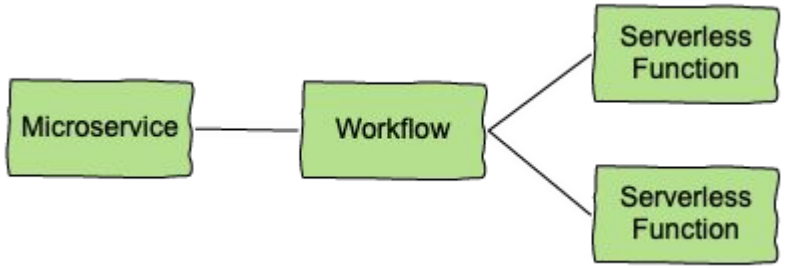
Netflix Cosmos

Guillaume VACHERIAS - Eric HERQUÉ

Introduction

Plateforme qui regroupe des microservices composés de :

- Workflow
- Function serverless



Plusieurs générations :

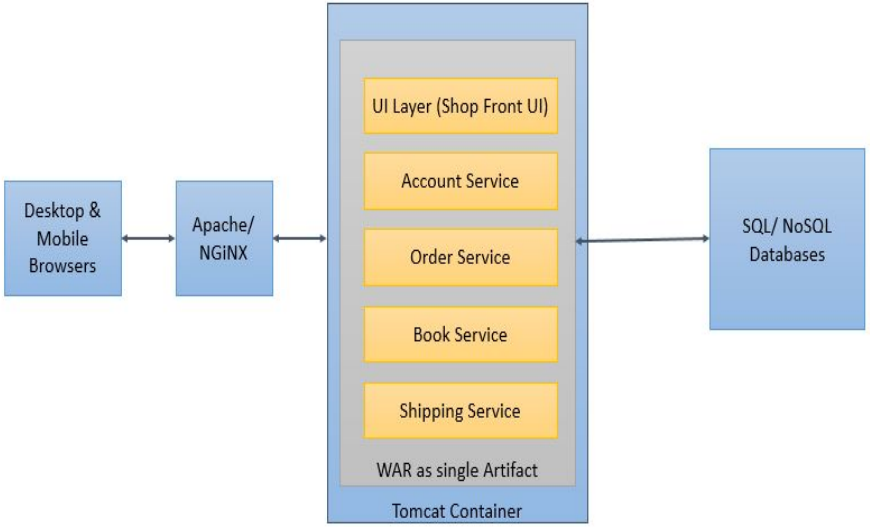
1. Architecture Monolithique qui permet les fonctions de streaming (2007)
2. Amélioration de la 1er génération
3. Reloaded (2017)

Architecture Monolithique

- Netflix Reloaded est basé sur un style d'architecture Monolithic

- Rapide à développer
- Simple à déployer (Java WAR File Format)
- Simple à mettre à grande échelle (scale up)

- Code source très lourd
- Temps de démarrage du container très long
- Problème de scaling developpment
- Difficile à changer d'outil de développement

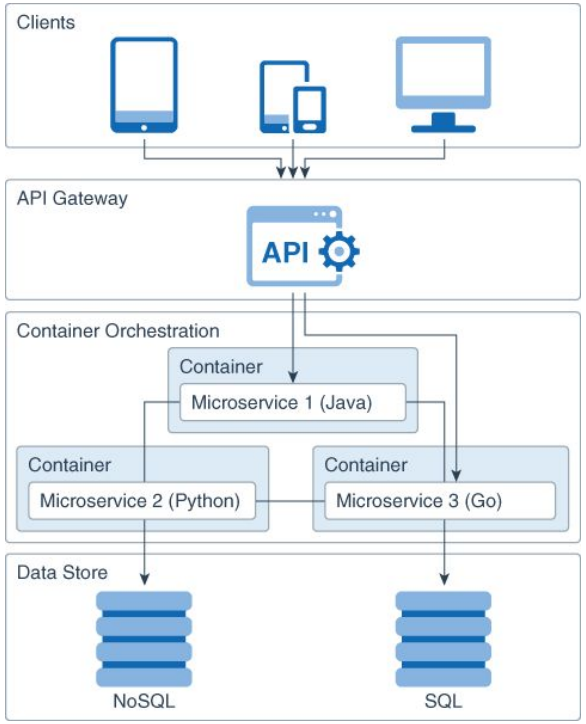


Architecture Microservice

- Cosmos se base sur une architecture similaire à celui d'un microservice

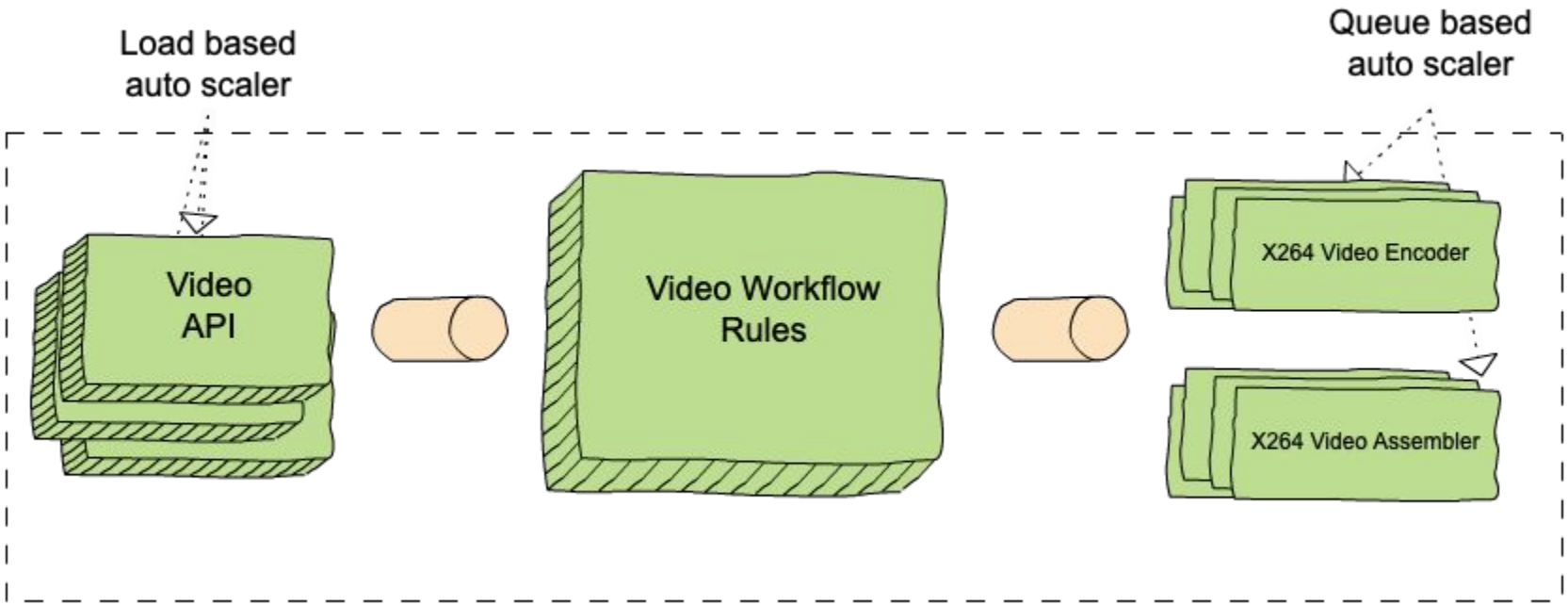
- Permet le déploiement en continue
  - Facilite les tests
  - Facilite le déploiement
  - Facilite la maintenance
- Isolation de fautes
- Possibilité de changement d'outil de développement

- Complexité des systèmes distribués



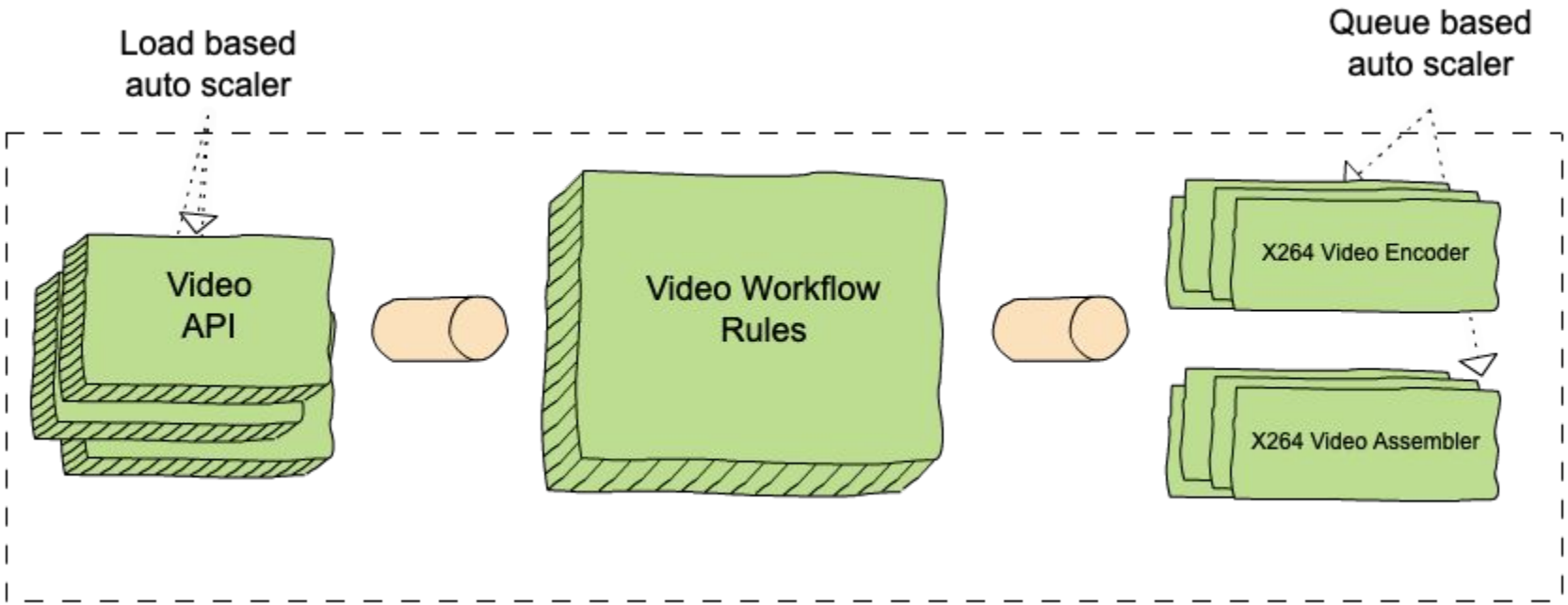
Le service Cosmos

- Même principe, mais ajout d’étapes
- Séparation du business logic

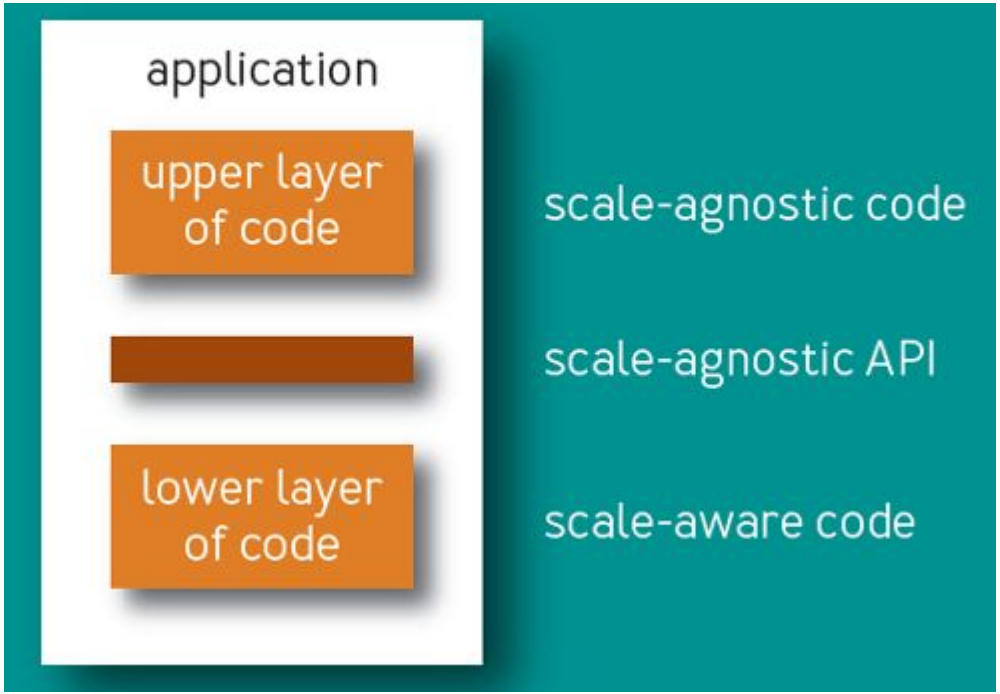


Séparation du business logic

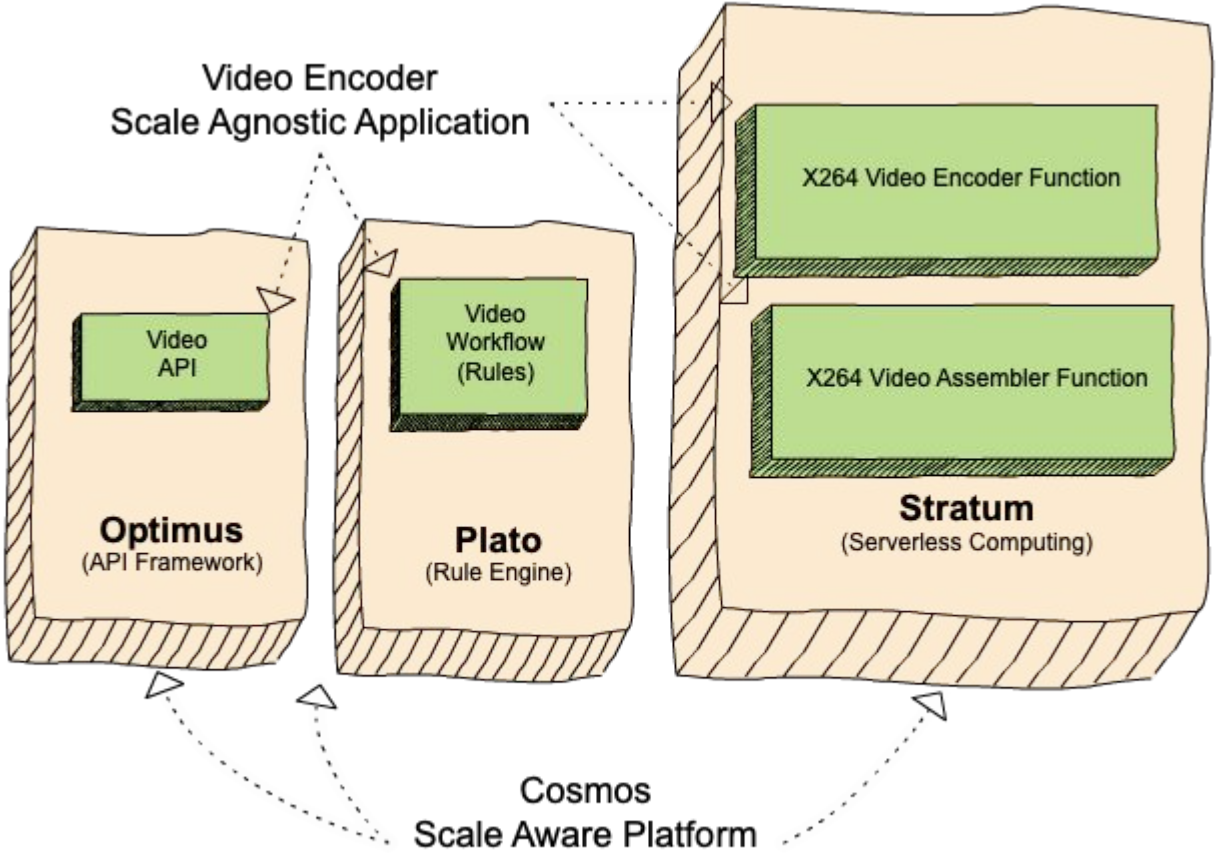
- Plateforme API fournit une abstraction
- Composants scale-agnostic et sous-systèmes scale-aware



Scale-agnostic et Scale-aware

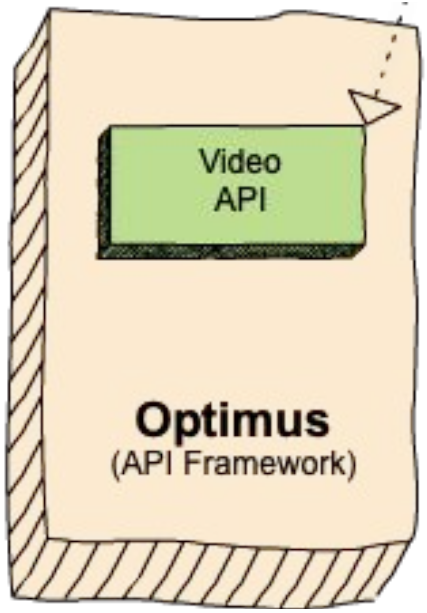


Le service Cosmos



Optimus

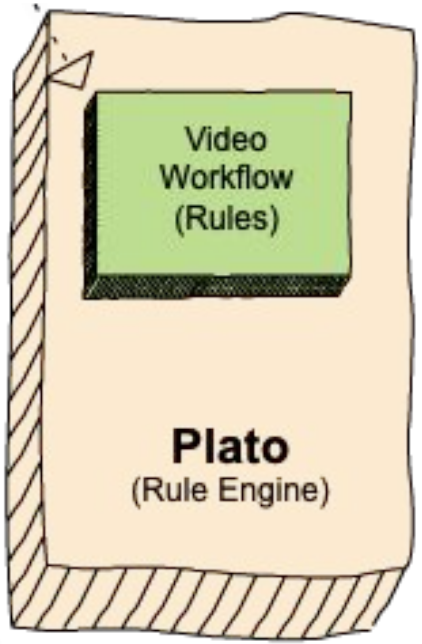
- Mapping entre requêtes et business logic



Plato

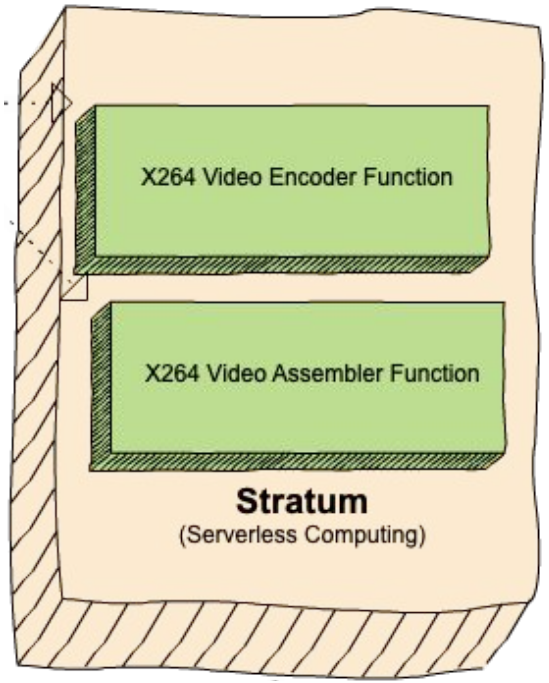
- Orchestration du workflow
- Définit par le développeur

- 1 règle = 4 parties:
  - Statement
  - Action
  - Reaction
  - Error

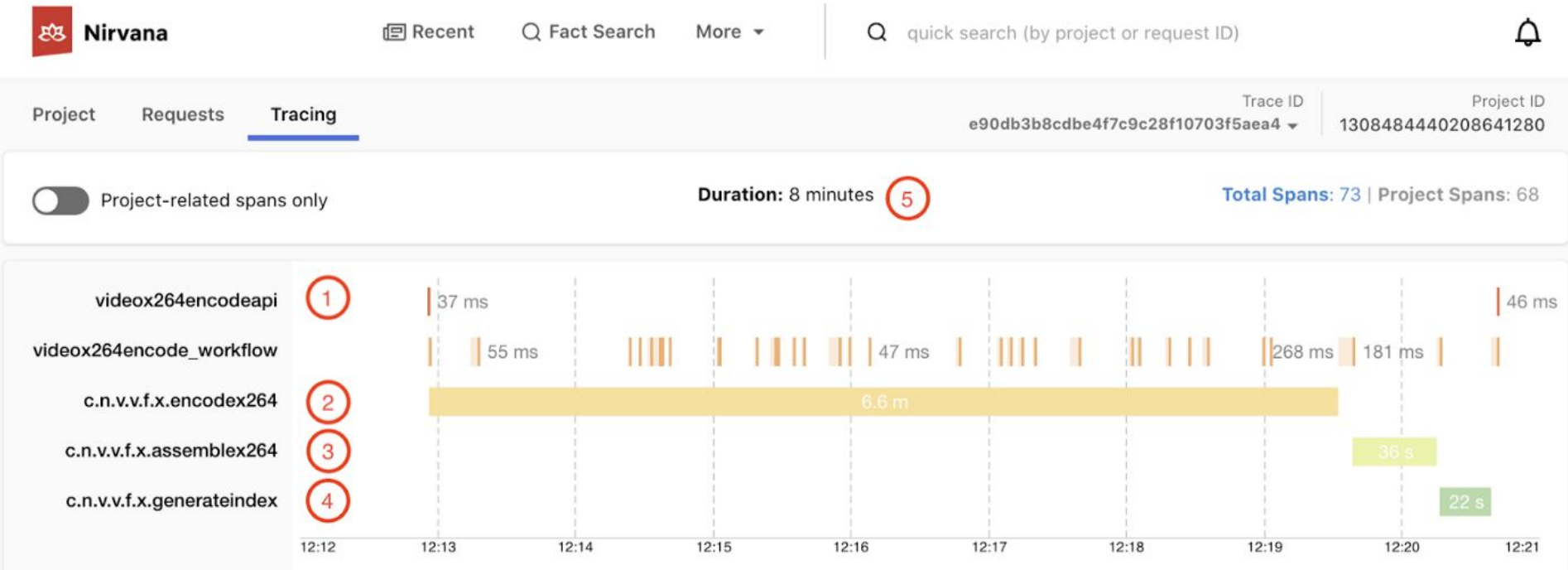


Stratum

- Couche serverless
- Regroupe les fonctions asynchrones



Exemple pour une requête

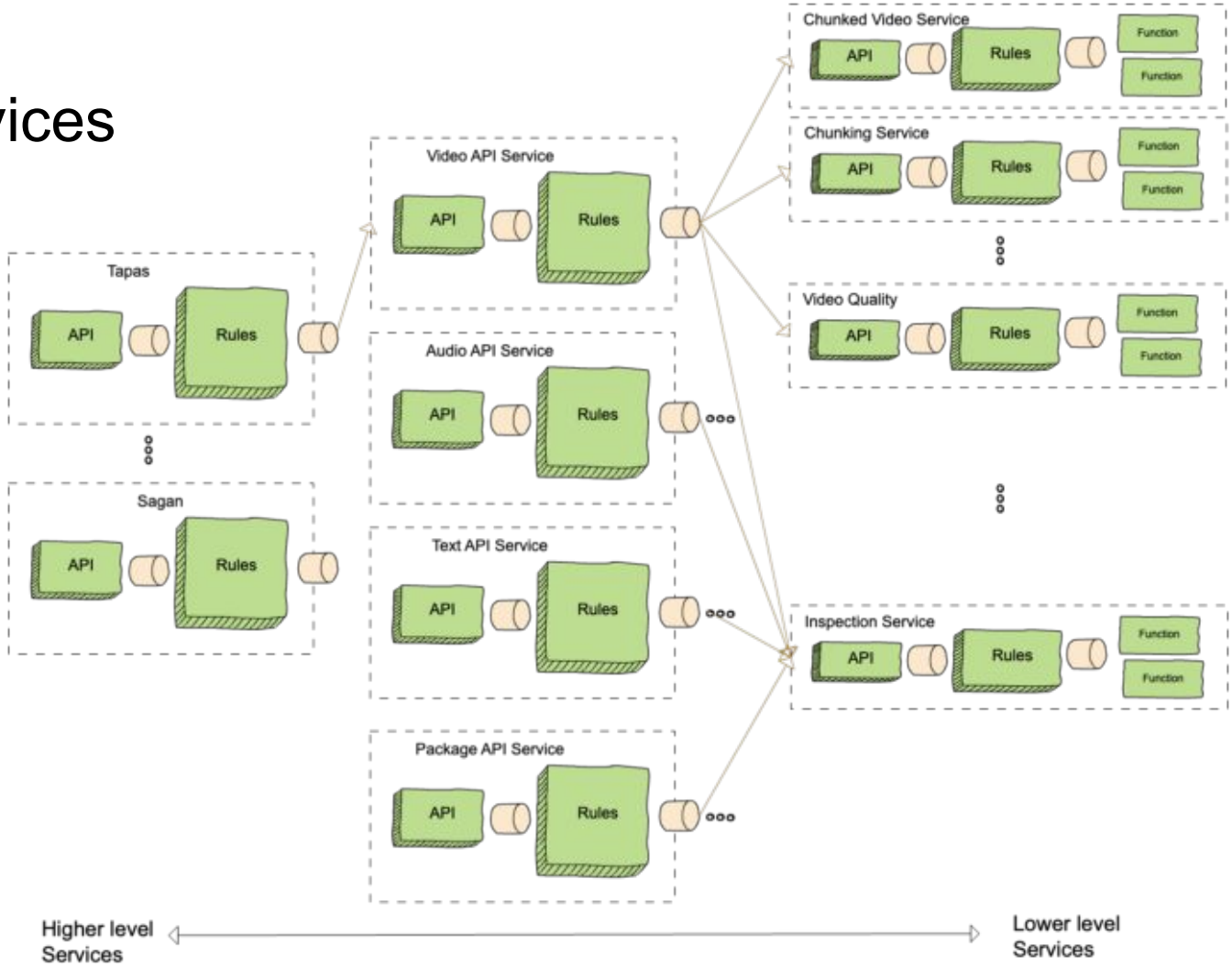


Layering de services

- Décomposition & superposition de services => architecture modulable
- Niveaux des services
- Les services sont orchestrés par des services de plus haut niveaux

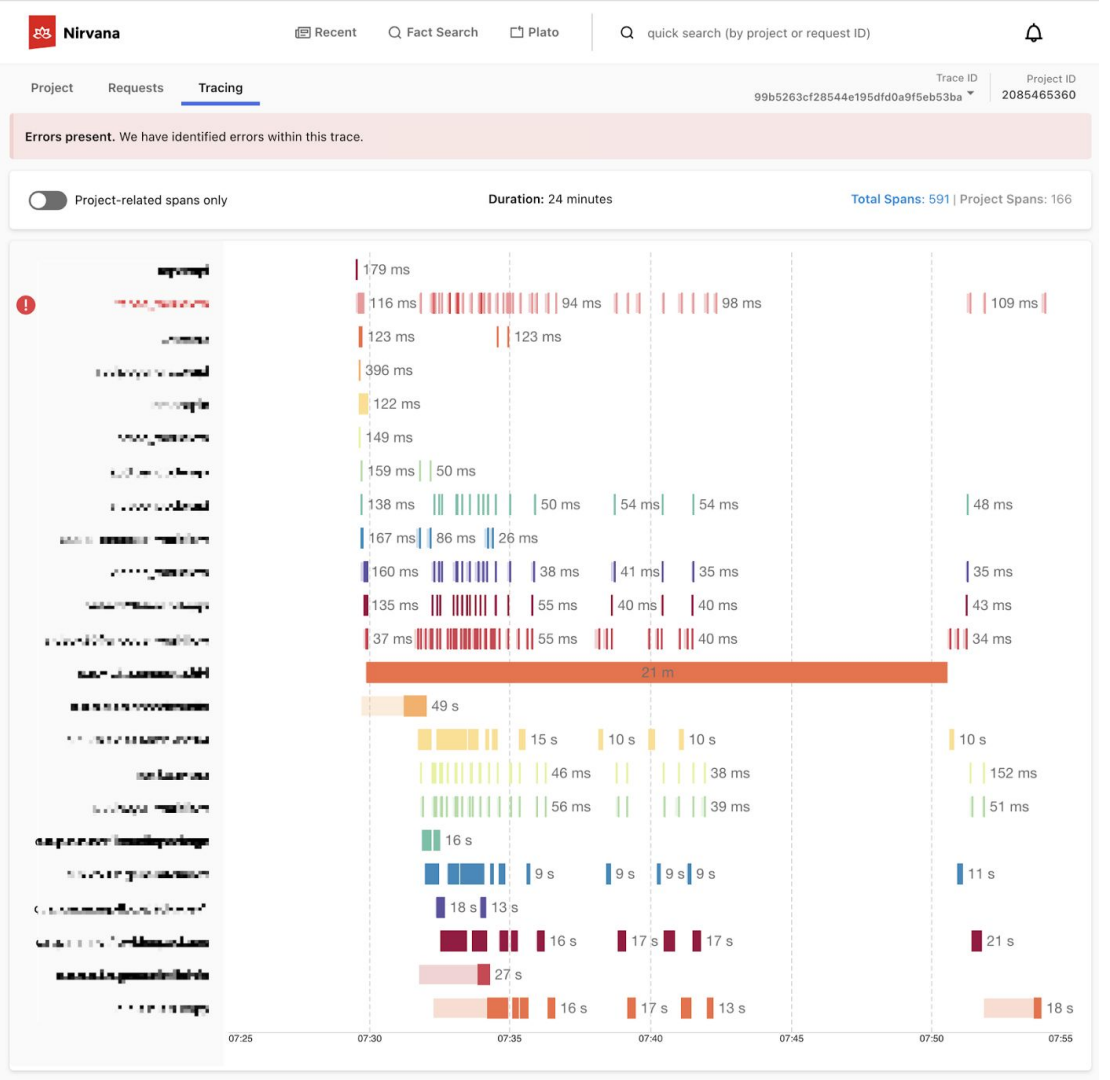
Layering de services

Exemple avec Tapas



Layering de services

Exemple avec Tapas



Netflix Video Quality Service

Mesure de qualité pour les vidéo en streaming :

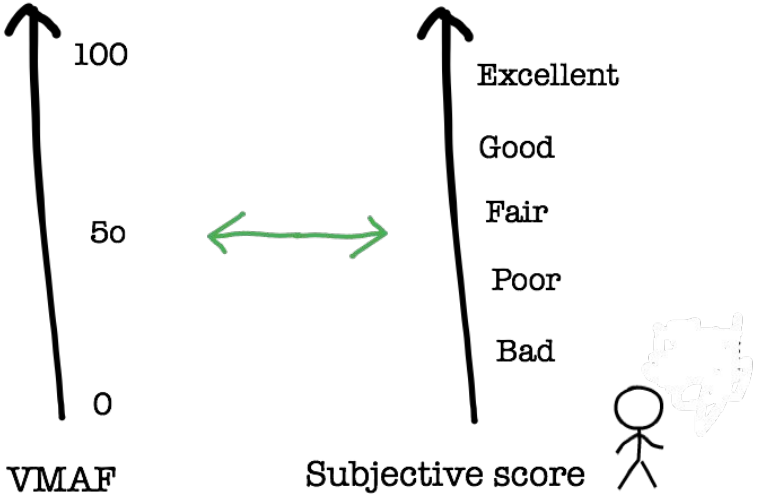
- Video codec comparison
- Amélioration de la Qualité d’Expérience (QoE)
- Video Multimethod Assessment Fusion (VMAF)

Fonctionnement :

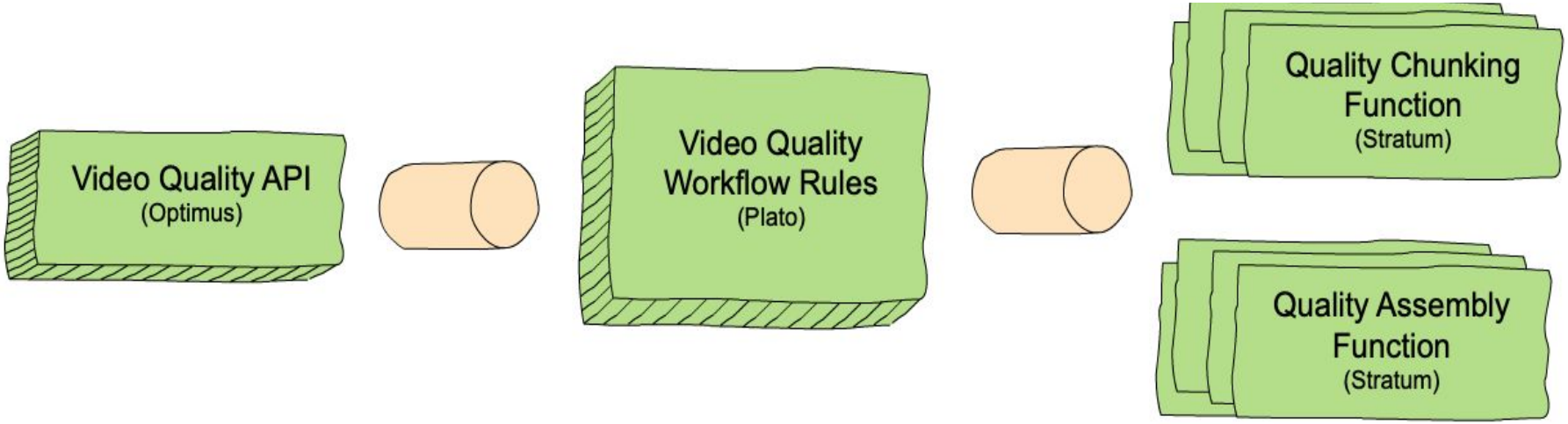
- Découpage de la vidéo en chunk
- Calcule du VMAF sur chaque chunk
- Fusion des valeurs VMAF pour obtenir une mesure VQS

Video Multimethod Assessment Fusion (VMAF)

- Mesure de la qualité du vidéo ressenti



VQS fonctionnement



Points de relais :

- *Request measureQuality(Video source, Video derivative)*
- *VQS getQuality()*

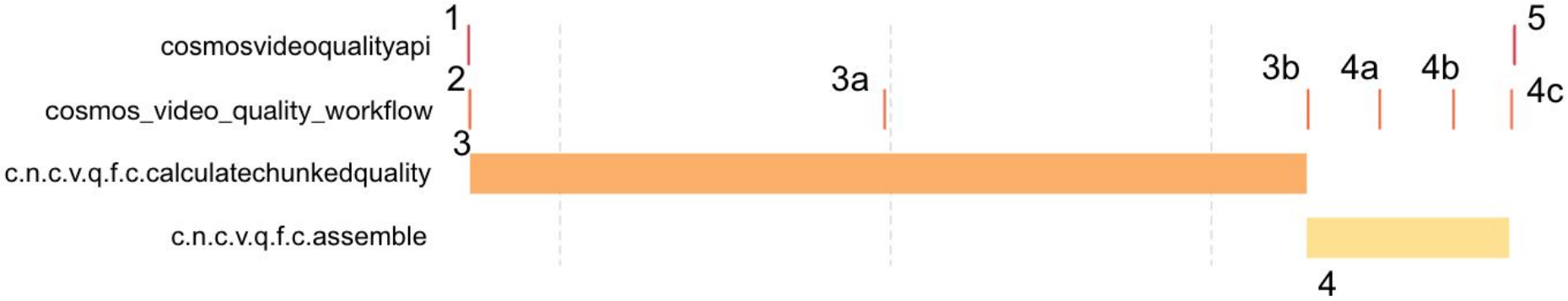
Workflow :

- Découpage en N chunk
- Appelle des fonctions pour le calcul de VMAF
- Appelle de la fonction fusion pour fusionner ces mesures

Couche des fonctions :

- *VMAF computeChunkQuality(chunk c)*
- *VQS assembleChunk(VMAF vmafC[])*

Workflow VQS



Conclusion

- Technologie récente (2019)
- 40 services pour l’instant
- Création d’un nouveau paradigme
- Efficace sur les systèmes complexes
- Perspectives futures