REST - The Web Style

Koranteng Ofosu-Amaah
Lotus Software, IBM

Working the Web

- REST - The Web Style
- Buzzwords (Marketing)
- Case Studies (Reflection)
- Leaky Abstractions (Architecture)
- Layer Stripping (Complexity)
- Glue Layer People (Composition)
- View Source Imperative (Comprehension)

Life on the Web

- The Low End Theory (Strategy)
- Why Specs Matter (Scaling)
- Technical Arteriosclerosis (Hygiene)
- Jokes (Relief after yesterday's events)
- Spaghetti Westerns (B-Movies)
- Bedtime Stories and Folktales (It's Friday)

Why this presentation?

- 10 years since the Web became Good Enough (TM) - Historical Insight
- 10 years of Moore's Law in the Datacenter
- In times of retrenchment, companies fall back on things that "just work"
- The Web Style works - Google/eBay/Amazon
- The Web Style scales - businesses run on it
- Technology Adoption and Systems Design
- Leverage is key for success
- We should aggresively leverage the web

Who am I?

- 10 years at Lotus
- Currently working on Forms
- Freelance Graphics - eSuite - K-station - WebSphere Portal - ODC editors - Forms in Workplace
- Opinionated, "ecletic", "all over the place"
- Verbose: writes "novel-length diatribes"
- of late there's a blog - Koranteng's Toli
  - **toli**: n. 1. A juicy piece of news. 2. The latest word or gossip. 3.The talk of the town, typically a salacious or risque tale of intrigue, corruption or foolishness.
- Remembers the day job (Hi there managers)
- Self image: mild-mannered Clark Kent type

First Questions

- Do you exist if you're not "on the web"?
  - Are any of our products and services not web-native or delivered via the web?
    - Other than the Green Terminals of course
    - Even those *mainframe people* are dying to "get on the web"
    - **What is the Web in WebSphere?**
- **Everyone** got the web religion circa 1995
- How did ***you*** get on the web?
- Why was the web so successful?
- What does it mean to be "on the web"?
- When was the last time you read about the web architecture?
- Word to the wise: Don't fight the web

A quote

- Oliver Hass, a 28 year-old chemist and graduate student from Oldenberg, Germany, wrote me recently about what the President's trip looked like to him. In introducing himself, Hass commented on

"how necessary it can be for a chemist to forget about molecules and think about real problems."

We're all wrapped in our current black boxes
What are we trying to fix or to produce?

Big picture

- My Grandma (in village without electricity or running water) suddenly wants to get "on the web" (see my photos, see what I'm up to, email me etc)
- The Mom Factor in software = Opportunities
- Communication (**People**)
- **Communities** (We're social beasts)
- **Processes** (pesky paperwork or "workflow")
- Commerce $$$ (e-Business Things)
- People, Places and Things (LDS / K-station)
- People, Processes and Things (Workplace)

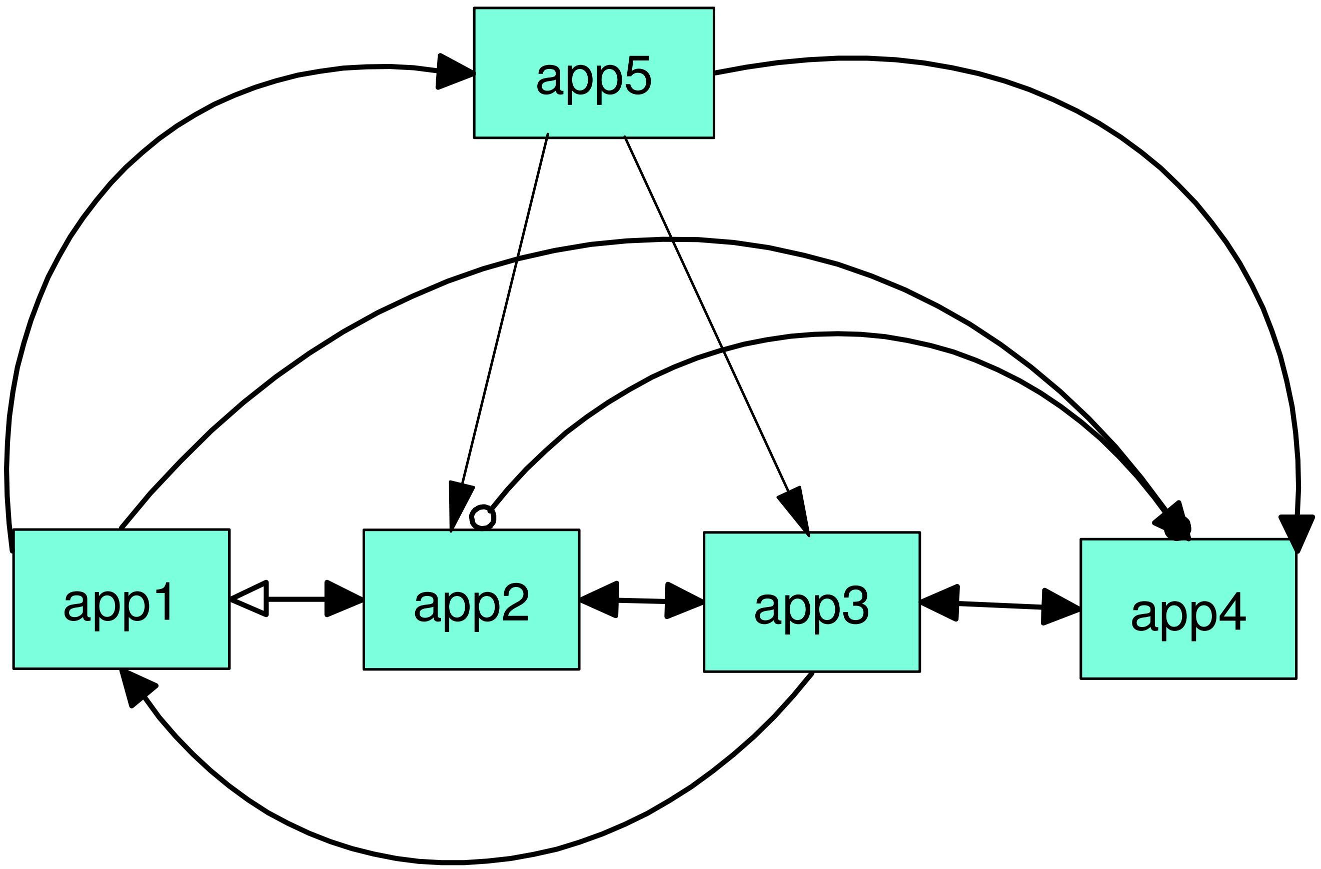
Some source material

- The Web style is mostly about **Resource Identification** and **Hyperlinking**
- Pointers to some source material
  - ► Roy Fielding - Designing a new protocol for the web
  - ► Sam Ruby - Attractive Nuisance
  - ► Adam Bosworth - Web of Data
  - ► Sam Ruby - Radical Simplification
  - ► Google for REST, Temple of REST, REST Wiki
  - ► http://del.icio.us/tag/rest
- A REST Intervention
  - ► http://koranteng.blogspot.com/2005/03/rest-intervention.html

Architecture of the World Wide Web

- The web has an architecture
- Really
- Truly
- Berners-Lee had the prototype
- Contact with the real world turned it into the product
- **The web was designed and architected**
- Roy Fielding - Apache, REST style
- Netscape, Mosaic, Microsoft, Sun, Apple, IBM etc.
- **The Usual Suspects** plus some new **Cats**

Enterprise Application Integration



4 applications = 6 integrations

5 applications = 10 integrations

6 applications = 15 integrations

From Roy Fielding

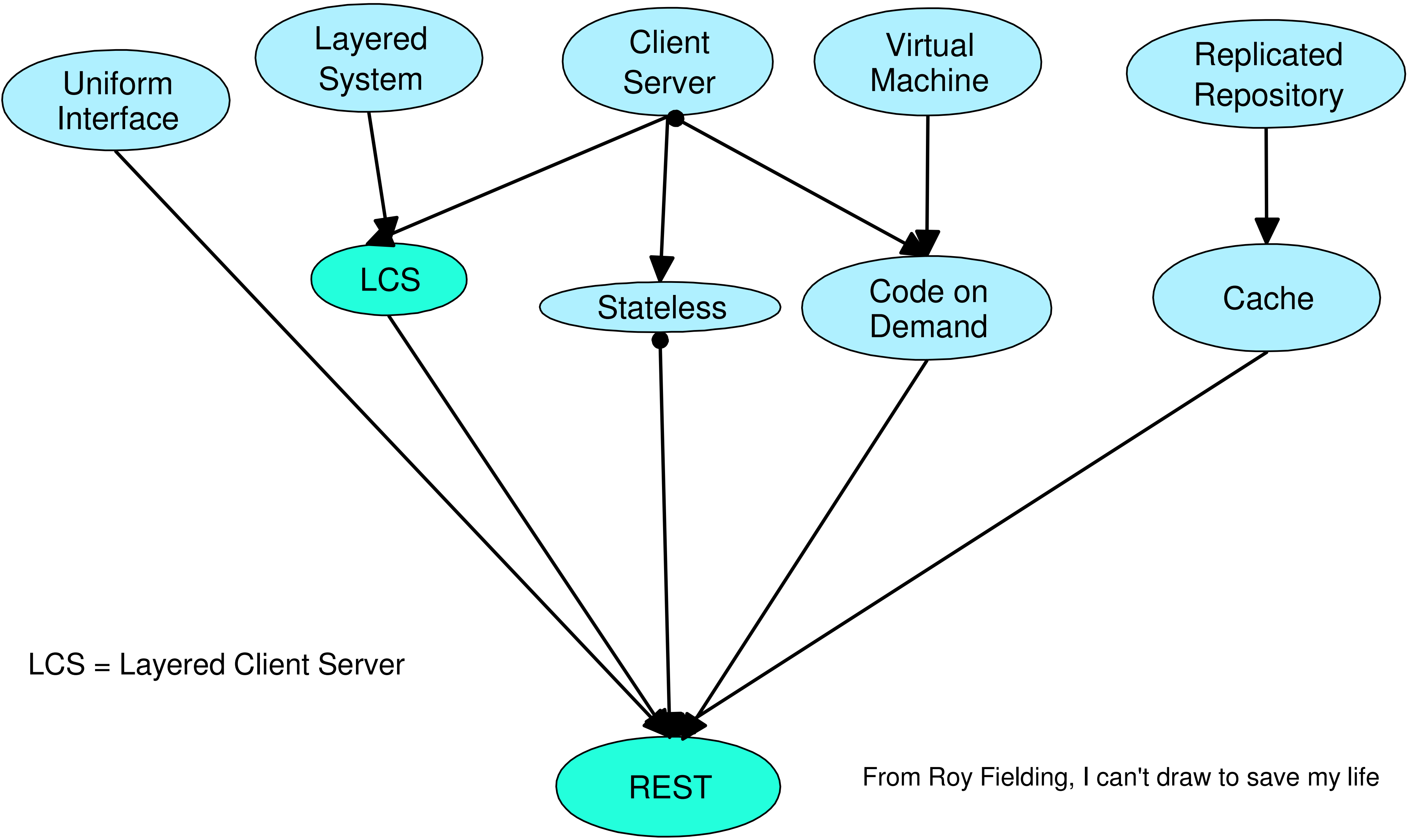
REST

- REST stands for **RE**presentational **S**tate **T**ransfer
- Roy Fielding's dissertation
  - **Architectural Styles and the Design of Network-based Software Architectures**
  - http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
- The meat is this chapter
  - Rest Architectural Style
  - http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

REST is a style

- An **Architectural Style** used to guide definition and implementation of design and architecture of the Web
  - modifications to HTTP and URI, HTML specs
  - implementations in Apache, libwww-perl, …
- REST is the "best of the lessons learned in developing the web".
- **REST = Design Patterns of the Web**

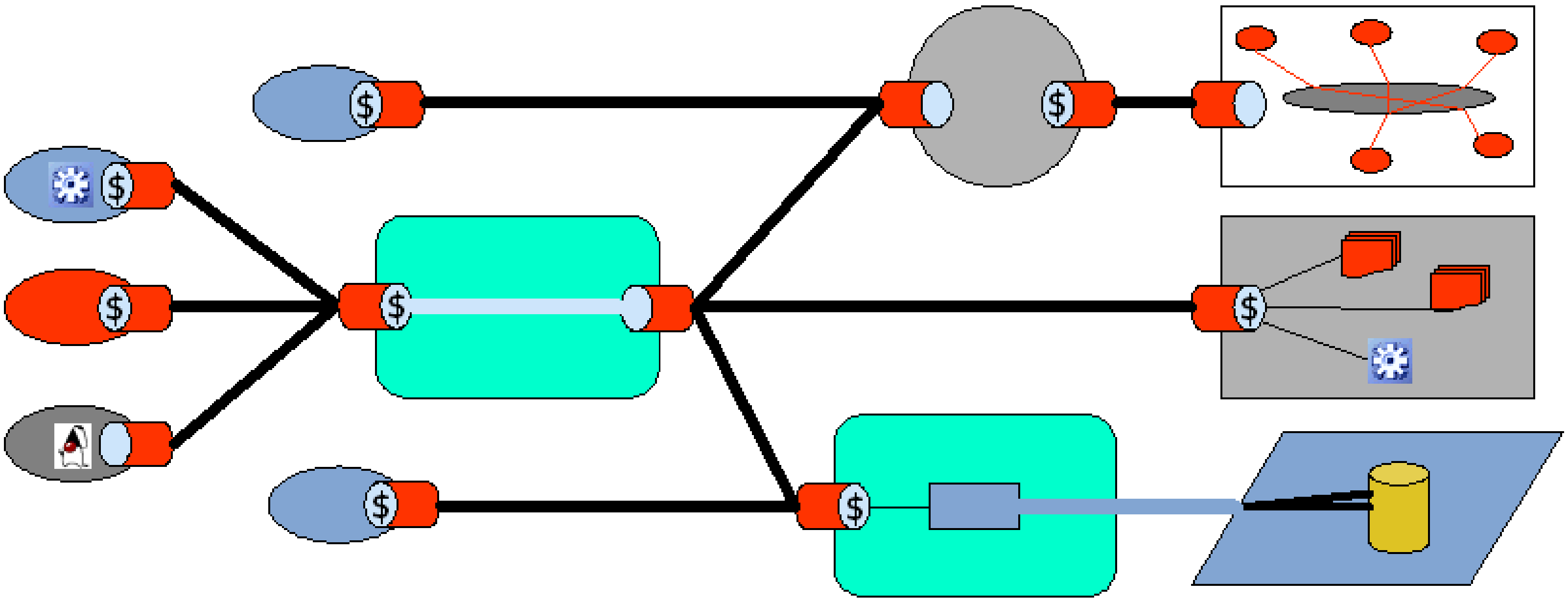
REST - origins of the style



LCS = Layered Client Server

From Roy Fielding, I can't draw to save my life

REST Process View



- Layered Client-Server
- Uniform Interface (like Pipe and Filter)
- Stateless, Cacheable Communication
- Optional Code-on-Demand

From Roy Fielding, I can't draw to save my life

High-level Web Requirements

(Fielding & Berners-Lee's motivation)

- Low entry-barrier
  - Hypermedia User Interface
  - Simple protocols for authoring and data transfer
  - Extensibility
- Multiple organizational boundaries
  - Anarchic scalability
  - Heterogeneous platforms
  - Gradual and fragmented change (deployment)
- Distributed Hypermedia System
  - Efficient for large data transfers
  - Sensitive to user-perceived latency
  - Capable of disconnected operation

Elevator Pitch

- REST is defined by four interface constraints:
  - identification of resources
  - manipulation of resources through representations
  - self-descriptive messages
  - hypermedia as the engine of application state

Four Horsemen of the Web

- **H**TTP
- **U**RI
- **H**TML
- **X**ML

- **HUHX** like Dr Huxtable of The Cosby Show
- (Nice Sweaters)

Four Verbs of the HTTP Web

- GET
- POST
- PUT
- DELETE

Web Dav adds a few more verbs to HTTP
97 % of the web gets by with 4 verbs
Early Browsers got along with 2 verbs (GET, POST)

Four Verbs of the Database World

- Create
- Retrive
- Update
- Delete

CRUD

Four Layers of the Internet

- TCP
- IP
- Datalink
- Physical Layer (Ethernet / Wi-Fi 802.11b)

Design Principles of the Internet

- End-to-End Argument
  - "functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level."
- Intelligence at the Endpoints
- Stupid Network
- Argument by Analogy to Distributed Computing and hence the web's design

How did the Web Happen?

Adam Bosworth

- **S**imple – Any “P” programmer can build apps.
- **S**loppy - add custom metadata or verbs
- **S**TANDARDS due to:
  - Single simple sloppy open wire format format
  - Runs on every platform, partly due to points 1 & 2.
- **S**cale (Massive and Linear) due to:
  - DNS (partitioning)
  - Caching
  - Stateless
  - Coarse Grained interactions

That's it.

- The Web Style (REST) is very simple
  - identification of resources
  - manipulation of resources through representations
  - self-descriptive messages
  - hypermedia as the engine of application state
- Live by it
- Become irrelevant without it

Hypertext Transfer Protocol

- The role of HTTP in Web Architecture
  - Extend uniform interface across the net
  - Minimize user-perceived latency
  - Enable layered processing
  - Enable caching
  - Enable extension and evolution
- Already survived a decade of evolution
- 1991-93: HTTP/0.9 [Berners-Lee]
- 1993-97: HTTP/1.0 [RFC 1945]
- 1996-now: HTTP/1.1 [RFC 2068/2616]

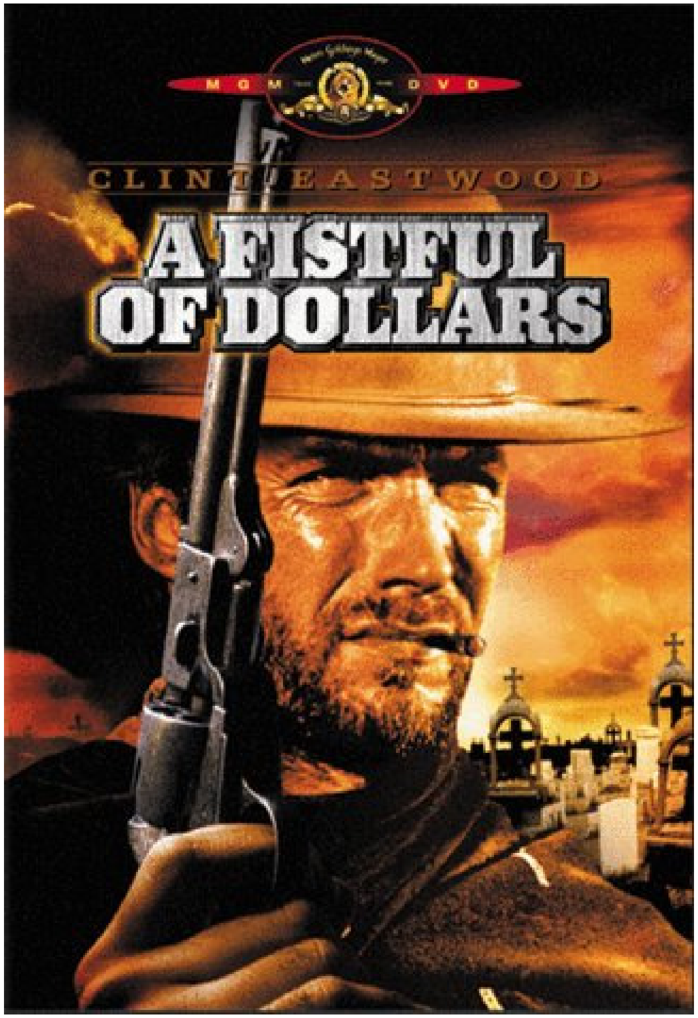
Monty Python - Spanish Inquisition Sketch

A REST Inquisition?

Nobody expects the RESTifarian Inquisition!

Our chief weapon is surprise...surprise and tedium ...tedium and surprise.... our two weapons are tedium and surprise...and ruthless disregard for unpleasant facts.... Our three weapons are tedium, surprise, and ruthless disregard ...and an almost fanatical devotion to Roy Fielding

Deadwood: Wild Wide Web



Criminals and Thugs (Cats)

IBM Technology Stack

- Customers charitably describe as "complex"
- Latest Buzzwords
  - Enterprise Service Bus (ESB)
  - Service Oriented Architecture (SOA)
  - WS-* (Web Services Nirvana or Tower of Babel?)
- Buses or Titanic?
  - Remember the InfoBus?
  - What about Icebergs?
- Busloads of Consultants?
- Architectures of Control?
- Get on the Buzzword Bus?

IBM Technolgy Stack (cont)

- Frameworks, Middleware, More Frameworks
- Politics and technology adoption
- Survival of the Fittest?
  - At a certain point there were 8+ groups at IBM working on a portal
- Widgets and Frameworks: WCL, JWL, ODC, JSF, Struts etc.
- Currently in Forms Area: XForms, AUIML (Abstract User Interface Markup), lots of others >8
- Data Modeling: SDO, Hibernate (open source), JService
- WSRP, WSXR, BOIL, WSIS, WSPT, WSUX
- Some of the above acronyms were invented

REST & the Dead WS spec Parrot

- **Mr. Praline:** Never mind that, my lad. I wish to complain about this specification what I purchased not half an hour ago from this very boutique.
- **Owner:** Oh yes, the, uh, the Big-Wizzdl...What's,uh...What's wrong with it?
- **Mr. Praline:** I'll tell you what's wrong with it, my lad. it's dead, that's what's wrong with it! Owner: No, no, it's uh,...it's RESTing.
- **Mr. Praline:** Look, matey, I know a dead specification when I see one, and I'm looking at one right now.
- **Owner:** No no it's not dead, it's, he's RESTin'! Remarkable spec, the Big-Wizzdl, idn'it, ay? Beautiful appendix!
- **Mr. Praline:** The appendix don't enter into it. It's stone dead.
- **Owner:** Nononono, no, no! it's RESTing!

REST & The Semantic Knight

- HACKER: What?
- SEMANTIC KNIGHT: None shall pass without using all sorts of semantic meta-meta-meta-stuff that we will invent Real Soon Now!
- HACKER: I have no quarrel with you, good Sir Knight, but I must get my work done on the Web. Stand aside!
- SEMANTIC KNIGHT: None shall find anything on the Internet without semantic metadata!
- HACKER: So be it!
- HACKER and SEMANTIC KNIGHT: Aaah!, hiyaah!, etc.
- [HACKER chops the SEMANTIC KNIGHT's first argument off by building efficent statistical/heuristic search engines]...

REST & The Semantic Knight

SEMANTIC KNIGHT: Just a flesh wound. [kick]

HACKER: Look, stop that.

SEMANTIC KNIGHT: You won't be able to get machine-machine services without an

ontology to formally describe all the relationships! [kick]

HACKER: Right!

[whop] [HACKER chops the SEMANTIC KNIGHT's third argument off by building

SOAPy and RESTful services with only implicit semantic descriptions]

SEMANTIC KNIGHT: Right. I'll do you for that!

HACKER: You'll what?

SEMANTIC KNIGHT: Come here!

HACKER: What are you going to do, bleed on me?

SEMANTIC KNIGHT: I'm invincible!

HACKER: You're a looney.

My Buzzwords - PITTS: Naming an Aesthetic

- Plain Old Distributed Computing (PODC)
- Pragmatic Inside-out Technology Types (PITTs)
- Good Enoughers
- View Sourcerers
- SHow ME the Coders (SHMEC)
- Radical Simplifiers
- Technology Buzzword Demystifyers (TBDs)
- Dark Matter of Technology (DMT)

My Buzzwords - Naming an Aesthetic

- Plain Old Loyal Oppositionists (POLOs)
- Layer Strippers
- Skeptical Original Debunkers (SODs)
- Technical Arteriosclerosis Terminators (TATs)
- Keep It Simple Sloppy (KISS)
  - Do the Simplest Thing that could possibly work
- Sexy Mom Factor Software
- Serendipity Manufacturers

Serendipity Manufacturers

- Google wrote Google Maps
- Craigslist - classifieds, housing (eBay but with a soul and with more vibrant community)
- 1 weeks work by third party developer
- Craigslist + Google maps
  - http://www.paulrademacher.com/housing/
  - the kind of magic that happens when you design for the web
- Neither Google nor Craigslist had anticipated that service, no WSDL, just nice URI and Resource Modeling

Glue Layer People

- **Life in the glue layer is about the outside-in**. **Pipes and filters** are your abstraction of choice, you might dream of Markov chains, the calculus of design heuristics and those old standbys, the rules of thumb, as you attempt to put order and infer structure where there was none.

- Where Mel Brooks' 2000 year-old-man considered Saran Wrap the greatest invention, I suspect for you it's **duct tape, spackle and wrenches** (or spanners as my Brit-colonized ears would prefer). For you it's all **Perl, Python, Ruby, shell scripts, URIs, bookmarklets** and the like.

- As an application designer my perspective has mostly been "inside out" and I've been forever amazed at the serendipitous magic that you glue layer people have been able to do with things I've built. My goal in life is to **find a way to encapsulate and codify the design patterns that would make your jobs easier**. I need to internalize that style as the best practices in what I develop.
- We need to help Glue Layer People, the **make users passionate** about our products, their magic brings **cold cash, Fistfuls of Dollars $$$**

Case Study: WebSphere Portal

"What feature was it, you might ask? There was no way to bookmark anything in WebSphere Portal."

- For the first 3 releases (2-3 years) of its existence you couldn't bookmark a page in WPS
- Jetspeed legacy?
- Not deemed important?
- Why?
- What about the Web in **Web**Sphere?

WebSphere Portal internalizes REST (v 5.1)

- No longer ignoring first plank of REST elevator pitch (Resource Identification)
- Now Customers can bookmark pages
- Can copy paste link to send in email or instant message
- Immediacy of **Communication, Collaboration, Communities**
- create teamspace, send email to new members with link
- Internally: hidden pages with portlets
- Addressability helps expose functionality in many different areas
- Build Workplace on top of Portal
  - lightweight layering through URIs
- **3 Releases spent fighting the web!**
- **Pain for us. Pain for Customers = $$$$$ lost**

WebSphere Portal internalizes the REST Style

- **Programmability** in the web sense was immediately enabled, the portal became a **composable** platform and we were able to layer the Lotus Workplace offering on top of it.
- **URIs** give **visibility to intermediaries** and so things like **caching** (where we had cool technologies like Dynacache) were far more easily enabled.
- Similarly for **logging** and profiling the portal, we could **use the same tools** for processing logs as exist for regular web servers like Apache.
- We had new opportunities for **pipelining** and **filter chains** (to do **transcoding** if needed).
- We had more options for **load balancing**, we could decide to deal with remote portlets through **iframe invocation** rather than through immature and complex protocols like WSRP. And so on...

Case Study: Google Web Accelerator

- client-side proxy, a whopping cache
- leverages Google's amazing server farms
- Exposed pervasive HTTP Abuse in minutes
- Developers have been misusing HTTP semantics e.g using hyperlinks to delete and cause side effects
- **Idempotency: GET SHOULD NOT have side-effects**
- GWA pre-fetched links on pages (smart spider)
- People Lost Data = Tears = Pain = $$$
- I'm Sorry I can't kiss and make it better
- Frameworks using HTTP should be better to mitigate these risks
- Still developers need to internalize the web architecture

Law of Leaky Abstractions

- All Abstractions Leak - Joel Spolsky
  - case study: Buttons and Forms (WCL, JSF, ODC, Asp.net all leak)
- Frameworks are good but can be dangerous in this case (each has different ideas about forms and buttons)
- ultimate abstraction is HTML form + button
- Recall the End to End argument
- Sometimes if you put too much into framework, you fight against the underlying abstraction
- Leaks - bugs - developer head-scratching - user head-scratching "not what I expected" - $$$$ lost
- Hence: We Need to plug leaks in our technology stack
- Aggressive Layer Stripping
- **"Lets not give users enough rope for them to shoot themselves in the foot"**

- Mixed Metaphor Koranteng Toli circa 1998

Case Study: SDO

- What's happened to Service Data Objects (SD0)?
  - SDO is anagramatically in SOD territory (Skeptical Original Debunkers).

> Service Data Objects (SDO) was proposed and accepted as a JSR in December 2003. However, almost eighteen months later, there's not even an expert community outside of IBM and BEA, no draft of the spec is available. What's happened to the SDO spec?
>
> -TheServerSide.com last week

Case Study: SDO (cont)

- The answer: we're working on it at IBM.
- Right?

Some SOD replied:

The Spec Has Been Renamed

Rumor: The actual progress of this JSR has been appropriately renamed to **"Stale Data Object".**

I just wonder how do I transfer my "business tier" objects / meta-objects to this SDO and then from SDO to my presentation objects? Maybe it is time to create another JSR named **"Duplicating Data Object (DDO) specification"** So you can transfer from

EJB->DDO->SDO->DDO->POJO->JSF->HTML.

How can we do EJB->DDO? You guess! ;-)

- Not picking on SDO but this is the **perception** about many of our offerings (unnecessary layers, overly complex etc)
- Complexity -> Consultants
- **No one *really* wants consultants**

View Source Imperative

- How did you learn HTTP?
- Did you learn HTTP?
- How did you learn HTML?

1. Reading the spec?
2. Formal education?
3. From a book?
4. View source?

Why Specs Matter

- **90% of everything is crap. That's Sturgeon's Law.** Software is not excluded from this principle. We live in a mass-market, low-bid, first-to-market world.Our goal ultimately is to be less sucky than our competitors.

- A written spec is key to giving an organization the flexibility to grow. Without it, adding new people to a project requires that the existing engineers take time out to bring the new members up to speed. This is the mechanism behind **Brook's Law ("adding people to a late project only makes it later")**.

- But a good set of project documentation can temper the effects of Brook's Law and provide some scalability...

- So, if you don't write a spec, your name will be cursed in languages unknown to you in far away countries, and also by kids in middle school today when they try to figure out your code 10 years from now.

- Rob Weir (he's in the Dark Matter of Technology at Lotus/IBM

Technical Arteriosclerosis
(Definition)

- The Case of the "Prototype that shipped"
- The "Framework from Hell"
- The bundle of code even experienced developers run away from
- "We can't fix A because it would break X, Y & Z"
- Clogged Arteries
- Sound Familiar?
- Hit close to home for anyone?

Technical Arteriosclerosis

How? and Why?

- "It is easier to write code than read code"
  - Joel Spolsky
- Developers are Busy
- No sharing of code
- Poorly documented code
- Insane schedules
- No consideration about maintenance costs
- Development organization needs a spine

Technical Arteriosclerosis

How to fix?

- View Source Imperative
  - Developer inertia can be overcome
  - Need lots of good code and good samples
  - Need to spam Google and evangelize
  - Good samples should be first search result
- Make time for specs upfront or design documents after the fact
- Early detection of clogged arteries
- Refactor pragmatically
- Clean Specs & Lightweight touch
  - (e.g. Atom vrs 8 variants of RSS - clogged arteries)

Some Questions

- Has everyone at IBM bought into the web and the HUHXtable Sweater Quartet (HTTP, URI, HTML, XML)?
- Is the web not Good Enough (TM) for our purposes?
- Have we embraced the web?
- Have we internalized the web style? Or are we fighting it?
- Have Layer Strippers and Radical Simplifiers cast their glances at our technology stack?
- Is componentization a buzzword?
- Documenting and sharing code vrs empire-building

IBM Software Group: The Baby Eating Starts Here

- The more I think about what service oriented architecture means the more I realize l**oosely coupled has to go beyond lip service.** Organizations as much as architectures must be decoupled, so they can be remixed. Its just so much horse manure to talk about SOA without a formal commitment to loose coupling. That is, open documented interfaces across granular components or services, with no funny business and hidden calls. Interoperability is not just a marketing term. You can't have SOA and attempt to drive lock in.

- The more I think about the problems of SOA the more its clear the culture of a company will be as important in delivering it, from a vendor perspective, as any set of technical assets. Monoliths are not service oriented. But, we can't break them down without freedom of disassociation.

-- James Governor, Analyst at RedMonk
Listen to "The Guvnor"

More Quotes

"As an ostensible organization of inventors, our choices are

1. use the rich assortment of tools out there to get things done or

2. Spend years architecting and struggling with cumbersome tools"

- Jonathan Feinberg

My comment:

Do we eat our own dogfood? Will customers eat it? Even with busloads of consultants?

Efficiency of the web style

- Jim Gray Distributed Computing Economics
- relative inefficiency of many in the industry that haven't embraced the web style.

"Megaservices like Yahoo!, Google, and Hotmail have relatively l**ow operations staff costs**. These megaservices have discovered ways to deliver content for less that the milli-dollar that advertising will fund.

For example, in 2002 Google had an **operations staff of 25** who manage d its two petabyte (2^15 bytes) database and 10,000 servers spread across several sites.

Hotmail and Yahoo! cite similar numbers - small staffs manage ~300 TB of storage and more than ten thousand servers."

Up until 2000, I believe the **operational staff at Yahoo was 8** and included the co-founder Jerry Yang.

Low End Theory

- Google's strategy
- Cheap hardware
- Exploit Moore's Law
- Software to implement redundancy
  - human fallibility
  - hardware reliability
- Minting money as we speak
- No layoffs that I know

Great album by the way

Innefficient IT structure at IBM

- Tim Bray (Mr XML, now at Sun) had this to say

  $46,213,000,000.00

I looked up the answer to the question: What is IBM's consulting revenue? In 2004, **IBM's gross revenue was $96B**, of which **$46B was Global Services, i.e. consulting.** I see that basically as testimony to how our profession, the IT profession, has failed our customers. Nothing against IBM; in fact, as solution-providers go, my experience is that IBM GS is pretty good. But if you see IBM as a microcosm of the industry, **it shouldn't cost $46B in consulting to deploy $50B worth of technology**. It's not going to be easy to get there, and it's going to take a long time, but we just have to focus on making things simpler.

Eating our babies

- Are we prepared to change our corporate structure and have profits come more from well-designed and usable software as opposed to consultantware?
- Are we prepared to eat our babies and make products that have much lower operational costs?
- I suggest that fully embracing the Web Style is a step in the right direction.

ARESTed Development?

- Resource Modeling
  - Every important resource should be identified
- Use the right verb
  - Don't abuse HTTP
  - Your users will hate you if you do
  - Life is easier using the web style: leverage
- Respect browser conventions
  - Bookmark
  - Back button
  - Refresh
- Layer Stripping & Radical Simplification

You're Under ArREST

Miles Davis Soundtrack

My Jaundiced Prose

- A REST Intervention
- Pitts: Naming an Aesthetic
- Get on the Bus
- The Unloved HTML Button and Other Folktales
- People, Processes and Things
- On Bleach, Entertainments, Forms, Atom, Kiss and Sexy MFs
- On Gmail and DHTML Architecture Again
- On Rich Web Apps, Alphablox and Oddpost
- Deadwood and the Web Application Leap
- http://koranteng.blogspot.com/

The Unloved HTML Button and Other Folktales

A long, long time ago in a far, far-away land, there lived an HTML Button.

It lived happily with its parents, Mr HTML and Mrs Form, and its siblings: the older brother Text Entry Box and younger sisters, Drop-down List and Radio Button. As a family, they didn't look like much, but they got on with things.

The Unloved HTML Button and Other Folktales

- Indeed some would say that they had been second thoughts since grand-parents, Netscape, Microsoft, Apache and Berners-Lee, preferred to fawn and monopolistically squabble over cousins, Blink and Marquee Tag, the mobile duo, Java Applet and ActiveX, and the gruesome but quietly efficient twins, CSS and JavaScript. Let's not forget the ugly outlier, the Behaviour, and that perennial favourite of old Sir Tim, Semantic Purity.

The Unloved HTML Button and Other Folktales

- Still those pragmatic matchmakers, Fielding and Andreesson got together, sacrificed a few goats on the altar of sloppy expediency and dotcom hubris respectively and got the marriage together. The good Reverend HTTP was the officiating priest. Thus they called for a great many festivities and plenty palm wine and Schnapps was consumed by the merry populace. Yahoo! and Google, they ululated as they devoured the fatted Browser calf.