

15ème
édition de la

Journée Française des Tests Logiciels

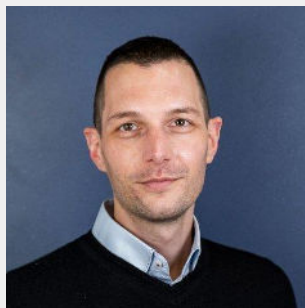


13 juin 2023



Beffroi de
Montrouge

Tester dans un monde plus distribué



Antoine CRASKE



Jean-François FRESI



“Tester dans un monde distribué”

Google avec du contenu R&D de 2013

Google search for "tester event driven". The search results include:

- JKU** (http://fmv.jku.at) - [Modbat: A model-based API tester for event-driven systems](#) (PDF) - Cited by 48. Software testing executes a system under test by giving it a series of inputs and comparing the outputs to expected values. Model-based testing gen- 2 pages.
- TU Wien** (https://publik.tuwien.ac.at) - [Testing Event-Driven Business Processes](#) (PDF) - Cited by 7. The tester can use the collected events for assertions within the test fixture. Integration Tests. An event service can depend on the results of other event ... 12 pages.
- springer.com** (https://link.springer.com) - [Modbat: A Model-Based API Tester for Event-Driven Systems](#) - Cited by 48. Our new tool "Modbat" is specialized for testing systems where these issues are common. Modbat uses extended finite-state machines to model syste...

ChatGPT ne nous aide pas plus

ChatGPT chat interface showing a question: "comment tester des applications orientees événements, ou dites event-driven". The response provides 5 steps to follow:

1. Comprendre les exigences de l'application : Il est important de comprendre les exigences de l'application et les événements qui sont attendus pour déclencher certaines actions. Cela permettra de planifier les tests de manière efficace.
2. Identifier les scénarios de test : Une fois que les exigences sont comprises, identifiez les scénarios de test qui permettent de couvrir tous les événements possibles. Cela peut inclure des tests pour vérifier la réception et le traitement corrects des événements, ainsi que des tests pour vérifier le comportement de l'application dans différentes situations d'événements.
3. Planifier les tests : Planifiez les tests en fonction des scénarios de test identifiés et créez des cas de test pour chacun de ces scénarios. Assurez-vous de couvrir tous les cas possibles d'événements pour garantir une couverture de test complète.
4. Exécuter les tests : Exécutez les tests en suivant le plan de test. Vérifiez que tous les scénarios de test ont été couverts et que l'application fonctionne correctement pour tous les événements.
5. Analyser les résultats : Analysez les résultats des tests pour identifier les problèmes et les anomalies. Si des problèmes sont identifiés, créez des rapports de bogues et attribuez-les aux

Buttons: Regenerate response

Un virage de l'industrie

72%

des entreprises mondiales

utilisent les architectures orientées événements¹.

x10

évolution du marché

De 9B\$ en 2022 à 90B\$ en 2023¹

27%

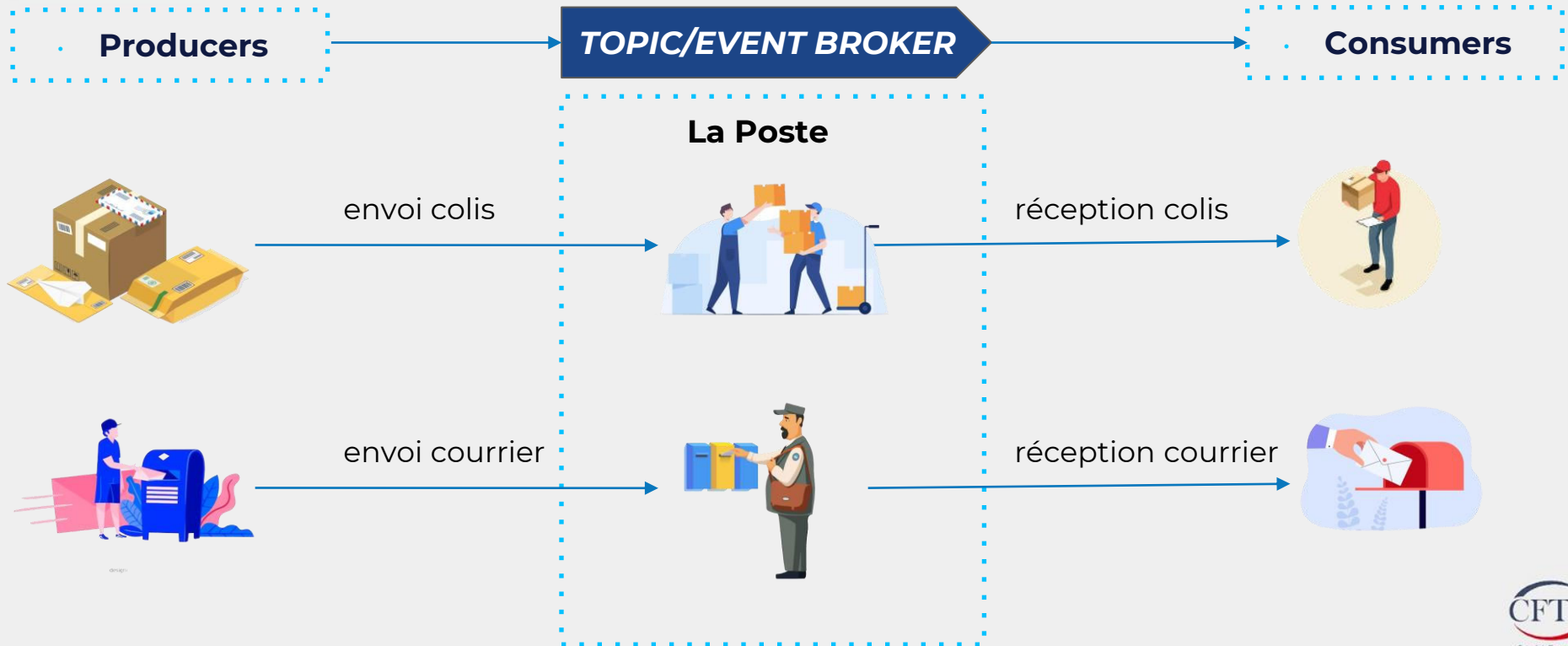
de parts de marché

du serverless jusqu'en 2023²

¹Source: Coleman Parkes global survey on event-driven architecture (EDA), Solace

²Source: Serverless Architecture Market Size By Service, Competitive Market Share & Forecast, 2023 – 2032

C'est quoi une architecture distribuée *event-driven* ?

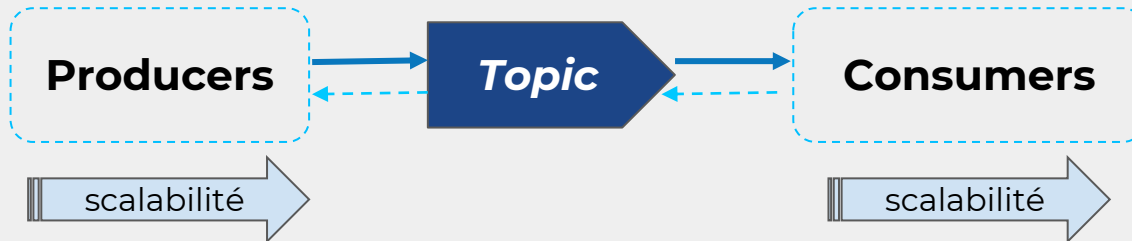


Événements et asynchrones, un processus d'APIs en "2 temps"

API



Événement



¹Source: Serverless Land

Pourquoi faire de l'événement ?

Event Producer



application devis
envoi de commande



application suivi des
expéditeurs



application GPS de
suivi

Event Consumer



gestion trajets



Gestion des tarifs



suivi temps réel



suivi financier



Analyse datas



Augmenter la compréhension du business

Notre monde est asynchrone

Réagir en quasi temps-réel avec évolutivité et fiabilité

Lors de l'expérience client, avec une expansion du SI

Augmenter la flexibilité future

Émission d'un flux fonctionnel facilitant l'ajout de consommateurs

Les événements permettent de mieux s'adapter à la réalité

Libérer

Mainframe



Moderniser

IoT ...



Exploiter

Clients

Business

Partenaires



Flux d'événements d'entreprise

Qui utilise des architectures orientées évènements?



Recommandations

- Recommandation musique
- Recommandation de playlist
- Traitement des logs



Digital Platform

- Système central
- Gestion en temps réel
- Reporting & Analytics



Traitement massif en temps réel

- Inventé Kafka
- Ingestion de très grands volumes de données
- Gestion du temps réel dans le fil d'actualités

Et beaucoup d'autres ...

“80% des Fortune 100 utilisent Apache Kafka”

Source: The Apache Software Foundation

Les challenges des tests d'architecture distribuée *event-driven*



Producer

nouveaux enregistrements
en continu

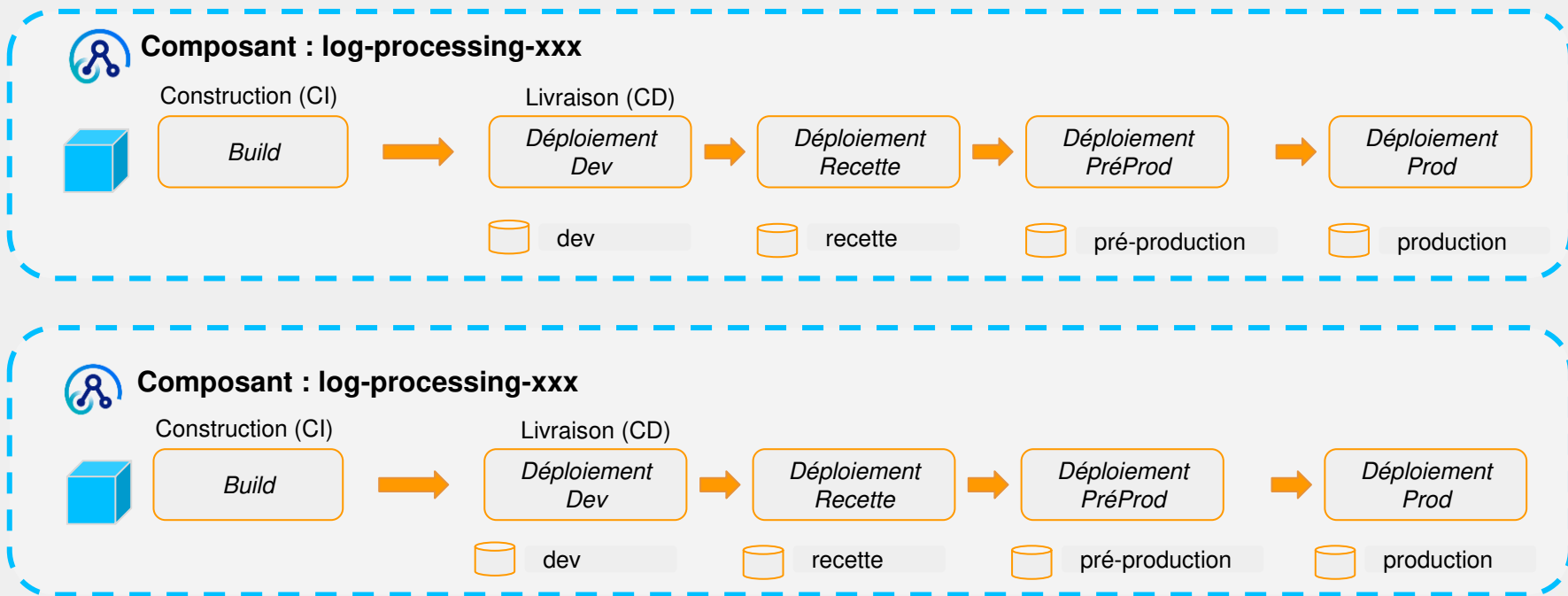
Consumer

consommation
et traitement en continu

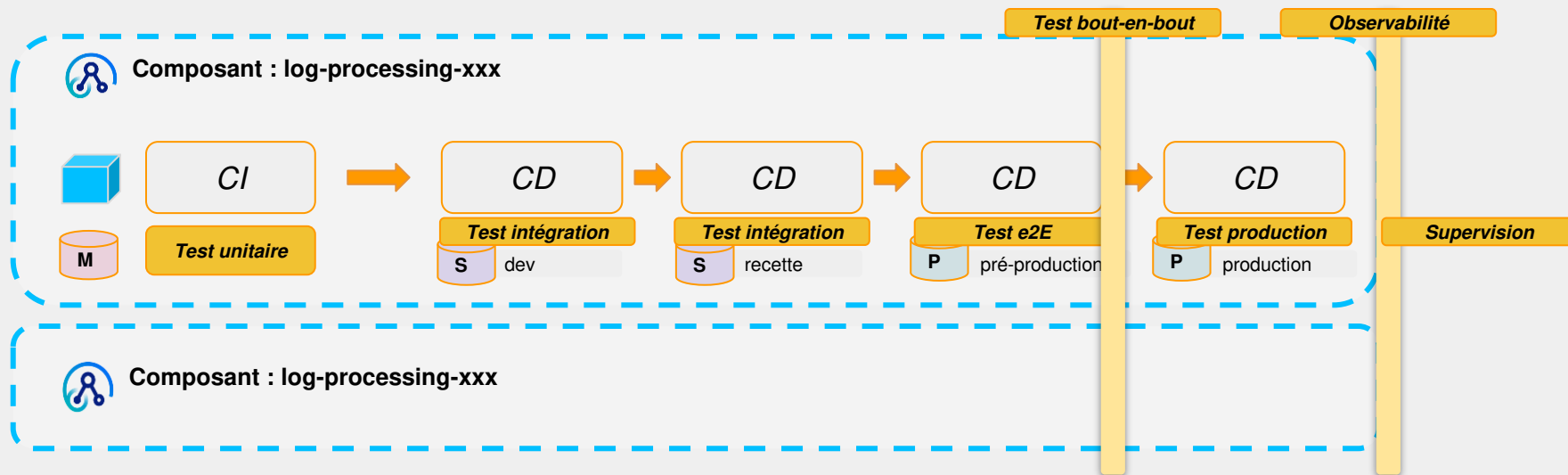


- ➔ Producer testing
- ➔ Consumer testing
- ➔ Producer & Consumer testing
- ➔ Test unitaire
- ➔ Test intégration
- ➔ Test de bout en bout
- ➔ Test production

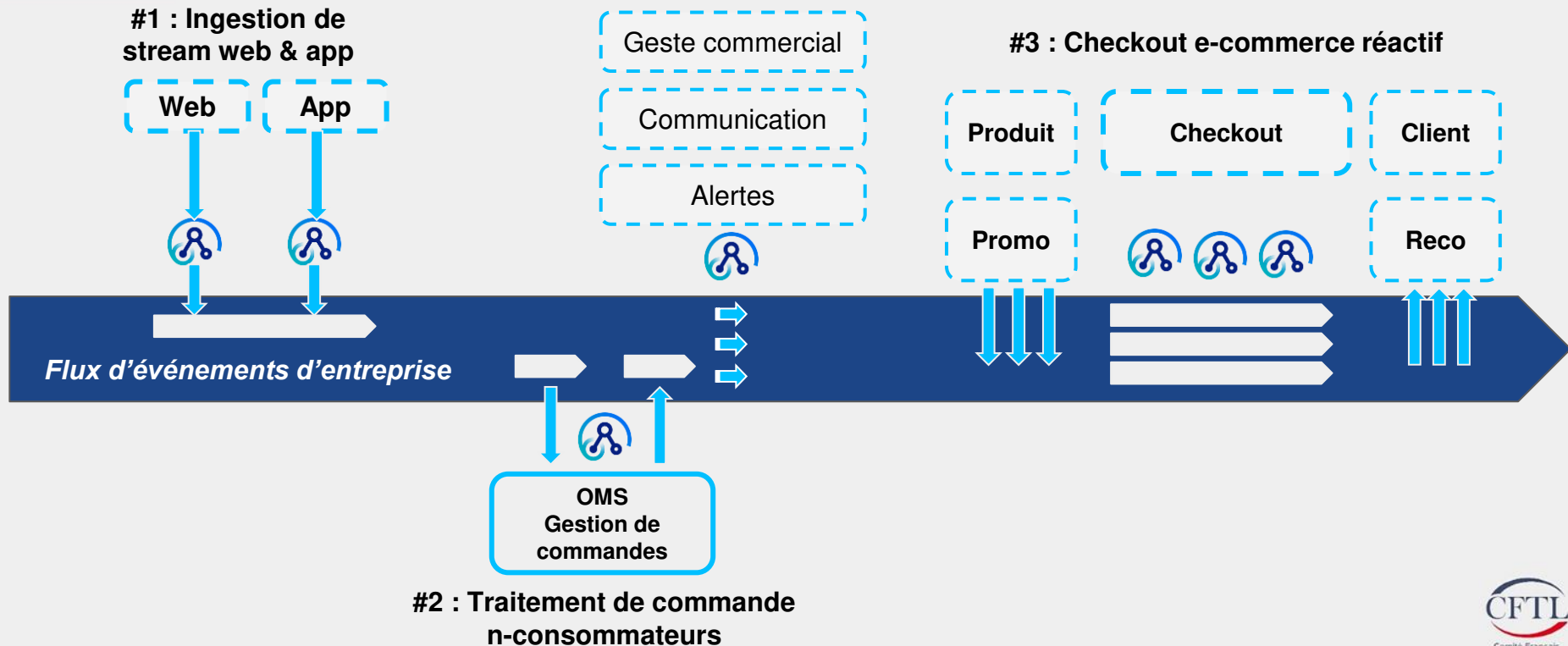
Architecture de déploiement et tests de composants événementiels



Architecture de déploiement et tests de composants événementiels



Les cas d'usages que nous allons couvrir



Cas #1

Interaction utilisateur sur un site web

#1 : Ingestion de
stream web & app

Web

App

Données brutes



Données formatées

Flux d'événements d'entreprise

Pics à plusieurs millions
d'événements par jour

Google
Analytics

Log Site Web
marchand

Log Dynatrace

Log Application
facturation

Log App gestion
stock



Service
promotion

Service Fidélité

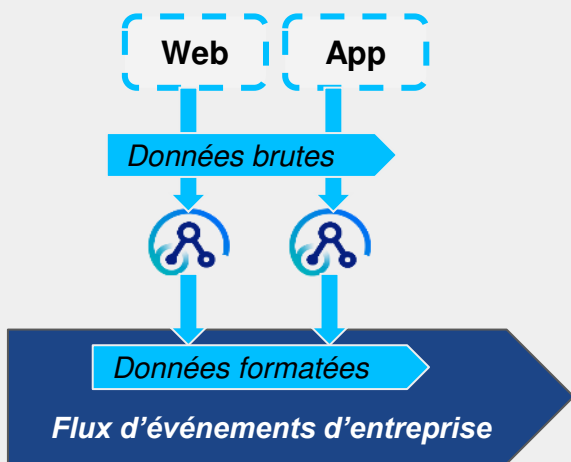
Service prod &
perf

Service
commandes

Cas #1

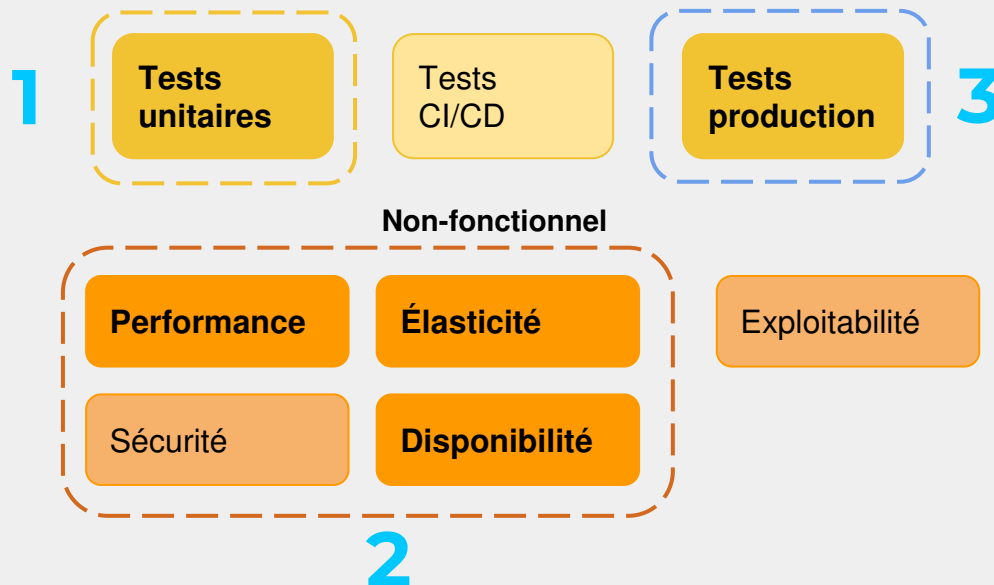
Quoi et comment tester ?

#1 : Ingestion de
stream web & app



Pics à plusieurs millions
d'événements par jour

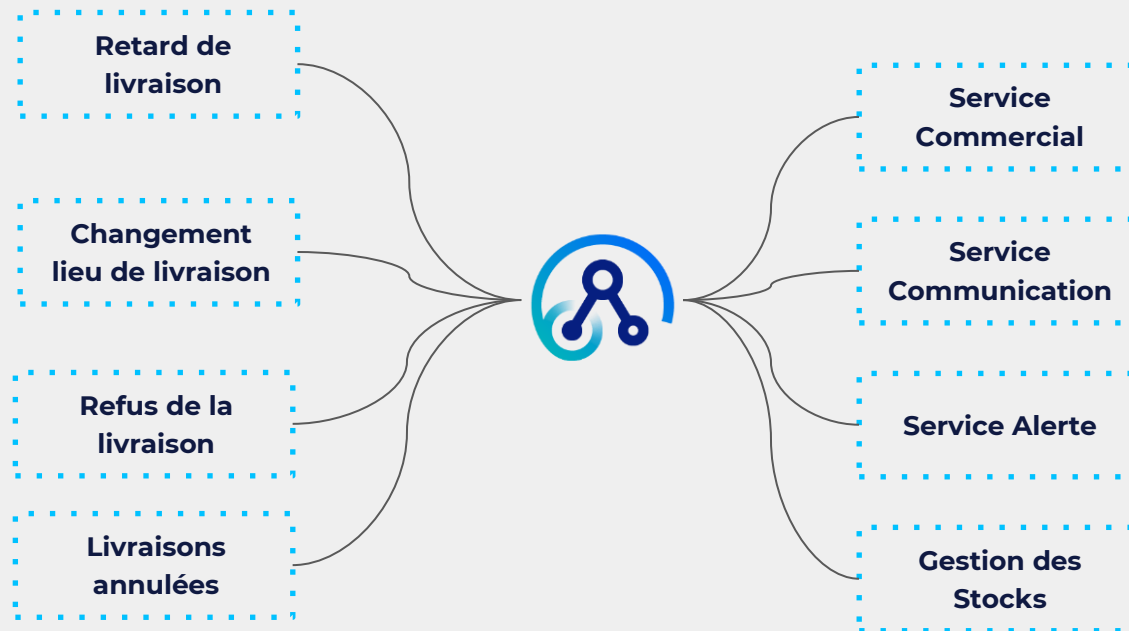
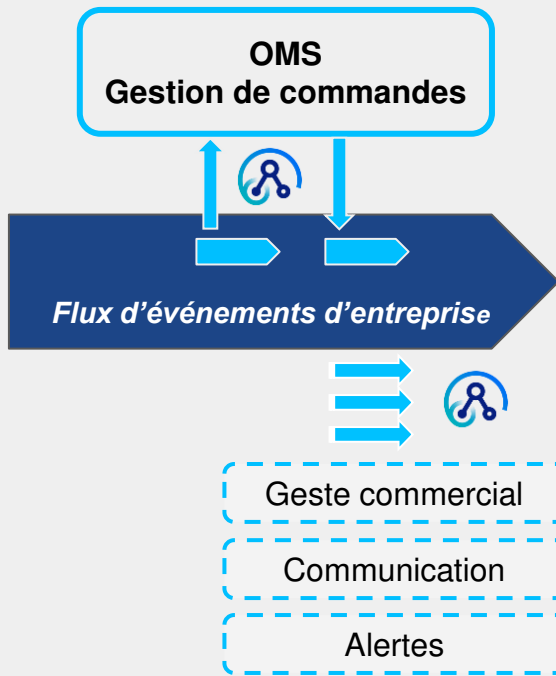
Typologie de tests par composant



Cas #2

Traitement de commande n-consommateurs

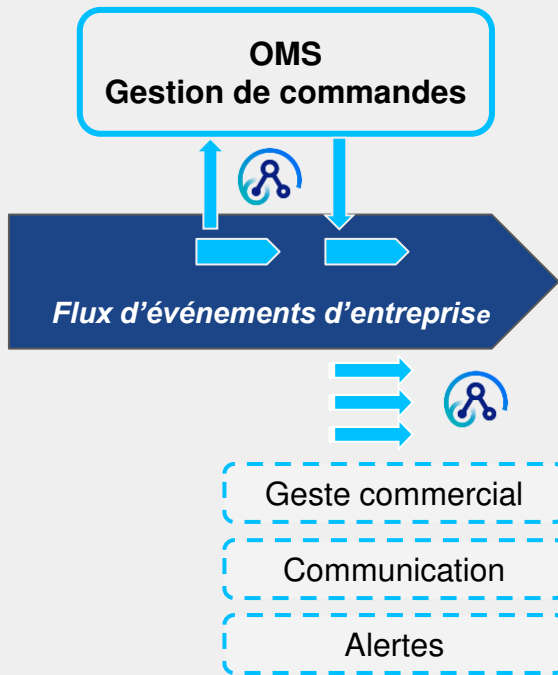
#2 : Traitement de commande
n-consommateurs



Cas #2

Quoi et comment tester ?

#2 : Traitement de commande
n-consommateurs



2

Tests bout-en-bout

Typologie de tests par composant

1

Tests
unitaires

Tests
CI/CD

Supervision

4

Non-fonctionnel

Performance

Maintenabilité

Exploitabilité

Intégrité

Disponibilité

Sécurité

3

Cas #3

Checkout e-commerce réactif

#3 : Checkout e-commerce réactif

Produit

Checkout

Client

Promo



Reco

Flux d'événements d'entreprise

Reco produits



Reco services

Reco produits
complémentaires

Panier

Compte

Paiement

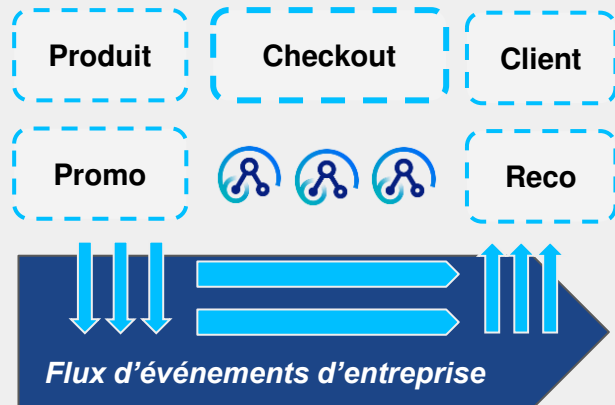
Livraison

Promos

Cas #3

Quoi et comment tester ?

#3 : Checkout e-commerce réactif

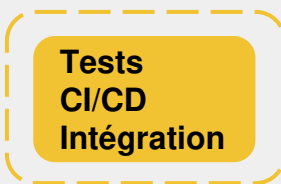


2

Tests bout-en-bout

Typologie de tests par composant

1



4

3

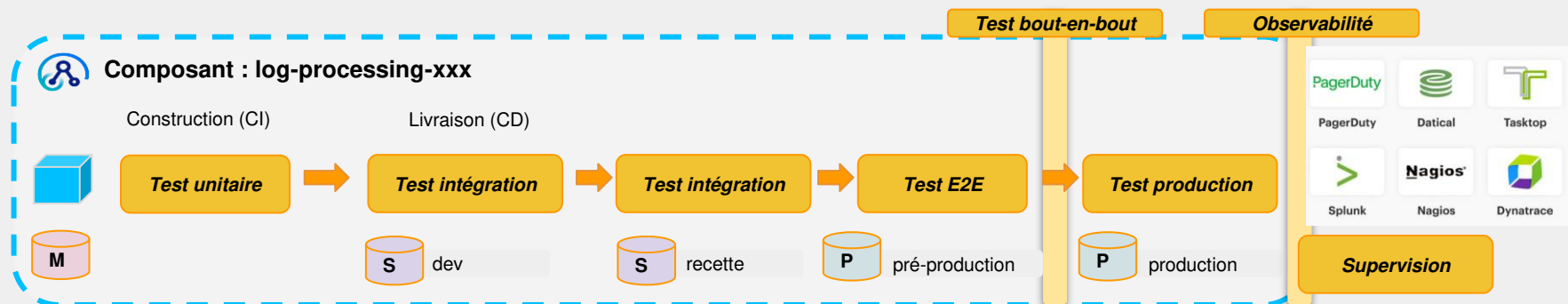
Non-fonctionnel



Comment automatiser ?

Une architecture de Quality Engineering similaire au test d'API, en sécurisant :

1. **Connecteurs**
2. **Gestion de l'asynchronisme**
3. **Intégration outillage (CI/CD, notifications, ...)**



- XUnit (JUnit)
- Mock (Mockito)
- Librairie
 - Kafka
 - AWS SQS



Pattern & anti-patterns

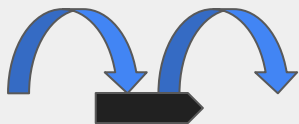
Anti-Patterns

- Approche locale test entrées/sorties APIs
- Négliger l'asynchronisme des données
- Vouloir tout tester

Patterns

- S'adapter au contexte
- Équilibrer Shift left et Shift Right Testing
- Focus fort sur fonctions et données

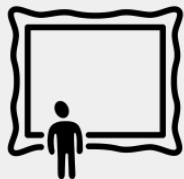
Takeways



Tests asynchrones = tests d'APIs en "2-temps"



Une complexité temporelle du flux de données



Garder une vision d'ensemble

Et pour terminer ...

“
Il y a un but, mais pas de chemin ; ce que nous nommons
chemin est hésitation.”



Franz Kafka

15ème
édition de la
**Journée
Française
des Tests
Logiciels**



13 juin 2023



Beffroi de
Montrouge

Merci de votre écoute !



Comité Français
des Tests Logiciels

QE
UNIT

The Quality Engineering Community