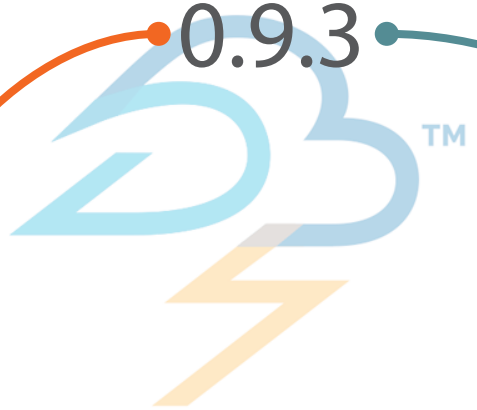
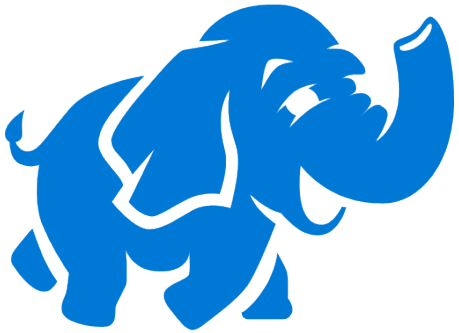
Using Storm for the Plumbing

Elton Stoneman

@EltonStoneman | www.geekswithblogs.net/eltonstoneman





2.6.0

0.9.3

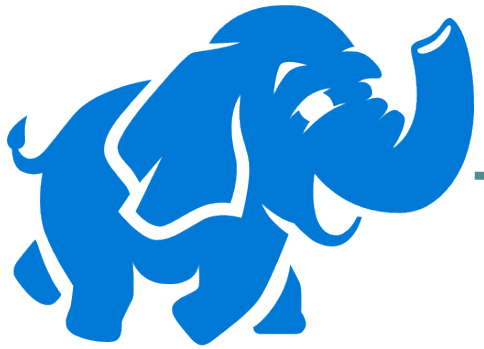
0.98.4

JULY 2014

NOV 2014

FEB 2015

APACHE HBASE



STREAM A INTO ...

"xyz 123 abc"

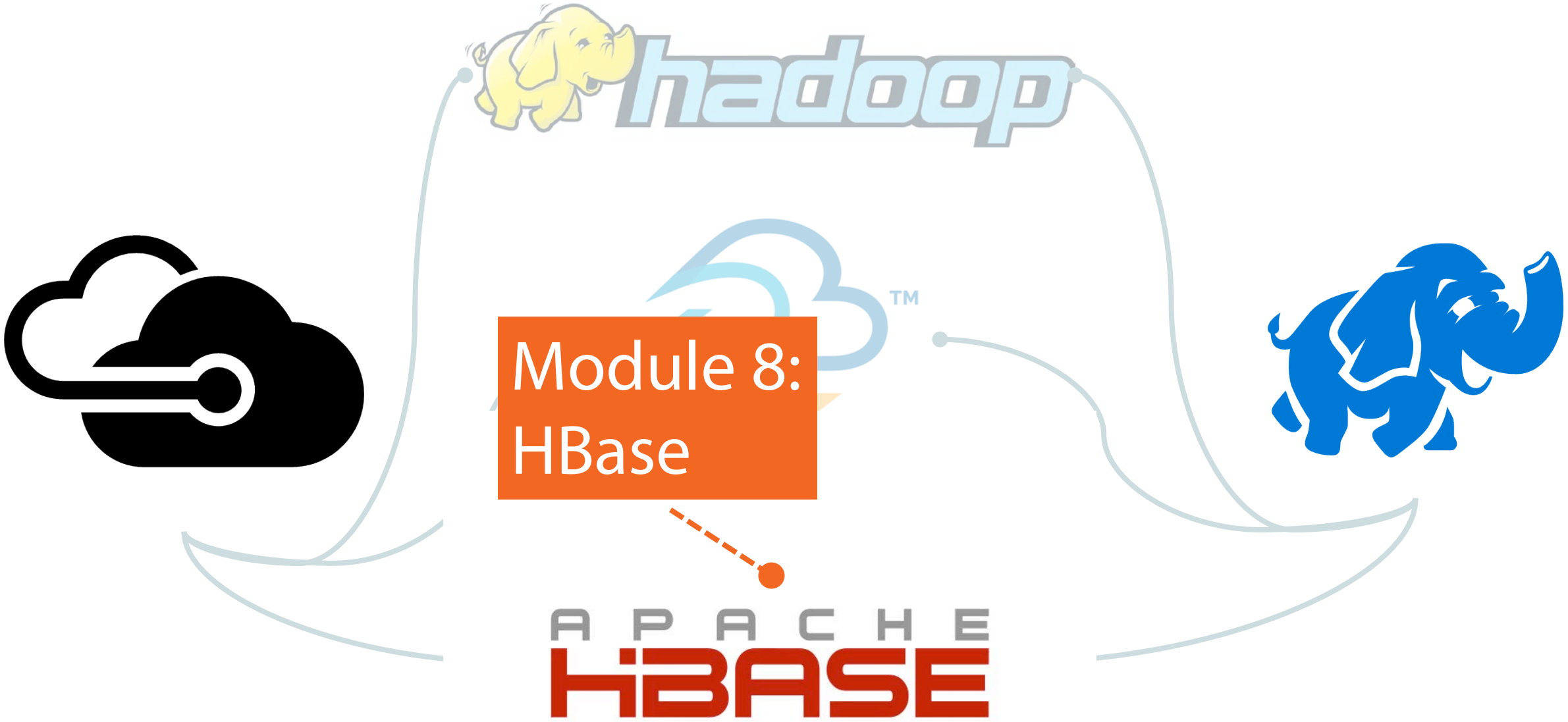
```
c:\>
```

Microsoft® .NET

```
register './lib/elephant-bird-pig-4.6.jar';
register './lib/My.DotNet.Assembly.dll';
```

Microsoft® .NET

Java™

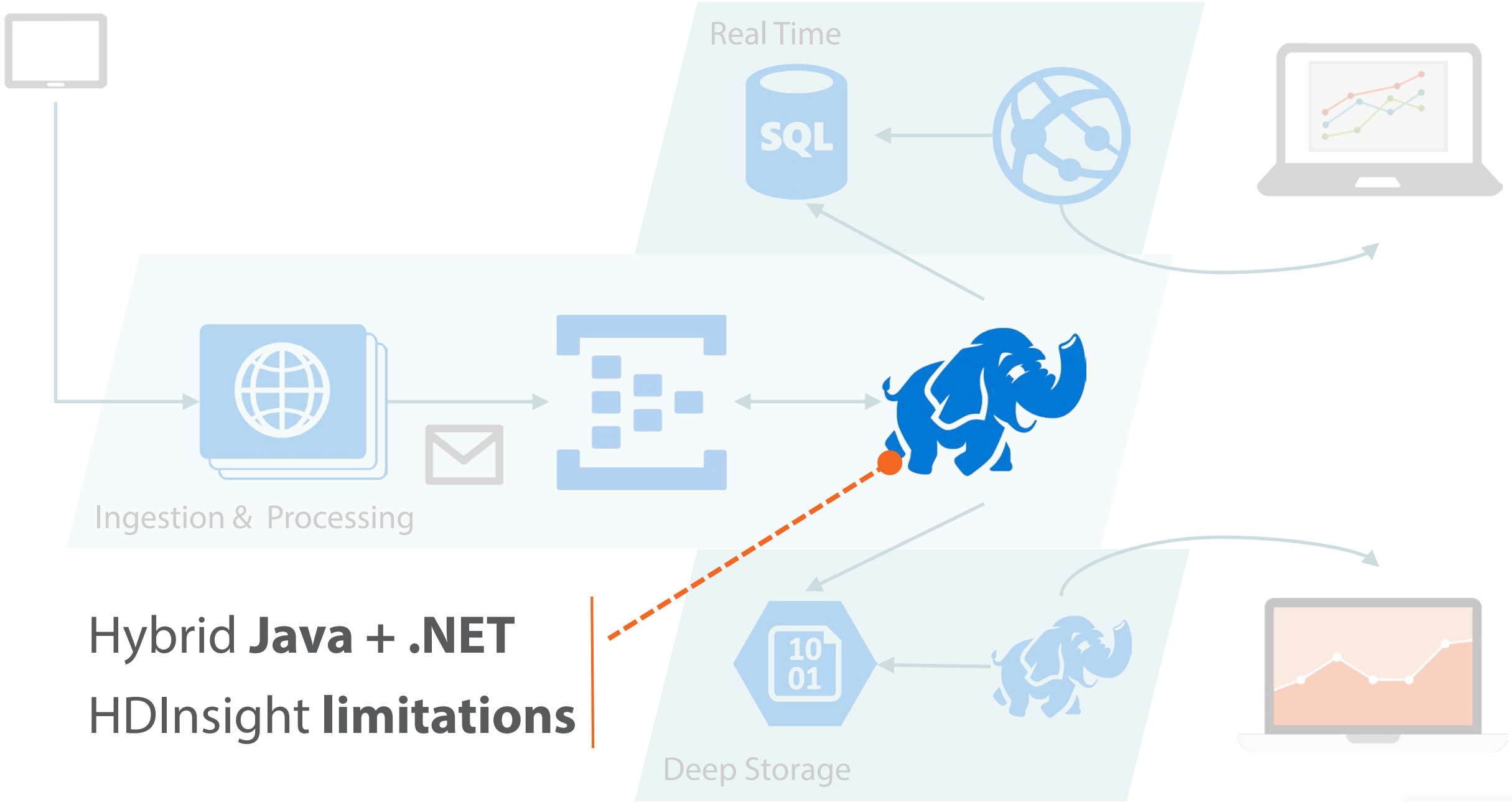

hadoop
Module 8:
HBase
APACHE
HBASE

Event Hub Connector

Event Processors

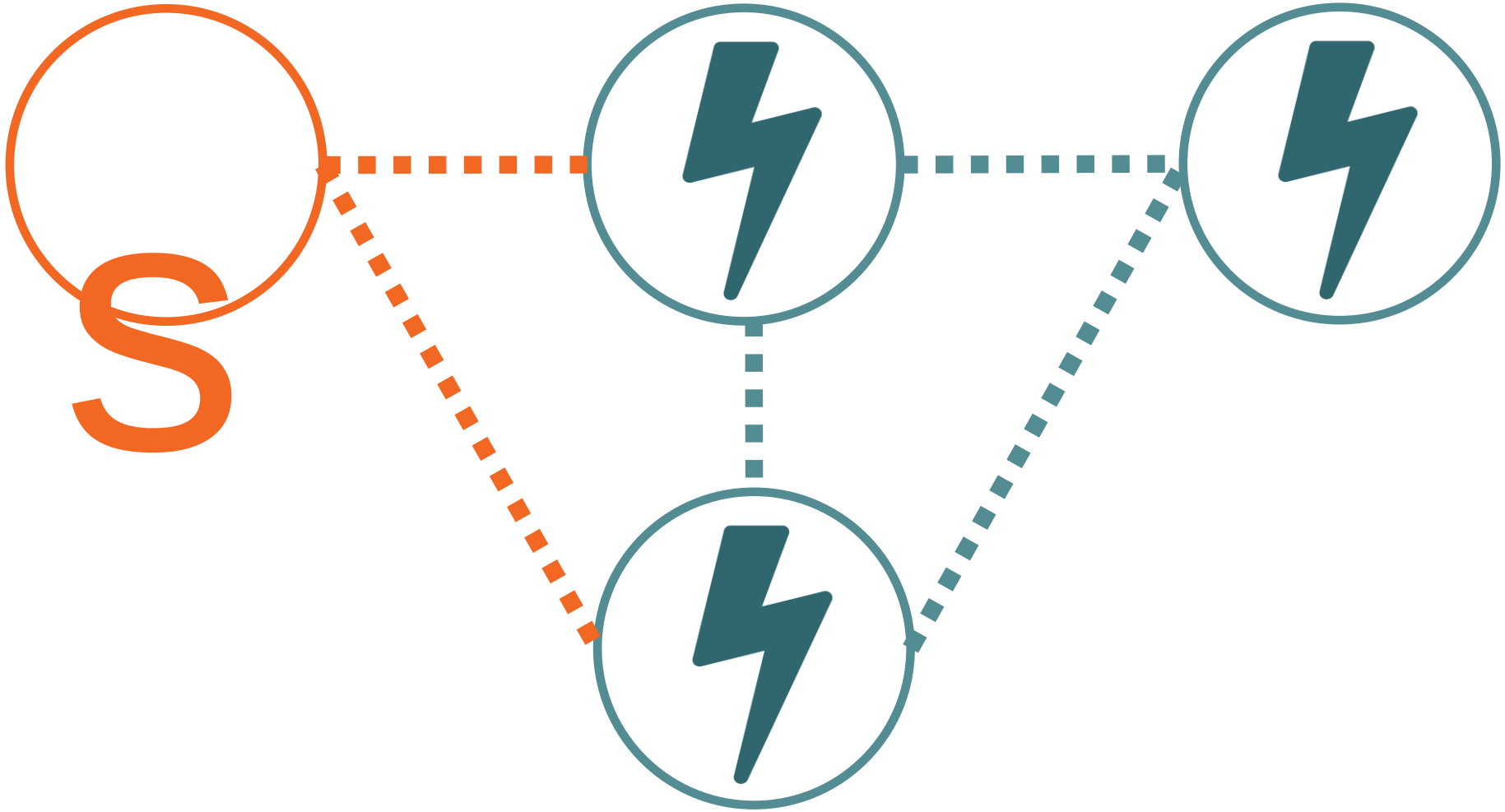
Java™

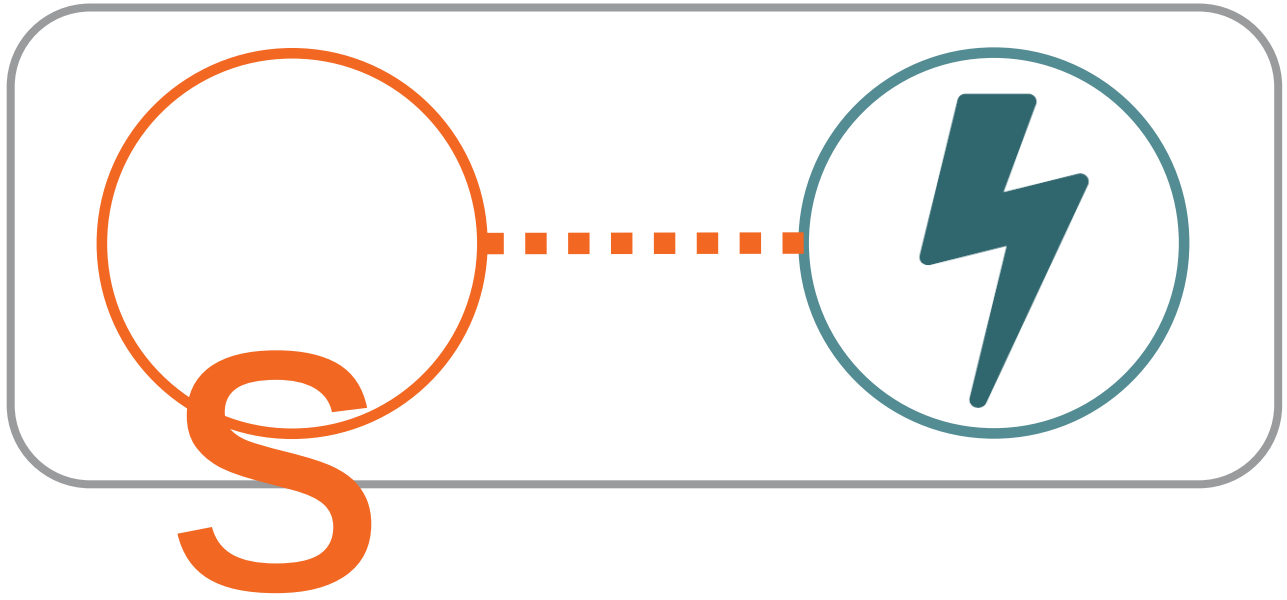
Microsoft®
.NET



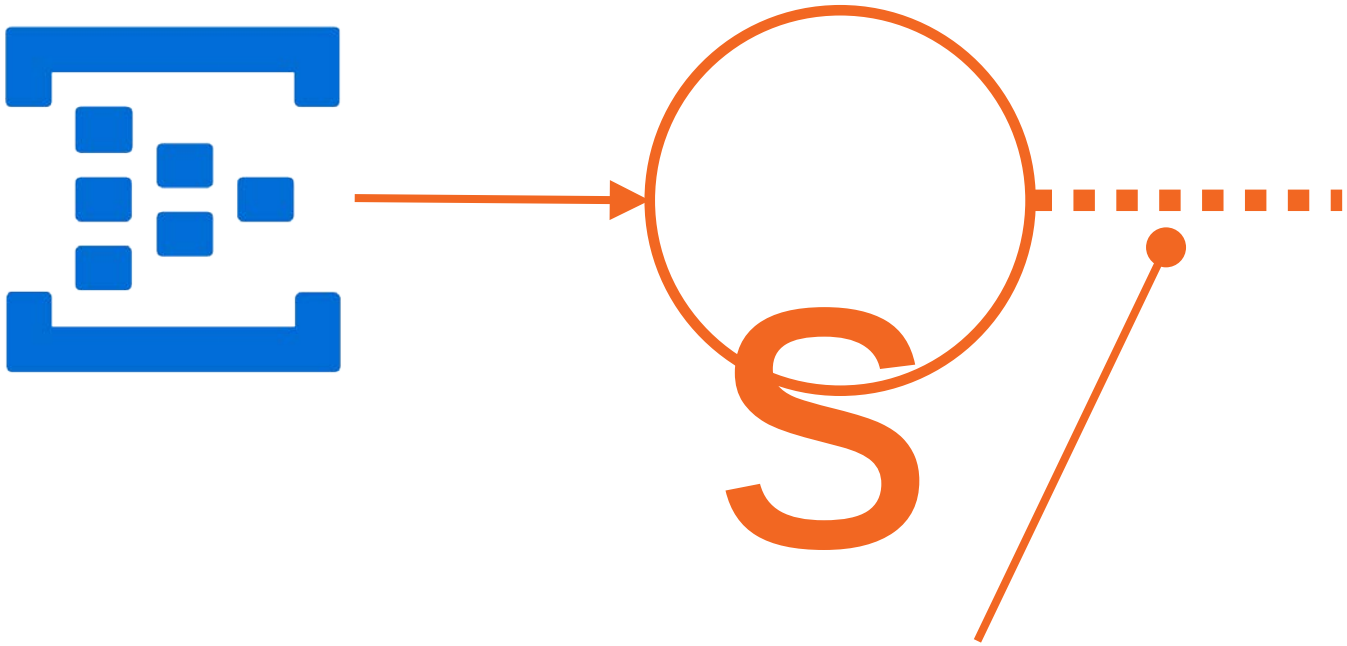
Hybrid **Java + .NET**

HDInsight **limitations**

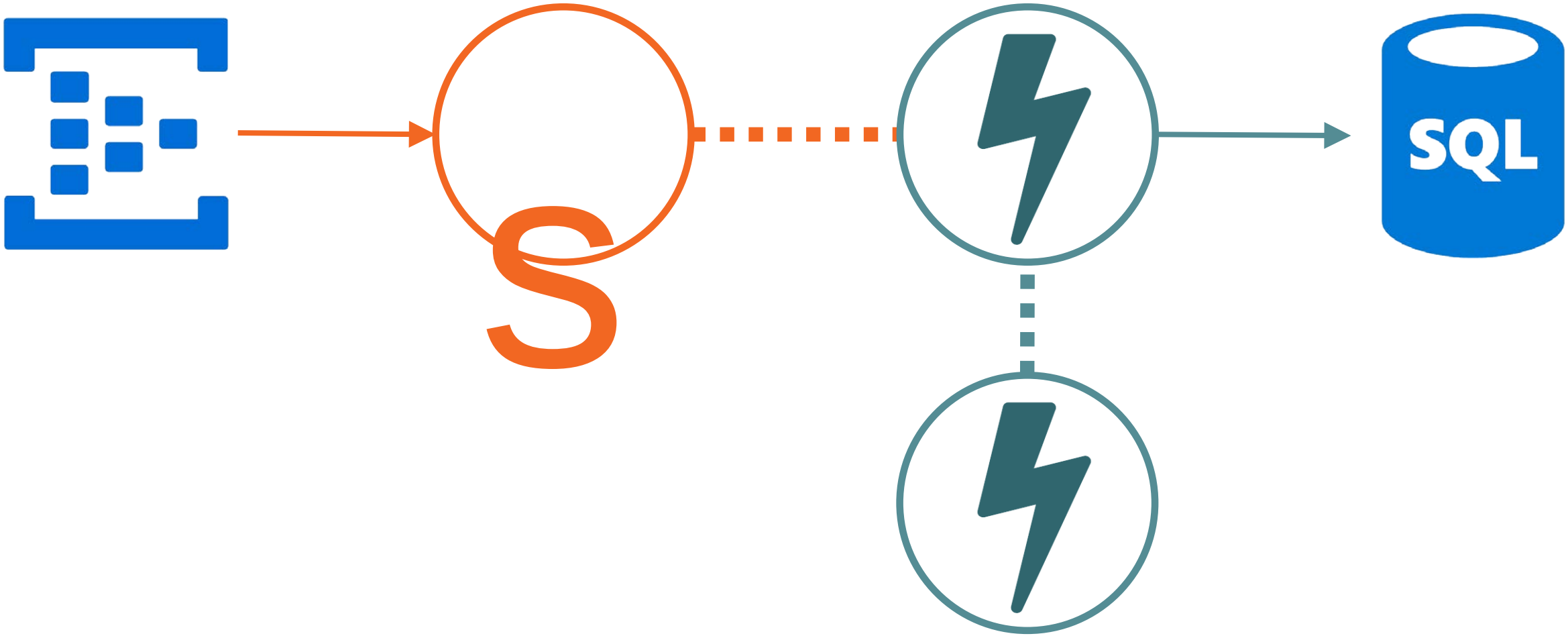




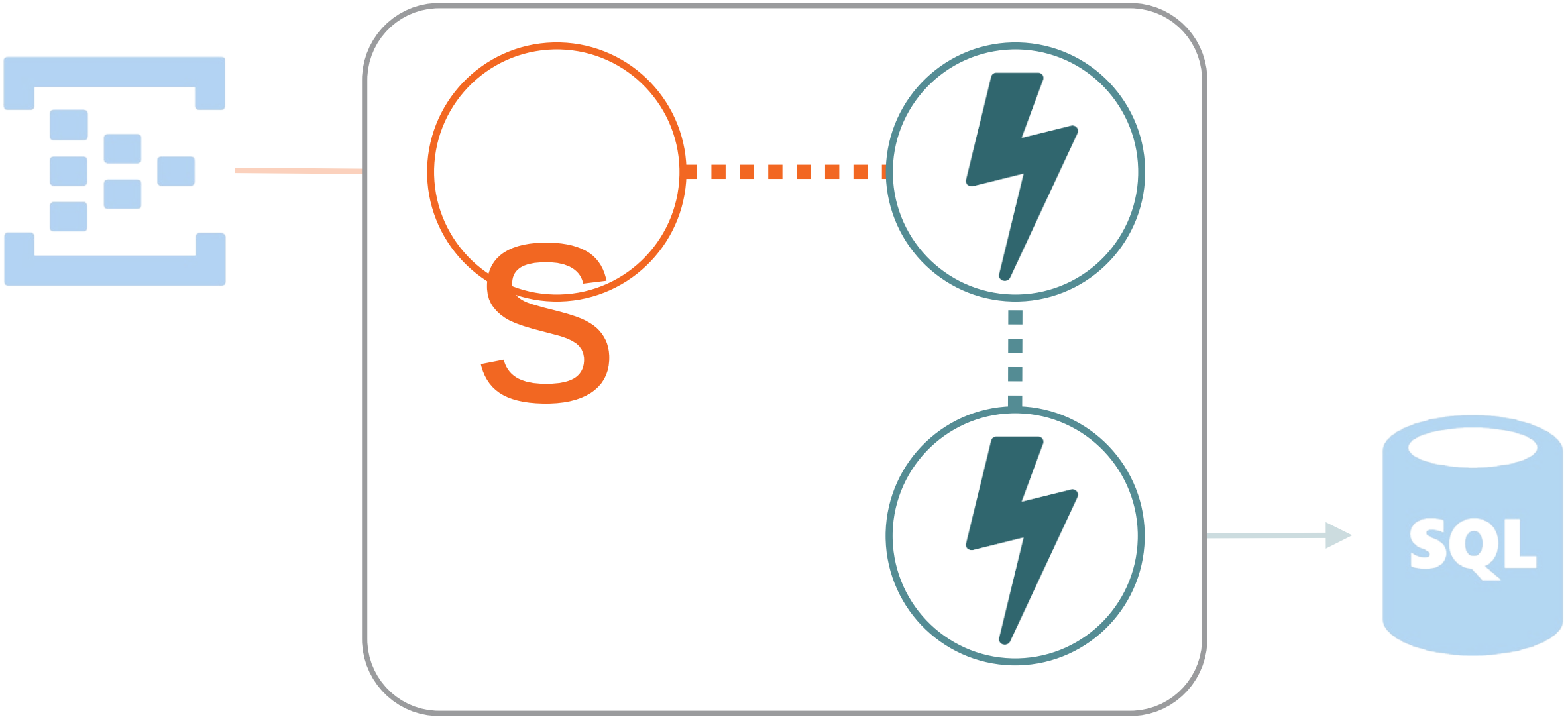
Tuple	
eventName	device.logs.message
period	2015060507
eventJson	{ "field" : "value" }

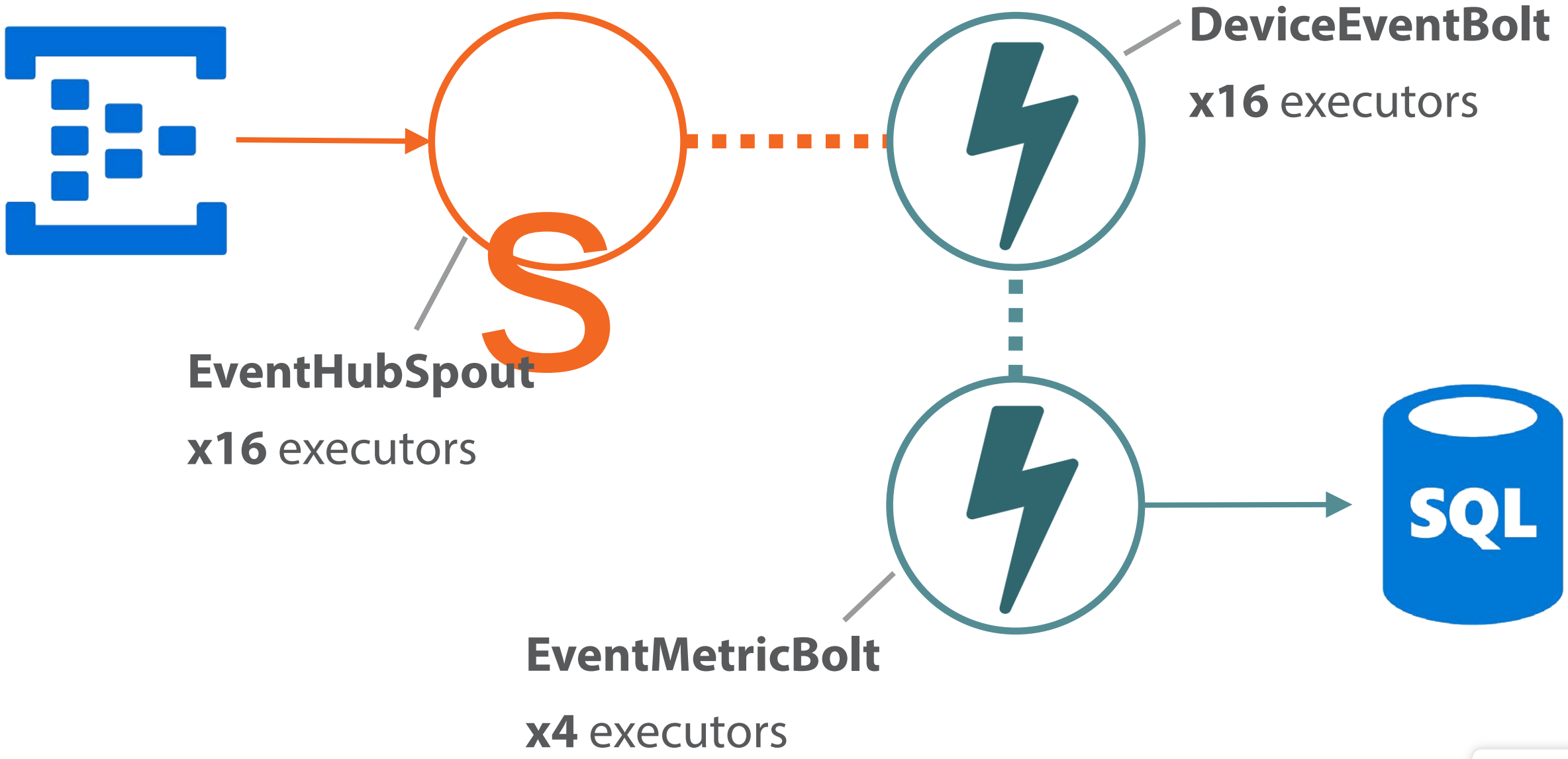


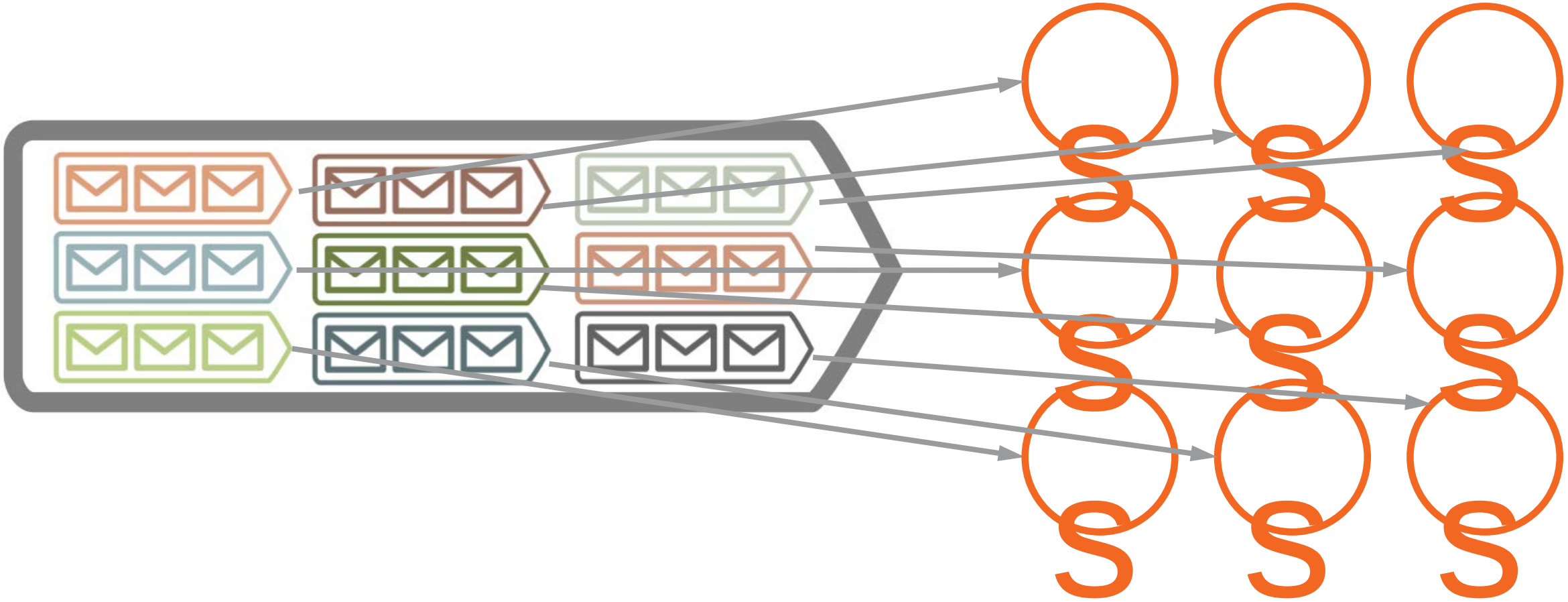
Tuple	
eventData	{ "field" : "value" }

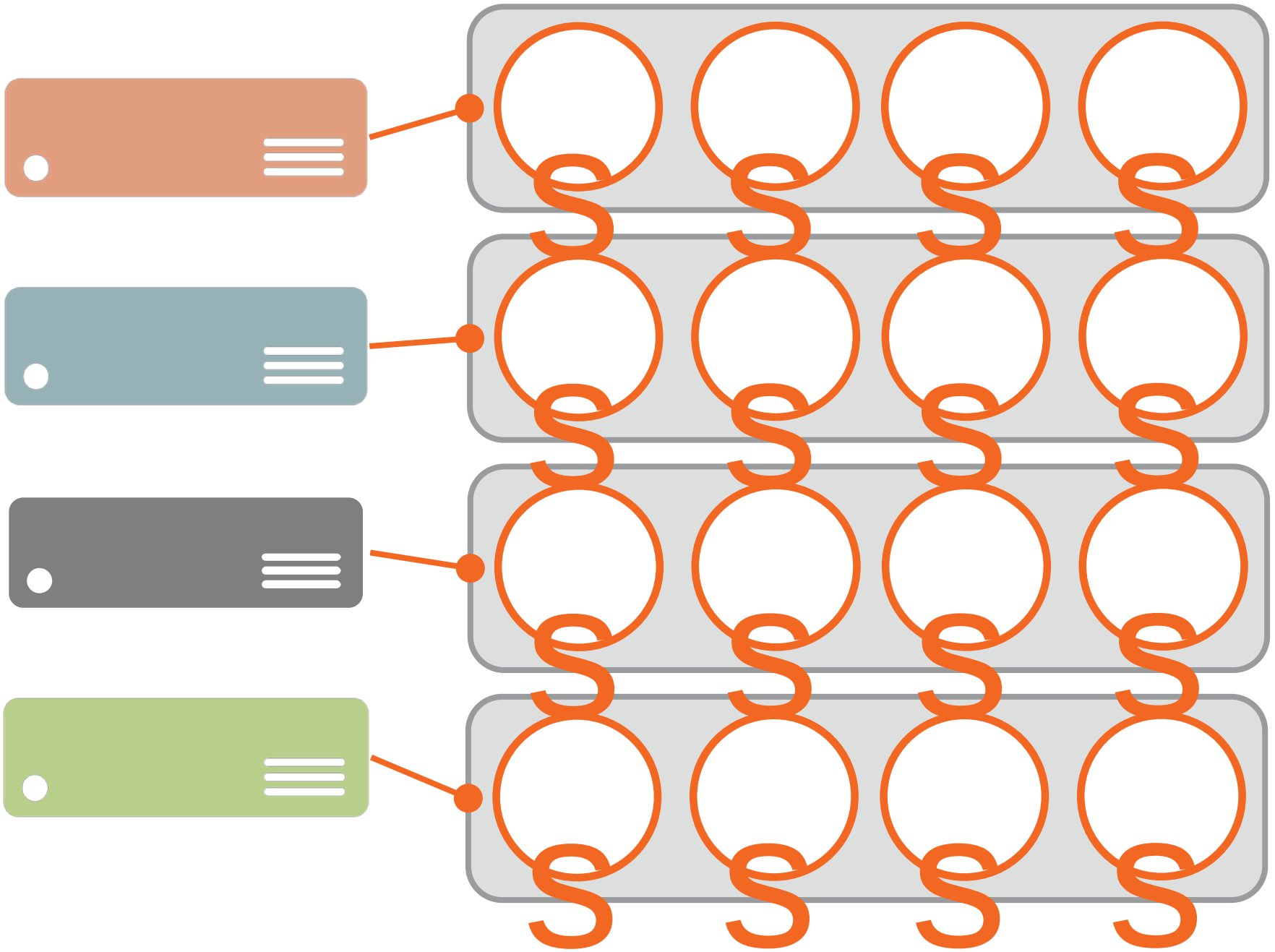


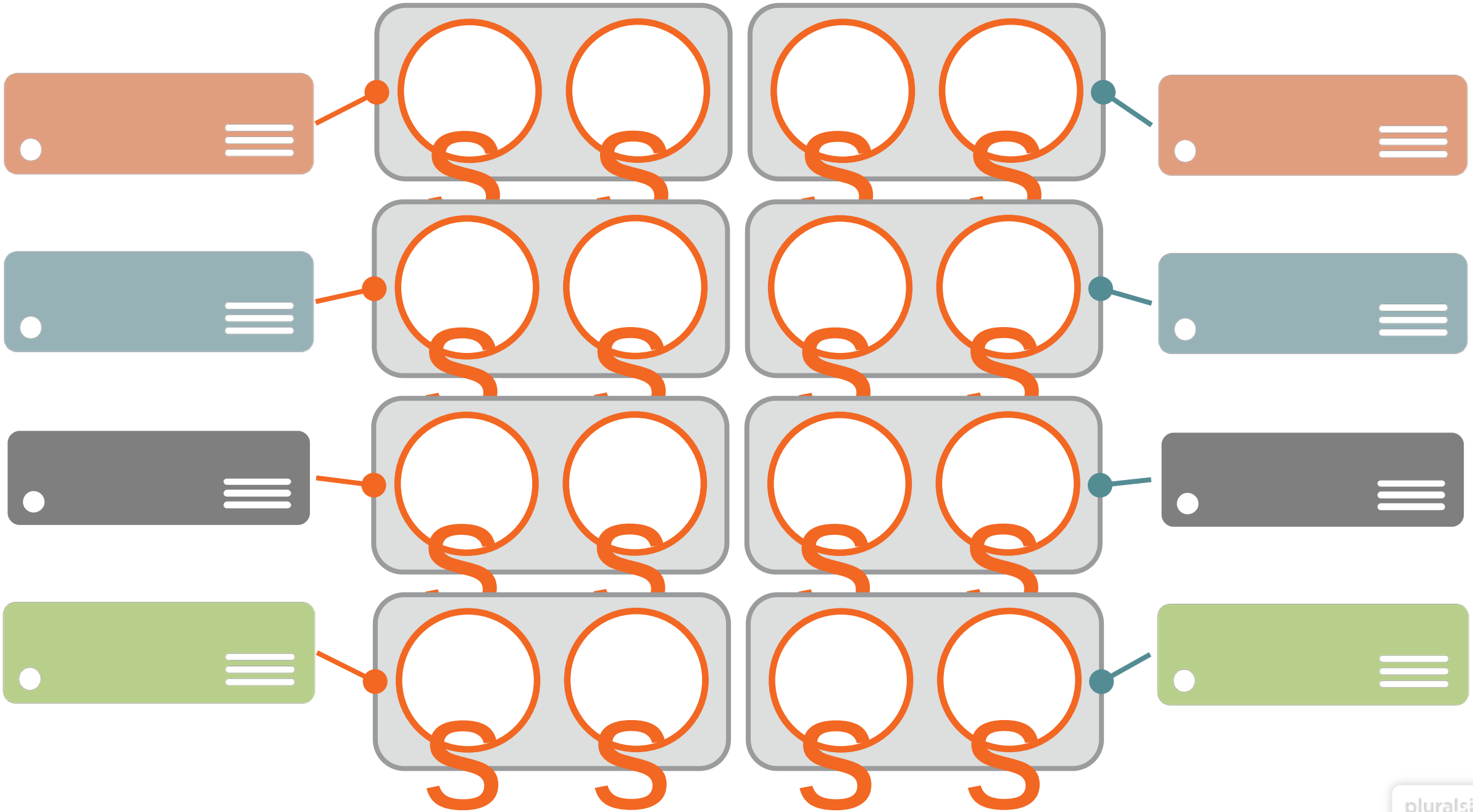

DeviceEventBolt
Emits event name
EventHubSpout
Emits event data
SQL
EventMetricBolt
Counts event name


S
SQL


DeviceEventBolt
x16 executors
EventHubSpout
x16 executors
EventMetricBolt
x4 executors
SQL

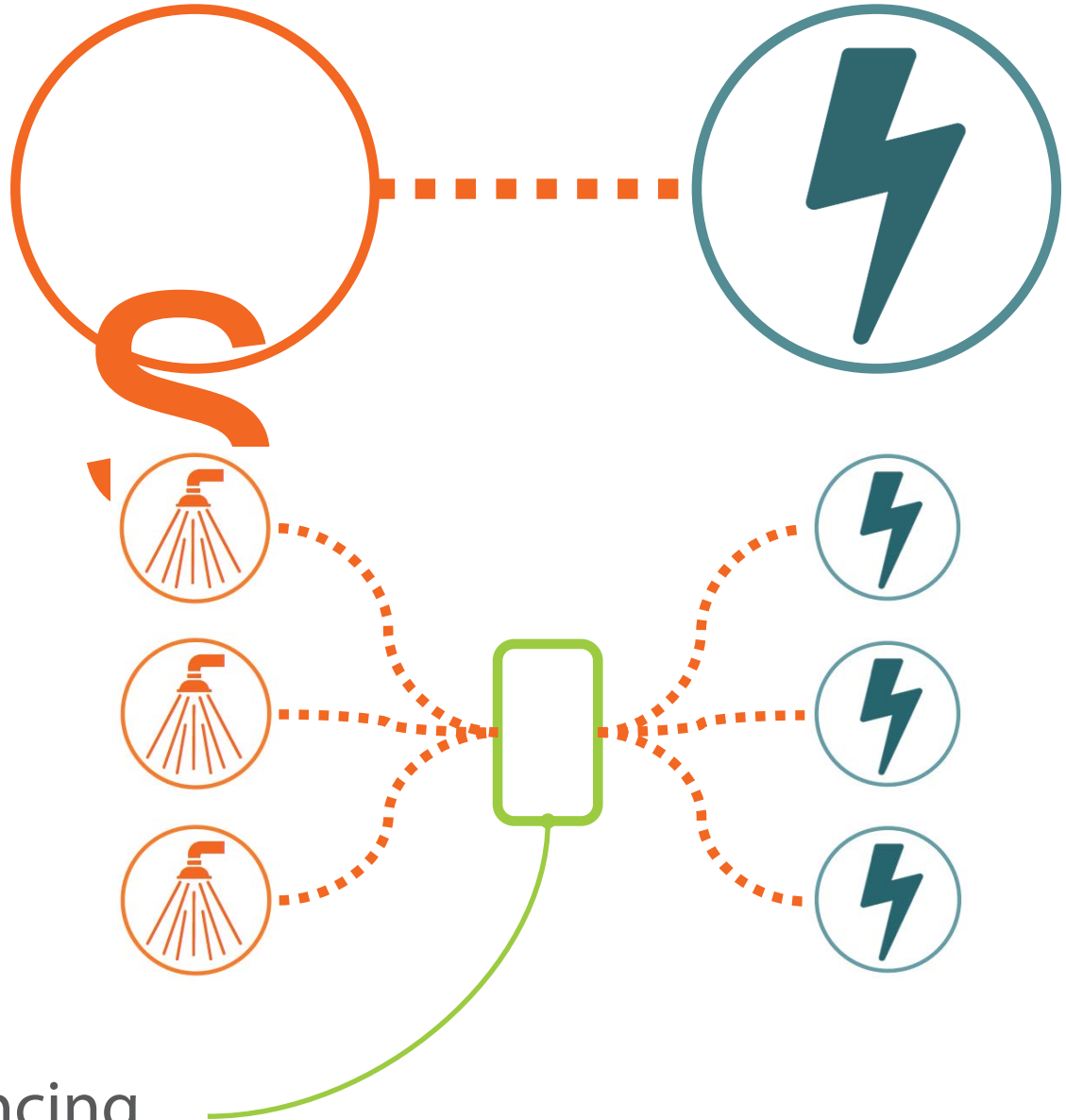






EventHubSpout

x16 executors



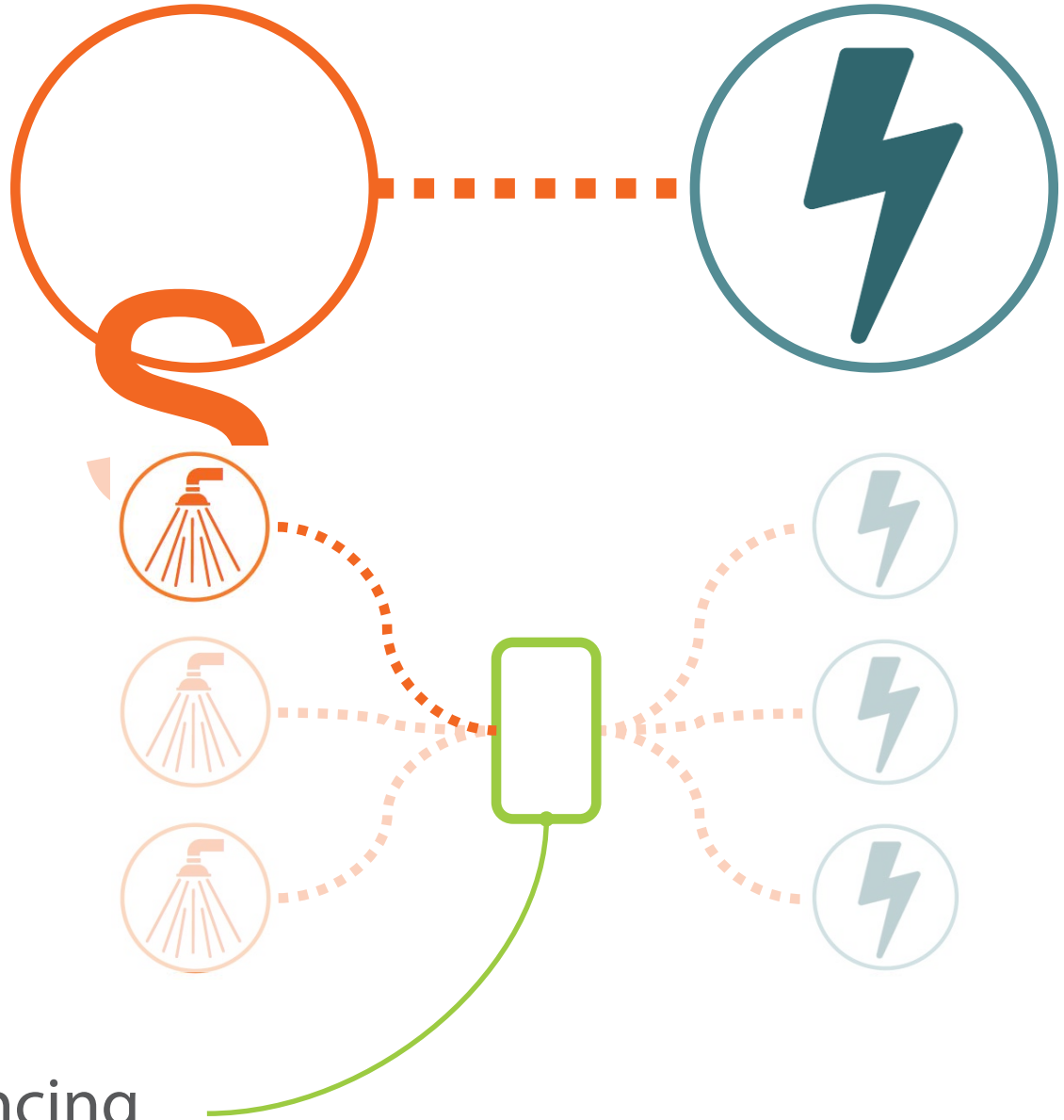
DeviceEventBolt

x16 executors

Stateless

Random load-balancing

EventHubSpout

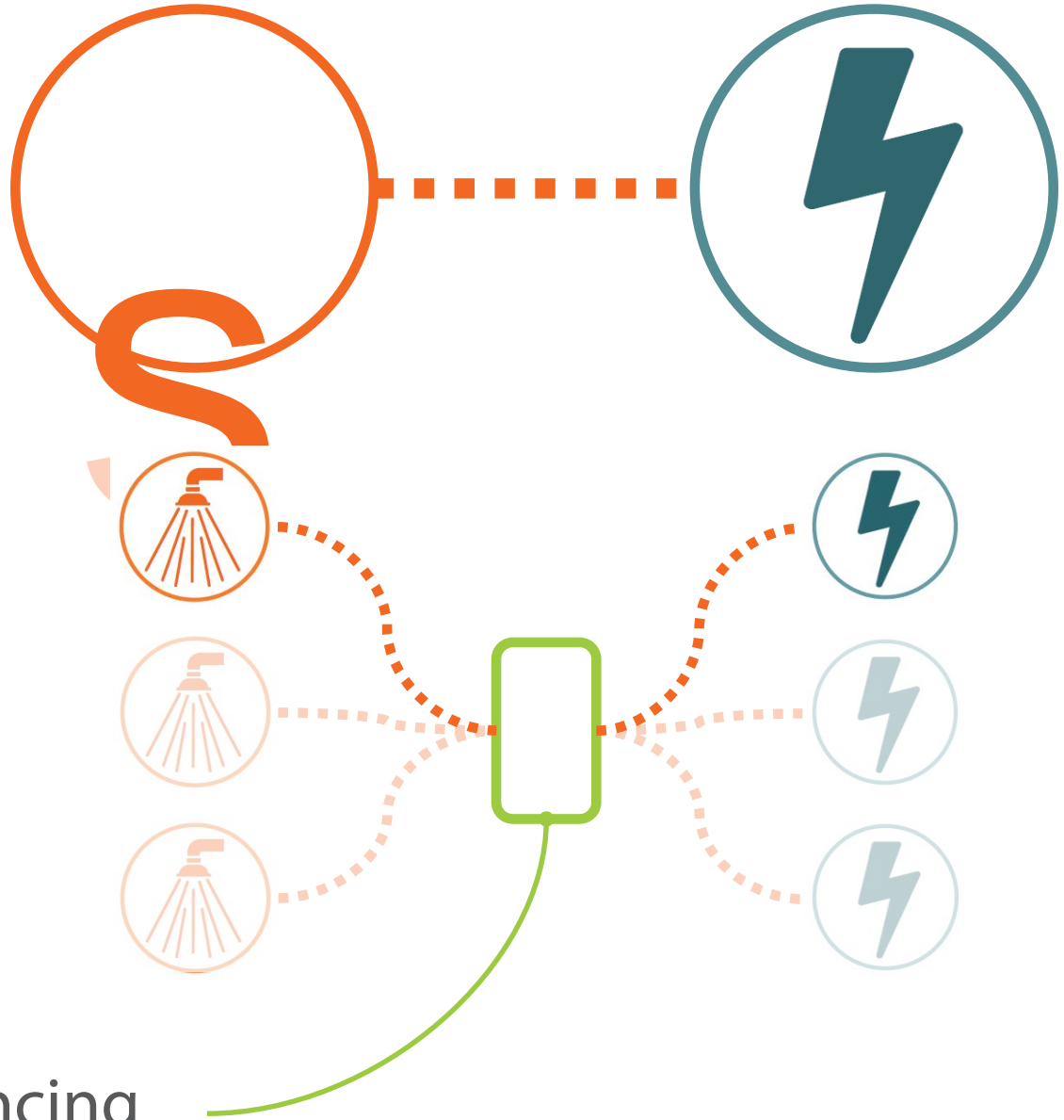


DeviceEventBolt

Stateless

Random load-balancing

EventHubSpout

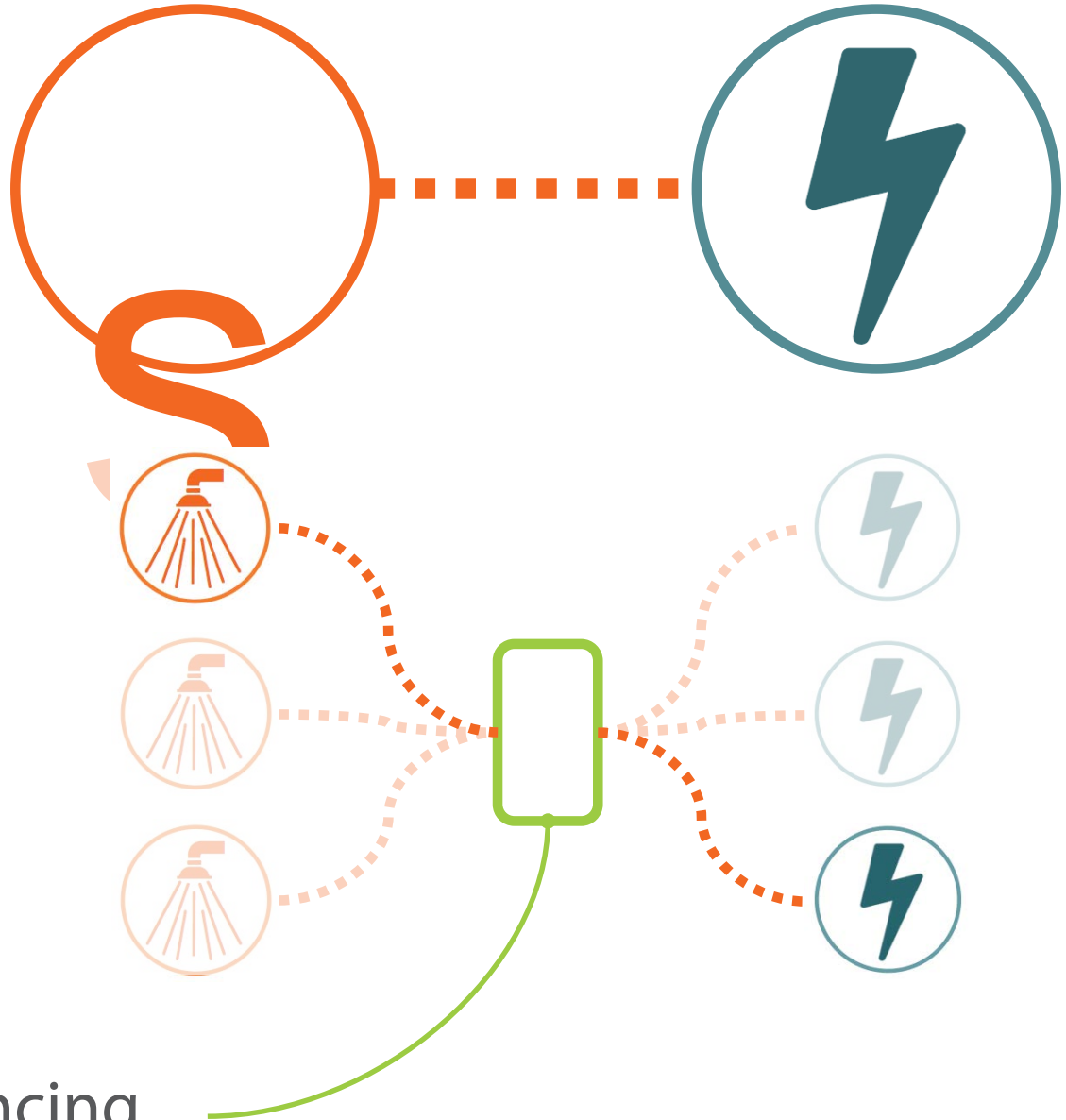


DeviceEventBolt

Stateless

Random load-balancing

EventHubSpout

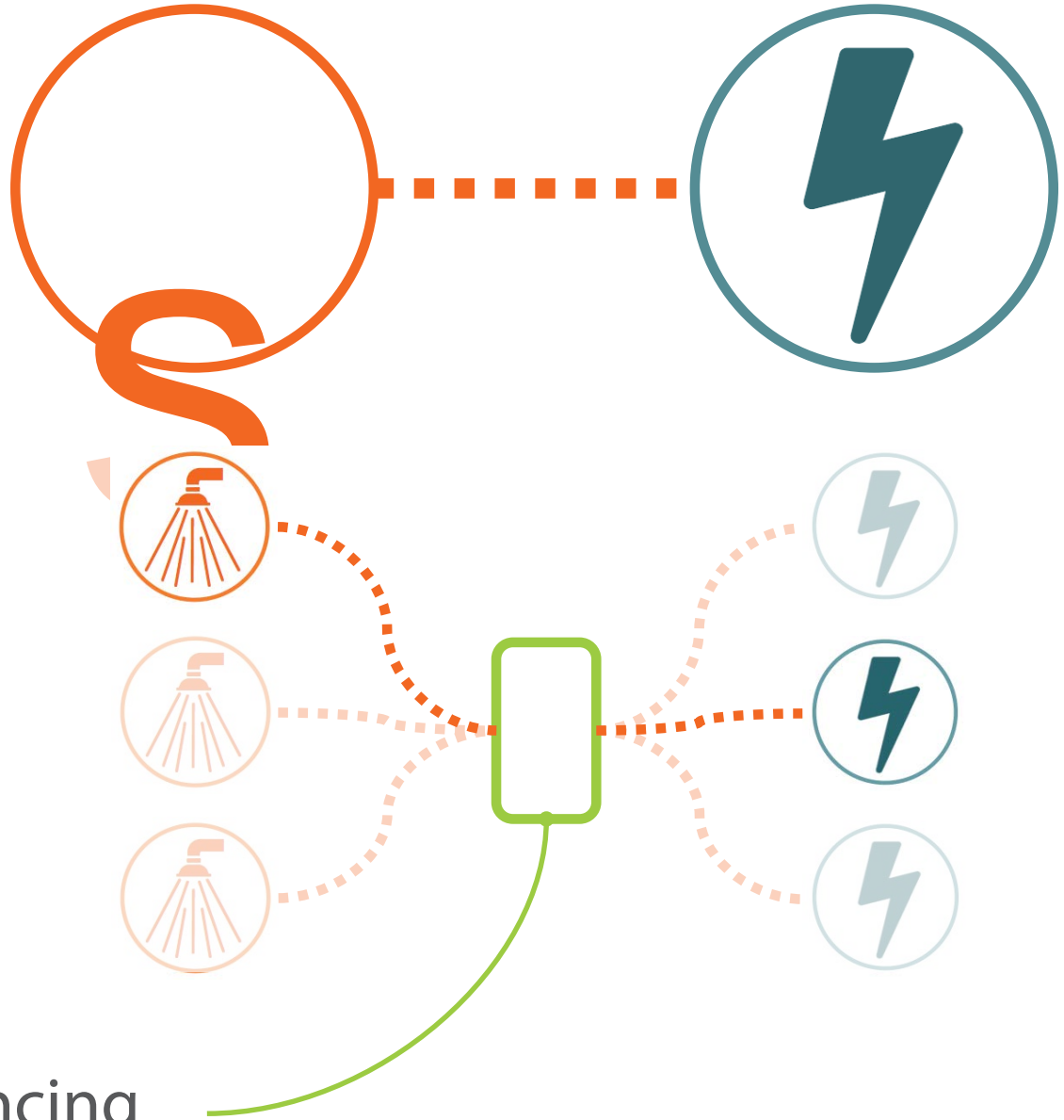


DeviceEventBolt

Stateless

Random load-balancing

EventHubSpout



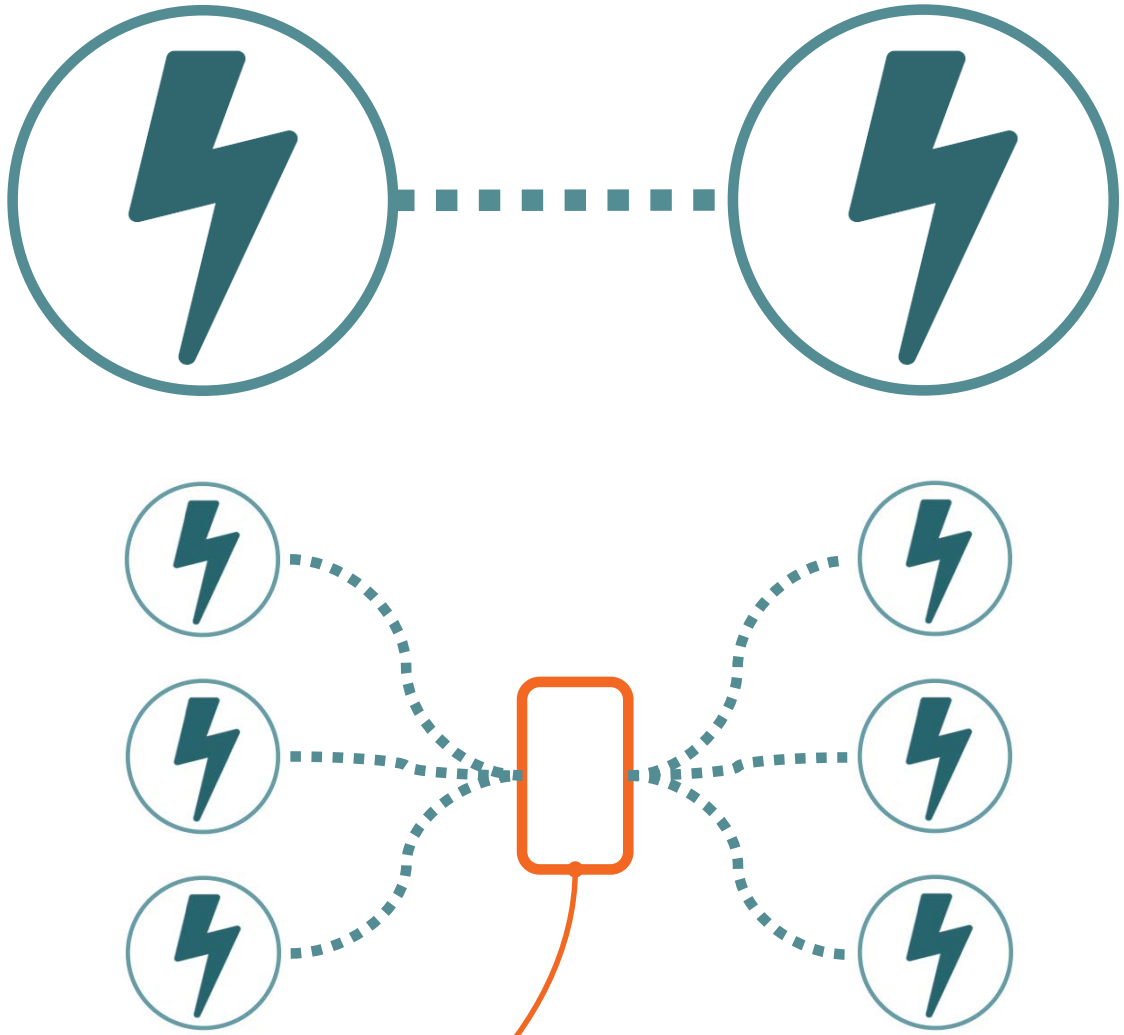
DeviceEventBolt

Stateless

Random load-balancing

DeviceEventBolt

x16 executors



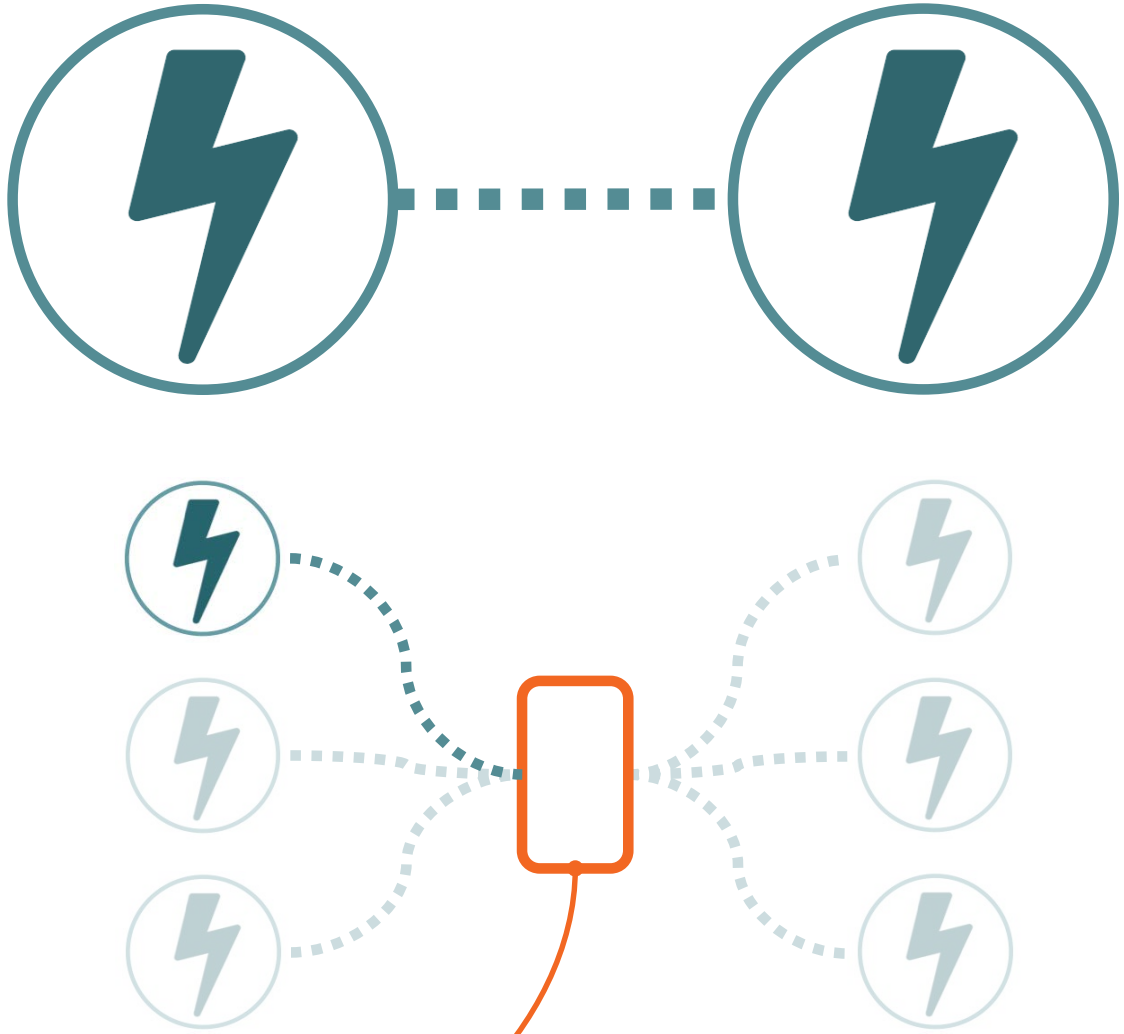
EventMetricBolt

x4 executors

Stateful

Field value load-balancing

DeviceEventBolt

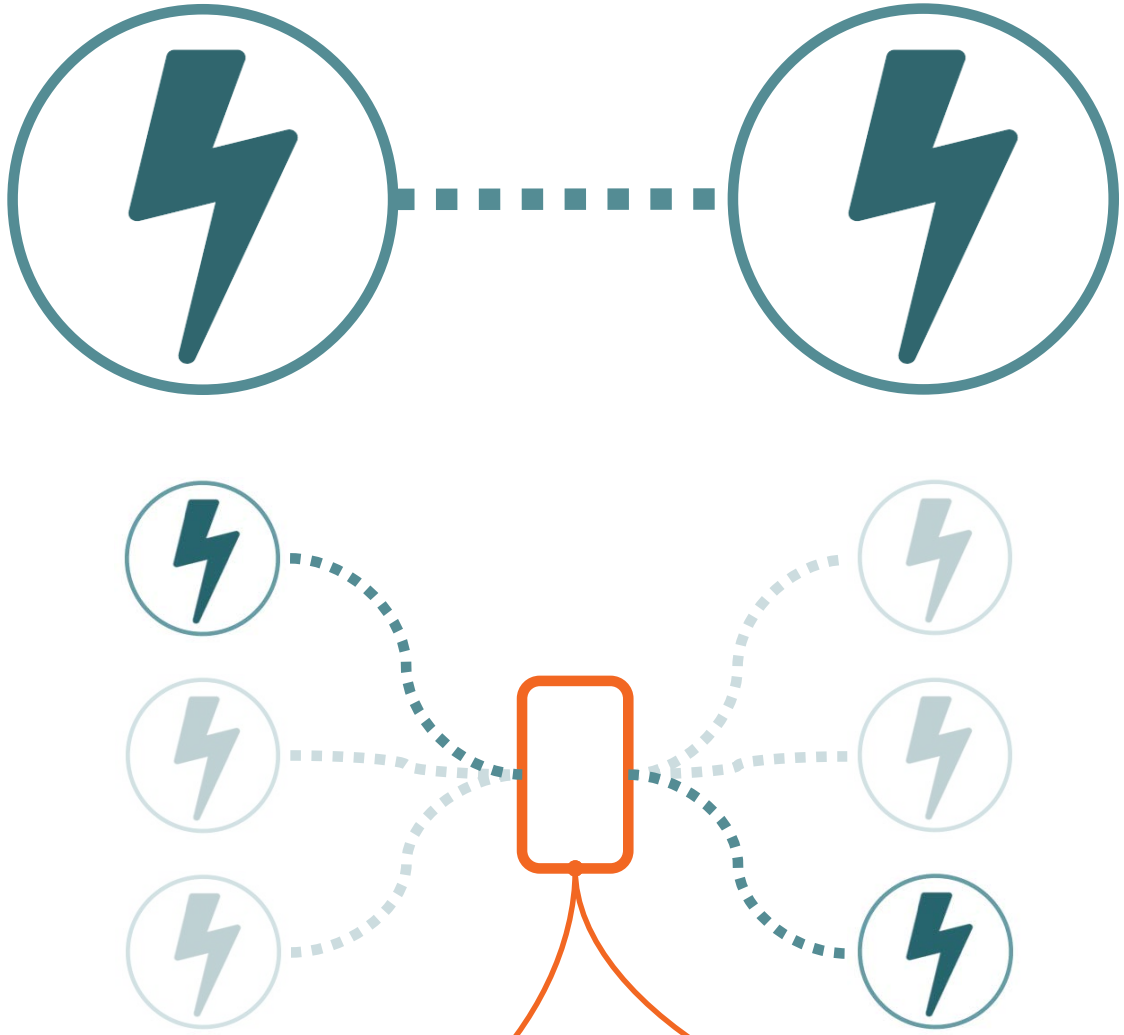


EventMetricBolt

Stateful

Field value load-balancing

DeviceEventBolt



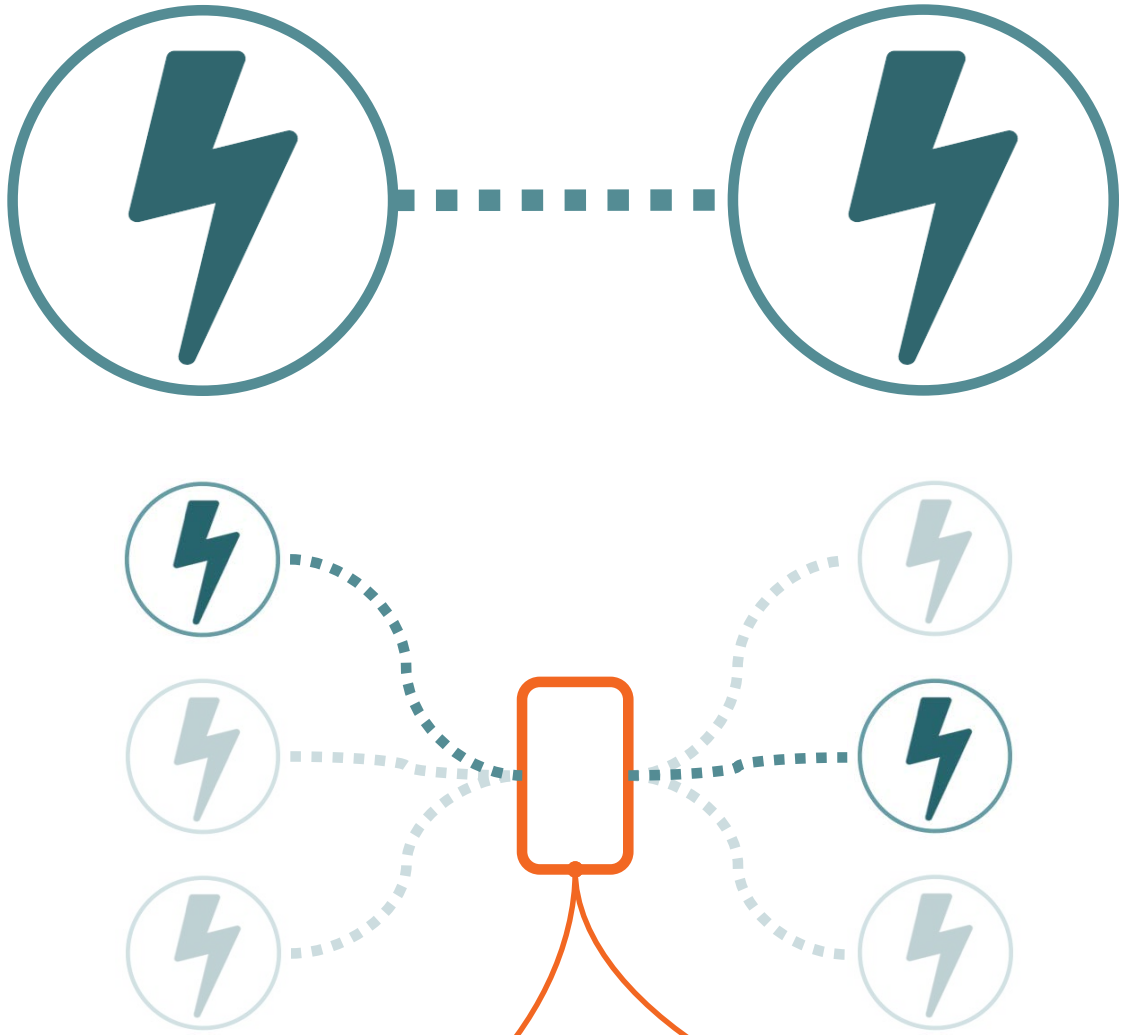
EventMetricBolt

Stateful

Field value load-balancing

Tuple	
eventName	**abc**

DeviceEventBolt



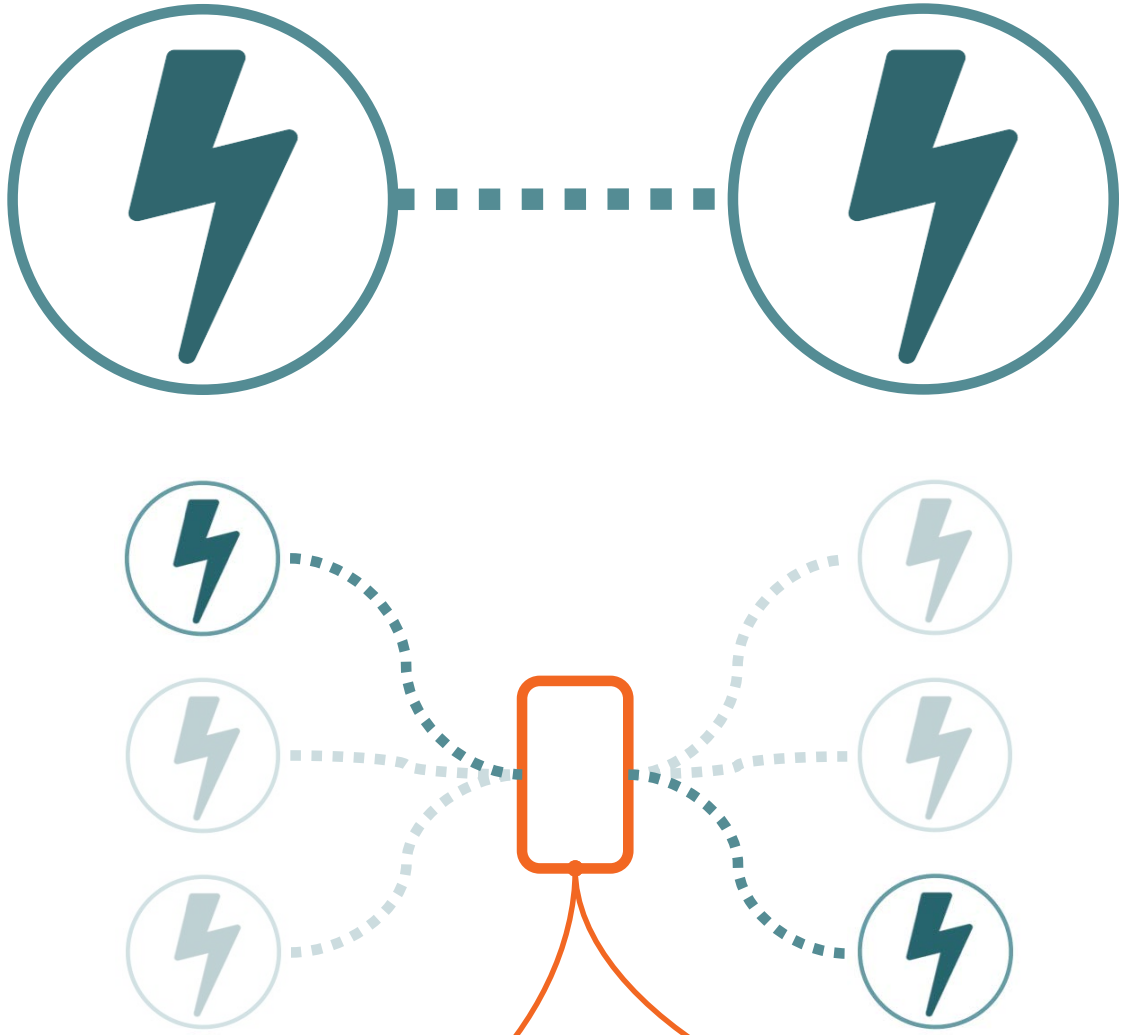
EventMetricBolt

Stateful

Field value load-balancing

Tuple	
eventName	123

DeviceEventBolt



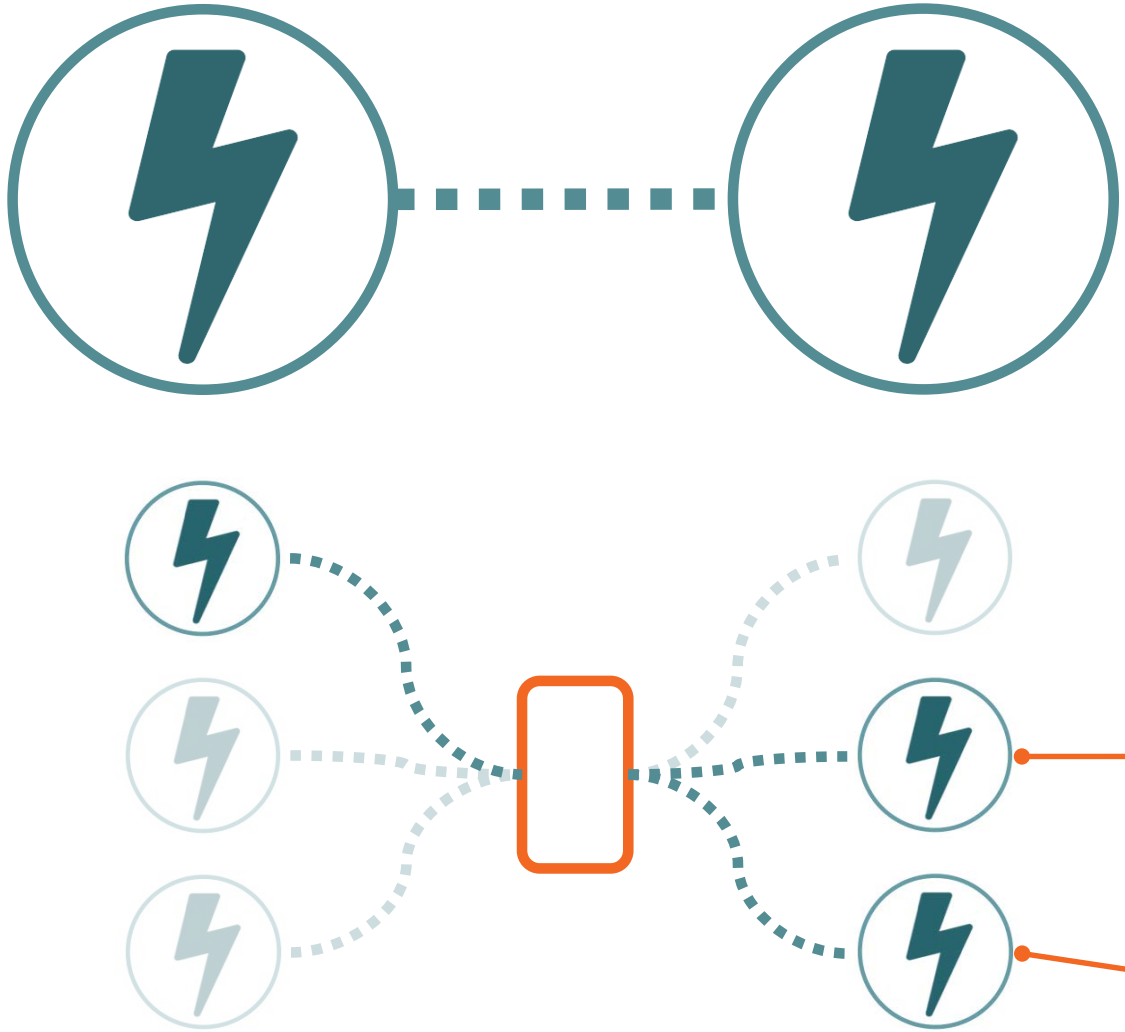
EventMetricBolt

Stateful

Field value load-balancing

Tuple	
eventName	**abc**

DeviceEventBolt

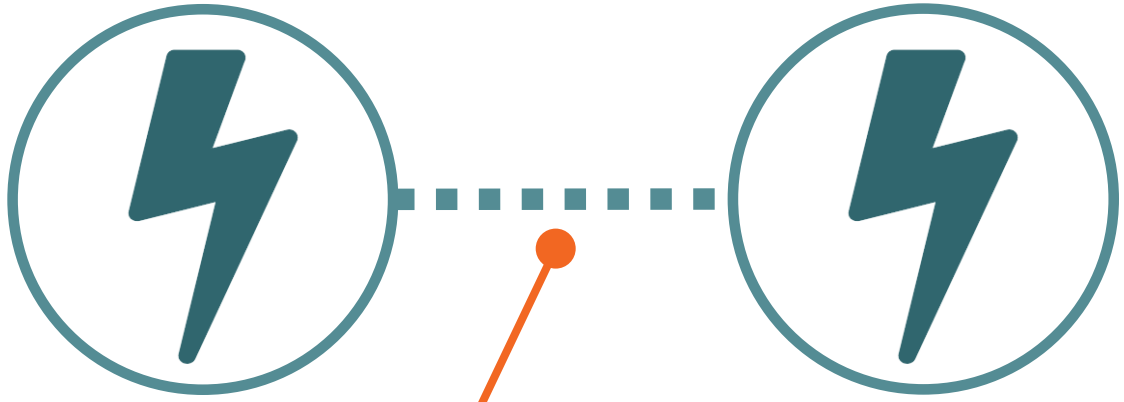


EventMetricBolt

eventName	123
count	1

eventName	abc
count	2

DeviceEventBolt

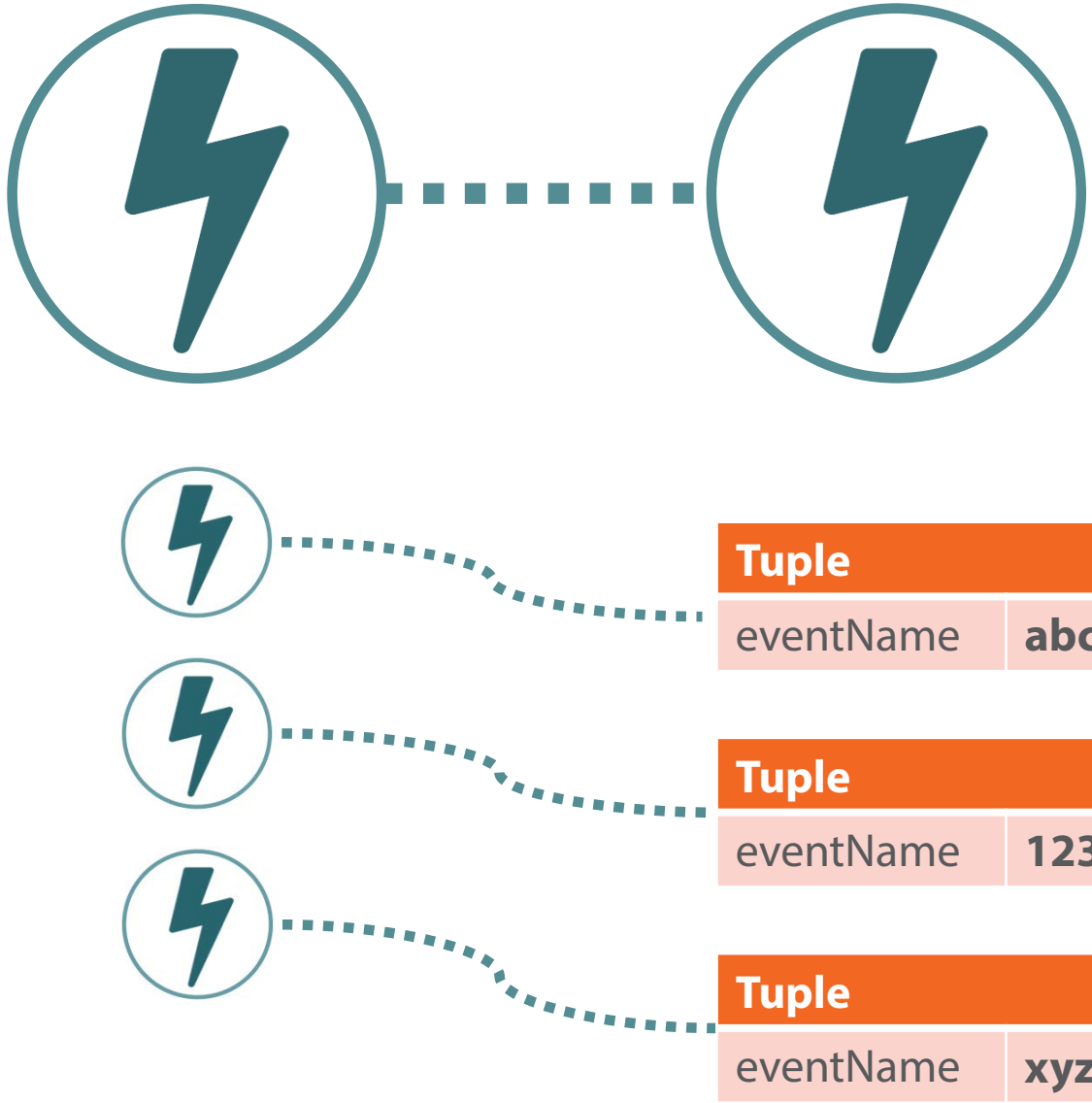


EventMetricBolt

Field grouping

By **eventName**

DeviceEventBolt

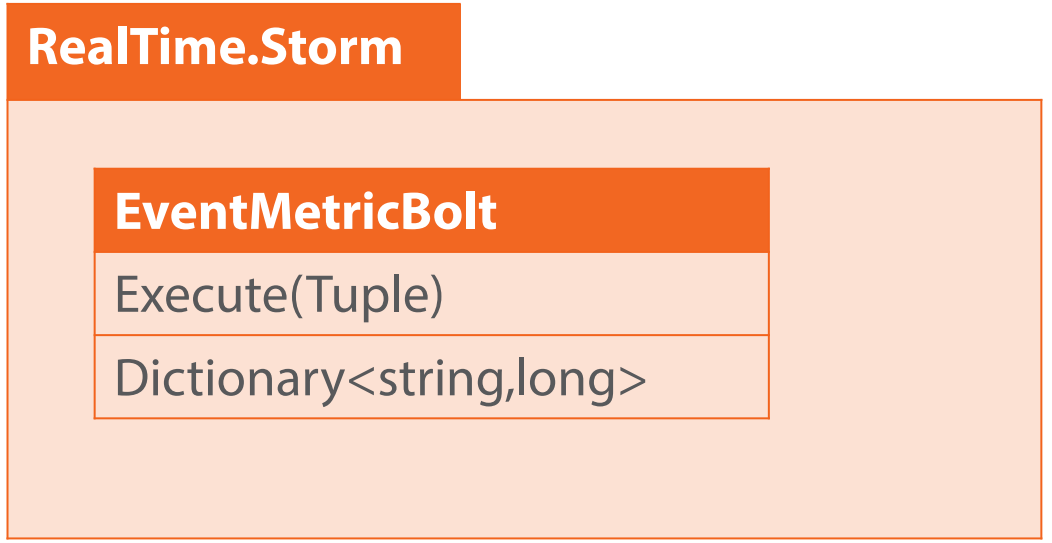


EventMetricBolt

```
{
    "timestamp": 1415878188000,
    "message": "Unhandled exception",
    "eventName": "device.logs.message",
    "deviceId": "1e565484571752"
}
```

eventName	device.logs.message
receivedAt	14158792533000

partitionId	11

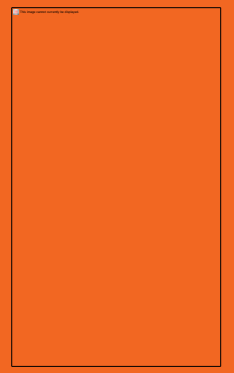


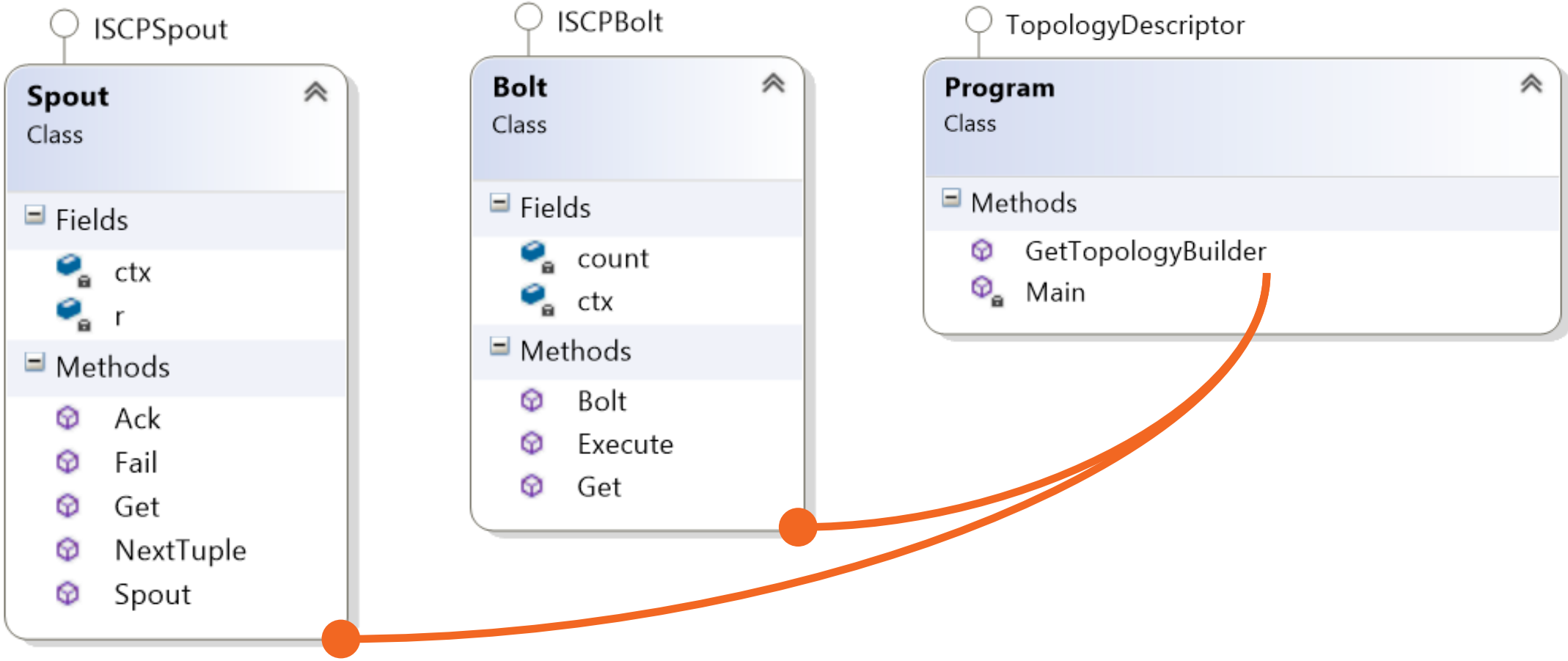
Demo: Storm Application

Storm project template

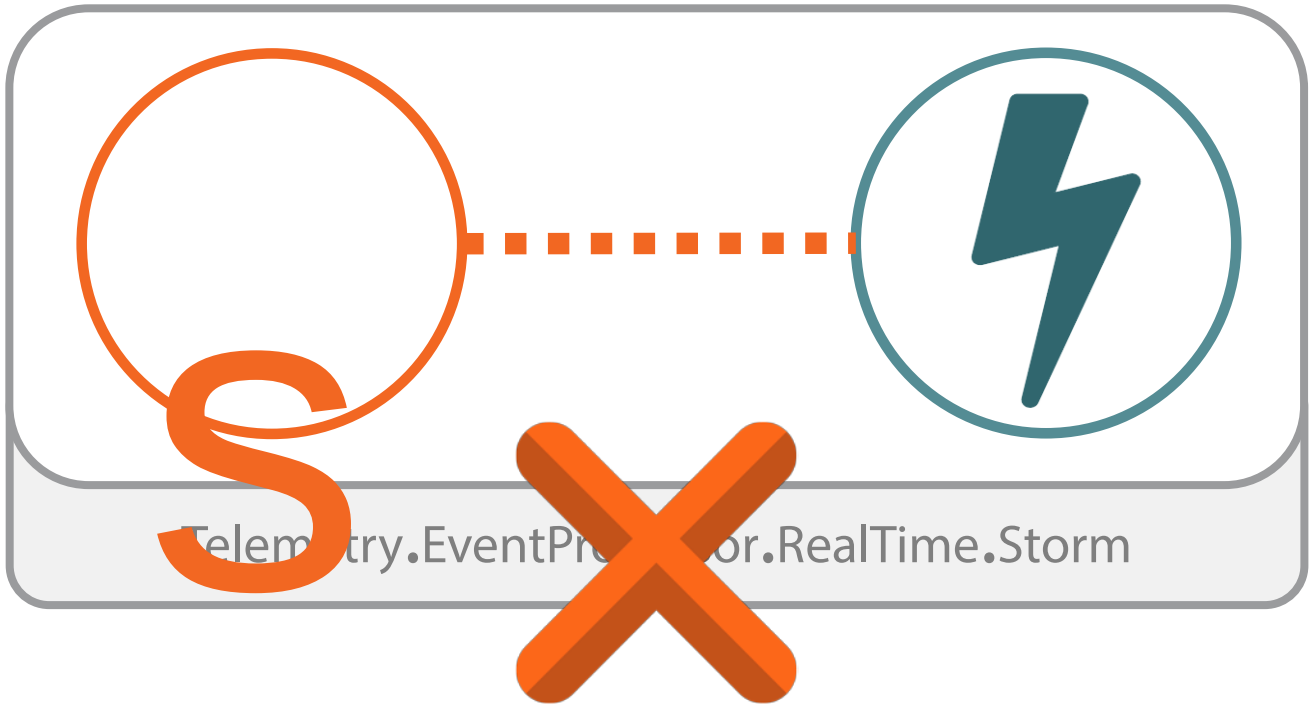
Event Hub Spout

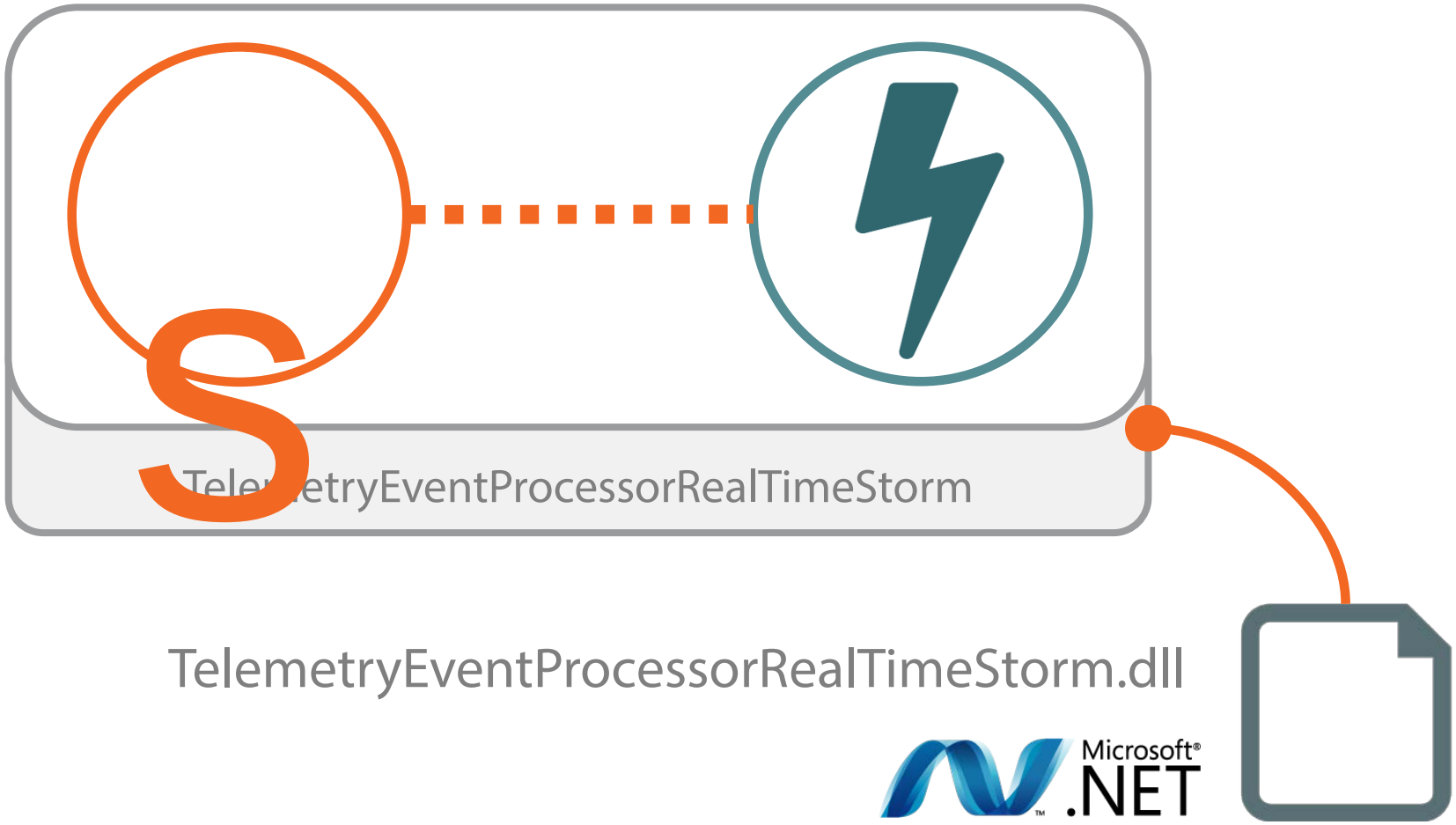
Bolt receiver

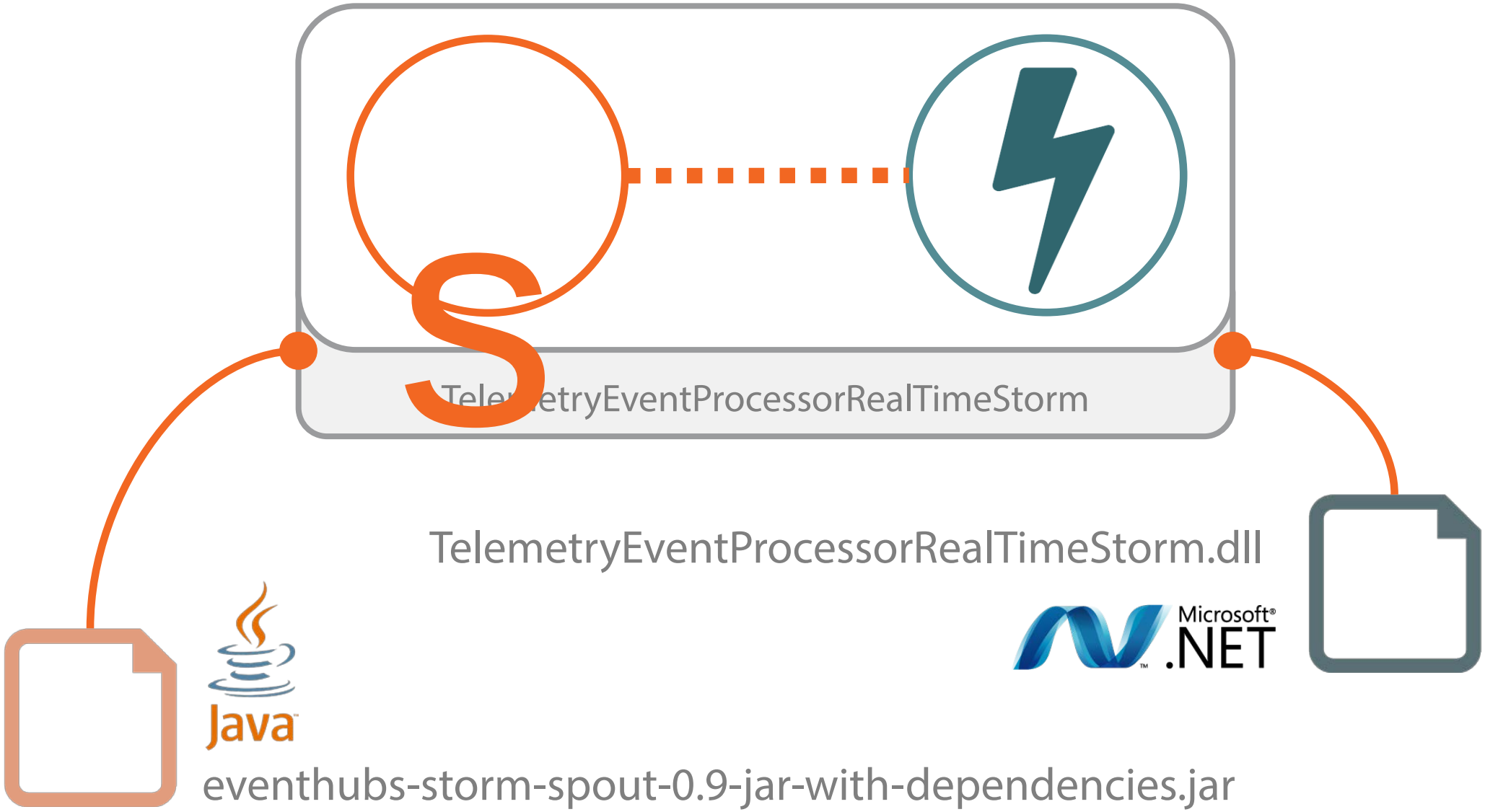




Storm project template







```
topologyBuilder.SetBolt("DeviceEventBolt"...
               .DeclareCustomizedJavaSerializer(customSerializer)
               .shuffleGrouping("EventHubSpout")
```

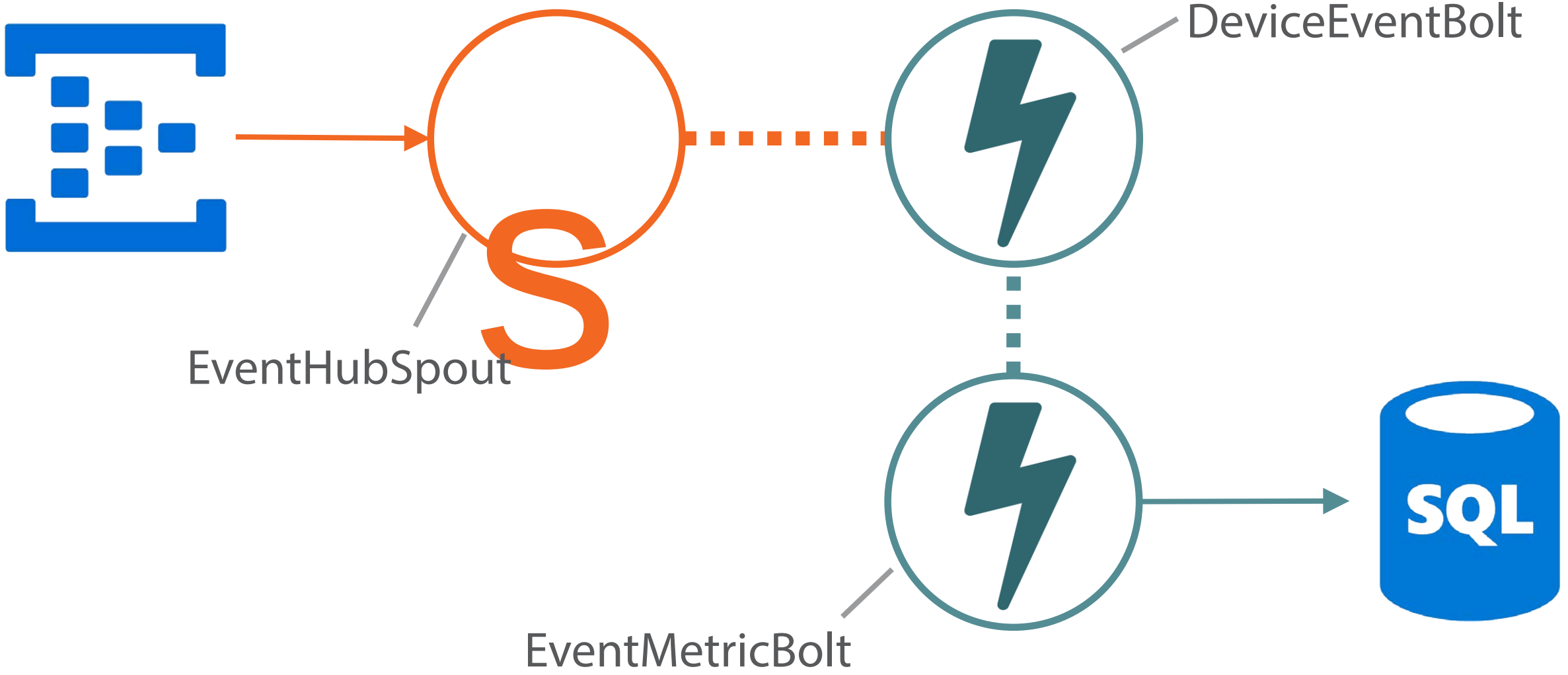
Interoperable serialization

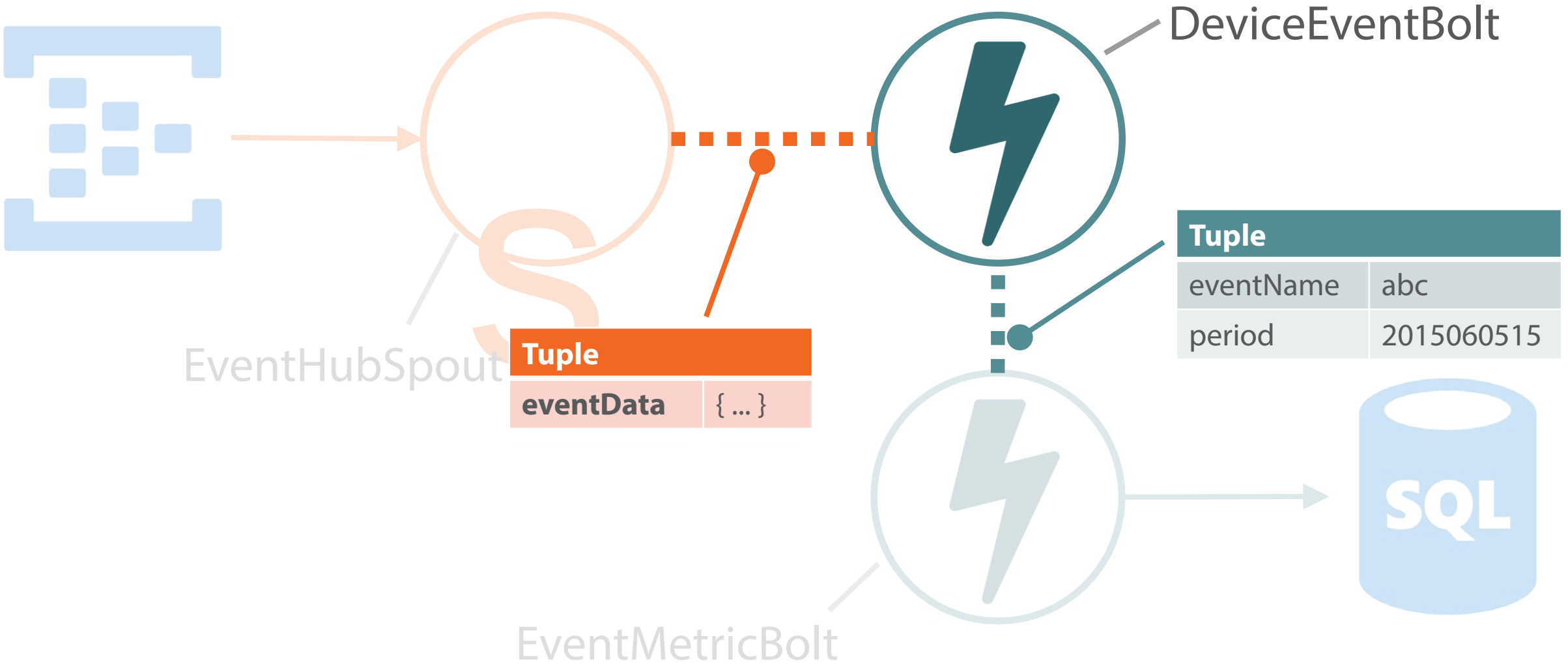
From Java spout to .NET bolt

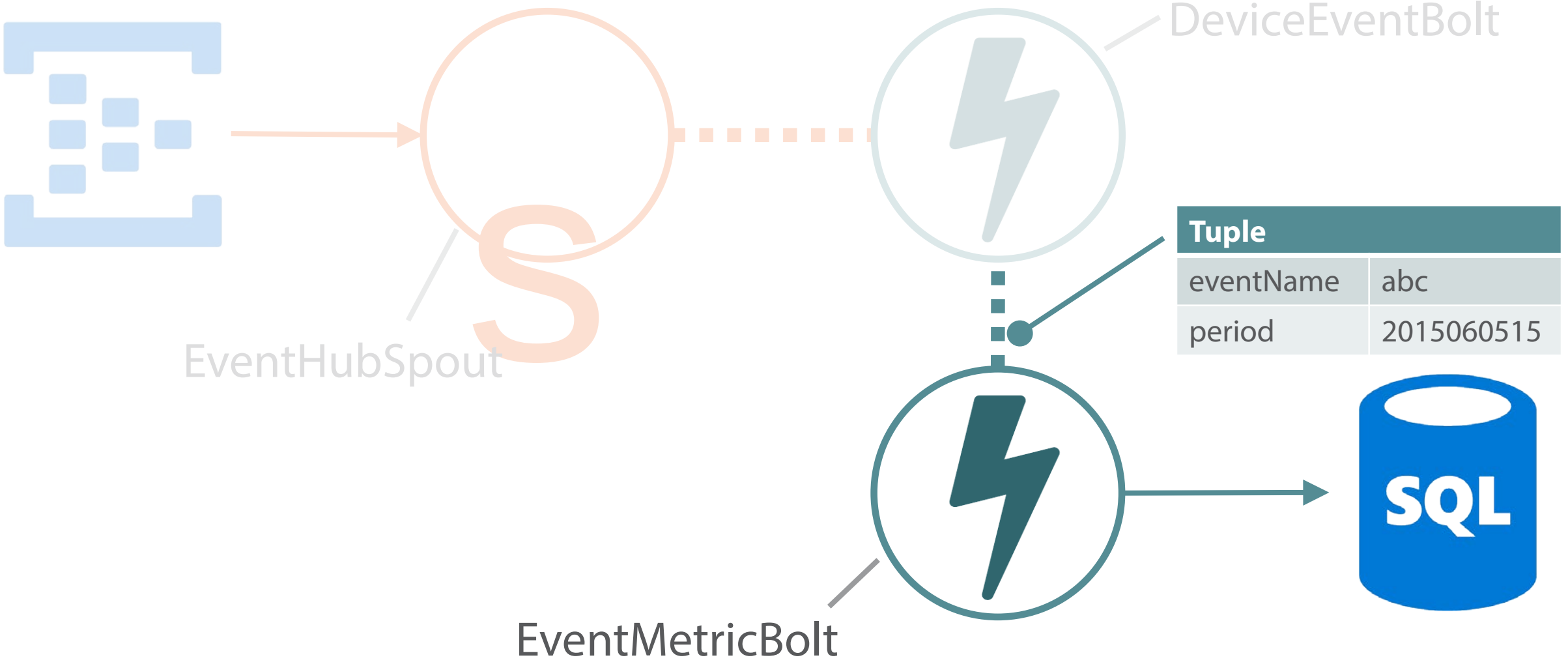
```
var inputSchema = new Dictionary<string, List<Type>>();
inputSchema.Add(Constants.DEFAULT_STREAM_ID,
                new List<Type>() { typeof(string) });
```

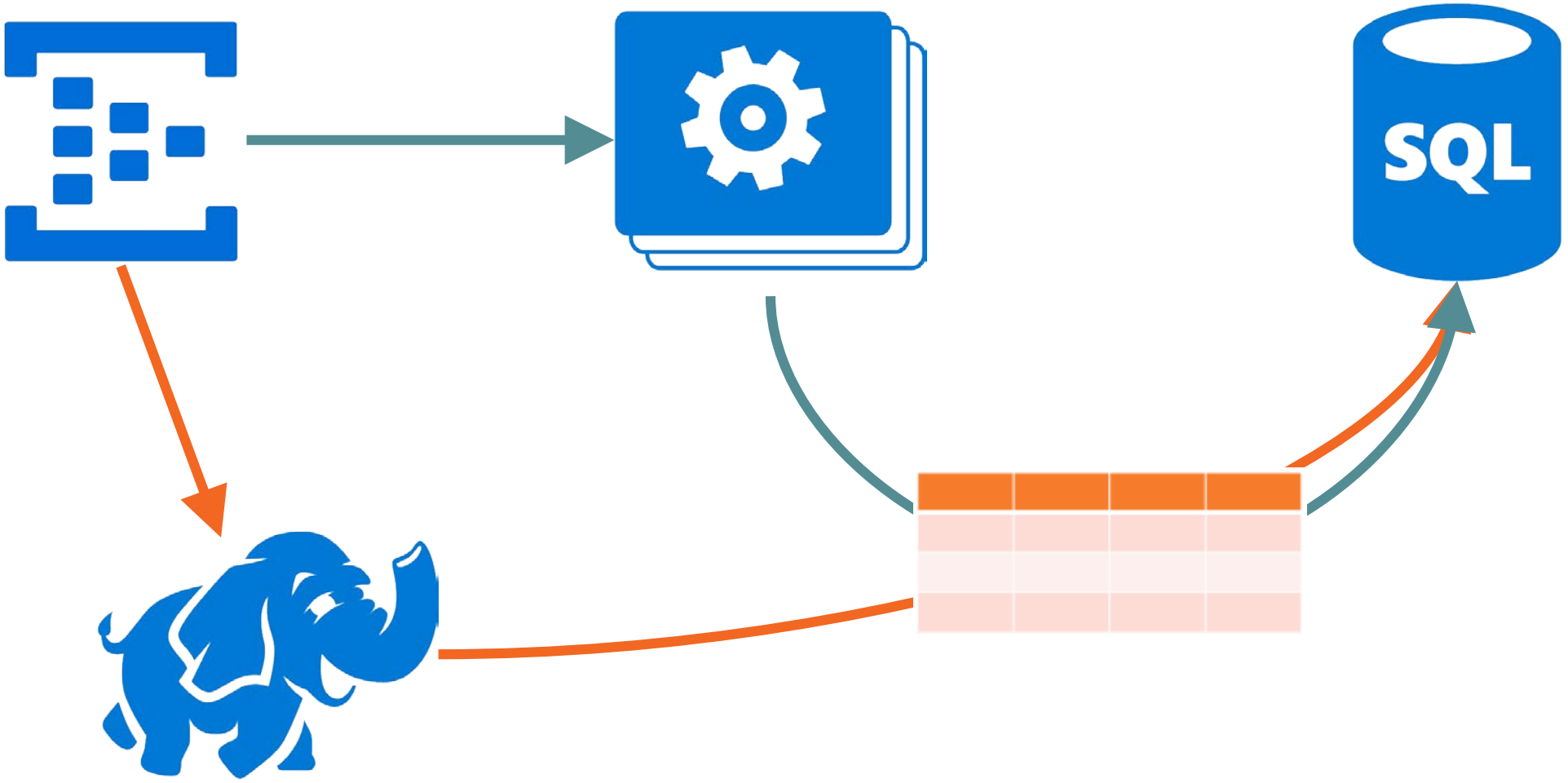
Matching schemas

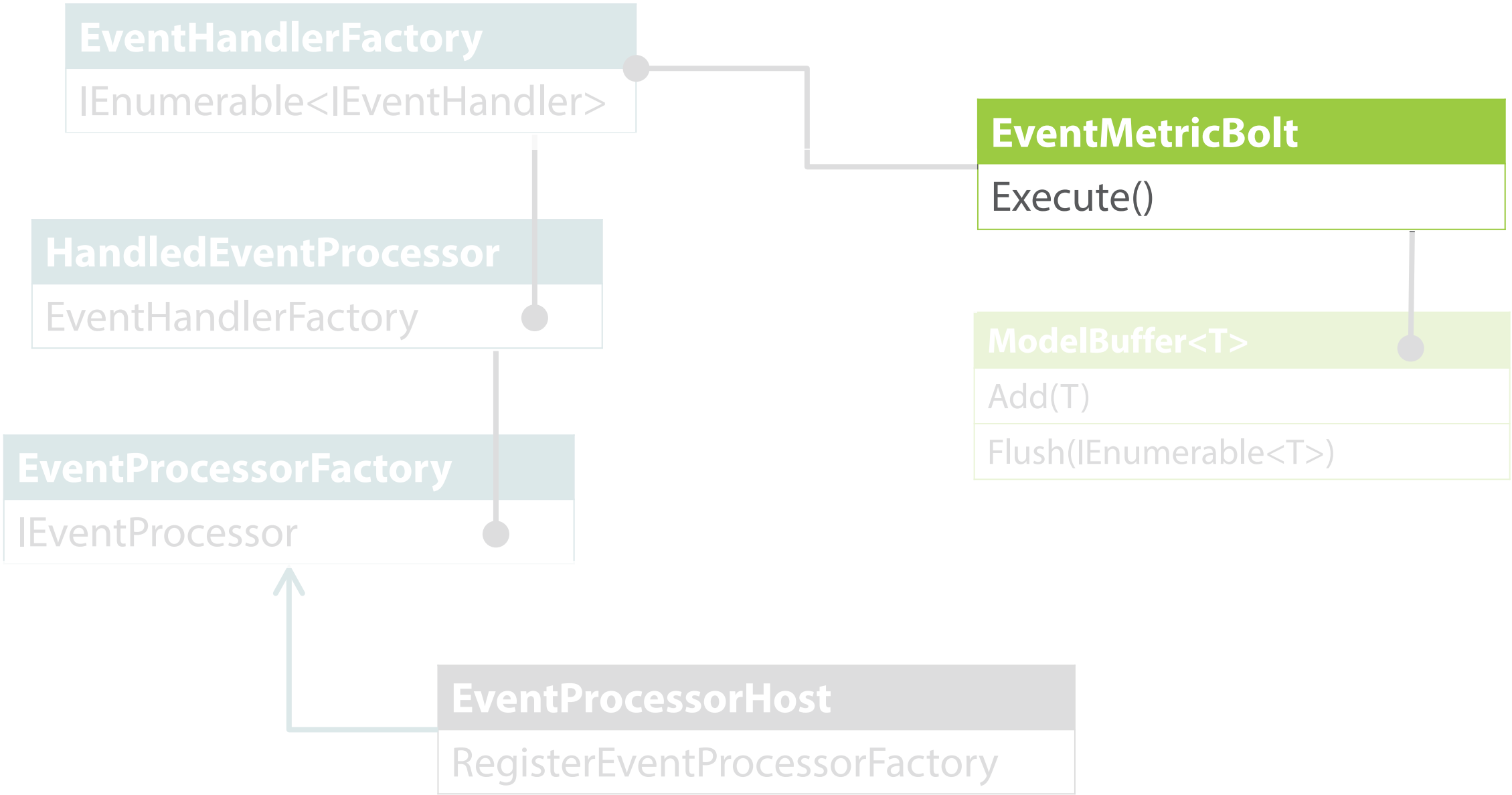
Output from spout and input to bolt

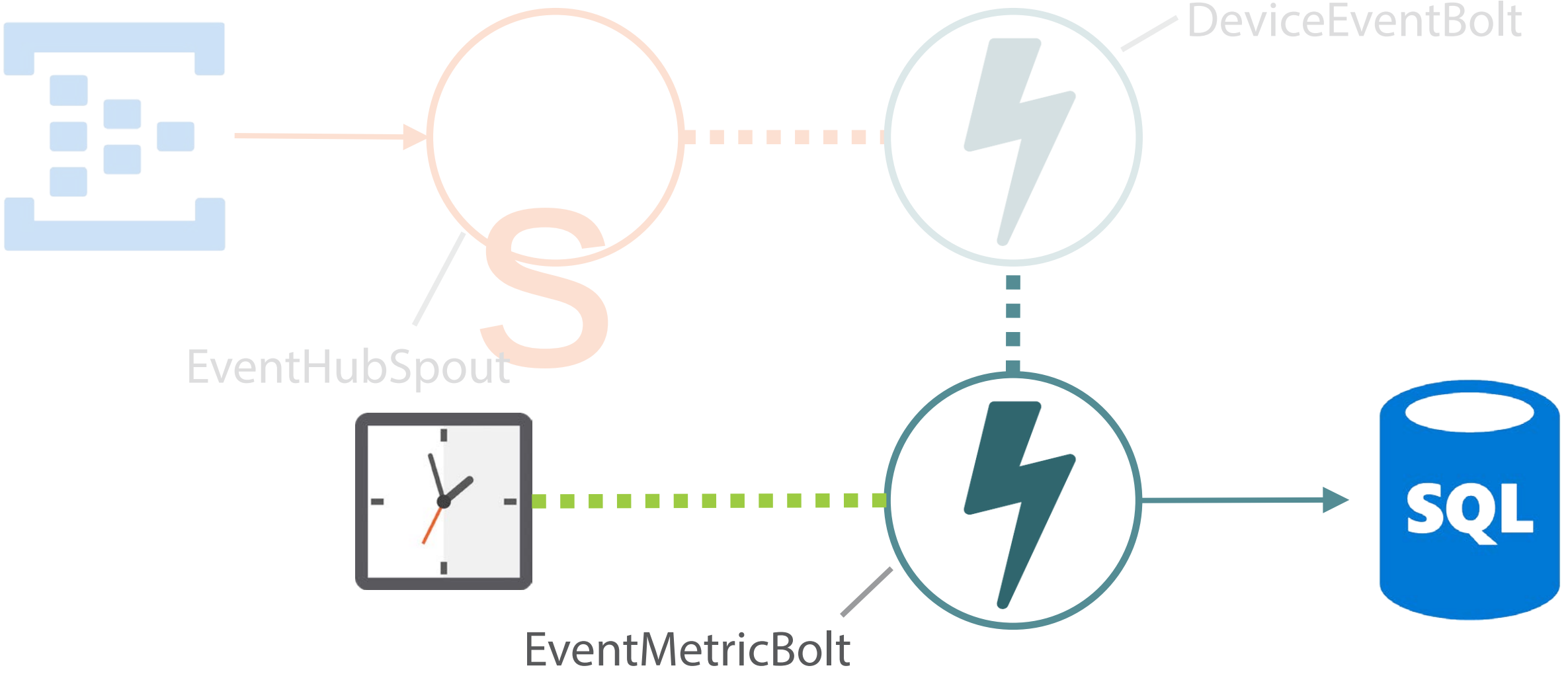

DeviceEventBolt
EventHubSpout
EventMetricBolt
SQL

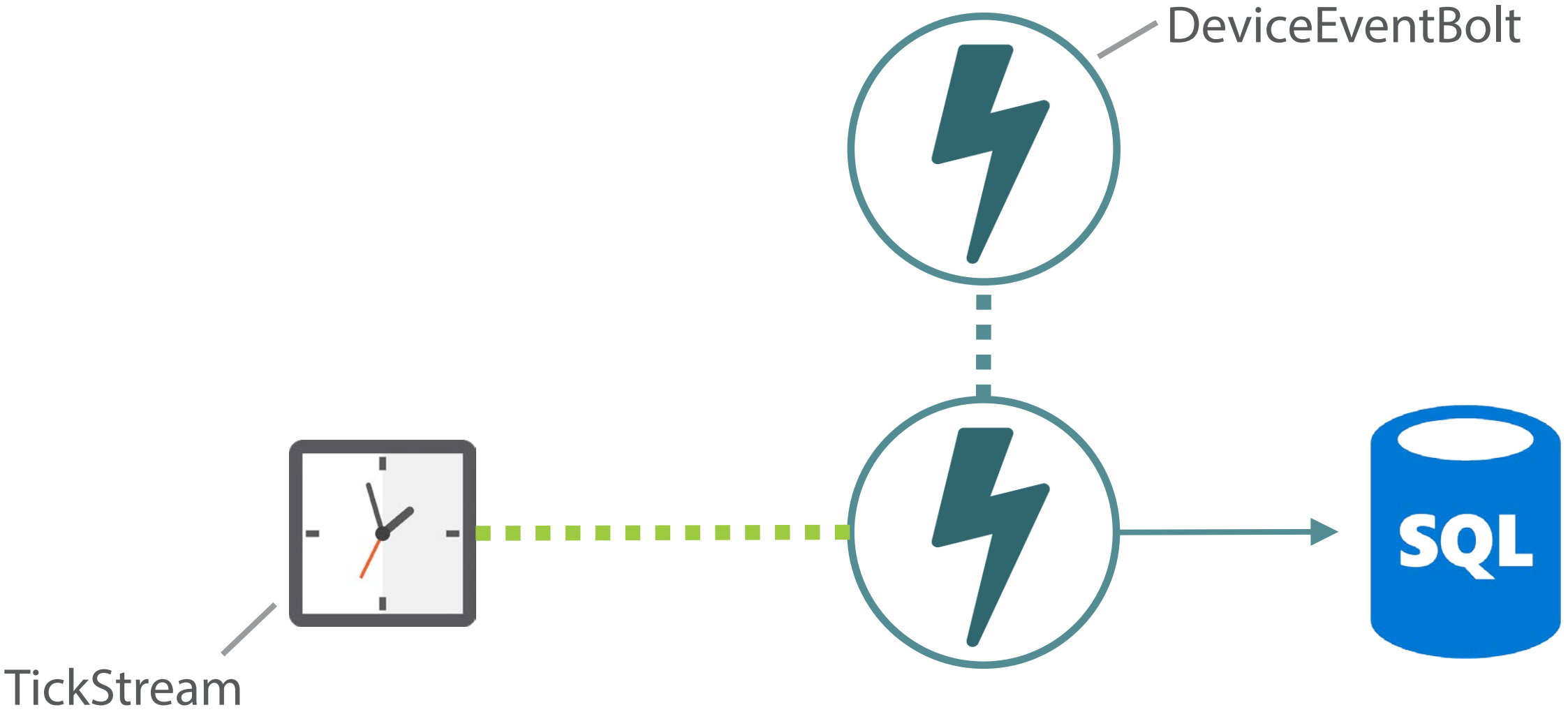

DeviceEventBolt
Tuple
eventName
abc
period
2015060515
Tuple
eventData
{ ... }
EventHubSpout
EventMetricBolt
SQL

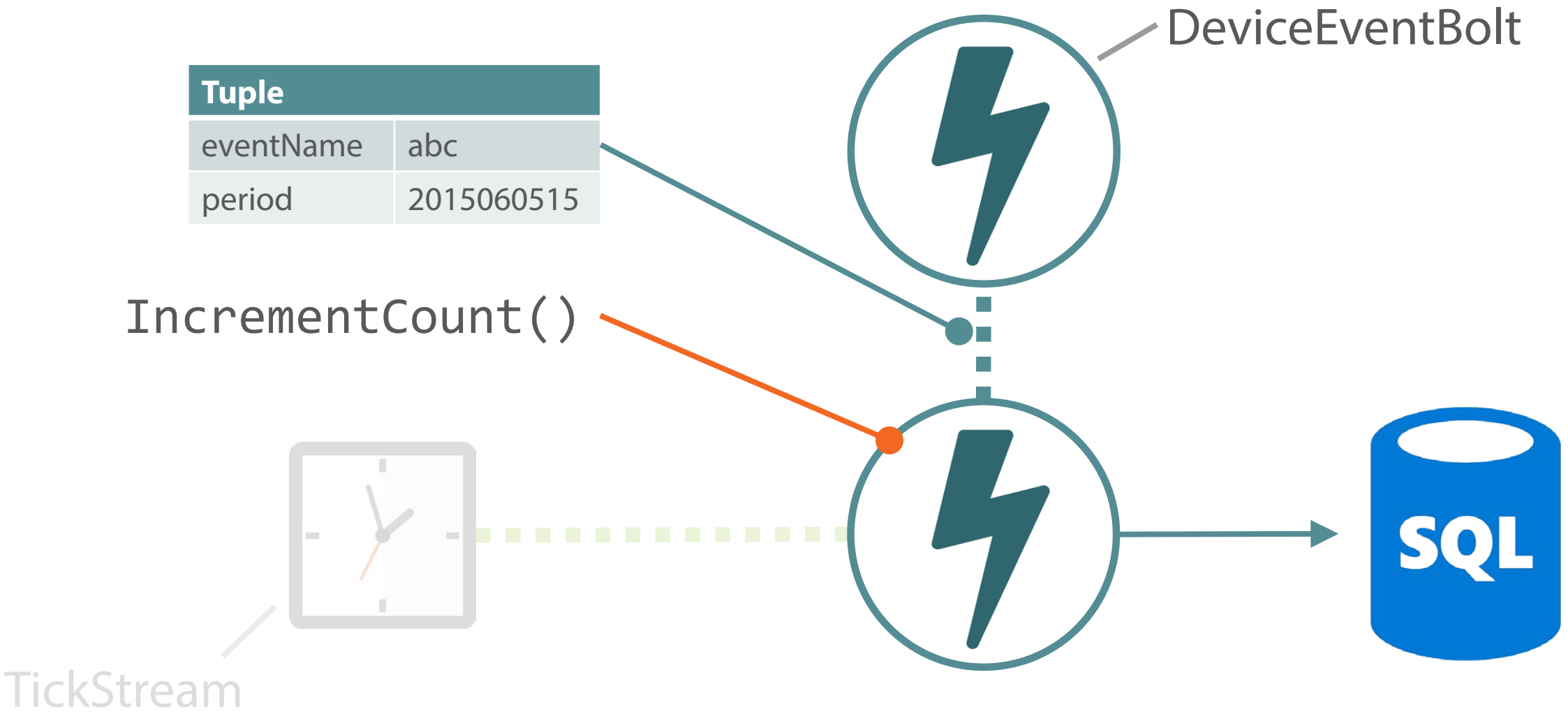

DeviceEventBolt
EventHubSpout
Tuple
eventName
abc
period
2015060515
SQL
EventMetricBolt

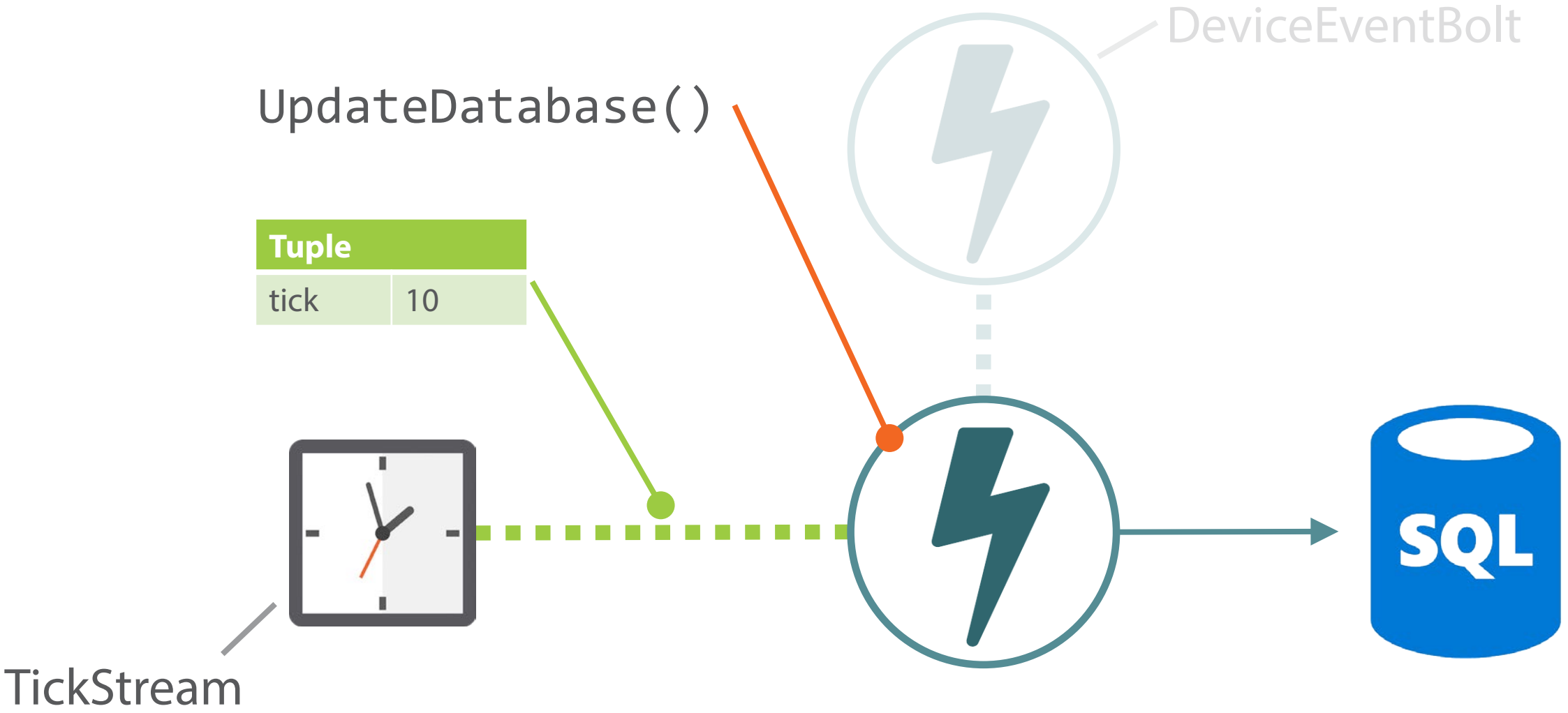


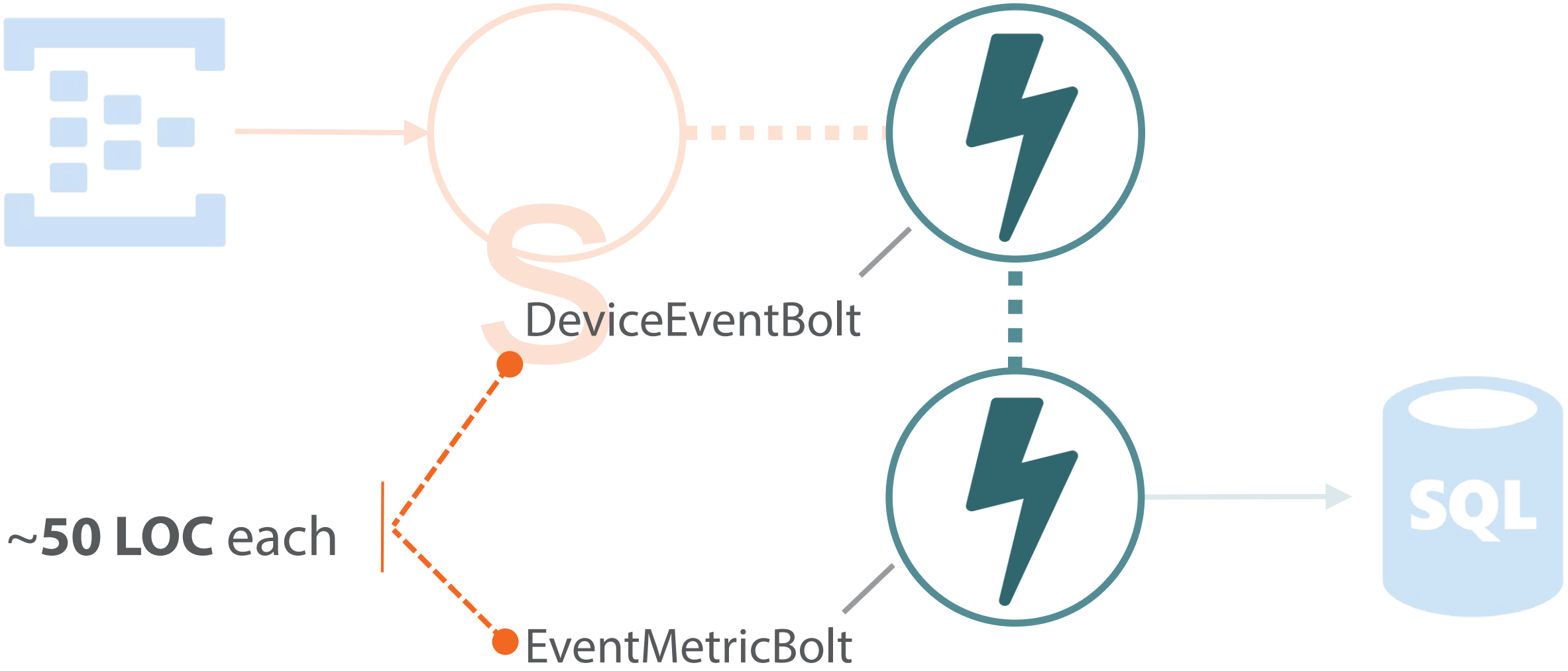





DeviceEventBolt
SQL
TickStream


DeviceEventBolt
Tuple
eventName
abc
period
2015060515
IncrementCount()
SQL
TickStream





```
var eventJson = (string)tuple.GetString(0);
var deviceEvent = JsonConvert.DeserializeObject<DeviceEvent>
                                        (eventJson, _JsonSettings);
```

DeviceEventBolt

Sparse JSON deserialization

```
var inputSchema = new Dictionary<string, List<Type>>();
inputSchema.Add(Constants.DEFAULT_STREAM_ID,
            new List<Type>() { typeof(string), typeof(string) });
inputSchema.Add(Constants.SYSTEM_TICK_STREAM_ID,
                                new List<Type>() { typeof(long) });
```

EventMetricBolt

Schemas for two input streams

```
var key = string.Format("{0}|{1}", eventName, period);
//...
_eventCounts[key] = eventCount;
```

Execute tuple from EventMetricBolt

Increment count for event

```
foreach (var key in _eventCounts.Keys)
{
    //...
    db.MergeEventMetric(eventName, period, count);
}
```

Execute tuple from TickStream

Push counts to SQL Server

```
topologyBuilder.SetBolt("DeviceEventBolt"...
               .DeclareCustomizedJavaSerializer(javaSerializerInfo)
               .shuffleGrouping("EventHubSpout")
```

Topology builder

Shuffle grouping from spout to DeviceEventBolt

```
topologyBuilder.SetBolt("EventMetricBolt"...
                .fieldsGrouping("DeviceEventBolt",
                                new List<int>() { 0 })
```

Topology builder

Field grouping from DeviceEventBolt to EventMetricBolt

```
using (var command = _connection.CreateCommand())
{
    command.CommandText = "UpsertEventMetric";
    command.CommandType = CommandType.StoredProcedure;
    //...
    command.ExecuteNonQuery();
```

ADO.NET data logic

Entity Framework graph too complex

HDInsight Limitations

Simple .NET solutions

Awkward Java to .NET interop

Azure runtime only

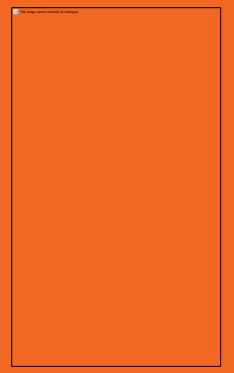
Manual deployment

Demo: Running on HDInsight

Submit to Storm Cluster

Post events to API

Verify summaries in SQL



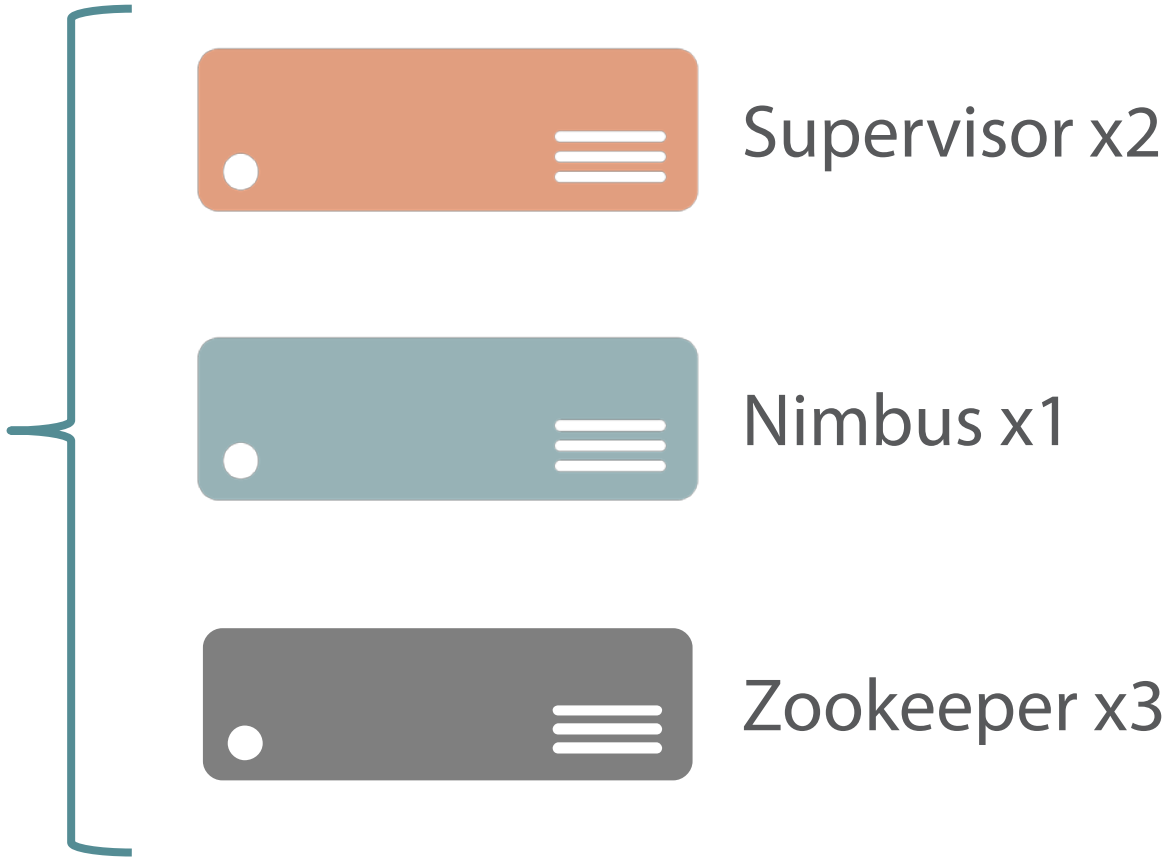
Managed platform

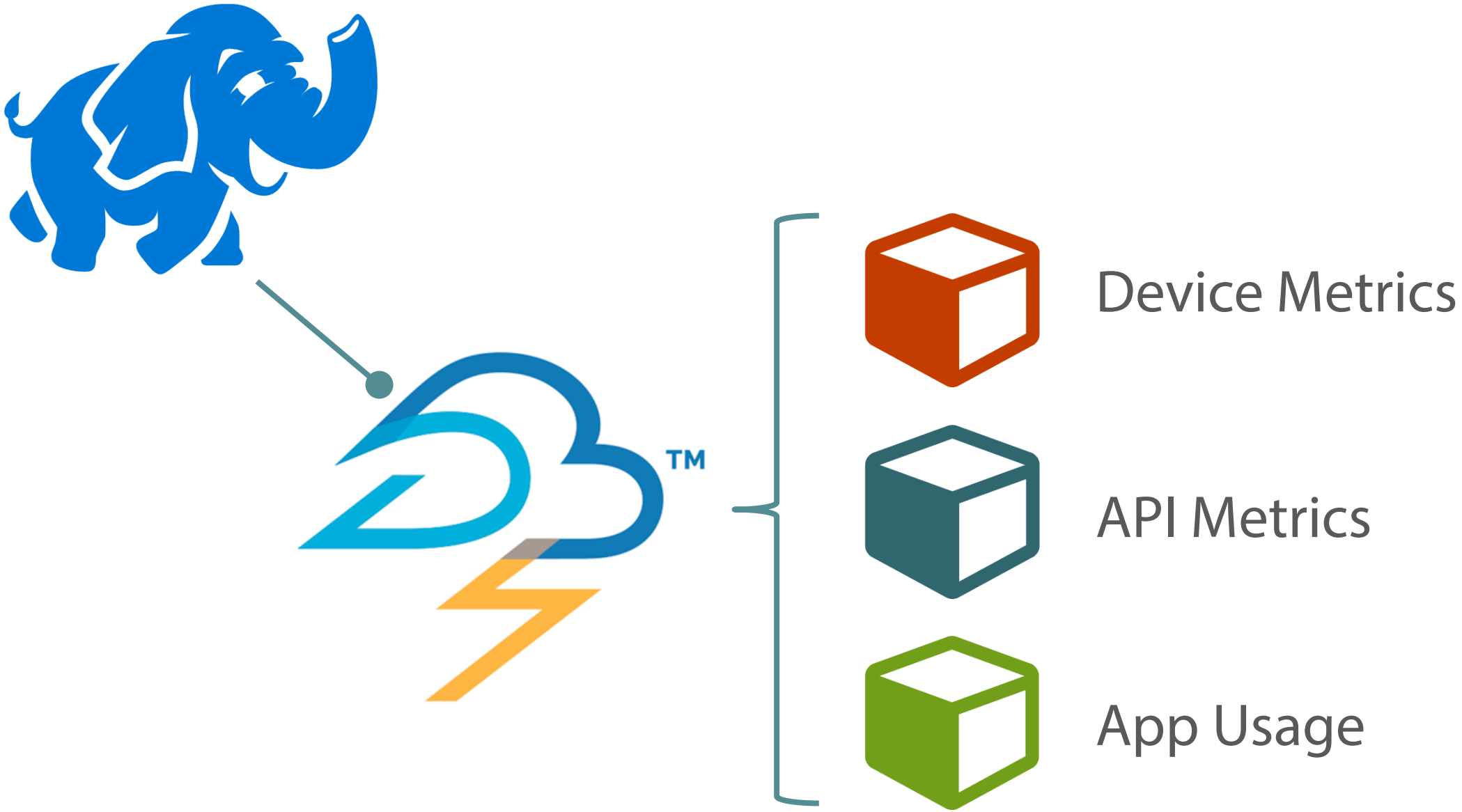
Automated **recovery**

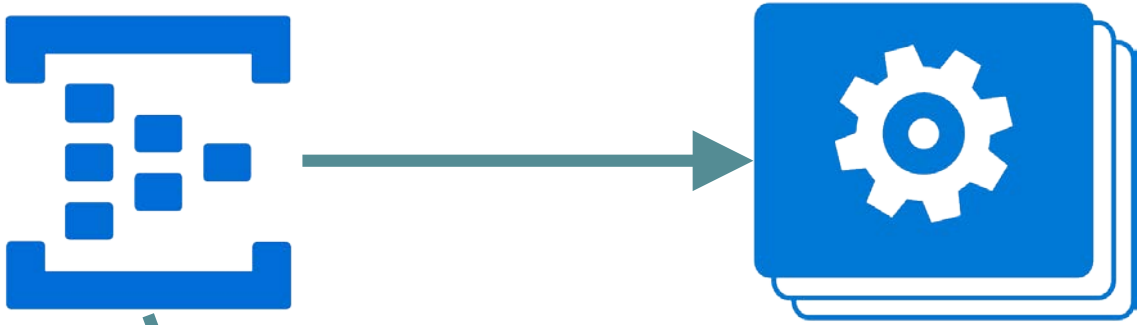
Guaranteed processing

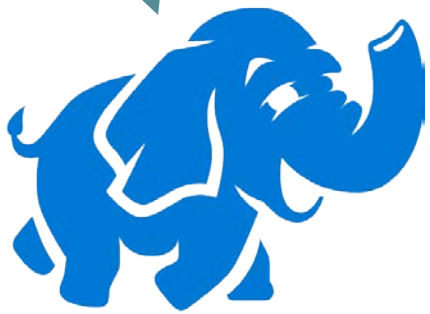
One-click **scaling**

$500+ /month

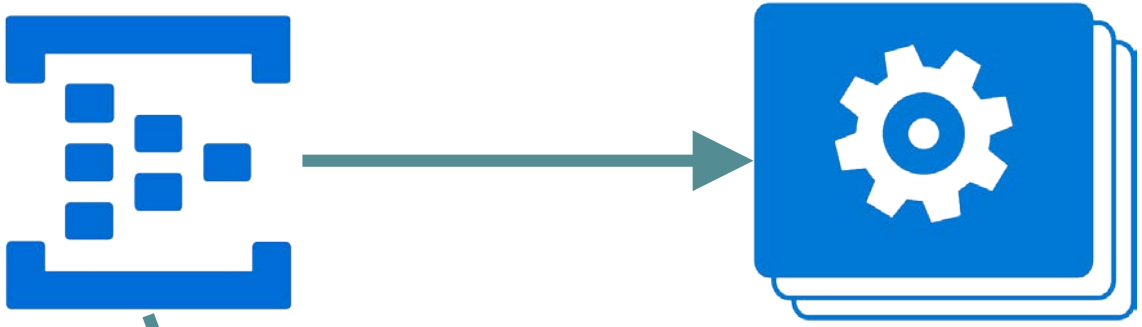


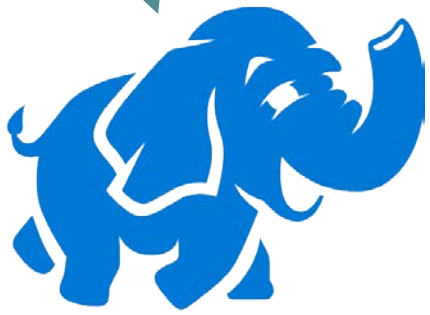




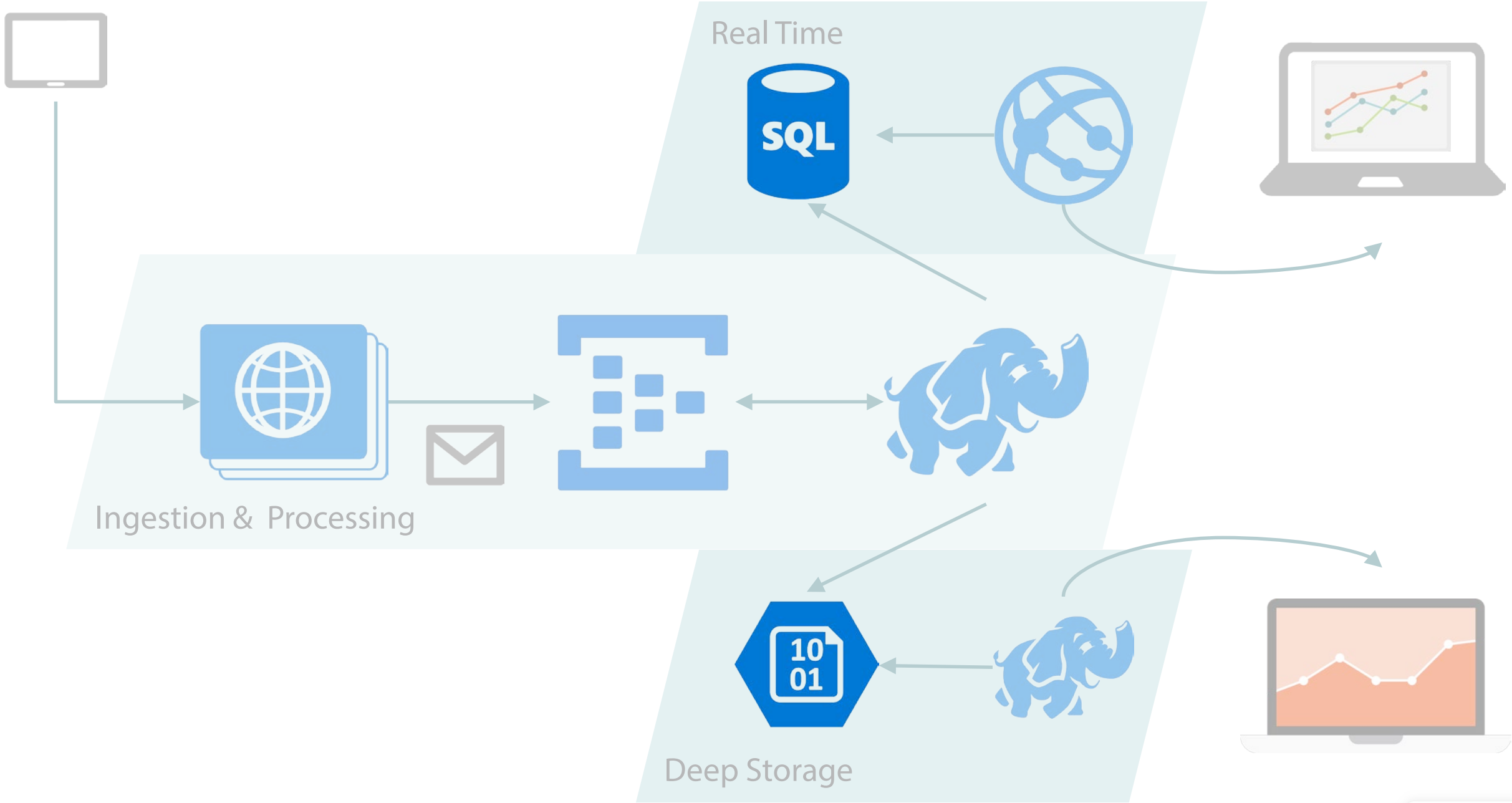


- Technical limitations
- Tried and trusted
- New on HDInsight
- Fast, solid platform





- Faster, better delivery
- Generic skills
- Running costs
- Tied to Azure

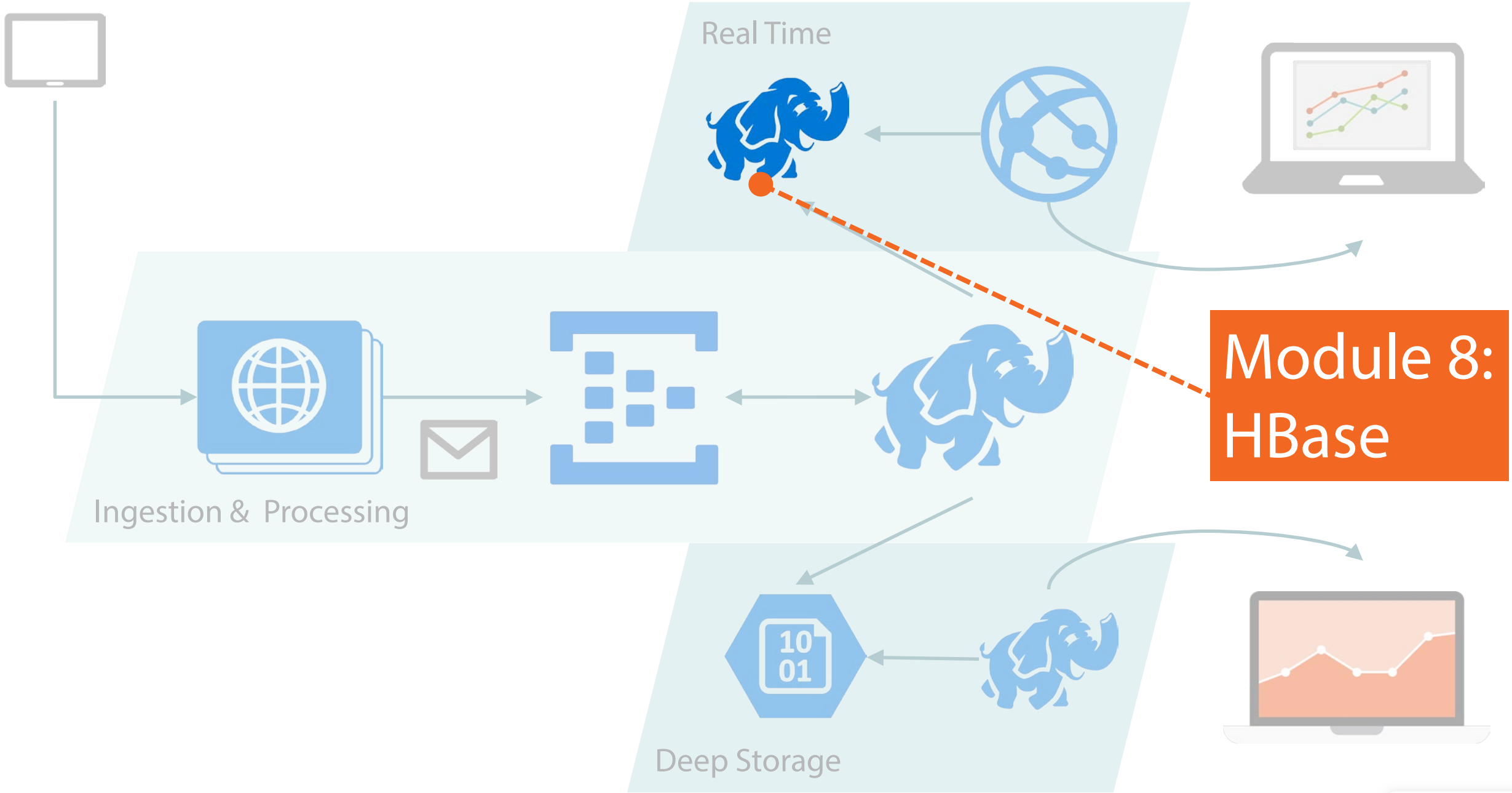

Real Time
SQL
Ingestion & Processing
10
01
Deep Storage

Big data scale

NoSQL database

Massive storage

Fast retrieval


Real Time
Ingestion & Processing
Deep Storage
10
01
Module 8:
HBase