Using REST for SOA

Stefan Tilkov, QCon SF 2010

Stefan Tilkov
stefan.tilkov@innoq.com
http://www.innoq.com/blog/st/
@stilkov

innoQ Deutschland GmbH
Halskestraße 17
D-40880 Ratingen
Phone +49 21 02 77 162-100

innoQ Schweiz GmbH
Gewerbestrasse 11
CH-6330 Cham
Phone +41 41 743 01 11

info@innoq.com · www.innoq.com

REST

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
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
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
GET /customers/1234
Host: example.com
Accept: application/vnd.mycompany.customer+xml
```

```
<customer>...</customer>
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
GET /customers/1234
Host: example.com
Accept: application/vnd.mycompany.customer+xml
```

```
<customer>...</customer>
```

```
GET /customers/1234
Host: example.com
Accept: text/x-vcard
```

```
begin:vcard
...
end:vcard
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
<order self='http://example.com/orders/3321'>
  <item>
    <amount>23</amount>
    <product ref='http://example.com/products/4554' />
  </item>
  <customer ref='http://example.com/customers/1234' />
  <link rel='items'
        ref='http://example.com/orders/3321/items' />
</order>
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
<order self='http://example.org/0E6C2BC1094C'>
  <item>
    <amount>23</amount>
    <product ref='http://amazon.com/products/A31138B3' />
  </item>
  <customer ref='http://example.net/4E8F-891D' />
  <link rel='items'
        ref='http://example.com/EFDBE4A38931' />
</order>
```

The REST Uniform Interface

identification of resources

self-descriptive messages

```
<order self='ht                                                    '>
  <item>
    <amount>23<
    <product re                                                /A31138B3' />
  </item>
  <customer ref                                                D' />
  <link rel='it
        ref='ht                                                ' />
</order>
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200 OK
Date:   Sun, 04 Oct 2009 19:36:25 GMT
Server:         Apache/2.2.11 (Debian)
Last-Modified:Fri, 02 Oct 2009 16:47:31 GMT
Etag:   "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:          bytes
Vary:   Accept-Encoding
Content-Encoding:   gzip
Content-Length:     7160
Keep-Alive:     timeout=15, max=91
Connection:   Keep-Alive
Content-Type:           application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

Standard Method

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:      XYZ 1.1
Accept:          text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:      300
Connection:      keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200 OK
Date:  Sun, 04 Oct 2009 19:36:25 GMT
Server:          Apache/2.2.11 (Debian)
Last-Modified:Fri, 02 Oct 2009 16:47:31 GMT
Etag:   "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:          bytes
Vary:   Accept-Encoding
Content-Encoding:    gzip
Content-Length:      7160
Keep-Alive:     timeout=15, max=91
Connection:   Keep-Alive
Content-Type:          application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

Standard Method

Media Type

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200 OK
Date:   Sun, 04 Oct 2009 19:36:25 GMT
Server:         Apache/2.2.11 (Debian)
Last-Modified:Fri, 02 Oct 2009 16:47:31 GMT
Etag:   "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:          bytes
Vary:   Accept-Encoding
Content-Encoding:   gzip
Content-Length:         7160
Keep-Alive:     timeout=15, max=91
Connection:   Keep-Alive
Content-Type:           application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

Standard Method

Media Type

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200 OK
Date:   Sun, 04 Oct 2009 19:36:25 GMT
Server:         Apache/2.2.11 (Debian)
Last-Modified:Fri, 02 Oct 2009 16:47:31 GMT
Etag:   "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:          bytes
Vary:   Accept-Encoding
Content-Encoding:   gzip
Content-Length:         7160
Keep-Alive:     timeout=15, max=91
Connection:   Keep-Alive
Content-Type:           application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

Data

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

Standard Method

Media Type

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200 OK
Date:   Sun, 04 Oct 2009 19:36:25 GMT
Server:         Apache/2.2.11 (Debian)
Last-Modified:Fri, 02 Oct 2009 16:47:31 GMT
Etag:   "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:          bytes
Vary:   Accept-Encoding
Content-Encoding:   gzip
Content-Length:         7160
Keep-Alive:     timeout=15, max=91
Connection:   Keep-Alive
Content-Type:           application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

Control Data

Data

The REST Uniform Interface

identification of resources

resource manipulation through representations

hypermedia as the engine of application state

self-descriptive messages

Standard Method

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent:     XYZ 1.1
Accept:         text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive:     300
Connection:     keep-alive
If-Modified-Since:      Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60

HTTP/1.1 200
Date:  Sun, 04 Oct 2009 19:16:25 GMT
Server:  Apache/2.2.11 (Debian)
Last-Modified: Fri, 02 Oct 2009 16:47:31 GMT
Etag:  "600028c-59fb-474f6852c9dab"
Cache-Control:  max-age=300
Accept-Ranges:  bytes
Vary:  Accept-Encoding
Content-Encoding:  gzip
Content-Length:  7160
Keep-Alive:  timeout=15, max=91
Connection:  Keep-Alive
Content-Type:  application/xml

<?xml version='1.0' encoding='utf-8' ?>
...
```

Visibility

Media Type

Control Data

Data

getOrderDetails()

submitApplicationData()

updateQuote()

findMatchingBid()

initiateProcess()

cancelSubscription()

listAuctions()

getUsers()

getOrderDetails()

findMatchingBid()

listAuctions()

getUsers()

initiateProcess()

submitApplicationData()

updateQuote()

cancelSubscription()

getOrderDetails()

findMatchingBid()

listAuctions()

getUsers()

initiateProcess()

submitApplicationData()

updateQuote()

cancelSubscription()

getOrderDetails()

findMatchingBid()

listAuctions()

getUsers()

GET

initiateProcess()

submitApplicationData()

POST

PUT

updateQuote()

DELETE

cancelSubscription()

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

Anything that knows
your app

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

Anything that knows
your app

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

generic

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

Anything that knows
your app

```
interface Resource {
      Resource(URI u)
      Response get()
      Response post(Request r)
      Response put(Request r)
      Response delete()
}
```

```
class CustomerCollection : Resource {
      ...
      Response post(Request r) {
            id = createCustomer(r)
            return new Response(201, r)
      }
      ...
}
```

generic

Any HTTP client
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

Anything that knows
your app

specific

Mapping Examples

Mapping Examples

getFreeTimeSlots(Person)

Mapping Examples

getFreeTimeSlots(Person) → GET /people/{id}/timeslots?state=free

Mapping Examples

getFreeTimeSlots(Person) → GET /people/{id}/timeslots?state=free

rejectApplication(Application)

Mapping Examples

getFreeTimeSlots(Person) → GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) → POST /rejections↵
<application>http://...</application>↵
<reason>Unsuitable for us!</reason>

Mapping Examples

getFreeTimeSlots(Person) → GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) → POST /rejections↵
<application>http://...</application>↵
<reason>Unsuitable for us!</reason>

---

performTariffCalculation(Data)

Mapping Examples

getFreeTimeSlots(Person) →GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) →POST /rejections↵
<application>http://...</application>↵
<reason>Unsuitable for us!</reason>

---

performTariffCalculation(Data) →POST /contracts↵
Data
←Location: http://.../contracts/4711
→GET /contracts/4711/tariff
←Result

Mapping Examples

getFreeTimeSlots(Person) →GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) →POST /rejections↵
<application>http://...</application>↵
<reason>Unsuitable for us!</reason>

---

performTariffCalculation(Data) →POST /contracts↵
Data
←Location: http://.../contracts/4711
→GET /contracts/4711/tariff
←Result

---

shipOrder(ID)

Mapping Examples

getFreeTimeSlots(Person)	→GET /people/{id}/timeslots?state=free
rejectApplication(Application)	→POST /rejections↵ <application>http://...</application>↵ <reason>Unsuitable for us!</reason>
performTariffCalculation(Data)	→POST /contracts↵ Data ←Location: http://.../contracts/4711 →GET /contracts/4711/tariff ←Result
shipOrder(ID)	→PUT /orders/0815/status↵ <status>shipped</status>

Mapping Examples

getFreeTimeSlots(Person) →GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) →POST /rejections↵
<application>http://...</application>↵
<reason>Unsuitable for us!</reason>

---

performTariffCalculation(Data) →POST /contracts↵
Data
←Location: http://.../contracts/4711
→GET /contracts/4711/tariff
←Result

---

shipOrder(ID) →PUT /orders/0815/status↵
<status>shipped</status>

---

shipOrder(ID) [variation]

Mapping Examples

getFreeTimeSlots(Person)	→GET /people/{id}/timeslots?state=free
rejectApplication(Application)	→POST /rejections↵ <application>http://...</application>↵ <reason>Unsuitable for us!</reason>
performTariffCalculation(Data)	→POST /contracts↵ Data ←Location: http://.../contracts/4711 →GET /contracts/4711/tariff ←Result
shipOrder(ID)	→PUT /orders/0815/status↵ <status>shipped</status>
shipOrder(ID) [variation]	→POST /shipments↵ Data ←Location: http://.../shipments/4711

SOA

Well-known architectural principles, applied at different scale

http://soa-manifesto.org/

Through our work we have come to prioritize:

Business value over technical strategy
Strategic goals over project-specific benefits
Intrinsic interoperability over custom integration
Shared services over specific-purpose implementations
Flexibility over optimization
Evolutionary refinement over pursuit of initial perfection

That is, while we value the items on the right, we value the items on the left more.

twitter

Home Profile Find People Settings Help Sign out

That "SOA Manifesto" is vacuous to the point of being insulting. "Intrinsic Interoperability" my ass.

5:08 AM Oct 24th, 2009 via twidroid

Reply Retweet

timbray

Tim Bray

"Unlike the Agile manifesto however, the SOA manifesto is nothing more than snake oil to give the vendor community a thin veneer of respectability atop their increasing bureaucracy, deteriorating levels of innovation, and increasingly painful pricing models."

– Jim Webber

http://jim.webber.name/2009/10/24/95bf2681-9a7a-4f94-94d6-2156a3a46411.aspx

Lesson Learned:

Don’t write manifestos …

… unless you look like this

… or this

Bad Stuff

Good Stuff

Monoliths

Monoliths

Redundant data

Monoliths

Redundant data

Redundant logic

Monoliths

Redundant data

Redundant logic

Resistance to change

- Monoliths
- Redundant data
- Redundant logic
- Resistance to change
- Late integration

Monoliths

Redundant data

Redundant logic

Resistance to change

Late integration

Vendor dependency

Monoliths

Redundant data

Redundant logic

Resistance to change

Late integration

Vendor dependency

Bad

Monoliths

Redundant data

Redundant logic

Resistance to change

Late integration

Vendor dependency

Bad

Loosely coupled modules

Monoliths

Redundant data

Redundant logic

Resistance to change

Late integration

Vendor dependency

Bad

Loosely coupled modules

Localized data

Monoliths	Loosely coupled modules
Redundant data	Localized data
Redundant logic	Single implementation
Resistance to change	
Late integration	
Vendor dependency	

Bad

Monoliths	Loosely coupled modules
Redundant data	Localized data
Redundant logic	Single implementation
Resistance to change	Flexibility
Late integration	
Vendor dependency	

Bad

Monoliths	Loosely coupled modules
Redundant data	Localized data
Redundant logic	Single implementation
Resistance to change	Flexibility
Late integration	Early interoperability
Vendor dependency	

Bad

Monoliths	Loosely coupled modules
Redundant data	Localized data
Redundant logic	Single implementation
Resistance to change	Flexibility
Late integration	Early interoperability
Vendor dependency	Reliance on standards

Bad

Monoliths	Loosely coupled modules
Redundant data	Localized data
Redundant logic	Single implementation
Resistance to change	Flexibility
Late integration	Early interoperability
Vendor dependency	Reliance on standards
Bad	**Good**

RESTful SOA?

The Web & SOA

The Web & SOA

Standard protocols	

The Web & SOA

Standard protocols	
Standard formats	

The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	

The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	
Mature and useful intermediaries	

The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	
Mature and useful intermediaries	
Support for loose coupling	

The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	
Mature and useful intermediaries	
Support for loose coupling	
Wide availability and adoption	

The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	
Mature and useful intermediaries	
Support for loose coupling	
Wide availability and adoption	
Well-defined architectural model	

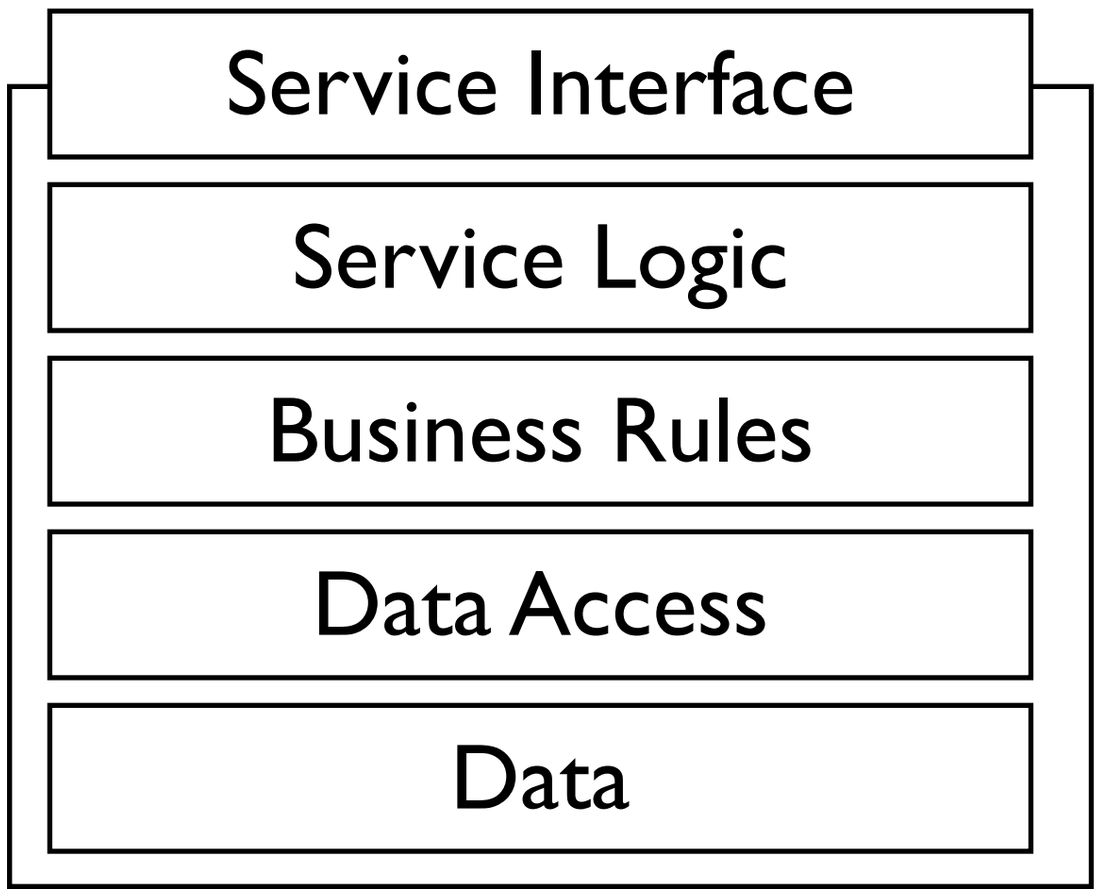
The Web & SOA

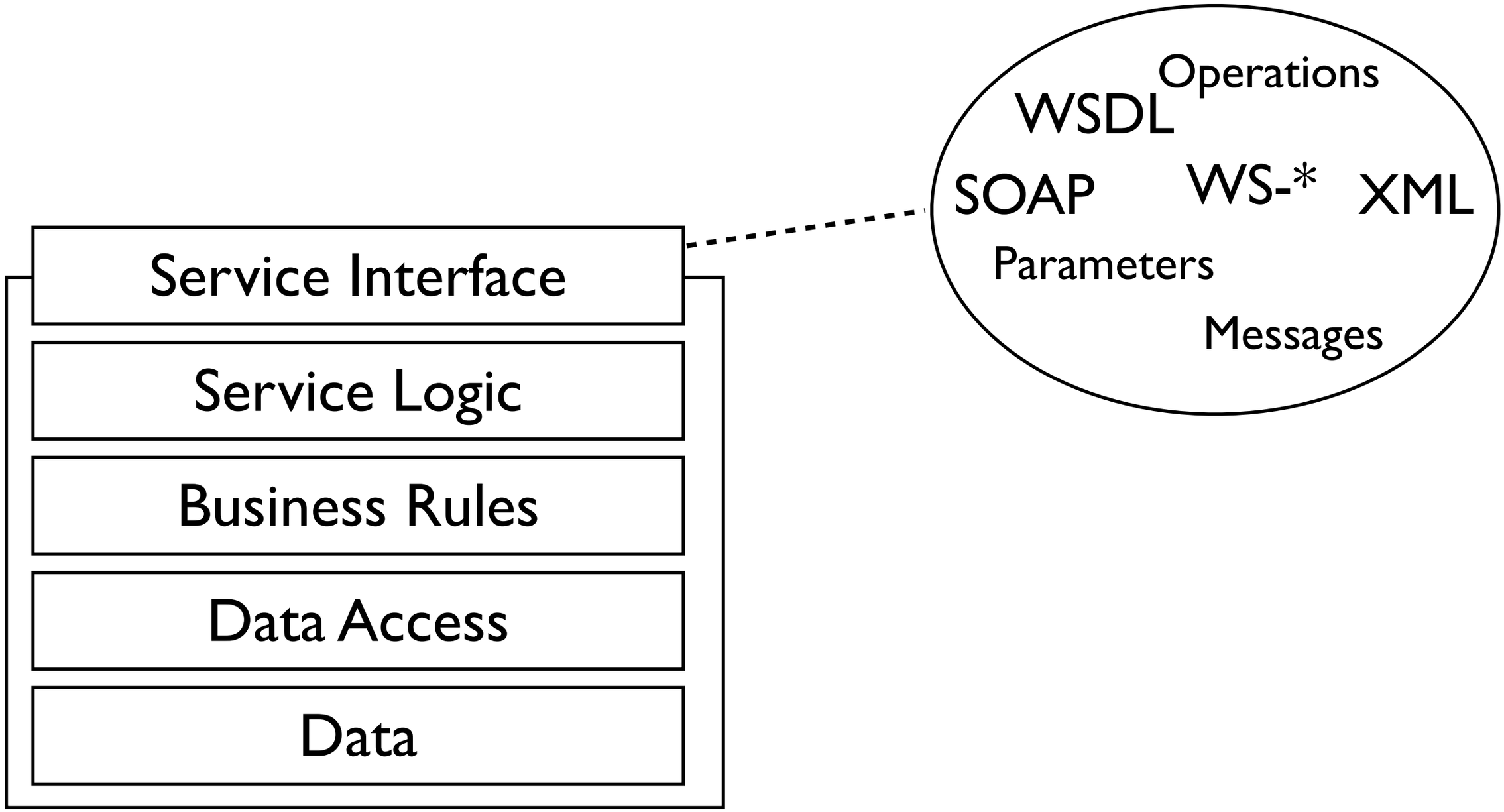
Standard protocols	✔
Standard formats	✔
Library and tool support	✔
Mature and useful intermediaries	✔
Support for loose coupling	✔
Wide availability and adoption	✔
Well-defined architectural model	✔

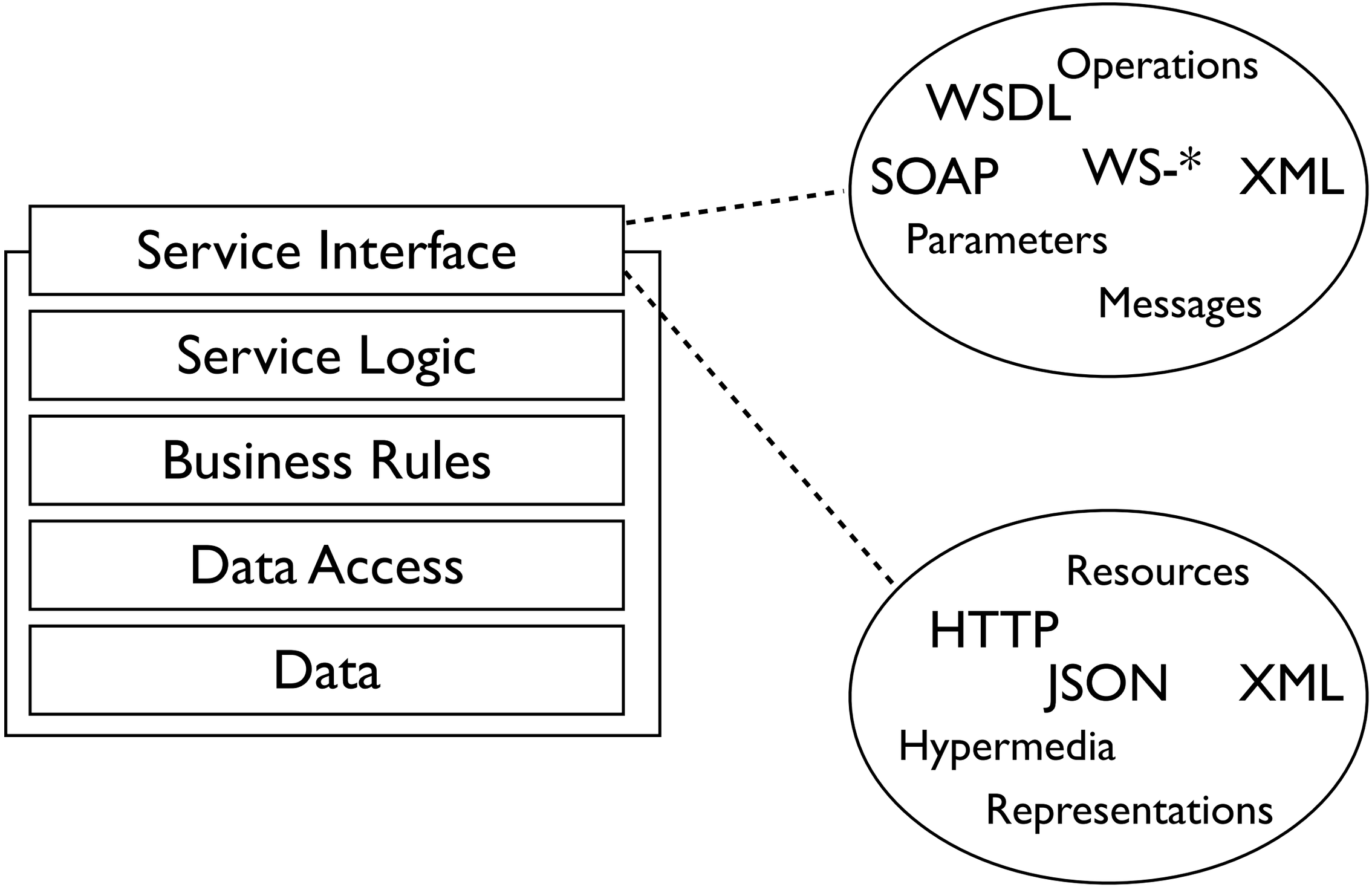
But what about ...

“Enterprisey” Stuff?

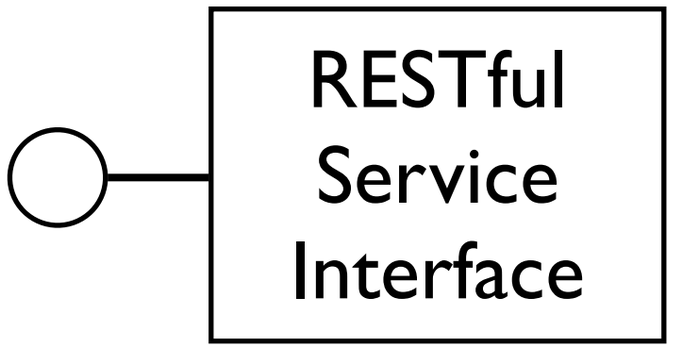
Encapsulation

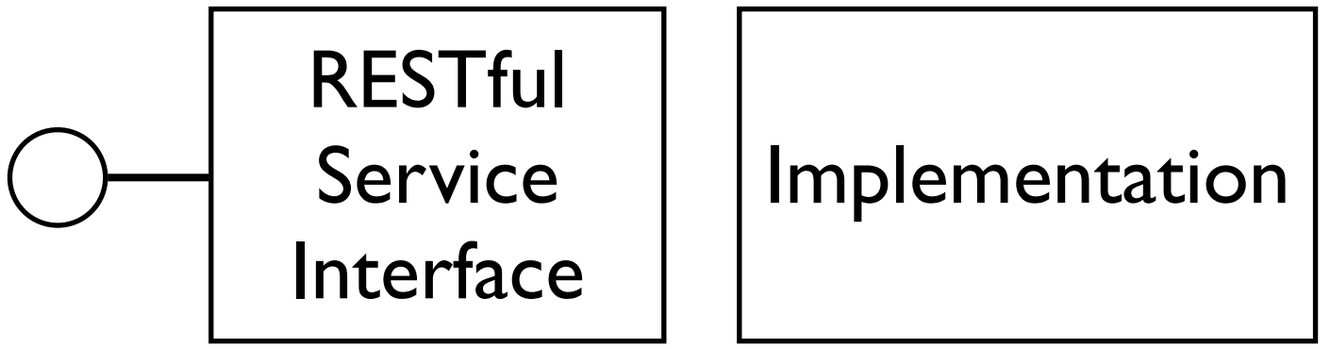


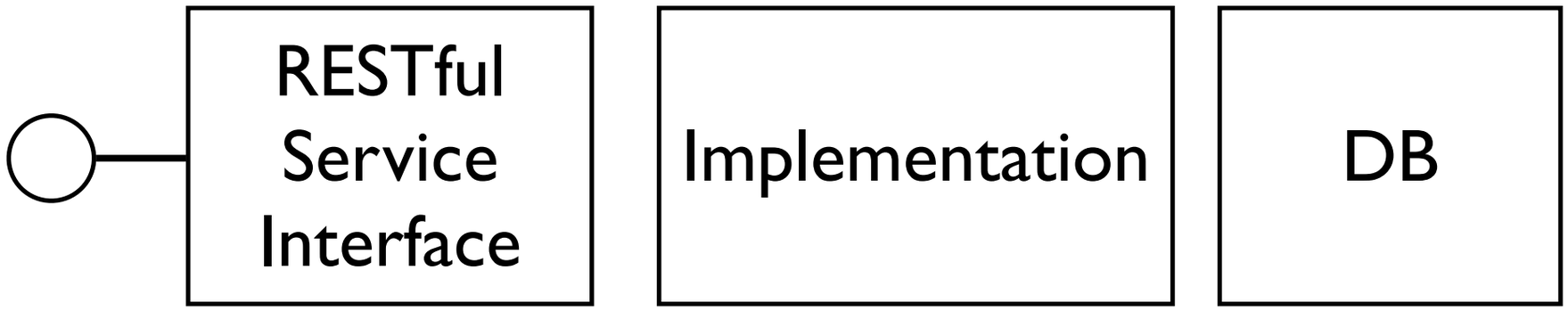


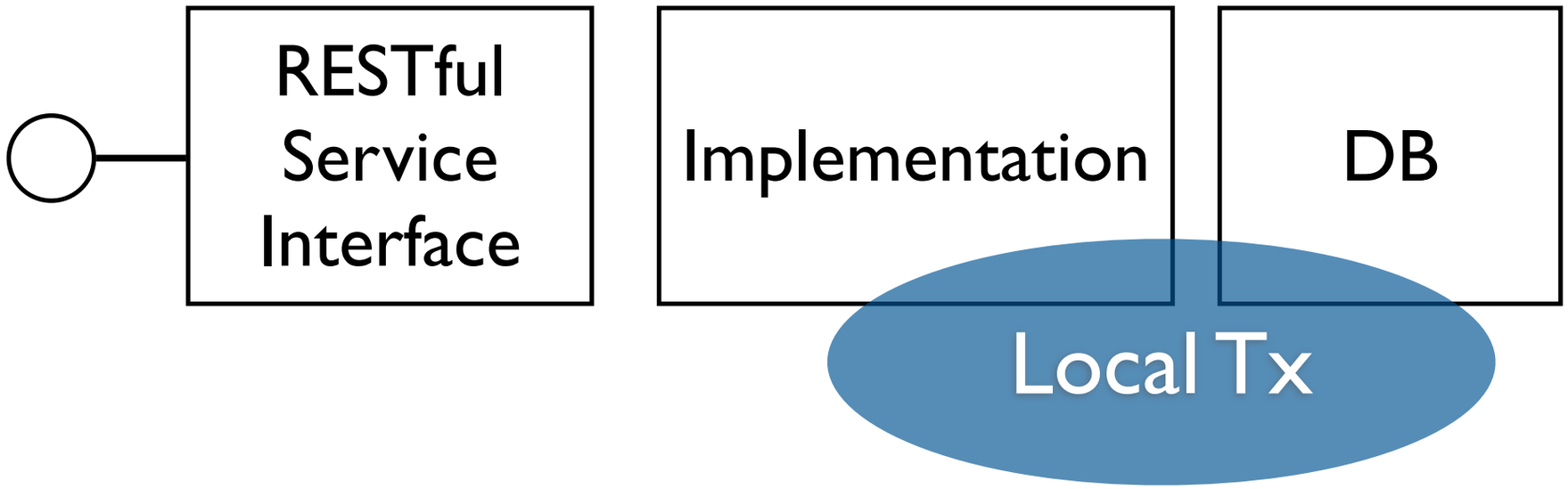


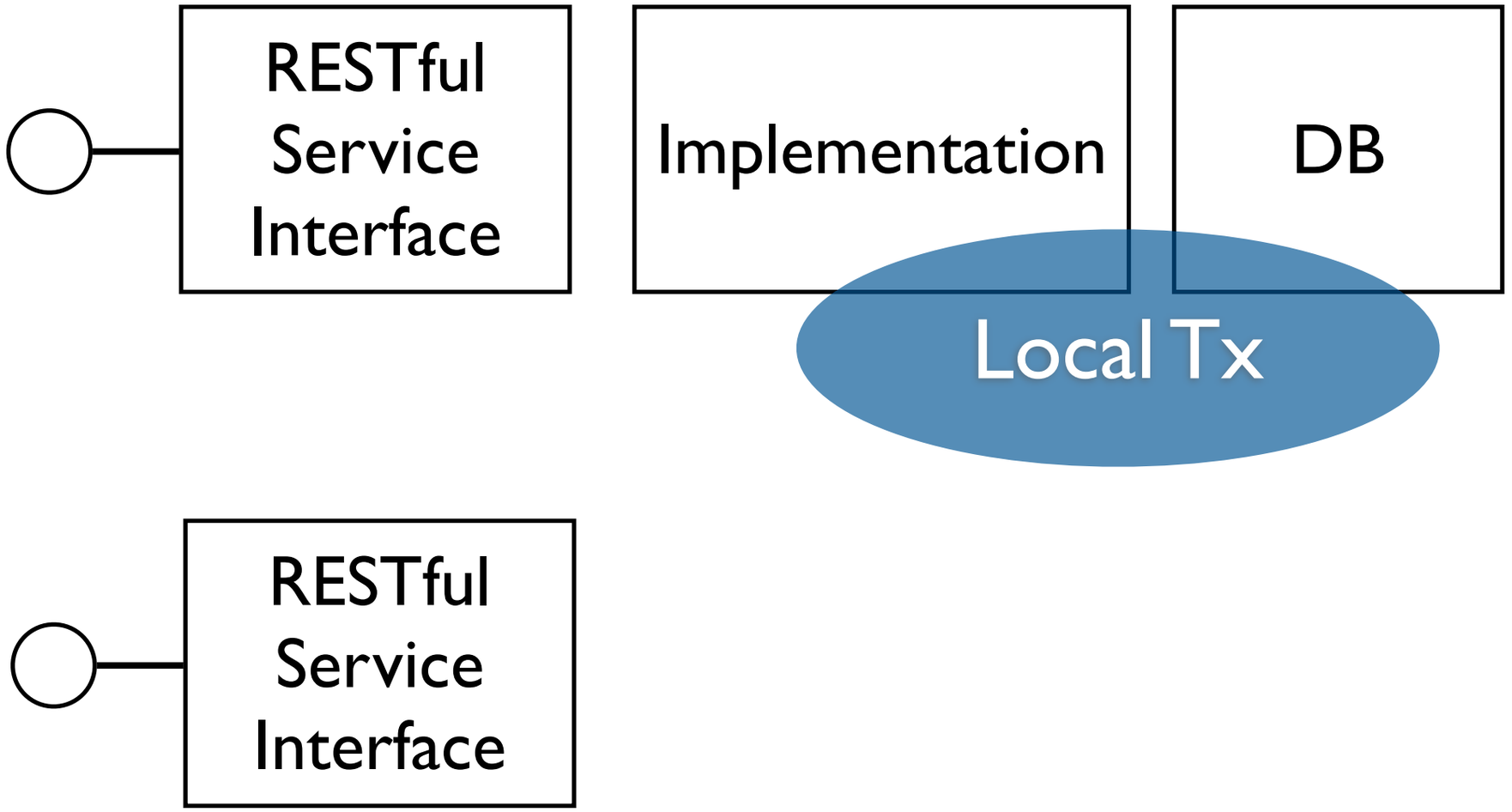
Transactions

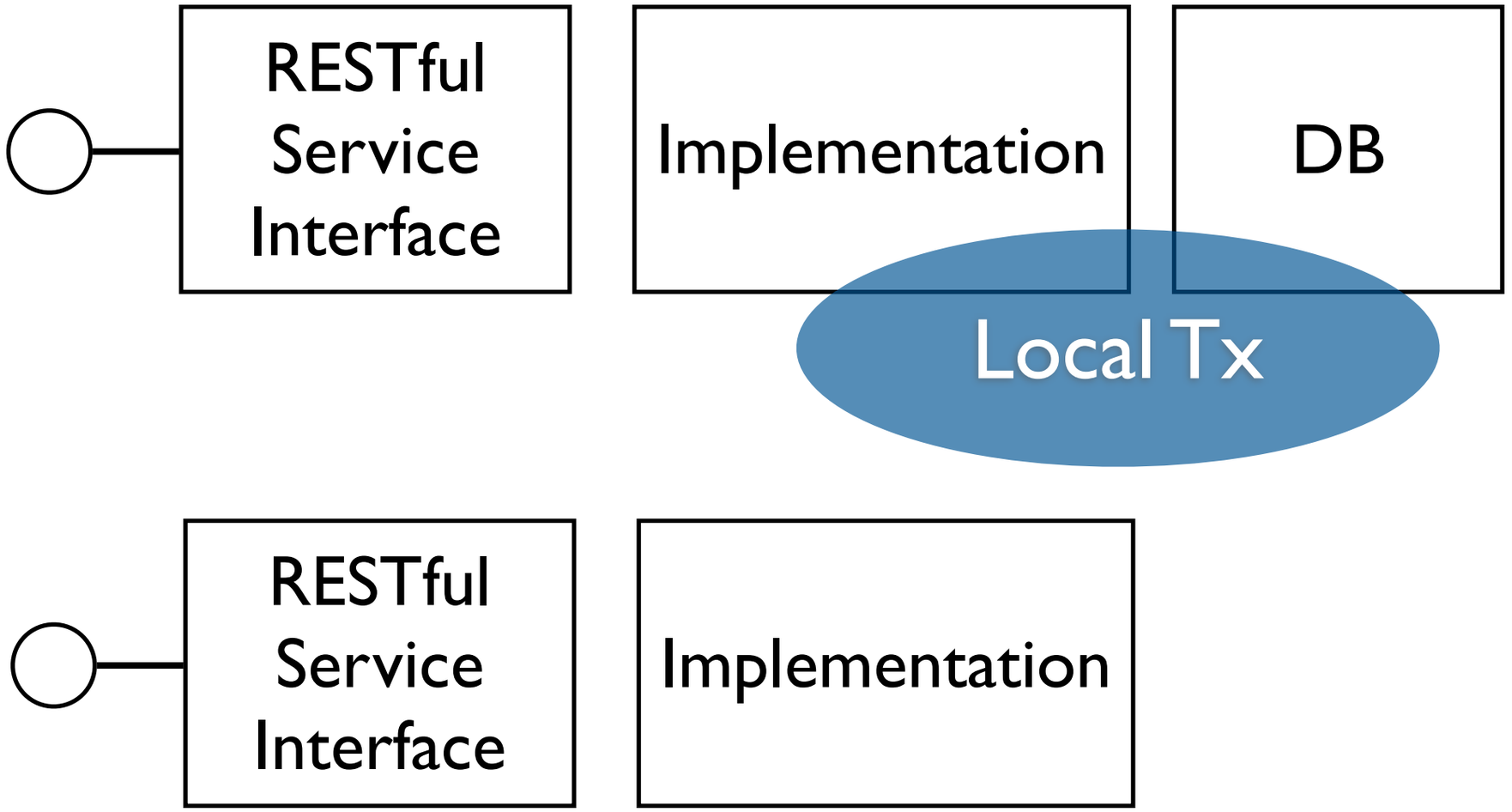


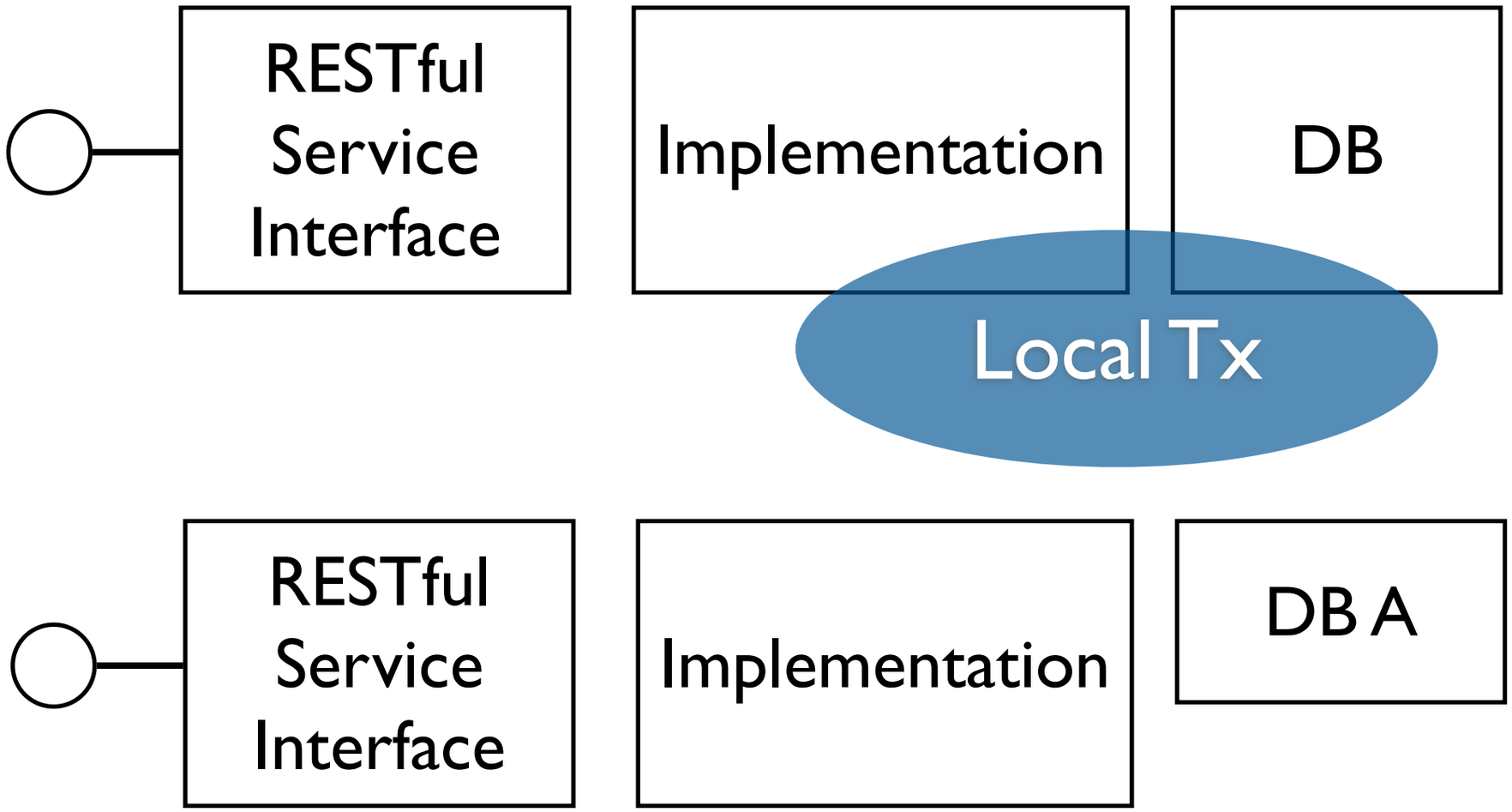


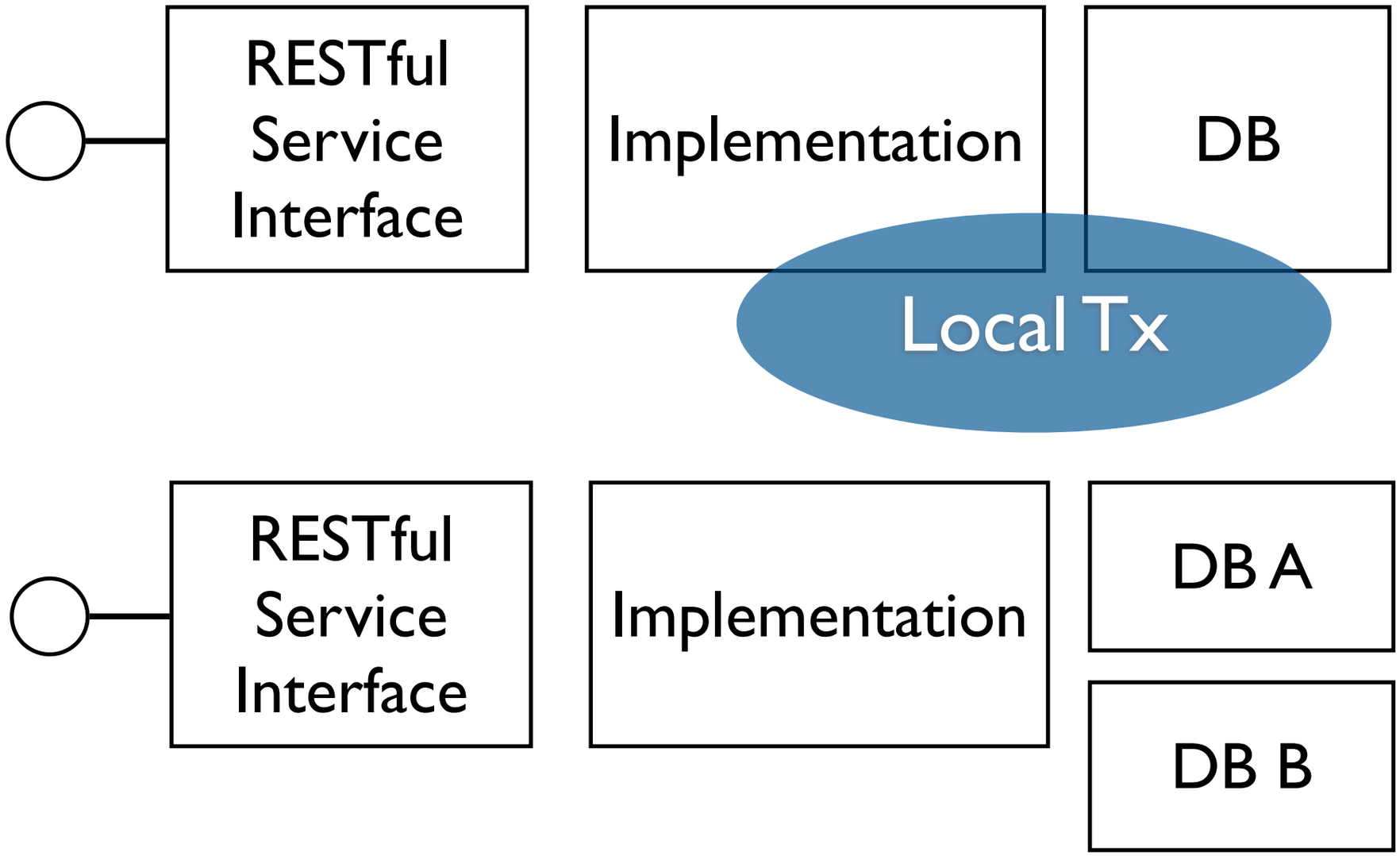


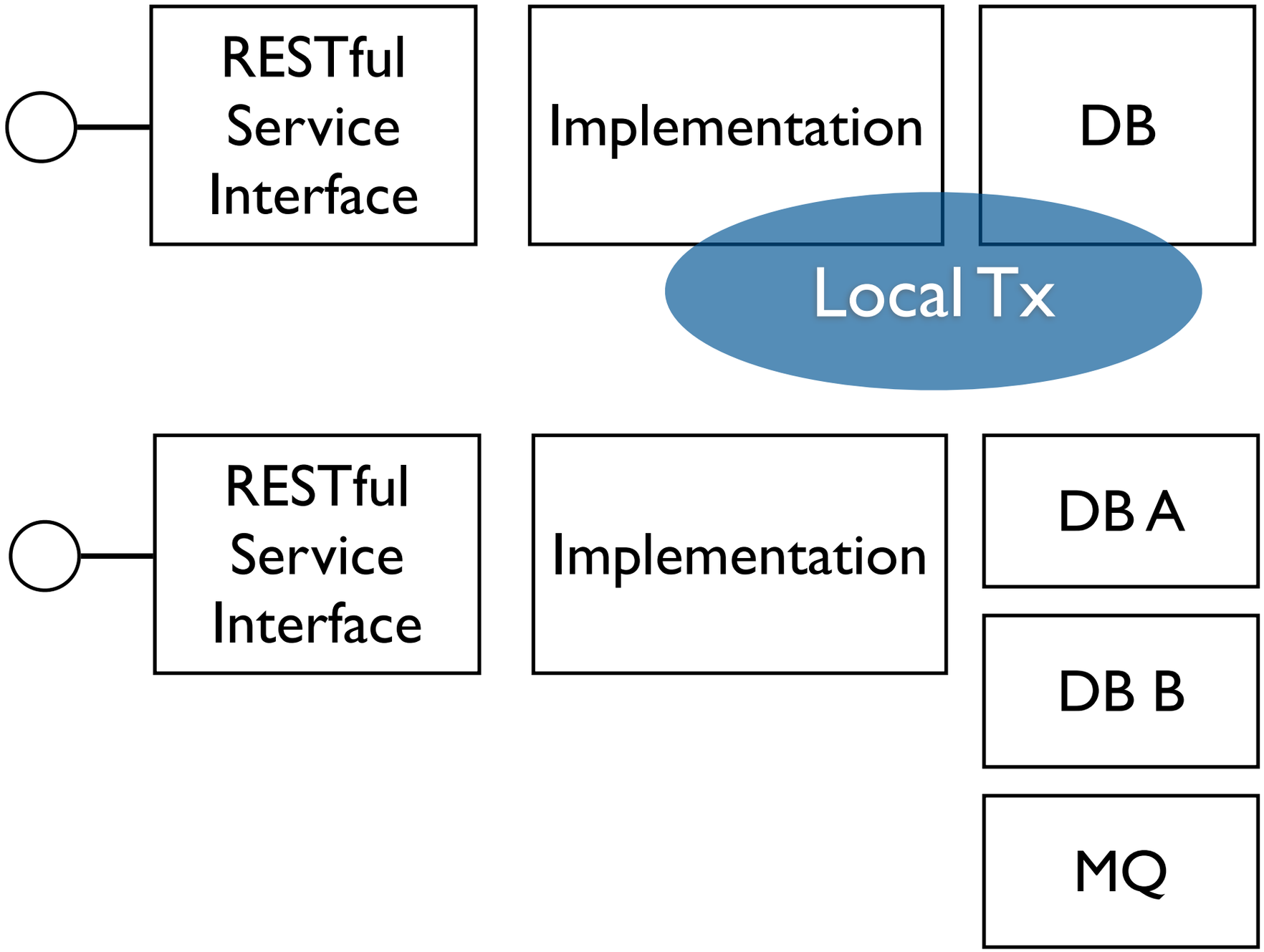


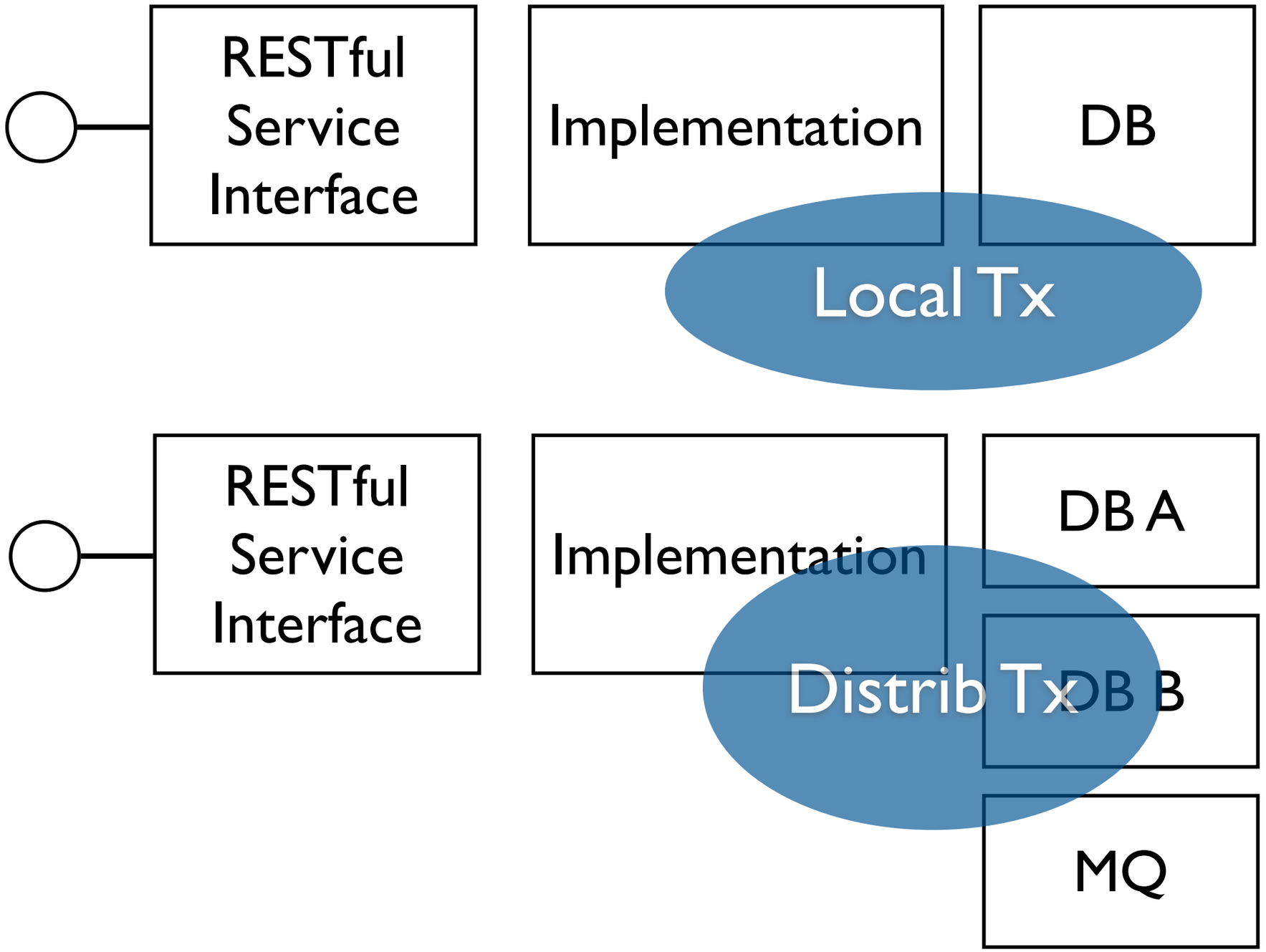


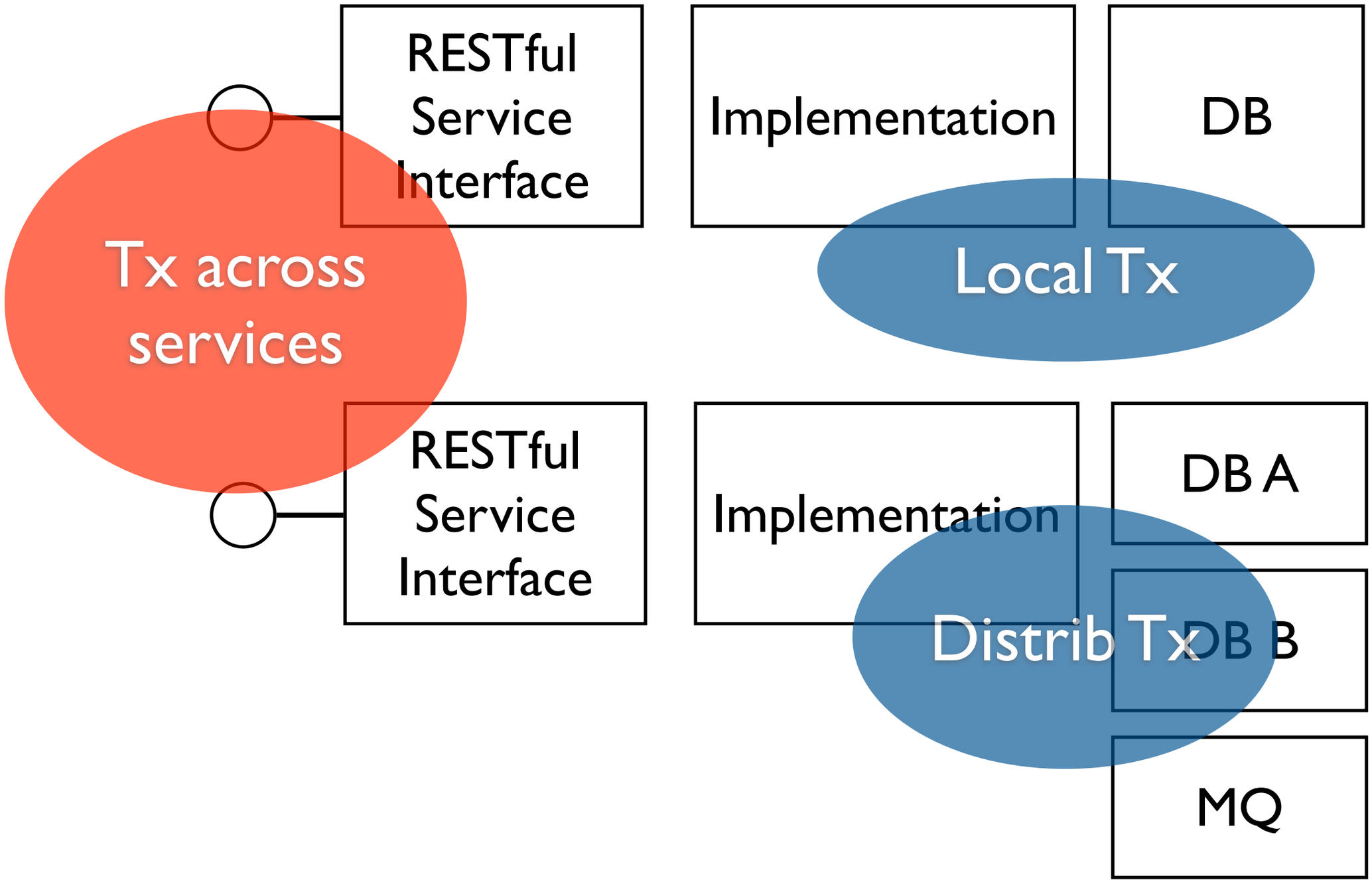












create new Tx resource

.....

create new Tx resource

→POST /transactions
←Location: http://.../transactions/4711
<status>in progress</status>

create new Tx resource

→POST /transactions
←Location: http://.../transactions/4711
<status>in progress</status>

---

augment state

---

create new Tx resource	→POST /transactions ←Location: http://.../transactions/4711 <status>in progress</status>
augment state	→POST /transactions/4711 ↵ ...data...

create new Tx resource

→POST /transactions
←Location: http://.../transactions/4711
<status>in progress</status>

augment state

→POST /transactions/4711 ↵
...data...

get state

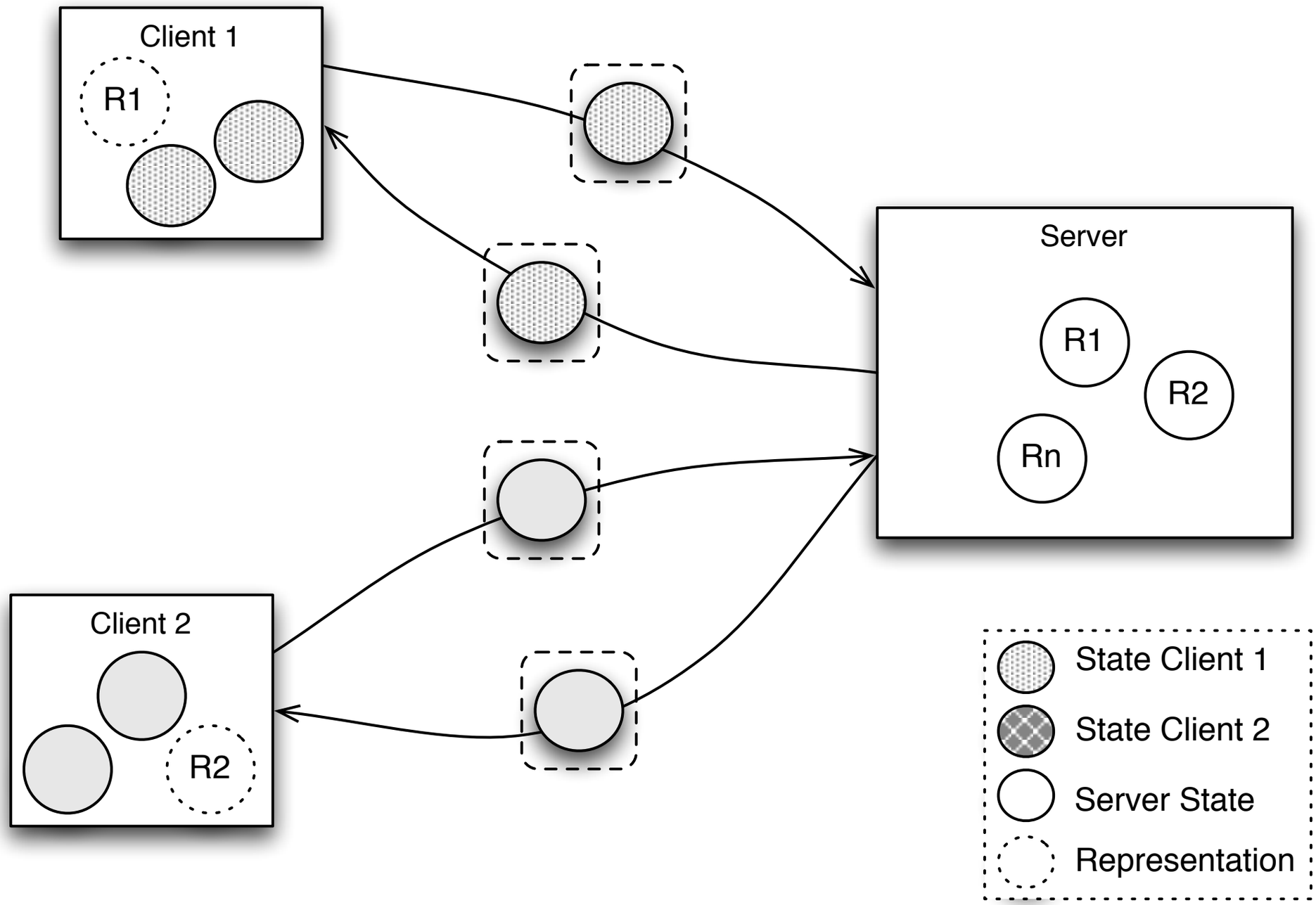
create new Tx resource	→POST /transactions ←Location: http://.../transactions/4711 <status>in progress</status>
augment state	→POST /transactions/4711 ↵ ...data...
get state	→GET /transactions/4711 ←... data ...↵ <link rel='status'>in progress</link>

create new Tx resource	→POST /transactions ←Location: http://.../transactions/4711 <status>in progress</status>
augment state	→POST /transactions/4711 ↵ ...data...
get state	→GET /transactions/4711 ←... data ...↵ <link rel='status'>in progress</link>
commit	

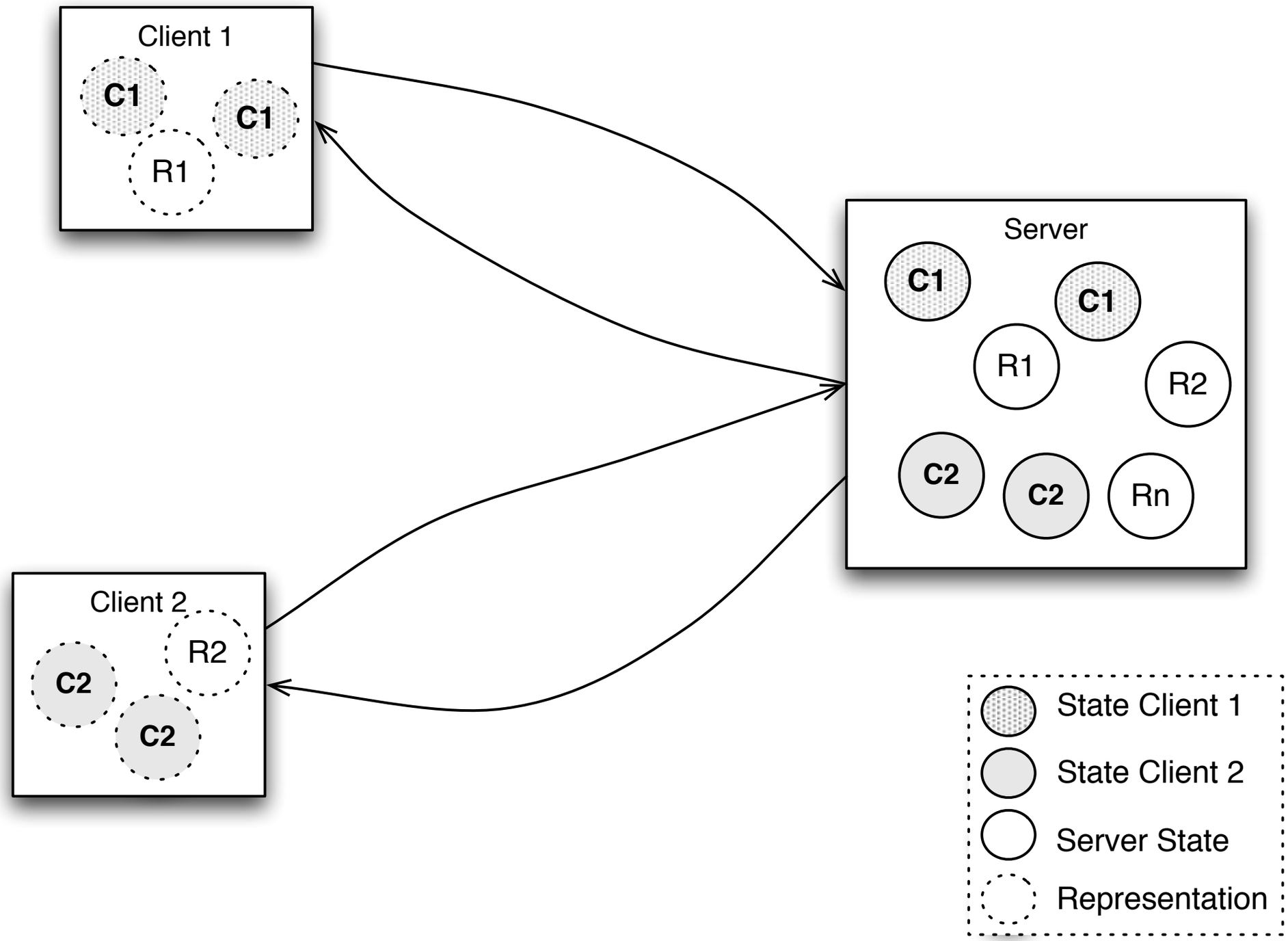
create new Tx resource	→POST /transactions ←Location: http://.../transactions/4711 <status>in progress</status>
augment state	→POST /transactions/4711 ↵ ...data...
get state	→GET /transactions/4711 ←... data ...↵ <link rel='status'>in progress</link>
commit	→PUT /transactions/4711/status↵ <status>committed</status>

Stateful Communication

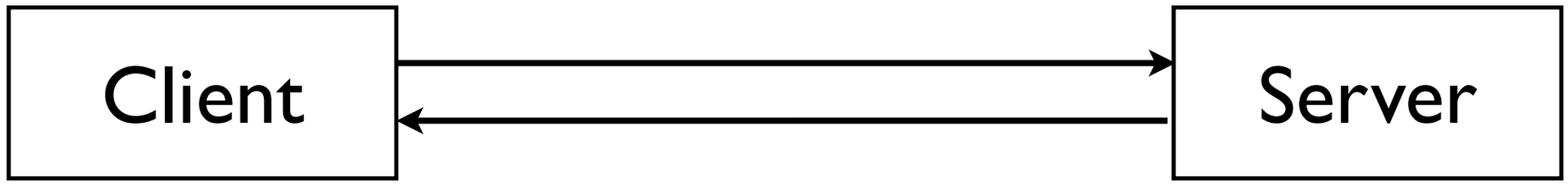
Turn session state …

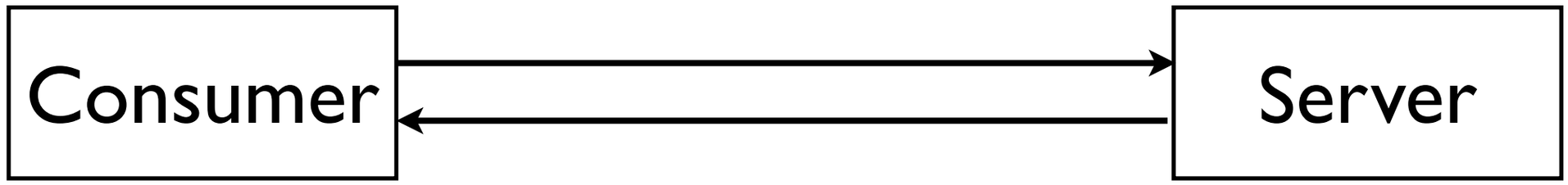


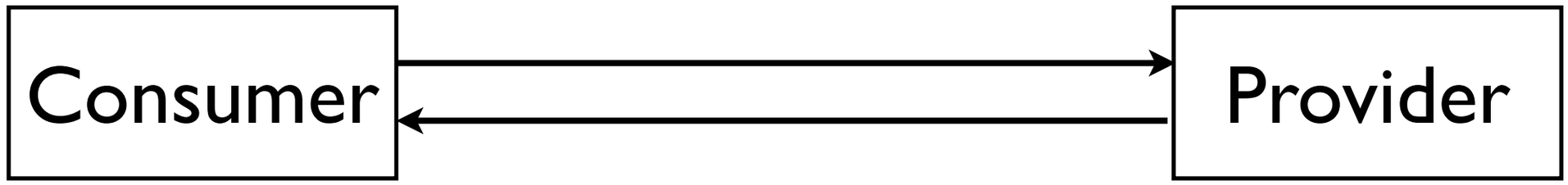
… into client or resource state

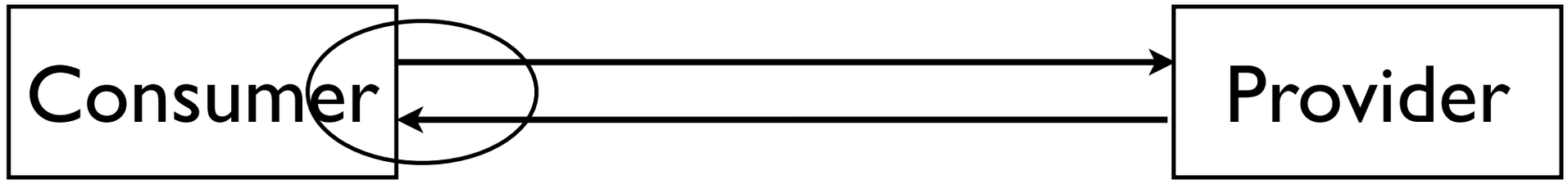


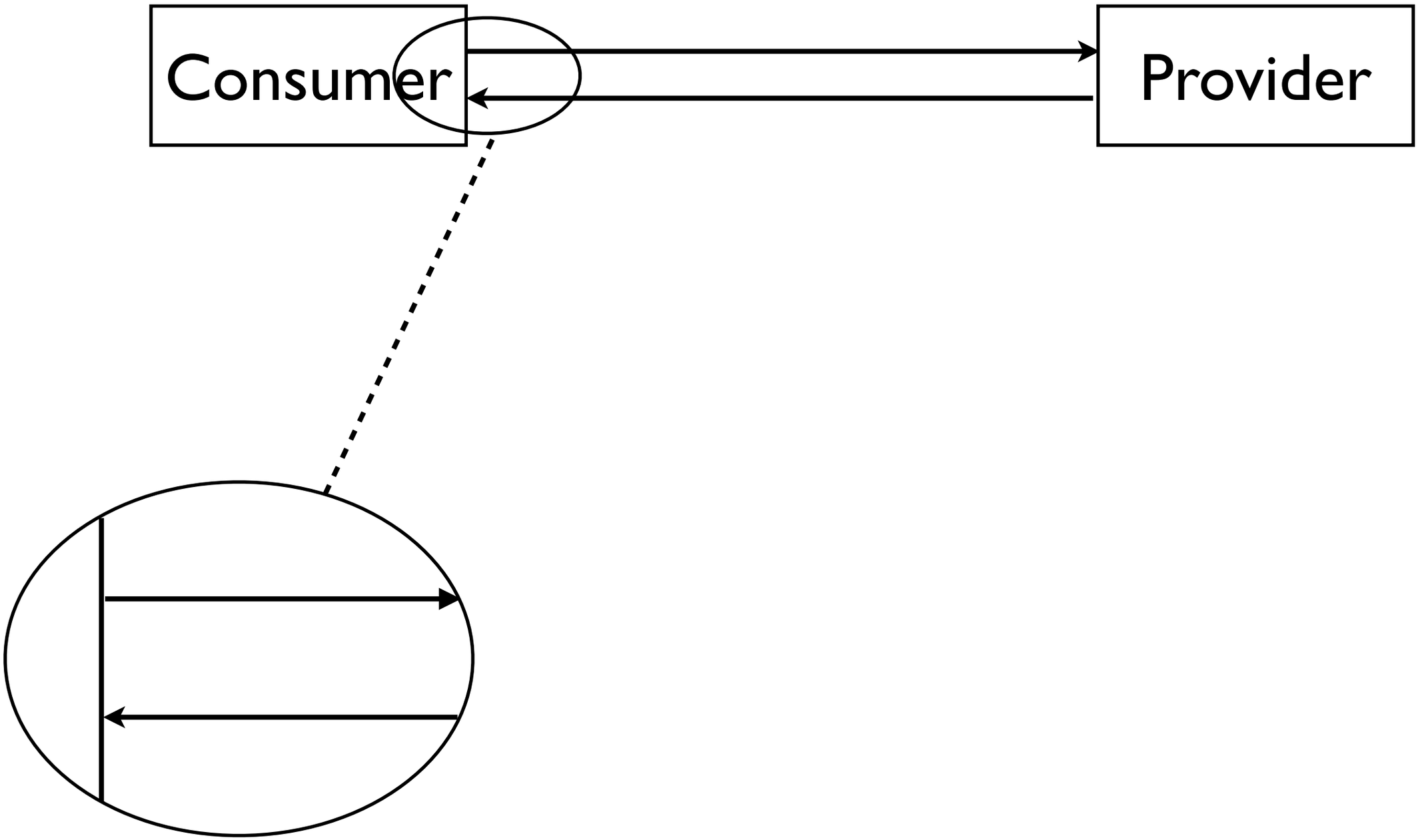
Reliable Messaging

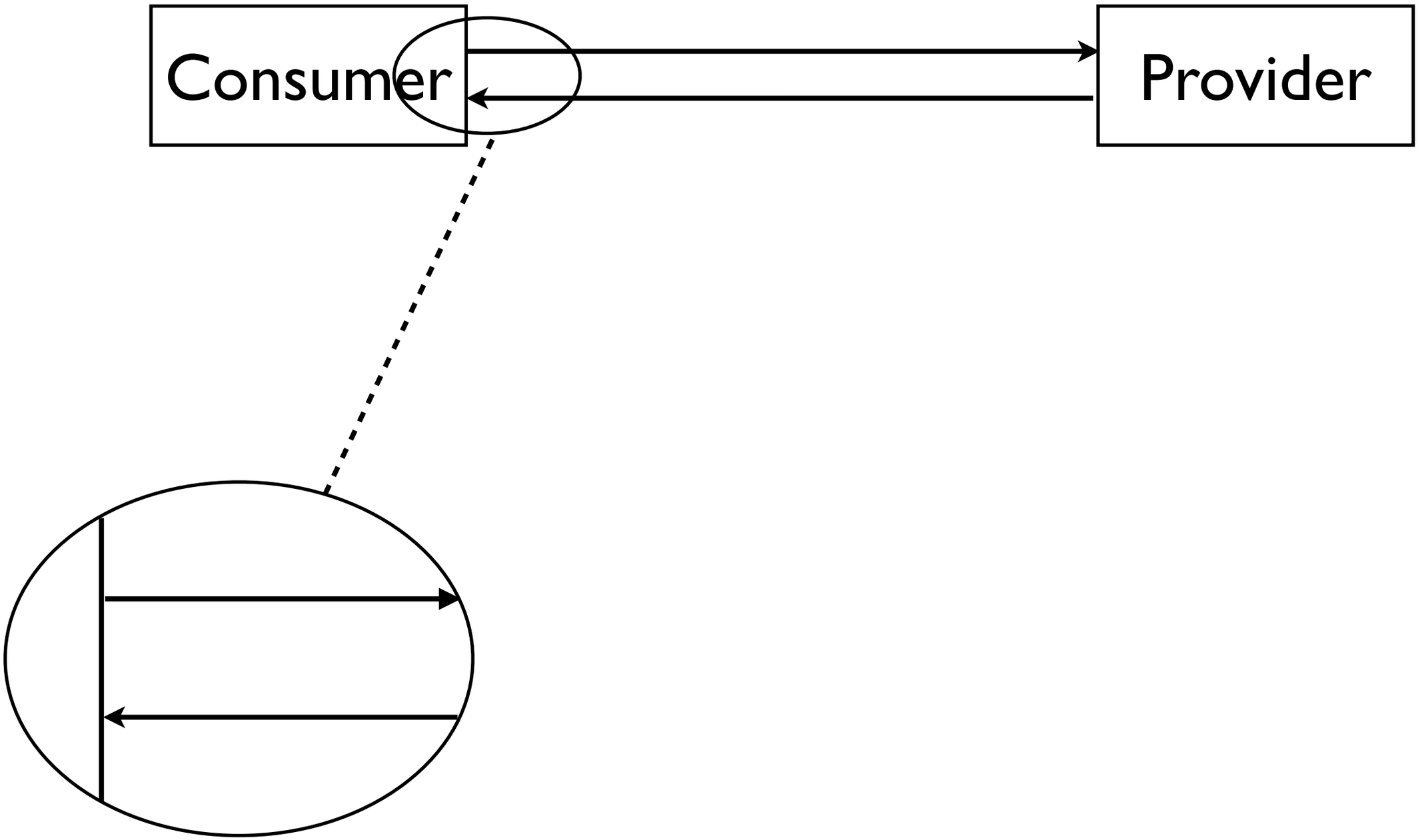


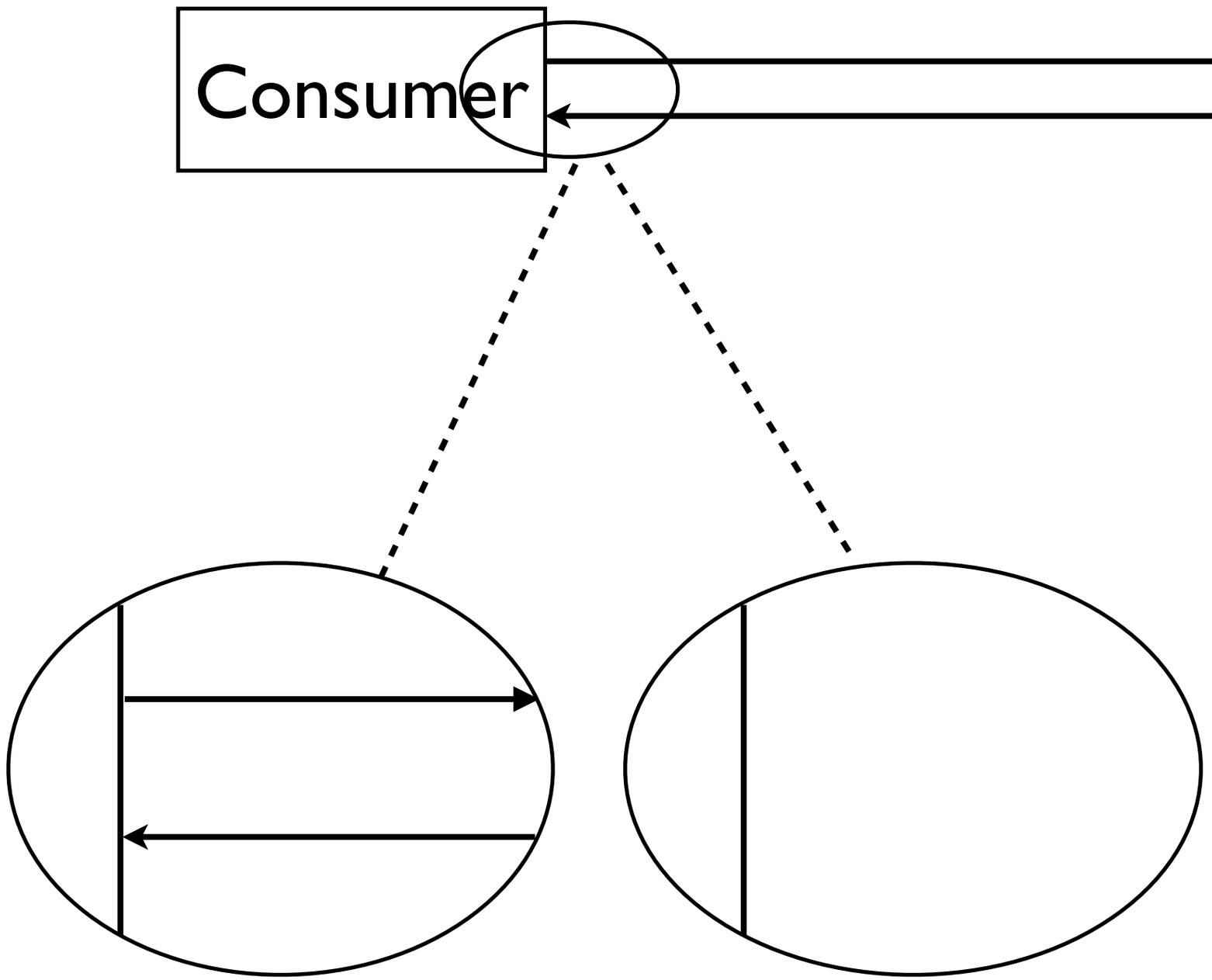




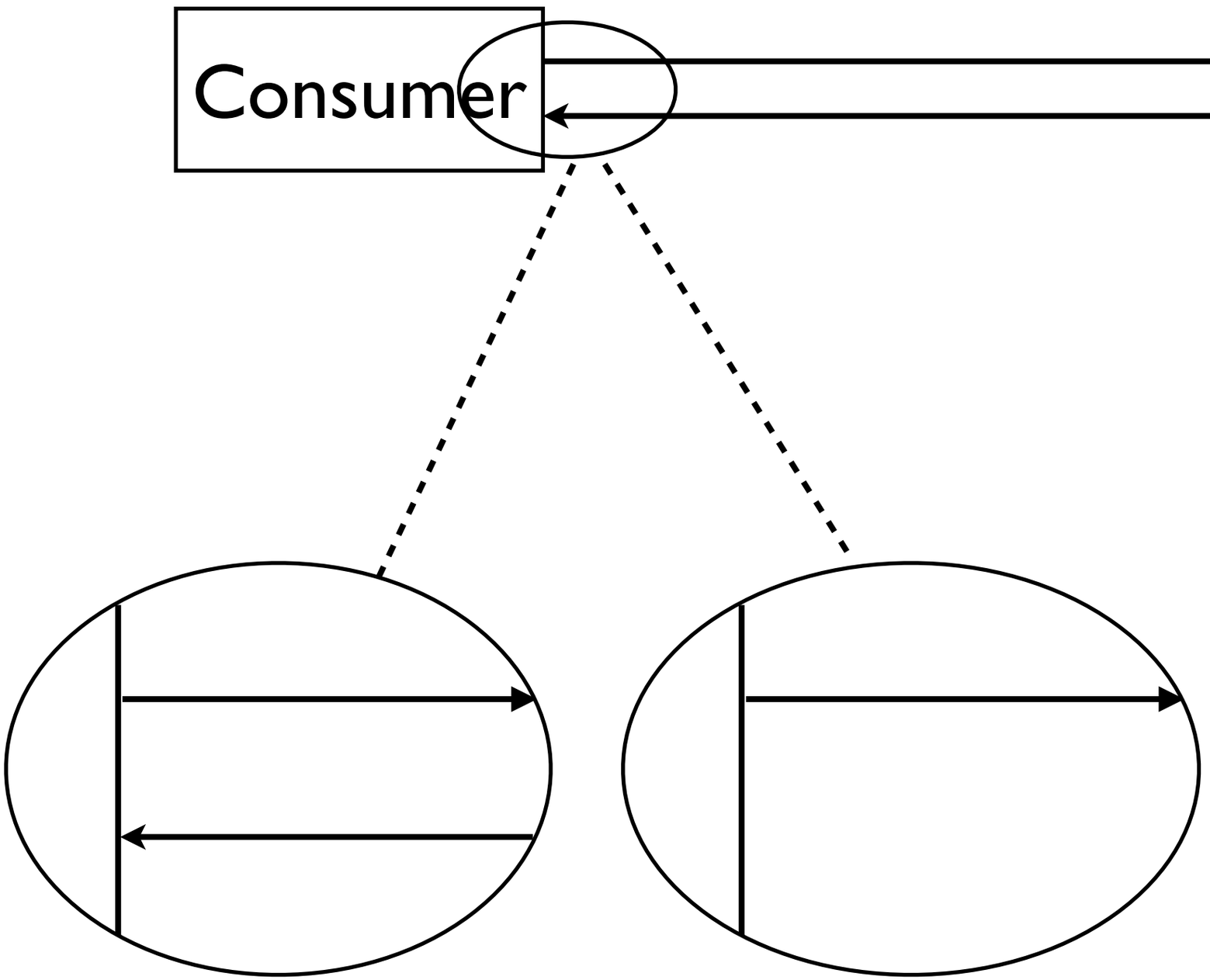




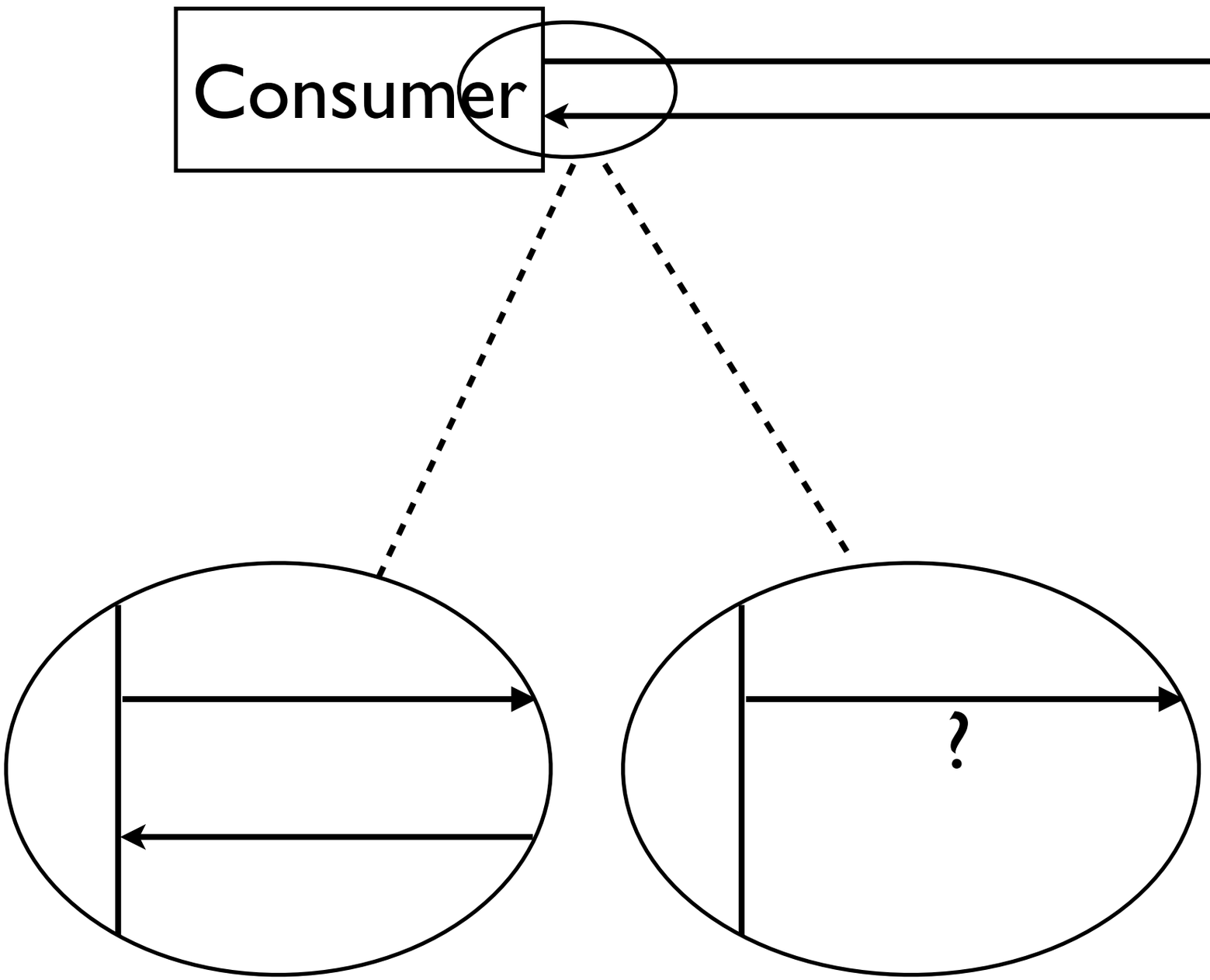




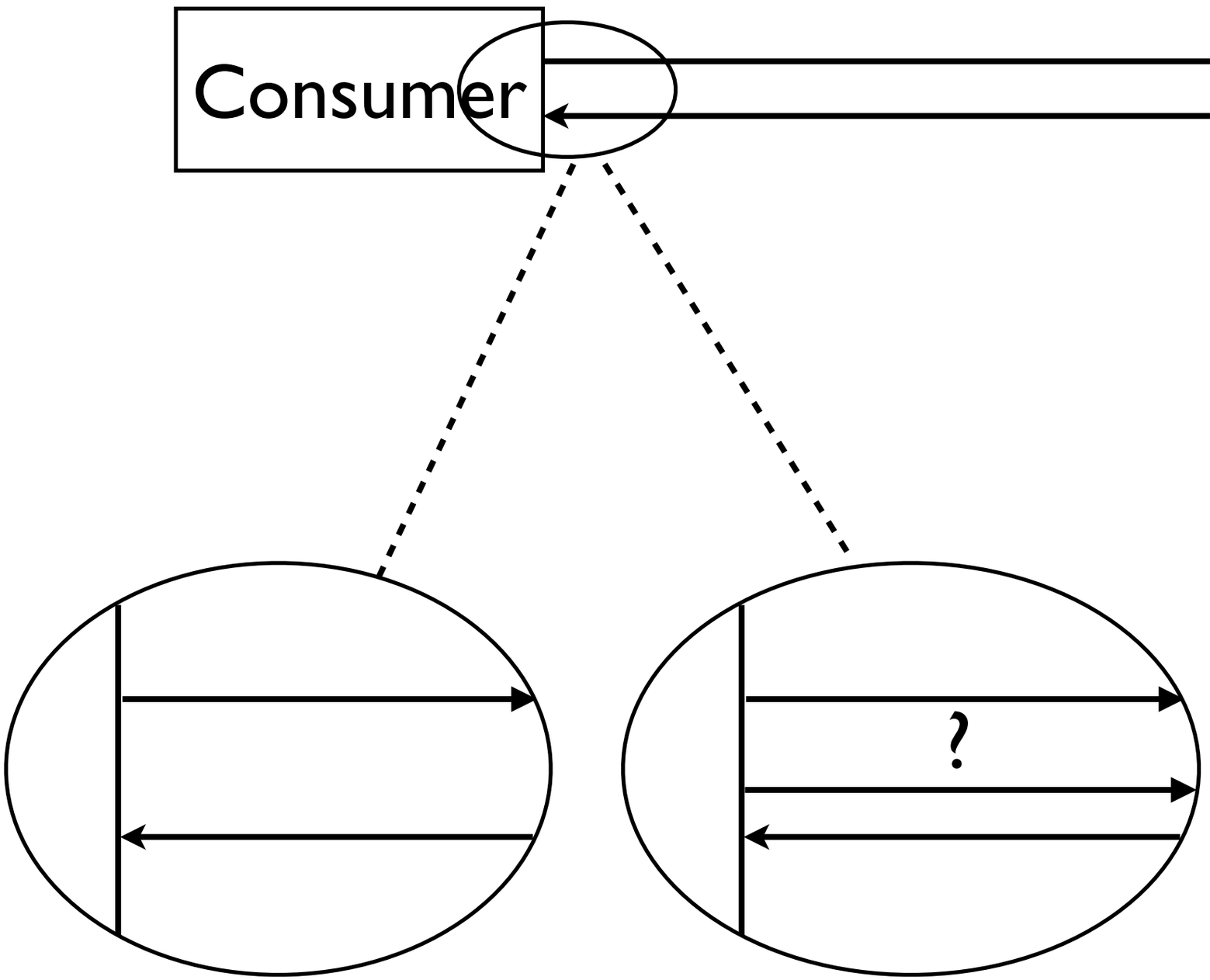
Success/
Error



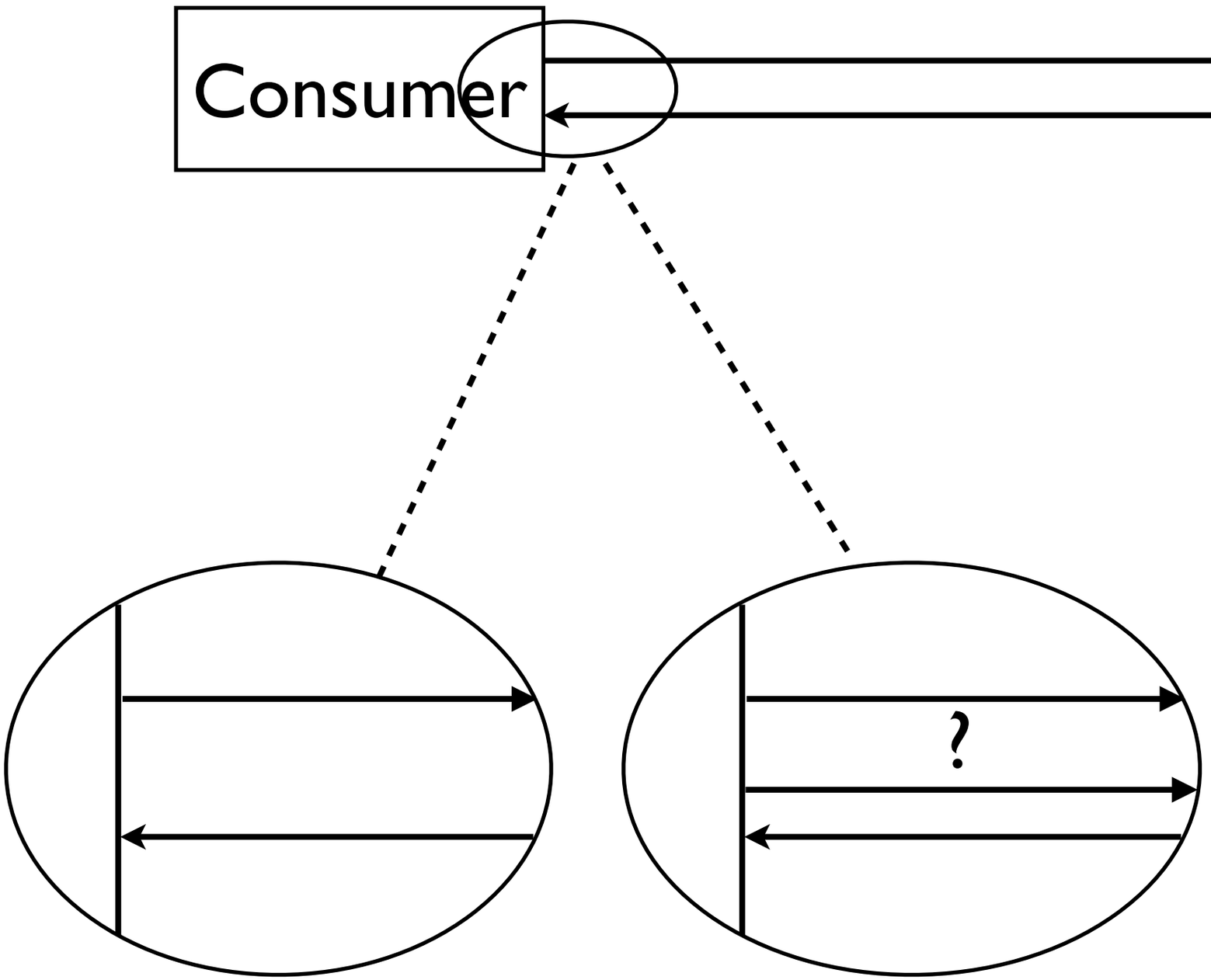
Success/
Error



Success/
Error

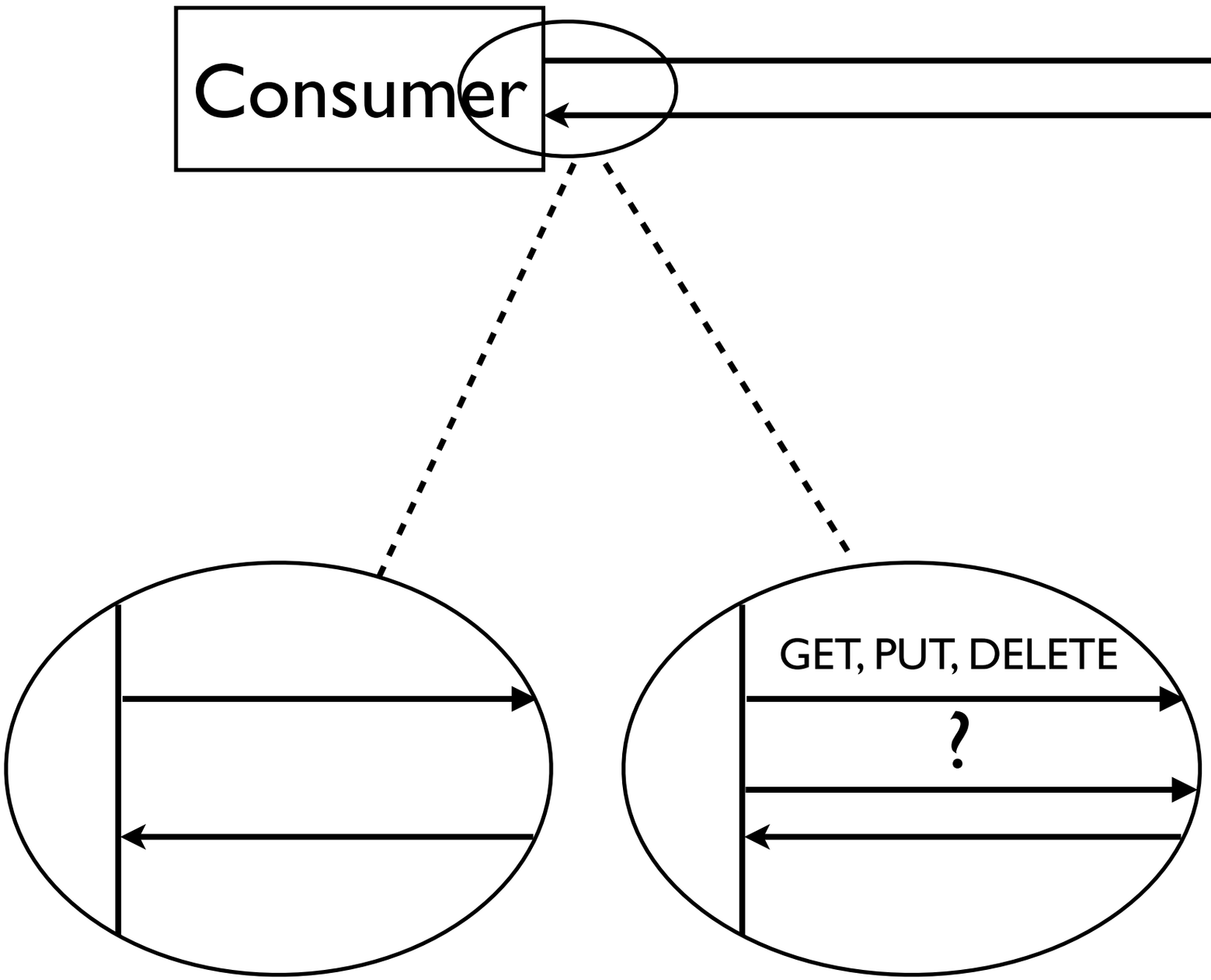


Success/
Error



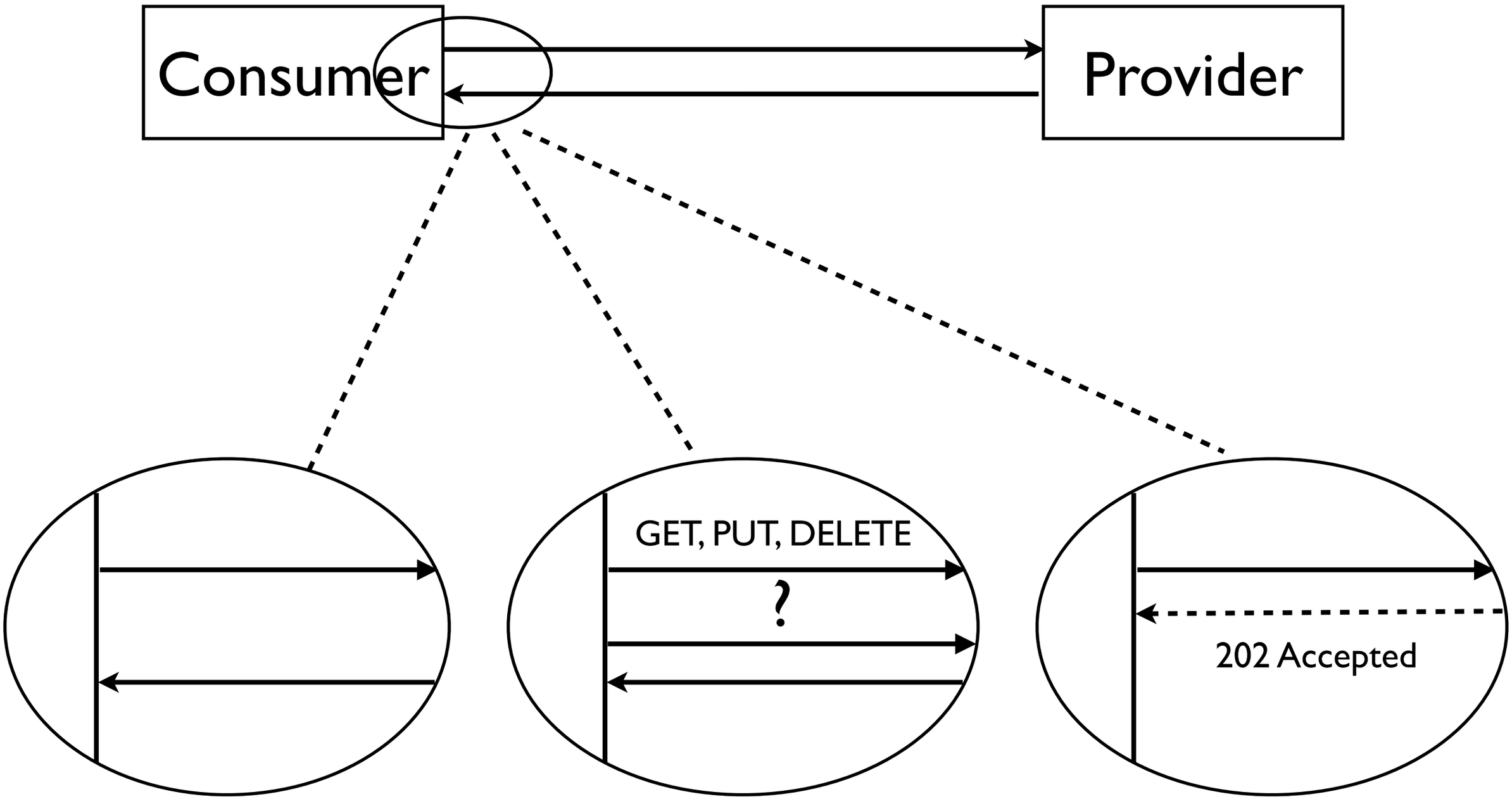
Success/
Error

Idempotent
Retry



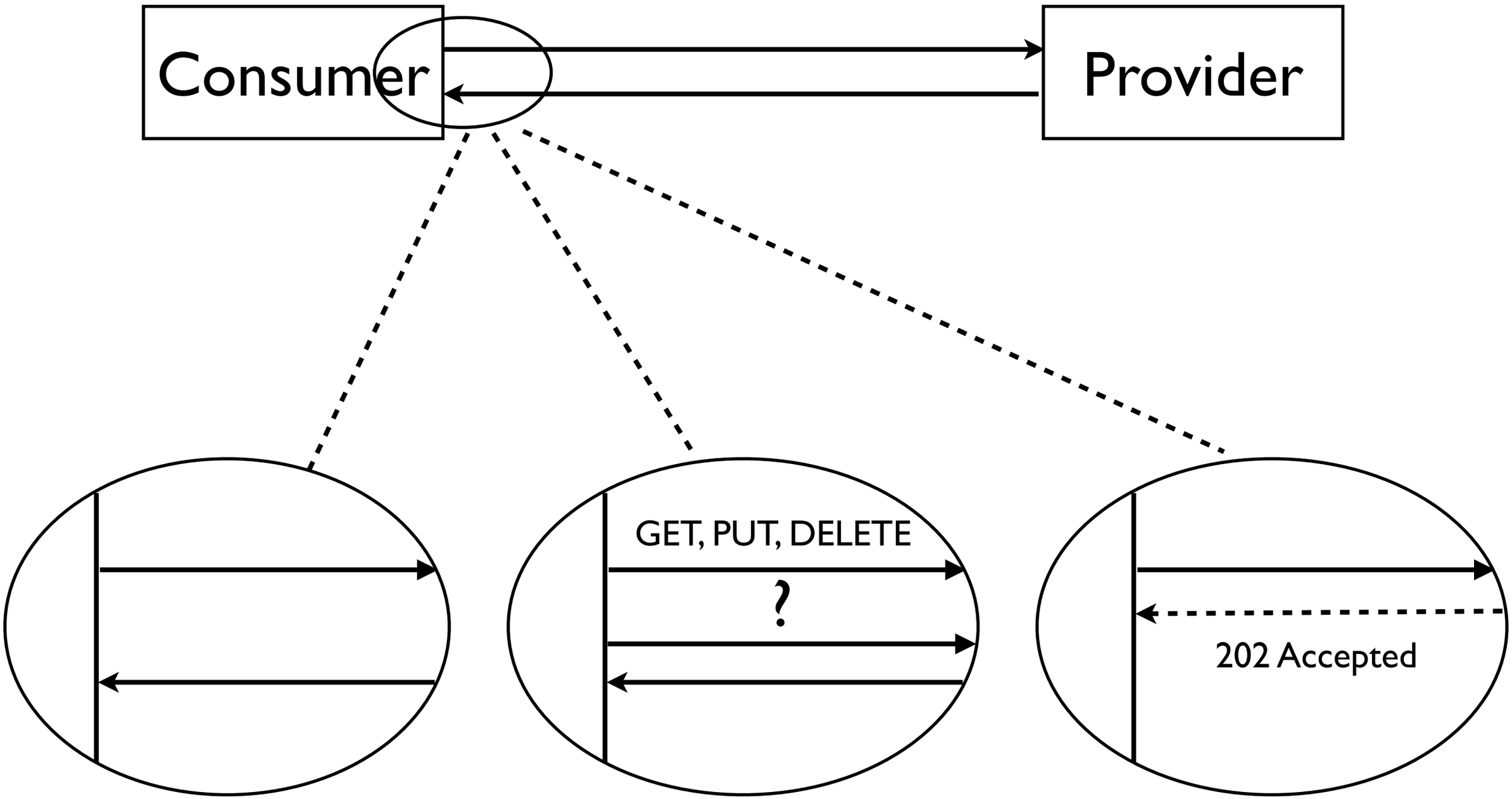
Success/
Error

Idempotent
Retry



Success/
Error

Idempotent
Retry



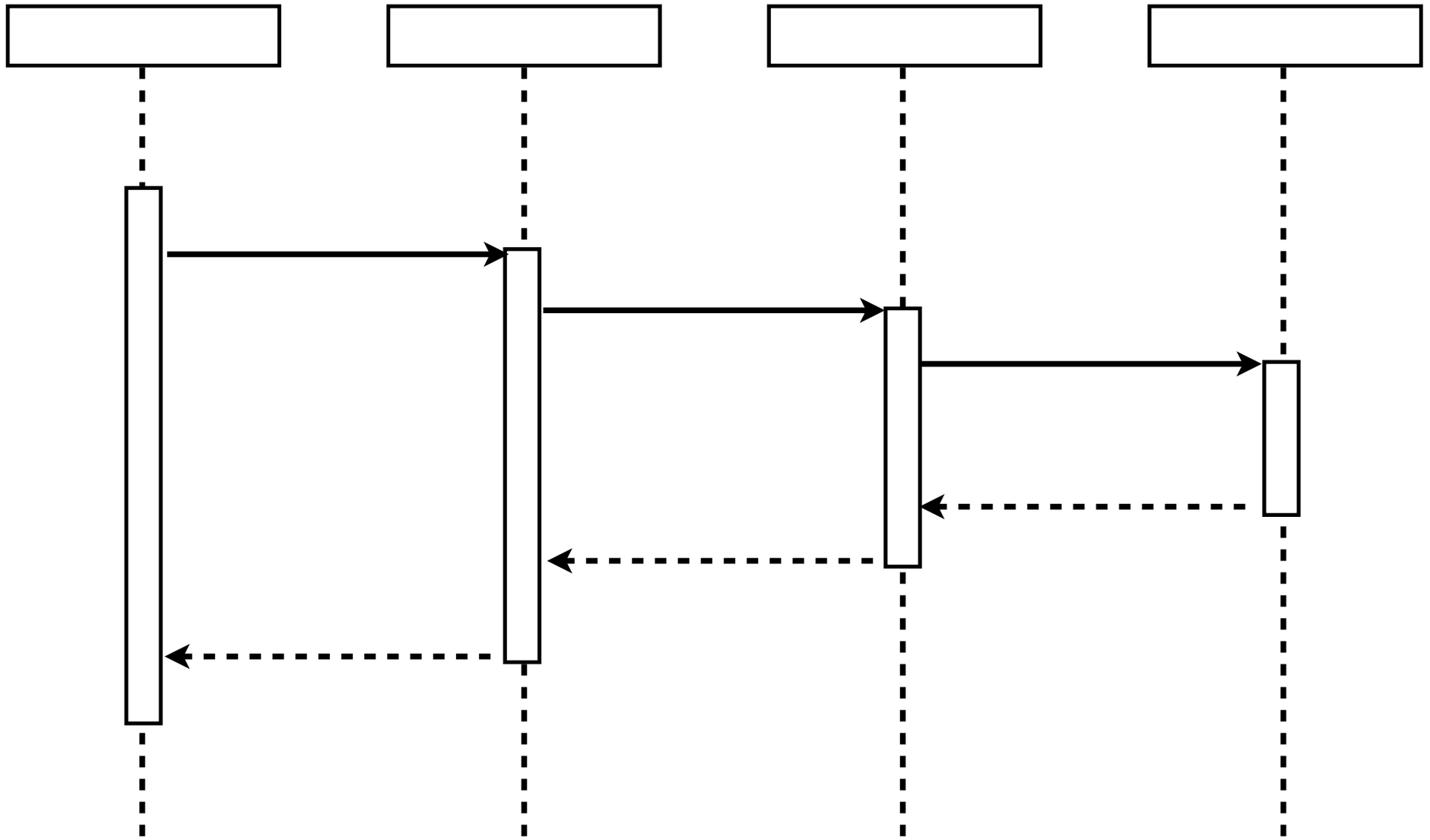
Success/
Error

Idempotent
Retry

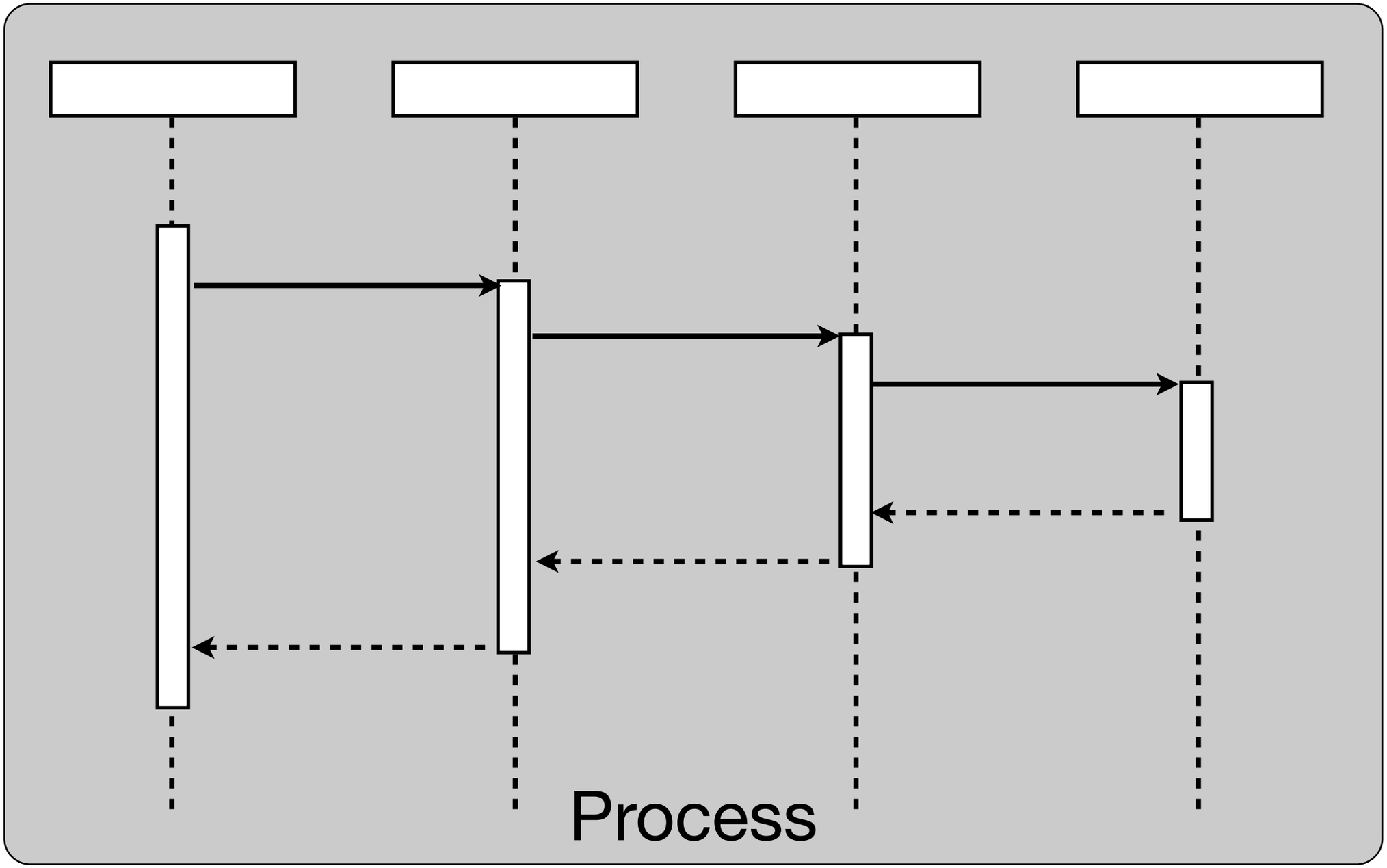
Async
Accepted

Notifications

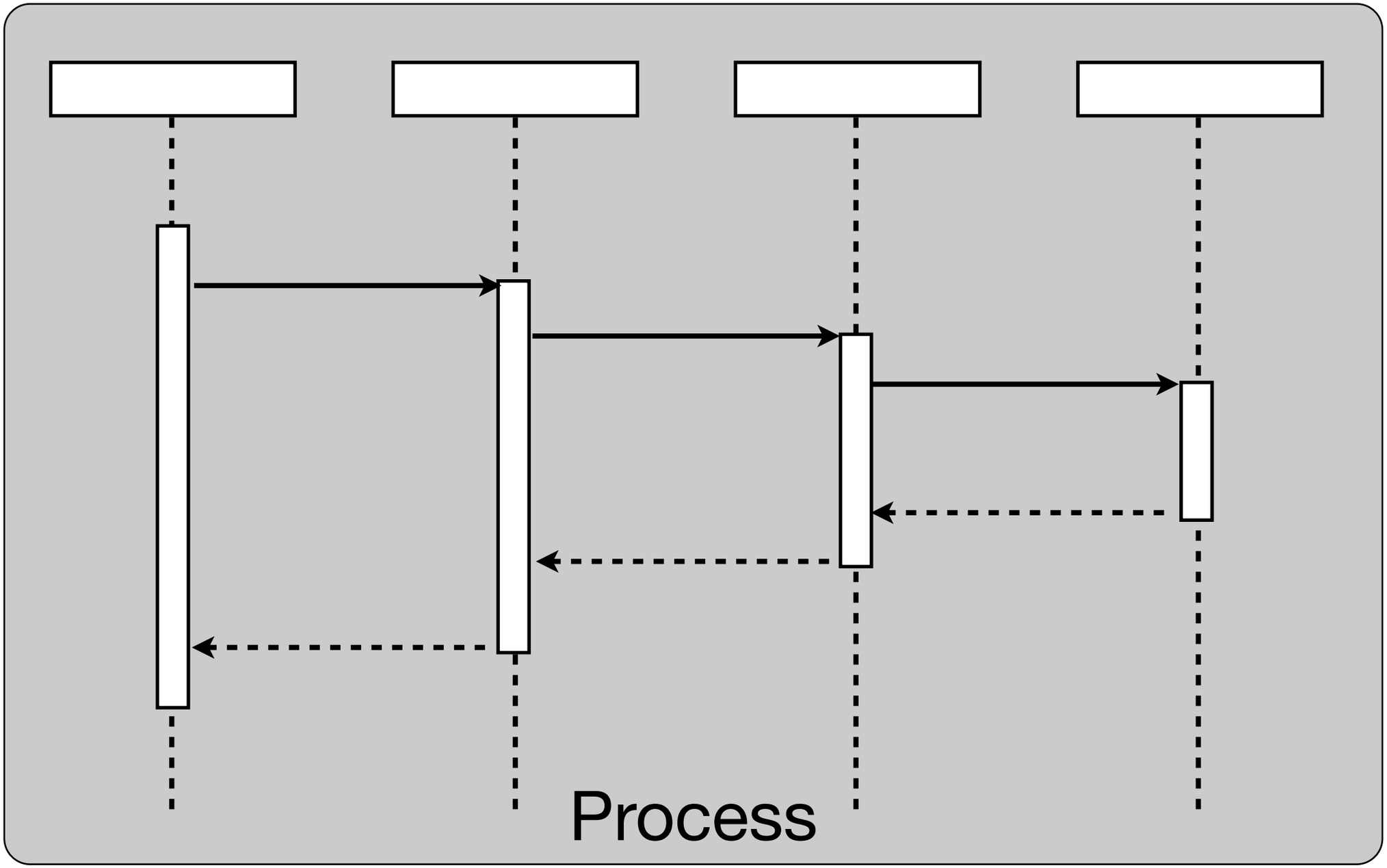
Call Stack



Call Stack

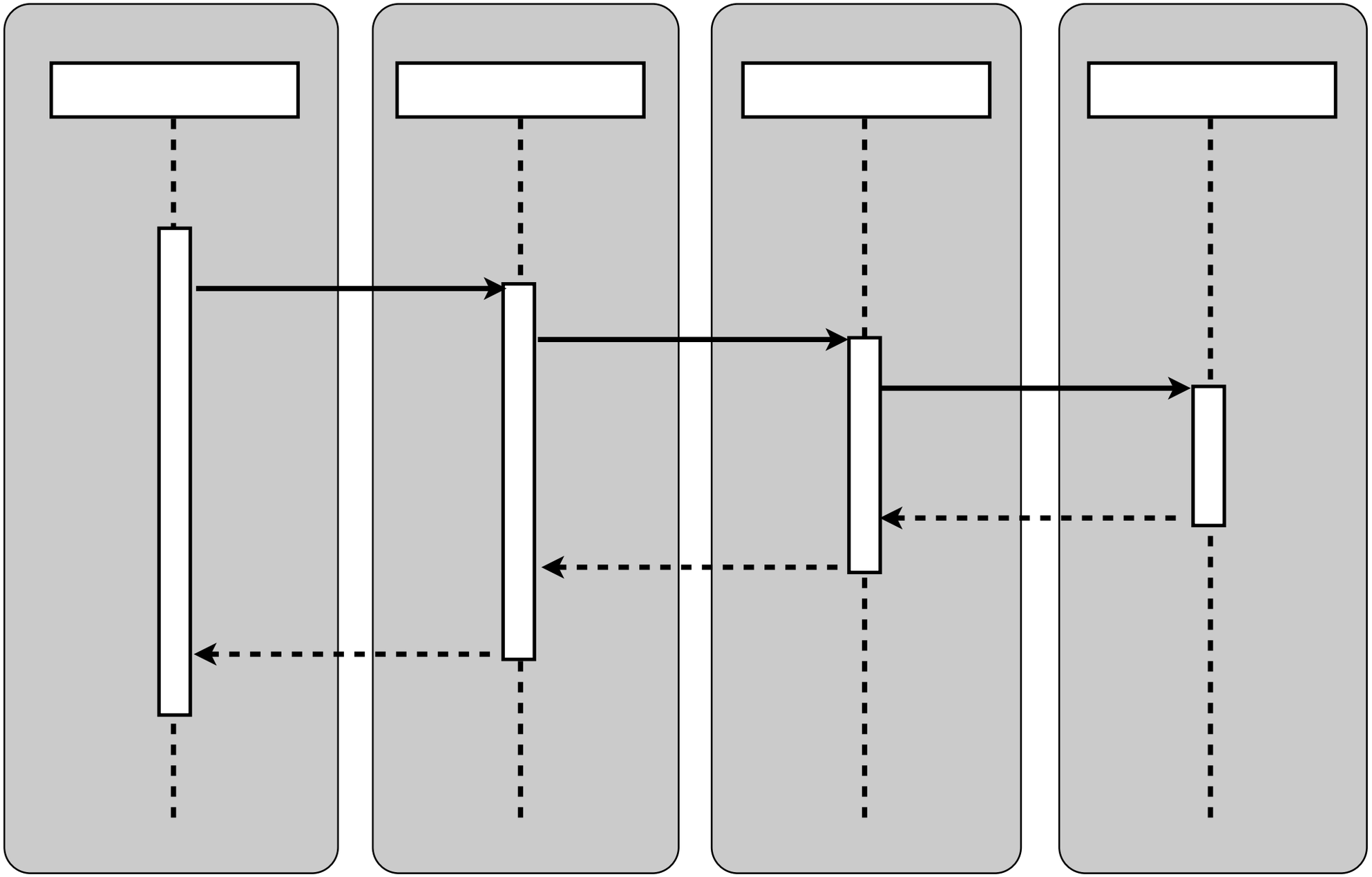


Call Stack

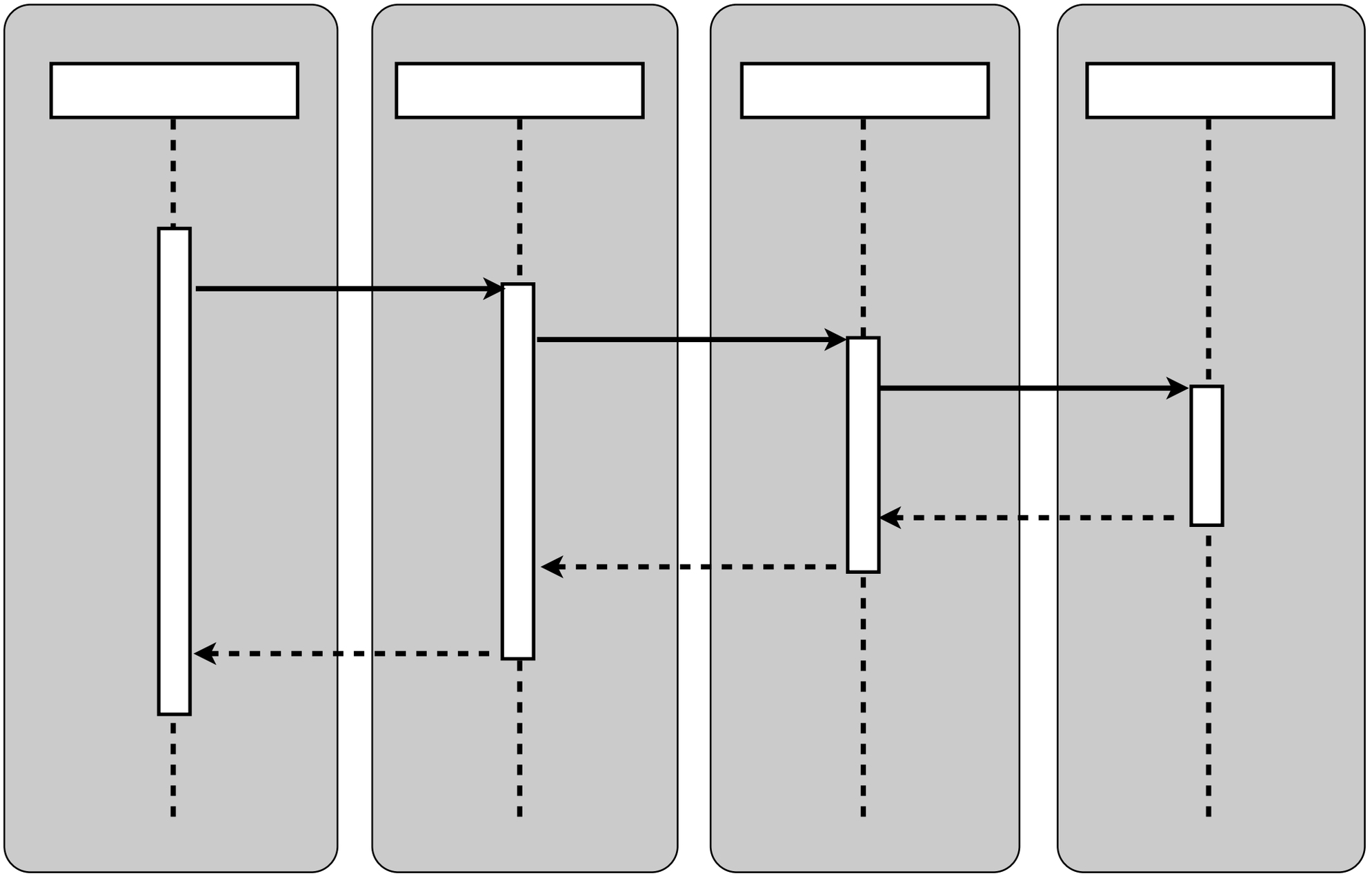


Assumed Success Probability: 99.9%

Distributed Call Stack

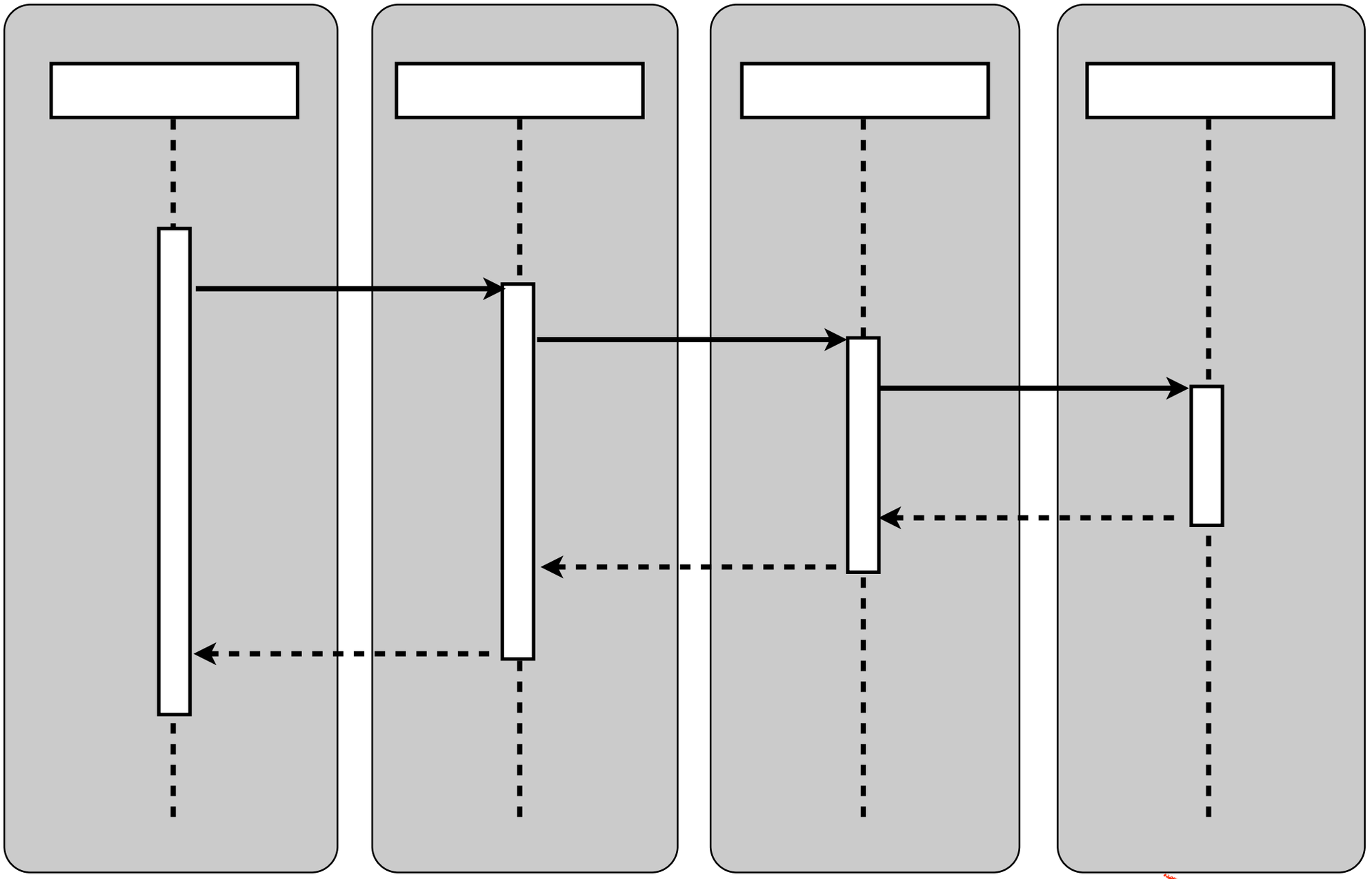


Distributed Call Stack



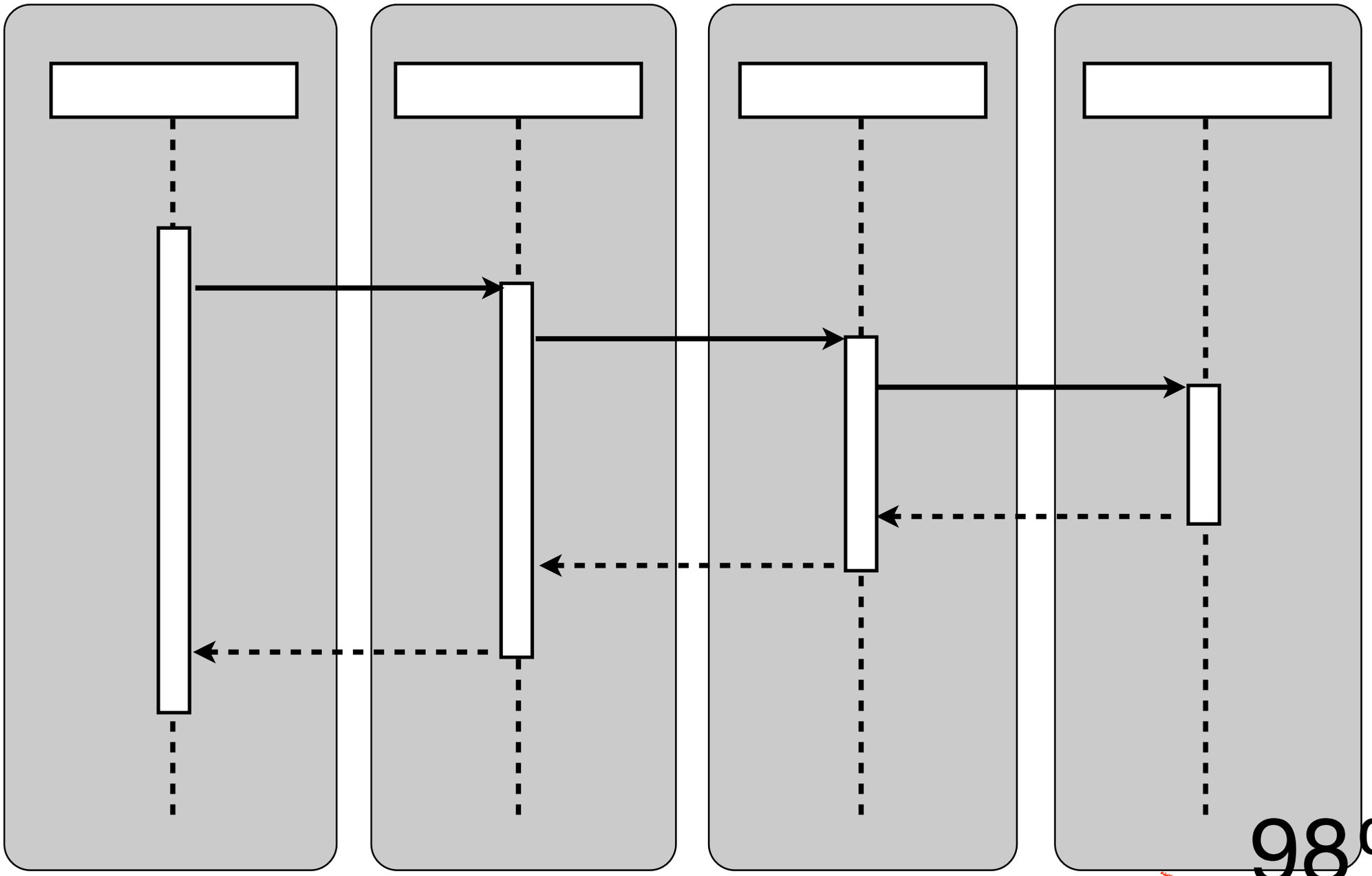
Assumed Success Probability: 99.6%

Distributed Call Stack



Assumed Success Probability: ~~99.6%~~

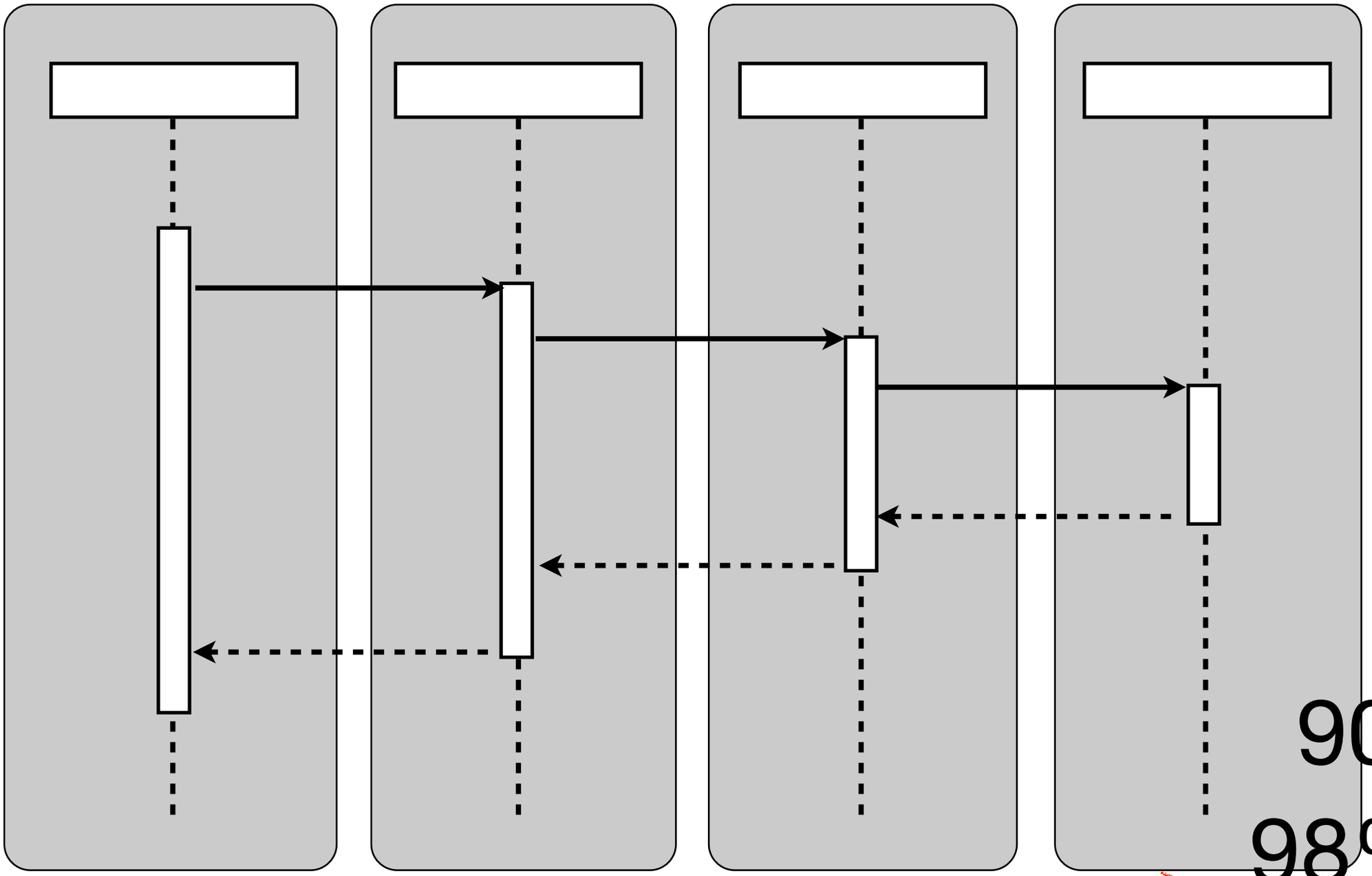
Distributed Call Stack



98%?

Assumed Success Probability: ~~99.6%~~

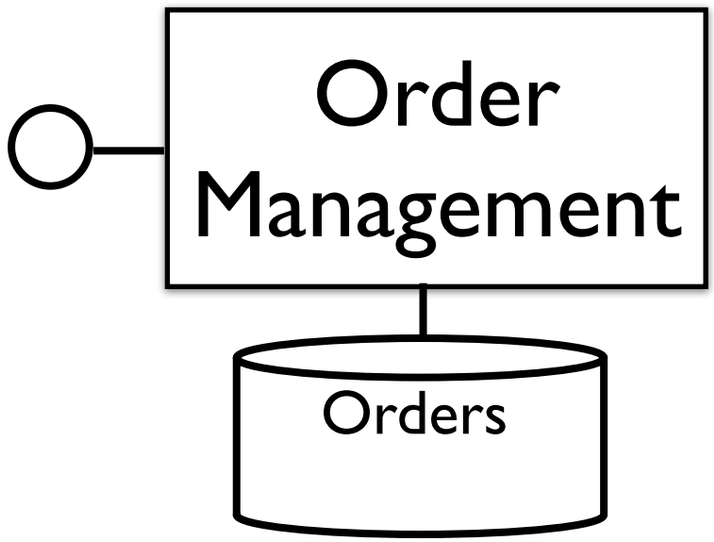
Distributed Call Stack

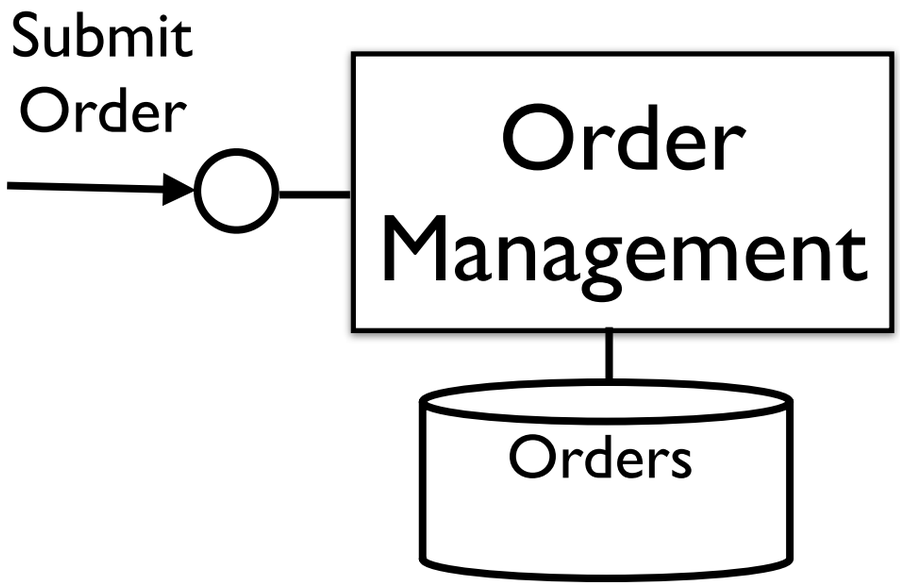


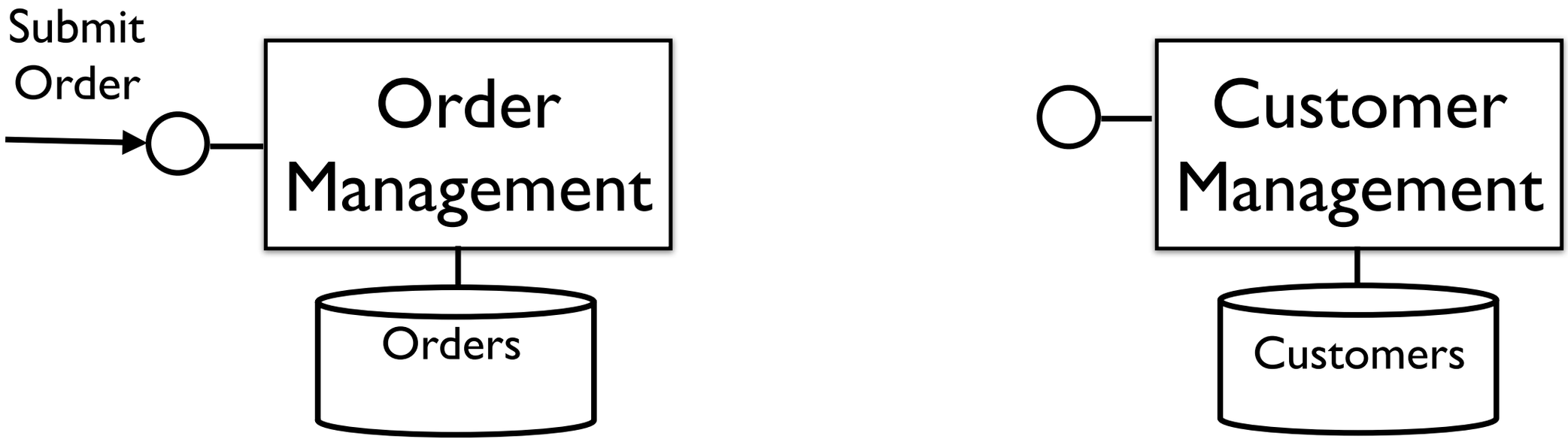
90%?

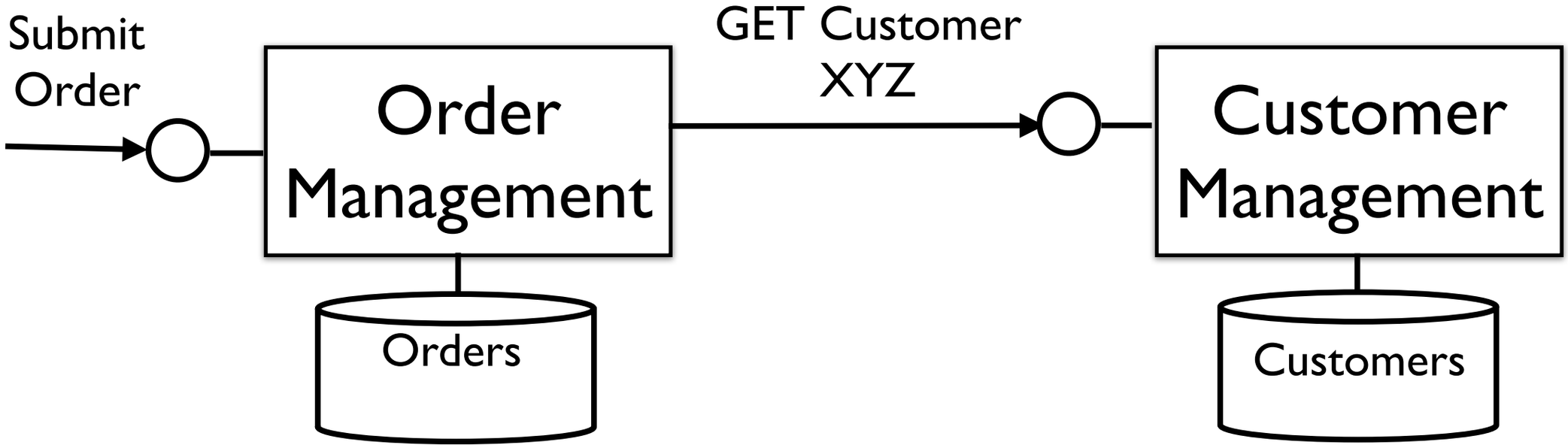
98%?

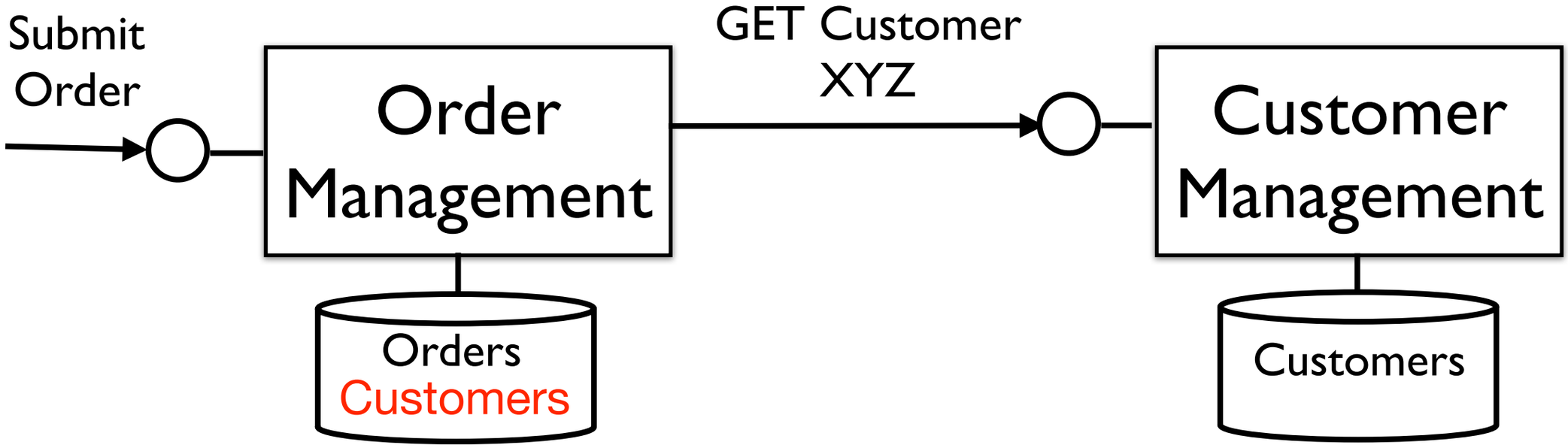
Assumed Success Probability: ~~99.6%~~

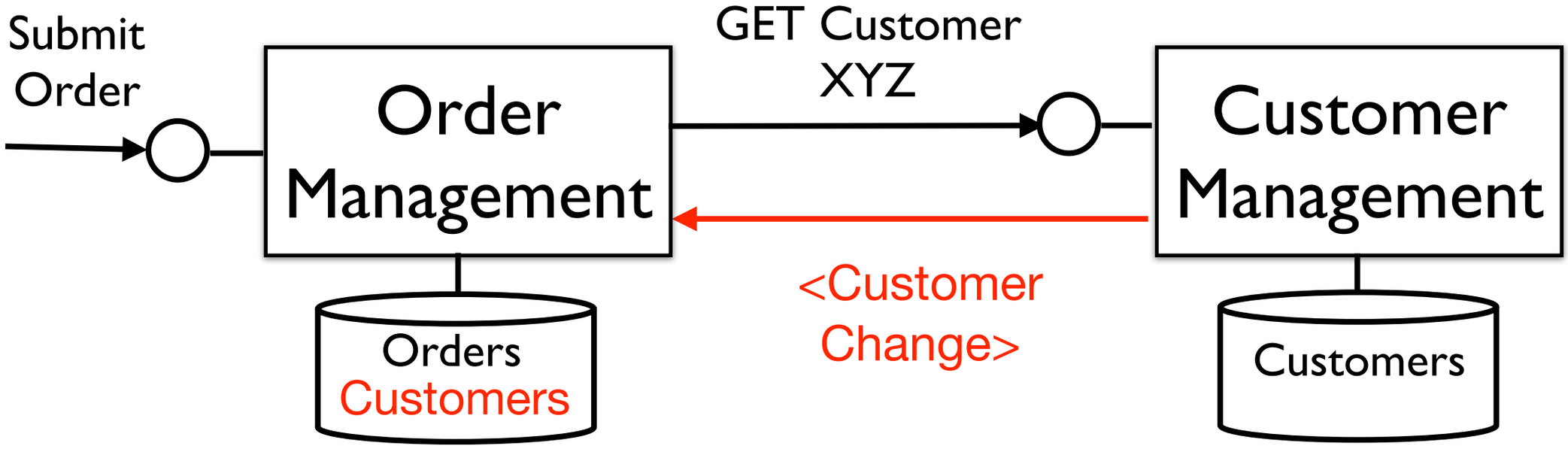


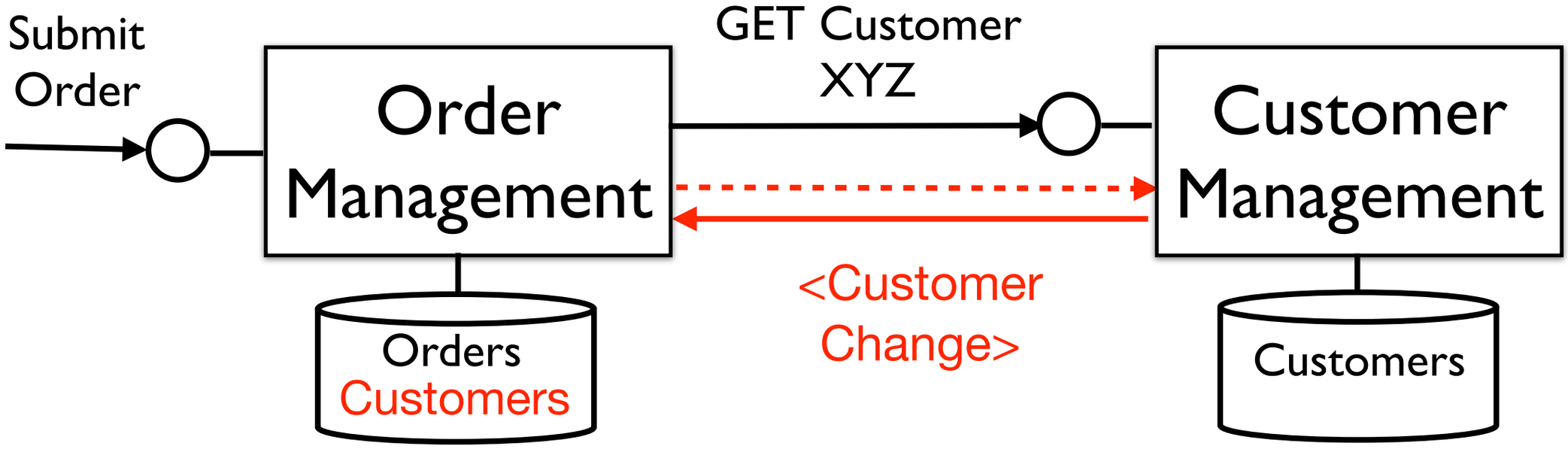


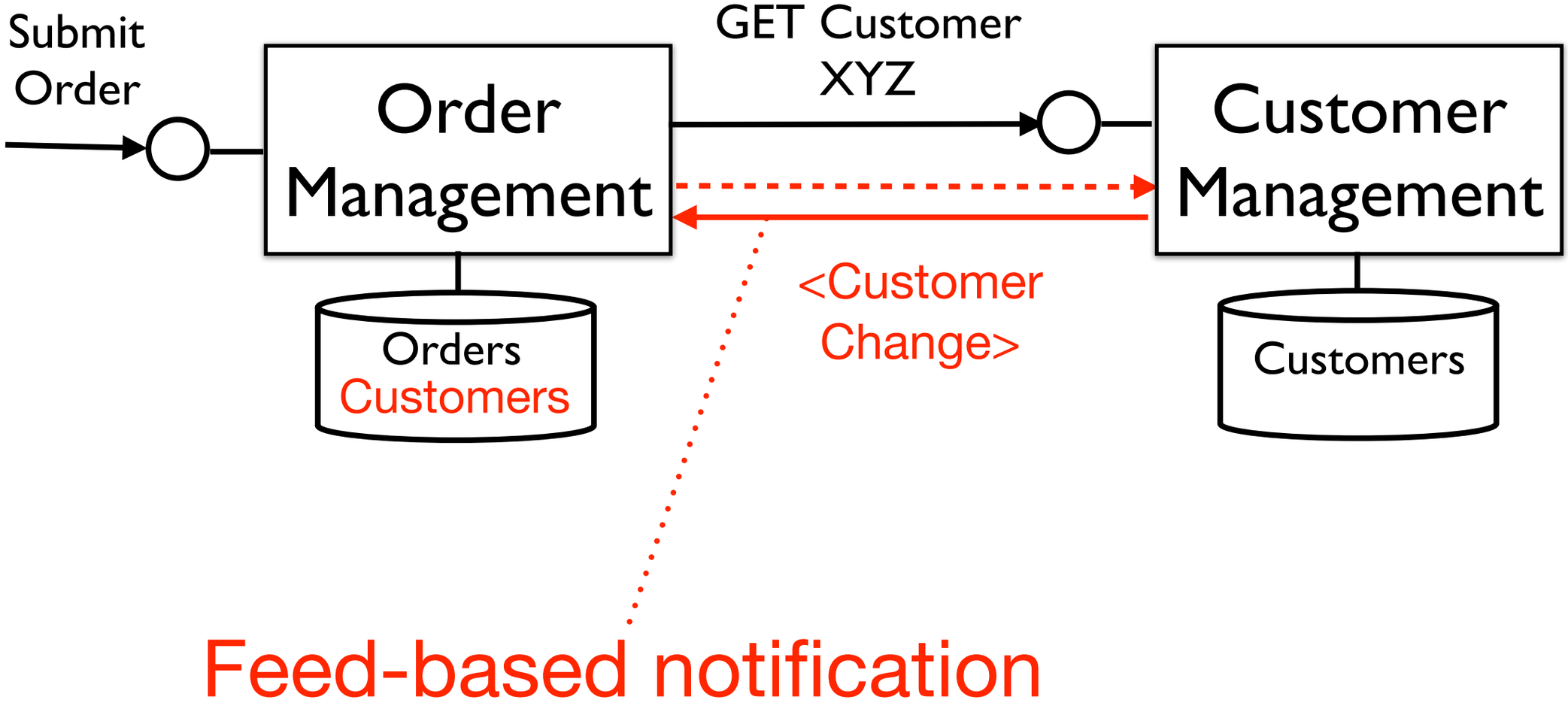






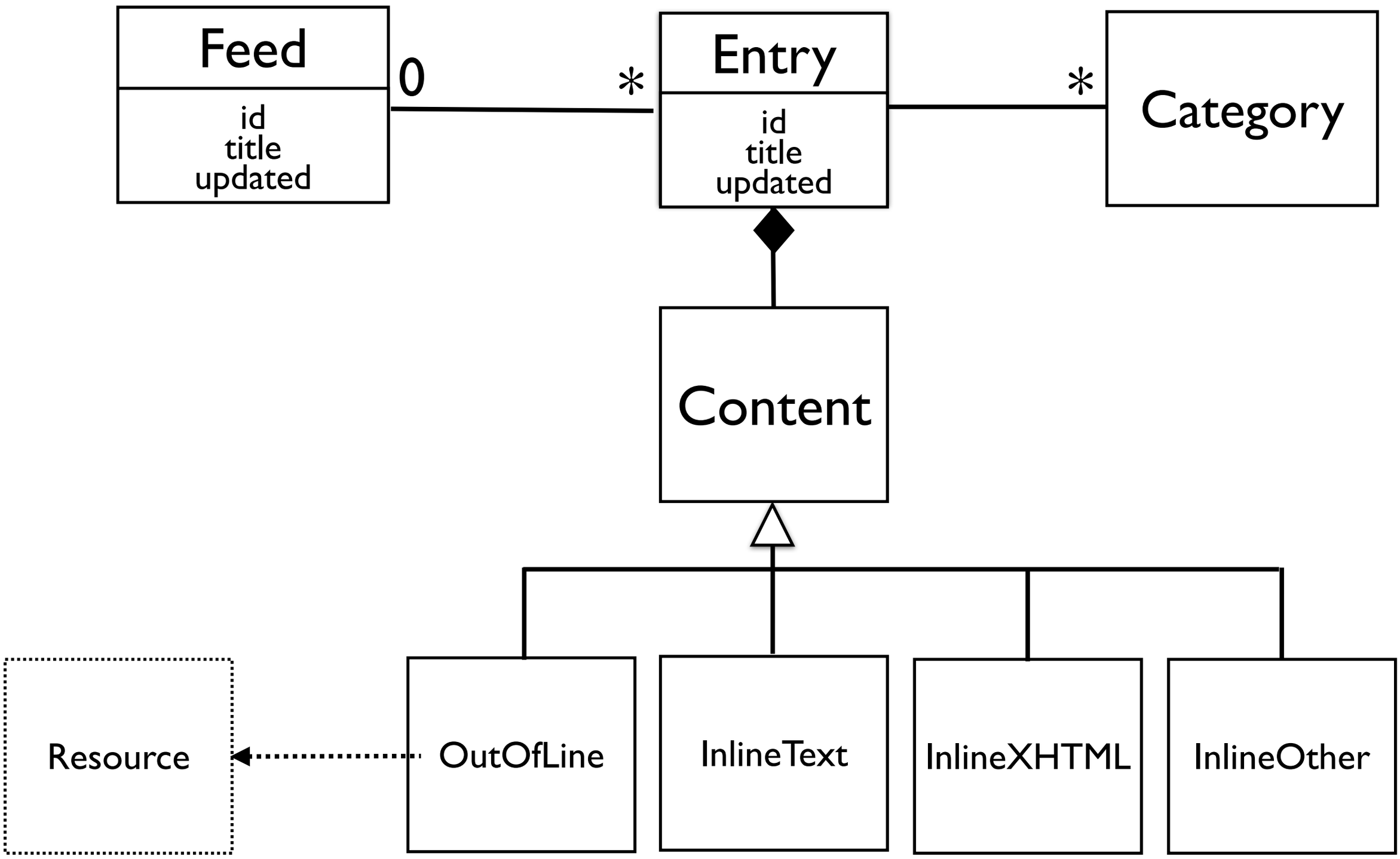






Feed-based notification

Atom Model



Security

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL
HTTPS

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL
HTTPS
REST

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL
HTTPS
REST

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

WSS

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL
HTTPS
REST

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

WSS
XML

Transport-based

encrypt communication channel

protection while in transit

fast, efficient, wide-spread

not end-to-end

not persistent

SSL
HTTPS
REST

Message-based

encrypt/sign individual messages

indefinite protection

Slow, inefficient, scarcely used

end-to-end

persistent

WSS
XML
WS-*

HTTP Security

HTTP Security

Extensible HTTP Authentication Mechanism

HTTP Security

Extensible HTTP Authentication Mechanism

HTTP + SSL + Basic Auth

HTTP Security

Extensible HTTP Authentication Mechanism

HTTP + SSL + Basic Auth

OpenID

HTTP Security

Extensible HTTP Authentication Mechanism

HTTP + SSL + Basic Auth

OpenID

OAuth

HTTP Security

Extensible HTTP Authentication Mechanism

HTTP + SSL + Basic Auth

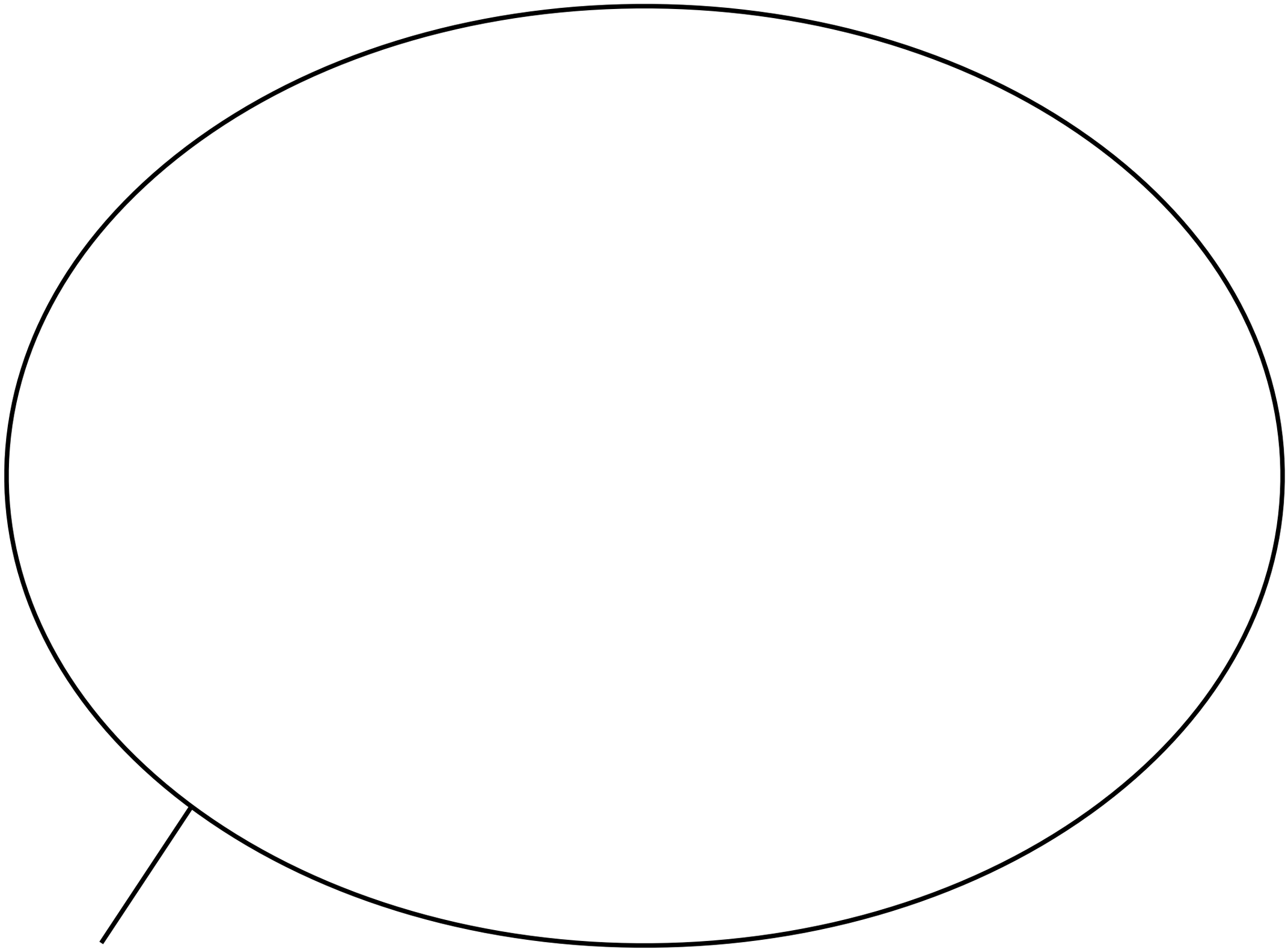
OpenID

OAuth

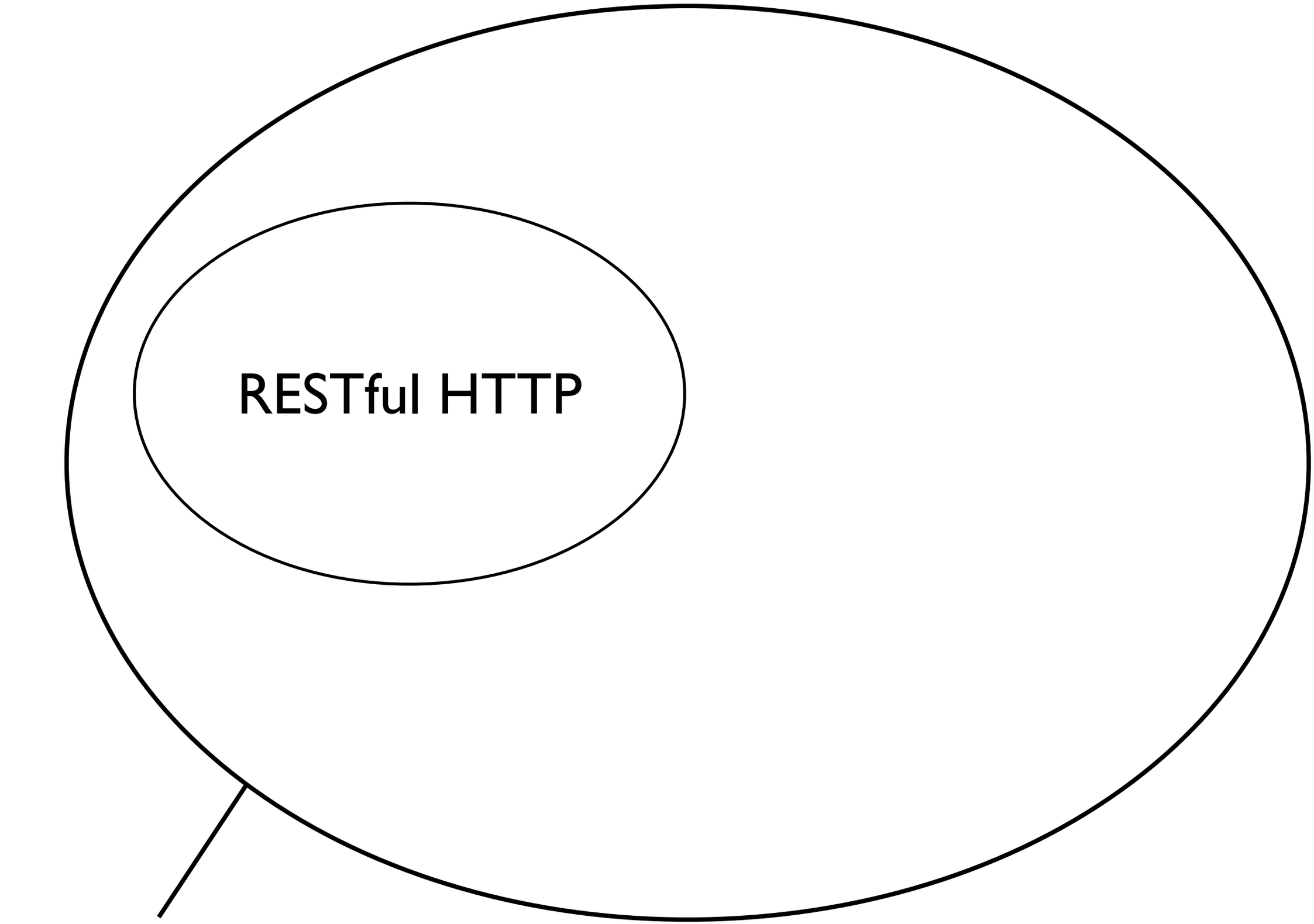
HMAC

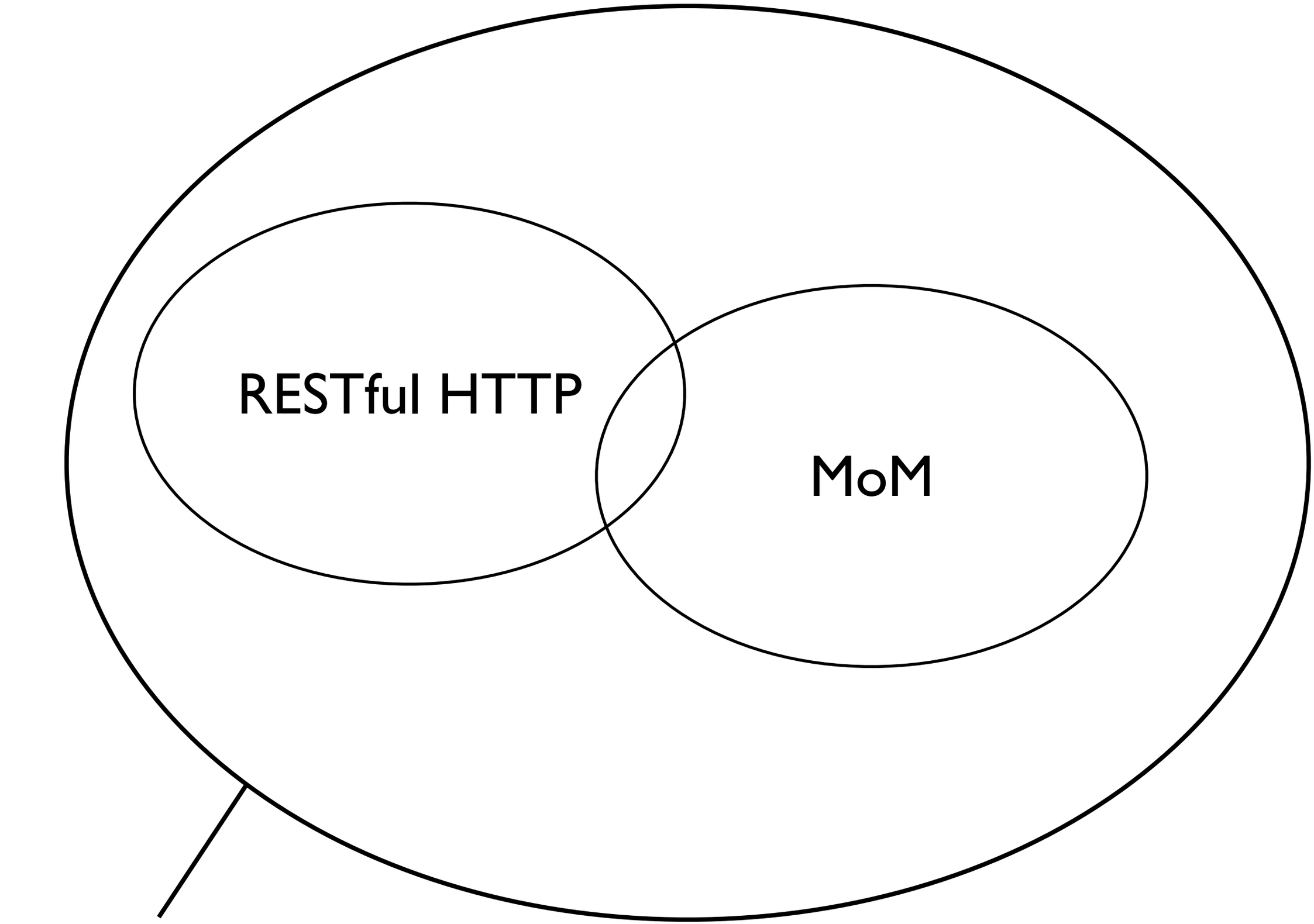
Mixing RESTful HTTP w/ WS-*

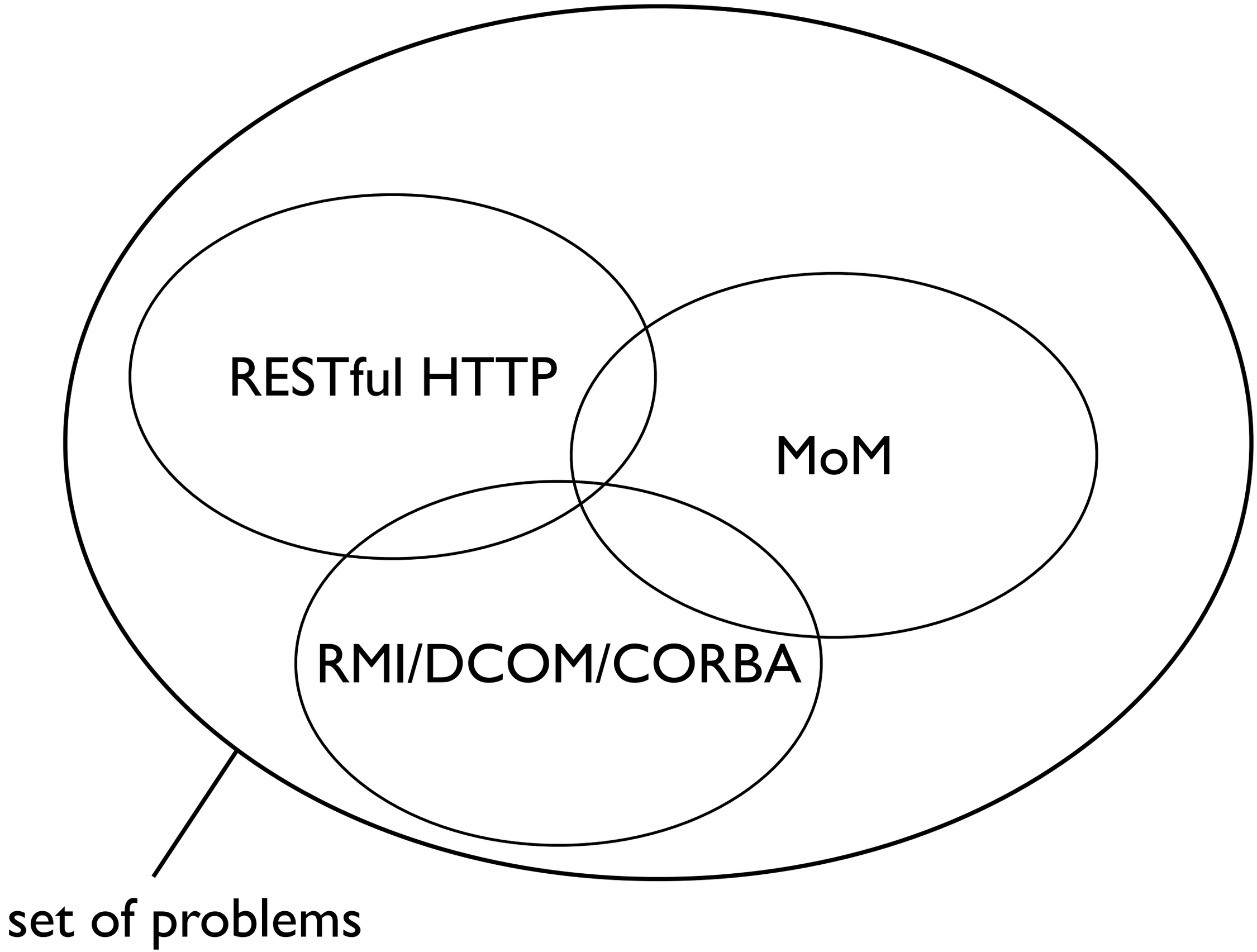
Disclaimer first

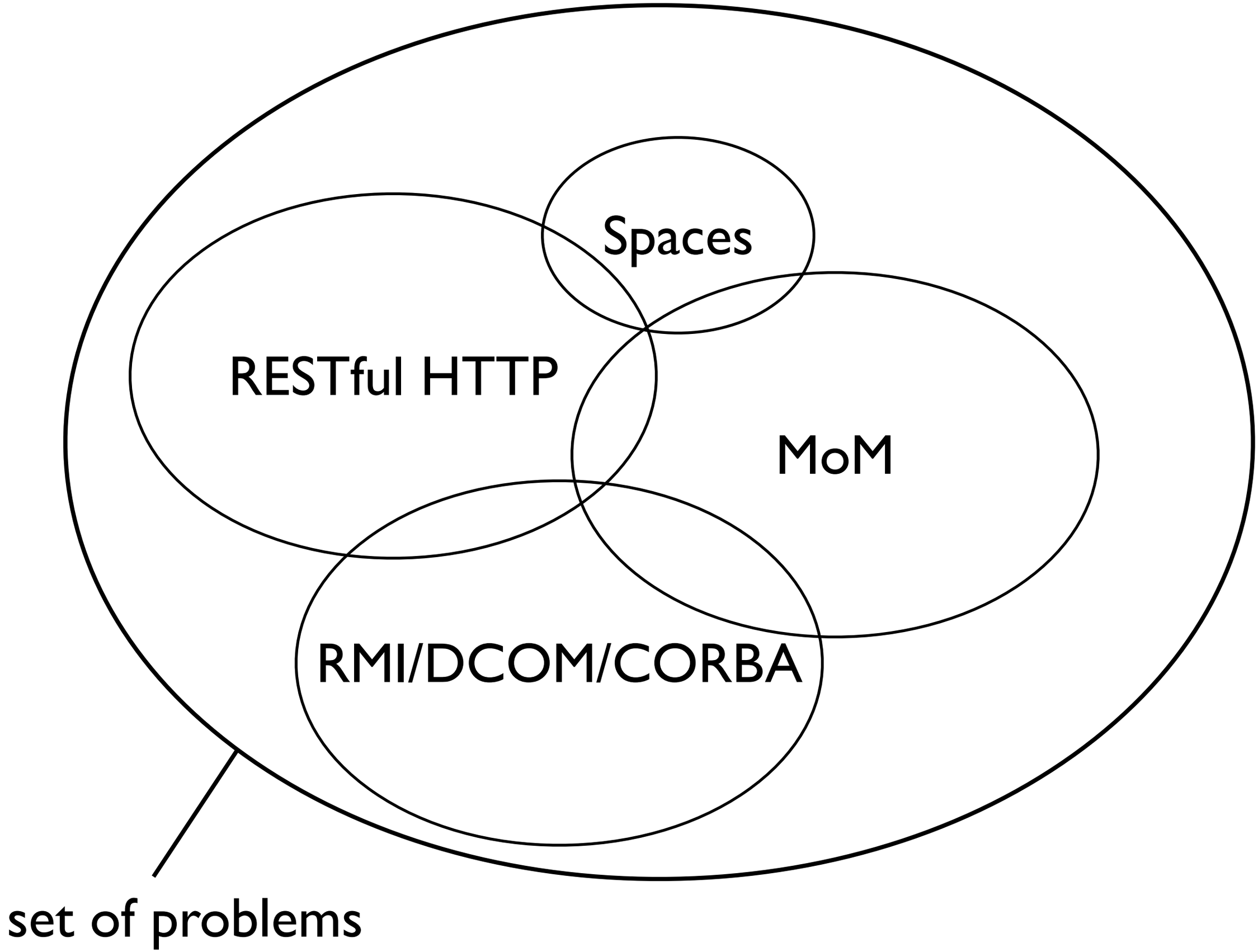


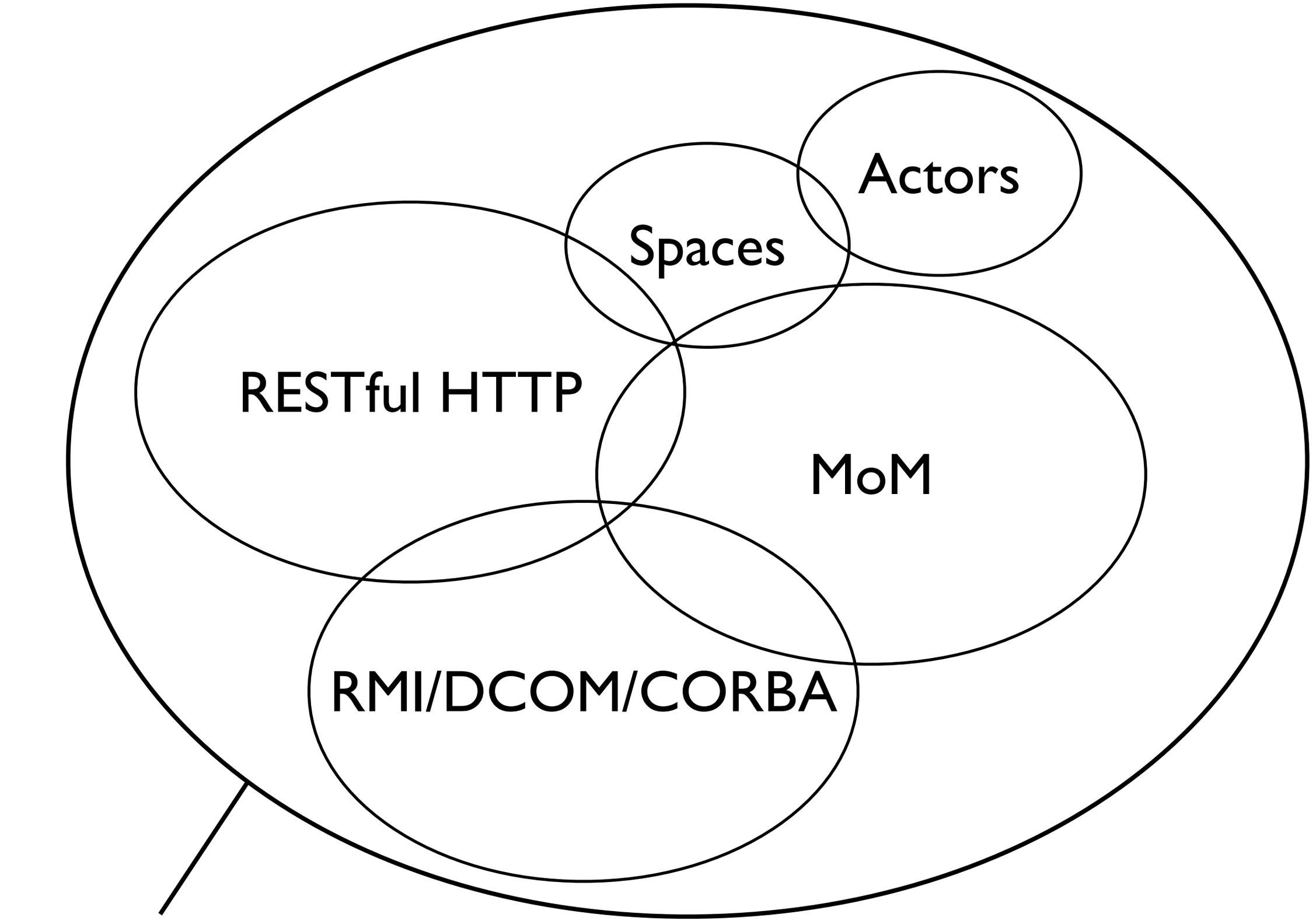
set of problems

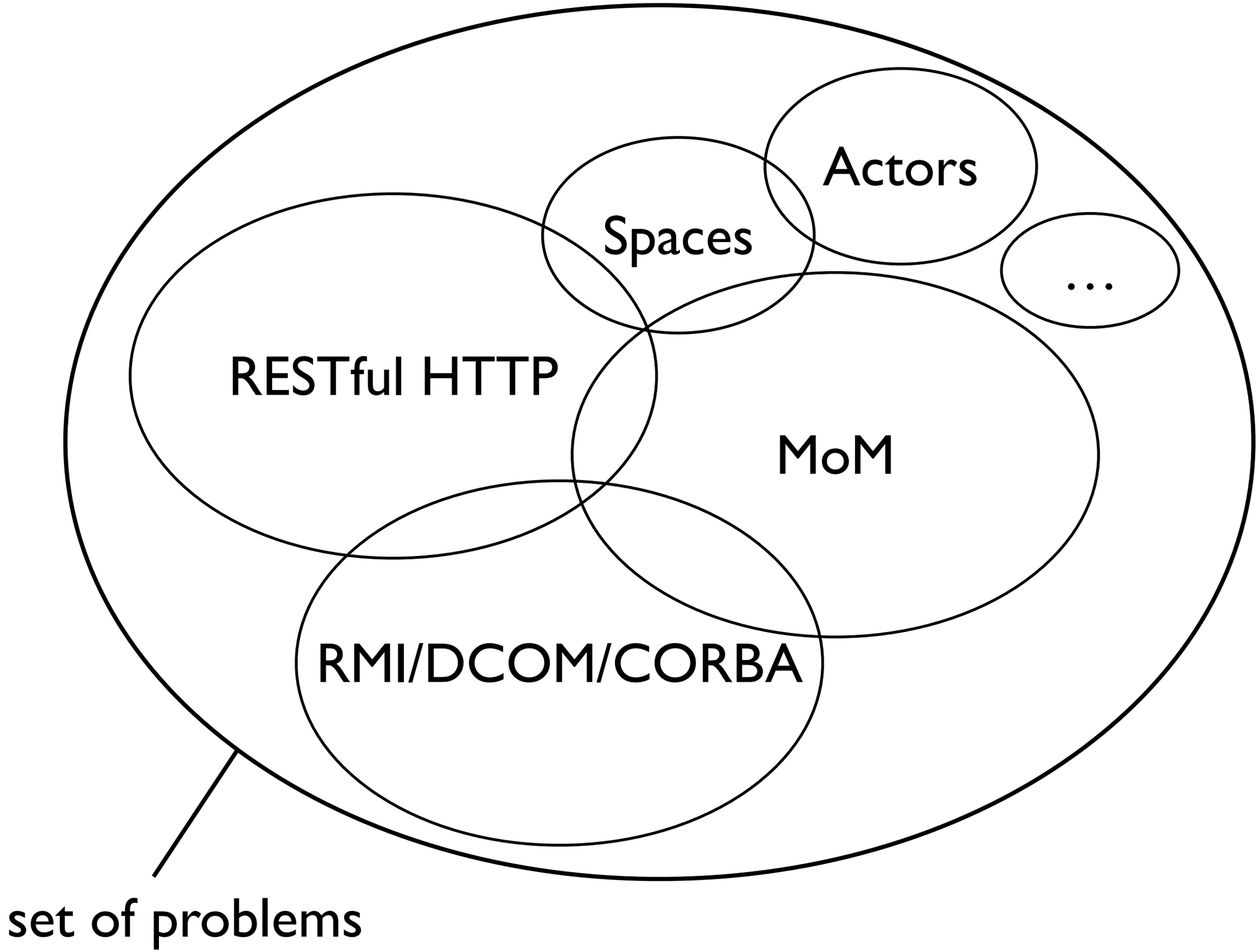


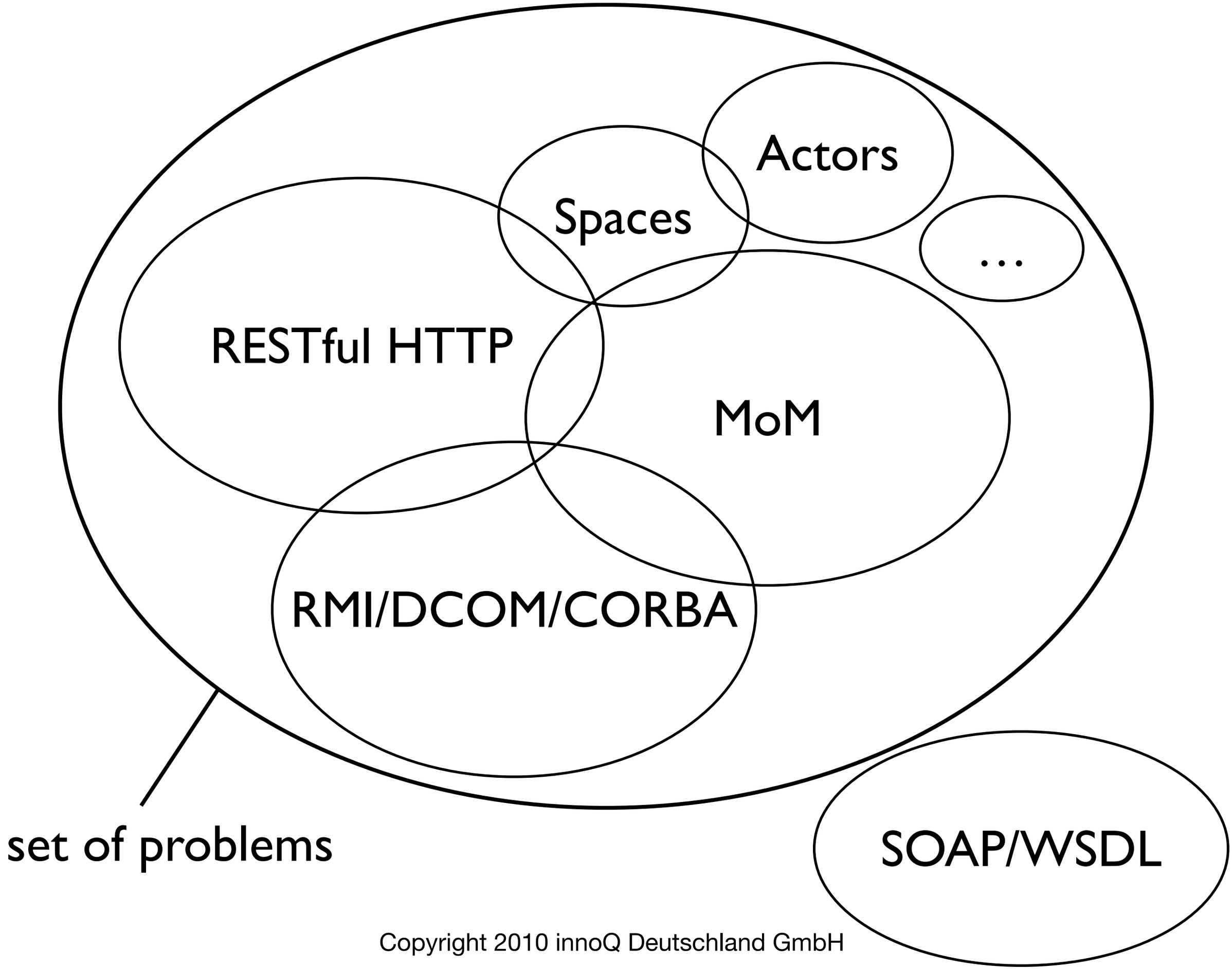












The SOAP/WSDL Problem

The SOAP/WSDL Problem

Each application is different

The SOAP/WSDL Problem

Each application is different

Each application requires its own protocol

The SOAP/WSDL Problem

Each application is different

Each application requires its own protocol

Need to learn a new API *every single time*

The SOAP/WSDL Problem

Each application is different

Each application requires its own protocol

Need to learn a new API *every single time*

WSDL as formal approach *for syntax only*

The SOAP/WSDL Problem

Each application is different

Each application requires its own protocol

Need to learn a new API *every single time*

WSDL as formal approach *for syntax only*

Separation of application and *metadata*

Anatomy of a WSDL File

	XML Schema
	Message Definitions
	Operation Names, Input, Output
	Meaningless Legacy
	Address Info

Anatomy of a WSDL File

80%	XML Schema
	Message Definitions
	Operation Names, Input, Output
	Meaningless Legacy
	Address Info

Anatomy of a WSDL File

80%	XML Schema
2%	Message Definitions
	Operation Names, Input, Output
	Meaningless Legacy
	Address Info

Anatomy of a WSDL File

80%	XML Schema
2%	Message Definitions
5%	Operation Names, Input, Output
	Meaningless Legacy
	Address Info

Anatomy of a WSDL File

80%	XML Schema
2%	Message Definitions
5%	Operation Names, Input, Output
10%	Meaningless Legacy
	Address Info

Anatomy of a WSDL File

80%	XML Schema
2%	Message Definitions
5%	Operation Names, Input, Output
10%	Meaningless Legacy
3%	Address Info

SOAP/WSDL

XML Schema
Message Definitions
Operation Names, Input,
Meaningless Legacy
Address Info

RESTful HTTP

SOAP/WSDL

XML Schema
Message Definitions
Operation Names, Input,
Meaningless Legacy
Address Info

RESTful HTTP

XML Schema *(If you care for it)*

SOAP/WSDL

XML Schema
Message Definitions
Operation Names, Input,
Meaningless Legacy
Address Info

RESTful HTTP

XML Schema *(If you care for it)*
GET, PUT, POST, DELETE

SOAP/WSDL

XML Schema
Message Definitions
Operation Names, Input,
Meaningless Legacy
Address Info

RESTful HTTP

XML Schema *(If you care for it)*
GET, PUT, POST, DELETE
URIs

SOAP/WSDL

XML Schema

Message Definitions

Operation Names, Input,

Meaningless Legacy

Address Info

“Informal” Documentation
(Word, PDF, HTML, ...)

RESTful HTTP

XML Schema
(If you care for it)

GET, PUT, POST, DELETE

URIs

SOAP/WSDL

XML Schema

Message Definitions

Operation Names, Input,

Meaningless Legacy

Address Info

“Informal” Documentation
(Word, PDF, HTML, ...)

RESTful HTTP

XML Schema
(If you care for it)

GET, PUT, POST, DELETE

URIs

“Informal” Documentation
(Word, PDF, HTML, ...)

SOAP/WSDL

XML Schema

Message Definitions

Operation Names, Input,

Meaningless Legacy

Address Info

“Informal” Documentation
(Word, PDF, HTML, ...)

RESTful HTTP

XML Schema
(If you care for it)

GET, PUT, POST, DELETE

URIs

Hypermedia

“Informal” Documentation
(Word, PDF, HTML, ...)

http://example.com/someService

SOAP HTTP “Endpoint”

http://example.com/someService

SOAP HTTP “Endpoint”

http://example.com/someService

Resource

Order

Customer

Fulfilment

Shipment

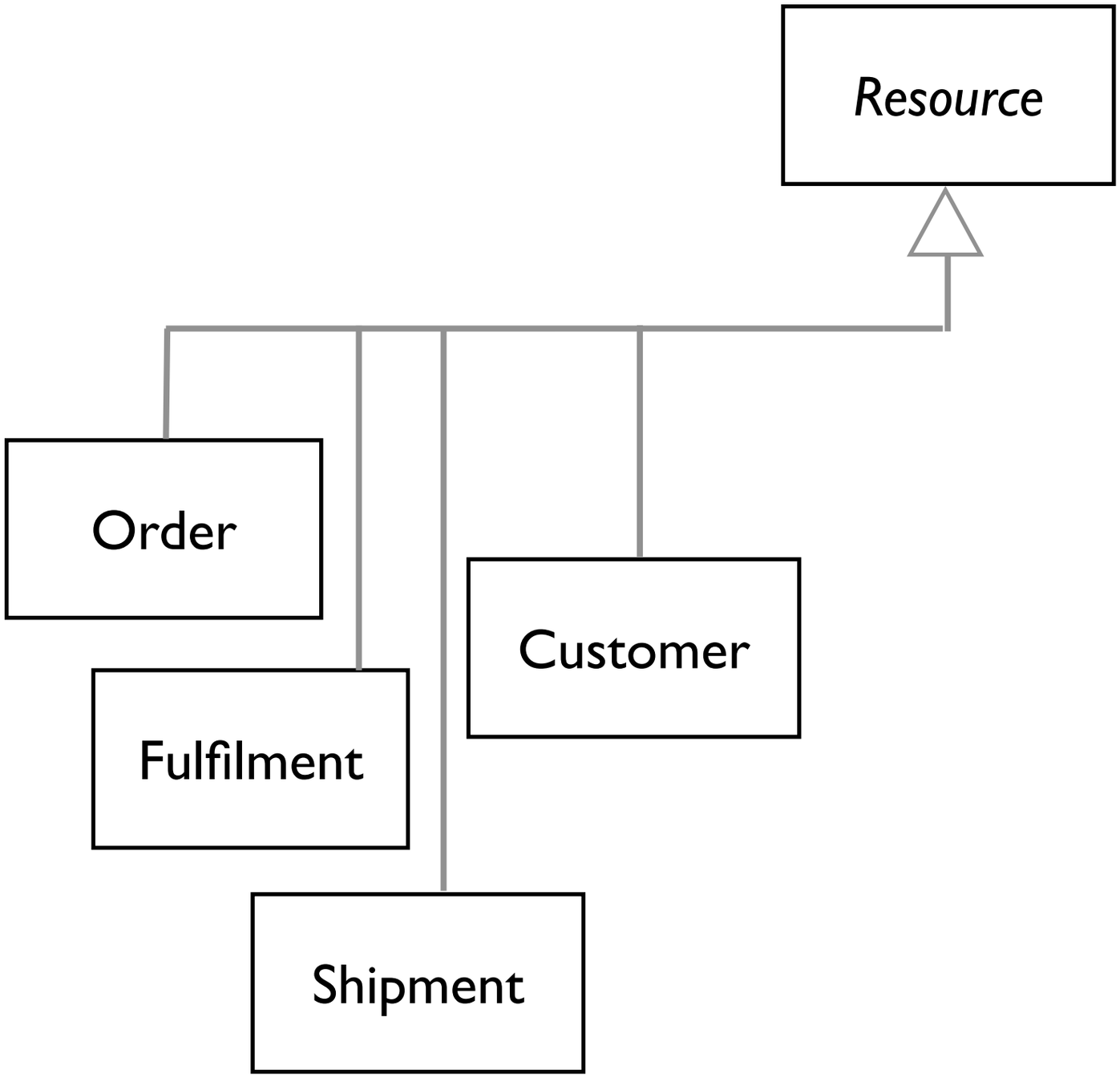
Resource

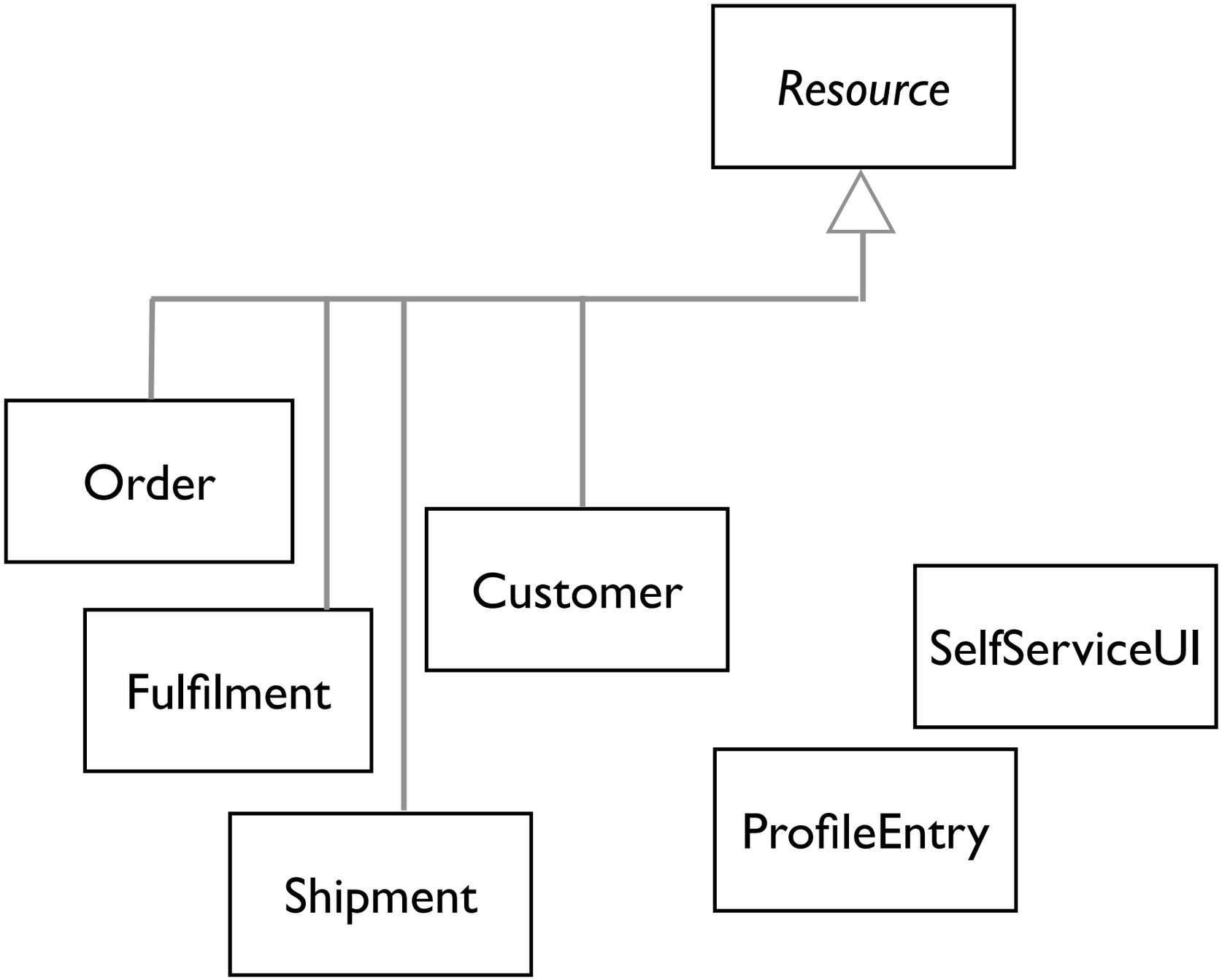
Order

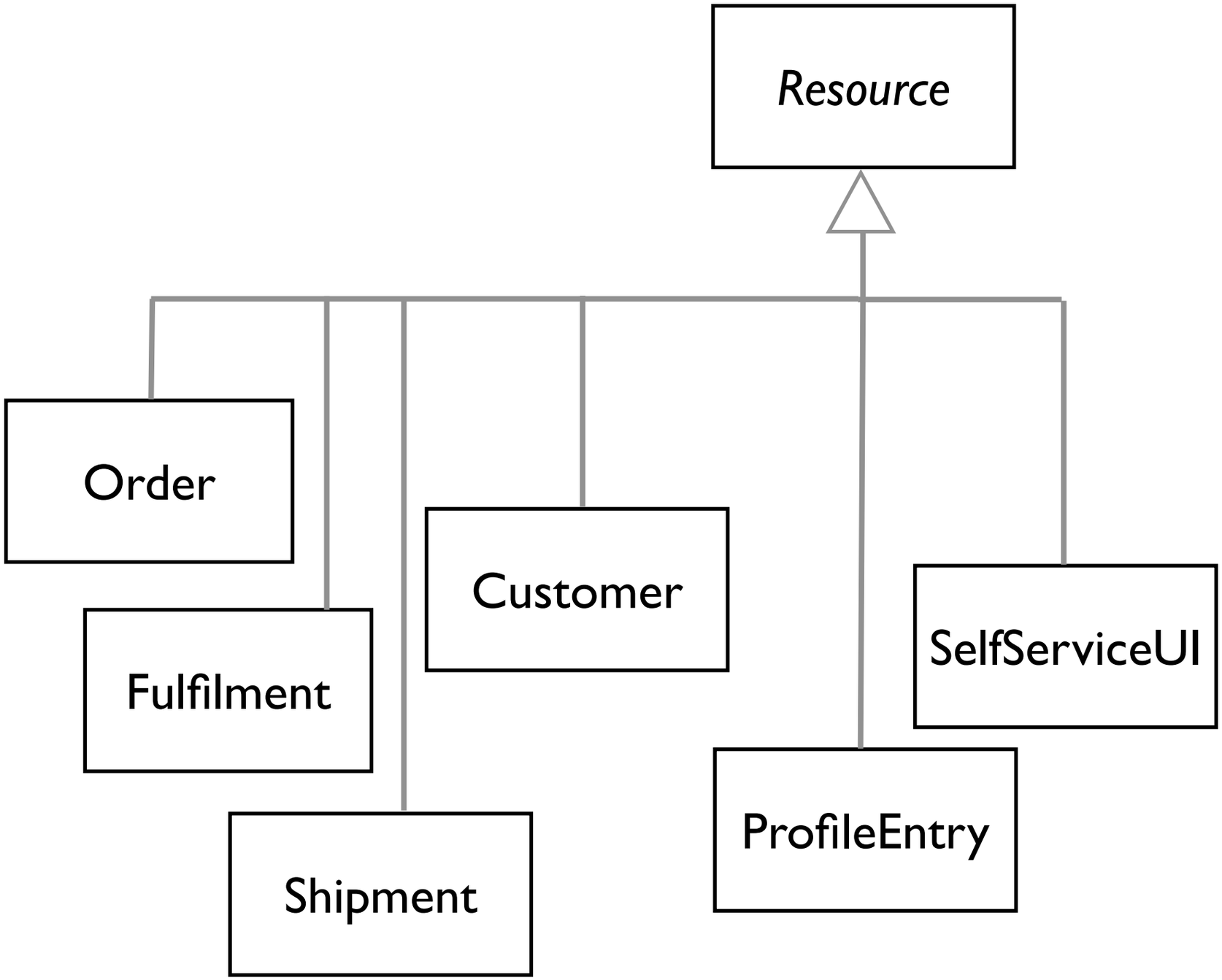
Customer

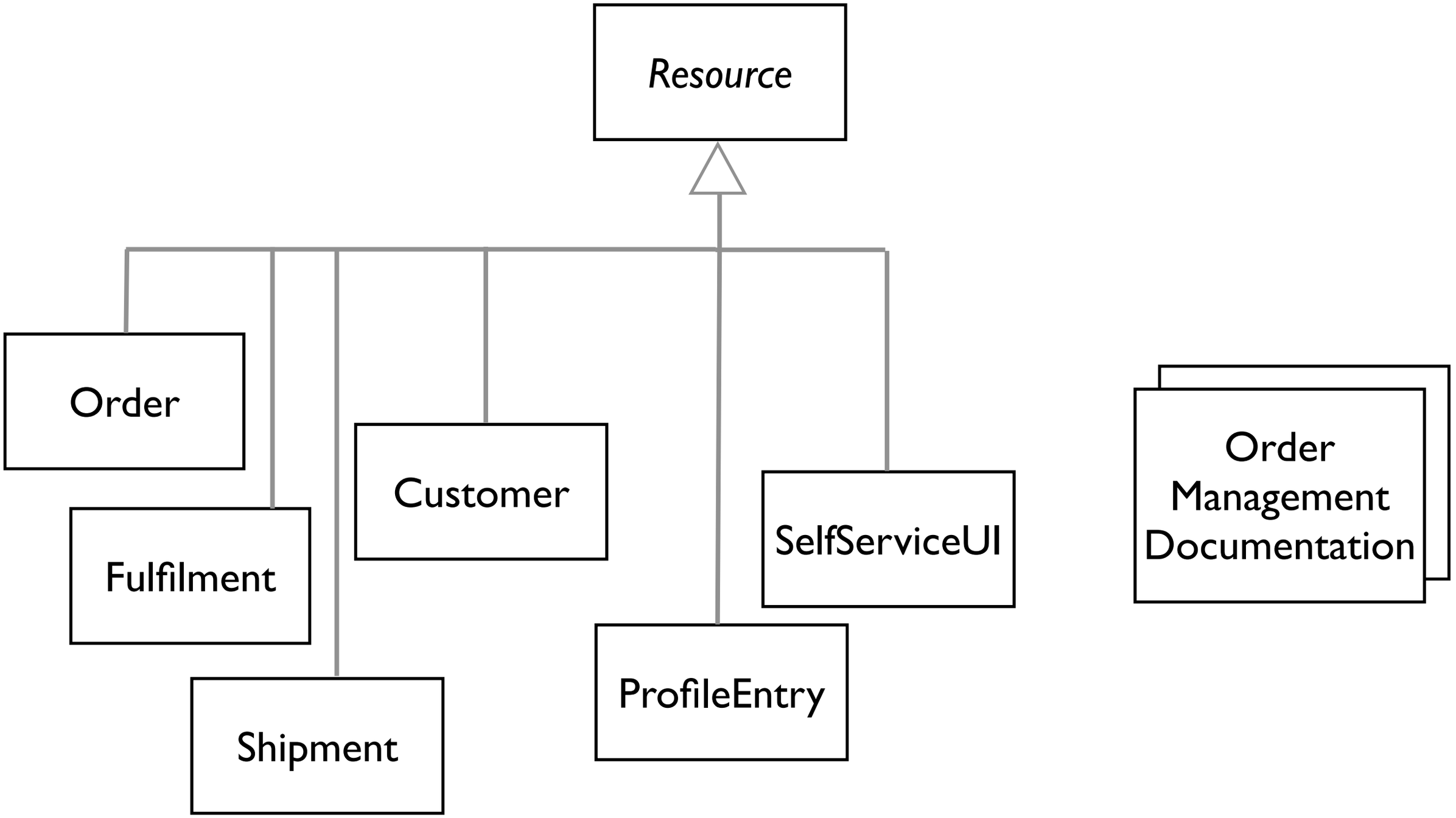
Fulfilment

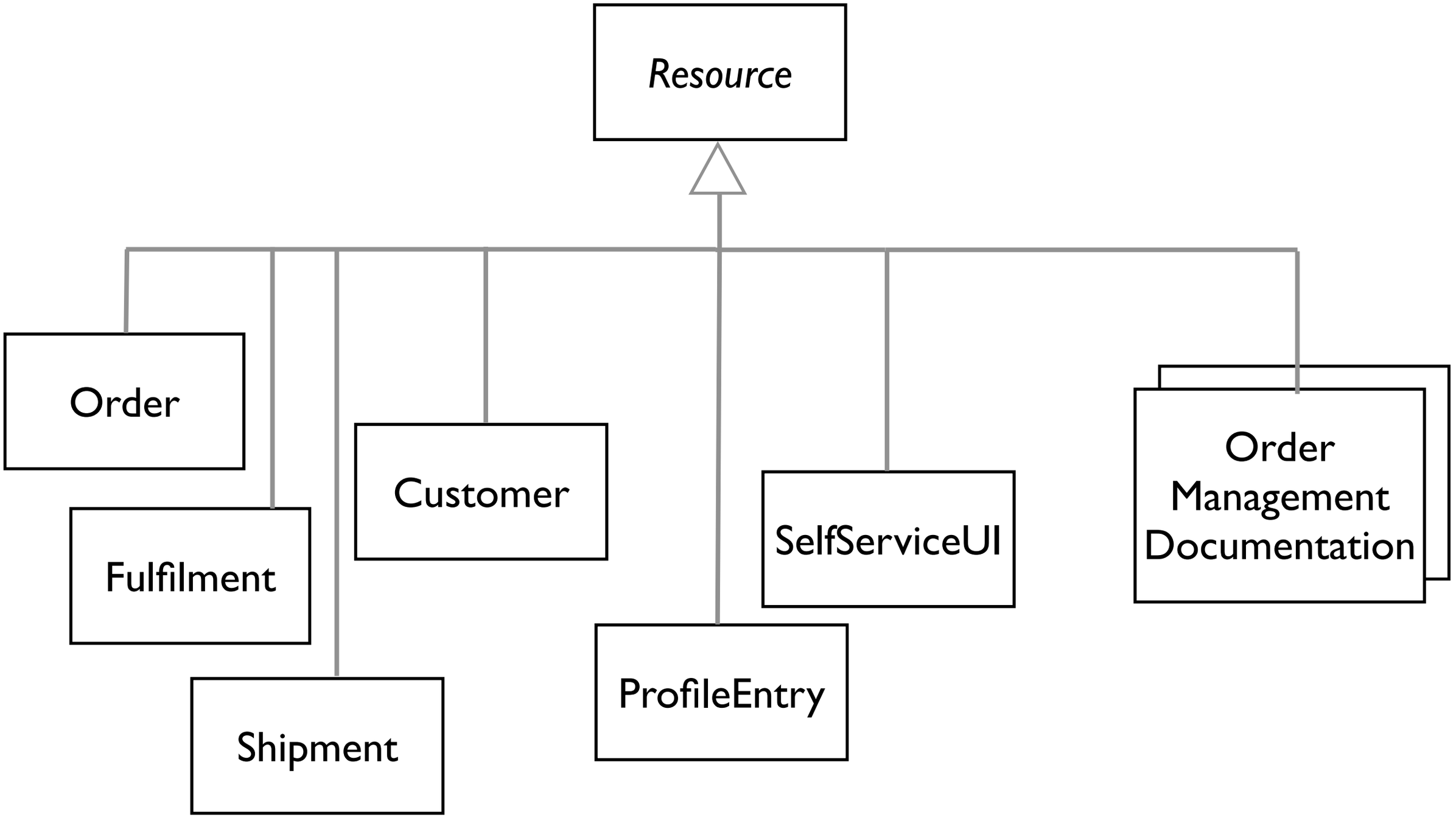
Shipment

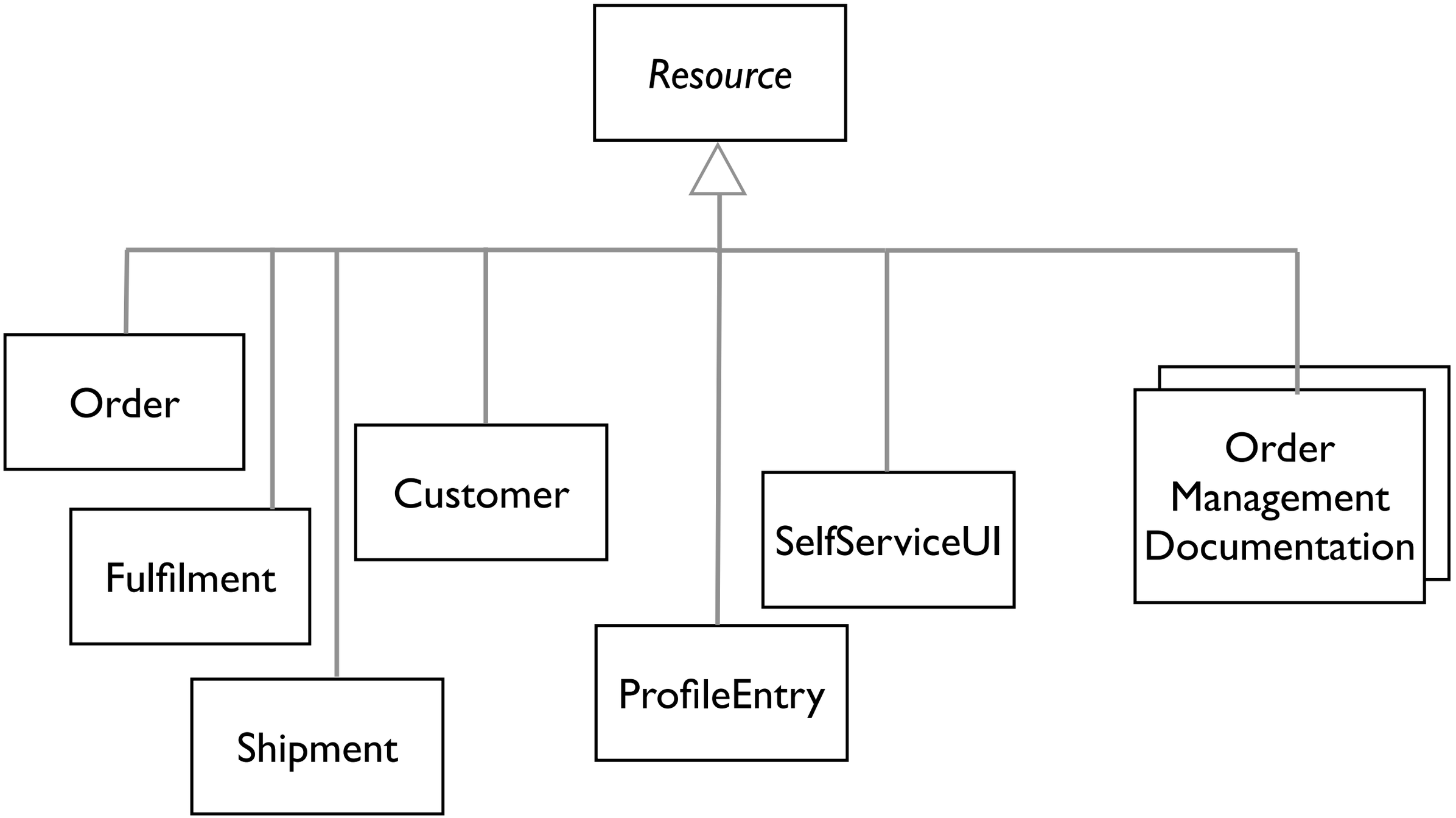




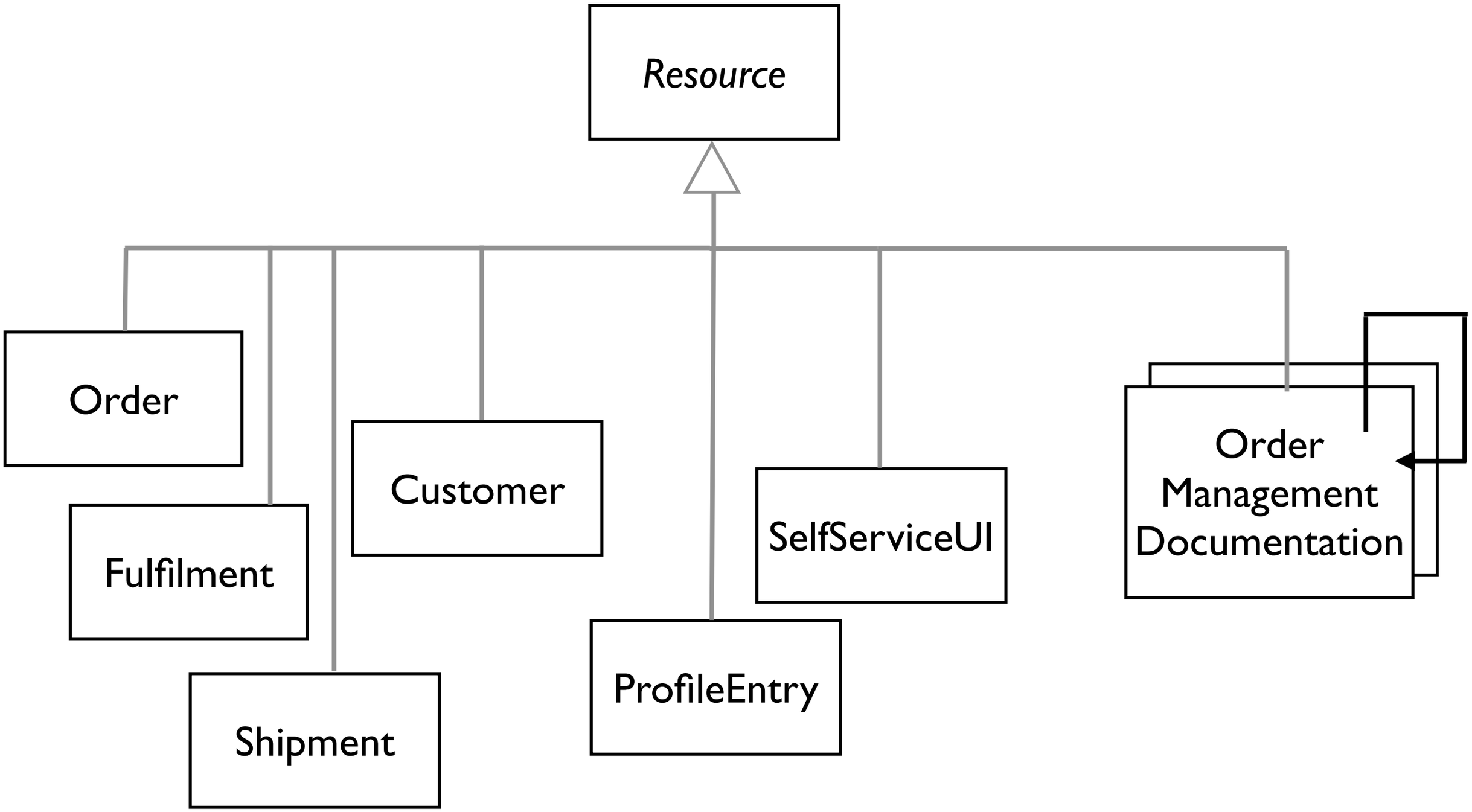




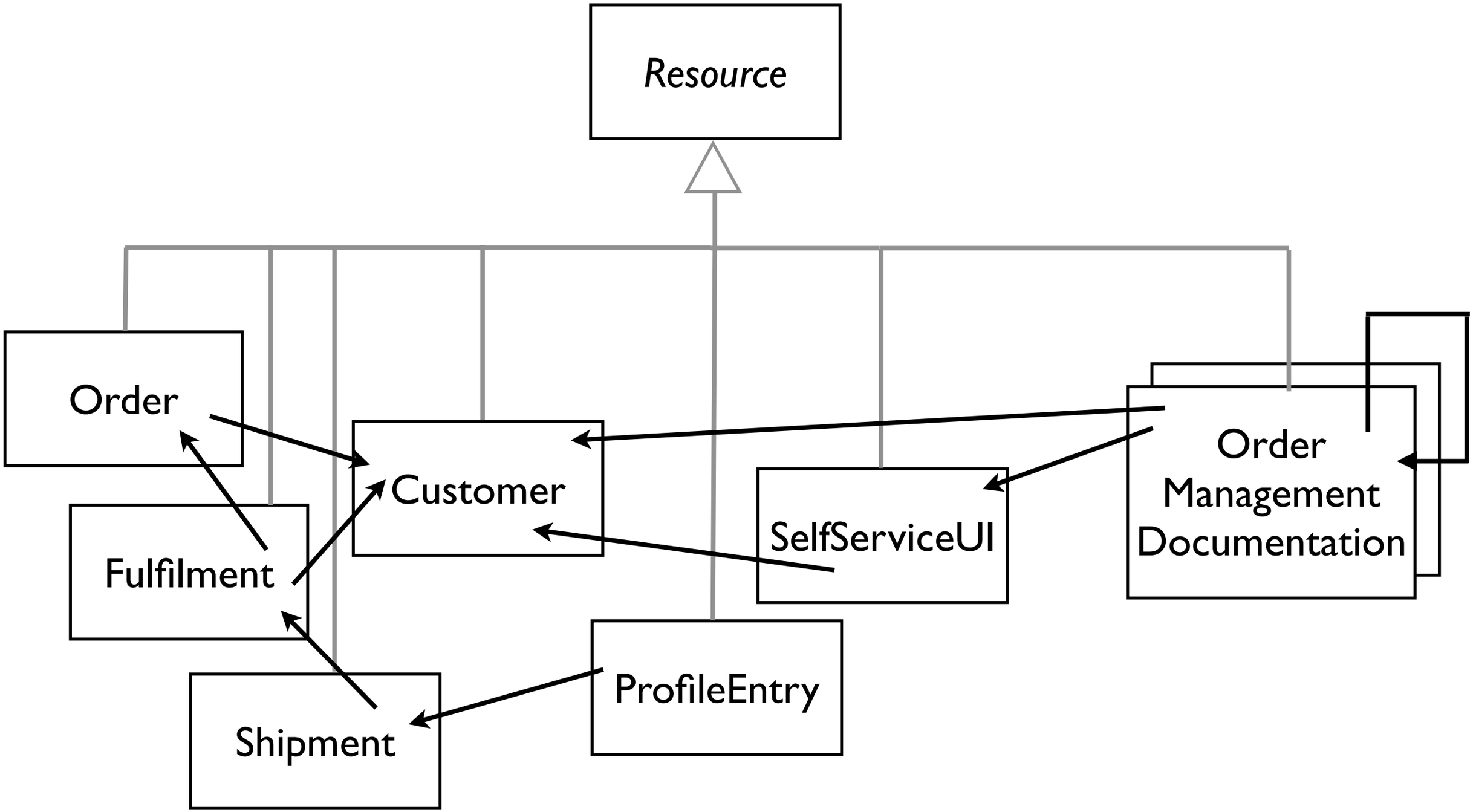




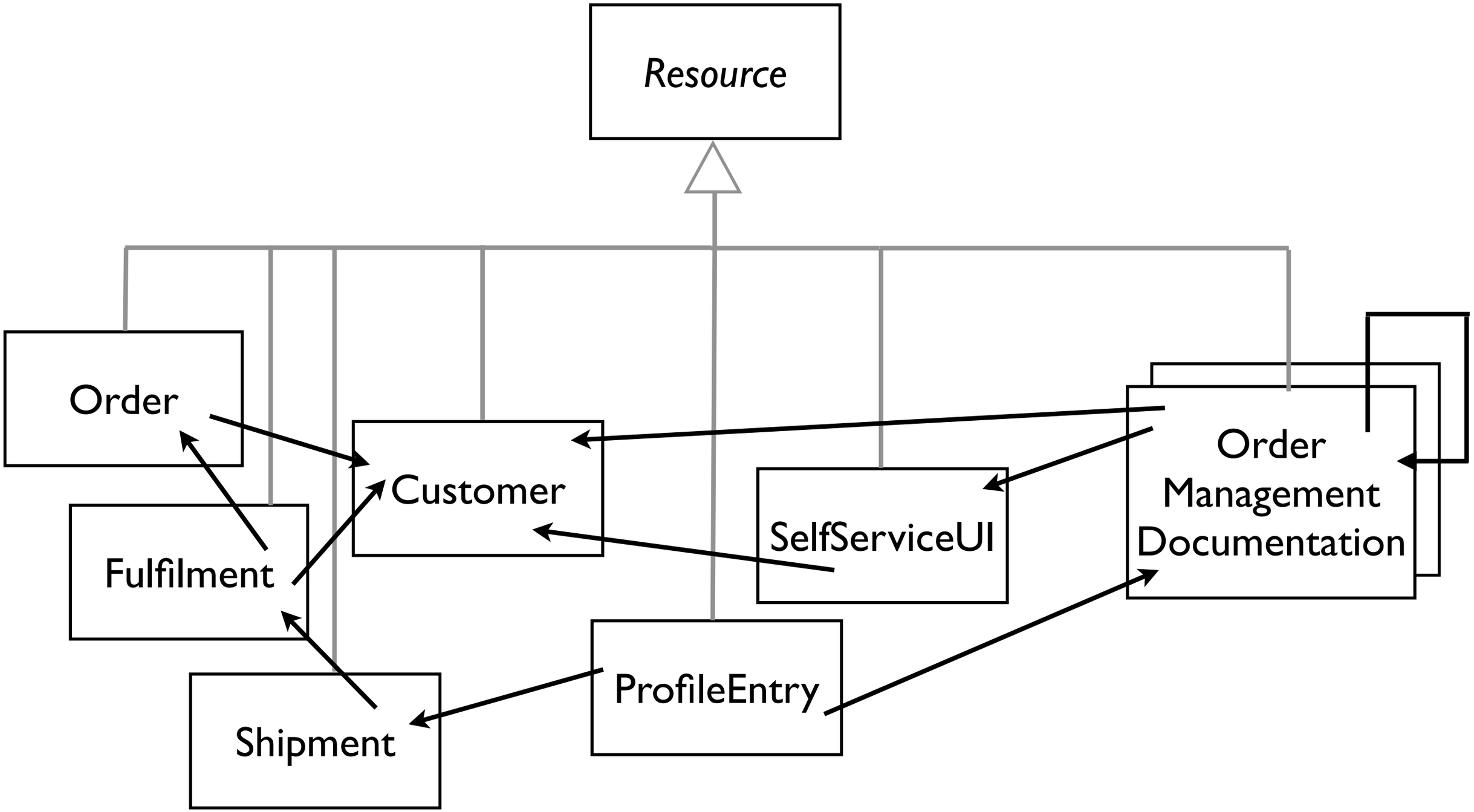
Resources All The Way Down



Resources All The Way Down



Resources All The Way Down



Resources All The Way Down

EJB 1.x

JDBC

EJB 1.x

JDBC

Enterprise-compliant

EJB 1.x

Enterprise-compliant

Complex

JDBC

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

JDBC

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

Simple

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

Simple

Lightweight

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

Simple

Lightweight

Integrated

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

Simple

Lightweight

Integrated

SOAP

EJB 1.x

Enterprise-compliant

Complex

Heavyweight

Managed

JDBC

Straightforward

Simple

Lightweight

Integrated

SOAP

Plain HTTP

SOAP

Plain HTTP

SOAP

Legacy Access

Plain HTTP

SOAP

Legacy Access

Transactional API

Plain HTTP

SOAP

Legacy Access

Transactional API

Processing-centric

Plain HTTP

SOAP

Legacy Access

Transactional API

Processing-centric

Secured messages

Plain HTTP

SOAP

Legacy Access

Transactional API

Processing-centric

Secured messages

Politically motivated

Plain HTTP

SOAP

Legacy Access
Transactional API
Processing-centric
Secured messages
Politically motivated

Plain HTTP

Web-related

SOAP

Legacy Access
Transactional API
Processing-centric
Secured messages
Politically motivated

Plain HTTP

Web-related
(Meta-)Data-centric

SOAP

Legacy Access
Transactional API
Processing-centric
Secured messages
Politically motivated

Plain HTTP

Web-related
(Meta-)Data-centric
Document-oriented

SOAP

Legacy Access
Transactional API
Processing-centric
Secured messages
Politically motivated

Plain HTTP

Web-related
(Meta-)Data-centric
Document-oriented
Interoperable

SOAP

Legacy Access
Transactional API
Processing-centric
Secured messages
Politically motivated

Plain HTTP

Web-related
(Meta-)Data-centric
Document-oriented
Interoperable
Scalable

Steps for introducing REST to a SOA Enterprise

#0: Convince Management

YAHOO! DEVELOPER NETWORK

APIs and Web Services

ADVERTISING & BUSINESS SERVICES

APT from Yahoo!

Interact with our advertising platform and ad exchange-based technology.

Y! Search Marketing

Create and manage campaigns, pull reports, set bids and more. All exposed via SOAP-based Web Services.

AUTHENTICATION & SIGN IN

BBAuth

Registered Yahoo! users can sign on for your services without having to complete yet another registration process.

OAuth

Allow users to sign on to your services in a secure and standardized way.

OpenID

Easy delivery of a simplified login experience for your users.

COMMUNICATION

Address Book

Access and update contacts in the Yahoo! Address Book.

Mail

Build applications to list messages, display folders, compose and send messages.

Messenger

Create add-ons with collaborative features that can run inside Yahoo! Messenger.

CONTENT

Finance

RSS feeds deliver timely data on

HotJobs

Post, edit, refresh and delete public

MyBlogLog

Build social networking services

YAHOO! DEVELOPER NETWORK

APIs and Web Services

ADVERTISING & BUSINESS SERVICES

APT from Yahoo!
Interact with our advertising platform and ad exchange-based technology.

Y! Search Marketing
Create and manage campaigns, pull reports, set bids

- Google Apps APIs
- Google Base Data API
- Blogger Data API
- Google Book Search Data API
- Google Calendar Data API
- Google Code Search Data API
- Google Contacts Data API
- Google Documents List Data API
- Google Finance Portfolio Data API
- Google Health Data API
- Google Notebook Data API
- Picasa Web Albums Data API
- Google Spreadsheets Data API
- Webmaster Tools Data API
- YouTube Data API

AUTH

COMM

Access and update contacts in the Yahoo! Address Book.

Build applications to list messages, display folders, compose and send messages.

Create add-ons with collaborative features that can run inside Yahoo! Messenger.

CONTENT

Finance
RSS feeds deliver timely data on

HotJobs
Post, edit, refresh and delete public

MyBlogLog
Build social networking services

YAHOO! DEVELOPER NETWORK

APIs and Web Services

ADVERTISING & BUSINESS SERVICES

APT from Yahoo!
Interact with our advertising platform and ad exchange-based technology.

Y! Search Marketing
Create and manage campaigns, pull reports, set bids

- Google Apps APIs
- Google Base Data API
- Blogger Data API
- Google Book Search Data API
- Google Calendar Data API
- Google Code Search Data API
- Google Contacts Data API
- Google Documents List Data API
- Google Finance Portfolio Data API
- Google Health Data API
- Google Notebook Data API
- Picasa Web Albums Data AP
- Google Spreadsheets Data A
- Webmaster Tools Data API
- YouTube Data API

amazon web services™

Infrastructure Services
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon SimpleDB
- Amazon Simple Storage Service (Amazon S3)
- Amazon CloudFront
- Amazon Simple Queue Service (Amazon SQS)
- AWS Premium Support

Payments & Billing

On-Demand Workforce

Alexa Web Services

Amazon Fulfillment & Associates

Access and update contacts in the Yahoo! Address Book.

Build applications to list messages, display folders, compose and send messages.

CONTENT

Finance
RSS feeds deliver timely data on

HotJobs
Post, edit, refresh and delete public

MyBlogLo
Build social networking services

YAHOO! DEVELOPER NETWORK

APIs and Web Services

ADVERTISING & BUSINESS SERVICES

APT from Yahoo!
Interact with our advertising platform and ad exchange-based technology.

Y! Search Marketing
Create and manage campaigns, pull reports, set bids

Windows Azure

- Google Apps APIs
- Google Base Data API
- Blogger Data API
- Google Book Search Data API
- Google Calendar Data API
- Google Code Search Data API
- Google Contacts Data API
- Google Documents List Data API
- Google Finance Portfolio Data API
- Google Health Data API
- Google Notebook Data API
- Picasa Web Albums Data AP
- Google Spreadsheets Data A
- Webmaster Tools Data API
- YouTube Data API

amazon web services

- **Infrastructure Services**
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon SimpleDB
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon CloudFront
  - Amazon Simple Queue Service (Amazon SQS)
  - AWS Premium Support
- **Payments & Billing**
- **On-Demand Workforce**
- **Alexa Web Services**
- **Amazon Fulfillment & Associates**

AUTH

COMM

Access and update contacts in the Yahoo! Address Book.

Build applications to list messages, display folders, compose and send messages.

Create add- collaborativ features tha run inside Y Messenger

CONTENT

Finance
RSS feeds deliver timely data on

HotJobs
Post, edit, refresh and delete public

MyBlogLo
Build social networking services

“My Internet is bigger than your enterprise.”

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

Google basel II Search Advanced Search Preferences

Web Results **1** - **10** of about **61**

Basel II – Wikipedia - [Translate this page]
Der Terminus **Basel II** bezeichnet die Gesamtheit der Eigenkapitalvorschriften, die vom **Basler** Ausschuss für Bankenaufsicht in den letzten Jahren ...
de.wikipedia.org/wiki/**Basel_II** - 72k - Cached - Similar pages -

Bundesbank - Bankenaufsicht - **Basel II** - [Translate this page]
Basel II - Die neue Baseler Eigenkapitalvereinbarung ... Während der im Jahr 1998 begonnenen Entwicklung des **Basel II** Regelwerkes standen die Aufsicht und ...
www.bundesbank.de/bankenaufsicht/bankenaufsicht_**basel**.php - 27k - Cached - Similar pages -

Bundesbank - Bankenaufsicht - **Basel II** - Säule **2**: Aufsichtliches ... - [Translate this page]
Die drei Saeulen von **Basel 2** Im Rahmen dieser so genannten zweiten Säule, die als integraler Bestandteil des neuen Kapitalakkords gleichberechtigt neben den ...
www.bundesbank.de/bankenaufsicht/bankenaufsicht_**basel**_saeule2.php - 17k - Cached - Similar pages -
More results from www.bundesbank.de »

Basel-II.info - Steuerkanzlei Christian Reichling, Köln - [Translate this page]
Dipl.-Kfm.(FH) Christian Reichling, Steuerberater aus Köln (Cologne) berät Sie gerne in allen Fragen der Bilanzierung und Besteuerung von Gesellschaften ...
www.**basel-ii**.info/ - 12k - Cached - Similar pages -

Definition **Basel 2, Basel II** - [Translate this page]
Ausführliche Informationen zum Thema **Basel II** – neue Bedingungen für Firmenkredite, Rating und Tipps.
www.foerderland.de/353.0.html - 48k - Cached - Similar pages -

Deutscher Industrie- und Handelskammertag / Homepage - [Translate this page]
Der Deutsche Industrie- und Handelskammertag (DIHK) ist die Spitzenorganisation der 80 Industrie- und Handelskammern (IHKs) in Deutschland.
www.dihk.de/inhalt/informationen/news/schwerpunkte/rating/**basel**.html - 6k - Cached - Similar pages -

VersicherungX

Hans Müller

Search

Advanced Search
Preferences

Web Results **1** - **10** of about (

[Versicherungsnehmer: **Hans Müller**](#) - [Translate this page]
Hans Müller, 40880 Ratingen, Kunde seit dem 1.3.2001, letzte Aktualisierung 1.1.2009 **...**
crm.example.com/customers/4711 - 72k - Cached - Similar pages -

[Kfz-Haftpflicht **Hans Müller**](#) - [Translate this page]
Kfz.-Haftpflicht für ME-HM 123, abgeschlossen 1.1.2005, Halter: **Hans Müller**, Audi A4 **...**
bestand.example.com/lv/26621 - 72k - Cached - Similar pages -

[Risiko LV **Anne Müller**](#) - [Translate this page]
Risiko LV Anne Müller, 40880 Ratingen, abgeschlossen 1.1.2004, Begünstiger: **Hans Müller**,
Versicherungssumme: €200.000 **...**
bestand.example.com/lv/26621 - 72k - Cached - Similar pages -

#1: Ensure your Web apps are RESTful

#2: Expose machine-readable information via HTTP GET

"I do think the REST-afarians are missing an opportunity by not driving home the secret sauce that is HTTP GET. […] **GET is one of the most optimized pieces of distributed systems plumbing in the world.** It's an absolute/objective slam dunk. No arguing/evangelism needed IMO. GET is the classic 'the first bag is free' kind of feature a platform builder dreams about."

"I do think the REST-afarians are missing an opportunity by not driving home the secret sauce that is HTTP GET. […] **GET is one of the most optimized pieces of distributed systems plumbing in the world.** It's an absolute/objective slam dunk. No arguing/evangelism needed IMO. GET is the classic 'the first bag is free' kind of feature a platform builder dreams about."

Don Box, Co-inventor of SOAP

#3: Distribute notifications via Atom feeds

#4: Manage Your Metadata with RESTful HTTP

#5: Ship RESTful client libraries

#6: Adopt existing infrastructure

#7: Use WS-* for read/write interactions if politics or legacy force you

#8: Draw your own Conclusions Watching the Adoption of WS-* vs. RESTful HTTP

Q&A

innoQ Deutschland GmbH
Halskestraße 17
D-40880 Ratingen
Phone +49 21 02 77 162-100

innoQ Schweiz GmbH
Gewerbestrasse 11
CH-6330 Cham
Phone +41 41 743 01 11

info@innoq.com · www.innoq.com

Stefan Tilkov, @stilkov
stefan.tilkov@innoq.com
http://www.innoq.com/blog/st/
Phone: +49 170 471 2625