

Architecture n-tiers

Université Française d'Égypte

Version 0.9 – 14/10/12

Richard Grin

Une application

- Elle comprend
 - des données persistantes (conservées entre 2 sessions de travail)
 - une interface utilisateur (pour récupérer des données entrées par l'utilisateur et lui présenter les résultats des traitements)
 - des traitements à effectuer sur les données manipulées par l'application

R. Grin

Architecture n-tiers

2

Application d'entreprise

- Application de grande taille qui :
 - utilise plusieurs bases de données
 - est répartie sur plusieurs ordinateurs
 - utilise des services non fonctionnels : sécurité d'accès, transactionnel, gestion de la concurrence,...
 - peut soutenir une charge importante et s'adapter à une augmentation de cette charge

R. Grin

Architecture n-tiers

page 3

Les traitements d'une application

- L'interface homme-machine : présente les informations à l'utilisateur et récupère les données que celui-ci a saisies
- Les traitements relatifs au domaine de l'application avec les règles de gestion en vigueur
- L'enregistrement, la récupération et la gestion des données dans la base de données (ou les fichiers)

R. Grin

Architecture n-tiers

4

Les traitements relatifs au domaine de l'application (1/2)

- Ce sont les « vrais » traitements : ils font le travail essentiel
- L'interface homme-machine fait seulement de la présentation
- La gestion des données sert seulement à rendre les données persistantes entre 2 manipulations

R. Grin

Architecture n-tiers

5

Les traitements relatifs au domaine de l'application (2/2)

- Types de traitements « métiers » : gestion d'un compte en banque, gestion du personnel, comptabilité, etc.
- Ils nécessitent des traitements (services) techniques, indispensables mais pas particuliers à une application, pour gérer la sécurité, le transactionnel, la concurrence,...

R. Grin

Architecture n-tiers

6

Applications distribuées

- Une application d'entreprise est très rarement contenue sur une seule machine
- Ne serait-ce que parce que la base de données est partagée entre de nombreux utilisateurs
- Les applications distribuées séparent au moins
 - le code qui gère les données persistantes
 - et le code pour l'interface utilisateur

R. Grin

Architecture n-tiers

7

Modèle 2-tiers (client-serveur)



- Une application cliente s'occupe de l'interface homme-machine
- Un serveur (le plus souvent un SGBD) s'occupe de l'enregistrement, de la récupération et de la gestion des données
- Les traitements « métier » sont répartis entre l'application cliente (le plus souvent) et le serveur de données

R. Grin

Architecture n-tiers

8

Types de modèle 2-tiers

- Avec client lourd : c'est le client qui s'occupe des traitements particuliers à l'application
- Avec client léger : c'est le SGBD qui s'occupe des traitements particuliers à l'application (procédures stockées en particulier)

R. Grin

Architecture n-tiers

9

Problèmes du modèle 2-tiers avec clients lourds

- Déploiement et mise à jour difficiles des clients sur tous les postes clients
- Manque de modularité : services techniques imbriqués dans les différents traitements
- Risque de dépendance des processus métier vis-à-vis de la couche de présentation donc réutilisation plus difficile

R. Grin

Architecture n-tiers

10

Problèmes du modèle 2-tiers avec clients légers

- Forte dépendance des processus métier vis-à-vis de la couche de persistance (portabilité des procédures stockées en particulier) donc réutilisation plus difficile
- Manque de modularité : services techniques imbriqués dans les différents traitements

R. Grin

Architecture n-tiers

11

Modèle 3-tiers (ou n-tiers)



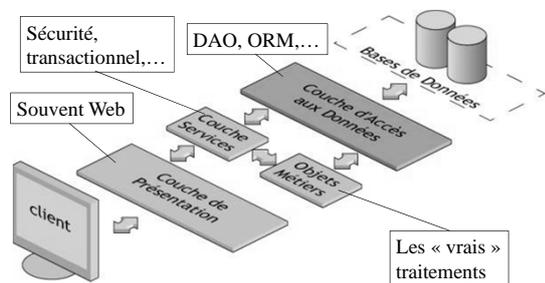
- Serveur d'application héberge les objets métier et s'occupe des traitements techniques :
 - transactions
 - sécurité
 - ...

R. Grin

Architecture n-tiers

12

Couches du serveur d'application



R. Grin

Architecture n-tiers

page 13

La couche de présentation

- Serveur Web inclus ou non dans le serveur d'application dans le cas de clients Web
- Composants serveur qui représentent l'IHM sur le serveur dans le cas d'utilisation de frameworks de type JSF

R. Grin

Architecture n-tiers

page 14

La couche de persistance

- Au moins la base de données
- Le plus souvent on y ajoute une couche qui effectue la correspondance (*mapping*) entre les objets et la base de données : DAO, ORM (par exemple JPA)
- Cette couche sert aussi de cache pour les objets récupérés dans la BD et améliore ainsi les performances

R. Grin

Architecture n-tiers

page 15

Avantages du modèle 3-tiers

- Client facile à installer (surtout si client Web)
- Encapsule les objets métier ; vers des objets métiers réutilisables...
- Facilite la programmation des objets métier qui n'ont pas à s'occuper de la sécurité, de la gestion des transactions

R. Grin

Architecture n-tiers

16

Quelques désavantages...

- Risque de dépendance vis-à-vis des serveurs d'application qui ne sont pas totalement standardisés
- Lourdeur de mise en place de l'environnement
- Mise au point parfois difficile des applications (sources d'erreurs multiples, les outils de mise au point pas vraiment multi-tiers)
- Coûts parfois élevés (serveur d'application et support pour leur utilisation)

R. Grin

Architecture n-tiers

17

Mais ça s'améliore

- Avec le temps et les nouvelles versions :
 - standardisation accrue
 - outils de meilleur qualité
 - serveurs d'application open-source avec des frais de maintenance limités

R. Grin

Architecture n-tiers

18

Indépendance vis-à-vis de la couche présentation

- Aucune des autres couches ne doit dépendre de la couche présentation
- En effet, la couche présentation est souvent modifiée (par exemple pour suivre les remarques et demandes des utilisateurs) et cela ne doit pas induire des modifications dans les autres couches

R. Grin

Architecture n-tiers

page 19

Indépendance de la couche métier

- La couche métier ne doit pas dépendre non plus de l'**implémentation** de la couche de persistance
- Le plus souvent un processus métier reste stable durant toute la vie de l'application
- On ne doit le modifier que pour l'améliorer ou le corriger mais pas pour tenir compte d'une modification (souvent technique) dans l'une des autres couches

R. Grin

Architecture n-tiers

page 20

« Fuites » de la logique métier vers la couche de persistance

- La couche de persistance doit dépendre le moins possible de la couche « métier »
- Une indépendance totale est évidemment impossible mais une bonne indépendance (par l'utilisation d'un framework ORM par exemple) permettra de limiter le travail de maintenance dans la couche de persistance en cas d'une modification du modèle objet métier

R. Grin

Architecture n-tiers

page 21

Contenu du cours

- Objets entités JPA pour la persistance
- Objets EJB pour les traitements métier
- JSF pour la couche de présentation
- Servlet pour l'implémentation de bas niveau

R. Grin

Architecture n-tiers

22