

The background of the slide is a photograph of a modern, open-plan office. In the foreground, a man is seated at a desk, working on a laptop. The office has wooden desks, green chairs, and various office supplies. A large, solid green shape, resembling a stylized hill or a wave, covers the bottom half of the image, serving as a backdrop for the title and authors' names.

# Democratizing Data at GOJEK

Prakhar Mathur | Rohil Surana

# Agenda

- BACKGROUND
- DAGGERS
- DIY PORTAL
- ALERTING AND MONITORING
- IMPACT



18 Products

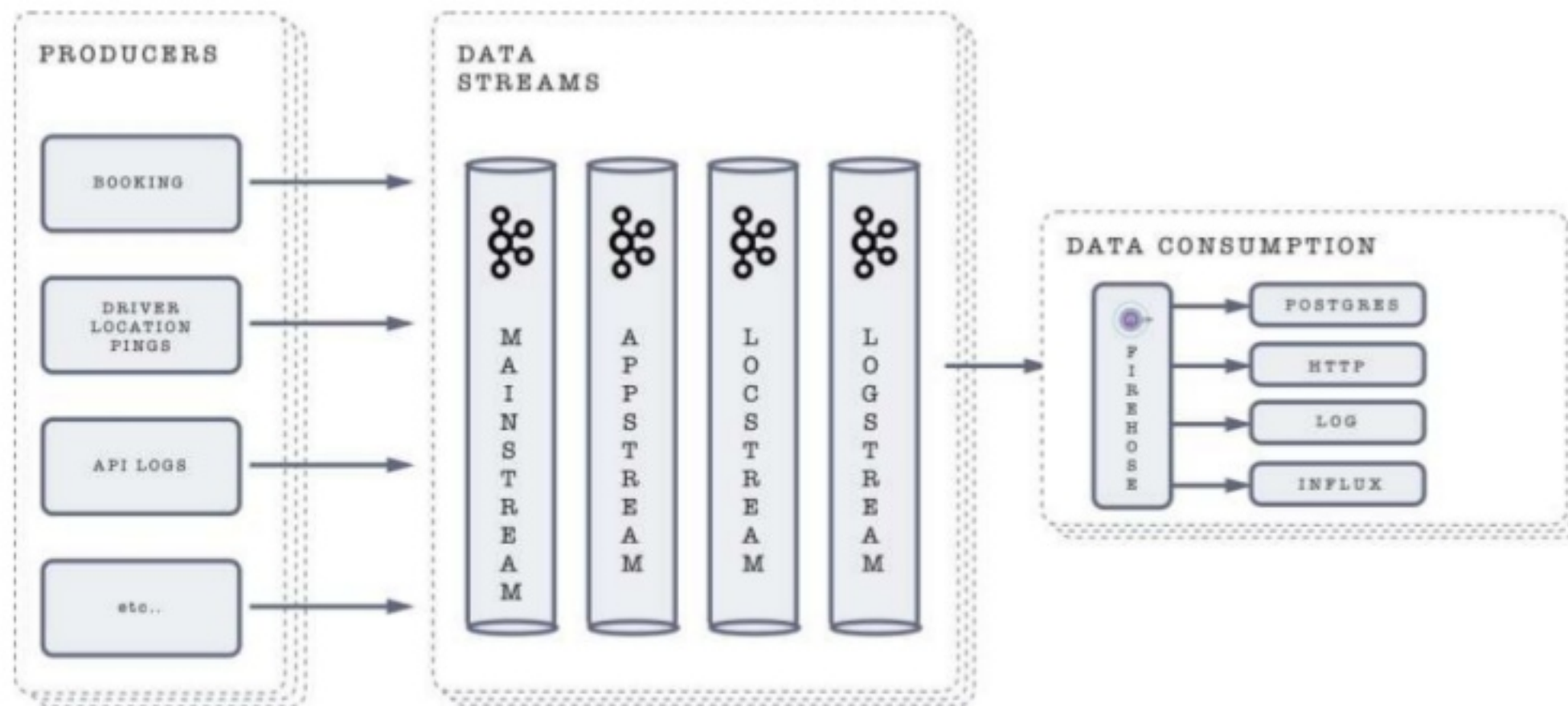
3M Orders

1M Drivers

500 Microservices

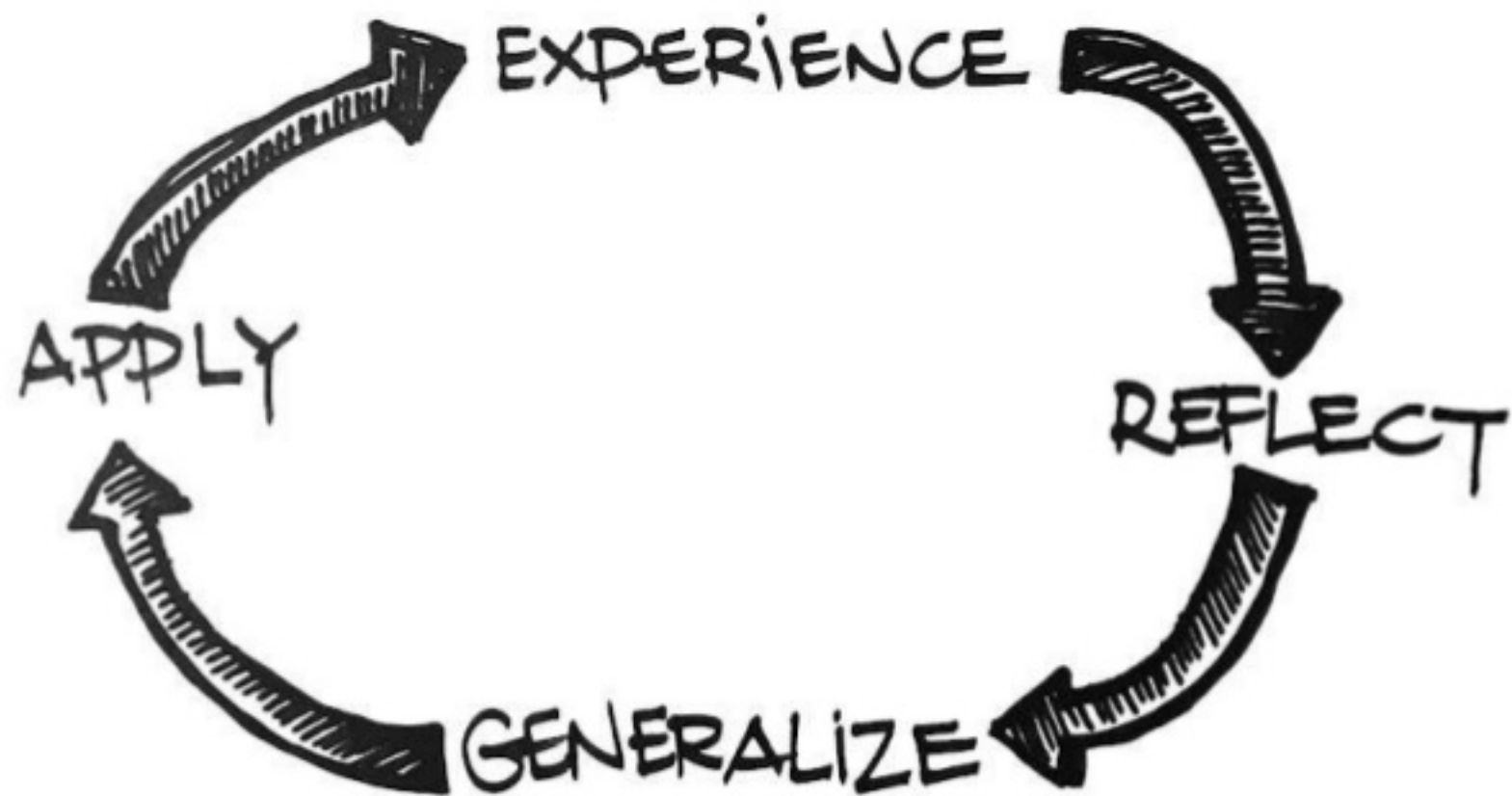
GO  JEK

# Data Pipeline



# Data Aggregation

- Started with one use case - Dynamic Surge Pricing
- Hand coded Flink jobs in 4 weeks
- 20 other use cases in pipeline
- Created a DIY platform



# Daggers

- Generic Flink Job
- Feeds data from Kafka
- Deserializes protobuf messages
- Can process upto 2 streams
- Aggregated data from stream(s) is sent to a sink



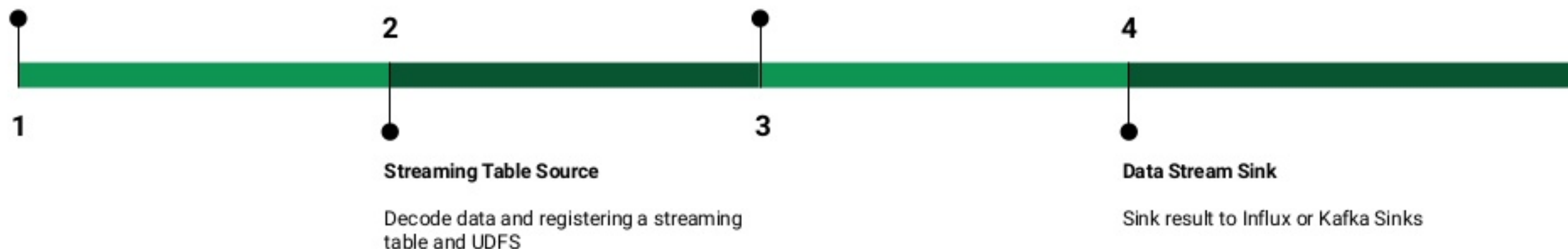
# Dagger Insights

## Kafka Connector

Consume Protobuf encoded data from Kafka

## Flink SQL

Apply SQL and generate result







O



T



OURSEL

F

# Datlantis

- DIY platform
- User friendly interface to a fully automated system
- Creates and deploys DAGGERS
- In minutes using SQL-like syntax

# What it does?

- Flink's Monitoring REST API
- Get status of currently running Jobs
- Create new dagger jobs on flink cluster
- Stop any running Job
- Edit any running Job

# Sample Query

```
1  SELECT city_id AS tag_city_id,  
2         S2Id(lat,long,13) AS tag_s2id,  
3         vehicle_type AS tag_vehicle_type,  
4         count(1) AS num_of_bookings,  
5         TUMBLE_END(rowtime,INTERVAL '60' SECOND) AS window_timestamp,  
6  FROM `data_stream`  
7  WHERE status='COMPLETED'  
8  GROUP BY TUMBLE(rowtime, INTERVAL '60' SECOND),  
9         city_id,  
10        S2Id(lat, long, 13),  
11        vehicle_type  
12
```



# Decentralisation

- Created DAGGERS cluster for different teams
- Provided them with dashboard for basic monitoring
- Added authorisation benefits



DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

# DAGGER CLUSTER LIST

Dagger Cluster Name

Playground

Yarn cluster

http://p-de-daggers-flin...

Flink jar

3d4cdb6e-3b64-48b5-b...

SHOW

Dagger Cluster Name

DE Cluster

Yarn cluster

http://p-de-daggers-flin...

Flink jar

7ff81603-af93-484e-97c...

SHOW

Dagger Cluster Name

Enigma

Yarn cluster

http://p-de-daggers-flin...

Flink jar

9ed9be66-f7d2-426a-b...

SHOW

Dagger Cluster Name

Allocations

Yarn cluster

http://p-de-daggers-flin...

Flink jar

0419f036-b006-4f48-aa...

SHOW

Dagger Cluster Name

Fraud

Yarn cluster

http://p-de-daggers-flin...

Flink jar

e8eb59b7-4cd9-4d67-b...

Dagger Cluster Name

System

Yarn cluster

http://p-de-daggers-flin...

Flink jar

5fa89dc2-2975-4c20-be...

Dagger Cluster Name

Aggregation

Yarn cluster

http://p-de-daggers-flin...

Flink jar

a168400e-566f-4096-94...

Dagger Cluster Name






Security

Yarn cluster

http://p-de-daggers-flin...

Flink jar

f4390752-84b3-4c9c-be...

-  DAGGERS
-  FIREHOSE
-  COLD STORAGE
-  ZEPPELIN
- SETTINGS
  -  KAFKA
  -  DAGGER CLUSTER
  -  GROUPS

DE CLUSTER

DAGGERS LIST

NEW DAGGER

1 2 3 4 5 ... Next » Last »

<div>Dagger Name all_api_health_mehak</div> <div>Topic Name p-logstream-all</div> <div>Table name in Influx all_api_health_mehak_i...</div> <div>Created By rohil.s</div> <div>Status NA</div> <div>DETAILS</div>	<div>Dagger Name booking_status_hodr</div> <div>Topic Name GO_CAR-booking-log,G...</div> <div>Table name in Influx booking_status_hodr_i...</div> <div>Created By prakhar.m</div> <div>Status NA</div> <div>DETAILS</div>	<div>Dagger Name conversion_all_services...</div> <div>Topic Name SAMEDAY_KILAT-booki...</div> <div>Table name in Influx conversion_all_services...</div> <div>Created By prakhar.m</div> <div>Status NA</div> <div>DETAILS</div>	<div>Dagger Name customer_cancelled_ho...</div> <div>Topic Name GO_CAR-booking-log,G...</div> <div>Table name in Influx customer_cancelled_ho...</div> <div>Created By prakhar.m</div> <div>Status NA</div> <div>DETAILS</div>
---	---	---	--

Dagger Name	Dagger Name	Dagger Name	Dagger Name
-------------	-------------	-------------	-------------



DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

## Create a new Dagger

Dagger Name①

Stream Details①

[View all protos](#)

Protobuf Class Name①

Select your proto

Data Stream [multiselect with Cmd/Ctrl]①

kong-logs-proto  
GO\_BIRD\_COMBO-booking-log  
GOKILAT\_SHOP-booking-log  
GO\_CAR-booking-log  
GO\_SHOP-booking-log

Table name①

Consumer Group①

Timestamp field Index①

[ADD ANOTHER STREAM](#)

Dagger sql query

[Read more about Dagger SQL](#)





DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

## Create a new Dagger

Dagger Name①

driver\_count\_demo

Stream Details①

[View all protos](#)

Protobuf Class Name①

com.gojek.esb.driverlocation.DriverLocationLogMessage

Data Stream [multiselect with Cmd/Ctrl]①

driver-location-ping

available-driver-location-ping

sinbin-filtered-driver-location-ping

available-driver-ping-test

available-driver-ping

Table name①

data\_stream\_0

Consumer Group①

dagger\_driver\_count\_demo\_group\_7637

Timestamp field Index①

event\_timestamp

ADD ANOTHER STREAM

Dagger sql query

[Read more about Dagger SQL](#)



DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

## SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

ADD ANOTHER STREAM

Dagger sql query

[Read more about Dagger SQL](#)

```
SELECT DistinctCount(driver_id) as drivers , S2Id(latitude, longitude, 13) as  
label_s2id, driver_status as label_driver_status, TUMBLE_END(rowtime,  
INTERVAL '60' SECOND) AS window_timestamp from `supply_data` GROUP  
BY TUMBLE ( rowtime, INTERVAL '60' SECOND ), S2Id(latitude, longitude, 13),  
driver_status
```



## Advanced Options

Watermark interval in ms<sup>①</sup>Watermark delay in ms<sup>①</sup>Table name in Influx<sup>①</sup>Parallelism<sup>①</sup>

PREVIEW



DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

## SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

## DRIVER\_COUNT\_DEMO

INFLUX DB: DAGGERSTEST

INFLUX MEASUREMENT: driver\_count\_demo\_influx\_3876

CREATED BY: PRAKHAR.M

STOP DAGGER

PUBLISH TO KAFKA

BACK



We are creating your dagger. Please wait for 3-4 minutes to start taking your data driven decisions !!!



DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

## DRIVER\_COUNT\_DEMO

INFLUX DB: DAGGERSTEST

INFLUX MEASUREMENT: driver\_count\_demo\_influx\_3876

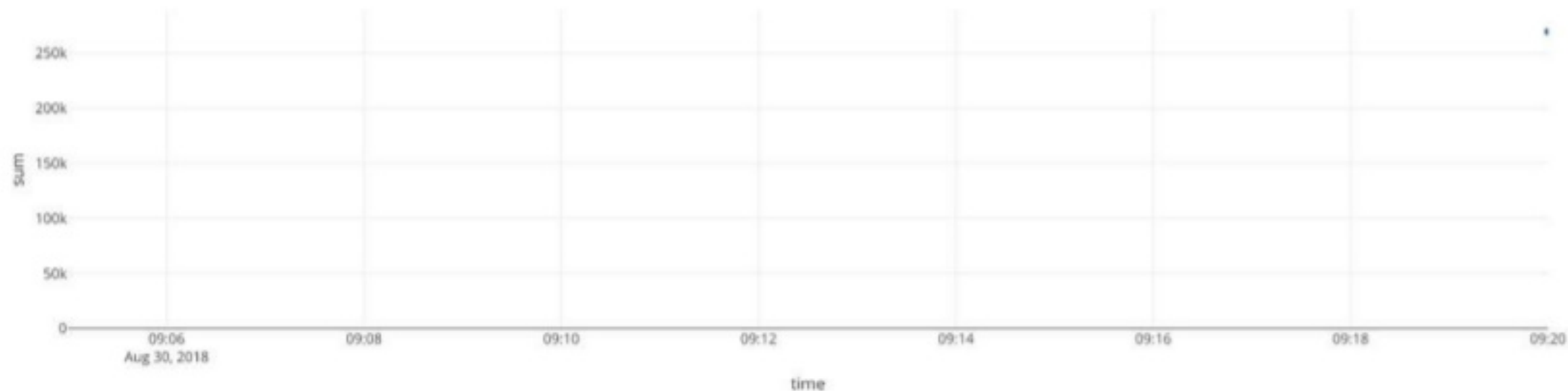
CREATED BY: PRAKHAR.M

STOP DAGGER

PUBLISH TO KAFKA

BACK

```
SELECT sum(*) FROM driver_count_demo_influx_3876 WHERE time > now() - 900s GROUP BY time(30s) fill(null)
```







DAGGERS



FIREHOSE



COLD STORAGE



ZEPPELIN

## SETTINGS



KAFKA



DAGGER CLUSTER



GROUPS

## DRIVER\_COUNT\_DEMO

INFLUX DB: DAGGERSTEST

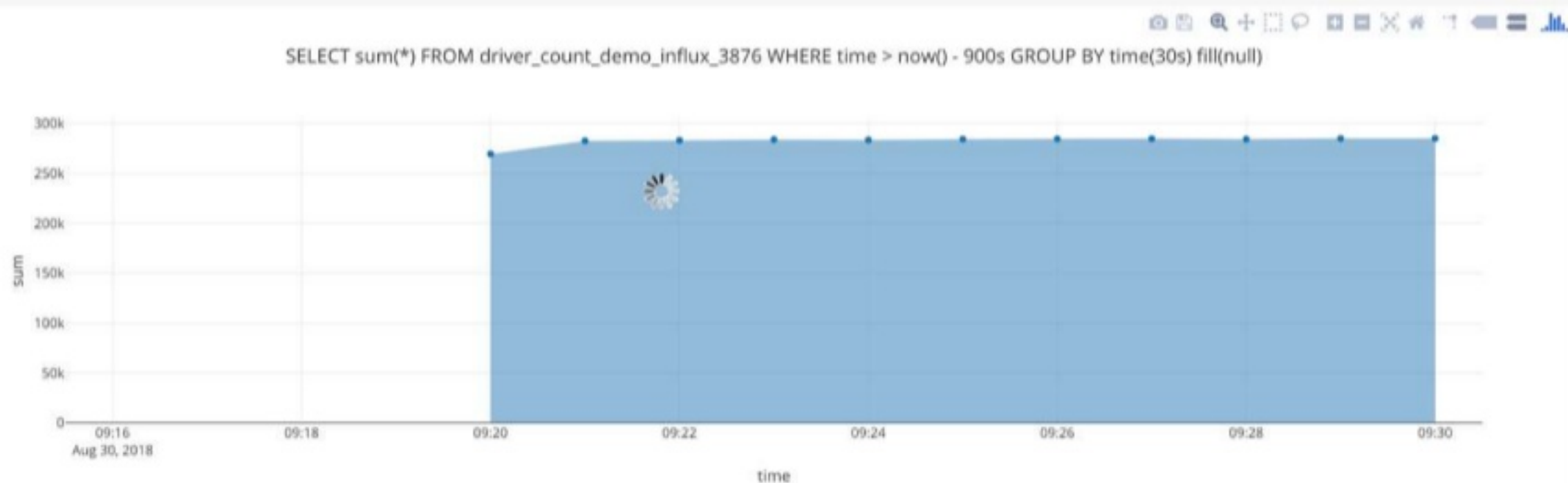
INFLUX MEASUREMENT: driver\_count\_demo\_influx\_3876

CREATED BY: PRAKHAR.M

STOP DAGGER

PUBLISH TO KAFKA

BACK



# Data Sinks

- Time series
- Kafka

# Time Series Sink

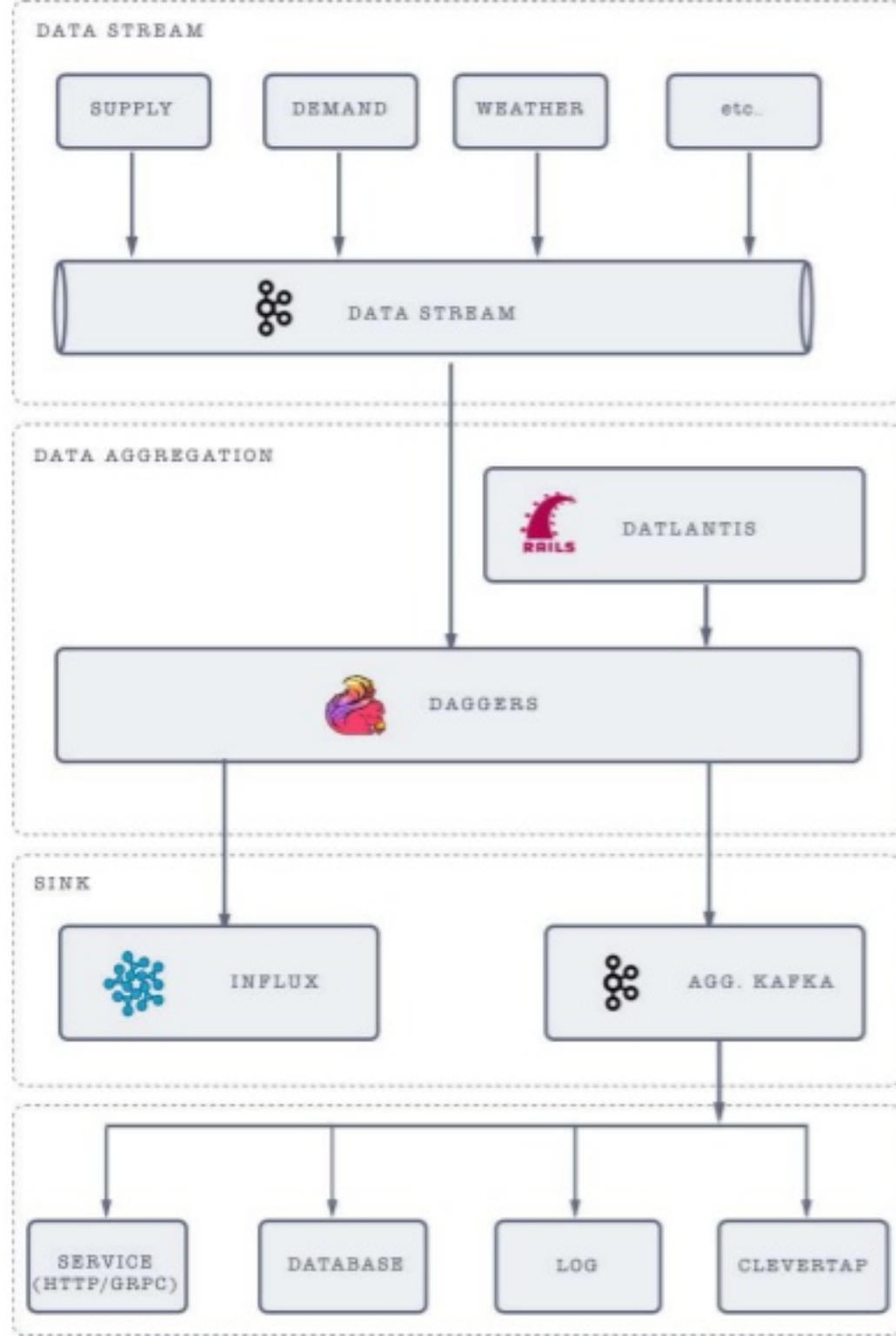
- Preview mode
- Default data sink
- Integrated with grafana - used for monitoring & alerting





# Kafka Sink

- Publish to Kafka topic
- Another DIY tool to sink from Kafka to one of the following:
  - Services - HTTP or GRPC
  - DB - relational OR time series
  - Analytics platforms - Clevertap or Mixpanel
  - Log - for debugging



# Architecture

# Atlas

- Geospatial Visualisation Platform
- Maps to actionable insights

MAP STYLE

SATELLITE

MAP TYPE

CHOROPLETH

SERVICE AREA

JAKARTA

SERVICE TYPE

ALL

COLOR

SELECT...

ELEVATION

SELECT...





# 1 K+

## REAL TIME DAGGERS

- Spanned over 6 Flink Cluster
- Most of it created by analysts
- Actively used for monitoring
- Dashboards created are used by city heads

# 2 min

## TO PRODUCTION

- Single Form to create DAGGER
- The data can be sent to a sink
- Data ready to be consumed as soon as generated

# 1+ TB

## DATA PROCESSED EVERYDAY

- Real time data analysis across all cluster
- Processed data is sent to one of the sinks

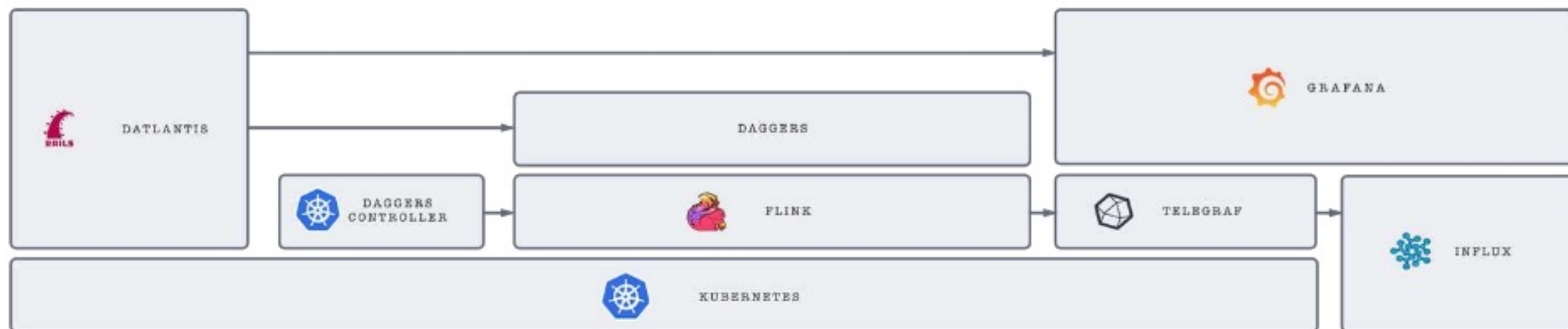
# Deployment

- We have Flink Clusters on Yarn and Kubernetes
- Checkpointing - HDFS and Google Cloud Storage
- Dagger Kubernetes controller -
  - Job JAR is available on Flink cluster
  - Scales cluster when more slots needed

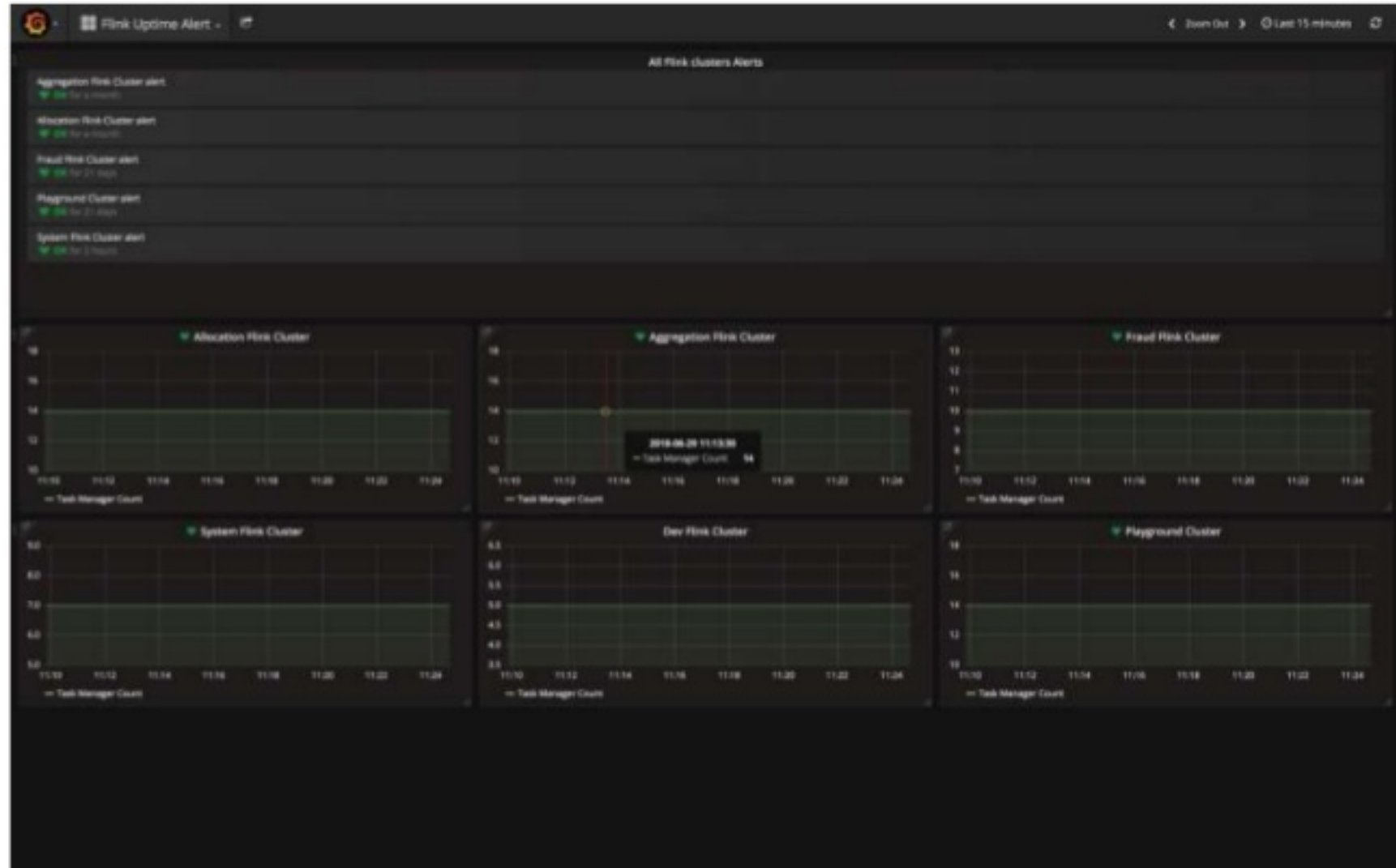


# Deployment

## DEPLOYMENT



# Monitoring





# Alerting

- Automated alerts from Datlantis
- Users are provided with a Health dashboard
- Alerts are sent to specific teams via their slack channels and pager duties

# Impact

01	5+ Billion Messages/day	<ul style="list-style-type: none"><li>• For system uptime</li><li>• Across 500 microservices</li></ul>
02	44,000 geolocation	<ul style="list-style-type: none"><li>• For dynamic surge pricing</li><li>• Demand &amp; supply</li></ul>
03	25+ Metrics	<ul style="list-style-type: none"><li>• For allocation metrics</li><li>• Created &amp; maintained by analysts</li></ul>
04	User segmentation & Real-time triggers	<ul style="list-style-type: none"><li>• For growth campaign</li><li>• 26% better conversion</li></ul>

# Let's talk !

Prakhar Mathur  
Medium : @prakharmathur\_345

Rohil Surana  
Medium : @rohilsurana