CONFLUENT

Getting Started with Confluent Schema Registry

Patrick Druley

Senior Solution Engineer @confluentinc

Agenda

https://www.linkedin.com/in/patrickdruley/

twitter: @PatrickLovesAK

1. Schema-less Kafka
2. Schema Registry Basics
3. 3 Good Habits
4. Schema Validation Demo

Schema-less Kafka

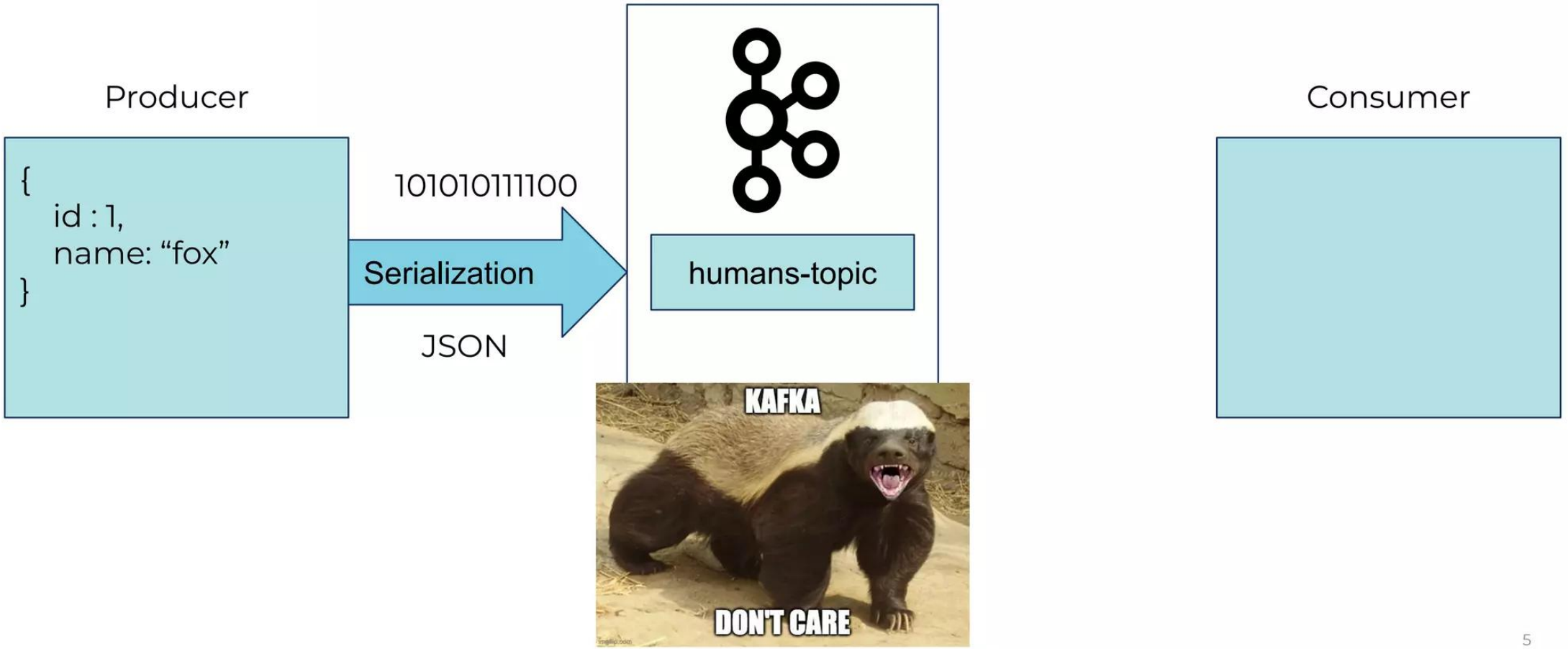
"Someone changed the date field from unix timestamp to datetime and broke all our reporting dashboards"

"I need to add a new field to support new functionality, but I don't want to update ALL the potential downstream consumers of this data"

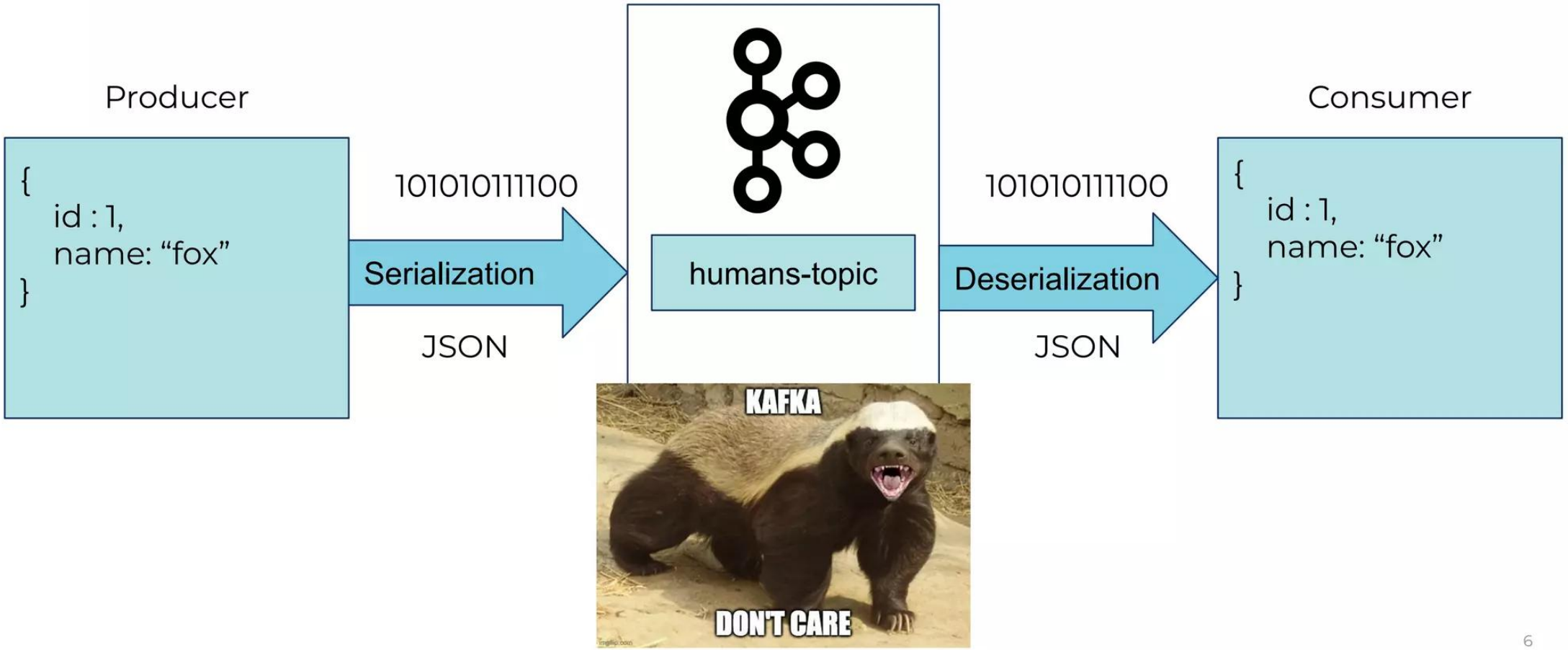
"My compliance team wants an audit to ensure there is no PII data in Kafka, there's no easy way to get all the message metadata"

"My data is 'democratized' in Kafka, but my dev teams have no way to know what data is *in* Kafka without domain knowledge or coming to my platform team"

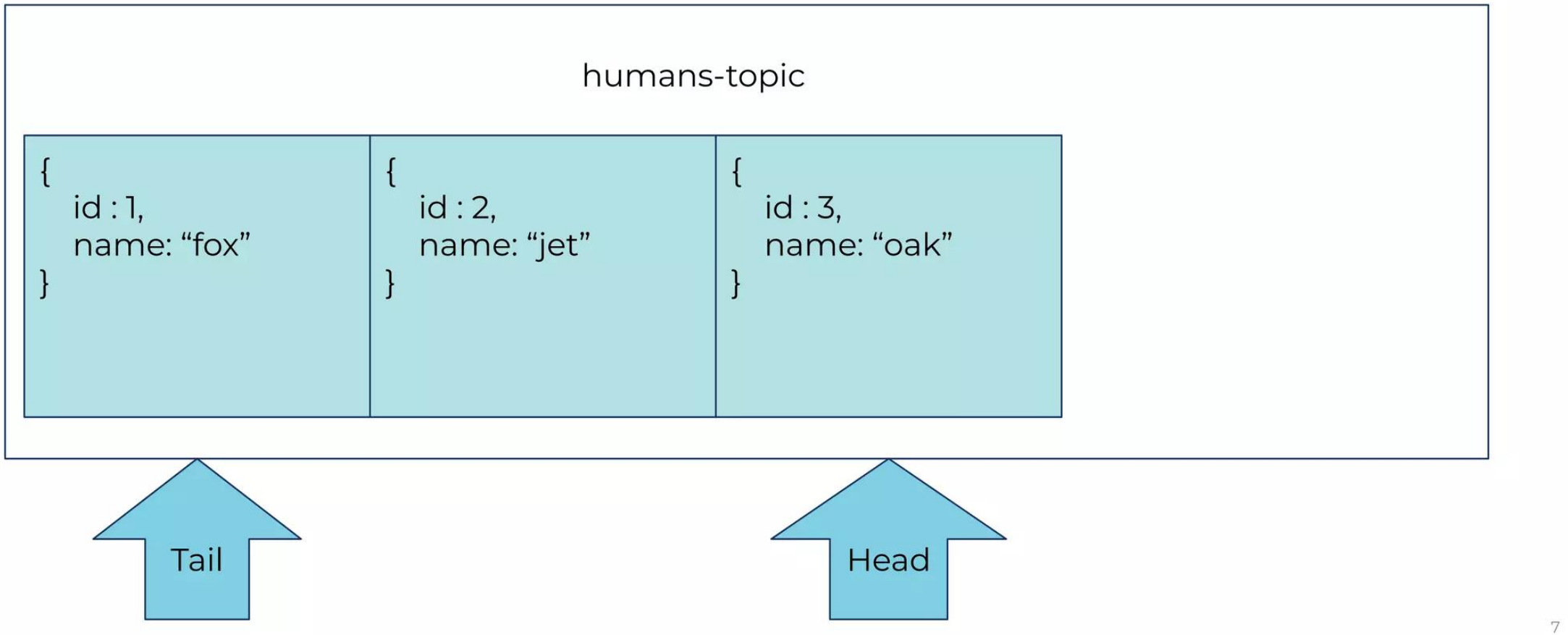
“My event is JSON” - Producer



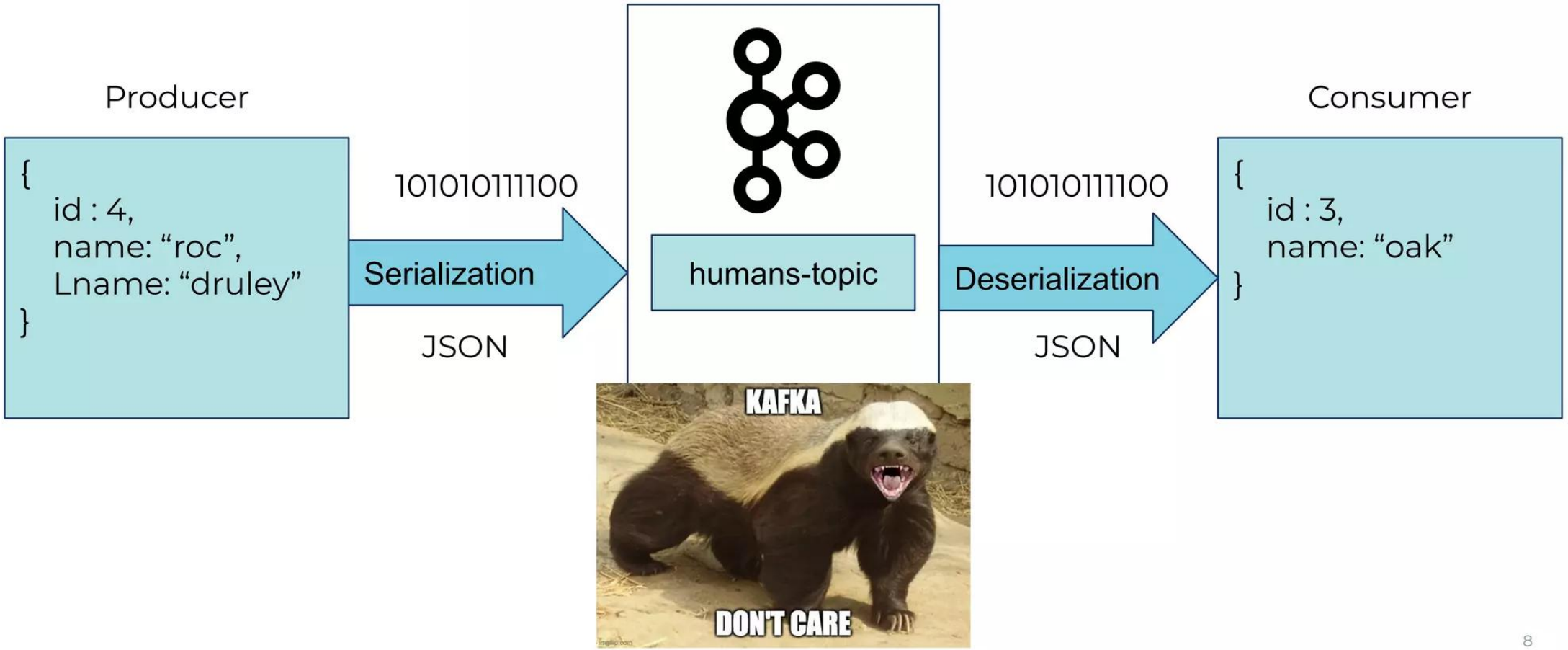
“Sounds great” - Consumer



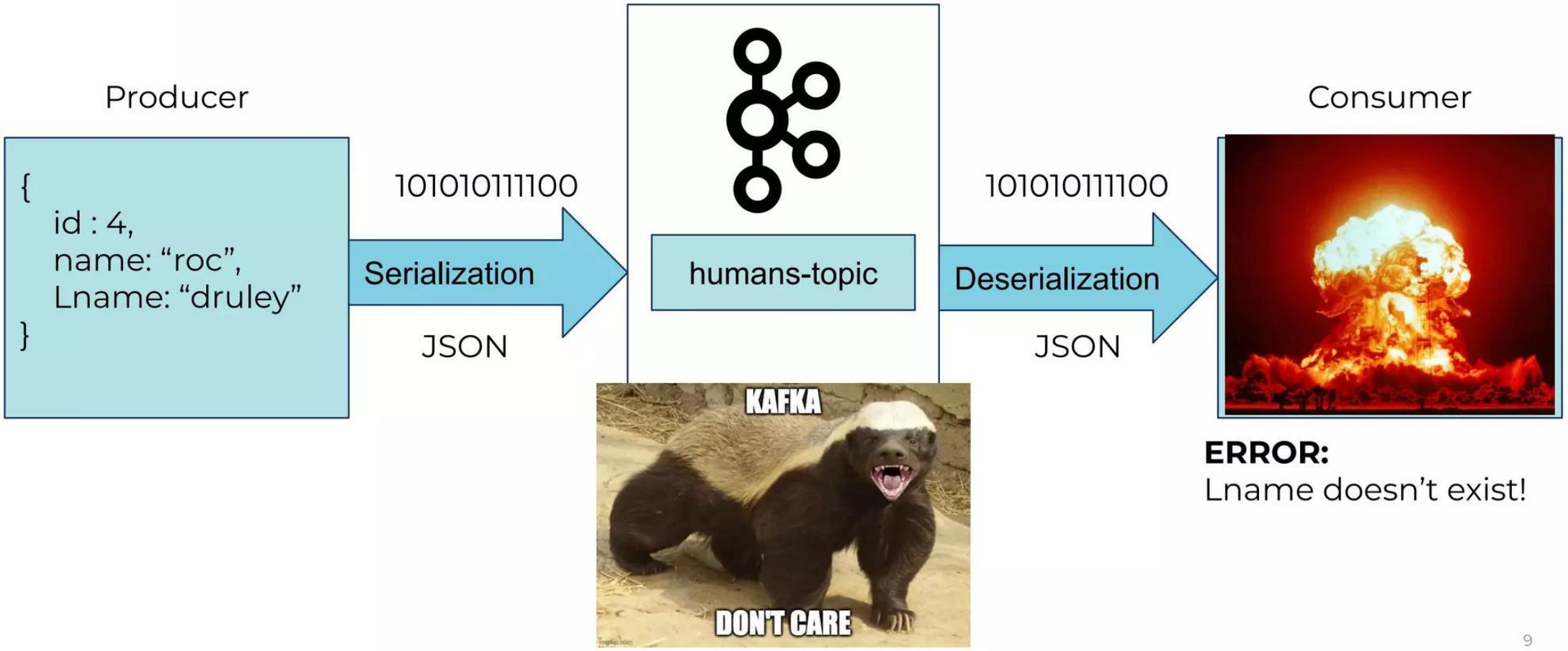
Time Passes



“I added a field” - Producer



“You did what?” - Consumer

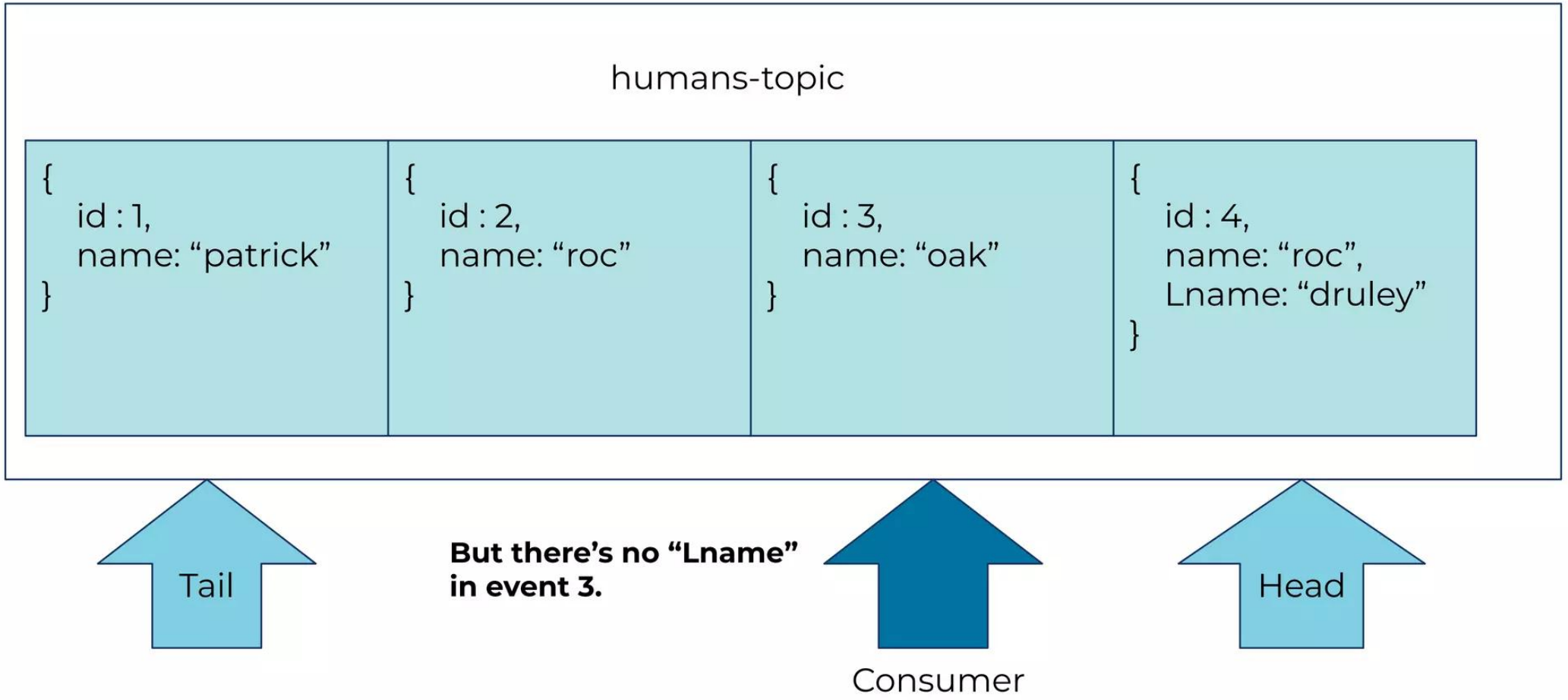


ERROR:
Lname doesn’t exist!

What Happened?

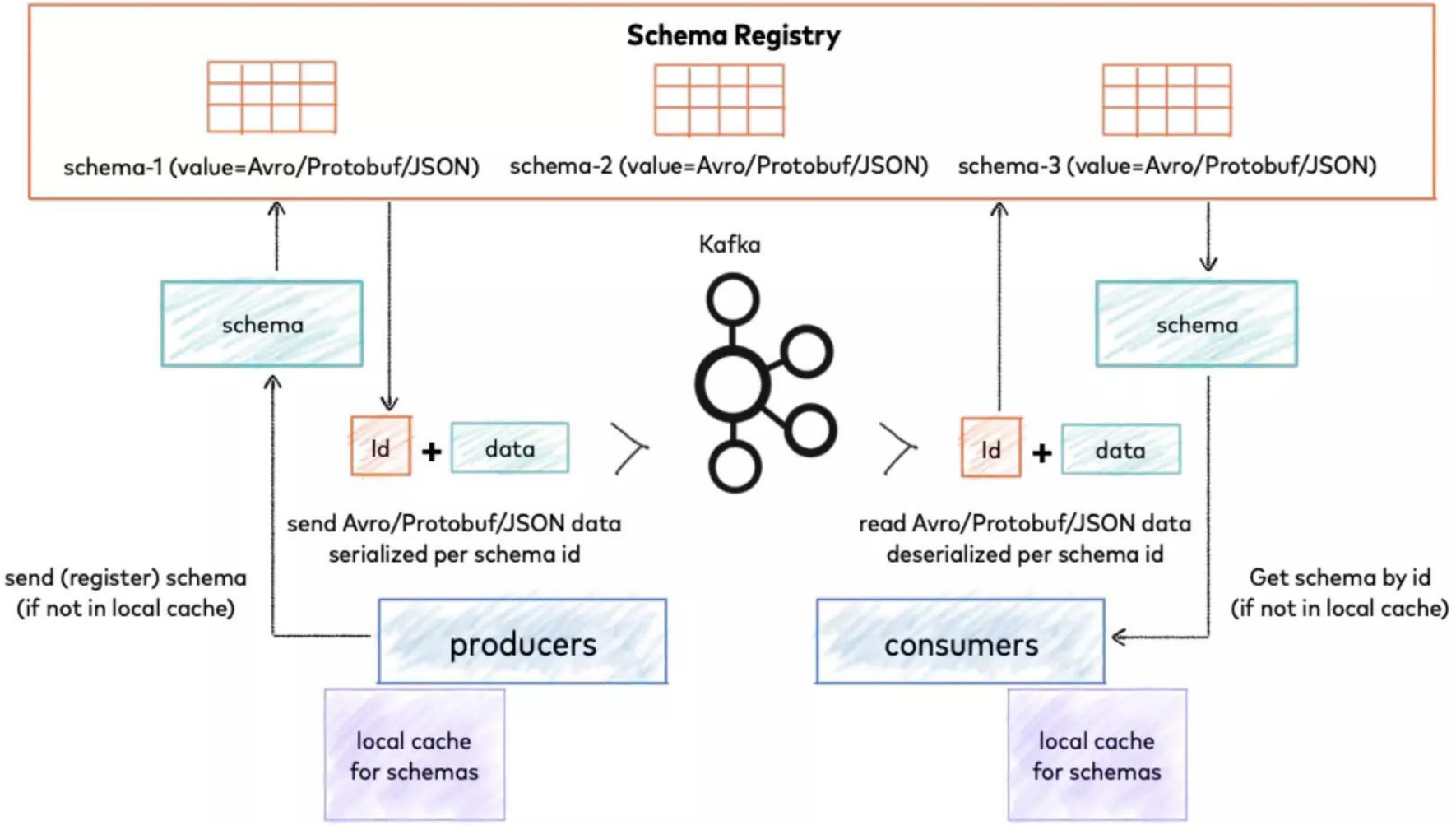
Consumer code breaks:
#new field Lname
Lname = humans[“Lname”]

KeyError: 'Lname'



Schema Registry Basics

Confluent Schema Registry



Schema Compatibility - Avro

Compatibility Type	Changes Allowed	Check against which schemas	Upgrade First
`BACKWARD`	• Delete Fields • Add optional fields	Last Version	Consumers
`BACKWARD_TRANSITIVE`	• Delete Fields • Add optional fields	All previous versions	Consumers
`FORWARD`	• Add Fields • Delete Optional Fields	Last Version	Producers
`FORWARD_TRANSITIVE`	• Add Fields • Delete Optional Fields	All previous versions	Producers
`FULL`	• Add Optional Fields • Delete Optional Fields	Last Version	Any order
`FULL_TRANSITIVE`	• Add Optional Fields • Delete Optional Fields	All previous versions	Any order
`NONE`	• All Changes Accepted	Compatibility checking disabled	Depends

Subject Naming Strategies

TopicNameStrategy

Subject Name =

Topic Name + [key|value]

Example:

Topic Name = mytopic

Value Subject Name = mytopic-value

Default

RecordNameStrategy

Subject Name =

Record Name + [key|value]

Example:

```
{"type":"record",
"name":"myrecord",
"fields":
      [{"name":"f1",
      "type":"string"}]
}
```

Value Subject Name = myrecord-value

Time Ordered Events

TopicRecordNameStrategy

Subject Name =

Topic Name + Record Name + [key|value]

Example:

Topic Name = mytopic

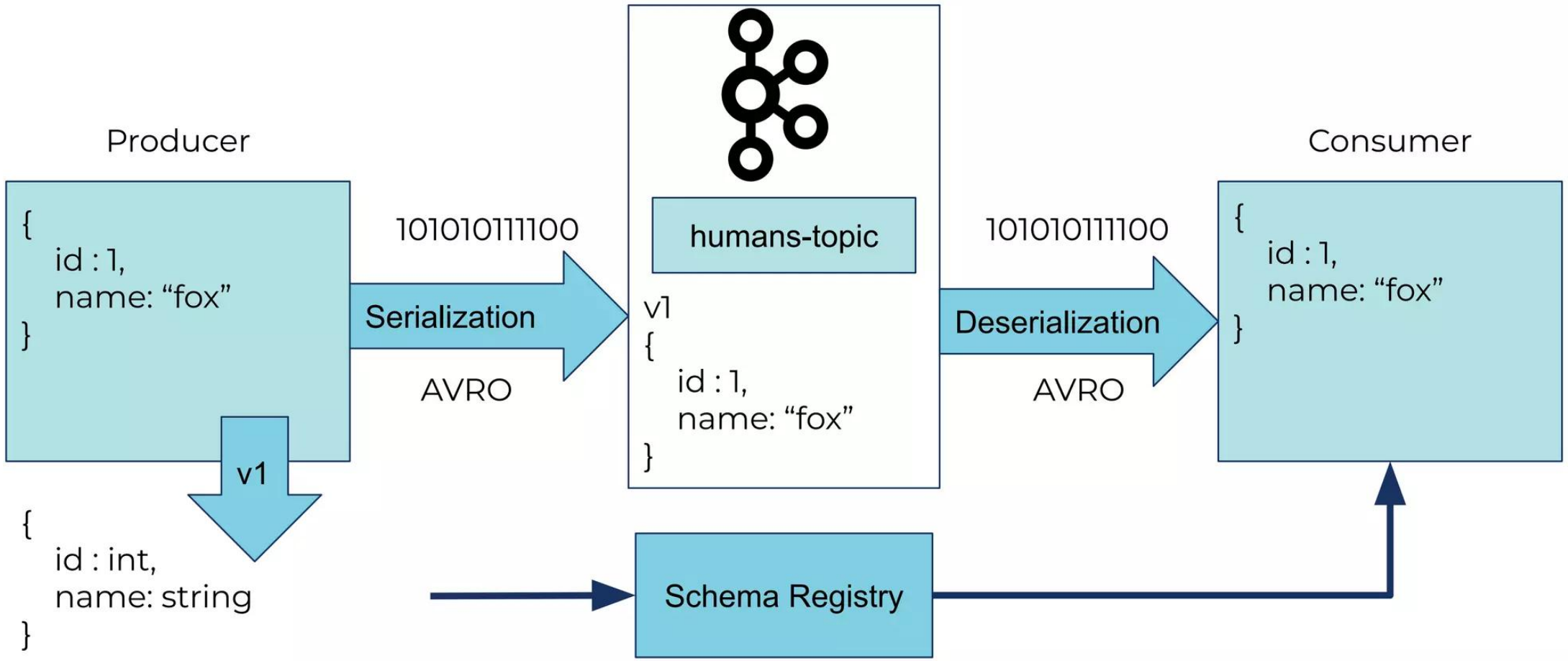
Record Name = myrecord

Value Subject Name = mytopic-myrecord-value

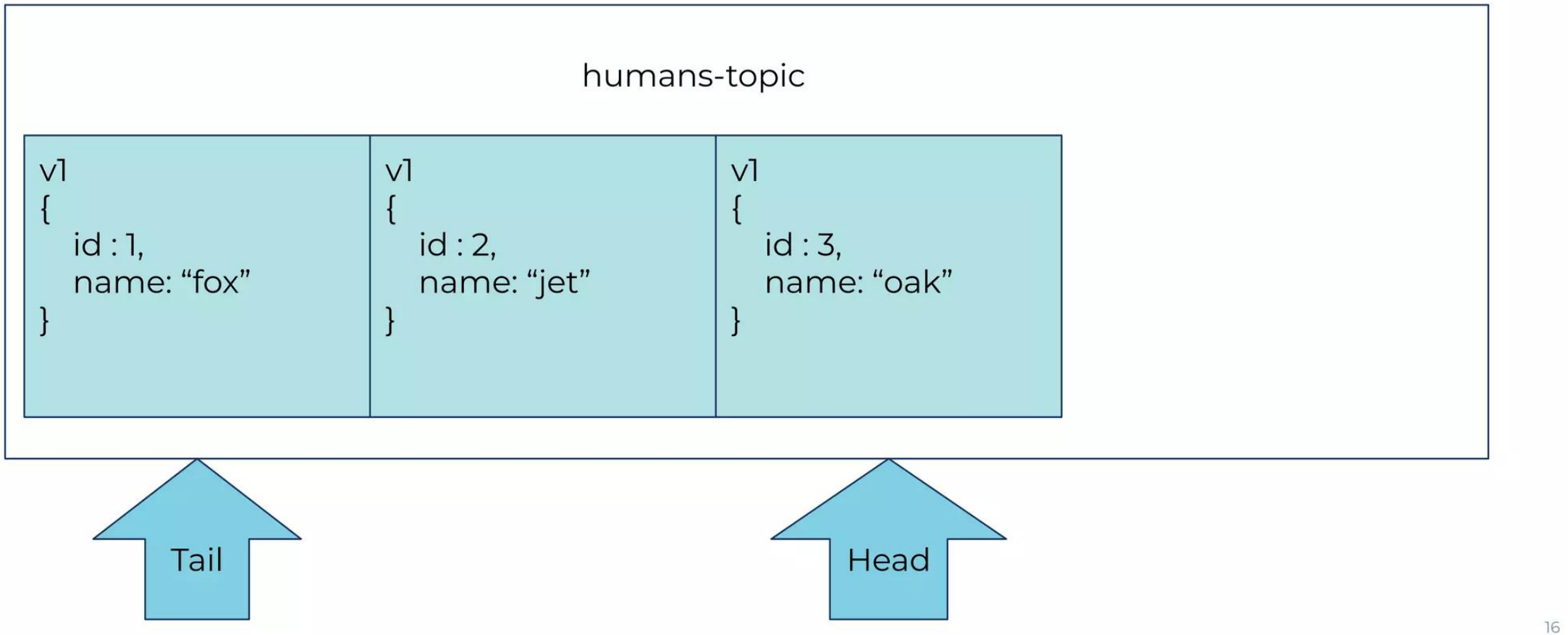
Most Granular

https://docs.confluent.io/current/schema-registry/serdes-develop/index.html#subject-name-strategy

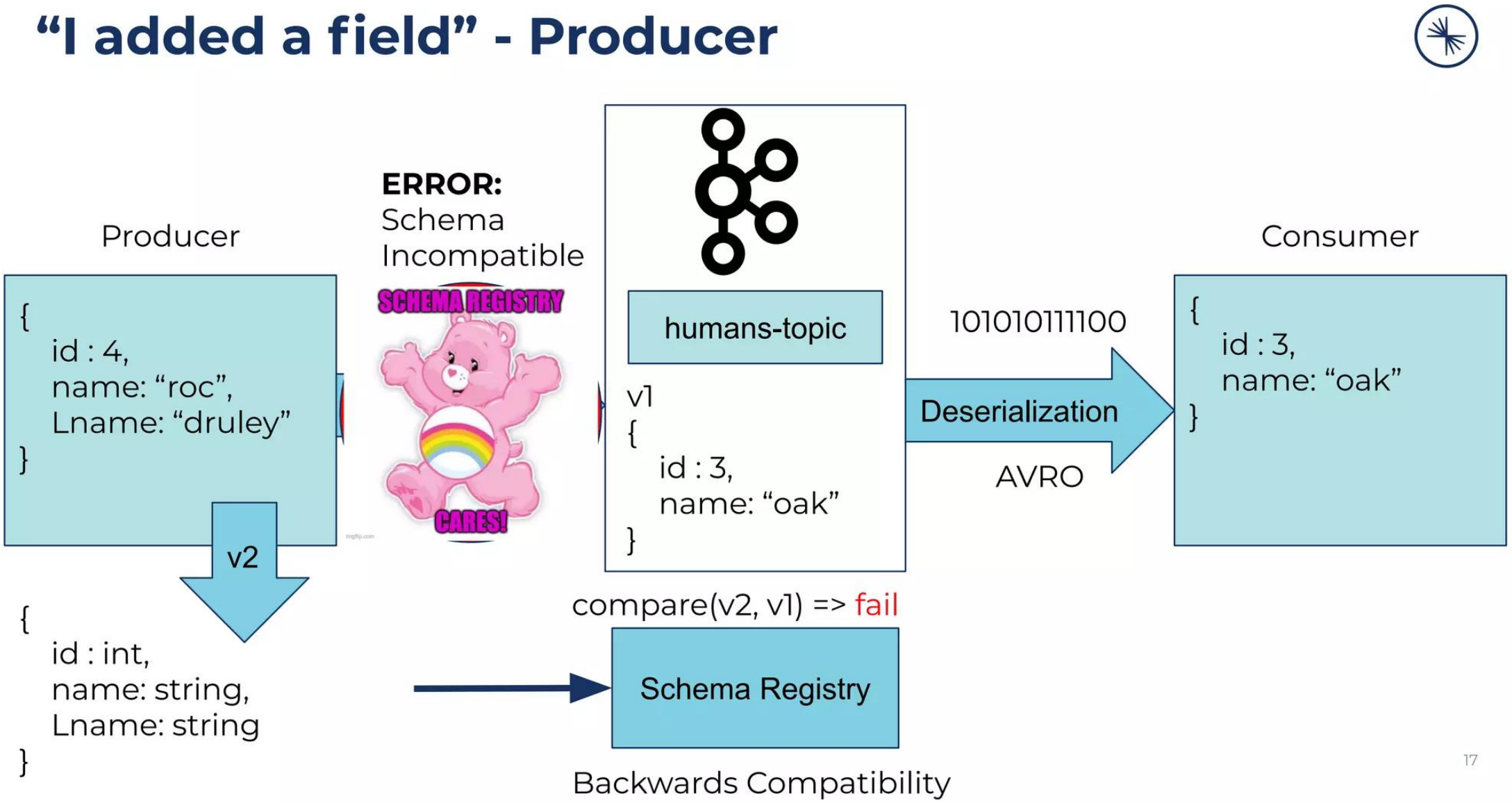
Kafka with Schema Registry



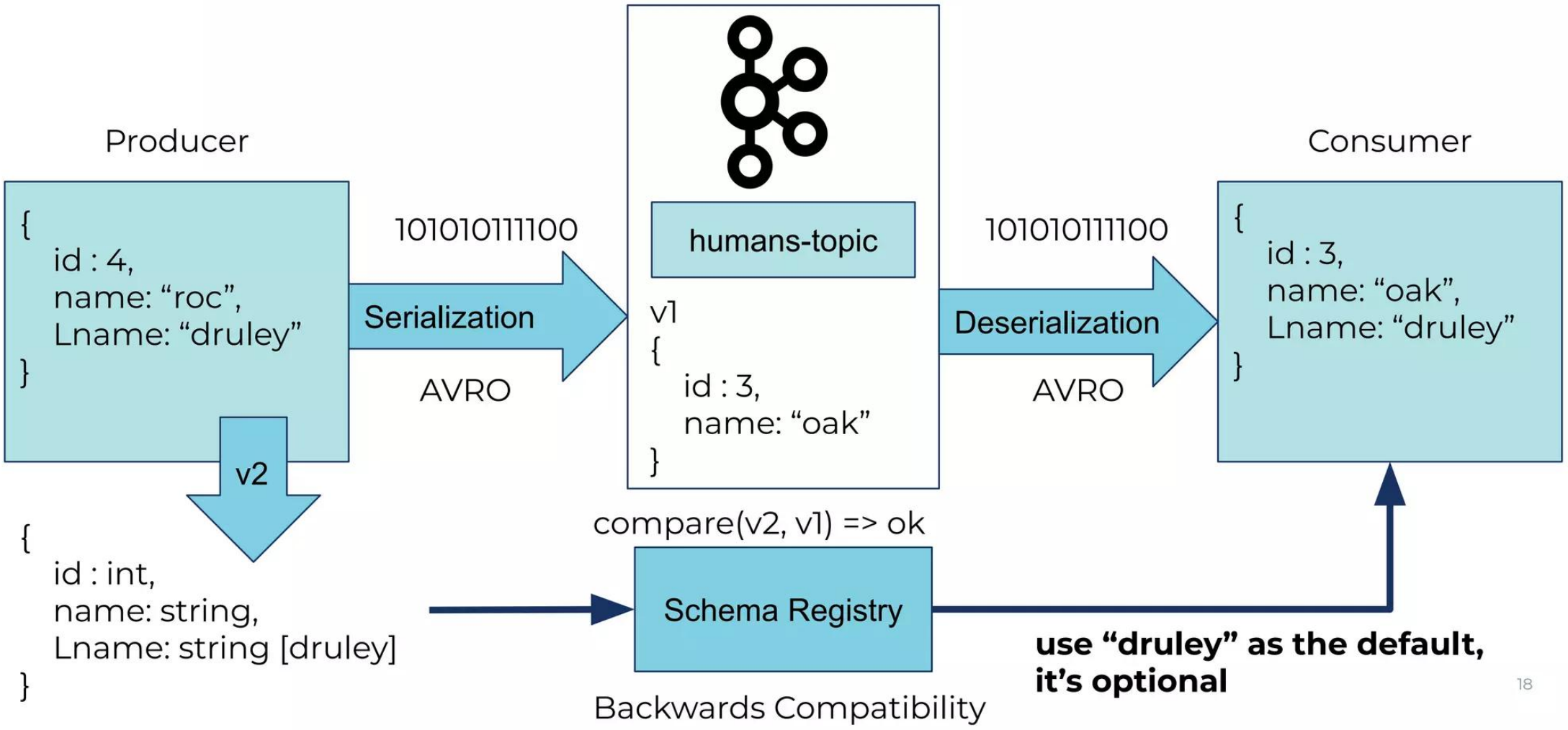
Time Passes, Again



“I added a field” - Producer



“I added an optional field” - Producer



3 Good Habits

1. Set auto register to false and add Schema Validation to CI/CD Pipeline.

Habit:

In prod and near-prod, clients should not automatically register new schemas. Ideally, this should be done in a CI/CD pipeline.

Producer Setting:

auto.register.schema = false

Exceptions:

1. Dev environments
2. Schema Registry ACLs are enabled using Confluent's security plugin https://docs.confluent.io/current/confluent-security-plugins/schema-registry/authorization/index.html#authorization-for-sr-operations-and-resources

Schema Registry Maven Plugin - pom.xml

```
    <plugin>
        <groupId>io.confluent</groupId>
        <artifactId>kafka-schema-registry-maven-plugin</artifactId>
        <version>${confluent.version}</version>
        <configuration>
            <schemaRegistryUrls>
                <param>${schemaRegistryUrl}</param>
            </schemaRegistryUrls>
            <userInfoConfig>${schemaRegistryBasicAuthUserInfo}</userInfoConfig>
            <subjects>

<transactions-value>src/main/resources/avro/io/confluent/examples/clients/basicavro/Payment2a.avsc</transactions-value>
            </subjects>
        </configuration>
        <goals>
            <goal>test-compatibility</goal>
        </goals>
    </plugin>
```

https://docs.confluent.io/current/schema-registry/schema_registry_onprem_tutorial.html#maven

2. Create new topics if you need to break compatibility.

Habit:

Figure out the right compatibility for you and create new topics in order to break it.

Don’t retrofit your compability just for one time major schema changes.

Exceptions:

-Dev

3. Use Confluent Schema Validation.

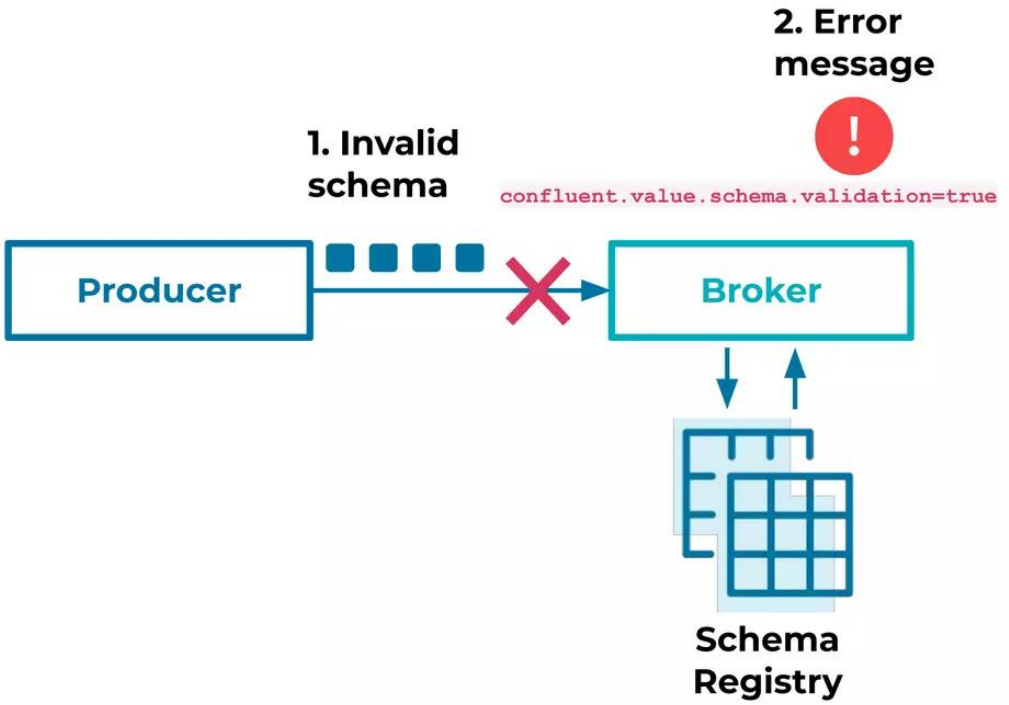
Scale schemas reliably

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

Granular control

- **Enabled validation** at the topic level
- Set **subject naming strategy** at the topic level

Broker Side Schema Validation



Schema Validation Demo

https://docs.confluent.io/current/schema-registry/schema-validation.html#sv

CONFLUENT

Project Metamorphosis

Unveiling the next-gen event streaming platform

For Updates Visit
cnfl.io/pm

Jay Kreps
Co-founder and CEO
Confluent

Q&A

https://www.linkedin.com/in/patrickdruley/

patrick@confluent.io

https://github.com/confluentinc/demo-scene/tree/master/industry-themes

cnfl.io/meetups **cnfl.io/blog** **cnfl.io/slack**