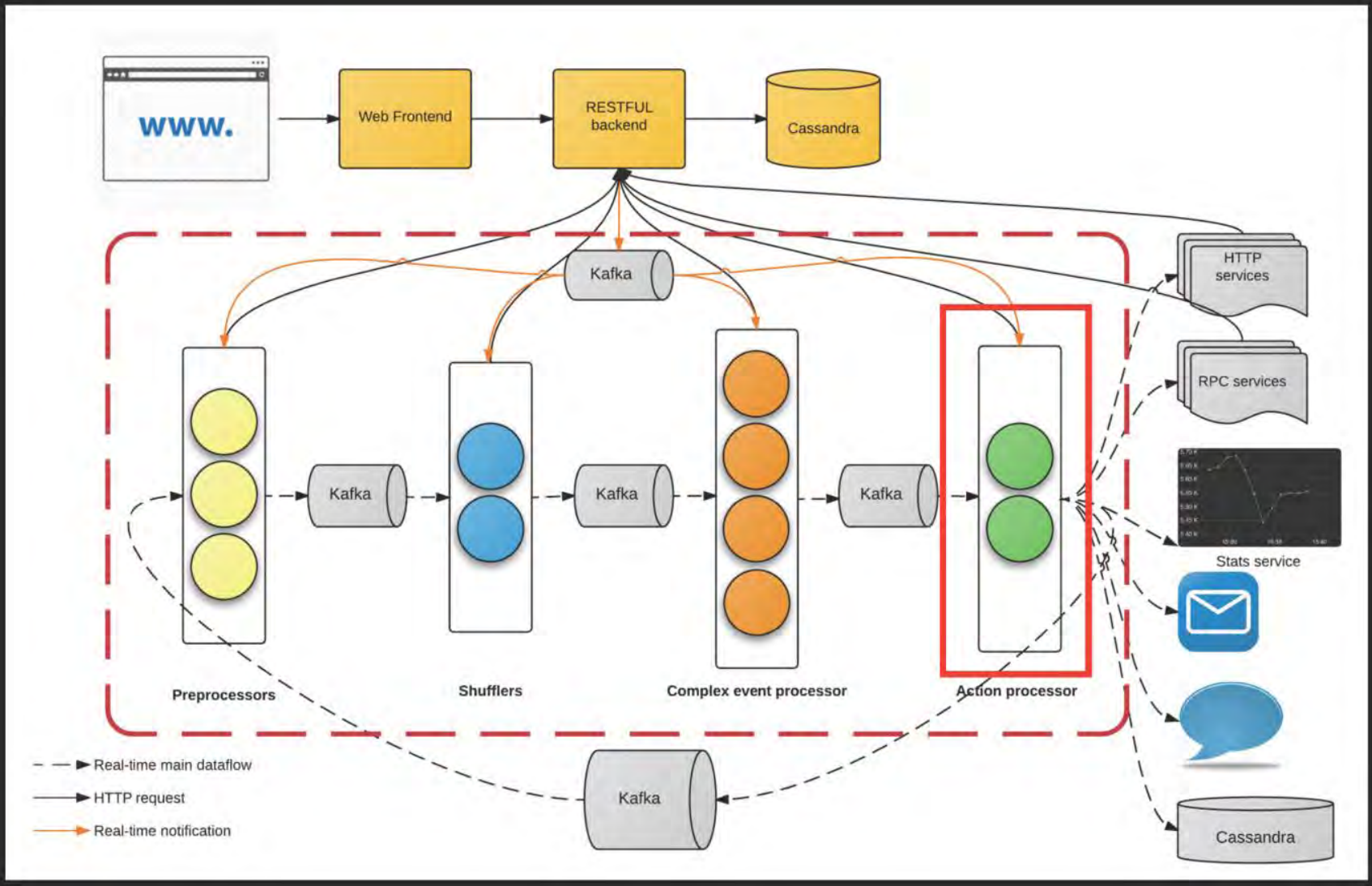
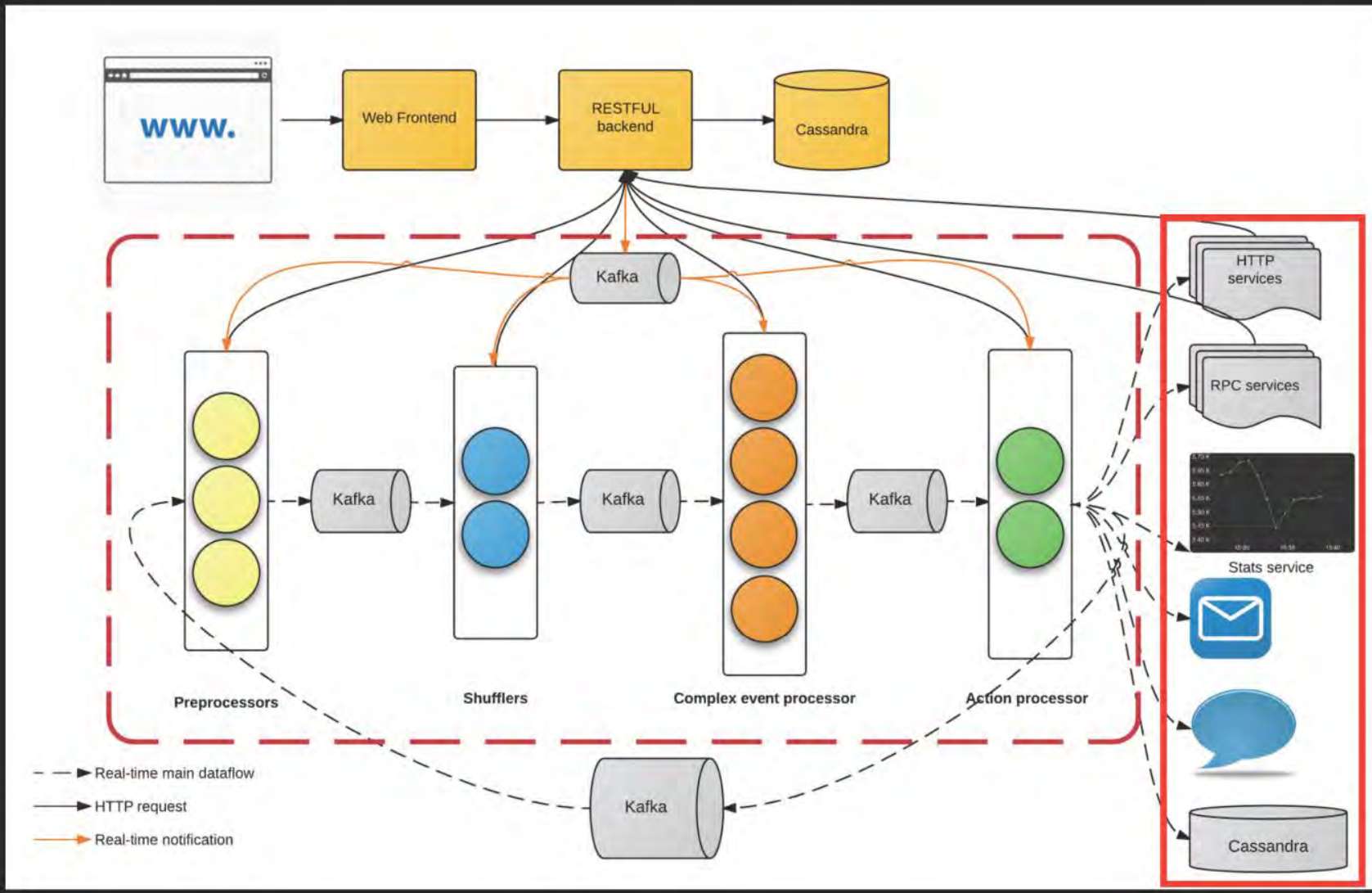
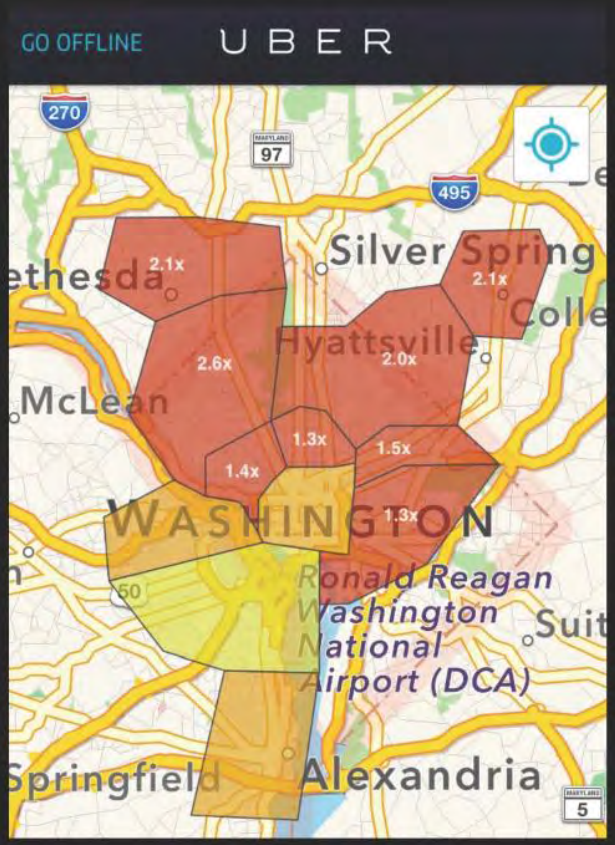

www.
Web Frontend
RESTFUL backend
Cassandra
Kafka
HTTP services
RPC services
Kafka
Kafka
Kafka
Stats service
Preprocessors
Shufflers
Complex event processor
Action processor
Kafka
Cassandra
Real-time main dataflow
HTTP request
Real-time notification

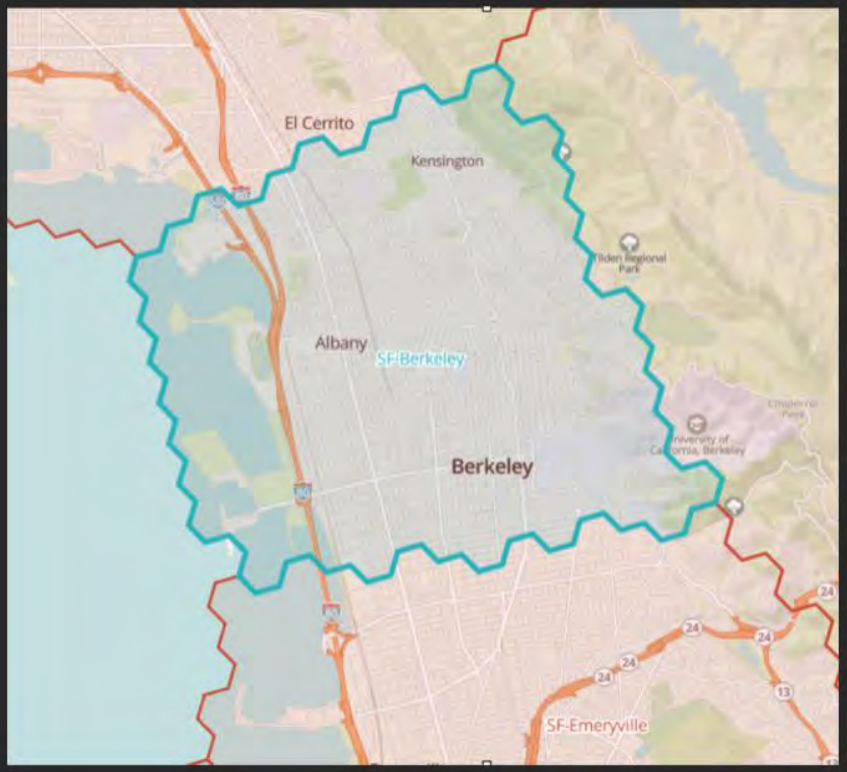

www.
Web Frontend
RESTFUL backend
Cassandra
Kafka
HTTP services
RPC services
Stats service
Kafka
Kafka
Kafka
Preprocessors
Shufflers
Complex event processor
Action processor
Kafka
Cassandra
Real-time main dataflow
HTTP request
Real-time notification

We need to evolve our architecture for other analytics

Clustering

Manually Created Cluster



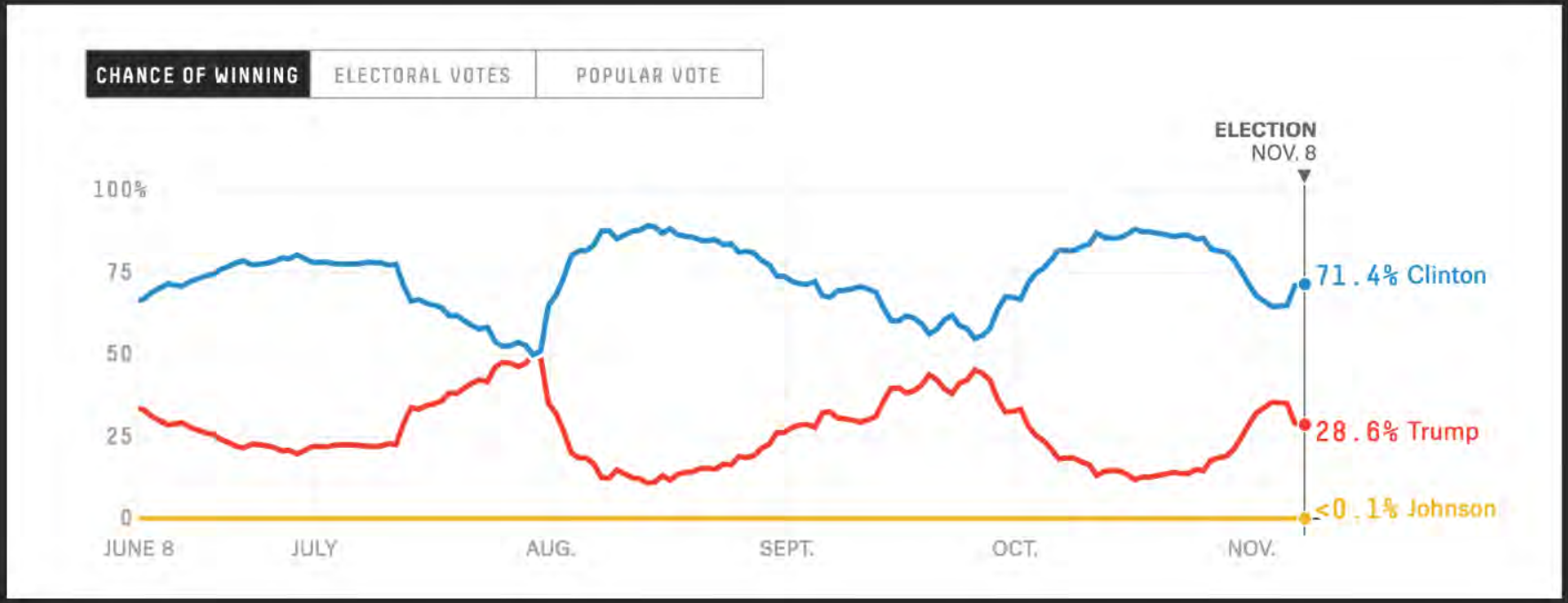


Call for algorithmically created clusters

- Clustering based on key performance metrics

Call for algorithmically created cluster

- Clustering based on key performance metrics
- Continuously measure the clusters



Call for algorithmically created clusters

- Clustering based on key performance metrics
- Continuously measure the clusters
- Different clustering for different business needs

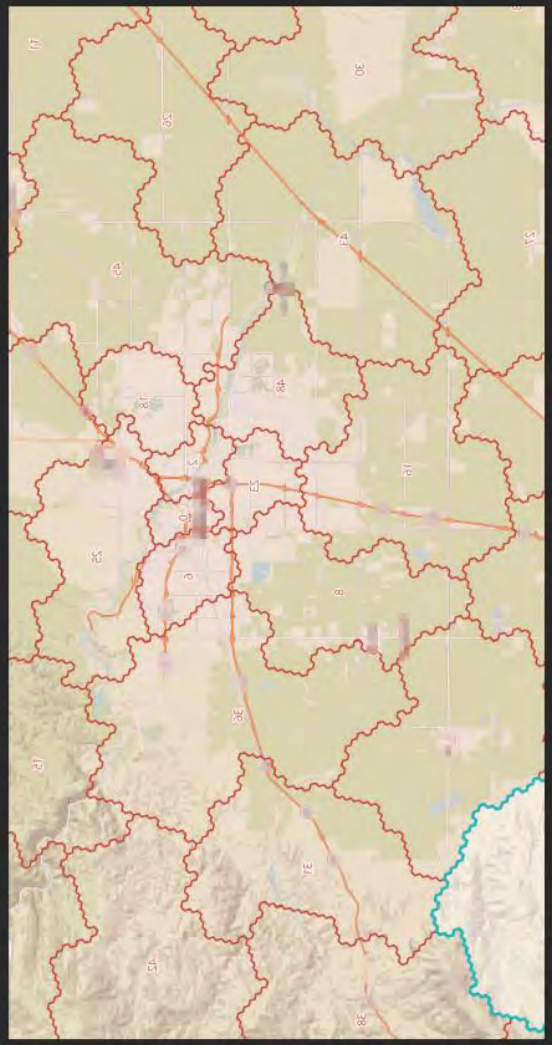
Call for algorithmically created clusters

- Clustering based on key performance metrics
- Continuously measure the clusters
- Different clustering for different business needs
- Create clusters in minutes for all cities

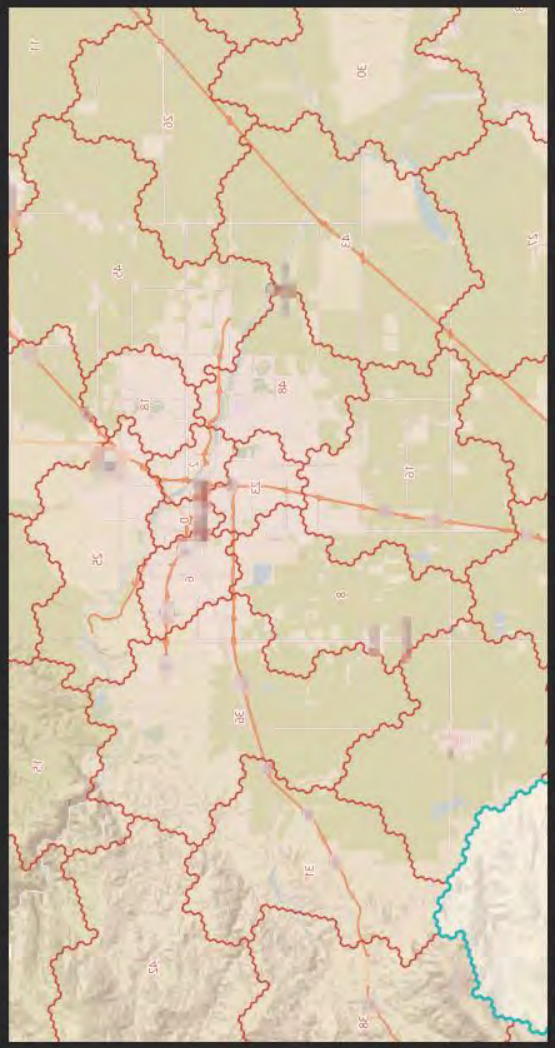
Call for algorithmically created clusters

- Clustering based on key performance metrics
- Continuously measure the clusters
- Different clustering for different business needs
- Create clusters in minutes for all cities
- Foundation for other stream analytics

Home-grown Clustering Service

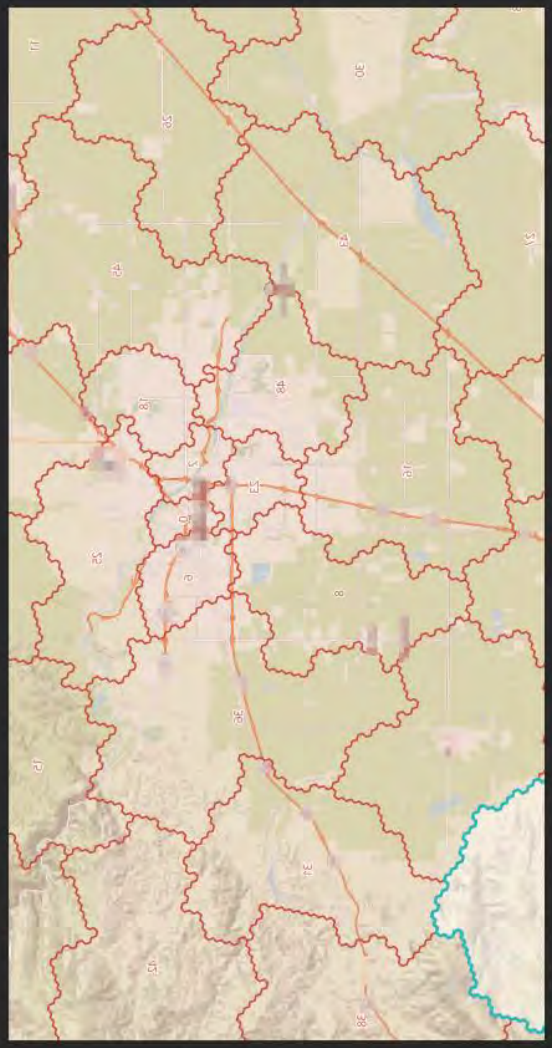


Home-grown Clustering Service



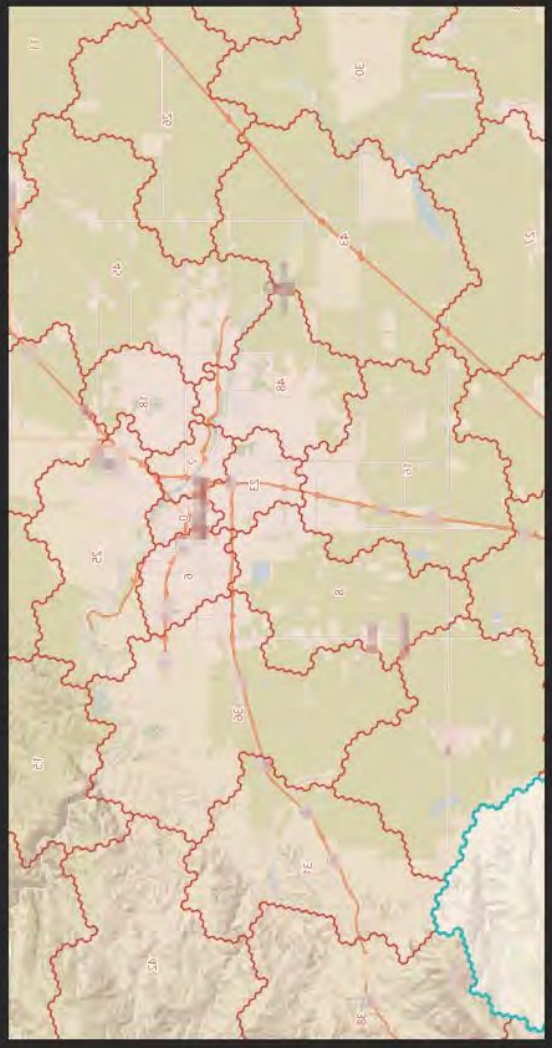
- All cities under 3 minutes

Home-grown Clustering Service



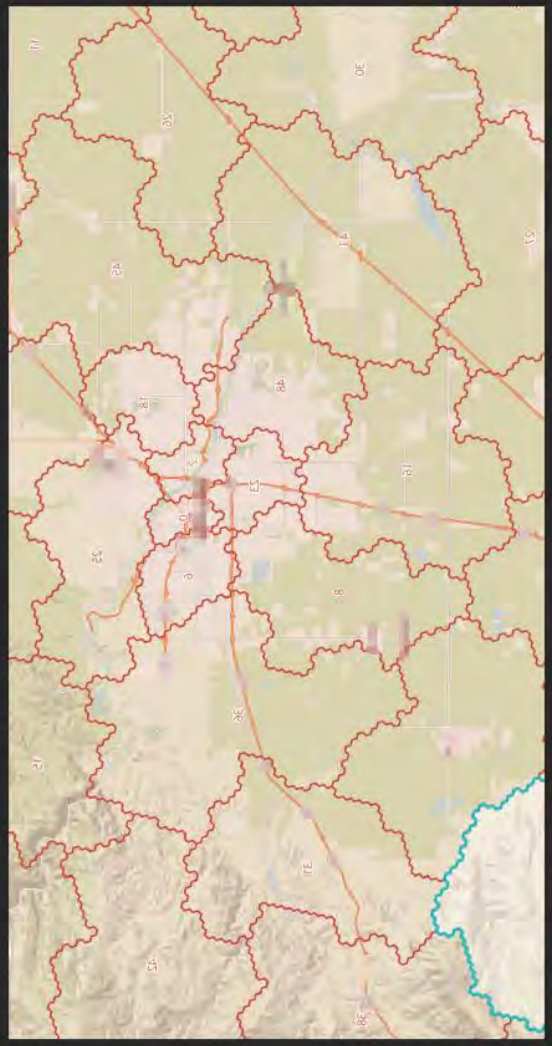
- All cities under 3 minutes
- Pluggable algorithms and measurements

Home-grown Clustering Service



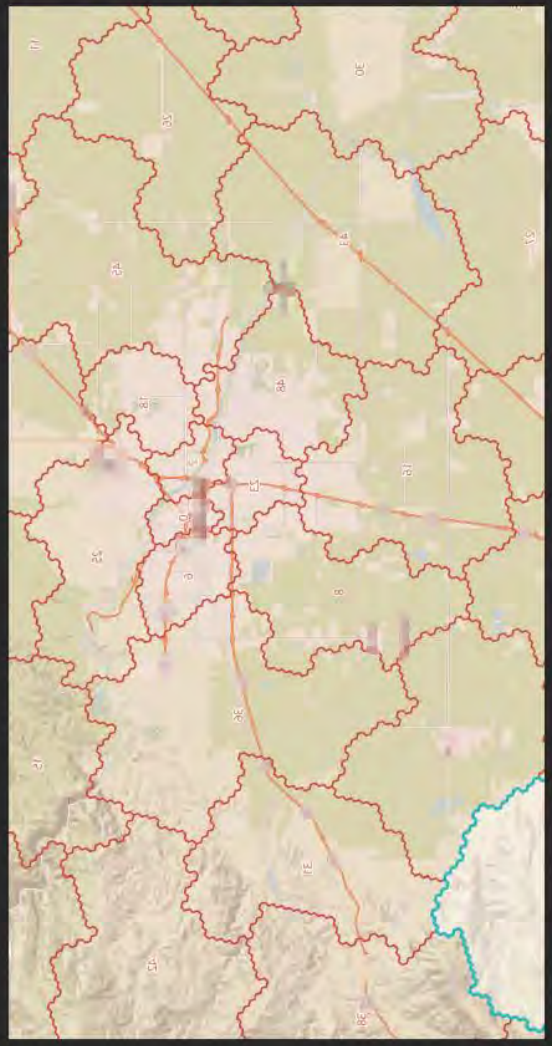
- All cities under 3 minutes
- Easily pluggable algorithms and measurements
- Historical geo-temporal data for clustering

Home-grown Clustering Service



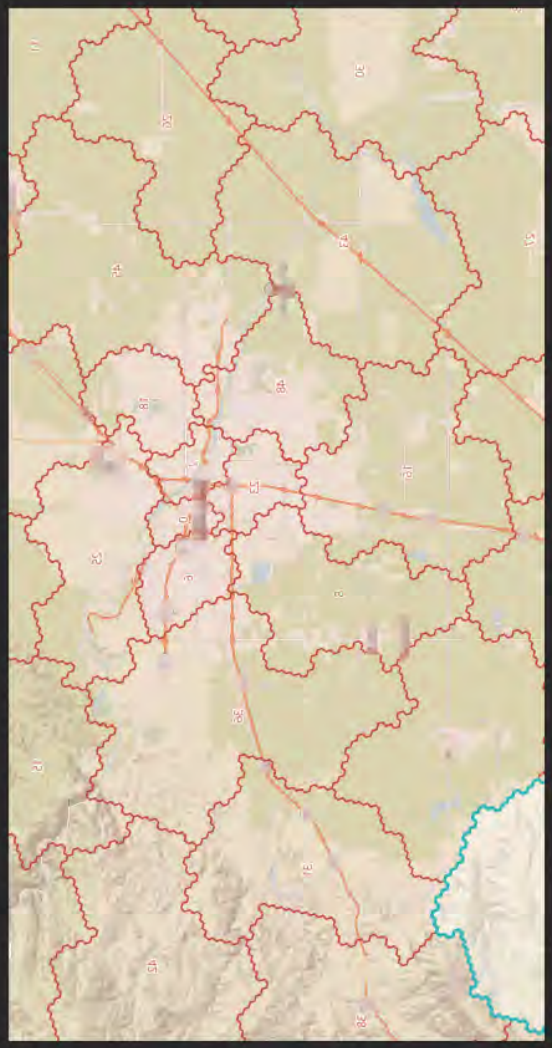
- All cities under 3 minutes
- Easily pluggable algorithms and measurements
- Historical geo-temporal data for clustering
- Real-time geo-temporal data for measurement

Home-grown Clustering Service



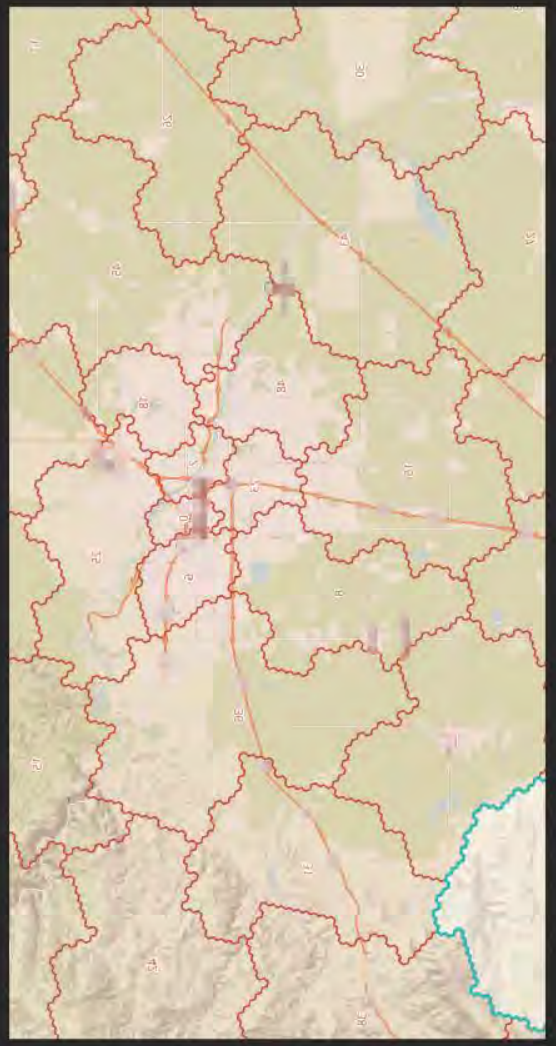
- All cities under 3 minutes
- Easily pluggable algorithms and measurements
- Historical geo-temporal data for clustering
- Real-time geo-temporal data for measurement
- Shared optimizations

Home-grown Clustering Service



- All cities under 3 minutes
- Easily pluggable algorithms and measurements
- Historical geo-temporal data for clustering
- Real-time geo-temporal data for measurement
- Shared optimizations. To put things in perspective:
  - 70,000 hexagons in SF
  - Naive distance function requires at least 70,000 x 70,000 = 4.9 billion pairs!

Home-grown Clustering Service



- All cities under 3 minutes
- Easily pluggable algorithms and measurements
- Historical geo-temporal data for clustering
- Real-time geo-temporal data for measurement
- Shared optimizations
  - Incremental updates
  - Compact data representation

Forecasting

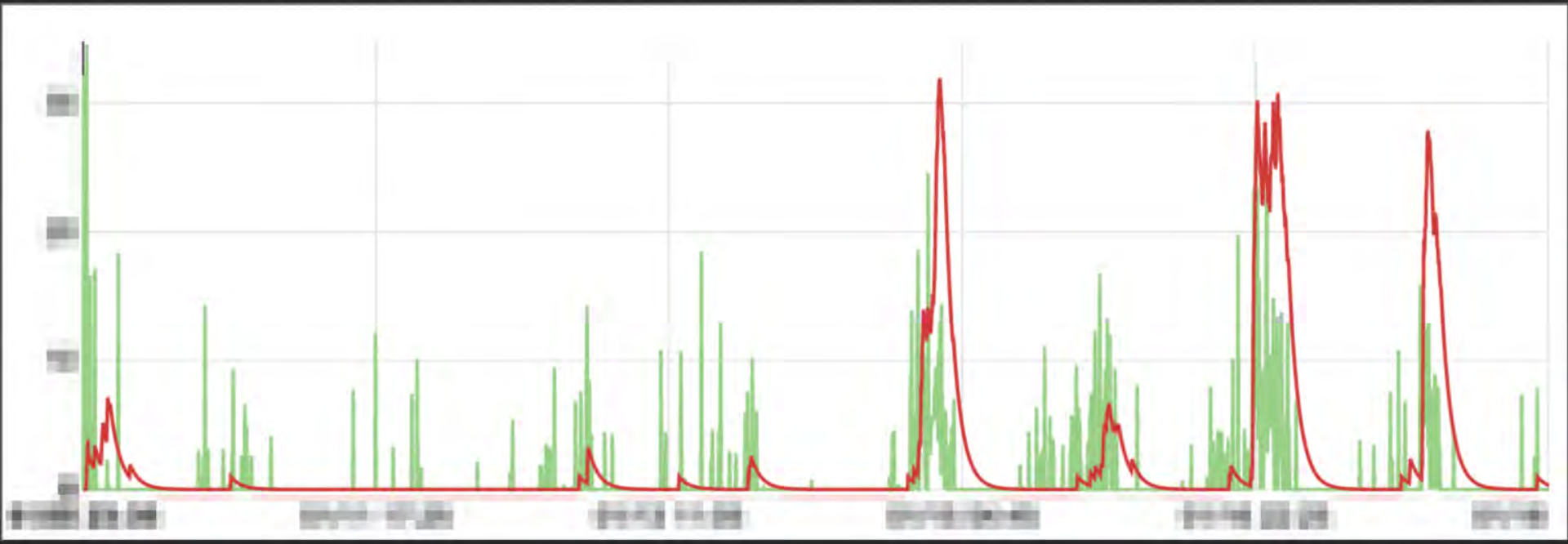
- Every decision is based on forecasting

Forecasting

- Forecasting based on both historical data and stream input

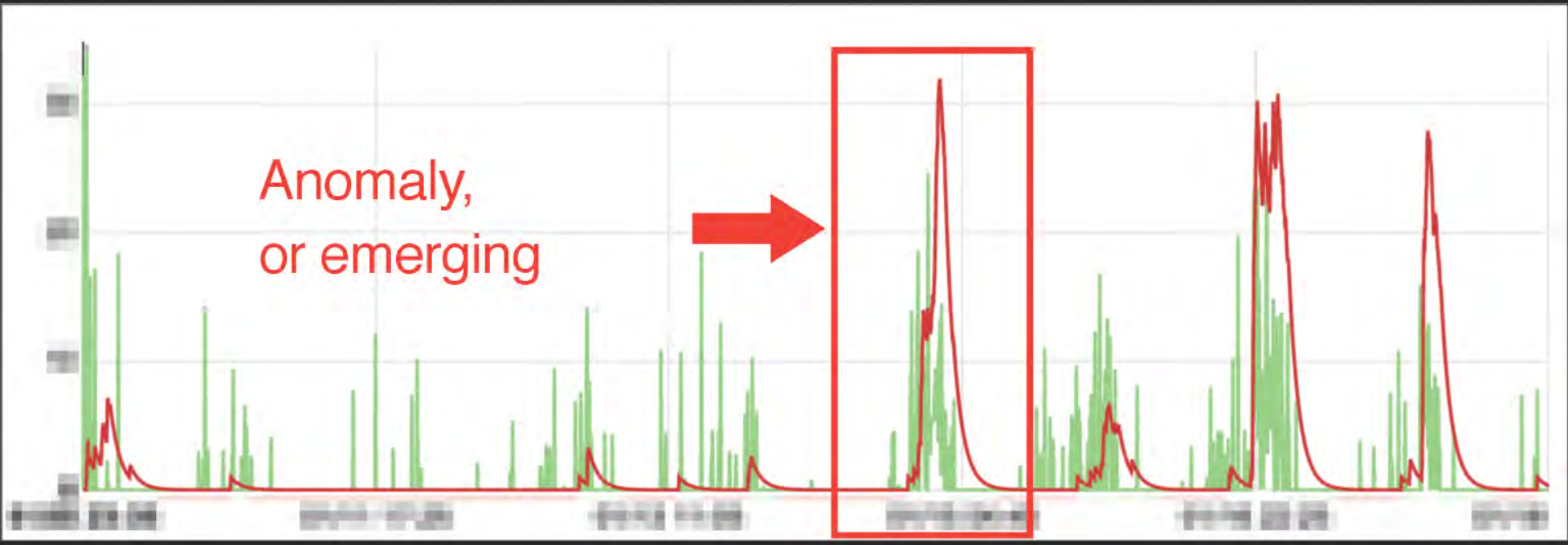
Forecasting

- Forecasting based on both historical data and stream input



Forecasting

- Forecasting based on both historical data and stream input



Forecasting

- Spatially granular forecasting - down to every hexagon

Forecasting

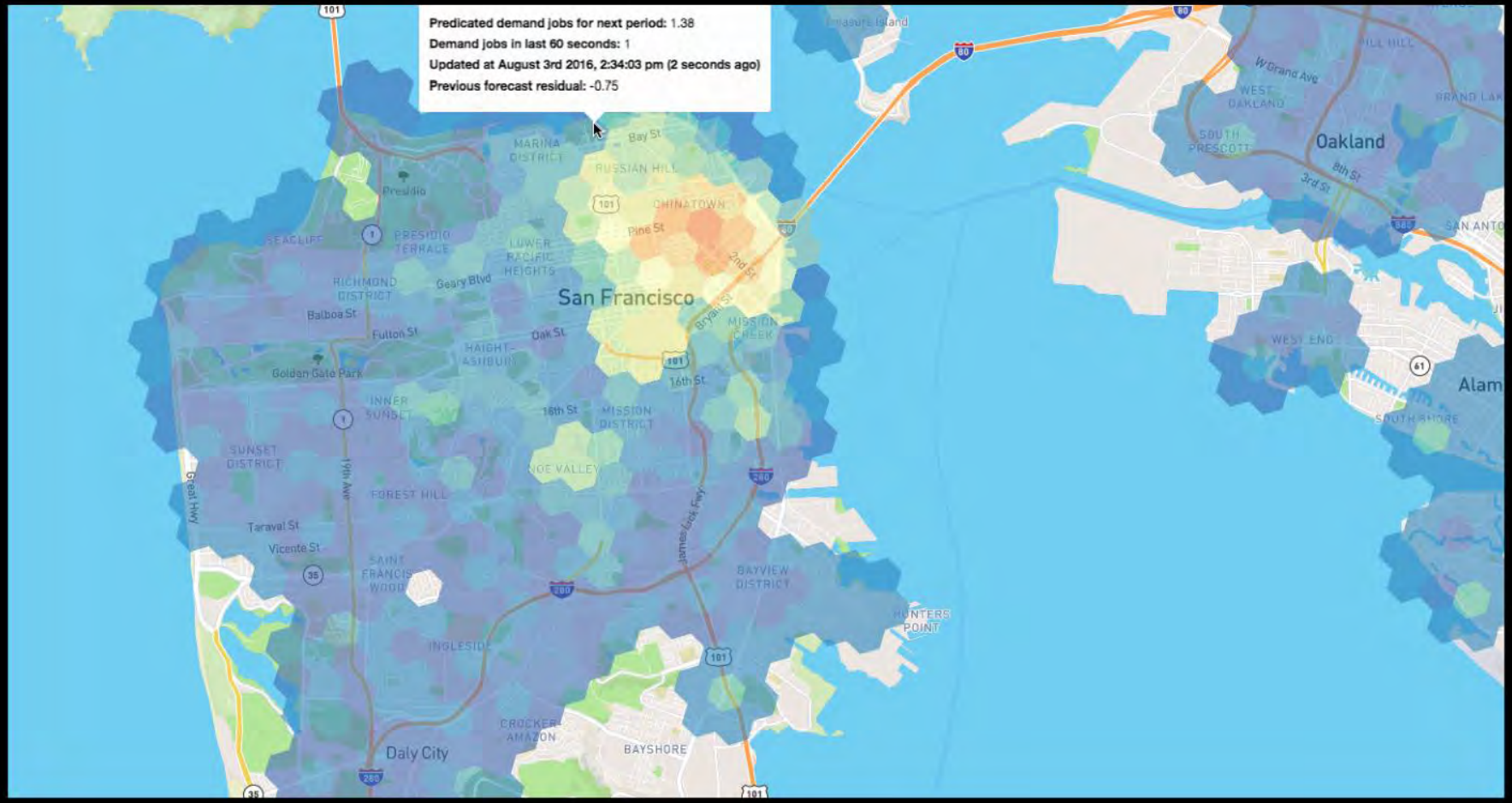
- Spatially granular forecasting - down to every hexagon

Forecasting

- Temporally granular forecasting - down to every minute

Forecasting

- Temporally granular forecasting - down to every minute



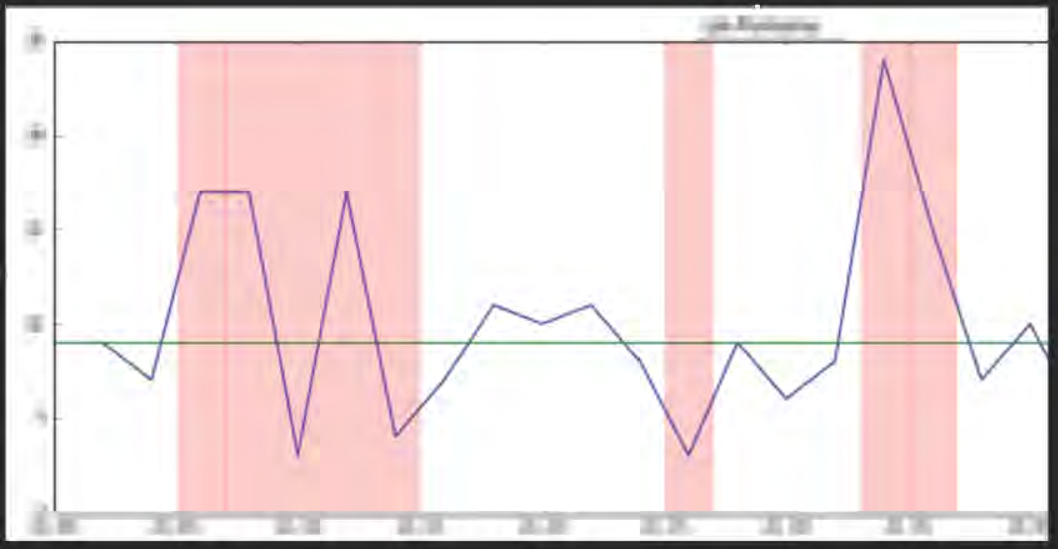
Pattern Detection

- Similarity of different metrics across geolocation and time
- Metric outliers across geolocations and time
- Frequent occurrences of certain patterns
- Clustered behavior
- Anomalies

Common Requirements in Pattern Detection

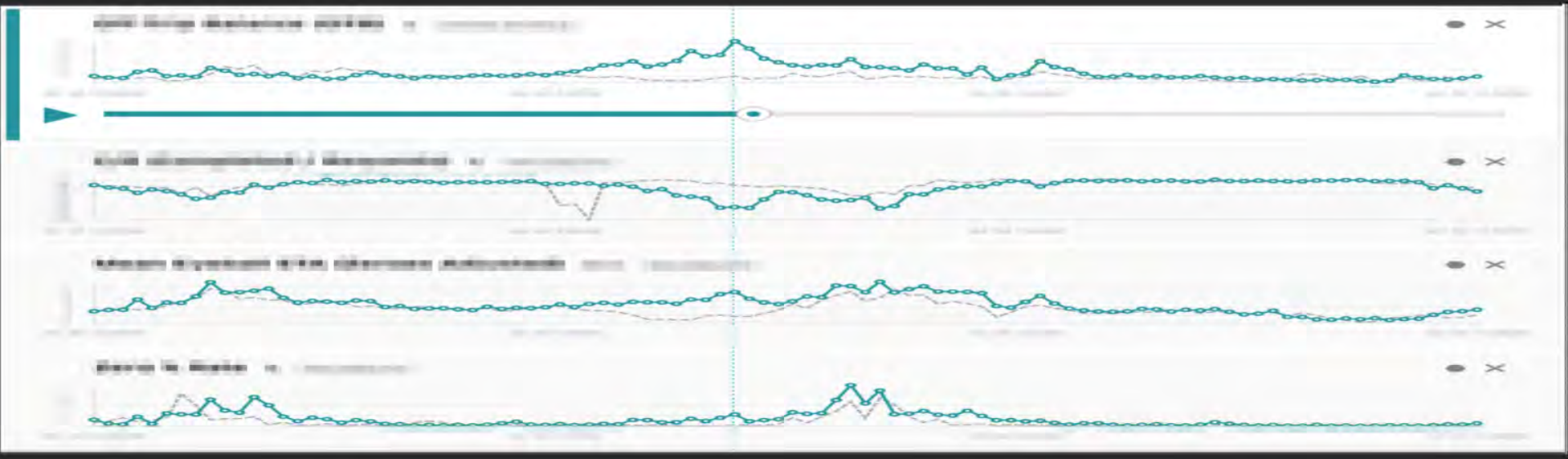
- Not just traditional time series analysis
- Incorporating insights on marketplace data
- Required both historical data and real-time input
- Spatially granular patterns - down to every hexagon
- Temporally granular patterns - down to every minute

Example: Anomaly Detection

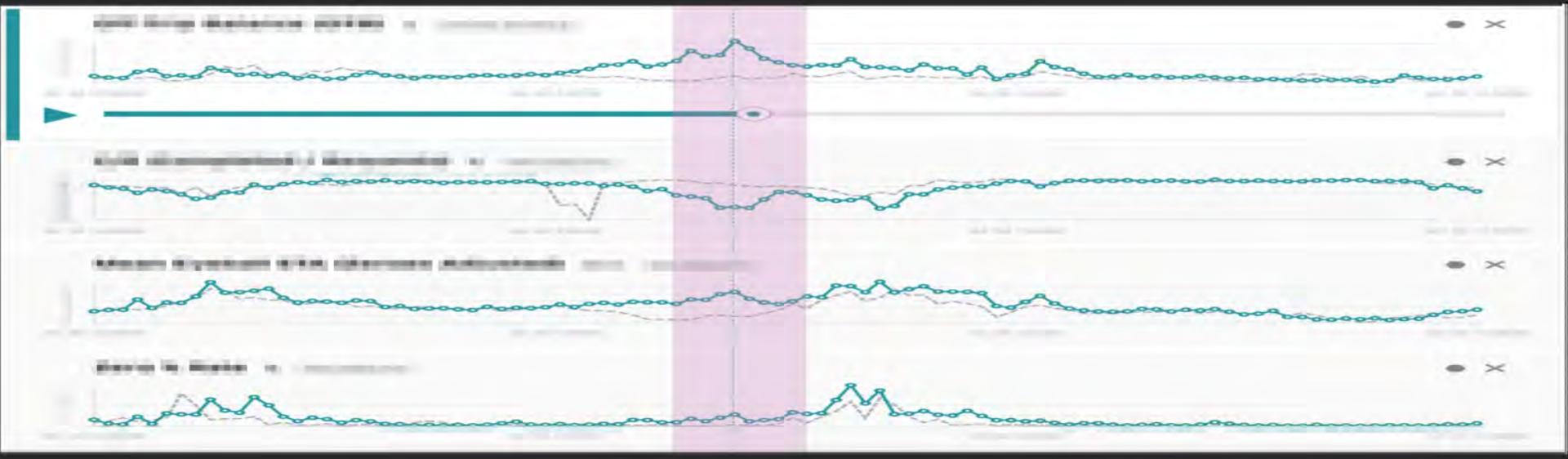


- Simple time series analysis
- For a single geo area
- Can be noisy

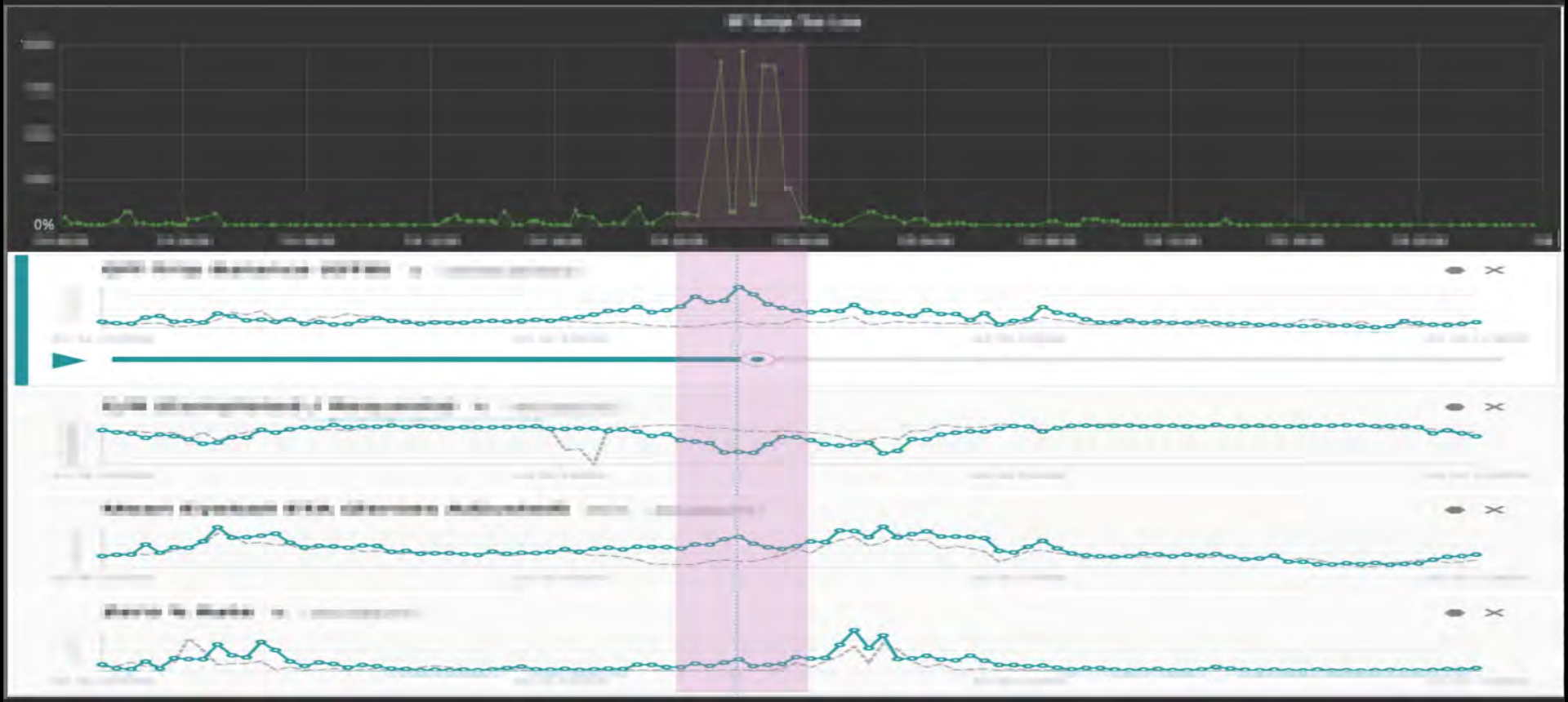
A More Realistic Anomaly Detection



Example: Anomaly Detection



Example: Anomaly Detection



What’s the right architecture to support the analytics use cases?

Shared abstraction: multi-dimensional geo-temporal data

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time

https://www.oreilly.com/ideas/the-world-beyond-batch-streaming-101

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time

https://www.oreilly.com/ideas/the-world-beyond-batch-streaming-101

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time

https://www.oreilly.com/ideas/the-world-beyond-batch-streaming-101

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time
- Flexible windowing - tumbling, sliding, conditionally triggered

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time
- Flexible windowing - tumbling, sliding, conditionally triggered
- e.g. event-based triggers

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time
- Flexible windowing - tumbling, sliding, conditionally triggered
- e.g. event-based triggers
- e.g., triggers of computation results

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time
- Flexible windowing - tumbling, sliding, conditionally triggered
- Stateful processing

Shared abstraction: multi-dimensional geo-temporal data

- Time series by event time
- Flexible windowing - tumbling, sliding, conditionally triggered