Design considerations to operate a stateful streaming pipeline as a service
TOC

Context
Example Problem
Design Principles
Example Solution
Operating Principles
Context

Data pipelines are services

Streaming is unforgiving

Observability is critical

User Expectations

Data Correctness

Data Quality

Performance

Latency

Alert fatigue

SRE principles
Example Problem

How taxi ride events arrive to PubSub

How we will group them to be able to calculate business properties e.g. calculating sessions for income taxi ride events based on event attribute
Design Principles

Extract

- Reads (input only)

Transform

- Enriching / hydrating → ✅
  - External state → ❌

Load

- Output only
Design Principles: but sinks have side effects?

Yes, and that's fine, but the details matter

How to write an IO for Beam

Jun-13 11:00-11:50 EDT
Room: Palisades

Writing an IO in Beam is hard. Distributed data reading and writing are inherently challenging, and its easy to make mistakes. This session is a walk through on the key design hurdles, and how to use Beam features to write a high quality IO.

https://beamsummit.org/sessions/2023/how-to-write-an-io-for-beam/
**Example Solution**

1. **Taxi Ride events**
   - When a specific event type is inspected, all events are read from the state and a session is calculated; otherwise set the timer to keep track of inactive rides.

2. **When a specific event type is inspected**, all events are read from the state and a session is calculated; otherwise set the timer to keep track of inactive rides.

3. **Timer to calculate session and clear state for inactive rides**

4. **Taxi Session events**
Example Solution

And cannot we just use windows?
  ●  Windows are also possible
  ●  If logic is not based on temporal properties, state & timers might be a better fit

In any case, for SRE principles, produce custom metrics
  ●  Distribution metrics are easier to use for SLOs
    ○  Stable metric that should not deviate too much from a "good" value
Beam Summit 2023

Example Solution

github.com/BhupiSindhwani/beam-stateful-processing
<table>
<thead>
<tr>
<th>ride_id</th>
<th>point_idx</th>
<th>latitude</th>
<th>longitude</th>
<th>timestamp</th>
<th>meter_reading</th>
<th>meter_increment</th>
<th>ride_status</th>
<th>passenger_count</th>
</tr>
</thead>
<tbody>
<tr>
<td>615d0ec-f7ae-48f4-be63-00df-1...</td>
<td>302</td>
<td>40.763</td>
<td>-73.979</td>
<td>2023-06-09 00:55:21.854890 U...</td>
<td>13.585863</td>
<td>0.0448963</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>77594d39-cdd0-4db4-ac3-51a-1...</td>
<td>235</td>
<td>40.760</td>
<td>-73.956</td>
<td>2023-06-09 00:55:46.710280 U...</td>
<td>6.413646</td>
<td>0.027292112</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>69c906dc-35a2-47dd-b28f-93c-5...</td>
<td>47</td>
<td>40.762</td>
<td>-73.999</td>
<td>2023-06-09 00:55:00.608380 U...</td>
<td>1.8488153</td>
<td>0.039336495</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1182459-8100-4826-9086-782-7...</td>
<td>365</td>
<td>40.753</td>
<td>-73.966</td>
<td>2023-06-09 00:55:11.731600 U...</td>
<td>12.111111</td>
<td>0.033181123</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>5ef3b3b-fc67-4308-91e-ead-7...</td>
<td>296</td>
<td>40.757</td>
<td>-73.996</td>
<td>2023-06-09 00:55:45.747990 U...</td>
<td>14.89705</td>
<td>0.05032787</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>c3c23fa-3c4d-4ea9-bb6-58d-1...</td>
<td>47</td>
<td>40.768</td>
<td>-73.952</td>
<td>2023-06-09 00:55:24.280720 U...</td>
<td>1.7284635</td>
<td>0.03677582</td>
<td>enroute</td>
<td>5</td>
</tr>
<tr>
<td>954877-f6a6-40e-9305-8c0-0...</td>
<td>153</td>
<td>40.739</td>
<td>-74.001</td>
<td>2023-06-09 00:56:03.446680 U...</td>
<td>6.667035</td>
<td>0.04357542</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>1ad40796-97cf-4b8b-b74-ad5-5...</td>
<td>41</td>
<td>40.759</td>
<td>-73.965</td>
<td>2023-06-09 00:54:50.456690 U...</td>
<td>1.3948454</td>
<td>0.03402618</td>
<td>enroute</td>
<td>6</td>
</tr>
<tr>
<td>294336f-5d10-4c6d-8a4b-ae2-1...</td>
<td>387</td>
<td>40.756</td>
<td>-73.974</td>
<td>2023-06-09 00:55:35.273970 U...</td>
<td>12.544704</td>
<td>0.032415256</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>c46b2a7-555e-4acc-b7-417-4...</td>
<td>28</td>
<td>40.744</td>
<td>-73.995</td>
<td>2023-06-09 00:55:49.093920 U...</td>
<td>1.0347826</td>
<td>0.036956523</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>91b2bd0-dc2b-4ff1-b16-096-8...</td>
<td>208</td>
<td>40.765</td>
<td>-73.964</td>
<td>2023-06-09 00:55:13.035110 U...</td>
<td>10.207284</td>
<td>0.04907348</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>562f42c7-1960-4ac8-86d-c97-9...</td>
<td>928</td>
<td>40.724</td>
<td>-73.993</td>
<td>2023-06-09 00:54:56.496310 U...</td>
<td>28.40294</td>
<td>0.030605616</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>1e53e565-6eb-8235-8647-66-8...</td>
<td>2304</td>
<td>40.741</td>
<td>-73.947</td>
<td>2023-06-09 00:56:16.300200 U...</td>
<td>53.22046</td>
<td>0.023099158</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>7f129f4-4c2b-403-4-b4a-202-4...</td>
<td>444</td>
<td>40.795</td>
<td>-73.976</td>
<td>2023-06-09 00:55:45.510820 U...</td>
<td>13.015213</td>
<td>0.029313544</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>0711967-29a1-41e-829e-a08-1...</td>
<td>18</td>
<td>40.766</td>
<td>-73.956</td>
<td>2023-06-09 00:55:10.906360 U...</td>
<td>0.838835</td>
<td>0.046601944</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>df23780-3e00-4ae8-aed-6f2-6...</td>
<td>153</td>
<td>40.755</td>
<td>-73.979</td>
<td>2023-06-09 00:55:52.182470 U...</td>
<td>5.881214</td>
<td>0.038493907</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>b9bcd1a-337e-4035-95b-5e-5ee...</td>
<td>179</td>
<td>40.764</td>
<td>-73.973</td>
<td>2023-06-09 00:55:39.544730 U...</td>
<td>6.237879</td>
<td>0.03484885</td>
<td>enroute</td>
<td>2</td>
</tr>
<tr>
<td>cc3caf-5d4d-45a-b6-cc-2...</td>
<td>101</td>
<td>40.781</td>
<td>-73.971</td>
<td>2023-06-09 00:54:54.731600 U...</td>
<td>17.0</td>
<td>0.16831638</td>
<td>droppend</td>
<td>2</td>
</tr>
<tr>
<td>69657b3-7a28-46e-616-c7-7...</td>
<td>139</td>
<td>40.7655</td>
<td>-73.997</td>
<td>2023-06-09 00:55:57.385810 U...</td>
<td>3.637383</td>
<td>0.026168223</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>1ae202-39b5-42f0-b4-6019-1...</td>
<td>29</td>
<td>40.730</td>
<td>-73.986</td>
<td>2023-06-09 00:56:07.073180 U...</td>
<td>1.7945545</td>
<td>0.06188119</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>197b90c8-693-43ef-9c6d-3...</td>
<td>65</td>
<td>40.761</td>
<td>-73.986</td>
<td>2023-06-09 00:56:04.838550 U...</td>
<td>3.6671126</td>
<td>0.05417115</td>
<td>enroute</td>
<td>1</td>
</tr>
<tr>
<td>f3600c52-b7a-c5-0-452-a1e6-b0...</td>
<td>2</td>
<td>40.757</td>
<td>-73.963</td>
<td>2023-06-09 00:55:34.929400 U...</td>
<td>0.1022178</td>
<td>0.05109896</td>
<td>enroute</td>
<td>2</td>
</tr>
</tbody>
</table>
# Example Solution

<table>
<thead>
<tr>
<th>session_id</th>
<th>total_meter_reading</th>
<th>passenger_count</th>
<th>journey_length_seconds</th>
<th>number_of_ride_eve</th>
<th>session_reason</th>
<th>session_start_time</th>
<th>session_end_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>8408e3fd-9935-41e4-a2fa-0f38388...</td>
<td>63.629997</td>
<td>1</td>
<td>1964.0</td>
<td>3205</td>
<td>DROPOFF</td>
<td>2023-06-09 23:21:38.731600 U...</td>
<td>2023-06-09 23:54:22.731600 U...</td>
</tr>
<tr>
<td>1692e757-4833-4962-a2e8-02...</td>
<td>60.8</td>
<td>3</td>
<td>2221.0</td>
<td>2750</td>
<td>DROPOFF</td>
<td>2023-06-09 23:17:42.731600 U...</td>
<td>2023-06-09 23:54:43.731600 U...</td>
</tr>
<tr>
<td>d985a1ce-1b9d-4dcb-9a56-510f...</td>
<td>69.6</td>
<td>1</td>
<td>1935.0</td>
<td>2964</td>
<td>DROPOFF</td>
<td>2023-06-09 23:22:35.731600 U...</td>
<td>2023-06-09 23:54:50.731600 U...</td>
</tr>
<tr>
<td>ceabf1c0-2afe-4eb-807d-6fceb...</td>
<td>64.3</td>
<td>1</td>
<td>1792.0</td>
<td>3177</td>
<td>DROPOFF</td>
<td>2023-06-09 23:24:57.731600 U...</td>
<td>2023-06-09 23:54:49.731600 U...</td>
</tr>
<tr>
<td>a7e235f0-5ca4-46eb-87b3-75b...</td>
<td>37.63</td>
<td>1</td>
<td>1770.0</td>
<td>1360</td>
<td>DROPOFF</td>
<td>2023-06-09 23:25:26.731600 U...</td>
<td>2023-06-09 23:54:56.731600 U...</td>
</tr>
<tr>
<td>be841d7e-c923-4150-9540-47f9...</td>
<td>42.36</td>
<td>1</td>
<td>1742.0</td>
<td>1802</td>
<td>DROPOFF</td>
<td>2023-06-09 23:25:56.731600 U...</td>
<td>2023-06-09 23:55:58.731600 U...</td>
</tr>
<tr>
<td>6a9ae3b3-b317-4b8-9a6-11a...</td>
<td>26.3</td>
<td>3</td>
<td>1846.0</td>
<td>1035</td>
<td>DROPOFF</td>
<td>2023-06-09 23:25:12.731600 U...</td>
<td>2023-06-09 23:55:58.731600 U...</td>
</tr>
<tr>
<td>437b85fe-4bdf-49f-8ad-7177...</td>
<td>51.6</td>
<td>6</td>
<td>1774.0</td>
<td>1510</td>
<td>DROPOFF</td>
<td>2023-06-09 23:26:40.731600 U...</td>
<td>2023-06-09 23:56:14.731600 U...</td>
</tr>
<tr>
<td>f423e27f-41ce-45f4-84f-8425...</td>
<td>47.4</td>
<td>2</td>
<td>1838.0</td>
<td>1674</td>
<td>DROPOFF</td>
<td>2023-06-09 23:25:47.731600 U...</td>
<td>2023-06-09 23:56:25.731600 U...</td>
</tr>
<tr>
<td>a8b729b6e-813d-4db2-9dd-888...</td>
<td>14.0</td>
<td>1</td>
<td>1764.0</td>
<td>245</td>
<td>DROPOFF</td>
<td>2023-06-09 23:27:10.731600 U...</td>
<td>2023-06-09 23:56:34.731600 U...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>session_id</th>
<th>total_meter_reading</th>
<th>passenger_count</th>
<th>journey_length_seconds</th>
<th>number_of_ride_eve</th>
<th>session_reason</th>
<th>session_start_time</th>
<th>session_end_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1343fd3c2-2142-42-eb8e-8490d...</td>
<td>5.255085</td>
<td>1</td>
<td>159.63559</td>
<td>113</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 07:27:25.731600 U...</td>
<td>2023-06-09 07:30:05.367190 U...</td>
</tr>
<tr>
<td>c7e4e27a-65cb-4d9b-8627-c9e...</td>
<td>5.555596</td>
<td>1</td>
<td>71.424</td>
<td>114</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 08:38:36.731600 U...</td>
<td>2023-06-09 08:42:48.155600 U...</td>
</tr>
<tr>
<td>5ab567d2-0f93-4df4-b9c8-f4f2...</td>
<td>5.558823</td>
<td>5</td>
<td>96.28876</td>
<td>115</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 10:07:20.731600 U...</td>
<td>2023-06-09 10:08:57.018350 U...</td>
</tr>
<tr>
<td>a556e1a2-3673-4f3a-88e8-1e4...</td>
<td>5.7543206</td>
<td>1</td>
<td>132.94488</td>
<td>115</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 10:58:30.731600 U...</td>
<td>2023-06-09 11:00:43.676480 U...</td>
</tr>
<tr>
<td>4b06b103-24c4-466b-82c2-101...</td>
<td>5.7588654</td>
<td>2</td>
<td>151.91489</td>
<td>116</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 11:03:37.731600 U...</td>
<td>2023-06-09 11:06:09.646490 U...</td>
</tr>
<tr>
<td>05c64e2-9363-482c-97ca-f21...</td>
<td>6.905197</td>
<td>3</td>
<td>168.66142</td>
<td>116</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 11:03:26.731600 U...</td>
<td>2023-06-09 11:06:15.393020 U...</td>
</tr>
<tr>
<td>6071f8d-46ba-4085-8905-6b0...</td>
<td>5.2576003</td>
<td>3</td>
<td>178.15</td>
<td>117</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 07:01:23.731600 U...</td>
<td>2023-06-09 07:04:22.291600 U...</td>
</tr>
<tr>
<td>0946f5ce-2a05-4bdc-5b94-135...</td>
<td>5.757037</td>
<td>1</td>
<td>178.66667</td>
<td>117</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 07:11:44.731600 U...</td>
<td>2023-06-09 07:14:43.398270 U...</td>
</tr>
<tr>
<td>95a3eb7c-587c-4143-b01b-fc6...</td>
<td>5.7579713</td>
<td>1</td>
<td>160.82069</td>
<td>117</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 08:03:33.731600 U...</td>
<td>2023-06-09 08:06:14.557690 U...</td>
</tr>
<tr>
<td>b01c6356-cdc4-43ac-9e22-05c...</td>
<td>6.2533336</td>
<td>5</td>
<td>160.58518</td>
<td>118</td>
<td>GARBAGE_COLLECTION</td>
<td>2023-06-09 07:33:54.939010 U...</td>
<td>2023-06-09 07:36:35.524190 U...</td>
</tr>
</tbody>
</table>
Example Solution

def process(self, element):
    element: Tuple[str, TaxiRideEvent],
    element_timestamp: Timestamp = beam.DoFn.TimestampParam,
    taxi_ride_events_bag = beam.DoFn.StateParam(TAXI_RIDE_EVENTS_BAG),
    max_timestamp_seen = beam.DoFn.StateParam(MAX_TIMESTAMP),
    gc_timer = beam.DoFn.TimerParam(GC_TIMER)) -> Iterable[TaxiStatEvent]:

taxi_ride_events_bag.add(element)
max_timestamp_seen.add(element_timestamp.seconds())
GenerateSessionsDoFn.ride_events_received.inc()

if element[1].ride_status == "dropoff":
    taxi_stat_event = GenerateSessionsDoFn.calculate_session(taxi_ride_events_bag,
                                                            TaxiStatEvent.Reason.DROPOFF)

    # Generate session and output TaxiStatEvent
    GenerateSessionsDoFn.sessions_processed.inc()
    yield beam.window.TimedValue(taxi_stat_event, Timestamp(max_timestamp_seen.read()))

    # Clear state for the key
    taxi_ride_events_bag.clear()
    max_timestamp_seen.clear()
    gc_timer.clear()
else:
    # Set the timer to be 5 minutes to keep track of inactive keys
    expiration_time = Timestamp(max_timestamp_seen.read()) + Duration(seconds=5 * 60)
    gc_timer.set(expiration_time)

def expiry_callback(self, taxi_ride_events_bag=beam.DoFn.StateParam(TAXI_RIDE_EVENTS_BAG),
                    max_timestamp_seen=beam.DoFn.StateParam(MAX_TIMESTAMP)) -> Iterable[TaxiStatEvent]:

    # We have not seen the drop-off message 5 minutes after the max timestamp, so let's exit this session now
    taxi_stat_event = GenerateSessionsDoFn.calculate_session(taxi_ride_events_bag,
                                                            TaxiStatEvent.Reason.GARbage_COLLECTION)
Example Solution: adding business-related SLIs

class GenerateSessionsDoFn(beam.DoFn):
    def __init__(self):
        self._distribution = Metrics.distribution('My sessions DoFn', 'vehicle_speed')

    journey_length_seconds = (end_time - start_time).total_seconds()
    distance = last_event.meter_reading  # Just an example, this could be actual distance
    session_speed = distance / journey_length_seconds
    self._distribution.update(session_speed)
## Operating Principles

<table>
<thead>
<tr>
<th>SLIs</th>
<th>SLOs</th>
<th>Error Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicator of the level of</td>
<td>Target levels for the reliability</td>
<td>Allowed bad events; user tolerance</td>
</tr>
<tr>
<td>service</td>
<td>of service</td>
<td></td>
</tr>
</tbody>
</table>

- SLIs: Indicator of the level of service
- SLOs: Target levels for the reliability of service
- Error Budget: Allowed bad events; user tolerance
Operating Principles

- Processing latency (90s)
- BigQuery session write latency (10s)
- Data age (90s)
Operating Principles: SLI
Operating Principles: error budget

- **Service level indicator**: 100%
- **Error budget**: 100%
- **Alerts firing**: 0/0

**Error budget**
Error budget fraction represents the actual percentage of error budget remaining for the compliance period.
Operating Principles: error budget

<table>
<thead>
<tr>
<th>Service level indicator</th>
<th>Error budget</th>
<th>Alerts firing</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>1.25%</td>
<td>0/0</td>
</tr>
</tbody>
</table>

Error budget: Error budget fraction represents the actual percentage of error budget remaining for the compliance period.
Operating Principles: error budget

- BigQuery session write latency (10s)
  - Service level indicator: 100%
  - Error budget: 1.25%
  - Alerts firing: 0/0

Error budget: Error budget fraction represents the actual percentage of error budget remaining for the compliance period.

Situation slowly improving in the last hours.
Reminder

Data pipelines are services

Streaming is unforgiving

Observability is critical

User Expectations

Data Correctness
Data Quality
Performance
Latency

Alert fatigue
SRE principles

Photo by Sigmund on Unsplash

Photo by Quaritsch Photography on Unsplash

Photo by Chris Liverani on Unsplash
QUESTIONS?

Israel Herraiz
linkedin.com/in/herraiz
github.com/iht

Bhupinder Sindhwan
linkedin.com/in/bhupindersindhwan
github.com/BhupiSindhwan

Example implementation:
github.com/BhupiSindhwan/beam-stateful-processing