REST API Design, Development & Management

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com

This deck is part of a online course on Summary of a course that covers the A to Z of RESTful API. More information.

REST API Design, Development & Management

http://www.acloudfan.com

raj@acloudfan.com

Summary of a course that covers the A to Z of RESTful API. More information.

Architectural Constraints

Updated: Dec 27th, 2016

raj@acloudfan.com
http://www.acloudfan.com

Is your Architecture RESTful?

REST Architecture Constraints

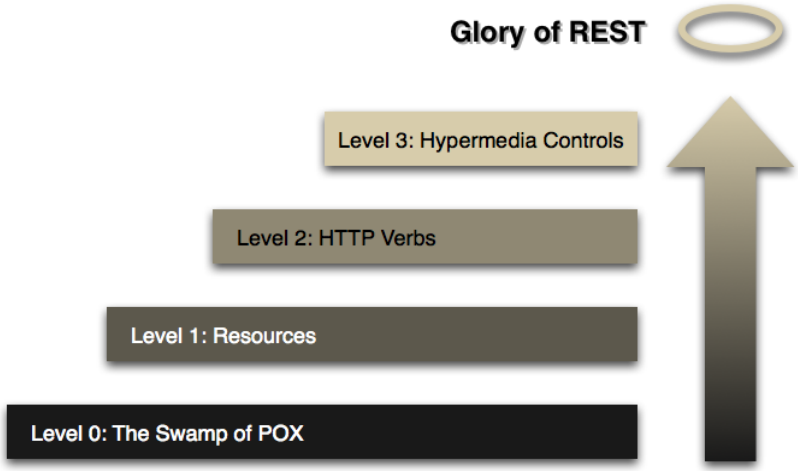
raj@acloudfan.com
http://www.acloudfan.com

1. Client - Server
2. Uniform Interface
3. Statelessness
4. Caching
5. Layered system
6. Code on demand (*optional*)

RESTful Architecture

raj@acloudfan.com
http://www.acloudfan.com

How do I know if my current Architecture is RESTful?



Richardson Maturity model
Assigns a score between 0 & 3

REST Architectural Constraints

#1 Uniform Interface

raj@acloudfan.com
http://www.acloudfan.com

Client and Server share a common technical interface

Interface =

- Contract for communication between client-server
- Contract has NO business context
- Contract defined using HTTP methods & media types

Uniform Interface De-Couples the architecture; components can evolve independently

REST Architectural Constraints

#1 Uniform Interface

raj@acloudfan.com
http://www.acloudfan.com

There are 4 guiding principles

1. Identity of the resource (URI/URL)

2. Representation of the resource

3. Self descriptive messages - metadata

4. Hypermedia

REST Architectural Constraints

#2 Client Server

raj@acloudfan.com
http://www.acloudfan.com

Client and server do not reside in same process

RPC over HTTP

SERVER Decoupled from CLIENT

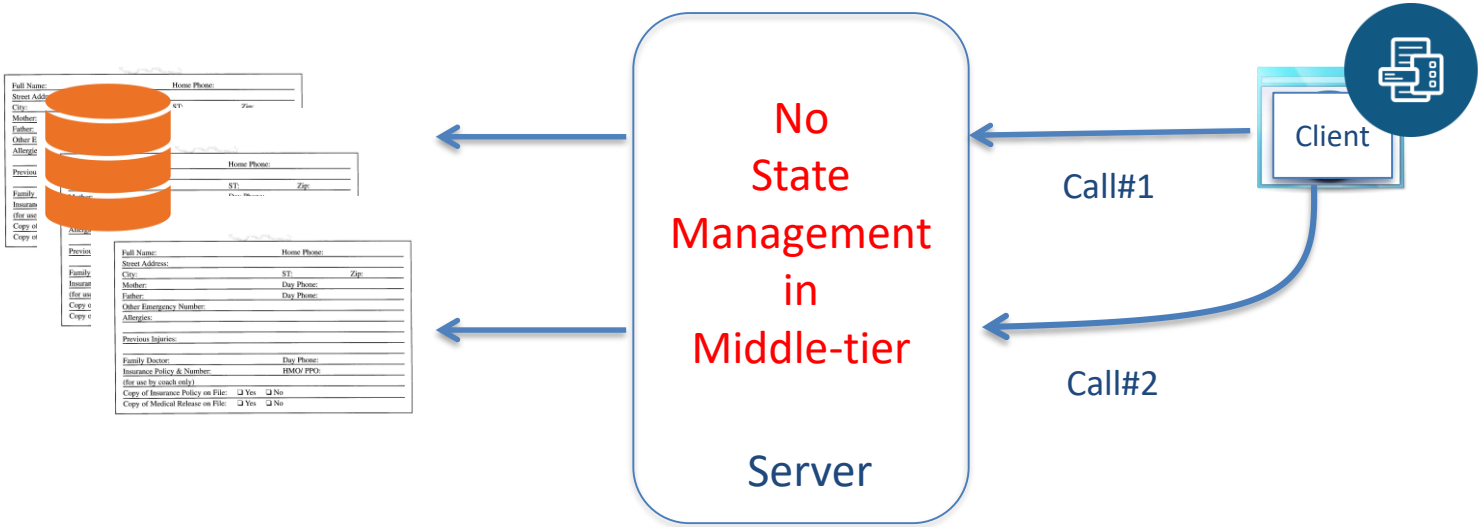
Client and Server can change/evolve **without** impacting each other

REST Architectural Constraints

#3 Statelessness

raj@acloudfan.com
http://www.acloudfan.com

- Each client request is independent
- Server receives all info it needs in the request



REST Architectural Constraints

#4 Caching

raj@acloudfan.com
http://www.acloudfan.com

Statelessness

- Negative impact on performance
- Chattiness
- Higher data traffic (impact's scalability)

Caching

- Performance
- Scalability
- Reduce chattiness

Caching counter-balances some of the negative impacts of Statelessness

REST Architectural Constraints

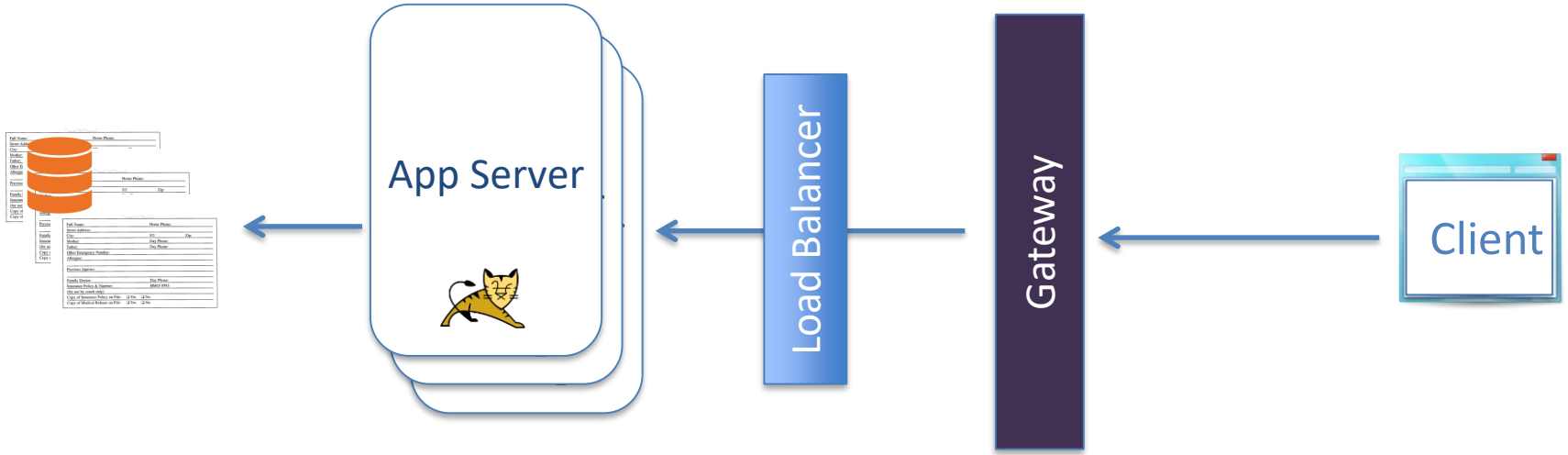
#5 Layered System

raj@acloudfan.com
http://www.acloudfan.com

- Client-Server architecture consist of multiple layers

  No one layer can see past the next

  Layers may be added, removed or modified based on needs



REST Architectural Constraints

#6 Code on demand (Optional)

raj@acloudfan.com
http://www.acloudfan.com

Server can extend client's functionality by sending the code