

Introduktion til distribuerede systemer i .NET med Rebus



Trifork g33x0r Night 3. september 2013

Mogens Heller Grabe

d60

mhg@d60.dk

<http://mookid.dk/oncode>

@mookid8000

Hvad?

- Hvad er dit problem?
- Mød Rebus
- Eksempler
- Afslutning
- Mere afslutning (g33k bonus)

Hvordan?

- Slides
- Snak
- 4 demoer
- Bonus

**Hvad er dit
problem?**

Stort system
størrelse

Integration med eksterne parter

(hallå? er der nogen?)

Kompleks logik

Ting, som får ting til at ske...

...der får ting til at ske...

...der får ting til at ske...

...og så går der noget tid...

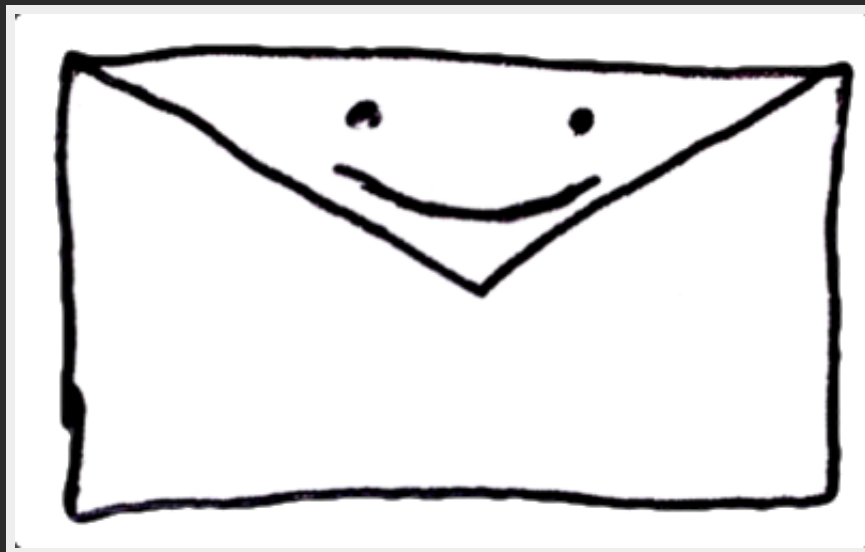
...og så sker der noget mere!

(kender I det?)

Opsummering af problemer

- Monolit
- Integration
- Koordinering

Løsningen: Messaging



(og med "løsningEN" mener jeg "EN løsning")

Messaging via durable, asynchronous køer



Windows FTW!!!

Mød Rebus



"Service bus"?



"Hippie bus"?



Seriøst - hvad er det?

- Messaging library
- Et lag ovenpå MSMQ
- En enkelt .NET 4 DLL
- Flere DLL'er hvis du vil have
 - RabbitMQ
 - Azure Service Bus
 - RavenDB
 - MongoDB
 - Castle Windsor
 - StructureMap
 - Unity
 - Autofac
 - Ninject
 - Log4net
 - NLog
 - ...

Motivation

- Jeg kan godt lide NServiceBus
- Men jeg kan ikke bruge NServiceBus alle de steder, hvor jeg har lyst
- Nogle gange har jeg kæmpet lidt med NServiceBus
- Ville gerne bruge MassTransit
- Ville fork'e NServiceBus da den var Apache V2

Filosofi

- Gratis
- Nem

Meta

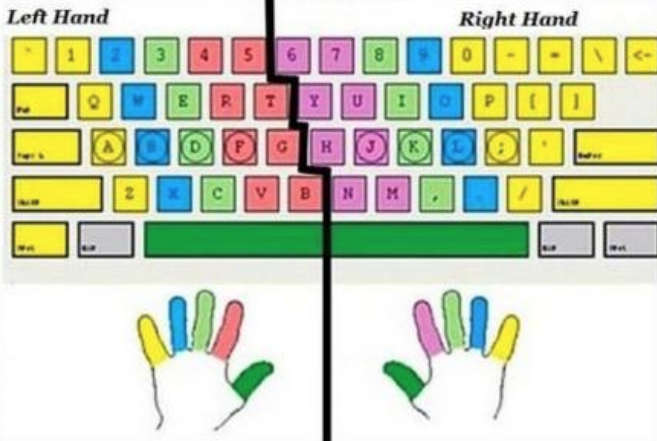
- 3700 linjer C# 4
- Koden er på GitHub:
<https://github.com/mookid8000/Rebus>
- Har bidrag fra 11 udviklere ud over mig
- Man får fat i Rebus via NuGet
<http://nuget.org/packages?q=rebus>
- Aktuelle version: 0.39.0
- Har flyttet penge rundt siden 0.14-alpha
- Har styret kraftværker siden 0.17-alpha

Godt at vide

- Beskeder er POCOs
- Hver service/endpoint/proces har sin egen input-kø
- Hver beskedtype ejes af 1 logisk service

Demo 0

How to Type:



How I Type



Eksempler

De 3 problemer, jeg snakkede om

1. System, der er ved at blive for stort
2. Integration med eksterne parter
3. Kompleks logik med koordinering og timing

1. problem

System, der er ved at bliver for stort

Resumé: Vi er ved at bygge en trading-platform, hvor traderne i "front office" slår handler med modparter og bogfører disse, mens administrativt personale i "back office" sikrer at modparterne bliver opkrævet.

Stikord

- ubiquitous language
- bounded context
- anti-corruption layer
- distributed domain-driven design

Demo 1

Adskil i separate Trading og Billing systemer

2. problem

Integration med ekstern part

Resumé: Når nye handler laves, skal "middle office" godkende disse - f.eks. afhængig af modpartens kreditstatus... Kreditstatus kan hentes ved at forespørge på SOAP-servicen "CreditAssessment", hvilket gøres i vores Confirmations-service.

Stikord

- asynchronous
- reliable
- automatic retries

Demo 2

Web service-forspørgsel via messaging-facade

3. problem

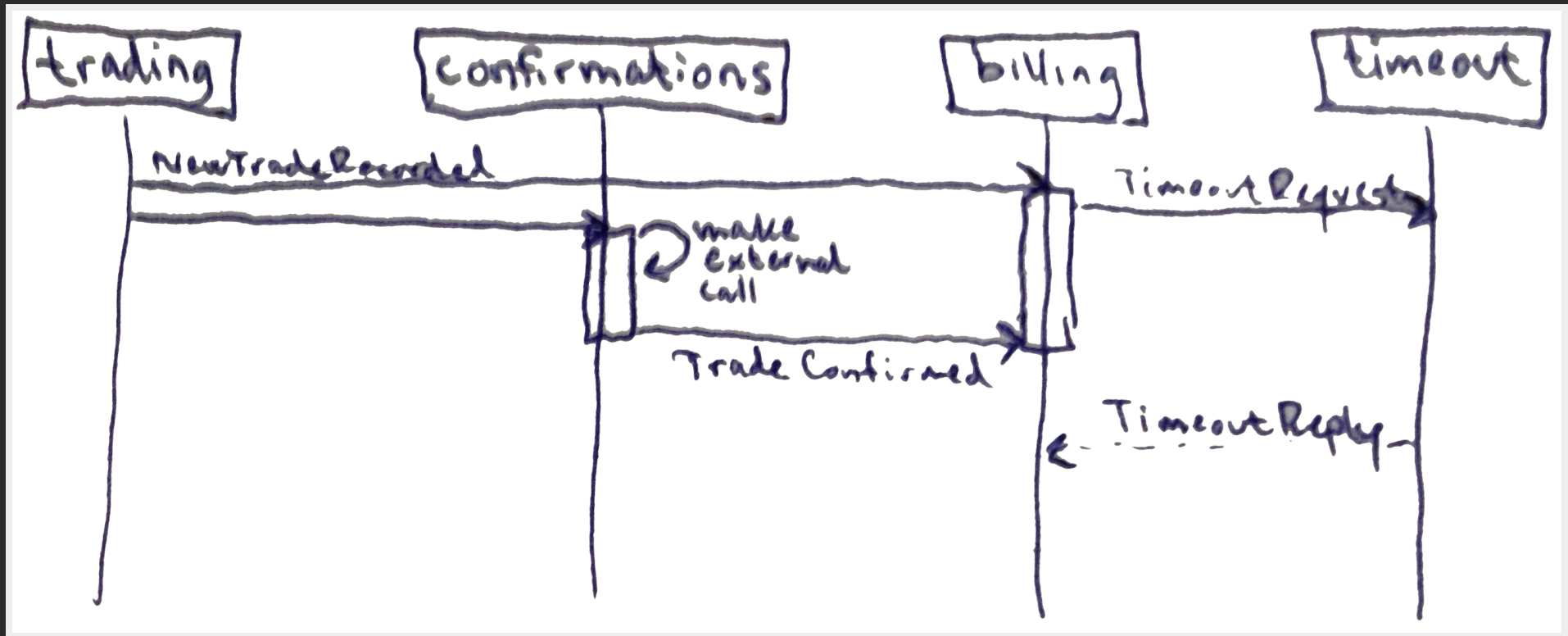
Kompleks koordinering og timing

Billing skal fakturere kunden, men denne proces kan optimeres ved at sende 1 stor samlet faktura såfremt kreditstatus ellers er OK. Derfor, når handler laves, vil Billing afvente dommen fra Confirmations for at afgøre hvilke faktureringsstrategi der skal benyttes.

For at undgå at "glemme" at sende fakturaer i tilfælde af at noget glipper, vil vi gerne gøre *noget* hvis faktureringsprocessen ikke er overstået inden for 10 sekunder.

Stikord

- process manager
- timeouts
- compensating actions



Demo 3

Kompleks logik, koordinering, timing



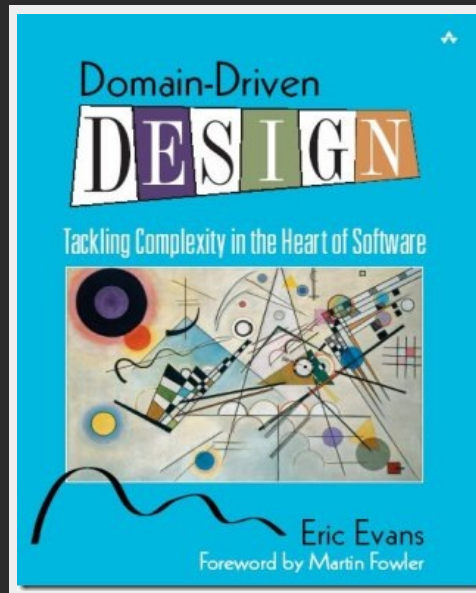
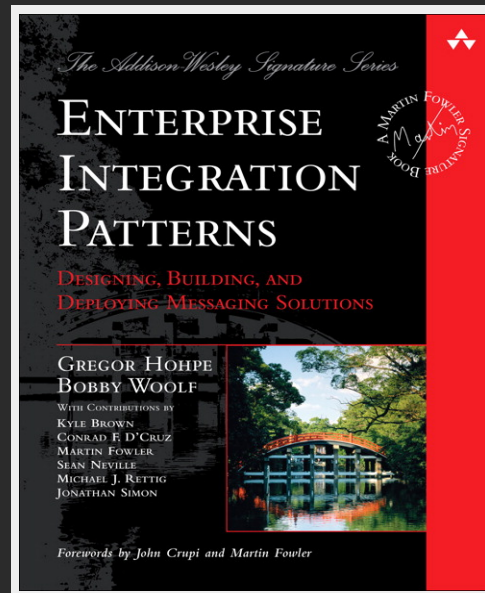
Ting, jeg ikke har vist

- Kan også bruge RabbitMQ, Azure Service Bus og SQL Server som transports
- Kan gemme subscriptions og sagaer i SQL Server, RavenDB og MongoDB
- Kan aktivere handlers med Castle Windsor, StructureMap, Autofac, Ninject og Unity
- Kan logge med NLog og Log4Net
- Kan sende beskeder i batches
- Kan lave handler pipeline re-ordering
- Kan lave polymorfisk dispatch
- Kan kryptere message bodies

Hvad så nu?

- 1.0
- Central monitorering
- HTTP gateway
- Flere transports (e.g. Apache ActiveMQ)
- Distributør
- ?

Litteratur



Vil du læse mere?

Tjek Udi Dahan, Greg Young, Dan North og Rinad Abdullin



Mere afslutning

Correlation IDs

Mere afslutning

Autentificering vha. polymorfisk dispatch og pipeline re-ordering

Mere afslutning

Transports

Mere afslutning

Kryptering/komprimering

Tak fordi I lyttede!

...og en stor tak til **Hakim** for at have lavet det fantastiske **reveal.js**

Mogens Heller Grabe

d60

mhg@d60.dk

@mookid8000

<http://mookid.dk/opcode>