The Road to REST

One link at a time

Rickard Öberg, Neo Technology

QCon code: 7817

How it all began

The Mission

A fork in the road

- RMI?
- SOAP?
- REST?

Exploring

Richardson Maturity model

- Level 0
  - SOAP, XML RPC, POX – Single URI
- Level 1
  - URI Tunnelling – Many URIs, Single verb
- Level 2
  - Many URIs, many verbs
  - CRUD services (e.g. Amazon S3)
- Level 3
  - Level 2 + Hypermedia – RESTful Services

REST to the rescue!

"Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web"

Guidelines

- Client–server
- Stateless
- Cacheable
- Layered system
- Uniform interface

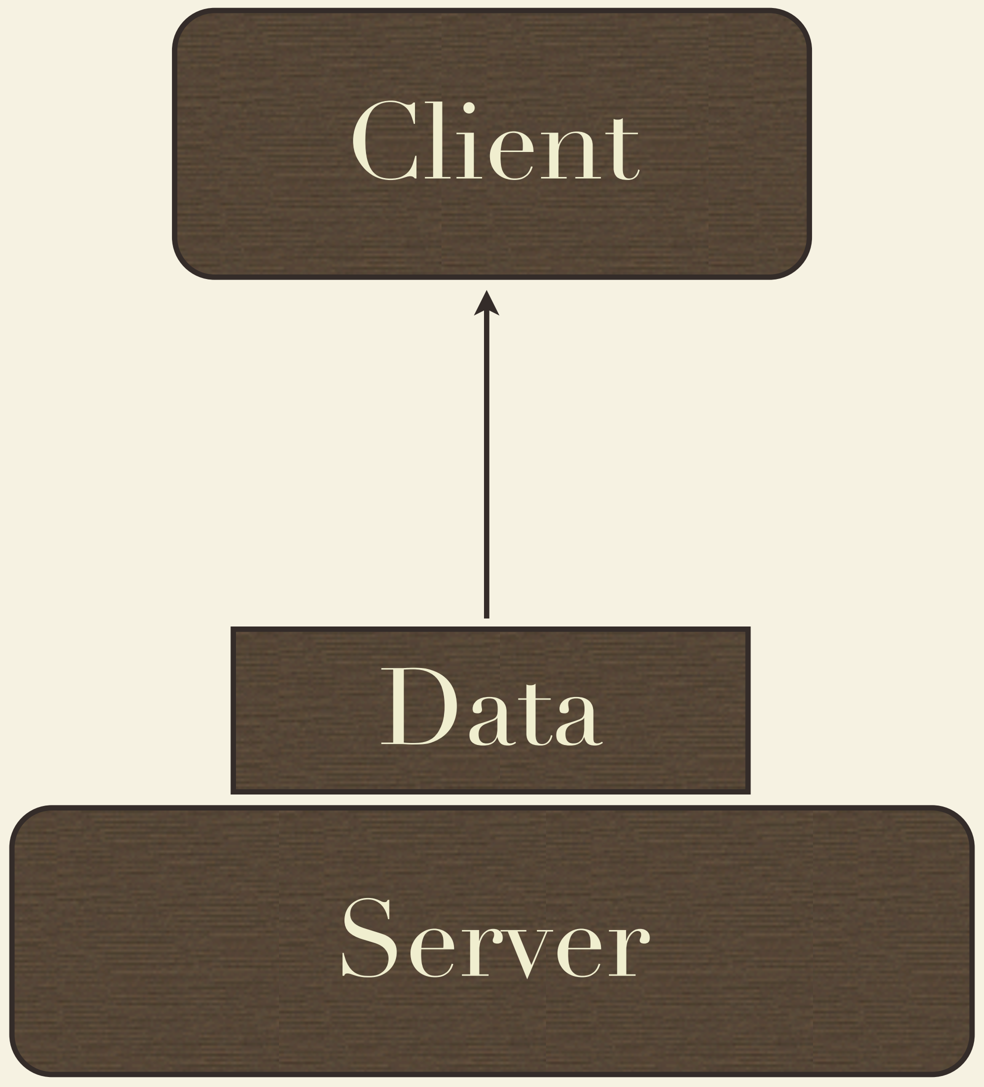
http://en.wikipedia.org/wiki/Representational_State_Transfer#Constraints

More guidelines

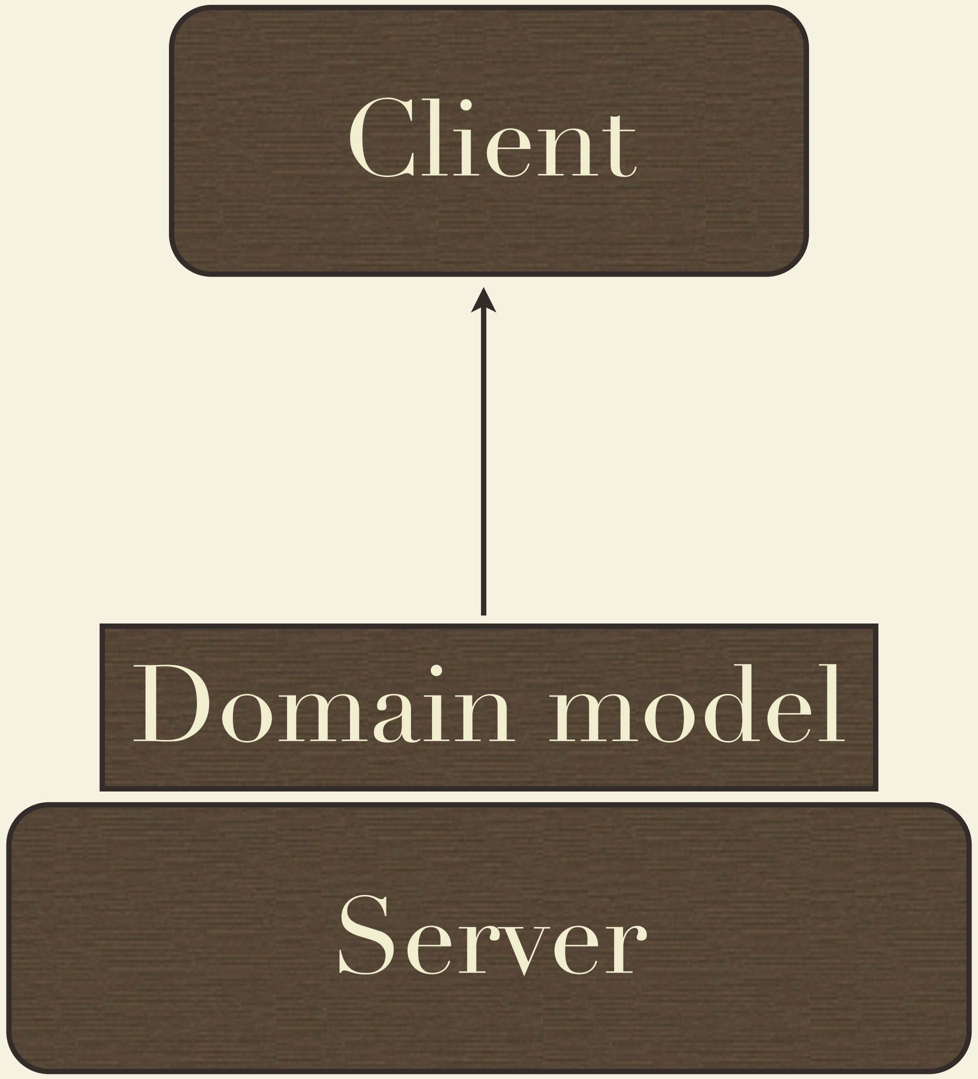
- Identification of resources
- Manipulation of resources through these representations
- Self-descriptive messages
- **Hypermedia As The Engine Of Application State** (HATEOAS)

The narrow road

What shall we expose?



What shall we expose?



How?

- Servlets?
- JAX-RS?
- Restlet
  - www.restlet.org

What the...?

- /users/rickard/changepassword
- /users/rickard/resetpassword

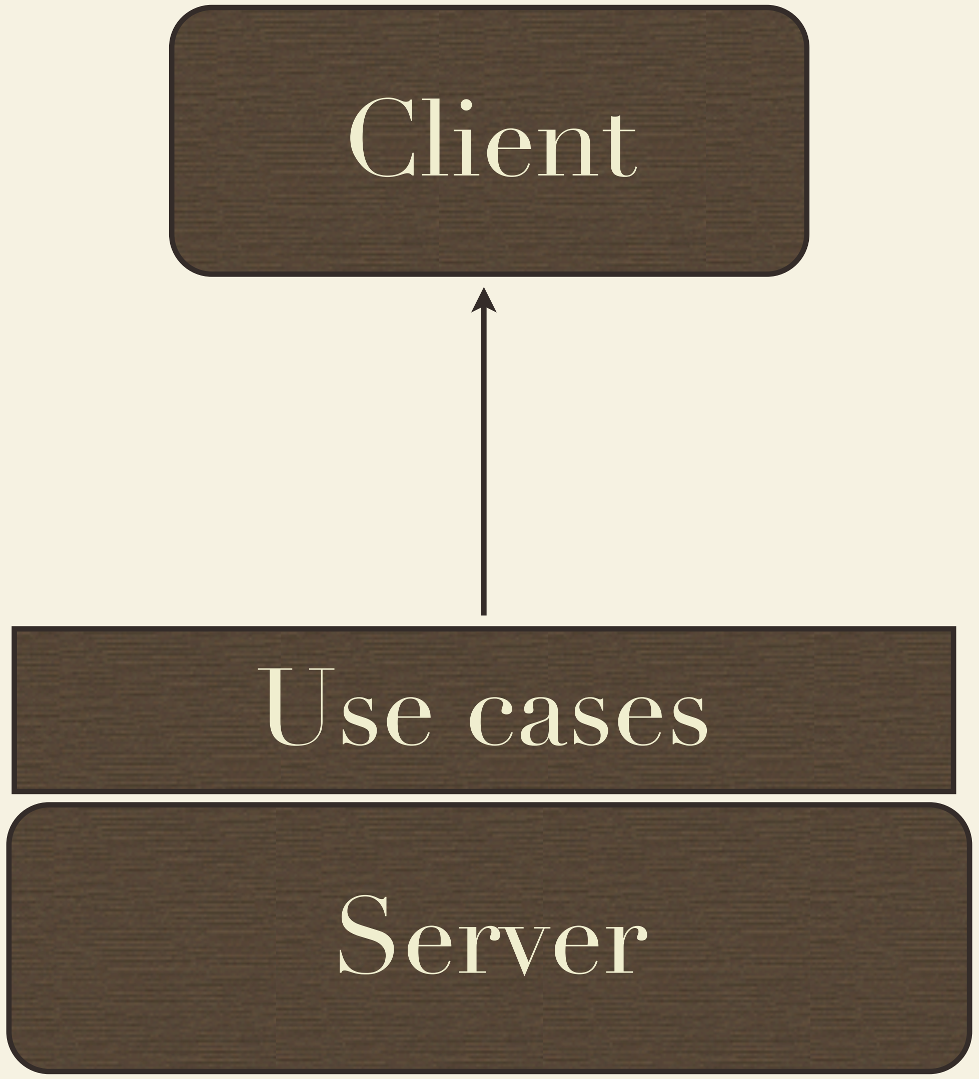
Where is HATEOAS?!?

“The **next control state** of an application resides in the representation of the first requested resource, … The application state is controlled and stored by the user agent … anticipate changes to that state (e.g., link maps and prefetching of representations) … The model application is therefore an engine that moves from one state to the next by **examining and choosing from among the alternative state transitions in the current set of representations**.” - Roy Fielding

What now?!?

One step at a time

What shall we expose?



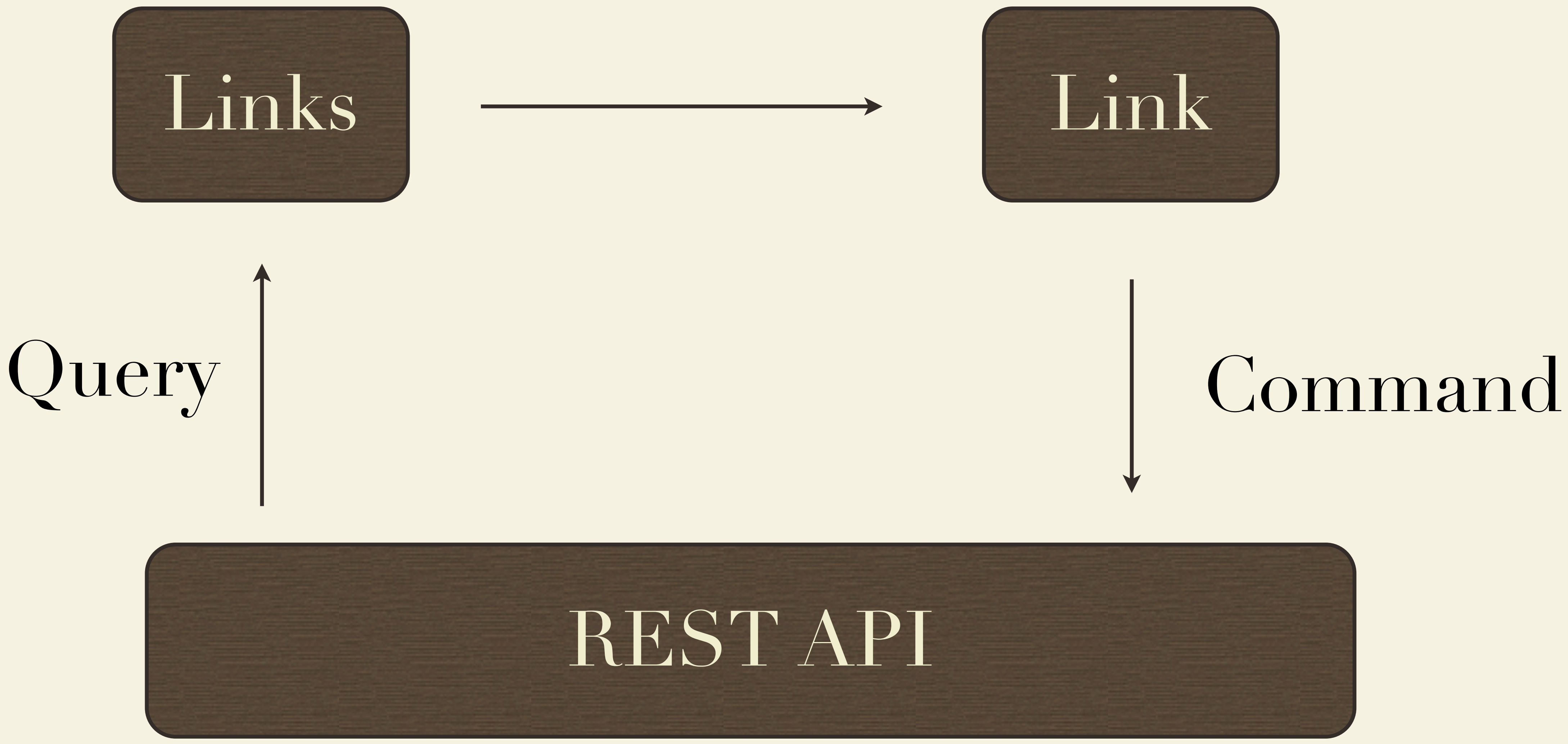
From the user perspective

- /account/changepassword
- /administration/server/users/rickard/resetpassword

Who can do what?

- /workspace/cases/<123>/
- /administrator/...
- /overview/...

Follow the named link



Application flow

- GET ..../index
- GET <link with rel="possiblelabels">
- addlabel?id=1234, rel=addlabel, "Urgent!"
- POST the link

Link checking

- What can the user do? Check for links!

Wohoo!

Versioning

- HATEOAS helps
- Exposing use cases helps
- Send header
  - Ex. X-STREAMFLOW-API: 1.2
- Client libraries

Performance

- Exposing use cases really helps
- Use e.g. GQL for table queries
    - Client decides columns and filtering from fixed set

TL;DL

- A Good REST API is like an ugly website
- RESTful clients should follow links and submit forms

Q & A

rickard.oberg@neotechnology.com
@rickardoberg
http://rickardoberg.wordpress.com
http://qi4j.org