Apache Flink:
The Latest and Greatest

Jamie Grier
@jamiegrier

dataArtisans

data-artisans.com

dataArtisans

dA
PLATFORM

Original creators of **Apache Flink®**

Providers of the **dA Platform**, a supported Flink distribution

The Latest Features

- ProcessFunction API
- Queryable State API
- Excellent support for advanced applications that are:
    - Flexible
    - Stateful
    - Event Driven
    - Time Driven

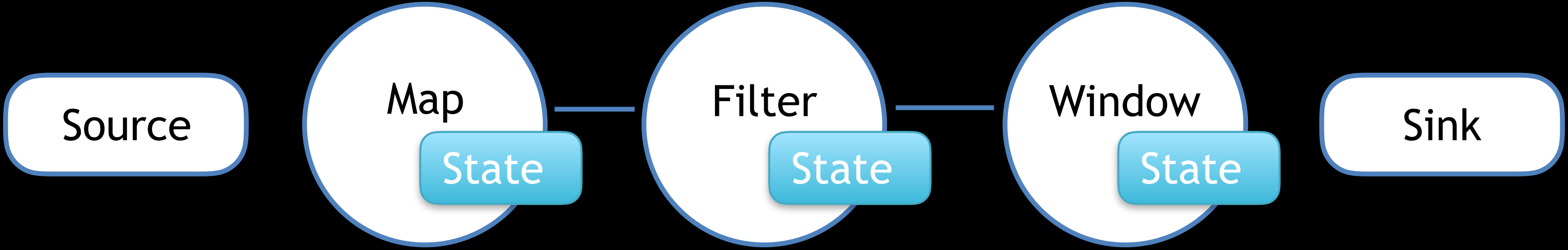
The Latest Features - Quick Overview

- Rescalable State
- Async I/O Support
- Flexible Deployment Options
- Enhanced Security

Rescalable State

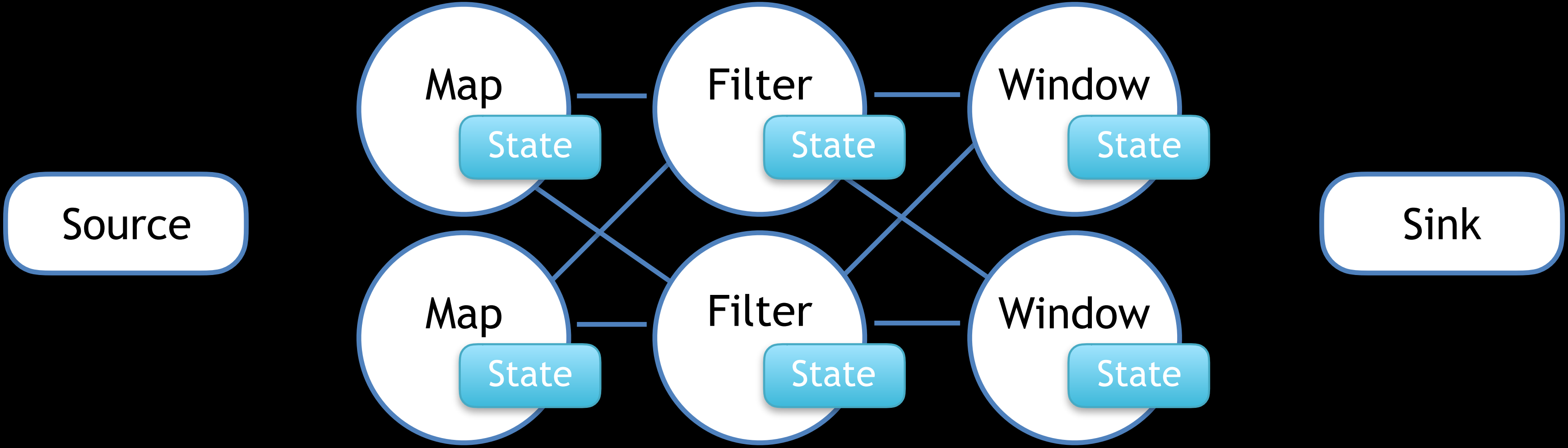
- Separates state parallelism from task parallelism
- Enables autoscaling integrations while maintaining stateful computations
- Handled efficiently via key groups

Rescalable State



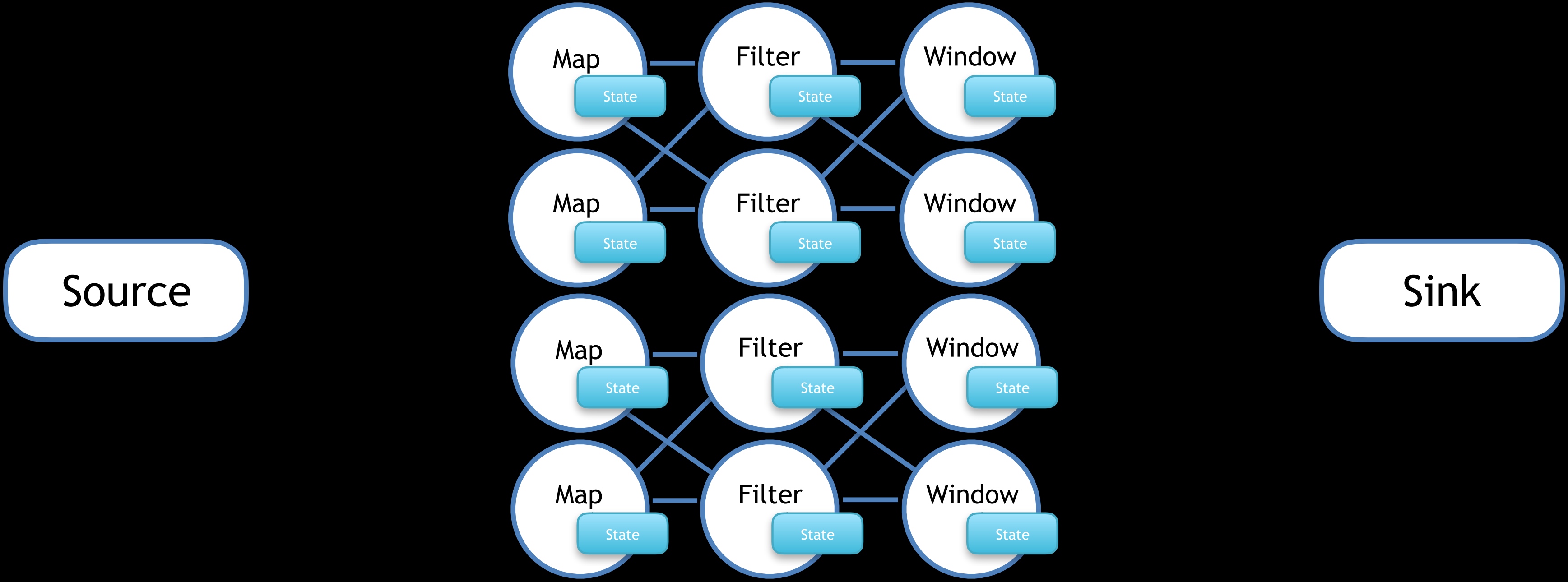
State is partitioned by key

Rescalable State



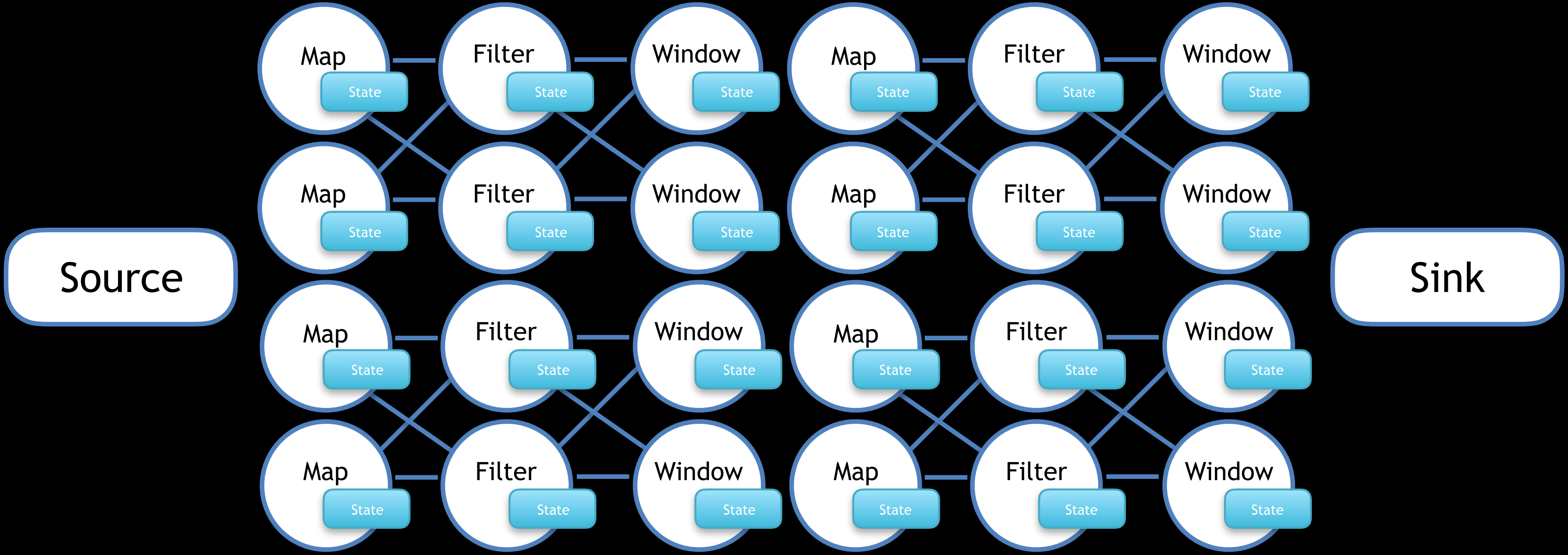
State is partitioned by key

Rescalable State



State is partitioned by key

Rescalable State



State is partitioned by key

Flexible Deployment Options

- YARN
- Mesos
- Docker Swarm
- Kubernetes

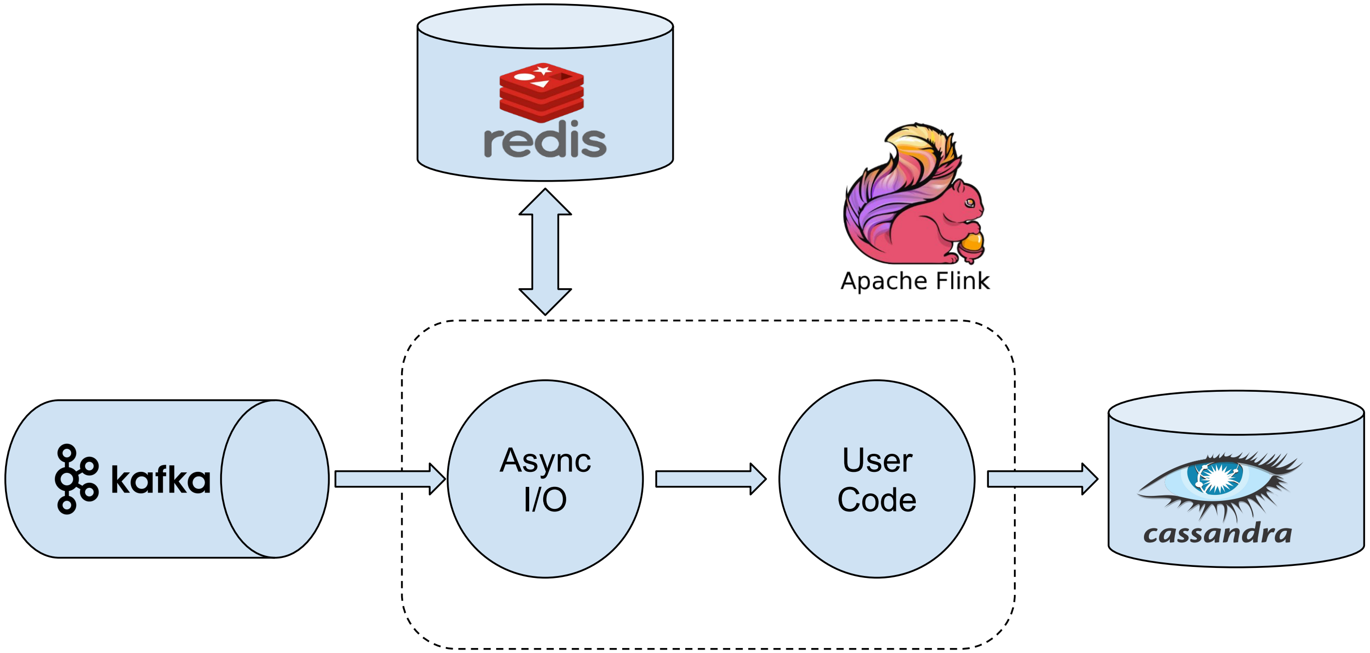
Flexible Deployment Options

- DC/OS
- Amazon EMR
- Google Dataproc

Asynchronous I/O Support

- Make aynchronous calls to external services from streaming job
- Efficiently keeps configurable number of asynchronous calls in flight
- Correctly handles failure scenarios - restarts failed async calls, etc

Asynchronous I/O Support

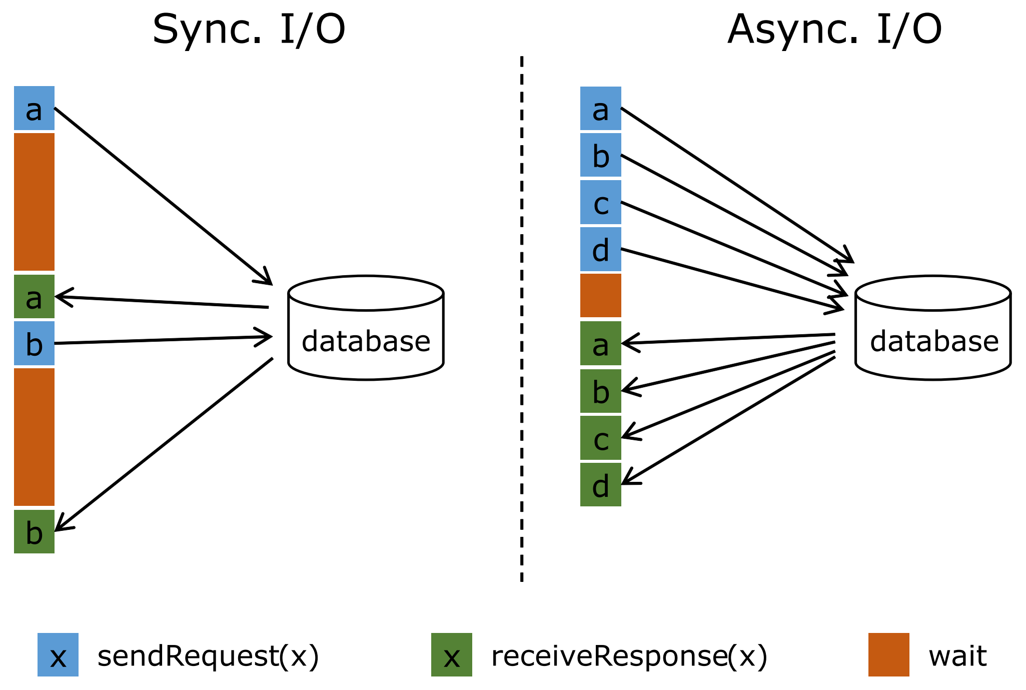


Asynchronous I/O Support

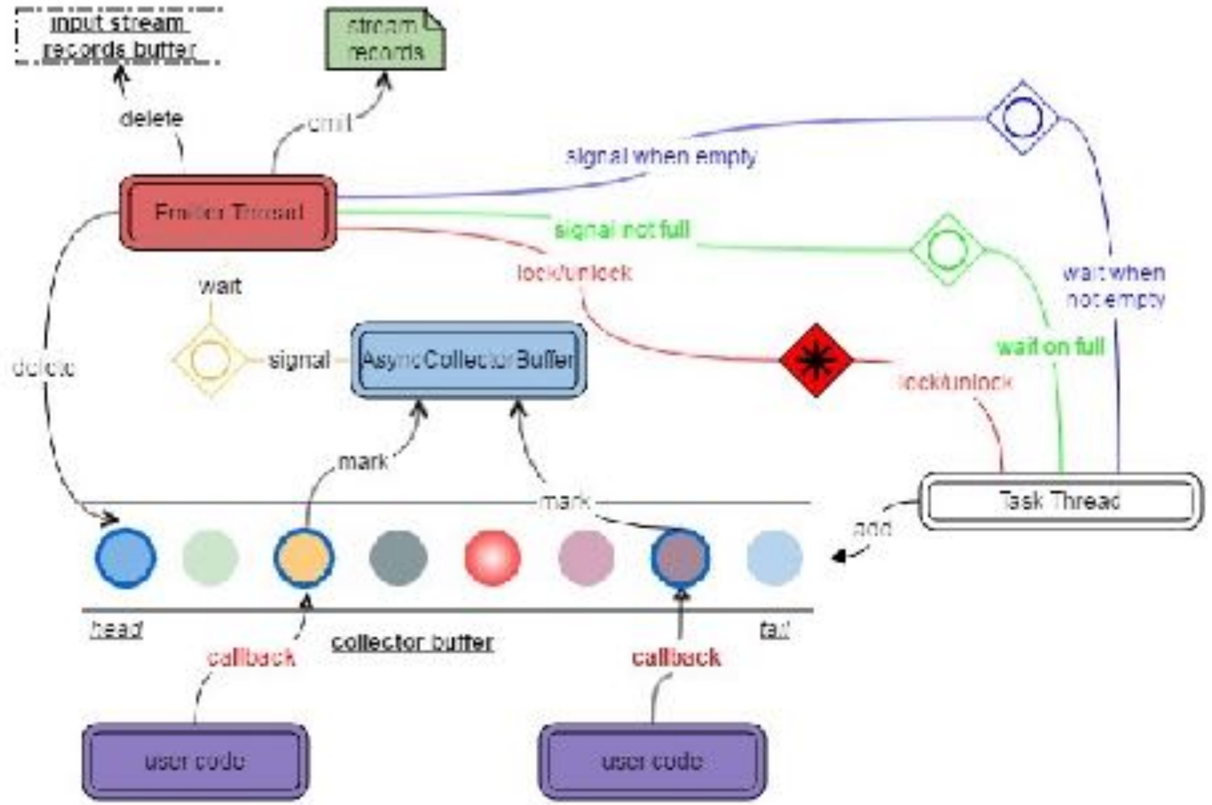
Little's Law:

throughput = occupancy / latency

Asynchronous I/O Support



Asynchronous I/O Support



Asynchronous I/O Support

```
// create the original stream
val stream: DataStream[String] = ...

// apply the async I/O transformation
val resultStream: DataStream[(String, String)] =

  AsyncDataStream.unorderedWait(
    input = stream,
    asyncFunction = new AsyncDatabaseRequest(),
    timeout = 1000,
    timeUnit = TimeUnit.MILLISECONDS,
    concurrentRequests = 100)
```

Asynchronous I/O Support

```
class AsyncDatabaseRequest extends AsyncFunction[String, (String, String)] {

    override def asyncInvoke(str: String, asyncCollector: AsyncCollector[(String, String)]): Unit = {

        // issue the asynchronous request, receive a future for the result
        val resultFuture: Future[String] = client.query(str)

        // set the callback to be executed once the request by the client is complete
        // the callback simply forwards the result to the collector
        resultFuture.onSuccess {
            case result: String => asyncCollector.collect(Iterable((str, result)));
        }
    }
}
```

Enhanced Security

- SSL
- Kerberos
    - Kafka
    - Zookeeper
    - Hadoop

Advanced Event-Driven Applications

- ProcessFunction API
- Queryable State API
- Excellent support for advanced applications that are:
    - Flexible
    - Stateful
    - Event Driven
    - Time Driven

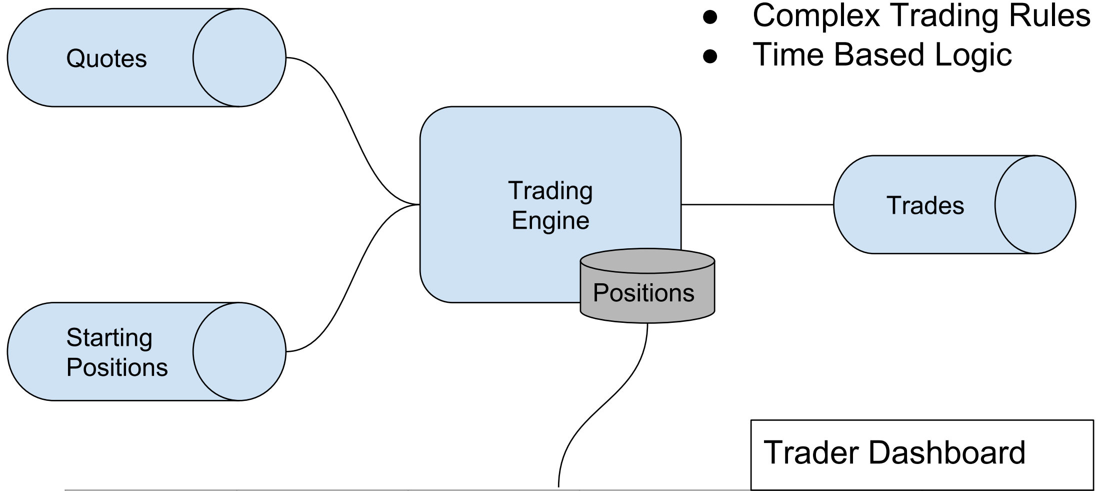
Example: FlinkTrade

- Overall Requirements:
    - Consume “starting position” and “quote” streams
    - Process complex, time-oriented, trading rules
    - Trade out of positions to our advantage if possible
    - Provide a dashboard of currently held positions to traders and asset managers

- Complex Rules:
    - We only make trades where the Bid Price is above our current Ask Price
    - When a trade is made we increase our Ask Price – looking to optimize our profits
    - Positions have a set time-to-live until we try to trade out of them more aggressively by decreasing the Ask Price over time

Example: FlinkTrade

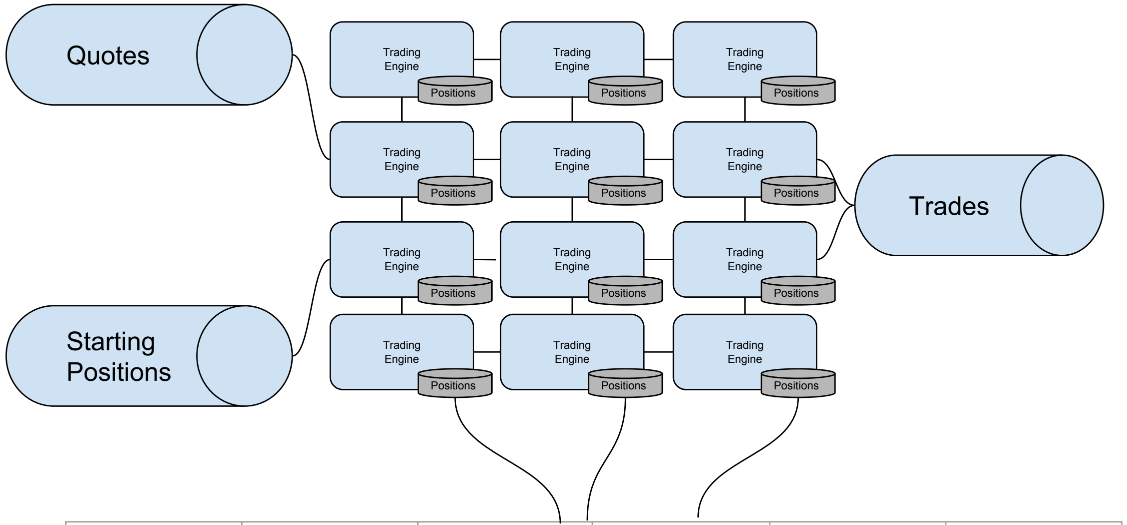
- Event Driven Processing
- Complex Trading Rules
- Time Based Logic



Trader Dashboard

SYMBOL	SHARES	BUY PRICE	ASK PRICE	LAST TRADE PRICE	Profit
AAPL	10,000	140.40	140.50	140.40	$10,921.00
GOOG	20,000	846.81	846.91	846.81	$12,021.00
TWTR	8,000	15.12	15.22	15.12	$4,032.00

Example: FlinkTrade



SYMBOL	SHARES	BUY PRICE	ASK PRICE	LAST TRADE PRICE	Profit
AAPL	10,000	140.40	140.50	140.40	$10,921.00
GOOG	20,000	846.81	846.91	846.81	$12,021.00
TWTR	8,000	15.12	15.22	15.12	$4,032.00

Example: FlinkTrade

Let’s look at the code

dataArtisans

We are hiring!

data-artisans.com/careers

@jamiegrier

@ApacheFlink

@dataArtisans