoS”, “alias services” and “central database”, which reach for 100% times. The rest all appear in this domain for 50% times.

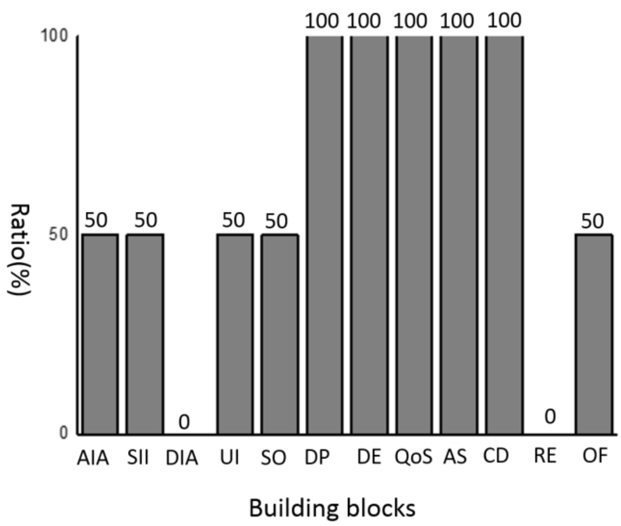


Fig. 28. The ratio of standard building block concepts appearance in transportation

(5) Standards catalogue

This section describes the standards catalogue of the RA for IPs, which is given in Textbox 3. Each standard provides options for corresponding principles in section 6.3. Standard 1 provides integration tools which can be applied by IPs. Developers can choose to use adapters or ESB as IPs. Standard 2 provides integration methods which can be applied by IPs. These integration methods are for different integration types. Standard 3 provides building blocks which can be selected by IPs based on requirements. The rules of using these standards will be given in principles and guidelines of the RA.

Textbox 3. Standards catalog for the RA

***Standard 1**: adapters, ESB*

***Standard 2**: traditional EAI methods, SOA, traditional device integration method*

***Standard 3**: application integration API, service integration interface, device integration API, user interface, data processing, data exchange, service orchestration, QoS, alias services, central database, registry, other files*

6.2. Reference model

The reference model inside the RA includes two parts: glossary and the conceptual model. The glossary describes terms that come across the whole system, which is shown in Textbox 4.

Textbox 4. Glossary for the RA

***Architecture**- The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [IEEE 1471, 2000].*

***Architecture framework**- An architecture framework provides guidelines and rules for structuring, classifying, and organizing architectures [DoDAF, 2007].*

***Design pattern**- A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in a system. [Gamma et al., 1995].*

***System**- A system is a construct or collection of different elements together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, politics, and documents, that is, all things required to produce system-level results [Maier and Rechtin, 2000].*

***User interface**- A user interface, also called a "UI" or simple an "interface", is the means in which a person controls a software application or hardware device. A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way [Techterms].*

***Viewpoint**- A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis [IEEE 1471, 2000].*

Based on the building blocks extracted from existing papers and corresponding concepts derived from them, an initial conceptual model is designed on Figure 29 [37].

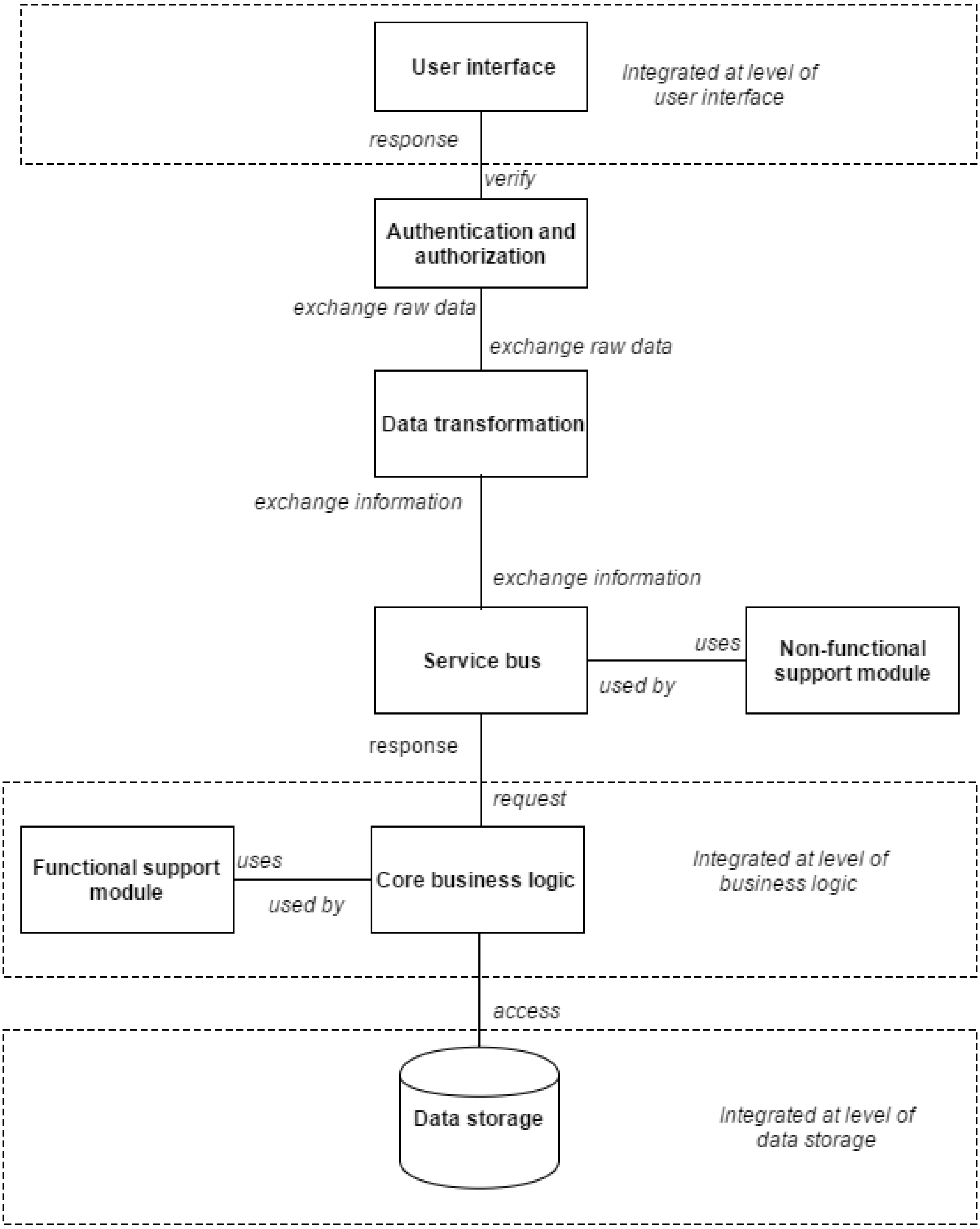


Fig. 29. Initial version of conceptual model for integration platforms [37]

The initial conceptual model of the RA comprises 8 modules. Firstly, the user interface is the connection point for interacting with external entities, such as people, applications, services or devices. It receives and replies messages to outside world. Secondly, an authentication and authorization module is introduced to provide security for the IP. Users or other external resources are assigned with a limited right to access the local resource. Then a data transformation module is applied to standardize the data format so that data from distributed resources with different format can integrated. The fourth module is a service bus, which is responsible for data exchange and message routing. Besides, the service bus also have a non-functional supporting module to enhance the performance of the IP. It may include reliability, maintainability, stability and others. The sixth module is a core business logic is responsible for deal with the business processes involved in the IP, which may be supported by a functional-supporting module. Finally, a database is adopted to store data for the whole IP and data sources coming from outside world.

Although the initial version of the reference model is able to describe the relationship between internal entities inside a general IP, it has some mistakes and limitations. First of all, a user interface module is not enough to interact with outside world. When it comes to application integration, service integration and other device integration, we need relevant adapters to communicate with external resources. Besides, a user interface is still useful to allow people to access the IP. Secondly, the authentication and authorization module is a virtual module which can be implemented in the interface layer. Thirdly, the data transformation module and non-functional supporting module can be embed inside an ESB. Fourthly, the core business logic of an integration platform is to orchestrate services or manage business processes, which can also be embed inside an ESB. The functional supporting module can be replaced by external applications or web services. Finally, the database can also be classified into the local database, registry and some other storage.

After improving the initial version of the conceptual model, a new version has been derived, which is represented in Figure 30.

The new version of the conceptual model can abstract the whole system into two parts: the IP and the external resources which are integrated by the IP. The data flow goes between the IP and the external world. External users can interact with the IP through a user interface. Applications, services and devices can access the IP through certain interfaces, which can be implemented by some adapters. Moreover, some authentication and authorization can be implemented in the interface layer.

Then it comes to the core part of the IP. It has a data processing module, a data transfer module, a service orchestration module, a QoS module and an alias service module. These modules can be chosen to be implemented in the integration platform according to real business requirements. Moreover, a data storage module is needed. It may comprise a central database, a registry and other storage.

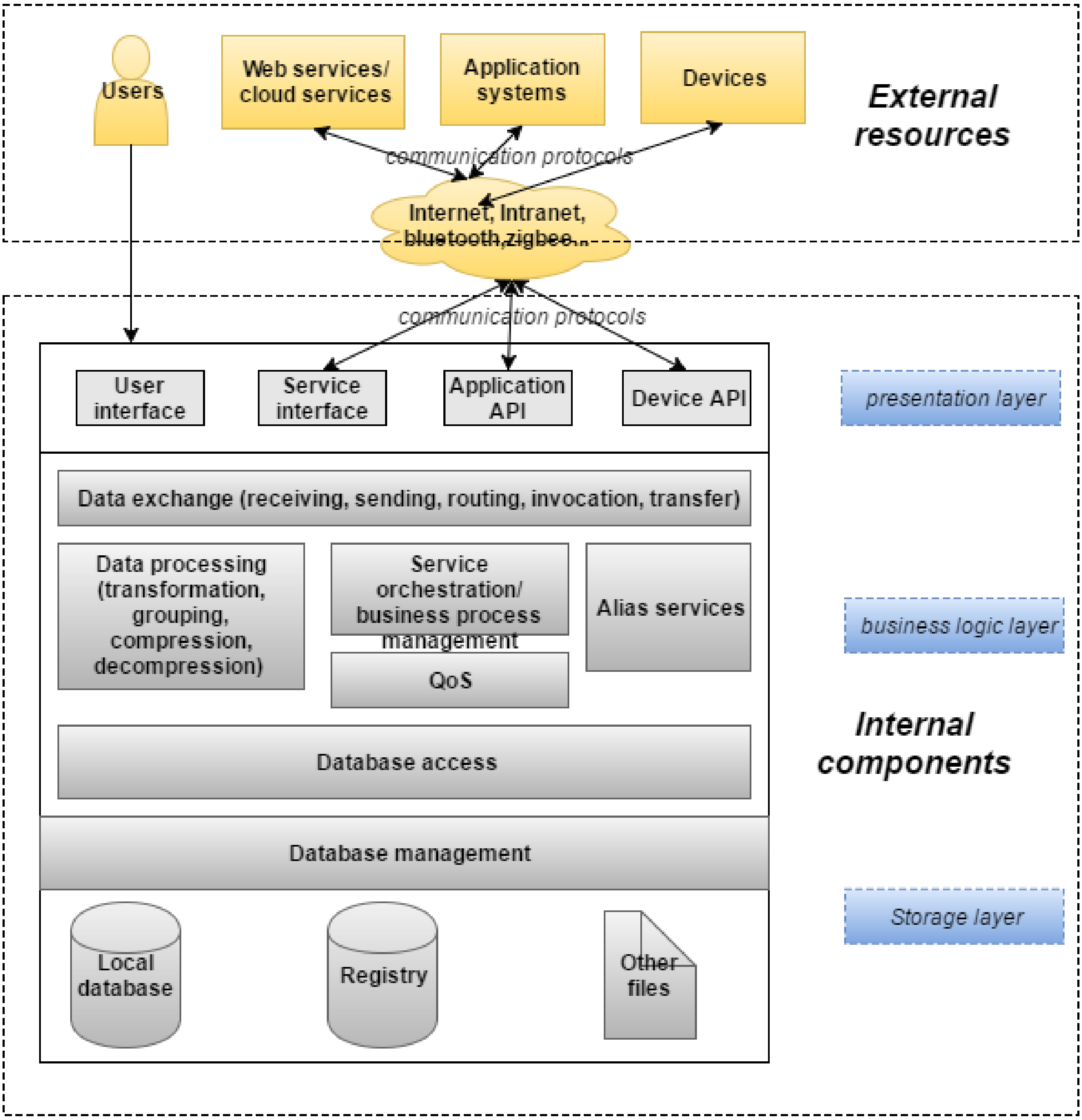


Fig. 30. Final version of conceptual model for integration platforms

6.3. Guidelines and principles

This component of the RA for IPs, on one hand provides a decision tree of pattern usage for architects, on the other hand provides the principles or properties of the IP. The decision tree is represented in Figure 31.

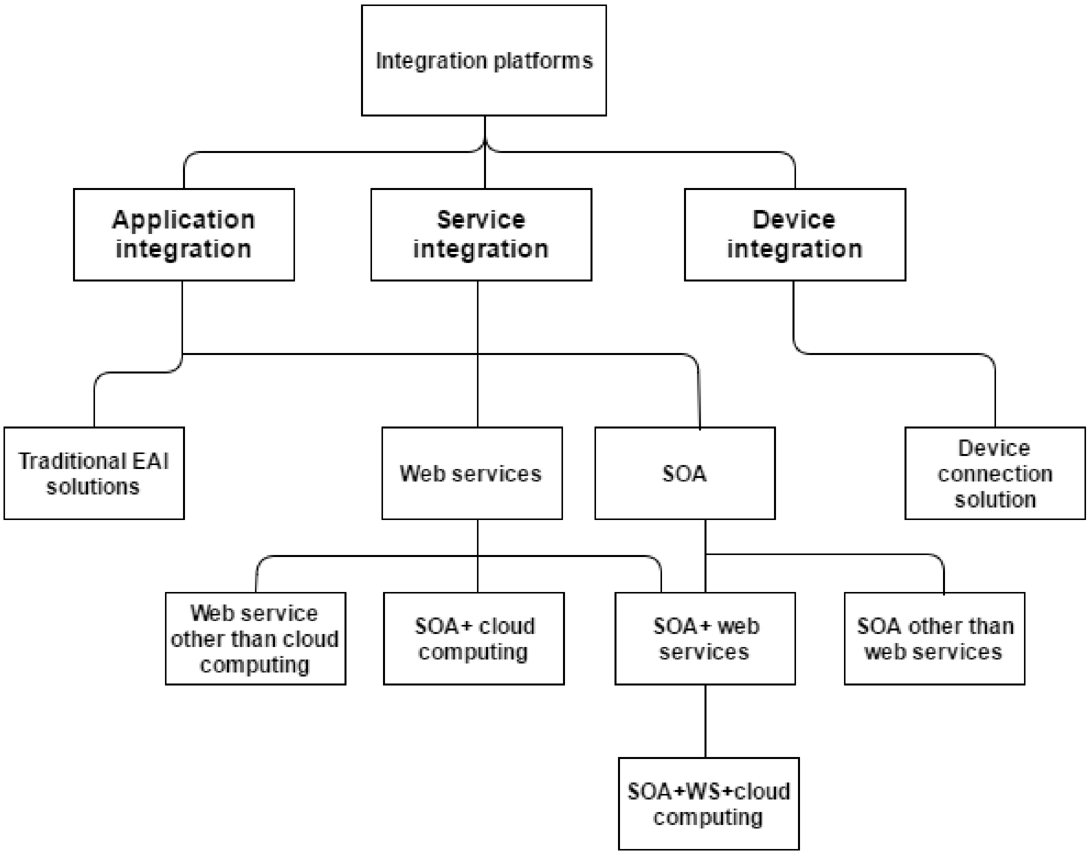


Fig. 31. A decision tree of design patterns for integration platforms

As can been seen from the decision tree, there are various solutions which can realize integration platform. Firstly of all, it is necessary to understand which entities are to be integrated. For application integration, it can use either a traditional application system integration method or it can treat applications as services to use service integration method. For service integration, there are five solutions: SOA other than web services, SOA with web services, SOA with web services and cloud computing, Web services other than cloud computing or SOA, and web services with cloud computing other than SOA. These five design patterns will not be illustrated here. For device integration, it is often to use adapters to integrate them.

Besides, we can also use a brief adapter or middleware to connect objects when there are only two or three objects are needed to be integrated. When it comes to more objects, an ESB will be currently the best choice.

On conclusion, the following three principles are listed in Textbox 5 for the reference architecture, which can help researchers better choose an appropriate method for designing their IPs. The principles of designing an IP can be referred to the standard catalogue in section 6.1. Each standard catalogue can be linked the one of the principles.

Textbox 5. Principles for the RA

***Principle 1:* Choose the number of objects to be integrated.**

If the number of to be integrated objects is small, we can use adapters. If the number of to be integrated objects is large and the extensibility property is required to add to them, then we can use an ESB.

Principle 2: Choose the object categories to be integrated.

If only applications are to be integrated, we will use traditional EAI solutions or SOA. If only devices are to be integrated, we will use traditional device integration solutions or SOA. If both applications and devices are to be integrated, we will use the SOA design pattern.

***Principle 3*: Choose appropriate building blocks for developing IPs.**

Application integration API: used when traditional application integration methods are applied;

Service integration interface: used when SOA design pattern is applied.

Device integration API: used when traditional device integration methods are applied;

User interface: used when people need to use the IP;

Data processing: used when external resources are in different formats or IPs requires a specific data format;

Data exchange: used for almost all IPs;

Service orchestration: used for business process management;

QoS: used when some qualities of IPs are required, such as security, load balancing and so on;

Alias services: used when extra functions of IPs are required, such as database operation, service encapsulation and so on;

Central database: used fro almost all IPs;

Registry: used when there are too many external resources to be applied;

Other files: used when specific storing information is required;

6.4. The reference architecture for integration platforms

After design and collect the reference specification, the reference model and principles and guidelines, a reference architecture for integration platforms is designed and shown in Figure 32.

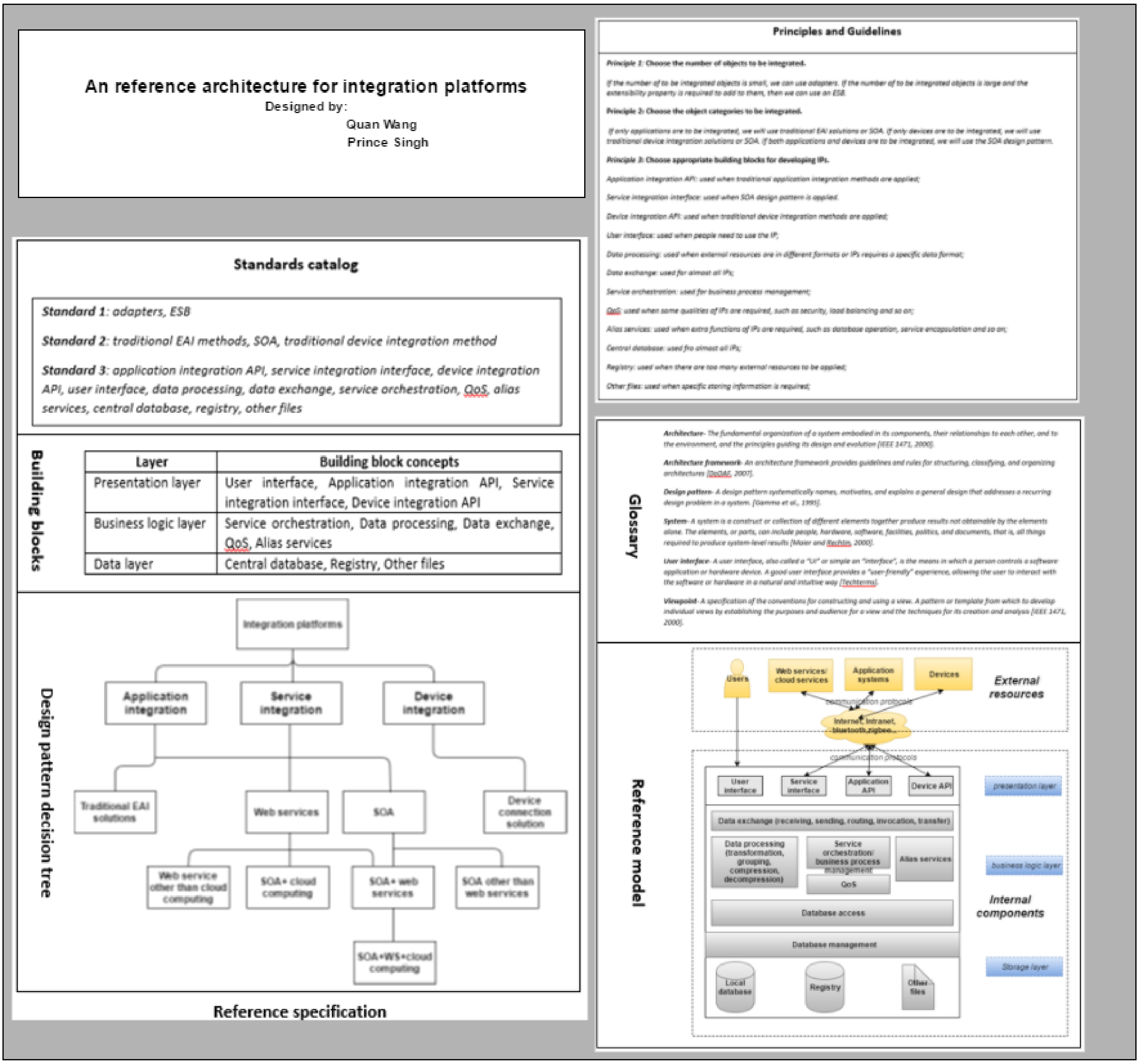


Fig. 32. The reference architecture for integration platforms

7. VALIDATION BASED ON JANSSEN LOGISTICS

In order to validate the RA for IPs designed in Chapter 6, a study case from Janssen Logistics b.v is used. The case study can be viewed on Appendix D. Janssen Logistics b.v is a logistic service provider (LSP), primary providing road and rail transport solutions based in Rotterdam. In this Chapter, we will first give an overview on Janssen Logistics b.v from the perspective of business activities. Secondly, we will give an overview of the enterprise architecture. Thirdly, we will present some problems that Janssen Logistics b.v is facing based on the enterprise architecture. Fourthly, we will design an IP for Janssen Logistics b.v by using the RA designed in Chapter 6. Fifthly, we will give a discussion about the validation result.

7.1. Business activities in Janssen Logistics b.v

The main activity of Janssen Logistics b.v is to transfer products for its clients from deep sea ports in the Netherlands to European countries. These products are from Asia, Asia-Pacific and South America. The transportation activities performed by Janssen Logistics b.v are executed by means of barges, trucks, rail ways and air planes. As Janssen Logistics b.v only have 55 trucks recently, it often outsources transportation activities to other companies.

As stated by Dennis Janssen, he planned to transform Janssen Logistics b.v into a 4PL enterprise, which makes full use of real time information to perform business activities. Based on the study case [Appendix D], a 4PL primarily plays the role of a broker in logistics. It survives in the market based on its ability to collect information on market supply and demand, market players and real time operational conditions. Access to the right information and smart planning are the main competitive advantages of a 4PL. Besides, he plans to add a bidding portal to the website of Janssen Logistics b.v so that their clients can bid for transportation activities from the website. Moreover, a warehouse is proposed to be built to make full use of trucks.

7.2. Enterprise architecture in Janssen Logistics b.v

Like many other logistics service providers, Janssen Logistics b.v also aims to restructure and re-invent to become a 4PL company. To realize this goal, it is necessary to analyze the enterprise architecture in Janssen Logistics b.v. An overview baseline architecture of Janssen Logistics b.v from the perspective of actors, business processes, applications and data objects is shown in Figure 33. The enterprise architecture was made by ArchiMate 2.1.

The enterprise architecture for Janssen Logistics b.v can be divided into four layers: actor layer, business process layer and application layer. Each business process is used by one or several actors. For example, the business process of “create waybills” is done by clients of Janssen Logistics b.v. Each application triggers one or several business processes. For example, the application of “Janssen Logistics b.v website” triggers business process of “create order files”. Each data object is used by one or several applications. For example, the data object of “product location” is used by the application of “Janssen Logistics b.v website”.

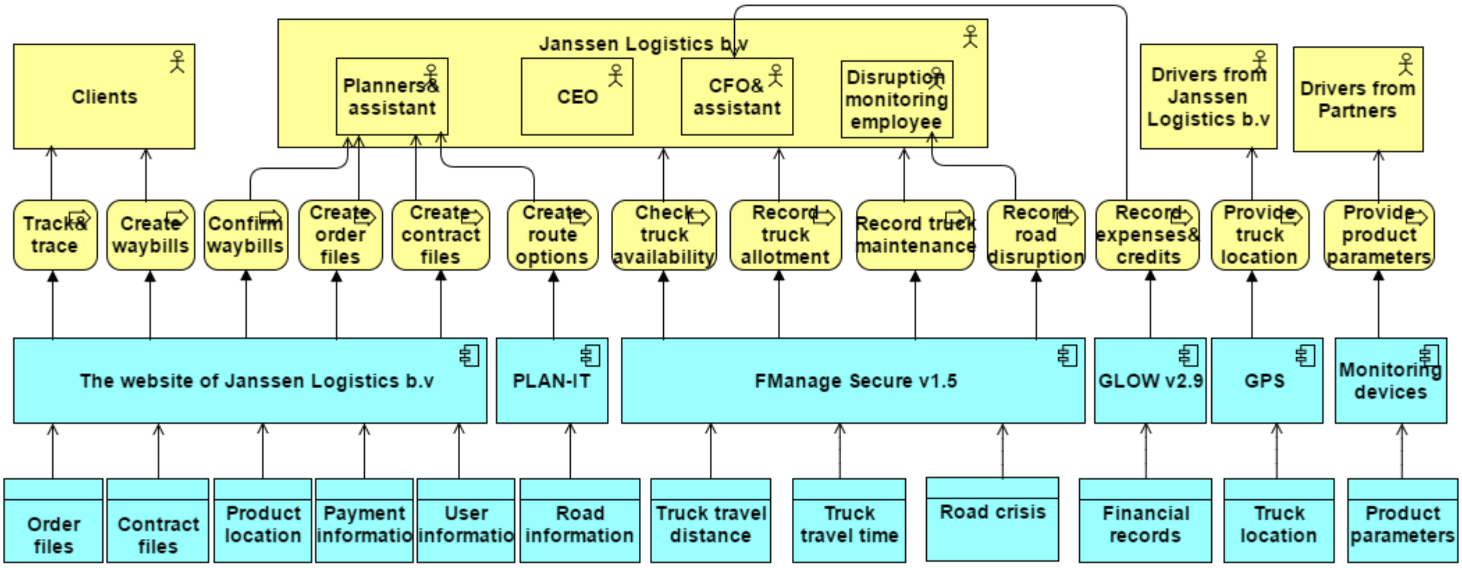


Fig. 33. Enterprise architecture for Janssen Logistics b.v

In order to have a better understanding of applications, a list of functionalities referred to applications is given in Table VII. From the Table VII, we can see that the baseline application in Janssen logistics consists of an order related website, PLAN-IT software, FManage Secure v1.5 software, truck devices and GloW v2.9 software.

The website of Janssen Logistics b.v consists of an order entry front end and an ORDER-IN back end application. Clients can create orders via the order entry. Planners in Janssen Logistics b.v can confirm or reject the order through ORDER-IN. If the order is accepted, then a contract file is generated and sent to clients based on additional information. Besides, an order file is also generated and exported to .xl for frequent viewing and editing. After the completion of the order, the order file will be uploaded in the database. Moreover, a track and trace functionality is provided by the website. The order related status can be viewed by querying the database.

PLAN-IT is a routing planning tool. It is mainly used in the case of one-off contracts for single delivery. It provides multiple route options, which can be marked as available or unavailable for certain time frames. The plan produced by PLAN-IT will be anyway validated by planners.

FManage Secure v1.5 is a fleet management software, which is hosted in a server. It records the availability of trucks, their maintenance schedule and allotment to different orders. After completion of an order, some information like km travelled and time taken are entered for scheduling maintenance. Besides, some feed disruption information is uploaded on FManage Secure v1.5 in the form of excel sheet by an employee.

Truck devices also play an important role in logistics. Trucks in Janssen Logistics b.v all have inbuilt GPS to show route maps. Besides, each truck is assigned with a device to show the pick up or drop off time, as well as the pick up or drop off location. Any other messages sent from the office can be shown in devices. Moreover, trucks from partners of Janssen Logistics b.v are all equipped with monitoring sensor devices. These sensors can scan some parameters of products, such as temperature and humidity.

GLOW v2.9 is a finance software which keeps track of all expenses and credits. The software is used for bookkeeping and other paid activities. Besides, it keeps track of all employee, orders expenditures, salaries, credits and profits. It is only accessible for CFO. When CFO is not in the office, her assistant can also access it.

TABLE VIII. BASELINE APPLICATIONS IN JANSSEN LOGISTICS B.V

Baseline applications	Functionalities
Website (ORDER-IN and ORDER ENTRY)	1. creating a waybill by customers; 2. sending an email alert to planners; 3. confirming an order by planners; 4. receiving additional information from customers; 4. generating a contract file and sending it to customers; 5. generating order files (XML) and uploading them to database; 6. providing track and trace for clients; 7. providing a database recording order status; 8. providing a database recording order data and financial payments;
PLAN-IT	1. creating suitable routes for certain time frames;
FManage Secure v1.5	1. recording availability of trucks; 2. keeping truck maintenance schedule; 3. keeping order allotment of trucks; 4. uploading the excel sheet of feed disruption information;
Truck devices	1. providing a GPS showing the route map; 2. providing a device showing pick up/ drop off location; 2. providing a device showing pick up/ drop off time;
GLOW v2.9	1. keeping all track of all expenses and credit;

7.3. Problems in Janssen Logistics b.v

Based on the enterprise architecture in Janssen Logistics b.v, an essential problem can be identified: Applications in Janssen Logistics b.v are distributed and work alone. They lack of communication and collaboration between each other. Besides, Janssen Logistics b.v aims to add a bidding portal for their partners. Moreover, Janssen Logistics b.v plans to building up warehouses so that their trucks will not return back in empty. To record products which will be stored in warehouses, a software used to record product storage information should be designed.

Based on these problems and aims mentioned above, an IP for Janssen Logistics b.v is proposed to be designed. The IP can be designed based on the RA designed in Chapter 6. To build up the

IP, these are some designing difficulties for the IP. If these difficulties are overcome by the RA, we can conclude that the RA designed in Chapter 6 can be used as a template for IPs design. The difficulties for designing the IPs are listed as follows.

Difficulty 1: How to deal with data format difference between external objects;

Difficulty 2: How to protect the system security;

Difficulty 3: How to realize the collaboration between applications and devices;

7.4. Target architecture in Janssen Logistics b.v

To deal with the problems mentioned in the previous section, an architecture of the IP for Janssen Logistics b.v based on the RA is designed here.

1. Guidelines and principles

In the first step, the guidelines and principles will be taken into consideration. According to principle 1, we need to first check the number of objects that is going to be integrated. In Janssen Logistics b.v, there are various applications which are required to be integrated, such as FManage Secure v1.5, GLOW v2.9, PLAN-IT and so on. Besides, we should also consider the extendibility of the IP for new developed applications. For example, we need a warehouse recording software to manage products in warehouses. We also need a bidding portal for partners of Janssen Logistics b.v to find transportation tasks. Hence, it is necessary to use an ESB instead of an adapter to integrate external objects.

According to principle 2, it is suitable to use SOA design pattern as both applications and devices will be involved for integration. In our case, we will use an ESB and web services to implement the SOA design pattern.

According to principle 3, we should pick up appropriate building blocks for the IP. In our case, these building blocks are listed in Table VII. The left column gives standard building block concepts from the RA, the right column derives specific building blocks for the IP. For the concept of “external resources”, we have website of Janssen Logistics b.v, PLAN-IT, GLOW v2.9, FManage Secure v1.5, truck monitoring devices, truck GPS device. Besides, a warehouse managing software is required to organize product information in to be built warehouses. For the concept of “application integration API” and “device integration API”, there is no corresponding building block which is needed by the IP because the SOA design pattern is applied. For the concept of “service integration interface”, we create these building blocks: waybill creation, order confirmation, information receiving, contract creation, order file creation, track and trace, route creation, availability of trucks, distance and time, feed disruption, truck allotment, expenditure and credits records, route map and pick up& drop off. Besides, three new developed service integration interfaces should be added to the IP: record product amount, record product location and bidding. For the concept of “user interface”, we create a building block of user interface with user validation service. The user validation service can be implemented as alias services. Besides, user account authorization service and database operation services are also implemented as alias services. For the concept of “data exchange”, we create building blocks of message routing and service routing. For the concept of “data processing”, we create a building block of data transformation. For the concept of "service orchestration", we create a building block of order creation process. For the concept of "central database", we use two oracle databases, one for daily work and the other for backup. For the concept of "registry", we create a building block of service registry. For the concept of "other files", we have building blocks of order files and contract files.

TABLE IX. IDENTIFICATION OF APPROPRIATE BUILDING BLOCKS FOR THE IP

Building block concept from RA	**Building block for IP**
External resources	Website of Janssen Logistics b.v, PLAN-IT software, FManage Secure v1.5, truck devices, truck GPS, GLOW v2.9 software, warehouse managing software
Application integration API	X
Device integration API	X
Service integration interface	Waybill creation, order confirmation, information receiving, contract creation, order file creation, track and trace, route creation, availability of trucks, distance and time, feed disruption, truck allotment, expenditure and credits records, route map and pick up& drop off
User interface	User interface
Alias services	User account authorization, user validation, database operation services
Data exchange	Messaging services, routing services
Data processing	Data transformation
Service orchestration	Order creation process
Central database	Oracle, backup database
Registry	Service registry
Other files	Contract file, order file

2. Reference specification

In the second step, the reference specification will be take into account. The reference specification of the RA is made up of three parts: standards catalogue, design patterns and building blocks. All these have been dealt with in the previous part.

3. Conceptual model

Based on building blocks identified above, the conceptual model can be designed and used as the target architecture of Janssen Logistics b.v IP. The target architecture of Janssen Logistics b.v IP is designed in Figure 34. From the target architecture, we can see that each application in

Janssen Logistics b.v is connected by the IP in the form of service interfaces. A new application is designed for managing warehouse products. It connects with the IP by providing two service interfaces: record product location and record product amount. To check whether the designed IP have solved problems in Janssen Logistics b.v or not, a discussion section is give below.

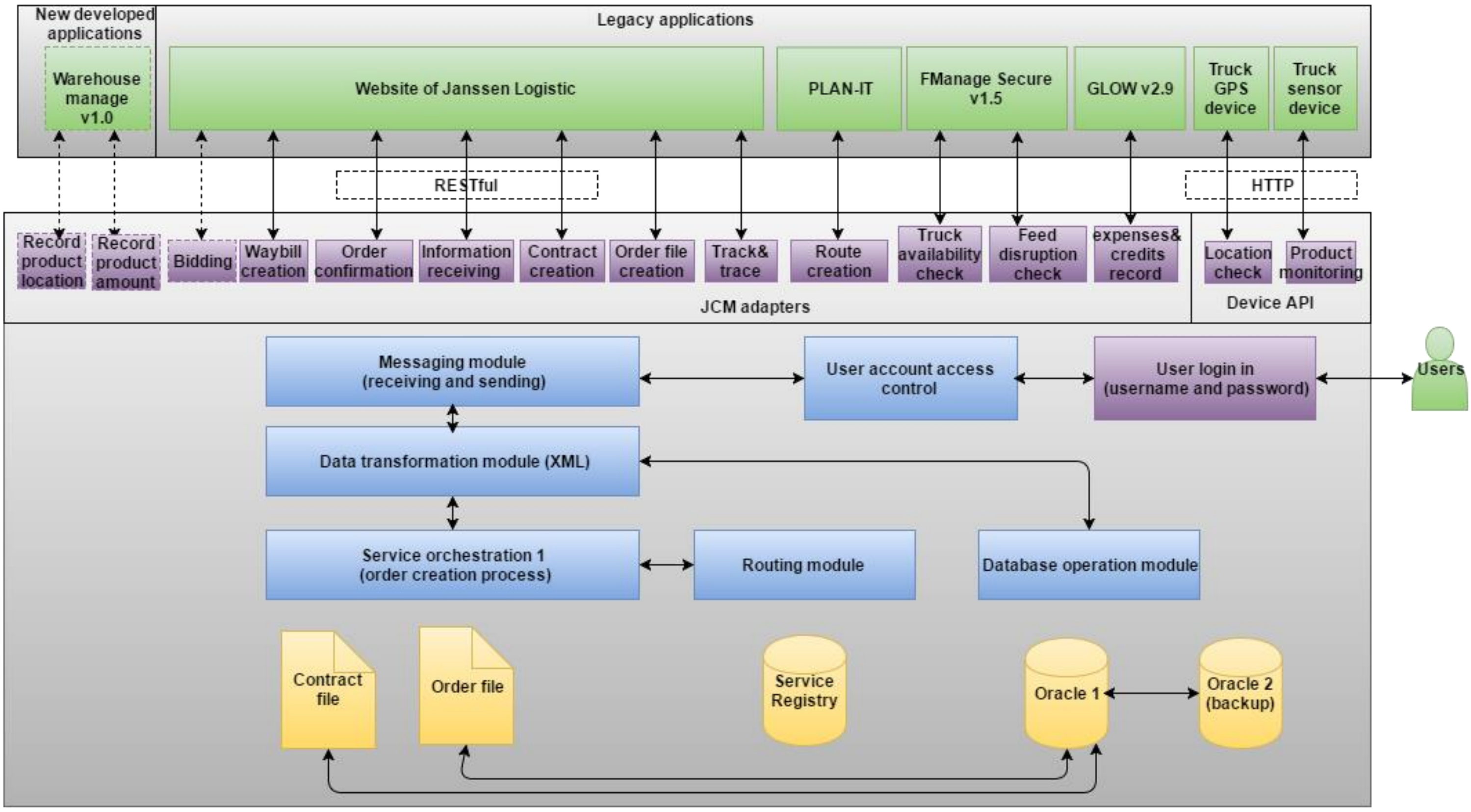


Fig. 34. The target architecture of the integration platform

7.5. Validation of the integration platform

In order to validate whether the RA can be used as a template for designing new IPs, we need to check whether problems are solved by the IP derived from the RA. We validate it from three aspects. First, we check whether the three problems are solved by the IP. Second, we check whether the three difficulties are overcome by the IP.

Check 1: Whether problems in Janssen Logistics b.v are dealt with by the IP?

In Janssen Logistics b.v, there are three problems. Firstly, applications and devices are distributed and lack of communication. This problem has been solved by the IP. Each object is connected with the IP and they are communicate via the IP. Secondly, a bidding portal should be added in Janssen Logistics b.v. This problem is solved by adding a new service to the website. Thirdly, a warehouse product management software is required for Janssen Logistics b.v. This problem is solved by designing the warehouse product management v1.0. Hence, all these three problems are solved by the IP.

Check 2: Whether design difficulties of the IP is overcome by the RA?

There are many difficulties of designing a new IP. The most popular one is to deal with data format. In Janssen Logistics b.v, data types in different applications are different, such as XML, JSON and so on. This difficulty is overcome by the building block of data transformation. Another difficulty in IPs is security. In our case, we add user account authorization and user validation services to the IPs. They guarantee the security of the IP.

As all these problems are solved by the IP and all design difficulties are overcome by the RA, we can make a conclusion that the RA can be used as a template for designing new IPs. Besides, we also identify some benefits of the IP here. First of all, all data and information can be managed and stored in a central location. Each object can share and use data in a standardized format. Secondly, the problem of security can be dealt with in a centralized way. Every user is assigned with specific authorization and authentication right. Thirdly, business processes can be automatically realized by adding a specific service orchestration module to the enterprise service bus. Moreover, new developed applications and new introduced devices can be easily integrated by the IP to fulfill more business requirements and business processes.

8. DISCUSSION ON THE WORK

In this research, we designed a RA for IPs. The RA was done and validated by three steps. Firstly, a SLR was carried out to extract existing IPs in literature. Secondly, the RA was developed by studying the SLR results. Thirdly, the RA was validated by a study case from Janssen Logistics b.v. All these three steps were carried out in correct ways and produced useful results in general.

The result of the SLR consists of 31 research papers. It provides different architectures of IPs in different domains, such as medication, transportation, supply chain and so on. These ahitectures and engineering documentations of IPs are fully used to develop the RA. However, there are three limitations on the SLR results. The first one is that only the term of "integration platforms" were applied for searching the SLR results. Some terms, such as "collaborative systems" and "cooperative platforms", are also relevant with IPs. These terms have the same meaninng and functions as IPs. To extract more IPs in literature, these terms can be used for retrieving more SLR results in the future. The second one is that the number of the SLR results is limited. It only gives 31 IPs in literature. To find more common elements from existing IPs, it is necessary to introduce more IPs from literature. The third limitation is that only research papers between 2010 and 2016 are taken into account. Although some papers in the past may be out of date, there might be some useful papers before 2010 as well. In general, the SLR results already have comprised most useful papers.

The development of the RA is done by extracting elements from the SLR results. Firstly, design patterns, building blocks and standards catelog are extracted from the SLR results. As some research papers have no specification or implication on which design patterns are used for their IPs, such as paper 5, 6, 9 and so on, it results in an incomplete collection of design patterns. The final result of design patterns extratced from the SLR results only comprises the SOA design pattern, web services, cloud computing and ESB. To provide more design choices for developers, more design patterns should be studied and added to the RA in the future. Besides, the classification of building blocks are only done by Quan Wang and Prince Singh, which may cause a potential incorrect classification of building blocks. For example, a "data filtering" building block can be classified into "data processing". It can also be classified into "QoS". This depends on researchers' experience. To increase the accuracy of the classification work, more experts in the field of information systems are required to involve in the classification work. Secondly, the reference model is designed by deploying and arranging building blocks. It provides a general model for application integration, service integration and device integration. However, there is only limited interpretation on the internal composition of each building block as well as relations between these building blocks. For example, there is no description of which tools or products can be used to realize "service orchestration". This can be done by learning more real implementations of IPs in the future. Thirdly, guidelines and principles are given. They not only provide steps of using the RA, but also offer rules of using standards catalog in the reference specification. These guidelines and principles can be improved after altering the reference specification and the reference model in the future. In all, the RA designed by studying the SLR results can be used as a template for developing IPs in any domains.

The validation of the RA is carried out by building up an IP architecture for Janssen Logistics b.v study case. The architecture of the IP is designed based on the RA designed in Chapter 6. By designing the IP for Janssen Logistics b.v, the enterprise problems in Janssen Logistics b.v are dealt with. The result of the validation proves that the RA for IPs can be used as a template for designing new IPs. As the Janssen Logistics b.v is only a study case, it more or less differs from real implementations of IPs. Hence, the validation of the RA needs more work. To improve the validation of the RA, an interview of experts in the domain of integration can be conducted in the future. Besides, a new software of IPs can be developed based on the RA to validate its correctness.

9. CONCLUSION AND FUTURE WORK

In this research, a RA for IPs is proposed. This RA can be used by developers to design new IPs. It provides guidelines and principles for developers to choose appropriate design patterns and rules. It also provides a reference model for developers to build and connect buidling blocks of IPs. Further, the RA provides some standards catalog and building blocks for developers. In general, our RA can be used as a template for designing IPs in any domains.

However, there is some future work to be done. First of all, the RA only comprises the design pattern of SOA. There are some other design patterns which can be used for design IPs, such as EDA and broker design pattern. They should be added to the RA to provide more options for IP developers. Seconldy, the validation of the RA is not rigorous since the study case of Janssen Logistics b.v is not a real case. Some other validation methods can be adopted to validate and improve the RA, such as carrying out an interview or designing a software based on the RA. Thirdly, the number of SLR research papers concerning IPs are limited, which may only pose a limited effect on the final result. Finally, the commonalities between IPs in different domains should be further figured out.

REFERENCES

[1] Fleischer, J., Häner, R., Herrnkind, S., Kloth, A., Kriegel, U., Schwarting, H., and Wächter, J.: An integration platform for heterogeneous sensor systems in GITEWS – Tsunami Service Bus, Nat. Hazards Earth Syst. Sci., 10, 1239-1252, doi:10.5194/nhess-10-1239-2010.

[2] Do LH, Esteves FF, Karten HJ and Bier E, "Booly: a new data integration platform," in BMC Bioinformatics, 2010, doi: 10.1186/1471-2105-11-513.

[3] Qing Li, Jian Zhou, Qi Rui Peng, Can-Qiang Li, Cheng Wang, Jing Wu and Bei-En Shao, "Business processes oriented heterogeneous systems integration platform for networked enterprises," in Computers in Industry, 2010, pp.127–144.

[4] Jinxing Hu, Qi Ma, Yiru Wang and Jingyue Zhou, "A GIServices-based transportation information resources integration platform," in SPIE Proceedings, 2010.

[5] P.Y. Wang, J. Shen, W.Q. Guo, C. Zhang and B. Zhang, "Cloud-based Government Procurement Information Integration Platform," in Journal of Digital Information Management 13(3):147-155 · June 2015.

[6] Fu Xiaolong, Liu Qixin and Fang Yuan, "Design and Implementation of University Level Unified Information System Integration Platform," in Computer Sciences and Convergence Information Technology (ICCIT), 2010.

[7] Zhexi Yang, Hailiang Huang, Ying Zhao and Yun Chen, "Development of Integrated Enterprise Application Integration Platform Facing Steel Circulation Supply Chain," in Computational Intelligence and Software Engineering (CiSE), 2010.

[8] "https://www.stir.ac.uk/media/schools/management/documents/centregradresearch/-How%20to%20do%20a%20systematic%20literature%20review%20and%20meta-analysis.pdf". Retrieved 23, May 2016.

[9] Vanessa Stricker et al., "Creating a Reference Architecture for Service-Based Systems- A Pattern-Based Approach," in Towards the Future Internet: Emerging Trends from European Research, 2010, pp.149-160.

[10] Undertaking Systematic Reviews of Research on Effectiveness. CRD's Guidance for those Carrying Out or Commissioning Reviews. CRD Report Number 4 (2nd Edition). NHS Centre for Reviews and Dissemination, University of York. March 2001.

[11] André Ferreira Ferrão Couto E. Vasconcelos, "Information Reference Architecture for the Portuguese Health Sector," in IEEE 17th Conference on Business Informatics (Volume:2), 2015.

[12] Hiroshi Satake, Tim Skutt, Mat Sherman and Wayne Eagleson, "Government Reference Architecture Extensions for Application to Base Stations," in IEEE Military Communications Conference, 2013.

[13] Ulrich Meissen, Fraunhofer ISST and Agnes Voisard, "Towards a Reference Architecture for Early Warning Systems," in Intelligent Networking and Collaborative Systems (INCOS), 2010.

[14] K.Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques, Springer, 2010.

[15] Dreher F1, Kreitler T and Hardt C, "DIPSBC--data integration platform for systems biology collaborations," in BMC Bioinformatics, 2012.

[16] Shivsubramani Krishnamoorthy and Ashok Agrawala, "Contextual information integration platform for humanitarian relief," in 1st International Conference on Wireless Technologies for Humanitarian Relief, 2011.

[17] Wu Deng, Huimin Zhao, Li Zou, Yuanyuan Li and Zhengguang Li, "Research on Application Information System Integration Platform in Medicine Manufacturing Enterprise," in Journal of Medical Systems, 2012.

[18] G. Randy Keller, "Scientific Workflows for the Geosciences: An Emerging Approach to Building Integrated Data Analysis Systems," in Geoinformatics: cyberinfrastructure for the solid Earth sciences, 2011.

[19] "https://www.informatica.com/nl/products/integration-platform-as-a-service.html". Retrieved 20, March, 2016.

[20] "http://www.tibco.nl/products/automation". Retrieved 20, March, 2016.

[21] "http://www.cisco.com/c/en/us/products/analytics-automation-software/integration-platform/index.html", Retrieved 20, March, 2016.

[22] Jerzy Brzezinski, Stanisław Czajka and Jacek Kobusinski, "Healthcare Integration Platform," in 5th International Symposium on Medical Information and Communication Technology, 2011.

[23] "Health Level Seven International," http://www.hl7.org/.

[24] "Integrating the Healthcare Enterprise," http://www.ihe.net/.

[25] "IHE IT Infrastructure Technical Framework, Volume 1, Integration Profiles," 2009.

[26] H. Zhiping and Z.Yi, "A New Information Integration Platform Based on Workflows," in 2010 Second International Conference on Communication Systems, Networks and Applications (ICCSNA), Hong Kong, 2010.

[27] M.Purgina et al., "An Approach for Developing a Mobile Accessed Music Search Integration Platform," in proceedings of the 2013 Federated Conference on Computer Science and Information Systems, 2013, pp. 267–273.

[28] "http://www.sun.com/service/refarch/". Retrieved 10, March 2016.

[29] "http://www.opengroup.org/soa/source-book/soa_refarch/intro.htm". Retrieved 10, March 2016.

[30] "http://dev2dev.bea.com/2006/09/SOAPGPart2.pdf". Retreived 10, March 2016.

[31] R.Cloutier et al., "The Concept of Reference Architectures," in Wiley InterScience, 2009.

[32] "http://www.service-architecture.com/articles/cloud computing/web_services_and_cloud_computing.html".

[33] Suhail Madoukh and Rebhi Baraka, "A SOA-Based e-Government Data Integration," in International Arab Journal of e-Technology, 2014.

[34] Yi Xu and Jun Yan, "A Cloud Based Information Integration Platform for Smart Cars," in Security-Enriched Urban Computing and Smart Grid, 2011.

[35] Qiaoyun Yun, "Research on Supply Chain Information Integration Platform for Remanufacturing," in E-Product E-Service and E-Entertainment (ICEEE), 2010.

[36] "https://www.quora.com/What-is-the-difference-between-enterprise-application-integration-software-and-web-services".

[37] M.J.P. Oude Weernink, "Development of an Integration Platform Metamodel," in 25th Twente Student Conference on IT, 2015.

Appendix A THE RESULT OF THE SYSTEMATIC LITERATURE REVIEW

The appendix A gives an overview of the total SLR result, the order of which is dependent on the English alphabet from “A” to “Z” of the title. The first column of the table assigns each research paper with a number. The second column gives the title of each paper. The last two columns give author information and publication year of each research paper.

No.	Title	Author	Year
1	A Best Practice of Enterprise Information Integration Platform in Telecom Industry	Teng, Ju-Ting	2013
2	A Cloud Based Information Integration Platform for Smart Cars	Xu, Yi	2011
3	A cloud service integration platform for web applications	Pinho, Eduardo	2014
4	A GIServices-based transportation information resources integration platform	Hu, Jinxing	2010
5	A new Information Integration platform Based on Workflows	Hu, zhiping	2010
6	An Agent/XML based information integration platform for process industry	X.Li	2010
7	An Approach for Developing a Mobile Accessed Music Search Integration Platform	Purgina, Marina	2013
8	An integration platform for heterogeneous sensor systems in GITEWS-Tsunai Service Bus	Dreher, Felix	2012
9	Analysis for Information Integration Platform on Warship	Xiong, Ying	2016
10	Booly: a new data integration platform	Dong, Long H.	2010
11	Business processes oriented heterogeneous systems integration platform for networked enterprises	Li, Qing	2010
12	Cloud-based government procurement information integration platform	Wang, P.Y.	2015
13	Contextual infomration integration platform for humanitarian relief	S. Krishnamoorthy	2011
14	Design and implementation of university level unified information system integration platform	Fu, Xiaolong	2010
15	Development of Integrated Enterprise Application Integration Platform Facing Steel Circulation Supply Chain	Z.Yang	2010
16	DIPSBC-data integration platform for systems biology collaborations	Dreher, Felix	2012
17	Healthcare Integration Platform	J. Brzezi	2011
18	HIP: Health integration platform	J. Woodbridge	2010
19	Human-centric data model and data integration platform enabling personalized product service systems for healthcare	G. Landolfi	2014
20	Integration Platform for a Leaning Environment	Butakov, Sergey	2013
21	Research on an information application integration platform based on SOA and web service	Wei, N.	2015
22	Research on Application Information System Integration in Medicine Manufacturing Enterprise	Deng, Wu	2012
23	Research on framework of integration platform for digital urban planning based- on service-oriented architecture	M. Fang	2010
24	Research on Supply Chain Information Integration Platform for Remanufacturing	Q.Yun	2010
25	Requirements for data integration platforms in biomedical research networks: A reference model	Ganzinger, M.	2015
26	SOA-Based E-Government Resource Integration Platform	Liu, Ying	2010
27	The design and implementation of collaboration service integration platform based on context-aware role based access model	Lu, S. P.	2014
28	The Design of Supply Chain Logistics Information Integration Platform for Auto Parts Enterprises Based on Multi-agent	Li, Ming	2015
29	The Research and Implementation of Supply Chain Resource Integration Platform on Textile Industry	G.Feipeng	2010
30	The Study and Implementation of Deep Integration Platform of Campus Information Based on SOA	Shen, jiquan	2012
31	Towards an Integration Platform for Bioinformatics Services	Llambias,Guzman	2014

Appendix B CLASSIFICATION OF SLR RESULT BASED ON IMPORATNCE/ RELEVANCE.

The appendix B gives a classification of the SLR result based on relevance. The relevance indicates how well the research paper is matched with our research. It is divided into three categories: "Most relevant", "Medium relevant" and "Most relevant". The first table shows the most relevant ones, with a number of 16 papers. The second table presents the medium relevant ones, with a number of 9 papers. The third table shows the least relevant ones, with a number of 6 papers. This classification can help other researchers to do future research in the domain of integration platforms.

No.	Title of most relevant papers
1	A Best Practice of Enterprise Information Integration Platform in Telecom Industry
4	A GIServices-based transportation information resources integration platform
5	A new Information Integration platform Based on Workflows
11	Business processes oriented heterogeneous systems integration platform for networked enterprises
12	Cloud-based government procurement information integration platform
14	Development of Integrated Enterprise Application Integration Platform Facing Steel Circulation Supply Chain
15	DIPSBC-data integration platform for systems biology collaborations
16	Healthcare Integration Platform
17	Heterogeneous System Integration Platform for Large Scale Business Group
18	HIP: Health integration platform
22	Research on Application Information System Integration in Medicine Manufacturing Enterprise
24	Research on Supply Chain Information Integration Platform for Remanufacturing
26	SOA-Based E-Government Resource Integration Platform
29	The Research and Implementation of Supply Chain Resource Integration Platform on Textile Industry
30	The Study and Implementation of Deep Integration Platform of Campus Information Based on SOA
31	Towards an Integration Platform for Bioinformatics Services

No.	Title of medium relevant papers
2	A Cloud Based Information Integration Platform for Smart Cars
3	A cloud service integration platform for web applications
6	An Agent/XML based information integration platform for process industry
9	Analysis for Information Integration Platform on Warship
10	Booly: a new data integration platform
13	Design and implementation of university level unified information system integration platform
21	Research on an information application integration platform based on SOA and web service
23	Research on framework of integration platform for digital urban planning based- on service-oriented architecture
25	Requirements for data integration platforms in biomedical research networks: A reference model

No.	Title of least relevant papers
7	An Approach for Developing a Mobile Accessed Music Search Integration Platform
8	An integration platform for heterogeneous sensor systems in GITEWS-Tsunai Service Bus
19	Information Management, Proposal for an Integration Platform Using Metadata
20	Integration Platform for a Leaning Environment
27	The design and implementation of collaboration service integration platform based on context-aware role based access model
28	The Design of Supply Chain Logistics Information Integration Platform for Auto Parts Enterprises Based on Multi-agent

Appendix C CLASSIFICATION OF BUIDING BLOCKS BASED ON THEIR FUNCTIONALITY

The appendix C lists all research papers from the SLR, with their building blocks, functionalities and mapping results. Each components or building blocks of an integration platform in the left column is mapped to a standard concept in the right column based on the functionalities of those building blocks.

1. A Best Practice of Enterprise Information Integration Platform in Telecom Industry

Building blocks		**Functionality**	**Mapping**
Internal components	Protocol transport	1. providing service interface for requesters;	Service interface
	Data transformation	1. transforming the format of service interface data into XML;	Data processing
	Access control list	1. constituting the authority of access interface;	QoS
	Flow control	1. limiting requests during exception;	QoS
	Message routing	1. integrating and routing the content for service requesters;	Data exchange
	Data type transformation	1. transforming data format;	Data processing
	Data translation	1. translating cross reference and correlating data across multiple incompatible data taxonomies	Data processing
	Database activation	1. publishing and subscribing proactively	Alias services
	Work flow	1. integrating complex business processes for the service requesters;	Service orchestration
	Monitoring system	1. providing message interception and reception, message filter, message grouping, time axis computing, error notification, statistics, cache, log report producing and user interface;	QoS+ data processing
External resources	Legacy systems	1. providing external web services;	External services and databases

2. A Cloud Based Information Integration Platform for Smart Cars

Building blocks		**Functionality**	**Mapping**
Internal components	Controller Area Network (CAN) bus sub system	1. providing simple and robust communications for in car networks; 2. providing an information sharing platform for all kinds of electronic systems; 3. integrating sensors, door locks and LED display;	Central database, Data exchange
	In car computer	1. collecting messages from sensors via CAN bus; 2. delivering these messages to services in clouds;	Service interface, Data exchange
	Gateway controller	1. converting CAN bus messages for in car computer use;	Data processing
	IT data bus	1. providing information of GPS, 3G module and Bluetooth for in car computer;	Data exchange
External resources	Cloud services	1. providing various car services;	External services and databases

3. A cloud service integration platform for web applications

Building blocks		**Functionality**	**Mapping**
Internal components	Cloud controller	1. handling authentication and access control to local cloud services; 2. managing new services; 3. aggregating user credentials; 4. providing service combination, decoration and orchestration; 5. providing API for other cloud services;	QoS, Alias services, Data processing, Service orchestration, Service interface,
	Database& BlobStore& Notification	1. storing cloud service resources;	Central Database, Other files
	Agent	1. representing the base SDCP user;	Alias services
	Client terminal	1. providing a user interface for accessing cloud services;	User interface
External resources	Cloud services	1. providing various functions;	External services& databases

4. A GIS-based transportation information resource integration platform

Building blocks		**Functionality**	**Mapping**
Internal components	Information conversion and exchange module	1. realizing multi-source traffic data conversion and exchange; 2. realizing distributed data warehouse construction; 3. realizing registration in meta-data and catalog management; 4. realizing database connection;	Data processing, Data exchange, Alias services, Service interface
	Information integration module	1. realizing information processing, information fusion, data analysis and data mining;	Data processing
	Macroeconomic statistics and decision support system	1. evaluating the whole or partial road network; 2. evaluating the effects of emergency or significant traffic management measures; 3. analysing data in integration information database according to user demands;	Alias services
	Information sharing and service module	1. providing multi-way and mixed information share, query and retrieval services;	Alias services
	Information publishing system	1. responsible for physical connection with traffic controllers and external public users;	User interface,
	Platform management system	1. monitoring the status of major hardware devices, software process and system resources, and the related environment state of the platform; 2. managing fault information of equipment and process and providing analysis for platform performance;	QoS
	Metadata database& integrated database	1. storing data;	Central database, Other files
External resources	Application systems	1. providing various traffic information for the integration platform;	External services& resources
	WebGIS traffic information service& others	1. allowing end users to retrieve traffic information;	External services

5. A new information integration platform based on workflows

Building blocks		Functionality	Mapping
Internal components	Data fusion	1. providing data schema for external resources when they are registered; 2. integrating external resources;	Data processing, Application API
	Workflow	1. constructing execution plan for applications; 2. producing the required output for applications; 3. managing workflows of business process;	Alias services, Service orchestration
External resources	Resource providers	1. providing various resources for the integration platform;	External resources
	Applications	1. retrieving data from various resources through the platform;	External applications

6. An agent/ XML based information integration platform for process industry

Building blocks		Functionality	Mapping
Internal components	User agent	1. implementing the interaction between end users and the integration platform;	User interface
	Management agent	1. managing active agents in the platform;	QoS
	Kernel agent	1. implementing data collection from subsystems; 2. realizing data analysis from subsystems; 3. realizing decision support;	Data exchange, Alias services
	Local resource agent	1. implementing the control of local process operation subsystems and new developed applications;	Application API, Alias services, Business process management
	Global database	1. storing data	Central database
External resources	Sub systems	1. providing various enterprise functions and resources;	External resources

7. An approach for developing a mobile accessed music search integration platform

Building blocks		Functionality	Mapping
Internal components	InputStyleSelection module	1. providing an user interface for end users to choose input style;	User interface
	MelodyContour& MusicScore& VirtualPiano& RhythmTapper	1. Providing interfaces of activities, such as melody contour, music score, virtual piano and rhythm tapper;	User interface
	Search output	1. returning searching results;	Data exchange
	FolkTuneFinder& PeachNote& Musicpedia	1. transferring user input to one search engine; 2. filtering outputs;	Data exchange, Data processing
	SOAP adapter& REST adapter	1. enabling appropriate communication transfer;	Service interface
External resources	Musicpedia& FolkTuneFinder& PeachNote	1. providing searching services and resources;	External services& resources

8. An integration platform for heterogeneous sensor systems in GITEWS-Tsunami service bus

Building blocks		**Functionality**	**Mapping**
Internal components (TSB)	Service interfaces	1. providing service interface for external applications;	Service interface
	Sensor interface	1. receiving all sensor incoming data; 2. managing sensors;	Device API
	SWE services	1. providing functions of sensor observation, sending alerts, asynchronous dialogues and tasking sensors;	Alias services
	Sensor registration	1. responsible for sensor metadata management;	Registry
	Service orchestration	1. managing business process;	Service orchestration
External resources	Sensors and sensor systems	1. providing sensor data for the warning system via TSB; 2. providing JMS adapter and CGPS Manager to enable sensor communication;	External resources
	Applications	1. providing various functions like decision support, registration GUI, monitoring and sensor system management;	External applications

9. Analysis for information integration platform on warship

Building blocks		**Functionality**	**Mapping**
Internal components	Task management and dispatching sub system	1. controlling and scheduling business processes; 2. providing an interface for external applications;	Business process management, Application API
	Jurisdiction control subsystem	No specific	Alias services
	Data access middleware	1. realizing data exchange, data transformation and other data processing; 2. providing data access agent;	Data exchange, Data processing, Alias services
	Task and information monitoring system	1. monitoring and authenticating tasks and status;	QoS,
	Database management system	1. storing data;	Central database
External resources	Applications	1. providing external resources and functions;	External resources

10. Booly: a new data integration platform

Building blocks		Functionality	Mapping
Internal components	Data warehouse& scripts	1. performing alias lookups and Boolean operations for medical experiments;	Central database, Alias services
	Data access control& user authentication	1. authenticating valid users; 2. responsible for data access control;	QoS
	Alias services	1. performing the task of alias resolution;	Alias services
	Web interface	1. providing a user interface for end users;	User interface
	Service interface	1.providing a service interface for external web services;	Service interface
	Application API	1. providing an application interface for medical application systems;	Application API
External resources	Medical web services	1. providing various medical related services;	External services& resources
	Medical application systems	1. providing medical functions and information to the integration platform;	External applications& databases

11. Business processes oriented heterogeneous systems integration platform for networked enterprises

Building blocks		Functionality	Mapping
Internal components	ESB server	1. deploying atom services and providing some QoS;	Alias services, QoS
	Platform database	1. registering local atom services and external web services; 2. registering service access agents (SAAs);	Registry
	Atom services	1. providing functionality as corresponding applications;	Alias services
	Service encapsulation	1. providing wizards to encapsulate applications to web services;	Alias services
	Service orchestration	1. providing a graphical user interface to generate complex service processes by orchestrating web services registered in the platform;	Service orchestration
	Service management	1. supporting service registration and service information access;	Alias services
	Service access agent	1. implementing authentication between two nodes, service authorization and transmission security;	QoS
	Platform management	No specific	Alias services, QoS
	Platform portal	1. providing a service ID and method ID for service invocation;	Service Interface
External resources	Sub node	1. providing external web services for the central node;	External services& sub databases

12. Cloud based government procurement information integration platform

Building blocks		Functionality	Mapping
Internal components	Physical resource module	1. Realizing computing and storage; 2. Providing network infrastructures that the platform needs	Central database, Alias services
	Resource abstract control module	1. Using virtualization technology to virtualize the underlying hardware resources to avoid the bottom hardware from failure 2. achieving monitoring and dynamic scheduling of computing, storage, network, and security;	Alias services, QoS
External services	Infrastructure as a service	1. building a virtual server, memory and network 2. supporting security services; 3. providing computing center for each unit of government procurement 4. allocating resources	External resources
	Platform as a service	1. addressing diversified data structures	External resources
	Software as a service	1. rebuilding business process of information resources, applications and platforms 2. encapsulating business process into a service	External resources
	Public service	1. providing cloud service portal for different users	External resources

13. Contextual information integration platform for humanitarian relief

Building blocks		Functionality	Mapping
Internal components	Location service& streaming service	1. resolving a GPS or Wifi-based coordinates to actual physical locations; 2. enabling any application to establish an audio or video stream with the system; 3. fetching or translating information to appropriate form for users;	Alias services, Data processing
	Information gateway	1. data processing;	Data processing
	Context module	1. making use of user information and their environment information; 2. providing context information of users and their environment information;	Alias services, Central database
	User interface	1. interfacing with end users;	User interface
	Application interface	1. providing API for external application systems;	Application API
	Service interface	1. interacting with external services;	Service interface
External resources	External services	1. enabling incorporation with external information to add to the clarity of the context (weather or transportation);	External services& resources
	External applications	1. retrieving information from integration platforms	External applications

14. Design and implementation of university level unified information system integration platform

Building blocks		**Functionality**	**Mapping**
Internal components (middleware service bus)	Information portal	1. Providing all types of users with a customized, secured and active pushed access environment;	User interface
	authentication& privilege	1. providing user authentication and resources access control;	QoS
	Data exchange	1. Exchanging data and information between different information systems via ESB 2. Providing a global data dictionary definitions	Data exchange, Data processing
	Integrated information service	1. implementing dynamic definition of business data query, query editing and definition of query results;	Alias services
	Information publishing service	1. Providing management control: unstructured information, document, storage, distribution, retrieval and version control	QoS
	Service interface	1. interacting with external applications and local services;	Application API, Service interface
	Service database& main database	1. storing data	Central database, Registry
External resources	Service information systems	1. providing functions of university information systems and resources;	External applications and resources

15. Development of integrated enterprise application integration platform for steel circulation supply chain

Building blocks		**Functionality**	**Mapping**
Internal components	Data interface	1. providing SOCKET interface, web service interface and SPA interface for integrating with various external application systems (receiving and sending data from and to application systems);	Application API, Service interface
	Data bus processing	1. controlling processing logic	Alias services
	Data transform module	1. transforming application data to platform data& platform data to application data	Data processing
	Data routing module	1. addressing of application data	Data exchange
	Data sending module	1. calling application systems and sending data	Data exchange
	Platform management	1. Configuration management; 2. Log management; 3. Authority management; 4. Monitoring management	QoS
External resources	Application systems	1. providing various functions and resources;	External applications& resources

16. DIPSBC-data integration platform for systems biology collaborations

Building blocks		Functionality	Mapping
Internal components	Java/ Perl parser	1. transforming raw data into XML format;	Data processing
	XML schema definition (XSD)	1. defining the structure of the XML files; 2. ensuring data integrity;	QoS
	eXtensible Stylesheet Language Transformations (XSLT) processor	1. normalizing XML files;	Data processing
	Curl	1. allowing adding or deleting documents, and optimizing index;	Alias services
	Foswiki web server	1. providing a convenient user interface;	User interface
	Foswiki client side	1. providing web search interface, such as Argo Genome browser, mzData viewer and graph viewer	Alias services
	Solr server	1. offering advanced query syntax and fast search routes;	Alias services
	Lecene index	1. can be accessed through user interface;	Central database
External resources	Other applications	1. providing data;	External resources

17. Healthcare integration platform

Building blocks		Functionality	Mapping
Internal components	Source web service	1. submitting patient personal data to index and registry services; 2. retrieving patient documents from hospital information systems or storing ; 3. converting data format from a hospital information system into documents;	Data exchange, Data processing,
	Index	1. identifying patients;	Registry
	Registry	1. organizing documents' meta data;	Registry
	Authorization	1. enabling a patient to grant access right to selected users and medical units;	QoS
	Mediator	1. providing application API for hospital information systems; 2. providing user interface for end users;	Application API, User interface
External resources	Hospital information systems	1. providing functions of hospital systems and storing patients information;	External applications& resources

18. HIP: Healthcare integration platform

Building blocks		Functionality	Mapping
Internal components	Data acquisition	1. Streaming raw data from sensors or local file systems and vice versa; 2. Providing an arbitrator that wraps a stream of data for multiple users; 3. providing API for sensors and file systems;	Device API, Application API, Data processing
	Data transfer	1. Realizing data transferring from sensors to PCs or phones; 2. Realizing data transferring from PCs or phones to a central server; 3. Realizing archived data streaming from a central server to sensors or phones;	Data transfer
	Data storage	1. Storing data in both sensors and PCs or phones	Central database
	Data processing	1. Providing data compression and decompression;	Data processing
	Event logging	No specifc	QoS
External resources	Sensors	1. generating raw data of patients' situation;	External resources
	Applications	1. providing functions of external applications and their resources to the integration platform;	External applications

19. Human-centric data model and data integration platform enabling personalized product service systems for healthcare

Building blocks		Functionality	Mapping
Internal components	Human body data model	1. providing a suitable data model for data representation;	Alias services
	Common services	1. representing the interface of DIP with the external services; 2. providing user interface;	Service interface, User interface
	Database	1. storing data from different sources;	Central database
External resources	Production system	1. providing information of production process of personalized goods;	External resource
	Biometric scanning device	1. providing customer data;	External resource
	Service applications	1. monitoring of health parameters; 2. collecting and managing sensor data;	External resource

20. Integration platform for a learning environment

Building blocks		Functionality	Mapping
Internal components	Content retrieval agent (CRA)	1. extracting content from local storage; 2. handling communication with external sources; 3. wrapping up content for information exchange between agents;	Data exchange, Data processing
	Delivery agent (DA)	1. delivering content regardless of the interface that is employed by the user;	Data exchange
	Assessment agent (AA)	1. Assessing content generated by users;	Alias services
	Assignment collector (AC)	1. collecting content generated by students;	Alias services
	Plagiarism detection agent (PDA)	1. checking assignment for plagiarism across internal and external sources;	Alias services
	Syllabi management agent (SMA)	1. controlling the content and structure of a syllabus;	Alias services
	Curriculum Management agent	1. controlling course sequence for the program;	Alias services
	User management agent (UMA)	1. authentication and session handling	QoS
	Course material management agent (CMMA)	1. defining the sequence learning objects to be delivered to the user;	Alias services
	Message switch	1. realizing message exchange between agents;	Data exchange
External resource	Application systems	1. providing functions of university information systems;	External resources

21. Research on an information application integration platform based on SOA and web service

Building blocks		Functionality	Mapping
Internal components	Enterprise portal	1. providing clients with an interface of access methods, authentication, authorization and so on;	User interface, QoS
	Service composition	1. composing autonomous services to achieve new functionality;	Alias services
	Business process	1. comprising a set of activities to accomplishing a specific organization goals;	Service orchestration
	Service monitor	1. monitoring and managing services, such as service objective management, dynamic monitoring management and security management and so on;	QoS
	SOAP router	1. realizing the message transmission of SOAP;	Data exchange
	Transaction service	1. realizing the transaction-driven asynchronous invoking by using JMS WSN;	Alias services
	Work flow	1. defining each step in business step so that a complete business operation is regarded as a series of activity in sequence;	Alias services
	Data service	1. supporting reliable transmission with various protocols, multiform communication models and persistent storage and transaction semantics and so on;	Alias services
	Message service	1. providing dynamic data conversion, data sharing, data synchronization and so on;	Data processing, Data exchange
	UDDI	1. storing data;	Central database
External resources	Web services, Applications, Databases	1. providing various web services, application functions and resources;	External services, applications and resources

22. Research on application information system integration platform in medicine manufacturing enterprise

Building blocks		Functionality	Mapping
Internal components	Edge server	1. dealing with the underlying complexities of the components of RFID infrastructure integration; 2. abstracting away underlying complexities of devices;	Device API
	RFID event processing	1. processing data and events from Edge server; 2. reducing redundant data, compressing the scale of events and converting information;	Data processing
	SOA exchange platform	1. connecting different functional components from different application programs through definition interface and contracts;	Application API
	Service component module	1. responsible for data filtering and input by data input module; 2. providing corresponding databases access interface by data access module;	Data processing, Alias services
	Electronic product code information service (EPCIS) and Object naming service (ONS)	1. returning results of global inquires and transforming EPC to URL; 2. responsible for the global EPC code registration and name services;	Alias services
	UDDI	1. responsible for web service registration;	Registry
External resources	Interior medical applications	1. providing medical functions and resources;	External applications
	Other medical applications	1. providing medical functions and resources;	External applications
	Frequency identity system	1. providing scanning data to the integration platform;	External resources

23. Research on framework of integration platform for digital urban planning based-on SOA

Building blocks		Functionality	Mapping
Internal components	Data management	1. responsible for data processing, data grouping , data filtering and so on;	Data processing
	Management and maintenance of exchanges	1. realizing management and maintenance of exchanges;	QoS
	Accepts of tasks	1. receiving data;	Data exchange
	Execution of applications	1. executing of applications;	Alias services
	Task processing	1. providing task processing;	Alias services
	Result handling	1. providing result handling;	Alias services
	Data post	1. realizing data post;	Alias services
External resources	Data module	1. storing data;	Central database
	Exchange module	1. providing external resources and services;	External services& resources

24. Research on supply chain information integration platform for remanufacturing

Building blocks		Functionality	Mapping
Information service platform	Workflow	1. managing workflow of remanufacturing;	Service orchestration
	Data exchange	1. exchanging data and messages between applications at different locations;	Data exchange
	Interface	1. providing application APIs for integrating application systems;	Application API
	Data warehouse	1. storing data from R&D systems;	Central database
	Data mart	1. storing data from remote applications;	Central database
External resources	Collaborative product R&D system	1. realizing cost analysis of product, life-cycle design of product and collaborative design system of product; 2. providing resources;	External applications& resources
	Other systems	1. providing functions of remanufacturing and resources;	External applications& resources

25. Requirements for data integration platforms in biomedical research networks

Building blocks		Functionality	Mapping
Internal components	Portlets	1. providing an interface to users;	User interface
	User management	1. providing user management for controlling access to portal pages and components like portlets;	QoS
	Document management	1. providing portal for integration of the user interface of the document management system;	User interface
	Statistic service	1. responsible for defining analytical methods and integration of data;	Alias services
	Security service	1. providing secure access to user account information by using Lightweight Directory Access Protocol (LDAP);	QoS
	Vocabulary service	1. responsible for the provision of network specific metadata to serve controlled vocabulary in standard formats;	Data processing
	Document management	1. defining analytical process;	Alias services
External resources	Service providers	1. providing various services and resources;	External services& resources

26. SOA-based e-government resource integration platform

Building blocks		Functionality	Mapping
Internal components (ESB)	Governmental and non-governmental application APIs	1. Providing access for both government and non-government institutions;	Application API
	Data routing& transfer	1. providing routing and transportation for web services;	Data exchange
	Other QoS	1. Ensuring QoS feature;	QoS
	Service registry	1. Providing a search point, database definitions and metadata for all services	Registry
	Government informational service	1. Providing access to basic informational queries	Alias services
	Service orchestration	1. Managing composite services;	Service orchestration
	Database management adapter service	1. Retrieving data from client systems; 2. Transforming data into a standard format; 3. Returning data to the requester;	Data exchange, Data processing
	Database replication service	1. managing replication between central database and ministries databases	Alias services
	System management service	1. Managing service bus and web services	Alias services
	Security assurance service	1. Providing authentication, authorization and non-repudiation for each web service	QoS
	e-Government portal	1. Providing access to the integration platform	User interface
	Ministries business applications	1. Providing business functions	Alias services
External resources	Governmental applications	1. providing various resources;	External resources
	Non-governmental applications	1. providing various resources;	External resources

27. The design and implementation of collaboration service integration platform based on context-aware role based access model

Building blocks		Functionality	Mapping
Internal components	Human-computer interface server	1. realizing account management (authentication/ authorization); 2. recording the creation, read, update and deletion events from users in log files; 3. offering new task notification and providing historical task notifications and status;	QoS, Alias services
	Content management server	1. realizing file management, file synchronizer, version control management and file sharing;	Alias services
	Workflow server	1. managing the task according to the order and logic of workflow; 2. grouping users into user, user role and user group and managing the authority of them;	Service orchestration, Alias services
	Database server	1. storing HCIS data, CMS data and WFS data, connection module and control module; 2. offering web services with properties to store and access;	Central database, Alias services
	Tools	1. being integrated with the integration platform for use;	Alias services
External resources	E-mail middleware service	1. allowing external programs to call APIs to send emails to others;	External services
	Watermark service	1. providing APIs for workflow server to add watermark into PDF files;	External services
	Automatic file management service	1. allowing external programs to update, upload, remove or download files;	External services

28. The design of supply chain logistics information integration platform for auto parts enterprises based on multi-agent

Building blocks		Functionality	Mapping
Internal components	Inbound logistic agent	1. responsible for purchase price maintenance, purchasing planning and purchase order generation;	Alias services
	Plant logistics agent	1. responsible for production plan;	Alias services
	Leaving logistics agent	1. responsible for docking business;	Alias services
	Tracing agent	1. responsible for relevant quality business functions;	Alias services
	Bar code tracking agent	1. tracking each step of production processes;	Alias services
	Interface platform agent	1. providing a unified platform of interfaces for sharing data;	Application API
	Production report agent	1. reporting related production and management aspects;	QoS
	Data processing	1. providing data management and data exchange;	Data processing, Data exchange
	Business processing	1. providing service orchestration;	Service orchestration
	Interface	1. providing user interface;	User interface,
External resources	Applications	Providing external application resources and functions;	External resources

29. The research and implementation of supply chain resource integration platform on Textile Industry

Building blocks		Functionality	Mapping
Internal components	Data conversion and sharing	1. providing information transmission, data integration, service composition, process integration, sharing of resources and others;	Data transformation, Data exchange, Service orchestration
	Application API	1. providing application API for external applications;	Application API
	Information security& authentication system	1. ensuring the authentication, legitimacy and legalization of customer information;	QoS
	Supporting module	1. providing corbelled functions;	Alias services
External resources	E-trading system	1. inquiring information; 2. browsing supply and demand information online; 3. sending or receiving messages online; 4. trading online	External applications& resources
	Logistic service system	1. providing logistic information service 2. logistic tracking; 3. portfolio and optimization of cargo 4. cargo transport scheduling and distribution	External applications& resources
	CRM system	1. Client finding ; 2. clients' management and retention;	External applications& resources

30. The study and implementation of deep integration platform of campus information based on SOA

Building blocks		Functionality	Mapping
Internal components	Data processing	1. publishing and managing integrative information;	Data processing
	Data exchange	2. realizing database integration and data exchange among distributed information systems;	Data exchange
	Business processing	3. realizing dataflow process reengineering;	Service orchestration
	Integrative database	1. providing a central database for distributed services;	Central database
	UDDI	1. responsible for service registration; 2. allowing service requesters to find desired services;	Registry
External resources	Other services	1. satisfying campus working activities; 2. providing relevant resources of the service;	External services and resources

31. Towards an integration platform for informatics service

Building blocks		Functionality	Mapping
Core of integration platform	C1: Asynchronous communication	1. realizing asynchronous communication between Traverna server and web services via ESB;	Data exchange
	C2: Events and notification	1. realizing event notification between service publisher and the platform;	QoS
	C3: Data transformation	1. transforming data format into the appropriate format required by services;	Data processing
	Bioinformatic resources	1. Providing resources for traverna server;	Central database
	Taverna server	1. enabling researchers to implement experiments by composing resources;	User interface
External resources	Web services	1. providing services and resources;	External services and resources

Appendix D JANSSEN LOGISTICS B.V CASE

Janssen Logistics b.v

The changing world of logistics

1. Introduction

Janssen Logistics b.v ® is a logistic service provider (LSP), primarily providing road and rail transport solutions, based in Rotterdam. It was established in 2000 by Dirk Janssen, who ran the company till he retired, in 2015. It is now being run by his son, Dennis Janssen. Janssen Logistics started in 2000 with a small fleet of 2 trucks but has now increased its fleet to 55 trucks. The year 2010 was important for the company as it then acquired 2 small logistics companies, which provide inland waterway transport using freight over barges. Most of the clients of Janssen Logistics are in The Netherlands but it also has many clients in Germany, Austria, Poland and Italy. These clients are big companies which need to transport goods from deep sea ports in The Netherlands to different inland locations in Europe. The origin of these good is Asia, Asia-Pacific and South America. After taking office, Dennis has a started major policy and strategic change. He was concerned with a stagnant profits and no real increase in market shares. A major policy change is moving towards becoming a 4PL[1].

In the words of Dennis, CEO Janssen Logistics "*…. The logistics domain is going to be even more disruptive in coming times. Our goal is to restructure and re-invent our operations and move towards becoming a 4PL logistics company. It is essential for us to remain relevant in our market….we have to be more context-aware and be able to swiftly adapt to changing situations. For example, currently our truck drivers get no information on how long to wait at Rotterdam port for the deep sea vessels to arrive and the cargo to be unloaded. It can take 1 hour or even 1 day. If you look at it objectively, essentially, it wastes a lot of company resources. We want to change that and use available data for better operation. Most of our competitors are doing it already, so why shouldn't we."*

[1] A 4PL primarily plays the role of a broker in logistics. It survives in the market based on its ability to collect information on market supply and demand, market players and real time operational conditions. Access to the right information and smart planning are the main competitive advantages of a 4PL.

2. Logistics domain

2.1 New entrants

The logistic domain has become quite complex in recent years. In 2000 when Janssen Logistics was established, the logistic scene in Netherlands was dominated by a few, large companies. These days different players (3PLs, 4PLs) are active and competition has increased. 4PLs have the freedom to offer a lower prices and choose between different transports providers. Moreover, there are companies which provide satellite data over traffic, congestion and expected weather situation. Previously, in a low competitive market, the usage of such data was rare. Firstly, such data was not available and secondly, the level of efficiency and planning was not necessary to survive in the market. Both these aspects have changes completely. Even the government (and semi-government) institutions make such data available to public. For example, Navic Logistics ® (previously a partner and now a competitor of Janssen Logistics) uses Tom-Tom® data to analyze the time in which certain highways are least likely to have a traffic jam. Based on this data, Navic logistics send its trucks preferably on those times to pick/deliver orders. Other companies integrate weather data and announced rail disruption (both accessible via website) with their planning. Not to mention big players like Procter and Gamble and DHL which having been doing this for quite some time. These companies also uses big data analysis approach to estimate demand peaks, optimal negotiable prices for orders, congestion scenario based on historic data and machine learning approach. The success of such novice ways is reflected in the profits of P&G and DHL.

2.2 Government regulations

In order to make freight transport more environmental friendly the European Union introduced measures to reduce the pollution being caused by logistics in Europe. As a part of EU 2020 goals, the EU wants to reduce pressure/congestion on roads. One of the measures is to let government offer logistic companies tax benefits when they use which use less polluting transport modes. The transport of cargo by road is not only more polluting but also takes more time. Nevertheless, it is the most flexible option for most LSPs. As a contribution towards the 2020 goals, new LSPs in the market, always tell/show to their client how much CO_2 will be emitted due to their order. Clients usually have the option to choose a greener option if they book their order *mode-free.* A mode free booking by a client implies that the client does not care what mode (road, rail, air or barge) is used to fulfill the order, as long as the order reaches the destination according to the agreed SLAs.

2.3 Better Services

Like every other business, logistics also has been effected by the internet boom. Clients demand round the clock access to data about their orders. They have become more demanding as they can now easily look for alternatives LSPs online and compare prices vis-à-vis of different LSPs. This could lead to the scenario where old clients of Janssen logistics can opt for better and new LSPs. Apart from the usual transport of goods, customers now want to see, live track and trace service, updates in case of delays and condition of the cargo (temperature, humidity, pressure etc.) especially in case of perishable goods. According to statistics, the inland transport demand of perishable consumable goods in Europe is expected to increase by 6.4% annually. This is one area where Janssen Logistics can expand.

Many new LSPs also provide more than one route option to the customers. These options have varied levels of CO_2 emissions, different add-on services and at times also gives the possibility of a using preferred mode of transport. Sometimes, it happens that due to operational disruptions the time of delivery is exceeded. In that case Janssen logistic has to pay a fine for the delay. Such things are mentioned in the contract when an order is accepted/drafted (the fine print, terms and conditions). With old client it is usually not a problem if the assignment reaches (a bit) late but one-off orders have to be met in time. This is required for good reputation management and further chances of contracts.

3. Operations

Transport has long term fixed contracts and one-off contracts for single delivery. The long term fixed contracts are with companies which have been clients of Janssen Logistics since years. They have fixed consignment, fixed loads, fixed destinations and fixed costs for the duration of the contract. These big companies also use the services of other transport providers besides Janssen Logistics. Currently, these companies are the main source of earning of Janssen Logistics as fixed contracts constitute 87% of all orders which Janssen Logistics undertakes. The remaining 13% are one off orders or short term contracts. Costs for long term fixed contracts are usually lower since booking in the case of transport by rail or barge can be done a long time in advance. This ensures cheaper rates, which thereby translates into better profits for Janssen Logistics. One-off contracts are contracts for cargo shipment which Janssen Logistics acquires via networking, contacts, bidding, marketing or word of mouth. These mainly involves the shipment of one large consignment from Rotterdam port to inland European destinations.

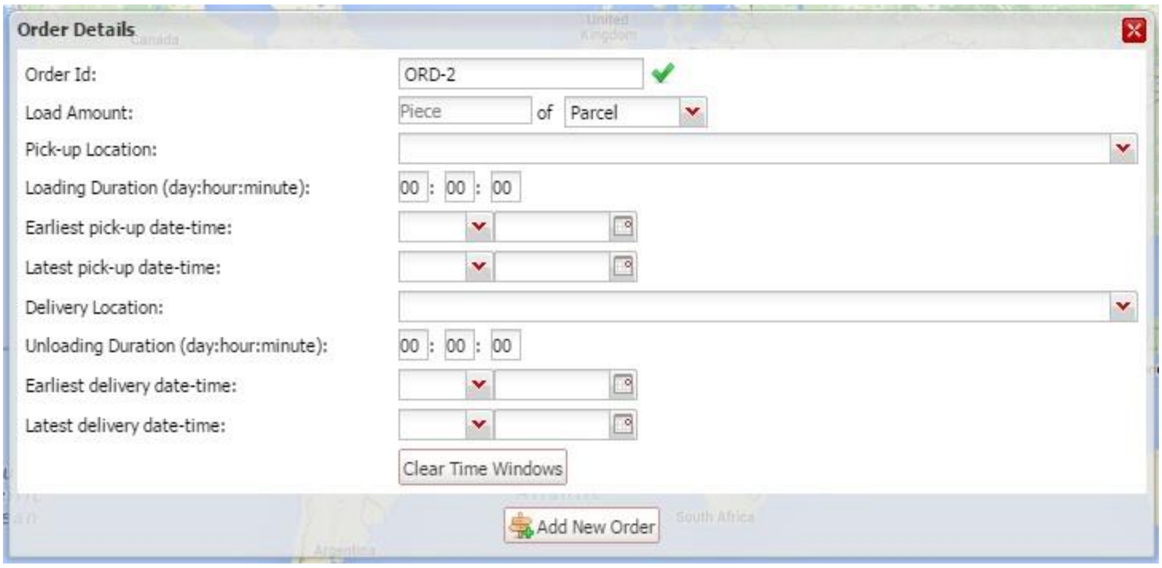
Sometimes, clients have special needs for special cargos. For example, if the shipment consists of transport of fruits/vegetables then temperature monitoring equipped trucks are required. The clients would like to see the current conditions of the orders at any time via internet. If certain conditions for transport (mentioned in SLAs) are threatened to be compromised during transport, sometimes the order has to be shifted to a different truck. Janssen Logistics has GPS enabled trucks but they are not fitted with special equipment to monitor good’s condition, for example, temperature of fruits/vegetables using temperature sensors. Therefore such consignments are outsourced to other companies and a commission is collected by Janssen Logistics. Each case is dealt uniquely and separately.

Order Entry

Clients can enter orders via creating a waybill on the website of Janssen Logistics. ORDER-IN is the back end application with an API for the website. Once an order is placed, an email alert is sent to the planners at Janssen Logistics, who check the order details. If they accept the order, they indicate that in the application (ORDER-IN) and an email is automatically generated and sent to the client. A part of the payment is received by Janssen Logistics so that the order is confirmed. Additional information like SLAs are sent to the email address of Janssen Logistics or conveyed via phone call to the planners. Based on these information, ORDER-IN generates a contract file and is in principle sent to the customer. The order file created by ORDER-IN is a XML file which is exported to .xl for frequent viewing and making changes. Excel sheet is open till the order is open, once the order is done it is uploaded in the database. In principle, this file can be edited only by the planners to add information or change plans. Only at times when the planners are not in the office, in that case, the assistant planner Ryan can also access the file and make changes.

Looking at the market, Janssen Logistics, would like to make a bidding portal, similar to what other 3PLs/4PLs have. Some or all order routes and legs can be made available for bidding. On this bidding portals, partners (or other logistics companies with have contract with Janssen Logistics) can view open transport routes and bid for them. In the long term this options gives more flexibility and price benefits. What is the best to integrate this portal is not yet explored but is one of the main project on the list of Dennis.

Figure 1: Order Entry at the website front end



Planning

Transport has a software tool named PLAN-IT. This software tools is basically a route planning software package, and holds the complete route map on which Janssen Logistics operates. This tool runs on the server and is updated when ever a new version is available. For fixed orders this software is rarely used because the planner knows very well which routes fixed orders takes. The tool is mostly used in case of one-off orders where the best route has to be decided. In case of more than one possible route the software provides more than one option. One feature of PLAN-IT is that routes can be marked as available or unavailable for certain time frames or deleted altogether. There are two experienced planners who have the final say as far as planning is concerned. The output plan produced by PLAN-IT can of course be superseded by the planners. Once a plan is made the concerned trucks/partners are informed and work schedule has to be made. Choosing the partners required a lot of calling and negotiation. In a survey conducted among employee 76% employee agreed that that coordinating and making up a resilient plan for an order is the most complex and time consuming job.

Crisis handling

Every now and then shipments are disturbed because of adverse environment conditions. They are usually solved by calling the concerned party like infrastructure provider[2] and getting information. There are still many area which may cause delay in shipment and consequently unhappy customers, not to mention monetary loss to Janssen logistics. A new employee was recruited and his work was to monitor disruption and feed disruption related information in an excel sheet. This excel sheet can be uploaded on FManage Secure v1.5 (See Order Reporting > Fleet Management > Truck) to update availability. Some time, if the disruption is long then the employee has to inform the planner, as it might require changes in current order and planned ones. Needless to say that a small traffic jam is not a classified as a crisis, but a truck break down or unannounced rail works is a crisis. The planners at Janssen logistics are experienced and 1 of them has been working since the company was formed. They decision is final and they are competent enough to handle any crisis and re-route orders in case it's required. Based on their intuition and knowledge they can times detect and foresee crisis prone routes.

Fleet Management

Truck - The fleet management is an important part of the business process for the company. The truck drivers are assigned orders which they have to fulfill in time. Situation which are out of control of the drivers are not used to judge the performance of the drivers. In case there are no crisis or disruptions the drivers don't call each day or hour to the office. When a customers wants to know the status of the order the truck driver is called. Janssen Logistic has a fleet management software, called FManage Secure v1.5, which is hosted on a server. This software maintains, records of available trucks, their maintenance schedule and allotment to different order. After the completion of an order, the details like kms travelled and time taken etc. are entered by an employee in the software. This information is used for maintenance and planning.

Inland Shipping – Barge shipping orders are taken separately as they have different commodities. The company has a good relationship with the most important terminal operators in the region. Since the inland shipping is less frequent, the company can call when there is an order and fixes a barge space for it. In high season there is sometimes a problem with finding sufficient barge space on demand (i.e. without advance booking). Either the planners (or the assistant planner) make advance booking for barge spaces or they send the order by truck if there is no barge space available during late booking. Each of the bookings can also be done on the website of the terminal operators and also via email/phone. The planners usually decides which order would be sent by barge and which by truck.

Rail – There are many complications in the rail transport. The service lines have to be booked in advance and the handling charges are also quite a lot. So usually the company doesn't use rail much. In case of disruptions the planner checks the website of the infrastructure provider and also which orders would be effected. In any case the orders would have to be delivered in time.

---

[2] Terminal operator in case of barges. Railway lines owner/maintenance companies in case of rail roads. Most of these companies' public disruptions or late arrivals on their websites just like airports display delayed and expected times of commercial airplanes. If this information is not available, Janssen logistics has to call them via phone to enquire about disruption/maintenance activities.

One more reason to avoid rail is that there are frequently many disruptions. Also, based on the working, the management thinks that they can only exploit the full benefits of transport by rail if they have warehouse facility of their own. (Renting warehouse is costly and availability is erratic). Dennis wants to emphasize and increase rail transport more, because that they he could prove to the Tax office (i.e. the government) that he is using suitable and more efficient ways of transport to reduce congestion on roads. But the planners don't seem to be too excited, because it needs them to work more in a domain which is complicated. A departure from current practices is never quite appreciated in any company and so is the case with Janssen Logistics.

Customer Care

The customer has to be told about the location of his order. There is standard operating procedure for this. For big clients the driver or the barge is to be called and the information is passed on to the client. For small orders, the standard reply is that the order will reach in time. The reception usually looks after the customer care and they see a common .xl file which is updated (based on disruption information) and is shared on the intranet with specific people. At the backend, there is a database server replying to these queries. An employee monitoring all current orders manually calls the truck drivers twice a day and asks the locations, then puts it in the databases. Therefore in weekends and holidays if a client queries (via the website) about the status of his/her order then the last updated status in the database from Friday (or day before) is shown. This database is backed up every night. Also the server is duplicated for security and resiliency.

Security

All computers are pass word protected. The data from the order entry is saved on disk and backed up every day. Also there is a database in place where all historic order data and financial payments are saved. It is also backed up on another server.

Customs

Different consignments require different customs clearance. The rules, mechanism, permits and documents required can be different in different countries and also depends upon the mode of transport. Since till now the consignment did not vary much the customs requirements were well known by the planners. It can be an obstacle for the smooth function of a 4PL which accepts all kinds of consignment and uses all modes.

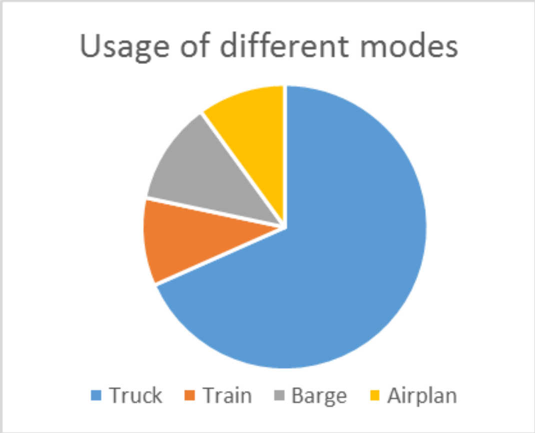
4. Infrastructure

4.1 Current Routes and map

The company has fixed corridors and lanes which have built up over the years. In the map above the main routes can be seen. Rotterdam is the location where deep sea terminals arrive. Arrival time of deep sea ship can be looked up at the website of the terminal. Based on these arrival times the departure times of truck is decided. Sometimes ships do come late (e.g. due to bad weather conditions, especially, during times of the year when weather conditions are less predictable).

There is an air transport route for fast shipping between the Rotterdam and London. Some cargo has to be transported from Rotterdam port to the Rotterdam Airport Cargo Terminal. Such shipments are relatively expensive and contributes to a less market share. Also for air delivery a lot of check are required at customs. Cargo from the company if it doesn't have all information from the order, it takes more time to get customs clearance. So the company doesn't take much order for air cargo. The service vs cost provided by other players in the air delivery market is much better than what Janssen Logistics can currently provided. Dennis is of the view that either is should be totally dropped from the portfolio or feasibility study should be don't to make air delivery of cargo cheaper and customer oriented.

The train routes on which the company operates are shown in black in the figure. There is a terminal at Dusseldorf from where there are further train routes to Berlin and Dortmund. The Dusseldorf terminal has intermodal facility also, i.e. the cargo can be shifted between barge and railway. Essen is currently terminal for barge transport from the company.



Once an order is final, changes to these orders are not that easy because the company has to 1. Make bookings in advance and the earlier they make the bookings, the cheaper it is. 2. Changes are expensive, based on contracts. The planner based on this knowledge knows what orders have to be combined with (half truck load [HTL] to full truck load [FTL] and half barge load [HBL to full barge load [FBL]). The order which is assigned to the truck driver is shown on a device, attached to every truck. It shows the pick up time/location, drop by time/location. Any other message sent from the office and be shown on this device. The truck has an inbuilt GPS showing the route map, and guidelines on best route based on the current traffic situation.

4.2 Warehousing

As mentioned before Janssen Logistics doesn't warehouses of its own. The current situation in this regard is aptly summarized in a snippet below from the interview with Henry, Chief Planner at Janssen Logistic.

In the words of Henry "*….it would give us a lot of freedom to plan and re-plan orders if we had our own warehouses. Almost all of the times, our trucks return empty, which cause revenue loss and extra expenditure. In case we had our own warehouse, our truck can pick and store orders there. In an ideal scenario, we could bid order routes for return journey from different partners.*

….points for switching between transports have to be identified and storage facility there would be critical. As I said before, we identified location where we could rent/buy warehouse space and use it for switching between barge and truck load to make flexible and more time efficient plans. These warehouse have different requirements and their back office, or how to say, IT is not connected to ours so we have to always makes some calls and do it manually. It is slow but so if we can we avoid it, we are happier (chuckles)"

4.3 Finance

Finance department uses GLOW v2.9, a finance software to keep all track of all expenses and credit. The software is used for bookkeeping but has additional paid feature to calculate for every order the expenditure for every leg/trip and how much it will cost. Of course, some basic information has to be filled on the basis of which this calculation can be made. Janssen logistics has the basic free version and it suffices to keep track of all employee, orders expenditure, salaries, credits and profit. CFO and her assistant only have access to the software.

5. Organizational Structure.

Although Dirk has retired yet he is still playing the role of a mentor, frequently attending meetings with new clients and advising Dennis over new ventures. Apart from that, there is a board consisting of CEO (Dennis), CFO, Chief Planners and HR Head to decide for important strategic decisions. There IT is looked after by Susan and Arthur. They have been associated with company from 2003 and 2007 respectively. The HR Head is an one-man department, looking after personnel management and drivers welfare. The trucks drivers sometimes on contract and sometimes permanent. They are mostly on trucks so they are not really present in office. The front desk consist of Sandra and Cristina. In case both of the chief planners are absent there is Ryan (assistant planner/trainee) to monitor the planning. Jan is the new employee on pay roll (on 2 year contract) for disruption reporting and management. He works closely with Ryan and the 2 chief planners. Finance department is headed by Elizabeth and her secretary Yolanda.

6. Questions

For all question below, make necessary assumptions wherever required. Along with the assumption clearly state any design choice you make.

1. Summarize the market changes in logistics domain. What better services must be provided by LSPs to remain relevant in market?
2. Model the current EA of Janssen Logistics b.v.
3. How can the current IT architecture of the company help in achieving the new services? Mention any missing IT capabilities?
4. Are there any additional business processes required to achieve the new services?
5. Model the to-be EA of Janssen Logistics b.v.
6. In the new scenario, how will the business model of Janssen logistics would change? Are there any changes in contract conditions and profit sharing among partners?