Beam in the Age of Agents

Kenneth Knowles
Apache Beam Project Management Committee Chair
kenn@apache.org

https://s.apache.org/beam-summit-2026-age-of-agents

Agenda

01 What is Beam? (to agents)

02 What is different? (with agents)

03 What is the same? How Beam is vital (for agents)

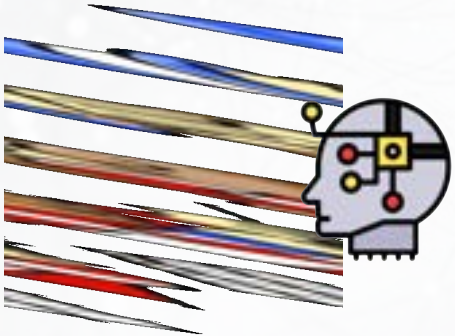
04 Coming soon: work that amplifies (agents)

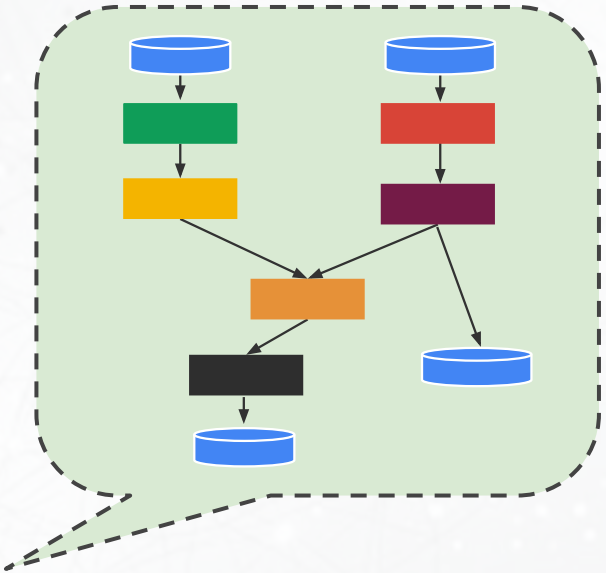
05 Future: opportunities to align (with agents)

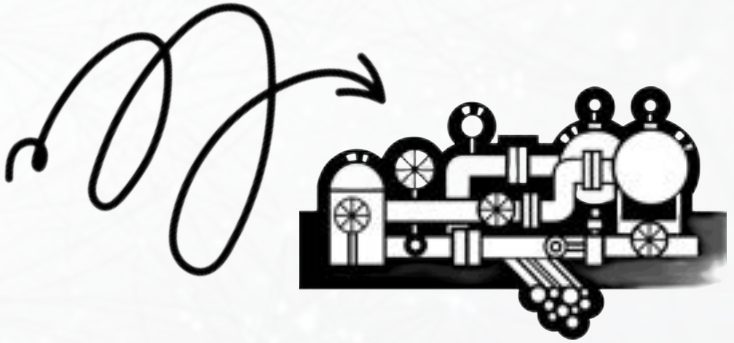
What is Beam?

(to agents)

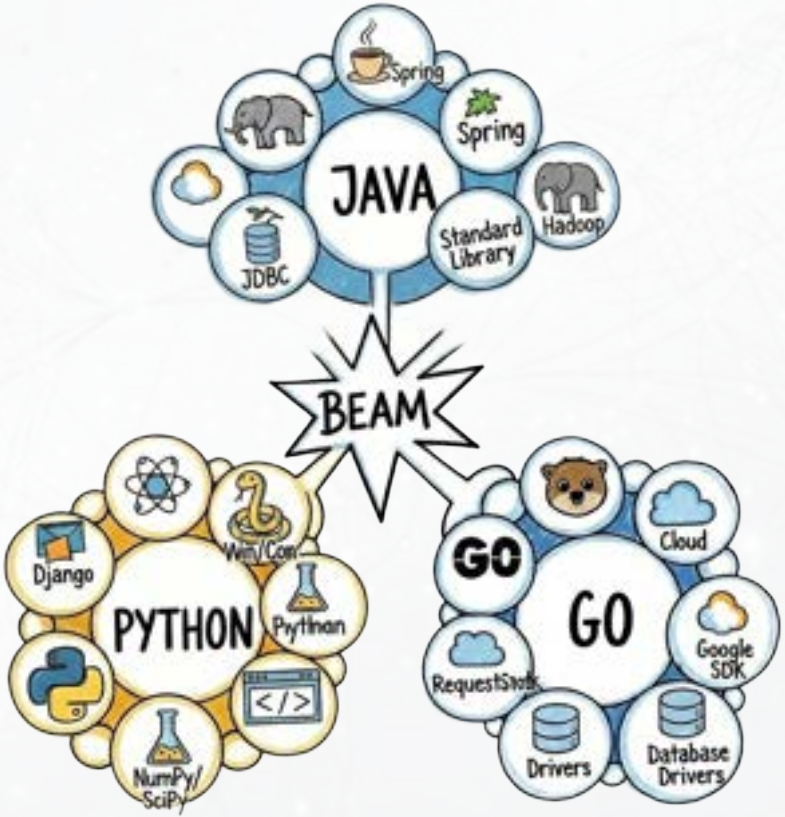
Beam is: intent



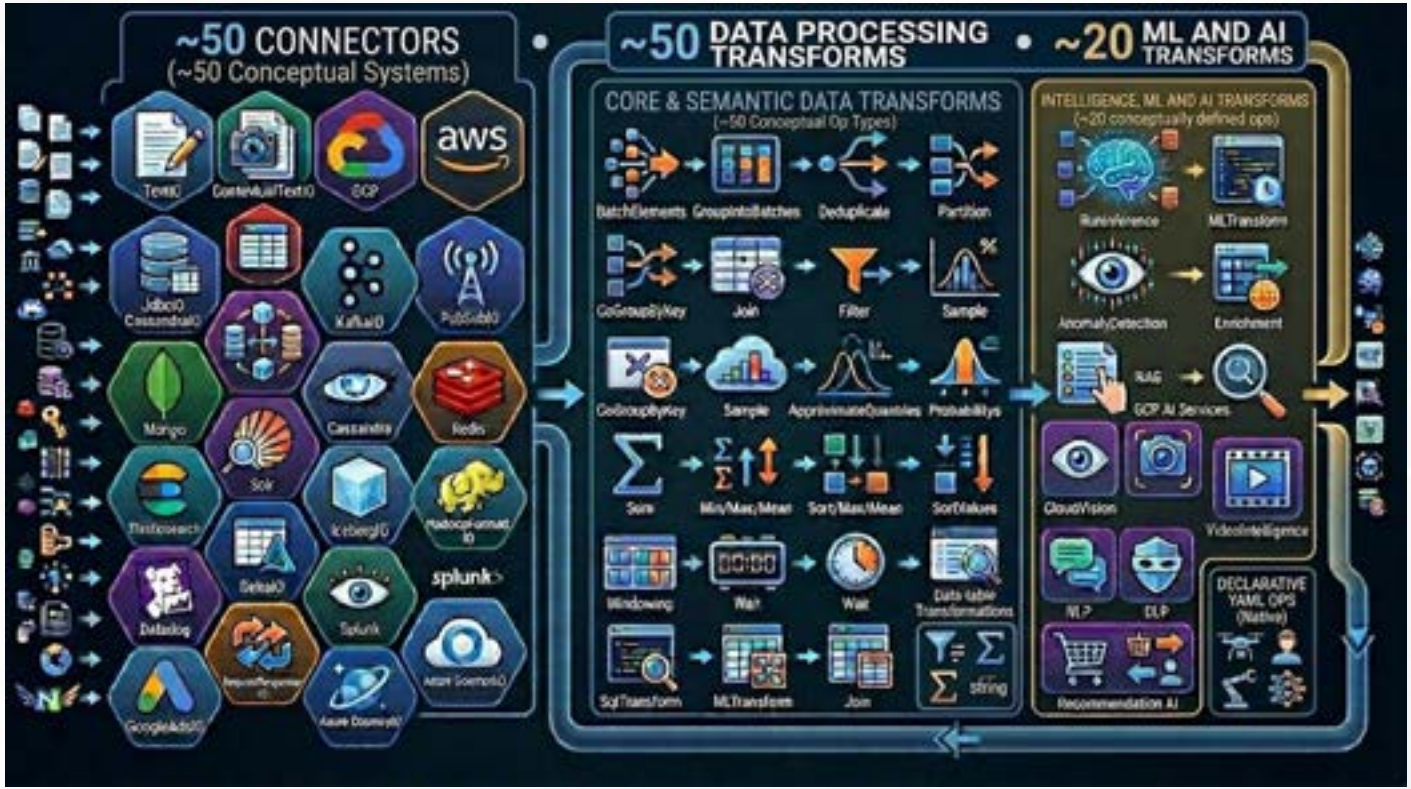




Beam is: a language bridge



Beam is: high-density tokens



What is different?

(with agents)

Different: getting started

> Write me a basic Kafka-to-Iceberg pipeline in Java

```
Pipeline p = Pipeline.create(options);

// 1. Read from Kafka
PCollection<String> kafkaMessages =
  p.apply(
      "ReadFromKafkaManaged",
      Managed.read(Managed.KAFKA).withConfig(kafkaReadConfig))

                                                    ⋮


// 4. Write to Iceberg
icebergRows.apply(
    "WriteToIceberg",
    Managed.write(Managed.ICEBERG).withConfig(icebergWriteConfig));
```

Different: language porting

> Port the pipeline to Python

```
with beam.Pipeline(options=pipeline_options) as p:
  # 1. Read from Kafka
  kafka_rows = p | "ReadFromKafkaManaged" >> managed.Read(
          managed.KAFKA, config=kafka_read_config
    )

                                                    ⋮


  # 4. Write to Iceberg using beam.managed.Write
  _ = iceberg_rows | "WriteToIceberg" >> managed.Write(
    managed.ICEBERG, config=iceberg_write_config
  )
```

Different: engine porting

> Port the pipeline to spark streaming

```
// 1. Create Spark Session with Iceberg configurations
SparkSession spark = SparkSession.builder()
                                                ⋮


// Load data from Kafka
DataStreamReader kafkaReader = spark
            .readStream()
            .format("kafka")
                                                ⋮


Dataset<Row> kafkaDF = kafkaReader.load();
                                                ⋮
// 5. Write the stream to Iceberg
StreamingQuery query = processedDF.writeStream()
  .format("iceberg")
  .outputMode("append")
  .trigger(Trigger.ProcessingTime("60 seconds")) // Configure trigger interval
                                                ⋮


  .start();
```

Different: things you always wanted to do

> I want to contribute to Beam, set up a dev environment

> Build a data generator to test this streaming transform

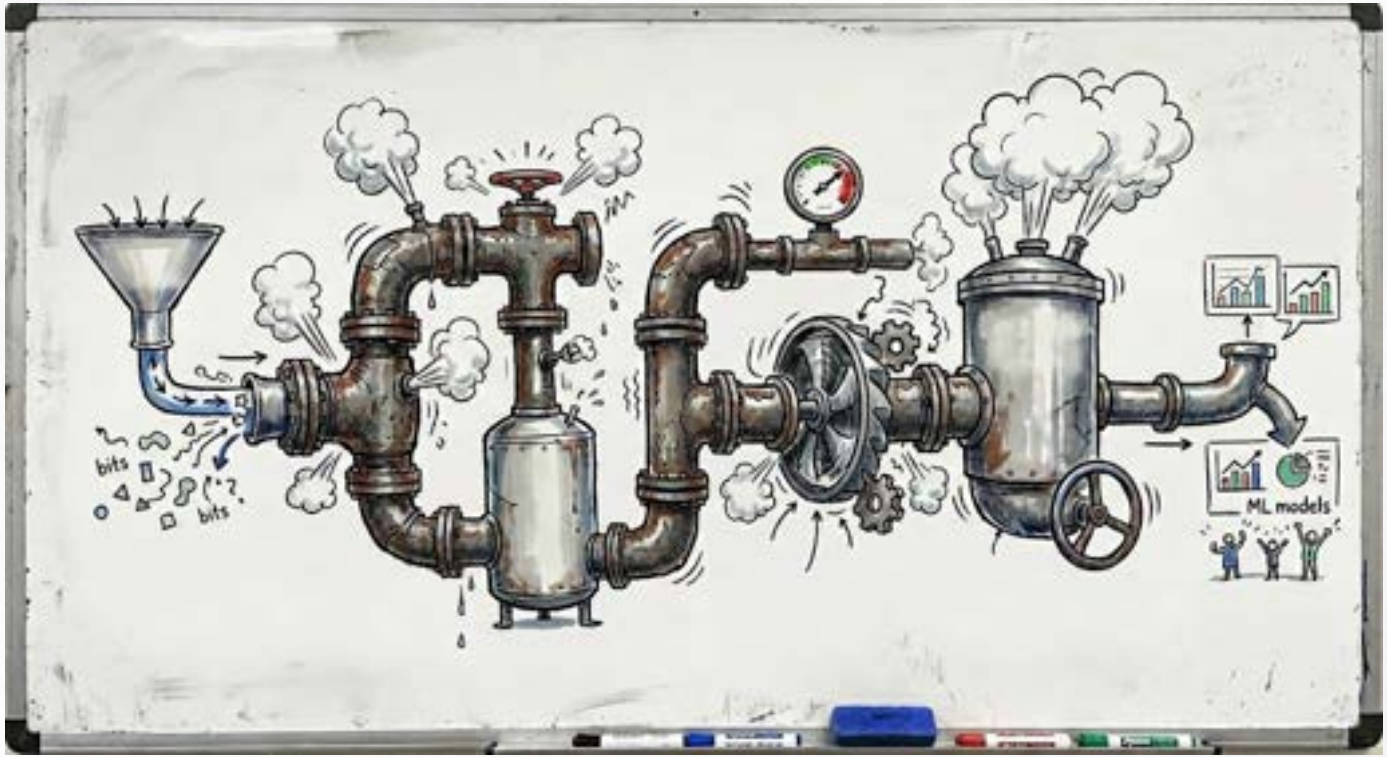
> Add type hints to my codebase

> Upgrade *<library X>* and fix any compile and test breakages, preserving semantics

What is the same?
How Beam is vital

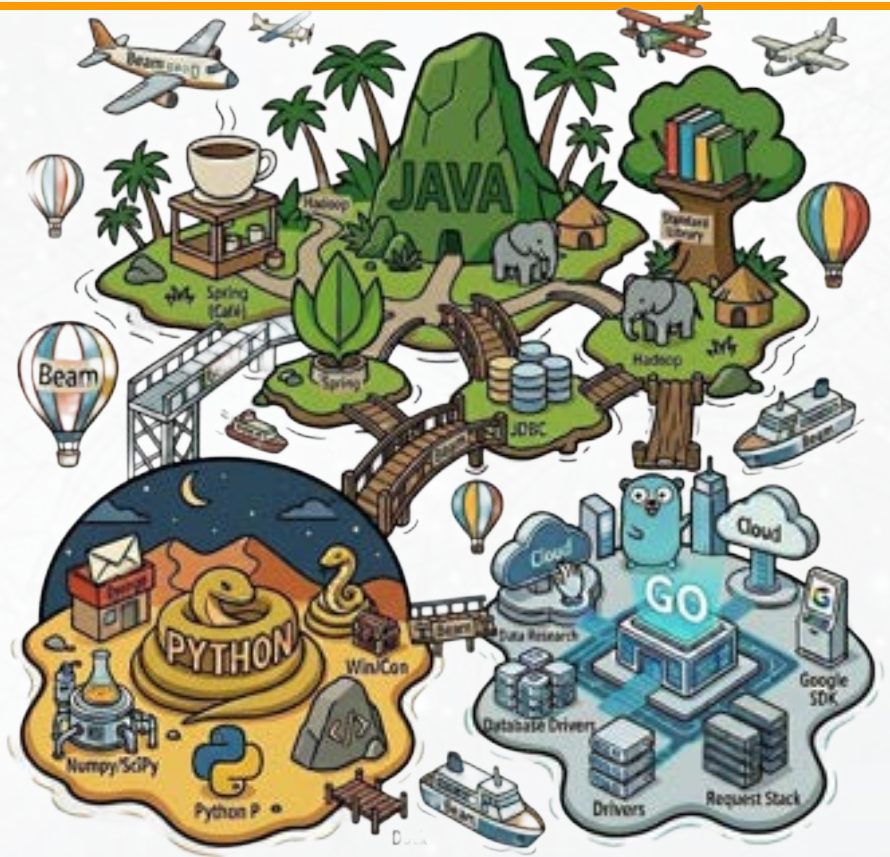
(for agents)

Same: running pipeline

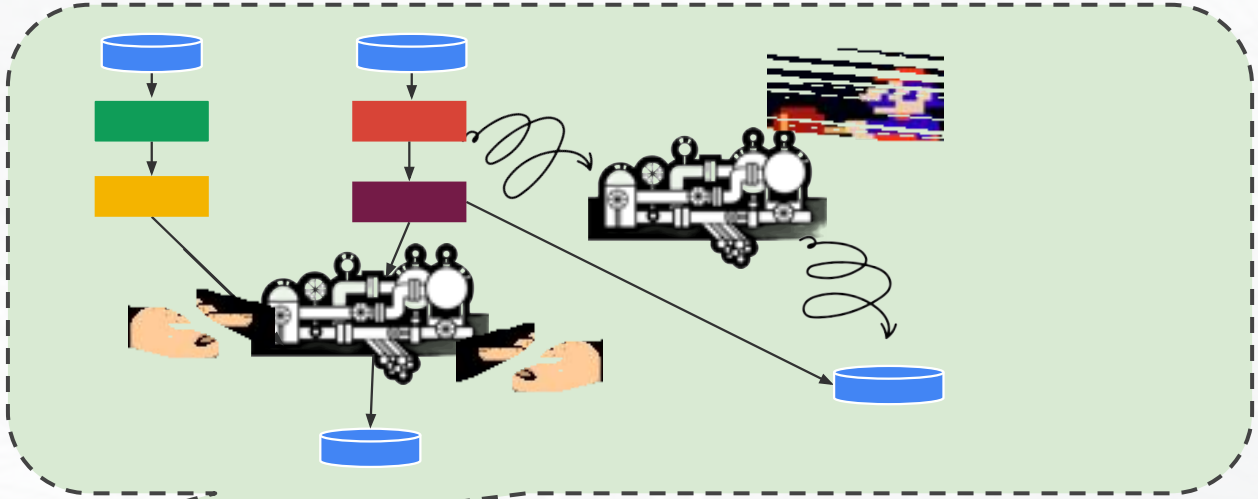


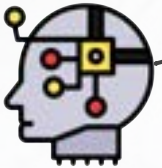
Same: languages are ecosystems

Beam: multi-ecosystem

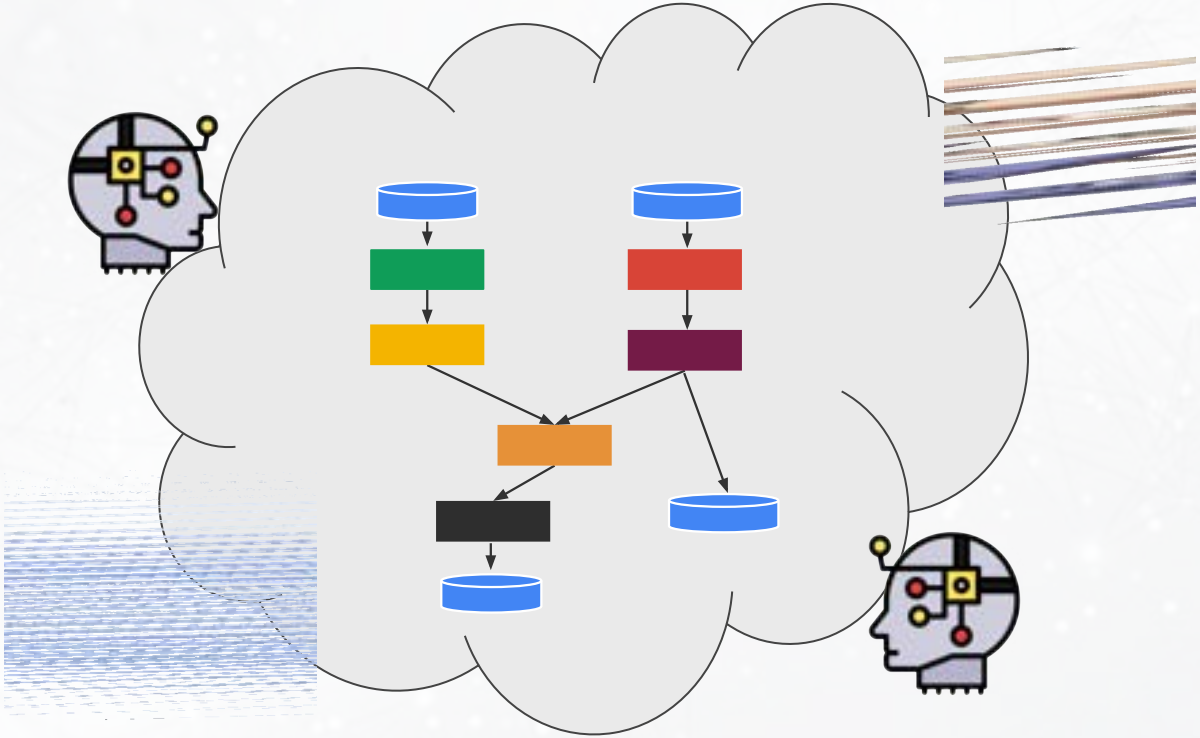


Same: engines aren't intent

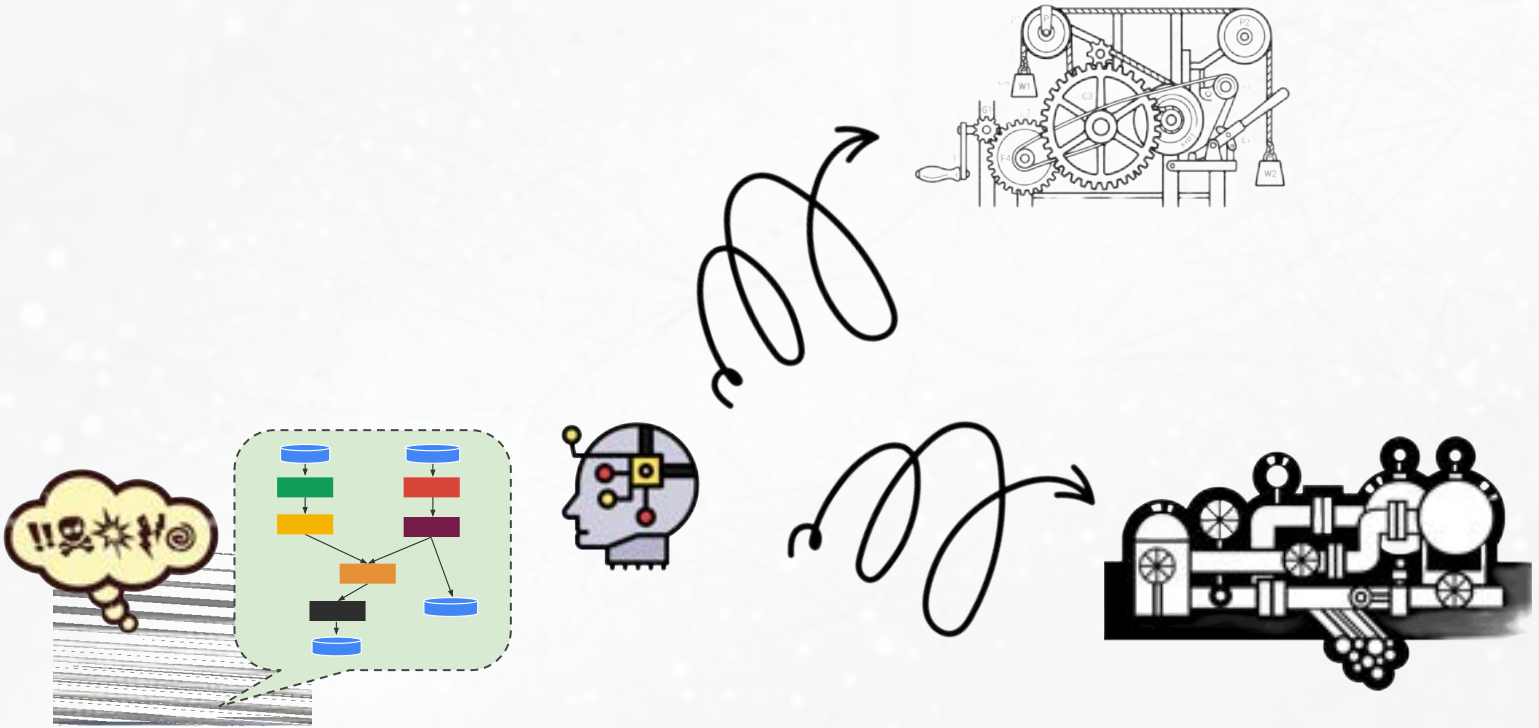




Beam: semantic APIs



Beam: Engine portability

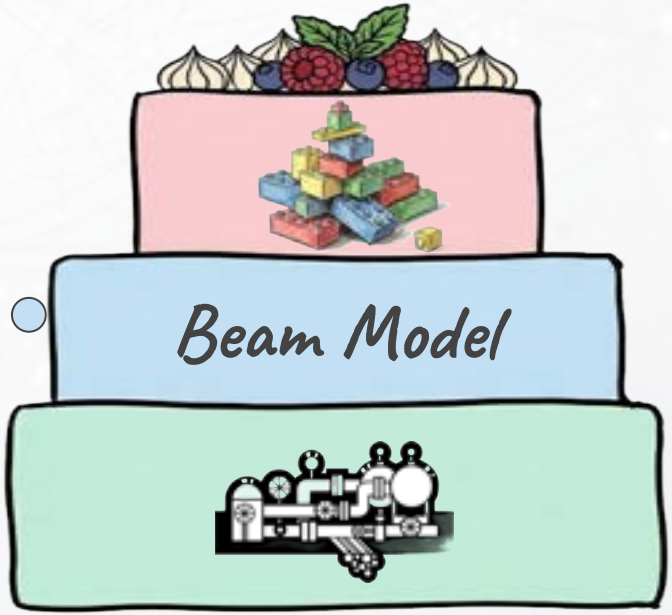


Coming soon: work that amplifies

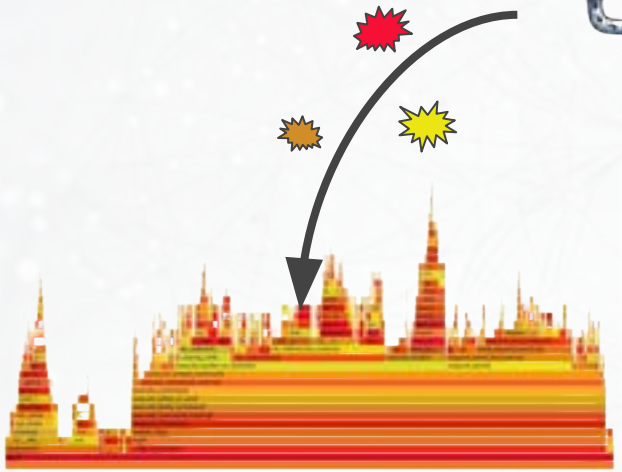
(agents)

Core model features

- Built-in CDC
- Per-element metadata
- State types
- DDL / Catalog



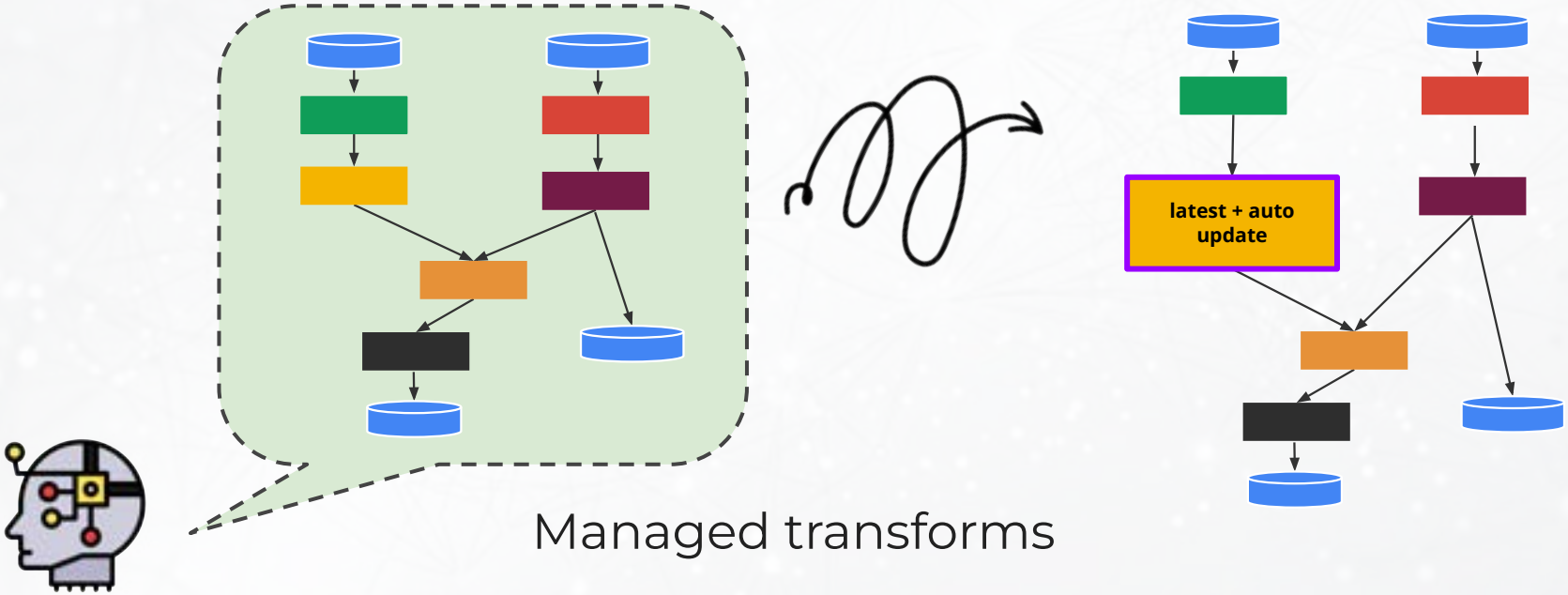
Observability



Python memory profiling

OpenTelemetry tracing

No knobs



Managed transforms

High-density tokens

RemoteInference

DeltaIO

BigQueryVectorWriter[Config]

DatadogIO

QdrantWrite[Config]

ADKModelHandler

PostgresVectorWriter[Config]

SpannerVectorWriter[Config]

AlloyDBVectorWriter[Config]

EnvoyRateLimiter

ReadBigQueryChangeHistory

ReadChangeStreamFromSpanner

AnthropicModelHandler

VertexAIModalEmbeddings

VertexAIImageEmbeddings

CloudSQLEnrichmentHandler

MilvusSearchEnrichmentHandler

OpenAIModelHandler

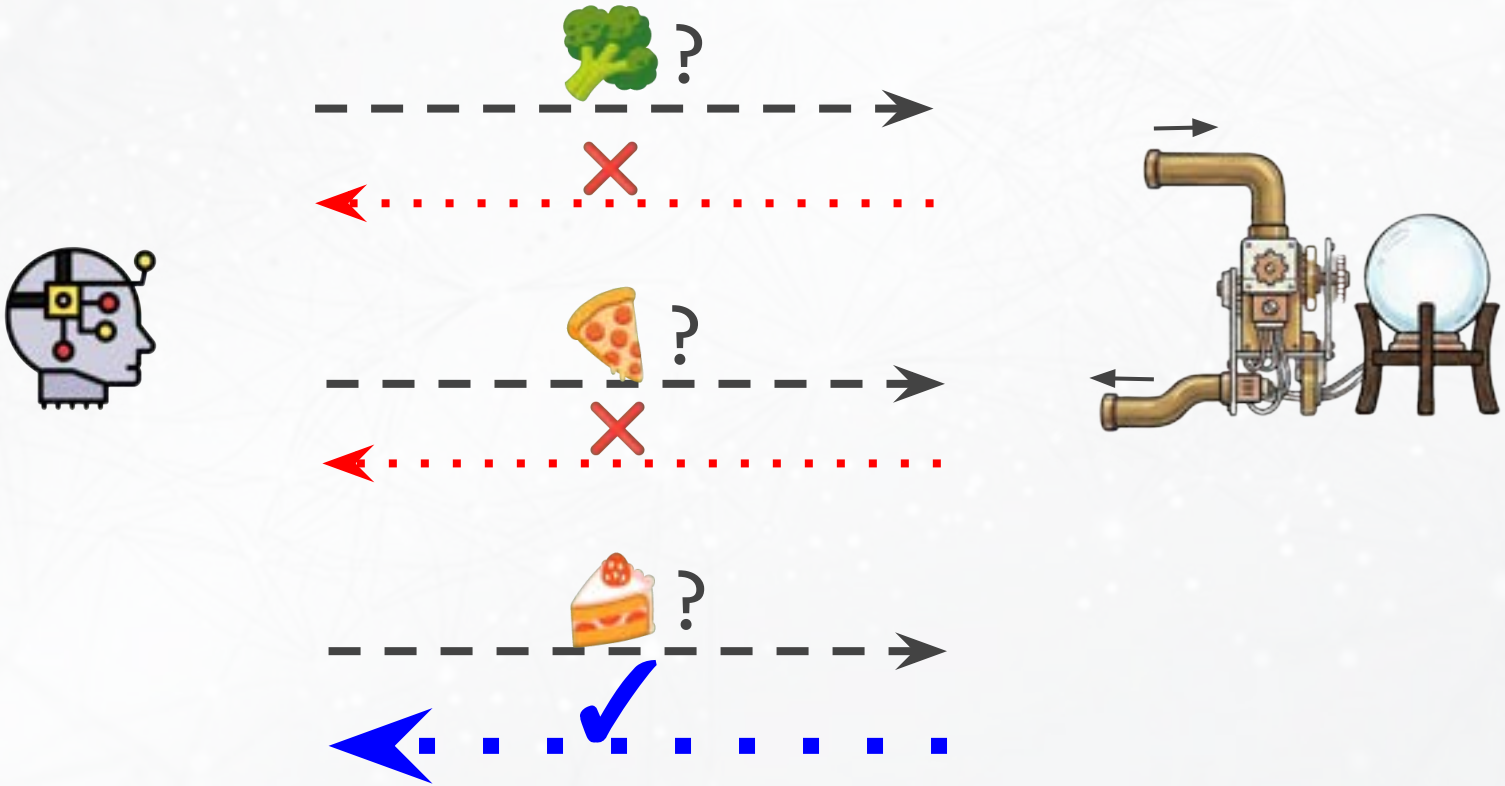
High-density tokens

HuggingFaceModelHandler
RemoteInference
DeltaIO
RunInference
BigQueryVectorWriter[Config]
RabbitMqIO
ADKModelHandler
DatadogIO
QdrantWrite[Config]
JdbcIO
PostgresVectorWriter[Config]
DebeziumIO
SpannerIO
SpannerVectorWriter[Config]
BigTableIO
AlloyDBVectorWriter[Config]
EnvoyRateLimiter
VLLMChatModelHandler
MqttIO
ReadBigQueryChangeHistory
ReadChangeStreamFromSpanner
KafkaIO
GeminiModelHandler
AnthropicModelHandler
KinesisIO
VertexAIModalEmbeddings
ElasticSearchIO
VertexAITextEmbeddings
TextIO
VertexAIImageEmbeddings
CloudSQLEnrichmentHandler
SolaceIO
RemoteModelHandler
IcebergIO
MilvusSearchEnrichmentHandler
MongoDbIO
MLTransform
RecommendationsAICatalog
PulsarIO
BigQueryEnrichmentHandler
OpenAIModelHandler
VLLMCompletionsModelHandler
ParquetIO

Future: opportunities to align

(with agents)

Verifiability



Verifiability

- Imagine agents as raccoons
  - Being able to "just try stuff forever" allows them to do level N+1 work
- Even more typing everywhere / more static analysis
  - already using: findbugs, checkerframework, python typing, beartype, ...
- Faster test runs
  - Make Prism-based local tests lightning fast
  - Highly limited, very cheap single-language test runners
- Exhaustive and/or randomized testing
  - Prevents agents overfitting code to your tests
- High fidelity errors
- Make more things verifiable in easier ways
  - Beam linter?

Simplicity

- More rapid & complete deprecation/update/removal cycle
  - else LLMs generate non-best-practice code
- Make more things "just code"
  - less container management, config, etc
  - easier to test
- Embrace boilerplate / "toil"
  - if the agent can do it all for you
  - if it simplifies the API surface
  - if it makes control flow straightforward
  - e.g. avoid overloading, inheritance, funny operators, magic conversions, parameters with ambiguous types
- Non-semantic breaking changes to get there?

Feed the machine: API docs

- You! Today!
- Add/improve docs for one API
- Use an agent to find one
- Use an agent to write the first draft
- Great way to "graduate" to contributor

Feed the machine: examples

- You! Today!
- Add an example for an API that needs it
- Use an agent to find one
- Use an agent to write the first draft
- Also a great first contribution!

Feed the machine: Share your experiences

- Agents choose solutions they have seen.
- We have more than enough writing on the Beam model
- We don't have enough writing on Beam solutions
- LLM output can be changed by less than 20 words

In Summary

- Now is the perfect time to embrace the change
- **Beam is potentially even more valuable in the agentic era**: deeply correct-by-construction model, multi-ecosystem
- **Double down on the things we are doing right**: model enhancements, observability, no knobs, connectors, ML transforms
- **Increase focus on things that are now more valuable**: low-latency feedback loops, constraining the space of programs, reference documentation, examples, detailed experience reports
- **It has never been easier to get started with Beam or get started contributing to Beam!**
- Thank you for being a part of the community!

Thank you!