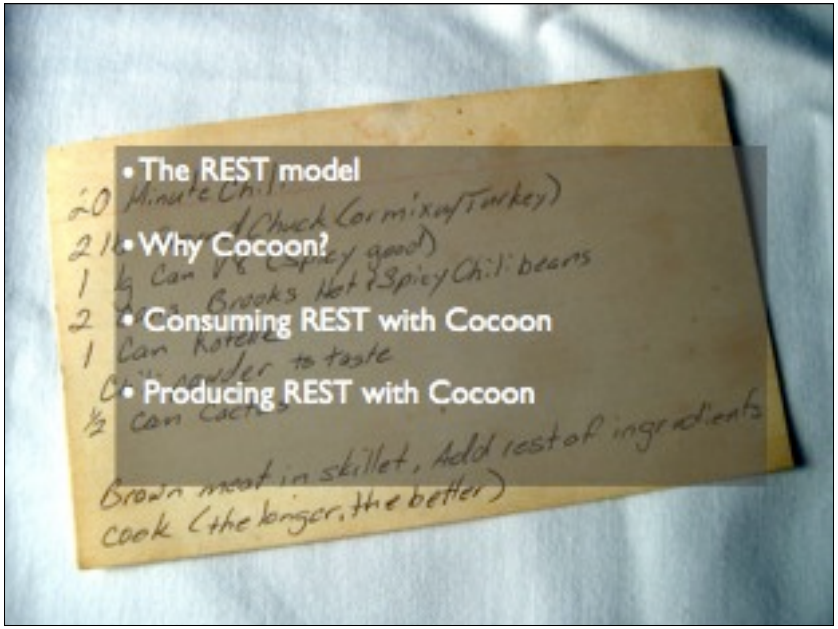
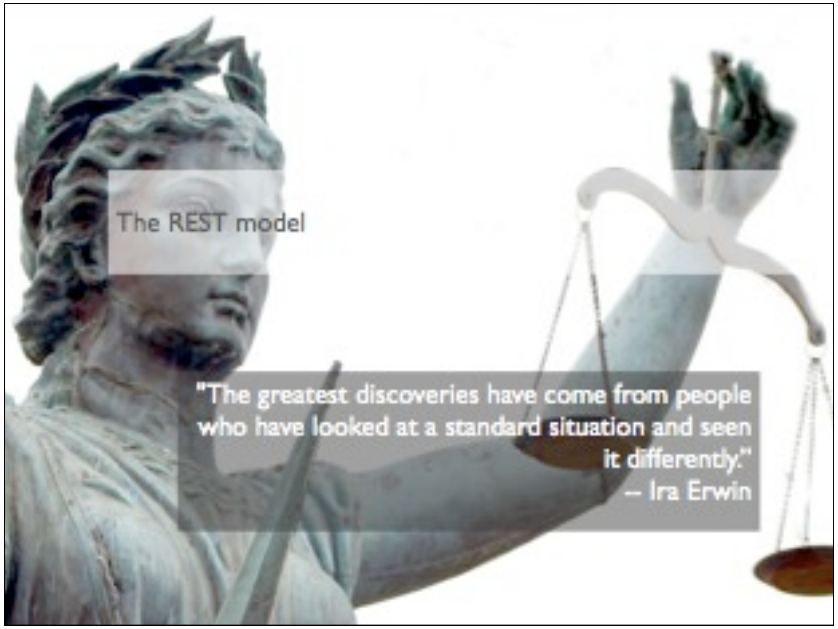
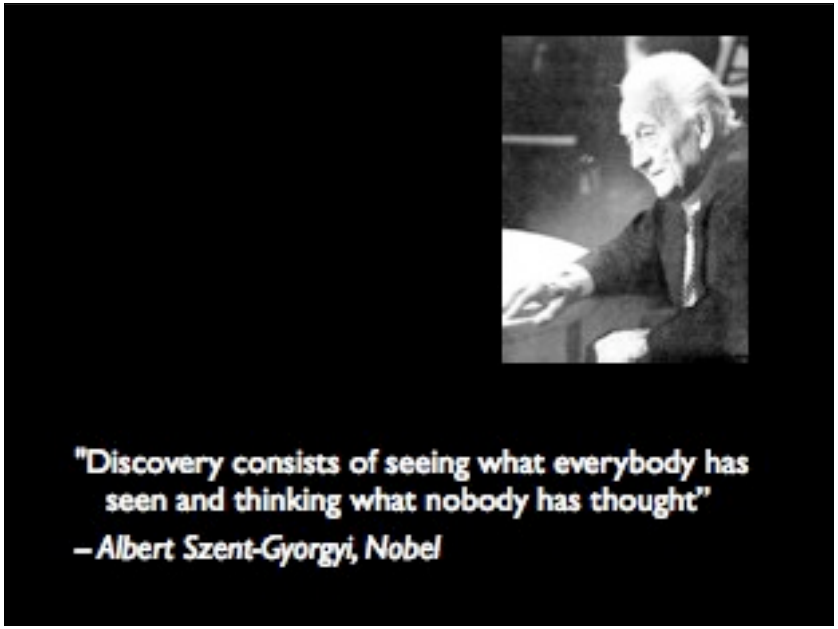

Slicing and dicing REST with Apache Cocoon
Gianugo Rabellino
ApacheCon US 2005


• The REST model
• Why Cocoon?
• Consuming REST with Cocoon
• Producing REST with Cocoon

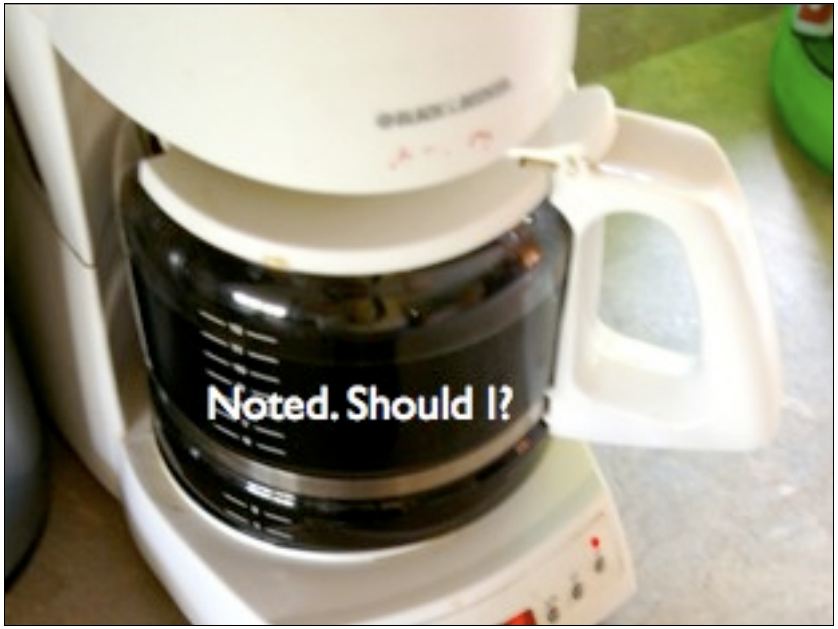

The REST model
"The greatest discoveries have come from people who have looked at a standard situation and seen it differently."
-- Ira Erwin


"Discovery consists of seeing what everybody has seen and thinking what nobody has thought"
– Albert Szent-Gyorgyi, Nobel


It's the Web, stupid!


What about SOAP, then ?

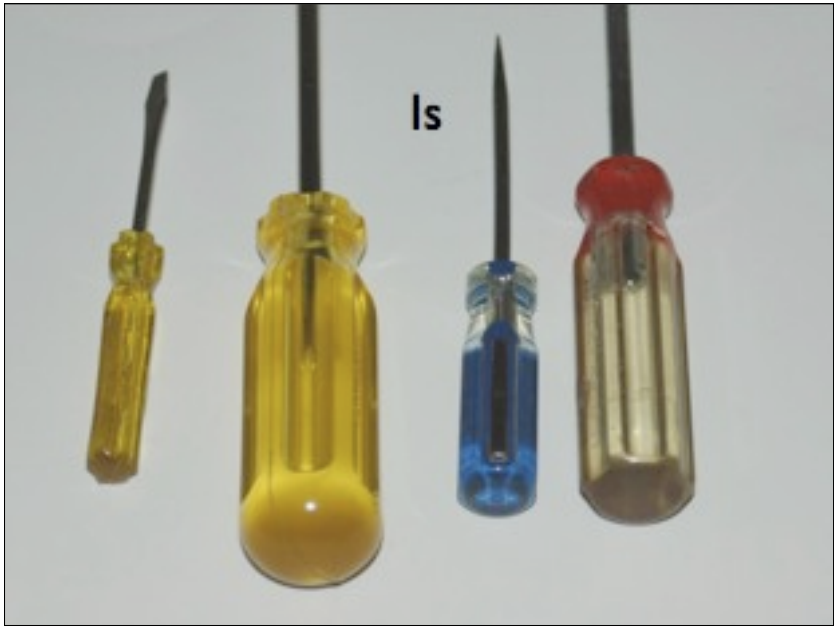

You can brew your coffee using SOAP!


Noted. Should I?

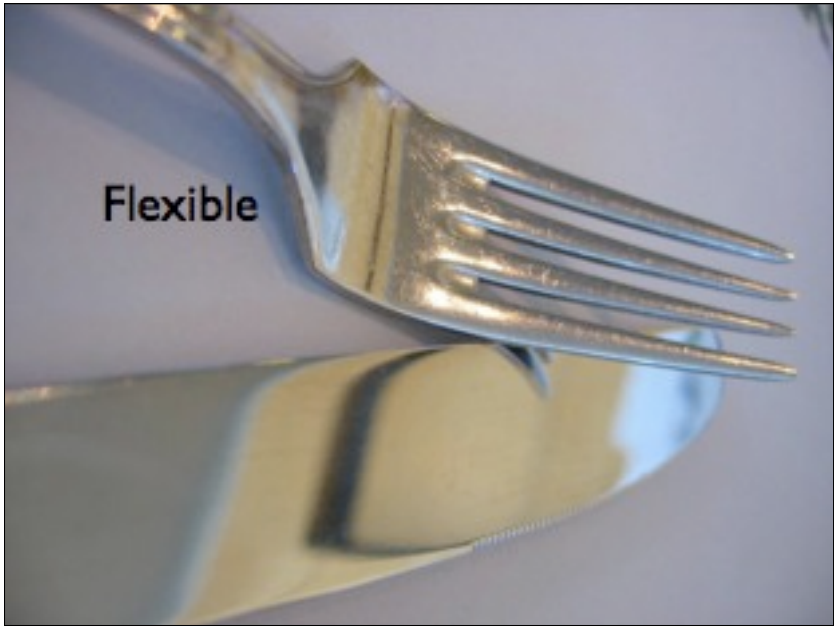

Soap is just fine

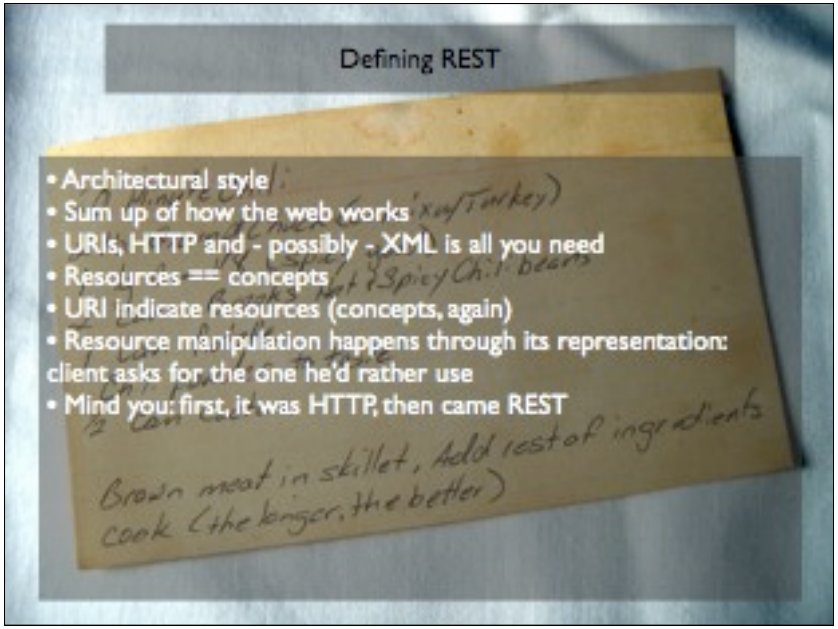

As long as you have the tools


REST, on the other hand…


Is


More


Flexible


Defining REST
• Architectural style
• Sum up of how the web works
• URIs, HTTP and - possibly - XML is all you need
• Resources == concepts
• URI indicate resources (concepts, again)
• Resource manipulation happens through its representation: client asks for the one he'd rather use
• Mind you: first, it was HTTP, then came REST


The role of XML
• REST doesn't dictate a format
• But XML is still the most flexible:
– Structured data
– Multiple representations (XSLT, anyone)

Flow

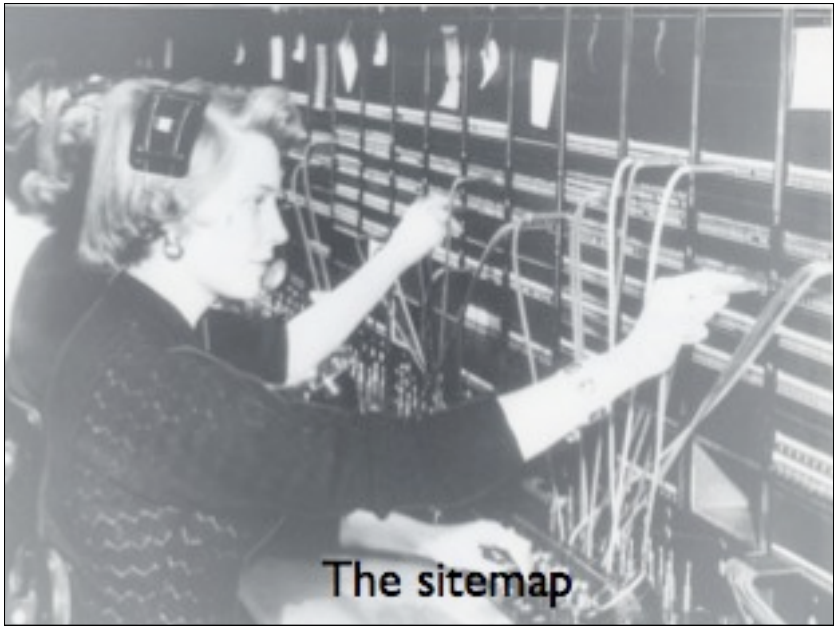
- This is the web, remember?
- All you need are (X)*links*
- And keep the state on the client

What about objects?

- Binding
- Or just forget about it altogether


Cocoon and the REST world

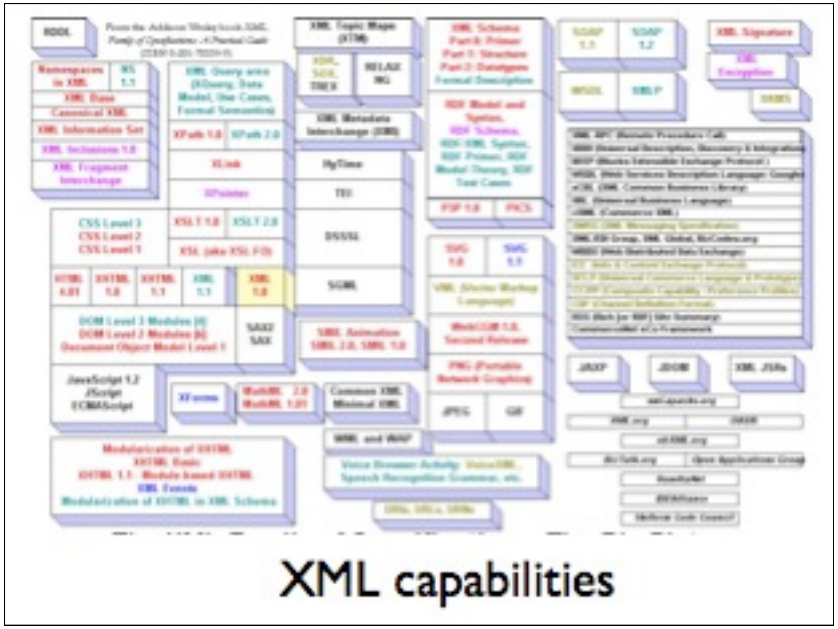

URLs with a clue
http://www.hallmark.com/ECardWeb/ECardPickupView.jsp?
&n=%60jmn&e=%01%7bx%7e%7cqIju%7ev7vr%7d7nm%7e
&s=wo%04%3a%40%3c%3c&p=Qn%03%298jvrn%29%2f%29
Snw%2966%29Qjyy%03%29%5dqjwt%7cpr%7frwp%2a%29%5c
jmu%035%29fn%7frw%29jwm%29R%29%01n%7bn%29wx%7d
%29jkun%29%7dx%29pn%7d%29qxvn%29%7dx%29%7cynwm
%29%7dqn%29qxurmj%03%29%01r%7dq%29Vxv%29%2f
%29Njm7%29Uxwp%29%7c%7dx%7b%035%29%7c%7dj%7b
%7drwp%29%01r%7dq%29%7dqn%29%7cwx%01%29jwm
%29lxum%29rw%29Lqrljpx5%29%01qrlq%29vnjw%7d%29%01n
%29lx%7eumw0%7d%29lqnlt%29rw%29%5bqxm%037%16%13
%16%13Qxyn%29%03x%7e0%7bm%29qj%7frwp%29j%29p%7bnj
%7d%29qxurmj%037%29%60ruu%29%7d%7b%03%29%7dx
%29ljuu%29%03x%7e%29uj%7dn%7b7%16%13%16%13Ux
%7fn5%16%13%60jmn%29%2f%29fn%7frw&f=y




```
<map:match pattern="products/*.html" type="wildcard">
    <map:generate src="products/product_{1}.xml" type="file"/>
    <map:transform src="products2html.xsl" type="xslt"/>
    <map:serialize type="html"/>
</map:match>

<map:match pattern="products/*.xml" type="wildcard">
    <map:generate src="products/product_{1}.xml" type="file"/>
    <map:transform src="products2REST.xsl" type="xslt"/>
    <map:serialize type="xml"/>
</map:match>
```

HTTP friendly
(well, mostly)

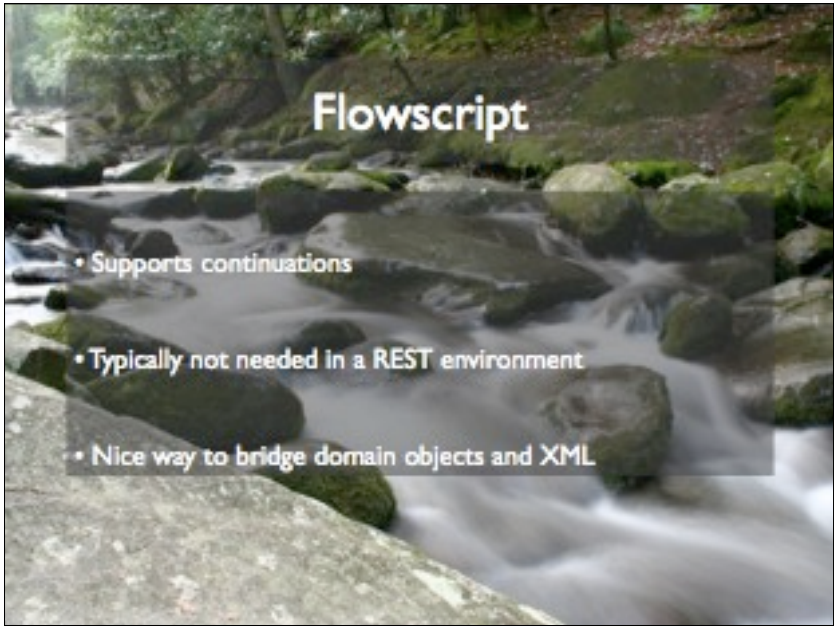

XML capabilities


Componentized resource assembling


Lego™ bricks, anyone?


Semantic capabilities

Multi-stage approach




Side orders
• XMLBeans
• JAXB
• Javascript (E4X?)