Building a Highly Scalable Service that Survived a Super Bowl

Keith Elder

Sr. Technology Evangelist

@keithelder

http://keithelder.net

Quicken Loans

Today's Talk

Some stories
Some demos
Some technologies we used and why
Some lessons learned

February 7, 2016

Quicken Loans

FutureLabs

Innovation

Blog Architecture

Web Request

Web Server

Fetch Data

Database

Popular Blog Architecture

Enterprise Blog Architecture

Web Request

Load Balancer

Web Farm

Fetch Data

Database Cluster

Cache Cluster

Queued Services Architecture

Get angle brackets (html)

Web Farm

Get curly braces (data)

Fetch Data

REST API

Database

Queue

Pub/Sub

Cache

Backend Service

Messaging

Queued Services Architecture

Get angle brackets (html)

Web Farm

Get curly braces (data)

Fetch Data

REST API

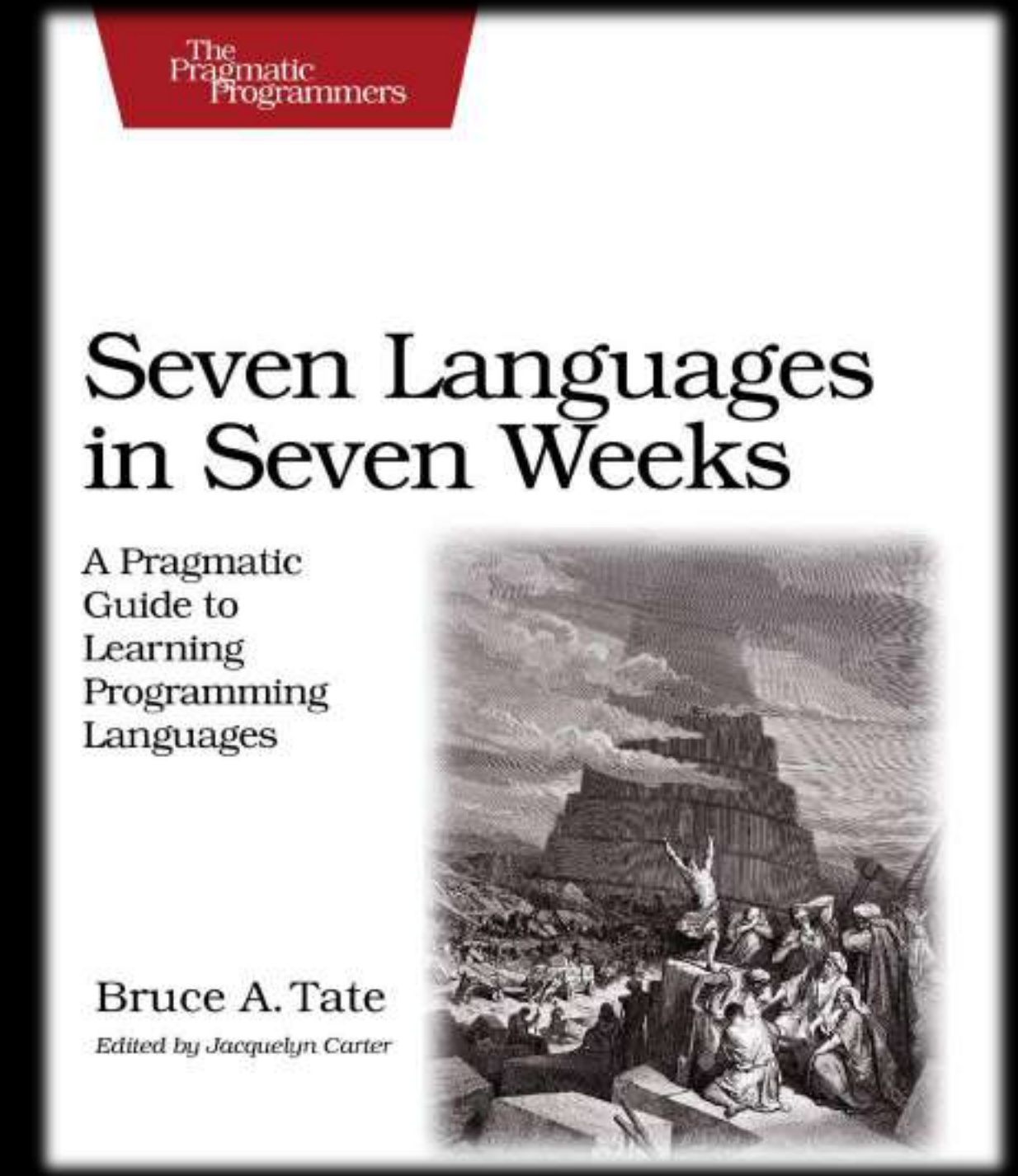
Database

Queue

Pub/Sub

Cache

Backend Service



Ruby

Scala

lo


ERLANG

Erlang In A Tweet

Erlang: a battle-hardened, X-platform, functional language that makes writing **reliable, concurrent, distributed** systems a joy.

Where Is Erlang Used?

Automotive, Gaming, Finance, Robots, Cars, Drone software (updates drone while flying), Health Care, Infrastructure, Social Networks, Media (Huffington Post, Boston Globe), Messaging


CALL OF DUTY
BLACK OPS

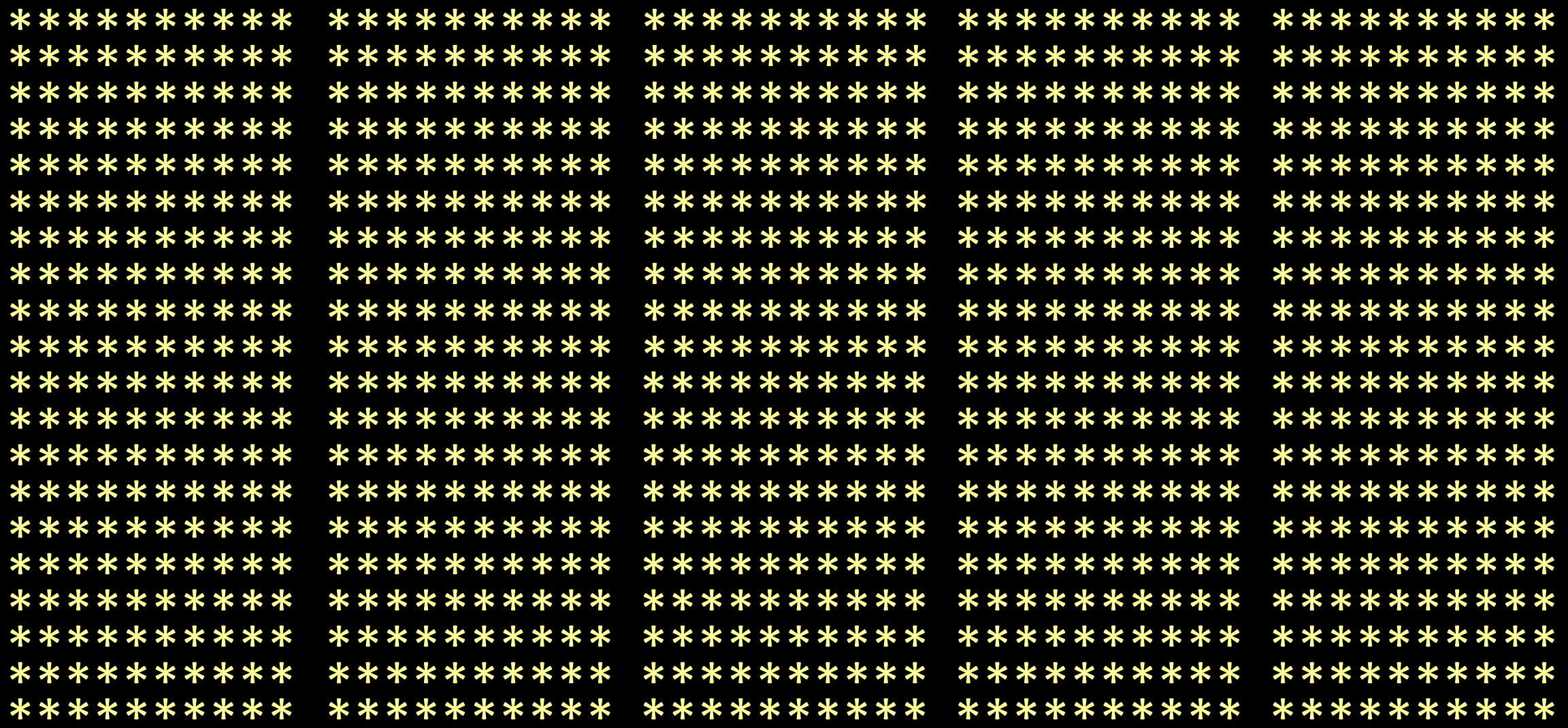

GUITAR HERO III
LEGENDS OF ROCK


DESTINY


SKYLANDERS
SUPERCHARGERS

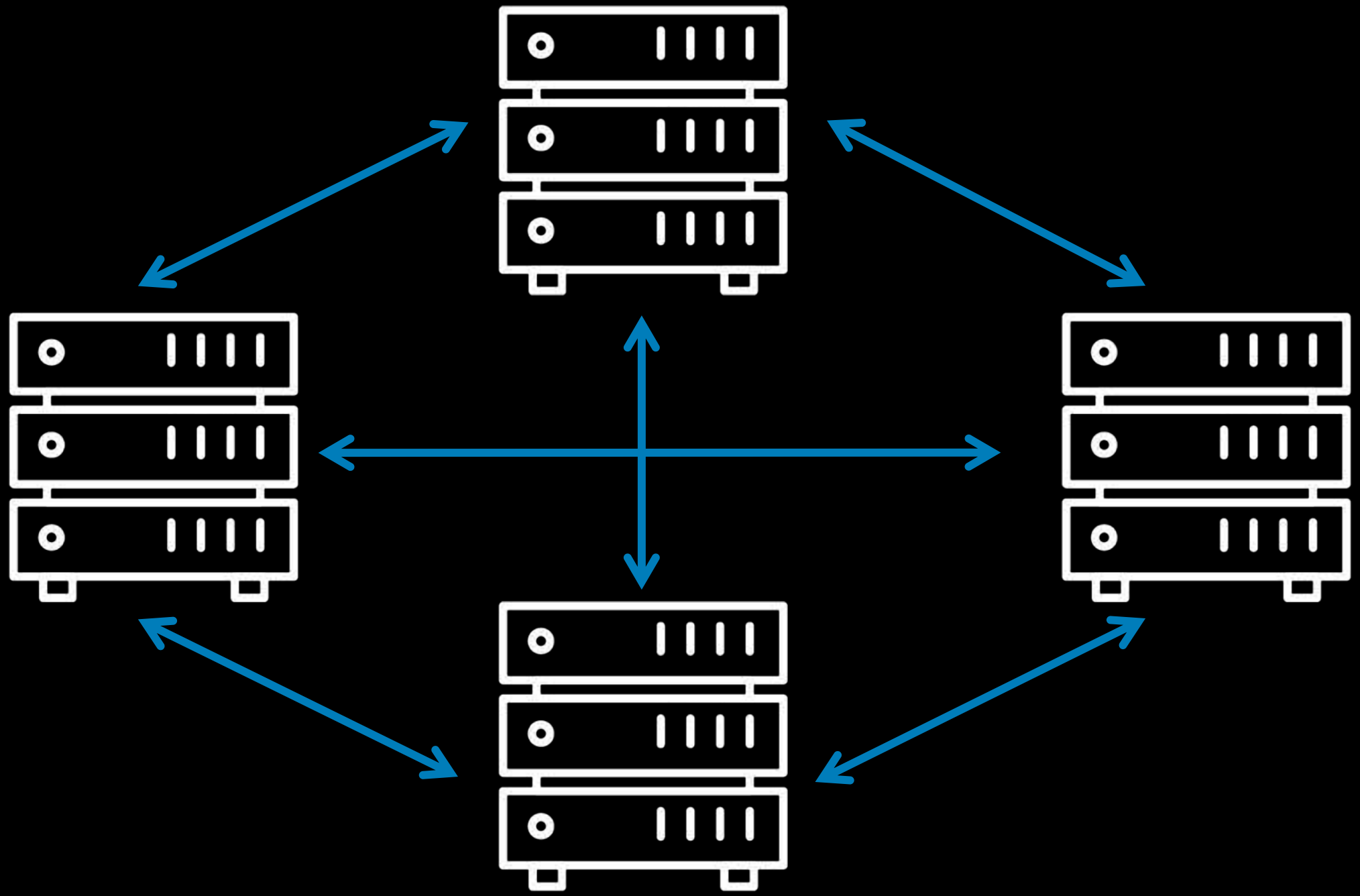

DIABLO III


CALL OF DUTY
ADVANCED WARFARE



.NET 4.0 Thread (1MB)

* One Erlang Process
(allocates one kilobyte)



A few demos

https://github.com/keithelder/presentations

Erlang Problem Domains

- Can’t fail
- Distributed
- Fault tolerance
- Upgrade while the app is running
- Performed without stopping the system
- Large number of concurrent activities

"My team has created a very innovative solution, but we're still looking for a problem to go with it."

Super Bowl is **302** days away

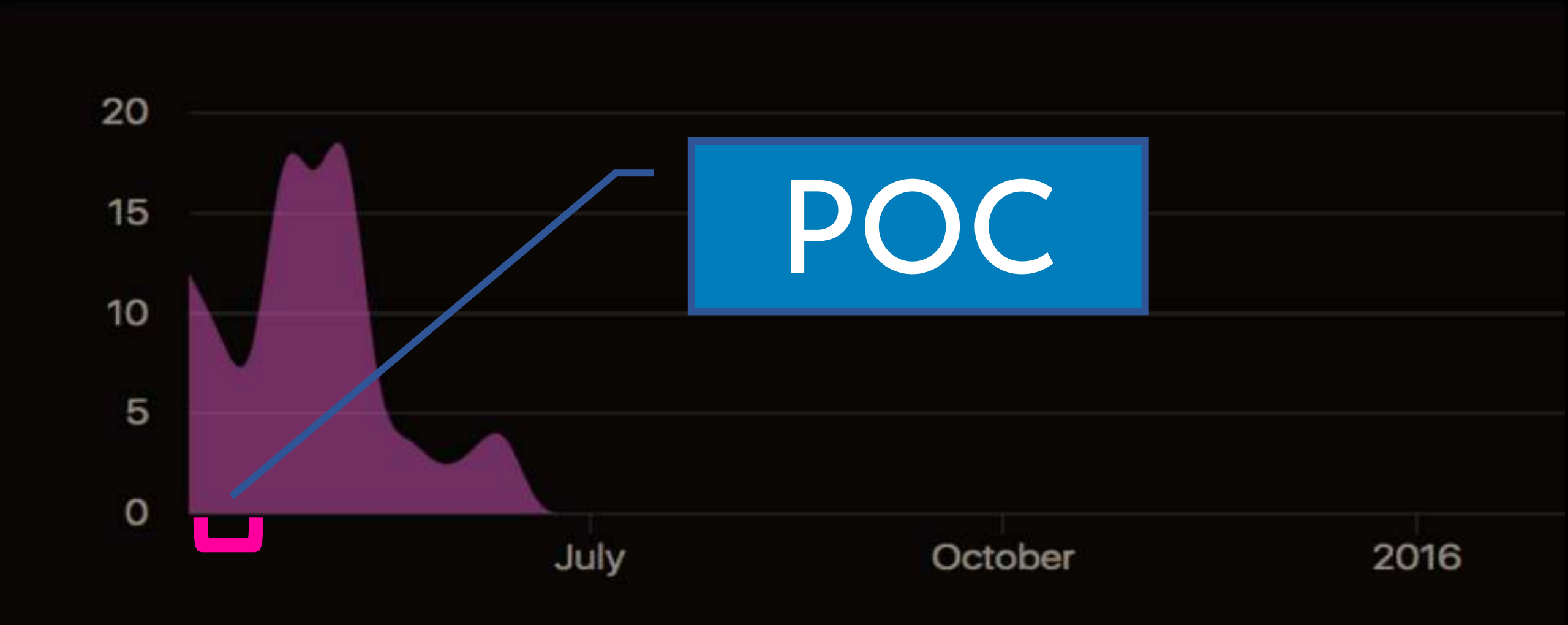
90+ days

to figure out a solution

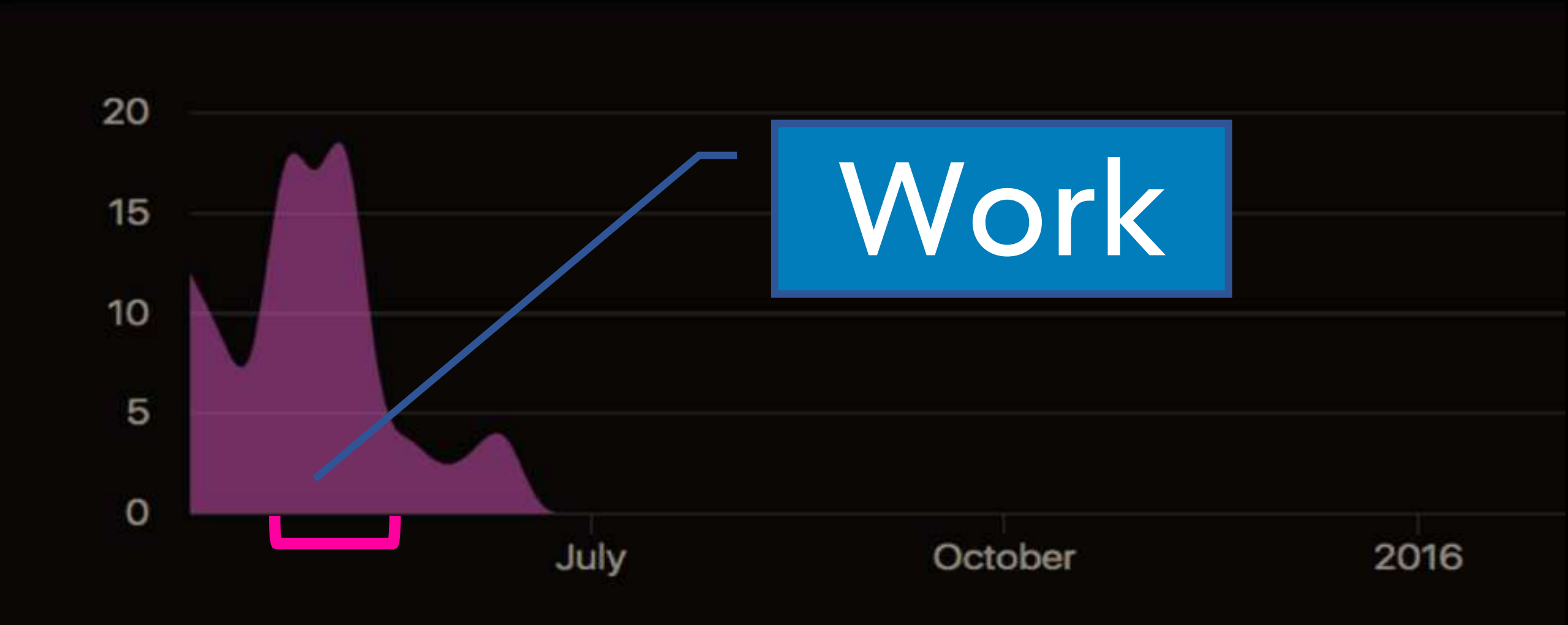
Requirements

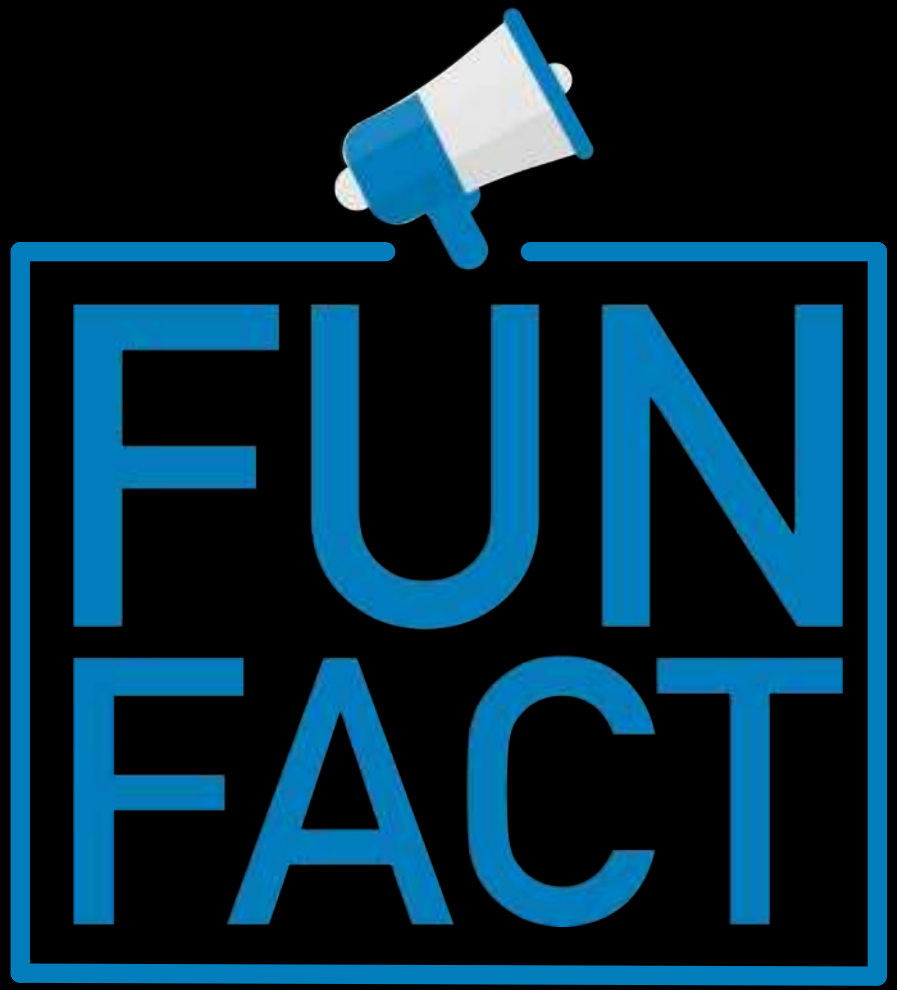
- Hard deadline
- 1-2 millisecond response time
- 30,000 transactions / second
- Always on
- Vast array of languages (C#, Java, Python, PHP)
- Hybrid cloud app
- Encryption during transit and at rest
- Easy for developers

April 12, 2015 – June 21, 2015



April 12, 2015 – June 21, 2015

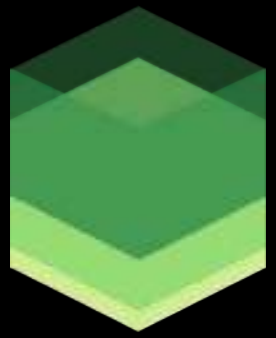




- 2,000 lines of code
- Two Senior Engineers
- The most used service in Rocket Mortgage

The Stack

riak ensemble



LEVELDB

Cowboy

libsodium

exometer

Moment of Truth

Load test dev

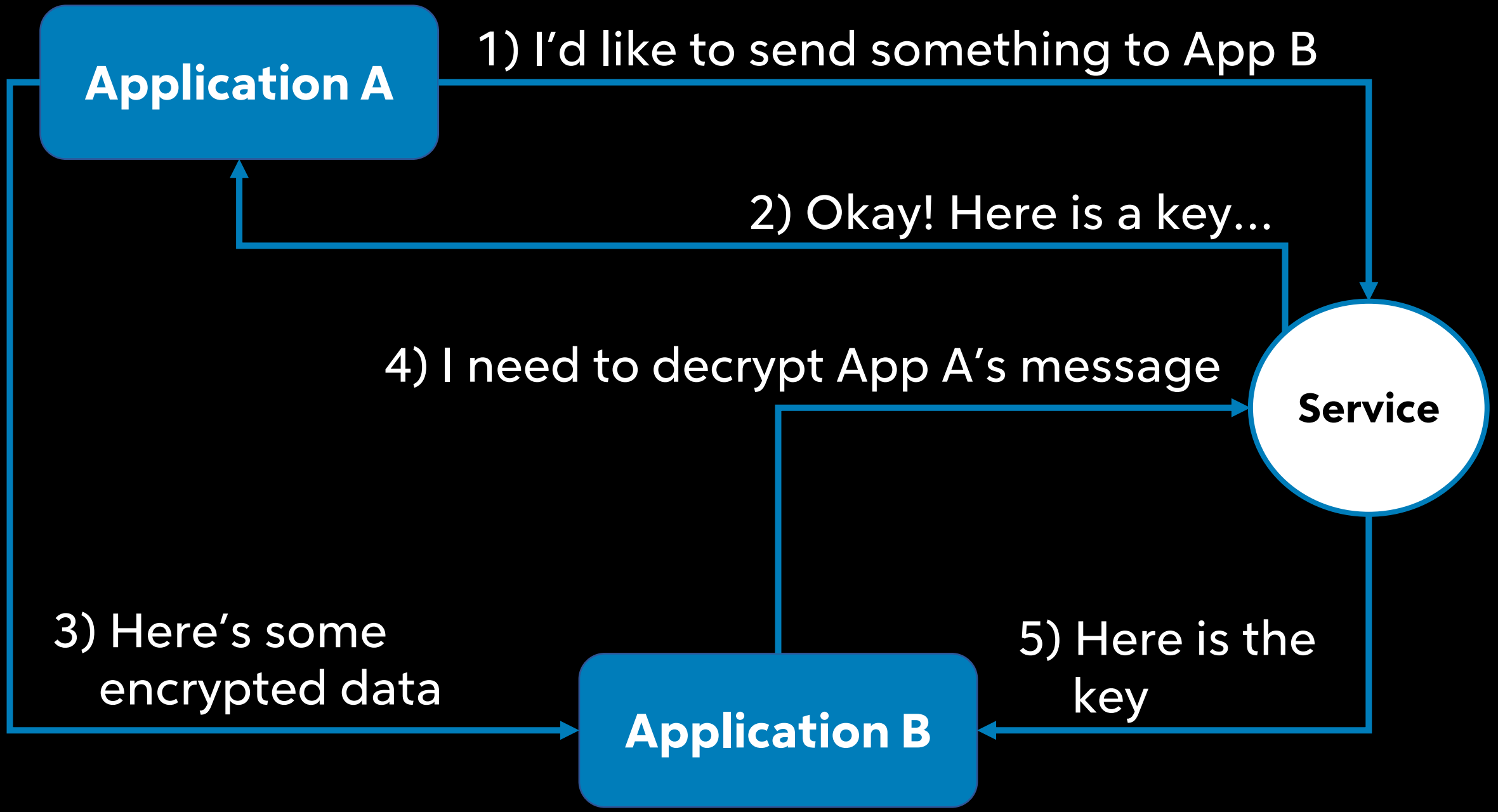
12,881

transactions per second

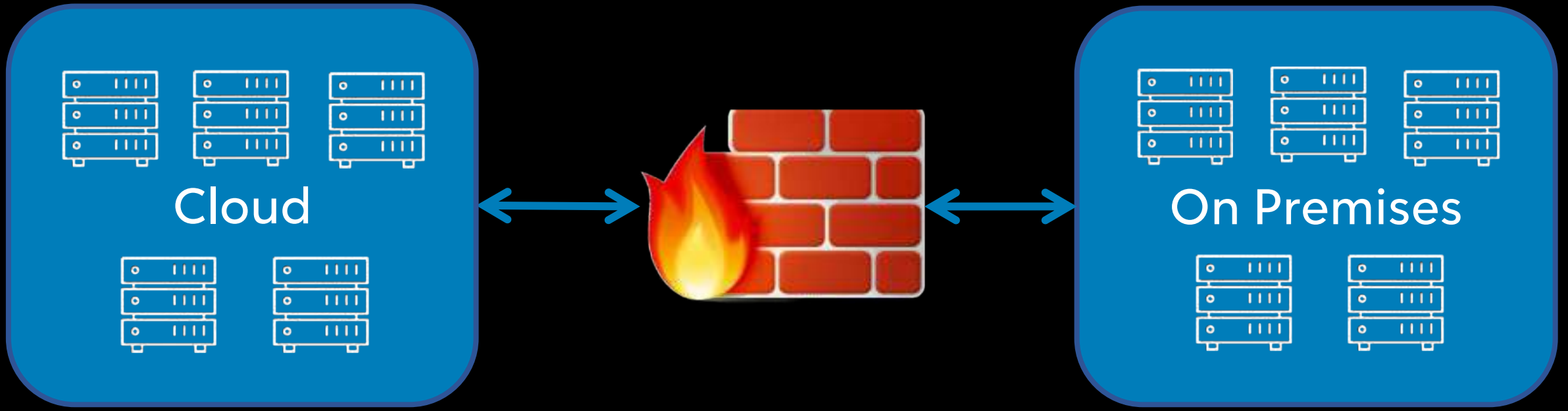
20,291

transactions per second

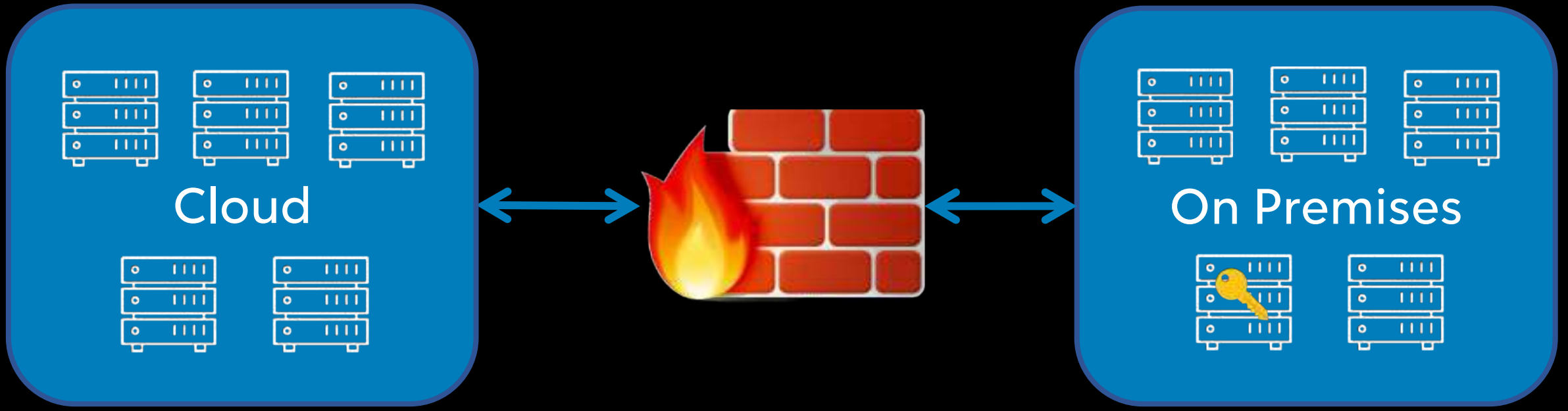
How It Works


Application A
1) I'd like to send something to App B
2) Okay! Here is a key...
Service
4) I need to decrypt App A's message
3) Here's some encrypted data
5) Here is the key
Application B

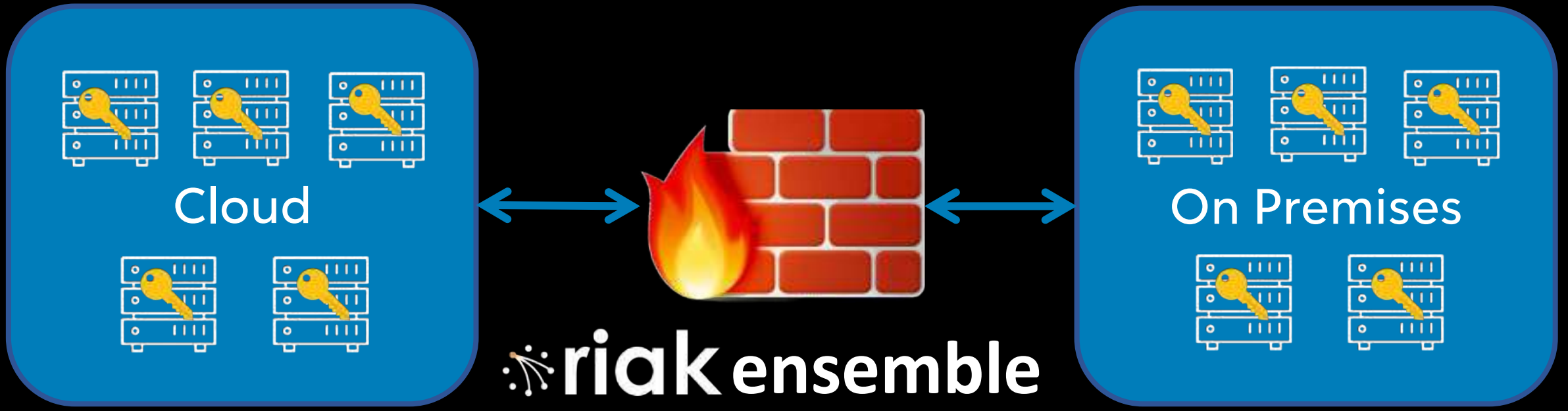
Hybrid Cloud – each node is self-contained



Hybrid Cloud – data is stored locally



Hybrid Cloud – kept in sync across all nodes



How Did We Do?

Requirements

- ☑ Hard deadline
- ☑ 1-2 millisecond response time
- ☑ 30,000 transactions / second
- ☑ Always on
- ☑ Vast array of languages (C#, Java, Python, PHP)
- ☑ Hybrid cloud app
- ☑ Encryption during transit and at rest
- ☐ Easy for developers

https://github.com/RockLib

http://rockframework.org

```
static void Main(string[] args)
{
    string json = SerializingCrypto.ToJson(new Foo { Bar = 123, Baz = 456 });
    Console.WriteLine(json);
    Foo fooFromJson = SerializingCrypto.FromJson<Foo>(json);
    Console.WriteLine($"fooFromJson.Bar:{fooFromJson.Bar}, fooFromJson.Baz:{fooFromJson.Baz}");

    Console.WriteLine();

    string xml = SerializingCrypto.ToXml(new Foo { Bar = 123, Baz = 456 });
    Console.WriteLine(xml);
    Foo fooFromXml = SerializingCrypto.FromXml<Foo>(xml);
    Console.WriteLine($"fooFromXml.Bar:{fooFromXml.Bar}, fooFromXml.Baz:{fooFromXml.Baz}");
}

public class Foo
{
    public int Bar { get; set; }

    [Encrypt]
    public int Baz { get; set; }
}
```

February 7, 2016

Quicken Loans


I'M SO EXCITED
I'M SO SCARED
memegenerator.net

What We Learned

Erlang (BEAM) is a

viable platform

for performant API'S

The only way to performance

test Erlang is with Erlang

In a distributed system keep each node as

independent

as possible (local data, etc.)

Load test to pinpoint

bottlenecks

with each release

Would We Do It Again?

FutureLabs

Innovation

Building a Highly Scalable Service that Survived a Super Bowl

Keith Elder

Sr. Technology Evangelist

@keithelder

http://keithelder.net

Quicken Loans