Beam Portability

Federico Patota

Cloud Consultant, Google Cloud

Agenda

Course Intro

Beam and Dataflow Refresher

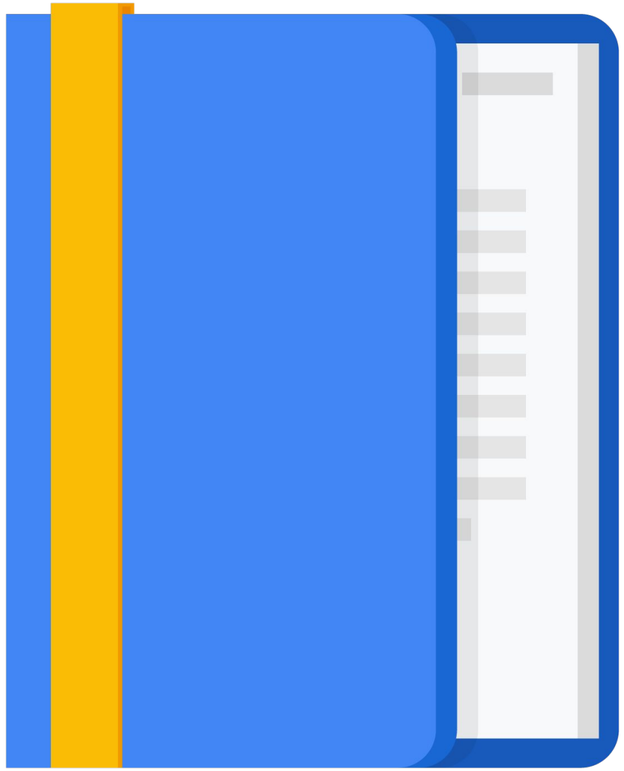
Beam Portability

Separating Compute and Storage

IAM, Quotas, and Permissions

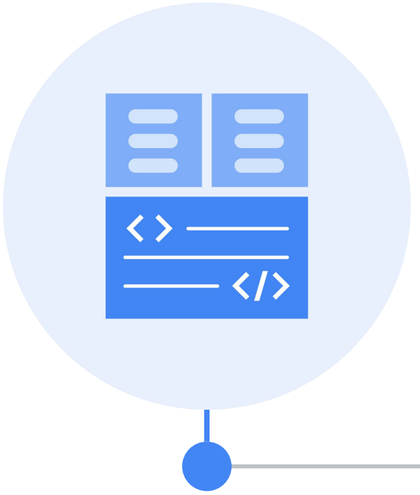
Security

Summary



Beam portability

Agenda



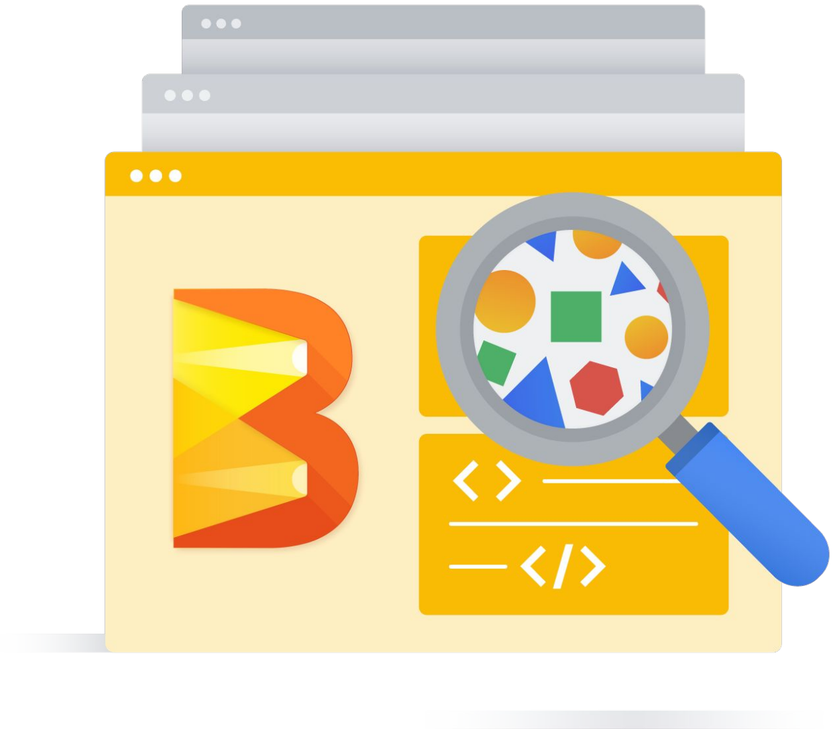
Beam portability

Runner v2

Container environments

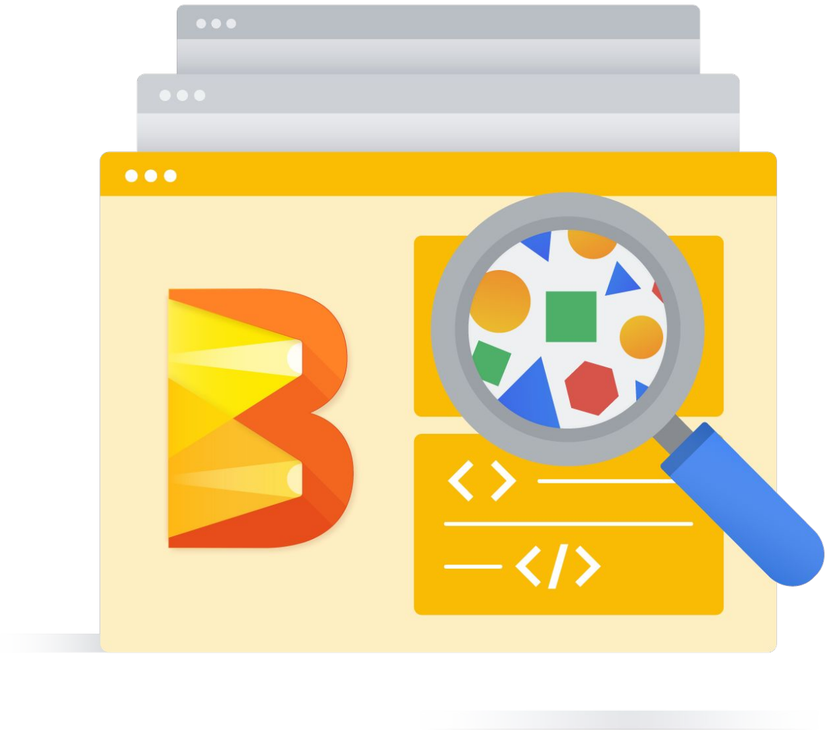
Cross-language transforms

The Beam vision

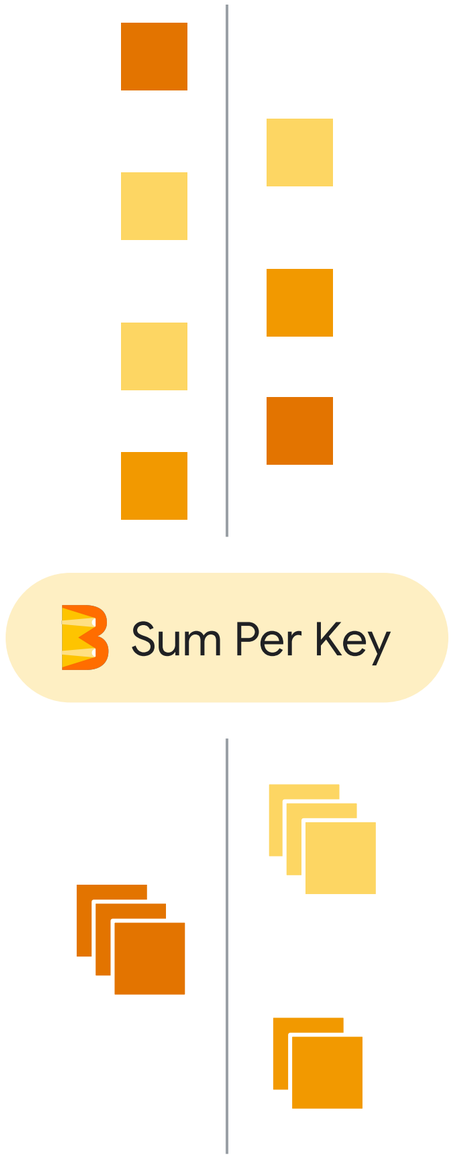


The Beam vision

Provide a **comprehensive portability framework** for data processing pipelines; one that allows you to write your pipeline once in the **programming language of choice** and run it, with minimal effort, on the **execution engine of choice**.



The Beam vision



The Beam vision

Java

Input.apply
(Sum.integersPerKey())

Python

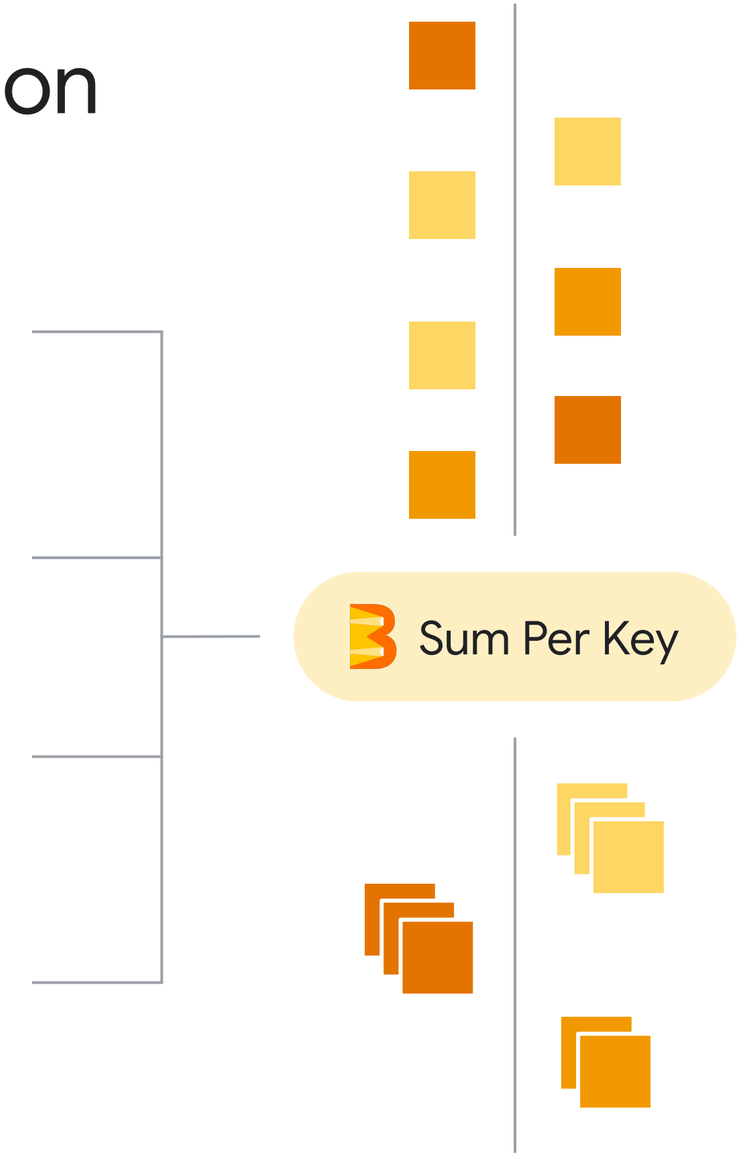
input | Sum.PerKey()

Go

stats.Sum(s, input)

SQL

SELECT key, SUM(value)
FROM input GROUP BY key



The Beam vision

Java

Input.apply
(Sum.integersPerKey())

Python

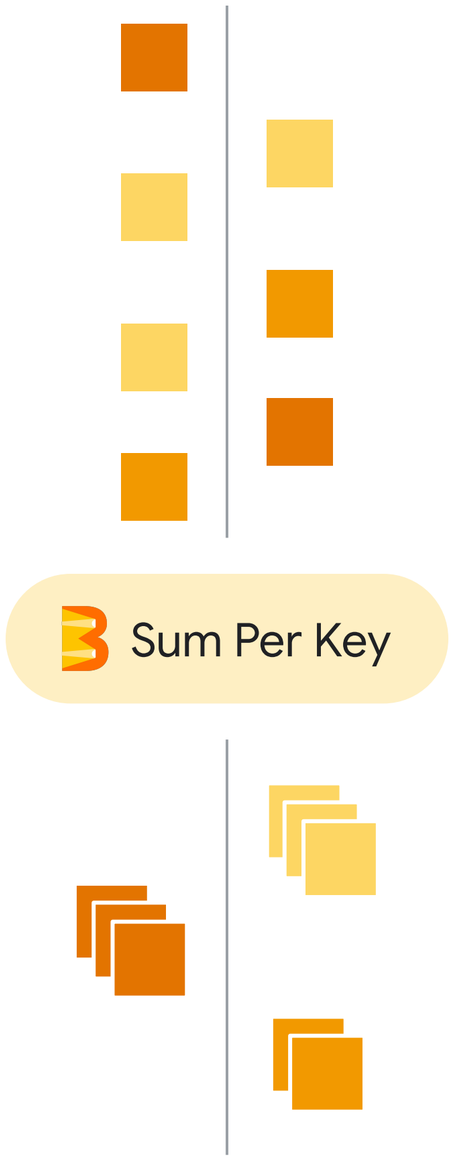
input | Sum.PerKey()

Go

stats.Sum(s, input)

SQL

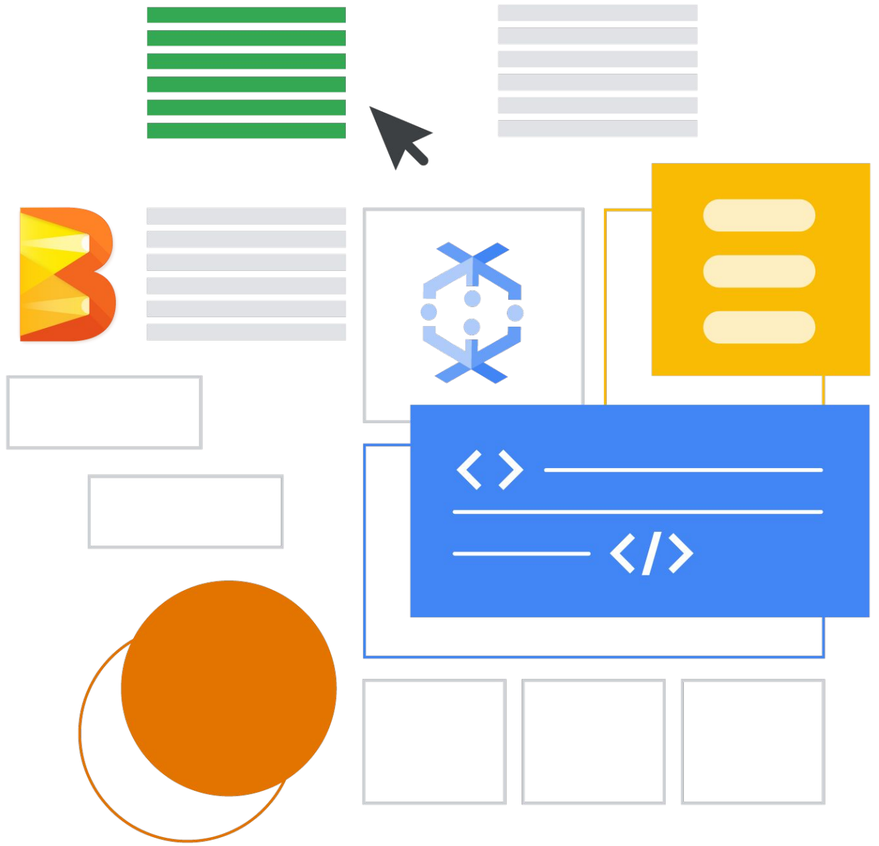
SELECT key, SUM(value)
FROM input GROUP BY key



- Dataflow
- Apache Flink
- Apache Spark
- Apache Apex
- Gearpump
- IBM Streams
- Apache Samza
- Apache Nemo (Incubating)

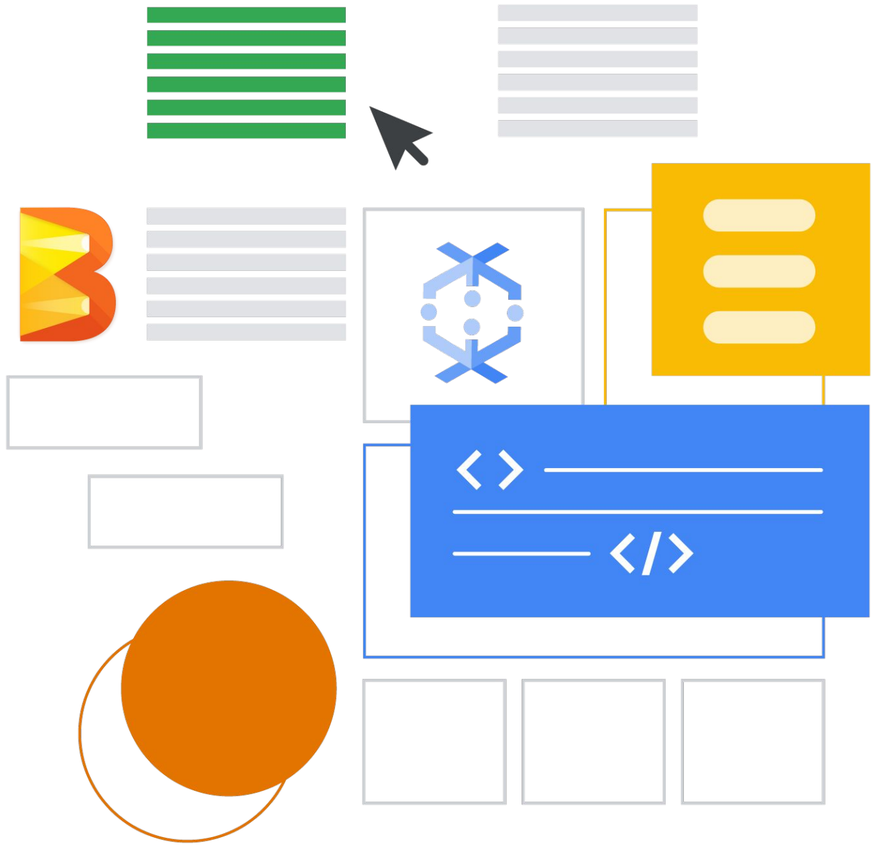
Portability framework

- **Language-agnostic** for representation and **protocol** for execution



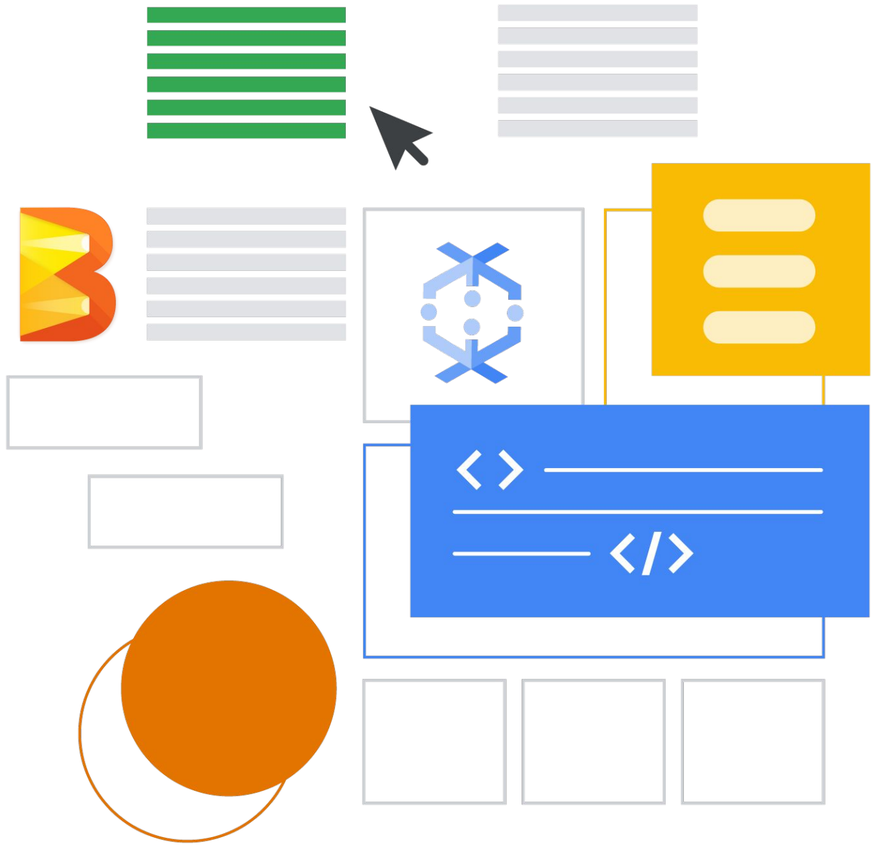
Portability framework

- **Language-agnostic** for representation and **protocol** for execution
- Interoperability layer = **Portability API**

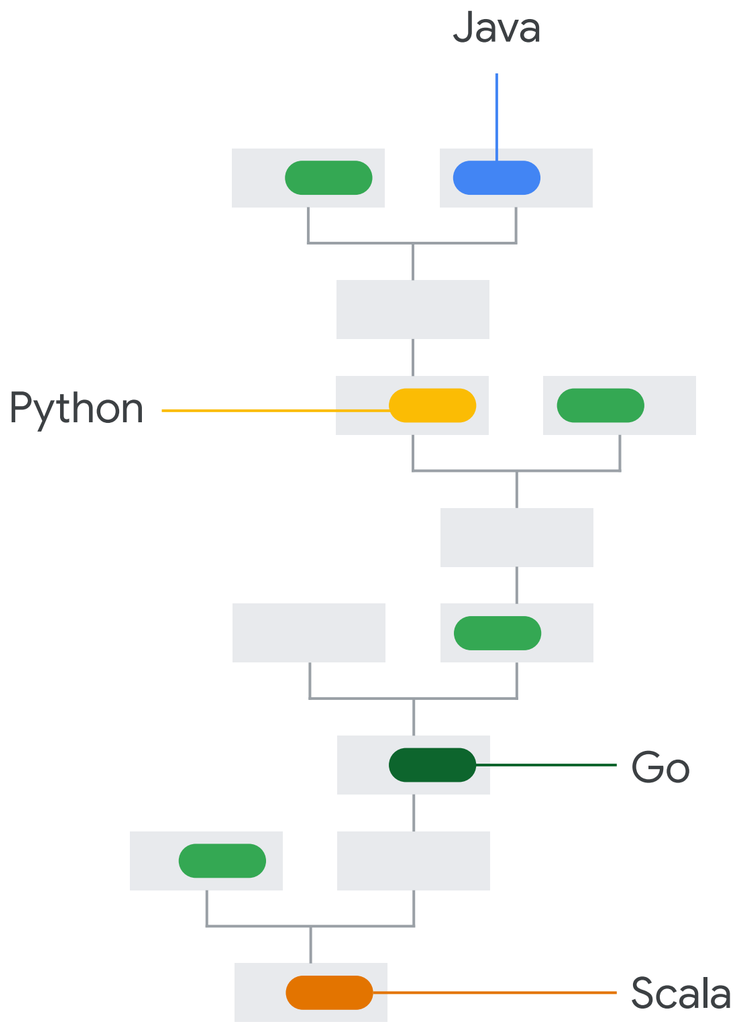


Portability framework

- **Language-agnostic** for representation and **protocol** for execution
- Interoperability layer = **Portability API**
- **Docker containerization** to customize your execution environment

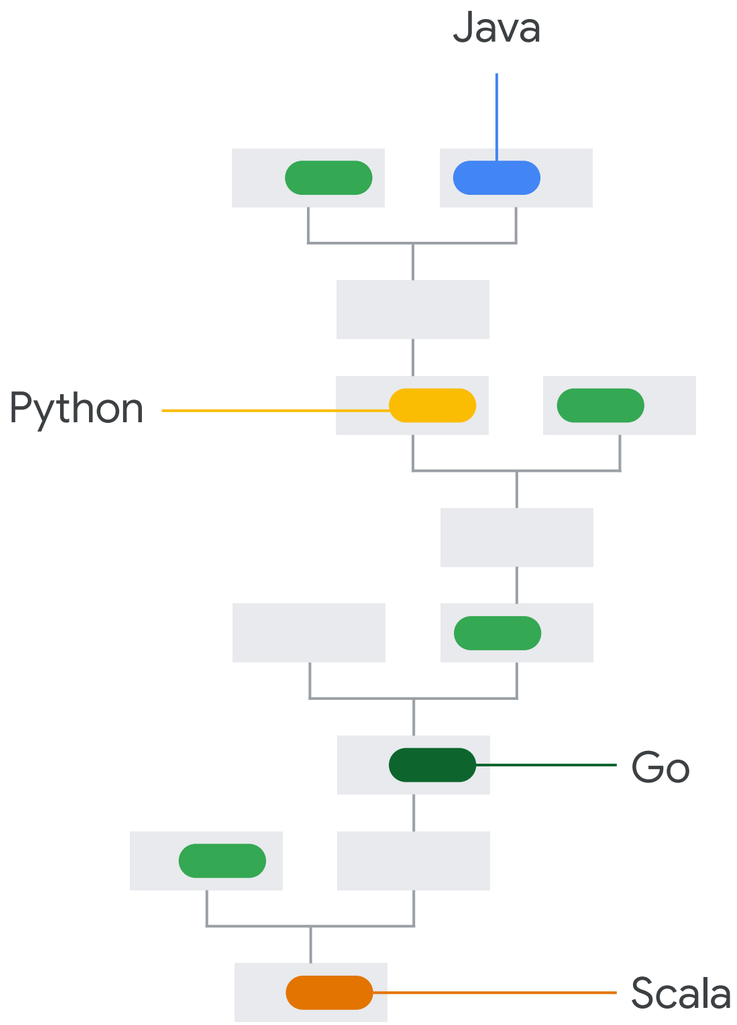


Benefits of portability



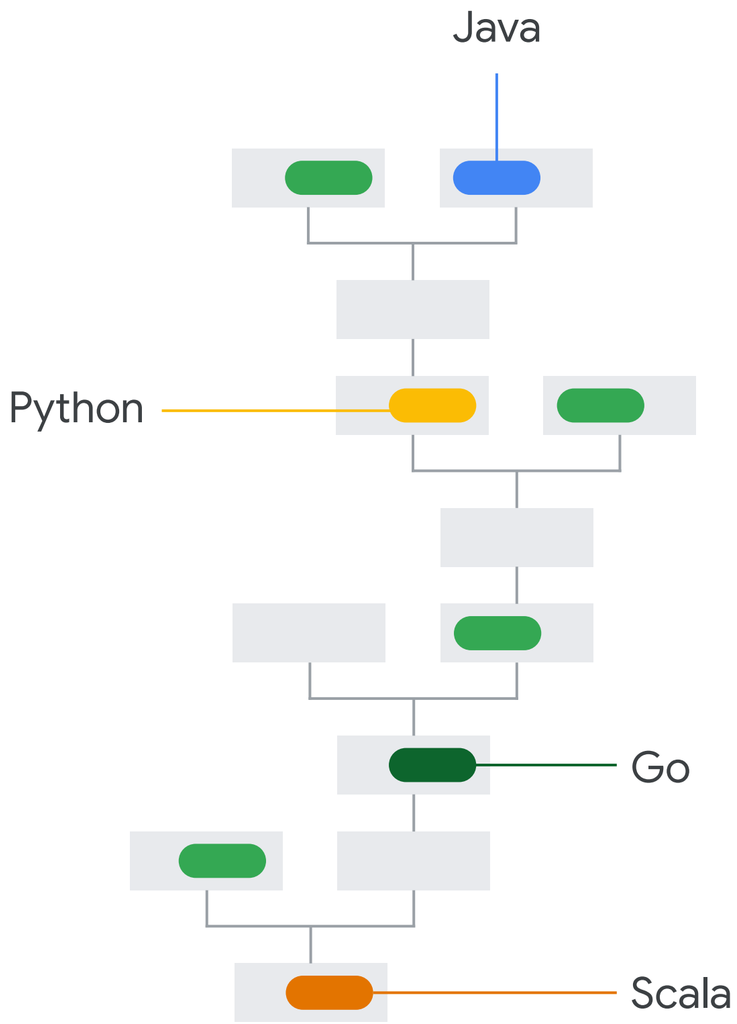
Benefits of portability

- **Every runner** works with **every language**.



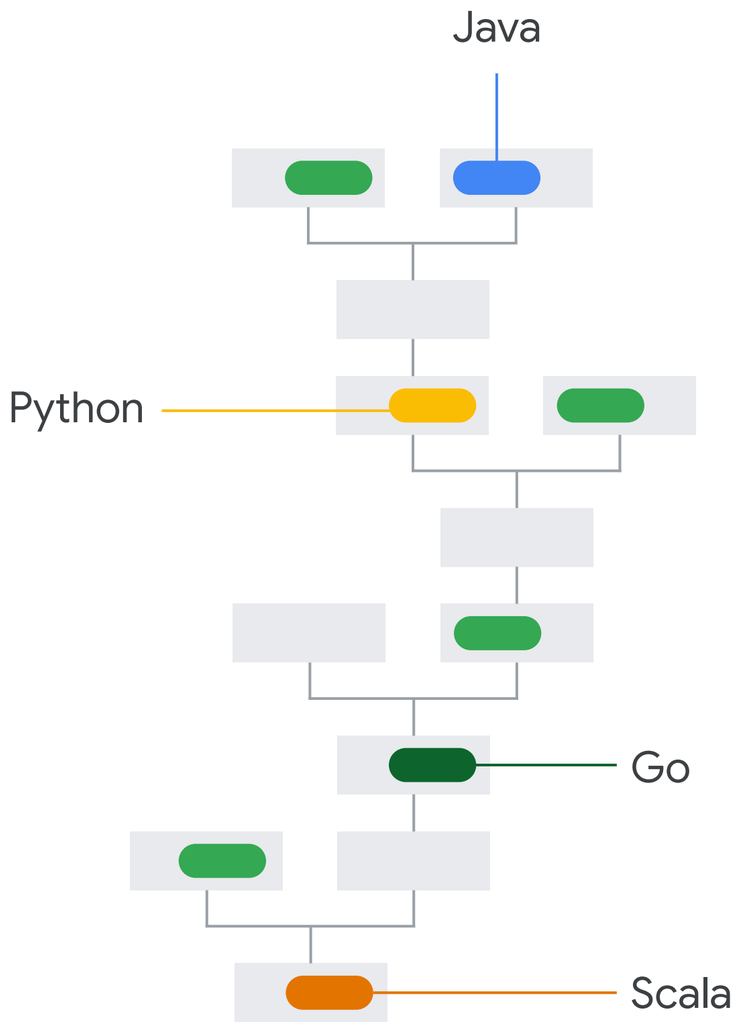
Benefits of portability

- **Every runner** works with **every language**
- Configurable, hermetic **worker environment**



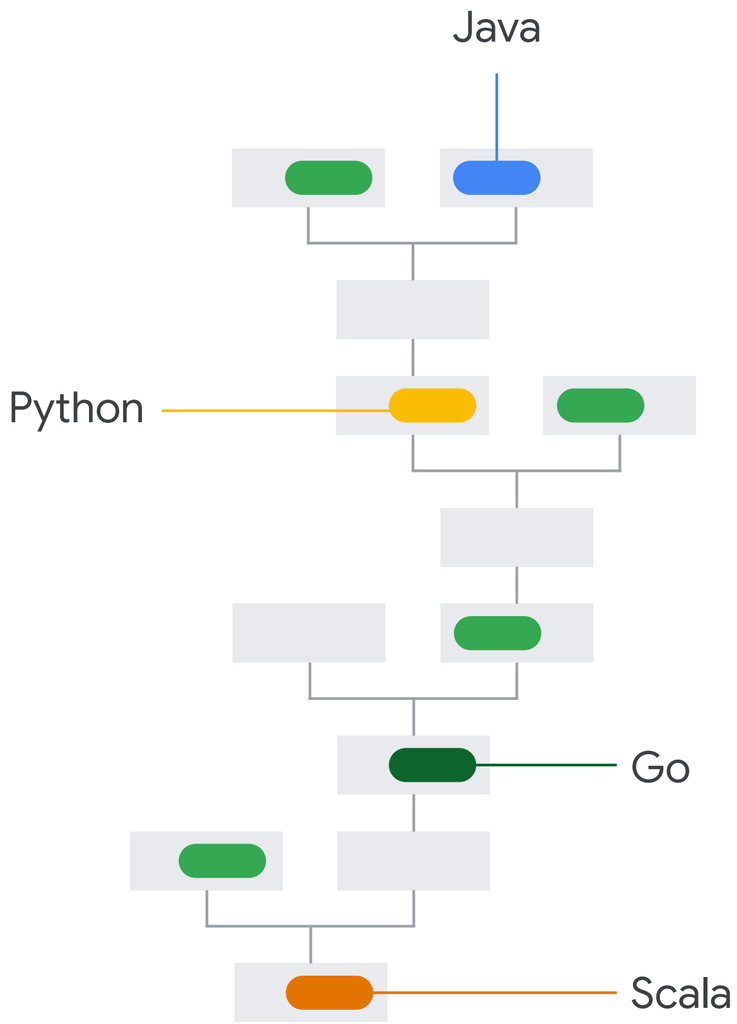
Benefits of portability

- **Every runner** works with **every language**
- Configurable, hermetic **worker environment**
- **Multi-language** pipelines



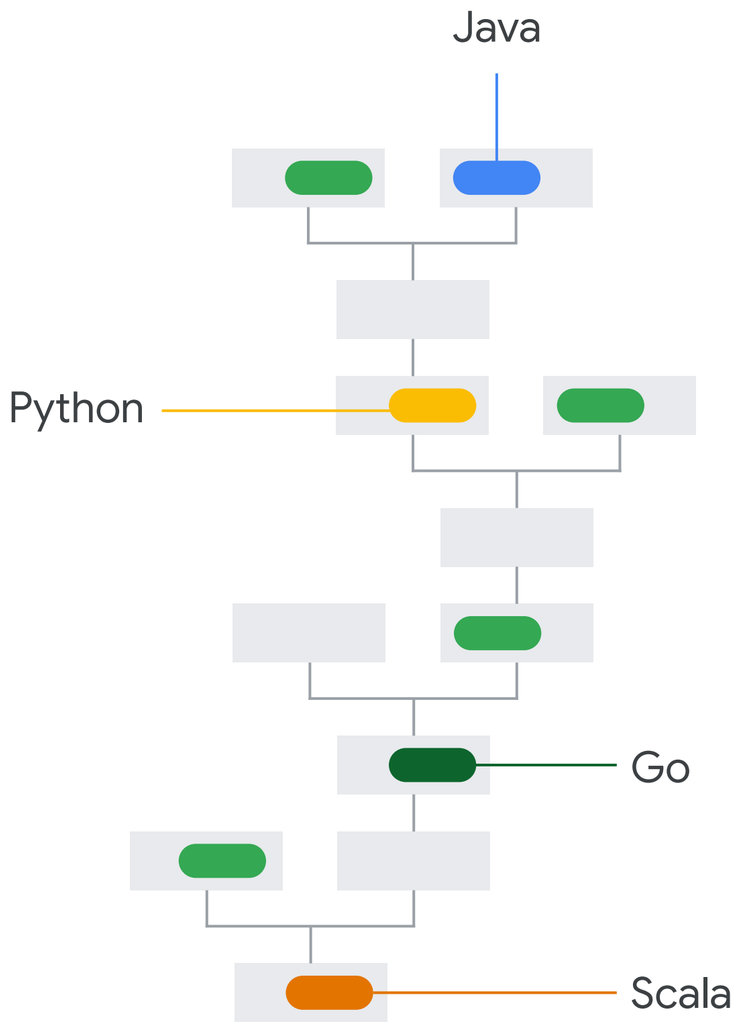
Benefits of portability

- **Every runner** works with **every language**
- Configurable, hermetic **worker environment**
- **Multi-language** pipelines
- **Cross-language** transforms



Benefits of portability

- **Every runner** works with **every language**
- Configurable, hermetic **worker environment**
- **Multi-language** pipelines
- **Cross-language** transforms
- Faster delivery of **new features**



Beam portability

Agenda

Beam portability

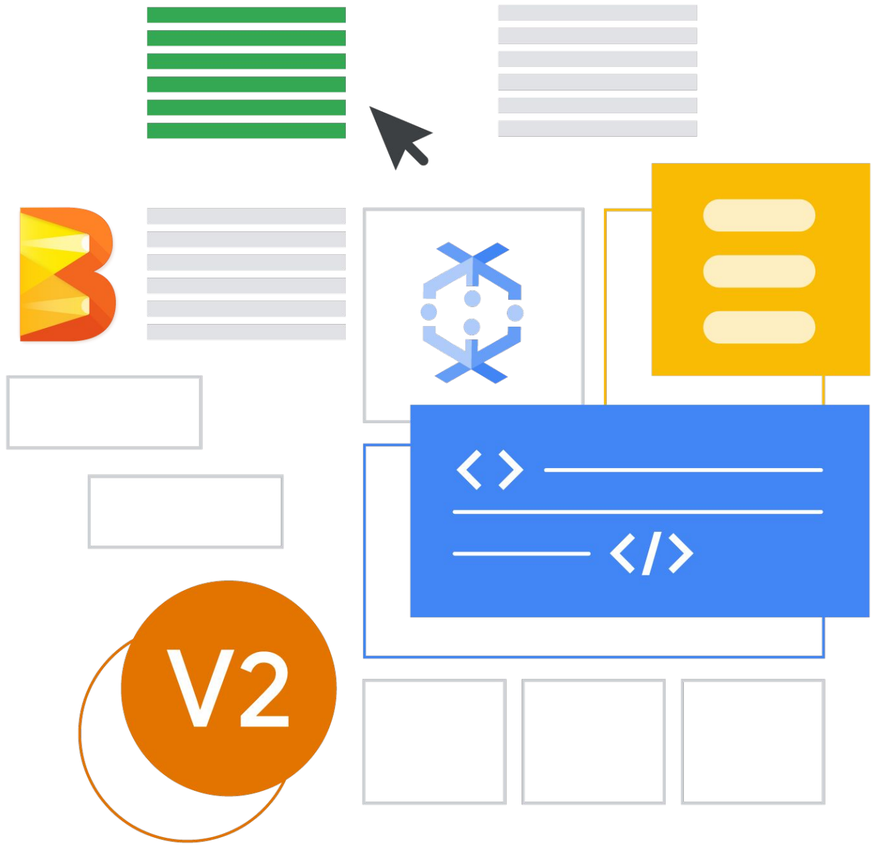
Runner v2

Container environments

Cross-language transforms

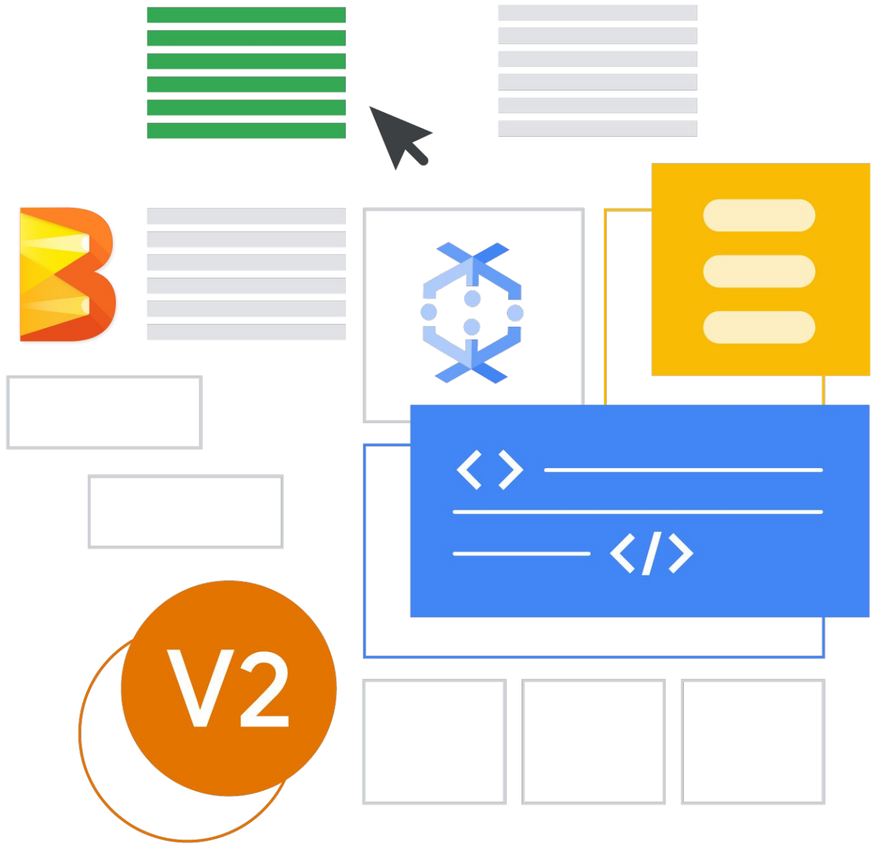
Dataflow Runner v2

- More efficient and portable worker architecture
- Based on Apache Beam portability framework



Dataflow Runner v2

- More efficient and portable worker architecture
- Based on Apache Beam portability framework
- Support for multi-language pipelines and custom containers

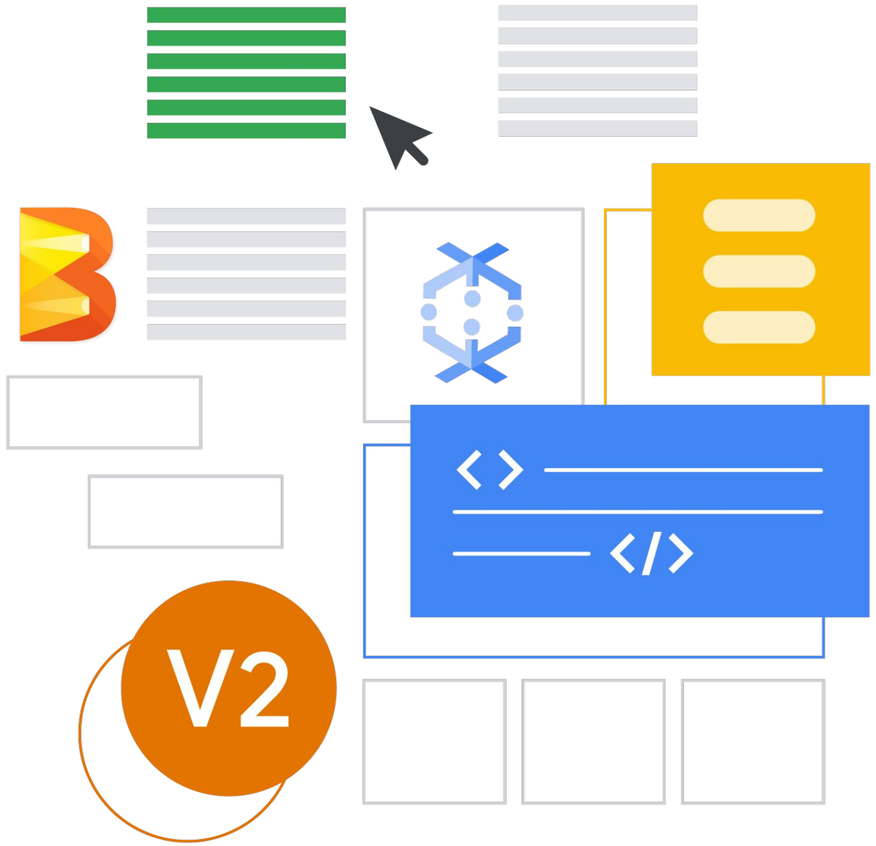


Dataflow Runner v2

- More efficient and portable worker architecture
- Based on Apache Beam portability framework
- Support for multi-language pipelines and custom containers

To enable Runner v2, refer to Dataflow official documentation:

https://cloud.google.com/dataflow/docs



Beam portability

Agenda

Beam portability

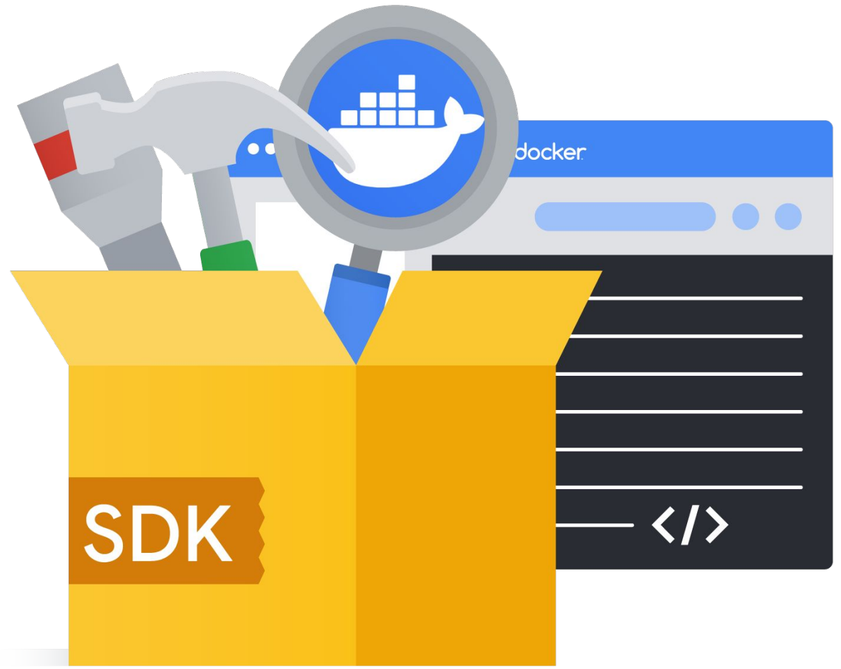
Runner v2

Container environments

Cross-language transforms

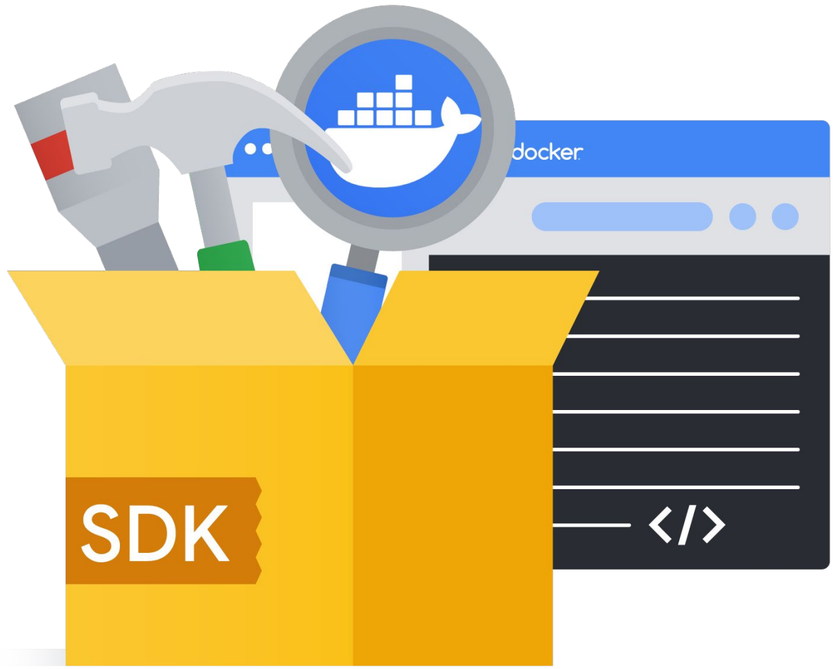
Container environments

- Containerized with Docker



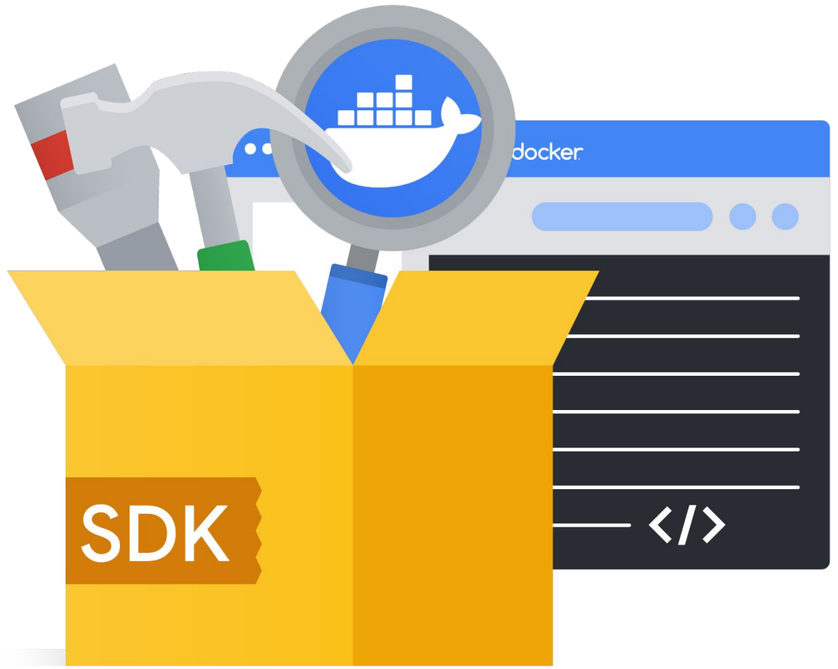
Container environments

- Containerized with Docker
- Per-operation execution environment



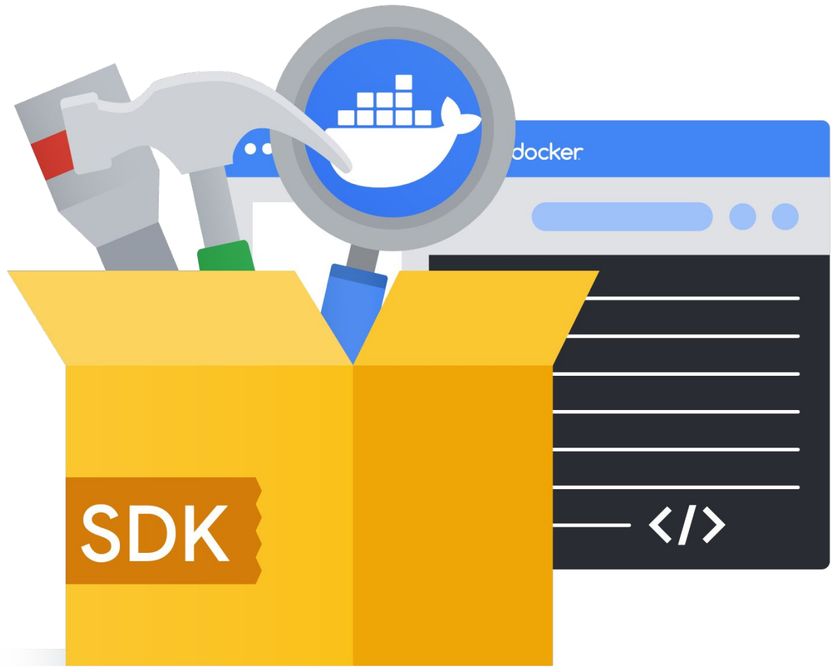
Container environments

- Containerized with Docker
- Per-operation execution environment
- Default environment per SDK



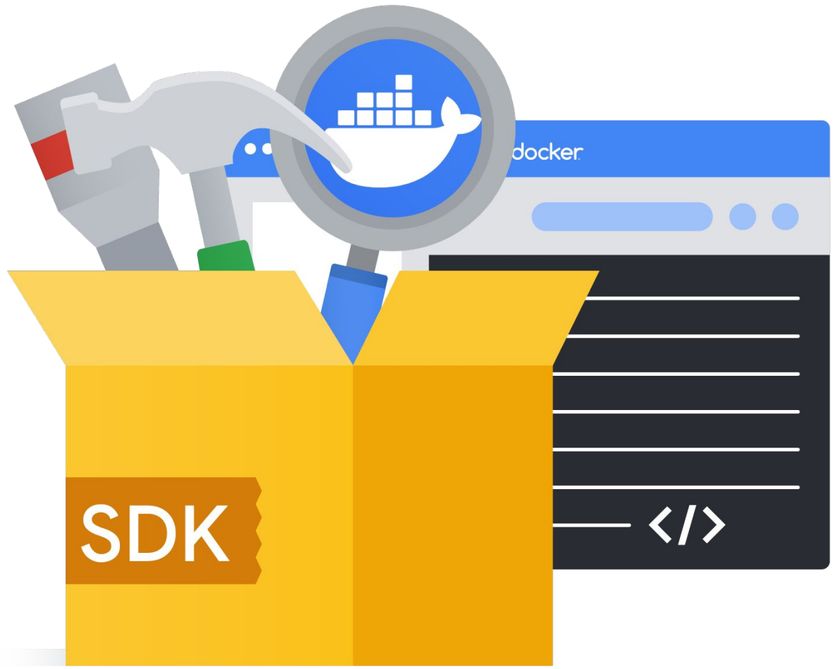
Container environments

- Containerized with Docker
- Per-operation execution environment
- Default environment per SDK
- Ahead-of-time installation



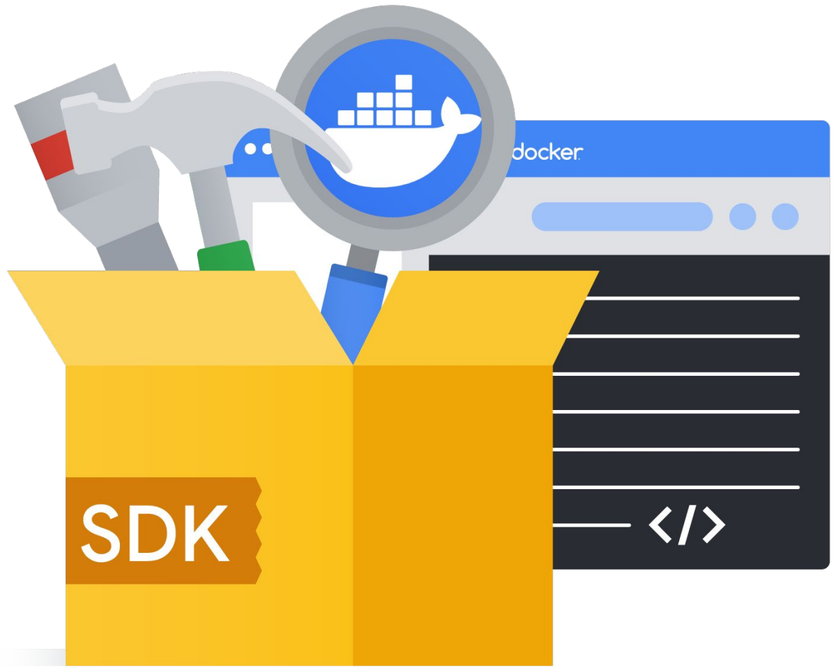
Container environments

- Containerized with Docker
- Per-operation execution environment
- Default environment per SDK
- Ahead-of-time installation
- Arbitrary dependencies



Container environments

- Containerized with Docker
- Per-operation execution environment
- Default environment per SDK
- Ahead-of-time installation
- Arbitrary dependencies
- Arbitrary customization



Custom container

Running your pipeline

Custom container

Running your pipeline

- Apache Beam SDK version 2.25.0 or later is required.

Custom container

Running your pipeline

- Apache Beam SDK version 2.25.0 or later is required.
- Docker is required if you want to test your pipeline locally.

Custom container

Running your pipeline

- Apache Beam SDK version 2.25.0 or later is required.
- Docker is required if you want to test your pipeline locally.
- You start by creating a Dockerfile:

```
# Specifying the base image with FROM instruction
FROM apache/beam_python3.8_sdk:2.25.0
# Adding an environment variable with ENV instruction
ENV MY_FILE_NAME=my_file.txt
# Copying files to add to the custom image with COPY instruction
COPY path/to/myfile/$MY_FILE_NAME ./
```

Custom container

Building your image

Custom container

Building your image

```
export PROJECT=my-project-id
export REPO=my-repository
export TAG=my-image-tag
export REGISTRY_HOST=gcr.io
export IMAGE_URI=$REGISTRY_HOST/$PROJECT/$REPO:$TAG
```

Custom container

Building your image

```
export PROJECT=my-project-id
export REPO=my-repository
export TAG=my-image-tag
export REGISTRY_HOST=gcr.io
export IMAGE_URI=$REGISTRY_HOST/$PROJECT/$REPO:$TAG
```

```
gcloud builds submit --tag $IMAGE_URI
```

→ Cloud Build

Custom container

Building your image

```
export PROJECT=my-project-id
export REPO=my-repository
export TAG=my-image-tag
export REGISTRY_HOST=gcr.io
export IMAGE_URI=$REGISTRY_HOST/$PROJECT/$REPO:$TAG
```

```
gcloud builds submit --tag $IMAGE_URI
```

→ Cloud Build

```
docker build -f Dockerfile -t $IMAGE_URI ./
docker push $IMAGE_URI
```

→ Docker

Custom container

Launching your job

```
python my-pipeline.py \
  --input=INPUT_FILE \
  --output=OUTPUT_FILE \
  --project=PROJECT_ID \
  --region=REGION \
  --temp_location=TEMP_LOCATION \
  --runner=DataflowRunner \
  --worker_harness_container_image=$IMAGE_URI
```

Beam portability

Agenda

Beam portability

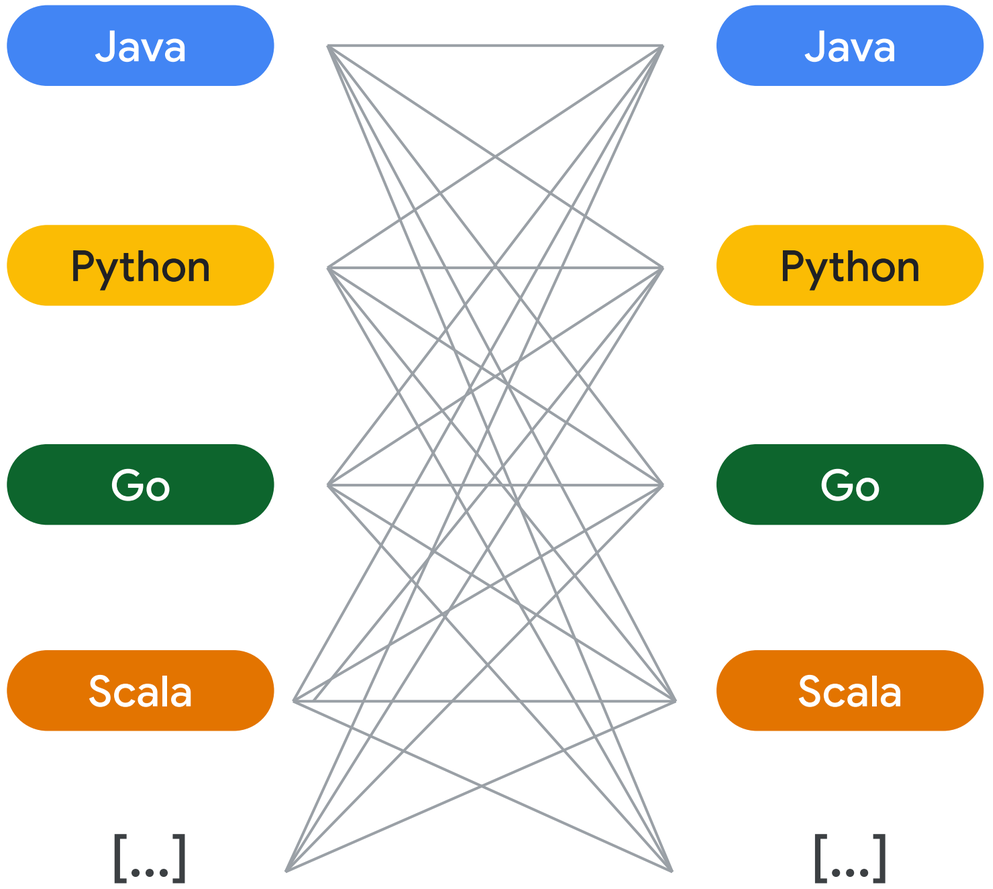
Runner v2

Container environments

Cross-language transforms

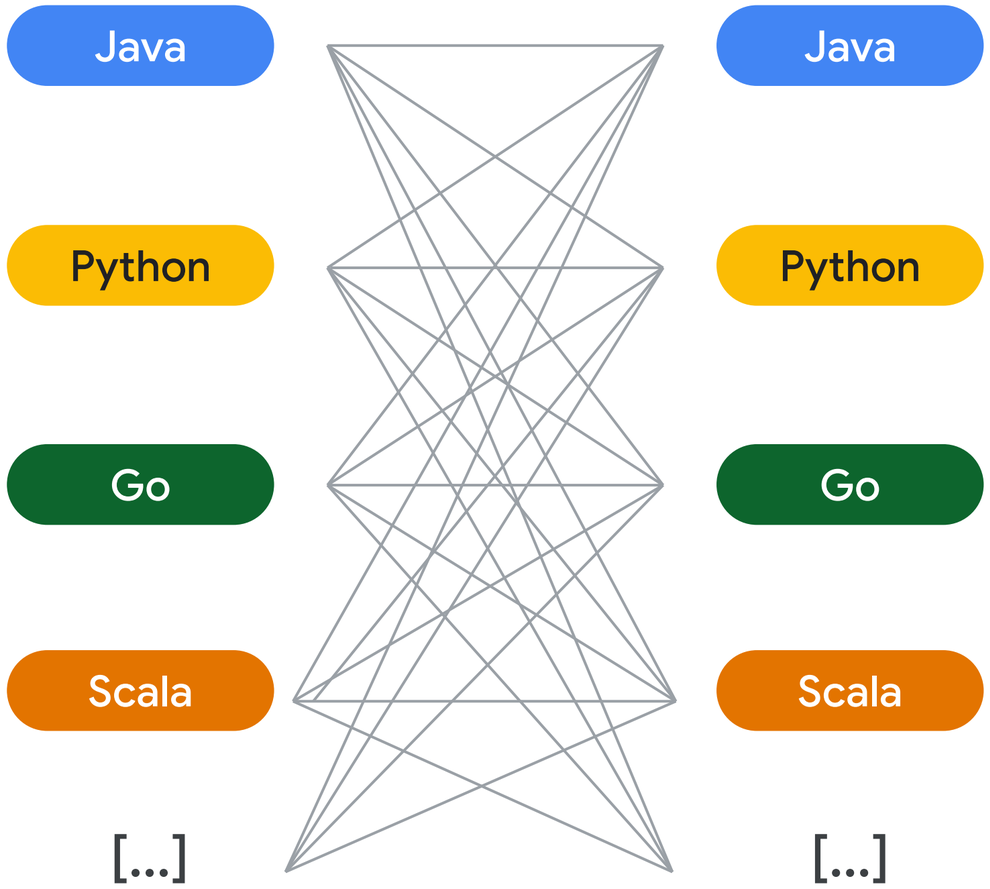
Cross-language transforms

- Transforms can be **shared** among SDKs.



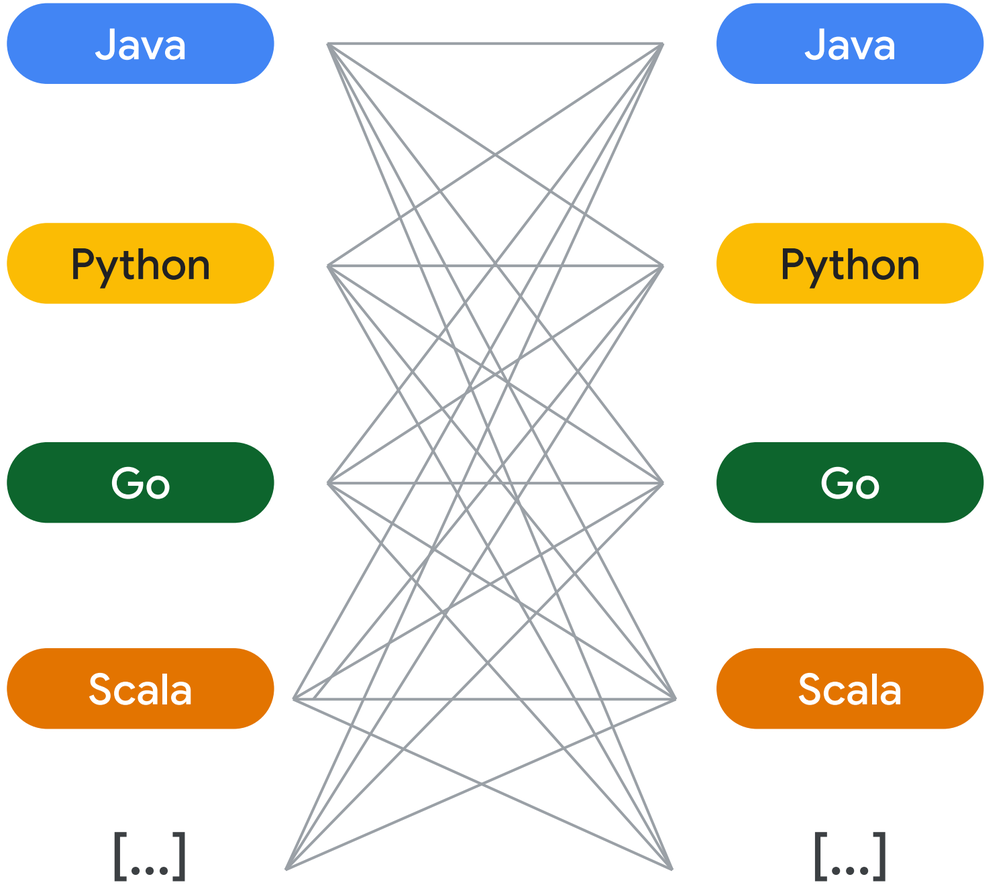
Cross-language transforms

- Transforms can be **shared** among SDKs.
- A **rich set of IOs** from Java is available everywhere.



Cross-language transforms

- Transforms can be **shared** among SDKs.
- A **rich set of IOs** from Java is available everywhere.
- More libraries are available in the **language of your choice**.

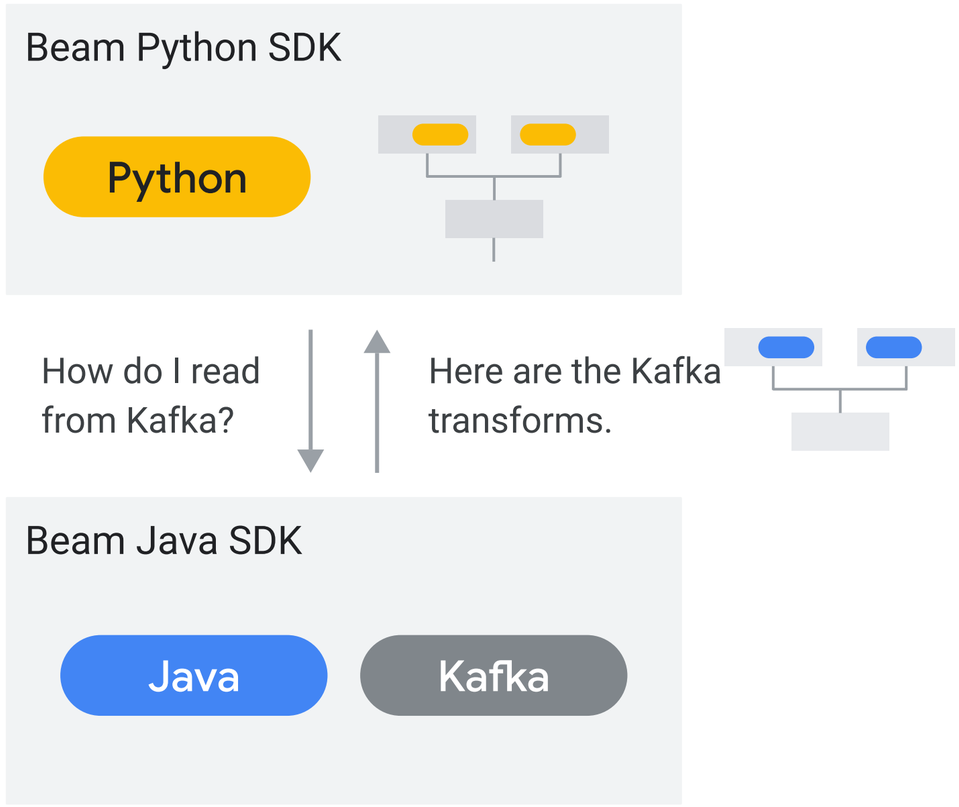


Example of cross-language transform

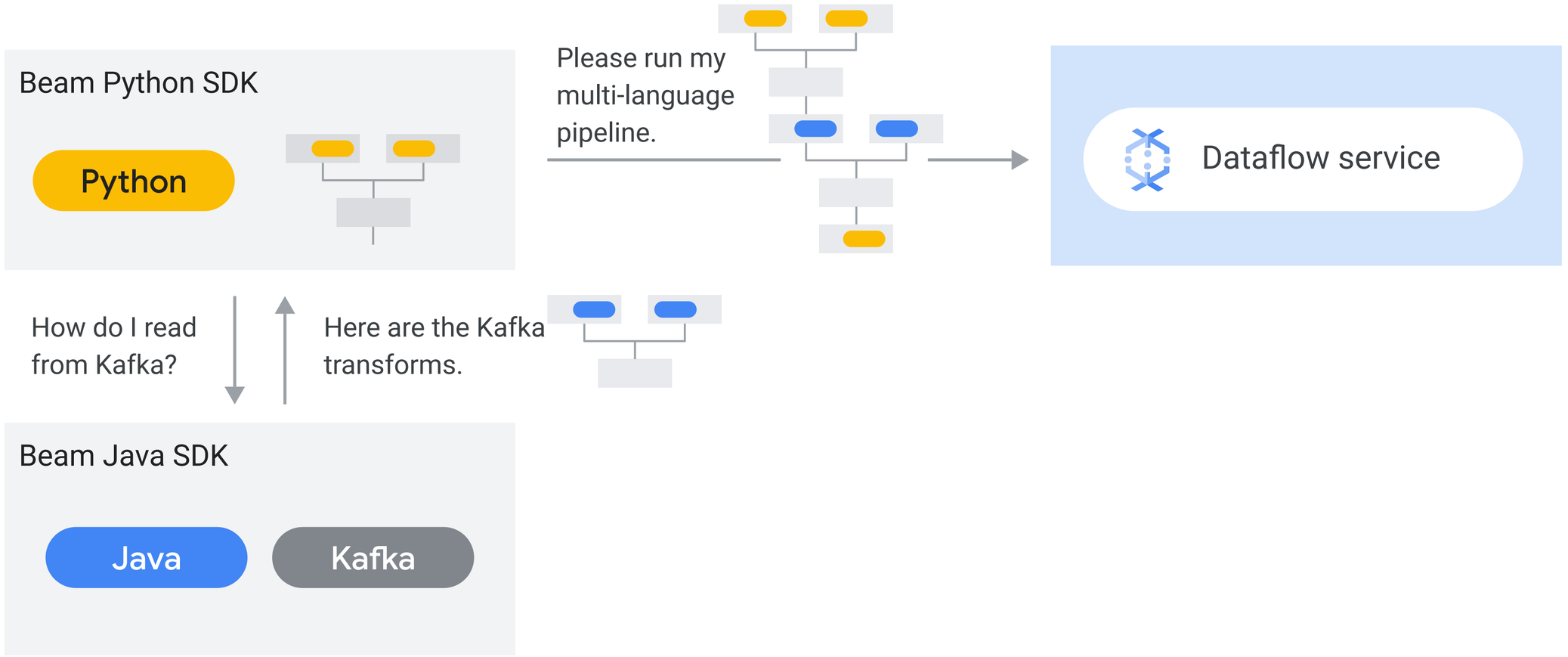
```
from apache_beam.io.kafka import ReadFromKafka

with beam.Pipeline(options=<Your Beam PipelineOptions object>) as p:
      p
      | ReadFromKafka(
          consumer_config={'bootstrap.servers': ‘<Kafka bootstrap servers list>’},
            topics=[<List of Kafka topics>])
```

Cross-language transforms



Cross-language transforms



Cross-language transforms