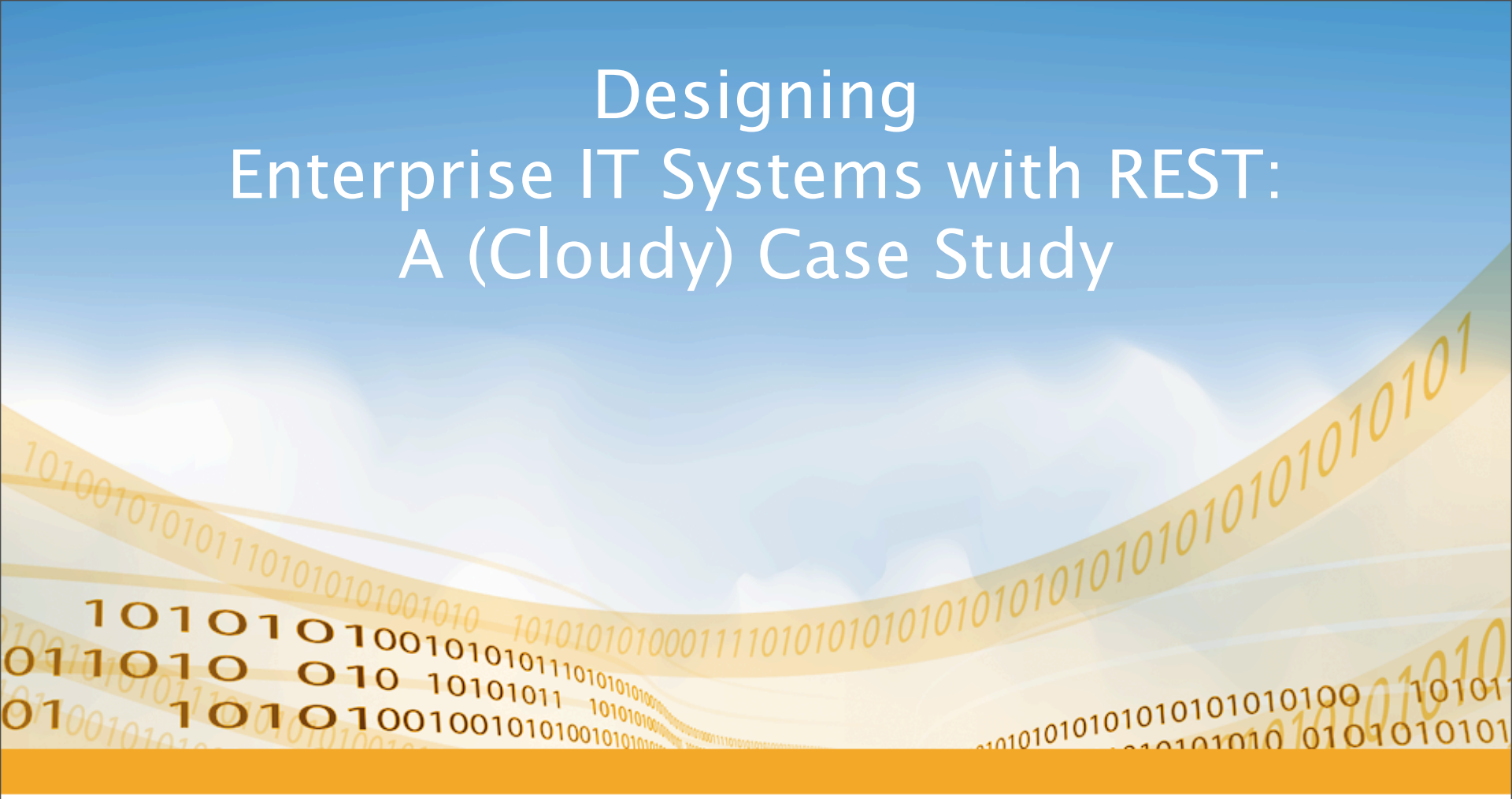
Designing Enterprise IT Systems with REST: A (Cloudy) Case Study



Stuart Charlton
Chief Software Architect, Elastra

Objectives

- What enterprise problem domains does RESTful architecture handle well?
- Understanding Hypermedia as a primary systems abstraction
- Suggested capabilities of RESTful Clients
  - Metadata & Versioning
  - Semantics & Querying
  - Security

About Your Presenter

- Stuart Charlton
  - Canadian,  now in San Francisco
- Chief Architect, Elastra
  - Responsible for technical direction & long-term product strategy
- In prior lives...
  - BEA Systems, Rogers Communications, Financial Services, global training & consulting
- **Stu Says Stuff**
  http://stucharlton.com/blog

Caveats

- REST is defined in Roy Fielding's thesis
  - Go there for truth, I'm merely a theologian
- Web architecture is an evolving field
  - This is most of what I'm talking about
- I don't expect everyone to agree with me
  - Some think actions via HTTP POST are a bad thing
  - I think they're essential; "REST is not File Storage"
- I actually like the Semantic Web
  - ...and thus cannot be trusted

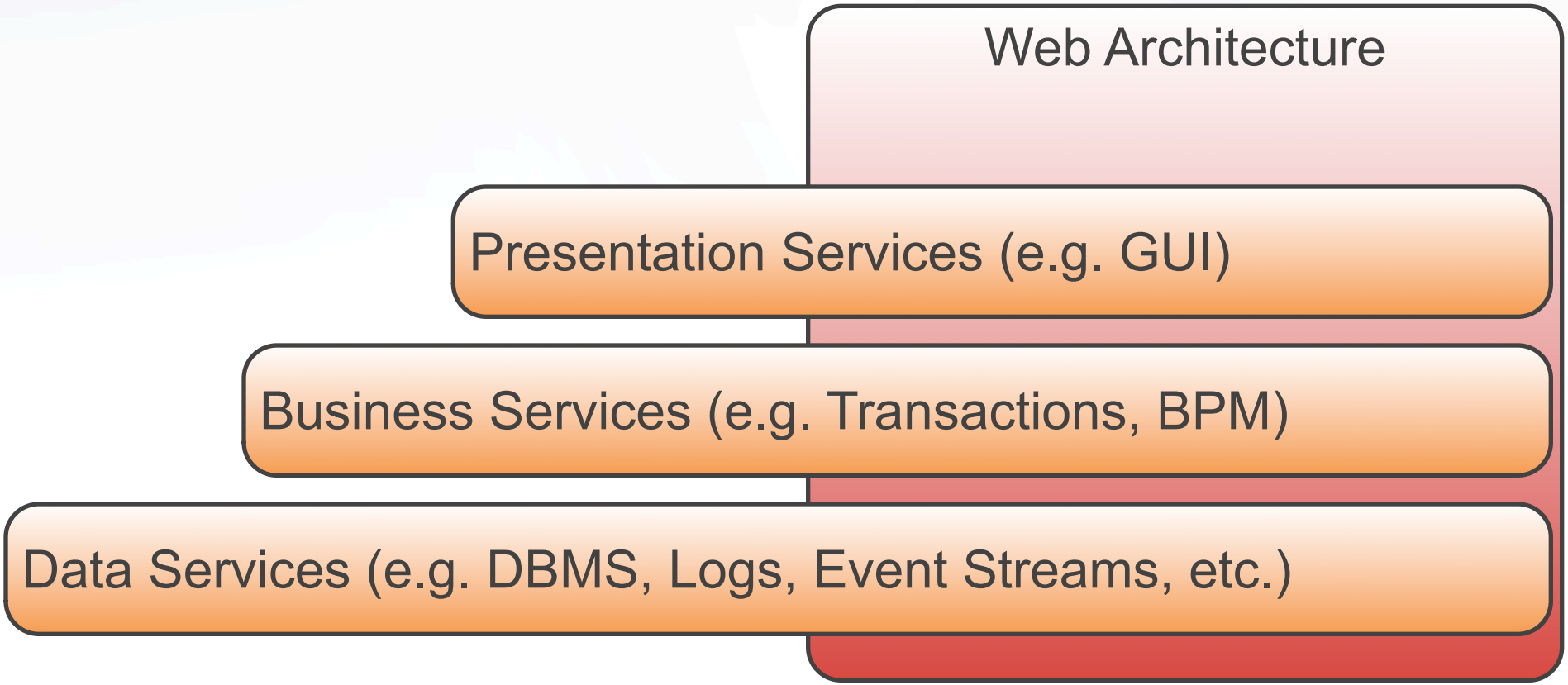
A Simple Reference Architecture

Presentation Services (e.g. GUI)

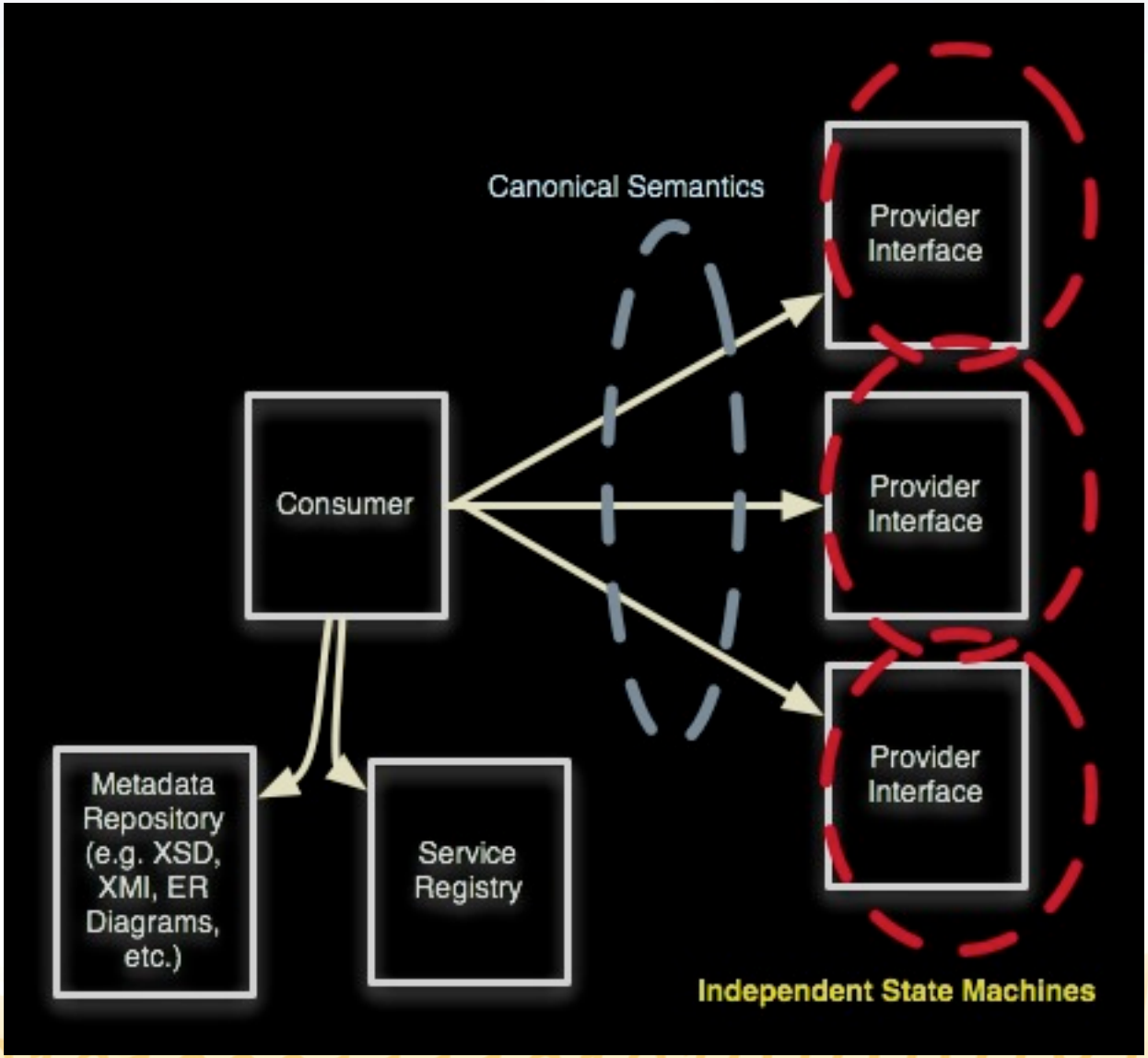
Business Services (e.g. Transactions, BPM)

Data Services (e.g. DBMS, Logs, Event Streams, etc.)

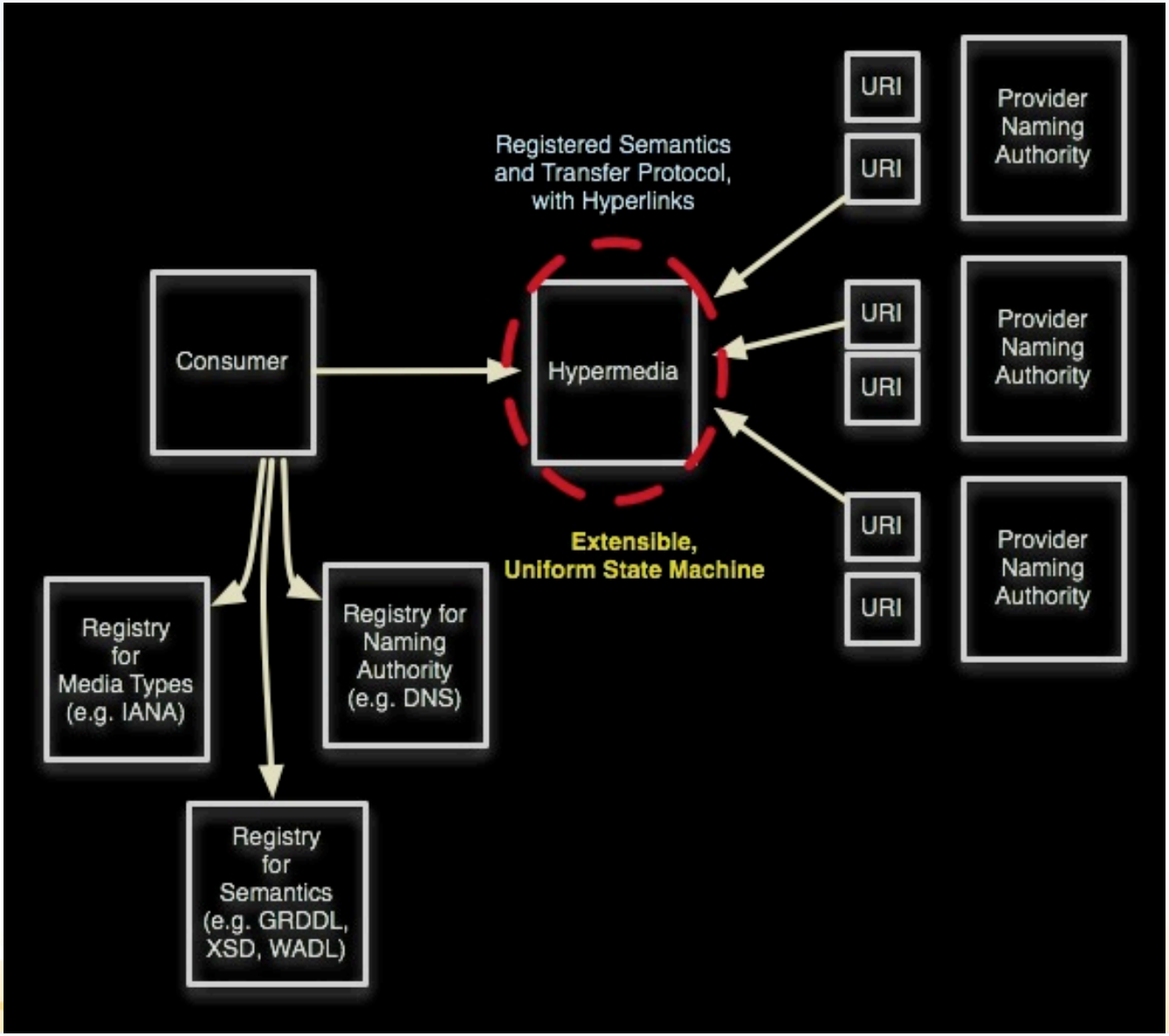
Web architecture helps to burst silos



Classic "Good SOA" Interfaces

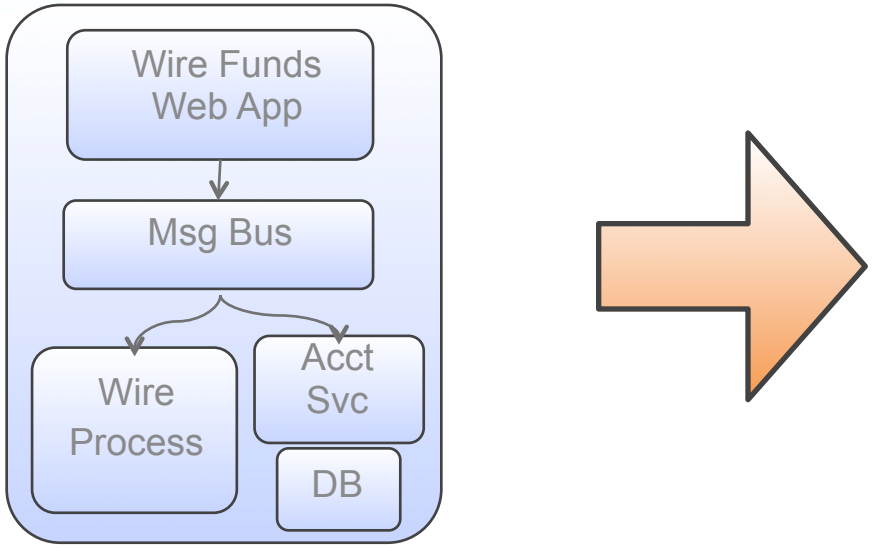


The Hypermedia Alternative



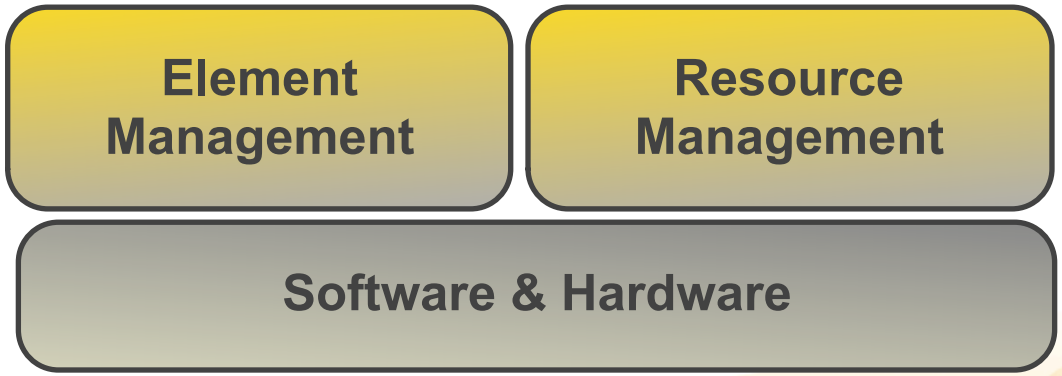
Problem Domain

- A Take on "Enterprise Cloud Computing"
  - Drastically reduced lead times to provision/change
  - Visible, declarative designs & infrastructure



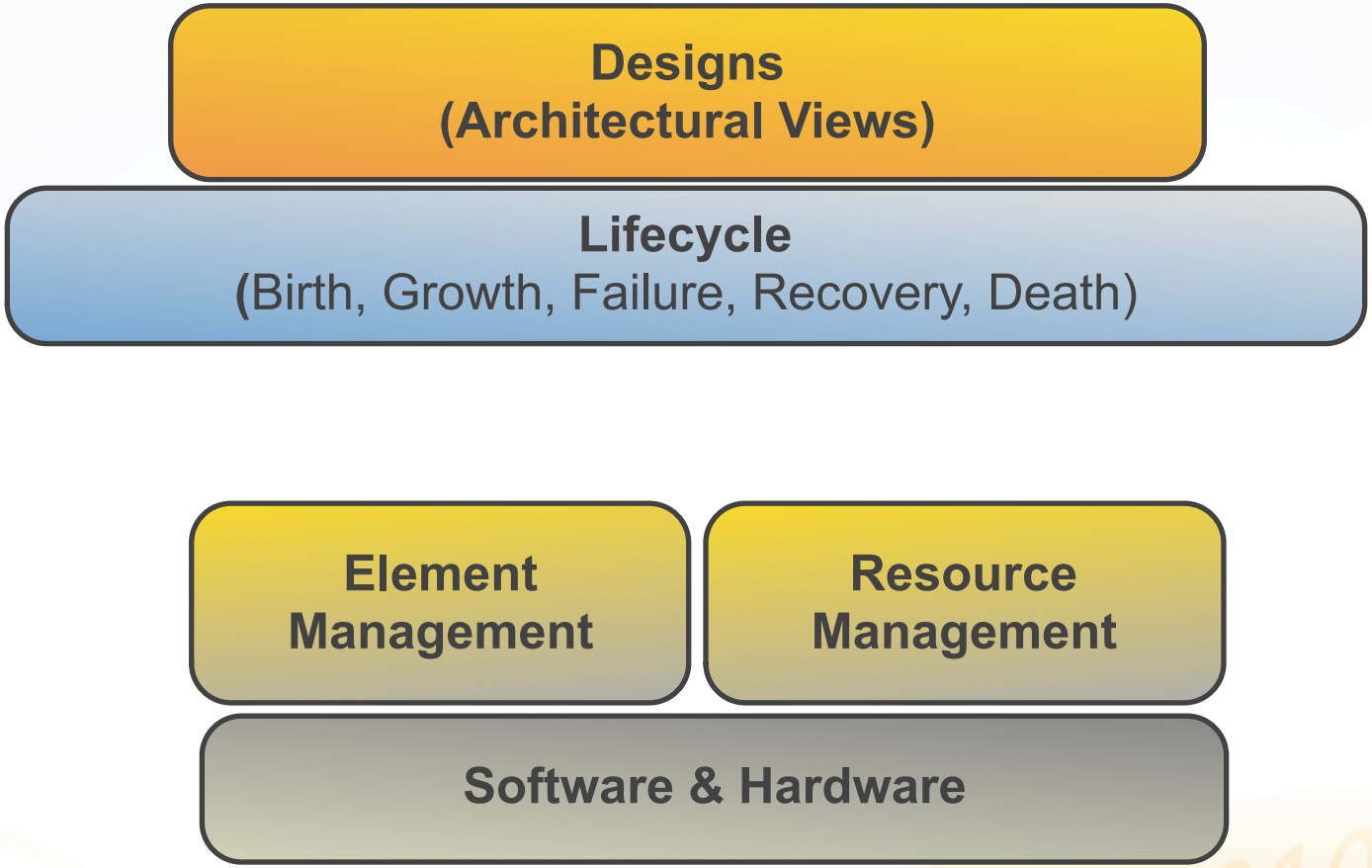
Problem Domain

- IT Services Management & Provisioning

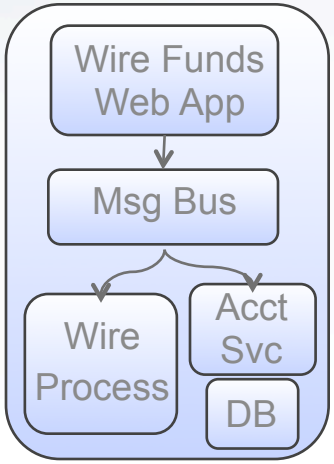


Problem Domain

- Architectural & Change Considerations



Organizational & Geographic Distribution



Design

MQ

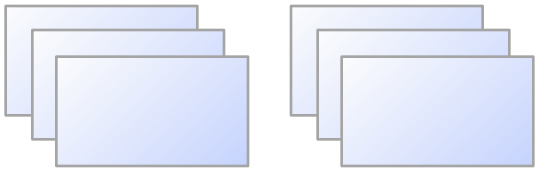
WLS 10.1

MySQL

Packages

Tomcat V 5.5

Configurations



Resource Pool A

amazon web services™

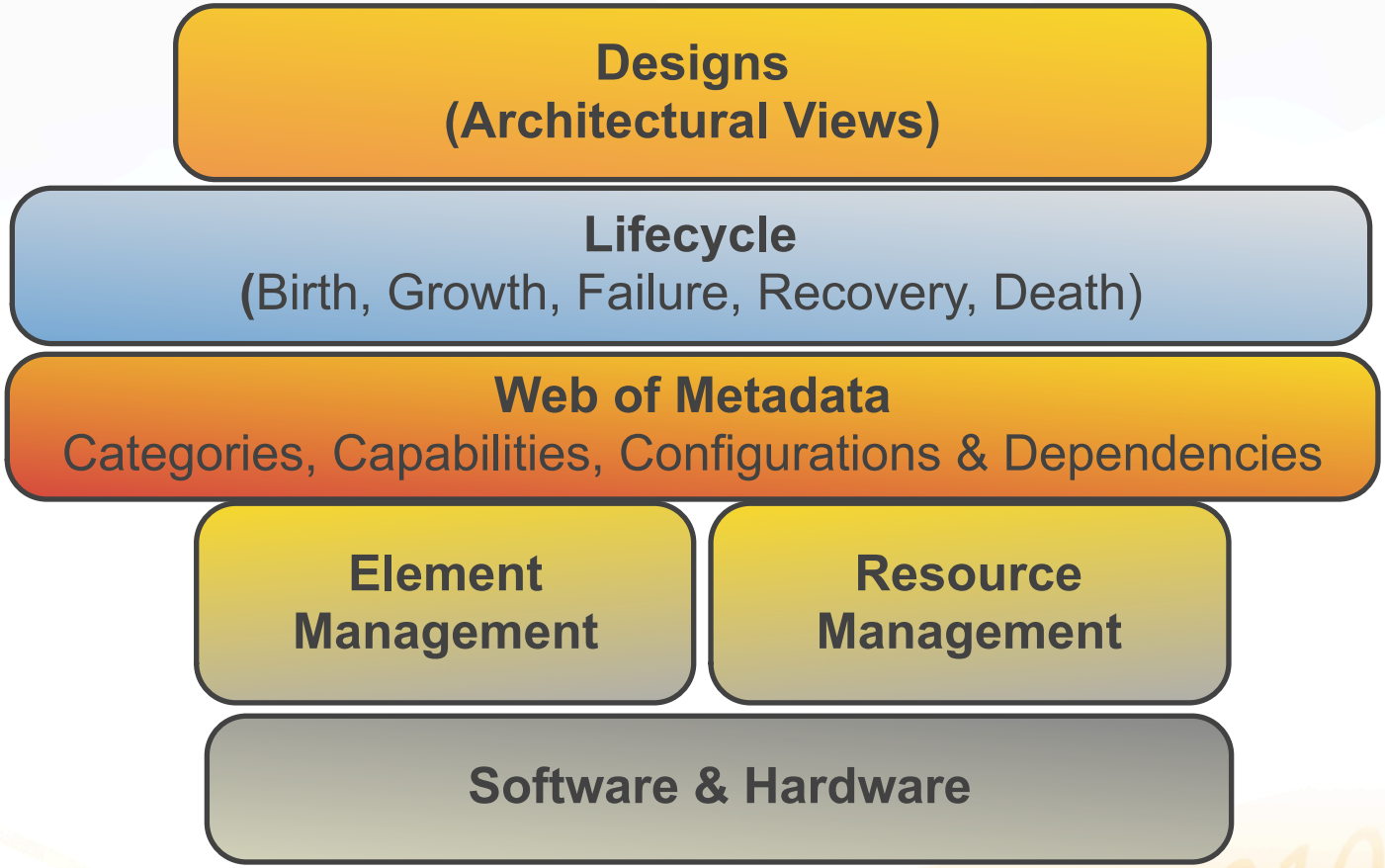
How can you affect change?

Resource Pool B

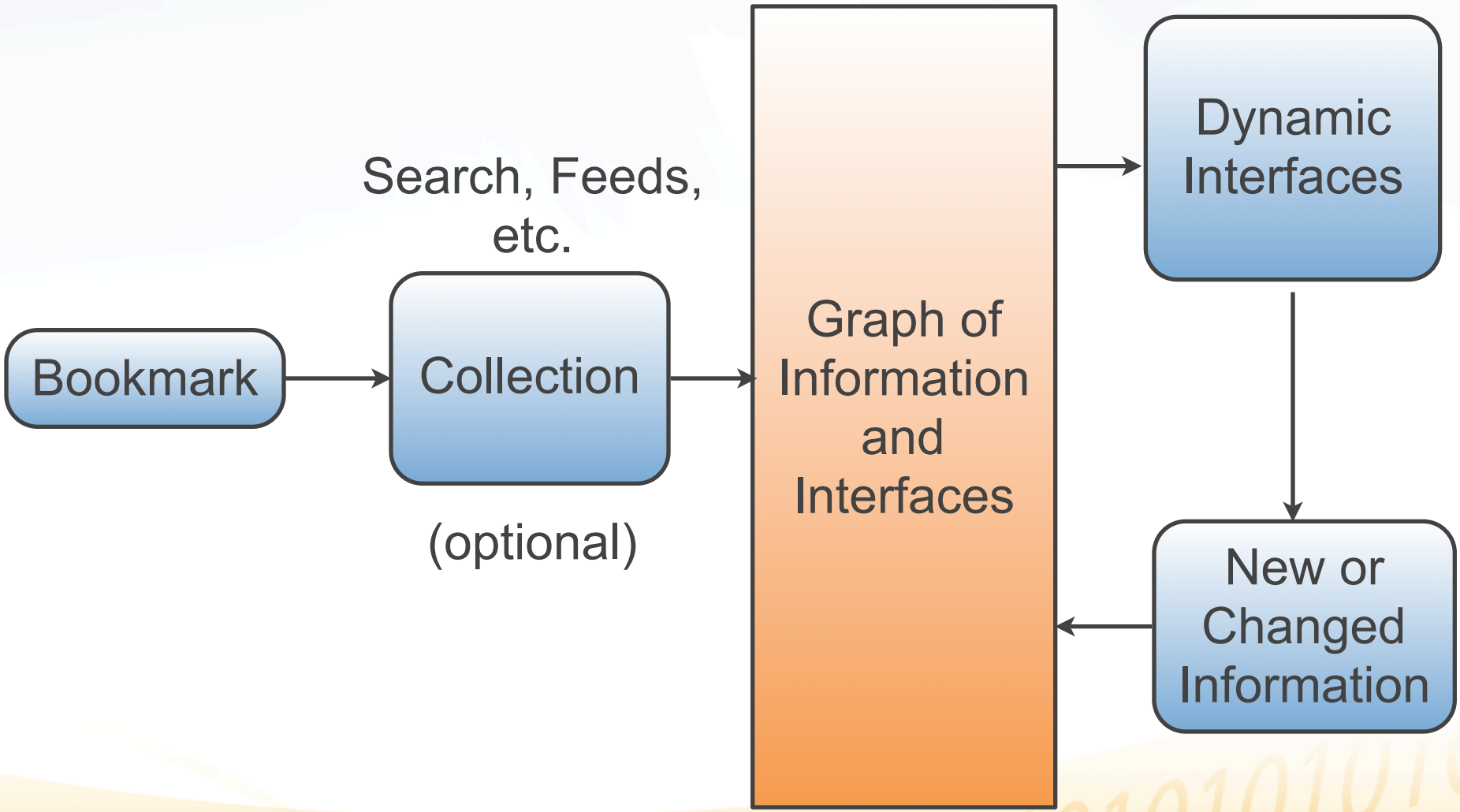
vmware

Hyperlinked Enterprise IT Infrastructure

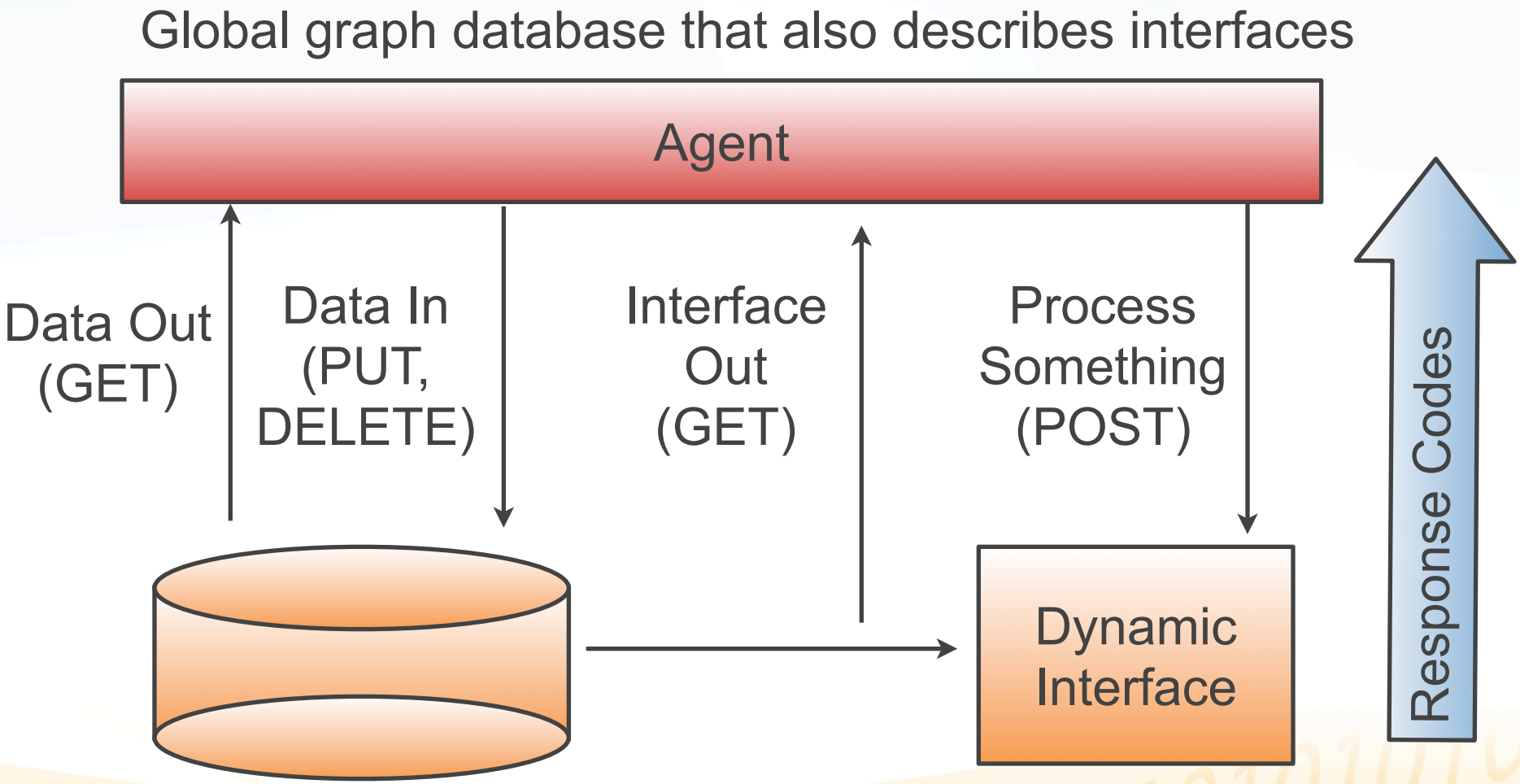
- The Decentralized, Declarative Data Center



Hypermedia Application Flow



Hypermedia is a mix of data and control

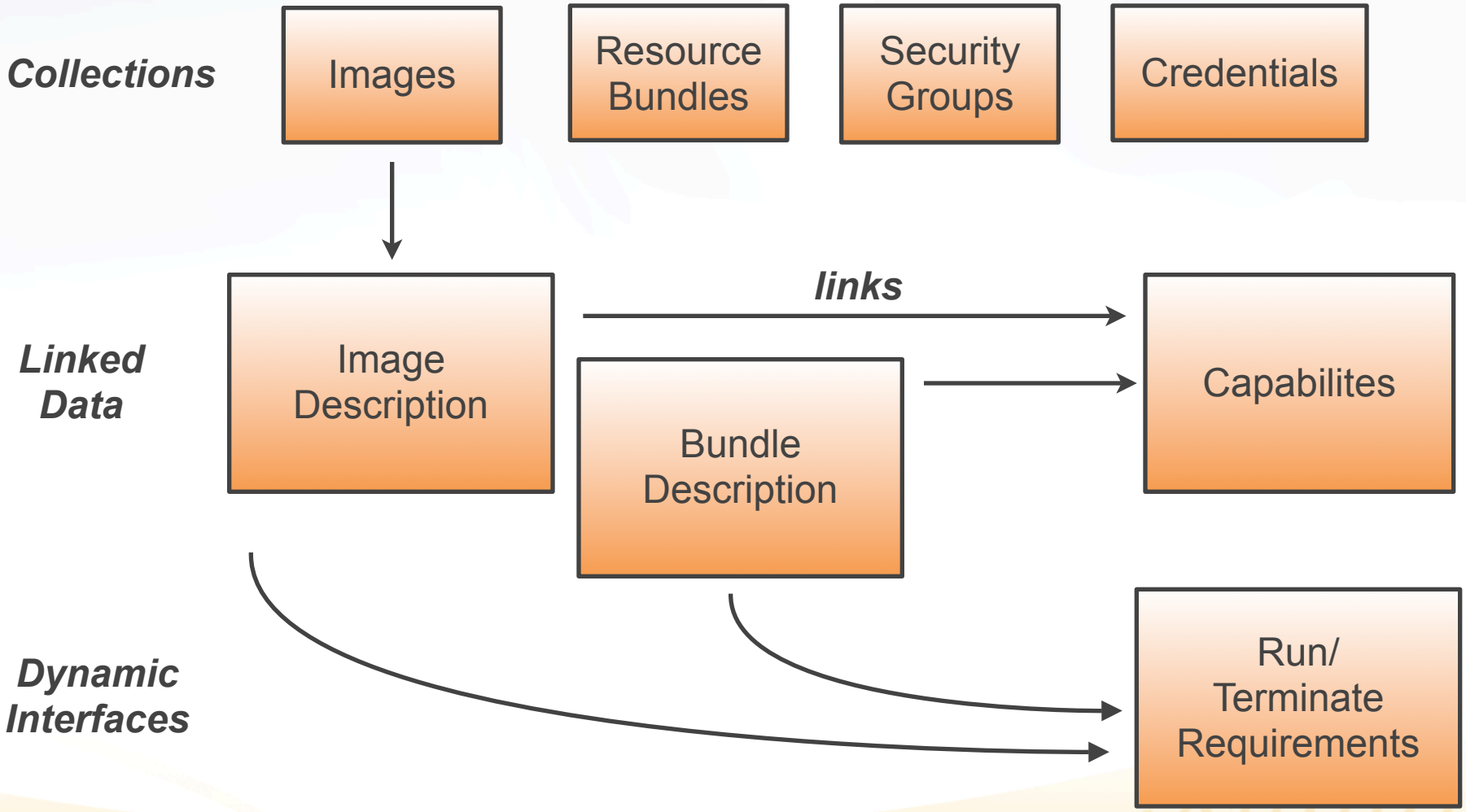


What's a Dynamic Interface?

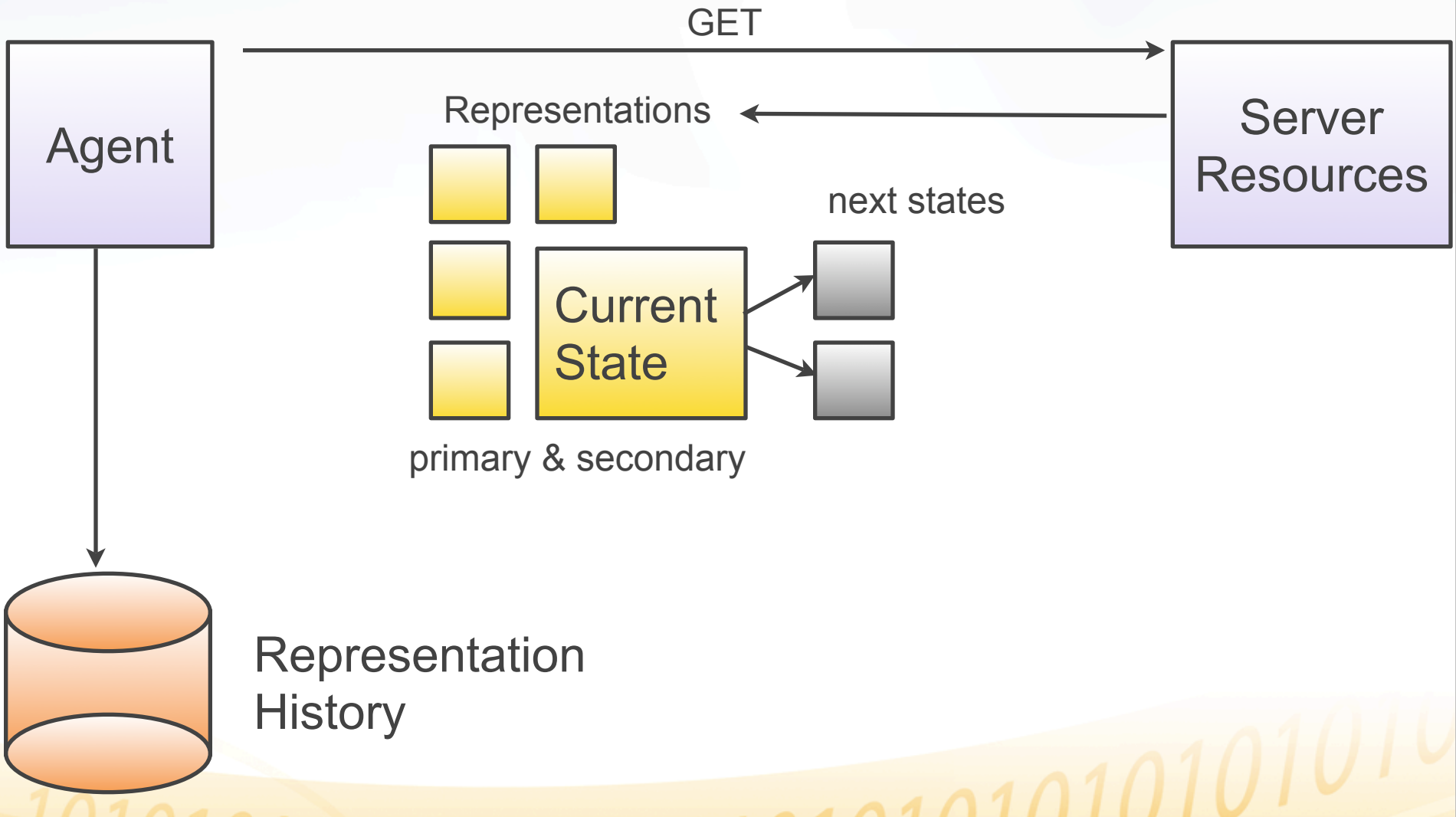
- Interaction port that is **bound at runtime**
  - CORBA Dynamic Invocation Interface
  - java.lang.Reflect
  - Capability negotiation (e.g. TELNET)
- Agent **matches** what they **know** to what's **available**
- The Big Difference? Metadata over Methods
  - The semantics are in the context of the **link**

```
<control about="http://myrobot.com/robot/controls/wave">
  <title>Wave</title>
  <description>This will ask your robot to wave X many times</description>
  <action href="http://myrobot.com/processor">
        <input rel="waveTimes" name="waves" type="xsd:positiveInteger"/>
  </action>
</control>
```

A RESTFul Amazon EC2



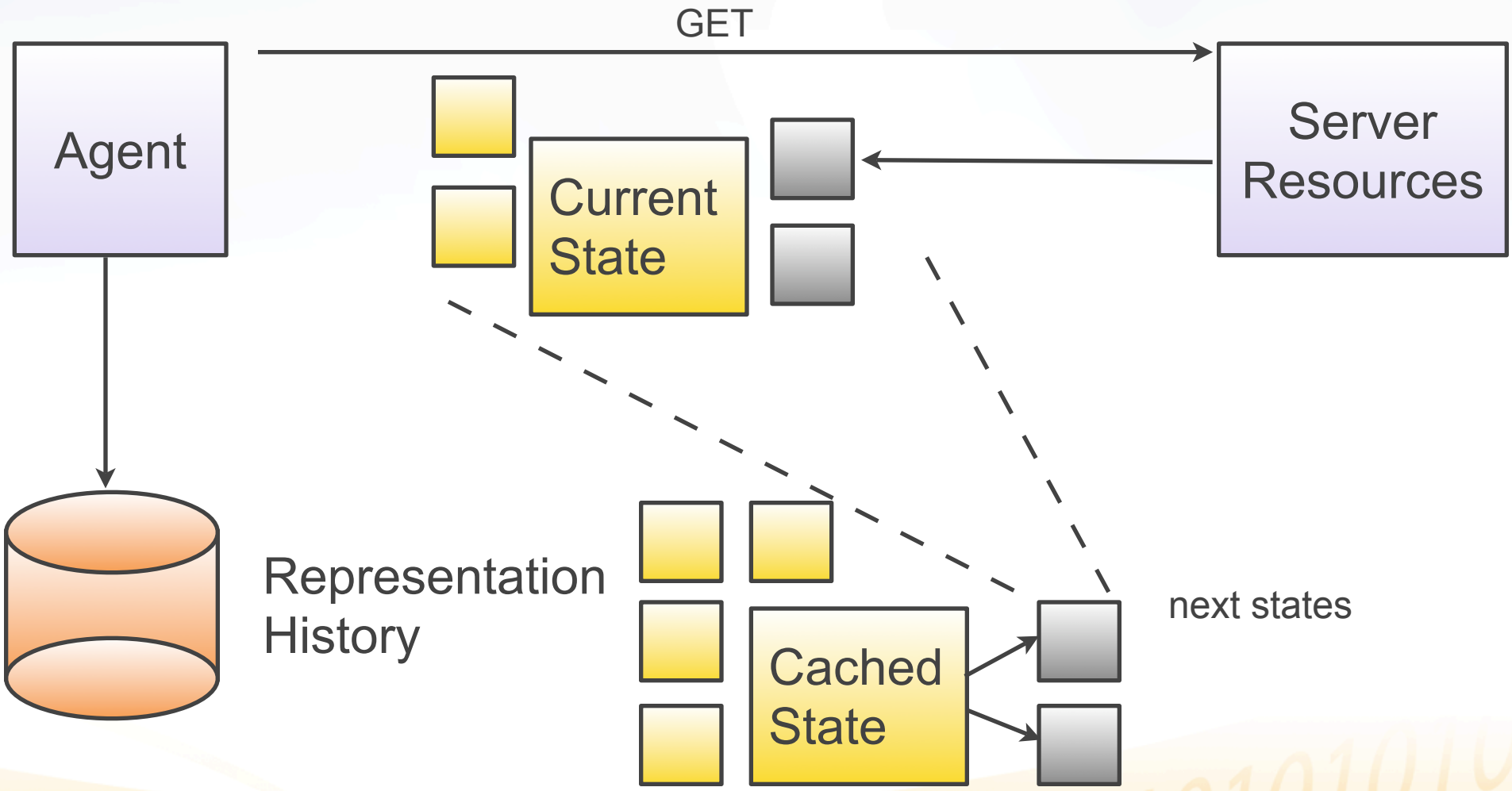
Observing Application State



A RESTFul Amazon EC2

```
<?xml version="1.0" encoding="utf-8"?>
   <feed xmlns="http://www.w3.org/2005/Atom">
   <title type="text">Amazon EC2 Bundles</title>
   <entry>
       <title>Amazon EC2 Standard Small Instance</title>
       <summary>1.7 GB Memory, 1 ECU Core, 160 Gigabytes of Storage</summary>
       <content type="application/edml+xml"
              src="http://www.mycloud.com/bundles/ec2/2008/08/Small"/>
       <updated>2008-07-31T12:29:29Z</updated>
       <published>2007-12-13T08:29:29-04:00</published>
   </entry>
   <entry>
       <title>Amazon EC2 Standard Large Instance</title>
       <summary>8 GB of Memory, 2 ECU Cores, 850 GB of Storage</summary>
       <content type="application/edml+xml"
              src="http://www.mycloud.com/bundles/ec2/2008/08/Large"/>
       <updated>2008-07-31T12:29:29Z</updated>
       <published>2007-12-13T08:29:29-04:00</published>
   </entry>
</feed>
```

Traversing a Link



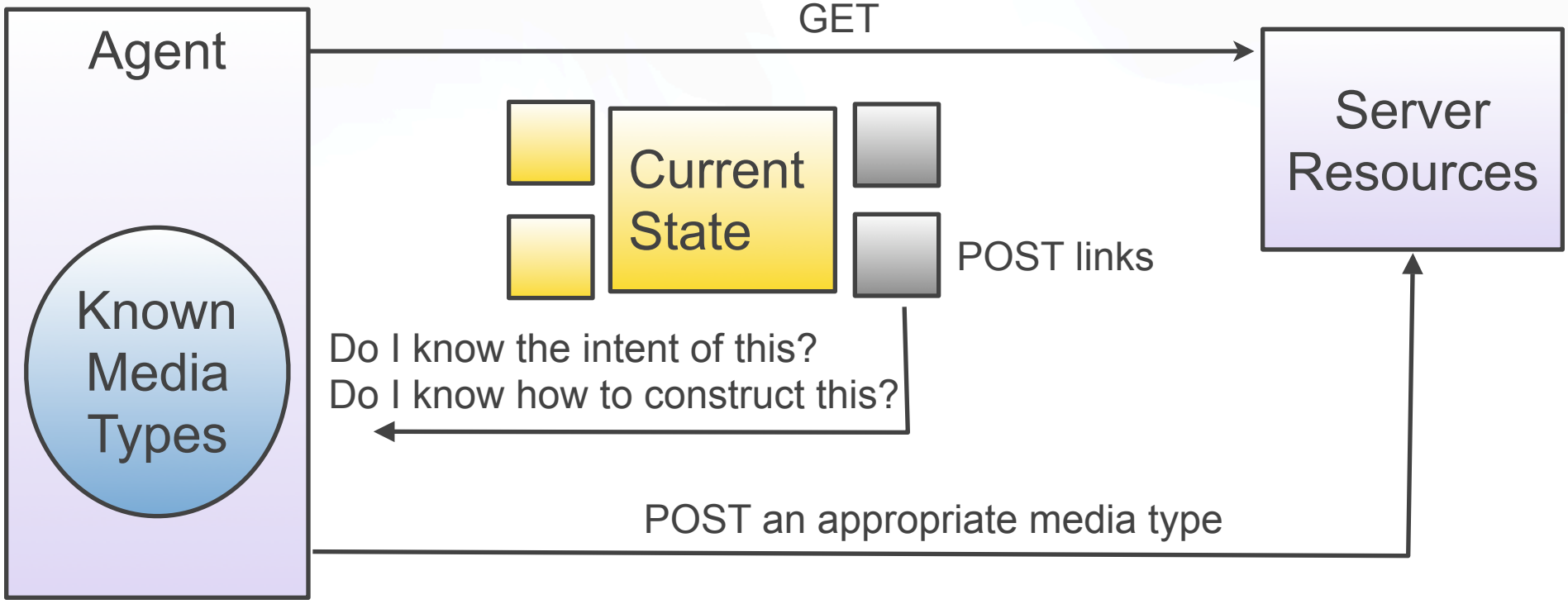
A RESTful Amazon EC2: Templates

```
GET http://www.mycloud.com/bundles/ec2/2008/08/Small
Host: www.mycloud.com
Date: Mon, 17 Nov 2008 09:15:16 PST
Authorization: <token here>
Accept: application/edml+xml

HTTP/1.1 200 OK
Host:  www.mycloud.com
Date: Mon, 17 Nov 2008 09:15:19 PST
Content-Type: application/edml+xml

   <ResourceBundle xmlns="http://exml.com/2008/09/edml">
        <name xml:lang="en">Amazon EC2 Standard Small Instance</rdfs:label>
        <provides>
        <capability href="http://www.mycloud.com/platforms/ec2#SmallStorage" />
        <capability href="http://www.mycloud.com/platforms/ec2#SmallCPU" />
        <capability href="http://www.mycloud.com/platforms/ec2#SmallMemory" />
        </provides>
    </ResourceBundle>
```

A RESTful Amazon EC2: Images

```
<Image xmlns="http://exml.com/2008/09/edml">
     <name>Fedora Core 8 , x86_64, build 3</name>
     <provides>
     <capability>http://purl.org/edml/OperatingSystems/Linux</capability>
     <capability>http://purl.org/edml/OperatingSystems/64Bit</capability>
     </provides>
     <requires>
     <capability>http://purl.org/edml/ChipArchitectures/64Bit</capability
     </requires>
     <run action="">
        <bundle />
        <quantity />
        <groups />
        <keypair />
     </run>
</Image>
```

Changing Resource State



Requesting an Instance

```
POST /images/ec2/FC8x86_64b3
Host: www.mycloud.com
Date: Mon, 17 Nov 2008 09:15:16 PST
Authorization: <token here>
Content-Type: applicaton/edml+xml

<run xmlns="http://exml.com/edmlv2">
   <bundle href="http://www.mycloud.com/bundles/ec2/2008/08/Small"/>
   <quantity>2</quantity>
   <groups>http://www.mycloud.com/groups/default</groups>
   <keypairs>http://www.mycloud.com/keypairs/stuartc-keypair</keypairs>
</run>
```

Receiving an Instance

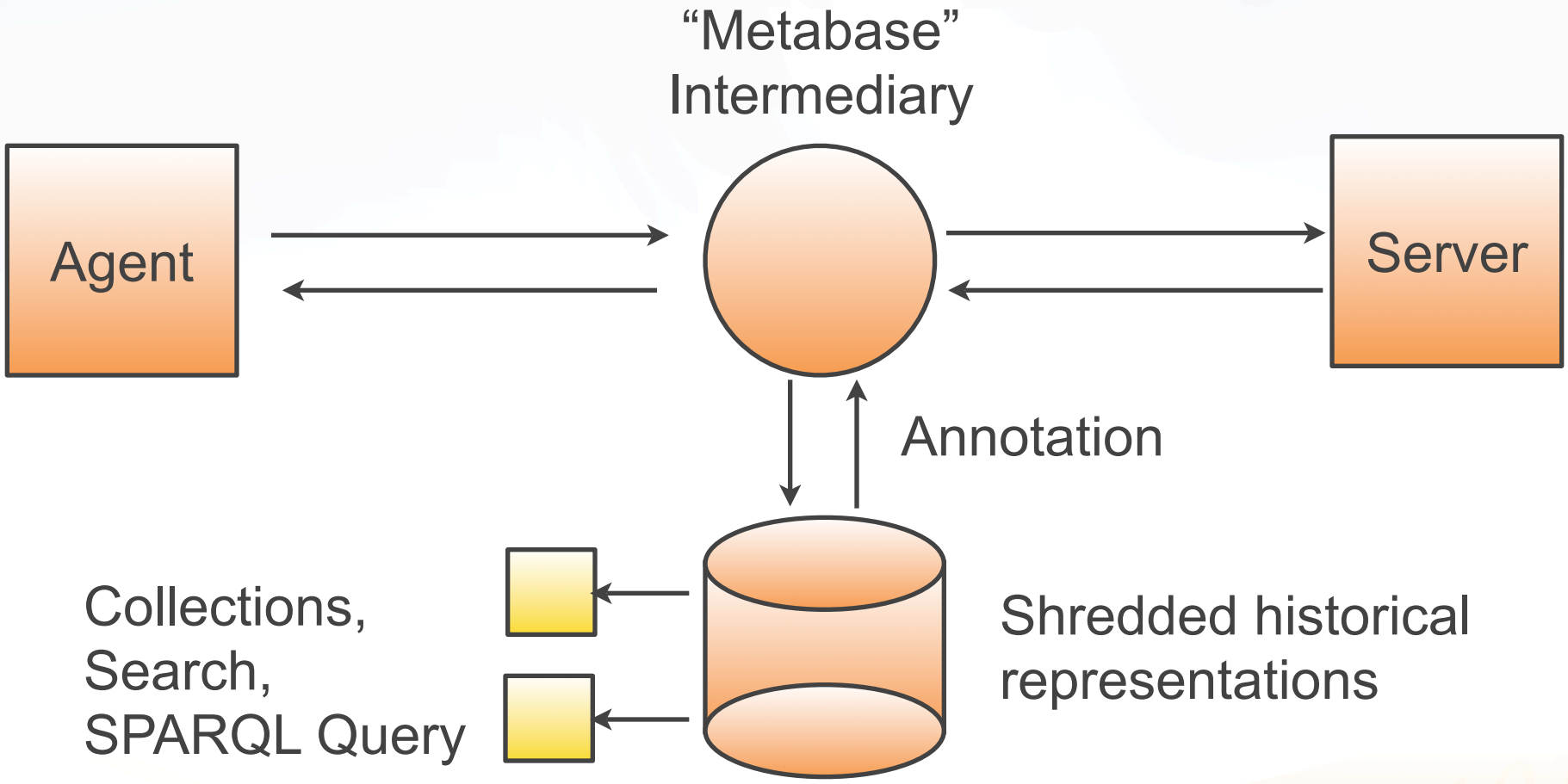
HTTP/1.1 201 Created
Location: /reservations/a9311f3
Host: www.mycloud.com
Date: Mon, 17 Nov 2008 09:15:19 PST
Content-Type: application/edml+xml

```
<Reservation xmlns="http://exml.com/2008/09/edml">
  <terminate action="http://www.mycloud.com/reservations/a9311f3"/>
  <Resource>
    <state>launching</state>
    <link rel="self" href="http://www.mycloud.com/reservations/a9311f3/1"/>
    <bundle href="http://www.mycloud.com/bundles/ec2/2008/08/Small"/>
    <image href="http://www.mycloud.com/bundles/ec2/FC8x86_64b3"/>
    <terminate action="http://www.mycloud.com/reservations/a9311f3/1"/>
  </Resource>
  <Resource>
    <state>launching</state>
    <link rel="self" href="http://www.mycloud.com/reservations/a9311f3/2"/>
    <bundle href="http://www.mycloud.com/bundles/ec2/2008/08/Small"/>
    <image href="http://www.mycloud.com/images/ec2/FC8x86_64b3"/>
    <terminate action="http://www.mycloud.com/reservations/a9311f3/2"/>
  </Resource>
</Reservation>
```

How can I describe my interfaces?

- Tightly Coupled
  - XML Schema Definitions with minOccurs > 0
- Looser Coupled
  - Dynamically generated XML Schema Definitions
  - Edit Link Relations (e.g. AtomPub Media Entries)
  - Forms (e.g. XForms, HTML)
  - Annotate each field with a Persistent URI

What about Versioning and Provenance?



Security: Federated Identity

- Federated Identity is increasingly needed
  - Best bet, treat it as orthogonal for now
    - **SAML**, (very robust in Java world)
    - **WS-Federation**, (for Microsoft integration)
    - **OpenID** (mind the phishing)
    - **Point-to-Point** (sadly)
- OAuth has promise but is very young

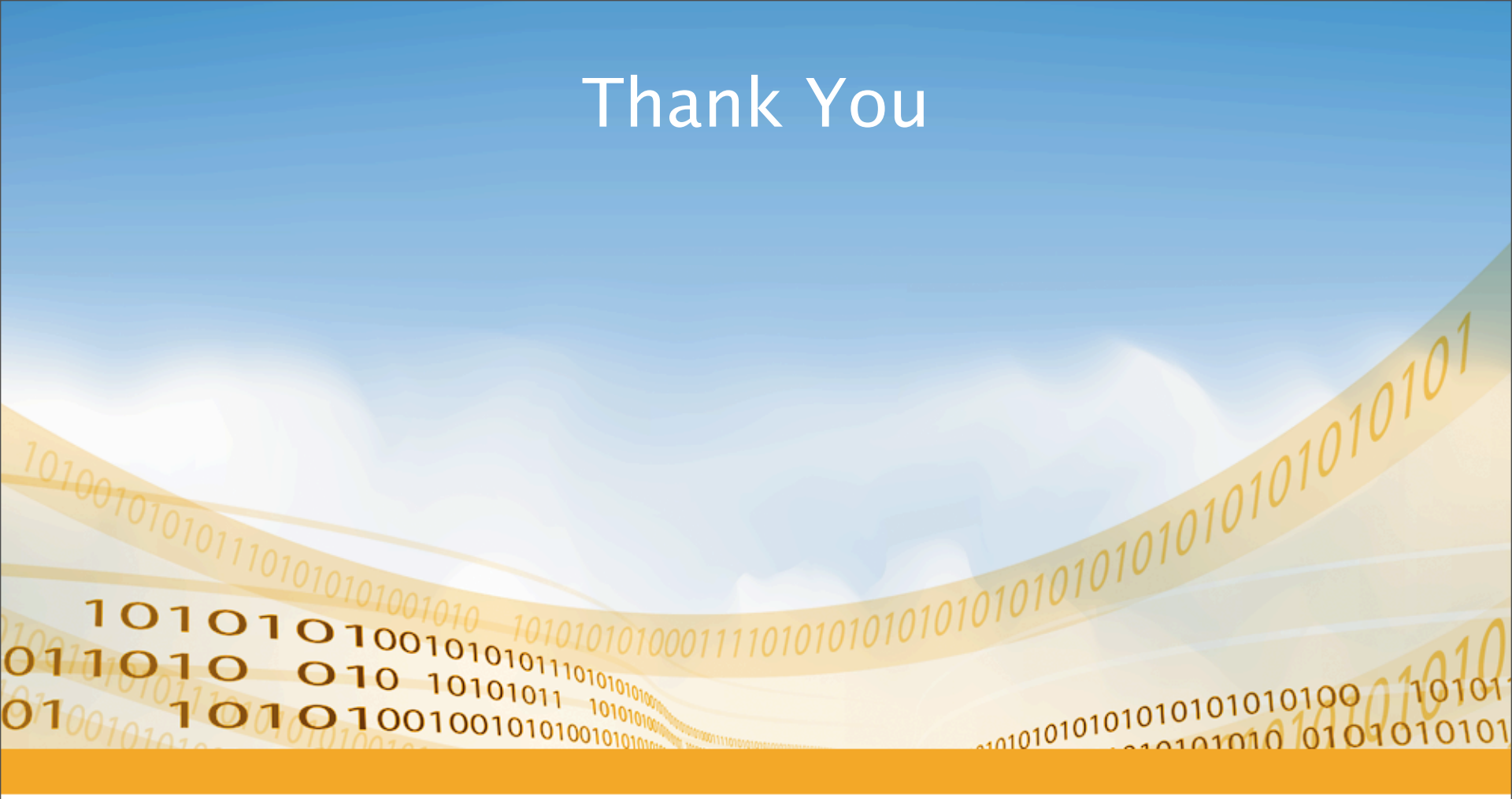
Towards the Semantic Web

- **It's not crazy, it's just…**
  - Layering logic on top of the Web (An Open-World RDBMS)
  - Enabling querying and mashing of web pages without neurosurgery
- **SPARQL** is **very** a big win for RESTful implementers
  - Query **databases** or the **web of hypermedia**
  - **Same syntax – nothing changes**
  - Declarative integrity enforcement for PUT and POST
- **RDFa and GRDDL are easy to use**
  - Just annotate your HTML or write your own XSLT
- **Semantic Web Client Library** – Query the Web
  - http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/

Conclusions

- **Hypermedia bursts traditional IT silos**
  - The same technology can handle **Data, Service, and Presentation**
  - At the expense of efficiency in some cases
- **Hypermedia gives you …**
  - A global database of linked data
  - Human interactions embedded with the data
  - Dynamic Interfaces embedded with the data
    - Dynamic interfaces are like CORBA DII or Reflection with a **ProcessThis**(...) method
    - Need to look at the surrounding context
- Semantic web technology is **practical today** for early adopters
  - Especially GRDDL, SPARQL and RDFa

Thank You



Stuart Charlton
Chief Software Architect, Elastra