

## A. Définition du concept

**Angular** est framework Javascript développé par Google permettant la réalisation des applications de type « Single Page Application : SPA ». Il est basé sur le concept de l'architecture MVC (Model View Controller : Modèle-vue-contrôleur). En respect de son cycle de publication d'une nouvelle version du framework, Angular aujourd'hui est sa 8<sup>ème</sup> version, et comme toute nouvelle version, Angular 8.0 apporte de nouvelles fonctionnalités.

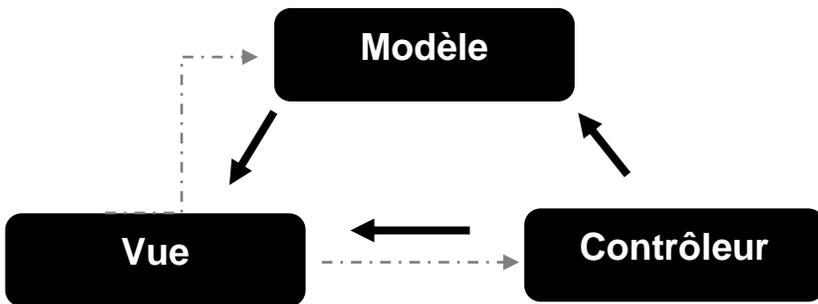


Fig 1 : l'architecture Modèle-Vue-Contrôleur (MVC)

### Modèle

Contient la logique.

### Contrôleur

Interagit avec le modèle pour créer des données à afficher.

### Vue

Restitue le contenu à l'utilisateur et relaie les commandes de l'utilisateur au contrôleur.

**React**, quant à lui est créé et maintenu par Facebook. React est souvent mentionné aux côtés d'autres frameworks JavaScript. Pourtant, **React** n'est pas un framework complet, ou tout autre type de framework.

En effet **React** est une bibliothèque Javascript permettant aux développeurs la création d'interface utilisateurs.

## B. Contexte et points de différences

- **TypeScript vs JavaScript**

**React** et **Angular** sont tous les deux servis par le gestionnaire de paquet tout-puissant **NPM**, et bien sûr les deux technologies produisent in fine du Javascript. Pourtant, là où React se base sur la norme ECMAScript 6, Angular a adopté depuis longtemps la surcouche de Microsoft TypeScript.

**ECMAScript 6** est le standard normalisé qu'implémente JavaScript (à l'instar d'ActionScript). C'est un langage de programmation à typage faible, orienté prototype donc faiblement objet, et événementiel donc massivement asynchrone. Alors qu'Angular utilise **TypeScript** est un langage à typage statique, ce qui signifie que vous devez définir le type de la variable (chaîne, nombre, tableau, etc.). Et sur l'approche objet (tout est classe ou interface, à la mode Java, donc une structure plus forte).

- ✓ Comparaison de propriété entre JavaScript et TypeScript

```
// JavaScript (ES6)
const name;

// TypeScript
const name: string; // <-- static typed!
```

- ✓ Comparaison de fonction entre JavaScript et TypeScript

```
// JavaScript (ES6)
function getName(name, age){
  return name + age;
}

// TypeScript
function getName(name: string, age: number){ // <-- static typed!
  return name + age;
}
```

- **Architecture : L'ultrastructure vs pas de structure**

L'architecture des deux technologies est basée sur des composants. Quand on parle de composant, on parle d'une partie du code indépendante, qui peut être réutilisable dans d'autres projets. A titre d'exemple, la partie « création du compte » ou « connexion » dans une application est presque la même pour toutes les applications.

Bien que leur architecture se base sur des composants, la constitution ainsi que le contenu de ces composants diffèrent :

**Angular** recommande l'utilisation du HTML, CSS, et Javascript des fichiers séparés, il exige une séparation de la partie d'interface utilisateur des composants en tant qu'attributs de balises HTML et leurs comportements sous la forme de code JavaScript. Alors que dans **React** au lieu du HTML, nous devons utiliser JSX (extension syntaxique de JavaScript) avec les méthodes, variables ES6, etc.

## Utiliser des expressions dans JSX

Dans l'exemple suivant, nous déclarons une variable appelée `name` et nous l'utilisons ensuite dans JSX en l'encadrant avec des accolades :

```
const name = 'Khaoula ELg';  
const element = <h1>Bonjour, {name}</h1>;
```

- **La liaison des données (Data Binding): two-way vs one-way- binding**

Une différence importante entre nos deux comparants est la liaison unidirectionnelle par rapport à la liaison bidirectionnelle. **Angular** utilise une liaison bidirectionnelle (**two-way binding**). Par exemple, si vous modifiez l'élément d'interface utilisateur, l'état du modèle correspondant change également. De plus, si vous modifiez l'état du modèle, l'élément de l'interface utilisateur change.

Cependant, **React** n'a qu'une liaison à sens unique (**one-way**). Tout d'abord, l'état du modèle est mis à jour, puis il rend la modification de l'élément d'interface utilisateur. Toutefois, si vous modifiez l'élément d'interface utilisateur, l'état du modèle ne change pas.

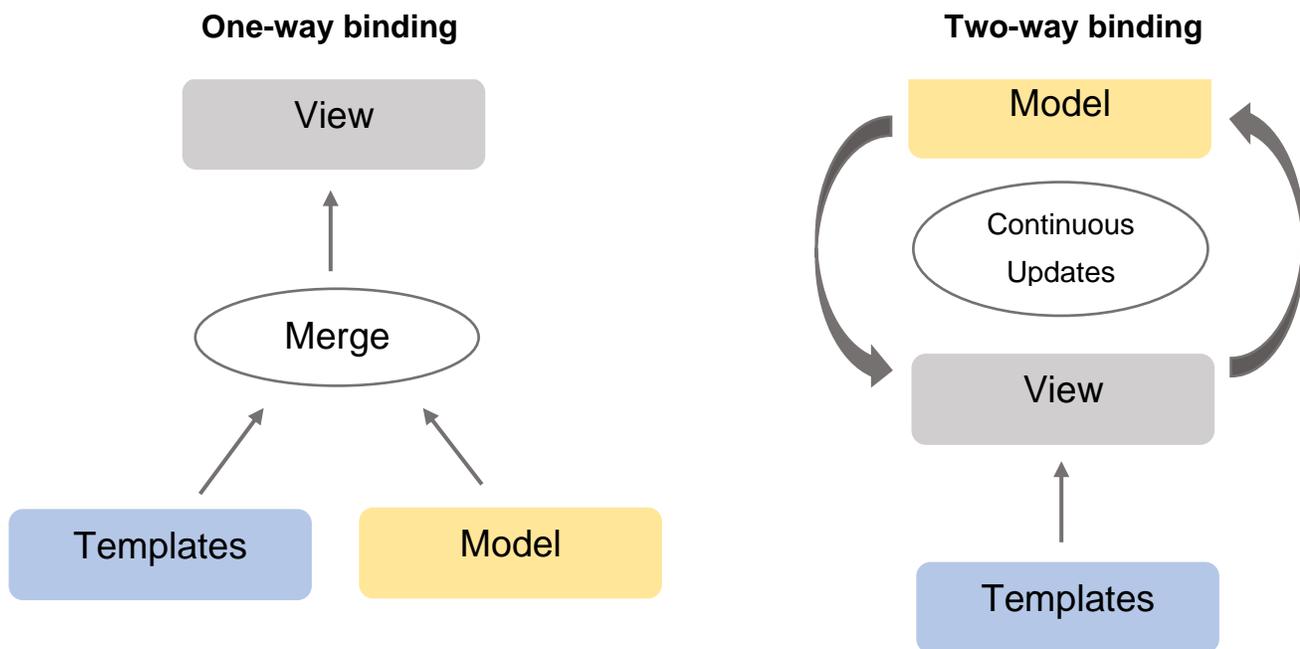


Fig 2 : two-way vs one-way-binding

- **Du web vers le mobile**

Les deux technologies **Angular** et **React** proposent des solutions pour la création d'applications mobiles.

**React Native** est une plate-forme développée par Facebook pour la création d'applications mobiles véritablement natives à l'aide de React. La syntaxe est légèrement différente, mais il y a beaucoup plus de similitudes que de différences.

**Ionic**, en revanche, est un framework de développement d'applications mobiles hybrides. Il utilise un conteneur Cordova incorporé avec **Angular**. Ionic fournit une bibliothèque de composants d'interface utilisateur robuste, facile à configurer et à développer avec des applications mobiles hybrides.

- **Usage au niveau des entreprises**

Les deux, **React** et **Angular** sont utilisés par de grandes entreprises, je vous citerai quelques exemples ci-dessous :

**React**

- Netflix
- WhatsApp
- Facebook
- Airbnb
- Dropbox
- Uber
- Instagram

**Angular**

- Sony
- Google
- Nike
- General Motors
- HBO
- Forbes

## C. Avantages/points forts

### ***Angular***

- Une documentation riche et bien développée permettant aux développeurs de trouver facilement toute information nécessaire.
- Améliorations constantes et régulières par le lancement de nouvelles versions régulièrement par l'équipe Angular.
- Compilation plus rapide (en moins de 3 secondes)
- Modèle MVVM (Model – View- View Model), qui permet aux développeurs de travailler séparément sur la même section d'application en utilisant le même ensemble de données.

- L'association bidirectionnelle de données, qui répercute les effets de chaque modification mineure des données, supprime le besoin de synchronisation supplémentaire des données de la vue et du modèle, et minimise les risques d'erreurs.
- Les avantages du TypeScript : vérification statique et opérationnelle de la saisie, fonctions de saisie très performantes

## **React**

- Possibilité de réutiliser les composants. Tout changement dans l'un ne provoque pas de réaction en chaîne.
- Avec le principe de "one-way data binding", la modification de l'entité enfant n'induit pas une modification de l'entité parent (stabilité du code).
- Beaucoup plus facile à apprendre qu'Angular. React exige de bien revoir ses bases en HTML et ne réclame pas de syntaxe complexe comme l'apprentissage du TypeScript.
- 100% Open Source, beaucoup d'améliorations et de contributions régulières de la communauté des développeurs.
- Intégration rapide pour mobile via React Native.

## **D. Inconvénients/points d'amélioration**

### **Angular**

- Spécificités du TypeScript, qui a une courbe d'apprentissage difficile.
- Les problèmes de migration qui peuvent apparaître lors du passage de l'ancienne version aux plus récentes.
- Syntaxe complexe : La syntaxe est assez difficile à maîtriser pour les juniors.

### **React**

- Manque de documentation officielle.
- React est *unopinionated*, ce qui signifie que les développeurs se trouvent perdus dans la masse de choix qu'ils ont.
- React.js n'est pas un framework, donc au cours du développement, parfois il faut faire appel à d'autres outils ou bibliothèques.

## **E. Conclusion**

D'après ce que nous venons de voir on peut déduire facilement que les deux Angular et React sont parmi les meilleures technologies du marché. Même s'ils ont des différences sur la façon dont un développement d'application doit être abordé, il y a une place pour les deux dans le monde du développement logiciel. Vous ne pouvez pas désigner un gagnant et un perdant dans cette comparaison mais si vous devez en choisir un, vous devez vous décider en fonction de votre besoin et de votre projet :

### **Choisissez React si :**

- Vous recherchez une solution hautement personnalisée qui laisse de la place pour votre projet.

- Votre projet est une application d'une seule page, comme une application de discussion en ligne, une application de visualisation de données.
- Vous envisagez de créer ensuite une application mobile multiplate-forme à l'aide de React Native.

### **Choisissez Angular si :**

- Vous construisez une application Web à page unique à grande échelle.
- L'application va être riche en fonctionnalités et contenir du contenu dynamique.
- Votre objectif est un projet à long terme.
- Vous envisagez de créer une application Web hybride ou progressive au lieu d'une application mobile native ou multiplateforme.

### **Références:**

- <https://blog.arca-computing.fr/javascript-comparatif-angular-react-vue/>
- <https://www.easypartner.fr/blog/que-choisir-entre-react-ou-angular/>
- <https://www.freecodecamp.org/news/a-comparison-between-angular-and-react-and-their-core-languages-9de52f485a76/>
- <https://www.hello-pomelo.com/le-developpement-en-react-et-react-native/>
- <https://openclassrooms.com/fr/courses/4664381-realisez-une-application-web-avec-react-js/4664388-decouvrez-lutilite-et-les-concepts-cles-de-react>
- <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>
- <https://blog.axopen.com/2018/06/react-vs-angular/>
- <https://www.supinfo.com/articles/single/6124-introduction-angular>
- <https://www.sitepoint.com/react-vs-angular/>
- <https://hackernoon.com/reactjs-v-s-angular-924e609b57da>
- <https://fr.reactjs.org/docs/introducing-jsx.html>
- <https://code.tutsplus.com/articles/angular-vs-react-7-key-features-compared--cms-29044>
- <https://programmingwithmosh.com/react/react-vs-angular/>
- <https://blog.impulsebyingeniance.io/angular-vs-react/>