A Streaming Quantitative Analytics Engine

Raj Subramani <raj@subramani.com>

The Challenge

Continued Regulatory Expansion

Principle based compliance

Move to Centralized Clearing

Lower margin

Higher Volumes

Non-financial Risks

Contagion risk

Model risk

Improved Decision Making

Bias recognition

Bias elimination

Technology and Analytics as Risk Muscle

Big data

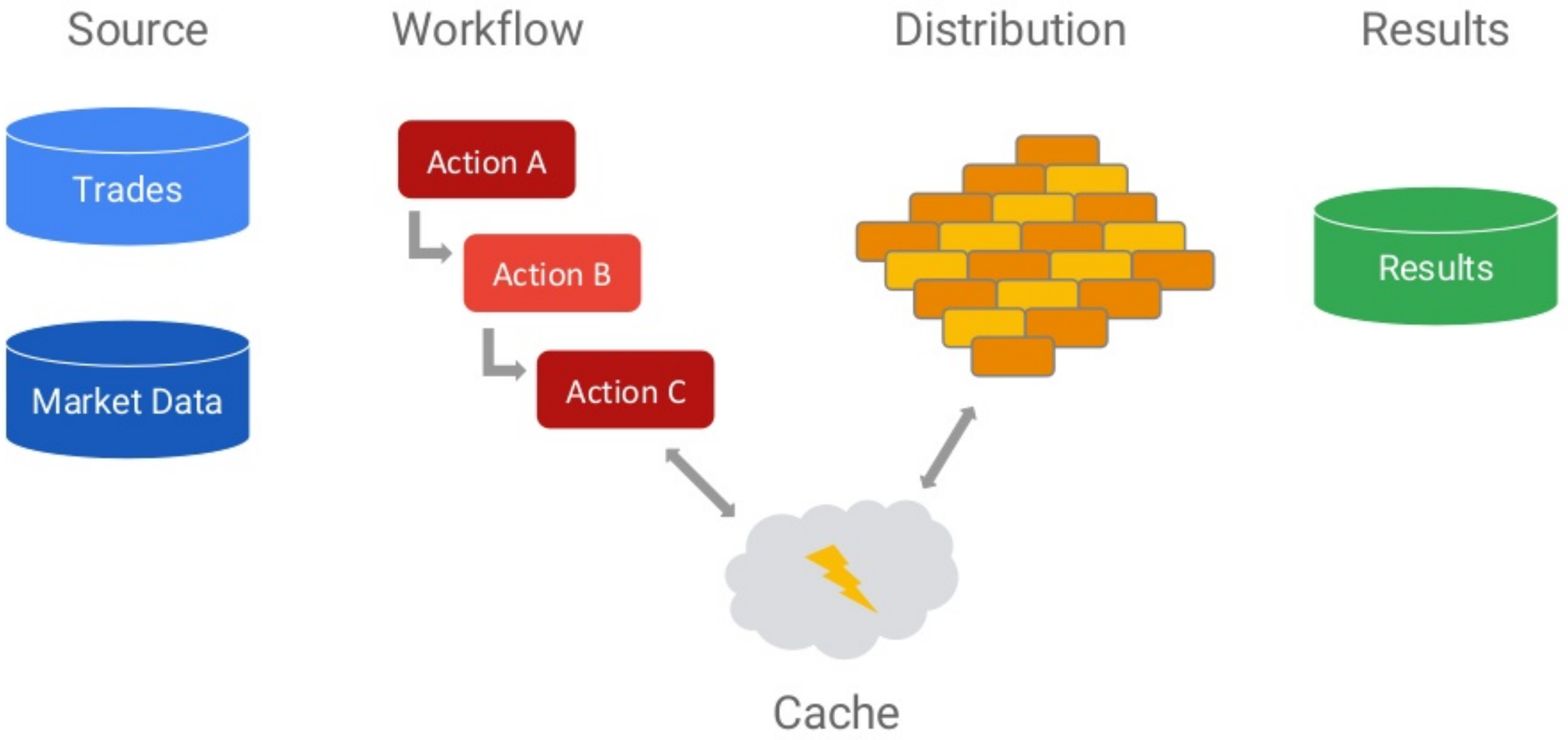
Machine learning

Cost Challenges

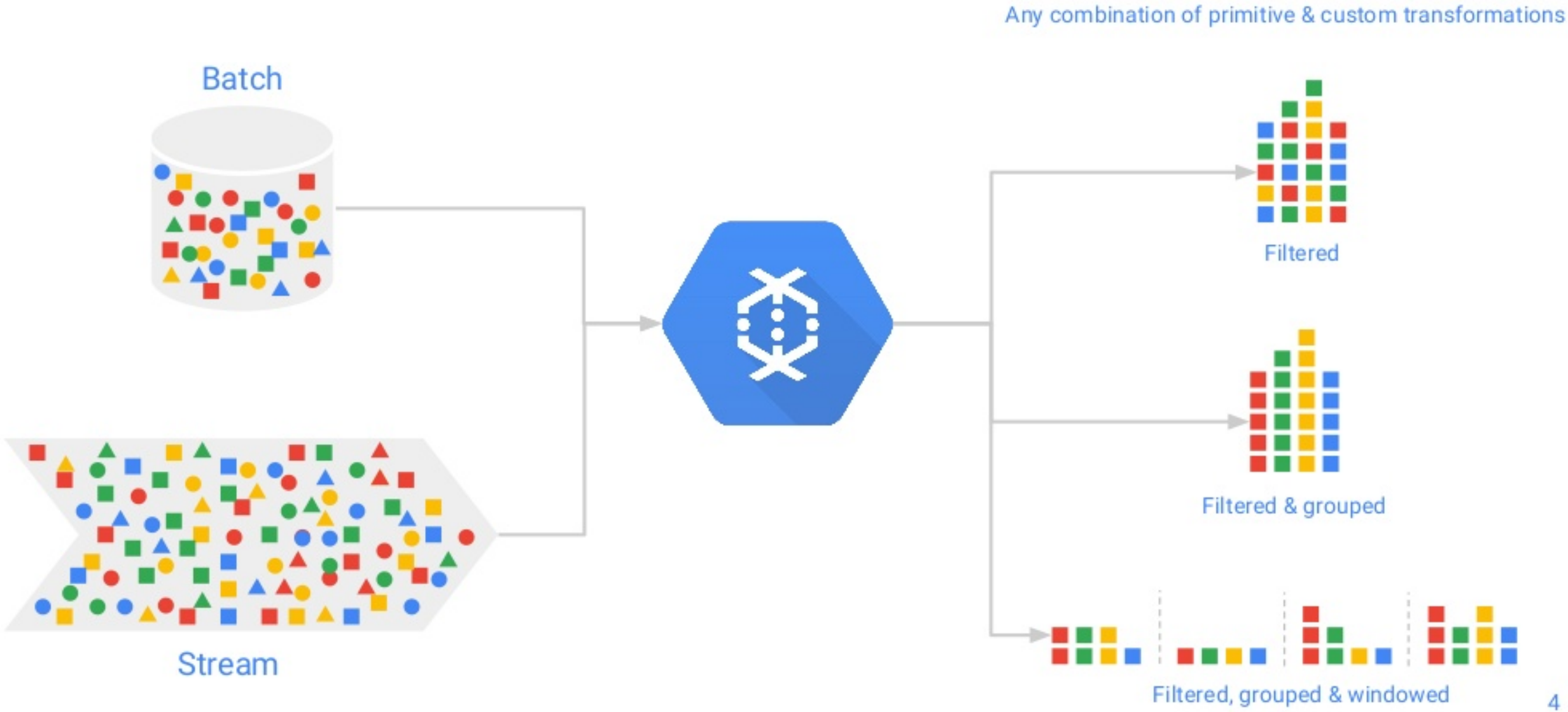
Capex

Opex

The Anatomy of a Risk Engine

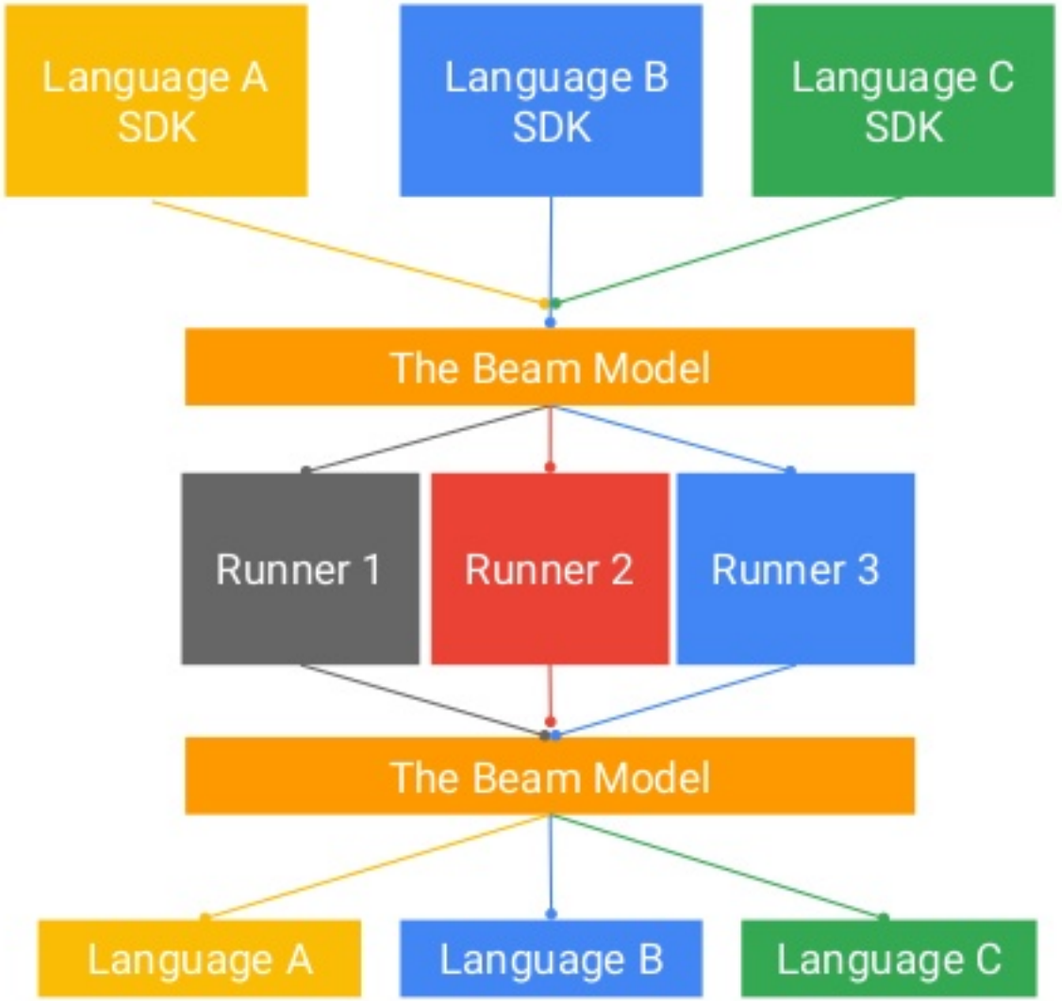


Cloud Dataflow - Unified Batch and Stream



Apache Beam

Apache Beam


Language A SDK
Language B SDK
Language C SDK
The Beam Model
Runner 1
Runner 2
Runner 3
The Beam Model
Language A
Language B
Language C

The Beam Model & Cloud Dataflow

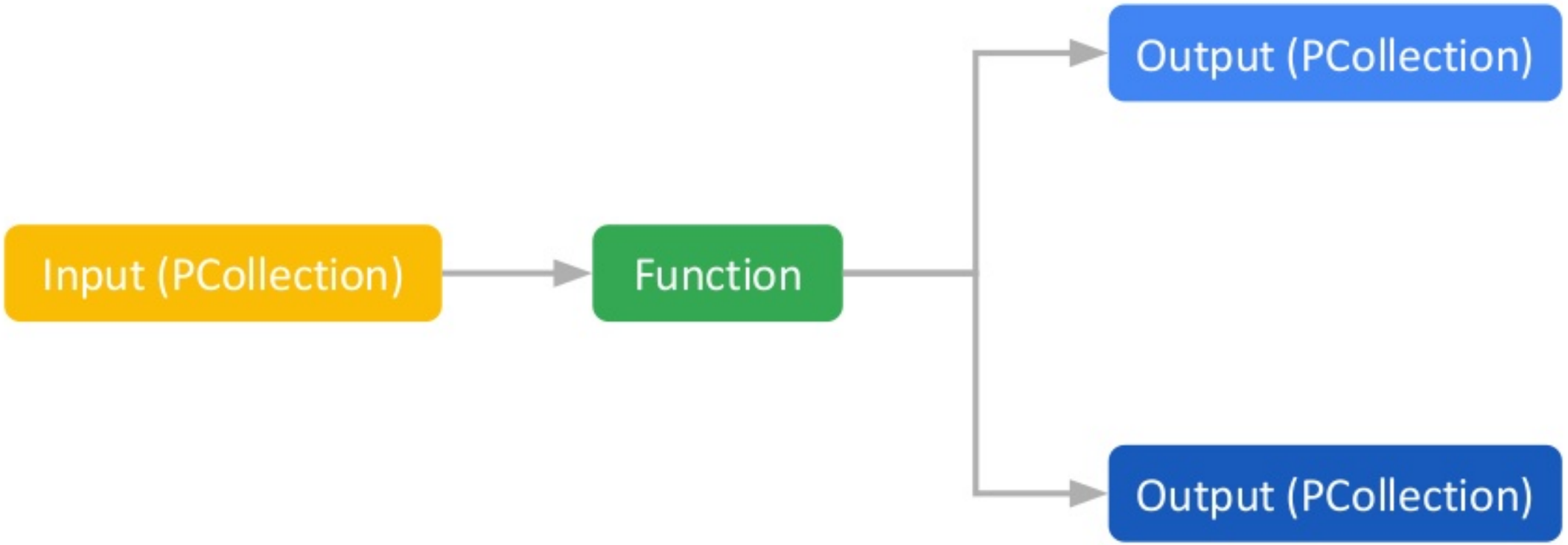
Apache Beam

a unified model for
batch and stream processing
supporting multiple runtimes

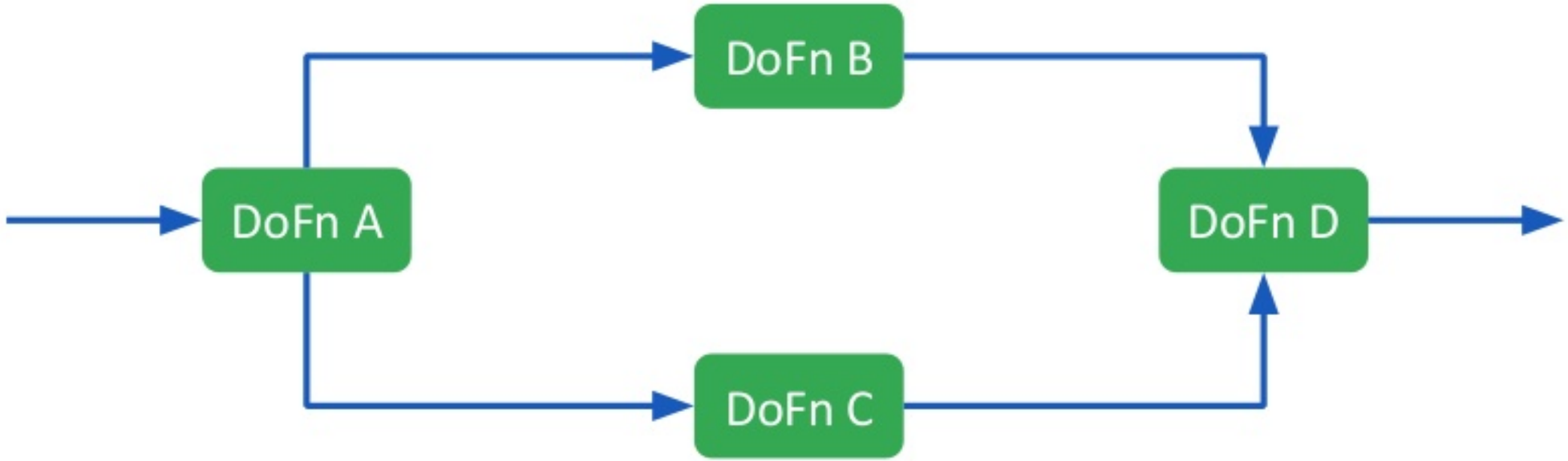
Google Cloud Dataflow

a great place to run Beam

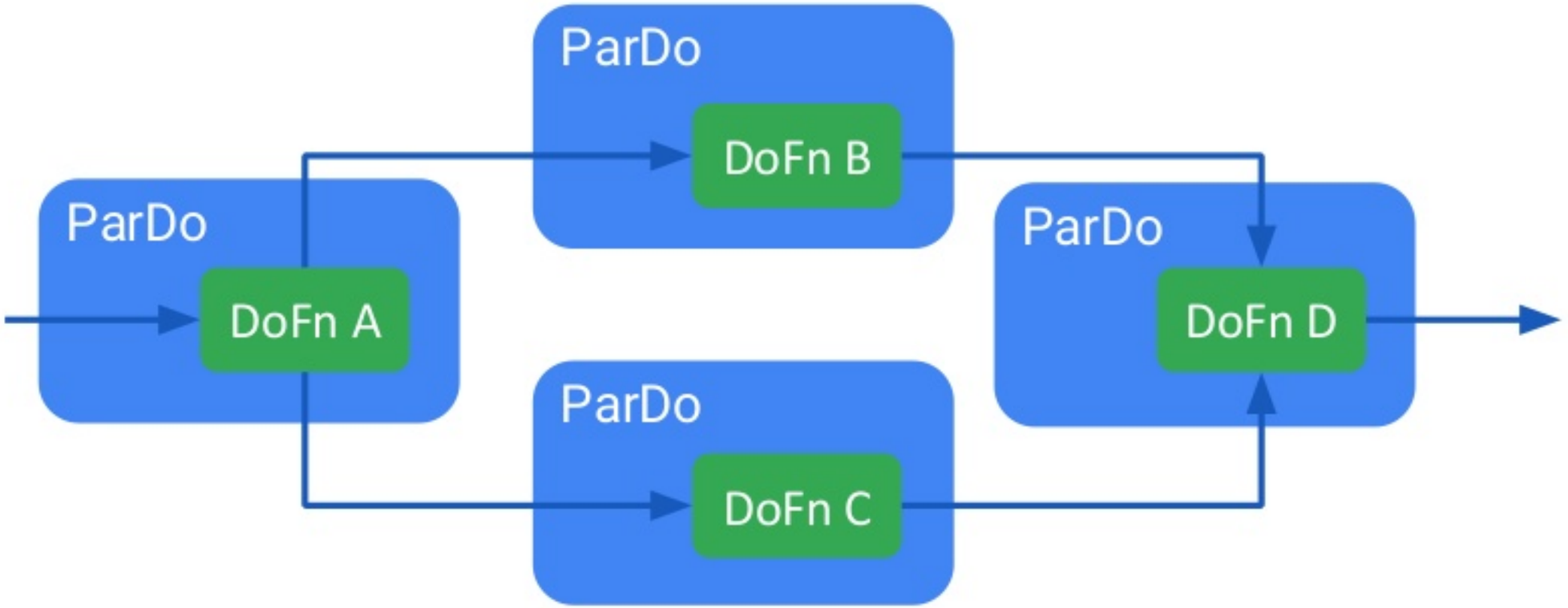
DoFn – Functional Programming Style



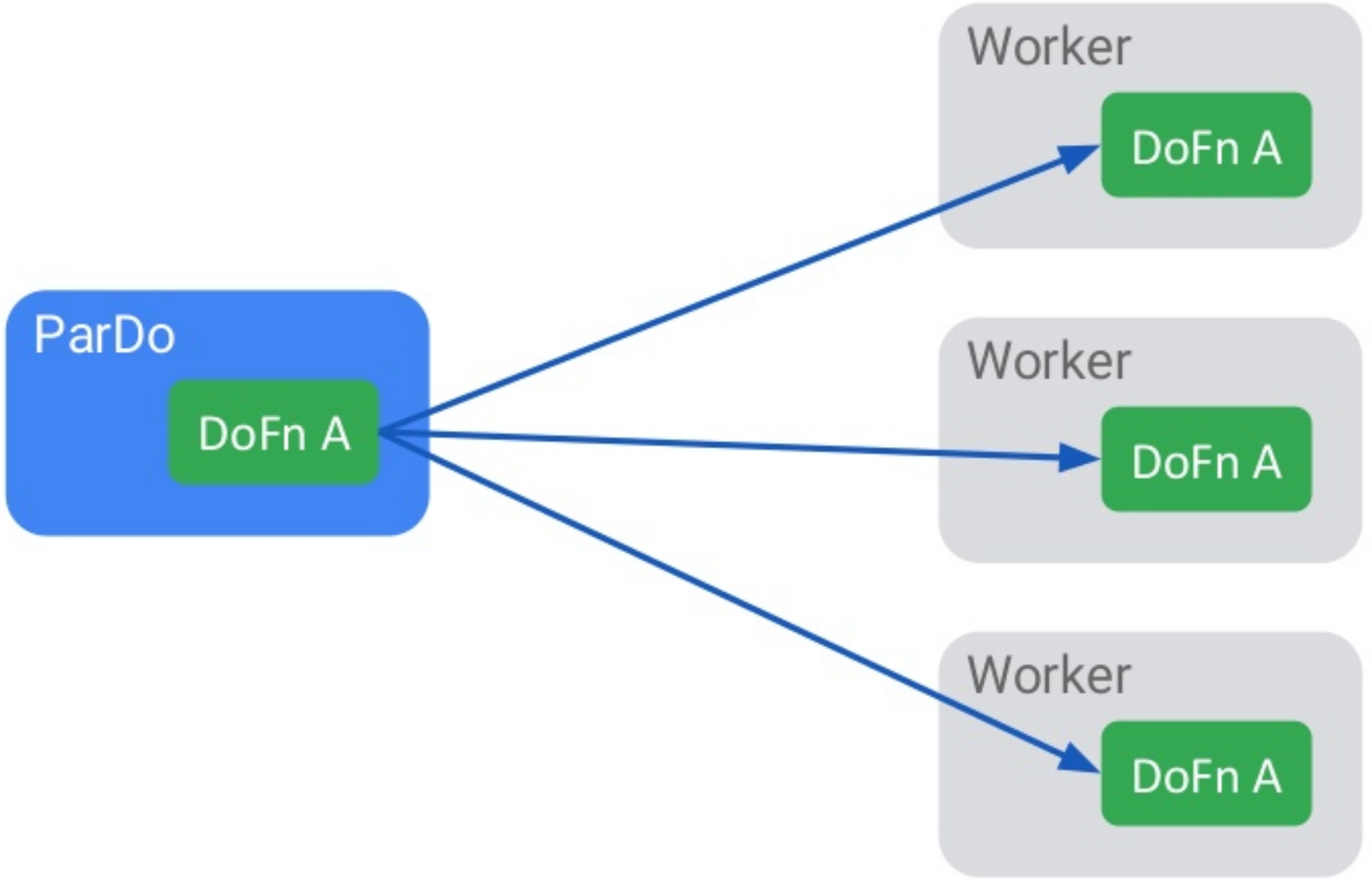
Workflow Pipeline



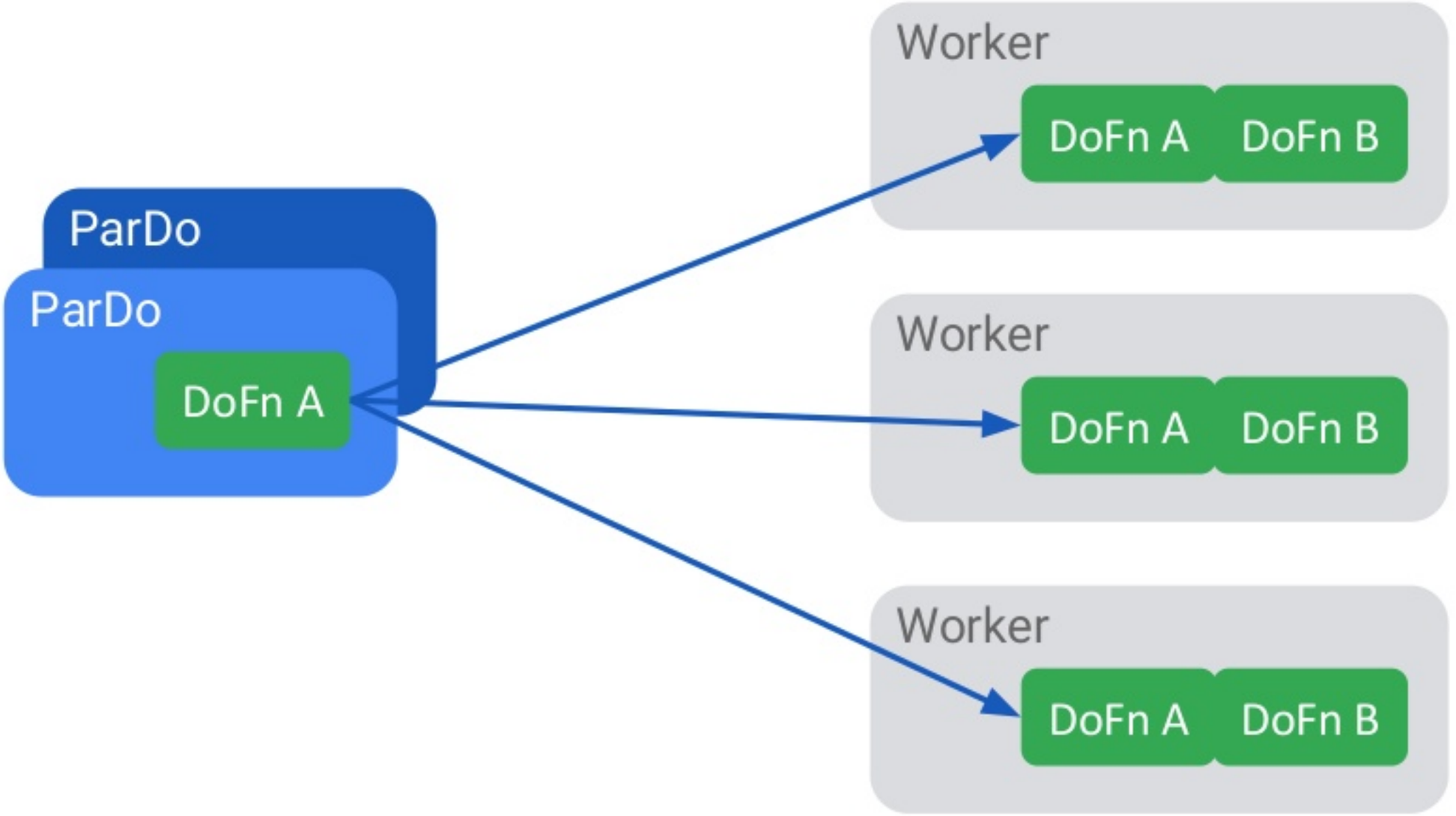
ParDo – Parallel Processing of DoFn's (Known as a PTransform)



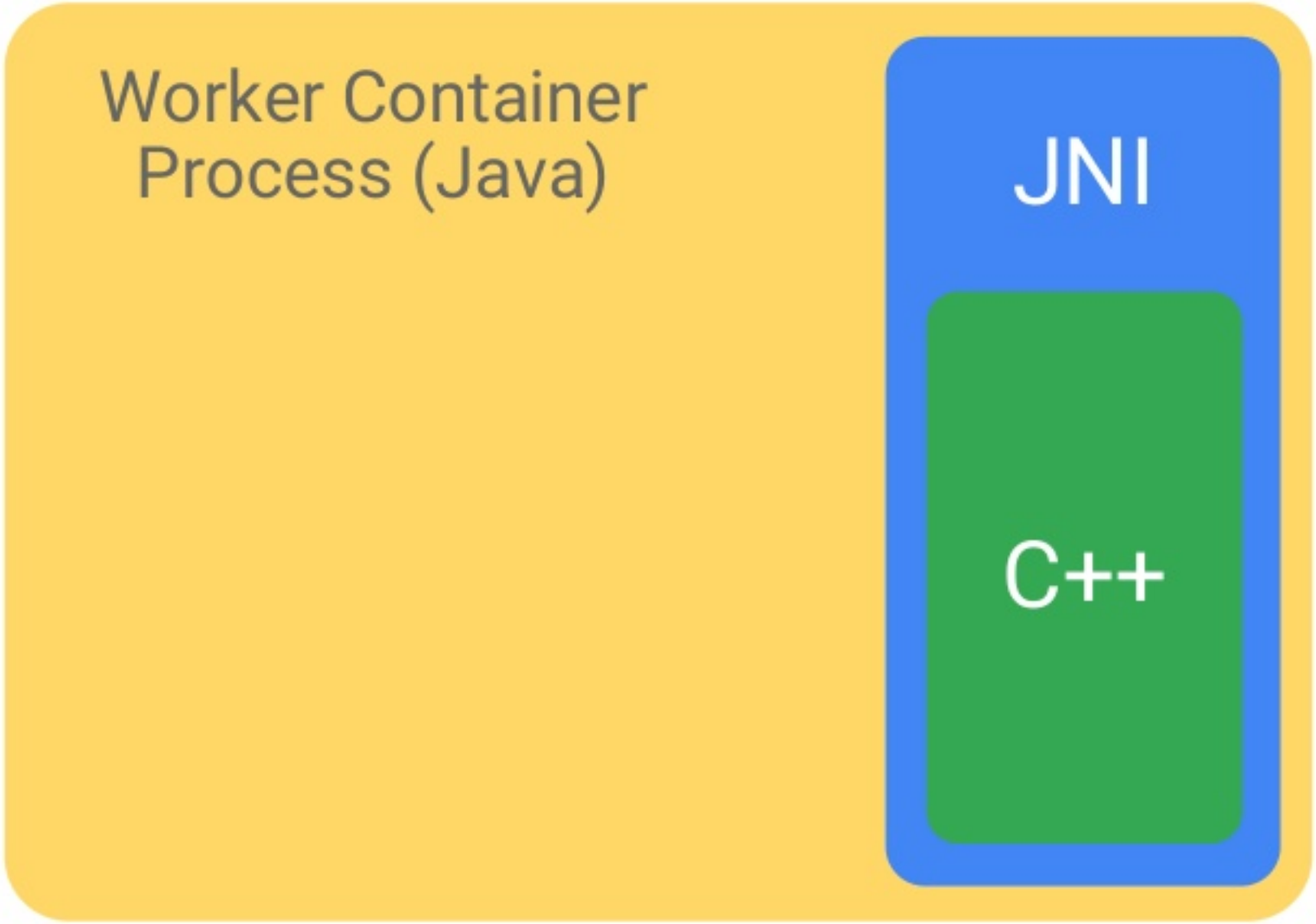
Distribution to workers



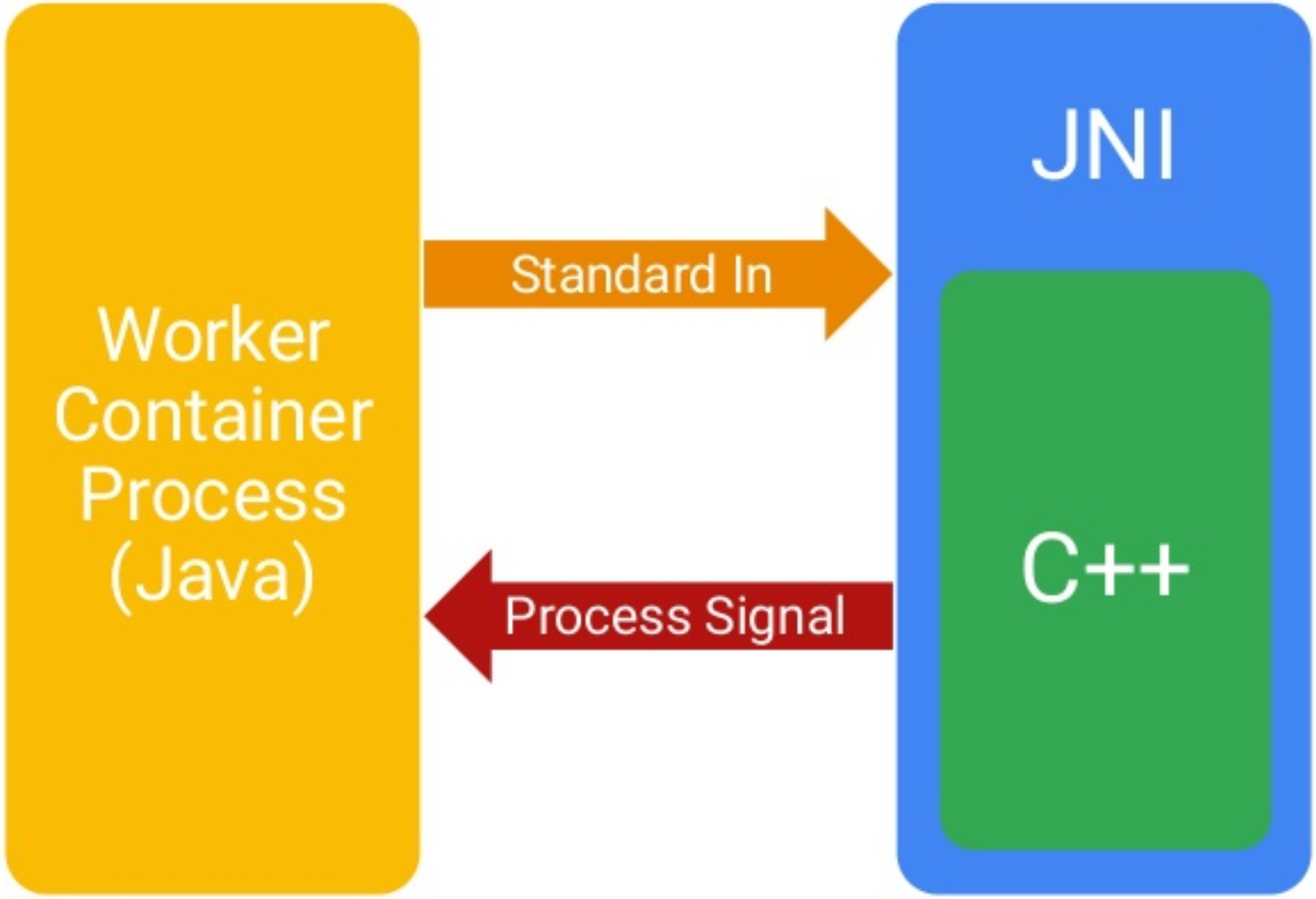
Dataflow Fusion



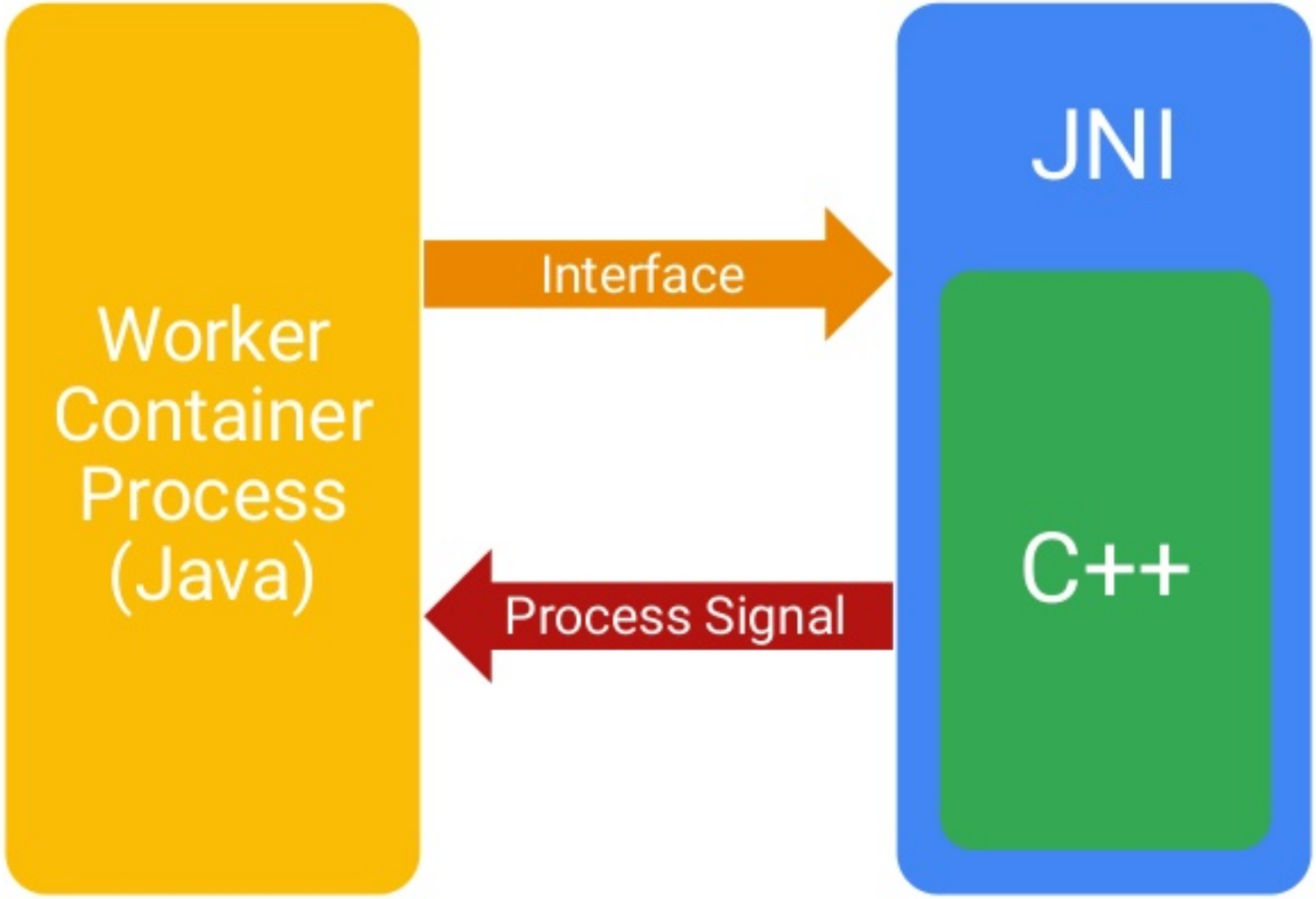
Native Code Execution



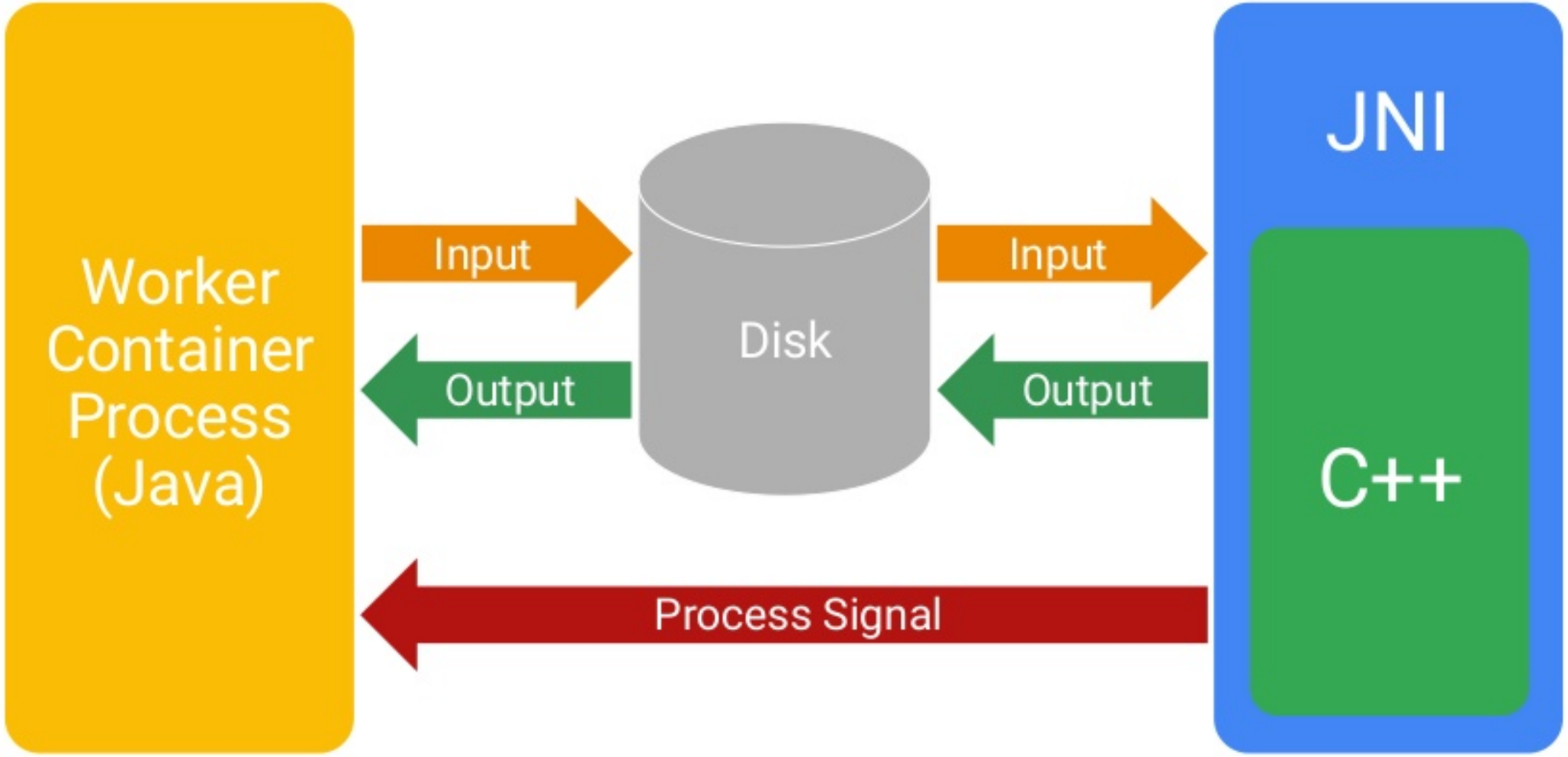
Out of Process Call



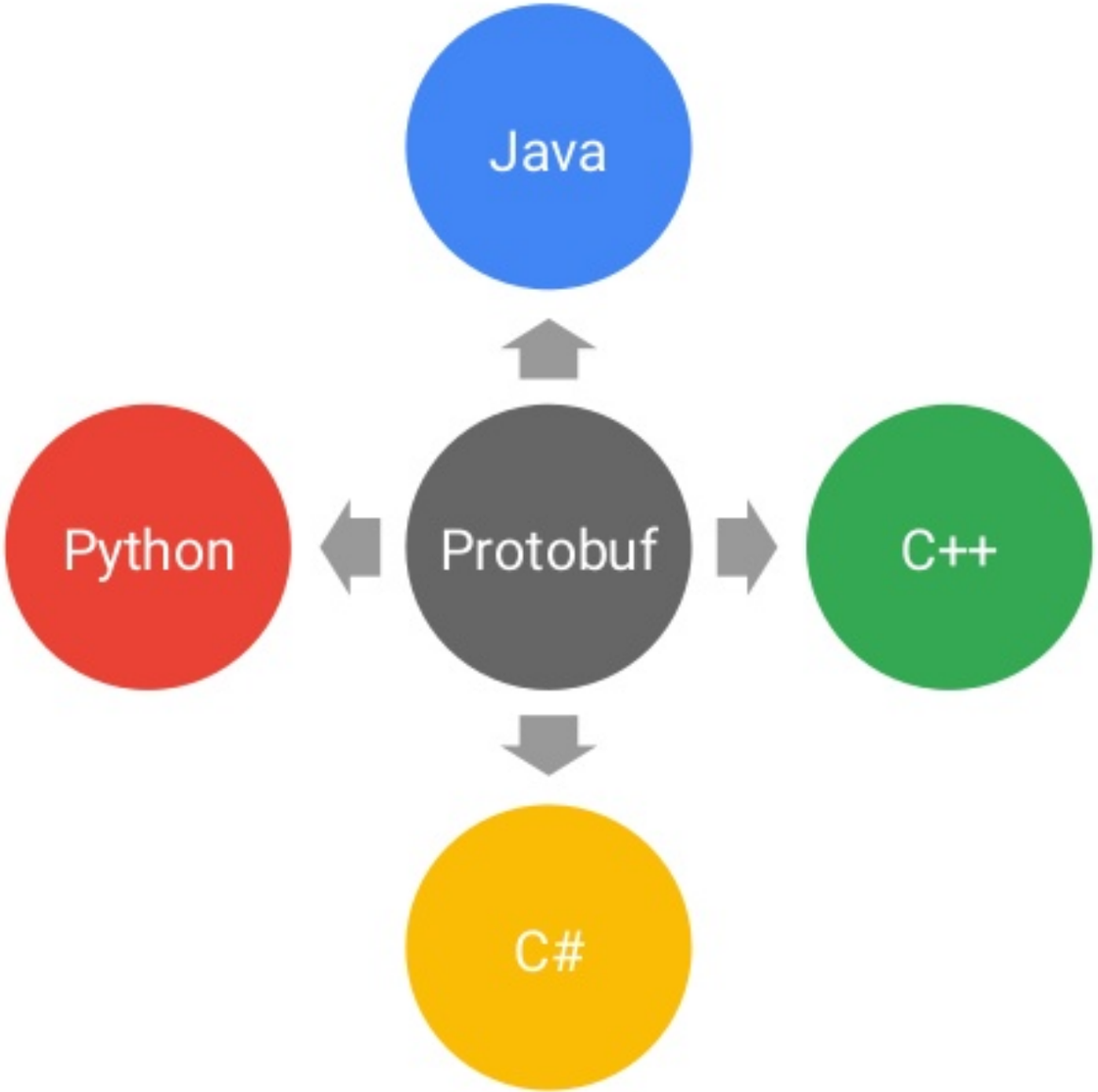
Out of Process Call



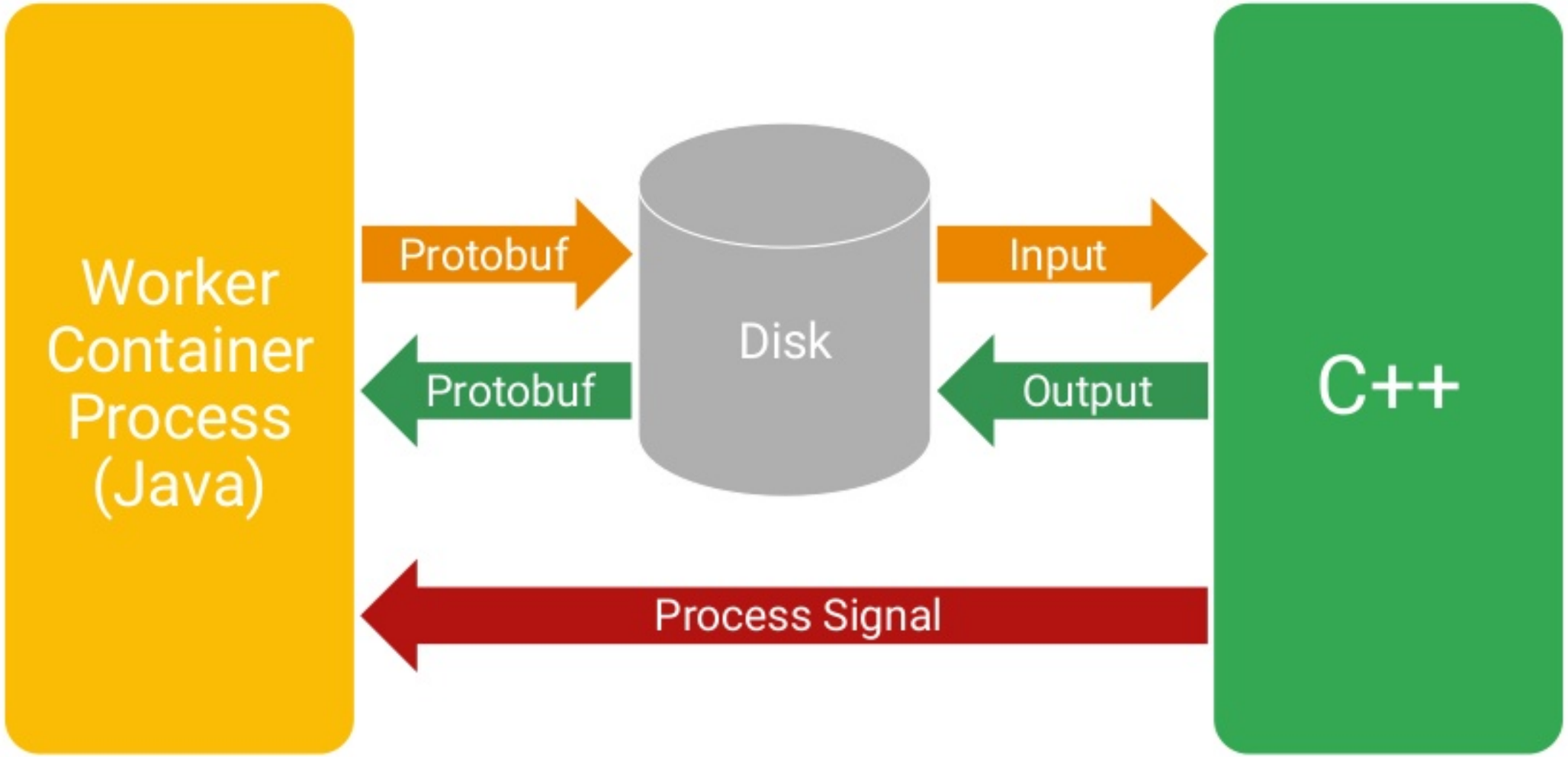
Out of Process Call



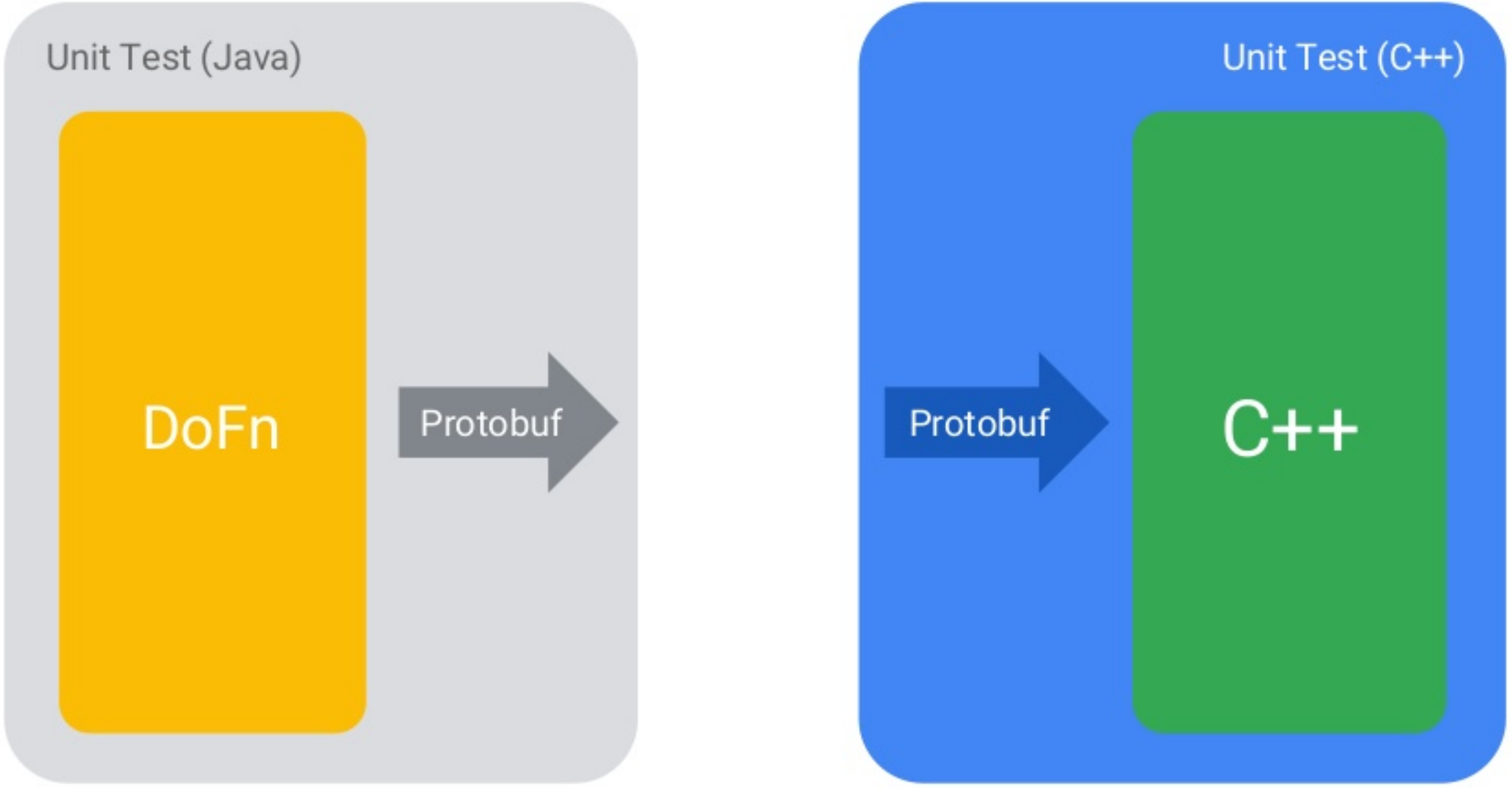
Out of Process Call



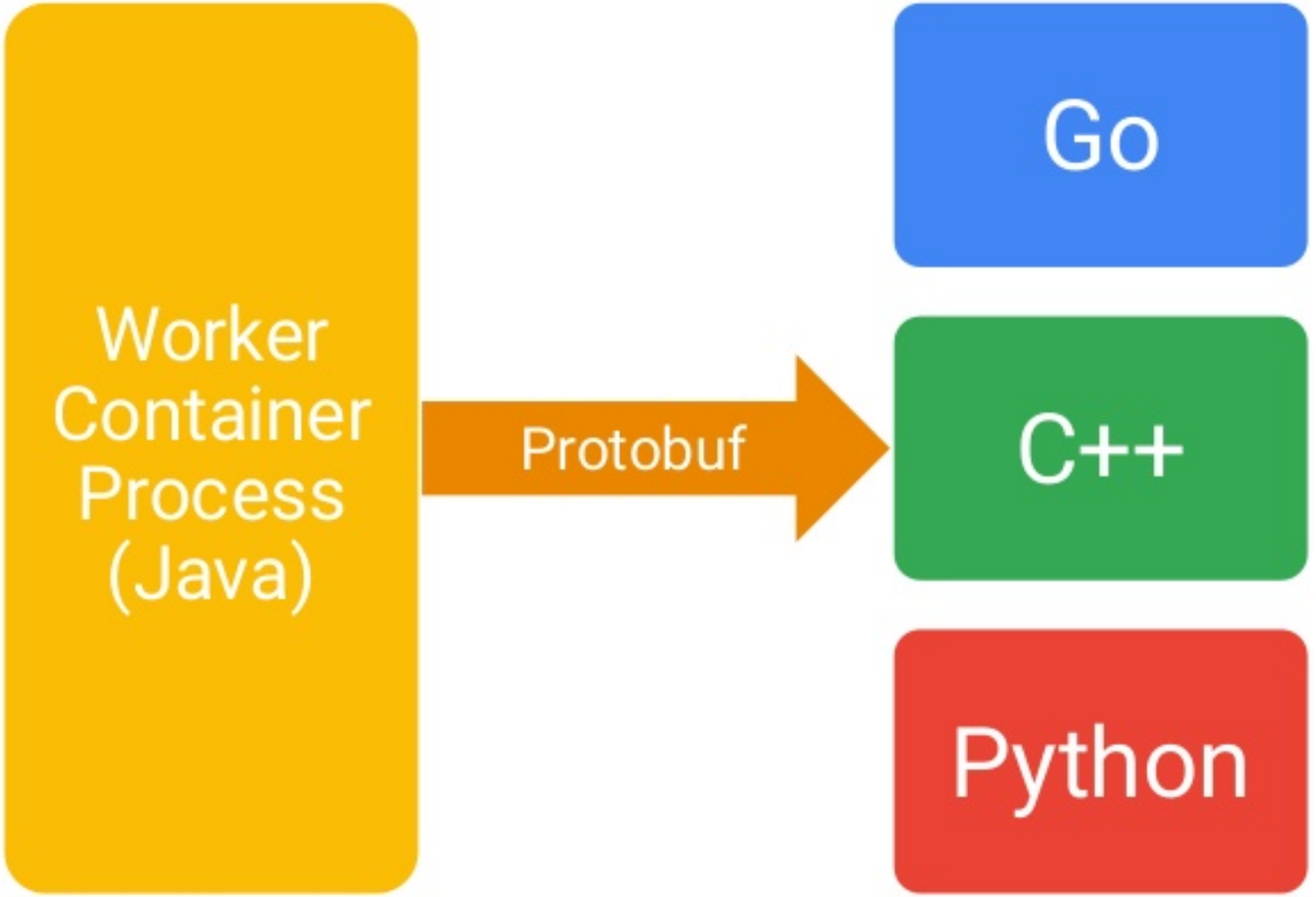
Out of Process Call



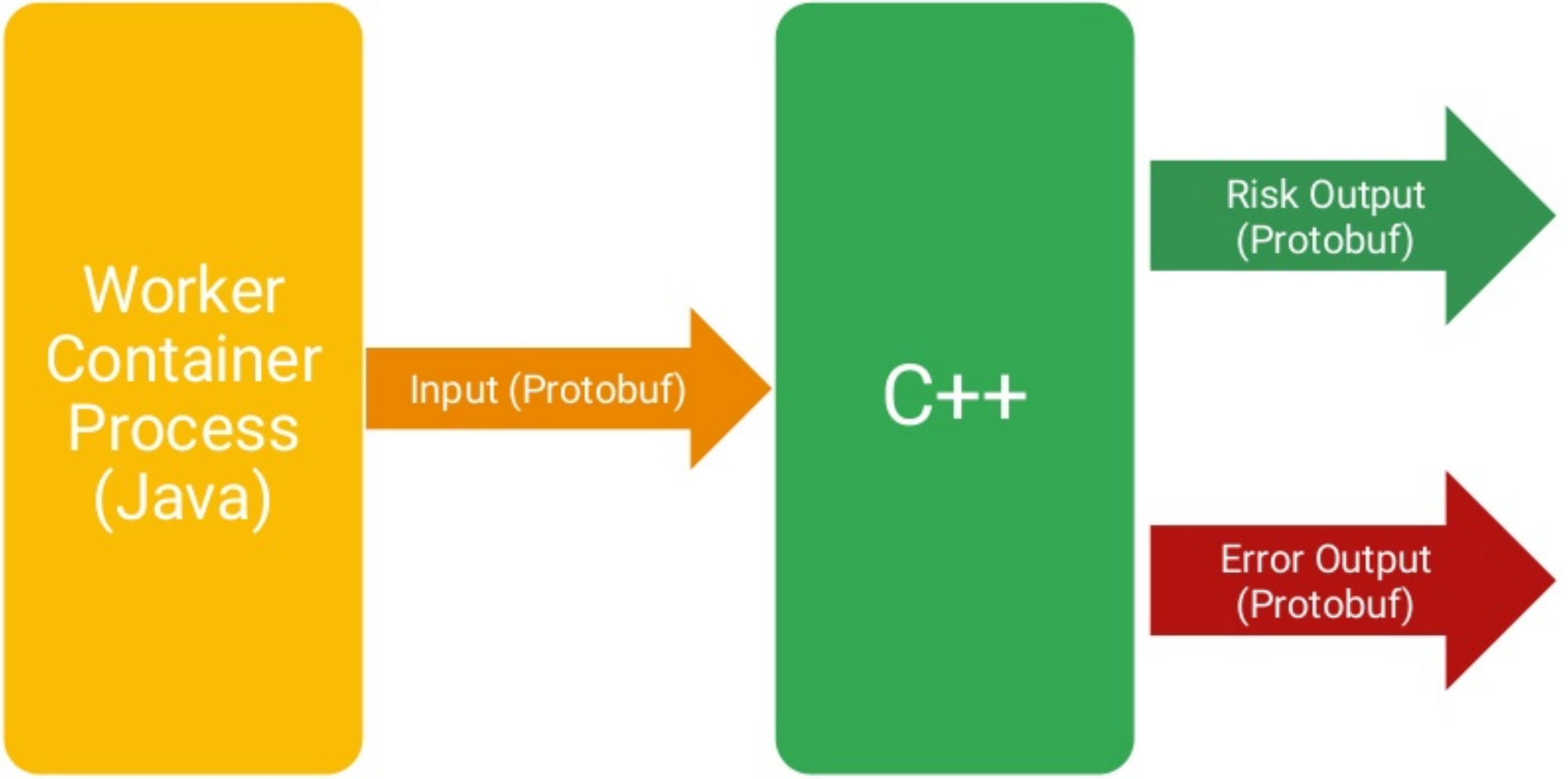
Out of Process Call: Testing!



Out of Process Call



Out of Process Call

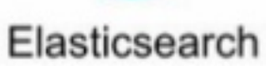


Module Separation

Module Separation

I/O

Pipeline code and functions



I/O

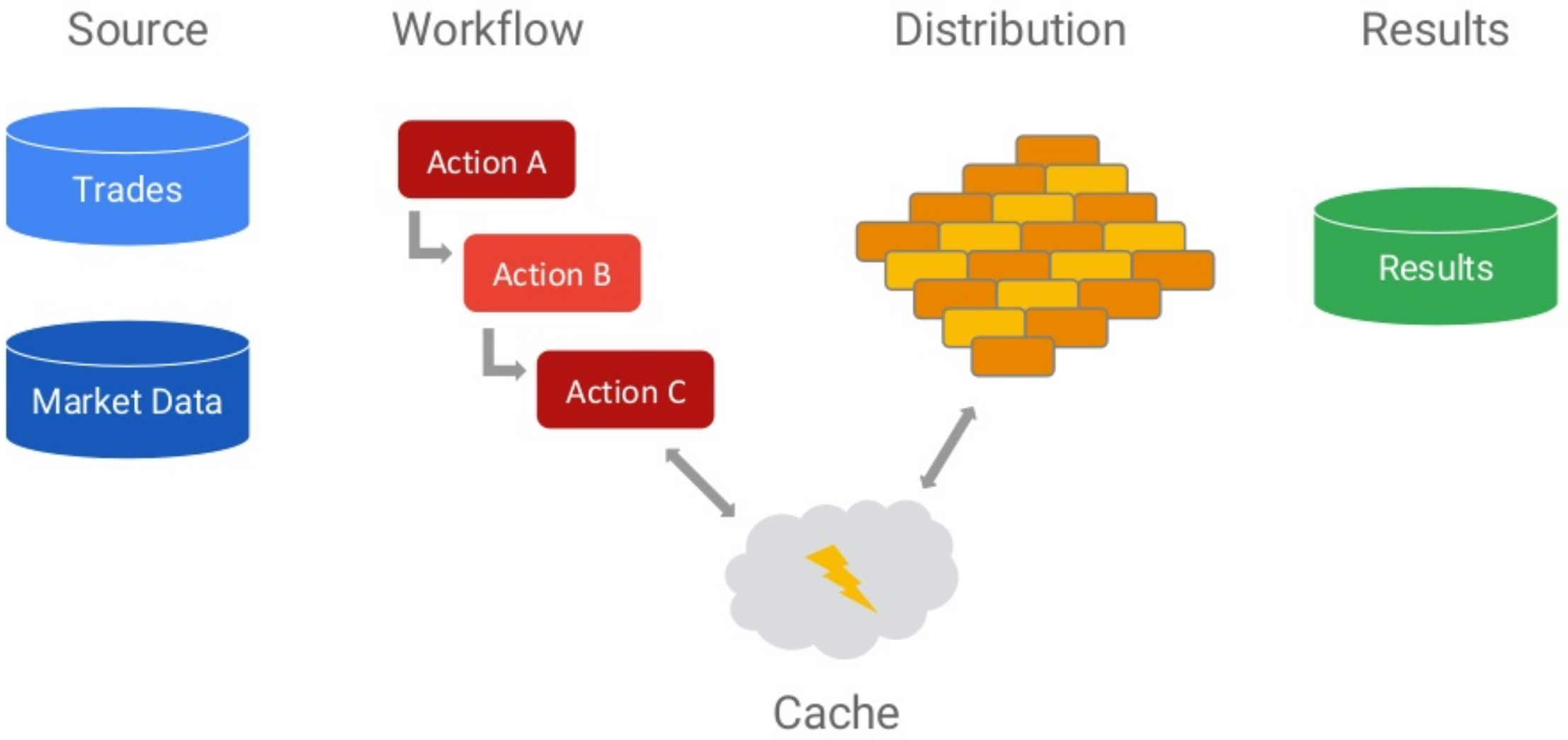
Apache Flink

What is Apache Flink?

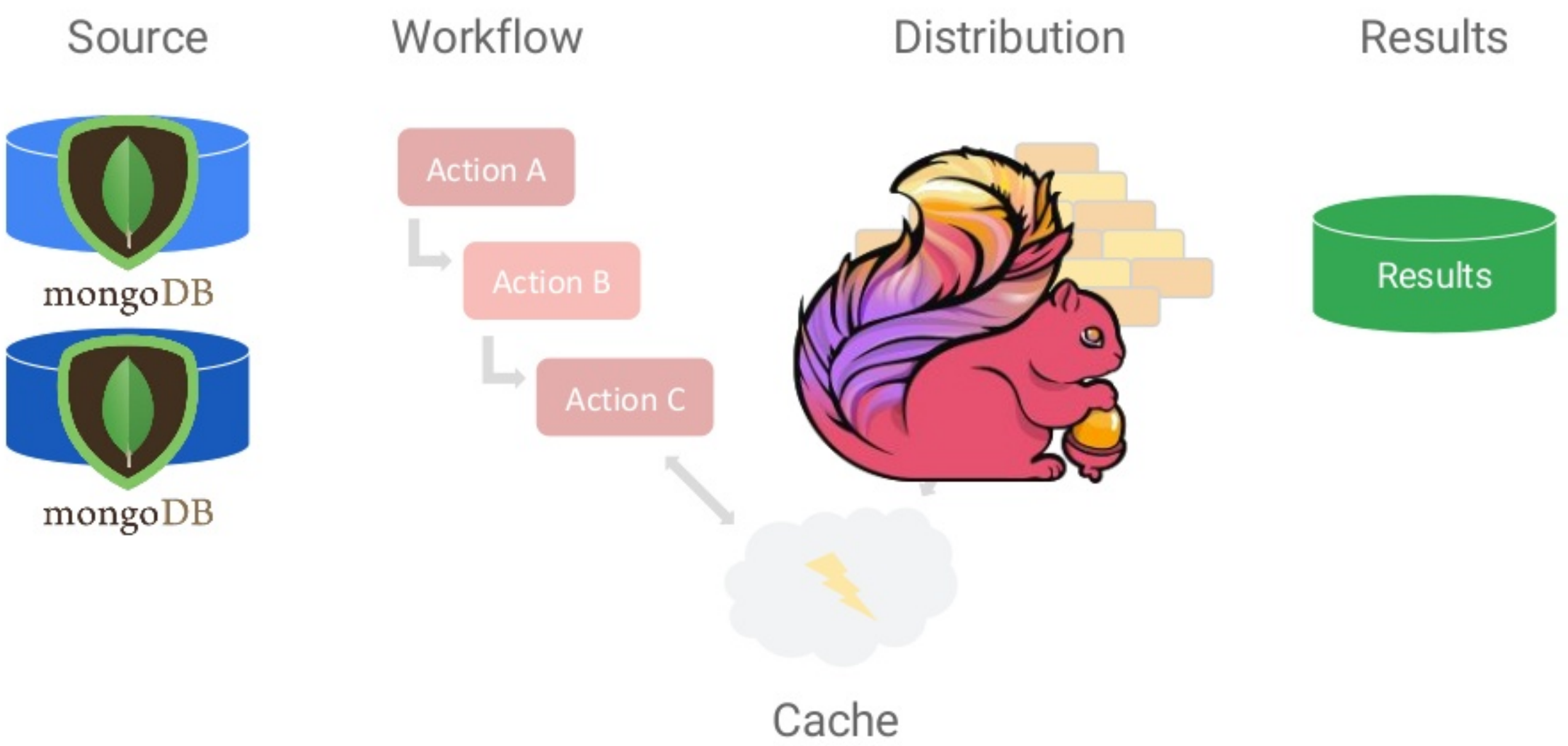
Apache Flink is open source stream processing framework

Code written for Apache Beam can run on Apache Flink

Building a Risk Engine

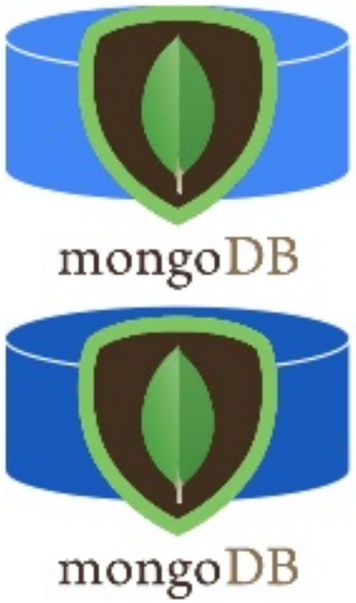


Building Blocks: Flink

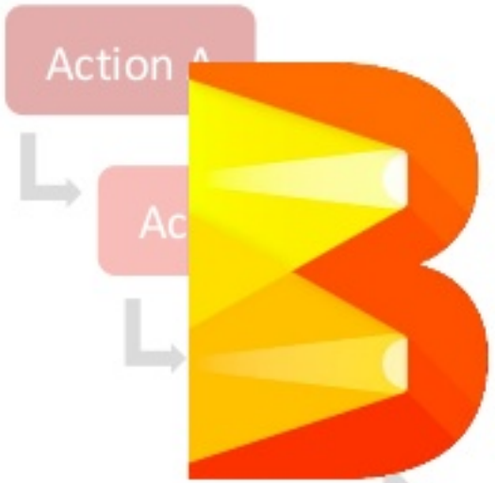


Building Blocks: Flink Running Beam

Source



Workflow



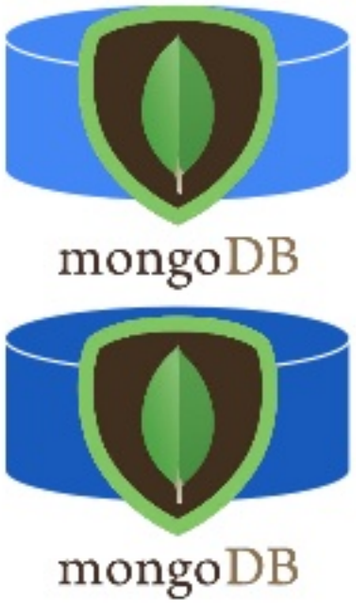
Distribution

Results

Cache

Building Blocks: Disk

Source



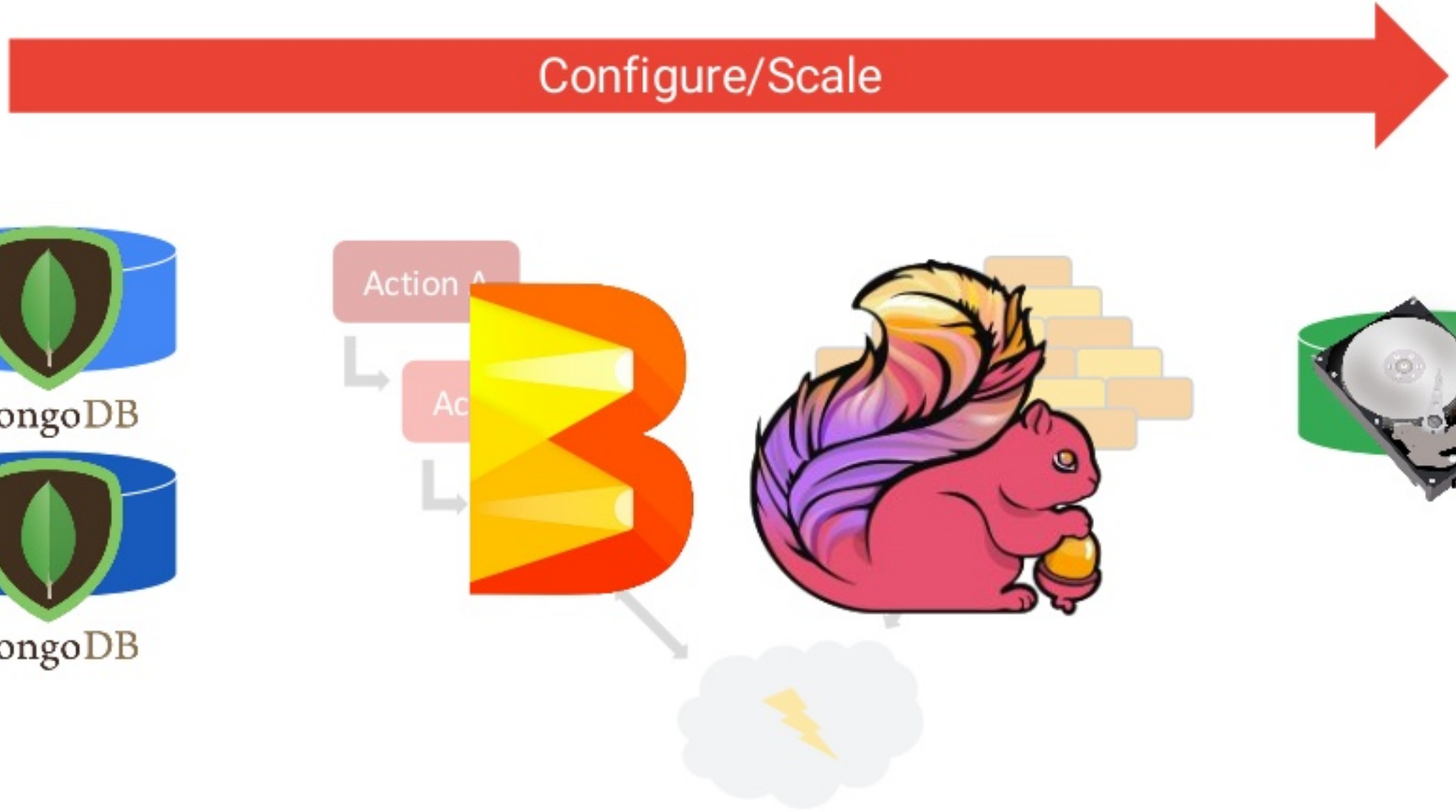
Workflow

Distribution

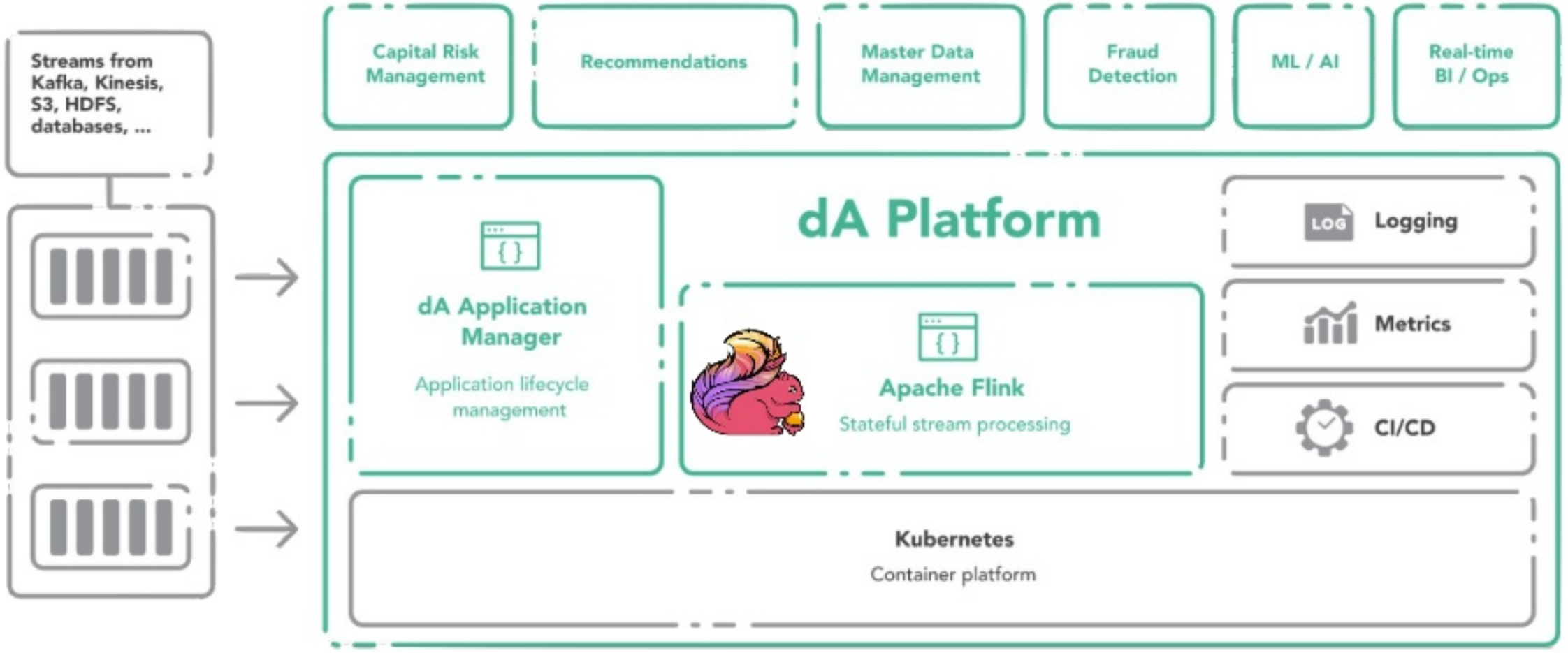
Results

Cache

What do You Need to do?



I Really Need to Run on Premises

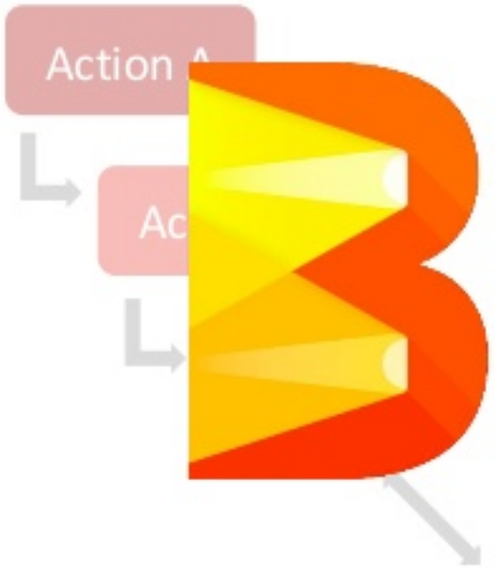


https://data-artisans.com/da-platform-2

During Development

Source

Workflow



Distribution

Results

Cache

During Development: Local Disks

Source

Workflow

Distribution

Results

Cache

During Testing: Use the Cloud

Watch Out for Runner Differences

I need a folder of Jars in Google Cloud Storage

I need a Uber Jar

Watch Out for Runner Differences

I'll decide parallelism for you

What level of parallelism do you need?

Test Setup

Analytics: Open source Quantlib v1.9.2 (for XML over JNI)
Open source Quantlib v1.10.0 (for Protobuf/Direct calls)

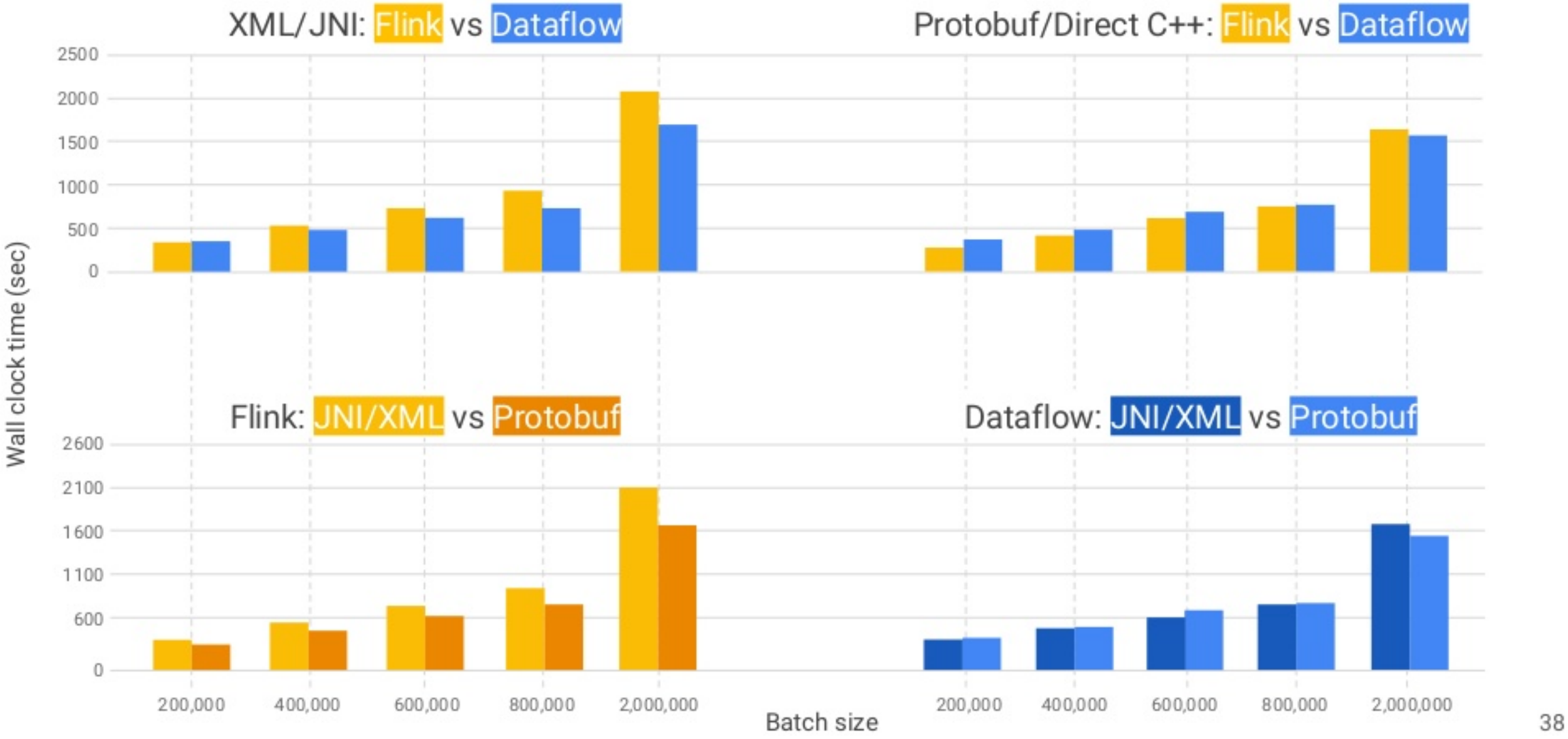
Trade data: 2,000,000 plain vanilla mono currency interest rate swaps
100,000 Bermudan Swaptions

Market data: Interest rate curves built using FRA, Futures and Swaps in 12 currencies

Batch Results

2,000,000 plain vanilla interest rate swaps
Interest rates curves from FRA, Futures & Swaps, OIS & Libor in 12 currencies
Open source Quantlib v1.10.1

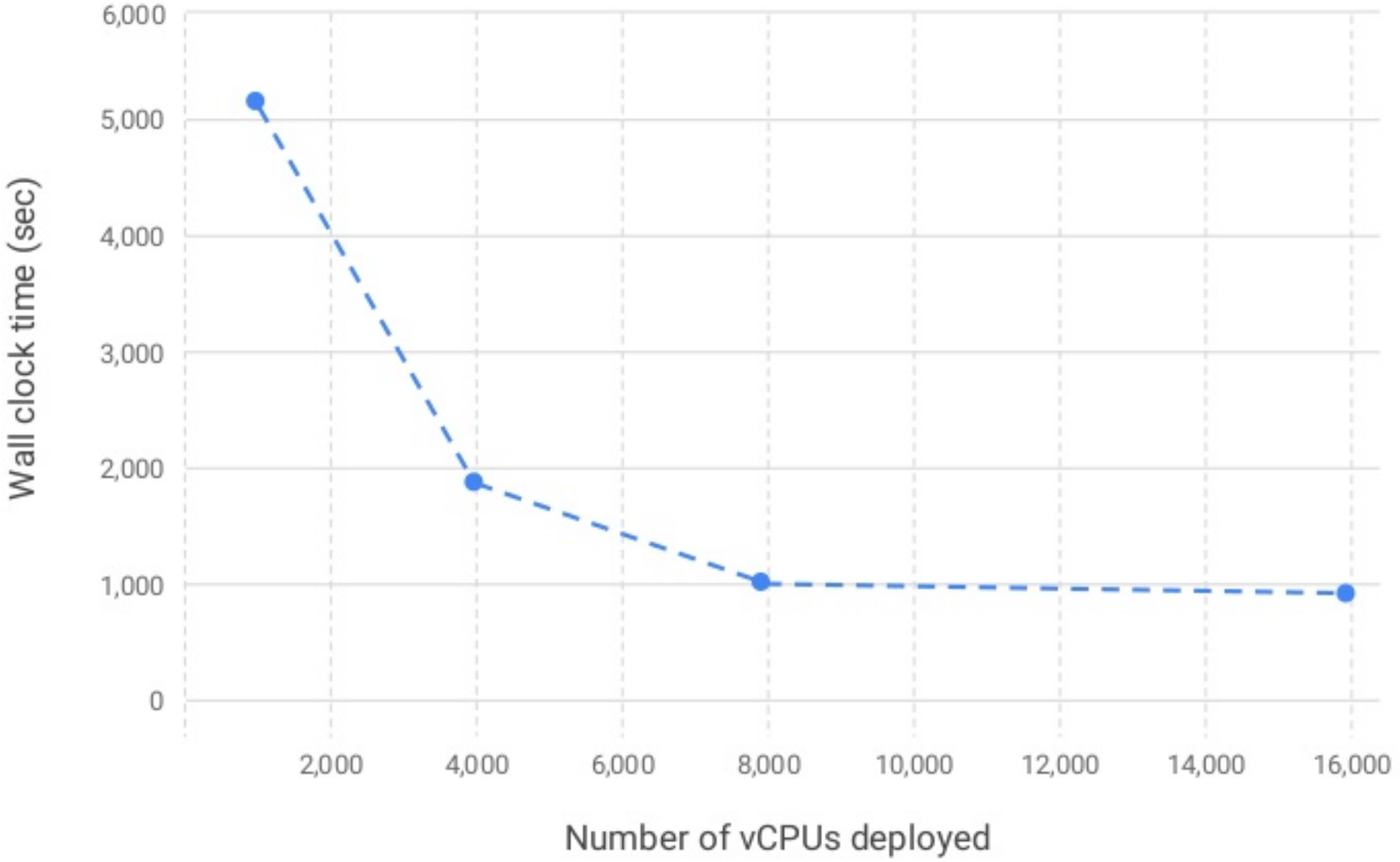
Trades
Market
Data Analytics



Scaling Out

2,000,000 plain vanilla interest rate swaps
Interest rates curves from FRA, Futures & Swaps, OIS & Libor in 12 currencies
Open source Quantlib v1.10.1

Trades
Market
Data Analytics



Scale out will depend on data structure and workflow logic

The more the workflow is controlled by Beam, the better the opportunity for dynamic rebalancing


Thank You