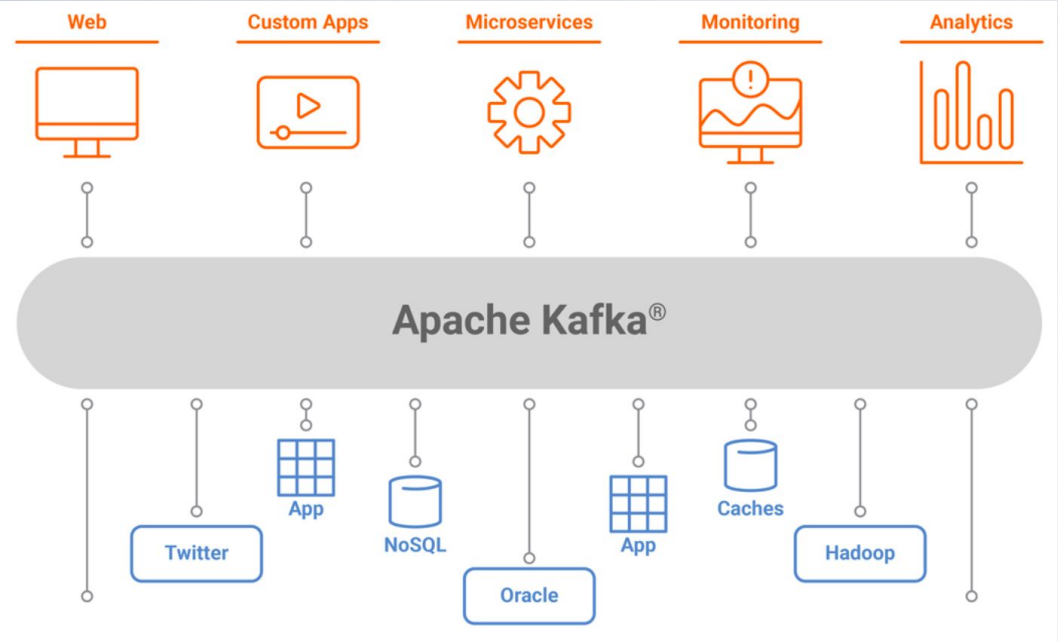
Kafka Streams

Tom SOLVERY
Ergi SALA



Kafka est généralement utilisé dans deux domaines d'application:

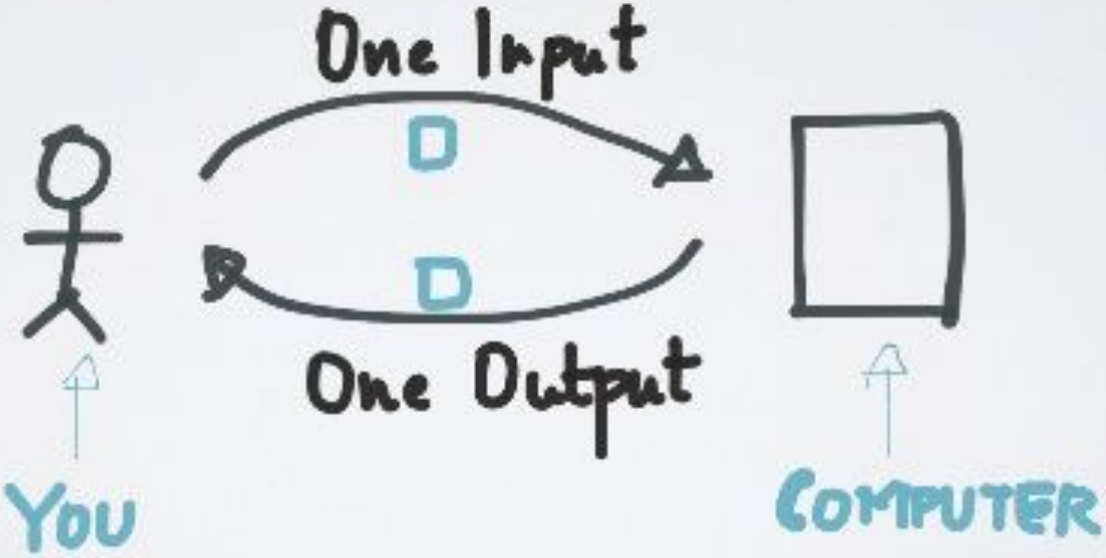
- Développement de pipeline de streaming de données en temps réel qui transfèrent les données d'une manière fiable entre systèmes et applications.
- Développement d'applications de streaming qui transforment ou réagissent aux flux des données.

Apache Kafka® est une plateforme de flux de données distribuées.

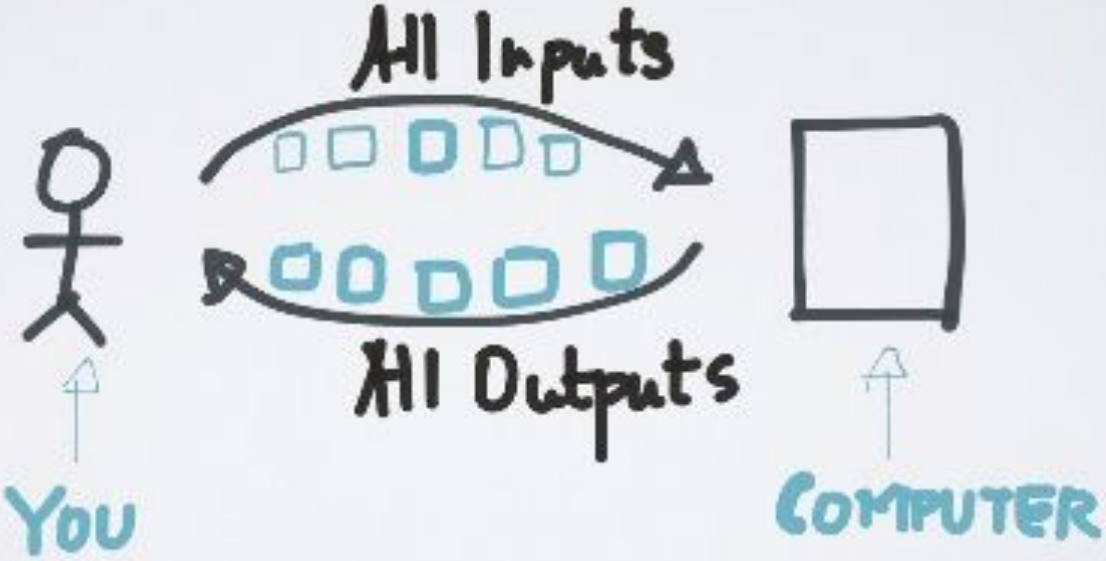
Une telle plateforme possède trois propriétés principales:

- Publier et s'abonner aux flux de données, similaire aux systèmes de messagerie.
- Sauvegarder l'historique dans une manière durable et résistant aux pannes.
- Traiter les flux en temps réel.
- Un cluster Kafka sauvegarde le flux en catégories qui s'appellent *topics*. Chaque enregistrement contient une clé, une valeur et un estampille horaire.

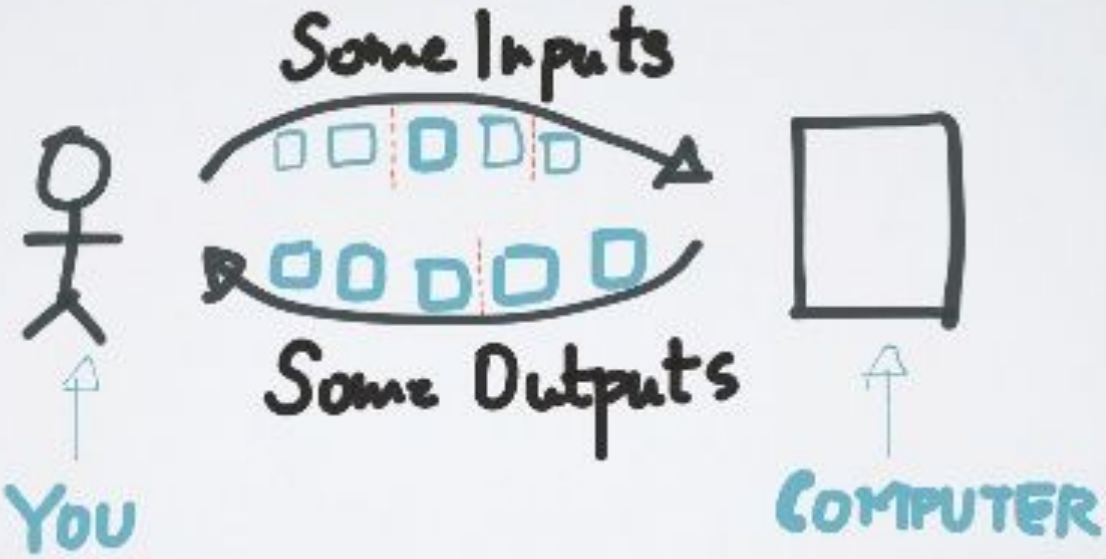
REQUEST/RESPONSE



BATCH

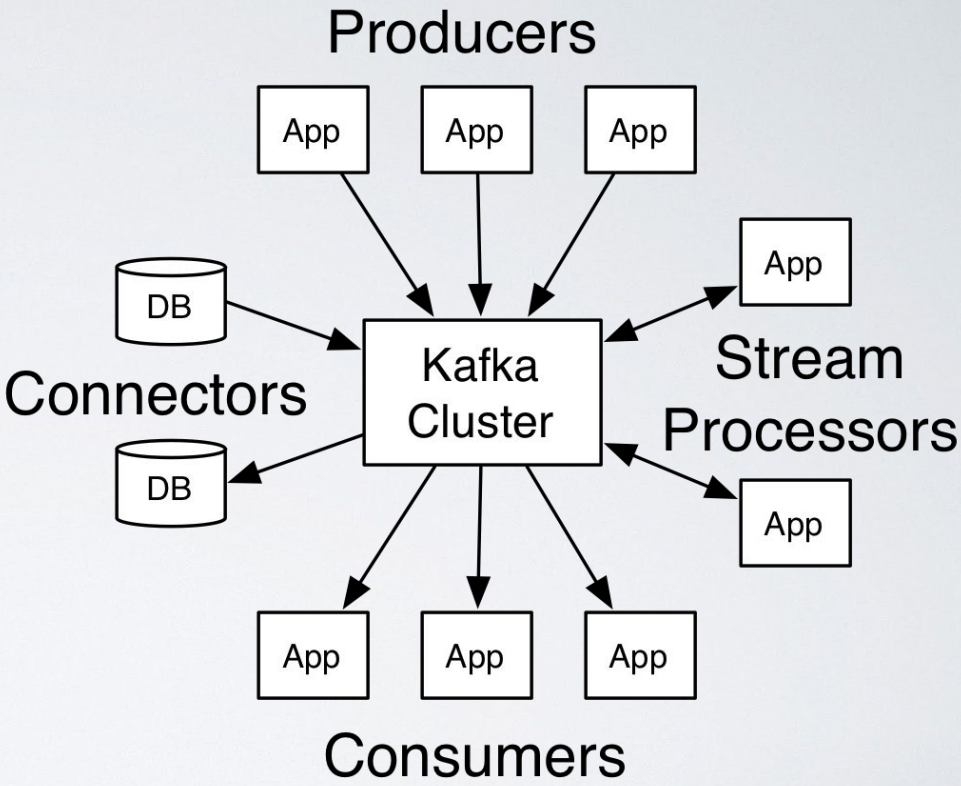


STREAM PROCESSING



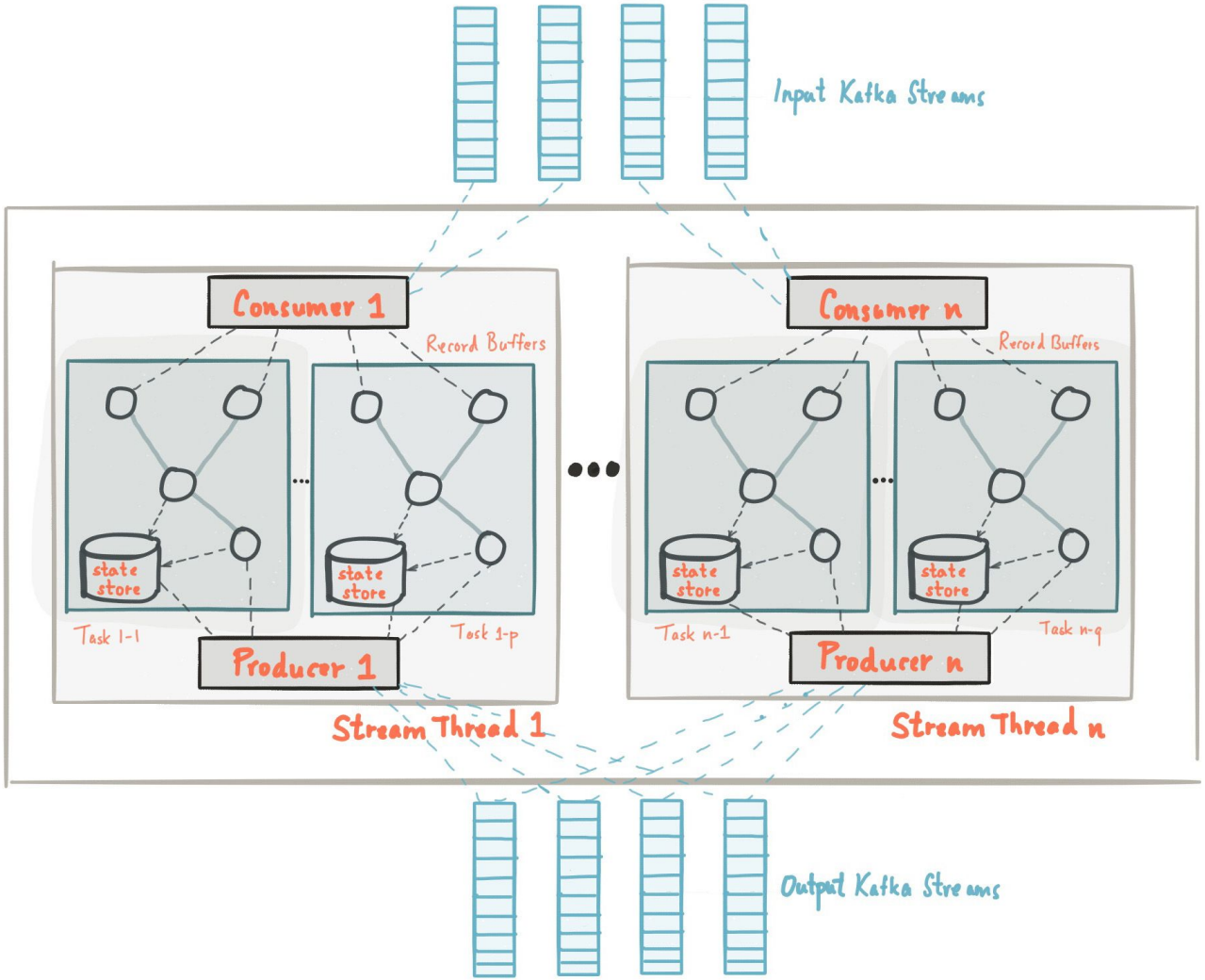
Kafka possede quatre APIs:

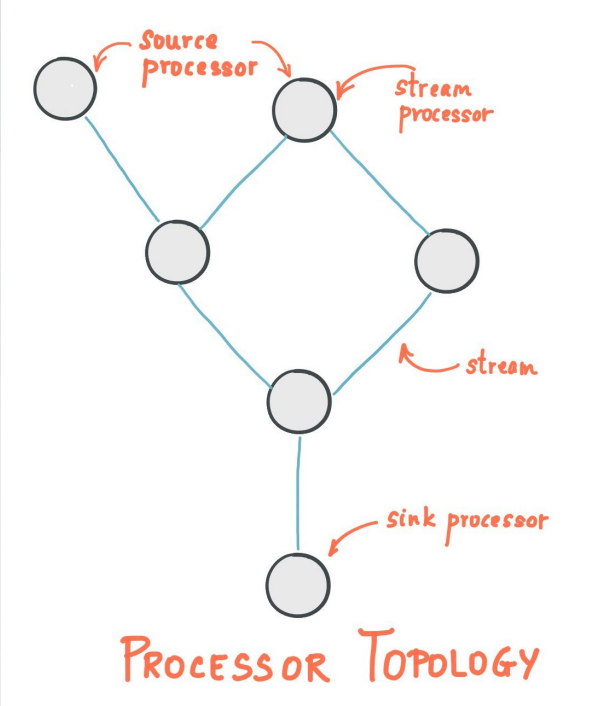
- Le Producer API permet une application de publier un flux de données dans un ou plusieurs Kafka topics.
- Le Consumer API permet à une application de s'abonner à un ou plusieurs topics.
- Le Streams API permet à une application de traiter le flux des enregistrements en consommant l'entrée à partir d'un ou plusieurs topics et en produisant un flux de sortie sur un ou plusieurs topics.
- Le Connector API permet le développement et l'exécution de producteurs et consommateurs réutilisables qui se connectent aux topics Kafka existants sur des applications ou base de données. Par exemple une connection a une base de données relationnel peut être utilisé pour détecter les changements dans les tableaux.



Kafka Stream

- Bibliothèque permettant de faire des applications de stream processing
- Facile à utiliser
- Distribuée
- Tolérance aux pannes


Input Kafka Streams
Consumer 1
Record Buffers
state store
state store
Task 1-1
Task 1-p
Producer 1
Stream Thread 1
Consumer n
Record Buffers
state store
state store
Task n-1
Task n-q
Producer n
Stream Thread n
Output Kafka Streams

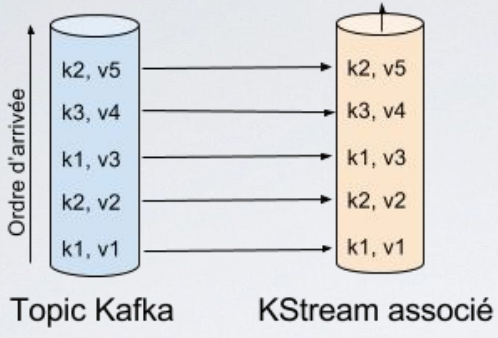


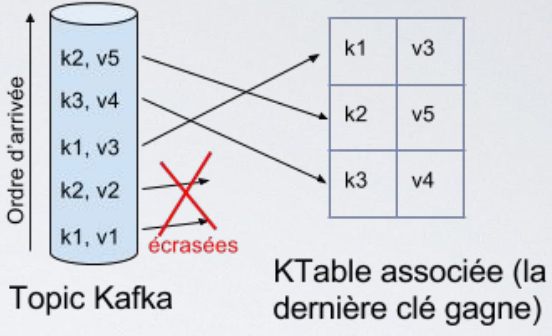
Une application est composé en différents noeuds

Chaque noeud réalise une transformation sur le flux de donnée

Source processor : noeud qui récupère les données provenant du consommateur.

Sink processor : noeud qui produit les données pour le producteur.





Chaque élément du flux de donnée est représenté par une paire clé/valeur.

- KStream : donnée sous forme d'un flux continue sans interruption
- KTable : donnée compactée sous une table gardant seulement la dernière valeur pour chaque clé

Exemple d’utilisation :

- New York Times : utilise Kafka stocker et publier en temps réel le contenu des articles sur différentes sites et applications.

- Trivago : utilise Kafka pour récupérer librement les données de la société afin de faire des analyses et exploiter les sources qu’ils possèdent

Demo:

- Word-Count
- Music Application

Sources:
Apache Kafka: https://kafka.apache.org/
Apache Kafka Intro: https://kafka.apache.org/intro
Kafka Streams Presentation:
https://www.slideshare.net/GuozhangWang/introduction-to-kafka-streams/4?fbclid=IwAR1ATbejTS71QlXMpRuN0HFGAe-mIqfvkU1Y20SzpyrYoXp9dfVCVZ9S_8Q
Confluent: https://docs.confluent.io/current/kafka/introduction.html
Blog des octos : https://blog.octo.com/kafka-streams-encore-un-framework-de-stream-processing/