European Commission - DIGIT

Toolbox
Software Architecture Document

Version <0.1>

[

Revision History

Date	Version	Description	Author
16/03/2011	0.1	First draft	SBU
24/10/2011	0.2	e-fulfillment is now part of the toolbox	SBU

Table of Contents

Software Architecture Document

1. Introduction

The Software Architecture Document provides an overview of the Software Architecture put in place for the Toolbox project. More specifically, the Toolbox is an application designed to allow read of EC invoices via a web interface: the toolbox application is a "global name", toolbox is currently composed of e-invoicing, e-ordering and e-fulfillment.

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This document describes the design of the Toolbox application only. It doesn't explain how the external applications, such as the web service provider or the Back Office, are designed.

1.3 References

e-PRIOR Interface Control Document:
https://webgate.ec.europa.eu/CITnet/svn/EPROCUREMENT/trunk/03-Development/01 e-PRIOR/0006 Deployment/ICD/e-PRIOR_Interface_Control_Document.doc

e-PRIOR SAD: https://webgate.ec.europa.eu/CITnet/svn/EPROCUREMENT/trunk/03-Development/01 e-PRIOR/0003 Analysis & Design/eINV_ARCH.doc

RefApp: http://www.cc.cec/wikis/display/RefApp/Home

2. Architectural Goals and Constraints

2.1 E-PRIOR

The toolbox is not responsible to store and archive legal documents, e-PRIOR is used for that purpose. Therefore e-PRIOR is called every time a document which is not in edition needs to be retrieved.

For reusability and portability reasons, e-PRIOR exposes the retrieve document functionalities as web services.

Other web-services from e-PRIOR are used:

- Internal Query Request Service
- Internal Status Request Service
- Internal View Request Service
- Internal Retrieve Request Service
- Send Invoice Service
- Send Order Response Simple
- Other services might be implemented later on.

More information about those services can be found in the SAD of e-PRIOR and/or in the e-PRIOR use cases (see reference section).

2.2 RefApp

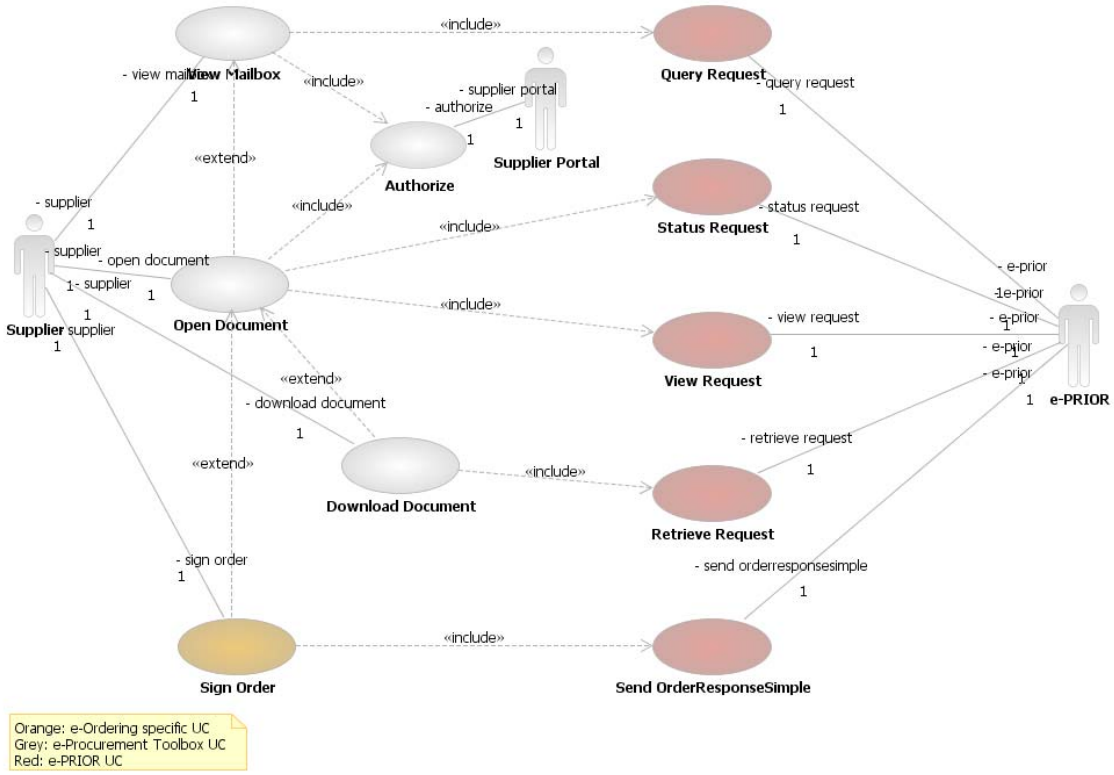
The application has been initialized using the RefApp (see http://www.cc.cec/wikis/display/RefApp/Home). Every architectural choice has been made to be compliant with the RefApp, this includes the following ones:

- Separate projects for each modules
- Build with Maven
- Log in files with Log4J
- Data access in JDBC (using SpringJDBC template)
- Data source defined in JNDI
- Use of:
    - aop caching
    - aop logging
    - aop transaction
    - aop query pagination
    - ECAS for authentication
    - JCore-Security for authorization
    - Ergonomics and Spring MVC
    - RefAff file upload
    - Spring
    - JMS
    - Web service client generation
    - JUnit
    - JMeter

3. Use-Case View

The use cases implemented by the toolbox application are located in the Toolbox repository:

https://webgate.ec.europa.eu/CITnet/svn/digit-sportal/Toolbox/Docs/trunk/003 Analysis & Design/Use cases

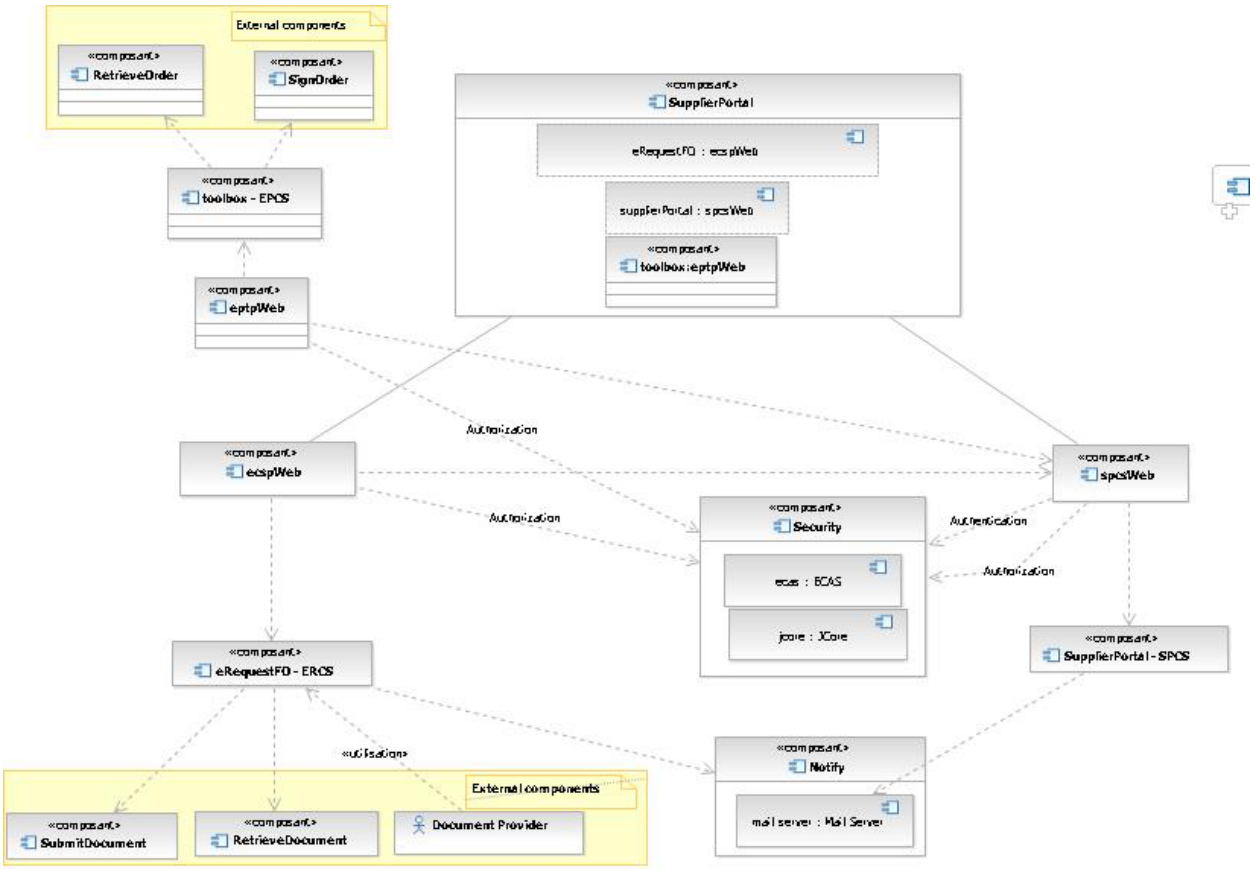


4. Logical View

The toolbox application is a web application embedded in the Supplier Portal application.

The main logical components are:

- spcsWeb: The supplier portal web application.
- SPCS: The supplier portal services.
- ecspWeb: The eRequest front office web application.
- ERCS: The eRequest front office services.
- eptpWeb: The toolbox front office web application
- EPCS: The toolbox front office services



The supplier portal , eRequest Front Office and the toolbox use a set of utility components:

- Security: for the authentication and authorization.

The ERCS component has dependencies to external components:

- RetrieveDocument: to retrieve documents.

4.1 Overview

This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.

4.1.1 eptpWeb

| *eu.ec.digit.eptp*

| *domain:* eptpWeb specific domain objects

| *controller*: web application controllers:

| *welcome*: the home page

| *mailbox:* the mailbox management

| *document*: the document management

| *errors*: the errors management

| *service*: service classes which use the ERCS services to provide the ecspWeb functionalities

| *mailbox*

| *document*

| *user*

| *support*: utility classes

4.1.2 EPCS

The EPCS component is composed of several modules packaged in separated JAR files.

Each module is described in the Architecturally Significant Design Packages chapter.

- epcsDomain: Domain classes
- epcsBase: Data layer
- epcsCore: Business layer
- epcsUtils: Utility classes

epcsDomain:

| *eu.ec.digit.epcs.domain.model*

epcsBase:

| eu.ec.digit.epcs.base

| *dao*

| *order*

| *query*

| *retrieve*

| *status*

| *view*

epcsCore:

| eu.ec.digit.epcs.core

| service

| *order*

| *query*

| *retrieve*

| *status*

| *view*

epcsUtils

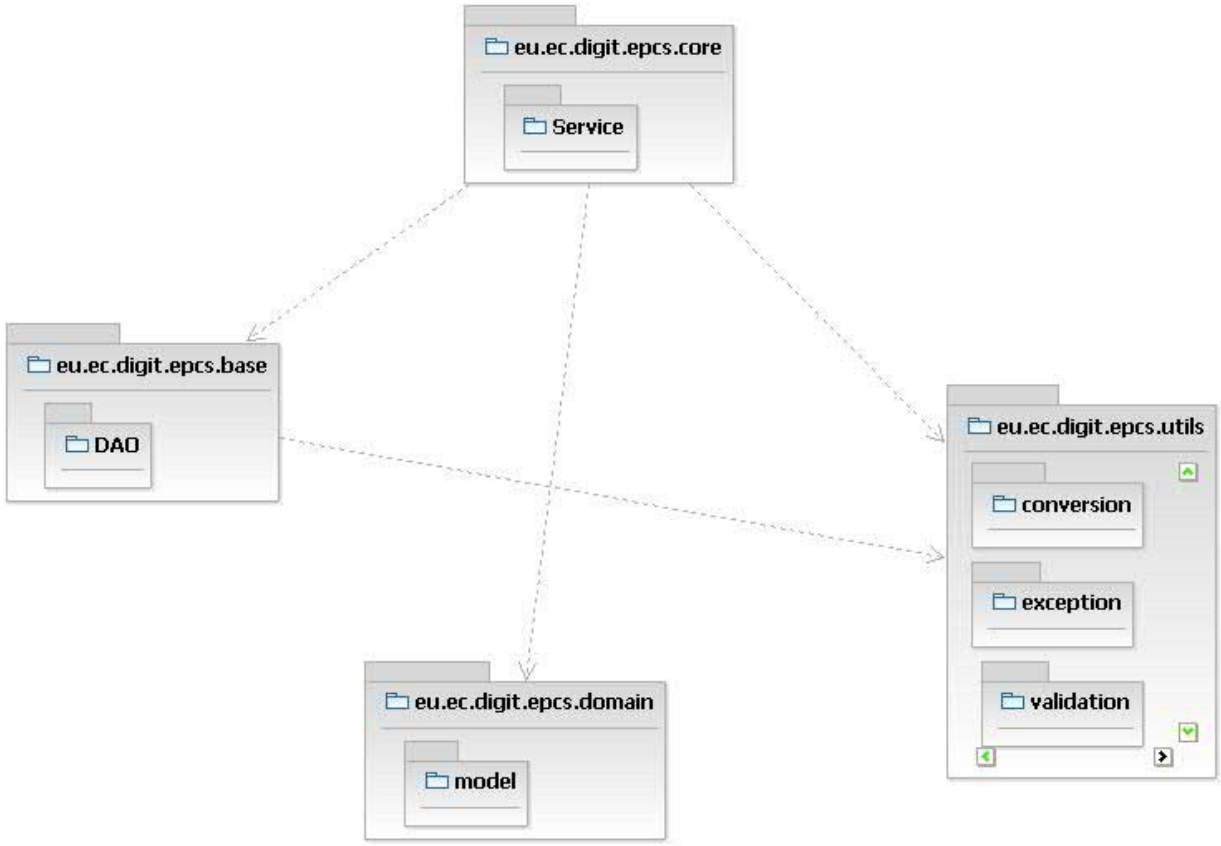
| eu.ec.digit.epcs.utils

| conversion

| exception

validation

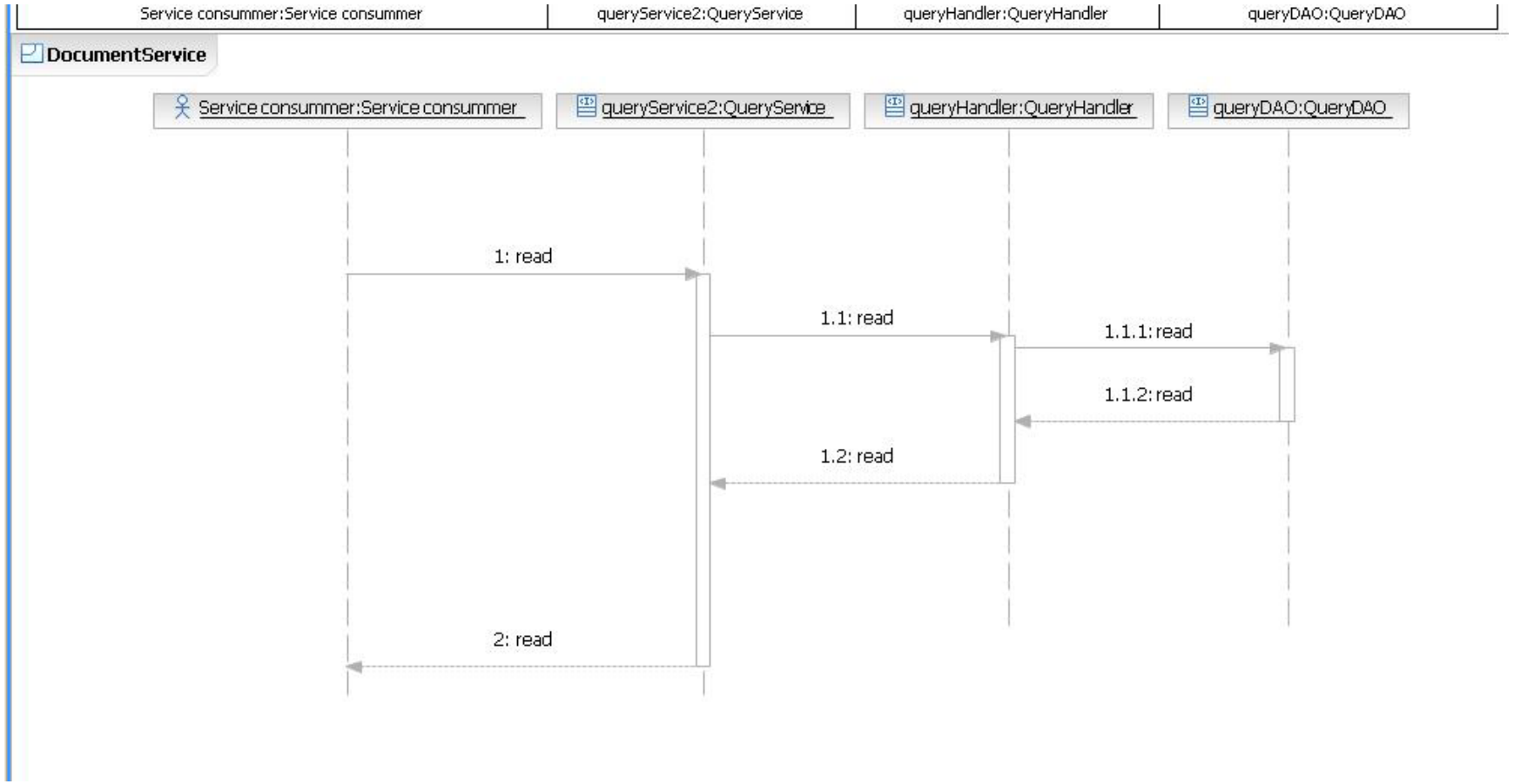
4.2 Architecturally Significant Design Packages



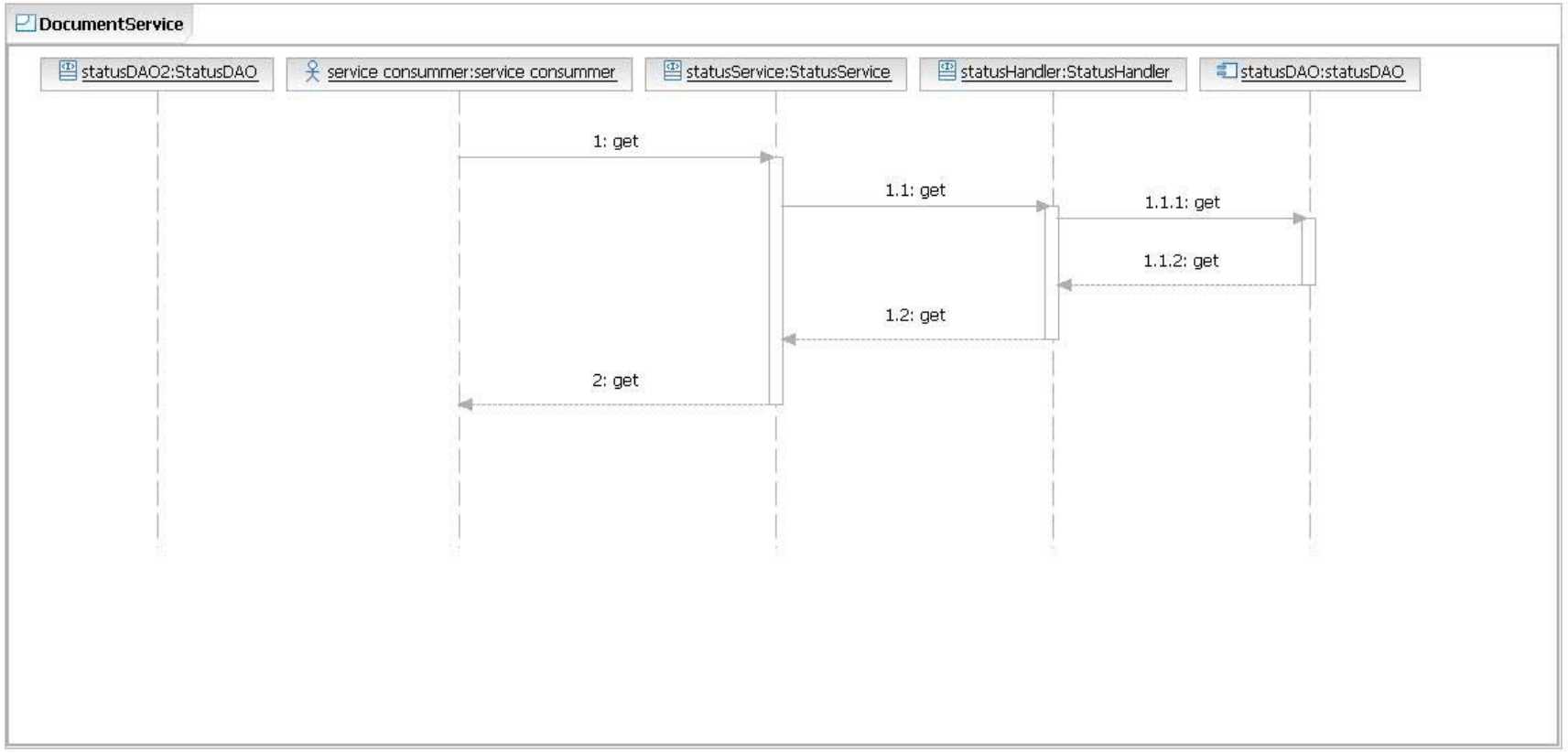
4.3 Use-Case Realizations

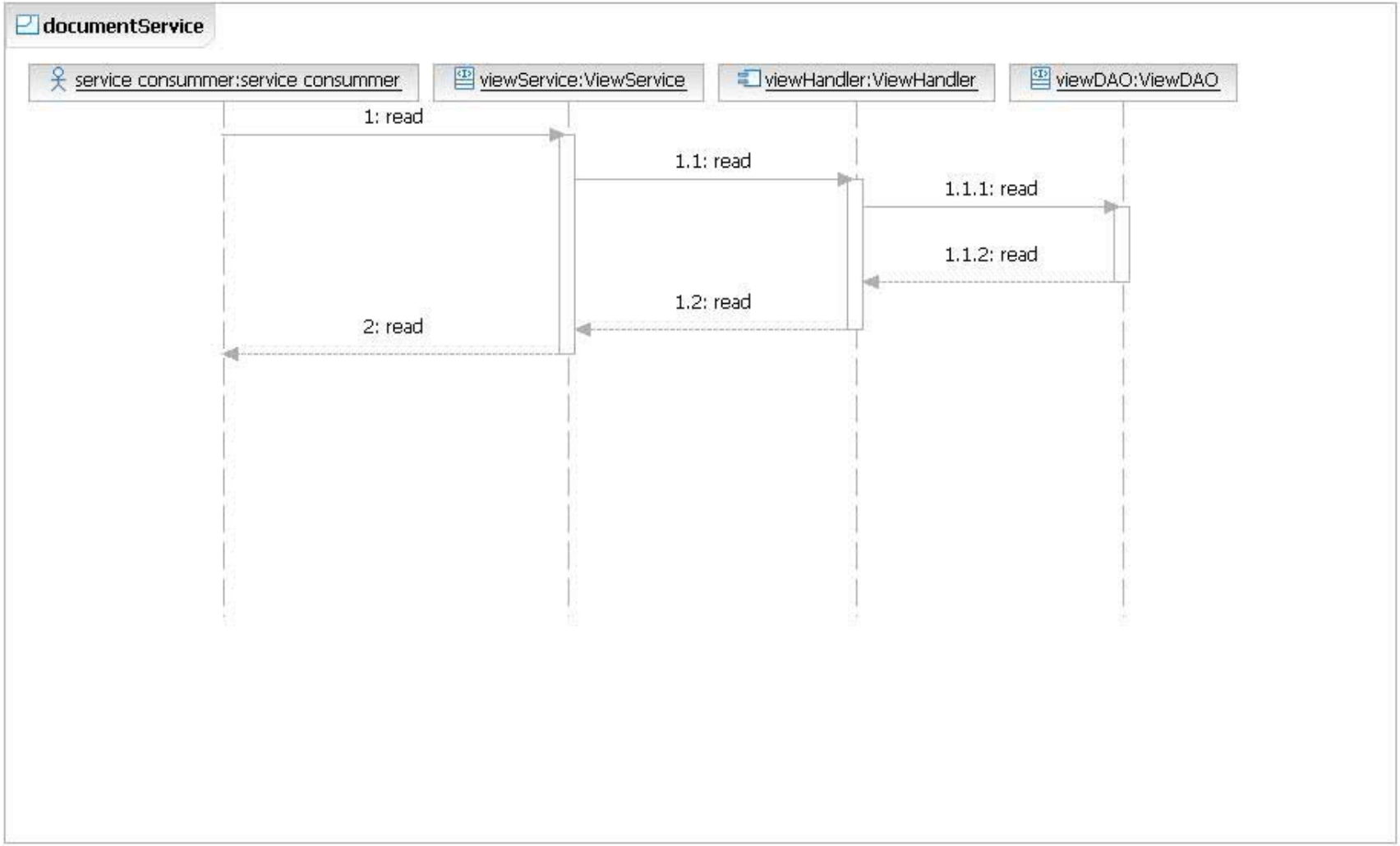
This section illustrates how the application actually works by giving a few selected use-case realizations, and explains how the various design model elements contribute to their functionality.

4.3.1 Mailbox

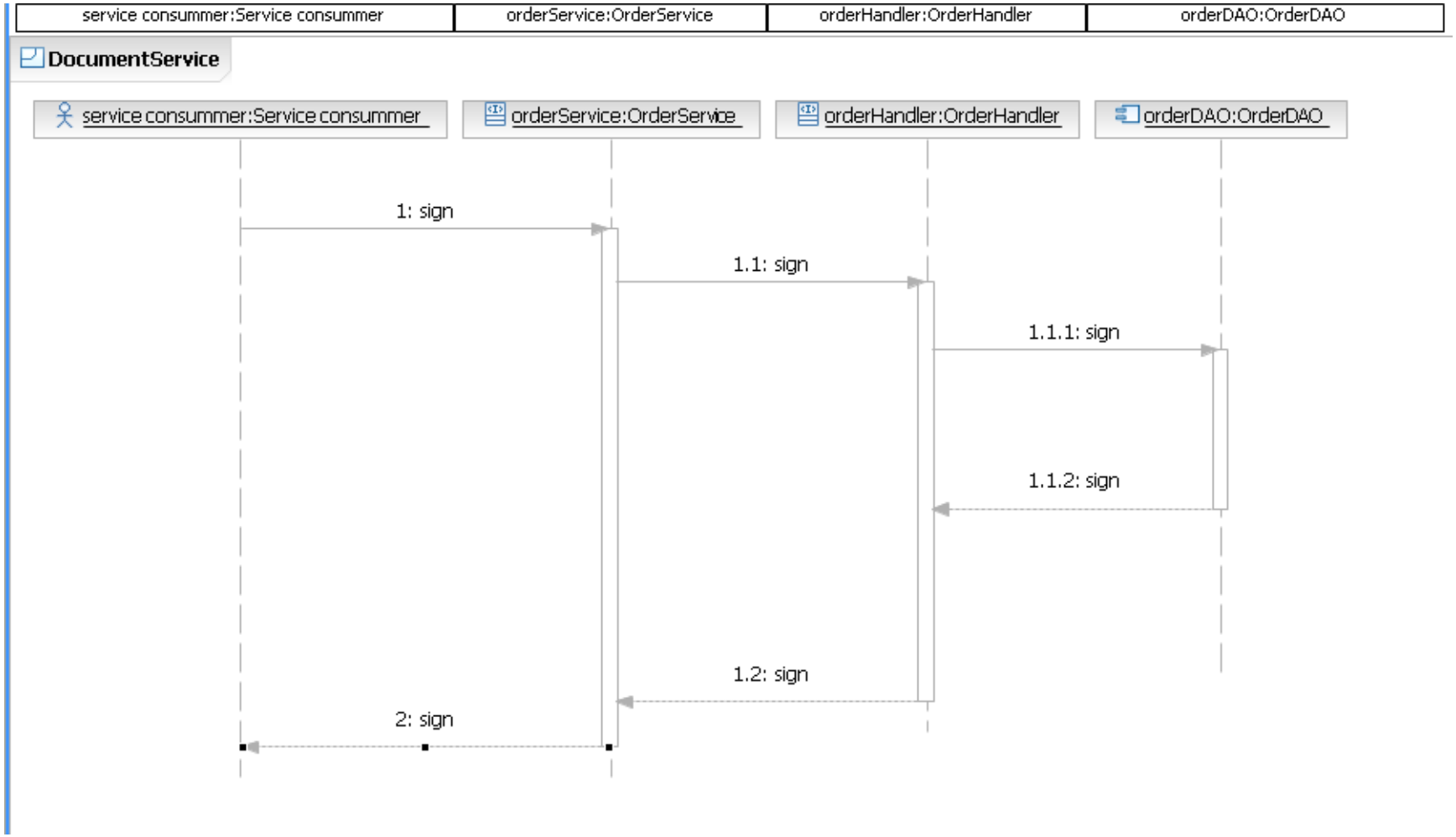


4.3.2 Open Document

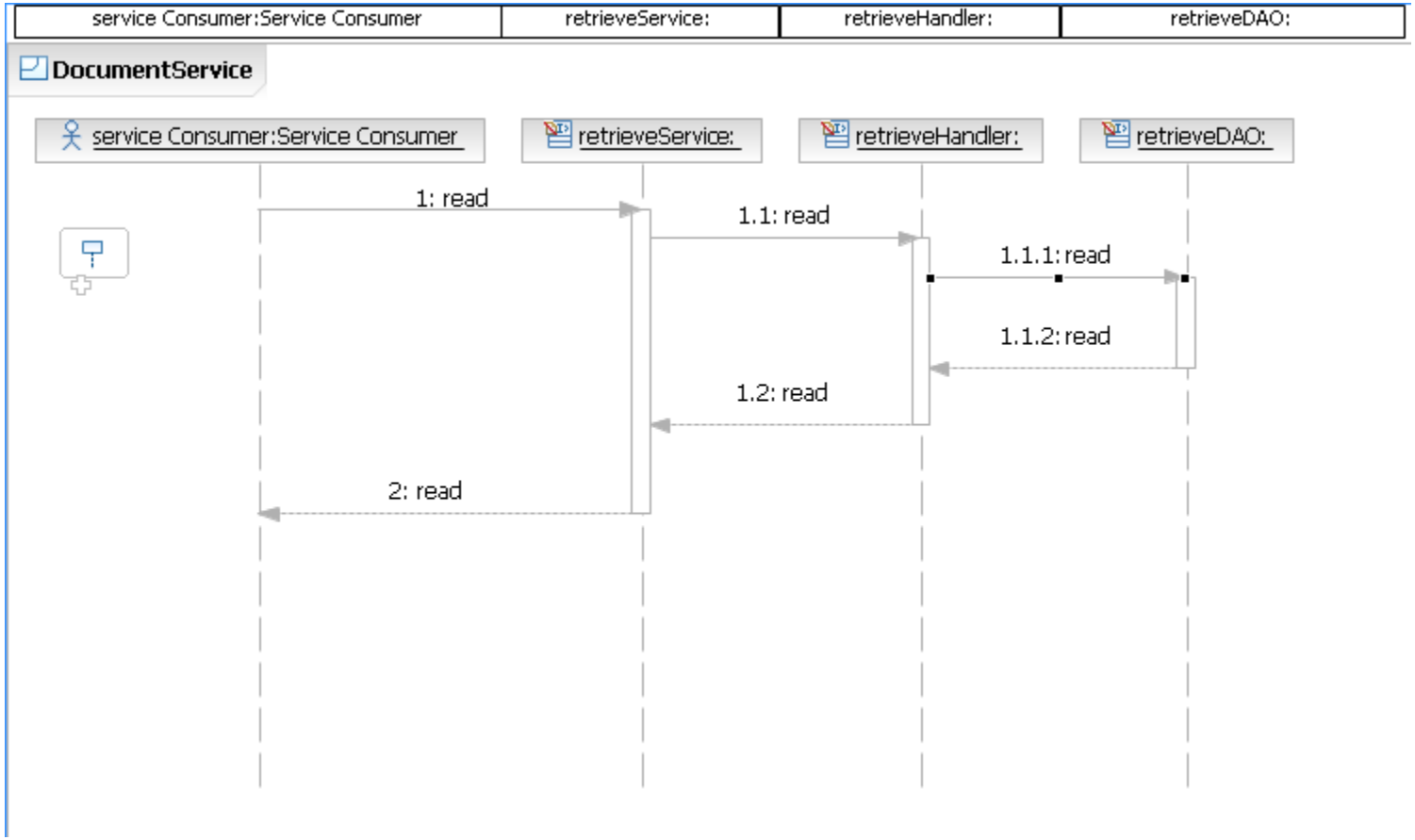




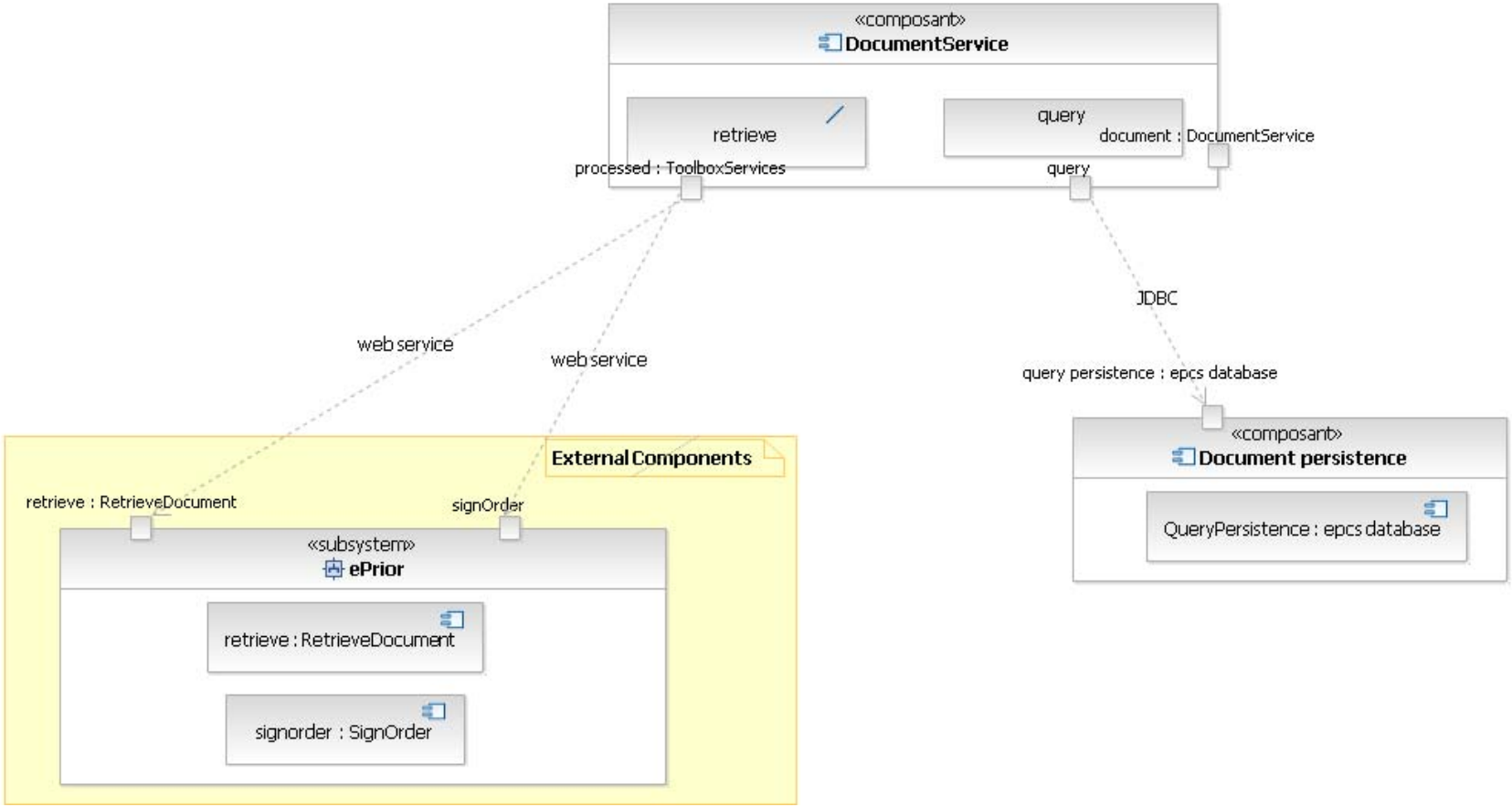
4.3.3 Sign Order



4.3.4 Download



5. Process View



6. Deployment View

The application server should be an weblogic server 10.3. The database should be an oracle 10g.
The CPUs, memory… should be aligned with the Mirella request.

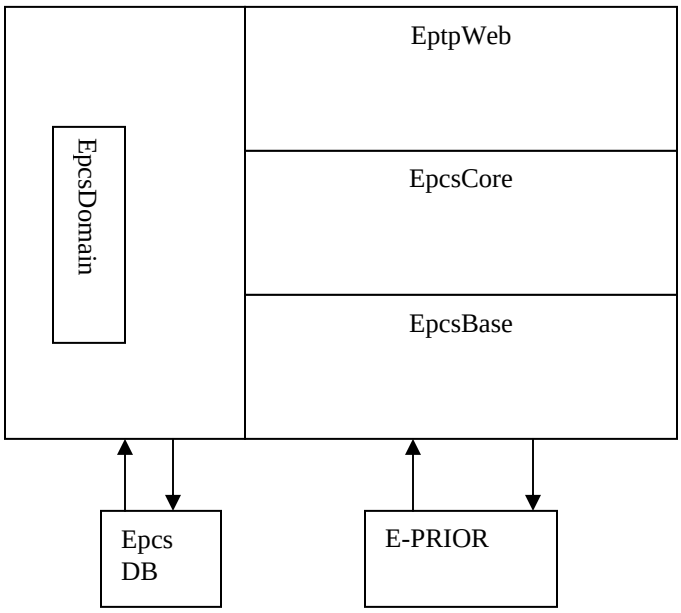
7. Implementation View

This section describes the overall structure of the implementation model, the decomposition of the application into layers and subsystems in the implementation model, and any architecturally significant components.

7.1 Overview

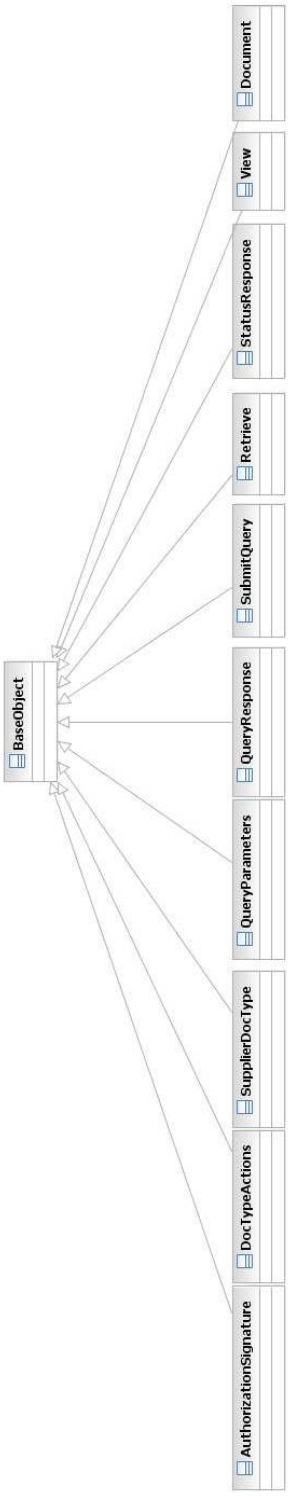
The EPCS project is composed of three main layers:

- eptpWeb: The presentation layer (implementing the MVC pattern)
- epcsCore: Business layer (services + handlers)
- epcsBase: Data access layer

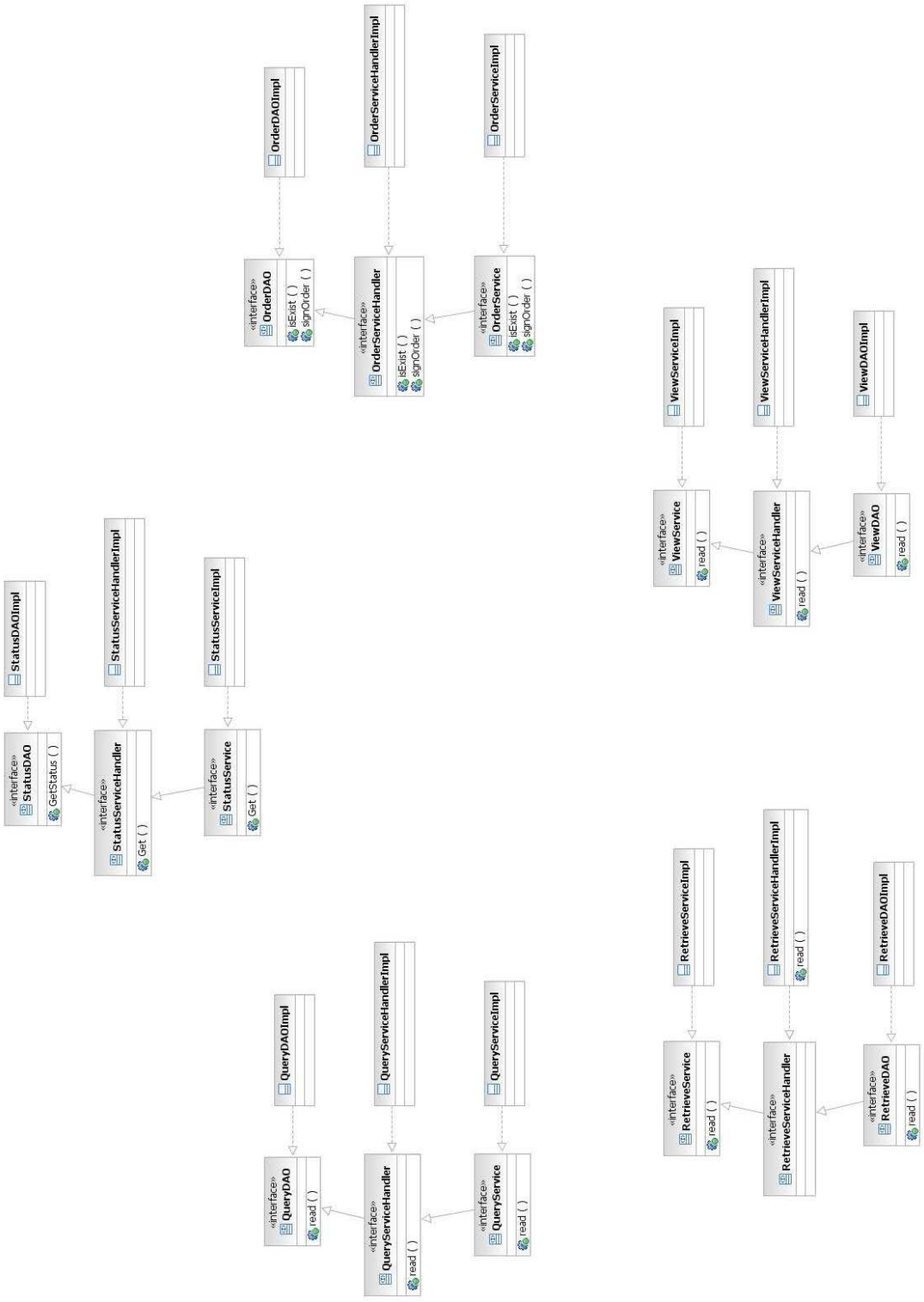


7.2 Layers

7.2.1 epcsDomain

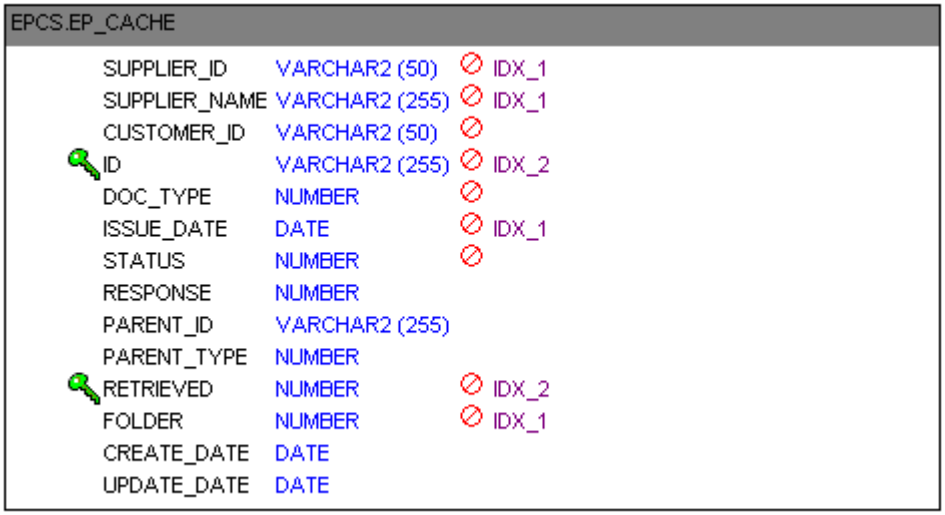


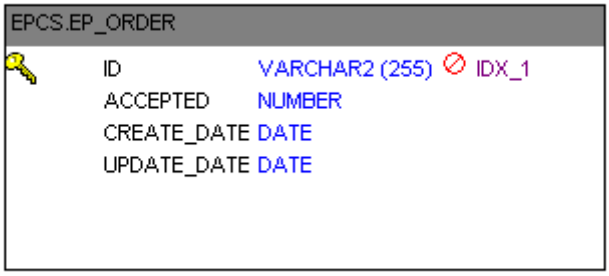
7.2.2 epcsCore and epcsBase



8. Data View

The EPCS database stores the primary data of the orders or invoices stored in e-PRIOR.





In order to use the supplier portal Front Office application, the user must be identified and authorized by the system. To do so, the supplier portal stores the user information in its database.

9. Size and Performance

The application should support 100 concurrents users. The response time should be a decent time

10. Quality

The application is based on a SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains.