

# Design systems and web components

A scalable and accessible foundation

Erik Kroes

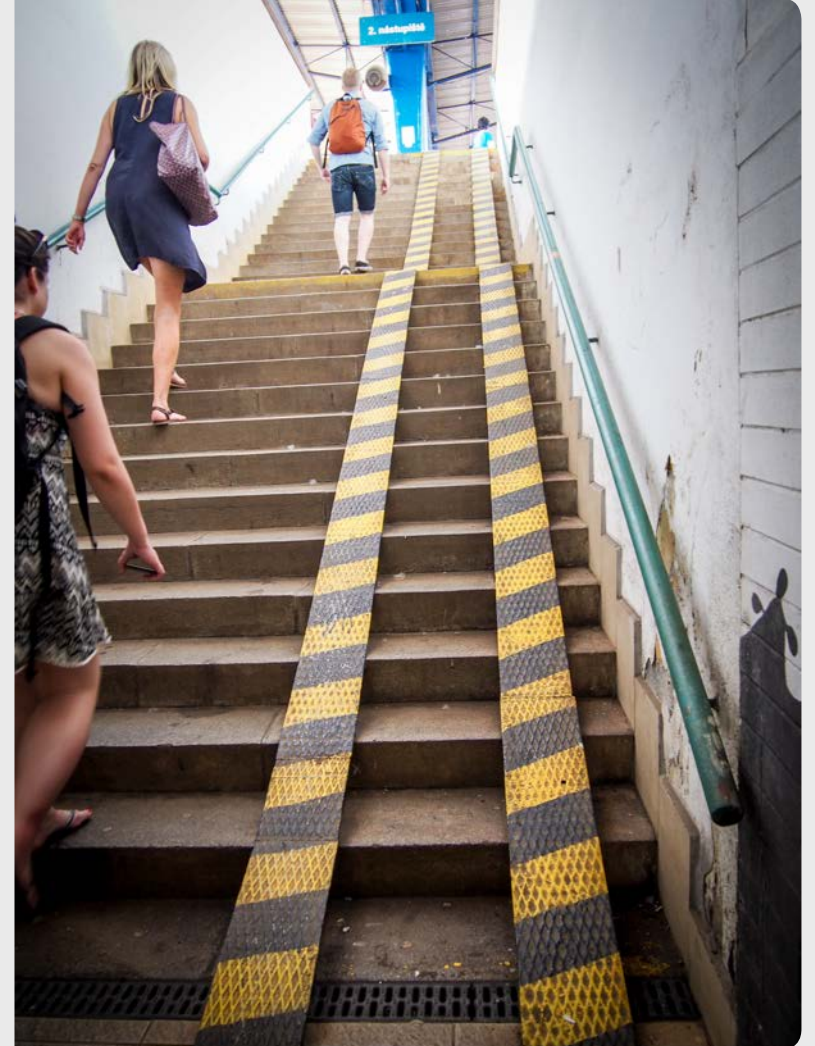
Driebergen • 16 May 2019

# Hi, I'm Erik Kroes

- Photographer
- Visual Artist
- Creative Coder
- Accessibility Specialist
- Front-end developer
- Fabric store owner

## The story today

1. The part where I am really enthusiastic and hope you are too
2. The part where I tell you it's not that easy
3. The part where I make you happy again



**The part where I am really enthusiastic  
and hope you are too**

# What is a Design System?

*“[The] ingredients [that] come together to tell the story of how your organization designs and builds products”*

**- Brad Frost**

# What is in a design system?

**In general it will have some of these elements**

- Corporate identity (colors, fonts, images, etc)
- Design principles
- Guidelines (Development and UX)
- Style guide
- Design library
- Coded components
- Component guidelines
- Way of working

# I Design Systems

- Separation of concerns
- Quality
- Speed
- Unified User Experience
- Efficiency, Create once use many

# What are Web Components

## Web Components are a set of standards

- Custom Elements: A way to create your own HTML elements
- ES Modules: Self-contained modules of JavaScript code that can be reused and extended
- HTML Templates: Reusable fragments of DOM
- Shadow DOM: Encapsulation of DOM

And when their powers combine...





# What are Web Components?

*“Reusable extendable encapsulated self-contained components for the web.”*

## A practical example

```
<google-map fit-to-markers api-key="AIzaSyD3E1D9b-Z7ekrT3tbhl_dy8DCXuIuDDRc">  
  <google-map-marker latitude="37.78" longitude="-122.4"  
    draggable="true"></google-map-marker>  
</google-map>
```

<https://github.com/GoogleWebComponents/google-map>

## A not so practical example

```
<spacer-gif height="1" width="1"></spacer-gif>
```

<https://github.com/erikkroes/spacer-gif>

# I 🦁 Web Components + Design Systems

- Reusable: Create once, use anywhere
- Extendable: Create variations
- Encapsulated and self-contained: Consistent, and controllable
- Support
- Flexible

**The part where I tell you it's not that  
easy**

What isn't used is a waste

*“Write guidelines, not books”*



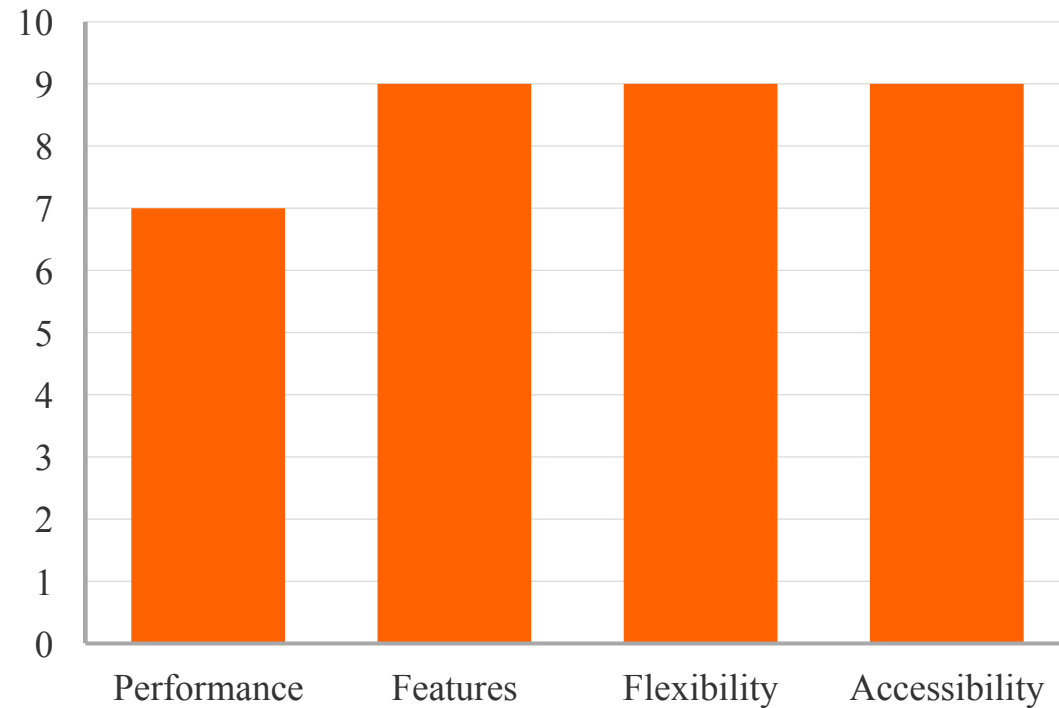
# One point of truth

- One point of failure
- One point of blame

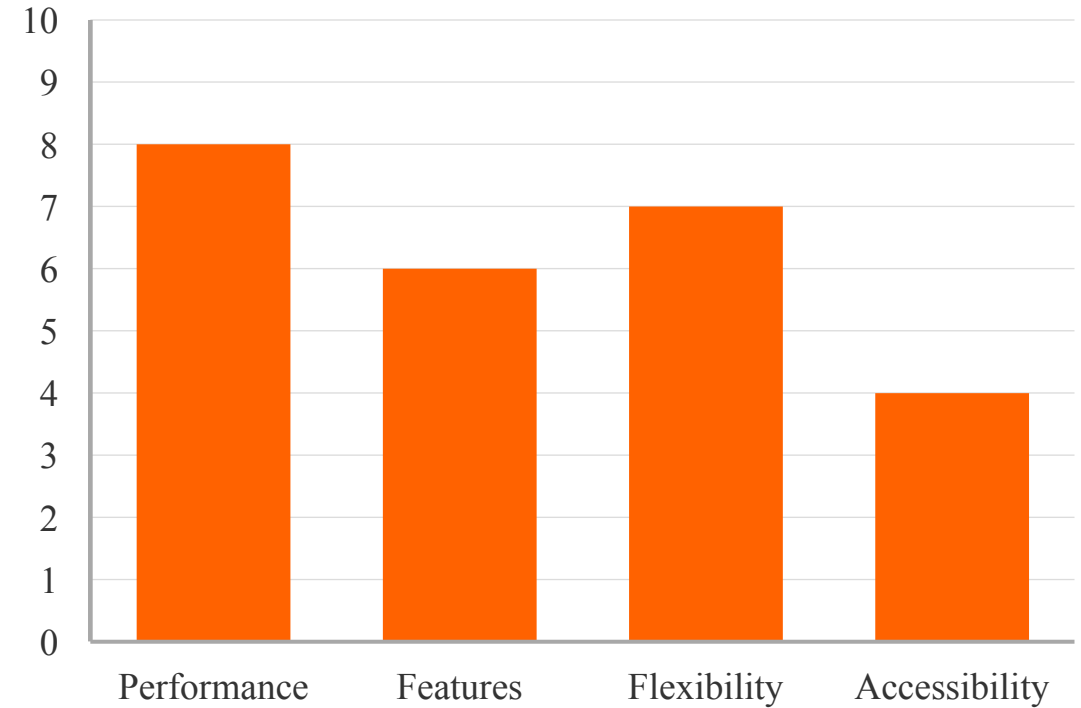


# Quality and contributions

Old component



New component



# Freedom

*“As much as need but as little as you can”*

# Making accessible UI components is hard





# Making accessible UI components is really really hard



# Making accessible UI components is really really hard

## Know this by heart

- Focusable
- Visible states (focus, active, hover)
- Functional states
- Keyboard interaction
- Semantics, role
- Accessible name
- Communicate state and properties
- Relationships
- Global standards/conventions
- Circumvent browser bugs and variations

If you'd ask me...

*“Everybody should use Web Components  
Few people should build them”*

**The part where I make you happy  
again**

# Web Components + Accessibility = 🦁

**Lion is a set of open source web components by ING**

- Performant
- Flexible
- Accessible
- White label



## What is included in 🦁?

- A form system with validation, localization and all kinds of form elements
- Icon system
- Overlay system with a dialog
- Steps system
- Ajax
- Calendar and datepicker

## A lion example

```
<lion-input-datepicker label="Date" .modelValue=${new Date('2017/06/15')}>  
</lion-input-datepicker>
```

<https://github.com/ing-bank/lion>

# Thank you

**A scalable and accessible foundation with Web Components is priceless  
but you can get it for free.**

- <https://github.com/ing-bank/lion>
- <https://lion-web-components.netlify.com>
- <https://twitter.com/erikkroes>