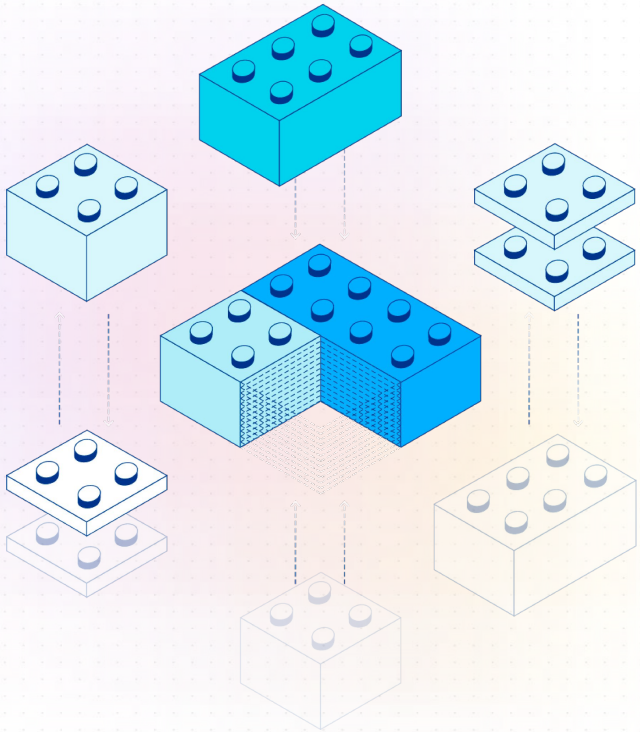
Open DataOps Foundation

Feature engineering at scale with Dagger and Feast



Ravi Suhag

https://github.com/odpf

Open DataOps Foundation

Ravi Suhag

VP Engineering, Gojek

Founder and Principal Architect,
Open DataOps Foundation

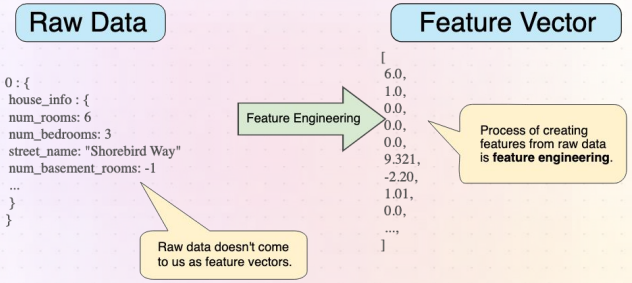
www.ravisuhag.com

https://github.com/odpf

Open DataOps Foundation

Feature Engineering

The process of transforming raw data into high quality input signal for models that better represent the underlying problem to the models.

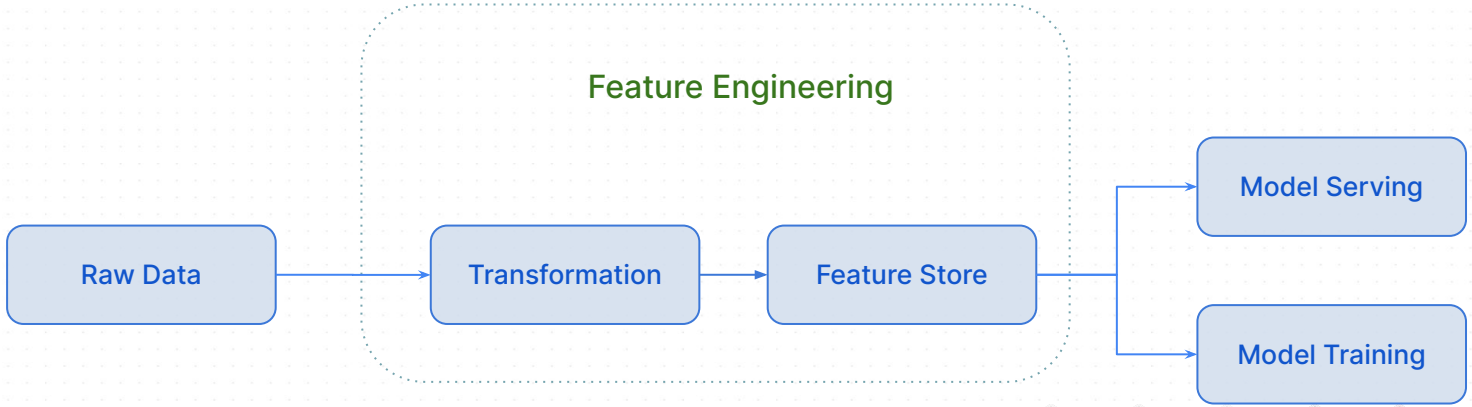


https://github.com/odpf

Picture Credits: GCP

Feature Engineering in ML Pipeline

Open DataOps Foundation



https://github.com/odpf

Open DataOps Foundation

Need for Feature Engineering

Better features means flexibility

Good data structure allows for better flexibility in choosing models

Better features means simpler models.

Easier to pick right parameters and models

Better features means better results.

Quality of model results is dependent on quality of features

https://github.com/odpf

Feature Engineering Challenges

- Inconsistency between training and serving
- Managing data pipelines
- Scaling data infrastructure
- Lack of standardization
- Real-time features required skilled data engineers

Feature Engineering Platform Goals

- Unified processing for streaming and batch
- Self-service platform
- Elastic infrastructure
- Standard and reusable way for feature engineering
- No added skill needed for real-time features.

Open DataOps Foundation

Dagger

Stream processing made easy

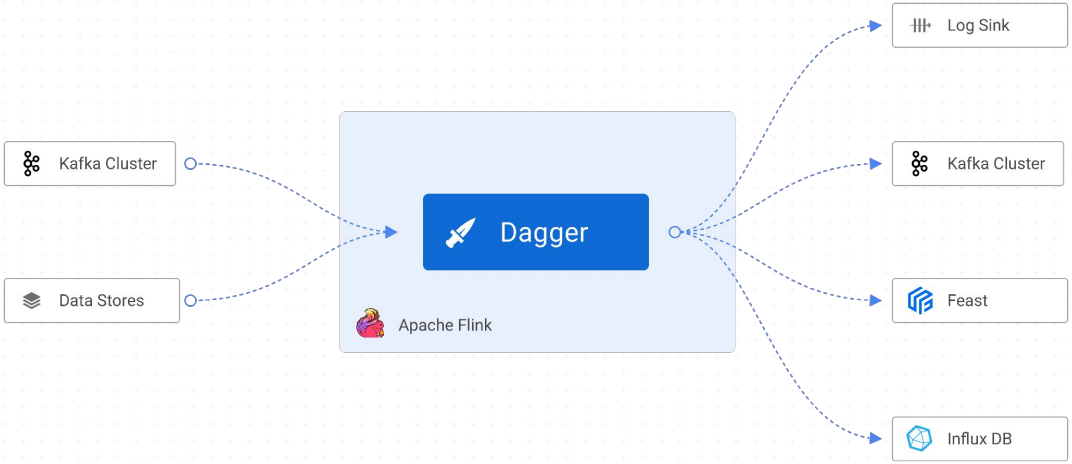
Open source stream processing framework for transforming, aggregating and enriching data with ease of operation and reliability.

https://github.com/odpf

Open DataOps Foundation

Dagger

Configuration over code, cloud-native framework built on top of Apache Flink for stateful processing of data.



https://github.com/odpf

Open DataOps Foundation

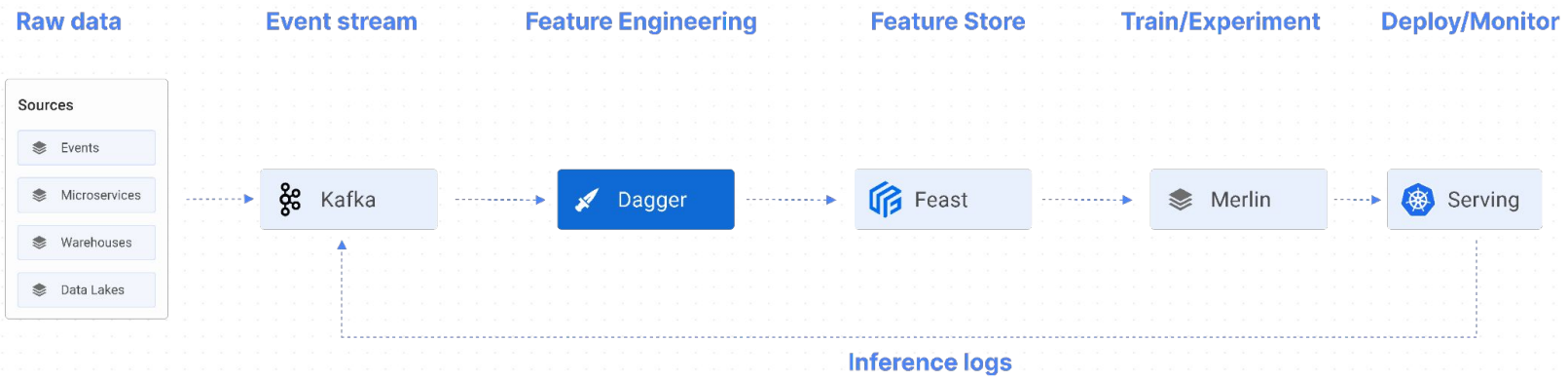
FEAST

Feature store for machine learning

Feast is an open-source feature store. It is the fastest path to operationalizing analytic data for model training and online inference.

ML pipeline with Dagger and Feast

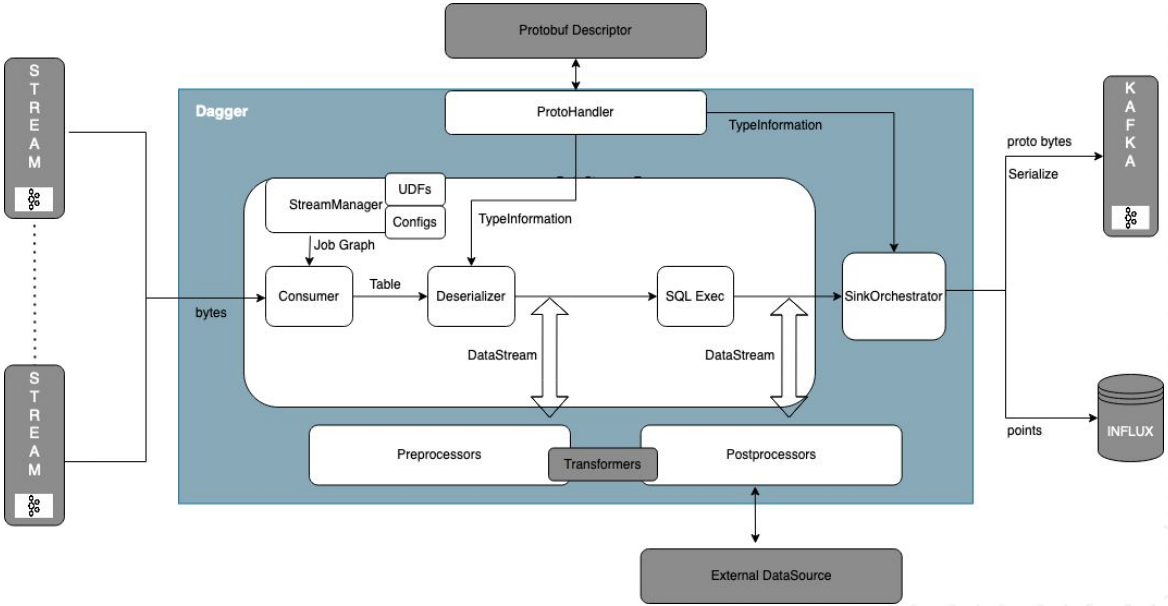
Open DataOps Foundation



https://github.com/odpf

Dagger Architecture

Open DataOps Foundation



https://github.com/odpf

Open DataOps Foundation

Dagger Key Features

SQL First

Query writing made easy through formatting, suggestions, auto-completes and template queries.

Flexibility

Add custom business logic in form of plugins with UDFs, Transformers, Preprocessors and Post Processors.

Stream Enrichment

Enrich Kafka messages from API endpoints or database sources to bring offline & reference data context to real-time processing.

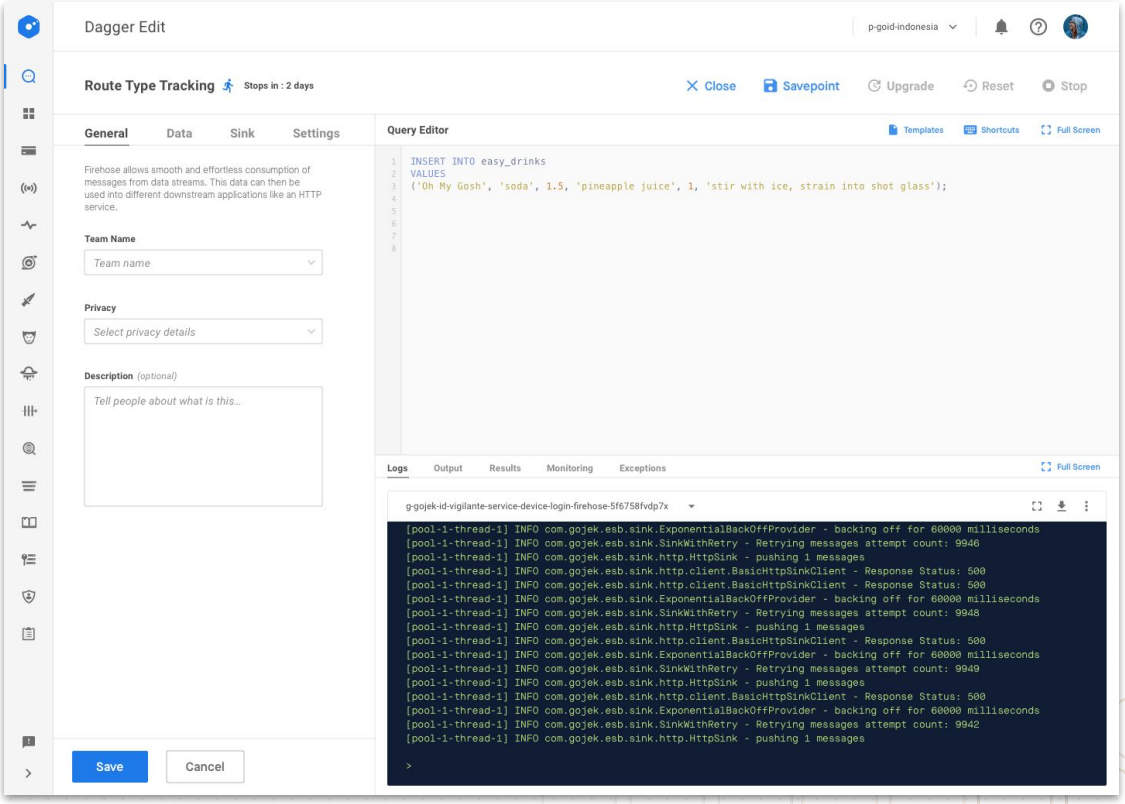
Learn more: https://odpf.github.io/dagger/

https://github.com/odpf

Open DataOps Foundation

Self service platform

Data Scientists can create and manage their feature engineering jobs through a complete self-service platform.



Use below link for more details on case study: www.p4suhag.com

https://github.com/odpf

Open DataOps Foundation

Managing with GitOps

Data Scientists can specify YAML specifications to create Dagger. This allows them to version control, use GitOps for managing feature engineering pipelines.

CODE EDITOR

```yaml
kind: daggerJob
description: demand (unique customers) for s2id level 11
entity: gojek
flink_name: p-godata-id-ds-marketplace
privacy: public
sink_type: kafka
configuration:
  FLINK_PARALLELISM: 1
  FLINK_SQL_QUERY: |-
    SELECT * FROM `data_streams_0`
  FLINK_WATERMARK_DELAY_MS: '4000'
  FLINK_WATERMARK_INTERVAL_MS: '60000'
  PROCESSOR_POSTPROCESSOR_ENABLE: false
  SINK_KAFKA_TOPIC: booking-S2L11-EW1m-agg
  SINK_TYPE: kafka
  …
  …
```

https://github.com/odpf

Open DataOps Foundation

SQL first

Dagger is built keeping SQL first philosophy in mind.

SQL is not “Turing complete” in a key way: it always terminates. Which makes it **less likely** to monopolize all the compute power in a data center.

```
SELECT
  api_name AS api_name,
  api_uri AS api_uri,
  api_method AS api_method,
  Cast(http_status_code AS BIGINT) as http_status_code,
  count(1) as request_count,
  Tumble_end(rowtime, INTERVAL '60' second) AS event_timestamp
FROM
  `api_logs`
GROUP BY
  api_name,
  api_uri,
  http_status_code,
  api_method,
  Tumble (rowtime, INTERVAL '60' second)
```

Open DataOps Foundation

UDFs

Allows custom business logic with User Defined Functions written in Python or Java for advanced use cases.

Example:

```
GeoHash(Double latitude, Double
longitude, int level)
```

Returns a geohash for a given level and lat-long for the given WGS84 point.

CODE EDITOR

```
SELECT
  data1_location.longitude AS long,
  data1_location.latitude AS lat,
  GeoHash(
    data1_location.longitude,
    data1_location.latitude,
    6
  ) AS geohashPickup,
  TUMBLE_END(rowtime, INTERVAL '60' SECOND) AS window_timestamp
FROM
  data_stream
GROUP BY
  TUMBLE (rowtime, INTERVAL '60' SECOND),
  data1_location.longitude,
  data1_location.latitude
```

https://github.com/odpf

Open DataOps Foundation

Data masking

Enables encryption on a set of fields as configured.

Used in data forwarding to clone production data to integration and local environments for model training with encryption on sensitive data fields.

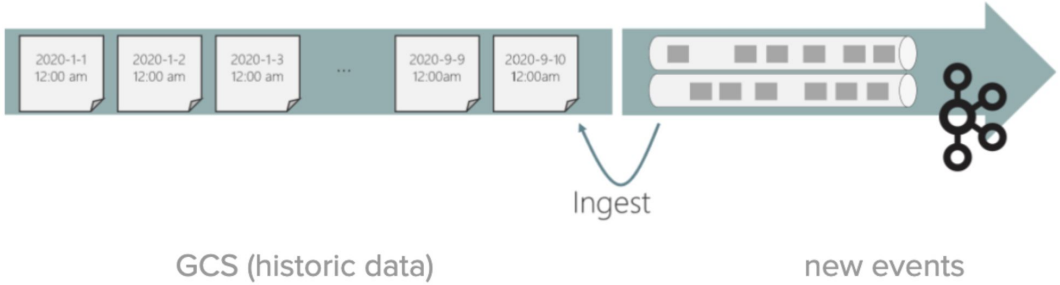
CODE EDITOR

```
SELECT
  event_timestamp,
  test_data
FROM
  data_stream
************* Post processor config: ***************
{
  "internal_source": [{
      "output_field": "*",
      "value": "*",
      "type": "sql"}],
  "transformers": [{
    "Transformation_class": "io.odpf.dagger.functions.transformers.HashTransfor
      "transformation_arguments": {
        "maskColumns": ["test_data.data1"]
      }}]
}
```

https://github.com/odpf

Open DataOps Foundation

Hybrid data source



- Consume data from multiple sources.
- Auto switch from batch source to stream source.

https://github.com/odpf

Open DataOps Foundation

Backfill with hybrid source

Unified processing across batch and stream data sources.

Backfill historic data for model training.

Join data across multiple streams for complex use cases.

CODE EDITOR

```
[{
   "INPUT_SCHEMA_TABLE": "booking_log_stream",
   "SOURCE_KAFKA_TOPIC_NAMES": "go-food-booking-log|go-ride-booking-log",
   "INPUT_SCHEMA_PROTO_CLASS": "com.esb.proto.BookingLogMessage",
   "INPUT_SCHEMA_EVENT_TIMESTAMP_FIELD_INDEX": "5",
   "SOURCE_KAFKA_CONSUMER_CONFIG_BOOTSTRAP_SERVERS": "localhost:9092,
localhost:7336, localhost:6262",
   "SOURCE_KAFKA_CONSUMER_CONFIG_GROUP_ID": "dummy-consumer-group",
   "SOURCE_PARQUET_UNREADABLE_FILE_BEHAVIOUR": "FAIL_WITH_EXCEPTION",
   "SOURCE_PARQUET_FILE_PATHS": [
     "gs://p-godata-id-mainstream-bedrock/GO_FOOD-booking-log/dt=2022-01-23/",
     "gs://p-godata-id-mainstream-bedrock/GO_RIDE-booking-log/dt=2021-01-23/"
   ],
   "SOURCE_PARQUET_READ_ORDER_STRATEGY": "EARLIEST_TIME_URL_FIRST",
   "SOURCE_PARQUET_SCHEMA_MATCH_STRATEGY": "BACKWARD_COMPATIBLE_SCHEMA",
   "SOURCE_DETAILS": [
      {"SOURCE_TYPE": "BOUNDED", "SOURCE_NAME": "PARQUET_SOURCE"},
      {"SOURCE_TYPE": "UNBOUNDED","SOURCE_NAME": "KAFKA_SOURCE"}
   ]
}]
```

https://github.com/odpf

Open DataOps Foundation

Stream enrichment

Use external data sources to fetch additional information about each event in processing.

Allows to use any external Data sources, Service endpoints, Object stores, Cache for enriching your stream.

Learn more:
https://odpf.github.io/dagger/docs/usecase/stream_enrichment

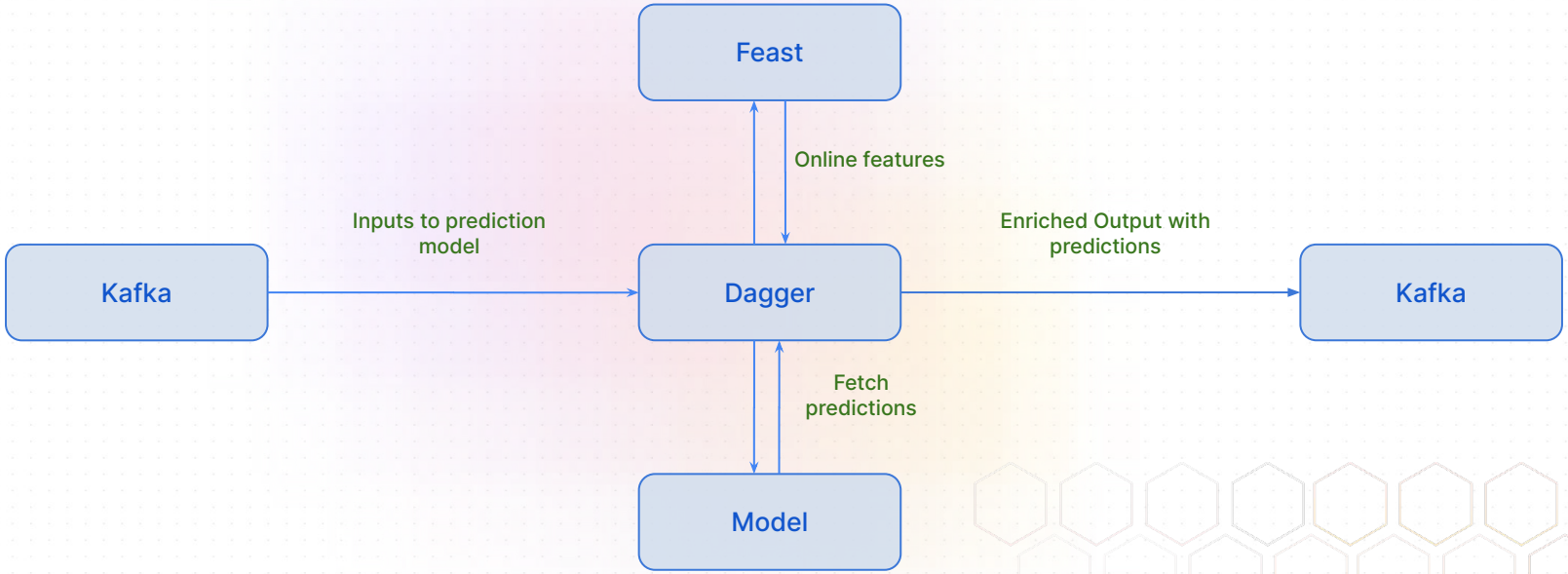
CODE EDITOR

```
[{"internal_source": [{
      "output_field": "booking_log",
      "type": "sql",
      "value": "*"
    }]},
{"external_source":{
      "es":[{
            "host":"127.0.0.1",
            "port":"9200",
            "endpoint_pattern":"/customers/customer/%s",
            "endpoint_variables":"customer_id",
            "stream_timeout":"5000",
            "connect_timeout":"5000",
            "capacity":"30",
            "output_mapping":{
               "customer_profile":{"path":"$._source"}
     }}]
 }}]
```

https://github.com/odpf

Stream inference

Open DataOps Foundation



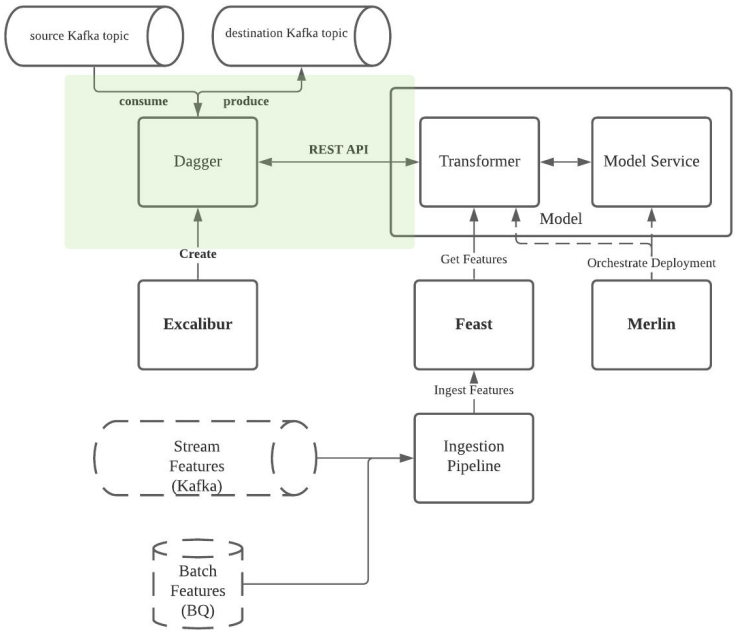
https://github.com/odpf

Open DataOps Foundation

Stream Inference

Stream inference allows real time predictions where multiple users might want to consume predictions of a single model.

Prediction logs can be stored for quality and monitoring purposes



https://github.com/odpf

Open DataOps Foundation

Stream Inference

Filtering and custom field mapping during stream inference.

Model Endpoint *

http://sklearn-3.sample.models.id.d.gods.golabs.io/v1/predict

Model Type *

Arbitrary

Request Payload Mapping *

```
SOURCE FIELD receipt_url AS <alias>
SOURCE FIELD service_type AS <alias>
SOURCE FIELD status AS <alias>
SOURCE FIELD order_number AS <alias>
SOURCE FIELD service_area_id AS <alias>
SOURCE FIELD driver_id AS <alias>
SOURCE FIELD <field>
```

```
{
  "receipt_url": "${receipt_url}",
  "service_type": "${service_type}",
  "status": "${status}",
  "order_number": "${order_number}",
  "service_area_id": "${service_area_id}",
  "driver_id": "${driver_id}"
}
```

Filter

```
WHERE
e.g. status = 'CREATED'
```

Specify the filter for queries from Dagger, if applicable

Resulting Type Mapping *

```
SOURCE FIELD service_type AS SINK FIELD service_type
SOURCE FIELD order_number AS SINK FIELD order_number
SOURCE FIELD receipt_url AS SINK FIELD receipt_url
FUNCTION CURRENT_TIMESTAMP AS SINK FIELD event_timestamp
SOURCE FIELD driver_id AS SINK FIELD driver_id
SOURCE FIELD service_area_id AS SINK FIELD service_area_id
SOURCE FIELD status AS SINK FIELD status
JSON PATH $.prediction_score AS SINK FIELD prediction_score
JSON PATH <$.predictions[0]> AS SINK FIELD <field>
```

https://github.com/odpf

Dagger adoption at Gojek

300+

Dagger jobs for feature engineering.

50+

Data scientists creating Dagger jobs.

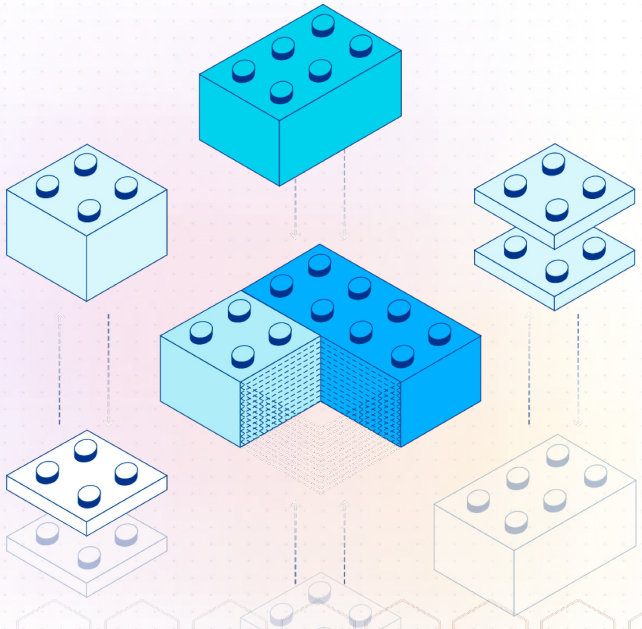
10+ TB

Data processed each day.

Open DataOps Foundation

Dagger is part of
Open Data Ops Foundation

ODPF is a modern DataOps platform that empowers organizations to discover, transform, analyse and secure data faster and efficiently.



https://github.com/odpf

Open DataOps Foundation

An experience-first approach

Discover

Search through large amount of data across the organization.

Understand

Get contextual knowledge with lineage, quality and other aspects of data.

Operate

Process, wrangle, transform or analyse data as per your needs.

Apply

Drive business value with data, models and insights.

https://github.com/odpf

Open DataOps Foundation

A fully-integrated suite of data products

A fully-integrated suite of open-source products that are required to build an end-to-end data platform for all your needs from ingestion to insights.

It also provides products for data management plane ranging across infrastructure orchestration, observability, security, access control and data catalog.

https://github.com/odpf

Open DataOps Foundation

DATA LIFECYCLE

Raccoon
Data ingestion

Dagger
Stream processsing

Firehose
Data loading

Optimus
Data transformation

Enigma
Operational analytics

DATA MANAGEMENT

Compass
Data catalog

Meteor
Metadata collection

Guardian
Data access control

Shield
Identity management

Predator
Data observability

DATA OPERATIONS

Entropy
Orchestration

Siren
Site reliability

Stencil
Schema registry

https://github.com/odpf

Open DataOps Foundation

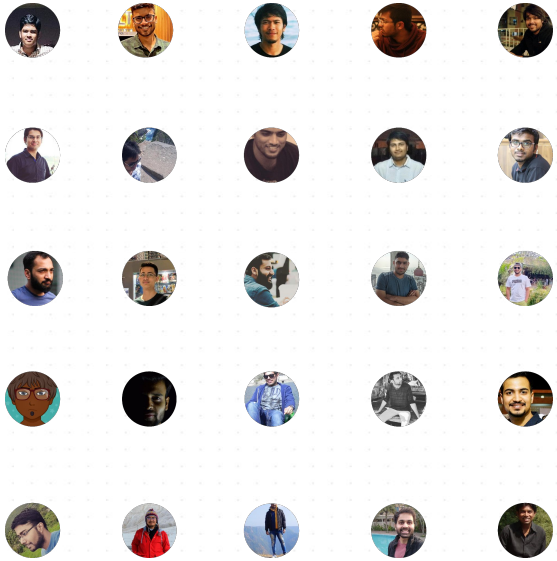
Powering data platform for large-scale data teams

gojek midtrans mapan moka

https://github.com/odpf

Open DataOps Foundation

Active community



200+

Contributors.

With 80% growth year over year

2000+

Commits last year.

With 39% growth year over year

1000+

Community members.

Across Github and Slack

https://github.com/odpf

Open DataOps Foundation

Get involved

Explore and contribute to ODPF data platform on Github.

https://github.com/odpf

Join the community on Slack and talk to maintainers.

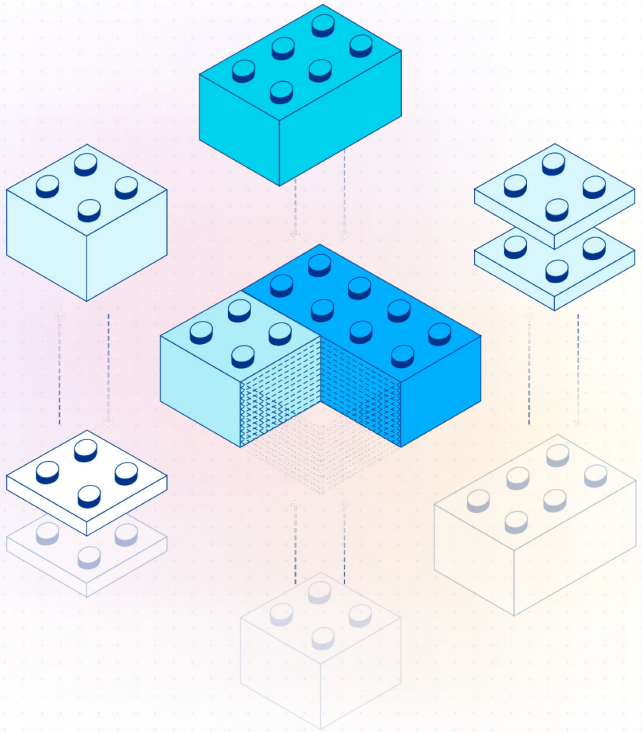
https://bit.ly/2RzPbtn

odpf.io

https://github.com/odpf

Open DataOps Foundation

Thank you



Email: suhag.ravi@gmail.com

Twitter: Ravi_Suhag

Github: ravisuhag

Website: www.ravisuhag.com

https://github.com/odpf