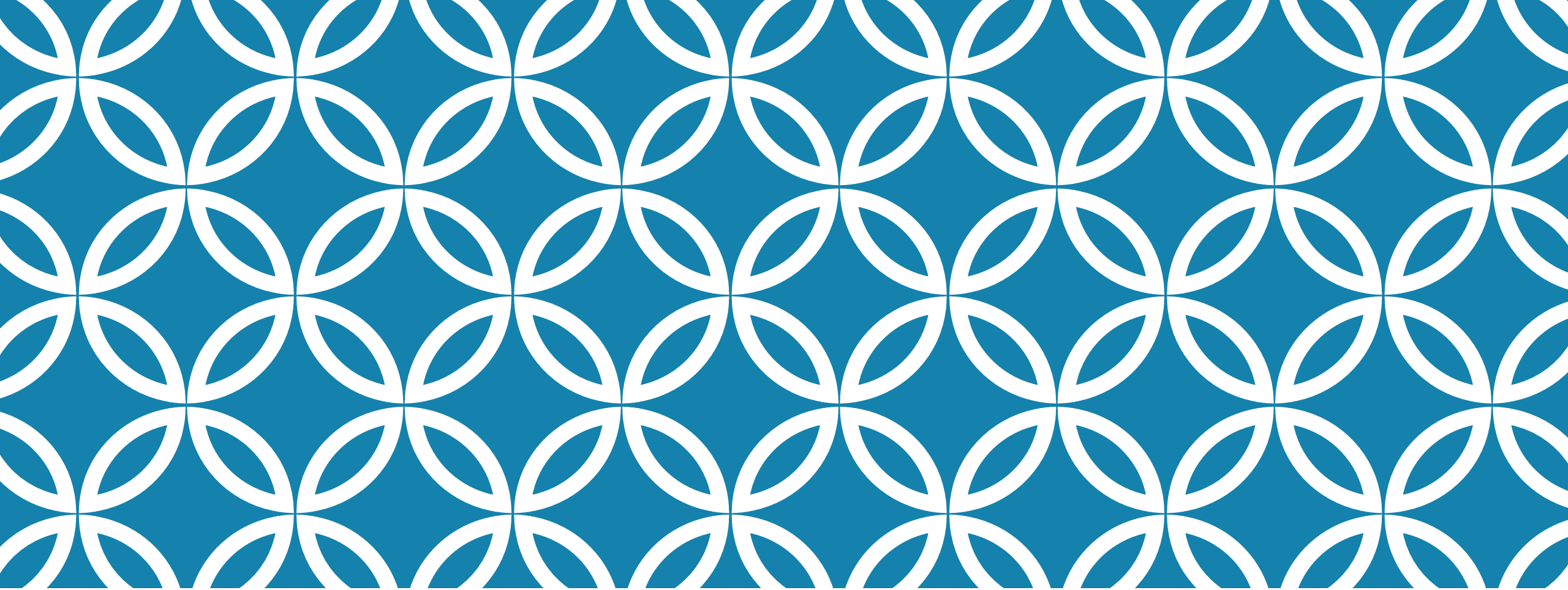


XML

XML

- XML = *eXtensible Markup Language* : langage de balisage extensible.
- XML est un langage pour structurer des contenus et définir une classe d'objets de données, par exemple:
  - ✓un dessin vectoriel (SVG), une page Web (XHTML), un flux (RSS)…
- Le langage est basé sur le concept de balisage des données.
- Un document XML:
  - ✓contient des déclarations, éléments, commentaires, définition de caractères spéciaux et instructions (facultatives) de traitement
  - ✓c'est un arbre: il doit avoir une racine et les éléments doivent s'imbriquer proprement
- Plus simple que SGML (langage de balisage généralisé normalisé, iso), plus complexe mais moins confus et plus performant que HTML
- Recommandation officielle du W3C (The World Wide Web Consortium) depuis le 10 février 1998
- Idéal pour l'échange de données semi-structurées
- Utilisable entre machines

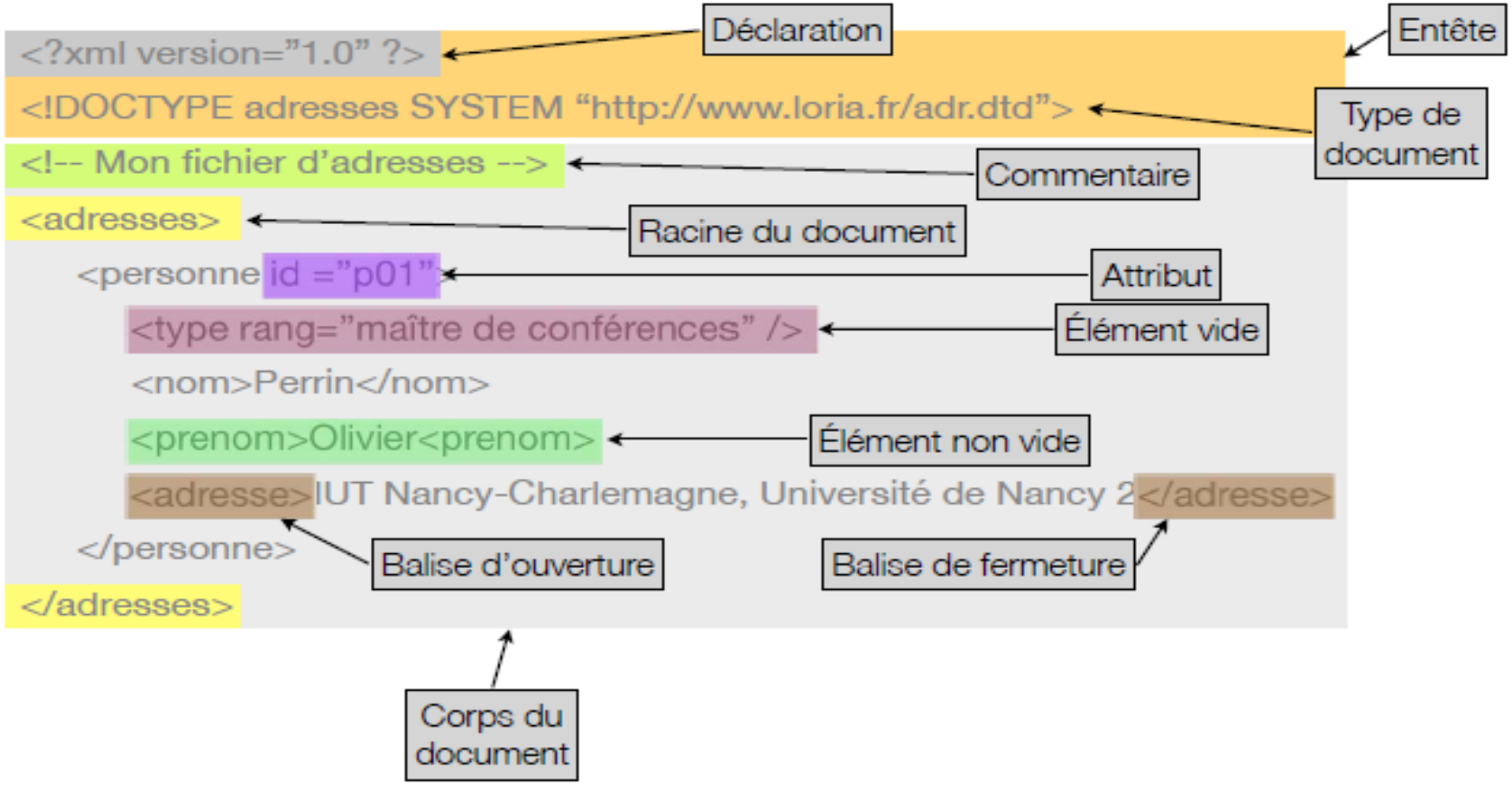
XML (2)

- XML, c'est donc…
  - ✓un méta-langage universel pour structurer les données…
  - ✓qui permet aux utilisateurs de délivrer du contenu…
  - ✓depuis les applications à d'autres applications (browsers par exemple)
- XML promet de standardiser la manière dont l'information est :
  - ✓échangée (XML)
  - ✓personnalisée/présentée (XSL/CSS)
  - ✓recherchée (XPath/XSLT/XQuery)
  - ✓sécurisée (Encryption, Signature)
  - ✓liée (XLink)

XML (3)

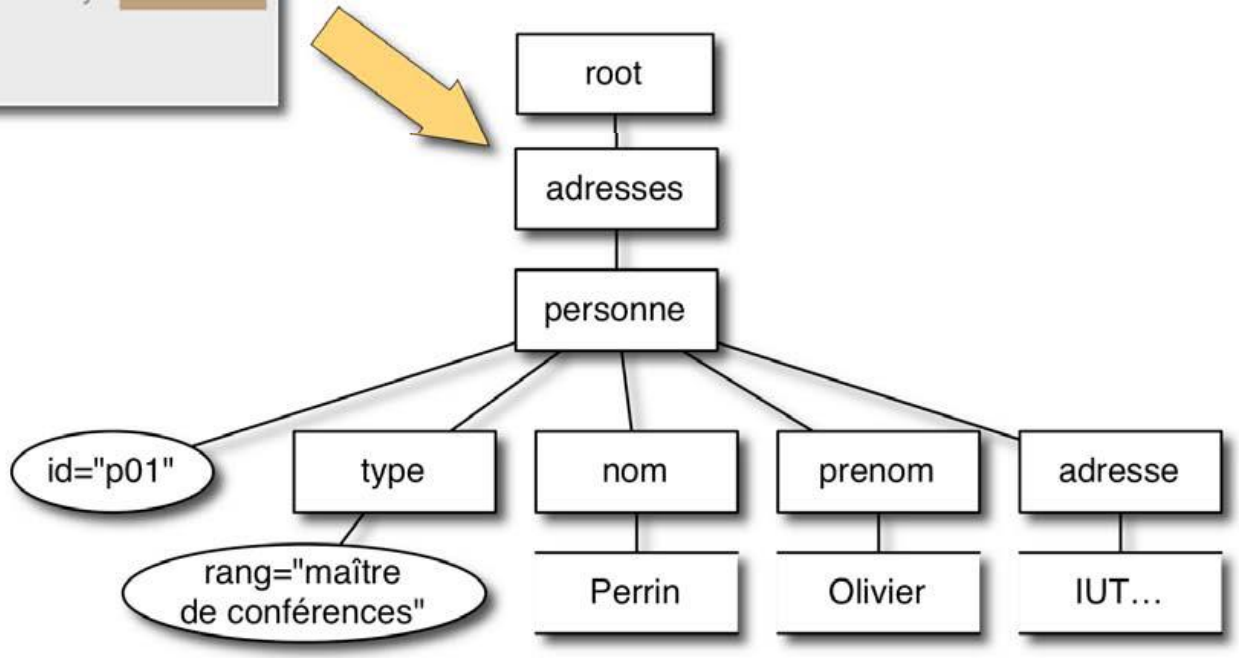
- Balise (ou tag ou label)
  - ✓marque de début et fin permettant de repérer un élément de données (textuel)
  - ✓forme: <balise> de début, </balise> de fin
  - ✓les balises indiquent la signification des sections marquées
- Elément de données
  - ✓texte encadré par une balise de début et une de fin
  - ✓les éléments de données peuvent être imbriquées
- Attribut
  - ✓couple nom="valeur" qualifiant une balise
  - ✓<producteur no="160017" region="Bourgogne"/>
- Les utilisateurs définissent leurs propres balises
- Il est possible d'imposer une grammaire spécifique (DTD, Schéma)

XML: UN EXEMPLE



XML: FORMAT INTERNE

```
<?xml version="1.0" ?>
<!DOCTYPE adresses SYSTEM "http://www.loria.fr/adr.dtd">
<!-- Mon fichier d'adresses -->
<adresses>
    <personne id ="p01">
        <type rang="maître de conférences" />
        <nom>Perrin</nom>
        <prenom>Olivier<prenom>
        <adresse>IUT Nancy-Charlemagne, Université de Nancy 2</adresse>
    </personne>
</adresses>
```



POURQUOI XML ?

- Définir vos propres langages d'échange
  - ✓commande, facture, bordereau de livraison, etc.
- Modéliser des données et des messages
  - ✓*Document Type Definition* (DTD)
  - ✓types et éléments agrégés (*XML Schema Definition*)
  - ✓passerelle avec *Unified Modelling Language* (UML)
- Publier des informations
  - ✓indépendante du format
  - ✓mise en forme avec CSS et XSL
  - ✓présentation possible en XHTML, PDF, WML,…
- Archiver des données
  - ✓auto-description des archives

COMPARAISON DONNÉES/DOCUMENTS

- Approche «Donnée»
  - structuration forte et simple
  - compatibilité SGBDR existants
  - mise à jour en place
  - intégrité sémantique
  - indexation exacte
  - adapté au transactionnel et décisionnel
  - performances «moyenne» à «fort» pour une volumétrie «moyenne »
- Approche «Document»
  - structuration faible et complexe
  - systèmes documentaires spécialisés
  - gestion de versions
  - recherche textuelle
  - indexation approchée
  - accès type moteur de recherche
  - performances «moyenne» pour une volumétrie «forte»

AVANTAGES DE XML

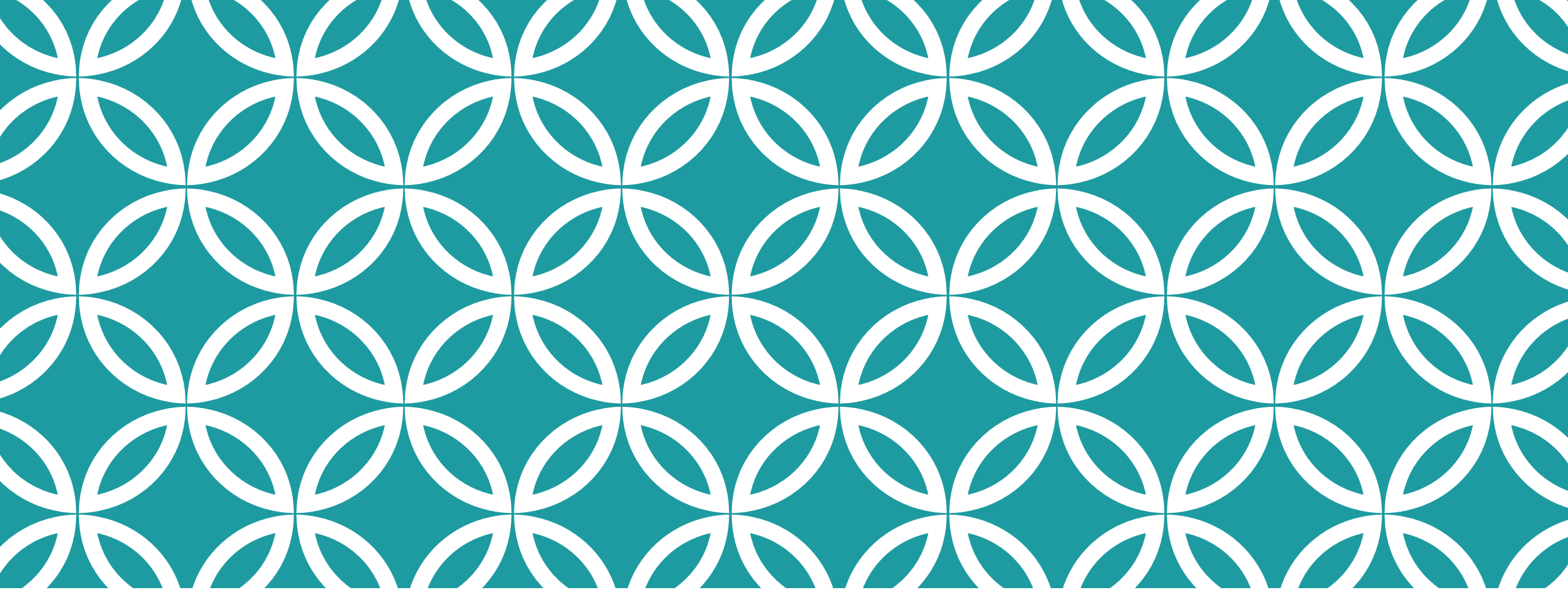
- Une technologie structurante
- Clarifie toutes les interfaces d'échange
- Transversale à l'entreprise
  - ✓ échanges de données
  - ✓ Bureautique
  - ✓ GED
  - ✓ sites Web
  - ✓ EDI
  - ✓ bases de données
  - ✓ intégration e-business
  - ✓ …
- Un choix stratégique de direction
  - ✓ ne pas rester isolé

FAIBLESSES DE XML

- Une syntaxe verbeuse
- Un méta-langage, mais de nombreux langages
- Coûteux en CPU
  - ✓Analyse
- Coûteux en mémoire
  - ✓instanciation

XML ET BD

- Intégration des données et méta-données
- Standard d'échange de données universel
- Les BD ne peuvent rester indifférentes :
  - ✓ nécessité de stocker les documents XML
  - ✓ nécessité de pouvoir interroger ces documents
  - ✓ évolution ou révolution ?
- Quel modèle de données ?
- Quel langage d'interrogation ?
- Quelle intégration avec l'existant ?



MODÈLES DE DONNÉES

DOCUMENT XML

- Un document XML peut être associé à:
  - ✓une DTD ou un schéma pour décrire la structure du document XML
  - ✓une feuille de style pour présenter les données
  - ✓DTD ou/et schéma permettent de définir son propre langage basé sur XML
  - ✓vocabulaire (balises)
  - ✓grammaire (imbrications)
- Deux types de documents
  - ✓*well-formed document*
  - ✓*valid document*

DOCUMENT BIEN FORMÉ (*WELL-FORMED*)

- Commence par une déclaration XML (attribut version obligatoire) avec possibilité de choisir un encodage (le défaut est utf-8):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- Structure hiérarchique:
  - ✓ les balises d'ouverture et de fermeture doivent apparaître et correspondre
  - ✓ pas de croisements de type ....
  - ✓ sensible à la casse: "LI" n'est pas égal à "li" par exemple
  - ✓ balises "EMPTY" utilisent la syntaxe XML "auto-fermante":
  - ✓ les valeurs d'attributs sont quotés: <a href="http://www.foo.fr/xml.html">
  - ✓ un seul élément racine (root):
    - l'élément root ne peut apparaître qu'une fois et ne doit pas apparaître dans un autre élément
- Caractères spéciaux (!!) : <, &, >,", '
  - ✓ utilisez < & > &aquot; ' à la place dans un texte !
  - ✓ les espaces sont préservés

DOCUMENT VALIDE (*VALID*)

- Un document “valide” doit être:
  - ✓“well-formed” (formé correctement)
  - ✓être associé à une DTD (ou une autre grammaire)
  - ✓et être conforme à cette DTD ou une grammaire