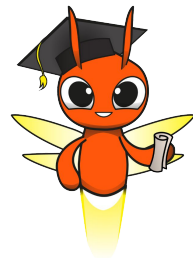




Beaming into the Future: How Apache Beam sets you up for a generative AI world

Mehran Nazir



The magic of **GenAI** captured
the world's imagination

Today organizations are
looking to turn that excitement
into **business results**

Better business results
requires leveraging business
data in **real time** with
streaming infrastructure



Your real-time data provides a competitive advantage

“This change in data from batch to real-time means the organization has to move on a heartbeat...”

Marco Argenti,
Goldman Sachs CIO, [Forbes Interview](#)



Your data is your edge:

Leveraging your unique data advantages allows you to stand out in an AI-driven environment for durable differentiation



Real-time decision making:

Make data-driven decisions in real time with your data and AI for better business outcomes

However... existing data architectures don't meet the needs of the real time AI era

19%

of companies excel at maximizing value from data

Accenture 2023,
[Cloud Data Value](#)



The lag in data processing, analyzing, inference, and reverse ETL without streaming causes teams to be unable to make data-driven decisions in a timely fashion.



Data lives in silos across disparate systems, and needs to be brought together and normalized for analysis and activation without an explosion of one-off tools



Data and AI tooling are often separate, making it more difficult to take advantage of new technologies on organizations' data assets with the appropriate infrastructure

Customers want an **easy-to-use** and **scalable** platform that accelerates **time-to-intelligence** using **AI/ML**

Explosion in real-time data

Systems, machines, partners, and other third party sources all are making real time data available, but it needs to be processed and activated for business results



New Generative AI capabilities

Leveraging the power of GenAI requires a platform to process large amounts of data in real-time with the right compute infrastructure



Higher customer expectations

Customers expect rich experiences, customized to their specific needs powered by the latest data but without increasing demands on in-house engineering teams



Your real-time data provides a competitive advantage

Real time visibility



Real time app & infrastructure monitoring

Audit & security

Optimize system performance

Anomaly detection

Automated system to resolve customer issues

Improve customer experience

Real time predictions



Recommendations

Demand forecasting

Audience segmentation

Predict customer lifetime value

Churn prediction

Prediction using vision analytics

Real time activation



Dynamic product offers

Flexible inventory management

Marketing automation

Reverse ETL

Customer 360

Logistics optimization



Apache Beam

What is Apache Beam?



Beam

Open source programming model

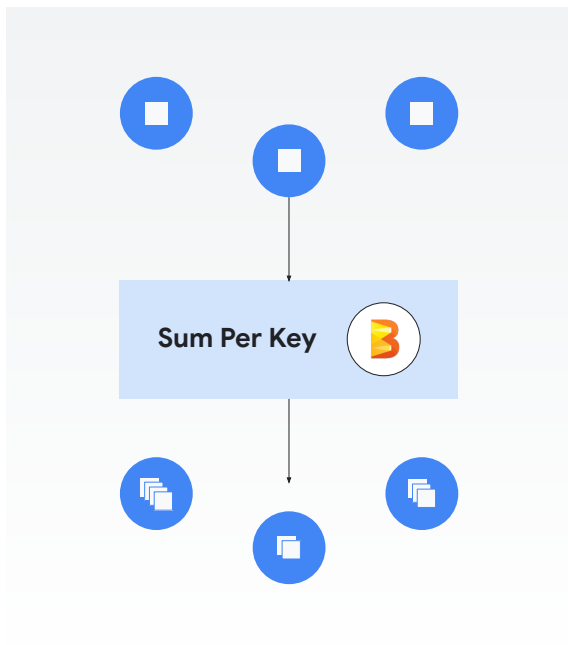
Top Apache project by dev@ activity

Unified batch and streaming

Runner and language portability

Apache Beam

Apache Beam



Apache Beam

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

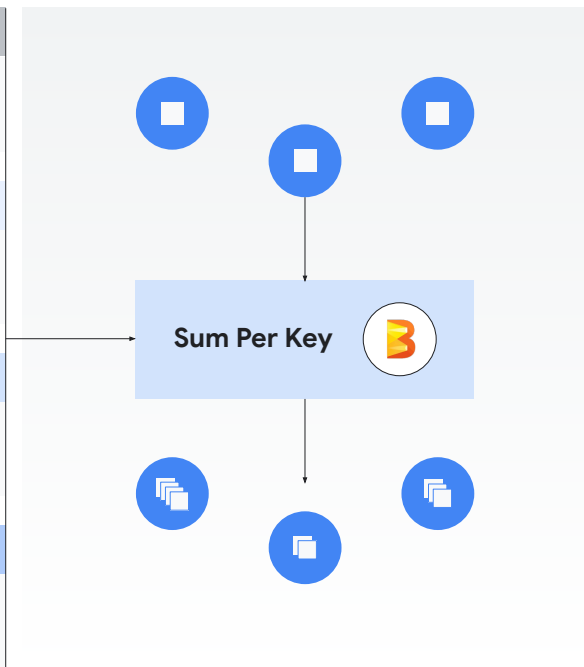
```
input | Sum.PerKey()
```

SQL

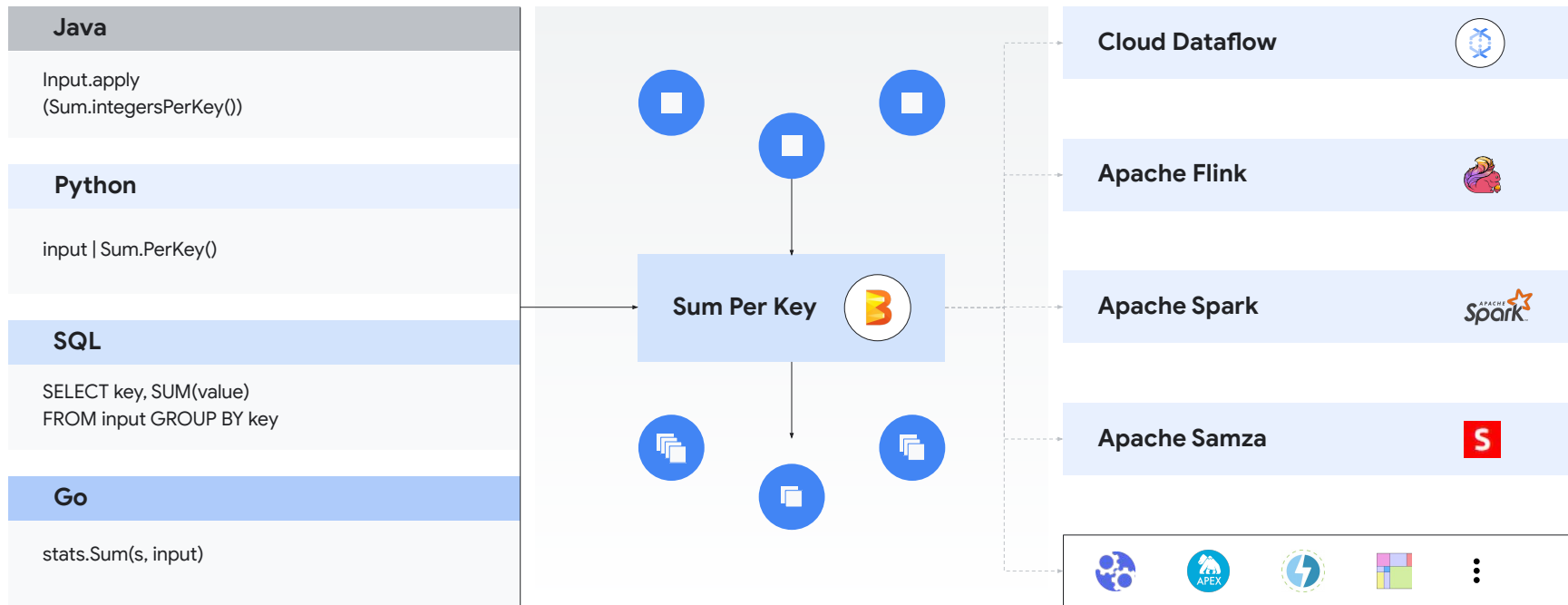
```
SELECT key, SUM(value)  
FROM input GROUP BY key
```

Go

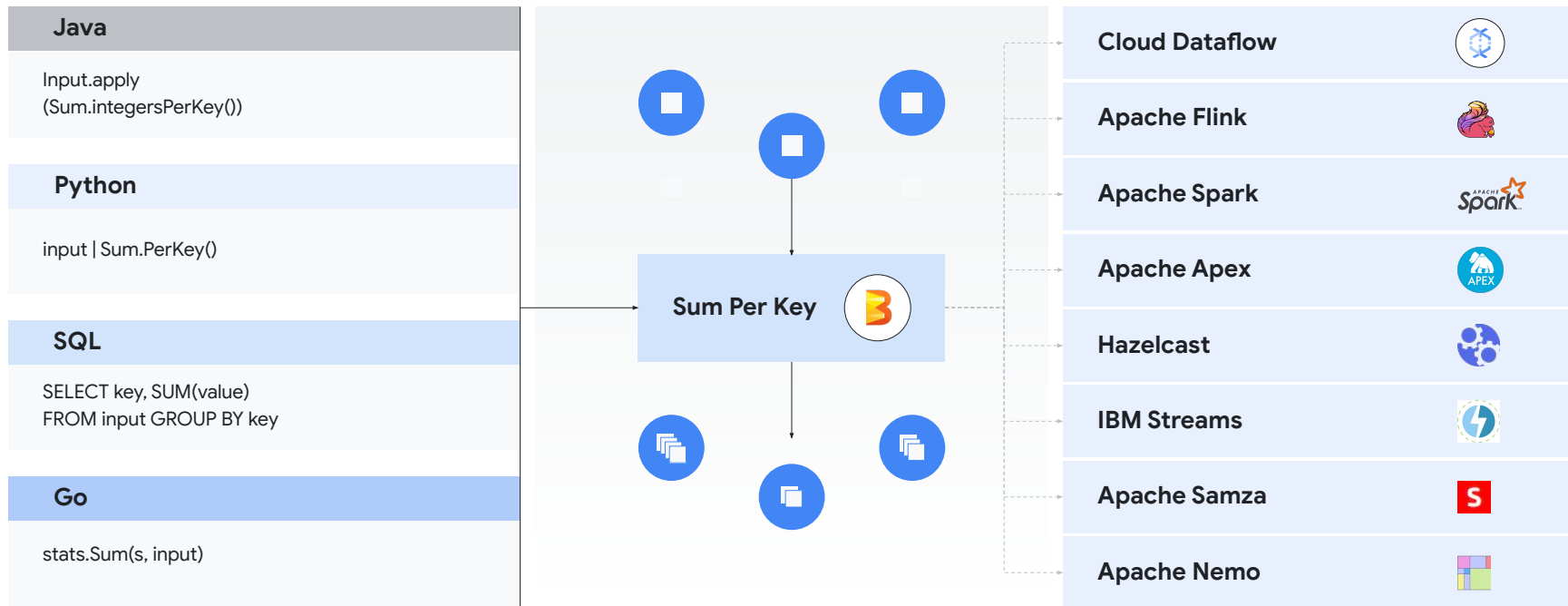
```
stats.Sum(s, input)
```



Apache Beam



Apache Beam: The Portability Vision



Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

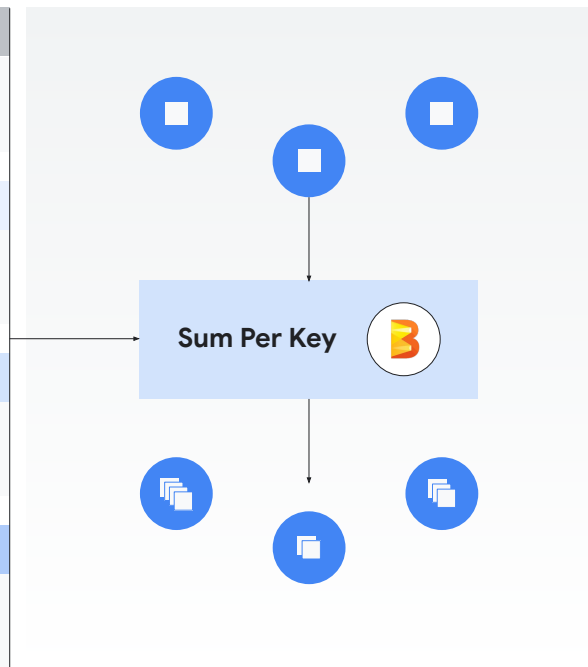
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios

Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

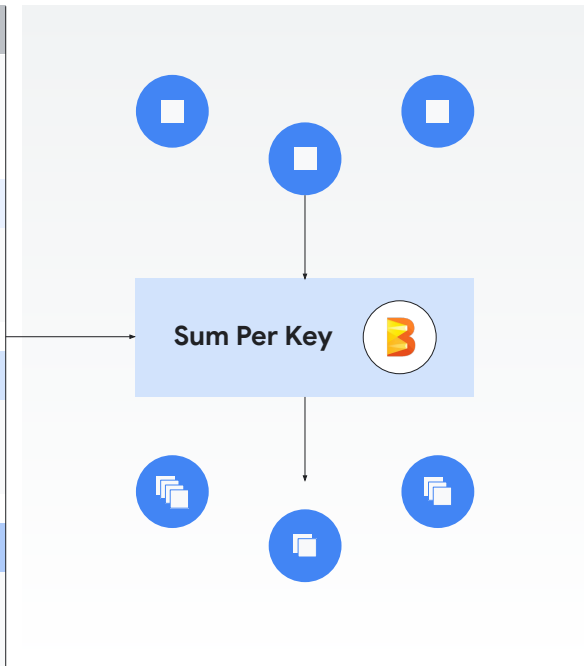
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios
- Portability - the ability to port the same workload across clouds, on-prem, and edge

Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

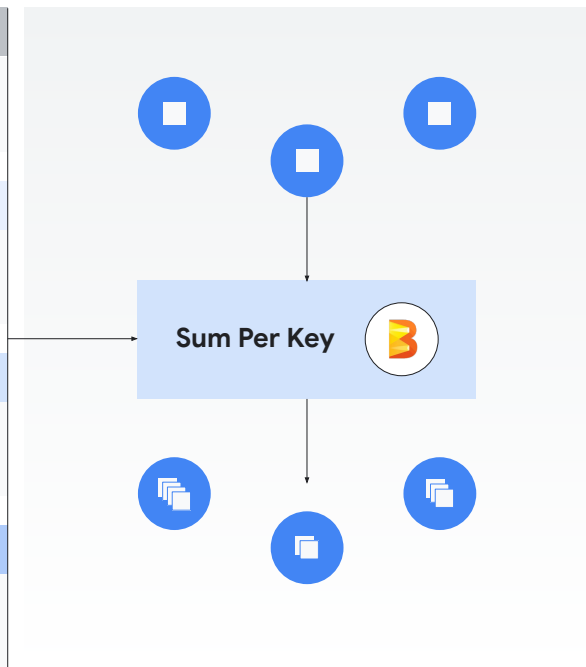
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios
- Portability - the ability to port the same workload across clouds, on-prem, and edge
- State & timer APIs

Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

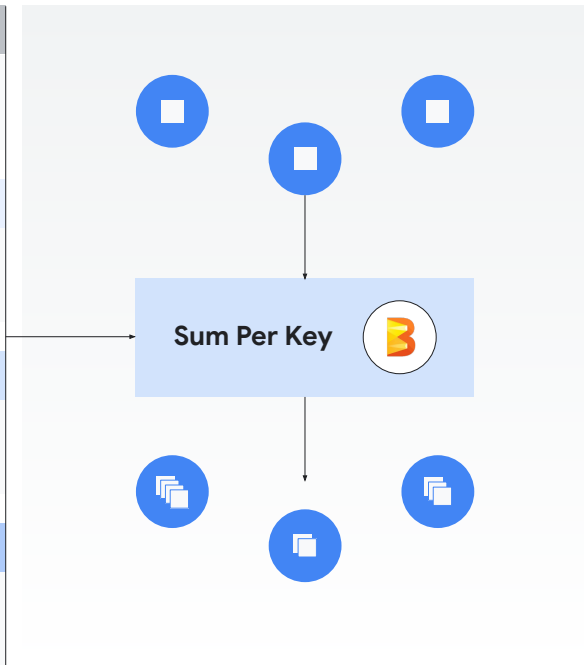
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios
- Portability - the ability to port the same workload across clouds, on-prem, and edge
- State & timer APIs
- Build your execution environment with custom containers

Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

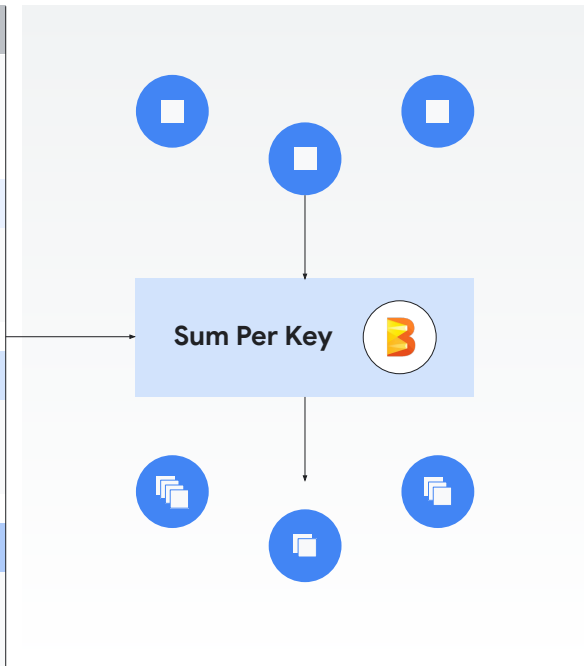
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios
- Portability - the ability to port the same workload across clouds, on-prem, and edge
- State & timer APIs
- Build your execution environment with custom containers
- Cross language pipelines

Apache Beam: Benefits

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

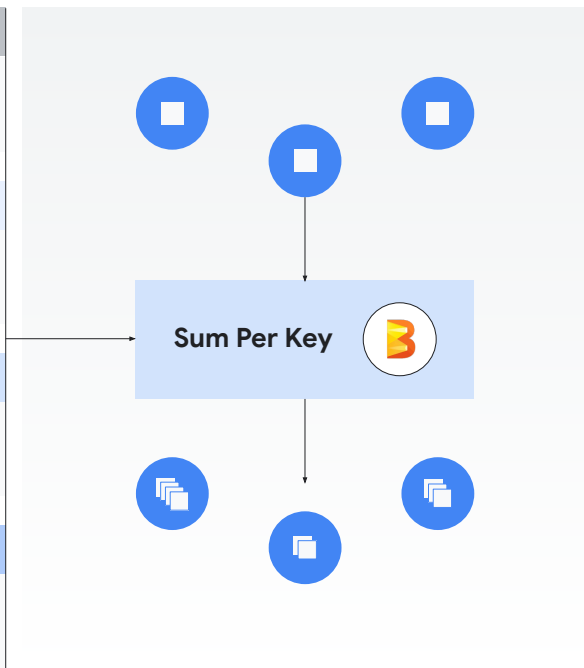
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Unique benefits

- Unified API for both batch and streaming scenarios
- Portability - the ability to port the same workload across clouds, on-prem, and edge
- State & timer APIs
- Build your execution environment with custom containers
- Cross language pipelines
- Windowing, triggers, and sessionization capabilities

Apache Beam: Benefits

Truly universal



Only Apache Beam provides a single API that supports both batch and stream processing

Developer friendly



Developers can write code in a wide variety of languages for greater efficiency and faster time to value

Portable



Beam code runs on various compute engines, across clouds, on premises, or on the edge



Apache Beam ML & Turnkey Transformations

Beam primitives are great but....

Java

```
Input.apply  
(Sum.integersPerKey())
```

Python

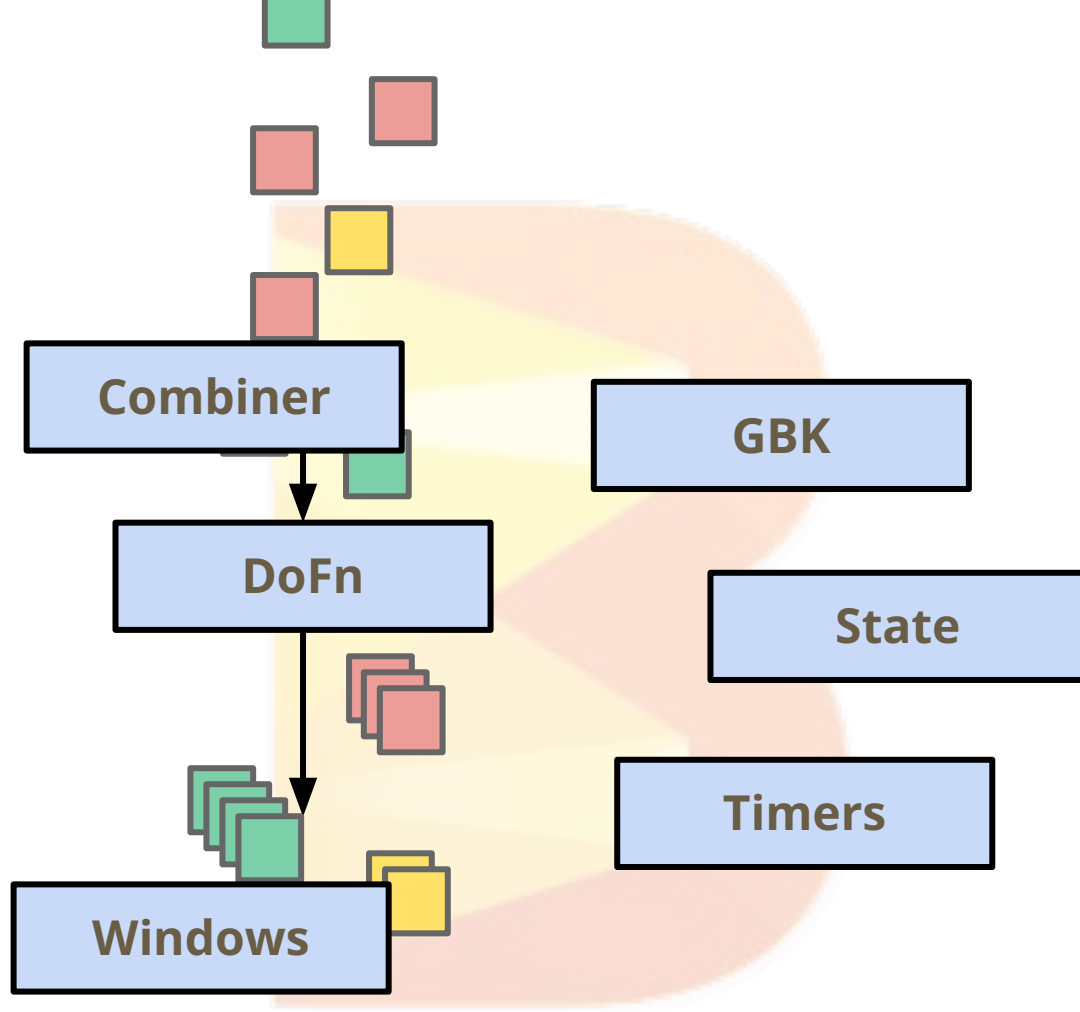
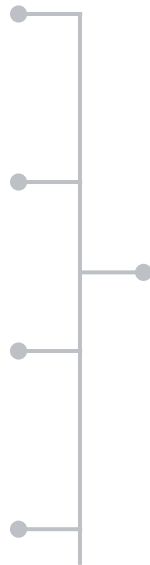
```
input | Sum.PerKey()
```

SQL

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

Go

```
stats.Sum(s, input)
```



Turnkey Transformations: accelerating developer productivity



Building blocks



Goal

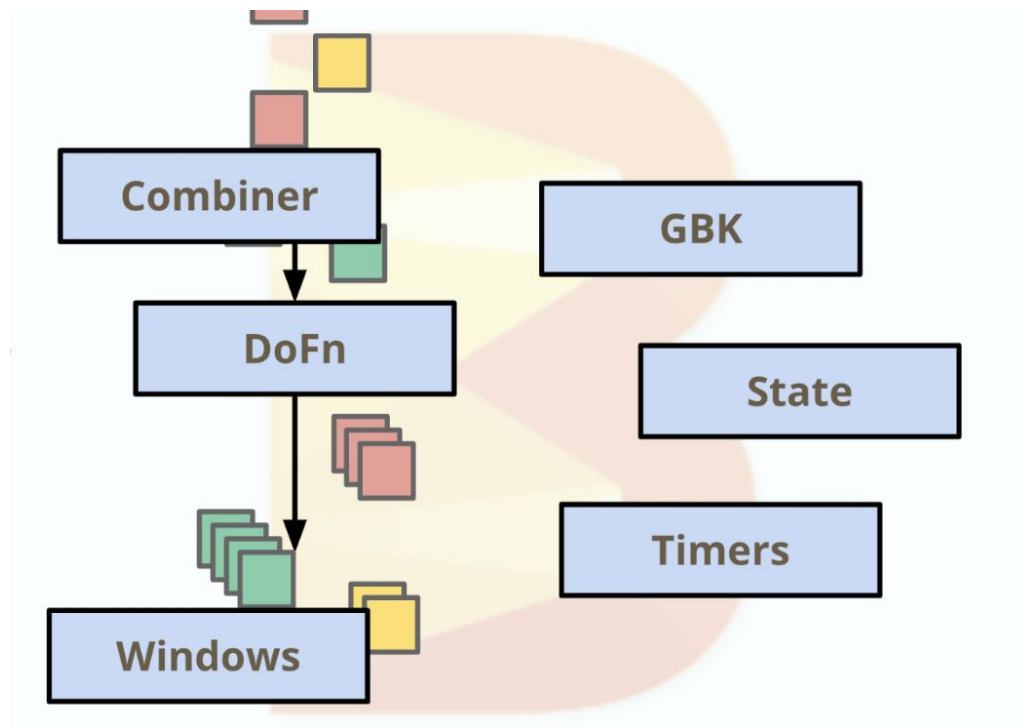
Turnkey Transformations: accelerating developer productivity

```
with beam.Pipeline(options=pipeline_options) as p:  
    (p  
     | beam.io.fileio.MatchFiles(gs://my_bucket/*)  
     | beam.io.fileio.ReadMatches()  
     | beam.Map(preprocess_image)  
     | beam.xxx_pattern_xxx(Configuration)  
     ...
```

Prediction / Inference in production

Productionisation

- Efficient sharing of the model
- Error handling and dead letter queues
- Comprehensive set of metrics for observability
- Pre and post processing encapsulation
- Configuration rather than code for ease of reuse
- Streaming Model updates



RunInference

Streaming predictions in one line of code

```
with beam.Pipeline(options=pipeline_options) as p:  
    (p  
     | beam.io.PubSubIO(my_topic)  
     | beam.Map(preprocess)  
     | beam.ml.inference.RunInference(model_handler)  
     | beam.Map(post_process)  
     ...
```

Enrichment

Augment streaming data in one line of code

Real time recommendations;

Requires the joining of users 'live' clickstream with the historic clickstream data.

Anomaly detection;

Join the IOT telemetry data, with historic information

FSI Index building

Join the current instrument ticks, with historic information metrics to build near real time indexes

Enrichment

Augment streaming data in one line of code

```
...  
with beam.Pipeline() as p:  
    output = (p  
        ...  
        | "Create" >> beam.Create(data)  
        | "Enrich with Bigtable" >> Enrichment(lookup_database)  
        ...  
    )
```

**Apache Beam ML:
More to come!**

Q&A

—

Join us at Beam Summit this September!



Beam Summit's 7th edition of this awaited event will take place September 4th-5th in-person only in Sunnyvale, CA for professionals in real-time processing, data and AI/ML engineering. Scan the QR code now to register!



Thank you!

LinkedIn: <https://www.linkedin.com/in/mehrannazir/>

