

# **Architecting and Designing Enterprise Applications**

**Presented By,  
M.Sangeetha,  
ap(srg),  
kec**

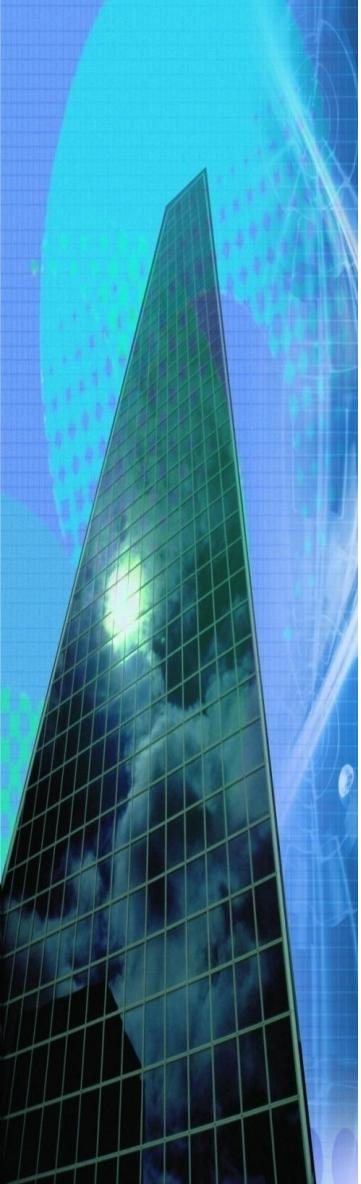


## Architecture, Views and Viewpoints

“ The fundamental organization of the system embodied in its components, their relationship to each other and to the environment and the principles guiding its design and evolutions” - ANSI/IEEE Std 1471 .

- Architecture serves as the blueprint for guiding the design of an application or a product.
- Architecture description is the concrete artifact to depict the architecture and is organized in the form of ‘views’ and ‘viewpoints’.
  - Viewpoint in general represent stakeholder’s view of the enterprise application.

Contd...  
..



## Architecture, Views and Viewpoints (Contd...)

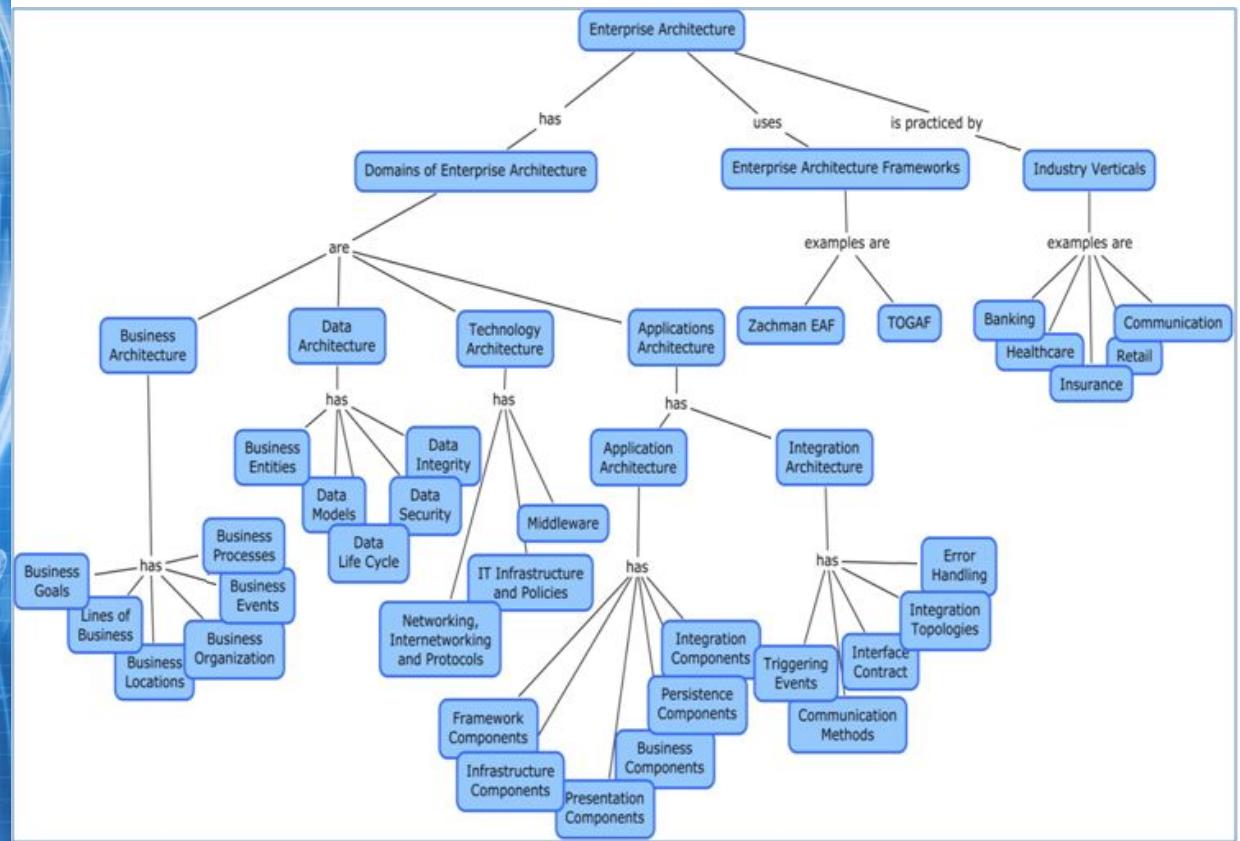
- View defines how the enterprise application looks like from a selected view point.
- The architecture description of an enterprise application consists of various aspects. These aspects are depicted by various views and, in turn, each view is represented as one or more appropriate models using languages like UML.



## Enterprise Architecture

- An Enterprise Application consists of data, users and business applications which interact with each other.
- Enterprise architecture helps to conceptualize, represent, understand and analyze these business scenarios, keeping business applications, user and data under consideration.

# Enterprise Architecture- an Overview



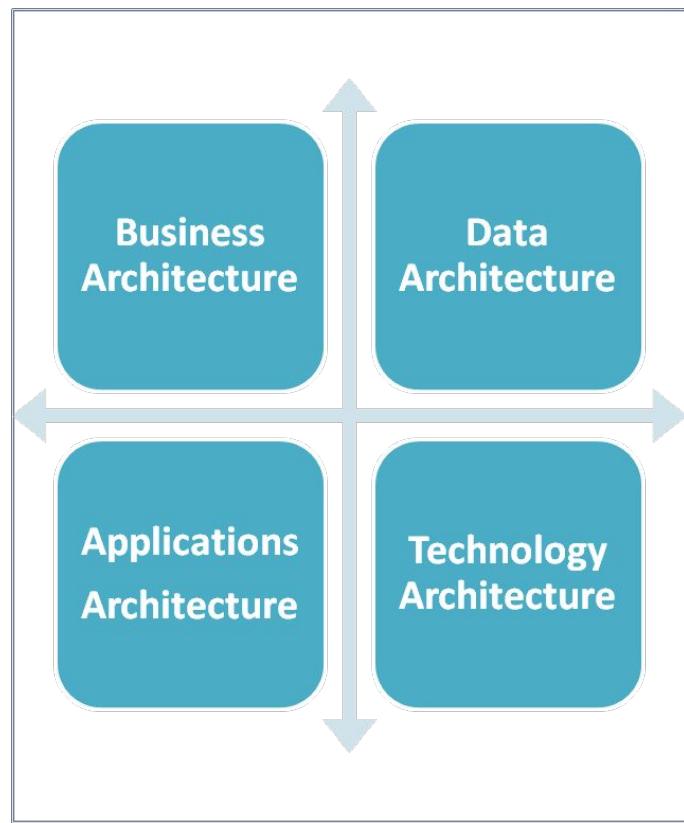


## Enterprise Architecture Framework

- An enterprise architecture framework specifies a set of views and viewpoints and comprises of principles, standards, guidelines, approaches, services, design concepts and models to facilitate the development of enterprise architecture.
- TOGAF( The Open Group Architecture Framework) is one of the most popular enterprise architecture framework.

Contd...  
..

## Enterprise Architecture Framework (Contd...)





## TOGAF Architecture Domains

- **Business architecture:** focuses on the overall strategy and governance of the business processes and other key elements of the business such as business goals, lines of business, business locations, business organization, and business events.
- **Applications architecture:** is the structure of interconnection and interactions of all the applications in an enterprise and maps to the underlying enterprise business processes.

Contd...  
..



## TOGAF Architecture Domains (Contd...)

- **Data architecture:** focuses on the overall data assets – both logical and physical that an organization possesses. It defines how the data is being organized in the application, Data models, data life cycle management and security of data. Data architecture is sometimes referred to as information architecture.
- **Technology architecture:** consists of building blocks which include the middleware components used to glue the applications, IT infrastructure hardware and software and networking and communication protocols.



## Blueprint of an Enterprise Application

- Enterprise architecture helps understanding how an application fits within the existing enterprise landscape, and guides arriving at the blueprint of an application.
- The blueprint of an enterprise application consists of the following viewpoints or perspectives:
  - Logical architecture
  - Technical architecture
  - Data architecture
  - Infrastructure architecture



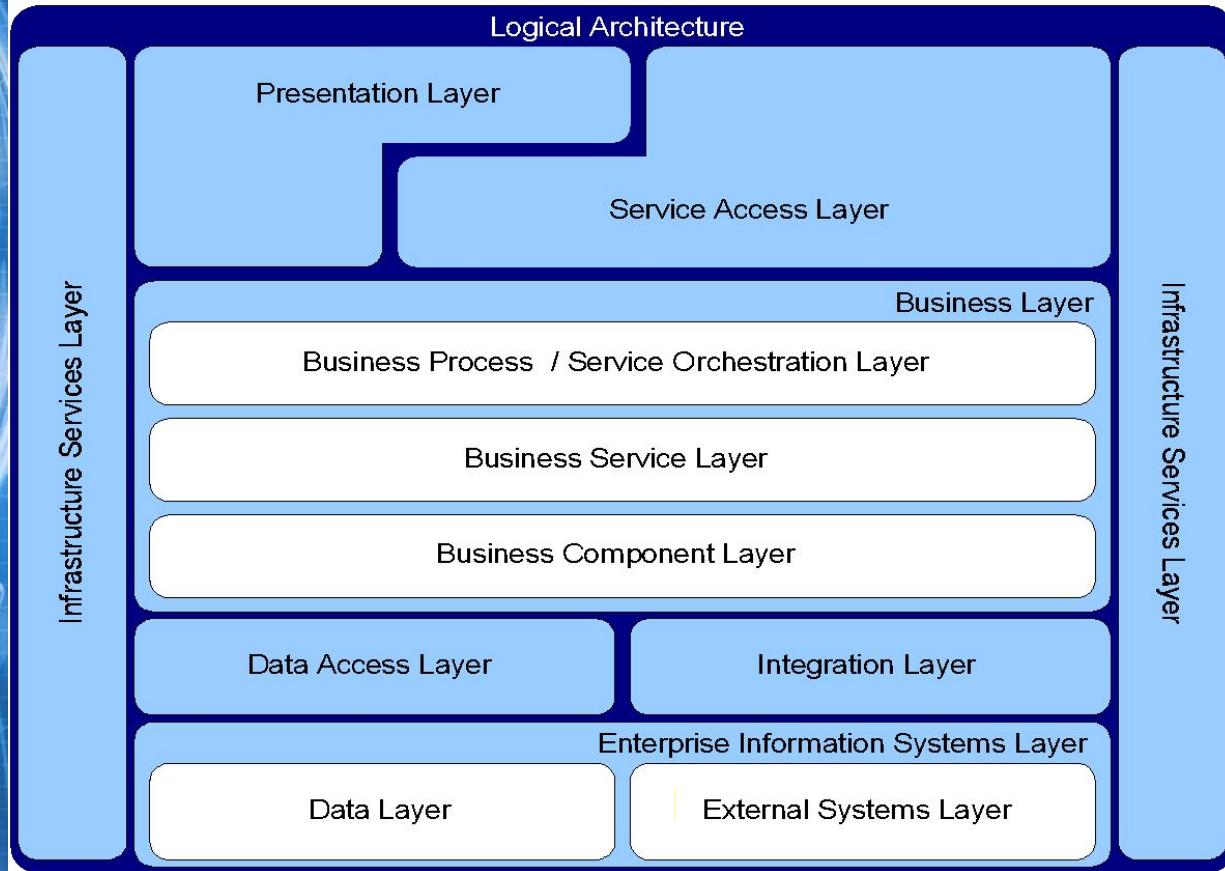
## Logical Architecture

- Elements of business architecture define the **functionality** of a given enterprise application.
- This functionality can be mapped to a set of **logical building blocks** which implement the enterprise application.
- These logical building blocks together define the **logical architecture** of an enterprise application.
- The logical architecture serves as the blueprint to the architects to arrive at an optimal enterprise application solution.

“Raising Enterprise Applications: A Software Engineering Perspective”

by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.

# Logical Architecture – an Overview



“Raising Enterprise Applications: A Software Engineering Perspective”

by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.



## Logical Architecture – Layers

- **Business Layer:** Business layer can be considered to comprise of three sub layers to represent business processes, business services and business components.
  - Business Process represents the business processes implemented in the application.
  - Business processes, in turn, are composed from one or more business services and are represented in the Business Service Layer.
  - Business Component Layer represents the lowest level of abstraction of business layer in terms of underlying business entities.

**Contd...**

•  
“Raising Enterprise Applications: A Software Engineering Perspective”  
by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.



## Logical Architecture – Layers (Contd...)

### □ Enterprise Information Systems (EIS) Layer

- The enterprise application has to connect with the database or other storage mechanisms to manage persistence of the application's data across invocations, this is done by Data Layer.
- External Systems Layer is used to represent the external applications with which the target application has to integrate with.
- These two layers are often combined together and are referred to as the (EIS) Layer.

**Contd...**

• “Raising Enterprise Applications: A Software Engineering Perspective”

by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.



## Logical Architecture – Layers (Contd...)

- **Data Access Layer** and the **Integration Layer** provide the abstractions that hide the physical details of storage and access mechanisms from the business layer.
- **Presentation Layer** is used by the human users.
- **Service Access Layer** is used by the external systems to access the business processes or business services hosted by the enterprise application.
- **Infrastructure Services Layer** provides the reusable and general purpose components like logging, caching, auditing etc.

“Raising Enterprise Applications: A Software Engineering Perspective”  
by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.



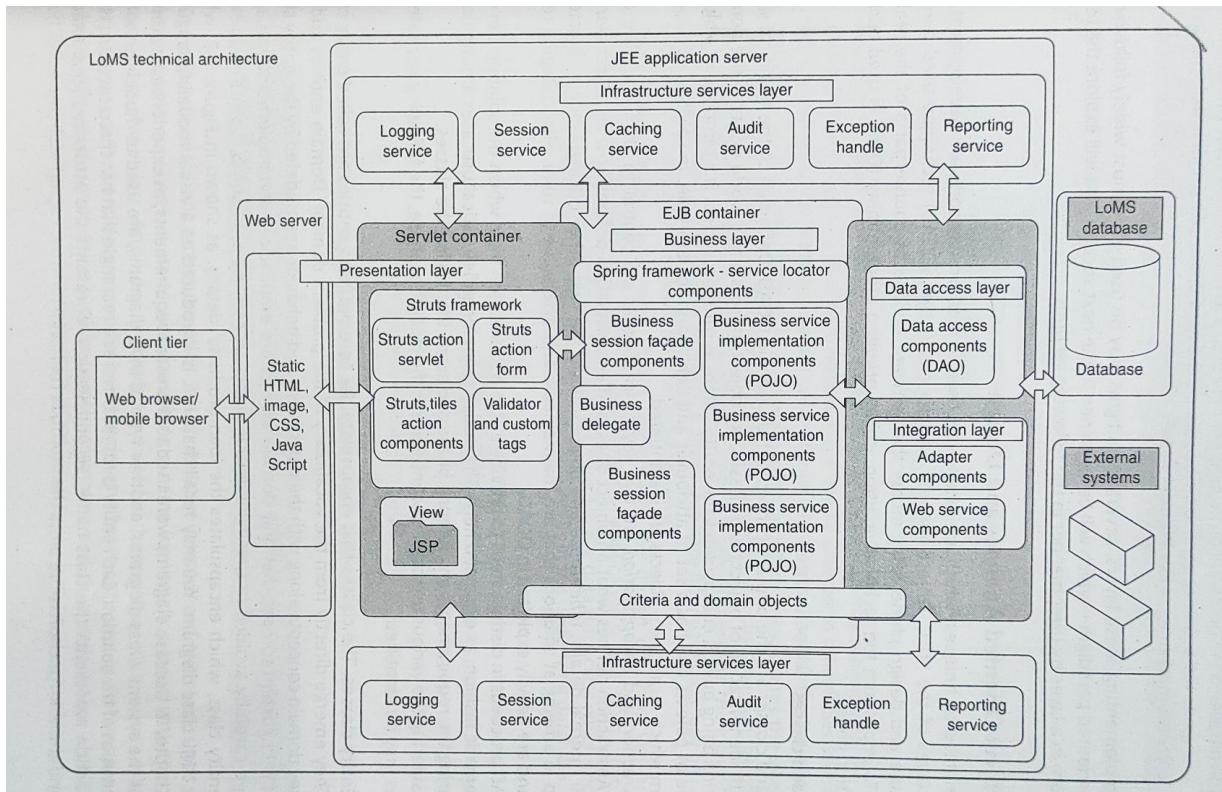
## Technical Architecture and Design

- The technical architecture serves as the starting point of the implementation and consists of application frameworks, design patterns, APIs and programming languages.
- Appropriate mechanisms for integration of the applications are also identified as part of technical architecture to achieve the integration of enterprise applications within the applications' landscape.
- Identifies technical layers and components.
- The software architecture patterns are widely used in the definition of the technical architecture.

“Raising Enterprise Applications: A Software Engineering Perspective”

by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.

# LoMS Technical architecture





## OOAD – an Introduction

- For implementing these layers several paradigms may be used -the most widely followed today is object oriented paradigm.
- Object oriented analysis and design (OOAD) techniques are used during the architecture and design phase helps model the software entities as counterparts of the real world objects.
- An object mirrors a real world entity in the domain of the solution being built. It encapsulates the entity's state at runtime.

**Contd...**

•  
“Raising Enterprise Applications: A Software Engineering Perspective”  
by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil Nallasamy, Veerakumar E  
Copyright © 2010 Wiley India Pvt. Ltd. All rights reserved.



## OOAD – an Introduction (Contd...)

- A class is a template or a blueprint for a type of objects and has a set of attributes and behaviors.
- Some of the other key features of OOAD are inheritance, polymorphism, encapsulation and abstraction.
- Abstraction
  - What an object does
- Encapsulation
  - How an object works and hides the internal details
- Considering the various use cases documented in inception phase, the following categories of classes are identified
  - Entity classes
  - Boundary classes
  - Data store classes



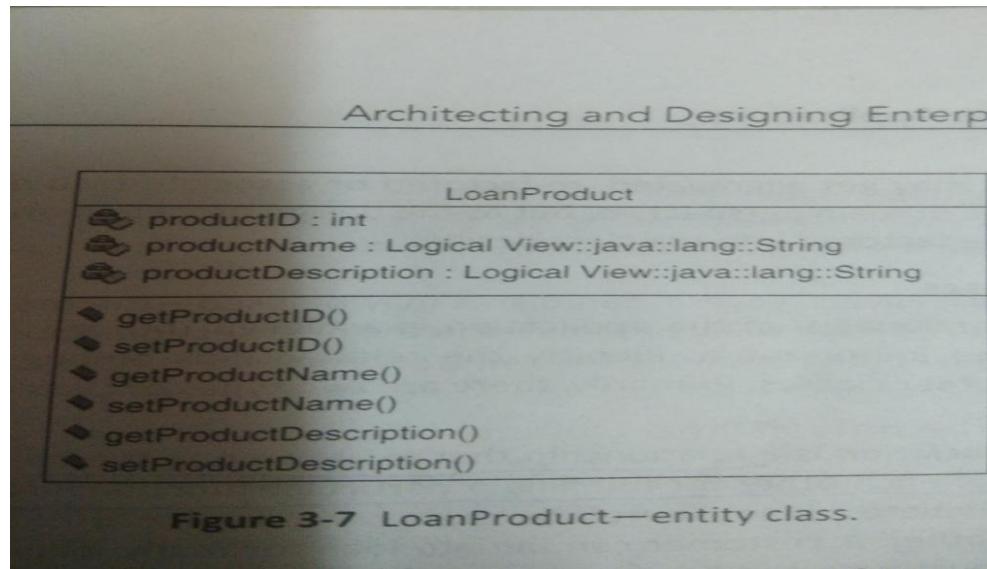
## Types of classes

- Entity classes
  - The classes that encapsulate the core business entities
  - They emerge directly from the vocabulary of the problem domain
  - Eg: In LoMS, LoanProduct is an entity class which encapsulated the loan product details
  - Like **use case diagrams** are used to represent **requirements**, **class diagrams** are used to represent **entities**



## Types of classes

- Entity classes
  - Each entity is represented as a rectangle that has three sections
    - First section represents class name
    - Second section represents attributes
    - Third section represents methods





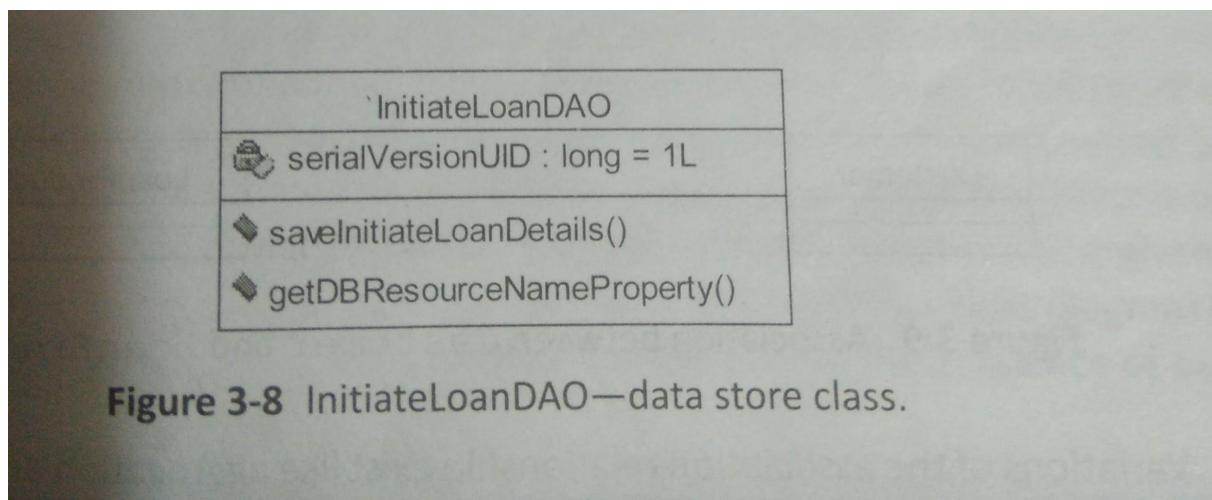
## Boundary classes

- The classes that encapsulate all aspects of interactions with the actors
- Created to model user interface interactions
- Eg: Elements of data entry forms are encapsulated as boundary classes
- In LoMS, InitiationForm.jsp is a boundary class, which encapsulates the interaction of 'Bank Representative' actor who initiates the loan on behalf of a customer



## Data Store Classes

- Encapsulate the design decisions like how enterprise applications store and retrieve data from the application's data stores
- Eg: In LoMS, `InitiateLoanDAO` is a data store class which encapsulates the logic of accessing and persisting the customer details in the underlying database.

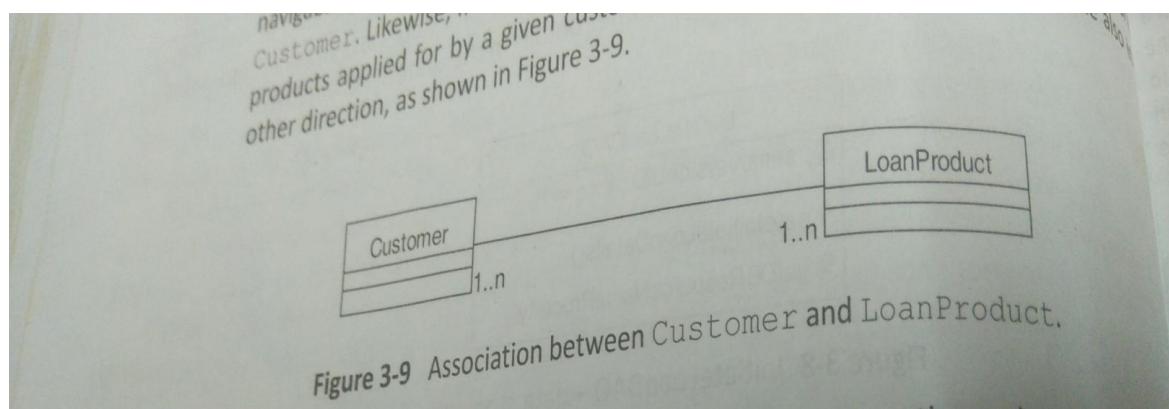


**Figure 3-8** `InitiateLoanDAO`—data store class.

# Relationships among classes

## □ Association

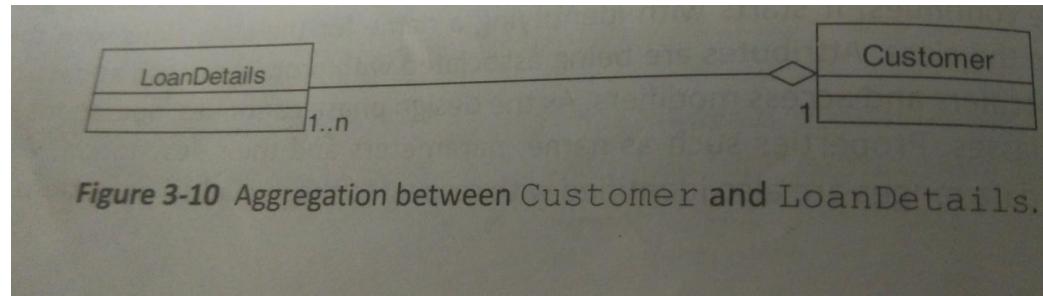
- A relationship that exists among the objects of the classes where they are aware of each other for fulfilling a particular functional requirements
- Eg: Association relationship between Customer and LoanProduct classes





## Relationships among classes(Contd.)

- Variations of Association relationship
  - Aggregation
    - Represents “whole-part” relationship between the objects of the classes

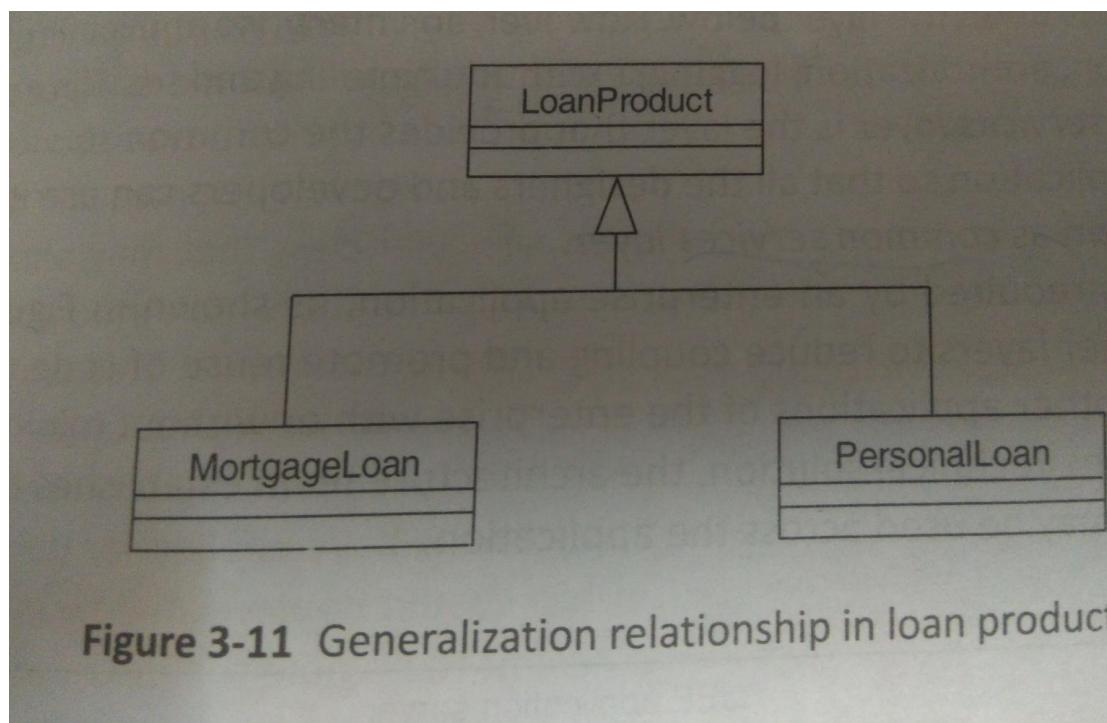


- Composition
  - Represents a complete “whole-part” relationship between classes

## Relationships among classes(Contd.)

### □ Generalization

- Represents “is a” or “a kind of” relationship among classes
- Also referred to as inheritance





## UML DIAGRAMS

- Class diagrams
  - To represent a partial view of the system
  - Model the static aspects of the application
- sequence diagrams
  - To visualize the sequence of actions
  - To convey the sequence of message passing or method calls among classes in a given use case
  - To capture the dynamic view
- Collaboration diagrams
  - Depicts how classes collaborate rather than the sequence of actions
- State chart diagram
  - To model various states that a particular object may go through during the execution of the application

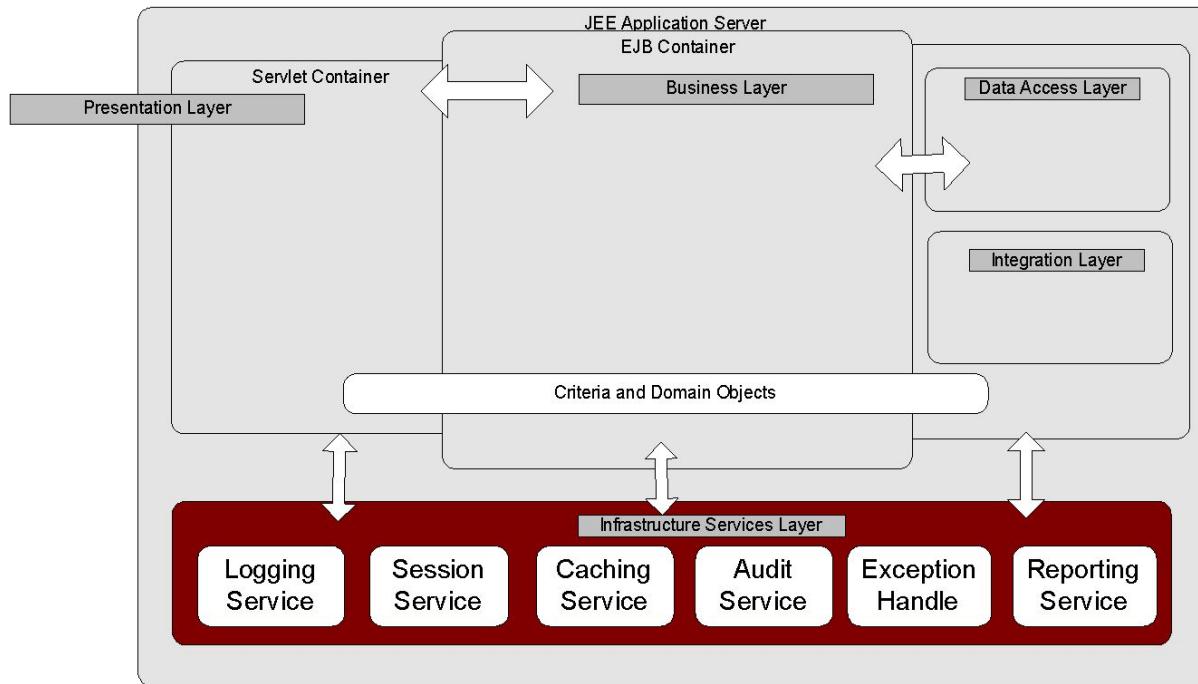


## Infrastructure Services Layer

- Infrastructure services layer is the layer that provides the **common services** for all the other layers in an enterprise application so that all the designers and developers can use it in a uniform manner.
- This layer is also known as **common services layer**.

**Contd...**

# Infrastructure Services Layer (Contd...)





## logging

- Need for auditing and debugging purposes
- To keep track of usage of various resources by the users of an enterprise application
- Eg: InitiateLoan use case logs events like getting the preapproved loan amount based on SSN, or saving the loan details after validation.
- LoMs uses Log4J as underlying logging mechanism



## Exception handling

- Exceptions occur when normal flow of program instructions is disrupted due to an event
- Exception handling mechanisms
  - User defined exceptions
  - Message display
  - Try-catch blocks
- In LoMS, InitiateLoan use case should provide a strategy to handle input of an invalid SSN number of a customer



## Session management

- Support for multiple concurrent users in an enterprise application
- Implemented to track session using a session ID which is a unique number associated with the session
- Effective session management
  - State management and performance
  - Clustering
    - Clustering of servers to provide redundancy for fault tolerance and load balancing
- In LoMS, session data is stored in primary memory and is created for each user when the user is successfully authenticated.
- It uses Jboss POJO cache solution



## caching

- Technique for improving the performance of an application
- It involves bringing frequently used data to the high speed primary memory from the secondary store where it is persisted so that the application can readily access it.
- Approaches
  - Pre-fetch
    - Pre-fetch the data at application startup and store it in the cache
  - Lazzy ladding
    - The data is retrieved from the data source at the time of the first request, and thereafter the data may remain stored in the cache
- In LoMS, Jboss caching is used

- AAA security requirements
  - Authentication
  - Authorization
  - Auditing
- Single sign-on (SSO) to eliminate the need for multiple authentications of the same user
- In LoMS, Java Open Single Sign-On (JOSSO) is used



## Presentation Layer

- Presentation layer is the **face of an enterprise** application from the perspective of an end user.
- It primarily dictates the **end user experience**, provides the content and data in a user friendly format and responds to end user generated events.
- Presentation layer acts like a bridge between the user and the enterprise application.
- It serves the static and dynamic content to the end user
- Static content is represented in HTML
- Dynamic content is generated by scripting, JSP in a typical model-view-controller (MVC) pattern



## Mvc architecture

- The MVC pattern deals with decoupling the business logic("model") from the user interface ("view")
- This pattern allows all interactions among view and model to be controlled by a central entity called "controller"
- MVC pattern defines the blueprint for presentation tier.
- The MVC pattern brings in the flexibility of keeping model data and view separate, making the entire presentation more flexible
- Struts has been the most popular framework for implementing the presentation layer

Contd...  
.

# Mvc architecture

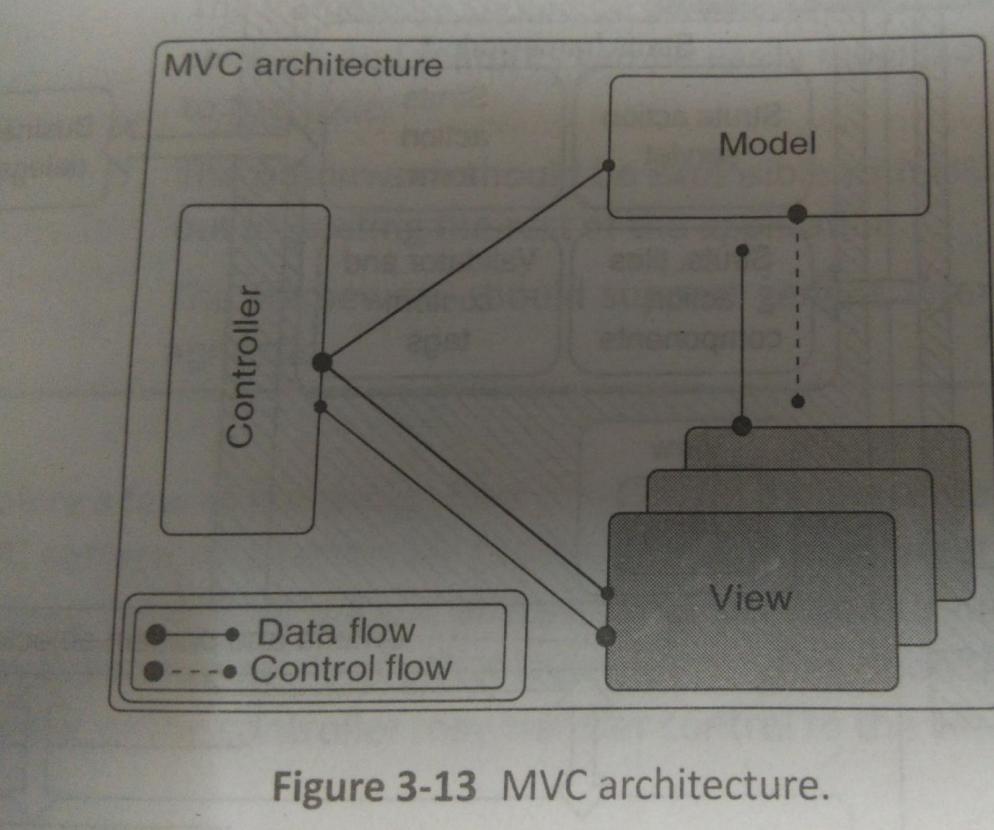
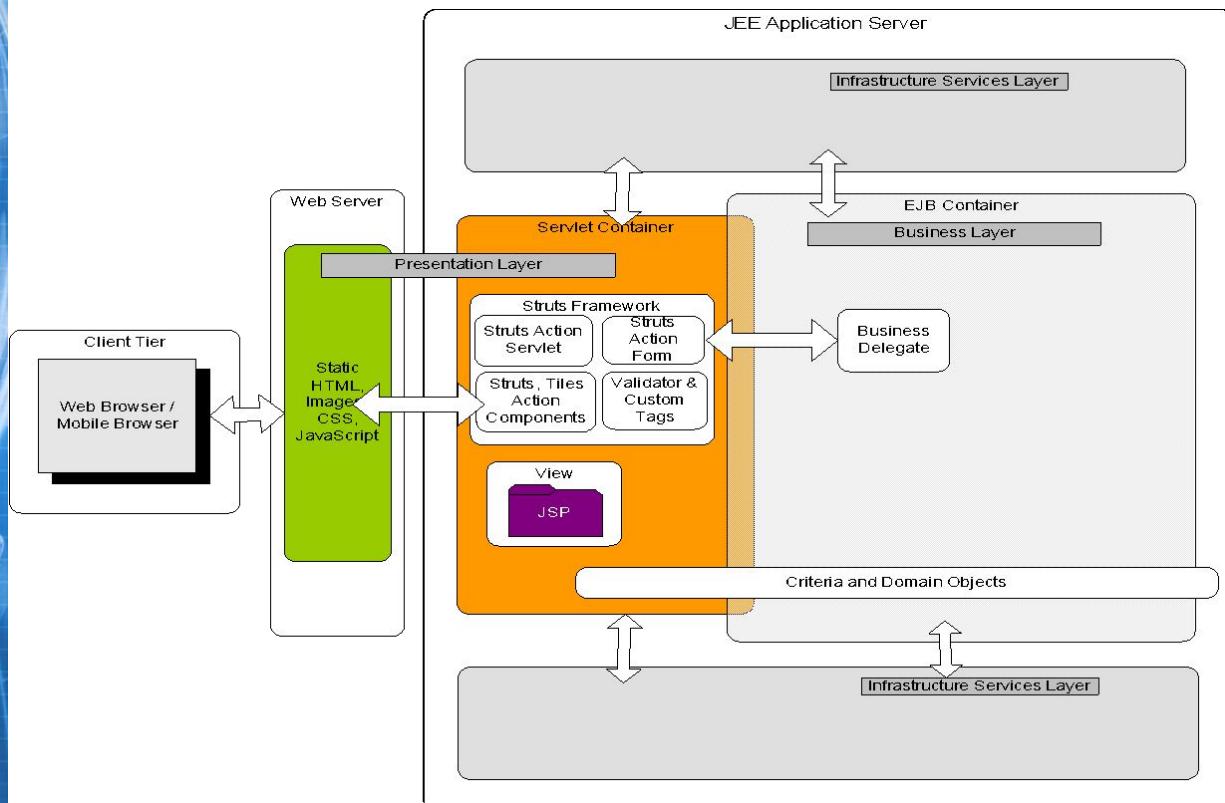


Figure 3-13 MVC architecture.

# Presentation Layer (Contd...)



## Request processing in presentation layer

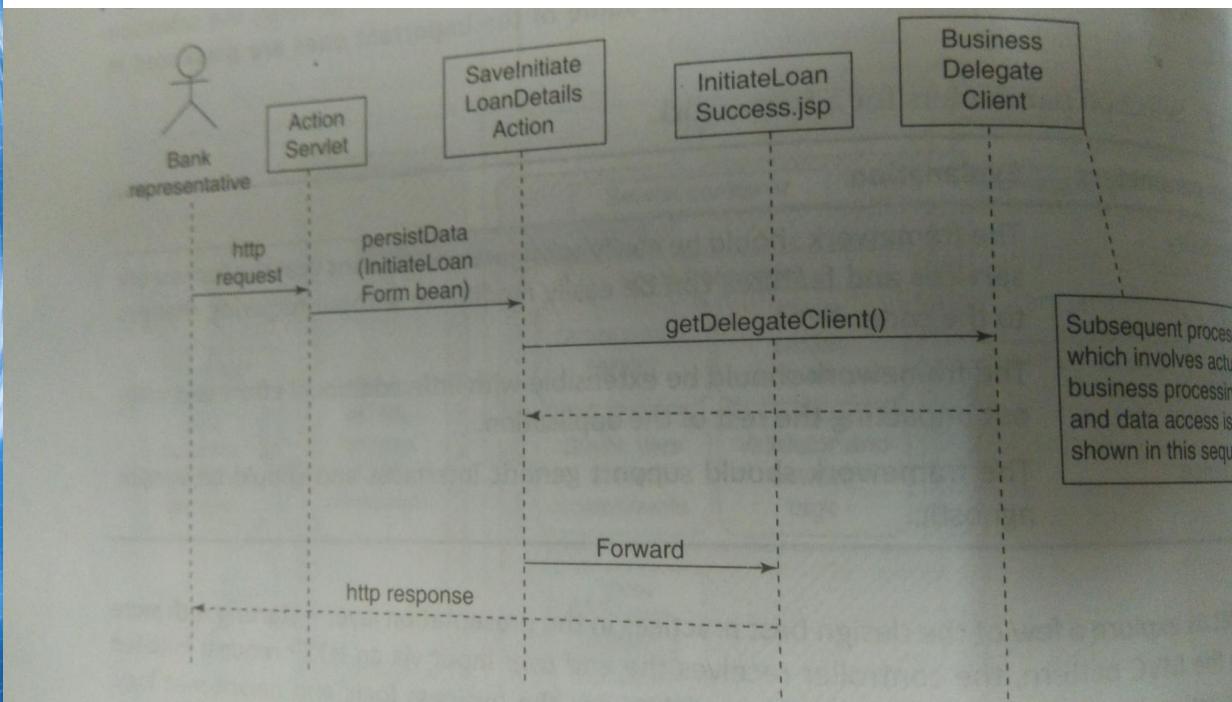


Figure 3-15 Request processing in the presentation layer.



## Presentation Layer-Design Aspects

Following are a few design aspects which help to ensure optimum design of the presentation layer.

- Usability
- Input validation
- Internationalization
- Data transformation
- Navigation
- Session and cache management
- Deployment considerations
- Reusability
- Security
- Portal Integration

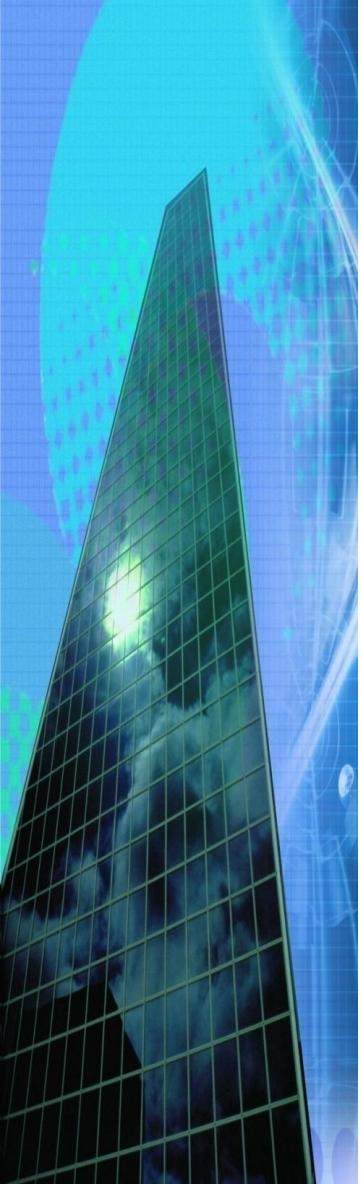


## Presentation Layer Best Practices

- Keep JSPs lean as it is comparatively difficult to debug JSPs, and for performance reasons.
- Avoid control logic in JSPs.
- Physically, keep JavaScript resources external and embed them logically wherever required by using the include mechanism through the <%include%> directive.
- Avoid exposing presentation layer data structures to the business layer to minimize the coupling between them.
- Avoid client side caching of dynamic pages to avoid issues of stale data.

**Contd...**

•



## Presentation Layer Best Practices (Contd...)

- Always place shared page elements in a separate file to achieve reuse benefits and consistent look and feel.
- Use JSTL (Java Server Pages Standard Tag Library) to handle the internationalization and localization aspects of the enterprise application. Using framework specific internationalization support, where available, is generally a good option.
- Consistent error messages should be used across the application.

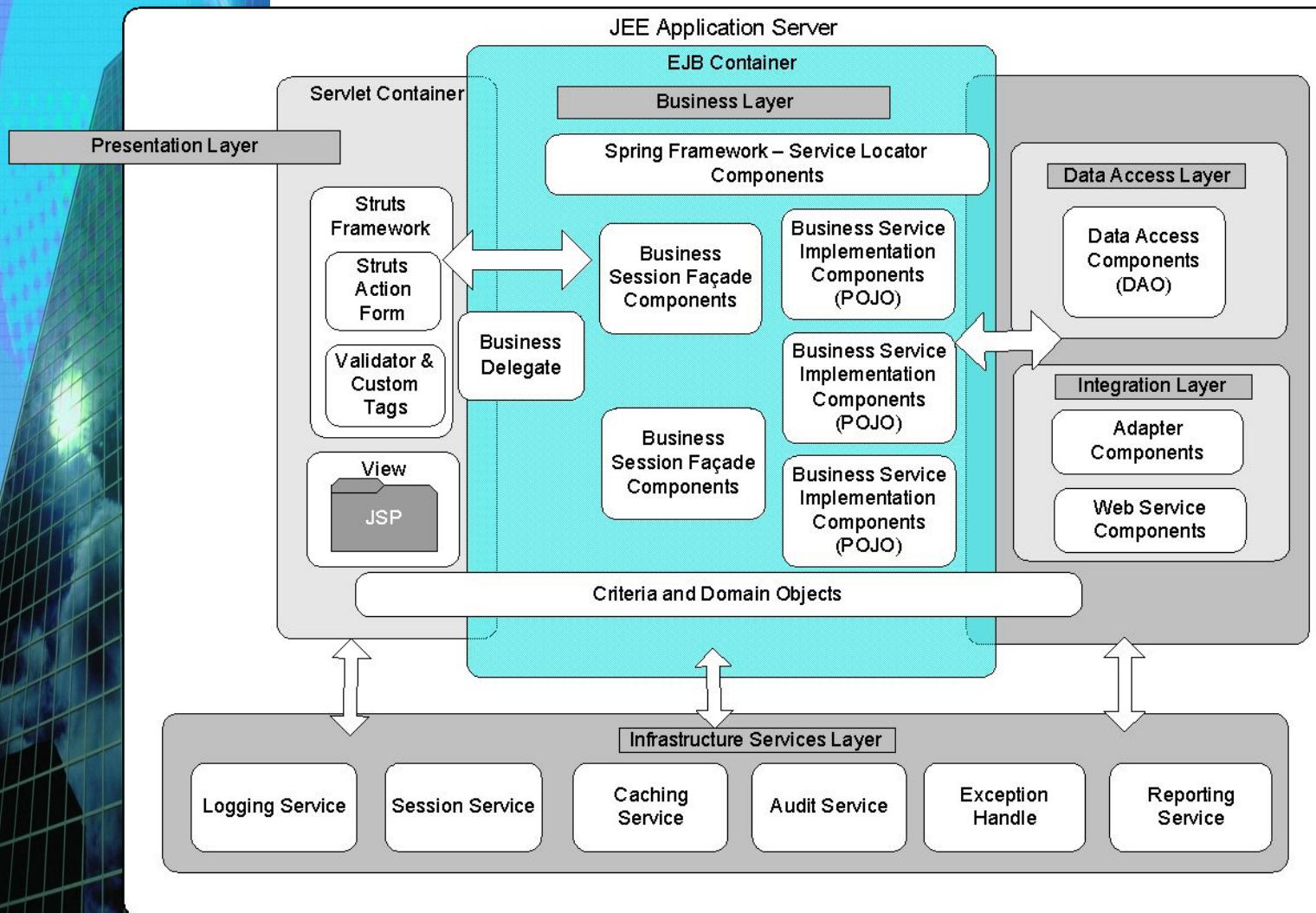


## Business Layer

- Business layer is the **heart** of an enterprise application.
- Business layer mirrors the business processes of the organization and the complexity of the business layer is directly proportional to the complexity of the business domain of the enterprise application.
- It is at the center of all other layers in a typical n-tier application, and is also known as middle tier.

Contd...  
•

# Business Layer (Contd...)





## Business Layer Components

In a typical enterprise application, business layer is usually divided into the following four components:

- **Business delegate components** act as a bridge between presentation and business layer by exposing the business processes and services to the presentation layer.
- **Session façade components** help in reducing the network traffic and latency by grouping several fine grained invocations of the business entities.
  - Implemented by enterprise session beans

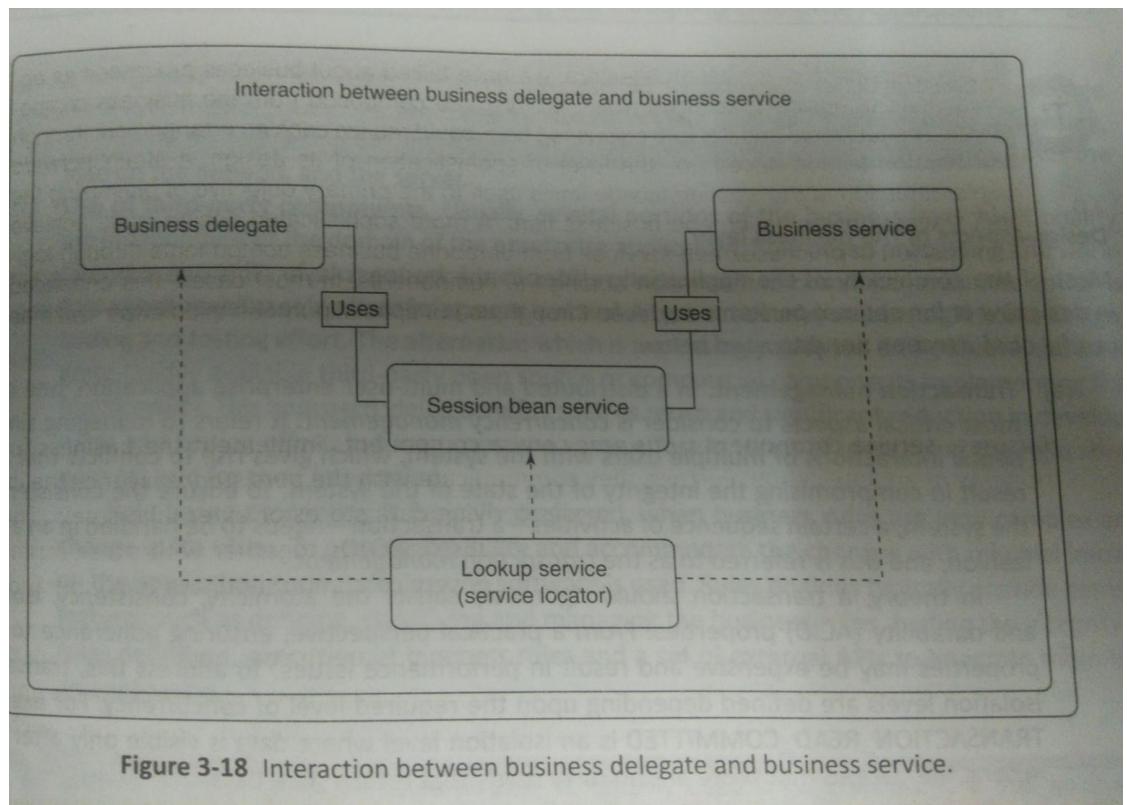


## Business Layer Components (Contd...)

- **Business Service components** implement the business functionality of an application.
  - Implemented as POJOs
- **Business model components** are the business entity classes which encapsulate the real business data and the methods to manage that data.
  - Implemented as EJB or POJOs

Contd..

# Interaction between business delegate and business service



## Request processing in the business layer

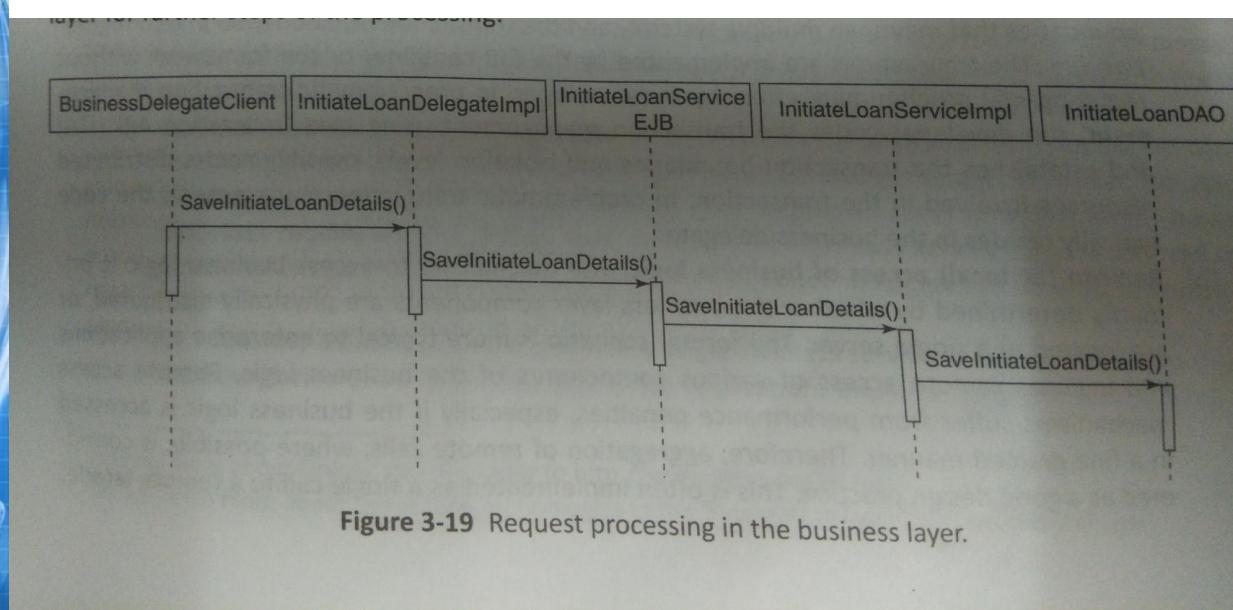


Figure 3-19 Request processing in the business layer.



## Business Layer-Design Aspects

Following are a few key aspects of the business layer design:

- ] Transaction management
- ] Remote or local access of business logic
- ] Use of third party components
- ] External business function integration

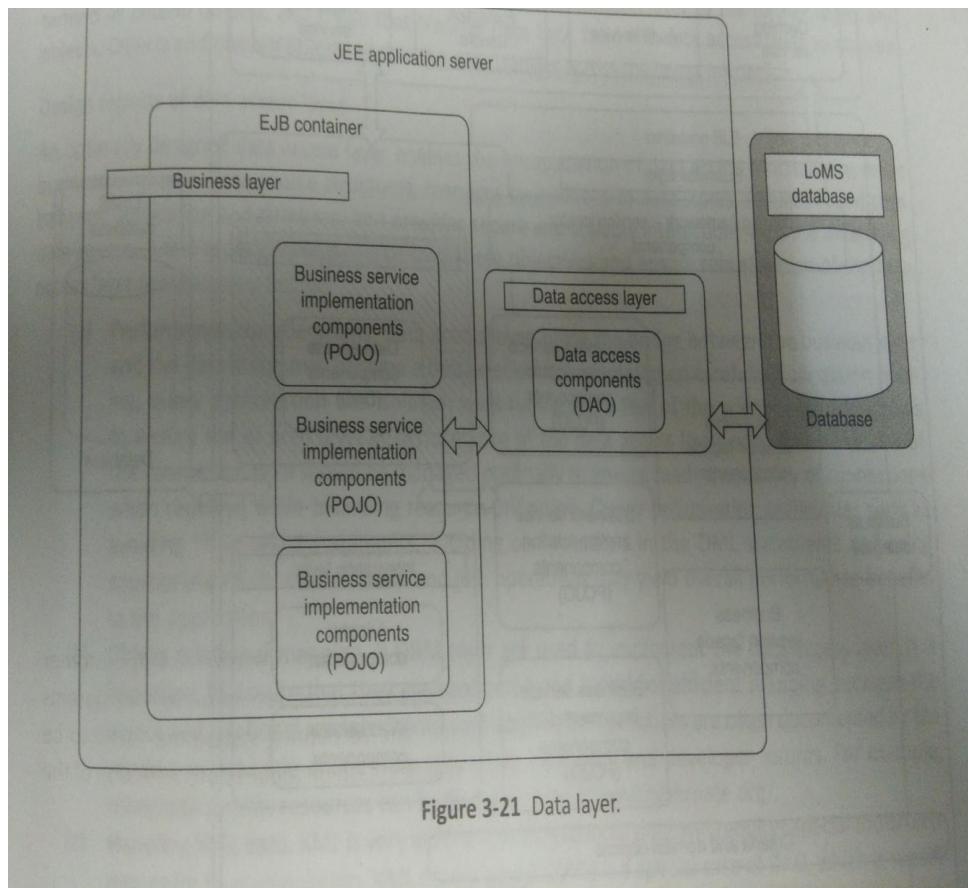


## Data layer and External Systems Layer

- Data layer physically stores the enterprise application data.
- It typically consists of relational databases, but may also have data present in several other data stores such as XML, directory and flat file structures.
- External systems layer represents the collection of other applications with which the enterprise application under consideration has to interact with.

## Data layer

- A data layer which has a relational database connected to the business layer through a data access layer



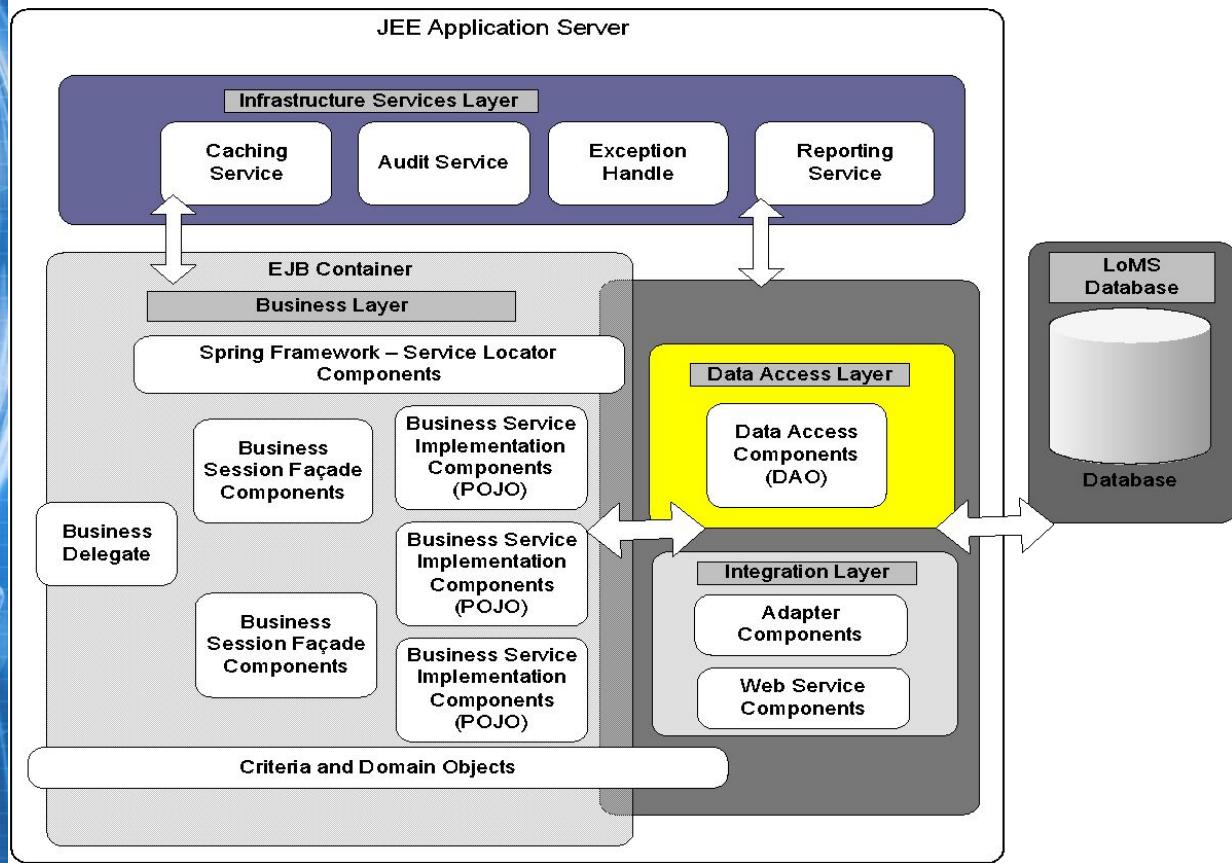


## Data Access Layer

- Data access layer of an enterprise application is responsible for accessing the elements of the underlying data layer which is usually a relational database, on behalf of the business layer.
- Data access layer typically uses Data Access Object (DAO) design pattern which encapsulates the details of access mechanism of the underlying data store.

Contd...  
..

## Data Access Layer (Contd...)





## Data Access Layer- Design Aspects

Following are a few aspects which help to ensure robust design of the data access layer:

- ] Performance considerations
- ] Object Relational Mapping
- ] Handling XML Data
- ] Security considerations
- ] Integration of distributed data



## Data Access Layer- best practices

- ] In dynamic queries that include user input data, ensure that it is always validated to minimize the possibility of an SQL injection attack
- ] Avoid idle database connections for long periods
- ] Co-locate the data access layer with the business layer to improve performance
- ] Consider caching of frequently accessed data in the data access layer



## External systems layer

- Represents the collection of other applications with which the enterprise application has to interact with.
- A partial list of the internally hosted systems
  - Core product system (loan) for getting the loan product details
  - Customer management system(CMS) for getting the customer demographic details
  - Customer relationship system (RL) for customer account relationship details
- A partial list of the externally hosted systems
  - Credit bureau system for getting customer credit rating details
  - Insurance system for applying insurance premiums for the loan products



## Integration Layer

- Integration layer provides the capability to connect to the external systems through their exposed interfaces.
- An integration layer typically consists of components that enable the enterprise application to either expose a certain capability of the application to the outside world, or similarly to consume the capabilities of an external system to implement some of its functionality.

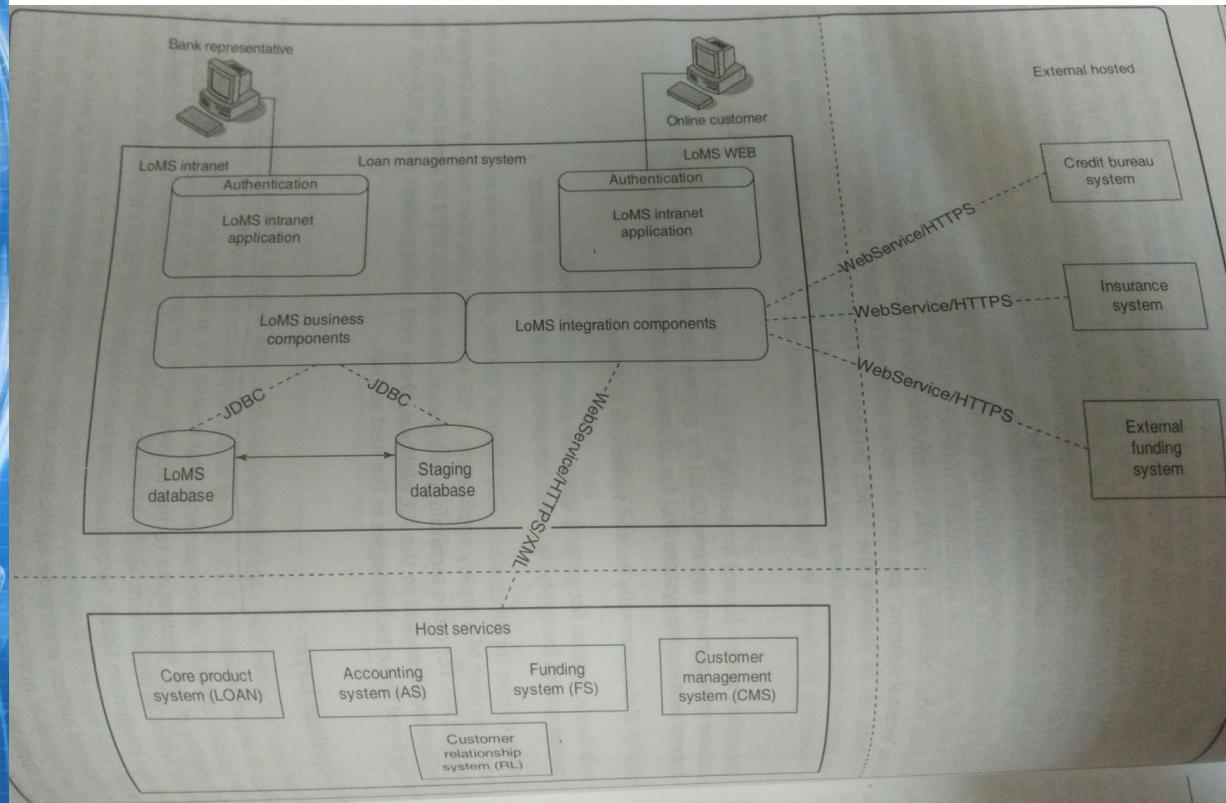
Contd...  
•



## Integration Layer (Contd...)

- The generic approach to arrive at the implementation of the integration layer is outlined as follows:
  - External interfaces with which the enterprise application has to communicate and integrate are identified.
  - Integration types are identified and designed for each external interface as dictated by the integration architecture.
  - The design team performs various other detailed design activities based upon the technology selected to achieve the requisite functionality of the integration layer.

# LoMS Integration landscape



## Integration layer

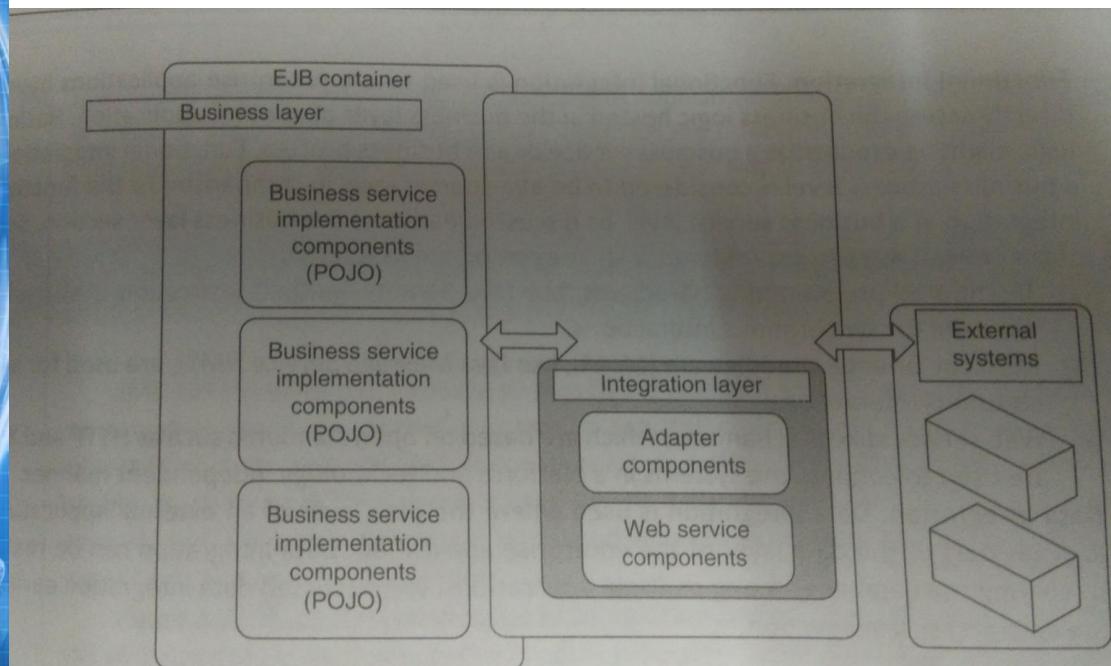


Figure 3-25 Integration layer.



## Integration Types

### □ UI integration

- Typically used where some part of the UI or views of an external application has to be part of an enterprise application.
- realized through primitive techniques like screen-scraping where the UI stream (usually HTML) is captured, parsed and composed to display the relevant UI elements. The other more sophisticated approach is to use portal integration.

### □ Functional Integration

- used where enterprise applications have to directly access the business logic hosted in the business layer of another application.

**Contd...**

•



## Integration Types (Contd...)

### □ Data Integration

- used where the data layer of an external application is logically part of the data layer of the enterprise application.



# Evolution of integration technologies

- ]] Third party EAI solutions
  - Request-reply, publish-subscribe methods
- ]] Java based application integration
  - Java message service (JMS)
  - Java Connector Architecture (JCA)
- ]] Web services
  - B2B, B2C applications
  - SOAP based and REST based web services
- ]] Enterprise service bus (ESB)
  - service oriented architecture



## Integration Layer-Design Aspects

Following are a few aspects which help to ensure reliable design of the Integration layer:

- ]) Communication Triggers
- ]) Interface Contract
- ]) Error Handling
- ]) Integration topologies
  - Point to Point topology
  - Message Broker topology
  - Message Bus topology

# Technical solution ecosystem

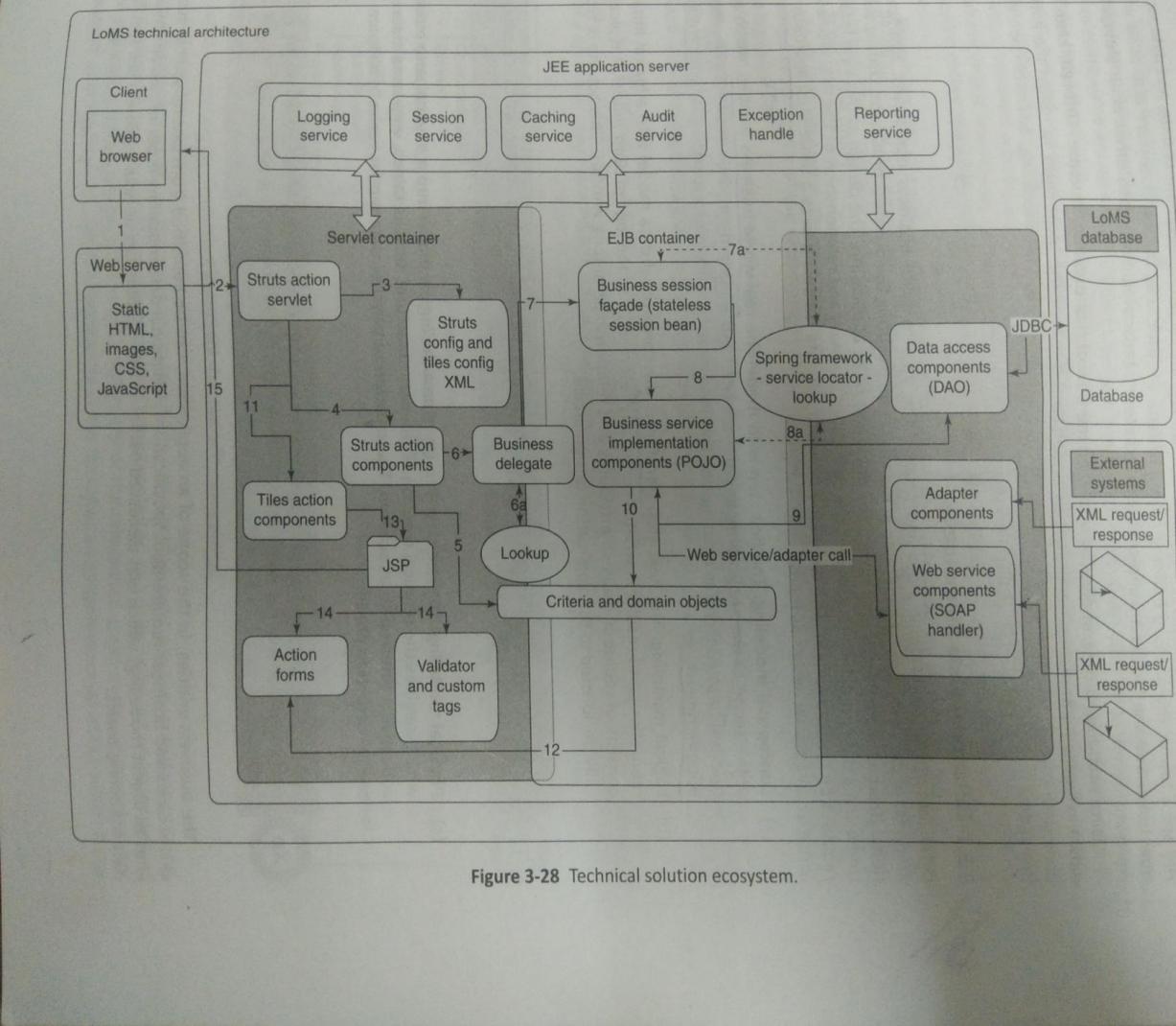


Figure 3-28 Technical solution ecosystem.



## Data Architecture and Design

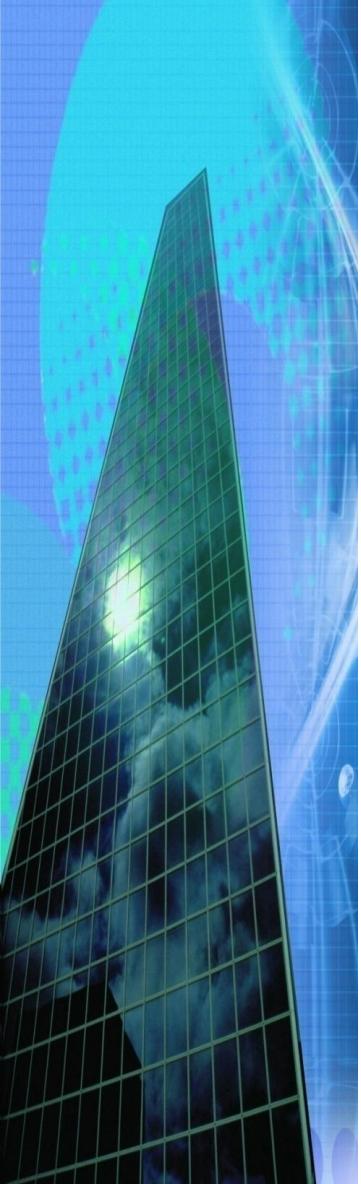
- Data architecture is one of the domains of enterprise architecture which helps in defining data models for businesses.
- Identification of business entities/data, data modeling, life cycle management of data, and data security are a few of the primary building blocks of data architecture.
- Data architecture also deals with how these building blocks are interrelated and their interactions with each other in an enterprise context.

Contd...  
.



## Data Architecture and Design (Contd...)

- Data modeling is one of the important building blocks of data architecture.
- Relational databases are typically used to implement the data layer of an application.



## Relational Data Modeling

The technique of Relational data modeling uses a three step approach to arrive at an optimal design as follows:

- **Conceptual modeling:** model the real world business entities, which are in the scope of the application, and the relationships among them.
- **Logical modeling:** transforms the conceptual data model into a logical implementation.
- **Physical modeling:** involves the physical creation of the data structures such as tables, views and indexes, collectively referred to as schema.



## Other data Representations

- XML
- Object oriented
- Directory
- Unstructured representations