

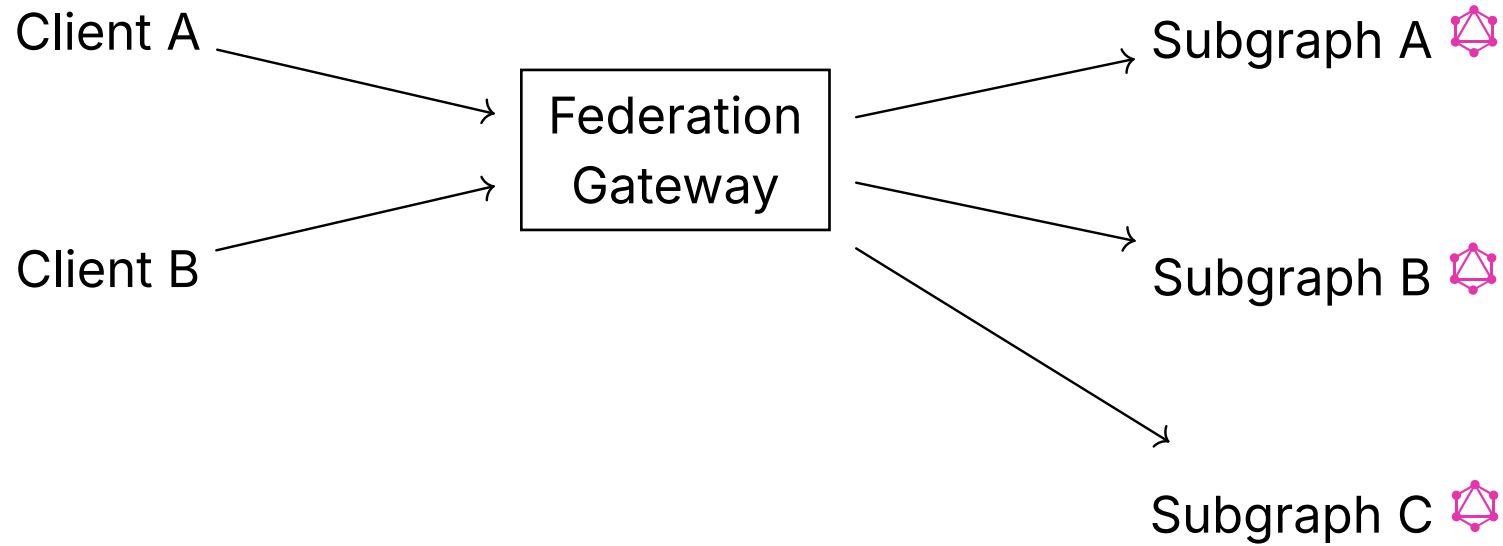


Beyond GraphQL Federation

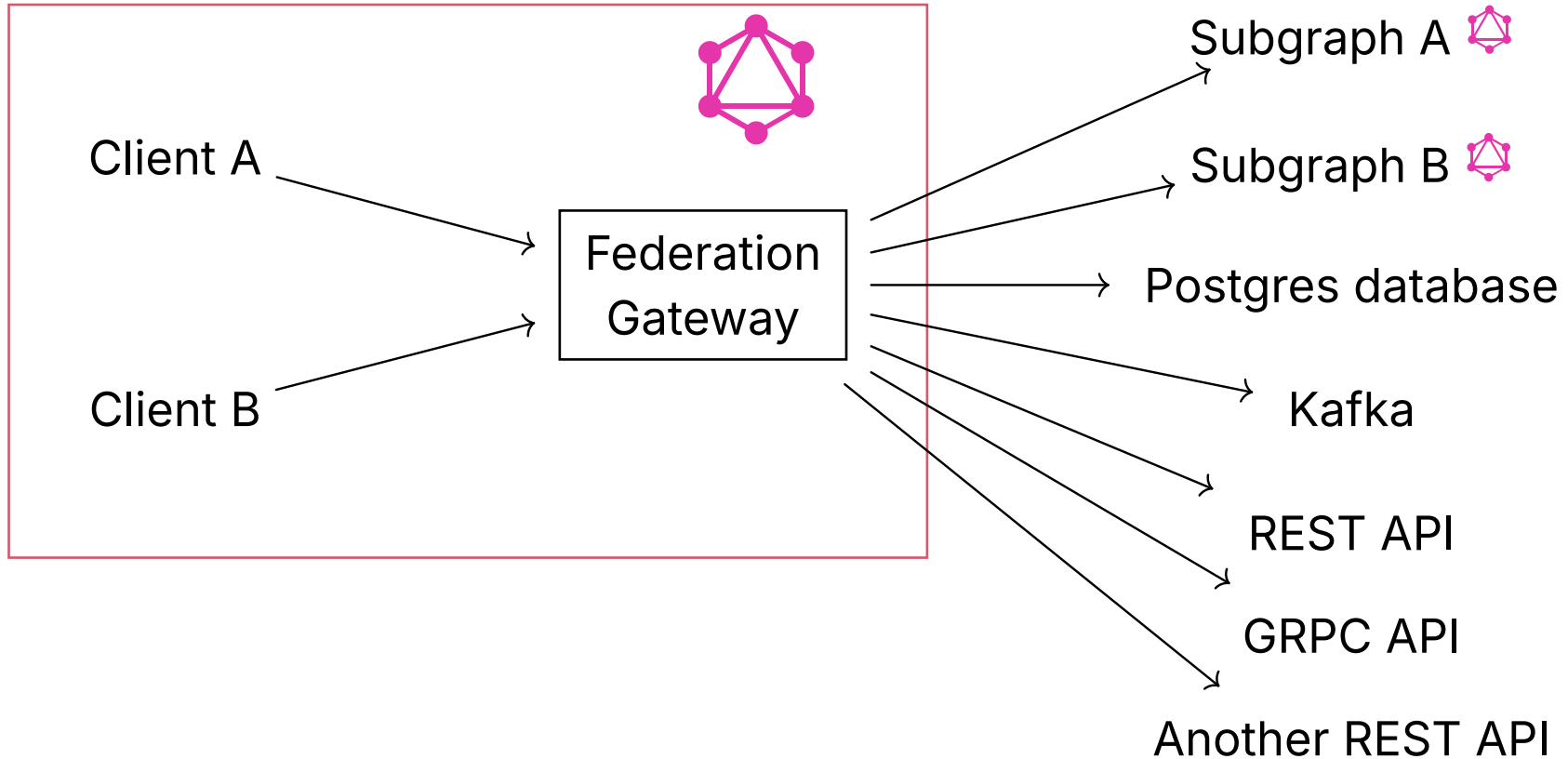
**How We Use Composite Schemas and WebAssembly to
Federate Non-GraphQL Data Sources**

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

Types of extensions

- Resolvers
- Authentication
- Authorization
- Schema contracts
- Hooks

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

```
1  type Product {
2    id: ID!
3    dimension: ProductDimension @external
4    delivery(
5      zip: String!
6      dimension: ProductDimensionInput!
7      @require(
8        field: ""
9        {
10          productSize: dimension.size,
11          productWeight: dimension.weight
12        }
13        ""
14      )
15    ): DeliveryEstimates
16  }
```

 GraphQL

```
17
18  type ProductDimension {
19    size: Int!
20    weight: Int!
21  }
22
23  input ProductDimensionInput {
24    productSize: Int!
25    productWeight: Int!
26  }
```


Links (1)

- [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](#)

Links (2)

- [gRPC + composite schemas blog_post](#)
- [Composite Schemas blog_post](#)
- [Schema contracts announcement blog_post](#)
- [gRPC extension announcement blog_post](#)
- [Kafka extension announcement blog_post](#)
- [Postgres extension announcement blog_post](#)
- [Commercetools case study with how they use hooks and schema contract extensions](#)
- [Pantheon case study. They also use extensions.](#)

Hands-on workshop



Workshop links

- ➔ <https://github.com/grafbase/federation-demo>
- CLI: <https://grafbase.com/docs/cli/installation>
Or `curl -fsSL https://grafbase.com/downloads/cli | bash`
- Gateway: <https://grafbase.com/docs/gateway/installation>
Or `curl -fsSL https://grafbase.com/downloads/gateway | bash`
- [MCP](#)
- [Extensions docs](#)

