



SaaS - Technology Enablers

Author
Karthik Viswanathan | SaaS Architect | Aspire Systems
<http://www.aspiresys.com>

Table of Contents

1. Introduction.....	2
2. Technology Concepts.....	3
2.1. Sharding.....	4
2.2. Service Oriented Data Architecture.....	4
2.3. Aspect Oriented Programming.....	4
3. Technology/Platform-Specific Features.....	5
3.1. Java.....	5
3.1.1. Spring Framework.....	5
3.1.2. Spring Web Services.....	5
3.1.3. eFace.....	6
3.2. .Net.....	6
3.2.1. Windows Workflow Foundation (WF).....	6
3.2.2. Windows Presentation Framework(WPF).....	7
3.2.3. Windows Communication Framework(WCF).....	7
3.3. XAML/XOML.....	7
3.4. Active Directory Application Mode.....	8
3.5. Oracle.....	8
3.5.1. Virtual Private Database(VPD).....	8
3.5.2. REF Partition Strategy in Oracle 11g.....	9
4. Integration Tools.....	9
4.1. Jitterbit.....	9
4.2. Mule Source.....	10
4.3. Talend Open Studio.....	10
4.4. Boomi.....	10
5. Conclusion.....	10
6. About Aspire.....	11

Technology enablers for SaaS

1. Introduction

A robust SaaS product can be built with a minimalist approach, for example, Tomcat, MySQL, XML, Struts 2.0, POJO and JSP are all it takes to build a SaaS application/product using Java technology. And it can work very well too. However, there are additional tools/techniques that can be utilized effectively when building SaaS products.

This article consolidates generic technology concepts that can be harnessed for SaaS, platform-specific features for .NET, Oracle and Java, and useful integration tools. The techniques and tool-kits mentioned have not been developed specifically for SaaS; in fact, some of them have been around for several years and are widely used today in building enterprise products. Nevertheless, they will come in handy when SaaS-enabling a product.

2. Technology concepts

- Service Oriented Architecture (SOA)
- Web services
- Sharding (Entity partitioning)
- Service Oriented Data Architecture (SODA)
- Aspect Oriented Programming (AOP)

As SOA and Web services are relatively well-known and discussed frequently, we will elaborate on the other three.

2.1. Sharding

In this federated model, an entity is split and stored in multiple database servers. For example, user ids 1-100,000 live on one instance while the next 100,000 is stored in another instance. From a SaaS perspective, tenants A & C can be stored on one server while data for tenant B is stored on another server. Though sharding is not perfect, it offers high availability and faster queries.

2.2 Service Oriented Data Architecture

This concept entails that the databases are partitioned according to well-defined and logical service boundaries. At a logical level, a database service exposes a well-documented interface to data. This is not a general database interface for reading and writing data, instead it provides very specific functionality. For example, an inventory database service might expose methods for checking inventory levels, reserving inventory, removing products from inventory, etc.

While this may sound similar to stored procedures, what is different is that access to data under the control of one database service is completely isolated from that of a different database service. Moreover, requests to database services are exposed as Web services (instead of a database connection).

2.3 Aspect Oriented Programming

Code related to cross-cutting concerns like transaction, security and logging are usually common across a product. If any of these needs to be modified, large parts of the code base are impacted. This increases the complexity of a product and makes product evolution difficult.

AOP attempts to solve this problem by allowing cross-cutting concerns to be expressed in stand-alone modules called aspects. For example, a security module can include ‘advice’ that performs a security check before accessing a bank account. The pointcut defines the instances (join points) when a bank account can be accessed. Using AOP, we can link advices and join-points during runtime, thus giving flexibility to link both dynamically.

3. Technology/Platform-specific features

3.1 Java

3.1.1. Spring Framework

Spring has a layered architecture and provides a better way to manage business objects. While it is most valuable in the middle tier as a good service design pattern, its configuration management services can be used in any layer.

Spring’s use of the ‘Inversion of Control principle’ (also called the Hollywood principle – ‘Don’t call me; I’ll call you’) is well documented and can be leveraged for building loosely coupled objects/components that form the foundation of a well-architected SaaS product. The latest version, Spring 2.5, released a fortnight ago, provides comprehensive support for configuration annotations, thus underlining the framework’s core themes of flexibility and power.

3.1.2. Spring Web Services

Spring Web Services (Spring-WS) is a product of the Spring community focused on creating document-driven Web services. Spring Web Services aims to facilitate contract-first SOAP service development, allowing for the creation of flexible web services using one of the many ways to manipulate XML payloads.

Spring-WS is based on Spring itself, which means you can use Spring concepts such as dependency injection, as an integral part of your Web service. Spring-WS supports best practices such as the WS-I basic profile, Contract-First development, and having a loose coupling between contract and implementation. The other key features of Spring Web Services are powerful mappings, XML API support, flexible XML marshalling, WS-Security, etc.

3.1.3. eFace

It is a presentation framework developed using Java. It is the XAML/WPF equivalent for the Java world. It enables one to develop products with rich UI using declarative programming. It is well-suited for SaaS-products because it allows for templated customization of UI.

3.2 .NET

3.2.1 Windows Workflow Foundation (WF)

Workflow provides a way of describing the order of execution and dependent relationships between pieces of short or long running work. Since SaaS products frequently support different workflows based on tenant preferences, WF can be used as the programming model and engine for building workflow-enabled applications on Windows. WF includes support for both system workflow and human workflow across a wide range of scenarios, including workflow within line of business applications, user interface page-flow, document-centric workflow, human workflow, composite workflow for service oriented applications, business rule driven workflow and workflow for systems management

3.2.2 Windows Presentation Framework (WPF)

It is the graphical subsystem feature of the .NET Framework 3.0. It provides a consistent programming model for building applications and provides a clear separation between the UI and the business logic. It also enables richer control, design, and development of the visual aspects of Windows programs. A powerful concept in the WPF is the logical separation of a control from its appearance.

3.2.3 Windows Communication Framework (WCF)

The WCF programming model unifies Web Services, .NET Remoting, Distributed Transactions, and Message Queues into a single Service-oriented programming model for distributed computing. It is intended to provide the rapid application development methodology for the development of web services, with a single API for inter-process communication in a local machine, LAN, or over the Internet. WCF uses SOAP messages for communication between two processes, thereby making WCF-based applications interoperable with any other process that communicates via SOAP messages.

3.3 XAML/XOML

XAML is a new descriptive programming language developed by Microsoft to write user interfaces for next generation managed applications. It is used extensively in the .NET Framework 3.0 technologies, particularly in Windows Presentation Foundation (WPF), where it is used as a user interface markup language to define UI elements, data binding, eventing, and other features. It is also used in Windows Workflow Foundation (WF), in which workflows themselves can be defined using XAML. The serialization format for workflows was previously called XOML, to differentiate it from UI markup use of XAML, but now they are no longer distinguished.

3.4 Active Directory Application Mode (ADAM)

It is an expanded capability of Active Directory for deployment as a lightweight directory service. It is a Lightweight Directory Access Protocol (LDAP) directory service that runs as a user service, rather than as a system service.

Business data is often organized in a unique way for a particular business or organization. When an organization must use a directory-enabled application, based on unique business logic or schema, ADAM can be the directory service for the application. For example, an application developer might want to expose application data through LDAP in a way that is unique and specific to the clients of the application. Using Active Directory Application Mode, and without altering the configuration of Active Directory, the application developer can provide an application with a tailored schema that works for business needs, data requirements and workflow processes.

3.5 Oracle

3.5.1 Virtual Private Database (VPD)

It works by transparently modifying requests for data to present a partial view of the tables to the users based on a set of defined criteria. During runtime, predicates are appended to all the queries to fetch only the rows that a user is supposed to see. For example, if the user is supposed to see only accounts of tenant XYZ, the VPD setup automatically rewrites the query “select * from accounts” to “select * from accounts where tenant='XYZ’”. This is achieved by having a security policy on the ACCOUNTS table. Oracle Label Security (OLS) is another feature that supports row-based-security for restricting user access to specific data. It builds on the Virtual Private Database feature.

3.5.2 REF Partition strategy in Oracle 11g

This feature allows partitioning of multiple tables together using the parent-child relationship of the tables. Note that the partitioning keys are not stored as actual columns in the child table. Eg: If Parent ORDER table is partitioned by tenant ID, the child ORDER_DETAILS table inherits the partition even though tenant ID is not stored in the child table. This is optimal because data for each tenant is stored together automatically and transparently in the respective partitions.

4. Integration tools

Integration of disparate systems is a key challenge in general and particularly so in SaaS, as we need to integrate an application/product 'in the cloud' and an on-premise system.

4.1. Jitterbit

This is an open source integration tool that helps to design, configure, test and deploy integration solutions. It supports most major document types and protocols. Like all integration products, Jitterbit is based on writing interfaces that translate simple user input into the queries or commands that insert and extract information into each system being integrated. Once the user has indicated that one system's "Employees" field is another system's "Staff" field, or whatever transformations are necessary, Jitterbit writes the information out to an XML file. This standardized package, called a Jitterpak, is the heart of the system. Essentially, a Jitterpak preserves all the information needed to connect to two different systems and share data between them.

4.2. MuleSource

It offers an open source enterprise service bus (ESB) solution and has its origins in the open source Mule project. It provides Web Services support using Axis & XFire, integrates with Spring and provides out-of-the-box integration with many legacy, commercial products.

4.3. Talend Open Studio

This data integration software generates data integration processes in Java and offers Extract, Transform and Load (ETL) functions for data warehousing and analytics, working on scale out, commodity hardware grids. It includes over 100 native connectors for common enterprise applications. On the operational side, Talend handles data migration, data loading, synchronization or replication of databases and data exchange between systems. It also has a SaaS version of its product called Talend On Demand.

4.4. Boomi

It is a provider of business integration solutions for small and mid-sized business (SMBs). Boomi's 'Visual Integration Technology' enables integration of applications, trading partners and data using drag and drop features.

5. Conclusion

SaaS is revolutionizing the software industry and it is expected to completely transform the software landscape. Distributed computer architectures and emergence of SaaS enablers are driving this growth. Technology enablers form a major component in accelerating the transition to on-demand model from on-

premise model and this article has illustrated some of the technology enablers that can be used towards this.

6. About Aspire

Aspire Systems is a Product Engineering Services organization focused on transforming product development for ISVs. Aspire works with ISVs helping them innovate and achieve their strategic goals by leveraging the experience gained by working on over 500 software products. Aspire has been providing SaaS enablement services on varying technologies for ISVs in a short duration of time and at lower R&D costs through its SaaS Center of Excellence and leveraging its partner ecosystem. For more information, please visit www.aspiresys.com.

