Programmation orienté composants :

**Définition** :

C’est une approche modulaire de développement des applications informatique, afin de crée des unités d’une façon indépendante de production et de déploiement et qui peut être combiné à d’autre unités (composants) pour former une application.

SRC : Programmation orientée composants-Stephane ´ Frenot ´ - TC - 20/12/2002 –

**Structure d’un composant :**

Le code de composants peut être séparées en deux parties :

* Les méthodes qui ne sont pas accessible de l’extérieur (méthodes privée).
* Les méthodes accessible (les interfaces) est le moyen de communication avec le code client, ou chaque composant possède des interfaces fournie et requise.

Les avantages de programmation orientée composants :

* **Spécialisation :** L'équipe de développement peut être divisée en sous-groupes, chacun se spécialisant dans le développement d'un composant
* **Sous traitance :** Le développement d'un composant peut être externalisé, à condition d'en avoir bien réalisé les spécifications au préalable.
* **Facilité de mise à jour :** La modification d'un composant ne nécessite pas la recompilation du projet complet.
* **Facilité de livraison/déploiement :** Dans le cas d'une mise à [jour](https://www.techno-science.net/definition/1509.html), d'un correctif de sécurité, ... alors que le logiciel à déjà été livré au client, la livraison en est facilitée, puisqu'il n'y a pas [besoin](https://www.techno-science.net/definition/686.html) de re-livrer l'intégralité du projet, mais seulement le composant modifié.
* **Choix des langages de développement :** Il est possible, dans la plupart des cas, de développer les différents composants du logiciel dans des langages de programmation différents. Ainsi, un composant nécessitant une fonctionnalité particulière pourra profiter de la [puissance](https://www.techno-science.net/definition/1737.html) d'un langage dans un domaine particulier, sans que cela n'influe le développement de l'ensemble du projet.
* **Productivité :** La réutilisabilité d'un composant permet un gain de productivité non négligeable car elle diminue le [temps](https://www.techno-science.net/glossaire-definition/Temps.html) de développement, d'autant plus que le composant est réutilisé souvent.

Contrainte programmation orienté composants :

Lorsque l'on parle de réutilisation, de facilité de déploiement, c'est que le développement est sinon achevé, du moins bien entamé. Mais factoriser un logiciel en composants nécessite un important travail d'analyse. La rédaction des signatures des méthodes devra être particulièrement soignée, car modifier une signature nécessitera de retravailler toutes les portions de codes du projet qui font appel au composant, et l'on perdrait alors les bénéfices de l'indépendance des briques logicielles.

Le fait de ne pas connaître l'implémentation d'un composant (à moins d'avoir accès au source), peut également gêner certains chefs de projets qui veulent garder un [contrôle](https://www.techno-science.net/definition/2787.html) [total](https://www.techno-science.net/definition/7163.html) sur leur logiciel.

SRC : https://www.techno-science.net/definition/5390.html

**Les modèles à composants :**

A l’heure actuelle, il existe principalement trois fournisseurs de modèles à composants : Sun MicroSystems, Microsoft et l’Object Management group (OMG).

Ces modèles sont classés en trois catégories :

1. Modèle orienté IHM / Client : OLE, COM, ActiveX, JavaBeans SUN, React JS, Angular JS.
2. Modèle orienté métier : ComT, MTS, .NET Microsoft, JavaBeans SUN, NodeJS.
3. Modèle généraliste : Fractale de consortium ObjectWEb, OSGI

**Pour quoi NextJS ?**

* Permet aux développeurs d'écrire facilement des applications universelles avec React de manière transparente, facile et efficace, un routage simple des pages.
* Code splittent : Cela signifie que vous pouvez charger votre code depuis le serveur uniquement quand vous en avez besoin ; vous pouvez le faire par page ou par composant.
* Il offre un contenu hautement interactif pour les utilisateurs.
* Génère le contenu avancé dans le serveur afin que l'utilisateur voie la page html entièrement rendu. Il a trois façons de générer le contenu de la page web ou il offre une grande flexibilité dans développement :

1. Génération statique : générer toutes les pages au temp d’exécution (on utilise cette méthode quand on a des données qui ne change pas souvent).
2. Server SIDE Rendering : générer chaque page au moment de la demande (les données ici change souvent).
3. Génération statique incrémentielle : génère des pages uniques en arrière-plan.

Conclusion : peut importe la technologie que vous avez utilise essay toujours d’etre confortable avec elle