

# ANATOMIA DI UN'APPLICAZIONE JAVASCRIPT ISOMORFICA

Luigi Maselli

web: <https://grigio.org>

web: <https://corso-javascript.it>

twitter: @grigio

# ARGOMENTI DEL GIORNO

- Cosa è un'app/lib isomorfica
- JS Framework & libs
- Demo: Minimal Boilerplate

**COSA È UN'APP/LIB JS  
ISOMORFICA/UNIVERSALE ?**

# ISOMORFICA

*In matematica, in particolare in algebra astratta, un isomorfismo (dal greco ἴσος, isos, che significa uguale, e μορφή, morphé, che significa forma) è un'applicazione biunivoca fra oggetti matematici tale che l'applicazione e la sua inversa siano omomorfismi. Wikipedia*

# ISOMORFICA (PROGRAMMAZIONE)

*.. "Isomorphic JavaScript." A term coined by Charlie Robbins at Nodejitsu and popularized by Spike Brehm of Airbnb, this has come to refer to JavaScript code that can run equally well on the client or on the server, meaning that you can write your JS once and run it anywhere.*

<http://info.meteor.com/blog/isobuild-why-meteor-created-a-new-package-system>

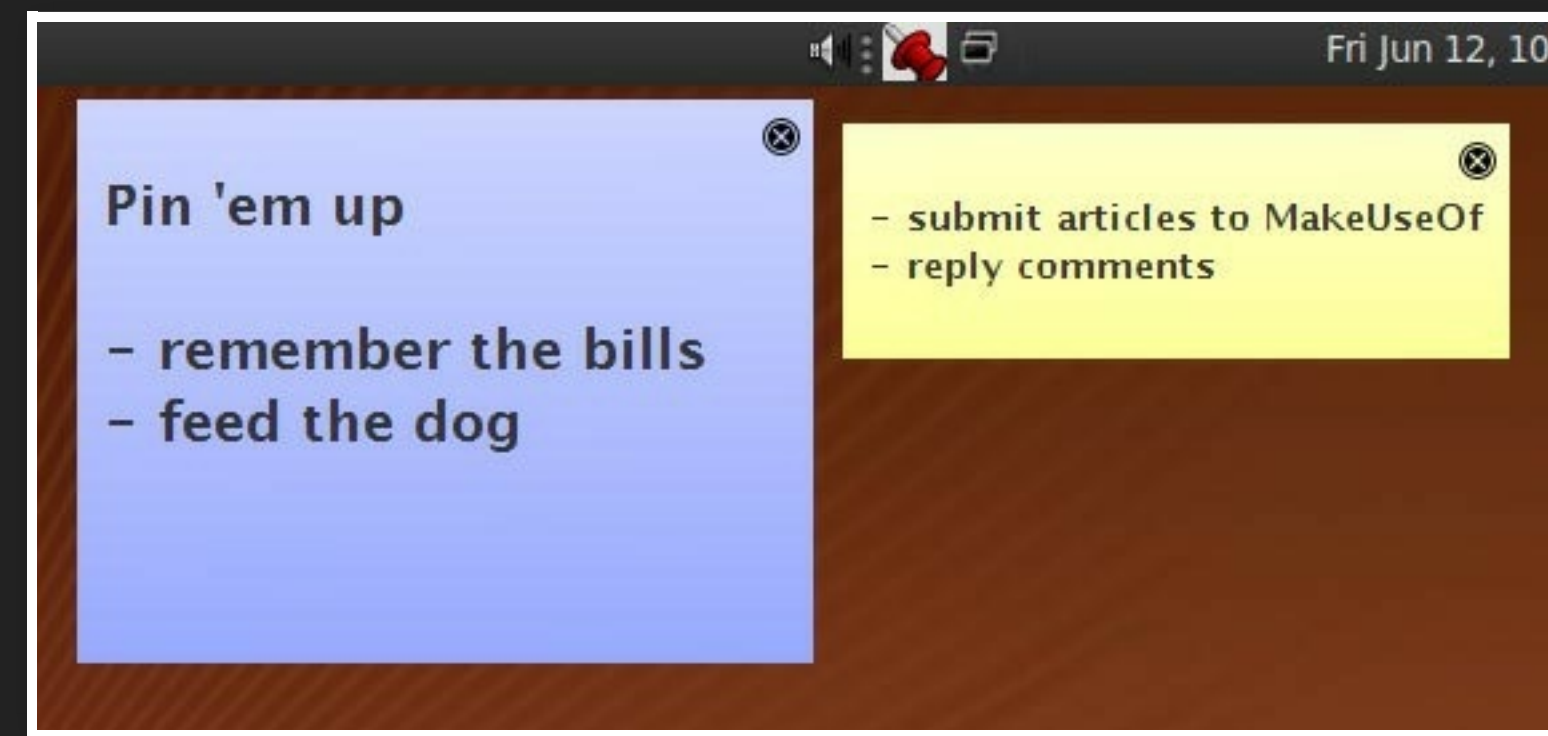
## (PARENTESI X GRAMMAR NAZI)

1. Si dice: isomorfa!
2. Dopo #petaloso e #educazione (al posto di istruzione..)
3. Se potete twittare a @AccademiaCrusca #isomorfica
4. Come isomorfa (ma più f\*\*a)

# ISOMORFICA?

Assomiglia a Java .. "Write once run everywhere"

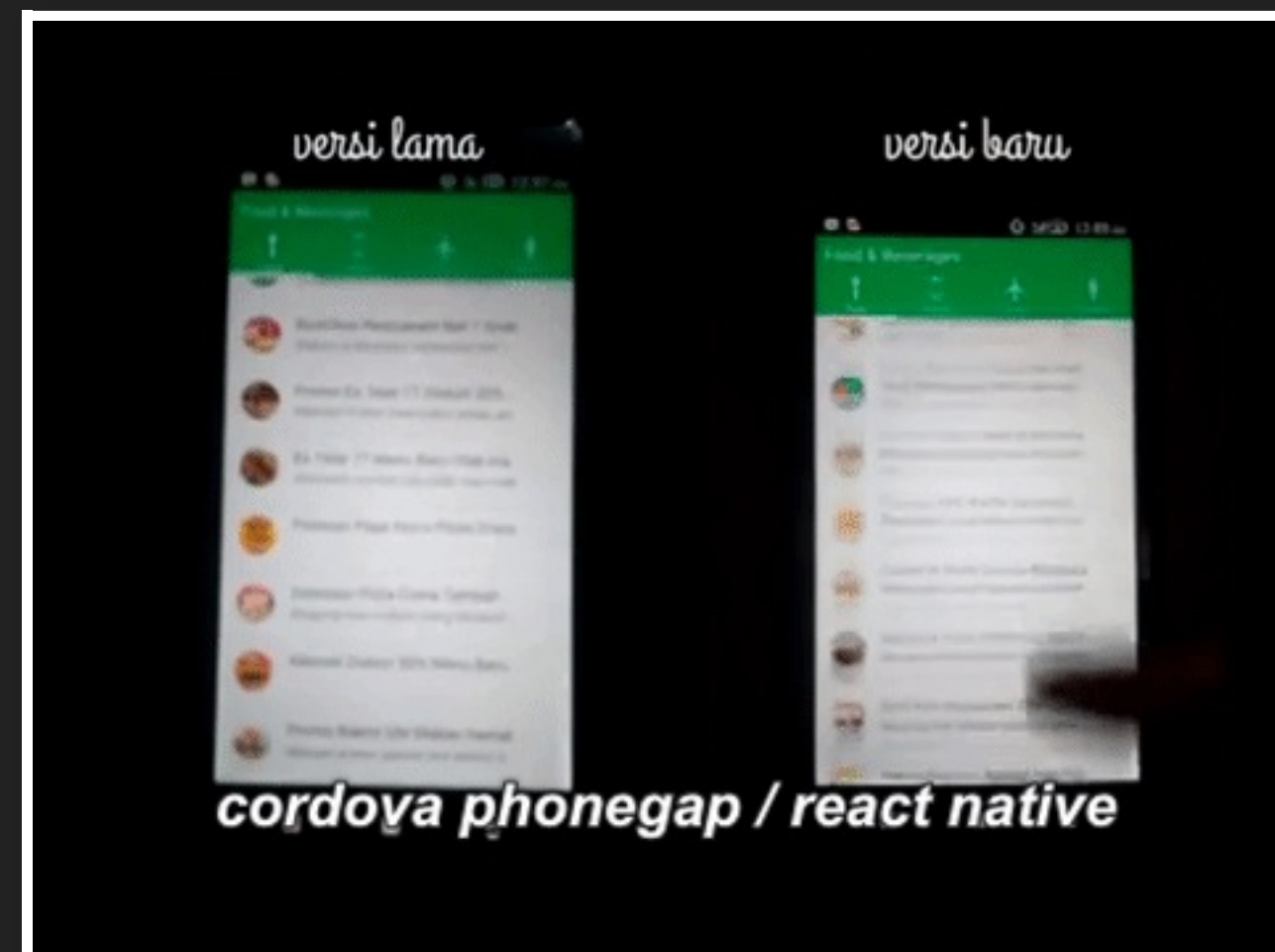




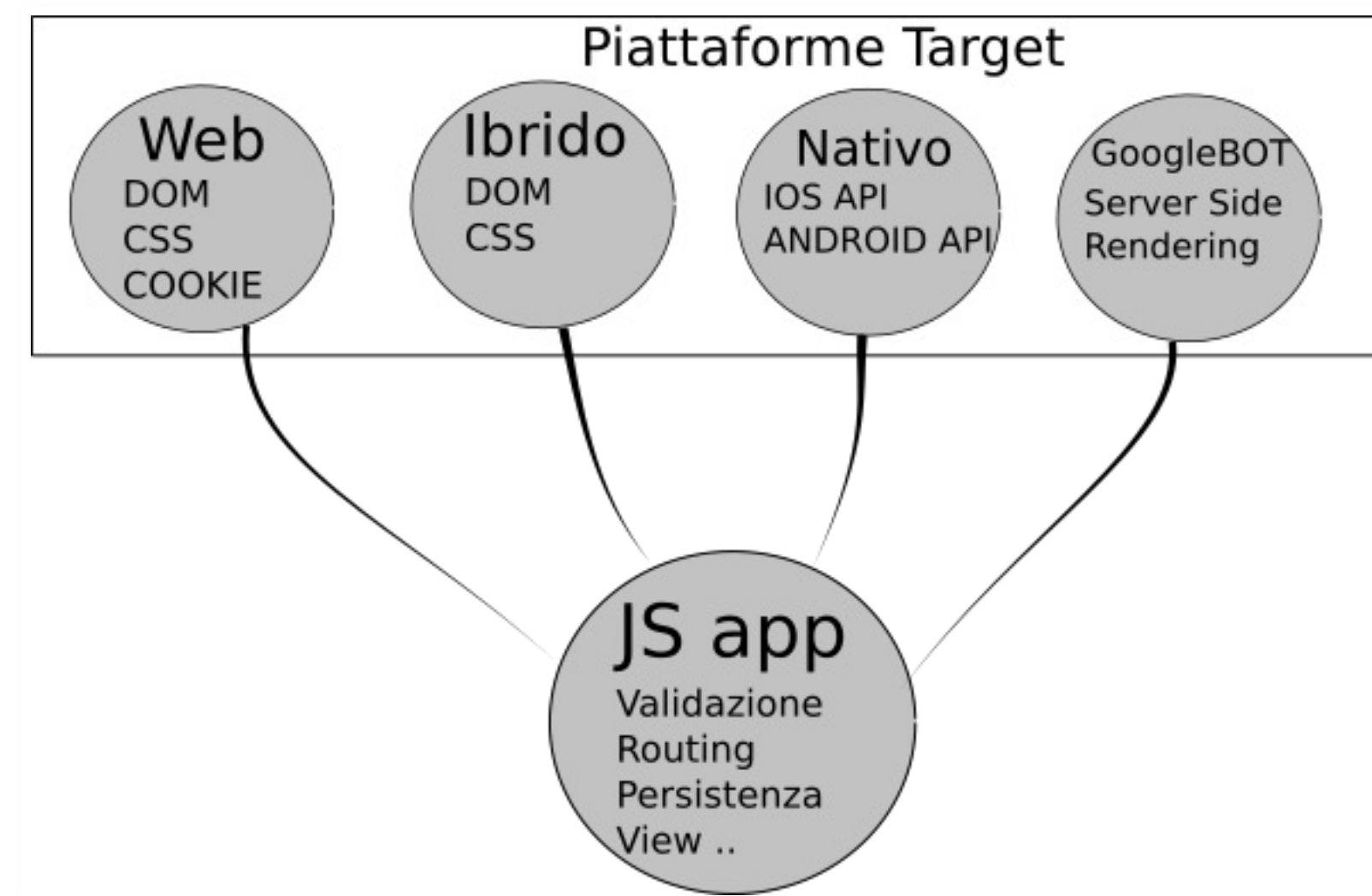
No, l'approccio "JavaScript isomorfo" è diverso!

1. "Learn once write everywhere": condividere quanto più codice e pattern possibili (ma anche poter sfruttare le specificità della piattaforma sottostante)
2. UI native: WebView oppure UI native

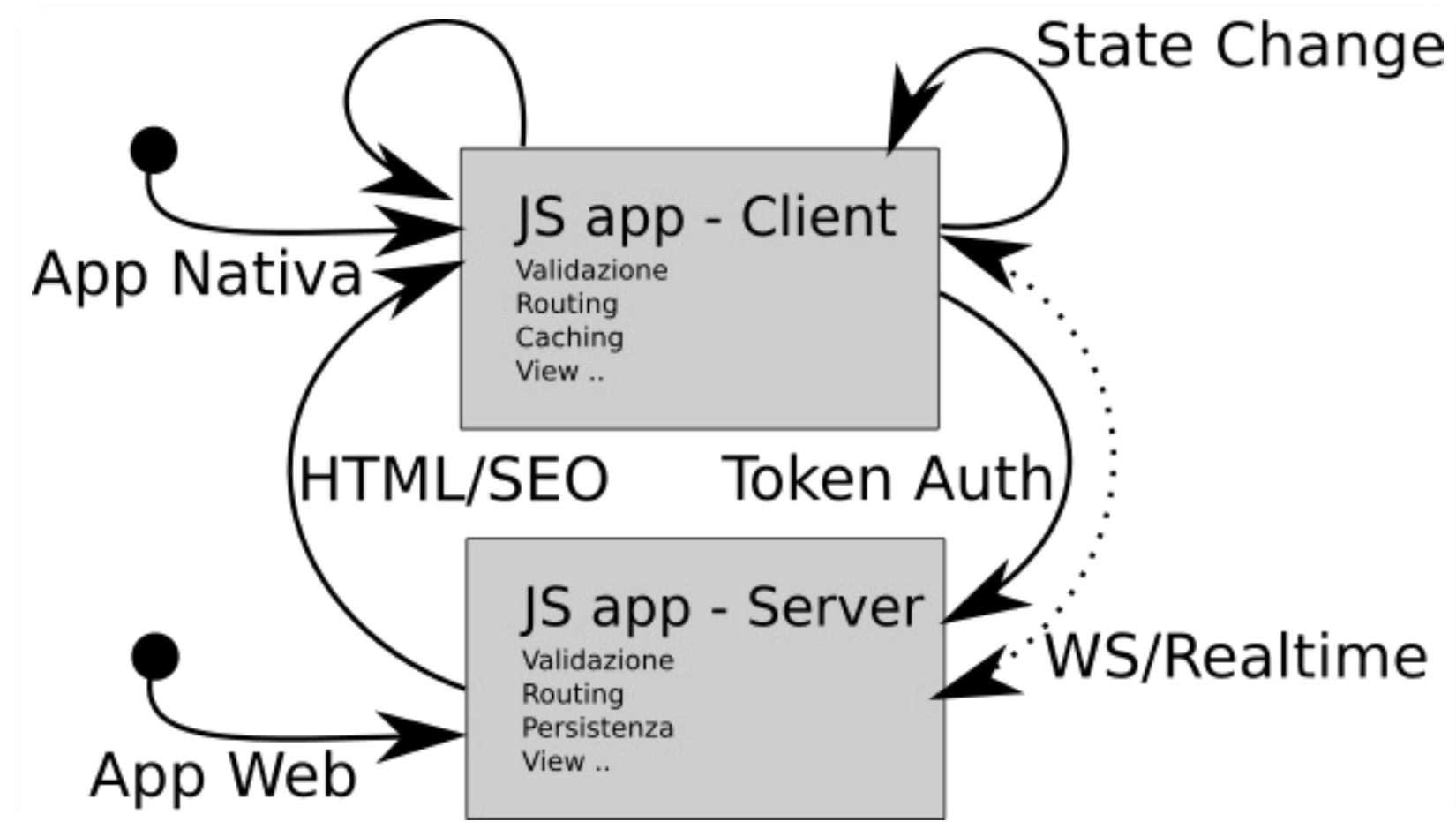
# PUÒ CAPITARE -\_(ツ)\_/



# JAVASCRIPT È OVUNQUE



# COSA POTREBBE FARE UN'APP JS



# ESEMPIO DI CODICE ISOMORFICO

```
// js agnostico (MyApp JS)

class Main extends React.Component {
  state = {counter:0};
  increment() {
    const {counter} = this.state;
    this.setState({counter: counter + 1 });
  }
  render() {
    return (
      <div>
        Main component, {this.state.counter} <button onClick=
      </div>
    );
  }
}
```

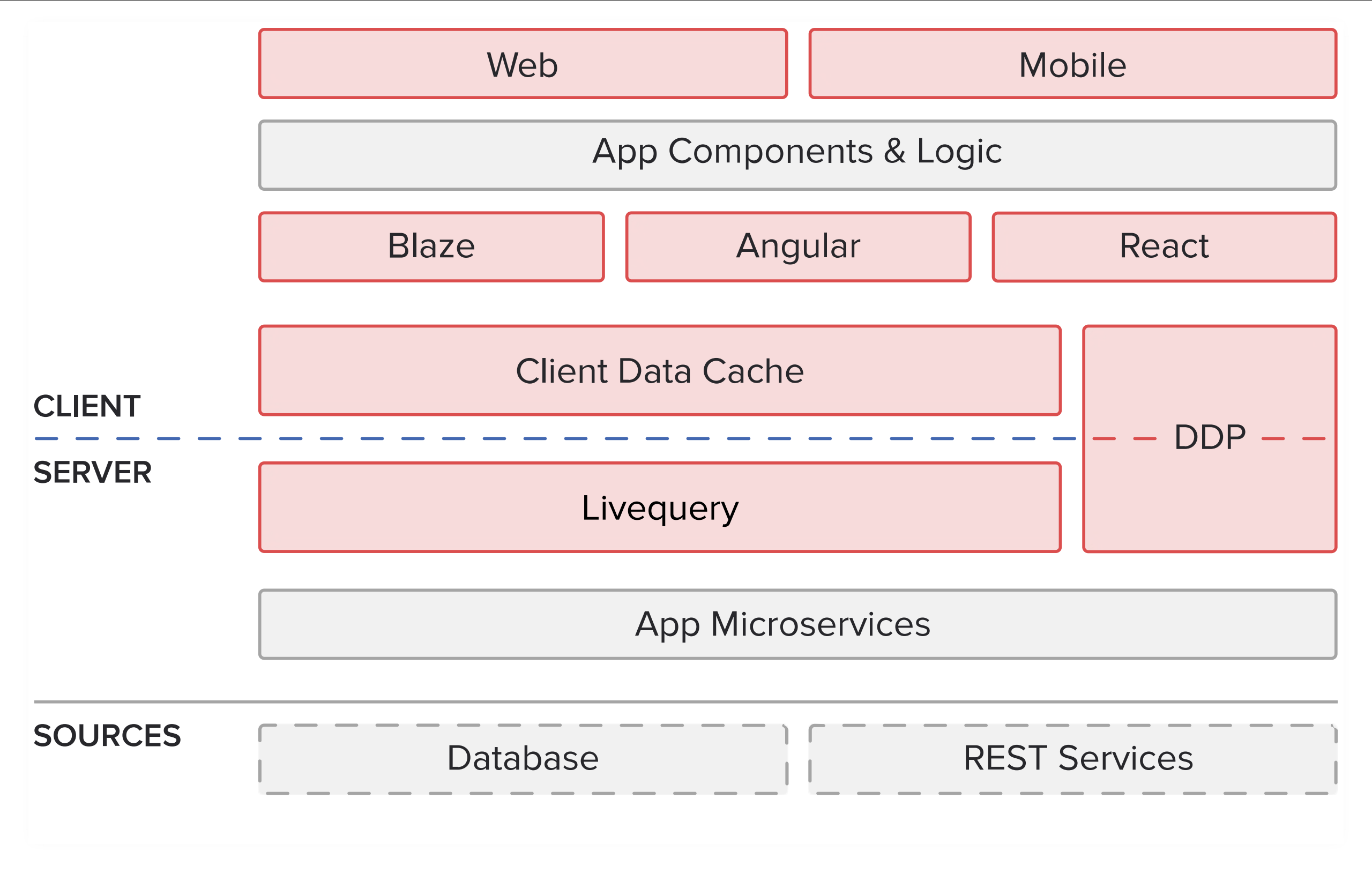




Bello JavaScript isomorfo da dove si parte?

## OPZIONE 1: METEOR

La piattaforma js isomorfica più conosciuta, comprende un set di librerie che gestiscono diversi aspetti isomorfici e client / server.



## Meteor :)

- It just works today (2012 \$11.2M)
- Semplice (se si rimane sul tracciato)
- Documentato
- Realtime build-in
- Autenticazione

Meteor :(

- Embed dipendenze obsolete
- Molte cose cambieranno. Inevitabilmente verso NodeJS/NPM (1.3beta16)
- Meteor è tanto ma non è tutto
- Praticamente solo MongoDB

# OPZIONE 2: DIY IL TUO JAVASCRIPT ISOMORFICO

Meatier, Mantra, Apollo (by Meteor + GraphQL) .. ... oppure  
componiti il tuo Meteor con quello che ti serve: react,  
webpack, babel, ..

# REACT STARTER PROJECT

Find a React Starter Project...

with:

without:

Separate search terms with commas.  
Examples: Try `redux` and `HMR`, no `gulp` or `minimal react native` or `vanilla JavaScript`.

68 matches Most stars

**kriasoft/react-starter-kit** Babel 6 ES6 HMR linter universal

React Starter Kit — isomorphic web app boilerplate (Node.js/Express, GraphQL, React.js, Babel 6, PostCSS, Webpack, BrowserSync)

7121  
79

**erikras/react-redux-universal-hot-example** Babel 6 ES6 HMR inline style linter live reload react-router Redux tests universal

A starter boilerplate for a universal webapp using express, react, redux, webpack, and react-transform

4405  
95

**mxstbr/react-boilerplate** Babel 6 ES6 HMR linter react-router Redux tests

:fire: Quick setup for performance orientated, offline-first React.js applications featuring Redux, hot-reloading, PostCSS, react-router, ServiceWorker, AppCache, FontFaceObserver and Mocha.

3243  
46

**davezuko/react-redux-starter-kit** Babel 6 ES6 HMR linter react-router Redux tests

Get started with React, Redux, and React-Router!

2862  
83

**coryhouse/react-slingshot** Babel 6 ES6 HMR linter react-router Redux tests

React + Redux starter kit with Babel, hot reloading, testing, linting and a working example app, all built in

2663  
41

**este/este** Babel 6 ES6 HMR linter react native react-router Redux universal

Dev stack and starter kit for functional and universal (browser, server, mobile) React apps

2652  
101

# MEATIER

Problem	Meteor's solution	My solution	Result
Database	<a href="#">MongoDB</a>	<a href="#">RethinkDB</a>	Built in reactivity, but you can use any DB you like
Database schema	<a href="#">Simple Schema</a>	<a href="#">GraphQL</a>	Can't have a hipster webapp without GraphQL!
Client validation	<a href="#">Simple Schema</a>	<a href="#">Joi</a>	Clean API for client validation, although the package is HUGE
Database hooks	<a href="#">Collections2</a>	<a href="#">GraphQL</a>	GraphQL is overkill for small apps (then again, so is meatier)
Forms	<a href="#">AutoForm</a>	<a href="#">redux-form</a>	state tracking awesomeness that works beautifully with react
Client-side cache	<a href="#">Minimongo</a>	<a href="#">redux</a>	Bonus logging, time traveling, and undo functionality
Socket server	<a href="#">DDP-server</a>	<a href="#">socketcluster</a>	super easy scaling, pubsub, auth, middleware
Authentication	Meteor accounts	<a href="#">JWTs</a>	JWTs can also serve to authorize actions, too
Auth-transport	<a href="#">DDP</a>	GraphQL (via HTTP)	Don't use sockets until you need to



Problem	Meteor's solution	My solution	Result
Database	<a href="#">MongoDB</a>	<a href="#">RethinkDB</a>	Built in reactivity, but you can use any DB you like
Database schema	<a href="#">Simple Schema</a>	<a href="#">GraphQL</a>	Can't have a hipster webapp without GraphQL!
Client validation	<a href="#">Simple Schema</a>	<a href="#">Joi</a>	Clean API for client validation, although the package is HUGE
Database hooks	<a href="#">Collections2</a>	<a href="#">GraphQL</a>	GraphQL is overkill for small apps (then again, so is meatier)
Forms	<a href="#">AutoForm</a>	<a href="#">redux-form</a>	state tracking awesomeness that works beautifully with react
Client-side cache	<a href="#">Minimongo</a>	<a href="#">redux</a>	Bonus logging, time traveling, and undo functionality
Socket server	<a href="#">DDP-server</a>	<a href="#">socketcluster</a>	super easy scaling, pubsub, auth, middleware
Authentication	Meteor accounts	<a href="#">JWTs</a>	JWTs can also serve to authorize actions, too
Auth-transport	<a href="#">DDP</a>	GraphQL (via HTTP)	Don't use sockets until you need to

## DIY PRO

- includi solo quello che ti serve
- mainstream NPM packages
- incrementale

## DIY CONTROLLO

- è tua responsabilità incollare tanti piccoli pezzettini

# DEMO

<https://github.com/corso-javascript/react-isomorphic-minimal>

<https://github.com/corso-javascript/docker-compose-nginx-nodejs>

QUINDI..







# CONCLUSIONE

- Preparati a nuovo mondo JavaScript centrato
- Non c'è un pattern che va bene per tutti, e sempre un compromesso tra il controllo che vuoi avere sull'app e quello che deleghi al framework lib.
- Non ottimizzare i non problemi



toddlertale

# GRAZIE!

Codice sconto **TOJS119**

[corso-javascript.it/?coupon\\_code=TOJS119](https://corso-javascript.it/?coupon_code=TOJS119)