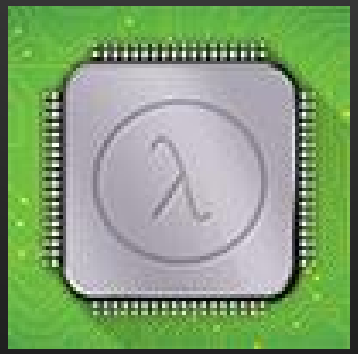
Reactive Real-time Big Data

with Open Source Lambda Architecture

Make Big Data as simple as possible, but not simpler



About speakers

Nguyễn Tấn Triều
from FPT Online

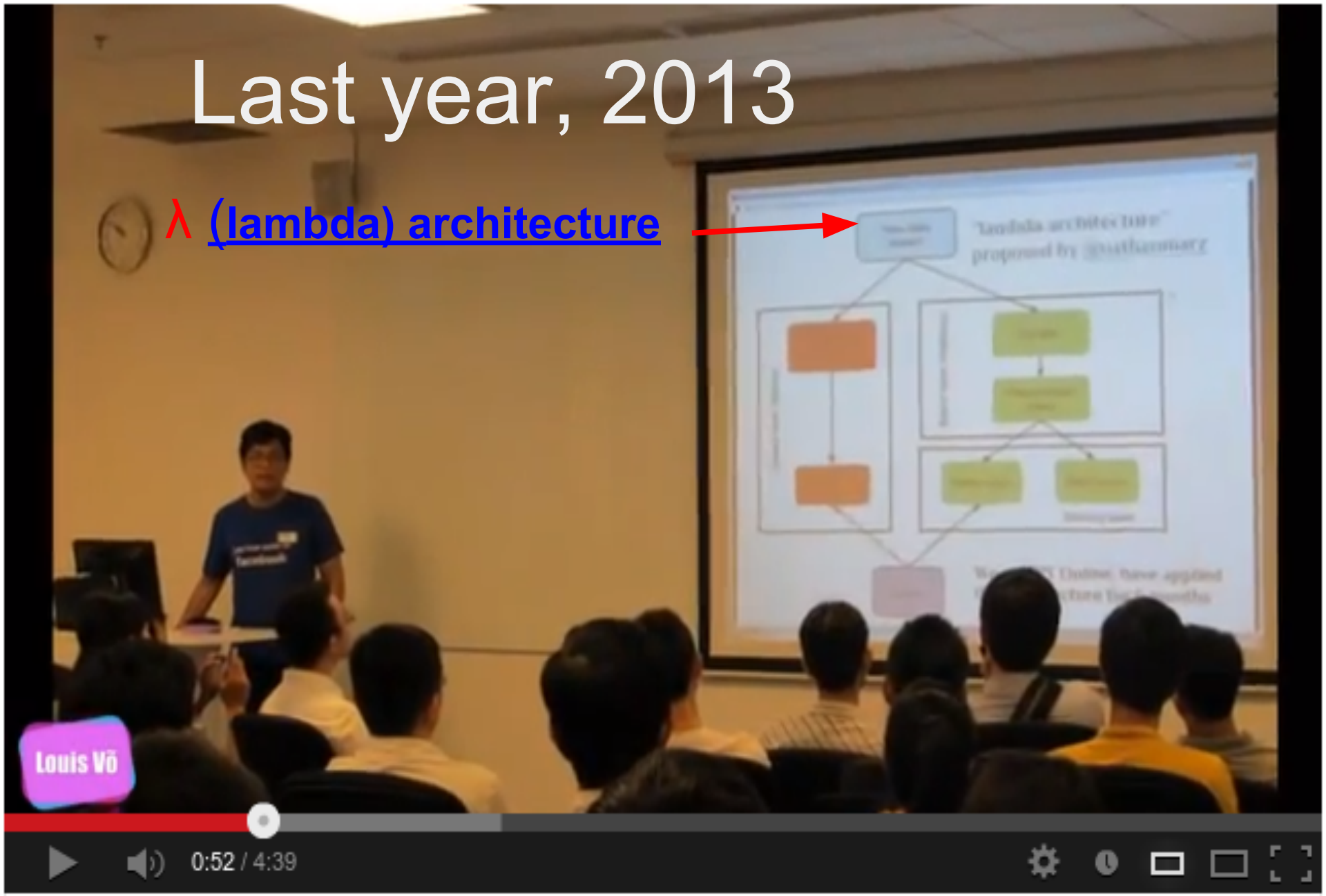
Lê Kiến Trúc
from InfoNam

Personal blog:

http://nguyentantrieu.info/blog

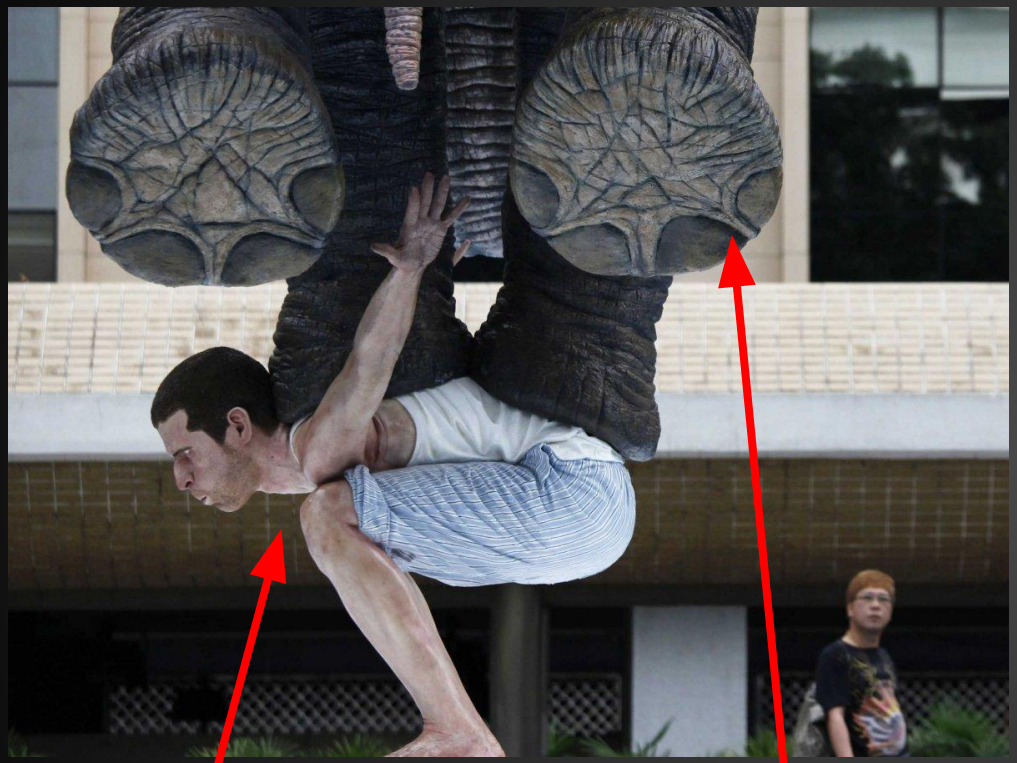
Reactive Big Data Blog:

http://www.mc2ads.com



Barcamp Saigon 2013

This year, 2014



1) Human

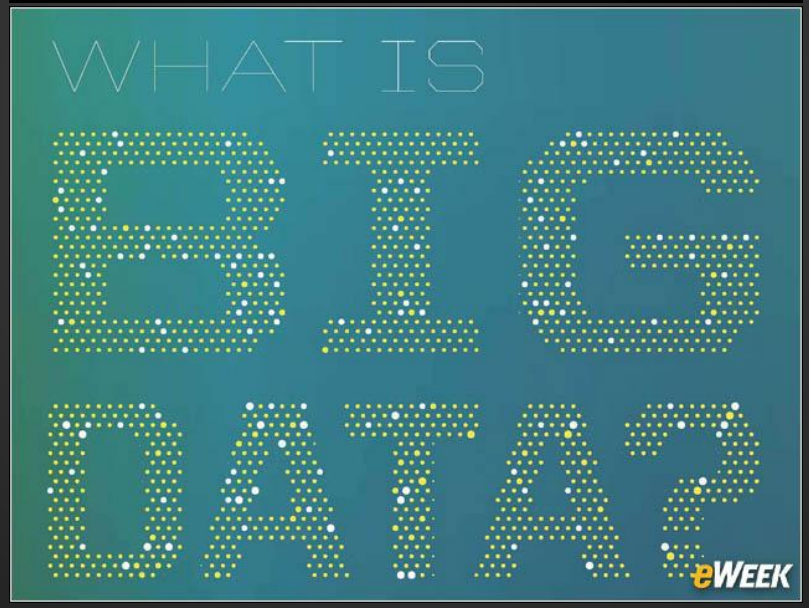
2) Big Data

3) Flappy Bird

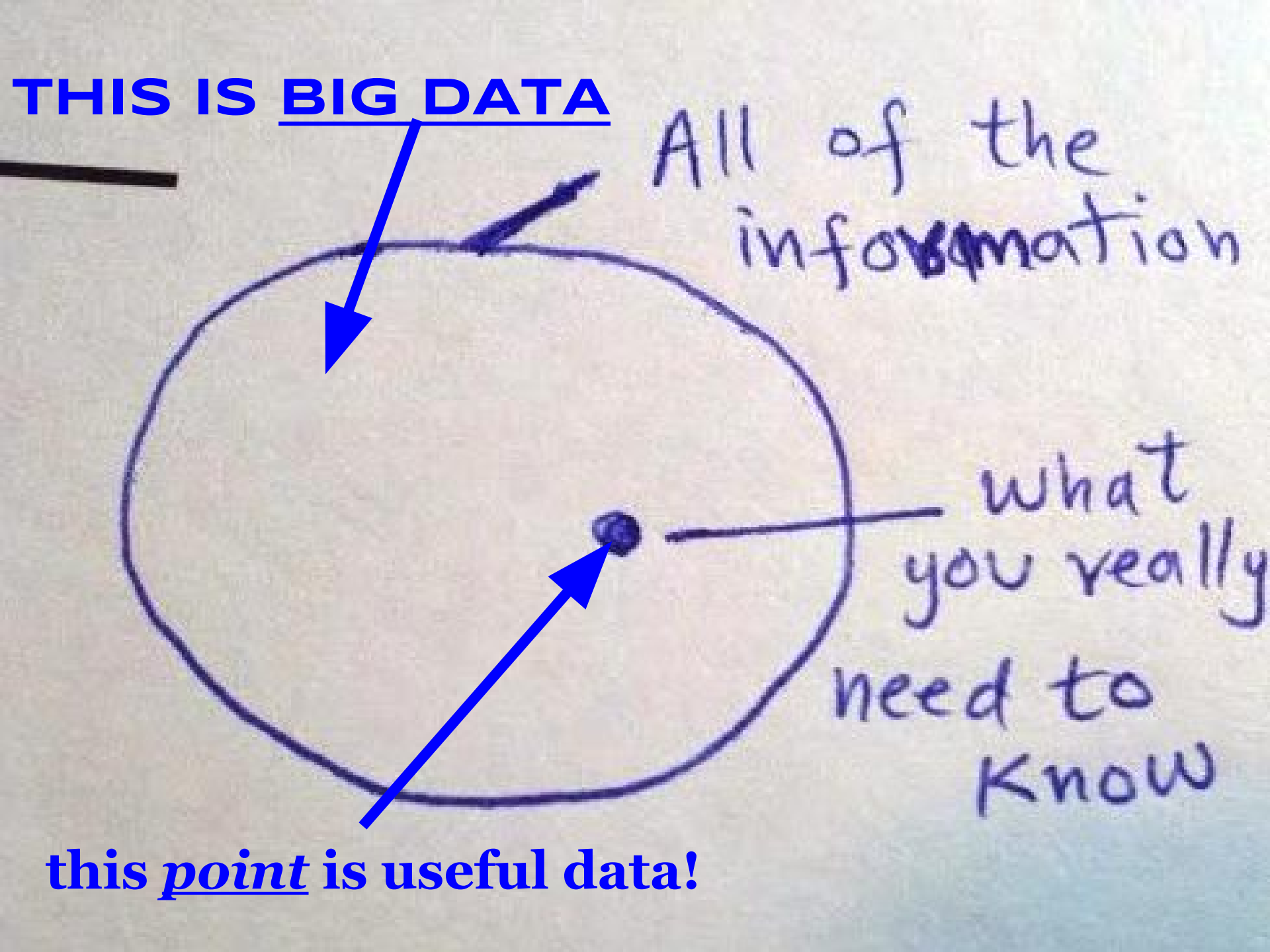
Contents

1. ***Big Data***, we will see in ONE picture
2. ***Demands*** → ***Realtime***
3. ***Solutions*** → ***Reactive***
4. ***Dreams*** in Data-Driven World in 21st century

We live in the Matrix ?



1) BIG DATA is ...?


THIS IS BIG DATA
All of the information
what you really need to know
this point is useful data!

Big Data can solve our problems?

NO

Big Data is just a buzzword.

You need (3R):

1. Solve right problems
2. Build the right team
3. Use right tools

2) Demands from Business

Big Data can solve these problems?

1. Predicting the future disasters?
2. Understanding our customers better?
3. Optimizing marketing campaigns in realtime?

Let's see 3 problems


Climate change the state of the science (data visualization)
Weather forecast
"many provinces in the Mekong Delta will be flooded by the year 2030"
→ Disaster Response System
Source:
http://en.wikipedia.org/wiki/Mekong_Delta
http://www.wired.co.uk/news/archive/2013-10/28/predicting-disasters
Vietnam
CYCLONE TRACKS
FROM 2000 TO 2012
FLOODED LANDS
SEA-LEVEL RISE OF 1 METER
PROBABILITY RANGE
LIKELY
66%
100%
Exit full screen
2:12 / 4:04

Connecting user data with social activity data ?

e.g: MySQL + Facebook Social Graph

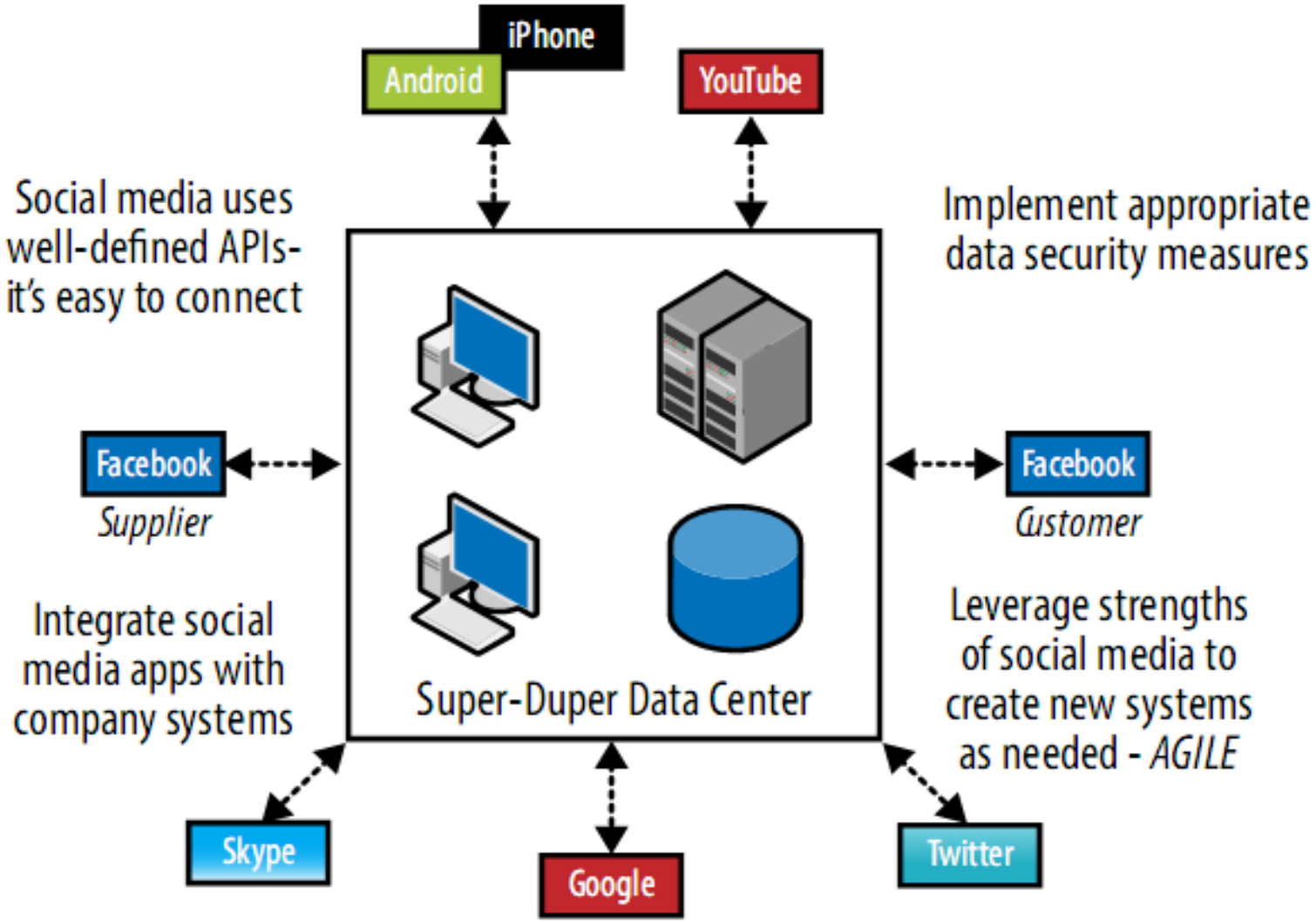
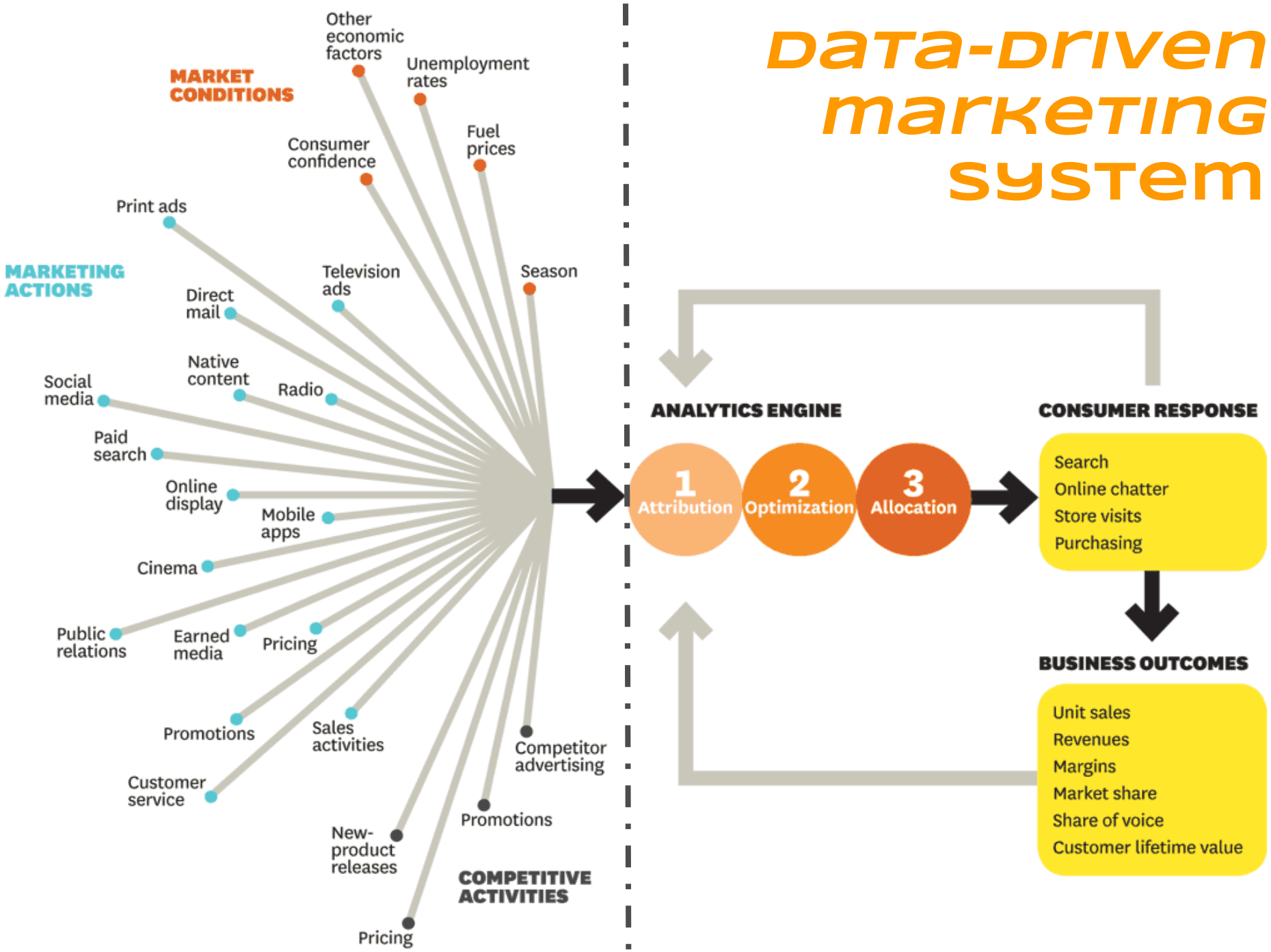


Figure 4-2. Social business connections

Data-driven marketing system



3) Solutions: the architecture and tools

Big data Ecosystem

- **Frameworks**: Hadoop Ecosystem, Apache Spark, Apache Storm, Facebook Presto, Storm, ...
- **Patterns**: MapReduce, Actor Model, Data Pipeline, ...
- **Platforms**: Amazon Redshift, Cloudera, Pivotal, HortonWorks , IBM, Google Compute Engine, ...
- **Best Practices**:
  - How Heineken Interacts With Customers Using Big Data
  - How Nestlé Understands Brand Sentiment Of 2.000 Brands In Real-time

Source:
http://azadparinda.wordpress.com/2013/10/11/projects-other-than-hadoop/
http://www.bigdata-startups.com/best-practices

Wait, is Hadoop the best solution?

Top *4 limitations* of Mapreduce

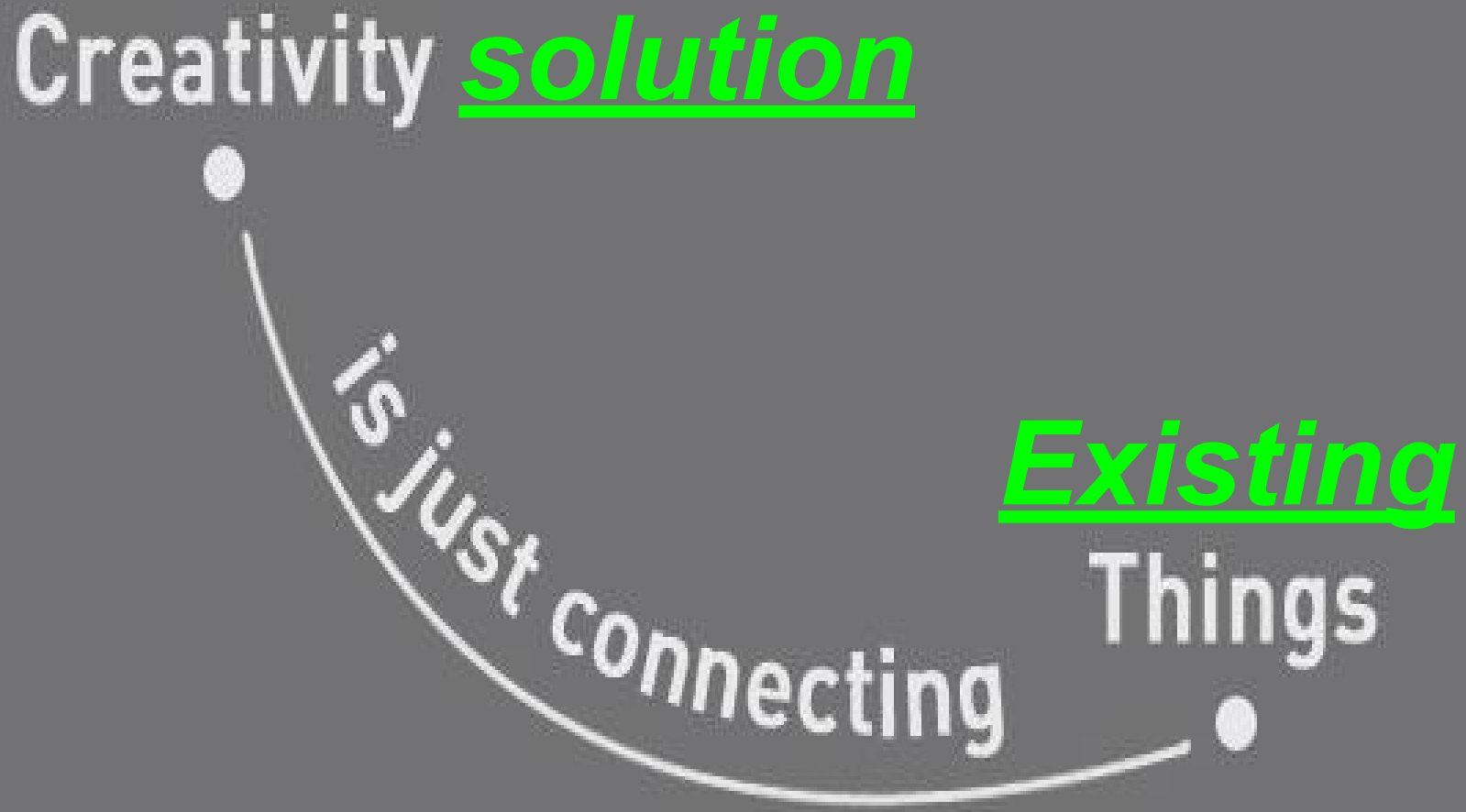
1. Computation depends on *previously computed values*
2. Full-text indexing or *ad hoc searching*
3. Algorithms depend on *shared global state*
4. Online learning, aka: *stream mining* (Reactive Functor will fix this issue)

Source:

http://csci8980-2.blogspot.com/2012/10/limitations-of-mapreduce-where-not-to.html

It's not {Realtime, Responsive}

→ Let's find out new creative solution


Creativity solution
is just connecting
Existing
Things

Stanford Report, June 14, 2005

'You've got to find what you love,' Jobs says

This is a prepared text of the Commencement address delivered by Steve Jobs, CEO of Apple Computer and of Pixar Animation Studios, on June 12, 2005.

I am honored to be with you today at your commencement from one of the finest universities in the world. I never graduated from college. Truth be told, this is the closest I've ever gotten to a college graduation. Today I want to tell you three stories from my life. That's it. No big deal. Just three stories.

The first story is about connecting the dots.

I dropped out of Reed College after the first 6 months, but then stayed around as a drop-in for another 18 months or so before I really quit. So why did I drop out?

Video of the Commencement address.

RELATED TO THIS STORY

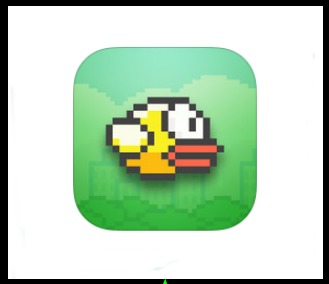
Steve Jobs' 2005 Stanford Commencement Address

» 2005 Stanford Commencement coverage

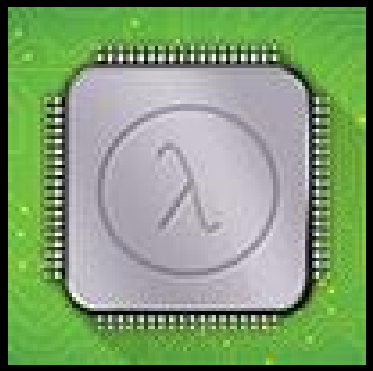
MORE STANFORD NEWS

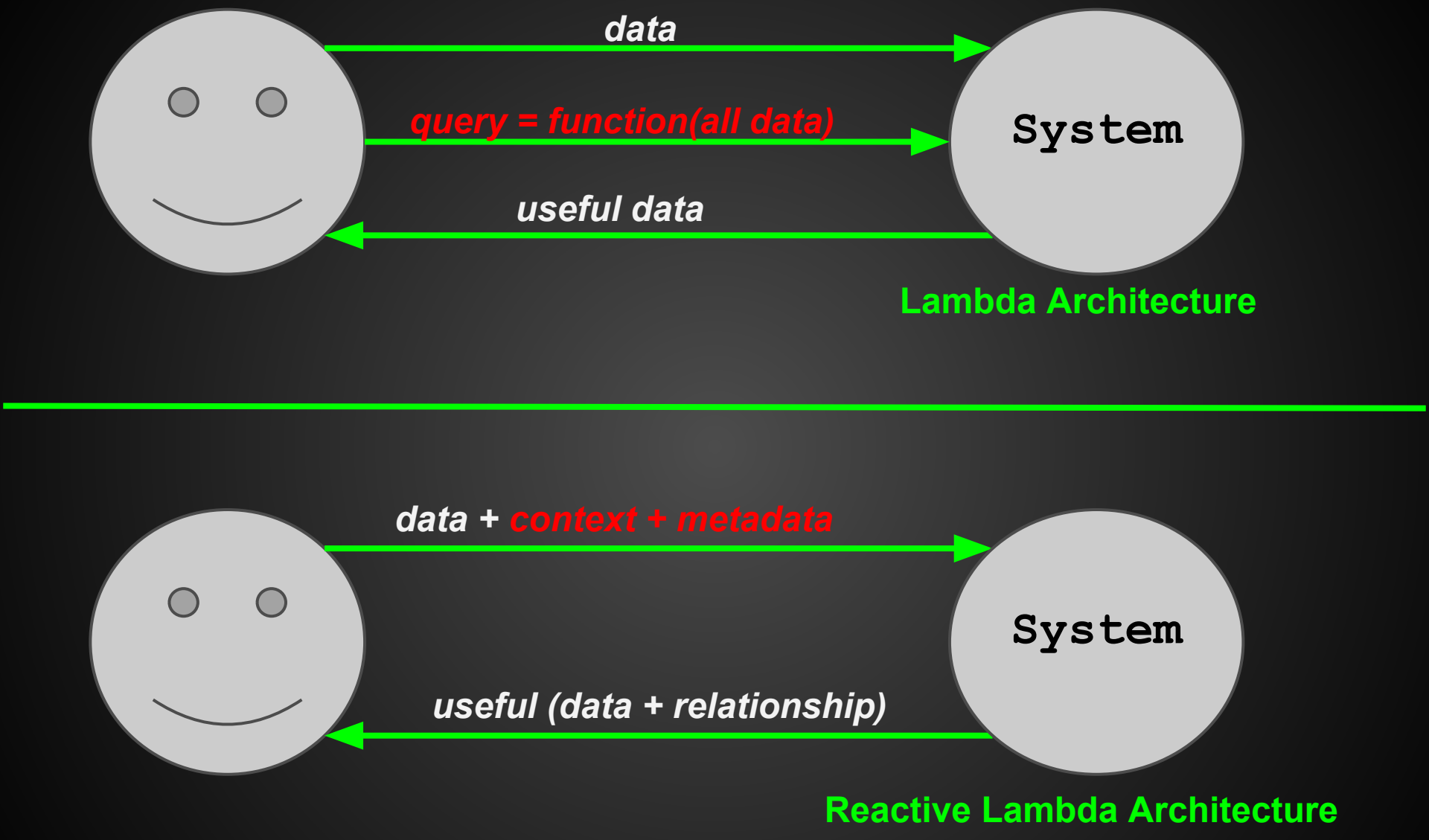
RECENT POPULAR SUBSCRIBE

Stanford economist suggests how to better allocate donor organs to increase their availability



Lambda + REACTIVE ⇢ CREATIVE




data
query = function(all data)
useful data
System
Lambda Architecture
data + context + metadata
useful (data + relationship)
System
Reactive Lambda Architecture

Home

About this project

Reactive Functor Database - the framework and a Unifying NewSQL Database for Reactive Big Data

Key philosophy “readily responsive to a stimulus”, aka: readily responsive to {data, context, relationship}. The data, the user and the context store as ONE entity. So we could do mining, analytic, search and recommend information faster in real-time.

the implementation: adapting Reactive Lambda Architecture

Backend

- Theory: [Actor Theory of Distributed Computation] http://arxiv.org/pdf/1008.1459.pdf , Code: http://akka.io Akka Framework
- Reactive Programming, http://www.reactivemanifesto.org , Code: RxJava Programming, https://github.com/Netflix/RxJava/wiki
- Graph Database https://github.com/orientechnologies/orientdb/wiki/Graph-Database-Tinkerpop
- Speed layer query using In-Memory NoSQL Redis http://redis.io
- Batch layer using Hadoop and Apache Spark http://spark.apache.org
- Networking tih Netty http://netty.io

Core concepts of **Reactive Lambda Architecture**

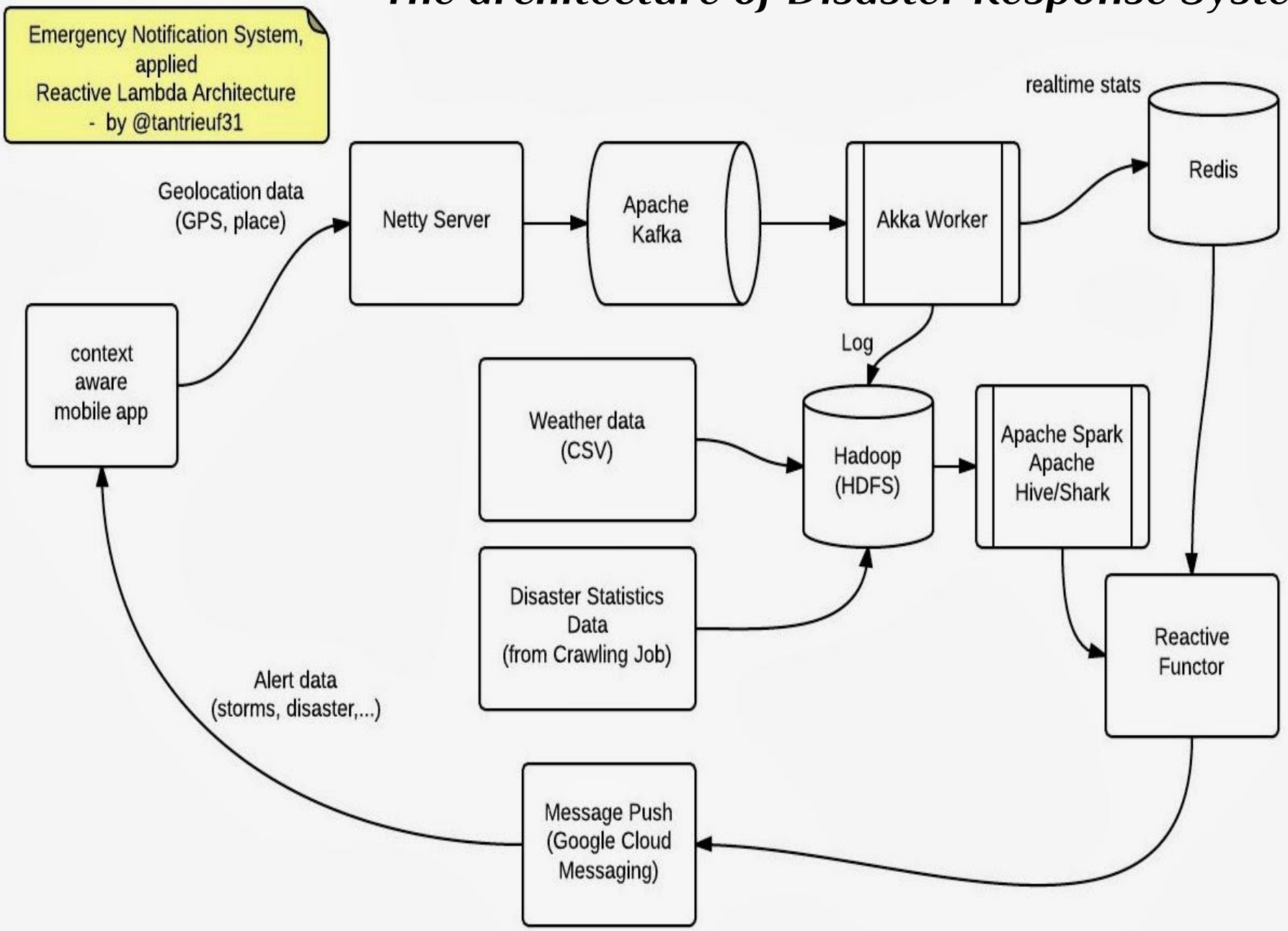
- **Reactive Functor**: functional ***actor*** that receives and responses data *reactively* to event source and context (just like *neuron cell* in your ***brain***)
  - *Original ideas, are got from my advisor in 2007*

Source: http://activefunctor.blogspot.com

- **Lambda Architecture**: the hydrid model, named by Nathan Marz, a software engineer at **twitter**.com for designing Big Data system with 3 core layers
  - Speed layer: query stream data (realtime processing)
  - Serving layer: query analyzer
  - Batch layer: query all data (batch processing)

Source: http://www.manning.com/marz

The architecture of Disaster Response System

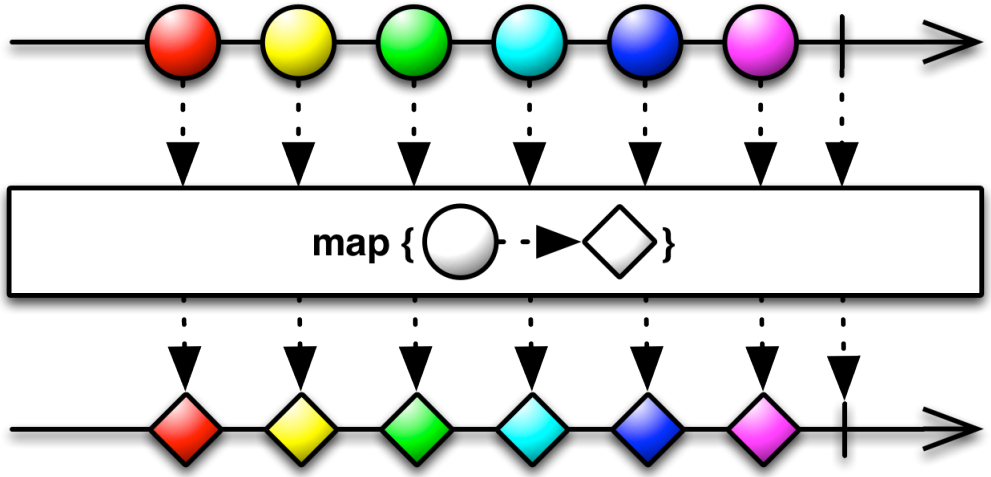


Why reactive ?

It's the *philosophy and pattern* for *designing a large application* at *Internet-scaled*.

There are 4 attributes in a reactive architecture

1. **event-driven**
2. **scalable**
3. **resilient**
4. **responsive**



Monday, February 4, 2013

Functional Reactive in the Netflix API with RxJava

by Ben Christensen and Jafar Husain

Our recent post on optimizing the Netflix API introduced how our web service endpoints are implemented using a "functional reactive programming" (FRP) model for composition of asynchronous callbacks from our service layer.

This post takes a closer look at how and why we use the FRP model and introduces our open source project RxJava – a Java implementation of Rx (Reactive Extensions).

Embrace Concurrency

Server-side concurrency is needed to effectively reduce network chattiness. Without concurrent execution on the server, a single "heavy" client request might not be much better than many "light" requests because each network request from a device naturally executes in parallel with other network requests. If the server-side execution of a collapsed "heavy" request does not achieve a similar level of parallel execution it may be slower than the multiple "light" requests even accounting for saved network latency.

Futures are Expensive to Compose

Futures are straight-forward to use for a single level of asynchronous execution but they start to add non-trivial complexity when they're nested.

Links

Netflix US & Canada Blog

Netflix America Latina Blog

Netflix Brasil Blog

Netflix UK & Ireland Blog

Open positions at Netflix

Netflix Website

Facebook Netflix Page

RSS Feed

Suggest Me Something

GPS: 10.753402399999999,106.6291329 Address: phường 13, Quận 6, Ho Chi Minh City, Vietnam

Eat something around here

7 Kỳ Quan - Nhà Hàng & Cafe - Quận 6-TP.Hồ chí Minh - Diadiemanuong.com **Facebook Score: 0**
[Review] QUÁN 6 - TIỆM HỦ TIẾU LÂU NĂM BÀ SỬU đường Cao Văn Lầu F2, Q6. **Facebook Score: 0**
[quận 6] trà sữa bánh plan phô mai thạch củ năng cực ngon cực la **Facebook Score: 2**
Ốc Ni quận 6 - Ốc cà na chấm với nước chấm Ni - ngon, rẻ, sạch sẽ - ai ăn là ghiền! **Facebook Score: 2**

Slashing wind (chém gió) ?

Enough. Show me your demo and code

User story and Demo

Problem: **Real-time Marketing for Social Media 3.0**

User story's details:

1. User read news from a website → tracking user activities
2. User does login with Facebook Account
3. User clicks on like Facebook button → tracking what user liked
4. The marketer/data analyst should see the trending most read article in real-time
   - → Personalized articles for the reader
   - → ***Native advertising in real-time***

Client side:
HTML5
D3
JavaScript
Service-side:
Netty
Groovy

RxSQL Query Parser (RxGroovy + SQL)

Reactive Functor Graph Engine (Actor + OrientDB)

Realtime Database (**Redis**)

Batch Database (**HDFS** + **HBase**)

Data Collector (**Netty**)

Messaging (**Kafka**)

Stream Topology (**Storm API + Akka Actor**)

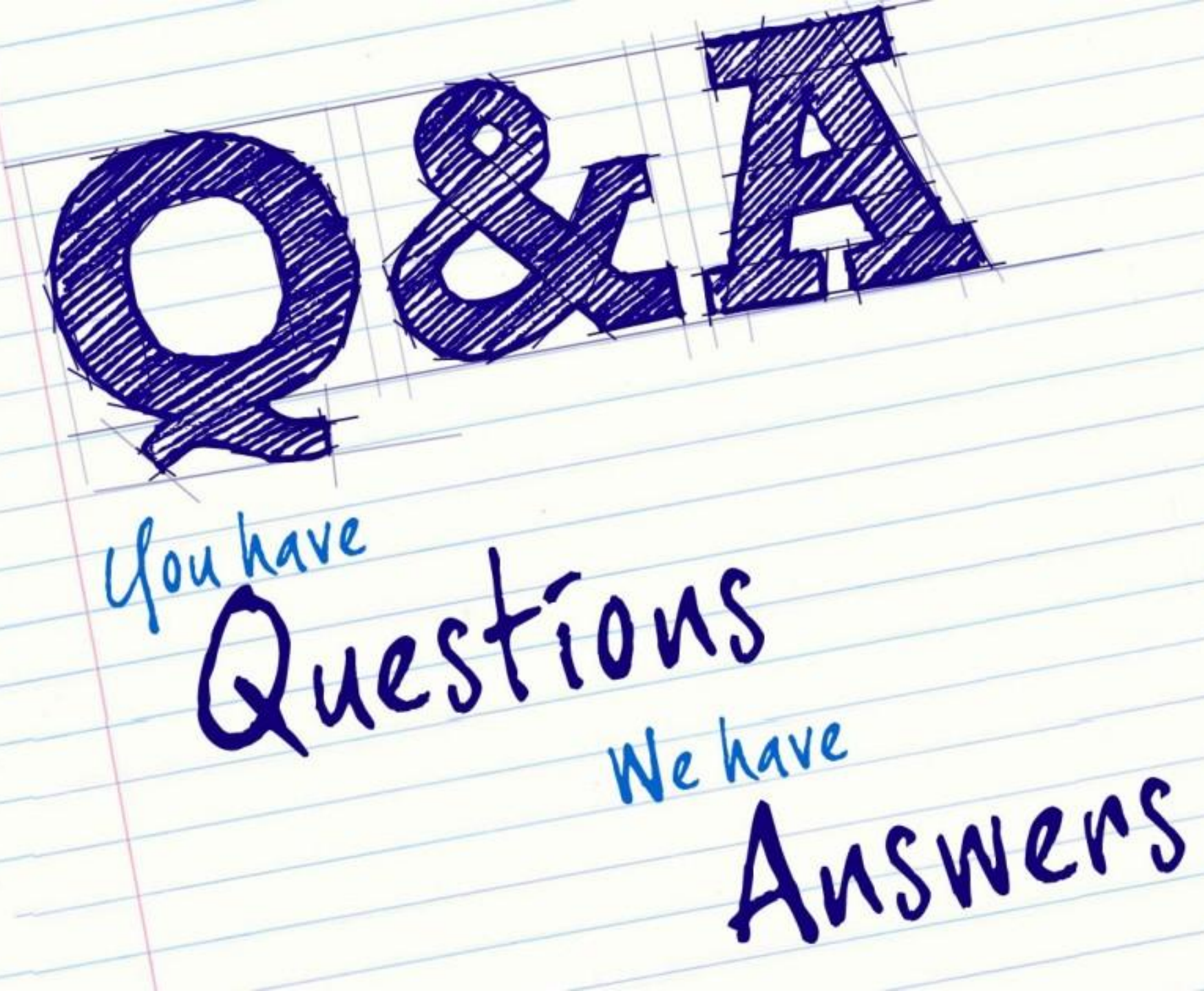
Data Crawler (**Crawling Actor**)

Intelligent Algorithms: **Spark + Hive**
Text Indexing: **Elasticsearch + Kibana**

Reactive **Lambda Architecture for Social Data Processing**

Open Source Technologies and Keywords


Mission of mc2ads:
Create the personal
big data brain for you
http://bigthink.com/praxis/dont-want-to-die-just-upload-your-brain


Q&A
You have
Questions
We have
Answers

More useful content at

http://www.mc2ads.com

"You may say I'm a **dreamer**
But I'm not the only one
I hope **someday** you'll **join us**
And the **world** will **live** as **one**"

John Lennon

Join with us at http://mc2ads.com