



Licence Pro MI-AW

Module INFO-1

*Développement Mobile Hybride
IONIC 4, Angular, Cordova*



<https://ionicframework.com/docs/v4>

Objectif de la Séance

- Présenter les principes de Ionic
- « Jouer » avec une petite application

Ce document est un survol rapide
d'un **excellent** tutoriel (4h)

© Maximilian Schwarzmüller
à revoir en VO (anglais germanique)

<https://www.youtube.com/watch?v=r2ga-iXS5i4>

Ionic?

One Codebase (HTML + JS + CSS)



Progressive Web App



iOS App

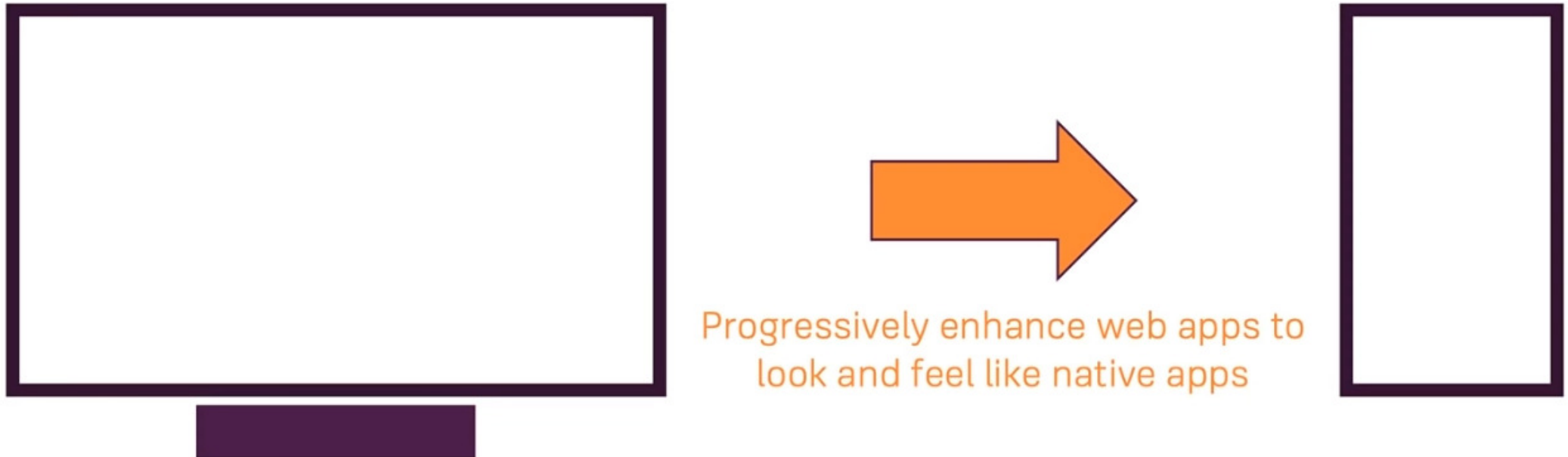


Android App



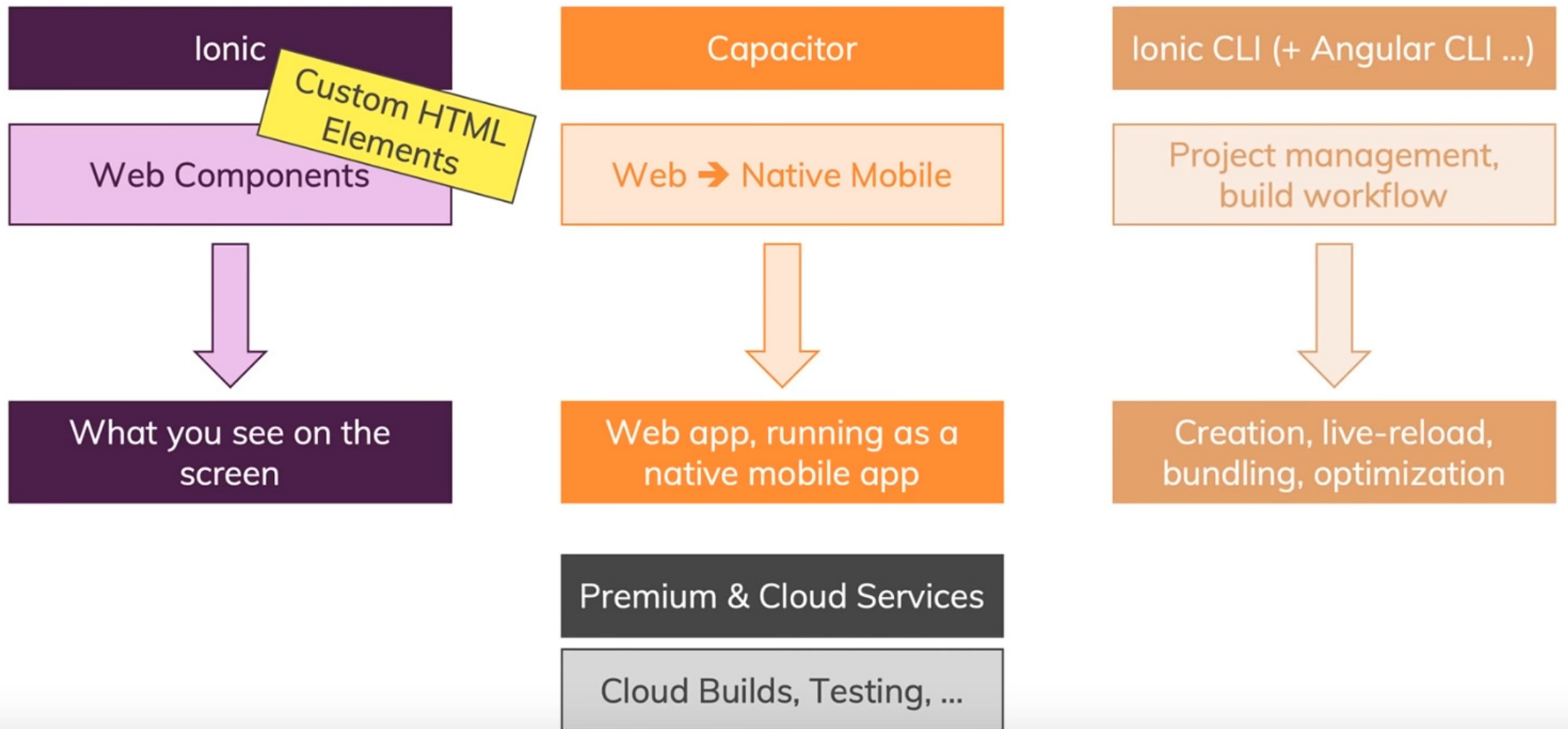
Electron (Desktop) App

What are Progressive Web Apps (PWAs)?



- Be **reliable**: Load fast and provide offline functionality
- **Fast**: Respond quickly to user actions
- **Engaging**: Feel like a native app on mobile devices

Ionic Platform



Understanding the Ionic Ecosystem

A Suite of platform-adaptive
Web Components (“HTML
Elements”)

Tools to “convert” a Web App
to a Native Mobile App

Tools to speed up Web &
Mobile Development

`<ion-card>`, `<ion-avatar>`,
`<ion-icon>`, ...

Hybrid App

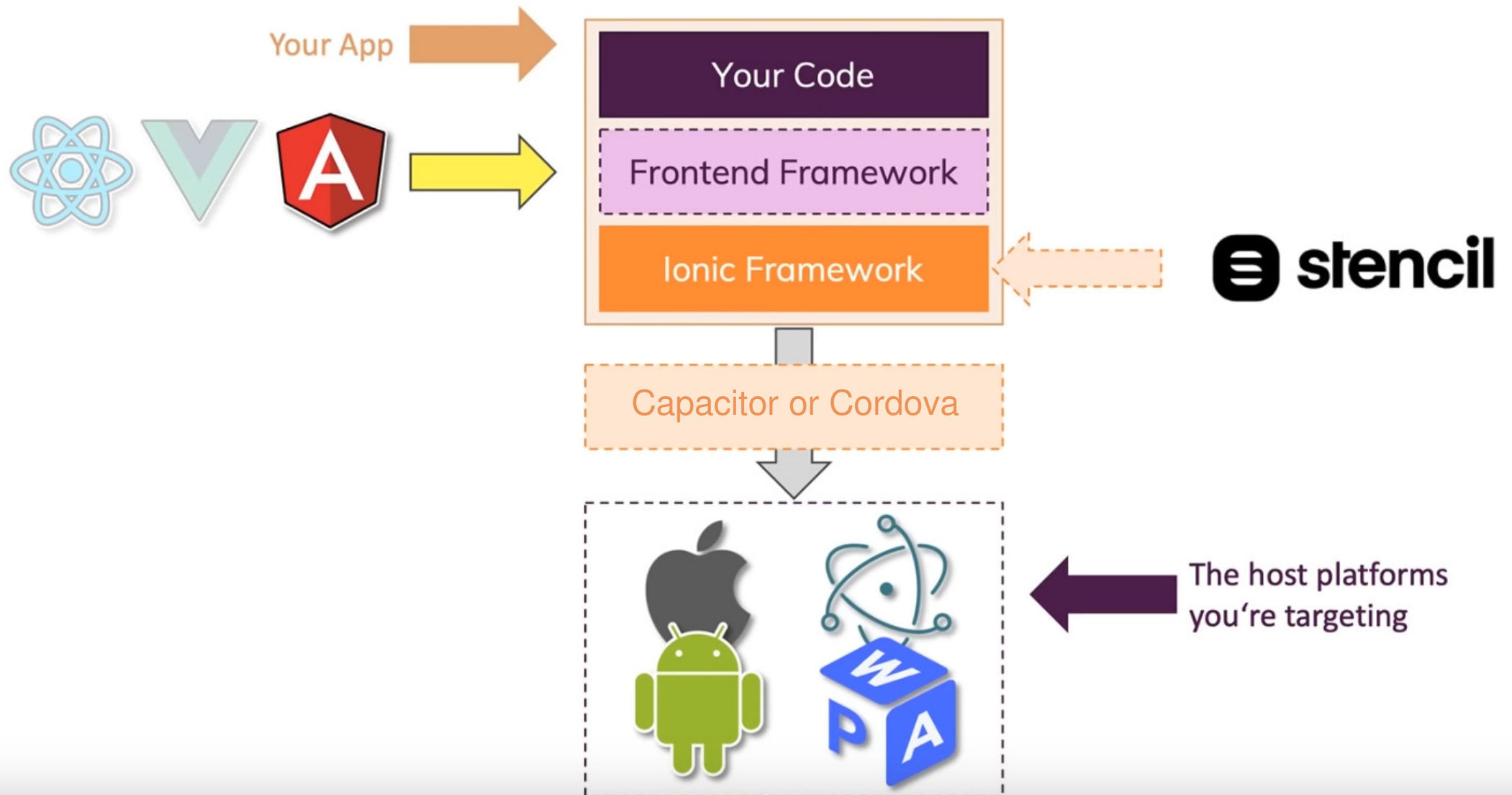
Runs as a Website & a Mobile
App (incl. Native Device
Features)

Ionic CLI, Hot Module
Replacement, Testing on
Emulators & Real Devices

Normal Web App (built with
or without a Framework)

(Progressive) Web App

Native Mobile App



What is Angular?



or Frameworks in General...

Allow you to create highly reactive JS-driven Web Apps

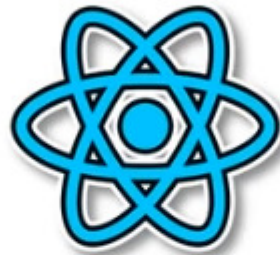
Totally optional but makes your life easier

Used to create web apps (often Single-Page Applications)

Helps with two main problems: **State/ Data Management** and **Routing/ Navigation (Lazy Loading)**

Angular Isn't Your Thing?

You can use Vanilla JS or other Frameworks, too!



Dive into the Quickstart First!

The Ionic Components + Core Concepts are the same!

Ionic History

2013

Ionic 1

Create mobile apps with
Angular 1

2016

Ionic 2

Create mobile apps with
Angular 2

2017

Ionic 3

Create mobile apps &
PWAs with Angular 4+

2019

Ionic 4+

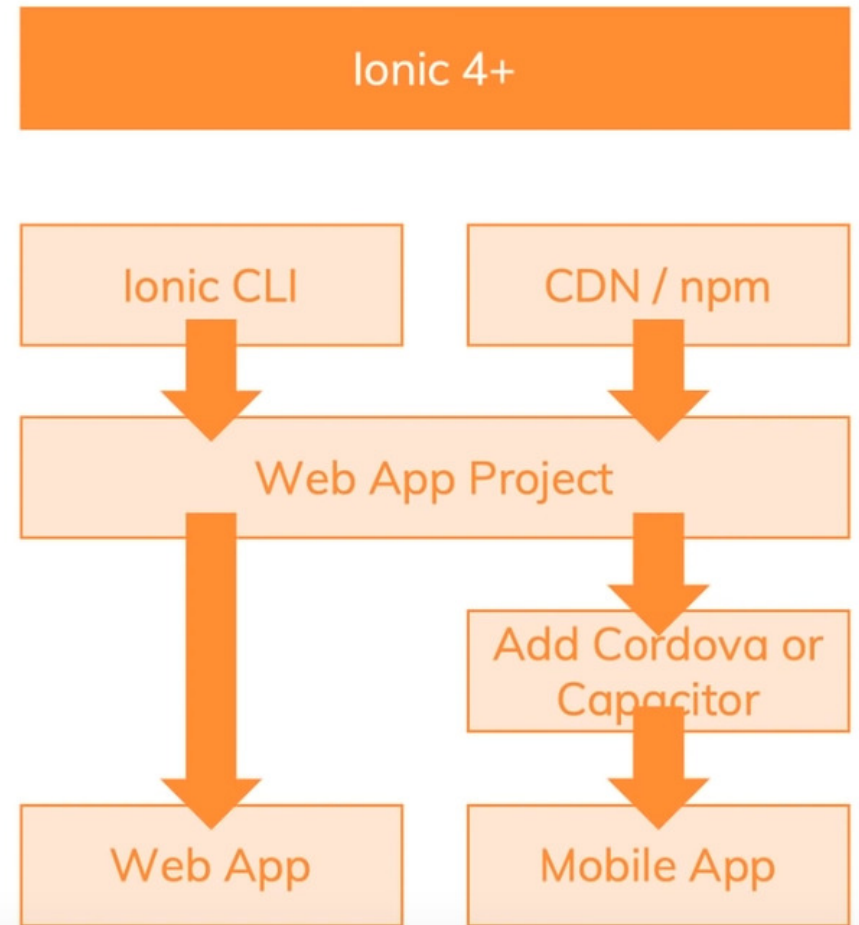
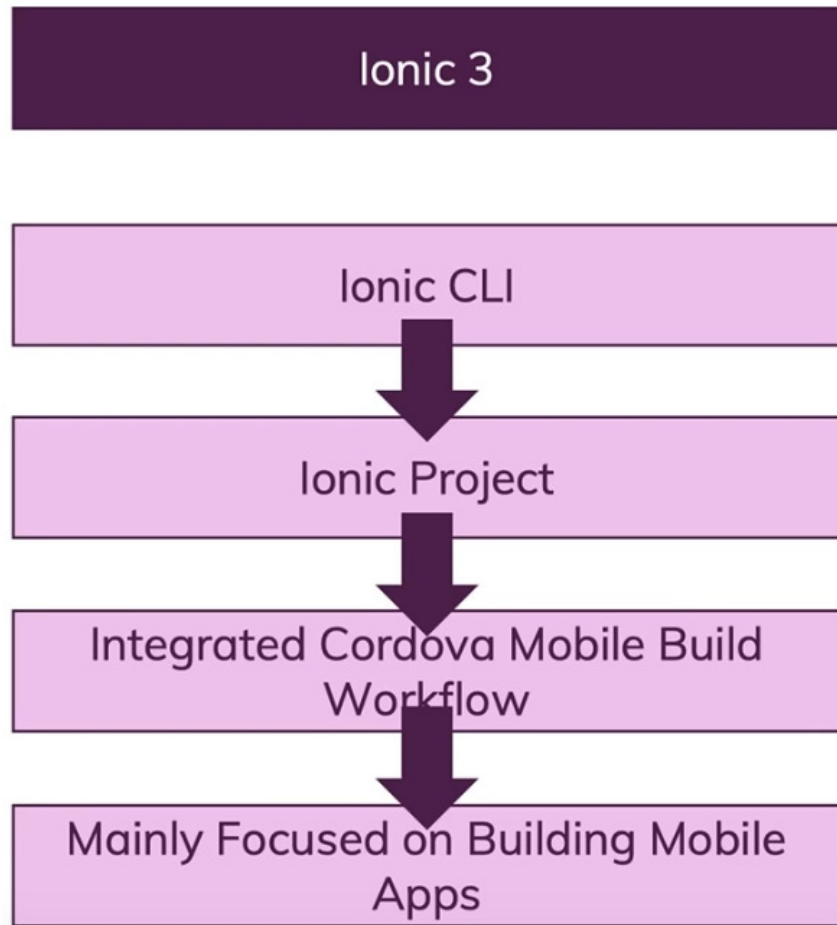
Create mobile apps &
PWAs with JavaScript

...
2022

...
Ionic 6

Intégration Facile de
React, Vue

Ionic 3 vs Ionic 4+



Core Building Blocks

UI Components

`<ion-card>`, `<ion-image>`, ...

Themes & Styles

CSS & CSS Variables

Navigation

Custom or Built-in Router

State Management

Passing Data Around

Native Device Features

Camera & More

Ionic CLI & Publishing

From Development to
Deployment

Using the Ionic Components

The Core of Ionic!

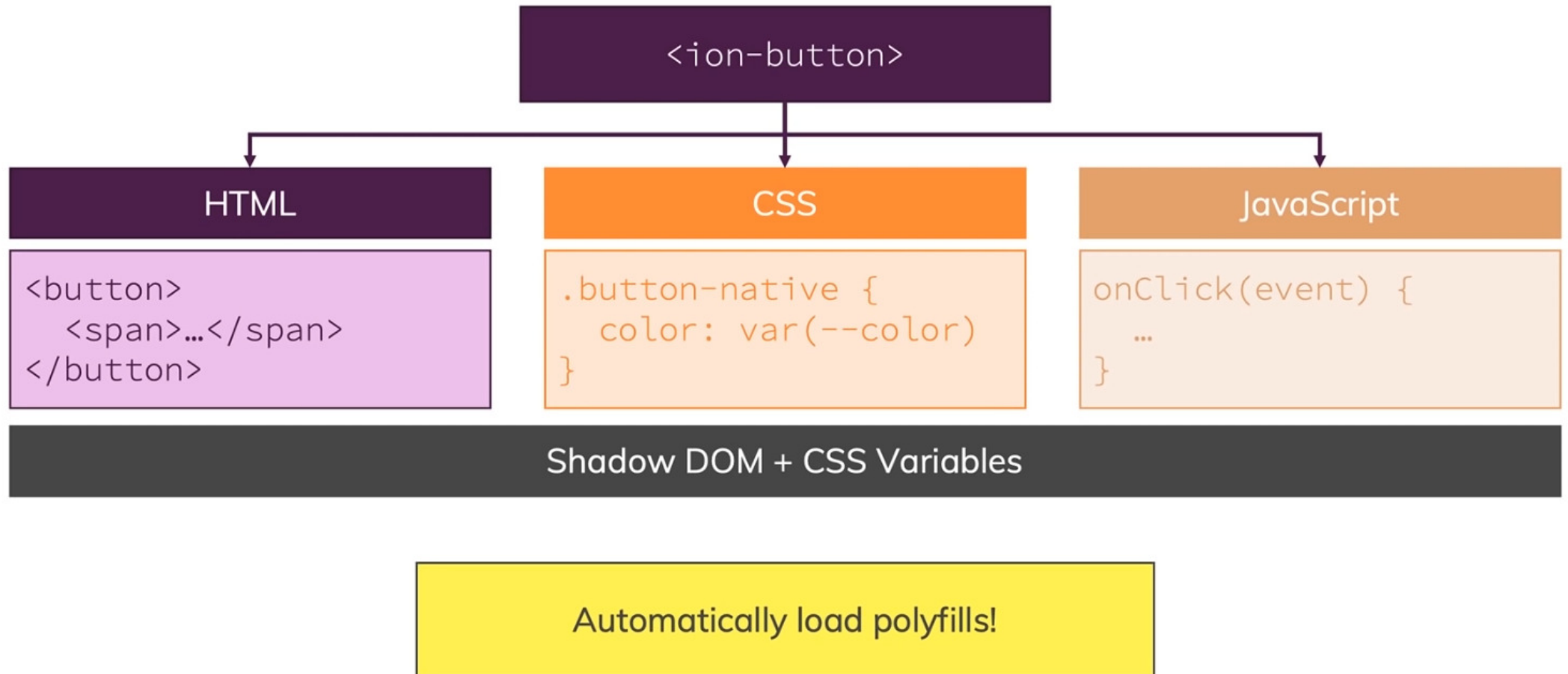
```
<ion-button fill="outline" color="primary">
```

Use like a normal
HTML Element

Supports Attributes
& Properties

Emits events

How Do Ionic 4 Web Components Work?



Core Component Types

Output

`<ion-img>`
`<ion-badge>`
`<ion-loading>`
`<ion-label>`
`<ion-title>`
`<ion-thumbnail>`
`<ion-toolbar>`
`<ion-alert>`
`<ion-toast>`
`<ion-modal>`
...

Layout

`<ion-grid>`
`<ion-row>`
`<ion-col>`
`<ion-list>`
`<ion-card>`
`<ion-infinite-scroll>`
`<ion-tabs>`
...

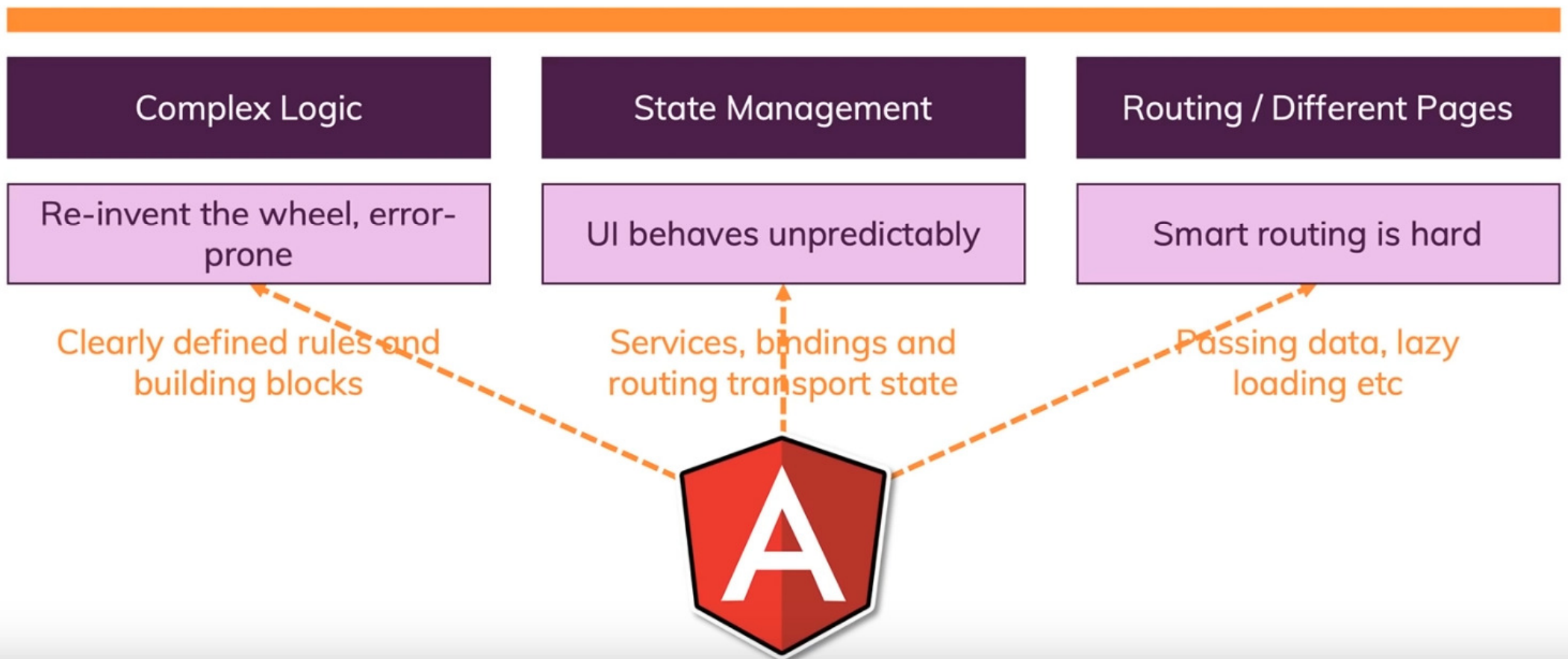
Input

`<ion-button>`
`<ion-input>`
`<ion-textarea>`
`<ion-menu>`
`<ion-select>`
`<ion-datetime>`
`<ion-fab>`
`<ion-toggle>`
...

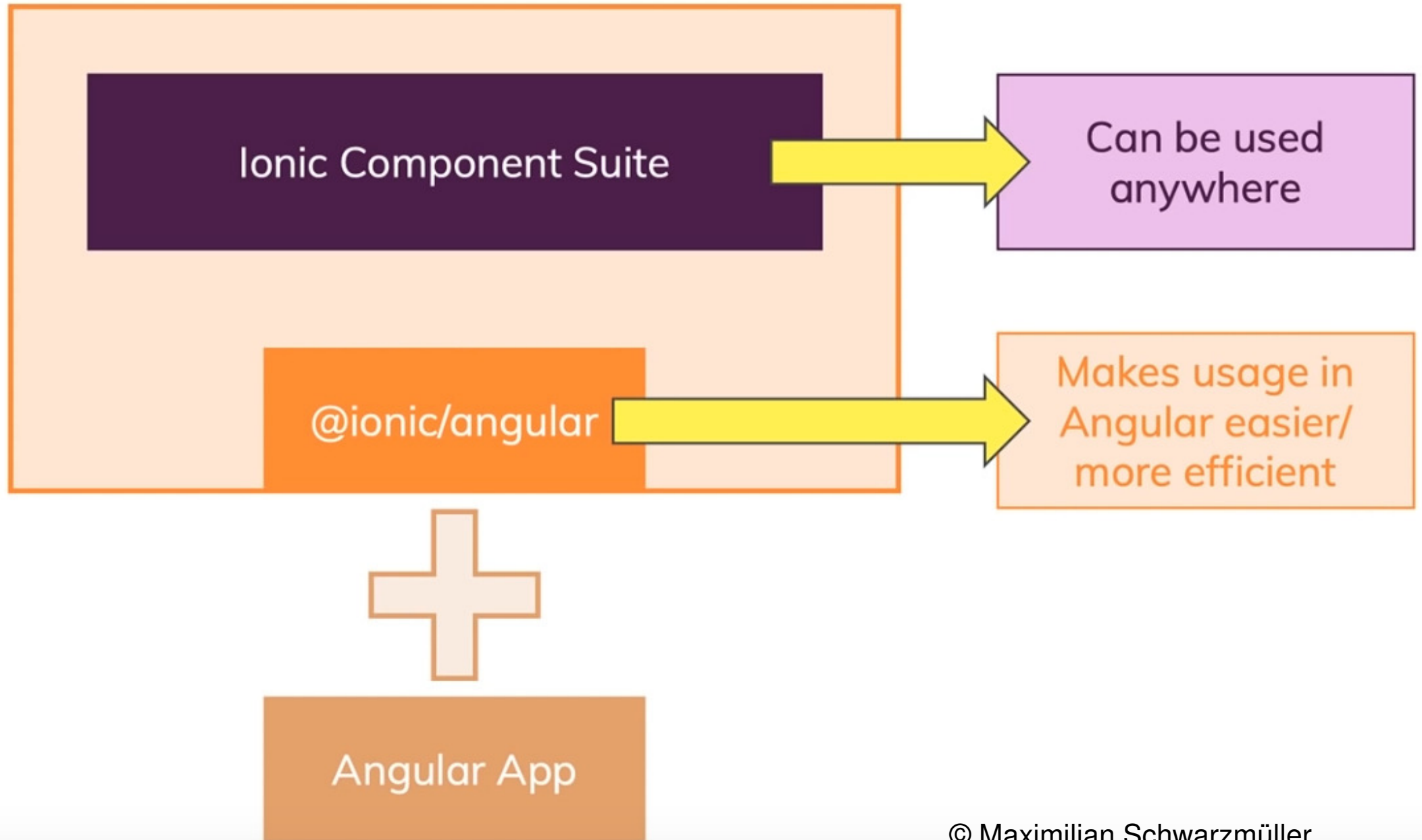
<https://ionicframework.com/docs/components>

Why Angular?

This will be hard!



The @ionic/angular Wrapper



Angular Components & Ionic Components

Angular

Created by adding `@Component()` to a class

Use Angular templates → Angular does the actual DOM rendering

Only usable inside of other Angular components / in an Angular app

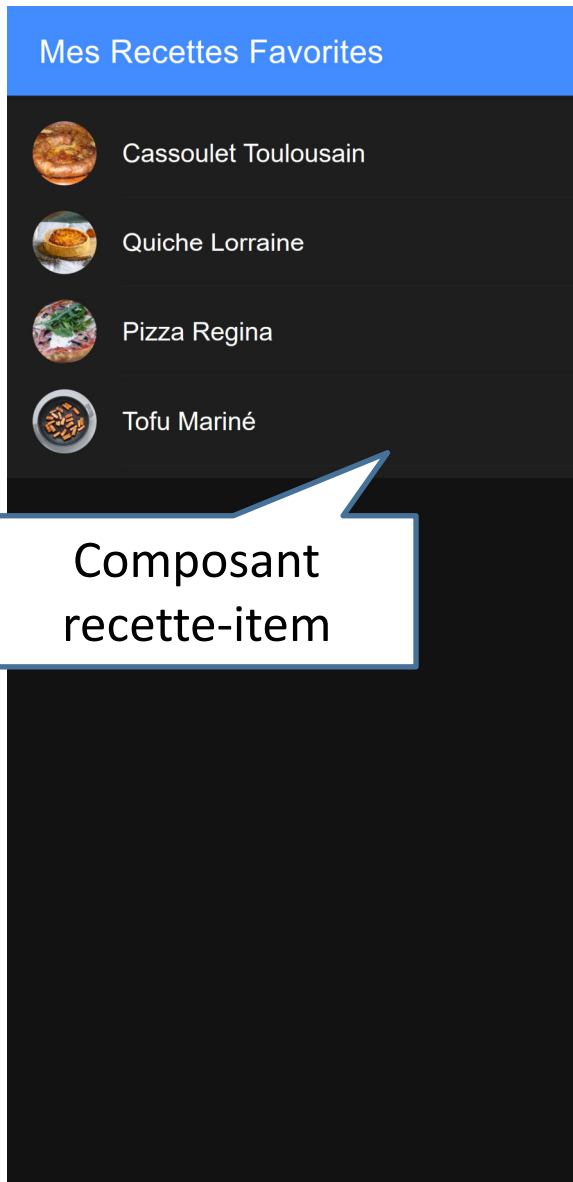
Ionic

Ships with pre-built components, not editable by you

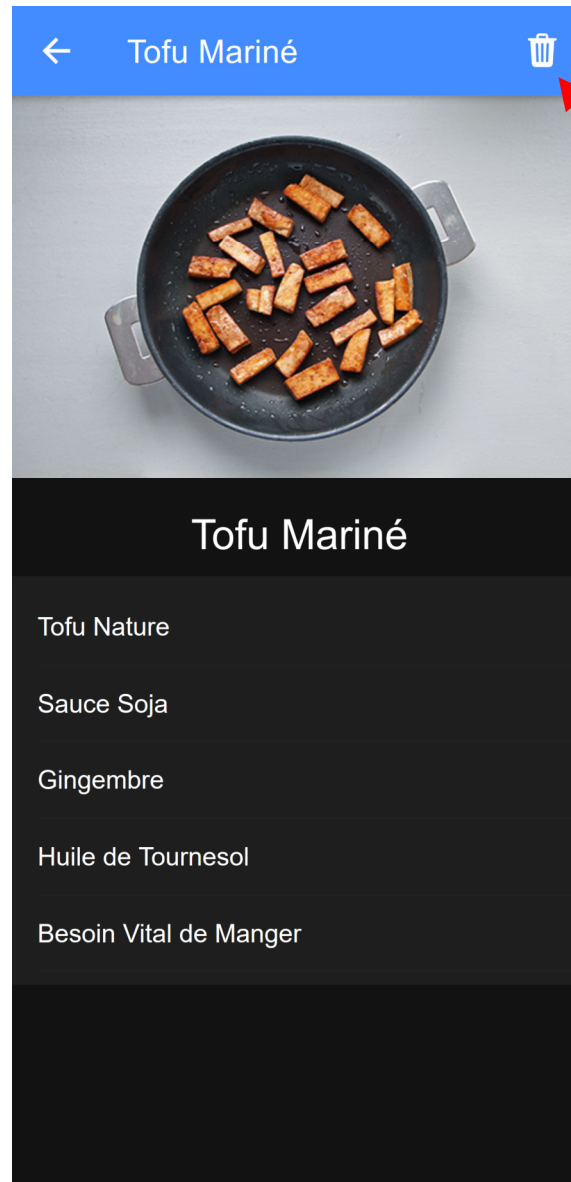
Use native web technologies → Web component specification

Can be used anywhere, where HTML elements can be used

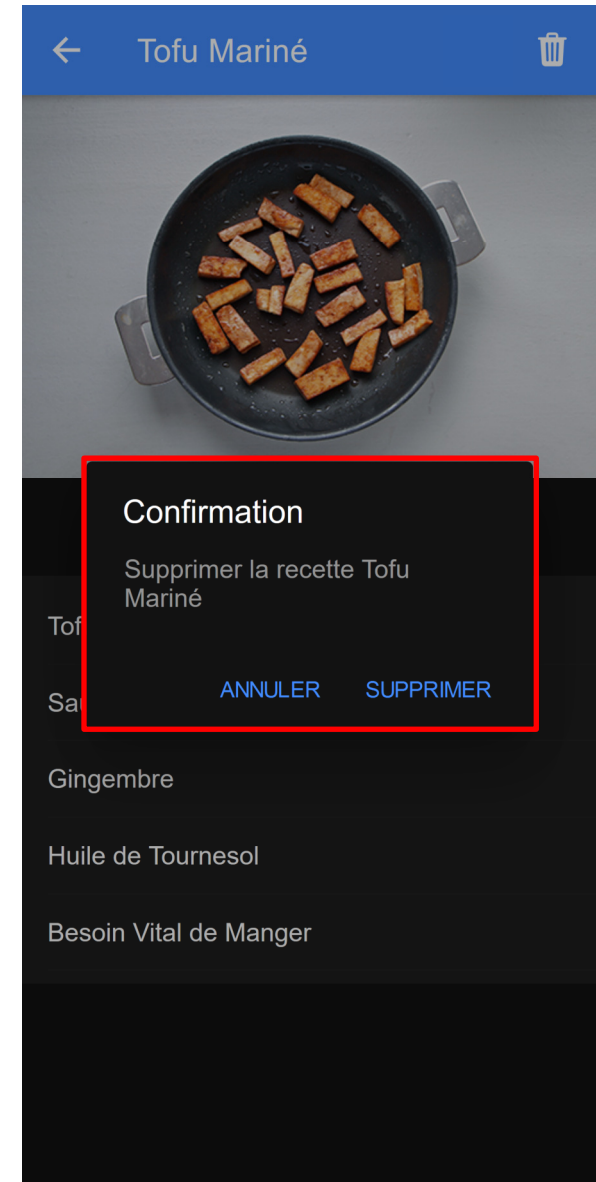
Exemple : Projet Recettes



Page recettes-liste



Page recette-detail

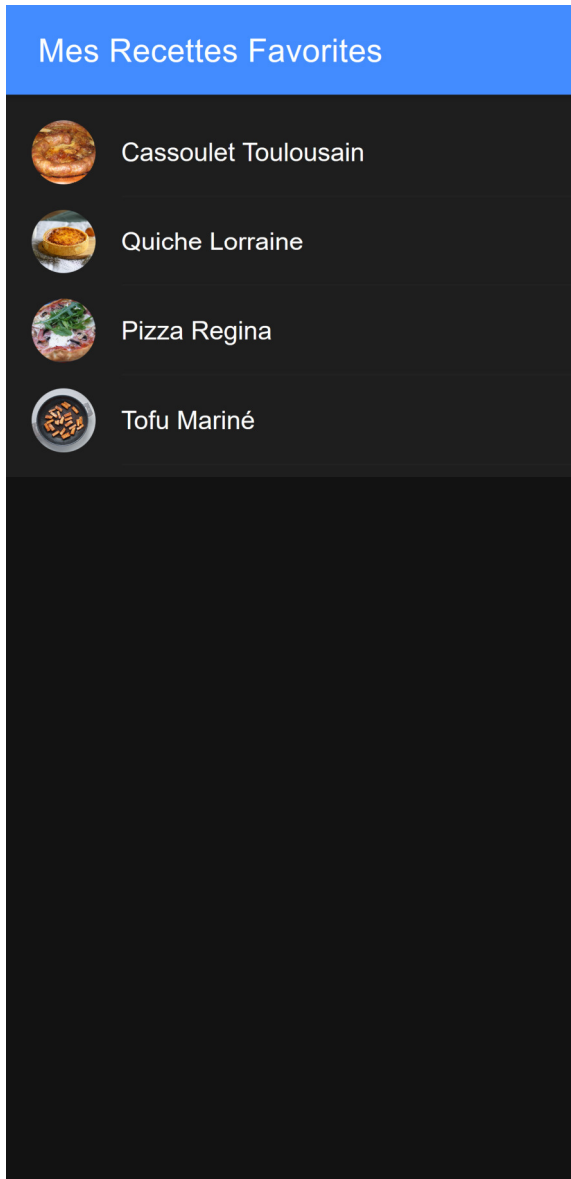


Composant AlertCtrl

Installation

1. Installer Ionic 4 :
`(sudo) npm install -g ionic@4`
2. Créer un dossier **recettes**
3. Copier le contenu de l'archive **recettes.zip** dans le répertoire **recettes**
4. Se placer à la racine du projet :
`cd recettes`
5. Installer les dépendances:
`npm install`
6. Construire le projet :
`ionic build`
7. Lancer le projet dans le navigateur :
`ionic serve`
8. Installer cordova pour android :
`ionic cordova platform add android@9`
9. Lancer le projet sous android :
`ionic cordova run android`

Page Recettes-liste



- Un « model » pour spécifier une entité recette :
-> **fichier model/recettes.model.ts**
- Un service **RecettesService** pour gérer une collection de recettes :
ionic generate service recettes
- Une page (composant angular) **recettes-liste** pour afficher la liste des recettes :
ionic generate page recettes/recettes-liste
Cette page sera visible à l'URL /recette-liste et sera définie comme la page d'accueil de l'application (voir src/app/app-routing.module.ts)
- Un composant recette-item (élément d'une liste) :
ionic generate component recettes/recette-item

Modèle : Interface Recette

src/app/model/recettes.model.ts

```
export interface Recette {  
  id: string;  
  titre: string;  
  visuel: string;  
  ingredients: string[];  
}
```

- On aurait pu aussi écrire une classe TS...

Service Recettes

src/app/recettes/recettes.service.ts

```
import { Injectable } from '@angular/core';
import { Recette } from '../model/recette.model';

@Injectable({
  providedIn: 'root'
})
export class RecettesService {
  private recettes: Recette[] = [
    {
      id: 'cassoulet',
      titre: 'Cassoulet Toulousain',
      visuel: 'https://img-3.journaldesfemmes.fr/...',
      ingredients: ['Haricots Secs', 'Couenne', 'Tomates', 'Saucisse', 'Cuisse de Canard']
    }, // ...
  ];
  constructor() {}
  getAllRecettes() { // ...
  }
  getRecette(recetteld: string) { // ...
  }
  deleteRecette(recetteld: string) { // ...
  }
}
```

Composant <app-recette-item>

src/app/recettes/recette-item/recette-item.component.html

```
<ion-item [routerLink]="['/recette-detail', recette.id]'>  
  <ion-avatar slot="start">  
    <ion-img [src]="recette.visuel"></ion-img>  
  </ion-avatar>  
  <ion-label>{{ recette.titre }}</ion-label>  
</ion-item>
```

**Pour naviguer vers la route
recette-detail/recette.id**

src/app/recettes/recette-item/recette-item.component.ts

```
import { Component, OnInit, Input } from '@angular/core';  
import { Recette } from '../../model/recettes.model';  
  
@Component({  
  selector: 'app-recette-item',  
  templateUrl: './recette-item.component.html',  
  styleUrls: ['./recette-item.component.scss'],  
})  
export class RecetteItemComponent implements OnInit {  
  @Input() recette: Recette ; // @input() : pour pouvoir lier de l'extérieur  
  constructor() { }  
  
  ngOnInit() { }  
}
```

Vue : page Recettes

src/app/recettes.recettes-liste/recettes.page.html

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>Mes Recettes Favorites</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-list>
    <app-recette-item
      *ngFor="let uneRecette of recettes"
      [recette]="uneRecette"
    >
    </app-recette-item>
  </ion-list>
</ion-content>
```

Mes Recettes Favorites



Cassoulet Toulousain



Quiche Lorraine



Pizza Regina



Tofu Mariné

Contrôleur : page Recettes

src/app/recettes/recettes.page.ts

```
import { Component, OnInit } from '@angular/core';
import { Recette } from '../recettes.model';
import { RecettesService } from '../recettes.service';

@Component({
  selector: 'app-recettes',
  templateUrl: '../recettes.page.html',
  styleUrls: ['../recettes.page.scss'],
})
export class RecettesPage implements OnInit {

  recettes: Recette[];

  constructor(private RecettesService: RecettesService) { }

  ngOnInit() {

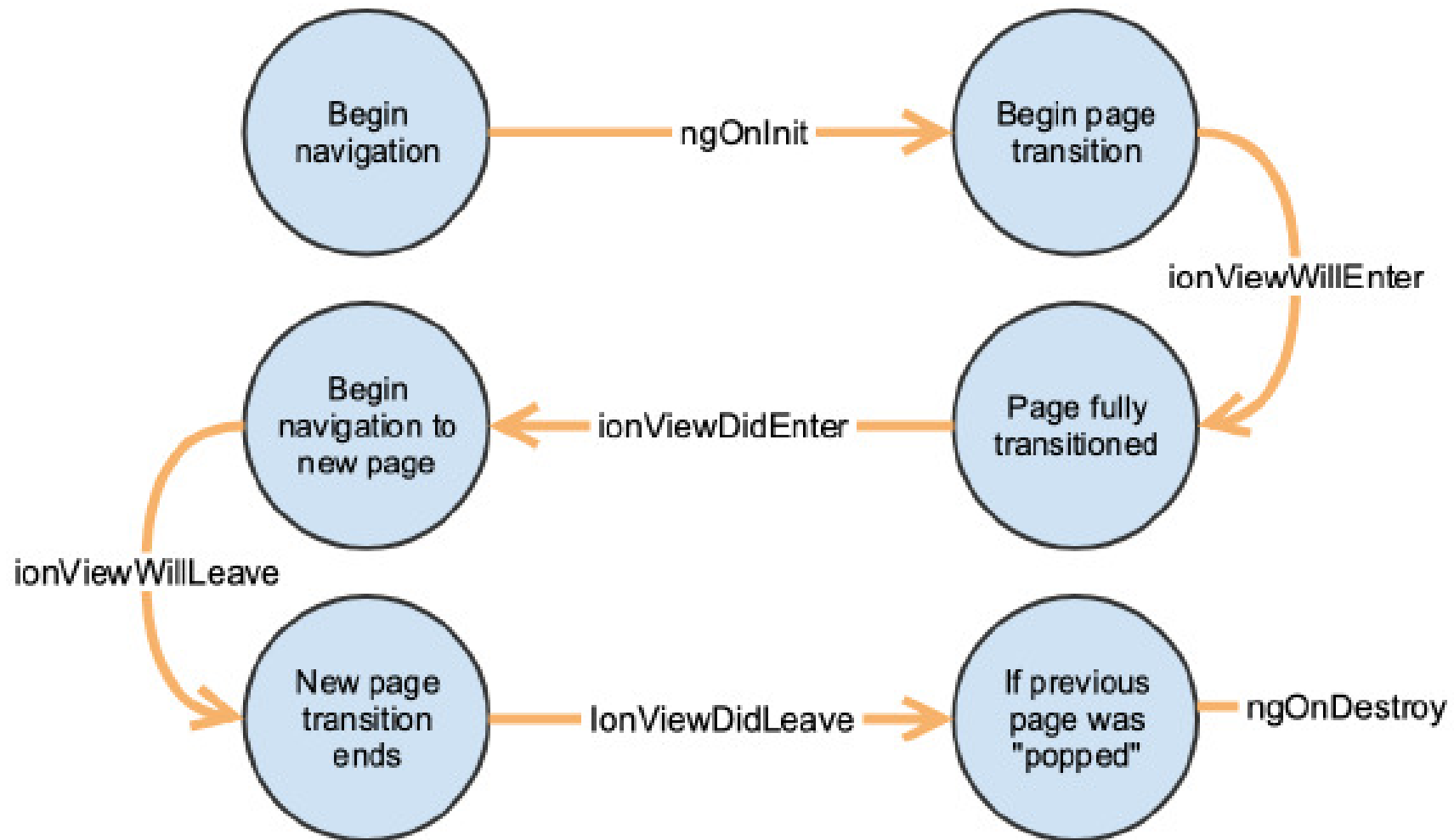
  }

  ionViewWillEnter() {
    this.recettes = this.RecettesService.getAllRecettes();
  }

}
```

**ionViewWillEnter est exécutée
avant chaque affichage de la page**

Cycle de Vie Ionic



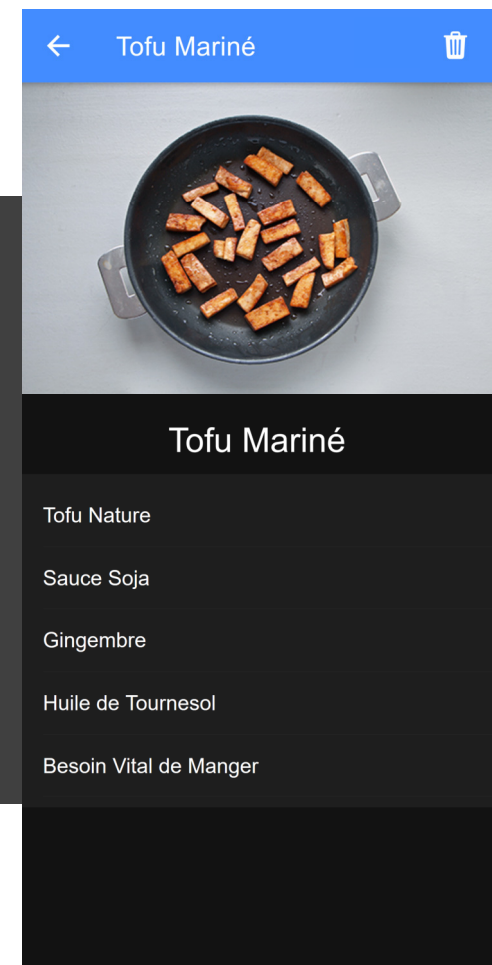
<https://ionicframework.com/docs/angular/lifecycle>

Vue : page recette détail (1)

- `ionic generate page recettes/recette-detail`
- Cette page sera visible à l'URL `/recette/{id}`
- C'est une « fille » de la page Recettes
(voir fichier `src/app/app-routing.module.ts`)

`src/app/recettes/recette-detail/recette-detail.page.html`

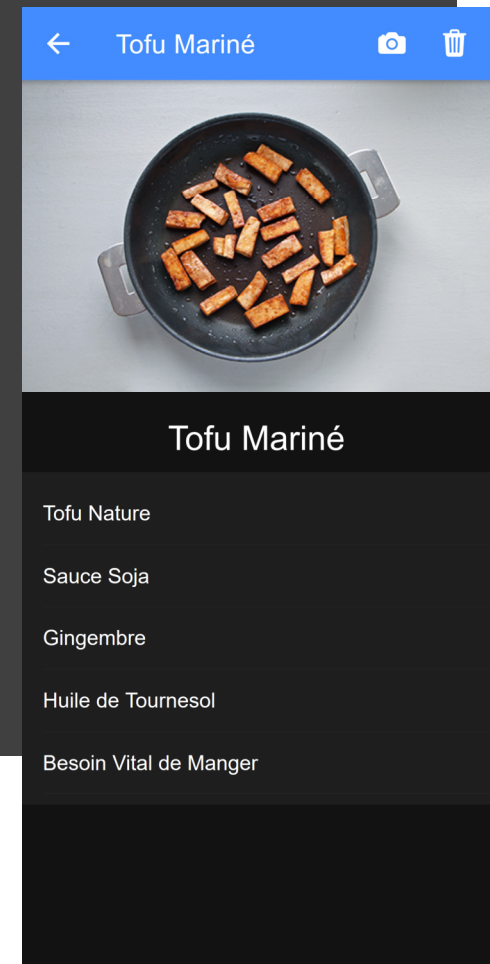
```
<ion-header>
  <ion-toolbar color="primary">
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/recettes"></ion-back-button>
    </ion-buttons>
    <ion-title>{{recette.titre}}</ion-title>
    <ion-buttons slot="primary">
      <ion-button (click)="onDeleteRecette();">
        <ion-icon name="trash" slot="icon-only"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```



Vue : page recette détail (2)

src/app/recettes/recette-detail/recette-detail.page.html

```
<ion-content>
  <ion-grid class="ion-no-padding">
    <ion-row> <ion-col class="ion-no-padding">
      <ion-img [src]="recette.visuel"></ion-img>
    </ion-col></ion-row>
    <ion-row><ion-col>
      <h1 class="ion-text-center">{{recette.titre}}</h1>
    </ion-col></ion-row>
    <ion-row><ion-col>
      <ion-list>
        <ion-item *ngFor="let ingredient of recette.ingredients">
          {{ ingredient }}
        </ion-item>
      </ion-list>
    </ion-col></ion-row>
  </ion-grid>
</ion-content>
```



Contrôleur : recette-detail (1)

src/app/recettes/recette-detail/recette-detail.page.ts

```
@Component({
  selector: 'app-recette-detail',
  templateUrl: './recette-detail.page.html',
  styleUrls: ['./recette-detail.page.scss'],
})
export class RecetteDetailPage implements OnInit {
  recette: Recette;

  constructor(private activatedRoute: ActivatedRoute,
    private RecettesService: RecettesService,
    private Router: Router,
    private AlertCtrl: AlertController // composant pour afficher une alerte
  ) { }
```

Contrôleur : recette-detail (2)

src/app/recettes/recette-detail/recette-detail.page.ts

```
ngOnInit() {  
  this.activatedRoute.paramMap.subscribe(paramMap => {  
    // si pas de paramètre recetteId dans la route...  
    if (!paramMap.has('recetteId')) {  
      this.Router.navigate(['/recettes']); // redirection vers /recettes  
      return;  
    }  
  
    const recetteId = paramMap.get('recetteId');  
    this.recette = this.RecettesService.getRecette(recetteId);  
  
    // si la recette n'existe pas dans le Service...  
    if (!this.recette.id) {  
      this.Router.navigate(['/recettes']); // redirection vers /recettes  
    }  
  });  
}
```

Contrôleur : recette-detail (3)

src/app/recettes/recette-detail/recette-detail.page.ts

```
onDeleteRecette() {  
  this.AlertCtrl.create({  
    header: 'Confirmation',  
    message: 'Supprimer la recette ' + this.recette.titre,  
    buttons: [  
      { text: 'Annuler', role: 'cancel' },  
      {  
        text: 'Supprimer',  
        handler: () => {  
          this.RecettesService.deleteRecette(this.recette.id);  
          this.Router.navigate(['/recettes']);  
        }  
      }  
    ]  
  }).then(alertEl => {  
    alertEl.present();  
  })  
}
```

TP « découverte »

- Donner la possibilité, sur la page detail-recette, de modifier la photo d'une recette :
 - Il faudra installer et utiliser le plugin camera
(<https://ionicframework.com/docs/v4/native/camera>) :
`ionic cordova plugin add cordova-plugin-camera`
`npm install @ionic-native/camera`
 - Créer un service **photo** permettant d'utiliser ce plugin
`ionic generate service photo`
 - Rajouter sur la page **detail-recette** un bouton et le code nécessaire pour interagir avec le service **photo**
 - Si la photo n'a pas pu être prise, il faudra afficher un toast :
<https://ionicframework.com/docs/v4/api/toast>
 - Compléter le service **Recettes** pour permettre la mise à jour d'une recette lorsque sa photo a été modifiée
- Ajouter la persistance au Service Recettes :
<https://ionicframework.com/docs/v4/native/native-storage>

S'il vous reste du temps :

- Développer une page permettant de saisir et enregistrer une nouvelle recette :
 - `ionic generate page recettes/recette-saisie`
 - La page devra contenir un formulaire permettant de saisir un identifiant de recette, son titre, une photo et des ingrédients
 - La page devra évidemment interagir avec le service **recettes** et service **photo**