Building an Enterprise App Store using Enterprise Mashups

A JackBe Webinar with Dion Hinchcliffe
Wednesday, April 28, 2010

Sponsored By
Introduction

Dion Hinchcliffe

- ZDNet’s Enterprise Web 2.0
- Social Computing Journal – Editor-in-Chief
  - [http://socialcomputingjournal.com](http://socialcomputingjournal.com)
- ebizQ’s Next-Generation Enterprises
  - [http://www.ebizq.net/blogs/enterprise](http://www.ebizq.net/blogs/enterprise)
- Dachis
  - [http://dachisgroup.com](http://dachisgroup.com)
  - mailto:dion.hinchcliffe@dachisgroup.com
- Web 2.0 University
  - [http://web20university.com](http://web20university.com)
  - : @dhinchcliffe
Webinar Overview

• **Open APIs**
  Why they are changing the software and integration landscape

• **Mashups**
  The problem they solve and key aspects of composite business apps

• **App Stores**
  Adding end-user distribution to mashups and maximizing ROI
Open Supply Chains
Open Supply Chains
also known as APIs
The Platform Overtakes the Web Site

Bandwidth Consumed by Amazon Web Services

Bandwidth Consumed by Amazon’s Global Websites
Open Platform vs. Closed Platform
The Market Share Opportunity

- The **vast majority** of **user activity** in software is **elsewhere**, on 3rd party Web sites and applications.

- If **firms** could **reach** this traffic, the growth potential is as large as the Web itself.

- **Reaching** this traffic **before competitors** do can result in successful marketshare “lock-out”

- Businesses able to **cost-effectively integrate** with a **large number of partners** to grow

- Access and **offer value** to existing **ecosystems** of customers
Opportunity: Going To the Customer and Mass Customization

Open Developer Ecosystem

- App Store
- External Consumers or Enterprise Workers
- Discover & Acquire
- Use
- Live Web Integration
- Open API
- Application
- More Reuse, Higher Agility, Better Solutions, Lower Cost
- Web App or Classical IT Silo
- $$ Monetization/Chargeback Boundary

Mashup Marketplace

Discover & Acquire
Use
Live Web Integration
Open API
Application
$$ Monetization/Chargeback Boundary

Wednesday, April 28, 2010
Platforms vs. Applications

**Distribution Models**

- Native App
- Web Application
- Open Widgets
- Facebook/Open Social
- Web API
  - SDK, Developer Community, SLA, Billing

**Target Audiences**

- Consumers
- Small Businesses
- Medium-Sized Business
- Power/Web Saavy Users
- Developers
- Businesses

Wednesday, April 28, 2010
Platforms vs. Applications

Distribution Models | Order of Magnitude | Distribution Method
---|---|---
Native App | 10M Users | Push
Desktop Client API | 10M Users | Pull
Open Widgets | 10-20M Users | Pull
Facebook/Open Social | |
Open Web API | 100M+ Users | Pull
SDK, Developer Community, SLA, Billing | |
Open Web APIs + Open Apps = The Real Killer App

- Reach every distribution channel possible
- Leverage 3rd party customer bases
- Cut off competitor’s growth OPPORTUNITIES
- Ride the MAXIMUM POTENTIAL growth curve
- Harness innovation of hundreds and thousands of 3rd party developers
New Web “distribution models” driving reach and network effect
Reasons Developers Select APIs

Key to initial adoption

• Provides access to functionality not possible to develop internally
• Easy to use and integrate with
• Good documentation and easy to get started

Key to long-term adoption

• Reliable, well-known, scalable provider that is trusted
• Developers can get answers to questions, support, and problems fixed when bugs are found
• Strong user base for 3rd party developers to tap
“Platforming” Your Business

- Requires opening the server-side to 3rd party developers
- Allowing the construction of widgets and Web apps offering some or of all of your functionality by external partners
- Harnessing the innovation on the network
- Generating the greatest potential reach, competitive lock-out, market share, and revenue
- Requires effective distribution in order to create the most opportunity and value

Key Point
Open Supply Chains: The bottom line

- Good repeatability
- Requires investment
- Unproven in some industries
- Proven ROI

Strategic Industry Play
Mashups
Connecting people and data

• SOA is a modular software architecture, and the modules are services designed to interact with each other.
  – Important Note: SOA also contains higher order constructs such as composite applications, orchestration, coordination, and more exist.

• We tend to rely on open standards interoperability to encourage automatic interoperability of services designed separately.
  – A good SOA could still violate this rule however
  – See Thomas Erl and Seven Principles of SOA
  – Bottom Line: To deliver business value, SOA must be about reach. So what is the best way to bring SOA to the masses?
A key Goal of Web 2.0 and SOA: Turning Applications Into Platforms

- Openly exposing the features of software and data to customers, end-users, partners, and suppliers for reuse and remixing

- This strategy requires documenting, encouraging, and actively supporting the application as a platform
  - Has serious governance implications and achieving good ROI requires effective reach

- Provide legal, technical, and business reasons to enable this:
  - Fair licensing, pricing, & support models
  - A vast array of services that provide data that uses need
  - A way to apply these services to business problems rapidly and inexpensively.
Demand for Breadth Integration

• “48 percent of the CIOs we surveyed said that they plan to implement service-oriented architectures for integration with external trading partners this year.” – McKinsey & Co.
The Two Top-Level Organizing Principles in Modern Software Continue to Converge

Centralized Production

Emergent

Planned

SOA

Web 2.0

heavyweight standards
security
interoperability
AJAX
Web orientation
Web scale
network effects
web orientation
mashups
architecture of participation
monetization
unintended uses
openness
software as services
applications as platforms
governance
enterprise scale
composite applications
integration focus
recombinant software
Empowering The User To Self-Service & Create

- Cut-and-Paste deployment anywhere on the Intranet
- Consumption of the SOA in any application that can use a URL
- Discovery of data via search
- Integration moves out of the spreadsheet
- Bringing business data and SOA to the masses
Definition: Mashup

• “A mashup is a Web site or Web application that seamlessly combines content from more than one source into an integrated experience.” - Wikipedia

• Content used in mashups is usually sourced from a 3rd party via a public interface (API)

• Other methods of sourcing content for mashups include Web feeds (e.g. RSS or Atom), and JavaScript/Flash “widgets”
Mashups

• Strong preference for reuse over coding
  – Innovation in assembly is the core value instead of ingenuity in coding

• Disruptive delivery model: Web-based with no install, no plug-ins, no admin rights, etc.

• Design focus is at the glue instead of the functionality

• Emphasis on simple, easy-to-use Web technologies over complex enterprise technologies
What are Enterprise Mashups?

- Enterprise Mashups create **realtime**, **secure** remixes of **disparate internal and external** data sources.
  - Enterprise Mashups can be expressed as **Data Services** and **Apps**.
Presto 3.0 - Wires
JackBe’s *Presto* Enterprise Mashup Ecosystem
Apps on the Web Today: Mashups Everywhere

- The growth of Web sites with highly valuable “portable” content and functionality
- Users putting modular Web parts on their blogs and profiles to host the pieces of the Web that they want to share
- By the tens of millions on sites like MySpace and Facebook
- The increasing realization that there is limited business value in being on a single site…
Dynamic, self-service connections between apps are now industry standard

- Building open platforms instead of stand-alone applications
- Forming self-distributing ecosystems
- Spreading products far beyond the boundaries of a site
  - APIs, widgets, badges, syndication -> mashups
- **In other words:** Being everywhere else on the network
- Building on the shoulder of giants
- Leveraging widgets, libraries, and APIs from Yahoo!, Amazon, and thousands of others and others
- The automated mass servicing of markets of low demand content and functionality (*The Long Tail*)
- Which represents the bulk of the demand
With traditional methods, many (perhaps most) software solutions are too expensive to build or buy today
Enterprise Mashups:
Using the Assets of the Web and Enterprise for Ad Hoc Self-Service Application Creation

Web

- Widgets
- Feeds
- Services
- Rich Media

SOA

- Content
- Services
- Portlets
- Enterprise Data

browser-based, visually assembled, simple Web parts made of XML, Javascript, and browser plug-ins like Flash
The Enterprise Focus: Rapid Business Solutions

- Full resources of the Web and the Intranet
- Enterprise context around management, security, privacy, etc.
- Gives everyone in the organization the ability to leverage the SOA.
- Lightweight, simple model.
- Inexpensive and extremely rapid results
Situating mashups in the workplace
App Stores
Key Aspects of App Stores

• Way to create an easy to use trusted conduit of software

• Model is widely understood by both consumers and developers of software

• Provides centralized control + value add including monetization, security, safety with wide distribution and easy-on-boarding

• An increasingly expected model for software acquisition in the future for business
Modern Distribution Methods Are Bringing Mashups to the Masses

Business

Customer Ecosystem

Developer Ecosystem

Applications

Mashup Marketplace & Channels
Site | Device | API

Reach

ROI

network effect

build
cultivate

commitment

use

buy

create

use

App Store

Wednesday, April 28, 2010
The Enterprise App Store

Benefits
- Reduced backlog
- Higher adoption
- Better IT solutions
- Improved quality
- Lower costs

Distributing SOA, SaaS, & Mashups with The Enterprise App Store

From http://www.ebizq.net/blogs/enterprise by Dion Hinchcliffe
Implications for Enterprises

- Provides a much-needed distribution model for mashups
- Makes it easy for apps to be built and shared by anyone
- Can greatly increase the reach of SOA, ESB, and open APIs
- Enables compelling internal business scenarios including chargebacks, innovation, high levels of reuse, better IT adoption, and improved agility
App Stores Unleash Developer, End-User, and Business Potential
Implications for SOA

• Up until now, SOA has had highly technical or ineffective distribution models to bring solutions & mashups to the masses

• Enterprise App Stores can offload the burden of leveraging SOA investment to the entire organization and make much of SOA self-service

• **Bottom Line:** Enterprise App Stores don’t require an SOA but are more effective and efficient if you have one
For More Info:

http://jackbe.com/enterpriseappstore

The app store: The new "must-have" digital business model

http://blogs.zdnet.com/Hinchcliffe/?p=1172

The Enterprise App Store And Self-Service IT: How SOA, Saas, And Mashups Will Thrive

http://www.ebizq.net/blogs/enterprise/2010/02/self-service_it_and_the_enterp.php

Wednesday, April 28, 2010