
Open Source Software at Orbitz, LLC

Leon Chism, Chief Internet Architect

Chuck Clark, Director of Platform Architecture

December 13, 2005

Goal

To understand the role open source software plays at Orbitz and how you can leverage our experience to increase your business and technical success.

Orbitz's Commitment to Open Source Software

- Where? In every environment
- What? Software at (almost) every layer of the stack.
- Who? All of
 - Orbitz.com
 - Cheaptickets.com
 - Travelport.com
 - Lodging.com
 - Orbitzforbusiness.com
 - Parts of American Airlines' website
 - Parts of United Airlines' website
 - Parts of Northwest Airlines' website
- Open source software has played an important role in the ability of Orbitz to compete successfully with entrenched companies that are much larger and much better funded.

Orbitz Overview

- Founded in 2000 by 5 leading airlines
- Site launched in 2001
- Top 3 brand travel site in just over two years
- Award winning air, car, and hotel products
- Public offering on 12/17/03
- Purchased by Cendant Nov. 2004

The screenshot displays the Orbitz website interface. At the top, the Orbitz logo is accompanied by the tagline "MOST LOW FARES MADE EASY". Navigation tabs include "Flights", "Hotels", "Cars", "Cruises", and "Vacations". A search bar is set to "Flight only" with options for "Round-trip", "One-way", and "Multi-city". The search criteria include "From" (ORD) and "To" (empty), with checkboxes for "Also search airports within 70 miles". Date selection is set to "Flexible dates - power search for savings (US & Canada)" with "Leave" on Dec 8 and "Return" on Dec 15. The number of travelers is set to 1 Adult, 0 Senior, 0 Youth, and 0 Child. A "Search" button is visible at the bottom of the search form.

On the right side, there are several promotional banners and sections:

- Weekend Hotel Specials from \$52** with a ">GO" link.
- A navigation bar with "My Deals", "Customer Service", "Travel Watch", and "My Stuff".
- A banner for **Disneyland Resort** featuring Mickey and Minnie Mouse, advertising "Play 2 Days FREE" and a "BOOK TODAY GO" button.
- A **DEALDETECTOR** banner with the text "Always searching for your target price - Activate".
- A **FLIGHT DEALS** section listing various deals with prices:

Deal	Price
Mexico Getaways!	From* \$212
Las Vegas: Win Big with Low Fares	\$244
London Is Now On Sale	\$224
Alaska Airlines: Coast-to-Coast Savings	\$144
Students Only- Airfare deals everywhere	\$86

Below the flight deals is a **OTHER TRAVEL DEALS** section with links like "Winter Escapes to Sunny Places" and "Big Savings in Las Vegas".

On the far right, there is a **SEE DEALS FOR** section with links for Florida, Caribbean, Europe, Hawaii, Mexico, Las Vegas, Holidays, United Kingdom, Gay travel, Ski travel, Golf, and Disney. Below this is a **ORBITZ CUSTOMER CARE** section with a "Travel Watch" link and a "Flying Forecast" section.

Business Context

- Orbitz launched 3 to 5 years after major competitors
- Air carriers saw trends driving distribution costs even higher.
- A notoriously competitive market with thin margins already
- Customer focused
 - Broad selection of lowest priced products
 - Features and functions that are relevant and easy to use
 - Fast feature delivery
 - Highly responsive and highly available web site
 - Customer care and service a key component
- Supplier friendly
 - Partnering with carriers
 - Reducing distribution costs
 - Long range commitments

Platform Priorities

- Speed to market
- Flexibility (both development and execution)
- Scalability
- Developer effectiveness
- Low cost (all else being equal)

Overview of Open Source Software at Orbitz

- Where? Every environment -- development, staging, production
- What? Software at (almost) every layer of the stack.
- Why? The best tools that support our business and technical needs
 - Reliable
 - Scalable
 - Malleable
 - Support model that fits our needs

Open vs. Closed

- Our primary concerns are
 - Reliability and availability (MTTR)
 - Developer effectiveness
- Likely profile
 - Commodity functions
 - Large, active development groups
 - Projects proven in production at scale
 - Software we already have the skills to support
 - Software we need immediate revisions and fixes for
- Support Model
 - Access to source is a higher priority than price
 - In some cases, we pay for support of open source software to augment our own efforts
 - As a culture, we are comfortable paddling our own canoe

Evolution of Open Source at Orbitz

- Linux has always been in production
- Based on preliminary performance and load tests, we migrated our application server OS to Linux
- The initial assessment was that we expected a 10x price performance improvement
- Post migration testing showed a 20x improvement
- Critical success factors
 - Java's multiplatform capability
 - Vendor support for chosen Linux distribution
 - Monitoring and management capabilities for a massively distributed system were already in place
 - Migrate non-critical systems first (a small bit at a time)
 - Testing testing testing

Team Dynamics

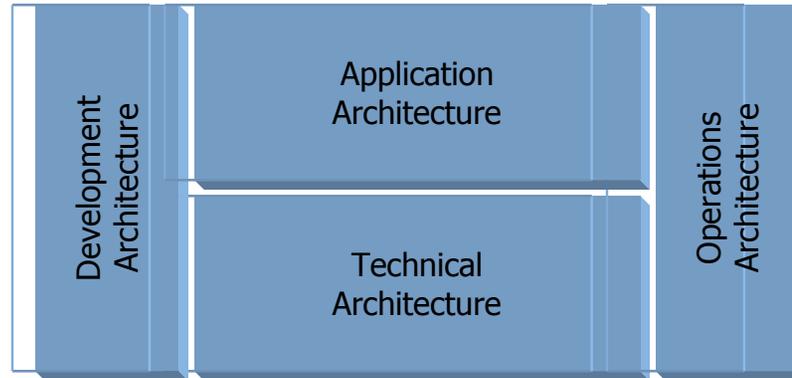
- Build a team capable of leveraging OSS
 - Risk management is a core competency
 - Architects and architectures capable of masking complexity and planning for change
 - Multi-language, multi-discipline resources
 - Naturally inquisitive developers
 - Motivated by the public recognition (and shame) that OSS development provides

“The [team] who doesn’t read [the source]...has no significant advantage over the [team] who can’t.”

The Development Landscape at Orbitz

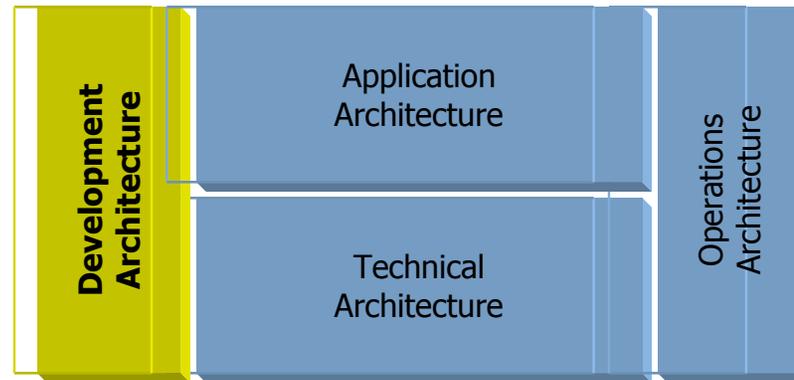
- More than 150 developers worldwide
 - Majority of development occurs in Chicago
 - Remote development sites in Denver, Israel and Bangalore
- Developers have state of the art Linux workstations
- Systems are primarily written in Java
- Major releases of the site occur every two weeks
- Millions of lines of code
- Code written in development and progresses through multiple environments until it reaches production with massive scale

Where Do We Use OSS at Orbitz?



- Open source software is pervasive throughout the architecture
- Linux and Java are used everywhere
- OSS libraries and tools in use have many origins
- The catalog of open source software used in Orbitz systems is constantly growing

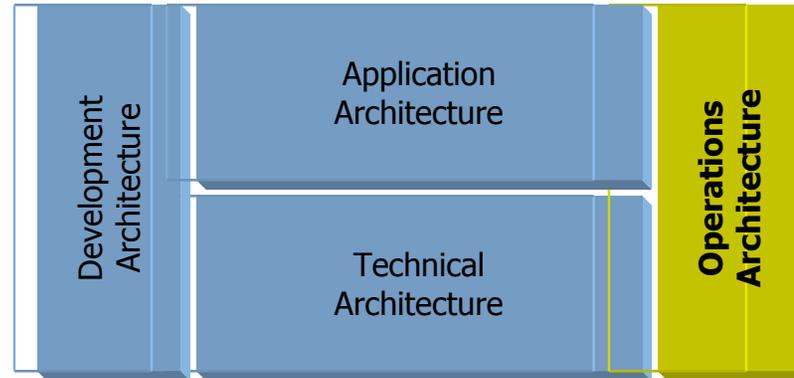
Where Do We Use OSS at Orbitz?



Allowing developers to create high-quality software effectively

- Apache Ant
- Cobertura
- Drupal
- FindBugs
- Jabber
- JUnit
- Jython
- Luntbuild
- MoinMoin
- Ruby

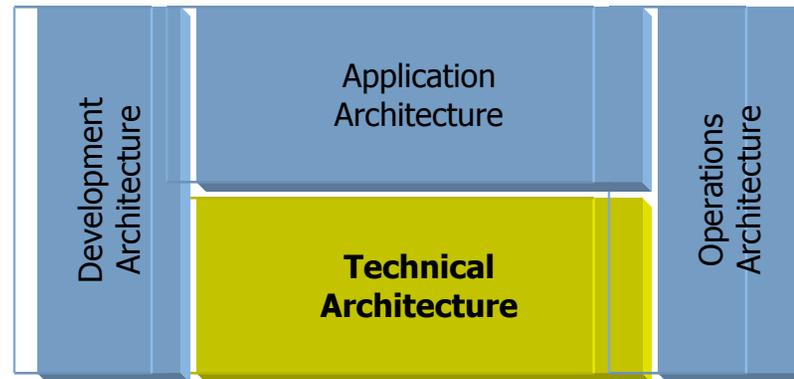
Where Do We Use OSS at Orbitz?



Enabling real-time monitoring and management of all production systems by a small team in the Orbitz Network Operations Center

- Cacti
- Distribulator
- Postgres
- Python
- R
- Red Hat Linux Workstation and Enterprise Server
- Round Robin Database (RRD)

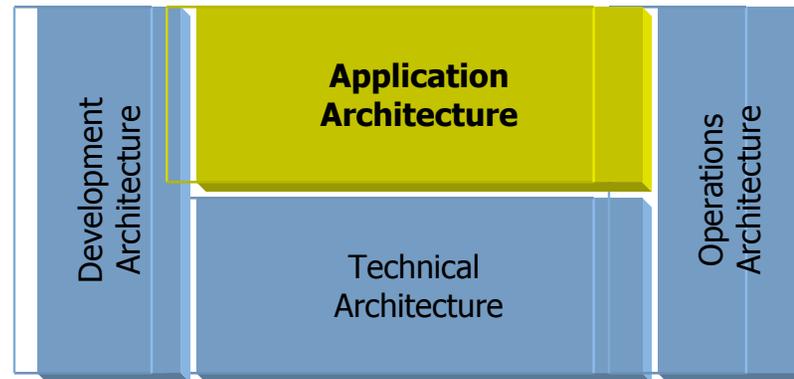
Where Do We Use OSS at Orbitz?



Frameworks and libraries not specifically related to the travel industry reused throughout the system

- Apache HTTP Server
- Apache commons-*
- codehaus Active MQ
- Drools
- Hibernate
- JBoss Application Server
- Spring Framework
- Sun Jini

Where Do We Use OSS at Orbitz?



Building industry leading websites and capitalizing on partnership opportunities

- Apache Axis
- Apache Jakarta Taglibs
- Apache Struts
- IBM International Components for Unicode
- Spring MVC

Openly Embracing Open Source Software

Project teams have authority to adopt and integrate new projects as necessary

- Identify a need and determine the requirements
- Survey the landscape of available products
- Choose a path:
 - Adopt and embrace an existing project
 - Blaze your own path
 - Build a proprietary solution for internal use only
 - Build and donate to the community
 - Start a new project
 - License a commercial package

Adopt an Existing Project

- Acknowledge de facto standards exist
 - Proven scalable, deployment track record
 - Online documentation and books
 - Faster developer ramp-up and distributed skill set
 - Focus on delivering business value not commodity code
- The 80/20 rule applies
 - Many projects have architectures to extend and customize to meet Orbitz specific needs
- Leveraging a large and active community around a project improves developer effectiveness
 - Bugs addressed in a timely fashion
 - Ensure project roadmap fits with our direction
- Contribute to and influence the future of the project
 - Interact with project owners and submit feedback
 - Contribute and become a committer

Blaze Your Own Path

Build a proprietary solution for internal use

- Why?
 - Some system components may be so critical to success it is too risky to not own and have full control over them
 - Candidate projects may not meet criteria
 - Easier to start fresh than “shoehorn” a poor fit
- Examples and Results
 - Orbitz Web Application Framework
 - Orbitz Build System
 - Full featured and flexible enough to meet *initial* needs
 - Open source alternatives continued to evolve and become more robust while Orbitz focused on business needs
 - Additional code to maintain or even worse *not* maintain

Blaze Your Own Path

Build a solution and release it as open source

- Orbitz Application Support team built an application to assist deploying and managing applications across hundreds of servers
- Released acceptable version 1.0 as open source
- Integrated community feedback with Orbitz requested improvements to deliver a much more robust version 2.0
- Bug reports, patches and features distributed across many users

Open source from the start

- Orbitz use and understanding of OSS is maturing to where this is now feasible
- Actively aware of new candidate projects inside the company

License a Commercial Package

Orbitz prefers but does not mandate OSS

- Recently migrated from CVS to AccuRev
- Subversion was the obvious replacement but is only an incremental improvement
- AccuRev has an innovative approach to source control deemed to best suit Orbitz needs:
 - change sets
 - atomic commits
 - minimal administrative overhead
 - a tool that works with the Orbitz release process
- But still needs to grow:
 - IDE plugins
 - Replication
- The choice is not final – continuing to watch projects like darcs, Monotone and arch

Risk Mitigation When Using Open Source Software

- Avoid using inactive or abandoned projects
- Be wary of adopting a project too early
- Awareness of projects with conflicting philosophies or heading in a different direction
- Inclination to upgrade to the latest RC, Alpha or Preview
- Evaluate and understand the ramifications of forking or modifying code without submitting patches

Process Optimization

- Planning
 - Account for the time required to evaluate projects
 - Plan for involvement in early stage projects
 - Actively manage the project portfolio
- Communications
 - Internally publicize the projects in use
 - Better communicate selection guidelines and criteria

As we continue to refine and expand our technology, open source software will play an increasingly prevalent role.

Observations and Unintended Benefits

- Orbitz has adopted many OSS processes and technologies
 - Wiki
 - Forums
 - RSS
 - Mailing lists
- Facilitates remote development
- Fast, iterative development
- Developers contribute to other projects in the company (sense of community code ownership)

Conclusions

- With a bit of planning, use and support of Open Source projects can be as good, if not better, than your experiences with closed source, commercial products
- Its isn't for everyone, or everything, yet, but its always worth considering
 - If you have the need then someone else does too
 - Don't under-estimate speed and effectiveness of the community
- Orbitz benefits from not only OSS products, but also OSS processes
- Open Source Software played a key role in the success of Orbitz, and it might just as important to your success

Questions?