

Web

Components
Polymer

canoo

Hendrik Ebbers



- Lead of JUG Dortmund
- Speaker, blogger & author
- Java Dev @ Canoo Engineering AG



GuiGarage
www.guigarage.com



Dolphin Platform

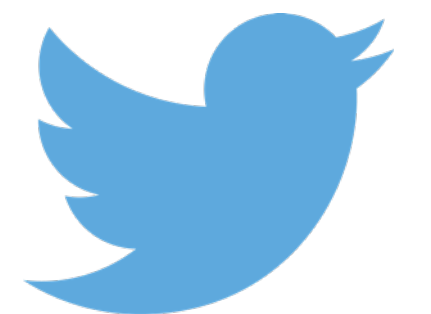


@hendrikEbbers

Michael Heinrichs

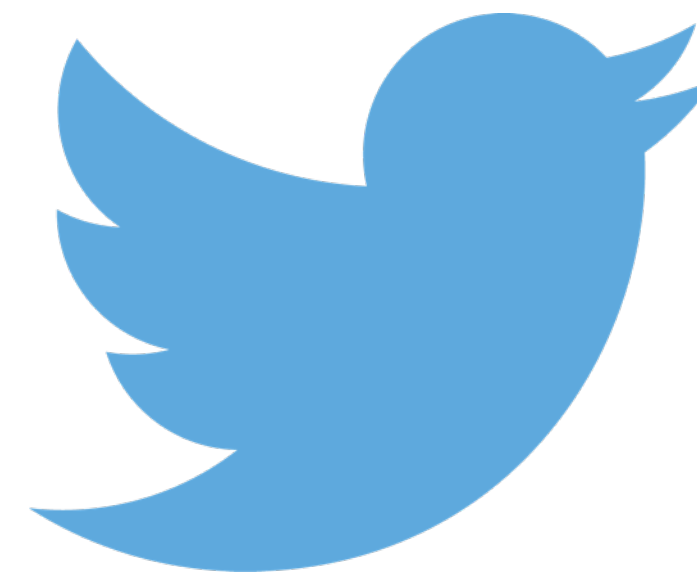
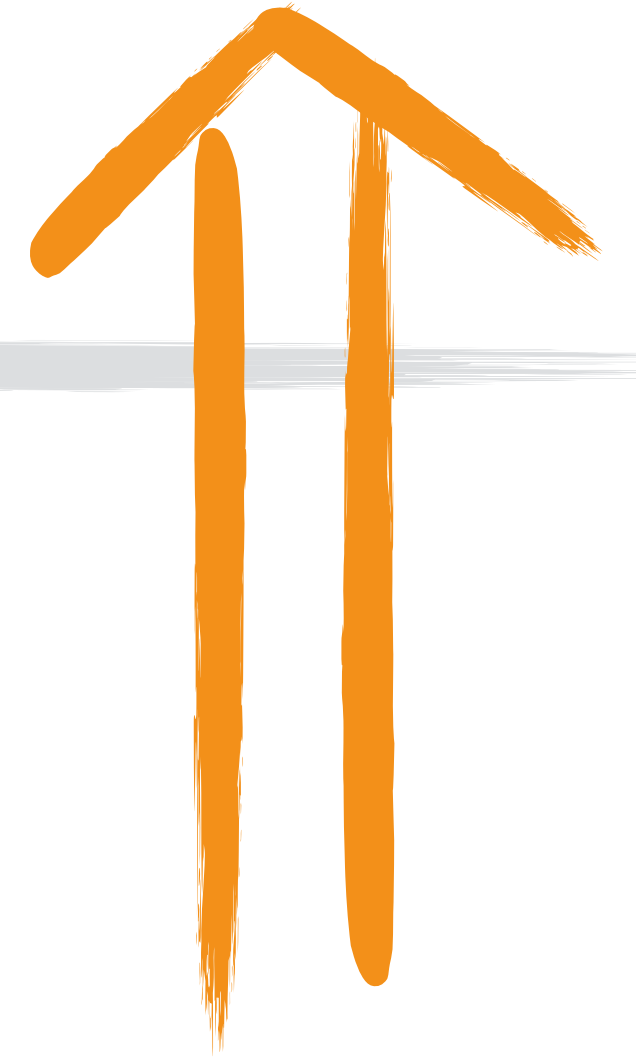


- UI magician
- Active speaker and writer
- Code Monkey @ Canoo Engineering AG
- <http://blog.netopyr.com>



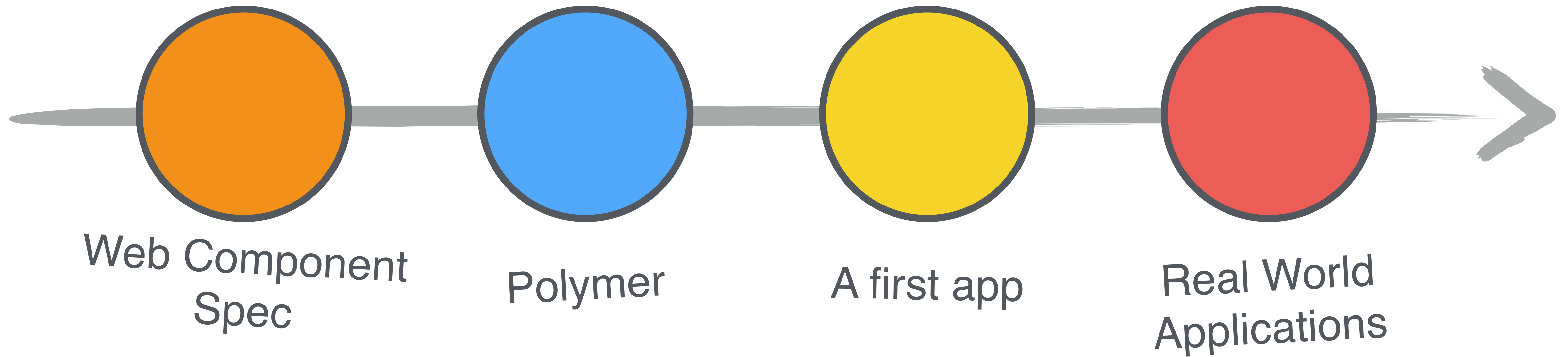
@net0pyr

Questions?



#JFokusWCU

Content



What to Watch

BEST OF YOUTUBE

- Popular on YouTube
- Music
- Sports
- Gaming
- Movies
- News

Browse channels

Sign in now to see your channels and recommendations!

Sign In

What to Watch

Music

Choose your language.

You're viewing YouTube in English (US). You can change this preference below.

Learn more



One of those days 2 - Candide Thovex

by Candide Thovex
12,168,667 views • 1 week ago



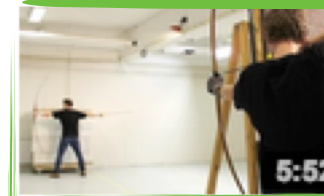
Dramatic footage filmed by woman who thought she was watching police arrest a woman... only to real

by Carrie Allison
47,948 views • 6 days ago



'Bart' The 'Zombie Cat' Claws Its Way Back From The Dead

by lugamedia
152,464 views • 1 day ago



Lars Andersen: a new level of archery

by larsandersen23
22,026,712 views • 6 days ago

Popular on YouTube - Switzerland Recommended channel

Subscribe 128,657



UNDRESS ME

by Tatia Pllieva
19,864,541 views • 6 months ago



Mutant Giant Spider Dog (SA Wardega)

by SA Wardega
129,461,366 views • 4 months ago



Selfie everyday for 7 years
12 yrs old to 19 yrs old!...

by Hugo Cornellier
5,350,552 views • 6 months ago



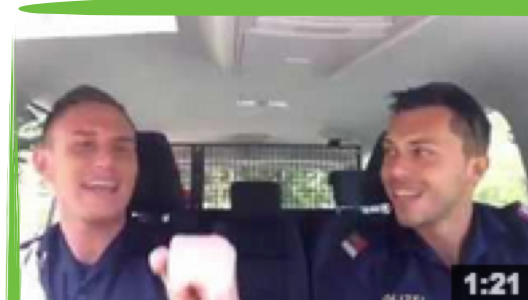
Phone 6 Plus Bend Test

by Unbox Therapy
53,704,941 views • 4 months ago



[FULL HD] UTair Aviation
767-300 NEAR MISS? GO...

by Barcelona-El Prat In'tl
28,776,077 views • 6 months ago

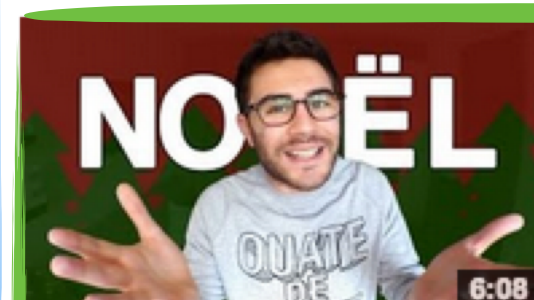


Polizisten hören Helene Fischer's 'Atemlos' im...

by Spass MussSein
3,542,577 views • 6 months ago

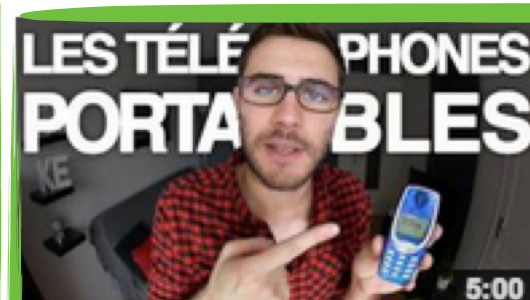
Cyprien Recommended channel

Subscribe 6,714,670



Cyprien - Noël

by Cyprien
13,862,593 views • 1 year ago



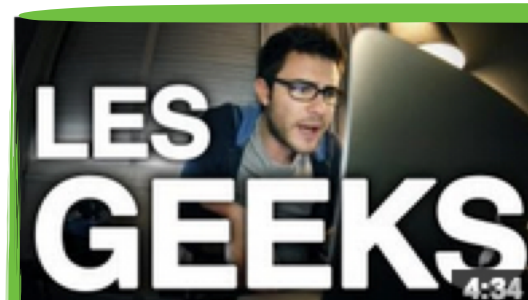
LES TÉLÉPHONES PORTABLES

by Cyprien
15,552,898 views • 3 years ago



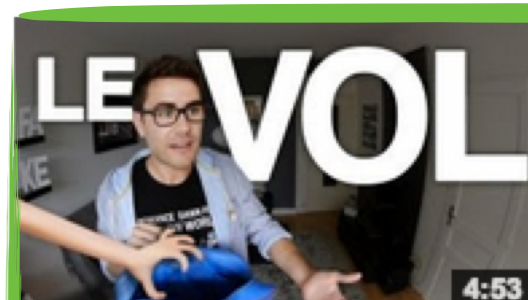
Cyprien - L'école

by Cyprien
20,670,324 views • 1 year ago



Cyprien - Les geeks

by Cyprien
18,941,535 views • 2 years ago



Cyprien - Le vol

by Cyprien
15,145,986 views • 2 years ago



Cyprien - Les maniaques de l'hygiène

by Cyprien
14,367,822 views • 1 year ago

shakiraVEVO Recommended channel

Subscribe 6,191,082

frontend code today

```
<li class="yt-shelf-grid-item yt-uix-shelfslider-item">
  <div class="yt-lockup yt-lockup-grid yt-lockup-video vve-check clearfix"
    data-context-item-id="naiLVvuPCAw"
    data-visibility-tracking=
      "CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0HkCMkLzc7-qi1J0B">
    <div class="yt-lockup-dismissable">
      <div class="yt-lockup-thumbnail contains-addto">
        <a aria-hidden="true" href="/watch?v=naiLVvuPCAw"
          class=" yt-uix-sessionlink      spf-link "
          data-sessionlink=
            "itct=CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0HjIKZy1oaWdoLXJjaA">
        <div class="yt-thumb video-thumb">
          
        </div>
      </div>
    </div>
  </div>
```

Web Applications
Today

frontend code today



iPhone 6 Plus Bend Test

by Unbox Therapy ✓

63,704,941 views • 4 months ago

Web Applications Today

```
<li class=" yt-ui-shelfslider-item">
  <div class="yt-lockup yt-lockup-grid yt-lockup-video vve-check clearfix" data-context-item-id="naiLVvuPCAw"
    data-visibility-tracking="CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0HkCMkLzc7-qilJ0B">
    <div class="yt-lockup-dismissable">
      <div class="yt-lockup-thumbnail contains-addto">
        <a aria-hidden="true" href="/watch?v=naiLVvuPCAw" class=" yt-ui-sessionlink   spf-link "
          data-sessionlink="itct=CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0HjIKZy1oaWdoLXJjaA">
          <div class="yt-thumb video-thumb">
            
          </div>
          <span class="video-time" aria-hidden="true">1:21</span>
        </a>
        <span class="thumb-menu dark-overflow-action-menu video-actions">
          <button onclick=";return false;"
            class="yt-ui-button-reverse flip addto-watch-queue-menu spf-nolink hide-until-delayloaded yt-ui-button yt-ui-button-dark-overflow-action-menu yt-ui-button-size-default yt-ui-button-has-icon no-icon-markup yt-ui-button-empty"
            aria-expanded="false" aria-haspopup="true" type="button">
          <span class="yt-ui-button-arrow yt-sprite"></span>
          <ul class="watch-queue-thumb-menu yt-ui-button-menu yt-ui-button-menu-dark-overflow-action-menu" style="display: none;">
            <li role="menuitem" class="overflow-menu-choice addto-watch-queue-menu-choice addto-watch-queue-play-next yt-ui-button-menu-item"
              data-action="play-next" onclick=";return false;" data-video-ids="naiLVvuPCAw">
              <span class="addto-watch-queue-menu-text">Play next</span>
            </li>
            <li role="menuitem" class="overflow-menu-choice addto-watch-queue-menu-choice addto-watch-queue-play-now yt-ui-button-menu-item"
              data-action="play-now" onclick=";return false;" data-video-ids="naiLVvuPCAw">
              <span class="addto-watch-queue-menu-text">Play now</span>
            </li>
          </ul>
        </span>
        <button class="yt-ui-button yt-ui-button-size-small yt-ui-button-default yt-ui-button-empty yt-ui-button-has-icon no-icon-markup addto-button video-actions spf-nolink hide-until-delayloaded addto-watch-later-button-sign-in yt-ui-tooltip"
          type="button" onclick=";return false;" title="Watch Later" role="button"
          data-video-ids="naiLVvuPCAw" data-button-menu-id="shared-addto-watch-later-login"><span
            class="yt-ui-button-arrow yt-sprite"></span></button>
        <button class="yt-ui-button yt-ui-button-size-small yt-ui-button-default yt-ui-button-empty yt-ui-button-has-icon no-icon-markup addto-button addto-queue-button video-actions spf-nolink hide-until-delayloaded addto-tv-queue-button yt-ui-
          tooltip"
          type="button" onclick=";return false;" title="TV Queue" data-video-ids="naiLVvuPCAw"
          data-style="tv-queue"></button>
      </div>
    <div class="yt-lockup-content">
      <h3 class="yt-lockup-title">
        <a href="/watch?v=naiLVvuPCAw"
          class=" yt-ui-ellipsis yt-ui-ellipsis-2 yt-ui-sessionlink   spf-link "
          data-sessionlink="itct=CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0HjIKZy1oaWdoLXJjaA"
          title="Polizisten hören Helene Fischer&#39;s &#39;Atemlos&#39; im Polizeiauto"
          aria-describedby="description-id-439757"
          dir="ltr">Polizisten hören Helene Fischer&#39;s&#39;Atemlos&#39; im Polizeiauto</a>
        <span class="accessible-description" id="description-id-439757"> - Duration: 1:21.</span>
      </h3>
      <div class="yt-lockup-byline">by <a href="/user/djgreyhair class=" yt-ui-sessionlink   spf-link g-hovercard" data-name=""
        data-sessionlink="itct=CFEQpDAYBSITCM70r_3JucMCFY6yHAodHk0ANii0Hg" data-ytid="UCCBrSuWhYxpWZYSTY7kkB4A">Spass MussSein</a>
      </div>
      <div class="yt-lockup-meta">
        <ul class="yt-lockup-meta-info">
          <li>3,542,577 views</li>
          <li>6 months ago</li>
        </ul>
      </div>
    </div>
  </div>
</li>
```

frontend code tomorrow

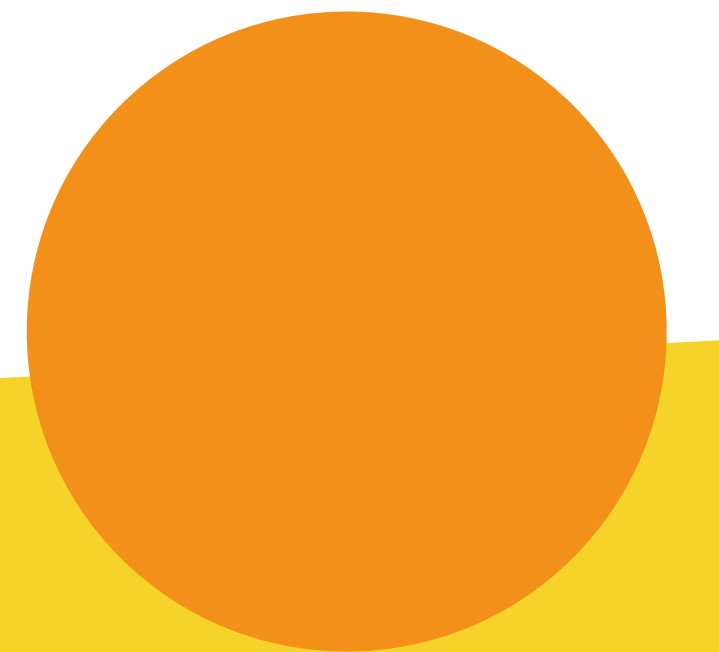
```
<shelf
  title="Popular on YouTube - Switzerland"
  subscribers="128,657">

  <shelf-grid-item
    title="iPhone 6 Plus Bend Test"
    url="https://www.youtube.com/watch?v=znK652H6yQM"
    thumbnail="https://i.ytimg.com/vi_webp/znK652H6yQM/mqdefault.webp"
    user="Unbox Therapy"
    userUrl="https://www.youtube.com/user/unboxtherapy"
    views="63,732,280"
    time="4 months ago">
```

...

Web Applications
Tomorrow

Web Component Spec



Web Component Spec

- Specified by W3C
- Current state of spec can be found online:
http://www.w3.org/standards/techs/components#w3c_all

Web Component Spec

- Divided in 4 parts:
 - HTML Templates
 - Shadow DOM
 - Custom Elements
 - HTML Imports

<template>

<template>

Activities



And **Michael** did that again.

Seconds ago



Hendrick did this again.

Minutes ago



Michael did that.

1 hour ago



Hendrick did this.

Yesterday

<template>

```
<div class="activity-stream">
  <h2>Activities</h2>
  <div class="activity">
    
    <div class="time">Seconds ago</div>
    <div class="content"><a>Michael</a> had fun coding.</div>
  </div>
  <div class="activity">
    
    <div class="time">Minutes ago</div>
    <div class="content"><a>Hendrik</a> did this again.</div>
  </div>
  ...
</div>
```


<template>

```
<template id="activity-template">
  <div>
    <img class="icon" src="" width="40" height="40">
    <div class="time"></div>
    <div class="content"></div>
  </div>
</template>
```

template tag

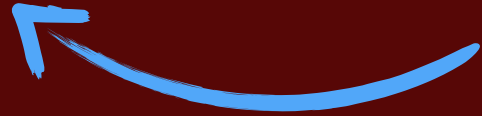
copy boilerplate

```
<div class="activity">
  
  <div class="time">Seconds ago</div>
  <div class="content"><a>Michael</a> had fun coding.</div>
</div>
```

<template>

```
<template id="activity-template">
  <div>
    <img class="icon" src="" width="40" height="40">
    <div class="time"></div>
    <div class="content"></div>
  </div>
</template>
```

```
var template = document.querySelector('#activity-template');
var clone = document.importNode(template.content, true);
document.body.appendChild(clone);
```

 use content property

<template>

```
<template id="activity-template">
  <div>
    
    <div class="time">{{item.time}}</div>
    <div class="content">{{item.content}}</div>
  </div>
</template>
```

No Data Binding

<template>



↑
22+



↑
26+ and
Android 4.4+



↑
7.1+



↑
13+

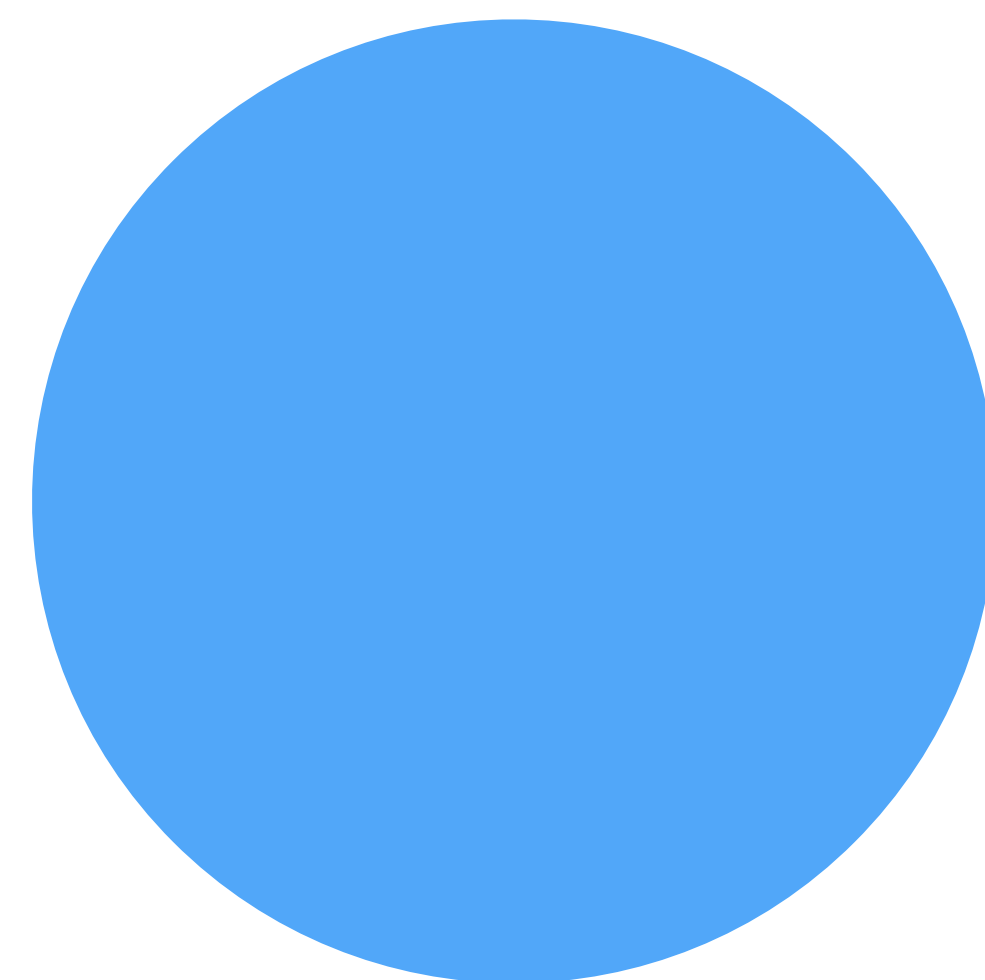
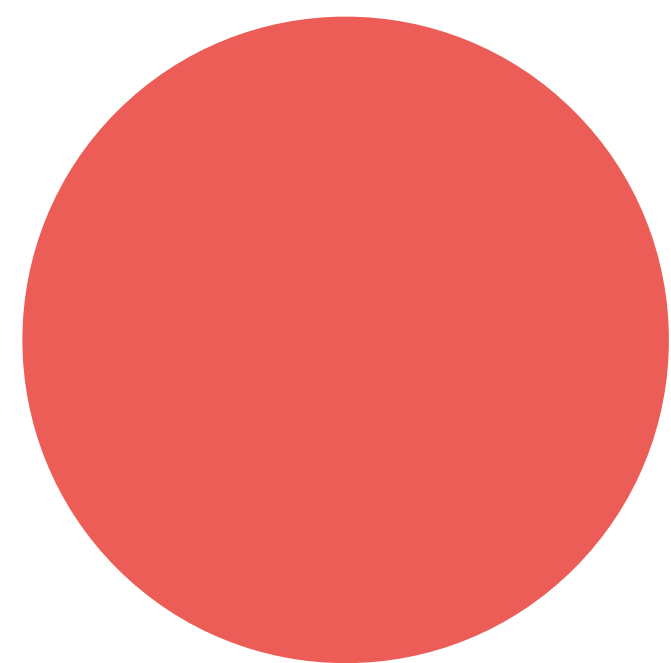


↑
15+

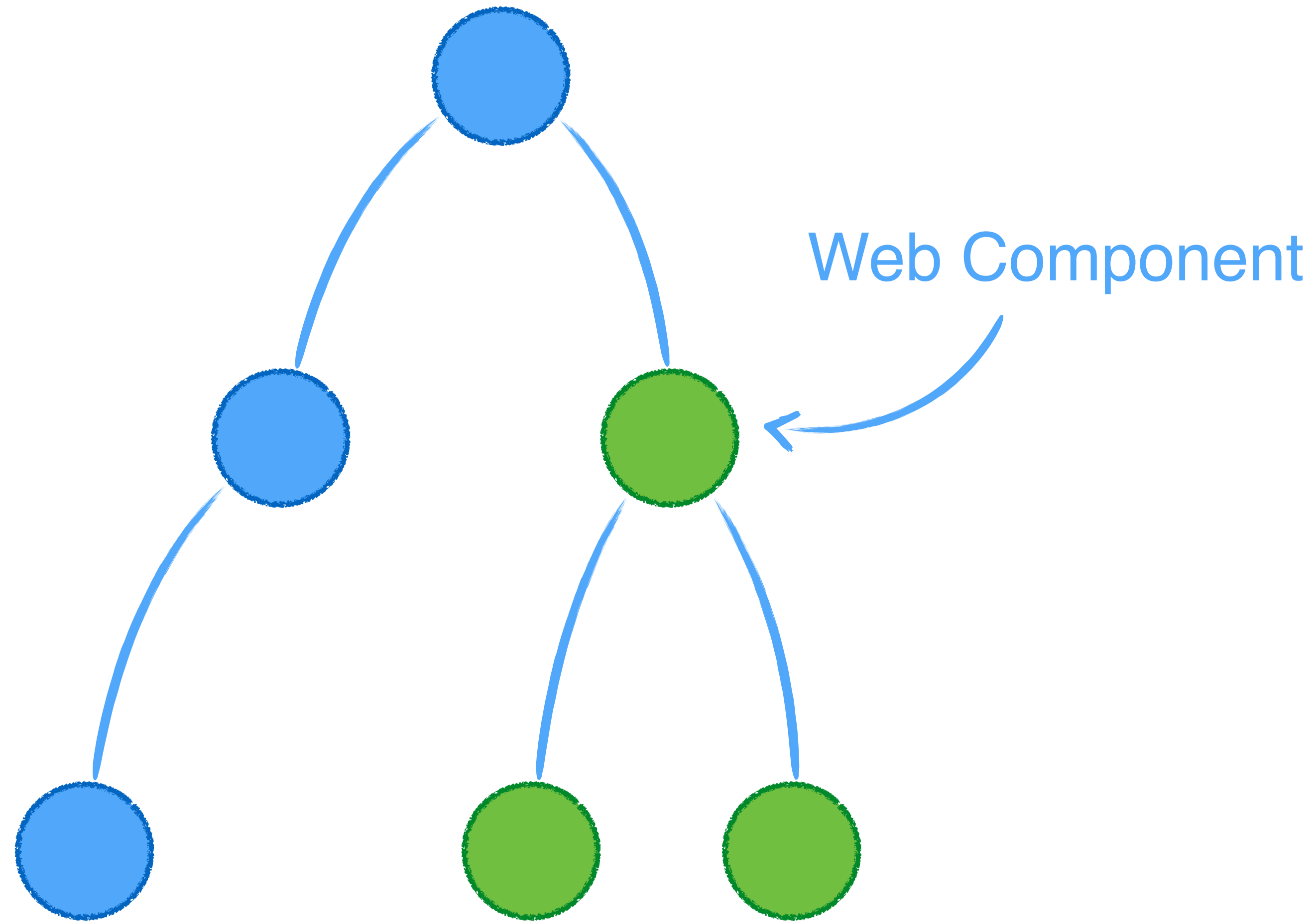


Shadow

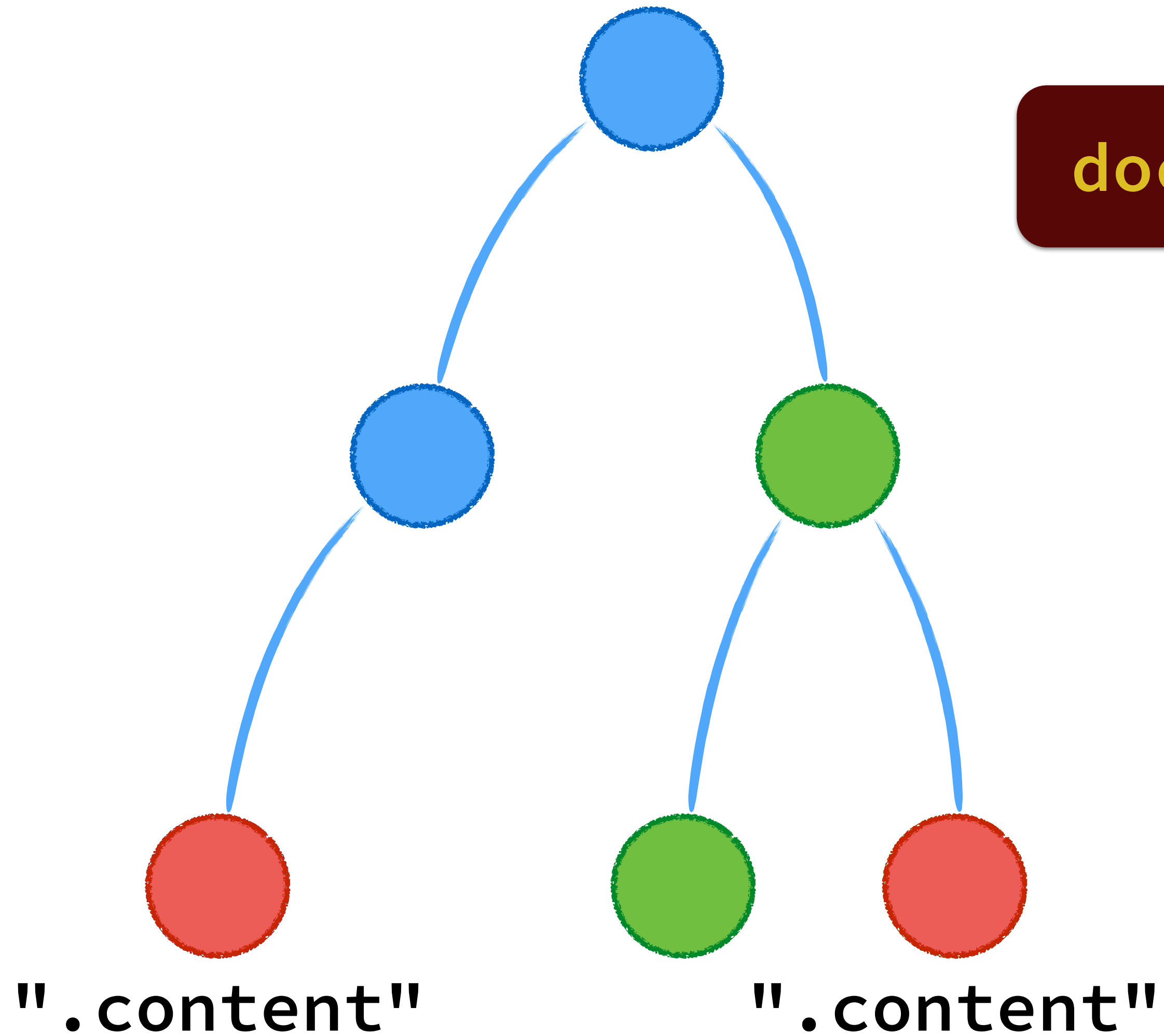
Dom



Shadow Dom



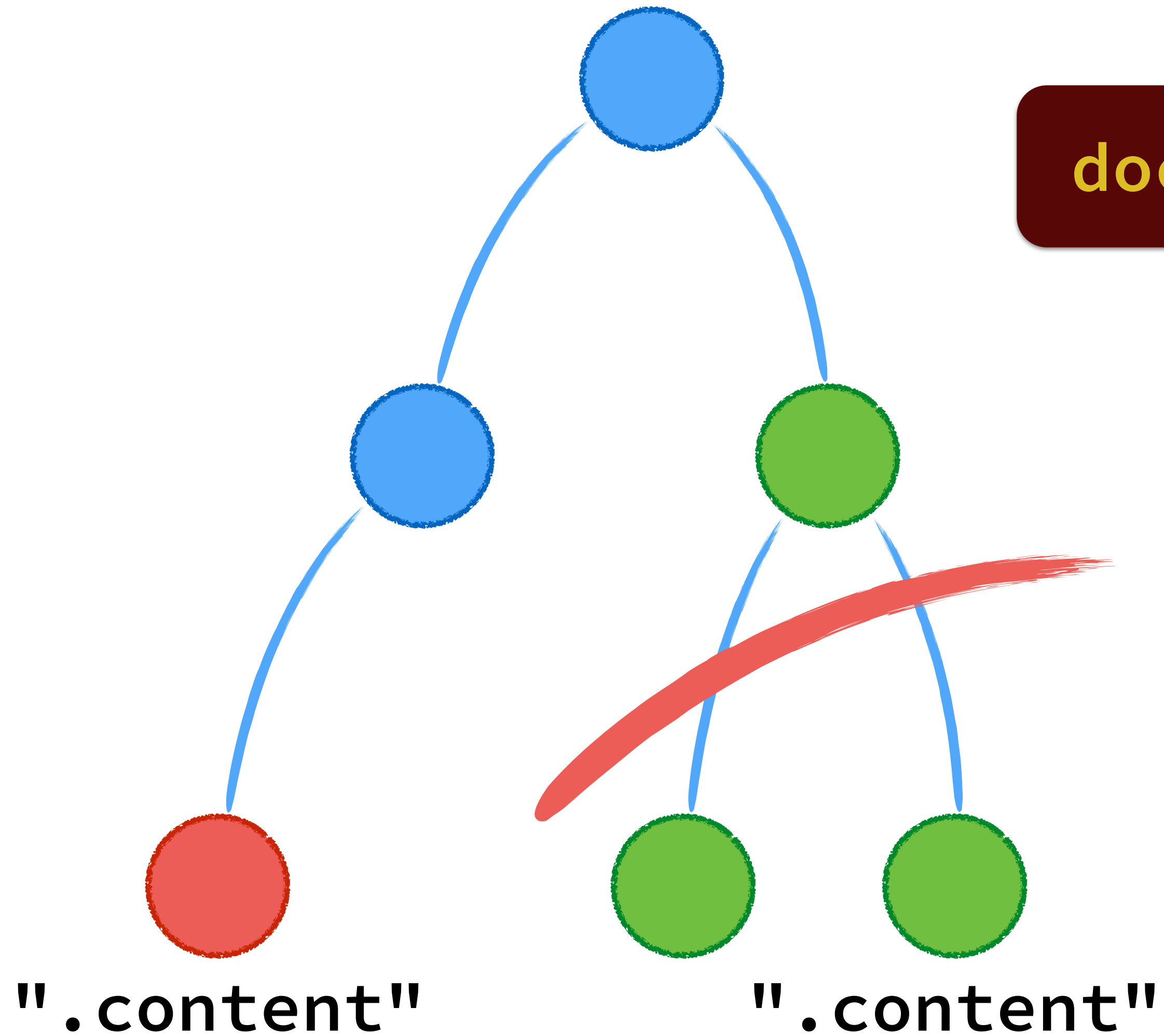
Shadow Dom



```
document.querySelector(".content")
```

```
.content {  
  color: blue;  
}
```

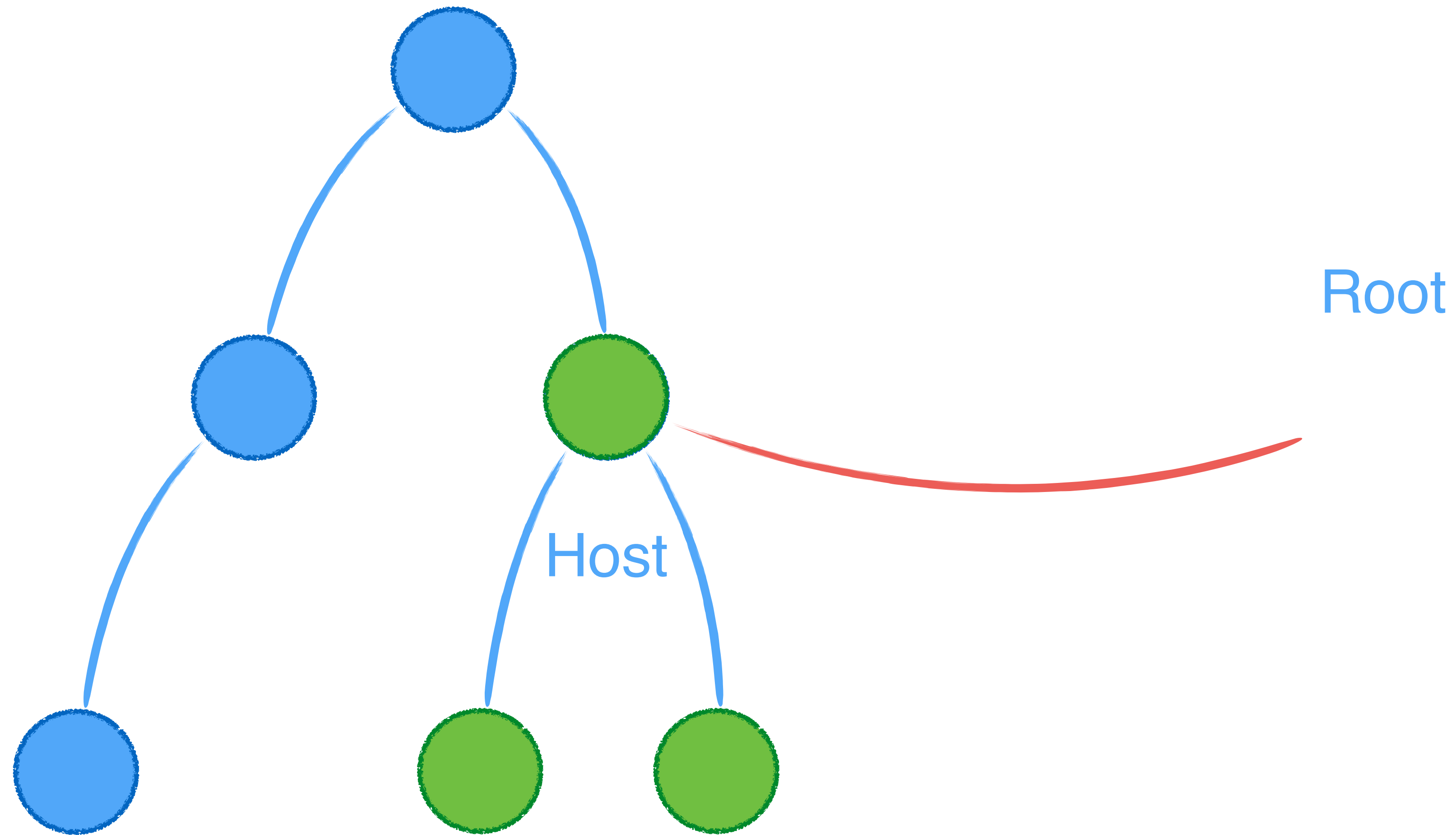
Shadow Dom



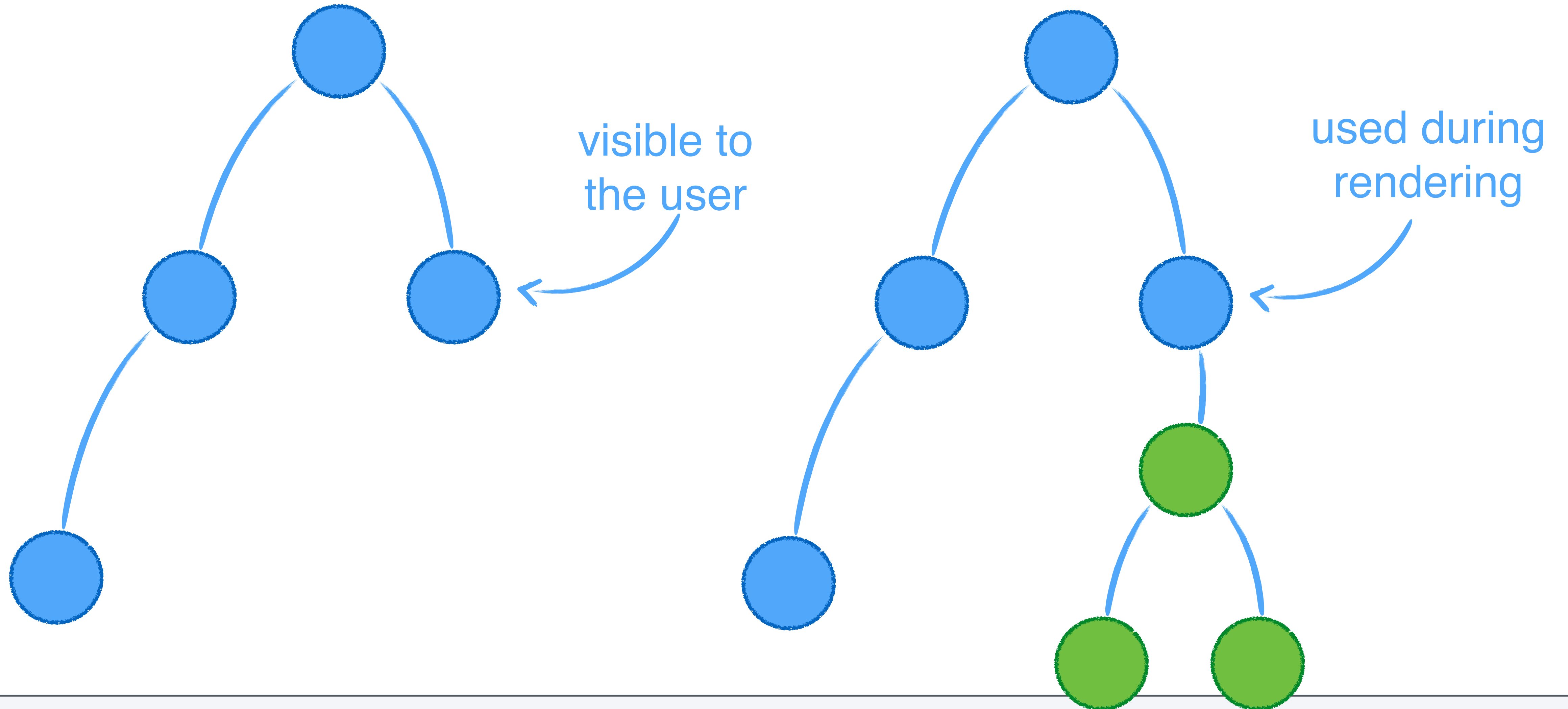
```
document.querySelector(".content")
```

```
.content {  
  color: blue;  
}
```


Shadow Dom



Shadow Dom



Shadow Dom

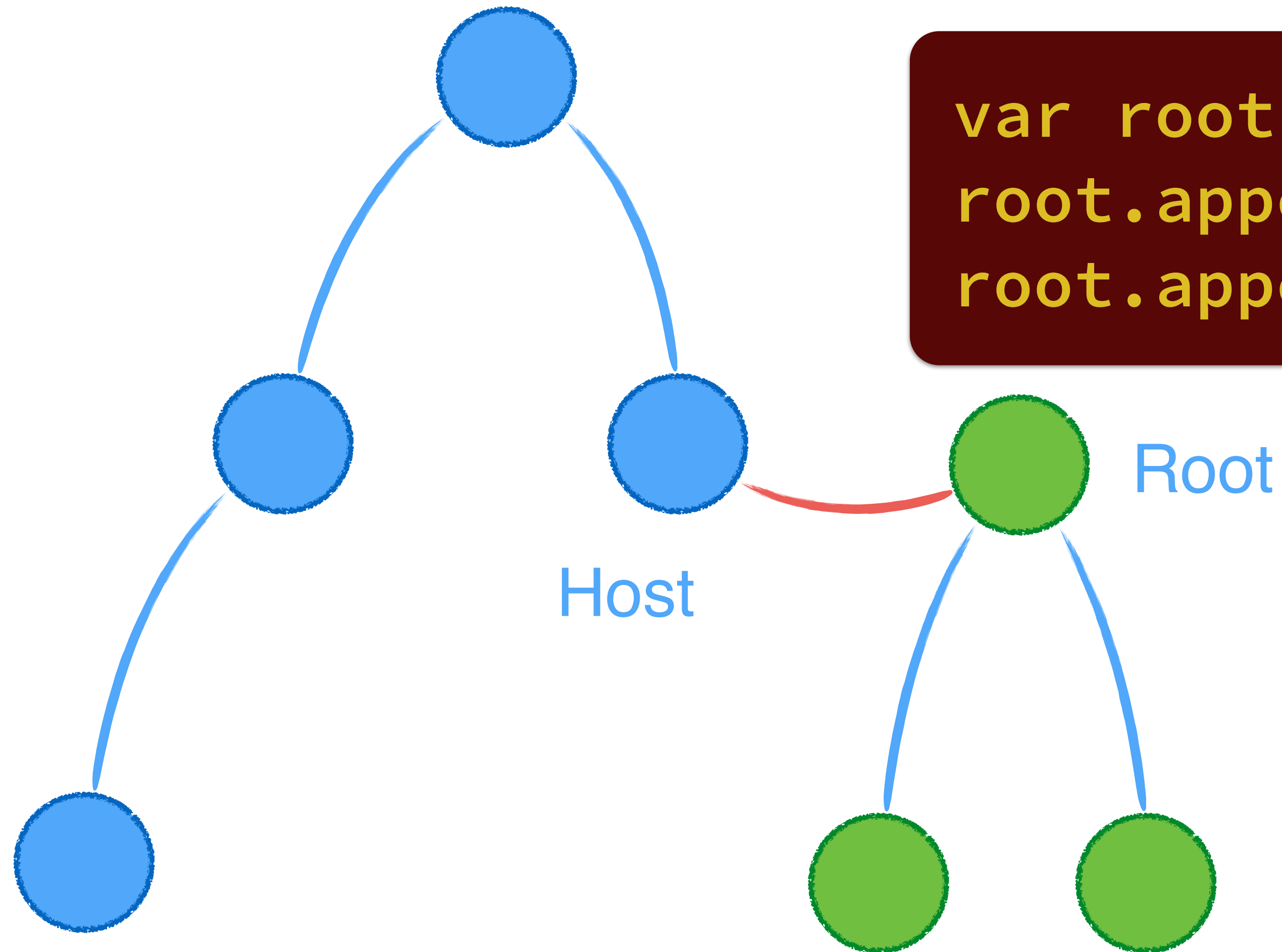
```
document.querySelector(".content")
```

```
.content {  
  color: blue;  
}
```

`".content"`

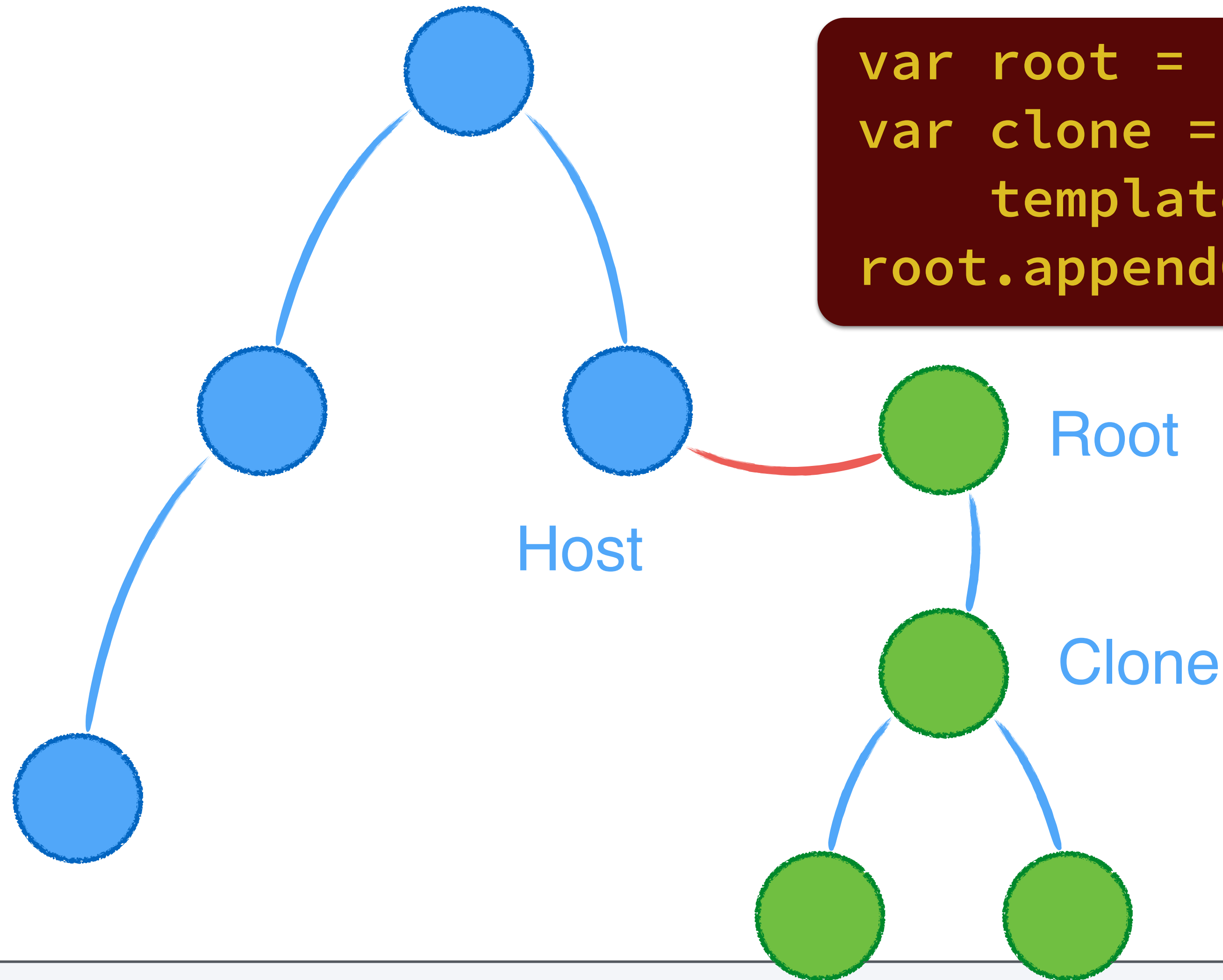
`".content"`

Shadow Dom



```
var root = host.createShadowRoot();  
root.appendChild(child1);  
root.appendChild(child2);
```

Shadow Dom



```
var root = host.createShadowRoot();  
var clone = document.importNode(  
  template.content, true);  
root.appendChild(clone);
```

Shadow Dom



25+ and
Android 4.4+



15+



Custom

Elements

Custom Elements

```
<div class="activity">  
  <a>Michael</a> had fun coding.  
</div>
```

How do we store the icon source and time?

What is a `<div>` with the class “activity” anyway?

Custom Elements

```
<div class="activity">  
  <a>Michael</a> had fun coding.  
</div>
```

```
<activity-card iconSrc="img/michael.jpg" time="Seconds ago">  
  <a>Michael</a> had fun coding.  
</activity-card>
```

Custom Elements

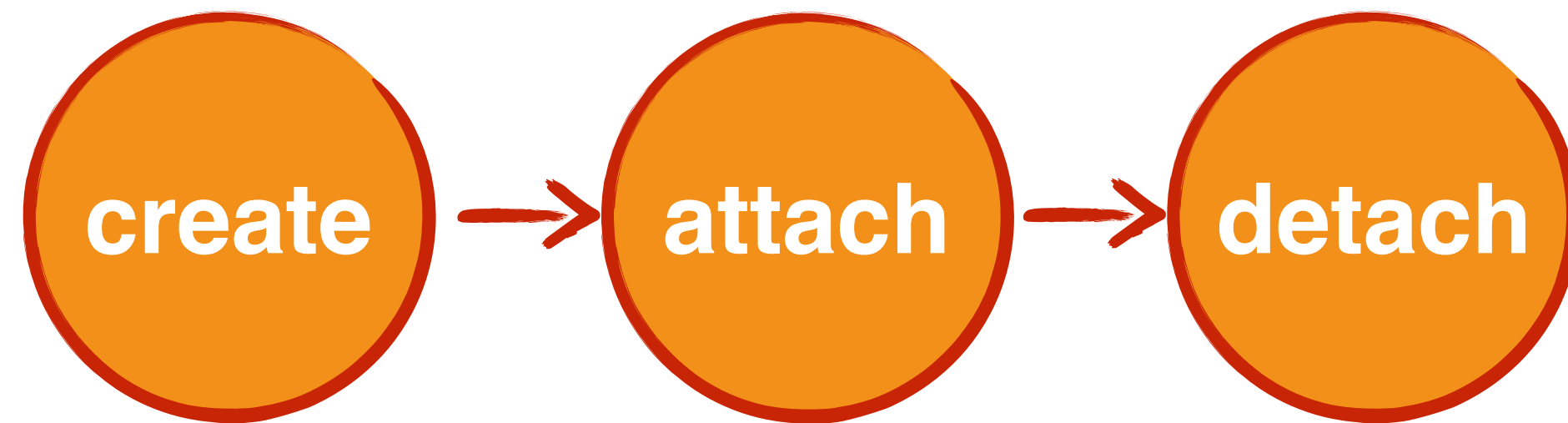
```
var activityCardPrototype = Object.create(HTMLElement.prototype);  
  
var options = {prototype: activityPrototype}  
document.registerElement("activity-card", options);
```

Custom Elements

```
var activityCardPrototype = Object.create(HTMLElement.prototype);  
  
var options = {prototype: activityPrototype}  
document.registerElement("activity-card", options);
```

```
<activity-card iconSrc="img/michael.jpg" time="Seconds ago">  
  <a>Michael</a> had fun coding.  
</activity-card>
```

Custom Elements



`createdCallback`

`attachedCallback`

`detachedCallback`



`attributeChangedCallback`

`(attrName, oldVal, newVal)`

Custom Elements

```
var activityPrototype = Object.create(HTMLElement.prototype);

activityPrototype.createdCallback = function() {
  var template = $("#activity-template");
  var clone = document.importNode(template.content, true);

  var host = $(this);
  $(".icon", clone).attr("src", host.attr("iconSrc"));
  $(".time", clone).text(host.attr("time"));

  var shadow = this.createShadowRoot();
  shadow.appendChild(clone);
};

document.registerElement("activity-card", {prototype: activityPrototype});
```

Custom Elements

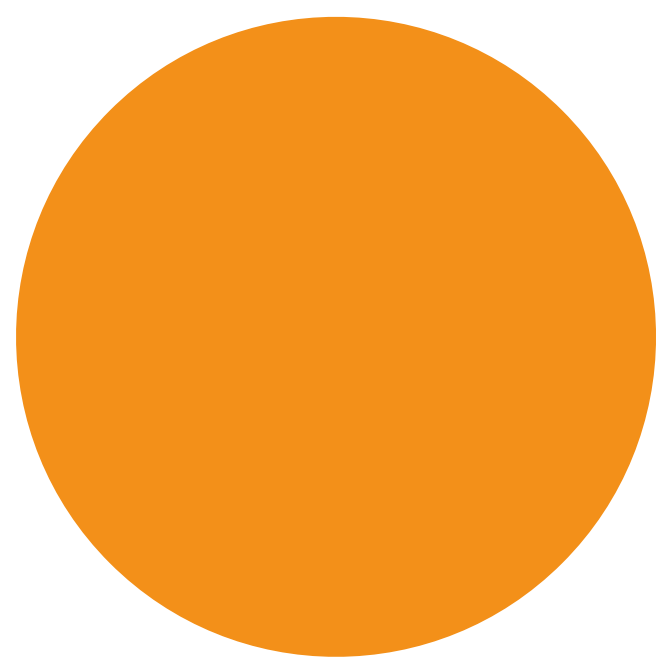


33+ and
Android 4.4.4+

20+



Html Import



Application Code

Component Code

Application Code

Html Import

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Activity Stream - Standard Web Component</title>
  <link href="stylesheet.css" rel="stylesheet">
  <script src="bower_components/jquery/dist/jquery.min.js"></script>
</head>
<body>
  <div class="activity-stream">
    <h2>Activities</h2>
    <activity-card iconSrc="../img/michael.jpeg" time="Seconds ago">
      <a href="profiles/michael">Michael</a> had fun writing web components.
    </activity-card>
    <activity-card iconSrc="../img/hendrick.jpeg" time="Minutes ago">
      <a href="profiles/hendrick">Hendrick</a> blogged on <a href="http://guigarage.com">GuiGarage</a>.
    </activity-card>
    <activity-card iconSrc="../img/michael.jpeg" time="1 hour ago">
      <a href="profiles/michael">Michael</a> needed an extra large cup of coffee.
    </activity-card>
    <activity-card iconSrc="../img/hendrick.jpeg" time="Yesterday">
      <a href="profiles/hendrick">Hendrick</a> watched a movie.
    </activity-card>
  </div>
</body>
</html>
```

`<link rel="import" href="activity-card.html">`

```
<template>
  <style>
    * {
      font-family: "Helvetica Neue Light", "HelveticaNeue-Light", "Helvetica Neue", Calibri, Helvetica, Arial, sans-serif;
    }
    .activity {
      width: 500px;
      height: 40px;
      padding: 10px;
      background-color: #f0f8ff;
      font-size: small;
      margin: 10px 0;
    }
    .activity .icon {
      float: left;
      border-radius: 100%;
    }
    .activity .time {
      float: right;
      color: #b7b7b7;
      font-style: italic;
    }
    .activity .content {
      margin-left: 60px;
    }
  </style>

  <div class="activity">
    <img class="icon" src="" width="40" height="40">
    <div class="time"></div>
    <div class="content"><content></content></div>
  </div>
</template>

<script>

  var activityPrototype = Object.create(HTMLElement.prototype);

  activityPrototype.createdCallback = function() {
    var template = document.querySelector('template');
    var clone = document.importNode(template.content, true);

    var host = $(this);
    $(".icon", clone).attr("src", host.attr("iconSrc"));
    $(".time", clone).text(host.attr("time"));

    var shadowRoot = this.createShadowRoot();
    shadowRoot.appendChild(clone);
  };

  // Register our new element
  document.registerElement('activity-card', {
    prototype: activityPrototype
  });
</script>
```

activity-card.html

Html Import



36+ and
Android Browser 46



23+



Use Web

Components today



webcomponents.js

- Today not all browsers support the new standards
- The community provides a polyfills to enable web components in browser that have no native support

```
$ bower install --save webcomponentsjs
```

install it with bower

use it in your code

```
<script src="bower_components/webcomponentsjs/webcomponents.js"></script>
```

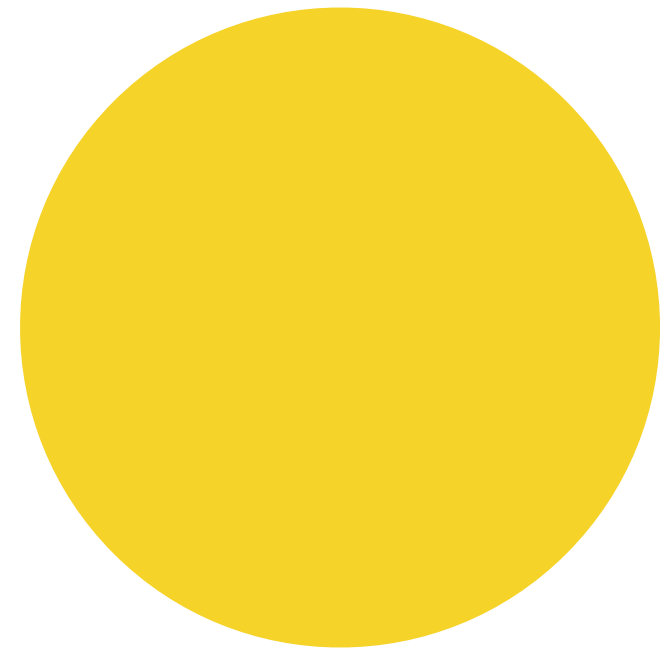
webcomponent.js

- The polyfills are the junction of X-Tag and Polymer basic libraries
- Mozilla created X-Tag as a polyfill to provide web components
- Google created Polymer as a polyfill to provide web components

webcomponent.js

- X-Tag and Polymer depends on webcomponents-js
- Both libraries provide additional features that are not part of the specification

webcomponents.org



Polymer



Polymer

- Adds sugar on top of web components
- Easy to use
- Extendable
- Polymer is created & supported by Google



Polymer

- Separated in several parts:

- Polymer API

- Iron Elements

- Paper Elements

- ...

www.polymer-project.org

1.0.4

Fe

Iron
Elements

Polymer core elements

1.0.5

Md

Paper
Elements

Material design elements

1.0.1

Go

Google Web
Components

Components for Google's APIs
and services

1.0.1

Au

Gold
Elements

Ecommerce Elements

1.0.0

Ne

Neon
Elements

Animation and Special Effects

1.2.0

Pt

Platinum
Elements

Offline, push, and more

1.0.0

Mo

Molecules

Wrappers for third-party
libraries

The logo for Iron Elements, featuring a stylized 'A' shape composed of blue and red geometric shapes.

Iron Elements

- Iron Elements are low level components
- Most Paper UI components base on Iron
- Icons, layouts ...

Icons



cloud-circle



cloud-done



cloud-download



cloud-off



content-copy



content-cut



content-paste



create



delete



description



dns



done



error



event



exit-to-app



expand-less



extension



face



favorite



favorite-outline

Icons



filter-b-and-w



filter-center-focus



filter-drama



filter-frames



filter-tilt-shift



filter-vintage



flare



flash-auto



flip



gradient



grain



grid-off



hdr-on



hdr-strong



hdr-weak



healing



iso



landscape



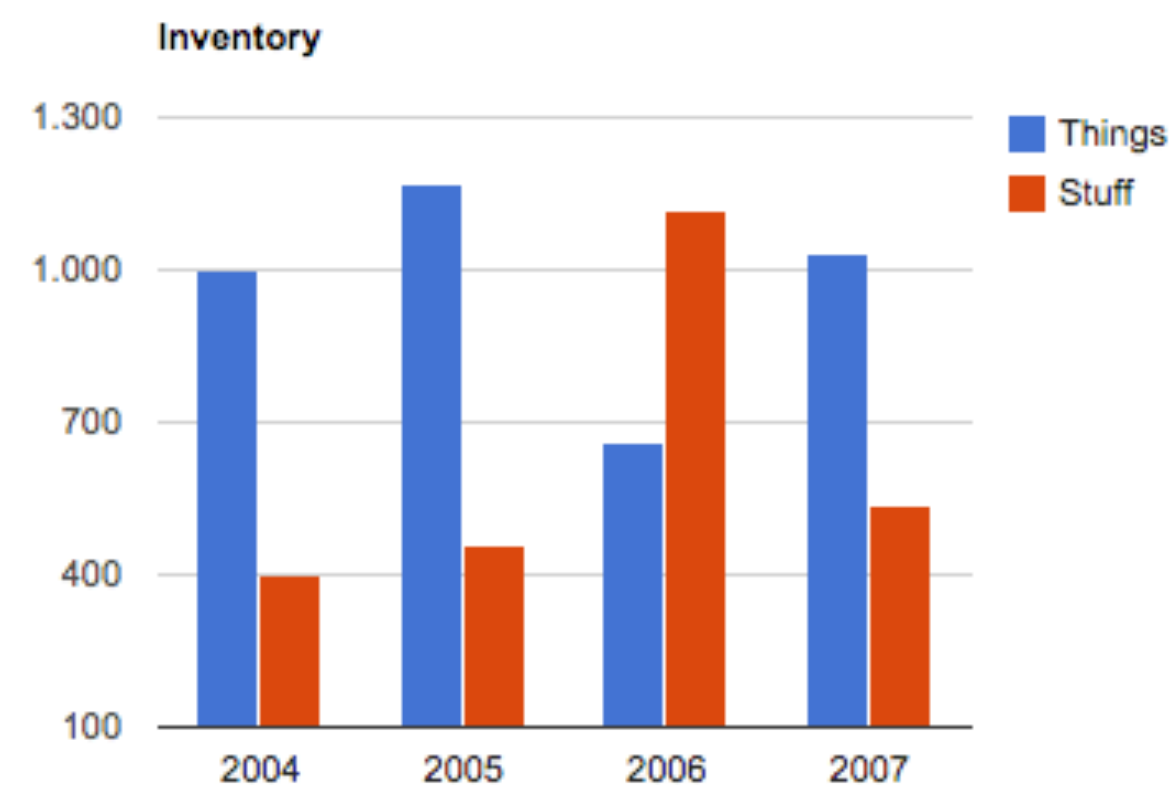
leak-add



leak-remove

Google Web Components

- Google is building a lot of components (maps, youtube...)



Hangout starten

```
<google-chart></google-chart>
```

```
<google-hangout-button></google-hangout-button>
```



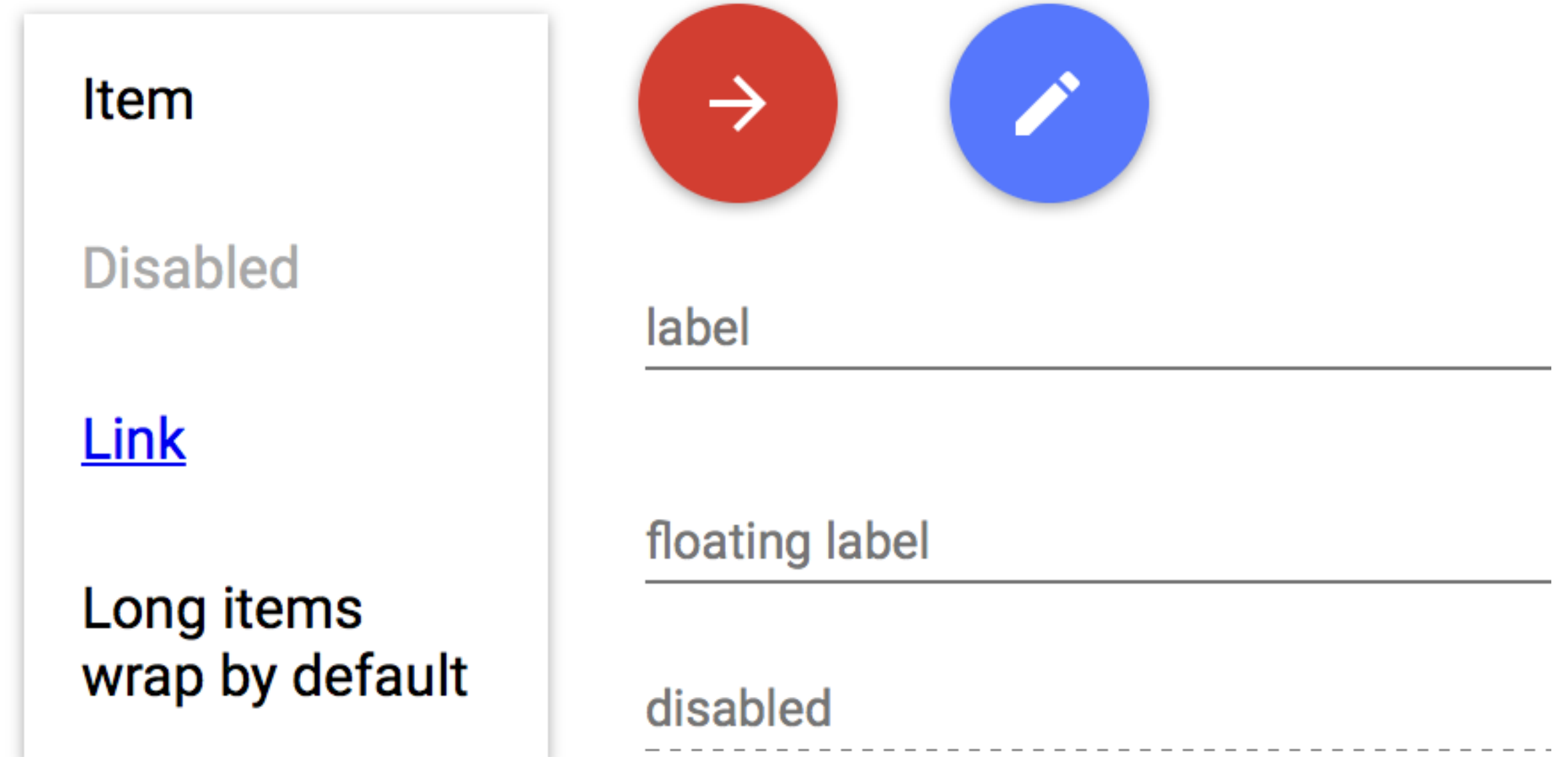
Reporter

Elemento



Polymer Paper

- Complete component library based on Material Design



Paper Elements

Checkbox

Radio Button

Toggle Button

Input

Toolbar

Spinner

Progress Bar

Slider

Toolbar

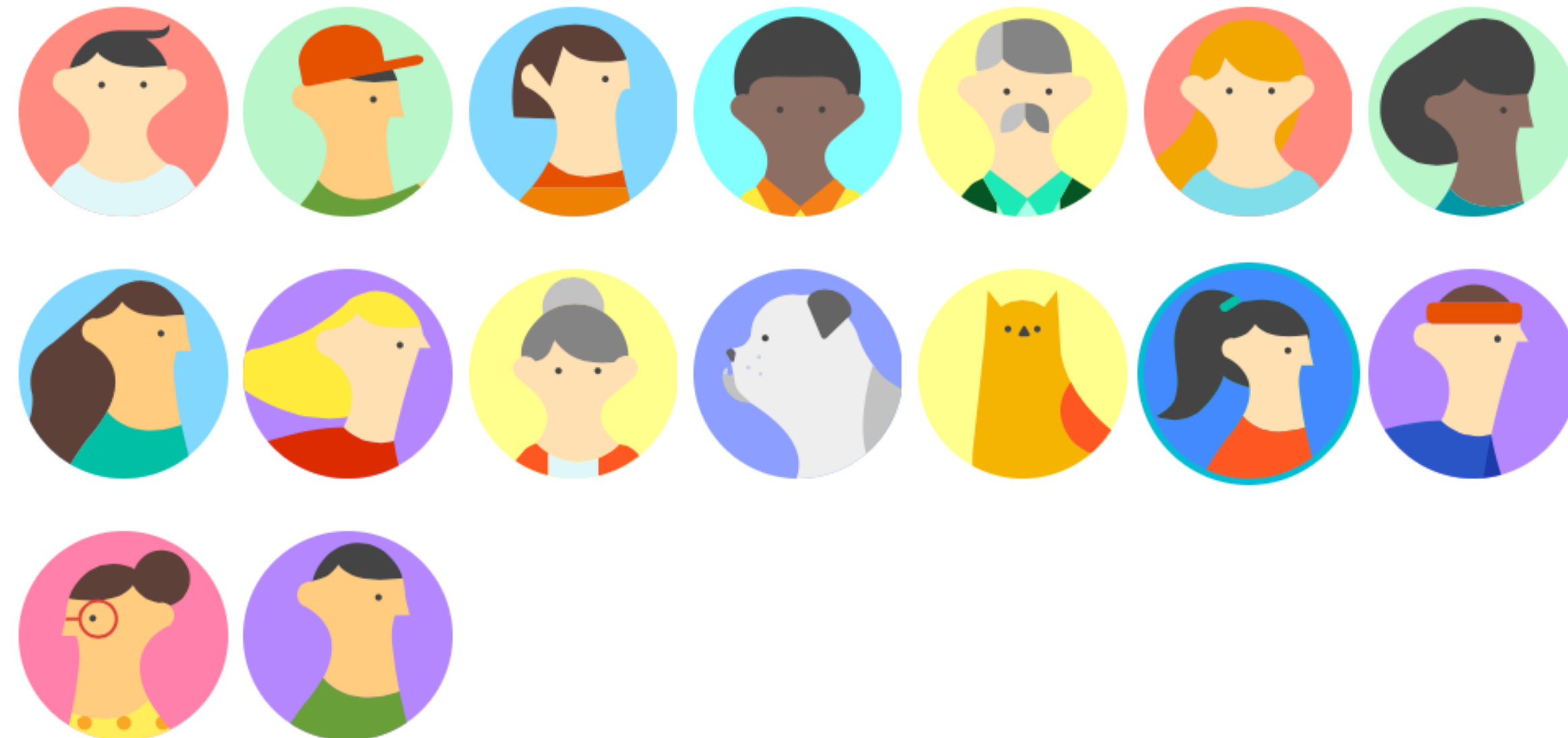


Sign In

Hendrik

Ebbers

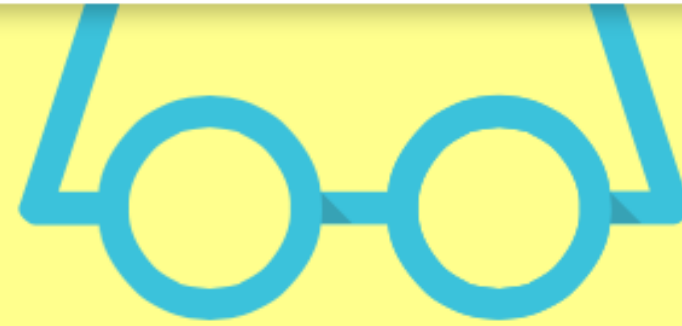
Choose an Avatar





Mole is a popular paste in Mexico. Mole is a mixture of chili's and what other tasty ingredient?

Fill in the Blank



General Knowledge

1 / 10

0

$$62+9$$

7

8

9

DEL

2nd

(

)

%

Deg

4

5

6

+

1/x

x^2

x^3

y^n

Rad

1

2

3

-

$x!$

$\sqrt{\quad}$

$\sqrt[n]{y}$

log

func

.

0

=

\div

sin

cos

tan

ln

Table

sinh

cosh

tanh

e^n

Graph

\times

EXP

π

EE

Rand

Bower



Bower

- A package manager for the web
- Search for dependencies and install them as packages
- Created by Twitter
- Open Source

www.bower.io



Bower

requires npm, node.js and git

```
$ npm install -g bower
```

in your project folder

```
$ bower init
```

download &
add module

add dependency to bower
file

```
$ bower install --save webcomponentsjs
```

The image features a white background with three yellow geometric shapes: a triangle in the top right corner, a triangle in the bottom left corner, and a triangle in the bottom right corner. The text is centered and consists of two lines.

Polymer Paper Components

How to use Components

- All components are build as web components
- A single component or a set can be added by using bower

```
$ bower install --save Polymer/paper-slider
```

import it in HTML



```
<link rel="import" href="bower_components/paper-slider/paper-slider.html">
```

Paper Button

- The default Button
- shows ripple animation on click

```
<paper-button>flat button</paper-button>  
<paper-button raised>raised button</paper-button>  
<paper-button noink>No ripple effect</paper-button>
```

Flat buttons

BUTTON COLORED DISABLED NOINK

Raised buttons

BUTTON COLORED DISABLED NOINK

Paper CheckBox

- A styled CheckBox
- State can be defined as attribute

```
<paper-checkbox></paper-checkbox>
```

```
<paper-checkbox checked></paper-checkbox>
```



Auto-updates

Auto-update apps over wifi only



Clear search history

Remove all the searches you have ever performed



Use as default browser

Paper Dialog

- A dialog
- Supports title, modality, actions, ...

```
<paper-dialog heading="Title">  
  <p>Some content</p>  
</paper-dialog>
```

Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

MORE INFO...

DECLINE

ACCEPT

Overview

label

floating label

disabled

ITEM ONE


ITEM TWO

ITEM THREE

Overview

Wi-Fi 

Bluetooth 

Custom Colors 

Disabled 

Radio button



Radio button (toggles)



Icon buttons



Regular



Item

Disabled

[Link](#)

Long items
wrap by default

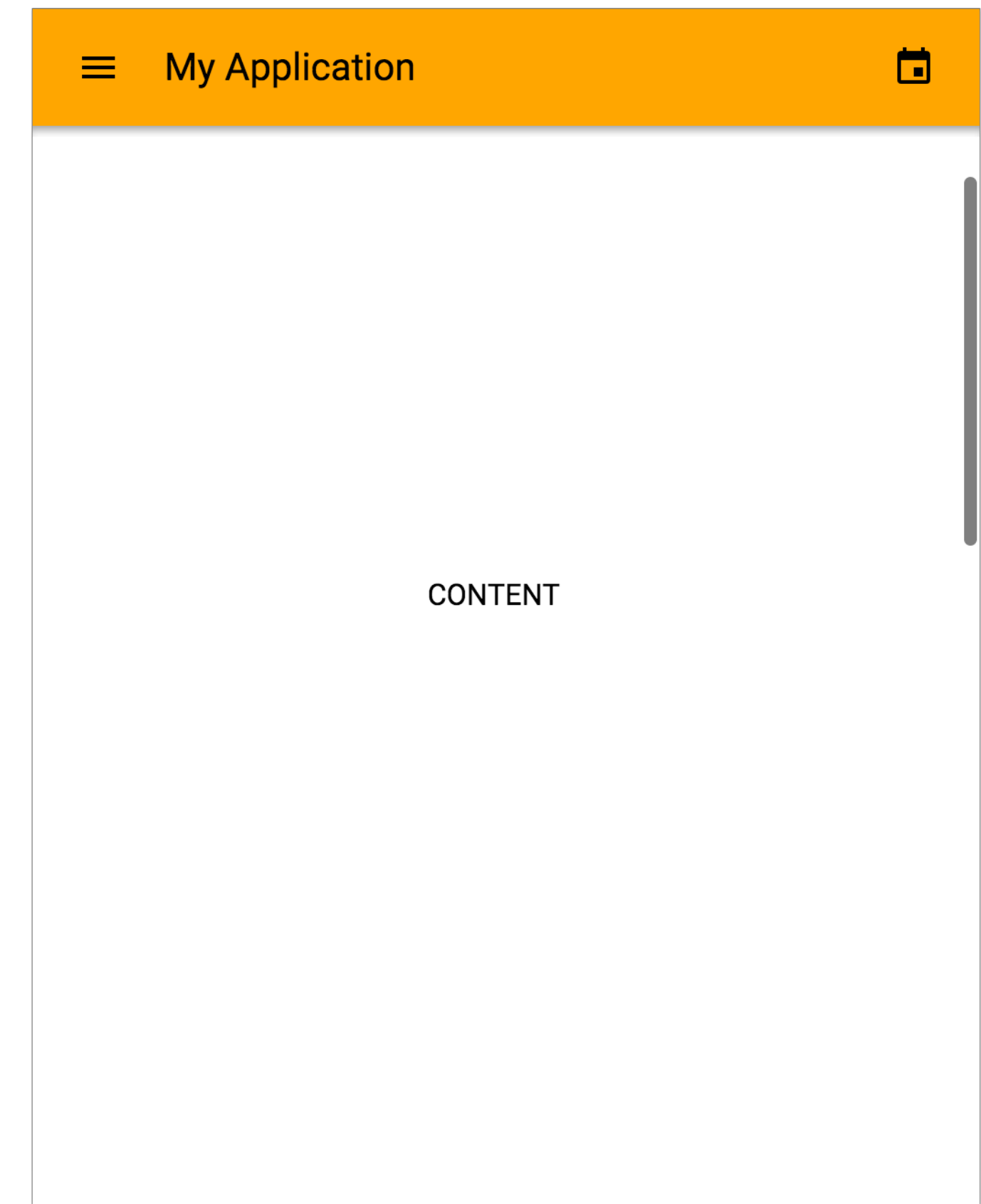
Layout

- HeaderPanel
- Toolbar
- DrawerPanel
- Scaffold

Toolbar

- An application toolbar
- Toolbar content will be aligned

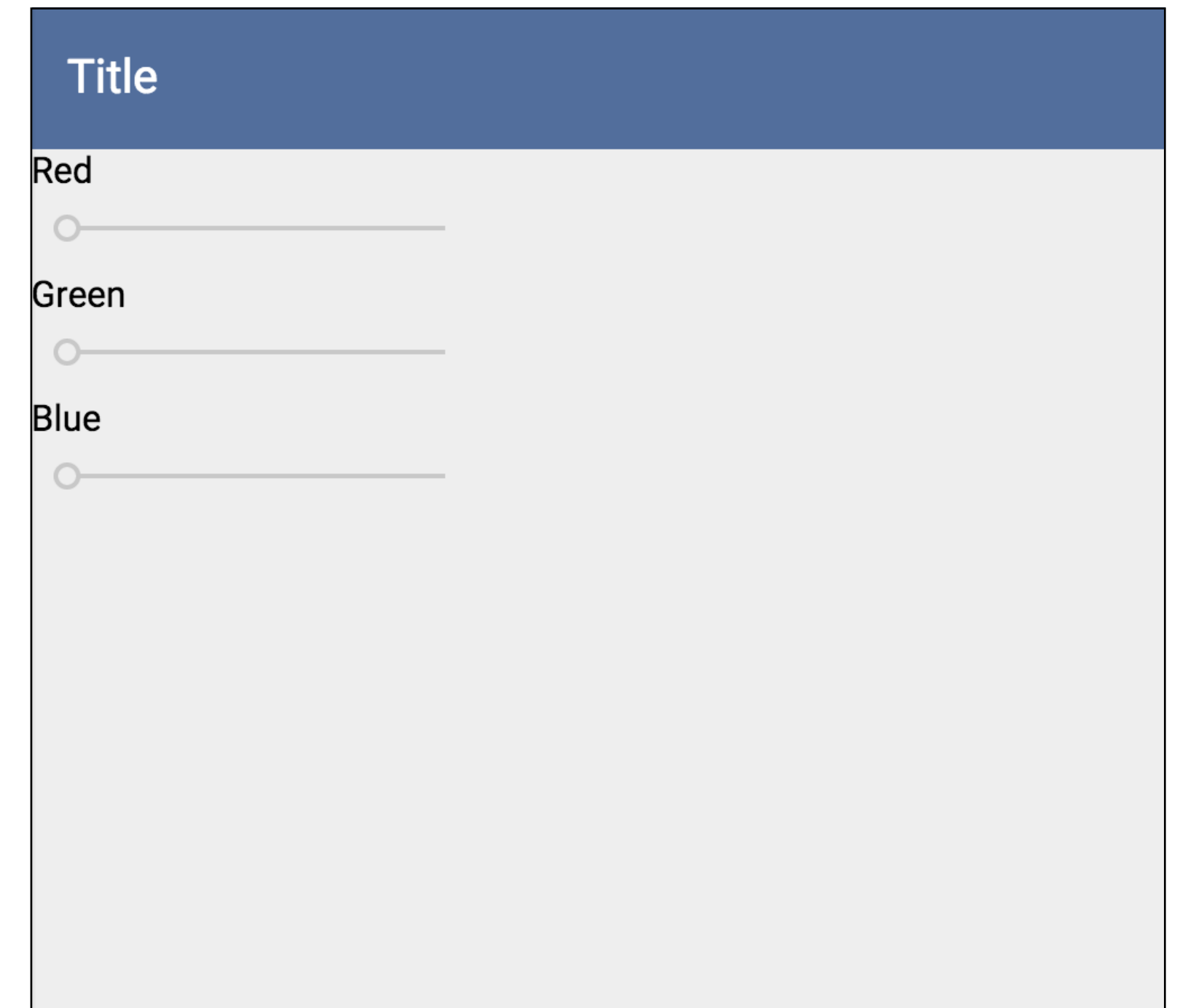
```
<core-toolbar>  
  <paper-icon-button icon="menu"></paper-icon-button>  
  <div>My Application</div>  
  <span flex></span>  
  <paper-icon-button icon="event"></paper-icon-button>  
</core-toolbar>
```



HeaderPanel



- Wrapper around toolbar and content
- Toolbar always on top
- Content scrollable

```
<core-header-panel flex>  
  <core-toolbar>Title</core-toolbar>  
  <div>content</div>  
</core-header-panel>
```



DrawerPanel

- Adds a responsive menu
- Defines attributes to open and close the menu
- Normally wraps 2 core-header-panel

Menu	Title
 One	Red <input type="text"/>
 Two	Green <input type="text"/>
	Blue <input type="text"/>

DrawerPanel

```
<core-drawer-panel>
  <core-header-panel drawer>
    <core-toolbar></core-toolbar>
    <core-menu>
      <core-item label="One"></core-item>
    </core-menu>
  </core-header-panel>
  <core-header-panel main>
    <core-toolbar>
      <paper-icon-button core-drawer-toggle icon="menu"></paper-icon-button>
    </core-toolbar>
    <div>content</div>
  </core-header-panel>
</core-drawer-panel>
```

drawer panel

main panel

show drawer

Scaffold

- Basic skeleton
- Best practice to create an application
- Contains all the shown features

```
<core-scaffold>
  <core-header-panel navigation flex>
    <!-- nav drawer -->
  </core-header-panel>

  <span tool>Title</span>

  <div>content</div>
</core-scaffold>
```

defines the drawer

defines the main toolbar

using Paper

Paper Demo

- Create a new folder
- Install needed modules with Bower

```
$ mkdir app  
$ cd app  
$ bower init  
$ bower install --save PolymerElements/paper-elements
```

Paper Demo

- Create index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>

  <body>Hello World</body>

</html>
```

Paper Demo

- add webcomponents.js

```
<head>  
  . . .  
  <script src="bower_components/webcomponentsjs/webcomponents.js">  
  </script>  
</head>
```


Paper Demo

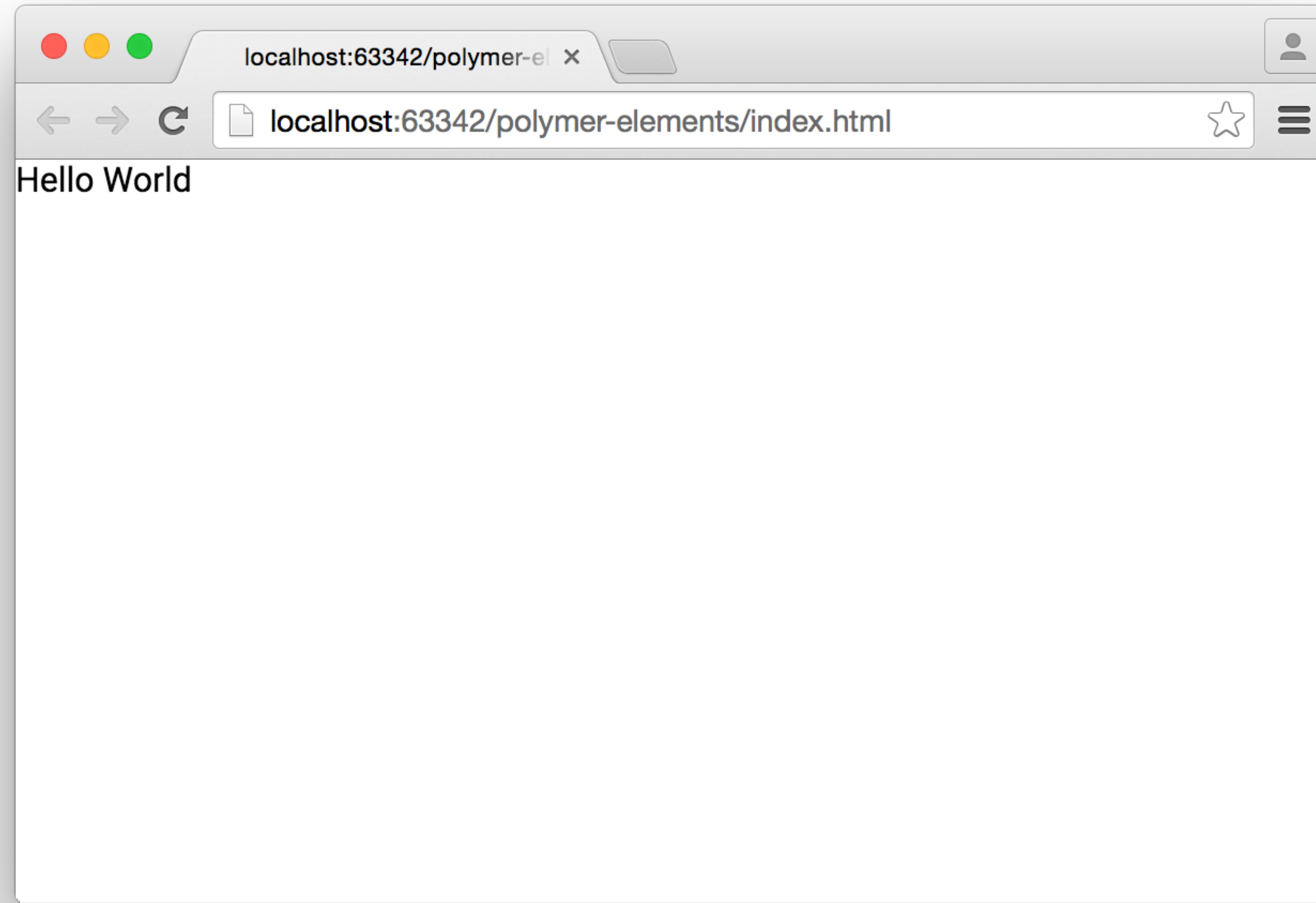
- add Roboto Font

```
<head>
  <link rel="import" href="bower_components/font-roboto/
  roboto.html">

  <style>
    html, body {
      font-family: 'RobotoDraft', sans-serif;
    }
  </style>

</head>
```

Paper Demo



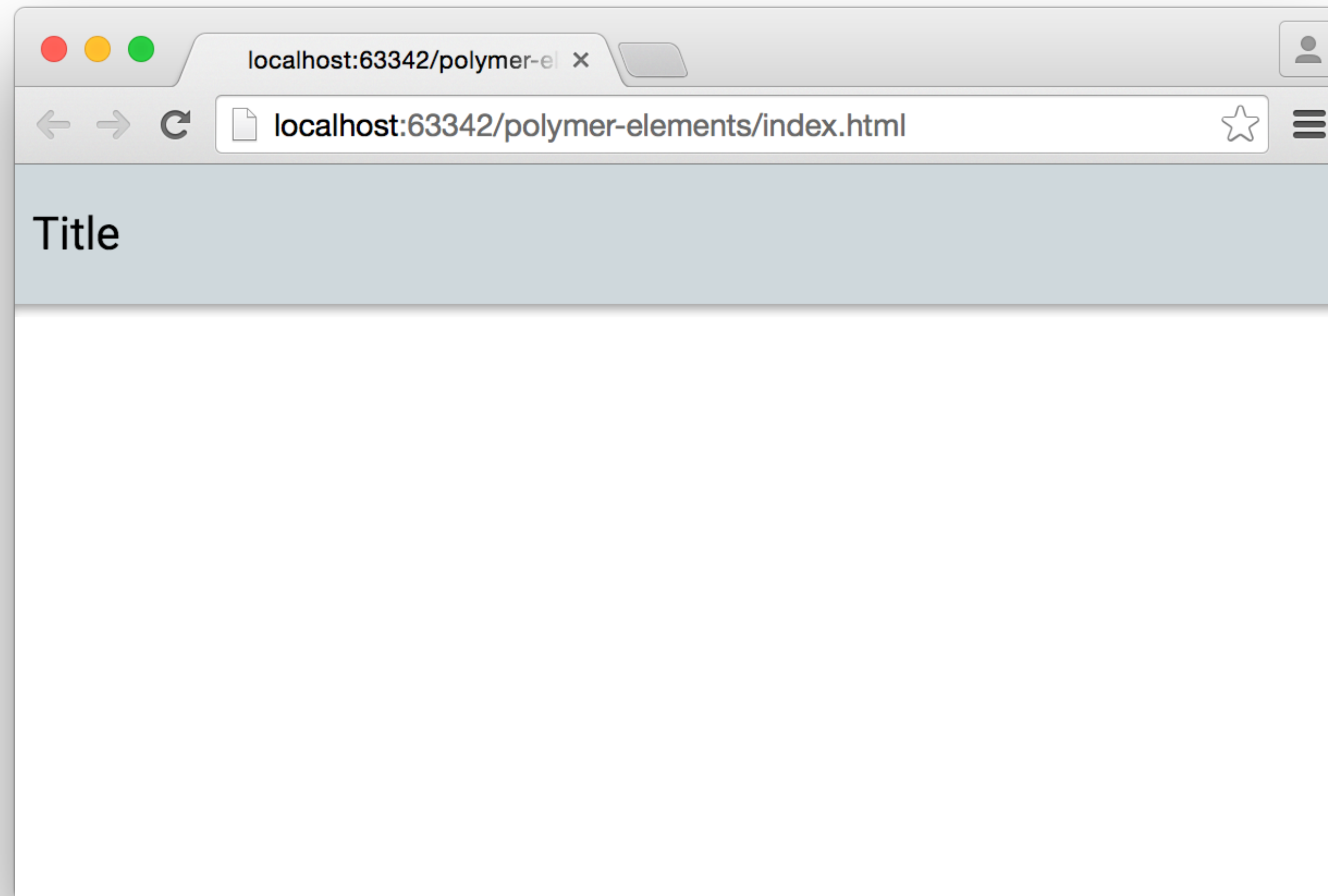
Paper Demo

- add a header panel with a toolbar

```
<link rel="import" href="bower_components/paper-styles/paper-styles.html">  
<link rel="import" href="bower_components/paper-header-panel/paper-header-panel.html">  
<link rel="import" href="bower_components/paper-toolbar/paper-toolbar.html">
```

```
<body class="fullbleed layout vertical">  
<paper-header-panel class="flex">  
  <paper-toolbar>  
    <div>Hello World!</div>  
  </paper-toolbar>  
</paper-header-panel>  
</body>
```

Paper Demo



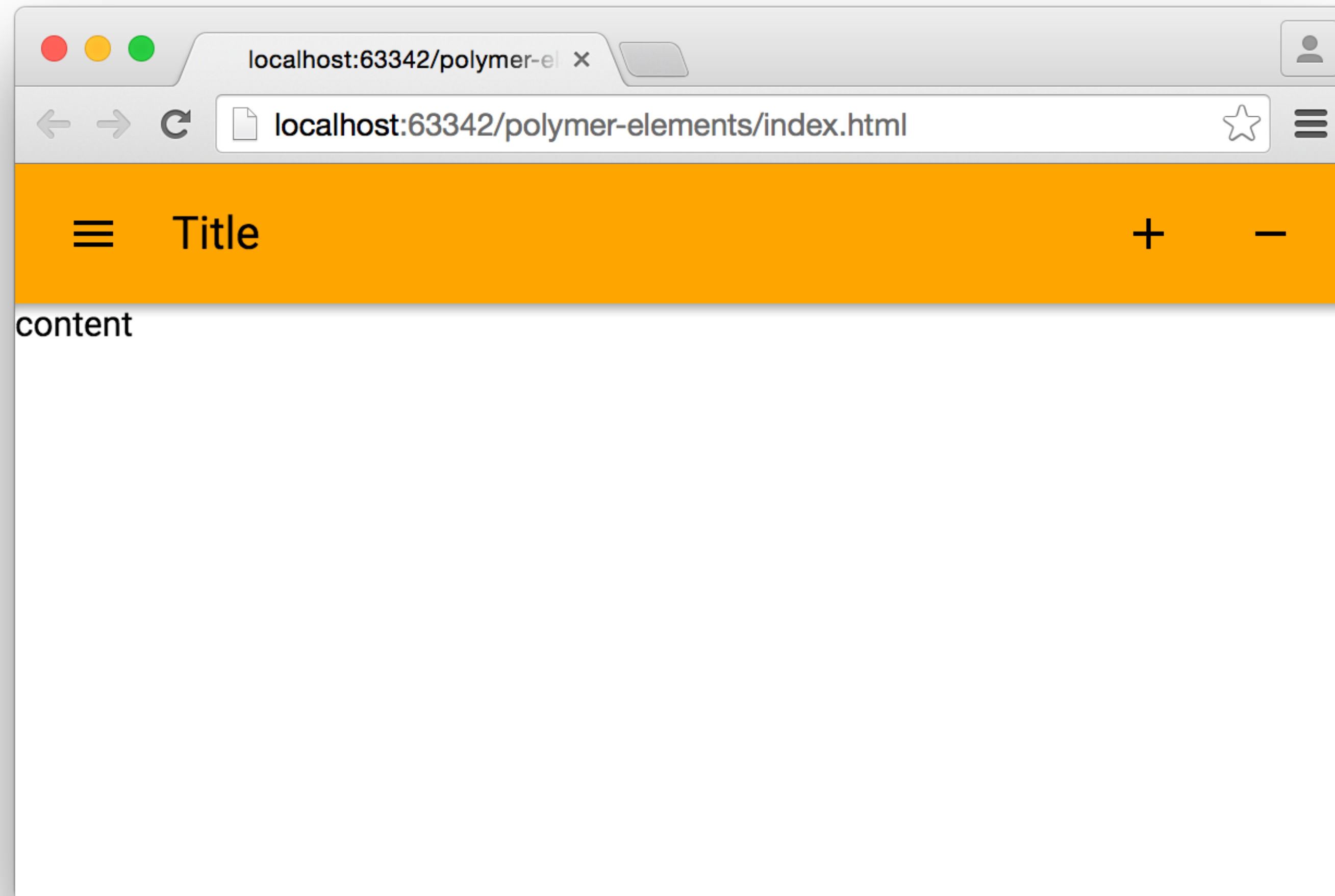
Paper Demo

- Add some content to the toolbar

```
<link rel="import" href="bower_components/iron-icons/iron-icons.html">  
<link rel="import" href="bower_components/paper-icon-button/paper-icon-button.html">
```

```
<paper-toolbar>  
  <paper-icon-button icon="menu"></paper-icon-button>  
  <span class="flex">Title</span>  
  <paper-icon-button icon="add"></paper-icon-button>  
  <paper-icon-button icon="remove"></paper-icon-button>  
</paper-toolbar>
```

Paper Demo



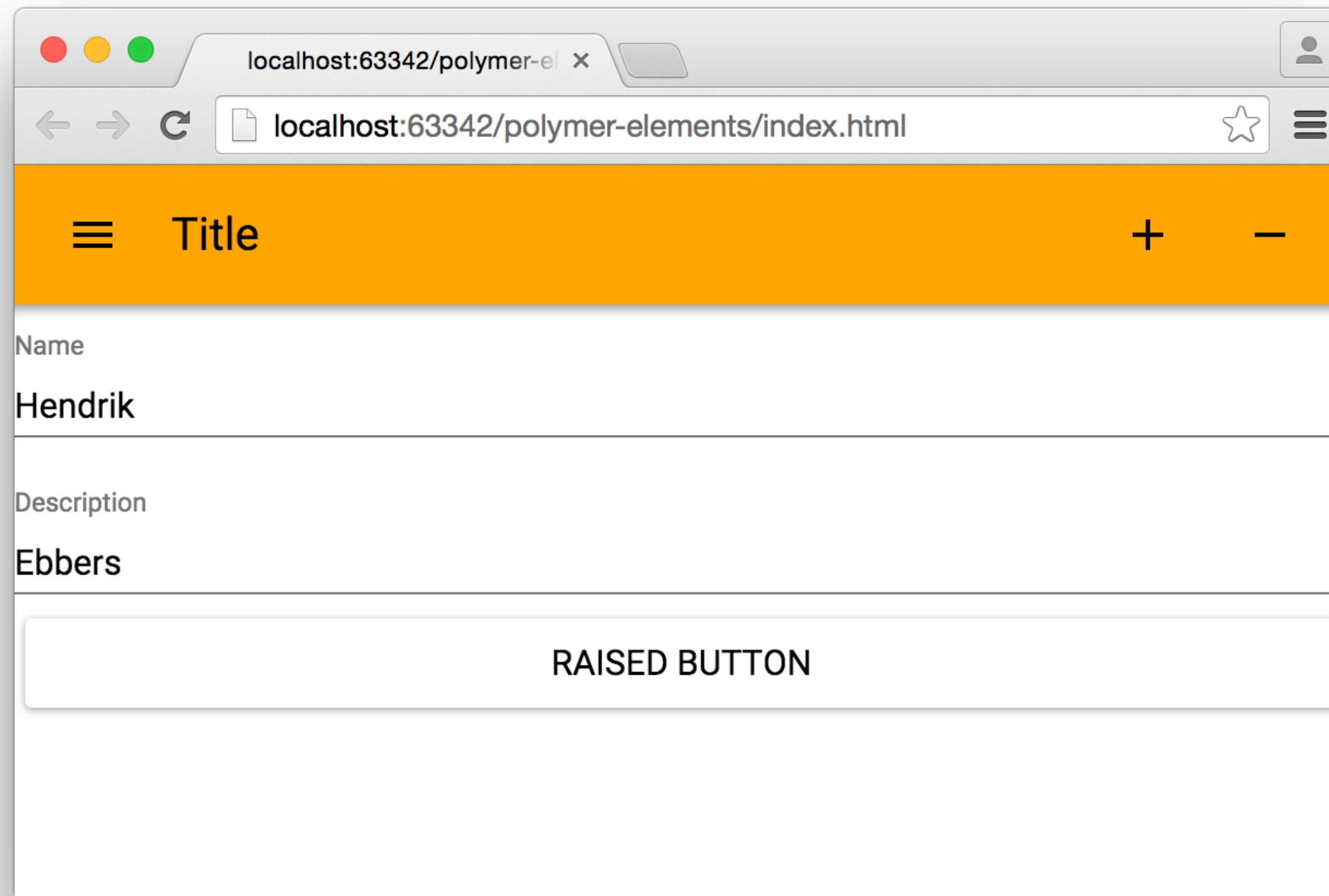
Paper Demo

- Add paper elements as content

```
<link rel="import"  
  href="bower_components/paper-input/paper-input.html">  
<link rel="import"  
  href="bower_components/paper-button/paper-button.html">
```

```
<div class="layout vertical">  
  <paper-input label="Name"></paper-input>  
  <paper-input label="Description"></paper-input>  
  <paper-button raised>save</paper-button>  
</div>
```

Paper Demo



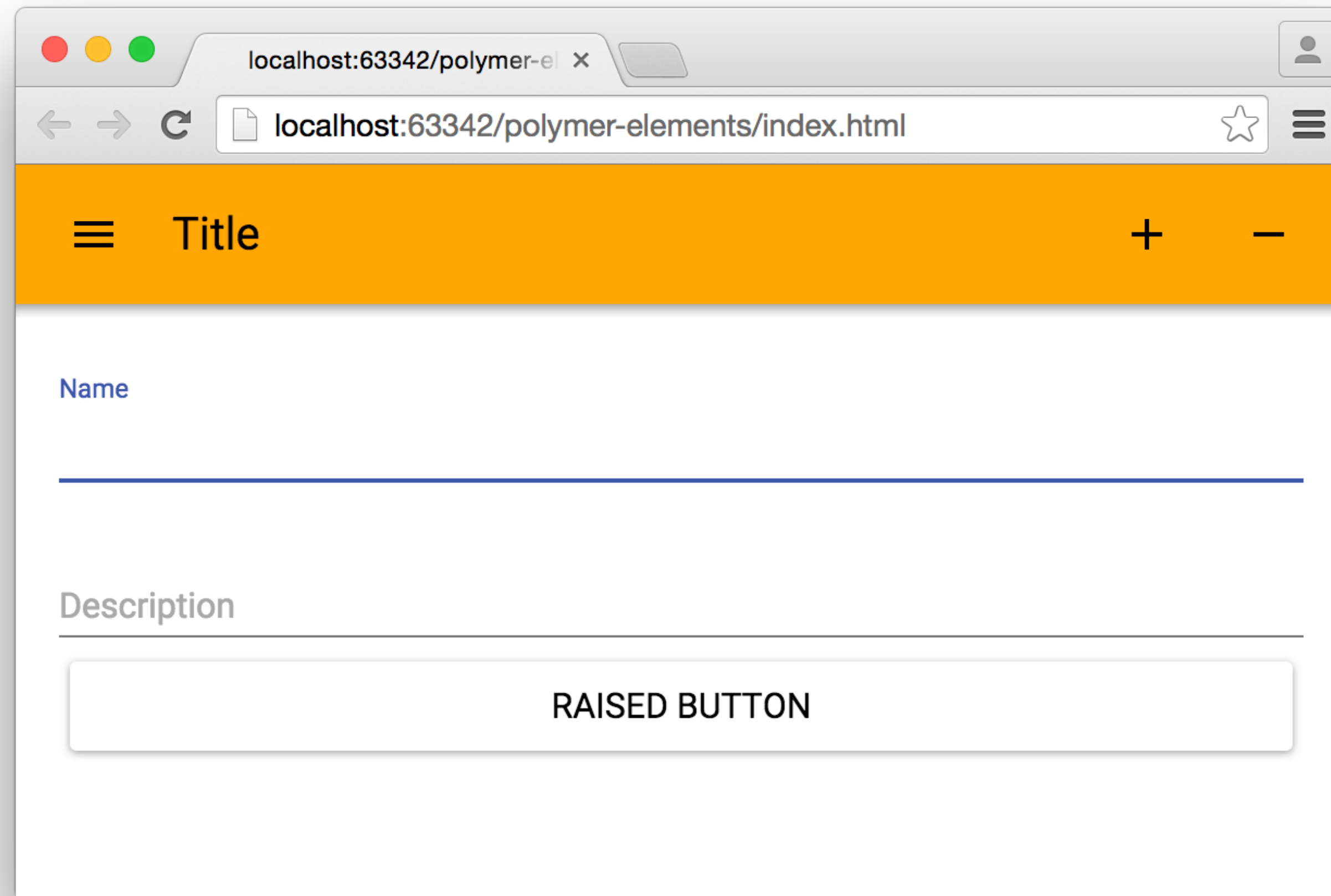
Paper Demo

- Add some padding

```
.content {  
    padding: 20px;  
}
```

```
<div class="content layout vertical">  
  <paper-input label="Name"></paper-input>  
  <paper-input label="Description"></paper-input>  
  <paper-button raised>save</paper-button>  
</div>
```

Paper Demo





Custom

Components



Custom Components

- Polymer can be used to create custom web components
- Defines a nice and modern API to create components
- Adds several features on top of the spec

A First Example

```
<link rel=„import" href="bower_components/polymer/polymer.html">
```

A First Example

```
<link rel=„import“ href=“bower_components/polymer/polymer.html“>
```

```
<dom-module id=“dom-element“>
```

```
</dom-module>
```

A First Example

```
<link rel=„import“ href=“bower_components/polymer/polymer.html“>
```

```
<dom-module id=“dom-element“>
```

```
  <template>
```

```
    <p>I'm a DOM element. This is my local DOM!</p>
```

```
  </template>
```

```
</dom-module>
```

A First Example

```
<link rel=„import“ href=“bower_components/polymer/polymer.html“>
```

```
<dom-module id=“dom-element“>
```

```
  <template>
```

```
    <p>I'm a DOM element. This is my local DOM!</p>
```

```
  </template>
```

```
  <script>
```

```
    Polymer({is: "dom-element"});
```

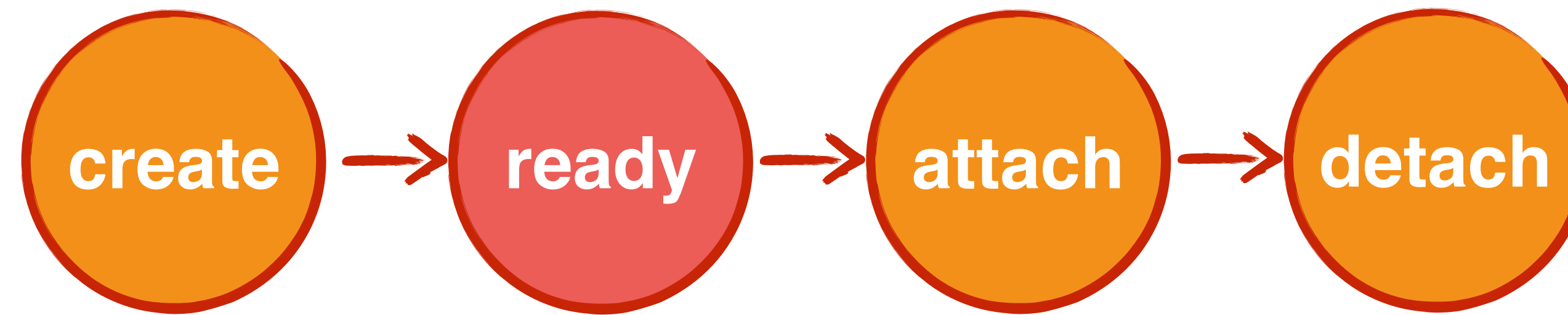
```
  </script>
```

```
</dom-module>
```


Custom Components

- Component name must contain a "-"
- Polymer script call must be included
- Template is not needed
- Once it is created it can be used as custom tag

Lifecycle



Example

```
<script>
  Polymer({
    is: „dom-element“,
    ready: function () {
      console.log('Ready!');
    },
    detached: function() {
      console.log(this.localName + '#' + this.id + ' was detached');
    },
    attributeChanged: function(name, type) {
      console.log('Attribute ' + name + ' was changed');
    }
  });
</script>
```

Property

- A property can be of the following types:
 - Boolean, Date, Number, String, Array or Object

Property

```
Polymer({  
  is: 'my-component',  
  properties: {  
    user: String,  
    isHappy: Boolean  
  }  
});
```

Property

- Properties can be set in JavaScript

```
ready: function () {  
  this.set('user', 'unknown');  
}
```

Properties can be set as HTML attribute

```
<my-component user='unknown' ></my-component>
```

Property

- Default value can be specified

```
properties: {  
  user: {  
    type: String,  
    value: 'unknown'  
  },  
  isHappy: Boolean  
}
```

Data Binding

- binds a property to an attribute of an element

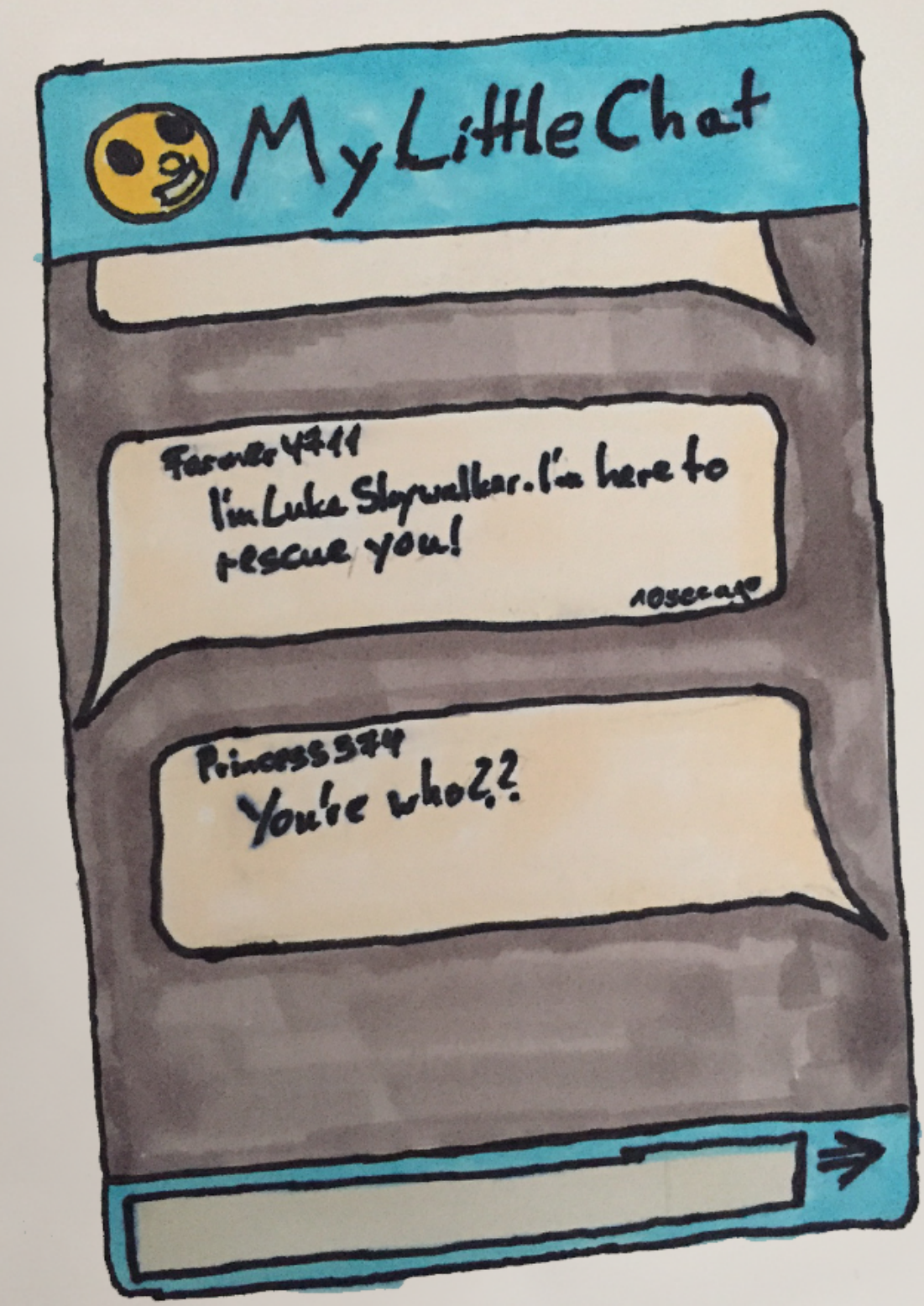
```
<child-element name="{{myName}}"></child-element>
```

- Use `[[prop]]` for one-way-binding
- Property defines if `{{prop}}` is one-way or bidirectional binding

Ok, let's create
an application

A solid yellow shape that starts as a thin line on the left and curves upwards to the right, filling the bottom portion of the slide.

Москир



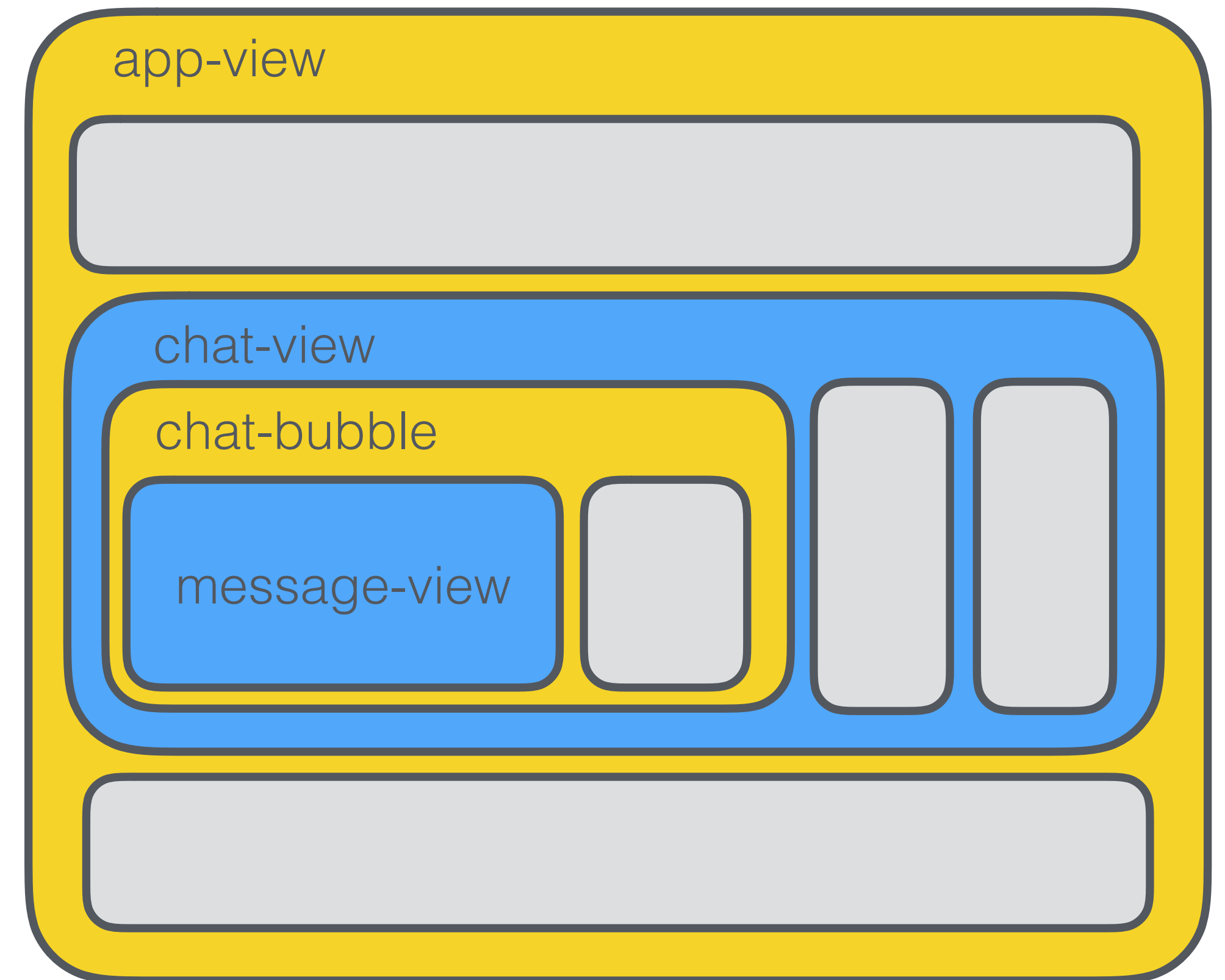


Application Structure

Application Structure

- A Polymer application is based on web components
- The components define a hierarchy
 - `chat-view` component contains several `chat-bubble` components

Application Structure



Like building futuristic Matrjoschkas with Lego



Application Structure

- The demo will use bower to handle all dependencies
- Real applications should use a build script
 - like Gulp or Grunt

index.html

- add web components polyfill script
- Import basic dependencies like Roboto-Font and Polymer layouts
- body contains only one web component: the application

imports

web components polyfill



```
<script src="bower_components/webcomponentsjs/webcomponents.js"></script>
```

font and layouts



```
<link rel="import" href="bower_components/paper-styles/paper-styles.html">  
<link rel="import" href="bower_components/font-roboto/roboto.html">
```

component-xy.html

- Contains the definition of one web component
- All components that are needed internally should be imported

Workspace

```
bower.json
```

```
index.html
```

```
chat-bubble.html
```

```
chat-view.html
```

```
chat-header.html
```

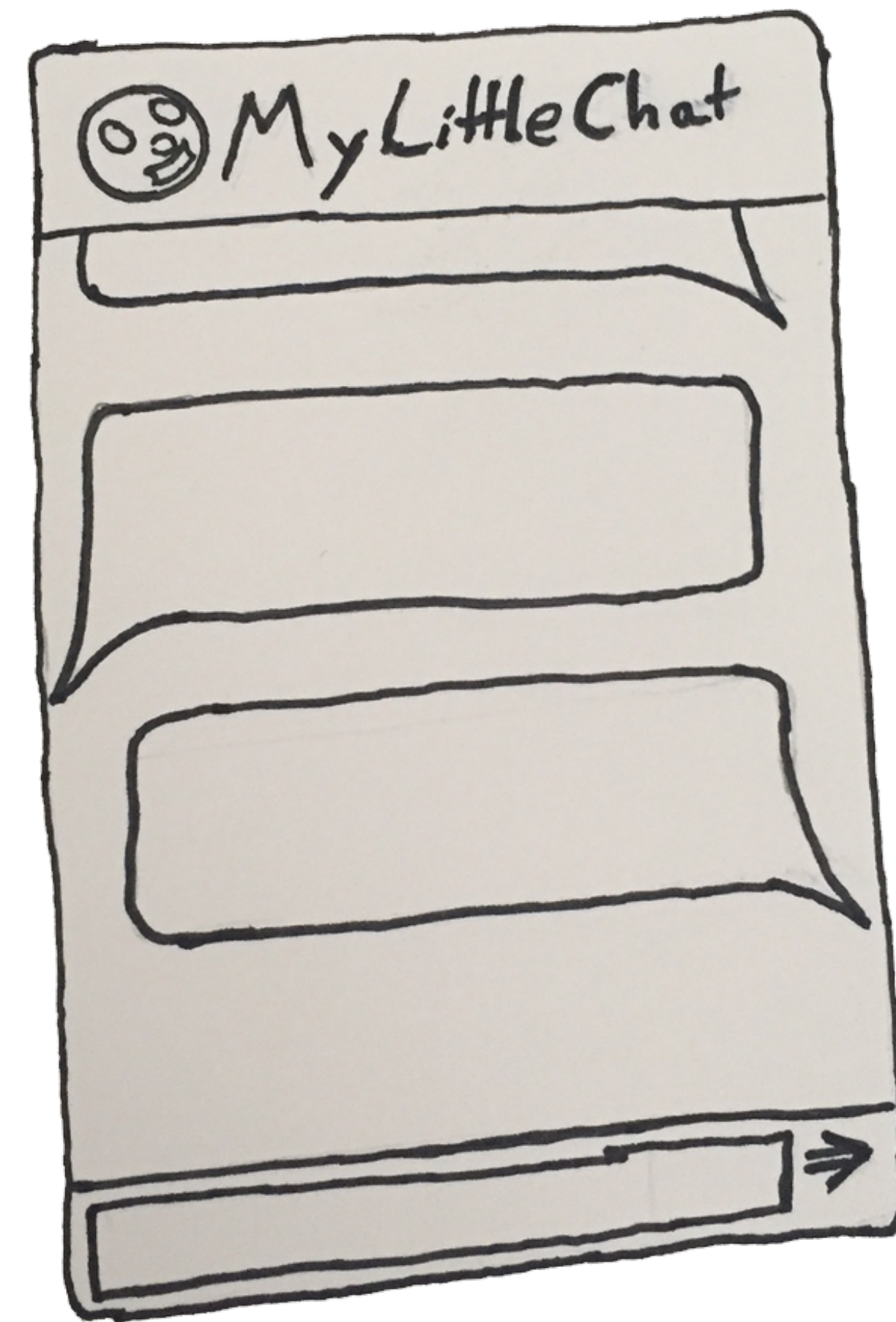
```
chat-footer.html
```

```
chat-app.html
```

Basic Layout

Application Basic Layout

- Header with fixed size
- Resizable main area
- Footer with fixed size

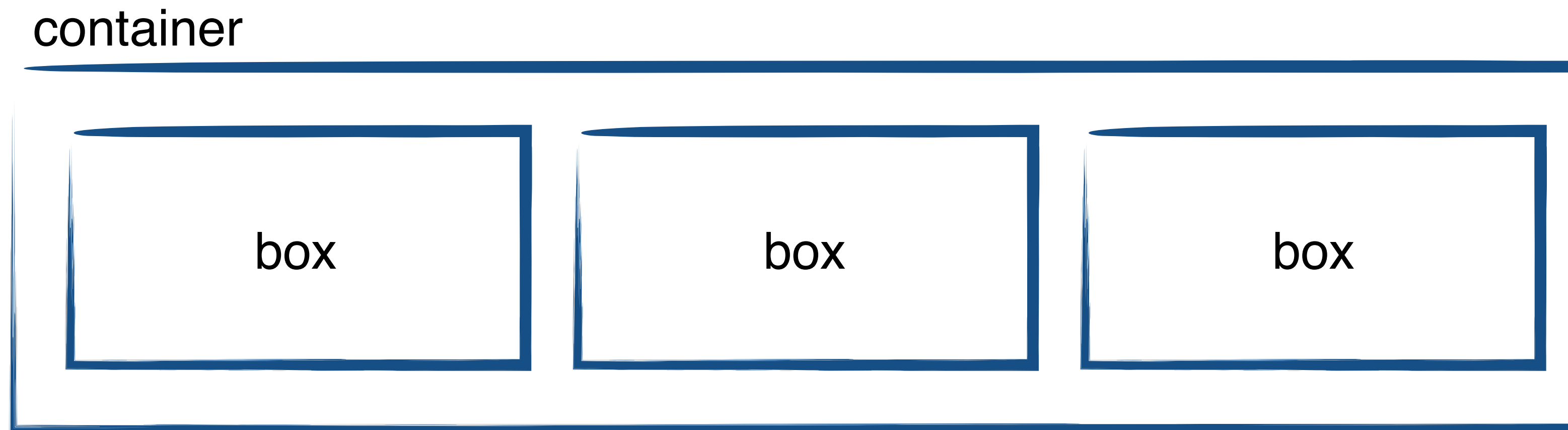


Polymer Layout

- The Polymer layout is based on flexbox
- Use Polymer layout classes to simplify the usage of flexbox

Flexbox

- New CSS 3 layout
- Flexbox consists of flex containers and flex items.

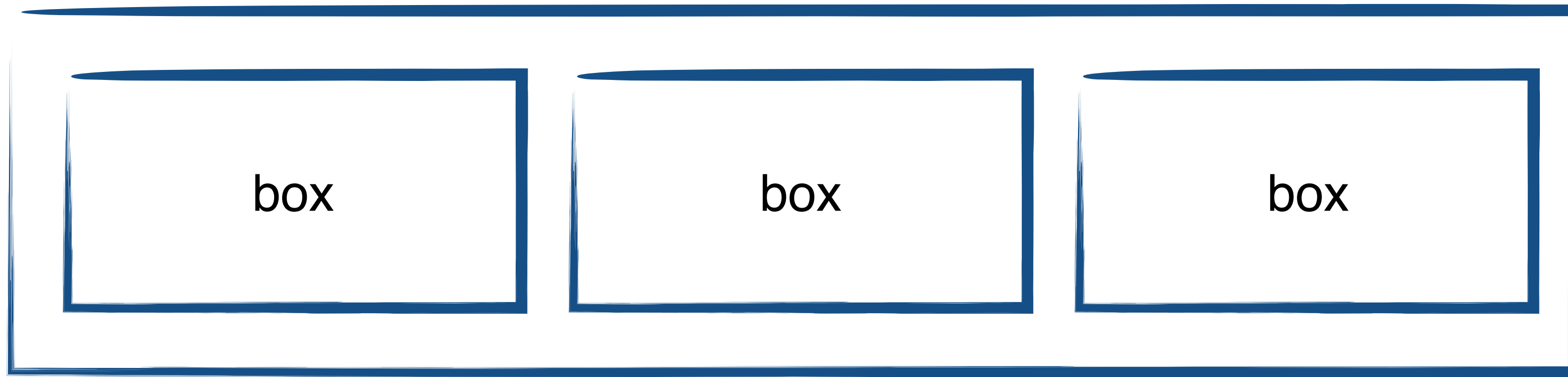


Flexbox

- Contents can be laid out in any flow direction (leftwards, rightwards, downwards, or even upwards!)
- Contents can be laid out linearly along a single (main) axis or wrapped into multiple lines along a secondary (cross) axis
- Contents can “flex” their sizes to respond to the available space
- ...

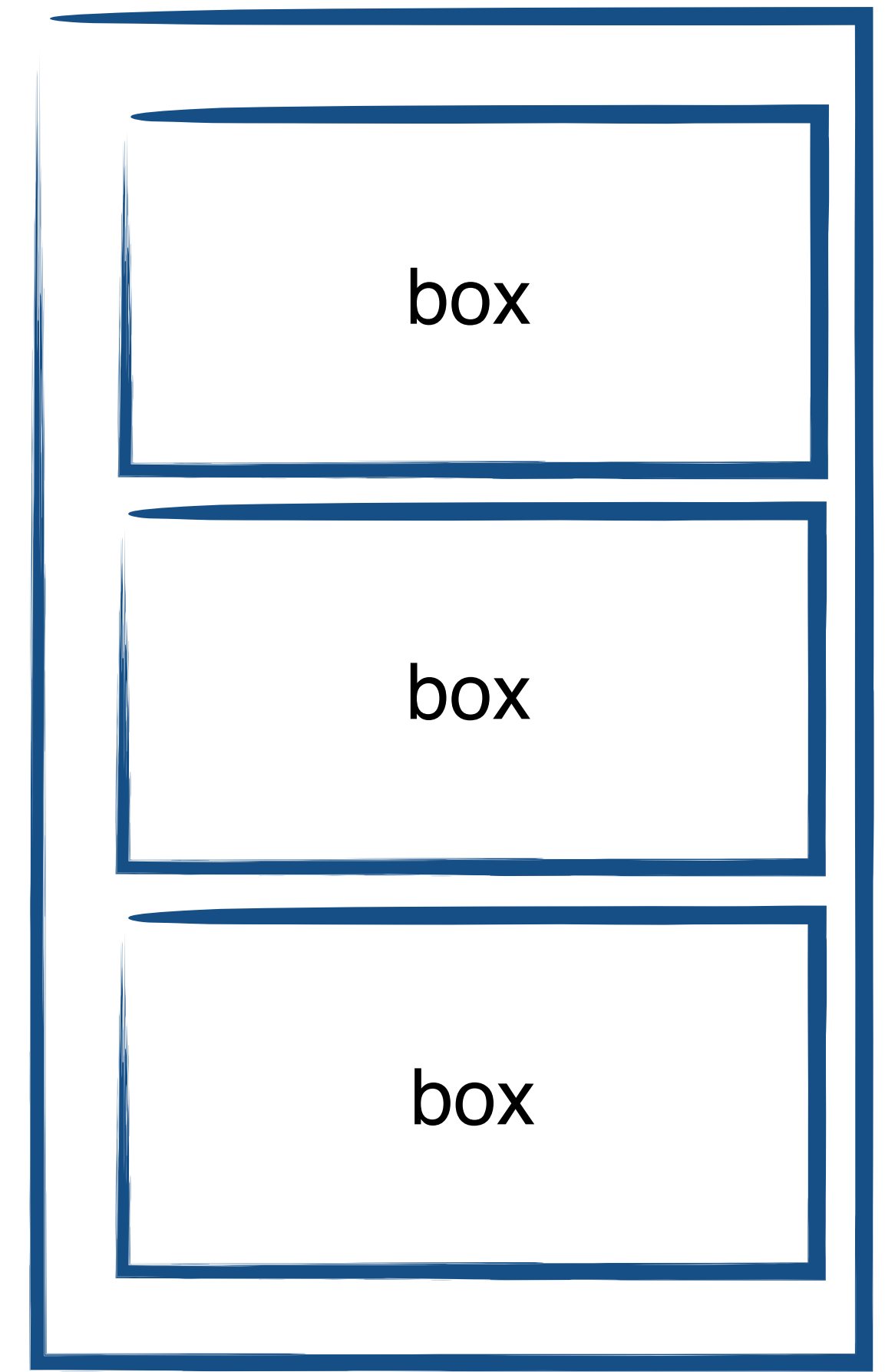
Flexbox

container



`flex-direction: row;`

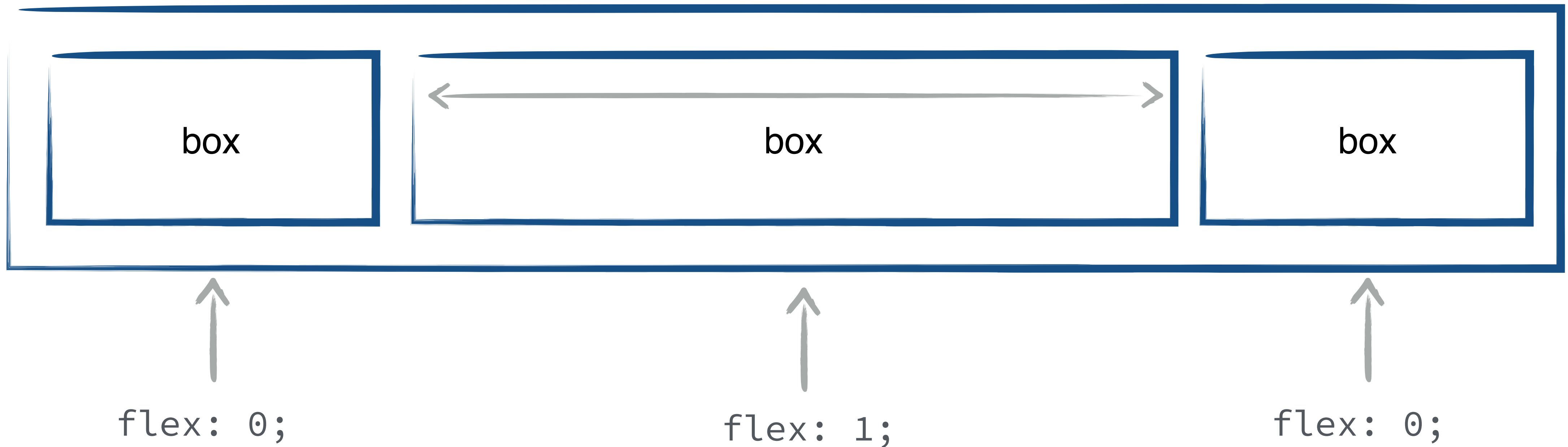
container



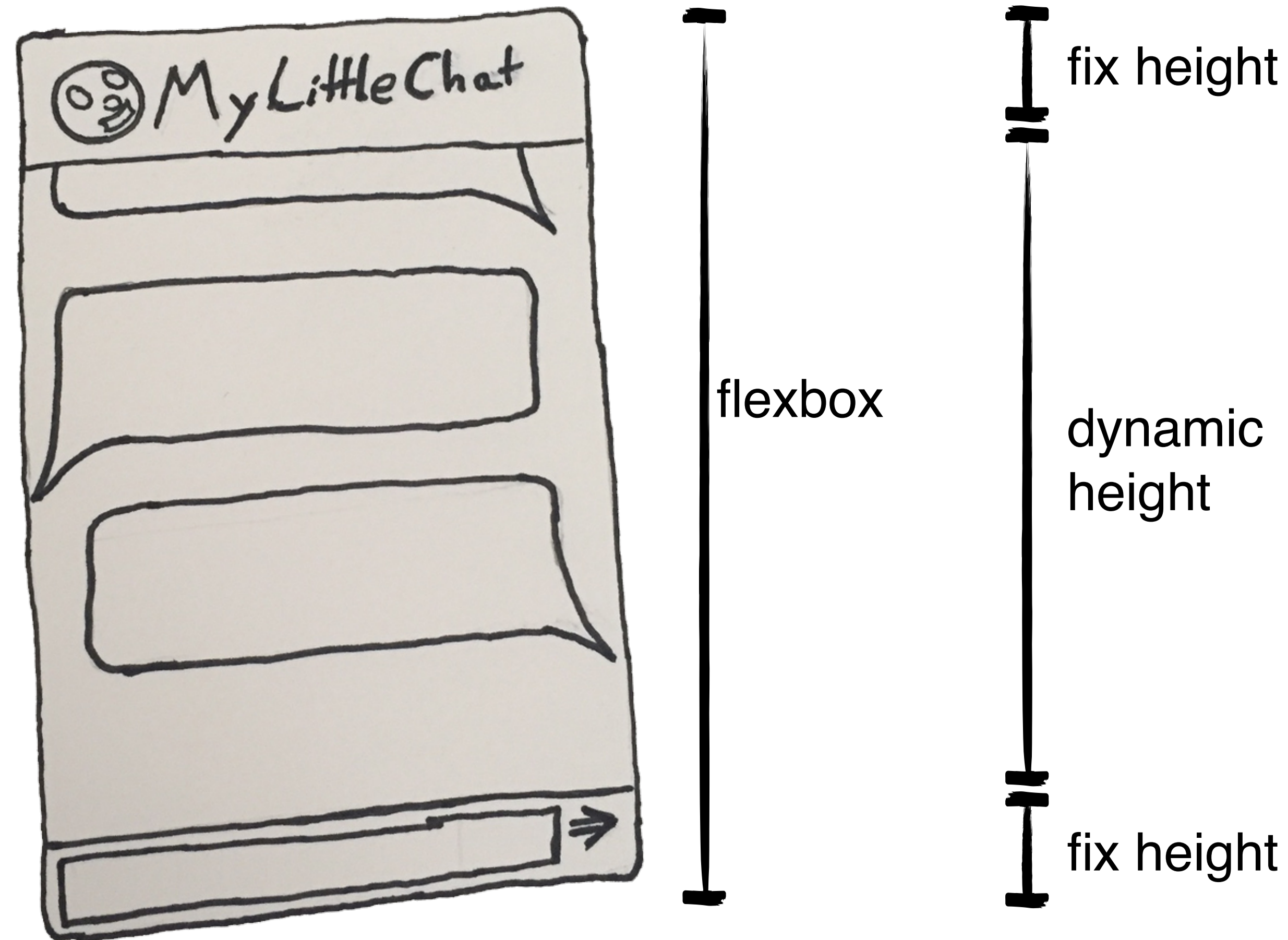
`flex-direction: column;`

Flexbox

container



Application Basic Layout

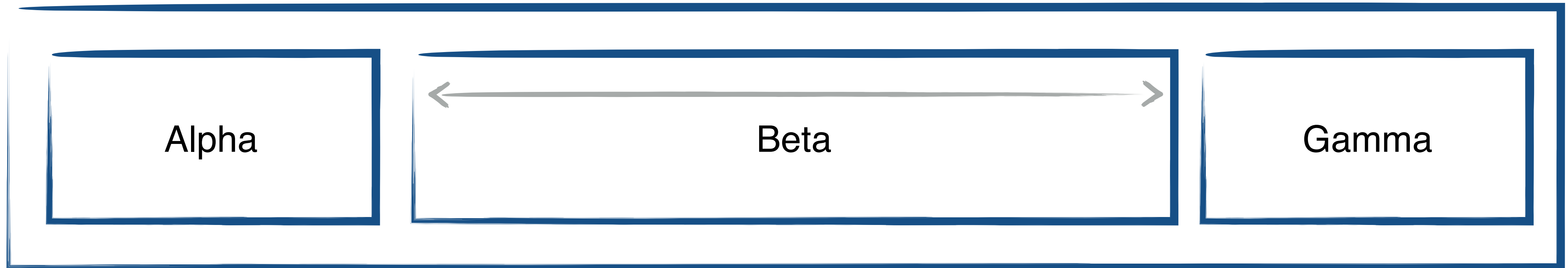


Using Flex Box in Polymer

- Polymer defines several layout classes
 - `layout`: basic class that must be set
 - `horizontal`: layout content in a row
 - `vertical`: layout content in a column
 - `flex`: content will fill space

Polymer Layout

```
<div class="horizontal layout">  
  <div>Alpha</div>  
  <div class="flex">Beta</div>  
  <div>Gamma</div>  
</div>
```





Chat Bubble

Chat Bubble

Sender

This is the chat message. The size of the bubble should fit to this message. Next to normal text emojis can be part of a message: 😊

5 min ago

Chat Bubble

depends on parent width

depends on text length

Sender

This is the chat message. The size of the bubble should fit to this message. Next to normal text emojis can be part of a message: 😊

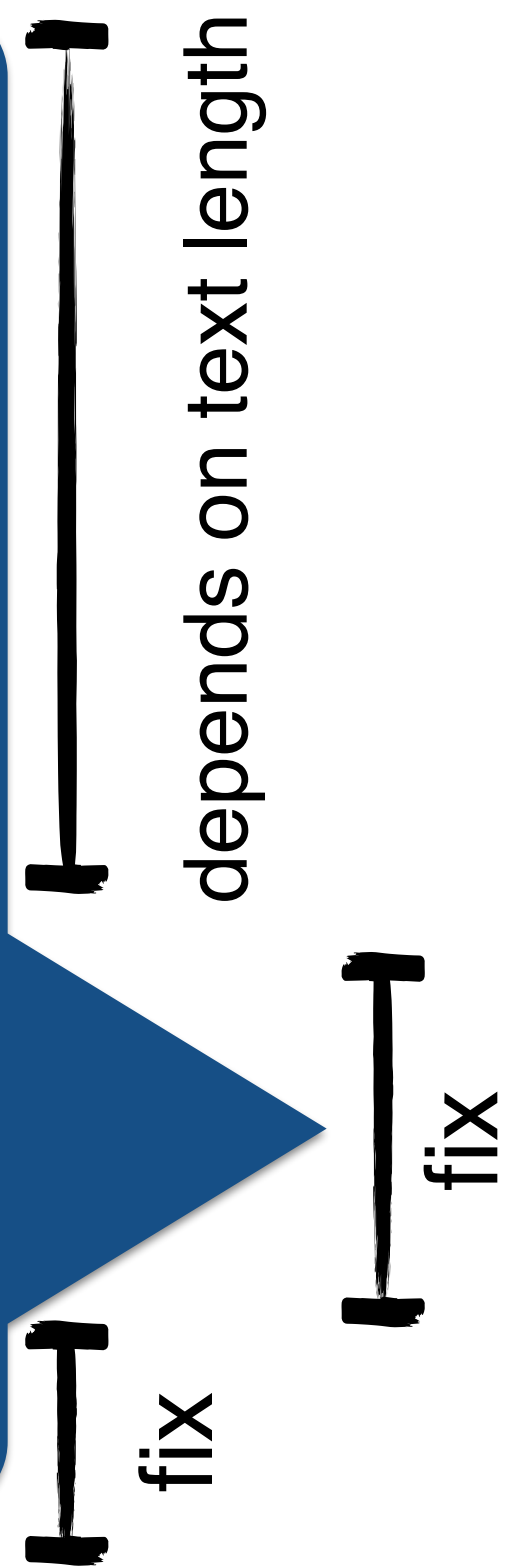
5 min ago

Chat Bubble

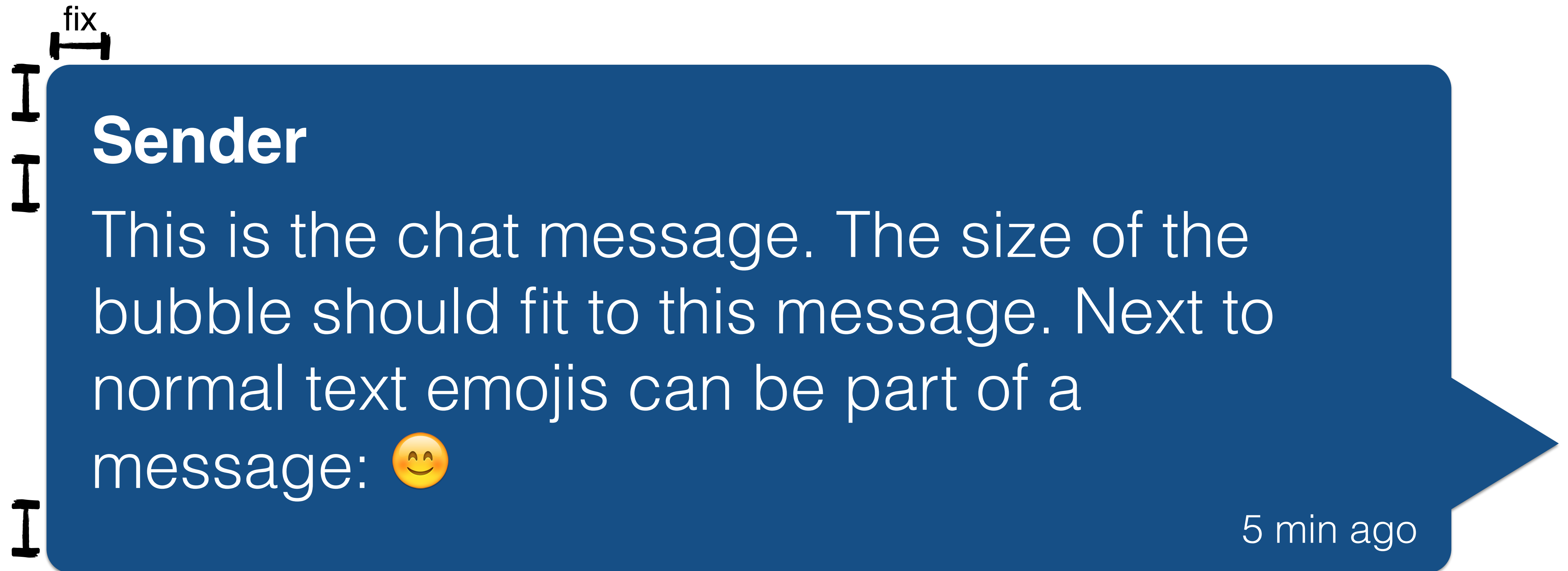
Sender

This is the chat message. The size of the bubble should fit to this message. Next to normal text emojis can be part of a message: 😊

5 min ago



Chat Bubble



Chat Bubble



Chat Bubble

```
<div id="bubble"><div>
```

```
#bubble {  
  background-color: #F2F2F2;  
  border-radius: 5px;  
}
```

Chat Bubble



Chat Bubble

```
<div id="bubble"><div>
```

```
#bubble {  
  background-color: #F2F2F2;  
  border-radius: 5px;  
  padding: 2px 12px 2px 12px;  
}
```

Chat Bubble



CSS ::after Selector

- The `::after` selector inserts something after the content of each selected element(s).
- Use the `content` property to specify the content to insert.

```
p::after {  
  content: " - Remember this";  
}
```

Chat Bubble

```
#bubble::after {  
  background-color: #F2F2F2;  
  display: block;  
  position: absolute;  
  content: "\00a0";  
  height: 16px;  
  width: 20px;  
  bottom: 11px;  
  transform: rotate(29deg) skew(-35deg);  
}
```

Chat Bubble Web Component

```
<dom-module id="chat-bubble">

  <style>
    #bubble { . . . }
  </style>

  <template>
    <div id="bubble"> . . . </div>
  </template>

</dom-module>

<script>
  Polymer({
    is: "chat-bubble"
  });
</script>
```

Chat Bubble Attributes

- Sender & message should be dynamic
- Can be defined as attributes of the component
- Use Polymer properties

```
properties: {  
  text: String,  
  sender: String  
}
```

Problem



configure Bubble with CSS

- We can specify 2 separate CSS classes: **me** and **you**

```
#bubble {  
  . . .  
}
```

```
.me {  
  . . .  
}
```

```
.you {  
  . . .  
}
```

change CSS class

- We can dynamically change the CCS class in JS code

- Access bubble div in Polymer

```
var bubbleDiv = this.$.bubble;
```



- Mutate class list

```
bubbleDiv.classList.remove('me');
```

```
bubbleDiv.classList.add('you');
```

Property Observer

- Polymer provides observers for properties

```
me: {  
  type: Boolean,  
  observer: '_senderChanged'}
```

- Observer will be called whenever property values changes

```
_senderChanged: function (newValue, oldValue) {  
  this.updateStyleClass(newValue);  
}
```


use boolean property

- The initial boolean property value reflects if a HTML attribute is set

„me“ property == true

`<chat-bubble me></chat-bubble>`

„me“ property == false

`<chat-bubble></chat-bubble>`



Chat View



Chat View

- Should contain a list of chat bubbles
- Scrollable
- based on data array (JSON)

Polymer Iron

- List of low level components
- Define behavior
- Don't define theme / skin

iron-iconset	Represents a set of icons
iron-iconset-svg	Manages a set of svg icons
iron-image	An image-displaying element with lots of convenient features
iron-input	An input element with data binding
iron-jsonp-library	Loads jsonp libraries
iron-list	Displays a virtual, 'infinite' scrolling list of items
iron-localstorage	Provides access to local storage
iron-media-query	Lets you bind to a CSS media query
iron-menu-behavior	Provides accessible menu behavior
iron-meta	Useful for sharing information across a DOM tree

iron-list

- Defines a list that can be scrolled
- Can contain thousands of elements. Only visible area is rendered
- Define a template for each cell

iron-list

```
<iron-list items="[[data]]" as="item">  
  <template>  
    <chat-bubble text="[[item.message]]" ...></chat-bubble>  
  </template>  
</iron-list>
```

provide dummy data

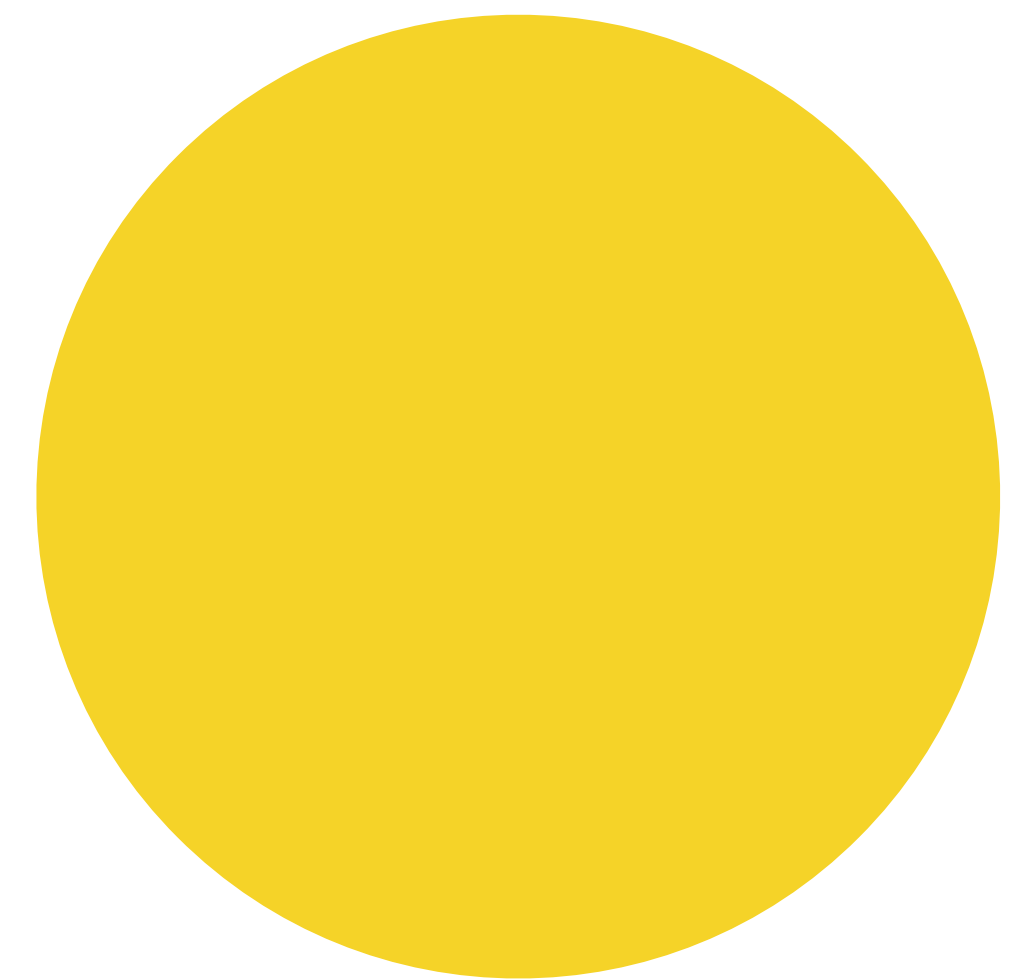
- Polymer provides Iron Element to load JSON data

```
<iron-ajax url="data.json" last-response="{{messages}}" auto></iron-ajax>
```




Application

Header



Centered Text

- The name should be centered
- Layout should not be effected by other nodes

```
.centered {  
  top: 18px;  
  left: 0;  
  position: absolute;  
  width: 100%;  
}
```

parent of text

```
.title {  
  font-size: 22px;  
  text-align: center;  
}
```

text

Rounded Image

- Avatar image should be rounded

```
.avatar {  
  width: 64px;  
  height: 64px;  
  border-radius: 32px;  
  border-style: solid;  
  border-width: 1px;  
  border-color: lightgrey;  
}
```

Dynamic content

- Add some Polymer properties

```
Properties: {  
  title: String,  
  state: String  
}
```

- Use them inline

```
<div class="centered vertical layout">  
  <div class="title">{{title}}</div>  
  <div class="subtitle">{{state}}</div>  
</div>
```



Application

Footer



Custom Textbox

- `<input>` can be styled with CSS
- Polymer provides a behavior to add data binding to a normal input

```
<input is="iron-input" value="{{message::input}}">
```

Custom Button

- Call a internal web component function when the button is clicked

```
<button on-click="sendMessage">send</button>
```

```
Polymer({  
  is: "input-view",  
  sendMessage: function (e) {...}  
});
```

Polymer events

- Web Components can fire events

```
this.fire('message-send', {message: this.message});
```

- Observer can be added

```
<input-view on-message-send="send"></input-view>
```

- Hint: trigger a resize for iron-list when

content changes: `this.$.itemsList.fire('resize');`





























Emoji

Support



About Emojis

- All emojis are defined by a a unicode character

1	U+1F600						
2	U+1F601						
3	U+1F602						
4	U+1F603						
5	U+1F604						

Emojis in HTML

- Most OS provide a font to render emojis
- As long as the browser supports unicode and your files are encoded in unicode you can use them

Parse Emojis

- Cool lib with word - emoji mapping can be found here:
`github.com/muan/emojilib`
- A Polymer Emoji selector can be found here:
`github.com/notwaldorf/emoji-selector`

Emojis in the chat

```
var converted = '';  
var words = input.split(' ');  
for (var i = 0; i < words.length; i++) {  
    var emoji = getMeAnEmoji(words[i]);  
    if(emoji != '') {  
        converted = converted + ' ' + emoji;  
    } else {  
        converted = converted + ' ' + words[i];  
    }  
}
```



Adding a

server



Dolphin Platform

Dolphin Platform

- Enterprise Framework with Presentation Model pattern
- Server centric logic
- Spring and JavaEE integration

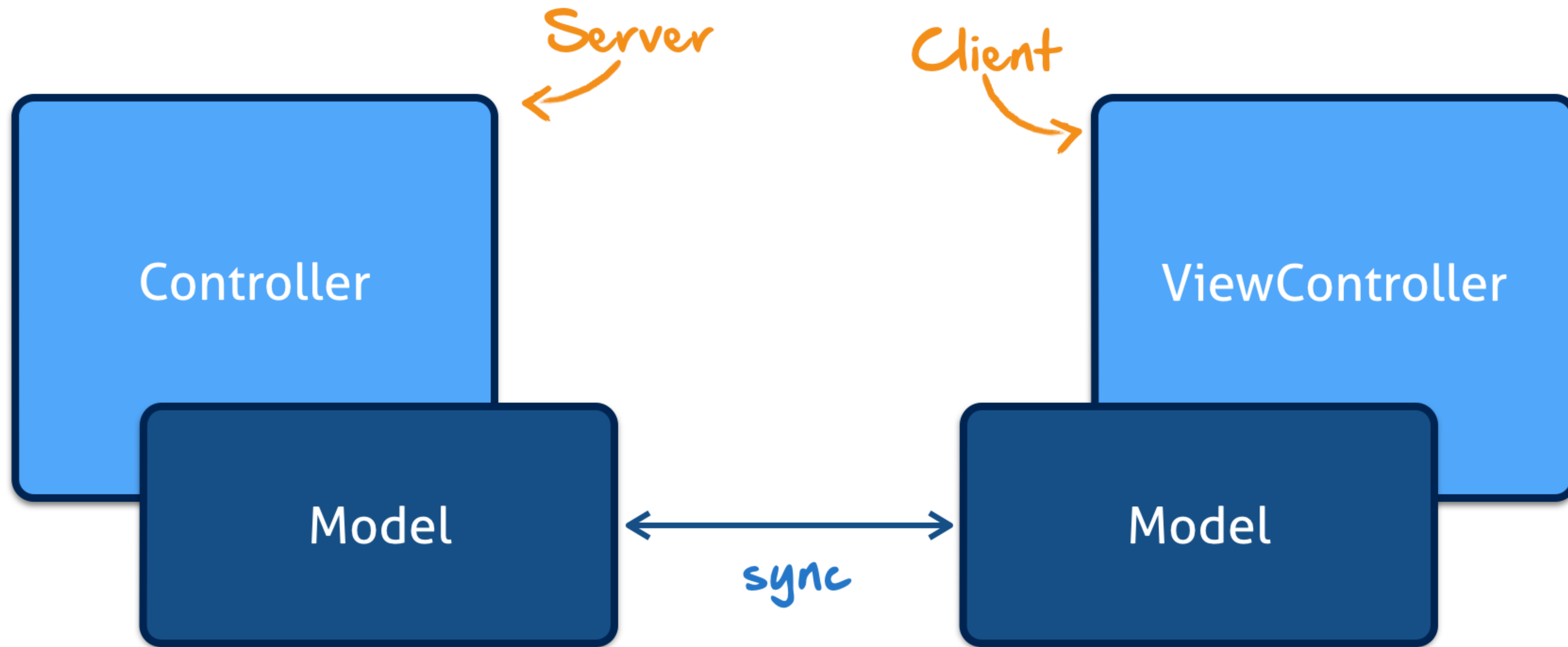
Dolphin Platform

- JavaFX, AngularJS & Google Polymer client APIs



... more will come

Dolphin Platform



Model

- Defined in Java
- Can be hierarchical
- Fully observable
- Collection support

Model

```
@DolphinBean
```

```
public class MyModel {
```

```
    private Property<String> name;
```

```
    public Property<String> nameProperty() {return name;}
```

```
}
```

*This is NOT a JavaFX
Property*



*Dolphin Platform property can be
used on client and server*



Controller

- Controller is defined on the server
- Will be managed by the container (Spring, JavaEE, ...)
- Full CDI support

Controller

```
@DolphinController
public class MyController {

    @DolphinModel
    private MyModel model;

    @PostConstruct
    public void init() {
        model.nameProperty().onChange(e -> System.out.println(„CHANGE“));
    }
}
```

Inject the model

Use container features

View

- The smallest part ;)
- Use the complete UI Toolkit power
- Simply bind your view to the model
- Trigger controller actions

View

```
<dom-module id="my-view">  
  <template>  
    <paper-input label="Task name" value="{{model.newItemText}}"></paper-input>  
  </template>  
</dom-module>
```

Dolphin Platform model



```
<script>  
  Polymer({  
    is: "my-view",  
    behaviors: [clientContext.createBehavior('MyController')]  
  });  
</script>
```

*Dolphin Platform provides
Polymer behavior*

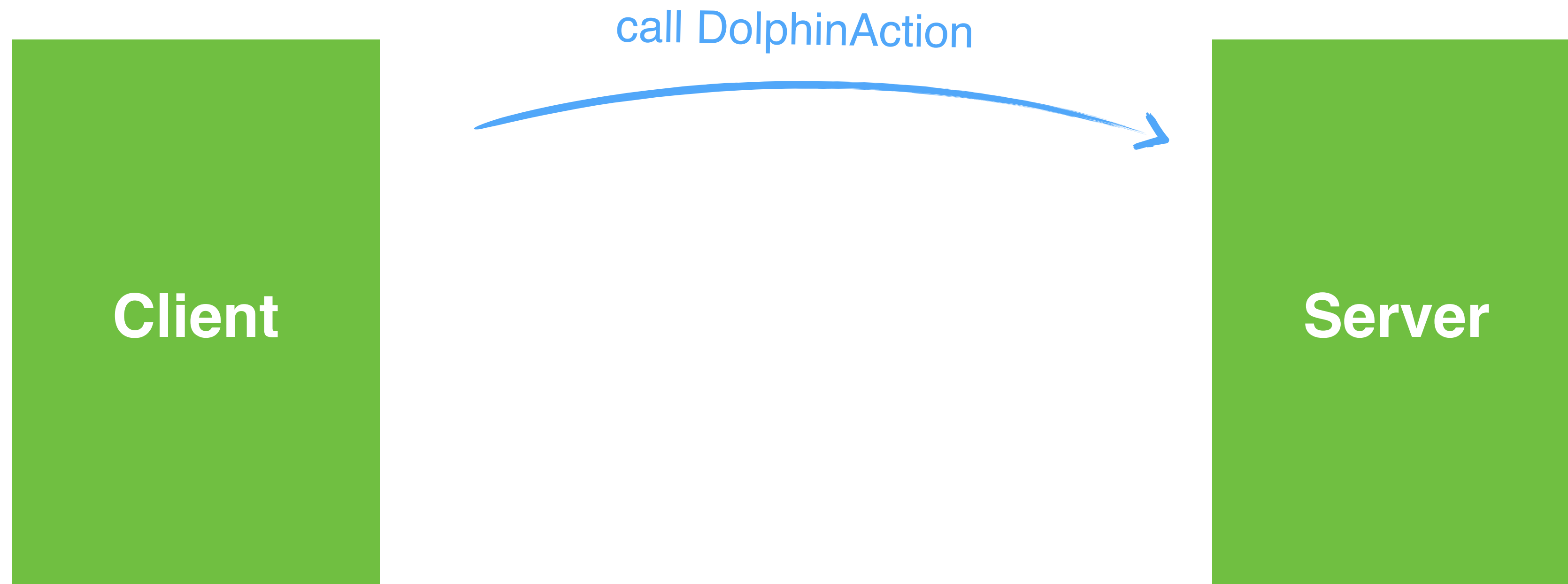


Server controller

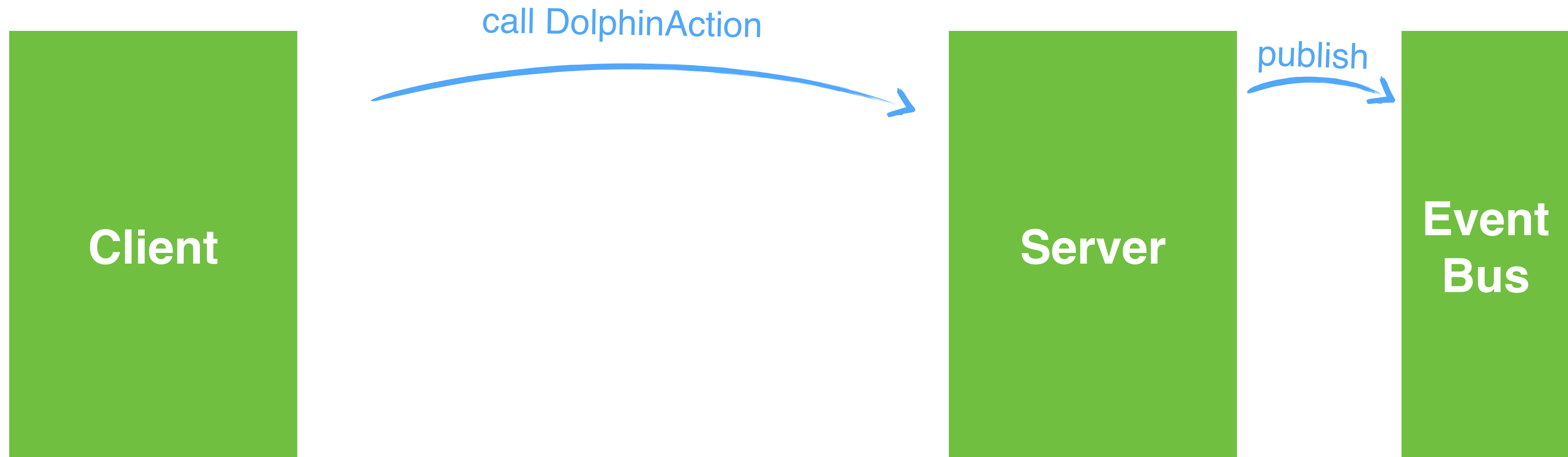


Chat App

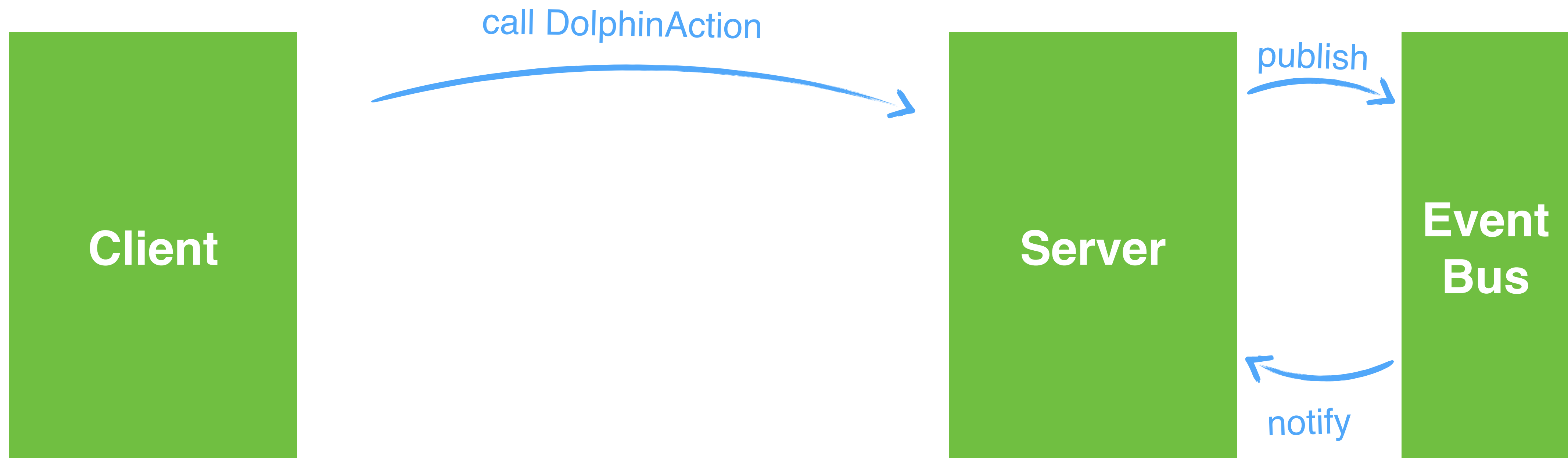
Sending a message



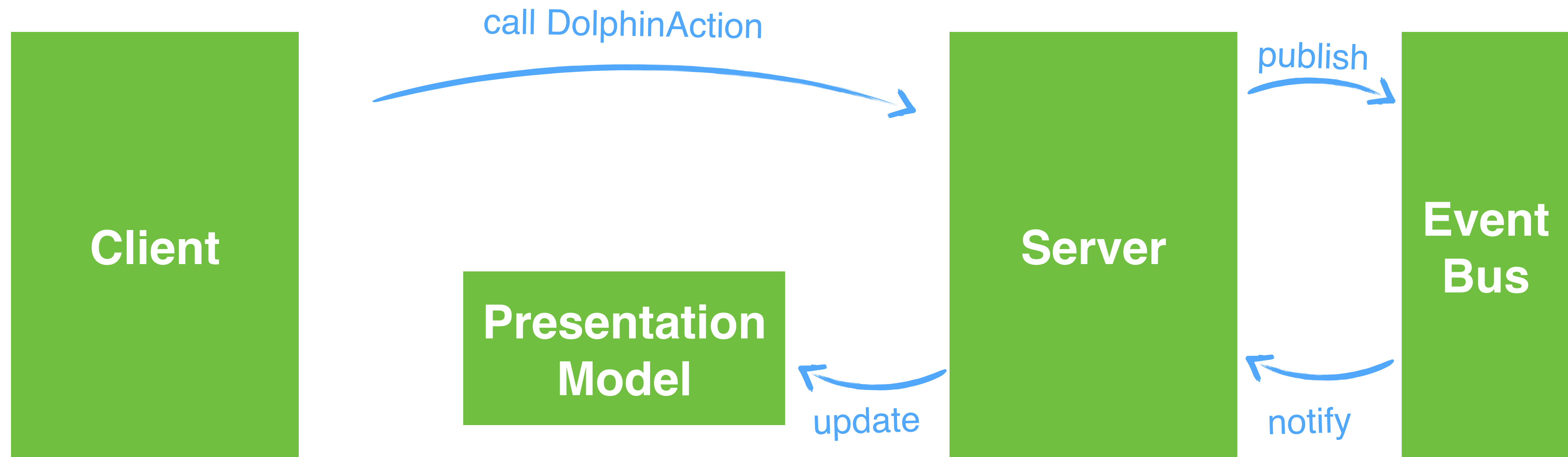
Sending a message



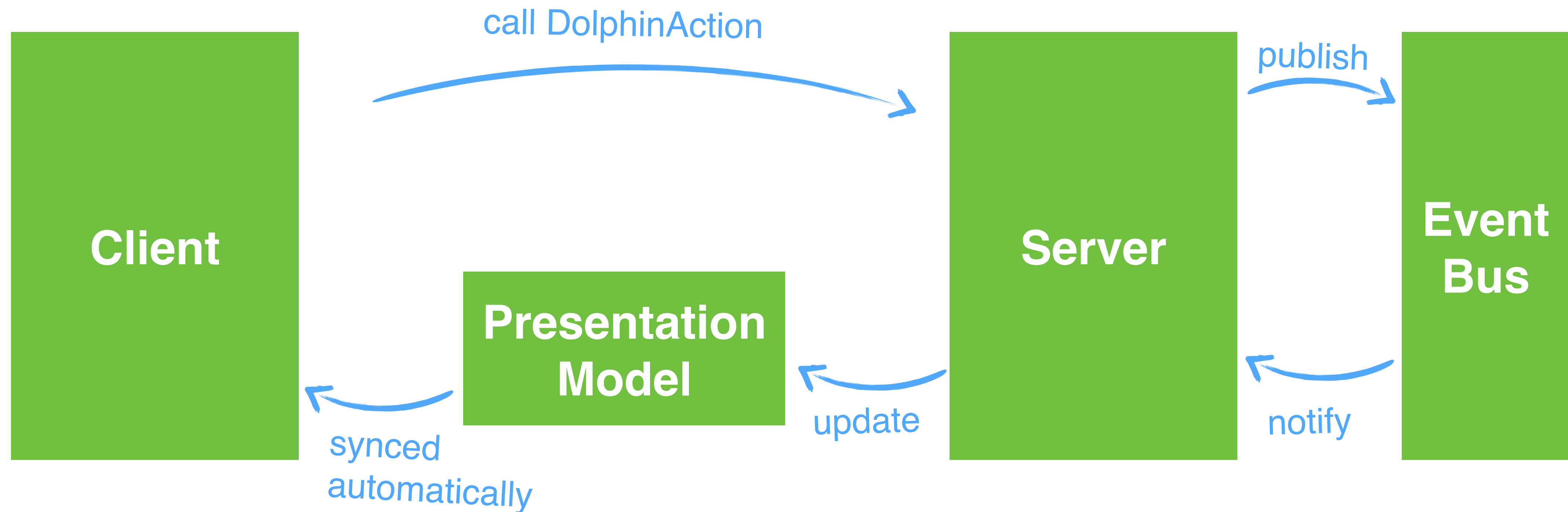
Sending a message



Sending a message



Sending a message

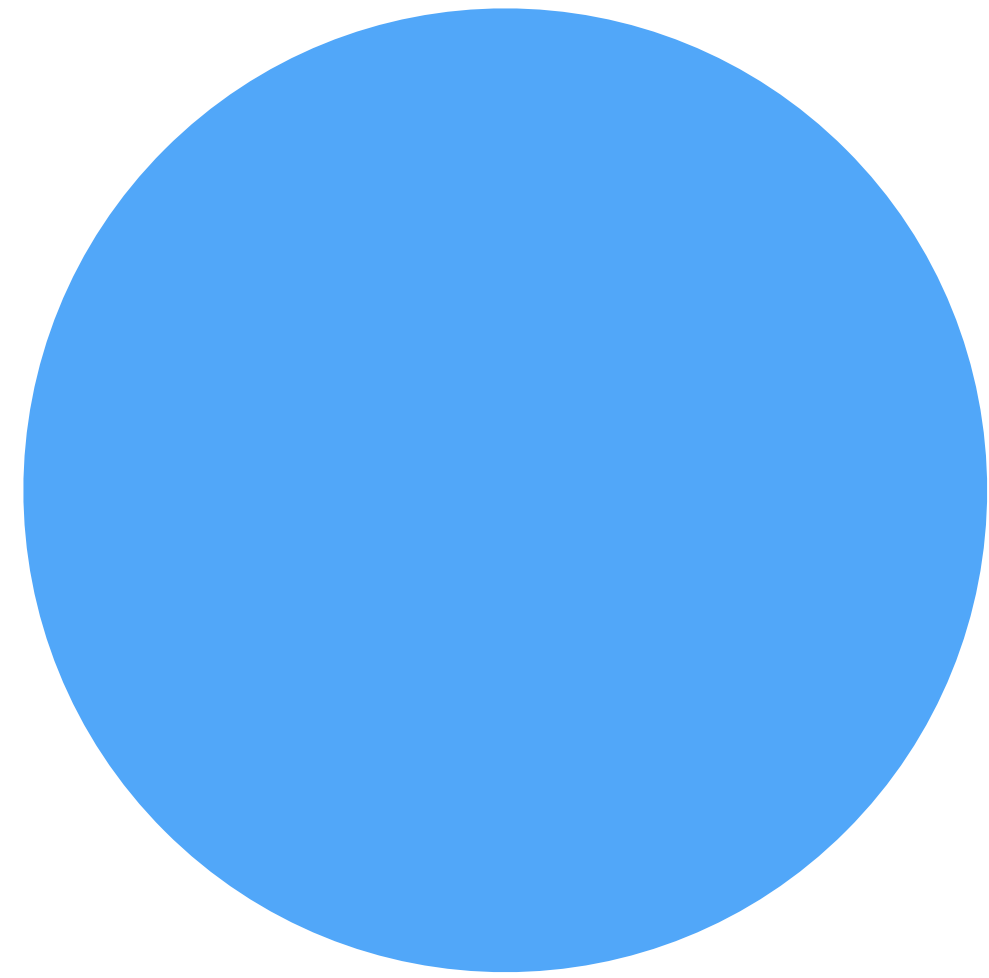
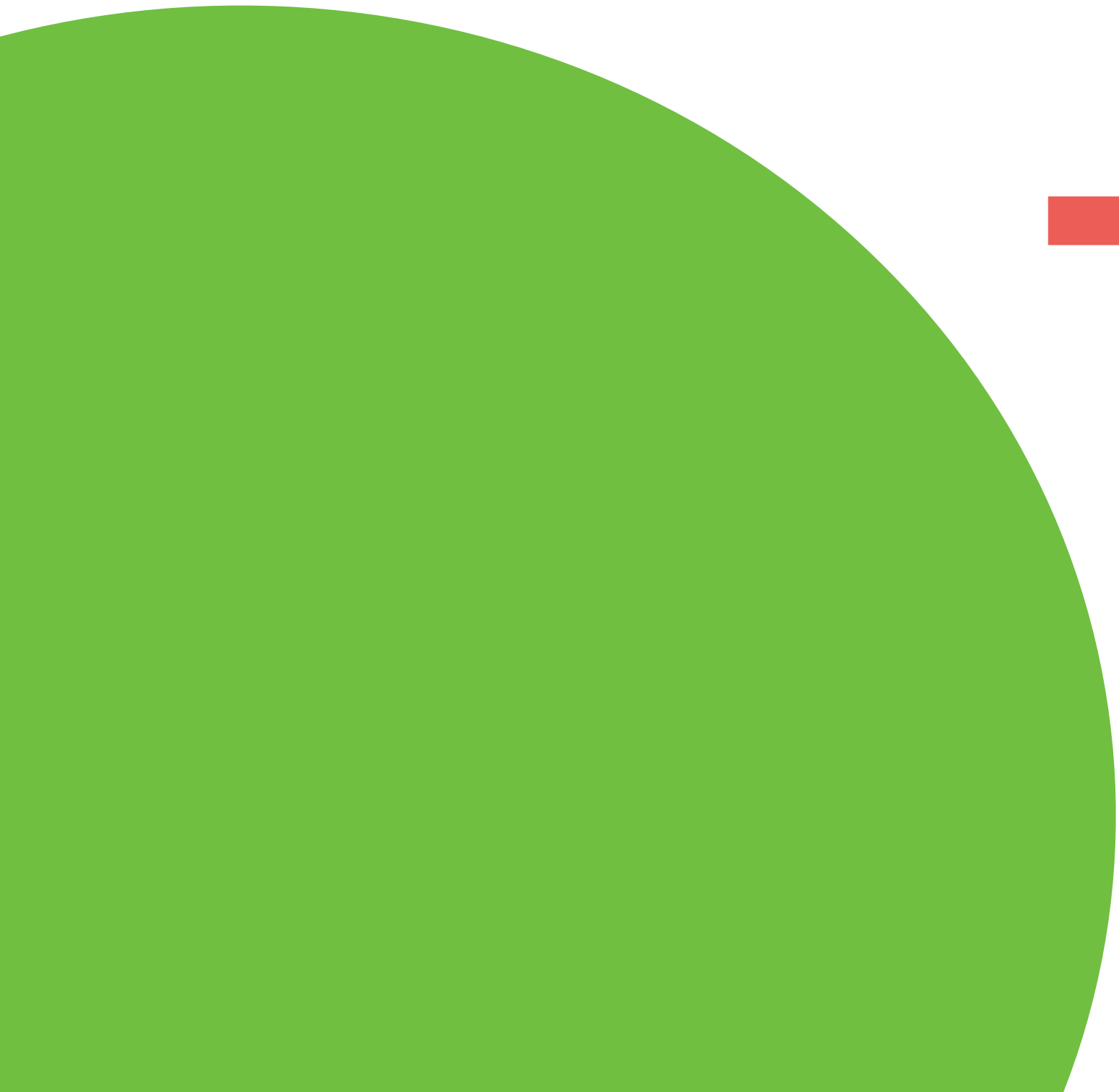


client code diff

```
27 27 <chat-header title="Michael" state="online"></chat-header>
28 28 <iron-list id="itemsList" class="container" items="{{messages}}" as="item">
29 29   <template>
30 30     <chat-bubble me="{{item.me}}" text="{{item.message}}" sender="{{item.sender}}">
31 31     </template>
32 32   </iron-list>
33 33   <input-view on-message-send="send"></input-view>
34 34 </div>
35 35 </template>
36 36 </dom-module>
37 37 <script>
38 38   Polymer({
39 39     is: "chat-view",
40 40     properties: {
41 41       messages: {
42 42         type: Array,
43 43         value: function () {
44 44           return [];
45 45         }
46 46       },
47 47     },
48 48     observers: [
49 49       "messagesChanged(messages.*)"
50 50     ],
51 51     messagesChanged: function (changeRecord) {
52 52       this.$.itemsList.fire('resize');
53 53       console.log("DONE");
54 54     },
55 55     addEmoji: function (input) {
56 56       var converted = '';
57 57       var words = input.split(' ');
58 58       for (var i = 0; i < words.length; i++) {
59 59         var emoji = getMeAnEmoji(words[i]);
60 60         if(emoji != '') {
61 61           converted = converted + ' ' + emoji;
62 62         } else {
63 63           converted = converted + ' ' + words[i];
64 64         }
65 65       }
66 66       return converted;
67 67     },
68 68     send: function (e) {
69 69       console.log("SEND " + e.detail.message);
70 70       this.push('messages', {message: this.addEmoji(e.detail.message), me: true,
71 71     }
72 72   );
73 73 </script>
```

Additional

Topics



Additional Topics

- Behaviors
- Animations
- Custom CSS properties
- DOM manipulation

Additional Topics

- Polymer contains more features
- Explorer the Polymer Catalog

<https://elements.polymer-project.org>

- Read the documentation

<https://www.polymer-project.org/1.0/docs/>

Real World

Applications

Polymer

Starter Kit

Polymer Starter Kit

- Fully functional mini-application
- Defines application structure
- Contains many useful configurations (e.g. for gitignore, jshint, travis)
- Sophisticated build script
- Sidenote: all files contain copyright headers

Polymer Starter Kit

- Minification (js, css, images)
- Inlining of HTML Imports
- Cache Config for Offline Usage
- Local Server (with browser-sync)
- Testing

The background features three large, tilted geometric shapes: an orange trapezoid at the top center, a yellow trapezoid on the left side, and a yellow triangle on the bottom right corner. The text is centered in a red, sans-serif font.

Web Component Tester

Web Component Tester

Installation

```
npm install -g web-component-tester
```

Usage

```
wct
```


Web Component Tester

- Integrates Mocha, Chai, and Sinon
- Runs with Selenium
- Integration with Sauce Labs provided
- By default runs all tests in test/
- Tasks for grunt and gulp available
(though gulp integration is sub-optimal)

Web Component Tester

```
<head>  
  <meta charset="utf-8">  
  <script src="../../bower_components/webcomponentsjs/webcomponents.min.js">  
  </script>  
  <script src="/web-component-tester/browser.js"></script>  
  <link rel="import" href="../../chat-bubble.html">  
</head>
```

web component polyfill

Web Component Tester

```
<head>
  <meta charset="utf-8">
  <script src="../../bower_components/webcomponentsjs/webcomponents.min.js">
  </script>
  <script src="/web-component-tester/browser.js"></script>
  <link rel="import" href="../../chat-bubble.html">
</head>
```

← provided by wct

Web Component Tester

```
<head>
  <meta charset="utf-8">
  <script src="../../bower_components/webcomponentsjs/webcomponents.min.js">
  </script>
  <script src="/web-component-tester/browser.js"></script>
  <link rel="import" href="../../chat-bubble.html"> ← component under test
</head>
```

Web Component Tester

```
<body>
  <test-fixture id="chatBubbleFixture"> ← provided by wct
    <template>
      <chat-bubble></chat-bubble>
    </template>
  </test-fixture>
  ...
```

Web Component Tester

```
<body>
  <test-fixture id="chatBubbleFixture">
    <template>
      <chat-bubble></chat-bubble>
    </template>
  </test-fixture>
  ...
```

web component under test ←

Web Component Tester

```
<script>
  describe('<chat-bubble>', function() {
    it('should show the sender', function(done) {
      var chatBubble = fixture('chatBubbleFixture');
      chatBubble.set('sender', 'Test Sender');
      setTimeout(function () {
        expect(chatBubble.$$('.sender'))
          .to.have.property('textContent')
          .that.contains('Test Sender');
        done();
      });
    });
  });
</script>
```

← create component

Web Component Tester

```
<script>
  describe('<chat-bubble>', function() {
    it('should show the sender', function(done) {
      var chatBubble = fixture('chatBubbleFixture');
      chatBubble.set('sender', 'Test Sender'); ← set property
      setTimeout(function () {
        expect(chatBubble.$$('.sender'))
          .to.have.property('textContent')
          .that.contains('Test Sender');
        done();
      });
    });
  });
</script>
```


Web Component Tester

```
<script>
  describe('<chat-bubble>', function() {
    it('should show the sender', function(done) {
      var chatBubble = fixture('chatBubbleFixture');
      chatBubble.set('sender', 'Test Sender');
      setTimeout(function () {
        expect(chatBubble.$$('.sender'))
          .to.have.property('textContent')
          .that.contains('Test Sender');
        done();
      });
    });
  });
</script>
```

needs to be asynchronous

Web Component Tester

```
<script>
  describe('<chat-bubble>', function() {
    it('should show the sender', function(done) {
      var chatBubble = fixture('chatBubbleFixture');
      chatBubble.set('sender', 'Test Sender');
      setTimeout(function () {
        expect(chatBubble.$$('.sender')) ← DOM-query
          .to.have.property('textContent')
          .that.contains('Test Sender');
        done();
      });
    });
  });
</script>
```

Web Component Tester

```
module.exports = {
  plugins: {
    sauce: {
      browsers: [
        {
          platform: 'Windows 8.1',
          browserName: 'internet explorer',
          version: '11.0'
        }
      ]
    }
  }
};
```

environment configuration

Create reusable
components

Situation

- All components are part of our project
- We can't reuse them in other projects
- The components are not isolated
- Looks like a bad mixup



Wish list

- We want reusable components
- A component should be independent and isolated
- Components can depend on each other / include other custom components
- Use a component as a bower dependency



seed-element

- Blueprint for custom component
- Defines demo, test, and main section
- Web component tester and polylserve integrated

polyserve

- Polymer offers a special web server to create reusable components
- serves project files under `/components/{bower-name}/`
- other component are served from `./bower_components/`

polyserve

- Your component can refer to external components like your component is in the `bower_components` folder
- Simple to create a github repo for one specific reusable component
- Polymer Paper is doing the same



Questions?