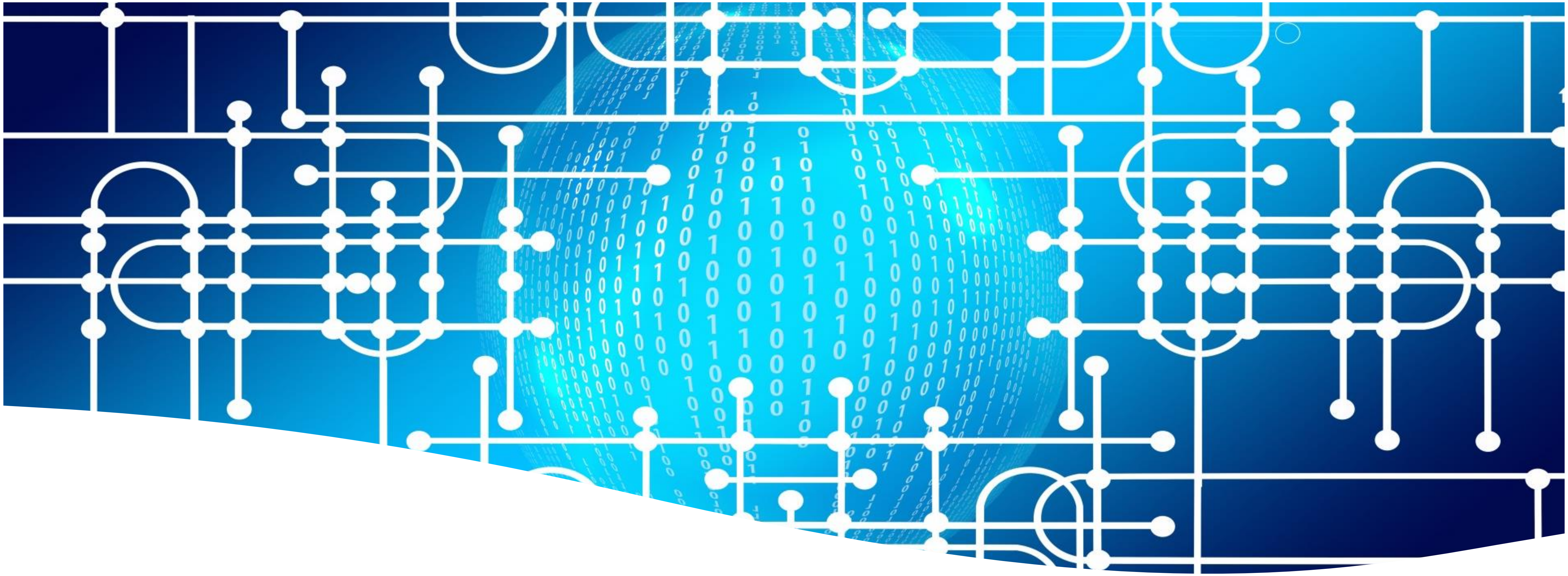


Frontend Application Development

React Native Navigation

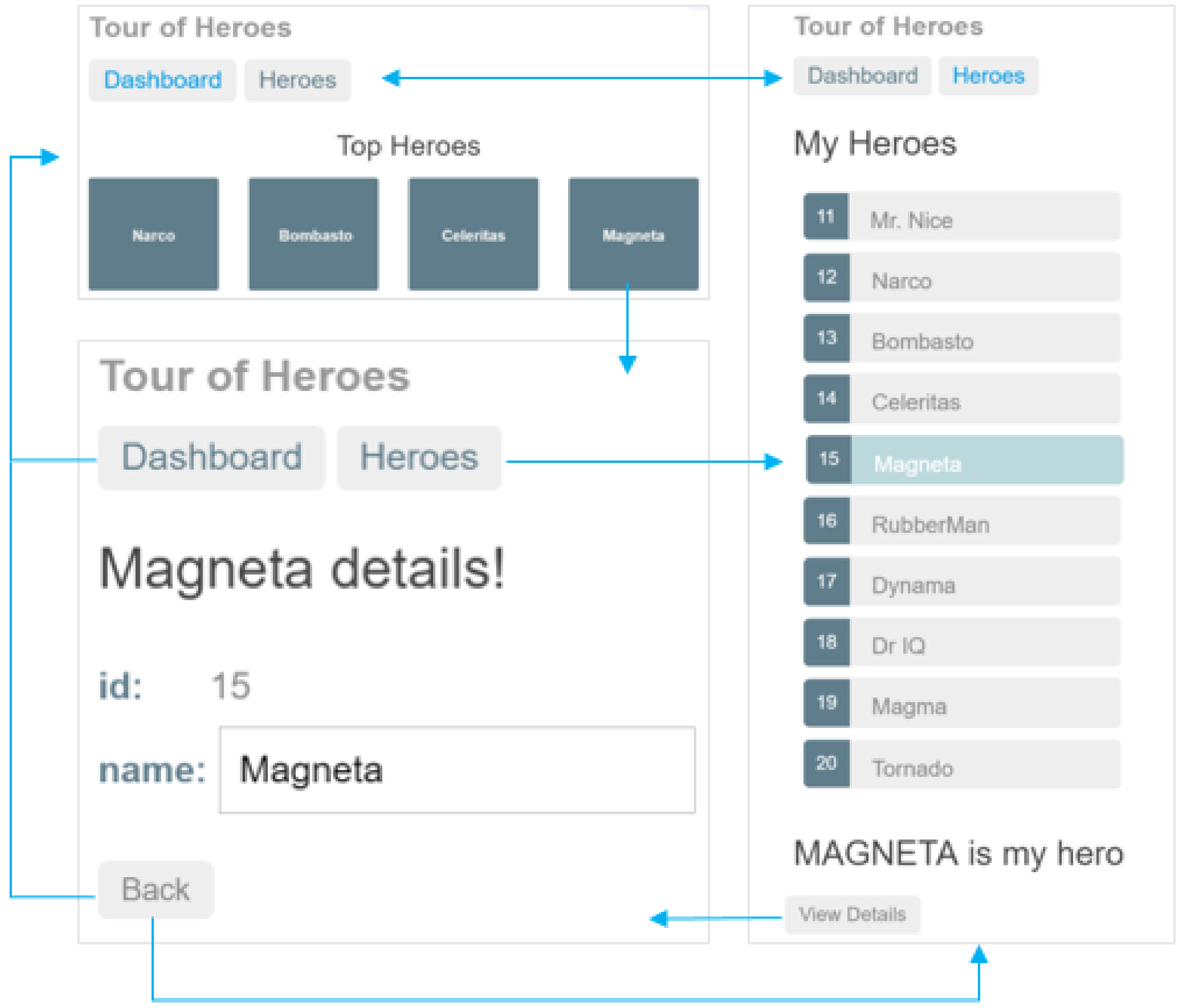
Where to Find The Code and Materials?

https://github.com/iproduct/react-native-training

Contemporary Web Applications

- Provide better **User Experience (UX)** by:
  - more interactive
  - loading and reacting faster in response (or even anticipation) of user's moves
  - able to work offline
  - supporting multiple devices and screen resolutions (responsive design)
  - are following design metaphors consistently (e.g. **Google Material Design - MD**)
  - looking more like desktop application than static web page

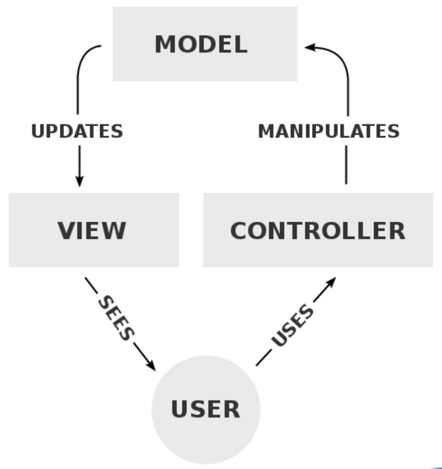
Single Page Applications (SPA)

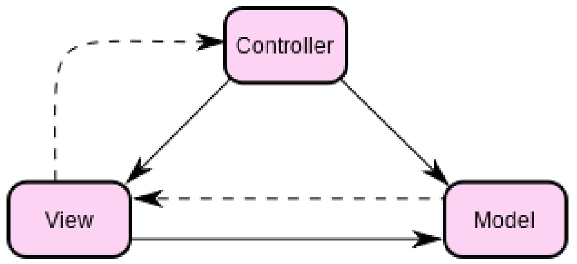


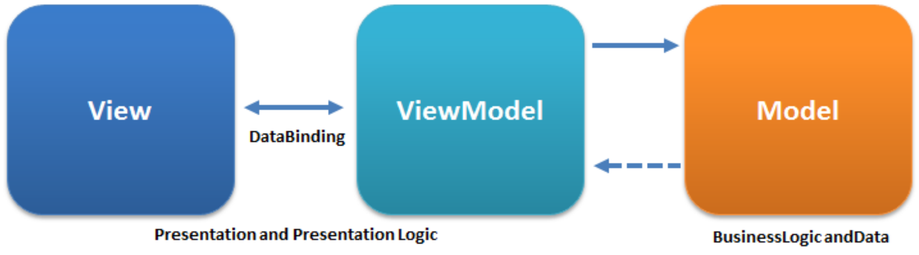
Source: Angular 2 Tutorial: Routing https://angular.io/docs/ts/latest/tutorial/toh-pt5.html
License: CC BY 4.0.

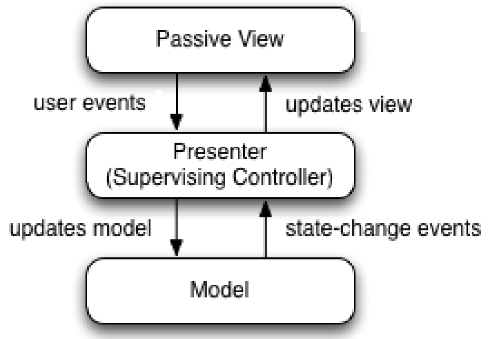
MVC Comes in Different Flavors

- MVC
- MVVM
- MVP









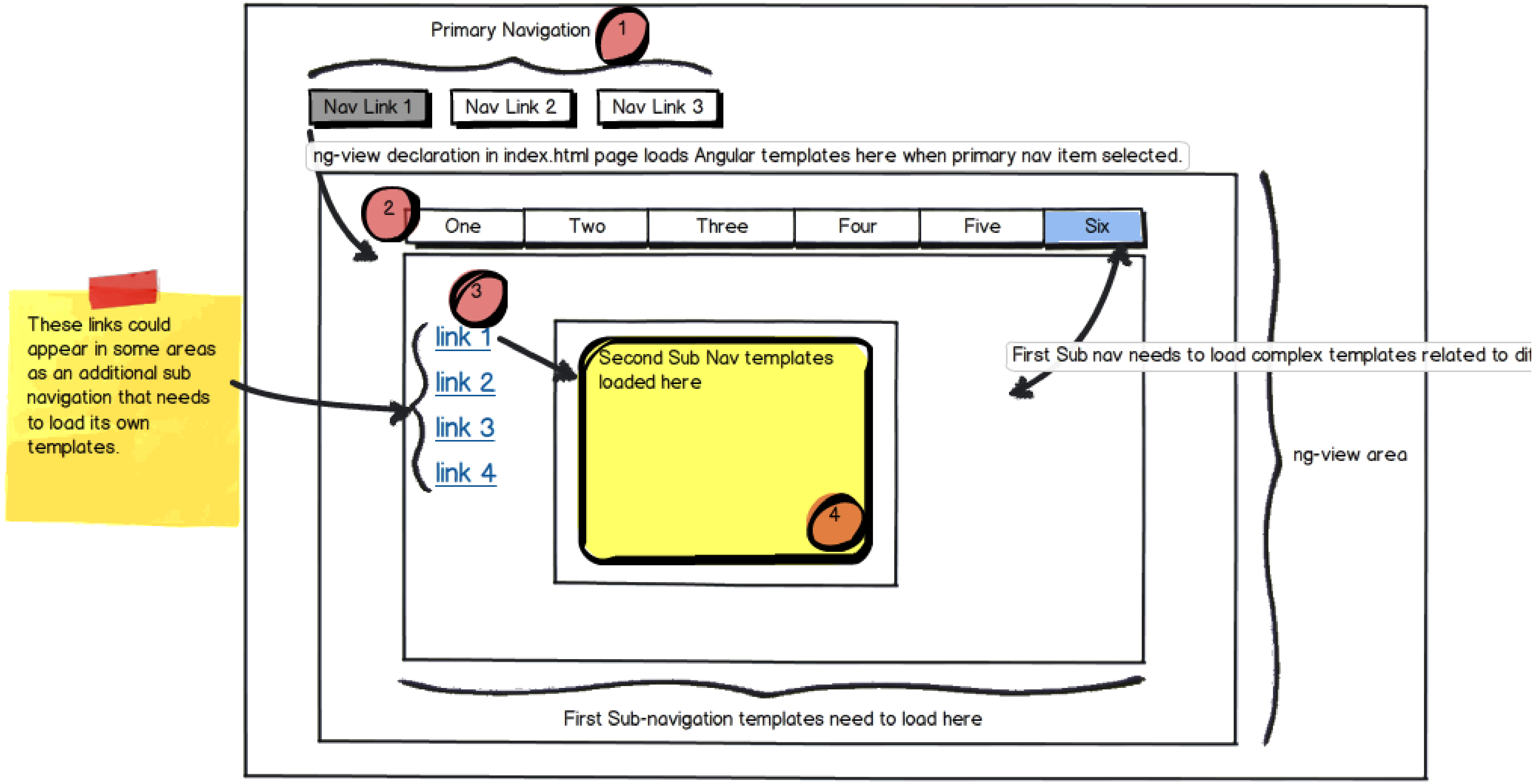
Why SPA?

- Page does not flicker – seamless (or even animated) transitions
- Less data transferred – responses are cached
- Only raw data, not markup
- Features can be loaded on demand (lazy) or in background
- Most page processing happens on the client offloading the server: REST data services + snapshops for crawlers (SEO)
- Code reuse – REST endopints are general purpose
- Supporting multiple platforms (Web, iOS, Android) → React Native

Developing Sinagle Page Apps (SPA) in 3 steps

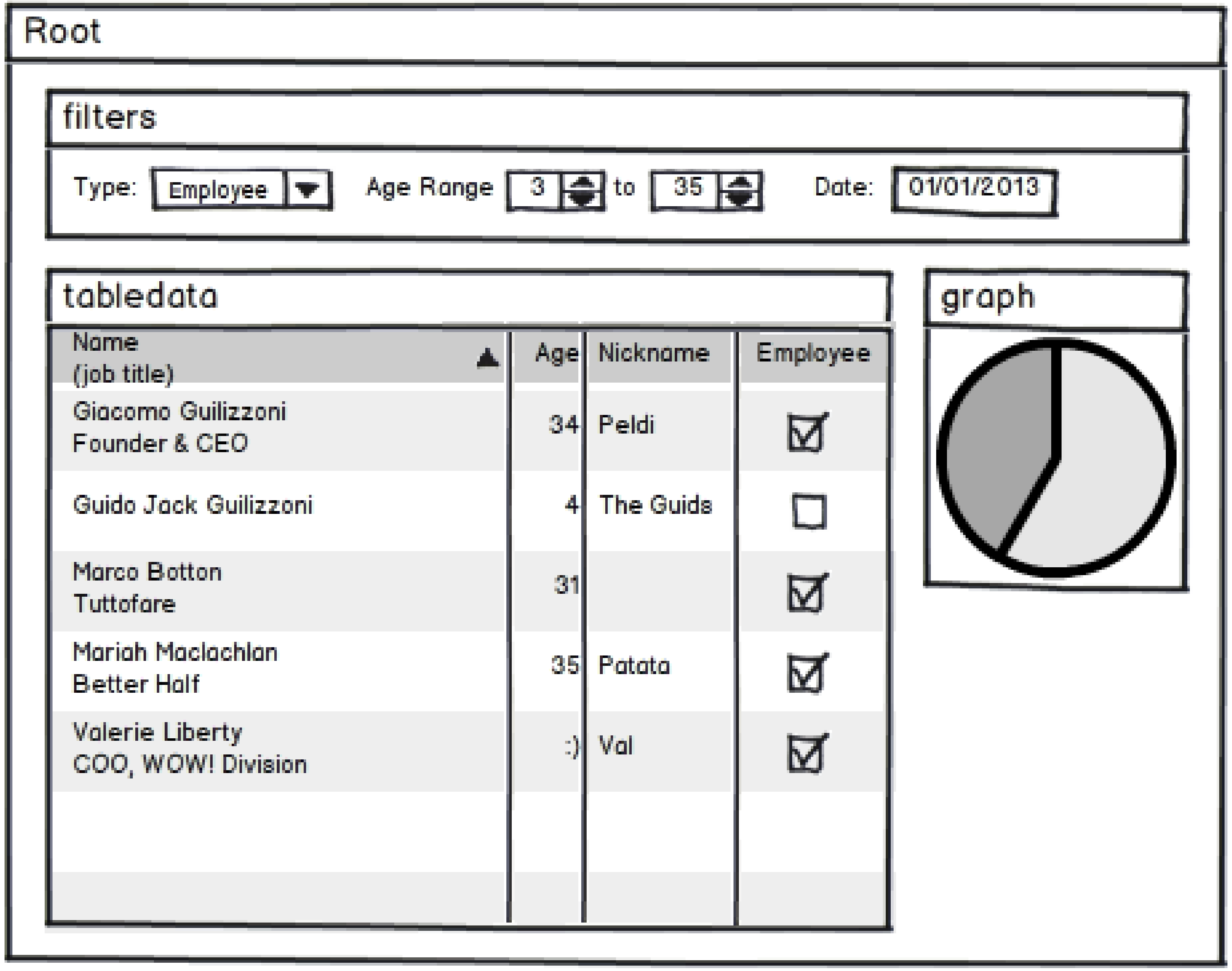
1) Setting up a build system – *npm, webpack, gulp* are common choices, *babel, typescript, JSX/TSX, CSS preprocessors (SASS, SCSS, LESS), jasmine, karma, protractor, live servers* ...
2) Designing front-end architecture components – *views & layouts* + *view models* (presentation data models) + *presentation logic* (event handling, messaging) + *routing paths* (essential for SPA)
3) Better to use component model to boost productivity and maintainability.
4) End-to-end application design – front-end: wireframes → views,
5) data entities & data streams → service API and models design,
6) **sitemap → router config**

Hierarchical Routing


Primary Navigation
1
Nav Link 1
Nav Link 2
Nav Link 3
ng-view declaration in index.html page loads Angular templates here when primary nav item selected.
2
One
Two
Three
Four
Five
Six
3
These links could appear in some areas as an additional sub navigation that needs to load its own templates.
link 1
link 2
link 3
link 4
Second Sub Nav templates loaded here
4
First Sub nav needs to load complex templates related to di
ng-view area
First Sub-navigation templates need to load here

Source: http://stackoverflow.com/questions/12863663/complex-nesting-of-partials-and-templates
Author: PhillipKregg

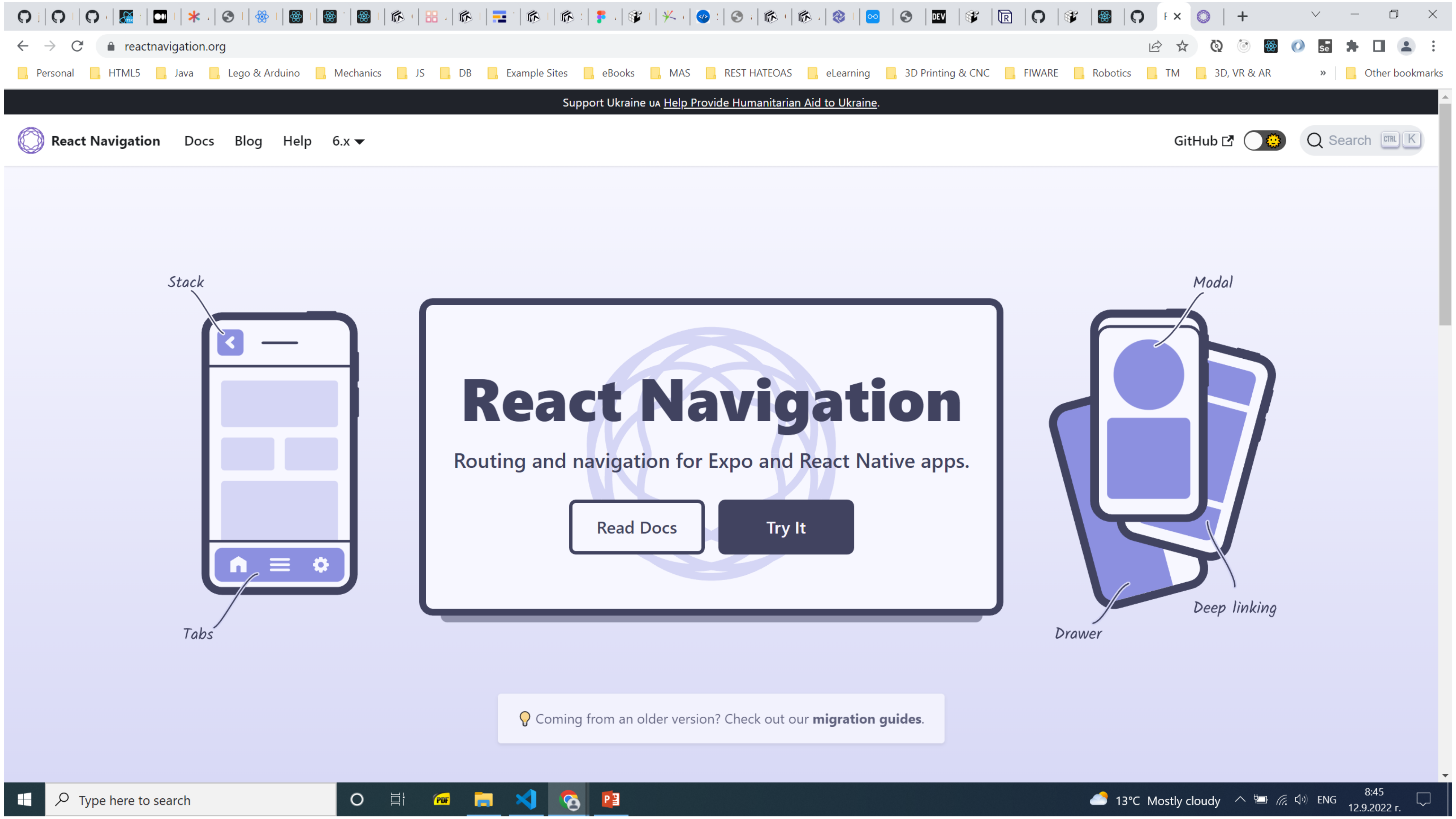
SPA with Multiple Router Outlets



Type: Employee Age Range 3 to 35 Date: 01/01/2013

Name (job title) ▲	Age	Nickname	Employee
Giacomo Guilizzoni Founder & CEO	34	Peldi	☑
Guido Jack Guilizzoni	4	The Guids	☐
Marco Botton Tuttofare	31		☑
Mariah Maclachlan Better Half	35	Patata	☑
Valerie Liberty COO, WOW! Division	:)	Val	☑

React Navigation - Routing and navigation for Expo and React Native apps



React Navigation

Routing and navigation for Expo and React Native apps.

Coming from an older version? Check out our **migration guides**.

Getting Started with React Navigation

- Create new project using *create-react-app*:

```
yarn add @react-navigation/native

npx expo install react-native-screens react-native-safe-area-context
```

- Wrapping your app in NavigationContainer:

```
export default function App() {
  return (
      <NavigationContainer>{/* Rest of your app code */}</NavigationContainer>
  );
}
```

Types of Navigation

- Stack Navigation - https://reactnavigation.org/docs/hello-react-navigation
- Tab navigation - yarn add @react-navigation/bottom-tabs

https://reactnavigation.org/docs/tab-based-navigation

- Drawer navigation - https://reactnavigation.org/docs/drawer-based-navigation
- Native Stack Navigator - https://reactnavigation.org/docs/native-stack-navigator
- Material Bottom Tabs / Top Tabs Navigator - https://reactnavigation.org/docs/material-bottom-tab-navigator

Stack Navigation Example

```
import * as React from 'react';
import { View, Text } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
function HomeScreen() {
  return (
      <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
        <Text>Home Screen</Text>
      </View>
  );
}
const Stack = createNativeStackNavigator();
function App() {
  return (
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen name="Home" component={HomeScreen} />
        </Stack.Navigator>
      </NavigationContainer>
  );
}
export default App;
```

Drawer Navigation Example - I

```
import * as React from 'react';
import { Button, View } from 'react-native';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { NavigationContainer } from '@react-navigation/native';

function HomeScreen({ navigation }) {
    return (
        <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
            <Button
                onPress={() => navigation.navigate('Notifications')}
                title="Go to notifications"
            />
        </View>
    );
}

function NotificationsScreen({ navigation }) {
    return (
        <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
            <Button onPress={() => navigation.goBack()} title="Go back home" />
        </View>
    );
}
```

Drawer Navigation Example - II

```
const Drawer = createDrawerNavigator();

export default function App() {
    return (
        <NavigationContainer>
            <Drawer.Navigator initialRouteName="Home">
                <Drawer.Screen name="Home" component={HomeScreen} />
                <Drawer.Screen name="Notifications" component={NotificationsScreen} />
            </Drawer.Navigator>
        </NavigationContainer>
    );
}
```

Nesting navigators

- Each navigator keeps its own navigation history
- Each navigator has its own options
- Each screen in a navigator has its own params
- Navigation actions are handled by current navigator and bubble up if couldn't be handled
- Navigator specific methods are available in the navigators nested inside
- Nested navigators don't receive parent's events
- Parent navigator's UI is rendered on top of child navigator

Thank’s for Your Attention!

Trayan Iliev

IPT – Intellectual Products & Technologies

http://iproduct.org/

http://robolearn.org/

https://github.com/iproduct

https://twitter.com/trayaniliev

https://www.facebook.com/IPT.EACAD