
SOUNDCLOUD

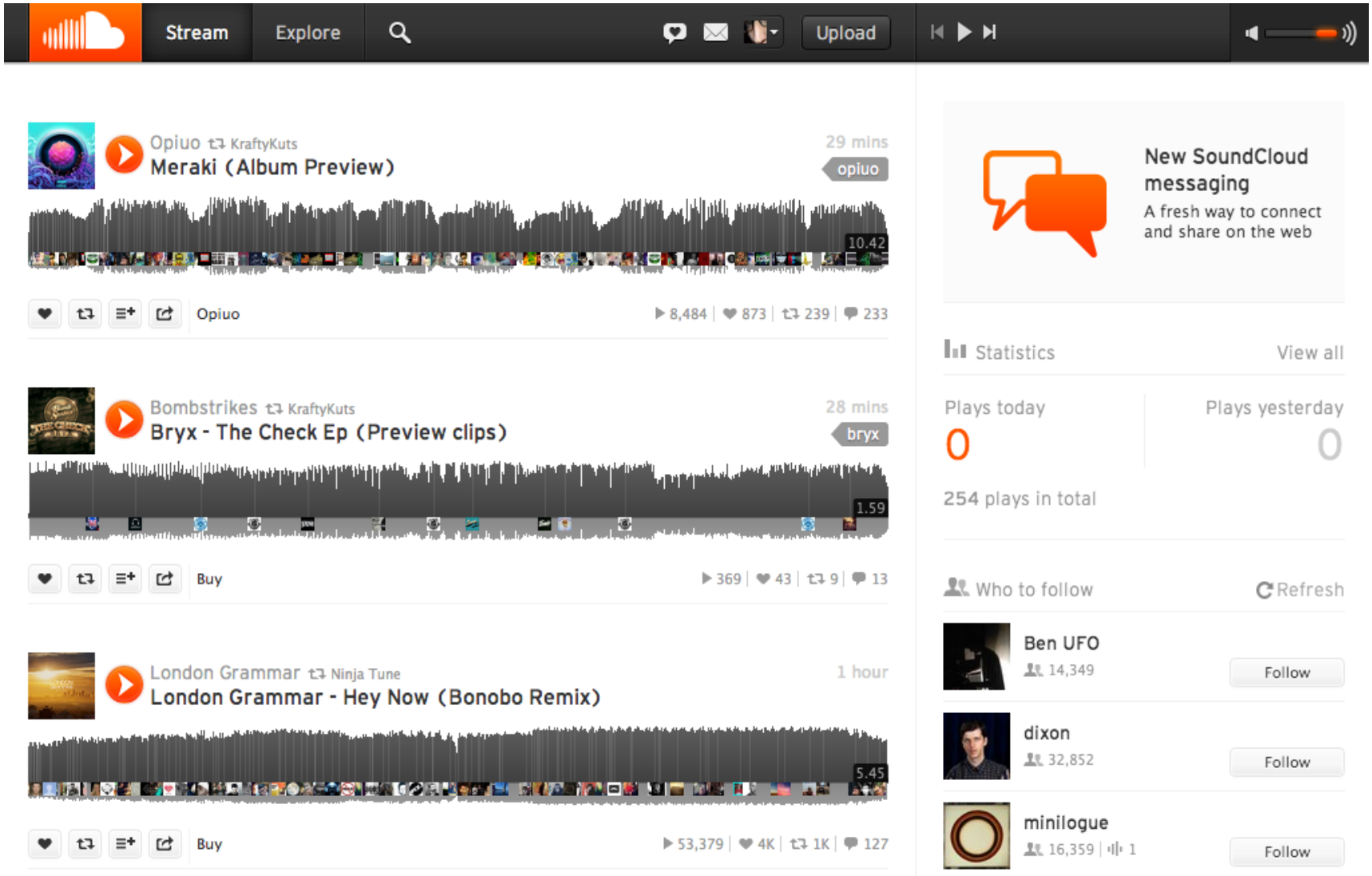
How We Built New SoundCloud Stats Using D3

Márton Salomváry, Engineer

Story of a Real Life Project

- Learning D3
- D3 ❤ Backbone
- Structuring code
- Taming browsers

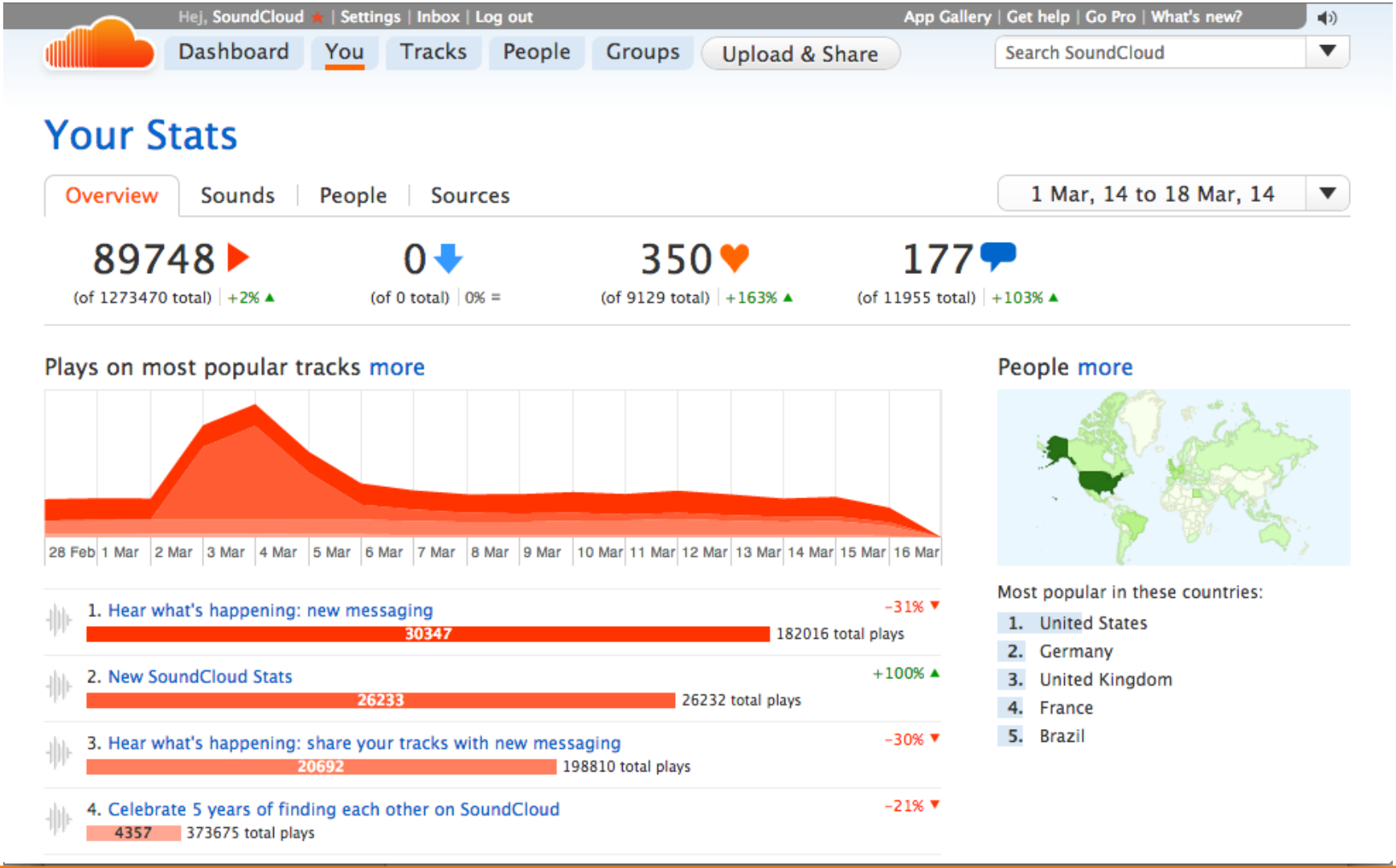
What is SoundCloud?



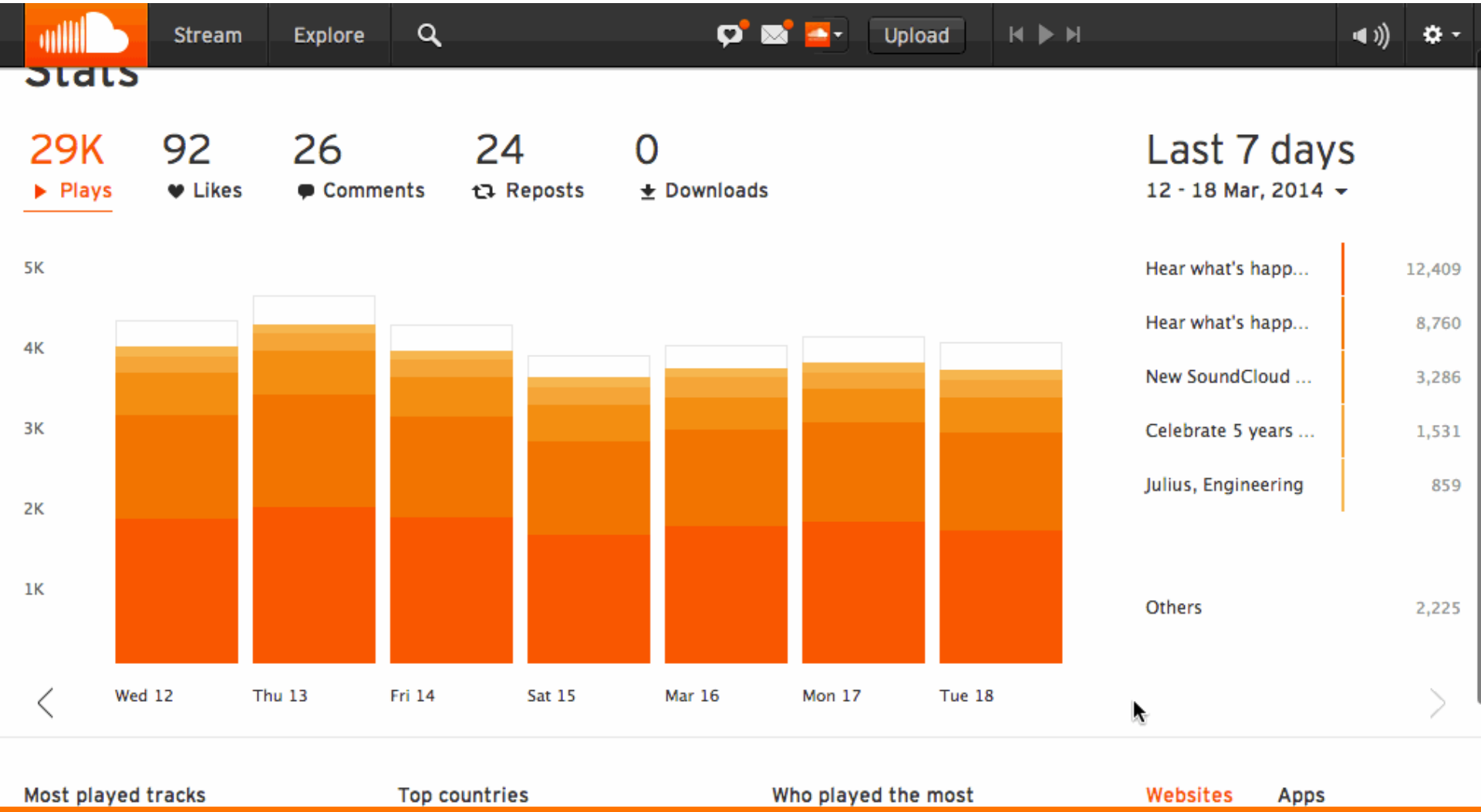
The Project

“Provide statistics to creators that help them understand the performance of their sounds.”

Old Stats


Hej, SoundCloud | Settings | Inbox | Log out
App Gallery | Get help | Go Pro | What's new?
Dashboard
You
Tracks
People
Groups
Upload & Share
Search SoundCloud
Your Stats
Overview
Sounds
People
Sources
1 Mar, 14 to 18 Mar, 14
89748
(of 1273470 total) +2%
0
(of 0 total) 0% =
350
(of 9129 total) +163%
177
(of 11955 total) +103%
Plays on most popular tracks more
28 Feb 1 Mar 2 Mar 3 Mar 4 Mar 5 Mar 6 Mar 7 Mar 8 Mar 9 Mar 10 Mar 11 Mar 12 Mar 13 Mar 14 Mar 15 Mar 16 Mar
1. Hear what's happening: new messaging
-31%
30347
182016 total plays
2. New SoundCloud Stats
+100%
26233
26232 total plays
3. Hear what's happening: share your tracks with new messaging
-30%
20692
198810 total plays
4. Celebrate 5 years of finding each other on SoundCloud
-21%
4357
373675 total plays
People more
Most popular in these countries:
1. United States
2. Germany
3. United Kingdom
4. France
5. Brazil

New Stats


Stream
Explore
Upload
Stats
29K
Plays
92
Likes
26
Comments
24
Reposts
0
Downloads
Last 7 days
12 - 18 Mar, 2014
5K
4K
3K
2K
1K
Wed 12
Thu 13
Fri 14
Sat 15
Mar 16
Mon 17
Tue 18
Hear what's happ...
12,409
Hear what's happ...
8,760
New SoundCloud ...
3,286
Celebrate 5 years ...
1,531
Julius, Engineering
859
Others
2,225
Most played tracks
Top countries
Who played the most
Websites
Apps
SOUNDCLOUD

Stats Stats

- Part of a 300 KLOC Backbone app
  - 2 REST services
- One BFF (Backend For Frontends)
- 70 modules for Stats (js, css, html)
  - 1200 LOC D3 code
  - in 10 modules

CHALLENGE #1

Learn D3

Learn D3

- Excellent tutorials and examples
- Understand enter/update/exit

Everything is just a configurable function.

Configurable Functions

```
stack = d3.layout.stack()
    .offset("wiggle")
    .values(function(d) {
        return d.values;
    });
stackedData = stack(data);
```

CHALLENGE #2

D3 ❤ Backbone

Separation of Concerns

- Backbone.View to wire up everything
- Treat D3 code as a template

```
var barChart = require('bar-chart');
// ...

View.extend({
    tagName: 'svg',
    // ...
    render: function() {
        var chart = barChart()
            .width(this.$el.width())
            .height(this.$el.width())
            .otherOption(this.options.otherOption);

        chart(this.el, this.getTemplateData());
    }
});
```

```
View.extend({
    // ...

    attributes: { width: '100%', height: '600' },

    initialize: function() {
        $(window).on('resize', this.render, this);
    },

    remove: function() {
        $(window).off('resize', this.render, this);
        View.prototype.remove.apply(this, arguments);
    }
});
```

CHALLENGE #3

Structuring Code

Choose a Module Format

- We author CommonJS
- Generate AMD on the fly

Choose whatever format you like and stick with it!

Write D3 Code as Configurable Functions

“To sum up: implement charts as closures with getter-setter methods.“

Towards Reusable Charts by Mike Bostock

```
var d3 = require('d3'),
    someDependency = require('some/dependency');

module.exports = function myChart() {
    var width;

    function chart(selection, data) {
        // do enter/update/exit on selection using width here
    }

    chart.width = function(value) {
        width  = value;
        return chart;
    };

    // implement other getters/setters here

    return chart;
};

function privateStuff() { // ... }
```

```
var d3 = require('d3'),
    someDependency = require('some/dependency');

module.exports = function myChart() {
    var width;

    function chart(selection, data) {
        // do enter/update/exit on selection using width here
    }

    chart.width = function(value) {
        width  = value;
        return chart;
    };

    // implement other getters/setters here

    return chart;
};

function privateStuff() { // ... }
```

```
var myChart = require('my-chart');

var chart = myChart()
        .width(500)
        .height(300);

chart(selection, data);
```

THAT WAS

EASY

but...

Complexity Will Grow

“Layer” Modules

- One “main” module
  - Point of entry
  - Creates and configures layers

- Layers are isolated modules
  - Single responsibility
  - Reusable
  - Testable

BTW Testing...

Test in isolation

Assert DOM

Do integration tests

```
before(function() {
        svg = $('<svg>');
        chart = myChart()
                .width(100)
                .height(300);
});

test('draws bars', function() {
         var data = [ { date: '2012-12-12', bananas: 124 },  // ... ];
         chart(svg, data);

         assertEqual(svg.find('rect').length, 2, 'rendered two bars');
         assertEqual(svg.find('rect:eq(1)').attr('x'), '50', 'x was set');
});
```

CHALLENGE #4

Browsers Out There

SVG is not a new thing...

...except in browsers.

SVG support in modern browsers is pretty good*.

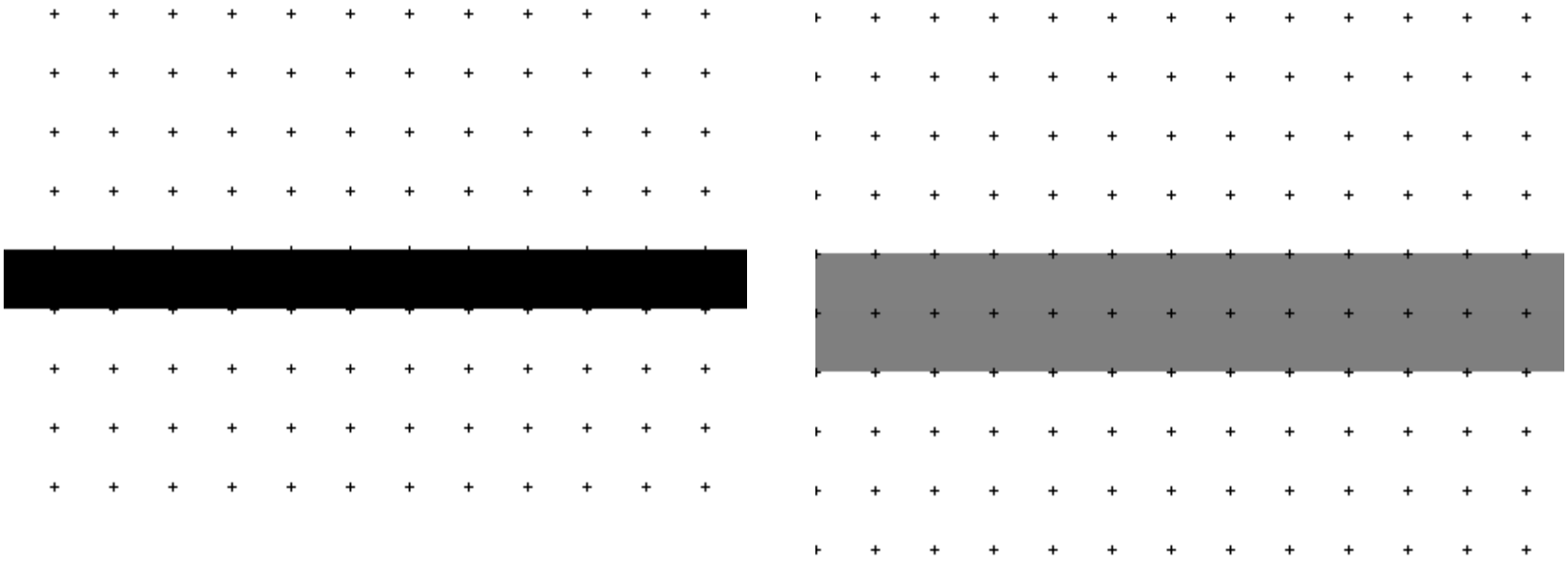
Except for a few corner cases…

(Check caniuse.com before you start.)

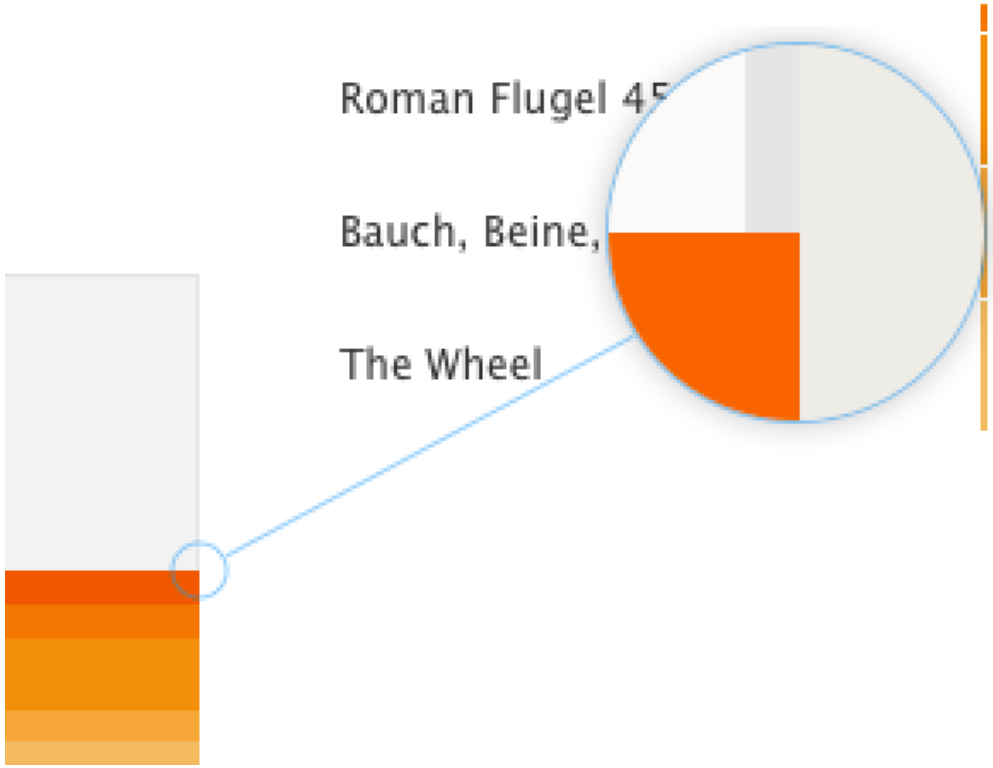
Drawing Sharp Lines

```
<line
    stroke-width="1"
    stroke="black"
     x1="10"
     x2="20"
     y1="10"
     y2="10"/>
```

Drawing Sharp Lines



Why Care?



Solution

```
<line
    stroke-width="1"
    stroke="black"
     x1="10"
     x2="20"
     y1="10.5"
     y2="10.5"/>
```

The bad news is...

Half pixel trick doesn't work in IE, FF needs shape-rendering: crispEdges.

But shape-rendering: crispEdges might not be what you want.

Other Quirks?

- **CSS transitions vs. D3 transitions**

Summary

- Separate D3 and Backbone
- Limit complexity (hint: JSHint)
  - Modules
  - Refactor, refactor, refactor
- Testing is easy
- Learn about “sharpness”

Thank you

Márton Salomváry

@salomvary / salomvary@soundcloud.com

soundcloud.com/jobs

Opensource?

- Backbone integration is very specific to our application
- Maybe “composing charts as layers”