

# Best of Breed

Eugene Ciurana

Open-Source, Scalability Evangelist  
CTO, CIME Software Labs



MARCH 24-25, 2009

# About Eugene...

- 15+ years of experience building mission-critical, high-availability systems infrastructure
- 12+ years of Java work
- Open-source evangelist
- Official adoption of open-source at Walmart
- Open-source active involvement since 1998
  - Contributor to Linux, Jetty, WeeChat, awk, Mule
- State-of-the-art tech for main line of business
  - Engaged by the largest companies worldwide
  - Retail
  - Finance
  - Oil industry
  - Background: industrial robotics to on-line retail systems



MARCH 24-25, 2009

# What This Presentation Is About

- Real life isn't binary
  - Open source and commercial software must coexist
- Decision-making must be data-driven
  - Don't buy into marketechtures
  - Don't let the open source zealots cloud your judgment
- Defining the cost:benefits ratio for your application
  - Time to market pressure
  - Licensing issues
  - Hiring people
- Using vs. participating in open source
  - Getting software from an open source vendor isn't any different than getting it from a commercial one if you're a user
  - Participation in the community is what makes it worthwhile



# What You'll Learn

- How to identify the best options when fulfilling a need
- How to define your evaluation criteria
- Define cost/benefits targets and validation tests
- Identify open source communities and open source vendors
  - They **ARE NOT** the same thing
- Why it is important to participate in the community
- Finding viable open source projects
  - Apache, GNU, the IRC universe, and more
- SLAs and open source
- Case studies
  - Online large retailer
  - LeapFrog Enterprises



# Best of Breed

- Best of Breed is the title given to the dog who has been judged the best representative specimen of its breed at a conformation show
- Dogs compete in a hierarchical fashion combining winners in sub-categories in new evaluations
- The winners in each class in each sex compete for Challenge (best) Dog and Challenge Bitch
- Each of these individuals will challenge one another for Best of Breed
- The runner up is selected Best of Opposite Sex
- The Best of Breed dogs and their progeny increase in value
- The selection process is a combination of objective and subjective criteria



# Best of Breed Software

- Best of Breed software is the application, middleware, or systems software that best solves a problem in your organization
- Evaluation is hierarchical, combining open-source and commercial offerings based on features vs. requirements
- The winners of each class compete as the best open source and the best commercial option
- Each of these will compete for your business as Best of Breed
- The runner up is put aside as a secondary option
- The Best of Breed software will help your organization increase its value and ROI
- The selection process is a combination of objective and subjective criteria



MARCH 24-25, 2009



# Open Source and Commercial Software

- Life isn't black and white
  - Commercial and open source systems must coexist
- You will find hybrid deployments at companies of all sizes
  - Do you run Exchange?
  - Does your company use Photoshop or Acrobat?
  - Do your applications run on Websphere?
- You may already be running open source software in your shop!
  - Software bundled in a commercial offering (e.g. Websphere)
  - Stealth deployments
  - Development tools
- Open source horizontal software works best for solving horizontal problems
  - Open source doesn't do so well in vertical applications because they require closer problem domain expertise



# Open Source and Commercial Software

## Avoid Marketectures

Don't let the vendors,  
commercial or open source,  
bamboozle you.



MARCH 24-25, 2009



# Decision-Making Must Be Data Driven

- Architectures are often confused with frameworks
  - In some cases, framework and architecture map 1:1
  - In other cases, vendors like to fan the confusion
- Is your architecture tied to a specific vendor or technology? Is this a good idea?
- Architectures are vendor- and technology-agnostic
- Lay out the architecture, then evaluate technology, and then figure out which products will become part of it
  - J2EE?
  - .Net?
  - Open source products?
  - Why?
- There is no silver bullet, whether open source or commercial software is involved



# Decision-Making Must Be Data Driven

- Larger enterprises sometimes don't benefit from using these tools
  - Corporate policy gets in the way
  - Fear of litigation
  - Aversion to being first
  - Inertia
  - Vendor lock-in
- The smart companies, though, figure out ways of implementing these
  - Pilot programs
  - Partnerships
  - Shift litigation through open-source subscriptions and indemnification

# Decision-Making Must Be Data Driven

- How do you go about changing corporate policy?
  - Through education!
    - Most large IT organizations are too busy with day-to-day business to pay attention
    - Many lack, or forego, ongoing formal training for their engineering and engineering management groups
- Remember: **vendors have their own best interest at heart!**
- Seek education from outside sources/third-parties who are in that business, not in the business of selling software
  - Open Source Initiative
  - Trade magazines
  - WWW: InfoWeek, TheServerSide, Slashdot, Reddit, java.net, Digg, etc.



# The Best Open Source Software

- So... how do you go about finding the best open source software?
  - Hint: More often than not, it DOESN'T come from your vendor
- The best places to find open source software are on the Internet
  - Ask your developers
  - Participate in the communities like Reddit, TheServerSide, DZone, Codehaus, etc.

**The best open source software  
ALWAYS  
has a thriving community around it!**



# The Key to Success Using Open Source

- Understand that there are no open source products
  - That's vendor-speak
- Open source is about projects
  - Projects are continuously evolving
- The only way to maximize your ROI is by participating in the community
  - Active participants get attention faster
  - Active participants can influence the direction of the project
  - Active participants get bug fixes faster
  - You make friends with the developers, project managers, etc.
- Getting professional services from a big vendor for open source is dumb
  - They often hire some guy off the street, cram the manuals in, and send him to help you charging you muchos \$\$\$ for things he may not understand
  - No street cred in the project -- slow turnaround in problem solving



# Who and What to Avoid

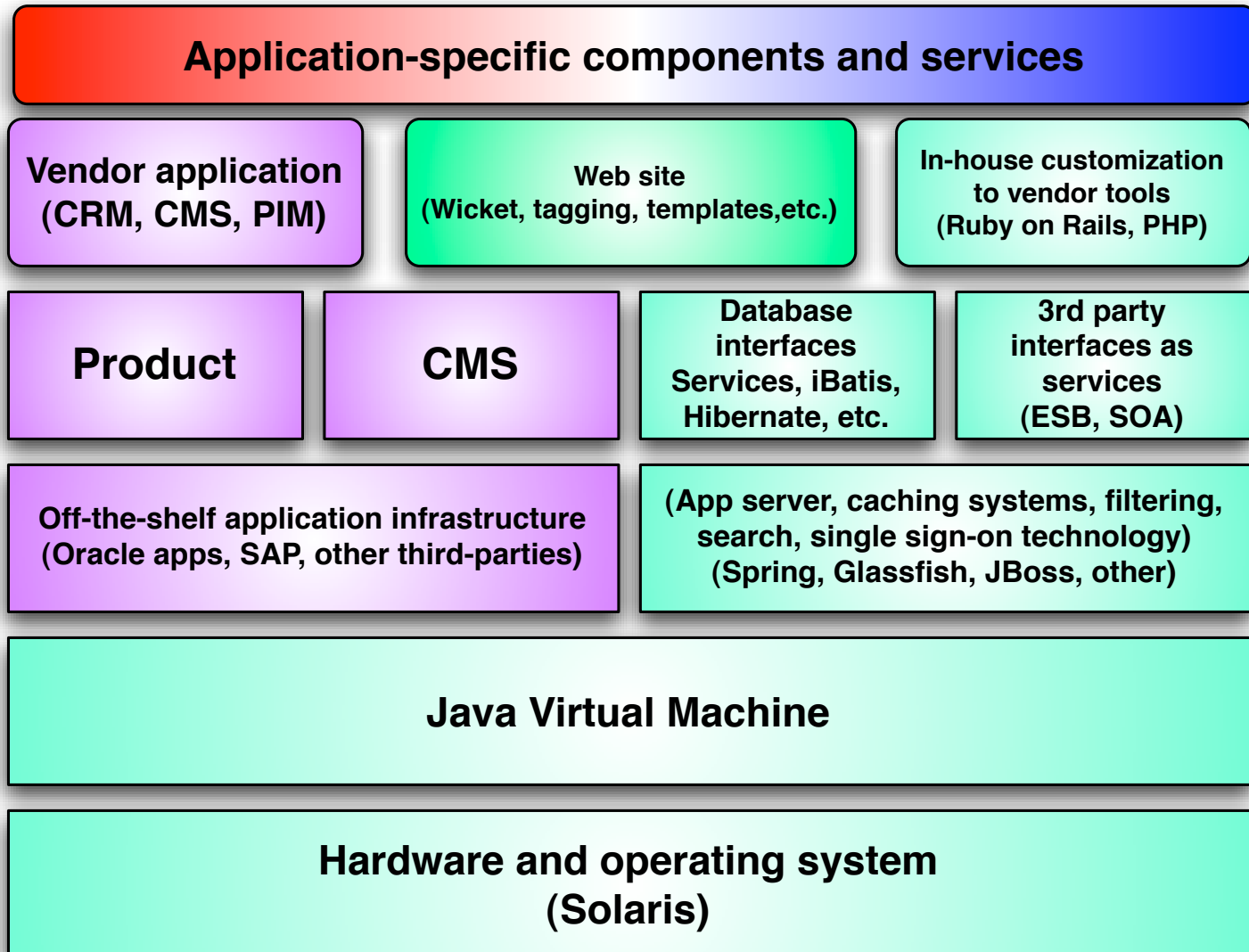
- Avoid the zealots
  - Not all open source projects are good
  - Not all commercial software is bad
- Avoid engaging projects without a thriving community
- Be suspicious of commercial software turned open source
  - Was it not selling?
  - Are there licensing traps associated with it?
- Avoid all-or-nothing migration strategies
  - Existing/legacy systems implement your company's know-how
  - Define an integration strategy instead that slowly replaces old/expensive commercial or in-house systems with open source equivalents
- Remember that open-source works best for satisfying infrastructural requirements
  - Operating systems, middleware, caching systems, services, dev tools





# Solving the Integration Problem

- Commercial software addresses domain-specific problems when it's cheaper to acquire than to build in-house





# Defining the Cost:Benefits Ratio

- Do you have time-to-market pressures?
- Assume that the professional services cost is equivalent for open source projects and commercial software
- The ratio will vary on different subsystems
  - Infrastructural pieces are better understood and have more tools
    - They require less attention
  - Application platforms require more professional services
    - Web front end is the most time-consuming
    - Many open source options
  - Integration with commercial and legacy systems is paramount
    - Use integration platforms instead of trying to integrate point-to-point
    - Establish integration objectives with your vendors in the statement of work
  - Use the service level agreements within the organization and with the vendors to define your cost:benefits ratio

# Defining the Cost:Benefits Ratio

- Professional services from commercial vendors suck
  - Avoid the “big names” that make you feel “safe”
    - These guys often don’t know what they’re doing
    - They have a much higher cost
- Engage the project participants for your professional services!
  - These guys know the internals of the project better than anyone
  - They offer lower rates (hour, project) than the “big names”
  - They’ll take less time to build your product or service
- Find your employees and augmented staff through the community!
  - Project community sites
  - Open source community events like users groups
  - Internet Relay Chat channels (Freenode)

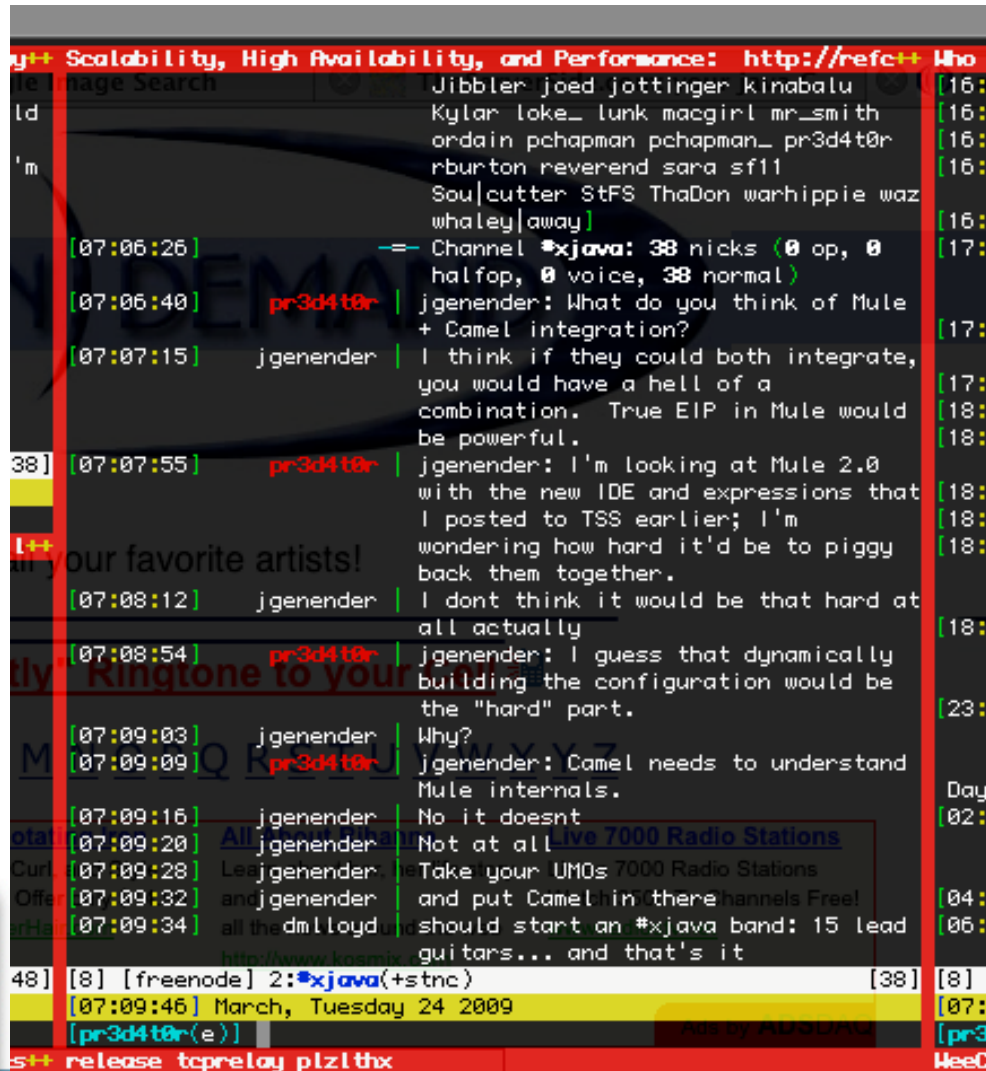
# IRC - The Secret Weapon

- Internet Relay Chat (IRC) channels in networks like Freenode (<http://freenode.org>) are some of the best community places to participate and learn
- It's a bit like the wild west - IRC channels are meritocracies
  - Ask a smart question, get a smart answer
- Every major and minor open source project has an IRC channel somewhere
  - That's where a geographically dispersed development team can meet, discuss issues, socialize, and interact
  - Some projects have multiple channels that accommodate both developers and users
- Give your developers IRC access!
  - IT departments tend to block it
  - Smart developers figure out ways around such silly blocks



# IRC - The Secret Weapon

- IRC is the most plentiful talent pool for open source projects



```
Scalability, High Availability, and Performance: http://refc++ Who
Image Search
ld
'm
[07:06:26] Channel #xjava: 38 nicks (0 op, 0 halfop, 0 voice, 38 normal)
[07:06:40] pr3d4t0r jgenender: What do you think of Mule + Camel integration?
[07:07:15] jgenender I think if they could both integrate, you would have a hell of a combination. True EIP in Mule would be powerful.
[07:07:55] pr3d4t0r jgenender: I'm looking at Mule 2.0 with the new IDE and expressions that I posted to TSS earlier; I'm wondering how hard it'd be to piggy back them together.
[07:08:12] jgenender I dont think it would be that hard at all actually
[07:08:54] pr3d4t0r jgenender: I guess that dynamically building the configuration would be the "hard" part.
[07:09:03] jgenender Why?
[07:09:09] pr3d4t0r jgenender: Camel needs to understand Mule internals.
[07:09:16] jgenender No it doesnt
[07:09:20] jgenender Not at all
[07:09:28] Lea jgenender: Take your UMOs 7000 Radio Stations
[07:09:32] and jgenender and put Camel in there channels Free!
[07:09:34] all the dmlloyd should start an #xjava band: 15 lead guitars... and that's it
[07:09:46] March, Tuesday 24 2009
[pr3d4t0r(e)]
s++ release taprelay plz!thx
```

- Conversations are both technical and social
- The most important thing to keep in mind: these guys are there because they believe in the project and like what they do!
- About \$5M have gone through that IRC channel since 2004!



# Case Study: Large Internet Retailer

- One of the largest web presences based on open source software
  - Tomcat
  - Java tools/Struts
  - Mule ESB
  - Linux and Open Solaris
- Parent company, bricks and mortar, preferred commercial products and services
  - Open source was non-existent for line of business applications
  - Main concerns were not the software
    - Licensing
    - Intellectual property
  - Blue shop - the Smurfs were running all over the data center
- Tensions between on-line retail and the rest of the company because of the differences in stack, skills, etc.



# Case Study: Large Internet Retailer

- Adoption of open source for main line of business became a priority
  - Sponsored by the CIO and CTO
  - Active participation from the on-line and traditional IT groups
- The issues were not technological
  - Inertia: the Smurfs saw license revenue decreases and FUDed
  - Education on licensing, development, tech was paramount
  - Data driven: demonstrated that open source software was better
    - Clear metrics for “better”
    - Tested in a variety of commercial and open sources OS, hardware, app servers and hardware from all major vendors
    - Data driven claims validation: Trust.... but verify
- Time line
  - Kick-off: April 2006
  - Proof of concept: November 2006
  - Open source companywide policy: January 2007
  - Production on-line retail: January 2009





# Case Study: Large Internet Retailer

- Best of Breed Environment
- Commercial software
  - Database
  - Order management system
  - Order capture / e-commerce system
- Open source
  - Enterprise integration (Mule)
  - Application servers (Tomcat)
  - All development tools
  - Operating system (Solaris, Linux)
  - Web application server stack (Apache Wicket)
- Notice something?
- The Best of Breed environment uses some standards, but not everything has to be standard!





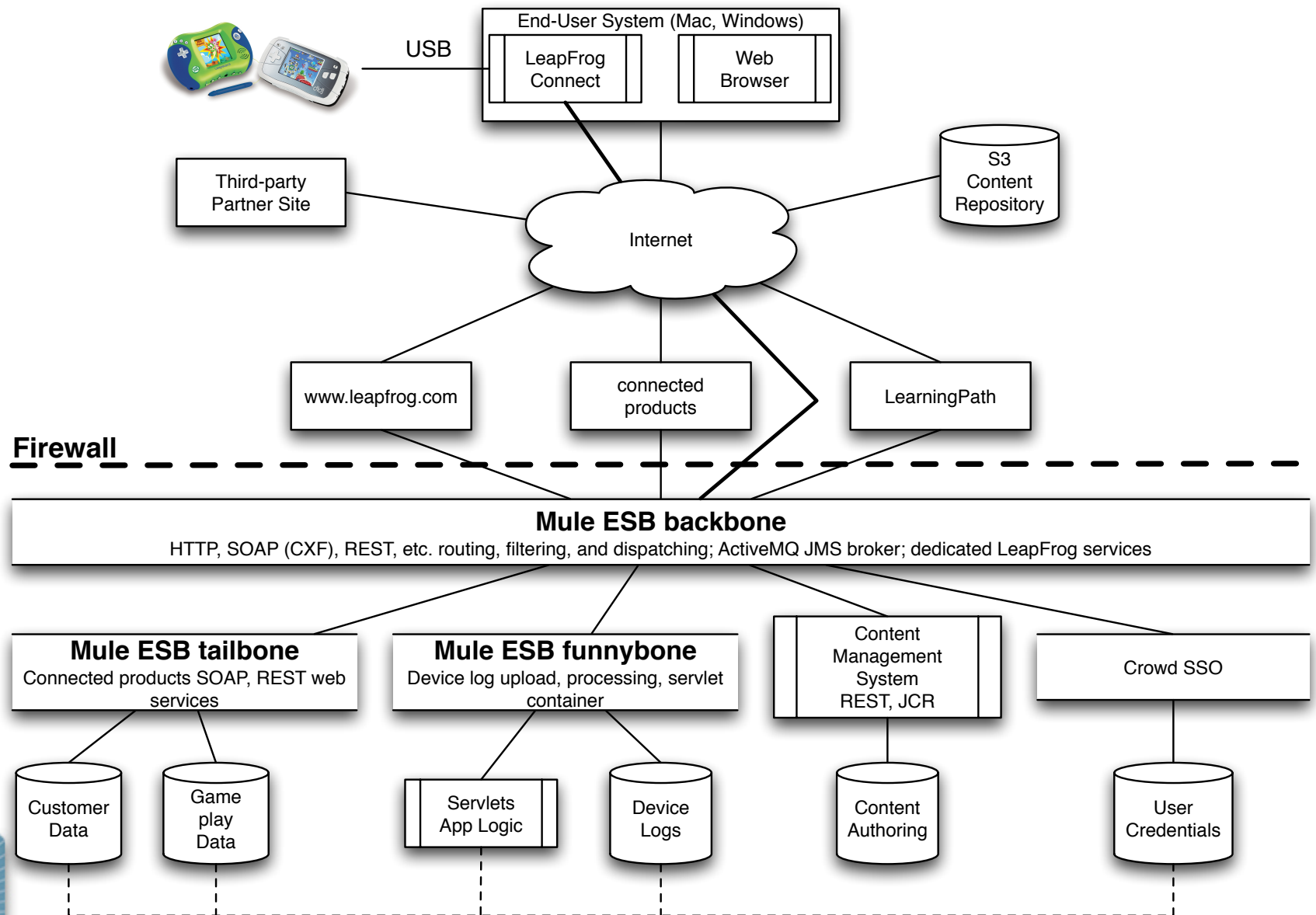
# Case Study: LeapFrog Enterprises

- The leading producer of innovative, technology-based learning products worldwide (six languages, 35 countries)
- At home and in schools (over 100,000 classrooms US)
- Titles: phonics, reading, writing, math, music, social studies, geography, spelling, vocabulary, science, more
- Millions of devices connected to the SOA infrastructure



<http://www.leapfrog.com/learningpath>

# Case Study: LeapFrog Enterprises



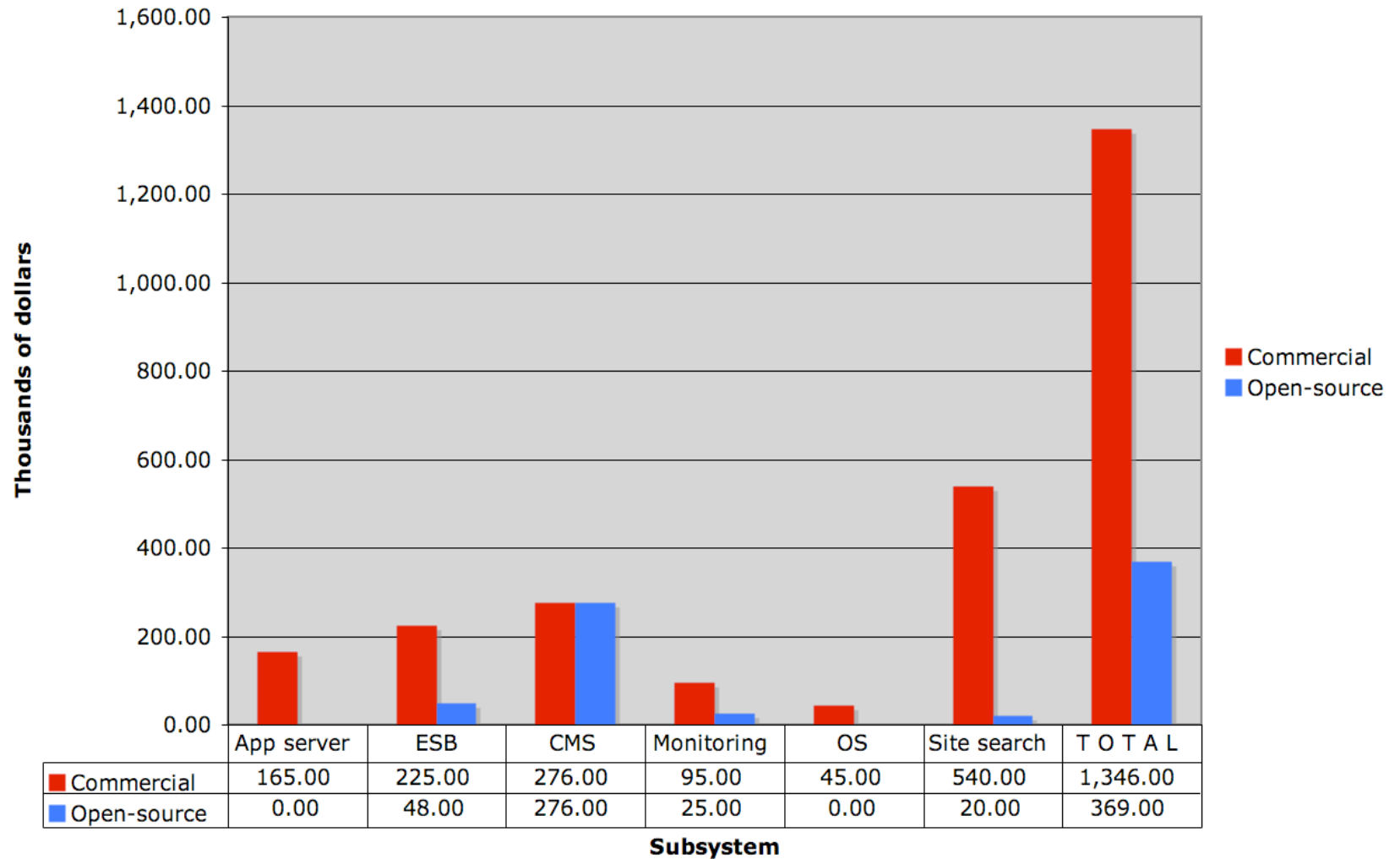
# Case Study: LeapFrog Enterprises

- Best of Breed Environment
- Commercial software
  - Commercial CMS
  - Commercial SaaS e-commerce suite
  - Commercial database
- Open source software
  - Mule ESB
  - Tomcat app servers
  - Apache Wicket
  - Maven and all open source development tools
  - Continuous integration
  - Operating systems (Linux)
- Staffed: 45 developers
  - 20 employees
  - 25 open source project contributors



# Case Study: LeapFrog Enterprises

## Cost Comparison Web Systems



# Q&A

Thanks for coming!

This presentation is at:  
<http://eugeneciurana.com>

Eugene Ciurana  
Open-Source, Scalability Evangelist  
CTO, CIME Software Labs

Contact: <http://eugeneciurana.com/contact>



MARCH 24-25, 2009