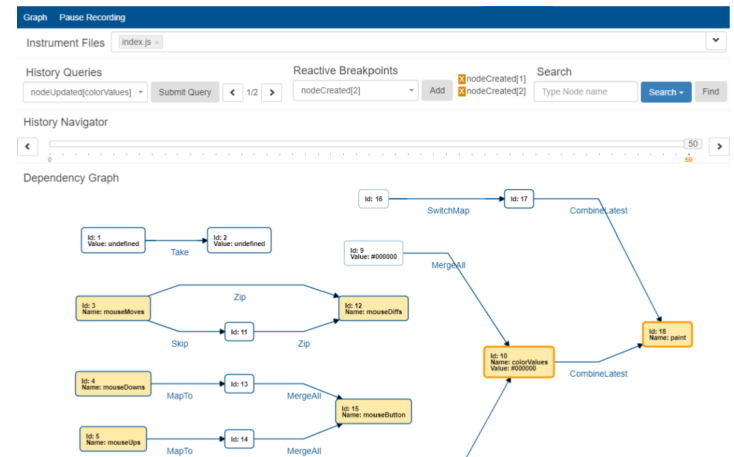
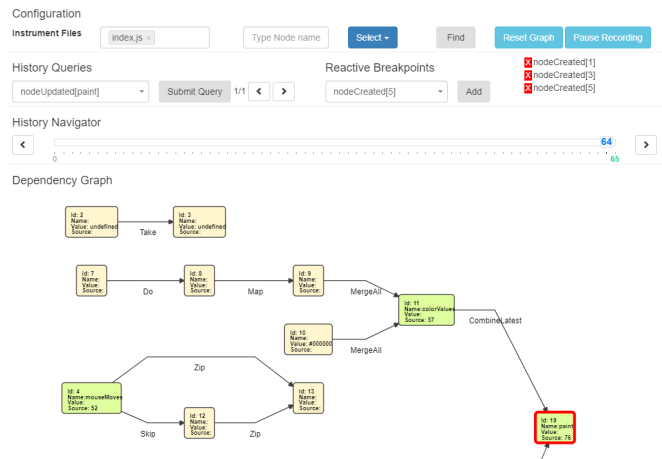


# Advancing the Chrome Reactive Inspector

Bachelor-Thesis by Benedikt Gross  
Supervisor Prof.Dr.Guido Salvaneschi



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Reactive Programming

- multiple values over time
- observables
- event streams
- Implementations in all major programming languages



ReactiveX

source: <http://reactivex.io/>



Bacon.js

source: <https://baconjs.github.io/>

# Traditional Debugging

---

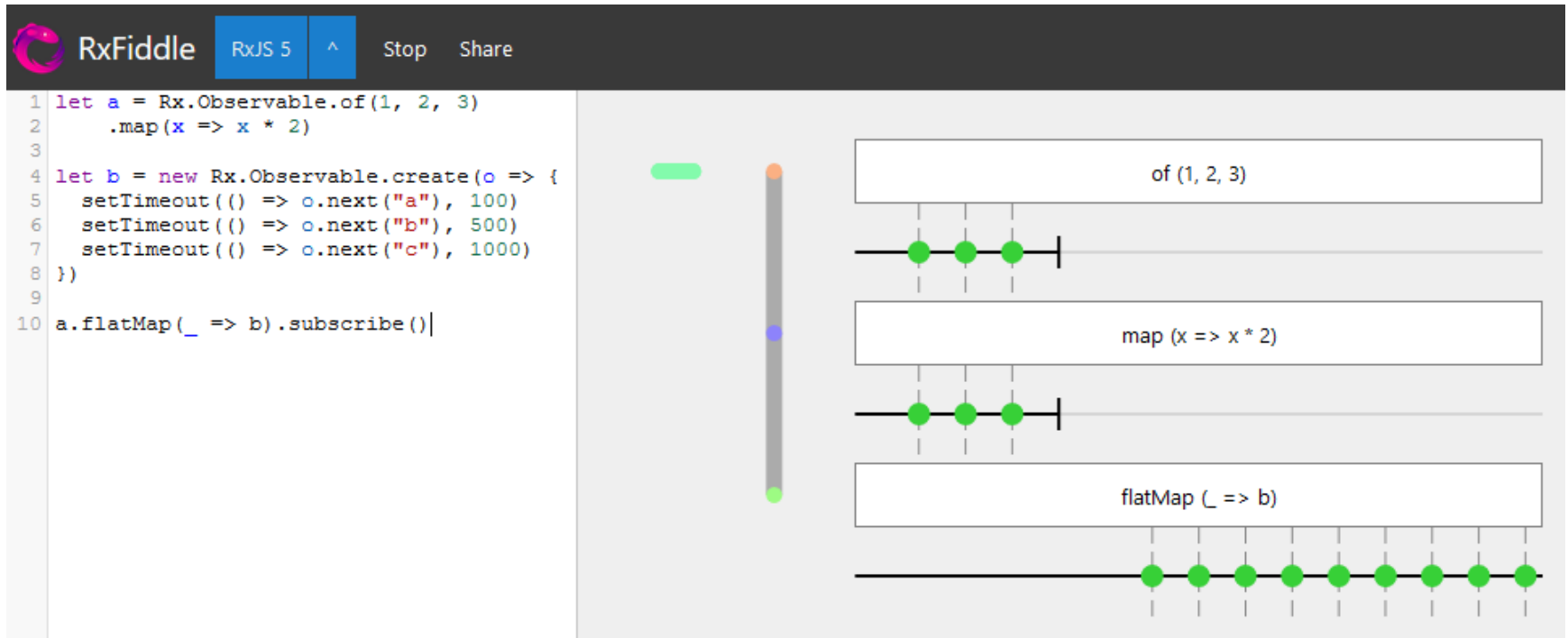
- breakpoints
- step by step execution
- event logging
- stack trace inspection
- *printf-debugging*

# Debuggin Reactive Code

---

- do-debugging
- abstraction of dependencies
- limited number of tools
- Reactive Inspector for Scala
- Chrome Reactive Inspector
- RxFiddle

# Debuggin Reactive Code - RxFiddle



## Screenshot of the RxFiddle Demo

Source: <https://rxfiddle.net/>



## The Chrome Reactive Inspector

# The previous User Interface



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Configuration

Instrument Files

index.js x

Type Node name

Select ▾

Find

Reset Graph

Pause Recording

## History Queries

nodeUpdated[paint] ▾

Submit Query

1/1



## Reactive Breakpoints

nodeCreated[5] ▾

Add

- ✗ nodeCreated[1]
- ✗ nodeCreated[3]
- ✗ nodeCreated[5]

## History Navigator



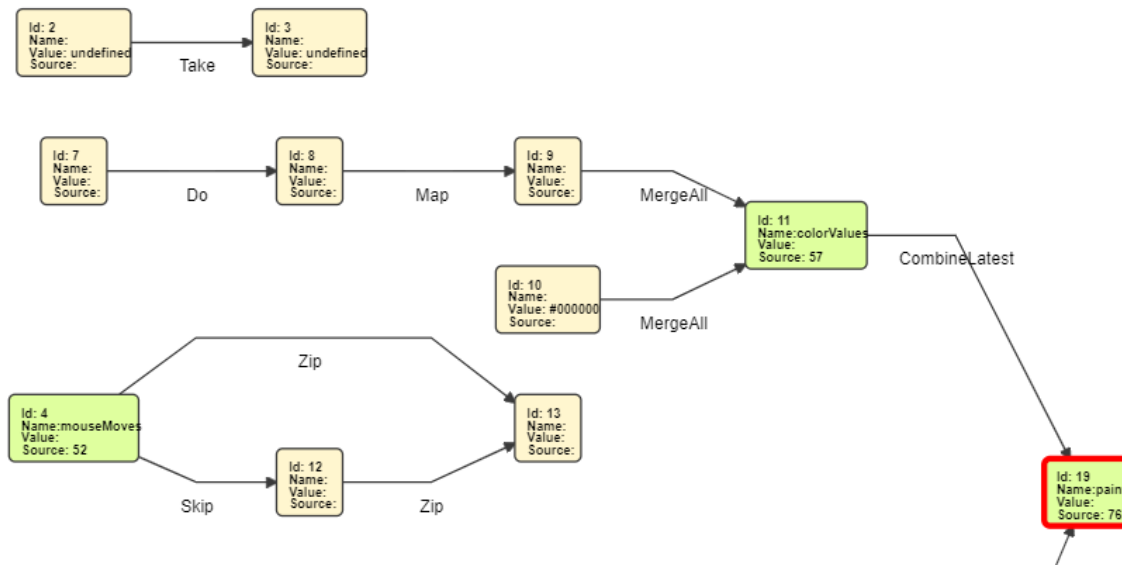
64



0

65

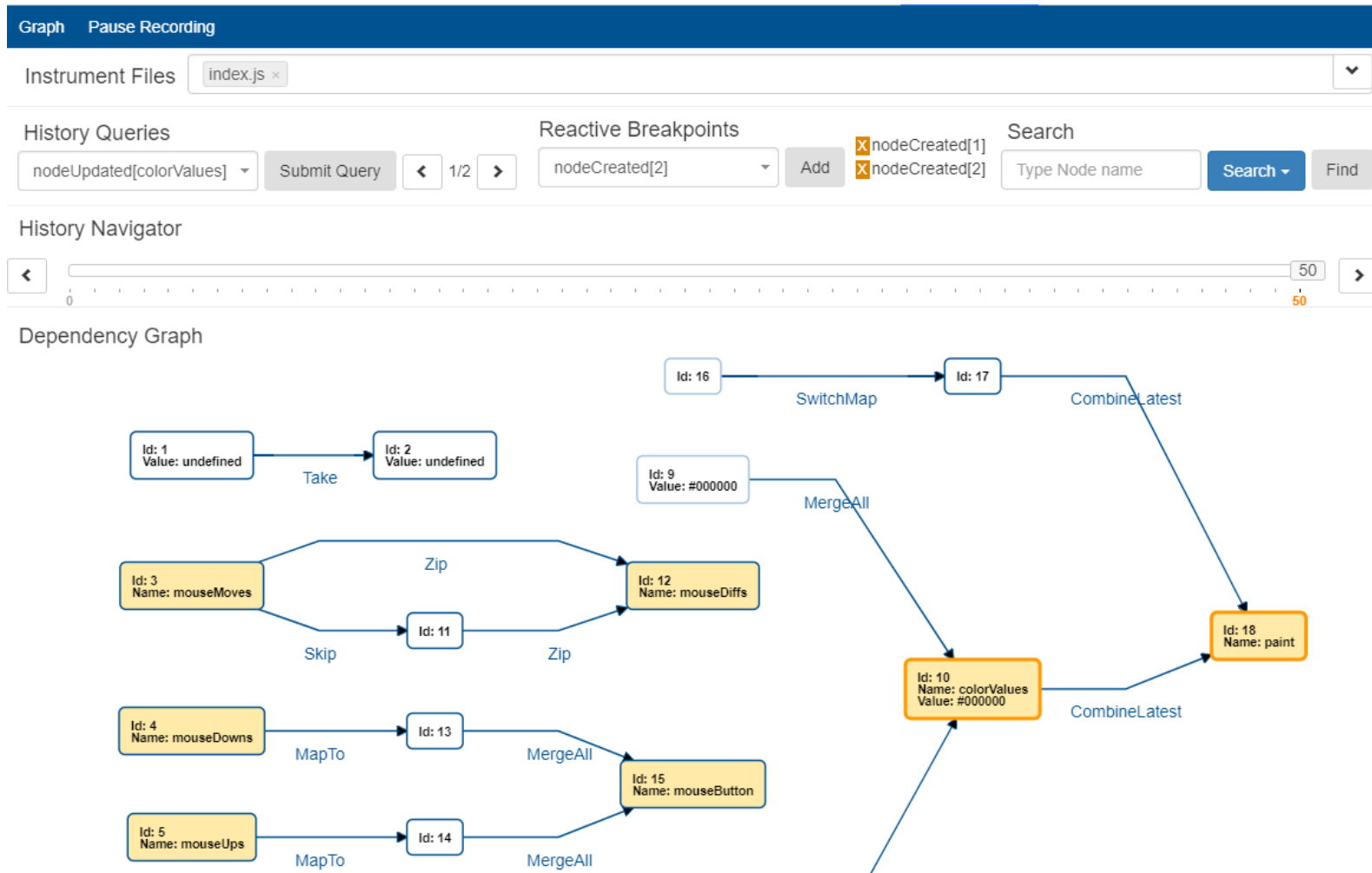
## Dependency Graph



# The new User Interface



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



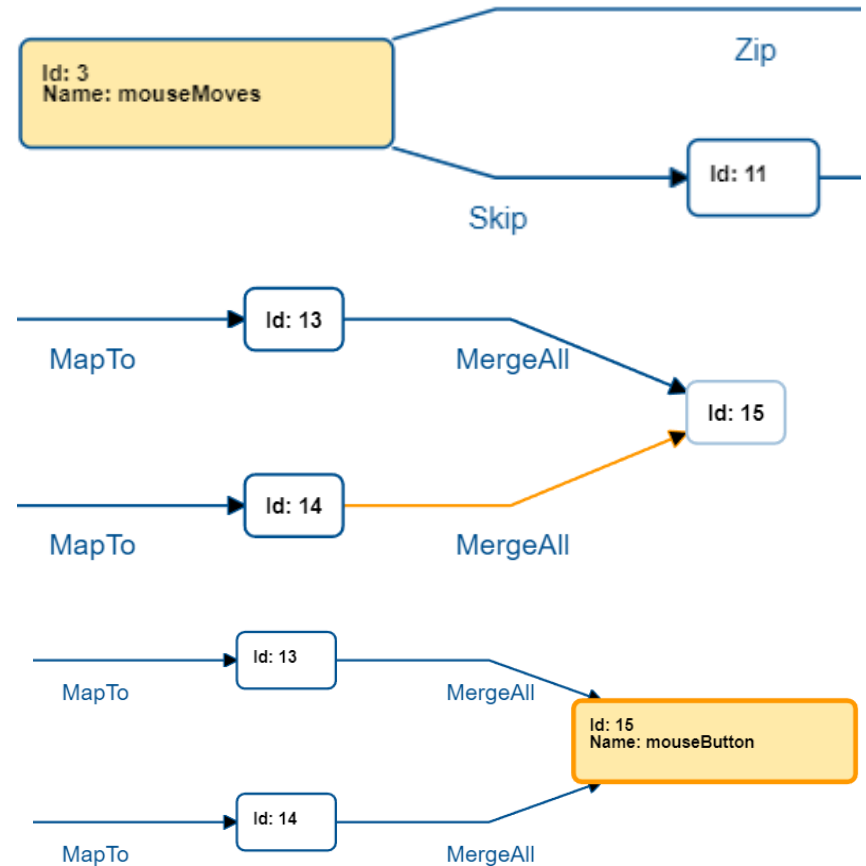
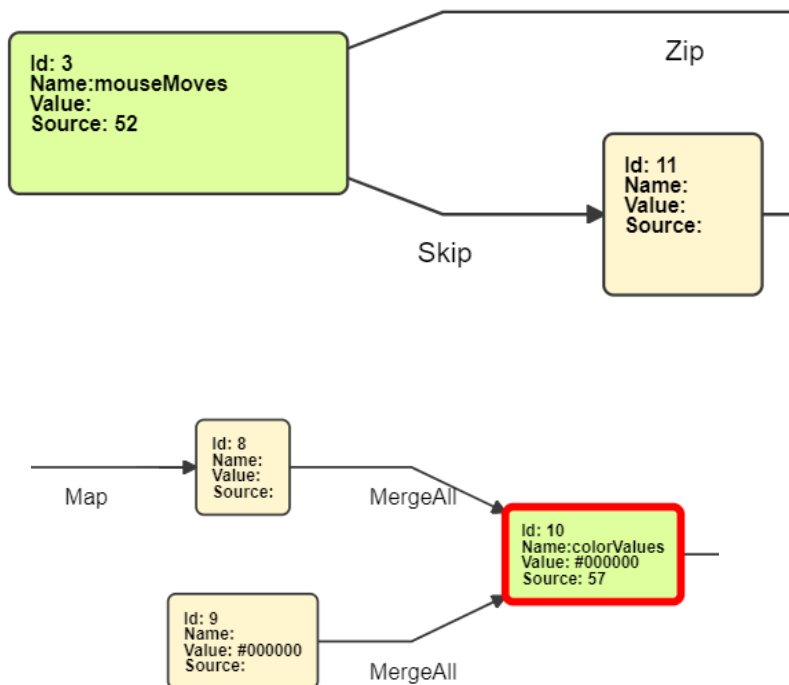


# Advancing the User Interface

## *Reducing the cognitive load*



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Advancing the User Interface

## *Reducing the cognitive load*



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Graph Pause Recording

Instrument Files index.js

History Queries Submit Query < 0/0 > Reactive Breakpoints Add Search Type Node name Search Find

History Navigator < 0 50 >

Dependency Graph

```
graph LR; Id16[Id: 16] -- SwitchMap --> Id17[Id: 17]; Id17 -- CombineLatest -->
```

Graph Pause Recording

History Navigator < 0 50 >

Dependency Graph

```
graph LR; Id16[Id: 16] -- SwitchMap --> Id17[Id: 17]; Id17 -- CombineLatest -->
```

Graph Pause Recording

Instrument Files index.js

History Queries

# Connecting abstract graph with JavaScript code

- provide additional context
- merge both views

Method Chaining:

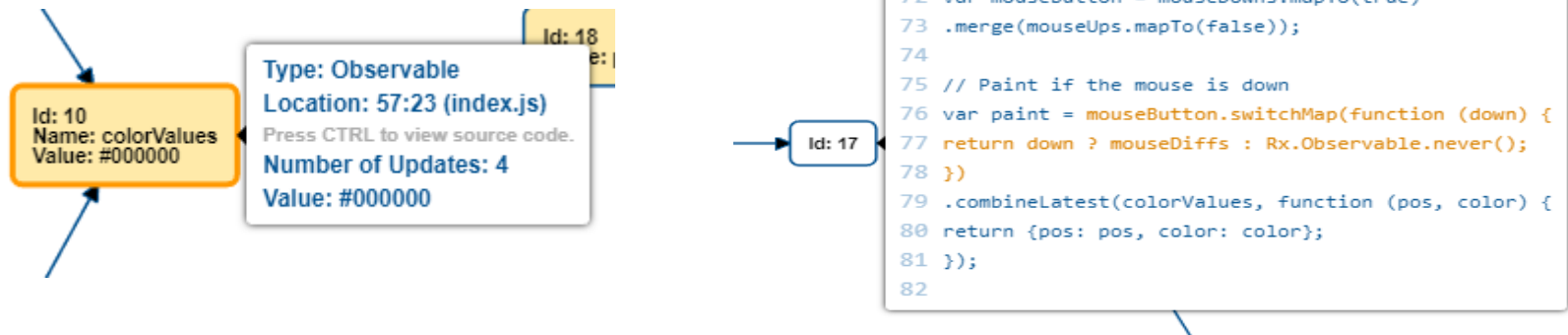
```
var intervalObservable = Rx.Observable.interval(5)
    .timestamp()
    .bufferCount(2, 1)
    .map(function (w) {
        return w[1].timestamp - w[0].timestamp;
    })
    .share();
```

# Connecting abstract graph with JavaScript code

- difficulty: instrumented JavaScript code

```
$sonWallet = J$.W(593, '$sonWallet', J$.F(585, J$.I(typeof $ === 'undefined'  
? $ = J$.R(569, '$', undefined, true, true) : $ = J$.R(569, '$', $, true,  
true)), false)(J$.T(577, '#wallet-son', 21, false)), J$.I(typeof
```

- covers *nearly* all nodes



# Connecting abstract graph with JavaScript code

- *Source Code Tooltips*

## Pro

- compliments dependency graph
- easily accessible
- highlight code
- existing workflow

## Con

- only snippets
- no search feature
- limited when using references to functions

## Function references:

```
function createFunction(add) {  
  return (value) => value + add;  
}  
  
let fatherWalletValue = sonWalletValue  
  .map(createFunction(10));
```

# Rapidly updated Observables

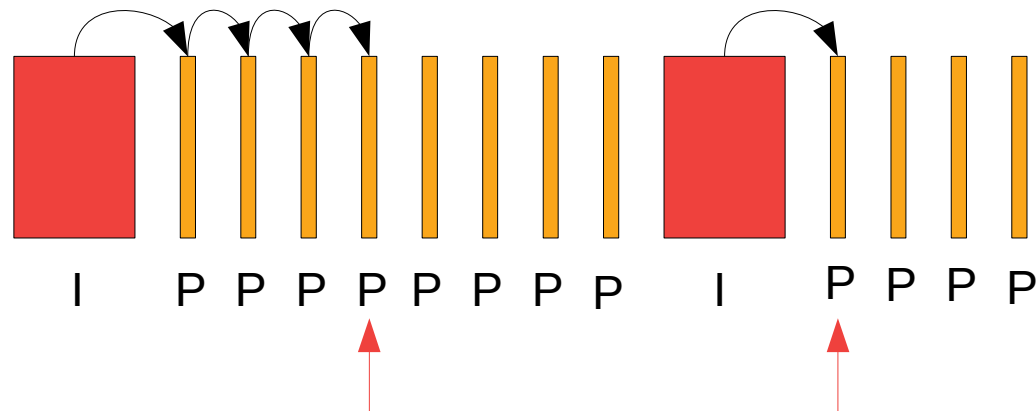
---

- timers, mouse movement, network traffic
- generate thousands of steps
- limited by recording performance
- may render CRI useless

# Rapidly updated Observables

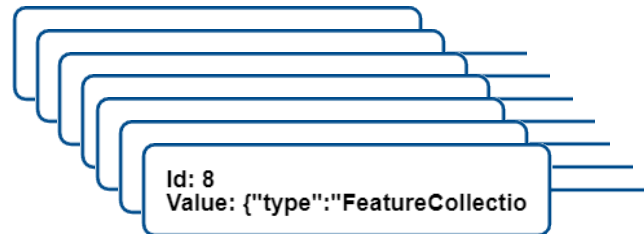
*improving performance*

- UI update throttling
- (software) Paging
- Delta Encoding
- improved similar to Video Compression



# Excessively created Observables

- describes a group of observables
- created as part of a loop
- created multiple times temporarily
- all originating from the same code
- require detection and representation in the graph





- changes to the UI
- Source Code Tooltips cover 75% of all nodes
- CPU Performance
  - Version 2: 154s
  - Version 3: 5.5s
- Memory
  - Version 2: 54mb - depending on steps
  - Version 3: 15mb - constant

# Evaluation

## *Comparison to RxFiddle*



Screenshot of RxFiddle with the  
test application

# Evaluation

## *Test Applications*

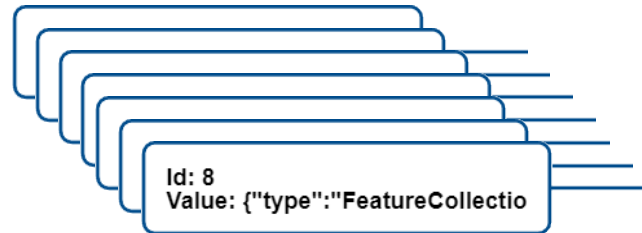
---

- verify robustness
- state of the project
- 9 specifications tested for each applicaiton
- 388 of 420 checks were successful
- notable findings:
  - Ambiguity
  - JavaScript in HTML attributes
  - breakdown of recording for a specific pattern

- module loaders
- ES6 and TypeScript support
- extend node exclusion
- excessively created observables



Source: <http://requirejs.org/logo.png>



## Advancing the Chrome Reactive Inspector

- Reducing the cognitive load on the user
- Connecting the abstract graph with JavaScript code
- Rapidly updated observables
- Excessively created Observables
- Evaluation

---

# Advancing the Chrome Reactive Inspector

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Thank you

---

# Advancing the Chrome Reactive Inspector

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Backup Slides:

- very similar source code

## Rx.js

```
addClick = Rx.Observable.fromEvent($addSon, 'click')
    .mapTo(1);
removeClick = Rx.Observable.fromEvent($removeSon, 'click')
    .mapTo(-1);
eventClick = addClick.merge(removeClick);
function plus(a, b) {
    return a + b
}
sonWalletValue = eventClick.scan(plus, 0)
fatherWalletValue = sonWalletValue
    .map(function (value) {
        return value + 10
    })
sonWalletValue.subscribe(
    function (data) {
        $sonWallet.val(data);
    }
);
```

## Bacon.js

```
addClick = $addSon.asEventStream('click');
addClickMap = addClick.map(1);
removeClick = $removeSon.asEventStream('click');
removeClickMap = removeClick.map(-1);
eventClick = addClickMap.merge(removeClickMap);
function plus(a, b) {
    return a + b
}
sonWalletValue = eventClick.scan(0, plus);
fatherWalletValue = sonWalletValue
    .map(function (value) {
        return value + 10
    });
sonWalletValue.assign($sonWallet, "val");
```



# Advancing the User Interface

## *Reducing the cognitive load*



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

