

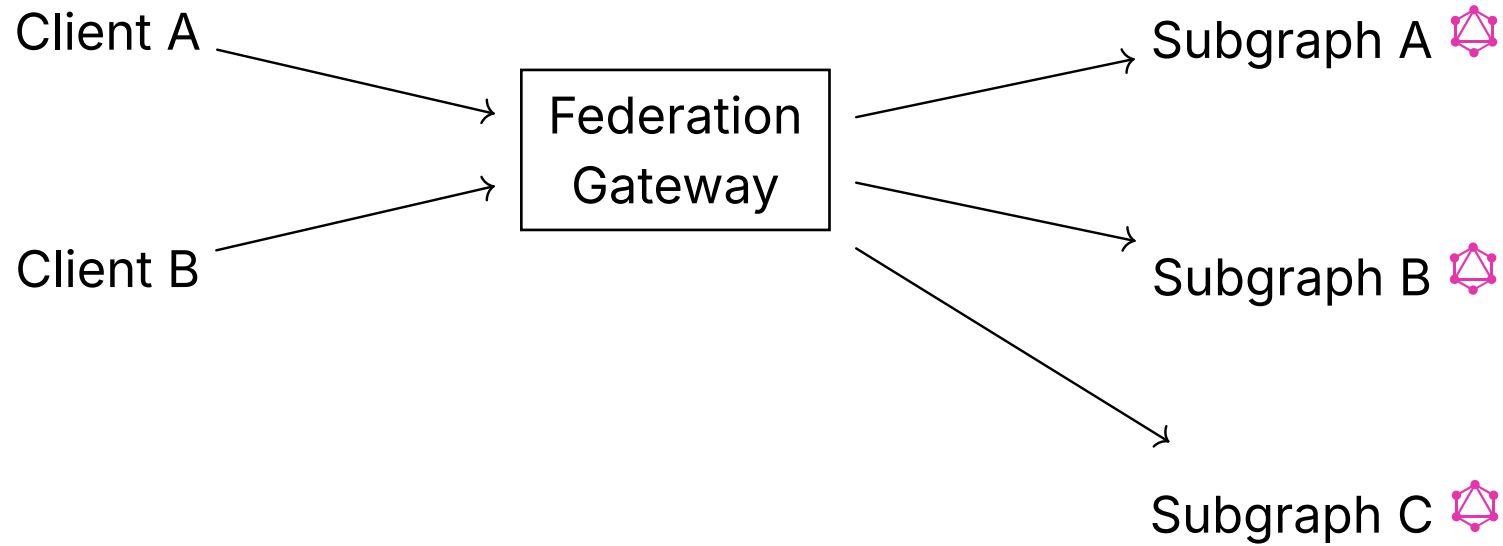


# Beyond GraphQL Federation

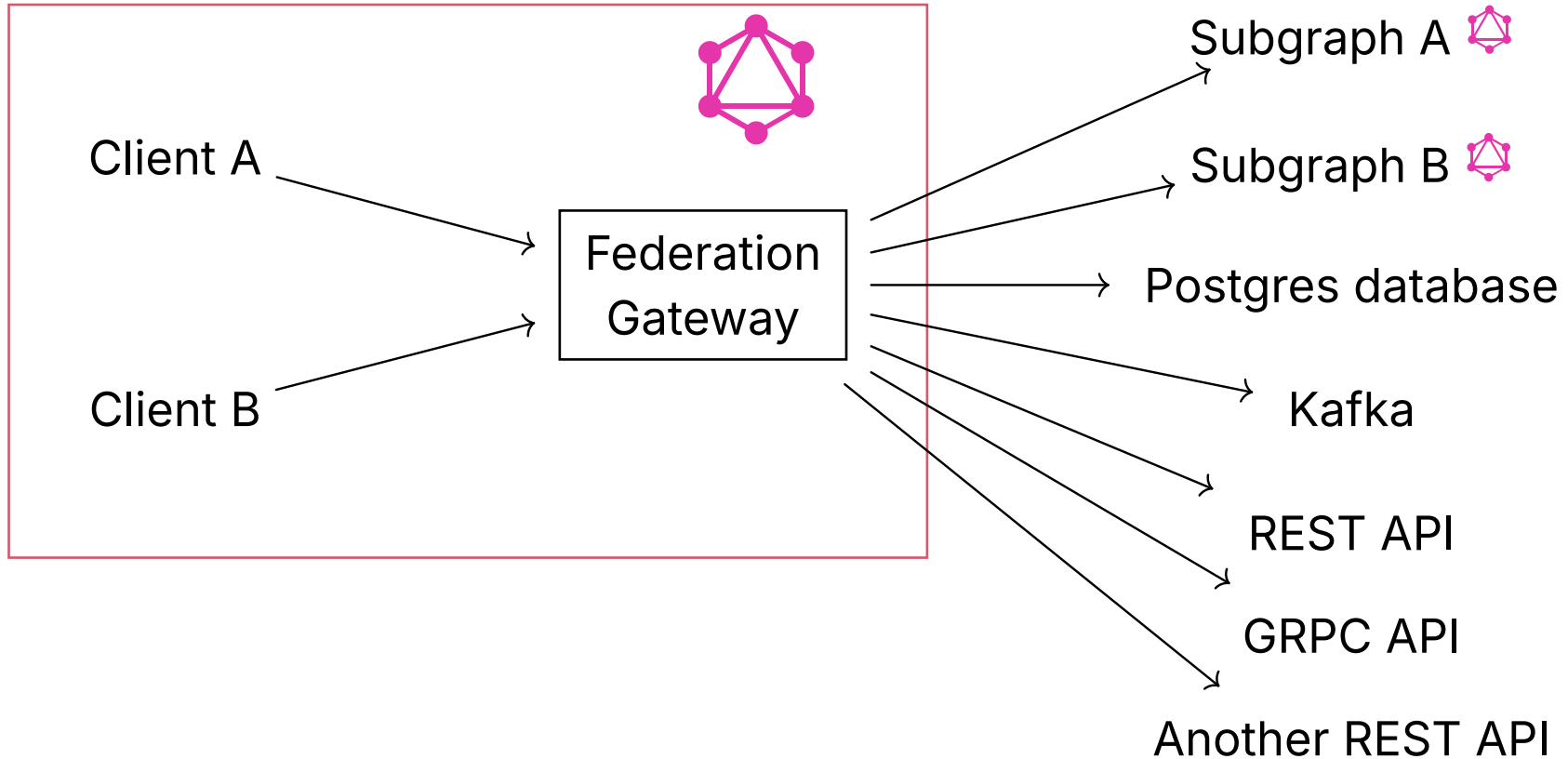
**How We Use Composite Schemas and WebAssembly to  
Federate Non-GraphQL Data Sources**

Benjamin Rabier and Tom Houlé

# Traditional GraphQL Federation



# Extensions-enabled GraphQL Federation



# Grafbase Extensions

- Wasm modules

# Grafbase Extensions

- Wasm modules
- Rust SDK

# Grafbase Extensions

- Wasm modules
- Rust SDK
- Open source extensions, or locally built

# Types of extensions

# Types of extensions

- Resolvers



# Types of extensions

- Resolvers
- Authentication

# Types of extensions

- Resolvers
- Authentication
- Authorization

# Types of extensions

- Resolvers
- Authentication
- Authorization
- Schema contracts

# Composite Schemas

- WIP
- Evolution of the Apollo Federation spec
- Collaborative effort in a GraphQL Foundation working group
- Smooth transition path

# Composite schemas differences: @lookup

Replaces **entity resolvers**.

```
1
2  type Query {
3    version: Int
4    productById(id: ID!): Product @lookup
5    productByName(name: String!): Product @lookup
6  }
7
8  type Product {
9    id: ID!
10   name: String!
11 }
```



## Composite schemas differences: @requires

@requires on fields becomes @require on arguments.

```
1 type Product {  
2   id: ID!  
3   weight: Int!  
4   delivery(  
5     zip: String!  
6     weight: Int! @require(field: "weight")  
7   ): DeliveryEstimates  
8 }
```



# Composite schemas differences: @require

More complicated example:

```
1  type Product {  
2    id: ID!  
3    delivery(  
4      zip: String!  
5      dimension: ProductDimensionInput!  
6        @require(  
7          field: "{ productSize: dimension.size, productWeight:  
8            dimension.weight }"  
9        )  
10   ): DeliveryEstimates  
11 }
```



```
12 type ProductDimension {
13     size: Int!
14     weight: Int!
15 }
16
17 input ProductDimensionInput {
18     productSize: Int!
19     productWeight: Int!
20 }
```



# Links

- Composite Schemas spec (draft)
- Composite Schemas WG meeting videos on the GraphQL Foundation Working Groups YouTube channel
- Grafbase SDK
- Open source extensions repo
- Grafbase Extensions docs
  - Importable types