

Thursday, July 25, 13

Hi All. As a warning, those of you with previous RabbitMQ experience may find this a bit boring, as it's an intro  
-I won't try to make anyone guess what this is  
-I should explain it a bit, as I'll be following a similar format for other diagrams.  
-Blue symbol represents some feature - browsing, managing, payments, cart  
-black box is a single web node or application  
-cloud is internet

<https://github.com/spember/gr8conf2013-mq-consumer>

Thursday, July 25, 13

Make sure to run-app so it grabs all the dependencies



# Message Oriented Architecture with Rabbit MQ

Steve Pember  
Today, 2013

Thursday, July 25, 13

Let's talk about complexity and scalability

# A Few Questions

Thursday, July 25, 13

- A few questions, before we begin.
- I'd like for you to keep the answers to these in your head as we go along

Have you ever...

**Wondered why your company's  
codebase is all in one repo?**

Thursday, July 25, 13

-or rather, is everything in trunk / master?  
-Have you ever had to find the needle in that giant haystack?

Have you ever...

**thought “why are there so many  
#@\$!% tables in this database?**

Have you ever...

**Dreaded executing  
the unit test suite?**

Thursday, July 25, 13

-do you miss TDD?  
Maybe delete this one

Have you ever...

**Gone through a major  
refactoring of your app?**

Thursday, July 25, 13

whether it be a total overhaul or just a few features.

Have you ever...

**Adjusted config files to  
ensure a Quartz job runs only on  
one load-balanced node?**

With “Traditional” Monolithic Architecture,  
One can Reach MVP quickly.

The entirety of the application's functionality  
is **in one convenient location.**

**But that's it.**



Thursday, July 25, 13

The Complexity will become enormous. Any gains you think you may have made at the outset will not continue, particularly once you start to attract users.  
And grow your team.



# WAIT!

Won't scale?

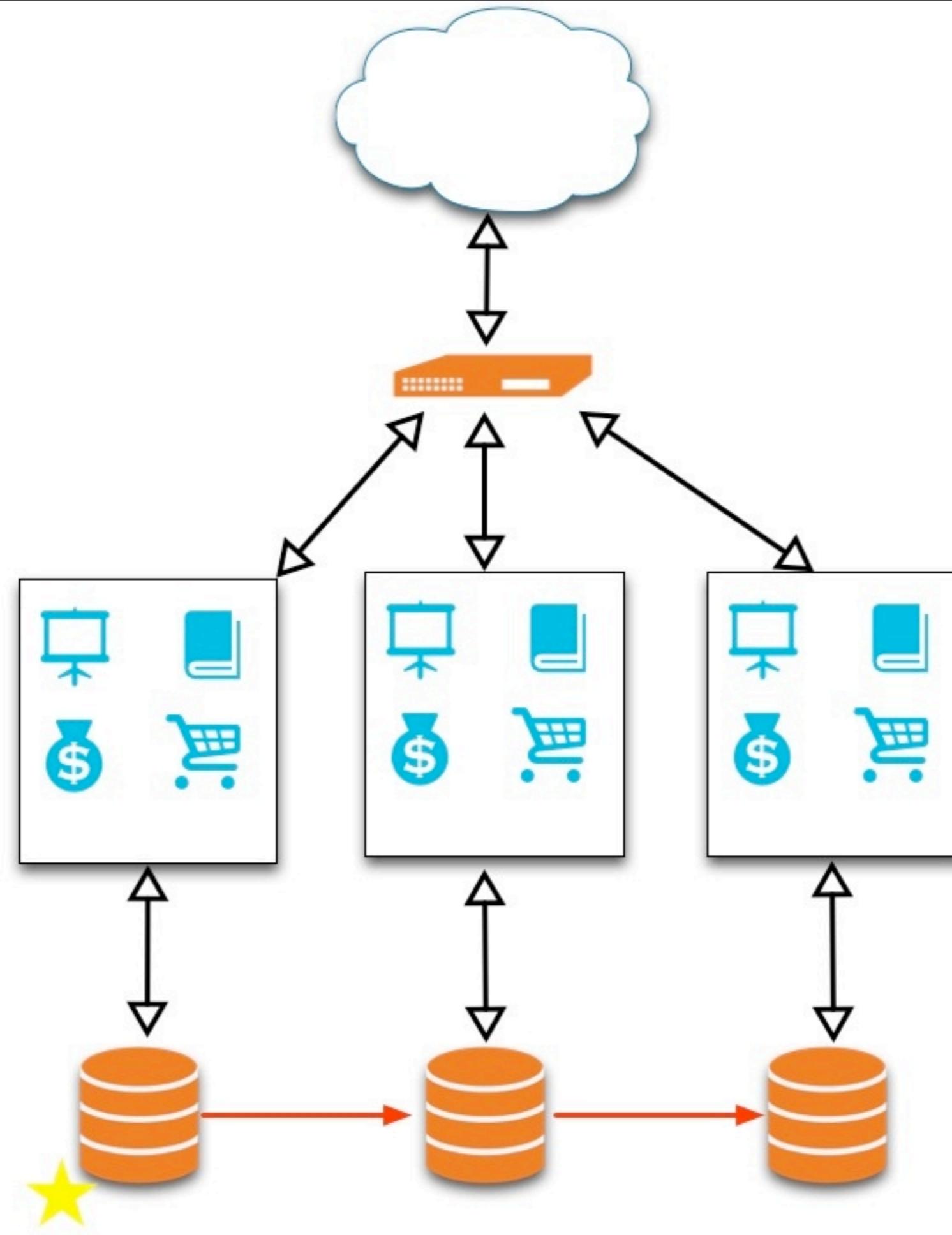
Thursday, July 25, 13

You may say to yourself, “Wait, Steve!” what do you mean it won’t scale?



Thursday, July 25, 13

I can load balance my nodes...

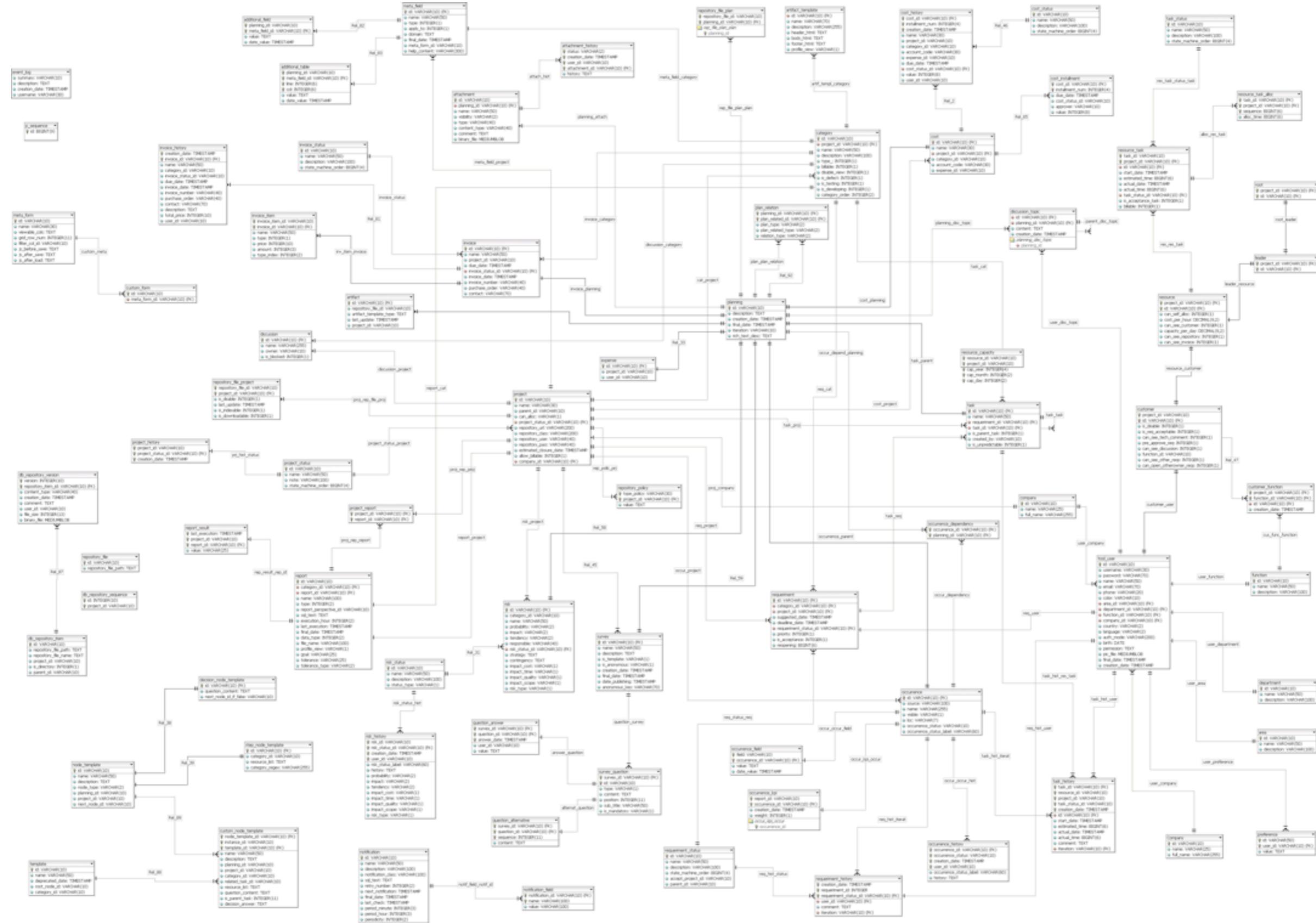


Thursday, July 25, 13

Swimlanes, master-slave dbs in case that doesn't work  
-or add caching in between my nodes

...Sure, but that's not what I mean.

# Normal?



Thursday, July 25, 13

Can you really look at a class diagram like this and really think, “Yeah, that’s awesome. Nailed it.”?

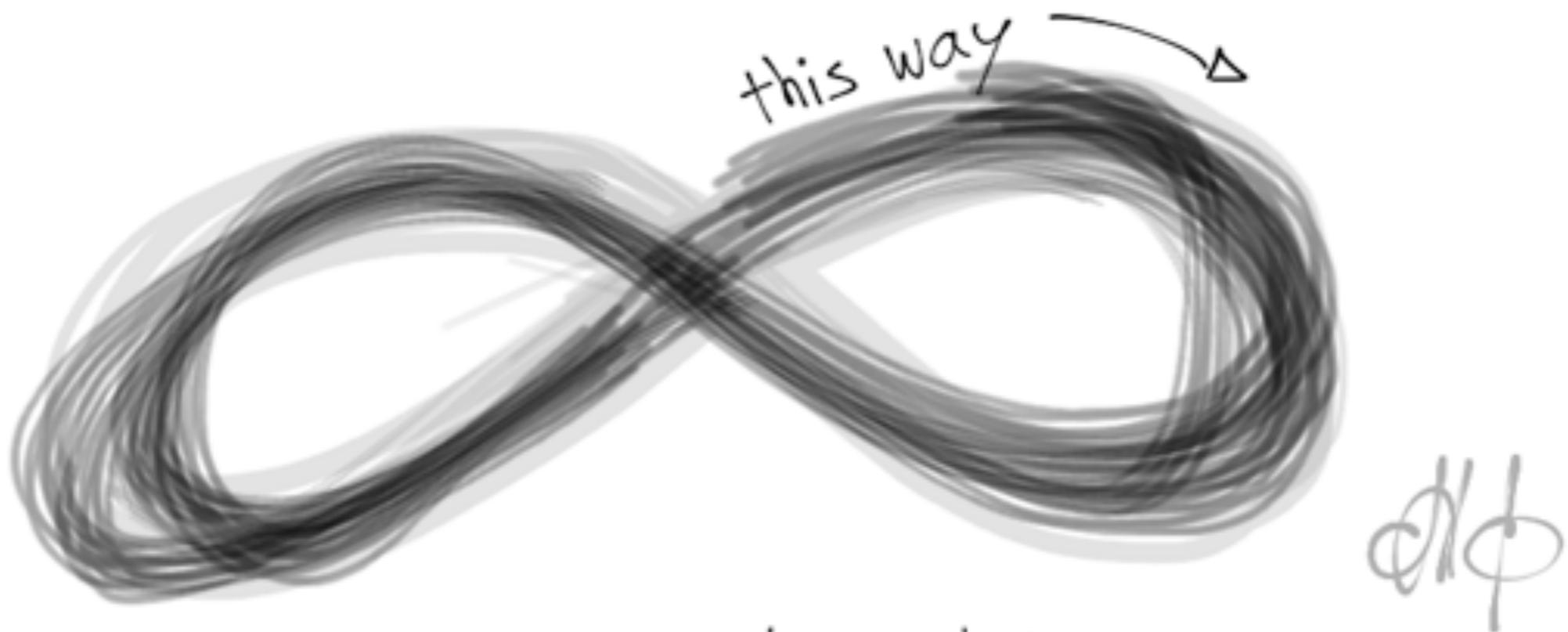
-this was the largest I could find on Google images. I'm sure we've all seen the massive versions of this that cover multiple pages, taped to the wall.



Thursday, July 25, 13

The ability to add new features, fix bugs, or resolve technical debt cannot scale with the size of the application. Throwing more developers at it also will not help

- One of the best passive-aggressive statements I've seen
- Passive aggressively scrawled on the wall: You can't make a baby in 1 month with 9 women.



Code refactoring — it won't be long.

Thursday, July 25, 13

-And trying to refactor anything?

# Concise Papers

## Cyclomatic Complexity Density and Software Maintenance Productivity

Geoffrey K. Gill and Chris F. Kemerer

**Abstract—**While the need for software metrics to aid in the assessment of software complexity for maintenance has been widely argued, little agreement has been reached on the appropriateness and value of any single metric. McCabe's cyclomatic complexity metric, a measure of

is to “ . . . identify software modules that will be difficult to test or maintain” [14, p. 308], and it is therefore of particular interest to researchers and practitioners concerned with maintenance. The McCabe metric has been widely used in research, and a recent article by Shepperd cites some 63 articles which are directly or indirectly related to the McCabe metric [20].

There is some question, however, as to the practical usefulness of the metric [4]. As noted by Shepperd, concerns about the external validity of the data and analyses in some of the previous studies can be used to mitigate some of the results, particularly those using data on small programs, often using student subjects. Therefore

Thursday, July 25, 13

This has been known for some time, and there's been papers written on the subject. Although it's difficult to measure this.

# Redacted Client Name: A Case Study

Thursday, July 25, 13

I'm generally very discrete and don't mention clients names, especially in presentations, but they explicitly asked me to do this.

\* Still, redacted it



139

Casio G-Shock – 'Garish Gold' Collection

#G-Shock, #Street

Heath\_Sheldt +



182

Young & Reckless Sun Glasses

#Young & Reckless, #Street, #Style, #Skate



125

White Cement 4

#Air Jordan, #Outdoor, #Street, #Style



119

Brabus B63 620 Widestar

Mercedes-Benz #Auto



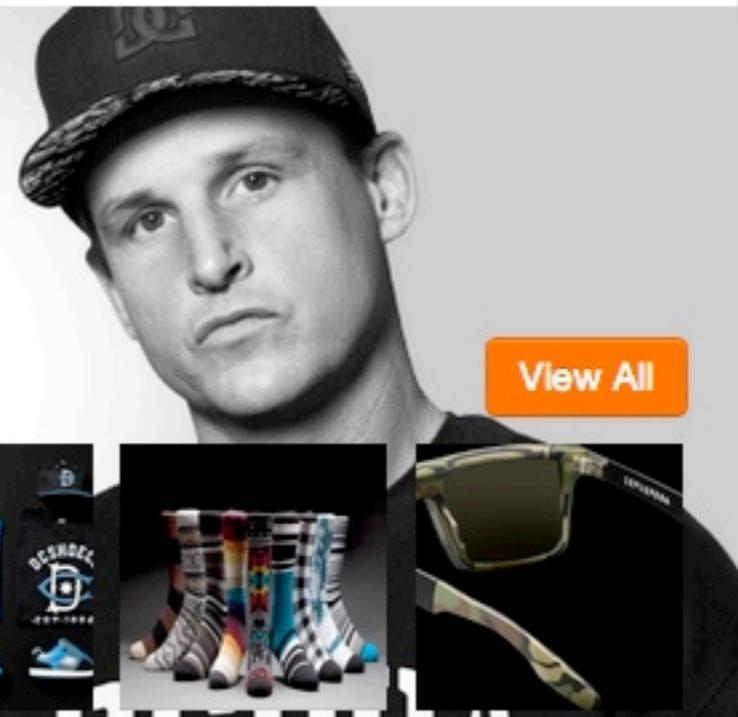
109

11 Lows

#Jordan #Daten #Street #Erlasse

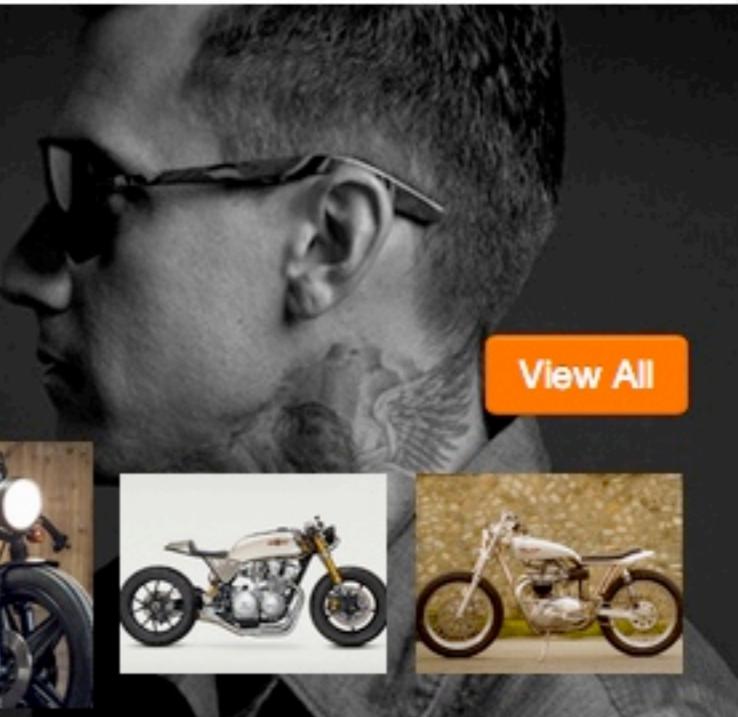
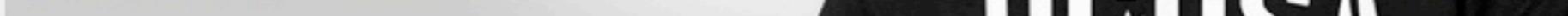
Thursday, July 25, 13

The client was/is a lifestyle brand site. I bit like pintrest  
-can share, rate, comment on other peoples style. See what's trendy.  
-I don't exactly understand the 'lifestyle' term  
-most importantly, it's a Grails application



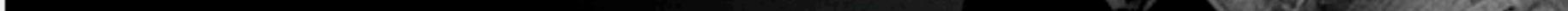
**RD** by Rob Dyrdek

[View All](#)



**Customs** by Carey Hart

[View All](#)



Thursday, July 25, 13

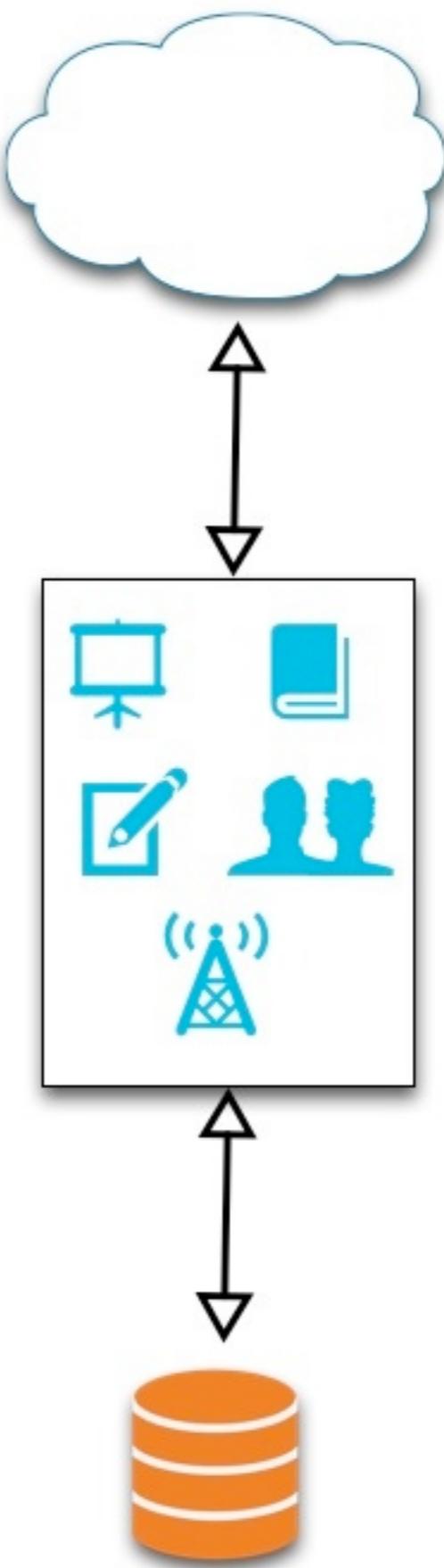
Even have some of these 'lifestyle' (I don't get the term) celebreties come in and create custom collections of things to share



© Peter M. Fisher/Corbis

Thursday, July 25, 13

-we / Cantina were brought in to help recover the application from a remote team. Development had stalled.  
-the remote team had constructed the application in your traditional sense. (Incidentally, they were making heavy use of Mixins – mixins depending on other mixins – in a large spider web. One change caused dozens and dozens of file compilations. Cautionary tale!)



Thursday, July 25, 13

When we first started working on it, this is the basic setup for client. A single monolithic .war file

- Product Browsing/ Catalog (Key feature)
- Product Curation
- User metrics
- Social Graph
- notifications

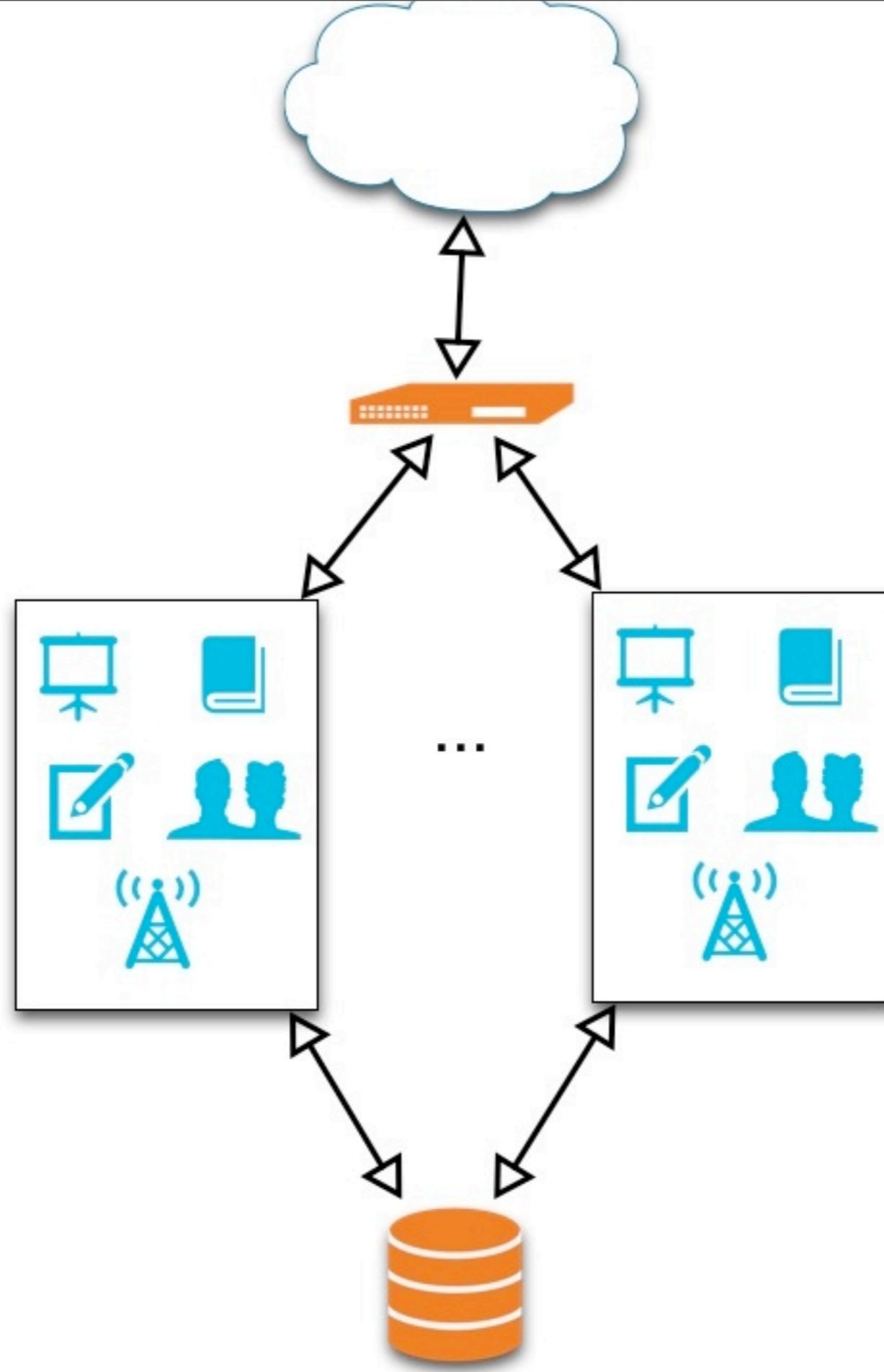
<sarcasm>**Luckily**</sarcasm>

everything was writing and  
reading from the same DB.

Thursday, July 25, 13

-mention the high-rate rate of the stats

Each feature set ran on each node.



Thursday, July 25, 13

-As you might expect, the original team setup multiple instances of the app and load balanced it.

-

Heavy caching and balancing  
mitigated the problems...

... **but** still under-performant and  
cumbersome to work with.

Thursday, July 25, 13

We added some distributed caching (love ehCache, btw) and few more nodes to the load balancer in order to get some initial wins...

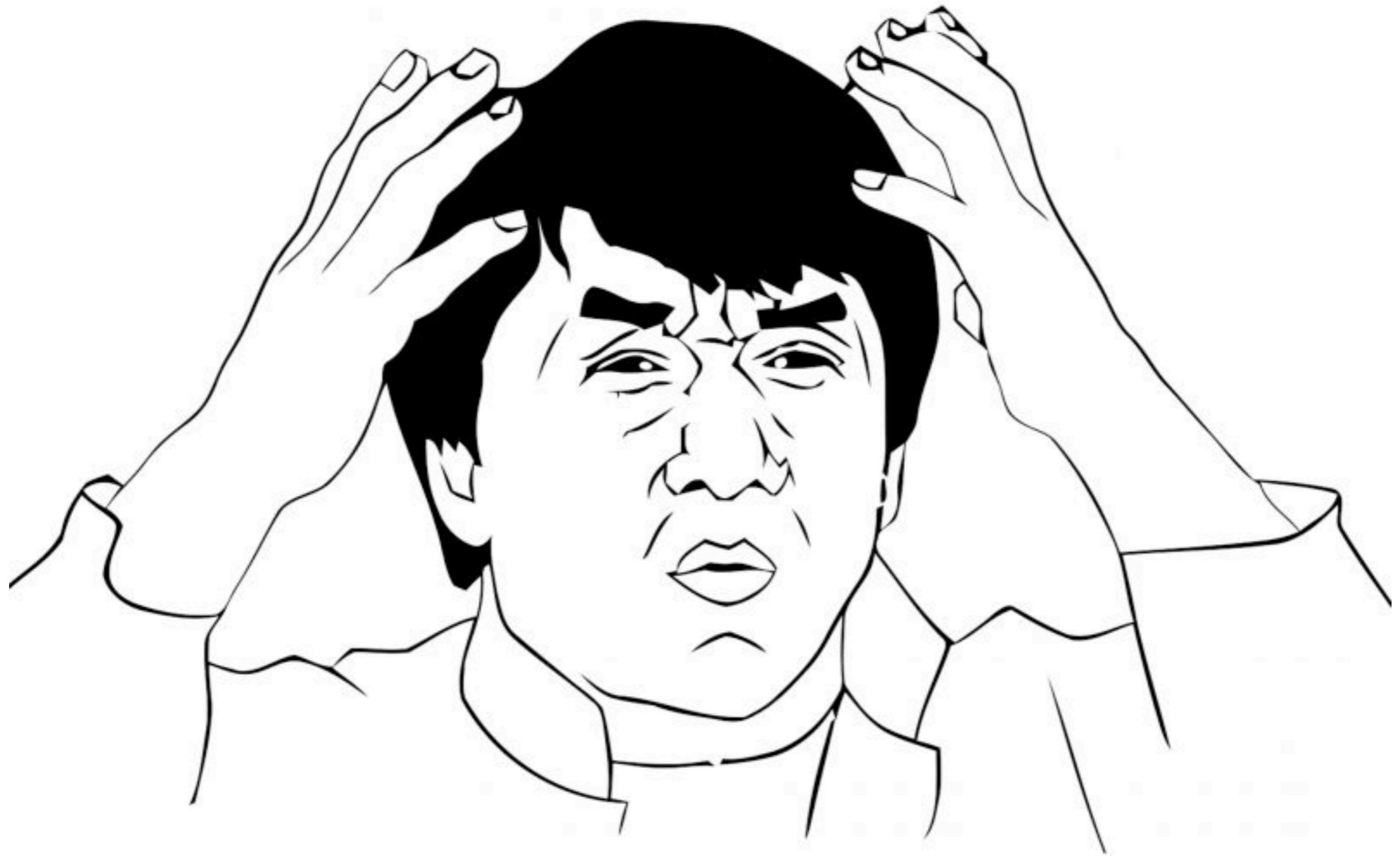
-but it wasn't enough, and very difficult to work with

...and the product owner had  
ambitious goals

# Performance and time hindered growth

Thursday, July 25, 13

- The growth of the product was hindered by the classic duo of performance and time.
- The site could hardly handle the current traffic.
- Member Growth was good, but not good enough.



Thursday, July 25, 13

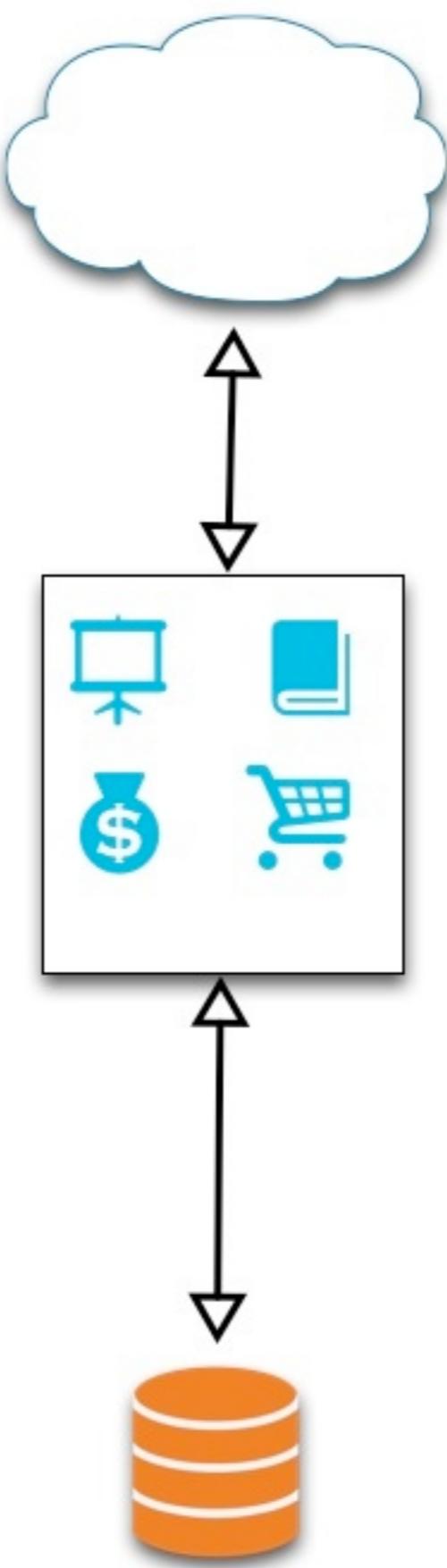
We didn't want to go down that same old route.

-Second thing we did was take a step back and look at their current setup.

# Enter SOA

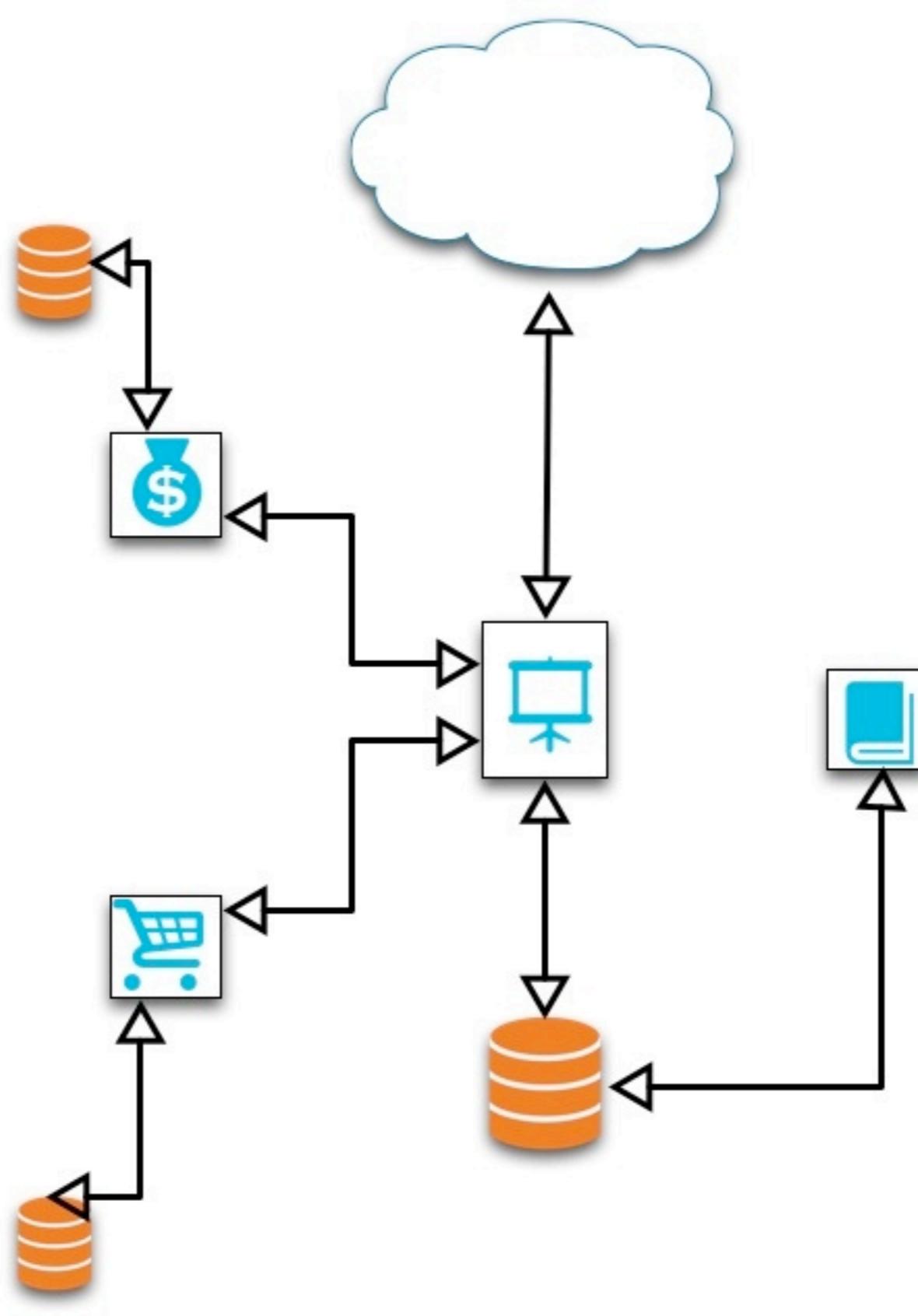
Thursday, July 25, 13

- our first tactic was to embrace Service Oriented Architecture (that's what the SOA stands for)
- Is everyone familiar with it?
-



Thursday, July 25, 13

-Let's take our anecdotal design, an e-commerce store.



Thursday, July 25, 13

With SOA, one would break up each individual component, and setup communication between each service over some direct service calls. Eg SOAP or REST.  
-Synchronous



Thursday, July 25, 13

Which keeps the individual components easier to manage and keep track of

Decoupling your app into disparate services keeps complexity down...

... which creates **faster, leaner** code ...

... which results in rapid long-term development time...

... and easier code **maintainability**...

**... which saves you Money.**



Thursday, July 25, 13

and other numbers that managers love

Also, it's fun! 

# Separation of Concerns is a **Good Thing**.

Thursday, July 25, 13

**possibly delete**



# However.

Thursday, July 25, 13

There's always a downside, eh?



Thursday, July 25, 13

I don't know who that is, but hold on.  
I was told he's kind of a big deal.

Following this approach requires an **architect** to drive the distributed vision and keep folks in line. 📣

Thursday, July 25, 13

- I can't stress this enough. Really gotta own it

There's some non-trivial upfront time investing in a service communication format.

Intra-Service communication has a cost,  
particularly if it's **synchronous**.



Thursday, July 25, 13

- \* Having a web of interconnected services can also grow out of control. Having to configure service nodes so that each knows where the others are located in the network can be cumbersome to manage.  
\*There are tools to help with this, of course. There's a tool that Twitter released called 'Figment' for th
- \*Also, ESB! Enterprise Service Bus, or even an EMB Messaging Bus.  
\*but that adds another layer and potential communication step, although it's necessary for larger SOA implementations for versioning and management. A bit more on that later.

# Enter MODA

The ‘D’ stands for ‘Decoupled’.

**Heavily Decoupled.**



Thursday, July 25, 13

Your basic approach. A single Producer, a Queue of Messages, and a Consumer. In this case, quite the prodigious one.  
Note the queue backup. He should probably share.

Route Messages **asynchronously** through  
a **message queue** when data needs  
processing

Thursday, July 25, 13

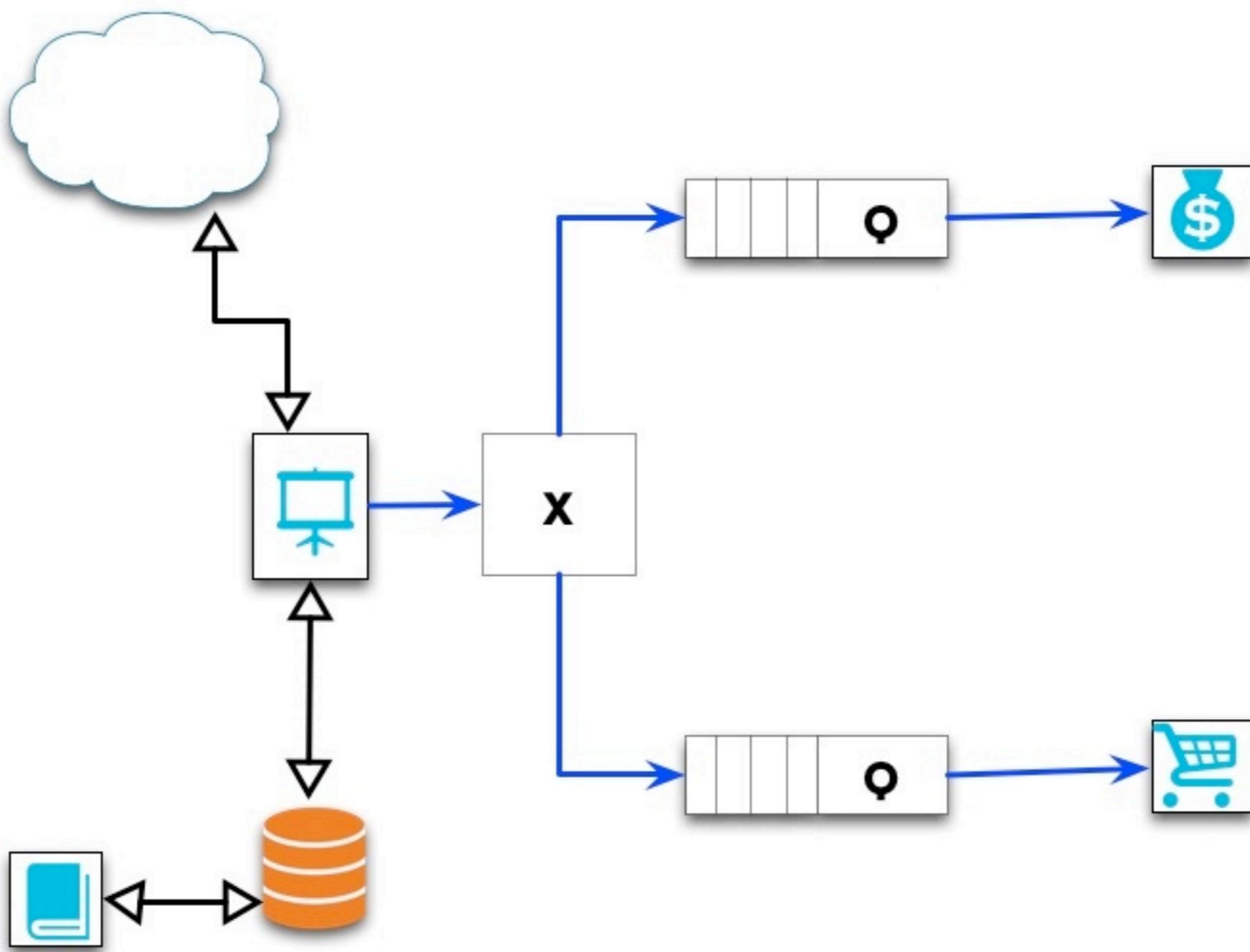
The key to a message oriented approach is when certain events occur to encapsulate data into individual messages and drop them on some queue or exchange

# Event-based instead of Procedural

# **RESTful** Messages and Events...

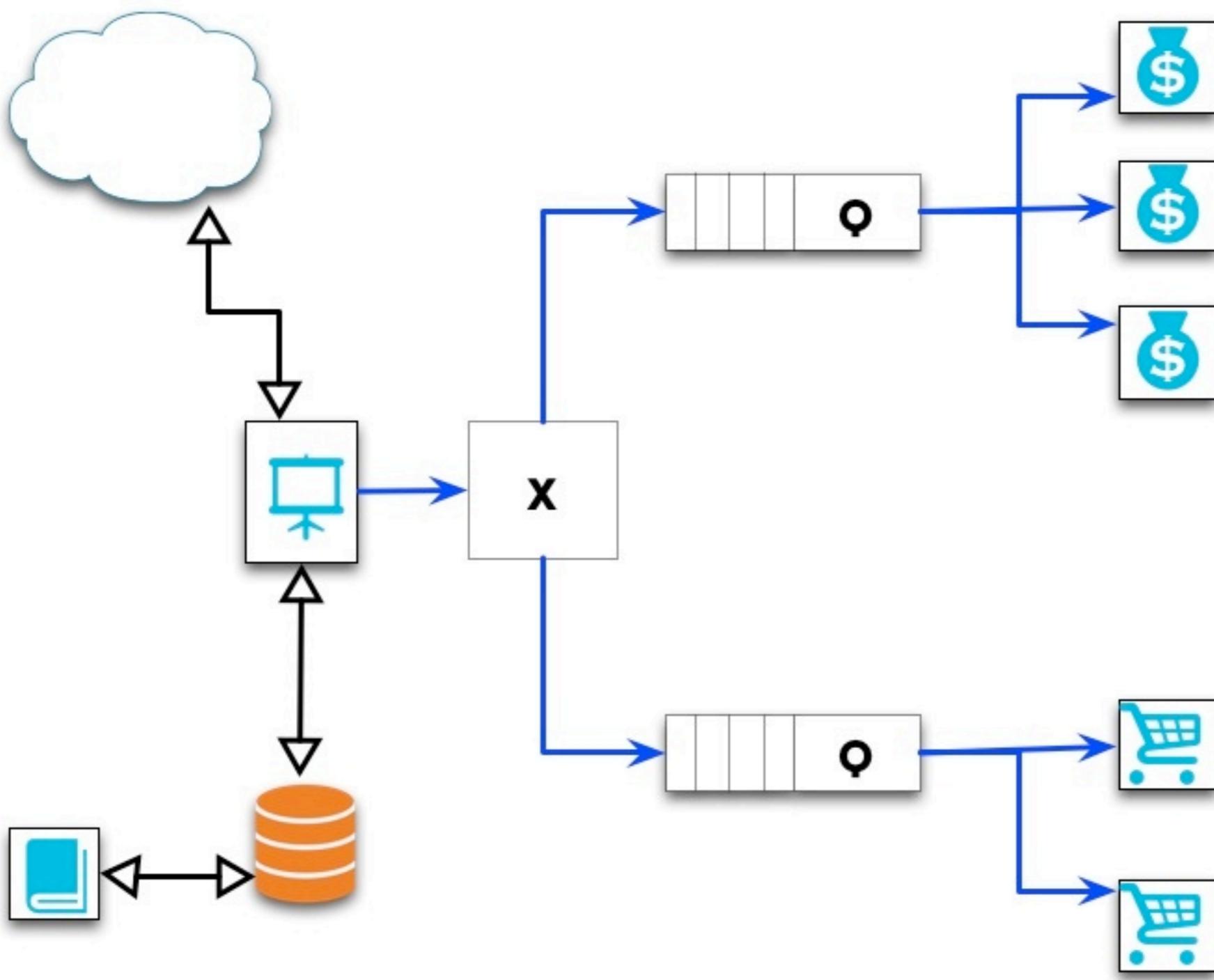
... Consumers only operate on data in the Message.

Add additional nodes to handle load  
without additional configuration



Thursday, July 25, 13

- Example from before, an e-commerce site
- we've broken the features into individual applications and spun up one node per each
- connected the users' cart to a queue, and our order processing into another queue



Thursday, July 25, 13

- suppose the cart and order processing start experiencing heavy load
- programmatically spin up new instances and you're good
-

# Additional Decoupling

 **Time**

 **Location**

 **Cardinality**

---

Thursday, July 25, 13

**TIME: Asynchronous**

**Location:** Nodes do not care where the other nodes are, only worry about the broker

**Cardinality:** Nodes do not care about the number of listeners of their messages

# Data Processing: Use a Message!

## Data Request: Consider a Service Request

Thursday, July 25, 13

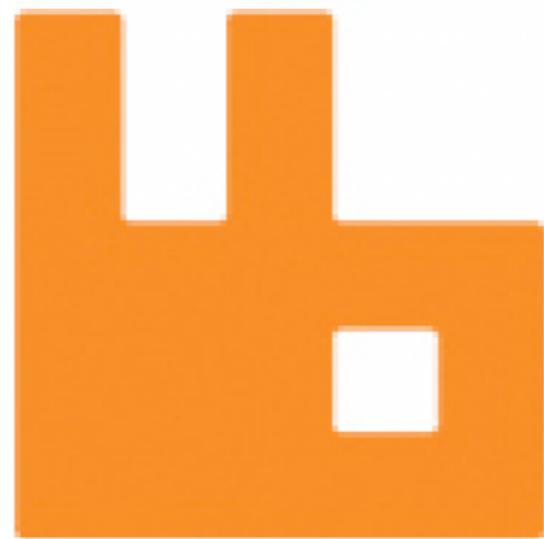
Not everything needs to be a message. I argue that a good rule of thumb for a specific event to be a candidate for Message Enqueuing is anything that requires some processing to be done on the event, may require a database write, and can get away with being synchronous. Frankly, anything that's not saving User Monetary information or dealing with some govt regulation is a good candidate.

Now, if you only need to grab some data that's in another service, you may be able to get away with a direct request.

Consider using the Events Push plugin and  
Async to handle responses for the client

# But which Broker to pick?

**ØMQ**



**ActiveMQ**

---

**OPEN**  
**AMQ**



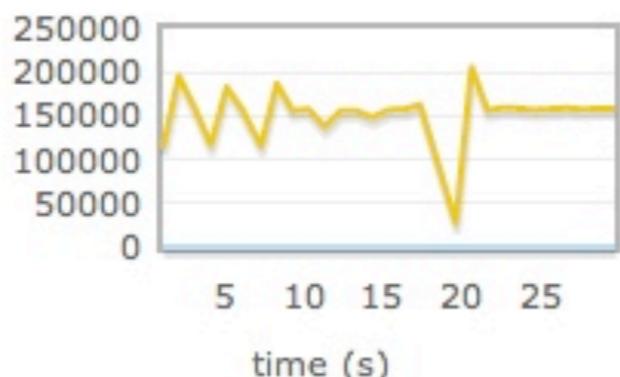
# Rabbit MQ

Language Agnostic (additional decoupling)

Message Persistence

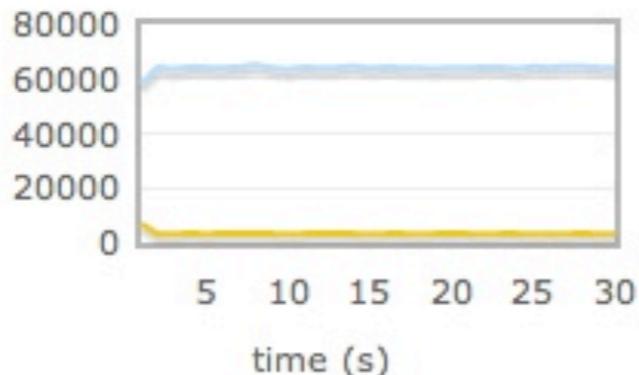
Message Recovery

max publish  
**149910**  
msg/s



This uses a couple of the cores on our server - but not all of them. So for the best headline-grabbing rate, we start a number of parallel producers, all publishing into nothing.

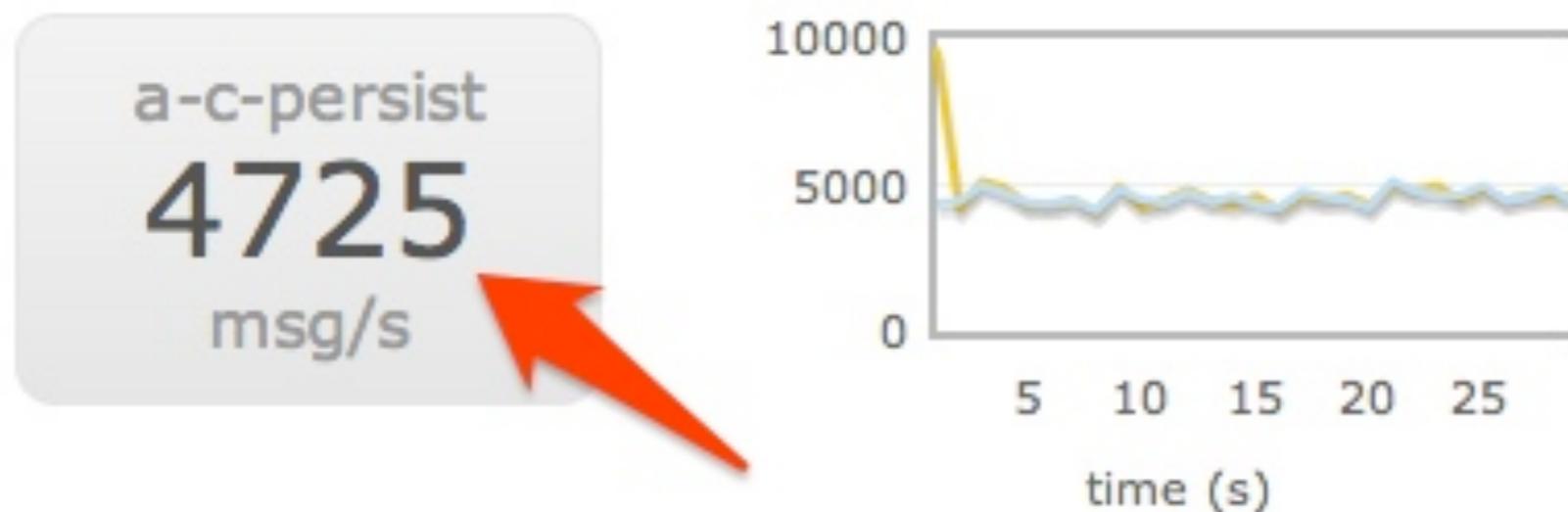
max consume  
**64315**  
msg/s



Of course, consuming is rather important! So for the headline consuming rate, we publish to a large number of consumers in parallel.

Slower than other Brokers due to the amazing features.

# Especially with Persistence





**spember@cantina.co**  
**@svpember**

Thursday, July 25, 13

If you're struggling to process 4k+ messages/ second, that sounds like an awesome engineering problem and you contact me.

RabbitMQ Plugin supported by SpringSource

Tags : messaging, rabbit, rabbitmq | Latest : 1.0.0 | Last Updated: 30 November 2012 | Grails version : 1.2

Authors : Jeff Brown, Peter Ledbrook | License : Apache License 2.0 | Organization : SpringSource

Dependency :

```
compile ":rabbitmq:1.0.0"
```

Documentation Source Issues

Thursday, July 25, 13

And finally, there's a fantastic plugin written by Peter Ledbrook and Jeff Brown  
–Why wouldn't I choose ActiveMQ? You can't go wrong, really, if you choose to use that instead.  
–ActiveMQ plugin isn't as well maintained as rabbits'. Author says ActiveMQ isn't "production ready"

# ... but what about ActiveMQ?

Thursday, July 25, 13

This question comes up often.

- Overall, you can't go wrong with ActiveMQ
- supports many of the same features as Rabbit, with similar performance
- but, clients do not exist for it in as many languages as Rabbit
- Minor point: based on JMS
- ...but...

## Description

### ActiveMQ Plugin

grails-activemq is a plugin to embed ActiveMQ in a Grails application. This release is very simple, in future releases we will provide more support for ActiveMQ advanced configuration.

This plugin can be used in conjunction with jms plugin to develop MDP (Message Driven Pogos).

This plugin at the moment is not production ready. Use at your risk for rapid prototyping with JMS without installing anymore.

## Code

Plugin code is located at <https://github.com/dmsovetov/grails-activemq>. If you wish to contribute, feel free to open a GitHub Account and the

Thursday, July 25, 13

... The grails plugin isn't ready yet.

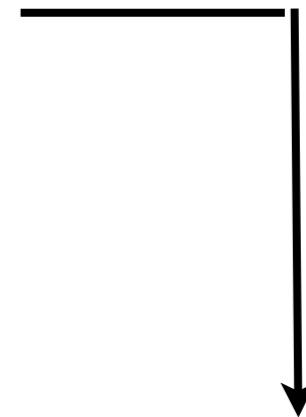
# ...What about an Enterprise Service Bus?

Thursday, July 25, 13

- \* Some of you have been thinking, “waiiiitttt... what about just using an ESB?”
- \* Can use one without the other, rabbit would make an excellent addition, working together
- \* Creating an ESB with rabbit would be another talk entirely, I think.
- \* Would be happy to discuss this afterwards!

# Leaping\* Into Rabbit MQ

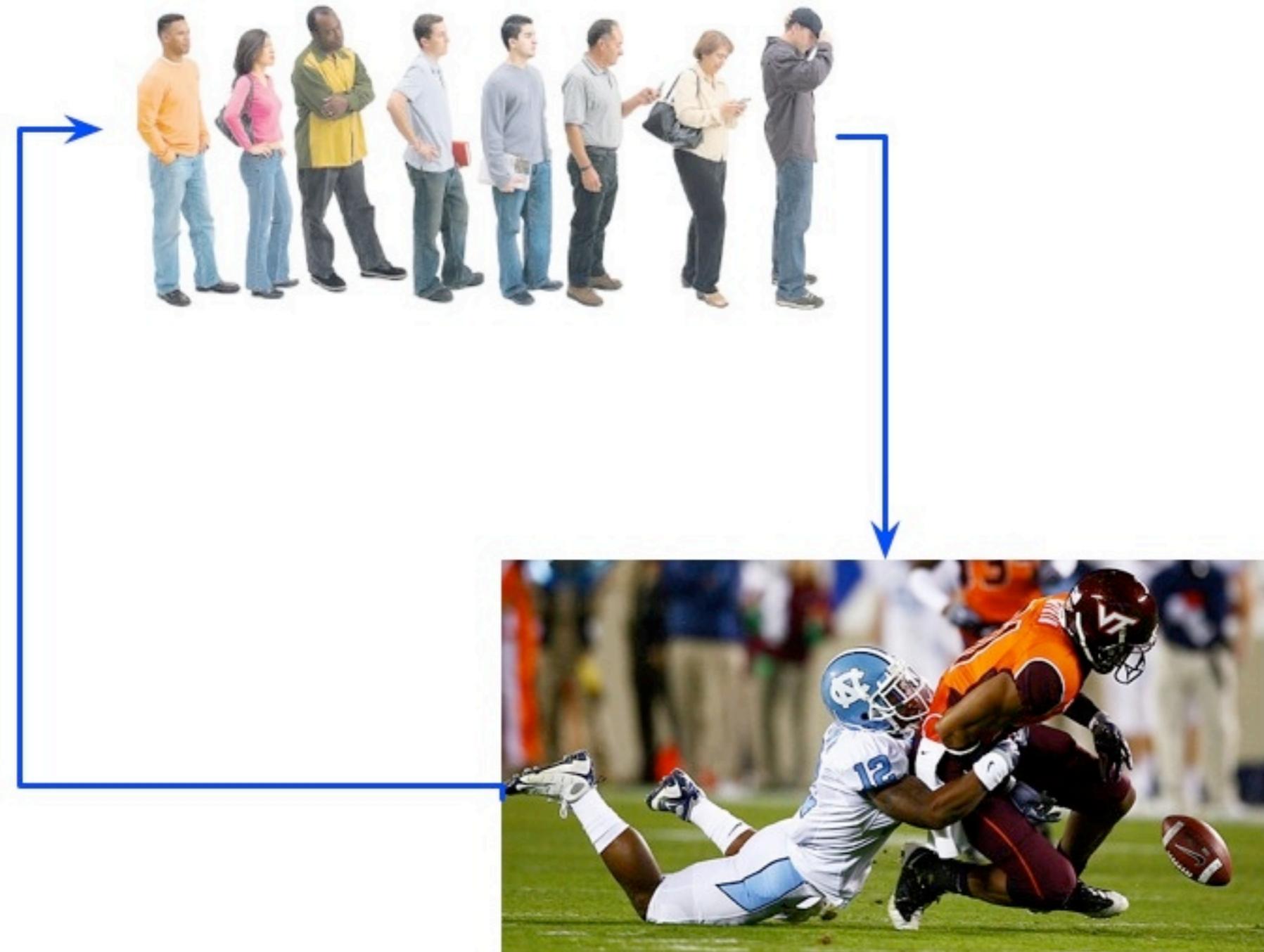
\*I'm sorry.



Thursday, July 25, 13

The previous description of message queues glosses over what is happening a bit.

- Messages are routed through an EXCHANGE in to one or many queues.
- Messages are then plucked off the queue by an attached, waiting Consumer. Or, the first available if multiple are attached.
- Exchanges can have different delivery techniques.



Thursday, July 25, 13

If a message is mishandled or the consumer crashes / has an error during message processing, message is returned to the queue for another consumer to pick it up or try again



Thursday, July 25, 13

Furthermore, Queues and exchanges can be made “durable”, meaning they’ll survive a restart or server crash with messages intact



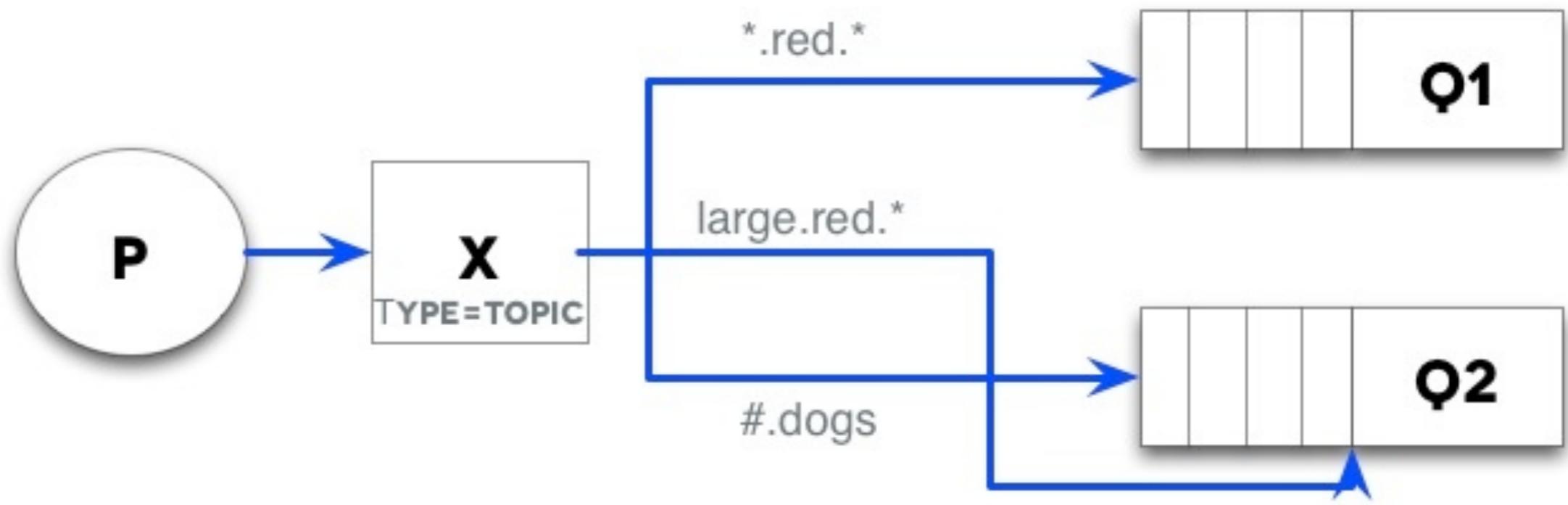
Thursday, July 25, 13

RabbitMQ nodes can be clustered  
-federated-> shared / copied  
-shoveled -> pushed



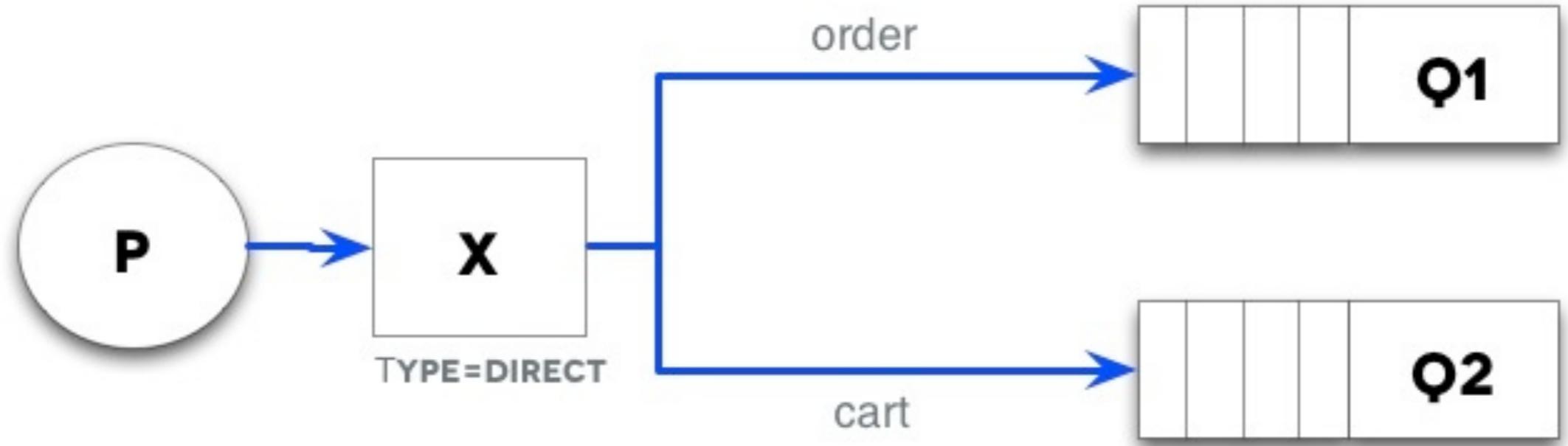
Thursday, July 25, 13

There are several message delivery strategies one can set up for their exchanges



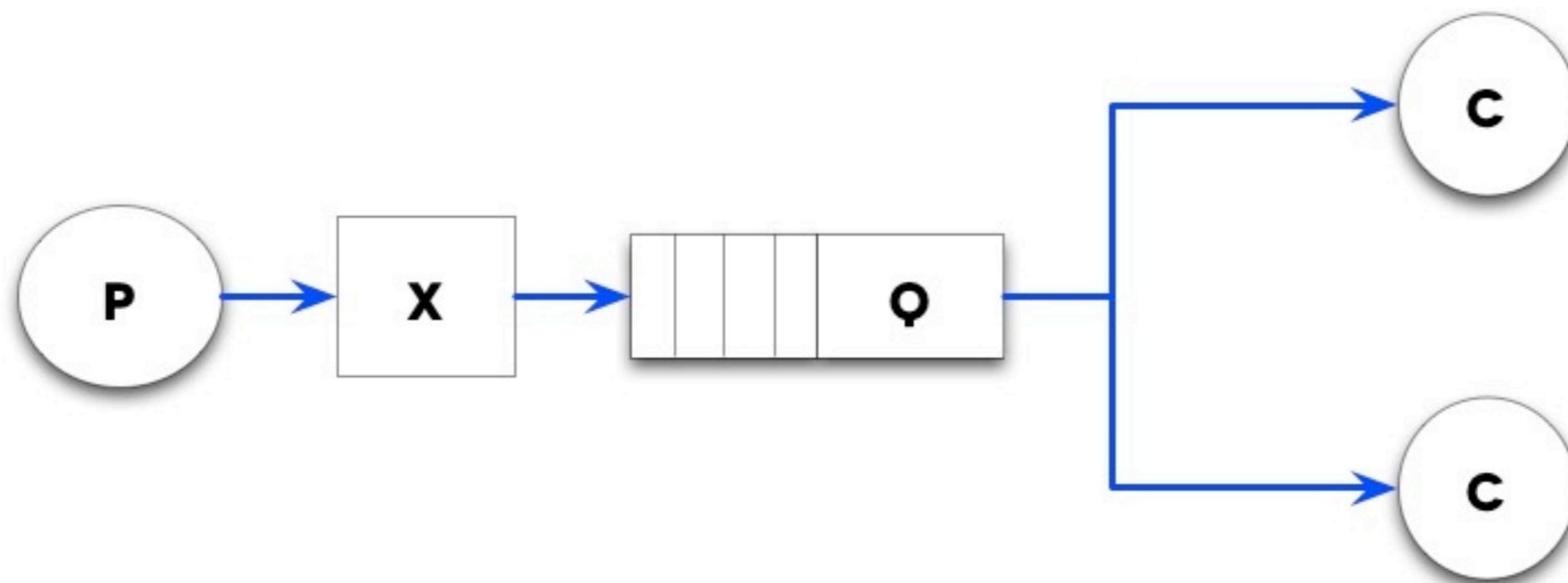
Thursday, July 25, 13

With the whole purpose of scaring you, here's the Topic Queue, the most complicated  
 -gets complicated  
 -allows for all sorts of wild routing schemes  
 -all messages go to the BACK of the queue



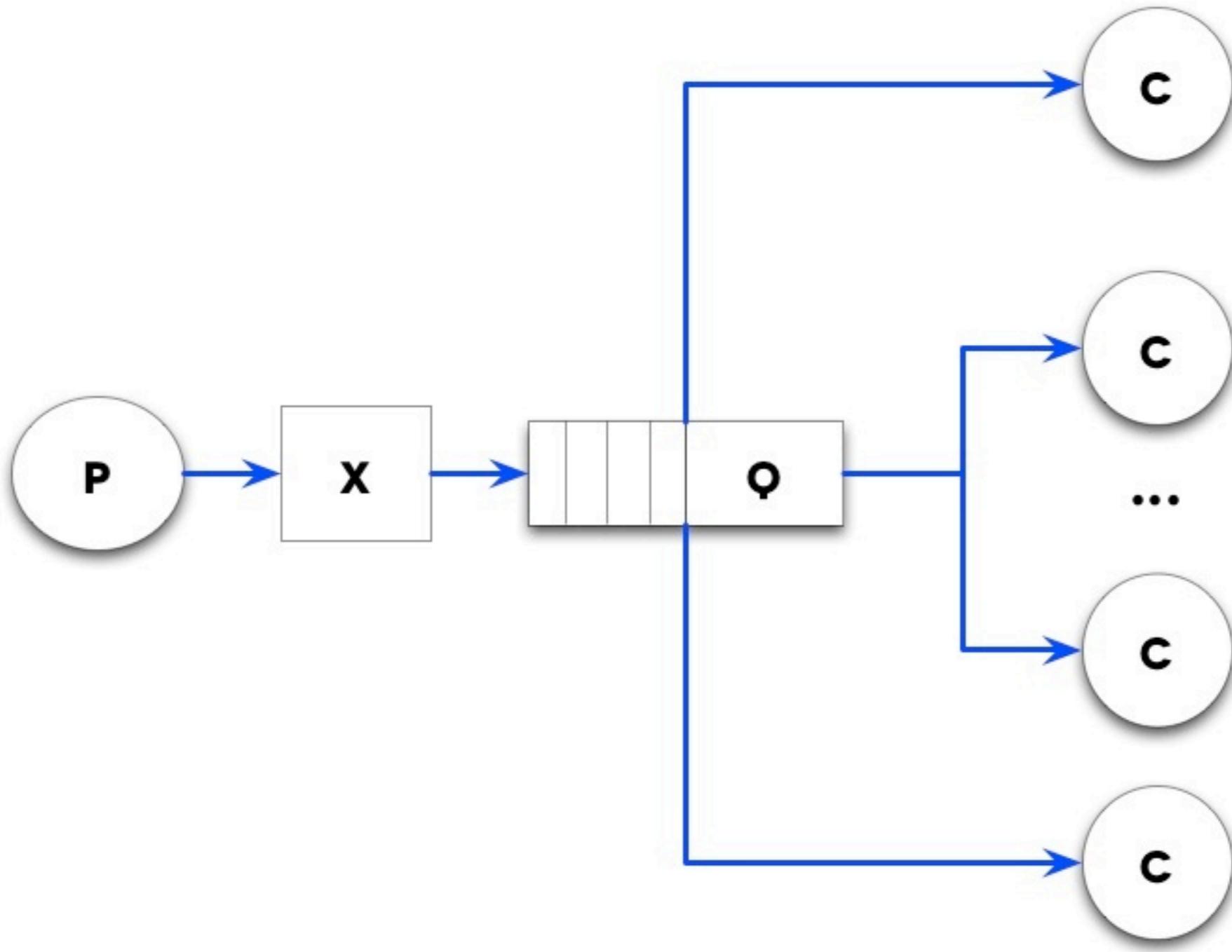
Thursday, July 25, 13

-may be the most useful, I think



Thursday, July 25, 13

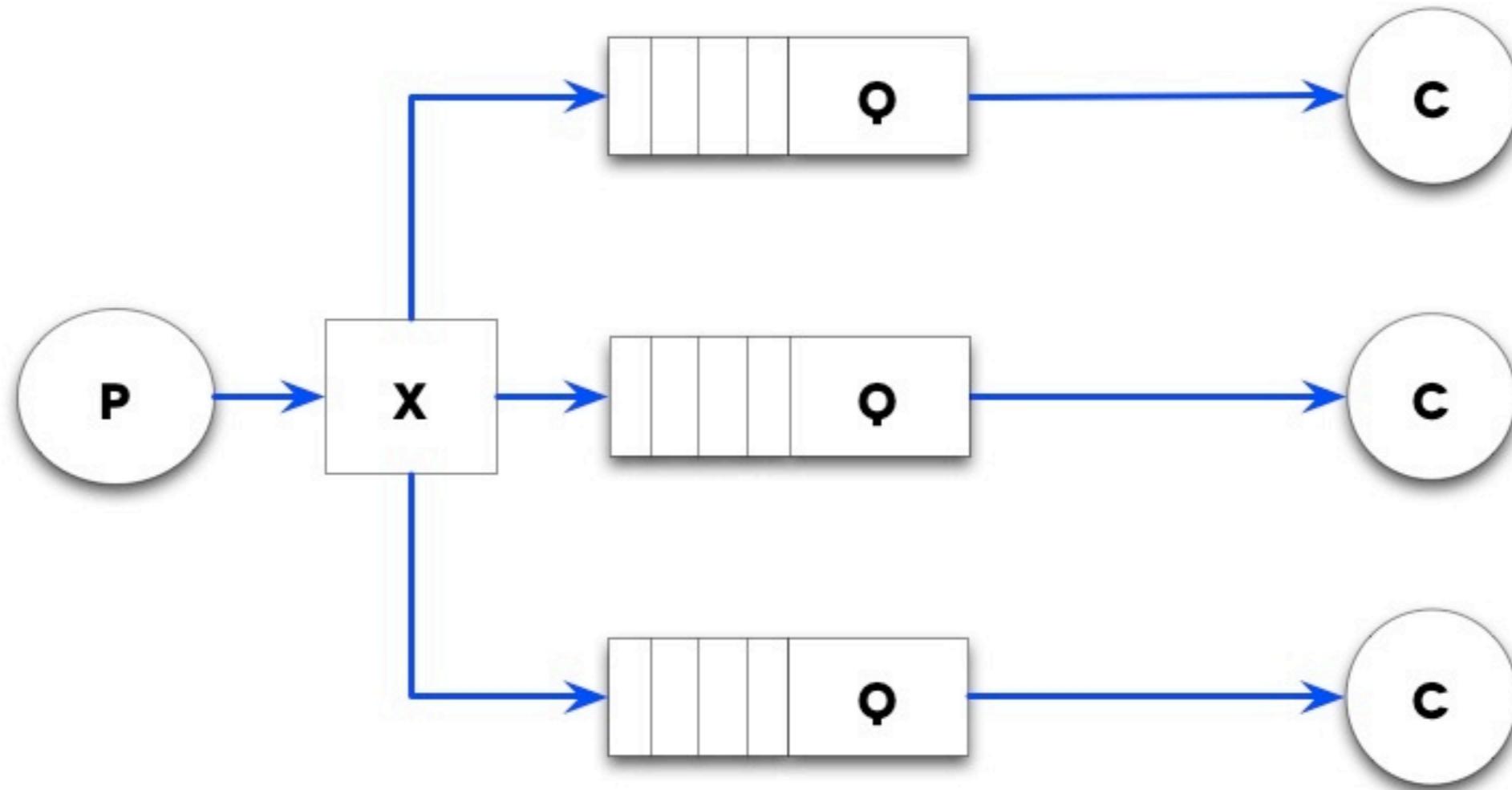
Work Queue



Thursday, July 25, 13

- Work queues operate on a first-come-first serve approach
- I think this is the coolest, one can monitor the state of the exchange and programmatically spin up or down the number of consumer nodes
- All of these consumers are peers...

# **Work Queues** will likely be your default pattern



Thursday, July 25, 13

- Finally, there's 'fanout', which distributes a single message to all attached queues. Good for a general broadcast notification / pub-sub to all your nodes.
- there needs to be a one-to-one queue to consumer relation, though, I believe.
- A special use, case, really.

# Back to the Story

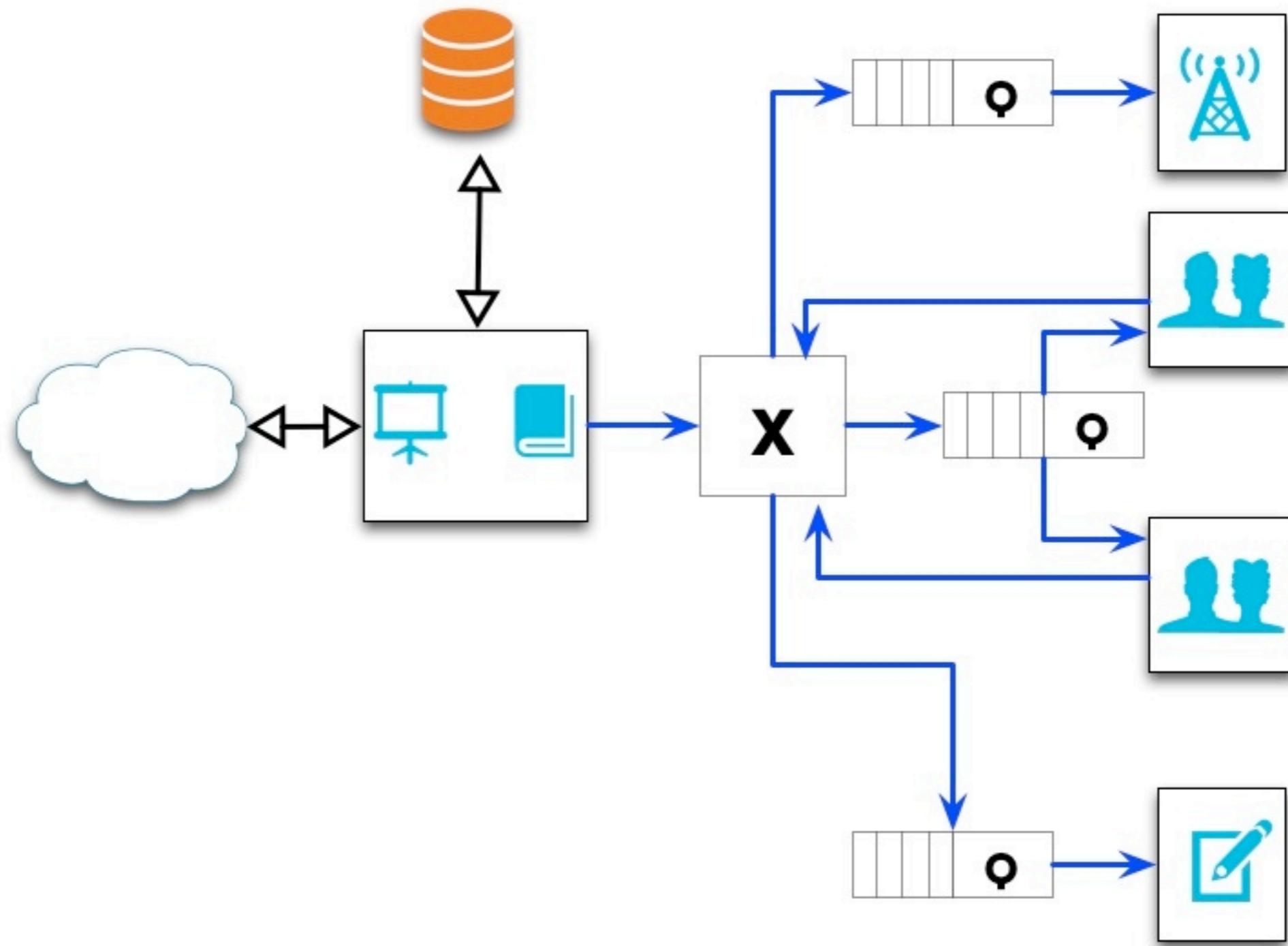
Thursday, July 25, 13

Anyway, back to the story

# A Sample Diagram:

Thursday, July 25, 13

Back to the redacted client. How'd we end up doing it?



Thursday, July 25, 13

- describe each node
- using only two developers
- in addition, they built the browsing as a SPA, and built a native iPhone app against the resulting API

Now, Pray to the demo gods.



**THANK YOU!**

**QUESTIONS?**

# Image Credits

Complexity (pipes): <http://www.flickr.com/photos/bhophoto/384574407/>

Simple Pipeline: <http://tierracos.com/our-companies/tierra-pipeline/>

Highway: [http://farnorthdallas.advocatemag.com/wp-content/uploads/2013/04/882349\\_500402383328407\\_539732406\\_o.jpg](http://farnorthdallas.advocatemag.com/wp-content/uploads/2013/04/882349_500402383328407_539732406_o.jpg)

Homer & doughnuts: <http://thechurchillreview.blogspot.com/2012/10/feeling-terror-too-young-aka-kiddie.html>

Tom Brady: <http://stamperdad.wordpress.com/category/tom-brady/>

Telephone Exchange Operator: <http://fineartamerica.com/featured/telephone-exchange-1920s-granger.html>

People in Queue: <http://www.guardian.co.uk/money/2010/mar/23/doe-queue-jobseekers-online>

Cookie Monster: [http://muppet.wikia.com/wiki/Cookie\\_Monster\\_Through\\_the\\_Years](http://muppet.wikia.com/wiki/Cookie_Monster_Through_the_Years)

Scrooge McDuck: <http://smallbusiness.uprinting.com/how-pennies-are-hurting-small-business/>

Too Many Cooks: <http://www.ecommercesystems.com/featured-articles/cooks-kitchen-driving-e-commerce-business/>

Clustered Macs: <http://www.uiowa.edu/mihpclab/micg.html>

Resuscitate: <http://www.dailymail.co.uk/health/article-2034160/Do-resuscitate-Theyre-fateful-words-meaning-doctors-wont-try-save-you-collapse-hospital.html>

Iron Man: <http://images.wikia.com/marvelmovies>

Mail Sorting: <http://riversidechamber.files.wordpress.com>