Programação na Internet

Turma i52d

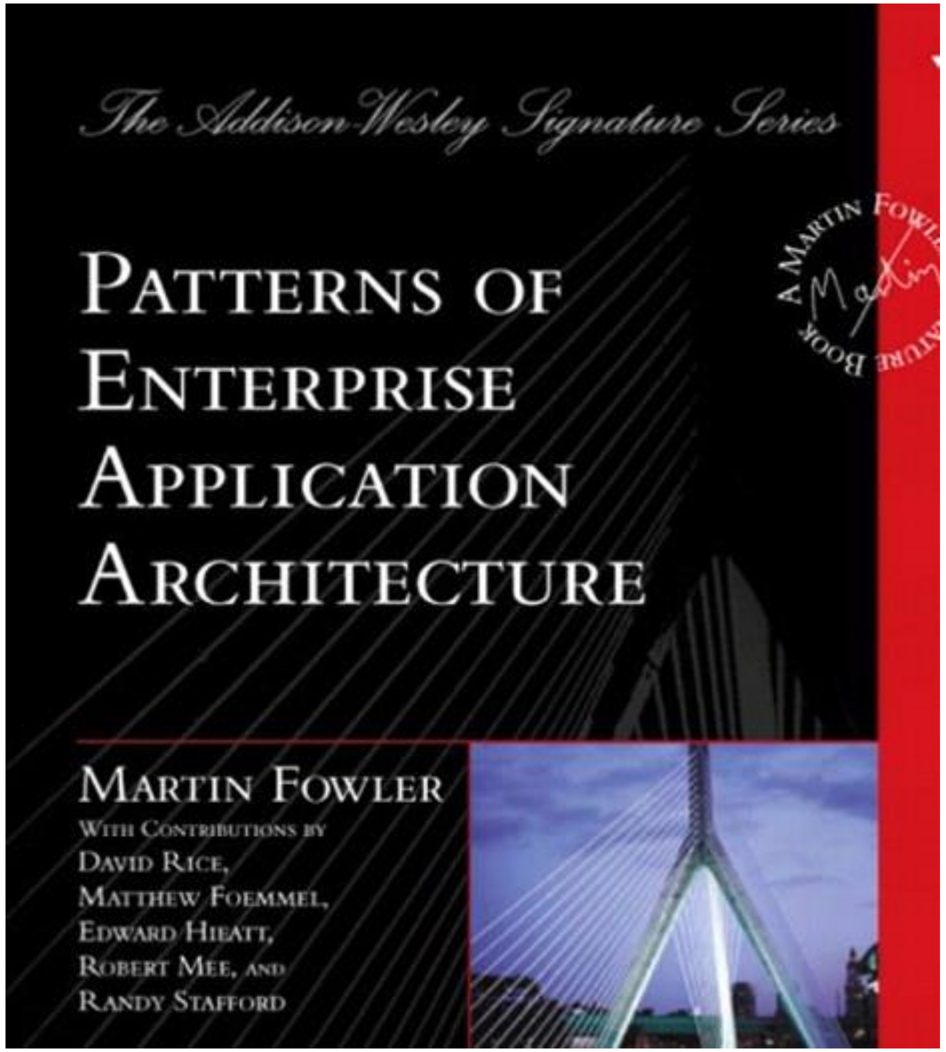
Aula 13

Web App and Routing

App Web = Enterprise Application

Definition of Martin Fowler that

“*Enterprise applications are about the display, manipulation, and storage of large amounts of often complex data and the support or automation of business processes with that data*” [Fowler,2003]



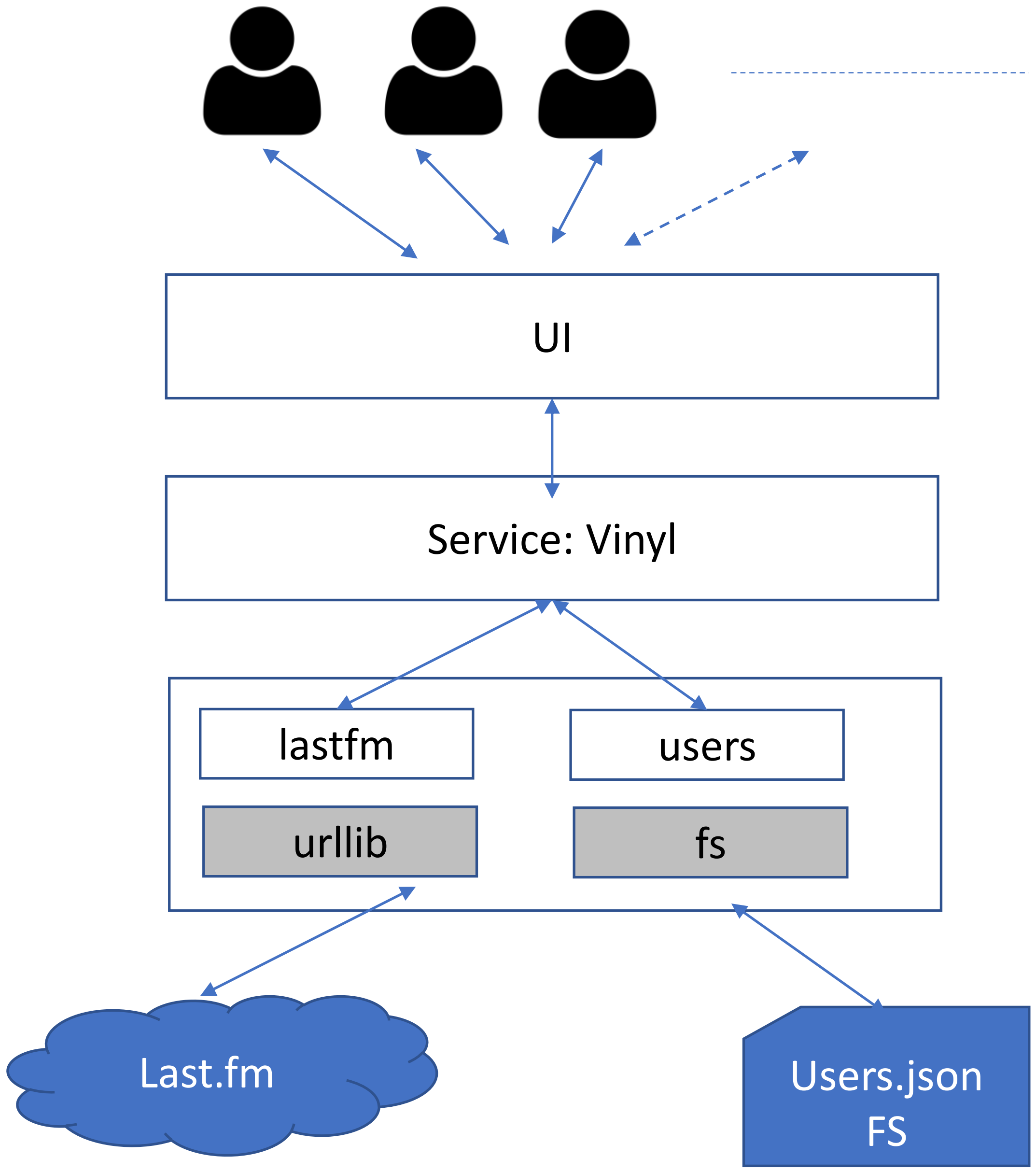
The Addison-Wesley Signature Series

PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE

MARTIN FOWLER

WITH CONTRIBUTIONS BY DAVID RICE, MATTHEW FOEMMEL, EDWARD HIEATT, ROBERT MEE, AND RANDY STAFFORD

Vinyl

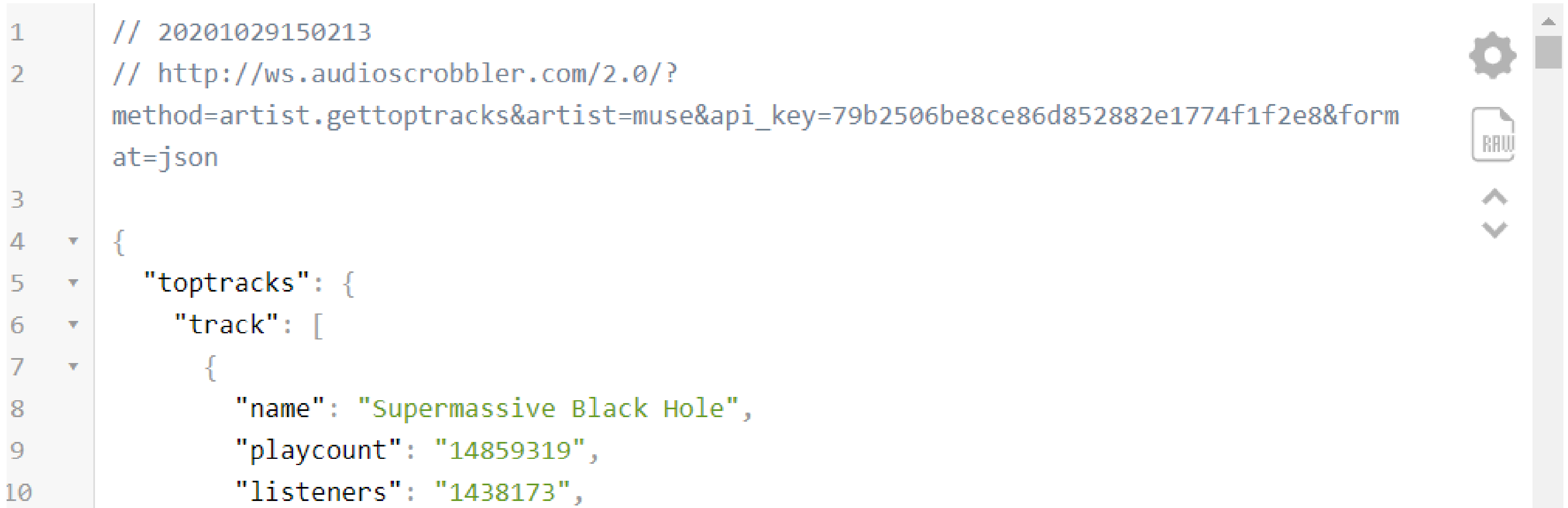

UI
Service: Vinyl
lastfm
users
urllib
fs
Last.fm
Users.json
FS

Last.fm

? Inicio da query string

http://ws.audioscrobbler.com/2.0/?method=artist.gettoptracks&artist=muse&api_key=79b2506be8ce86d852882e1774f1f2e8&format=json

Pares chave=valor



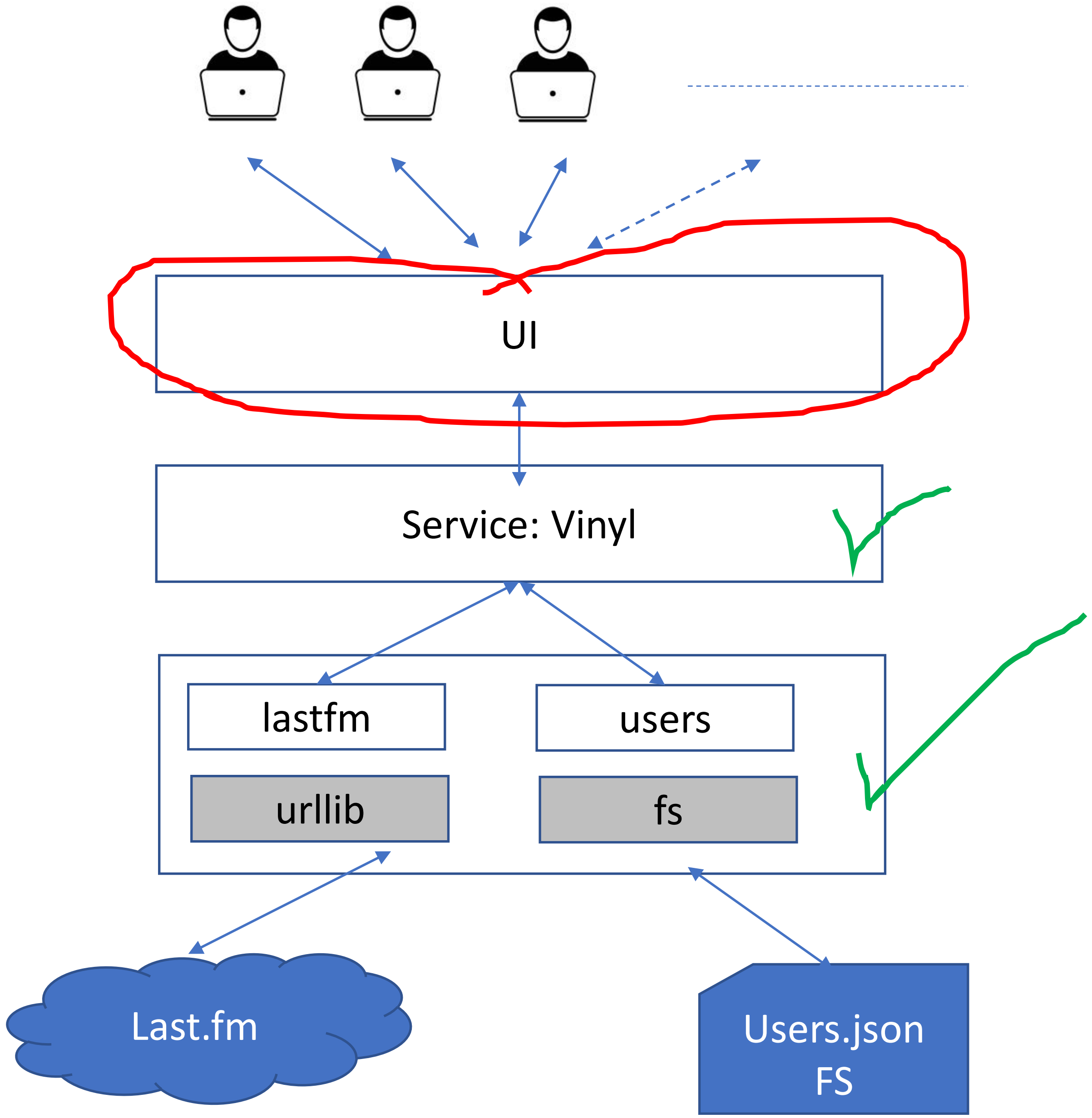
```
// 20201029150213
// http://ws.audioscrobbler.com/2.0/?
method=artist.gettoptracks&artist=muse&api_key=79b2506be8ce86d852882e1774f1f2e8&form
at=json

{
  "toptracks": {
    "track": [
      {
        "name": "Supermassive Black Hole",
        "playcount": "14859319",
        "listeners": "1438173",
```

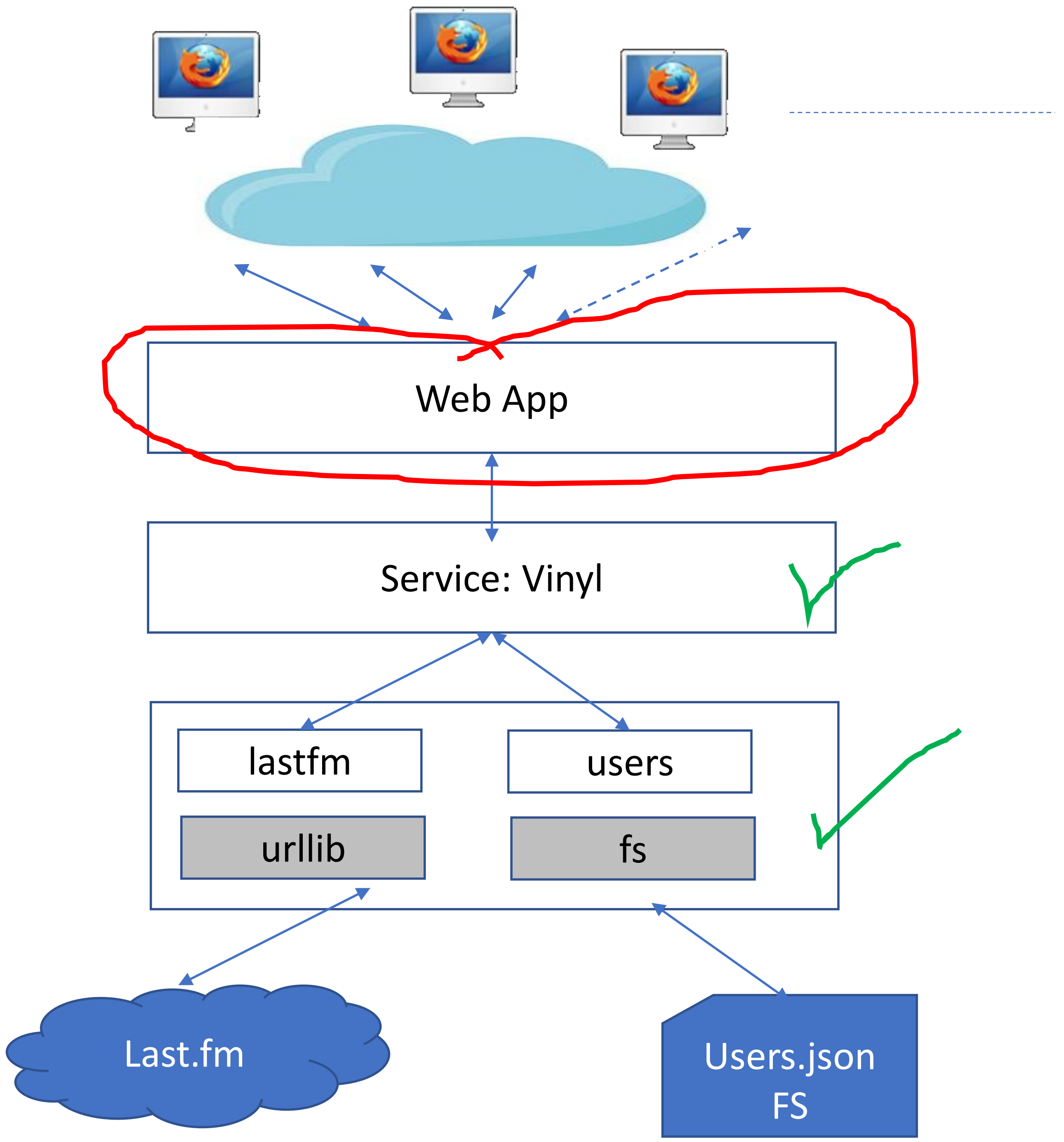
users

```
[
    {
        "username": "gamboa",
        "artists": ["muse", "killers", "new order", "Franz Ferdinand", "Faith no more"]
    },
    {
        "username": “papoila",
        "artists": [“u2", “police"]
    }
]
```

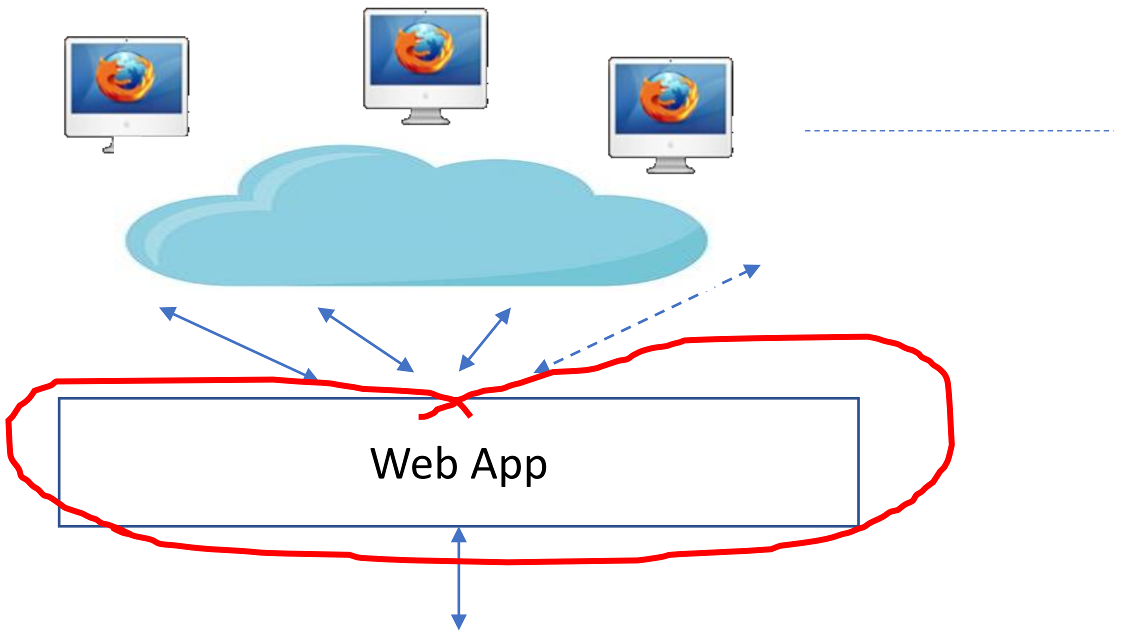
Vinyl


UI
Service: Vinyl
lastfm
users
urllib
fs
Last.fm
Users.json
FS

Vinyl

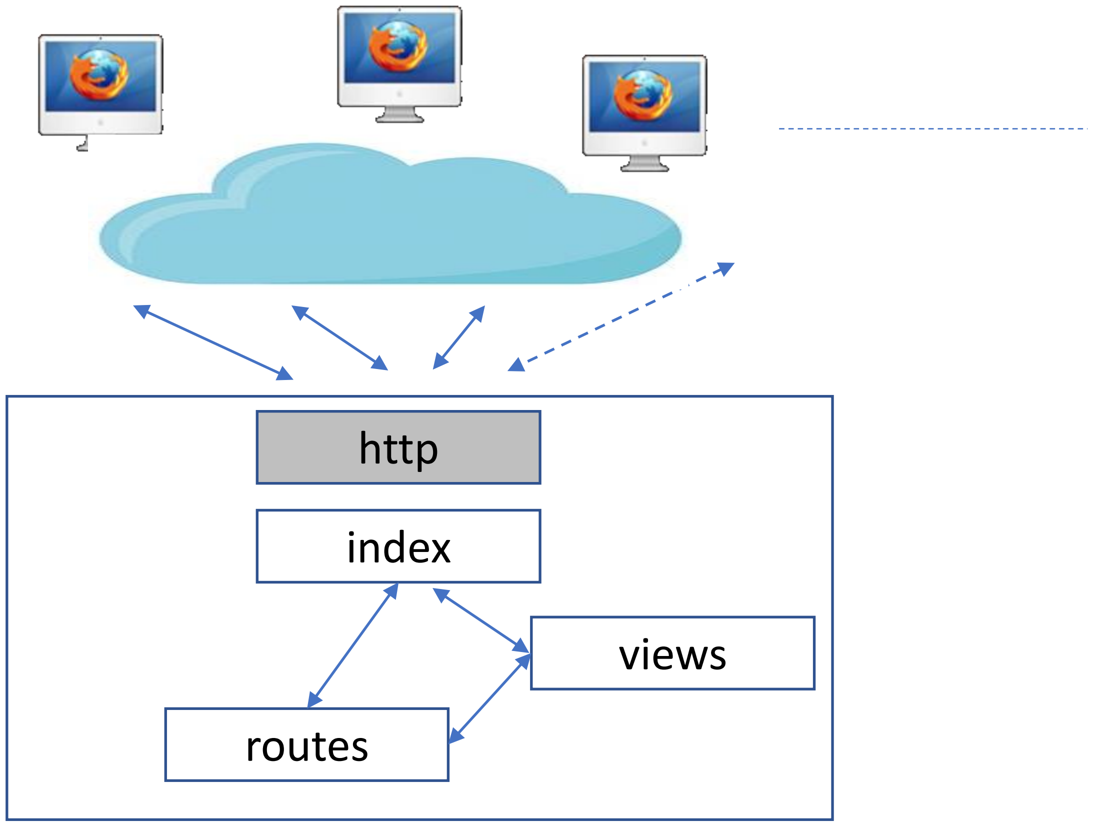

Web App
Service: Vinyl
lastfm
users
urllib
fs
Last.fm
Users.json
FS

Vinyl



- Servidor HTTP
- View (e.g. HTML, JSON, other) – Representation of a Resource
- Routing

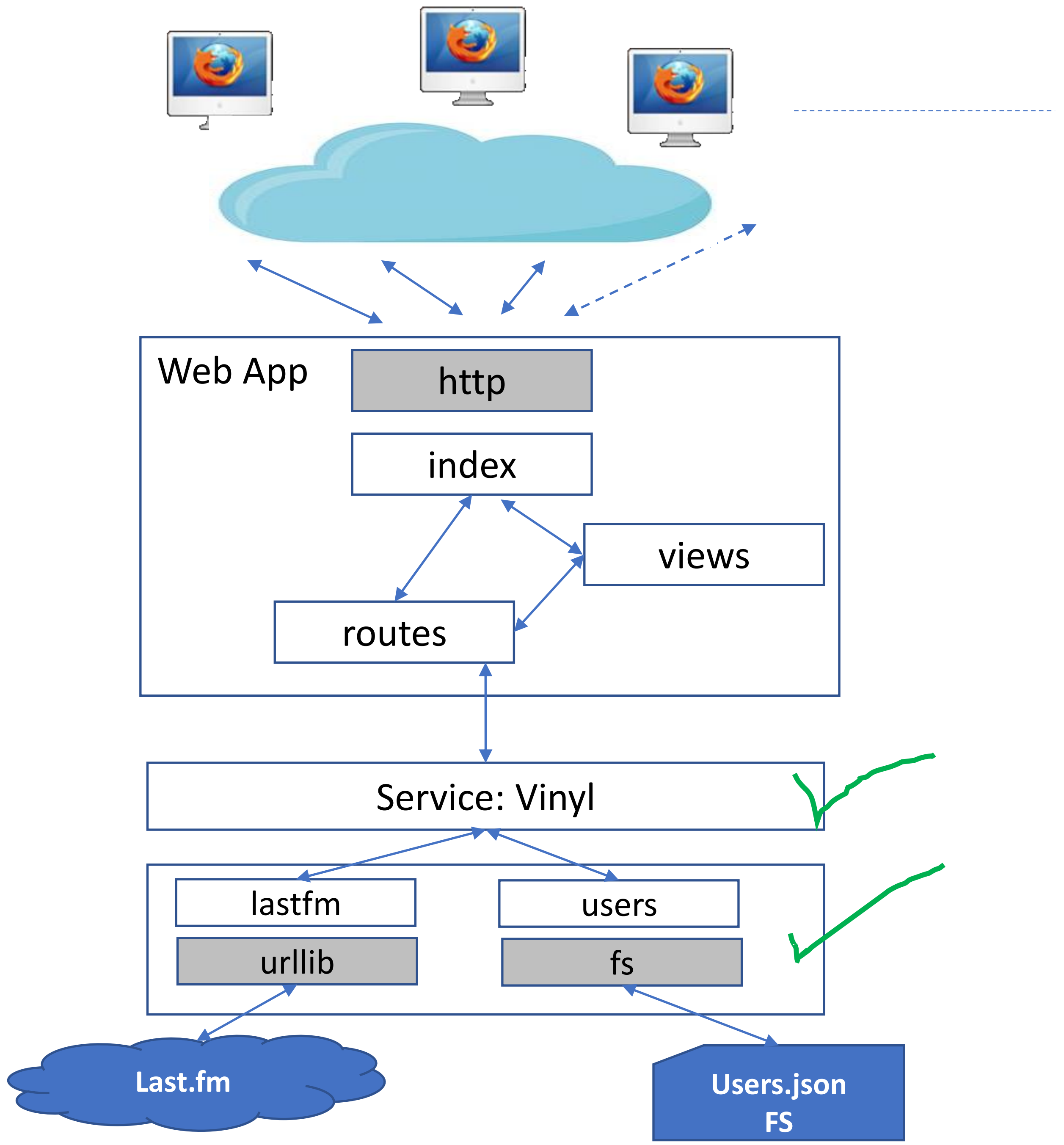
Vinyl



- Servidor HTTP
- View (e.g. HTML, JSON, other) – Representation of a Resource
- Routing

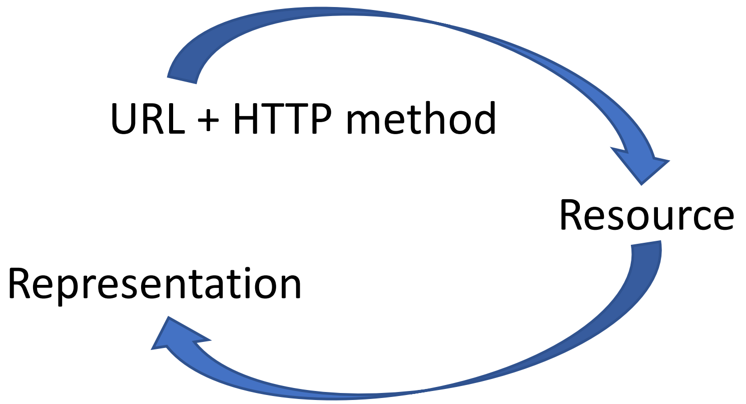
Vinyl

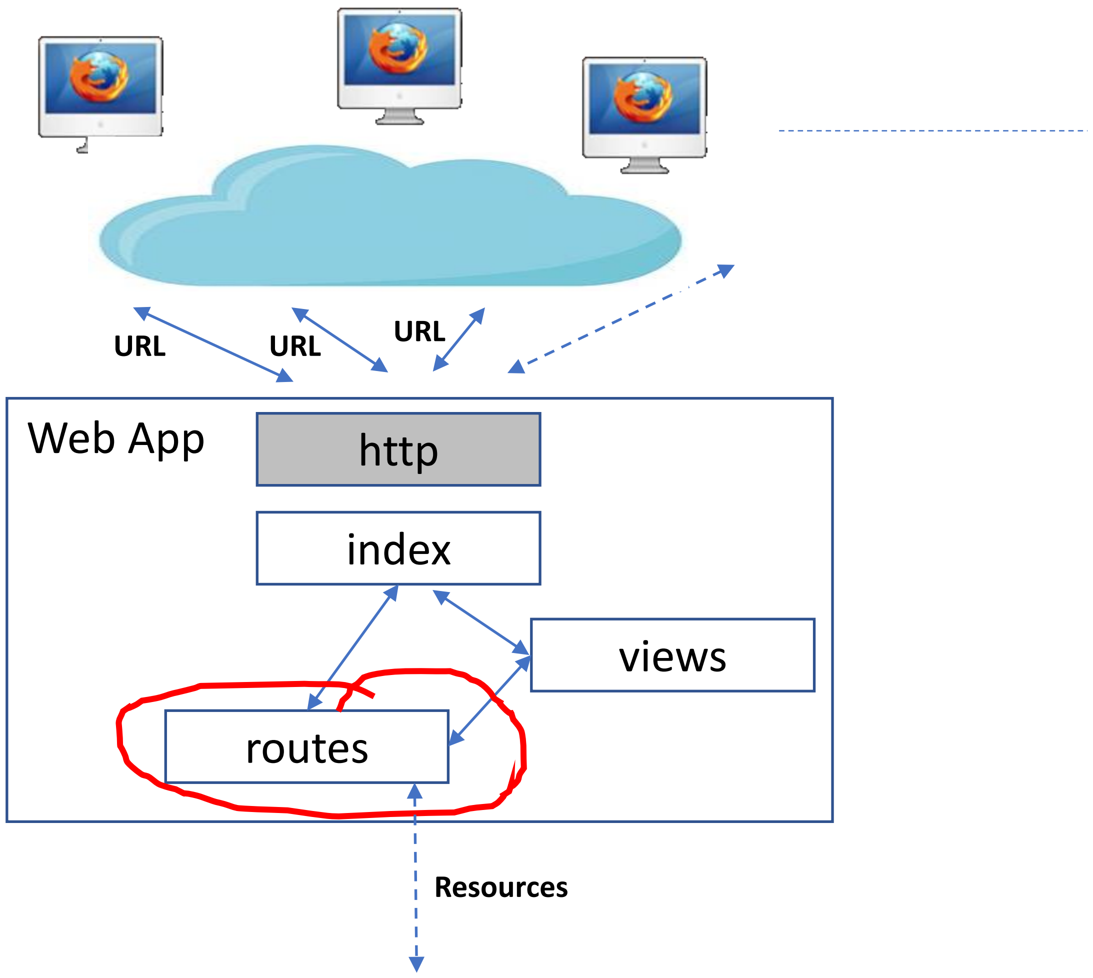
- Servidor HTTP
- View (i.e. JSON)
- Routing



Vinyl

- Servidor HTTP
- View (i.e. JSON)
- **Routing:**

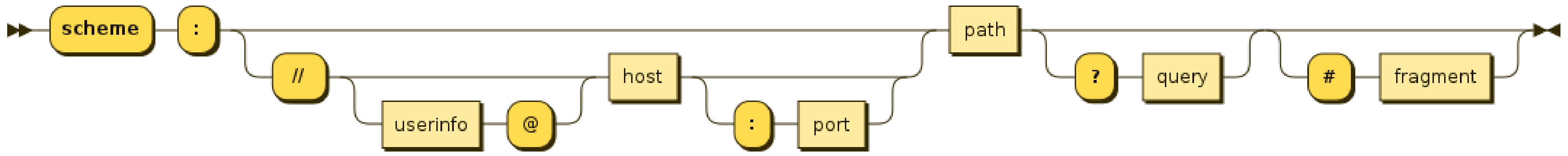




Uniform Resource Identifier

string of characters that unambiguously identifies a particular resource:

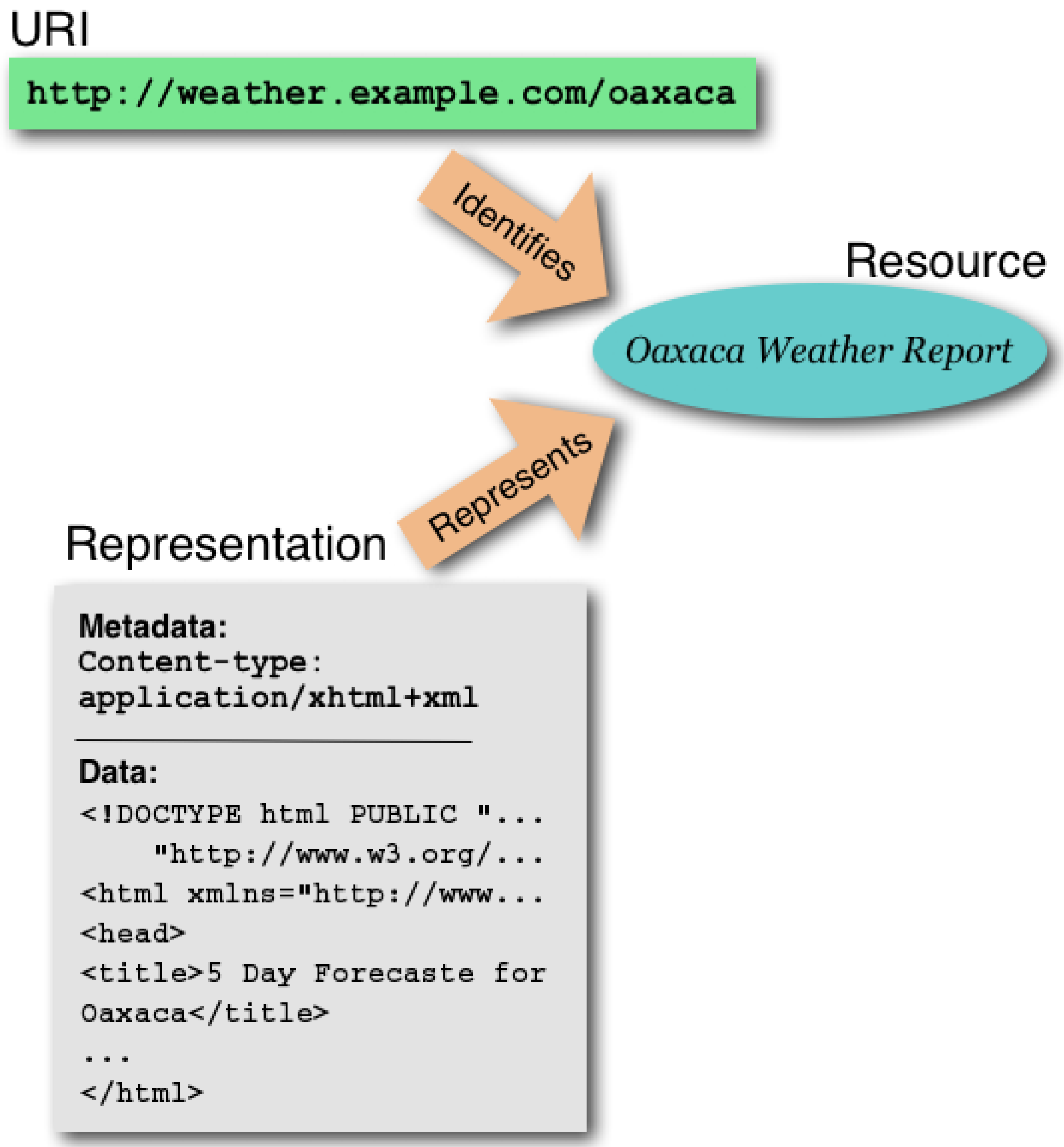
```
    scheme:[//authority]path[?query][#fragment]
```



Architecture of the World Wide Web, Volume One

W3C Recommendation 15 December 2004

From https://www.w3.org/TR/webarch/



Metadata:
Content-type:
application/xhtml+xml

Data:

```
<!DOCTYPE html PUBLIC "...
    "http://www.w3.org/...
<html xmlns="http://www...
<head>
<title>5 Day Forecaste for
Oaxaca</title>
...
</html>
```

REST

Representational state transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation.

E.g. users

```
GET    /vinyl/users/<username>          => getUser
GET    /vinyl/users                     => getUsers
DELETE /vinyl/users/<username>          => removeUser
```

The same path but different verbs (HTTP methods)

E.g. users

```
GET    /vinyl/users/<username>          => getUser
GET    /vinyl/users                     => getUsers
DELETE /vinyl/users/<username>          => removeUser
PUT    /vinyl/users/<username>          => addUser
```

addUser is idempotent, because username is unique. Thus PUT

E.g. users

```
GET    /vinyl/users/<username>           => getUser
GET    /vinyl/users                      => getUsers
DELETE /vinyl/users/<username>           => removeUser
PUT    /vinyl/users/<username>           => addUser
POST   /vinyl/users/<username>/artists   => addArtist
```

addArtist allows repetitions, thus it is not idempotent. So, POST is right choice!

Put <versus> Post

PUT for requests that are *idempotent*

Idempotent property is defined by RFC 7231 as:

> *A request method is considered "idempotent" if the intended* ***effect on the server*** *of multiple identical requests with that method* ***is the same as the effect for a single*** *such request.*

HTTP Status codes

- *1xx informational response* – the request was received, continuing process
- *2xx successful* – the request was successfully received, understood, and accepted
- *3xx redirection* – further action needs to be taken in order to complete the request
- *4xx client error* – the request contains bad syntax or cannot be fulfilled
- *5xx server error* – the server failed to fulfil an apparently valid request