République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et la Recherche Scientifique
Université d'Abou Bekr Beklkaid-Tlemcen
Faculté des Sciences
Département d'Informatique
Spécialité: 3 ème Ingénieur: Réseaux.
Matière: Modélisation des Systèmes d'Information

Automne 2025 13:30-15:00

TD N° 4: Styles architecturaux

I. Objectifs du TD

L'objectif de ce TD est de :

- Identifier et comprendre les différents styles architecturaux des logiciels
- Appliquer les concepts d'architecture logicielle à des cas pratiques
- Analyser les avantages, les limites et les contextes d'utilisation de chaque style
- Explorer les tendances modernes en architecture logicielle
- Proposer une architecture adaptée à un besoin réel ou à un mini-projet

Exercice 1: Architecture web et cloud

- Quelle différence entre une application web et une application cloud ?
- Donnez un exemple d'architecture cloud.
- Citez deux services cloud courants utilisés dans les SI.

Exercice 2: Identification des principaux composants cloud utilisés par Jumia (ou un site similaire).

- Identifiez les principaux composants cloud utilisés par Jumia (ou un site similaire).
- Expliquez comment le scaling automatique permet de gérer les pics de trafic.

- Décrivez la stratégie de sauvegarde et de récupération typique d'un site e-commerce.

Exercice 3: Identification d'une architecture SI réelle

Une entreprise de livraison en ligne (type Jumia Express) possède :

- Une application mobile pour les livreurs.
- Un site web pour les clients.
- Une API centrale pour gérer les commandes et le suivi.
- Une base de données centralisée sur le cloud.
- Identifiez le type d'architecture utilisé.
- Représentez cette architecture sous forme d'un schéma.
- Expliquez deux avantages de ce choix.

Exercice 4: Choix d'architecture adaptée

Associez à chaque besoin ci-dessous l'architecture SI la plus adaptée, et justifiez votre choix :

Besoin	Architecture possible	Justification
Application bancaire interne sécurisée	?	?
Application e-commerce internationale	?	?
Logiciel local de gestion comptable	?	?
Plateforme d'enseignement en ligne	?	?

Figure 1: Tableau Choix d'architecture logicielle

Exercice 5:

Un système d'information universitaire doit gérer :

- Les inscriptions, notes et emplois du temps.
- Les connexions simultanées de 5 000 étudiants.
- Des applications internes pour enseignants et administrateurs.
- Quelle architecture recommandez-vous ?

- Proposez un modèle simplifié de cette architecture.
- Comment garantir la sécurité et la disponibilité du système ?

Exercice 6: Cloud computing

Une entreprise souhaite migrer son SI local vers le cloud. Ses besoins : disponibilité 24/7, sauvegarde automatique, sécurité et coûts maîtrisés.

- Quel type de service cloud recommanderiez-vous (**IaaS, PaaS, SaaS**) ?
- Citez deux fournisseurs possibles.
- Décrivez les étapes de migration du système.

Exercice 7: Intégration et interopérabilité

Une entreprise possède trois applications indépendantes :
- Gestion des clients (CRM).
- Gestion des commandes (ERP).
- Gestion des paiements (Finance).

Elle souhaite que toutes communiquent entre elles.
- Quelle approche d'intégration recommandes-tu ?
- Donnez un exemple technique concret.
- Quels sont les avantages et les limites de cette intégration ?

Corrigé des exercices

- **Corrigé exercice 1**

Différence entre une application web et application cloud:

- Application web : hébergée sur un serveur et accessible via un navigateur.

- Application cloud : utilise une infrastructure virtuelle (IaaS, PaaS, SaaS) hébergée sur des plateformes comme AWS, Azure ou GCP.

Types du cloud:

-**Cloud privé:** interne à l'entreprise

-**Cloud publique:** AWS, GCP, Microsoft Azure.

- **Cloud hybride:** combinaison du cloud privé et publique

- **Corrigé exercice 2**

  Composants cloud :

  - Serveurs applicatifs hébergés sur AWS EC2.
  - Stockage d'images sur Amazon S3.
  - CDN pour la mise en cache des contenus (CloudFront).
  - Bases de données managées (RDS, DynamoDB).

  Scaling automatique :

  - AWS Auto Scaling ajoute ou retire des serveurs selon la charge réseau.

  Cela permet de maintenir les performances même pendant les soldes ou promotions.

  Sauvegarde & récupération :

  - Backups quotidiens automatisés vers le cloud.
  - Réplication des données sur plusieurs zones géographiques (multi-region redundancy).
  - Plan de reprise d'activité (PRA) défini avec monitoring et alertes.

- **Corrigé exercice 3**

  Type d'architecture :

  → Architecture n-tiers (souvent 3-tiers) hébergée dans le cloud.

  Schéma logique :

  [Client Web / Mobile]
  ↓ [API Gateway / Serveur Applicatif]
  ↓ [Base de Données Cloud (MySQL/AWS RDS)].

Avantages :

- **Scalabilité** : le système s'adapte aux volumes de commandes.
- **Interopérabilité** : l'API permet de connecter plusieurs plateformes (mobile, web).

- **Corrigé exercice 4**

Besoin	Architecture	Justification
Application bancaire interne sécurisée	**Client-serveur privé**	Sécurité forte, données centralisées dans un intranet.
Application e-commerce internationale	**Web + Cloud**	Disponibilité mondiale, montée en charge facile.
Logiciel local de gestion comptable	**Monolithique locale**	Simple, fonctionne sans réseau, maintenance interne.
Plateforme d'enseignement en ligne	**3-tiers / Web**	Séparation logique (interface, logique, données) et accès via navigateur.

Figure 2: Choix d'architecture

- **Corrigé exercice 5**

- **Architecture recommandée : → Architecture web 3-tiers :**

- **Interface** : portail web et mobile.

- **Logique** : serveur applicatif (Spring Boot / Django).

- **Données** : SGBD central (PostgreSQL).

- **Modèle :** [Étudiants / Enseignants]
↓ [Serveur Web – Interface (HTML, JS)]

↓ [Serveur Applicatif (API REST)]
↓ [Base de Données (PostgreSQL)]

- Sécurité & disponibilité :

- Authentification LDAP / SSO.
- Chiffrement SSL.
- Réplication de base de données et sauvegardes automatiques.
- Load balancing (HAProxy, Nginx).

- **Corrigé exercice 6**

Service recommandé : PaaS (Platform as a Service) → équilibre entre gestion simplifiée et flexibilité.

Exemple: hébergement sur **Azure App Service ou AWS Elastic Beanstalk.**

Fournisseurs :

- Amazon Web Services (AWS).
- Microsoft Azure.
- Google Cloud Platform (GCP).

Étapes de migration :

- Audit du système existant (bases, serveurs, sécurité).
- Sauvegarde et exportation des données locales.
- Déploiement sur une infrastructure cloud (machines virtuelles ou conteneurs).
- Tests de performance et validation de la connectivité.
- Formation du personnel et bascule progressive.

- **Corrigé exercice 7**

Approche : Utiliser un middleware orienté services (SOA) ou une architecture à base d'API REST.

Exemple concret :

- Chaque application expose une API :
- Clients pour le CRM
- Commandes pour l'ERP
- Paiements pour la finance
- Un bus applicatif (ESB) comme MuleSoft ou WSO2 synchronise les échanges.

Avantages :

- Réutilisation des services.
- Flexibilité et évolutivité.
- Réduction des doublons de données.

Limites :

- Complexité de mise en œuvre.
- Besoin de normalisation des formats de données (JSON, XML).

***B*on courage**