An Introduction to GraphQL
Tutorial at ISWC 2019, October 27, 2019

5. New Developments

Olaf Hartig[a], Ruben Taelman[b]

(a) Dept. of Computer and Information Science, Linköping University, Sweden

(b) Ghent University – imec – IDLab, Belgium

LINKÖPING
UNIVERSITY

Schema Federation

- Use case and motivation:
  - Break a monolithic GraphQL schema (and, thus, server) into several components
  - Separation of concerns
  - Micro service architecture
  - Avoid fragile “schema stitching” code
- Implementation approach: Apollo Server
  - @apollo/federation: primitives for “implementing services” to make their individual schemas composable
  - @apollo/gateway: set up an instance of Apollo Server as a gateway that distributes incoming GraphQL operations across one or more implementing services

Schema Federation – Example

```
type User @key(fields: "id") {
  id: ID!
  username: String!
}
```

accounts
service

```
extend type User @key(fields: "id") {
  id: ID! @external
  reviews: [Review]
}
type Review {
  comment: String
  author: User
  product: Product
}
```

reviews
service

Schema Federation – Example (cont'd)

```
const gateway = new ApolloGateway({
 serviceList: [
  { name: 'accounts', url:'http://localhost:4001' },
  { name: 'products', url:'http://localhost:4002' },
  { name: 'reviews',  url:'http://localhost:4003' }
 ]
});

const server = new ApolloServer({ gateway });
server.listen();
```

www.liu.se

LINKÖPING
UNIVERSITY