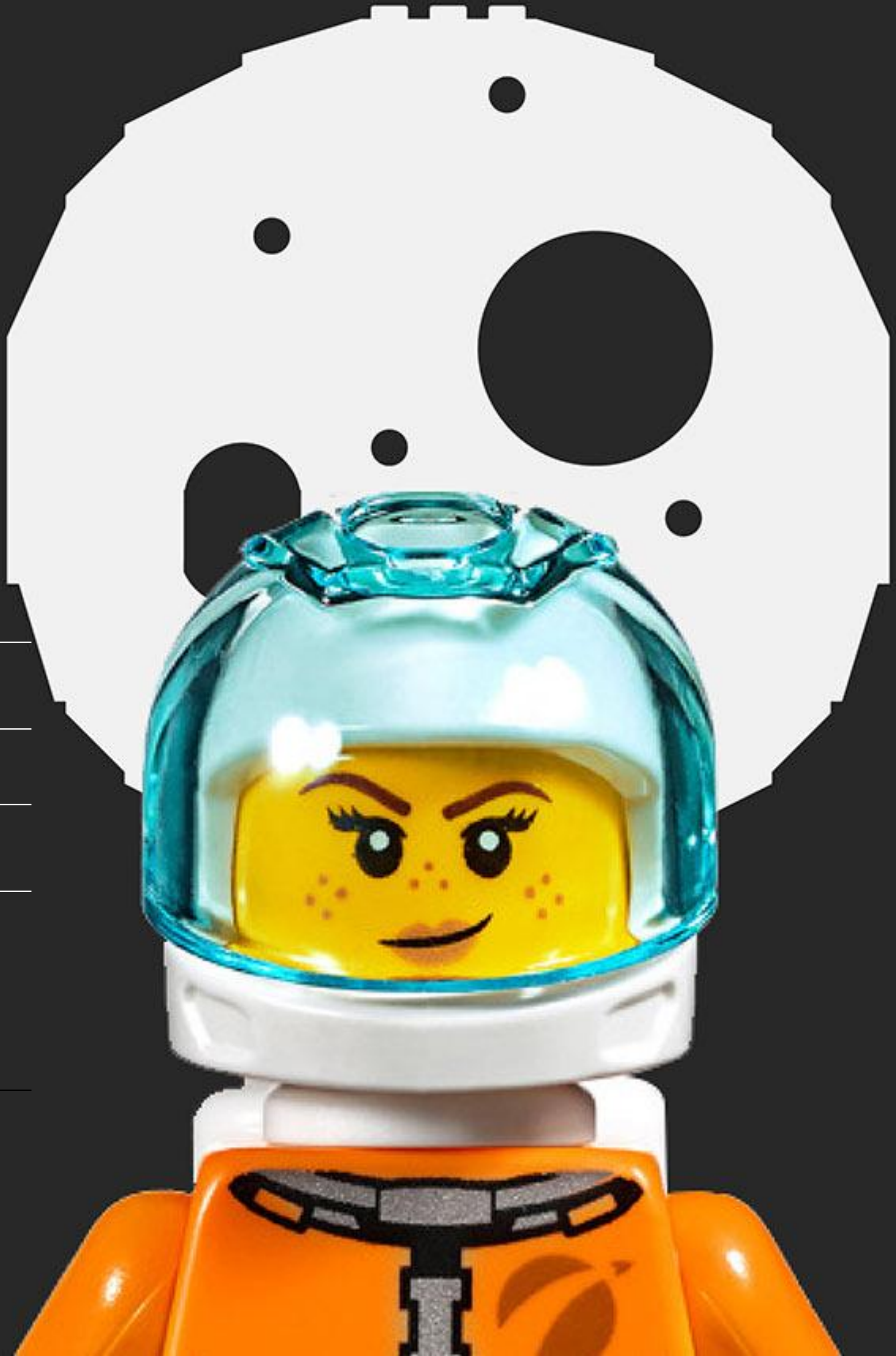
Modern C# For Python Developers

ASP.NET Core Service Jam Session

October 17, 2025

Jesper Gulmann Henriksen

Lead IT Engineer
BrickLink Technology DK
jesper.henriksen@LEGO.com



Agenda for Session 5

Introduction

5.1 Controllers

- Scaffold a New Web API Project
- Controllers

5.2 Middleware

- Processing Pipelines
- Example of Middleware

5.3 Architecture

- A More Realistic Example
- Domain-centric architectures

5.4 Adding Tests

- Unit testing with xUnit
- AAA

5.5 Minimal APIs

- Scaffold Another New Web API Project
- Minimal APIs

5.6 API Matters

- LEGO Guidelines
- REST Naming and Resources
- Idempotency

Conclusion

Module 5.1
Controllers

ASP.NET Web API Project Creation

- Never create project manually from scratch ☺

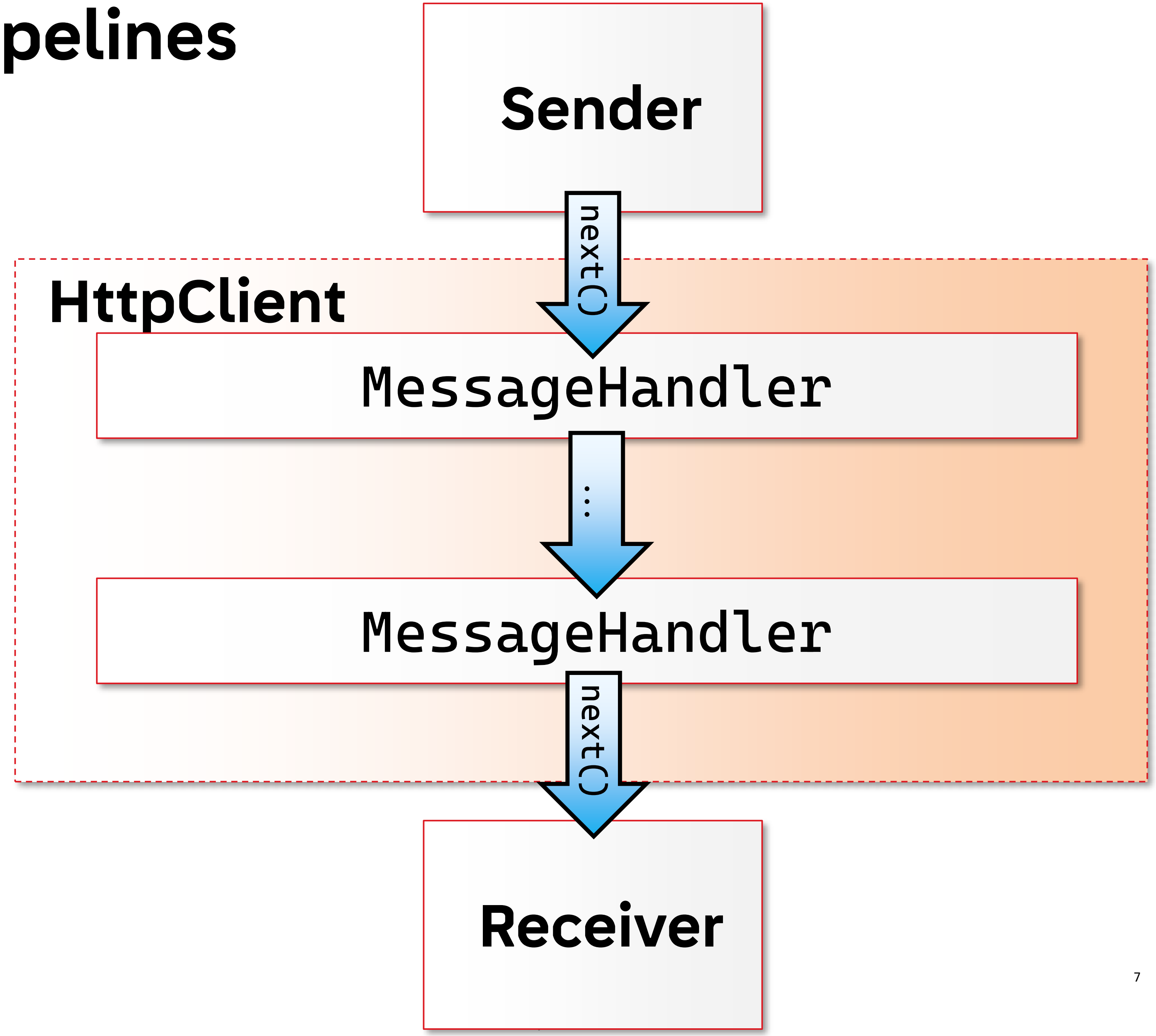
- Valid approaches
  - Use tools provided with IDE, or
  - Use command-line, i.e. `dotnet new` , or
  - Copy from existing template / project skeleton

Interesting Points

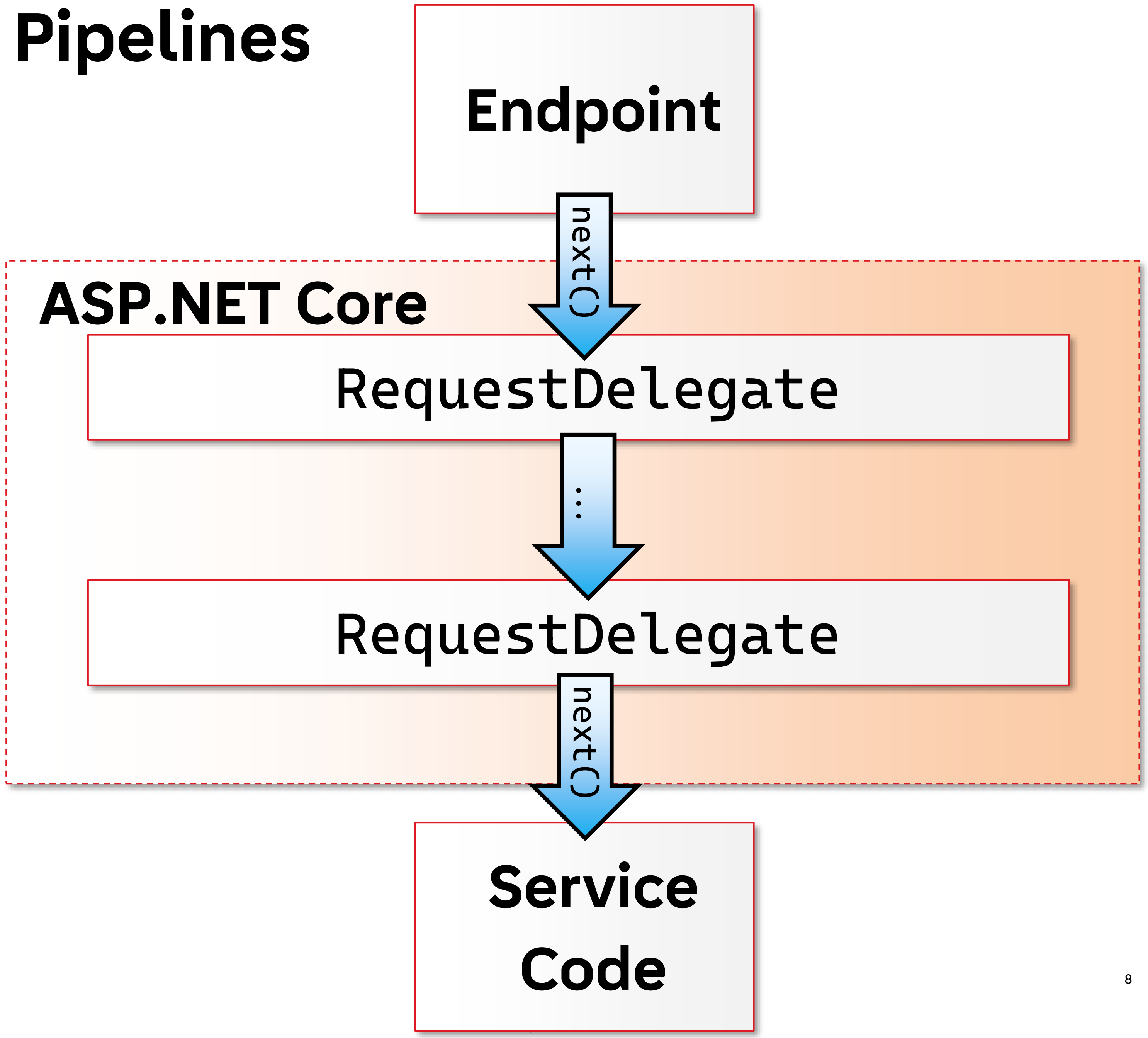
- Processing Pipeline
- Dependency Injection
- .http files
- …

Module 5.2 Middleware

Client Pipelines



ASP.NET Pipelines



The Full (and Complicated) Answers

Simple introduction to three kinds of custom middleware:
https://www.milanjovanovic.tech/blog/3-ways-to-create-middleware-in-asp-net-core

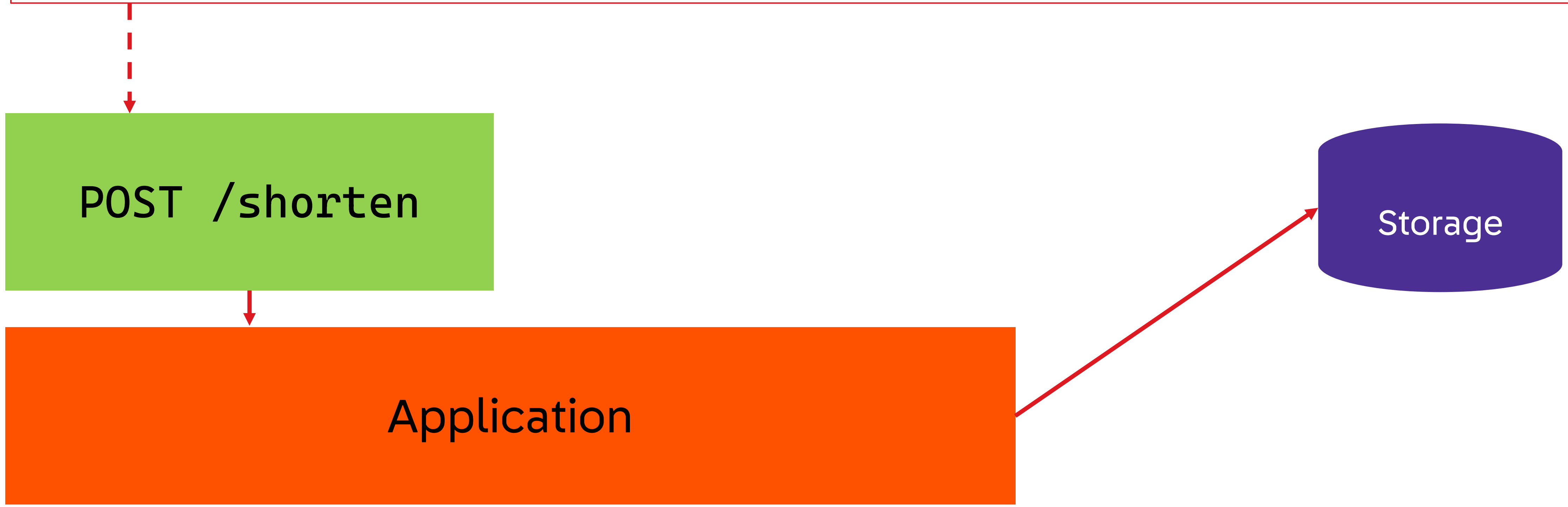
The full processing cycle of ASP.NET Core:
https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-8.0

Module 5.3 Architecture

Request: POST /shorten

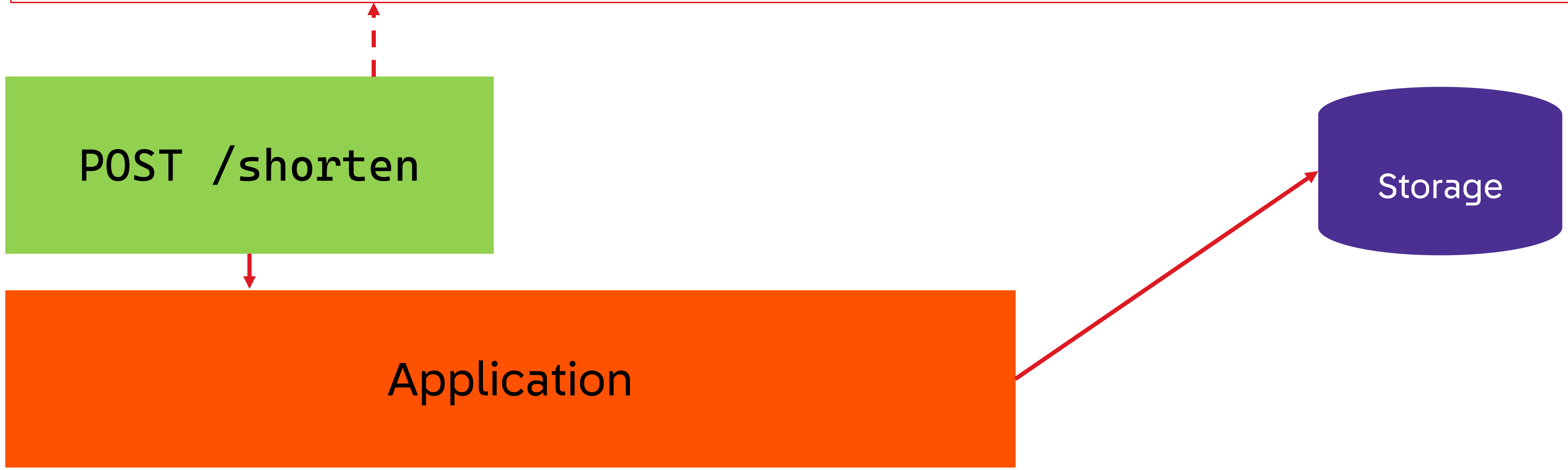
```
POST {{host}}:{{port}}/shorten
Content-Type: application/json

{
    "code": "falcon",
    "longUrl": "https://www.lego.com/en-dk/product/millennium-falcon-75192"
}
```



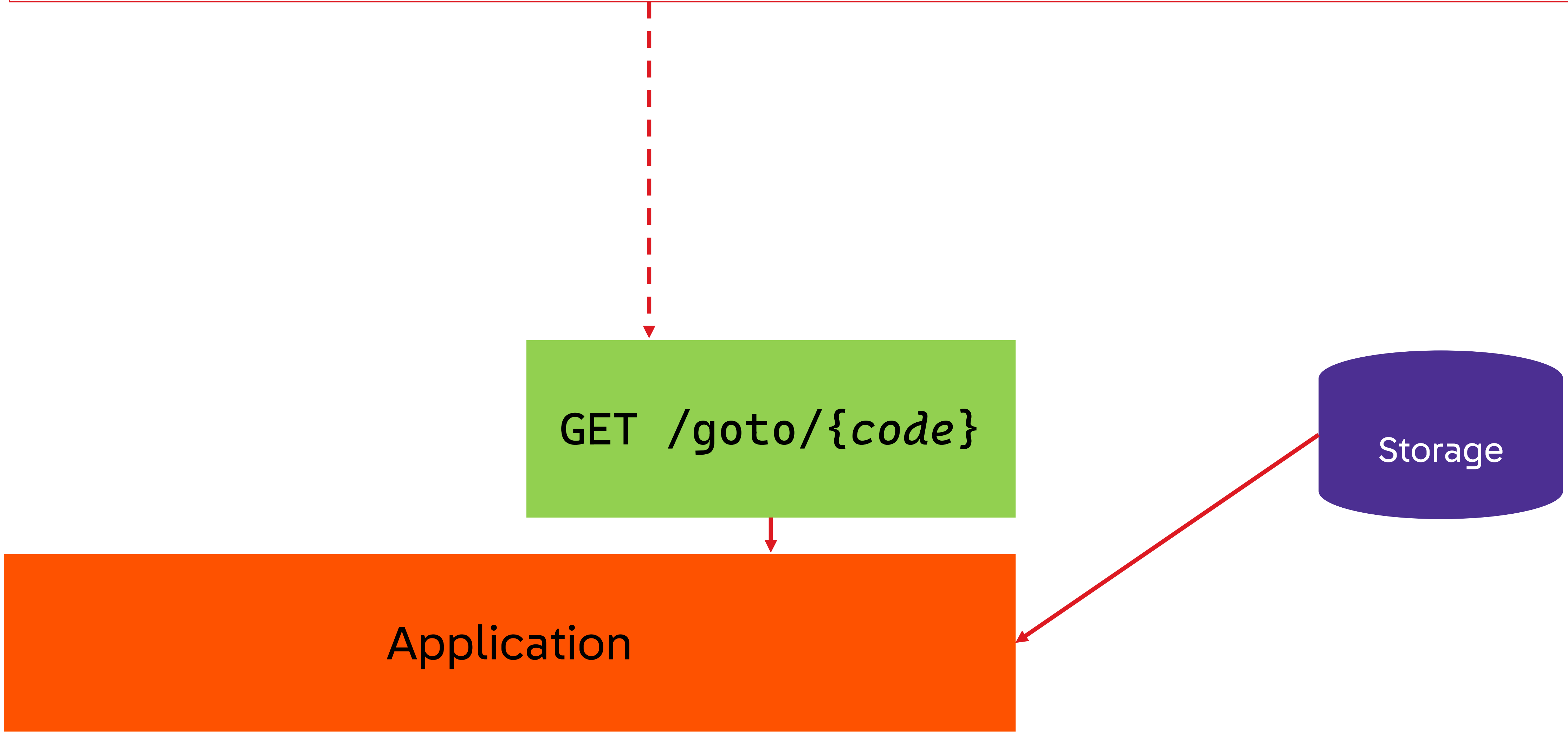
Response: 201 Created

```
HTTP/1.1 201 Created
Location: {{host}}:{{port}}/goto/falcon
 ...
{
  "id": "e5140523-d457-4961-882c-0e0477ec722b",
  "resultingUrl": "https://www.lego.com/en-dk/product/millennium-falcon-75192",
  "created": "2025-10-13T15:10:28.1728378Z"
}
```



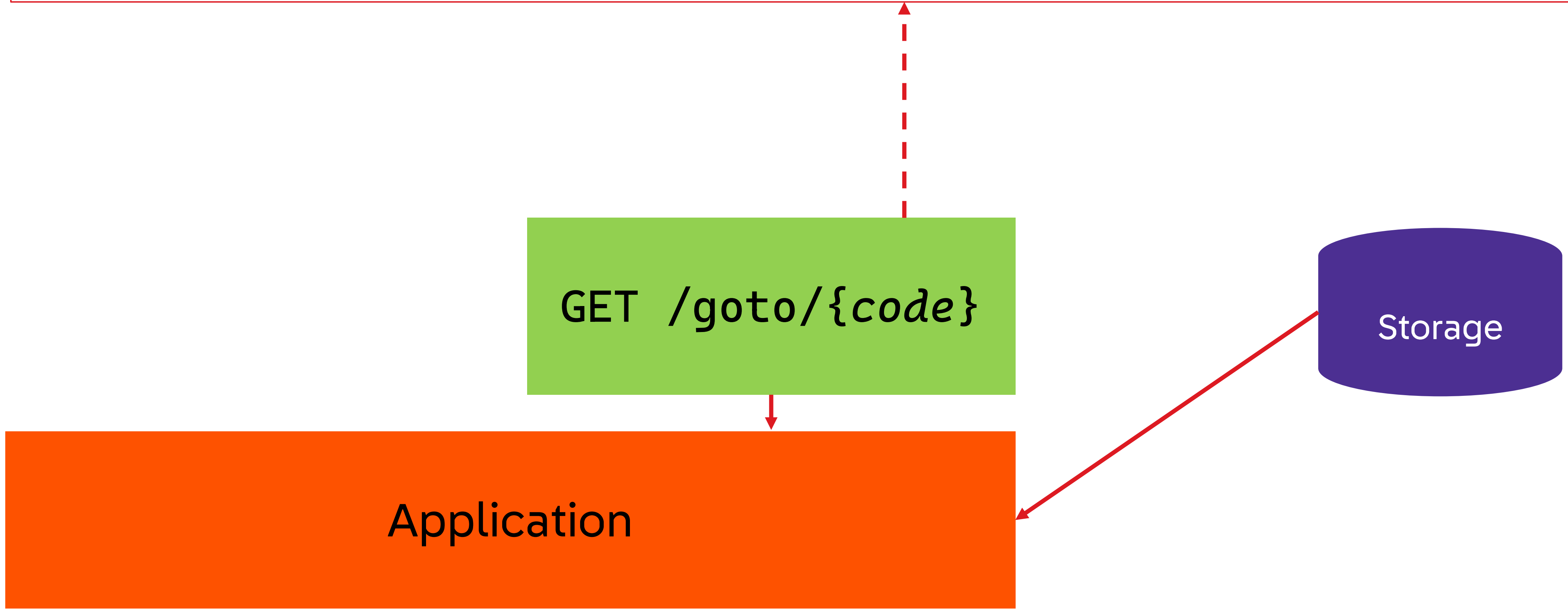
Request: `GET /goto/{code}`

```
GET {{host}}:{{port}}/goto/falcon
```



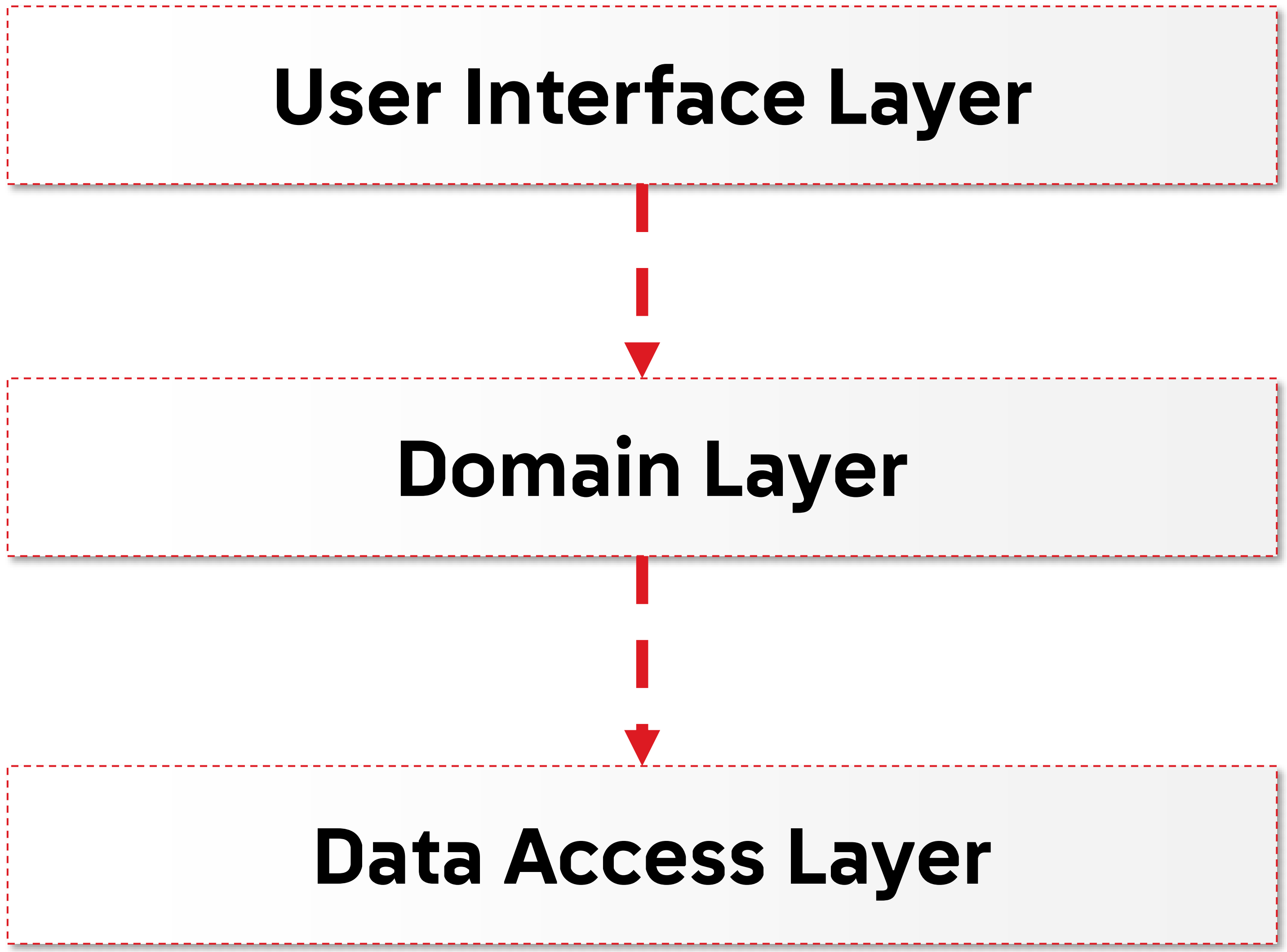
Response: `302 Found`

```
HTTP/1.1 302 Found
Location: https://www.lego.com/en-dk/product/millennium-falcon-75192
```

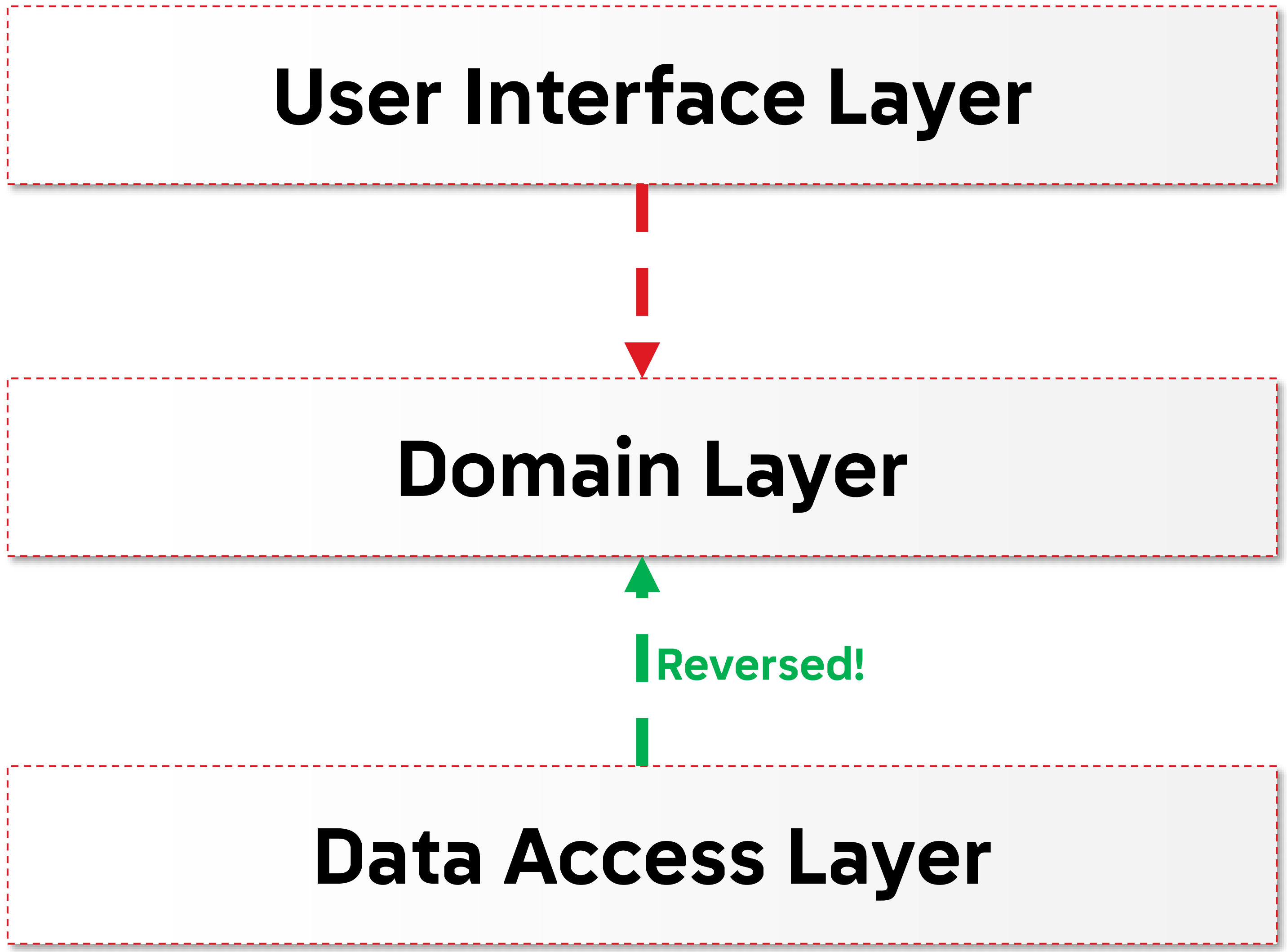


Demo: 00

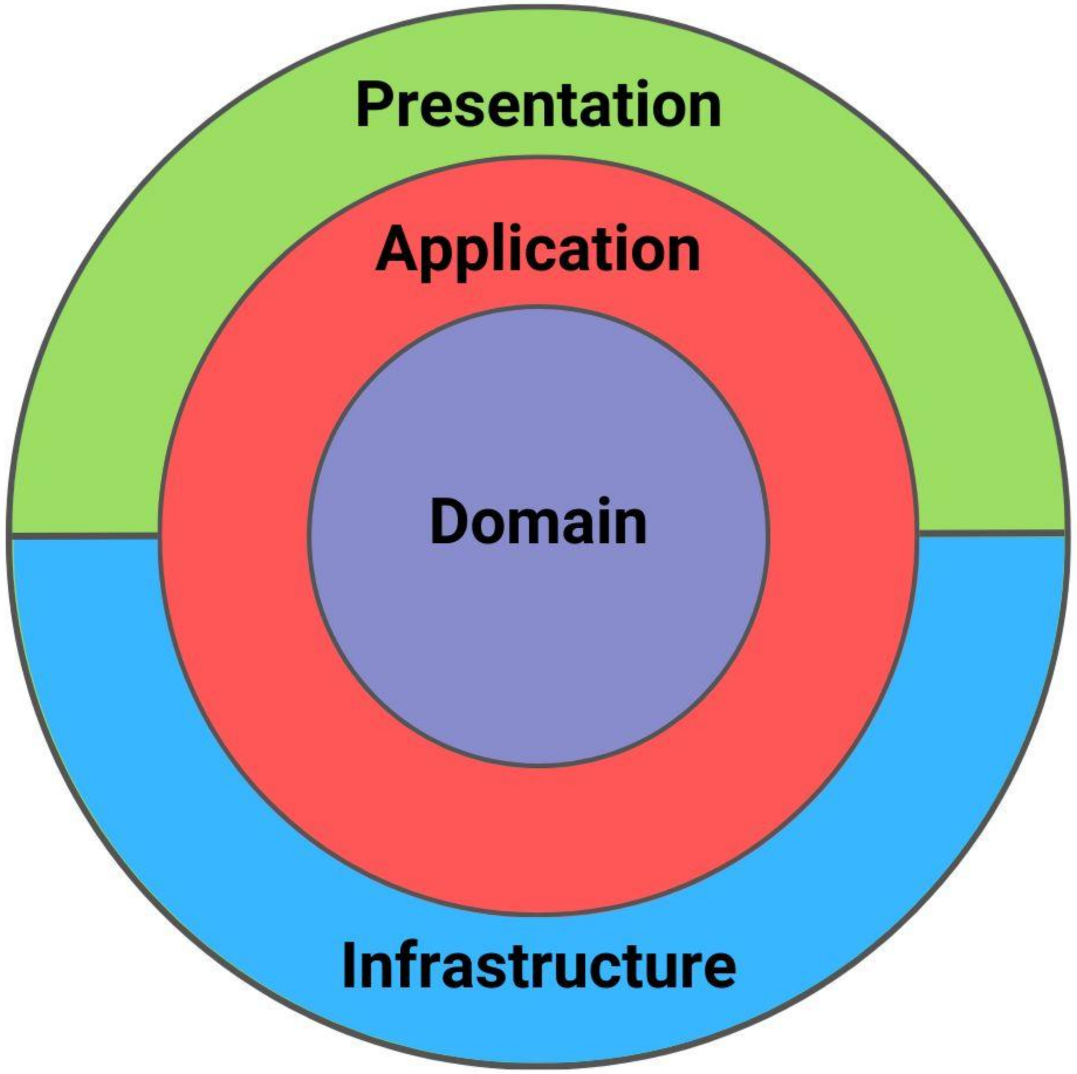
Beautiful Layered Design?



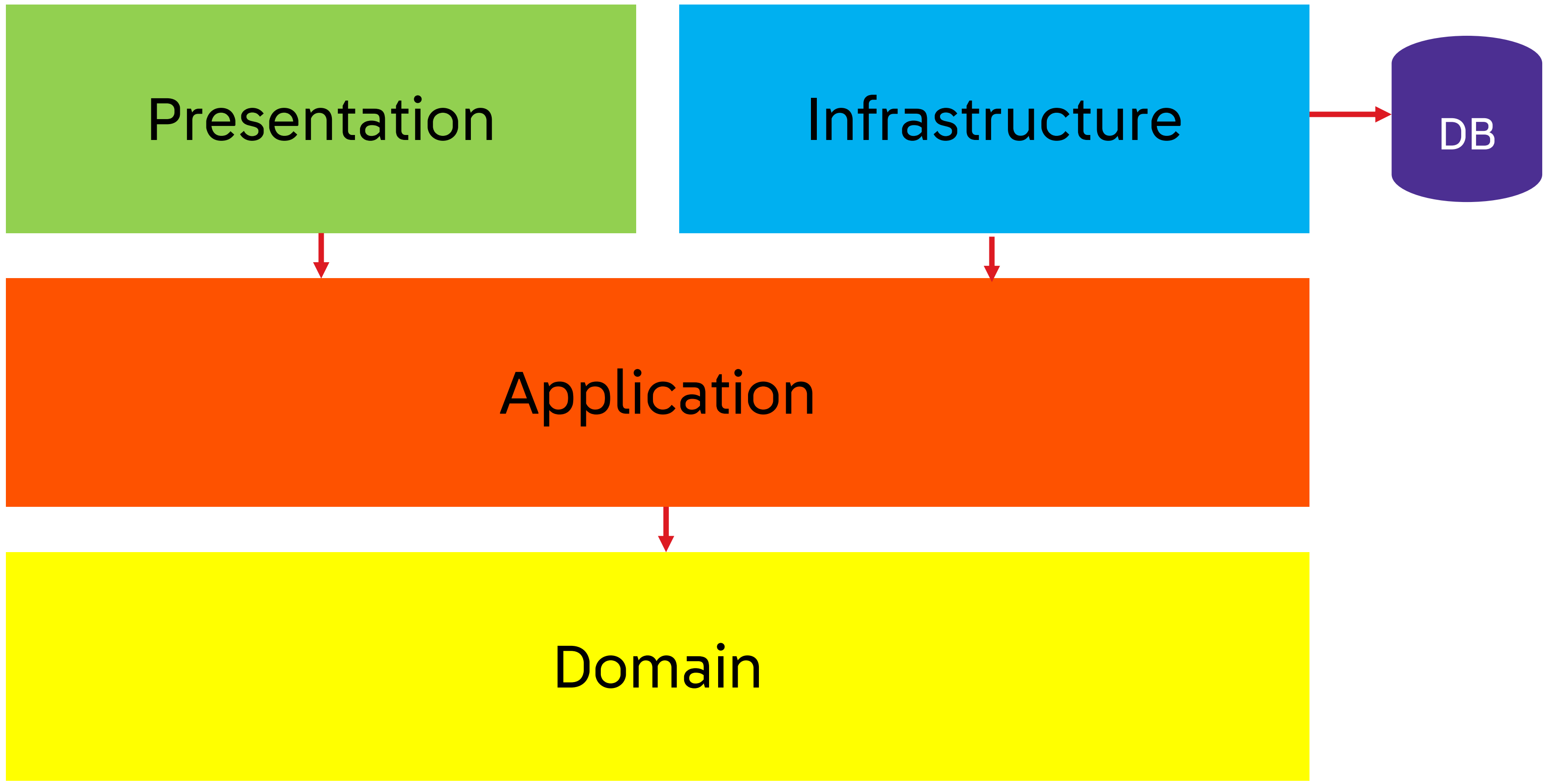
Better SOLID Design



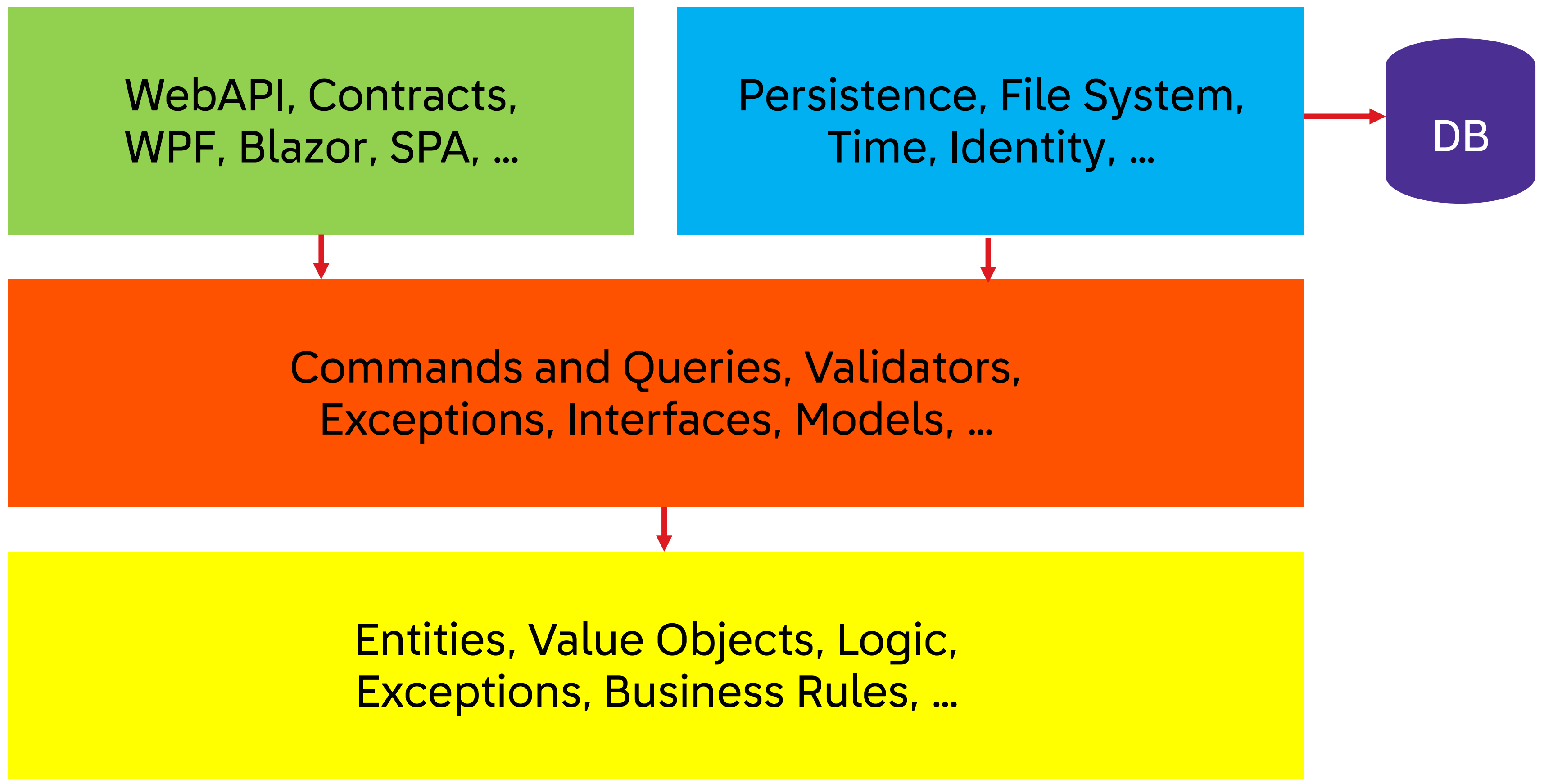
Usually Presented as...



Clean Architecture

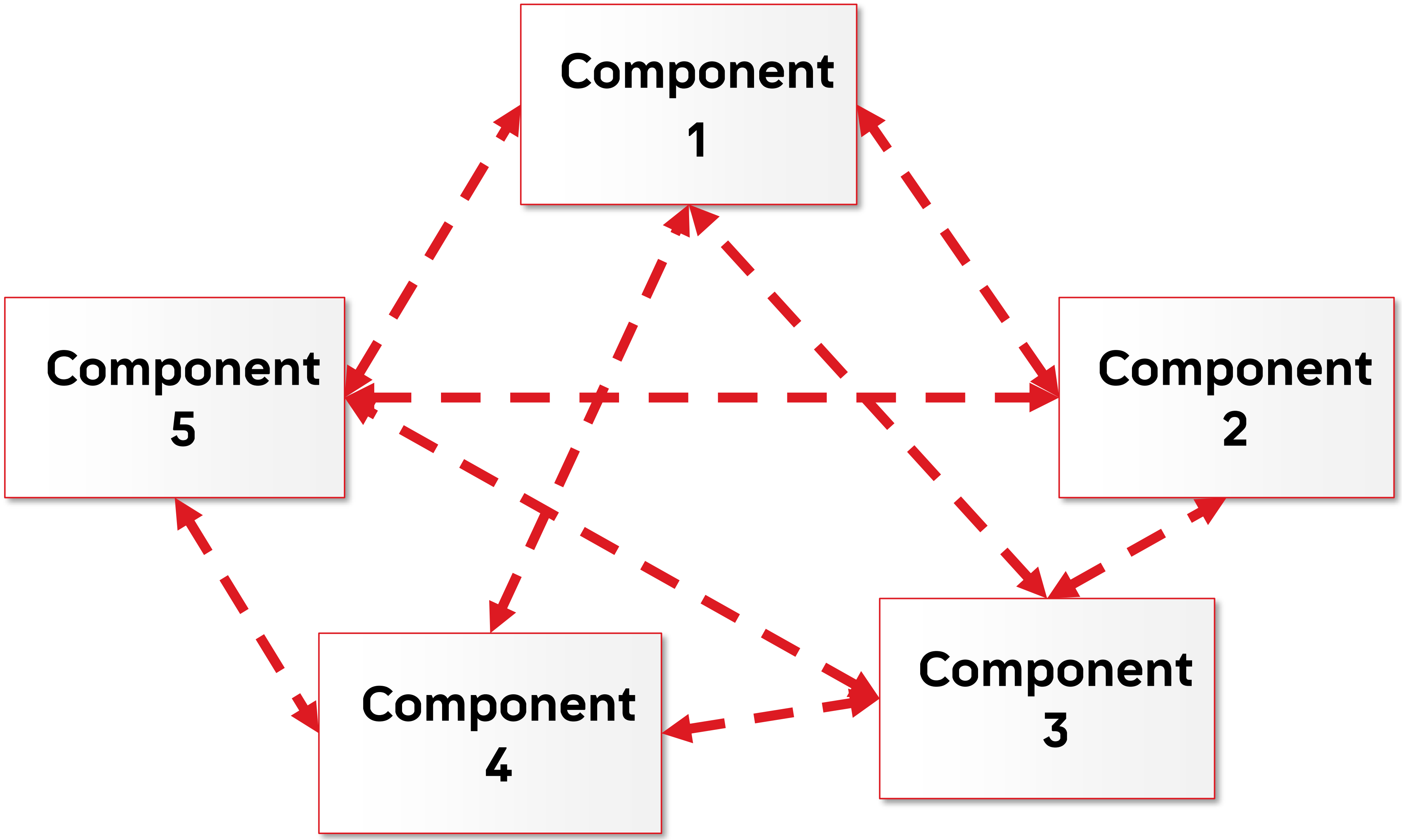


A More Detailed Look Inside

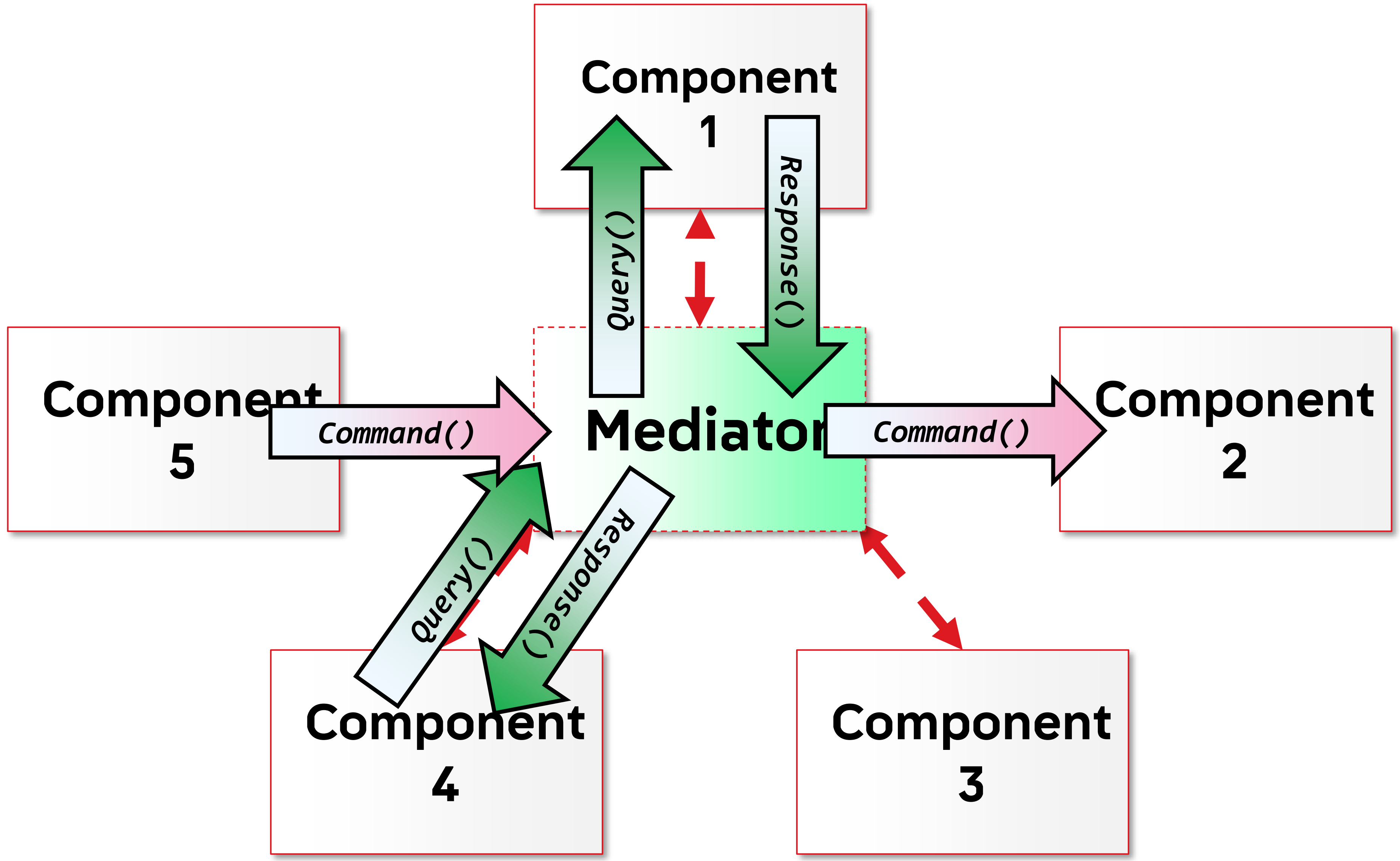

WebAPI, Contracts, WPF, Blazor, SPA, …
Persistence, File System, Time, Identity, …
DB
Commands and Queries, Validators, Exceptions, Interfaces, Models, …
Entities, Value Objects, Logic, Exceptions, Business Rules, …

Demo: 01

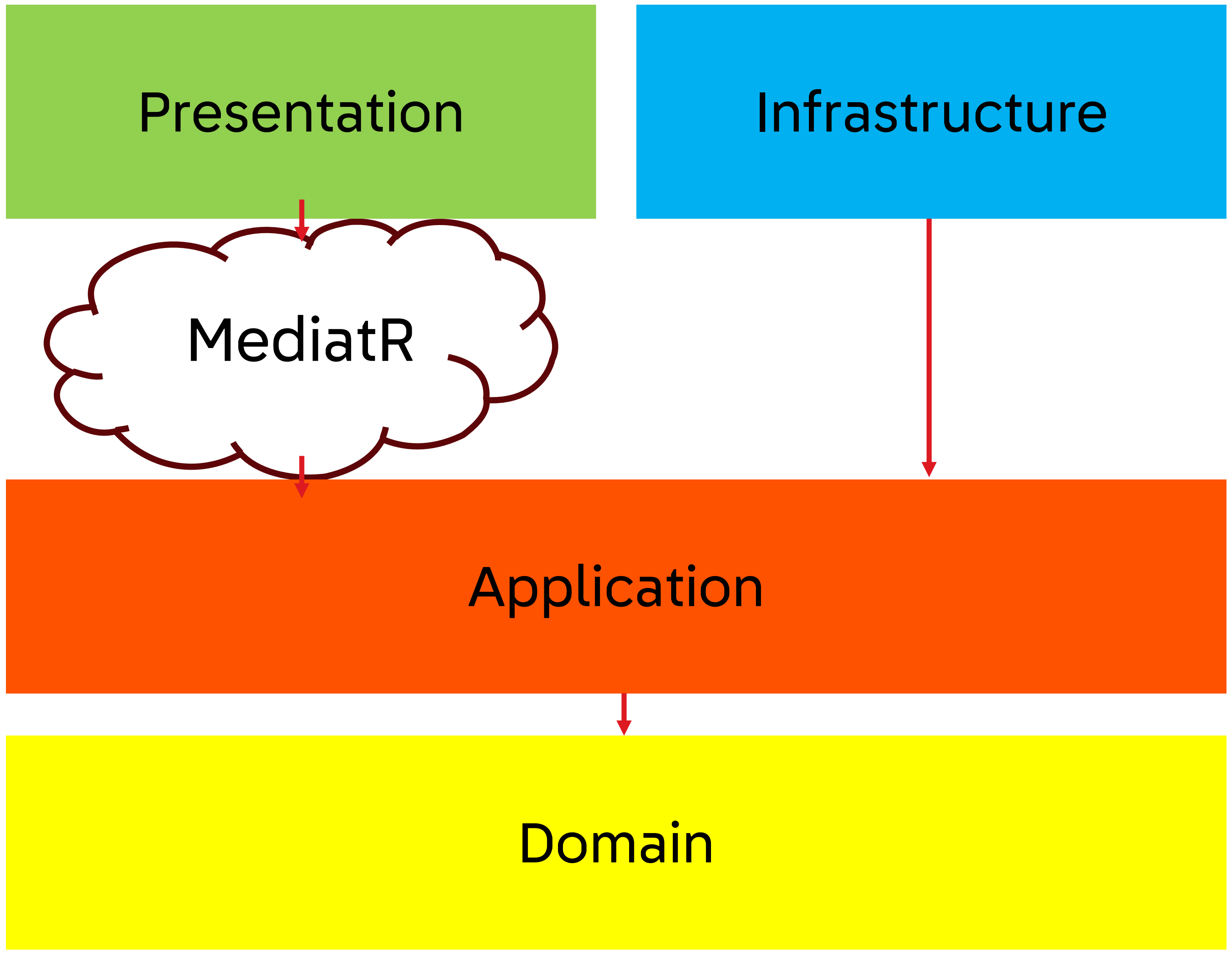
Without the Mediator Pattern



With the Mediator Pattern



Loosely Coupled by Mediator Pattern



Demo: 02 + Variations

Module 5.4 Adding Tests

Automatic Testing

- Testing frameworks
  - **xUnit**
  - MSTest
  - nUnit
  - …
- Add "xUnit Test Project" to solution
- Consider `InternalsVisibleTo` in source project file

```
<ItemGroup>
   <InternalsVisibleTo Include="Tests" />
</ItemGroup>
```

Demo: 03

Testing Rule of Thumb

- Use Arrange-Act-Assert
- Always make tests 100% deterministic!
- Consider sharing setup code
- Consider using mocking frameworks
  - NSubstitute
  - MockHttp
  - …

Testing Pyramid

- UI Tests / End-to-End Tests
- Integration Tests
- Component Tests
- Unit Tests

More recently:

- Architecture Tests

Demo: 04

Why Test Architecture?

Architecture rules and best practices are enforced by conventions and code reviews

- Tend to ”fade” over time with in/out-flux of developers
- Project complexity
- Expresses and maintains the design decisions in code

Self-testing Architecture

- Can be checked automatically in pipelines etc.

Architecture Tests

Checks could include:

- Classes and interfaces are in correct namespace and projects given their type
- Dependency rules are adhered to
- Naming conventions are obeyed
- Visibility and access modifier constraints are satisfied
- No cross-version references for endpoints

Note: This is not the same as linting..! ☺

Module 5.5 Minimal API

Newest ASP.NET Web API Project Creation

- New and more performant application structure

- Easier and lightweight
- Better compile-time support for return codes etc.

- But… Enables a “tour de force” of spaghetti code unless **very** careful!

- Patterns to use
  - Request-Endpoint-Response (REPR)
  - Elements from Plug-in Architecture

Demo: 05

Module 5.6
API Matters

The LEGO API Matters Community

- LEGO Guidelines
  https://github.com/LEGO/api-matters
- REST naming conventions
- Idempotency
- …

Summary

00	Introduction
01	Controllers
02	Middleware
03	Architecture
04	Adding Tests
05	Minimal API
06	API Matters

Thank you