

Do You Do REST?

Subbu Allamaraju
Yahoo! Inc

QCon San Francisco Nov 3-5, 2010

Solutions for Improving Scalability and Simplicity



O'REILLY®

| YAHOO! PRESS

Subbu Allamaraju

Take-away

Interfaces for interoperability

Do you do REST? 

 ... we don't do pure REST

But where did REST come from?

Quality Attributes

- Performance
- Scalability
- Simplicity
 - Separation of concerns
- Modifiability
 - Evolvability, extensibility, customizability, configurability, reusability
- Portability
 - By making a constrained set of choices for representing data and code
- Reliability
 - Avoiding single points of failure, redundancy, monitoring

Constraints

- Identification of resources
- Manipulation via representations
- Self-descriptive messages
- Hypermedia as the engine of application state
- Code on demand

Application

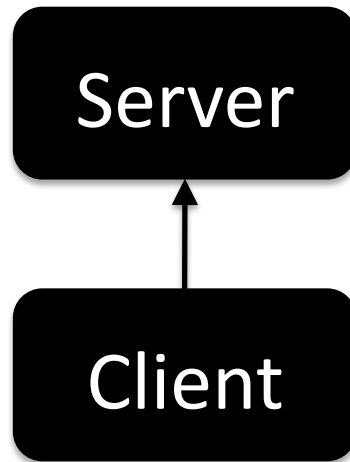
to the web

- URIs as resource identifiers
- HTTP as the uniform interface
- MIME-like message format
- Hypertext (controls like forms and links)
- Code on demand



Scalable and pervasive Network-effect in action





Application and impact
with mixed results

Level 3: Hypermedia

Martin Fowler's
"Glory of REST"

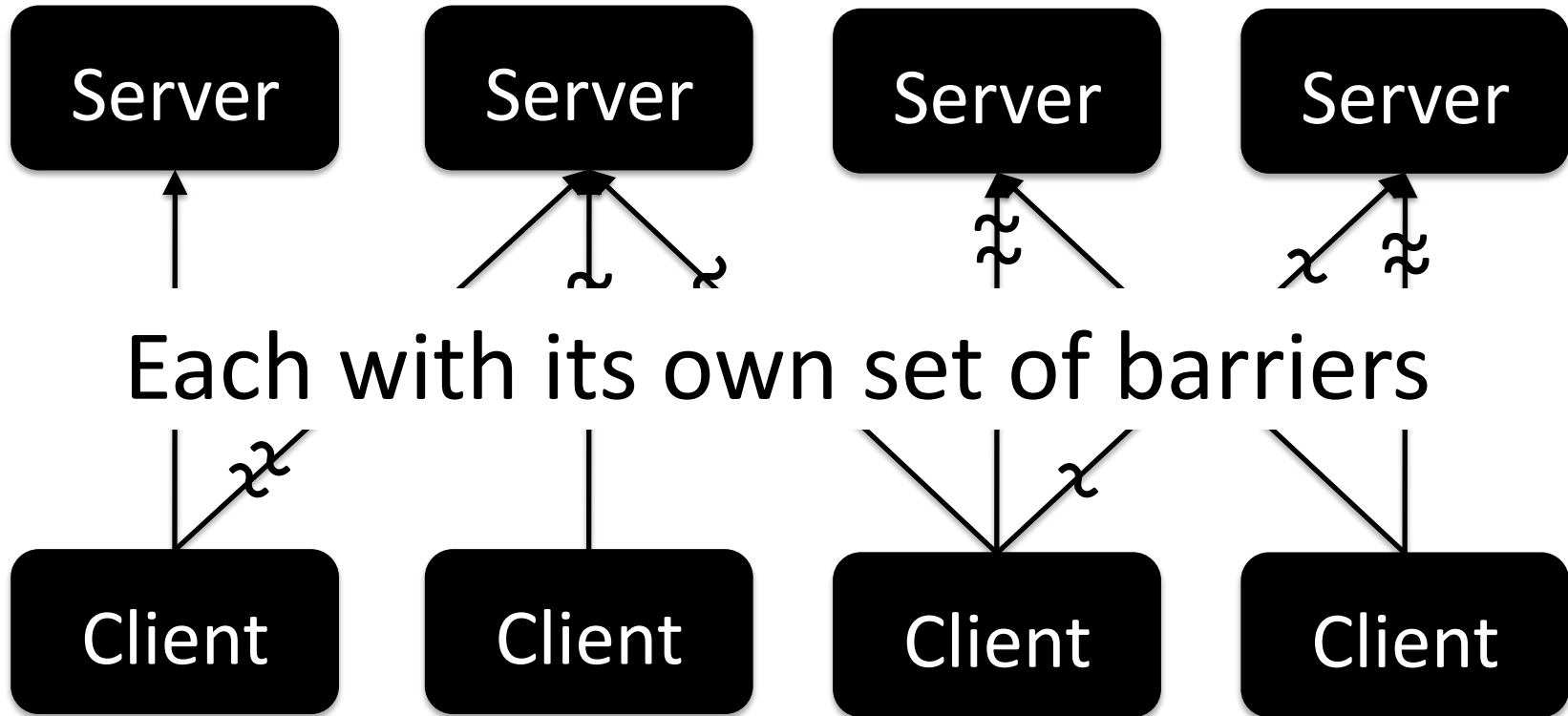
Level 2: Many URIs, many HTTP methods

Level 1: Many URIs, one HTTP method

Level 0: One URI, one HTTP method

[Leonard Richardson's Maturity Model](#)

- Q. What is the contract?
- Q. What needs to be governed?
- Q. How to deal with versioning?
- Q. How to make services extensible?



Each with its own set of barriers

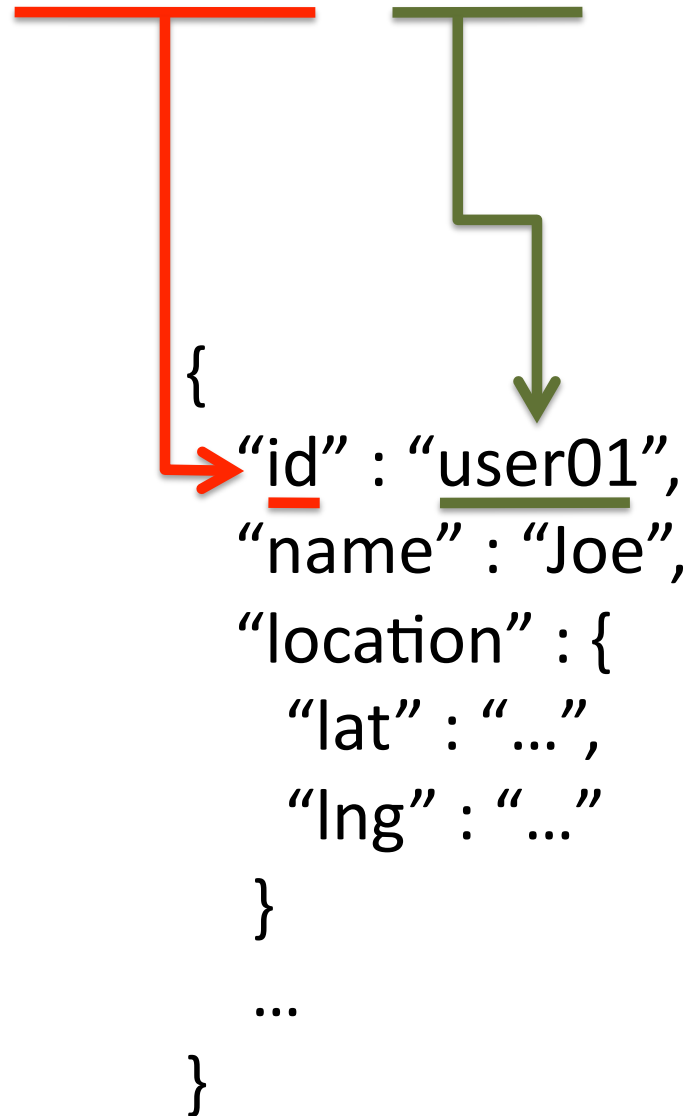
Inconsistent URIs, IDs, HTTP usage, security, formats, ...

`http://foo.com/person/1234/address/6789`

`http://bar.com/user?id=1234&filter=address`

`http://baz.com/user;id=1234/address;id=6789`

```
<comment id="1234" posted_by="user01">  
  <text>...</text>  
  ...  
</comment>
```



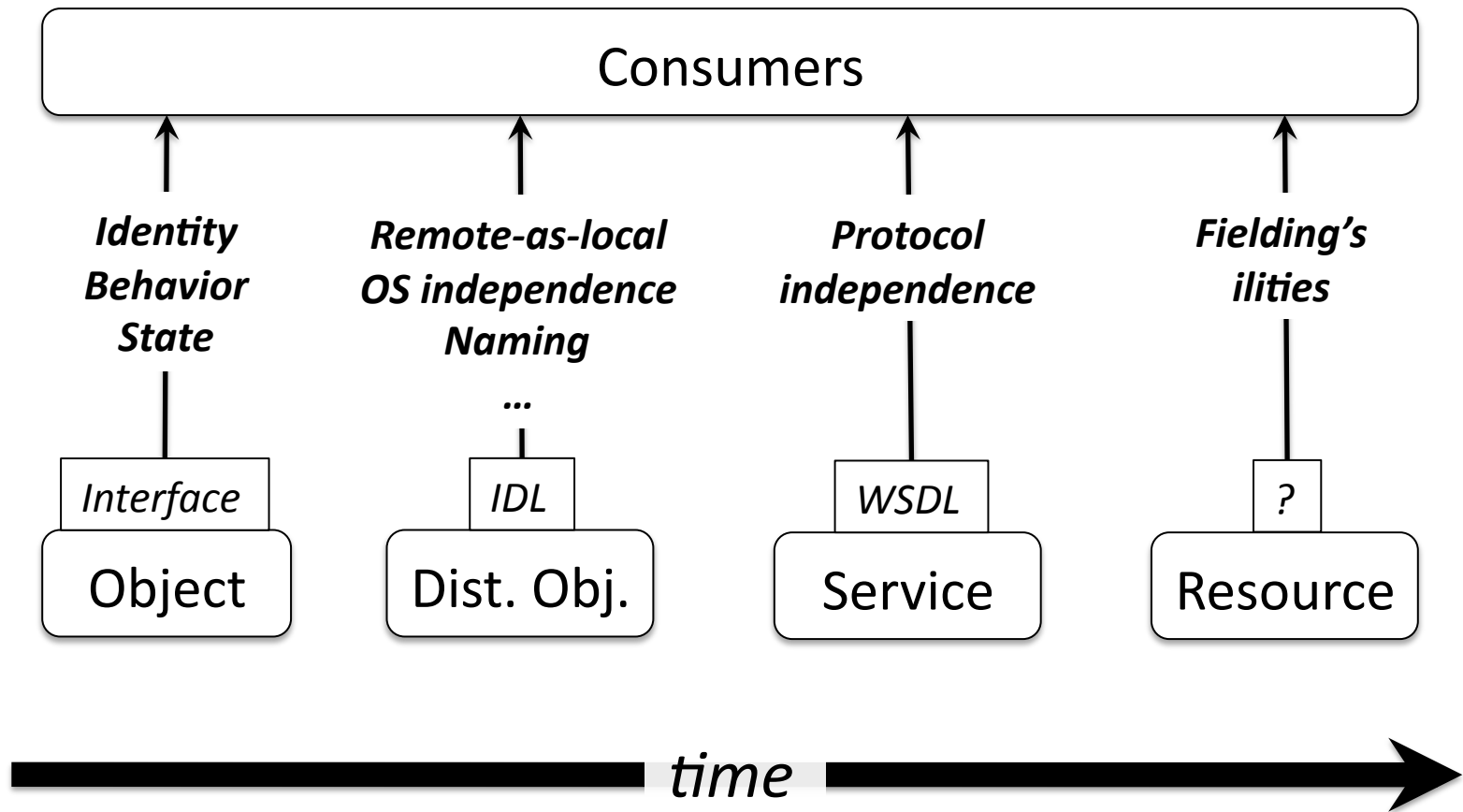


Challenge: Interoperability

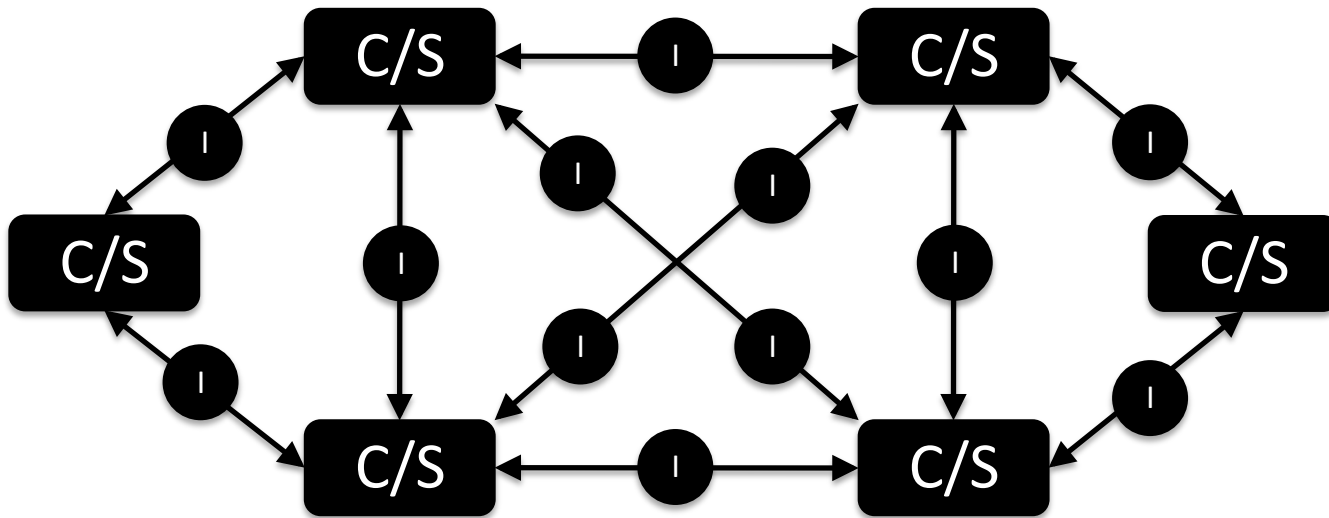
“...the ability of diverse systems and organizations to work together” [[Wikipedia](#)]

What is the interface of the Web?

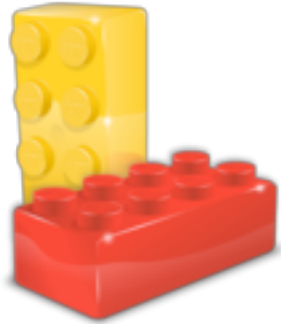
Thinking about interfaces



Decentralized



Agreements among disagreements



~~Single interface~~
Many interfaces

Most given + Some application defined

Interoperability = $f(\text{interfaces})$

Interface: URIs

Interface: URIs

[\[RFC 3986\]](#)

1. Syntax

`{scheme}://{authority}/{path}?{query}#{frag}`

2. Resolution

Base: `http://www.example.org`

Relative: `/path?query`

3. Normalization and Comparison

Is `http://www.example.org/path?query` the same as `http://example.org/path/?query`

+ Dereferenceable

http://www.subbu.org/blog/tag/rest

http://api.nytimes.com/svc/search/v1/
article?query=haiti&api-key={api-key}

jar:http://mycorp.com/jars/foo.jar!
mycorp/User.class

jms:topic:Stocks.Prices

+ Identity

- Identifier in some local or global scope
- Be able to make assertions

```
<! A review of RESTful Web Services
    Cookbook -->
<review>
  <link href="http://amzn.com/0596801688"
        rel="review"/>
  ...
</review>
```


+ Hackable and Consistent

`http://api.twitter.com/version/statuses/
show/:id.format`

`http://api.twitter.com/1/statuses/
user_timeline.format?user_id=:userid`

`https://my.social.net/contacts/userEmail`

`http://my.social.net/videos/username/
playlists`

Interface: Resources

Interface: Resources

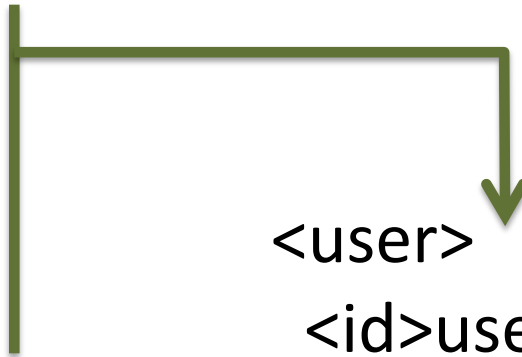
None

“(T)he term ‘resource’ is used in a general sense for whatever might be identified by a URI” [\[RFC-3986\]](#)

+ Many Resources

- Everything that a client needs has a URI
- Consistent formats and patterns for representations

```
<comment>
  <id>1234</id>
  <user>
    <id>user01</id>
    <link rel="user"
      href="..." />
  </user>
  <text>...</text>
  ...
</comment>
```



```
<user>
  <id>user01</id>
  <name>Joe</name>
  <location>
    <lat>...</lat>
    <lng>...</lng>
  </location>
</user>
```

Interface: HTTP

Interface: HTTP

[\[RFC 2616\]](#)

1. HTTP methods GET, PUT, POST, DELETE, HEAD and OPTIONS

Fixed set of methods lets a lot of people agree and support them – a key for interoperability

Interface: HTTP

[\[RFC 2616\]](#)

2. Message syntax
 - Headers + body
3. Method safety
 - GET, HEAD, OPTIONS
4. Method idempotency
 - GET, HEAD, OPTIONS, PUT, DELETE

See Chapter 1 in RESTful Web Services Cookbook

+ Protocol

1. Content negotiation (?)
 - Accept-* headers or URIs for each format
2. Caching
 - Last-Modified, Cache-Control
3. Conditional requests
 - If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match
4. Custom headers (?)

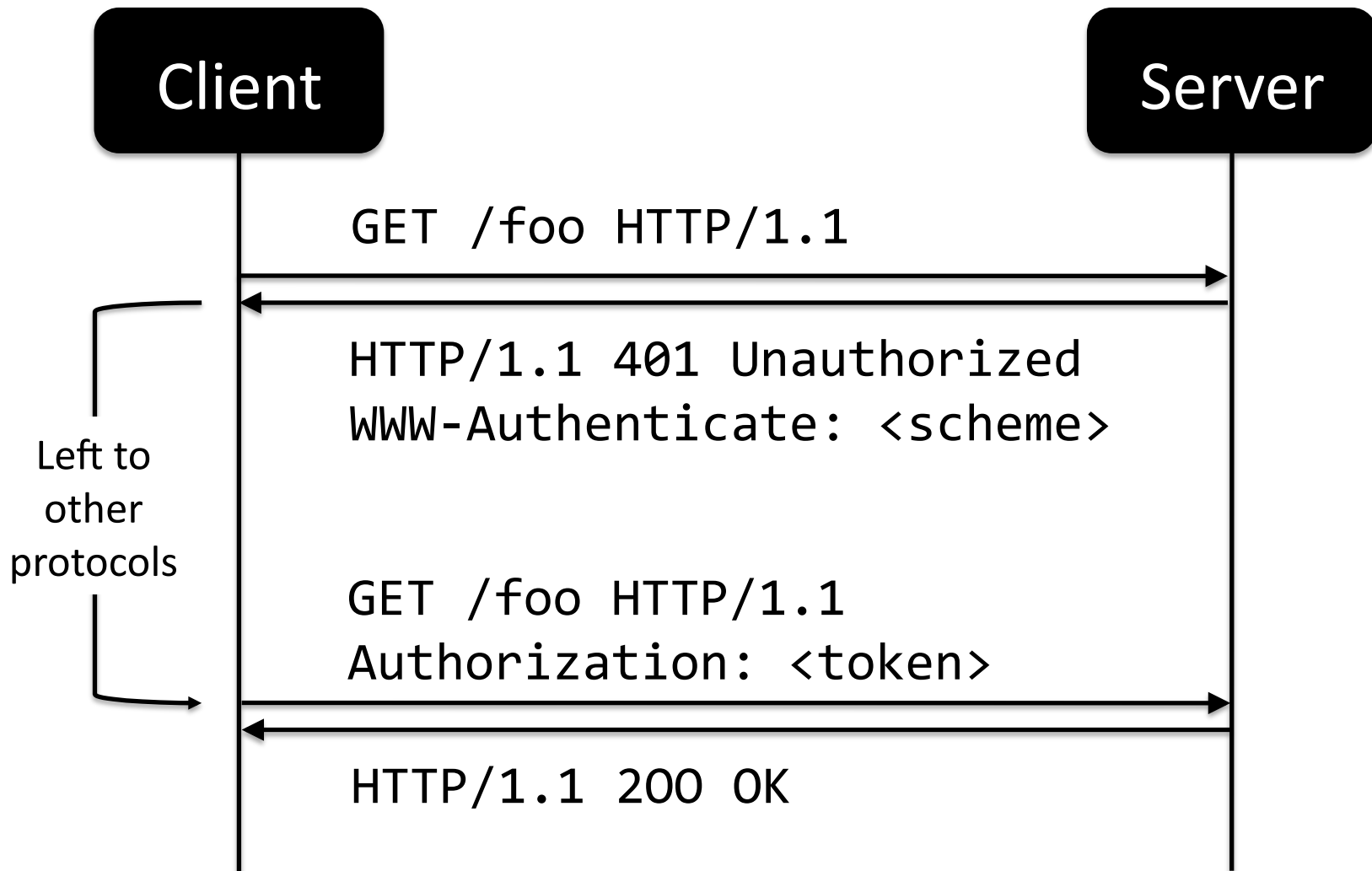
See Chapters 7, 9 and 10 in RESTful Web Services Cookbook

Interface: Security Protocols

Interface: Security

[\[RFC 2616\]](#)

- Support for WWW-Authenticate and Authorization headers



Interoperability Hurdles

- Early protocols
 - RFC 2617
 - Poor adoption
 - Basic auth was weak and Digest auth was complex
- Cookies are widely interoperable, but poorly specified
- Security policies within corporate networks are often unique to each organization

+ Consistency

1. Consistent HTTP based protocols
 - Basic
 - OAuth 1.0 and/or OAuth 2.0
 - Custom but *consistent*

Interface: Hypertext

Interface: Hypertext

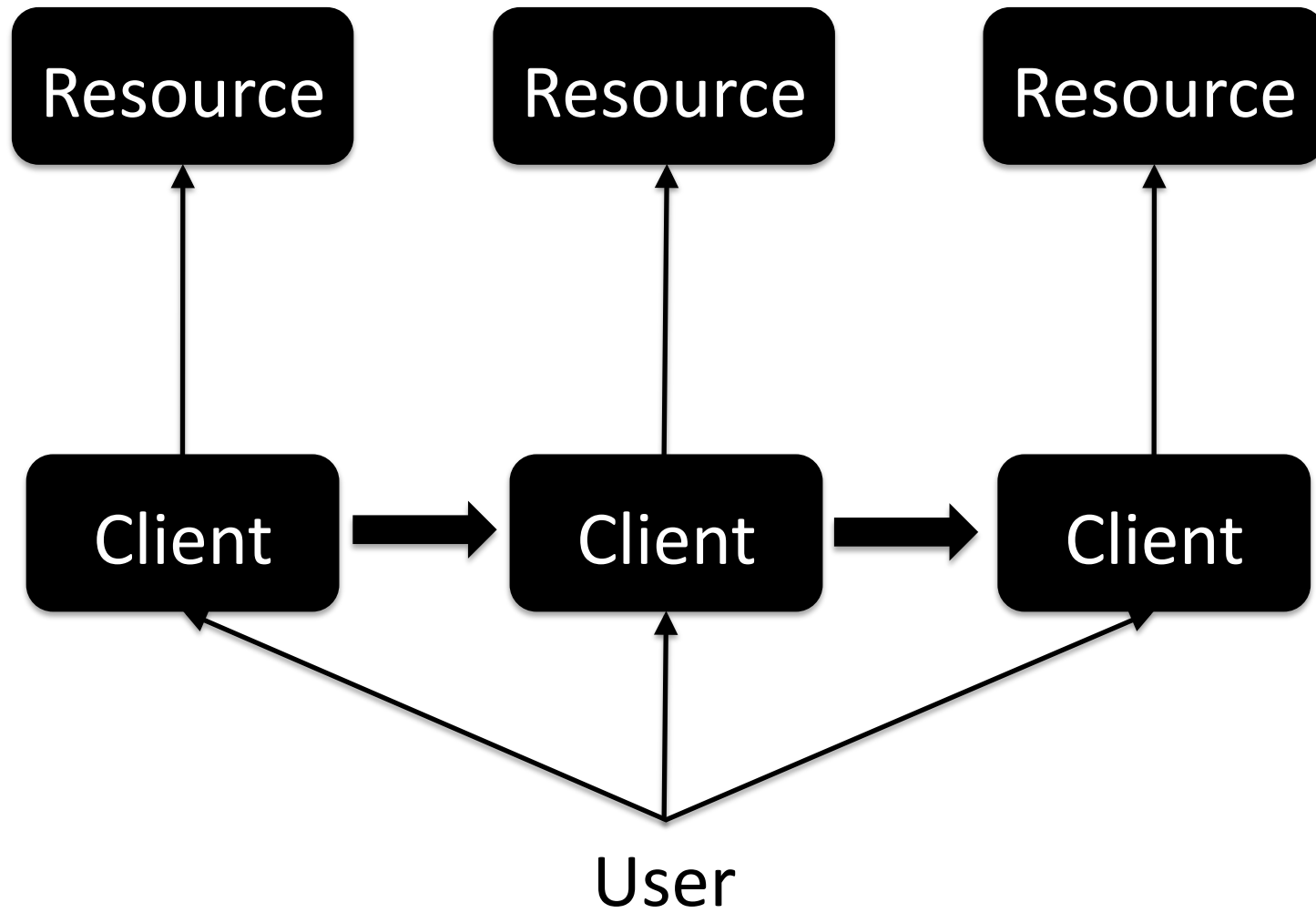
“Apparently you can book tickets, check email and do all sorts of stuff online without knowing that each website has a distinct address called a URL.”

[Compared to you, most people seem dumb](#)



Human-Web

- HTML as the main interface
- Well-defined hypermedia controls (forms and hyperlinks)
- Automatic composition in the browser (CSS, images, video etc.)
- A user leisurely driving the client



```
GET /order/1234 HTTP/1.1
Host: orders.example.org
```

```
HTTP/1.1 200 OK
Content-Type: application/vnd.order+xml
```

```
<order>
  <id>1234</id>
  <status>pending</status>
  <link rel="http://www.example.org/rel/order.submit"
        href="http://sales.example.org/order/1234?
              t=asd2342"/>
  ...
</order>
```

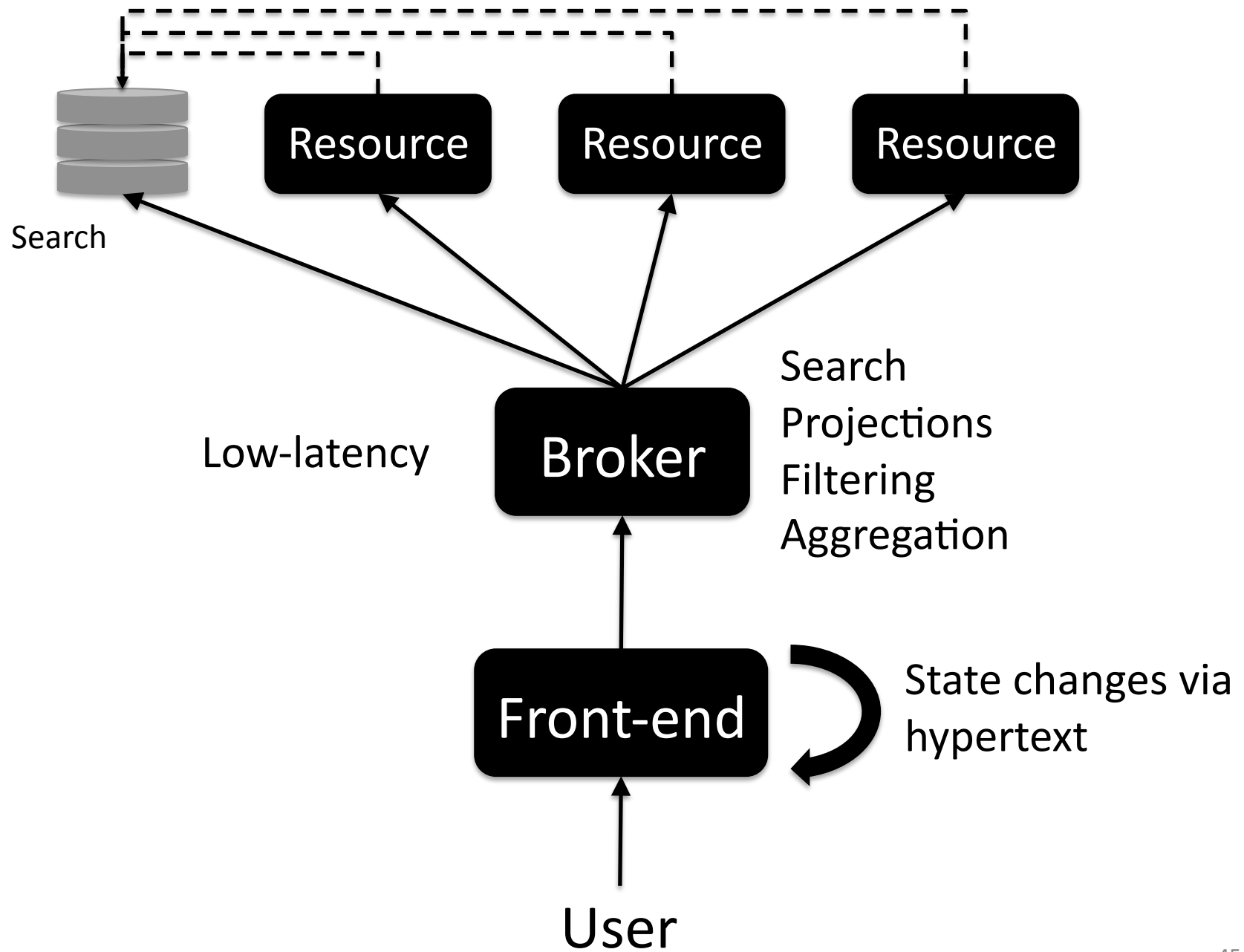
application/vnd.order+xml

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis suscipit, augue ut mattis venenatis, purus elit dapibus nibh, nec tempus nunc lectus viverra nulla ...



<http://www.example.org/re1/order.submit>

Suspendisse in dignissim erat. Donec ut arcu sed libero semper imperdiet ornare ac eros. Cras hendrerit volutpat nibh, at convallis purus porttitor in. Nullam scelerisque placerat dui, id facilisis diam laoreet non. ...



HEAD /user/1234 HTTP/1.1

Host: www.example.org

HTTP/1.1 204 No Content

Link: <<http://www.example.org/comments?uid=1234>>;
rel="related <http://www.example.org/comments>"

Link: <<http://www.example.org/profile/1234>>;
rel="related <http://www.example.org/profile>"

Link: <<http://www.example.org/videos?uid=1234>>;
rel="related <http://www.example.org/media>"

HEAD /user/1234 HTTP/1.1

Host: www.example.org

HTTP/1.1 200 OK

Content-Type: application/xml; charset=UTF-8

<user>

 <id>1234</id>

 <link href="http://www.example.org/comments?uid=1234"
 rel="related http://www.example.org/comments"/>

 <link href="http://www.example.org/profile/1234"
 rel="related http://www.example.org/profile"/>

 ...

</user>

Interface: Hypertext

1. HTML when the hypermedia driver is a user
 - UI converging to HTML
2. Special media types for local interoperability
3. Links expressed as Link headers [[RFC 5988](#)] or in the body or both
4. Link relation types

What are the interfaces of the
Machine-Web?

Identifiers			Watch-out
Patterns			Non-negotiable
Many			
Resources	Cust. Headers (?)		
	Conneg (?)		
URI Templates	Cond. Req.	Consistent	
Hackable	Caching	Basic	
Consistent	Idempotency	Cookies	Link Rel. Reg.
Dereferenceable	Safety	OAuth	Links in body
URI Syntax	Syntax	HTTP Based	Link Header
URI	HTTP	Auth	Hypertext

Interfaces of the Web

- The architecture of the web relies on many loosely related interfaces
- Each focusing on different aspect of interoperability
- Some pre-defined – others application-defined
- Interoperability is a function of supporting these interfaces

Subbu Allamaraju

<http://www.subbu.org>

[@sallamar](#)