Building Eudoxus.gr with FOSS and Open Standards

Fotis Stamatelopoulos (@fstama)

ΕΛ/ΛΑΚ Conference 2011
May 21, 2011
What is Eudoxus

- A distributed system that supports and streamlines processes and operations related to the textbooks distribution for the higher education institutions of Greece


- Developed, operated and supported by GRNET http://grnet.gr/

- Users: Students, Secretariats, Publishers, Distribution Points (e.g. Bookstores), related Ministry depts, Courier service, Helpdesk- More than 270,000 individual users

- Multiple APIs and integration with other ISs

http://eudoxus.gr
Presentation Focus

- Design and software architecture
  - by design based on 100% FOSS infrastructure
  - adopting open standards

- Building within a very tight schedule and ever changing requirements
  - agile methodology
  - FOSS development tools

- Evaluating the results of the 100% FOSS decision (a year later)
Design Goals - Challenges

- Tight schedule, fixed milestones - deadlines
  - Incrementally release modules in production
- Not finalized requirements
  - Be flexible, handle changing requirements
- High availability
  - Redundant architecture
  - Live application updates
- Impossible to predict usage load
  - Scalability, elastically use infrastructure to serve growing usage
- Safeguard the transactional nature of related processes
- Synchronize data and connect to other systems
Technology decisions

- Web-based user interfaces
- RESTFul APIs
  - use SOAP-based web services for integration in specific cases (e.g. courier service)
- Java-based core (business logic tier)
- Javascript-based / GWT-based rich GUIs, AJAX
- Back-end storage: RDBMS vs noSQL decision
  - noSQL options offer better scalability than typical RDBMs
  - Needed at least a minimal transactional core - eventual consistency is not acceptable
  - Dropped the initial hybrid approach - full RDBMS design
- Shibboleth-based authentication for students
High level architecture

Browser App (AJAX)

Browser App (AJAX)

Browser App (AJAX)

web cache (html, css, images, REST responses)

Business logic tier

Horizontally scalable, Stateless business functions

ORM, second level caching

storage / back-end tier

user state stored in the browser app
Infrastructure Software (100% FOSS)

- Server-side hosted on GRNET's cloud infrastructure (debian VMs)
- Google Web Toolkit (GWT) for the rich GUI clients (browser)
- Apache, nginx front-ends
- Varnish web cache
- JBoss Application Server
  - stateless EJBs implementing core business logic
  - JPA ORM (hibernate substrate)
  - JMS Queues and Message Driven Beans
  - JMX MBeans for scheduled tasks and maintenance
- PostgreSQL RDBMS
- Solr (Lucene server) indexing / search server
- A multitude of FOSS libraries and frameworks
Meeting Tight Schedules

- Typical setup
  - VCS (mercurial), wiki for requirements, issue tracker
  - dedicated release engineer, dedicated DBA
  - local development -> test server -> production server
  - FOSS dev tools (Eclipse, ant, etc)
- Agile approach
- Documentation was the first victim of the pressing schedule :
- Updated unit tests was the second :
- A few statistics on the team
  - ~10 developers
  - ~10 testers
  - ~20 helpdesk members for end-user support
  - Full GRNET NOC support!
Eudoxus & FOSS: a year after

- A multitude of mature FOSS libraries and frameworks to try and select from. This lead to choosing optimal (or near optimal solutions) for the task at hand

- Very efficient and flexible development environment 100% based on FOSS

- Eudoxus proved (again) that you can successfully built and operate a large scale, business critical, high availability, distributed information system based 100% on FOSS software
http://eudoxus.gr

linkedin.com/in/fstamatelopoulos

@fstama