



Hvað er REST?

Örkynning á REST Vefþjónustu
(RESTful HTTP / RESTful Services)

Stefán Baxter
Netlausnir

Hvað er REST?



Þrjár mismunandi skýringar á REST eru í algengar...

Hvað er REST?



1. Stílbrigði við högun (Architectural style)

- Eitt stílbrigði af mörgum sem Roy Fielding setti fram í doktorsritgerð árið 2000
- REST er aðferðafræði frekar en staðall sem tilgreinir ákveðin skilyrði sem lausnir þurfa að uppfylla til að geta talist til REST (RESTful)
- REST útskýrir margt af því sem notað í HTTP og er af mörgum talin eftiráskýring á HTTP
- Vísar á Vefinn sem algengustu útfærsluna af REST

Hvað er REST?



2. Vefurinn notaður rétt

- Kerfis- eða hugbúnaðarhögun (System or Application Architecture)
- Nýtir HTTP/URI og aðra vefstaðla rétt
- Gögn lifa “á Vefnum” frekar að vera flutt um Vefinn (Vefurinn / Internetið)
- Einniga kallað ROA, WOA eða **RESTful HTTP** (eða jafnvel RESTful Services)



Hvað er REST?

3. XML án SOAP

- Sending á XML skilaboðum yfir HTTP
“SOAP án umslags”
- Á eitthvað skillt við WS-*
- Misnotar gjarnan GET/POST til að kalla á föll/stefjur
- Einnig kallað POX (Plain old XML)



Hvað er REST – 1, 2 eða 3?

- Númer 1 er rétt ... því þannig lýsir Roy Fielding REST í doktorsritgerðinni sinni.
- Skilgreining númer 2 er þó ekki fjarri lagi og rauður þráður í þessari kynningun.
- Skilgreining nr. 3 er svo alfarið fyrir þá sem skilja líklega hvorki WS-* lausnir né REST.

Hvaðan er REST komið og hver er þessi Roy?



- Roy Fielding er aðalhöfundur HTTP 1.1 staðalsins sem var gefinn út í sinni endanlegu mynd árið 1999.
- R.F. kynnti REST (**R**epresentational **S**tate **T**ransfer) í ritgerðinni “Architectural Styles and the Design of Network-based Software Architectures“ árið 2000.
- Upprunalega nafn REST var “HTTP Object Model” sem segir eitthvað um hugsunina að baki og er líklega söluvænlegra nafn :o)
- REST þjónustur hafa verið kallaðar “önnur kynslóð vefþjónusta” en í raun má segja að WS-* (Web Services-*) séu “Internetþjónustur” á meðan REST sé í raun fyrsta kynslóð “Vefþjónusta”
- Í dag snýst notkun á REST aðferðafræðinni um að nýta allt sem við höfum lært af Vefnum við að samþætta lausnir á máta sem nýtir alla kosti HTTP og Vefjarins til hins ýtrasta.



Meginmarkmið REST er:

- Skalanleiki og markviss samskipti aðskildra kerfa
- Almennt, nánast algilt, viðmót á upplýsingar
- Aðgreind uppsetning eininga
- Nýting milliliða til að auka hraða, halda uppi öryggi og hjúpa eldri lausnir (Proxy-Caching, Load-Balancing o.s.frv)



5 grunnreglur REST eru:

1. Öll gögn/föll (resources) fá einkvæma gagnaslóð
2. Allir nota sömu grunnaðgerðirnar
3. Getur birtir gögn á mismunandi vegu (Gagnasnið)
4. Hlekkir eru notaðir til að vísa á tengt efni (Hyperlinks)
5. Byggir á stöðulausum samskiptum



1. Öll gögn fá einkvæma gagnaslóð

- URI = Universal Resource Identifier
- URI samanstendur af:
[skema:][//lén][slóð][?fyrirspurn][#gagnahluta]
- Dæmi:
 - <http://www.vis.is>
 - <https://tryggingabanki.vis.is>
 - <http://www.google.com/search?query=vís>
 - http://www.vis.is/Einstaklingar/Tryggingar/#Fjolskyldan_og_heimilid



1. Öll gögn fá einkvæma gagnaslóð

- Gagnaslóðir...:

- ... eru notaðar til að endurspeigla skipulag (Hierarchy)
- ... eiga að lýsa uppbyggingu og vera merkingarbærar enda viðmót
- ... eiga að hafa notagildi og vera heilstæðar einingar frekar en að endurspegla gagnagrunna, gagnatöflur, gagnaraðir eða dálka
- ... innihalda hvorki sagnorð né aðgerðir
 - <http://www.vis.is/vidkiptavinir?action=delete&id=123>
- Gagnaslóðir eru hluti af Vefnum (Universally addressable)
 - ekki þar með sagt að þær séu öllum aðgengilegar (Universally accessible)

- Dæmi:

- | | |
|--------------------------------------|------------------------------------|
| – /hotel | - Listi yfir öll hótél |
| – /hotel/hotelborg | - Ítarupplýsingar fyrir Hótél Borg |
| – /hotel/hotelborg/bokanir | - Listi yfir bókanir |
| – /hotel/hotelborg/bokanir/oafgreitt | - Listi yfir óafgreiddar bókanir |
| – /hotel/hotelborg/bokanir/54 | - Ítarupplýsingar fyrir eina bókun |



1. Öll gögn fá einkvæma gagnaslóð

- URI = Nafnorð og einkvæmir lykklar, ekki sagnorð eða aðgerðanöfn
 - Gott merki um að eitthvað sé að er þegar sögn eða heiti aðgerðar er að þvælast inn í URI slóð.
 - Fyrirspurnarþáttur slóða (?x=y) skal aðeins nota sem færíbreytur inn í stefjur/föll
 - Gagnalón (Proxy/Cache) kerfi styðja fæst síður sem innihalda fyrirspurnarþátt ?id=14
- HTTP inniheldur 4 sagnir og aðgerðir sem eru EINU aðgerðirnar sem rétt er að nota til að vinna með gögn (tvær þess utan til að afla upplýsinga) – en meira um það á eftir
 - Hefðbundnar WS-* vefþjónustur byggja frekar á einu URI sem stendur fyrir “þjónustu” en öll önnur samskipti fara fram með notkun á fyrirspurnarhluta.

2. Hlekkir eru notaðir til að vísa á tengt efni



- Svar sem inniheldur tengla (Hyperlinks)

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

- HTTP getur einnig sagt til um breytta staðsetningu
 - Status 307 – Tímabundin hjáleidd (Temporary redirect)
 - Status 301 – Flutt (Permanently moved)
- Hlekkir geta breyst
- Hlekkir geta vísað á efni óháð staðsetning (ef það er á vefnum er hægt að vísa á það)
- Hlekkir eru t.d. mikilvægur þáttur í því að koma “opnu” efni á framfæri á leitarvélum (þær kunna að elta tengla upp á 10)



3. Allir nota sömu aðgerðirnar

- HTTP skilgreinir fjórar megin aðgerðir (sagnir)
 - GET : Nær í gögn
 - Hættulaust (Safe)
 - Mögulegt að geyma gögn í gagnalóni til að flýta fyrir afhendingu
 - Gögn eru merkt með geymslureglum (pragma-nocache, ETag, Expires o.fl.)
 - PUT : Skráir eða uppfærir gögn með uppgefnu auðkenni
 - Uppfærir gögnin sem gagnaslóðin vísar á með því efni sem var sent
 - Stofnar nýja færslu ef slóðinn er viðurkennd og gögn eru ekki til fyrir
 - Sjálfvalda (Idempotent) - $f(x) = f(f(x))$ - hægt að kalla á oft án afbrigða
 - Til dæmis þar til ásættanlegt svar fæst (Status 201 created)
 - POST : Stofnar nýja skráningu, eða undirskráningu, án uppgefins auðkennis
 - Location svarhausnum (header) gefur upp varanlega slóð á nýju gögnin (Auðkenni)
 - Ekki sjálfvalda því ný skráning verður til við hvert kall (PUT vs. POST)
 - DELETE : Eyði skráningu
 - Sjálfvalda
 - hægt að kalla á án frekar aukaverkana þar til ásættanlegur svarkóði (200 OK) fæst



4. Getur birtir gögn á mismunandi vegu

- Accept header sér um að panta gagnaframsetningu
 - Accept header inniheldur lista yfir hvaða gagnategundir eru studdar af biðlara, listaðar upp í forgangs röð. Miðlarinn afhendir fyrstu framsetninguna sem hann kann að afhenda.
 - Umbreyting gagna er því mikilvægur partur af REST. Það að geta fengið sömu gögnin matreidd á mismunandi hátt frá sömu gagnaslóðinni er ótrúlega öflugt.
 - Gögn eru alltaf sett fram og afhent með ákveðinni frammsetningu en ekki sem grunngögn og það vísar í fullt nafn REST (**R**epresentational **S**tate **T**ransfer)
- HTML, XML og JSON eru algeng gagnasnið
- Einnig er hægt að nota PDF, JPEG og önnur gagnasnið sem eru “studd” á Vefnum
- Mælt með að þekt gagnasnið séu notuð þó að hægt sé að nota sértæk snið
- MIME tegundamerkingar í haus (header) svars gefur til kynna hvaða gagnasnið er notað (application/pdf – text/xml)



4. Getur birtir gögn á mismunandi vegu

- **Dæmi 1**

```
GET /customers/1234
```

```
Host: example.com
```

```
Accept: application/vnd.mycompany.customer+xml
```

- **Skilar:**

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

- **Dæmi 2**

```
GET /customers/1234
```

```
Host: example.com
```

```
Accept: text/x-vcard
```

- **Skilar:**

```
begin:vcard
```

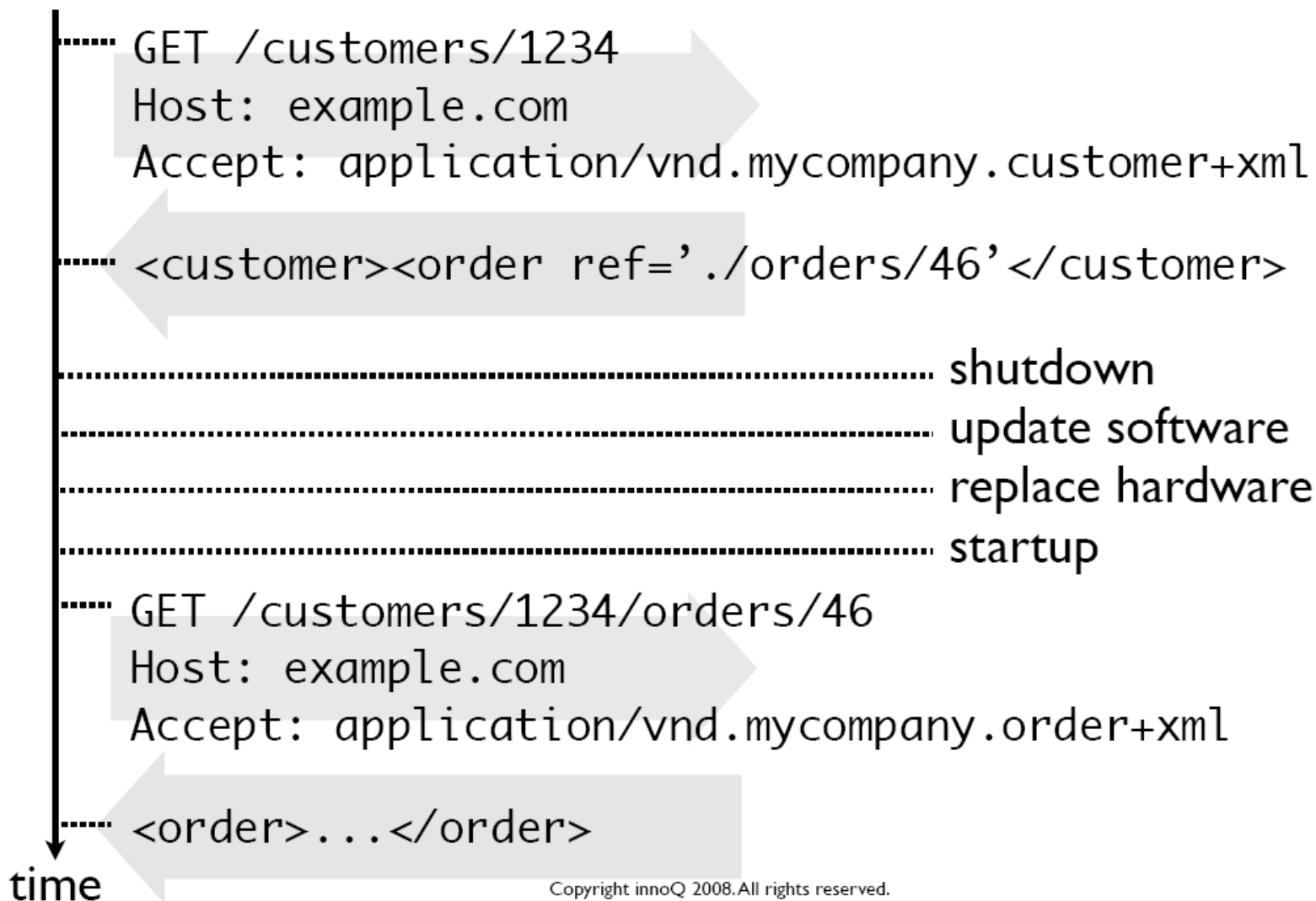
```
...
```

```
end:vcard
```

- Fyrir biðlara sem styðja ekki “Accept” hausa er brugðið á það ráð að setja endingar á slóðir eins og t.d. /customers/1234.xml



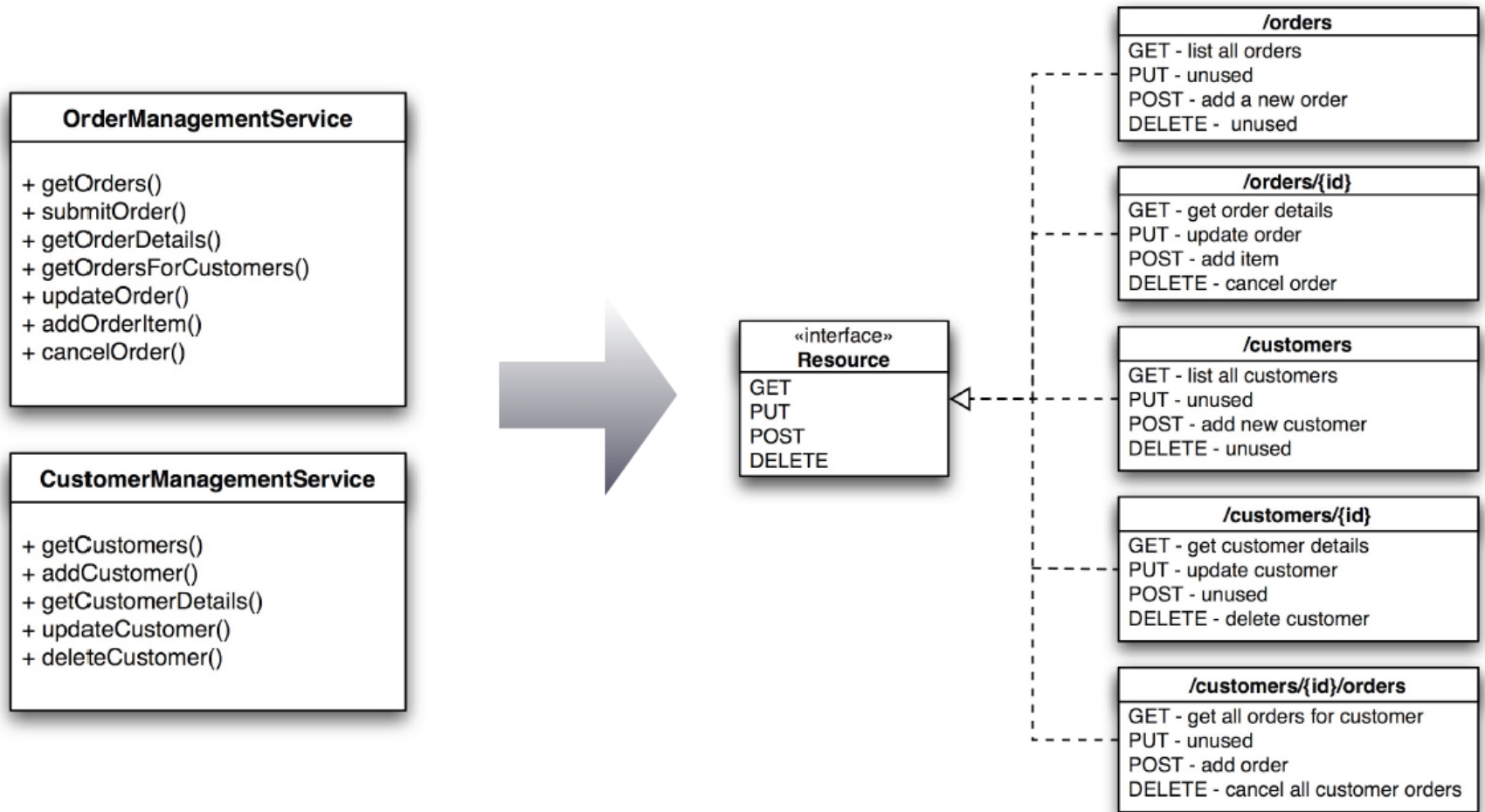
5. Byggir á stöðulausum samskiptum





Afleiðingar

WS* -> REST

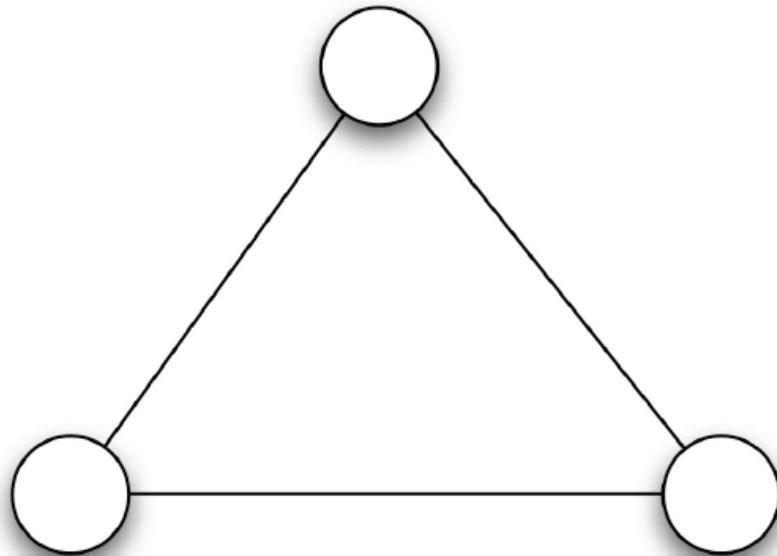


WS* samsetning



many

Data types



Operations

many

Instances

very few
(one per service)

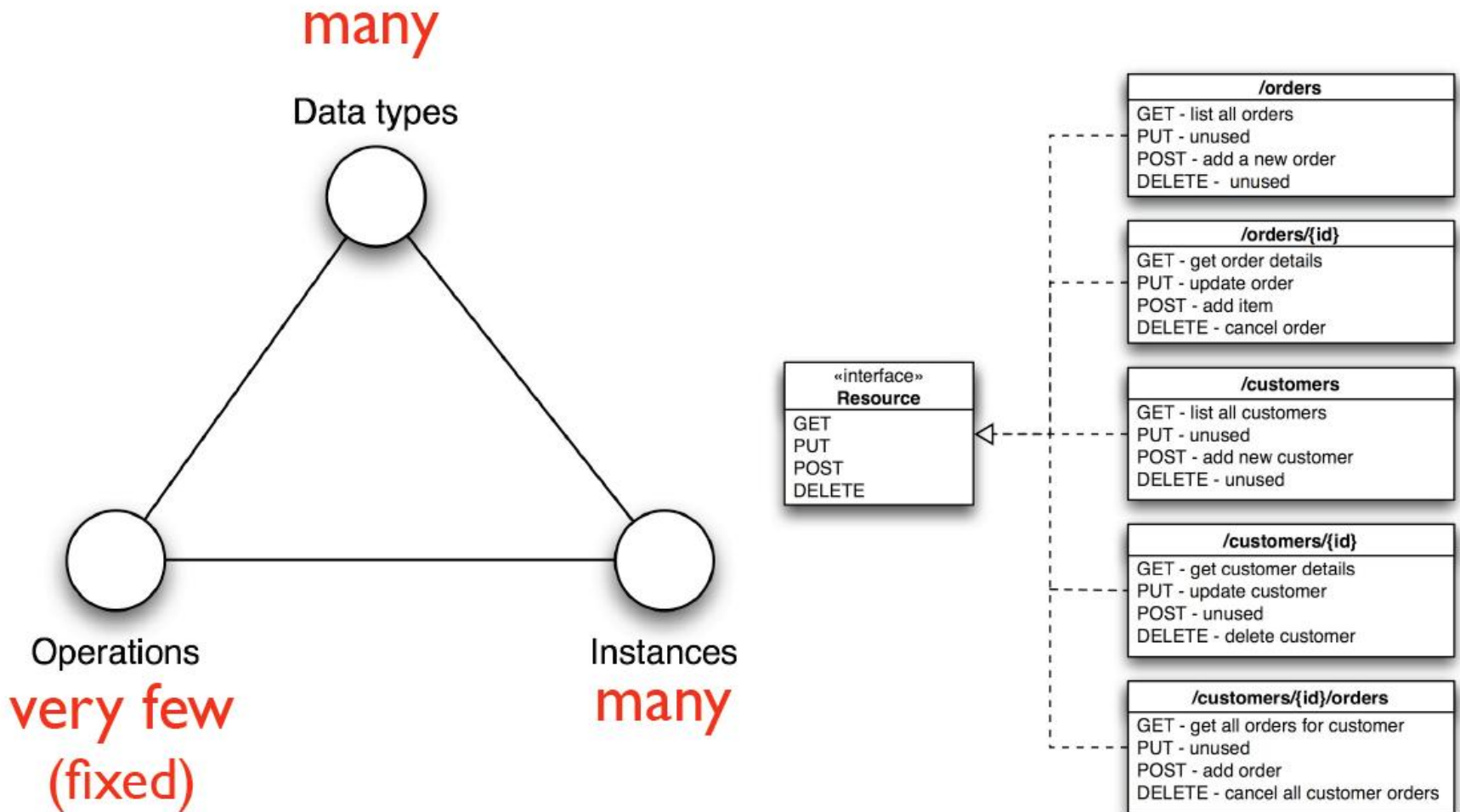
OrderManagementService

```
+ getOrders()
+ submitOrder()
+ getOrderDetails()
+ getOrdersForCustomers()
+ updateOrder()
+ addOrderItem()
+ cancelOrder()
```

CustomerManagementService

```
+ getCustomers()
+ addCustomer()
+ getCustomerDetails()
+ updateCustomer()
+ deleteCustomer()
```

REST samsetning





Þegar HTTP er notað rétt fæst:

- Færri, staðlaðar, aðgerðir:
 - **GET, POST, PUT & DELETE** (HEAD, OPTIONS, TRACE, CONNECT)
- Gagnasniðssamningar (Accepts header) (Content negotiation)
- Gagnasnið er uppgefið með MIME (eins og önnur gögn á vefnum)
- Staðlaðir og merkingarbærir svarkóðar
 - 200 OK, 201 Created, 202 Accepted, 404 Not Found, 500 Internal Server Error, 307 Tímabundin hjáleið, 301 Flutt, 409 Conflict o.fl. o.fl.
- Flytiminni á víð og dreif um Vefinn (eða á fyrirtækjanetinu)
- Stuðningur við hjáleiðir og flutning á gögnum (Redirection)
- Gagnþjöppun (HTTP 1.1 inniheldur zip stuðning)
- Bútun (Chunking) – Ekkert mál að afhenda stórar gagneneiningar
 - Reyndu að afhenda stórt video með SOAP :o)
- Hægt að loka fyrir skrif aðgang á eldveggju o.s.frv með því að loka á allt annað en GET



Af hverju HTTP?

- Mikið um innbyggðan stuðning við HTTP
 - Forritunarmál, stýrikerfi og hugbúnaður – nánast allir kunna HTTP
- Skalast örugglega – spurðu bara vefinn
- Mikið til að þroskuðum lausnum
 - Vefþjónum, proxy þjónum, eldveggjum, eftirlitskerfum, álagsdreifingarkerfum, annálabúnaði og öryggisbúnaði
- Styður XML eins og öll hin gagnasniðin sem eru notuð á vefnum
- Er þroskuð tækni sem hefur sannað sig
- Er application staðall en ekki flutnings- eða samskiptastaðall

Hvað er REST?



Hverjir eru að nota þetta – frekari upplýsingar



Skiptir máli hverjir eru að nota þetta?

- **Amazon.com** er með REST útgáfu af vefverslunarviðmótunum sínum [developer interface](#)
- **eBay** er með REST vefþjónustu [developer interface](#);
- **Facebook** er með REST vefþjónustu [developer interface](#);
- **Yahoo!** er með REST vefþjónustu [developer interfaces](#).
- **Youtube** er með REST vefþjónustu [developer interfaces](#).
- En hugsanlega er þetta bara nýjasta tískufyrirbrigðið

Meira?



- Var þetta allt og sumt?
 - Of lítið um HTTP Header og hvernig þeir eru notaðir t.d. fyrir gagnalón (Cache/Proxy)
 - Nánast ekkert MIME tegundir og auðkenningu gagnasniða
 - Mikilvægi þess að nota stöðluð gagnasnið
 - Of lítið um það hvernig biðlari heldur utan um hugbúnaðarstöðu en miðlari um gagnastöð (resource state)
 - Atom/AtomPub/WADL og hvernig við skilgreinum XML í RESTful þjónustum
 - Praktísk mál eins og breytingar á gögnum (mismunandi útgáfur)
 - Samþættingu lauskúpplaðra kerfa án EAI/ESB (hva ber að varast) (18m-Gartner)
 - REST patterns og anti-patterns (Tvímælalaust áður en við byrjum)

Meira?



- Lestrarefni
 - Roy Fielding - [Architectural Styles and the Design of Network-based Software Architectures](#)
 - Steve Vinoski - [REST Eye for the SOA Guy](#)
 - Steve Vinoski - [Presentation: Steve Vinoski on REST, Reuse and Serendipity](#)
 - Stefan Tilkov - [A Brief Introduction to REST](#)
 - Wikipedia - [Fallacies of Distributed Computing](#)
 - Gregor Roth - [Server load balancing architectures](#)
 - Gregor Roth - [Asynchronous HTTP and Comet architectures](#)