Cognitive defusion application

Developing a single page application for android and iOS

Overview

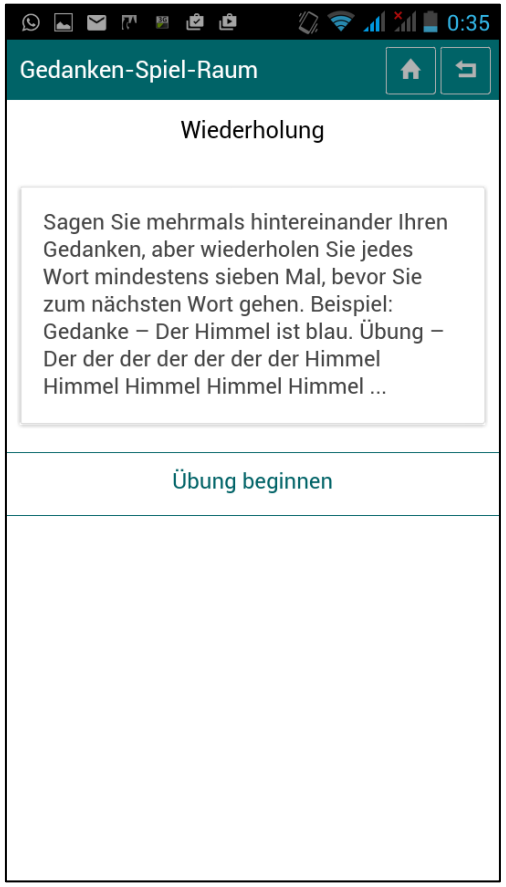
- Cognitive defusion in short
- Bird’s eye view
- Cordova
- Ionic/AngularJS
- Testing and calabash-android
- Live presentation

Cognitive defusion

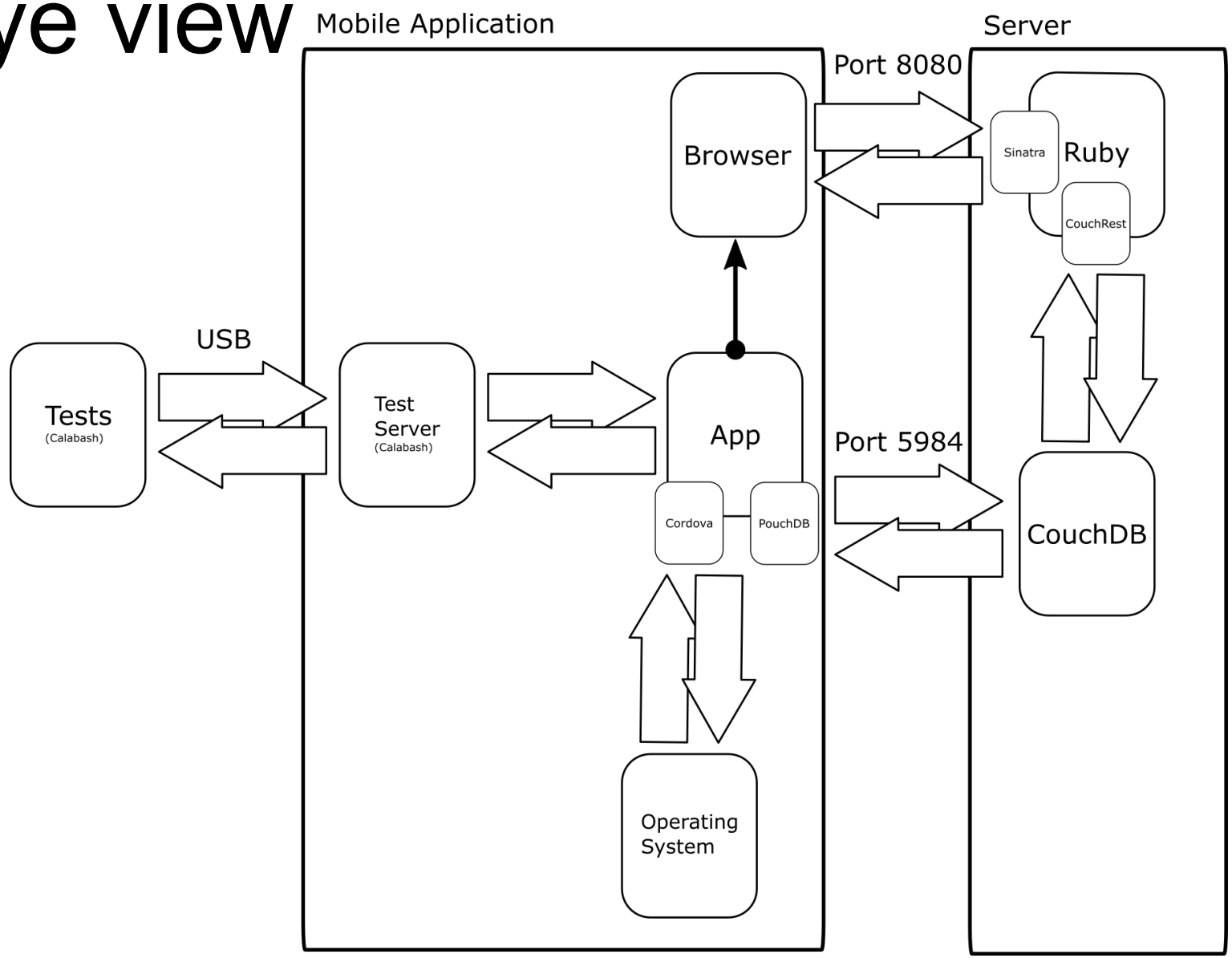
In very short

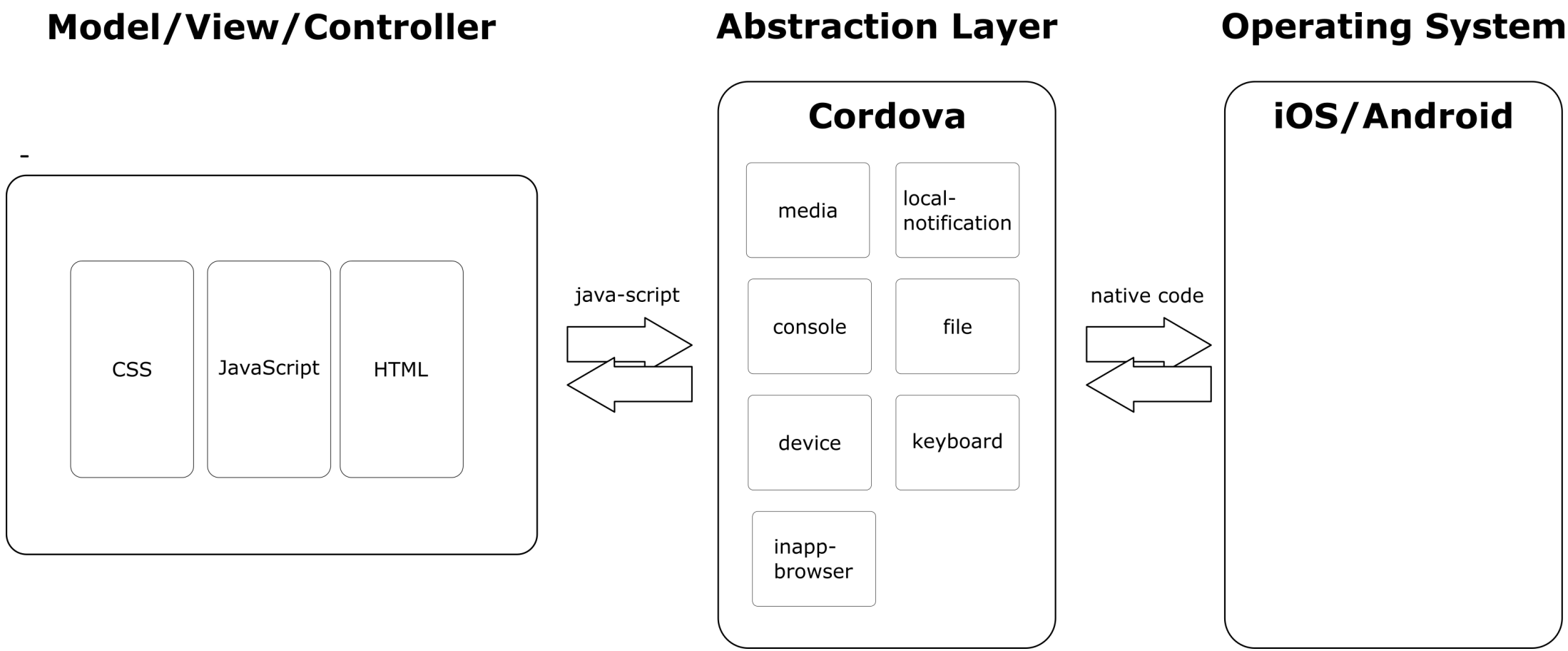
Decouple thoughts from reality by putting them into surreal situations.

«Imagine your mind is a parrot sitting on your shoulder and squaking your thought in your ear over and over again»



Bird’s eye view




Model/View/Controller
Abstraction Layer
Operating System
-
CSS
JavaScript
HTML
java-script
Cordova
media
local-
notification
console
file
device
keyboard
inapp-
browser
native code
iOS/Android

App/Cordova

- Ionic
- AngularJS

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">
6      <title>Kognitive Defusion</title>
7
8      <link href="lib/ionic/css/ionic.css" rel="stylesheet">
9      <!-- for customization purpose -->
10     <link href="css/style.css" rel="stylesheet">
11
12     <!-- pouchDB script for data storage and data synchronization -->
13     <script src="lib/pouchdb/pouchdb-3.3.0.min.js"></script>
14
15     <!-- uuid generator -->
16     <script src="lib/uuid/uuid.js"></script>
17
18     <!-- ionic/angularjs -->
19     <script src="lib/ionic/js/ionic.bundle.js"></script>
20
21     <!-- cordova script (this will be a 404 during development) -->
22     <script src="cordova.js"></script>
23
24     <!-- your app's js -->
25     <script src="js/classes.js"></script>
26     <script src="js/app.js"></script>
27     <script src="js/controllers.js"></script>
28     <script src="js/services.js"></script>
29     <script src="js/language.js"></script>
30
31   </head>
32   <body ng-app="starter" ng-controller="AppController">
33         <div id="show" c
34             <div clas                'show-text"></span></div>
35         </div>
36         <ion-nav-bar      gn-title="left" ng-hide="hide_header" name="header" class="bar bar-header bar-balanced"></ion-nav-bar>
37         <ion-nav-view></ion-nav-view>
38   </body>
39 </html>
```

External libraries

Model/View/Controller files (my code)

Filled dynamically

App/Cordova

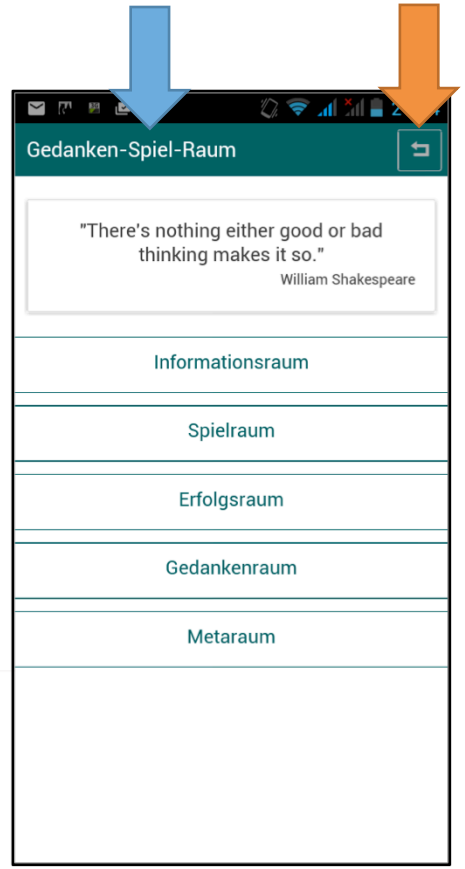
CSS

JavaScript

HTML

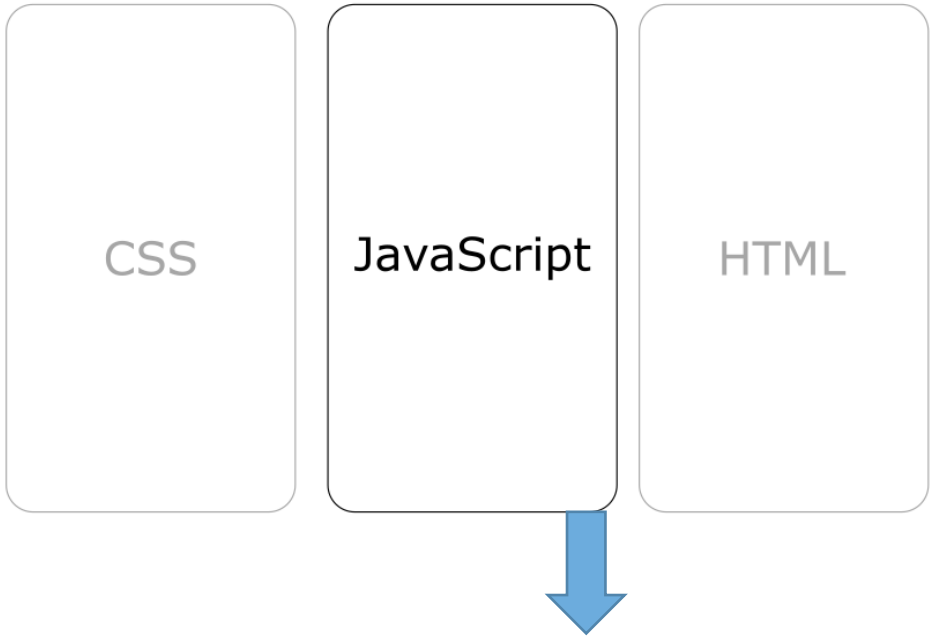
- Ionic/AngularJS

Look and Feel of the app




```
1 <ion-view name="overview" title="{{wording[lang].overview.name}}">
2     <ion-nav-buttons side="right">
3         <a class="button button-outline icon-left ion-arrow-return-left" ng-click="back()"></a>
4     </ion-nav-buttons>
5
6     <ion-content>
7         <div class="card">
8             <div class="text-center item item-text-wrap item-text-center">
9                 {{wording[lang].overview.quote.text}}<span class="author">{{wording[lang].overview.quote.author}}</span>
10            </div>
11        </div>
12
13        <a ui-sref="cognitive-defusion-info" class="button button button-full button-outline button-balanced">{{wording[lang].overview.info}}</a>
14        <a ui-sref="exercise-information" class="button button-full button-outline button-balanced">{{wording[lang].overview.games}}</a>
15        <a ui-sref="hitlist" class="button button-full button-outline button-balanced">{{wording[lang].overview.hitlist}}</a>
16        <a ui-sref="feedback" class="button button-full button-outline button-balanced">{{wording[lang].overview.feedback}}</a>
17        <a ui-sref="about" class="button button-full button-outline button-balanced">{{wording[lang].overview.about}}</a>
18    </ion-content>
19 </ion-view>
```

App/Cordova

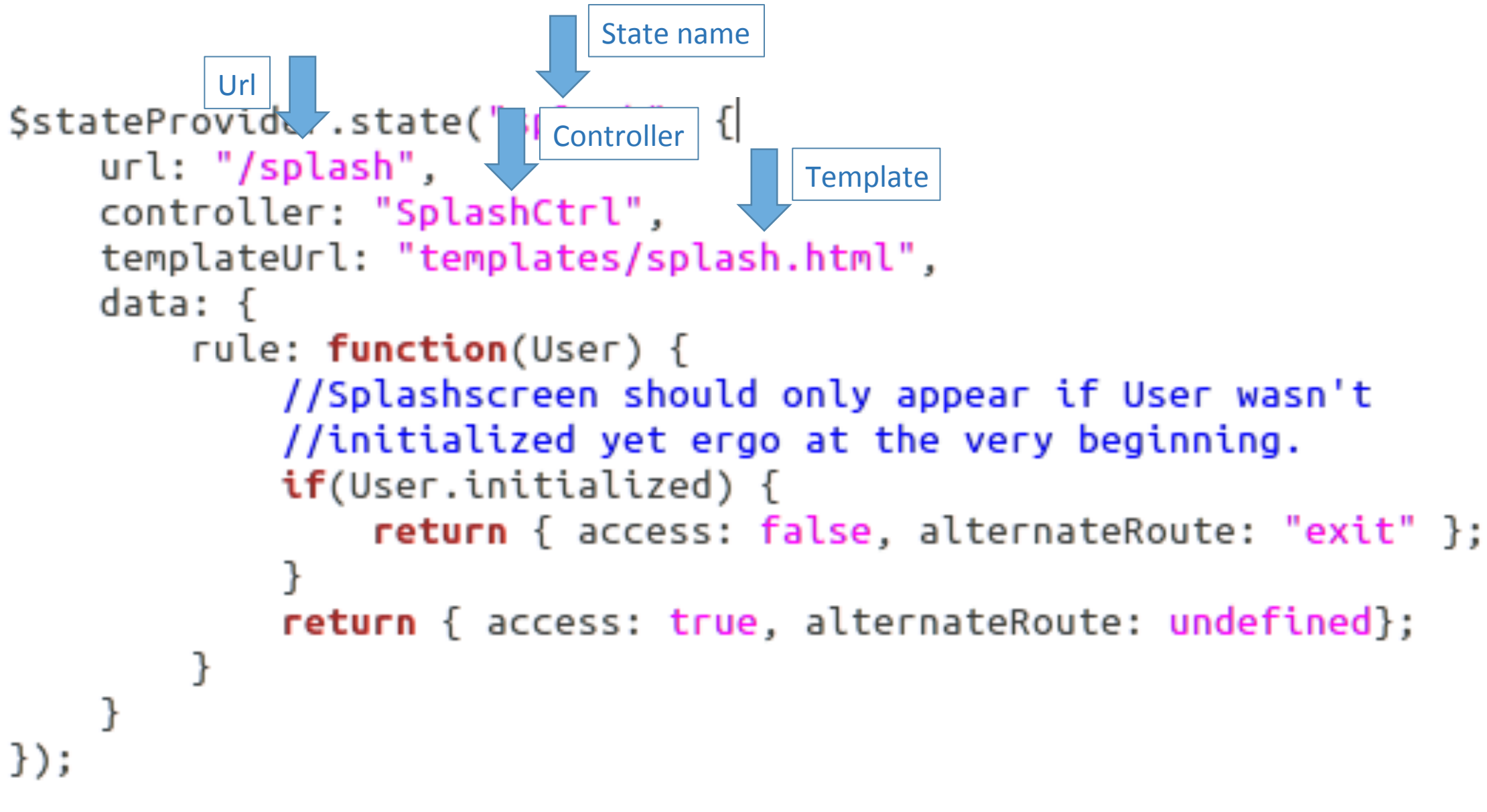


- Ionic/AngularJS

Look and Feel for the app
Model/View/Controller

```
<body ng-app="starter" ng-controller="AppController">
      <div id="show" class="hide">
          <div class="show-content"><span id="show-text"></span></div>
      </div>
      <ion-nav-bar align-title="left" ng-hide="hide_header" name="header"
      <ion-nav-view></ion-nav-view>
</body>
```

```
$stateProvider.state( lash", {
    url: "/splash",
    controller: "SplashCtrl",
    templateUrl: "templates/splash.html",
    data: {
        rule: function(User) {
            //Splashscreen should only appear if User wasn't
            //initialized yet ergo at the very beginning.
            if(User.initialized) {
                return { access: false, alternateRoute: "exit" };
            }
            return { access: true, alternateRoute: undefined};
        }
    }
});
```


State name
Url
Controller
Template
$stateProvider.state("splash", {
url: "/splash",
controller: "SplashCtrl",
templateUrl: "templates/splash.html",
data: {
rule: function(User) {
//Splashscreen should only appear if User wasn't
//initialized yet ergo at the very beginning.
if(User.initialized) {
return { access: false, alternateRoute: "exit" };
}
return { access: true, alternateRoute: undefined};
}
}
});

Two ways of changing states

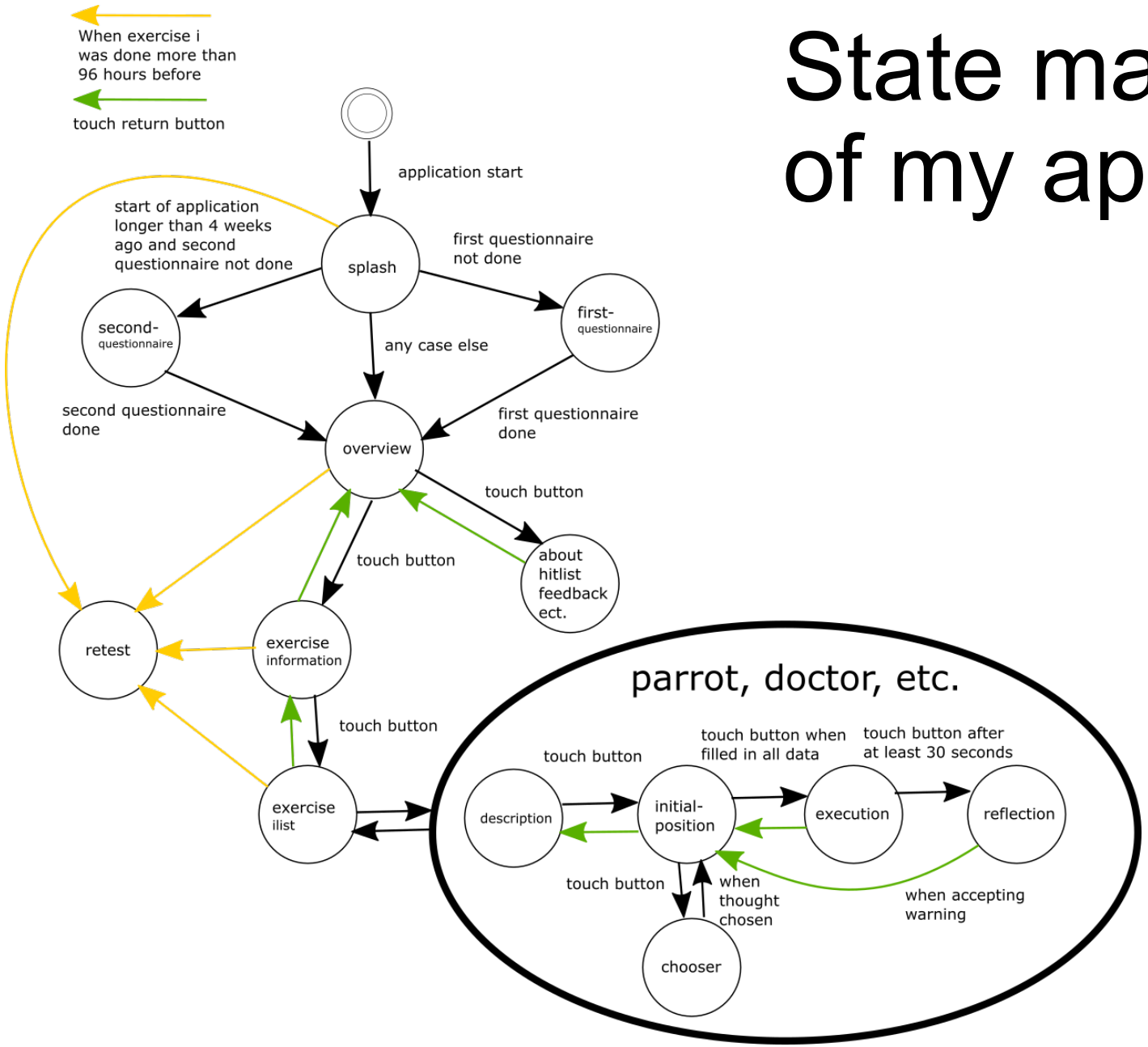
ui-sref

```
<a class="button button-outline icon-left ion-home" ui-sref="overview"></a>
```

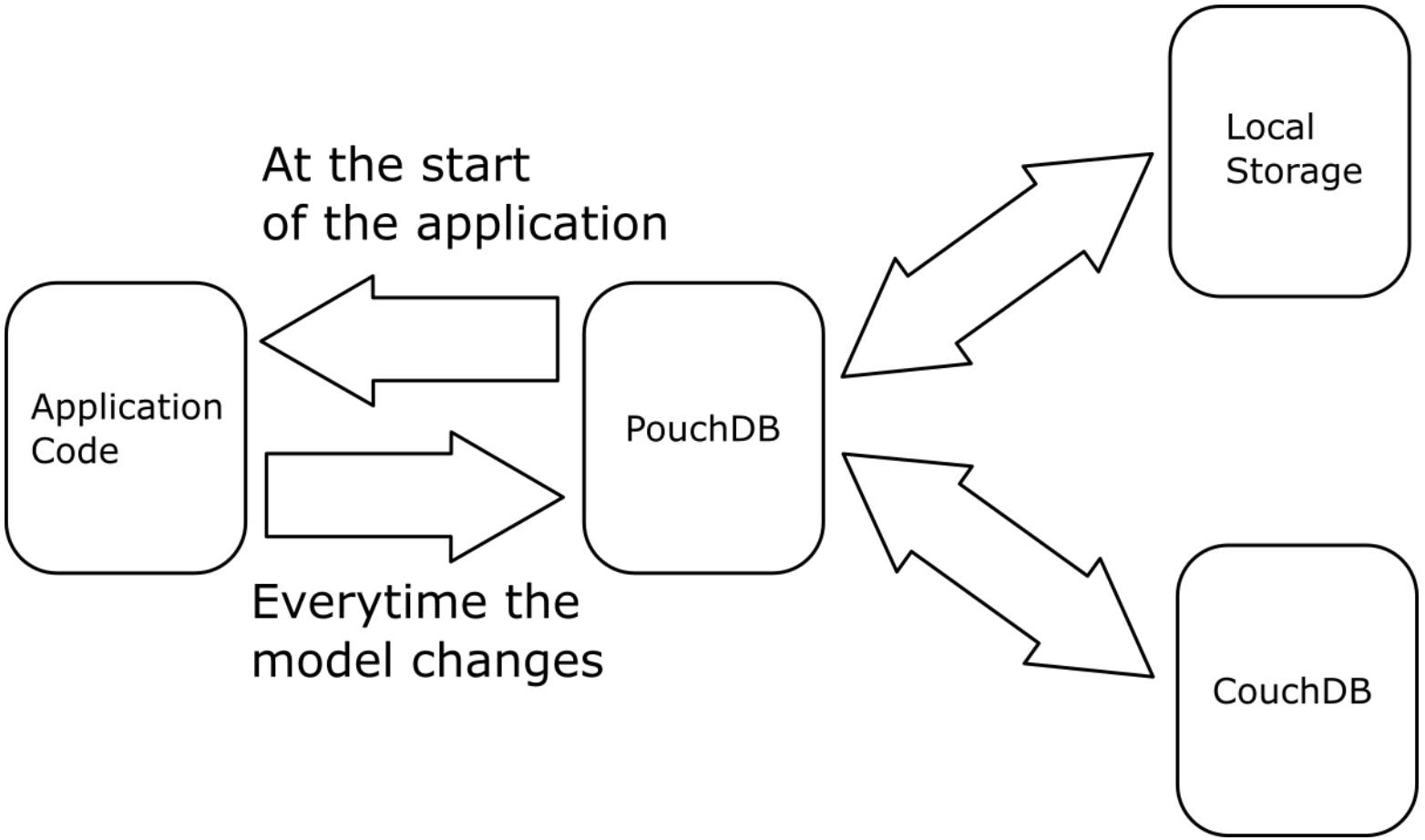
programmatically

```
$state.go("exit");
```

State machine of my application



PouchDB

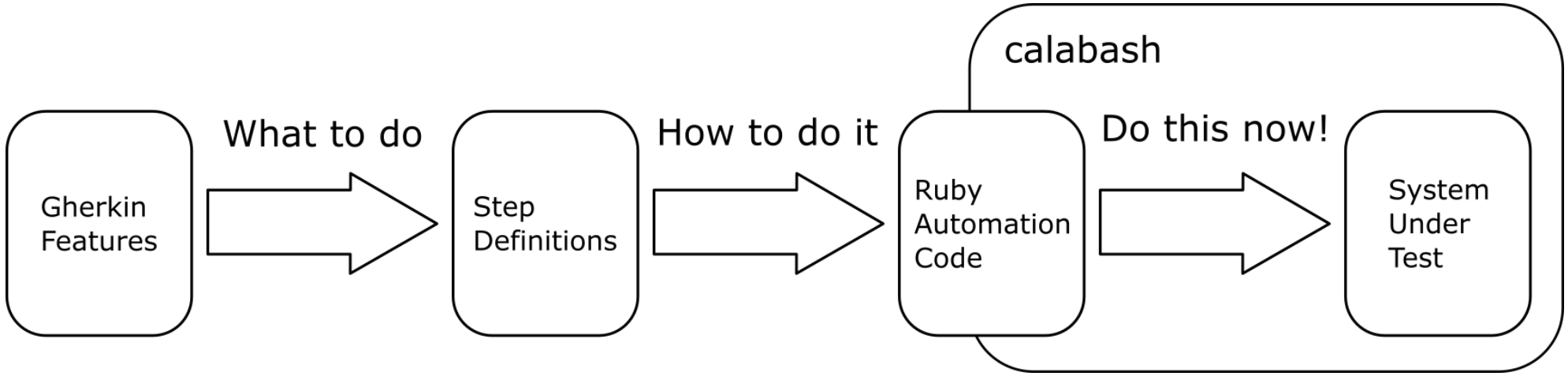

At the start
of the application
Application
Code
PouchDB
Local
Storage
Everytime the
model changes
CouchDB

Design choice for storage

- No live binding between model and permanent storage
  - Model is loaded completely at the beginning from the permanent storage
  - Changes in the model must always be stored in the model and in the permanent storage separately
  - Advantages: fast-access to the model, simple manipulation in tests with no need to manipulate permanent storage, promises have to be resolved only once at the beginning of the application (less error prone)
  - Disadvantages: each change has to be stored twice (more error prone), with big data it uses large portions of the working memory

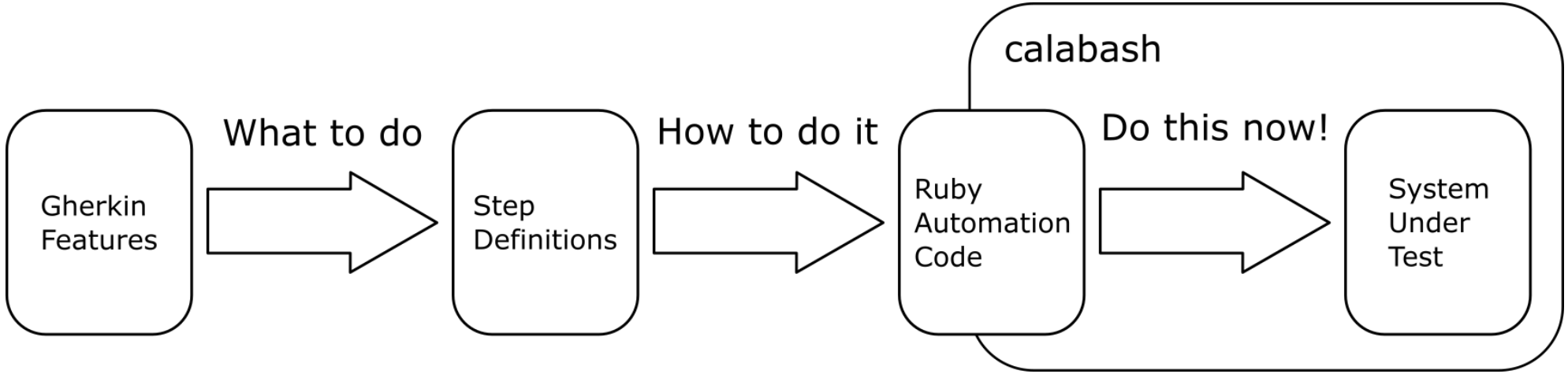
Automated Acceptance Testing

Gherkin and step definitions in calabash



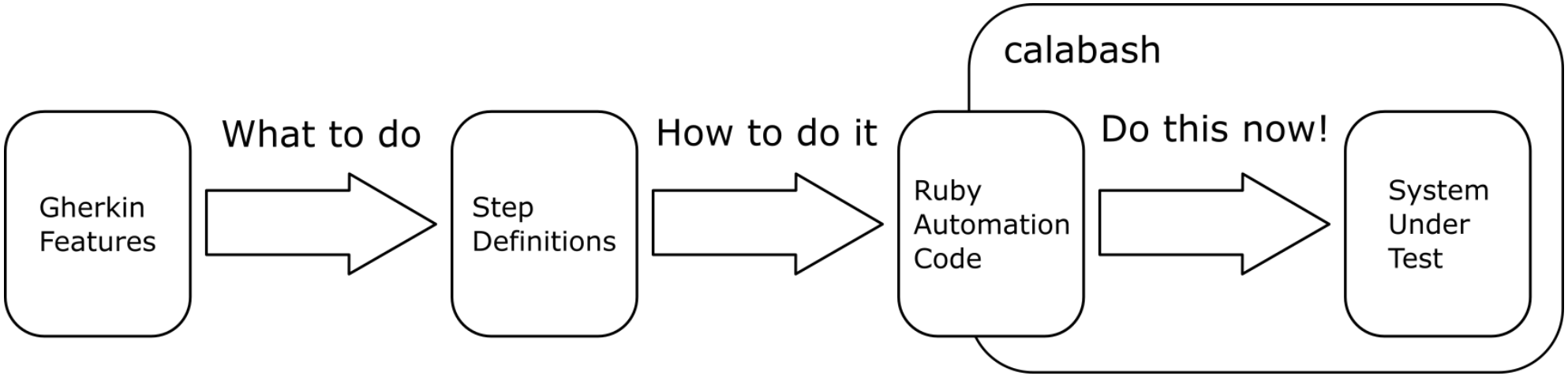
Slightly altered from Wynne, M. &. (2012). *The Cucumber Book.* Raleigh: Pragmatic Programmers, llc., page 41

Gherkin and Step Definitions in Calabash



Slightly altered from Wynne, M. &. (2012). *The Cucumber Book.* Raleigh: Pragmatic Programmers, llc., page 41

Gherkin

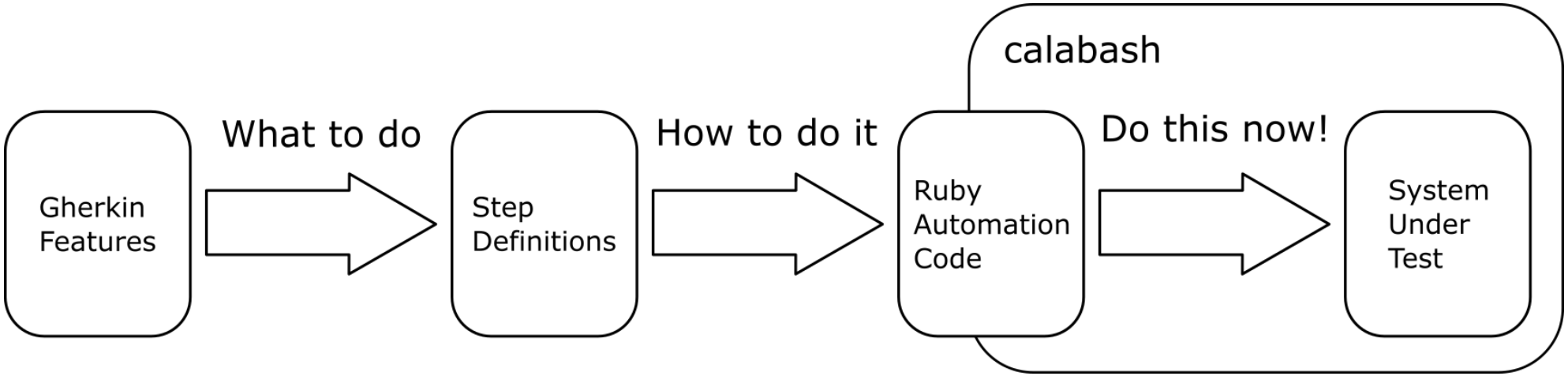



```
1 Feature: Everything belonging to the Feedback
2     Background:
3         Given I have finished first questionnaire
4
5     ...
6
7     Scenario: As a user I can return to the overview page
8         Given I am on the "feedback" page
9         When I touch home button
10        Then I see the "overview" page
11
12    ...
```

Keywords: feature, scenario outline, scenario, background, given, when, then, and. but

Slightly altered from Wynne, M. &. (2012). *The Cucumber Book.* Raleigh: Pragmatic Programmers, llc., page 41

Gherkin

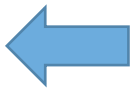



```
1 Feature: Everything belonging to the Feedback
2     Background:
3         Given I have finished first questionnaire
4
5     ...
6
7     Scenario: As a user I can return to the overview page
8         Given I am on the "feedback" page
9         When I touch home button
10        Then I see the "overview" page
11
12    ...
```

Keywords: feature, scenario outline, scenario, background, given, when, then, and. but

Slightly altered from Wynne, M. &. (2012). *The Cucumber Book.* Raleigh: Pragmatic Programmers, llc., page 41

Step Definitions



Gherkin step, may consist strong ruby pattern matching, e.g. (.*?)

```ruby
When(/^I touch home button$/) do
    go_home
end

def go_home()
    button = nil

    # This button can only be found if
    # the home button is belongs to class "ion-home".
    wait_for(:timeout => 5, :timeout_message => "No home button could be found.") {
        results = query("cordovaWebView css:'*.ion-home'")

        button = nil

        if(results.size() == 1)
            button = results[0]
        elsif (results.size() > 1)
            fail("Ambigeous buttons found. Couldn't find out which one is home button. Found #{results.size()} buttons.\n" + results.to_s);
        end

        not button.nil?
    }

    touch(button)
end
```

Ruby Automation Code

```ruby
When(/^I touch home b
    go_home
end

def go_home()
    button = nil

    # This button can only be found if
    # the home button is belongs to class "ion-home".
    wait_for(:timeout => 5, :timeout_message => "No home button cou
        results = query("cordovaWebView css:'*.ion-home'")

        button = nil

        if(results.size() == 1)
            button = results[0]
        elsif (results.size() > 1)
            fail("Ambigeous buttons found. Couldn't find out which one is home button. Found #{results.size()} buttons.\n" + results.to_s);
        end

        not button.nil?
    }

    touch(button)
end
```

Ruby automation code created by me

Ruby automation code provided by calabash

Live App Demonstration

Conclusion

- Each framework fulfilled a specific requirement/solved a specific problem
- Tests (cucumber/calabash) help to communicate requirements and are a nice way of documentation
- Successful application which worked on iOS and Android