Particular Software

Welcome

Rearchitect your code

towards async/await

Particular Software

Solution Architect
Enthusiastic Software Engineer
Microsoft MVP for systems integration

@danielmarbach
particular.net/blog
planetgeek.ch

Goals

target

CPU-bound vs IO-bound

Threads and Tasks

Async best-practices

Why async is the future

Premise


Terminology
Why
WrapUp

The die

is cast

javascript

ES2015

```
async function chainAnimationsPromise(elem, animations)
{
    let ret = null;
    try {
    for(const anim of animations) {
       ret = await anim(elem);
    }
    catch(e) { /* ignore and keep going */ }
    return ret;
}
```

```
$ npm install babel-plugin-syntax-async-functions
$ npm install babel-plugin-transform-async-to-generator
```

dart

release 1.9

```
runUsingAsyncAwait() async {
 //...
 var entrypoint = await findEntrypoint();
 var exitCode = await
     runExecutable(entrypoint, args);
 await flushThenExit(exitCode);
}
```

python

release 3.5

```
import asyncio

async def http_get(domain):
  reader, writer =
     await asyncio.open_connection(domain, 80)

  async for line in reader:
    print('>>>', line)
```

httpclient

```
using (var client = new HttpClient()) {
  var response = await
      client.GetAsync("api/products/1");
  if (response.IsSuccessStatusCode)
  {
     var product = await
        response.Content.ReadAsAsync<Product>();
  }
}
```

Azure SDK

```
var queryable =
client.CreateDocumentQuery<Entity>(...)
    .AsDocumentQuery();

while (queryable.HasMoreResults)
{
  foreach(var e in await
queryable.ExecuteNextAsync<Entity>())
  {
    // Iterate through entities
  }
}
```

async

event-driven

Task

uniform

Task

IO-bound

Task

CPU-bound

concurrent

concurrent

concurrent

concurrent

concurrent

interleaved

parallel

parallel

simultaneous

Continuation

function

```
await cleanLaundry;
await dryLaundry;
```

Recap

best-practices

Use async Task instead of async void

Async all the way, don't mix blocking and asynchronous code

Async / await

is viral

but

It kicks your

servers

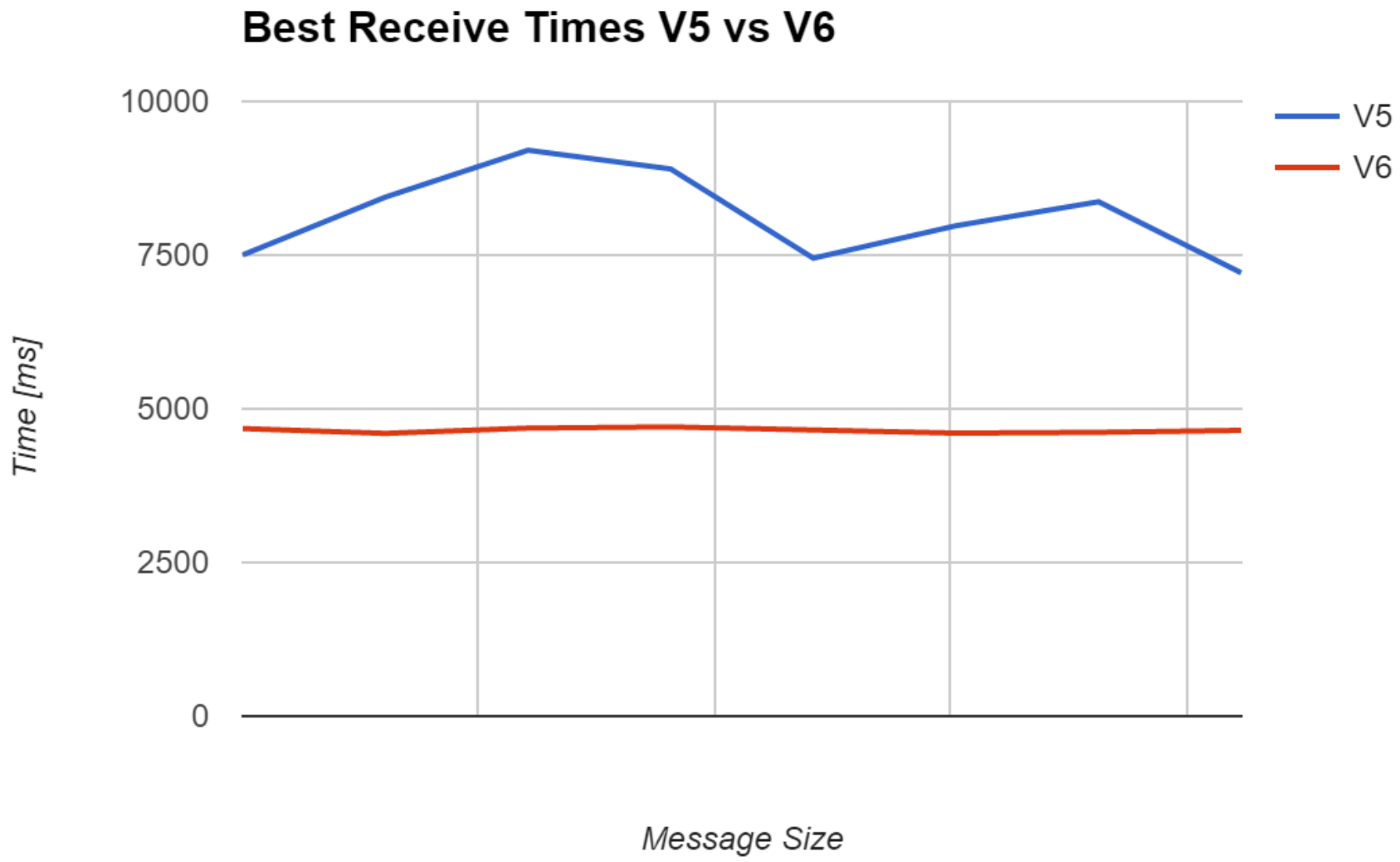
butt

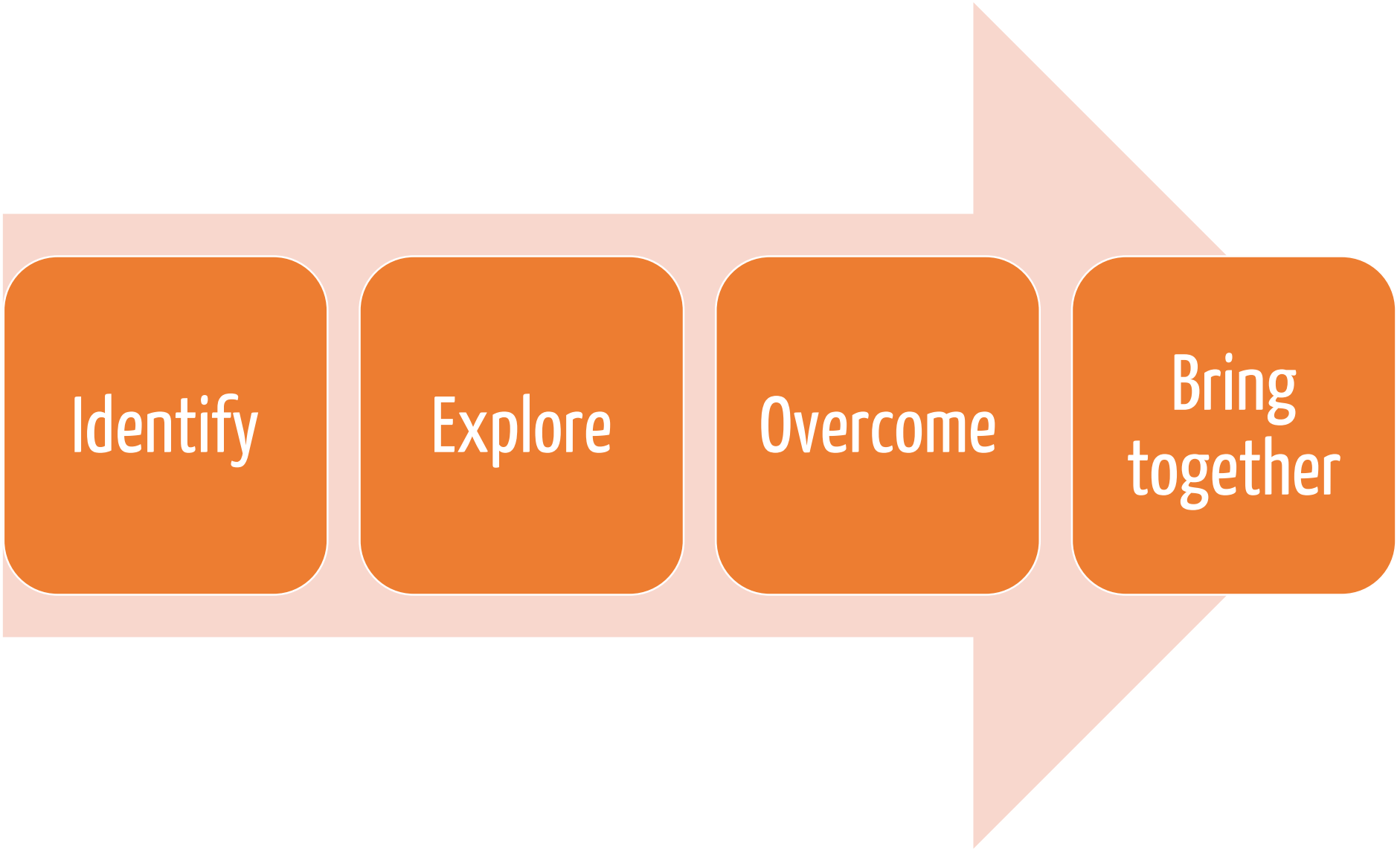
NServiceBus

Azure Service Bus	26 times
Azure Storage Queues	6 times
MSMQ	3 times

more message throughput

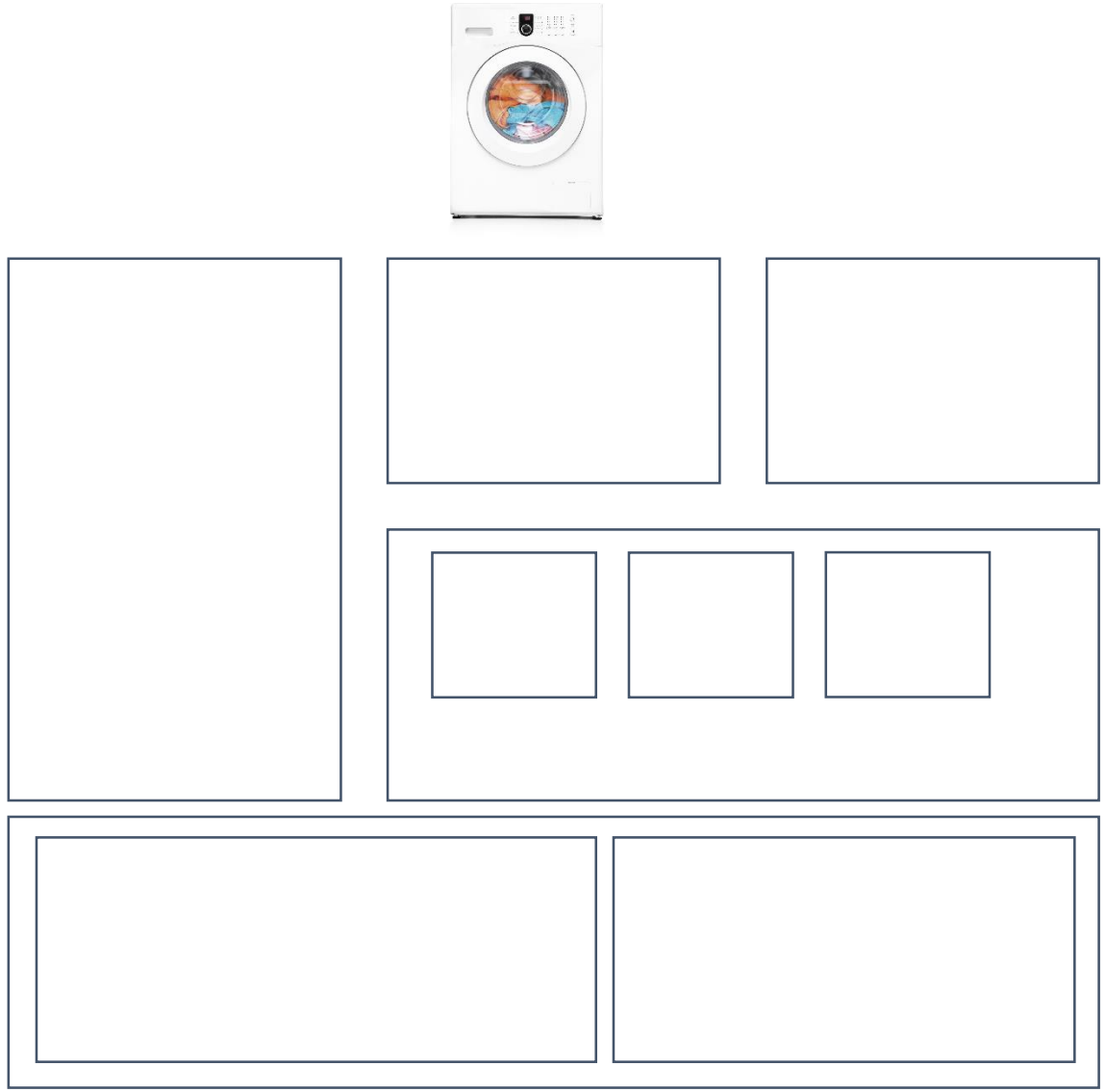
NServiceBus.SqlServer




Identify
Explore
Overcome
Bring together

Identify

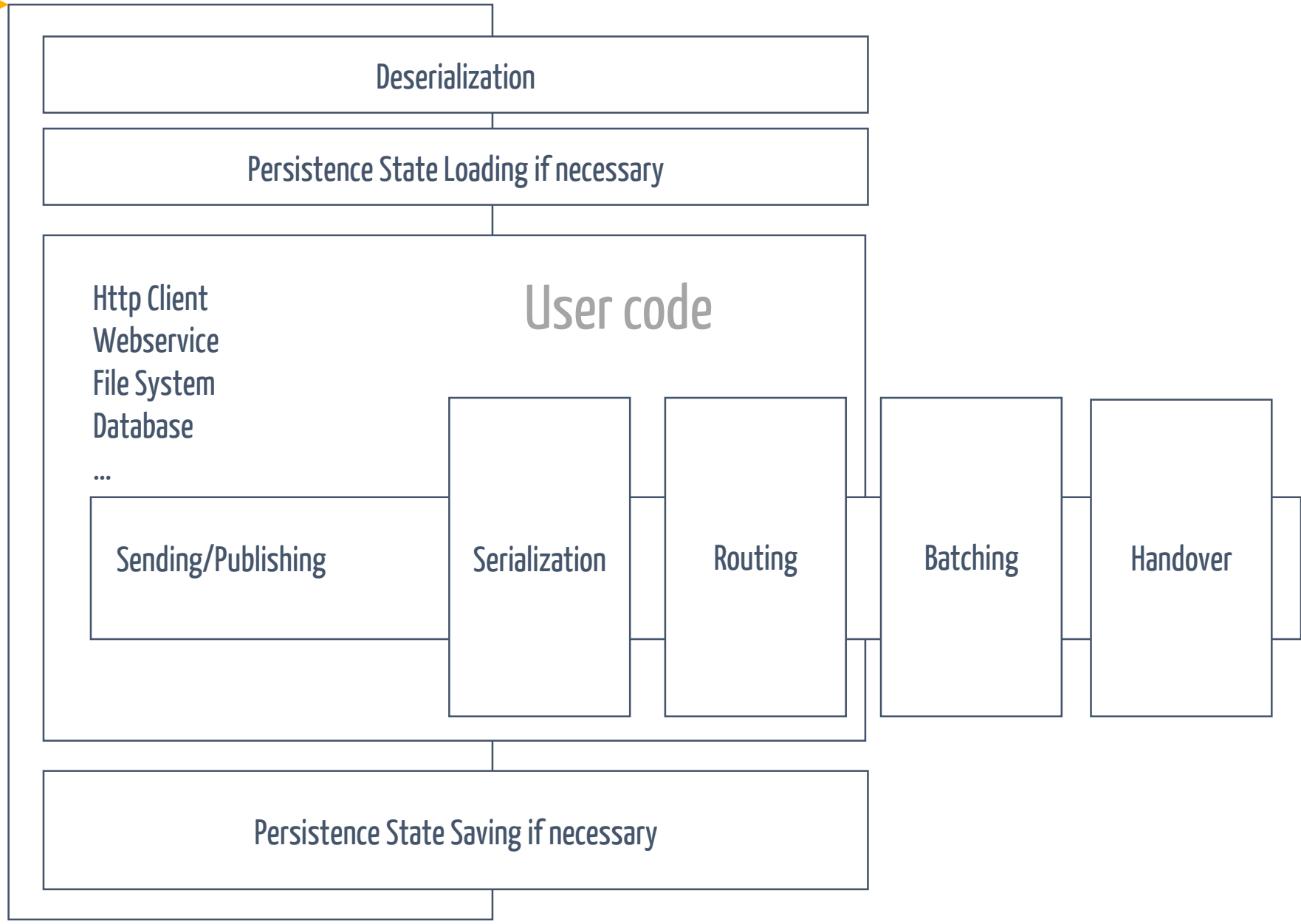
IO-bound



Queue

NServiceBus

IO-bound



Task.Run
Task.Factory.StartNew
Parallel.For
Parallel.ForEach

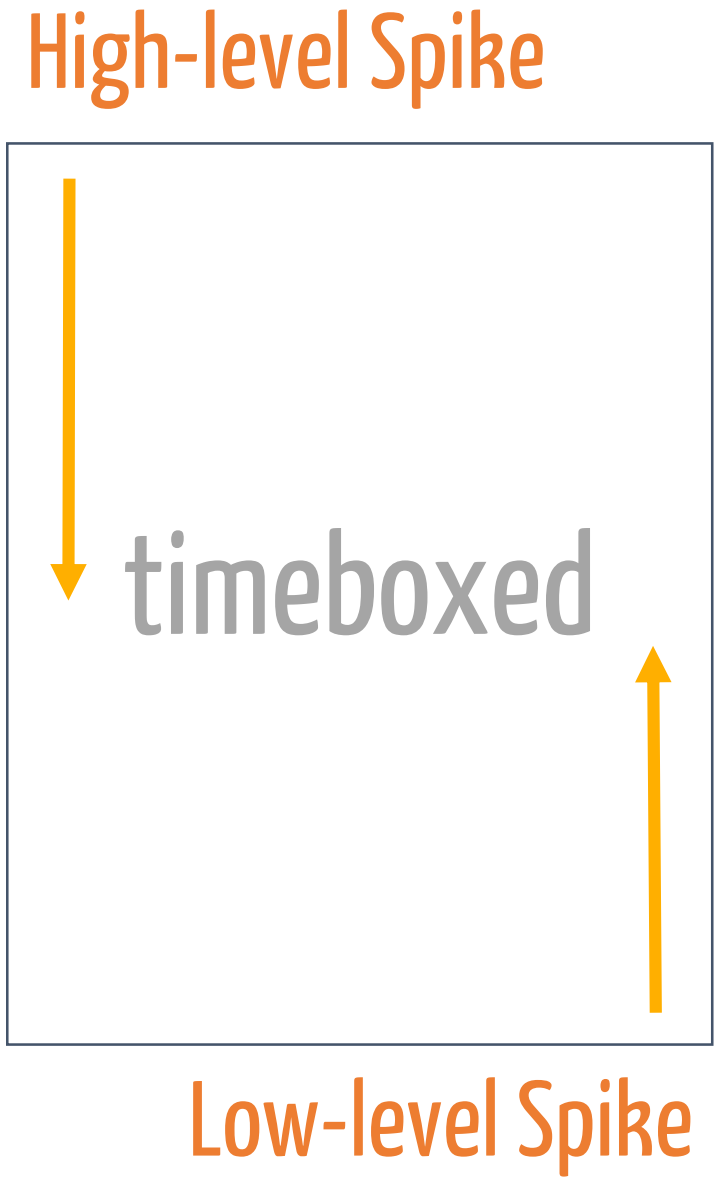
Worker Threadpool

IO Threadpool

await iobound
iobound.FireForget()

Explore

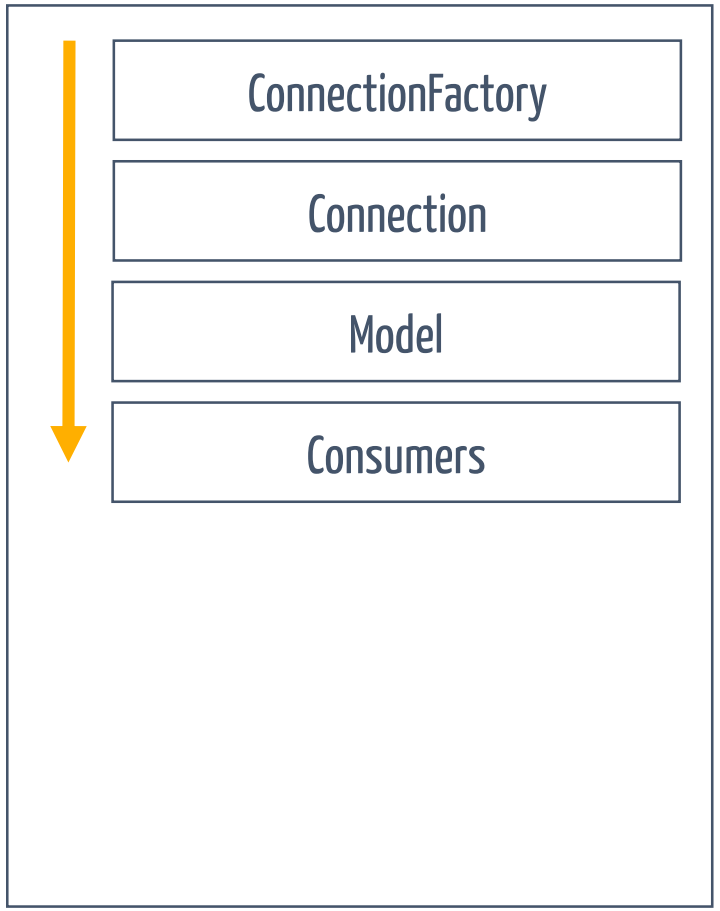
IO-bound



RabbitMQ Client

IO-bound

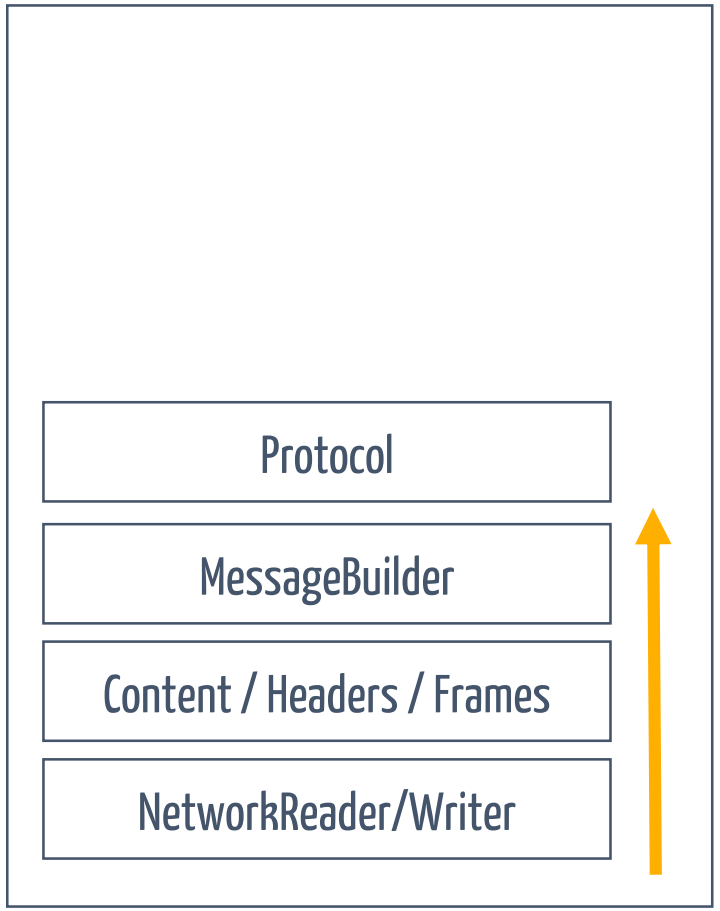
High-level Spike



github.com/rabbitmq/rabbitmq-dotnet-client/pull/151

RabbitMQ Client

IO-bound



Low-level Spike

github.com/rabbitmq/rabbitmq-dotnet-client/pull/149

Event handlers
Locks
Monitor
Semaphore / Mutex
Auto/ManualResetEvent
Thread

void stinks wait smells

Remember

Async all the way means avoid blocking code

Forget thread!

think Task

Terminology

Why

WrapUp

Recap

reminder

Use Task.Run, Factory.StartNew for CPU-bound work

Use Task directly for IO-bound work

Use async Task instead of async void

Recap

reminder

Libraries and frameworks should use ConfigureAwait(false)

Async all the way, don't mix blocking and asynchronous code

Slides, Links...

github.com/danielmarbach/RearchitectTowardsAsyncAwait

Particular Software

await Q & A

Particular
Software

Thanks