



Emergent Architecture: Web 2.0, SOA & SAAS

A wave of innovation has embraced the Enterprise I.T. landscape! We're entering an era of delivering rich client applications, social software and mash-up based solutions for commercial purposes. As we enter the 'web as a platform' paradigm organizations are finding it necessary to understand how to leverage these new architectural styles to deliver software that has more features, is more reusable, cost effective and agile.

"Emergent Architecture: Web 2.0, SOA & SaaS" provides both breadth and depth of training in the most important aspects of this next generation computing platform. The course covers both concept and realization of architecting software with services and leveraging the recombinant application design as well as social computing concepts to bring together the core tenets of this movement.

Objectives:

- < Understand the underlying tenets of Web 2.0, SOA and SAAS
- < Understand designs, concerns & mitigation strategies related to recombinant applications: Remixes, Mash-ups & composite software
- < Understand best practices in AJAX based development
- < Understand the concepts of Client/Service computing
- < Understand the skills, practices, and processes related to creating emergent architectures.
- < Understand why and when to use 'classic Web services' versus light-weight approaches
- < Understand the practical details of legacy enablement

Audience:

This course is designed for technicians who will be involved in the actual design or implementation of Emergent Architectures.

Prerequisites:

Attendees should understand the basic concepts of Web development. Attendees may optionally participate in hands-on labs (emphasis in JavaScript, XML, etc.)

Duration:

2 days

Outline:

1. Architectural Evolution

- Review classic computing
 - Monolithic & Client/Server
 - Three Tiered & N-Tiered
- Discuss SOA
 - What is Service Oriented Architecture (SOA)
 - Why all the fuss?
- Review Web Services
 - WS-I Basic Profile (SOAP, WSDL, UDDI, XML Schema)
 - Advanced Web Services (WS-*)
- The case for lighter weight solutions
- Introduce light-weight protocols, formats and identifiers

2. Introducing AJAX

- Review the need for rich client applications
- Designing for asynchronous behavior
- Review of JavaScript capabilities
- Consuming and producing XML Messages
- Using DHTML
- A simple AJAX application

3. Connecting AJAX to Services

- Review JSON
- Using POX (plain old XML)
- Using SOAP calls
- Using SOAP with WS-*

4. Using AJAX Libraries

- Survey of available libraries
- Hands on: Prototype library
- Hands on: Scriptaculous
- Hands on: Behaviour library
- Using multiple libraries together

- Hands on: Tibco General Interface
- Serving up AJAX applications
 - AJAX with Java
 - AJAX with .Net
 - AJAX with Ruby on Rails

4. The CICS Architecture

- Review Client/Service
- Review Client-Intermediary-Service
- Review Client-Intermediary-Composite Service
- Discuss types of intermediaries (functional and technical)
- Review types of design-time structural composition
 - Orchestration driven
 - Event driven (event/condition/action)
 - Service Component Architecture
- Review types of run-time structural composition
 - Deferred composition
 - Remixes
 - Client side mash-ups
- Discuss versioning issues
- Service level agreements
- Discuss consumption & transport techniques

5. Information, Syndication and Notification

- The evolution of content: Publish, Search and Subscribe
- The importance of near real time
- The importance of targeted information
- Giving information structure
 - XML & XML Schema
 - RDF
- Giving information meaning
 - Tagging
 - Ontologies

- Folksonomies
- Microformats
- Syndicating
 - RSS and ATOM
 - SSE (Simple Sharing Extensions)
 - OPML
- Re-syndicating

6. Community Solutions

- The role of community and participation in Web 2.0
- The role of 'self service applications'
- Mark-up / Mark-down languages
- The Blog
- The Wiki & the Bliki
- Podcasting
- Case Studies
 - Blogger
 - Flickr
 - Del.icio.us

7. The Web as a Platform

- Hosted solutions
- Hosted services
- Recombinant clients
- Enterprise Web Platforms
 - Salesforce.com
 - Reardon Commerce

- The role of Software as a Service
- How can your organization participate:
 - As a user of platforms
 - As a platform provider

8. Architectural Tenets

- Simplicity over complexity
- Designing for unknown consumers
- Self service applications
- Information finds meaning through use
- Recombinant systems over capping
- Trust is key to relationships

9. The Obstacles

- Moving from art to science
- Managing rapidly changing libraries and technologies
- Versioning, change management and impact analysis
- Service Level Agreements
- Common trust, privacy, authorization, access control and encryption
- Transactions (all or none)
- Supporting the users