Diplomarbeitspräsentation der Fakultät für Informatik

State of the art of Enterprise Web Application Development

FAKULTÄT FÜR !NFORMATIK

Magisterstudium Wirtschaftsinformatik

Markus Demolsky

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme
Arbeitsbereich: Business Informatics
Betreuung: o.Univ.-Prof. Dipl.-Ing. Mag. Dr. Gerti Kappel

Abstract

The software industry is a very dynamic market that forces software companies to react as fast as possible and to adjust their software products to the actual market requirements. To enable this, one of the most important factors of software develop ment is the underlying architecture of the software and component reuse.

An enterprise application is very valuable for the enterprise. Enterprise applications usually involve persistent data and many people access data concurrently. There is also a rich set of graphical user interfaces to offer the users a convenient way to do their work. In addition, it is also necessary to intergrate other enterprise applications scattered around the enterprise:

With the rise of object-oriented thinking, frameworks became more popular and sprout out than never before. A framework is an approach of software reuse and enables developers to establish the software on an abstract layer. On the web there are a lot of frameworks for web development. Current frameworks include Java Server Faces, Struts, Spring, Hibernate, Cocoon, OpenLaszlo, JBPM, Maven and many others.
Software developers recognize that due to the use of such frameworks most of the code in an application becomes declarative which enables software companies to react fast to new requirements in their software product. To develop successful framework-based applications, developers should know what frameworks are and how to deal with them. This is an important part, because the use of frameworks influences the underlying architecture of the application.

It depends on the requirements of an application, which set of frameworks will be used. Hence you take a different set of frameworks for a server side application than for a rich internet application (RIA).

This diploma thesis gives an introduction to the world of frameworks related to enterprise web application development and how to use and combine them. Therefore, a couple of frameworks will be evaluated by studing their capabilities, each one pursuing different goals. The diploma thesis is attended by a practical example which demonstrates the employment of the frameworks in an enterprise application, and explores their compatibility and practical usability. This practical example uses a set of frameworks to make the software components maintainable, re-useable, exensible, and configurable through declarative programming.

Open Laszlo

OpenLaszlo is an open source technology with the goal to develop Rich Internet Applications. In October 2004 the Laszlo Presentation Server became open source and is now available in version 3.2. Since version 3.2 OpenLaszlo can produce both flash and DHTML from a single LZX source. In the case of flash this technology uses the flash player as common rendering engine for their components. Behind the engine a set of XML files are burrowed that define the components. Applications in OpenLaszlo are written in a laszlo specific markup language called LZX. These LZX files consists of several XML tags and Java script code. OpenLaszlo provide mechanism for animation, layout, data binding, serverc ommunication and declarative UI.

JavaServer Faces

JavaServer Faces (JSF) is the first official standard for web application development and was developed under the Java Community Process (JCP) as JSR-127. The first early draft of the specification was released in September 2002. In March 2004 the final release of the JavaServer Faces specification has appeared JSF consists of three parts:

A set of prefabricated UI components

An event-driven programming model

A component model that enables thirdparty developers to supply additional components

JSF is based on technologies like JavaServer Pages (JSP),Java Serlvets, Tag Libraries and XML.

Struts

The Struts project was born in May 2000 by Craig R. McClanahan and provid a variation of MVC [18] that was adapted for web development, called Model 2. Struts 1.0 was finally released in 2001 and is now part of the Jakarta project from the Apache Foundation. With 26 java packages and more than 250 java classes, the Struts framework is one of supplest web frameworks available on the internet. The Struts framework provides built-in support for many common elements of web application like internationalization, exception handling, message and error logging.

Spring

The key points of Spring are:

Lightweight
Spring is lightweight in terms of both size and overhead. Spring consists of several modules that can be used as required. These modules are distributed as JAR files. Furthermore, Spring is no intrusive: objects in a Springenabled application typically have no dependencies on Spring specific classes

Inversion of control
Applying IoC, the container gives the dependencies to the object at instantiation without waiting to be asked.

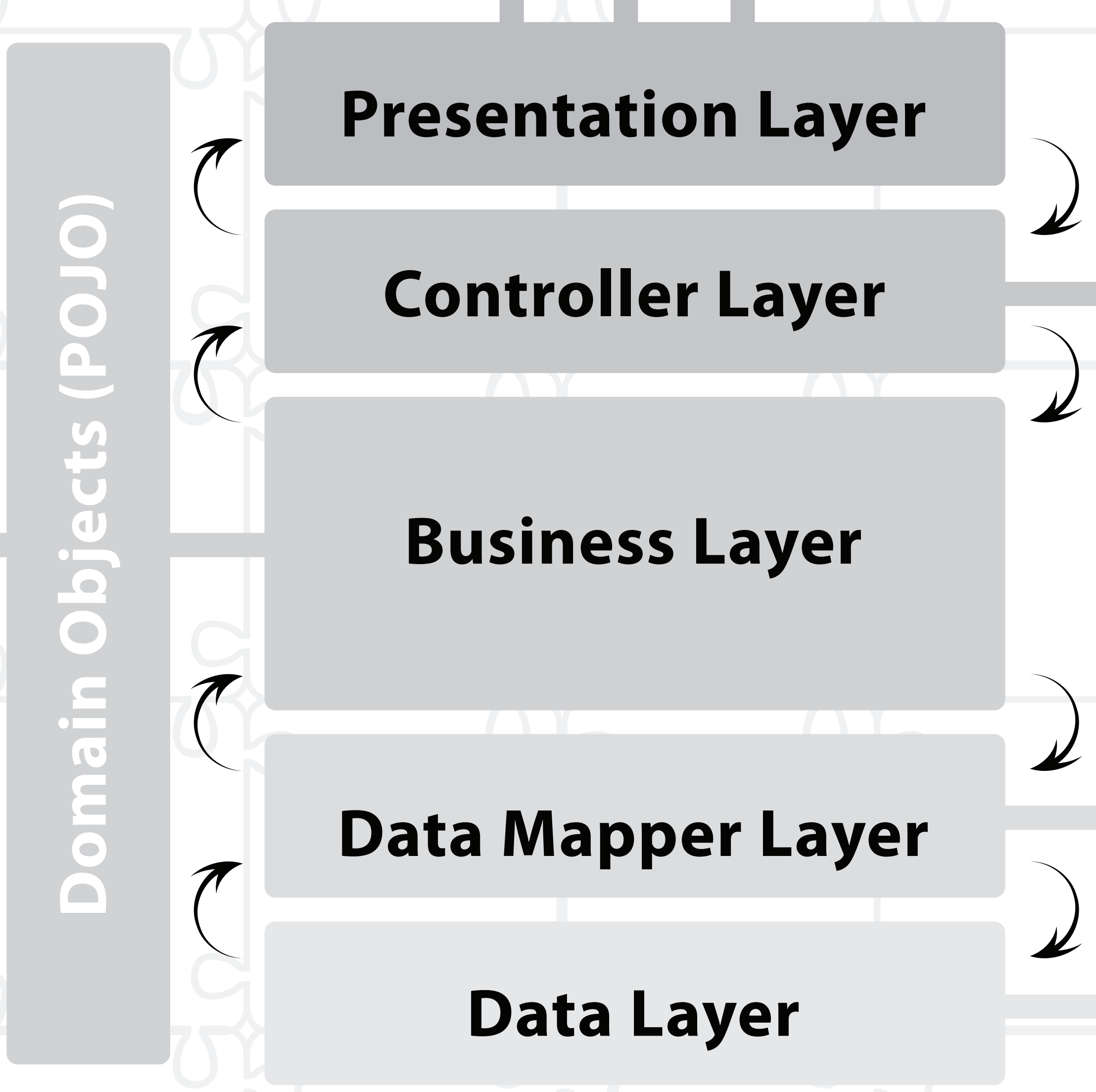
Aspect-oriented
Especially in Version 2.0 the framework comes with rich support for aspect-oriented programming that enables cohesive development by separating application business logic from system services.

Container
Spring is a container in the sense that it contains and manages the life cycle and configuration of application objects.

Framework
Spring makes it possible to configure and compose complex applications from simpler components.

Easy access to other APIs,
like EJB, JMS and others

Integration of other Frameworks,
like Hibernate, iBatis, JSF or Struts



Spring Web Flow

MVC frameworks, such as Struts or JavaServer Faces, are not well designed to image business processes. Building a process that takes place over a series of steps with Struts, results in a crowd of struts configurations. To faciliate such scenarios, Keith Donald originates Spring Web Flow. Spring Web Flow (SWF) is part of the Spring web component and focus on the definition and execution of page flows within a web application.
Spring Web Flow do not replace existing MVC frameworks. Furthermore it can be used to complement the framework with a Flow manager.

There are some major benefits coming with Spring Web Flow:

Flow definitions should be naturally decipherable by business analysts

A flow definition should be easily engineerable to/ from a visual diagram, such as a UML state diagram

Flow definitions should be self-contained

A flow execution and its associated artifacts should be easily testable in isolation

Build on Spring MVC, Struts, or other MVC Frameworks

Hibernate

Nowadays many enterprise applications use relational databases to persist their information. Classical programming languages, such as Java, provide an object-oriented view of the application. To combine these two views, an object-relational mapping technique is needed. Features of an ORM solution:

An API for performing basic CRUD operations an objects of persistent classes

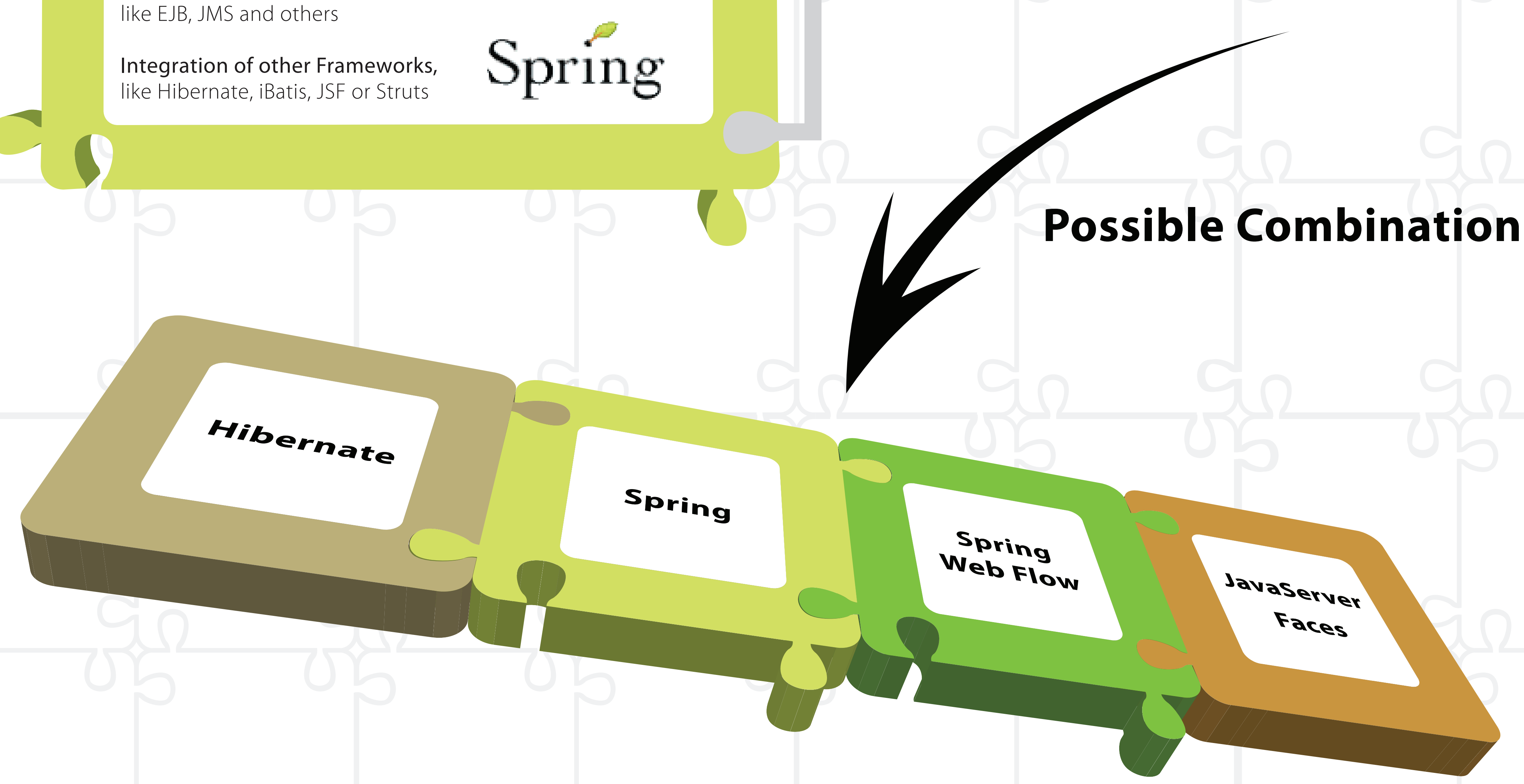
A language or API for specifying queries that refer to classes and properties of classes

A facility for specifying mapping metadata

A technique for the ORM implementation to interact with transactional objects to perform dirty checking, lazy association fetching, and other optimization functions

Hibernate is one of the leading ORM technologies for enterprise developers and joined JBoss.org in late 2003. People can purchase commercial support and training from JBoss Inc. There is also a lot of Hibernate related stuff available on the Internet that boost the popularity of Hibernate.

Possible Combination


Hibernate
Spring
Spring Web Flow
JavaServer Faces