

AAD / Backend Development

Backend Layers

Erik van der Arend / Jan Willem Boer

23/24 Q3



What to submit

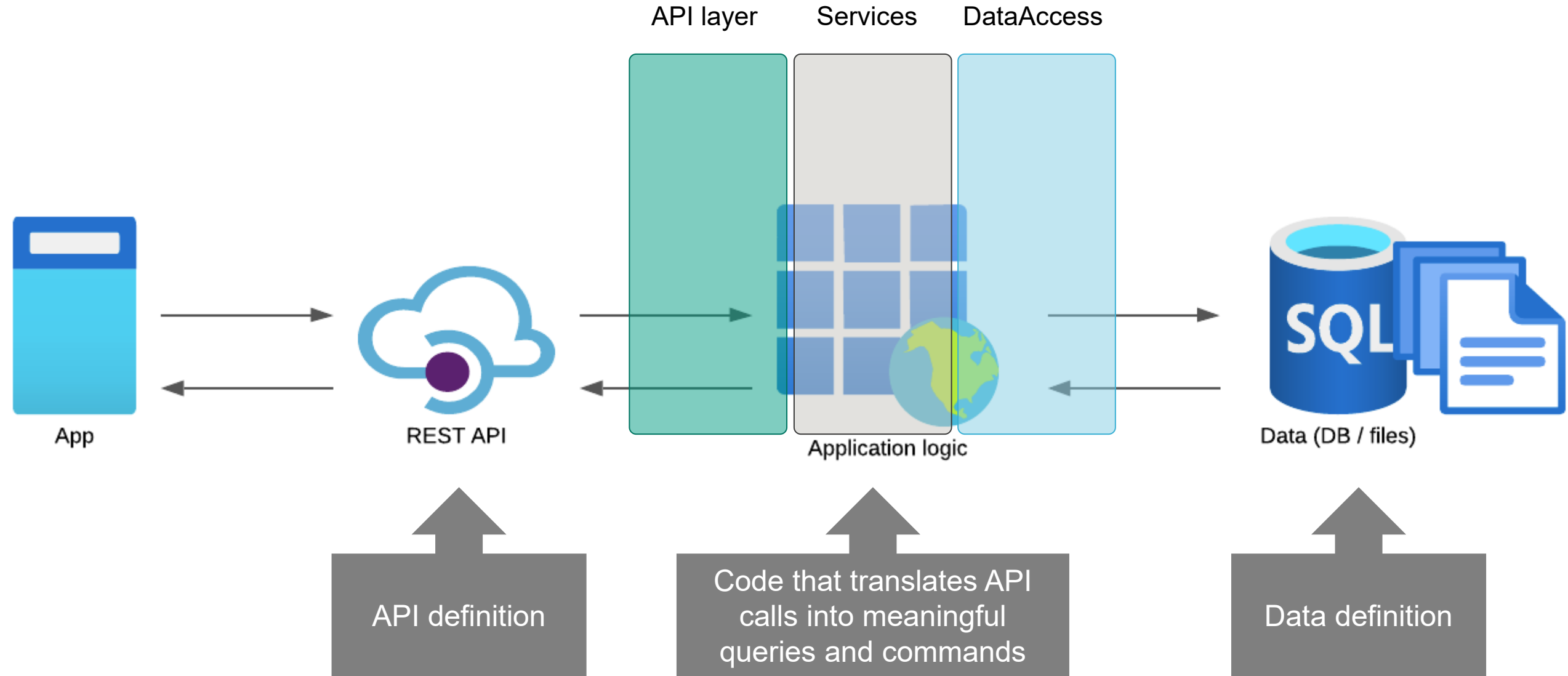
- PDF, documenting:
 - Step 1 (research, criteria, score, rationale)
 - Step 2 (reflection on both prototypes)
 - Step 3 (conclusion with rationale)
- ZIP file with prototype 1
- ZIP file with prototype 2

Layers

Separation of Concerns

To keep code readable and testable, use Separation of Concerns (SoC) everywhere:

- In methods
- In classes
- In “layers”



API layer

Has

- Controllers
- Actions
- ...



Does

- Routing
- Authentication
- Logging/errorhandling
- DTO / model mapping
- ...

Services

Has

- Services
- Helpers
- Domainmodel
- ...



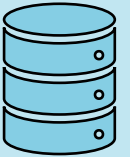
Does

- The actual work
- Calculations
- Calling other components
- Interaction and manipulation of domain objects

Data access layer

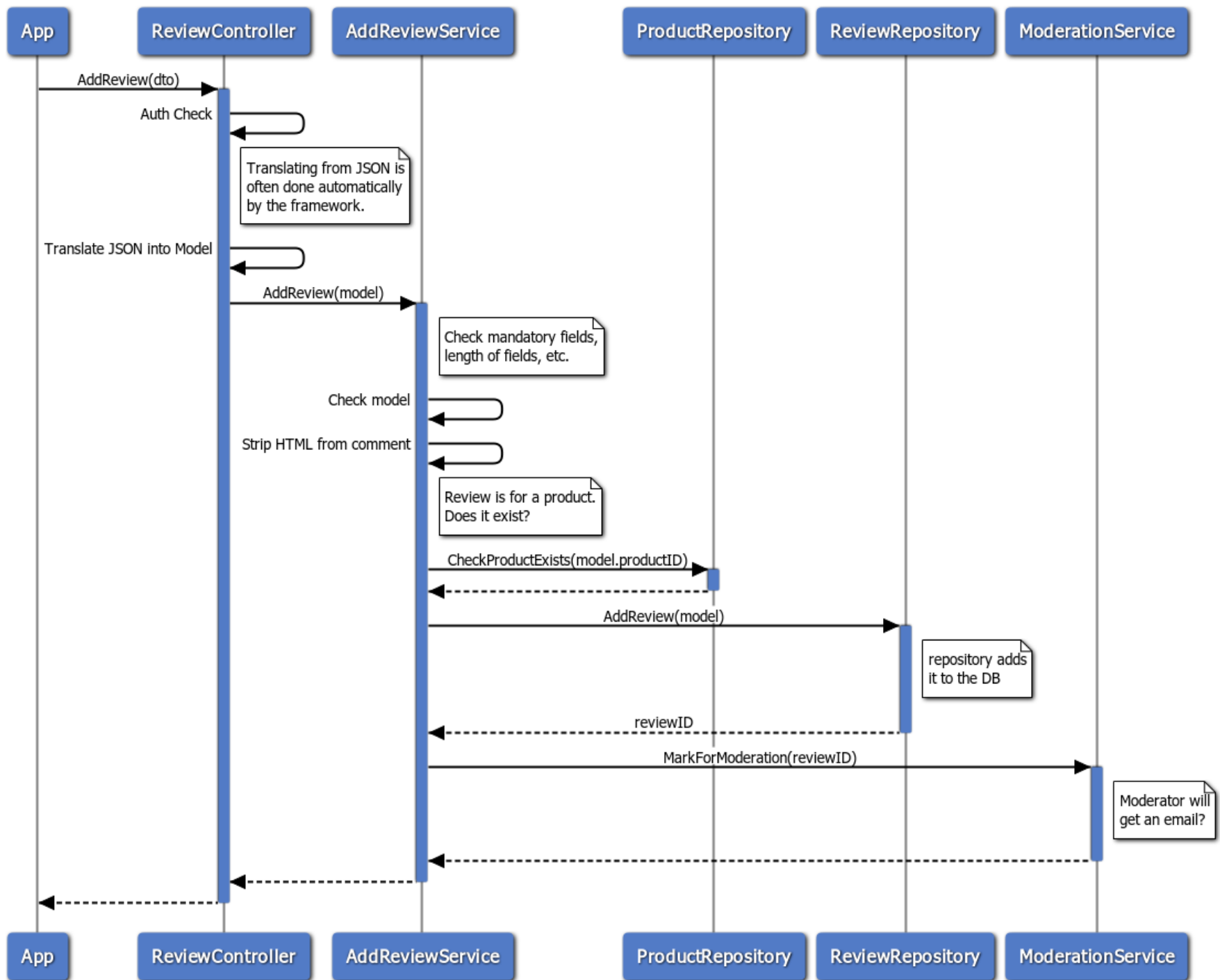
Has

- Data definition
- Repositories
- Commands / queries
- ...



Does

- Interaction with the database
- Interaction with the filesystem



Organizing layers in your code

Directory per layer

```
└─ Controllers
   │ C# OrderController.cs
   │ C# ProductController.cs
   │ C# ReviewController.cs
   └─ DataAccess
      │ C# OrderRepository.cs
      │ C# ProductRepository.cs
      │ C# ReviewRepository.cs
      └─ Models
         │ C# Order.cs
         │ C# Product.cs
         │ C# Review.cs
         └─ Services
            │ C# AddReviewService.cs
            │ C# ChangeReviewService.cs
            │ C# ModerationService.cs
            │ C# PlaceOrderService.cs
            │ C# UpdateProductService.cs
```

Directory per subject

```
└─ Order
   │ C# OrderController.cs
   │ C# OrderModel.cs
   │ C# OrderRepository.cs
   │ C# PlaceOrderService.cs
   └─ Product
      │ C# ProductController.cs
      │ C# ProductModel.cs
      │ C# ProductRepository.cs
      │ C# UpdateProductService.cs
      └─ Review
         │ C# AddReviewService.cs
         │ C# ChangeReviewService.cs
         │ C# ModerationService.cs
         │ C# ReviewController.cs
         │ C# ReviewModel.cs
         │ C# ReviewRepository.cs
```


Practice

- There is a crappy server which works ...
...but has all layers combined in one file
- Clone or fork the project at

`https://edu.nl/bcpay`

or

`https://github.com/janwillemboer/crappy-project`

- Fix the layers in the server by restructuring it



Nice, but how does this apply to me?

- Research how to organize the layers in the framework you chose.
- Organize the layers in you application
- Document this in your technical documentation

Note that using layers/components is mandatory for being graded.

Now what

This week

- Finish Assignment 1 (create the backend in two different frameworks)
- Document your steps
- Look at the rubric to know what to submit
- Submit at last on Sunday, September 24th, 23:59 for feedback.

Next week

- Continue with Assignment 2 (Continue with one of the frameworks).