

Pro<DOC/>™ e-Commerce Technology - An Introduction

From Rightangle Technologies Private Limited (www.righthangle.co.in)

1 Problem Statement

Today's on-line e-Commerce applications are characterized by the following-

1. Quick time to market
2. Fast response to business changes
3. Horizontal scaling to quickly add products and features to the running site
4. Minimize downtime to make application updates
5. Vertical scaling to meet increasing user load
6. Modular and loosely coupled architecture (Service Oriented Architecture) to allow easy reconfiguration and adaptation to new technology
7. Database/server neutrality to allow deployment in different environments
8. Process agility
9. Separation of the User Interface (Maintained by experts in User Interface engineering) and business process (Maintained by experts in application development).

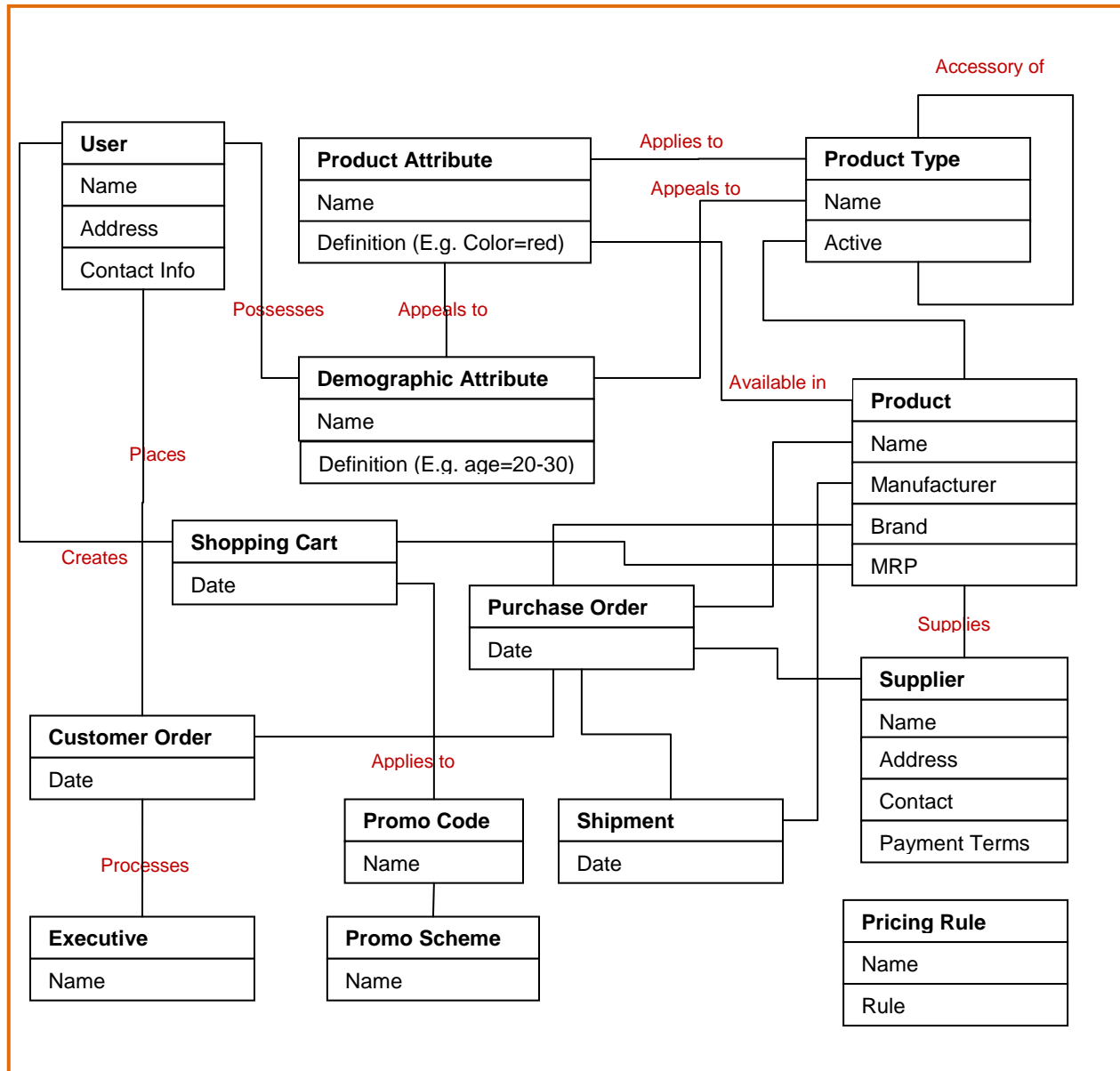
When trying to develop these applications using conventional 3GL technologies such as (DotNet, J2EE, PHP etc.) there are several problems-

1. The solution implementation requires multitude of technology frameworks/API's. E.g. in a J2EE environment, a typical implementation would require
 - Spring/Struts framework for MVC and object lifecycle management,
 - Hibernate for Database access,
 - XML processing API,
 - A suitable logging framework for logging management,
 - JAXB framework/POJOs for Web-service provision and consumption,
 - Cache management API for data caching and optimization,
 - Application Properties file management,
 - Distributed Session Management,
 - Many other technologies depending upon the nature of the application.
2. The programming resources in most of these environments are scarce and it is difficult to recruit/retain resources that are skilled in of all these technologies.
3. Once these applications are built they become very complex and in the long run they become a liability to maintain. A sizeable team is required just to keep the system running.
4. Making architectural changes becomes difficult, unless the design is very modular and substantial effort is invested upfront in keeping it clean and modular.
5. The job market is changing drastically and the quality and cost of resources will continue to pose a challenge to the management.

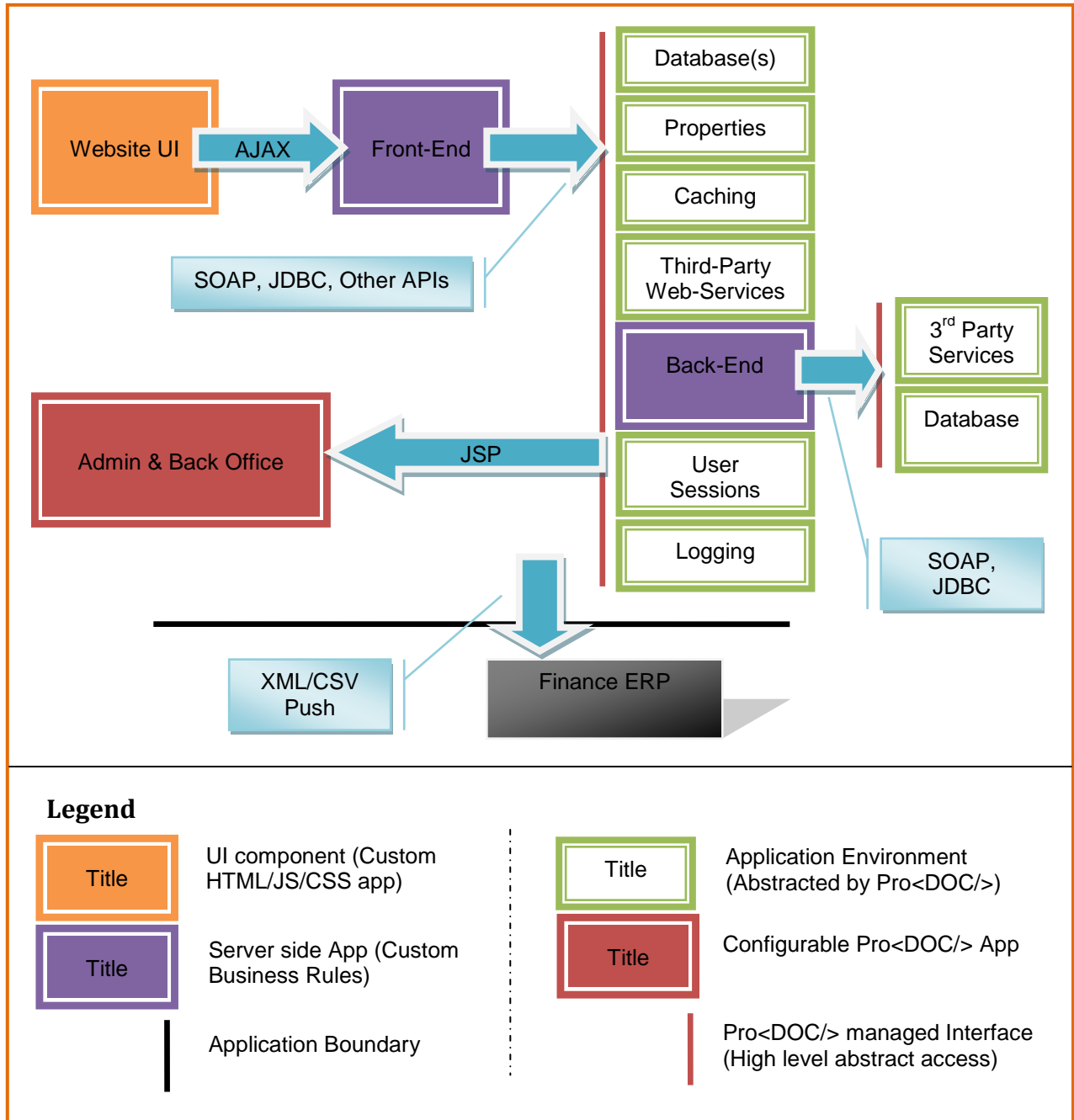
2 Introducing Pro<DOC/ >

- ✓ Complete application development platform for on-line e-Commerce business applications
- ✓ Business Rules driven – focus on business logic rather than technical intricacies, which Pro<DOC/> hides from the developers
- ✓ Cuts development and maintenance costs down to 20% (Save 80% in time and cost)
- ✓ Supports and encourages the Service Oriented Architecture and design
- ✓ Supports hosting in cloud environment
- ✓ Total separation of UI from the application allows independent development and maintenance of User experience from the business processes
- ✓ Allows application changes on the fly – without requiring server restart or interruption to live users
- ✓ Reduce skill requirements to core programming skills only
- ✓ Reduce “Time to Market” and increase responsiveness to business changes
- ✓ Reduce total cost of ownership
- ✓ Full coverage – Front End, Back End, Back Office
- ✓ Full support for all major databases (MySQL, MS-Sql, and Oracle)
- ✓ J2EE based

3 On-line e-Commerce Application - Data Model



4 On-line e-Commerce Application - Architecture



5 Pro<DOC/> as a platform for On-line e-Commerce Applications

Pro<DOC/> is a complete platform for building On-line applications fully covering the following major components-

1. Front-End
2. Back-End
3. Back-Office & CRM

Much of the business processing would reside in these blocks and Pro<DOC/>™ simplifies the development by hiding the details of technical intricacies of interfacing with most of the environmental components. The high-lights of the platform are-

1. An integrated application designer, which allows the applications to be built using a high-level Business Rules interface with minimal or no programming. Low-level programmed modules can be added as Java classes and methods.
2. Provides a high-level abstraction of Databases, Memory Caches, Session Management, Logging, Property files, Web-Services, etc. allowing these to be used without any knowledge of the underlying programming APIs.
3. Reduces the skill requirement for programmers in development and maintenance teams. The developers only need to know core Java and the Pro<DOC/>™ framework. No knowledge of frameworks like Spring/Hibernate or Struts, JDBC, Logging frameworks, SOAP, AJAX, JSON, POJO, Web-Services/WSDL etc. is required. A typical engineer with 0-3 years of Java programming experience can be hired for development.
4. Supports all major databases automatically, the user is completely shielded from the internal SQL.
5. Cuts the development and maintenance effort/time down to 20%.
6. Increases maintainability many fold.
7. Helps create business process documentation.
8. Installs and runs on Windows as well as Linux based environments.

5.1 Front-End Development

The Pro<DOC/> front-end is developed as an HTTP server for JSON/AJAX objects for use from the HTML UI through AJAX calls. This approach helps keep the design modular by creating a separation between the UI and the server side code.

1. Develop Business Procllets from a high-level designer UI, with minimum or no programming.
2. Automatically import XSD schema, Database definitions, and WSDLs into the designer.
3. Access Databases (choose your database from Oracle, MySQL or MS-SQL), Web-Services, Memory Caches, Session data, Logging, Properties etc. as high-level abstractions, without having to deal with individual APIs.

4. Integrate user-written business logic in Java.
5. Render results as JSON or XML.

5.2 Back-End Development

The Pro<DOC/> back-end is developed as SOAP Web-Services for use from the Front-end or third-party applications.

1. Develop Business Proclets from a high-level designer UI, with minimum or no programming.
2. Automatically export WSDL, for third-party integration.
3. All high-level abstracts available to Front-End are available to Back-End as well.

5.3 Back-Office Development

1. Create business processes, schema, forms, workflow rules from a designer – no programming required.
2. Integrate documents into the processes.
3. Initiate processes automatically from Front-end, back-end processes or receive through XML push or Email.
4. Built-in CRM module – can be easily enhanced or replaced with custom implementation
5. Full audit trail of changes for accountability and traceability.
6. Integrated data analytics.
7. Create workspaces on the fly – no programming required.
8. Write integration code with minimal programming.
9. Write business validations and rules with minimal programming.