# EVENT-BASED ARCHITECTURE

SCALABLE INTERNET SERVICES
UCSB - NOV 2021

SEAN MALONEY
Amazon AWS
ex Riot Games, ex Appfolio

@SEAN_SEANNERY

seamalo@amazon.com

# WHO IS THIS GUY?

- AWS CodeCommit Lead
- Riot Games Big Data
- UCSB Lead TA
- US Dept of Labor / Energy

FUN FACT:

Was a student in this class many years ago. Intern at Appfolio

# EVENT-BASED ARCHITECTURE

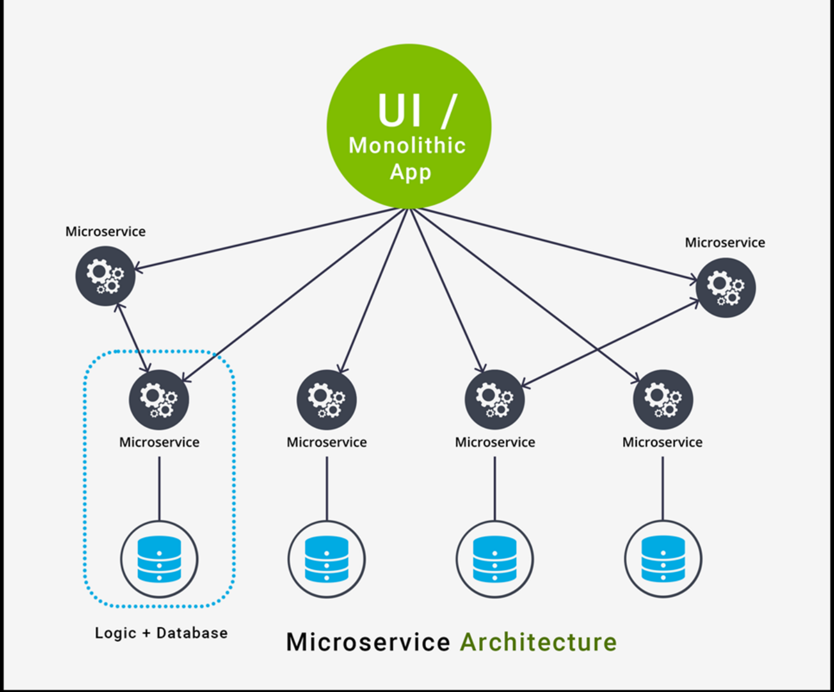
1. THE PROBLEM WITH MICROSERVICES
2. EVENT-BASED ARCHITECTURE
3. CASE STUDY: RIOT GAMES

# THE PROBLEM WITH MICROSERVICES

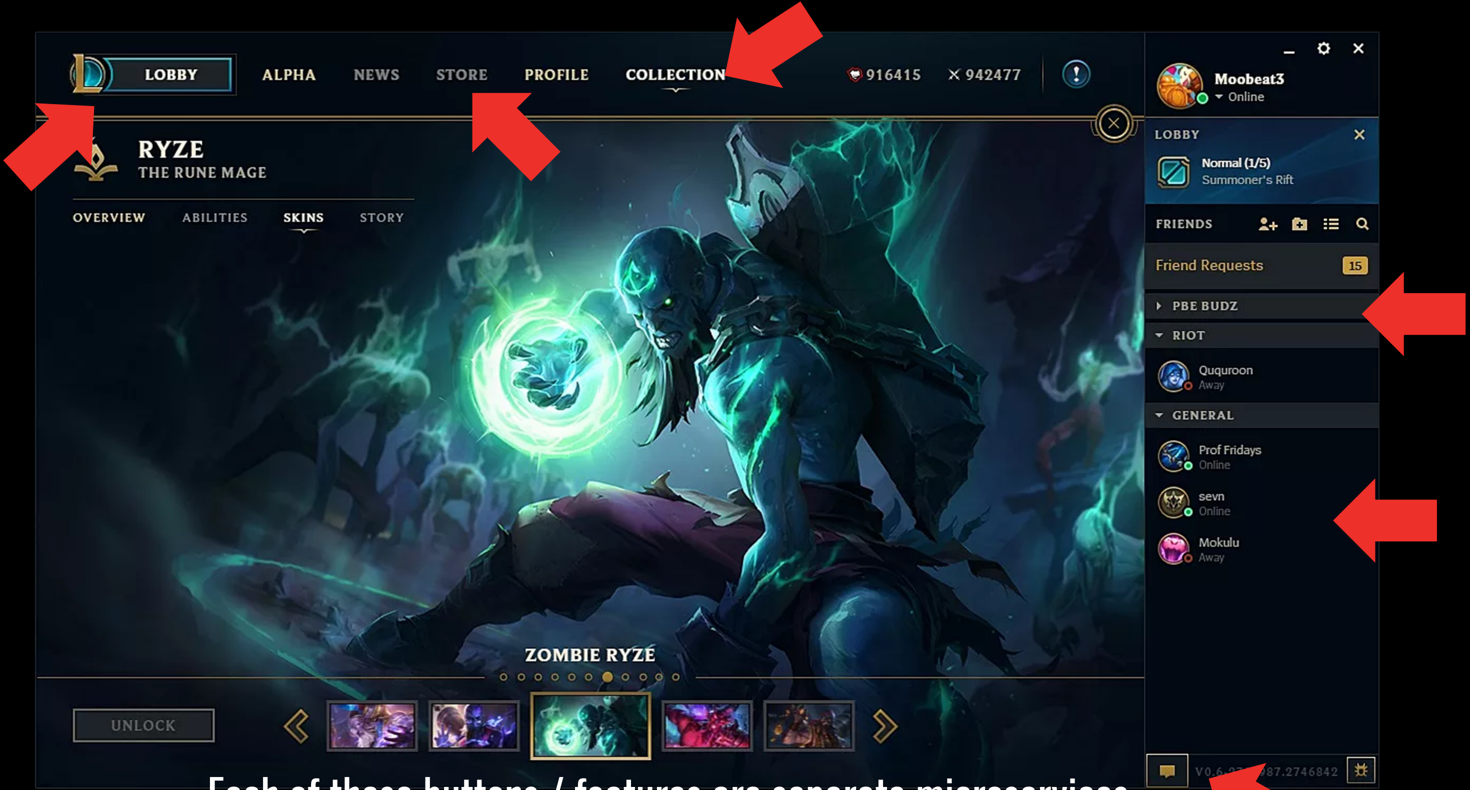
# MICROSERVICE ARCHITECTURE

Loosely Coupled Services responsible for doing one thing well.

Scales well for large enterprises.  One team can own one service. Can scale separately for different traffic



# e.x. League Client

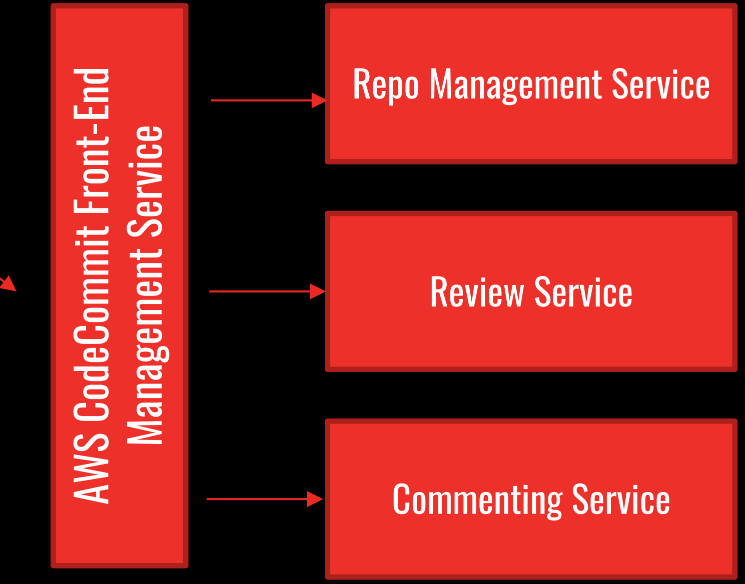


Each of these buttons / features are separate microservices behind the scenes

# e.x. AWS CLI

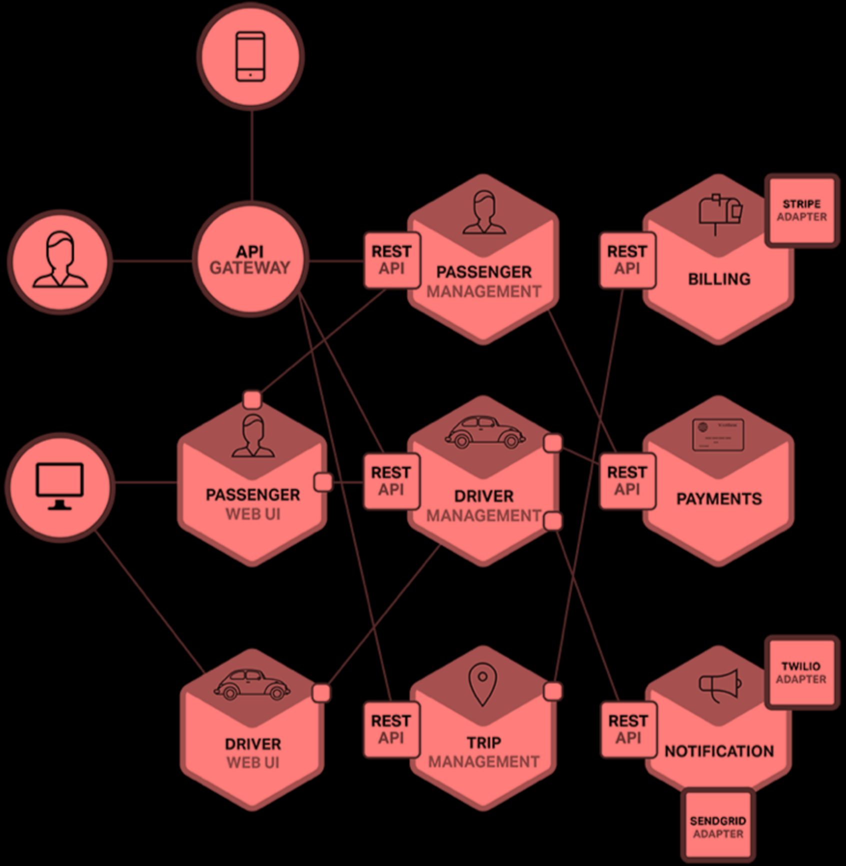
## AWS CodeCommit Command Line Interface:

```
$> aws codecodecommit [subcmd]

--create-repository
--delete-repository
--get-file

--create-pull-request
--delete-pull-request

--create-comment
--delete-comment
```



CodeCommit commands are often handled by separate services.
In case there is an outage, customers could still do pull requests or comments

# How it Breaks



In a large enterprise, many services have dependencies on many other services

What happens when a service dies, or experiences latecy? (like payments svc?)

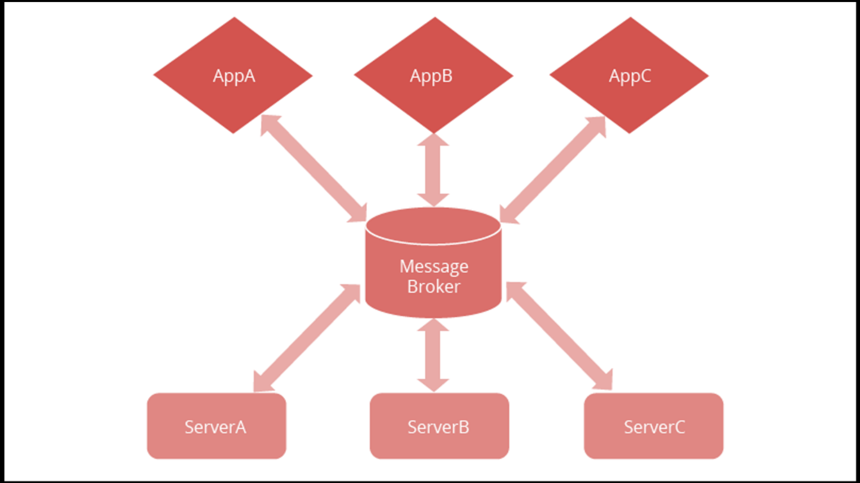
What happens when a service changes it's API structure?

How do I get a unified view of business health?

# EVENT-BASED ARCHITECTURE

# Event-based Architecture

A architectural pattern promoting the production, consumption and processing of events



# Event-based Architecture

HELPS WITH 3 THINGS:

1.) Buffering load between services

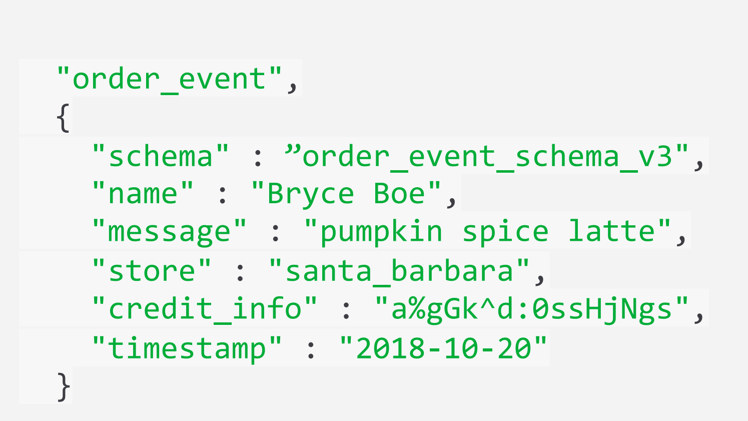
2.) Interoperability / features between services

3.) Up-to-date analytics data
(compared to batch ETLs)

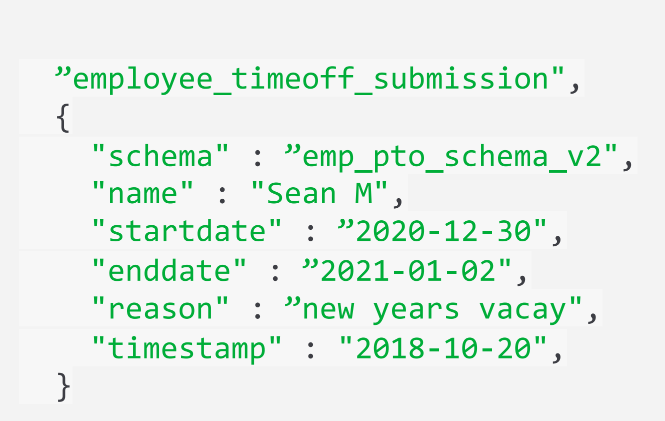
# Event-based Architecture

## What is an event?

Data describing an instance of something happening at a specific point in time



```
"order_event",
{
  "schema" : ”order_event_schema_v3",
  "name" : "Bryce Boe",
  "message" : "pumpkin spice latte",
  "store" : "santa_barbara",
  "credit_info" : "a%gGk^d:0ssHjNgs",
  "timestamp" : "2018-10-20"
}
```



```
”employee_timeoff_submission",
{
  "schema" : ”emp_pto_schema_v2",
  "name" : "Sean M",
  "startdate" : ”2020-12-30",
  "enddate" : ”2021-01-02",
  "reason" : ”new years vacay",
  "timestamp" : "2018-10-20",
}
```

# Event-based Terminology

Streams - an unbounded set of similar events

Producers - generate events and send them off

Consumers - receive events for consumption

Processors - type of consumer that routes or transforms the events and pushes them back into the queue

# A different approach

For example:

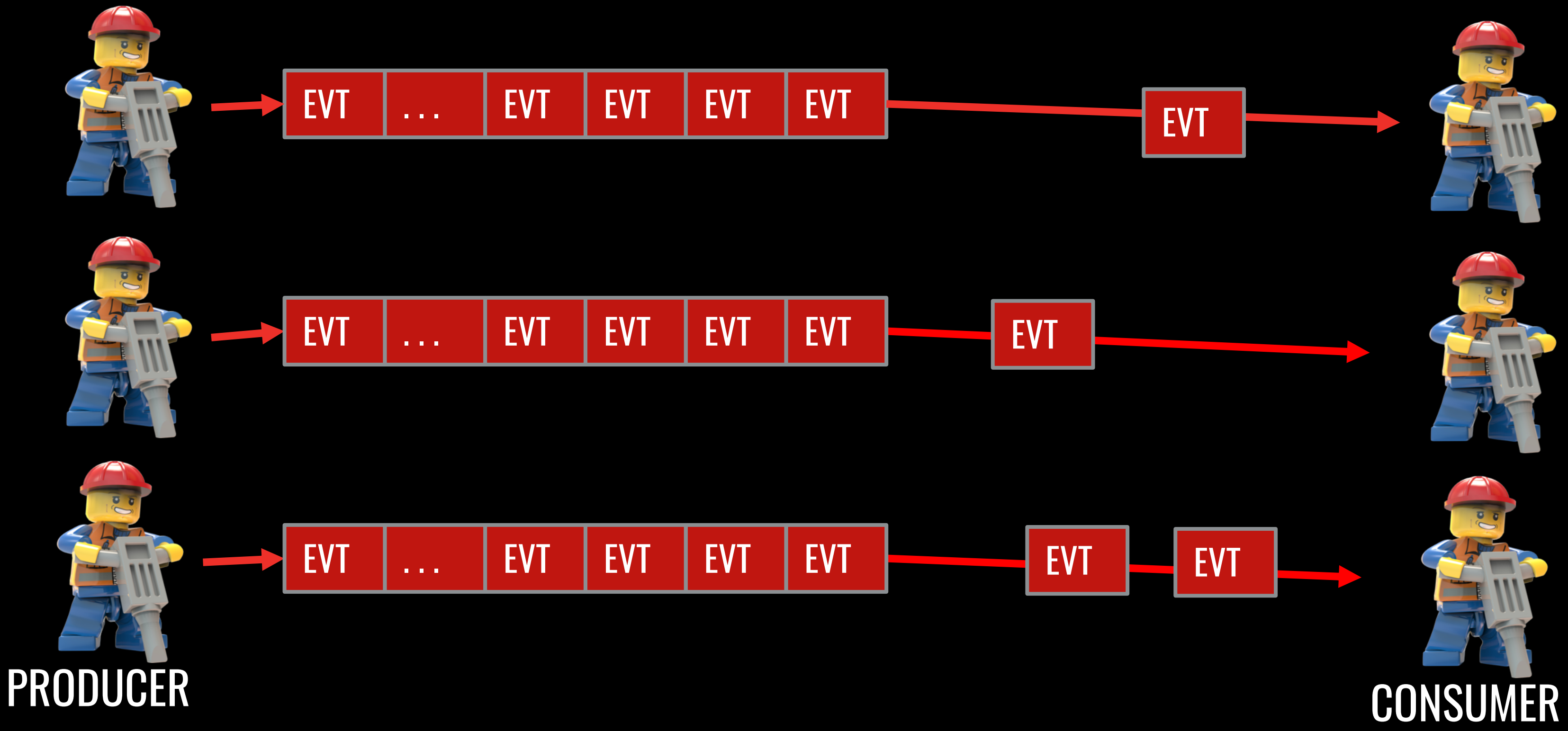
Instead of ...

```
creditcardservice.purchaseItem()
warehouseservice.updateInventory()
```

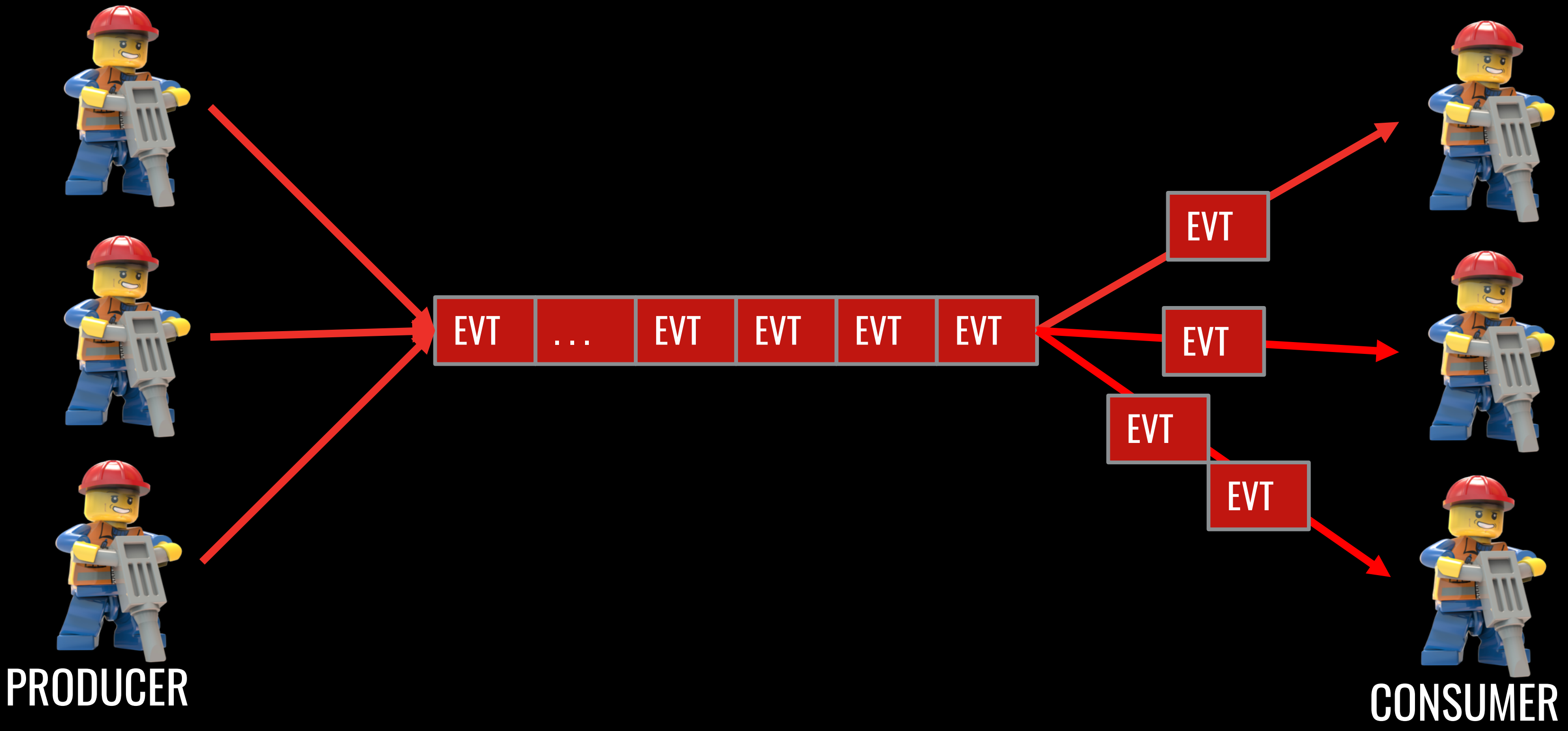
Do ...

```
creditcardservice.processEvent(purchase_event)
warehouseservice.processEvent(purchase_event)
```

# Message Queues



# Event Bus



# Message Queues

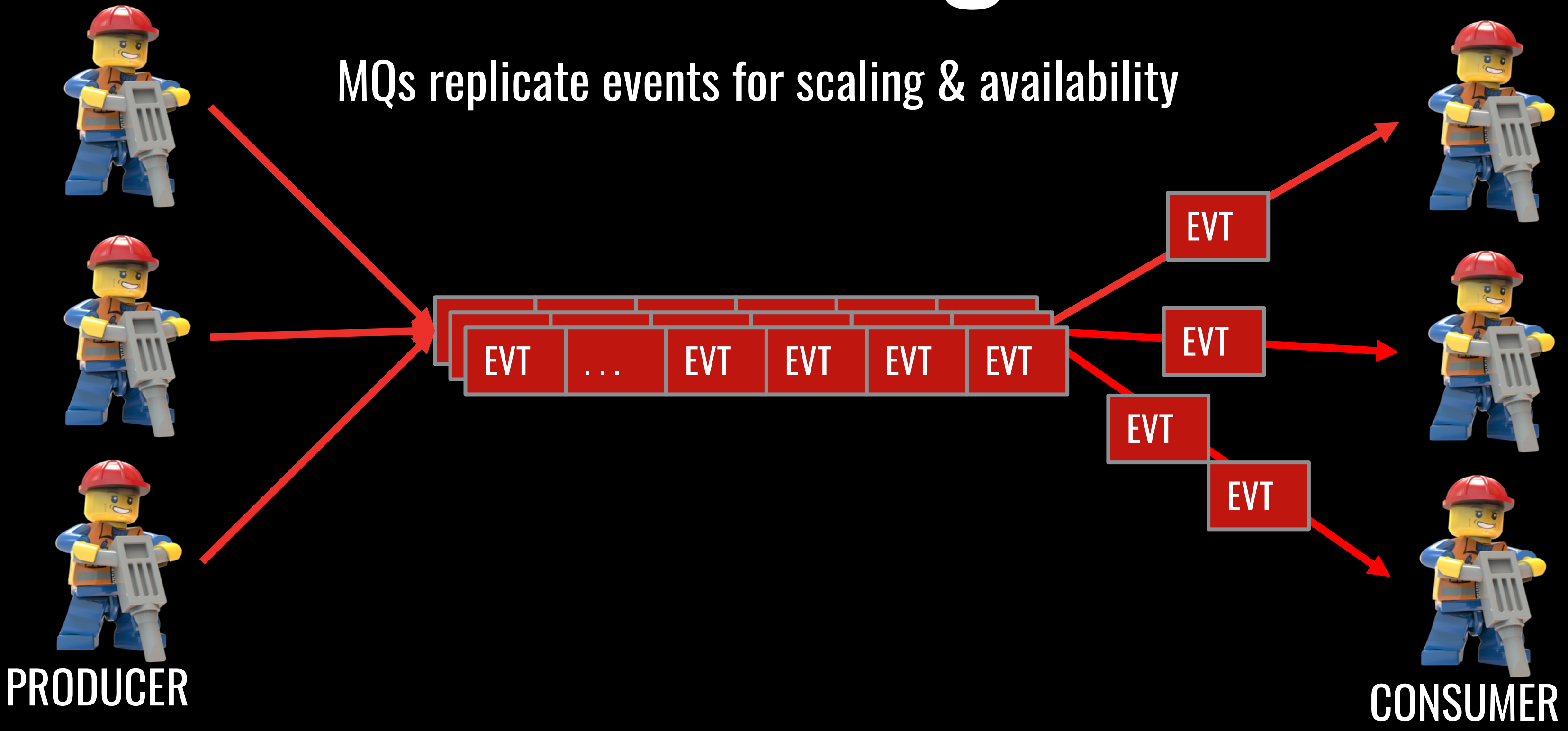
- Google PubSub
- Amazon Simple Queue Service (SQS)
- Kafka
- RabbitMQ
- Amazon Kenisis
- Microsoft MQ  (MSMQ)

# Message Queues

- Redundancy
- Delivery Guarantees
- Easy to Scale
- Asynchronous Communication
- Abstraction / Decoupling

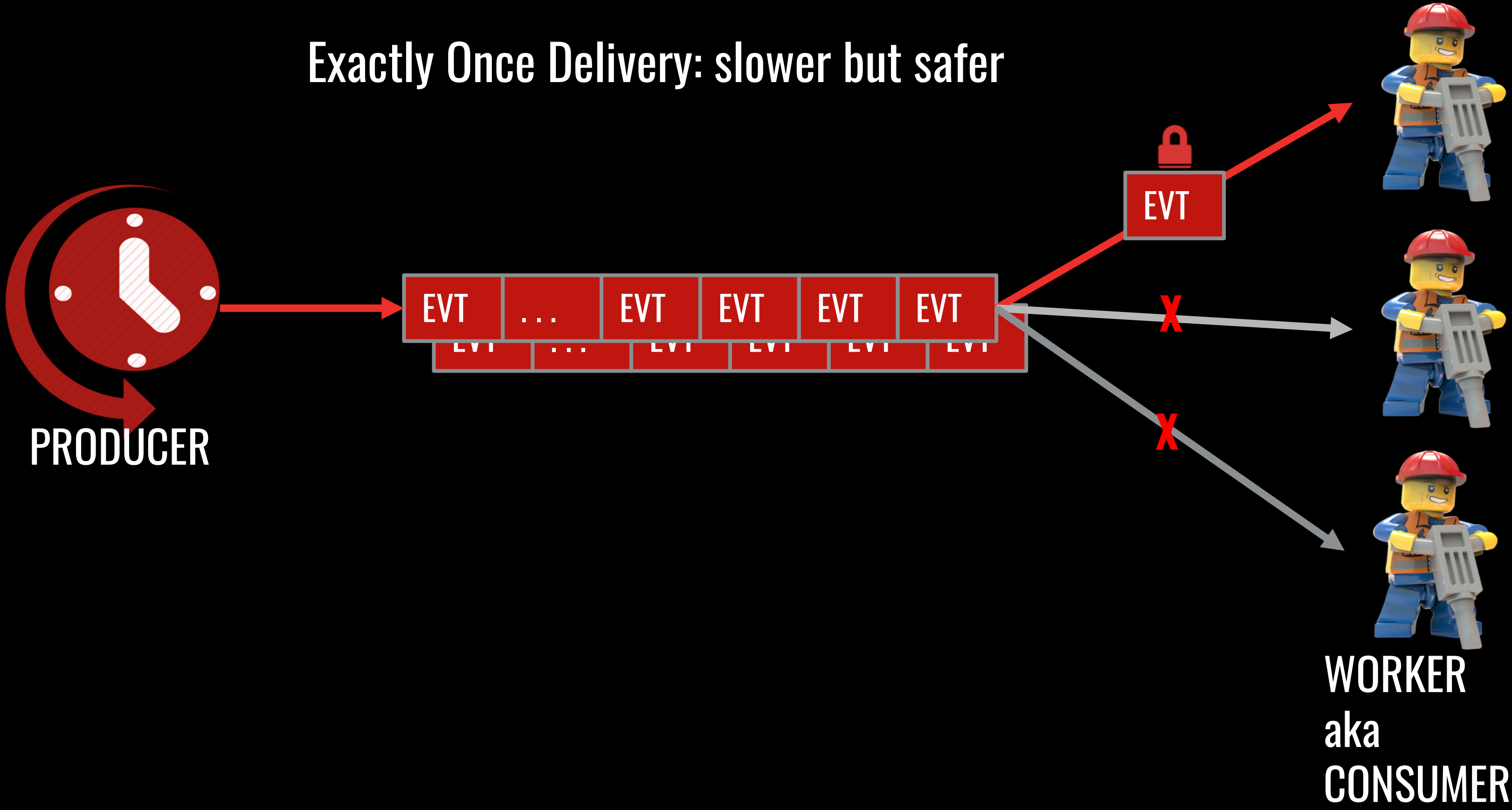
# Scaling

MQs replicate events for scaling & availability



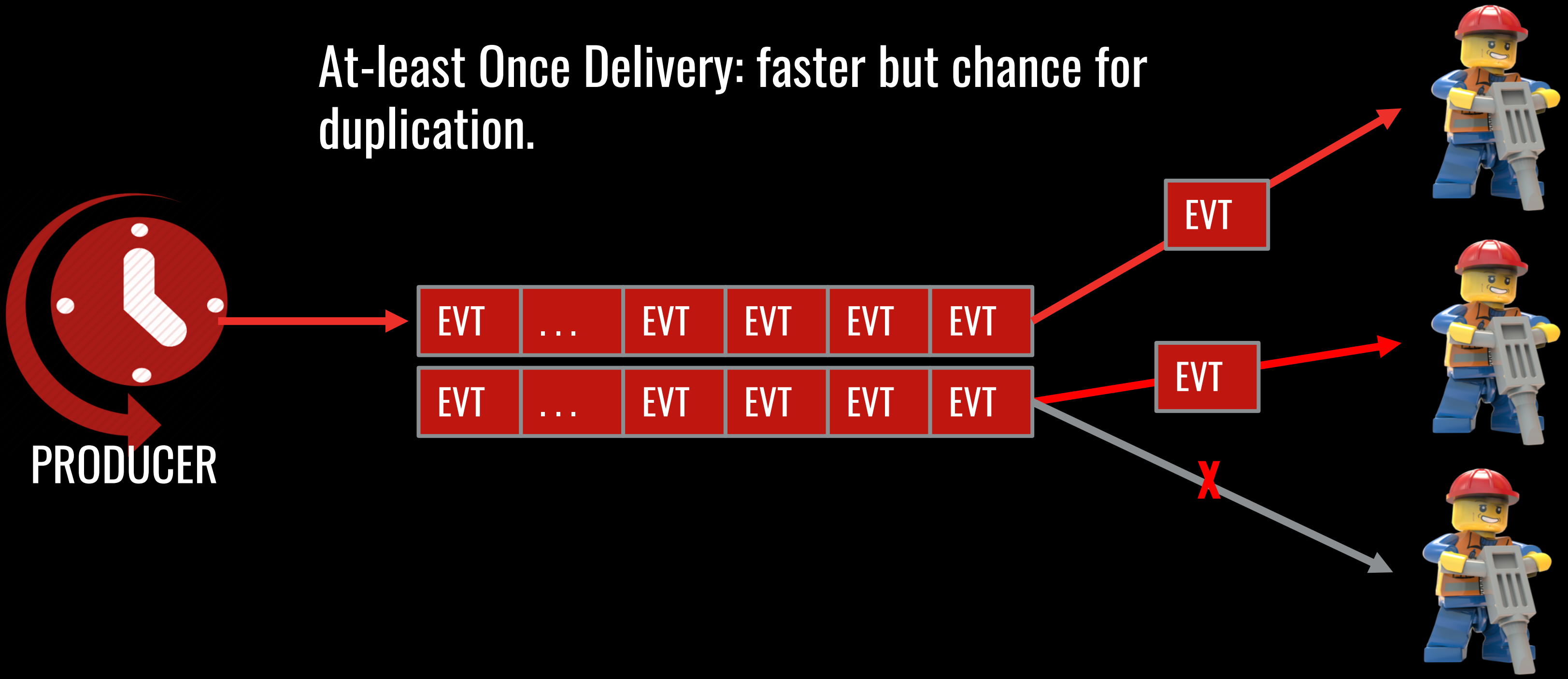
# Delivery Guarantees

Exactly Once Delivery: slower but safer



# Delivery Guarantees

At-least Once Delivery: faster but chance for duplication.



PRODUCER

EVT ... EVT EVT EVT EVT

EVT ... EVT EVT EVT EVT

EVT

EVT

X

WORKER
aka
CONSUMER

# Idempotency

Idempotent – A way to handle duplicate events. An idempotent operation will produce the same results if executed once or multiple times

EXAMPLE:

Non-Idempotent:
- `x = x * 5;`
- Submitting a purchase

Idempotent:
- `abs( abs(x) ) = abs(X)`
- Cancelling a purchase

# Idempotent?

In the transactional OLTP world....

```
INSERT INTO games_played
(SELECT * FROM games_played_na
 WHERE date >= ‘2015-10-25’)
```

Potentially with ACID – could get an id already exists

# Idempotent?

In the big data / OLAP world….

```
INSERT INTO games_played
(SELECT * FROM games_played_na
 WHERE date >= ‘2015-10-25’)
```

Probably not with noSQL – could get duplicates

# Idempotency

Add application logic to make idempotent

```
msg = queue.pop;
if (processed_games.contains( msg.game_id )
{
    return; //do nothing
else {
    process_game(msg);
}
```

# CASE STUDY: RIOT GAMES

# WHAT IS LEAGUE OF LEGENDS?

2009
LAUNCH

ONLINE
MULTIPLAYER

WINDOWS
/ OSX

30-40 MIN
GAMES

YOUR CHAMP

THE TEAM

THE BATTLE GROUND

**12 BILLION**
GAME RELATED EVENTS

**0.5 TRILLION**
DATA POINTS

**50 TB**
STORAGE

DAILY

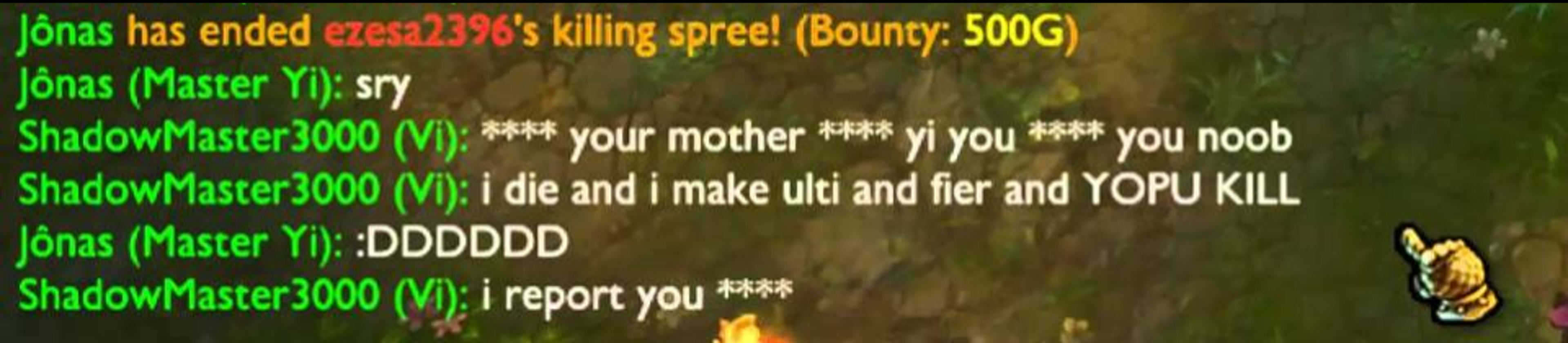
**26 PETABYTES**
PLAYER DATA

SINCE BETA

# OFFENSIVE CHAT DETECTION

Data science team queries all chat messages in game

Sentiment analysis and classification



Jônas has ended ezesa2396's killing spree! (Bounty: 500G)

Jônas (Master Yi): sry

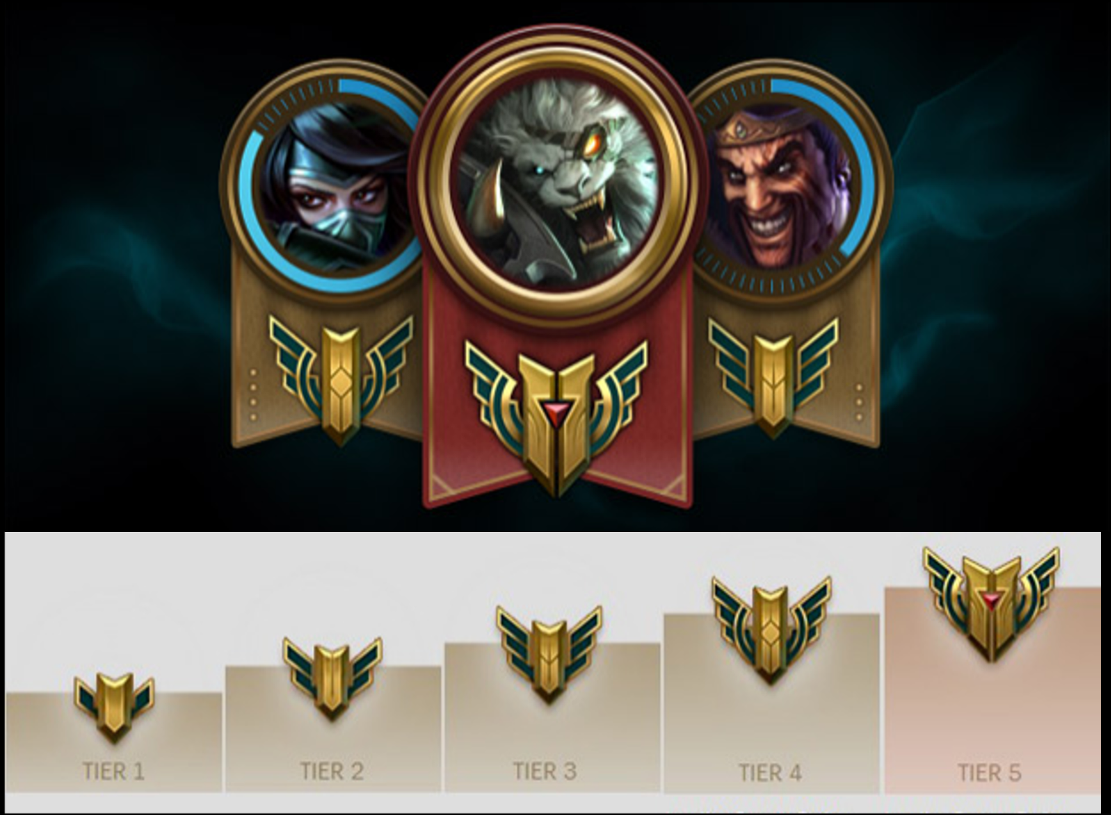
ShadowMaster3000 (Vi): **** your mother **** yi you **** you noob

ShadowMaster3000 (Vi): i die and i make ulti and fier and YOPU KILL

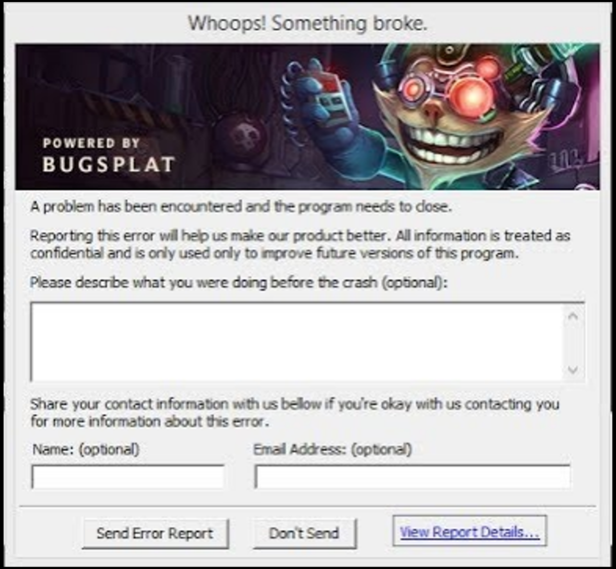
Jônas (Master Yi): :DDDDDD

ShadowMaster3000 (Vi): i report you ****

# CHAMPION MASTERY



# PLAYER SUPPORT



Whoops! Something broke.

A problem has been encountered and the program needs to close.

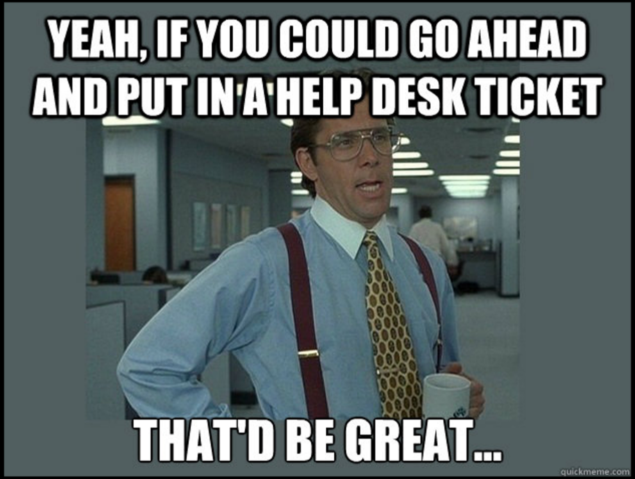
Reporting this error will help us make our product better. All information is treated as confidential and is only used only to improve future versions of this program.

Please describe what you were doing before the crash (optional):

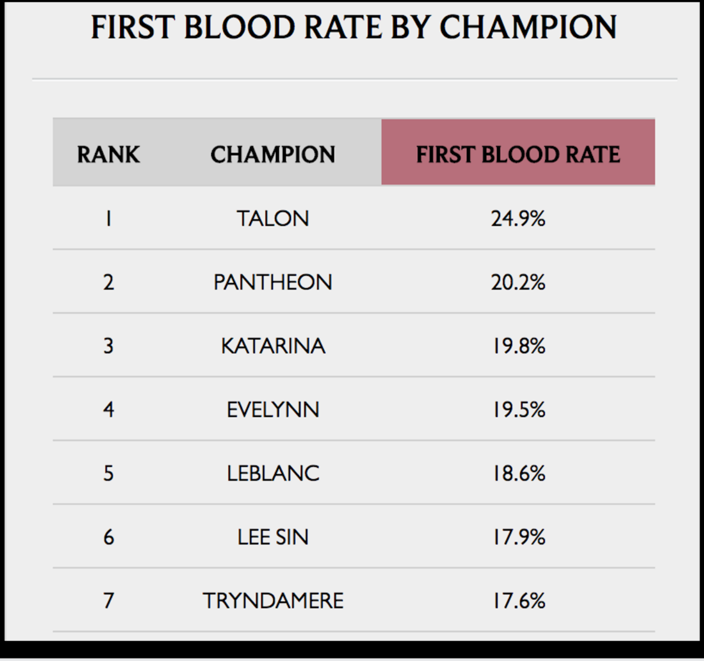
Share your contact information with us bellow if you're okay with us contacting you for more information about this error.

Name: (optional)

Email Address: (optional)

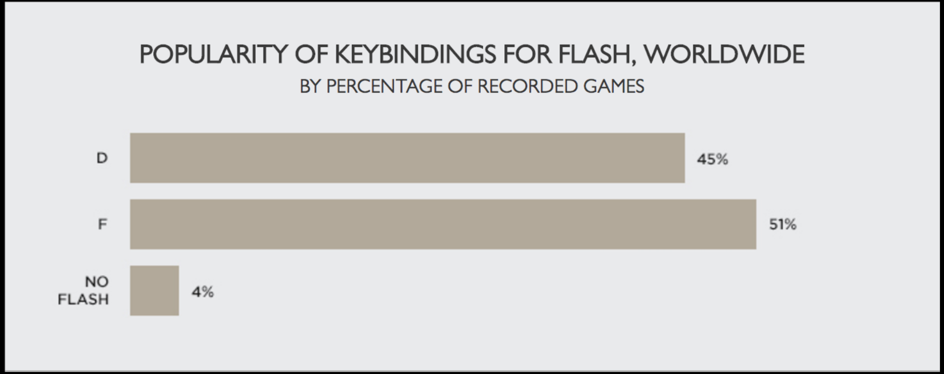


# GAME BALANCE



## FIRST BLOOD RATE BY CHAMPION

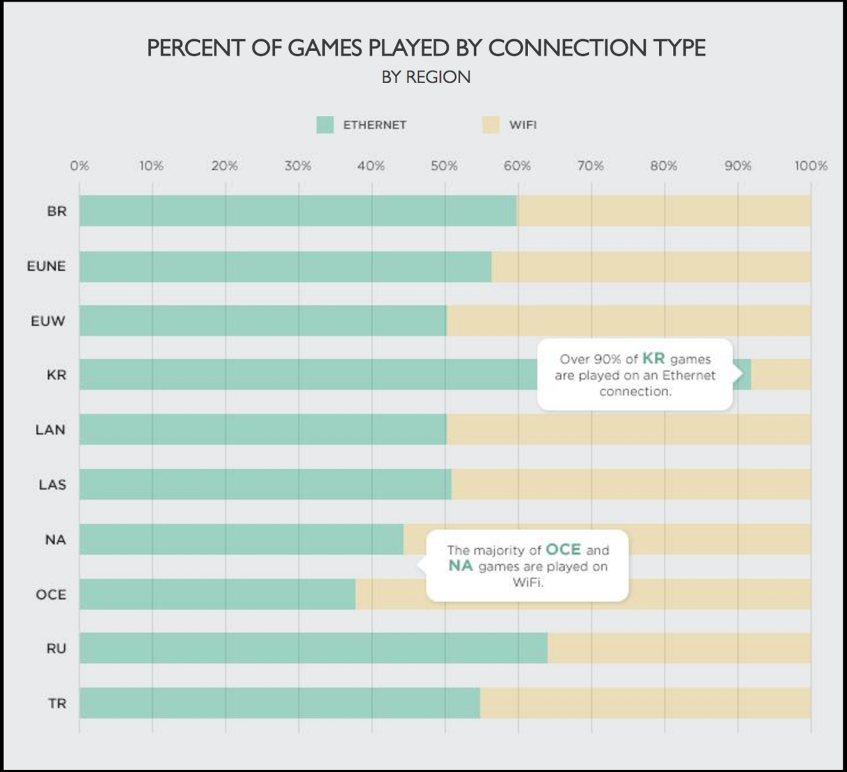
| RANK | CHAMPION | FIRST BLOOD RATE |
|---|---|---|
| 1 | TALON | 24.9% |
| 2 | PANTHEON | 20.2% |
| 3 | KATARINA | 19.8% |
| 4 | EVELYNN | 19.5% |
| 5 | LEBLANC | 18.6% |
| 6 | LEE SIN | 17.9% |
| 7 | TRYNDAMERE | 17.6% |



POPULARITY OF KEYBINDINGS FOR FLASH, WORLDWIDE

BY PERCENTAGE OF RECORDED GAMES

# LATENCY AND NETWORK

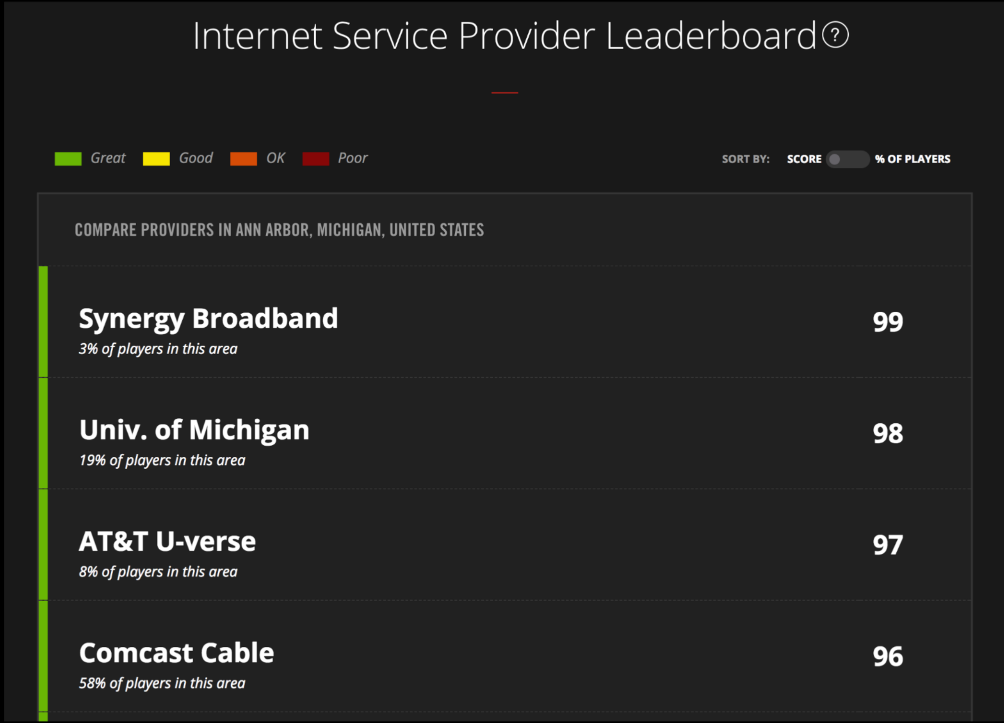


PERCENT OF GAMES PLAYED BY CONNECTION TYPE

BY REGION

Over 90% of KR games are played on an Ethernet connection.

The majority of OCE and NA games are played on WiFi.



Internet Service Provider Leaderboard ?

COMPARE PROVIDERS IN ANN ARBOR, MICHIGAN, UNITED STATES

**Synergy Broadband** 99

*3% of players in this area*

**Univ. of Michigan** 98

*19% of players in this area*

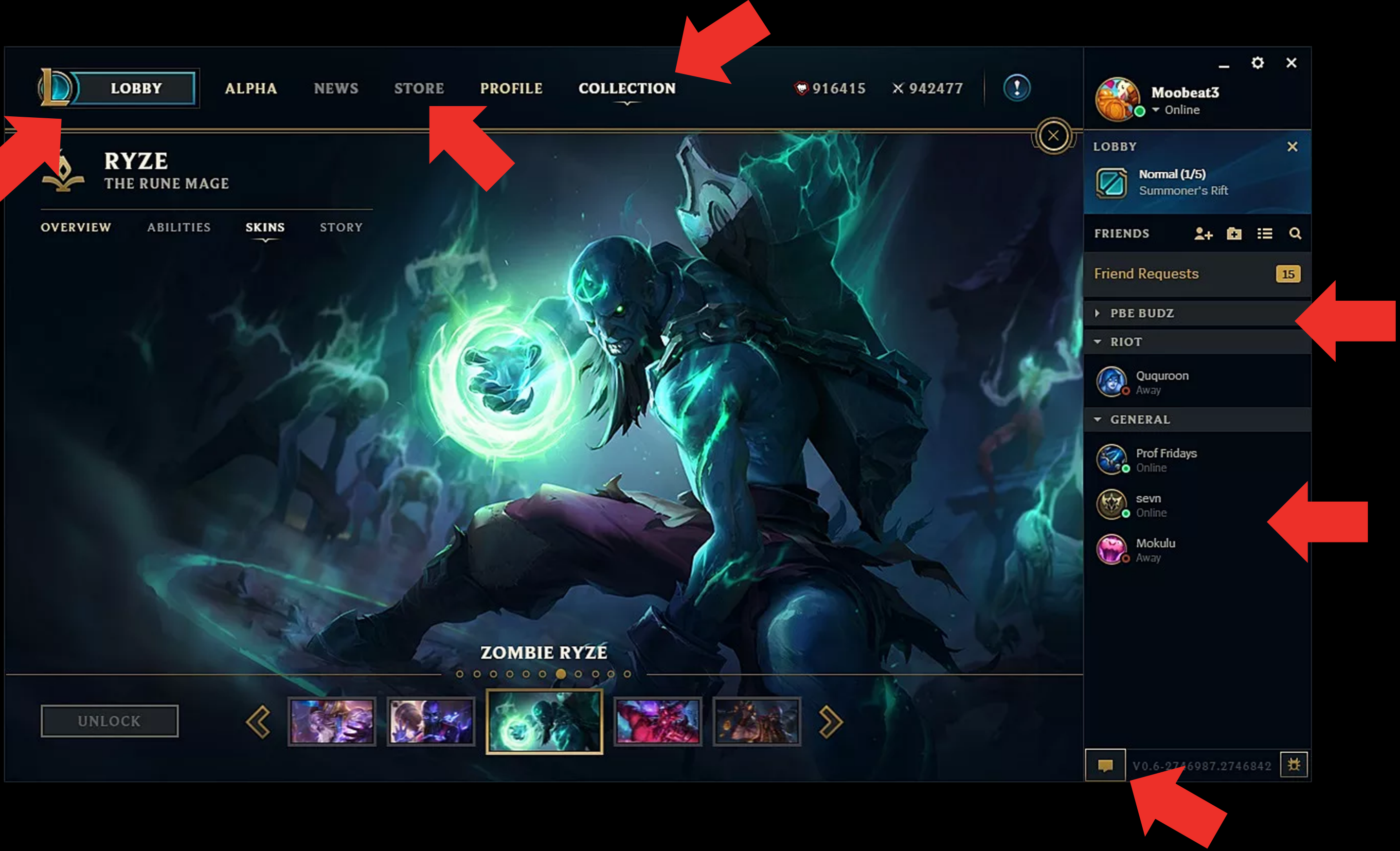
**AT&T U-verse** 97

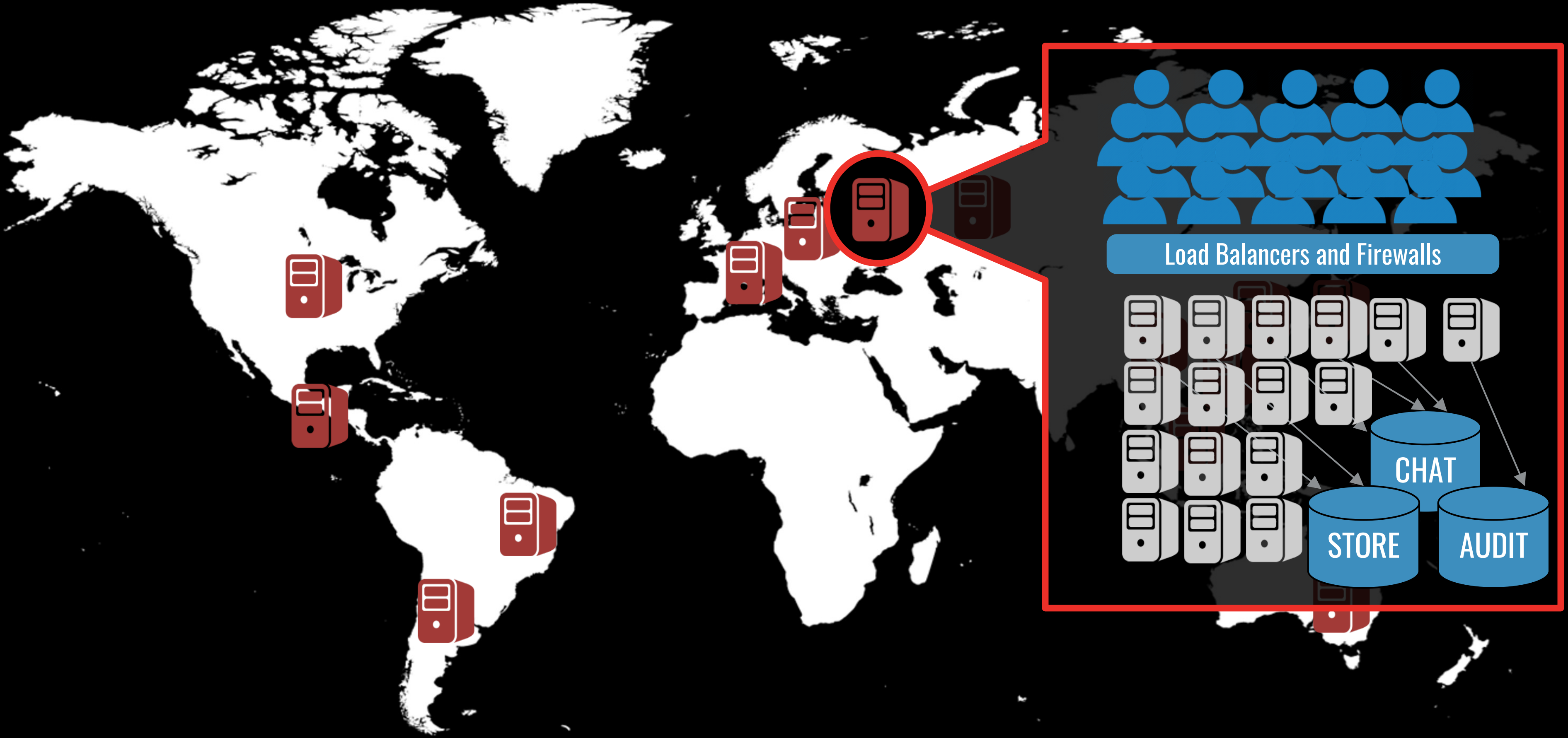
*8% of players in this area*

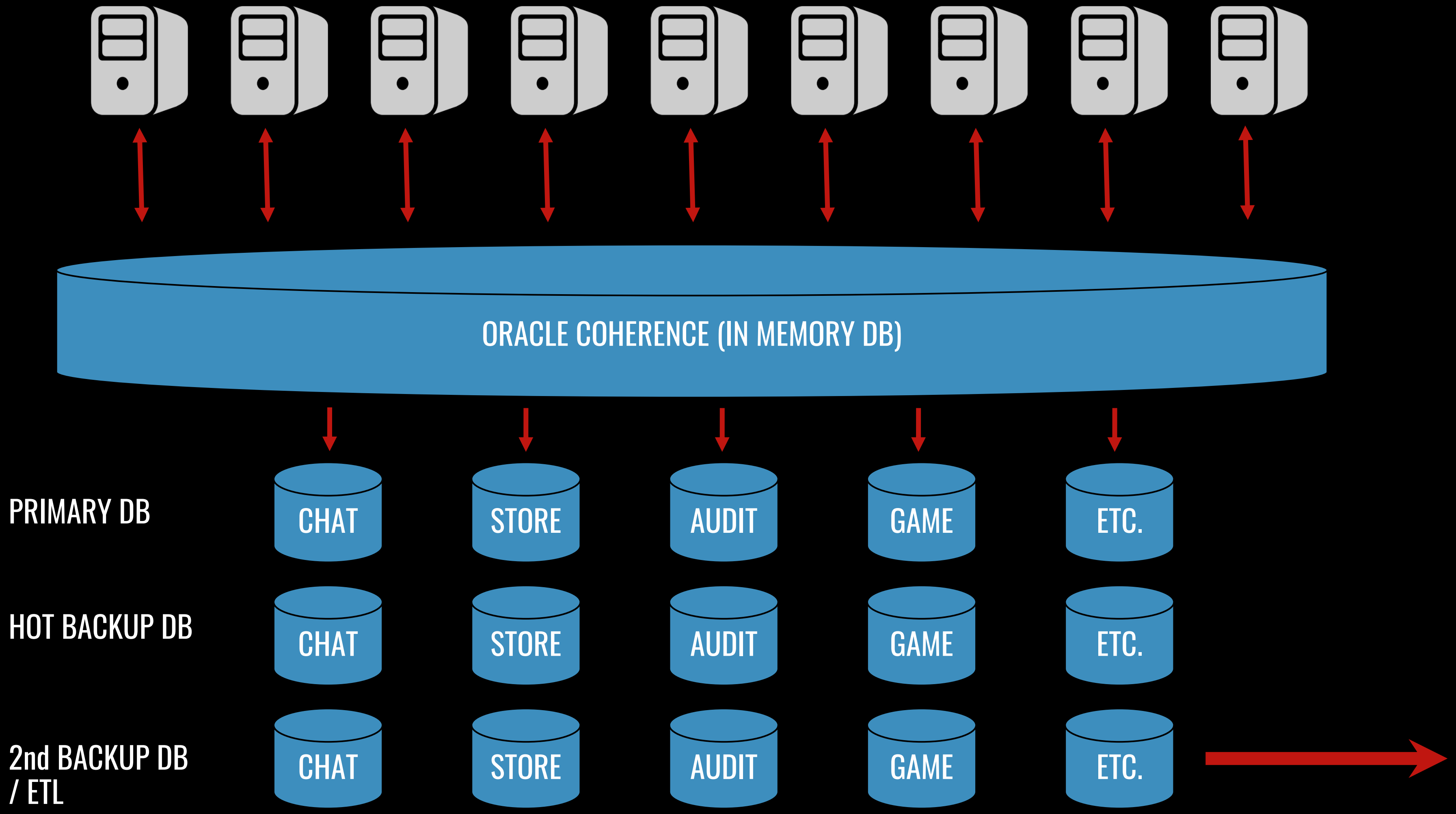
**Comcast Cable** 96

*58% of players in this area*

# MICROSERVICES




Load Balancers and Firewalls
CHAT
STORE
AUDIT

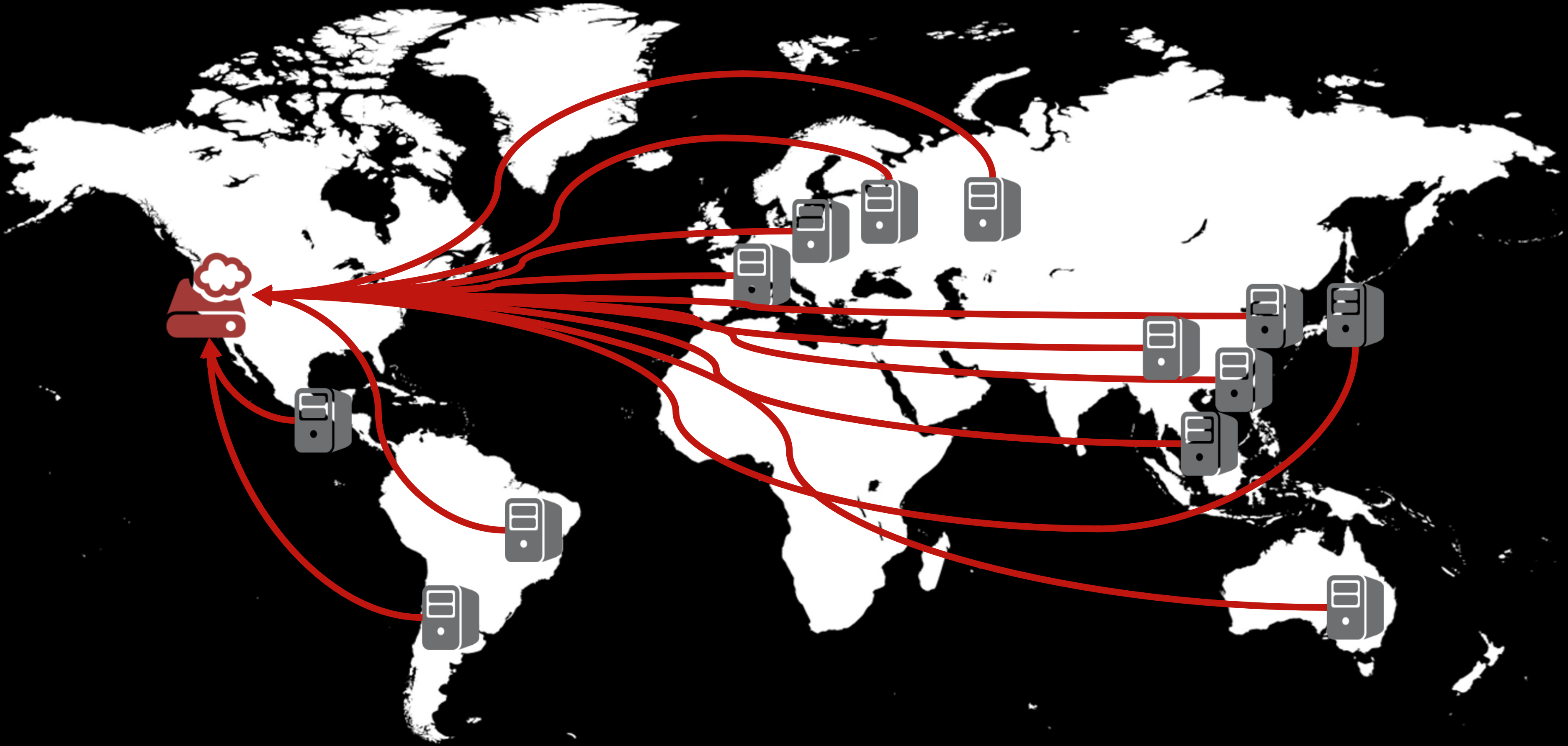


ORACLE COHERENCE (IN MEMORY DB)

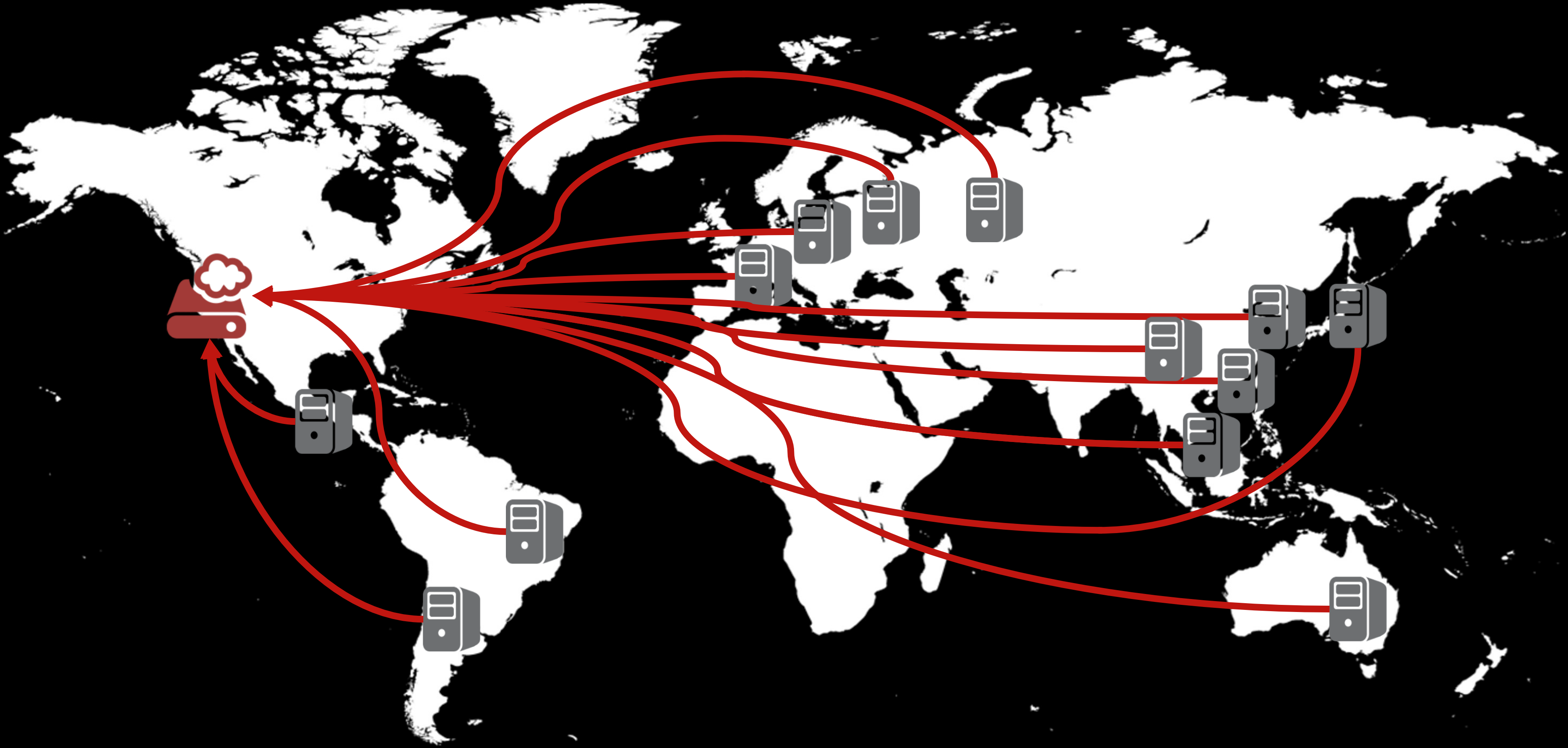
PRIMARY DB
CHAT STORE AUDIT GAME ETC.

HOT BACKUP DB
CHAT STORE AUDIT GAME ETC.

2nd BACKUP DB / ETL
CHAT STORE AUDIT GAME ETC.



# EVENTS AT RIOT



## Kafka

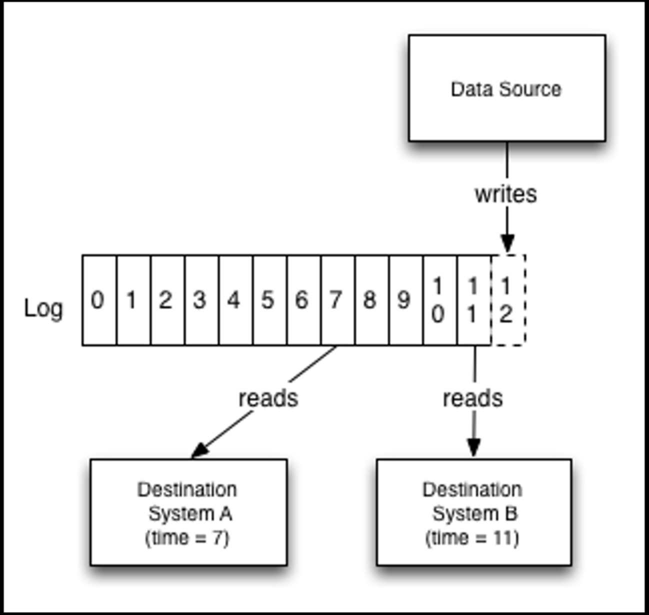
Open-source project maintained by Confluent

Very fast distributed event bus

Data is replicated across “partitions” to ensure no loss

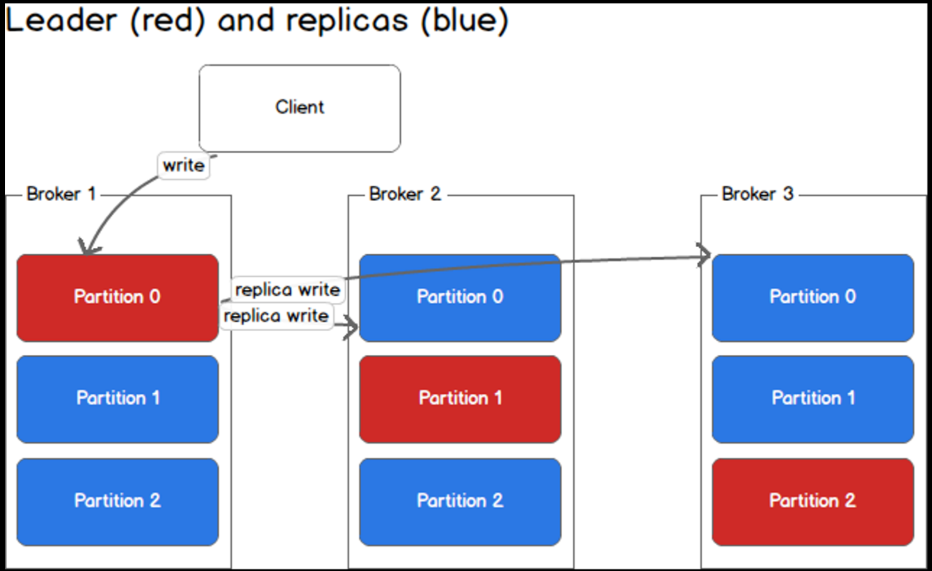
Kafka

# Has a DB Commit Log
(ooh revolutionary - can replay events)


Data Source
writes
Log
0
1
2
3
4
5
6
7
8
9
1 0
1 1
1 2
reads
reads
Destination System A (time = 7)
Destination System B (time = 11)

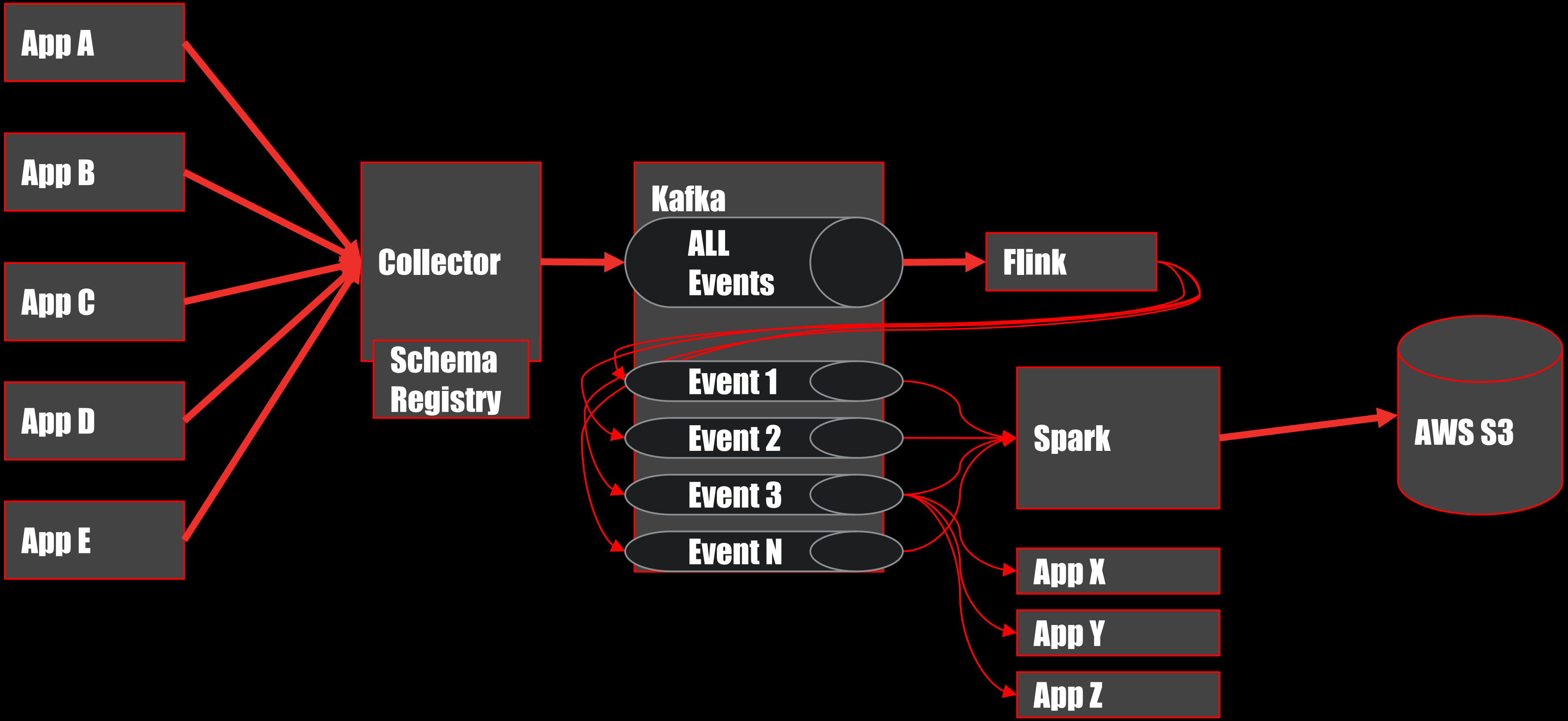
Kafka

# Replicates data and scales for outages



Leader (red) and replicas (blue)

# EVENTS AT RIOT



# Outages

- Network failure between producer and data collection api?

  Producer client a.) exponential backoff retries

  b.) in-memory queue to buffer

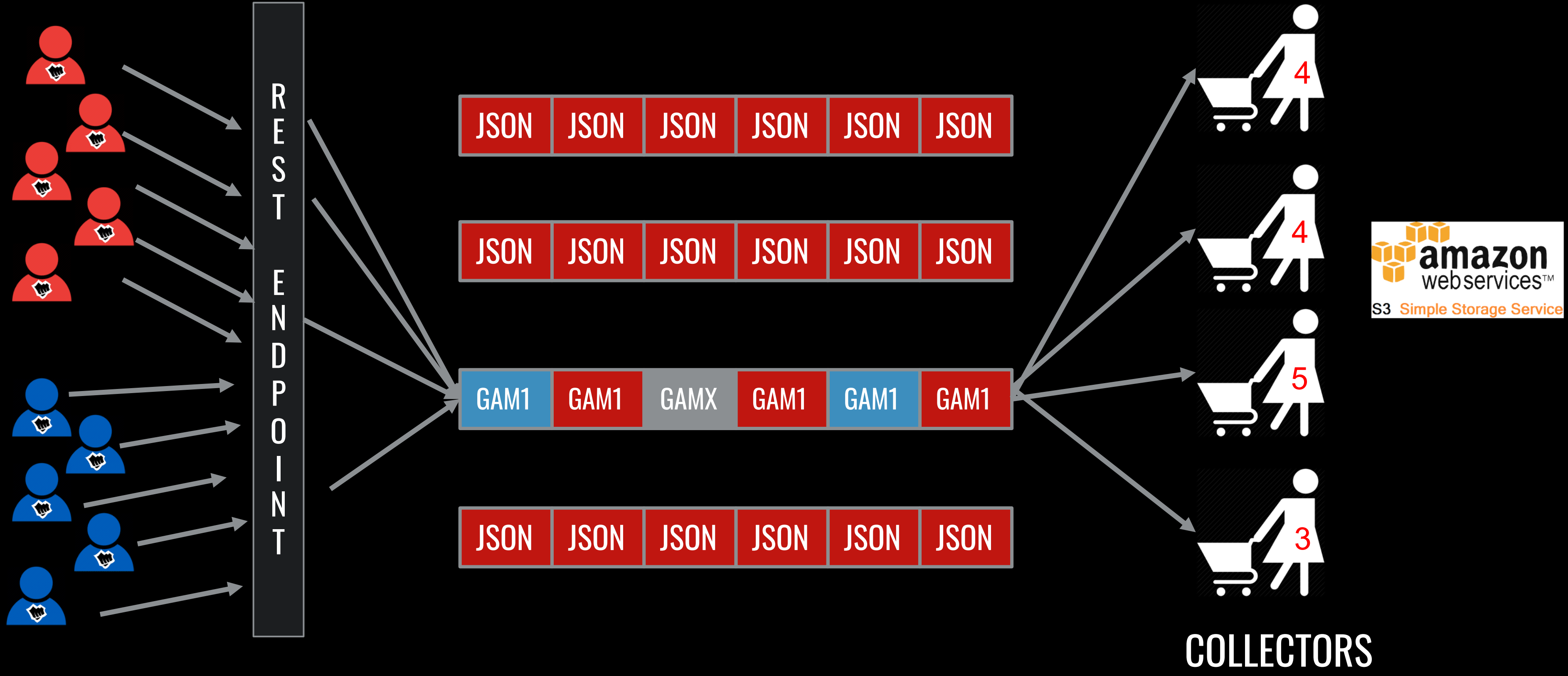
- Entire Kafka cluster dies?

  API returns 503 informing client the data wasn't persisted and to try again / buffer

- S3 fails, or other consumer?

  Data is buffered for 7 days in Kafka until they recover

# Idempotent?


REST ENDPOINT
JSON JSON JSON JSON JSON JSON
JSON JSON JSON JSON JSON JSON
GAM1 GAM1 GAMX GAM1 GAM1 GAM1
JSON JSON JSON JSON JSON JSON
4
4
5
3
amazon webservices™
S3 Simple Storage Service
COLLECTORS

# QUESTIONS?

SEAN MALONEY

✉ SEAMALO @ amazon.com

@SEAN_SEANNERY