Doodle
powered by Java^{TM}

Paul E. Sevinç, Dr. sc. ETH

Co-founder and CTO

Doodle AG

doodle.com

The problem

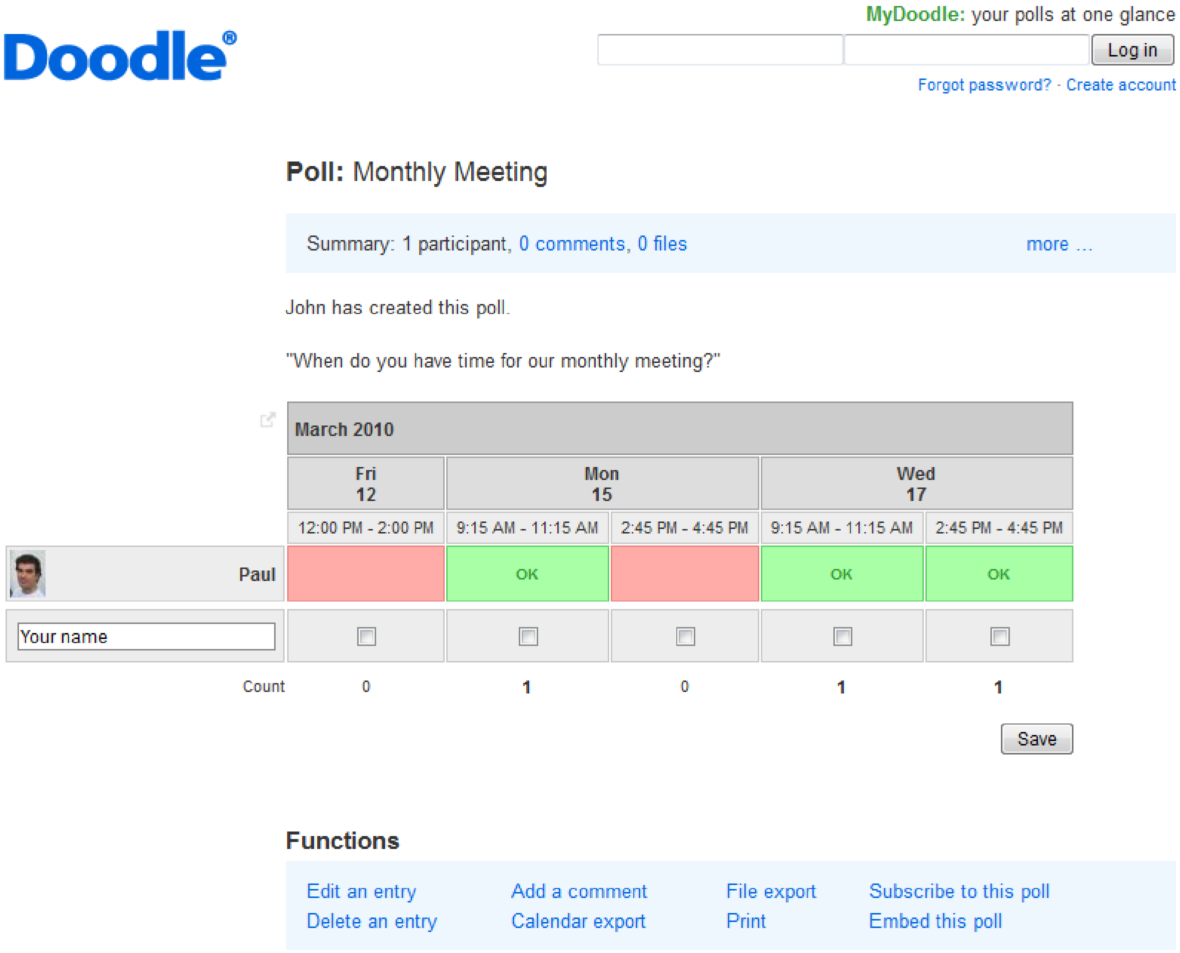
The solution

The advantages

The problem

- A group of people needs to agree on a date/time
- A group member (or her assistant) could coordinate by e-mail, SMS, phone, etc.
  - Tedious
  - Time-consuming
  - Error-prone

The solution



MyDoodle: your polls at one glance

Poll: Monthly Meeting

Summary: 1 participant, 0 comments, 0 files more ...

John has created this poll.

"When do you have time for our monthly meeting?"

	March 2010				
	Fri 12	Mon 15		Wed 17	
	12:00 PM - 2:00 PM	9:15 AM - 11:15 AM	2:45 PM - 4:45 PM	9:15 AM - 11:15 AM	2:45 PM - 4:45 PM
Paul	🟥	OK	🟥	OK	OK
Your name	□	□	□	□	□
Count	0	1	0	1	1

Functions

Edit an entry
Delete an entry
Add a comment
Calendar export
File export
Print
Subscribe to this poll
Embed this poll

The advantages

KISS

- Keep it simple, Sevinç!

Lightweight

- No registration
- No installation
- No cost

International

- More than 25 languages
- Language-dependent date and time formats
- Time-zone support

Engineering & Operations

Architecture

Back end

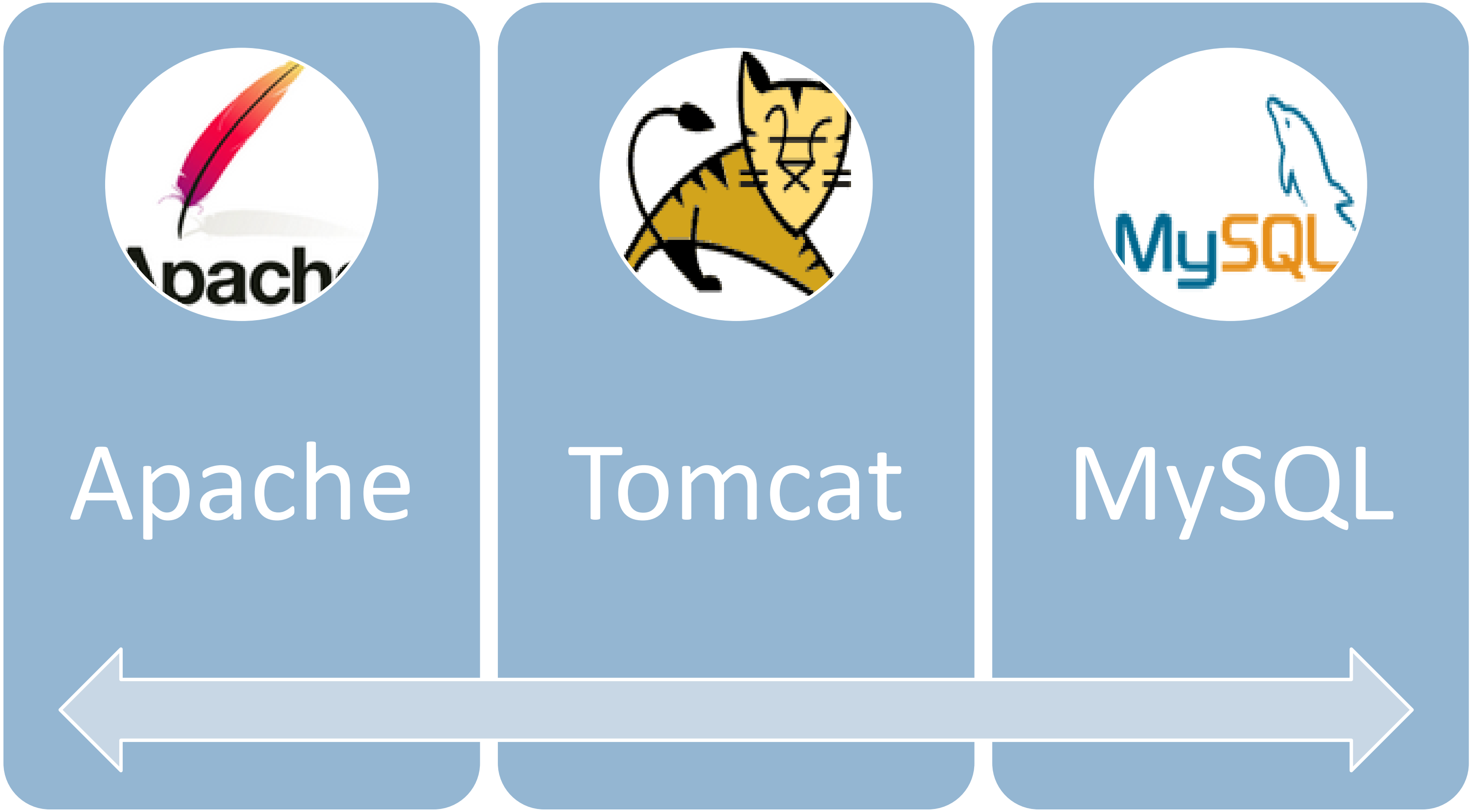
Front end

Name dropping

Tools

Books

Architecture: conceptual



Apache

Tomcat

MySQL

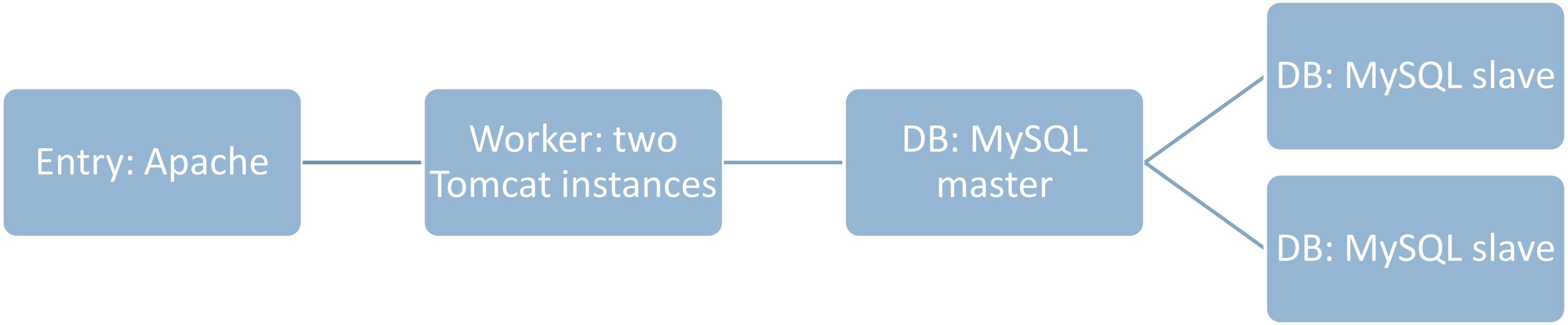
Web server → App server

Today (since 2007): Tomcat 6

Tomorrow: Glassfish 3

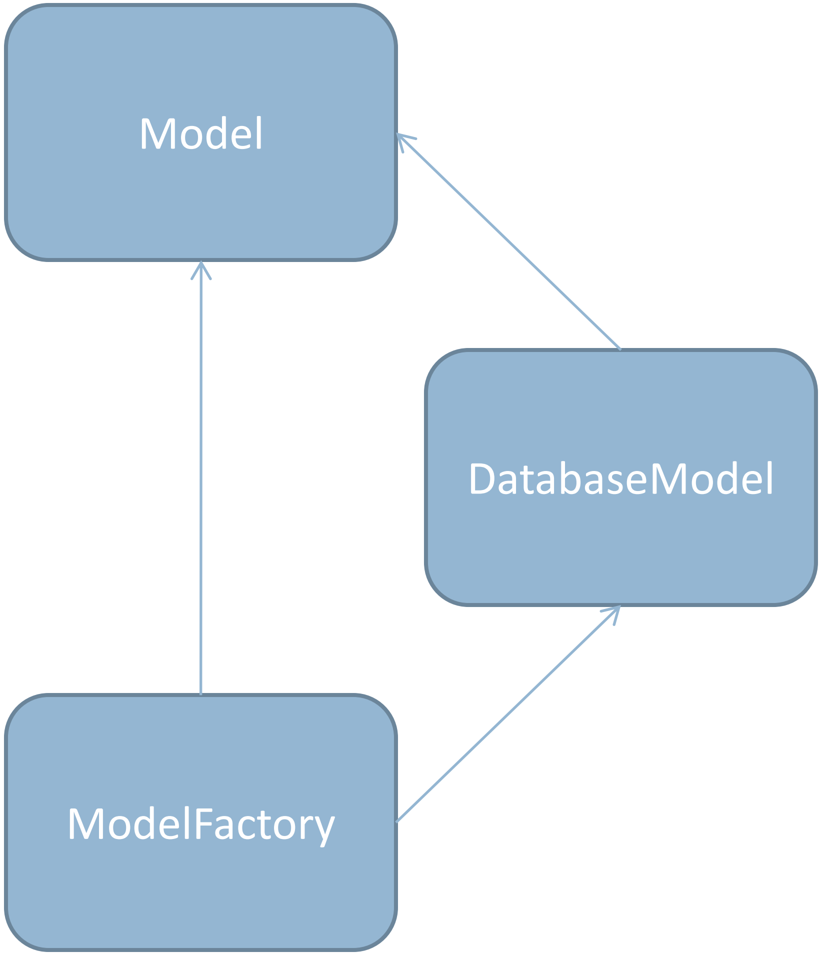
Or would you recommend JBoss 6?

Architecture: physical



By the way, the servers are in Zurich, not in the cloud.

Back end: model

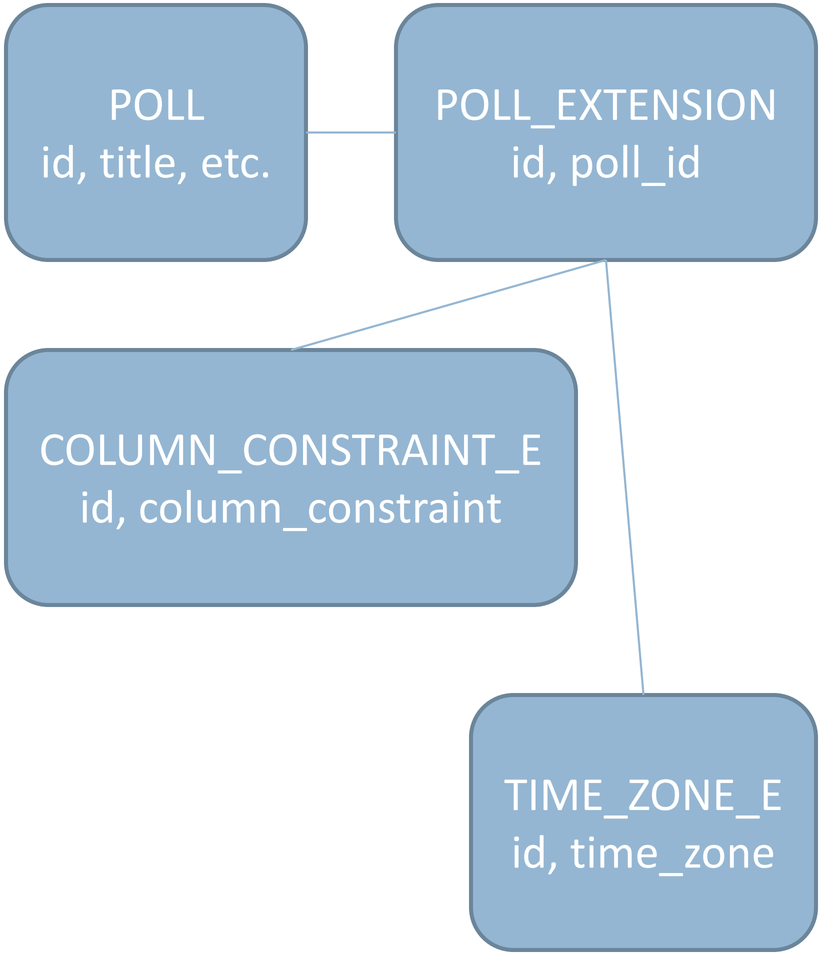


- Doodle is organized in modules, three of which are
  - the model,
  - the database model, which depends on the model,
  - and the model factory, which depends on the model and the database model (and consists of only one class without any run-time dependency injection).
- The other modules depend on the model and the model factory, but not directly on the database model.
- The model features interfaces (e.g., Poll) and value-object classes (e.g., EmailAddress).

Back end: DB model—Hibernate

- The database model is implemented using Hibernate, originally version 3.2 and today 3.3.
- In September 2009, we made the switch from mapping files (e.g., DatabasePoll.hbm.xml) to annotations *and stopped worrying about bi-directional relations*.
- At some point, we accidentally disabled the connection pool (when we updated from 3.2 to 3.3 in 2008?)—*and seriously considered not enabling it anymore*.

Back end: DB model—schema



- When we designed the database schema, we allowed for future requirements (an example being row constraints) to be met by means of an extension mechanism.
- That was overkill.
- Adding a column to a table is not a problem (at least with MySQL), even if the table contains maaany rows.

Front end: human web

- HTML is generated with JavaServer Faces, originally MyFaces (until JSF 1.2) and today Mojarra (Glassfish, since JSF 2.0).
- JSF in general and Facelets in particular are great, but are inherently stateful.
- Until recently, we did not require any JavaScript support and handled user interactions on the server.
- We now believe that it is safe to assume that JavaScript is supported by and large and have made the strategic decision to use jQuery.
- JSF vs. jQuery?

Front end: programmable web

- Version 1 of our API is RESTful, is based on a plain Java servlet, exchanges XML, and uses OAuth.
- The Doodle iPhone app is our API success story.
- Version 2 of our API will still be RESTful, will be based on Jersey (Glassfish), will exchange JSON, and will still use OAuth.
- http://doodle.com/about/APIs.html

Namedropping

- iCal4J → iCalendar 👎
- iText → PDF 👍
- JavaMail → SMTP 👍
- POI → Excel 👍
- etc.
- PS: Look, ma', no Spring!

Tools: RT

- Doodle's ticketing system is RT.
- Support is provided by Doodle's CEO, CTO, product manager, operations engineer, and software engineers.
- Listen to your users and customers:
  - Good support for your users and customers is good research for you.
  - *"Everything is resolved...thanx so much for your help...that is, if you're a real person and not a computer generated response."*
- Understand their problem, not necessarily their solution:
  - *"If I had asked people what they wanted, they would have said faster horses."* Henry Ford

Tools: Subversion

- Doodle's revision control system is Subversion.
- The entire company uses Subversion, not only Engineering & Operations.

Tools: Ant

- Doodle's build system is based on Ant.
- Today, I would probably choose Maven over Ant.
- But there is no compelling reason to migrate the build system from Ant to Maven given that it meets our requirements and that the Ant project is still alive (version 1.8 was released in February 2010)—or is there?

Tools: Eclipse

- In theory, every Doodle engineer could develop with the IDE of his choice.
- In practice, every Doodle engineer chooses Eclipse.

Tools: Basecamp

- Doodle's project-management system is 37signals' Basecamp.
- We even use Basecamp for what little bug tracking we need to do.
- We seem to have outgrown Basecamp, though, and are thinking about going http://www.atlassian.com/starter/
- PS: For my personal to-do lists, I swear by http://www.rememberthemilk.com/

Tools: Hudson

- Doodle's continuous-integration server is Hudson.
- In 2008, I decided to do continuous integration at Doodle. In 2010, we finally took the one day it only takes to properly set up Hudson. (I don't know what took us so long other than that we never have had serious QA issues.)
- Take advantage of Hudson plug-ins such as FindBugs.
- Next major goal: continuous deployment

Books

- We have a technical library of 100+ books
- Mandatory reading for all Doodle engineers
  - Joshua Bloch. Effective Java. 2nd edition
  - Douglas Crockford. JavaScript: The Good Parts. 1st edition
  - Joel Murach. Murach's Java Servlets and JSP. 2nd edition
  - Gunther Popp. Konfigurationsmanagement. 3rd edition as well as http://www.km-buch.de/downloads/Ant-Kapitel.pdf
  - Steve Souders. Even Faster Web Sites. 1st edition
  - Steve Souders. High Performance Web Site. 1st edition
  - Andreas Vdovkin. jQuery - kurz & gut. 1st edition
- Recommended reading for all Doodle engineers
  - Joshua Bloch and Neal Gafter. Java Puzzlers: Traps, Pitfalls, and Corner Cases. 1st edition
  - Karl Eilebrecht and Gernot Starke. Patterns kompakt: Entwurfsmuster für effektive Software-Entwicklung. 3rd edition
  - Antonio Goncalves. Beginning Java EE 6 Platform with GlassFish 3. 1st edition
  - Leonard Richardson and Sam Ruby. RESTful Web Services. 1st edition
  - Leon Shklar and Rich Rosen. Web Application Architecture: Principles, Protocols and Practices. 2nd edition
  - Stefan Tilkov. REST und HTTP. 1st edition
  - Nicholas C. Zakas. Professional JavaScript for Web Developers. 2nd edition

And now for something completely different …

- *"Here's a tip from someone who has read thousands of resumes. When you're applying to a startup, or a software company with less than, say, 100 employees, you may want to highlight the Banging Out Code parts of your experience, while deemphasizing the Middle Management parts of your experience."* http://www.joelonsoftware.com/items/2009/01/02b.html

Keeping it simple since 2003

+ Christoph + Benjamin
- Oliver + Oliver