

# Exercice Scientifique et technique

## Plateforme Salesforce

**COHEN Hugo**

# Sommaire

---

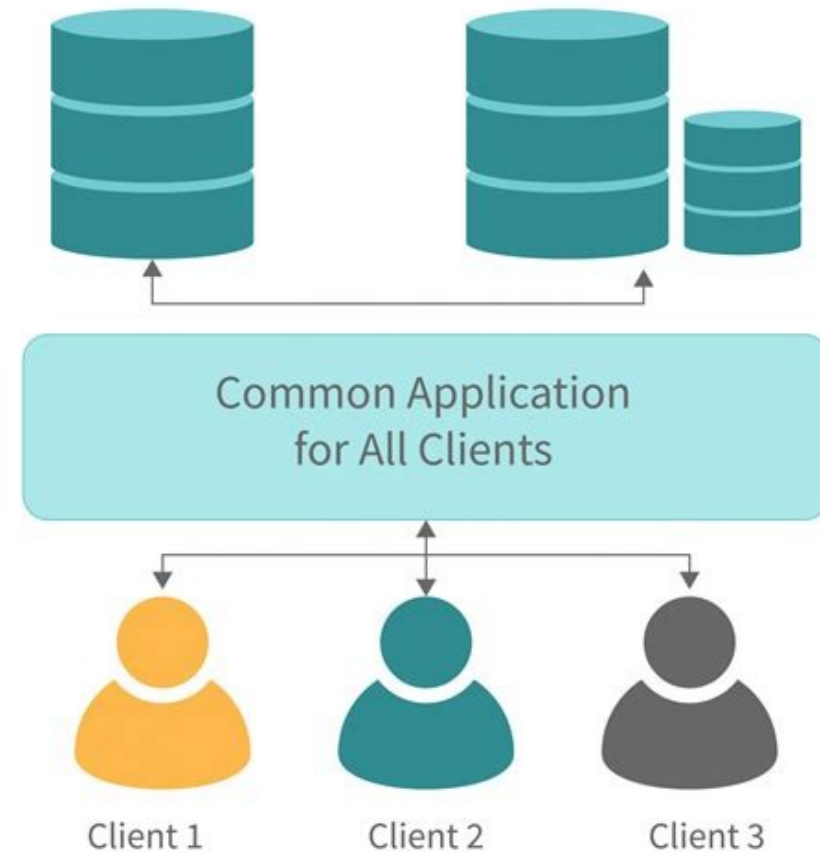
1. Présentation de Salesforce
2. Présentation de APEX
3. Limites en comparaison à Java 25
4. Langage de query SOQL
5. Trigger et intégration BDD
6. Flows et les limites du No-Code
7. Conclusion

# Présentation de Salesforce

CRM (Customer Relationship Management)  
Hybride SaaS / PaaS  
Architecture Multi-tenant

Chiffres clés :

- **20%** de parts de marché
- Plus de **150 000** entreprises clientes

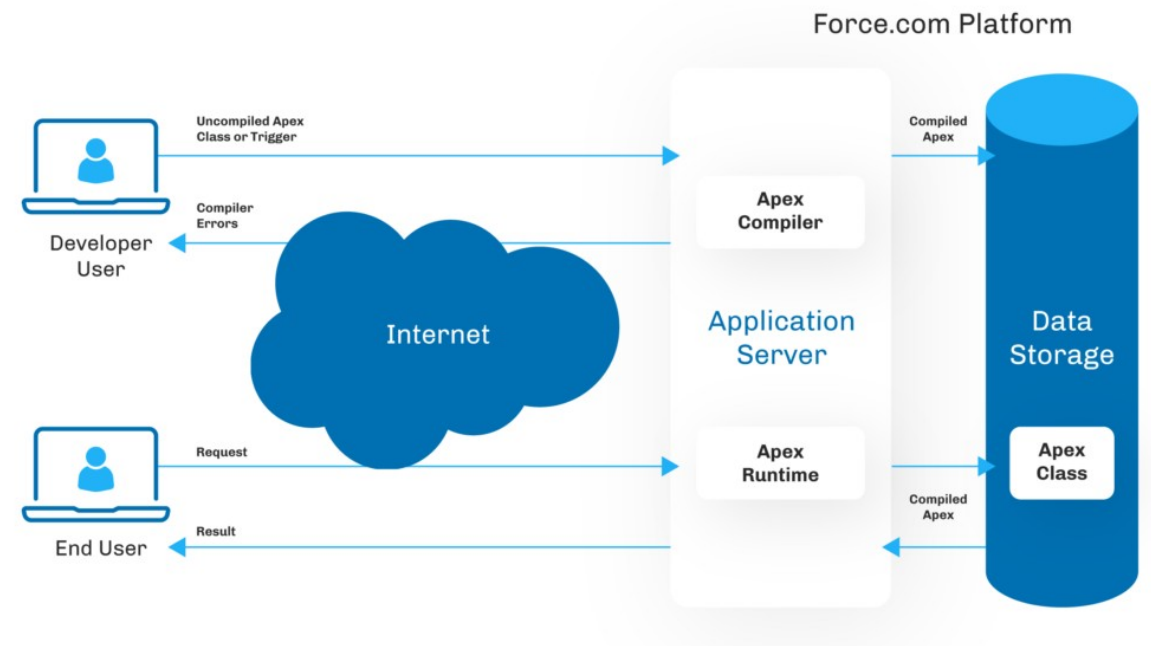


# Présentation de APEX

Langage propriétaire, fortement typé et orienté objet

Aucune exécution **locale** (côté client)

**Intégration native** (DML et requêtes intégrées)



# Limites en comparaison à Java 25

---

- Syntaxe et philosophie basée sur Java 8 (*pas de Lambdas ni de Streams, pas d'inférence de type, pas de records ...*)
- Governor Limits
- Mécanisme de sauvegarde strict du Multi-tenancy
- Pas de multi-threading
- Heap-Size plafonnée

# Langage de *query* SOQL

---

**SOQL** (*Salesforce Object Query Language*) : Langage de requêtage conçu pour préserver les performances de la base de données partagée.

Navigation par l'arbre d'objets :

- Enfant vers Parent (Many-to-One)
- Parent vers Enfants (One-to-Many)

Limites du langage :

- Absence de la clause *JOIN*
- Interdiction stricte du *SELECT \**

# Trigger et intégration BDD

---

Exécution automatique déclenchée par des opérations **DML** (*Data Manipulation Language : Insert, Update, Delete, Undelete*).

Deux contextes temporels :

- **BEFORE** (*validation/modification avant sauvegarde*)
- **AFTER** (*logique asynchrone ou action sur d'autres objets*).

Nécessité de « *bulkifié* »

# ***Flows et les limites du No-Code***

---

Paradigme *No-Code* : Interface visuelle générant du code ou des requêtes sous-jacentes.

## **Coût et limites techniques :**

- Opacité de l'ordre d'exécution
- Consomme beaucoup de requêtes SOQL
- Gestion des exceptions et rollbacks rudimentaire

Stocké sous forme de *XML* donc **versionnage** complexe.



# Bibliographie

---

- <https://www.salesforce.com/news/stories/idc-crm-market-share-ranking-2025/>
- <https://en.wikipedia.org/wiki/Salesforce>
- [https://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Platform_as_a_service)
- [https://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Software_as_a_service)
- <https://www.salesforceben.com/salesforce-flow-builder-vs-apex/>
- [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/langCon\\_apex\\_dml.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/langCon_apex_dml.htm)
- [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_gov\\_limits.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm)
- <https://medium.com/@justusvandenbergsalesforce-apex-optimization-large-strings-vs-heap-size-and-cpu-time-66ee6621ec26>