



10 min \Rightarrow 1 composant Confluent

10 minutes pour implémenter une gouvernance des données avec Schema Validation et Schema Registry

Olivier Laplace, Solutions Engineer, Confluent France



01 - Schema Validation & Registry



Confluent Platform 5.5

Operations and Security

Security plugins | Role-Based Access Control

Control Center | Replicator | Auto Data Balancer | Operator

Audit Logs | **Schema Validation** | Tiered Storage | Multi-Region Cluster

Development & Stream Processing

Connectors

Clients | REST Proxy
MQTT Proxy | **Schema Registry**

KSQLDB

Apache Kafka

K-Connect

Continuous Commit Log

K-Streams

**Mission-critical
Reliability**

Complete

**Event Streaming
Platform**

Self-Managed Software

Fully-Managed Service

Freedom of Choice

Datacenter

Public Cloud

Confluent Cloud

Data Governance = several domains



- **Data Catalog**

Question: What data do I have and what is it about?

- **Data Lineage**

Question: Where is this data coming from and going to?

- **Data Quality**

Question: How can I make sure that no bad data is polluting my topics?

- **Data Security**

Question: Who can access my data, what can they view, how is it being protected?

⇒ Schema Registry + Schema Validation = Data Quality ₄

Deploy Confidently in Production

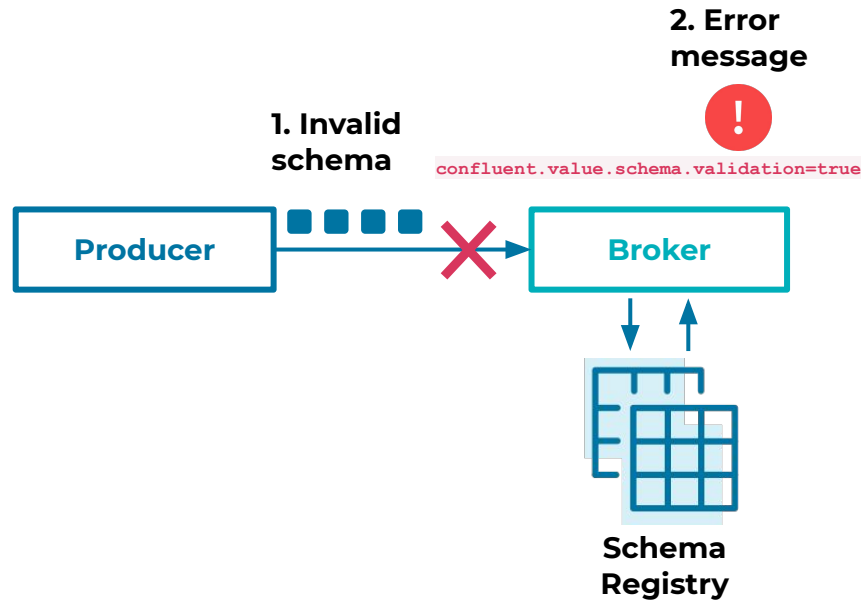
Scale schemas reliably

- **Automated broker-side** schema validation and enforcement
- **Direct interface** from the broker to Confluent Schema Registry

Granular control

- **Enabled validation** at the topic level

Schema Validation



Enable Application Development Compatibility

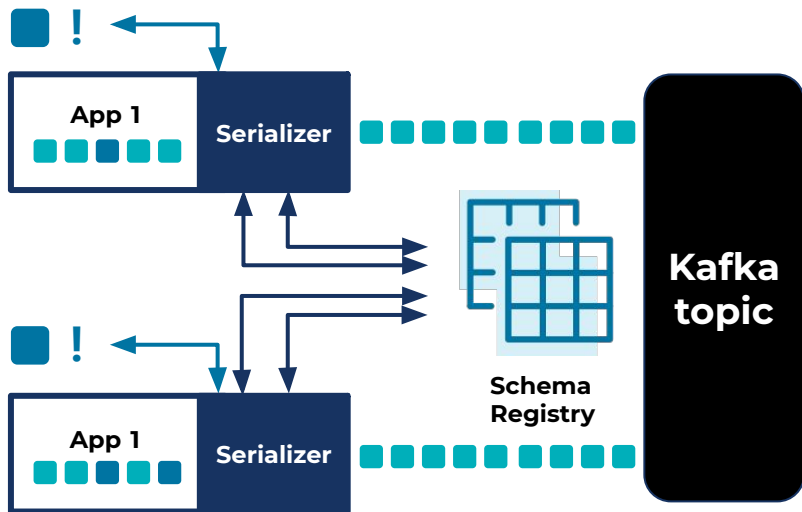
Develop using standard schemas

- **Store and share** a versioned history of all standard schemas
- **Validate data compatibility** at the client level

Reduce operational complexity

- **Avoid time-consuming** coordination among developers to standardize on schemas

Schema Registry



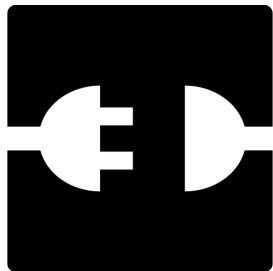
Support multiple format in Schema Registry and throughout the Confluent Platform



Plugins



Custom plugin



Schema Registry

Schema Id: 1
Type: Avro

Schema Id: 2
Type: Protobuf

Schema Id: 3
Type: JsonSchema

Schema Id: 4
Type: Custom



ksqlDB

Kafka Streams

Kafka Connect

Control Center

REST Proxy

CLI

Kafka Clients

Support in Schema Registry

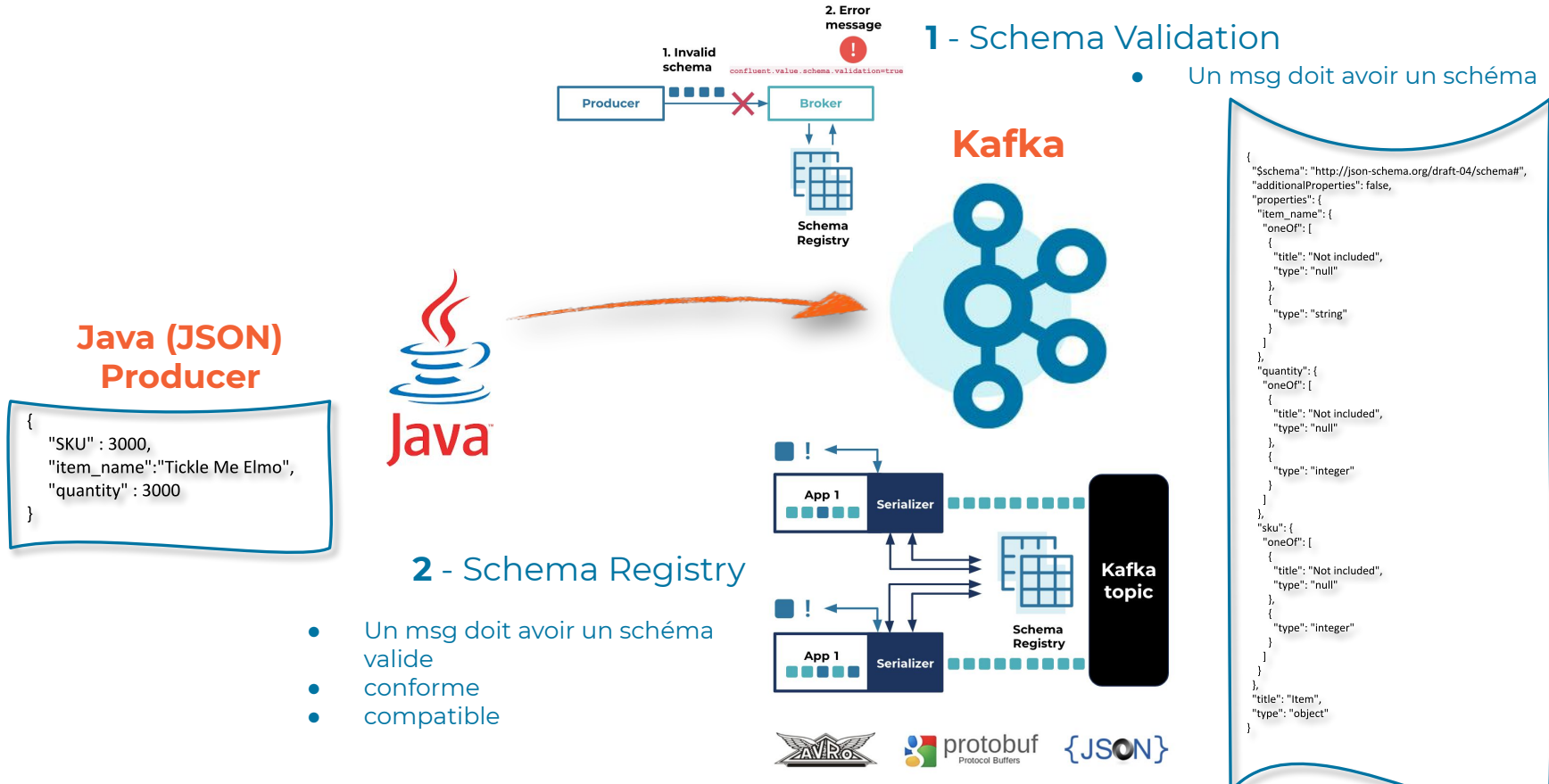


- Schemas management and data compatibility with **Avro**, **Protobuf** and **JSON**.
- **Compatibility** levels previously available in Avro have been extended to Protobuf and JSON: BACKWARD, FORWARD, FULL.
- **New serdes** available, `KafkaProtobufSerializer` and `KafkaProtobufDeserializer` for Protobuf and `KafkaJsonSchemaSerializer` and `KafkaJsonSchemaDeserializer` for JSON Schema.
- **Open framework** to add new/custom schema types in Schema Registry.
 - Schema Registry plugins can be build by implementing the new interfaces *SchemaProvider* and *ParsedSchema*.
 - Robert's blog post on "Playing chess with Confluent Schema Registry" that exemplifies how custom plugins work.
 - <https://yokota.blog/2020/02/26/playing-chess-with-confluent-schema-registry/>



02 - Implémentation (JSON Schema)

Topology of the demo: Send different JSON msg



Pré-requis et informations



- **Téléchargez et installez Confluent Platform 5.5**
 - Téléchargement : <https://www.confluent.io/download>
 - Documentation : <https://docs.confluent.io/current/getting-started.html>
 - Pour la démo : Confluent 5.5 installé on-prem
- **Pour produire les messages JSON**
 - Option 1 : utilisez *kafka-json-schema-console-producer*
 - Option 2 : utilisez la classe *ProducerJsonDemo.java* (lien GIT page suivante)
 - *pom.xml* présent dans GIT
- **La démo montre le côté Producteur de messages**
 - La même chose est possible / recommandée pour les consommateurs
- **Il est possible de gérer plusieurs schémas par Topic**
 - `value.subject.name.strategy = RecordNameStrategy` (défaut: `TopicNameStrategy`)

Accédez aux éléments de la démo sur GIT



- **Pour cette démo spécifiquement :**

<https://github.com/olaplace/Confluent10Minutes/tree/master/SchemaRegistry>



03 - Cas d'usage et valeur

Cas d'usage



- **Data Quality, Data Quality and... Data Quality**
- **Validate that a message has a schema (when it should)**
- **Validate that a message is valid in regard to its schema**
- **Insure compatibility between versions of messages**
- **Manage Data Quality with all components of Confluent Platform**

VALEUR :

Garantir la qualité des messages entrants et sortants.

Valider la cohérence métier et technique des messages.

Gérer l'évolution du modèle de données dans le temps avec le niveau de compatibilité des schémas approprié.

⇒ Utiliser Kafka comme référentiel dans la gestion d'évènements, sans compromis quant à la qualité technique et fonctionnelle des données



Merci !

olaplace@confluent.io

<https://developer.confluent.io/>

<https://www.confluent.io/blog/>

<https://docs.confluent.io/current/>