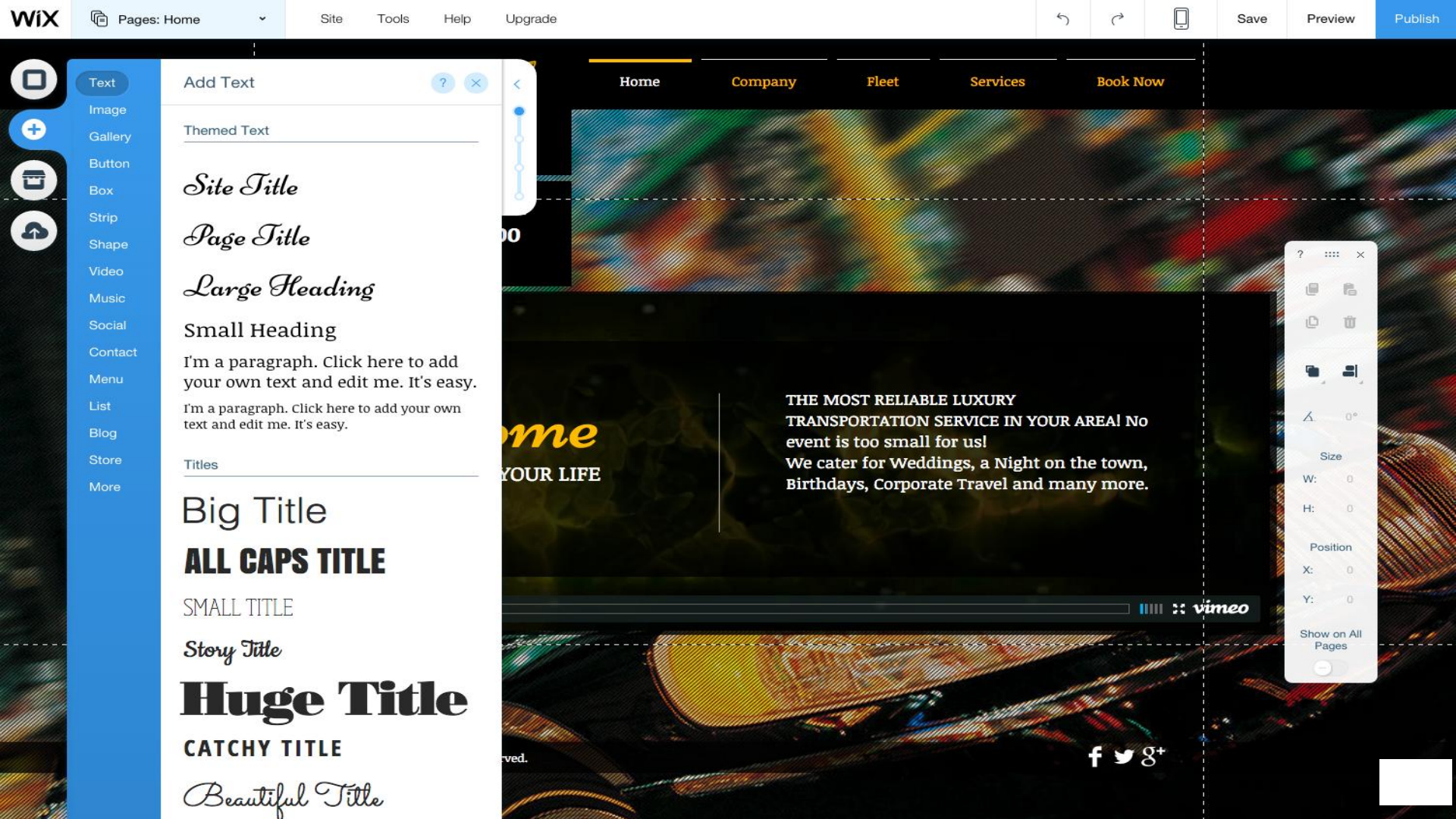
Microservices and DevOps Journey at Wix.com

Aviran Mordo
Head of WixEngineering

www.linkedin.com/in/aviran

@aviranm

http://www.aviransplace.com


Pages: Home
Site
Tools
Help
Upgrade
Save
Preview
Publish
Text
Image
Gallery
Button
Box
Strip
Shape
Video
Music
Social
Contact
Menu
List
Blog
Store
More
Add Text
Themed Text
Site Title
Page Title
Large Heading
Small Heading
I'm a paragraph. Click here to add your own text and edit me. It's easy.
I'm a paragraph. Click here to add your own text and edit me. It's easy.
Titles
Big Title
ALL CAPS TITLE
SMALL TITLE
Story Title
Huge Title
CATCHY TITLE
Beautiful Title
Home
Company
Fleet
Services
Book Now
THE MOST RELIABLE LUXURY TRANSPORTATION SERVICE IN YOUR AREA! No event is too small for us!
We cater for Weddings, a Night on the town, Birthdays, Corporate Travel and many more.
vimeo
Size
W:
H:
Position
X:
Y:
Show on All Pages

Wix In Numbers

- Over 80M users
- Static storage is >2Pb of data
- 3 data centers + 3 clouds (Google, Amazon, Azure)
- 2B HTTP requests/day
- 1000 people work at Wix

Over 200 Microservices on Production

Microservices - What Does it Take

How to Get There? (Wix's journey)

http://gpstrackit.com/wp-content/uploads/2013/11/VanishingPointwRoadSigns.jpg

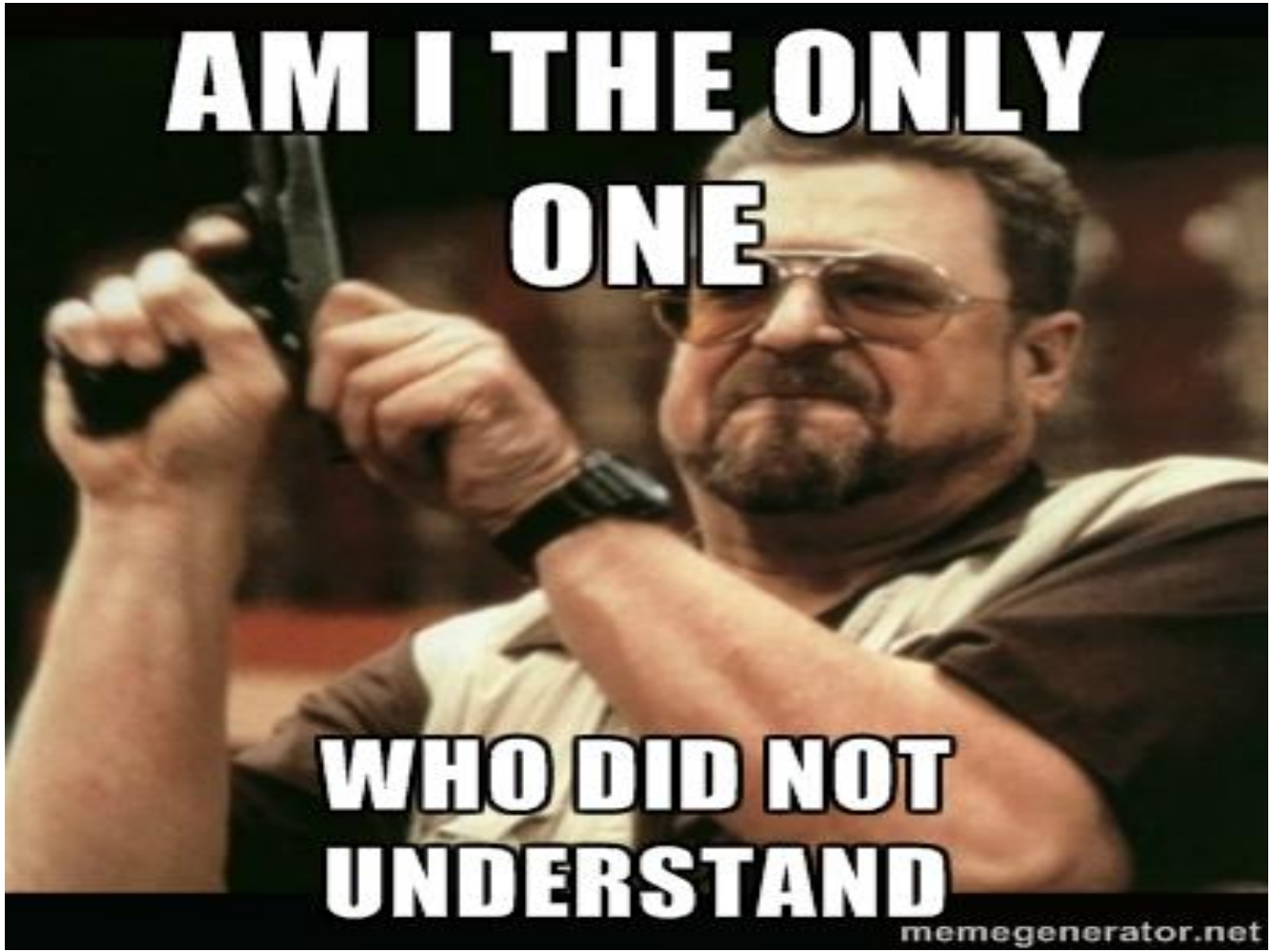

In the Beginning of Time
About 5 years ago
http://p1.pichost.me/i/11/1339236.jpg

The Monolithic Giant

- One monolithic server that handled everything
- Dependency between features
- Changes in unrelated areas caused deployment of the whole system
- Failure in unrelated areas will cause system wide downtime

Breaking the System Apart

https://upload.wikimedia.org/wikipedia/commons/6/67/Broken_glass.jpg


AM I THE ONLY ONE
WHO DID NOT UNDERSTAND
memegenerator.net

Concerns and SLA

Edit websites

- Many feature request
- Lower performance requirement
- Lower availability requirement
- Write intensive

View sites, created by Wix editor

- Not many product changes
- High performance
- High availability
- Read intensive

Phase 1

Extract Public Service

Editor service (Mono-Wix)

Public service

Divide and Conquer

Editor service

Public service

Guideline: No runtime, deployment or data dependency

Separation by Product Lifecycle

- Decouple architecture => Decouple teams
- Deployment independence
- Areas with frequent changes

Editor service

Public service

Separation by Service Level

- Scale independently
- Use different data store
- Optimize data per use case (Read vs Write)
- Run on different datacenters / clouds / zones
- System resiliency (degradation of service vs. downtime)
- Faster recovery time

Editor service

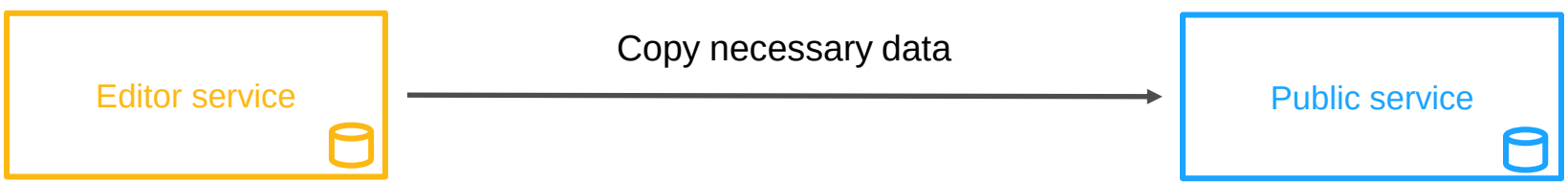
Public service

http://blogs.adobe.com/captivate/2011/03/training-adding-interactivity-to-elearning-courses-with-adobe-captivate-5.html/time-to-learn-clock

Service Boundary

Separation of Databases

- Copy data between segments
- Optimize data per use case (read vs. write intensive)
- Different data stores



Serialization

Serialization / Protocol

- Binary?
- JSON / XML / Text?
- HTTP?

Serialization / Protocol - Tradeoffs

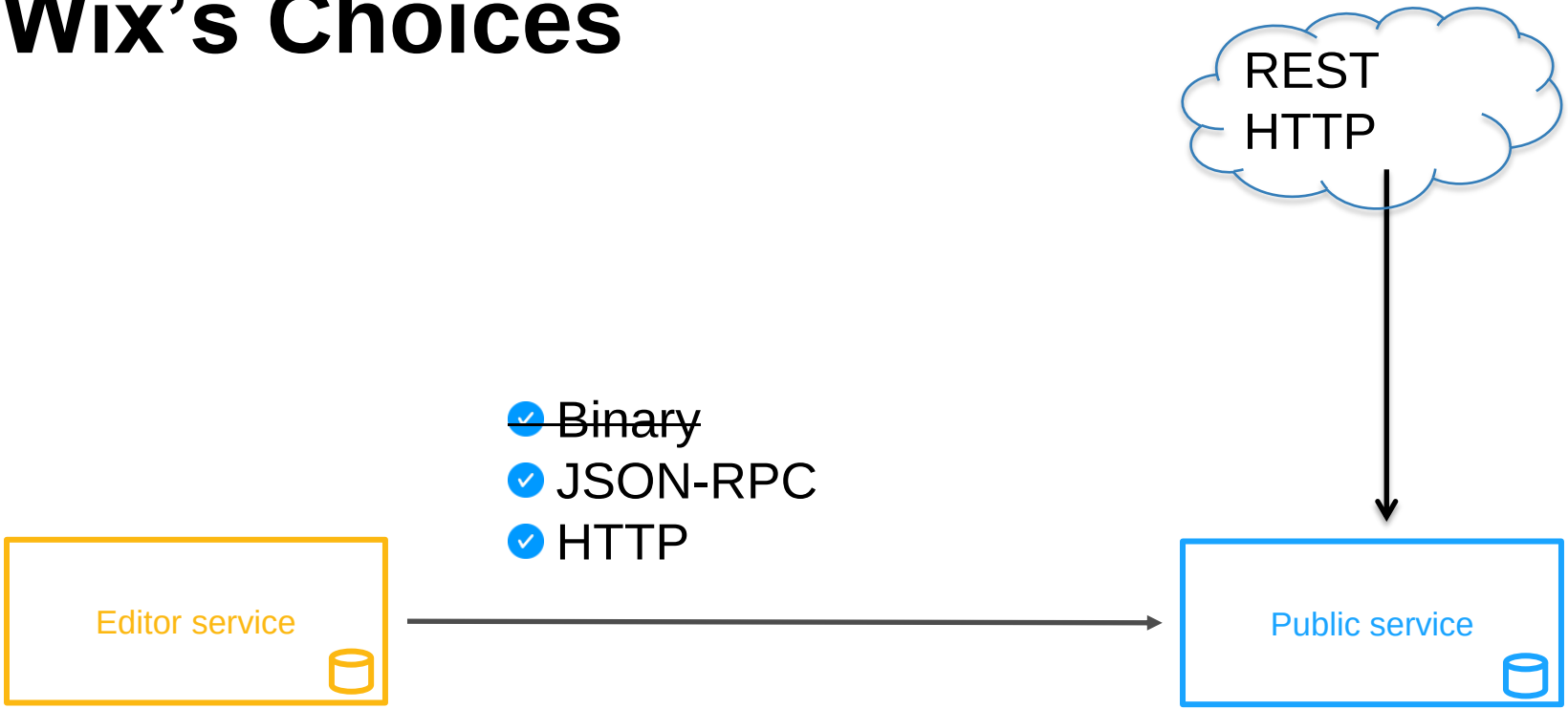
- Readability?
- Performance?
- Debug?
- Tools?
- Monitoring?
- Dependency?

API Transport/Protocol

How to Expose an API

- REST?
- RPC?
- SOAP?

Wix's Choices

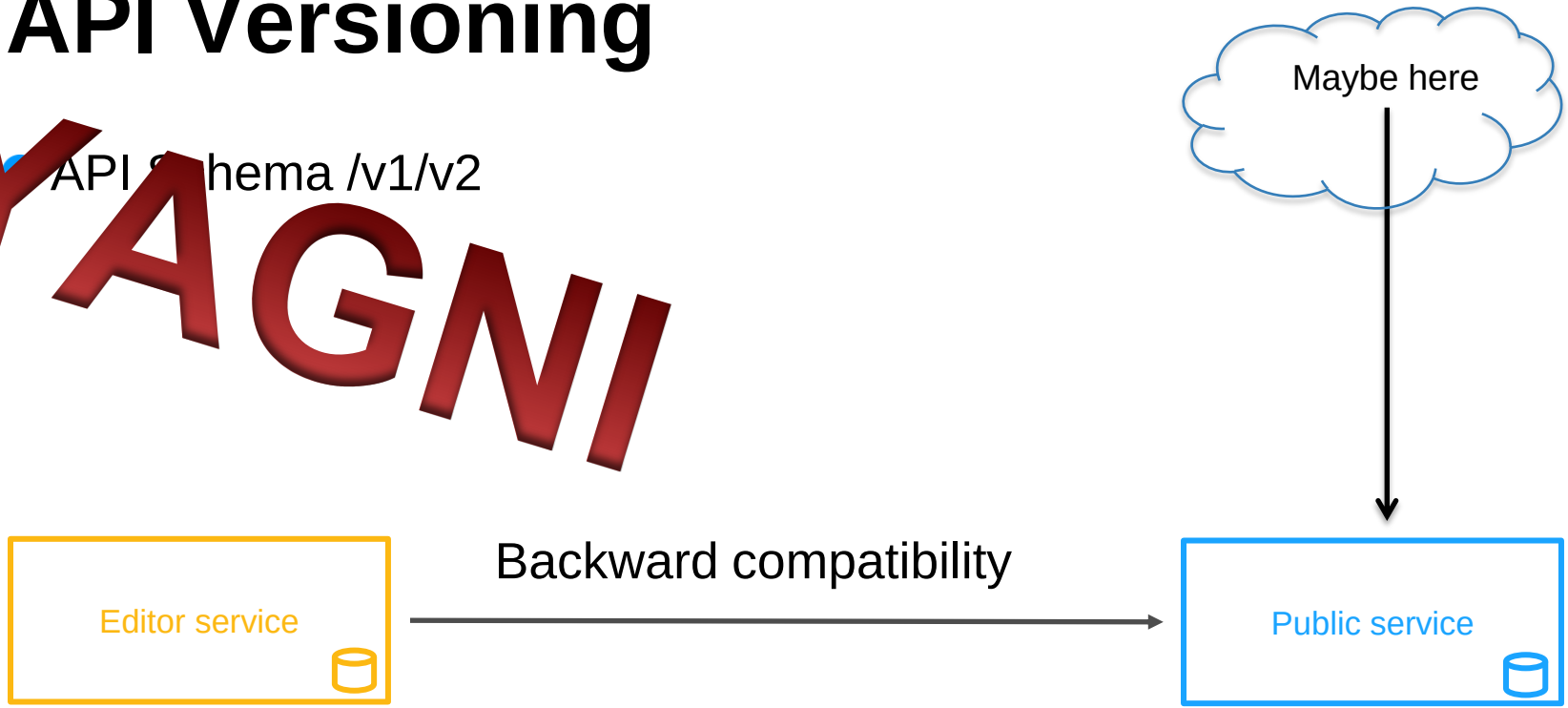


API Versioning

API Versioning

- API Schema /v1/v2

YAGNI



A-Synchronous

Which Queuing System to Use

- Kafka?
- RabbitMQ?
- ActiveMQ?
- ???

YAGNI

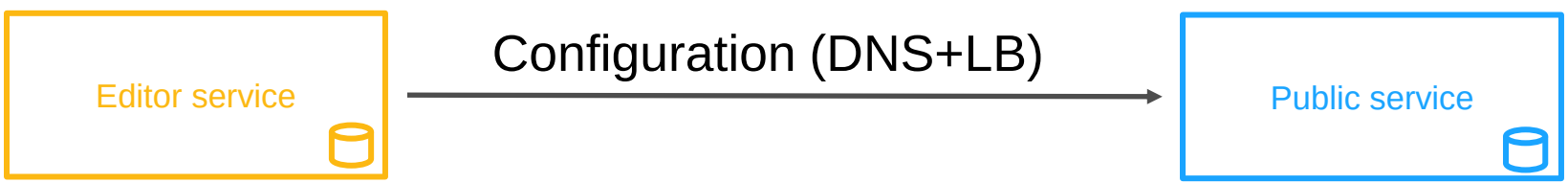
devops

Service Discovery

Service Discovery

- Zookeeper?
- Consul
- Etcd?
- Eureka?

YAGNI devops



Resilience

What does the Arrow Mean?

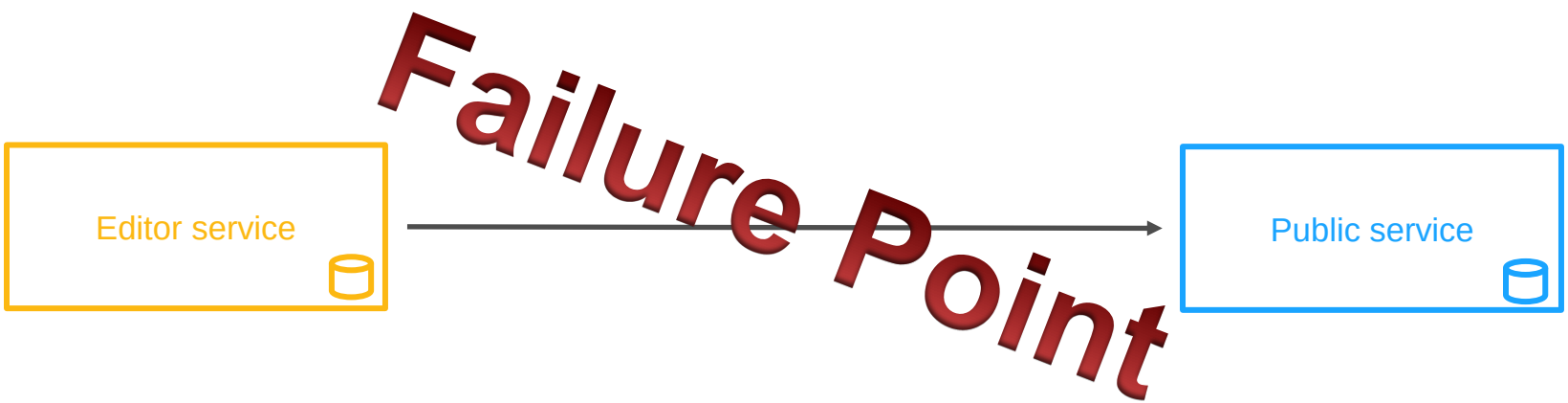
Every Network Hop Reduces Availability

Failure Points = Network I/O

- Retry policy
- Circuit breaker
- Throttlers

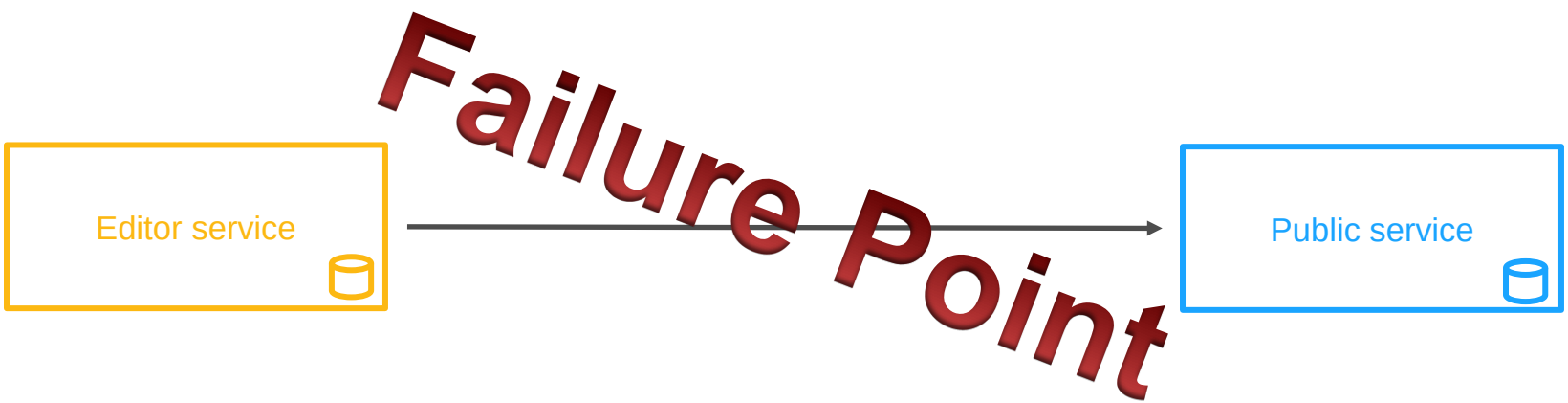
Retry only on idempotent operations

Be careful – you may cause downtime



Degradation of Service

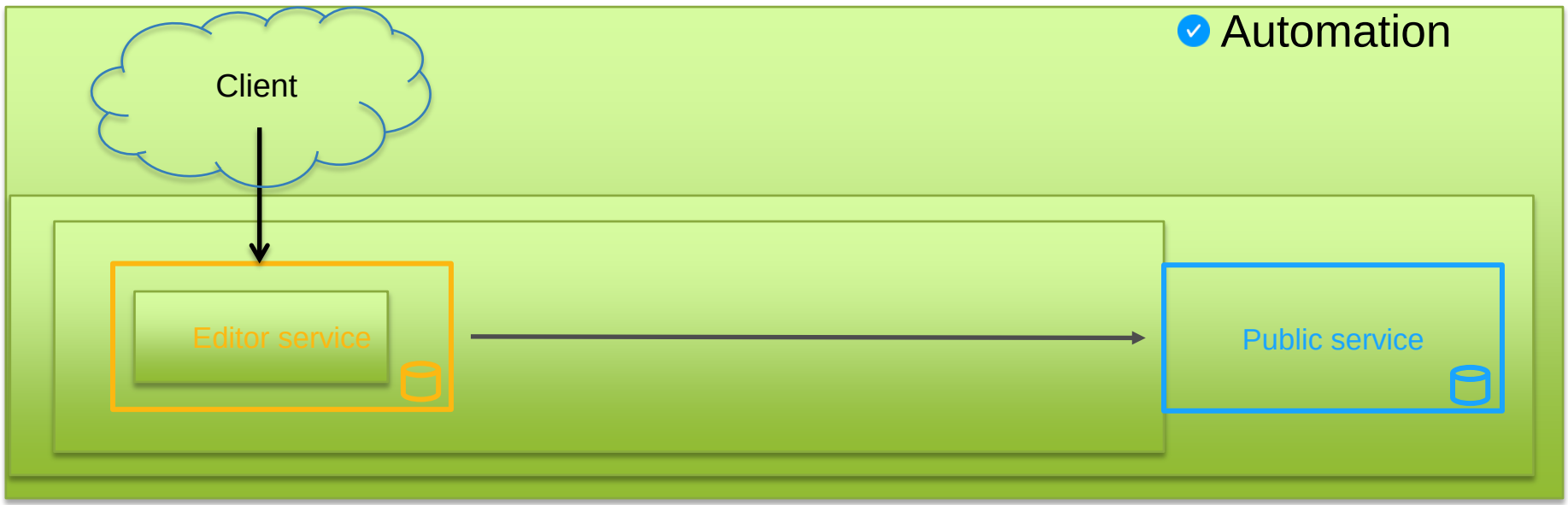
- Feature killer (Killer feature)
- Fallbacks
- Self healing



Testing

Test a Distributed System (at Wix)

- Unit Test
- Integration Test
- Server E2E
- Automation



YAGNI

Distributed Logging

Build visibility into service

Metrics | Resource Pools | ActiveMQ Stats | Local TPX List

Application Information

Title	App Integration Bus – WebApp
Artifact	app-integration-bus-web
Version	1.179.0-SNAPSHOT
Build	cd45eff4f2b3bddda09ffc1d7bf30597782458e8
Build Timestamp	20140608-1049
Server Name	[redacted]
Uptime	3 days, 20 hours and 21 minutes.
Server Time Zone	America/Chicago
Server Startup	08/06/2014 07:05:30.421
Server Current Time	12/06/2014 03:27:20.653

Usage Summary (Incoming Calls Only)

	1h	48h
Total Calls	15	713
Total Successfull calls	15	708
Throughput	0.3 rpm	0.2 rpm
Error Rate	0.00 %	0.70 %
System Errors	0/0/0/0	0/0/0/5
Business Errors	0/0/0/0	0/0/0/0

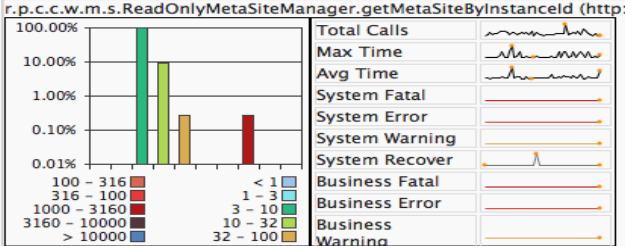
Prev. Startup Information

Version 1.178.0-SNAPSHOT	01/06/2014 09:30:42.651
Version 1.176.0-SNAPSHOT	29/05/2014 03:15:30.798
Version 1.175.0-SNAPSHOT	29/05/2014 01:39:05.230

Usage Statistics

Method		Calls	Throughput (rpm)	Average (mSec)	Max (mSec)	System Errors	Business Errors
METER c.w.a.d.CachedReadOnlyLocalAppsRepository.getAppByIdOpt	48h	277,611	97.46	0.0	77.8	0/0/0/0	0/0/0/0
	1h	1,037	17.78	0.0	0.0	0/0/0/0	0/0/0/0
RPC_CLIENT r.p.c.c.w.m.s.ReadOnlyMetaSiteManager.getMetaSiteByInstanceId (http://a	48h	76,499	26.86	8.3	6,505.0	1/0/0/0	0/0/0/0
	1h	370	6.34	10.7	1,088.0	0/0/0/0	0/0/0/0

r.p.c.c.w.m.s.ReadOnlyMetaSiteManager.getMetaSiteByInstanceId (http://api.aus.wixpress.com/meta-site-manager/ReadOnlyMetaSiteManager)



1h Call Times (msec.) 48h Summary

time	Exception	Message
11/06/2014 01:38:51.808	com.wixpress.framework.rpc.client.exceptions.RpcTransportException	c.w.f.r.c.e.RpcTransportException – Server connection timed out [endpoint= uri=\'http://api.aus.wixpress.com/meta-site-manager/ReadOnlyMetaSiteMa at com.wixpress.framework.rpc.client.RpcOverHttpClient$class.renderFault(R at com.wixpress.framework.rpc.client.BlockingRpcOverHttpClient.renderFaul at com.wixpress.framework.rpc.client.BlockingRpcOverHttpClient$$anonfun$ex at

Ownership

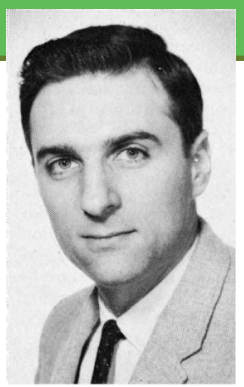
Team Work

- Microservice is owned by a team
- You build it – you run it
- No microservice is left without a clear owner
- Microservice is NOT a library – it is a live production system

What is the Right Size of a Microservice?

The Size of a Microservice is the Size of the Team That is Building it.

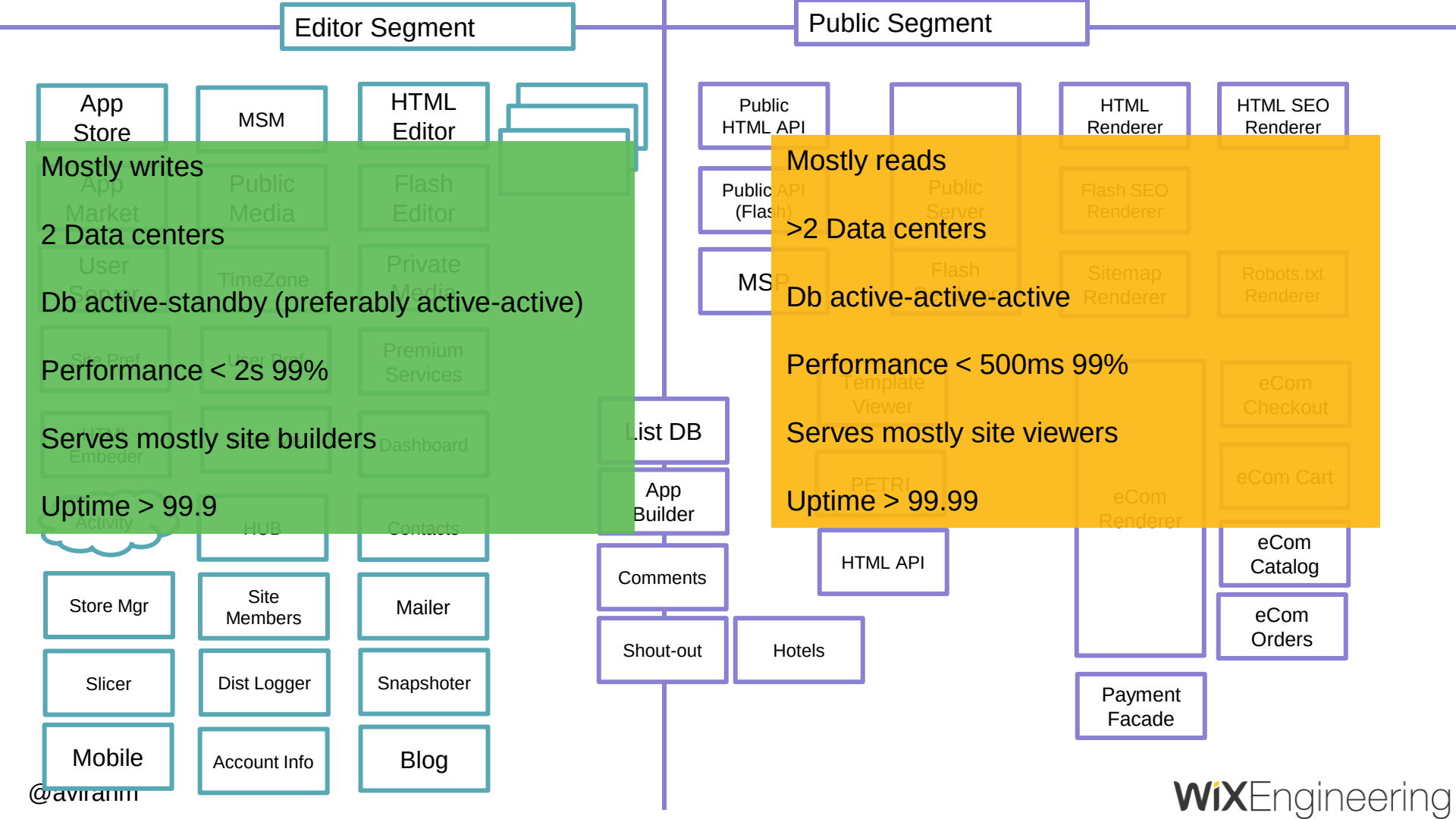
“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations”
Conway, Melvin



What did you Learn from Just 2 Services

- Service boundary
- Monitoring infrastructure
- Serialization format
- Synchronous communication protocol (HTTP/Binary)
- Asynchronous (queuing infra)
- Service SLA
- API definition (REST/ RPC / Versioning)
- Data separation
- Deployment strategy
- Testing infrastructure (integration test, e2e test)
- Compatibility (backwards / forward)

Continue to Extract More Microservices


Editor Segment
Public Segment
App Store
MSM
HTML Editor
Public HTML API
HTML Renderer
HTML SEO Renderer
Mostly writes
2 Data centers
Db active-standby (preferably active-active)
Performance < 2s 99%
Serves mostly site builders
Uptime > 99.9
Mostly reads
>2 Data centers
Db active-active-active
Performance < 500ms 99%
Serves mostly site viewers
Uptime > 99.99
List DB
App Builder
Comments
Shout-out
Hotels
HTML API
eCom Checkout
eCom Cart
eCom Catalog
eCom Orders
Payment Facade
Store Mgr
Site Members
Mailer
Slicer
Dist Logger
Snapshoter
Mobile
Account Info
Blog
@aviranm
WIXEngineering

When to Extract a New Microservice

Microservice or Library?

I need time zone from an IP address

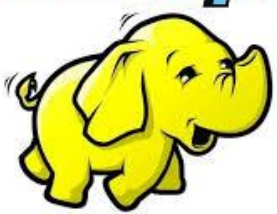
- Do I create deployment dependency?
- What is DevOps overhead ?
- Who owns it?
- Does it have its own development lifecycle?
- Does it fit the scalability / availability concerns?
- Can a different team develop it?

Microservice has Ops, Library is Only Computational

Which Technology Stack to Use

Free to Chose?

- Microservices gives the freedom to use a different technology stacks.
- Enables innovation

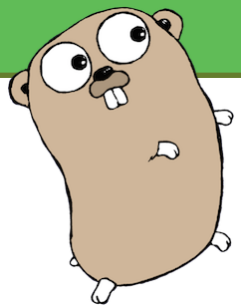


Default to the Stack You Know how to Operate.

Innovate on Non Critical Microservices and Take Full Responsibility for its Operation.

@aviranm

Polyglotic System?



Limit your Stack

- Code reuse
- Cross cutting concerns (session, security, auditing, testing, logging…)
- Faster system evolution
- Development velocity

http://wallpaperbeta.com/dogs_kiss_noses_animals_hd-wallpaper-242054/

What else will you learn

- Distributed transactions
- System monitoring
- Distributed traces
- Tradeoff of a new microservice vs. extending an existing one
- Deployment strategy and dependency
- Handling cascading failures
- Team building/splitting

Summary

Microservices is the First Post DevOps Architecture

@aviranm

WiXEngineering

Every Microservice is a devops Overhead

It is all about trade-off

Microservices Guidelines & Tradeoffs

- Each service has its own DB schema (if one is needed)
  - Gain - Easy to scale microservices based on service level concerns
  - Tradeoff – system complexity, performance
- Only one service should write to a specific DB table(s)
  - Gain - Decoupling architecture – faster development
  - Tradeoff – system complexity / performance
- May have additional read-only services that accesses the DB
  - Gain - Performance gain
  - Tradeoff - coupling
- Services are stateless
  - Gain - Easy to scale out (just add more servers)
  - Tradeoff - performance / consistency

MONOLITH

MICRO SERVICES

Thank You

Q&A

http://goo.gl/32xOTt

Aviran Mordo
Head of Engineering
WIX.com
http://engineering.wix.com
@WixEng

www.linkedin.com/in/aviran
@aviranm
http://www.aviransplace.com