

# Getting started with Web3 and React

5 ingredients for awesome DApps

Razvan Dinicut  
@rdinicut



# TweetStream

Hi Alice!

Message

Mood

😊

Send tweet

🎉

Alice says : Yes we are, this is great!

🎉

Brian says : Are you having fun?

📚

Brian says : What's up guys?

👤

Alice says : This day is great!

🐪

Alice says : I am so happy

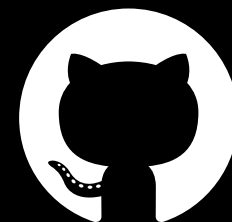
# TweetStream

Register a user using Metamask

Send a tweet

Load all the past tweets using event logs

Subscribe to all incoming tweets



<https://github.com/rdinicut/tweet-stream>

5 ingredients for awesome DApps

# 1. Developer tools



GANACHE

```
npm install ganache-cli --save-dev
```



TRUFFLE

```
npm install truffle --save-dev
```

## 2. Monorepo

```
"scripts": {  
  "start-react": "PORT=3006 react-scripts start",  
  "start-ganache": "ganache-cli --gasLimit 4712388 --networkId 1234567 ...",  
  "build": "react-scripts build",  
  "test": "react-scripts test --env=jsdom --coverage",  
  "eject": "react-scripts eject",  
  "migrate": "rm -rf ./src/contracts/* && truffle migrate"  
},
```

```
var path = require('path');  
module.exports = {  
  contracts_build_directory: path.join(__dirname, "src/contracts/"),  
  ...  
};
```

## 2. Monorepo

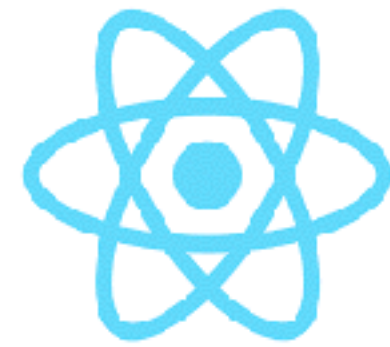
```
import * as tweetStreamConfig from '../contracts/TweetStream.json';
let TweetStream = null;
let selectedNetwork = null;
let supportedNetworks = Object.keys(tweetStreamConfig.networks);

export default class Contracts {
  static getSupportedNetworks = () => {
    return supportedNetworks;
  };

  static setNetwork = (networkId) => {
    selectedNetwork = networkId;
    TweetStream = web3.eth.contract(...//Code to create web3 contract;
  }

  static TweetStream() {
    if (!TweetStream) throw new Error('error')
    return TweetStream;
  }
}
```

### 3. Framework, Styling and Components



REACT

```
npm install react react-dom --save
```



STYLED COMPONENTS

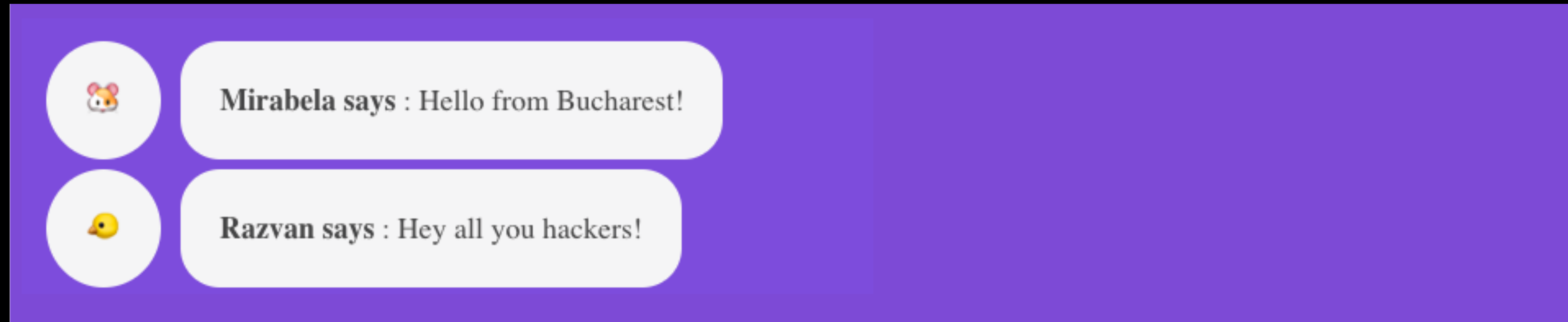
```
npm install styled-components --save
```



```
npm install grommet@next --save
```



### 3. Framework, Styling and Components



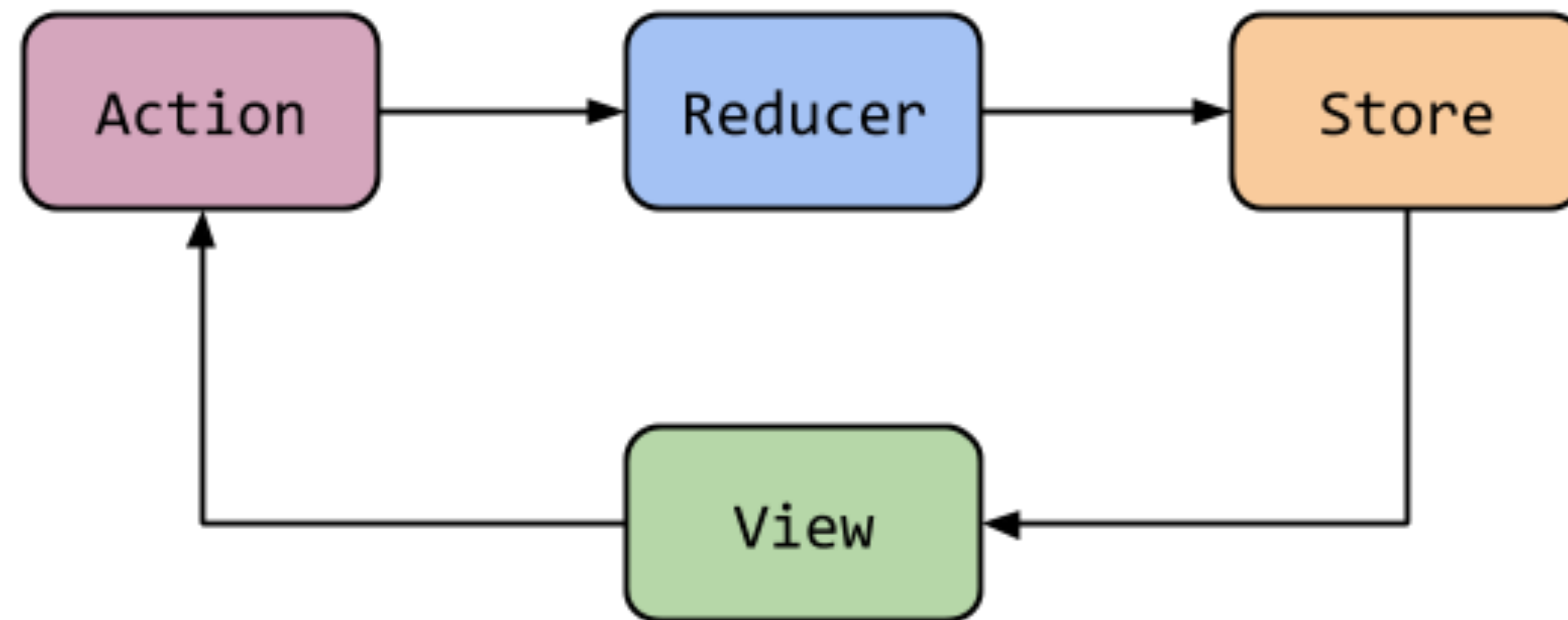
```
<Box gap="small" direction="row" margin="xsmall"
    animation={{type: "fadeIn", duration: 500}}>
  <Box justify='center' round="full"
    pad="medium" background="light-1">
    {String.fromCharCode(tweet.mood)}
  </Box>
  <Box round="medium" pad="medium" background="light-1">
    <Text><b>{tweet.name}</b> : {tweet.message}</Text>
  </Box>
</Box>
```

## 4. Redux



REDUX

npm install --save redux



## 5. Observables

Extension of the [observer pattern](#) to support sequences of data/events and adds operators that allow you to compose sequences together declaratively



RXjs

`npm install rxjs --save`



RXjs

+



REDUX

=



+



=



REDUX-OBSERVABLE

## 5. Observables

```
import {ajax} from 'rxjs/ajax';

// action creators
const fetchUser = username => ({type: FETCH_USER, payload: username});
const fetchUserFulfilled = payload => ({type: FETCH_USER_FULFILLED, payload});

// epic
const fetchUserEpic = action$ => {
  return action$.ofType(FETCH_USER)
    .mergeMap(action =>
      ajax.getJSON(`https://api.github.com/users/${action.payload}`).pipe(
        map(response => fetchUserFulfilled(response))
      )
    )
}

// later...
dispatch(fetchUser('torvalds'));
```

## 5. Observables

Call the transaction function

Wait for the transaction hash

Listen to a web3 event

## 5. Observables

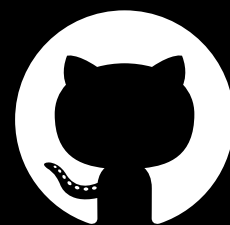
```
return action$.ofType(TWEET_SUBMIT)
  .switchMap(act => {
    return observableFromWeb3(
      cb => Contracts.TweetStream().tweet(act.message, act.mood, cb)
    )
    .switchMap((transaction) => {
      return observableFromWeb3(cb => Contracts.TweetStream().newTweet({
        message: action.message,
        mood: action.mood
      })).watch(cb))
    })
    .map(tweetSubmitted)
    .catch(e => Observable.of(tweetFailed(e)))
  })
```

## 6. Extra spice

REACT DAPP REQUIREMENTS

npm install --save-dev react-dapp-requirements

```
<DappRequirements  
  supportedNetworks={["99999","4"]}   
  onNetworkIdReceived={this.onNetworkIdReceived}  
  onAccountChange={this.onAccountChange}>  
  <App/>  
</DappRequirements>
```



<https://github.com/centrifuge/react-dapp-requirements>



Thank you for joining!

 Centrifuge  
[centrifuge.io](https://centrifuge.io)

Razvan Dinicut  
[razvan@centrifuge.io](mailto:razvan@centrifuge.io)