I.P.A.D.S

Innovative Precision Agriculture Decision-support System

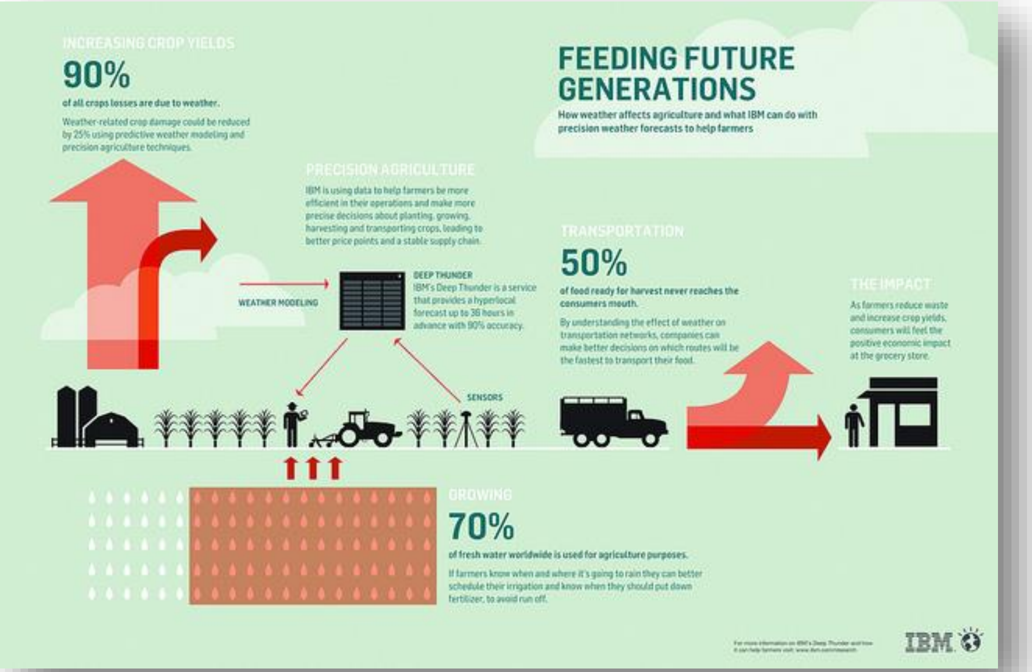
Group 2: Yue Li, Mingxuan Li, Xingdi Li

Agenda

- Problem Definition
- Design
- Implementation
- Demo

Why Agriculture?

- Sustainable pressure
  - Population increase
  - Climate change
  - Economic change
- Precision-Ag motivation
  - Cost of human labor growth
  - Quick response to climate
  - Tech is the king in modern Agriculture



FEEDING FUTURE GENERATIONS

How weather affects agriculture and what IBM can do with precision weather forecasts to help farmers

INCREASING CROP YIELDS

90%

of all crops losses are due to weather.

Weather-related crop damage could be reduced by 25% using predictive weather modeling and precision agriculture techniques.

PRECISION AGRICULTURE

IBM is using data to help farmers be more efficient in their operations and make more precise decisions about planting, growing, harvesting and transporting crops, leading to better price points and a stable supply chain.

DEEP THUNDER

IBM's Deep Thunder is a service that provides a hyperlocal forecast up to 38 hours in advance with 90% accuracy.

TRANSPORTATION

50%

of food ready for harvest never reaches the consumers mouth.

By understanding the effect of weather on transportation networks, companies can make better decisions on which routes will be the fastest to transport their food.

THE IMPACT

As farmers reduce waste and increase crop yields, consumers will feel the positive economic impact at the grocery store.

GROWING

70%

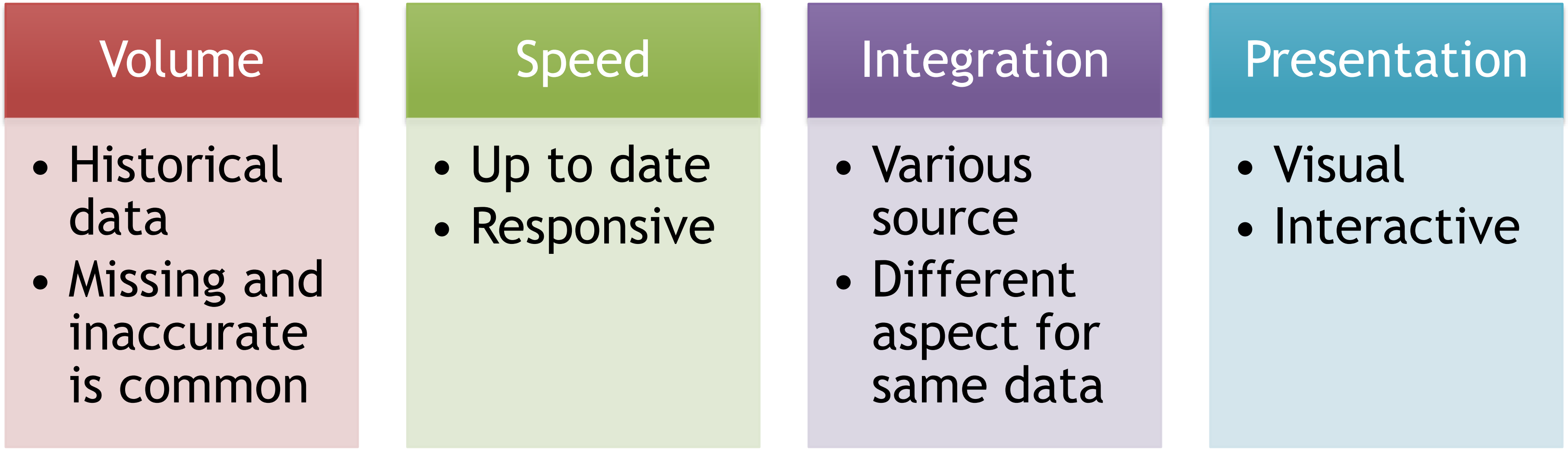
of fresh water worldwide is used for agriculture purposes.

If farmers know when and where it's going to rain they can better schedule their irrigation and know when they should put down fertilizer, to avoid run off.

For more information on IBM's Deep Thunder and how it can help farmers visit: www.ibm.com/research

Source: IBM

Challenge of getting information



Volume

- Historical data
- Missing and inaccurate is common

Speed

- Up to date
- Responsive

Integration

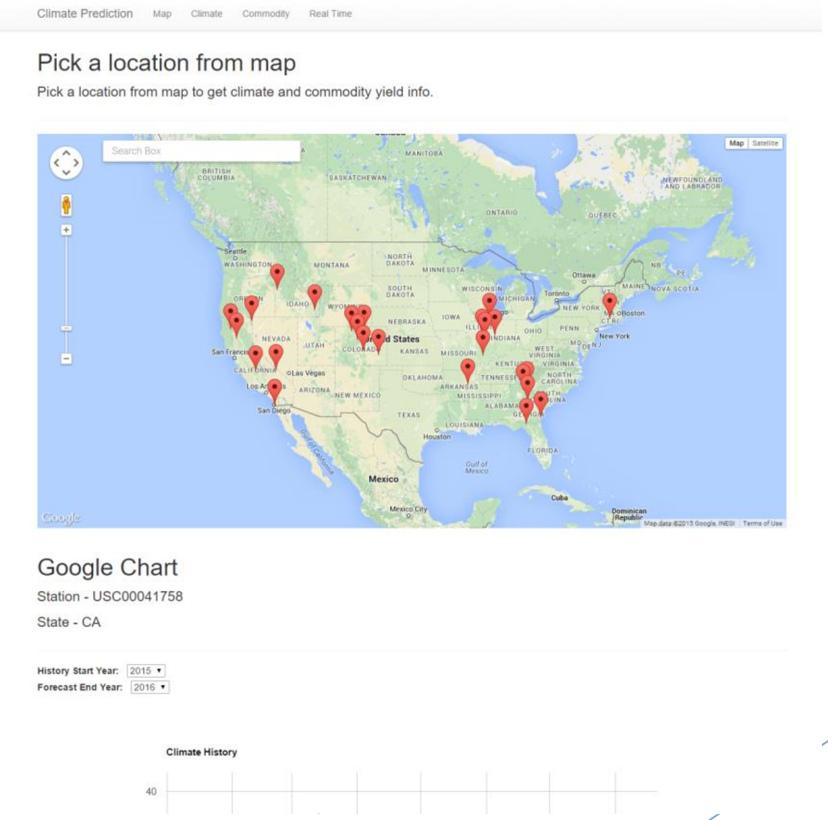
- Various source
- Different aspect for same data

Presentation

- Visual
- Interactive

Our solution

- Real-time, comprehensive data integration
- Advanced analytic algorithm
- Unified access portal
- Interactive and customizable Data Visualization



Pick a location from map

Pick a location from map to get climate and commodity yield info.

Google Chart

Station - USC00041758

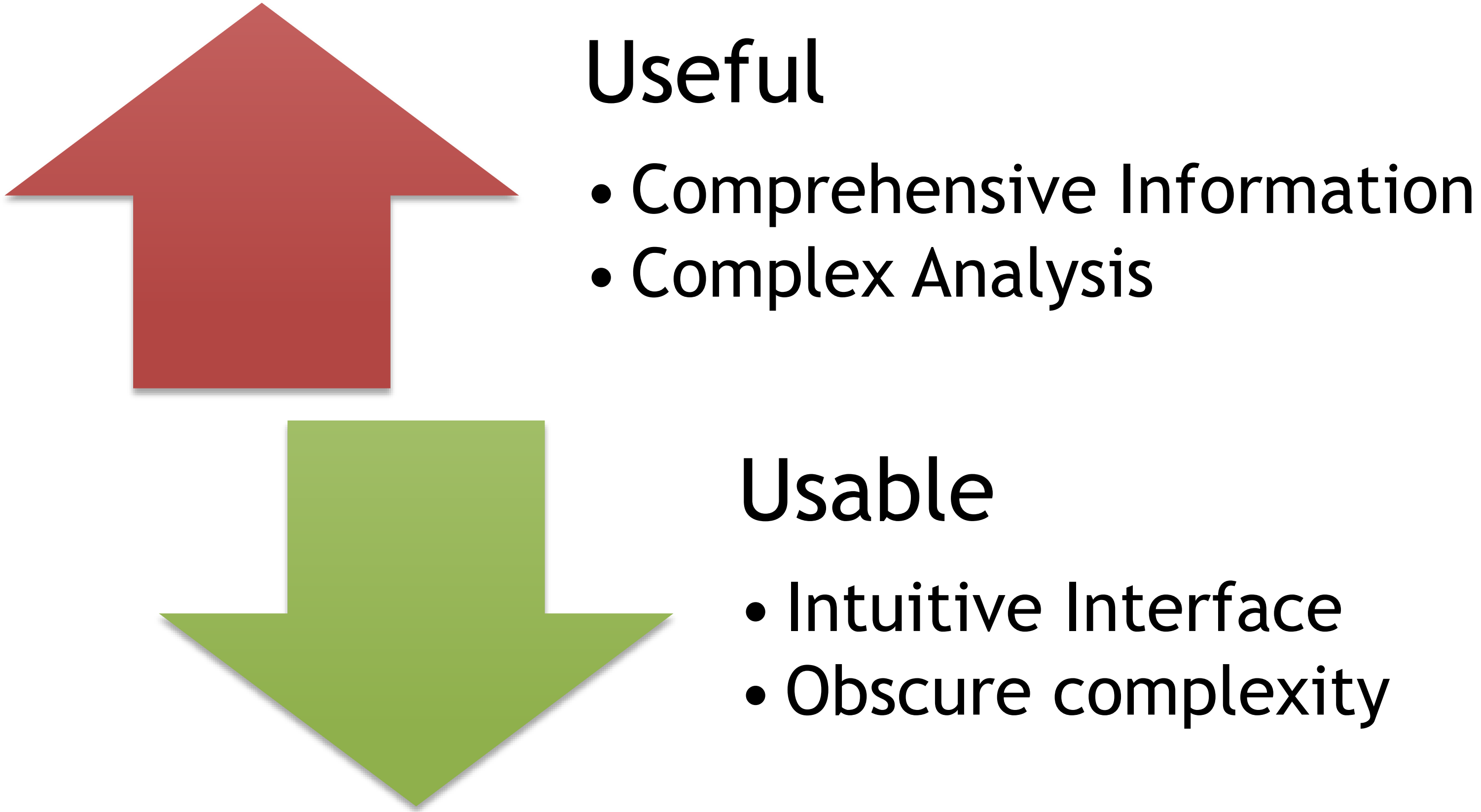
State - CA

History Start Year: 2015 ▾

Forecast End Year: 2016 ▾

Design

Goal – U2U



Useful

- Comprehensive Information
- Complex Analysis

Usable

- Intuitive Interface
- Obscure complexity

Use Case

Climate in my location?

- Query climate data by location
- Show historical and prediction data

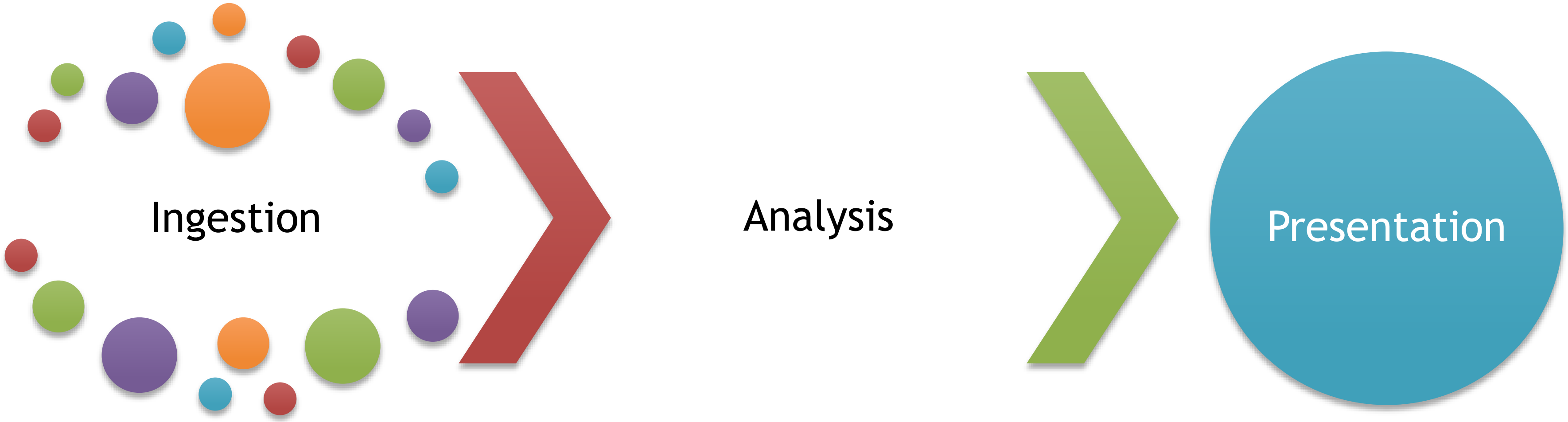
Agriculture Yield in my state?

- Query crop yield by state of station
- Show crop yield and market index

Other location with similar climate?

- List location with similar climate
- Query Anything with these location

Data Flow



Ingestion

Analysis

Presentation

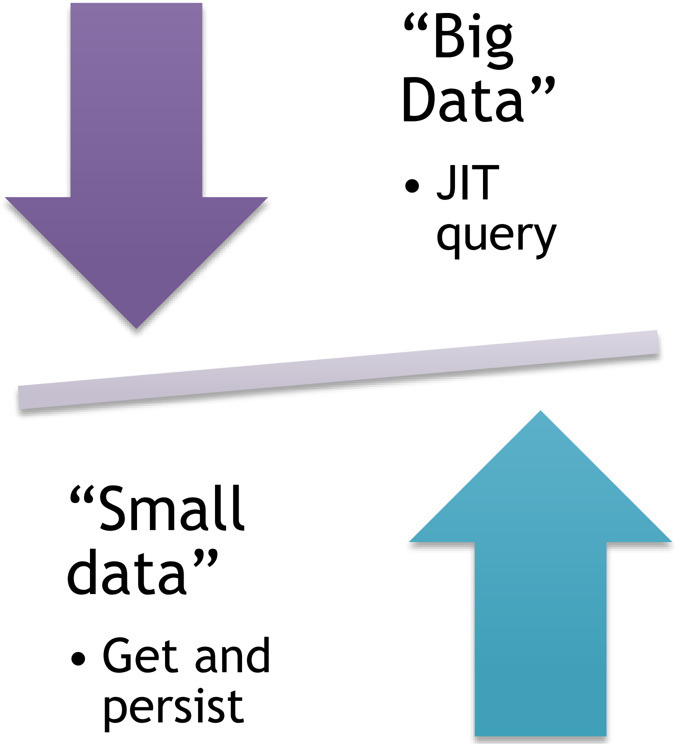
- NASS Quickstat
- ERS ARMSNASS
- NWS Climate Service
- WeatherMap

- Climate Prediction
- Climate Locality
- Yield Aggregation

- Visual Chart
- Interactive Map

Ingestion (1/2)

- Historical Data
  - Query API
  - Batch Reduce
  - Bulk store
- Real-time Data
  - Crawler
  - Streaming process
  - Incremental update



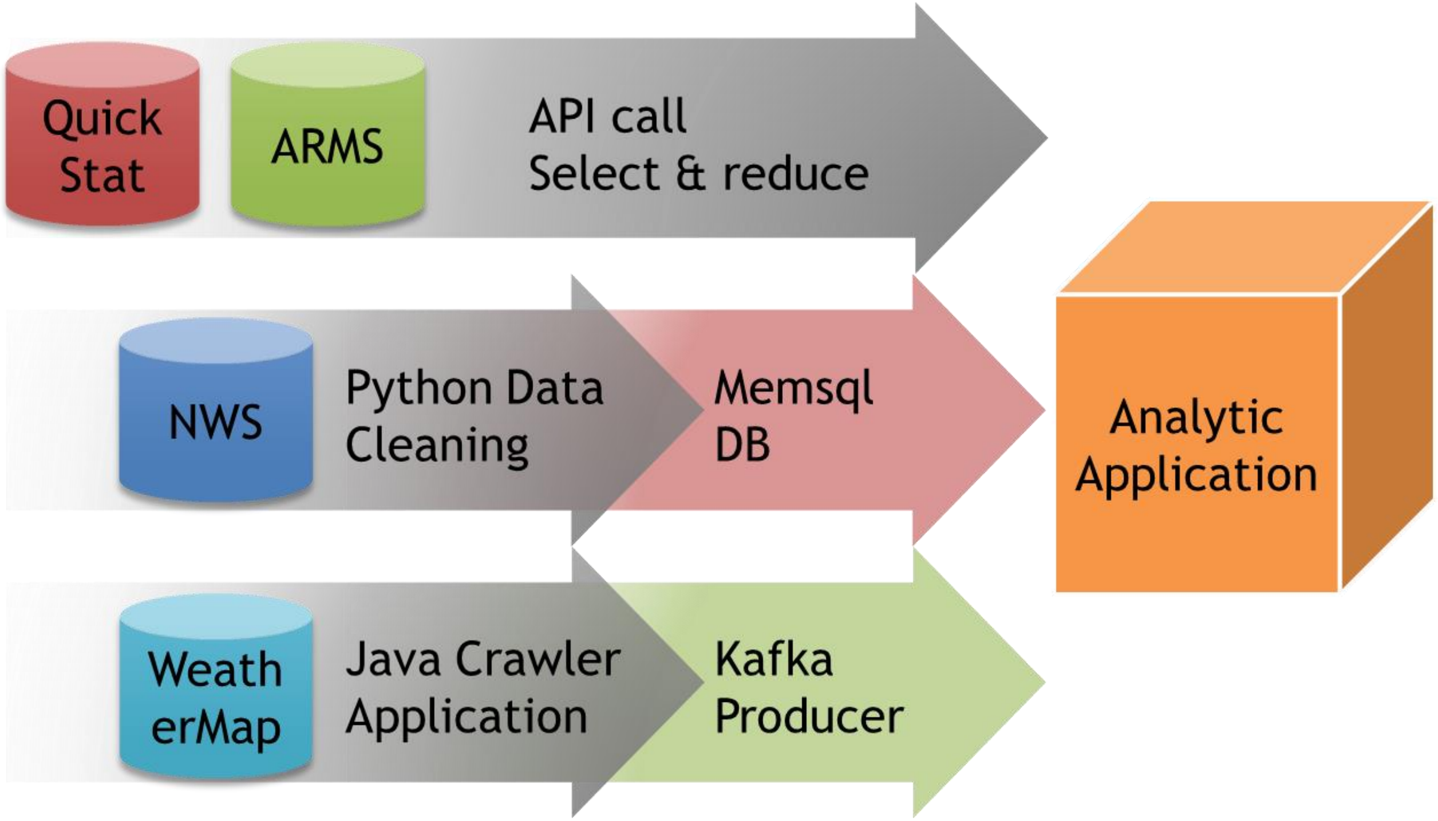
“Big Data”

- JIT query

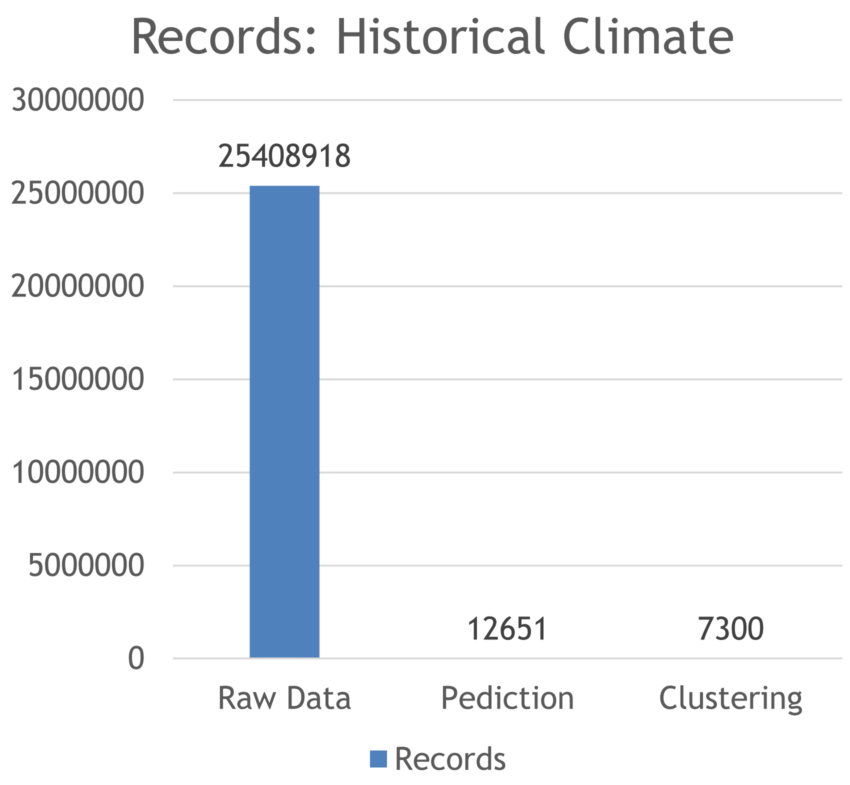
“Small data”

- Get and persist

Ingestion (2/2)

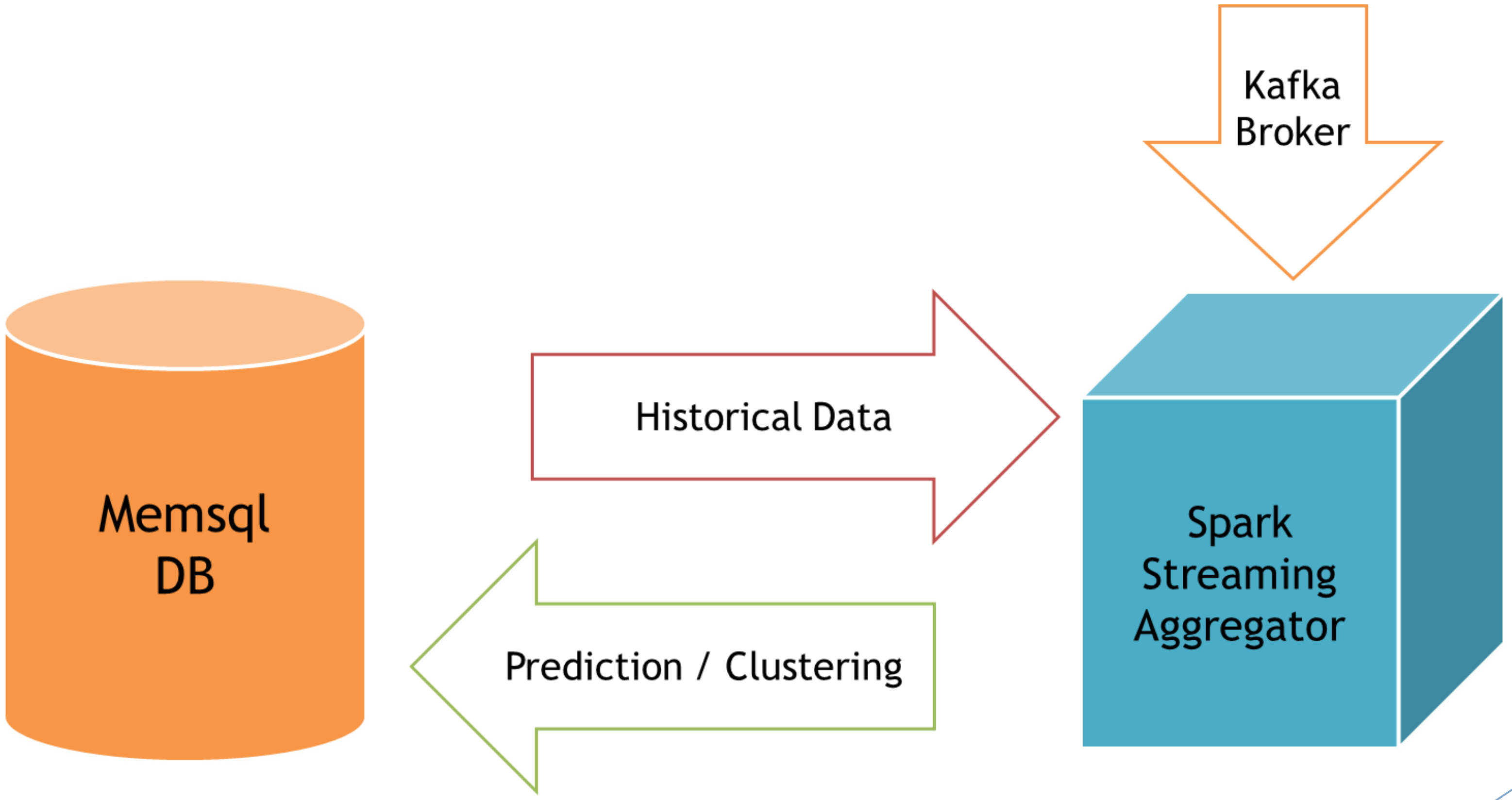


Analysis (1/2)



- Aggregation
  - Map reduce
- Locality
  - K-means
  - DBSCAN
- Prediction
  - ARIMA
- Distributed Algorithm: reason

Analysis (2/2)

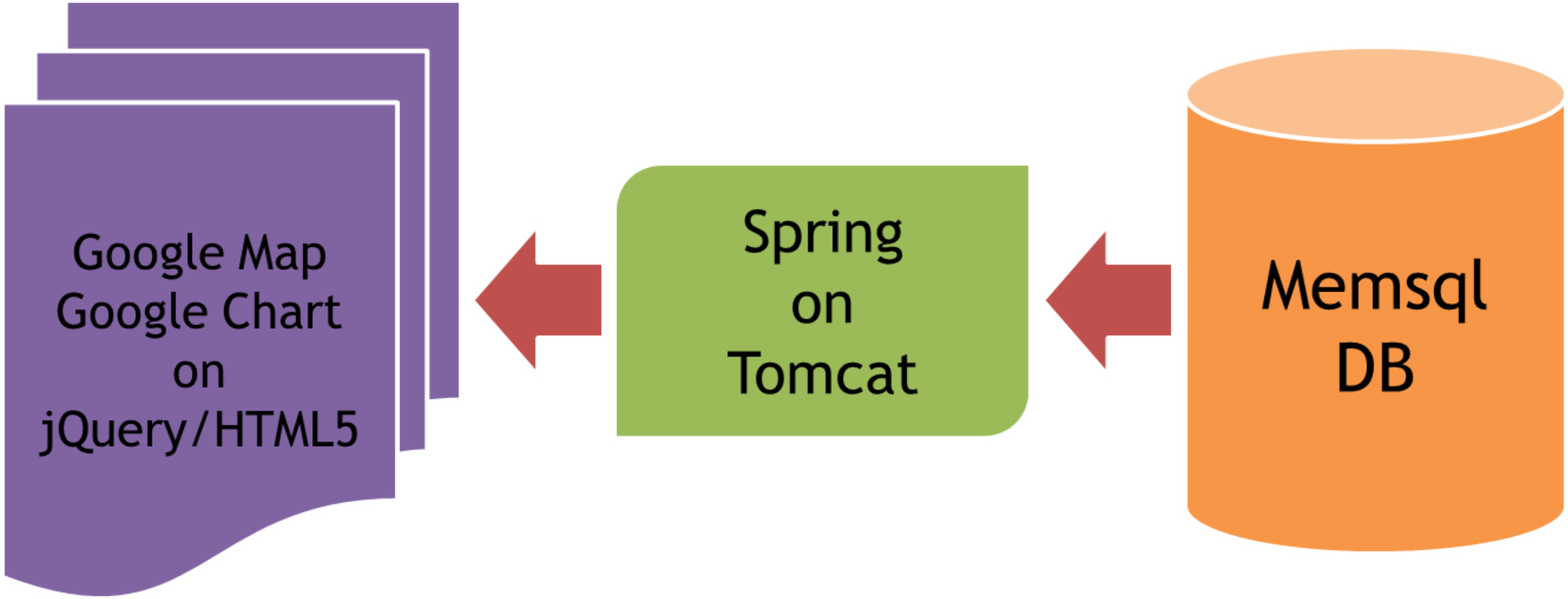


Kafka
Broker
Memsql
DB
Historical Data
Spark
Streaming
Aggregator
Prediction / Clustering

Presentation (1/2)

- Visual
  - User: everyone, not only data scientist
  - Intuitive
- Interactive
  - Data up to date
  - Present content on demand

Presentation (2/2)

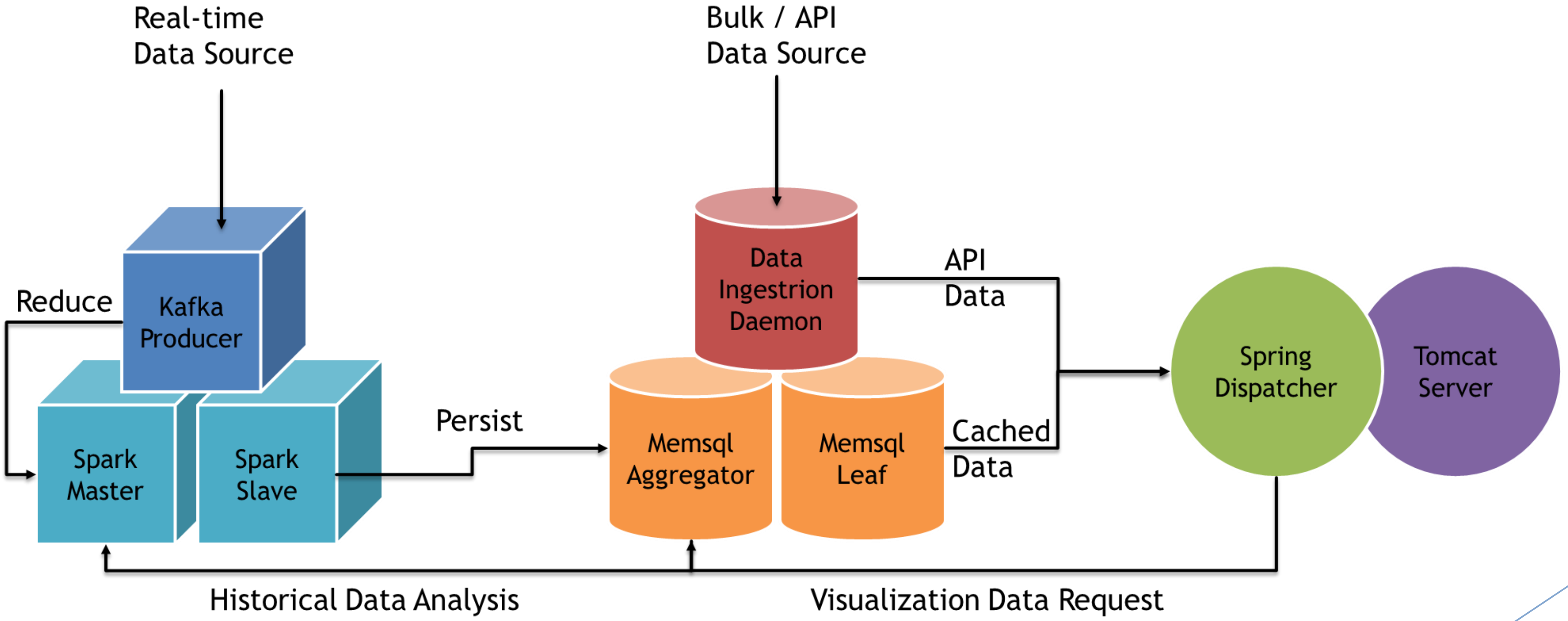


Google Map
Google Chart
on
jQuery/HTML5

Spring
on
Tomcat

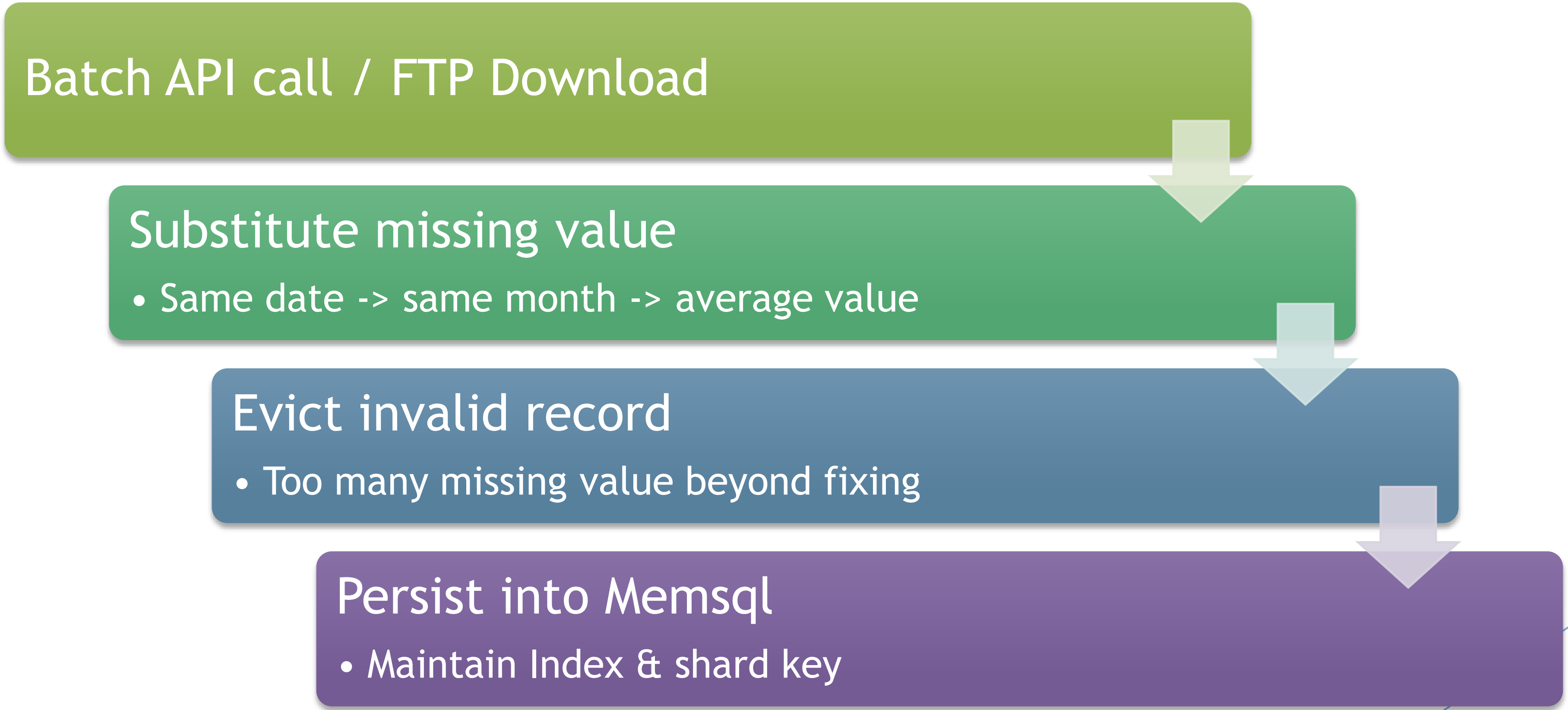
Memsql
DB

Full stack


Real-time
Data Source
Bulk / API
Data Source
Reduce
Kafka
Producer
Data
Ingestrion
Daemon
API
Data
Spring
Dispatcher
Tomcat
Server
Spark
Master
Spark
Slave
Persist
Memsql
Aggregator
Memsql
Leaf
Cached
Data
Historical Data Analysis
Visualization Data Request

Implementation

Data Cleaning



Batch API call / FTP Download

Substitute missing value

- Same date -> same month -> average value

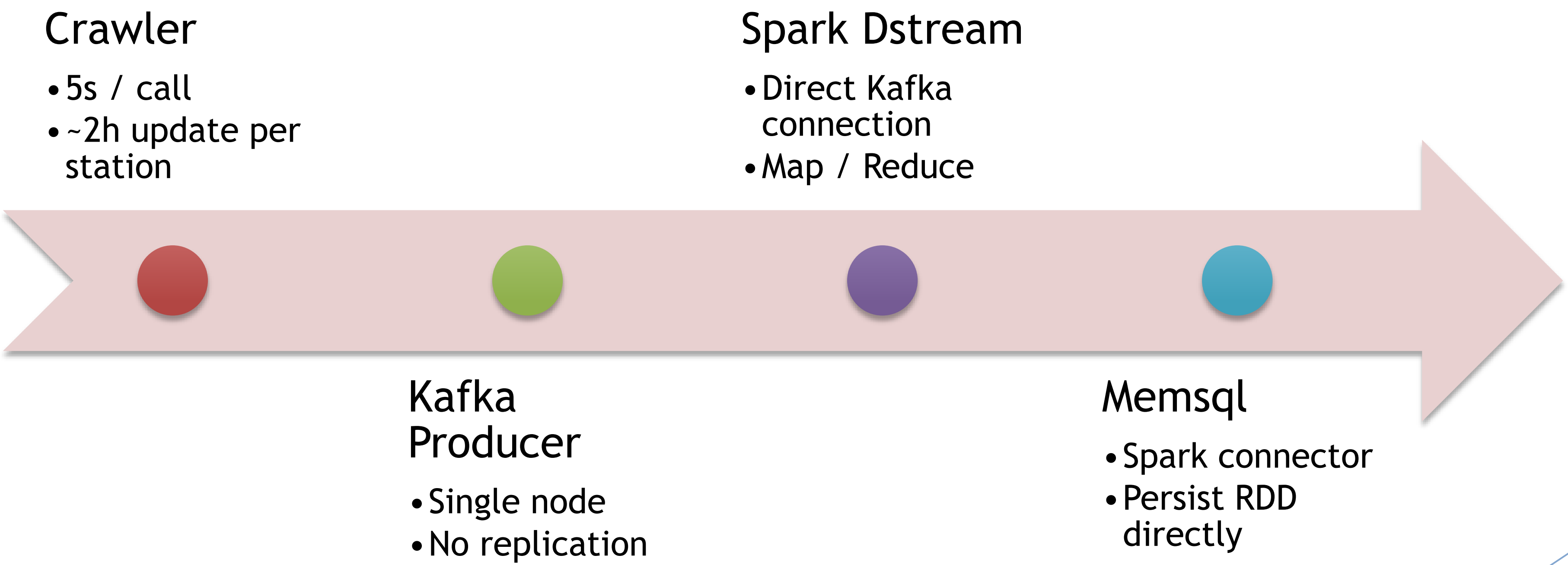
Evict invalid record

- Too many missing value beyond fixing

Persist into Memsql

- Maintain Index & shard key

Real-time Weather Data



Crawler

- 5s / call
- ~2h update per station

Kafka Producer

- Single node
- No replication

Spark Dstream

- Direct Kafka connection
- Map / Reduce

Memsql

- Spark connector
- Persist RDD directly

Analysis Algorithm

ARIMA

- Distributed version by Cloudera
- Unfortunately it’s non-seasonal, we implemented seasonal version
- Final model: ARIMA(2,0,0) * (0,1,1) [365]

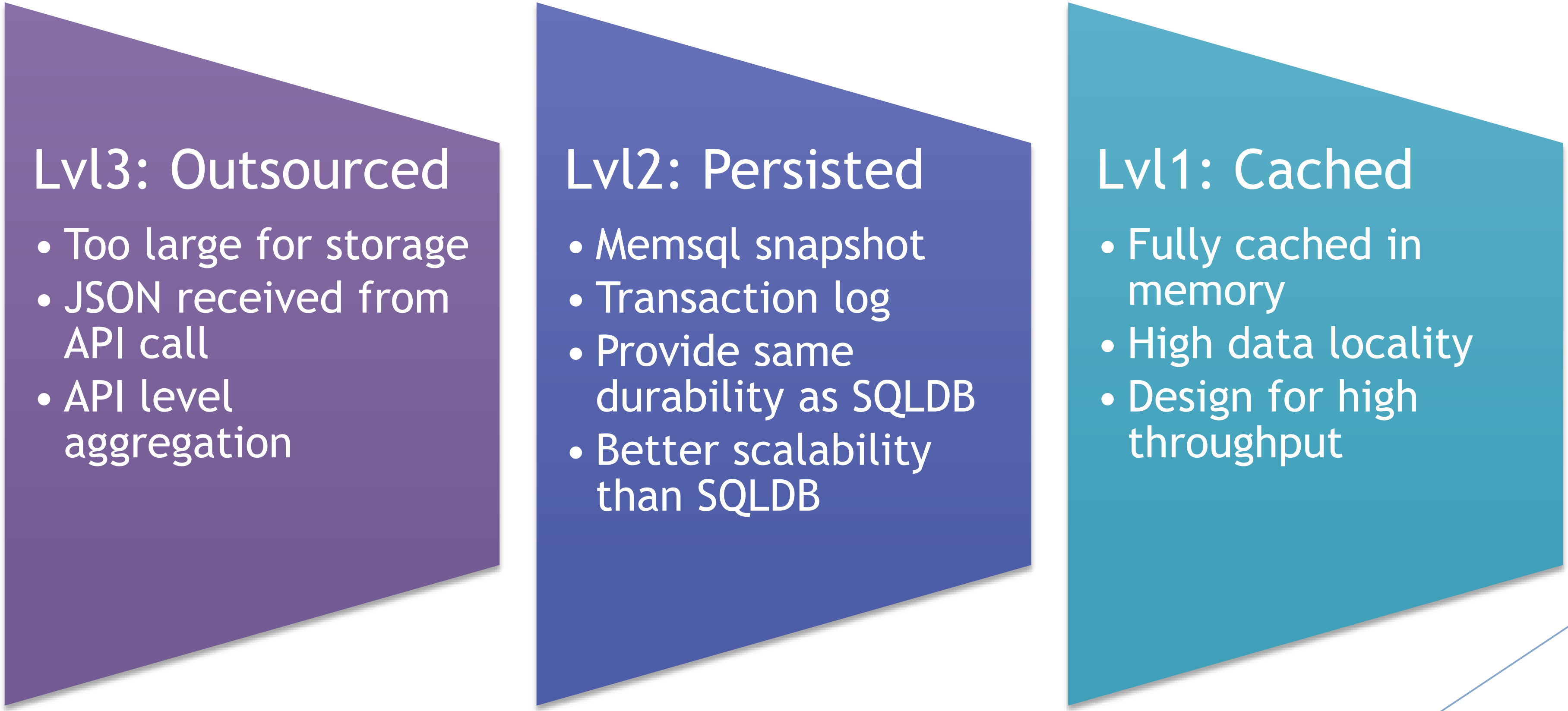
DBSCAN

- Open sourced package, but with serious performance issue
- Clustering result align with Kmeans
- Some location marked as “noise”, does not fit domain knowledge

Kmeans

- Spark MLLIB Implementation, fast and robust
- Ensure every location to be categorized
- We choose to use Kmeans

Data Storage



Lvl3: Outsourced

- Too large for storage
- JSON received from API call
- API level aggregation

Lvl2: Persisted

- Memsql snapshot
- Transaction log
- Provide same durability as SQLDB
- Better scalability than SQLDB

Lvl1: Cached

- Fully cached in memory
- High data locality
- Design for high throughput

Data Presentation

Back-end: Spring framework

- Full MVC structure, with MySQL JDBC driver (compatibly with Memsql)
- Run on Tomcat, WAR deployment

Front-end: jQuery, Google Map API, Google Chart

- Single page application, fully utilized AJAX, asynchronized update from multiple data source
- Dynamic and device-independent UI build on Bootstrap CSS

DEMO

Thank you!