

# L'architecture en couches

# Table des matières

<b>I. Qu'est-ce que l'architecture en couches ?</b>	<b>3</b>
<b>II. Exercice : Quiz</b>	<b>5</b>
<b>III. Différents types d'architecture en couches</b>	<b>6</b>
<b>IV. Exercice : Quiz</b>	<b>11</b>
<b>V. Essentiel</b>	<b>12</b>
<b>VI. Auto-évaluation</b>	<b>12</b>
A. Exercice .....	12
B. Test .....	12
<b>Solutions des exercices</b>	<b>13</b>

## I. Qu'est-ce que l'architecture en couches ?

**Durée :** 1 h

**Prérequis :** avoir suivi le parcours jusqu'ici

**Environnement de travail :** ordinateur connecté à internet

### Contexte

L'architecture logicielle décrit les composants de base d'un système logiciel et leur composition interne. Dans le développement logiciel, l'architecture logicielle est la première décision de conception. Elle est essentiellement déterminée par des critères de qualité tels que la modificabilité, la maintenabilité, la sécurité et les performances. Une fois qu'une architecture logicielle est établie, elle ne peut être modifiée que plus tard avec un effort considérable. Par conséquent, la décision concernant sa conception est l'un des points les plus critiques et les plus importants du processus de développement logiciel.

Pour fonctionner avec succès, l'architecture logicielle doit s'adapter à d'autres éléments du projet logiciel. Une architecture logicielle bien configurée permet aux utilisateurs et aux développeurs de comprendre facilement le système. Les facteurs importants affectant l'adéquation de l'architecture logicielle sont la planification de projet, l'analyse des risques, l'organisation, le processus de développement, le flux de travail, le matériel, l'assurance qualité et les exigences. Les architectures à base de couches se concentrent sur l'attribution hiérarchique des rôles et des responsabilités, offrant une séparation très efficace des tâches. Les rôles représentent des modèles et des types d'interactions avec d'autres couches, et les responsabilités représentent la fonctionnalité en cours de développement. C'est ce que vous allez voir dans ce cours.

Toutes les applications contiennent un code de présentation, un code de traitement de données et un code de stockage de données. L'architecture de l'application varie en fonction de la manière dont le code est distribué. L'objectif est d'unifier les applications PC, les applications client-serveur et les applications web, ce qui est possible pour les applications de toute taille. Aujourd'hui, de nombreuses informations importantes sont stockées dans des applications telles que les systèmes de messagerie et sous la forme d'une série de questionnaires et d'interfaces conçues pour pouvoir accéder à ces informations d'une manière ou d'une autre.

L'architecture en couches est une architecture logicielle constituée d'une conception d'un système sous forme de superposition de strates. Chacune des strates définies aura une responsabilité particulière. La première utilisation de l'architecture en couches fut pour l'architecture client-serveur.

Pour créer des logiciels, il est nécessaire de faire appel à des bibliothèques. Le principe de l'architecture en couches est qu'une bibliothèque spécifique va utiliser des bibliothèques moins spécifiques, qui elles-mêmes seront amenées à utiliser des bibliothèques génériques. En créant une bibliothèque logicielle, le développement deviendra plus efficace et les nouveaux composants devront utiliser les anciens composants et les anciens composants utiliseront des composants encore plus anciens (il s'agit d'un empilement de composants). La cohésion des composants sera permise par la division en couches, il s'agira d'un empilement de paquetage de composants. Les composants des couches supérieures dépendent du fonctionnement des composants des couches en dessous.

### La décomposition des couches se fait en plusieurs parties

Il y a d'abord la partie présentation, il s'agit de la partie au niveau le plus haut de l'architecture en couches. Elle regroupe les fonctions les plus importantes de l'interface et les transforme en ce que l'utilisateur peut utiliser.

Ensuite il y a la couche logique, cette partie consiste à la coordination d'application de processus de commande permettant d'afficher les éléments logiques d'évaluation et de performance de calcul. Elle permet également de faire évoluer les données entre la couche supérieure (présentation) et la couche inférieure (données).

La couche la plus basse est la couche d'accès aux données, il s'agit de toutes les informations qui seront disponibles dans la base de données ou les fichiers système. Ces informations seront transmises à la première couche supérieure (couche logique) et suivant leur utilité seront renvoyées à la couche de présentation si elles sont utiles pour l'expérience utilisateur.

Il faut savoir qu'une architecture logicielle ne sera jamais 100 % adaptée à toutes les exigences d'un projet. Il est donc important de choisir l'architecture qui soit la plus adaptée à votre projet.

Parmi toutes ces architectures se trouve l'architecture en couches. C'est-à-dire que le logiciel sera organisé sous forme de différentes couches et chaque couche ne pourra communiquer qu'avec les couches qui lui sont adjacentes. Toutes les responsabilités seront séparées, ce qui facilitera la compréhension des échanges au sein de l'application. Si chaque couche correspond à un processus distinct alors on parlera d'architecture n-tiers qui désignera le nombre de couches présentes. Une architecture en couches 3-tiers peut correspondre à un navigateur web qui accède à différentes pages dynamiques, qui elles-mêmes accèdent à des informations provenant d'une base de données.

#### Définition

L'architecture n-couche correspond au partitionnement de la logique des applications en couches distinctes. Les 3 couches les plus courantes dans ce type d'architecture sont la couche interface utilisateur, la couche logique métier, la couche accès aux données. Elles seront appelées par couches de niveau présentation (application web), couche de niveau métier (API REST), couche de base de données (base de données SQL). Bien que toutes les couches soient séparées les unes des autres, elles n'en restent pas pour le moins liées de façon logique.

La plupart des applications Java utilisent une architecture en couches. Dans une architecture en couches, les couches sont organisées en couches horizontales. Chaque couche va avoir un rôle spécial dans l'application. Bien sûr, un modèle d'architecture en couches ne peut pas donner exactement le nombre et le type de couches qui devront être utilisés. Pourtant, il est possible de dire que la plupart des architectures en couche se composent de 4 couches standard (présentation, métier, persistance, base de données). Les petites applications nécessiteront uniquement 3 couches alors que d'autres, plus grandes, pourraient contenir 5 couches ou plus.

#### Définition

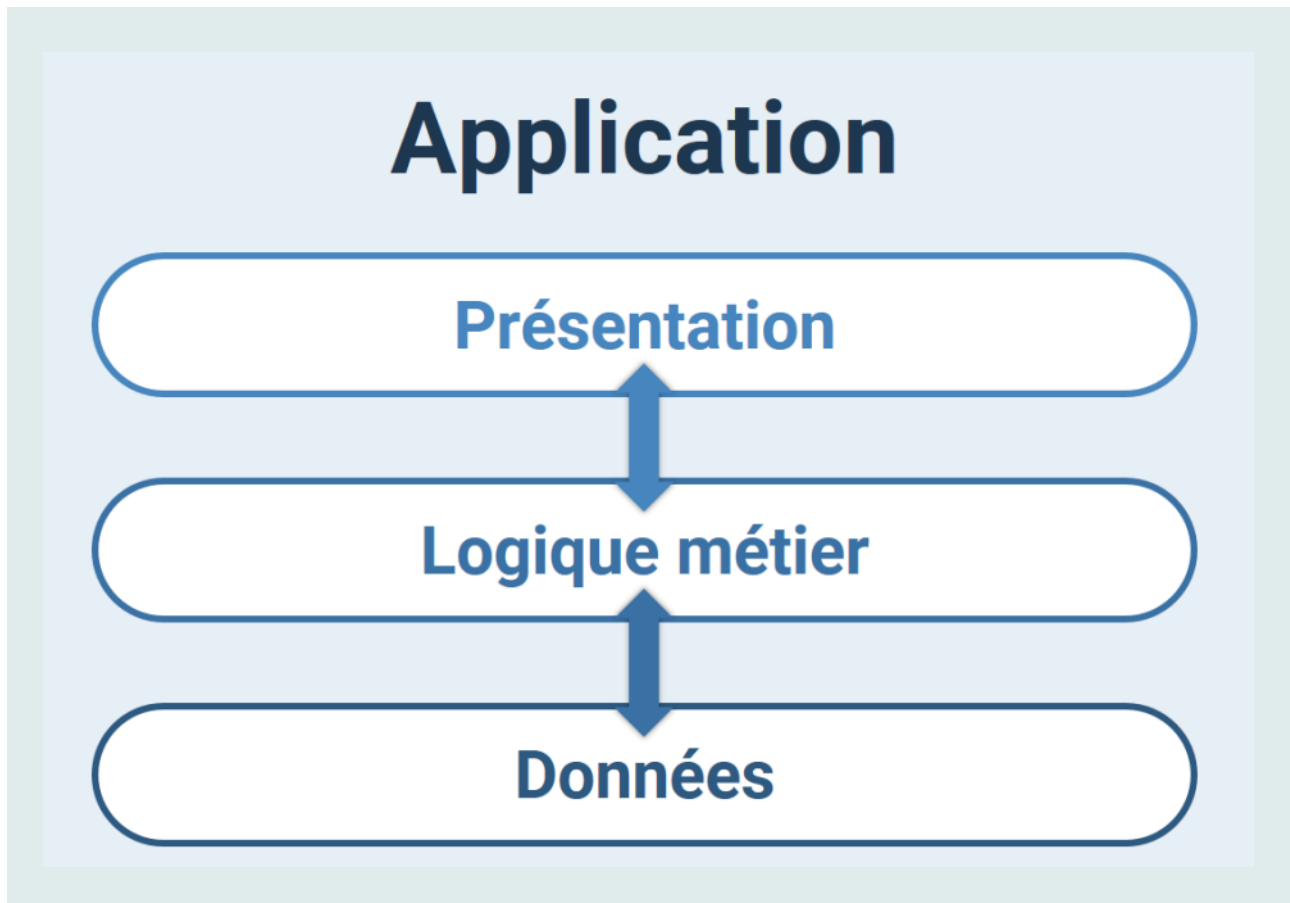
#### Chaque couche va représenter un élément spécifique dans l'application

**La couche présentation** correspond à la logique de communication du navigateur et la gestion de l'interface utilisateur. C'est elle qui va effectuer le dialogue avec l'utilisateur, c'est ce qu'on appelle l'interface homme-machine. Elle correspond à une interface *asp* qui va être exploitée par un navigateur web. Il s'agit d'une interface qui peut avoir différentes facettes tout en gardant la même finalité à l'application. C'est la couche présentation qui va envoyer les requêtes à la couche métier, puis en retour, elle présentera les informations renvoyées par la couche inférieure.

**La couche métier** va permettre d'exécuter la logique métier. Elle transmettra les données à la couche présentation. C'est la couche qui va gérer la logique applicative en mettant en place un ensemble de règles de gestion. C'est elle qui va offrir à la couche présentation des services applicatifs. Pour cela, elle s'appuie sur les données du système donné par la couche inférieure. En retour, elle envoie à la couche présentation le résultat de ses calculs logiques. La couche métier permettra d'exécuter les différentes règles associées à la demande de l'utilisateur. Il faut garder à l'esprit que chaque couche est indépendante de l'autre. En effet, la couche métier n'a pas besoin de se soucier de la façon dont formater les données clients ni même de se demander d'où proviennent les données puisque la couche persistance s'en chargera.

**La couche de persistance** est l'intermédiaire entre la couche métier et la couche base de données.

**La couche de base de données** permet le stockage des données. Il s'agit des données qui sont vouées à être conservées. Pour être stockées, ces données peuvent être placées dans des fichiers texte, des fichiers XML, ou une base de données. C'est cette couche qui donnera l'accès aux données à la couche supérieure.



Les principes communs qui s'appliquent lors de la conception de ce style architectural incluent :

- Abstraction : l'architecture basée sur les couches résume la vue du modèle dans son ensemble, tout en fournissant suffisamment de détails pour comprendre les relations entre les couches.
- Encapsulation : la conception ne fait aucune hypothèse sur les types de données, les méthodes, les propriétés ou les implémentations.
- Fonctionnalité bien définie : la conception définit clairement la séparation entre les fonctions de chaque couche. Les couches supérieures, telles que la couche de présentation, envoient des commandes aux couches inférieures, telles que la couche métier et la couche de données, et les données circulent dans les deux sens.
- Cohésion : chaque couche contient des fonctionnalités directement liées à la tâche de cette couche.
- Réutilisable : les couches inférieures ne dépendent pas des couches supérieures, ce qui permet de les réutiliser dans d'autres scénarios.
- Séparation : la communication entre les couches est basée sur des abstractions qui assurent le découplage entre les couches.

## Exercice : Quiz

[solution n°1 p.15]

### Question 1

La première partie de l'architecture en couches est celle de la présentation.

- ☐ Vrai
- ☐ Faux

### Question 2

La base de données regroupe les fonctions les plus importantes de l'interface et les transforme en ce que l'utilisateur peut utiliser.

- ☐ Vrai
- ☐ Faux

Question 3

La couche la plus basse est la couche d'accès aux données.

- ☐ Vrai
- ☐ Faux

Question 4

Une architecture logicielle ne sera jamais 100 % adaptée à toutes les exigences d'un projet.

- ☐ Vrai
- ☐ Faux

Question 5

Chaque couche peut communiquer avec n'importe quelle couche.

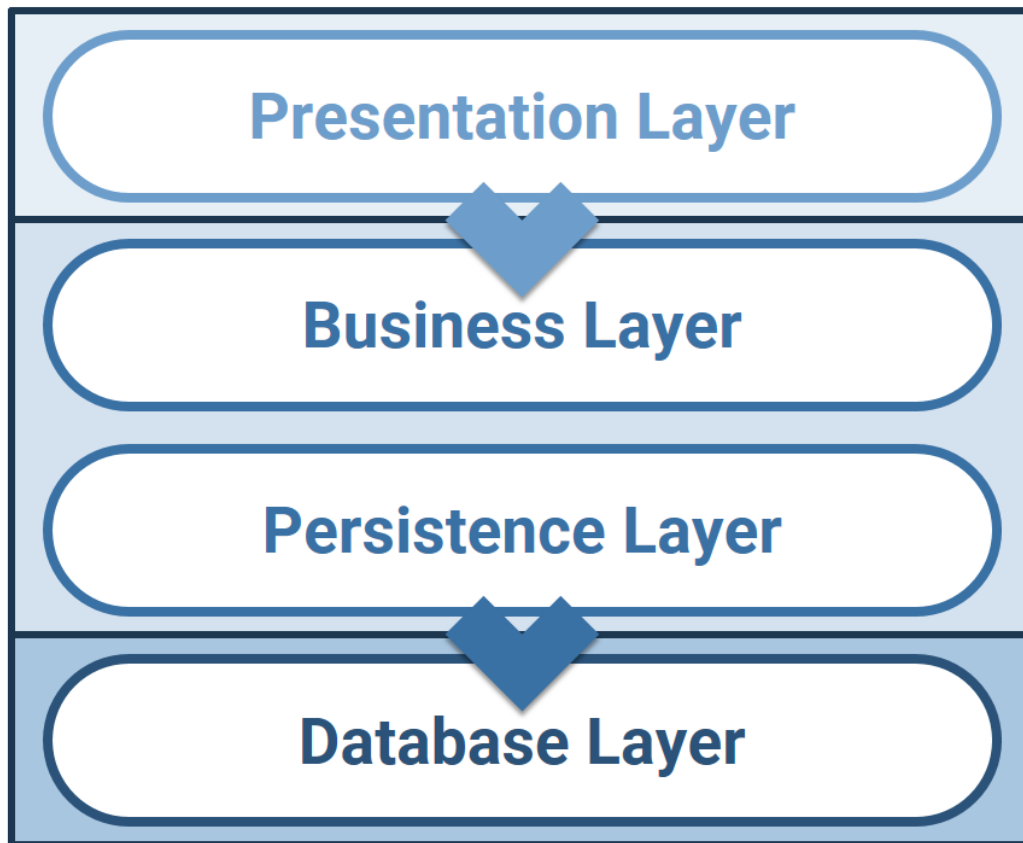
- ☐ Vrai
- ☐ Faux

### III. Différents types d'architecture en couches

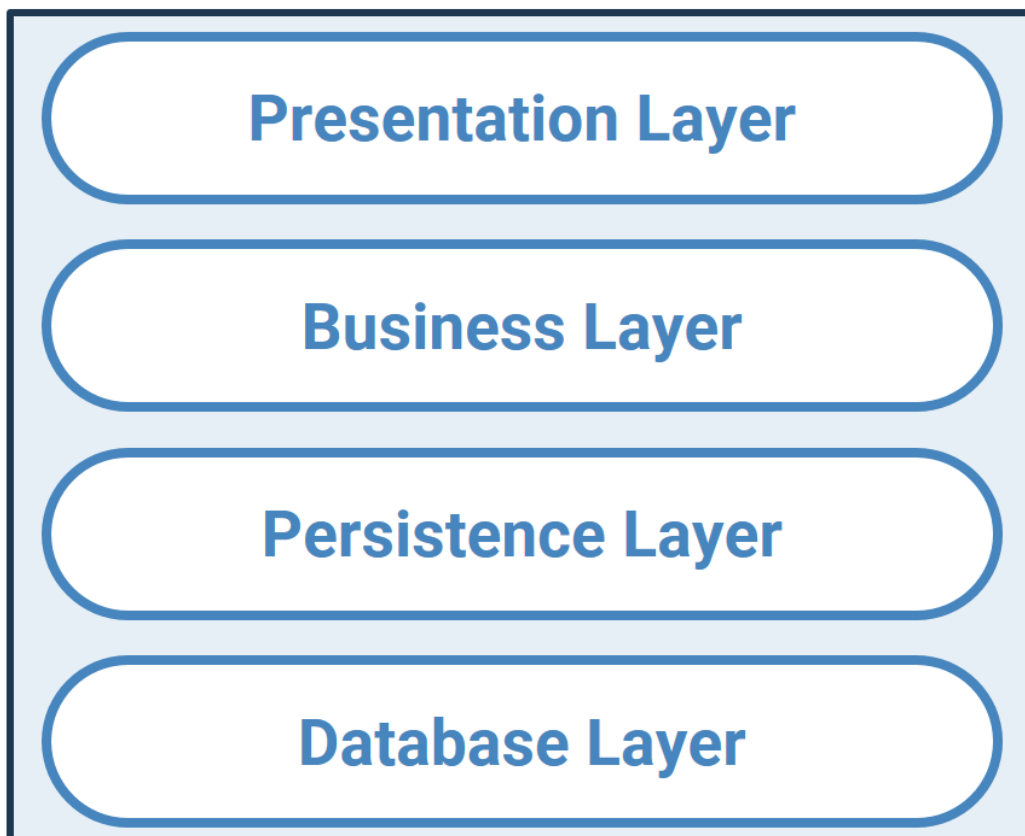
Ces couches sont le plus souvent séparées en trois principaux types de séparation.

L'architecture en couches la plus utilisée consiste en la séparation de la couche base de données des autres couches. La couche de base de données sera déployée physiquement et séparée des autres couches.

Le second style d'architecture en couches consiste en la séparation de la couche présentation et de la couche base de données du reste de l'application.



La dernière consiste en la conservation de toutes les couches dans la même unité physique. C'est l'architecture qui sera le plus souvent utilisée pour les petites applications.



Ce qui caractérise l'architecture en couches est vraiment la séparation entre chaque couche puisque chaque couche ne doit s'occuper que d'elle-même. Toute la logique traitée par une couche ne la concerne qu'elle-même. Les composants de la couche de présentation ne traiteront que de la logique liée à la présentation, les composants qui traitent de la couche métier de traitement que de la logique métier, etc.

L'architecture en couches facilite le développement, le test, la gouvernance, et la maintenance de l'application. En effet, dans ce modèle d'architecture, les composants sont bien définis et ont une portée limitée et cela facilite leur gestion.

L'architecture en couches peut être modifiée par l'ajout de couche supplémentaire. Cela s'appelle l'architecture hybride en couche. C'est le cas notamment de la couche de service entre la couche métier et la couche persistance. Pour que ce modèle fonctionne, il faut changer certains paramètres de notre architecture en choisissant la bonne communication pour toutes les couches. Cette communication pourra être faite entre couche fermée et couche ouverte.

### Complément

#### Architecture en trois couches :

Les services sont placés sur un réseau et fonctionnent ensemble pour prendre en charge un ou plusieurs processus métier. Dans ce modèle, une application devient un ensemble de services utilisateur, métier et de données qui satisfont les processus métier. Étant donné que les services sont conçus pour une utilisation générale et suivent les directives d'interface publiées, ils peuvent être réutilisés et partagés entre plusieurs applications.

Les différentes couches :

- La couche de présentation ou interface utilisateur : cette couche est constituée du formulaire et des contrôles présents dans le formulaire. C'est la couche avec laquelle l'utilisateur interagit.
- Couche métier : elle se compose d'entités qui représentent des objets qui seront manipulés ou utilisés par l'ensemble de l'application. Dans ce cas, ils sont représentés par des classes créées et des « *tables de données* ».
- Couche d'accès aux données : contient des classes qui interagissent avec la base de données, ces classes hautement spécialisées existent dans l'architecture du système et permettent à toutes les opérations de la base de données d'être effectuées de manière transparente pour les utilisateurs de la couche métier à l'aide de procédures stockées générées.

#### Architecture en quatre couches :

L'utilisation de quatre niveaux offre une plus grande évolutivité s'il existe également des clients non web dans le système qui s'exécutent directement sur le serveur de modèles. Cette décomposition en quatre couches permet de déterminer facilement le rôle que joue l'architecture dans une organisation. Chaque couche agit comme un client pour la couche inférieure et comme un serveur pour la couche supérieure. Les architectes ne doivent pas perdre de temps sur des problèmes liés à l'infrastructure, tels que le système d'exploitation. La meilleure façon de séparer l'architecture de l'infrastructure est de garder à l'esprit les quatre couches.

- La couche présentation : cette couche n'effectue aucun calcul, requête ou mise à jour sur le domaine, en fait la couche domaine n'a même pas de visibilité. L'idée de base est de séparer tout ce qui touche à la programmation web de l'application.
- La couche application : responsable de l'accès de cette couche au domaine, simplifiant les informations du domaine en les convertissant en types de données que l'interface peut comprendre, tels que : entiers, nombres réels, chaînes, dates et classes de conteneurs (conteneurs, collections). Une façon d'organiser cette nouvelle couche d'application est de la considérer comme une façade sur le domaine.
- La couche de domaine d'application : illustre les conceptions architecturales qui facilitent le développement d'applications.
- La couche de référentiel : cette couche prend en compte l'infrastructure telle que le matériel ou le réseau physique.



**Architecture en sept couches :**

Les services web sont un ensemble de protocoles et de normes permettant d'échanger des données entre applications. Les normes ouvertes et l'accent mis sur la communication et la collaboration entre les personnes et les applications ont créé un environnement dans lequel les services web deviennent une plate-forme d'intégration d'applications. Ces applications logicielles peuvent être développées dans différents langages de programmation et s'exécuter sur n'importe quelle plate-forme, et grâce à des services, elles peuvent échanger des données entre elles via Internet.

Les différentes couches :

- Couche de présentation : il s'agit de l'interface qui s'affiche dans le navigateur de l'utilisateur final de l'application. Cette couche de présentation fait référence à l'interface de service web et envoie des types de données encapsulés dans des messages connus uniquement de l'interface. Cette couche doit connaître les types de données et les types de messages que la couche d'interface de service peut recevoir.
- Couche d'interface de service : il s'agit de la couche exposée à l'extérieur de l'application. Elle contient le type de message utilisé par la méthode de service web. Elle permet aux applications web et aux applications mobiles de communiquer avec la logique métier.
- La couche d'implémentation de service : dans cette couche se trouve la classe d'adaptateur, qui est responsable de la communication entre la couche de présentation et la couche de logique métier. Étant donné que la logique métier ne traite que des entités métier et que la couche de présentation ne traite que des types de données, il existe une classe de convertisseur responsable de la conversion des entités en types de données et vice versa.
- La couche de type de données : elle contient l'ensemble de la collection de types de données manipulés par la couche de présentation et transmise à l'implémentation du service, où les données deviendront des entités commerciales.
- La couche de logique métier : elle contient le cœur du système. Il s'agit d'un ensemble de routines qui normalisent les actions de l'utilisateur en les envoyant à la couche de stockage des données. Ces informations sont envoyées et encapsulées dans l'entité commerciale.
- La couche d'entité commerciale : il s'agit de la représentation des objets traités dans le système ainsi que des tables de la base de données. Elle permet de transférer des données de l'extérieur vers la base de données et inversement. Elle traite des principes de programmation qui contiennent des objets qui représentent des propriétés qui représentent des données physiques. Elle agit comme un intermédiaire entre la couche de mise en œuvre et le référentiel de données.
- La couche de stockage des données : il s'agit de la couche responsable de l'insertion et de l'extraction des données de la base de données. Le référentiel communique uniquement avec la base de données et la logique métier.

Une seule couche communiquera avec sa couche supérieure immédiate et sa couche inférieure immédiate, sinon le principe de communication inter-couches sera rompu.

**La communication entre les couches****Les couches fermées**

L'un des concepts phares de l'architecture en couches est la possibilité de marquer les couches de l'architecture comme étant fermées. Toutes les modifications qui seront apportées dans une couche de l'architecture n'auront pas d'impact et n'affecteront pas les composants des autres couches puisque celle-ci est fermée. Cela permet de gérer chaque couche indépendamment. En effet, si les couches n'étaient pas fermées, les modifications apportées à la couche de persistance en SQL pourraient avoir un impact sur les couches supérieures. Cela créerait des interdépendances entre les composants, il devient alors plus difficile et plus coûteux de modifier l'application. Lorsque les couches sont fermées, cela signifie que, pour se déplacer dans l'architecture, elles vont devoir passer par les couches juste en dessous afin d'atteindre la couche suivante.

## Les couches ouvertes

La couche ouverte signifie que les demandes sont autorisées pour contourner les couches adjacentes ce qui permet à une couche de communiquer directement avec une couche plus éloignée. Avec une architecture en couches fermées, tous les changements au sein de l'architecture ne touchent que la couche concernée, on parle de couche d'isolation. Dans certains cas, il peut être utile que les couches soient ouvertes.

En effet, si entre la couche métier et la couche persistance on ajoute une couche service et que la couche présentation n'as pas besoin de passer par la couche service, alors on peut la mettre en tant que couche ouverte. C'est-à-dire qu'une requête provenant de la couche présentation ne sera pas obligée de passer par la couche service mais pourra passer directement de la couche métier à la couche persistance.

L'architecture en couches doit donc être bien documentée afin que les développeurs puissent pleinement prendre conscience des couches ouvertes et des couches fermées qui s'y trouvent. Cela les aidera à comprendre les différentes restrictions d'accès aux couches présentes dans l'architecture.

### Les avantages de l'architecture en couches sont les suivants :

- Il s'agit d'une architecture facile à comprendre et simple à mettre en place.
- Chaque couche est protégée des autres couches puisqu'elles sont dépendantes les unes des autres et que chaque couche a une fonction définie.
- Comme les composants sont séparés les uns des autres, les tests sont plus faciles. Les composants peuvent être testés individuellement.
- Son utilisation permet de réduire les coûts.

### Les inconvénients :

- C'est une architecture difficile à faire évoluer puisqu'elle n'est pas faite pour être croissante.
- Bien que chaque couche soit séparée les unes des autres, le système fonctionne comme une seule unité, donc un changement trop spécifique sur une couche peut entraîner des dysfonctionnements sur les autres couches.
- Il existe des interdépendances entre toutes les couches puisque chaque couche dépend de la couche au-dessus d'elle.
- Ce n'est pas un système parallèle, il n'est donc pas possible de faire du traitement parallèle.

L'architecture en couches est recommandée pour les petites applications simples. C'est une architecture qui peut également être utilisée pour des applications qui doivent être construites rapidement car elle est facile à comprendre et à mettre en œuvre. C'est également une architecture qui peut être utilisée si les développeurs n'ont pas beaucoup de notions en architecture logicielle. C'est une architecture qui est beaucoup utilisée pour la mise en place de micro-services.

Les principaux atouts de cette architecture sont sa simplicité d'utilisation et son coût global. Les architectures en couches sont de type monolithique et n'ont donc pas de complexité associée au style d'architecture. Elles sont donc peu coûteuses à réaliser et à entretenir. Par contre, à partir du moment où l'architecture est amenée à évoluer, c'est-à-dire à devenir plus grande et plus complexe, alors l'architecture en couches peut rapidement être dépassée. Un changement de quelques lignes dans une couche peut nécessiter le redéploiement de toute l'unité. Le fait qu'une architecture en couches soit de type monolithique la rend intolérante aux pannes. Cette intolérance est due au manque de modularité de l'architecture. Le temps de réponse global de l'application peut aussi être impacté par ce style d'architecture qui peut aller au démarrage de 2 minutes pour une petite application jusqu'à 15 minutes pour une application de plus grande ampleur.

Comme l'architecture en couches est une architecture simple à mettre en place, elle est peut-être idéale si vous envisagez une application avec peu d'utilisateurs (moins de 200) et que cette application n'est pas vouée à changer dans le temps.

**Complément**

Une architecture en couches permet de gérer tous les composants d'une application (couche métier, couche de présentation et couche de base de données) sous un même toit. L'architecture en couches est le plus courant pour la plupart des logiciels, et son point central est la base de données qui contient des informations. Tout le code de l'architecture est organisé de manière à ce qu'il soit accessible via la couche supérieure, qui transmet la requête à la couche inférieure jusqu'à ce qu'elle atteigne la couche la plus basse. Les services sont placés sur un réseau et fonctionnent ensemble pour prendre en charge un ou plusieurs processus métier. Dans ce modèle, une application devient un ensemble de services utilisateur, métier et de données qui satisfont les processus métier ou dont le processus a besoin pour les prendre en charge. Étant donné que les services sont conçus pour une utilisation générale et suivent les directives d'interface publiées, ils peuvent être réutilisés et partagés entre plusieurs applications. Il convient de souligner les caractéristiques suivantes que cette forme architecturale apporte :

- Utilisation de schémas plus complexes,
- Les données et les services web semblent être séparés,
- Séparer facilement les données de la logique métier,
- Plus de sécurité pour les données d'entreprise,
- Le client reçoit des données et des informations indirectement via le serveur.

**Exercice : Quiz**

[solution n°2 p.15]

## Question 1

L'architecture en couches la plus utilisée consiste en la séparation de la couche base de données des autres couches.

- ☐ Vrai
- ☐ Faux

## Question 2

Chaque couche est concerné par les autres.

- ☐ Vrai
- ☐ Faux

## Question 3

L'architecture hybride en couche n'existe pas.

- ☐ Vrai
- ☐ Faux

## Question 4

L'un des concepts phares de l'architecture en couches est la possibilité de marquer les couches de l'architecture comme étant fermée.

- ☐ Vrai
- ☐ Faux

## Question 5

La couche ouverte signifie que les demandes sont interdites pour contourner les couches adjacentes.

- ☐ Vrai
- ☐ Faux

## V. Essentiel

L'architecture en couches est aussi appelée l'architecture multi-tiers. C'est une architecture logicielle constituée d'une conception d'un système sous forme de superposition de strates. Chacune des strates définies aura une responsabilité particulière. Il faut savoir qu'une architecture logicielle ne sera jamais 100 % adaptée à toutes les exigences d'un projet. Il est donc important de choisir l'architecture qui soit la plus adaptée à votre projet.

Parmi toutes ces architectures, se trouve l'architecture en couches, c'est-à-dire que le logiciel sera organisé sous forme de différentes couches et que chaque couche ne pourra communiquer qu'avec les couches qui lui sont adjacentes. Toutes les responsabilités seront séparées ce qui facilitera la compréhension des échanges au sein de l'application. Dans une architecture en couches, on retrouve la couche présentation, la couche métier, la couche persistance et la couche de base de données. Ce qui caractérise l'architecture en couches est la séparation entre chaque couche puisque chaque couche ne doit s'occuper que d'elle-même. Toute la logique traitée par une couche ne la concerne qu'elle-même. Les composants de la couche de présentation ne traiteront que de la logique liée à la présentation, les composants qui traitent de la couche métier de traitement ne traiteront que de la logique métier, etc.

## VI. Auto-évaluation

### A. Exercice

#### Question 1

[solution n°3 p.17]

Citez les avantages de l'architecture en couches :

#### Question 2

[solution n°4 p.17]

Citez les inconvénients de l'architecture en couches :

### B. Test

#### Exercice 1 : Quiz

[solution n°5 p.17]

##### Question 1

L'architecture n-couche correspond au conditionnement de la logique des applications.

- ☐ Vrai
- ☐ Faux

##### Question 2

La plupart des applications n'utilisent pas d'architecture en couches.

- ☐ Vrai
- ☐ Faux

##### Question 3

Dans une architecture en couches, les couches sont organisées en couches verticales.

- ☐ Vrai
- ☐ Faux

Question 4

L'architecture en couches est facile à comprendre et simple à mettre en place.

- ☐ Vrai
- ☐ Faux

Question 5

Une architecture en couches permet de gérer tous les composants d'une application.

- ☐ Vrai
- ☐ Faux

**Solutions des exercices**




**Exercice p. 5 Solution n°1****Question 1**

La première partie de l'architecture en couches est celle de la présentation.

☒ Vrai

☐ Faux


 La première partie de l'architecture en couches est la partie présentation. C'est la partie au niveau le plus haut.

**Question 2**

La base de données regroupe les fonctions les plus importantes de l'interface et les transforme en ce que l'utilisateur peut utiliser.

☐ Vrai

☒ Faux


 La base de données regroupe les fonctions les plus importantes de l'interface et les transforme en ce que l'utilisateur peut utiliser sont regroupés dans la partie présentation.

**Question 3**

La couche la plus basse est la couche d'accès aux données.

☒ Vrai

☐ Faux


 La couche d'accès aux données est la couche la plus basse. Il s'agit des informations disponibles dans la base de données ou les fichiers système.

**Question 4**

Une architecture logicielle ne sera jamais 100 % adaptée à toutes les exigences d'un projet.

☒ Vrai

☐ Faux


 Comme une architecture logicielle ne sera jamais 100 % adaptée à toutes les exigences d'un projet. Il est donc important de choisir l'architecture la plus adaptée.

**Question 5**

Chaque couche peut communiquer avec n'importe quelle couche.

☐ Vrai

☒ Faux

 Chaque couche ne peut communiquer qu'avec les couches adjacentes.


**Exercice p. 11 Solution n°2**

### Question 1

L'architecture en couches la plus utilisée consiste en la séparation de la couche base de données des autres couches.

☒ Vrai

☐ Faux


 L'architecture en couches la plus utilisée correspond à la séparation de la couche base de données des autres couches. La couche de base de données sera physique et séparée des différentes couches présentes.

### Question 2

Chaque couche est concerné par les autres.

☐ Vrai

☒ Faux


 Ce qui caractérise l'architecture en couches est la séparation entre chaque couche puisque chaque couche n'est concernée que par elle-même.

### Question 3

L'architecture hybride en couche n'existe pas.

☐ Vrai

☒ Faux


 L'architecture en couches peut être modifiée par l'ajout de couche supplémentaire. C'est l'architecture hybride en couche.

### Question 4

L'un des concepts phares de l'architecture en couches est la possibilité de marquer les couches de l'architecture comme étant fermée.

☒ Vrai

☐ Faux


 L'un des concepts importants de l'architecture en couches est celui des couches de l'architecture fermées. Les modifications apportées dans une couche de l'architecture n'auront pas d'impact sur les composants des autres couches.

### Question 5

La couche ouverte signifie que les demandes sont interdites pour contourner les couches adjacentes.

☐ Vrai

☒ Faux

 La couche ouverte signifie que les demandes sont autorisées permettant ainsi de contourner les couches qui lui sont adjacentes.



**p. 12 Solution n°3**

Il s'agit d'une architecture facile à comprendre et simple à mettre en place. Chaque couche est protégée des autres couches puisqu'elles sont dépendantes les unes des autres et que chaque couche a une fonction définie. Comme les composants sont séparés les uns des autres, les tests sont plus faciles. Les composants peuvent être testés individuellement et cette utilisation permet de réduire les coûts.

**p. 12 Solution n°4**

C'est une architecture difficile à faire évoluer puisqu'elle n'est pas faite pour être croissante.


Bien que chaque couche soit séparée l'une de l'autre, le système fonctionne comme une seule unité donc un changement trop spécifique sur une couche peut entraîner des dysfonctionnements sur les autres. Il existe des interdépendances entre toutes les couches puisque chacune dépend de la couche au-dessus d'elle. Ce n'est pas un système parallèle, il n'est donc pas possible de faire du traitement parallèle.

**Exercice p. 12 Solution n°5****Question 1**

L'architecture n-couche correspond au conditionnement de la logique des applications.

☒ Vrai

☐ Faux


 L'architecture n-couche correspond au conditionnement de la logique des applications en couches distinctes.

**Question 2**

La plupart des applications n'utilisent pas d'architecture en couches.

☐ Vrai

☒ Faux


 La plupart des applications, notamment Java, utilisent une architecture en couches.

**Question 3**

Dans une architecture en couches, les couches sont organisées en couches verticales.

☐ Vrai

☒ Faux

 Elles sont organisées en couches horizontales, chacune a son rôle.

**Question 4**

L'architecture en couches est facile à comprendre et simple à mettre en place.

☒ Vrai

☐ Faux

Q C'est un avantage non négligeable de l'architecture en couches.

#### Question 5

---

Une architecture en couches permet de gérer tous les composants d'une application.

☒ Vrai

☐ Faux

Q Elles peuvent toutes être gérées au même endroit.