

# SmartNews のデータサイエンティストの 高速イテレーションを支える広告システム

SmartNews, Inc.

KOMIYA Atsushi  
Yuyang Lan





KOMIYA Atsushi  
Engineer, SmartNews Ads



Yuyang Lan  
Engineer, SmartNews Ads



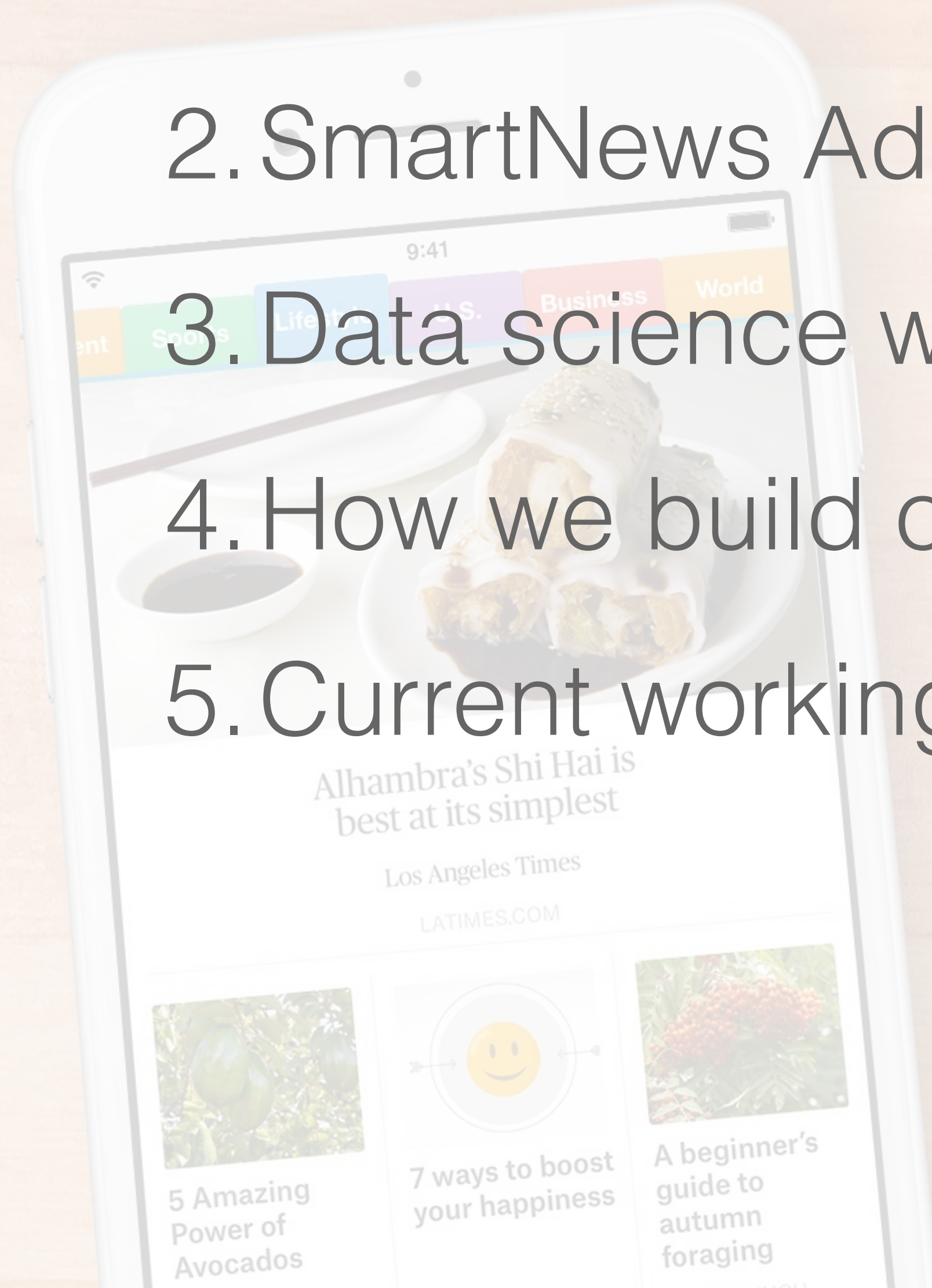
1. Introduction

2. SmartNews Ads and AWS

3. Data science with AWS

4. How we build our ad system with AWS

5. Current working





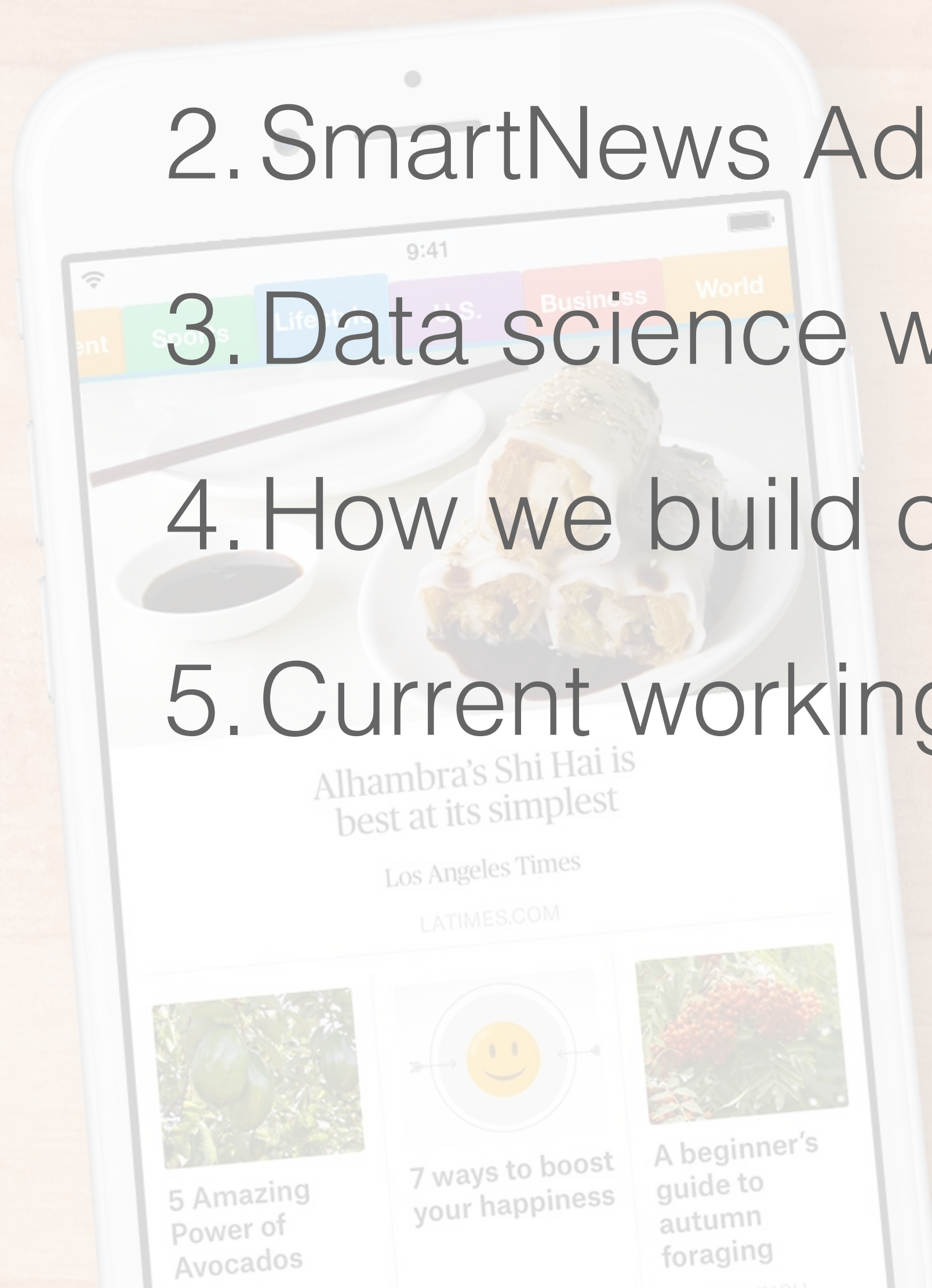
# 1. Introduction

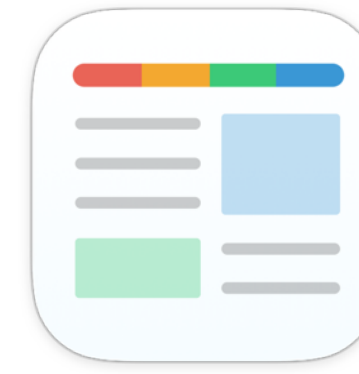
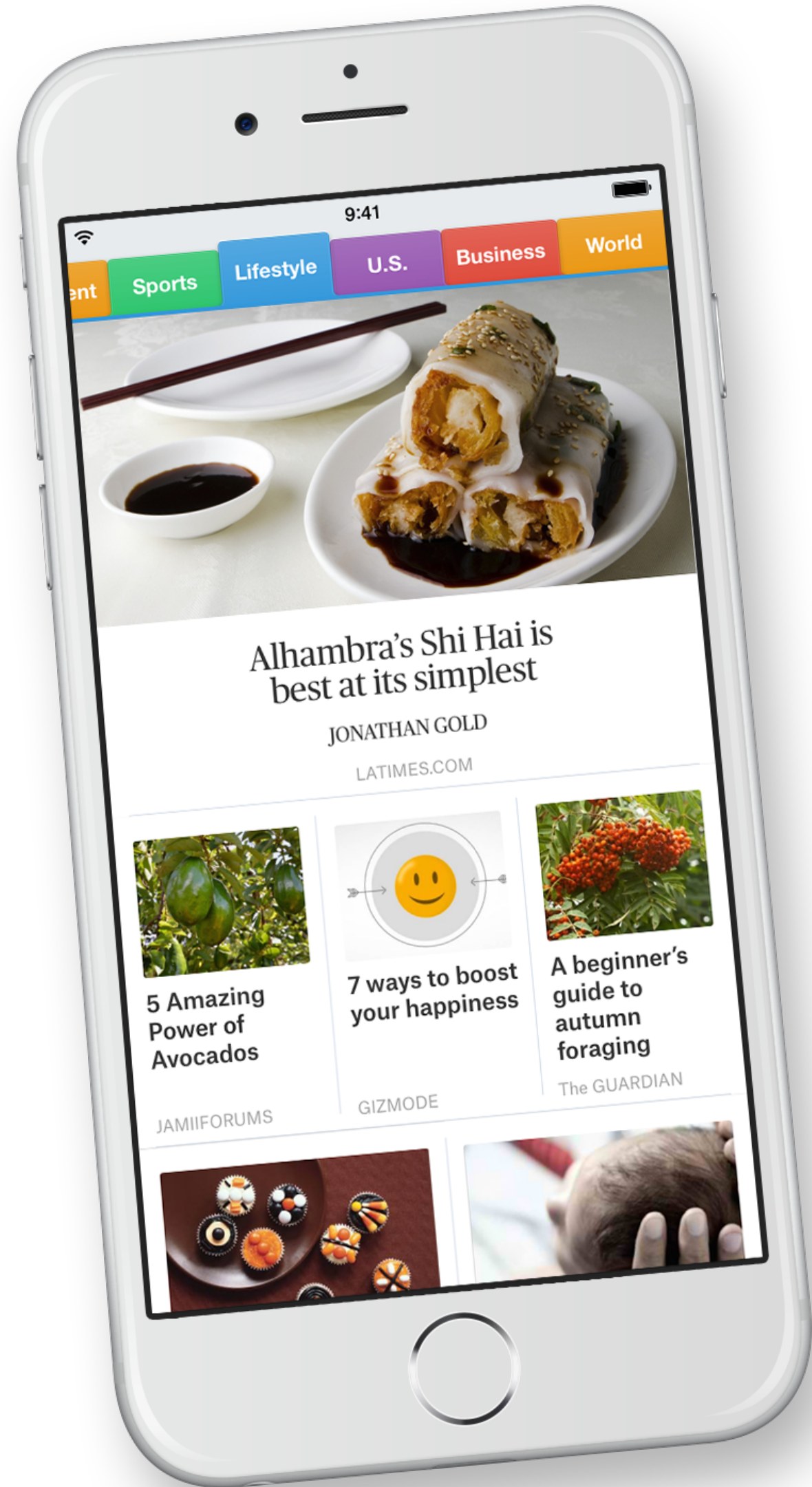
2. SmartNews Ads and AWS

3. Data science with AWS

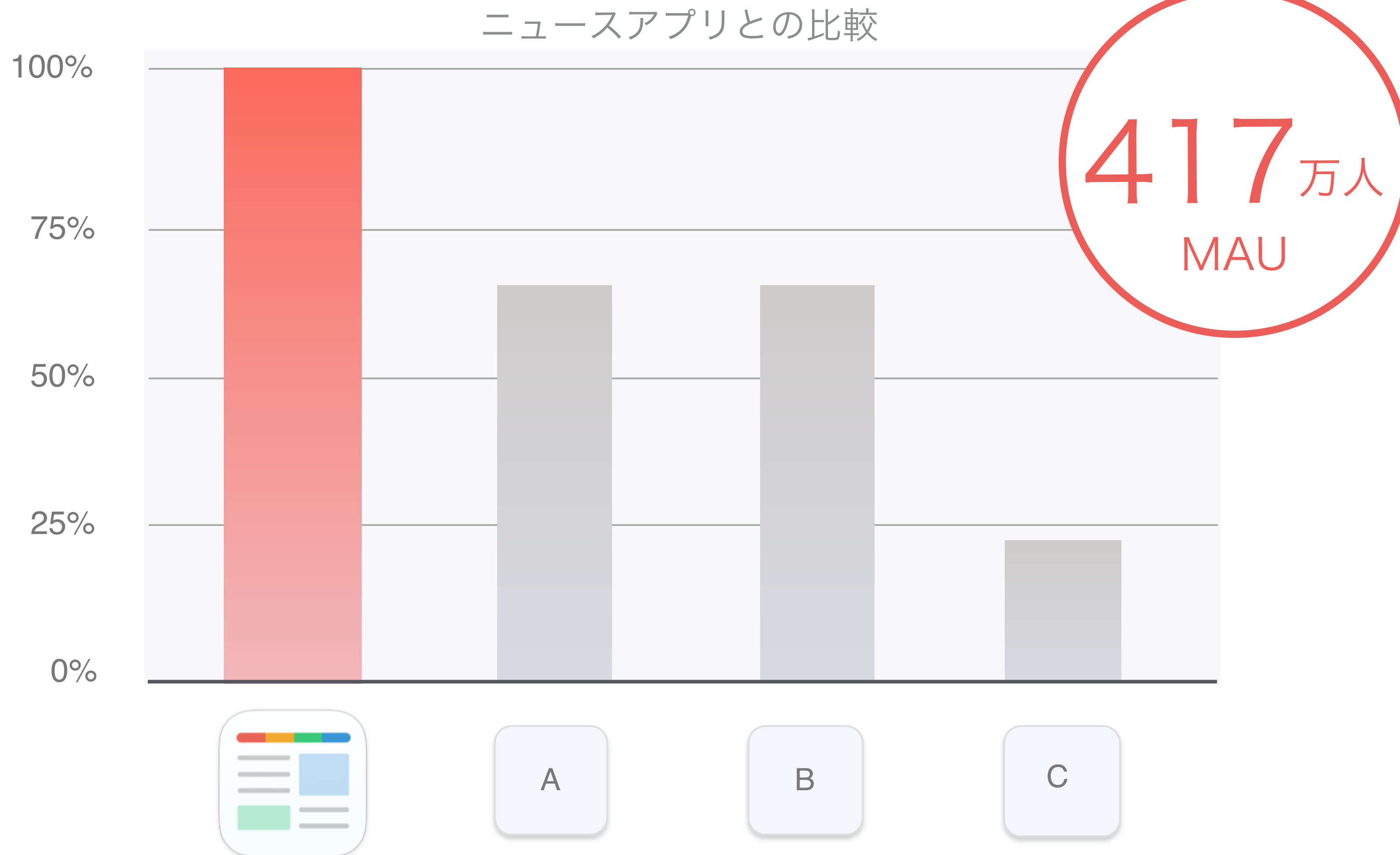
4. How we build our ad system with AWS

5. Current working





SmartNews



✔ 月間のアクティブユーザー数が**417万超**\*で国内No.1

\*Nielsen Mobile NetView 2015年3月



ニュースアプリとの比較\*



1,335万時間  
SPENDS

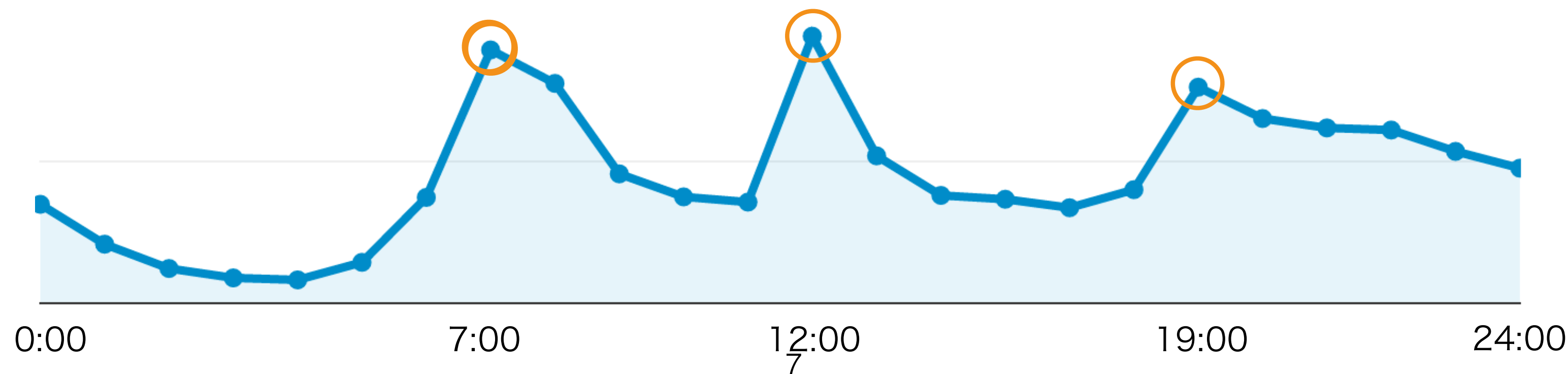


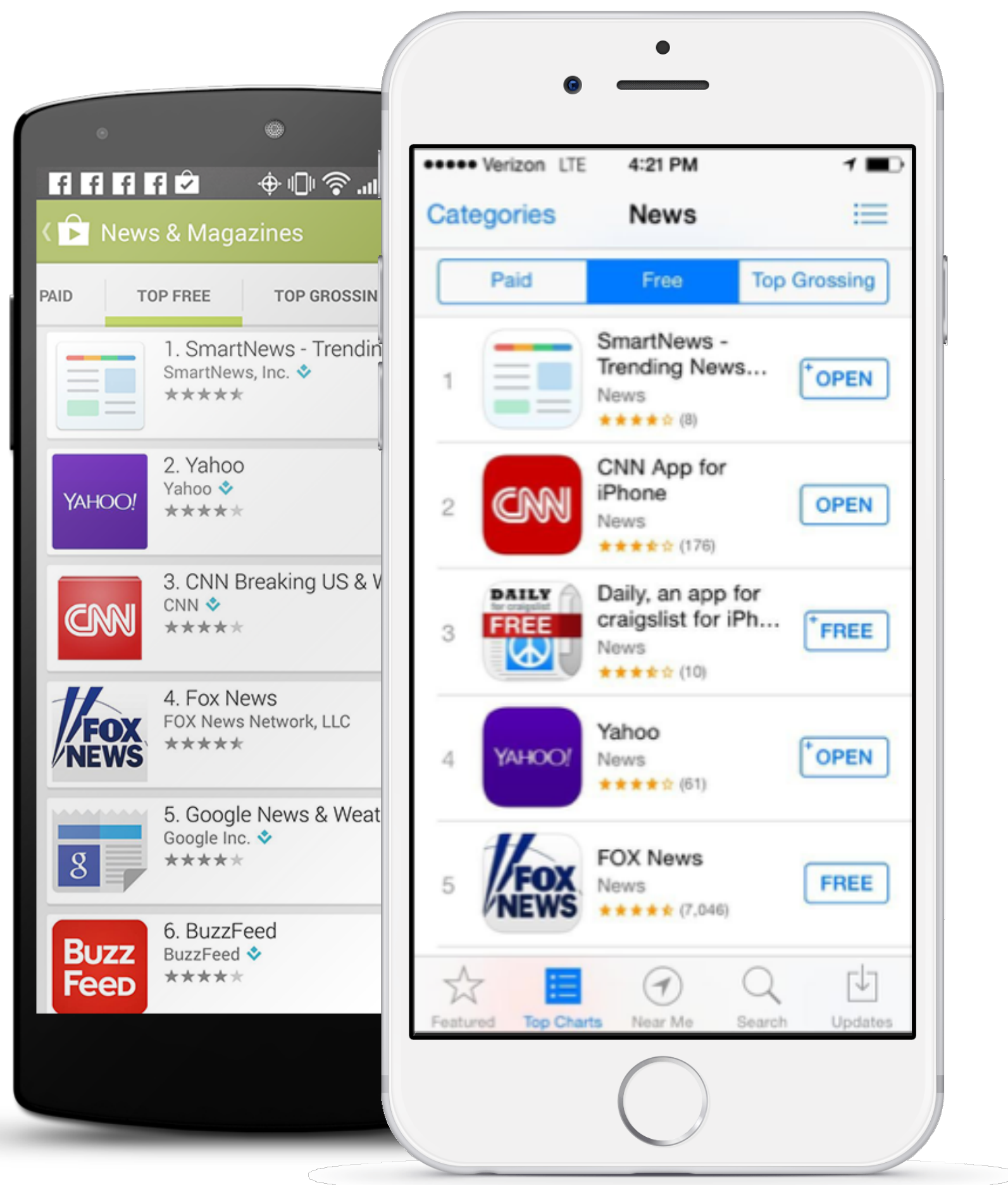
総利用時間  
(千時間)

\*Nielsen Mobile NetView 2015年3月



アクセスのピークは1日3回





2014年10月1日リリース  
米国AppStore/Google Playの2大  
ストアにてニュースアプリでNo.1\*  
を記録



全世界で累計1200万DLを突破！\*\*



\*App Storeより2014年10月6日現在。  
Google Playより2014年11月14日現在。  
SmartNews2.0は2014年10月1日リリース。  
\*\*2015年5月18日リリースより。





10M+ daily

The Internet

Trending URLs

Article Information Extraction

Locale Recognition

Classification

Importance Estimation

1,000+ daily





10M+ daily

The Internet

Trending URLs

Article Information Extraction

We're Machine Learning Company

Classification

Importance Estimation

1,000+ daily





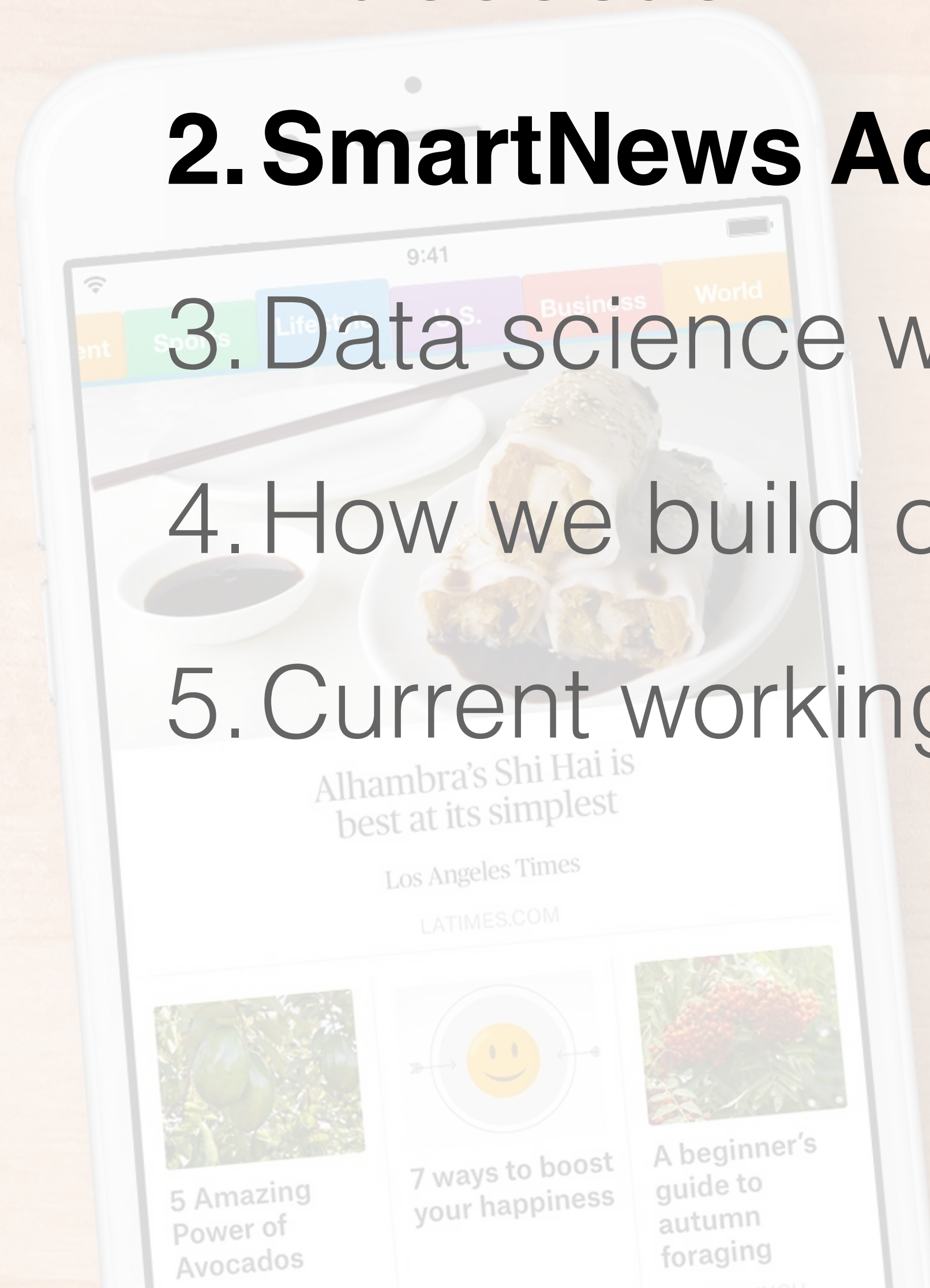
1. Introduction

## 2. SmartNews Ads and AWS

3. Data science with AWS

4. How we build our ad system with AWS

5. Current working





# Standard Ads

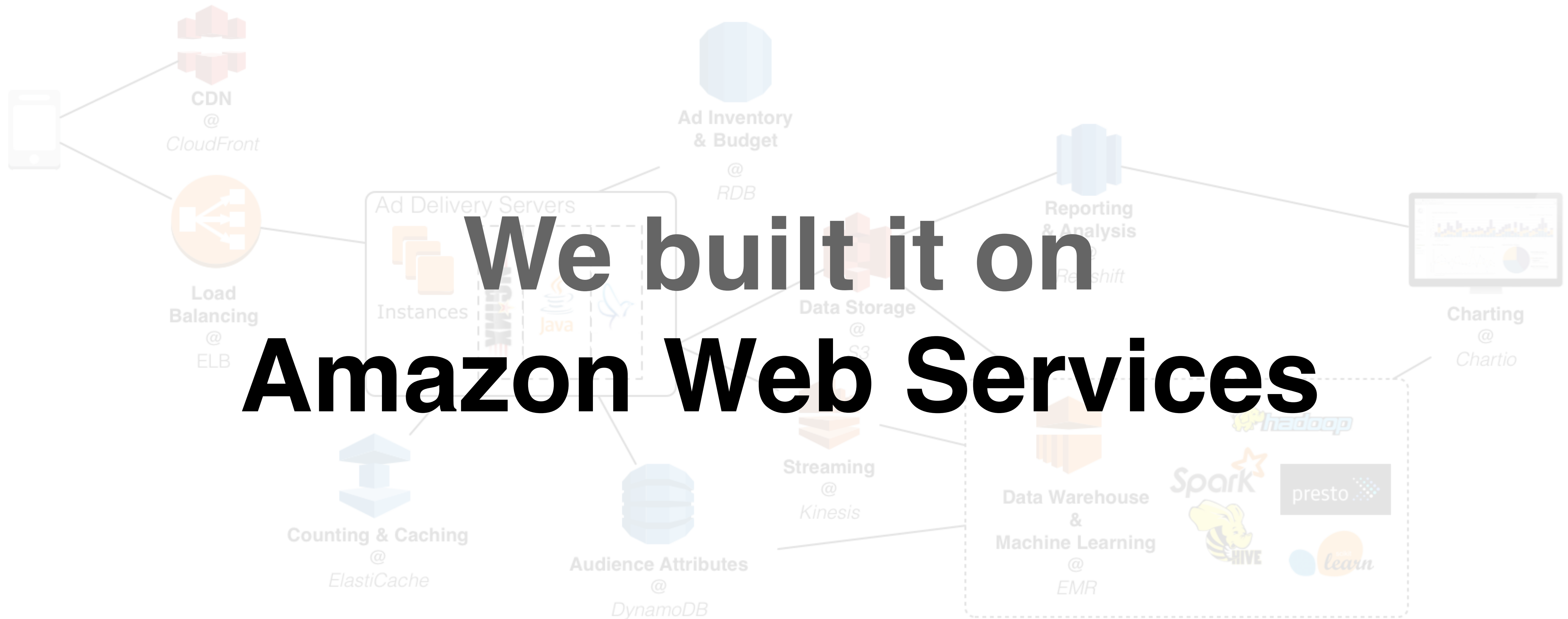


# Premium Movie Ads






- 大量の広告リクエストをごく短時間で処理する Ad server
- 広告のクリエイティブ画像を効率よく配信する仕組み
- ログデータを分析して、ユーザと広告をマッチさせる  
配信ロジック
- 広告主別・メディア別のパフォーマンスを計測・表示する  
レポート機能
- 広告入稿と審査をするための管理コンソール、などなど





インフラ専任エンジニアが一人も居ない  
SmartNews  におけるクラウド活用法

---

#SmartTechNight / SmartNews Tech Night Vol.2

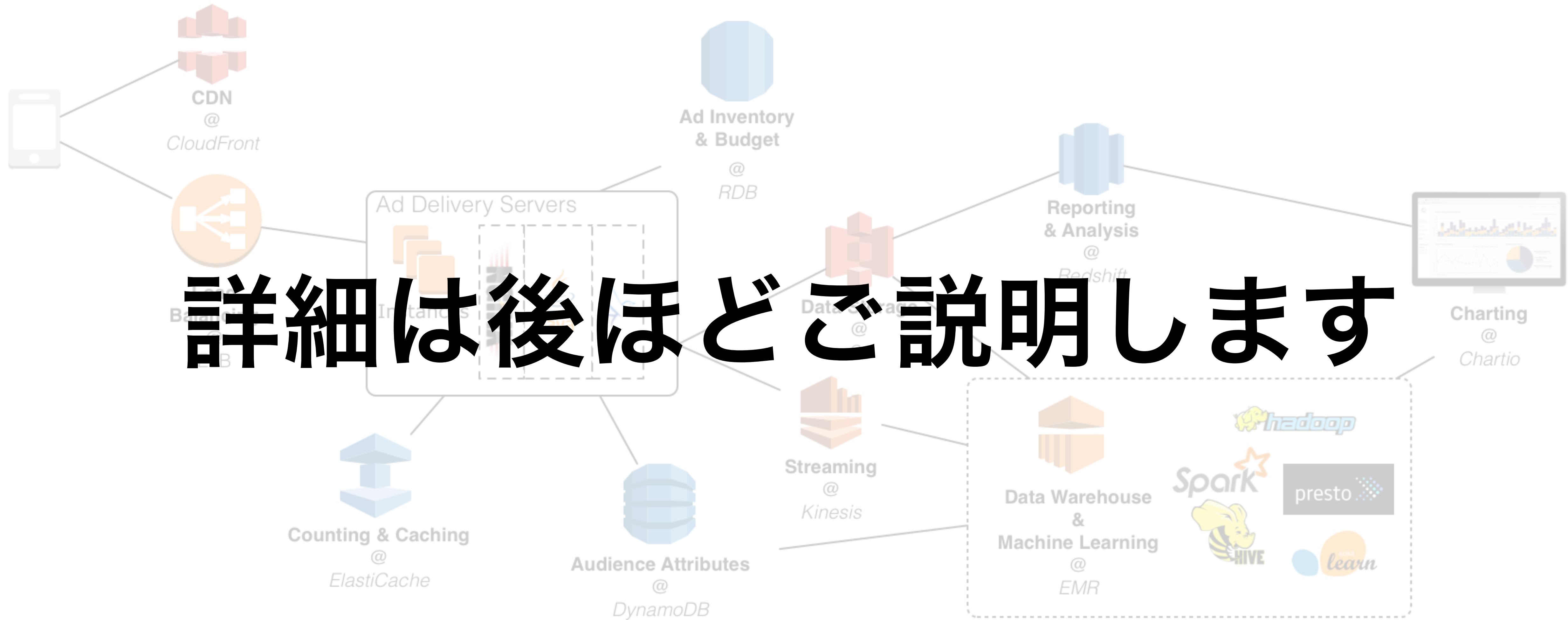
<http://www.slideshare.net/smartnews/20150415-smartnews-technightrev3>



- インフラ**専任**のエンジニアはいない
  - ≠ インフラに詳しいエンジニアがいらない
- 少数**精鋭**の組織
- **本当に大事なところ**に注力したい
- 外部サービス・リソースを有効活用









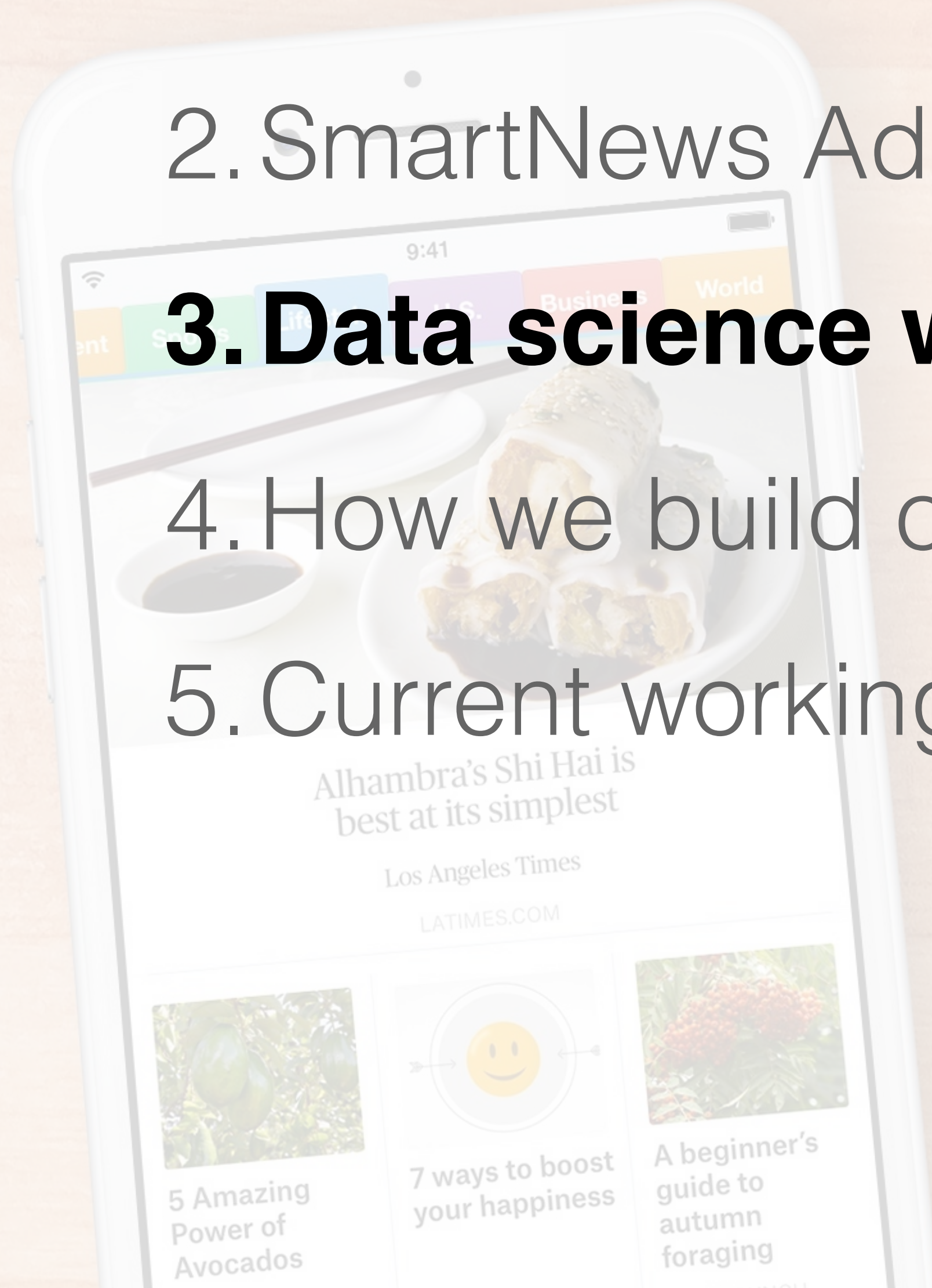
1. Introduction

2. SmartNews Ads and AWS

**3. Data science with AWS**

4. How we build our ad system with AWS

5. Current working





- 広告システムにおいても、様々な**機械学習・最適化タスク**が存在する
  - CTR / CVR prediction
  - Creative optimization
  - CPA optimization
  - Budget smoothing
  - Ad allocation optimization
  - User profiling, etc.
- データサイエンスによる Iterative な改善が求められる



- 広告システムにおいても、様々な機械学習・最適化タスクが存在する
  - CTR / CVR prediction
  - Creative optimization
  - CPA optimization
  - Budget smoothing
  - Ad allocation optimization
  - User profiling, etc.
- Iterative なデータサイエンスによる改善が求められる

## **A/B testing**



## SmartNews 開発者ブログ

---

### SmartNewsの広告システムにおける、データサイエンスへの取り組み方～空気を読まない高速Iteration～

📅 20 5月 2015 ✎ WatabeTakuya 📄 その他

Pocket B! 41 いいね! 95 ツイート 69

SmartNewsで広告プロダクト責任者をやっております、渡部と申します。  
今回はSmartNews Adsのデータサイエンスへの取り組みをご紹介します

詳細は[AWS Summit Tokyo 2015 「SmartNews のデータサイエンティストの高速イテレーションを支える広告システム」](#)にて当社エンジニアが登壇いたしますのでそちらもお楽しみに!

<http://developer.smartnews.com/blog/>



**4+ running tests / day**

**10+ times testing / month**



## 1. 仮説を立てる

- ログデータの分析

## 2. 準備・開発する

- 実験計画・ロジック実装

## 3. A/B テストを実施する

## 4. 結果を集計・統計的に解釈する

- レポーティング・仮説検定

## 5. 意思決定をする



## 1. 仮説を立てる

- ログデータの分析

長期間のログデータをいろんな軸で  
さくさく集計したい

## 2. 準備・開発する

- 実験計画・ロジック実装

## 3. A/B テストを実施する

## 4. 結果を集計・統計的に解釈する

- レポーティング・仮説検定

## 5. 意思決定をする





- **やりたいこと**

- 過去のログを長期間遡って、いろいろな集計・分析をしたい

- **課題**

- ログは日毎に増大していく（それなりの規模）
- 途中でログの項目が増えることもある
- 投げるクエリは目的によって大きく異なる



- **S3 + EMR (+ Presto)** で解決！
  - ログデータの蓄積に S3 を利用する
  - Presto on EMR で S3 上のログデータに対してクエリを投げる
  - ログ項目の追加にも、スキーマ変更は比較的柔軟





## 1. 仮説を立てる

- ログデータの分析

## 2. 準備・開発する

- 実験計画 **ロジック実装**

新しい配信ロジックを production で  
試してみたい

## 3. A/B テストを実施する

## 4. 結果を集計・統計的に解釈する

- レポーティング・仮説検定

## 5. 意思決定をする



- **やりたいこと**

- 広告配信の新しいロジック（予測モデル）を production で試してみたい

- **課題**

- 複雑な配信ロジックほど時間計算量は大きくなる傾向にある
  - 大量のリクエストをごく短時間でさばかなければならない Ad server には重いタスク
  - パフォーマンス改善には相当の時間がかかり、実装も総じて難しい
- 精度が向上するか断言できない機能の準備に、時間をかけ過ぎたくない



- **EMR (+ Hivemall) + DynamoDB** で解決！
  - EMR (と Hivemall) でお手軽に予測モデルを構築
  - 広告の配信を制御するデータを予測モデルから事前計算し、あらかじめ決められたデータ構造で表現 & DynamoDB にストア
  - Ad server から参照 & 広告配信を制御する
  - DynamoDB のレイテンシは十分に小さい (数ms)





- データサイエンティスト的には、
  - 時間・空間計算量的に高性能が求められる Ad server の実装に頭を悩ませる必要がない
  - 自身のタスクである分析・予測モデル構築に集中できる
- リアルタイム性が損なわれるため、精度が低下してしまう欠点がある
- A/B テストによって新しいロジックに優位性があると判断されれば、Ad server に最適化して組み込まれる



## 1. 仮説を立てる

- ログデータの分析

## 2. 準備・開発する

- 実験計画・ロジック実装

## 3. A/B テストを実施する

## 4. 結果を集計・統計的に解釈する

- レポーティング・仮説

A/B テストの状況を手軽に確認したい

## 5. 意思決定をする



- **やりたいこと**
  - 実施中の A/B テストの状況を手軽に確認したい
- **課題**
  - 都度 SQL を用意して集計 → Excel でレポートニング、  
というのは手間がかかり過ぎる
  - レポート閲覧用の Web アプリケーションを作るのも  
開発&メンテナンスコストがかかる





- **Redshift (+ Chartio)** で解決！
  - A/B テストすべてにおいて、共通の集計軸とメトリクス（複数）を利用する
  - 毎時に集計バッチを実行して、ログデータから A/B テストのレポートを Redshift 上で集計・作成する
  - A/B テストの結果を Redshift から取得して表示する Chartio ダッシュボードを用意する
  - 集計軸とメトリクスが同じなので、異なる A/B テストでも同じダッシュボードを再利用できる





- SmartNews の広告システムは AWS を活用して作られている
- 本当に大事なところ（事業）に集中するため、マネージドサービスを積極的に活用する
- データサイエンスの高速な iteration を実現するために、AWS の各種サービスのよいとこどりをしている



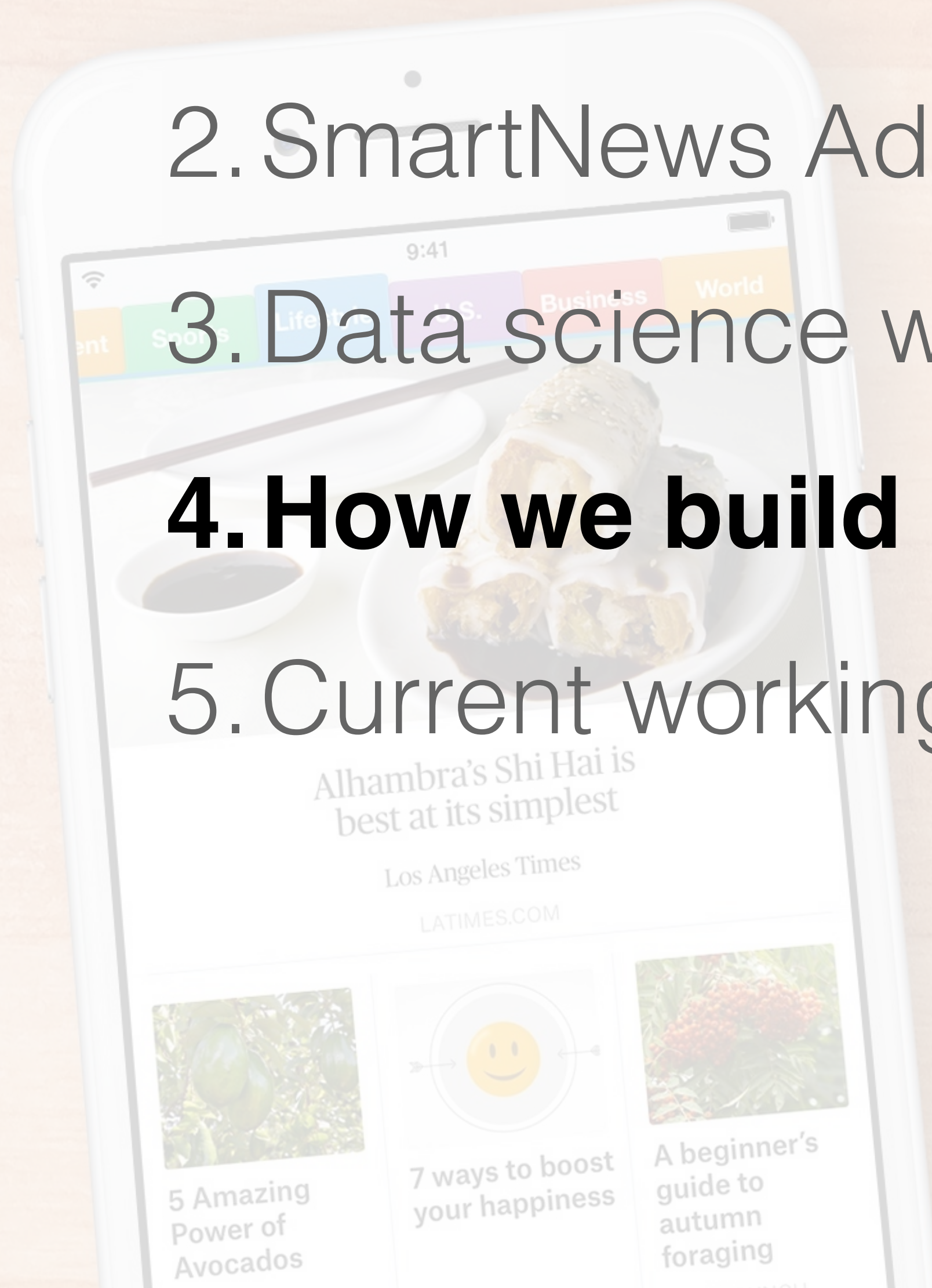
1. Introduction

2. SmartNews Ads and AWS

3. Data science with AWS

**4. How we build our ad system with AWS**

5. Current working

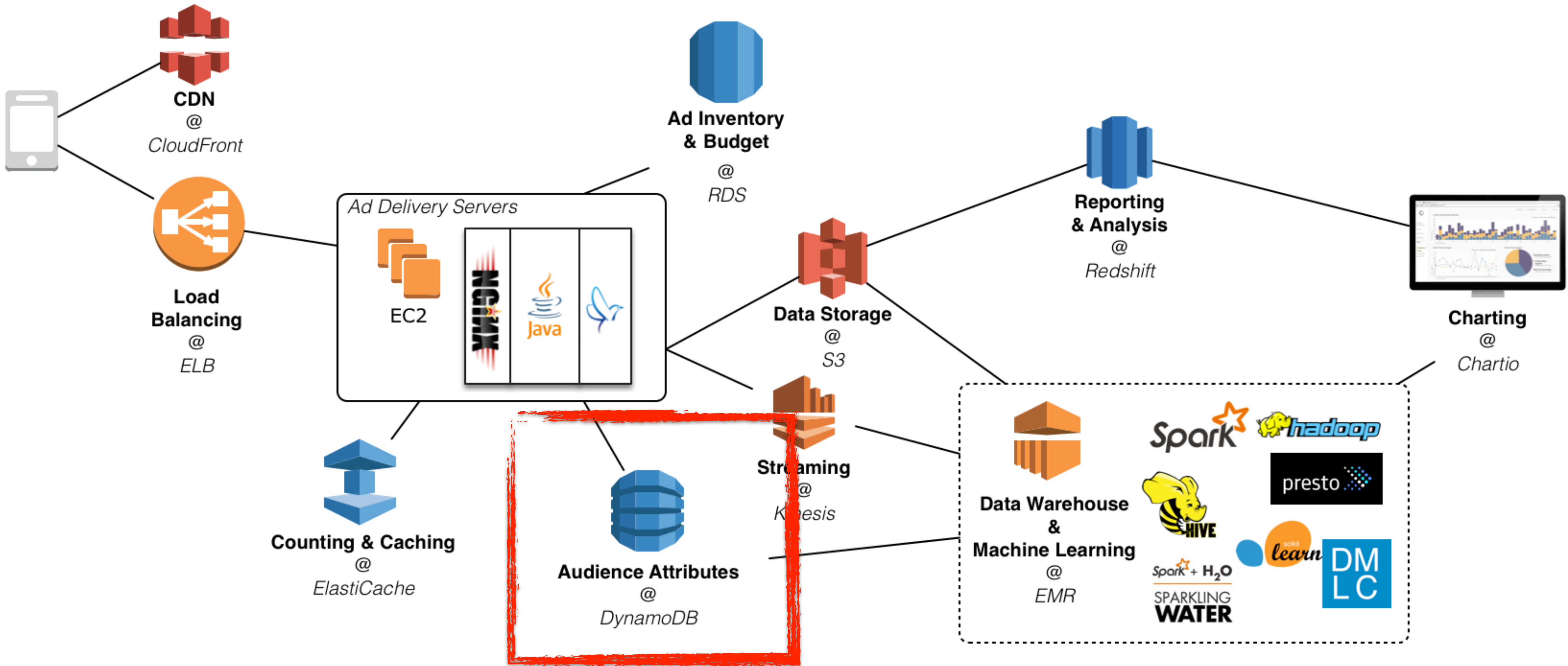




# **When We Make Tech Choices**

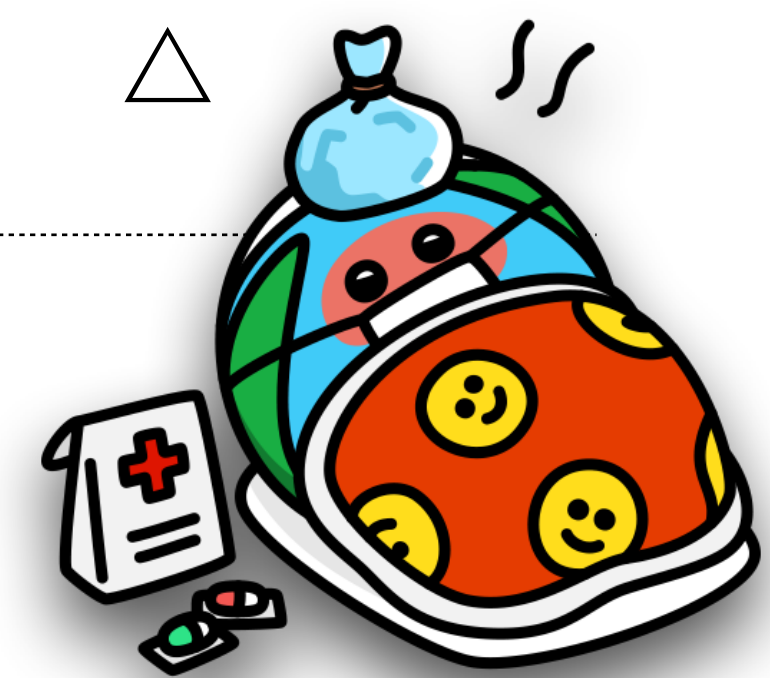


- Speed (*50ms MAX latency*)
- Availability
- Scalability
- Operationally **“Raku”**
- Decouple-able
- People-Friendly (*Engineers, Data Scientists, ..*)





	Speed	Avalaibility	Scalability	People Friendly	Decouple-able	Operationally "Raku"
<b>HBase</b>	★	△	★	○	○	△
<b>Redis</b>	★	△	△	★	★	○
<b>Memcached</b>	★	○	△	○	★	○
<b>Cassandra</b>	★	★	★	★	★	△





	<b>Speed</b>	<b>Avalaibility</b>	<b>Scalability</b>	<b>People Friendly</b>	<b>Decouple-able</b>	<b>Operationally “Raku”</b>
<b>HBase</b>	★	△	★	○	○	△
<b>Redis</b>	★	△	△	★	★	○
<b>Memcached</b>	★	○	△	○	★	○
<b>Cassandra</b>	★	★	★	★	★	△
<b>DynamoDB</b>	★	★	★	★	★	★





**No Single Engineer Needed To Operate It**



# OPERATION COST

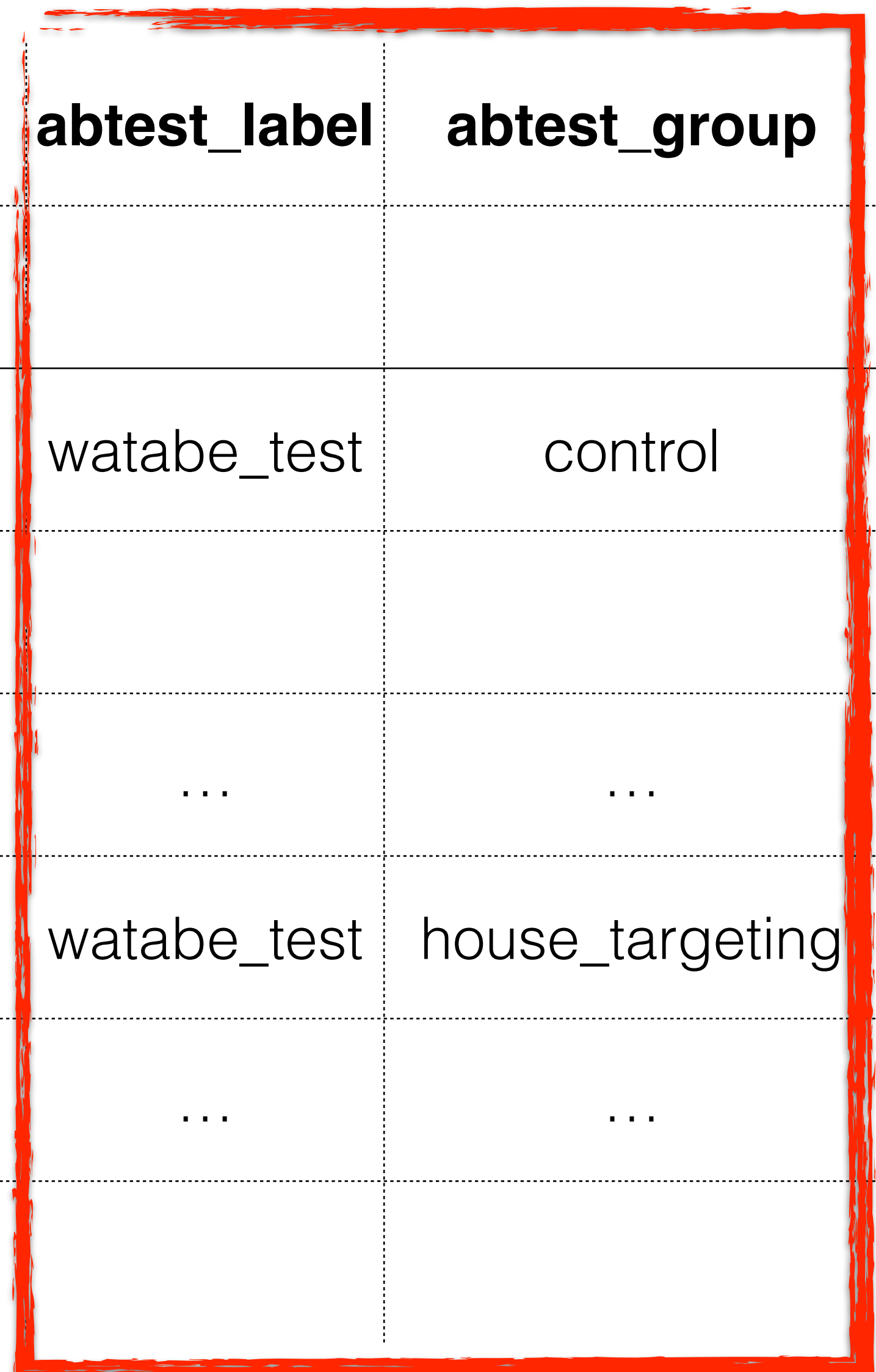
**\$0**

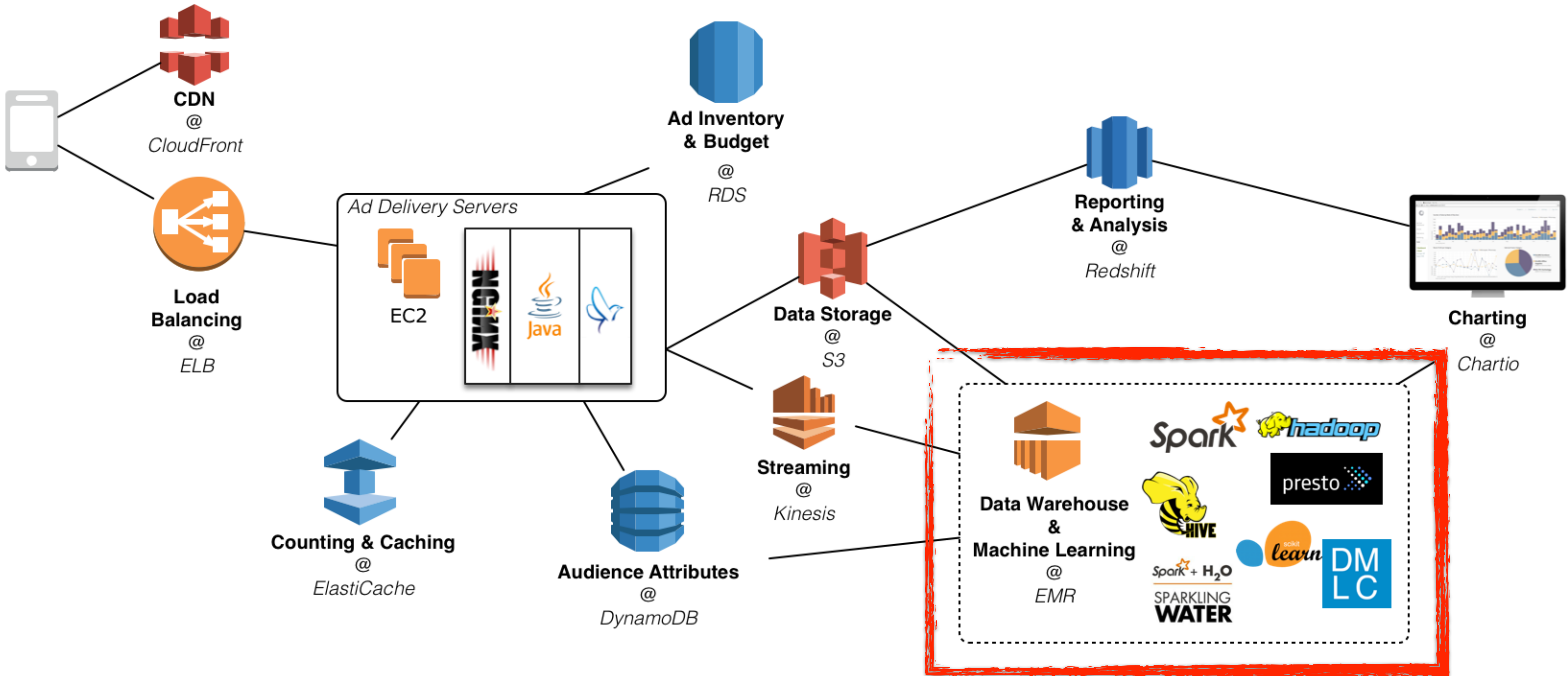






<b>device_id</b>	<b>media</b>	<b>attr_1</b>	<b>attr_2</b>	<b>scores</b>	<b>tags</b>	<b>...</b>	<b>abtest_label</b>	<b>abtest_group</b>	<b>...</b>
<i>Hash Key</i>	<i>Range Key</i>			<i>map&lt;campaign, score&gt; score</i>	<i>set &lt;tag&gt;</i>				
d101	SN	B		{123=>3.5}	...		watabe_test	control	
d102	APP_2		2.5						
d102	SN	A	...	...	{akb48}		...	...	
d302	SN	...		...	{car, hourse}		watabe_test	house_targeting	
d888	SN	...		...	...		...	...	
...									





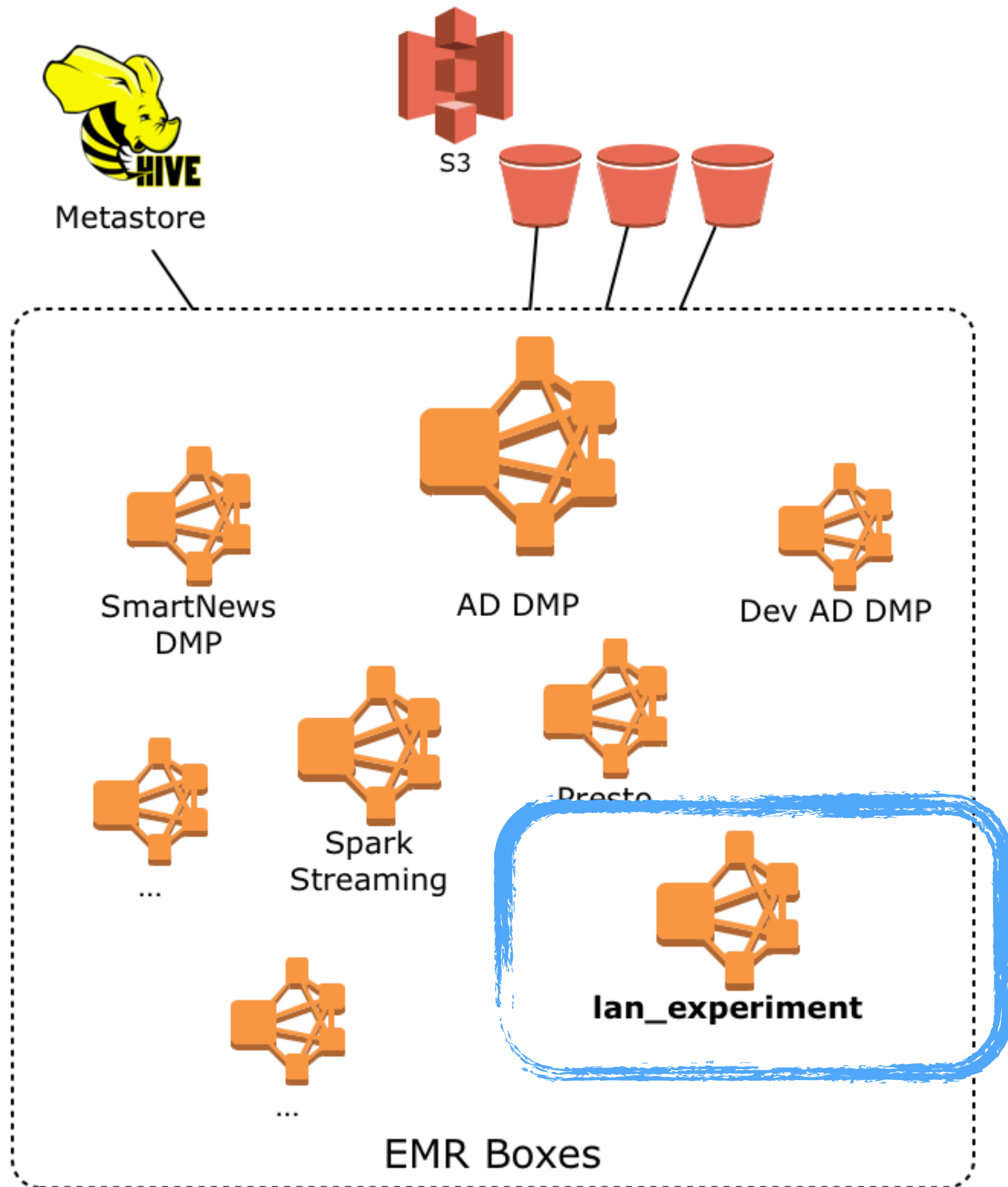


- Yarn + Hive + Spark + Python\* + *alpha*
- All data managed by Hive, stored in S3
- Store no states inside the cluster
- HDFS is used as a tmp/cache layer (*dfs.replication = 1*)
- Use it elastically
  - Scale-out for large jobs
  - Multiple small clusters for difference purposes



# **When Need Do Experimental • Ad-hoc Data Work (e.g. New Machine Learning Experiment, ..)**





```
$ ./start_new_cluster.sh lan_experiment \  
  --template ad_hoc \  
  --cores 10:c3.8xlarge:0.4
```

Provisioning ...

...

Starting ...

{

**"ClusterId": "j-99ABC12305QT"**

}

...

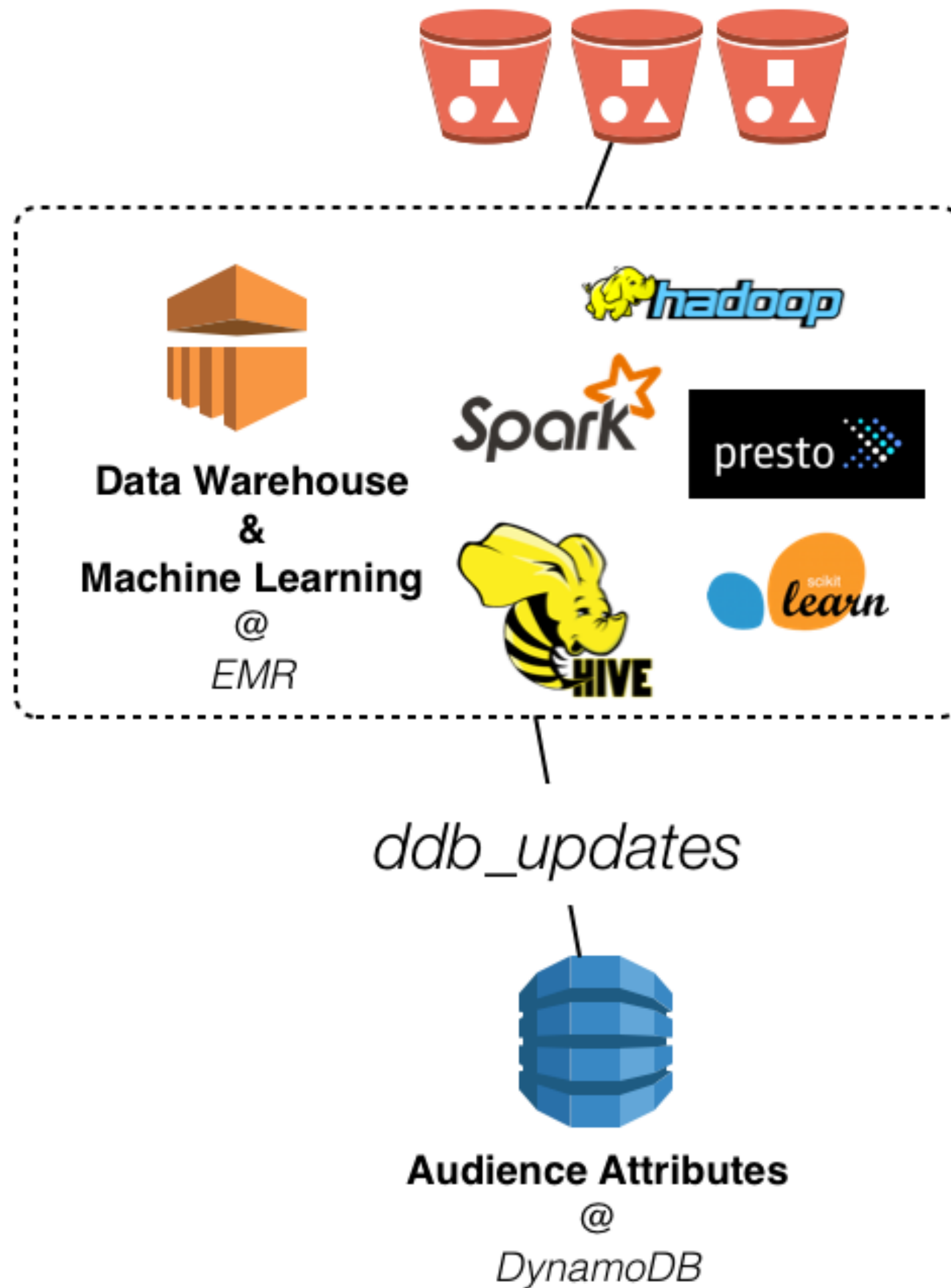
Done !

Master is:

ec2-123-45-67-89.ap-northeast-1.compute.amazonaws.com

Dashboard Link is:

<https://ap-northeast-1.console.aws.amazon.com/elastic>



```
set ddb_update.table.name=prod_smartad_dmp_audience;
set mapred.reduce.tasks=25;
```

```
-- set ddb_update.write.throughput=2500;
set ddb_update.write.throughput.percentage=0.98;
```

```
select sum(write) from (
select
ddb_update(device_id,
'scores_my_test', scores,
'scores_my_test_ts', now()
) as write
```

```
from (
select *
from upload.scoring_my_test
cluster by rand()
) data
) t;
```



y-lan / **monkey-spanner**

Unwatch 4

Star 8

Fork 6

branch: master



monkey-spanner / src / main / java / spanner / monkey / hive / **GenericUDFDynamodbUpdate.java**

y-lan on Feb 23 updated log in ddb\_update

1 contributor

222 lines (188 sloc) | 9.697 kb

Raw Blame History

```

1 package spanner.monkey.hive;
2

```





Sometimes you even don't need to distribute

SmartNews

```
top - 16:14:29 up 1:15, 2 users, load average: 21.36, 19.05, 11.40
Tasks: 289 total, 21 running, 268 sleeping,
Cpu0 : 60.0%us, 1.3%sy, 0.0%ni, 38.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1 : 49.7%us, 2.7%sy, 0.0%ni, 47.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2 : 46.0%us, 1.7%sy, 0.0%ni, 52.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3 : 32.9%us, 0.0%sy, 0.0%ni, 67.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu4 : 38.2%us, 0.3%sy, 0.0%ni, 61.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu5 : 34.8%us, 0.7%sy, 0.0%ni, 64.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu6 : 32.6%us, 0.3%sy, 0.0%ni, 67.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu7 : 42.5%us, 0.3%sy, 0.0%ni, 57.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu8 : 54.5%us, 2.0%sy, 0.0%ni, 43.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
```

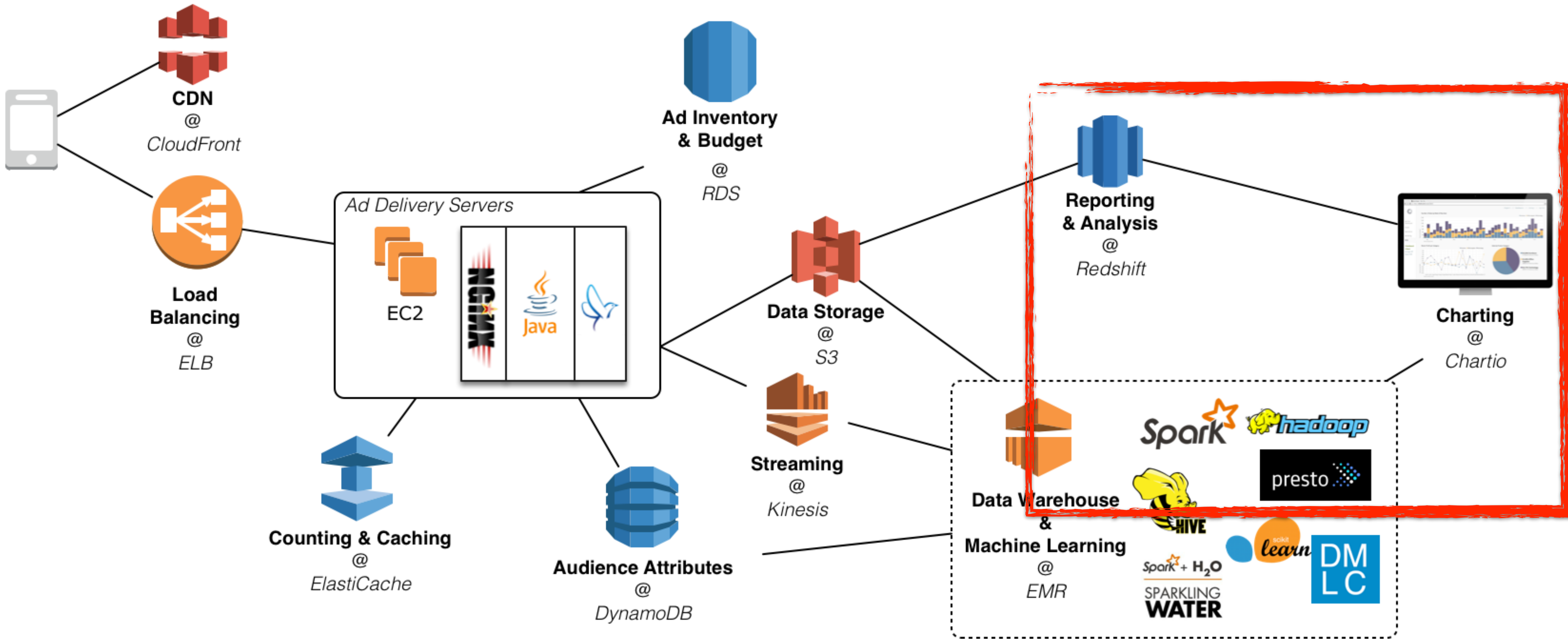
Model	vCPU	Mem (GiB)	SSD Storage (GB)

# COST

# \$1.0

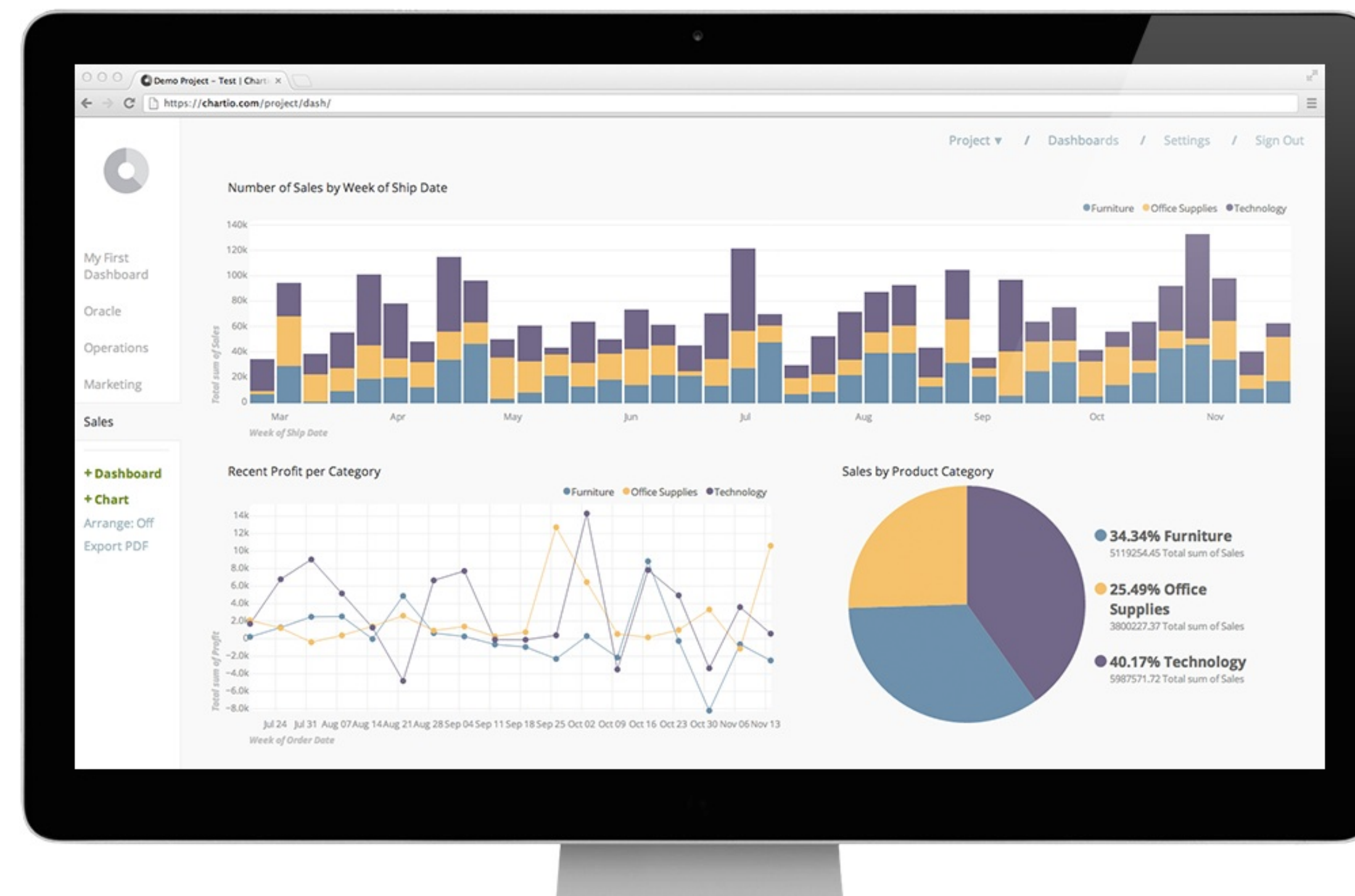
```
Cpu22 : 75.5%us, 24.7%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu23 : 100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu24 : 67.9%us, 32.1%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu25 : 100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu26 : 74.0%us, 26.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu27 : 90.0%us, 10.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu28 : 98.3%us, 1.7%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu29 : 98.7%us, 1.3%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu30 : 100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu31 : 78.7%us, 16.6%sy, 0.0%ni, 4.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 251913536k total, 115126296k used, 136787240k free, 59204k buffers
Swap: 0k total, 0k used, 0k free, 0k cached
```

```
parallel --jobs 20 \
./predict.py ./experiment.model
```





- Hourly reporting in **Redshift**
- Query raw log in S3 through **Presto**
- Query realtime data in Kinesis through **Spark Streaming**
- Charting using **Chartio**





	<b>Redshift</b>	<b>Presto</b>
Purpose	Business Report	General
<b>Data</b>	<b>Core Structured Data</b>	<b>Data in S3, Tables in RDS, etc.</b>
Stable	Good	OOM sometimes Occasionally down is OK
Performance	Good	Good but high variance
User	More Business side	More Engineering side
<b>Extensible</b>	<b>No</b>	<b>Custom Patch , Functions, Connectors</b>



facebook / presto

Unwatch 454

Unstar 3,451

Fork 803

# Fix InvalidRange error and SocketTimeoutException in PrestoS3FileSystem #2647

Edit

**Closed** y-lan wants to merge 2 commits into facebook:master from smartnews:feature/fix\_s3fs

Conversation 9

Commits 2

Files changed 1

+23 -8



y-lan commented on Apr 7

This PR fixes the following errors which may happen when querying data in S3 and cause query to fail.

- When calling `S3.getObject(...).withRange(start, end)`, if `start` exceeds the length of object, AWS will return an `InvalidRange` error with status code `416`. This shouldn't cause the query to fail.
- When calling `skip()` on S3 inputstream, sometimes we may get `SocketTimeoutException` or other I/O exception. This currently cause the whole query to fail, but we can recovery it by reopening the inputstream.

### Labels

None yet

### Milestone

No milestone

### Assignee

electrum







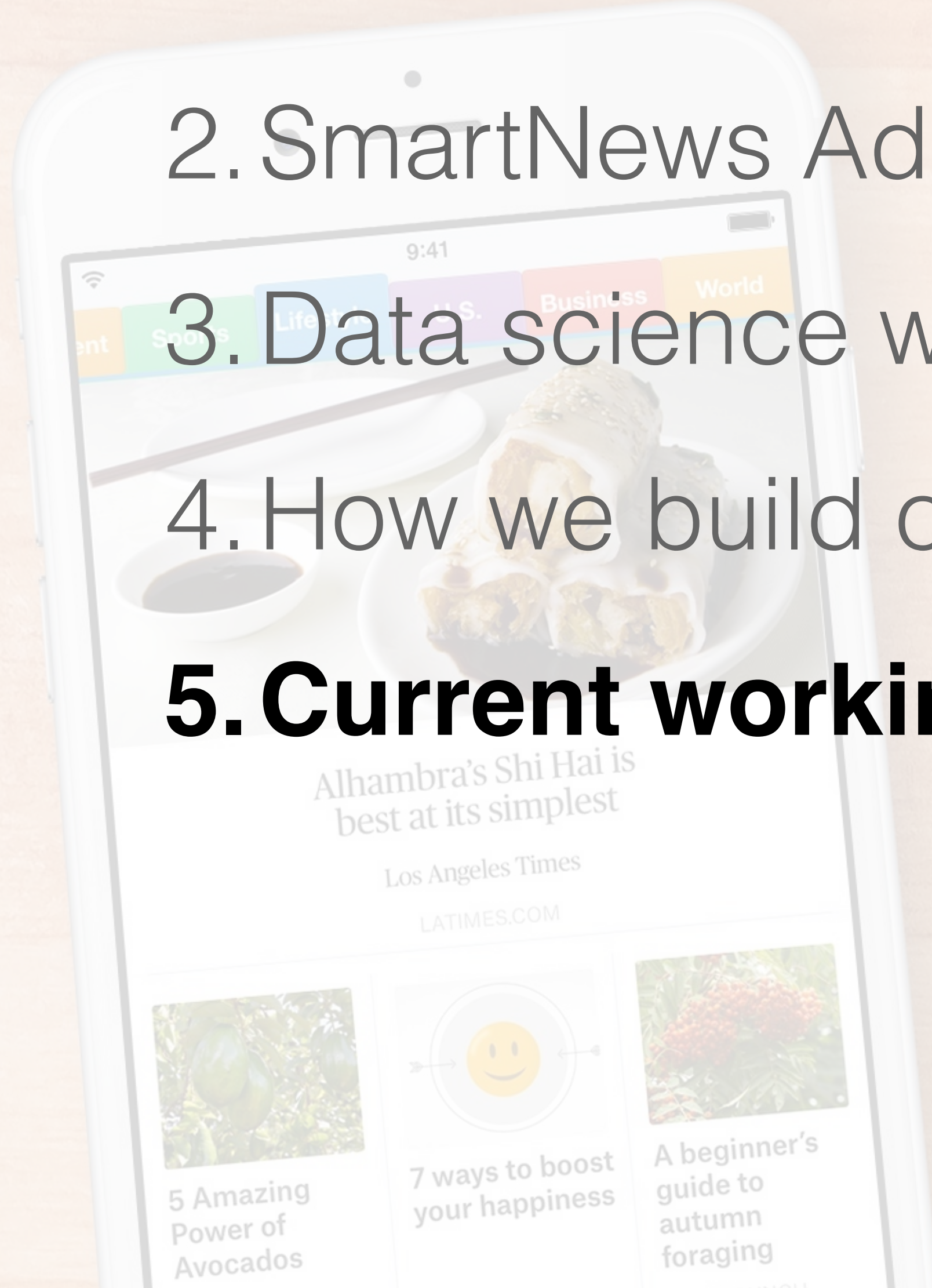
1. Introduction

2. SmartNews Ads and AWS

3. Data science with AWS

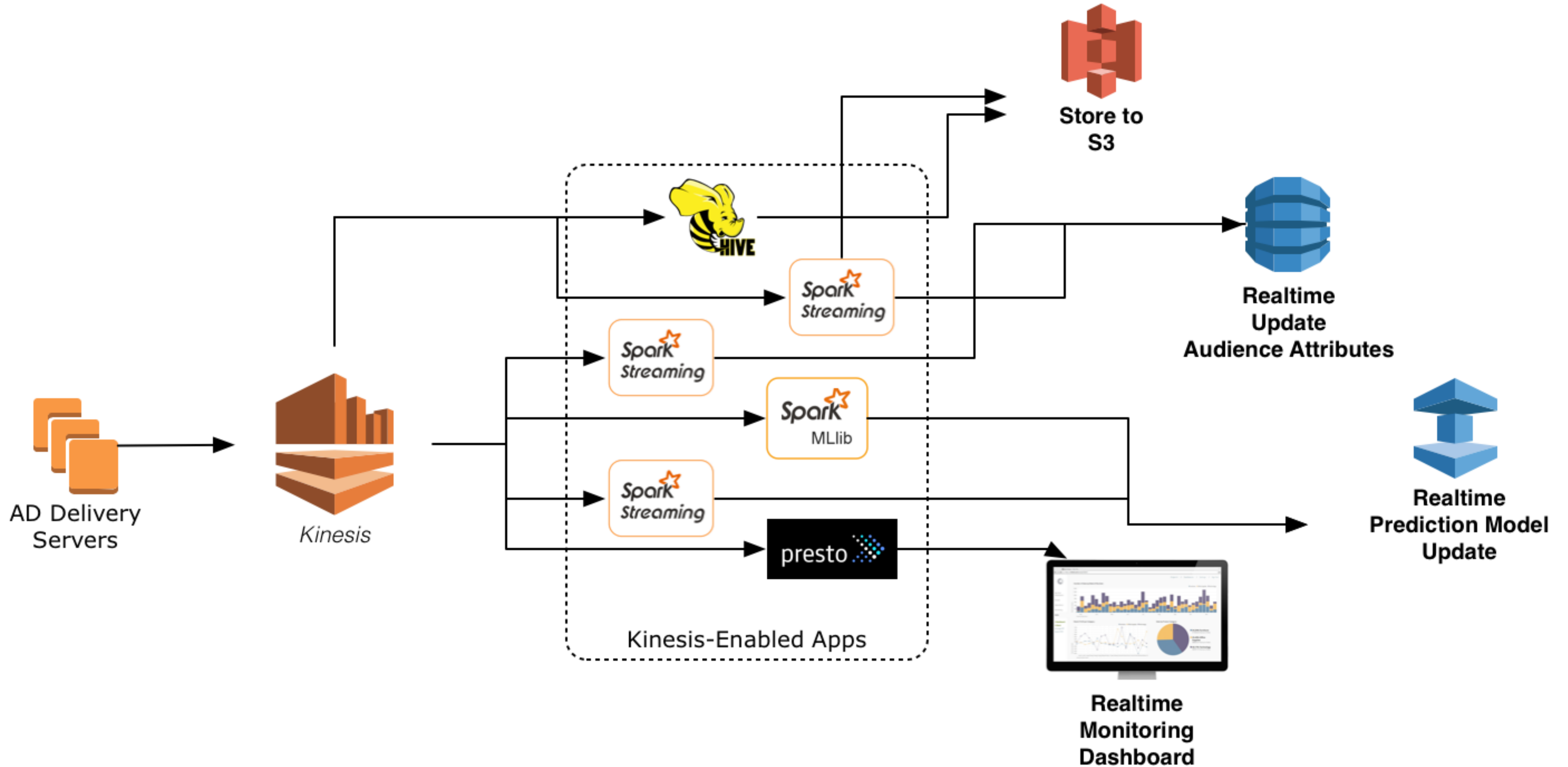
4. How we build our ad system with AWS

**5. Current working**





# **Kinesis**-enabled System





# The Last

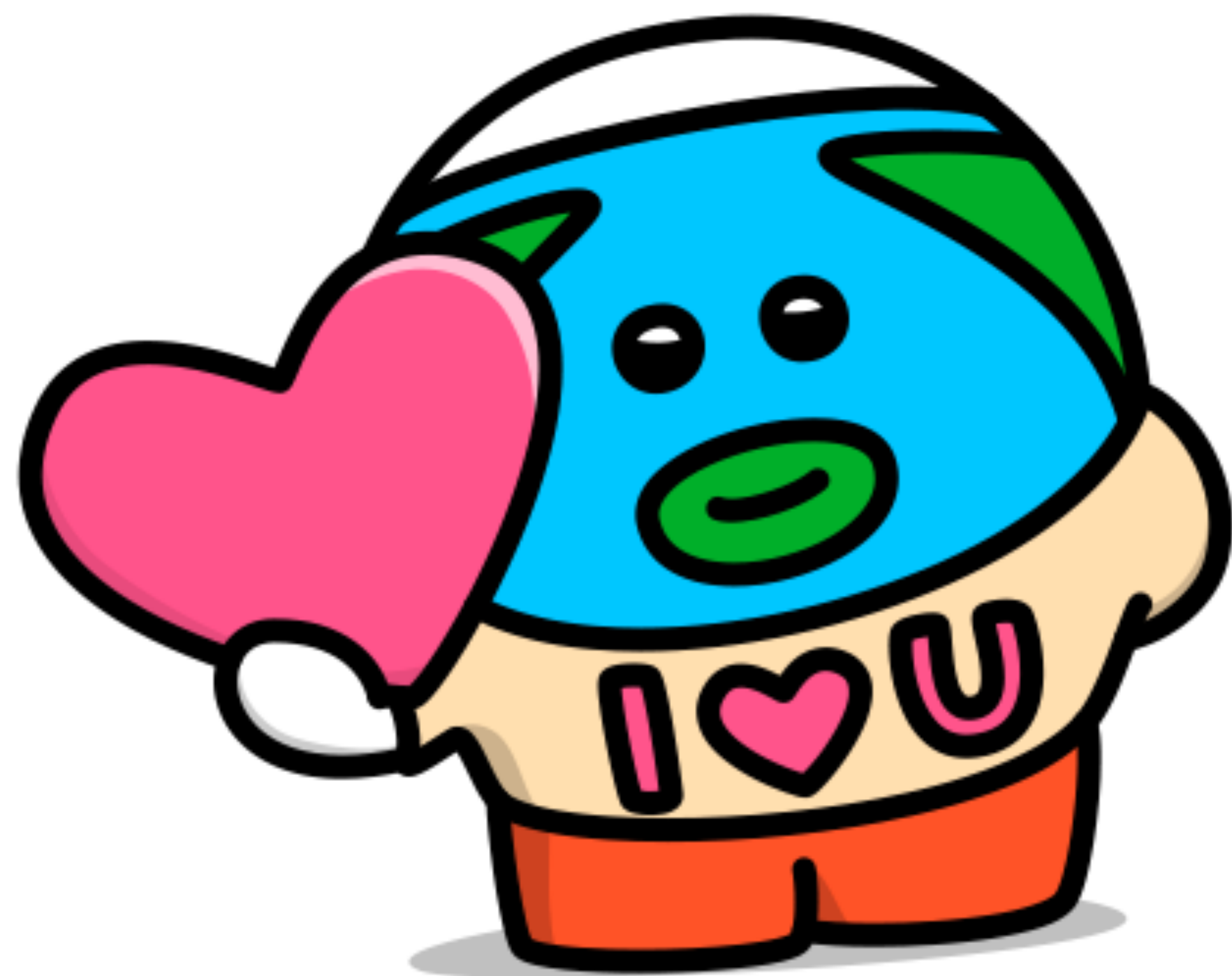


- Lambda Available to Tokyo Region
- Better way to control DynamoDB write capacity
- Time-to-live support in DynamoDB
- UDF in Redshift



Thanks!

**We're hiring!**



iOSエンジニア / Androidエンジニア  
/ Webアプリケーションエンジニア  
/ プロダクティビティエンジニア  
/ 機械学習 / 自然言語処理エンジニア  
/ グローブハックエンジニア  
/ サーバサイドエンジニア  
/ 広告エンジニア...

<http://about.smartnews.com/ja/careers/>