Cupcakes*, Kafka, and .NET Core

*Cupcakes Not Included

Introductions

We are:

- Jacob Zweifel
- Srini Gummadidala

Jacob Zweifel
jzweifel

@TribalScale
Boston, MA

Srini
srigumm

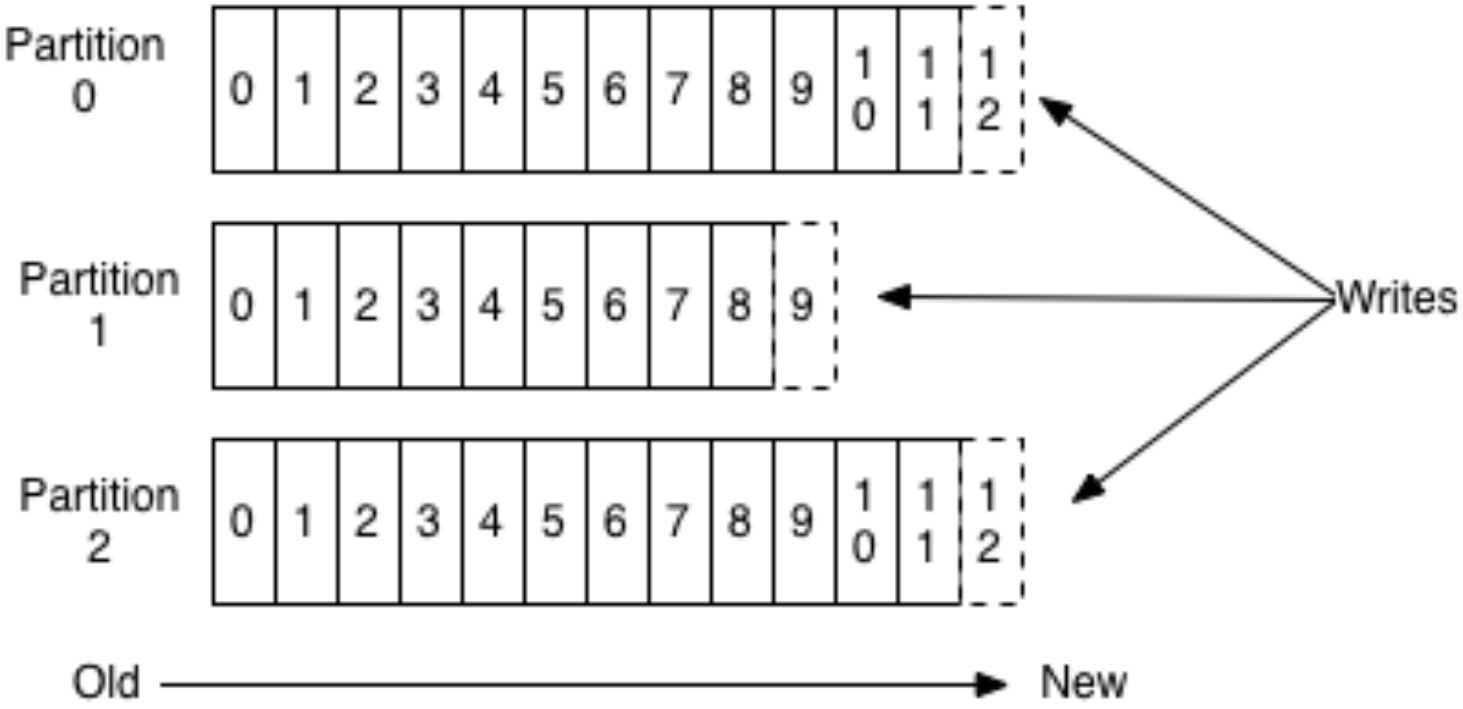
Tribalscale
Boston
linkedin.com/in/srigumm

What is Kafka?

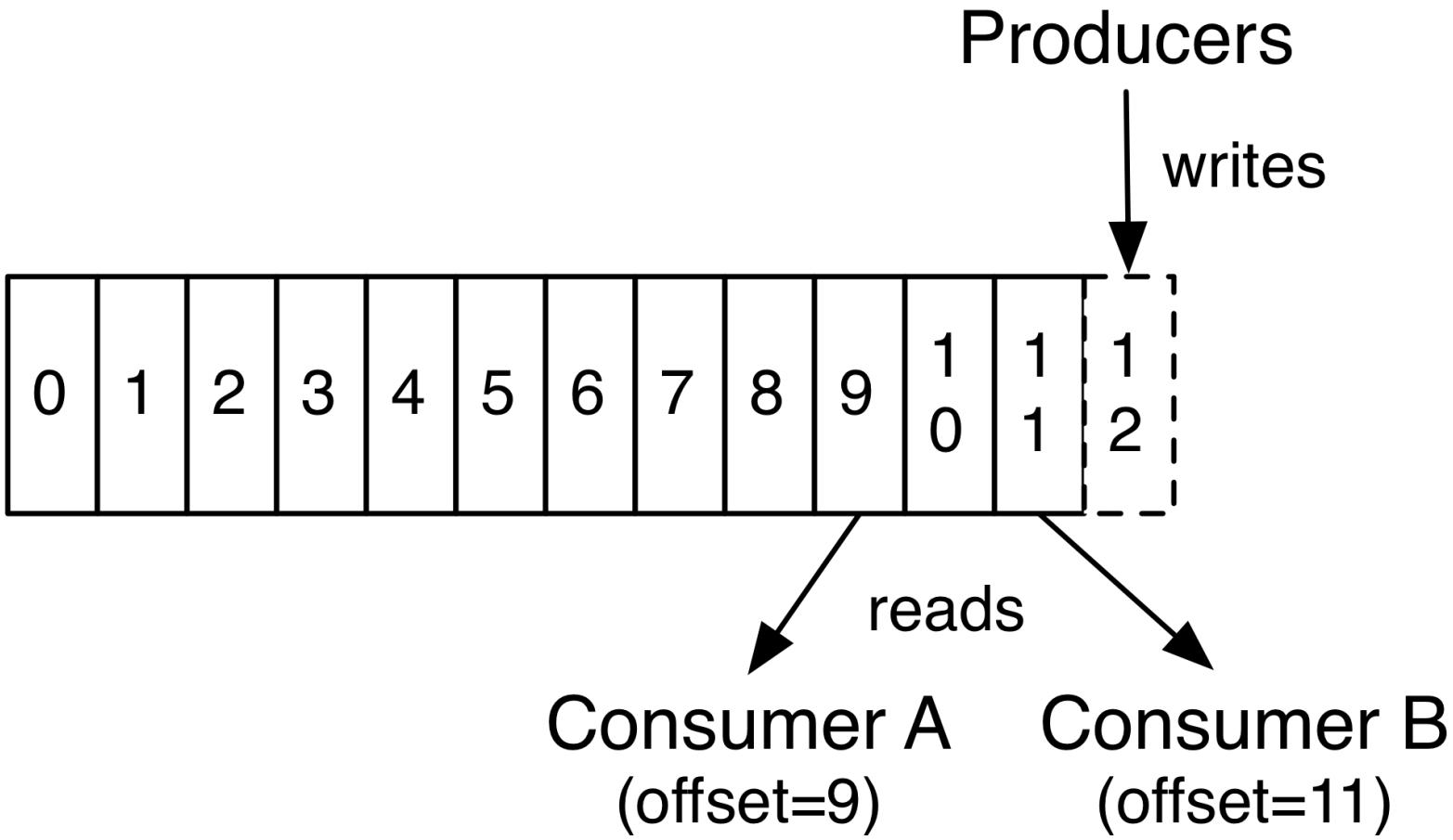
Kafka is a distributed, horizontally-scalable, fault-tolerant, commit log.

Topics - the core abstraction

Anatomy of a Topic



Producers and Consumers


Producers
writes
0
1
2
3
4
5
6
7
8
9
10
11
12
reads
Consumer A
(offset=9)
Consumer B
(offset=11)

Kafka as a Messaging System

Kafka as a Storage System

Kafka as the stream processing, cupcake baking, traditional ETL killer!

Crusty Cupcake Factory (ETL)

- Purpose-built all-in-one kitchen (SSIS)
- Batch processing, likely scheduled
- Rigid, sharded, painful to scale

Castle Cupcake Factory (Stream Processing)

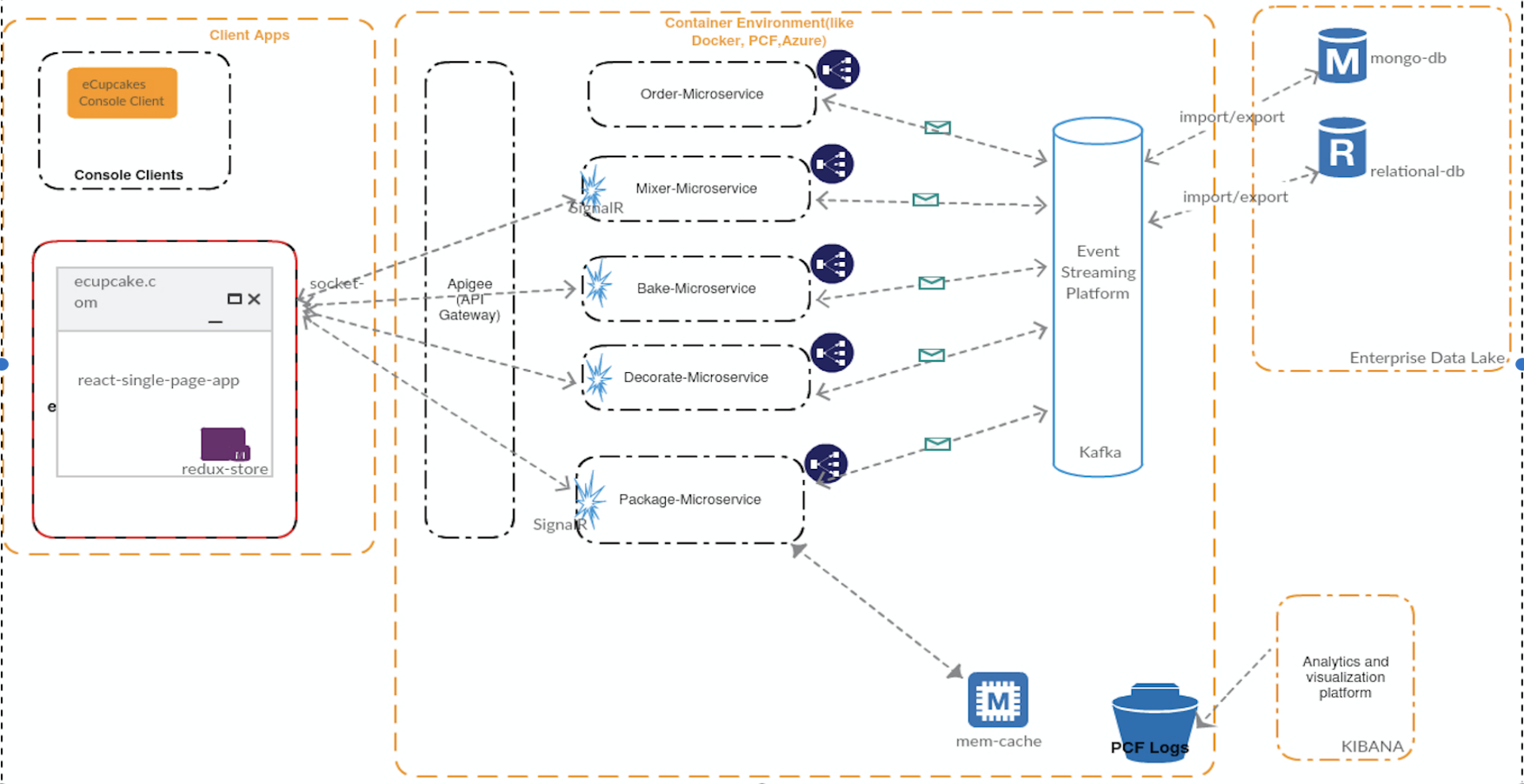
- Individual, task-based specialists (microservices)
- Event sourced processing
- Micro-scaling

What We’ve Built

eCupcakesFactoryOnContainers

- .NET Core v2.2
  - Web API
  - SignalR
  - IHostedService
- Apache Kafka
  - Consumer/Producer API
  - Confluent Cloud (hosted Kafka brokers)
- React
- Docker
- Kubernetes
- Google Cloud Platform

eCupcakeFactory reference application



Demo Time!

Demo Details

http://bit.do/cupcakery

The Cupcakery: End-to-End

1. Customer places an order for a cupcake using our web client (React).
2. Mixer in the kitchen gets notified about the new order, mixer prepares the mix and updates the order status as “mixed”.
3. A baker gets notified about this. The baker pours batter, bakes in oven and updates the order status as “baked”.
4. Now a decorator is notified about the baked order. The decorator decorates them creatively and updates the order status as “decorated”.
5. Finally, packaging team gets notified about the baked order, they pack the order in the right size box and updates the order status as “packaged”.

End-to-End: Drilldown

- React client application, on startup:
  - Connects to SignalR hub to establish WebSocket connections
  - Each WebSocket connection represents a Kafka Consumer
  - Each SignalR hub represents a Consumer Group
- On action (order, mix, bake, package):
  - Performs POST to Web API service
  - Web API service uses Kafka Producer to produce to a topic

Basic Kafka Producer

```dotnet-cpp-cli
using System;
using System.Threading.Tasks;
using Confluent.Kafka;

public class Producer
{
    public static async Task Main(string[] args)
    {
        var config = new ProducerConfig { BootstrapServers = "localhost:9092" };

        using (var p = new ProducerBuilder<Null, string>(config).Build())
        {
            try
            {
                var dr = await p.ProduceAsync("orders-to-bake",
                    new Message<Null, string> { Value = "new cupcake order" });
                Console.WriteLine($"Delivered '{dr.Value}' to '{dr.TopicPartitionOffset}'");
            }
            catch (ProduceException<Null, string> e)
            {
                Console.WriteLine($"Delivery failed: {e.Error.Reason}");
            }
        }
    }
}
```

Basic Kafka Consumer

```
using System;
using System.Threading;
using Confluent.Kafka;

public class Consumer
{
    public static void Main(string[] args)
    {
        var conf = new ConsumerConfig
        {
            GroupId = "cupcake-bakers",
            BootstrapServers = "localhost:9092",
            AutoOffsetReset = AutoOffsetReset.Latest
        };

        using (var c = new ConsumerBuilder<Ignore, string>(conf).Build())
        {
            c.Subscribe("orders-to-bake");
            while (true)
            {
                try
                {
                    var cr = c.Consume();
                    Console.WriteLine($"Consumed message '{cr.Value}' at: '{cr.TopicPartitionOffset}'.");
                }
                catch (ConsumeException e)
                {
                    Console.WriteLine($"Error occured: {e.Error.Reason}");
                }
            }
        }
    }
}
```

Questions?

Thank You!

- Microsoft DevBoston Meetup @DevBostonDotOrg
- Meetup organizer Jason Haley @haleyjason
- Meetup host Acadian Asset Management
- Meetup host Bryan Hogan @bryanjhogan
- Books sponsor Viktor Gamov (Confluent) @gAmUssA

Thank You!

- GitHub: srigumm/eCupcakesFactoryOnContainers
- Demo UI: http://bit.do/cupcakery
- http://bit.do/cupcake-meetup

Srini
srigumm

Tribalscale
Boston
linkedin.com/in/srigumm

Jacob Zweifel
jzweifel

@TribalScale
Boston, MA