

The Salesforce logo, which consists of a blue cloud shape with the word "salesforce" written in white lowercase letters inside it.

salesforce

THE CUSTOMER SUCCESS PLATFORM

# Webhooks

---

Near-real time event processing with guaranteed delivery of HTTP callbacks

HBaseCon 2015

# Alan Steckley

---

Principal Software Engineer, Salesforce

The Salesforce logo, which consists of the word "salesforce" in a white, lowercase, sans-serif font, centered within a blue, multi-lobed cloud shape.

salesforce

# Poorna Chandra

---

Software Engineer, Cask



# Safe Harbor

Safe harbor statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.

# Overview

- Salesforce Marketing Cloud
- Webhooks use case
- Implementation in CDAP
- Q&A

# What is the Salesforce Marketing Cloud?

- Connects businesses to their customers through email, social media, and SMS.
- 1+ billion personalized messages per day
- 100,000's of business units
- Billions of subscribers
- Hosts petabytes of customer data in our data centers
- Handles a wide range of communications
  - Marketing campaigns
  - Purchase confirmations
  - Financial notifications
  - Password resets

# What is Webhooks?

- Webhooks is a near-real time event delivery platform with guaranteed delivery
  - Subscribers generate events by engaging with messages
  - Deliver events to customers over HTTP within seconds
  - Customers react to events in near real time

# Example use case

A purchase receipt email fails to be delivered

A mail bounce event is pushed to a service hosted by the retailer

Retailer's customer service is immediately aware of the failure



# General problem statement

1. Process a stream of near real time events based on customer defined actions.
2. Guarantee delivery of processed events emitted to third party systems.

# Primary concerns

High data integrity

Commerce, health, and finance messaging subject to government regulation

Horizontal scalability

Short time to market

Accessible developer experience

Existing Hadoop/YARN/HBase expertise and infrastructure

Open Source

# Implementation concern - Joins

Some events need pieces of information from other event streams

Example: An email click needs the email send event for contextual information

Wait until other events arrive to assemble the final event

Join across streams

Configurable TTL to wait to join (optional)

# Implementation concern - Delivery guarantees

Configurable per customer endpoint

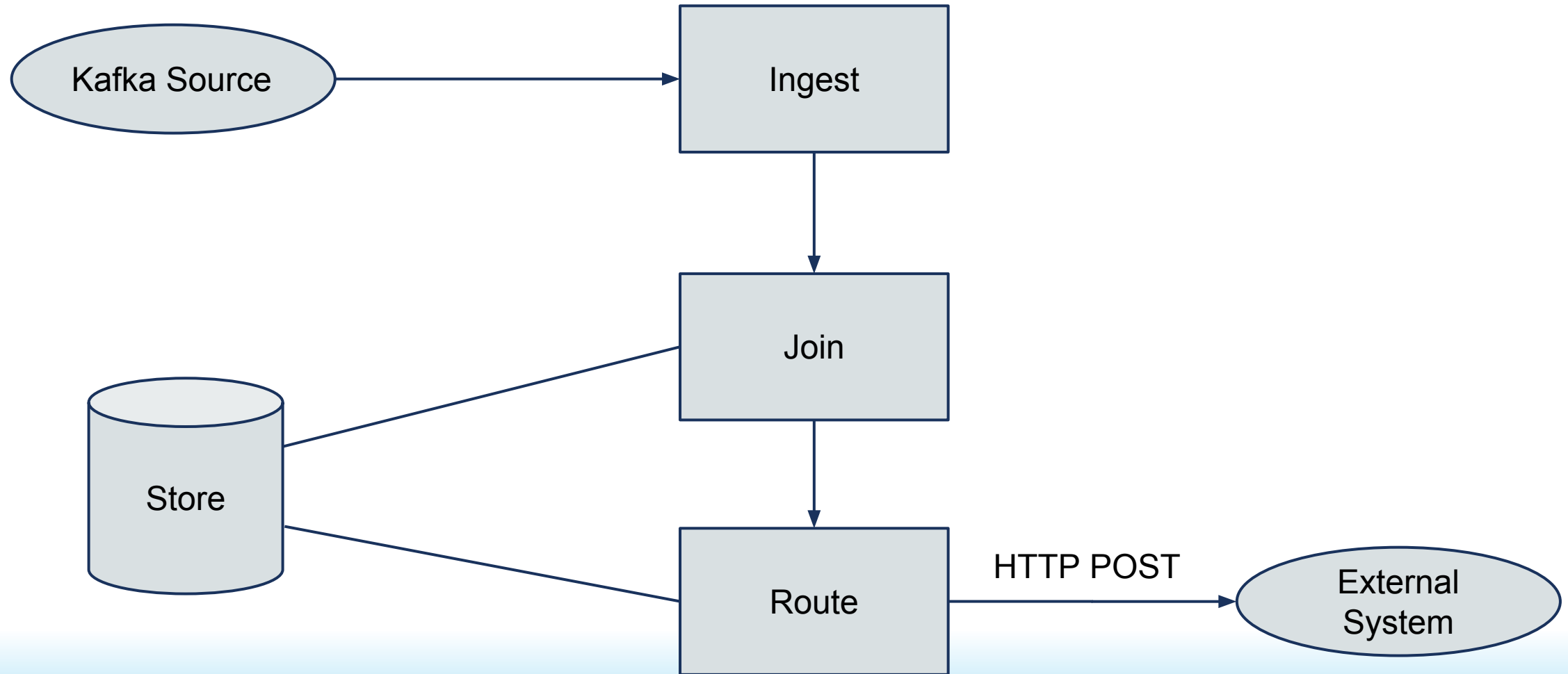
- Retry

- Throttle

- TTL to deliver (optional)

Reporting metrics, SLA compliance

# High level architecture



# Business logic

```
public class EventRouter {
    private Map<EventType, Route> routesMap;

    public void process(Event e) {
        Route route = routesMap.get(e.clientId());
        if (null != route) {
            httpPost(e, route);
        }
    }
}
```

# Business logic

```
public class EventJoiner {
    private Map<JoinKey, SendEvent> sends;

    public void process(ResponseEvent e) {
        SendEvent send = sends.get(e.getKey());
        if (null != send) {
            Event joined = join(send, e);
            routeEvent(joined);
        }
    }
}
```

# How to scale?

- Scaling data store is easy - use HBase
- Scaling application involves
  - Transactions
  - Application stack
  - Lifecycle management
  - Data movement
  - Coordination



 CDAP

# Cask Data Application Platform (CDAP)

- An open source framework to build and deploy data applications on Apache™ Hadoop®
- Provides abstractions to represent data access and processing pipelines
- Framework level guarantees for exactly-once semantics
- Transaction support on HBase
- Supports real time and batch processing
- Built on YARN and HBase

# Webhooks in CDAP

# Business logic

```
public class EventJoiner {
    private Map<JoinKey, SendEvent> sends;

    public void process(ResponseEvent e) {
        SendEvent send = sends.get(e.getKey());
        if (null != send) {
            Event joined = join(send, e);
            routeEvent(joined);
        }
    }
}
```

# Business logic in CDAP - Flowlet

```
public class EventJoiner extends AbstractFlowlet {
    @UseDataSet("sends")
    private SendEventDataset sends;
    private OutputEmitter<Event> outQueue;

    @ProcessInput
    public void join(ResponseEvent e) {
        SendEvent send = sends.get(e.getKey());
        if (send != null) {
            Event joined = join(e, send);
            outQueue.emit(joined);
        }
    }
}
```

# Access data with Datasets

```
public class EventJoiner extends AbstractFlowlet {
    @UseDataSet("sends")
    private SendEventDataset sends;
    private OutputEmitter<Event> outQueue;

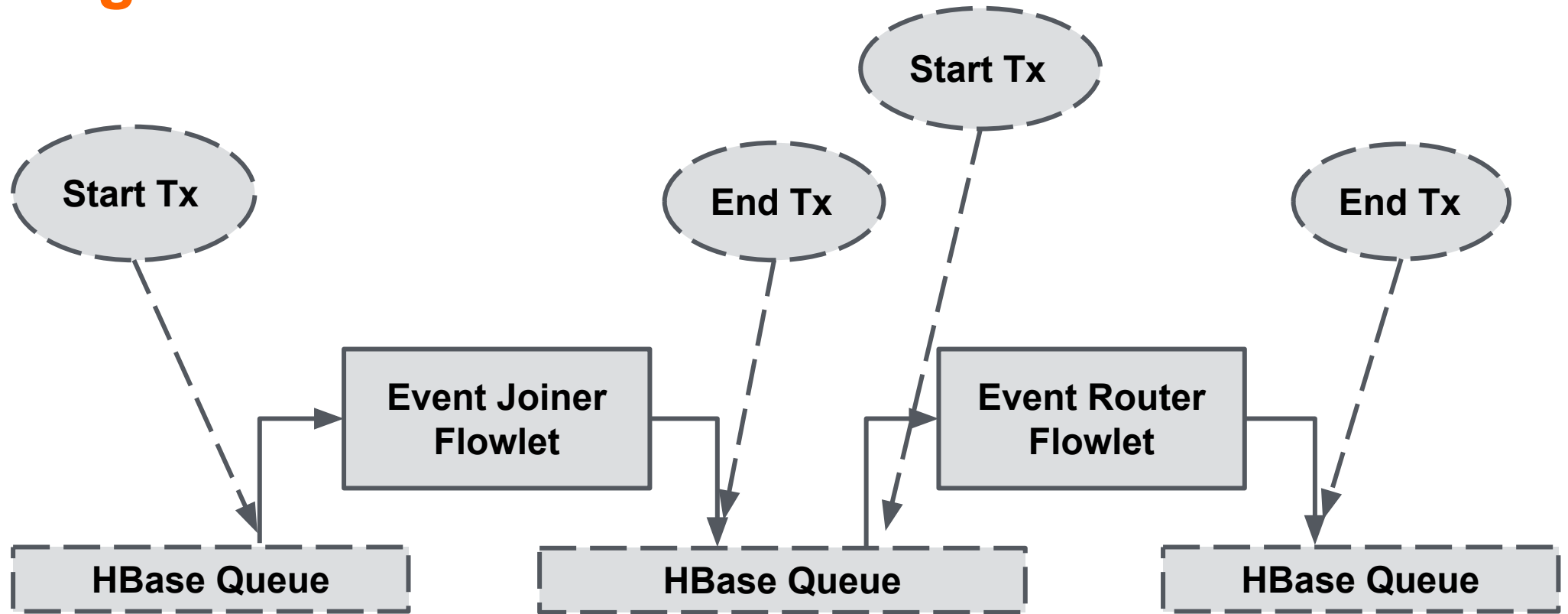
    @ProcessInput
    public void join(ResponseEvent e) {
        SendEvent send = sends.get(e.getKey());
        if (send != null) {
            Event joined = join(e, send);
            outQueue.emit(joined);
        }
    }
}
```

# Chain Flowlets with Queues

```
public class EventJoiner extends AbstractFlowlet {
    @UseDataSet("sends")
    private SendEventDataset sends;
    private OutputEmitter<Event> outQueue;

    @ProcessInput
    public void join(ResponseEvent e) {
        SendEvent send = sends.get(e.getKey());
        if (send != null) {
            Event joined = join(e, send);
            outQueue.emit(joined);
        }
    }
}
```

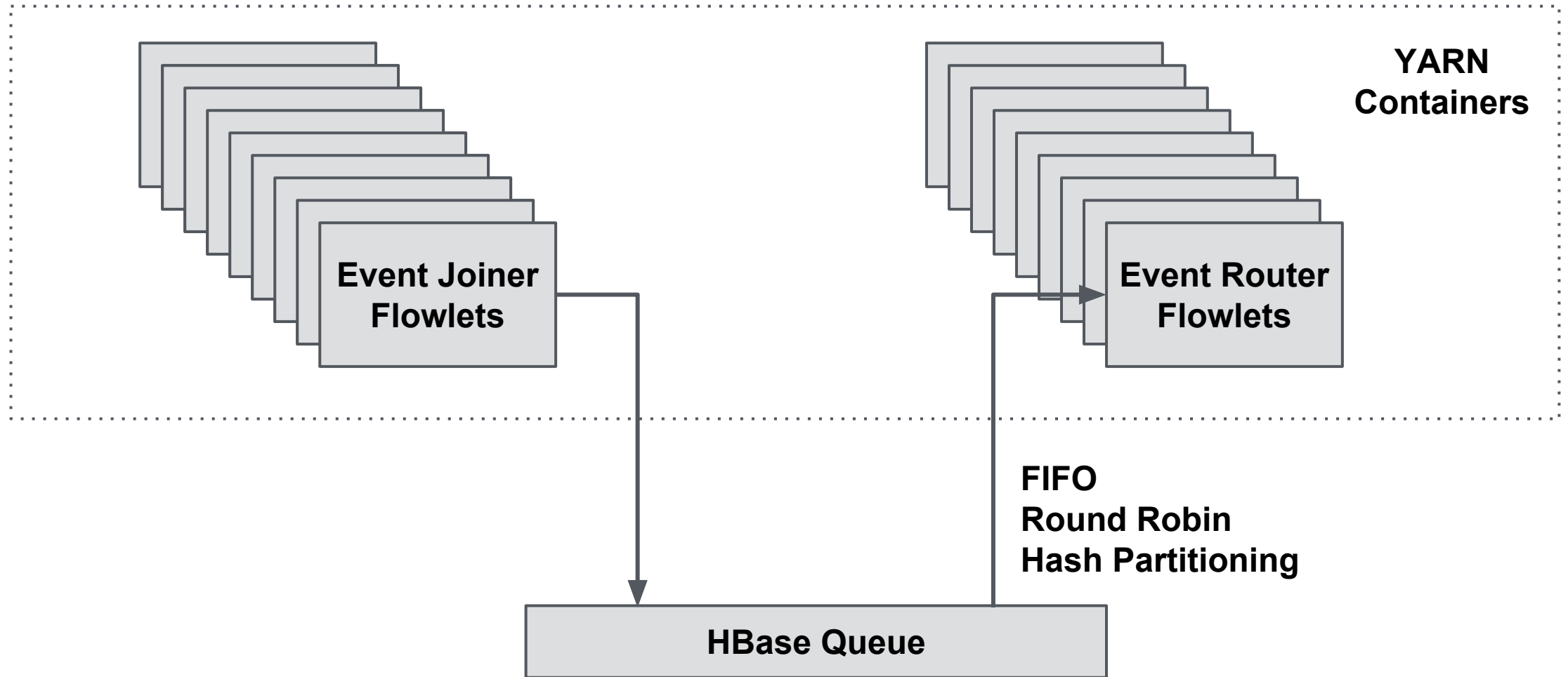
# Tigon Flow



- Real time streaming processor
- Composed of Flowlets
- Exactly-once semantics



# Scaling Flowlets



# Summary

- CDAP makes development easier by handling the overhead of scalability
  - Transactions
  - Application stack
  - Lifecycle management
  - Data movement
  - Coordination

# Datasets and Tephra

# Data abstraction using Dataset

- Store and retrieve data
- Reusable data access patterns
- Abstraction of underlying data storage
  - HBase
  - LevelDB
  - In-memory
- Can be shared between Flows (real-time) and MapReduce (batch)

# Transaction support with Tephra

- Transactions make exactly-once semantics possible
- Multi-row and across HBase regions transactions
- Optimistic concurrency control (Omid style)
- Open source (Apache 2.0 License)
- <http://tephra.io>



**Tephra**<sup>™</sup>

# Use and contribute

- Used today in enterprise cloud applications
- CDAP is open source (Apache 2.0 License)



<http://cdap.io/>

# Q&A

**Alan Steckley**

[asteckley@salesforce.com](mailto:asteckley@salesforce.com)

<http://salesforce.com>

**Poorna Chandra**

[poorna@cask.co](mailto:poorna@cask.co)

<http://cdap.io>