

# DESIGNING FOR RAPID RELEASE

**Sam Newman**  
*ThoughtWorks*



Not Him



**GOTO: Aarhus 2012**

**@samnewman**

# ThoughtWorks®

**GOTO: Aarhus 2012**

**@samnewman**



[Overview](#) | [My Work](#) | [Planning](#)

Group by: Status

Rank cards

**Ready for Development**  
(8)

Subscribe #86  
to RSS  
feed

Merge #85  
contacts

Export my #84  
contacts to  
iCal

Debug your te  
with real-time

## Pipelines » Go » 2.0.0.5082

dev dist smoke-firefox dist-all dist-sol

Run: 1 of 1 Passed | Triggered: by dave | Duration: 00:06:02

[Overview](#) | [Pipeline Dependencies](#) | [Materials](#) | [Jobs](#) | [Tests](#)

UPSTREAM

Mercurial  
trunk - 18e5db65024e

Git  
twist - ef2b2e0

CURRENT

go

Label: 2.0.0.5082

DOWNSTREAM

deploy-personal-build-server

2.0.0.5082

acceptance

2.0.0.2.0.0.5082

Reuse existing workflows and optimize your test suites with inline content-assist.

**We're hiring...**

**GOTO: Aarhus 2012**

**@samnewman**

# Why is rapid release important?

Scaling

Performance

Compliance

Durability

What criteria influence  
design?

Geographical

Technology

What about making it  
easy to release?



Make it quick to make a change

Make it quick to deploy the  
change

Reduce the risk of releasing

# MusikShop v2

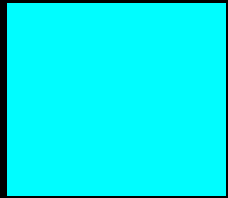
v23

v2

v9

v14

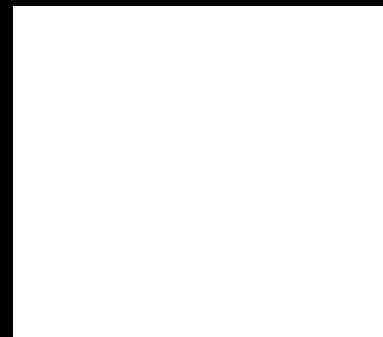
MusikShop



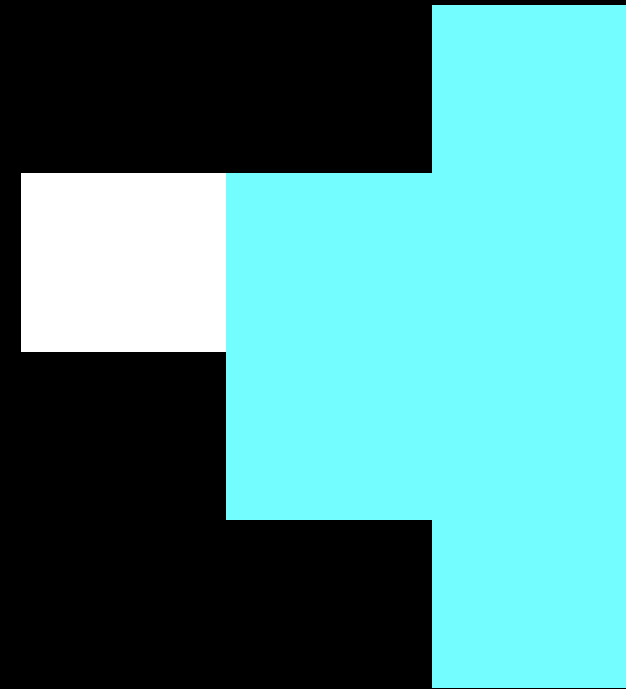
The Same Change...



...resulting in very different  
deployments







Each release is a rollback point

and a data gathering point



and a release process  
practiced often is more likely  
to work and to be efficient

**Small, Incremental  
Releases FTW!**

So how do we design  
our systems to allow  
for small changes?

MusikShop

Rekomend  
Lib

Invoice  
Creation

...

String Utils

Small units - easier to  
reason about

Fast CI turnaround  
cycles



MusikShop v1

Rekomend  
Lib v134

MusikShop v2

Rekomend  
Lib v200

Change to a statically linked library =  
new deployment of the application

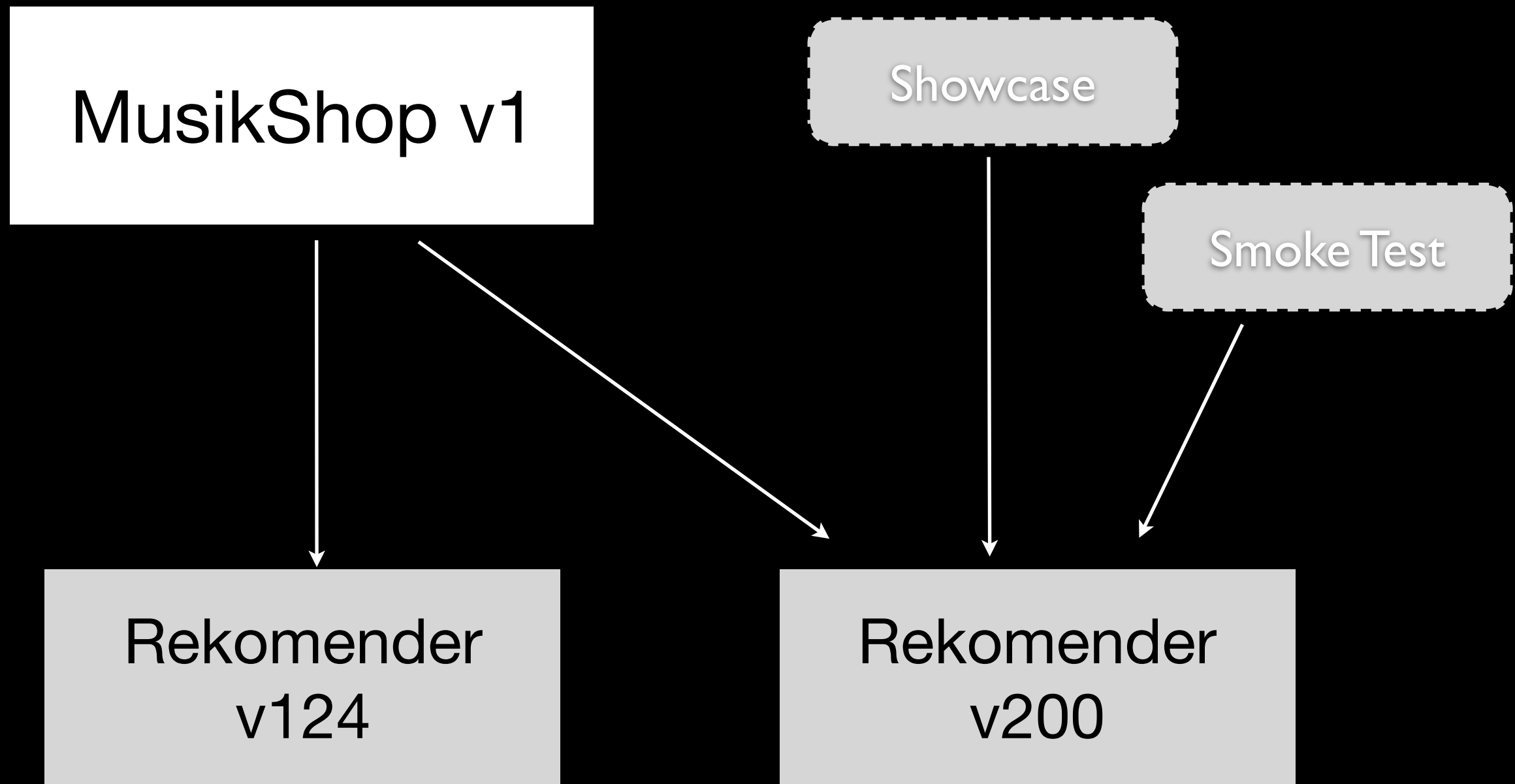
MusikShop v1

```
graph TD; A[MusikShop v1] --> B[Rekomender v124]; A --> C[Rekomender v200];
```

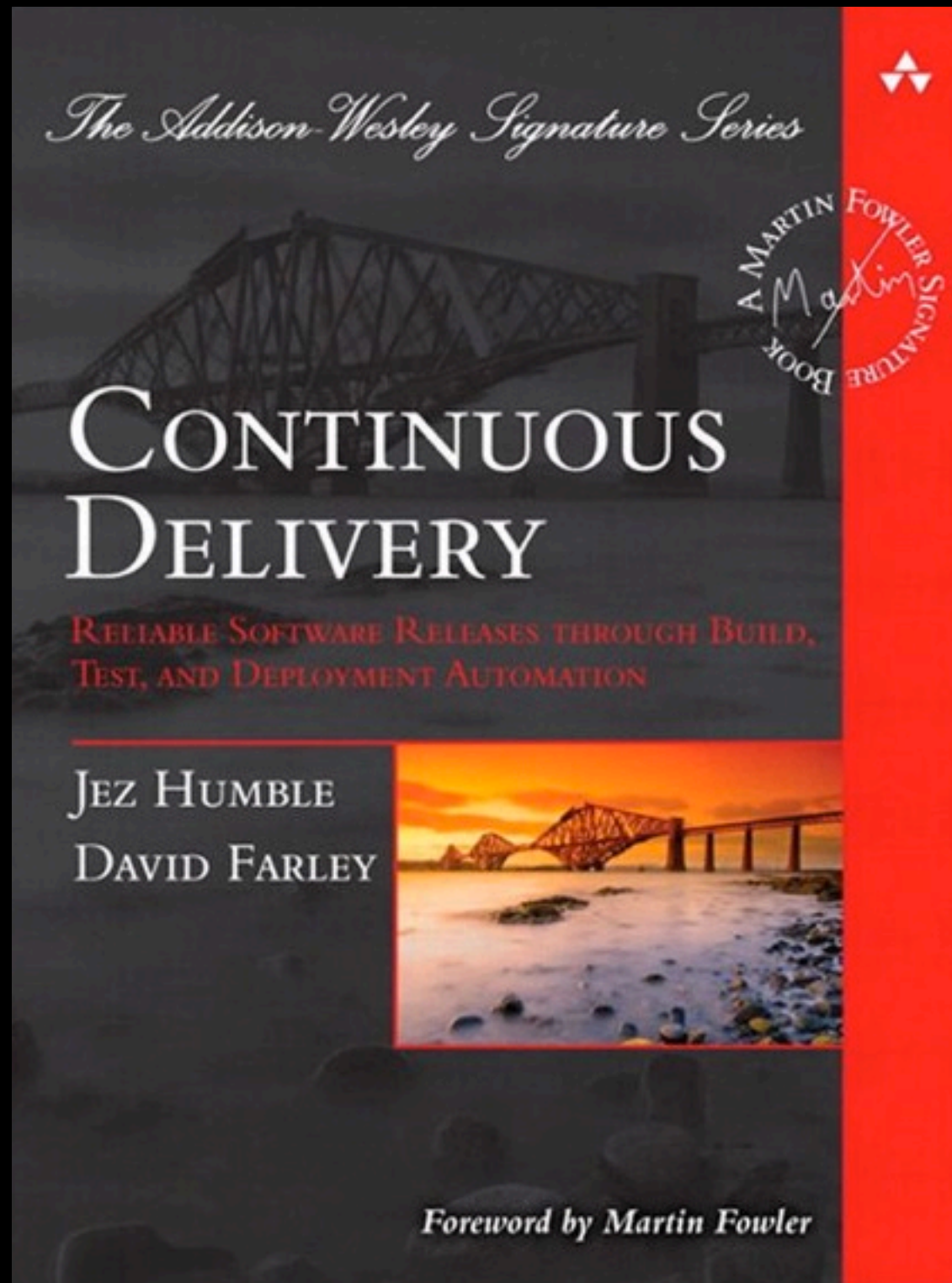
A diagram showing a top-level box labeled 'MusikShop v1' with two arrows pointing down to two separate boxes labeled 'Rekomender v124' and 'Rekomender v200'.

Rekomender  
v124

Rekomender  
v200



## Blue/Green Deployments

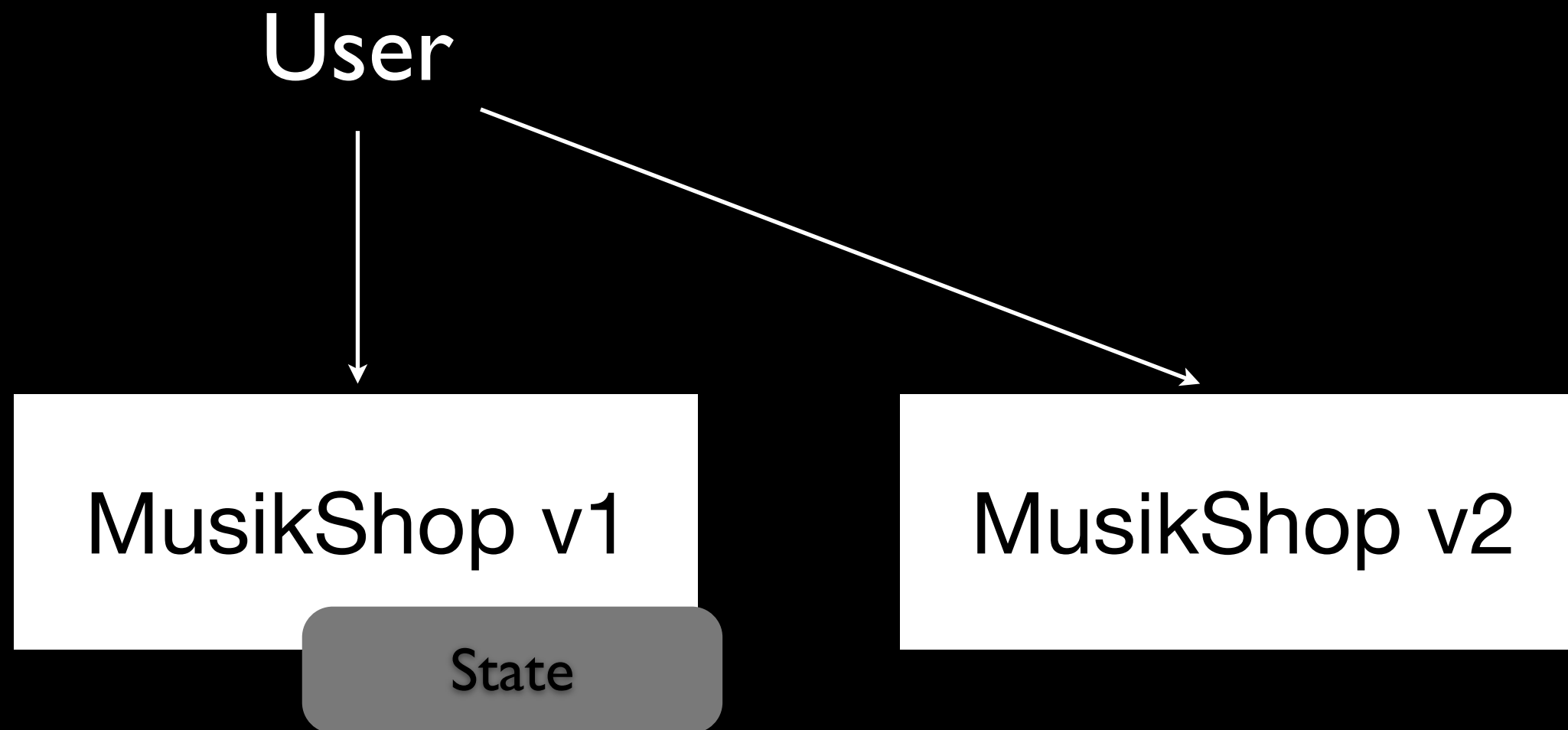


**GOTO: Aarhus 2012**

**@samnewman**



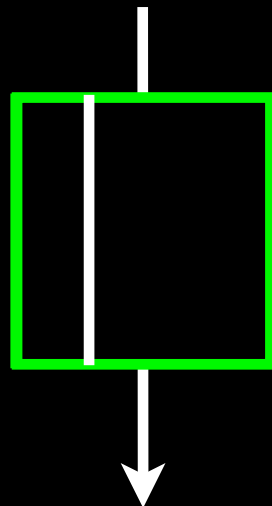
# What can stop blue/ green deployments?



Avoid stateful services

# Expensive Hardware!

MusikShop v1



Rekomender  
v200

www.MusikShop

Queens Of The  
Stone Age

Take That

Snoop Dogg

We Can't Rekomend  
Right Now!

Smoke Test

Manual Circuit Breaker



The  
Pragmatic  
Programmers

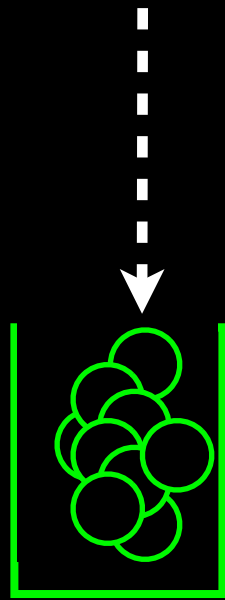
# Release It!

Design and Deploy  
Production-Ready Software



*Michael T. Nygard*

MusikShop v1



Rekomender  
v200

www.MusikShop

Queens Of The  
Stone Age

Take That

Snoop Dogg

Waiting...

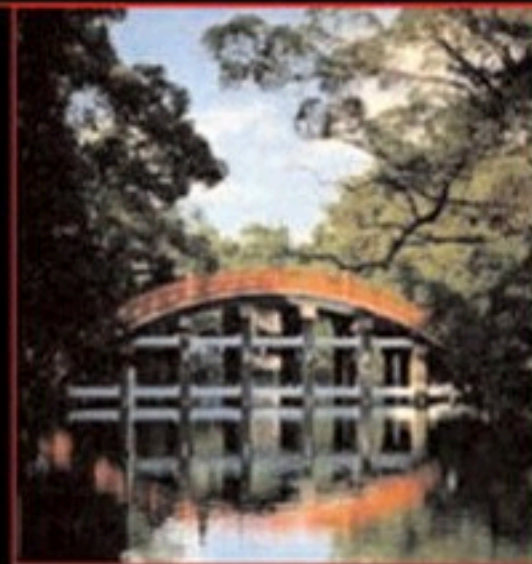
# Async Behaviour

*The Addison-Wesley Signature Series*

# ENTERPRISE INTEGRATION PATTERNS

DESIGNING, BUILDING, AND  
DEPLOYING MESSAGING SOLUTIONS

GREGOR HOHPE  
BOBBY WOOLF

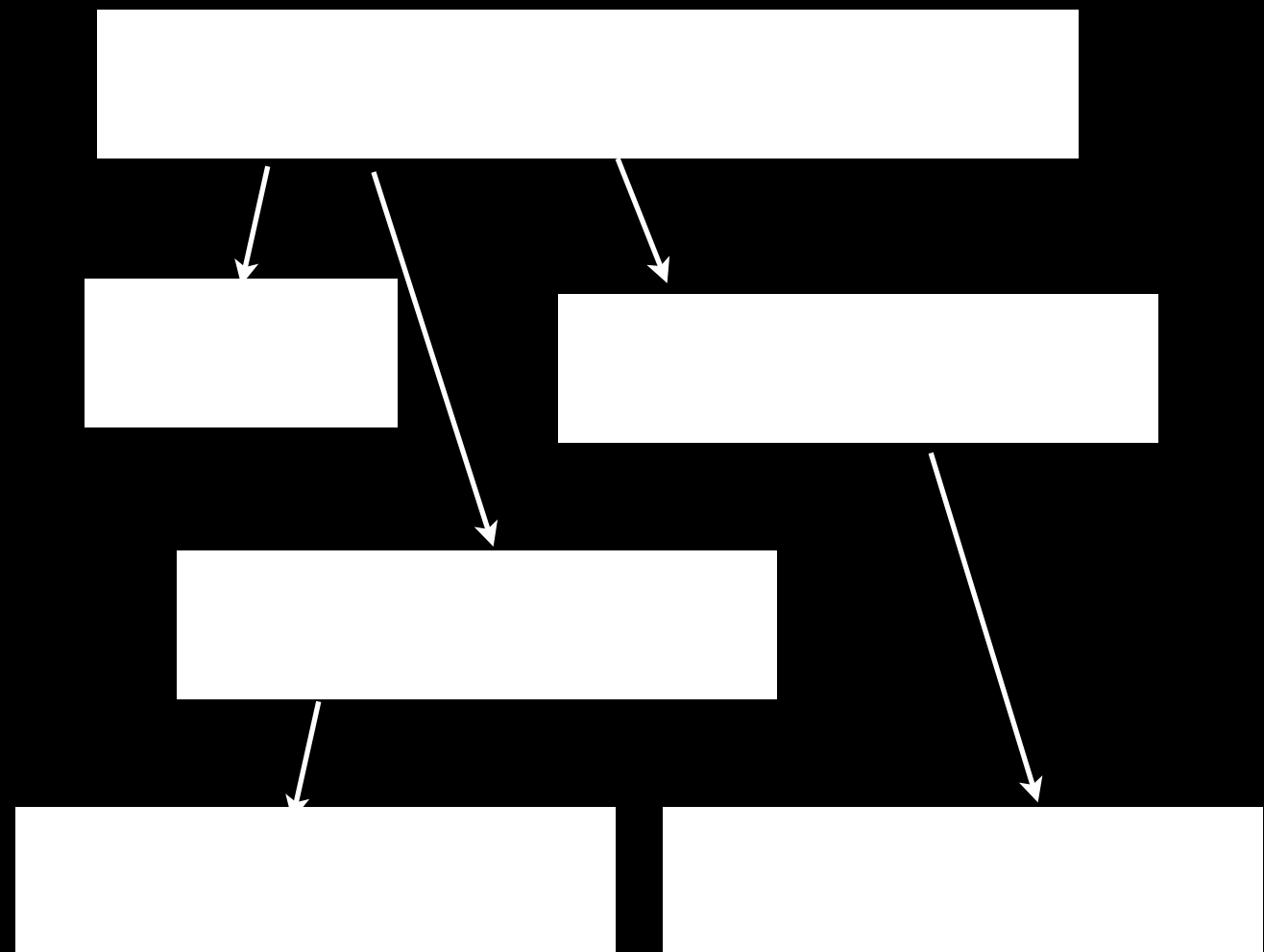
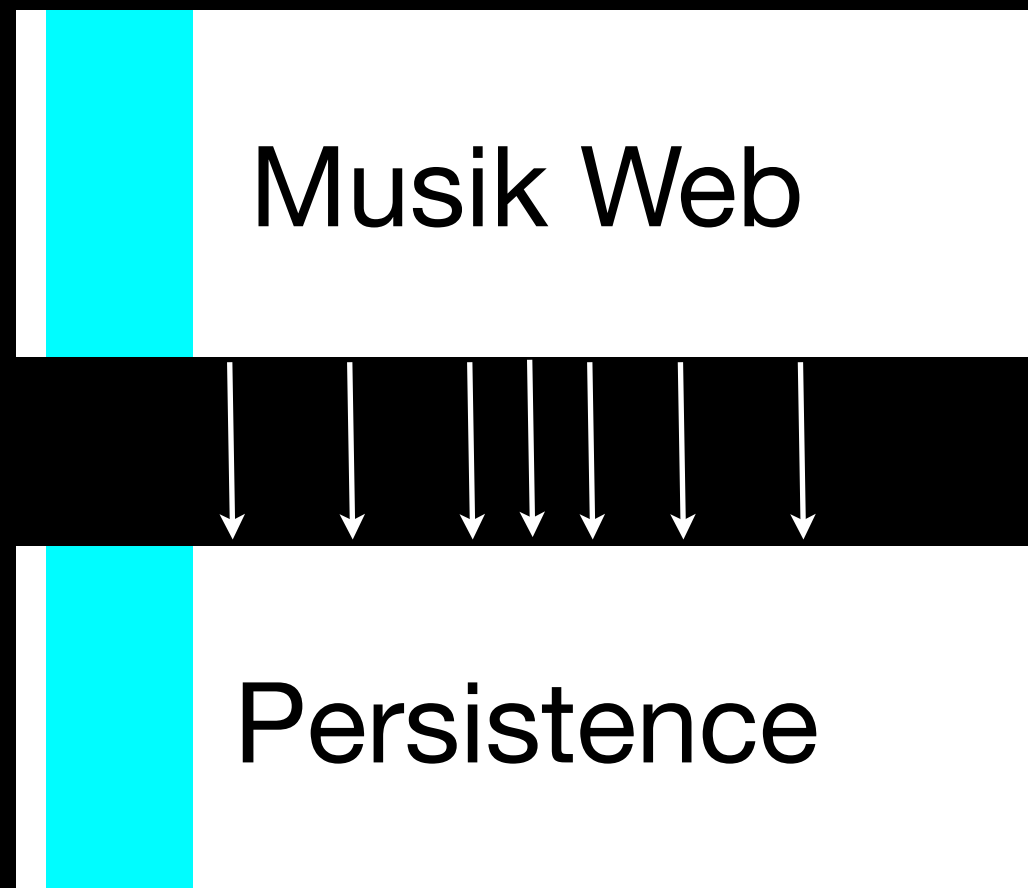


A MARTIN FOWLER SIGNATURE  
BOOK  
*Martin*

**GOTO: Aarhus 2012**

**@samnewman**

# Things to watch for...



# The Trifle





**GOTO: Aarhus 2012**

**@samnewman**





**GOTO: Aarhus 2012**

**@samnewman**

Tuesday, 2 October 12

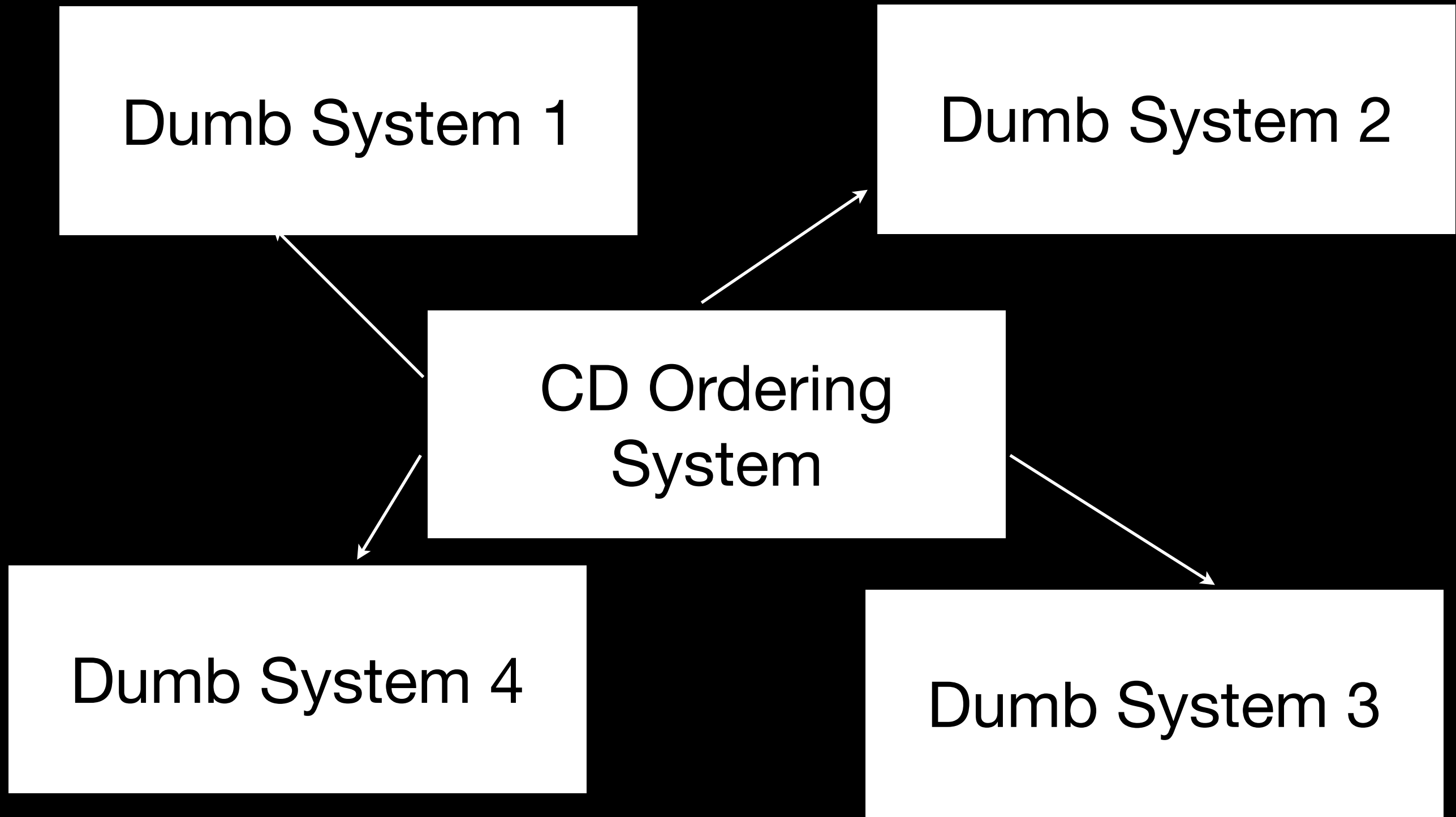




**GOTO: Aarhus 2012**

**@samnewman**

Tuesday, 2 October 12



# The Spider





*“A set of capabilities on  
an endpoint”*

Shopping Cart

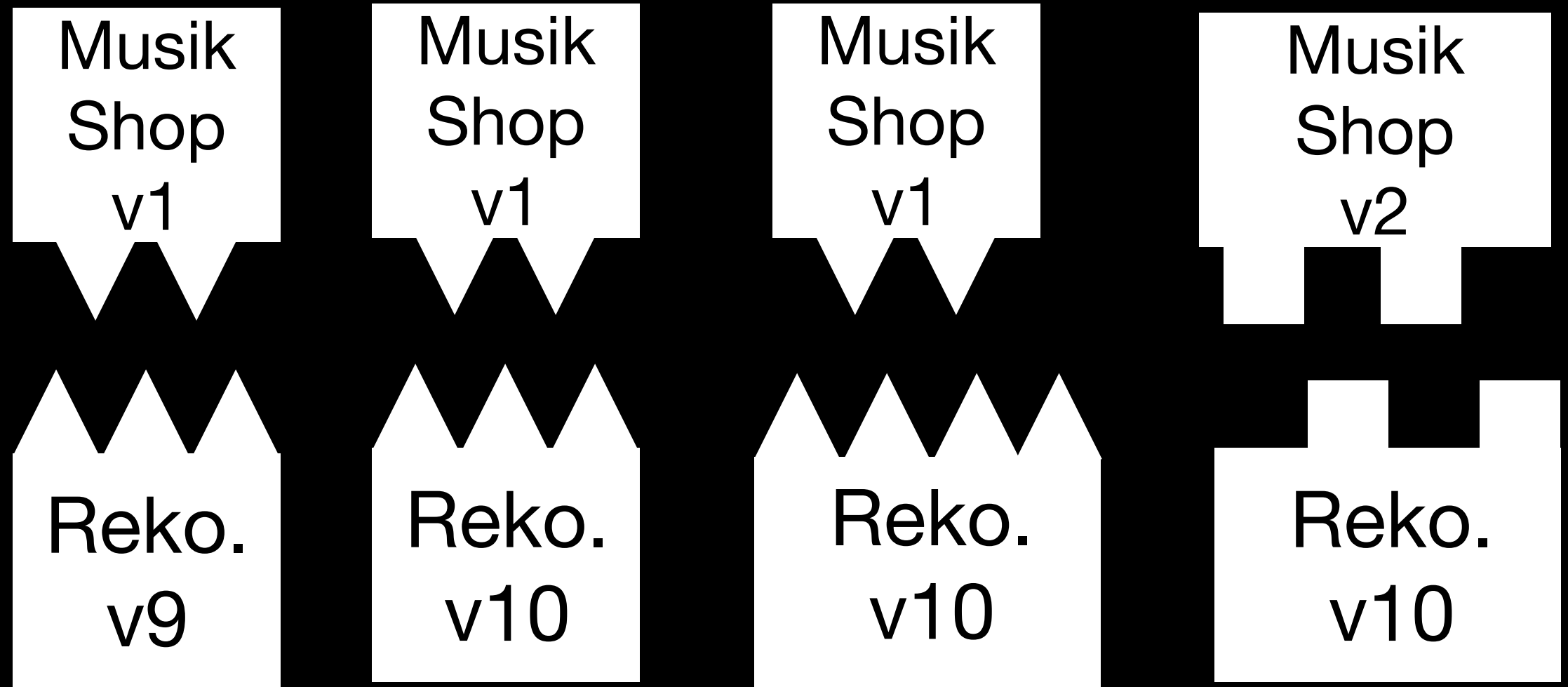
Catalog

View Latest  
Releases

Checkout

Music Library

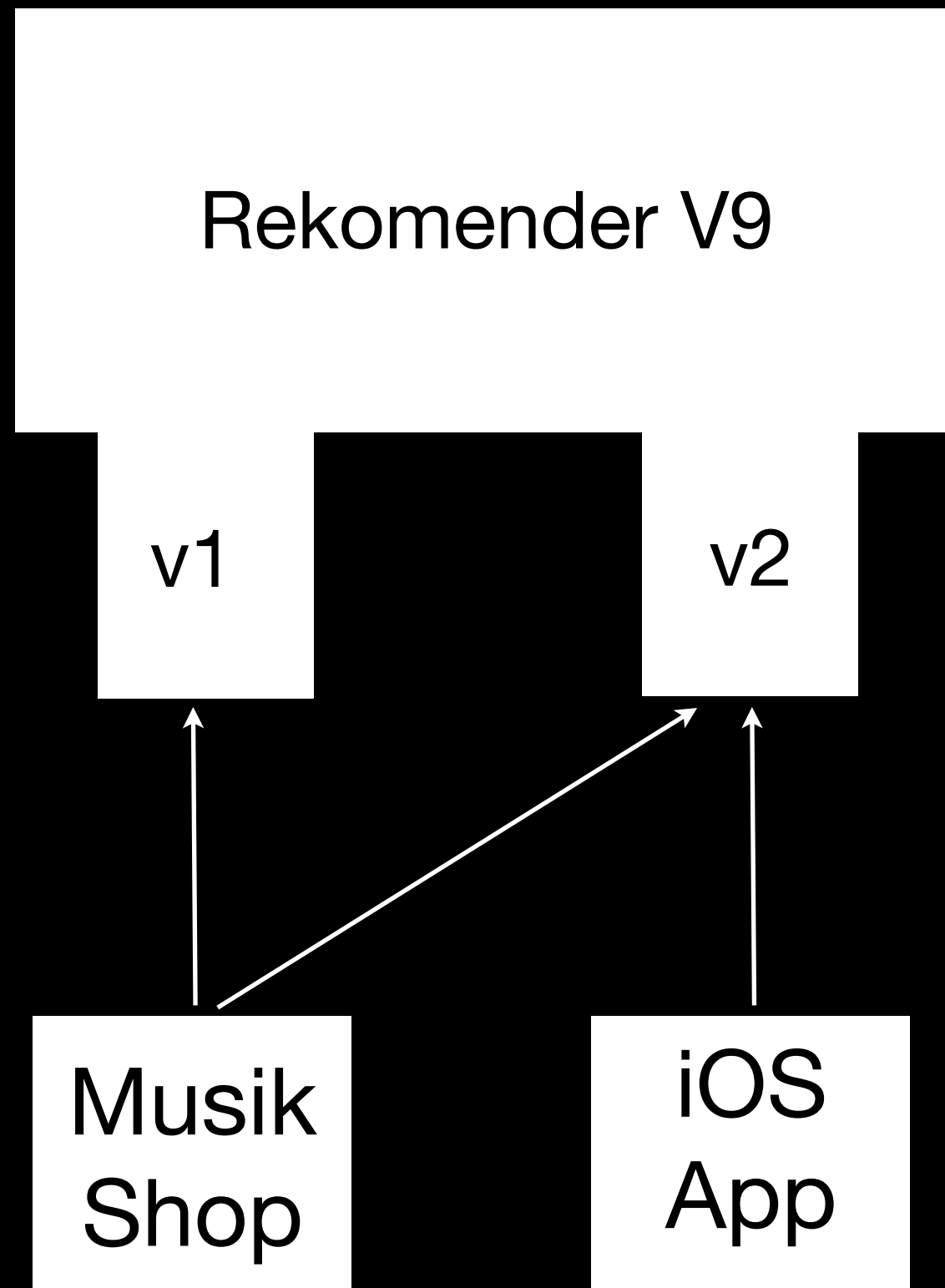
# Model Services Based On Your Business Domain



Internal  
Change

Expansion

Interface  
Change



Non-breaking expansion  
is to be preferred - but  
if you can't...

Consider maintaining  
multiple service  
endpoints



Service A

Shared Lib v2

Beware of shared  
serialization protocols

WSDL-binding  
JAXB  
Java Serialization

Service B

Shared Lib v1

# RFC 761

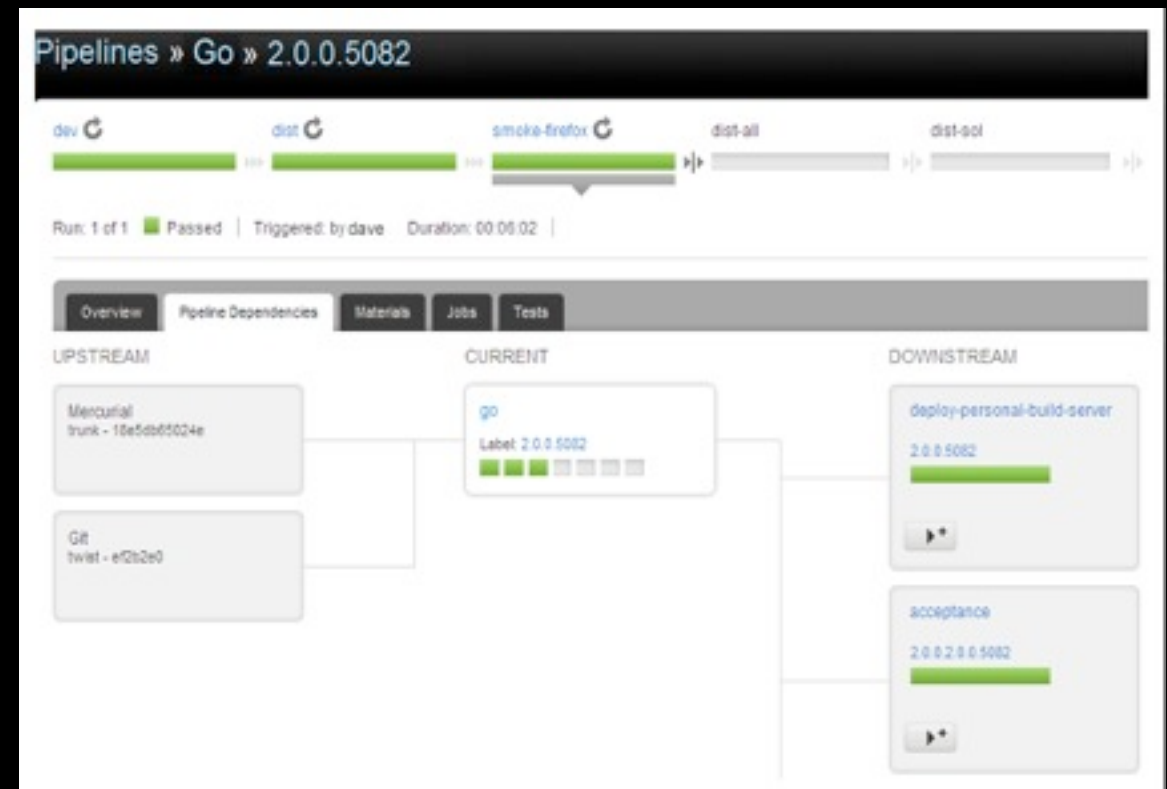
# Postel's Law:

“Be conservative in what you do, be liberal in what you expect”

# MusikShop

# Rekomender

```
26
27 def asHtml(): Node = {
28   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
29     <head>
30       <link rel="stylesheet" href="/static/common.css" type="text/css" />
31       <link rel="stylesheet" href={ cssForDisplayType } type="text/css" />
32       <meta http-equiv="refresh" content="30" />
33     </head>
34     <body>
35       { content(builds) }
36     </body>
37   </html>
38 }
39
40 private def content(builds: List[Build]): Elem = {
41   displayType match {
42     case "single" => <div> { builds.map(build => asTable(build)) } </div>
43     case "smart" => {
44       if (builds.length == 1) {
45         <div> { builds.map(build => asTable(build)) } </div>
46       } else {
47         <ul class="builds">{ builds.map(build => buildToHtml(build)) }</ul>
48       }
49     }
50     case _ => <ul class="builds">{ builds.map(build => buildToHtml(build)) }</ul>
51   }
52 }
53
54 private def asTable(build: Build): Elem = {
55   <table class={ "build " + build.getStatus.name.toLowerCase }>
56     <tr valign="middle" align="center">
57       <td>{ linkToBuild(build) }</td>
58     </tr>
59   </table>
60 }
61 }
```



# Consumer Driven Contracts

GOTO: Aarhus 2012

@samnewman

# DATABASES!

**GOTO: Aarhus 2012**

**@samnewman**

**DATA IS COOL**

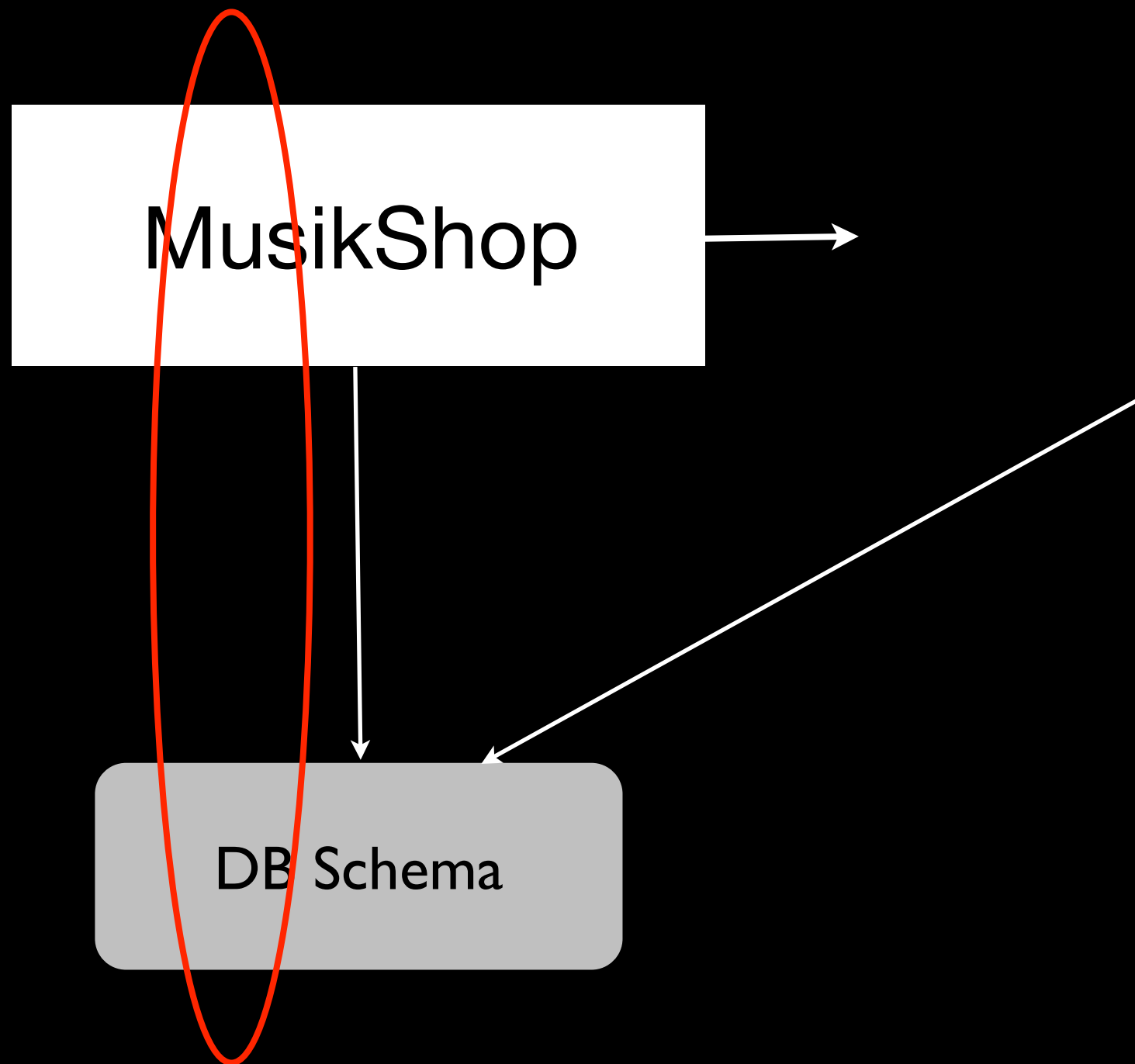
**GOTO: Aarhus 2012**

**@samnewman**

# DATABASES ARE EVIL

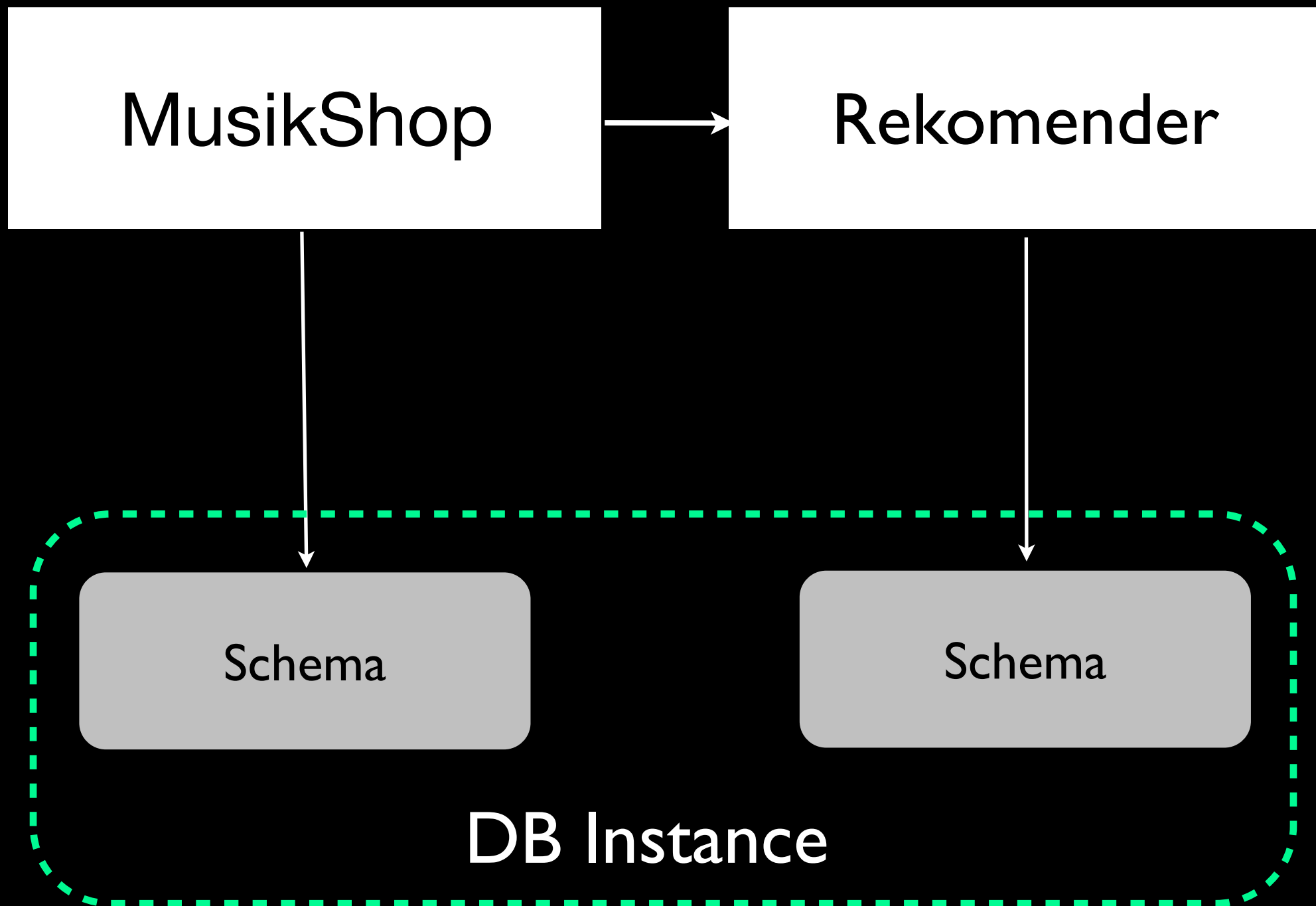
**GOTO: Aarhus 2012**

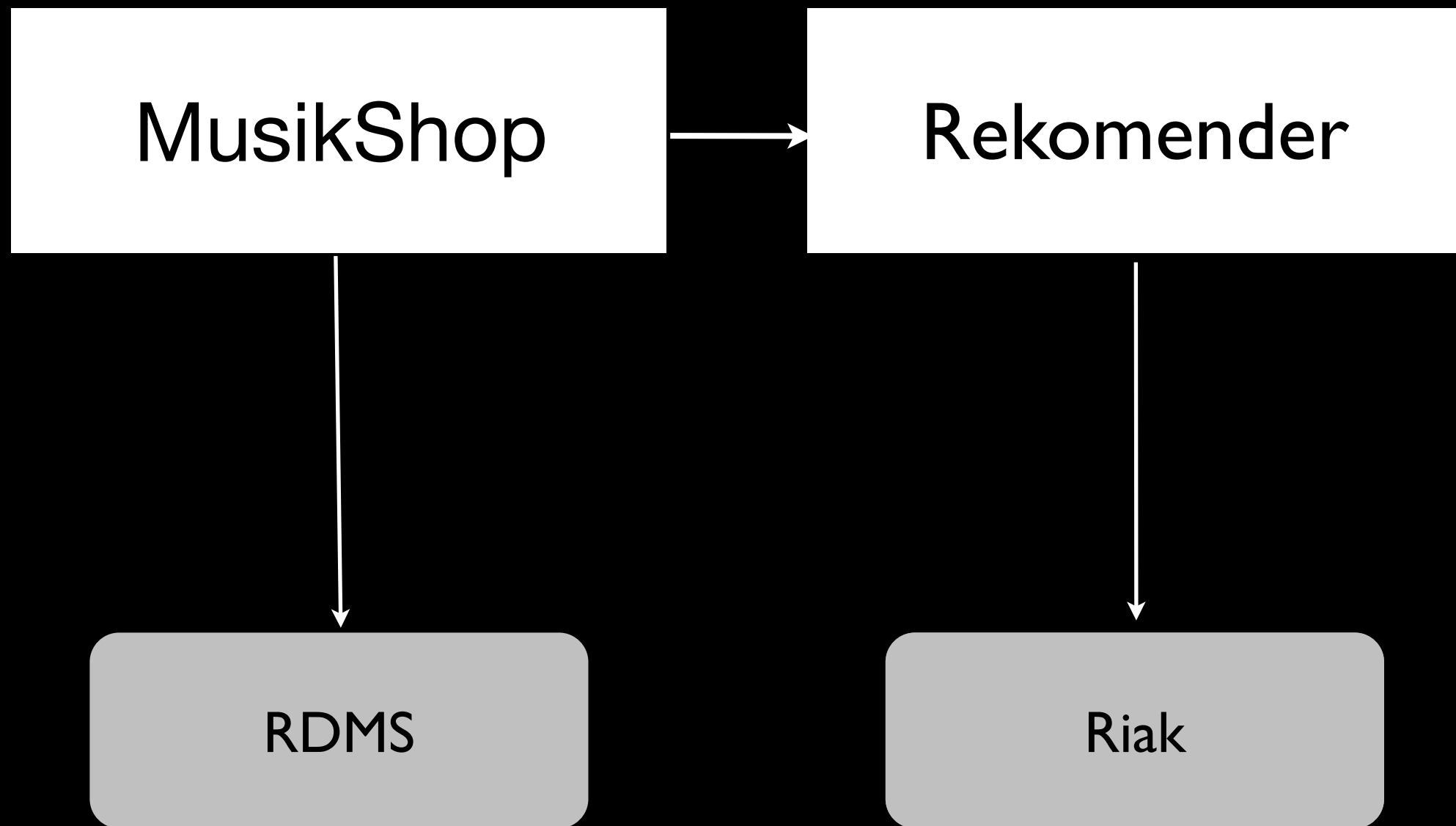
**@samnewman**

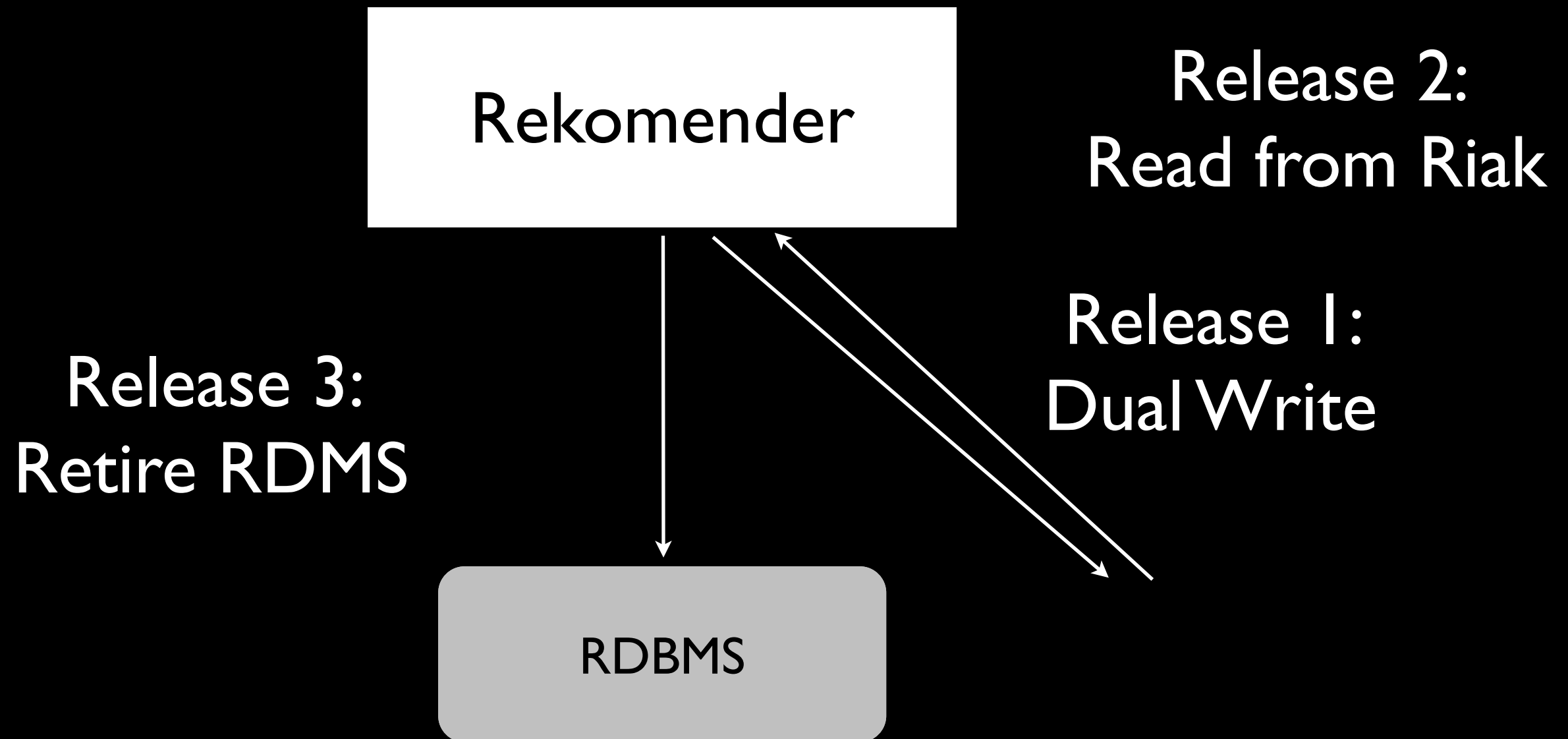




# Each service owns its own data







MusikShop

Dark Launching!

Rekomender

AceSuggest

Decompose Your Systems

Model Your Domain

Get Interfaces Right

Seperate Deployment From Release

# Questions?

# Thanks!

@samnewman

# We're hiring!