



Enterprise IT vs. WWW

Stefan Tilkov | @stilkov | JUG Ostfalen | AutoUni 31.03.2012





2. iSAQB Architecture Days/

ObjektSPEKTRUM Information Days

„Softwarearchitektur im Team. Softwarearchitektur im Wandel“

20.-22.11.2012

Call for Papers: osid2012@isaqb.org



<http://www.innoq.com>

**Beratungsunternehmen mit Schwerpunkt auf
Softwarearchitektur & effizienter Entwicklung**

Gründungsmitglied iSAQB e.V.

~60 Mitarbeiter (Q1 2012)

Beratung, Schulung, Entwicklung

Java/Java EE, Ruby on Rails u.a

Standorte in Düsseldorf, Frankfurt, München, Zürich

@stilkov



(Prelude)

REST

RESTful HTTP

URIs Identifies Resources

`http://example.com/orders?year=2008`

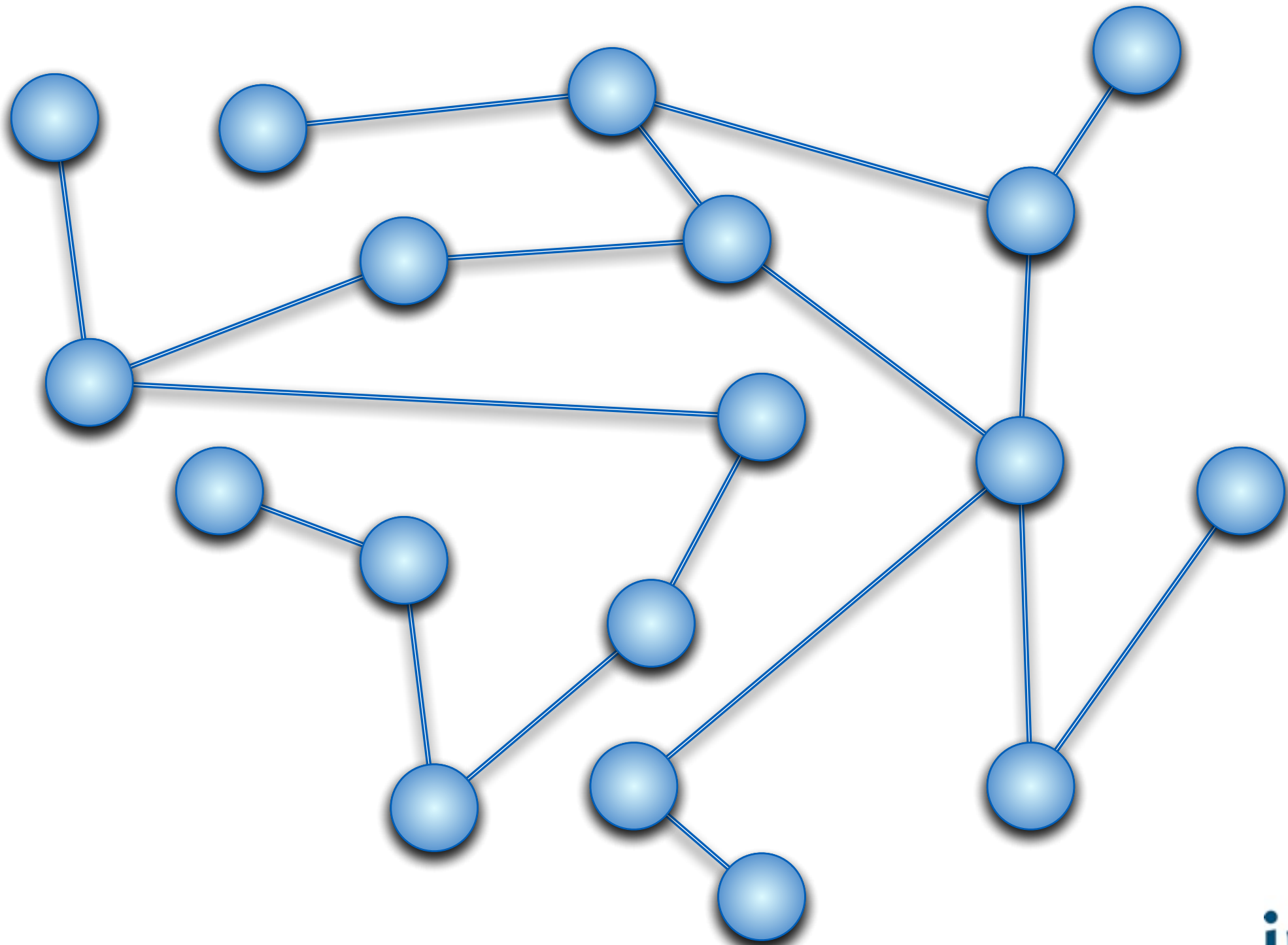
`http://example.com/customers/1234`

`http://example.com/orders/2007/10/776654`

`http://example.com/products/4554`

`http://example.com/processes/sal-increase-234`

Resources are Linked



Representations in different Formats

XML

JSON

YAML

HTML

XHTML

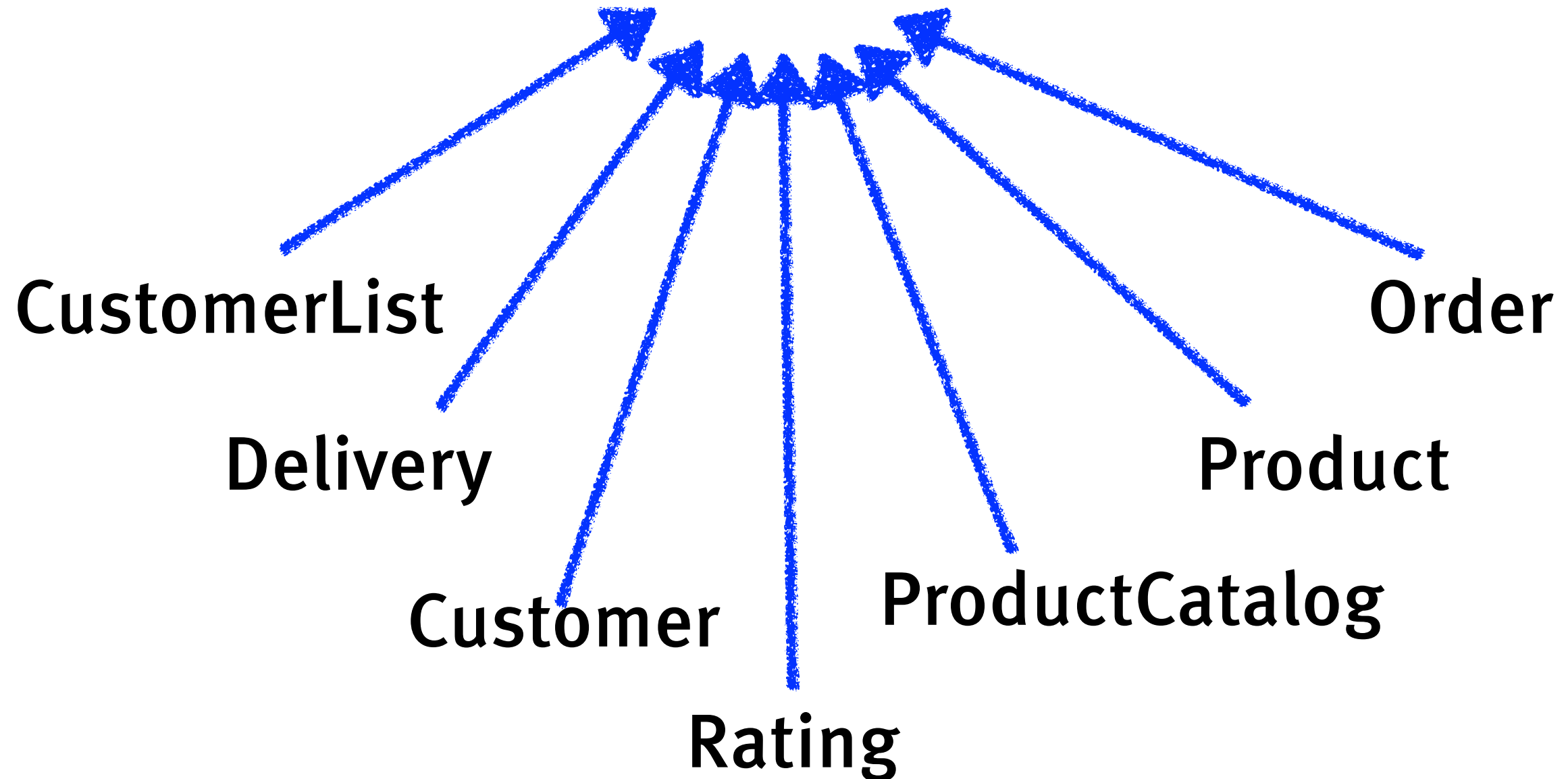
PDF

Binary

Plain Text

«interface»

Resource



«interface»

Resource

GET

PUT

POST

DELETE

A photograph of a tunnel entrance. The tunnel is arched and lined with reddish-brown bricks. The walls are covered in a dense network of white and grey cables or pipes. The floor is dark and uneven. A bright light source is visible at the far end of the tunnel, creating a strong glow. The word "Tunneling" is overlaid in large white letters.

Tunneling

(a.k.a. The SOAP Way)

POST **http://example.com/Customermgmt**

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

Method

ID

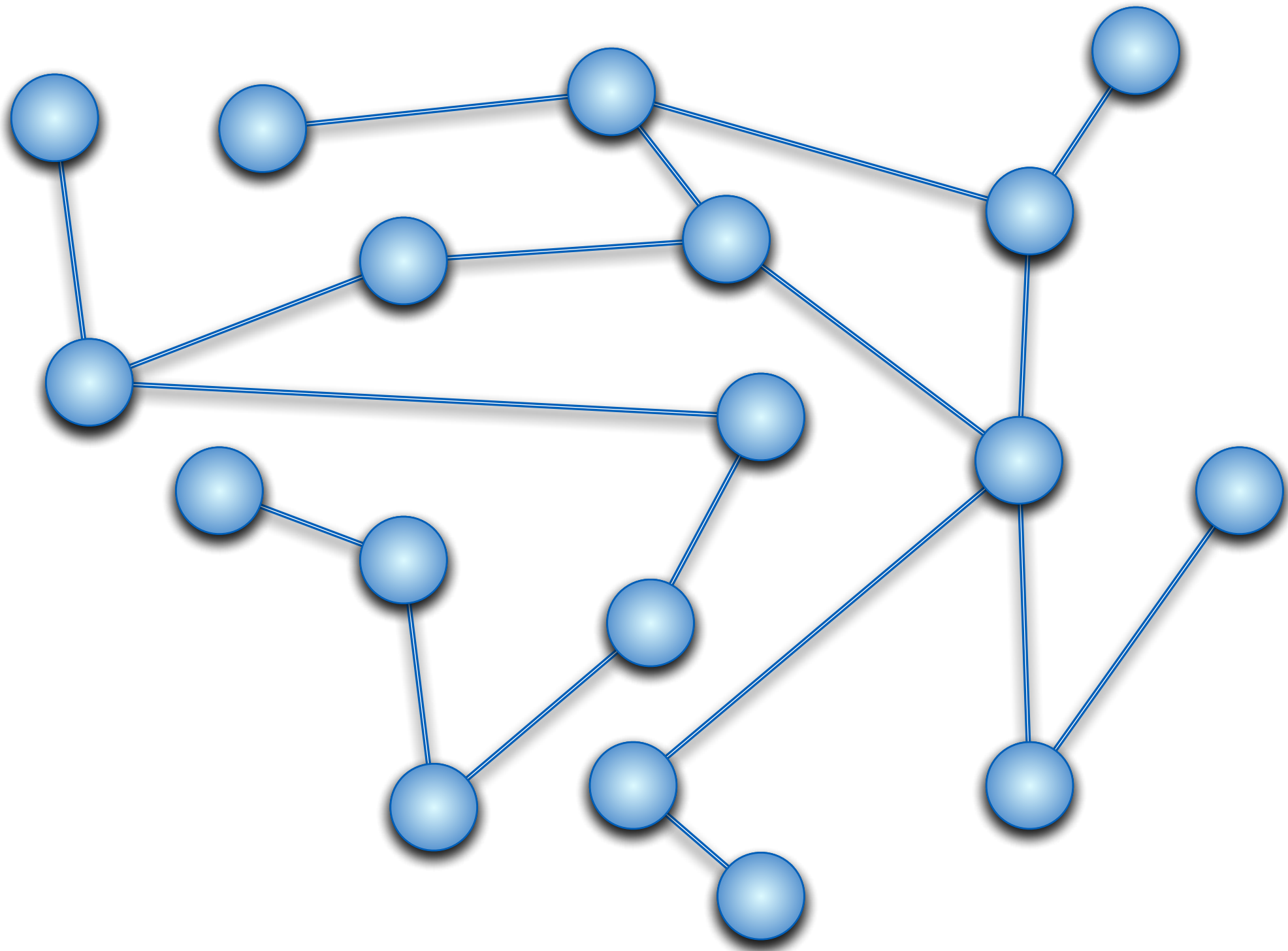
Endpoint



ENTER

DEAD
END





Enterprise

Web

"Here are the missing parts of the Web!"



"Thank you for nothing."



@stilkov

innoQ

<http://rest-http.info>



@stilkov

innoQ

The Human Web

amazon.com®

YAHOO!®

twitter



ebay

facebook®

XING®

Google™

Microsoft®

innoQ

@stilkov

Resources
Pretty URLs
Bookmarks
Collaboration
Caching
Hypermedia

Web Sites vs. Web Applications

Using the Web vs. Abusing the Web

There should be no applications

OrderEntryForm

ProductMaintenanceForm

CustomerView

Order

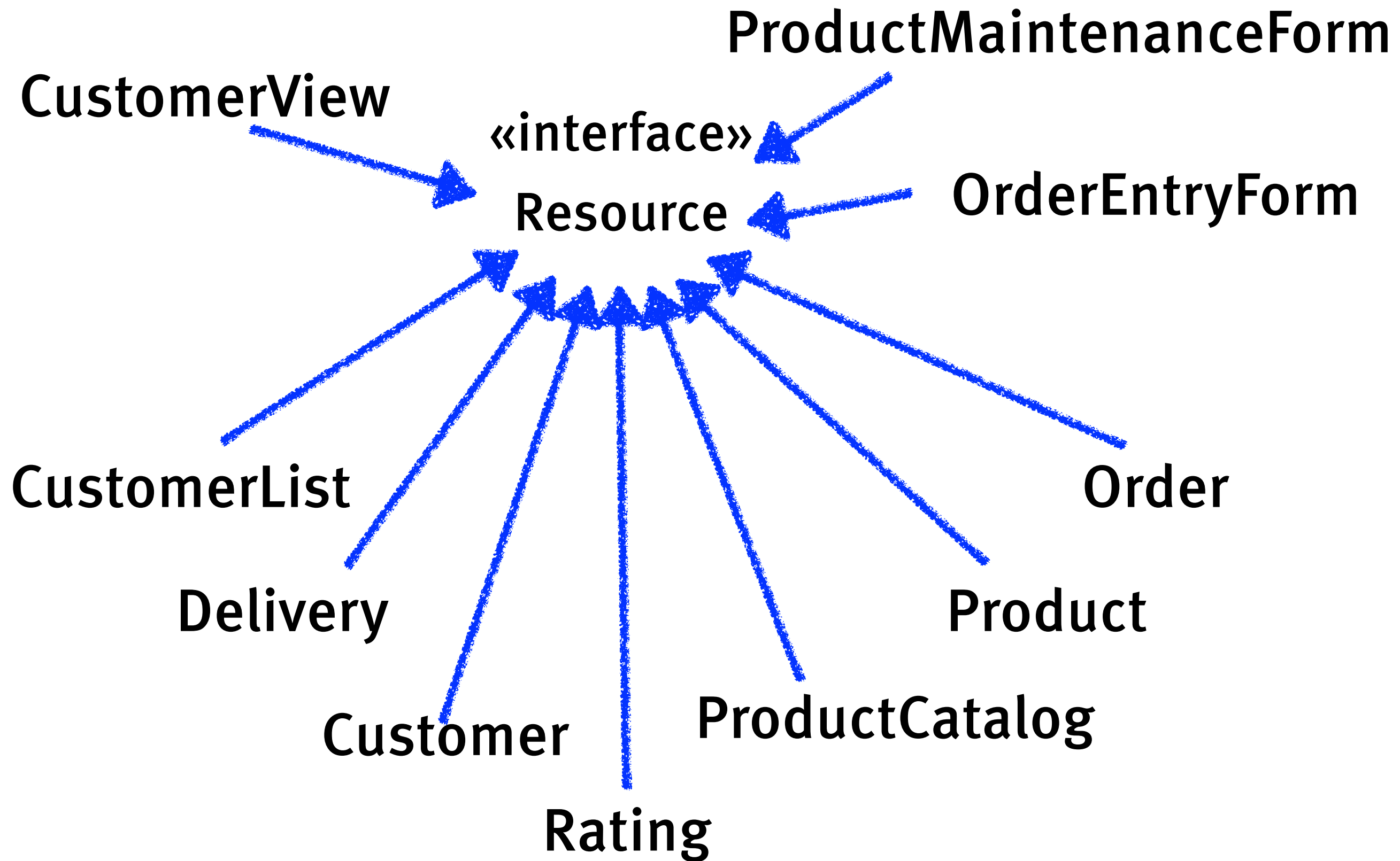
Delivery

CustomerList

Customer

Product

ProductCatalog



Stop hiding the truth



@stilkov

innoQ

© 2012 innoQ Deutschland GmbH

Thursday, May 31, 12

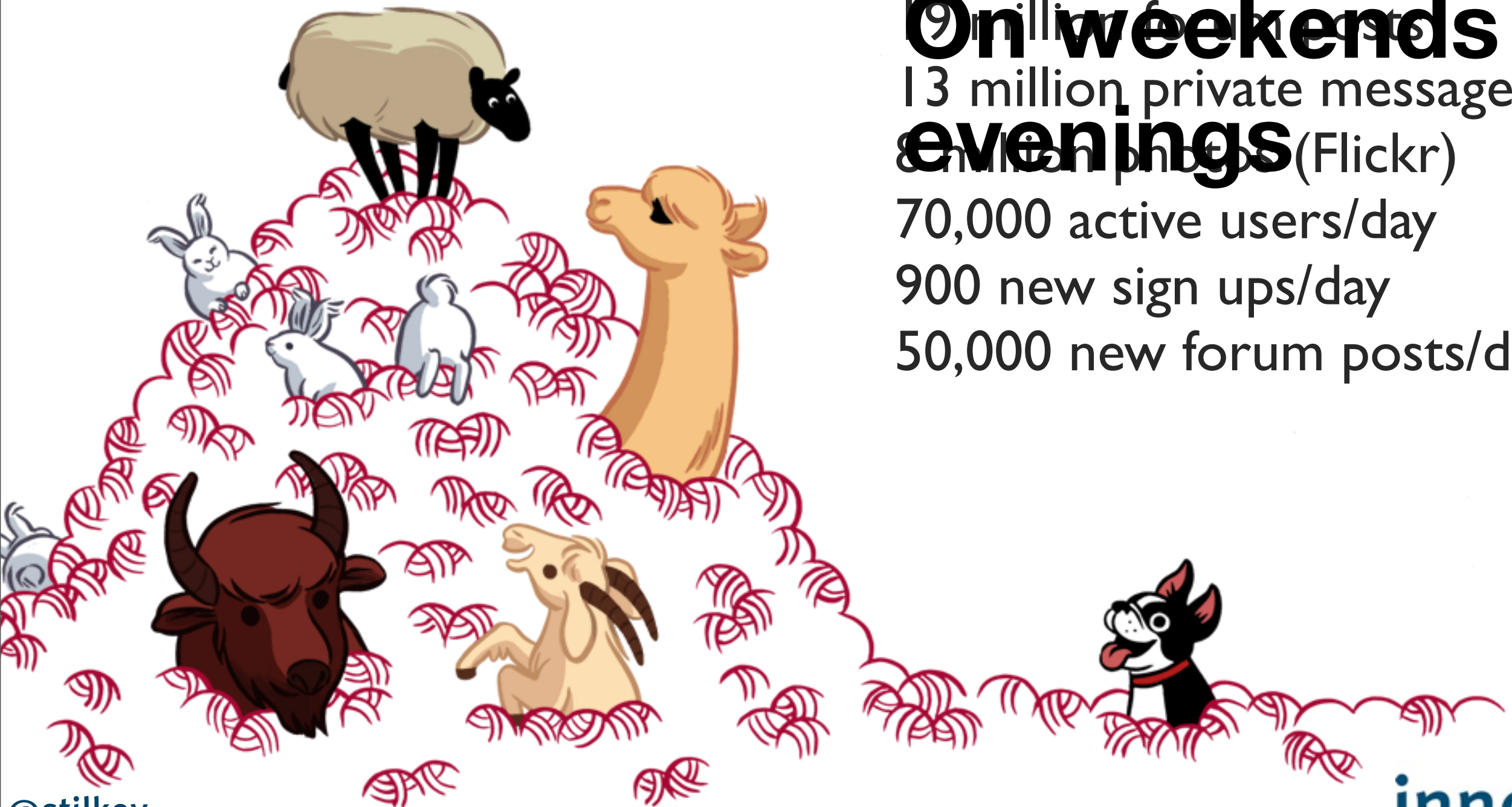
URIs
HTTP
HTML
CSS
Javascript

Syndication
Web Servers
Caching Proxies
Load Balancers
CDNs





10 million requests/day
4 Months
3.6 million pageviews/day
430,000 registered users
1 Developer
2.3 million projects
On weekends & evenings
19 million forum posts
13 million private messages
8 million photos (Flickr)
70,000 active users/day
900 new sign ups/day
50,000 new forum posts/day



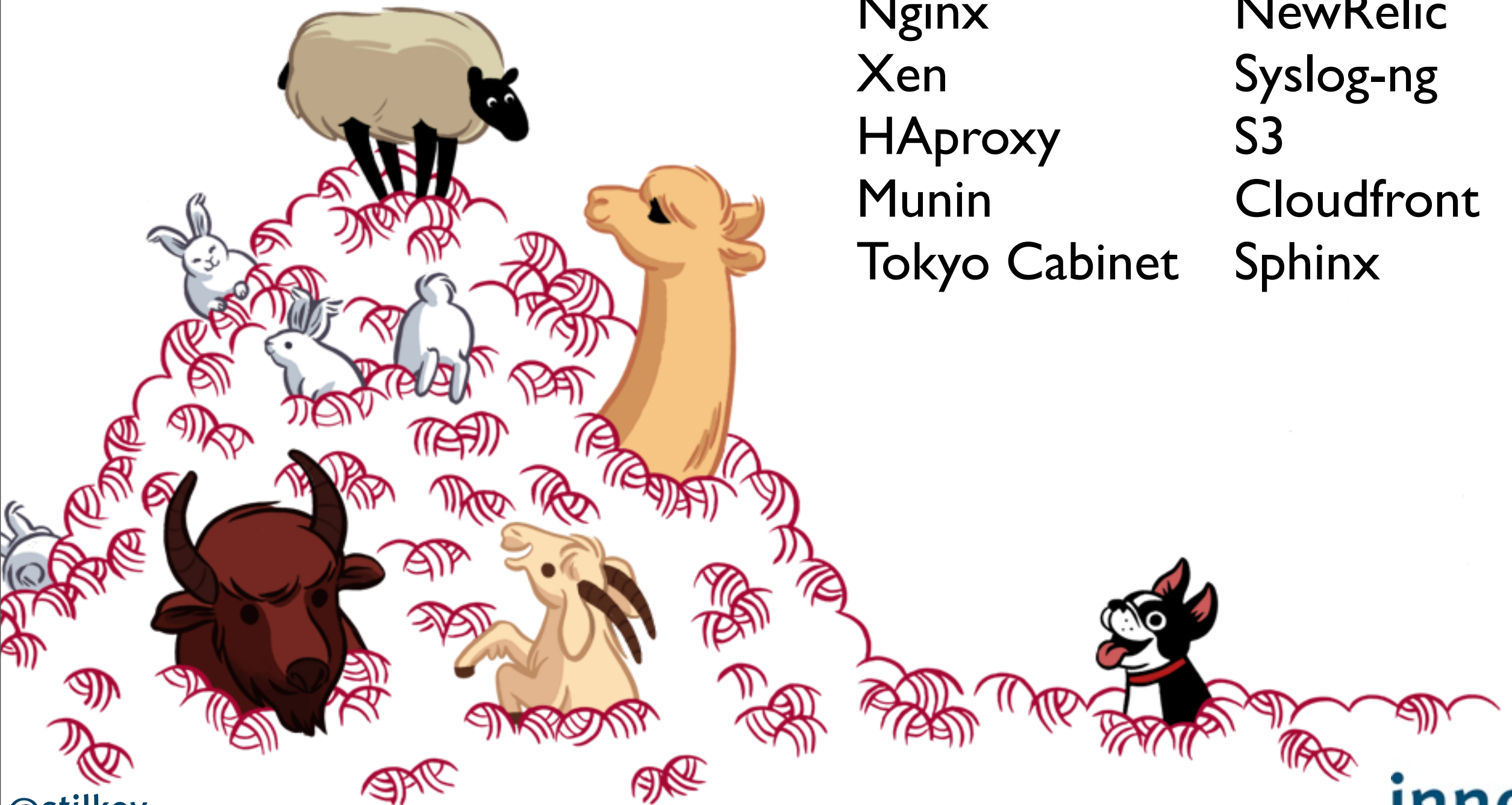
@stilkov





Ruby on Rails
MySQL
Gentoo Linux
Capistrano
Nginx
Xen
HAproxy
Munin
Tokyo Cabinet

Memcached
Tyrant
Nagios
HopToad
NewRelic
Syslog-ng
S3
Cloudfront
Sphinx



@stilkov

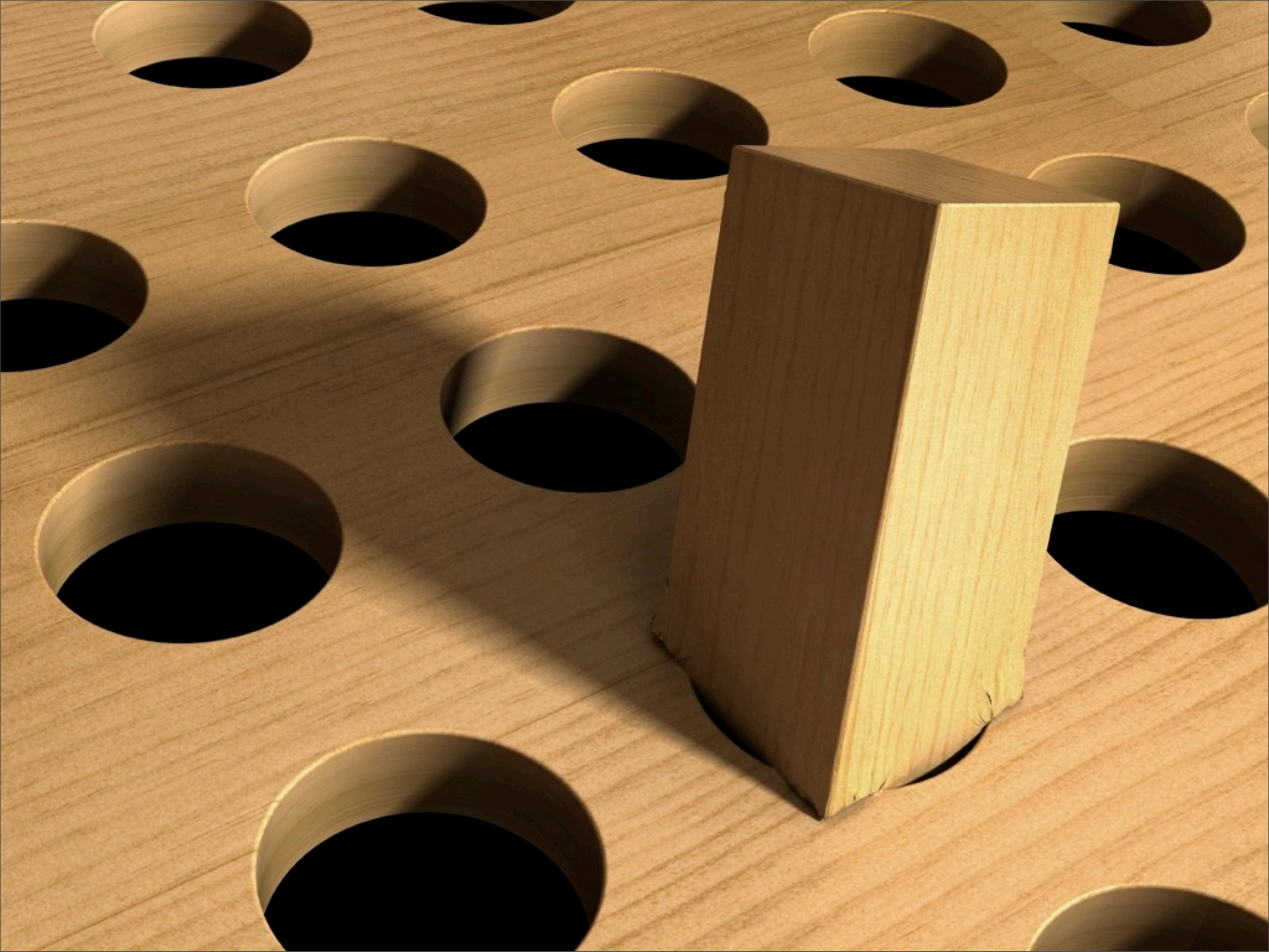




What I'm writing here is the single most important take-away from my Sun years, and it fits in a sentence: The community of developers whose work you see on the Web, who probably don't know what ADO or UML or JPA even stand for, deploy better systems at less cost in less time at lower risk than we see in the Enterprise.

Tim Bray

<http://www.tbray.org/ongoing/When/201x/2010/01/02/Doing-It-Wrong>



“My Internet is bigger than your enterprise.”

Paraphrasing Dare Obasanjo,
see <http://tinyurl.com/dare-enterprise>

Scale vs. Complexity

Inherent Complexity

VS.

Incidental Complexity

Portal Server

Portlets

Web Framework

Transfer Objects

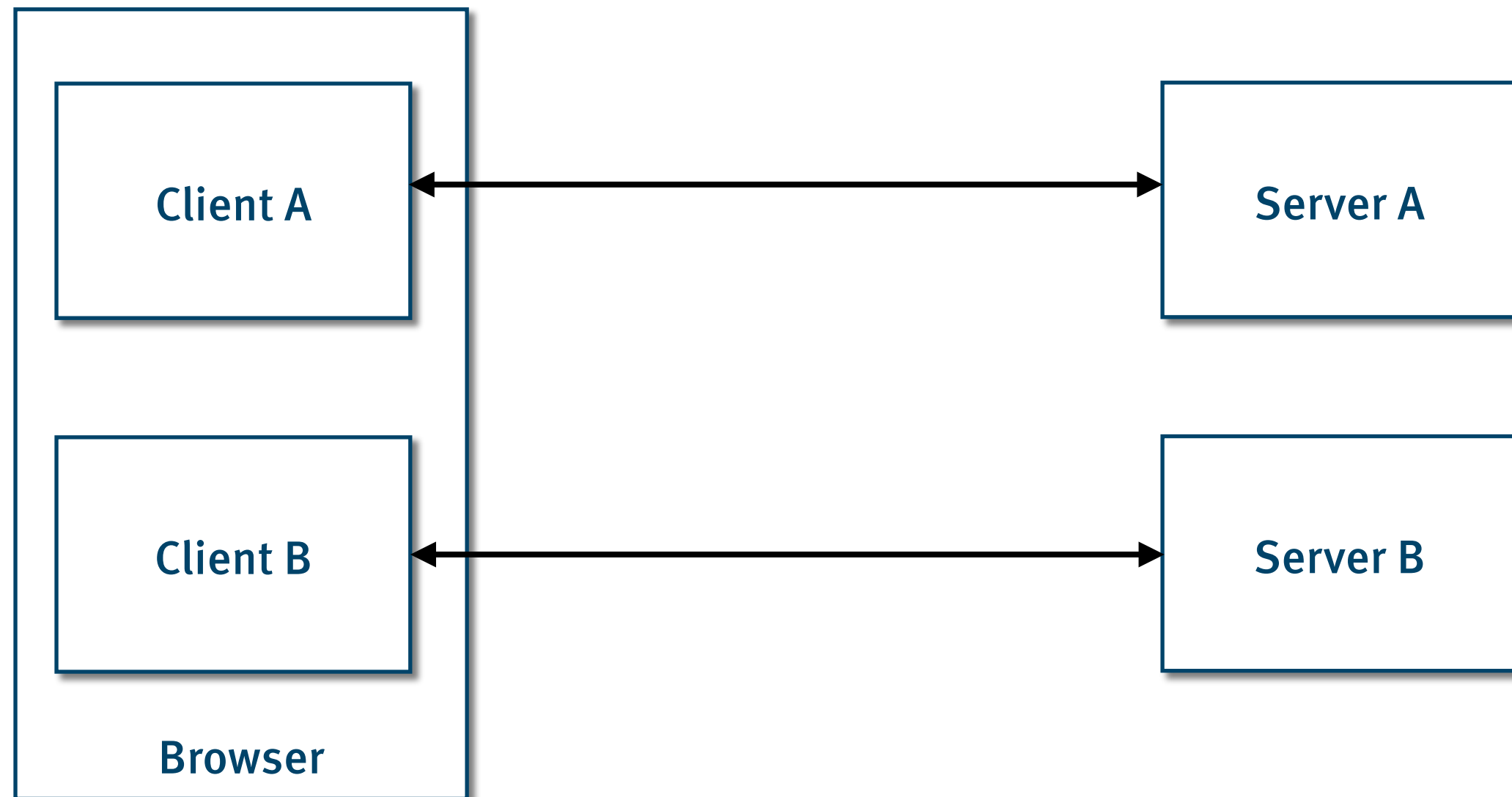
Session Beans

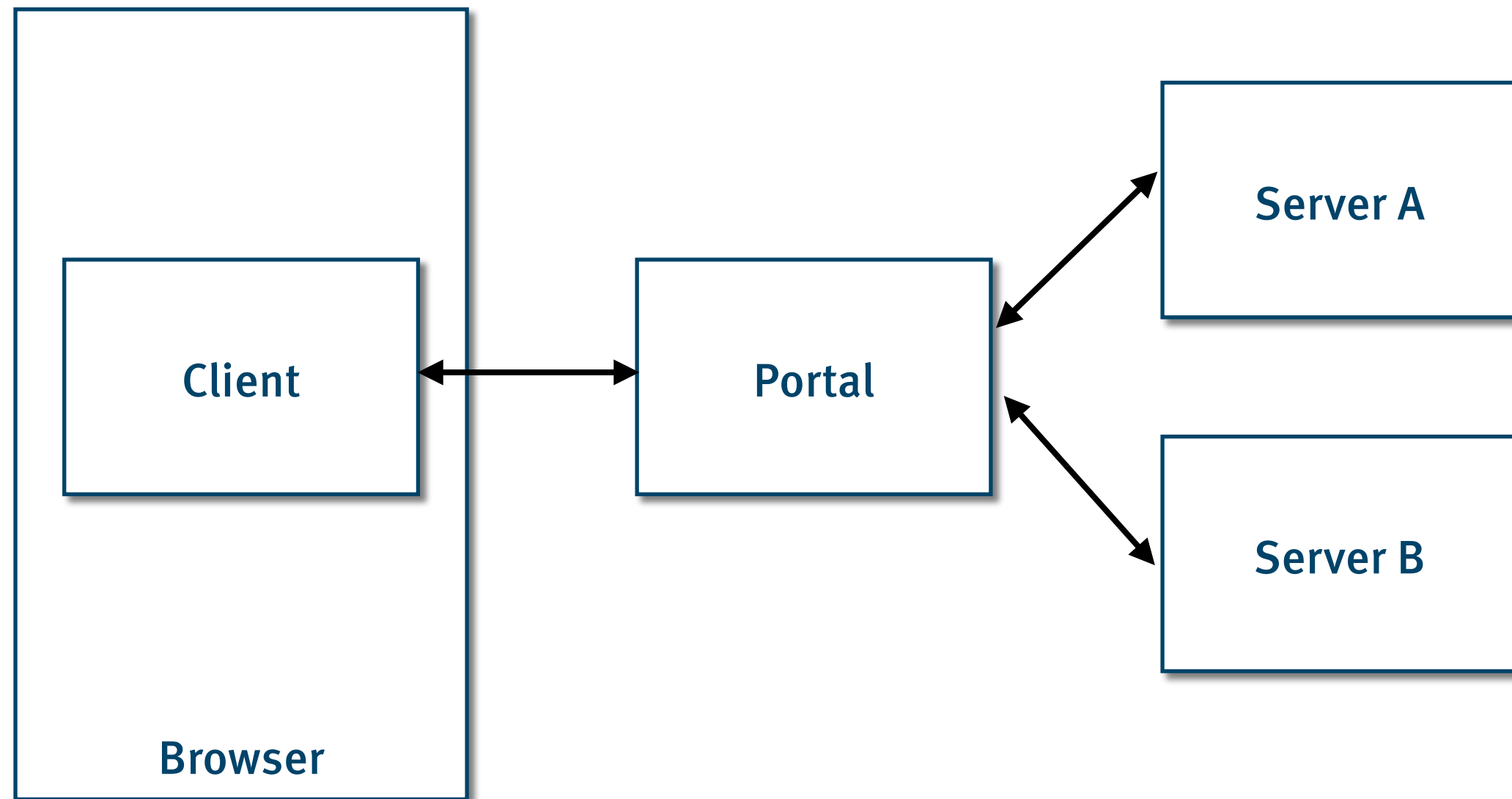
JAXB Java Beans

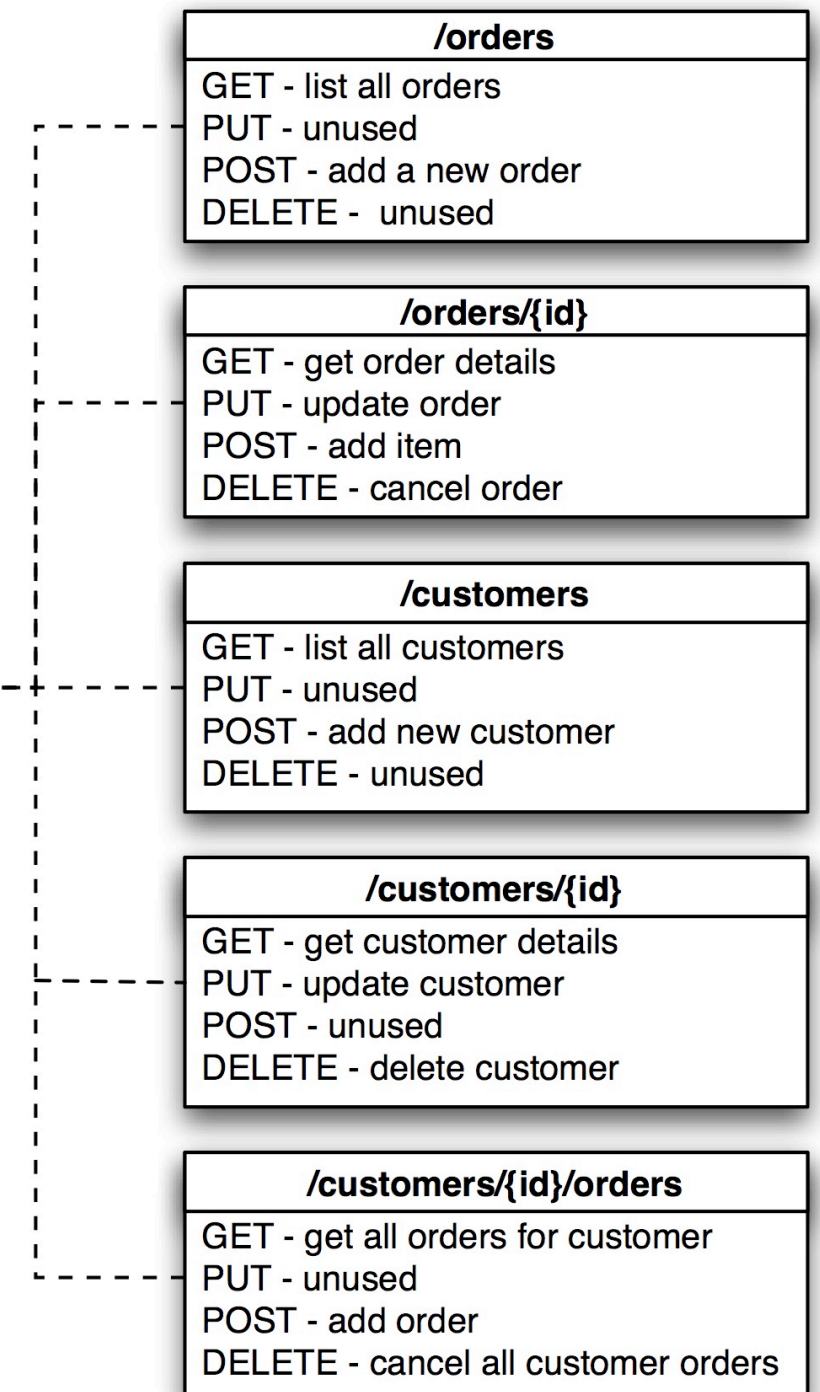
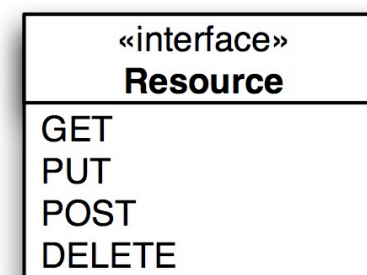
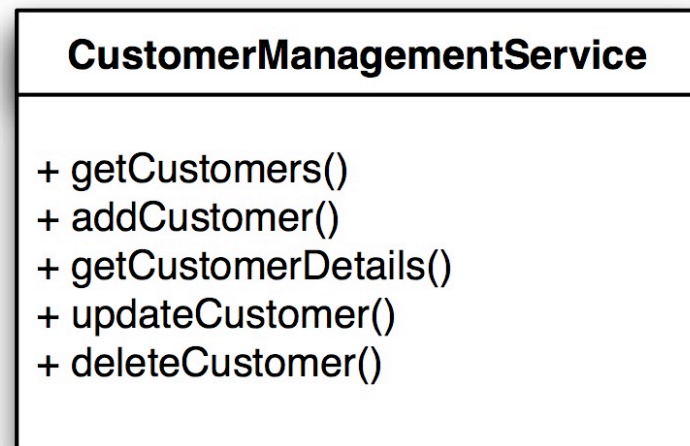
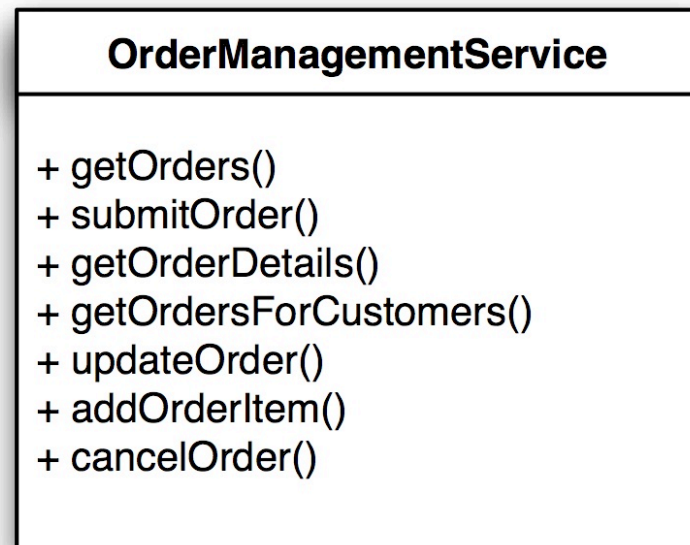
XML Stock Quotes

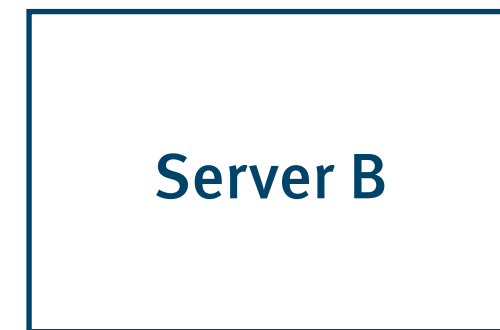
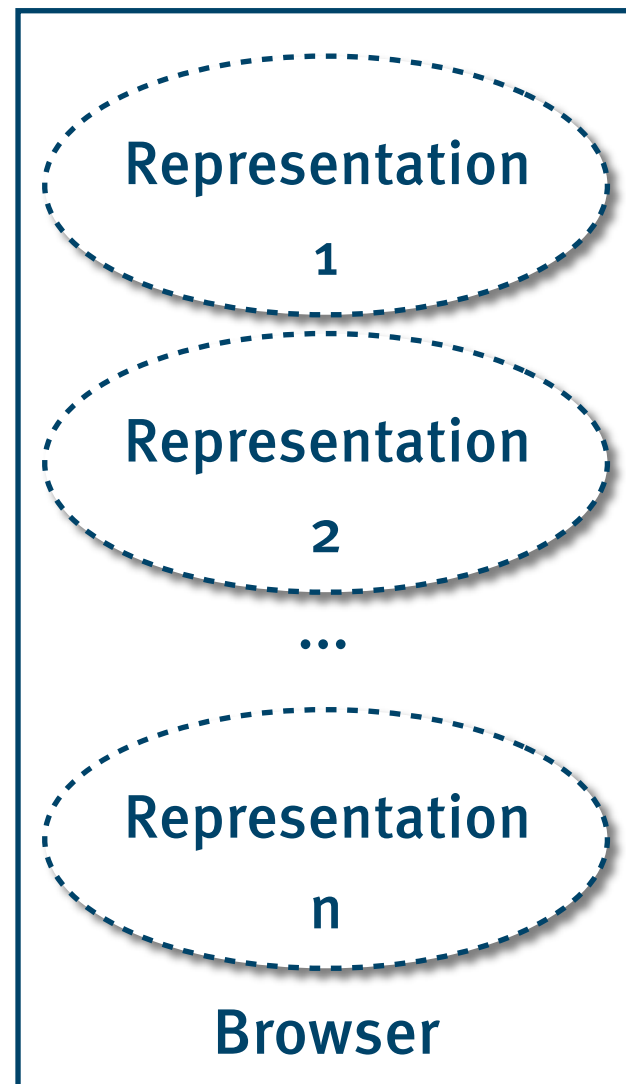


What's in an app?

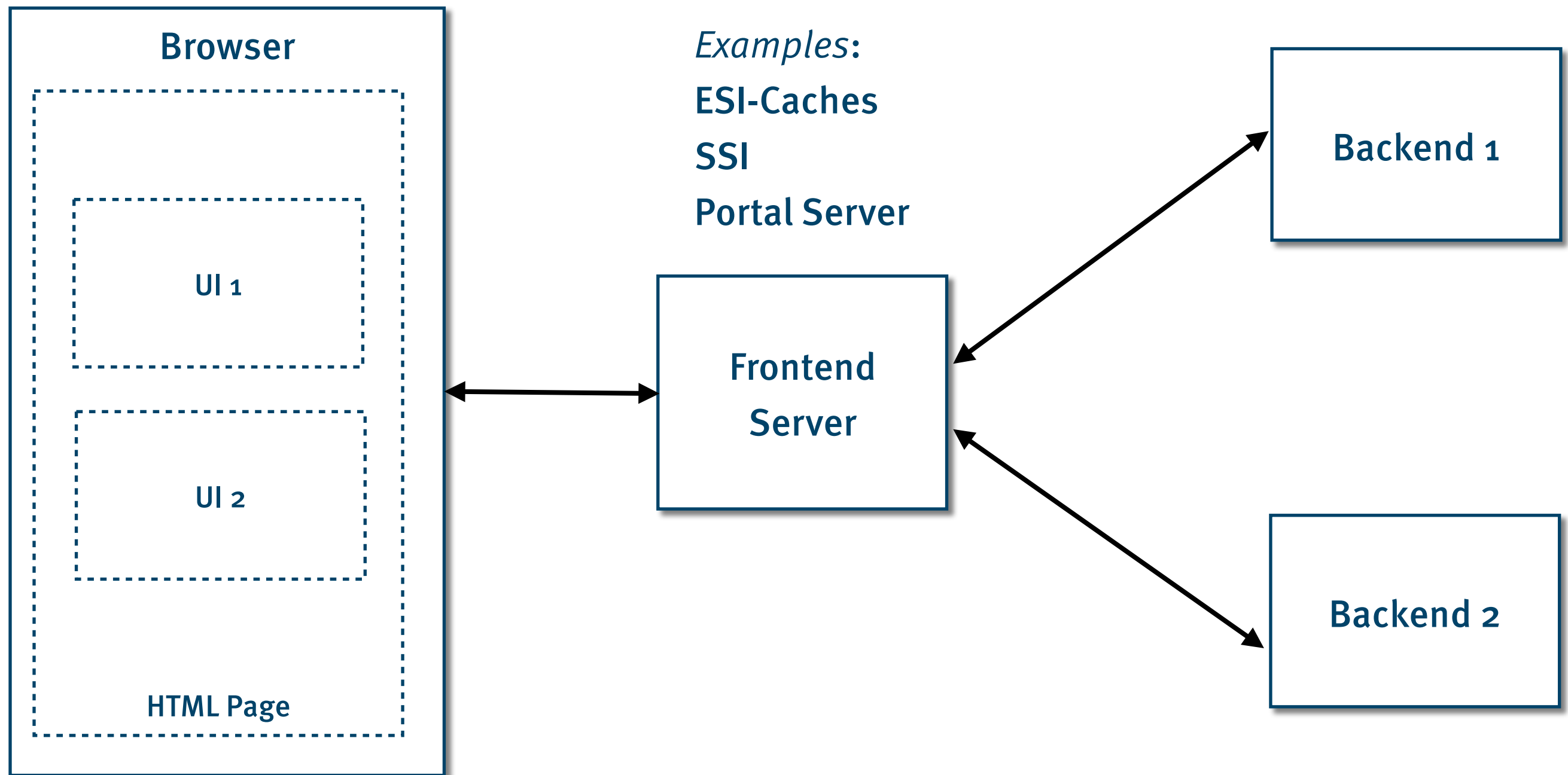




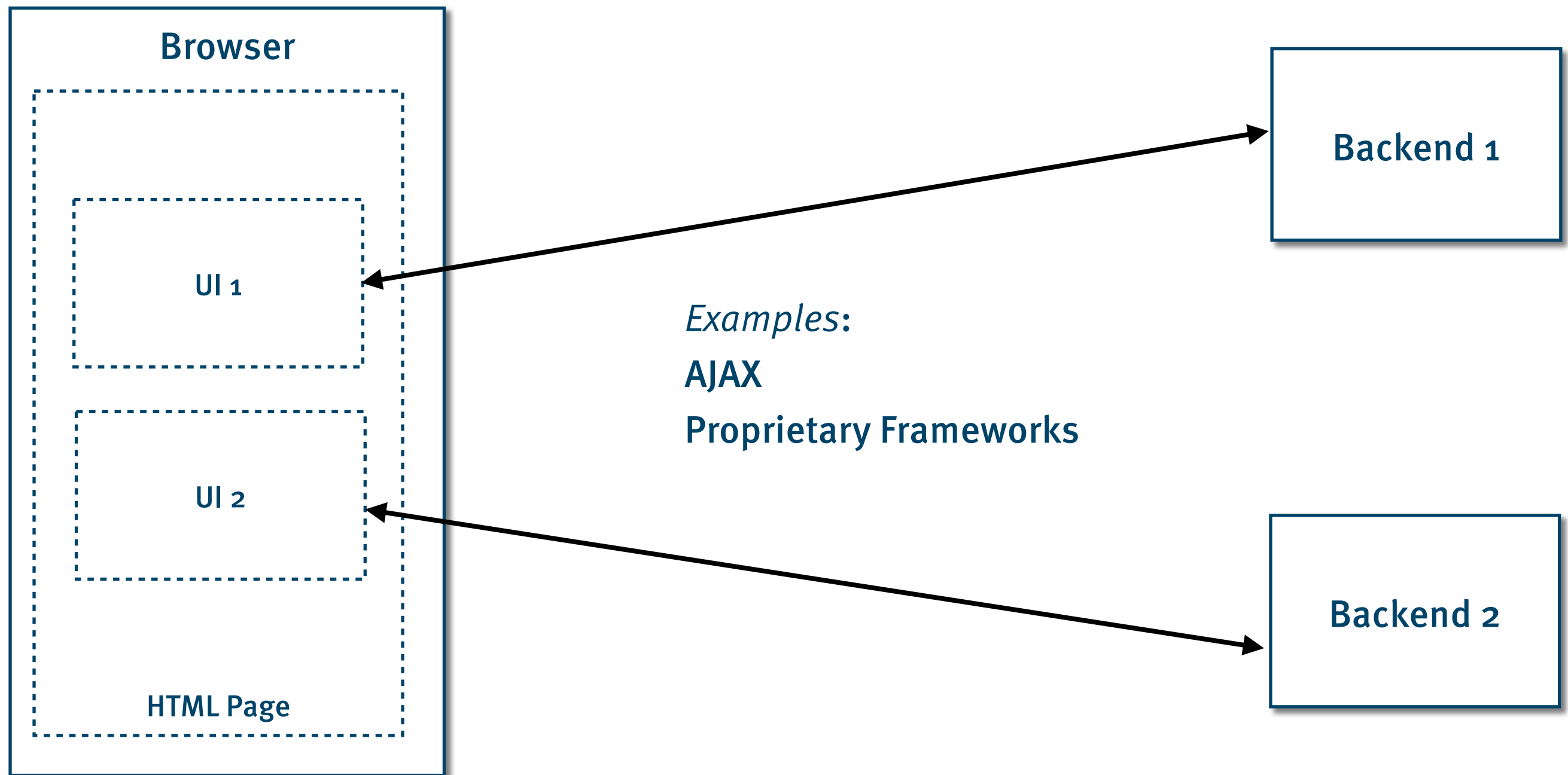




Server-side integration



Client-side integration



Multiple tabs

Multiple windows

Menus

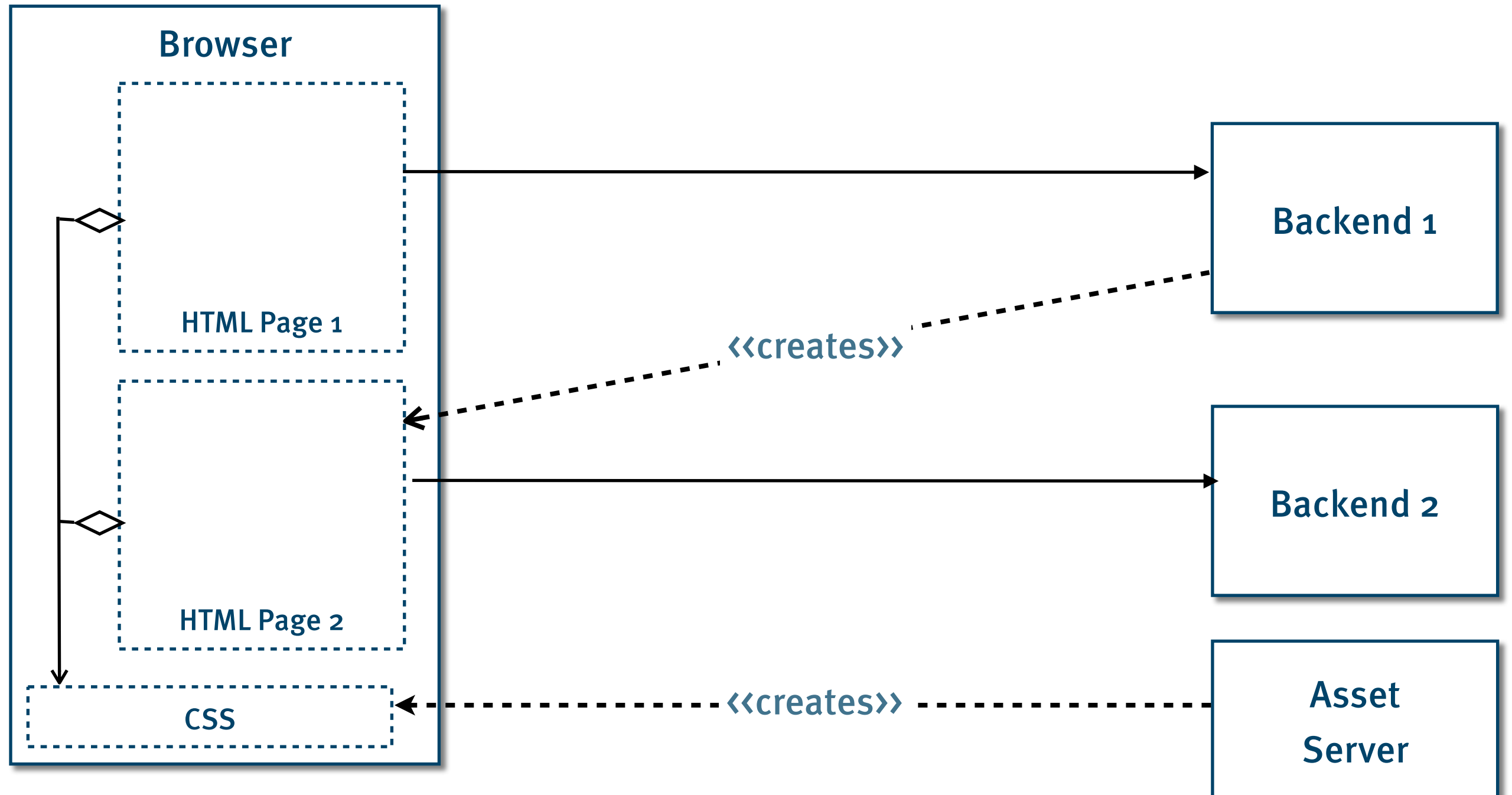
Bookmarks

Personalization

UI Integration

My browser is my portal

Links



Choosing a Web framework

How much HTTP would you like with that?

Server-side component-based

Wicket

JSF

Tapestry

Vaadin

ASP.NET

Request/Response-oriented

Rails

Struts

ASP.NET MVC

PHP

Play

Django

Grails

Spring MVC

Rich-client

Silverlight

Applets

Flash/Flex

Echo

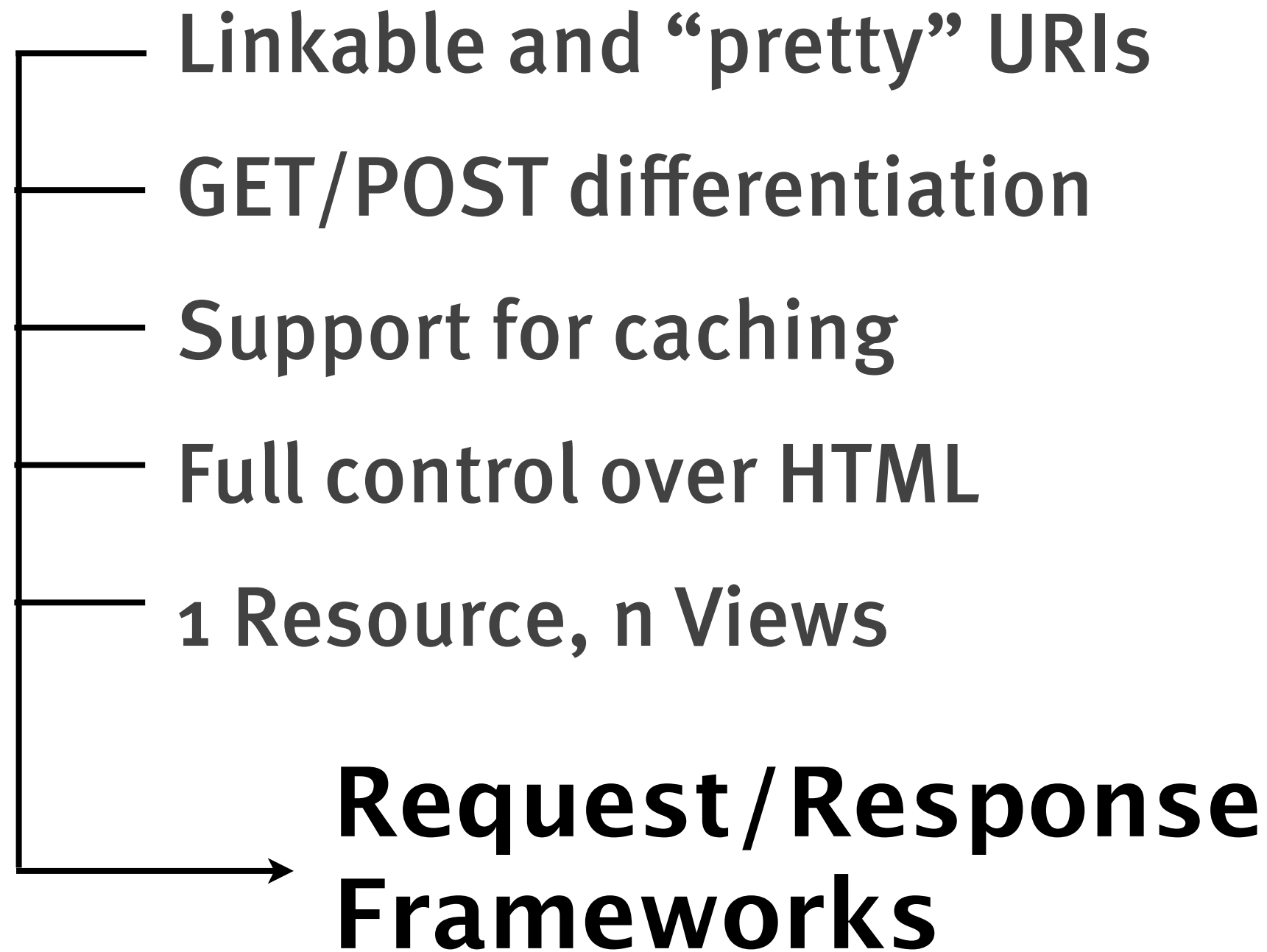
JavaFX

ZK

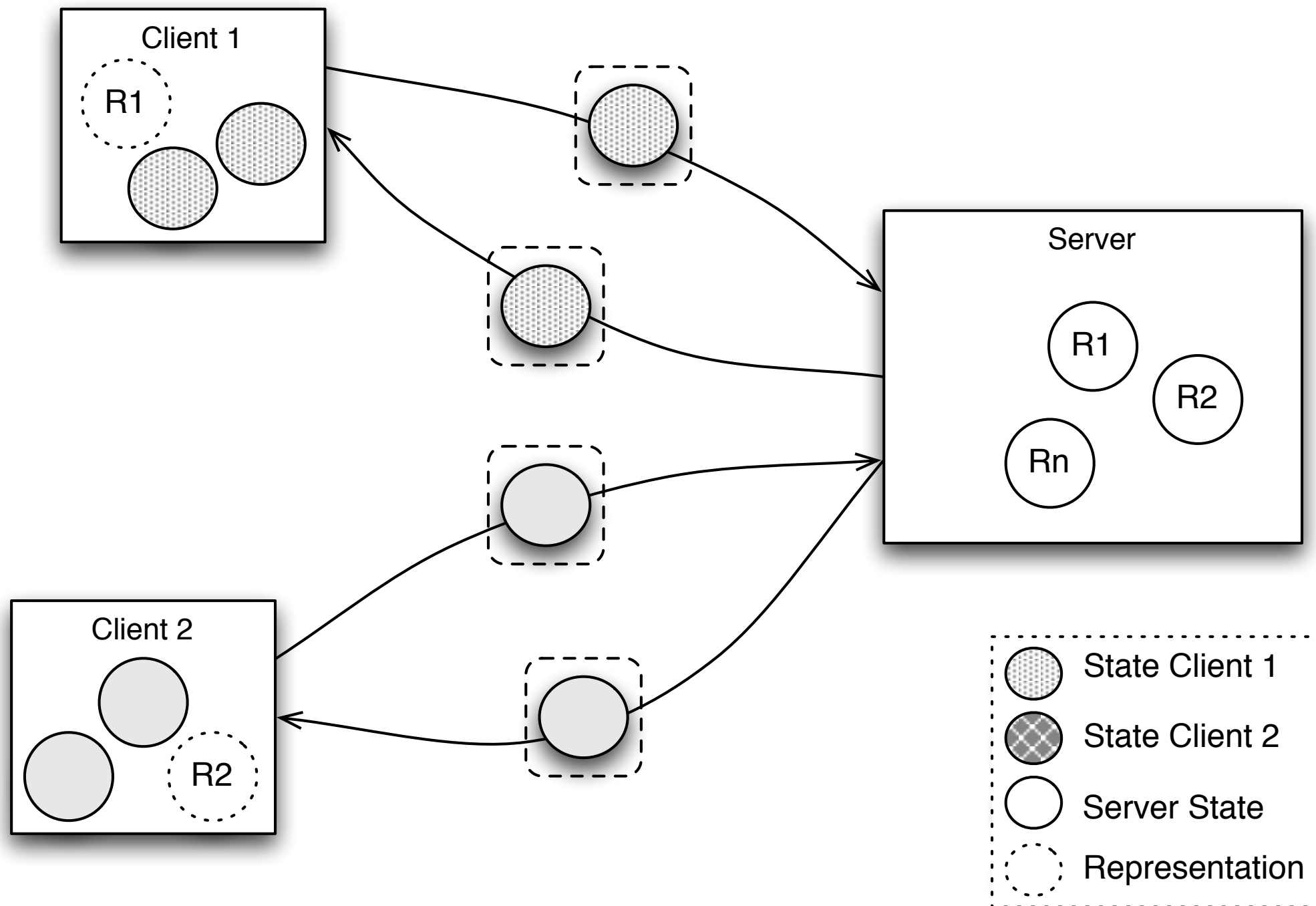
GWT

Backbone.js

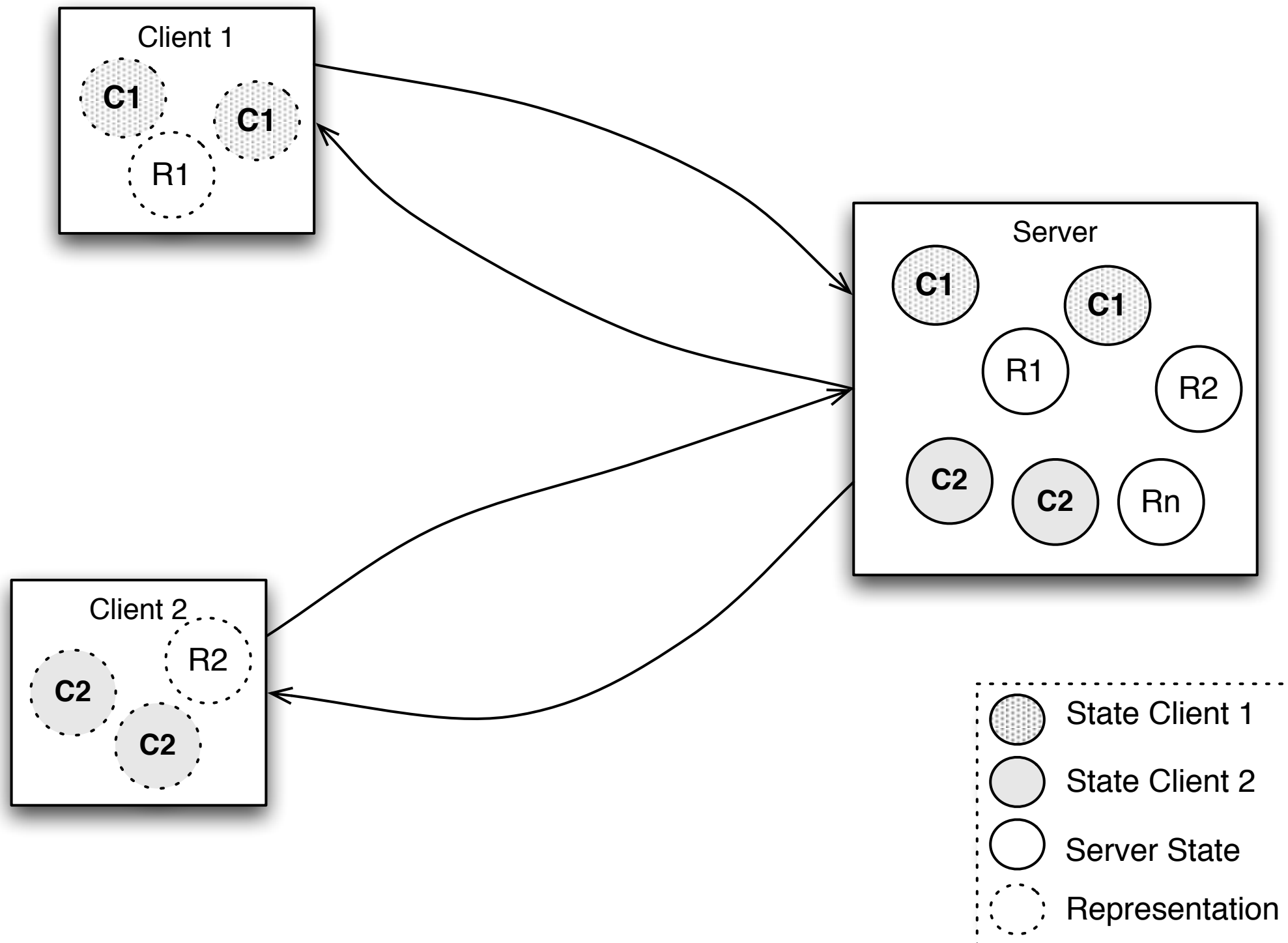
Meteor

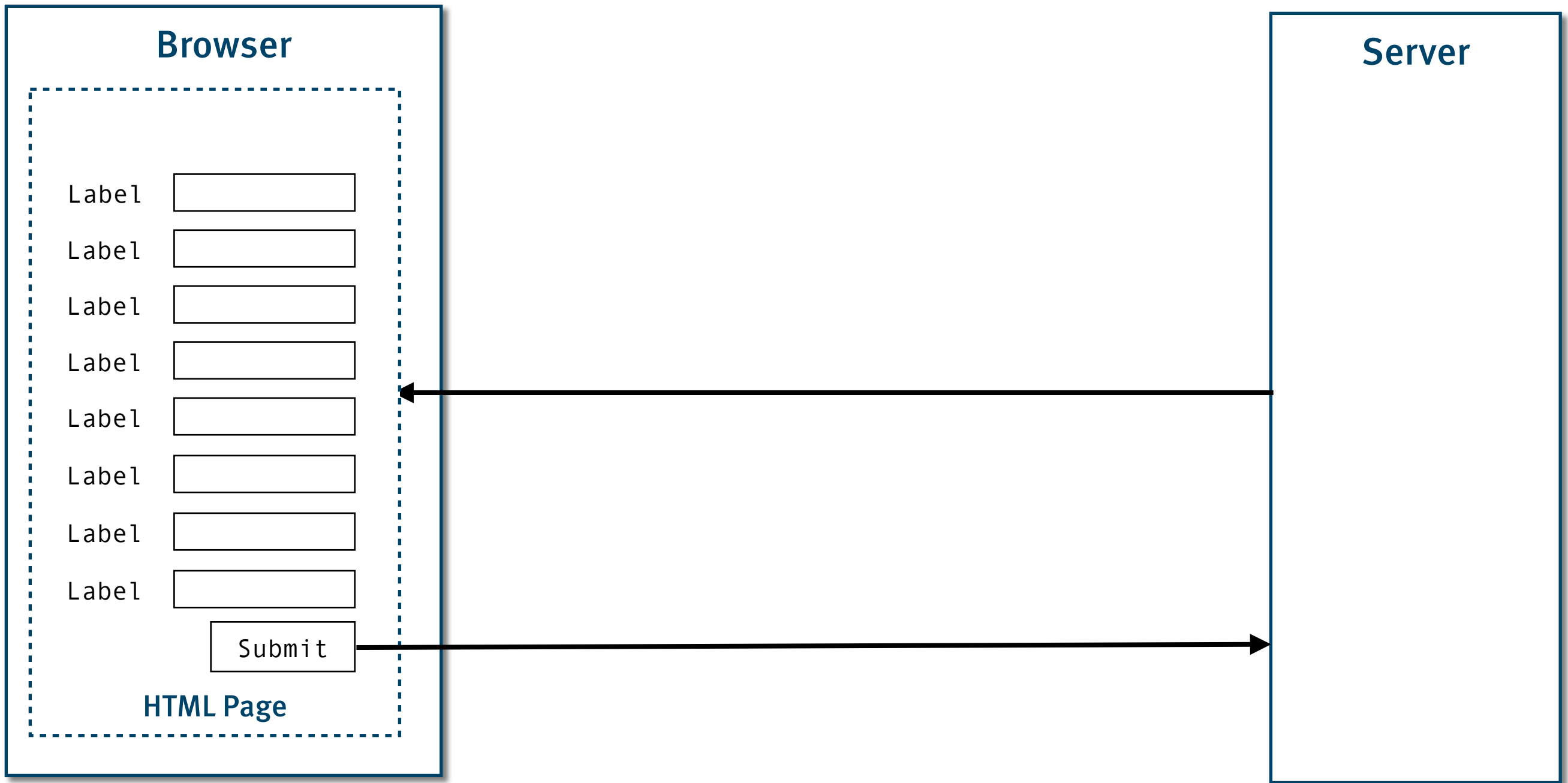


Turn session state ...



... into client or resource state





Browser

Label

Label

Label

Next

Label

Label

Label

Next

Label

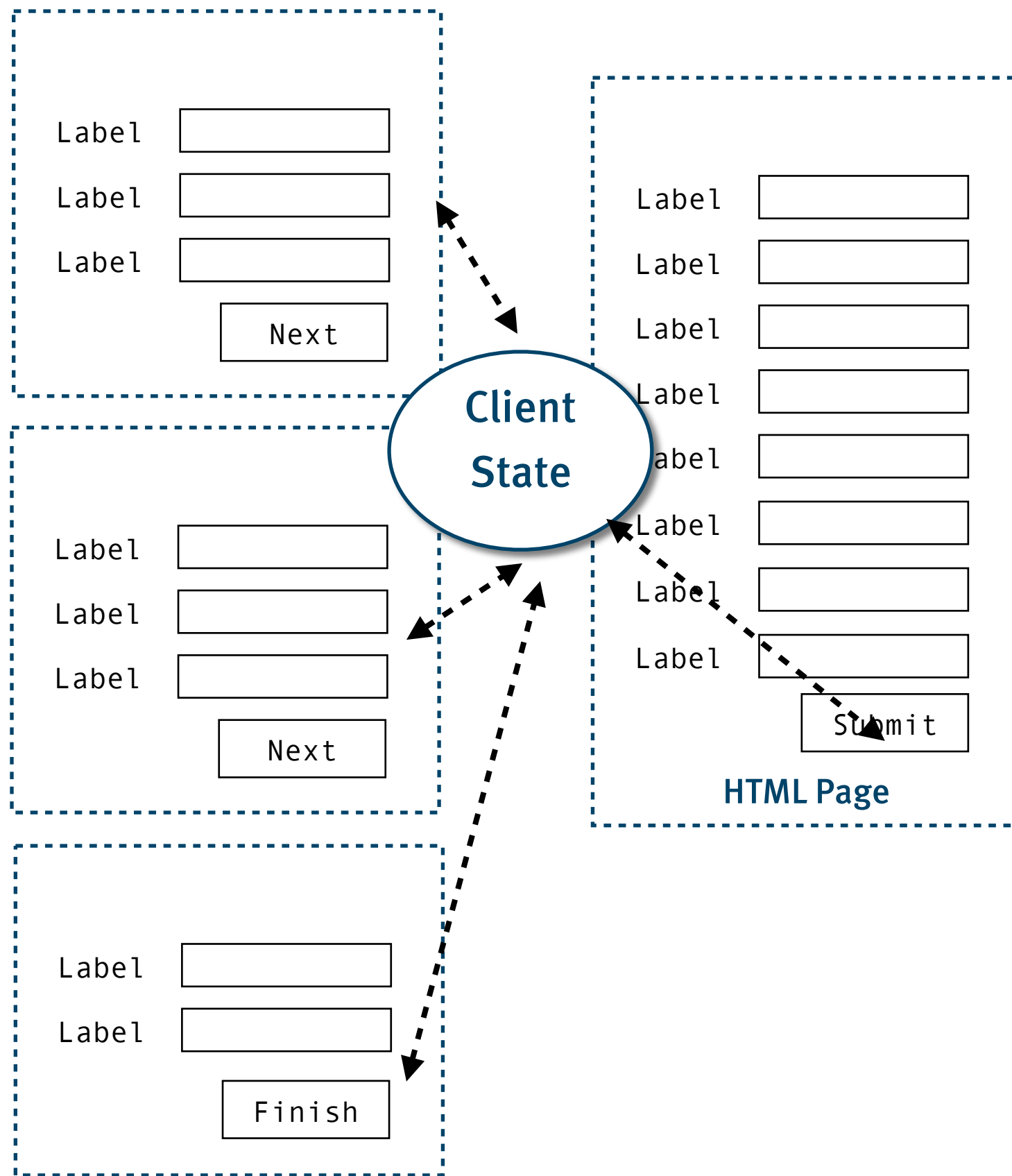
Label

Finish

Server

Session
State

Browser

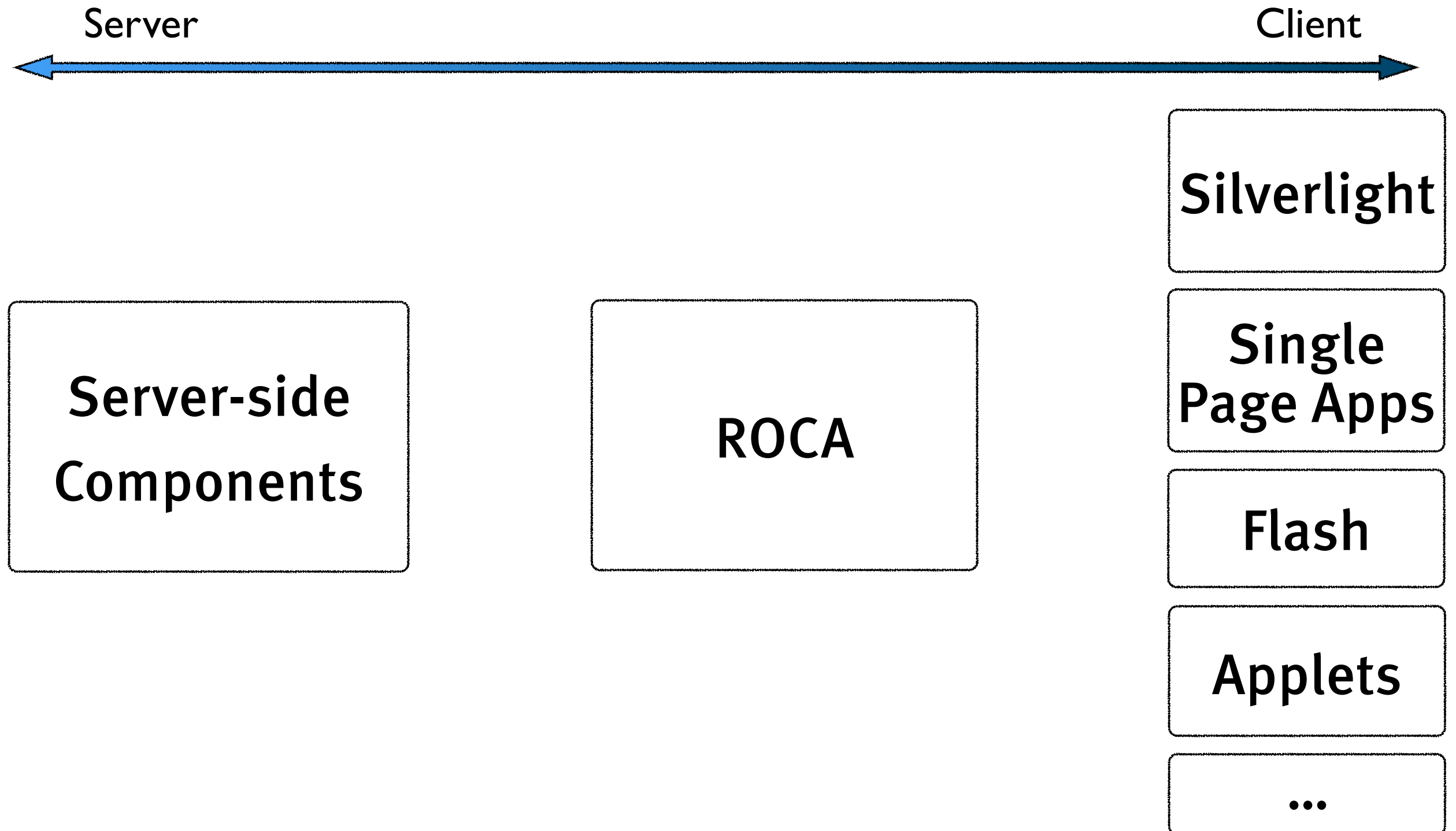


Server

ROCA

<http://roca-style.org/>

From server to client





MUST

Photo under CC by mendhak

© 2012 innoQ Deutschland GmbH

Thursday, May 31, 12

- 1. REST principles**
- 2. Server-side application logic**
- 3. No redundant logic**
- 4. Non-browser access**
- 5. Server sends structured HTML**
- 6. Progressive enhancement/
unobtrusive JavaScript**
- 7. No dynamic JavaScript**



SUGGESTED

Photo under CC by Sarah Reid

- 1. Additional representations (JSON, XML, ...)**
- 2. Authentication via HTTP mechanisms, cookies if unavoidable**
- 3. Use of HTML 5 history API**

VIOLETIONS



@stilkov

innoQ

- 1. Broken accessibility**
- 2. Session state**
- 3. Broken back/forward/refresh**
- 4. No bookmarks/links**
- 5. JS “engines” in client**
- 6. Server-side dependency on JS-generated markup**
- 7. Hashbangs et al.**

Summary

Don't be afraid of Web technologies

Embrace the Web, Don't fight it

**Standardize on the outside,
not on the inside**

**If you can't link to it,
does it exist?**

**If you can't build real Web apps
with it, is it worth your time?**

Vielen Dank!

Backup

Necessary Rules & Guidelines

Cross-system

Responsibilities

UI integration

Communication protocols

Data formats

Redundant data

BI interfaces

Logging, Monitoring

System-internal

Programming languages

Development tools

Frameworks

Process/Workflow control

Persistence

Design patterns

Coding guidelines

(Deployment, Operations)

Domain
Architecture

1.0

1.1

Cross-system
Rules

1.0

1.1

1.2

System-internal
Rules

1.0

1.1

2.0

2.1

t

Thank you!

Q&A

Stefan Tilkov

stefan.tilkov@innoq.com

[http://www.innoq.com/blog/st/
@stilkov](http://www.innoq.com/blog/st/@stilkov)

Phone: +49 170 471 2625



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany

Phone: +49 2173 3366-0
<http://www.innoq.com>

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland

Phone: +41 41 743 0116
info@innoq.com

@stilkov

