

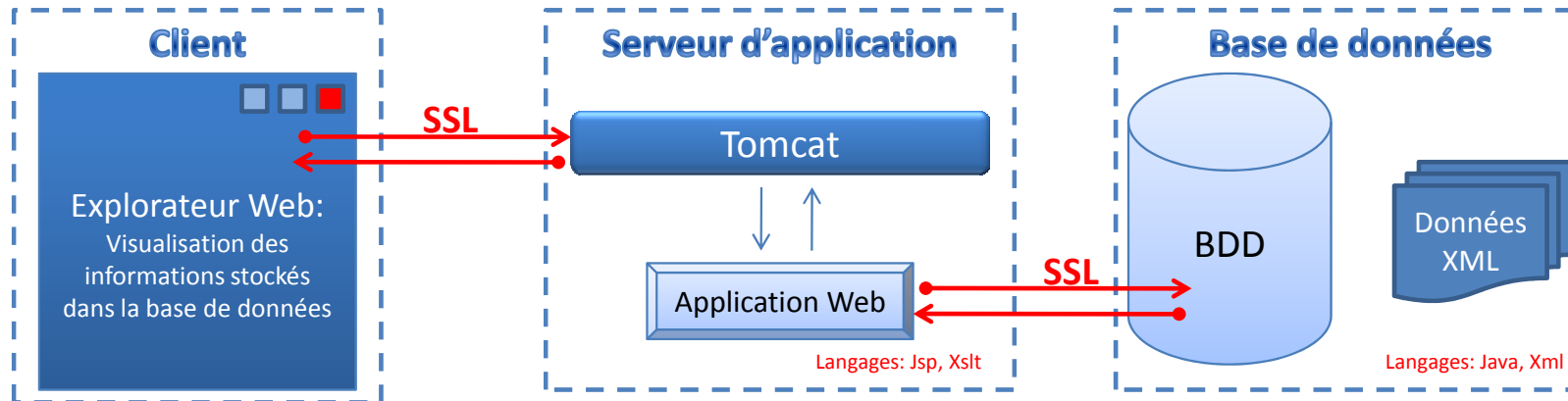
# Gestionnaire de données confidentielles

- **Introduction:** Perdre des données est quelque chose qui peut s'avérer critique dans certaines situations. La sauvegarde sur serveur est un réel avantage pour ce type de problématique car moins sensible aux pertes et par conséquent limite considérablement ce risque. Néanmoins, le stockage à distance donne lieu à un nouveau problème lorsque ces mêmes données sont confidentielles. Nous allons donc à travers ce projet tenter de répondre à ces différentes problématiques. L'application que nous allons développer nous placera pour cela dans le cas d'une société d'audit souhaitant conservé des données confidentielles (tels que les salaires, et autres observations) relatives aux employés. Celle-ci fera donc appel à notre application web car nécessitant uniquement l'utilisation d'un explorateur internet. Les données seront donc accessibles pour les chargés d'audit à partir de n'importe quel poste situés dans les différentes usines de production de l'entreprise audité. De plus l'application permettra une visualisation directe et simple des données grâce à une interface web.
- **Avantage de l'architecture utilisée :**
  - Dans le cadre de notre application, les données se doivent d'être persistantes. Il nous faut donc faire une sauvegarde de celles-ci, nous utiliseront donc une base de données pour les stocker. Nous utiliseront un modèle original de base de données appelé : « base de données native XML ». En effet l'application étant web, elle permettra un affichage des données dans l'explorateur web directement. Il est donc judicieux ici d'utiliser ce modèle de base de données couplés avec du XSLT car les données seront ainsi directement interprété par l'explorateur web.
  - La partie sécurisation consiste essentiellement à sécuriser le transfert d'informations entre deux programmes: l'application client (web) et la base de donnée autrement dit il s'agit des échanges au niveau des connexions client/serveur. Cette sécurisation sera assurée par l'utilisation du protocole SSL (secure sockets layer). En effet, grâce à ce dernier on pourra garantir l'identité de l'émetteur (le client) et le récepteur (la base de données), ainsi que la confidentialité et l'intégrité des informations échangées. Voici les étapes principales qui ont lieu lors de ce processus confidentiel:
    - Identification de l'émetteur et le récepteur.
    - Confidentialité des informations échangées : les données envoyées sont cryptées afin de les rendre inintelligibles pour ceux qui ne sont pas autorisés à "regarder" le transfert. Ainsi, on évite les problèmes de type MITM (Man In The Middle). (La procédure de cryptage pourra se reposer sur l'utilisation des clés symétriques. (ex: clé de Jules César))
    - Intégrité des informations échangées.Cette supervision d'échange de données confidentielles offre donc un bon compromis entre la base de données et le client (application web) est de sécuriser l'accès aux données XML.
- **Livrable:**
  - Au niveau du livrable, il s'agit donc d'une application web de gestion de données confidentielles.
- **Techniques utilisés:**
  - Connexion sécurisée SSL

- Sémaphores (problème lecteurs-écrivain)
- Threads (utilisation optimum des ressources)
- Arbres et Arborescence (accès au données : table de correspondance et binary tree)
- XPath
- **Langages:**
  - Java
  - XML
  - XSLT
  - JSP
- **Liens utiles:**
  - [http://en.wikipedia.org/wiki/XML\\_database](http://en.wikipedia.org/wiki/XML_database)
  - <http://www.xml.com/pub/a/2001/10/31/nativexmlldb.html>
  - <http://www.tarunworld.com/databases/tutorial-xml-databases.html>
  - <http://www.vldb.org/conf/2001/P341.pdf>
- **Schéma de l'architecture :**

# Réalisation d'un gestionnaire de données confidentielles

Consultable partout dans le monde, sécurisée par un protocole SSL pour protéger les données sensibles, utilisant une BDD native XML pour plus de simplicité, notre application Web vous permettra d'avoir en permanence accès à vos informations et ceci de manière sécurisée.



Objectifs:  
Création de l'application Web  
Sécurisation du protocole  
Conception de la BDD

Livrable:  
Application Web  
accédant à la BDD et  
transférant les données  
de manière sécurisé

