

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines and dashed blue lines. Small blue dots are placed at various points along these lines, and small blue 'x' marks are scattered across the background.

# Data Schemas

The background of the slide is a light blue map with various wavy lines, some solid and some dashed, and several small blue dots scattered across it. A white rectangular box with a blue border is centered on the slide, containing the title text.

# **Introduction to Data Schemas**

The background of the slide features a light blue map-like pattern. It includes several solid blue lines of varying thicknesses that curve across the frame. Interspersed among these are dashed blue lines, some of which end in small 'x' marks. There are also several small, solid blue dots scattered across the map. A white rectangular box with a thin blue border and a subtle drop shadow is centered on the slide, containing the title text.

# **Understanding Data Schemas**

# What is a Data Schema?

---

Data schemas set **expectations** about the shape and types of our data

```
{  
  "id": 123, int  
  "first": "ben", string  
  "last": "goldberg", string  
  "email": "ben@email.io", string  
  "phone": "1234567890", string  
  ...  
}
```

# What is a Data Schema?

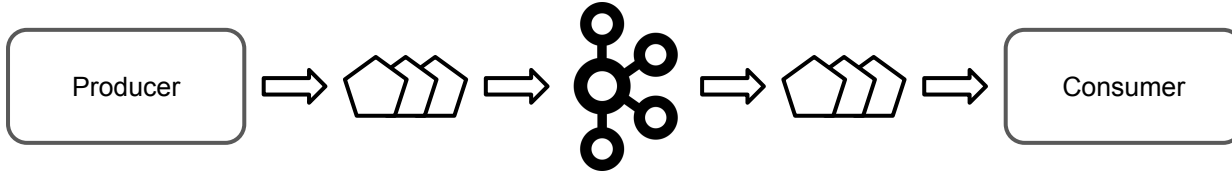
---

SQL Databases enforce schemas on tables

| <b>id</b> <span>int</span> | <b>first</b> <span>string</span> | <b>last</b> <span>string</span> | <b>email</b> <span>string</span> | <b>phone</b> <span>string</span> |
|----------------------------|----------------------------------|---------------------------------|----------------------------------|----------------------------------|
| 123                        | ben                              | goldberg                        | ben@email.io                     | 1234567890                       |
| ...                        |                                  |                                 |                                  |                                  |

# What is a Data Schema?

Schemas decrease coupling between applications



The background of the slide is a light blue map. It features several solid blue lines that curve across the frame, resembling topographical contour lines or perhaps stylized paths. Interspersed among these solid lines are dashed blue lines. Small, solid blue dots are placed at various points along both the solid and dashed lines. Additionally, there are small, light blue 'x' marks scattered across the map, some near the dashed lines and others in open areas. In the center of the slide, there is a white rectangular box with a thin blue border and a subtle drop shadow, containing the text 'Real-World Usage'.

## **Real-World Usage**

## Where are Data Schemas Used?

Declaring a table in Postgres or MySQL is an example of using a schema

```
CREATE TABLE store_location (
```

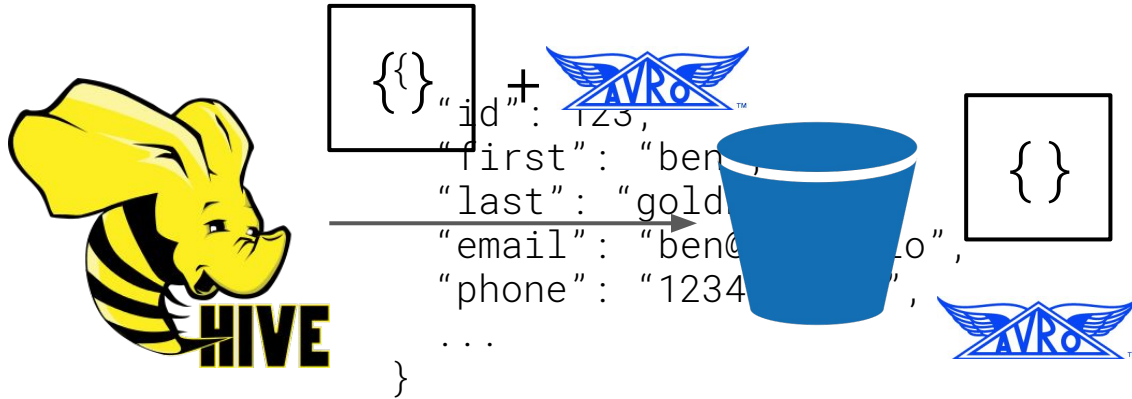
| id  | name          | city    | latitude | longitude |
|-----|---------------|---------|----------|-----------|
| 123 | cool_clothing | chicago | 67.14721 | 12.78431  |
| ... |               |         |          |           |

```
name VARCHAR(80),  
city VARCHAR(40),  
latitude NUMERIC(10),  
longitude NUMERIC(10)  
);
```



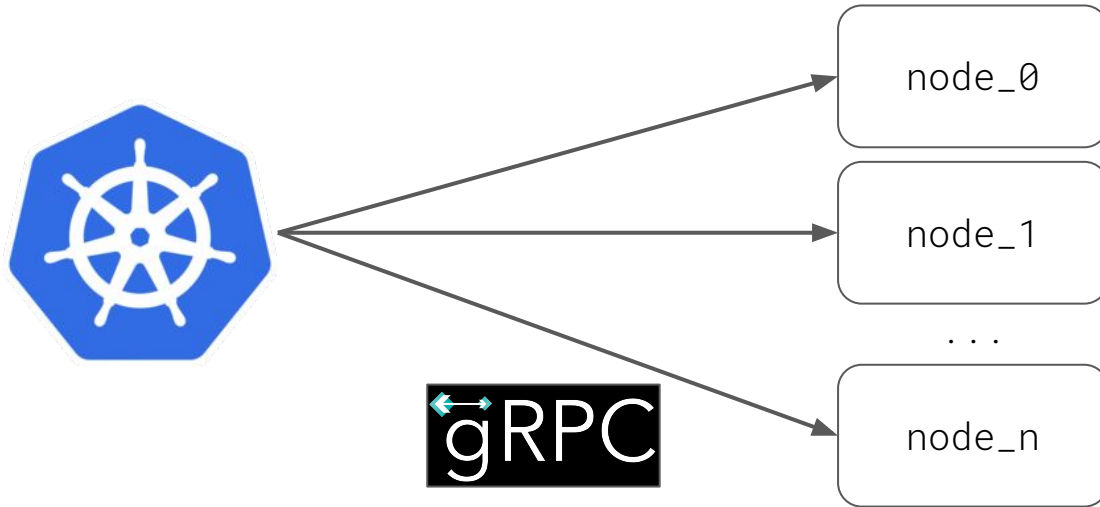
## Where are Data Schemas Used?

The Hadoop ecosystem uses defined schemas to load data



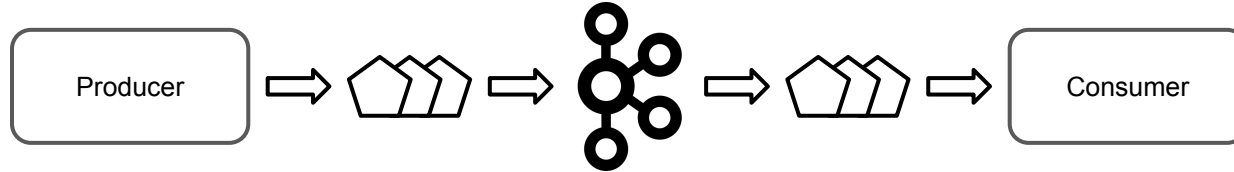
## Where are Data Schemas Used?

Kubernetes uses gRPC to communicate with system components



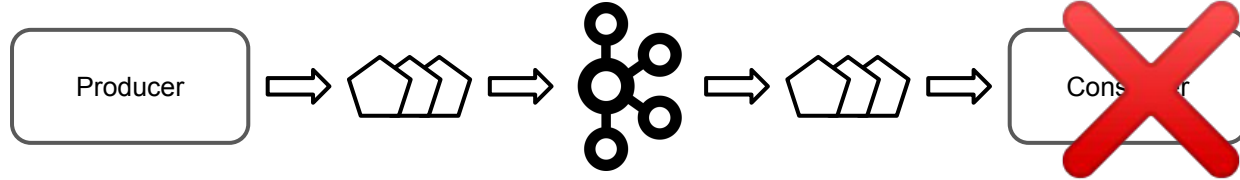
## Data Streaming Without Schemas

Scenario: A system is released with no schema. All goes well at first.



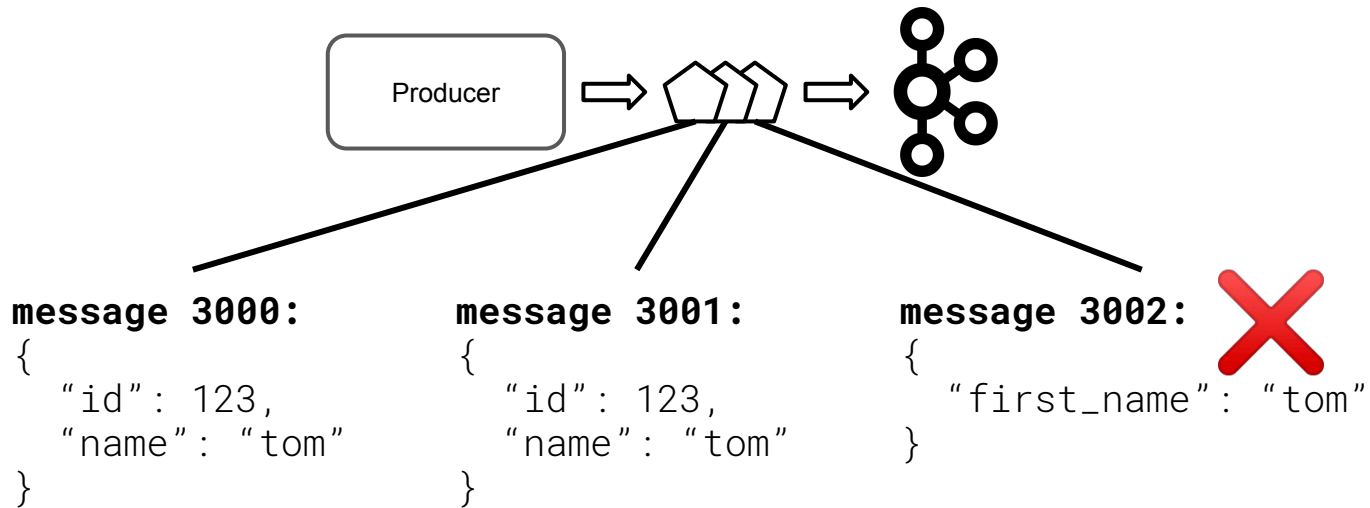
## Data Streaming Without Schemas

Scenario: A few weeks later, our consumer mysteriously dies!



# Data Streaming Without Schemas

Scenario: A renamed and missing field is crashing the consumer



The background of the slide is a light blue map with various wavy lines, some solid and some dashed, and several small blue dots scattered across it. A white rectangular box with a blue border is centered on the slide, containing the title text.

## **Demo: Data Streaming Without Schemas**

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these lines are small, solid blue dots and faint, dashed blue lines, some of which end in small 'x' marks, suggesting a network or a series of data points.

# **Data Streaming with Schemas**

# Data Streaming with Schemas

---

Why they matter

- Data streams are constantly evolving
- No schema = broken consumer on every data change
- Schemas allow consumers to function without updates
- Schemas provide independence and scalability
- Schemas can communicate version compatibility



The background of the slide features a light blue pattern of wavy lines, some solid and some dashed, with small blue dots scattered throughout, resembling a stylized map or topographical chart.

# Apache Avro

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these lines are small, solid blue dots and faint, dashed blue lines, some of which end in small 'x' marks, suggesting a technical or geographical theme.

# Apache Avro

# What is Apache Avro?


Avro is a data serialization system that uses binary compression



```
{
  "id": 123,
  "first": "ben",
  "last": "goldberg",
  "email": "ben@email.io",
  "phone": "1234567890",
  ...
}
```

≠

```
{
  "fields": [
    { "name": "id", "type": "int" },
    { "name": "first", "type": "string" },
    { "name": "last", "type": "string" },
    { "name": "email", "type": "string" },
    { "name": "phone", "type": "string" },
    ...
  ]
}
```

A diagram showing a box representing an Avro schema. The box has the Avro logo at the top. Inside the box, there are four lines of binary code: 01001010101, 01010101010, 10101010101, and 1111010. The box is positioned between the two JSON snippets, indicating the binary representation of the schema.

The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and several dashed blue lines that also curve. Small blue dots are placed at various points along these lines. Additionally, there are small 'x' marks scattered across the background, some near the dashed lines and others near the solid lines.

## **How Avro Schemas are Defined**

# How Avro Schemas are Defined

---

Avro schemas are defined as JSON records



```
{  
  "type": "record",  
  "name": "user",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": "string"},  
  ]  
}
```

# How Avro Schemas are Defined

---

The required **name** field identifies the Avro schema uniquely



```
{  
  "type": "record",  
  "name": "user",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": "string"},  
  ]  
}
```

# How Avro Schemas are Defined

The optional **namespace** field groups the Avro schema with others



```
{  
  "type": "record",  
  "name": "user",  
  "namespace": "com.udacity",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": "string"},  
  ]  
}
```

# How Avro Schemas are Defined

All Avro schemas have a **type** and the root type is always **record**



```
{  
  "type": "record",  
  "name": "user",  
  "namespace": "com.udacity",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": "string"},  
  ]  
}
```



# How Avro Schemas are Defined

All Avro records have **fields** that define expected data keys and types



```
{  
  "type": "record",  
  "name": "user",  
  "namespace": "com.udacity",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": "string"},  
  ]  
}
```

# How Avro Schemas are Defined

**Optional** fields may be **null** or another primitive type



```
{
  "type": "record",
  "name": "user",
  "namespace": "com.udacity",
  "fields": [
    {"name": "id", "type": "int"},
    {"name": "first", "type": "string"},
    {"name": "last", "type": "string"},
    {"name": "email", "type": "string"},
    {"name": "phone", "type": ["null", "string"]},
  ]
}
```

# How Avro Schemas are Defined

The below fields consist of **primitive** types, ex: null, string, int



```
{
  "type": "record",
  "name": "user",
  "namespace": "com.udacity",
  "fields": [
    {"name": "id", "type": "int"},
    {"name": "first", "type": "string"},
    {"name": "last", "type": "string"},
    {"name": "email", "type": "string"},
    {"name": "phone", "type": ["null", "string"]},
  ]
}
```

# How Avro Schemas are Defined

The **record** type is a **complex** type, ex: record, map, array



```
{  
  "type": "record",  
  "name": "user",  
  "namespace": "com.udacity",  
  "fields": [  
    {"name": "id", "type": "int"},  
    {"name": "first", "type": "string"},  
    {"name": "last", "type": "string"},  
    {"name": "email", "type": "string"},  
    {"name": "phone", "type": ["null", "string"]},  
  ]  
}
```

The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and several dashed blue lines, some of which end in small 'x' marks. There are also several solid blue dots scattered across the map. A white rectangular box with a thin blue border is centered on the slide, containing the title text.

# **Defining your first Avro Record Demonstration**

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these lines are small, solid blue dots and small 'x' marks, some of which are connected by short dashed lines, suggesting a path or a series of locations on a map.

## **Demo: Defining your first Avro Record**

The background of the slide features a light blue map-like pattern. It includes several solid blue lines of varying thicknesses that curve across the frame. Interspersed among these are dashed blue lines, some of which end in small 'x' marks. There are also several small, solid blue dots scattered throughout the background. A white rectangular box with a thin blue border is centered in the middle of the slide, containing the title text.

# **Apache Avro Data Types**

# Avro Data Types

---

## Primitive Types

- `null`
- `boolean` (`true` / `false`)
- `int`, `long`, `float`, `double` (`1` / `123.37`)
- `bytes` (`b'AE002448FF'`)
- `string` (`"hello world"`)



# Avro Data Types

---

## Complex Types

- record
- enum
- array
- map
- union
- fixed

# Avro Data Types

---

Enumerations are a set of named symbols

```
{  
  "type": "enum",  
  "name": "us_states",  
  "symbols": ["AL", "AK", "AZ", "AR", "CA", ...]  
}
```

# Avro Data Types

Arrays store ordered fields of **primitive** or **complex** types

## Primitive

```
{
  "type": "array",
  "items": "string",
}
```

## Complex

```
{
  "type": "array",
  "items": {
    "type": "record",
    "fields": [
      {"name": "id", "type": "int"}
    ]
  }
}
```

# Avro Data Types

Maps store fields as a **string key** to **value** of **primitive** or **complex** type

## Primitive

```
{  
  "type": "map",  
  "values": "int",  
}
```

## Complex

```
{  
  "type": "map",  
  "values": {  
    "type": "record",  
    "fields": [  
      {"name": "id", "type": "int"}  
    ]  
  }  
}
```

# Avro Data Types

---

**Unions** denote that more than one type may be used.

```
{
  "type": "map",
  "values": {
    "type": "record",
    "fields": [
      {"name": "zipcode", "type": ["null", "int", "string"]}
    ]
  }
}
```

# Avro Data Types

---

**Fixed** denotes a fixed size entry in **bytes**

```
{  
  "name": "md5",  
  "type": "fixed",  
  "size": 16  
}
```

The background of the slide is a light blue map with various wavy lines, some solid and some dashed, and several small blue dots scattered across it. A white rectangular box with a blue border is centered on the slide, containing the title text.

## **Complex Records in Apache Avro**

The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and dashed blue lines that also curve. Small blue dots are placed at various points along these lines. Additionally, there are small 'x' marks scattered across the background, some of which are also connected by dashed lines.

## **Demo: Defining a Complex Avro Record**



The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these lines are small, solid blue dots and faint, dashed blue lines, some of which end in small 'x' marks, suggesting a network or a series of points of interest.

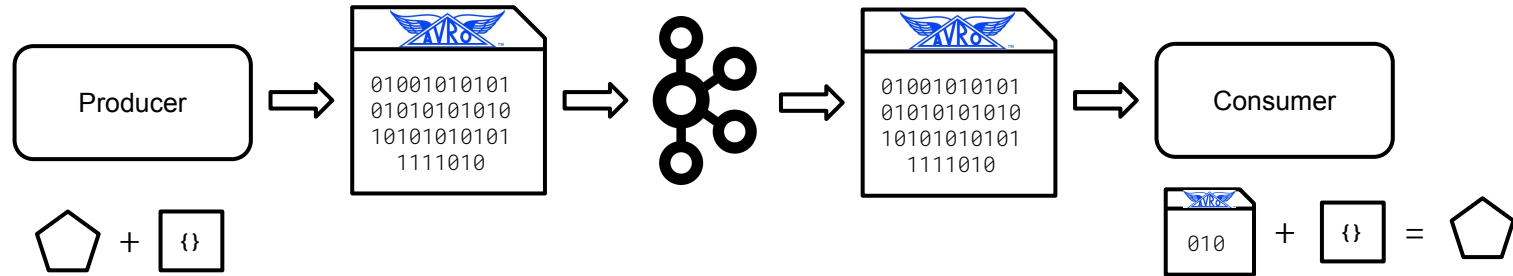
# **Apache Avro and Kafka**

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these lines are small, light blue 'x' marks and solid blue dots. Some of the 'x' marks are connected to the lines by short, dashed blue segments. The overall aesthetic is clean and modern, resembling a technical or geographical map.

# **Apache Avro and Kafka**

# Producing and Consuming Kafka Data with Apache Avro

The Producer must define an **Avro** schema and **encode the data**

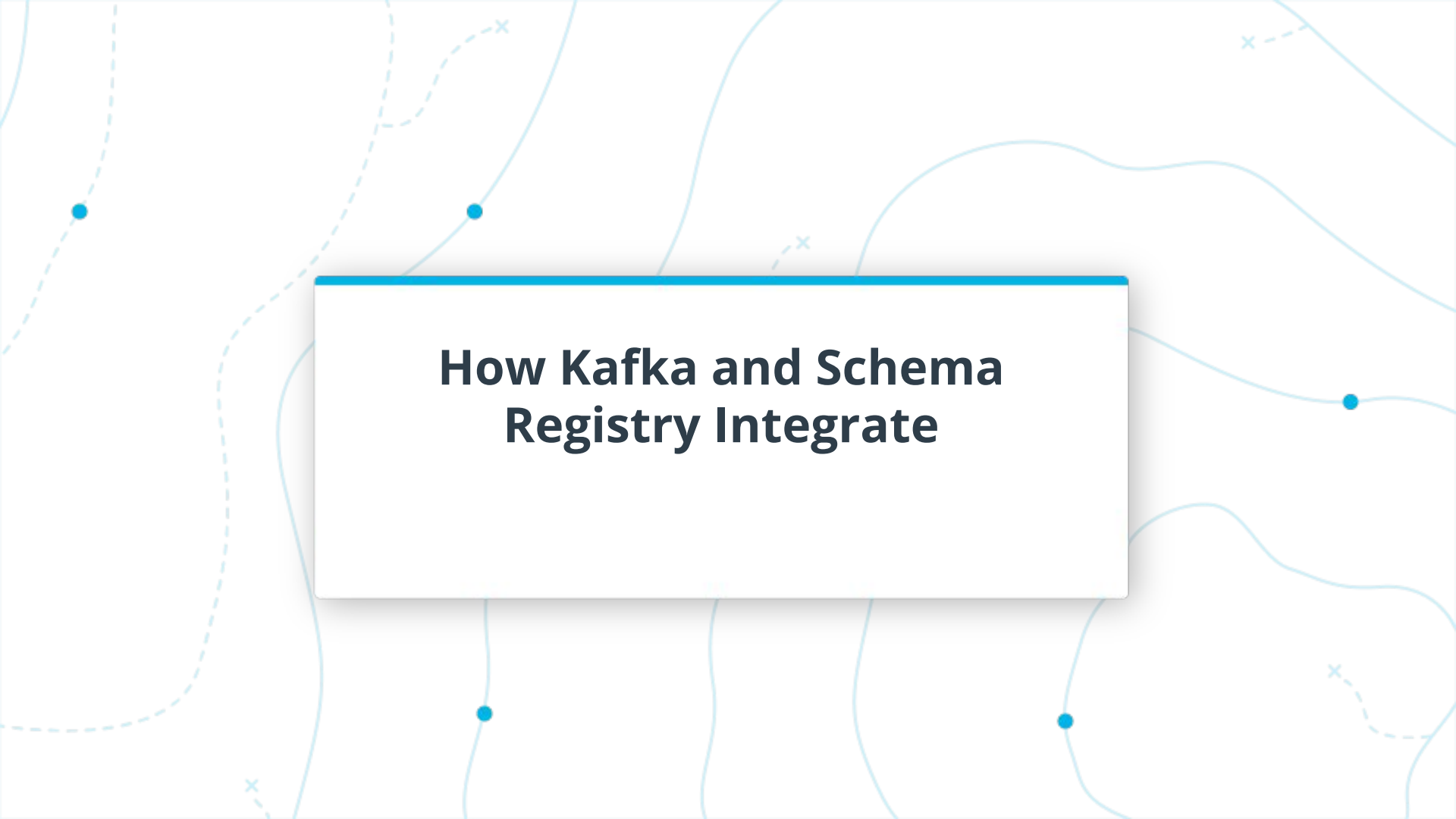


The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines and dashed blue lines. Small blue dots are placed at various points along these lines, and small blue 'x' marks are scattered across the background.

# Schema Registry

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines and dashed blue lines. Small blue dots are placed at various points along these lines, and small blue 'x' marks are scattered across the background.

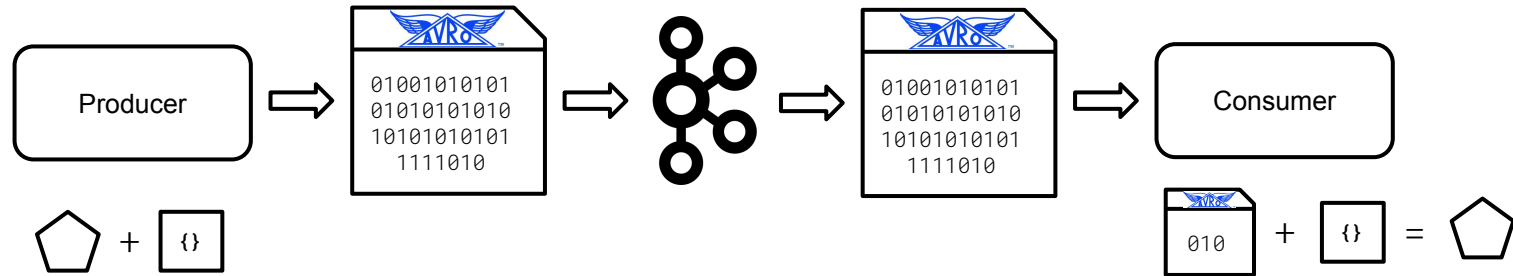
# Schema Registry

The background of the slide is a light blue map with various wavy lines, some solid and some dashed, and several small blue dots scattered across it. A white rectangular box with a blue border is centered on the slide, containing the title text.

# **How Kafka and Schema Registry Integrate**

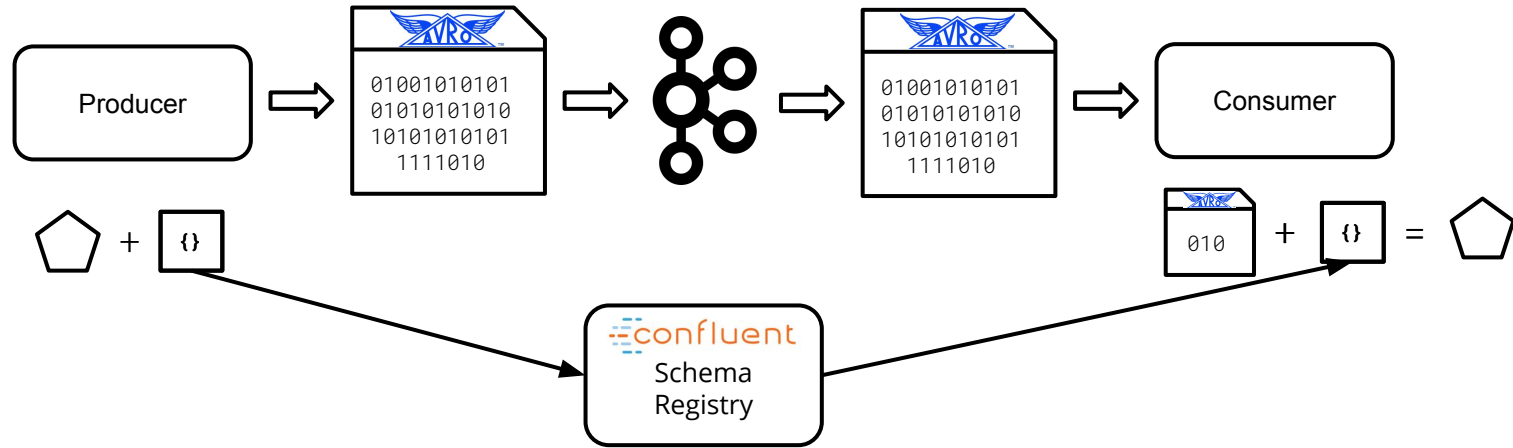
# Producing and Consuming Data with Schema Registry

Sending a schema definition with every message **adds overhead**



# Producing and Consuming Data with Schema Registry

Sending a schema definition with every message **adds overhead**





# Key Points

## Schema Registry



- Schema Registry stores state in Kafka itself
- Schemas only need to be sent to Schema Registry once
- Clients fetch schemas as needed from the registry
- Does not support deletes
- Has an HTTP REST Interface
- May use with any application, not just Kafka apps!

# Architecture

---

## Schema Registry



- Built in Scala and Java, runs on the JVM
- High portable, runs on nearly all OSes
- Stores all of its state in Kafka topics, not a database
- Exposes an HTTP web-server with a REST API
- Can run standalone or clustered with many nodes
- Uses ZooKeeper to choose leader in cluster mode

The background of the slide features a light blue map-like pattern. It includes several solid blue lines of varying thicknesses that curve across the frame. Interspersed among these are dashed blue lines, some of which end in small 'x' marks. There are also several small, solid blue dots scattered throughout the background. A white rectangular box with a thin blue border is centered on the slide, containing the title text.

# **Integrating Schema Registry**

The background of the slide features a light blue map-like pattern. It includes several solid blue lines of varying thicknesses that curve across the frame. Interspersed among these are dashed blue lines, some of which end in small 'x' marks. Additionally, there are several small, solid blue dots scattered throughout the background. A white rectangular box with a thin blue border is centered on the slide, containing the main title text.

## **Demo: Producing and Consuming Data with Schema Registry**

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these solid lines are dashed blue lines, some of which terminate in small 'x' marks. Additionally, there are several small, solid blue dots scattered throughout the map. A white rectangular box with a thin blue border is positioned in the center of the slide, containing the title text.

# **Schema Evolution and Compatibility**

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these solid lines are dashed blue lines, some of which terminate in small 'x' marks. Additionally, there are several small, solid blue dots scattered throughout the map. A white rectangular box with a thin blue border is positioned in the center of the slide, containing the title text.

# **Schema Evolution and Compatibility**

The background of the slide features a light blue map-like pattern. It includes several solid blue lines of varying thicknesses that curve across the frame. Interspersed among these are dashed blue lines, some of which end in small 'x' marks. There are also several solid blue dots scattered across the map. A white rectangular box with a thin blue border and a subtle drop shadow is centered on the slide, containing the title text.

# **Understanding Schema Evolution**

## Schema Evolution

---

The process of changing the schema of a given dataset is referred to as **schema evolution**. Modifying, adding, or removing a field are all forms of schema evolution.

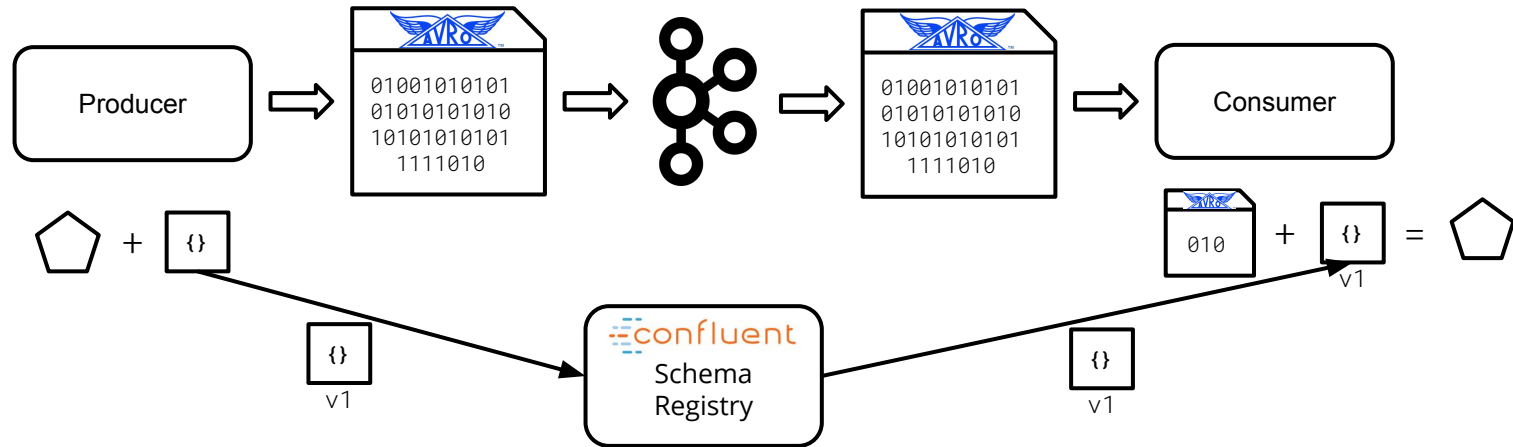


The background of the slide is a light blue map with various wavy lines, some solid and some dashed, and several small blue dots scattered across it. A white rectangular box with a blue border is centered on the slide, containing the text.

**What is Schema Compatibility?**

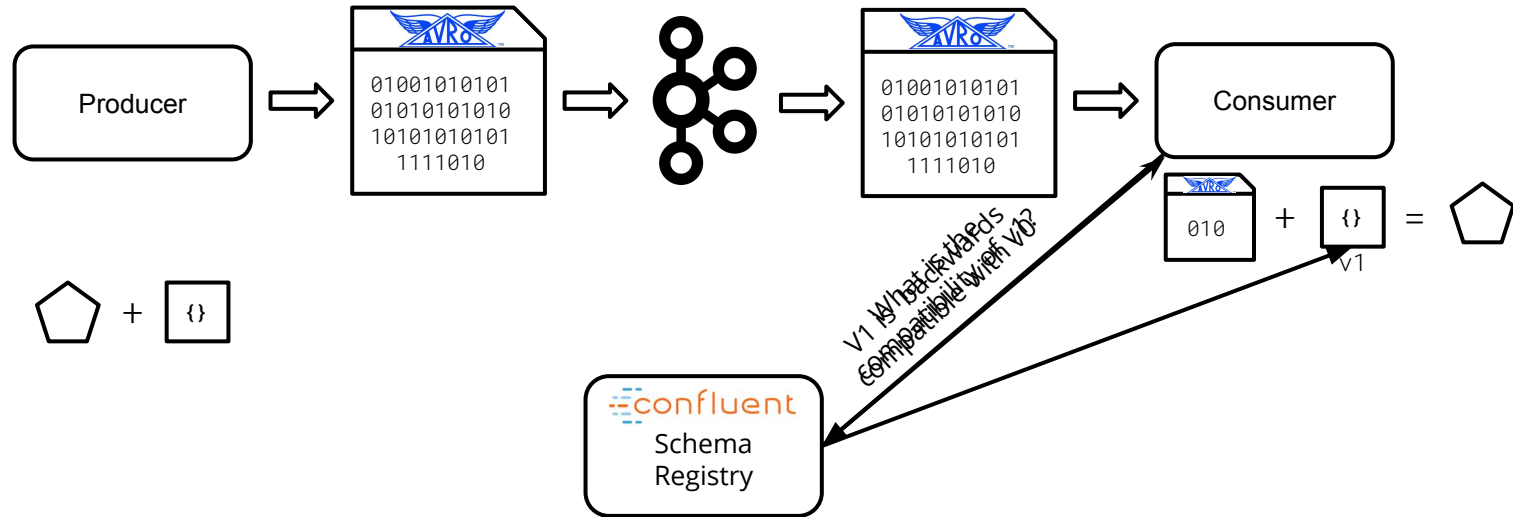
# Schema Compatibility

Schema Registry **tracks compatibility** between schema versions



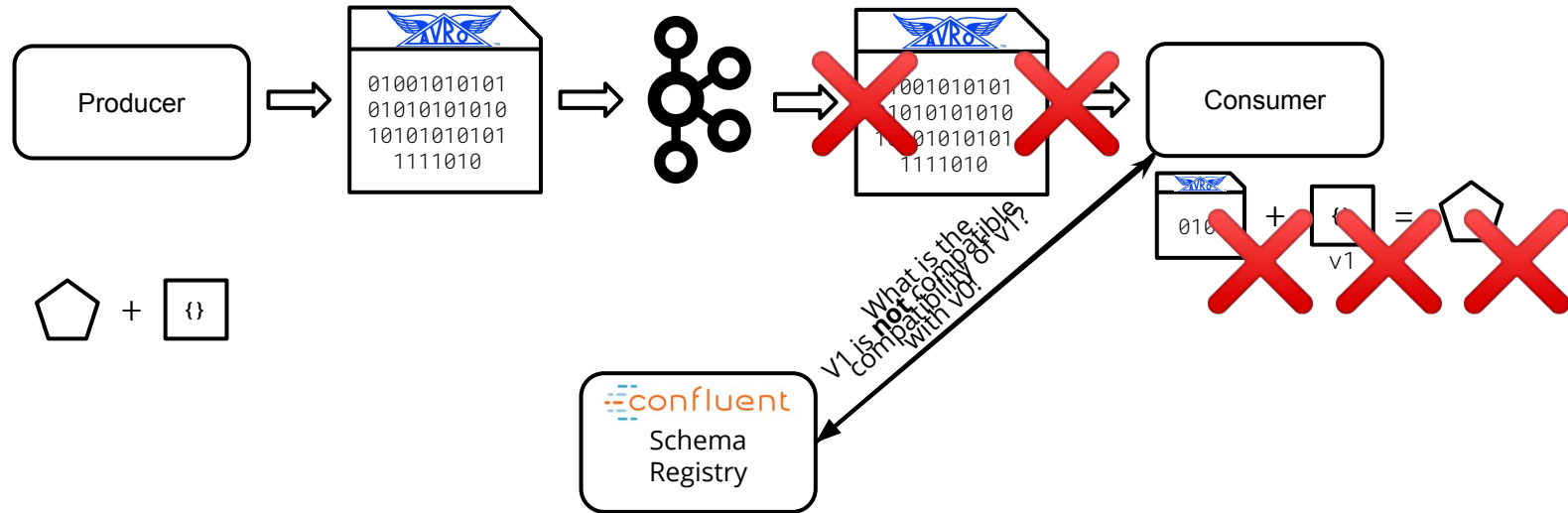
# Schema Compatibility

If the schema is compatible, the consumer continues consumption



# Schema Compatibility

If the schema is incompatible, the consumer will cease consumption



The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and several dashed blue lines, some of which end in small 'x' marks. There are also five solid blue dots scattered across the map. A white rectangular box with a thin blue border is centered on the slide, containing the text 'Backward Compatibility'.

# **Backward Compatibility**

# Backward Compatibility

Consumers developed against the **latest** schema can use **older** data

- Addition of **optional** fields or the **deletion** of a field in the latest schema are backward compatible changes

```
{
  "type": "record",
  "name": "purchase",
  "fields": [
    {"name": "username", "type": "string"},
    ["name": "amount", "type": "floating"],
    {"name": "amount", "type": "float"},
    {"name": "email", "type": "string"},
  ]
}
```

The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and several dashed blue lines, some of which end in small 'x' marks. There are also five solid blue dots scattered across the map. A white rectangular box with a thin blue border is centered on the slide, containing the text 'Forward Compatibility'.

# **Forward Compatibility**

# Forward Compatibility

Consumers developed against the **previous** schema can use the **latest**

- Addition of **new** fields or the **deletion** of a **optional** fields in the new schema are forward compatible changes

```
{
  "type": "record",
  "name": "purchase",
  "fields": [
    { "name": "username", "type": "string" },
    { "name": "amount", "type": "float" },
    { "name": "email", "type": "string" },
    { "name": "memo", "type": [ "null", "string" ] },
  ]
}
[ { "name": "area_code", "type": "string" } ]
}
```



The background of the slide features a light blue map-like pattern. It includes several solid blue lines that curve across the frame, and several dashed blue lines, some of which end in small 'x' marks. There are also five solid blue dots scattered across the map. A white rectangular box with a thin blue border is centered on the slide, containing the text 'Full Compatibility'.

# Full Compatibility

# Full Compatibility

---

The change is both **backward** and **forward** compatible

- Changing the default for a field is fully compatible

```
{
  "type": "record",
  "name": "purchase",
  "fields": [
    {"name": "username", "type": "string"},
    {"name": "amount", "type": "float"},
    {"name": "email", "type": "string"},
    {"name": "memo", "type": ["null", "string"]},
    {"name": "area_code", "type": "string"},
    {"name": "action", "type": "string", default: "pending_payment"}
  ]
}
```

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines and dashed blue lines. Small blue dots are placed at various points along these lines, and small blue 'x' marks are scattered across the background.

**No Compatibility**

# None Compatibility

---

None compatibility indicates that compatibility is not tracked

- **Do not use** None compatibility!
- If your schema has changed in a breaking fashion, **always create a new topic** and update your consumers to use that topic
- None does *not* indicate a breaking change, it is more akin to “unknown” since Schema Registry no longer tracks the compatibility

The background of the slide features a light blue map-like pattern. It consists of several wavy, solid blue lines that meander across the frame. Interspersed among these solid lines are dashed blue lines, some of which terminate in small 'x' marks. Additionally, there are five solid blue dots scattered across the map: one in the upper left, one in the upper middle, one in the lower middle, one in the lower right, and one on the right side.

# Summary