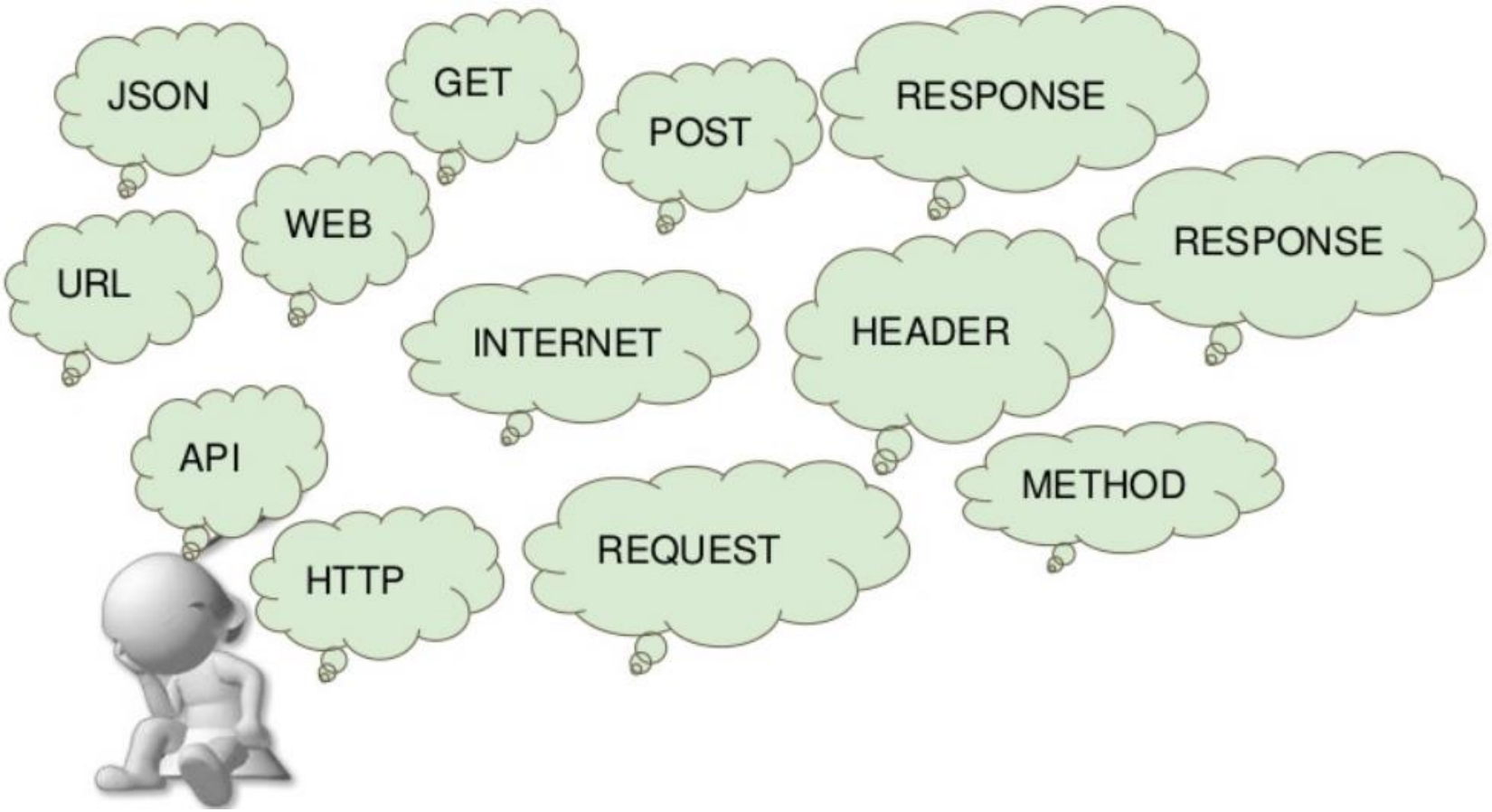
Serviços e Servidores de Rede

Prof. Dr. Eng. Miguel Molina
miguel.molina@ifsp.edu.br

Introdução HTTP REST API

What is it ?


JSON
GET
POST
RESPONSE
WEB
URL
RESPONSE
INTERNET
HEADER
API
METHOD
HTTP
REQUEST

What is it ?

REST means

REpresentational

State

Transfer

REpresentational ? State ? Transfer ?

It represent the state of database at a time. But how???

REST is an architectural style which is based on web-standards and the **HTTP** protocol.

In a REST based architecture everything is a **Resource**.

A resource is accessed via a common interface based on the HTTP standard methods.

You typically have a REST server which provides access to the resources and a REST client which accesses and modifies the REST resources.

REpresentational ? State ? Transfer ?

Every resource should support the HTTP common operations.

Resources are identified by global IDs (which are typically **URIs** or **URLs**).

REST allows that resources have different representations, e.g., text, XML, JSON etc.

Stateless in nature. **Excellent** for **distributed system.**

Stateless components can be freely redeployed if something fails, and they can **scale** to accommodate load changes.

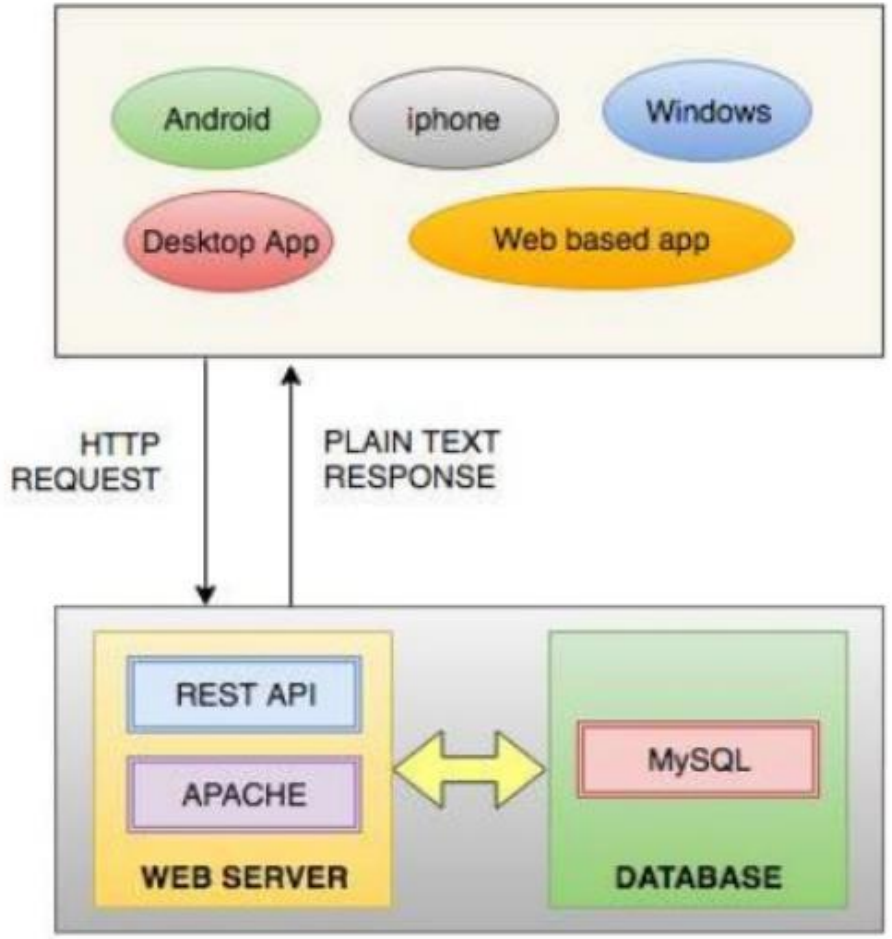
This is because any request can be directed to any instance of a component.

HTTP Methods

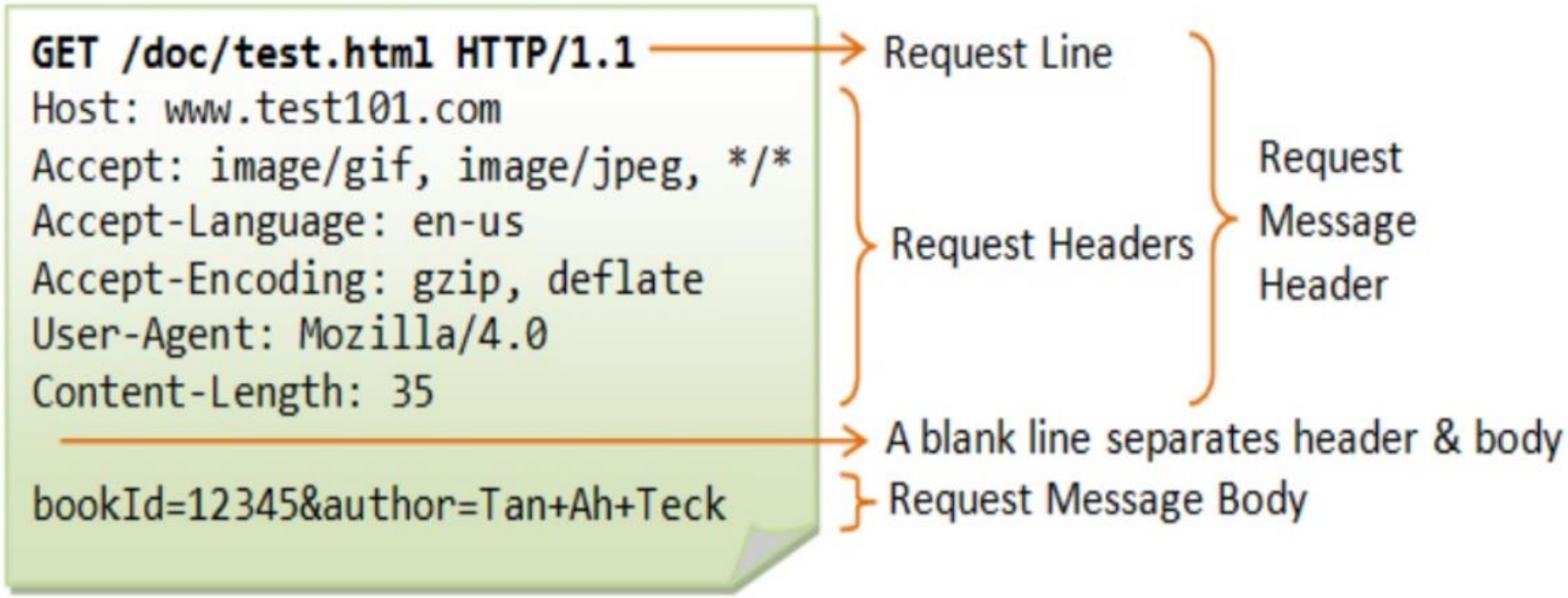
The *PUT*, *GET*, *POST* and *DELETE* methods are typically used in REST based architectures. The following table gives an explanation of these operations:

HTTP Method	CRUD Operation	Description
POST	INSERT	Addes to an existing resource
PUT	UPDATE	Overrides existing resource
GET	SELECT	Fetches a resource. The resource is never changed via a GET request
DELETE	DELETE	Deletes a resource

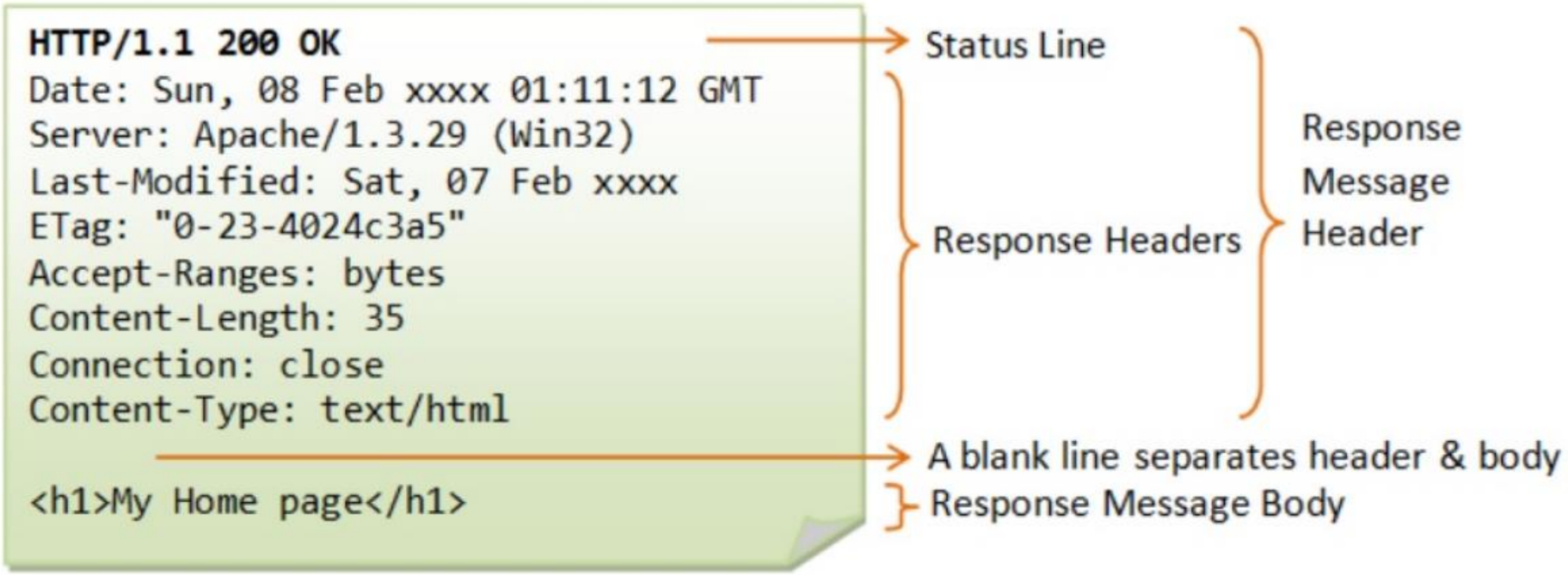
Architecture:



HTTP Request Example:



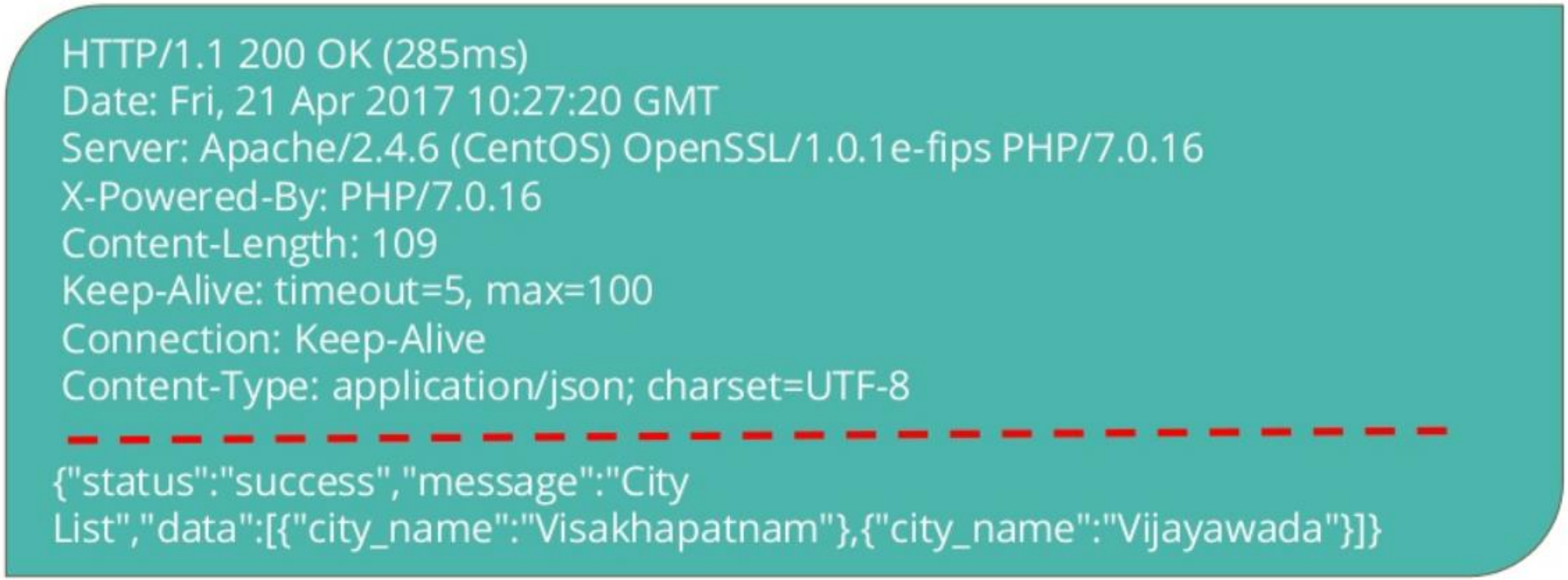
HTTP Response Example:



HTTP REST Request:

HTTP REST API Request

HTTP REST Response:



HTTP REST API Response

HTTP Response Status Code:

1xx	Informational Codes
2xx	Successful Codes
3xx	Redirection Codes
4xx	Client Error Code
5xx	Server Error Codes

List at here: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Points to be noted

For REST APIs

Use Nouns but no verbs in path/URI:

Purpose	Method	Incorrect	Correct
Retrieves a list of users	GET	/getAllCars	/users
Create a new user	POST	/createUser	/users
Delete a user	DELETE	/deleteUser	/users/10
Get balance of user	GET	/getUserBalance	/users/11/balance

Use plural nouns:

Do not mix up singular and plural nouns. Keep it simple and use only plural nouns for all resources.

/cars instead of /car

/users instead of /user

/products instead of /product

/settings instead of /setting

GET method should not alter the state:

Use **PUT, POST** and **DELETE** methods instead of the **GET** method to alter the state.

Do not use **GET** method or Query parameters for state changes:

GET /users/711?activate or

GET /users/711/activate

Use HTTP headers for serialization formats:

Both, client and server need to know which format is used for the communication. The format has to be specified in the HTTP-Header.

Content-Type defines the request format.

Accept defines a list of acceptable response formats.

Handle Errors with HTTP Status code:

200	OK (Everything is working)	403	Forbidden (The server understood the request, but is refusing it or the access is not allowed)
201	OK (New resource has been created)	404	Not found (There is no resource behind the URI)
204	OK (Resource successfully deleted)	405	Method not allowed
400	Bad Request (The request was invalid or cannot be served. The exact error should be explained in the error payload. E.g. „The JSON is not valid“)	408	Request timeout
401	Unauthorized (The request requires an user authentication)	500	Internal server error

Filtering:

Use a unique query parameter for all fields or a query language for filtering.

GET /cars?color=red (Returns a list of red cars)

GET /users?name=tom (Returns a list of users whose name matches tom)

Sorting:

Allow ascending and descending sorting over multiple fields.

GET /cars?sort=-manufacturer,+model

This returns a list of cars sorted by descending manufacturers and ascending models)

Paging:

Use limit and offset. It is flexible for the user and common in leading databases.

The default should be limit=20 and offset=0

GET /cars?offset=10&limit=5

Want to develop REST APIs easily???

Checkout Slim - A microframework for php.

Documentation: https://www.slimframework.com/docs/

Dúvidas???

Obrigado.