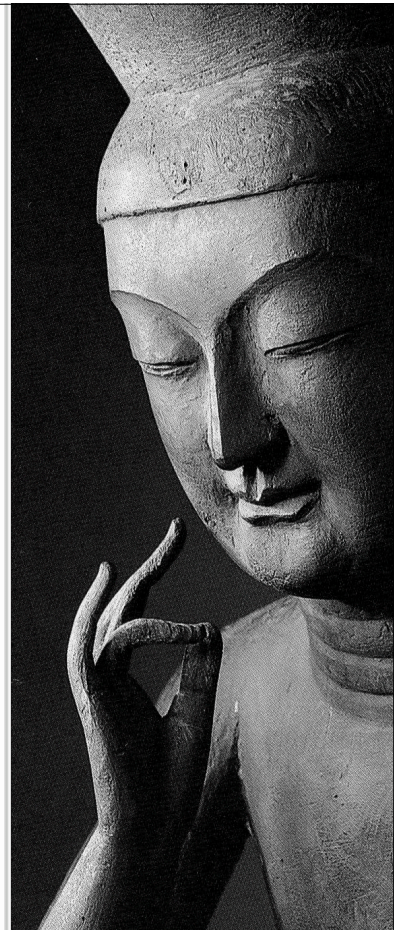


Semantic Architecture : Composing Resource-Oriented Systems

Brian Sletten (@bsletten)

Speaker Qualifications

- ⊗ 18 Years Experience
- ⊗ Specialize in benefits and applicability of next-generation technologies
- ⊗ Author “Resource-Oriented Architectures : Building Webs of Data” Addison-Wesley 2011
- ⊗ Contributor to “Beautiful Architecture”
- ⊗ One of SemanticWeb.com’s “Top 100” Semantic Web People



	Introduction REST Semantic Web NetKernel Semantic Composition	
--	---	--

	BASIC Assembly Pascal C C++ Perl Java Python Groovy Ruby Objective-C Scala Erlang Clojure	
--	---	--



Atari 800

Atari ST

DOS

Windows

SunOS

AIX

HPUX

FreeBSD

Linux

OS X



Spaghetti

Structured

OO

Functional

Logical

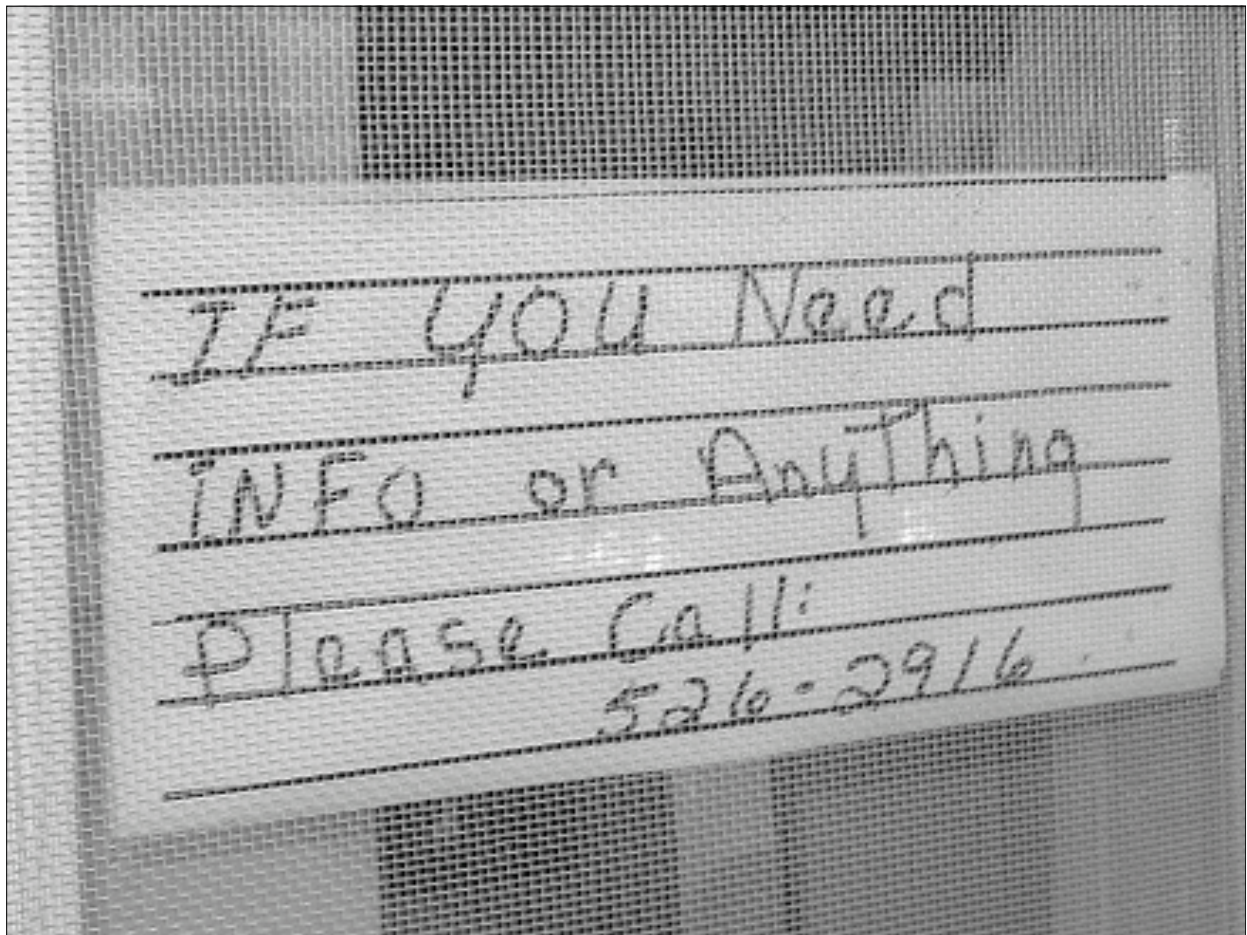
Distributed

Web Services

REST

Resource-Oriented

Semantic Web



(Spaghetti Code)

```
1010101010101000
10010101111010101
1010101010100101
010101001010100100
1111010011001010101
010101010101001010
101001010101110001
010101010001010101
010101010101010110
```


(Modular Code)

```
101010101010100010  
0101011111010101101  
010101
```

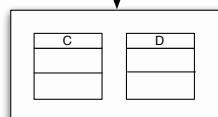
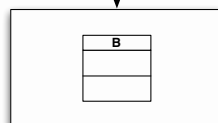
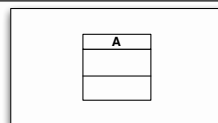


```
10101001010101010010  
10100100111101001100
```

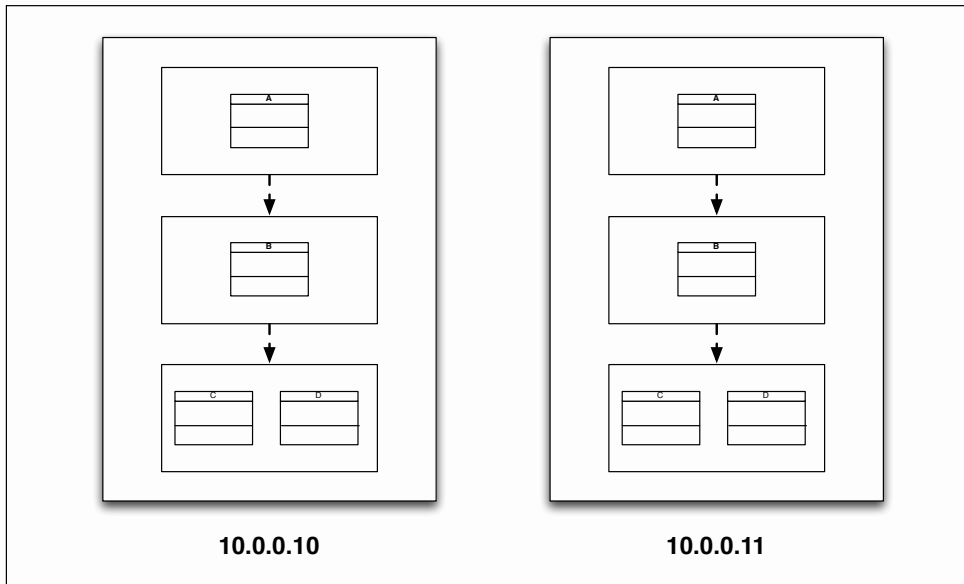


```
010101010101001010  
1010010101011100010  
10101010001
```

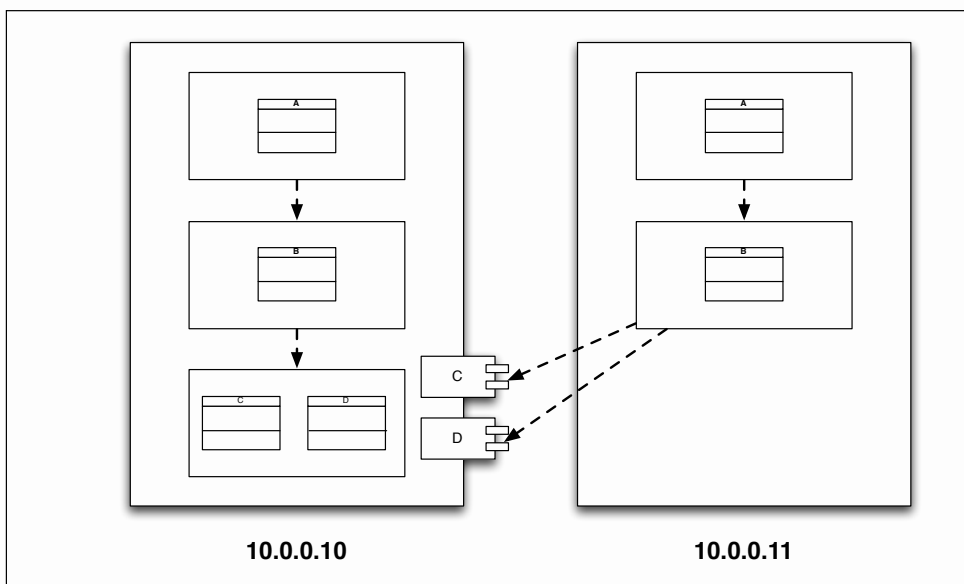
(Libraries and Frameworks)



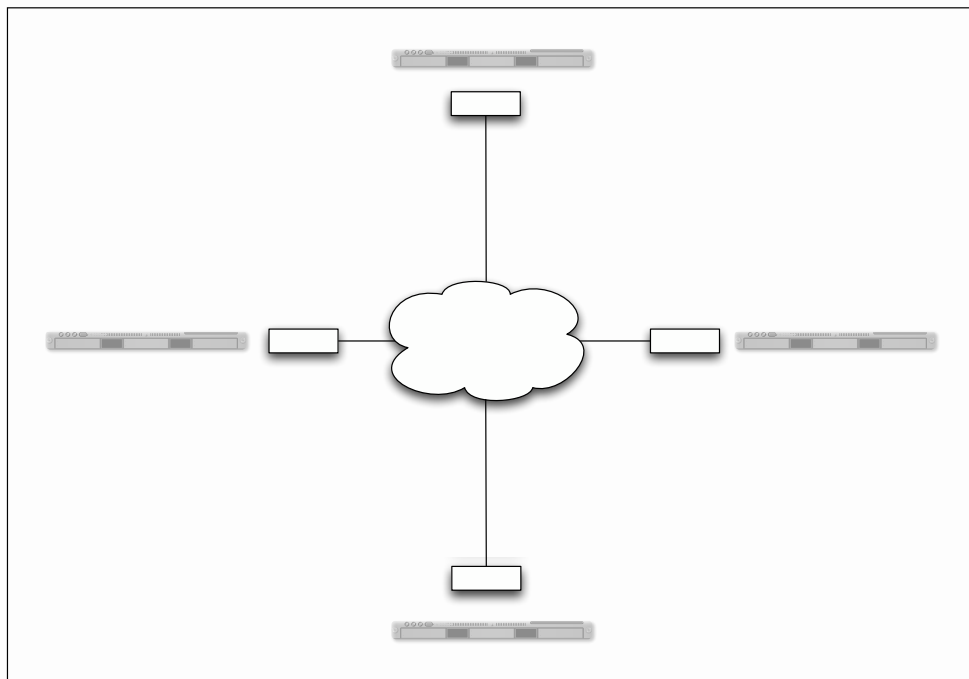
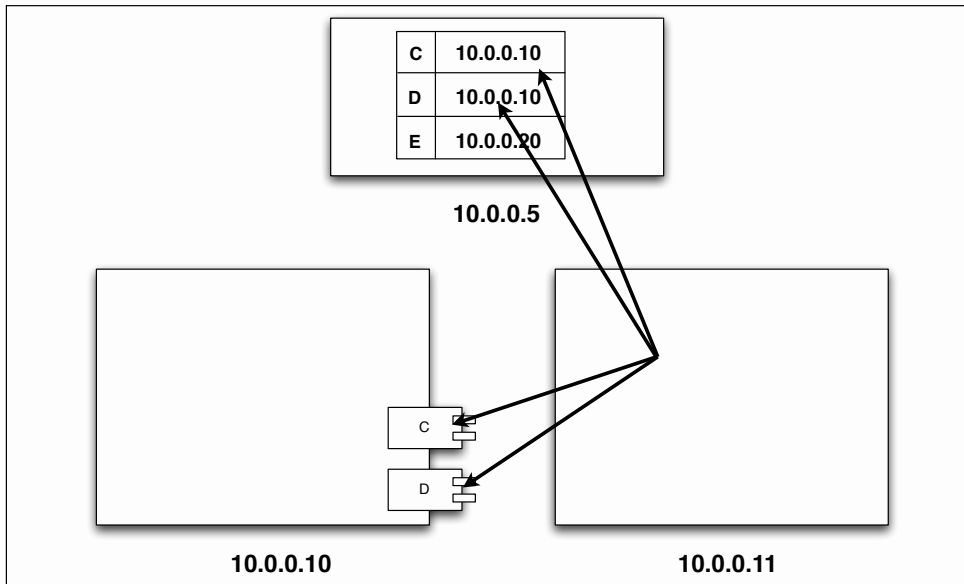
Replicated Code

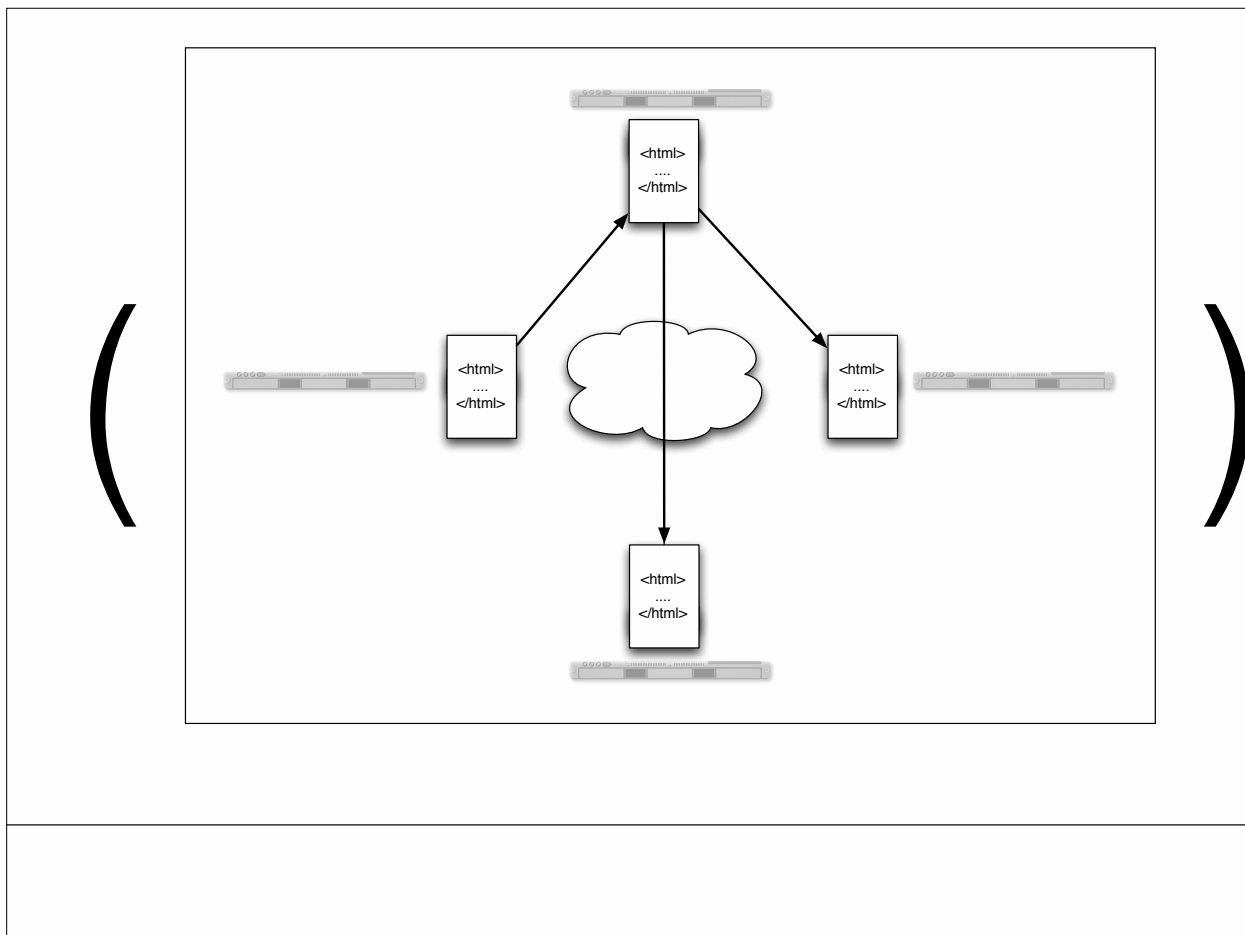


Remote Procedure Calls



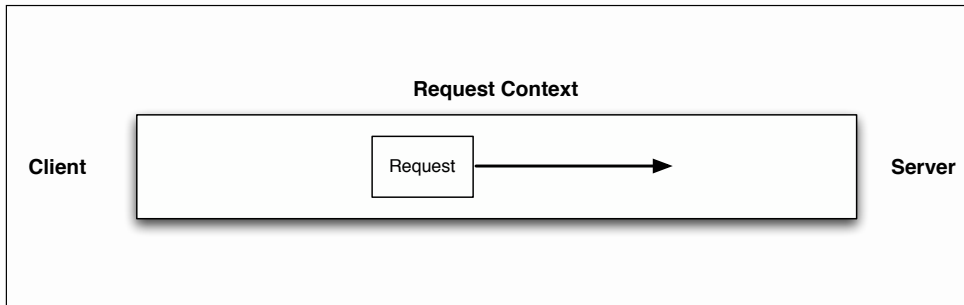
Distributed Computing



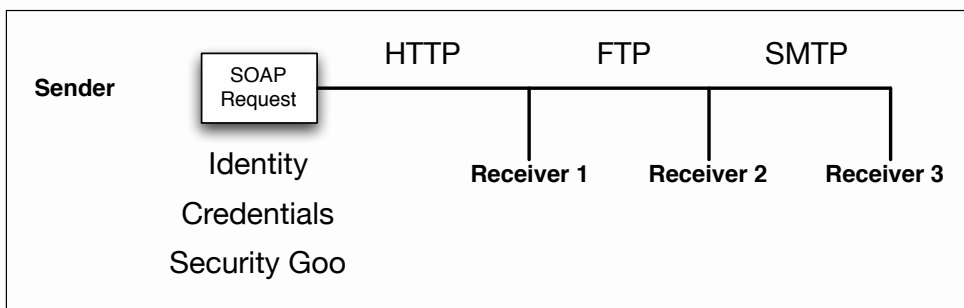


	Introduction
	REST
	Semantic Web
	NetKernel
	Semantic Composition

Requests in a Context



Decontextualized Requests



(What REST Isn't)

- ⊗ RPC through URLs
- ⊗ A direct replacement for SOAP
- ⊗ XML over HTTP
- ⊗ Insecure
- ⊗ A toy
- ⊗ Whatever you want it to be

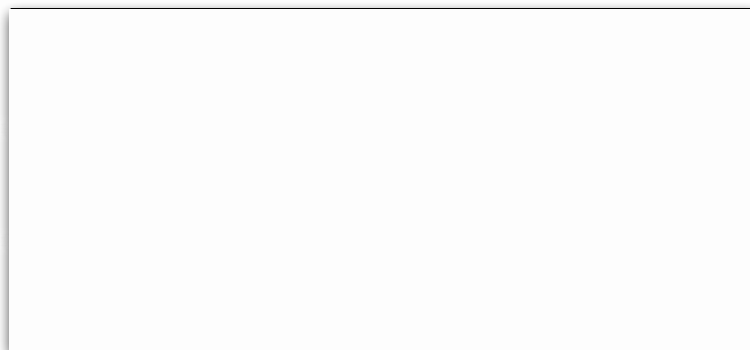
(Resources)

- ⊗ A gene sequence
- ⊗ The weather in a location
- ⊗ Your bank balance
- ⊗ You
- ⊗ Your employer
- ⊗ A customer
- ⊗ A customer's order
- ⊗ All orders over \$1,000 in last 24 hours

(Representations)

- ⊗ HTML
- ⊗ XML
- ⊗ JSON
- ⊗ RDF
- ⊗ Excel spreadsheet
- ⊗ Image

(RMM - Level 0)



(RMM - Level 1)

URL

(RMM - Level 2)

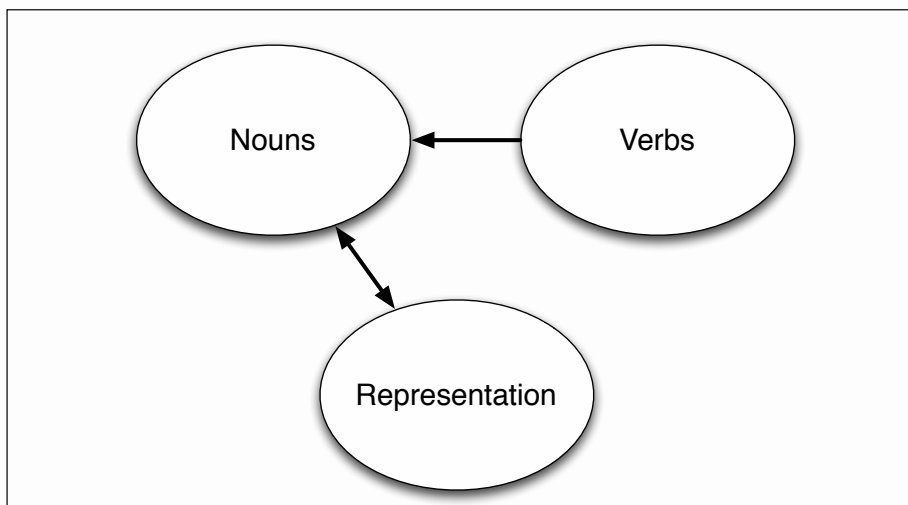
HTTP

URL

(RMM - Level 3)

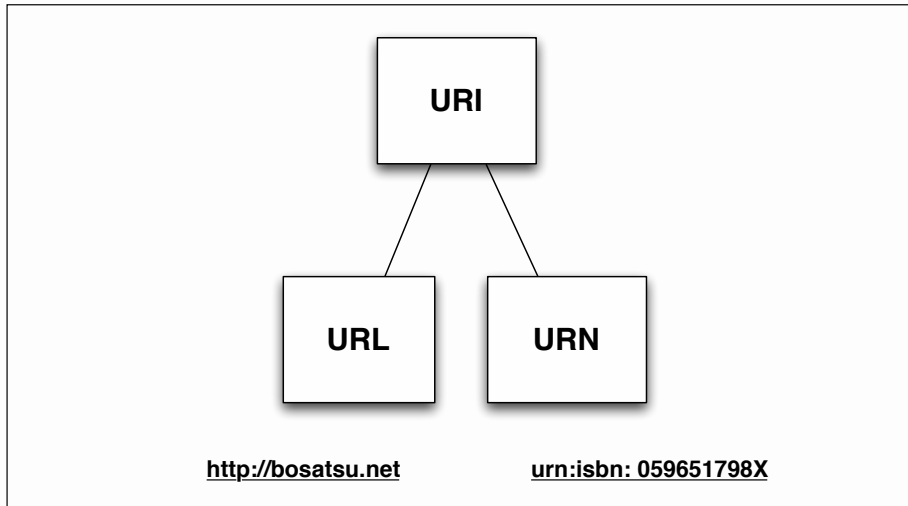
Hyper Media
HTTP
URL

(REST)



<http://rest.blueoxen.net/cgi-bin/wiki.pl?RestTriangle>

(URIs, URLs and URNs)



(Uses For URLs)

Documents	http://myserver.com/docs/reports/2009/qtr/3
Data	http://myserver.com/gene/ade2
Services	http://myserver.com/order?type=open
Concepts	http://purl.org/net/bsletten

~~http://server/get~~employees?status=hourly

http://server/employee

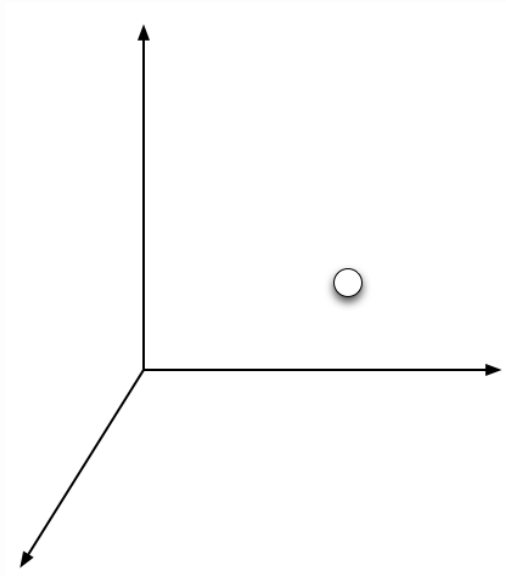
http://server/employee?status=hourly

http://server/employee/status/hourly

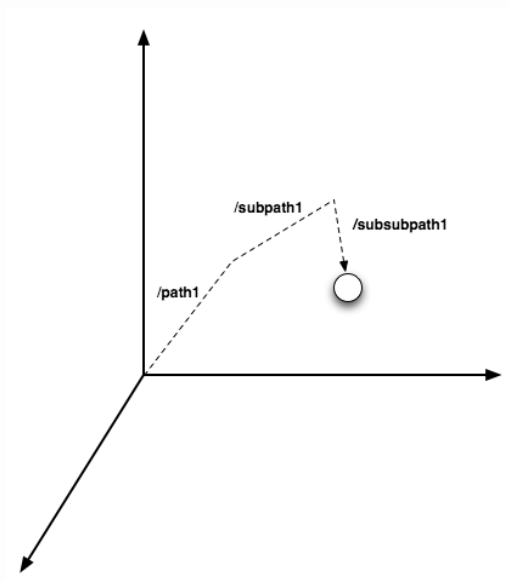
http://server/employee/bsletten

http://server/map/color;lat=50;long=20;scale=32000

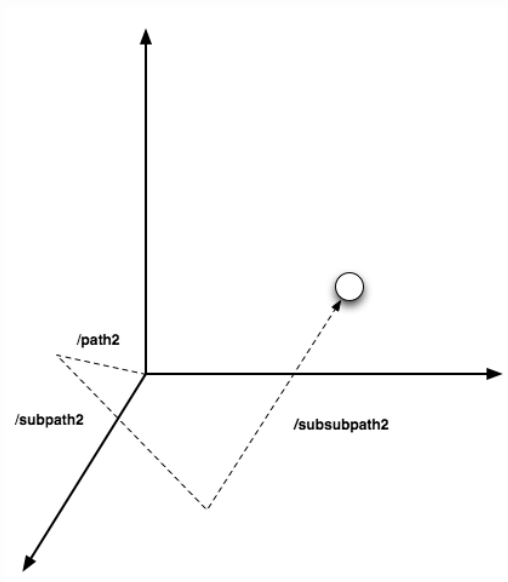
Information Spaces



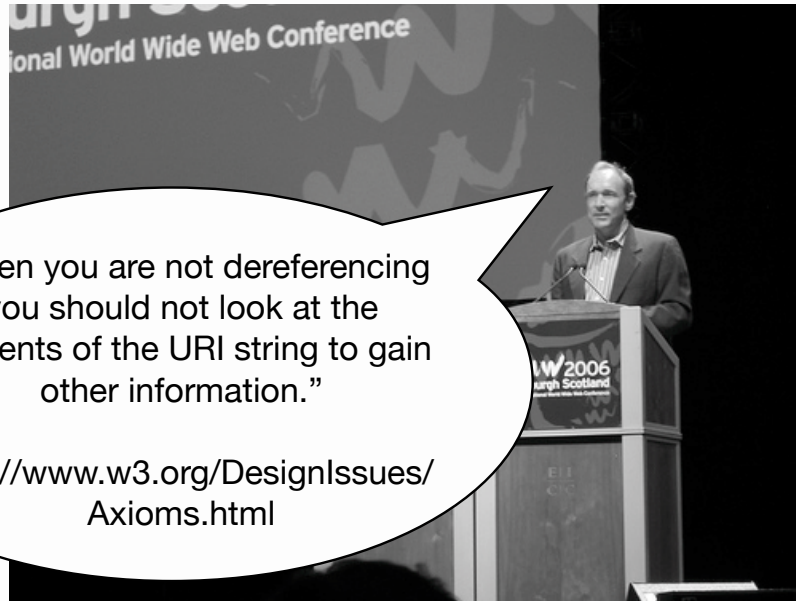
(“Vector 1”)



(“Vector 2”)



URI Opacity



"When you are not dereferencing
you should not look at the
contents of the URI string to gain
other information."

[http://www.w3.org/DesignIssues/
Axioms.html](http://www.w3.org/DesignIssues/Axioms.html)

<http://www.flickr.com/photos/ajstarks/152558066/>

URI Opacity

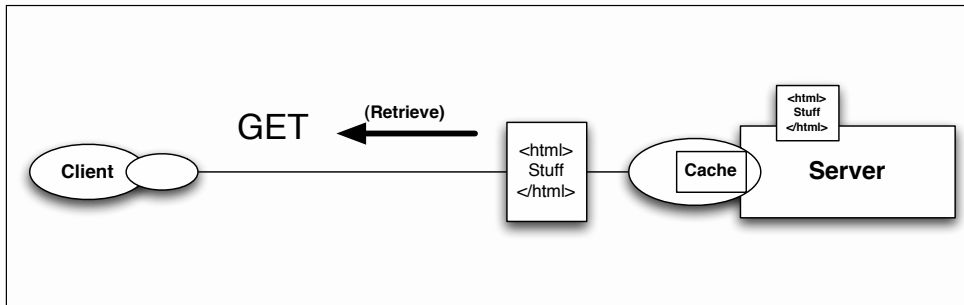


"REST does not require that a URI be opaque.
The only place where the word opaque occurs in
my dissertation is where I complain about the
opaqueness of cookies. In fact, RESTful
applications are, at all times, encouraged to use
human-meaningful, hierarchical identifiers in
order to maximize the serendipitous use of the
information beyond what is anticipated by the
original application."

[http://tech.groups.yahoo.com/group/rest-
discuss/message/3232](http://tech.groups.yahoo.com/group/rest-discuss/message/3232)

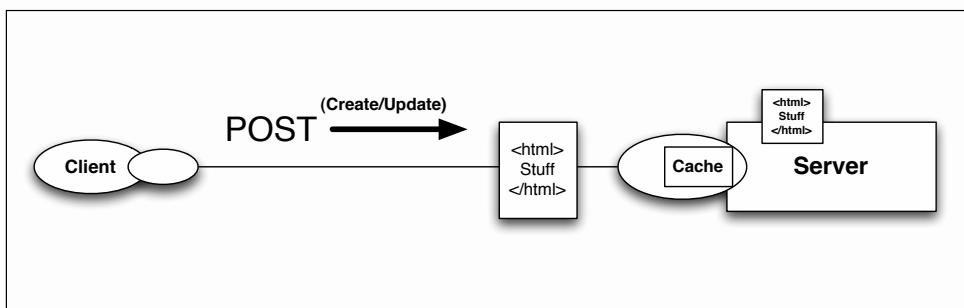
(GET

)

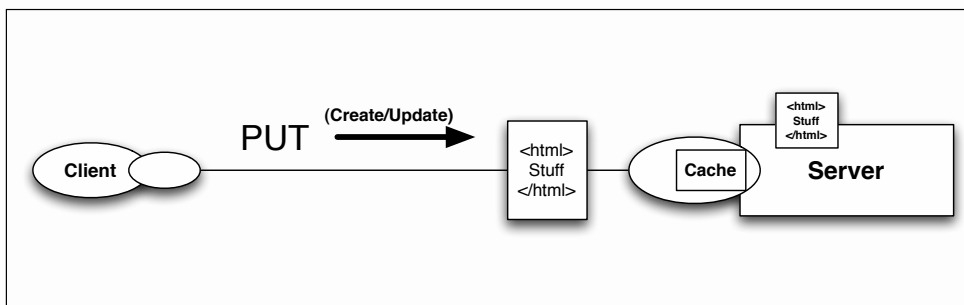


(POST

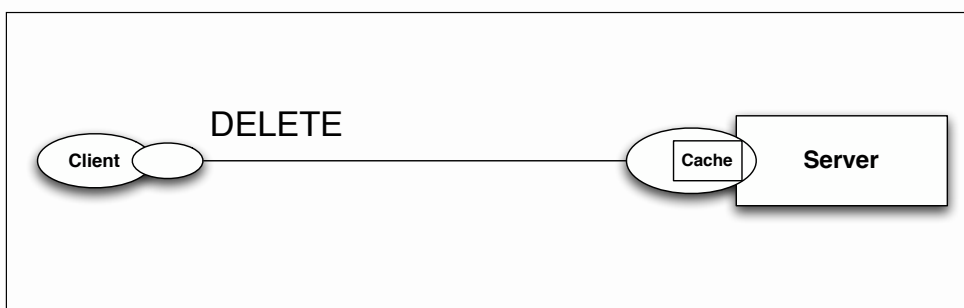
)



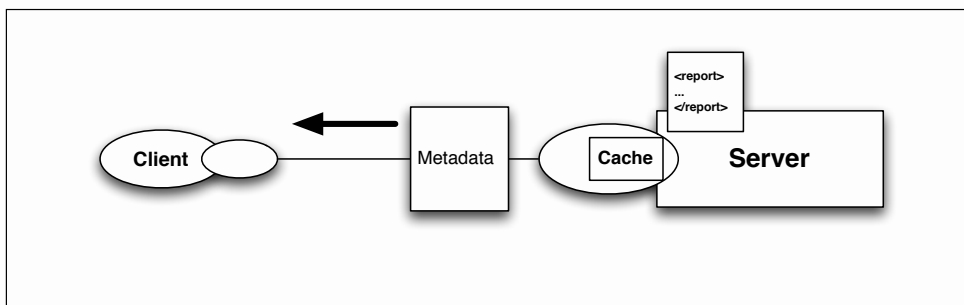
(PUT



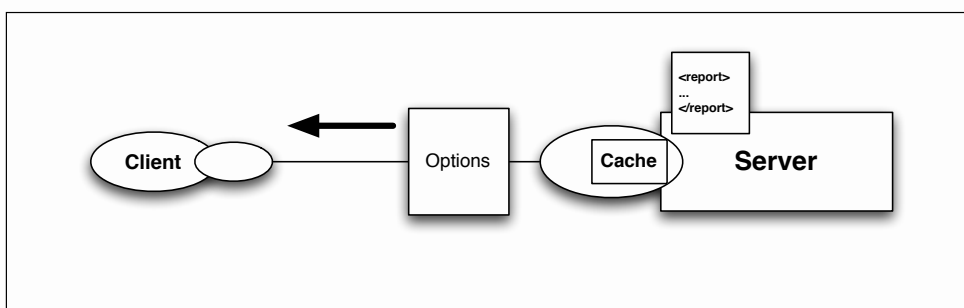
(DELETE



(HEAD



(OPTIONS



(Idempotency

GET	✓
PUT	✓
POST	X
DELETE	✓

(Response Codes

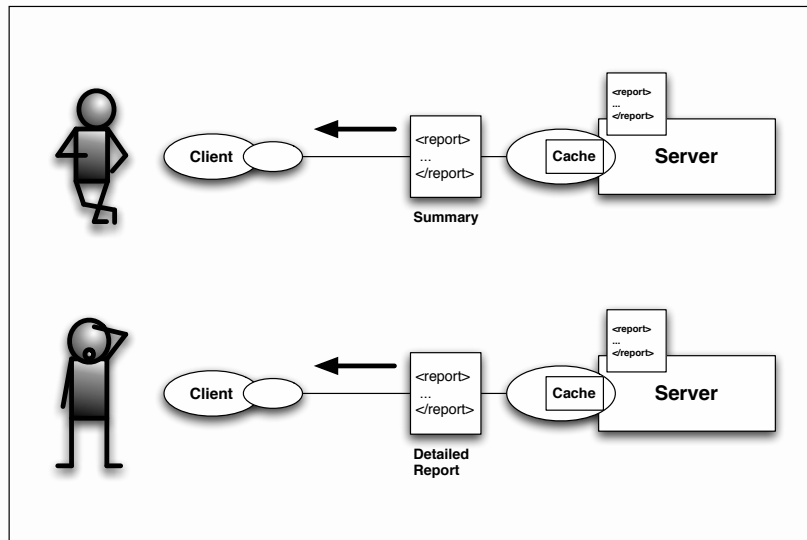
200	Ok
201	Created
202	Accepted
204	Success

301	Moved Permanently
302	Found
303	See Other
304	Not Modified

401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
409	Conflict
411	Length Required
413	Entity Too Long
415	Unsupported Media Type

500	Internal Error
503	Service Unavailable

Information-Based Access Control



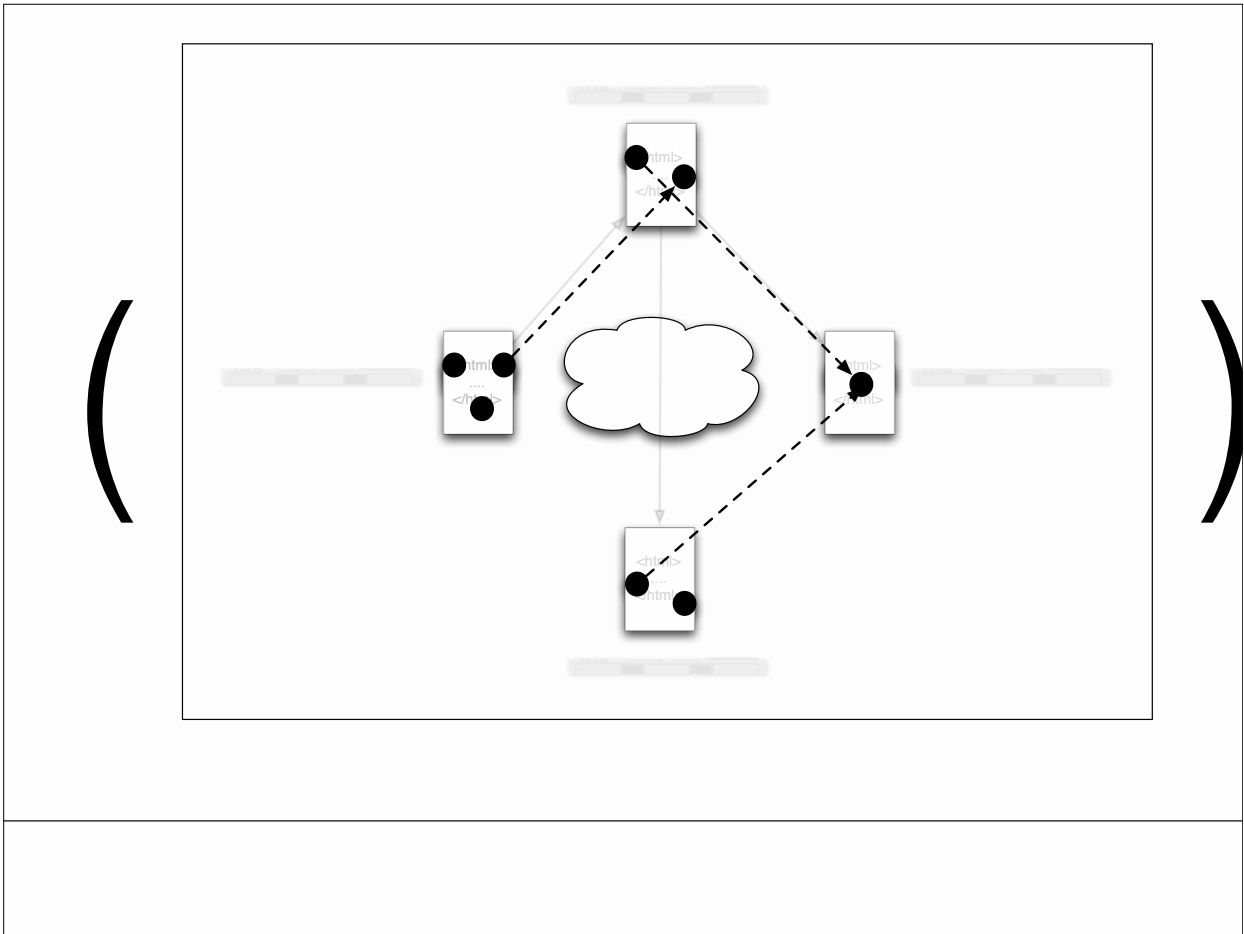
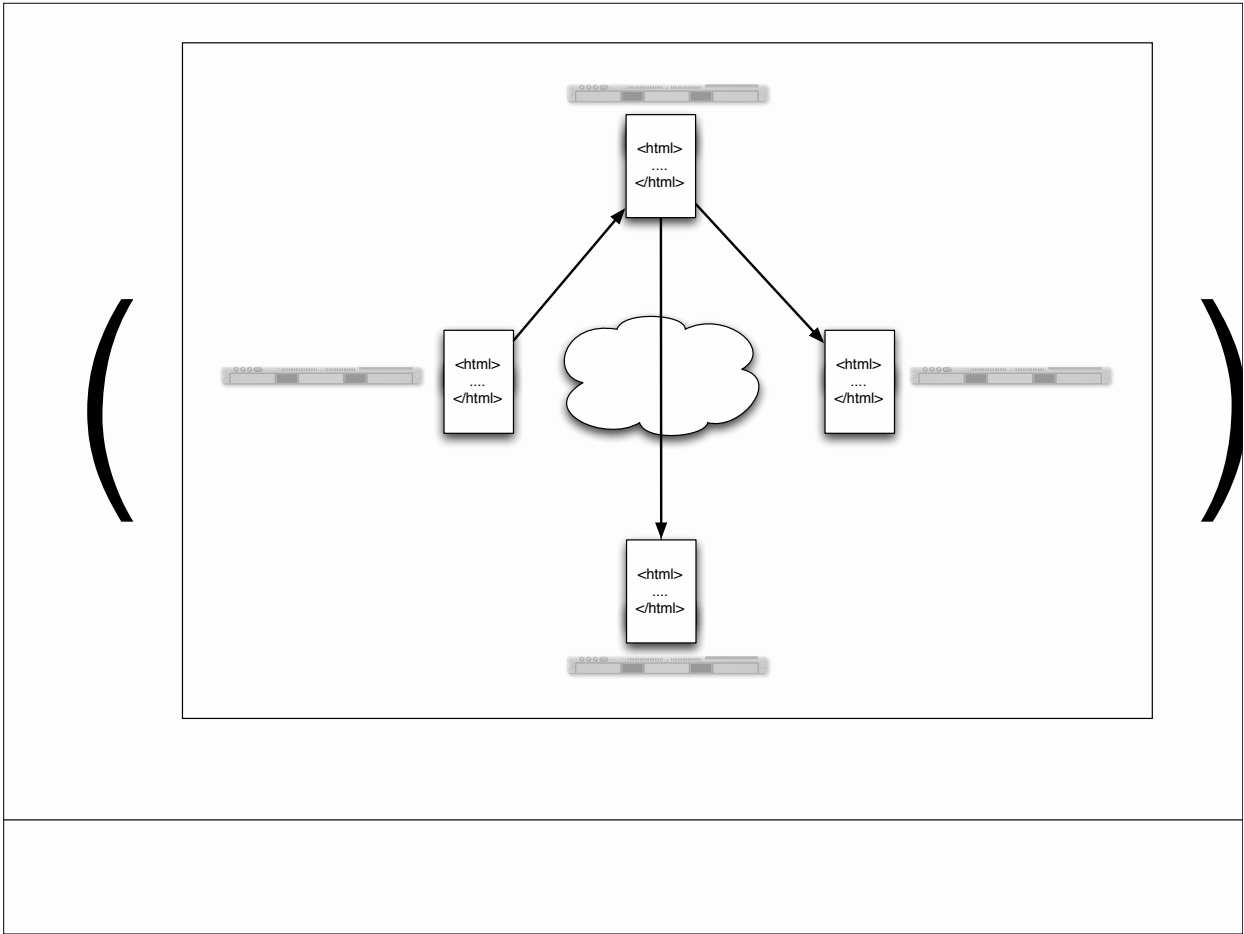
Introduction

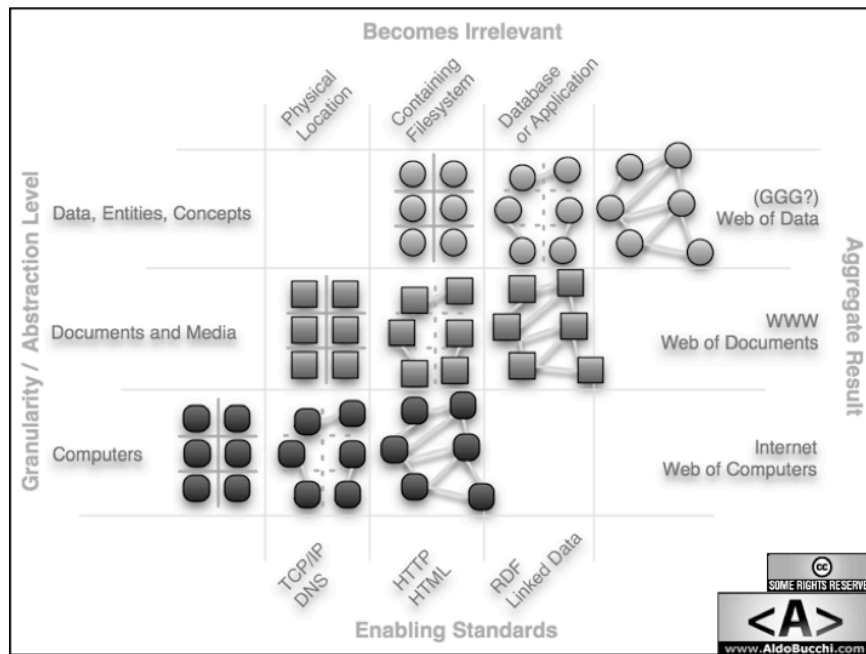
REST

Semantic Web

NetKernel

Semantic Composition





OWL/SKOS

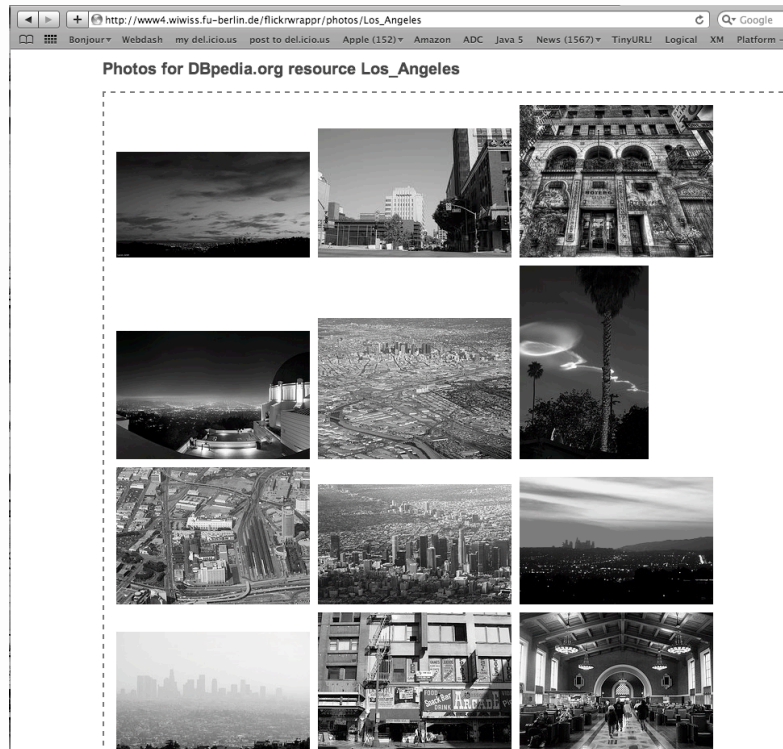
Taxonomies/Ontologies

RDF

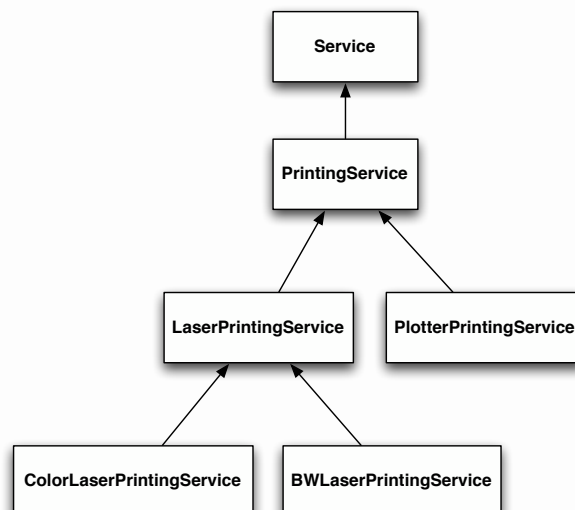
Data model/Linkage

Web Architecture/REST

Identity/Handles/
Negotiated Form



Service Taxonomy



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/subject/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix : <http://example.com/examples/semanticsoa#> .
```

```
:BWLaserPrintingService
rdfs:subClassOf :LaserPrintingService ;
    skos:prefLabel "B&W Laser Printing Service" .

:ColorLaserPrintingService
rdfs:subClassOf :LaserPrintingService ;
    skos:prefLabel "Color Laser Printing Service" .

:LaserPrintingService rdfs:subClassOf :PrintingService ;
    skos:prefLabel "Laser Printing Service" .

:PlotterPrintingService rdfs:subClassOf :PrintingService ;
    skos:prefLabel "Plotter Printing Service" .

:PrintingService rdfs:subClassOf :Service.

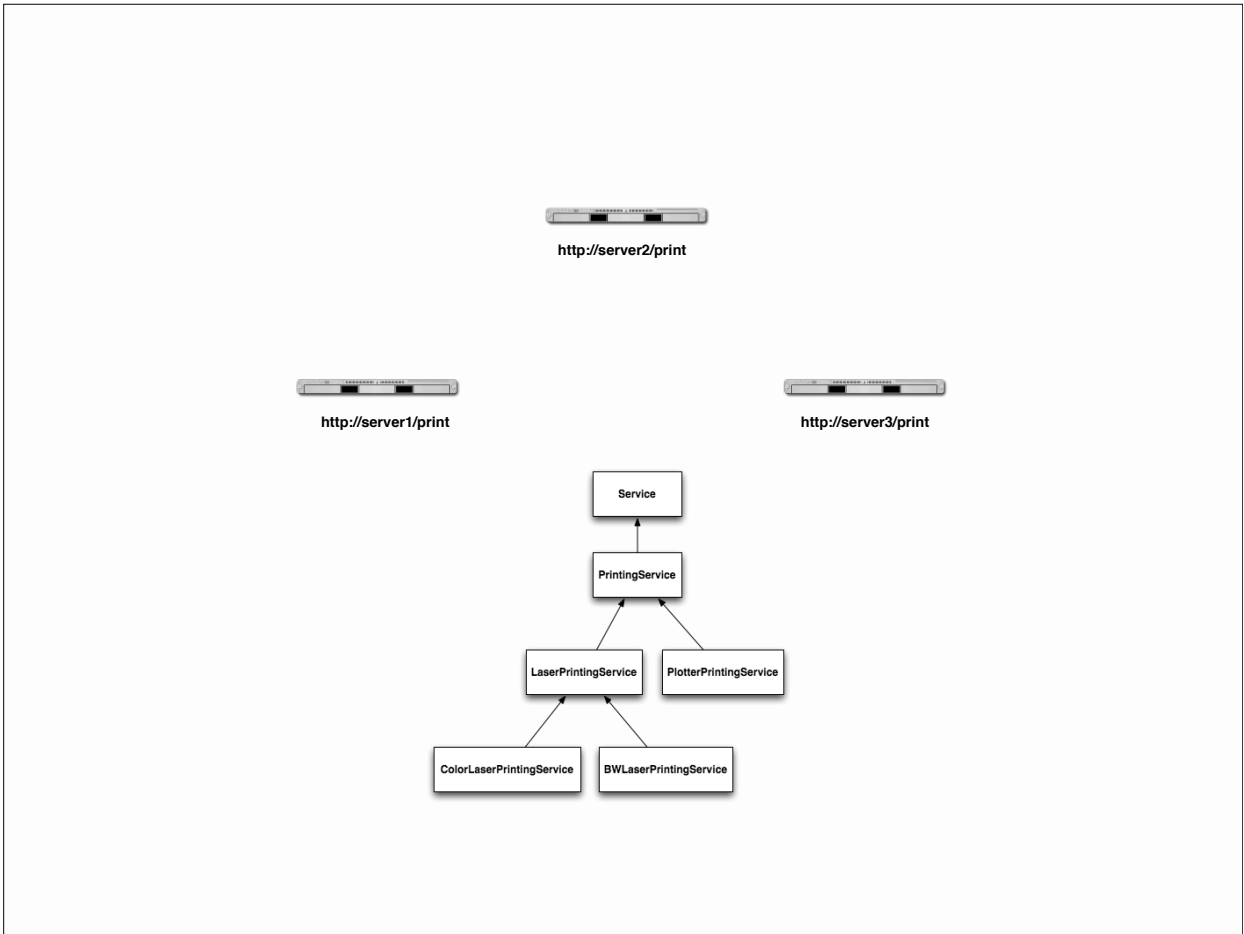
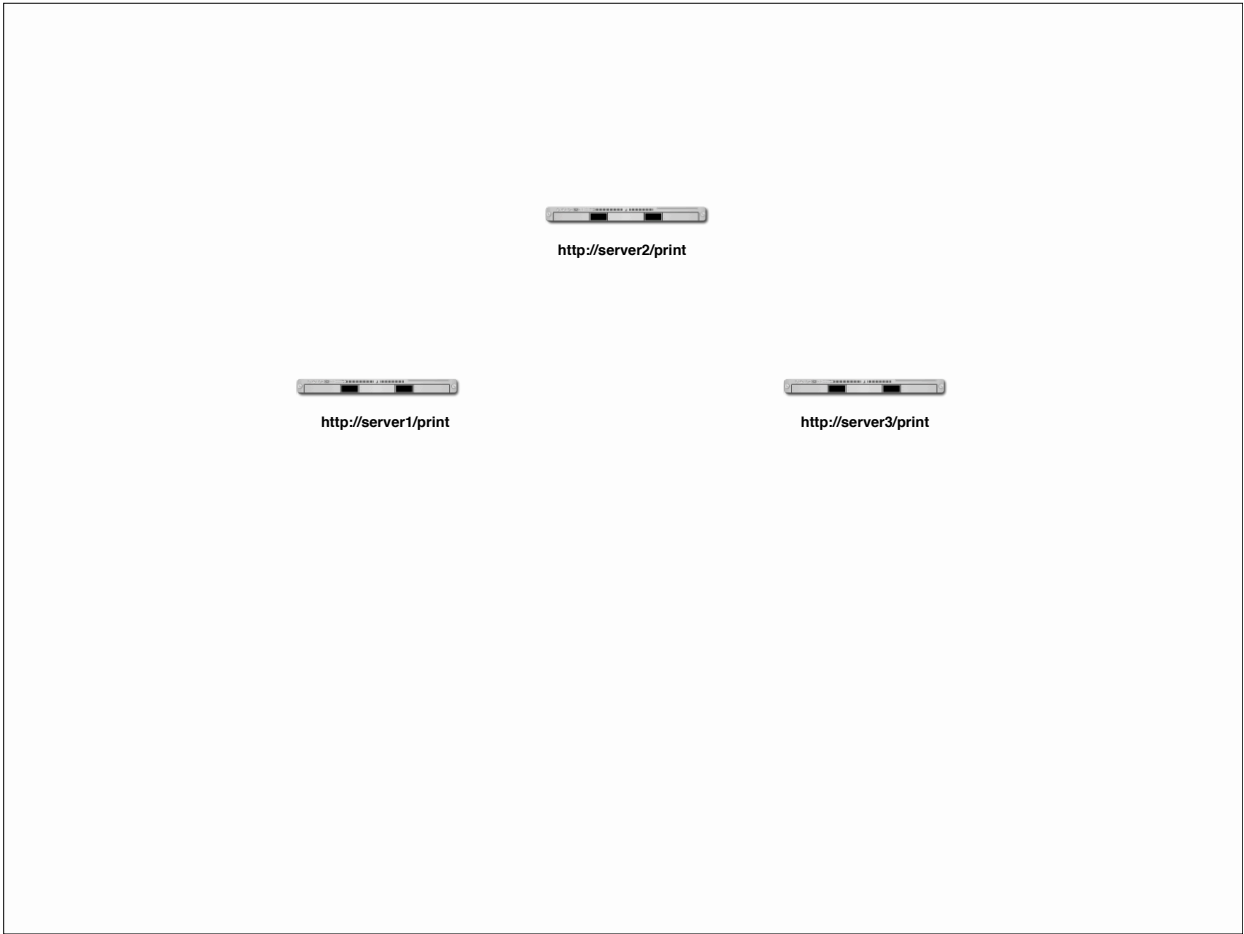
:Service a owl:Class ;
    rdfs:comment "An rdfs:Class of services." .
```

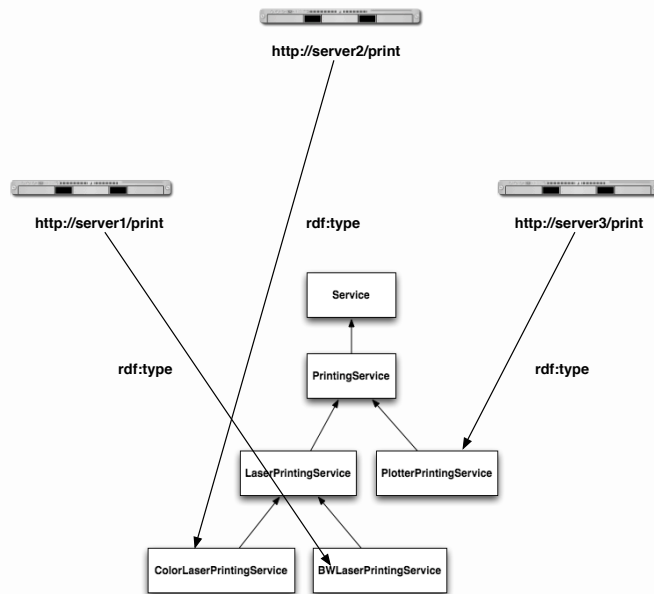
```
// Instance Information
```

```
<http://server1/print1> a :BWLaserPrintingService ;
    dc:description "HP LaserJet 12 - Rm 212" .

<http://server2/print2> a :ColorLaserPrintingService ;
    dc:description "Brother Color Laser 2000 - Rm 500" .

<http://server3/print3> a :PlotterPrintingService ;
    dc:description "PPlot+ 9000 - Basement" .
```





`http://example.com/examples/semanticsoa#PrintingService`

`http://server3/print3`
PPlot+ 9000 - Basement
`http://server1/print1`
HP LaserJet 12 - Rm 212
`http://server2/print2`
Brother Color Laser 2000 - Rm 500

`http://example.com/examples/semanticsoa#LaserPrintingService`

`http://server1/print1`
HP LaserJet 12 - Rm 212
`http://server2/print2`
Brother Color Laser 2000 - Rm 500

“Another theme ... is that self-organization and formal organization are not simple alternatives. Nor are they simply complementary. They live in tension with one another.”

-- Seely Brown and Duguid

Introduction

REST

Semantic Web

NetKernel

Semantic Composition

This software transformation, whatever it is, is coming. It must come; because we simply cannot keep piling complexity upon complexity. We need some new organizing principle that revamps the very foundations of the way we think about software and opens up vistas that will take us into the 22nd century.

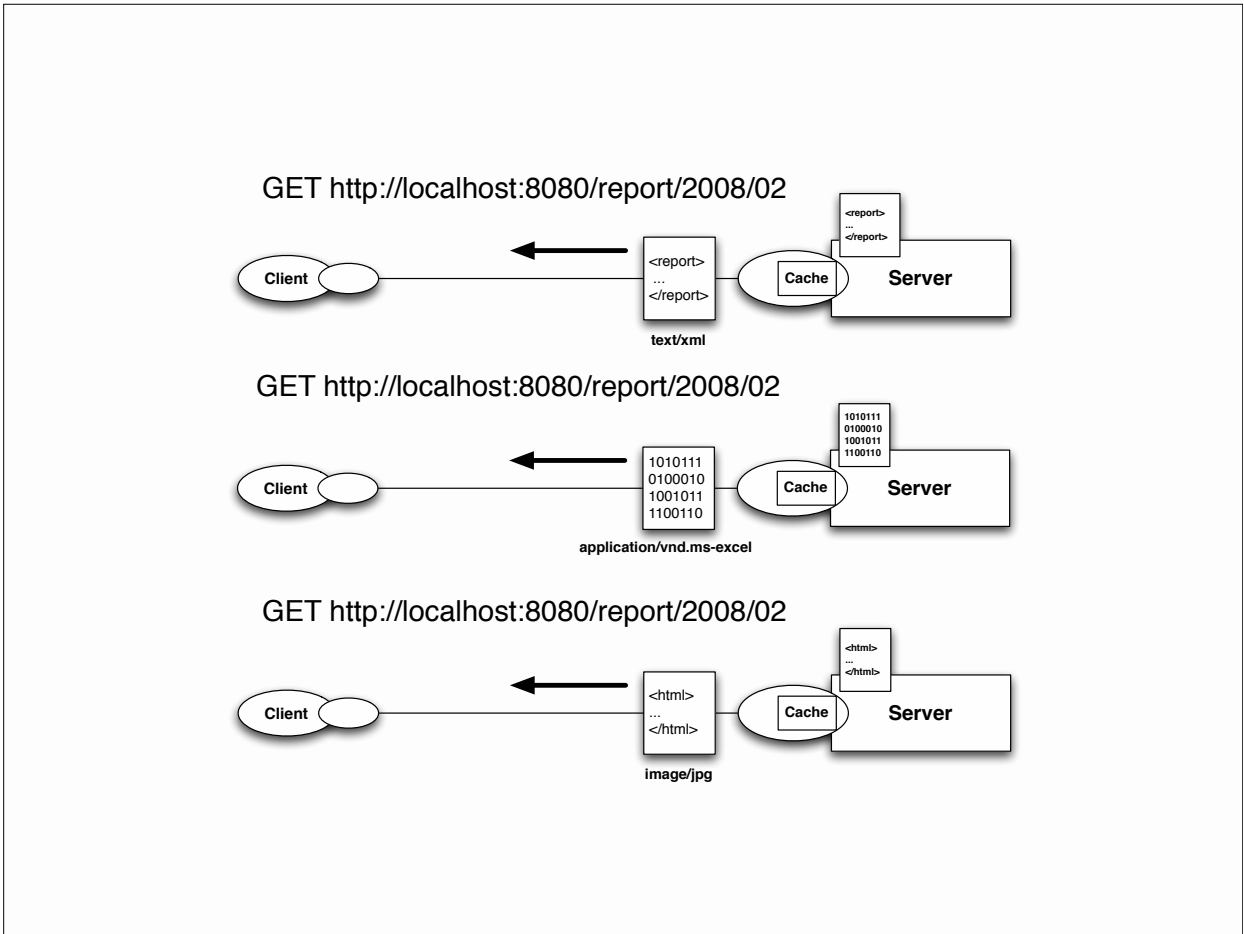
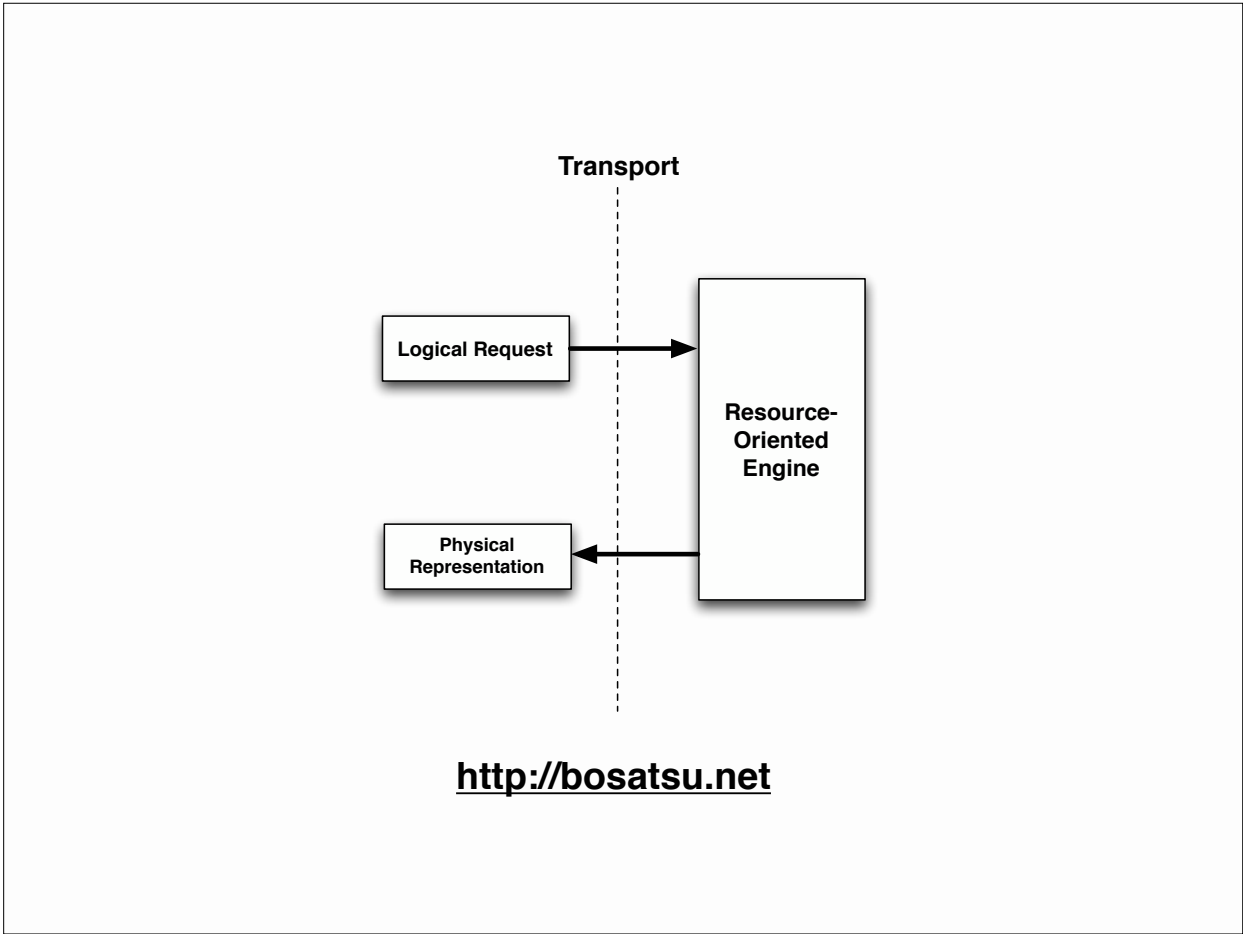
- “Uncle Bob” Martin

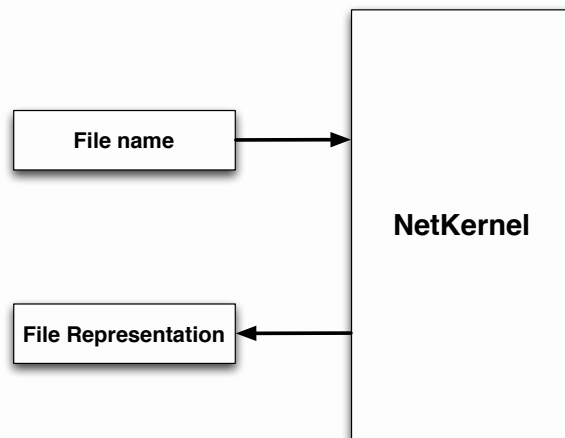
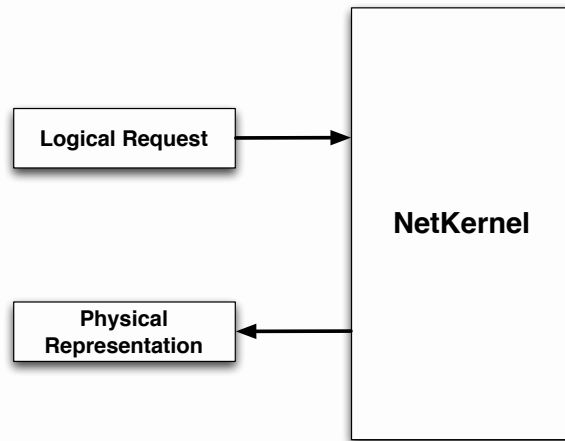
[http://blog.objectmentor.com/articles/2010/07/05/
software-calculus-the-missing-abstraction](http://blog.objectmentor.com/articles/2010/07/05/software-calculus-the-missing-abstraction)

Resource-Oriented MicroKernel

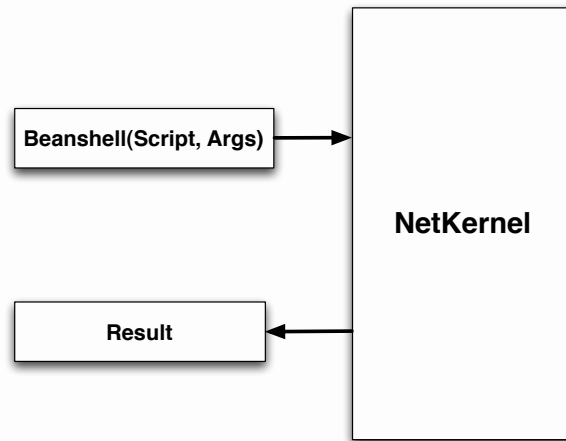
Yes, But What is It?

- ⊗ Software platform w/ properties of The Web
- ⊗ Architecture toolkit
- ⊗ A new set of abstractions for thinking about software

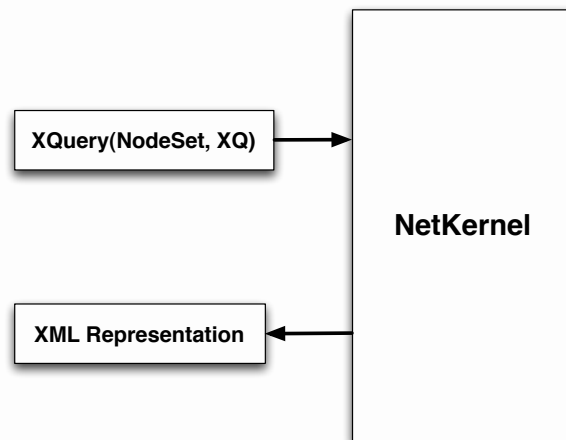




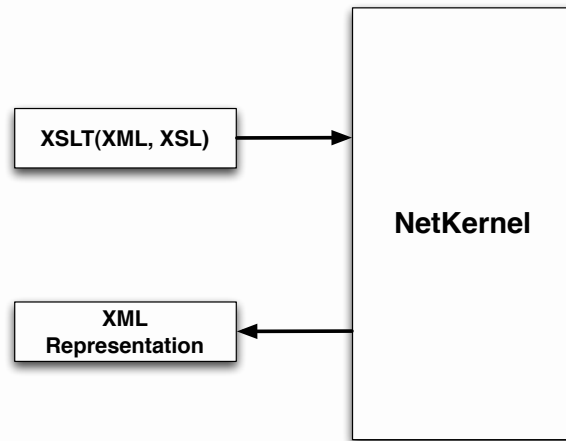
res:/data/somefile.txt



active:beanshell+operator@hello.bsh+arg@foo.xml

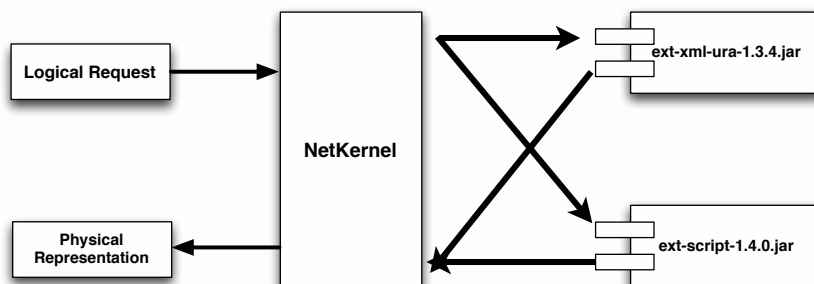


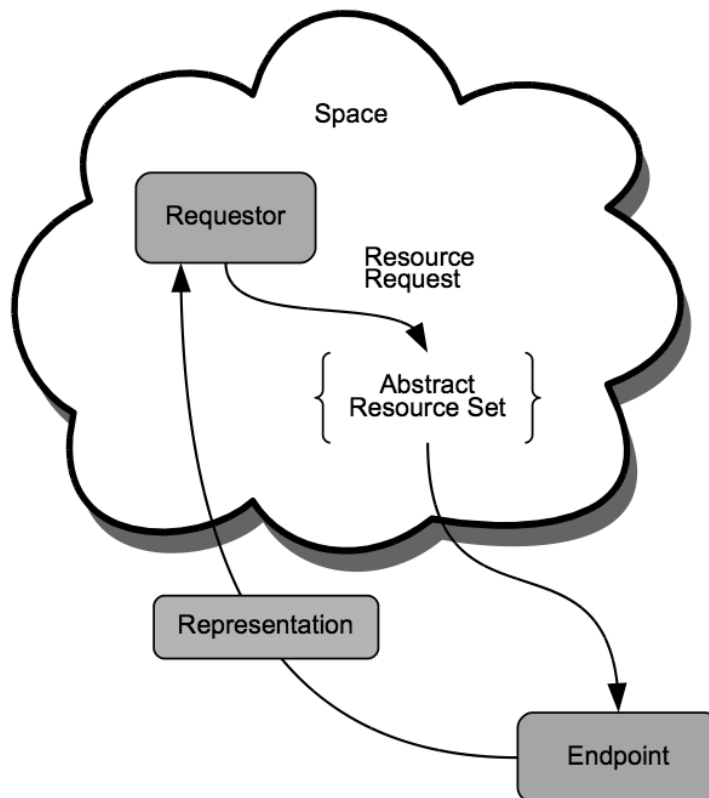
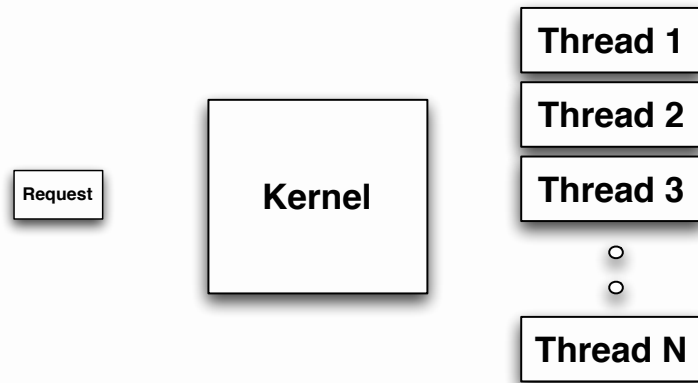
active:xquery+operator@hello.xq+arg@foo.xml



active:xslt+operator@stylesheet.xsl+arg@foo.xml

~~active:xslt+operator@stylesheet.xsl+arg@foo.xml~~





Endpoint Classification

Transport

Initiates Root Requests
based on an event

Accessor

Responds to Requests,
Optionally Initiates Requests,
Responds with Representations

Transreptor

Responds to Representation
type transformation Requests,
Optionally Initiates Requests

Overlay

Bridges requests from one space
To another

(**NetKernel
Walkthrough**)

```
(  
    function fib(n)  
    {  
        if(n<2) return n;  
        else  
        {  
            return fib(n-1)+fib(n-2);  
        }  
    }  
)
```

```
(  
    <endpoint>  
        <grammar>fib(  
            <group name="n">  
                <regex type="integer" />  
            </group>  
        </grammar>  
        <request>  
            <identifier>active:groovy</identifier>  
            <argument name="operator">res:/org/  
netkernel/tutorial/embedded/endpoint/  
FibonacciEndpoint.gy</argument>  
            <argument name="n">arg:n</argument>  
        </request>  
    </endpoint>  
)
```

```

<endpoint>
  <grammar>fib(
    <group name="n">
      <regex type="integer" />
    </group>)
  </grammar>
  <request>
    <identifier>active:groovy</identifier>
    <argument name="operator">res:/org/
netkernel/tutorial/embedded/endpoint/
FibonacciEndpoint.gy</argument>
    <argument name="n">arg:n</argument>
  </request>
</endpoint>

```

fib(3)

```

<endpoint>
  <grammar>fib(
    <group name="n">
      <regex type="integer" />
    </group>)
  </grammar>
  <request>
    <identifier>active:groovy</identifier>
    <argument name="operator">res:/org/
netkernel/tutorial/embedded/endpoint/
FibonacciEndpoint.gy</argument>
    <argument name="n">arg:n</argument>
  </request>
</endpoint>

```

fib(3) —

active:groovy+operator@FibonacciEndpoint.gy+n@arg:n


```

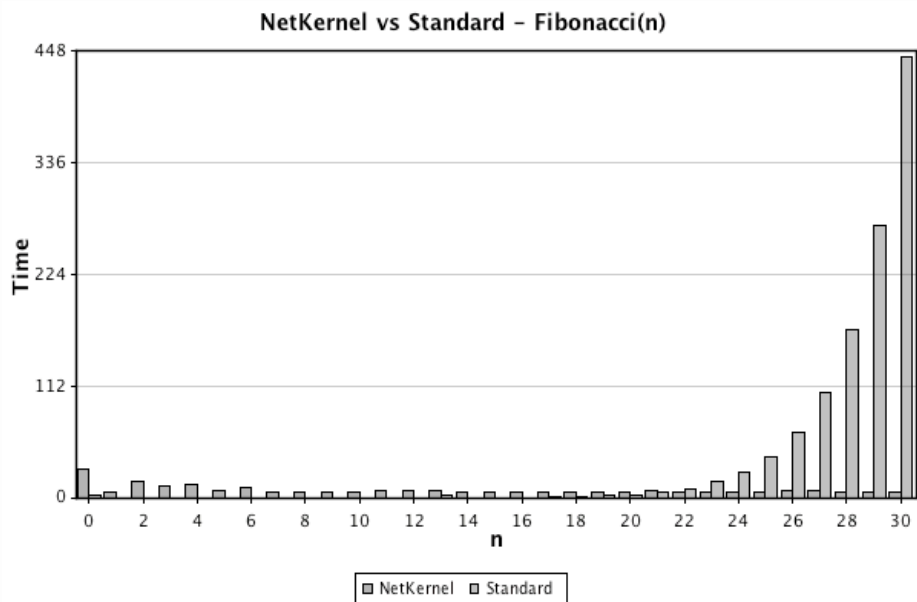
import org.netkernel.layer0.nkf.INKFRequestContext;
import org.netkernel.layer0.nkf.INKFRequestReadOnly;
import org.netkernel.layer0.nkf.NKFException;
import org.netkernel.module.standard.endpoint.StandardAccessorImpl;

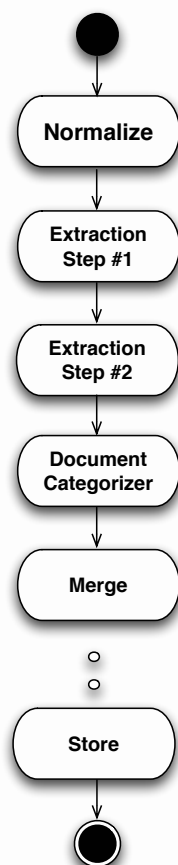
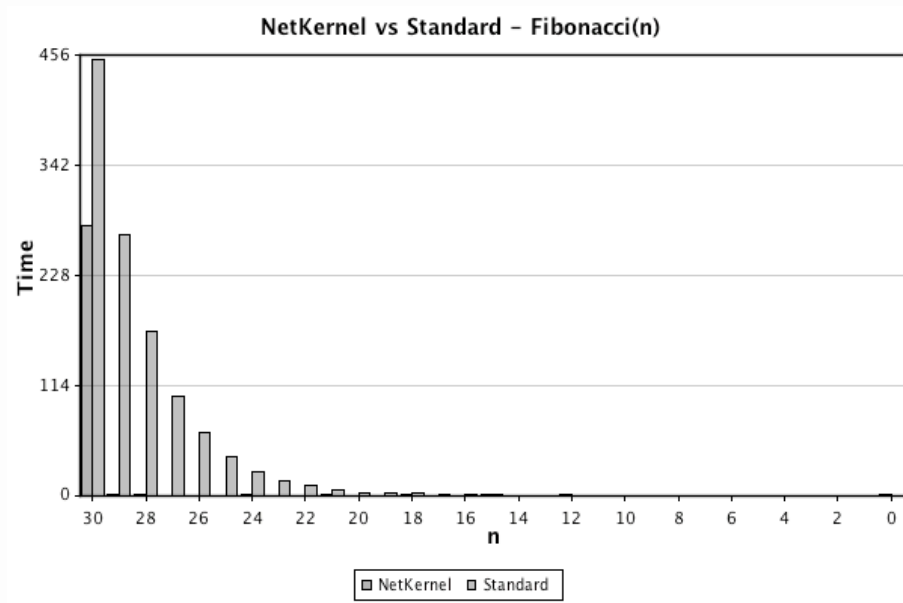
request = context.getThisRequest();
n = Long.parseLong(request.getArgumentValue("n"));
System.out.println("Computing fib for [" + n + "]");

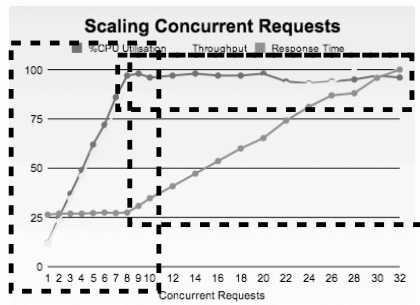
if (n<0) { context.createResponseFrom("Negative values not allowed"); }

if (n==0){ context.createResponseFrom("0") }
if (n==1){ context.createResponseFrom("1") }
if (n>=2)
{
    a = context.source("fib(" + (n-2) + ")", String.class)
    b = context.source("fib(" + (n-1) + ")", String.class)
    answer = Long.parseLong(a)+Long.parseLong(b)
    context.createResponseFrom("" + answer)
}

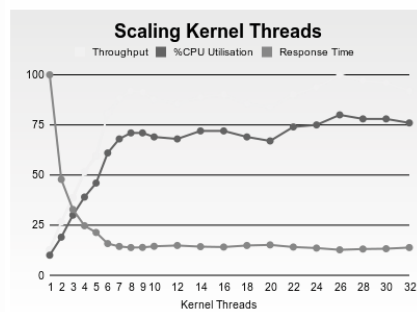
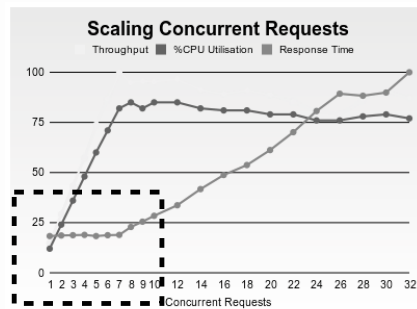
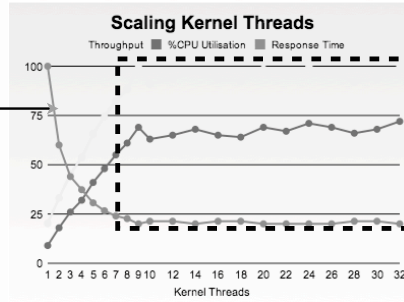
```

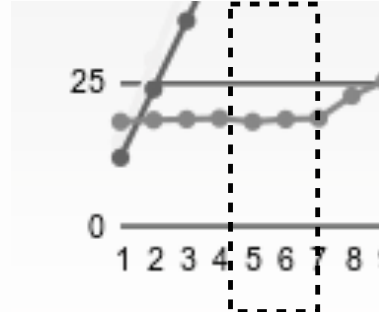
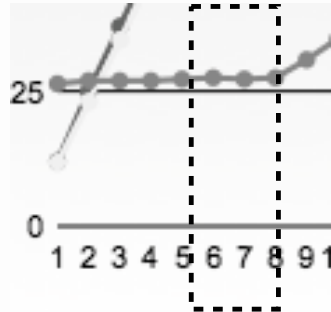






1/x





Introduction

REST

Semantic Web

NetKernel

Semantic Composition

**RDF
Data
Source**

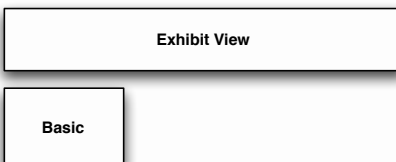
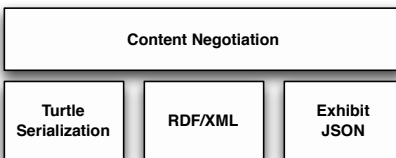
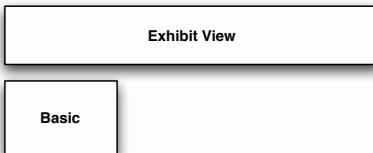
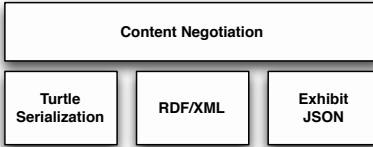
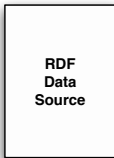
**RDF
Data
Source**

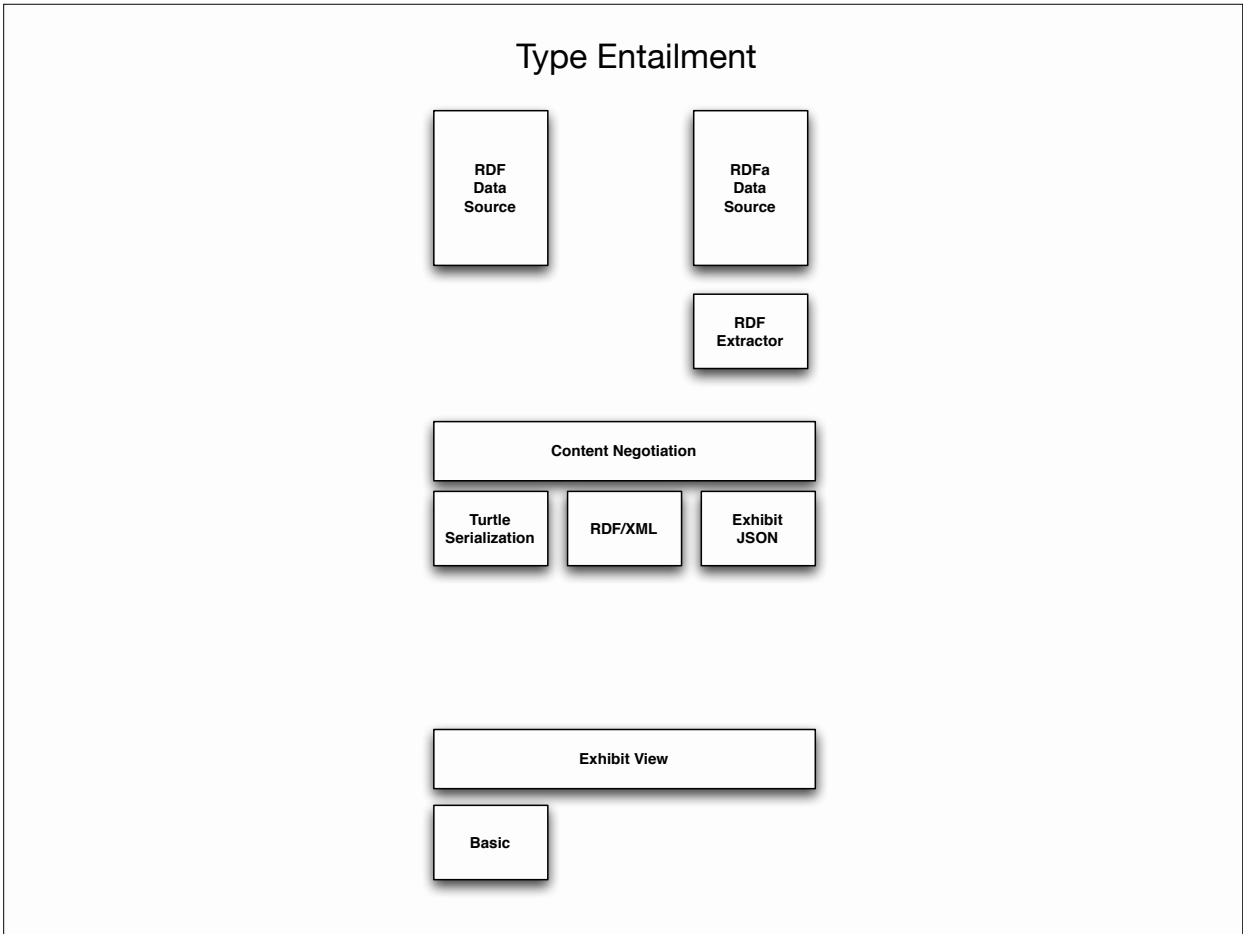
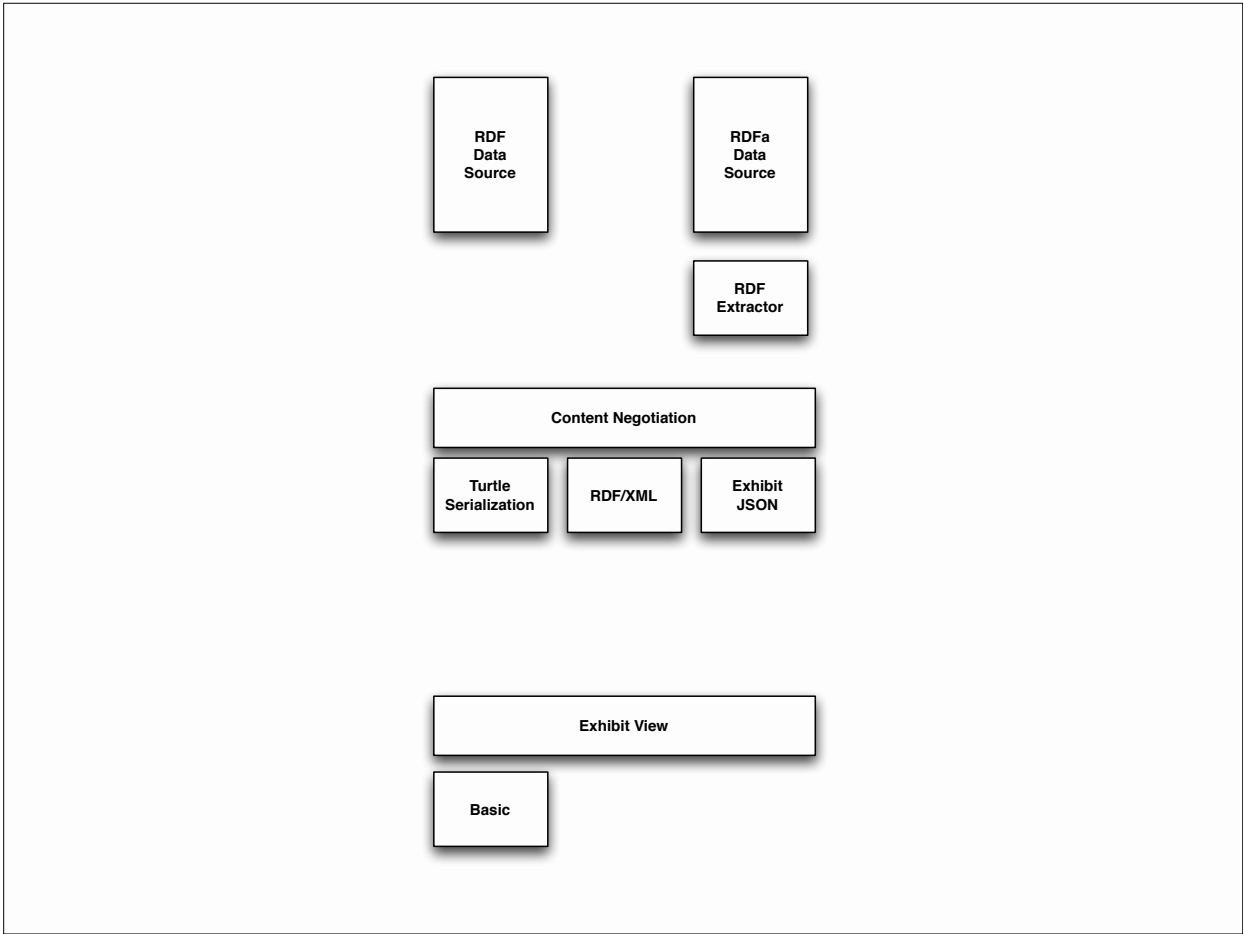
Content Negotiation

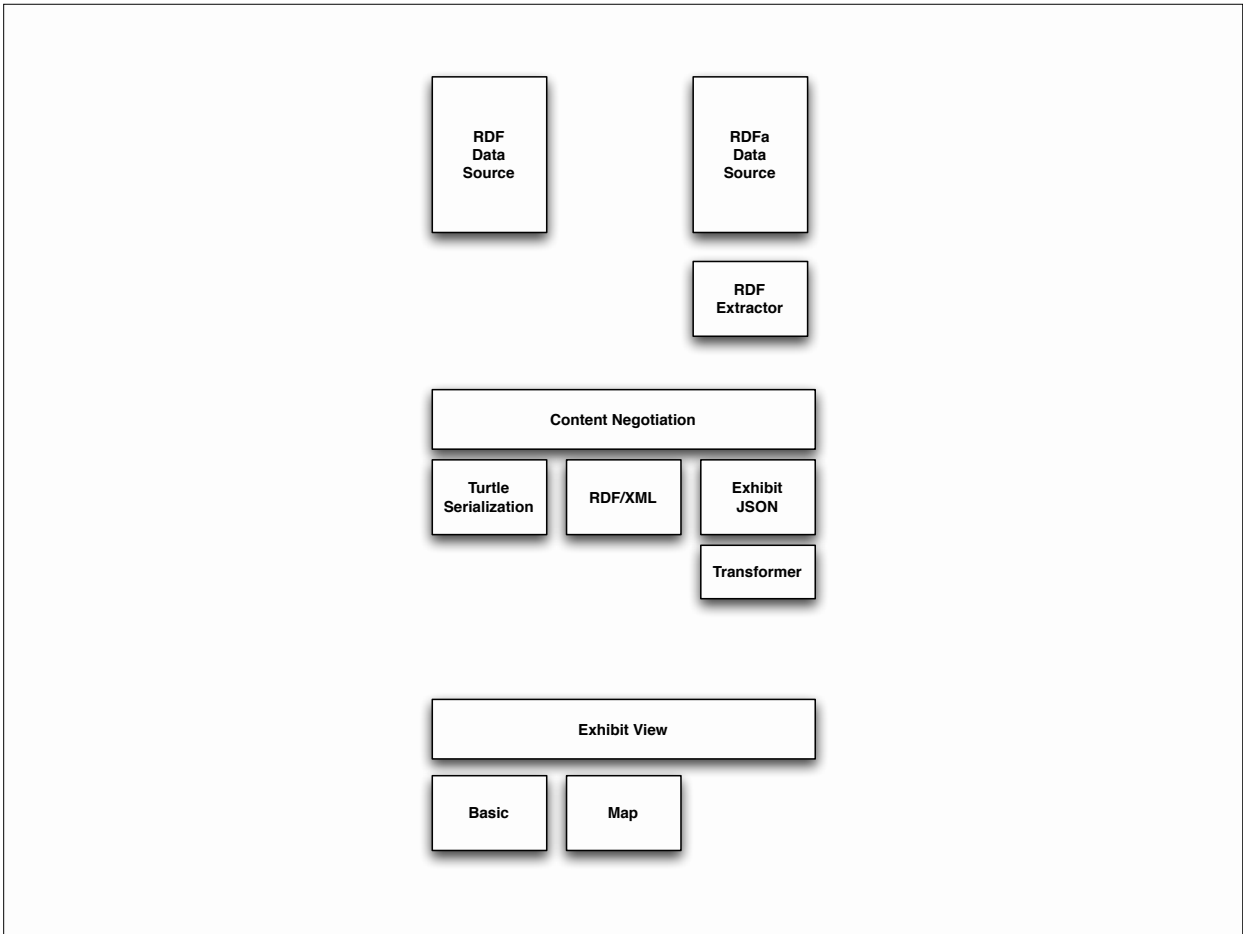
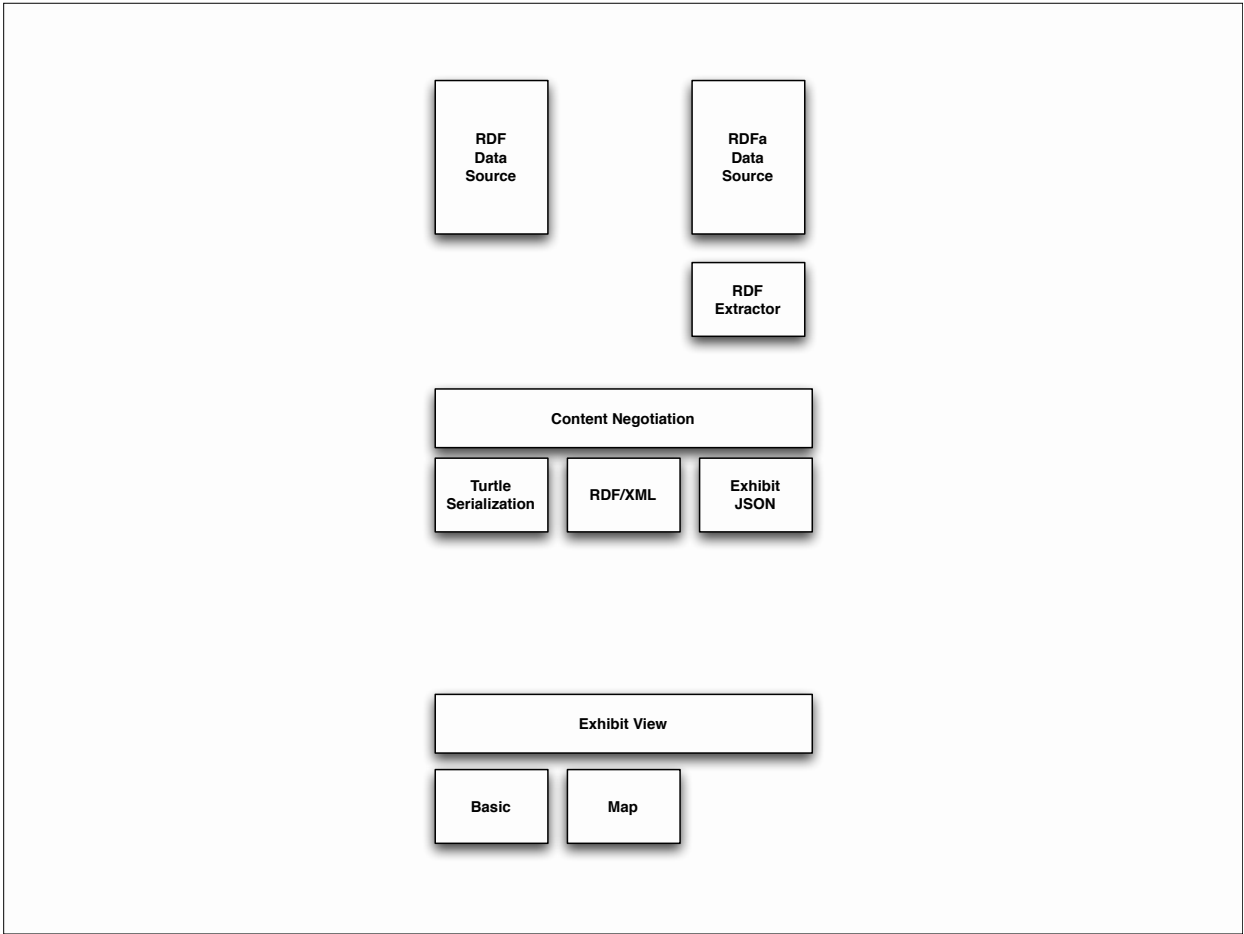
**Turtle
Serialization**

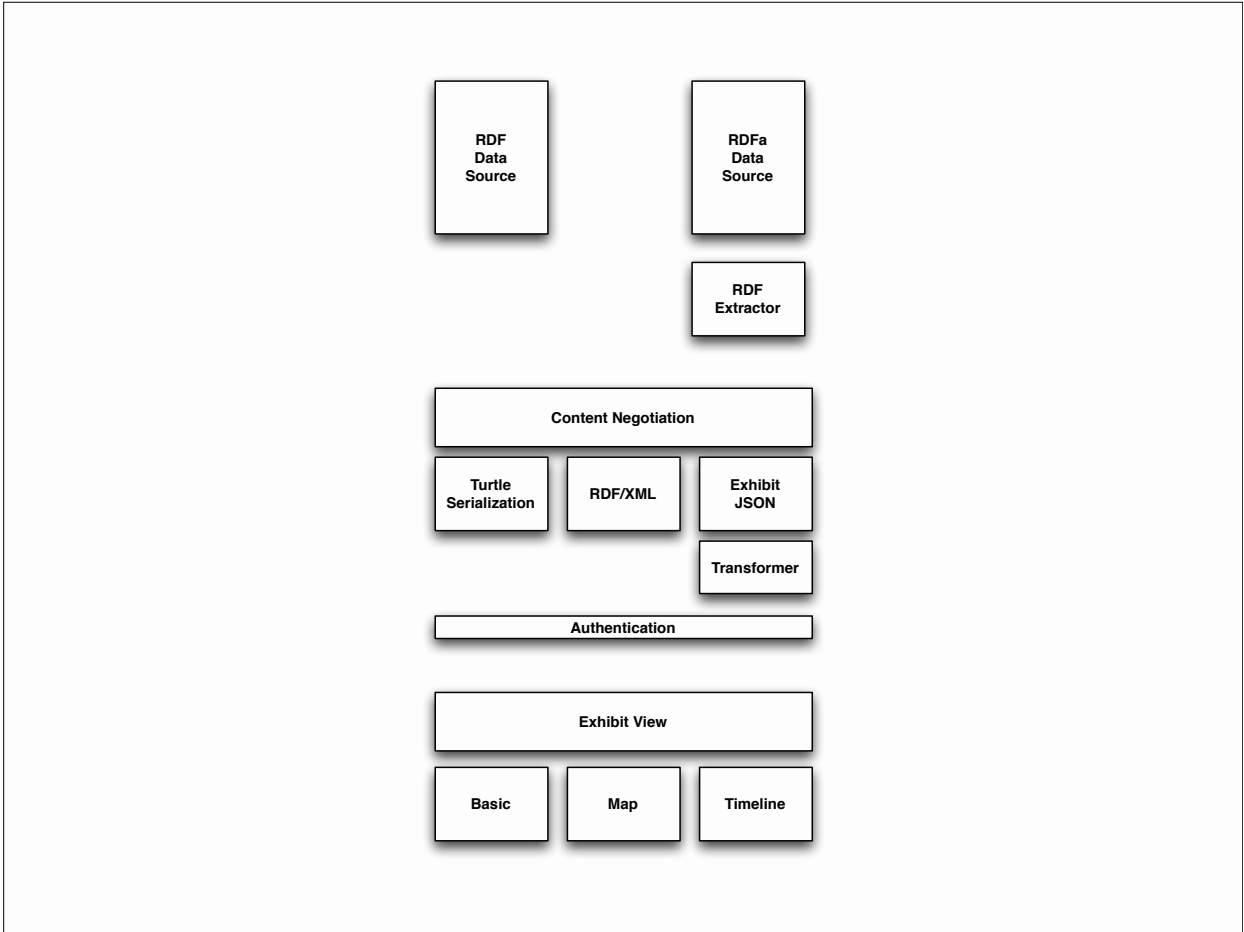
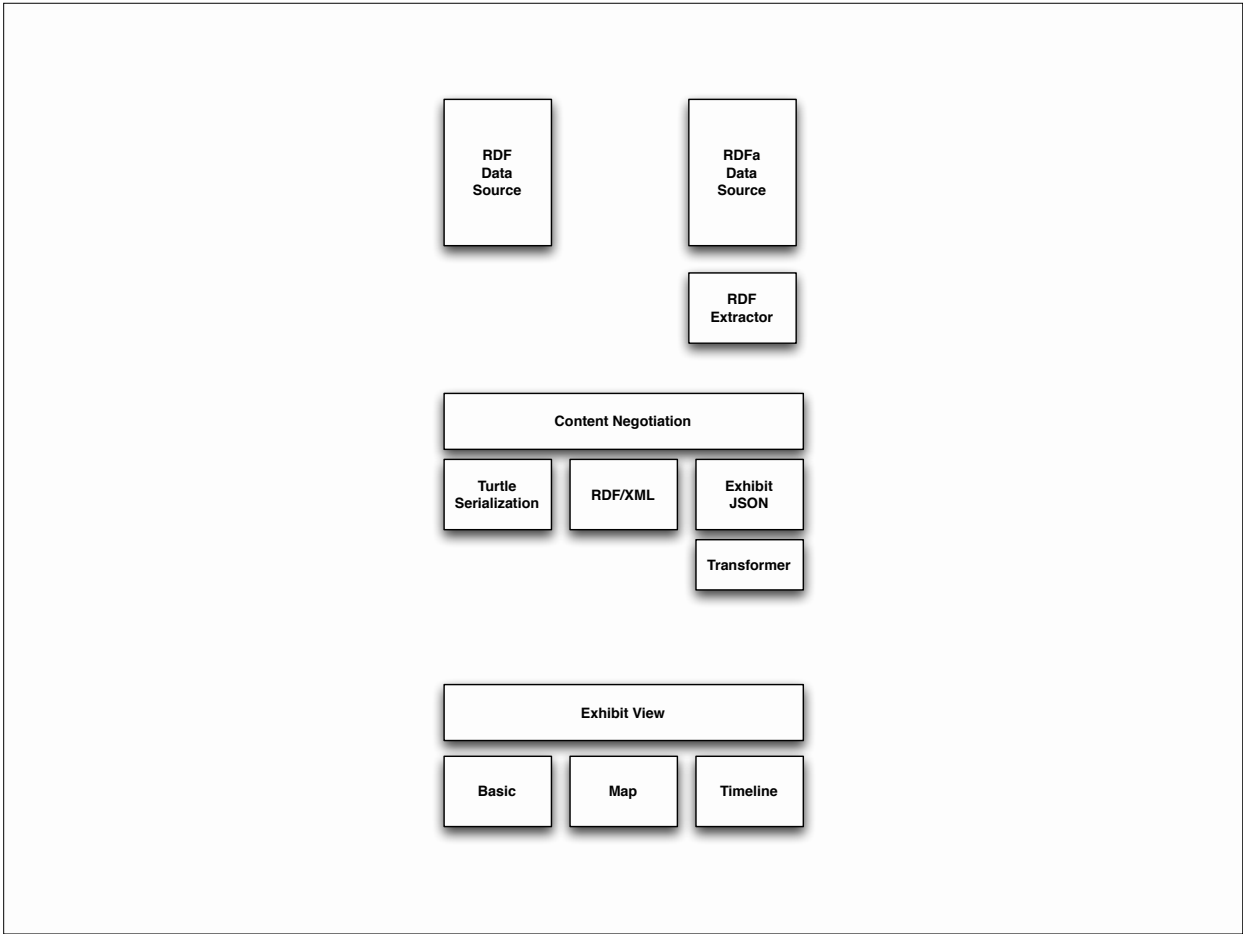
RDF/XML

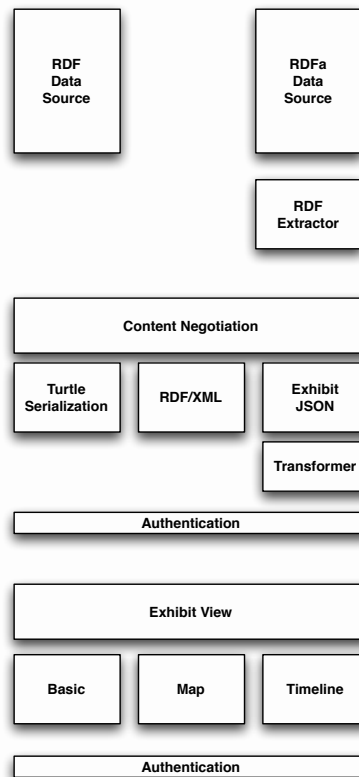
**Exhibit
JSON**











Questions?

brian@bosatsu.net
[@bsletten](https://twitter.com/bsletten)