Welcome to the

Work from Home Webinar and Ask the Expert Series

Every Friday @ 11am PT
Full schedule to be published shortly

The Southern California team hopes you and your families are healthy and safe!

SoCal MongoDB Team

Sigfrido "Sig" Narvaez
Solutions Architect
sig@mongodb.com

John Dohoney
Solutions Architect
john.dohoney@mongodb.com

Dan Midura
Regional Director
Dan.Midura@mongodb.com

Matt Quinn
Account Executive
matthew.quinn@mongodb.com

Eric Diggins
Account Executive
eric.diggins@mongodb.com

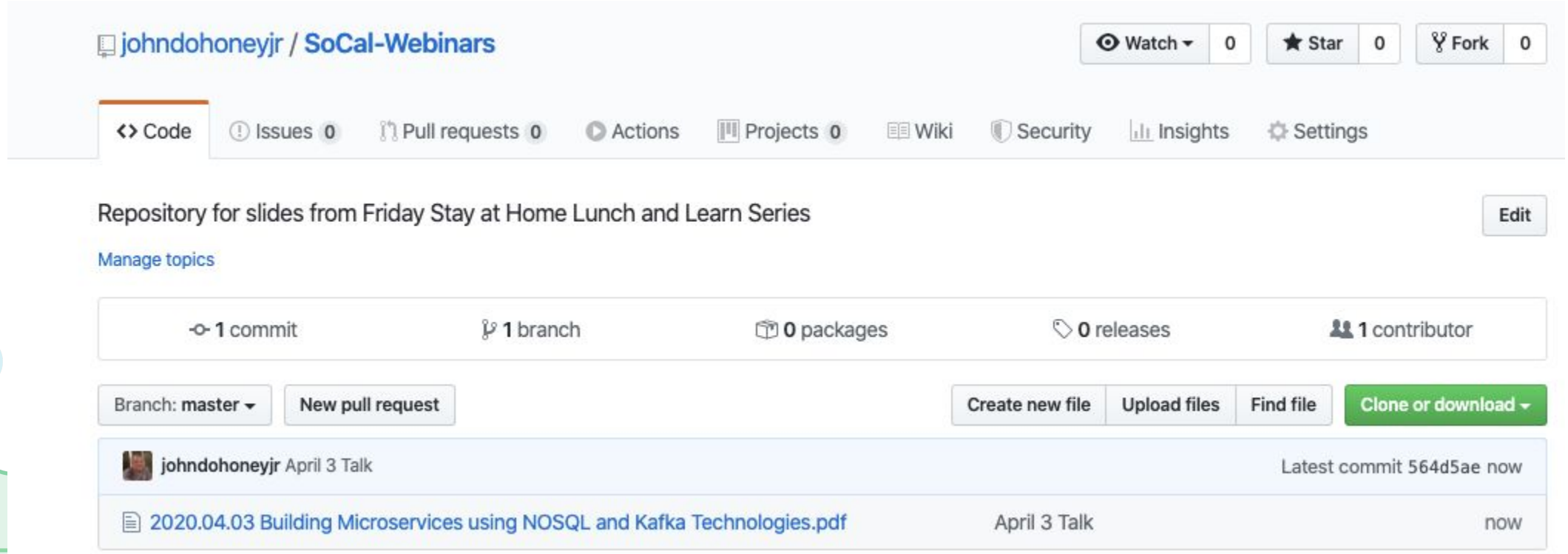
Danny Govea
Account Executive
danny.govea@mongodb.com

Miles King
Account Executive
miles.king@mongodb.com

Kyle Wilgus
Account Executive
kyle.wilgus@mongodb.com

Presentation Slides

https://github.com/johndohoneyjr/SoCal-Webinars



SoCal MongoDB Webinar Series

Building Microservices using MongoDB and Kafka Technologies

John Dohoney, Jr.

Senior Solution Architect

@johndohoneyjr

April 3 2020

Agenda

Digital Transformation

What it is not…

Why bother?

Are Microservices a lot of hype?

Lift and Shift Anti-Pattern

Strangle? Is that Legal?

"DDD" is not a part of the Government

But I like Waterfalls…

How can technology Help?

Redesign, or perpetuate the legacy?

How can technology Help? (continued)

Document Model — What I don't have to do

Schema-less, or Schema When I need it

Distributed Computing

Why Cross Region is really HA

Scale Independently — Kafka and Infrastructure as Code

Disclaimer

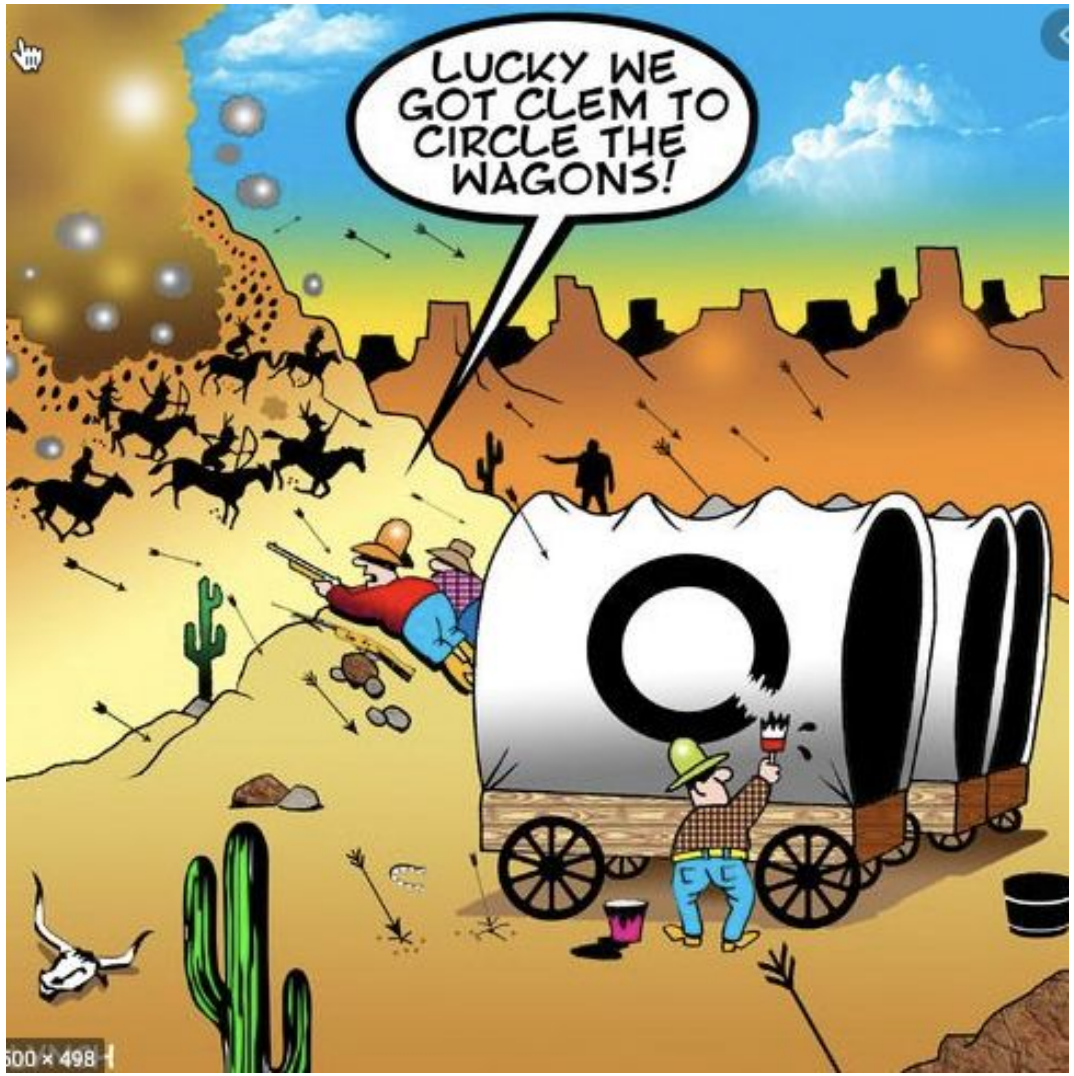
We know the home office can be a little crazy.

If the “Indians” can’t sit still, take care of them, this will be recorded

If they have a question, let them ask it, I am game. All Q&A will be at the end of the talk.

Any pressing questions, put them in the Zoom Chat

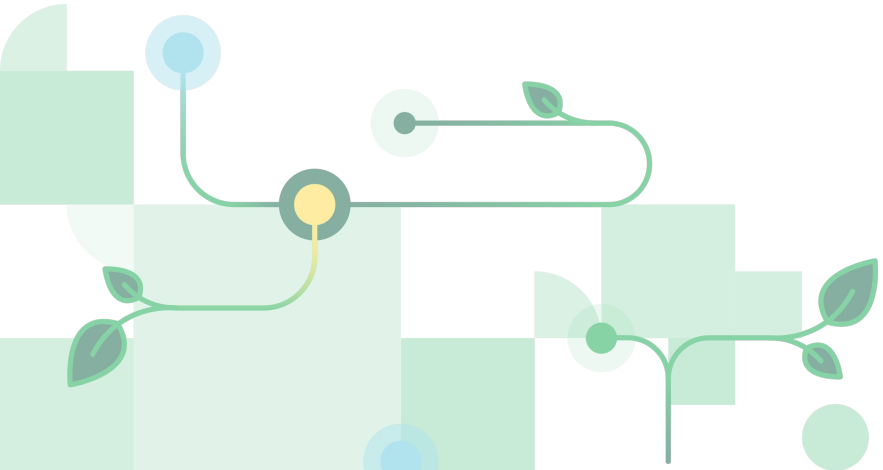
ENJOY…..



Digital Transformation?

It's Moving to the cloud...right?

No

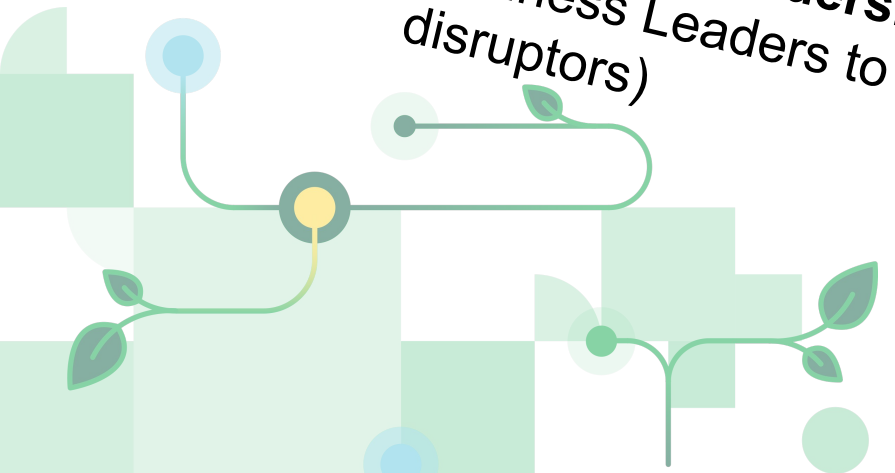


Digital Transformation does include Technical elements, but more so as technology affects the outcome of:

Process and Process improvement (Agile transitions, shorter release cycles, smaller quantums of work to affect change)

Cultural changes *(DevOps -- breaking down barriers to productivity between development and Operations)*

Corporate Leadership *(from Business Leaders to Sector disruptors)*



Improvements in Customer Experience (Personalized, targeted, Omni-Channel)

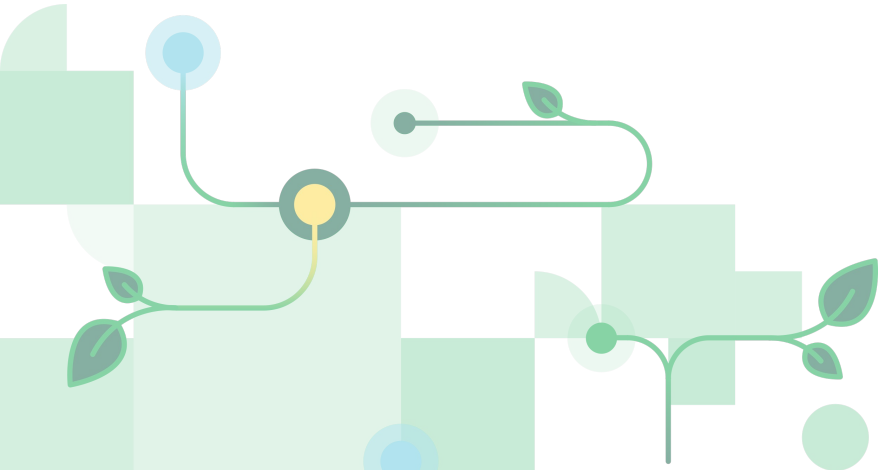
How do Microservices fit in

Smaller units of work

Fits into shorter release cycles

Different packaging -- Containers and Pods with all dependencies internal to the deployment unit

Supports autonomous operations (Scale Out, Scale in, Load Balancing, Self-Healing, Circuit Breaking (cascading errors)



Does Lift and Shift help get us there?

- Well, sort of, as long as it is not the end. It must be step one in a larger process that we will explain.

- If it is just the end, all it accomplishes is moving the point of operations from a data center to the cloud

So, assuming this is step one, what is next ….

We decompose our Monolith, in real time ...

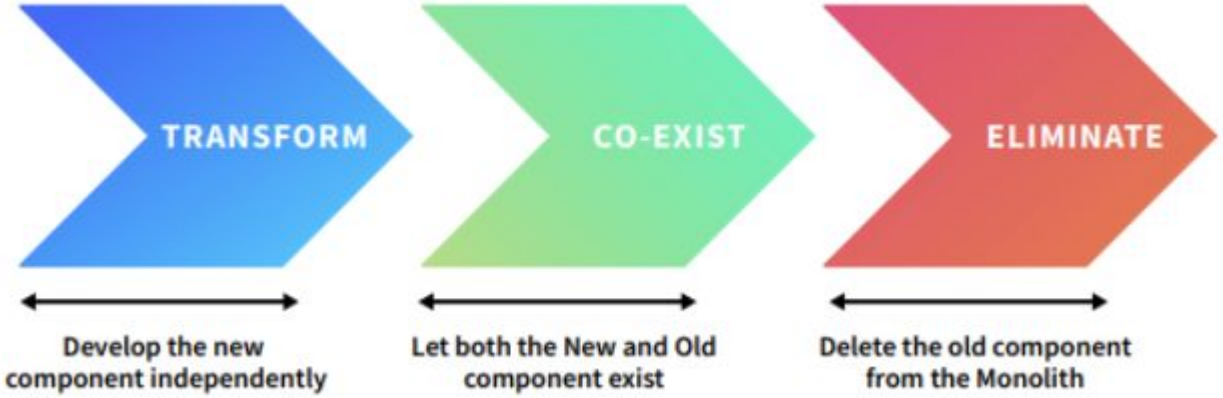
Based on the the “Strangler Application” presented by Martin Fowler.

Fowler suggests “Asset capture” as the way to migrate assets to “micro” services

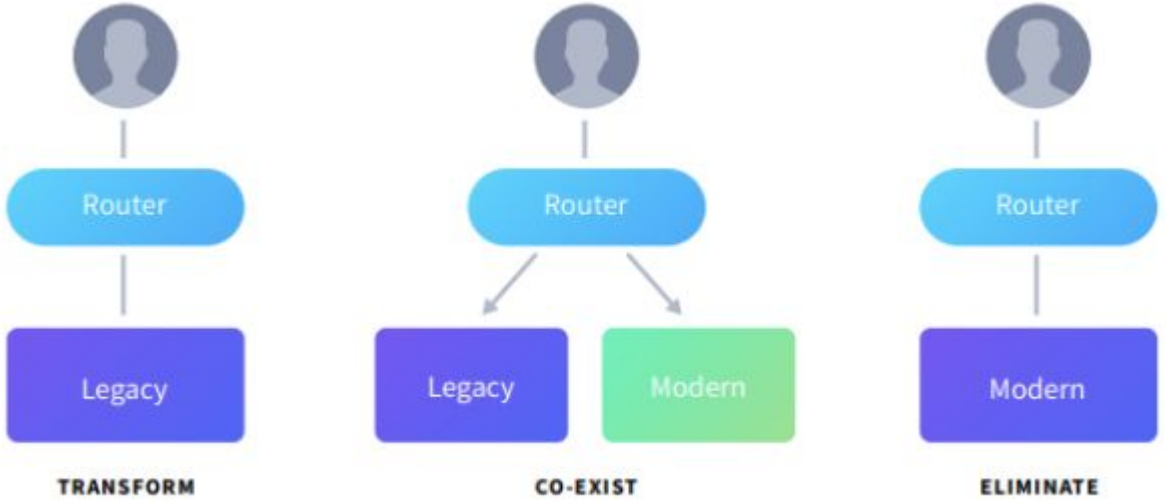
He also suggest usage of an Event Interceptor so all events are captured, as well as the Content Based Router

that is way too much theory, how does it work ?...

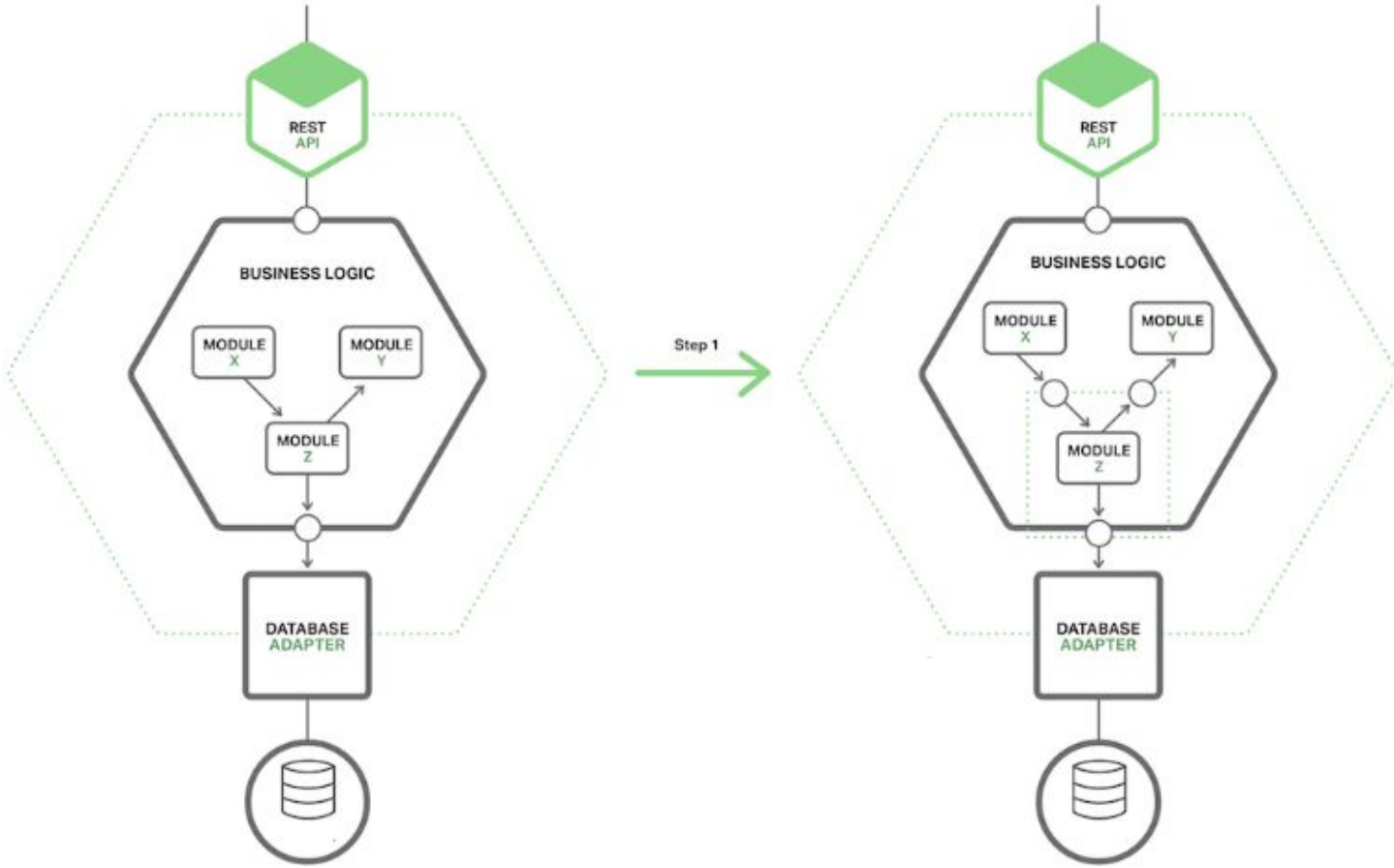
Logical Transform and Eliminate



Physical Transform and Eliminate



LIft and Shift (or, could be done as a Hybrid Cloud play)



Data Center

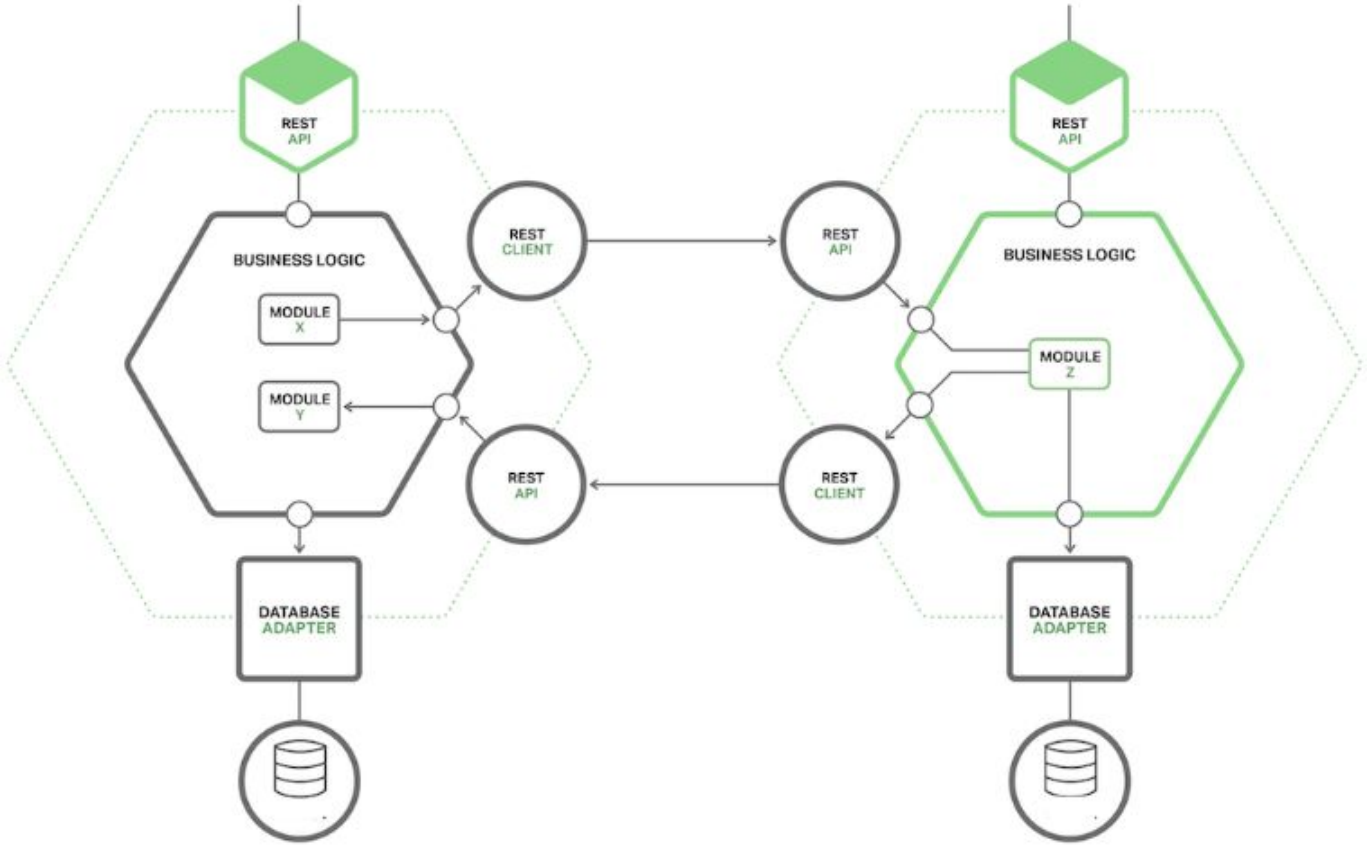
Cloud Provider

Transform and Eliminate (Strangle)

Two schools of thought on “Strangling”

1. “Two Way” communication
   Allow the service to call back to its dependent services in the Monolith

2. “One Way” Communication
   Microservices CAN NOT call back to the Monolith. Pro’s Breaks Monolith dependency early, Con’s typically you move more than one microservice (service with its dependent services)

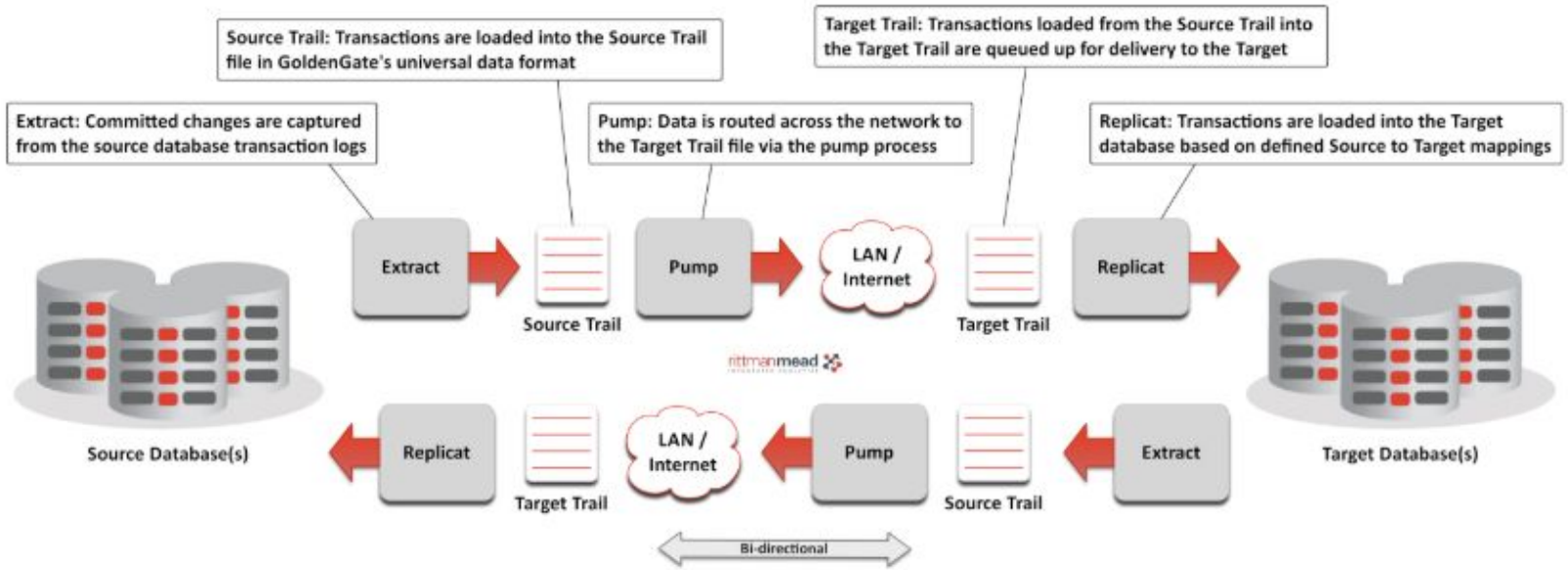


Cloud Provider

How do we transform Legacy Data for Agile and Microservices?

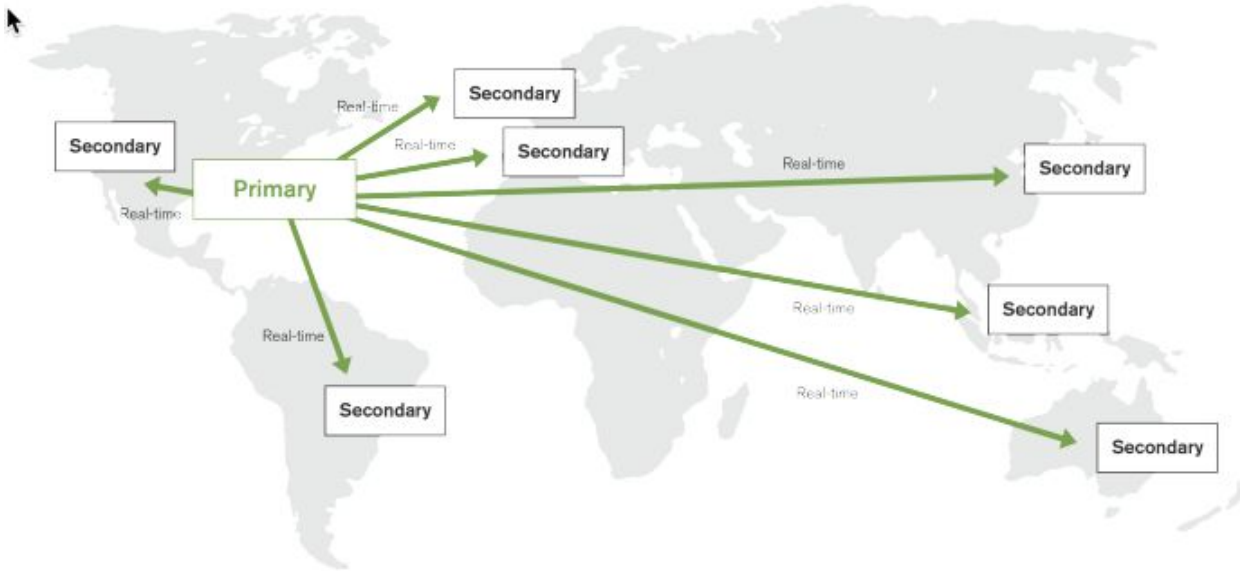
Moving from an On-Prem to a Cloud Relational Model still:

1. Requires adherence to a rigid schema
2. Imposes tasks on the development pipeline, otherwise would not be necessary
3. Limits Data Design
4. Static in Nature, hard to evolve
5. Operational Limitations with Vertical Scaling
6. Expensive to Scale Globally, requires staff with special skills

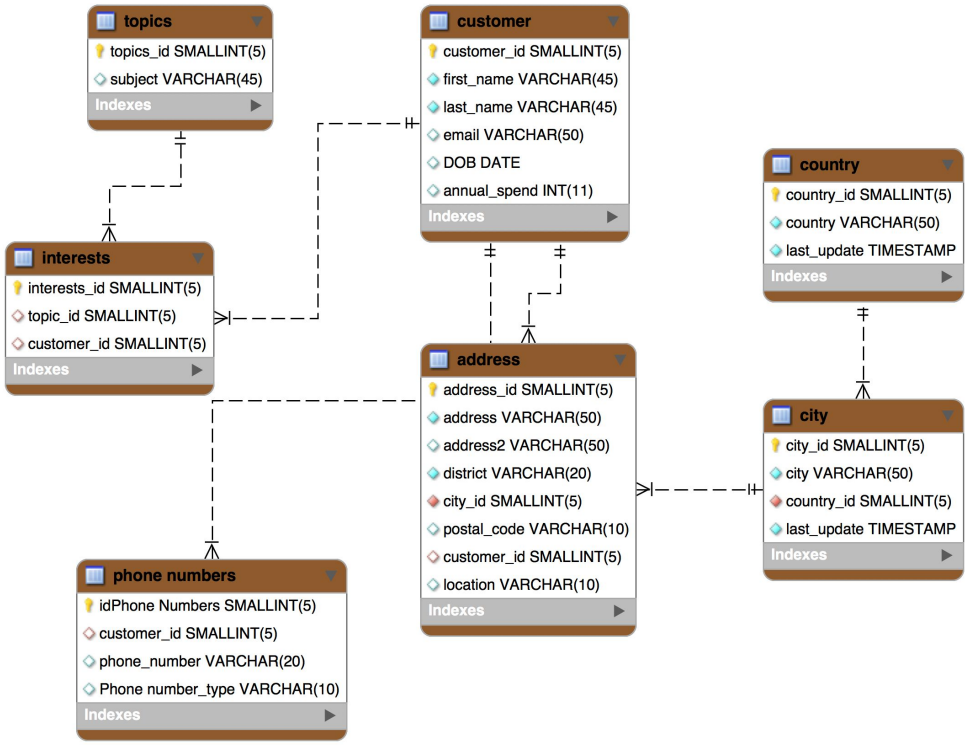


The Big Ideas are (Simplicity, Completeness, and Operational Excellence)

1. Apply schema when necessary, it does not have to be an impediment to progress
2. No tickets to push schema changes, No ORMs, and No Object Relations Mapping (Code objects are serialized as is, to the database)
3. Sub-Documents can be designed differently than parent documents
4. Dynamic in Nature, easy to evolve
5. Simple Vertical Scaling (up and down)
6. Global Scale is as simple as knowing where to put end points in the Global Cluster, your data sizes, and capabilities to operate a simple, intuitive GUI



Easy: Contrasting data models



Tabular (Relational) Data Model

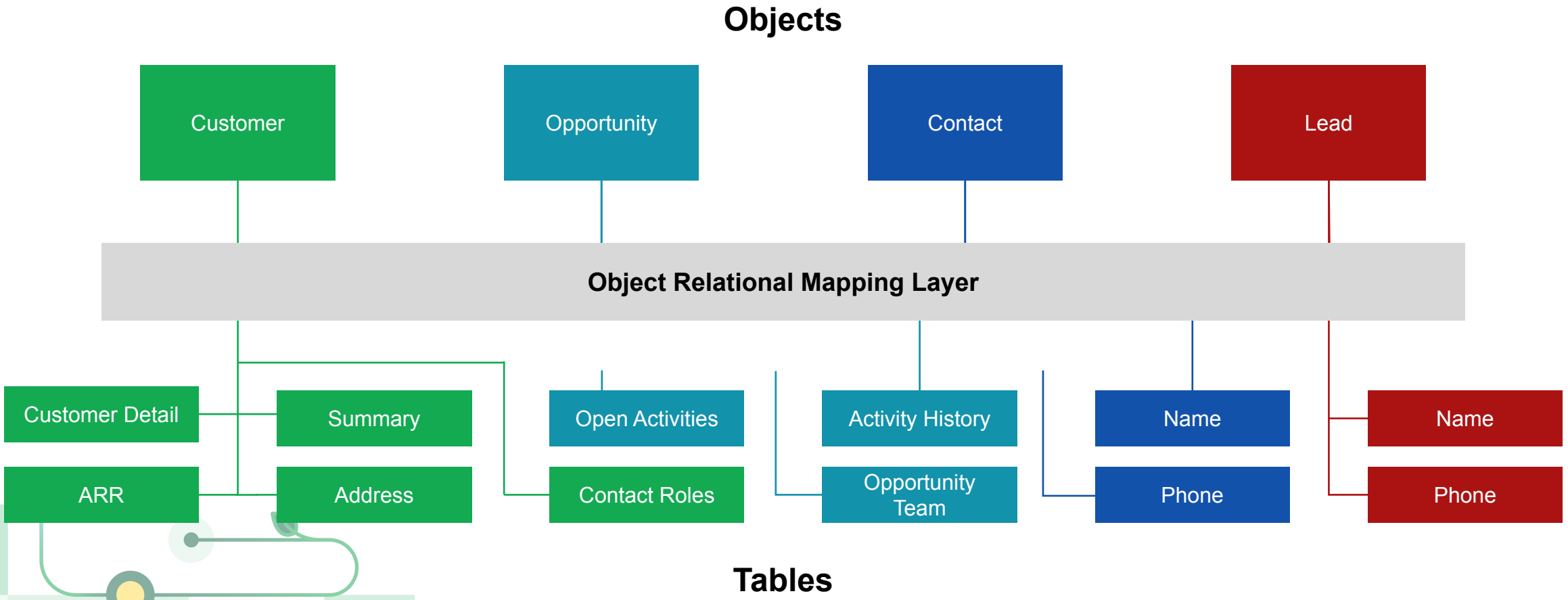
Related data split across multiple records and tables

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : {
    "first" : "John",
    "last" : "Doe" },
  "address" : [
    { "location" : "work",
      "address" : {
        "street" : "16 Hatfields",
        "city" : "London",
        "postal_code" : "SE1 8DJ"},
      "geo" : { "type" : "Point", "coord" : [
        51.5065752,-0.109081]}},
+   {...}
  ],
  "phone" : [
    { "location" : "work",
      "number" : "+44-1234567890"},
+   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

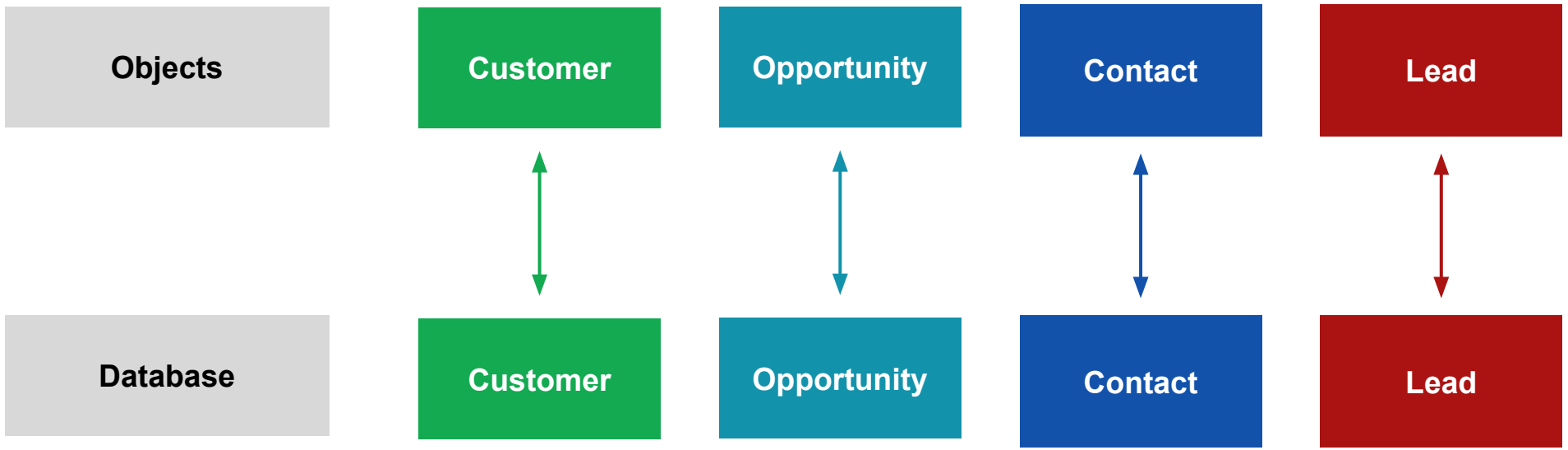
Document Data Model

Related data contained in a single, rich document

Go from this….

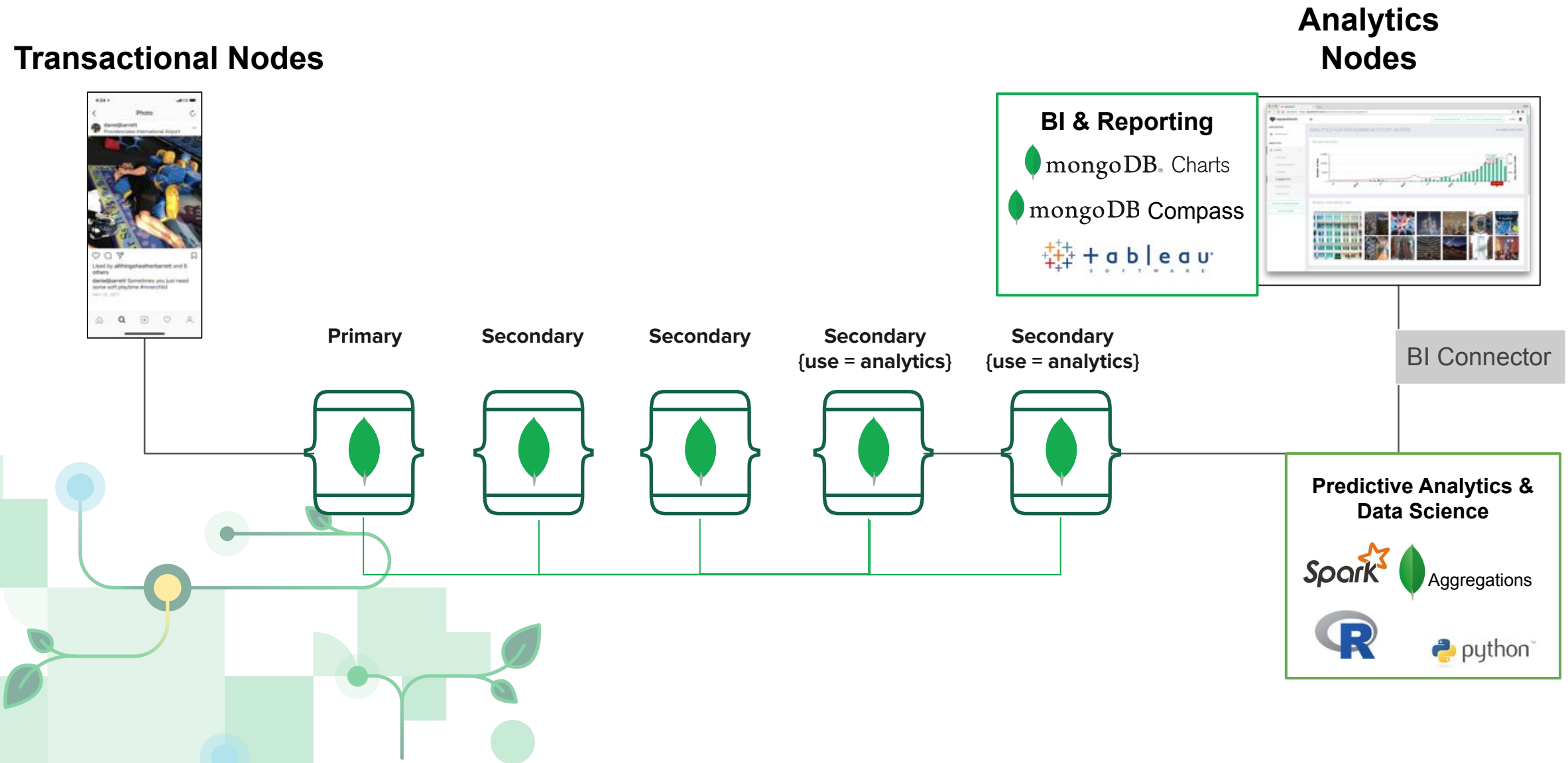

Objects
Customer
Opportunity
Contact
Lead
Object Relational Mapping Layer
Customer Detail
Summary
Open Activities
Activity History
Name
Name
ARR
Address
Contact Roles
Opportunity Team
Phone
Phone
Tables

To this: store objects directly…

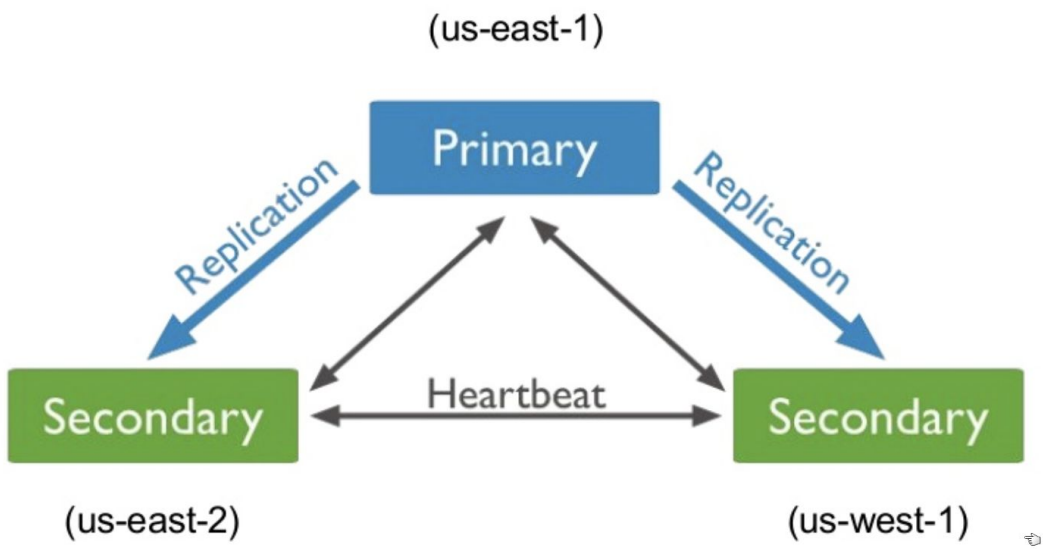


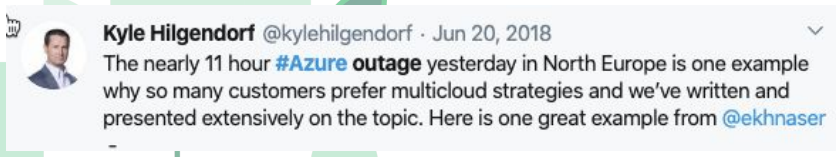
Simple and functional Cluster Design

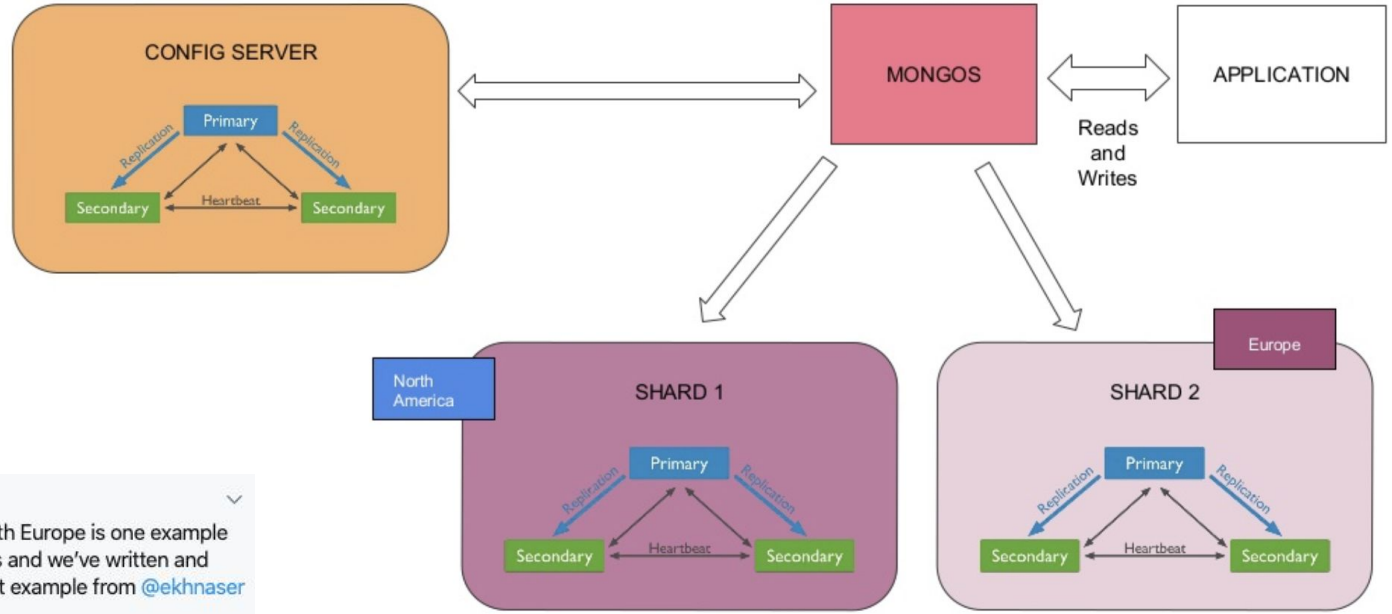
No need for ETL - Co-located operational and analytical workloads



Regional Stretch Clusters improve on “Multi-AZ” Clusters



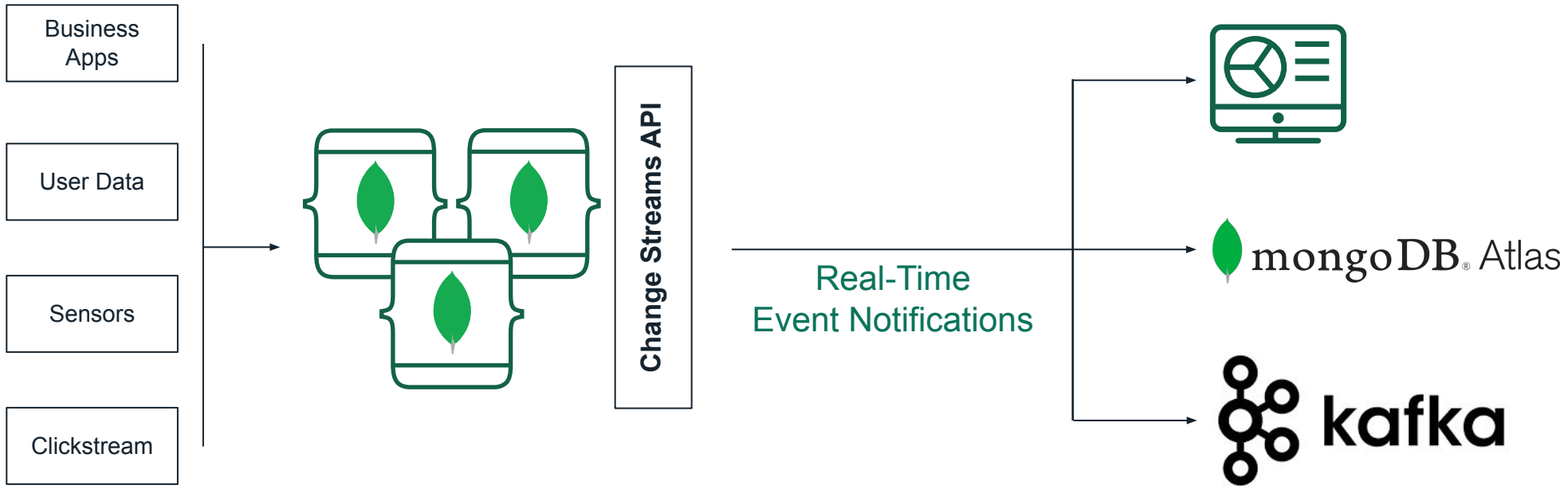




What about plugging into the rest of the Enterprise ...

- Processes need to be inserted into pipelines to Transform, Enhance, Aggregate, and Summarize raw data into actionable information
- The architecture should enforce loose coupling and independent scaling
- Support asynchronous, reactive, and event driven systems design
- Latency should be minimized between data sources and data sinks
- Allow for secure data transmission into and out of the enterprise

Versatile: MongoDB Change Streams



Enabling developers to build
reactive, real-time services

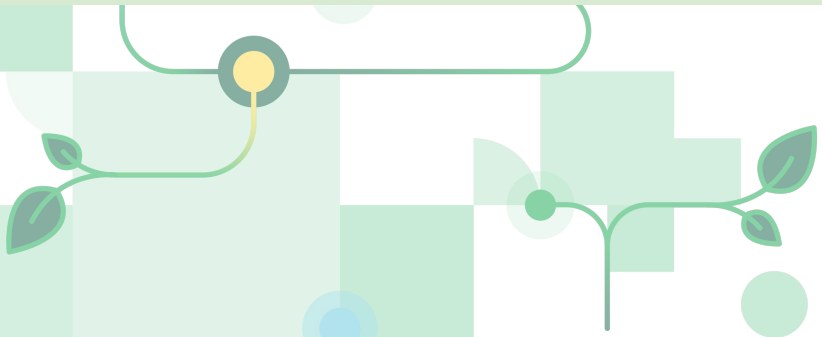
MongoDB Connector for Apache Kafka

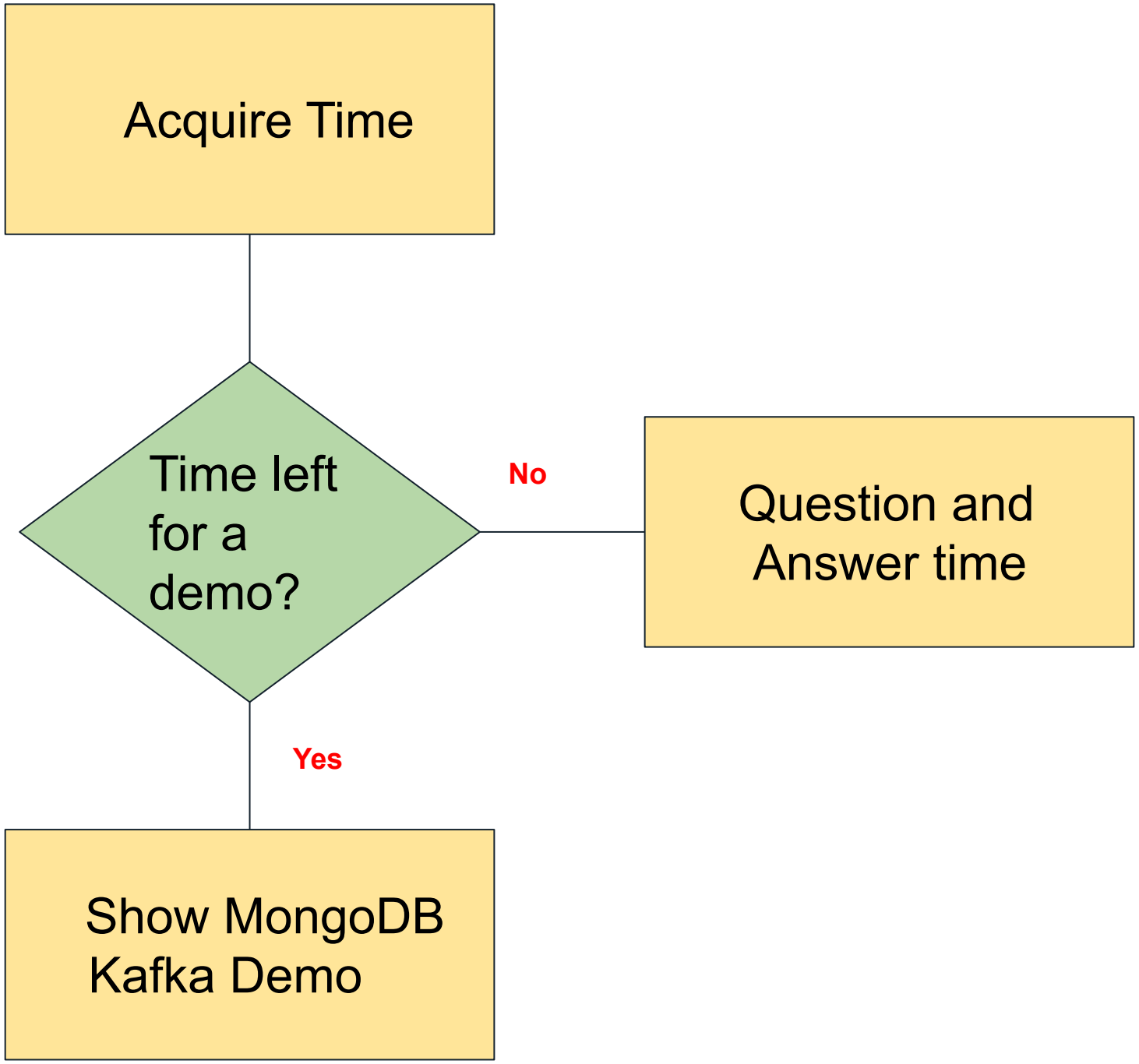
- Build robust data pipelines for microservices and Event Driven Architectures
- Developed with the community and supported by MongoDB engineers, verified by Confluent
- Supports MongoDB as a sink and a source for Kafka
- Integrate with Change Streams and Atlas triggers to create fully reactive, event driven pipelines

GitHub Confluent Hub

A little configuration goes a long way - NO CODE

```
curl -X POST -H "Content-Type: application/json" --data '
  {"name": "mongo-sink",
   "config": {
     "connector.class":"com.mongodb.kafka.connect.MongoSinkConnector",
     "tasks.max":"1",
     "topics":"pageviews",
     "connection.uri":"mongodb+srv://johndohoney:TopSecret@demo-store-hbwxn.mongodb.net/test",
     "database":"test",
     "collection":"pageviews",
     "key.converter": "org.apache.kafka.connect.storage.StringConverter",
     "value.converter": "org.apache.kafka.connect.json.JsonConverter",
     "value.converter.schemas.enable": "false"
}}' http://localhost:8083/connectors -w "\n"
~
```




Acquire Time
Time left for a demo?
No
Question and Answer time
Yes
Show MongoDB Kafka Demo

db.runCommand(“Thank you!!”);

db.QnA.find({}).explain();