

Welcome

Async / Await

Chain of Responsibility



Solution Architect
Enthusiastic Software Engineer
Microsoft Azure MVP

@danielmarbach
particular.net/blog
planetgeek.ch

Goals

target

Chain of Responsibility



```
graph LR; A[Pattern] --> B[Build it]; B --> C[WrapUp];
```

Pattern

Build it

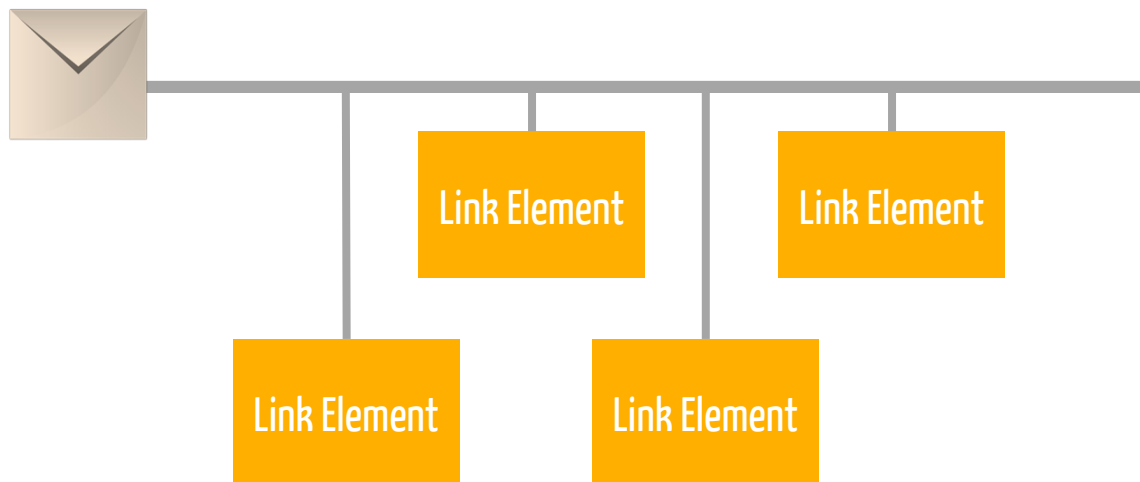
WrapUp



Pattern

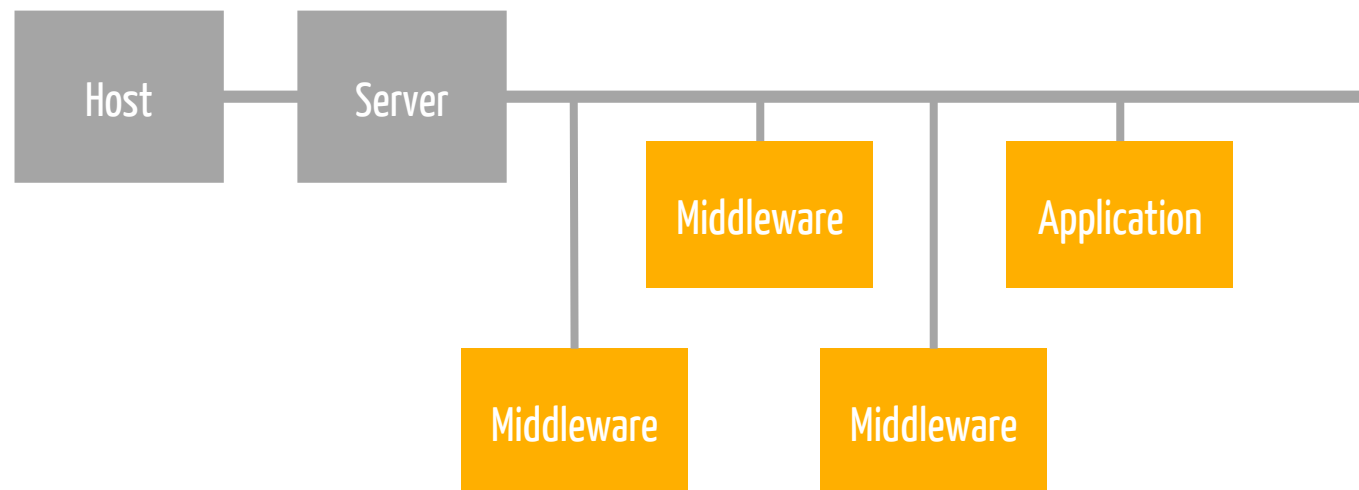
Build It

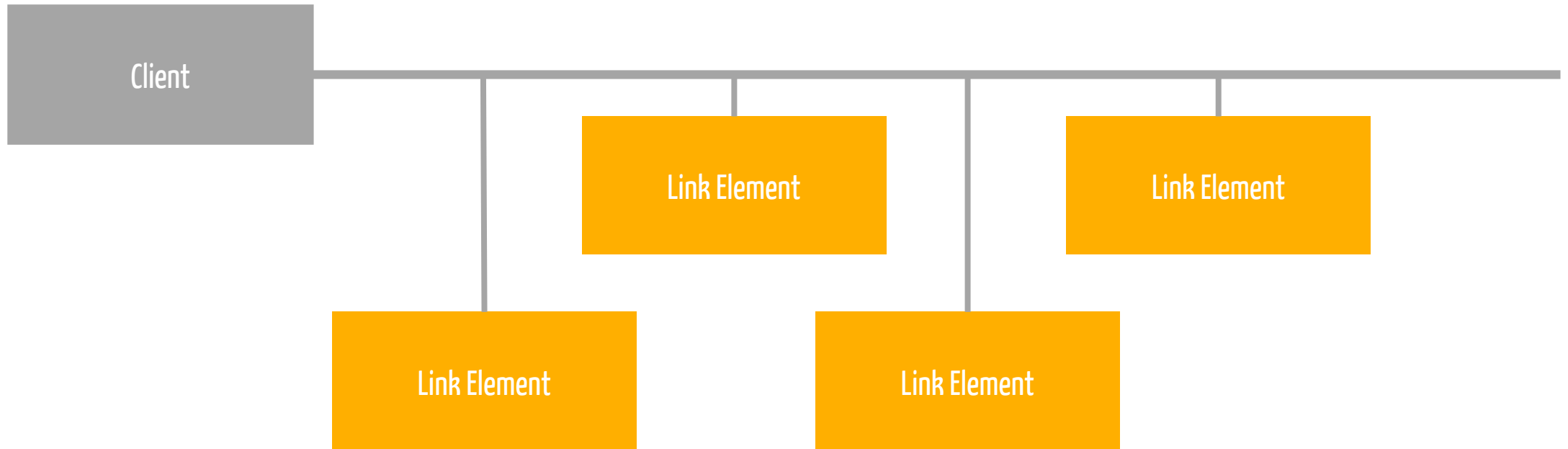
WrapUp



messaging

OWIN





```
namespace Microsoft.AspNet.Mvc.Filters
{
    public interface IAsyncActionFilter : IFilterMetadata
    {
        Task OnActionExecutionAsync(ActionExecutingContext context,
        ActionExecutionDelegate next);
    }
}
```

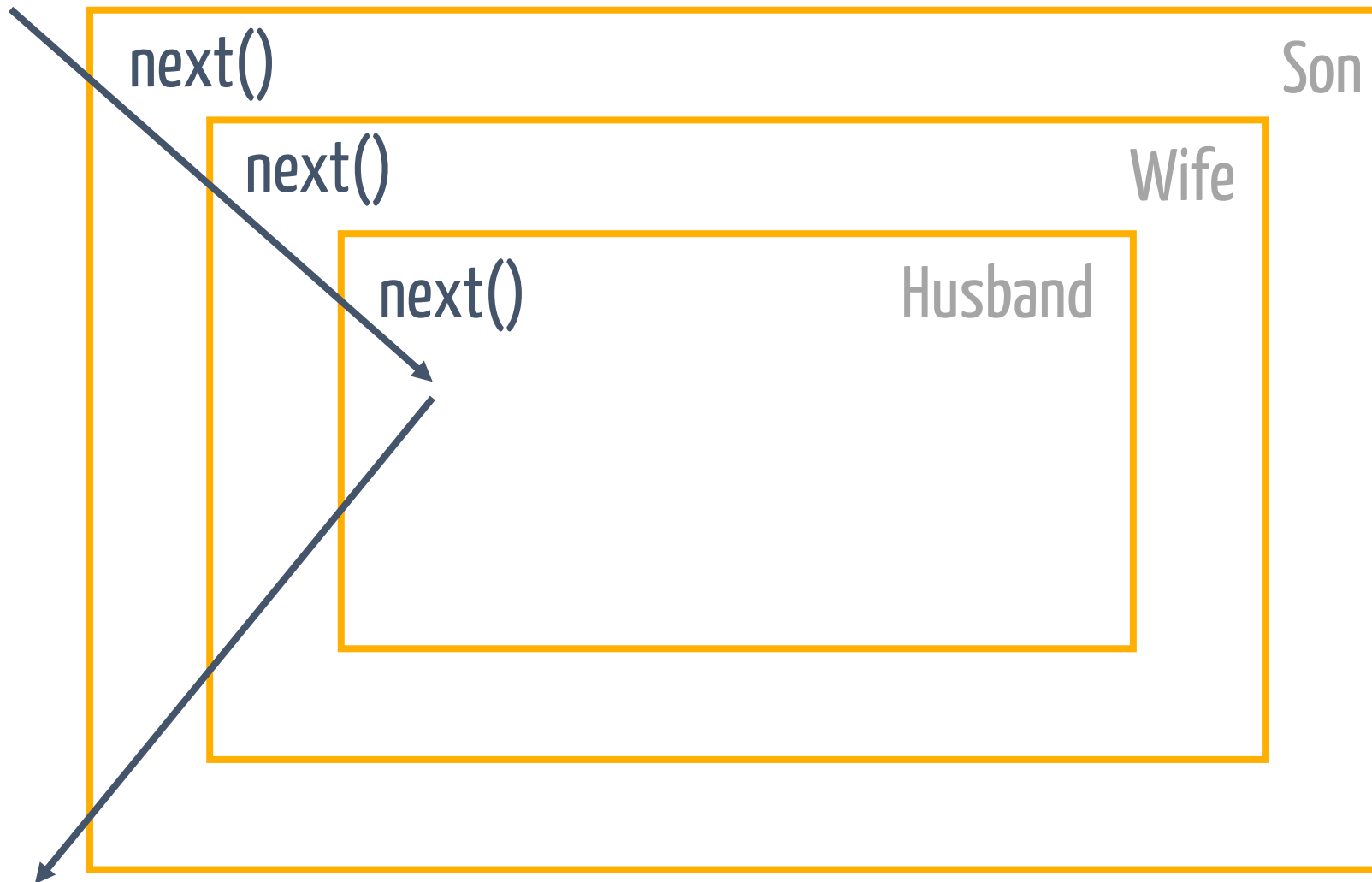



son wife husband

```
static void Person(Action next)
{
    // Implementation
    next();
}
```

```
public void ManualDishwasherUnloading()  
{  
    Son(()) => Wife(()) => Husband(()) => Done());  
}
```

await Demo

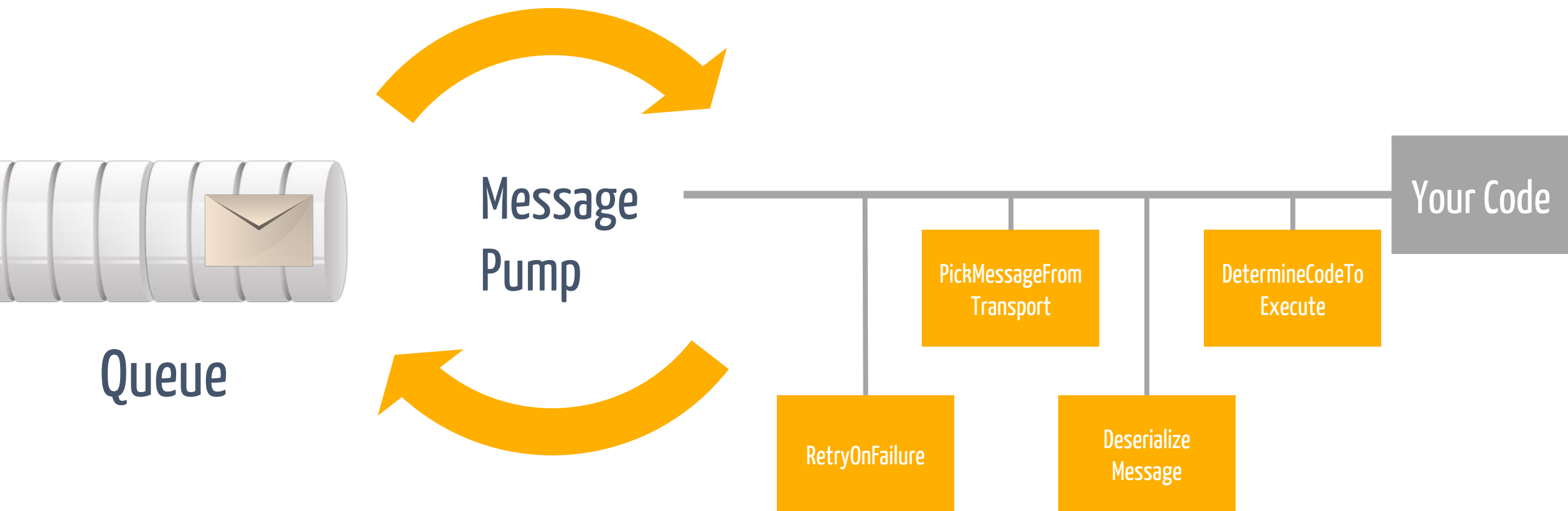


cumbersome

await Demo

```
static void IgnoreDishStillWetException(Action next)
{
    try {
        next();
    }
    catch(DishStillWetException) { }
}
```


await Demo





```
graph LR; A[Pattern] --> B[Build It]; B --> C[WrapUp];
```

Pattern

Build It

WrapUp

await Demo



Pattern

Build It

WrapUp

NSB v6

Will be **Async** all the way

Uses the **chain of responsibility** pattern heavily

particular.net/blog/async-await-its-time

docs.particular.net/nservicebus/pipeline/customizing-v6

Recap

reminder

Chain of Responsibility or Russian Dolls
is a **flexensible** pattern ideally suited
to build **robust IO bound pipelines**

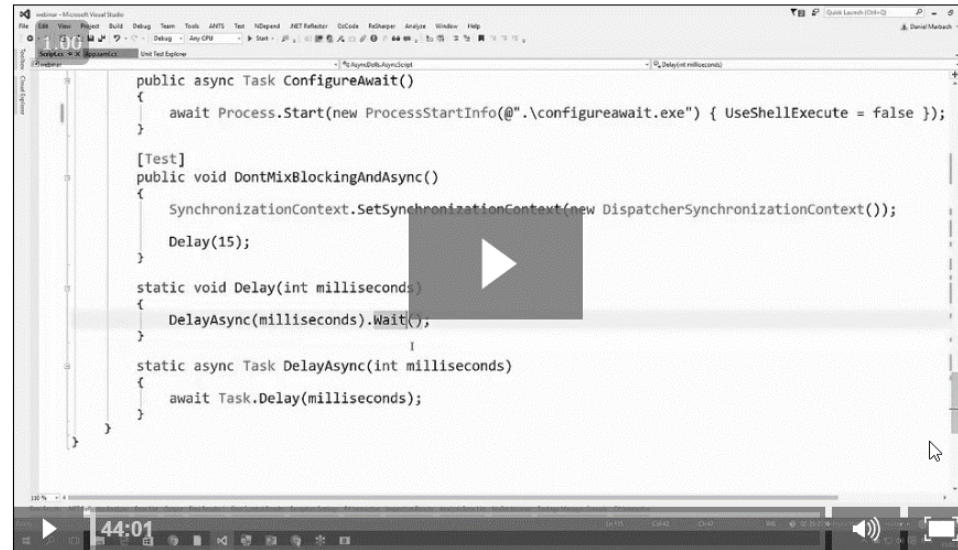
The pattern is used in **many OSS**
projects

Know it, learn it, love it *

Async/Await Webinar Series: Best Practices

See how to avoid common pitfalls in asynchronous code bases

go.particular.net/DNC016



[f](#) [G+](#) [Twitter](#) [in](#) [Share](#) [Samples](#) [Slides](#) [Comments \(0\) →](#)

Summary

Daniel Marbach shows how to avoid common pitfalls in asynchronous code bases.

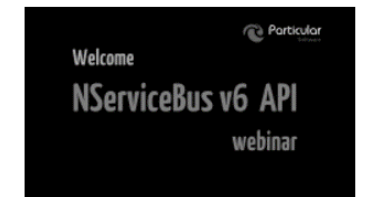
Learn how to:

- Differentiate between IO-bound vs CPU-bound work and how this relates to Threads and Tasks
- Avoid serious production bugs as a result of asynchronous methods returning void
- Opt-out from context capturing when necessary
- Deal with synchronous code in the context of asynchronous code

OTHER VIDEOS IN THE SERIES



► TPL & Message Pumps



► NServiceBus v6 API Update

Slides, Links...

github.com/danielmarbach/async-dolls

await Q & A

Thanks