What RESTful API is not

Tomasz Skowroński
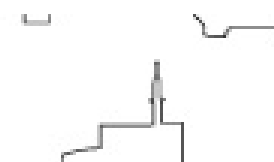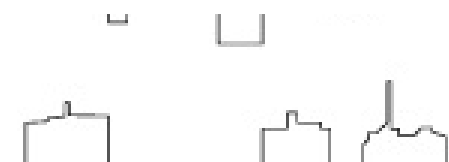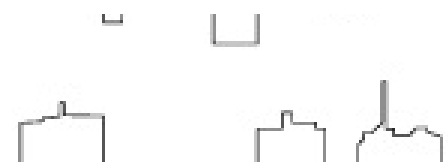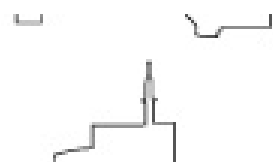
Agenda

Rest?

Rest?

1. standard

2. golden hammer

3. framework

4. protocol

5. web 2.0 SOAP

6. architectural pattern

7. web 2.0 SOA

8. pl.wikipedia.org/wiki/rest

Rest!

Rest!

01. /**

02. * Representational State Transfer

03. *

04. * @author Roy Thomas Fielding

05. * @date 2000

06. */

07. Rest.prototype = new ArchitecturalStyle();

Restful ABC

nouns + verbs + adverbs

HTTP methods

GET /balls

DELETE /balls/3

The panels of the soccer ball are matched together.

Stitching

Air

Pumping up ball.
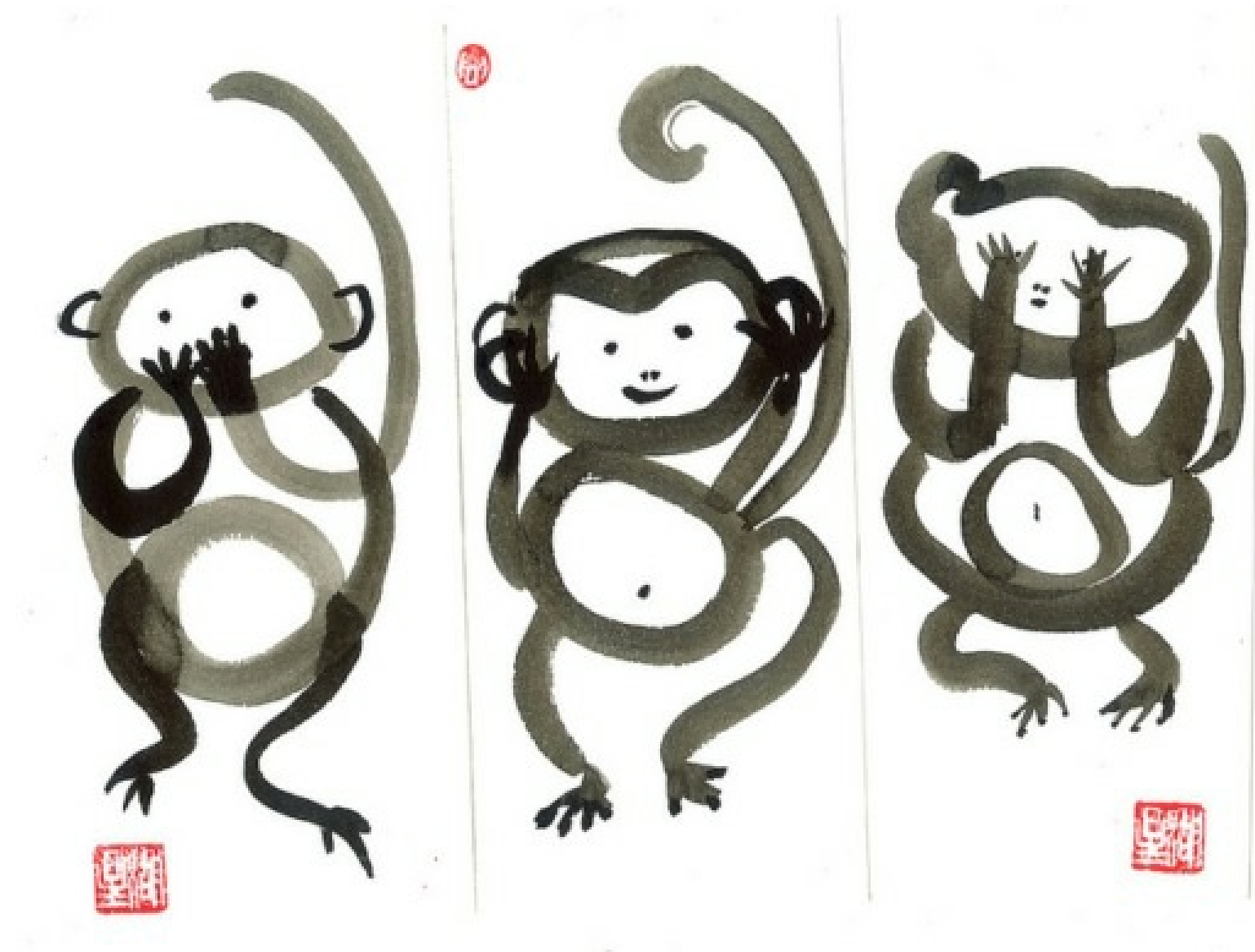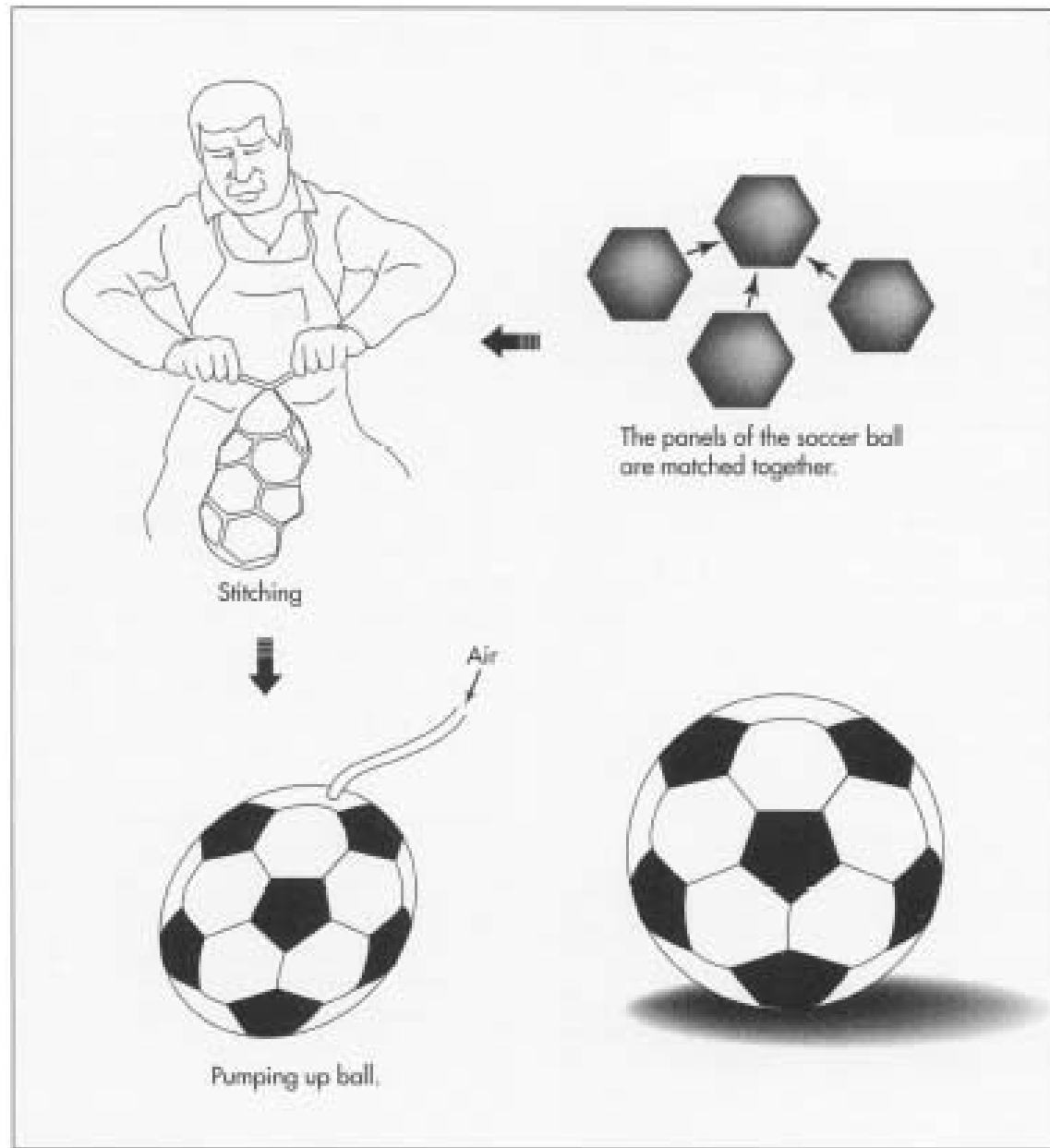
POST /balls

PUT /balls/3

PATCH /balls/3

Rest != CRUD

POST /balls/3/kicks

GET /balls/3/kicks/1

Rest != CRUD$_2$

Read methods

safe, idempotent

- GET

- HEAD

- OPTIONS

Replace methods

not safe, idempotent

- PUT
- DELETE

Edit methods

not safe, not idempotent

- POST

- PATCH

Five-Spice Rest recipe

1. Client-server

2. Stateless server

3. Cache

4. Layered system

5. Uniform interface

6. Code-on-demand (optional)

1-3 Client-Cache-Stateless-Server
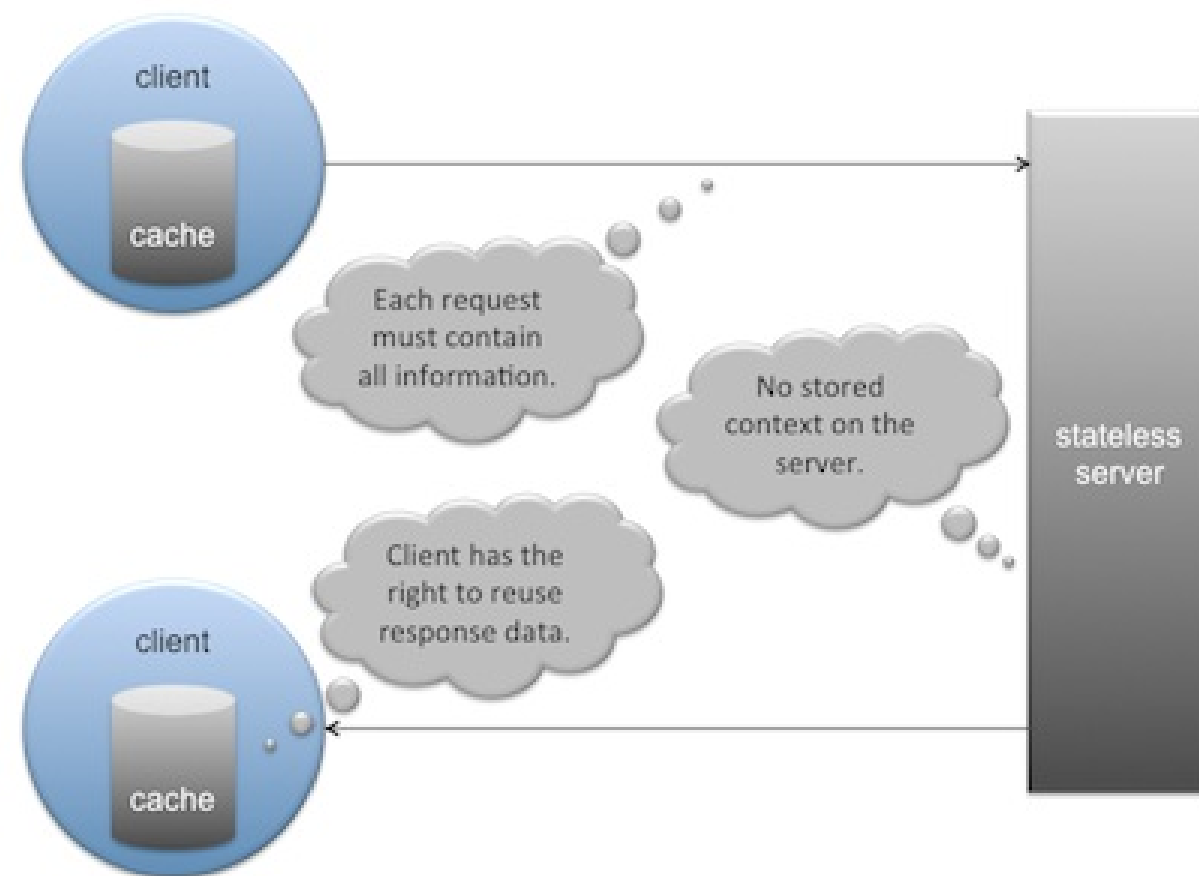
Rest != P2P, RS,RDA,EBI

4 Layered System
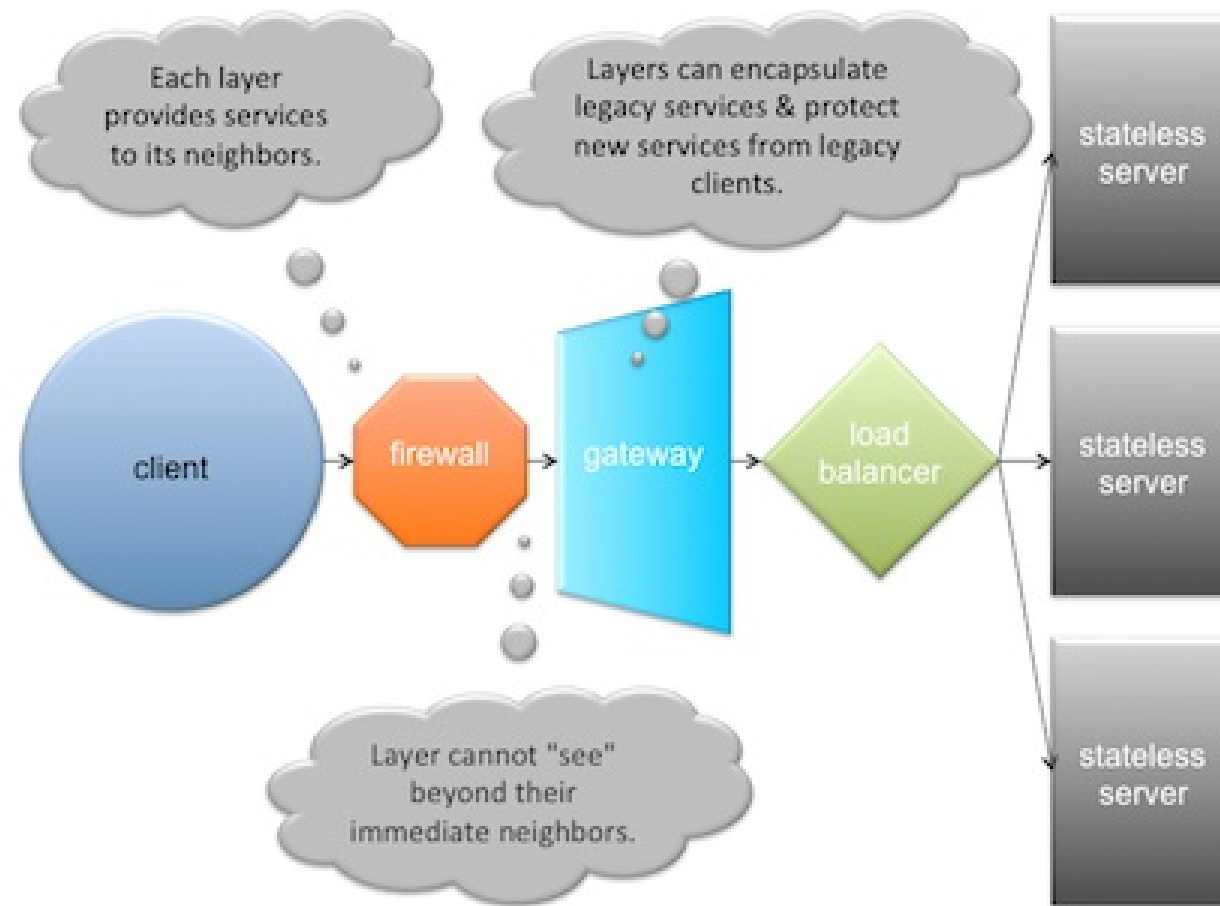
1-4 CS-SS-CC-LS advantages

1. Modifiability

2. Scalability

3. Visibility (SS, LS)

4. Reliability (SS)

5. Simplicity (LS)

6. Performance (CC)

Rest != SOA

5 Uniform interface

Rest != nice urls

5.1 identify via URIs

Rest != Http + Json

5.2 manipulate via representations

5.3 react via methods and statuses

Rest !=
ok ? 200 : 500

HTTP statuses

1xx Informational

100 Continue   101 Switching Protocols   102 Processing (WebDAV)

2xx Success

★ 200 OK   ★ 201 Created   202 Accepted
203 Non-Authoritative Information ★ 204 No Content   205 Reset Content
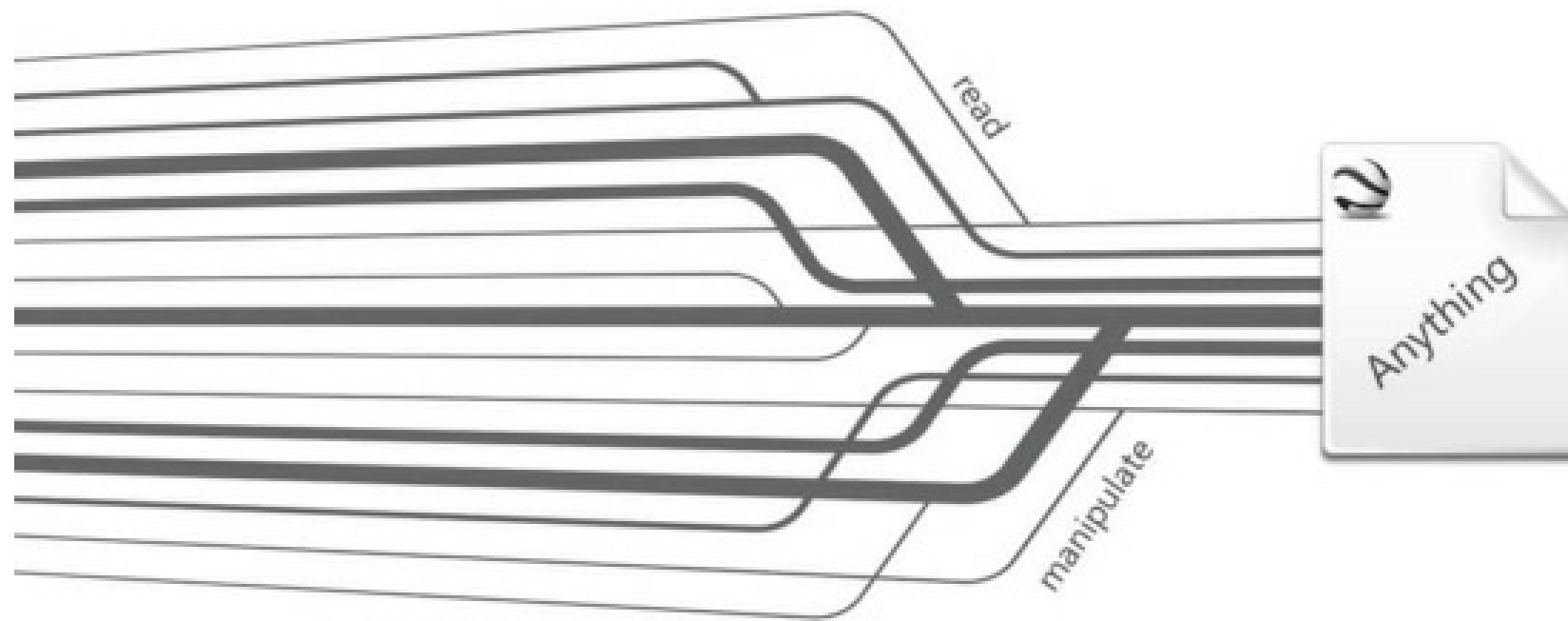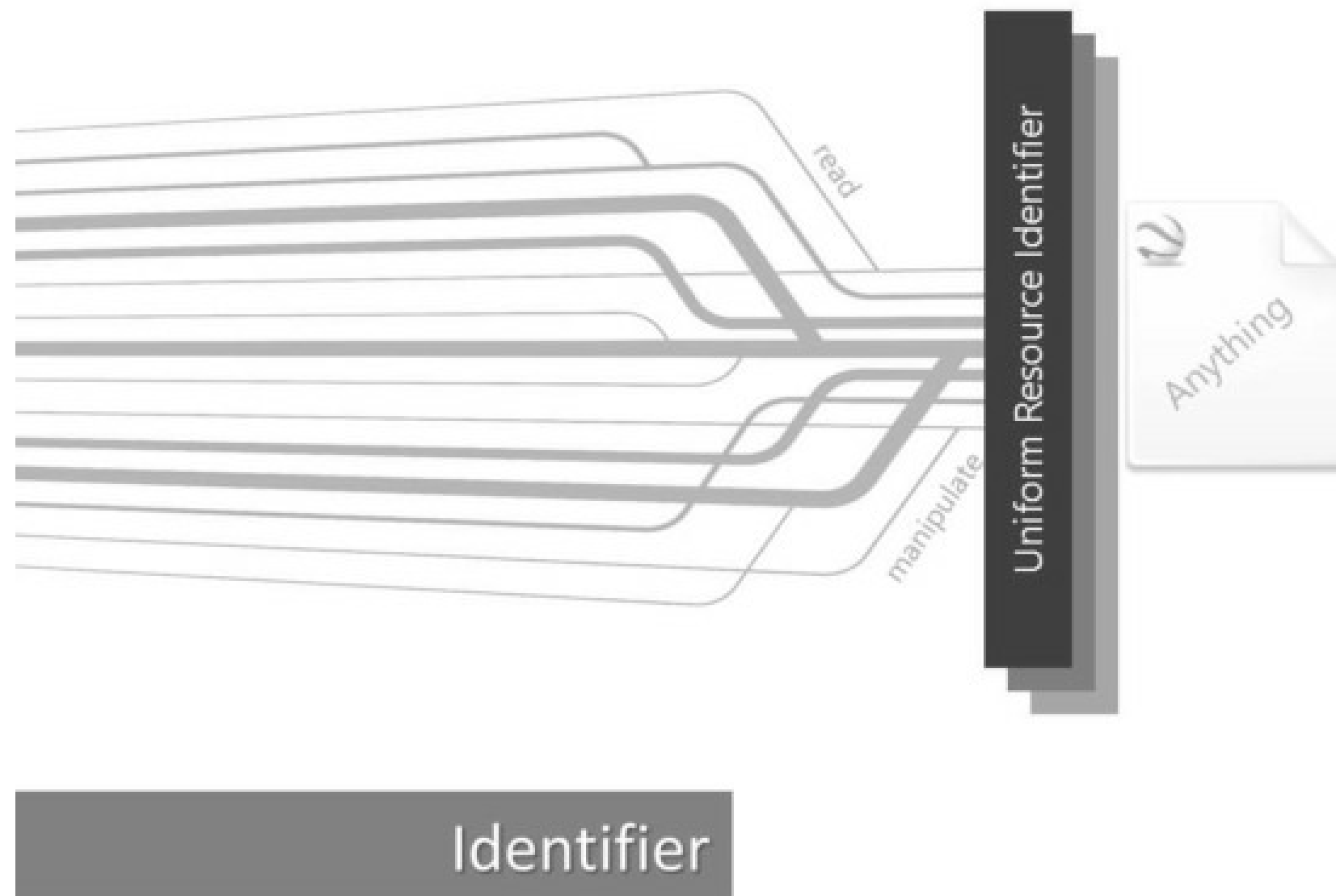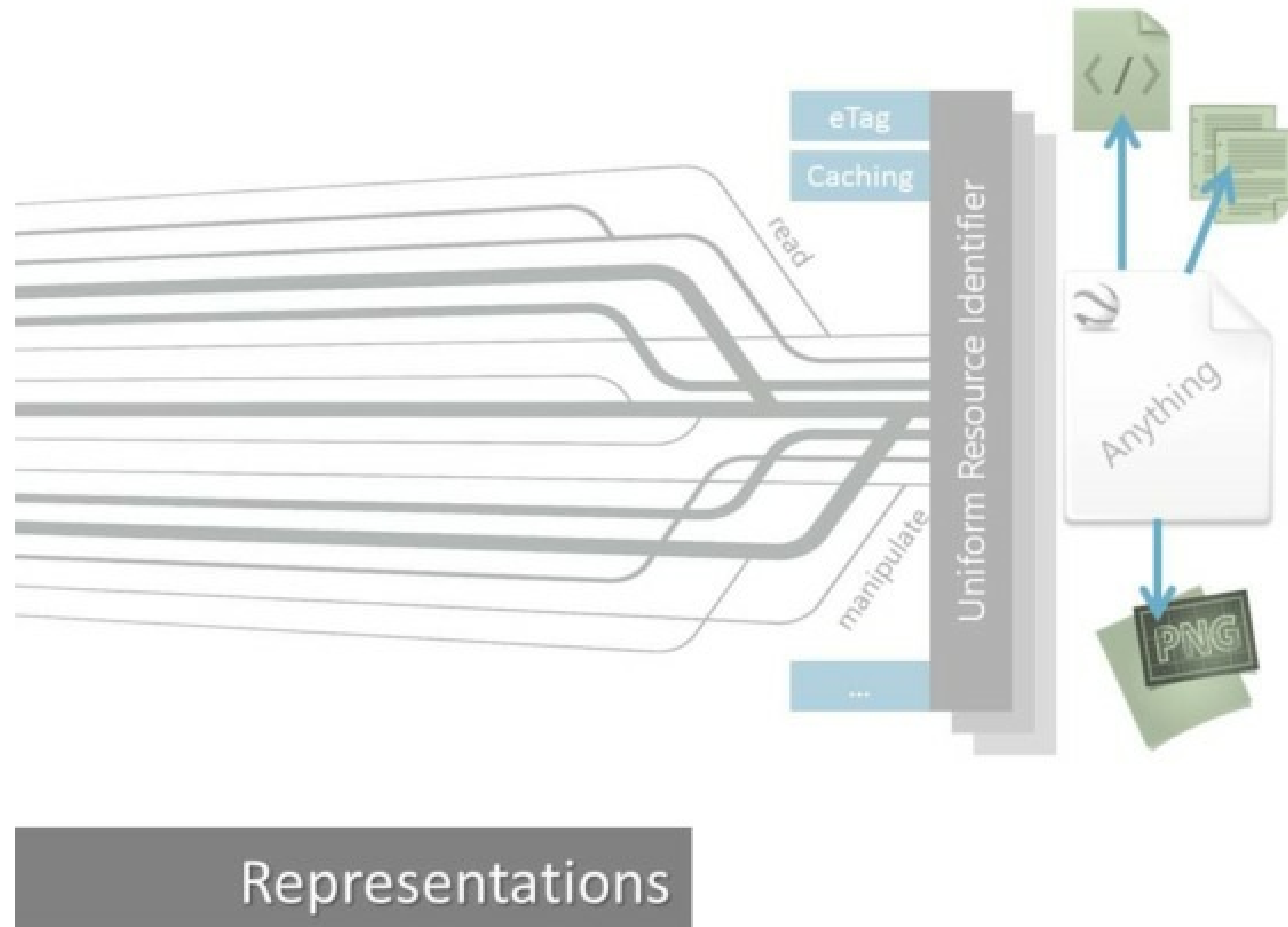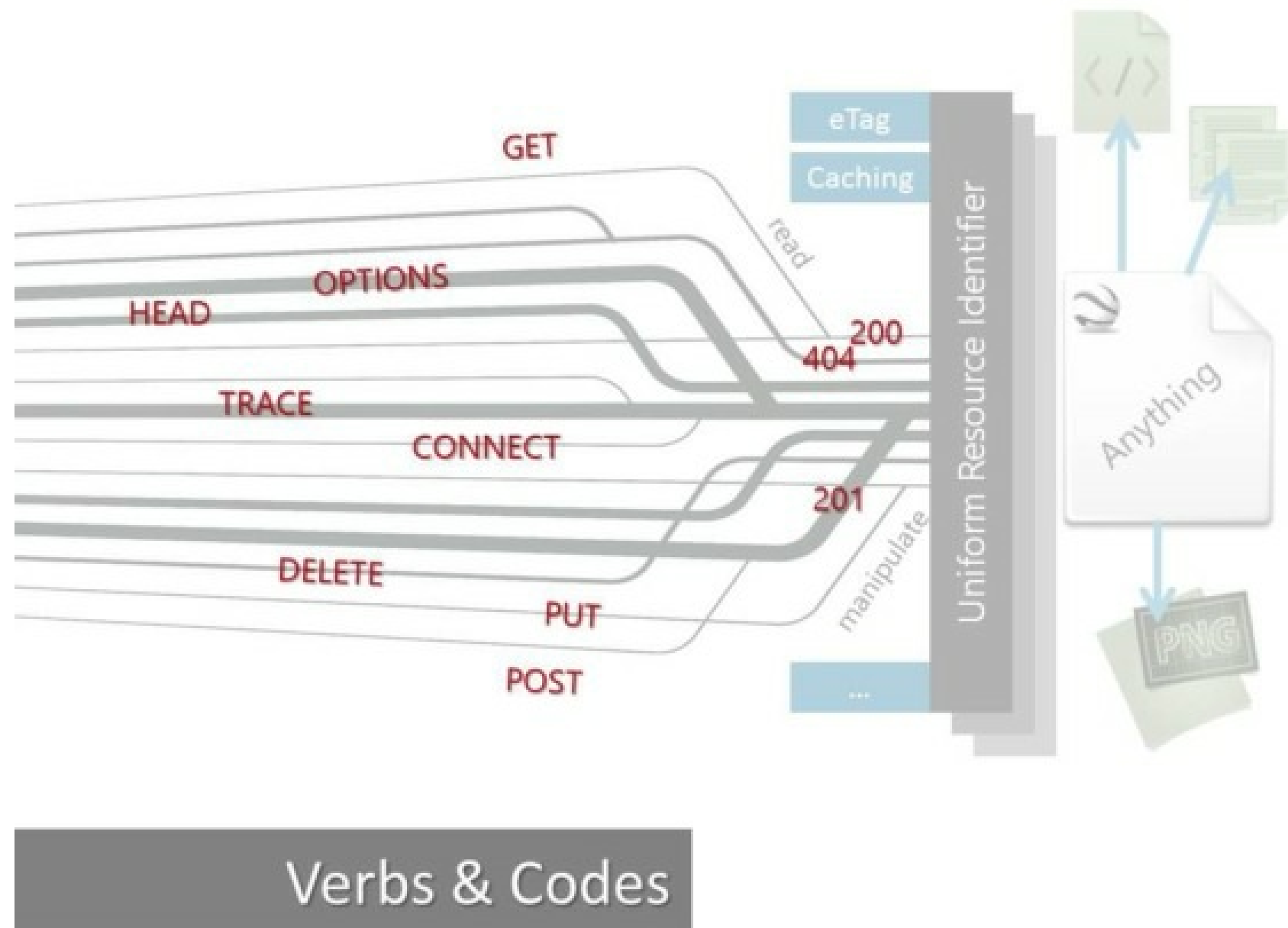206 Partial Content   207 Multi-Status (WebDAV)   208 Already Reported (WebDAV)
226 IM Used

3xx Redirection

300 Multiple Choices   301 Moved Permanently   302 Found
303 See Other   ★ 304 Not Modified   305 Use Proxy
306 (Unused)   307 Temporary Redirect   308 Permanent Redirect (experimental)

4xx Client Error

★ 400 Bad Request   ★ 401 Unauthorized   402 Payment Required
★ 403 Forbidden   ★ 404 Not Found   405 Method Not Allowed
406 Not Acceptable   407 Proxy Authentication Required   408 Request Timeout
★ 409 Conflict   410 Gone   411 Length Required
412 Precondition Failed   413 Request Entity Too Large   414 Request-URI Too Long
415 Unsupported Media Type   416 Requested Range Not Satisfiable   417 Expectation Failed
418 I'm a teapot (RFC 2324)   420 Enhance Your Calm (Twitter)   422 Unprocessable Entity (WebDAV)
423 Locked (WebDAV)   424 Failed Dependency (WebDAV)   425 Reserved for WebDAV
426 Upgrade Required   428 Precondition Required   429 Too Many Requests
431 Request Header Fields Too Large 444 No Response (Nginx)   449 Retry With (Microsoft)
450 Blocked by Windows Parental Controls (Microsoft) 499 Client Closed Request (Nginx)

5xx Server Error

★ 500 Internal Server Error   501 Not Implemented   502 Bad Gateway
503 Service Unavailable   504 Gateway Timeout   505 HTTP Version Not Supported
506 Variant Also Negotiates (Experimental) 507 Insufficient Storage (WebDAV) 508 Loop Detected (WebDAV)
509 Bandwidth Limit Exceeded (Apache) 510 Not Extended   511 Network Authentication Required
598 Network read timeout error   599 Network connect timeout error

★ "Top 10" HTTP Status Code.

5.4 states and relationships via hypermedia
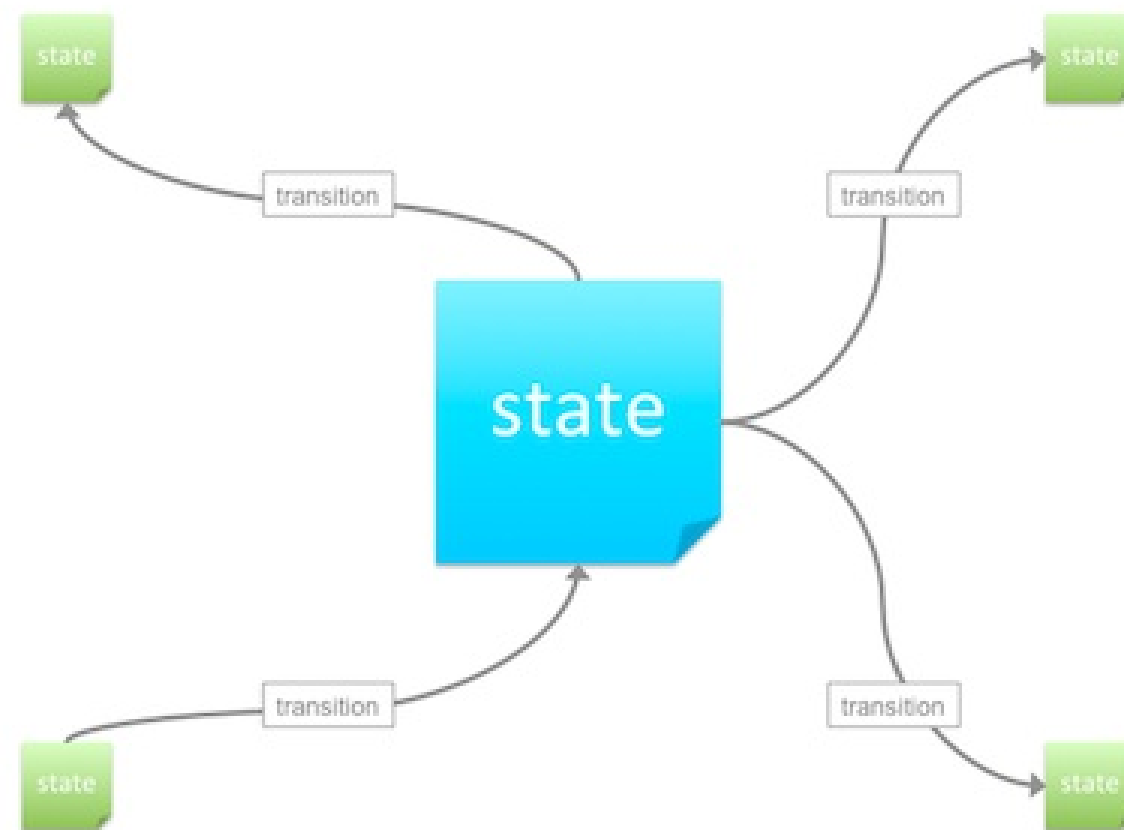
5 Uniform interface advantages

USB-A   Micro-B   Mini-B   USB-B

1. Visibility

2. Modifiability

3. Simplicity

What RESTful API is not

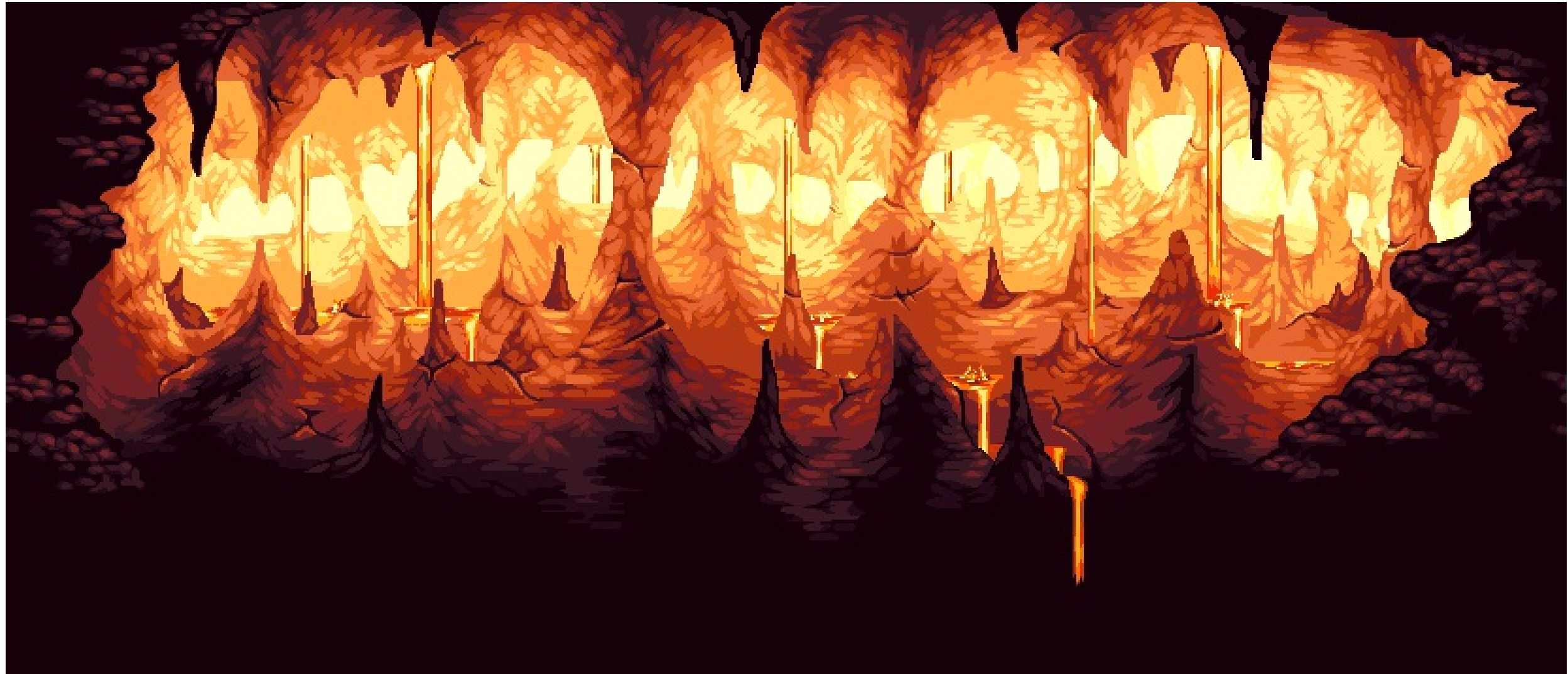Richardson Maturity Model

Level 0 - No Rest

Level 0 - No Rest

01. POST /findBikesByStation?id=1234

02. `200` [1000,5234,2345]

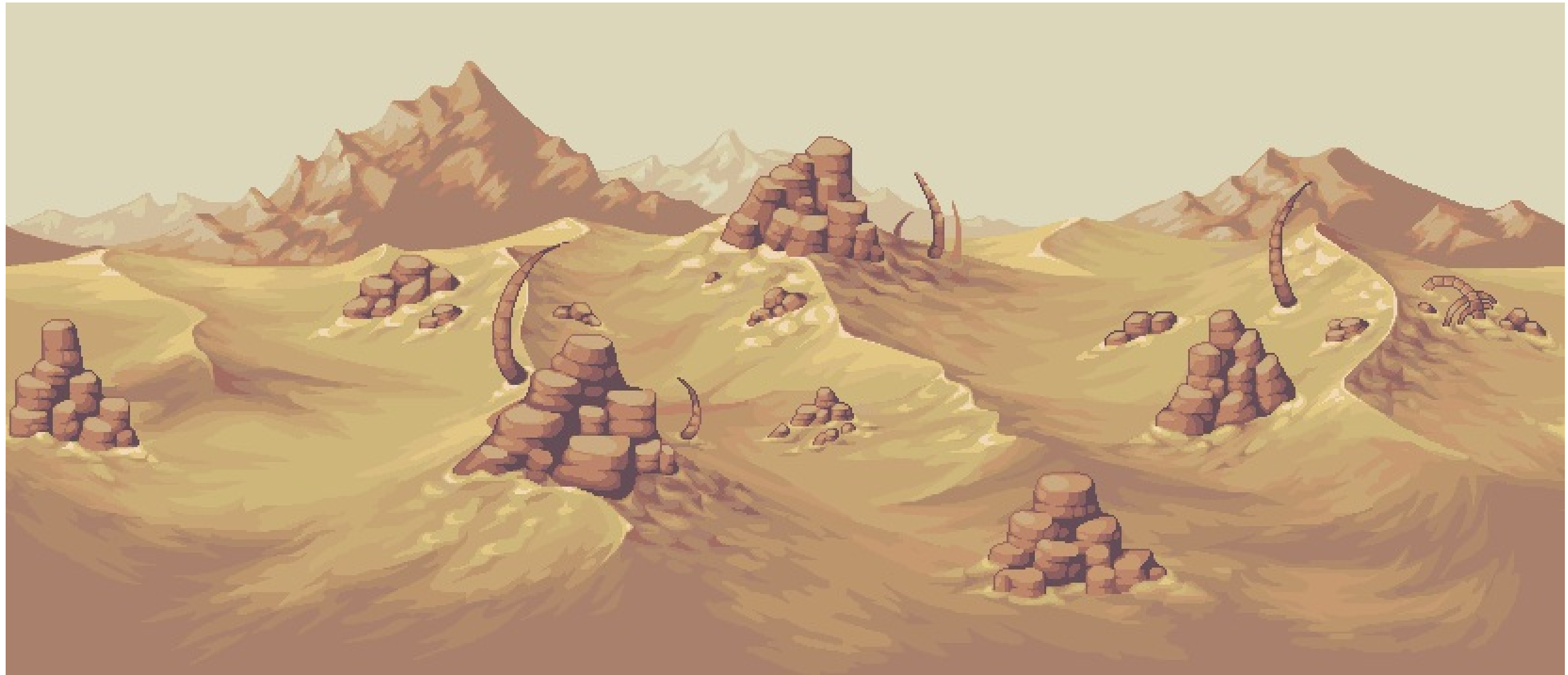01. POST /rentBike/1000

02. `200`

Rest != RPC

Level 1 - Resources

Level 1 - Resources

01. POST /stations/1234/bikes

02. `200` [1000,5234,2345]

01. POST /bikes/1000/rentBike

02. `200`

Rest != Http transport

Level 2 - HTTP Verbs
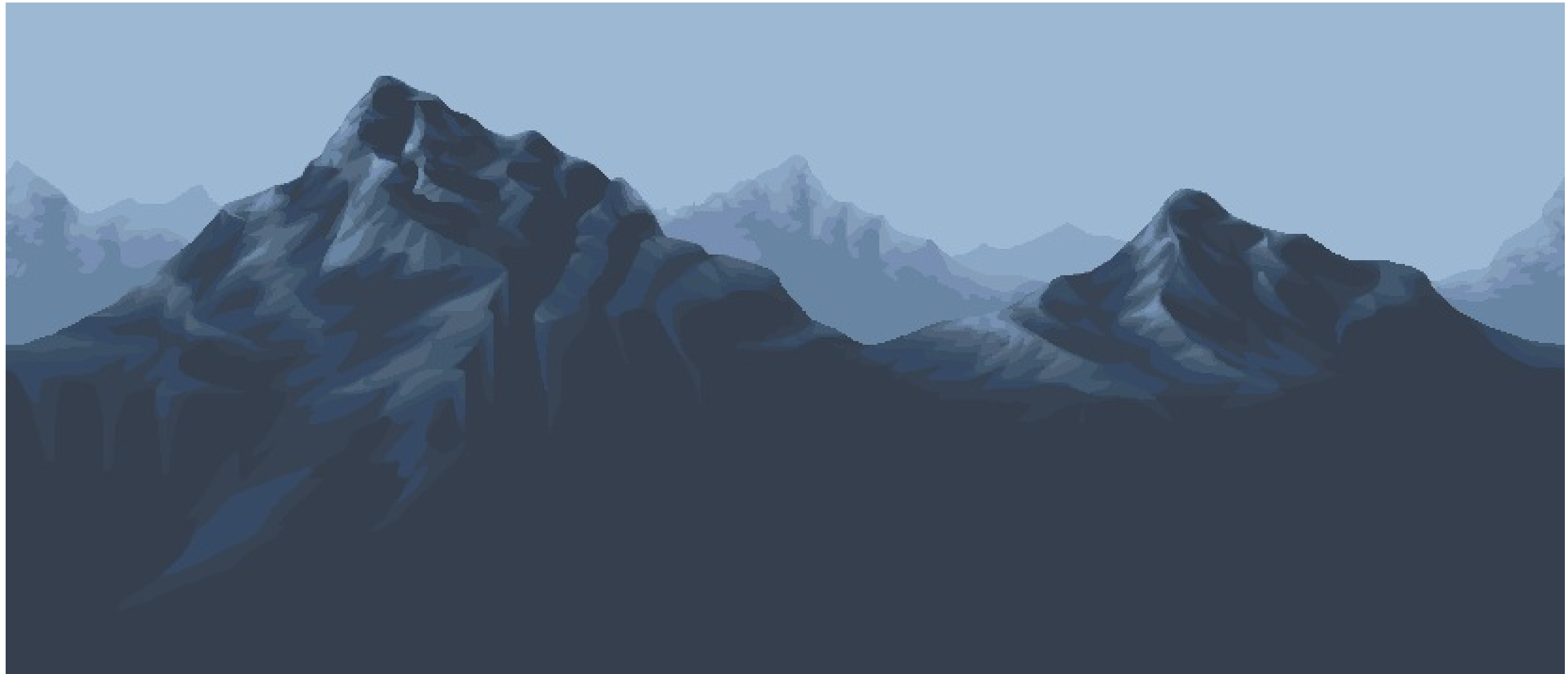
Level 2 - HTTP Verbs

01. GET /stations/1234/bikes

02. `200` [1000,5234,2345]

01. POST /bikes/9999/rents

02. `200` {"error":"already rented"}

Level 2.5 - HTTP Verbs, Statuses

01. GET /stations/1234/bikes

02. `200` [1000,5234,2345]

01. POST /bikes/9999/rents

02. `409`

01. POST /bikes/1000/rents

02. `301` Location: /bikes/1000/rents/7788

Representational State Transfer

Rest != only resources

Level 3 - Hypermedia

Level 3 - Hypermedia (HATEOAS)

01. POST /bikes/1000/rents

02. `301` Location: /bikes/1000/rents/7788

03. {"content": {"start": "2014-06-23T18:45"},

04. "links":[

05. { "rel": "self", "method": "get", "href": "/bikes/1000/rents/7788"},

06. { "rel": "return", "method": "delete", "href": "/bikes/1000/rents/7788"},

07. { "rel": "bike", "method": "get", "href": "/bikes/1000"}

08.]}

Level 3 - Hypermedia Controls

POST /bikes/1000/rents

rel	method	href
self	get	/bikes/1000/rents/7788
return	delete	/bikes/1000/rents/7788
bike	get	/bikes/1000

Summary

Summary

1. Client-Cache-Stateless-Server-Layered architectural style

2. HTTP features as uniform interface

   - URIs, representations, methods, statuses, hypermedia

3. levels: from resources to hypermedia

References

1. **http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm**

2. http://design.inf.usi.ch/sites/default/files/talks/SOA2009-REST-Patterns.pdf

3. http://whatisrest.com/rest_constraints/index

4. https://blog.apigee.com/detail/api_design_ruminating_over_rest

5. http://restlet.com/blog/tag/rest-api/

6. http://www.planetgeek.ch/2012/11/27/rest-fundamentals/

7. http://www.restapitutorial.com/httpstatuscodes.html

8. http://4.bp.blogspot.com/_yPMl1jV3I5Q/TKnyMjGudvI/AAAAAAAAAJE/kRdTMZDueEE/s1600/the-letter-r.jpg

9. https://c1.staticflickr.com/5/4075/4810872878_71e129163f_z.jpg

10. https://31.media.tumblr.com/tumblr_md0d70S9sy1rsvtbb.jpg

11. http://s3.amazonaws.com/churchplantmedia-cms/3strand_church_network/rest-stop.jpg

12. https://dlnmh9ip6v2uc.cloudfront.net/tutorialimages/USB_Guide/USBTypes2.jpg

13. http://pukahuna.deviantart.com

Thanks