# GraphQL Java and Spring: The Latest Features

Rossen Stoyanchev

@rstoya05

# About me

## Spring Framework team

Spring MVC, WebSocket, Messaging, WebFlux, RSocket, Testing

## Spring for GraphQL

Started through collaboration with GraphQL Java

**Rossen Stoyanchev**

# About You

# Agenda

Spring for GraphQL 1.3

GraphQL Java 22

Collaboration with Netflix DGS

# Demo:

https://github.com/rstoyanchev/spring-graphql-activity

# Spring for GraphQL 1.3

Virtual threads

Blocking `GraphQlClient` with synchronous API

`DgsGraphQlClient` for DGS codegen

Schema inspection of unions, interfaces and arguments

Interface field mapping

# Federation Support in Spring for GraphQL 1.3

`FederationSchemaFactory`

Entities query via `@EntityMapping` methods

Batch loading

# Demo:

https://github.com/apollographql/federation-jvm-spring-example

# More in Spring for GraphQL 1.3

SSE transports

WebSocket auth via “`connection_init`”

WebSocket keep-alive

GraphQL endpoint Jackson codec

# Examples Repository:

https://github.com/spring-projects/spring-graphql-examples

# GraphQL Java 22

Incremental Delivery via @defer

Prioritize data in response

Experimental API

# Demo:

https://github.com/felipe-gdr/spring-graphql-defer

# More in GraphQL Java 22

Performance improvements

Schema diffing tool for change analysis

Built-in DataLoader support (not via instrumentation)

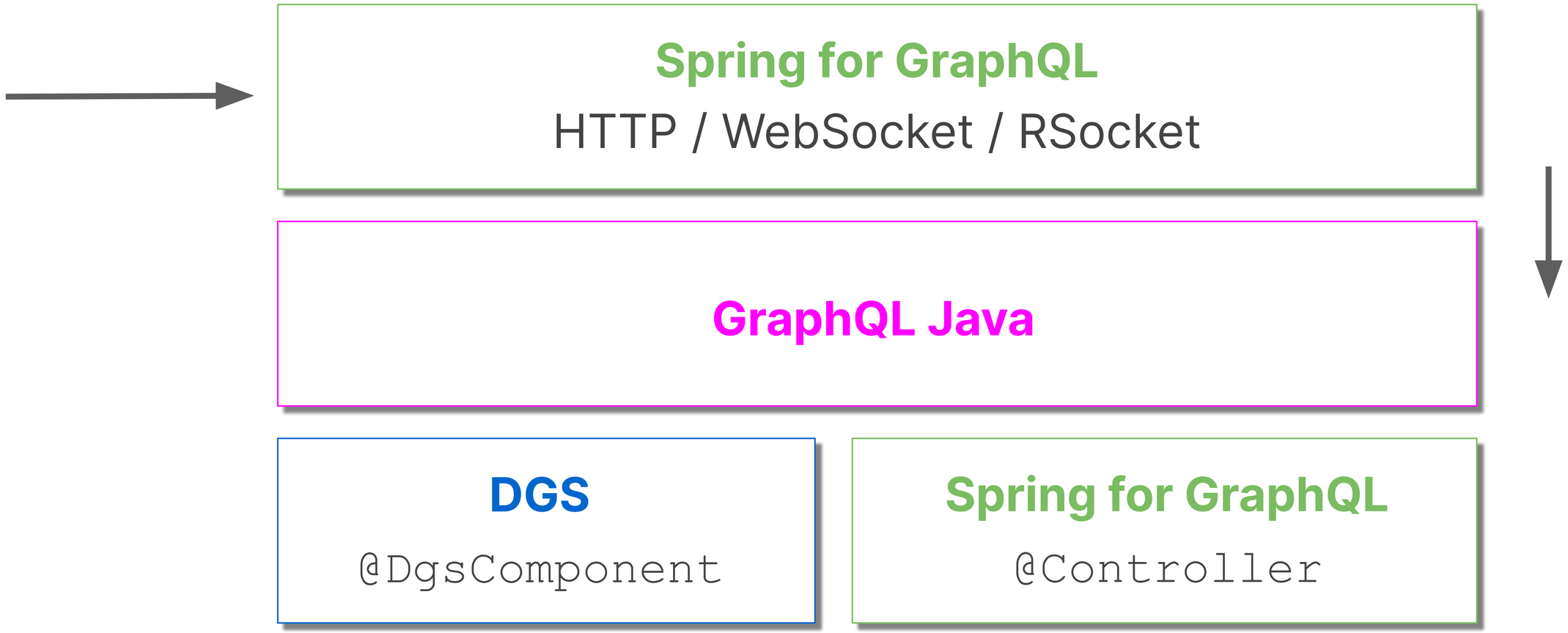
@oneOf directive

# DGS Integration with Spring for GraphQL

Common transport and execution foundation

Both programming models

https://netflix.github.io/dgs/spring-graphql-integration/


Spring for GraphQL
HTTP / WebSocket / RSocket
GraphQL Java
DGS
@DgsComponent
Spring for GraphQL
@Controller

# start.spring.io

GraphQL DGS code generation

Netflix DGS

# THANKS!!!

Rossen Stoyanchev

@rstoya05