



Architecture Logicielle : Tinder

Berdah Clément & Hamsek Fayçal

Sommaire

01

Tinder

Présentation

02

Etat de l'art

Aspect Commercial
Aspect Technique

03

Architecture
logicielle

Client - serveur
Microservices

04

Architecture
Tinder

05

Conclusion

Points clés



01

Tinder



Tinder



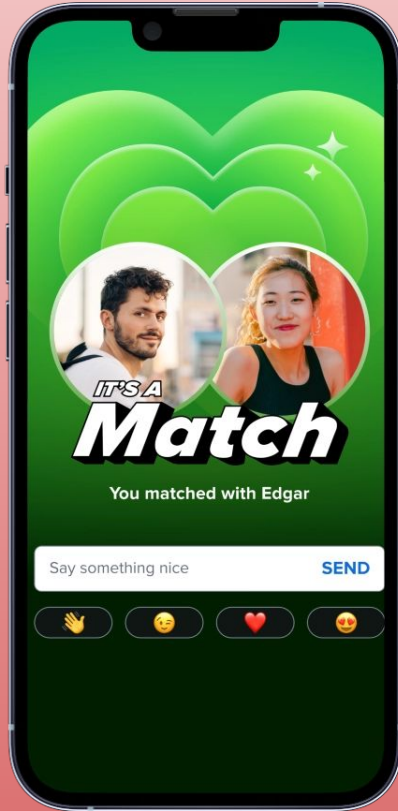
Tinder

But

Dating app

Communication

Tchat



Système

iOS / Android



Principe

Swipes / match

Comparaison chiffres 2022



13%

■ Bumble

30%

■ Tinder

12%

■ Badoo

Source



02

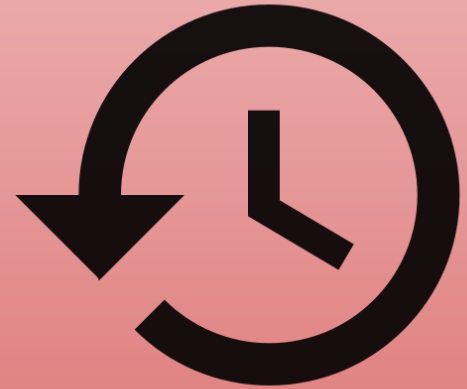
Etat de l'art



Contexte historique

Applications de rencontres - 1995.
Match.com

Grande évolution du numérique
→ marché des réseaux sociaux.



Tinder : innovation.

Lancé en 2012 - Période de croissance.

Interactions sociales numériques.
→ Besoin de simplicité, efficacité.

Invention du swipe, des matches.

Solutions intuitives.



Tinder : Transparence.

Polémique :

- Cyberharcèlement.
- Mensonges
- Relations platoniques



Mobile

IOS

- Objective - C
- Swift

Android

- Java
- Kotlin



Site Web

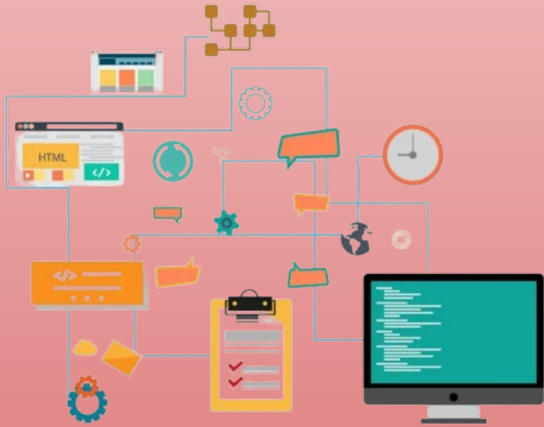
FRONT-END

- HTML5
- Javascript
- jQuery
- React

BACK-END

- Node.js
- Java
- SQS
- Amazon Kinesis



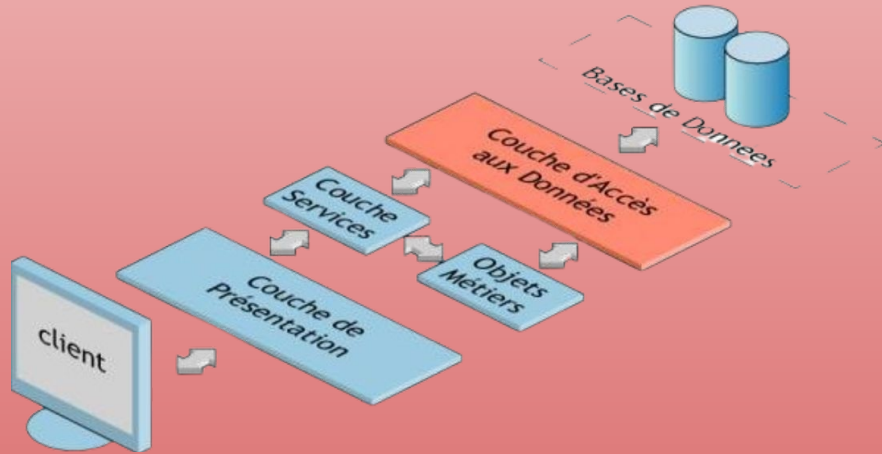


03

Architecture Logicielle



Architecture Logicielle



Schématiser

- Conception
- Intéractions
- Relations

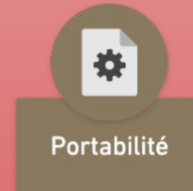
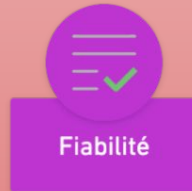
Anticiper

- Bug
- Redondance
- Dégradation logiciel

Architecture Logicielle

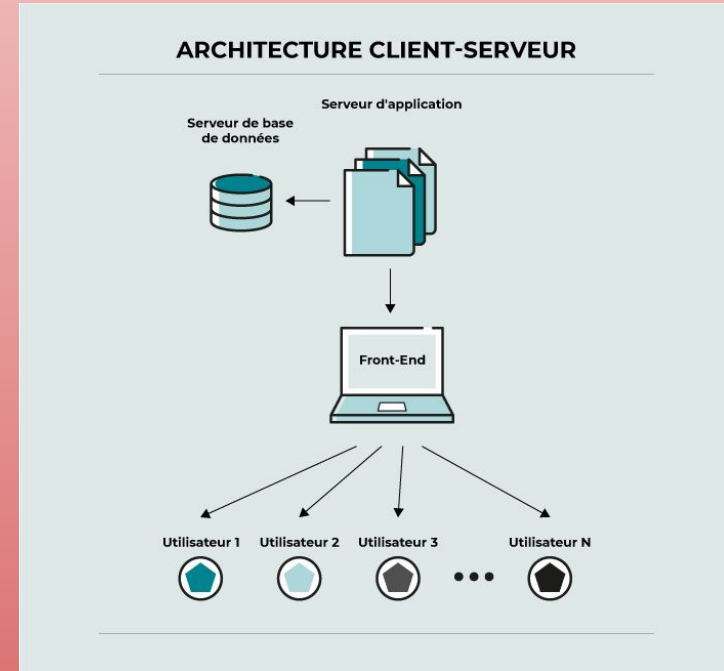
Norme ISO 25010

Critères de qualité logicielle

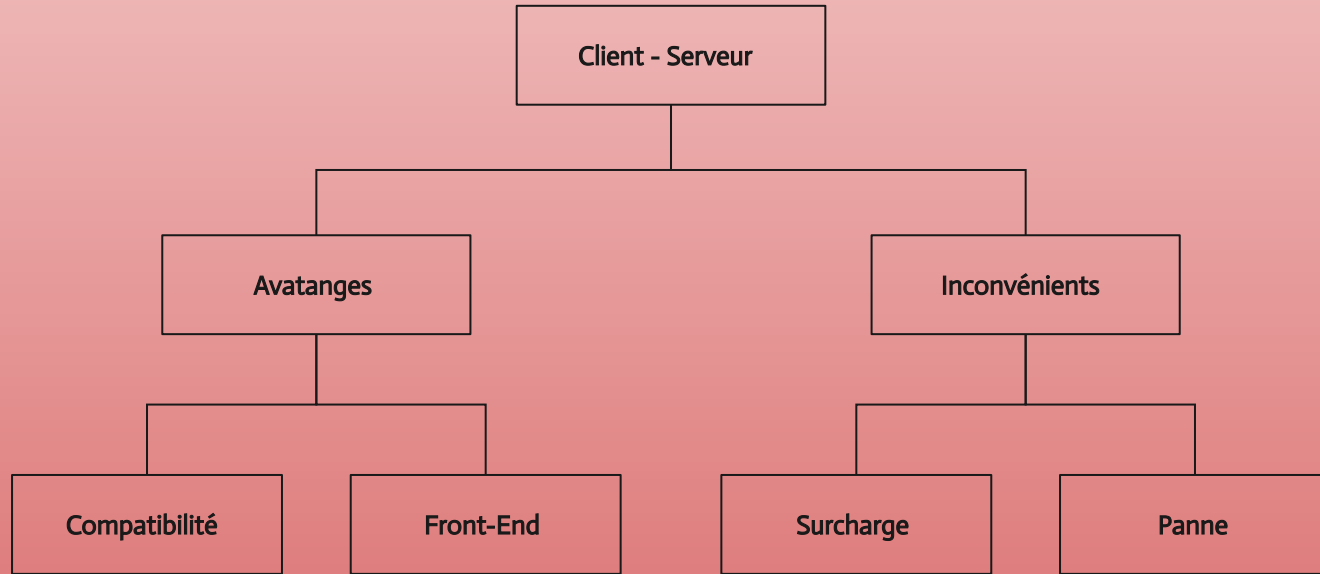


Architecture client-serveur

- 3 parties
 - Front-end
 - Serveur d'application
 - Serveur BDD
- Communication
 - Serveur d'application
- Allégorie : Restaurant

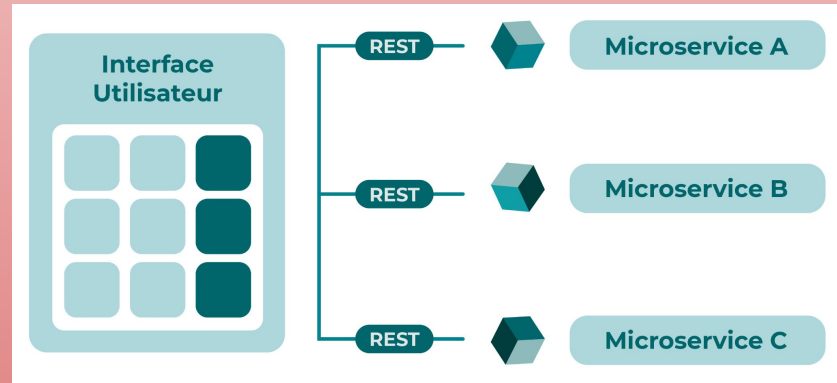


Avantages / Inconvénients

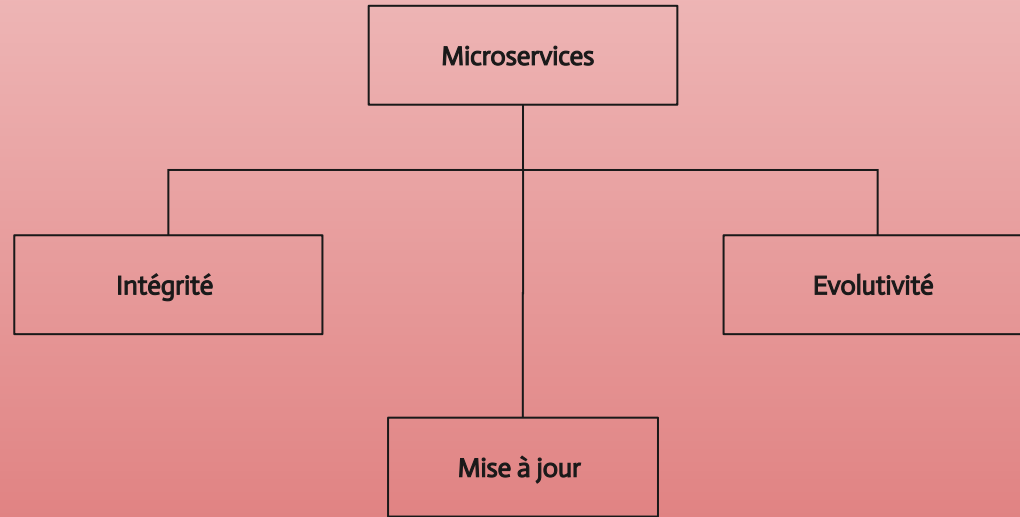


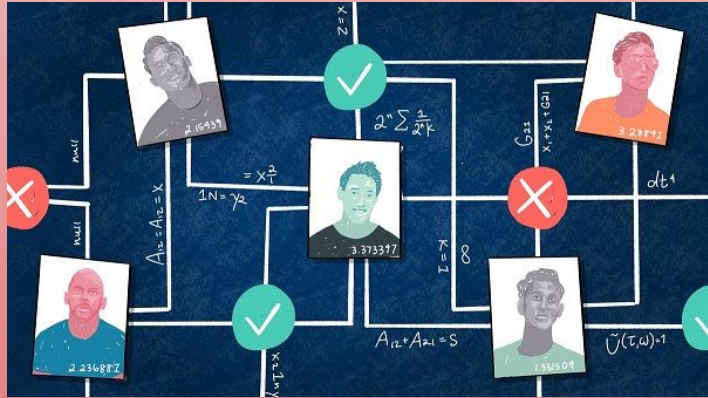
Architecture Microservices

- Challenge techniques
 - Ressources
- Nouveaux services
 - Streaming
 - E-commerce
- Performance
- Coup financier



Problèmes résolus





04

Architecture Tinder



Architecture client-serveur

Client : smartphones | ordinateur

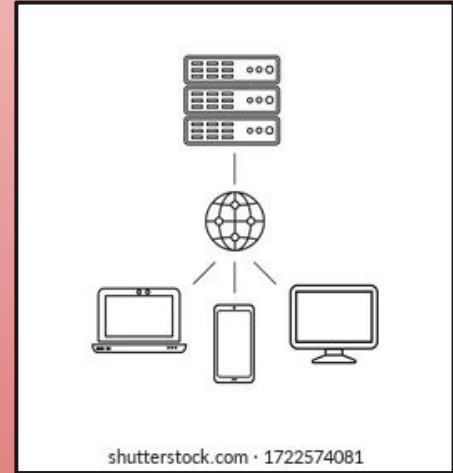
Mobile :

- Interface
- Interactions locales

Envoie de requêtes

Serveur : Gérer les requêtes - BDD

- Swipe
- Notifications



Architecture microservices

Recommandations

Messagerie

Gestion des utilisateurs

Usage de microservices permettant :

- Une optimisation
- Une meilleure scalabilité
- Flexibilité
- Résilience





05

Conclusion

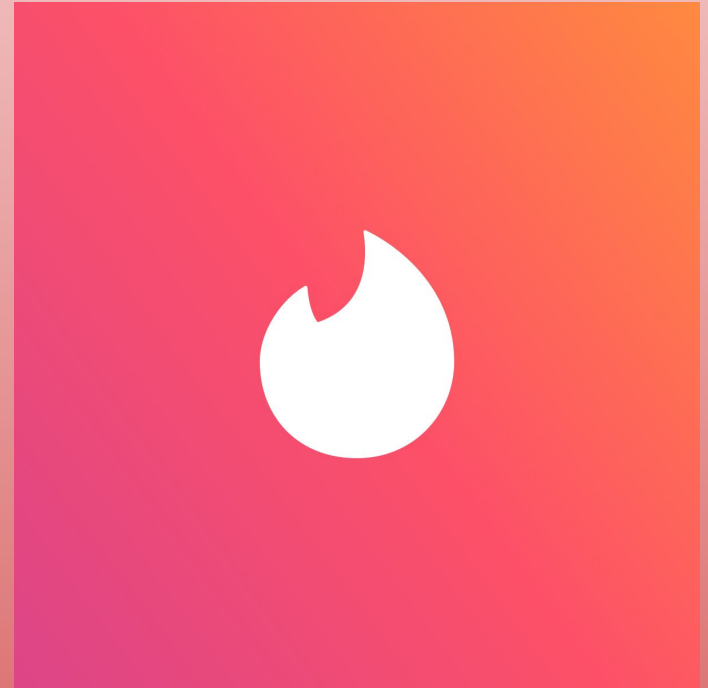


Tinder : Une application optimisée

Numéro 1 sur le marché.

Un service optimisé, une bonne expérience utilisateur.

Architecture : Client-serveur
Microservices



Merci !

Questions ?

