



JavaScript App Platform

# What We'll Cover

---

- What is Meteor?
- Why Meteor?
- Demo

# What is Meteor?

---

- Full-stack JavaScript open source platform
- Web and mobile apps
- 100% JavaScript, frontend and backend

# Meteor Tools

---

- CLI Tool – meteor – create, run, deploy
- Full stack package systems – atmosphere and npm
- Build system – browser and mobile (Cordova)

# Meteor components - Collections

---

- Server and client-side Mongo Collections – Minimongo on the client (oplog based LiveQuery system)

```
1 Todos = new Mongo.Collection('Todos');
```

- JSON schema

```
1 Lists.schema = new SimpleSchema({  
2   name: {type: String},  
3   incompleteCount: {type: Number, defaultValue: 0},  
4   userId: {type: String, regex: SimpleSchema.RegEx.Id, optional: true}  
5 });
```

# Meteor components – Pub/Sub

---

- Publications (server APIs)

```
1 Meteor.publish('todos.inList', function(listId) {  
2   // we need to check the `listId` is the type we expect  
3   new SimpleSchema({  
4     listId: {type: String}  
5   }).validate({ listId });  
6  
7   // ...  
8 });
```

- Subscriptions (data connections) using DDP or REST  
– bridge between server and client collections

```
1 Meteor.subscribe('todos.inList', list._id);
```

# Meteor components – Server Methods

---

- Server methods (server APIs / RPC) - synchronous, non-blocking, sequential

```
1 Meteor.methods({
2   'todos.updateText'({ todoId, newText }) {
3     new SimpleSchema({
4       todoId: { type: String },
5       newText: { type: String }
6     }).validate({ todoId, newText });
7
8     const todo = Todos.findOne(todoId);
9
10    if (!todo.editableBy(this.userId)) {
11      throw new Meteor.Error('todos.updateText.unauthorized',
12        'Cannot edit todos in a private list that is not yours');
13    }
14
15    Todos.update(todoId, {
16      $set: { text: newText }
17    });
18  }
19 });
```

# Meteor components - Accounts

- Users and accounts (Oauth, email, roles)

```
1  {
2    _id: "bbca5d6a-2156-41c4-89da-0329e8c99a4f", // Meteor.userId()
3    username: "cool_kid_13", // unique name
4    emails: [
5      // each email address can only belong to one user.
6      { address: "cool@example.com", verified: true },
7      { address: "another@different.com", verified: false }
8    ],
9    createdAt: Wed Aug 21 2013 15:16:52 GMT-0700 (PDT),
10   profile: {
11     // The profile is writable by the user by default.
12     name: "Joe Schmoe"
13   },
14   services: {
15     facebook: {
16       id: "709050", // facebook id
17       accessToken: "AAACCgdX7G2...AbV9AZDZD"
```

## Accounts

```
#user
#userId
#users
#loggingIn
#logout
#logoutOtherClients
#loginWithPassword
#loginWith<ExternalService>
  Requesting Permissions
  Configuring Services
  Popup versus redirect flow
{{ currentUser }}
{{ loggingIn }}
#ui
```



# Meteor components - Views

---

- UI rendering libraries: Blaze (Handlebars)

```
1 <template name="Todos_item">
2   <div class="list-item {{checkedClass todo}} {{editingClass editing}}">
3     <label class="checkbox">
4       <input type="checkbox" checked={{todo.checked}} name="checked">
5       <span class="checkbox-custom"></span>
6     </label>
7
8     <input type="text" value="{{todo.text}}" placeholder="Task name">
9     <a class="js-delete-item delete-item" href="#">
10      <span class="icon-trash"></span>
11    </a>
12  </div>
13 </template>
```

```
1 {{#each todo in todos}}
2   {{> Todos_item (todoArgs todo)}}
3 {{else}}
4   <div class="wrapper-message">
5     <div class="title-message">No tasks here</div>
6     <div class="subtitle-message">Add new tasks using the field above</div>
7   </div>
8 {{/each}}
```

# Meteor components - Views

---

- React (JSX)

```
import React from 'react';

export default class HelloWorld extends React.Component {
  render() {
    return (
      <h1>Hello world</h1>
    );
  }
}
```

- Angular (HTML with special attribute syntax)

```
1 <div>
2   <ul>
3     <li *ngFor="let party of parties">
4       {{party.name}}
5       <p>{{party.description}}</p>
6       <p>{{party.location}}</p>
7     </li>
8   </ul>
9 </div>
```

# Meteor components - View

---

- UI/UX patterns: Event handling, Subscription readiness, Pagination, Optimistic UI

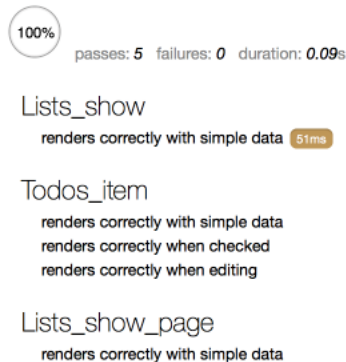
```
6   Template.Todos_item.events({
7     // update the text of the item on keypress but throttle the event to ensure
8     // we don't flood the server with updates (handles the event at most once
9     // every 300ms)
10    'keyup input[type=text]': _.throttle(function(event) {
11      updateText.call({
12        todoId: this.todo._id,
13        newText: event.target.value
14      }, (err) => {
15        err && alert(err.error);
16      });
17    }, 300)
18  });
```

- Reusable components: Iteration, Helpers, Reactive variables

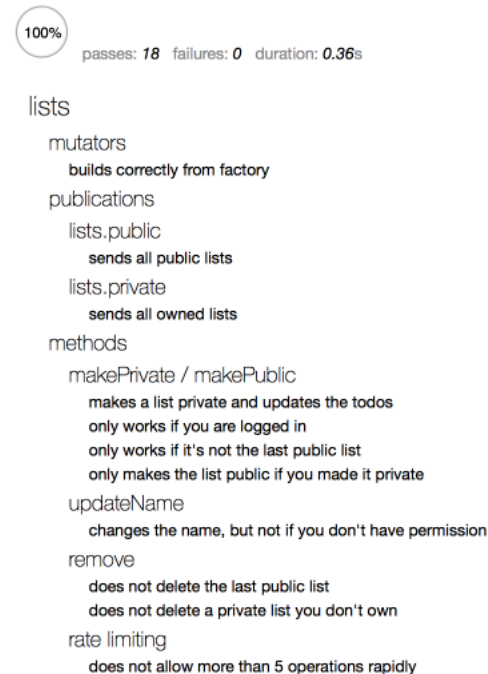
# Meteor components - Testing

- Testing – “meteor test”
  - Unit (mocha), Integration (mocha), Acceptance (chimp), in CI (mocha-phantomjs)

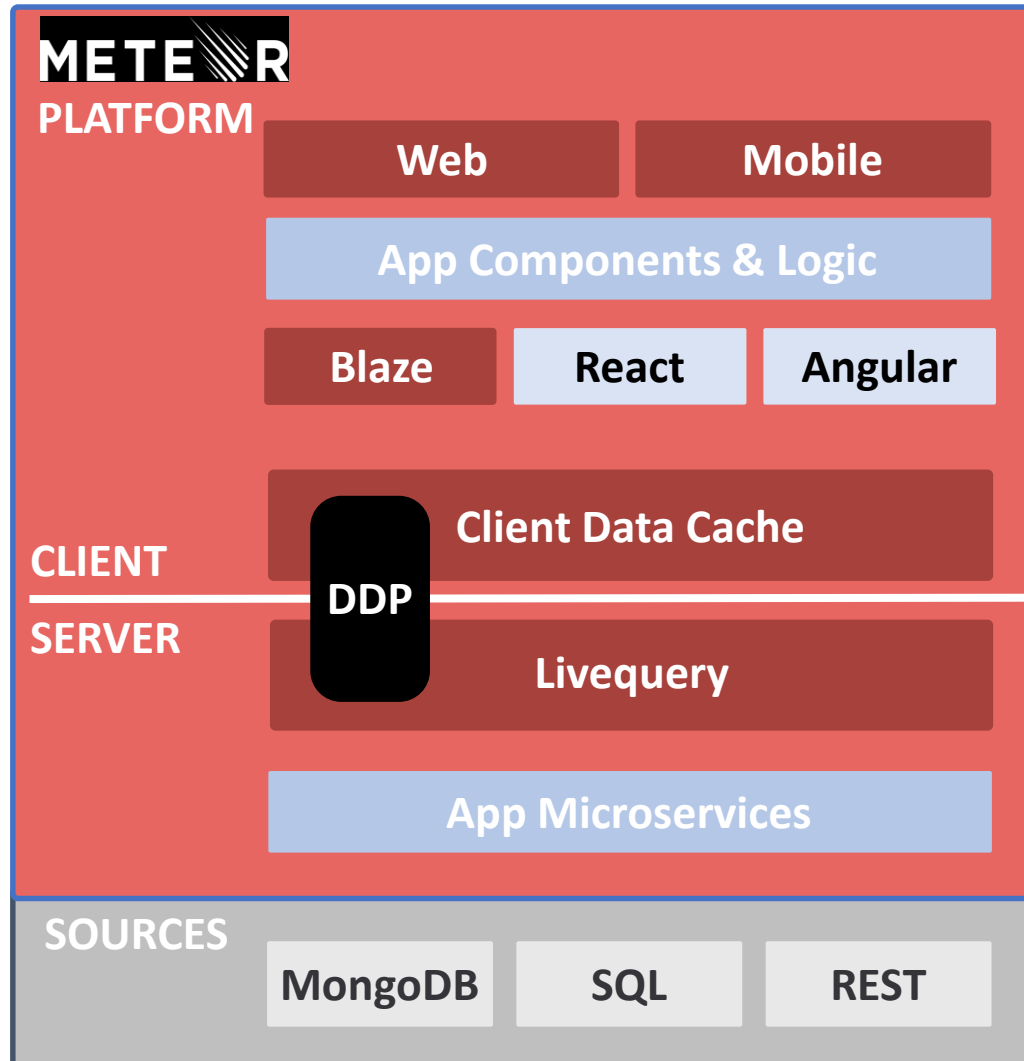
## Client tests



## Server tests



# Meteor Platform



# Meteor - Build

---

- Packages:
  - Atmosphere (native)
  - Npm
- Cordova
- Build System
  - Reloads app on file change
  - Compiles, combines and minifies code
  - Hot code push

# Meteor - Production

---

- Deployment and Monitoring
  - meteor build (noje.js app)
  - analytics
  - meteor-up tool

```
mkdir .deploy  
cd .deploy  
mup init <your project>
```

make changes to `mup.js` . and add a `settings.json` file.

```
mup setup  
mup deploy
```

# Meteor - Production

---

- Security:
  - Collection client access rules
  - Method validation
  - Rate limiting
  - `meteor deploy myapp.com --settings production.json`



# Meteor - Production

- meteor-up

```
meteor: {
  name: 'app',
  path: '../app',
  volumes: { //optional, lets you add docker volumes
    "/host/path": "/container/path", //passed as '-v /host/path:/container/path' to the docker
    "/second/host/path": "/second/container/path"
  },
  servers: {
    one: {}, two: {}, three: {} //list of servers to deploy, from the 'servers' list
  },
  buildOptions: {
    serverOnly: true,
    debug: true,
    mobileSettings: {
      yourMobileSetting: "setting value"
    }
  },
  env: {
    ROOT_URL: 'app.com',
    MONGO_URL: 'mongodb://localhost/meteor'
  },
  log: { //optional
    driver: 'syslog',
    opts: {
      "syslog-address": 'udp://syslogserverurl.com:1234'
    }
  }
  dockerImage: 'madushan1000/meteord-test', //optional
  deployCheckWaitTime: 60 //default 10
},
```

# Why Meteor?

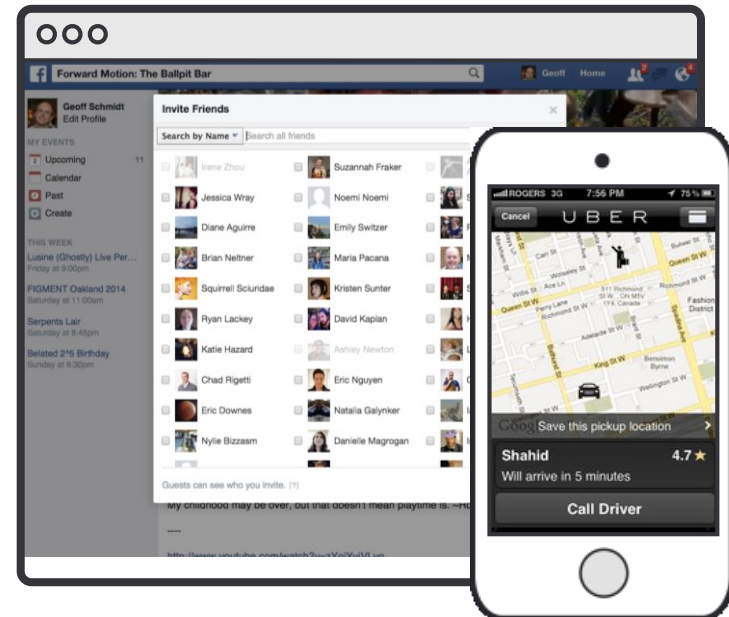
- Then

- Browser
- Refresh button
- Individual sessions
- Links and forms

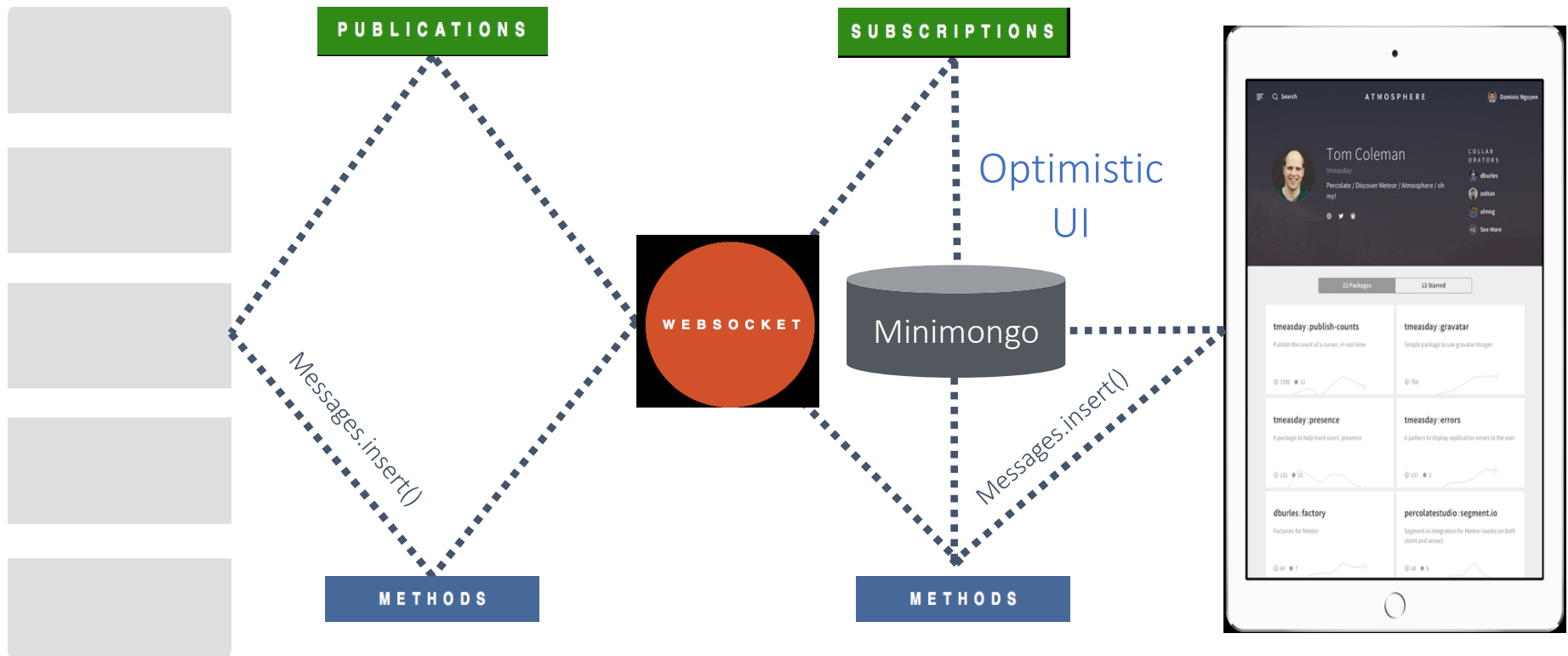


- Now

- Browser and Mobile
- Live updates
- Multiuser collaboration
- Native-style GUI



# Cool Stuff



# Cool Stuff

---

- Full-stack packages
  - Reactive rendering
  - Database sync
  - Intuitive, isomorphic API (client and server shared code)
  - Minimal boilerplate code
- 
- Fast install to deploy
  - Hot code push
  - Multi-platform support

# Learning Meteor

---

[meteor.com/tutorials](https://meteor.com/tutorials)

[guide.meteor.com](https://guide.meteor.com)

[themetorchef.com](https://themetorchef.com)

[github.com/Urigo/awesome-meteor](https://github.com/Urigo/awesome-meteor)

Thank You!

# Demo

---

- Medtech