stencil

From **Frameworks** to **Compilers**

Manu Mtz.-Almeida

IONIC

Manu Mtz.-Almeida

Senior Software Engineer on the
Ionic Framework Team

@manucorporat

@manucorporat

based in a true story

12 years ago


jQuery

≈


JS


codiqa

some years later…

AngularJS!!!!

DIRECTIVES


ionic


Angular 2!!


ionic

Angular 4, 5, 6, 7…

Vue

Ember

React

Preact

Polymer

ionic *vue*

ionic *react*

ionic _____


AWARD
gifbin.com

The end of

Framework Churn

Frameworks

👍 **Great** for final products

👎 **Terrible** for reusable components

Web Components

Universal Model

Web Component → Angular Component
React Component
Vue Component
Ember Component

Standard → Stable
Future proof

				
HTML `<template>`	26	13	22	8
Shadow DOM	53	in development	63	10.1
Custom Elements	54	in development	63	10.1
ES Modules	61	16	60	10.1

Problem:

WC is a low level API

Solution:

create an abstraction

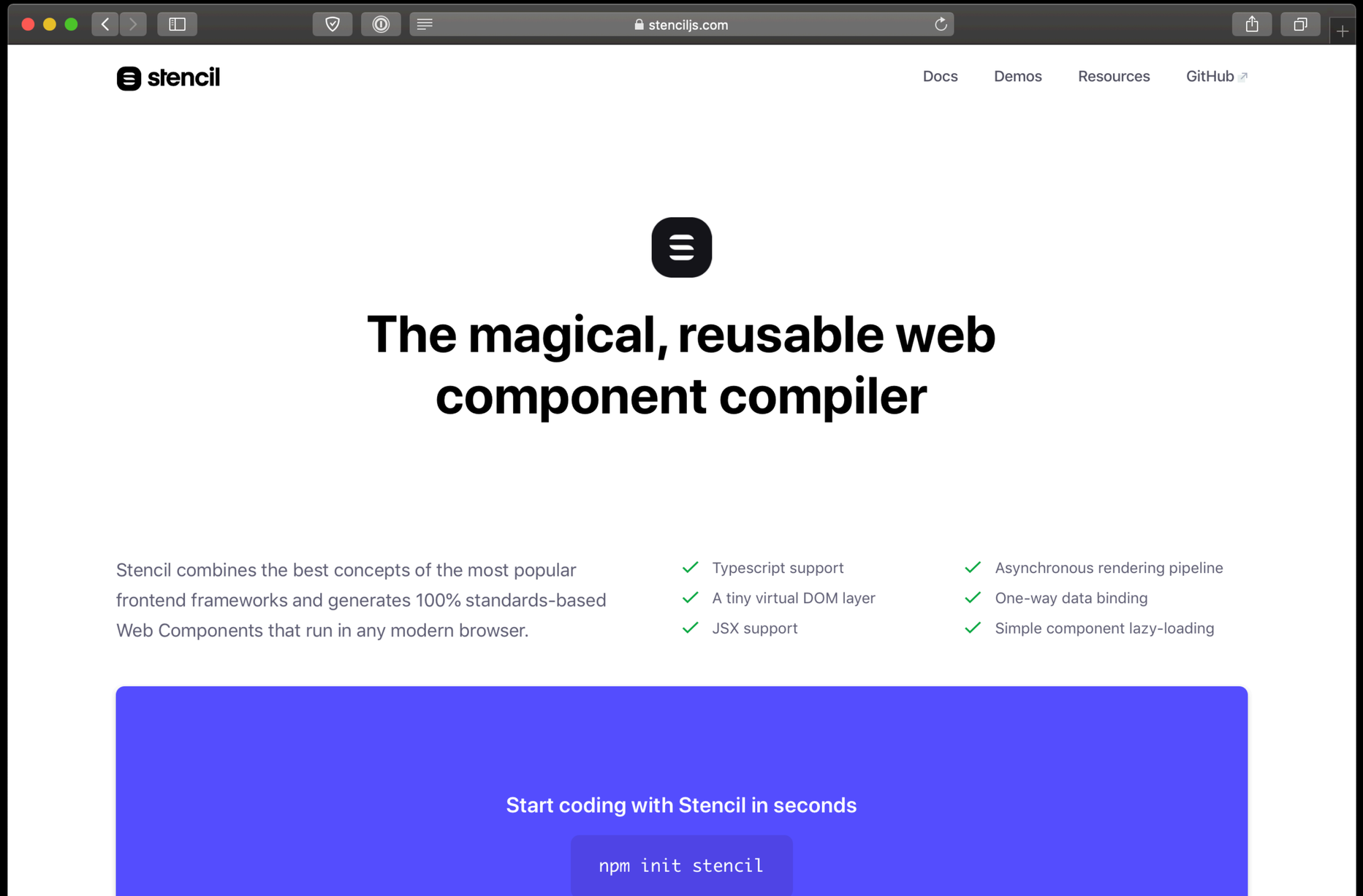
New Problem:

overhead

Unless…

build-time abstraction

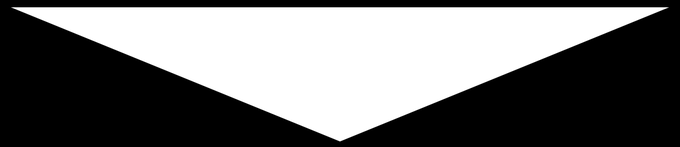
stenciljs.com



stencil?

It’s a build-time abstraction

Ionic Requirements



stencil

Requirements

- **Productivity**
- Raw Web Components with minimal overhead
- Stability
- Scalability
- Performance
- Lazy loading

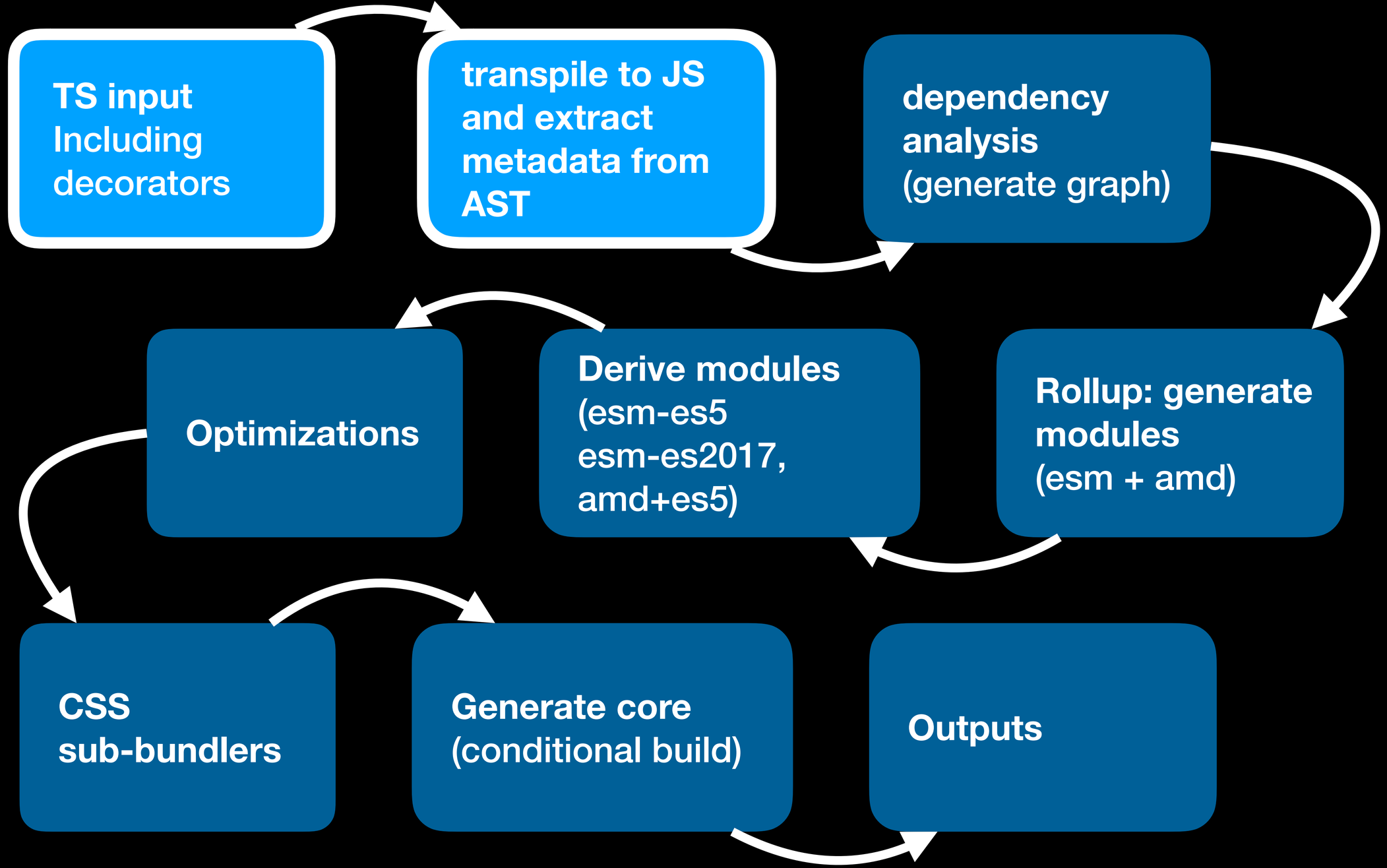
Deprecations → AST transforms

Small API / features → Stability

Standard features → Stability

Compiler-time feature > Run-time feature

Deep Dive


TS input
Including
decorators
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Optimizations
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Rollup: generate
modules
(esm + amd)
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

```tsx
import { Component, Prop } from '@stencil/core';

export type Color = 'primary' | 'secondary' | 'dark';

@Component({
  tag: 'app-home',
  styleUrl: 'app-home.css',
  shadow: true
})
export class AppHome {

  /**
   * El color theme del componente.
   */
  @Prop() color: Color = 'primary';

  /**
   * Anchura del componente
   */
  @Prop() width?: number;

  /**
   * Un objeto que no es un atributo html.
   */
  @Prop() obj?: Object;

  render() {
    return (
      <div class={{
        'contenedor': true,
        [`color-${this.color}`]: true
      }}>
        <slot></slot>
      </div>
    );
  }
}
```

Tag-name: “app-home”
Style: “app-home.css”
Shadow: true

Property: “color”
Type: Color => union type
Primtive: string | undefined
Docs

Property: “width”
Type: width => number
Primitive: number
Docs

```tsx
import { Component, Prop } from '@stencil/core';

export type Color = 'primary' | 'secondary' | 'dark';

@Component({
  tag: 'app-home',
  styleUrl: 'app-home.css',
  shadow: true
})
export class AppHome {

  /**
   * El color theme del componente.
   */
  @Prop() color?: Color = 'primary';

  /**
   * Anchura del componente
   */
  @Prop() width = 200;
```

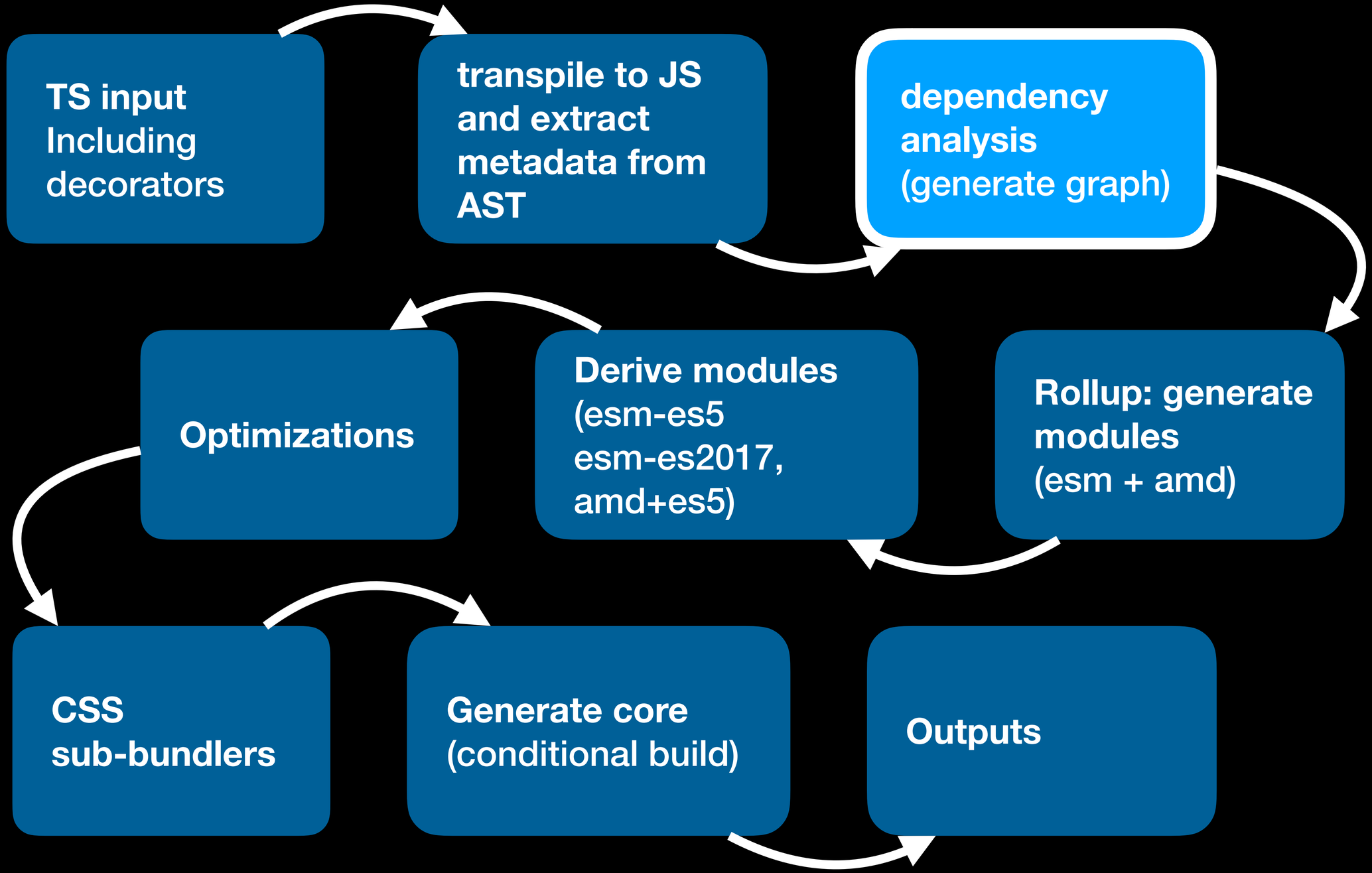
render() used
class object used
Slot used

```
    * Anchura del componente
    */
  @Prop() width?: number;

  /**
    * Un objeto que no es un atributo html.
    */
  @Prop() obj?: Object;

  render() {
    return (
      <div class={{
        'contenedor': true,
        [`color-${this.color}`]: true
      }}>
        <slot></slot>
      </div>
    );
  }
}
```

lowering features

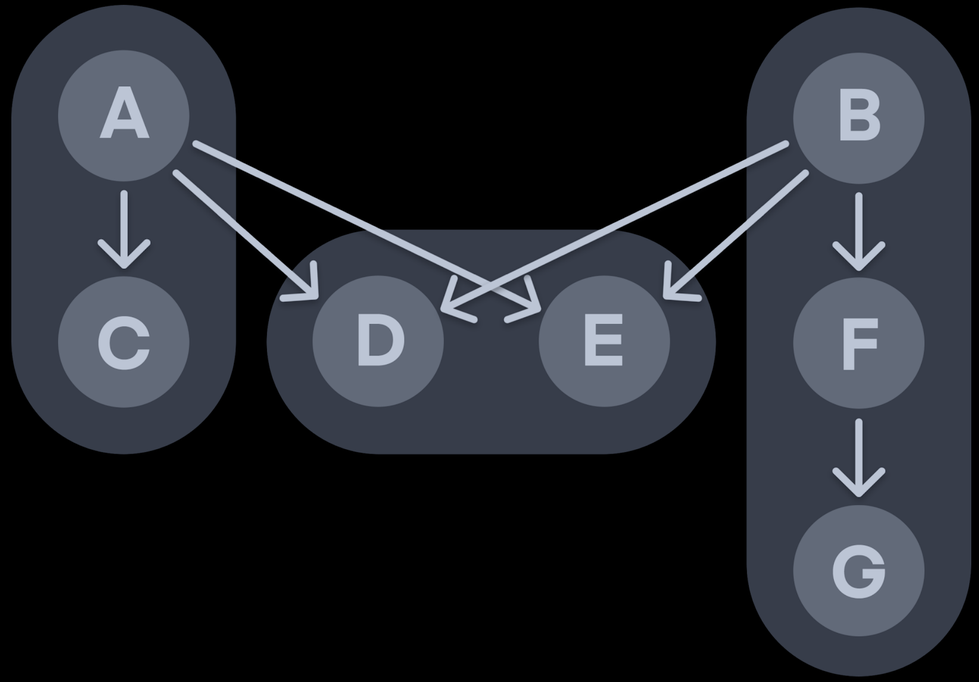
```ts
/**
 * When using a router, it specifies the transition direction when navigating to
 * another page using `href`.
 */
@Prop() routerDirection?: RouterDirection;

/**
 * Contains a URL or a URL fragment that the hyperlink points to.
 * If this property is set, an anchor tag will be rendered.
 */
@Prop() href?: string;

/**
 * The button shape.
 * Possible values are: `"round"`.
 */
@Prop({ reflectToAttr: true }) shape?: 'round';

/**
 * The button size.
 * Possible values are: `"small"`, `"default"`, `"large"`.
 */
@Prop({ reflectToAttr: true }) size?: 'small' | 'default' | 'large';

/**
 * If `true`, activates a button with a heavier font weight.
 */
```

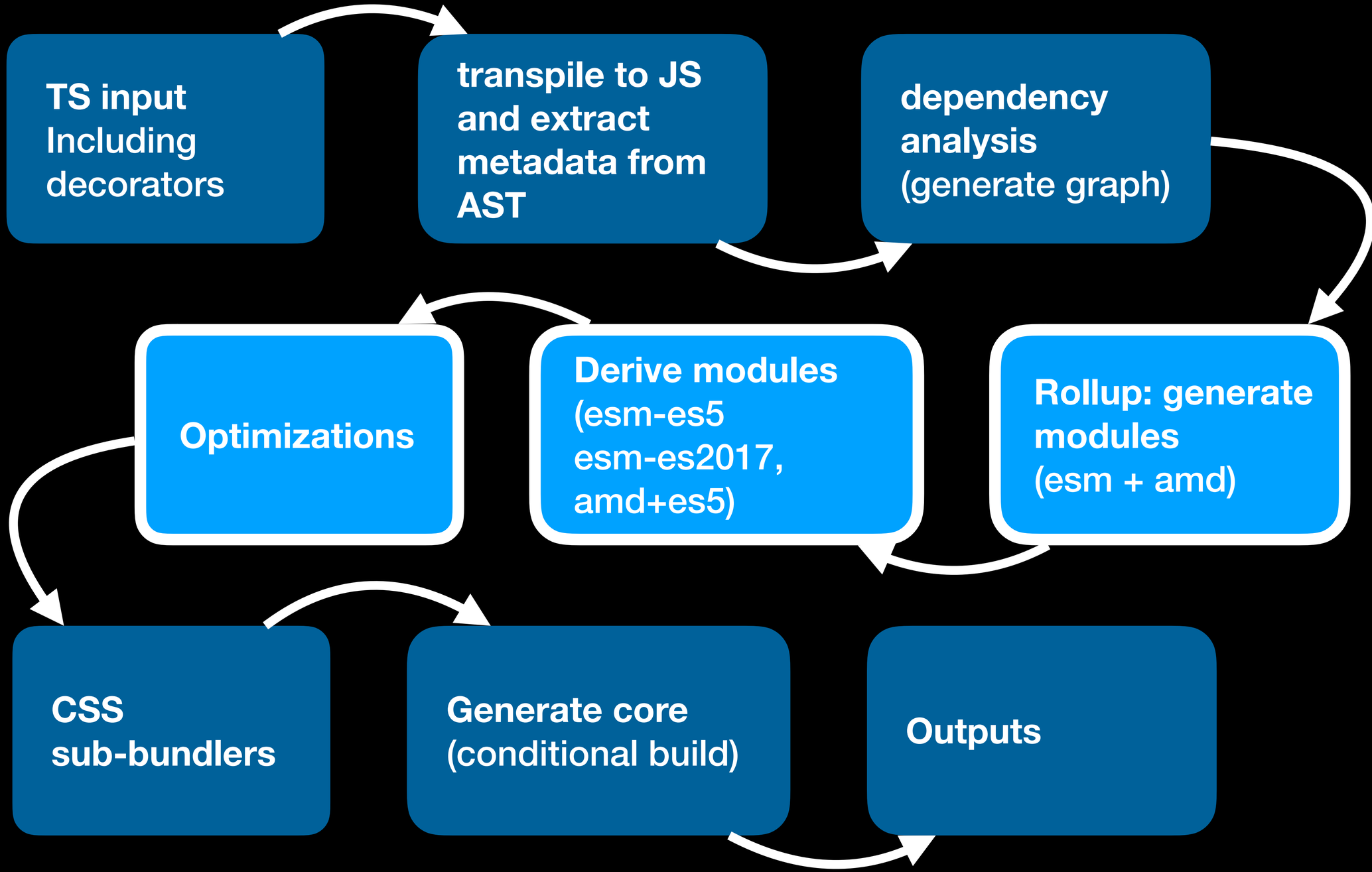
```
            "f": 193
        },
        "href": {
            "f": 1
        },
        "keyFocus": {
            "f": 16
        },
        "mode": {
            "f": 1
        },
        "routerDirection": {
            "attr": "router-direction",
            "f": 1
        },
        "shape": {
            "f": 129
        },
        "size": {
            "f": 129
        },
        "strong": {
            "f": 4
        },
        "type": {
            "f": 1
```


TS input
Including
decorators
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Optimizations
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Rollup: generate
modules
(esm + amd)
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

graph analysis

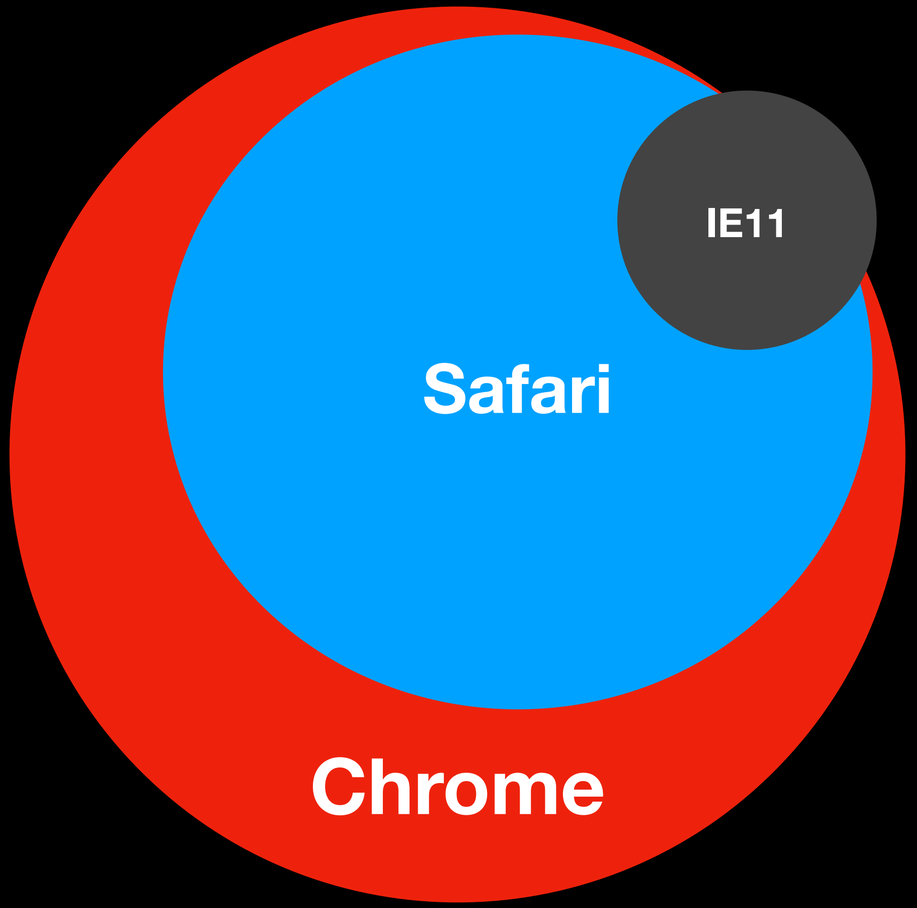
- **The compiler generates a dependency graph based in a static analysis**
- Components are optimized further by groups them optimally
- HTTP network requests are reduced



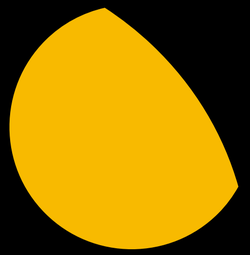

TS input
Including
decorators
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Optimizations
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Rollup: generate
modules
(esm + amd)
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

multiple targets

- **Modern browsers**
  - ES Modules nativos
  - No polyfills
- **Legacy**
  - AMD
  - Polyfills

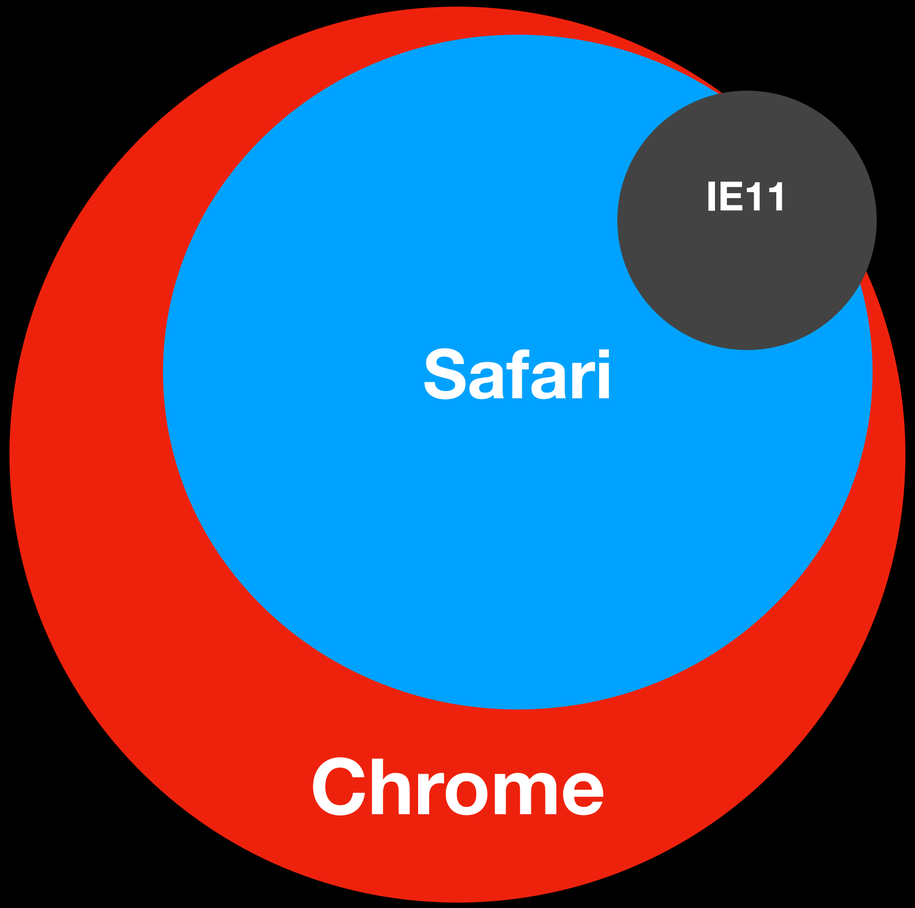


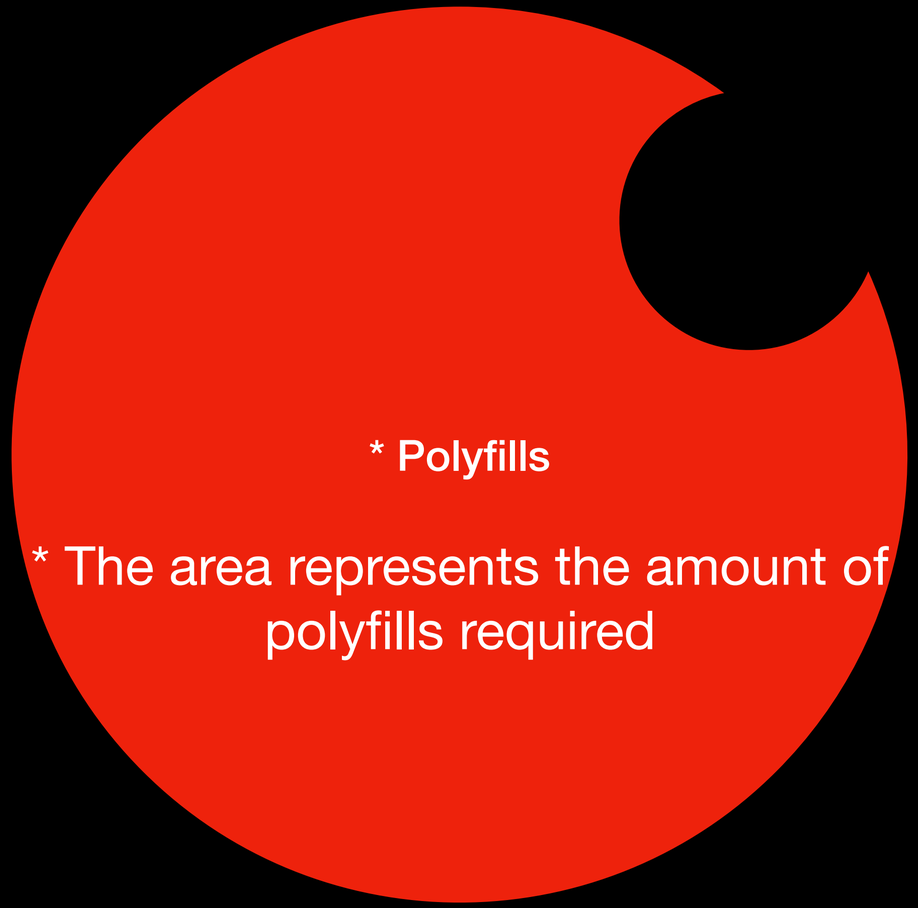
APIs developers can use



* The area represents the amount of APIs available for each browsers

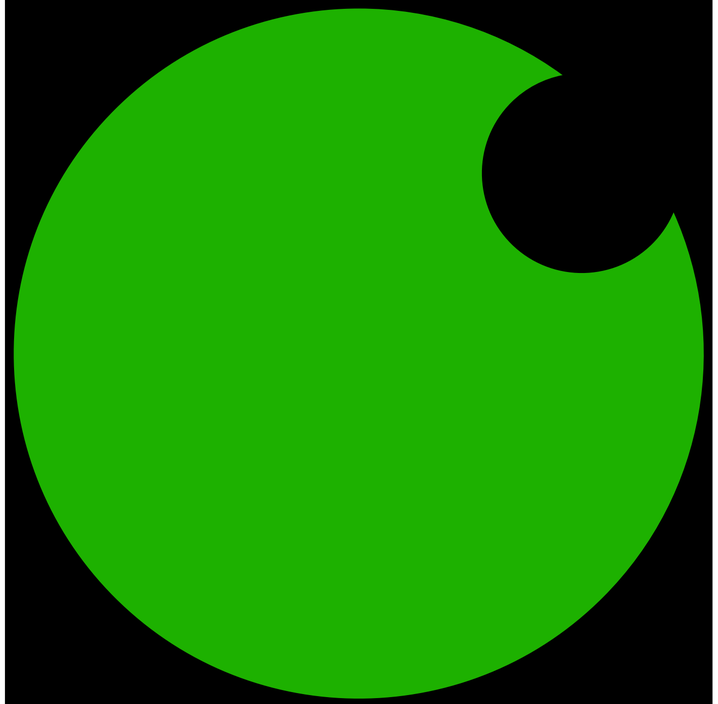
#1: Polyfill for everyone





#1: Polyfill for everyone

Polyfills for **IE11**

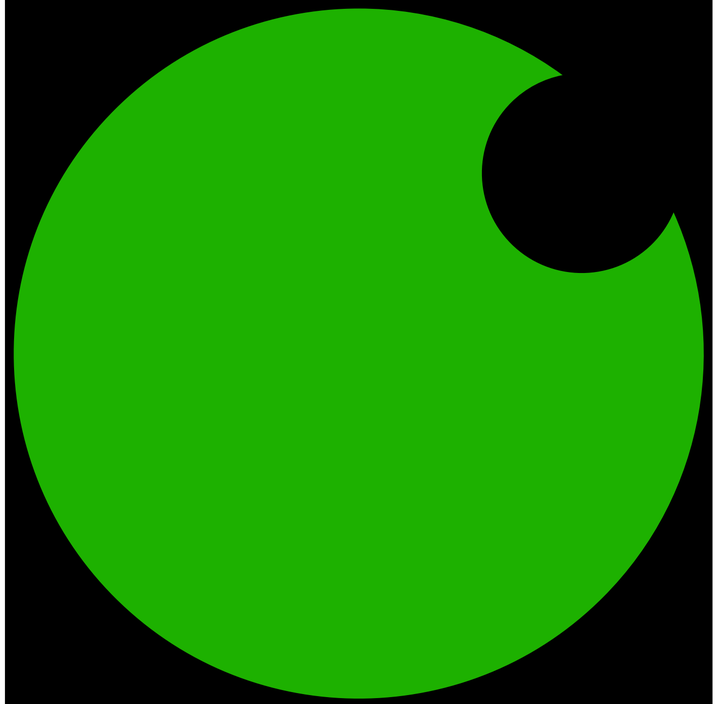


Polyfills for **Firefox**

Polyfills for **Chrome**

#2: Polyfills on demand

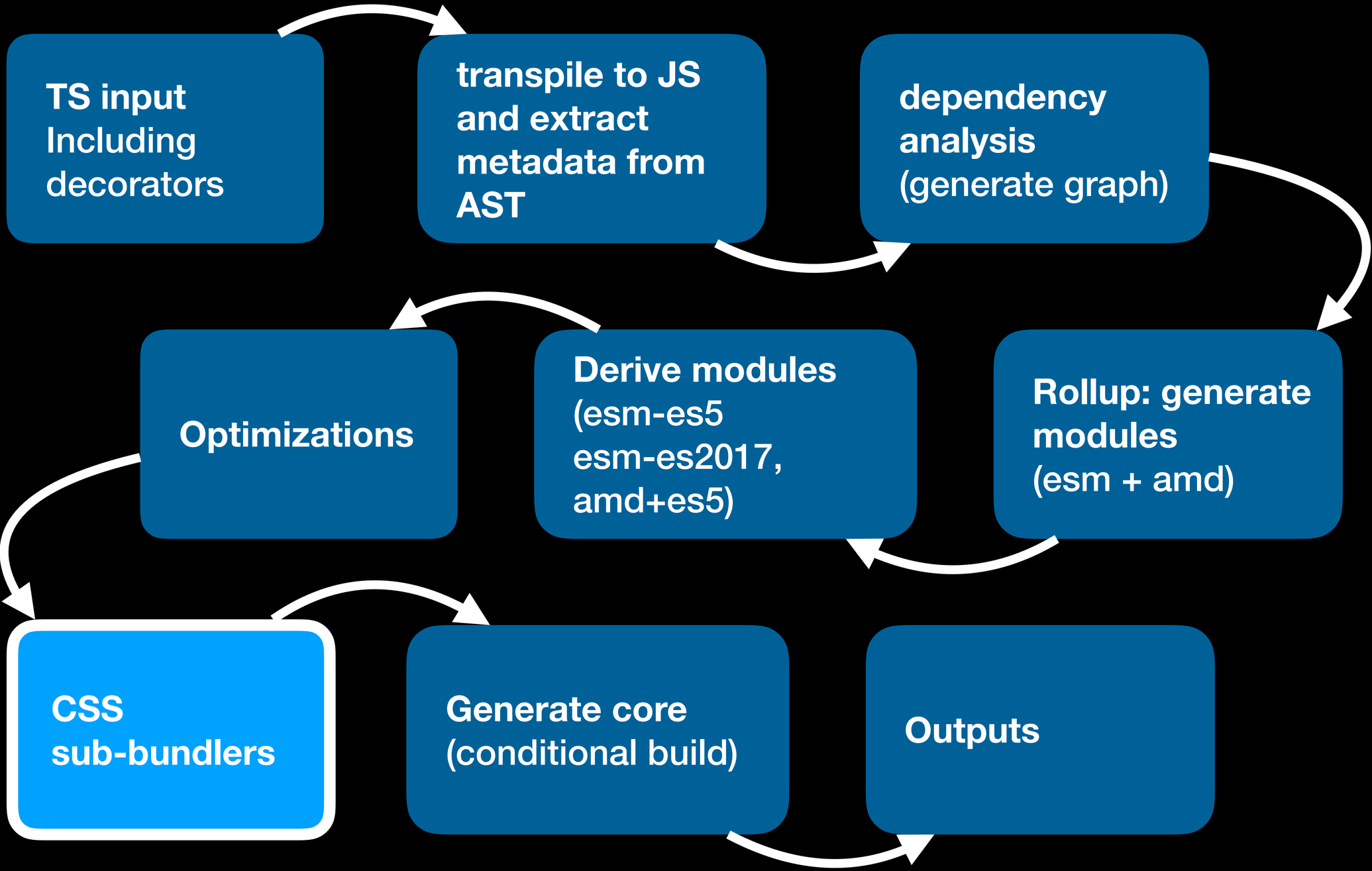
Polyfills for **IE11**



Polyfills for **Firefox**

Polyfills for **Chrome**

D E M O

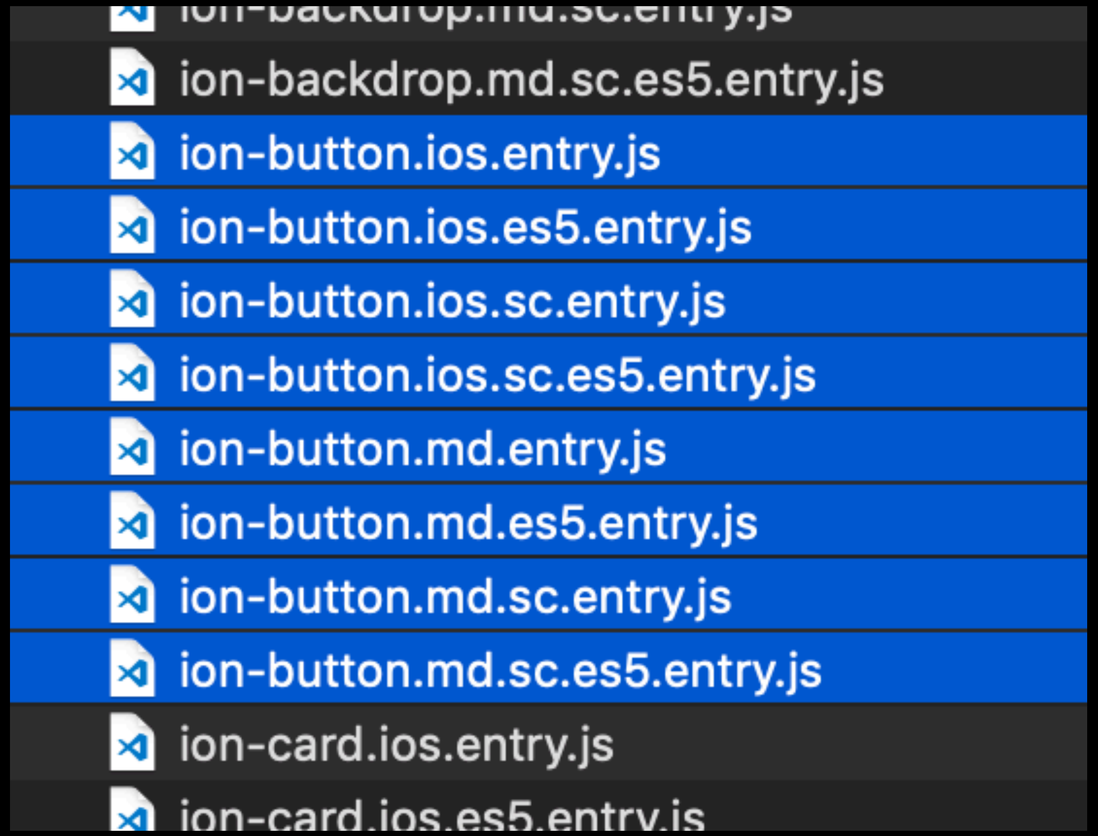

TS input
Including
decorators
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Optimizations
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Rollup: generate
modules
(esm + amd)
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

CSS submodules

```
@Component({
  tag: 'ion-button',
  styleUrls: {
    ios: 'button.ios.scss',
    md: 'button.md.scss'
  },
  shadow: true,
})
export class Button implements
```

Different styles for
the same component

```
<ion-button mode="md">Material Design Button</ion-button>
<ion-button mode="ios">iOS Button</ion-button>
```



- Build with iOS and MD styles
- Build in ESM (modern)
- Build in ES5 (legacy, IE11)
- Build SC: browser that doesn’t support native shadow-dom

CSS autoprefixing

Original CSS

```css
.example {
    display: flex;
    align-items: centered;
    user-select: none;
}
```

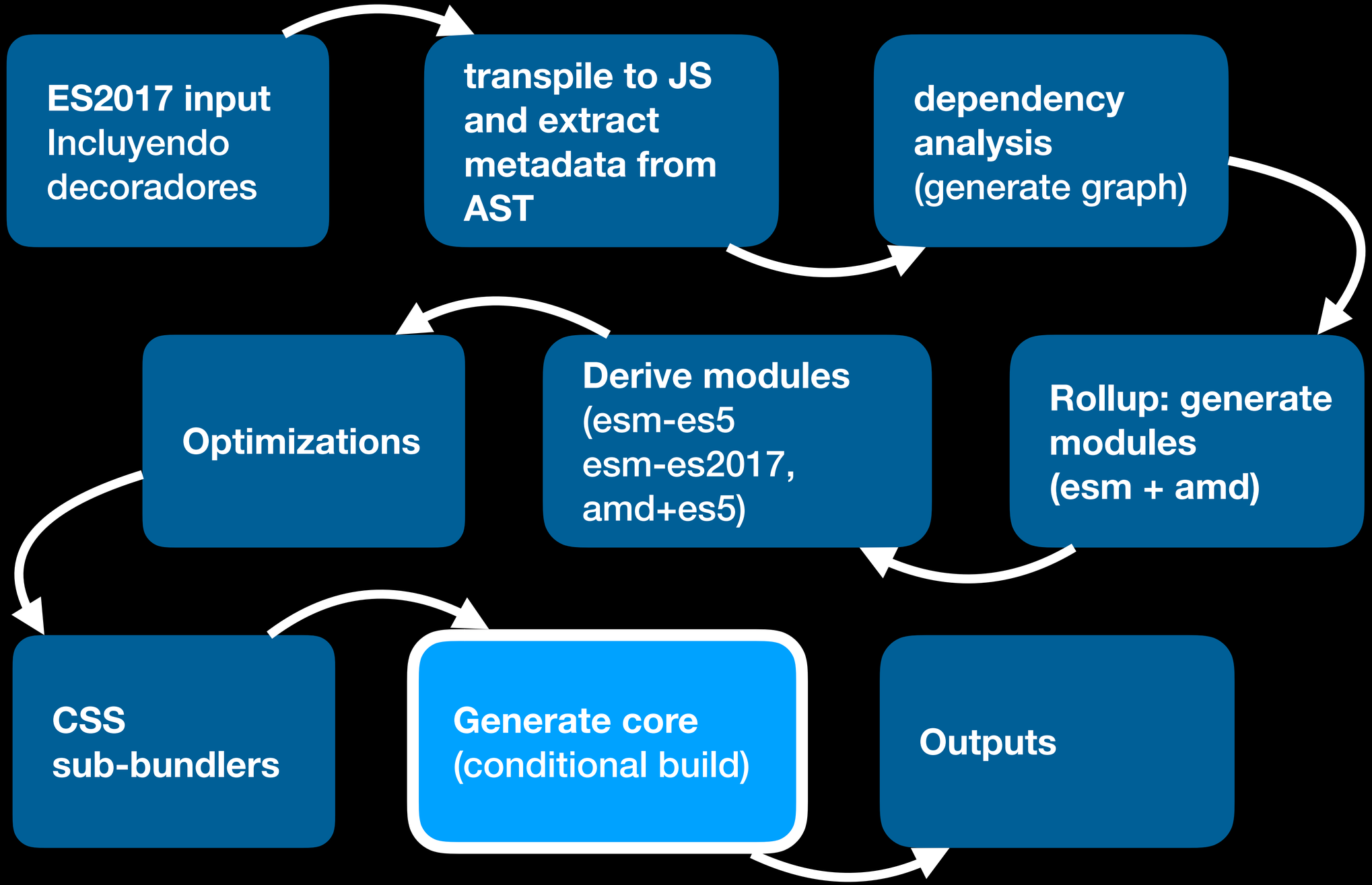
ES5 - legacy browsers

```css
.example {
    display: -webkit-box;
    display: -webkit-flex;
    display: -moz-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-align: centered;
    -webkit-align-items: centered;
       -moz-box-align: centered;
        -ms-flex-align: centered;
            align-items: centered;
    -webkit-user-select: none;
       -moz-user-select: none;
        -ms-user-select: none;
            user-select: none;
}
```

ESM - modern browsers

Firefox, Chrome, Safari

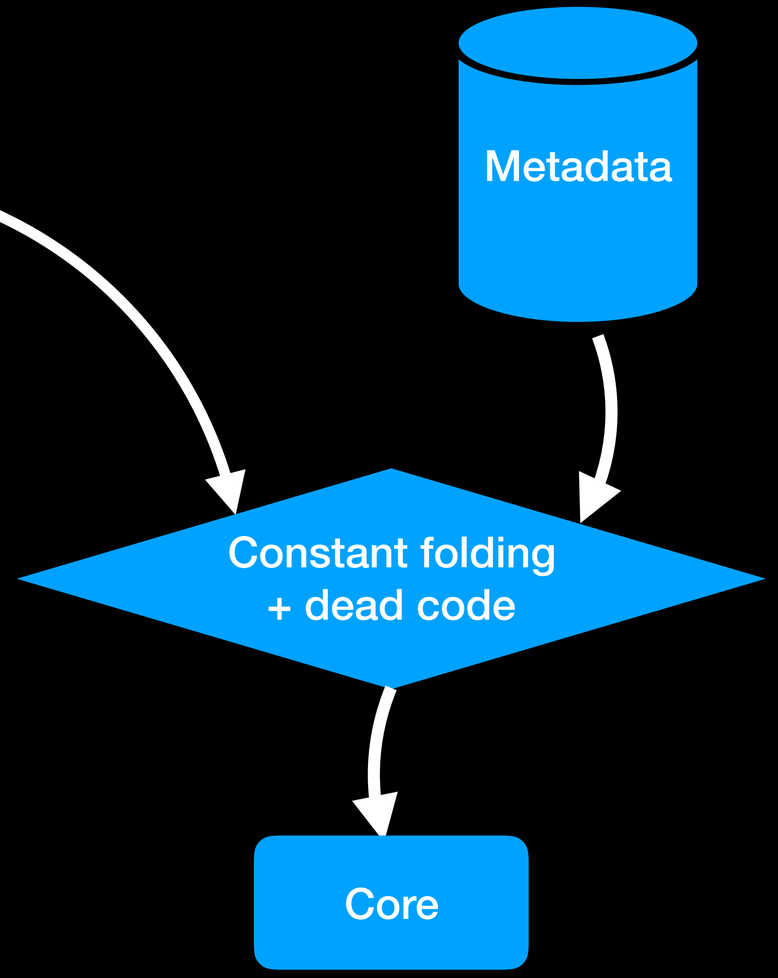
```css
.example {
    display: flex;
    align-items: centered;
    user-select: none;
}
```

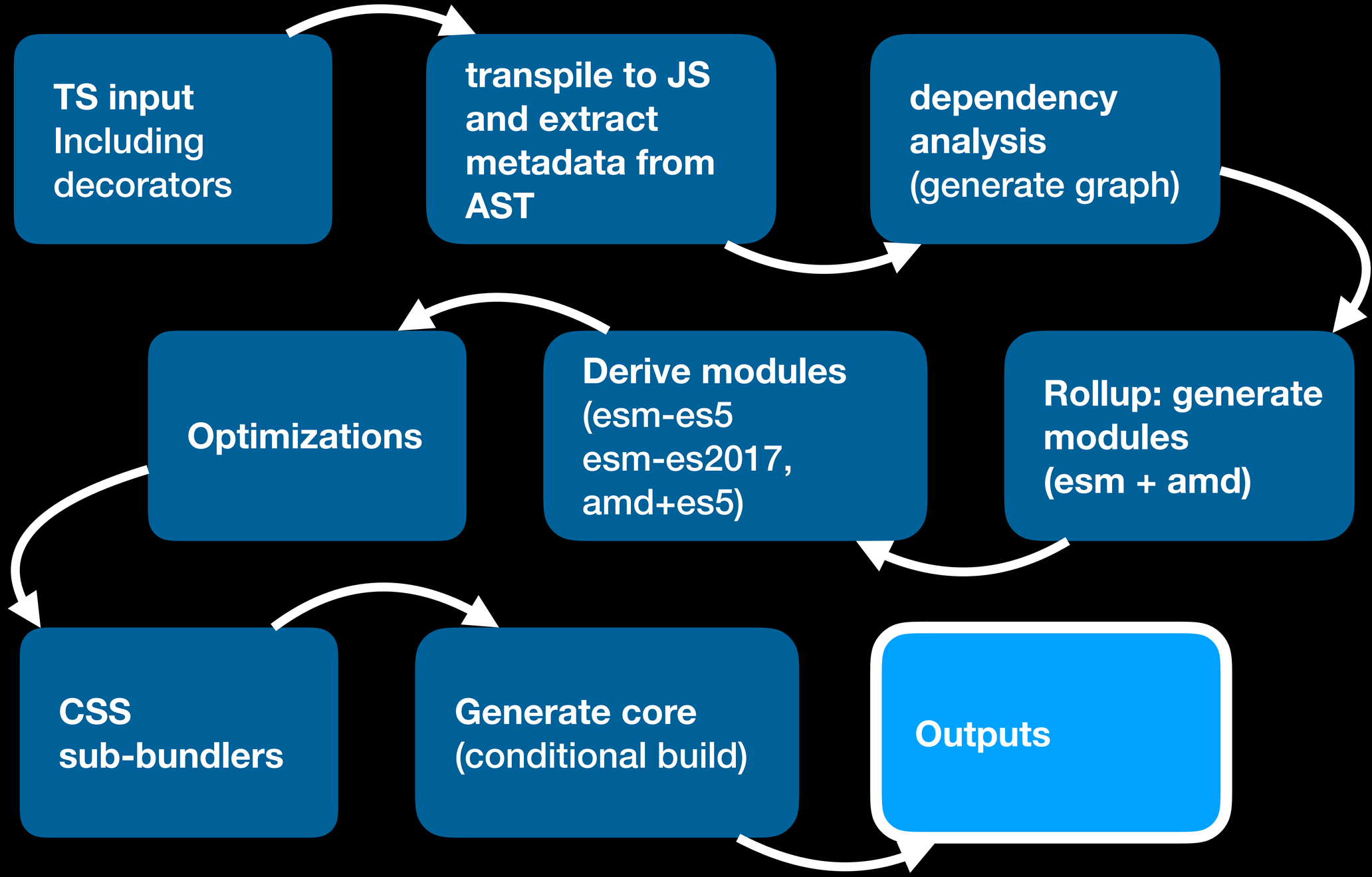

ES2017 input
Incluyendo
decoradores
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Rollup: generate
modules
(esm + amd)
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Optimizations
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

conditional builds

$f(components, jsV, type) \Rightarrow core$

```
if (__BUILD_CONDITIONALS__.event) {
  // add each of the event emitters which wire up instance methods
  // to fire off dom events from the host element
  initEventEmitters(plt, componentConstructor.events, instance);
}

if (__BUILD_CONDITIONALS__.listener) {
  try {
    // replay any event listeners on the instance that
    // were queued up between the time the element was
    // connected and before the instance was ready
    queuedEvents = plt.queuedEvents.get(elm);
    if (queuedEvents) {
      // events may have already fired before the instance was even ready
      // now that the instance is ready, let's replay all of the events that
      // we queued up earlier that were originally meant for the instance
      for (i = 0; i < queuedEvents.length; i += 2) {
        // data was added in sets of two
        // first item the eventMethodName
```




TS input
Including
decorators
transpile to JS
and extract
metadata from
AST
dependency
analysis
(generate graph)
Rollup: generate
modules
(esm + amd)
Derive modules
(esm-es5
esm-es2017,
amd+es5)
Optimizations
CSS
sub-bundlers
Generate core
(conditional build)
Outputs

framework bindings

```
export const config = {
  outputTargets: [
    {
      type: 'angular',
      componentCorePackage: '@ionic/core',
      directivesProxyFile: '../angular/src/directives/proxies.ts',
    }
  ]
};
```

framework bindings

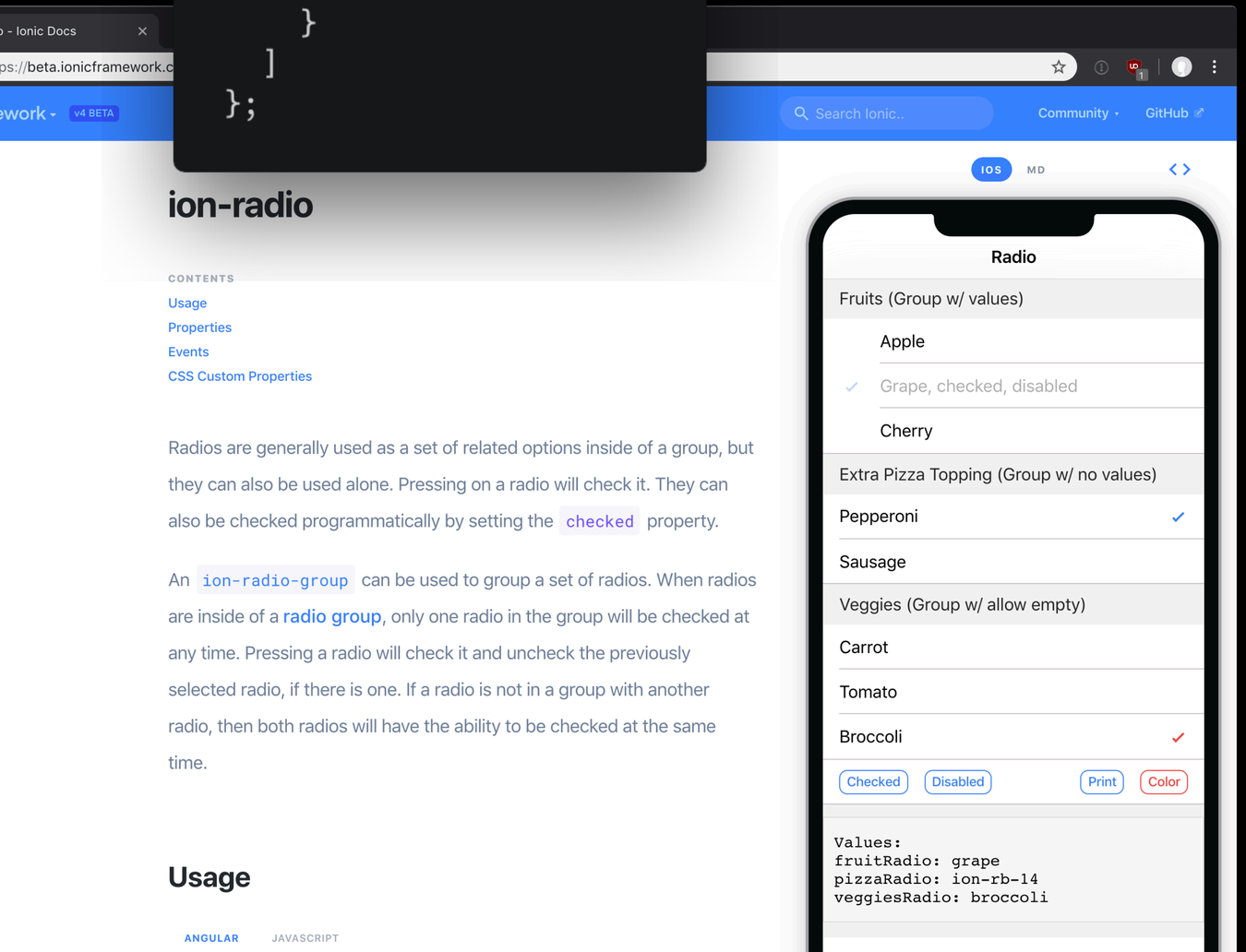
```
import { Component, ViewChild } from '@angular/core';
import { Slides } from '@ionic/angular';

@Component({
  selector: 'page-tutorial',
  templateUrl: 'tutorial.html',
})
export class TutorialPage {

  @ViewChild(Sli

  slideNext() {
    this.slides.slideNext();
  }
```

```
(property) Components.IonSlides['slideNext']: (speed?: num
ber, runCallbacks?: boolean) ⇒ Promise<void>
```

Transition to the next slide.

Docs generation

```
export const config = {
  outputTargets: [
    {
      type: 'docs',
    }
  ]
};
```



readme.md

ion-radio

Radios are generally used as a set of related options inside of a group, but they can also be used alone. radio will check it. They can also be checked programmatically by setting the `checked` property.

An `ion-radio-group` can be used to group a set of radios. When radios are inside of a radio group, only group will be checked at any time. Pressing a radio will check it and uncheck the previously selected rad one. If a radio is not in a group with another radio, then both radios will have the ability to be checked at

Properties

Property	Attribute	Description
`checked`	`checked`	If `true`, the radio is selected. Defaults to `false`.
`color`	`color`	The color to use from your application's color palette. Default options are: `"primary"`, `"secondary"`, `"tertiary"`, `"success"`, `"warning"`, `"danger"`, `"light"`, `"medium"`, and `"dark"`. For more information on colors, see theming.
`disabled`	`disabled`	If `true`, the user cannot interact with the radio. Defaults to `false`.
`mode`	`mode`	The mode determines which platform styles to use. Possible values are: `"ios"` or `"md"`.
`name`	`name`	The name of the control, which is submitted with the form data.
`value`	--	the value of the radio.

Bonus: CSS variables

CSS Custom Properties

Name	Description
`--background`	Background of the button
`--background-activated`	Background of the button when activated
`--background-focused`	Background of the button when focused
`--border-color`	Border color of the button
`--border-radius`	Border radius of the button
`--border-style`	Border style of the button
`--border-width`	Border width of the button
`--box-shadow`	Box shadow of the button
`--color`	Text color of the button

SSR and Prerendering

D E M O

Thank You

@manucorporat