

SOA & REST

Ola Angelsmark

What is SOA?

From Wikipedia, the free encyclopedia



This article contains **weasel words**: vague phrasing that often accompanies **biased or unverifiable** information. Such statements should be **clarified or removed**. *(February 2015)*

- Service oriented architecture
- A loosely-coupled architecture designed to meet the business needs of the organization. [1]

SOAP

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://
bencws.foobar.com/doc/2008-01-01/"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/">
  <soap:Header xmlns:foo="http://safe.foobar.com/doc/2007-01-01/" xmlns:oof="http://
www.w3.org/2005/08/addressing">
    <foo:AccessKeyId>0PKRFZMV7GRJ11N791R2</foo:AccessKeyId>
    <foo:Timestamp>2008-03-07T23:55:22.693Z</foo:Timestamp>
    <foo:Signature>someencodedstring</foo:Signature>
    <oof:Action>SomeAction</oof:Action>
    <oof:To>http://bencws.foobar.com</oof:To>
    <oof:MessageID>120493412293</oof:MessageID>
    <oof:ReplyTo>
      <oof:Address> http://www.w3.org/2005/08/addressing/anonymous</oof:Address>
    </oof:ReplyTo>
  </soap:Header>
  <soap:Body>
  ...
```

SOA

- Services are distinct entities
- Loosely coupled with OS and lower layers
- Can be combined and reused
- Allows for
 - Modular programming
 - mashups
 - SaaS
 - Cloud computing

What is REST?

- Architectural style for loosely coupled systems
- ~~Anything HTTP/XML that is not SOAP~~

REST

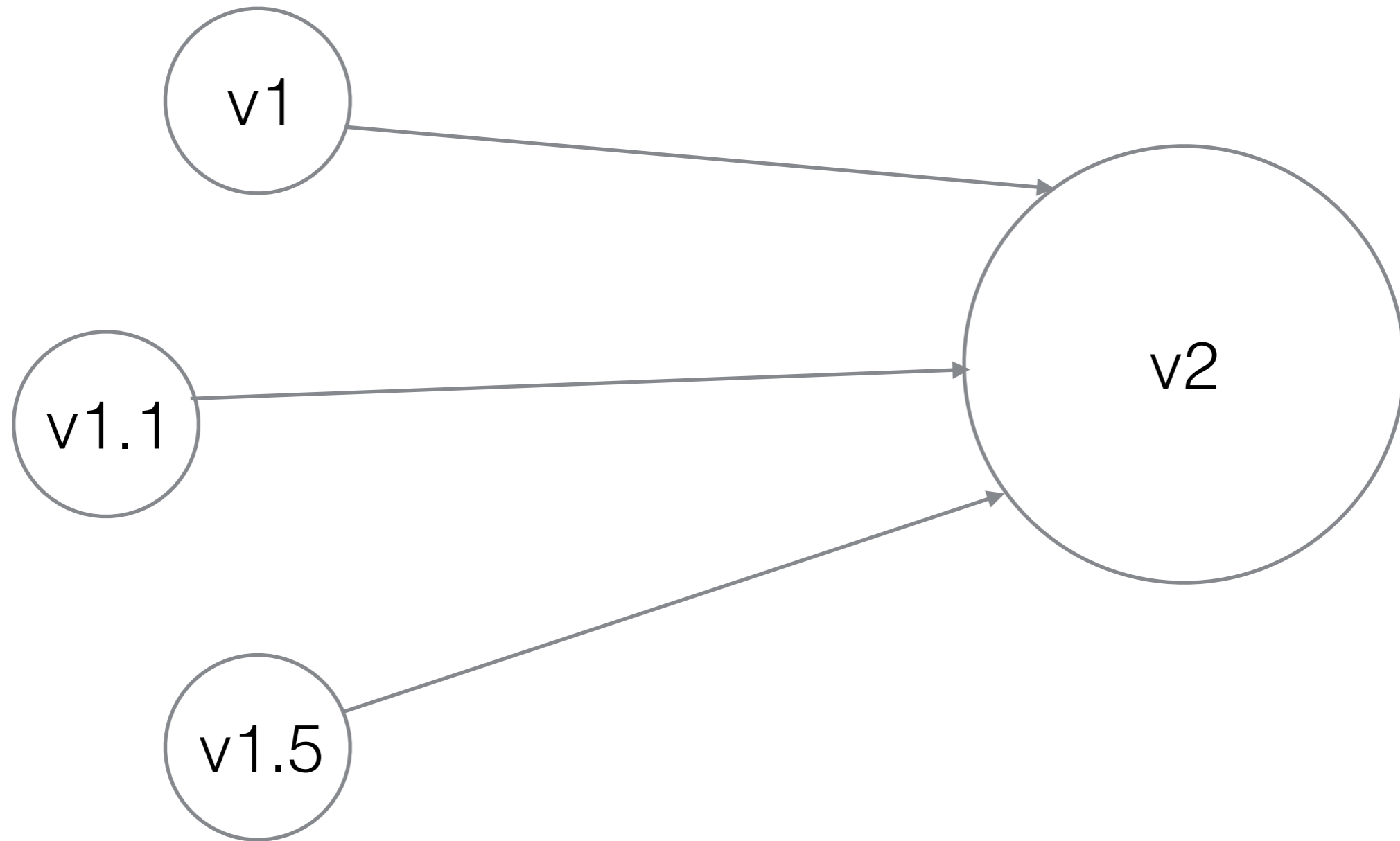


- Roy T. Fielding
 - HTML / URI
 - Apache HTTP Server
 - OpenSolaris
 - Representational State Transfer [2]

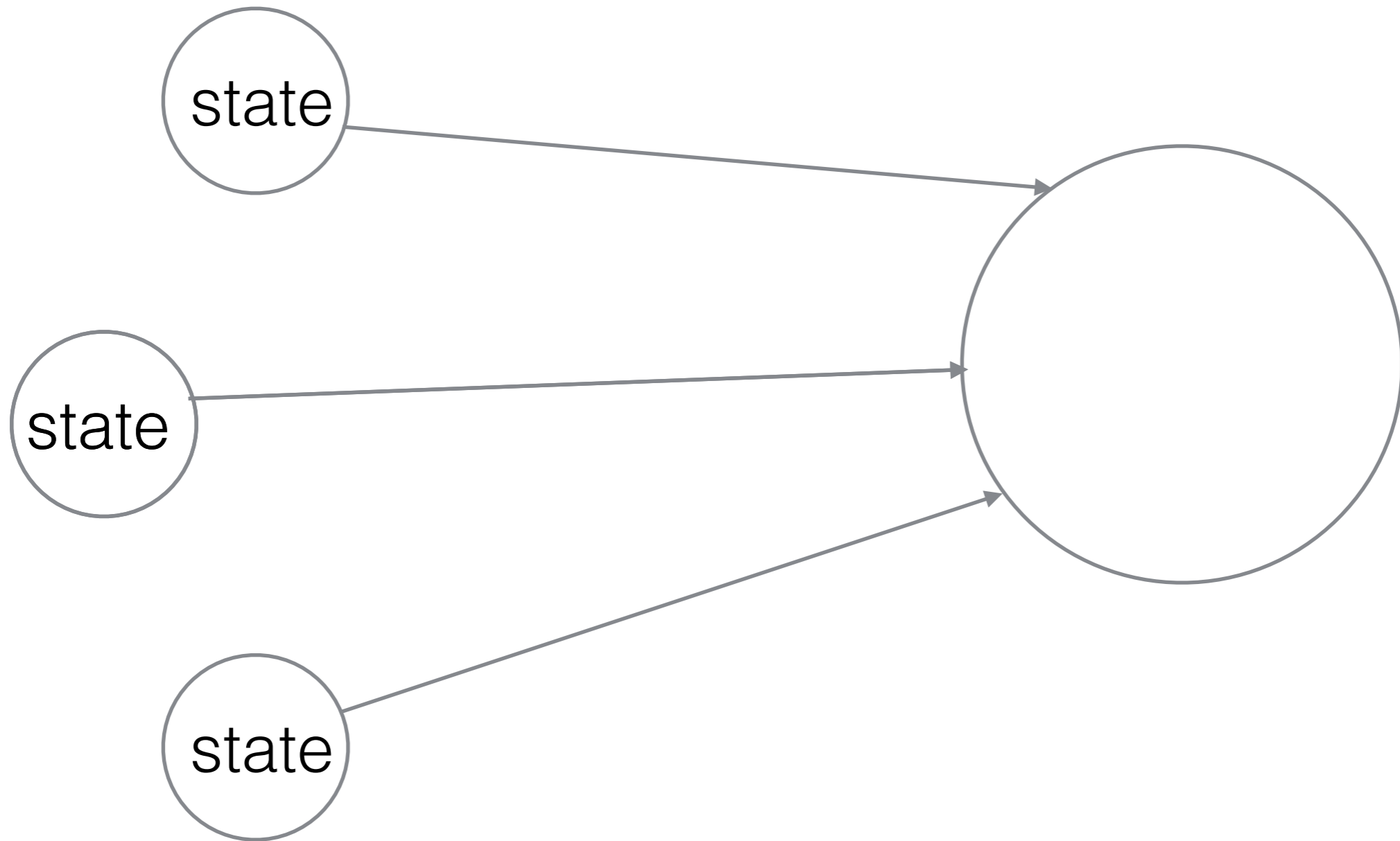
RESTful

Constraint	Pros	Cons
Client-Server	<ul style="list-style-type: none">• Separation of concerns• Scalability• Independent evolution	<ul style="list-style-type: none">• -
Stateless	<ul style="list-style-type: none">• Visibility• Reliability• Scalability	<ul style="list-style-type: none">• Repetitive data• Client dependency
Caching	<ul style="list-style-type: none">• Improved efficiency• Scalability• Performance (perceived)	<ul style="list-style-type: none">• Stale data• Decreased reliability
Uniform interface	<ul style="list-style-type: none">• Simplified architecture• Independent evolution• Decoupled implementation	<ul style="list-style-type: none">• Reduced efficiency
Layered system	<ul style="list-style-type: none">• Encapsulation• Simplified components• Scalability	<ul style="list-style-type: none">• Overhead• Reduced performance (perceived)
Code-on-demand	<ul style="list-style-type: none">• Simplified clients• Improves extensibility	<ul style="list-style-type: none">• Reduced visibility

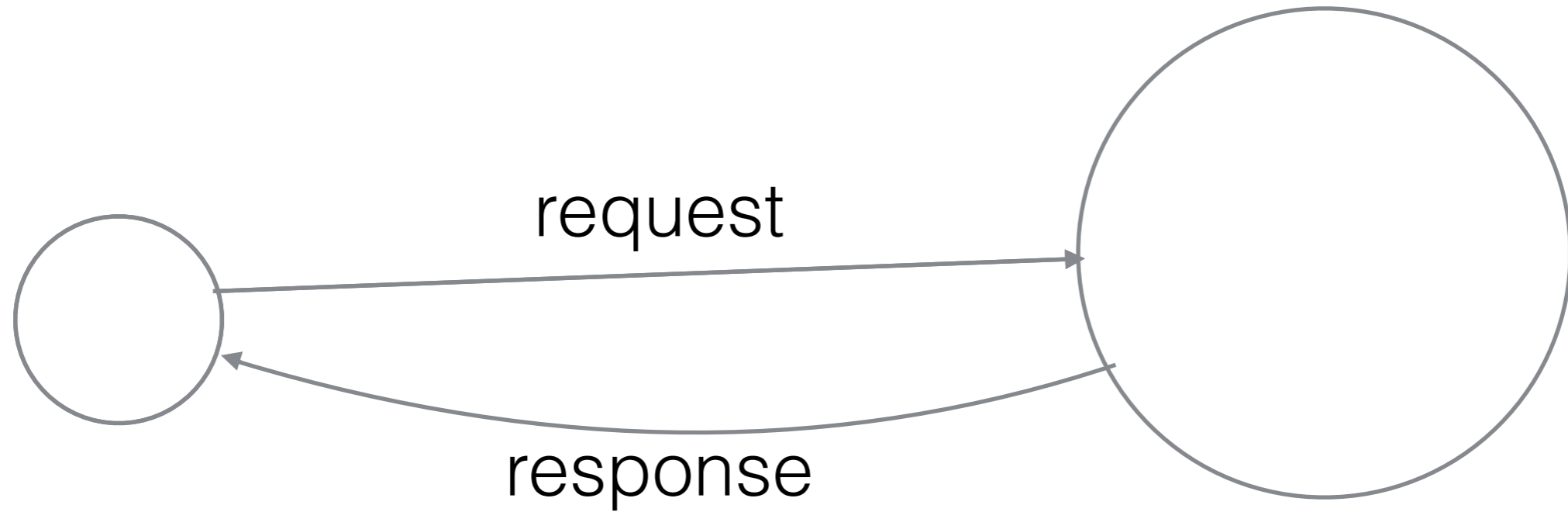
Client-Server



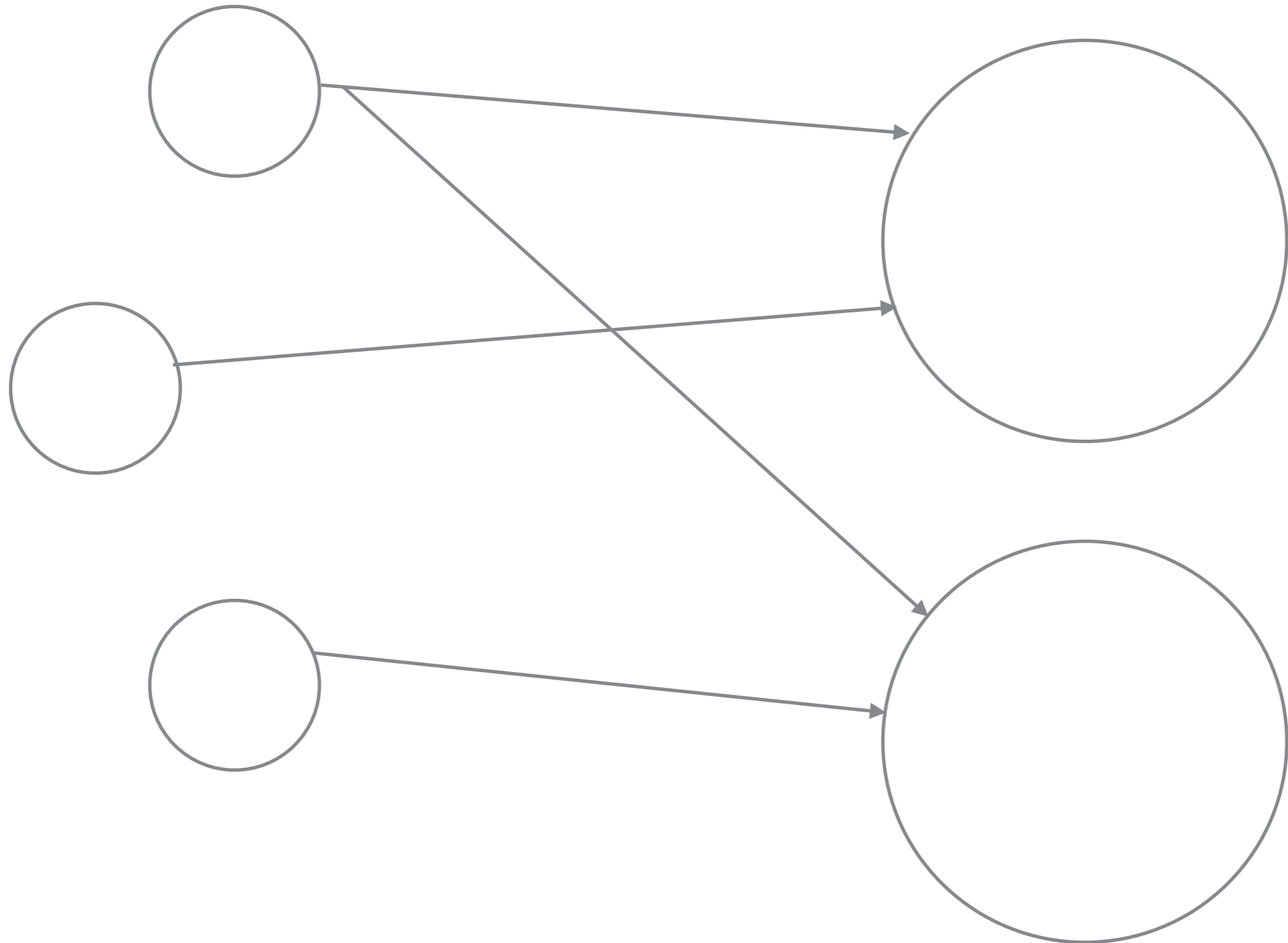
Stateless



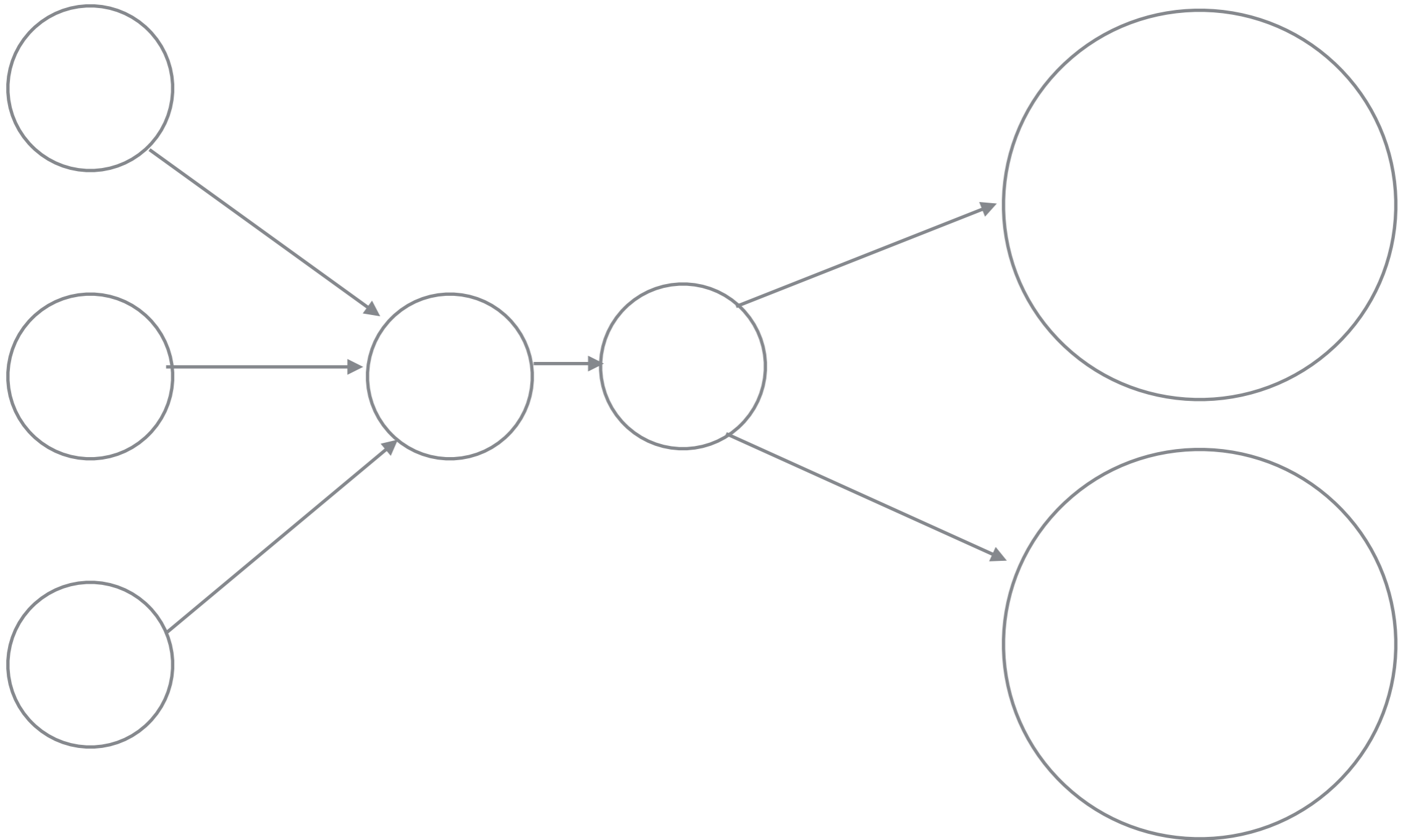
Cache



Uniform interface

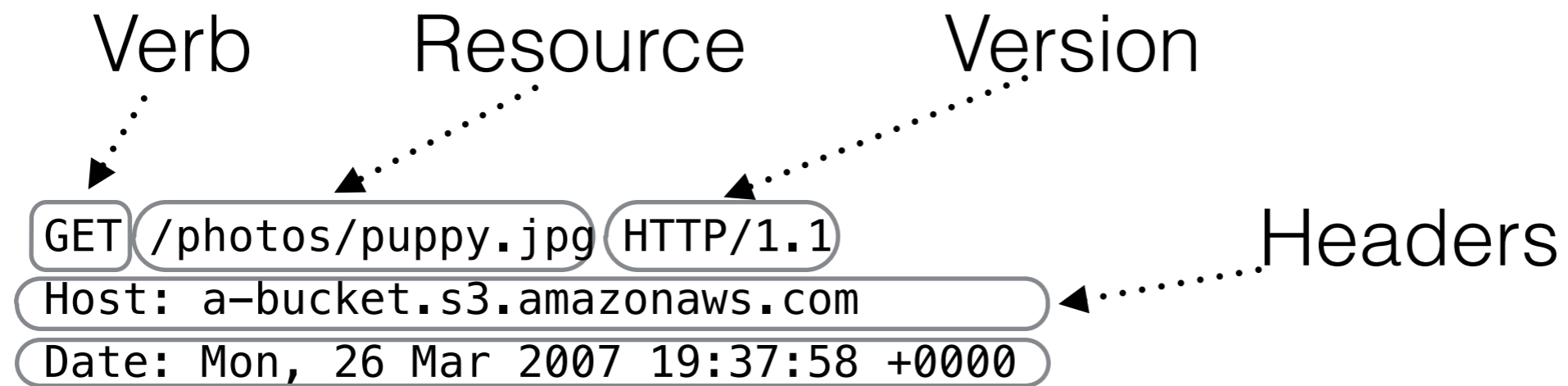


Layered system



Code-on-demand

HTTP



HTTP[6]

Method	Safe	Idempotent	
HEAD	X	X	Retrieve
GET	X	X	Retrieve
PUT		X	Create/Replace
DELETE		X	Remove
POST			Any of the above
...			

HTTP[7]

Code	Type	Example	Meaning
1xx	Information	100	Continue
2xx	Success	201	Created
3xx	Redirect	301	Moved
4xx	Client error	418	I'm a teapot
5xx	Server error	501	Not implemented

REST in practice

- Netflix [3]
 - Excellent API, discontinued
- Flickr [4]
 - REST an alias for HTTP [5]
- Amazon
 - Pragmatic REST



API Description

Create bucket

PUT / HTTP/1.1

Host: BucketName.s3.amazonaws.com

Content-Length: length

Date: date

Authorization: authorization string

```
<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  
  <LocationConstraint>BucketRegion</LocationConstraint>  
</CreateBucketConfiguration>
```



API Description

List items in bucket

```
GET / HTTP/1.1  
Host: BucketName.s3.amazonaws.com  
Date: date  
Authorization: authorization string
```



python vs shell

```
s3handle = boto.connect_s3()  
bucket = s3_handle.get_bucket("mybucket")  
k = boto.s3.key.Key(bucket)  
k.key = "file.txt"  
data = k.get_contents_as_string()  
print data
```

```
$ curl "http://mybucket.s3.amazonaws.com/file.txt"
```

REST Authorization

- No session or login
- Often home-cooked

```
Authorization = "AWS" + " " + AWSAccessKeyId + ":" + Signature
```

```
Signature = Base64( HMAC-SHA1( SecretAccessKey, StringToSign ) ) )
```

```
StringToSign = HTTP-Verb + "\n" +  
Content-MD5 + "\n" +  
Content-Type + "\n" +  
Date + "\n" +  
Amazon Headers +  
Resource
```



Amazon authorization

GET / HTTP/1.1
Host: my-bucket.s3.amazonaws.com
Date: Wed, 22 Apr 2015 05:57:59 +0000

GET\n\n\nWed, 22 Apr 2015 05:57:59 +0000\n/my-bucket/

SecretAccessKey

Access Key Id

Authorization: AWS AKIAIXCJIZFRW504N0SA:AvFJJcFWwyjL5A1Yo420P8b4VaA=



Amazon authorization

```
GET / HTTP/1.1
Host: my-bucket.s3.amazonaws.com
Date: Wed, 22 Apr 2015 05:57:59 +0000
Authorization: AWS AKIAIXCJIZFRW504NOSA:AvFJJcFWwyjL5AlYo420P8b4VaA=
```

```
<Prefix/>
<Marker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>>false</IsTruncated>
<Contents>
  <Key>file.txt</Key>
  <LastModified>2015-04-21T07:32:05.000Z</LastModified>
  <ETag>"1401d2cc11ef4e21380f3289068b3d46"</ETag>
  <Size>31021</Size>
  <Owner>
    <ID>c6d7aafc4ecfae7f5a8f614eaafcebc46787278be416e3ea258c11838ce77e47</ID>
    <DisplayName>owner</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
</Contents>
</ListBucketResult>
```



Short demonstration

- HTTP, headers, authorisation
- Check for bucket existence
 - forbidden / not found / ok
- List contents of bucket
- Fetch item from bucket
- Delete item from bucket

Bibliopgrahy

- [1] MSDN. *SOA in the Real World*
- [2] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*, PhD Dissertation
- [3] <http://apievangelist.com/2013/03/12/netflix-api-is-much-more-than-a-public-api/>
- [4] <https://www.flickr.com/services/api>
- [5] <http://roy.gbiv.com/untangled/2008/no-rest-in-cmis>
- [6] <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>
- [7] <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10>
- [8] <http://www.infoq.com/articles/webber-rest-workflow>