Developing GraphQL Server

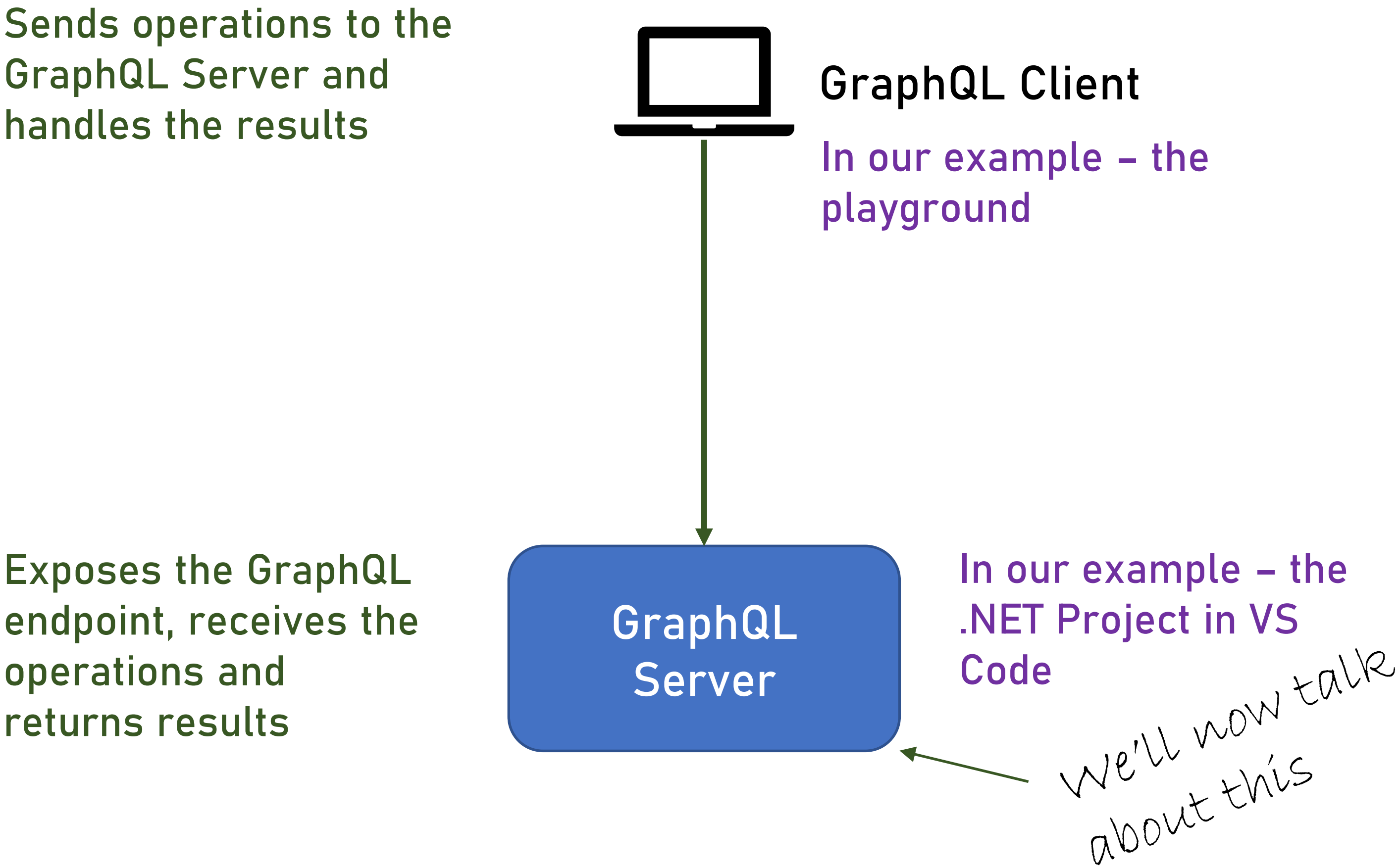
Memi Lavi
www.memilavi.com

Developing GraphQL Server

- We learned quite a lot about GraphQL specifications and syntax
- Now it's time to actually develop GraphQL-based system

Developing GraphQL Server

- Two components in GraphQL-based system:



Sends operations to the GraphQL Server and handles the results
GraphQL Client
In our example – the playground
Exposes the GraphQL endpoint, receives the operations and returns results
GraphQL Server
In our example – the .NET Project in VS Code
We'll now talk about this

GraphQL Server Roles

- The GraphQL Server has the following roles:

Define GraphQL Schema

Expose GraphQL Endpoint

Validate GraphQL Operations

Route GraphQL Operations to Code

Convert to code entities

Return GraphQL Results

Must follow GraphQL Specifications!

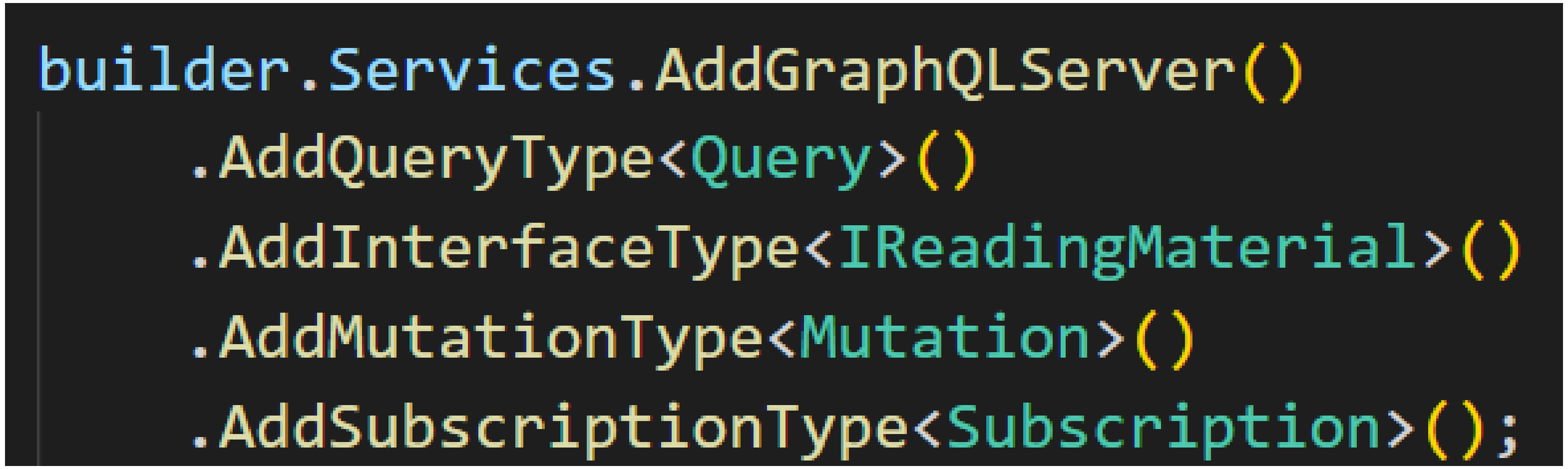
GraphQL Server Roles

Define GraphQL Schema

- The GraphQL Server defines the schema using code elements in the server or explicitly using the Schema Definition Language
  - Usually based on classes / modules
  - Each server has its own schema definition mechanism

GraphQL Server Roles

Define GraphQL Schema



```
builder.Services.AddGraphQLServer()
    .AddQueryType<Query>()
    .AddInterfaceType<IReadingMaterial>()
    .AddMutationType<Mutation>()
    .AddSubscriptionType<Subscription>();
```

GraphQL Server Roles

Expose GraphQL Endpoint

- All GraphQL operations are sent to a specific, single, endpoint
    - As opposed to REST API which exposes an endpoint for each operation:
        - `GET /api/v1/books/{id}`
        - `POST /api/v1/employee`

GraphQL Server Roles

Expose GraphQL Endpoint

- The endpoint is usually named `graphql`, and is accessed with a `POST` verb
- The server expects to receive JSON payload in this endpoint

GraphQL Server Roles

Validate GraphQL Operations

- The server receives the JSON payload and validates it against the schema
- If a mismatch is found – the server returns a standard error message
- No coding is required for this validation

GraphQL Server Roles

Route GraphQL Operations to Code

- The server converts the JSON payload to objects and triggers methods to process the operations and return the data
- These methods are called `Resolvers`

GraphQL Server Roles

Route GraphQL Operations to Code

- The server traverses through the query and runs a resolver *for every field*
- If the field returns a scalar value – this is what's returned
- If the field returns an object value – the server runs resolvers for each field in the object

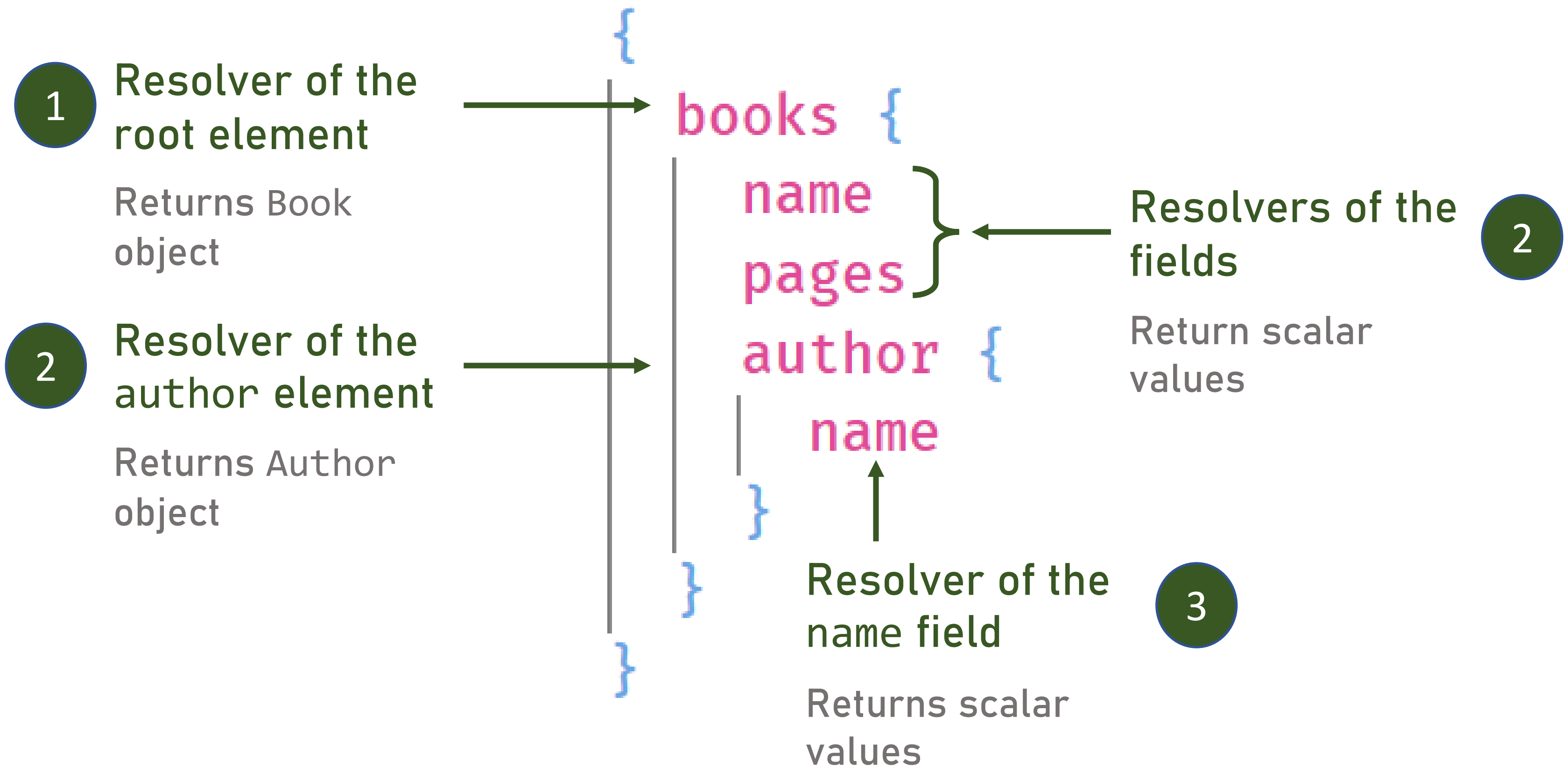
GraphQL Server Roles

Route GraphQL Operations to Code

- …and so on until a scalar is returned
- Resolvers in the same hierarchy run in parallel
- You can’t assume one will run before the other
- NEVER use data from another resolver in your resolver

GraphQL Server Roles

Route GraphQL Operations to Code



```
{
  books {
    name
    pages
    author {
      name
    }
  }
}
```

1 Resolver of the root element

Returns Book object

2 Resolver of the author element

Returns Author object

2 Resolvers of the fields

Return scalar values

3 Resolver of the name field

Returns scalar values

GraphQL Server Roles

Route GraphQL Operations to Code

- Most implementations automatically create resolvers for fields
  - As in our case
- Custom resolvers can be specified
- Resolvers can accept arguments with context data about the query

GraphQL Server Roles

Return GraphQL Results

- After running the resolvers and merging the results –
- The server converts the object to JSON and returns it to the client

Developing GraphQL Server

- We'll learn how to develop GraphQL in two platforms:
  - .NET
  - NodeJS
- Always use existing libraries for doing that, never develop your own implementation

Developing GraphQL Server in .NET

- There are various libraries for GraphQL in .NET
- One of the most popular is HotChocolate
- This is what we'll use in our demo