REST Design with Go programming language

Harchaoui Abdel Ali / Master Degree In Educational Technology & Mathematics

@HARCHAOUI_Abdel

HARCHAOUI Abdel Ali

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
\section{المصفوفات  Arrays}
تشكل المصفوفات البنية الأساسية التي تبني عليها البنات الأخرى كالشرائح $\mathbf{Slices}$ والخرائط $\mathbf{Maps}$. \\
عند فهم الجيد لكيفية عمل المصفوفات سوف تتمتع بإكتشاف قوة الأنواع الأخرى والأناقة التي توفرها.\\
خاصيات المصفوفات:
\begin{itemize}
    \item
سلسلة من الحدود (عناصر) المتقاربة في الذاكرة.
    \item
يمكن لحدود المصفوفة أن تكون إما من الأنواع الأولية كالأعداد والنصوص، أو الأنواع المعدلة $\mathbf{(struct)}$
    \item
يمكن الوصول إلى حدود المصفوفة برتبتها.
    \item
عدد حدود المصفوفة يكون محددا مسبقا عند إنشاءها.
    \item
نوع الحدود المخزنة في المصفوفة يمنحنا فكرة عن المسافة التي نحتاجها للوصول والتنقل بين هذه العناصر.
    \item
التنقل بين حدود مصفوفة سريعا وثابت.
\end{itemize}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

المصفوفات :

تشكل المصفوفات البني الأساسية التي تبني عليها البنات الأخرى كالشرائح **Slices** و الخرائط **Maps**

عند فهم الجيد لكيفية عمل المصفوفات سوف تستمتع عند إكتشاف قوة الأنواع الأخرى والأناقة التي توفرها

خاصيات المصفوفات :

سلسلة من الحدود (عناصر) المتقاربة في الذاكرة .

يمكن لحدود المصفوفة أن تكون إما من الأنواع الأولية كالأعداد والنصوص، أو الأنواع المعدلة (**struct**)

يمكن الوصول إلى حدود المصفوفة برتبتها .

عدد حدود المصفوفة يكون محددا مسبقا عند إنشاءها .

نوع الحدود المخزنة في المصفوفة يمنحنا فكرة عن المسافة التي نحتاجها للوصول والتنقل بين هذه العناصر .

التنقل بين حدود مصفوفة سريعا وثابت .

المصفوفات من الرتبة الأولى

إنشاء مصفوفة وتهيئها

لتكن **M** مصفوفة من الرتبة الأولى ذات نوع **T** وعدد الحدود **n** (يسمى طول المصفوفة **M**).

نقوم بإنشاء المصفوفة **M** كمالي :

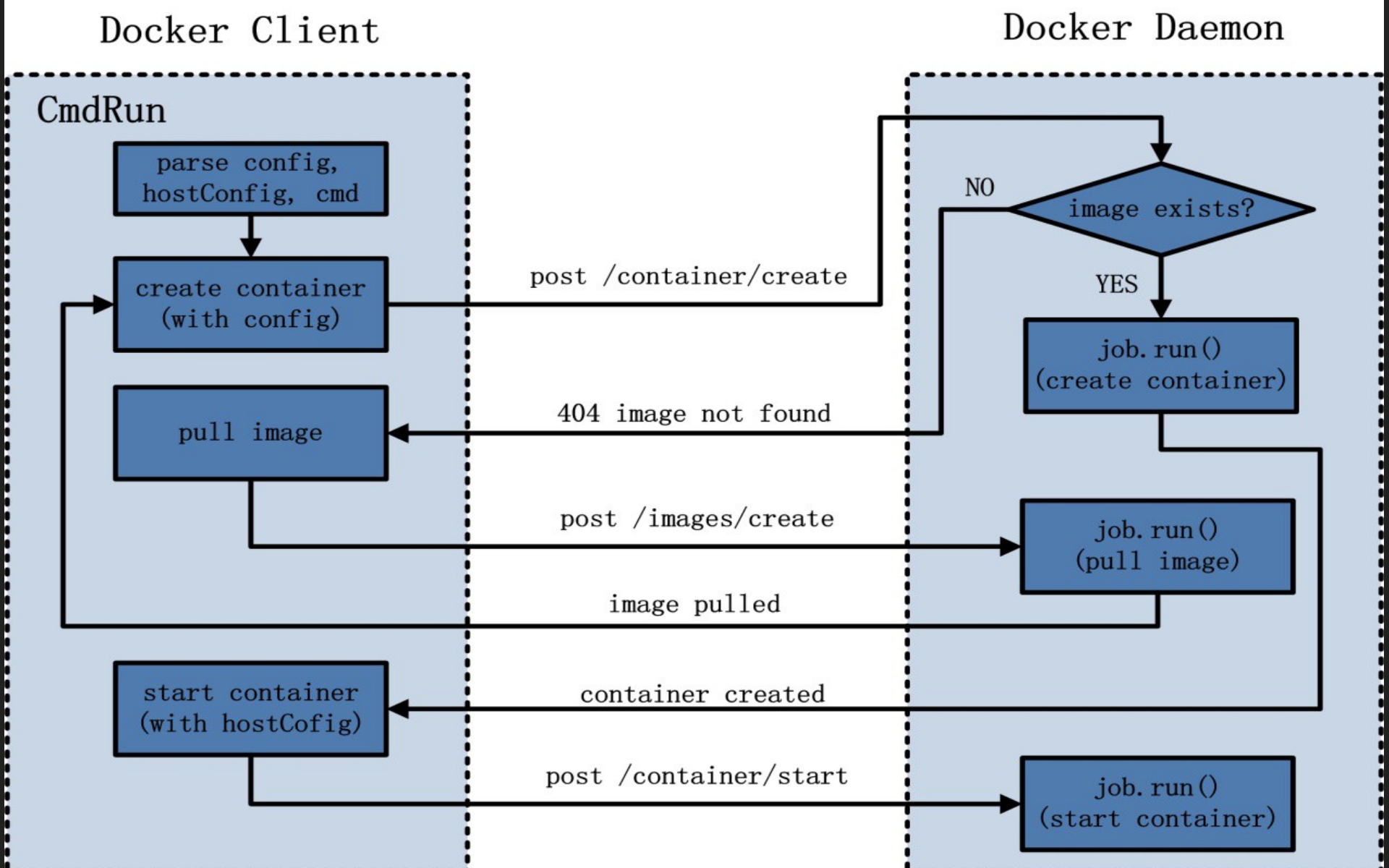
على
ث عن
الرتبة
Unknow
02/11/20

Modern Web App In Go
برمجة تطبيقات الويب الحديثة بلغة Go

vim ⌘1 bash ⌘2

```go
/*
* this demo shows how to implement a full RESETful web-app
* with data Persistence "using postgresSQL as our data-store"
**/
package main

import (
        "database/sql"
        "fmt"
        "log"
        "net/http"

        _ "github.com/lib/pq"
)

type Customer struct {
        uid   string
        name  string
        email string
}

var (
        dbConnection *sql.DB
)

func init() {

        var err error

        //sql.DB object it returns is not a database connection - it's an abstraction representing a pool of underlying connections
        // we can set db.SetMaxOpenConns and db.SetIdleOpenConns
        dbConnection, err = sql.Open("postgres", "user=gopher password=1111aaaa dbname=reset_demo sslmode=disable")
        if err != nil {
                log.Fatal(err)
        }
```


Why?


Docker Client
Docker Daemon
CmdRun
parse config, hostConfig, cmd
create container (with config)
post /container/create
NO
image exists?
YES
job.run() (create container)
pull image
404 image not found
post /images/create
job.run() (pull image)
image pulled
start container (with hostCofig)
container created
post /container/start
job.run() (start container)

Rest API | Location | Find jobs

Like this search?
Get notified when new jobs match what you're looking for

Email address

Create a job alert

Location

- San Francisco, California (331)
- New York, New York (219)
- Chicago, Illinois (204)
- Seattle, Washington (129)
- Austin, Texas (110)

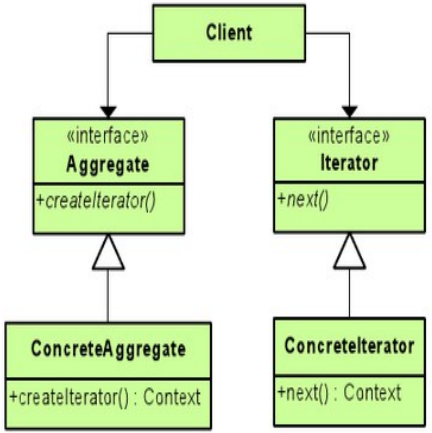
See more

4,106 **Rest API** jobs in **United States**

REST API Developer
Bloomberg LP
New York, New York
In order to do this, we need engineers with experience in designing and building **REST** APIs for web delivery of human and machine readable...

Senior REST API Developer
Andiamo Partners
New York, New York
In order to do this, we need engineers with experience in designing and building **REST** APIs for web delivery of human and machine readable...
Apply

.NET Developer - C#, ASP.net/MVC, Web API (REST)
CyberCoders



Iterator

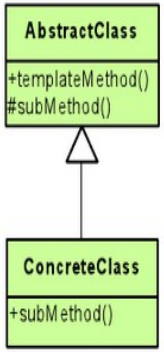
Type: Behavioral

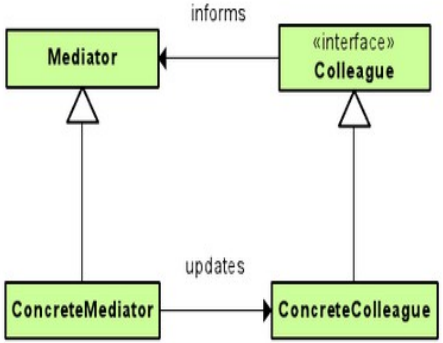
What it is:
Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

Template Method

Type: Behavioral

What it is:
Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure.





Mediator

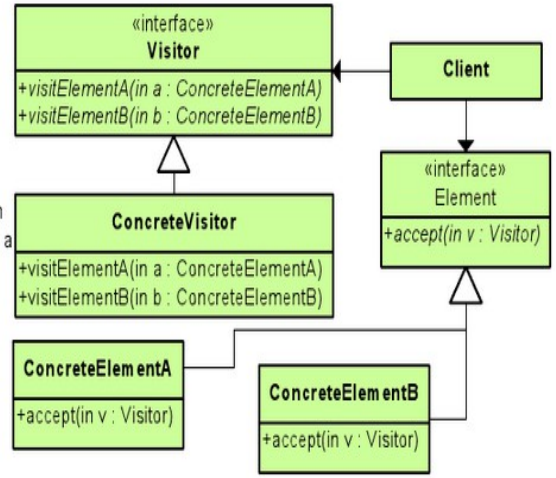
Type: Behavioral

What it is:
Define an object that encapsulates how a set of objects interact. Promotes loose coupling by keeping objects from referring to each other explicitly and it lets you vary their interactions independently.

Visitor

Type: Behavioral

What it is:
Represent an operation to be performed on the elements of an object structure. Lets you define a new operation without changing the classes of the elements on which it operates.



Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Massachusetts: Addison Wesley Longman, Inc..

All nodes and edges in the Graph API can be read simply with an HTTP `GET` request to the relevant endpoint. For example, if you wanted to retrieve information about the current user, you would make an HTTP `GET` request as below:

```
GET /v2.5/me HTTP/1.1
Host: graph.facebook.com
```

Most API calls must be signed with an access token. You can determine which permissions are needed in this access token by looking at the Graph API reference for the node or edge that you wish to read. You can also use the Graph API Explorer to quickly generate tokens in order to play about with the API and discover how it works.

The `/me` node is a special endpoint that translates to the `user_id` of the person (or the `page_id` of the Facebook Page) whose access token is currently being used to make the API calls. If you had a user access token, you could retrieve all of a user's photos by using:

```
GET graph.facebook.com
  /me/photos
```

REST APIs

The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are available in JSON.

If your intention is to monitor or process Tweets in real-time, consider using the Streaming API instead.

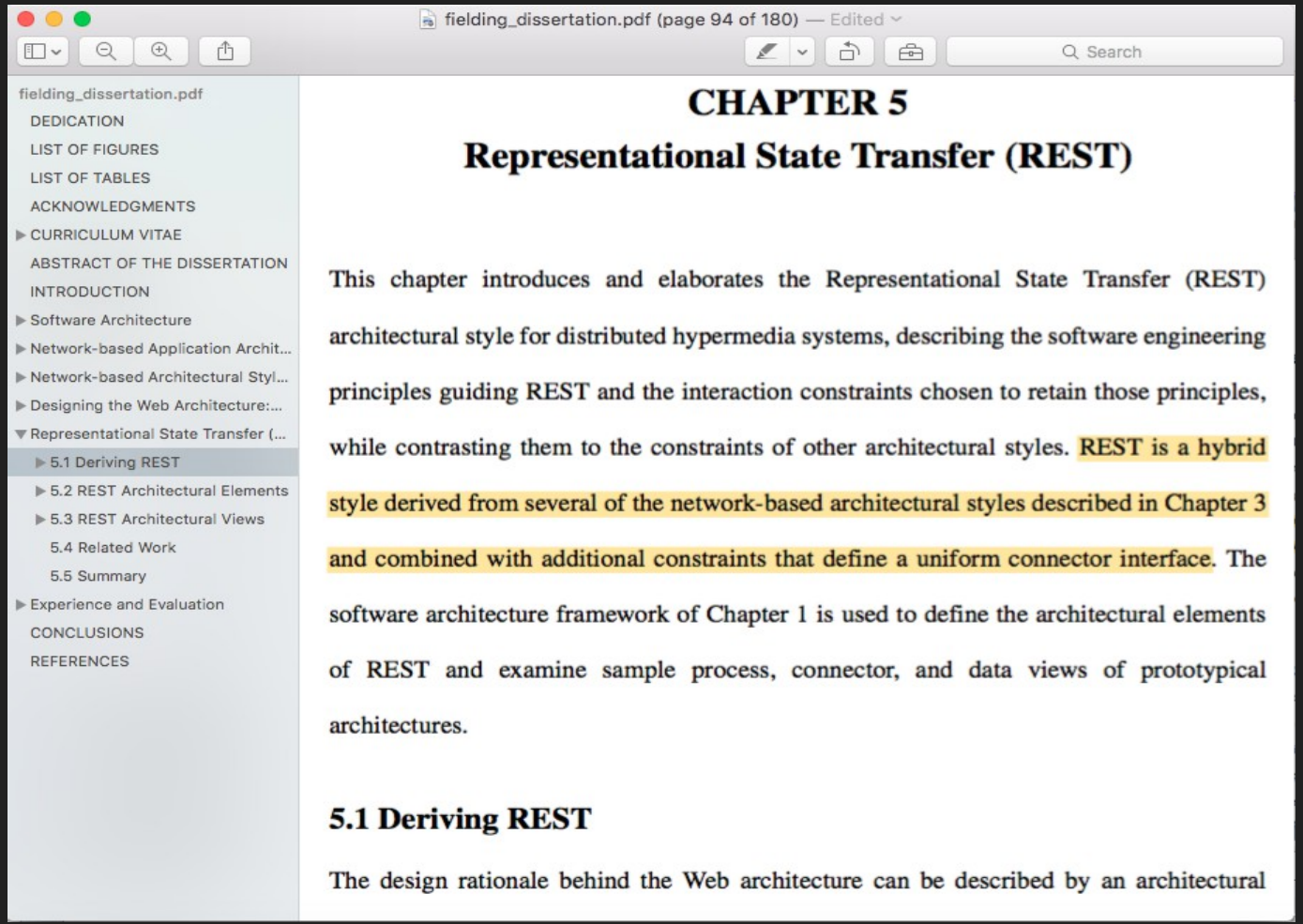
Overview

Below are the documents that will help you get going with the REST APIs as quickly as possible

- API Rate Limiting
- API Rate Limits
- Working with Timelines
- Using the Twitter Search API
- Uploading Media
- Multiple Media Entities in Statuses


Discovery


http
http://
http
http
http
http



https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm


http
http://
http
http
http
http


http://


http://
THE
APACHE®
SOFTWARE FOUNDATION

Some Theory

Resources

Resources
URI

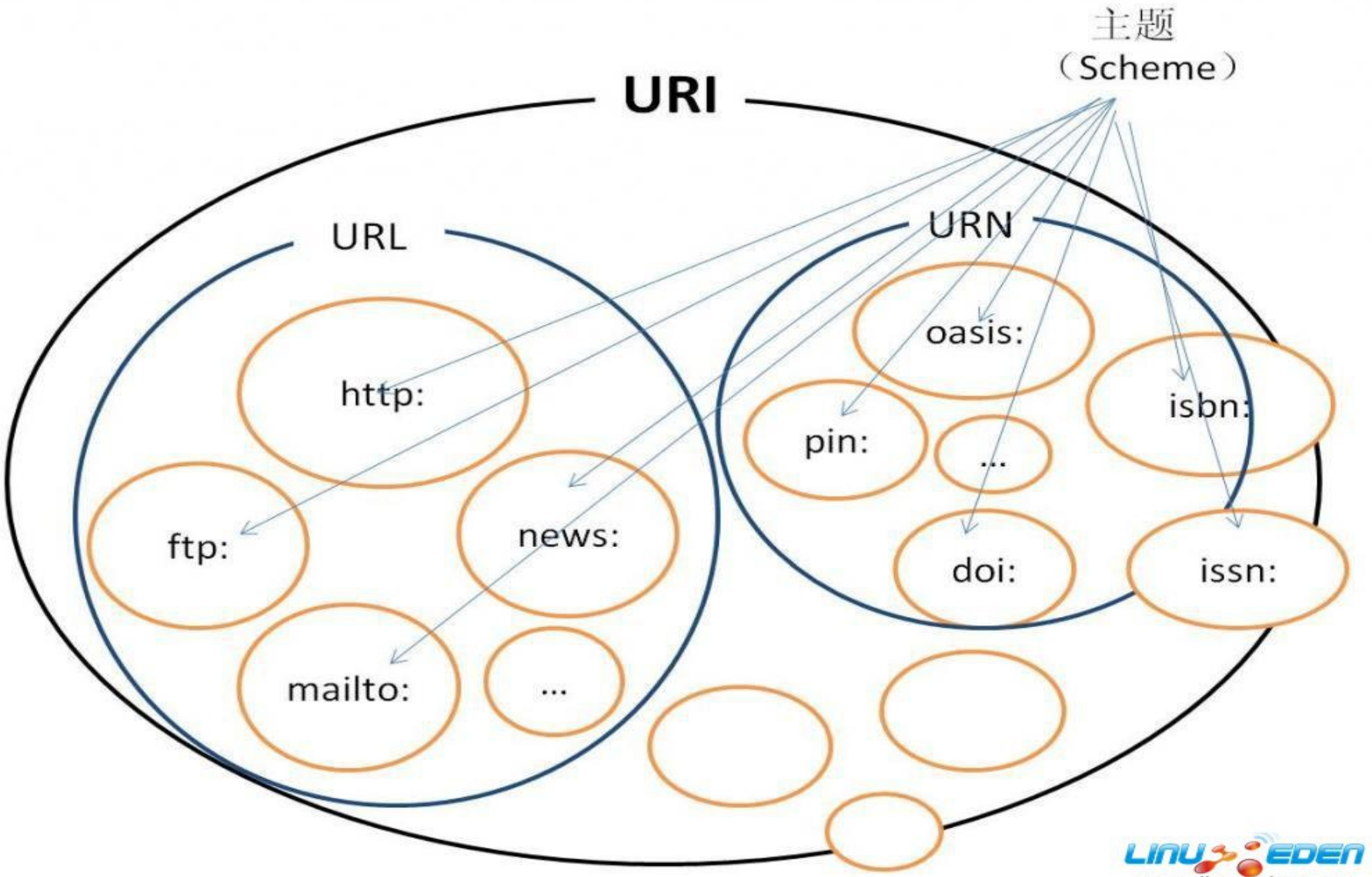
Resources
URI
Representation

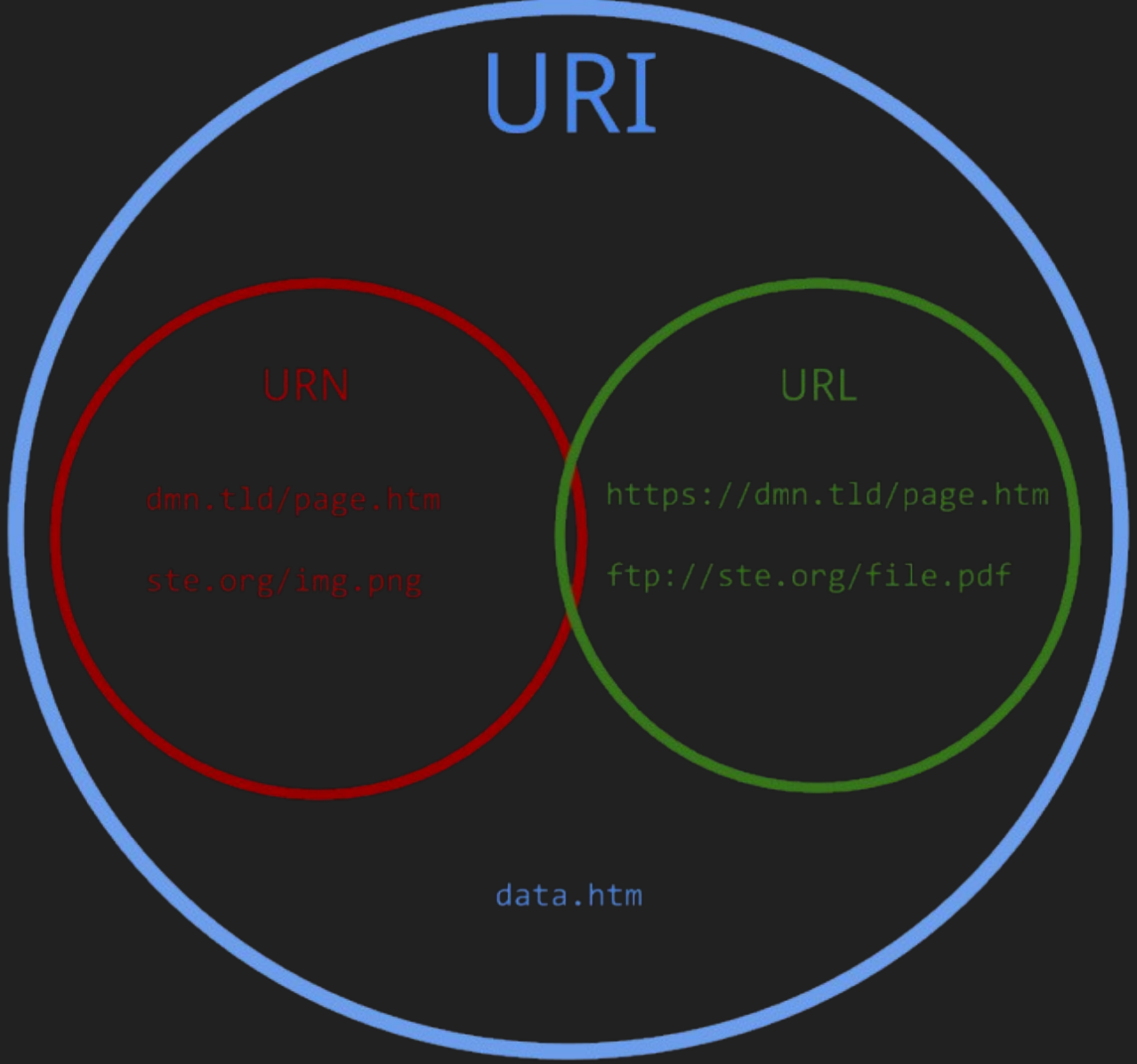
Resources
URI
Representation
State Transfer

Resources


URI =>

URI
Unified Resource identifier




URI
URN
dmn.tld/page.htm
ste.org/img.png
URL
https://dmn.tld/page.htm
ftp://ste.org/file.pdf
data.htm

Examples of absolute URIs [edit]

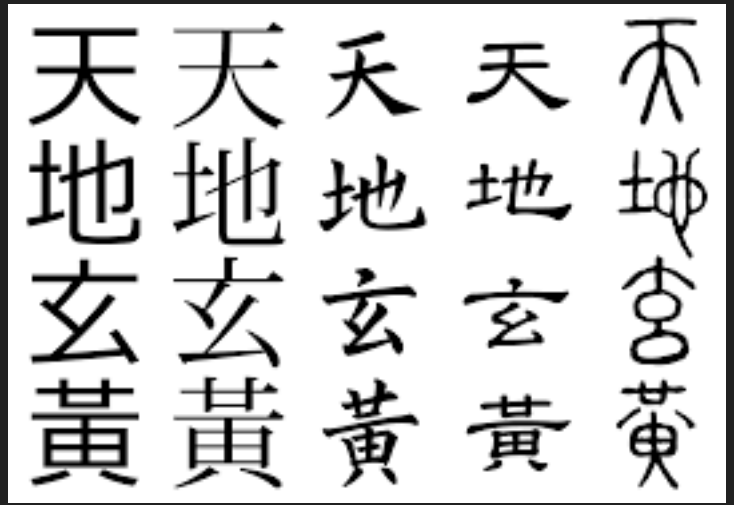
- https://example.org/absolute/URI/with/absolute/path/to/resource.txt
- ftp://example.org/resource.txt
- urn:ISSN:1535–3613

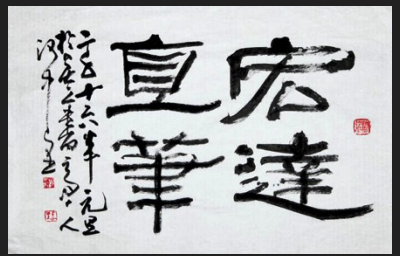
Examples of URI references [edit]

- https://example.org/absolute/URI/with/absolute/path/to/resource.txt
- //example.org/scheme-relative/URI/with/absolute/path/to/resource.txt
- /relative/URI/with/absolute/path/to/resource.txt
- relative/path/to/resource.txt
- ../../../resource.txt
- ./resource.txt#frag01
- resource.txt
- #frag01

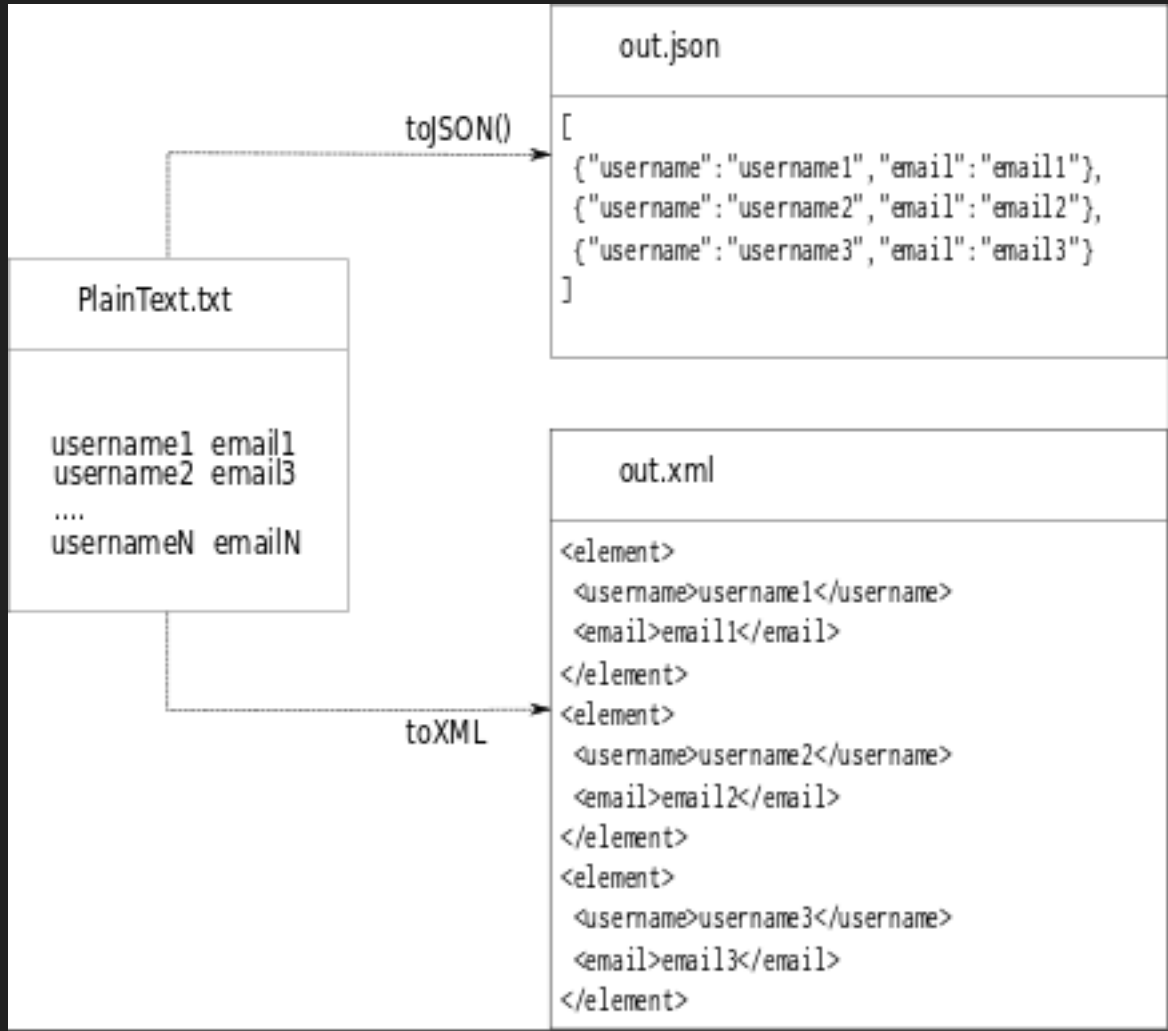
Representation

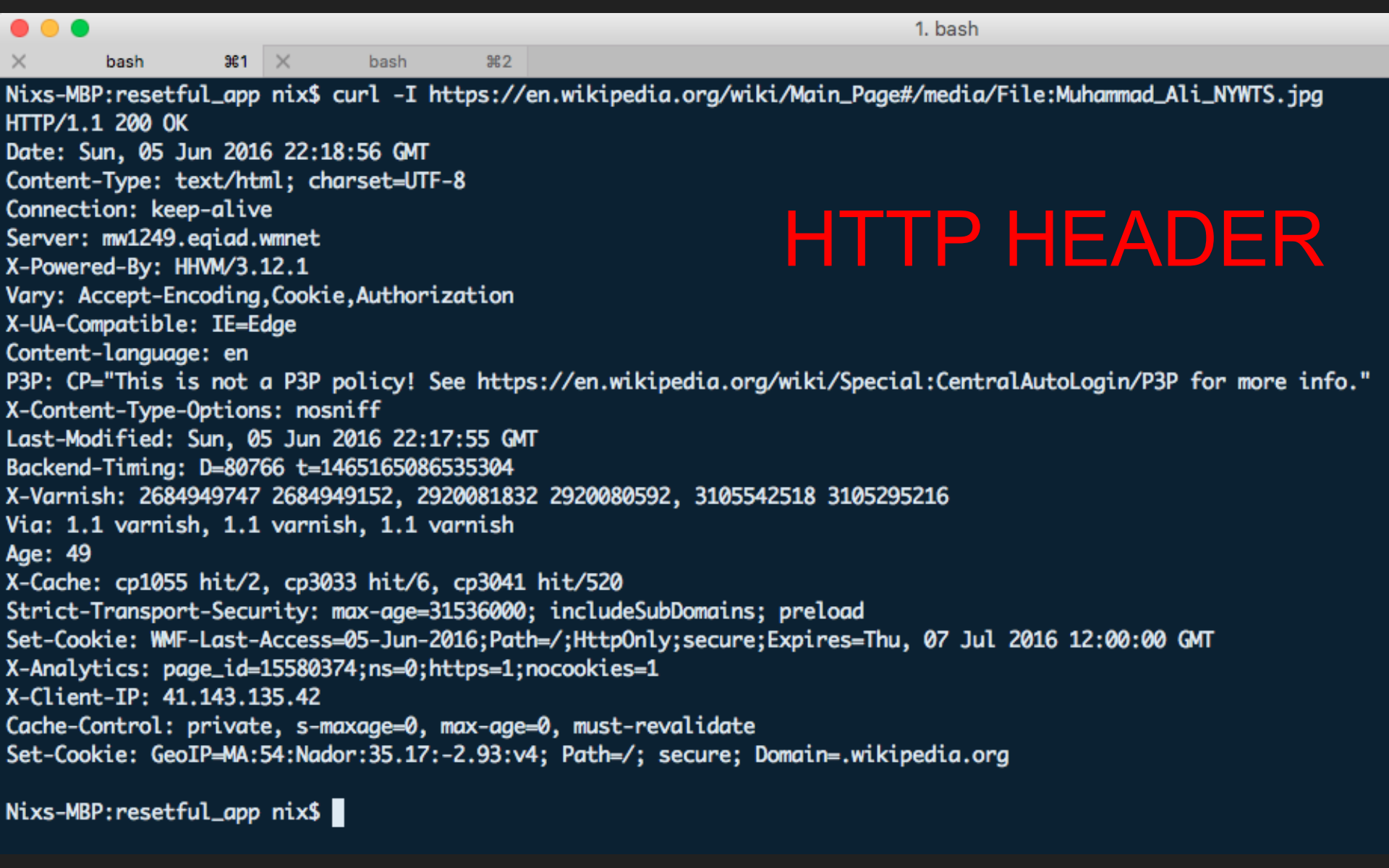

NOTE: PRACTICE ON THIS SOLID PAGE & REST ON THE OUTLINE PAGE. MAKE ENOUGH COPIES. SUBMIT 3 SELECTED ONES OF OUTLINE!


天地玄黄
天地玄黄
天地玄黄
天地玄黄


宏達
真筆

Text
TO
{JSON}


out.json
toJSON()
[
{"username":"username1","email":"email1"},
{"username":"username2","email":"email2"},
{"username":"username3","email":"email3"}
]
PlainText.txt
username1 email1
username2 email3
....
usernameN emailN
out.xml
toXML
<element>
<username>username1</username>
<email>email1</email>
</element>
<element>
<username>username2</username>
<email>email2</email>
</element>
<element>
<username>username3</username>
<email>email3</email>
</element>


1. bash
bash ⌘1
bash ⌘2
Nixs-MBP:resetful_app nix$ curl -I https://en.wikipedia.org/wiki/Main_Page#/media/File:Muhammad_Ali_NYWTS.jpg
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2016 22:18:56 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Server: mw1249.eqiad.wmnet
X-Powered-By: HHVM/3.12.1
Vary: Accept-Encoding,Cookie,Authorization
X-UA-Compatible: IE=Edge
Content-language: en
P3P: CP="This is not a P3P policy! See https://en.wikipedia.org/wiki/Special:CentralAutoLogin/P3P for more info."
X-Content-Type-Options: nosniff
Last-Modified: Sun, 05 Jun 2016 22:17:55 GMT
Backend-Timing: D=80766 t=1465165086535304
X-Varnish: 2684949747 2684949152, 2920081832 2920080592, 3105542518 3105295216
Via: 1.1 varnish, 1.1 varnish, 1.1 varnish
Age: 49
X-Cache: cp1055 hit/2, cp3033 hit/6, cp3041 hit/520
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Set-Cookie: WMF-Last-Access=05-Jun-2016;Path=/;HttpOnly;secure;Expires=Thu, 07 Jul 2016 12:00:00 GMT
X-Analytics: page_id=15580374;ns=0;https=1;nocookies=1
X-Client-IP: 41.143.135.42
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Set-Cookie: GeoIP=MA:54:Nador:35.17:-2.93:v4; Path=/; secure; Domain=.wikipedia.org
Nixs-MBP:resetful_app nix$
HTTP HEADER

State Transfer


IBM
Blue Gene/P

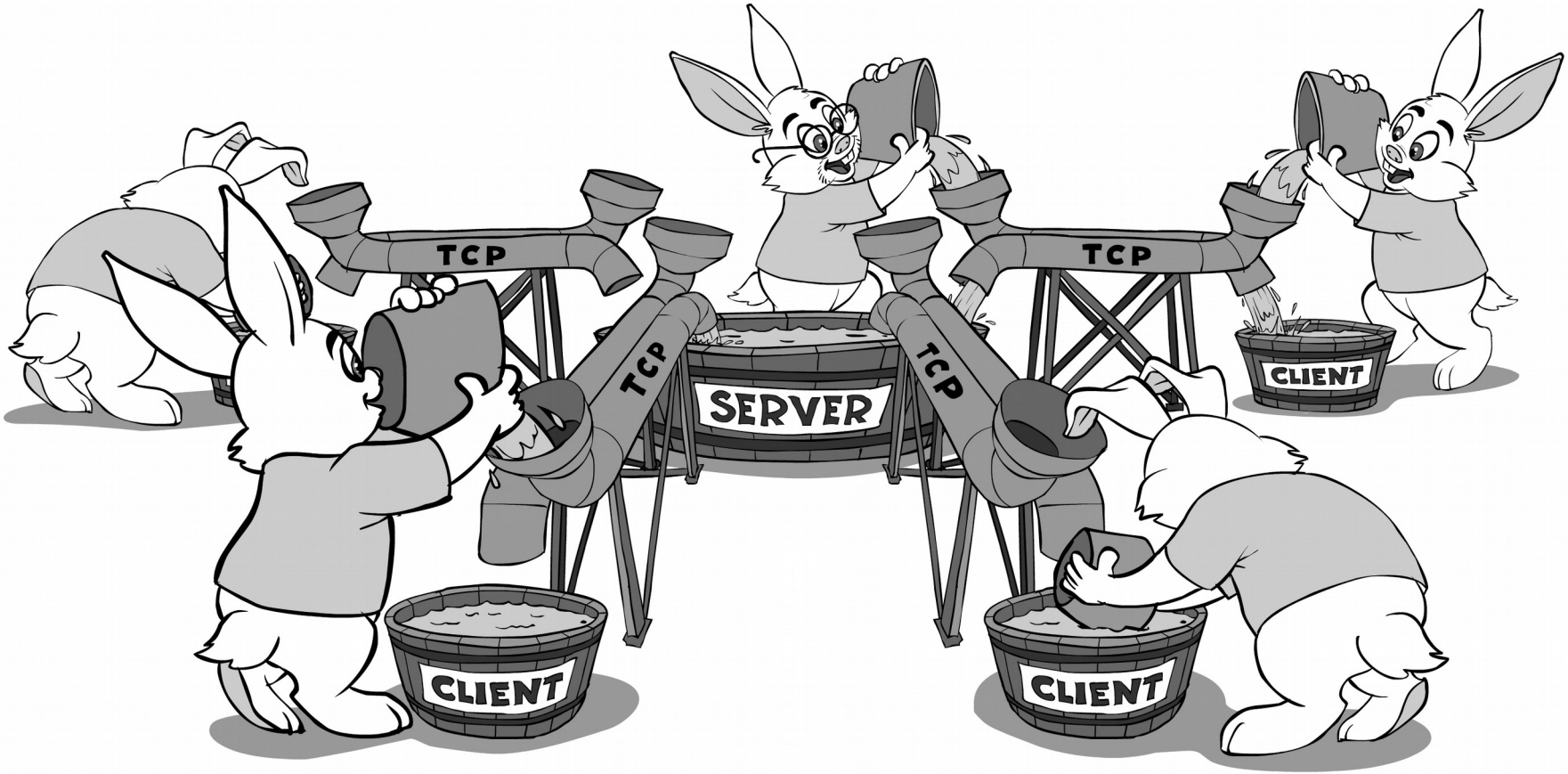

IBM
Blue Gene/P


IBM
Blue Gene/P

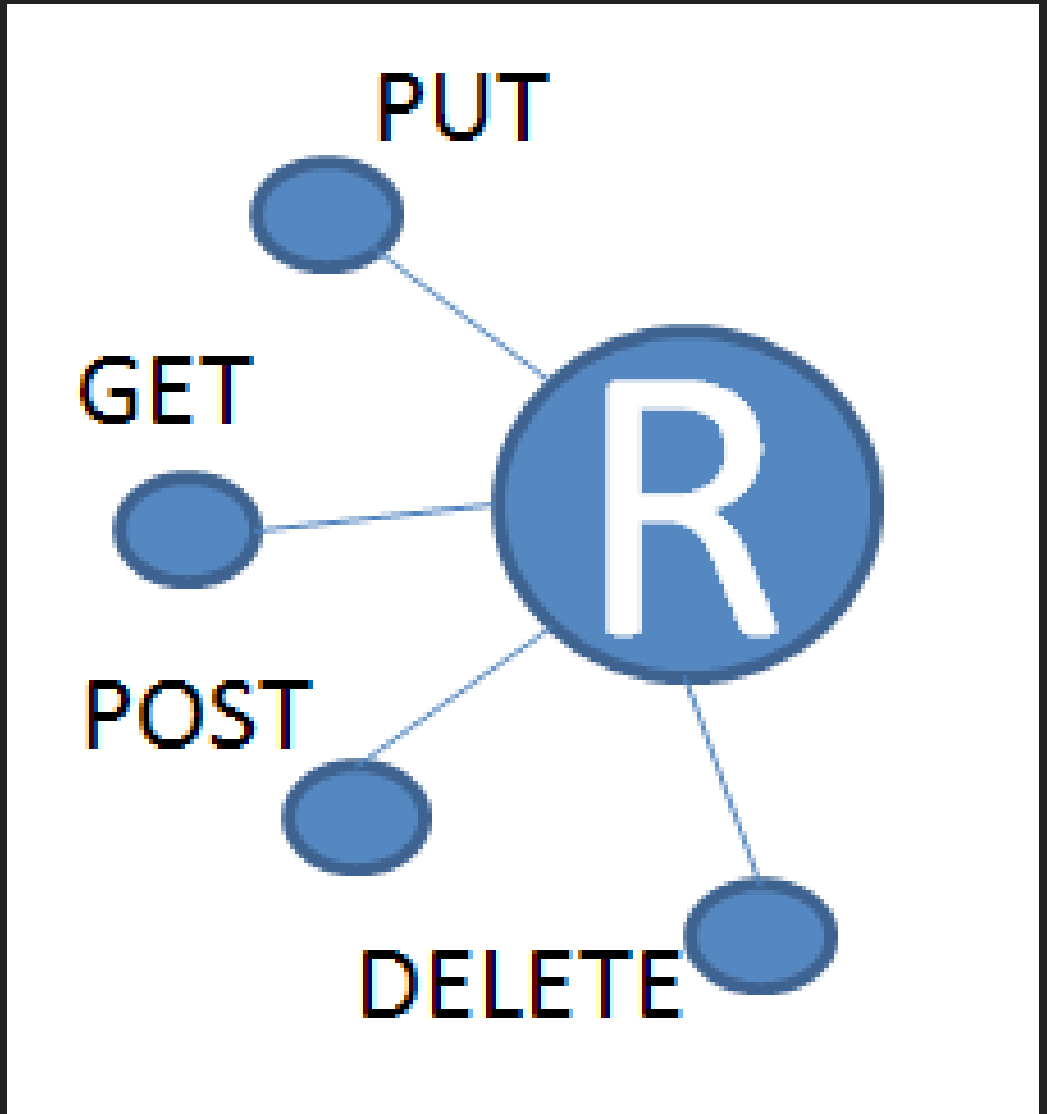

cURL

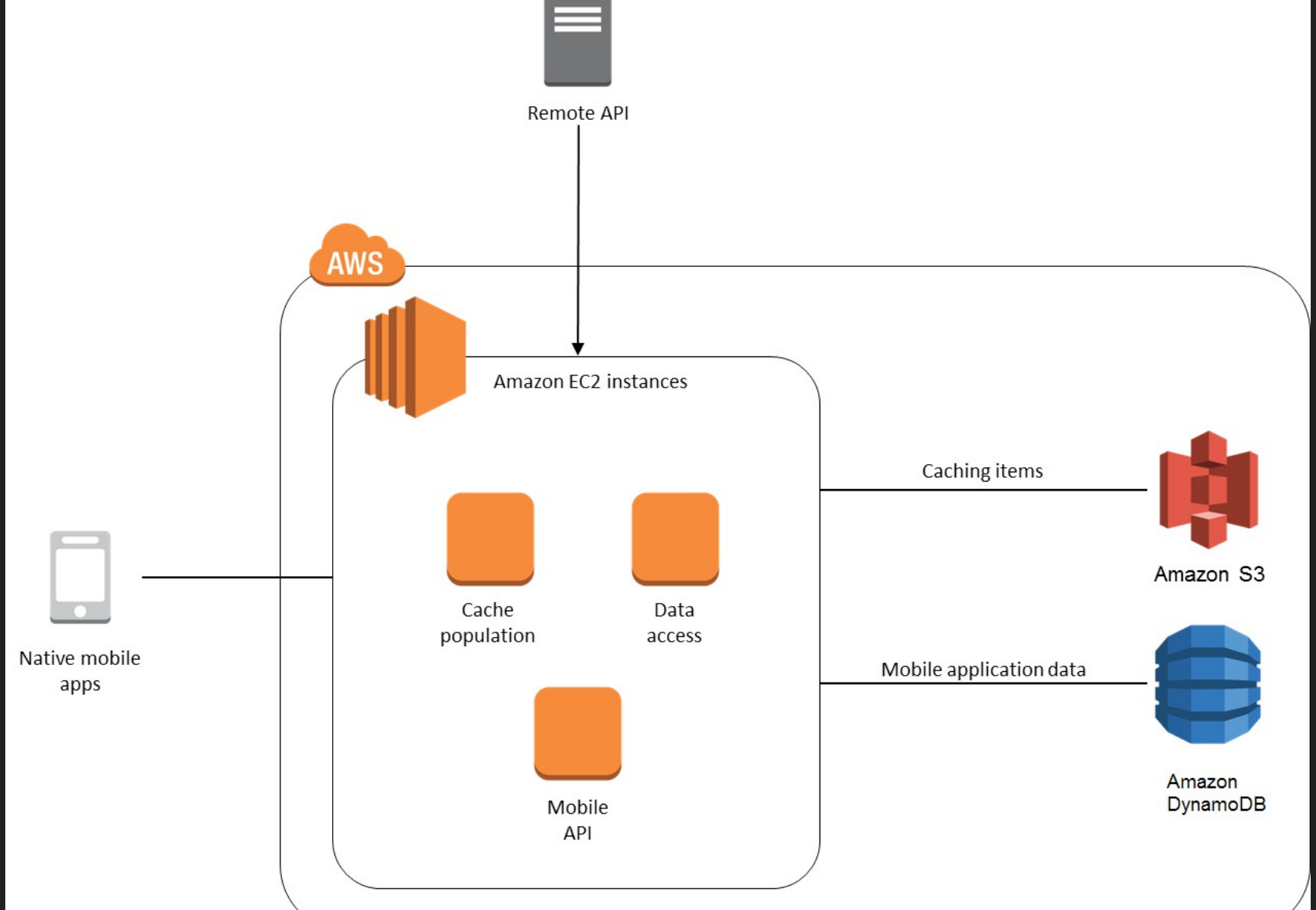

http://

IS Stateless


TCP
TCP
TCP
TCP
SERVER
CLIENT
CLIENT
CLIENT

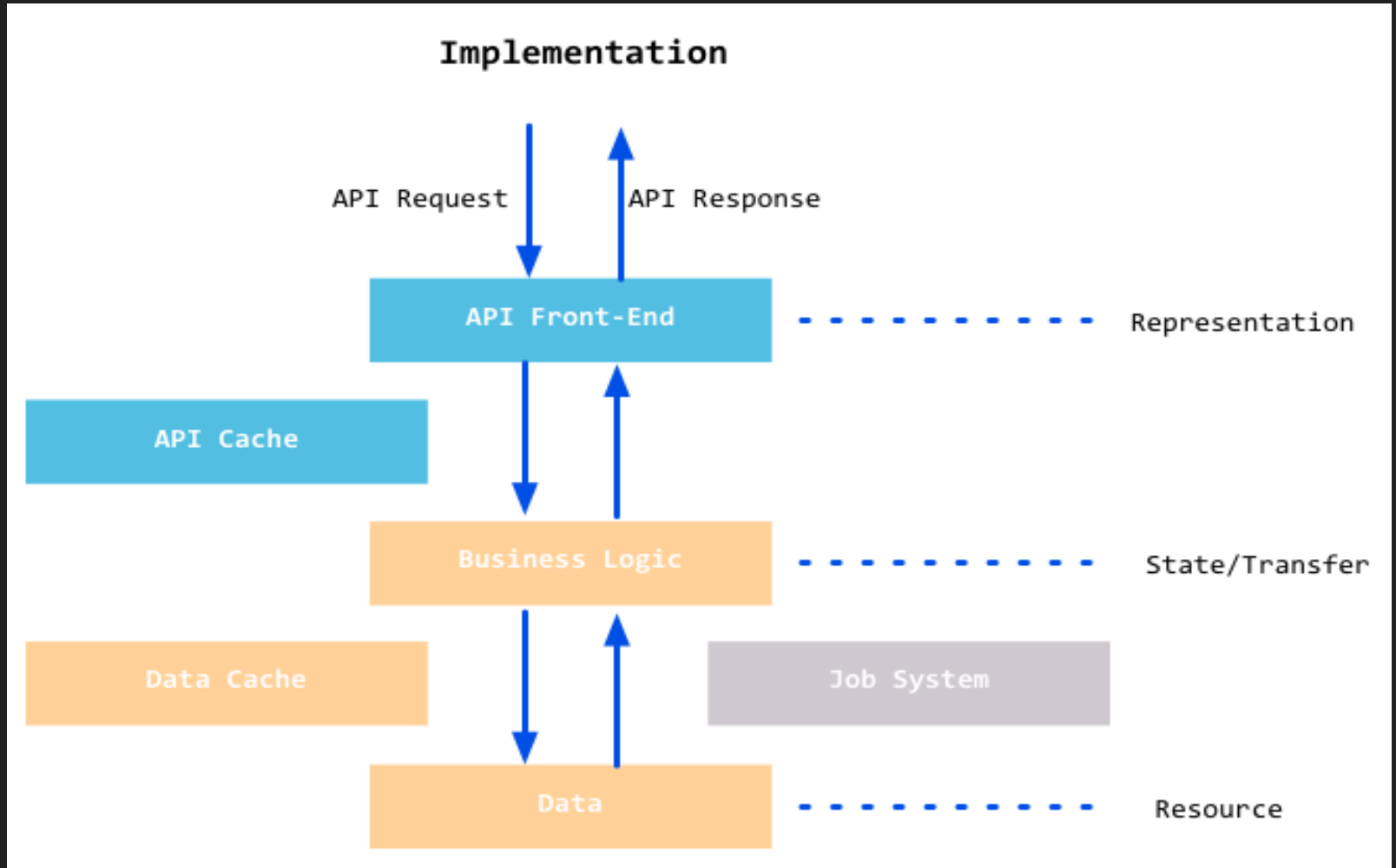

NATIONAL SECURITY
ALL
YOUR
DATA
AGENCY


PUT
GET
R
POST
DELETE

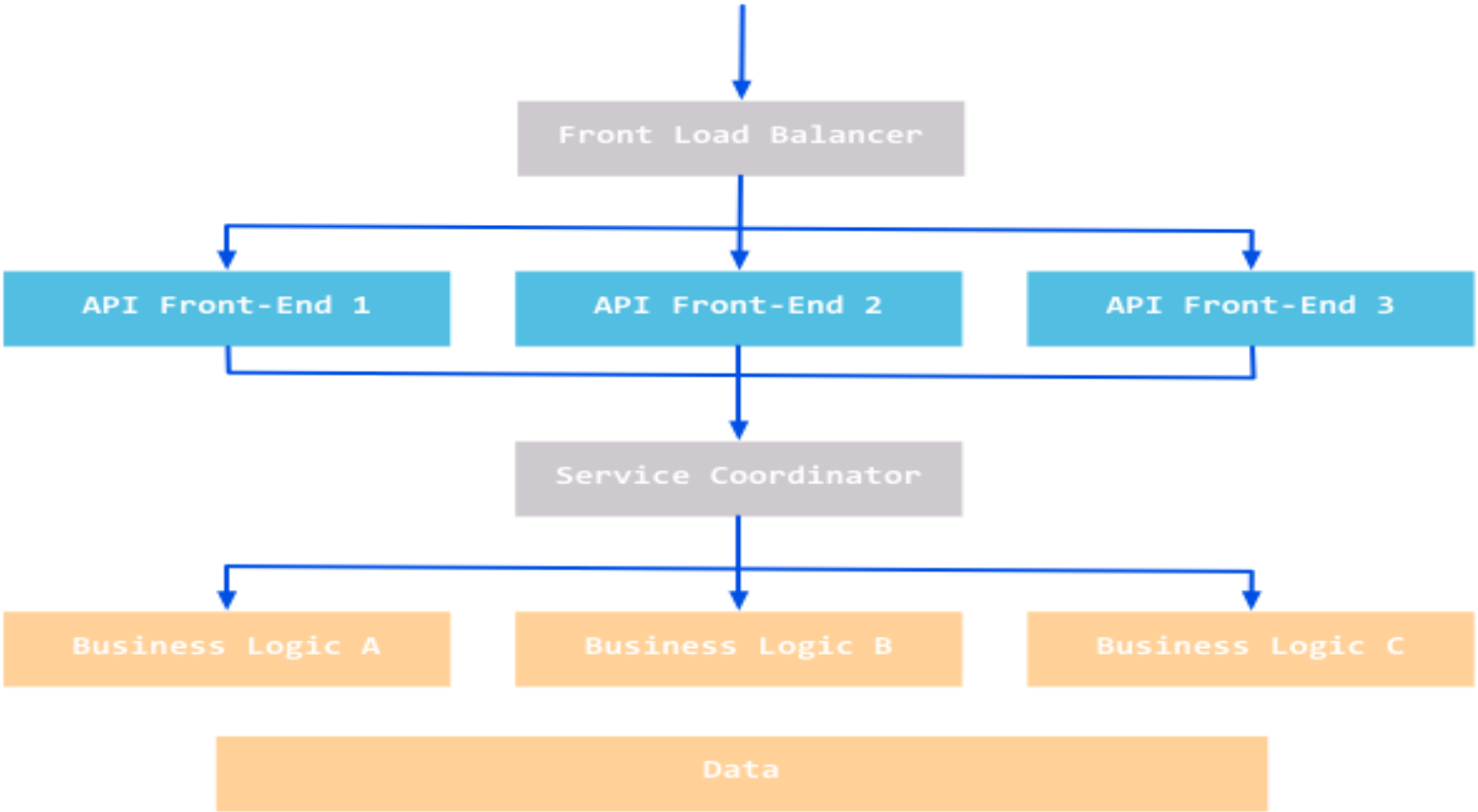

Remote API
AWS
Amazon EC2 instances
Caching items
Amazon S3
Native mobile apps
Cache population
Data access
Mobile application data
Amazon DynamoDB
Mobile API

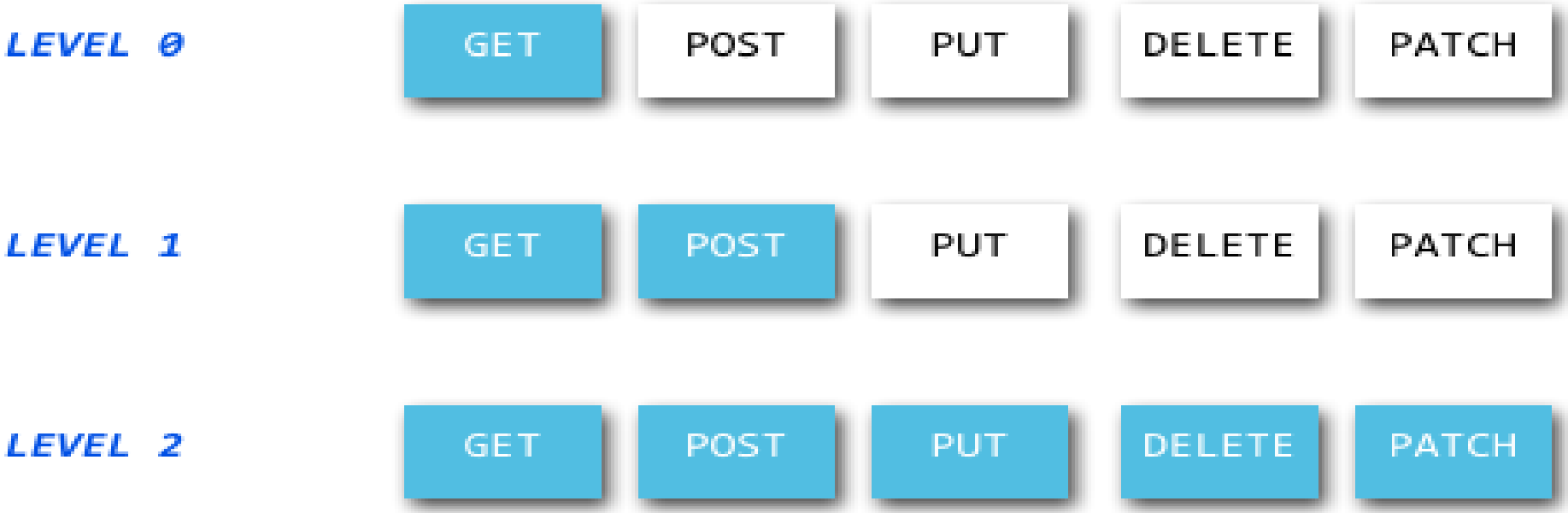

A


elixir
M2U51264DS8HC3G-5T
512MB DDR-400MHz-CL3
PC3200U-30331
Warranty Void If Removed
0542.MN05A1107A.5YM0410N08.XX.CN
elixir
M2U51264DS8H
512MB DDR-40
PC3200U-303
Warranty Void
0542.MN05A11

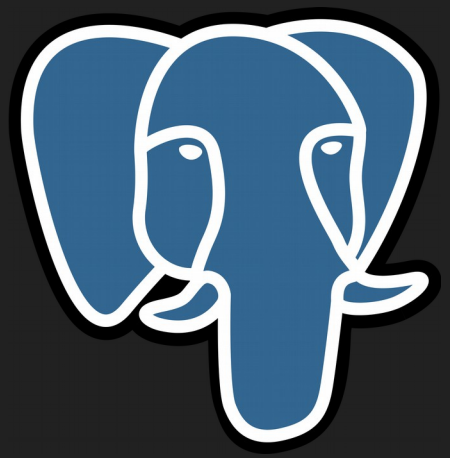

Implementation
API Request
API Response
API Front-End
Representation
API Cache
Business Logic
State/Transfer
Data Cache
Job System
Data
Resource

Scaling


Front Load Balancer
API Front-End 1
API Front-End 2
API Front-End 3
Service Coordinator
Business Logic A
Business Logic B
Business Logic C
Data


LEVEL 0
GET
POST
PUT
DELETE
PATCH
LEVEL 1
GET
POST
PUT
DELETE
PATCH
LEVEL 2
GET
POST
PUT
DELETE
PATCH

Go & REST



PostgreSQL

RESTful Routes

الرابط	عملية بروتوكول HTTP	الحدث المرافق	ملاحظات
/customers	GET	customers	إظهار لائحة الزبائن
/customers/show?uid	GET	customersShow	إظهار زبون معين بناءا على uid
/customers/create	POST	customersCreate	إضافة زبون

Localhost:3000/customers

```
52
53 //Implement the GET Method ( list all Customers from The DB )
54 func customers(w http.ResponseWriter, r *http.Request) {
55
56         if r.Method != "GET" {
57                 http.Error(w, http.StatusText(405), 405)
58                 log.Fatal(w)
59                 return
60         }
61
62         //Building The Query
63
64         query := "SELECT * FROM reset_demo.customer"
65         //Query the DB
66         rows, err := dbConnection.Query(query)
67
```

Localhost:3000/customers/show?uid=2

```
111
112 func customersShow(w http.ResponseWriter, r *http.Request) {
113
114         //Check the Method Sent ! GET  else Die
115         if r.Method != "GET" {
116                 http.Error(w, http.StatusText(405), 405)
117                 log.Fatal(w)
118                 return
119         }
120
121         //Get the Parameter form the URL
122         uid := r.FormValue("uid")
123         //if This uid is empty or nil
124         if uid == "" {
125                 http.Error(w, http.StatusText(400), 400)
126                 log.Fatal(w)
127                 return
128
129         }
130
131         //Build a query based on the UID
132         query := "SELECT * FROM reset_demo.customer WHERE uid =$1"
```

Curl -i -X POST -d “uid=90&name=salah&email=salah@gmail.com” localhost:3000/customers/create

```
160 func customersCreate(w http.ResponseWriter, r *http.Request) {
161
162         // Create   -> Post   uid=xxx & name=something & email= something@gmail.com
163
164         // Check If This r.Method=Post
165
166         if r.Method != "POST" {
167                 http.Error(w, http.StatusText(405), 405)
168                 log.Fatal(w)
169                 return
170
171         }
172
173         // GET uid,name, email
174
175         uid := r.FormValue("uid")
176         name := r.FormValue("name")
177         email := r.FormValue("email")
178
179         //Check if the params are Not Empty
180
```