Synthèse — Introduction aux architectures de données

1. Pourquoi une architecture de données ?

Une entreprise utilise de nombreux systèmes : ERP, CRM, site web, applications mobiles, caisses, outils marketing, fichiers Excel, objets connectés, etc.

Ces systèmes peuvent produire des données :

- dispersées dans des **silos** ;
- stockées dans des formats différents ;
- mises à jour à des fréquences différentes ;
- dupliquées ou contradictoires ;
- difficiles à relier entre elles ;
- insuffisamment documentées ou sécurisées.

Une architecture de données vise à organiser le parcours de la donnée afin de la transformer en information fiable, puis en décision métier.

> **Une architecture de données n'est pas seulement un ensemble d'outils : elle comprend des systèmes, des flux, des règles, des responsabilités et des usages.**

2. Le parcours de la donnée

Le parcours décisionnel peut être représenté ainsi :

```
Sources → Ingestion → Stockage → Traitement → Consommation → Décision
```

Brique	Question essentielle	Exemples
Sources	D'où vient la donnée ?	ERP, CRM, site web, caisse, API, fichier Excel
Ingestion	Comment la donnée entre-t-elle dans l'architecture ?	ETL, ELT, API, réplication, événements
Stockage	Où et sous quelle forme est-elle conservée ?	Base relationnelle, Data Warehouse, Data Lake, Lakehouse
Traitement	Comment est-elle nettoyée, enrichie ou agrégée ?	SQL, règles métier, pipelines, traitements Big Data
Consommation	Qui utilise la donnée et pourquoi ?	BI, reporting, application, IA, recommandation
Décision	Quelle action est prise ?	Ajuster les stocks, lancer une campagne, modifier un prix

3. Les fonctions transverses

Elles concernent l'ensemble de l'architecture :

- **Qualité :** exactitude, complétude, cohérence, fraîcheur et unicité ;
- **Sécurité :** droits d'accès, chiffrement, sauvegardes et traçabilité ;
- **Gouvernance :** règles, responsabilités, définitions communes et conformité ;
- **Métadonnées :** origine, signification, propriétaire et transformations de la donnée ;
- **Supervision :** suivi des flux, alertes, journaux et reprise sur incident.

> Une architecture peut fonctionner techniquement tout en produisant des résultats inutilisables si la qualité et la gouvernance sont insuffisantes.

4. Les principaux enjeux

Lorsqu'une architecture est conçue ou évaluée, il faut notamment prendre en compte :

- **l'intégration :** faire communiquer des systèmes hétérogènes ;
- **la qualité :** produire des données fiables et cohérentes ;
- **la fraîcheur :** disposer de données suffisamment récentes ;
- **la scalabilité :** absorber l'augmentation des volumes et des usages ;
- **la performance :** répondre rapidement aux requêtes ;
- **la sécurité et la conformité :** protéger les données sensibles ;
- **l'interopérabilité :** permettre aux systèmes de fonctionner ensemble ;
- **le coût et la complexité :** rester cohérent avec les moyens de l'entreprise.

5. Quelques familles d'architectures

Ces familles ne sont pas exclusives. Une entreprise peut les combiner.

Famille	Idée principale	Besoin privilégié
Architecture centrée sur les données	Un référentiel ou une plateforme de données partagée	Cohérence et vision commune
Architecture événementielle	Les systèmes réagissent à des événements	Temps réel et réactivité
Architecture orientée services / microservices	Les fonctionnalités sont réparties entre services autonomes	Modularité et évolutivité applicative
Data Mesh	Les domaines métier sont responsables de leurs produits de données	Responsabilisation et décentralisation
Lambda / Kappa	Organisation de traitements batch et/ou temps réel	Historique et données récentes
Data Lakehouse	Association de la souplesse du Data Lake et des garanties du Data Warehouse	Données variées, SQL, BI et IA

> Le choix d'une architecture dépend du métier, des volumes, de la fraîcheur attendue, des compétences, du budget et des contraintes de sécurité.

6. Architecture applicative et architecture de données

Architecture applicative	Architecture de données
Organise les logiciels, services et échanges entre applications	Organise la localisation, la circulation, la transformation et la fiabilité des données
Exemples : microservices, API Gateway	Exemples : Data Warehouse, Data Lake, modèle en étoile

Elles sont complémentaires, mais ne répondent pas à la même question.

7. Les idées essentielles à retenir

- Une même donnée peut être présente dans plusieurs systèmes : il faut gérer les doublons et l'identité commune.
- Un système peut être à la fois une source, un stockage opérationnel et un consommateur.
- Toutes les données ne nécessitent pas le même niveau de fraîcheur : batch, quasi temps réel ou temps

réel.
- Il faut définir qui est responsable de la donnée et quelle source fait foi.
- Une donnée destinée à l'IA doit être accessible, documentée, suffisamment fraîche, de qualité et protégée.
- Une architecture moderne n'est pas nécessairement la meilleure : elle doit être adaptée au besoin réel.
- L'objectif final est de passer de données hétérogènes à des décisions fiables et actionnables.

Question de synthèse

Pour répondre de manière fiable à une question métier, il faut :

```
identifier les sources,
organiser les flux,
choisir les stockages adaptés,
transformer et contrôler les données,
les rendre accessibles aux bons utilisateurs,
et définir les règles de qualité, de sécurité et de gouvernance.
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/eadl/bloc5/fm1/synthese-td

Last update: **2026/08/30 20:46**