

Adaptable Enterprise Software Architecture – A Case Study

Rizwan Ahmed PMP CISSP
IT Systems Architect

Index

- i. **Web Based Systems Architecture**
- ii. **Roadmap to J2EE**
- iii. **Summary of J2EE Technology Components**
- iv. **Deployment view of J2EE multi-tier architecture**
- v. **Roadmap to Web Services (WS)**
- vi. **Service Oriented Architecture**
- vii. **Summary of WS Technology Components**
- viii. **Deployment view of WS multi-tier architecture**
- ix. **Case Study: SCRS Forms Application Solutions Architecture**

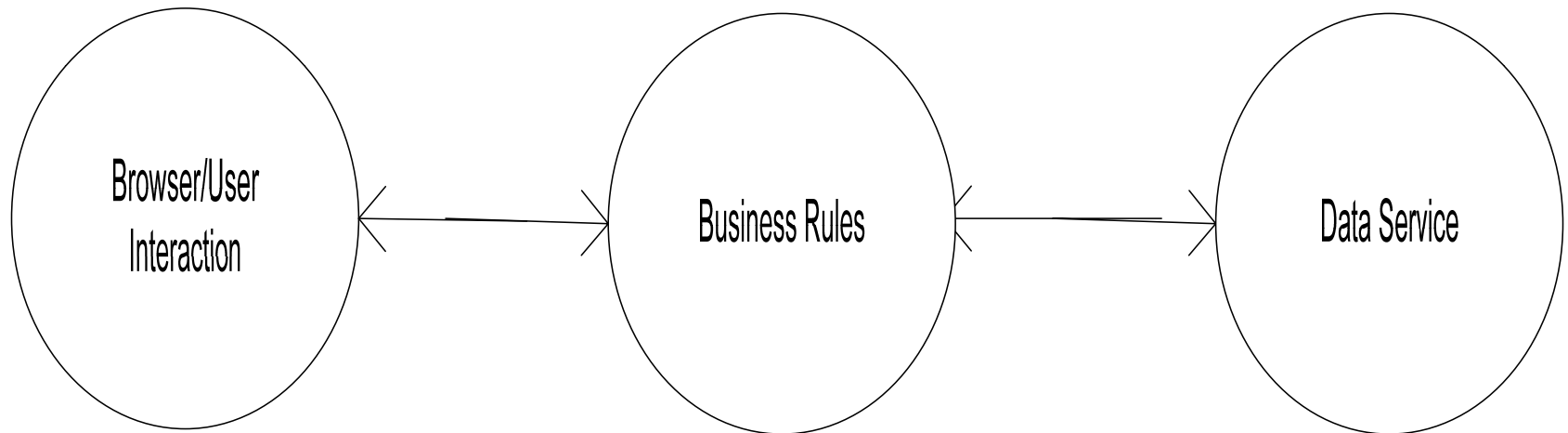
Introduction

- **To give a formal definition, software architecture is the structure of a system comprising system/software components, the externally visible properties of those components and the relationships among them**
- **Every software system has an Architecture**

Architectural Requirements for a Web Based System

- **High Performance**
- **High Availability**
- **Scalability**
- **Security**

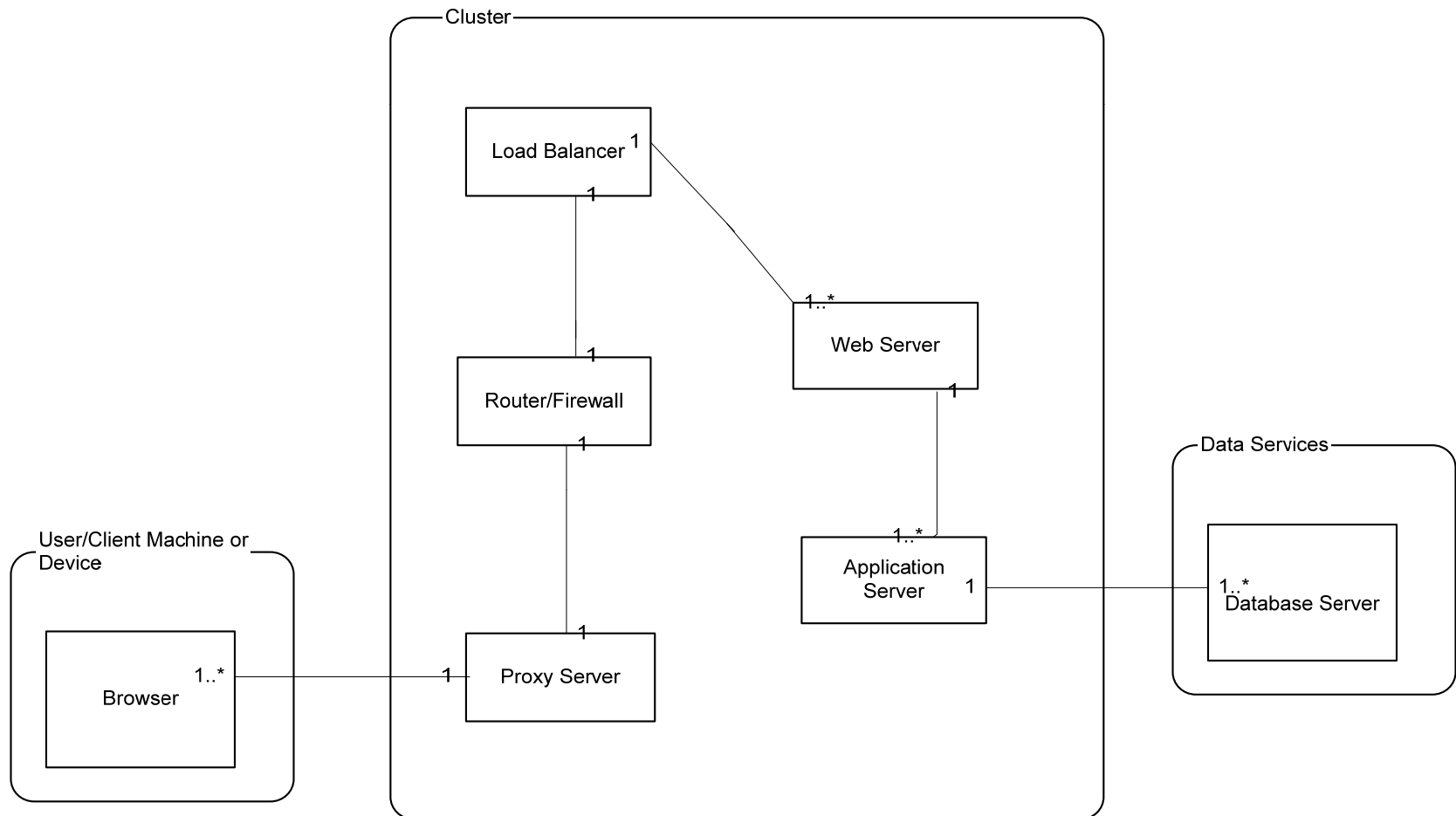
Web Based System Reference Architecture



This is referred to as an n -tier architecture (here, $n=3$).

A tier is a partitioning of functionality that may be allocated to a separate physical machine

Typical Web Based System Architecture



Web Browsers for Modifiability

- An end user typically initiates a request for information by interacting with a HTTP or WAP enabled browser
- Many technologies for creating sophisticated UI's (HTML, XML, Flash, ActiveX, Java Applets, AJAX etc.)
- HTTPS (HTTP-Secure) is typically used for secure transmissions. Uses SSL as a sub-protocol beneath HTTP and 128 bit public key cryptography to encrypt communications

Proxy Servers for Performance

- **Requests from browsers first arrive at a proxy server which exists to improve the performance of the web based system**
- **These servers cache frequently accessed web pages so that users may retrieve them without having to access the web site.**
- **Proxy servers are used by companies that want to restrict their employees access to certain areas of the system so in some cases it acts like a firewall (filtering requests)**

Routers and Firewalls for Security

- Requests from the proxy server arrive at a router, located on the network that may include a firewall for security
- The router may implement *Network Address Translation* (NAT), which translates an externally visible IP address into an internal IP address
- The purpose of the firewall is to prevent unauthorized information flows or accesses from the outside world, an example of the “limit access” tactic.

Load Balancing for Performance, Scalability and Availability

- The job of the load balancer is to distribute the “load” – incoming HTTP and HTTPS requests among a pool of computers running web servers
- In choosing which server to redirect the traffic to, the load balancer may select in a *round-robin* or *load characteristics* of each server to which it is connected
- load balancer supports horizontal scaling (adding more instances of a given resource)

Web Servers for Performance

- **Modern web servers are multi-threaded utilizing a pool of threads, each of which can be dispatched to handle an incoming request**
- **A multi-threaded server is less susceptible to bottlenecks when a number of long running HTTP or HTTPS requests (such as credit card validations) arrive because other threads in the pool are still available**
- **Concept of Vertical scaling achievable**

Application Servers for Performance and Scalability

- **These servers implement business logic and connectivity with the EIS tier**
- **Application servers typically support the following models: COM (or more recently .NET), CORBA, J2EE and Web Services**
- **enable transaction processing, queuing, transactional integrity and workload balancing**

Databases for Performance, Scalability and Availability

- Finally, the request for service arrives at the database, where it is converted into an instruction to add/modify/retrieve information
- Modern databases use internal replication or cluster technologies for performance, scalability and availability
- Caching for higher performance

Roadmap to J2EE

- Having web based systems open to the internet brings new pressures to bear on applications that are not experienced by traditional networked IS
- *Sun Microsystems*, in developing J2EE aimed to provide a basis for technology that supports construction of such systems
- J2EE provides transaction and state management, multi-threading and resource pooling

Summary of J2EE Technology Components

Enterprise Java Beans – API that allows developers to create, deploy and manage enterprise strength server side component based applications

Java Server Pages – Provides a method for creating dynamic HTML/XML content in response to a web client request

Java Servlet – Provides a mechanism for extending the functionality of a web server

Java Messaging Service – Provides J2EE applications with support for asynchronous messaging

Summary of J2EE Technology Components and Services (contd.)

Java Naming and Directory Interface – J2EE's directory service allows Java client and Web-tier servlets to retrieve references to user-defined objects such as EJB's and other environment entries

Java Transaction Service – Enables EJB's and their clients to participate in transactions

Java Connector Architecture – Defines a standard architecture for connecting the J2EE platform to heterogeneous Enterprise Information Systems

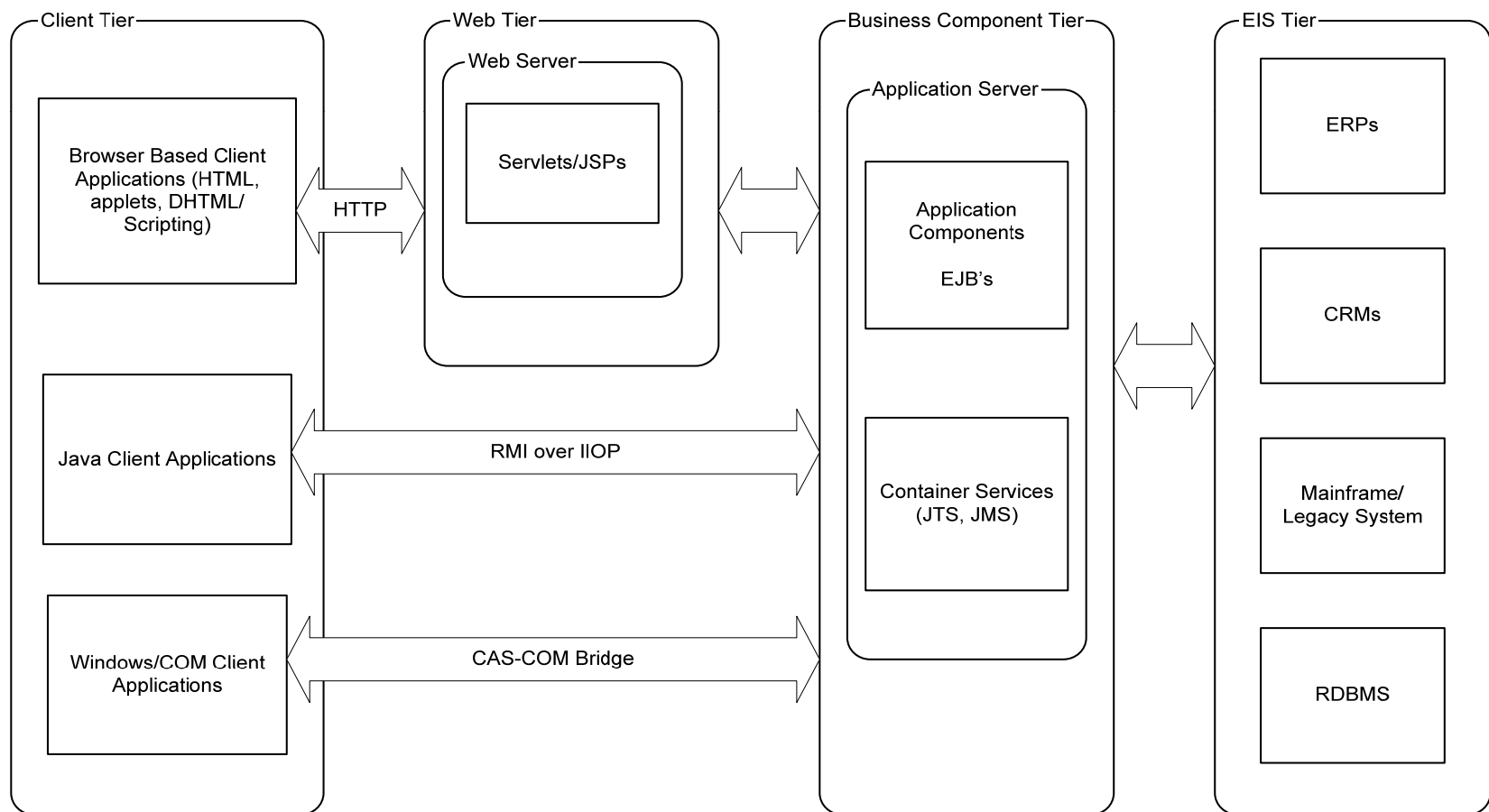
Summary of J2EE Technology Components and Services (contd.)

CAS-COM Bridge – Allows integration between COM and J2EE applications across the network. Allows access to J2EE server side components by COM enabled client applications

RMI over IIOP – Provides developers with an implementation of Java RMI API over OMG's industry standard Internet Inter-ORB protocol (IIOP)

Java Database Connectivity – Provides developers with uniform interface to a wide range of relational databases

Deployment view of the J2EE multi-tier Architecture



Roadmap to Web Services – SOA

- **The Service Oriented Architecture defines business applications as reusable, easily changed components of business functionality**
- **Discrete business functions or processes are created as independent components with standard interfaces which can be accessed by other applications**
- **Service can be flexibly combined and architecture supports creation of composite applications assembled from new or existing services**

Roadmap to Web Services – *Benefits of SOA*

- **Enable business agility**
- **Provide higher return on investment**
- **Enable IT agility**
- **Create better modular application designs**
- **Support multiple client types and platforms – *based on industry accepted open standards***
- **Speed development time**

Service Specification

1. Service Definition

Each service should be described in terms of its functions and systems used to create the service

Customer Verification	Check Customer	Verify if customer exists	CRM
Credit check	Check Credit	Interface to online credit check provider	Financials

SOA – Service Specification (contd.)

2. Service Interface

Service interface provides the information necessary for creating Web Services or other application or component interfaces.

Service:	Customer Verification
Input:	Customer Id; name, ssn,
Outputs:	Customer Id; address; shipping information; email; preferences
Methods:	verifyCustomer(custId, name, ssn)
Implementation:	SOAP based web service. Backend implementation using Java classes talking to any EIS database

Web Services Basics

- **A Web service is an implementation of the Services specification independent of language, development tools and platform**
- **Based on industry accepted open standards that use internet protocols as the methodology to transport data**
- **They work with .NET, J2EE and CORBA**

Summary of Web Services Technology Components

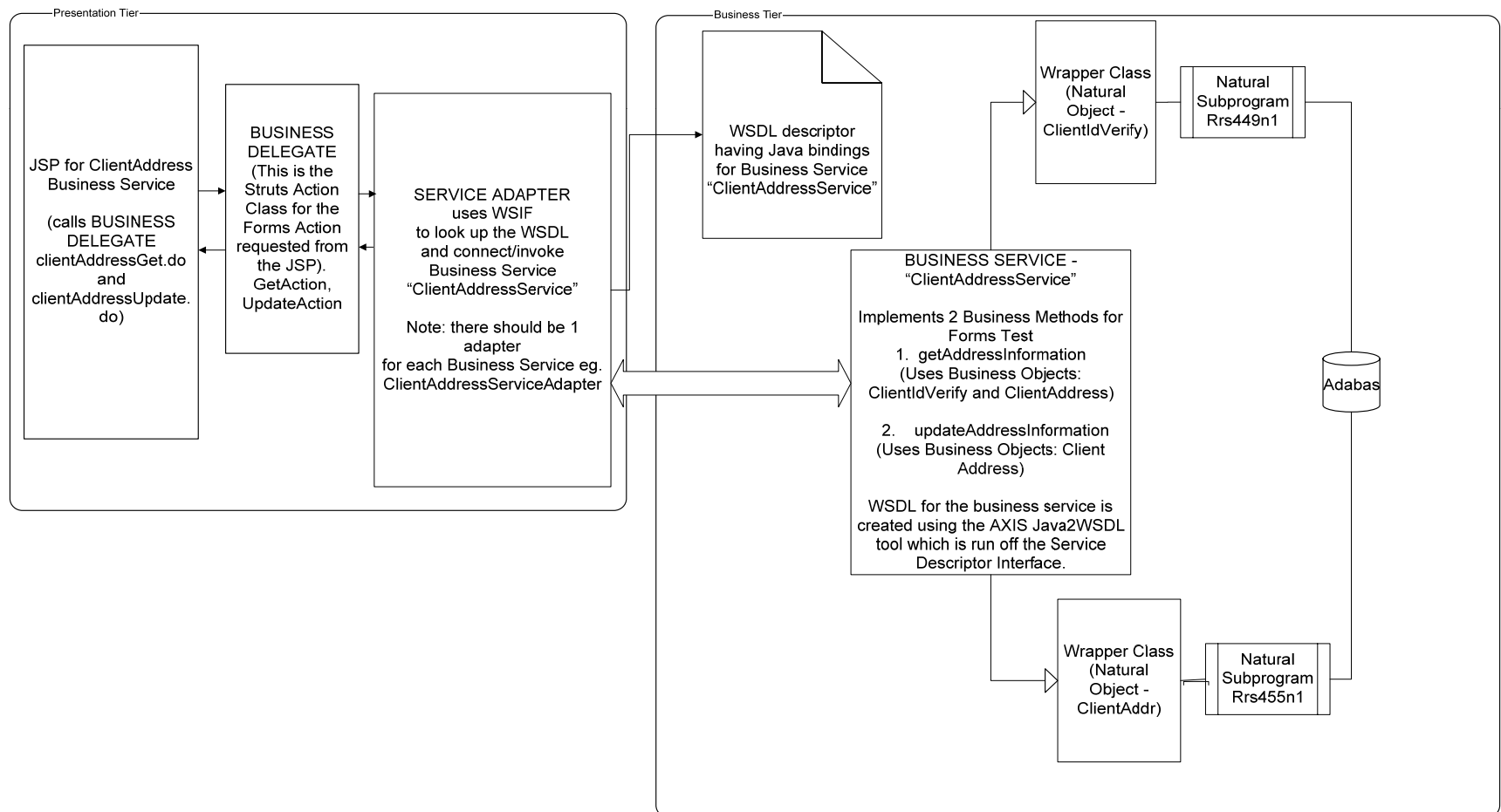
XML – Data representation in an independent fashion

Simple Object Access Protocol – Protocol specification for exchanging structured information in the implementation of web services

Universal Description Discovery and Integration – provides a means of locating services via a common registry of providers and services

Web Services Definition Language – XML based language that provides a model for describing web services

Deployment view of a typical Web Services multi-tier Architecture

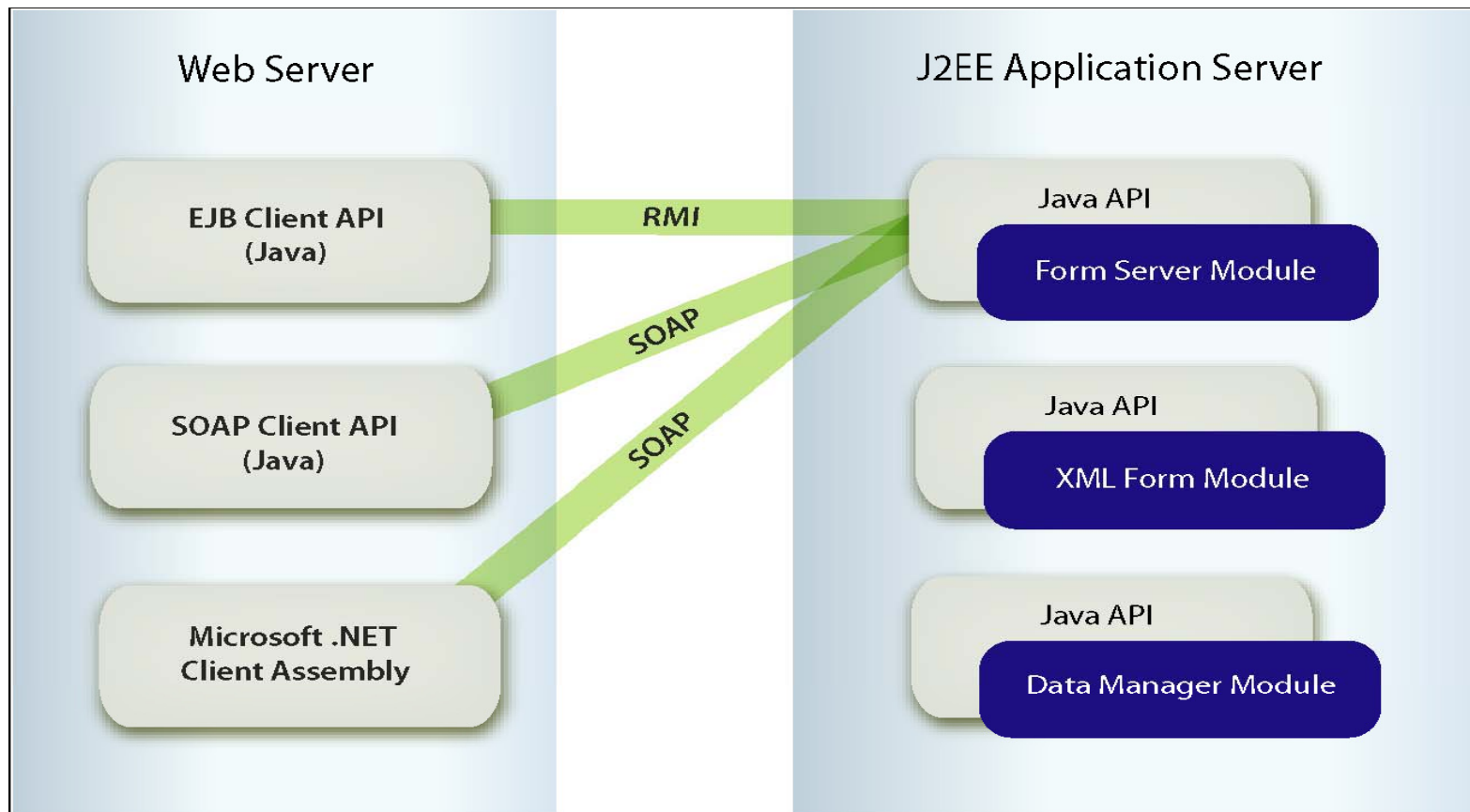


SCRS Enterprise Forms Application

The architecture was developed to:

- Leverage a J2EE Java and XML based API framework supported by the latest release of Adobe LiveCycle Forms Server
- Provide standardization across in-house developed web products by using the same hosting platform leveraging the popular, powerful and open source J2EE Application Server JBOSS
- Further standardization leveraging the popular open source Apache Struts MVC (Model View Controller) framework leading to easy usability, operability and extensibility.

Adobe LiveCycle Forms



Components Overview

Forms Server Module (FSM) API

- Forms Server Module was used to create the interactive data capture application
- FSM validates, processes, transforms and delivers forms created in LiveCycle Designer
- Form author can design a single form design that FSM can process and/or render in HTML or PDF formats
- When the Form Server Module receives a request for a form, it uses a set of transformations to merge data with a form design and then delivers the form in a format that best matches the presentation and form filling capabilities of the target browser

Components Overview

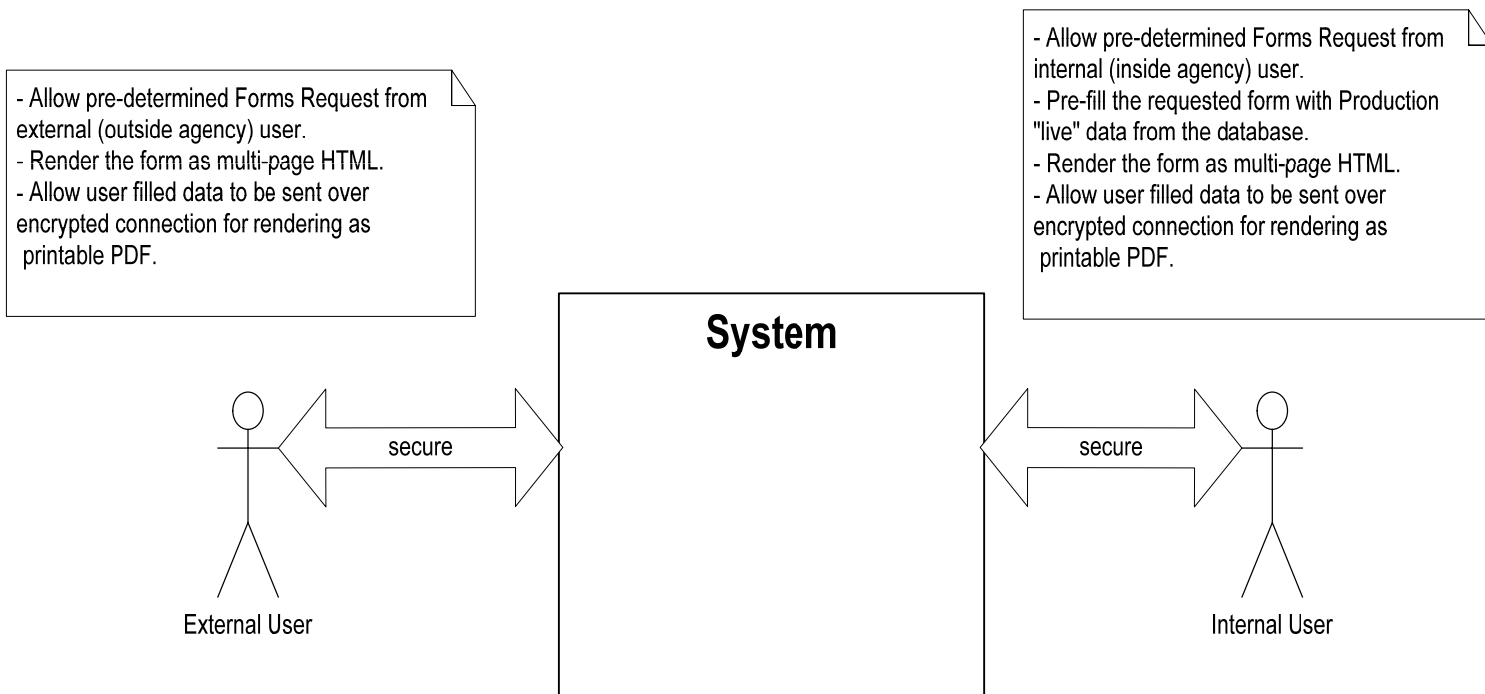
Forms Server Module (FSM) API (..Continued)

- Provide server-side execution that is in the form design. The Forms Server Module executes the intelligence (validation, calculations) that is there in the form design (written either in JavaScript or FormsCalc) and returns the results onto the HTML content to the browser
- Dynamically generates a PDF or an HTML document of the form design with or without data. An HTML form can deliver multi-page forms page by page. In contrast, a PDF form delivers all the pages at once
- Enables a non-technical user to amend a form design by using LiveCycle Designer to meet ongoing business requirements. In contrast, a web application that displays HTML pages may require a user to modify HTML or XML source code to make changes to a web page.

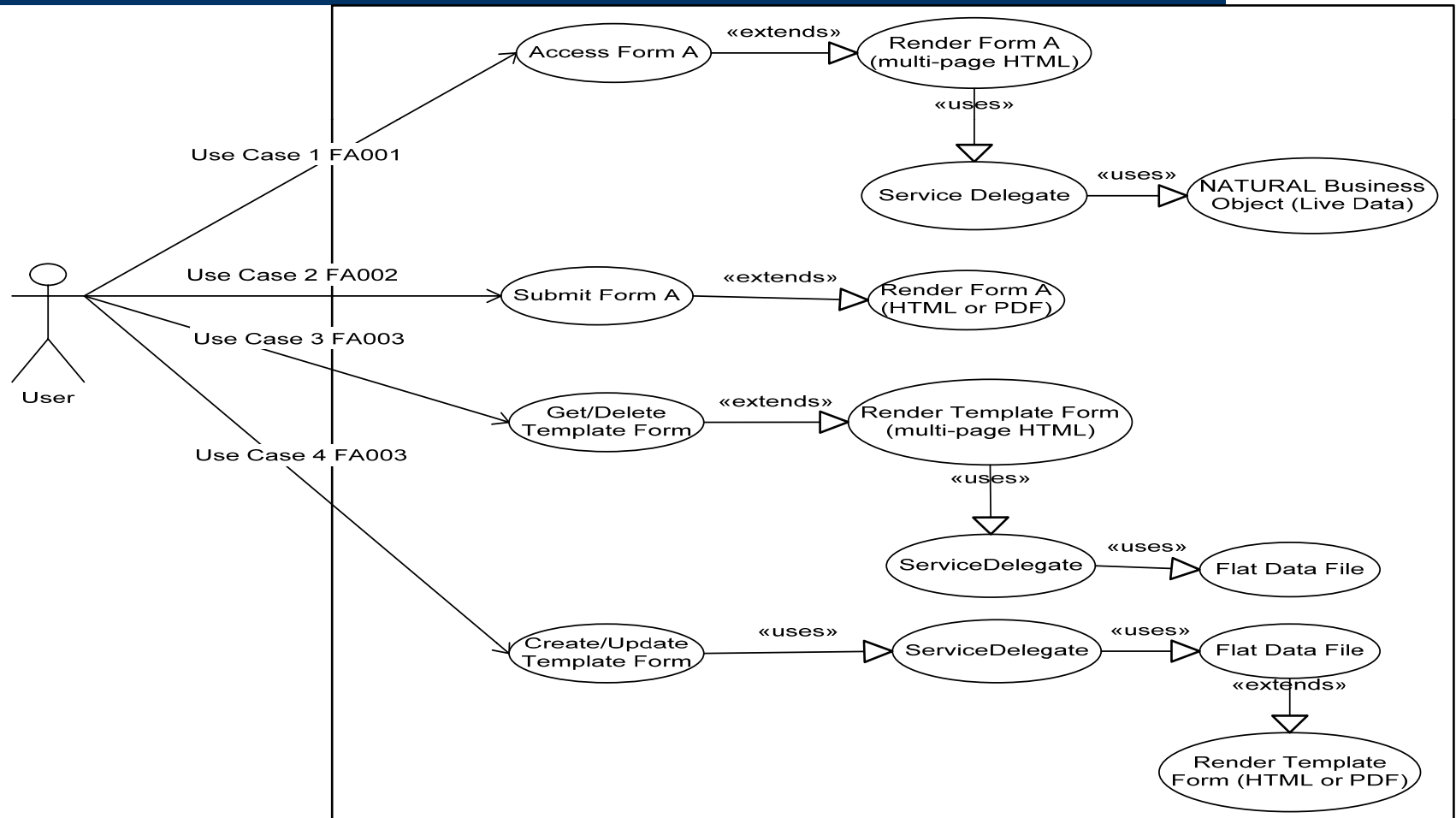
Security Model

- Due to the sensitive nature of data being transmitted (client and demographics information including Social Security Numbers etc.), SSL was used to allow Forms users the ability to communicate securely with an 128 bit public/private key pair encrypted session
- Short for Secure Sockets Layer, SSL provides a level of security and privacy for those wishing to conduct secure transactions over the internet. The SSL protocol works at the Transport Layer 4 of the OSI and adds a layer of encryption upon HTTP
- Port 443 and the SSL Connector has to be configured on the web server

High level Use Case View



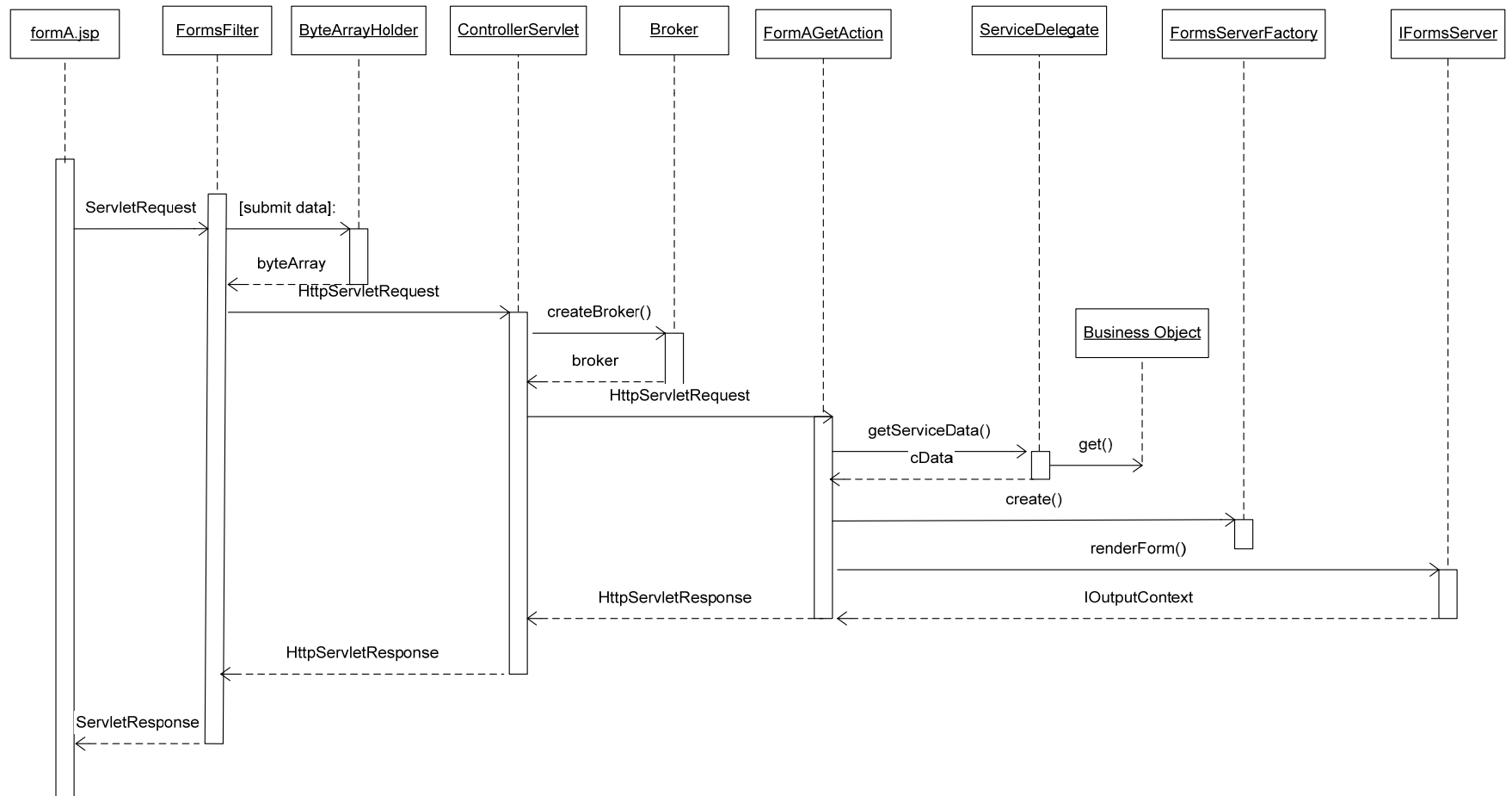
Use Case Realization



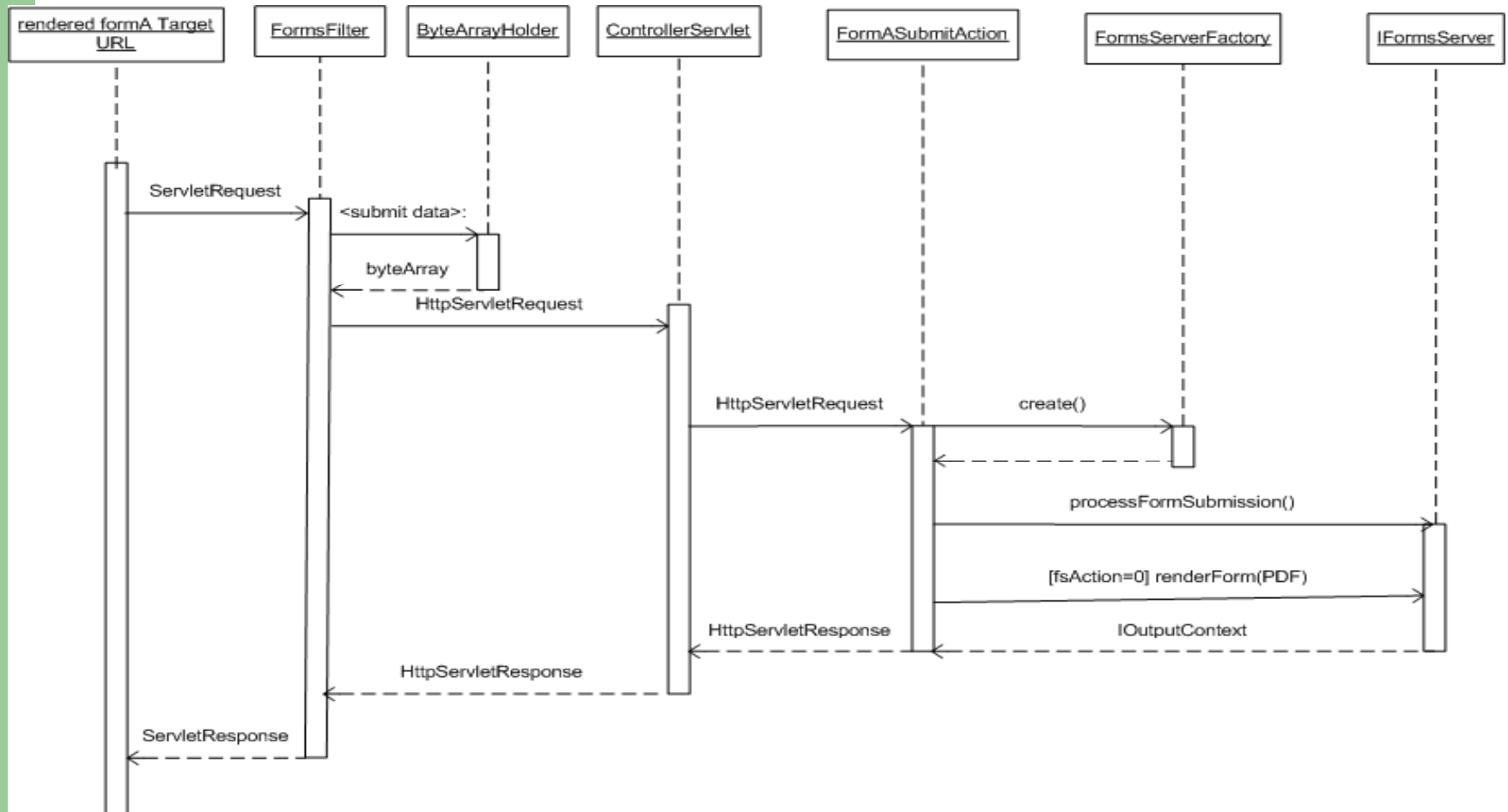
Process View

This section provides a description of the process view of the architecture. Diagrams the tasks (processes and threads) and related activity timelines involved in the system's execution and their interactions.

Use Case FA001 (Sequence Diagram)



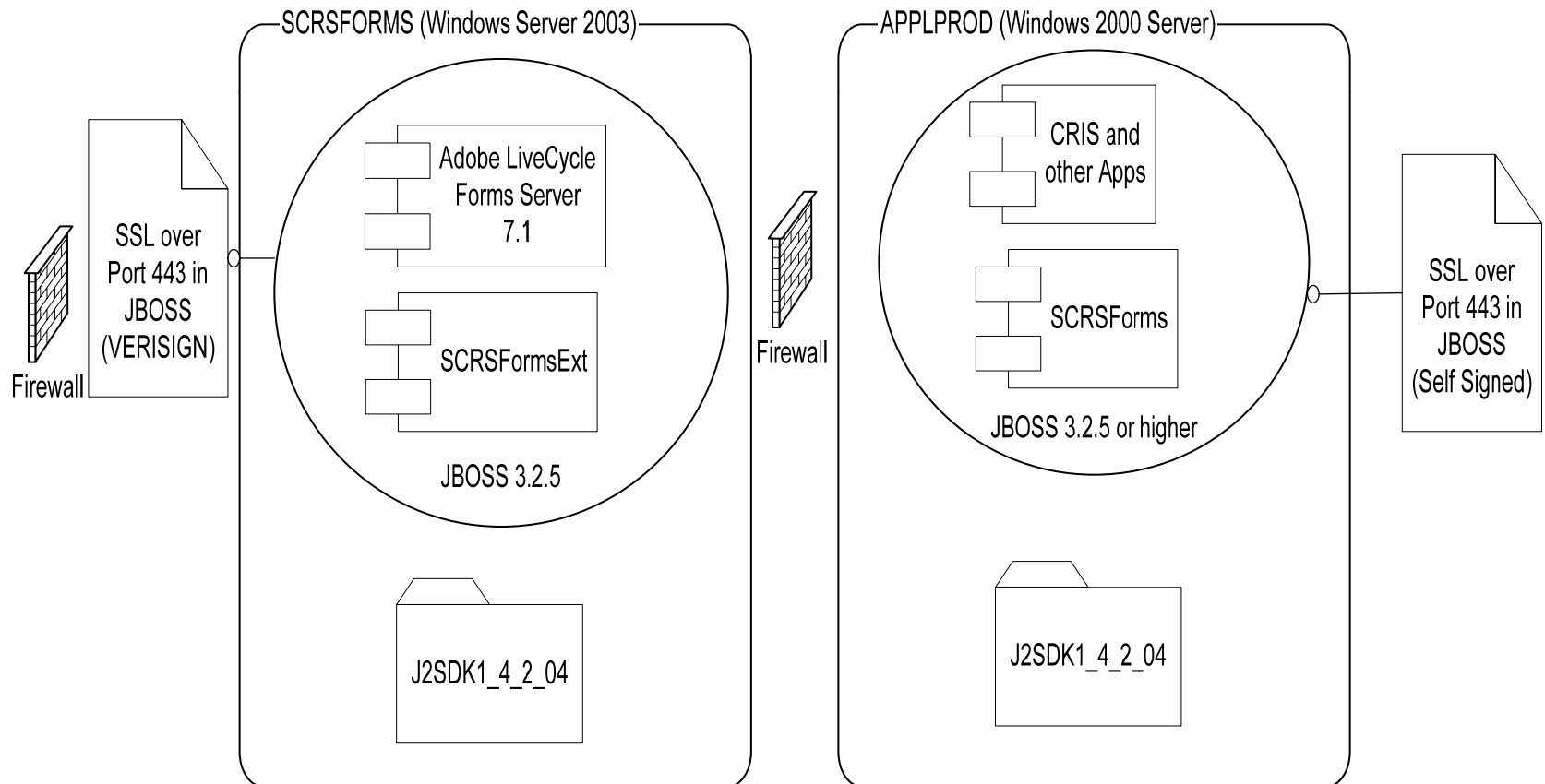
Use Case FA002 (Sequence Diagram)



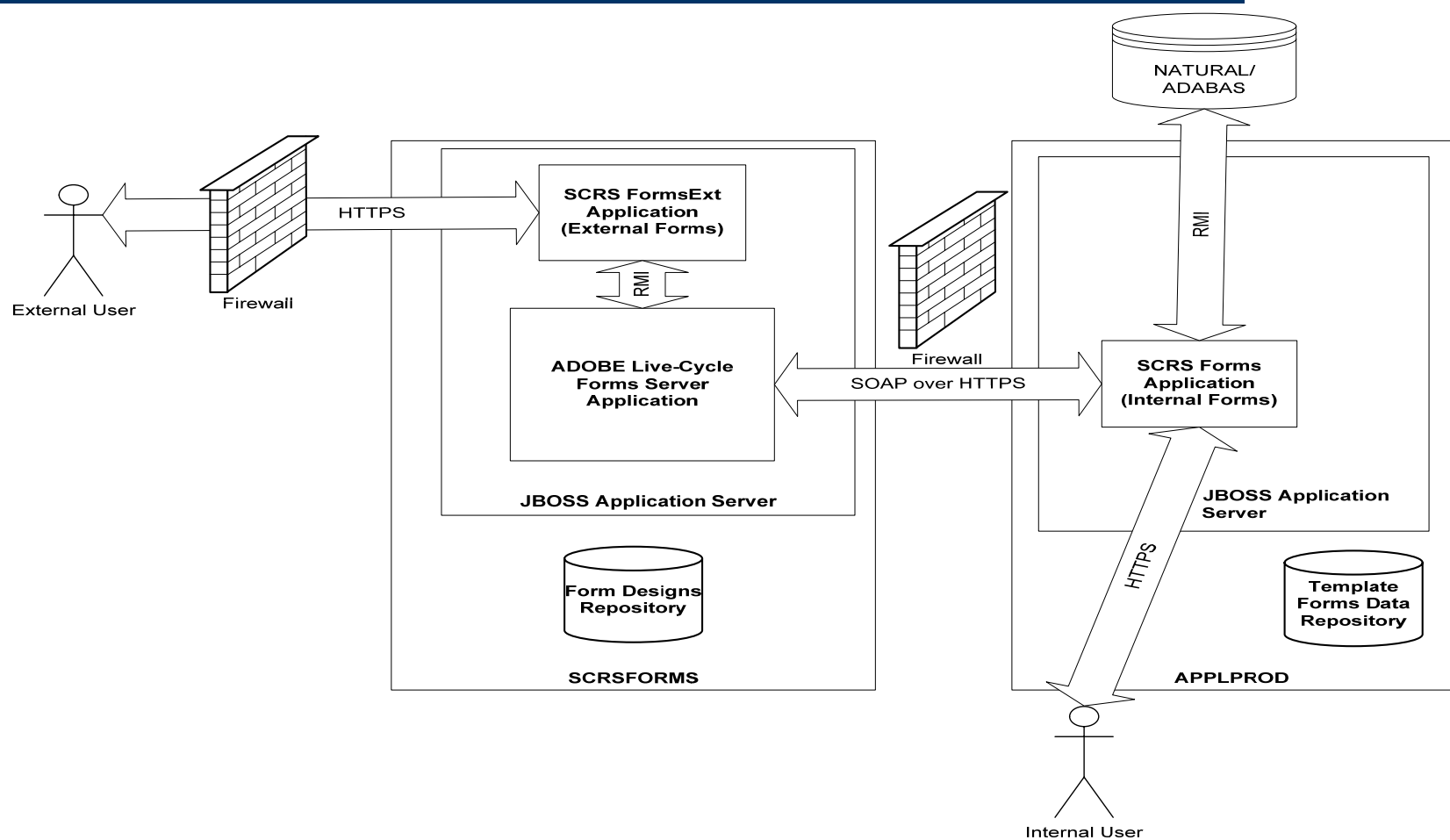
Deployment View

This section provides a description of the deployment view of the architecture and physical network configuration followed by a mapping of processes to each processor.

Deployment View



Implementation View



Questions/Comments?

