•Les données semi-structurées

- *L3 SI*
- *« Chapitre 1 : Contexte et problématiques»*

2022-2023

Le contenu

- Chapitre 1 : *Contexte et problématique, Documents et hyper documents multimédias*
- Chapitre 2 : *Noyau XML*
- Chapitre 3 : *Galaxie XML*
- Chapitre 4 : *BD XML et BD semi-structurées*
- Chapitre 5 : *XQUERY et les BD*

Objectifs

Cette matière va permettre aux étudiants :

- Faire la différence entre les données **structurées/semi-structurées** et **non structurées** (introduction pour le BIG DATA …).
- Avoir des connaissances liées aux formats d'échanges et de transportation de données entre les systèmes d'informations (XML, JSON, CSV).
- Savoir utilisez les technologies XML (XPath, XQuery….).

Chapitre 1
Concepts de base

- Les données informatiques
- Problèmes et nouveaux besoins.
- Documents structuré, semi-structuré, non structuré.
- Introduction XML

Explosion de données manipulées par les SI : type, volume, ….

- **Les données d'une application** : les utilisateurs, les produits, historiques d'opérations => stockées généralement dans des BD.
- **Les documents informatiques** : **Hypertexte**, multimédias, hypermédia…
- **Les réseaux sociaux : c**ommentaires, opinions, …
- **Les données géospatiales : g**oogle Maps.

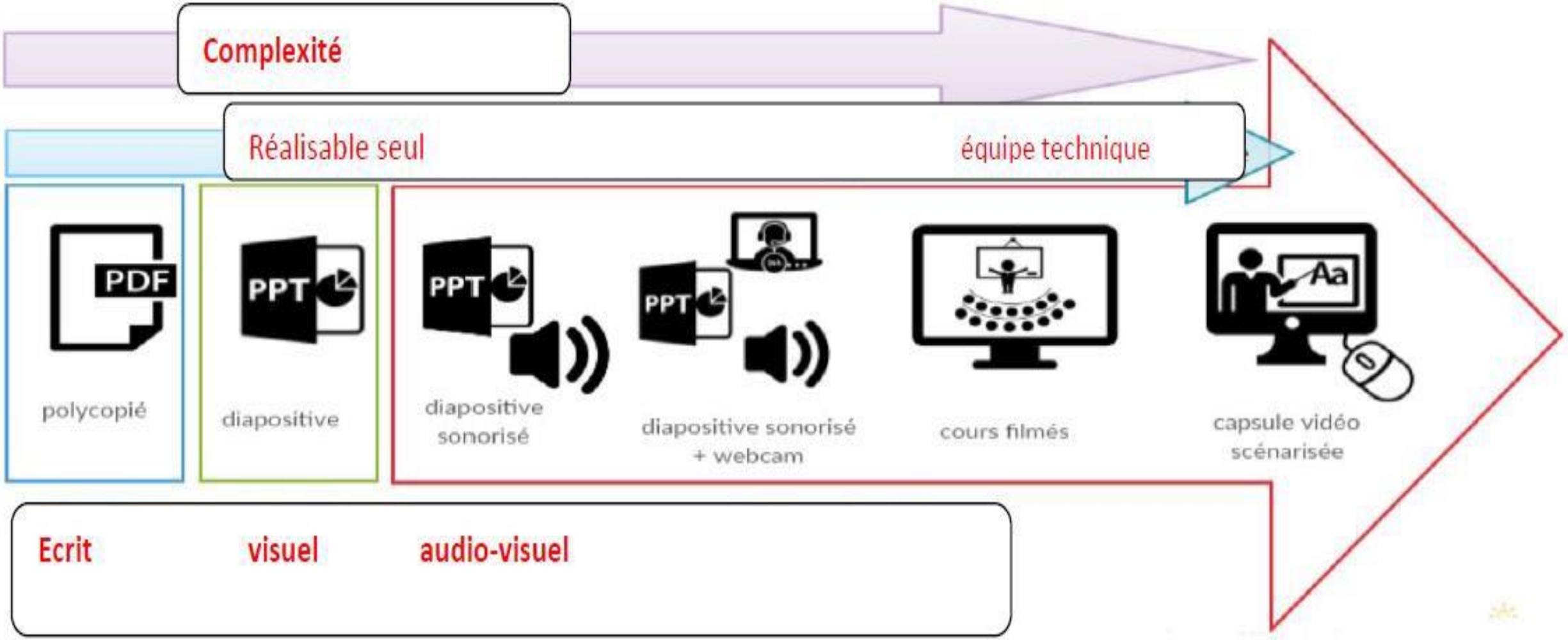
1. Les données spécifiques du WEB

- En plus les données habituelles (multimédias, clients,….),stockées au serveur ,on a besoins de stocker et manipuler les données **concernant le client** :

Ex: Les données de **l'historique de navigation** des utilisateurs (les cookies, les **sessions**, **les champs de formulaire de connexion**,…).

- **Utilisation :** enregistrer les activités des user (comme le contenu d'un panier d'achat d'une session précédente), afin de **Personnaliser** les **préférences** du site, **recommandations**, …..

2. Les documents multimédias



Les documents : hypertexte, et hypermédia

- **Hypertexte :** Système de renvois permettant de passer directement d'une partie d'un document à une autre (**nœuds et de liens**), ou d'un document à d'autres document (lecture non linéaire grâce à la présence de liens entre les documents).
- **Hypermédia** : Extension de l'hypertexte à des données multimédias, permettant d'inclure des liens entre des éléments textuels, visuels et sonores.

Exemple : Encyclopédie Médico-Chirurgicale (EMC)

Les problèmes commencent, ….

- **Plus de données** => **des bonnes décision**, meilleurs expérience utilisateur (*user experience*), compétitivité,……
- Comment **stocker** ces données ?
- Comment **Échanger** ces données avec les différentes **applications** du SI ?
- Comment **analyser** ces données ?

Stockage de données

- Différentes solutions :
  - Les structures volatile (tableaux, matrices, ….)
  - Les fichiers (textes, multimédias, …)
  - **Les bases de données**
  - Les DataWerhouse (pour des besoins d'analytique)

Historique des BD (selon le modèle de données)

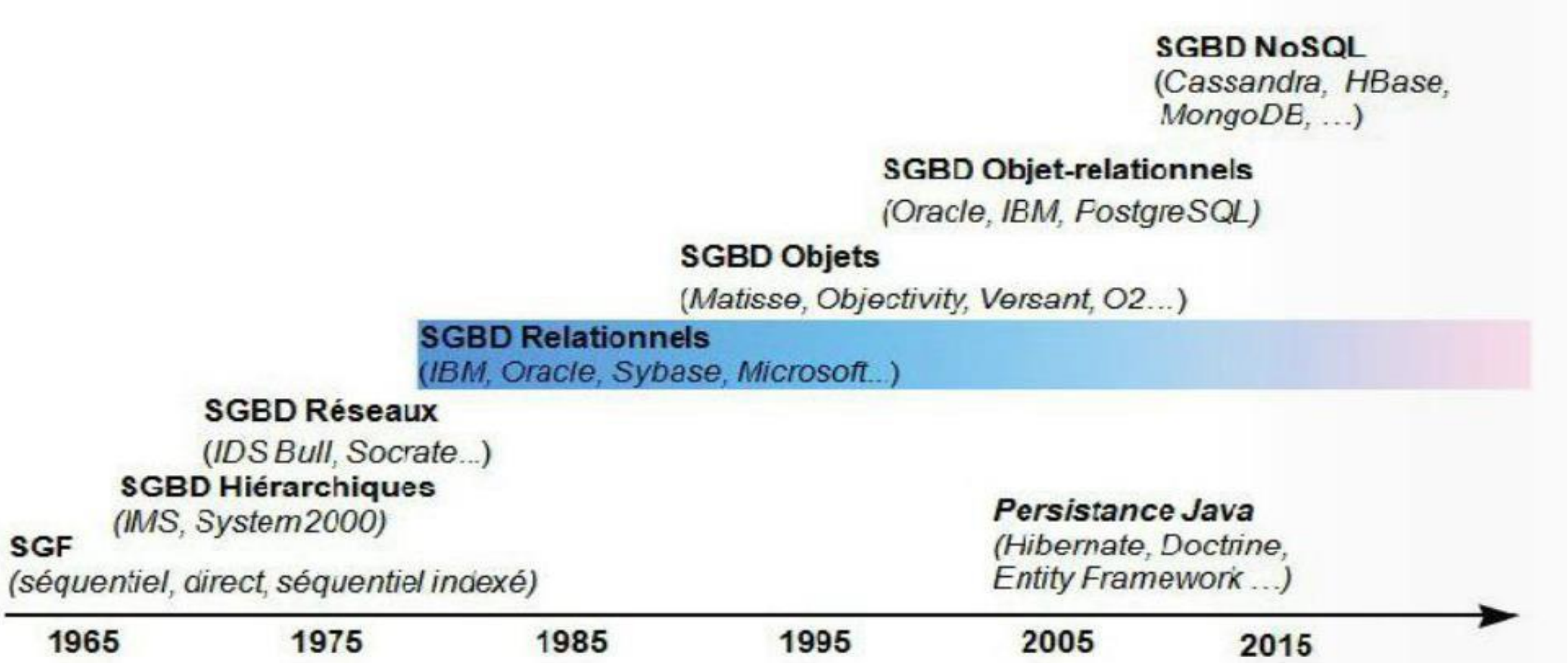


***Figure 0-1**. Historique des bases de données*

Stockage et échange des données du WEB

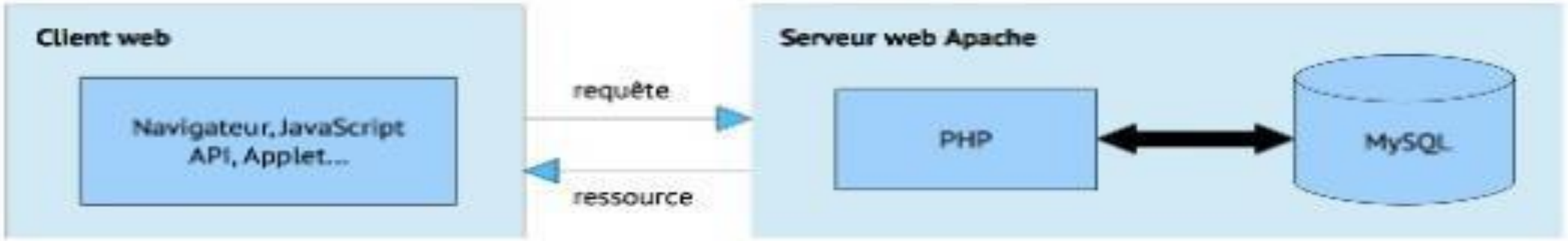


Figure : La stack LAMP.

Stacks habituelles :

- LAMP : Linux, Apache, MySQL, PHP — la plus commune ;
- LEMP : Linux, Nginx, MySQL, PHP — commence à remplacer LAMP ;
- MEAN : MySQL, Express, AngularJS, Node.js — pour du JavaScript côté serveur.

Nouveaux Besoins:
1. Échange de données

- **Exemple 1** : Exportation de données dans Excel.

  - Ce type d'export de données est se fait manuellement utilisant des outils.

⇒**MAIS** comment le programmeur peut réaliser ça dans ces programmes automatiquement ?

Nouveaux Besoins:

2. Description de données spécifiques (mathématique, chimie, images, ……)

- Multimédia,
- Mathématique,
- Imagerie,
- WEB,
- ….

Nouveaux Besoins:

3. Sauvegarde standards de données de petite taille (pour la rapidité de la lecture)

- Les fichiers de configuration des applications ,

Solutions

Utiliser les modèles de données des SGBD pose des problèmes :

- La structure est trop rigide
- les données peuvent ne pas être conformes au schéma
  (==> valeurs nulles, difficultés de traitement et ambiguïtés)
- l'évolution fréquente de la structure de données conduit à des évolutions de schéma pas toujours maîtrisées.
- les données du Web sont indexées par des moteurs de recherche dont les services sont limités, l'interrogation souvent imprécise.

- La majorité de données échangées ne sont pas **structurées**

Nécessité d'un modèle général et souple, «sans schéma», avec un langage de requêtes associé : modèles semi-structurés

Données strucutrées, semi-structures, non-structurées

1. Données structurées

- Les données structurées sont des informations qui ont été formatées et transformées en un modèle de données bien défini= > Les tables de bases de données constituées de lignes et des colonnes,
- Il constitue 20% du total des données des entreprises.

2. Données non structurées

- Les données non structurées peuvent être tout ce qui n'est pas dans un format spécifique.
- Exemple: Word, PDF, texte, logs (fichier journal).
- Un exemple de données non structurées pourrait être les fichiers journaux qui ne sont pas faciles à séparer. Commentaires et publications sur les réseaux sociaux qui doivent être analysés.

Voici un exemple de données non structurées à partir d'un fichier journal.

```
38,P-R-38636-6-45,P-R-39105-1-11,P-R-38036-1-5,P-R-35697-1-13,P-R-35087-1-2
Wed Sep 23 2020 05:21:01 GMT+0500
```

2. Données non structurées : l'analyse nécessite des techniques de **l'analytique**

- Un exemple d'analyse de données non structuré est la détection de modèles dans les e-mails frauduleux.

2. Données semi-structurées

Les données semi-structurées sont des informations qui ne résident pas dans une base de données relationnelles, mais qui possèdent des propriétés organisationnelles facilitant leur analyse. Avec certains processus, vous pouvez les stocker dans la base de données relationnelles

```xml
<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<document>
  <employee>
    <name>Alex</name>
    <age>22</age>
  </employee>
  <employee>
    <name>Bob</name>
    <age>24</age>
  </employee>
  <employee>
    <name>Emily</name>
    <age>32</age>
  </employee>
</document>
```

Exemple : XML, JSON, Avro, Parquet, ORC,...

Différence entre : données structurées /Données semi-structurées

- Les données structurées nécessitent un schéma fixe qui est défini avant que les données puissent être chargées et interrogées dans un système de base de données relationnelle. Les données semi-structurées ne nécessitent pas de définition préalable d'un schéma et peuvent évoluer constamment, c'est-à-dire que de nouveaux attributs peuvent être ajoutés à tout moment.

- Contrairement aux données structurées, qui représentent les données sous forme de table plate, les données semi-structurées peuvent contenir des hiérarchies d'informations imbriquées à n niveaux.

Table de comparaison

Structurées	Semi-structurées	Non structurées
Il est basé sur les tables de base de données relationnelle	Il est basé sur XML/RDF	Il est basé sur des caractères et des données binaires
Il est dépendant du schéma et moins flexible	Il est plus flexible que les données structurées mais moins que les données non structurées	Très flexible et l'absence de schéma
Il est très difficile de mettre à l'échelle le schéma de base de données	La mise à l'échelle est plus simple que les données structurées	C'est très facile à mettre à l'échelle

Introduction au XML

XML : nouveau standard adopté par le World Wide Web Consortium (W3C) comme complément de HTML permettant un échange aisé de données de sur le web.

- Le but principal de XML n'est pas de décrire un format de texte, mais de **structurer** logiquement un contenu.
- Les balises ont le rle de **classer des données** selon une hiérarchie **définie par l'auteur** du document XML.

Introduction au XML

- Avec XML, la mise en forme textuelle est effectué dans une *feuille de style*, un document separé qui associe des formes de présentations (texte en gras, en italique, centré, etc.) aux balises. Des feuilles différentes permettent des formattages différents du même document.
- Des outils permettent de convertir un document XML en HTML, afin de pouvoir afficher une page web.

XML ?

XML = Extensible Markup Language

C'est un langage permettant de représenter et structurer des informations à l'aide de balises que chacun peut définir et employer comme il le veut.

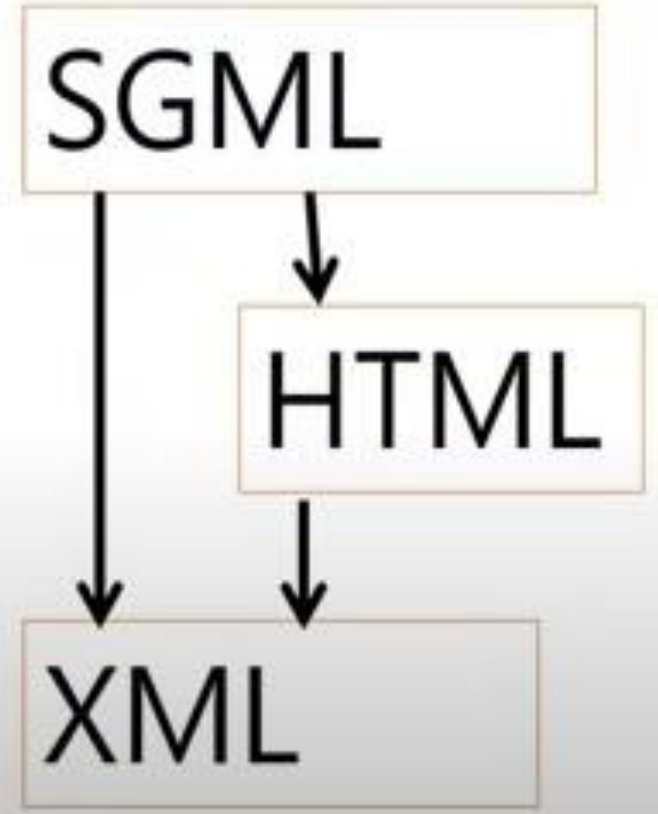
```
texte ... <BALISE> ... texte </BALISE> ...
```

Origine de XML

Standard Generalized Markup Language
(Sépare les données la structure des données et la mise en forme)

Hyper Text Markup Language
(Mélange les données et la mise en forme)

eXtensible Markup Language
(Sépare les données la structure des données et la mise en forme)



Applications de XML

Le format XML est au cœur de nombreux processus actuels :

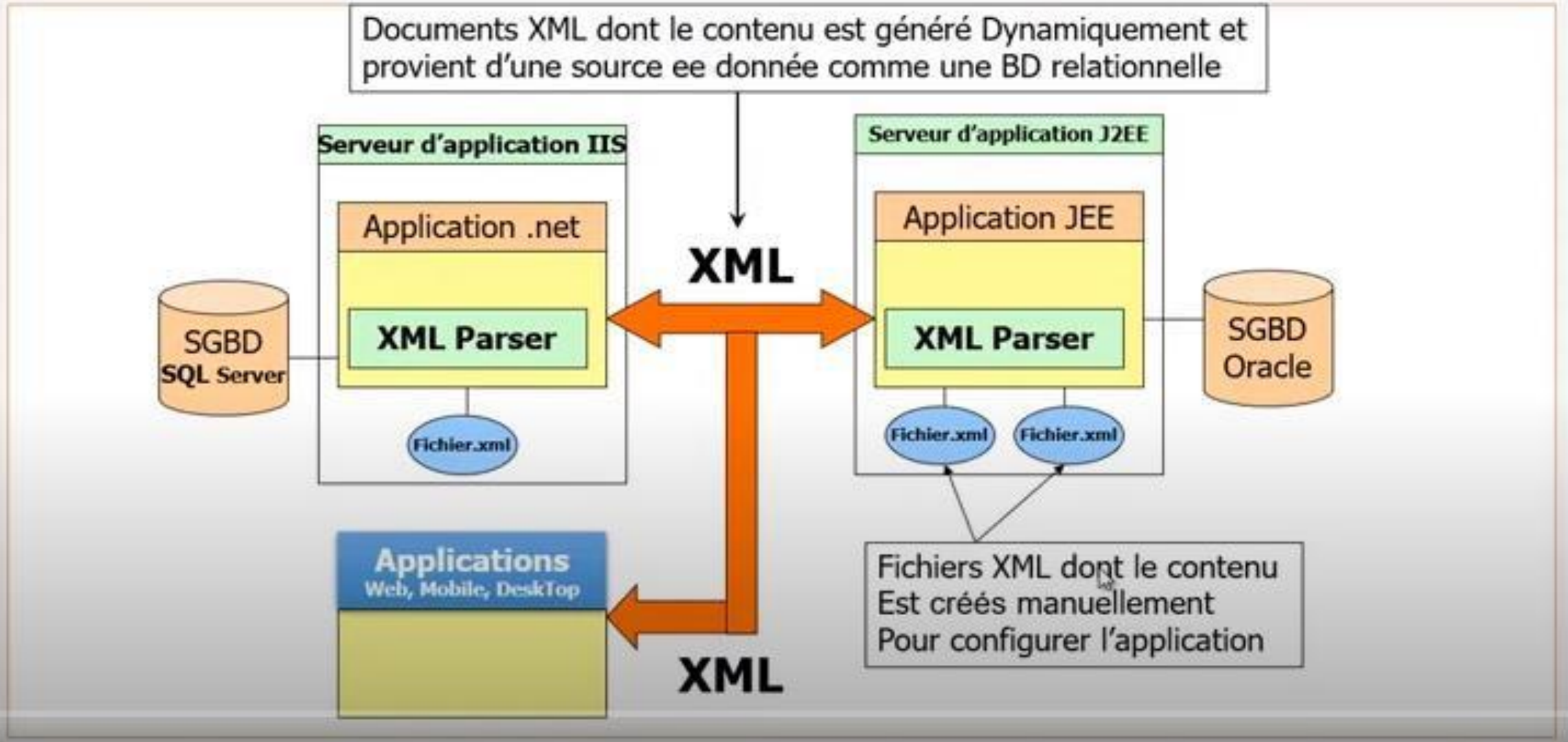
- format d'enregistrement de nombreuses applications,
- échange de données entre serveurs et clients,
- outils et langages de programmation,
- bases de données XML natives.

XML comme format de description de données

Les Langage :

- SMIL (multimédia),
- MathML (mathématique),
- SVG (imagerie, animation,…),
- XHTML (WEB),
- ….

XML pour l'échange de données entre les Systèmes d'Information



Exemple de fichier XML

Un fichier XML représente des informations structurées : 

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- itinéraire fictif -->
<itineraire>
    <etape distance="0km">départ</etape>
    <etape distance="13km">tourner à droite</etape>
    <etape distance="22km">arrivée</etape>
</itineraire>
```

Cet exemple modélise un itinéraire composé d'étapes.

XML permet de choisir la représentation des données sans aucune contrainte. On choisit les balises et les attributs comme on le souhaite.

Structure d'un document XML

Arborescence d'éléments

Un document XML est composé de plusieurs parties :

- Entête de document précisant la version et l'encodage,
- Des règles optionnelles permettant de vérifier si le document est valide
- Un arbre d'*éléments* basé sur un élément appelé *racine*
  - Un *élément* possède un *nom*, des *attributs* et un *contenu*
  - Le contenu d'un élément peut être :
    - rien : élément vide noté `<nom/>` ou `<nom attributs.../>`
    - du texte
    - d'autres éléments (les éléments enfants).
  - Un élément non vide est délimité par une *balise ouvrante* et une *balise fermante*.
    - une balise ouvrante est notée `<nom attributs...>`
    - une balise fermante est notée `</nom>`

Choses interdites

Les règles d'imbrication XML interdisent différentes configurations qui sont plus ou moins tolérées en HTML :

- plusieurs racines dans le document,
- des éléments non terminés (NB: XML est sensible à la casse),
- des éléments qui se chevauchent.

```
<element1>
    <element2>
    </Element2>
    <element3>
    </element1>
</element3>
```

En XML, cela crée des erreurs « *document mal formé* ».

I-B-3. Attribut

Un élément peut être porteur d'un ou plusieurs attributs.

xml Sélectionnez

```xml
<Element att1="test1" att2="test2">...</Element>
```

Ici Element est porteur de deux attributs att1 et att2.

Les noms des attributs suivent les mêmes règles d'écriture que ceux des éléments.

Les attributs ne peuvent contenir que du texte.

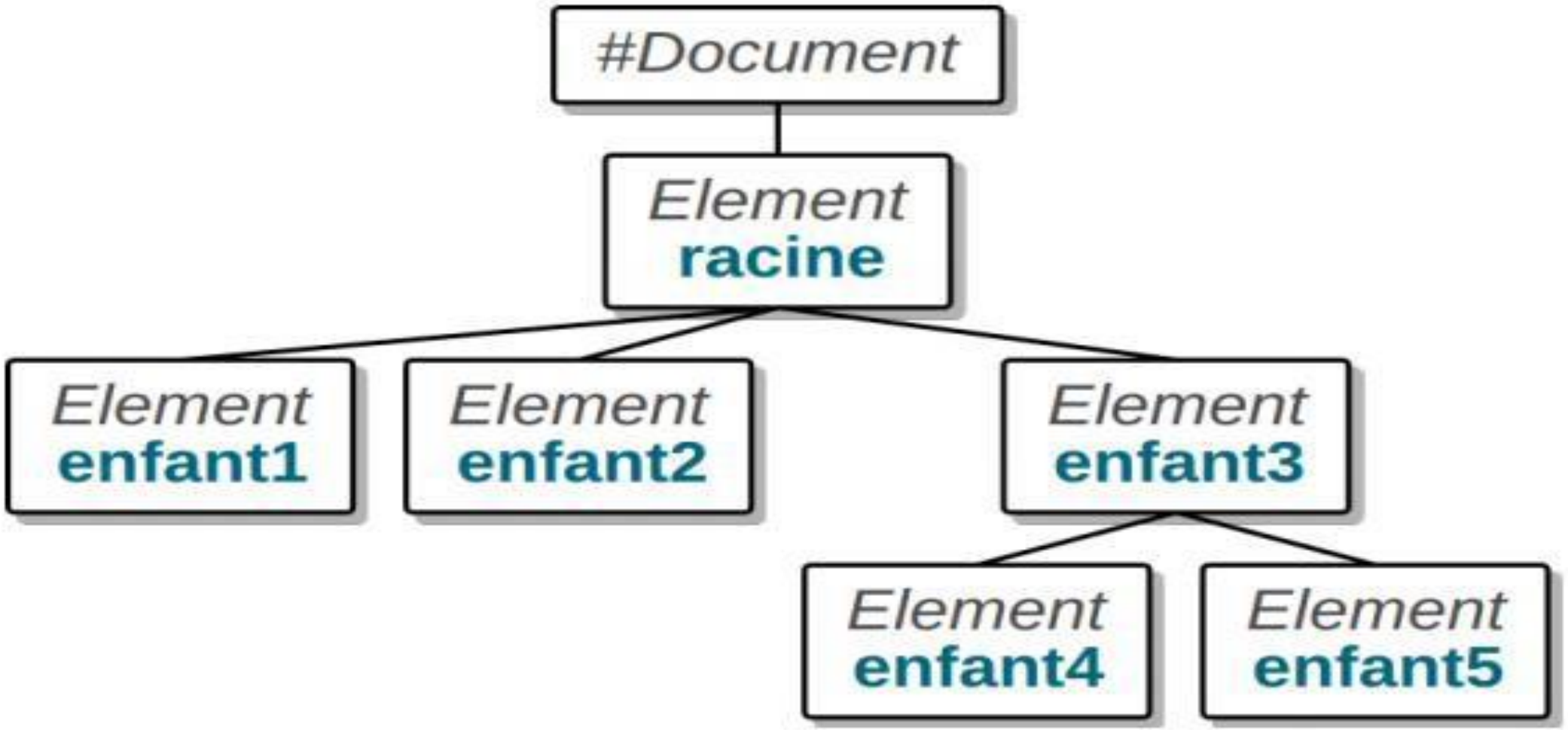
Un élément ne peut posséder deux attributs de même nom, ainsi le code suivant est faux

xml Sélectionnez

```xml
<Element att1="test1" att1="test2">...</Element>
```

Vocabulaire

Soit cet arbre XML :



Vocabulaire (suite)

Voici comment on désigne les différents nœuds les uns par rapport aux autres :

- `<racine>` est le nœud **parent** du nœud **enfant** (*child*) `<enfant3>`, lui-même parent de `<enfant4>` et `<enfant5>`,
- `<racine>`, `<enfant3>` sont des nœuds **ancêtres** (*ancestors*) de `<enfant4>` et `<enfant5>`,
- `<enfant4>` et `<enfant5>` sont des **descendants** (*descendants*) de `<racine>` et `<enfant3>`,
- `<enfant1>` est un nœud **frère** (*sibling*[1]) de `<enfant2>` et réciproquement.

[1] *siblings* = fratrie.