

# WHY METEORJS

# INTRODUCTION TO METERJS + DEMO APP

# LUIGI MASELLI

FullStack Developer freelance

+ hackerpreneur  
remote worker  
blogger

*http: [grigio.org](http://grigio.org)  
twitter: [@grigio](https://twitter.com/grigio)*

We use Meteor

Luigi Maselli

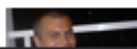
Sign Out



282 Meteor users are on the map!

Zoom and click to find your geohash area or click on an existing one.

Latest users

 Stefan Smiljkovic ( [shfefcs](#) )

# 10 PROGRAMMO

PER ESPERTI

java 8 SE update 5

SUL CD NUOVA VERSIONE SENZA BATA

MOBILE & DESKTOP "CUCINA" LA TUA APP PER WINDOWS E WINDOWS PHONE

GRATIS PER LO MICROSOFT APP STUDIO

SITI PERFETTI SU DESKTOP E TELEFONINI

Con Foundation 5 e Wordpress le tue pagine si mostrano sempre al meglio.

WEB 2.0 SONDAGGI IN TEMPO REALE CON METEOR JS

Realizziamo una Web App con l'unica piattaforma per sviluppare in Javascript sia su client che su server

to e scopri come

## #trenzi è un Social Media Monitor sul Governo di Matteo Renzi (@matteorenzi) con Smart Buzz Sentiment Analysis™.

### TOP 10 HASHTAG

- ♥ #Renzi
- ♥ #IceBucketChallenge
- ▲ #renzi
- ♥ #M5S
- ▲ #RENZI
- ▲ #PIL
- ♥ #tasse
- ♥ #GDF
- ♥ #Inonda
- ♥ #PD

### ULTIMI 10 HASHTAG

- #donatagata (0)
- #renzi (397)
- #Renzi (155)

### COS'È #TRENZI?

#trenzi è un progetto #openfuffa , per sensibilizzare giornalisti e politici sull'importanza del "veri" #opendata.

Tweet: 52 Mi piace: 42

#### NUMERO DI TWEET

# 597

nell'ultima ora @ 0.22 tweet/s

#### VOLUME DI TWEET

22:00

MANCANO AL 2018

1228

giorni a partire da oggi.

Scopri come utilizzare i servizi web di Amazon per gestire dati di qualsiasi dimensione

TECNOLOGIA DELL'INFORMATICA

**CONTEXT**

# OL' WEB

client → server  
client ← (html) server  
(x 100 req)

- ✓ Simple
- ✓ Google Search Indexable
- ☹ Bandwidth
- ☹ Interaction
- Email notification

# WEB2.0 BUZZ

Evolution..

# APP MANIA

app → server

app ← (json) server

(app store download x 100 req)

☹ Simple

☹ Google Search Indexable

✓ Bandwidth

✓ Interaction

Push notification + Email

RAGE

I want everything and simple!

# FUTURE (MAYBE)

frontend app ↔ backend

- ✓ Simple
  - ✓ Google Search Indexable
  - ✓ Bandwidth
  - ✓ Interaction
  - ✓ Offline limited usage
- Push notification + Email

# CONTEMPORARY TECH COMPONENTS

NodeJS

noSQL, cloud, PaaS

client side routing, virtual dom

data binding, cordova, localstorage

minification, requirejs, grunt

gulp, react.js, es6

MongoDB

But how to wire them?



ipt  
work



**ANOTHER BOILERPLATE? NAH...**

nie in javascript-lib

reactjs, single-page application

# React.js and How Does It Fit In With Everything Else?

---

## Introduction

Random quote time

*"Simplicity is the ultimate sophistication"*  
*Leonardo Da Vinci*

# JACKPOT: BROWSER + SERVER + DB



# METEORJS INGREDIENTS

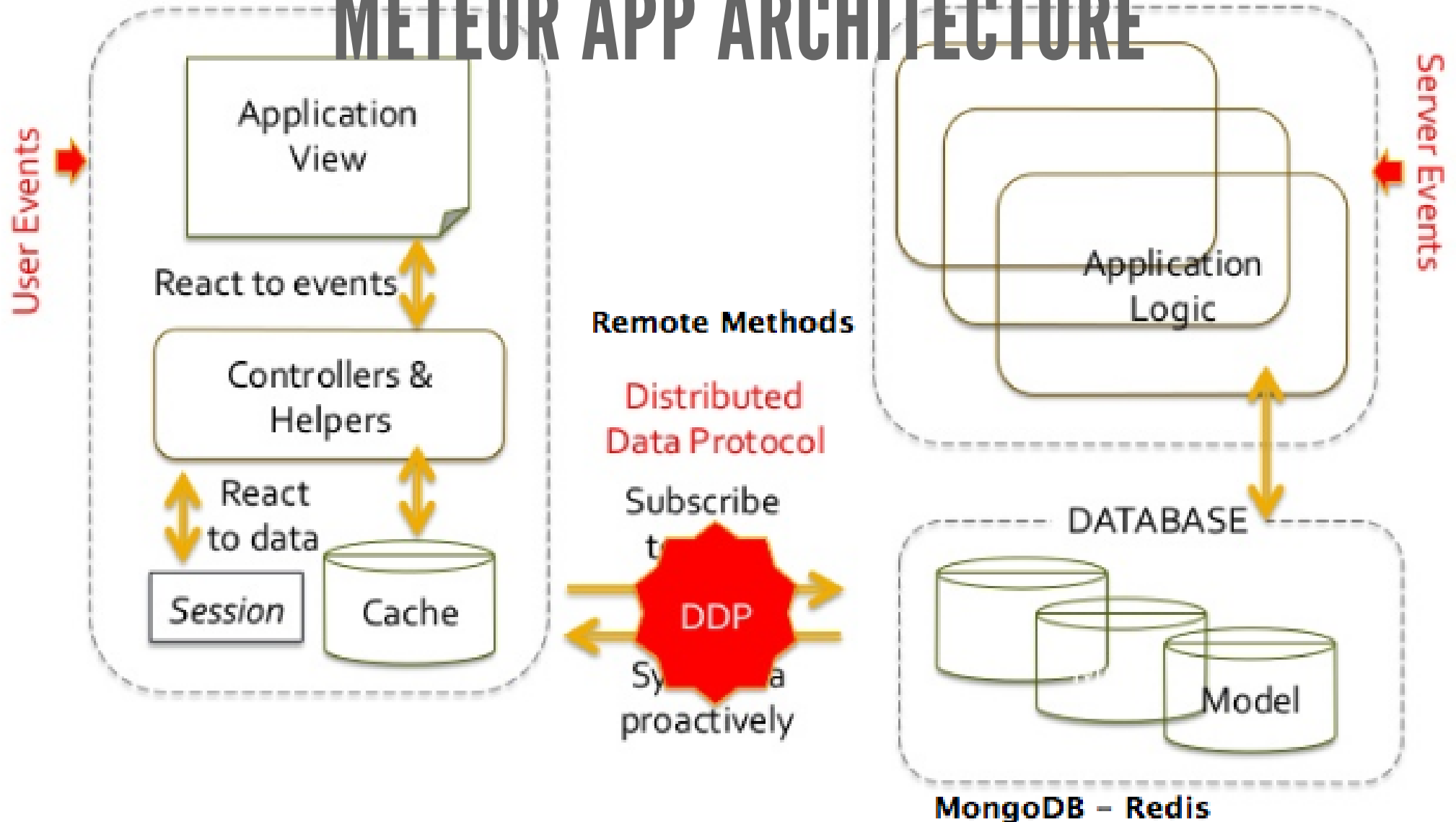
- Isomorphic javascript (based on NodeJS)
- JS on the client, on the server, on db (MiniMongo, MongoDB)
- Shared code: data validations, routing, functions
- Views: handlebars like (json + HTML holes)
- Start simple refine later (Convention over configuration)
- Packages “plug and play”. Integration with NPM or Bower
- Builtin: Virtual DOM, minification,..
- Pluggable: less, coffeescript, ES6 Harmony, classic & social authentication support,..
- Meteor can serve: static files, DDP messages(ws,HTTP), HTTP Requests (the trinity..)

Web - Cordova Android - Cordova IOS

METEOR CLIENT

SMART SERVER

# METEOR APP ARCHITECTURE



# METEOR KEY CONCEPTS

- Reactivity (redraw on data changes and not events)
- Virtual DOM
- Database everywhere (but not the same data)
- Latency compensation
- Remote Methods
- Platform (Web, Phonegap / Cordova Android, iOS)

# THE MINIMAL CHAT

```
<!-- client/chat.html -->
<head>
  <title>Meteor Minimal Chat</title>
</head>

<body>
  {{> chatList}}
  {{> chatEntry}}
</body>

<template name="chatList">
  <h1>Messages</h1>
  <ul>
    {{#each messages}}
      <li><strong>{{user}}</strong> {{text}}</li>
    {{/each}}
  </ul>
</template>

<template name="chatEntry">
  <input id="user"/> <input id="text"/>
  <button id="send">send</button>
</template>
```

```
// collections.js
Messages = new Meteor.Collection('messages');

// client/chat.js
Template.chatList.helpers({
  'messages': function () {
    return Messages.find({});
  }
});

Template.chatEntry.events({
  'click #send' : function () {
    Messages.insert(
      { user:$('#user').val(), text:$('#text').val() }
    );
    $('#text').val('');
  }
});
```

# HACKERCHAT

(a more real world Meteor example)

[github.com/grigio/meteor-hackerchat](https://github.com/grigio/meteor-hackerchat)

# RESOURCES

<http://meteor.com> (official site and docs)  
<https://www.discovermeteor.com> (book, free in non-english)  
<http://atmospherejs.com> (meteor packages)  
<http://www.eventedmind.com> (screencasts)  
<http://meteorhacks.com> (blog)  
<http://weuse.meteor.com> (MeteorJS users on a map)  
<http://madewith.meteor.com> (apps open or not showcase)  
<http://telesc.pe> / microscope (Real world open source HN/Reddit)

# FINE

Meteor Italia Meeting  
Torino, 7/11/2014  
<http://meteor-italia.github.io>