

# Using SHACL as an ontology modeling language to implement data-driven infrastructures

Ian Fleming  
August 12, 2021

# Agenda

**RDF and SHACL**

---

**Polyglot data infrastructure**

---

**Schema deployment**

---

**Example**

---

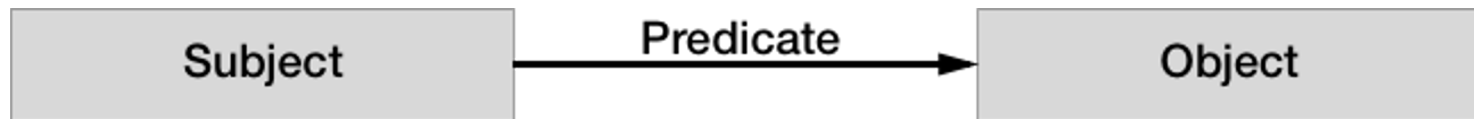
**Additional Notes and Considerations**

---

# Resource Description Framework

## RDF

- a model for data interchange on the Web
- A W3C standard for knowledge modeling and information exchange
- Based on http URIs
- Strong semantics
- Triples



- Most often viewed as a directed graph
- <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

# Hello, world!

## A very simple example (ttl serialization)

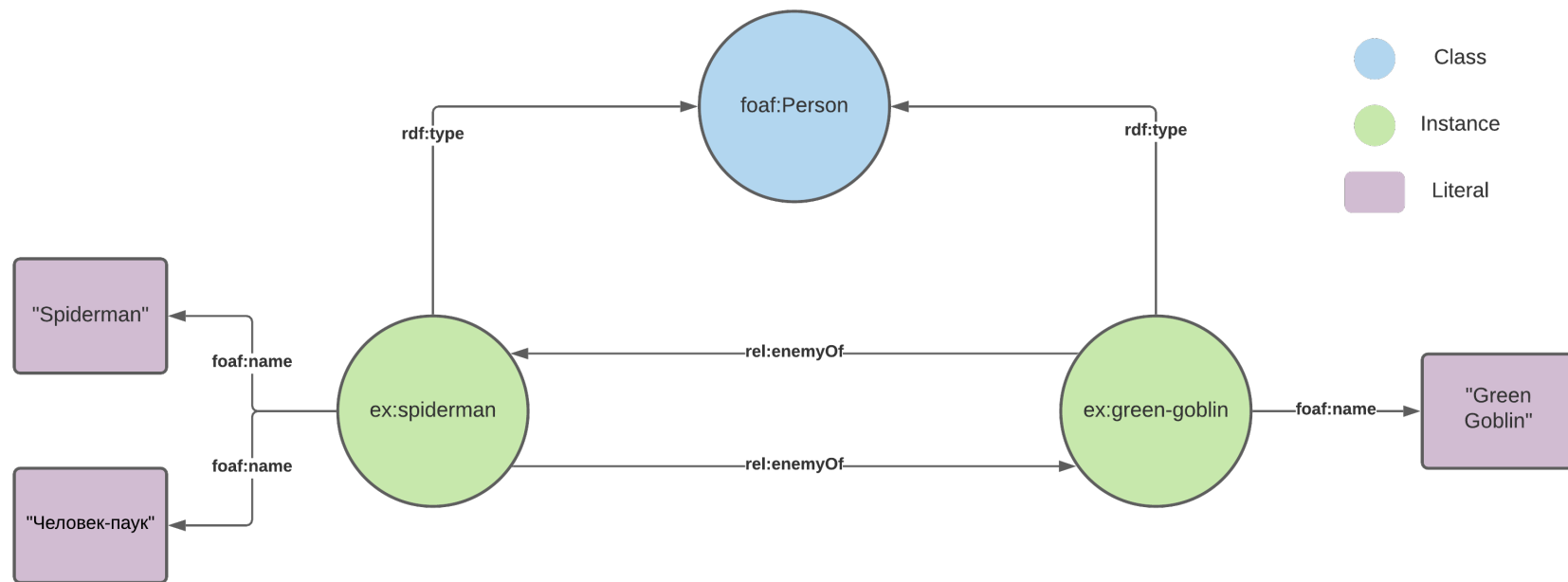
```
@prefix ex: <http://example.org/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rel: <http://www.perceive.net/schemas/relationship/> .
```

```
ex:spiderman a foaf:Person ;  
    foaf:name "Spiderman"@en ;  
    foaf:name "Человек-паук"@ru ;  
    rel:enemyOf ex:green-goblin .
```

```
ex:green-goblin a foaf:Person ;  
    foaf:name "Green Goblin"@en ;  
    rel:enemyOf ex:spiderman .
```

# Hello, world!

A very simple example (directed graph representation)



# Shapes Constraints Language

## SHACL

- a language for constructing and validating RDF graphs against a set of conditions
- A W3C standard for modeling schematic constraints in RDF
- Types of constraints (not exhaustive)
  - Predicates
  - Cardinality
  - Datatypes
  - Values
- <http://www.w3.org/ns/shacl#>

# Hello, world!

## A very simple SHACL shape

```
mds:Person a sh:NodeShape ;  
  rdfs:label "Person"@en ;  
  skos:definition "A human being"@en ;  
  sh:targetClass foaf:Person ;  
  sh:property [  
    sh:path schema:givenName ;  
    sh:datatype xsd:string ;  
    sh:minCount 1 ;  
    sh:maxCount 1 ;  
  ] ;  
  sh:property [  
    sh:path schema:familyName ;  
    sh:datatype xsd:string ;  
    sh:minCount 1 ;  
  ] ;  
  sh:property [  
    sh:path schema:height ;  
    sh:datatype xsd:float ;  
    sh:minExclusive 0 ;  
  ] .
```

# Hello, world!

## Some RDF data

```
ex:Ian a foaf:Person ;  
    schema:givenName "Ian" ;  
    schema:familyName "Fleming" ;  
    schema:height 70 .  
  
ex:Lydia a foaf:Person ;  
    schema:givenName "Catherine" ;  
    schema:additionalName "Lydia" ;  
    schema:familyName "White" ;  
    ex:preferredName "Lydia White" .  
  
ex:Ian ex:spouse ex:Lydia .
```

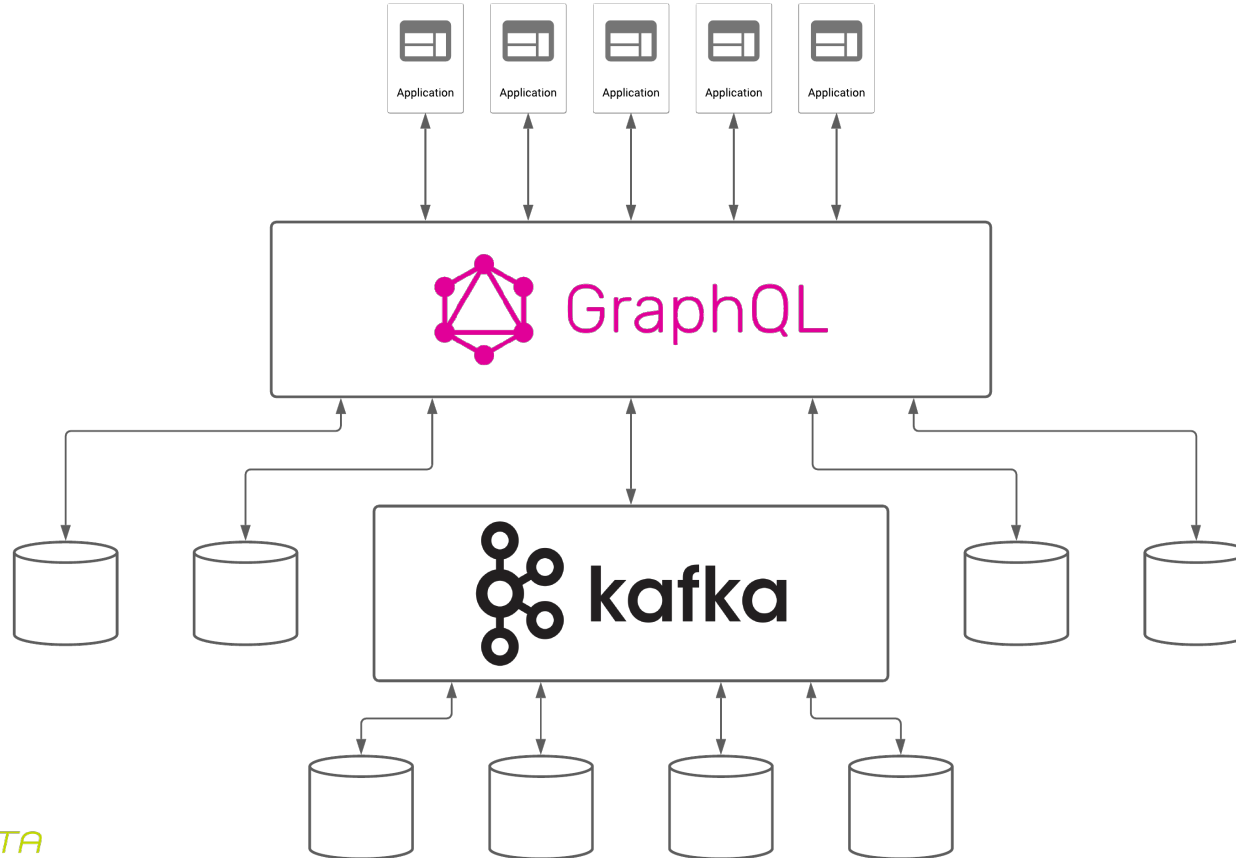


# A clinical example

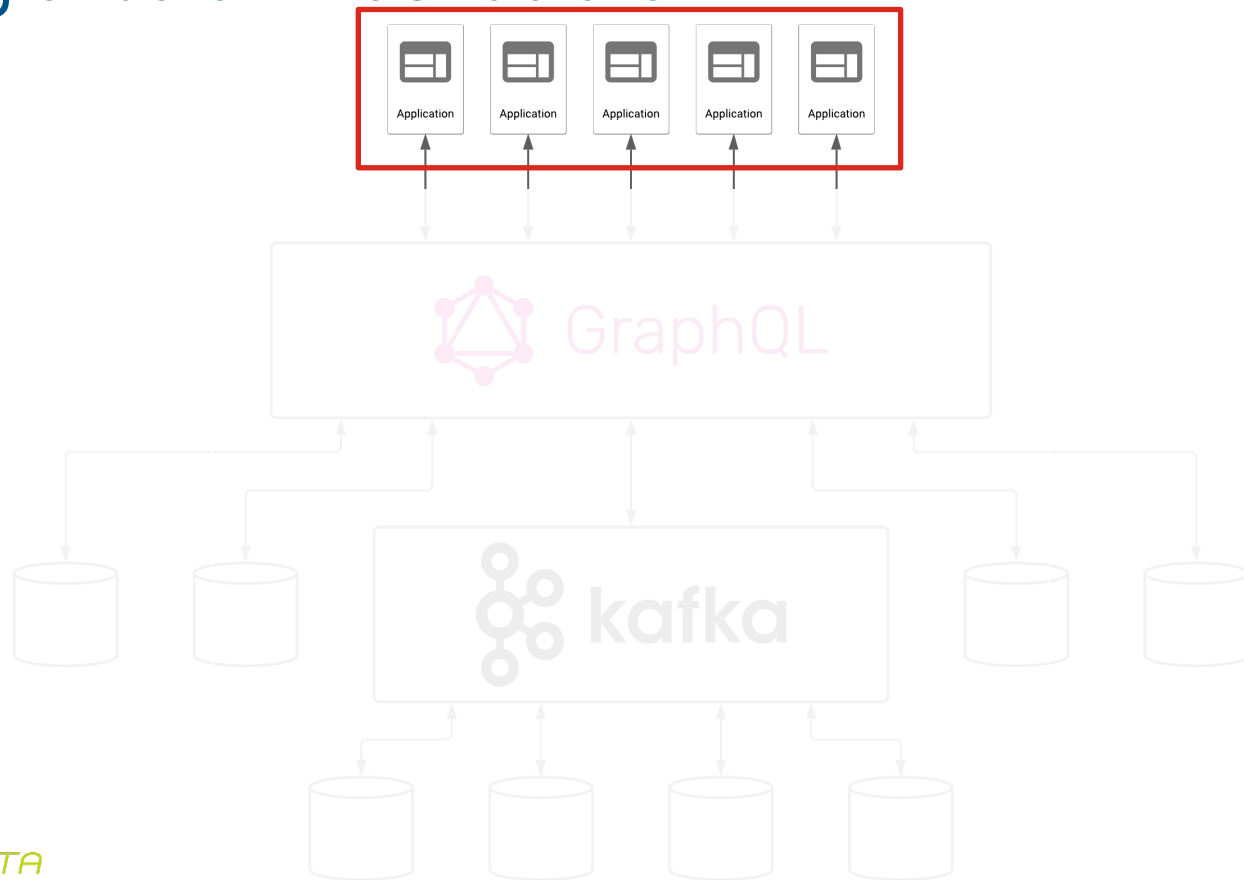
```
mds:ClinicalStudy a sh:NodeShape ;
  rdfs:label "Clinical Study" ;
  skos:definition "Research conducted with human subjects or on material of human
    origin in which an investigator directly interacts with human
    subjects; includes development of new technologies, study of
    mechanisms of human diseases, therapy, clinical trials,
    epidemiologic, behavior, and health services research." ;

sh:node mds:StudyShape ;
sh:property [
  sh:path mdo:targets ;
  sh:class ncit:C41184 ;
  sh:minCount 1 ;
] ;
sh:property [
  sh:path mdo:evaluates ;
  sh:class ncit:C25218 ;
  sh:minCount 1 ;
] ;
sh:property [
  sh:path mdo:sponsor;
  sh:node mds:Organization ;
  sh:minCount 1 ;
  sh:maxCount 1 ;
]
```

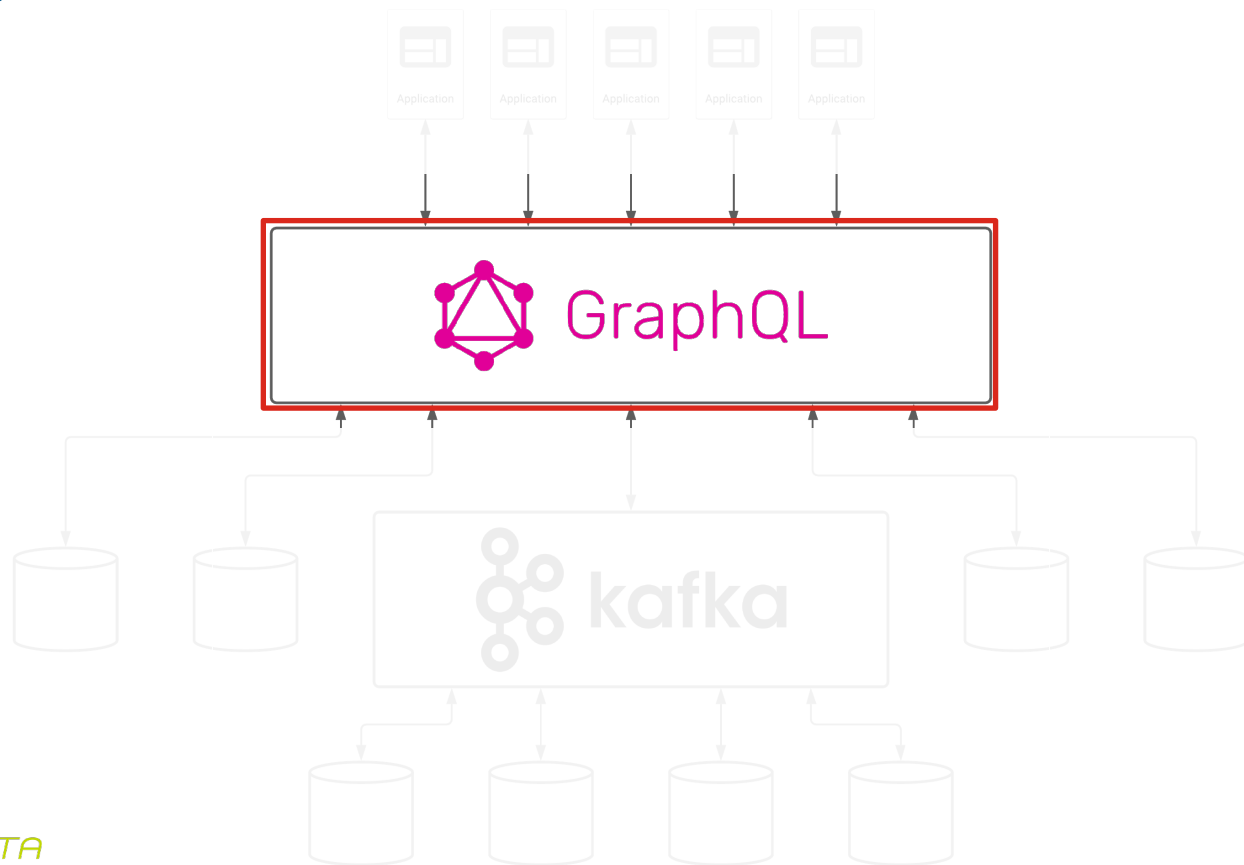
# Polyglot data infrastructure (data flow)



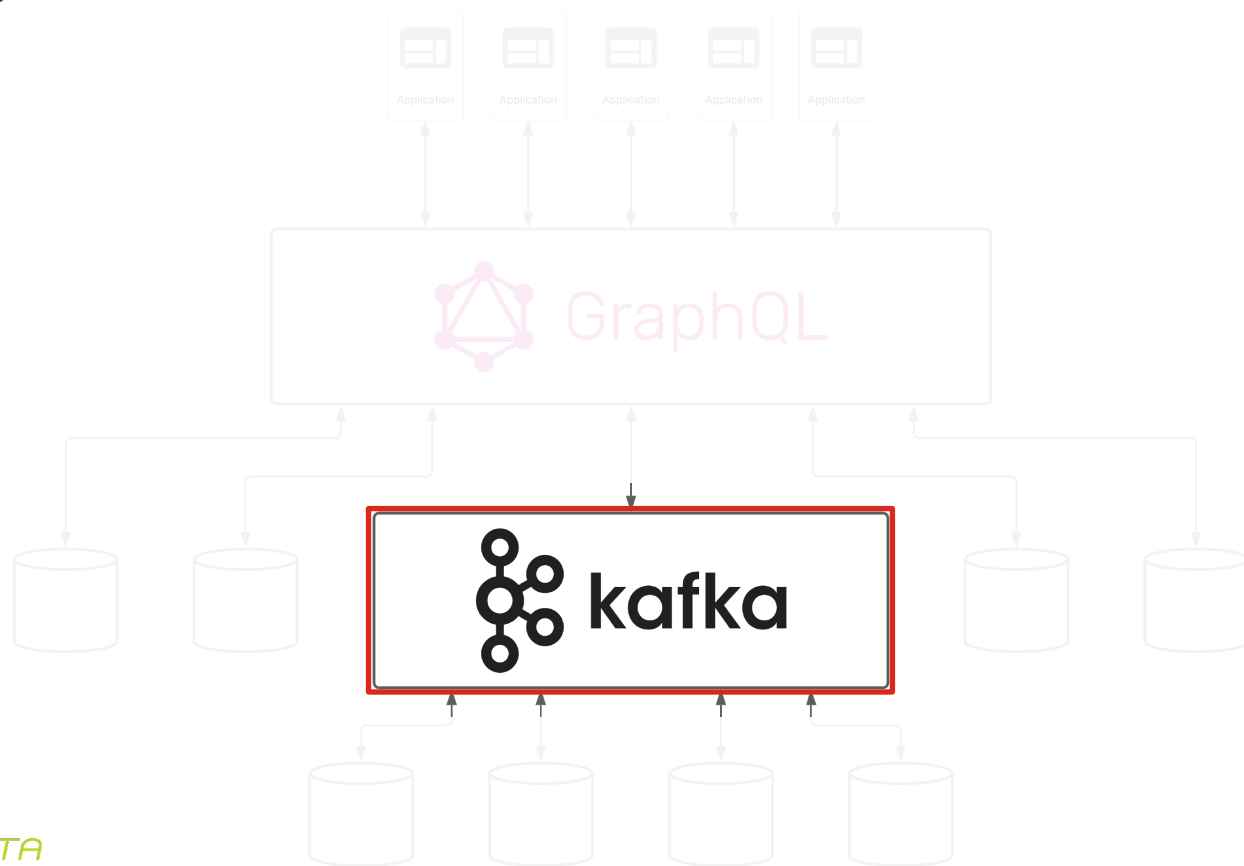
# Polyglot data infrastructure



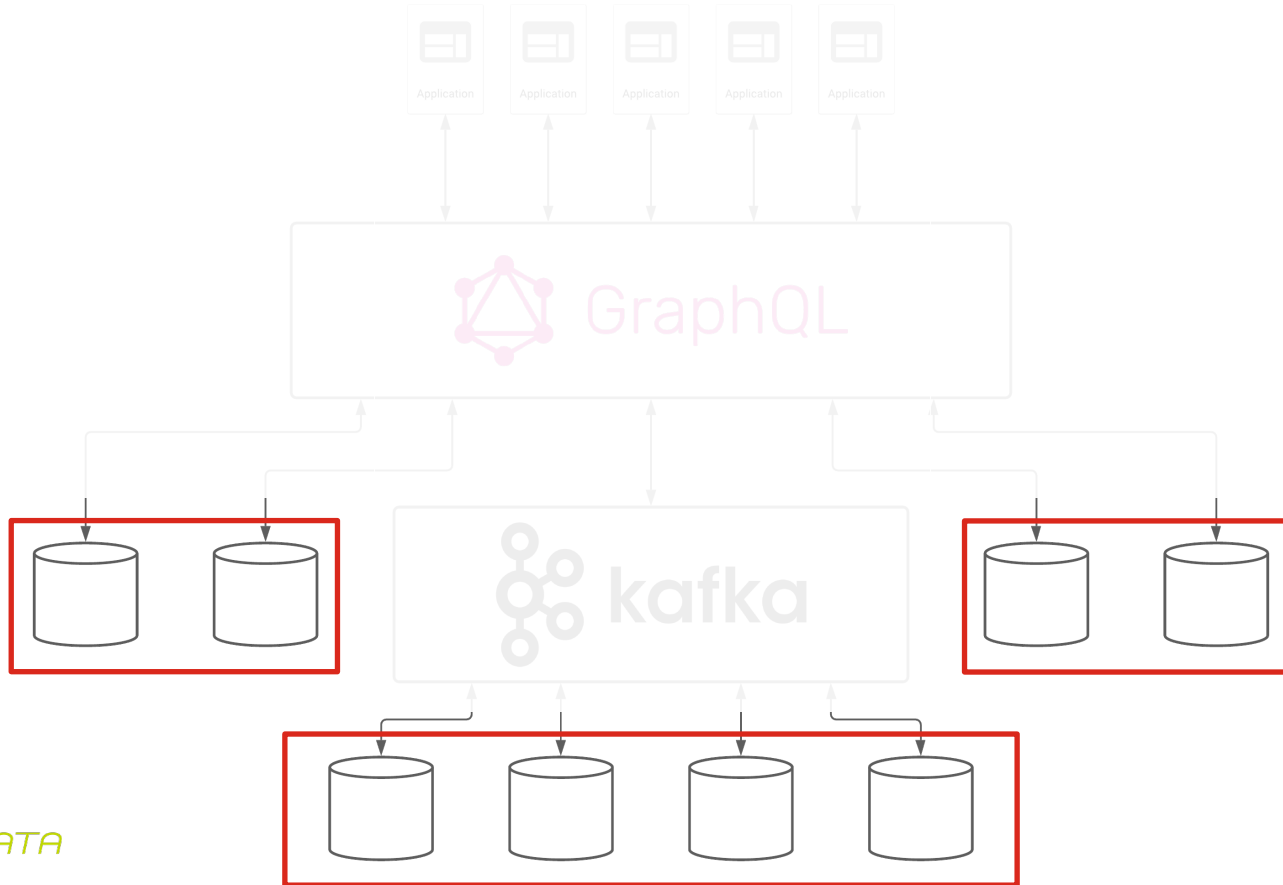
# Polyglot data infrastructure



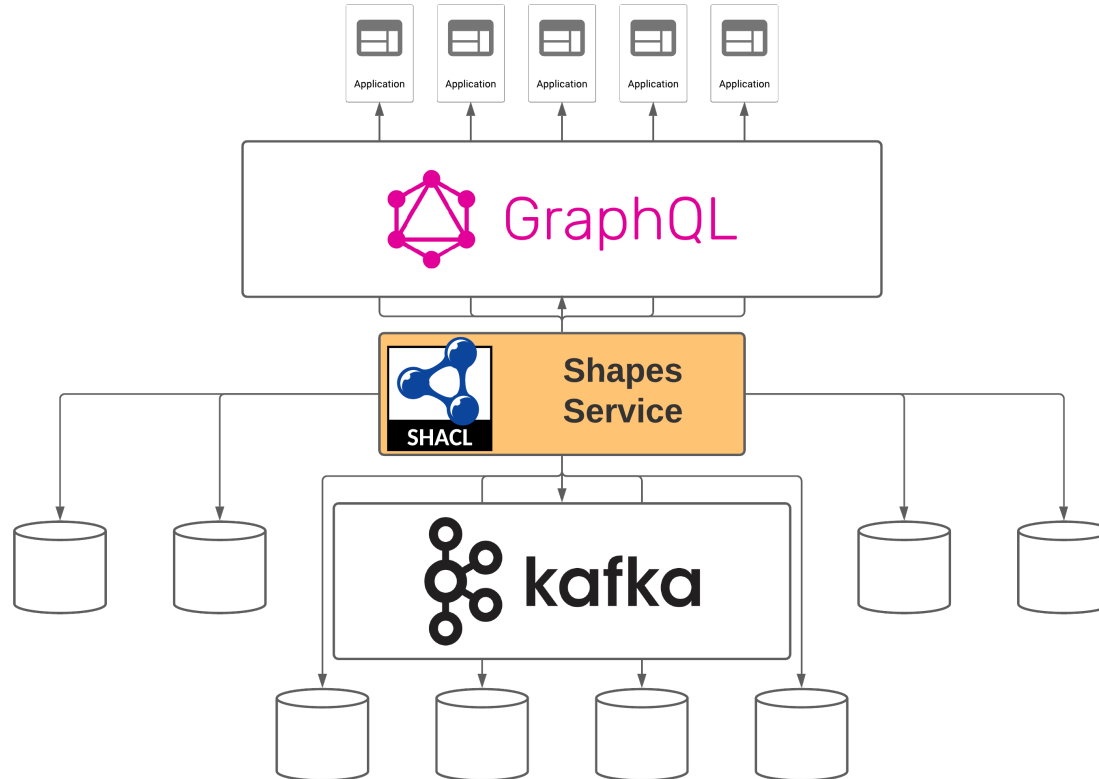
# Polyglot data infrastructure



# Polyglot data infrastructure



# Metadata deployment (schemas)



# The canonical view

## A robust conceptual model

```
mds:Person a sh:NodeShape ;
  rdfs:label "Person"@en ;
  skos:definition "A human being"@en ;
  sh:targetClass foaf:Person ;
  sh:property [
    sh:path schema:givenName ;
    sh:datatype xsd:string ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
  ] ;
  sh:property [
    sh:path schema:familyName ;
    sh:datatype xsd:string ;
    sh:minCount 1 ;
  ] ;
  sh:property [
    sh:path schema:height ;
    sh:datatype xsd:float ;
    sh:minExclusive 0 ;
  ] .
```



# Applications

## Collection and presentation based on SHACL shapes

### Person

Given Name:

Family Name(s):  

Height:

```
<label>Person</label><br>
<form action="/person.php">
  <label for="givenName">Given name:</label><br>
  <input type="text" id="givenName" name="givenName" required><br>
  <label for="familyName">Family Name(s) :</label><br>
  <input type="text" id="familyName" name="familyName" [] required>
  <button type="button" class="addFamilyName">+</button><br>
  <label for="height">Height:</label><br>
  <input type="float" id="height" name="height"><br>
</form>
```

# GraphQL Schema

Objects are composed of SHACL shapes

```
type Person {  
  id:ID!  
  givenName: String!  
  familyName: [String]!  
  height: Float  
}
```

# Kafka messages (avro)

Schemas are based on SHACL shapes

```
{ "type" : "record",
  "namespace" : "core",
  "name" : "Person",
  "fields" : [ { "name" : "givenName" ,
                 "type" : "string"
               },
               { "name" : "familyName" ,
                 "type" : "array",
                 "items" : "string"
               },
               { "name" : "height" ,
                 "type" : ["float", "null"]
               }
             ]
}
```

# Relational DBs

DDL files are based on SHACL shapes

```
create table person (  
    id varchar NOT NULL,  
    givenName varchar NOT NULL,  
    familyName varchar [] NOT NULL,  
    height float  
);
```

# Additional notes and considerations

- Object models can be cast to languages to support working with shapes during processing
- Data Validation
  - RDF validation uses raw SHACL capabilities
  - [greatexpectations.io](https://greatexpectations.io) auto-generation and deployment based on SHACL defined constraints
- Model serialization and deployment is tracked within the shapes service
- Versioning tracked at both the individual shape and shape collection levels

# Knowledge @ Medidata

PHUSE SDE - 12 Aug 2021

Questions?