Anne Thomas Manes
VP & Research Director
amanes@burtongroup.com
www.burtongroup.com
Twitter: @atmanes

Selling REST to your Boss

Tuesday, 6 October 2009

JAOO conference
- for Developers by Developers

Introduction

Anne Thomas Manes
VP & Research Director, Burton Group

Provide research and advice on:

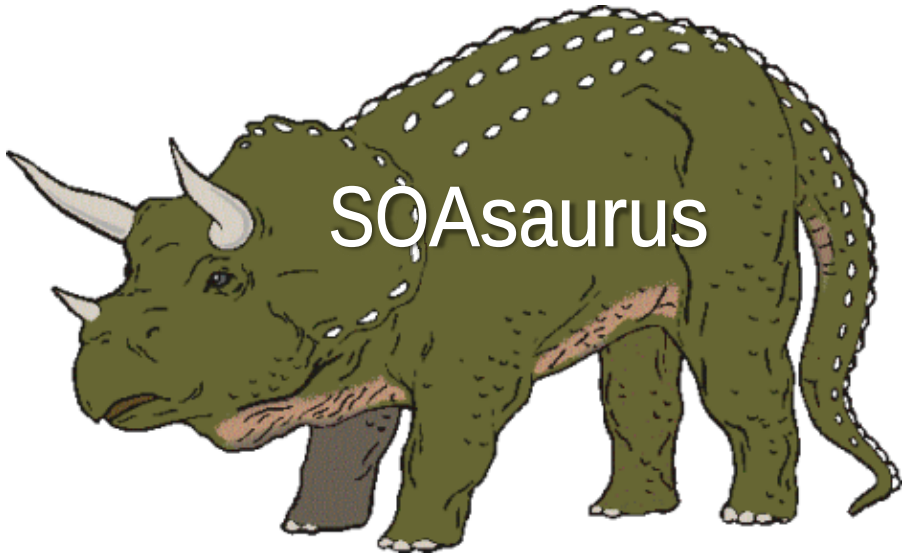
- Software development lifecycle
- Application architecture
- Application platforms

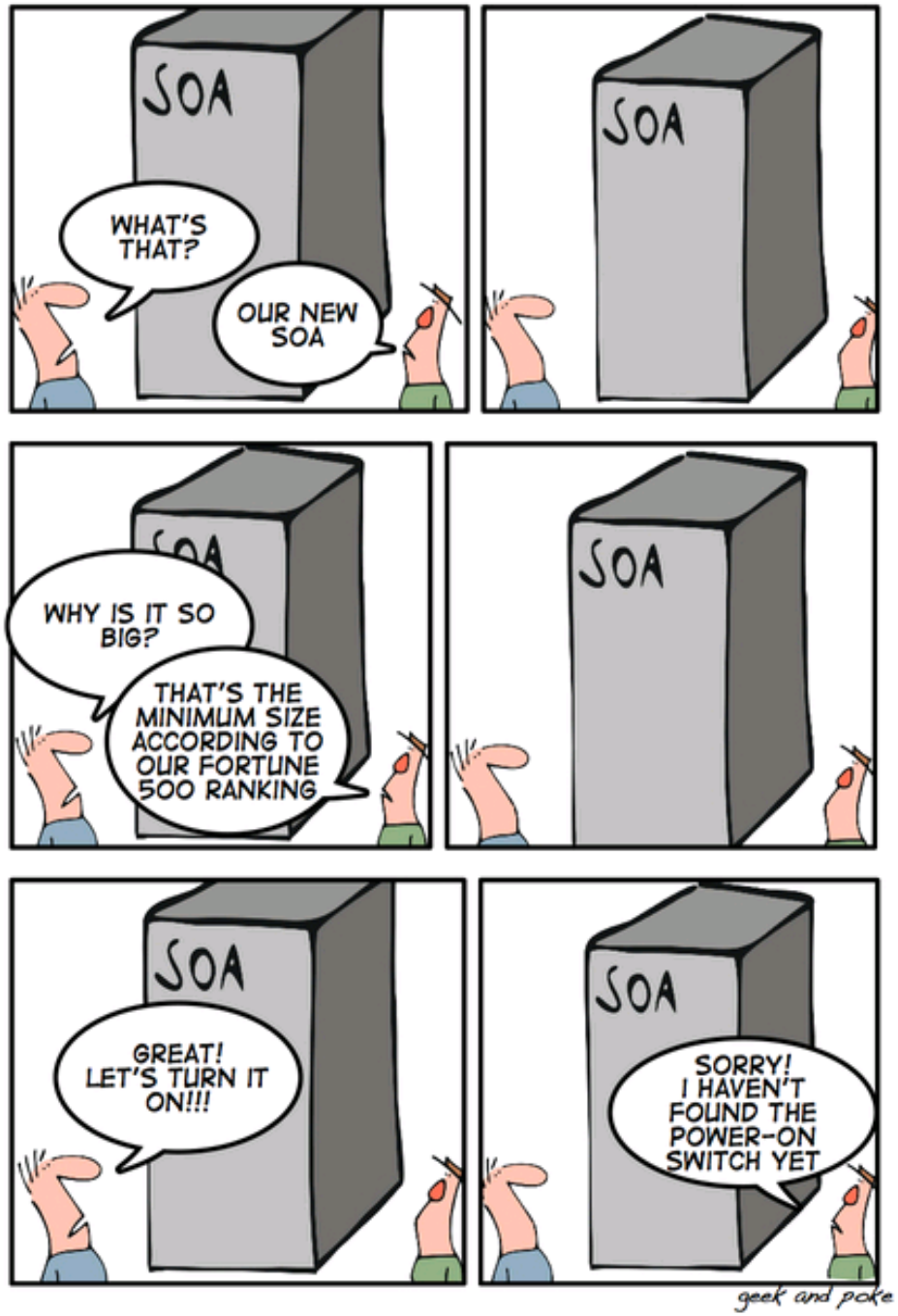
Prior work experience:

- CTO at Systinet [now HP]
- Director of Market Innovation at Sun [soon to be Oracle]
- Before that: OEC [Embarcadero], DEC [HP], Cullinet [CA], IBM
- Standards development at W3C, OASIS, JCP

SOA Obituary

SOA met its demise on January 1, 2009, when it was wiped out by the catastrophic impact of the economic recession.
SOA is survived by its offspring: mashups, SaaS, Cloud Computing, BPM, and all other architectural approaches that depend on "services".



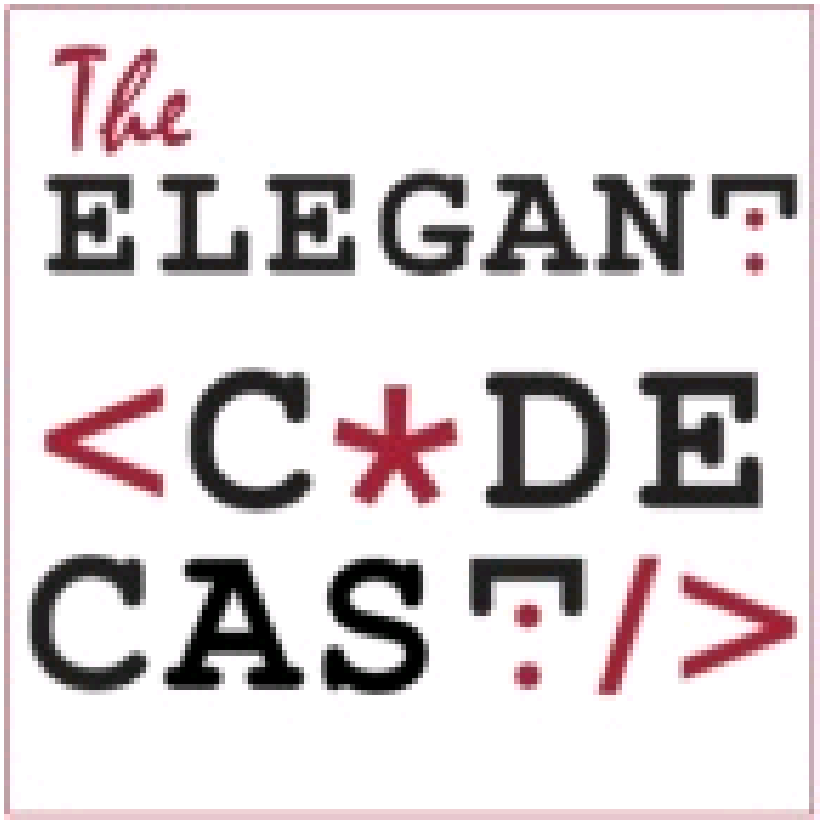

SOA
WHAT'S THAT?
OUR NEW SOA
SOA
SOA
WHY IS IT SO BIG?
THAT'S THE MINIMUM SIZE ACCORDING TO OUR FORTUNE 500 RANKING
SOA
SOA
GREAT! LET'S TURN IT ON!!!
SOA
SORRY! I HAVEN'T FOUND THE POWER-ON SWITCH YET
geek and poke

Reasons why you want to do REST

burton GROUP™

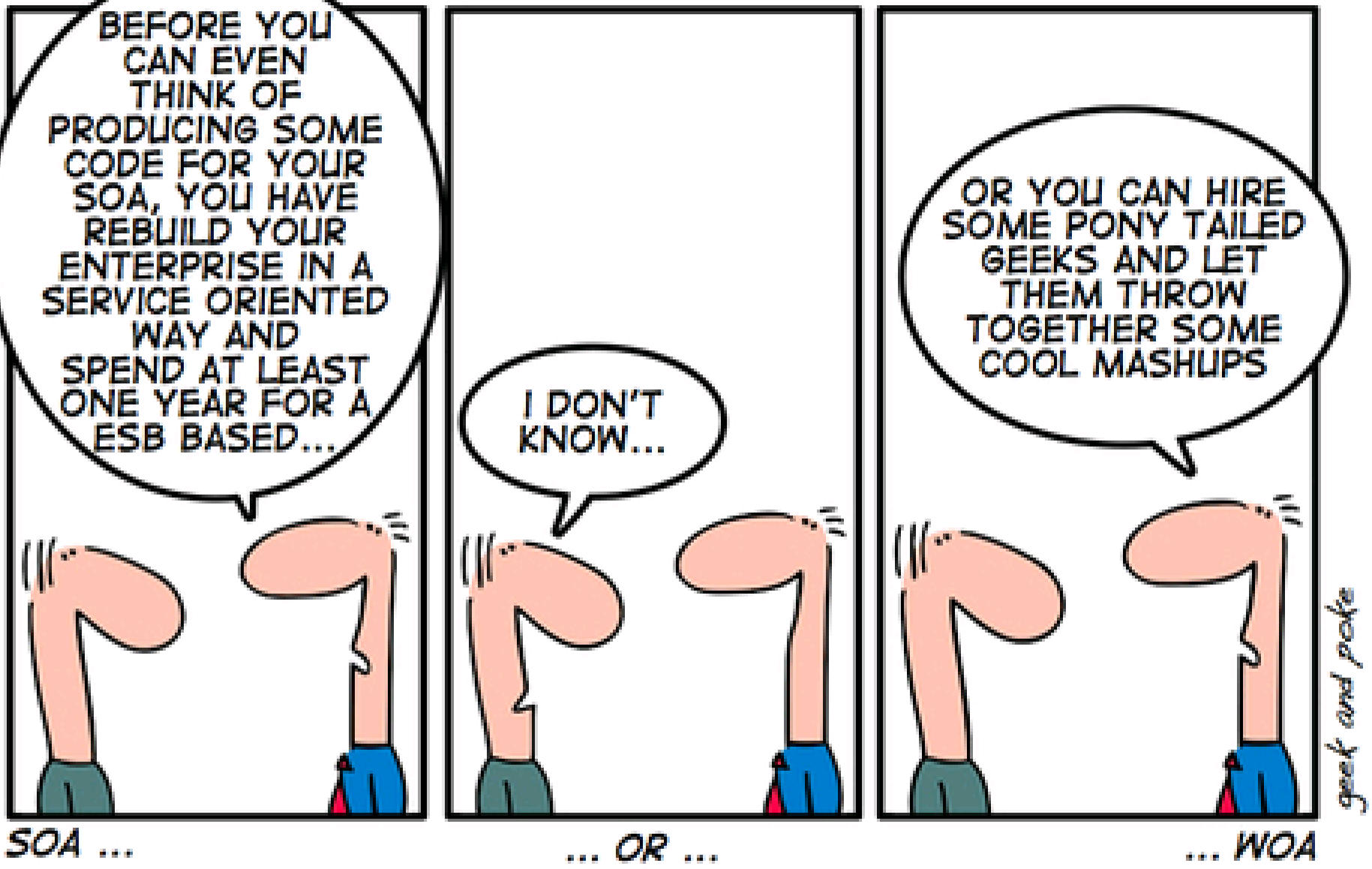
It's cool

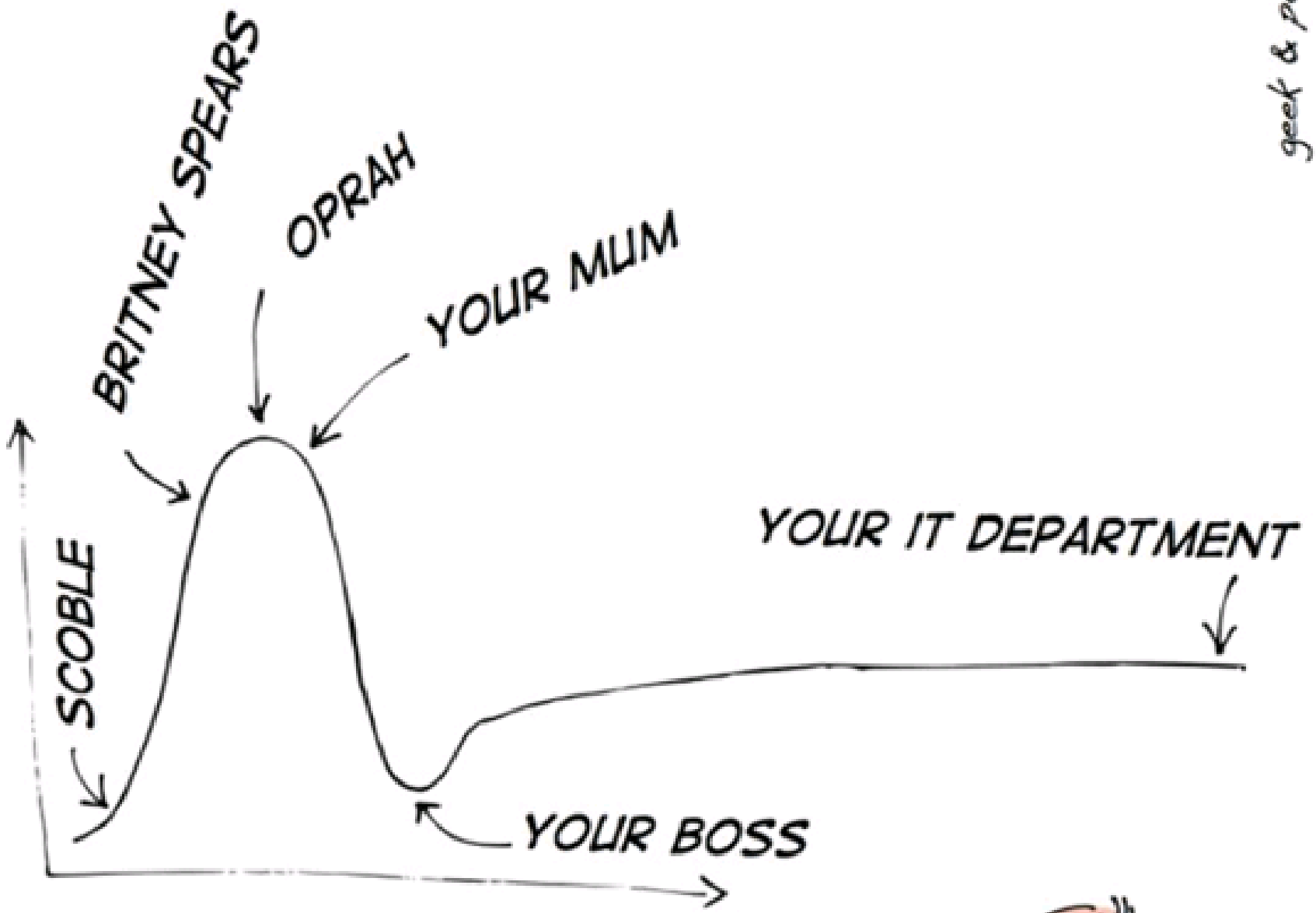
It’s elegant



It would look great on your CV

CONSULTANTS HANDBOOK – CHAPTER 10: LISTEN TO YOUR CUSTOMER




BRITNEY SPEARS
OPRAH
YOUR MUM
SCOBLE
YOUR IT DEPARTMENT
YOUR BOSS

Selling REST to your Boss

Thesis

- REST is the best architectural style to use for applications that run on the Web
  - Web-native
  - Simpler, more elegant applications
  - REST traits: scalability, evolvability, serendipity, …
- So what?
  - Elegance doesn't sell
- You need to provide compelling business arguments
  - Tie your proposal to business goals

Selling REST to your Boss

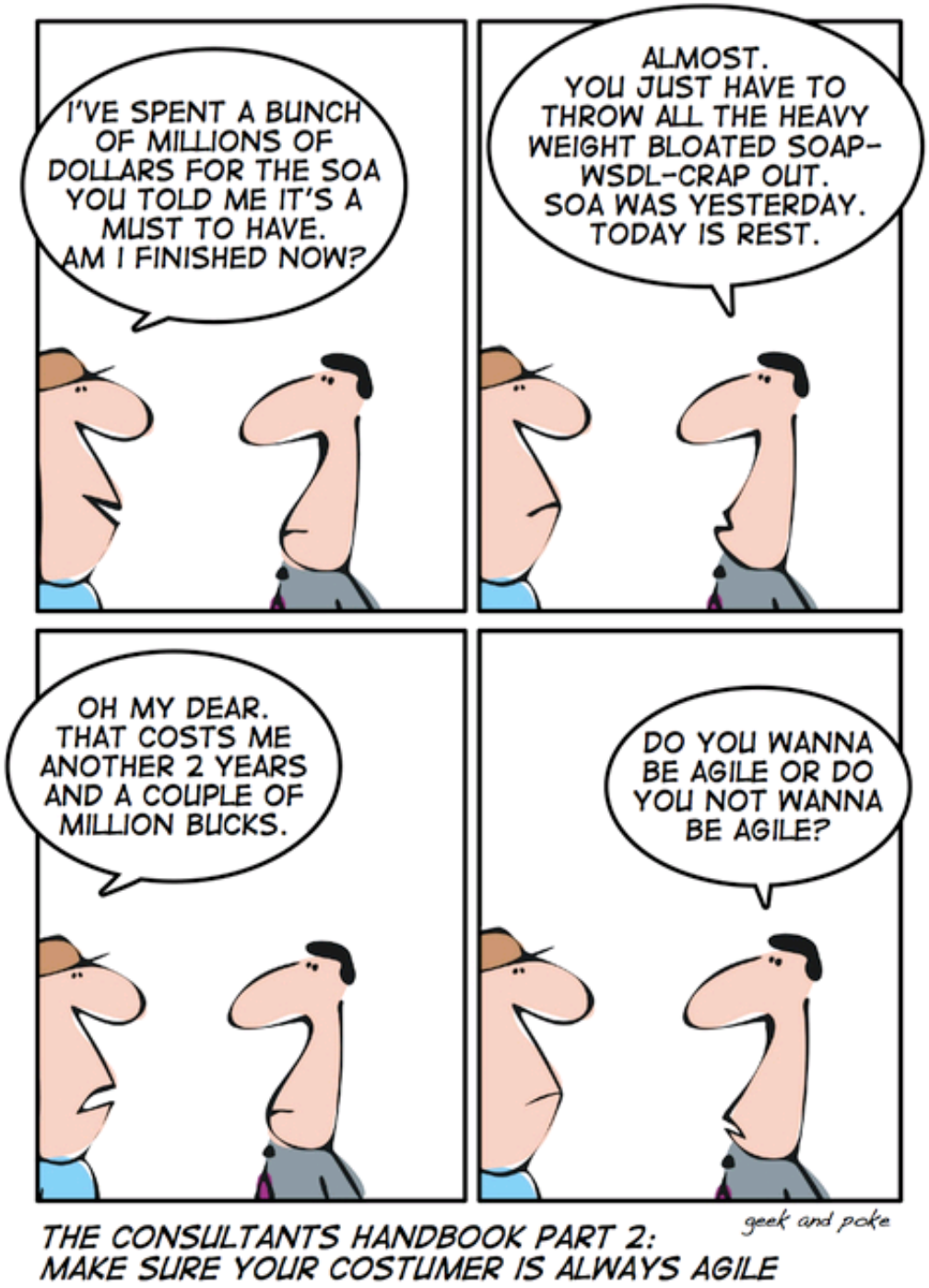
Agenda

- Hurdles that you must overcome
- Establishing vocabulary
- Developing a compelling business case

Hurdles

burton GROUP™

SOA Backlash



Technology Doesn't Matter

"Technology alone is rarely the key to unlocking economic value: companies create real wealth when they combine technology with new ways of doing business."

Source: "Eight Business Technology Trends to Watch, McKinsey Quarterly December 2007"

Impressions that REST isn't robust enough for enterprise applications

Patently Untrue

The Web is a RESTful application

- Largest, most scalable, most resilient application ever
- Many existing technologies and patterns for delivering robust features:
  - Security
  - Reliability
  - Transactionality
  - Fault tolerance
- Requires a different design approach from traditional middleware frameworks
- Nonetheless: REST is not appropriate for every application


Bill Day
The Commercial Appeal
Tools!

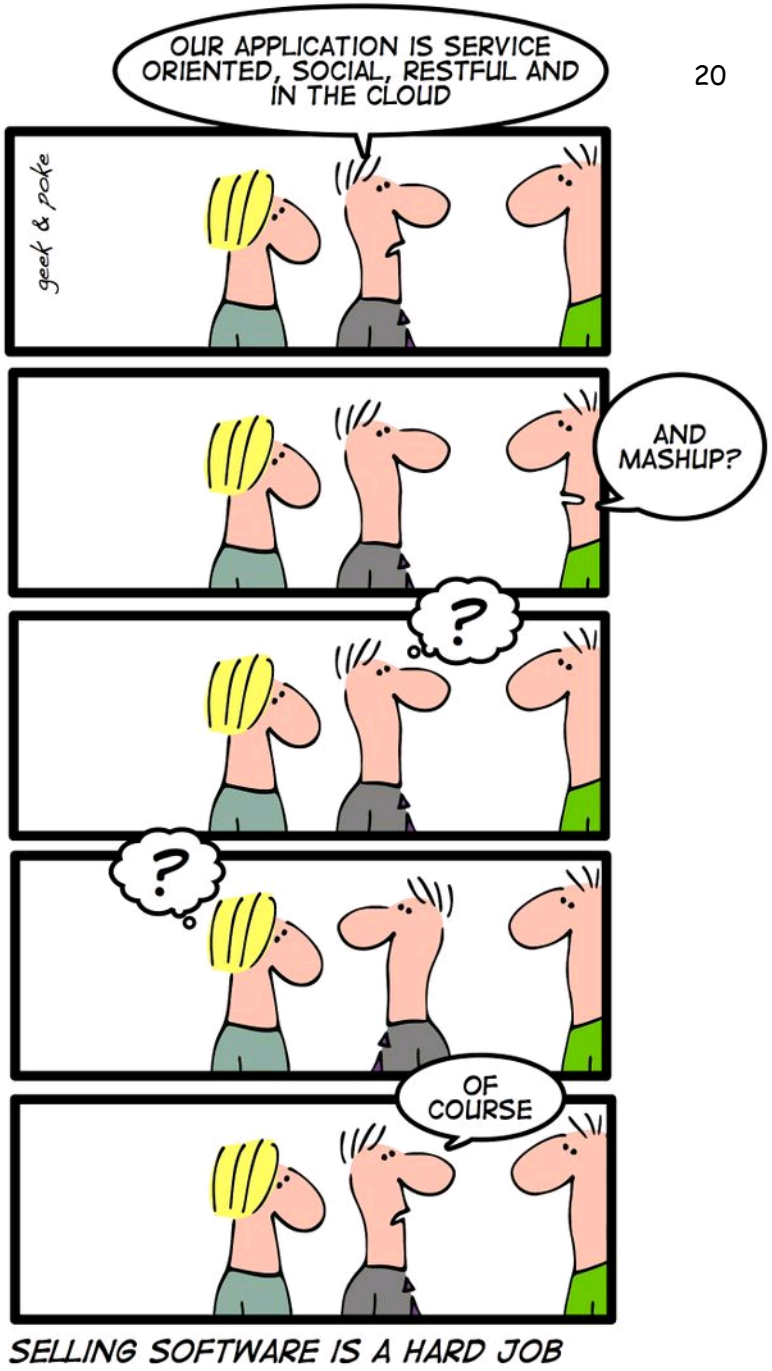
REST, HTTP, and POX

REST frameworks

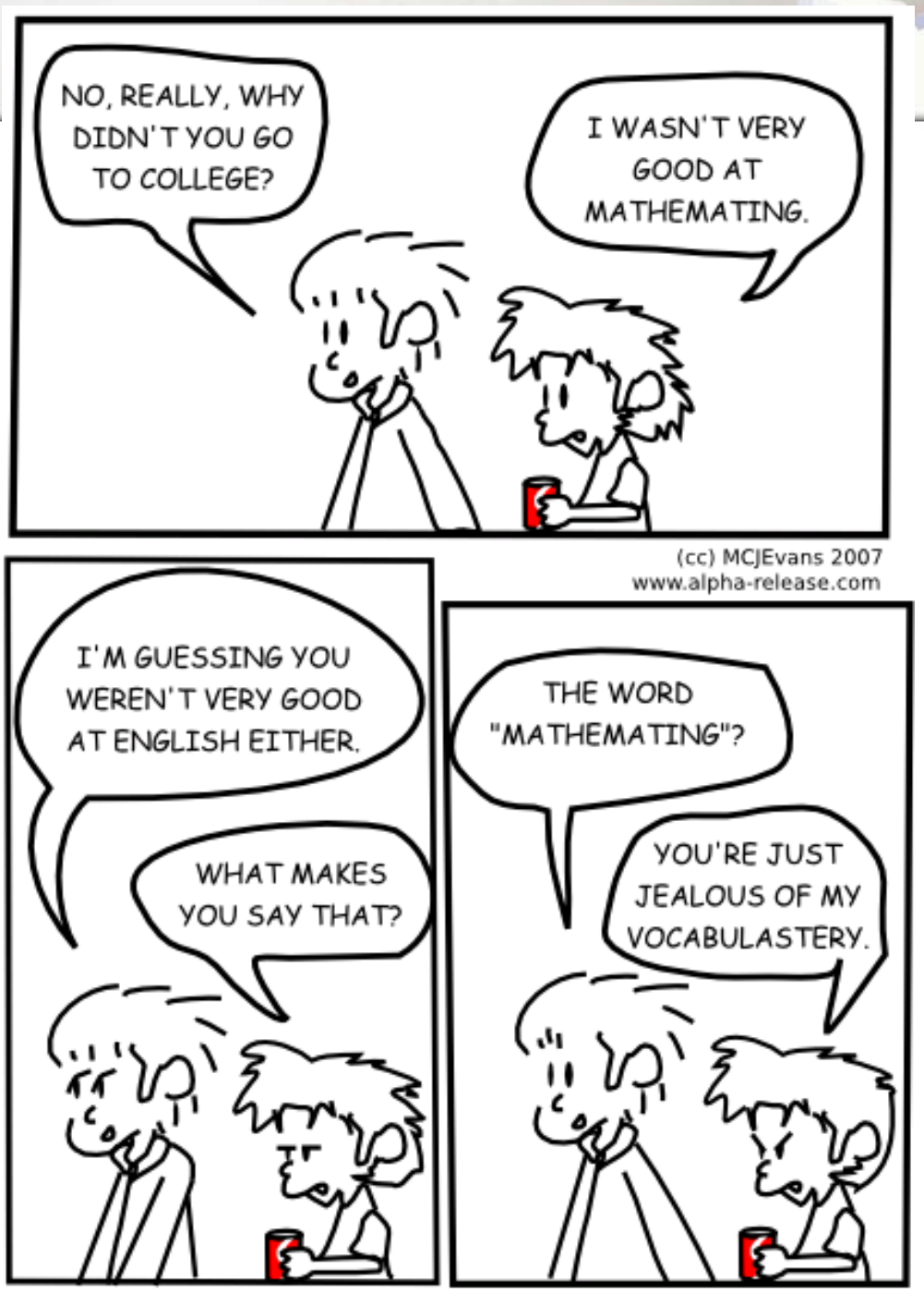
- It is possible and legitimate to build REST systems with any HTTP-enabled application environment
  - It's just not fun
- Some emerging REST frameworks:
  - JSR 311/JAX-RS — Java API for RESTful Web Services
  - RESTlet — Java, open source
  - NetKernel — Java, open source
  - Struts2 (w/ REST plug-in) — Java, open source
  - Ruby on Rails — Ruby, open source
  - Django — Python, open source
  - CherryPy — Python, open source
  - ADO.NET Data Services — .NET, beta, commercial
  - .NET 4.0 WCF — .NET, beta, commercial
  - IBM sMash — Groovy/PHP, commercial product

Only geeks think REST is cool

Others have no clue what it means



Vocabulary


NO, REALLY, WHY DIDN'T YOU GO TO COLLEGE?
I WASN'T VERY GOOD AT MATHEMATING.
(cc) MCJEvans 2007
www.alpha-release.com
I'M GUESSING YOU WEREN'T VERY GOOD AT ENGLISH EITHER.
WHAT MAKES YOU SAY THAT?
THE WORD "MATHEMATING"?
YOU'RE JUST JEALOUS OF MY VOCABULASTERY.

What is “REST”?

REST: The formal definition

- REST = Representational State Transfer
- REST is an architectural style
  - Defines the architecture of the World Wide Web
- Dissertation by Roy Fielding: “Architectural Styles and the Design of Network-based Software Architectures”
  - Defines a set of architectural principles and constraints
  - Describes how HTTP and the Web work
- Applications that adhere to the constraints are RESTful
  - Exhibit many desirable benefits: simplicity, scalability, performance, reliability, visibility, evolvablility, serendipity, and more

What is “REST”?

REST: A more pragmatic definition

- REST = Using the Web correctly
  - is “on” the Web, not tunneled through it
  - exploits and augments network effects
  - also called “WOA”, “ROA”, “RESTful HTTP”
  - I like “Thing Oriented Design” (TOD)

What is “REST”?

Five basic rules

1. Every *thing* has an ID
   → URL
2. Every *thing* links to other *things*
   → Hyperlinks
3. Every *thing* exposes standard methods
   → GET, PUT, POST, DELETE
4. A *thing* may have multiple representations
   → e.g., XML, JSON, Atom, microformats, images, etc.
5. You interact with *things* statelessly;
   → use hyperlinks for state

Understanding REST

The World Wide Web is a massive hypermedia application

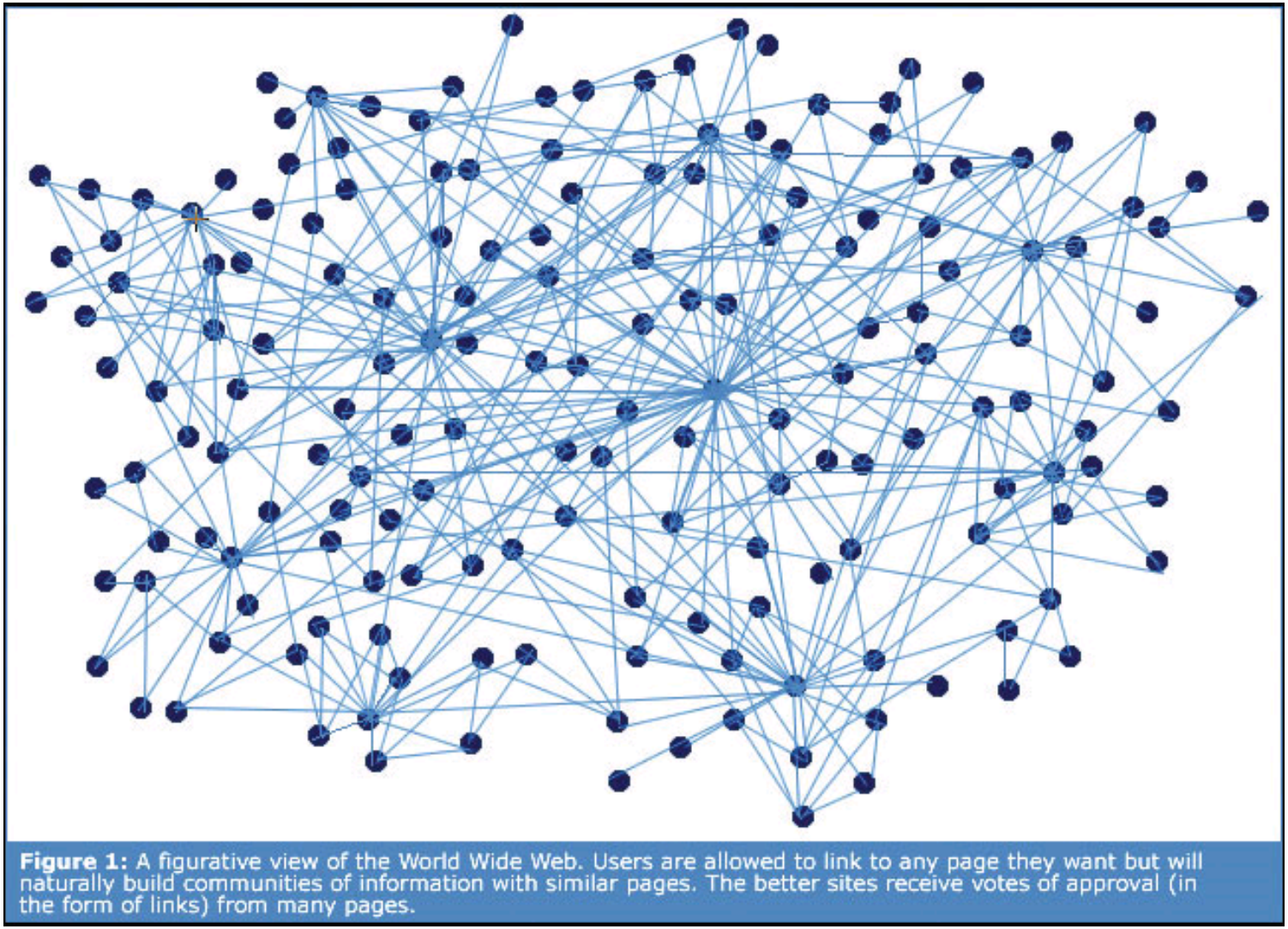


Figure 1: A figurative view of the World Wide Web. Users are allowed to link to any page they want but will naturally build communities of information with similar pages. The better sites receive votes of approval (in the form of links) from many pages.

Common REST Mythconceptions

Myth: REST = POX over HTTP

- REST applications don't have to use XML
  - REST supports any media type (i.e., multiple representations)
- Many POX applications are not RESTful
  - Tunneling RPCs, inadequate use of URIs and hypermedia, stateful
- There's nothing unRESTful about an envelope:
  - Atom, SOAP, etc.

Mythguided efforts

- JBoss's REST-* initiative
  - Defining RESTful interfaces to traditional middleware
  - (But you shouldn't use traditional middleware with REST)

What is Not RESTful?

Tunneling RPCs through URLs:

- http://my.com/customers?method=insert&name=Smith
- http://my.com/orders?method=deleteOrder&id=12345

What is Not RESTful?

Tunneling RPCs through HTTP POST:

```
POST http://my.com/CustomerMgmt
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <deleteCustomer xmlns="http://example.com/ns1">
      <customerId>13</customerId>
    </deleteCustomer>
  </soap:Body>
</soap:Envelope>
```

Method ID Endpoint

Providence (PVD) to San Diego (SAN) - Mozilla Firefox

File Edit View History Bookmarks Tools Help del.icio.us

http://www.expediacorporate.com/pub/agent.dll

wilcox tavern ri

Latest Headlines | Burton Group Client ... | Gmail - Inbox (11513... | Conference 4.7 Admi... | Microsoft Office Live ... | TypePad | Application Platform ...

whump.co... | the ryan k... | faq - Micr... | what-can-... | hcard - Mi... | hcalendar ... | implement... | Provid... | Google Maps

Expedia® Corporate Travel

Burton Group, Inc.

home | flights | hotels | cars | maps | specials

Welcome, **Anne** Sign out

My Trips | My Profile | Customer Support

Choose a departure flight or view complete roundtrips

Your search New search

Providence (PVD) to San Diego (SAN) Sun Jun 22

San Diego (SAN) to Providence (PVD) Sat Jun 28

No airline preference, Economy/Coach

At a glance | Filters | Nearby airports | New search

Stops / Times	All results (33)	US Airways	United	Continental	Delta	Northwest	American Airlines
Nonstop							
1 or nonstop	$595	**$595**	$643	$751	$999	$826	$1085
All results	$595	**$595**	$643	$751	$793	$826	$1085

Best value flight more details

US Airways 3159	Depart **11:40 AM (PVD)**	Arrive **6:48 PM (SAN)**	1 stop	$701	SELECT
Continental 6462	Depart **12:38 PM (SAN)**	Arrive **10:58 PM (PVD)**	1 stop		

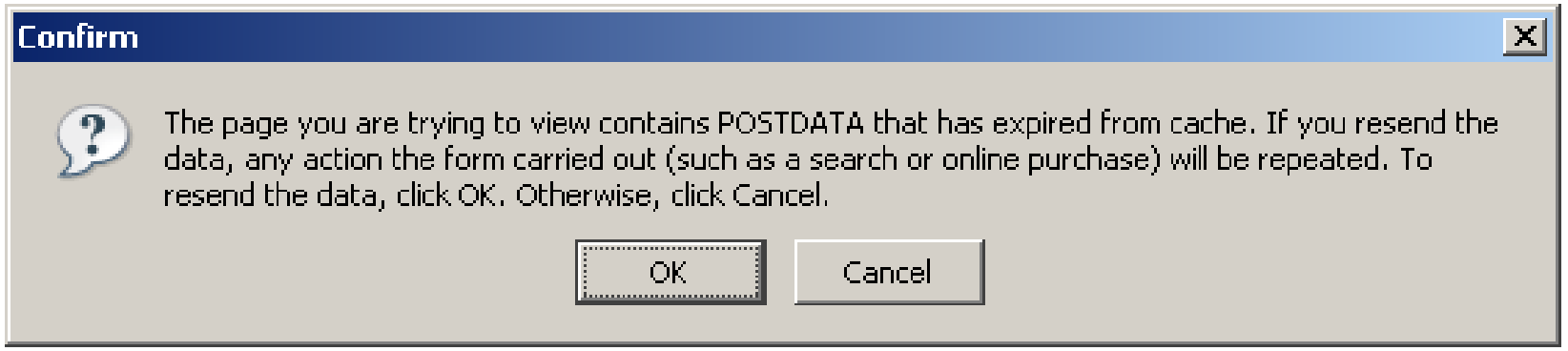
Find: formDefault Next Previous Highlight all Match case

Done

What is Not RESTful?

Not using hypermedia as the engine of state (HATEOS)

- Use hyperlinks to reference state and related resources
- Don't rely on session state across interactions:

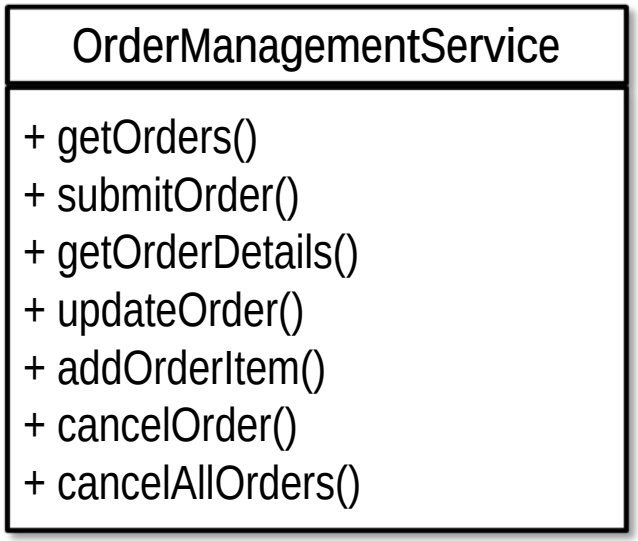


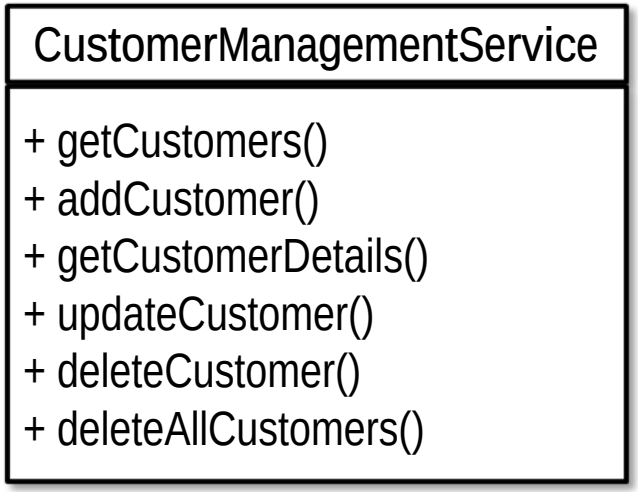
Key take-away

REST is about *design*, not technology

- REST = Using the Web correctly
- Resource-oriented model
  - A RESTful service's interface is the set of resources (URIs) it exposes
  - The more resources exposed, the more value the system contributes to the Web
- Specific technology is not important
  - Just because you're using a REST framework, that doesn't guarantee that you will produce RESTful applications

Method Oriented Interfaces

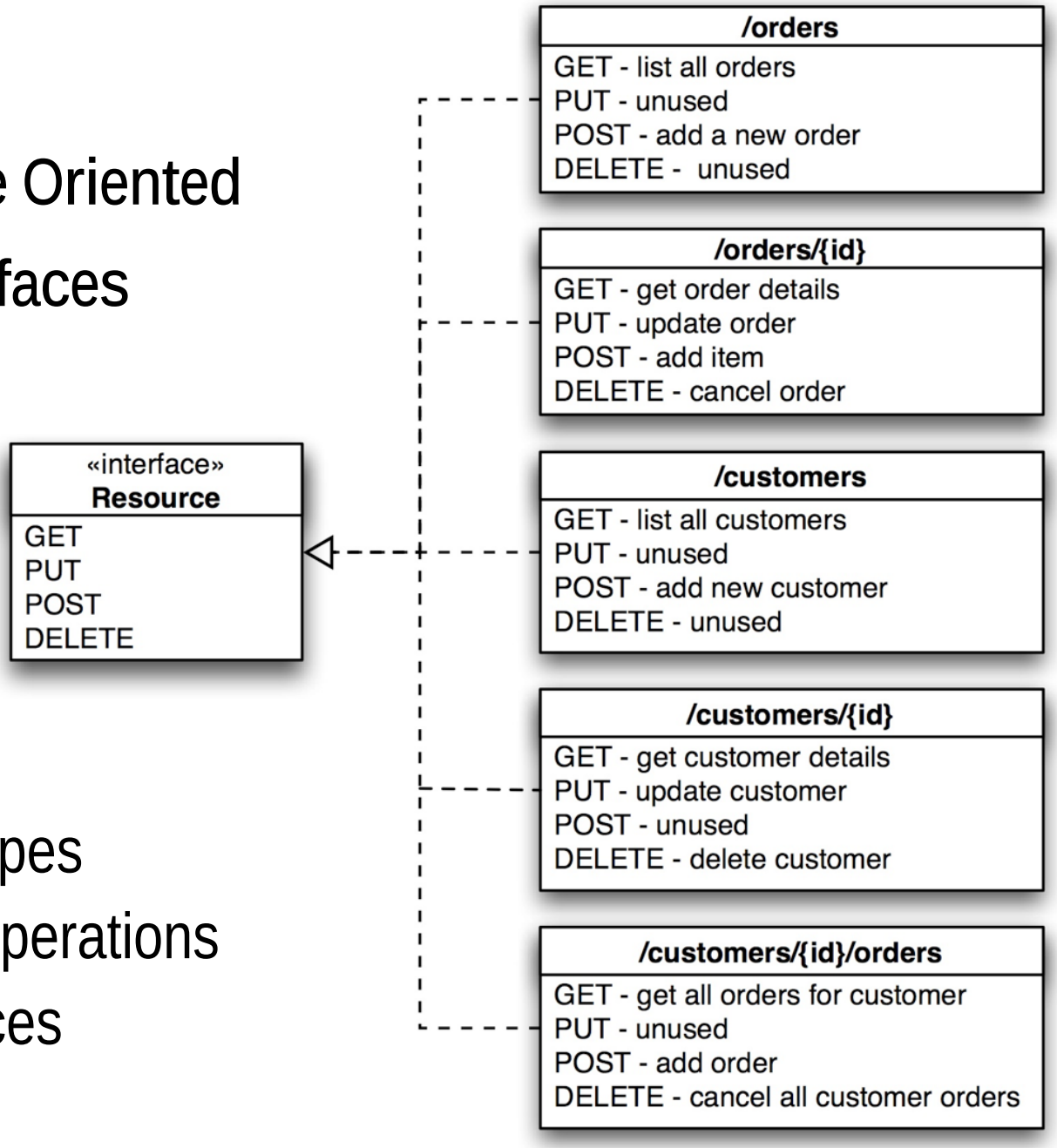




- Many data types
- Many operations
- Few instances

Example borrowed from Stefan Tilkov

Resource Oriented Interfaces



- Many data types
- Few (fixed) operations
- Many instances

Example borrowed from Stefan Tilkov

Understanding REST

RESTful service's contribution to the Web's value

- Millions of URLs
  - Every customer
  - Every order
- Addressable, linkable, cacheable
- Easily consumable, mashable
  - Supports multi-modal clients

What is “REST”?

REST: A business definition

- REST = Using the Web to your best advantage
  - Delivering higher-quality applications
  - that reach the widest possible audience
  - that deliver a better user experience and
  - enable users to use the information more effectively

Developing a compelling business case

Developing a Business Case

Describe the pros and cons of a potential investment

Building block	Description
Deliverables and benefits	What will we get out of this investment?
Costs	What do we need to invest?
Risks	What can go wrong?
Roadmap and timeframe	What do we have to do and how long will it take?
Assumptions	What's the starting point?
Alternatives	What other options do we have?
Metrics	How do we measure success?

Developing a Business Case

REST deliverables

- Web applications
  → It might be easier to start here
- Web services
  → Move here later
  - Services should have a RESTful design but support multiple interactions patterns (e.g., resource, method, and message oriented)
- Architectural patterns and infrastructure that supports robust capabilities for RESTful applications
- Governance program to foster good practices

Developing a Business Case

Business benefits of REST

- Improves usability and user experience
  - The application behaves the way you expect it to
  - You can share things with your friends just by sending them a URL
- Enables easier access to information and services
  - Low barrier to entry
  - Reachable by the largest possible audience
  - Consumable by the largest number of user agents
    → Desktop, web, mobile, mashup, programmatic clients
  - No special software required
  - Searchable via Google, Bing, etc,

Developing a Business Case

Business benefits of REST

- Increases agility
  - Extend the system at runtime
  - Alter resources without impacting clients
  - Direct client behavior dynamically
- Makes systems scalable, reliable, and high performing
  - Simple
  - Cacheable
  - Atomic

Developing a Business Case

Business benefits of REST

- Minimal investment required
  - REST *is* the Web – the infrastructure is in place
  - Very little additional middleware or technology required
- Simple programming model
  - Rapid development and delivery of applications

Business Case Secrets

Understand your audience

- Tailor the business case to address their pain points
- What keeps them up at night?
  - Cost containment
  - Time to market
  - Customer retention
  - Supporting mobile users
  - Enabling easier B2B integration
  - Easier access to information
  - Heterogeneous interoperability
  - Scalability concerns
- Position REST in these terms

Developing a Business Case

REST Costs

- Training and mentoring
  - A lot more than just learning a new framework
  - New development models and patterns
- Very little additional frameworks or middleware
  - Most frameworks are open source
  - Typically builds on existing Web infrastructure
- New and expanded use of Web infrastructure
  - May require additional/expanded licensing of Web infrastructure
    → Web access management, proxies, load balancers, gateways, etc
  - May require additional training
- Governance program

Developing a Business Case

REST Risks

- New and very different programming models
  - Tooling doesn't provide guard rails
- Misapplication of REST principles and constraints will not produce desired benefits
  - Simplicity, visibility, ease-of-use, serendipity, etc.
- Unfamiliarity with new QoS patterns causes security, reliability, integrity, performance, and usability risks
  - (This is why REST has a reputation as not robust)

Developing a Business Case

Roadmap and timeframe

- Basic training in REST principles and constraints
  - Don't underestimate training time
- Pilot project: web application
  - Learn how to use the Web the way it is intended
- Pilot project: web service
  - Pick a low-risk, reasonably visibly project
  - Start with a read-only service
  - Learn about resource models and the power of GET
- Follow-on projects: web services
  - Develop patterns to support QoS requirements
  - Develop infrastructure models to support these patterns

Developing a Business Case

Assumptions

- Web infrastructure is already in place
  → although it may not be used to its full potential

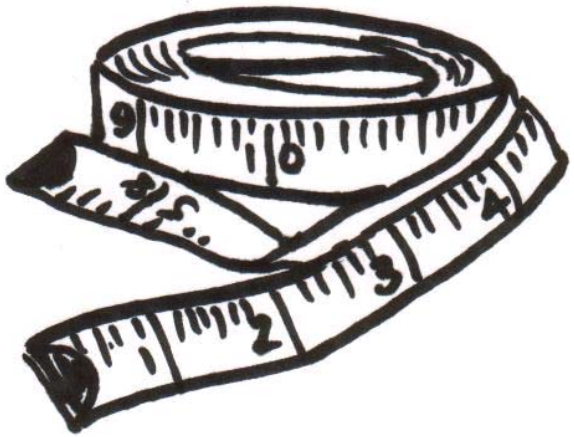
Alternatives

- For web applications:
  → what you use today w/ less than optimal design
- For web services:
  → WS-* and method-oriented model

Developing a Business Case

Metrics

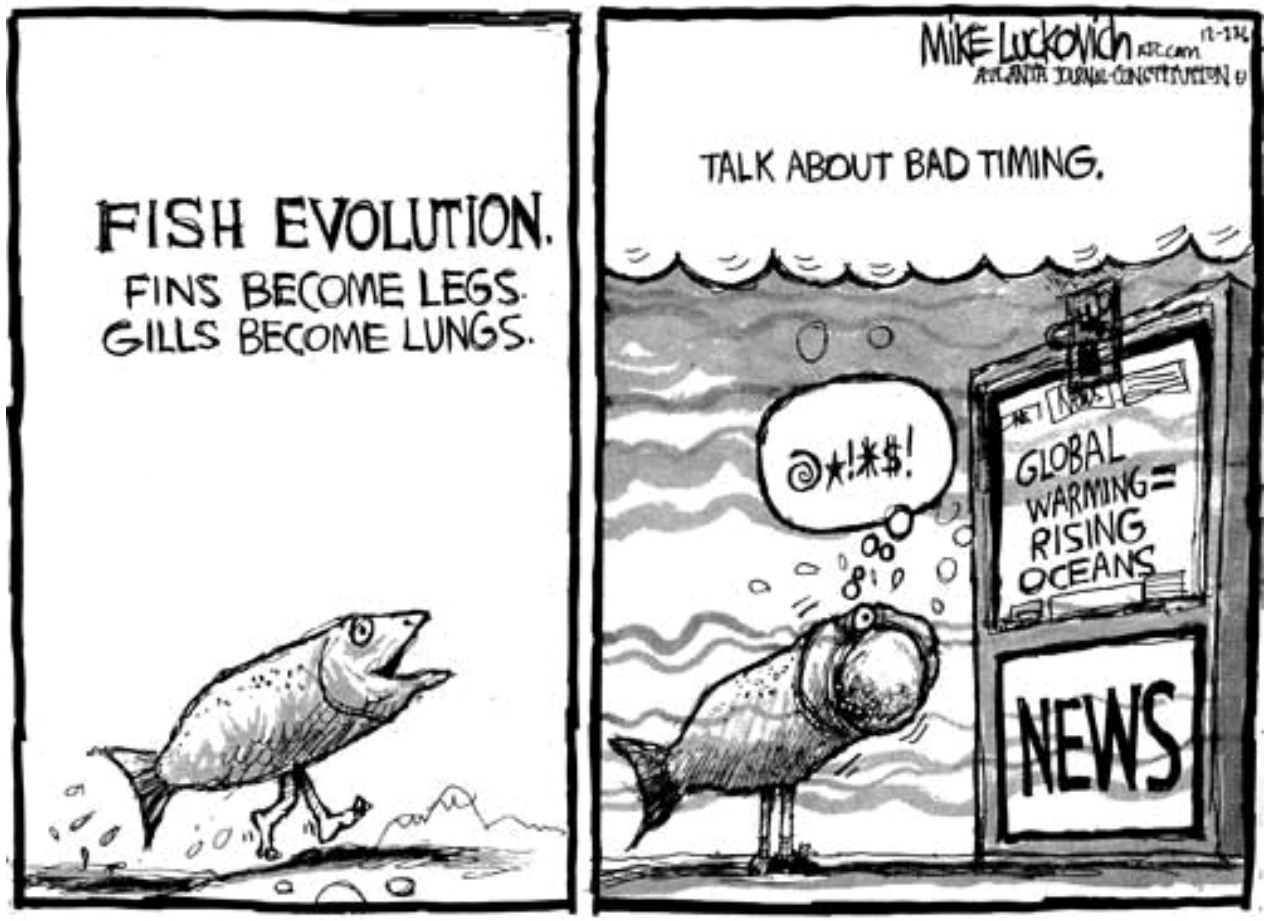
- How do you measure success?
- Metrics should correspond to the desired benefits defined in the business case
- The best metrics measure realized business value
  - Reduced costs
  - Increased revenues



Business Case Secrets

Timing may be crucial

- Avoid inopportunities and distracting initiatives



Business Case Secrets

Watch where you step

- Work within cultural constraints
- Determine how best to introduce a new approach
  - Recruit sponsors and supporters
  - Find external references and proof points

Deliver proof points

Selling REST to your Boss

Recap

- REST is the best architectural style to use for applications that run on the Web
  - Simpler, more elegant, Web-native applications
- REST is about design, not technology
  - Simple, but not necessarily easy
  - The risks are real
- You need to provide compelling business arguments
  - Tie your proposal to business goals
  - Then deliver demonstrable value