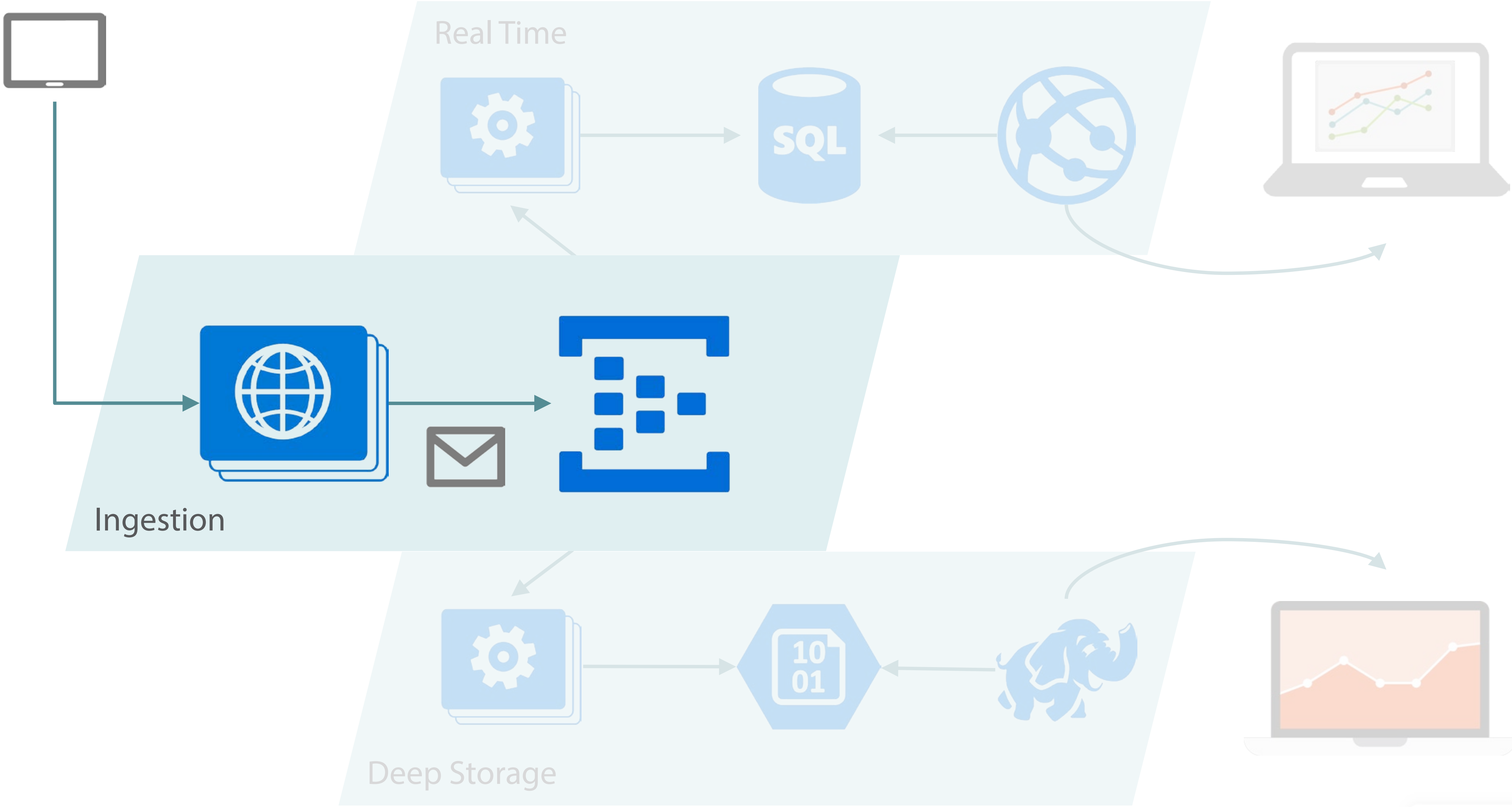
Normalizing Event Data for Real-Time Queries

Elton Stoneman

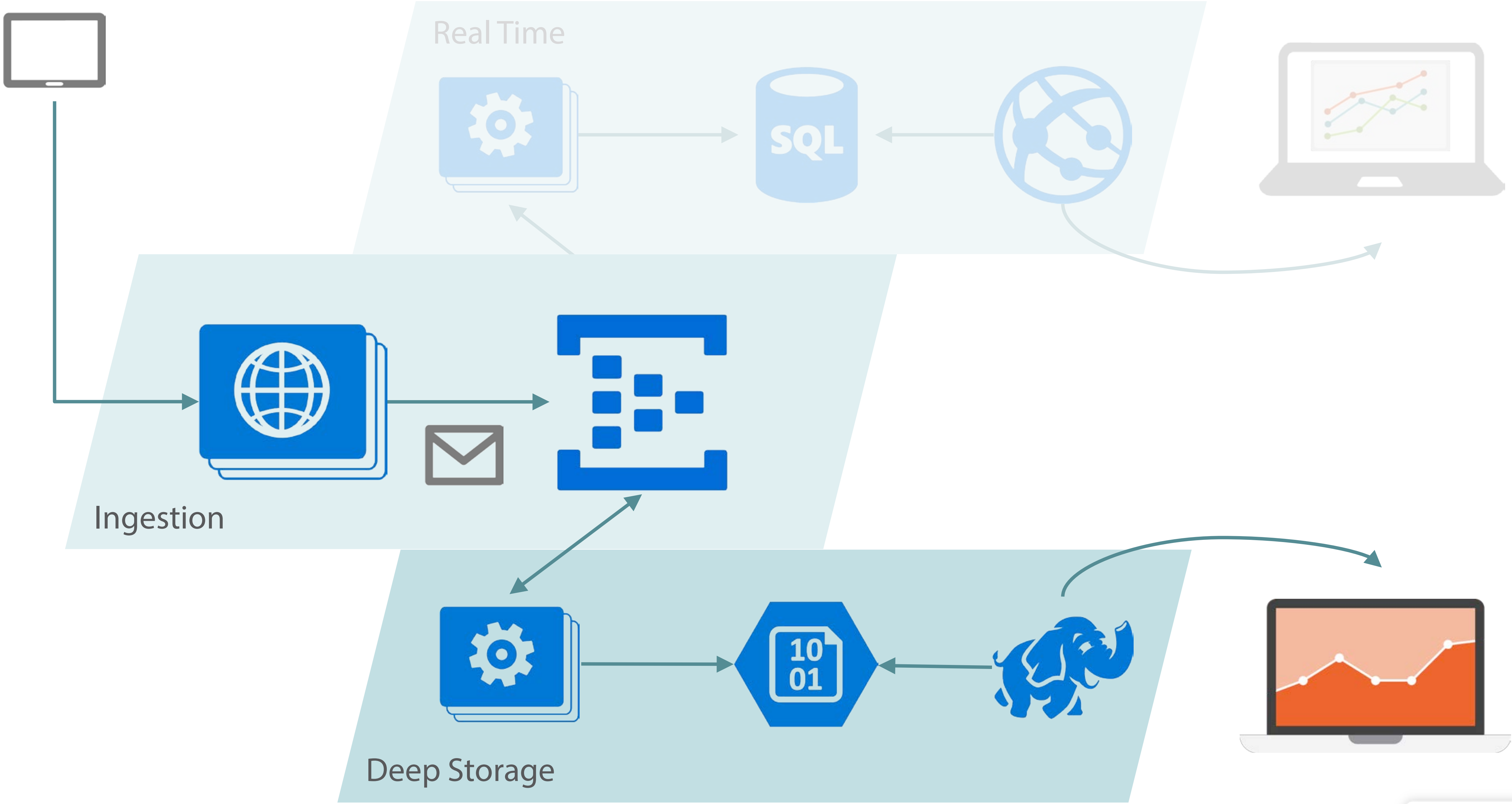
@EltonStoneman | www.geekswithblogs.net/eltonstoneman

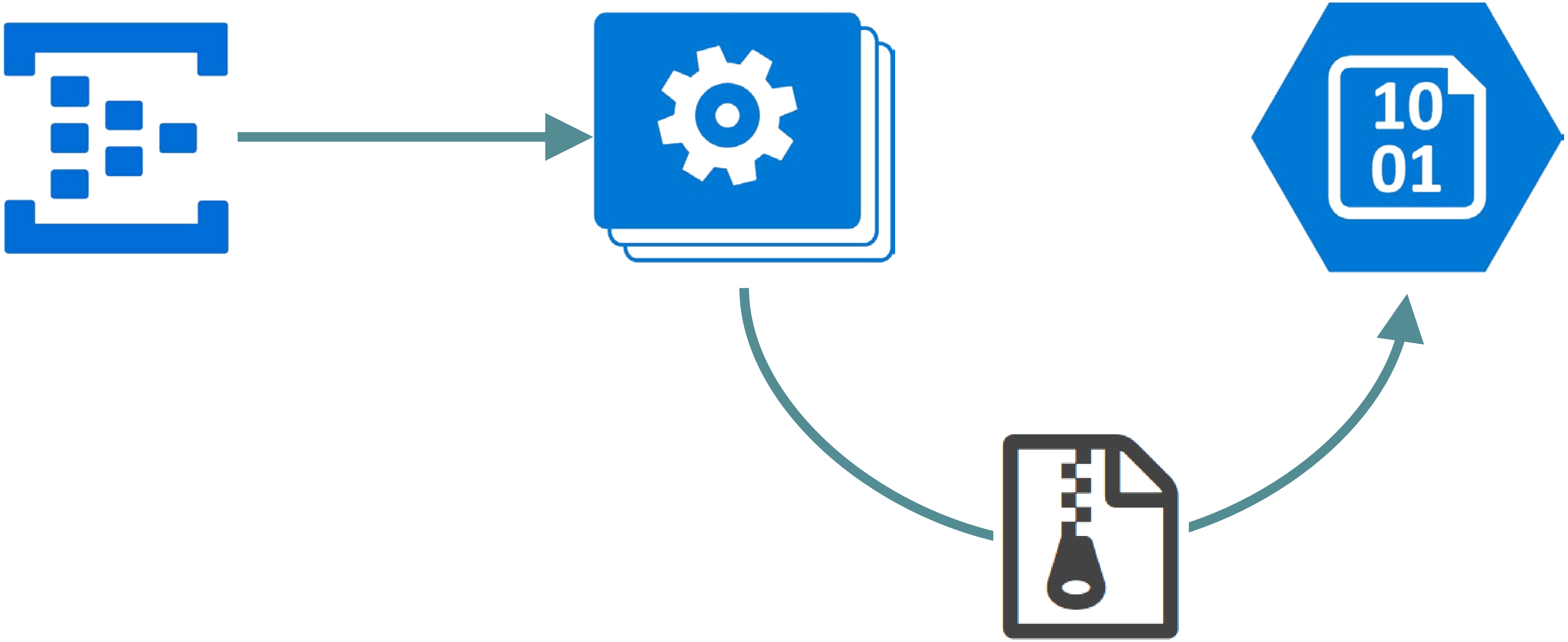


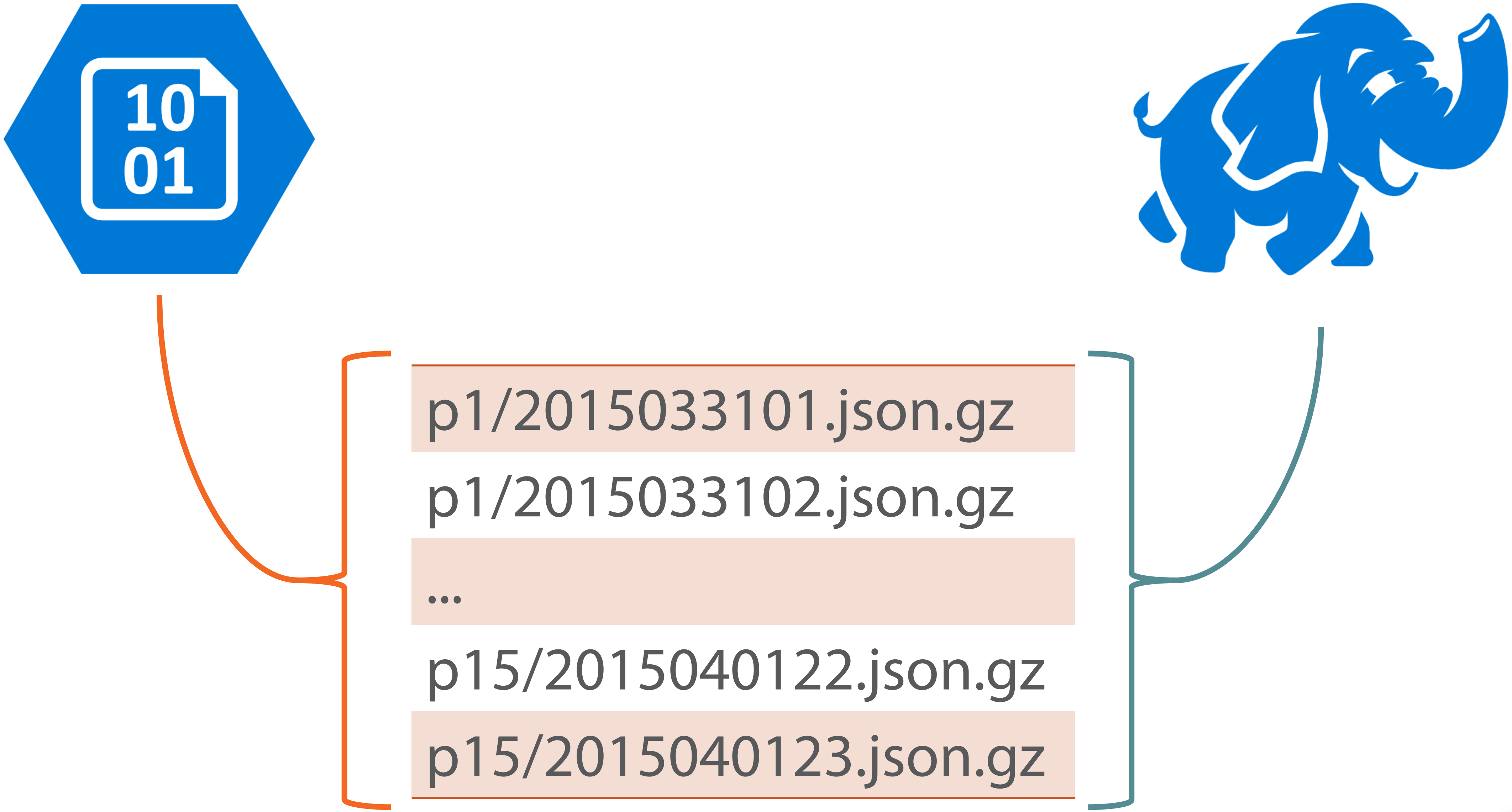
Real Time

Ingestion

Deep Storage


Real Time
SQL
Ingestion
10
01
Deep Storage


10
01



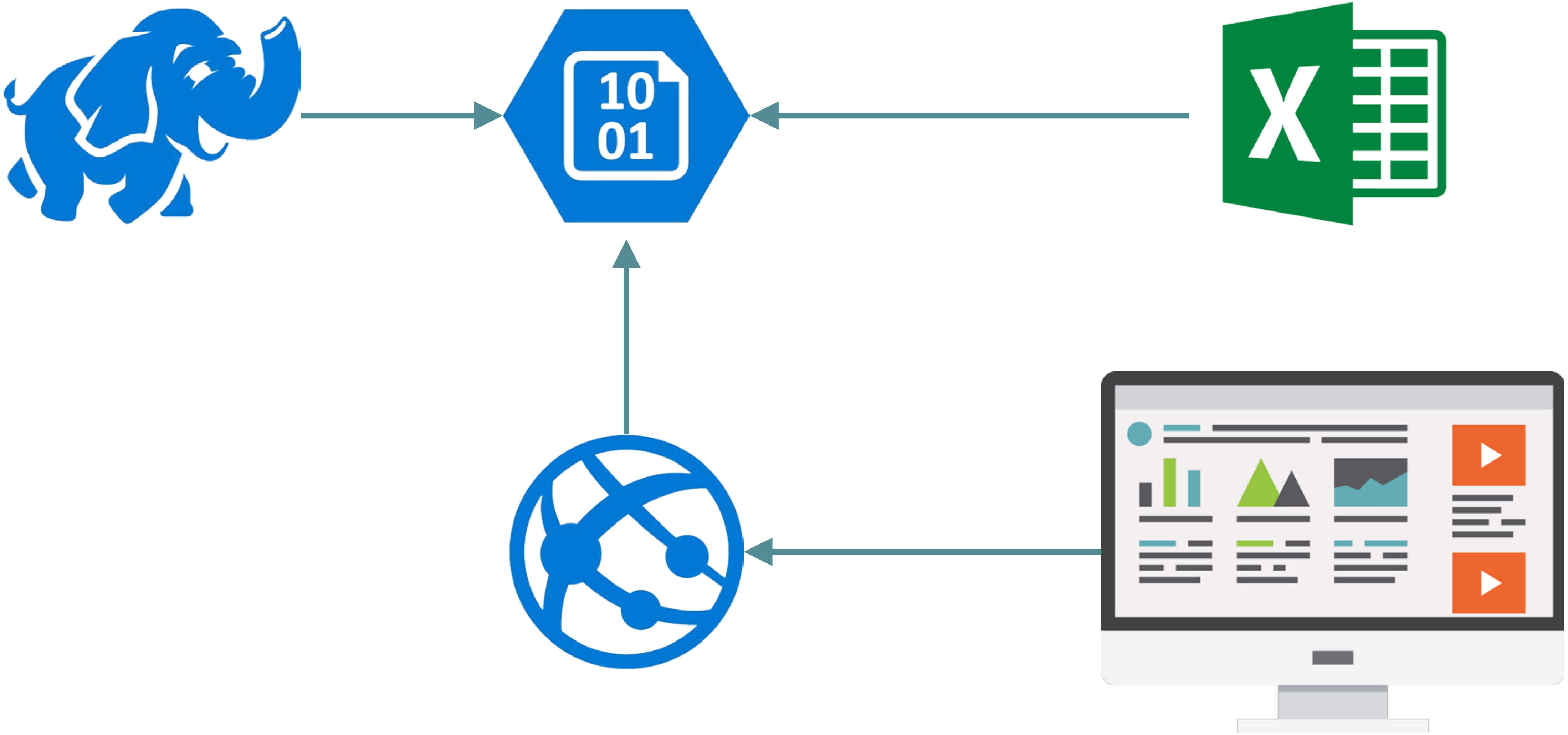
p1/2015033101.json.gz

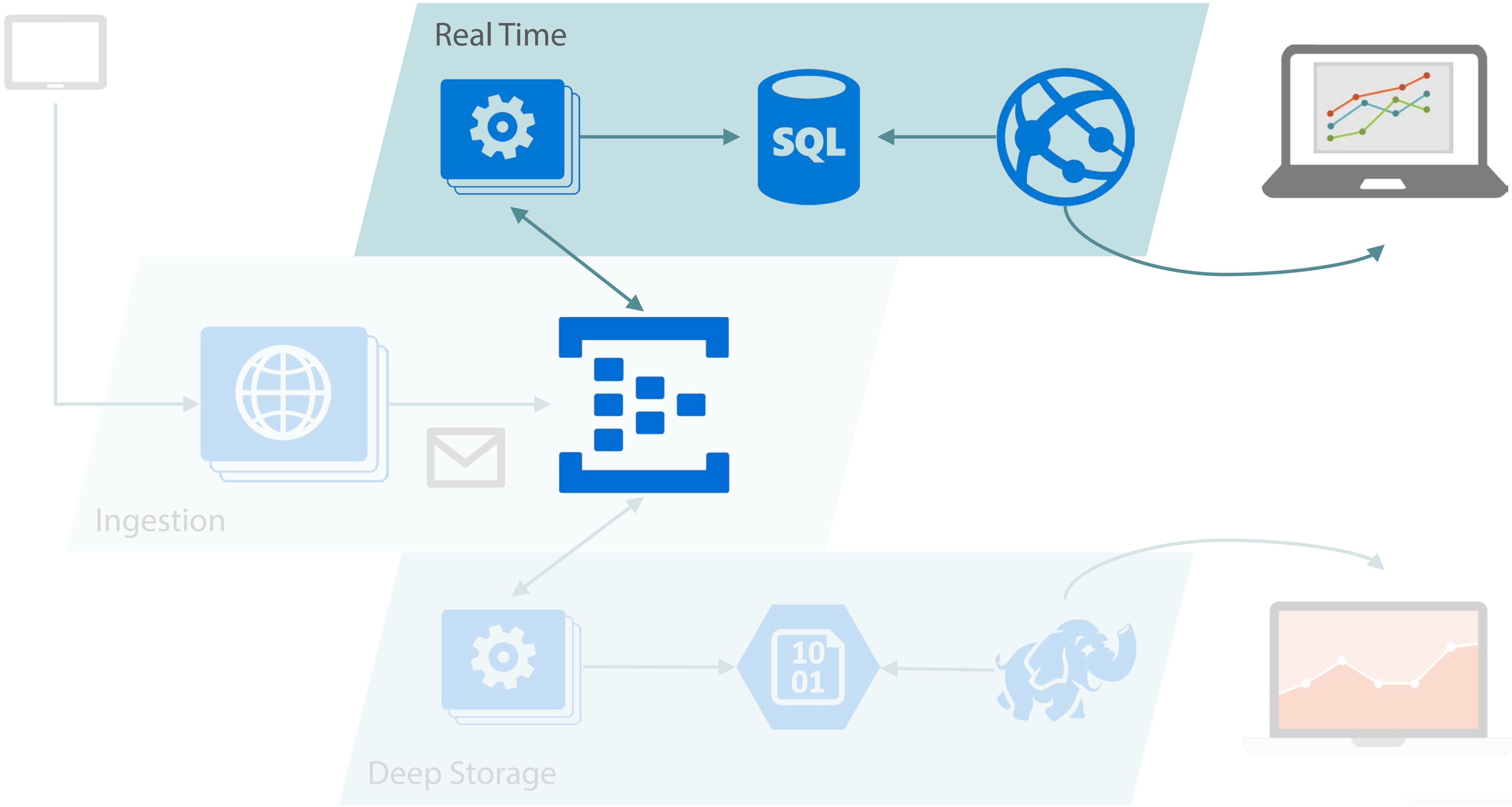
p1/2015033102.json.gz

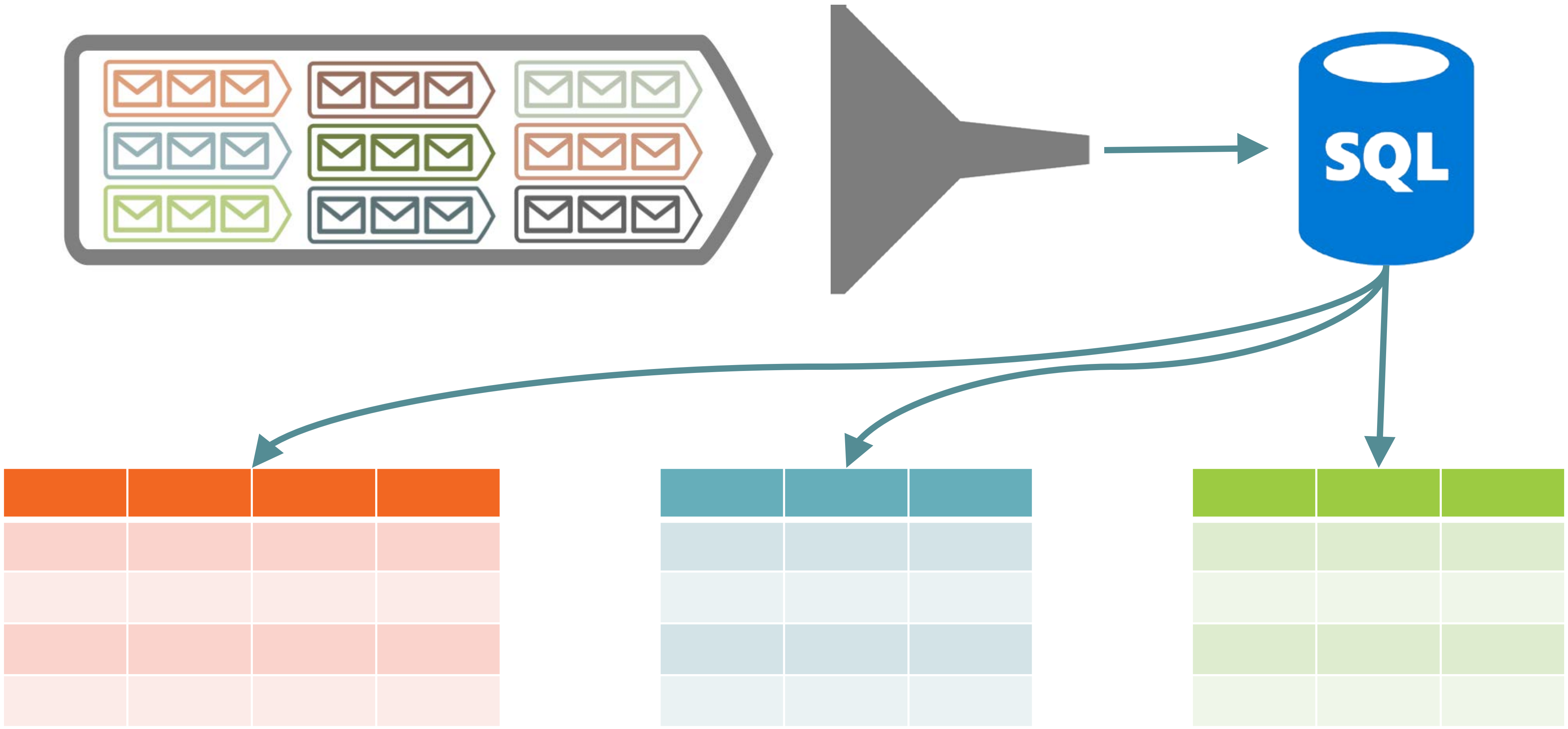
...

p15/2015040122.json.gz

p15/2015040123.json.gz


10
01

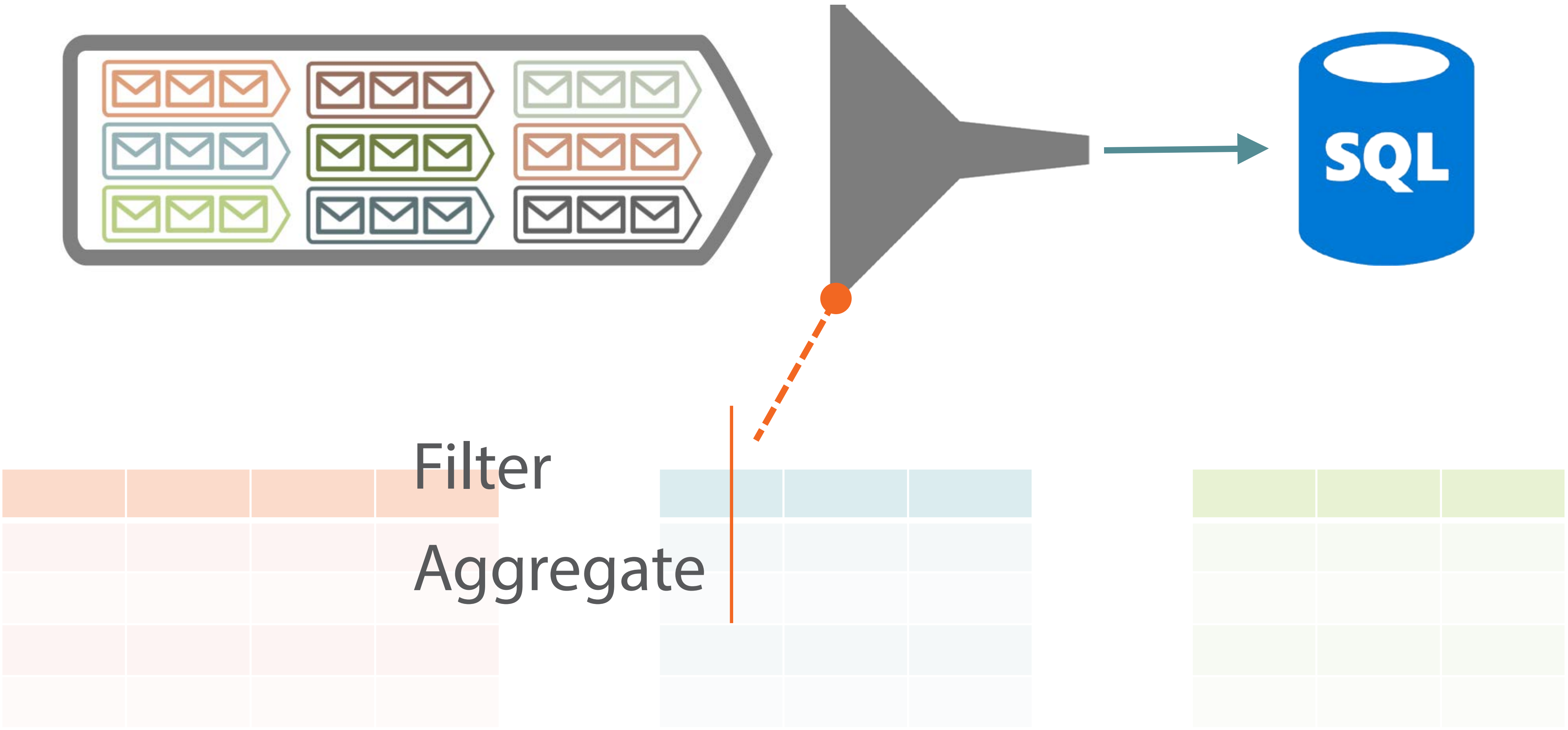

Real Time
SQL
Ingestion
10
01
Deep Storage



SUM...

COUNT...

DISTINCT...



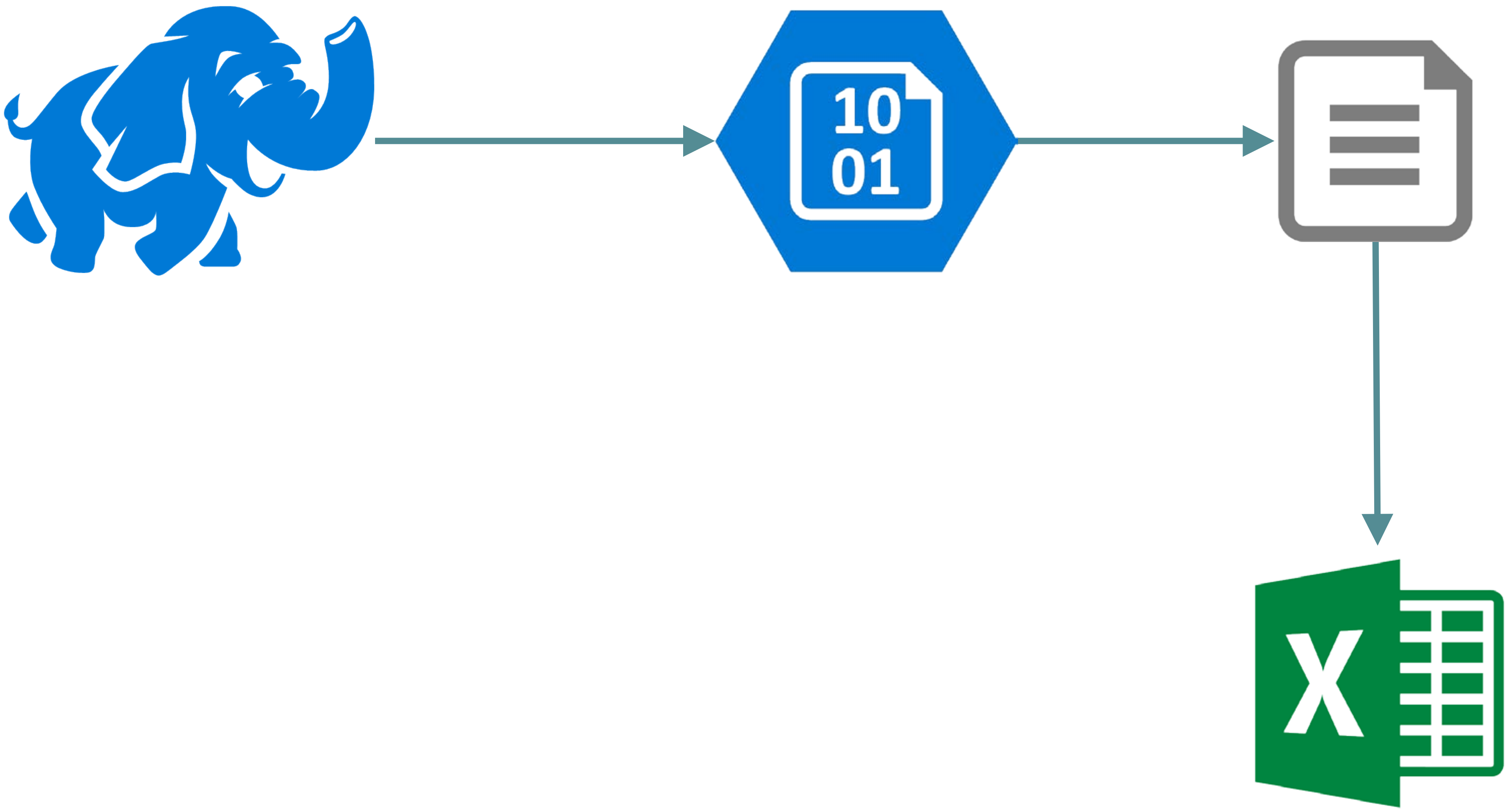
Filter

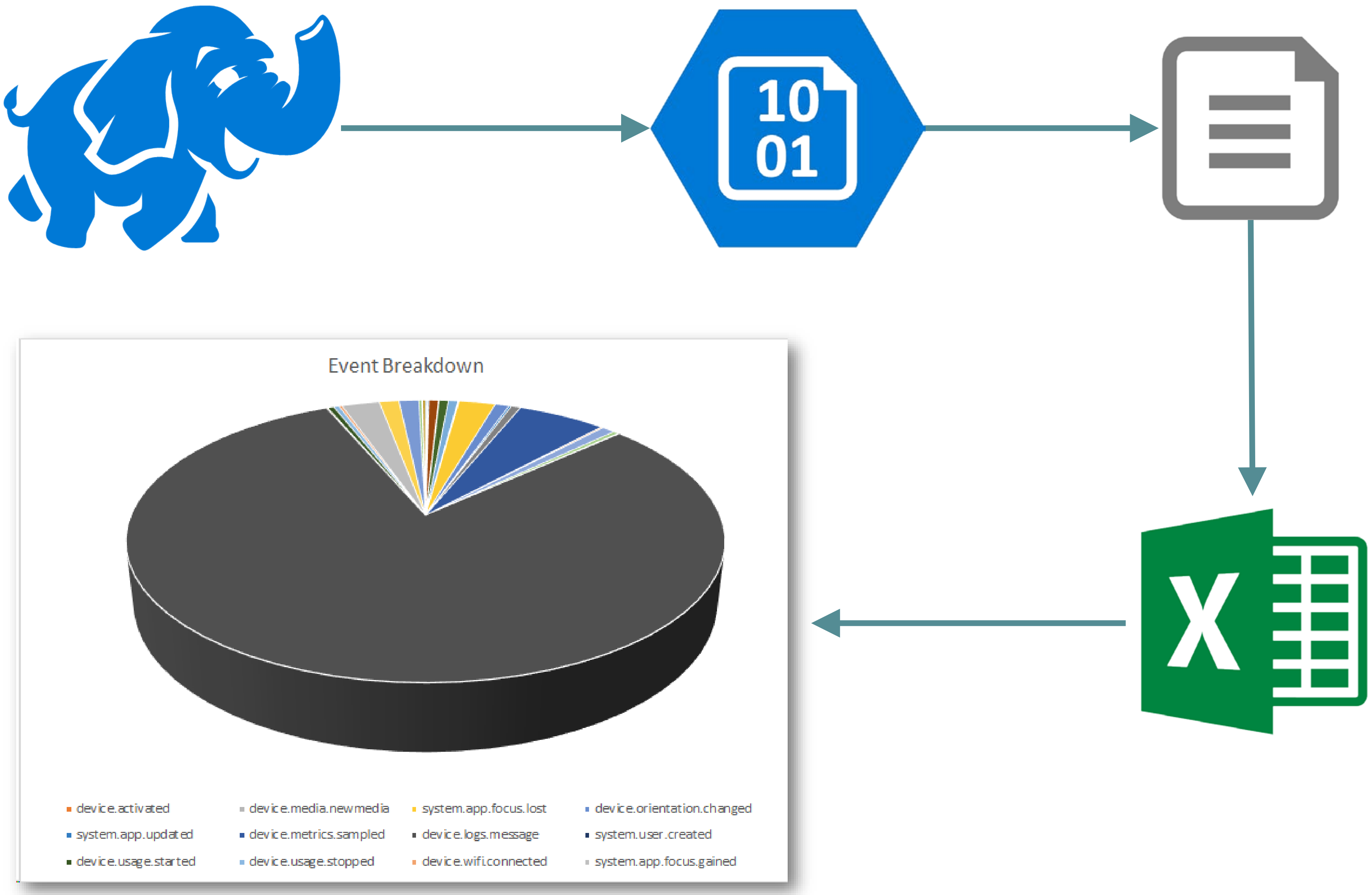
Aggregate

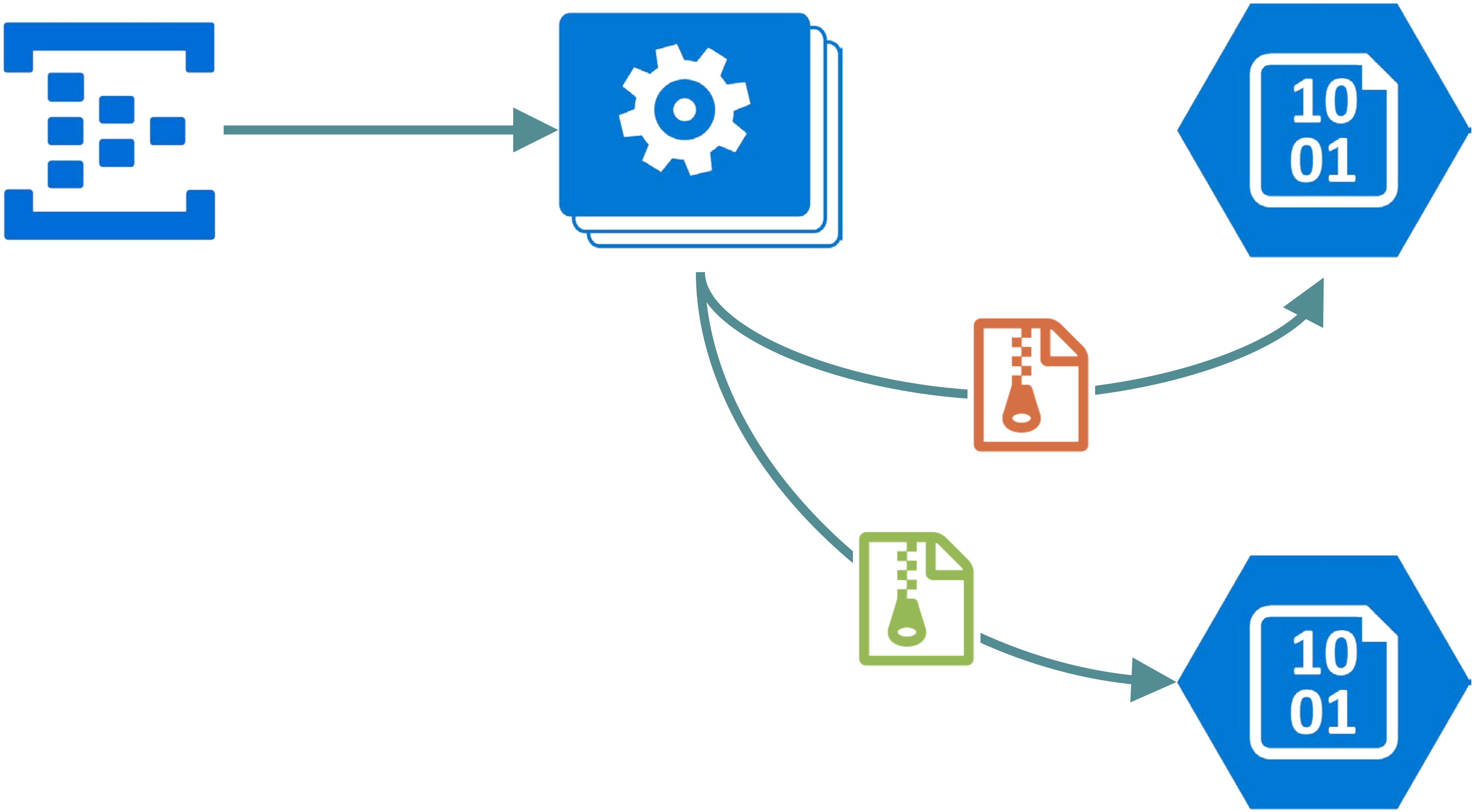
SUM...

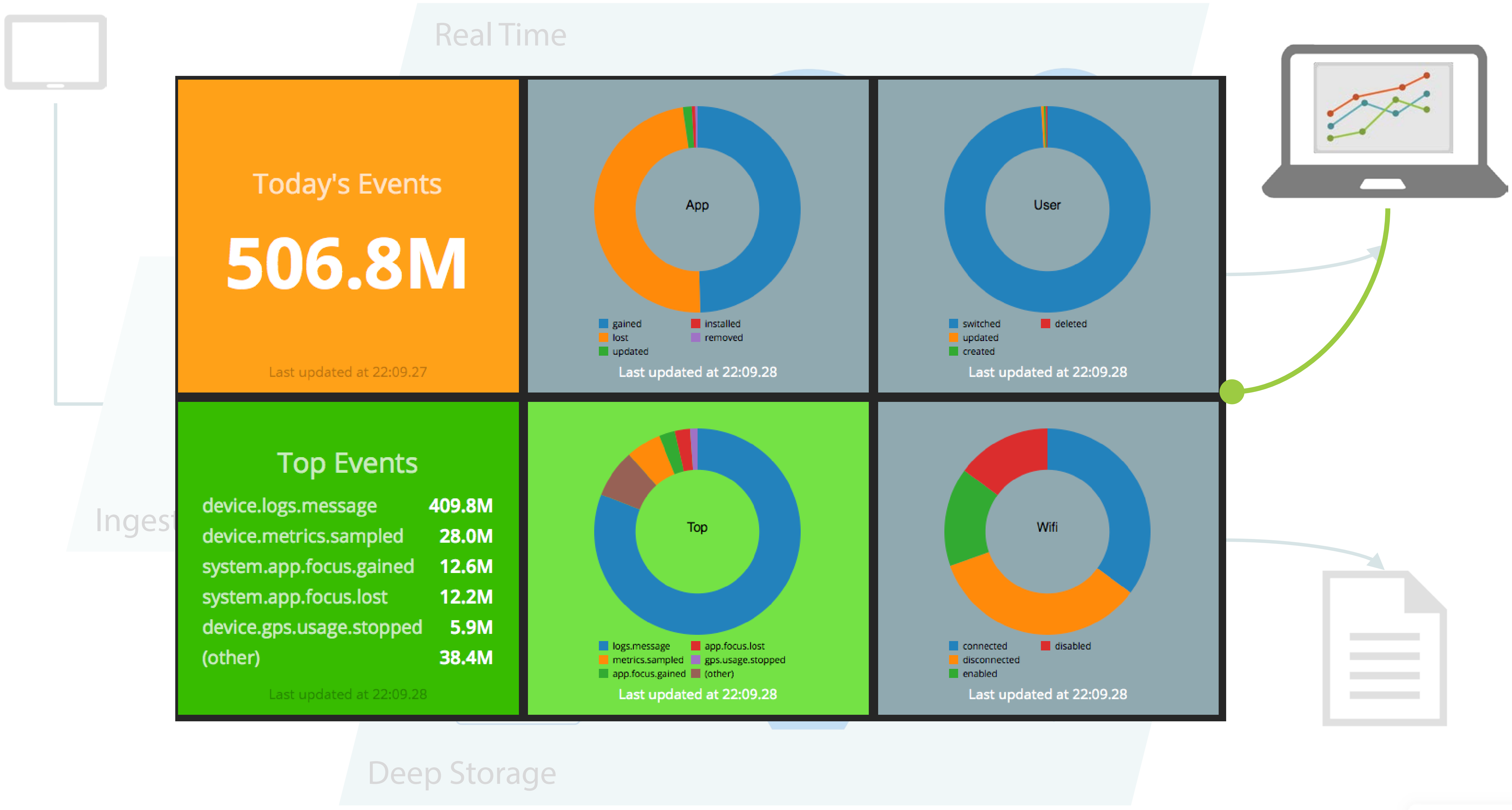
COUNT...

DISTINCT...




Event Breakdown
device.activated
device.media.newmedia
system.app.focus.lost
device.orientation.changed
system.app.updated
device.metrics.sampled
device.logs.message
system.user.created
device.usage.started
device.usage.stopped
device.wifi.connected
system.app.focus.gained


10
01
10
01



Today's Events

506.8M

Last updated at 22:09.27

Last updated at 22:09.28

Last updated at 22:09.28

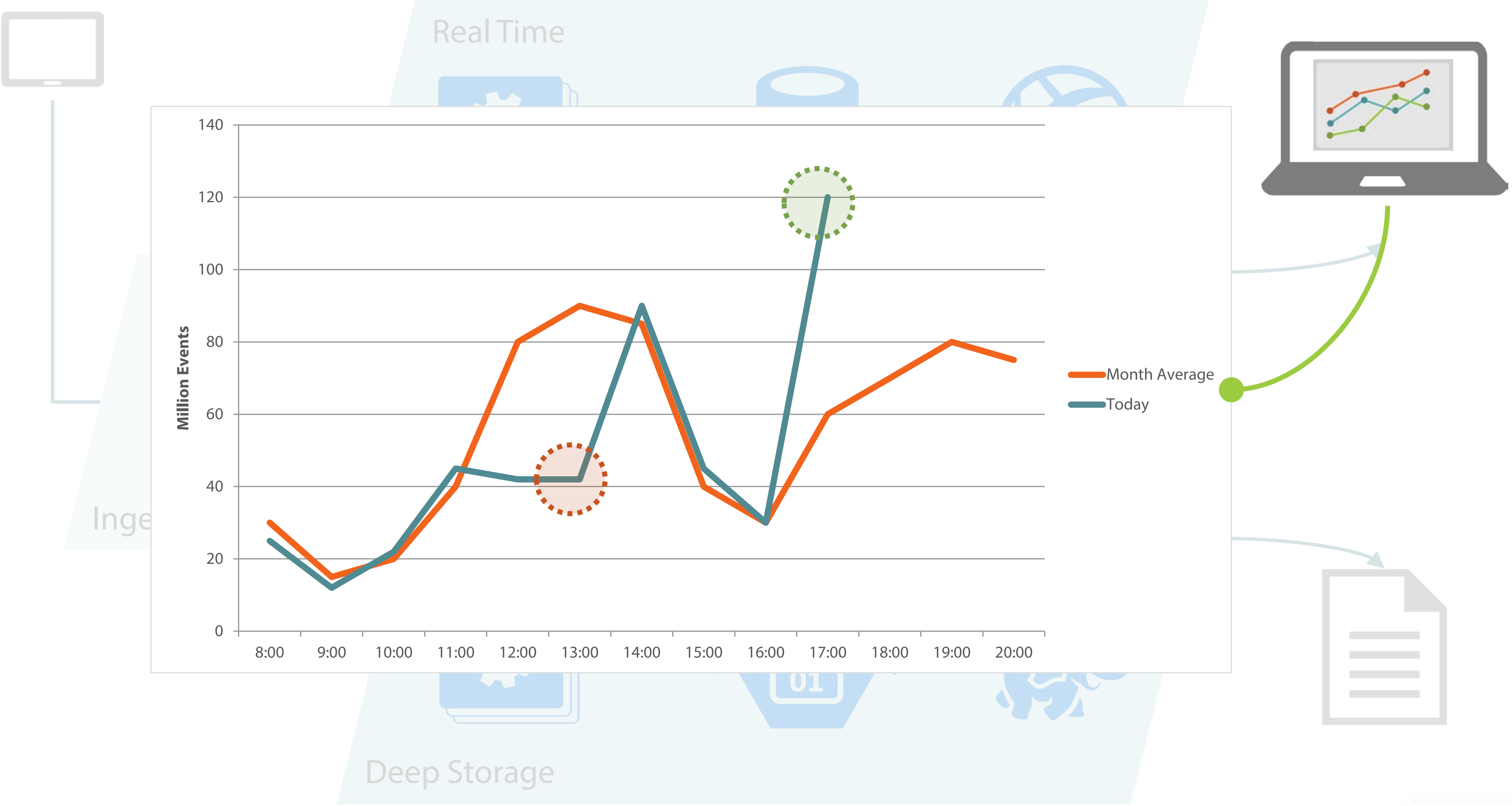
Top Events

device.logs.message	409.8M
device.metrics.sampled	28.0M
system.app.focus.gained	12.6M
system.app.focus.lost	12.2M
device.gps.usage.stopped	5.9M
(other)	38.4M

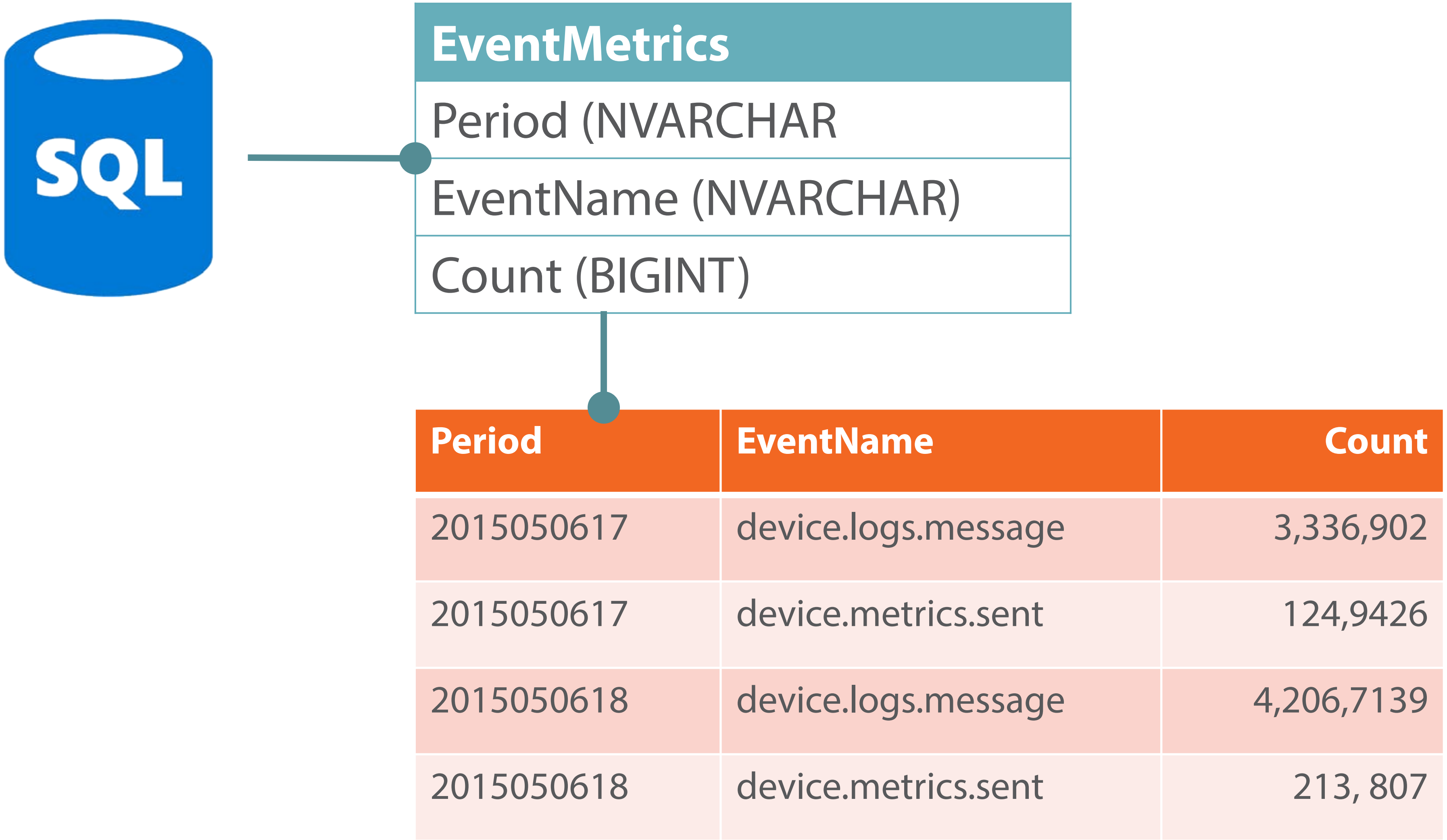
Last updated at 22:09.28

Last updated at 22:09.28

Last updated at 22:09.28

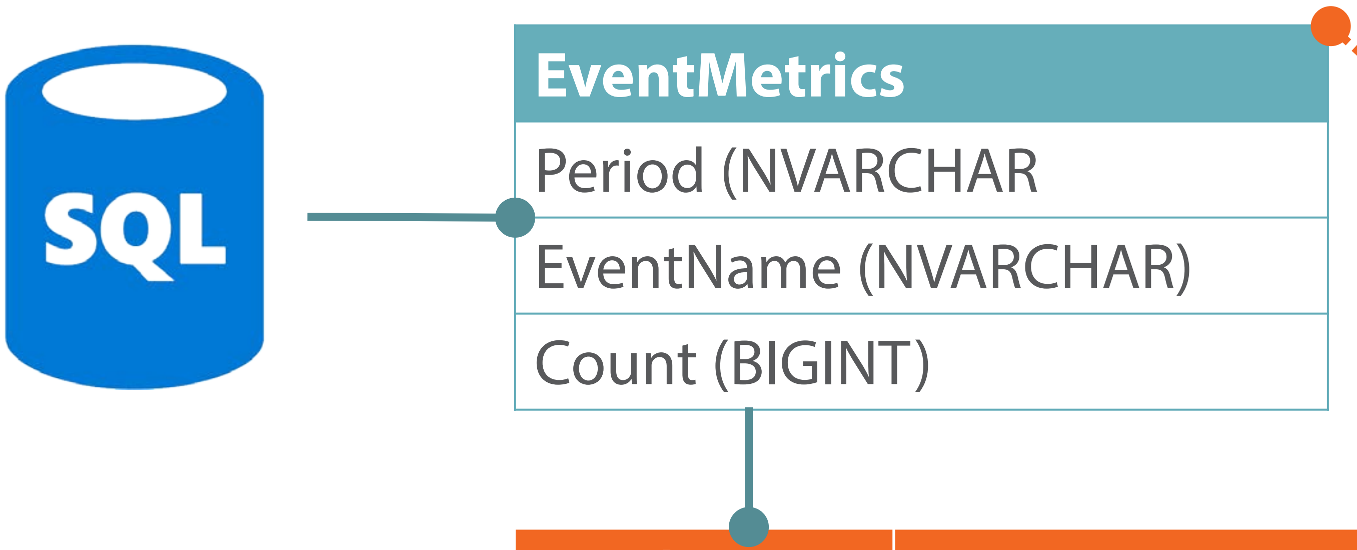


Real Time
Inge
Deep Storage
01
Million Events
140
120
100
80
60
40
20
0
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
Month Average
Today



EventMetrics
Period (NVARCHAR
EventName (NVARCHAR)
Count (BIGINT)

Period	EventName	Count
2015050617	device.logs.message	3,336,902
2015050617	device.metrics.sent	124,9426
2015050618	device.logs.message	4,206,7139
2015050618	device.metrics.sent	213, 807



EventMetrics

Period (NVARCHAR

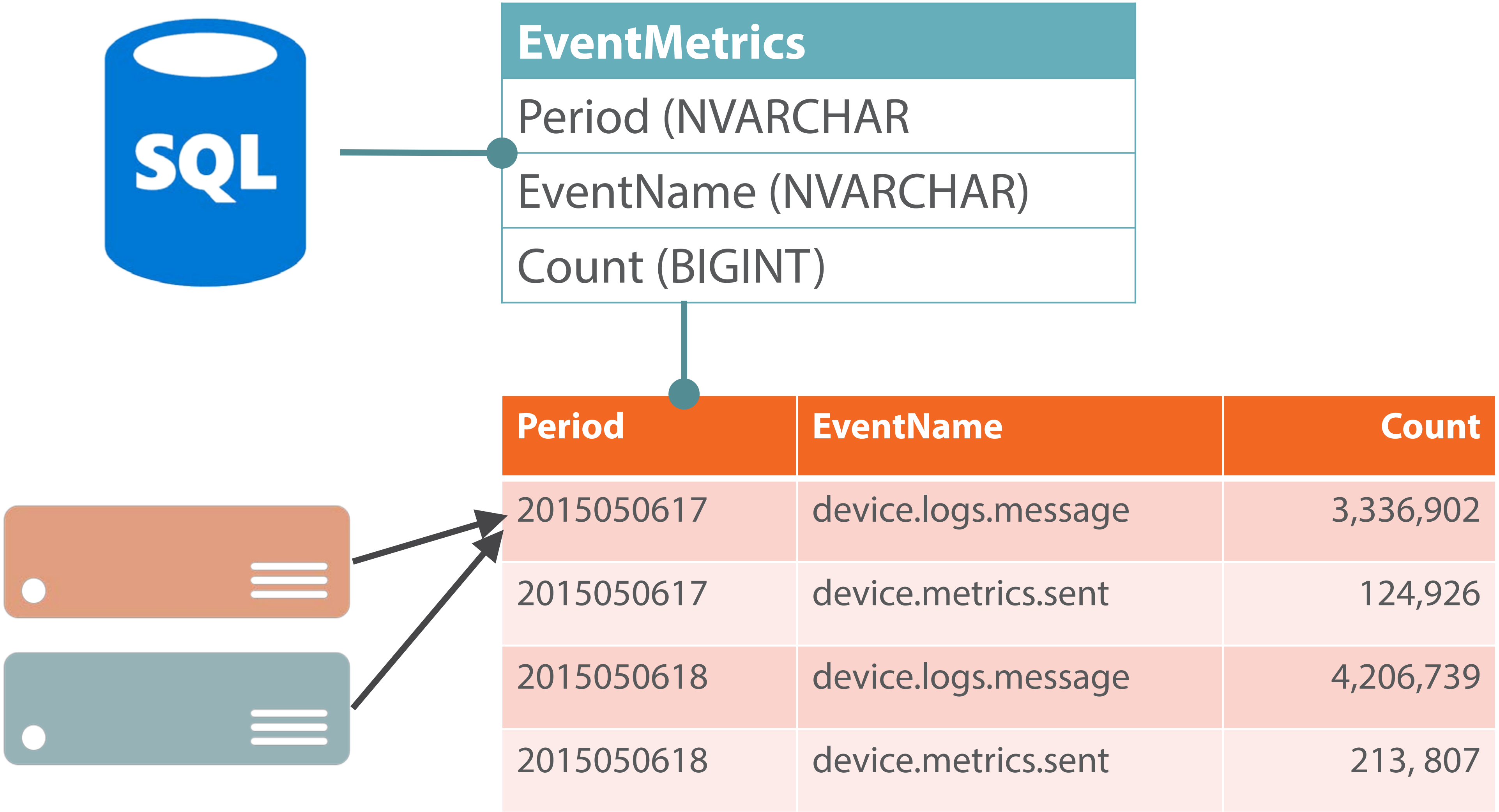
EventName (NVARCHAR)

Count (BIGINT)

@40 event types

960 rows /day

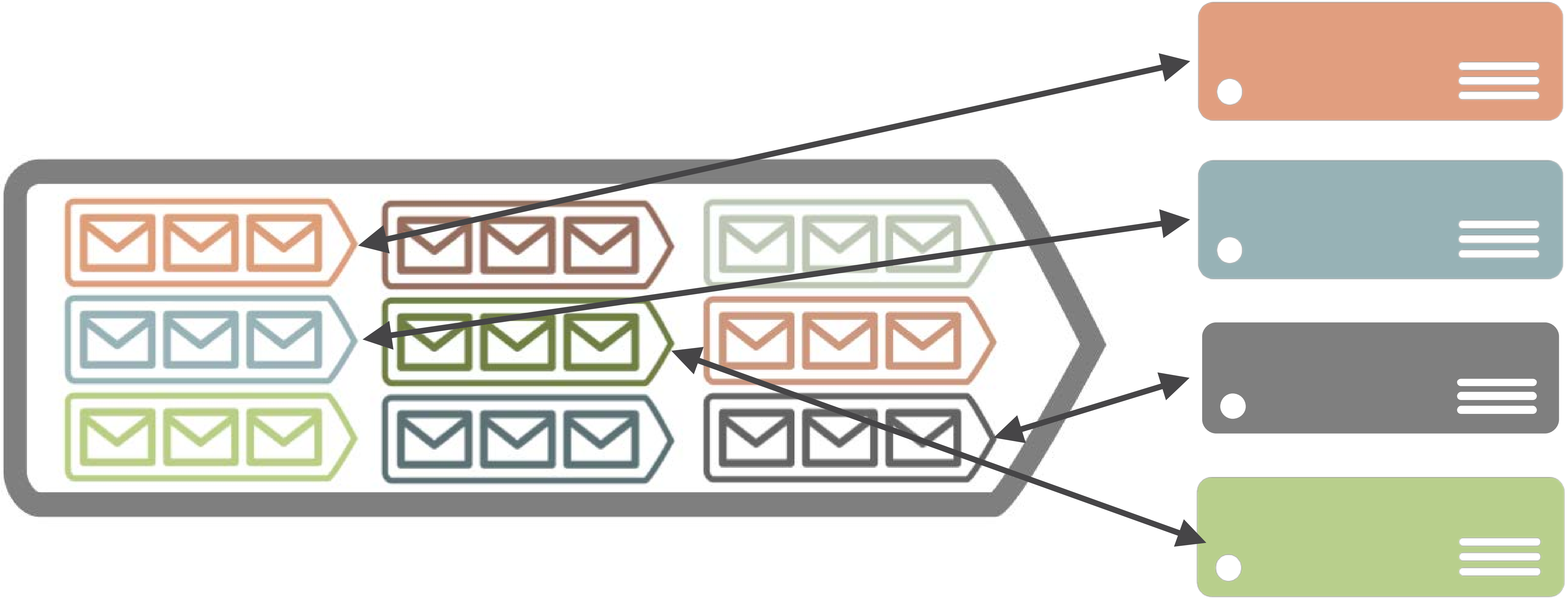
Period	EventName	Count
2015050617	device.logs.message	3,336,902
2015050617	device.metrics.sent	124,9426
2015050618	device.logs.message	4,206,7139
2015050618	device.metrics.sent	213, 807

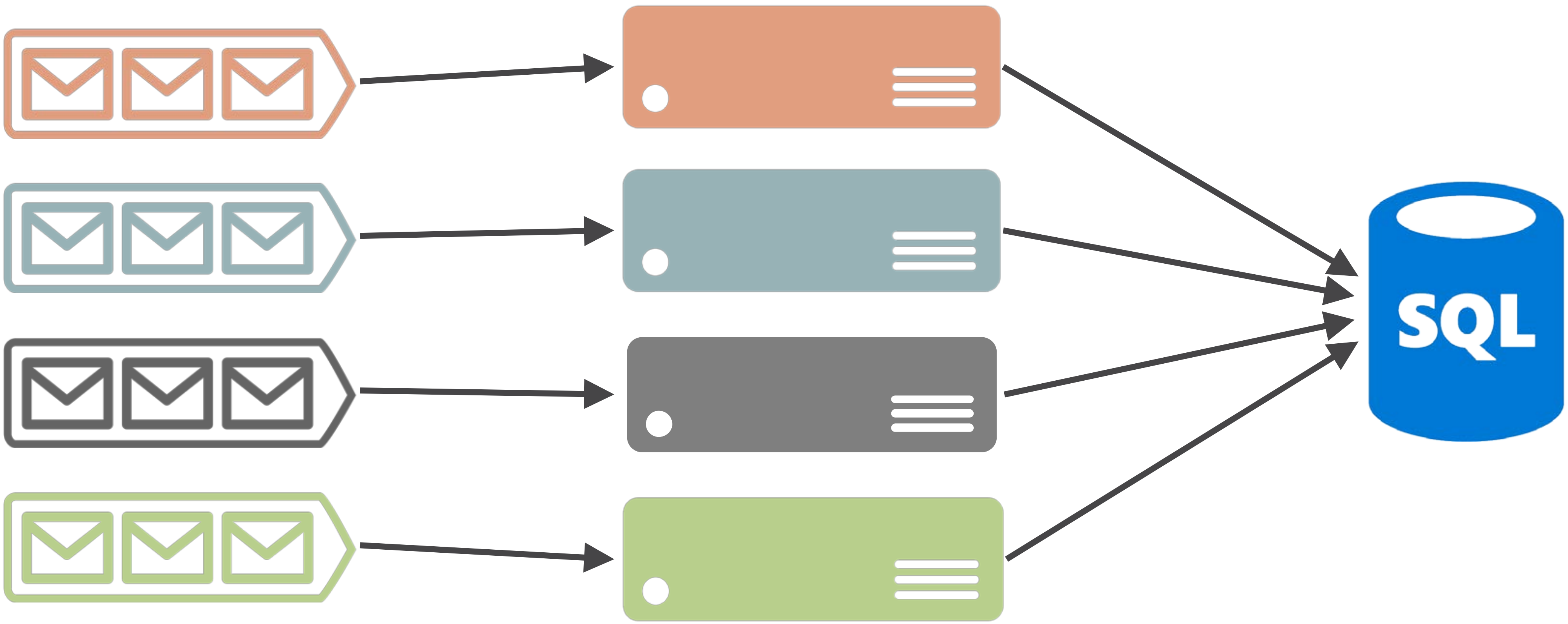


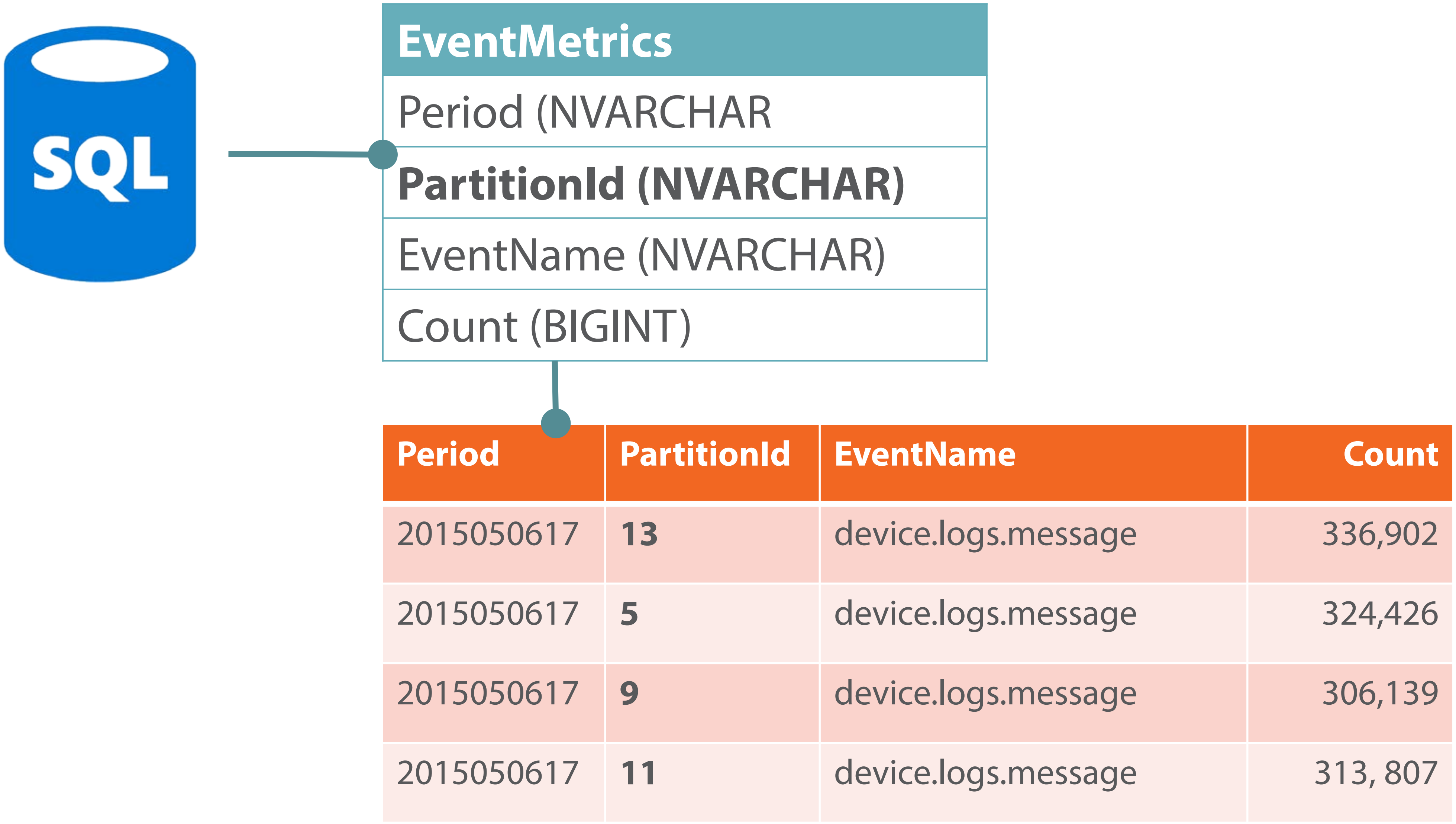
EventMetrics

- Period (NVARCHAR
- EventName (NVARCHAR)
- Count (BIGINT)

Period	EventName	Count
2015050617	device.logs.message	3,336,902
2015050617	device.metrics.sent	124,926
2015050618	device.logs.message	4,206,739
2015050618	device.metrics.sent	213, 807




SQL



EventMetrics

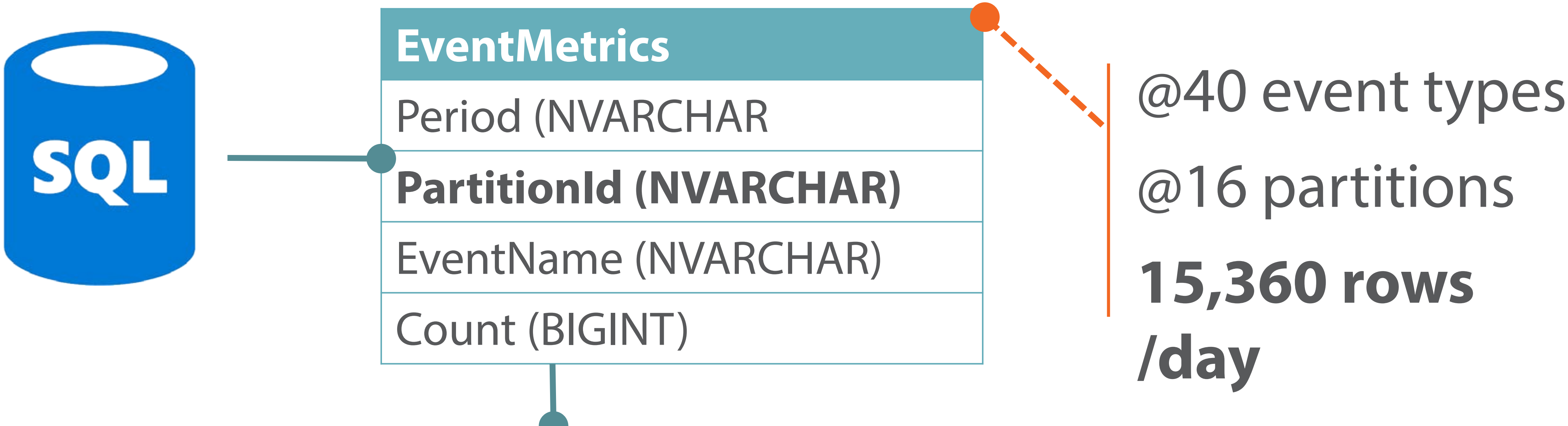
Period (NVARCHAR

PartitionId (NVARCHAR)

EventName (NVARCHAR)

Count (BIGINT)

Period	PartitionId	EventName	Count
2015050617	**13**	device.logs.message	336,902
2015050617	**5**	device.logs.message	324,426
2015050617	**9**	device.logs.message	306,139
2015050617	**11**	device.logs.message	313, 807



EventMetrics

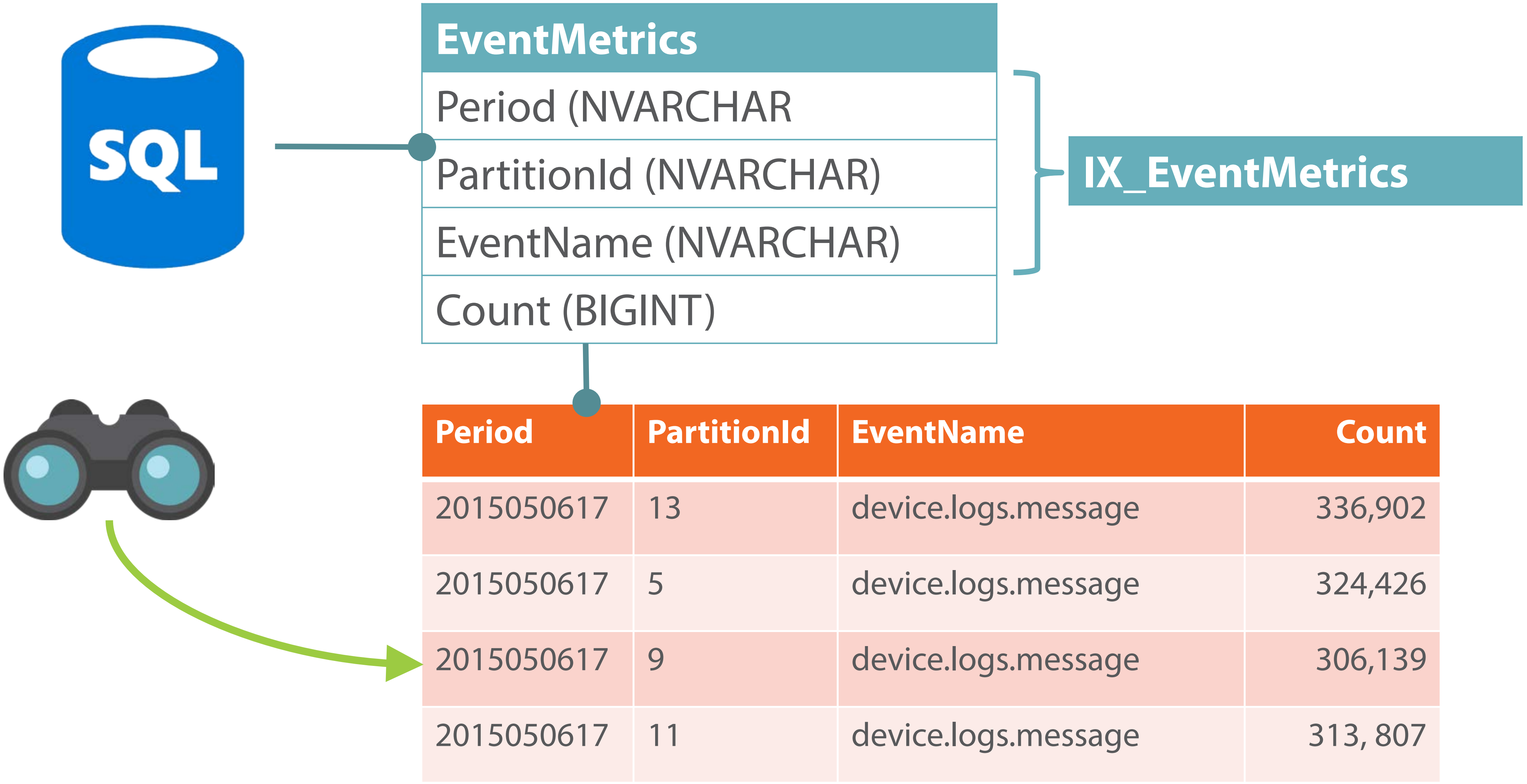
- Period (NVARCHAR
- **PartitionId (NVARCHAR)**
- EventName (NVARCHAR)
- Count (BIGINT)

@40 event types

@16 partitions

15,360 rows /day

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	336,902
2015050617	5	device.logs.message	324,426
2015050617	9	device.logs.message	306,139
2015050617	11	device.logs.message	313, 807



EventMetrics

- Period (NVARCHAR
- PartitionId (NVARCHAR)
- EventName (NVARCHAR)
- Count (BIGINT)

IX_EventMetrics

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	336,902
2015050617	5	device.logs.message	324,426
2015050617	9	device.logs.message	306,139
2015050617	11	device.logs.message	313, 807

Demo: Events DB

SQL Database Project

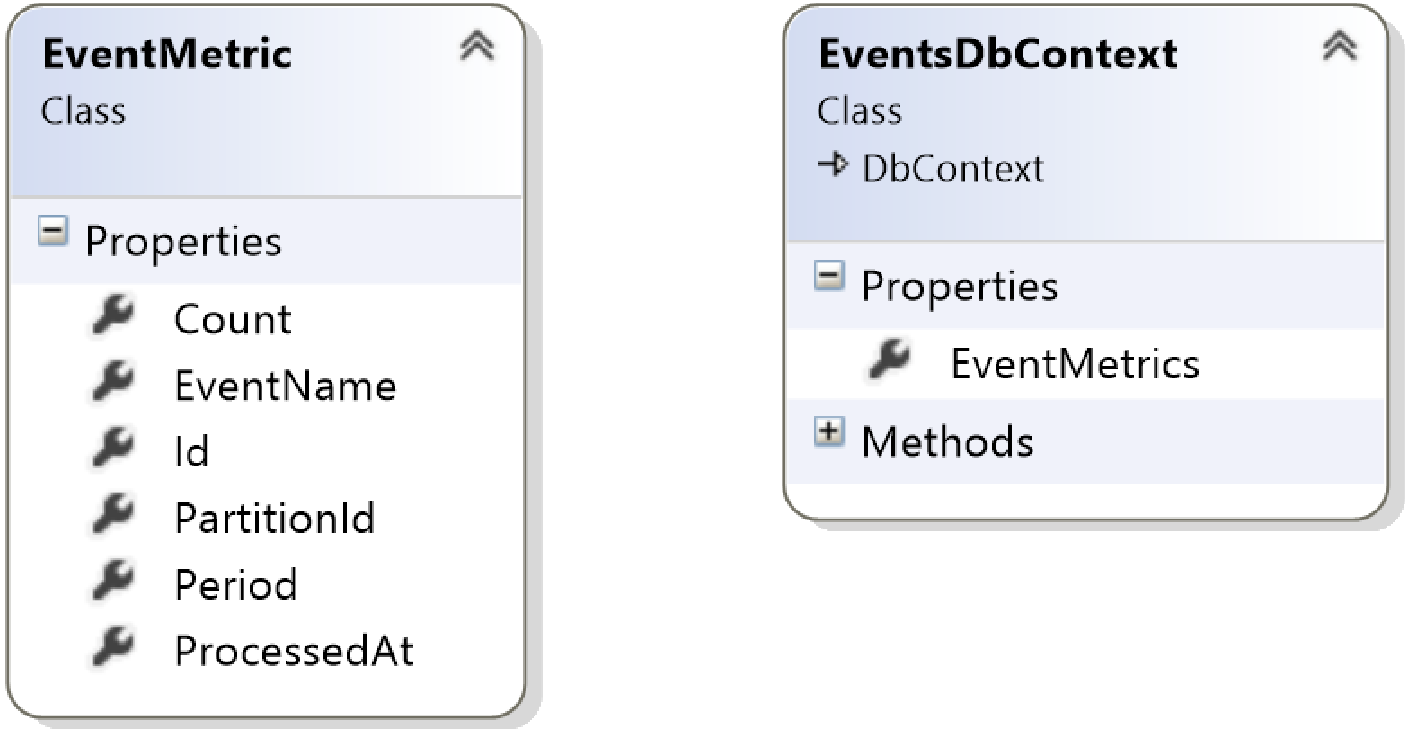
Sample data

Entity Framework model

```
CREATE TABLE [dbo].[EventMetrics] (
[Id] INT PRIMARY KEY IDENTITY,
[EventName] NVARCHAR(40),
[Period] NVARCHAR(10), [PartitionId] NVARCHAR(5),
[Count] BIGINT, [ProcessedAt] DATETIME )
```

EventMetrics

Aggregated event data and processing timestamp



EventMetric
Class

Properties
- Count
- EventName
- Id
- PartitionId
- Period
- ProcessedAt

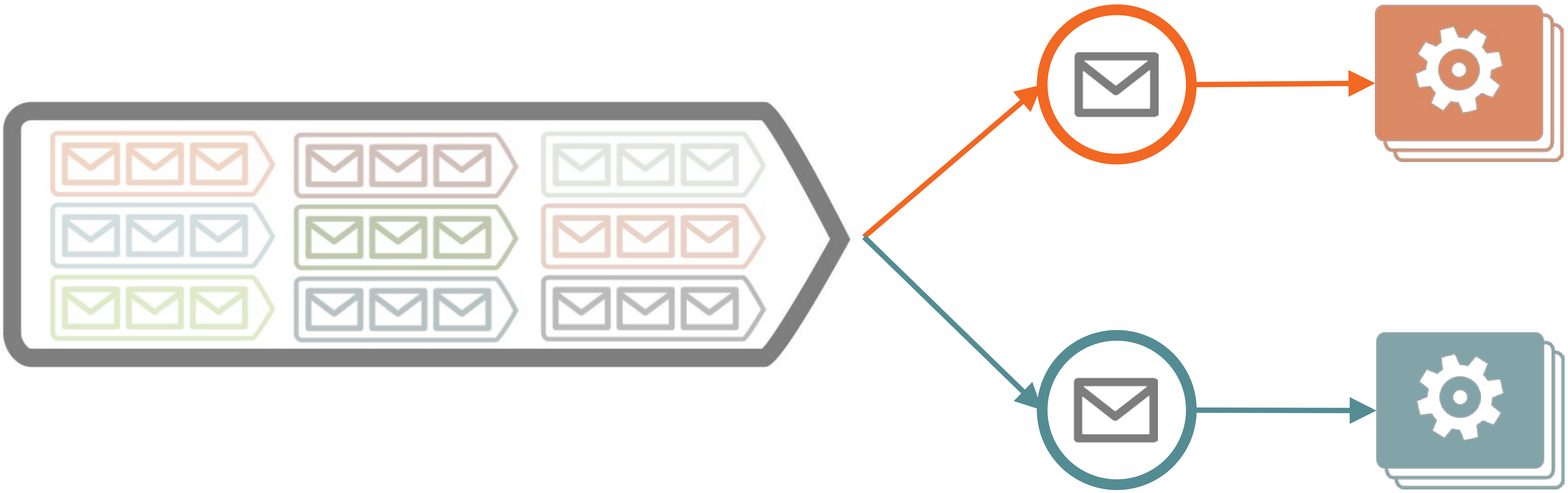
EventsDbContext
Class
→ DbContext

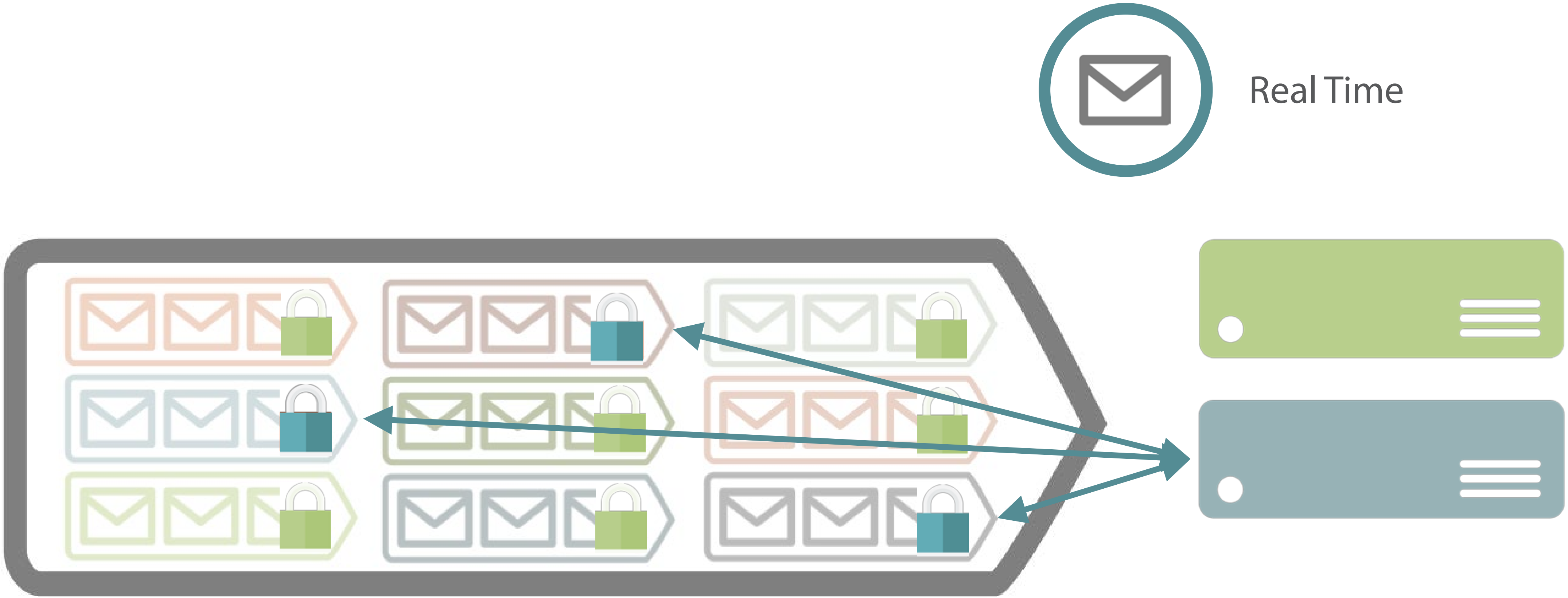
Properties
- EventMetrics

Methods

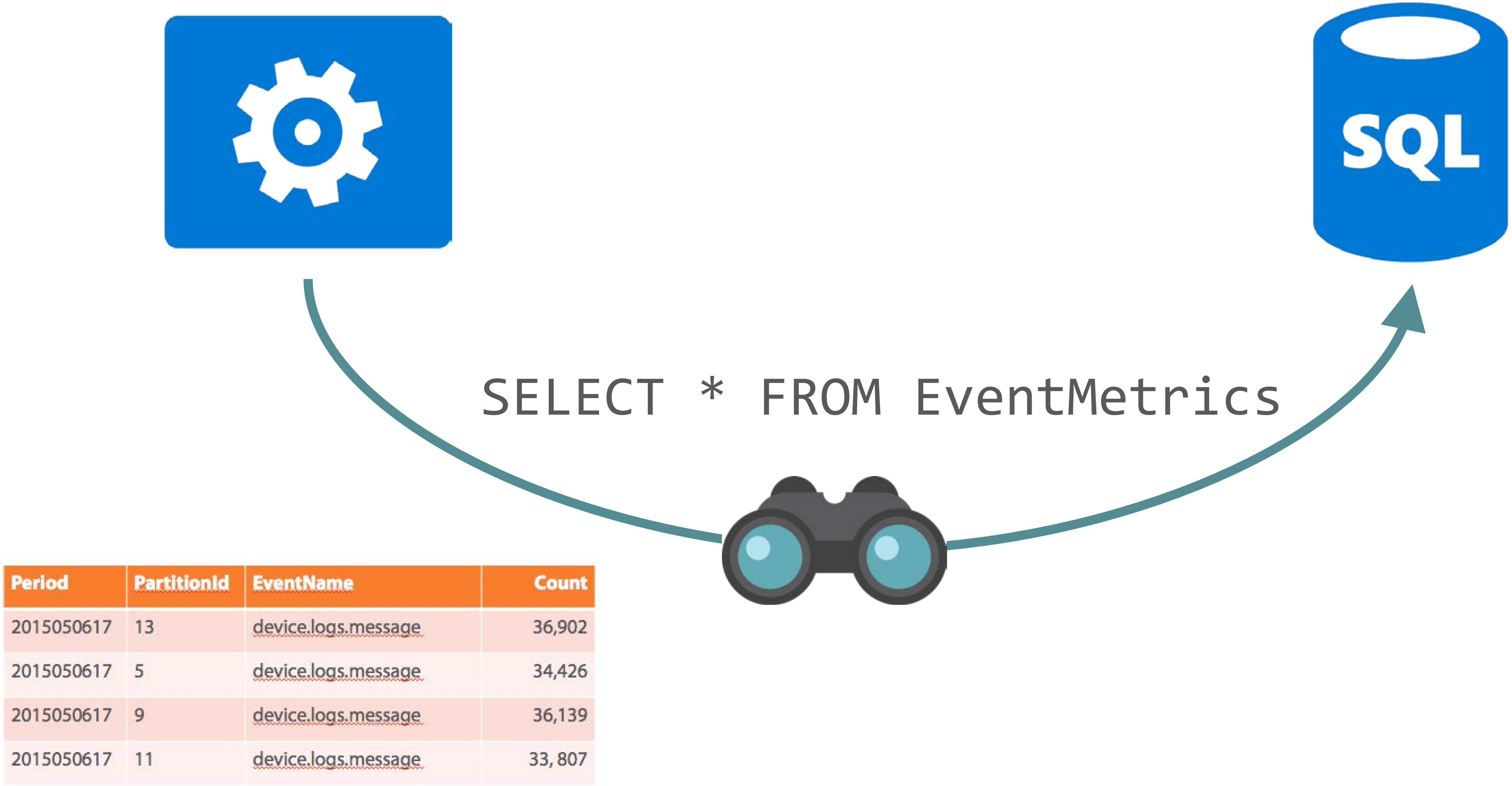
Entity Framework

Model & database context


Deep Storage
Real Time

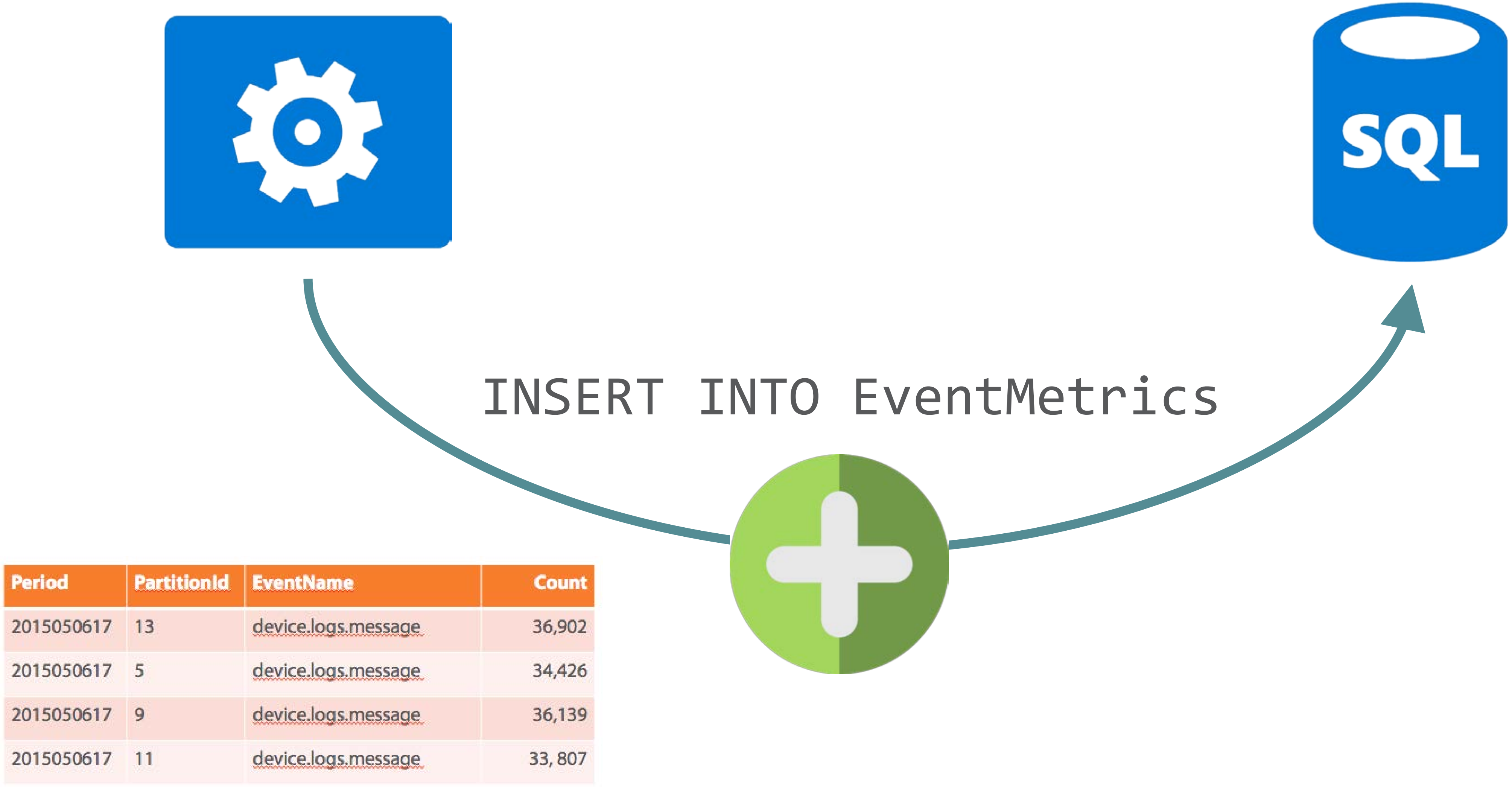

Real Time

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



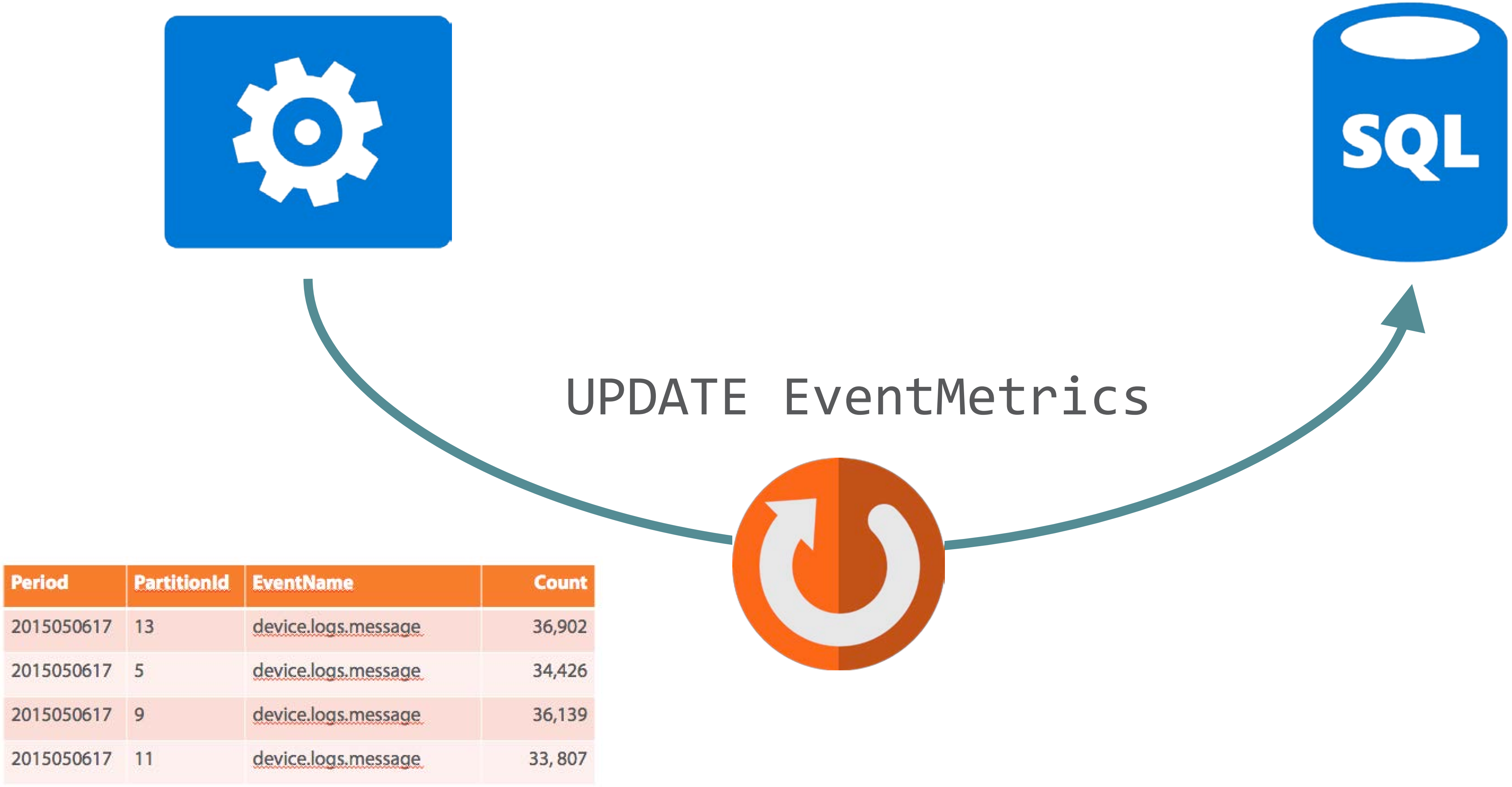
SELECT * FROM EventMetrics

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



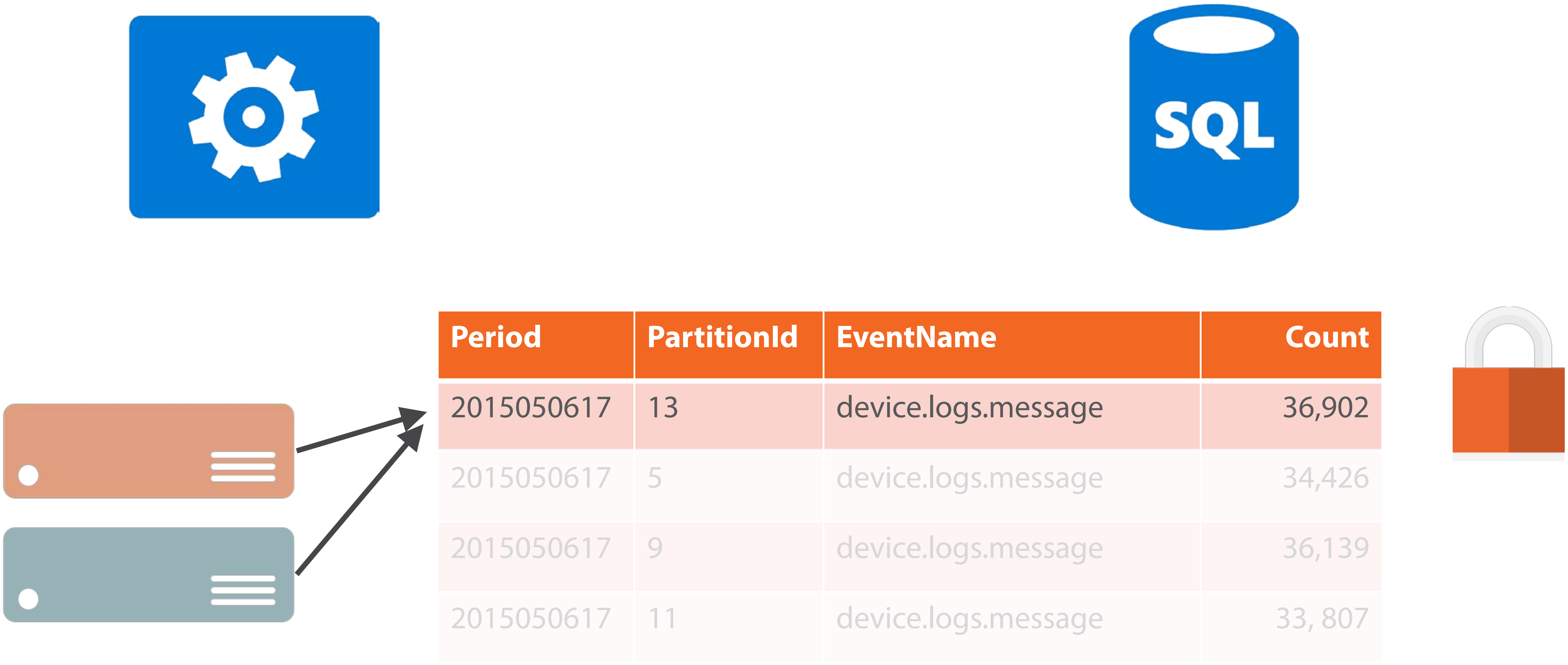
INSERT INTO EventMetrics

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807

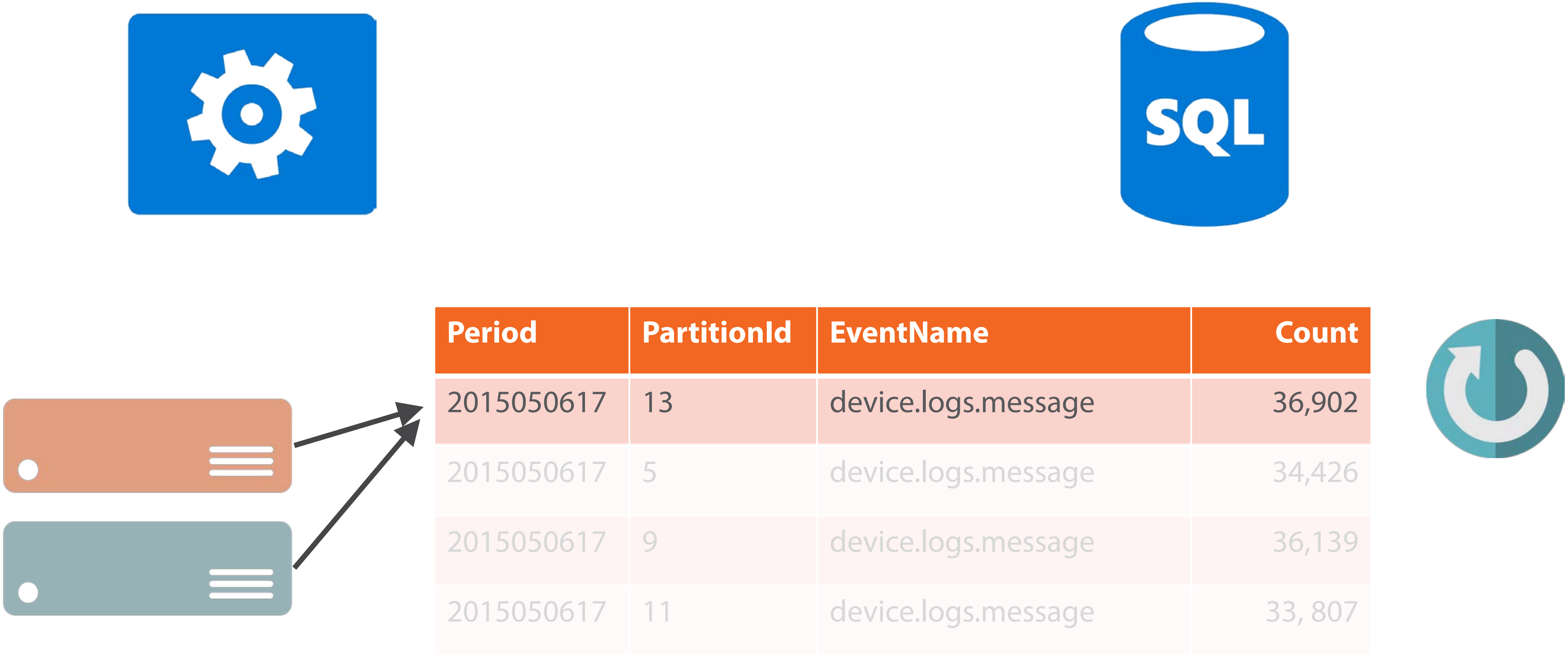


UPDATE EventMetrics

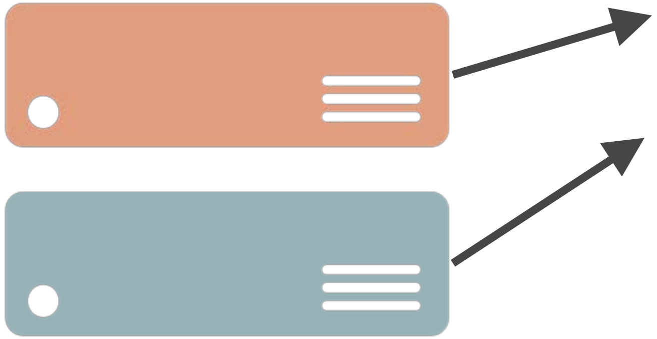
Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



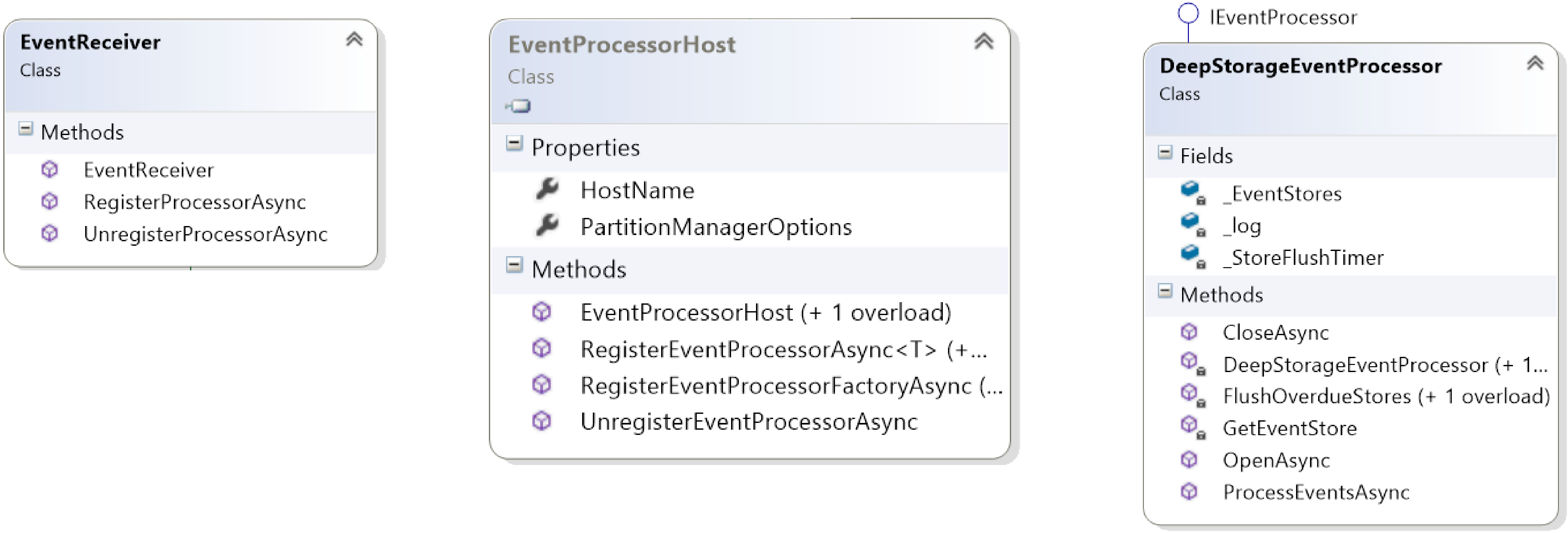
Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426
2015050617	9	device.logs.message	36,139
2015050617	11	device.logs.message	33, 807



EventReceiver
Class

Methods
- EventReceiver
- RegisterProcessorAsync
- UnregisterProcessorAsync

EventProcessorHost
Class

Properties
- HostName
- PartitionManagerOptions

Methods
- EventProcessorHost (+ 1 overload)
- RegisterEventProcessorAsync<T> (+...
- RegisterEventProcessorFactoryAsync (...
- UnregisterEventProcessorAsync

IEventProcessor

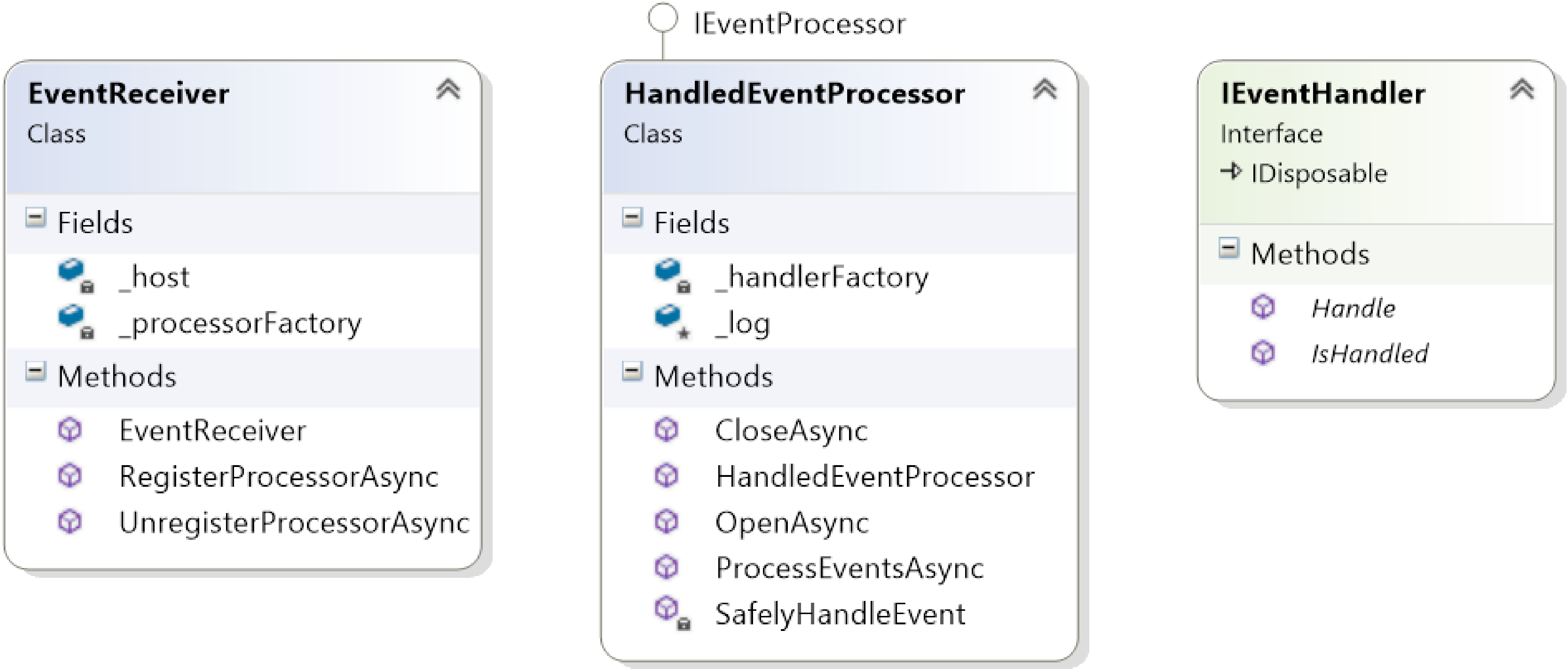
DeepStorageEventProcessor
Class

Fields
- _EventStores
- _log
- _StoreFlushTimer

Methods
- CloseAsync
- DeepStorageEventProcessor (+ 1...
- FlushOverdueStores (+ 1 overload)
- GetEventStore
- OpenAsync
- ProcessEventsAsync

Deep Storage Worker

EventReceiver, EventProcessorHost and IEventProcessor



IEventProcessor

EventReceiver
Class

Fields
- _host
- _processorFactory

Methods
- EventReceiver
- RegisterProcessorAsync
- UnregisterProcessorAsync

HandledEventProcessor
Class

Fields
- _handlerFactory
- _log

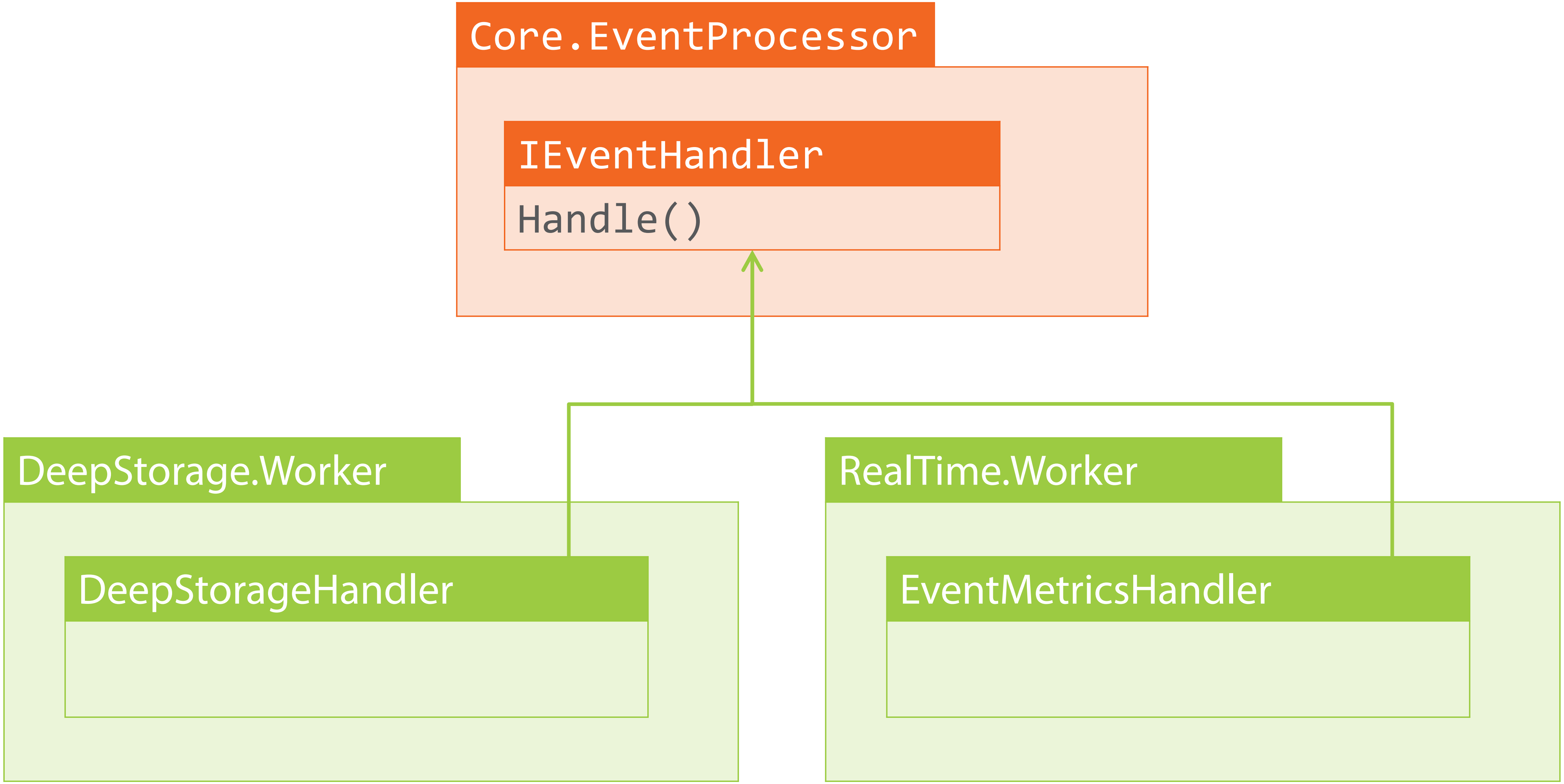
Methods
- CloseAsync
- HandledEventProcessor
- OpenAsync
- ProcessEventsAsync
- SafelyHandleEvent

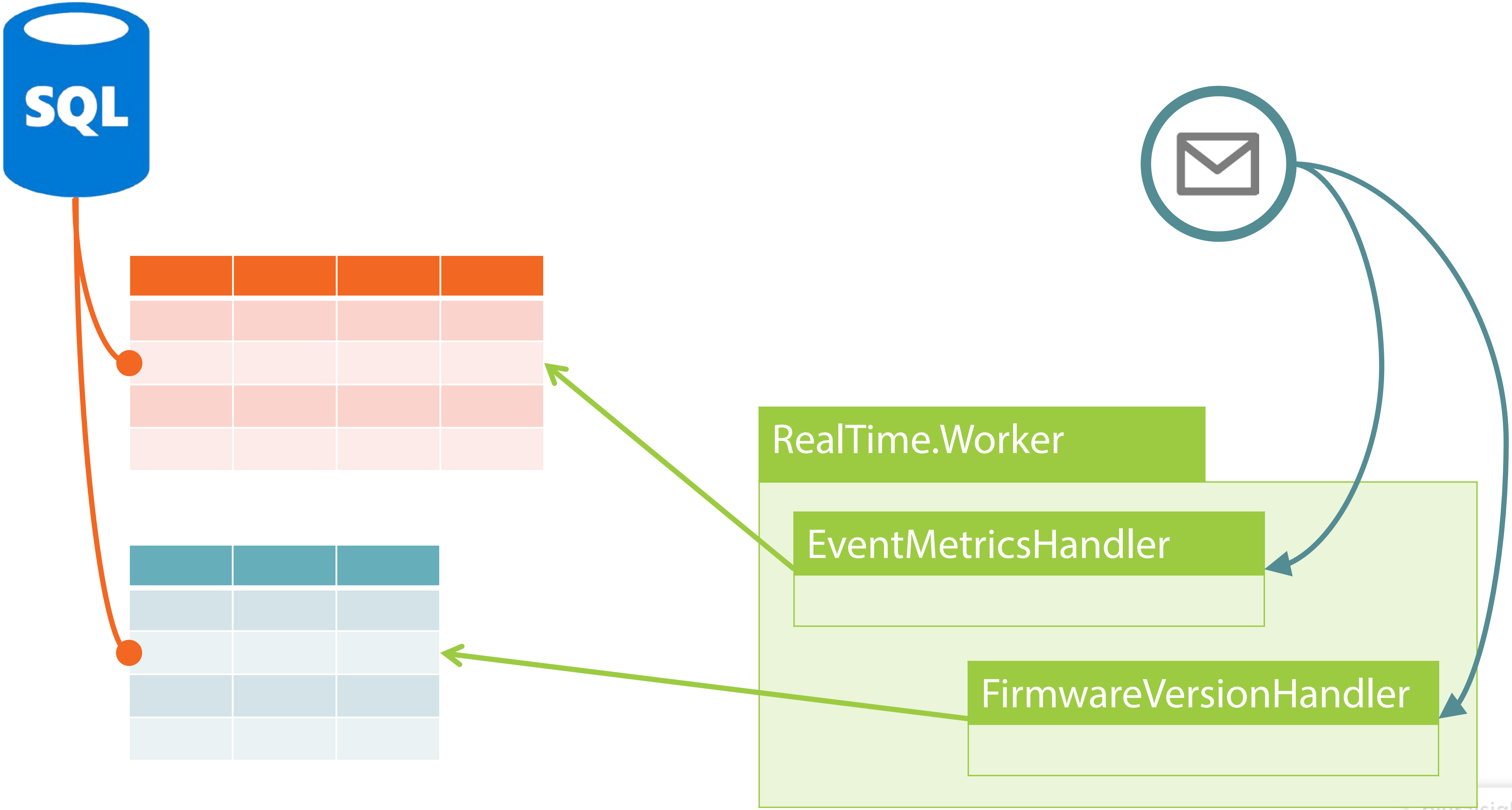
IEventHandler
Interface
→ IDisposable

Methods
- *Handle*
- *IsHandled*

Refactored Deep Storage Worker

EventReceiver, EventProcessorHost and IEventProcessor


Core.EventProcessor
IEventHandler
Handle()
DeepStorage.Worker
DeepStorageHandler
RealTime.Worker
EventMetricsHandler


SQL
RealTime.Worker
EventMetricsHandler
FirmwareVersionHandler

Demo: Core.EventProcessor

Refactored Deep Storage Worker

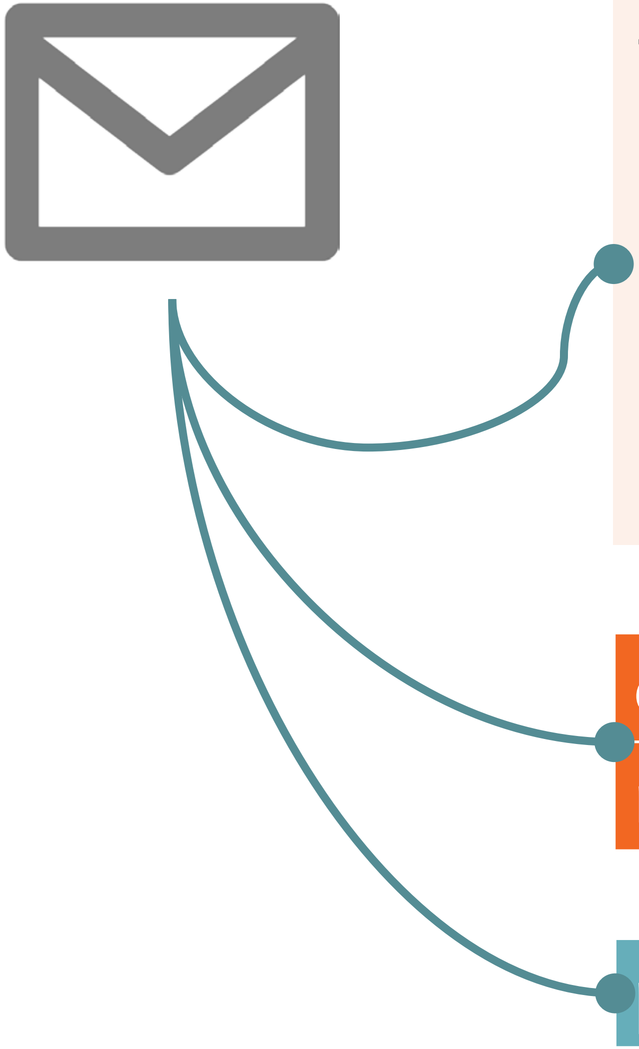
Common EventReceiver

Custom IEventHandler

```
public interface IEventHandler : IDisposable
{
    bool IsHandled(string eventName);
   void Handle(EventData eventData, string partitionId);
}
```

IEventHandler

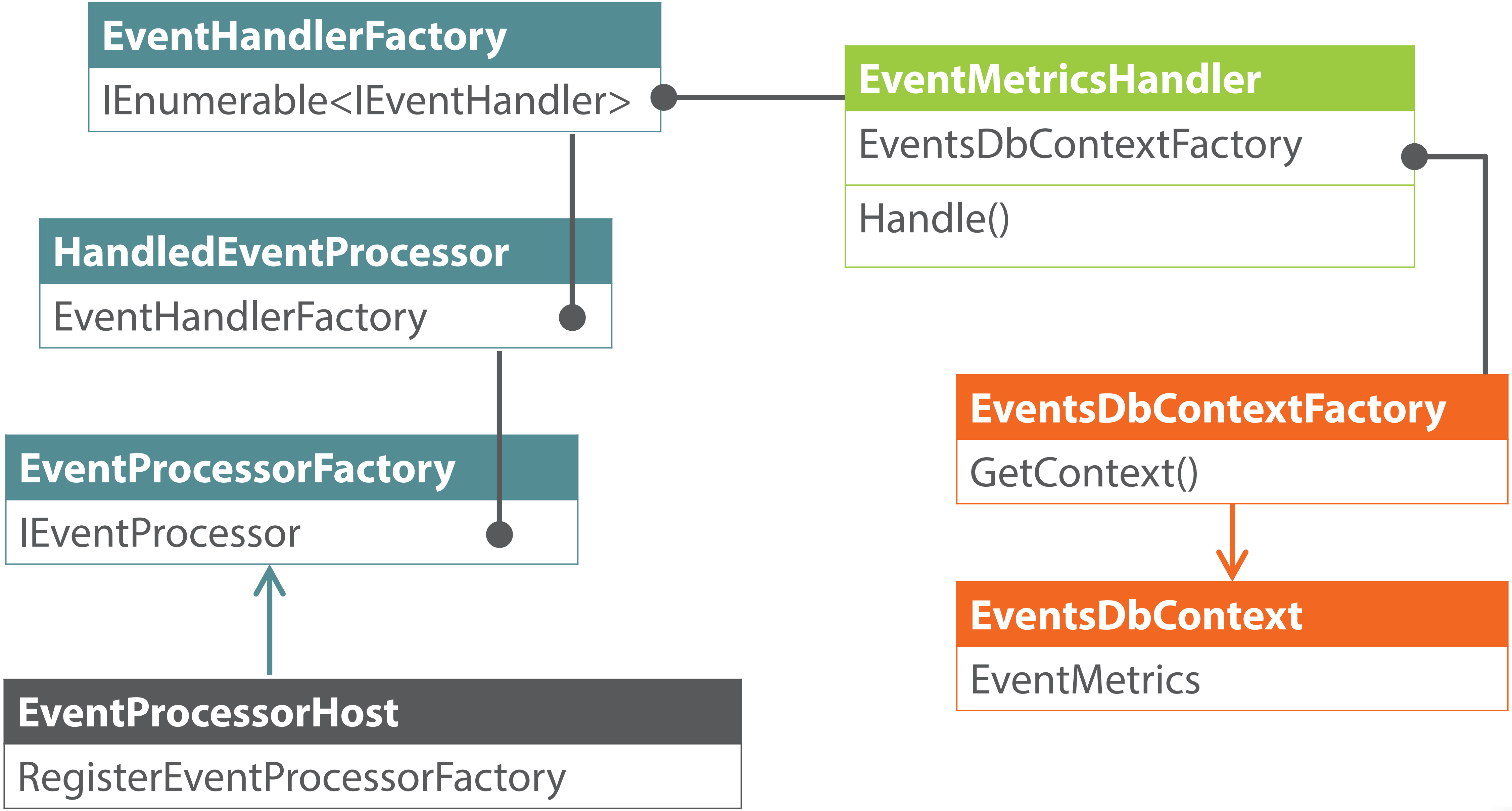
Operates on single EventData



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

eventName	device.logs.message
receivedAt	1415879253000

partitionId	11



EventHandlerFactory

IEnumerable<IEventHandler>

HandledEventProcessor

EventHandlerFactory

EventProcessorFactory

IEventProcessor

EventProcessorHost

RegisterEventProcessorFactory

EventMetricsHandler

EventsDbContextFactory

Handle()

EventsDbContextFactory

GetContext()

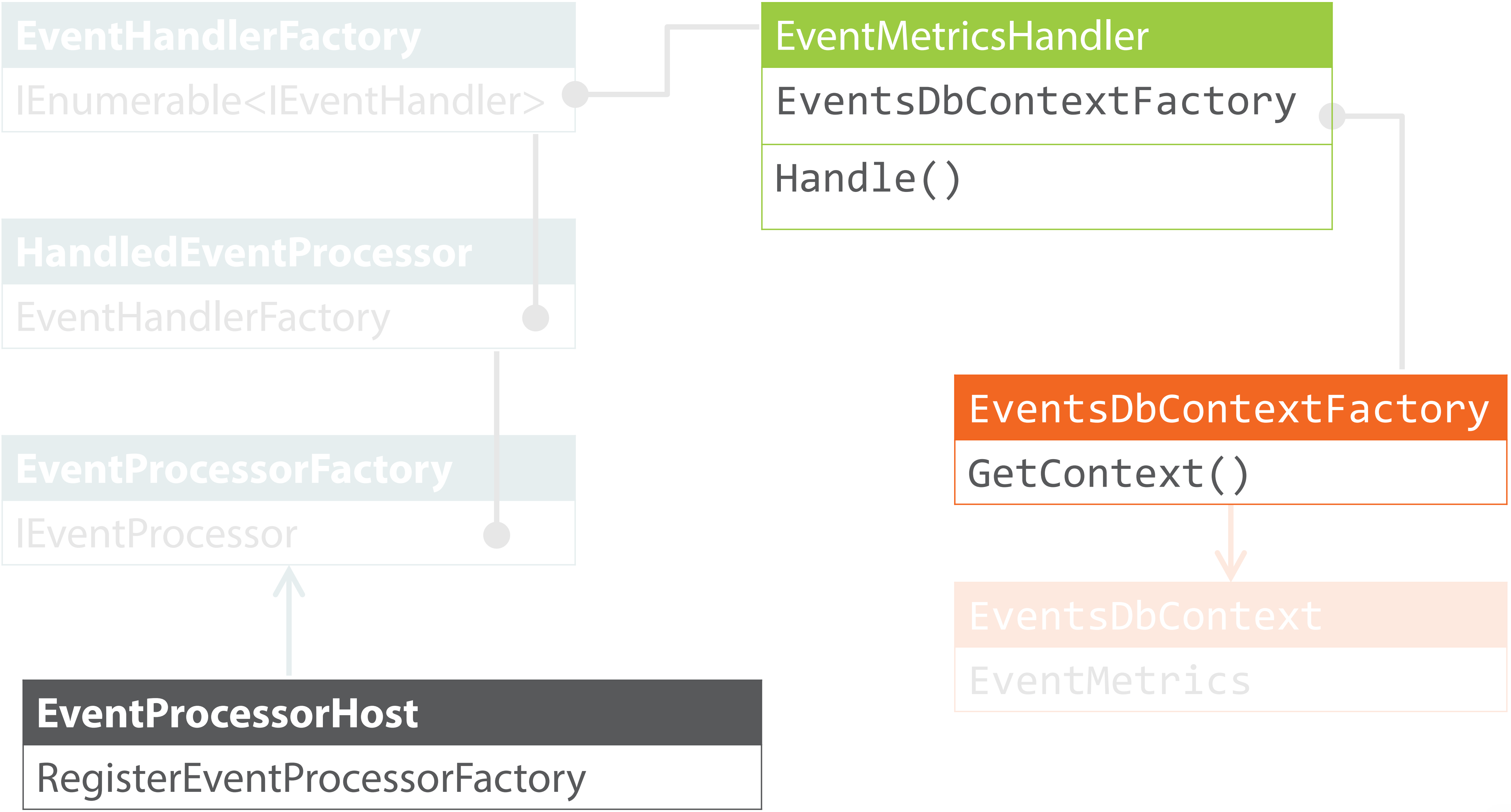
EventsDbContext

EventMetrics

```
container.RegisterType<IEventHandler,
                   DeepStorageEventHandler>("DeepStorageEventHandler");
container.RegisterType<EventHandlerFactory>();

container.RegisterType<IEventProcessor, HandledEventProcessor>();
container.RegisterType<IEventProcessorFactory,EventProcessorFactory>();
```

Worker Role

Register dependency chain



EventHandlerFactory

IEnumerable<IEventHandler>

HandledEventProcessor

EventHandlerFactory

EventProcessorFactory

IEventProcessor

EventProcessorHost

RegisterEventProcessorFactory

EventMetricsHandler

EventsDbContextFactory

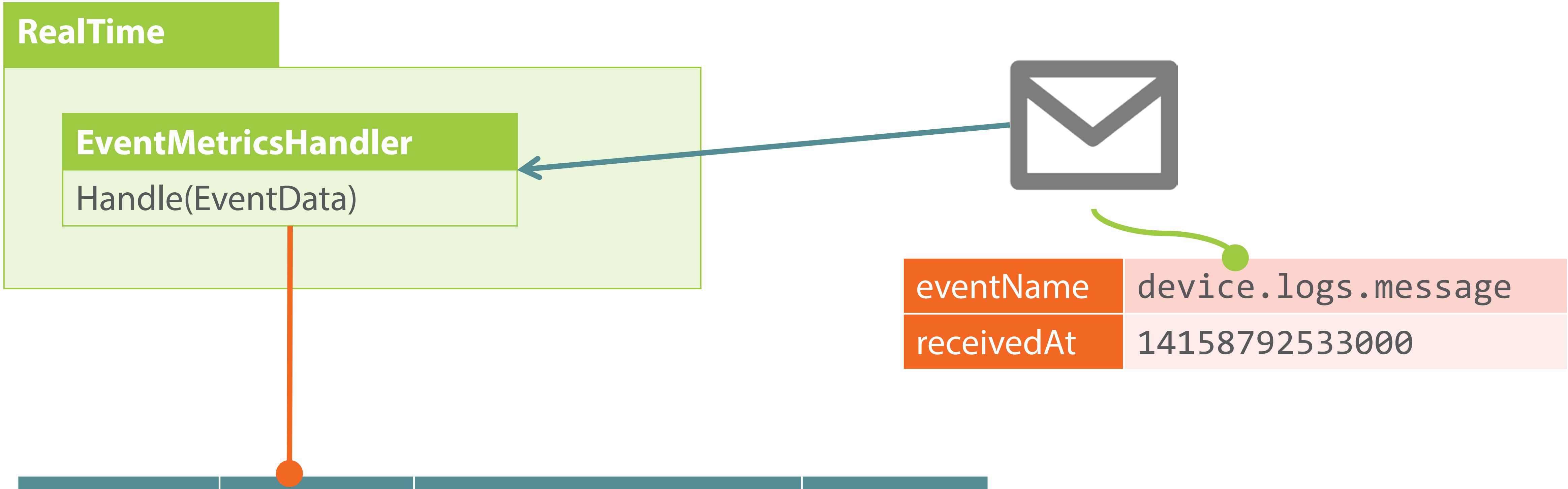
Handle()

EventsDbContextFactory

GetContext()

EventsDbContext

EventMetrics



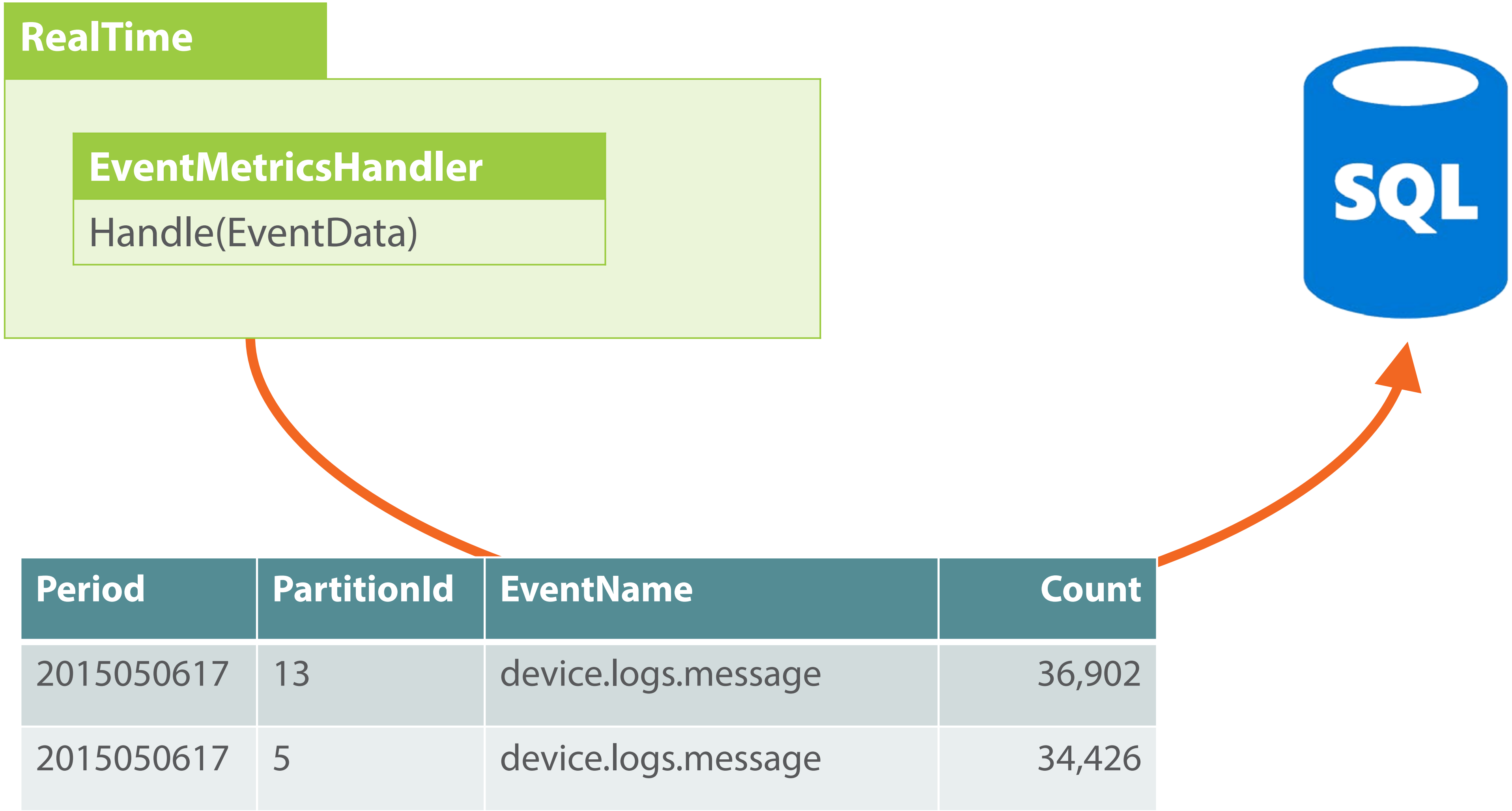
RealTime

EventMetricsHandler

Handle(EventData)

eventName	device.logs.message
receivedAt	1415879253300

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426

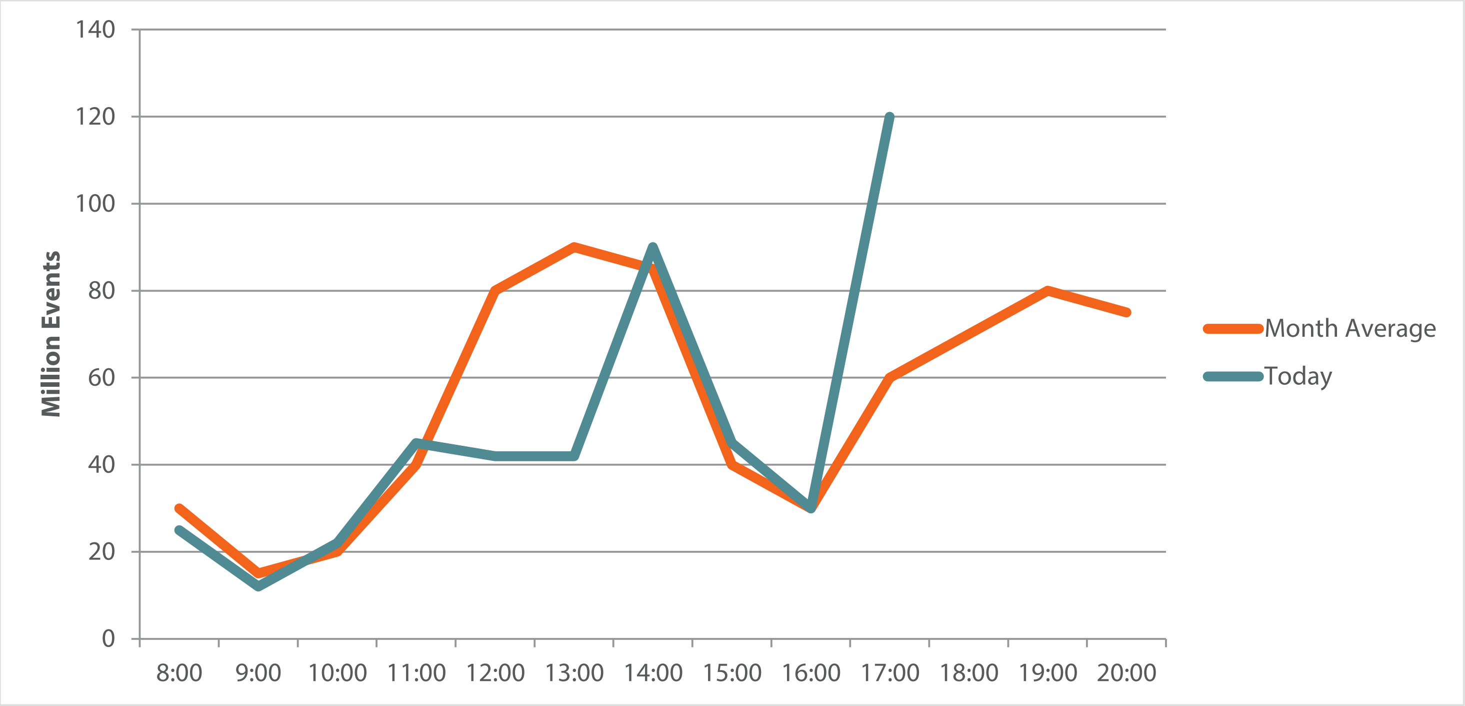


RealTime

EventMetricsHandler

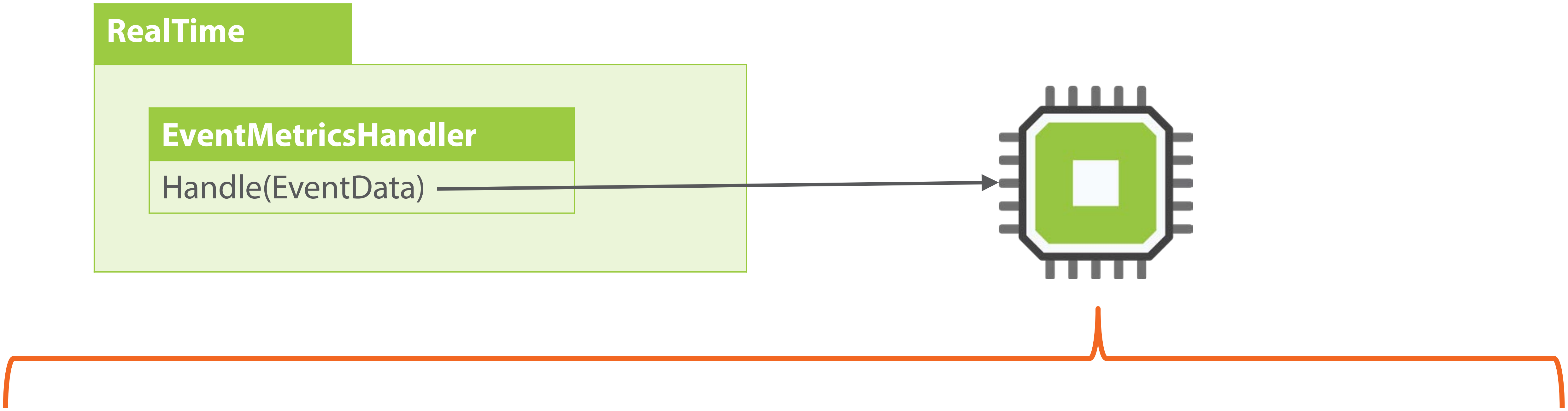
Handle(EventData)

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426



SQL

Period	PartitionId	EventName	Count
2015050617	13	device.logs.message	36,902
2015050617	5	device.logs.message	34,426



RealTime

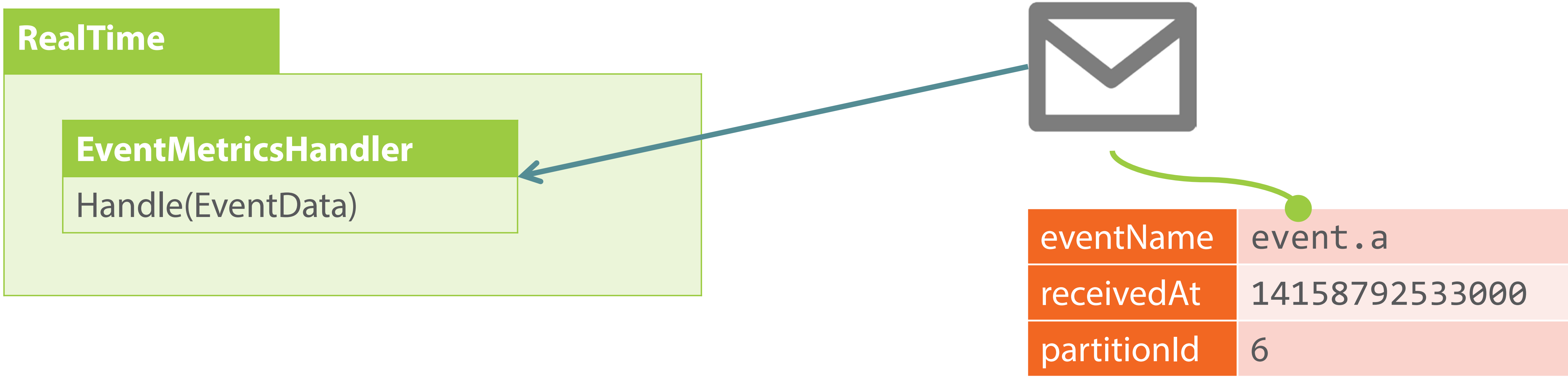
EventMetricsHandler

Handle(EventData)

```
{
  "period":2015050717,
  "partitionId" : "11",
  "eventName": "event.a",
  "count": 32016
}
```

```
{
  "period":2015050717,
  "partitionId" : "6",
  "eventName": "event.a",
  "count": 31943
}
```

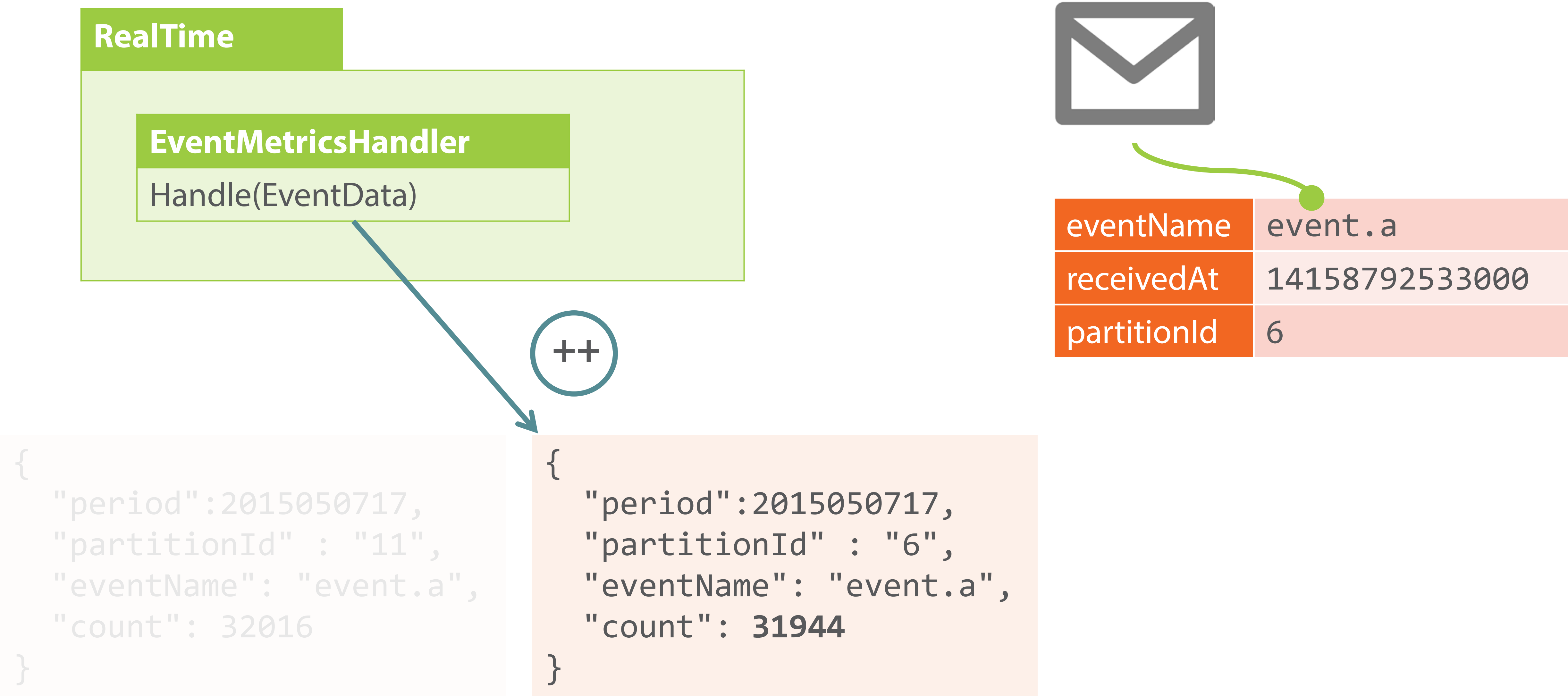
```
{
  "period":2015050717,
  "partitionId" : "1",
  "eventName": "event.b",
  "count": 184
}
```



eventName	event.a
receivedAt	14158792533000
partitionId	6

```
{
   "period":2015050717,
   "partitionId" : "11",
   "eventName": "event.a",
   "count": 32016
}
```

```
{
   "period":2015050717,
   "partitionId" : "6",
   "eventName": "event.a",
   "count": 31943
}
```



RealTime

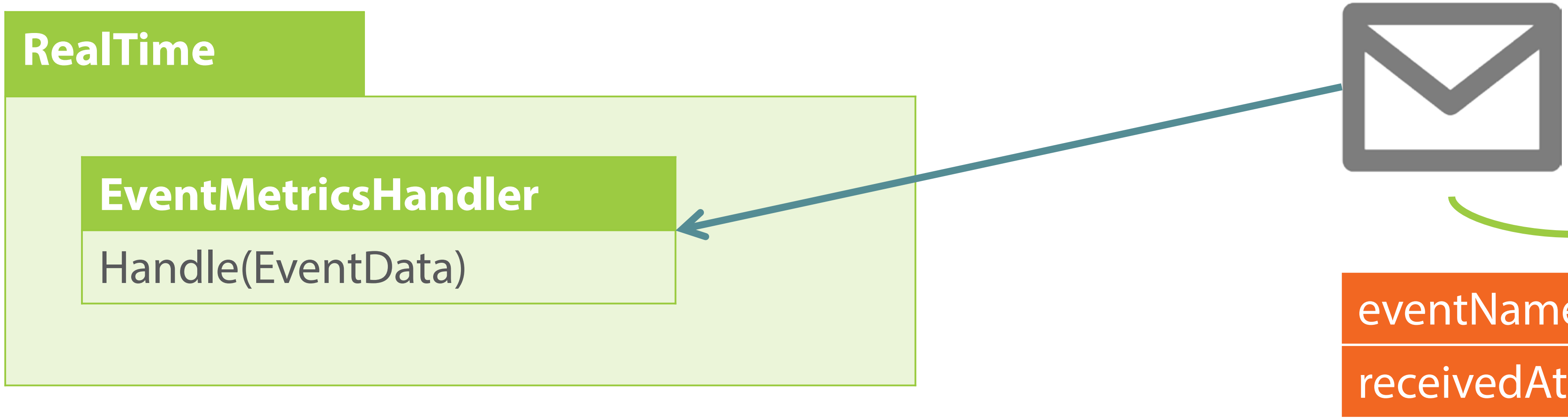
EventMetricsHandler

Handle(EventData)

eventName	event.a
receivedAt	1415879253300
partitionId	6

```
{
  "period":2015050717,
  "partitionId" : "11",
  "eventName": "event.a",
  "count": 32016
}
```

```
{
  "period":2015050717,
  "partitionId" : "6",
  "eventName": "event.a",
  "count": 31944
}
```



RealTime

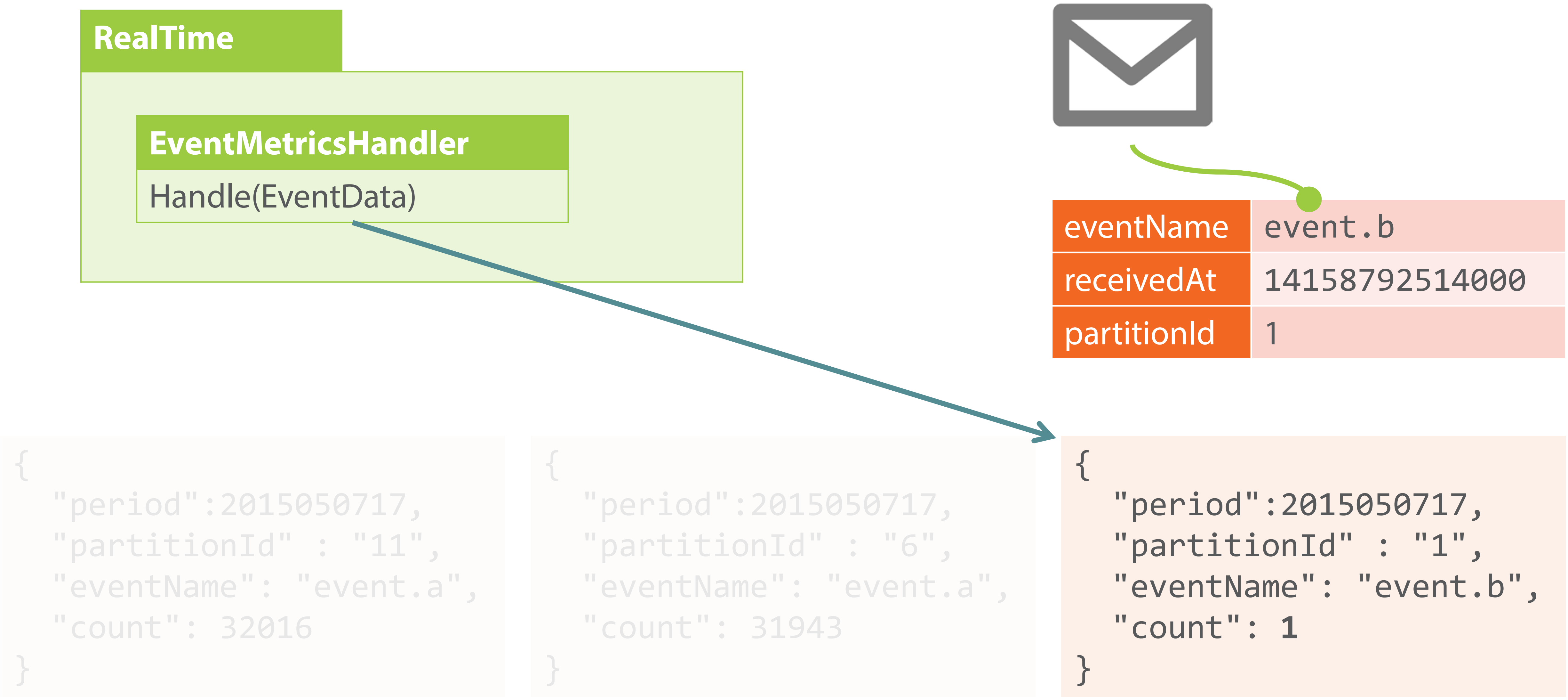
EventMetricsHandler

Handle(EventData)

eventName	event.b
receivedAt	14158792514000
partitionId	1

```
{
   "period":2015050717,
   "partitionId" : "11",
   "eventName": "event.a",
   "count": 32016
}
```

```
{
   "period":2015050717,
   "partitionId" : "6",
   "eventName": "event.a",
   "count": 31944
}
```



RealTime

EventMetricsHandler

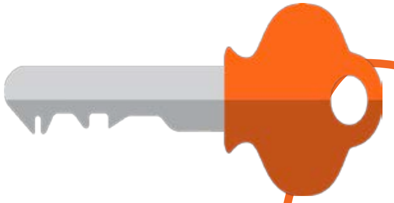
Handle(EventData)

eventName	event.b
receivedAt	1415879251400‌0
partitionId	1

```
{
  "period":2015050717,
  "partitionId" : "11",
  "eventName": "event.a",
  "count": 32016
}
```

```
{
  "period":2015050717,
  "partitionId" : "6",
  "eventName": "event.a",
  "count": 31943
}
```

```
{
  "period":2015050717,
  "partitionId" : "1",
  "eventName": "event.b",
  "count": 1
}
```



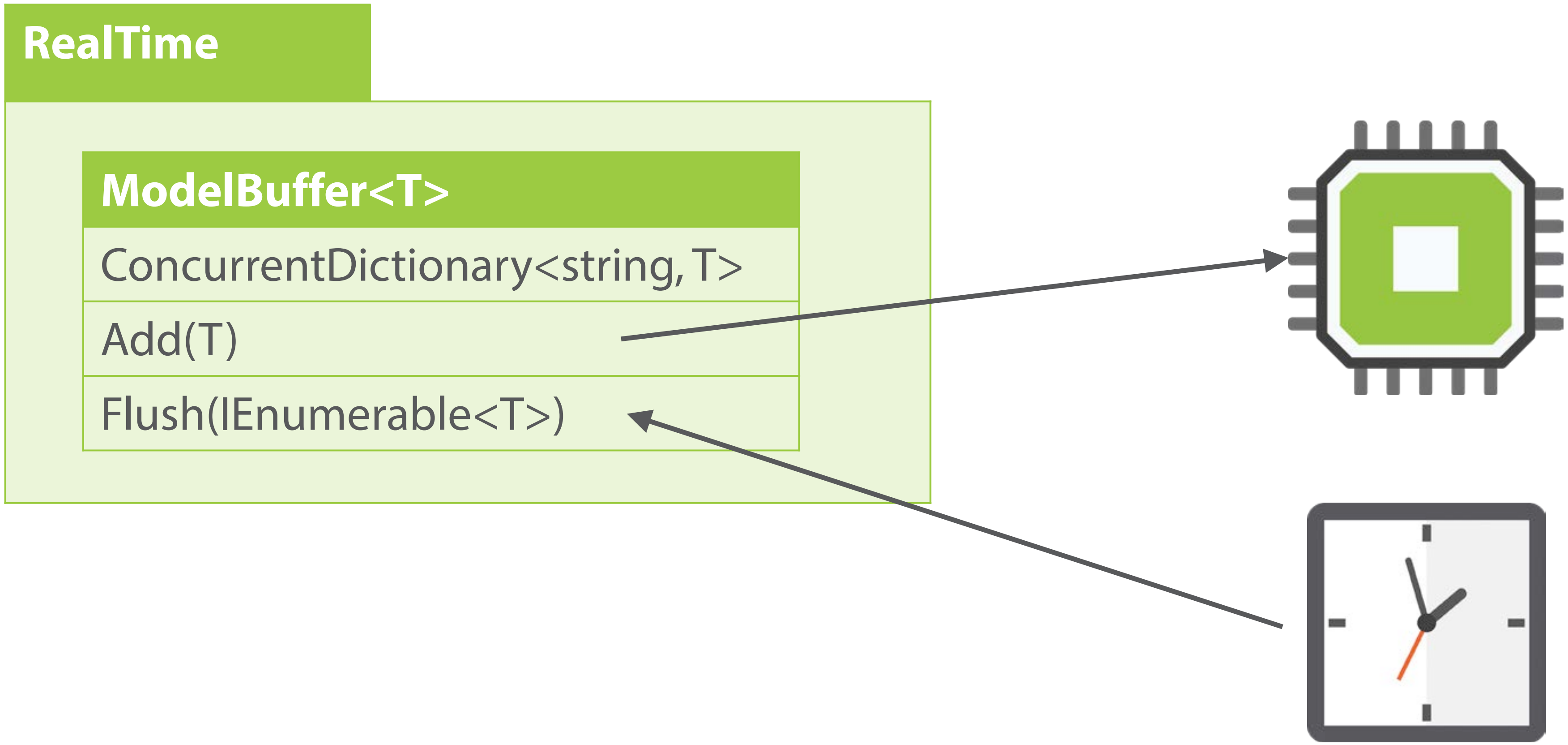
eventName	event.b
receivedAt	14158792514000
partitionId	1

```
ConcurrentDictionary<string, EventMetric>
```

```
{
  "period":2015050717,
  "partitionId" : "11",
  "eventName": "event.a",
  "count": 32016
}
```

```
{
  "period":2015050717,
  "partitionId" : "6",
  "eventName": "event.a",
  "count": 31943
}
```

```
{
  "period":2015050717,
  "partitionId" : "1",
  "eventName": "event.b",
  "count": 184
}
```



RealTime

ModelBuffer<T>

ConcurrentDictionary<string, T>

Add(T)

Flush(IEnumerable<T>)

Demo: ModelBuffer<T>

Buffer in-memory

Flush at elapsed time

Used by Event Handler

```
public class ModelBuffer<TModel>
{
    private Action<IEnumerable<TModel>> _doSave;
    private Func<TModel, string> _getKey;
    private ConcurrentDictionary<string, TModel> _models;
```

ModelBuffer<TModel>

TModel is a code-first EF type

```
_models.TryAdd(key, model)
_models.TryGetValue(key, out model)
_models.Select(x => x.Value)
```

ConcurrentDictionary<TKey, TValue>

Use threadsafe methods with minimal locking

```
var buffer = new ModelBuffer<StubModel>(
   x => SaveModels(x),
   y => string.Format("{0}p{1}i{2}",
                      y.Period, y.PartitionId, y.Index),
   TimeSpan.FromMilliseconds(200));
```

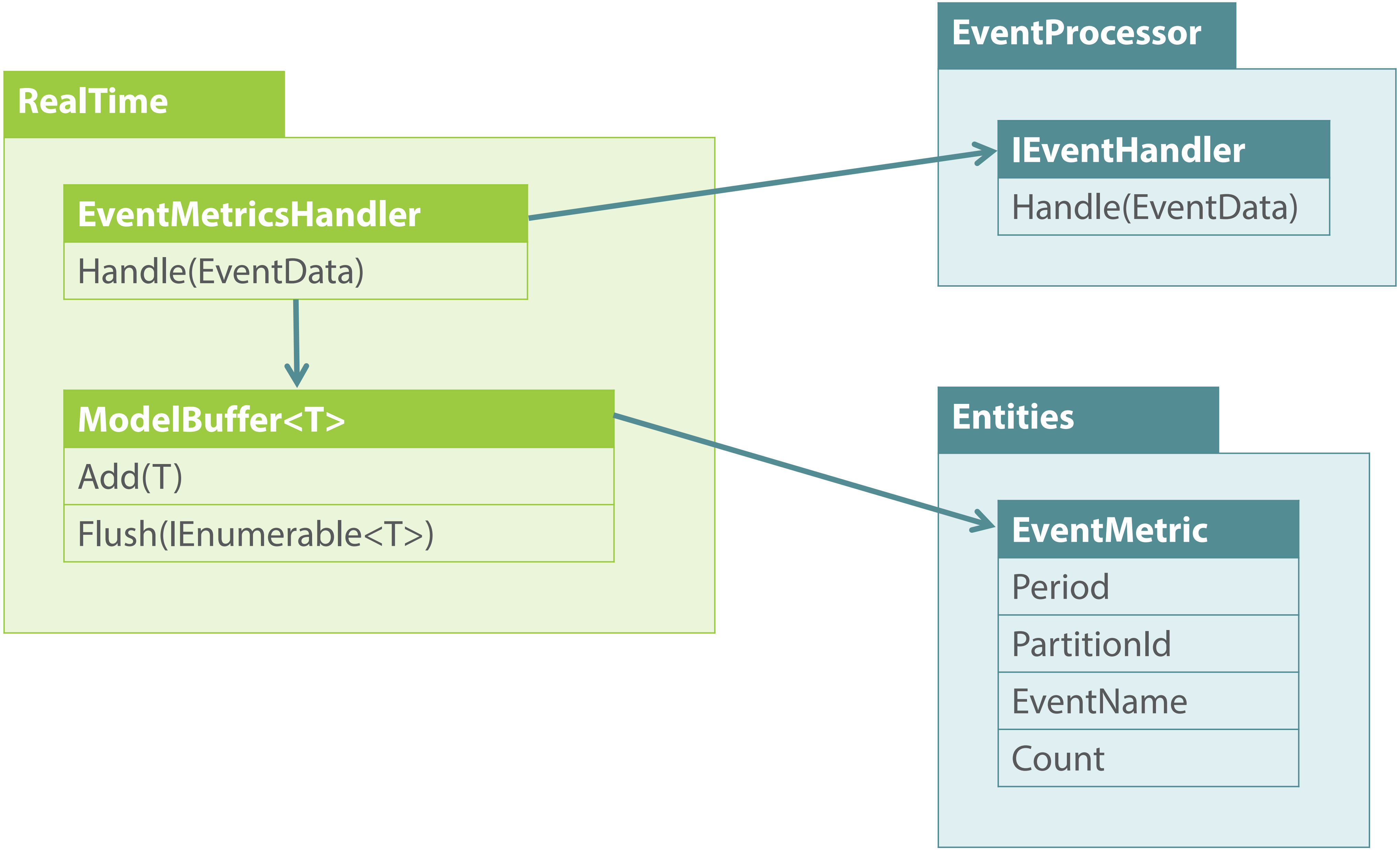
ModelBuffer<T> usage

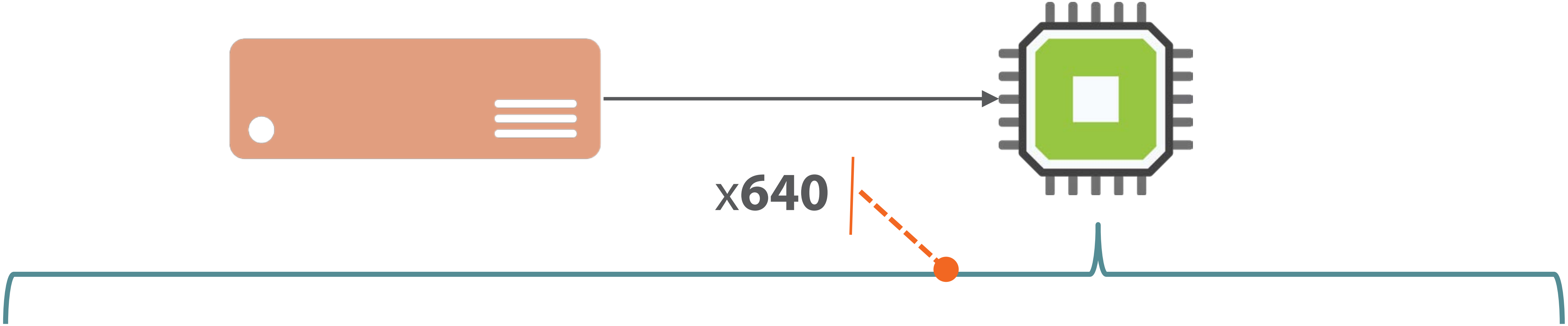
Flush action, key builder func and flush period

```
<add key="ModelBuffers.EventMetric.BufferFlushTime"
     value="00:00:00.05" />

<add key="ModelBuffers.BufferFlushTime"
     value="00:00:05" />
```

App Settings

Custom & default flush periods

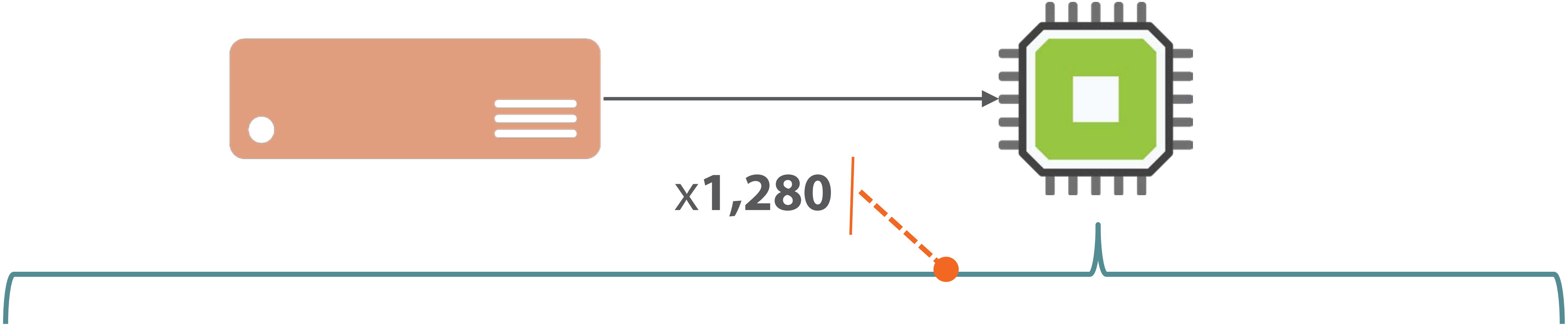

RealTime
EventMetricsHandler
Handle(EventData)
ModelBuffer<T>
Add(T)
Flush(IEnumerable<T>)
EventProcessor
IEventHandler
Handle(EventData)
Entities
EventMetric
Period
PartitionId
EventName
Count



```
{
  "period":2015050720,
  "eventName": "event.a",
  "count": 32016
}
```

```
{
  "period":2015050720,
  "eventName": "event.b",
  "count": 12961
}
```

```
{
  "period":2015050720,
  "eventName": "event.c",
  "count": 184
}
```



```
{
  "period":2015050720,
  "eventName": "event.a",
  "count": 51974
}
```

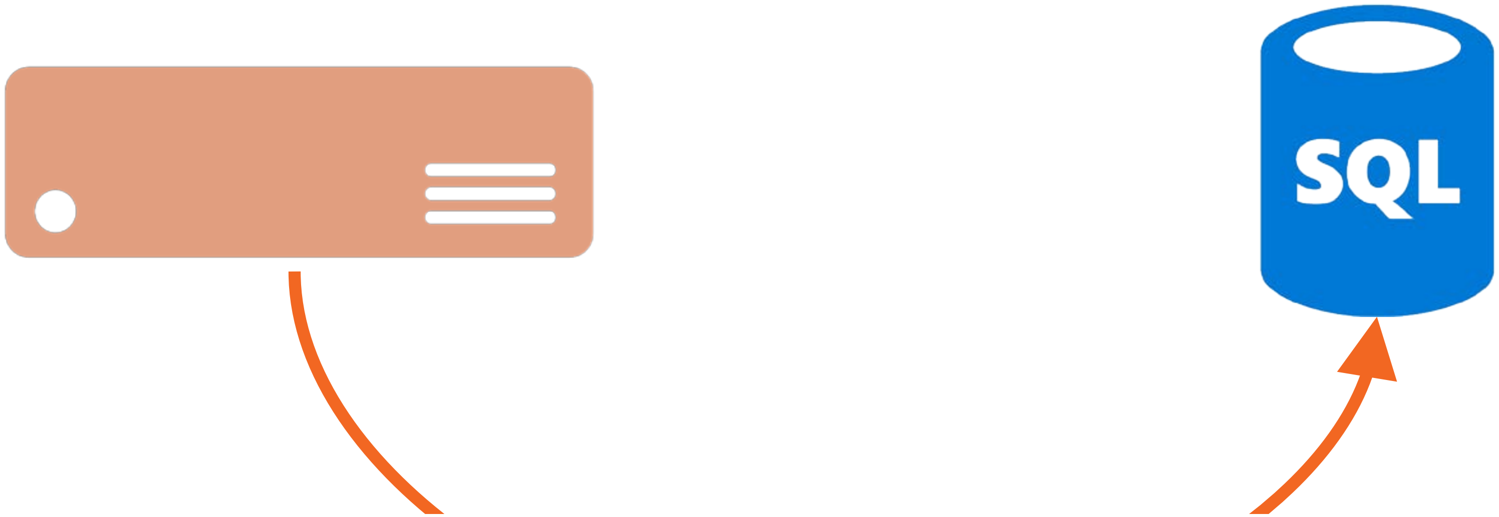
```
{
  "period":2015050720,
  "eventName": "event.b",
  "count": 18621
}
```

```
{
  "period":2015050720,
  "eventName": "event.c",
  "count": 348
}
```

```
{
  "period":2015050721,
  "eventName": "event.a",
  "count": 1574
}
```

```
{
  "period":2015050721,
  "eventName": "event.b",
  "count": 357
}
```

```
{
  "period":2015050721,
  "eventName": "event.c",
  "count": 2
}
```



```
{
  "period":2015050720,
  "eventName": "event.a",
  "count": 51974
}
```

```
{
  "period":2015050720,
  "eventName": "event.b",
  "count": 18621
}
```

```
{
  "period":2015050720,
  "eventName": "event.c",
  "count": 348
}
```

```
{
  "period":2015050721,
  "eventName": "event.a",
  "count": 1574
}
```

```
{
  "period":2015050721,
  "eventName": "event.b",
  "count": 357
}
```

```
{
  "period":2015050721,
  "eventName": "event.c",
  "count": 2
}
```

SQL

Period	PartitionId	EventName	Count
2015050617	1	device.logs.message	336,902
2015050617	2	device.logs.message	324,426
		...	
2015050617	14	device.logs.message	306,139
2015050617	15	device.logs.message	313, 807

Demo: EventMetricsHandler

Implements IEventHandler

DI for EventsDbContext

ModelBuffer<EventMetric>

```
public class EventMetricsHandler : IEventHandler
{
  public bool IsHandled(string eventName)
  {
    return true;
  }
```

EventMetricsHandler

Handles all event types

```
var eventName = eventData.GetEventName();
var period = eventData.GetReceivedAtHour();
var metric = GetMetric(eventName, period, partitionId);
metric.Count += 1;
metric.ProcessedAt = DateTime.UtcNow;
```

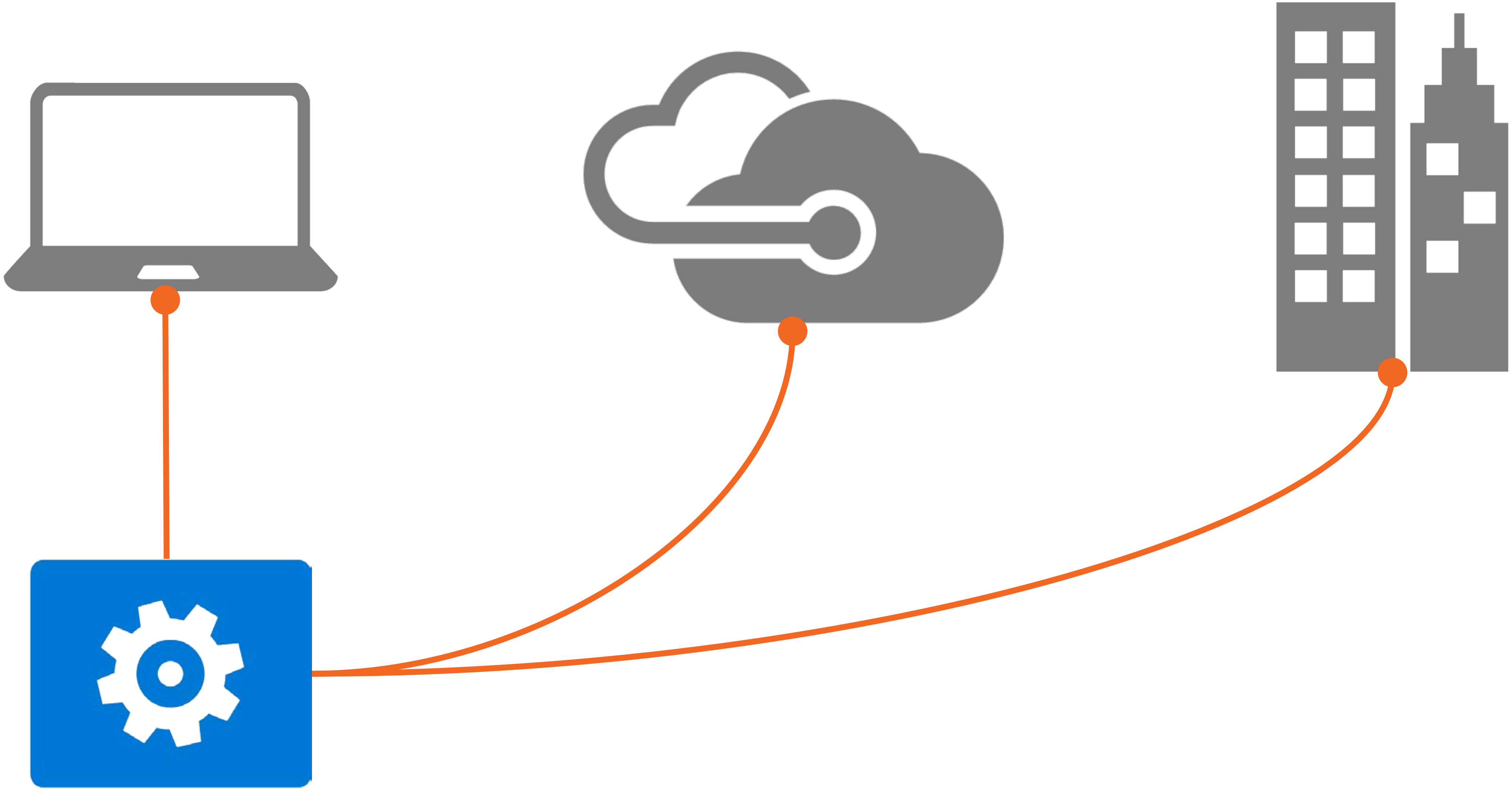
Handle(EventData)

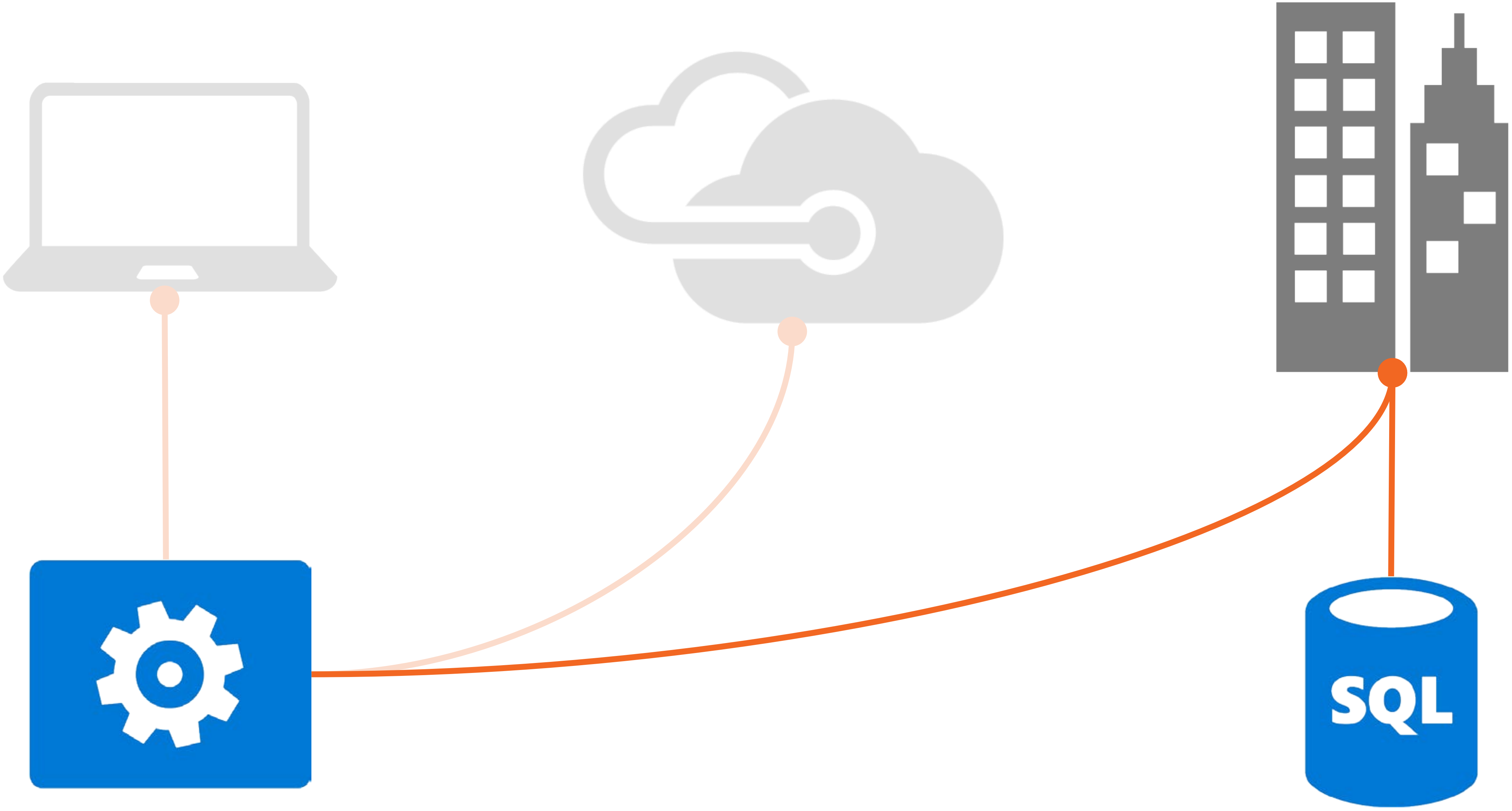
Finds EventMetric from buffer and updates

```
var existingMetric = db.EventMetrics.FirstOrDefault(
       x => x.PartitionId == metric.PartitionId &&
            x.Period == metric.Period &&
            x.EventName == metric.EventName);
```

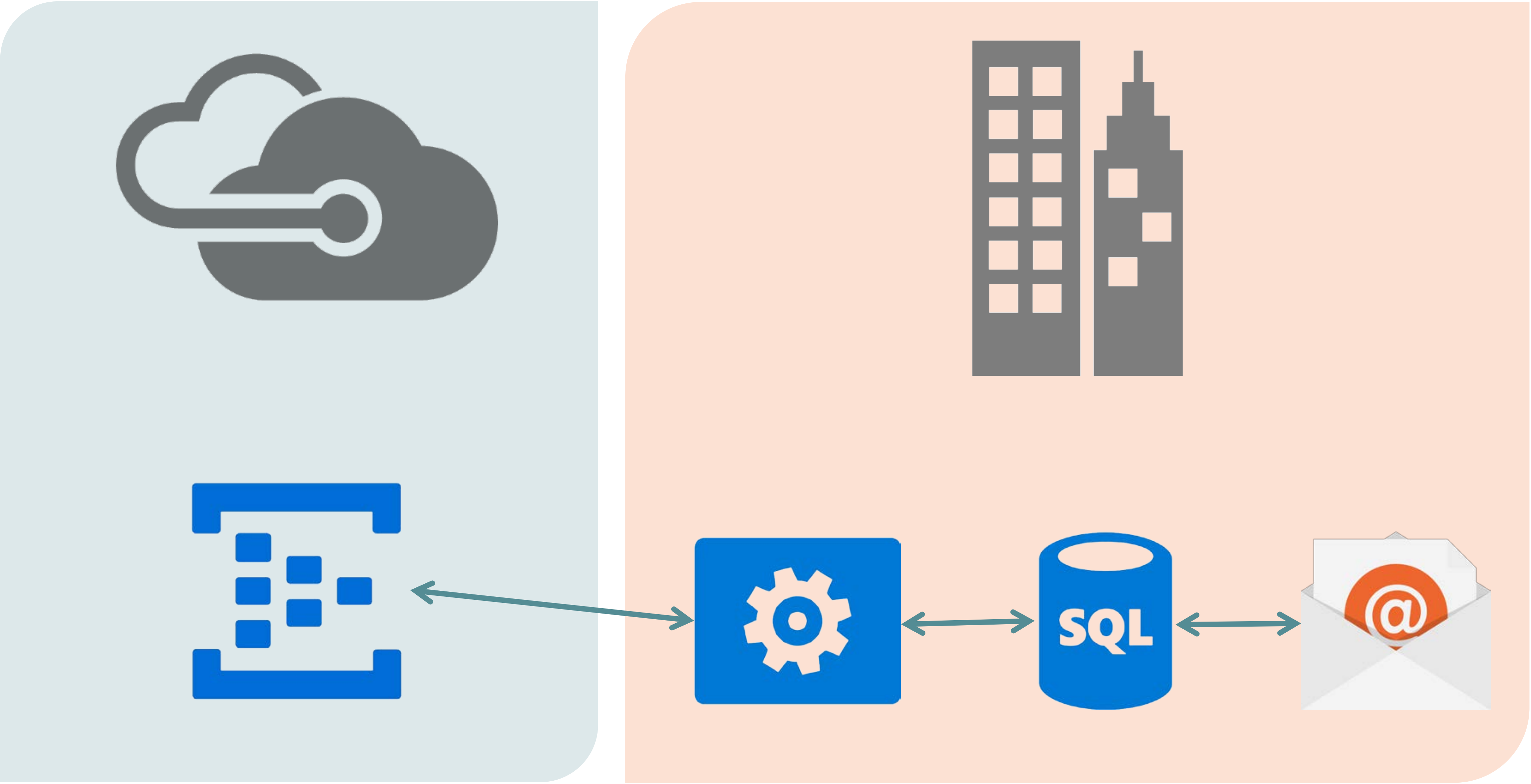
SaveMetrics

Checks for row in database and updates/inserts





SQL


SQL

Demo: Real Time Worker Role

Configure container

Start EventReceiver

Update SQL Azure

```
container.RegisterType<IEventHandler,
                       EventMetricsHandler>("EventMetricsHandler");
container.RegisterType<EventHandlerFactory>();
container.RegisterType<IEventProcessor, HandledEventProcessor>();
container.RegisterType<IEventProcessorFactory,EventProcessorFactory>();
```

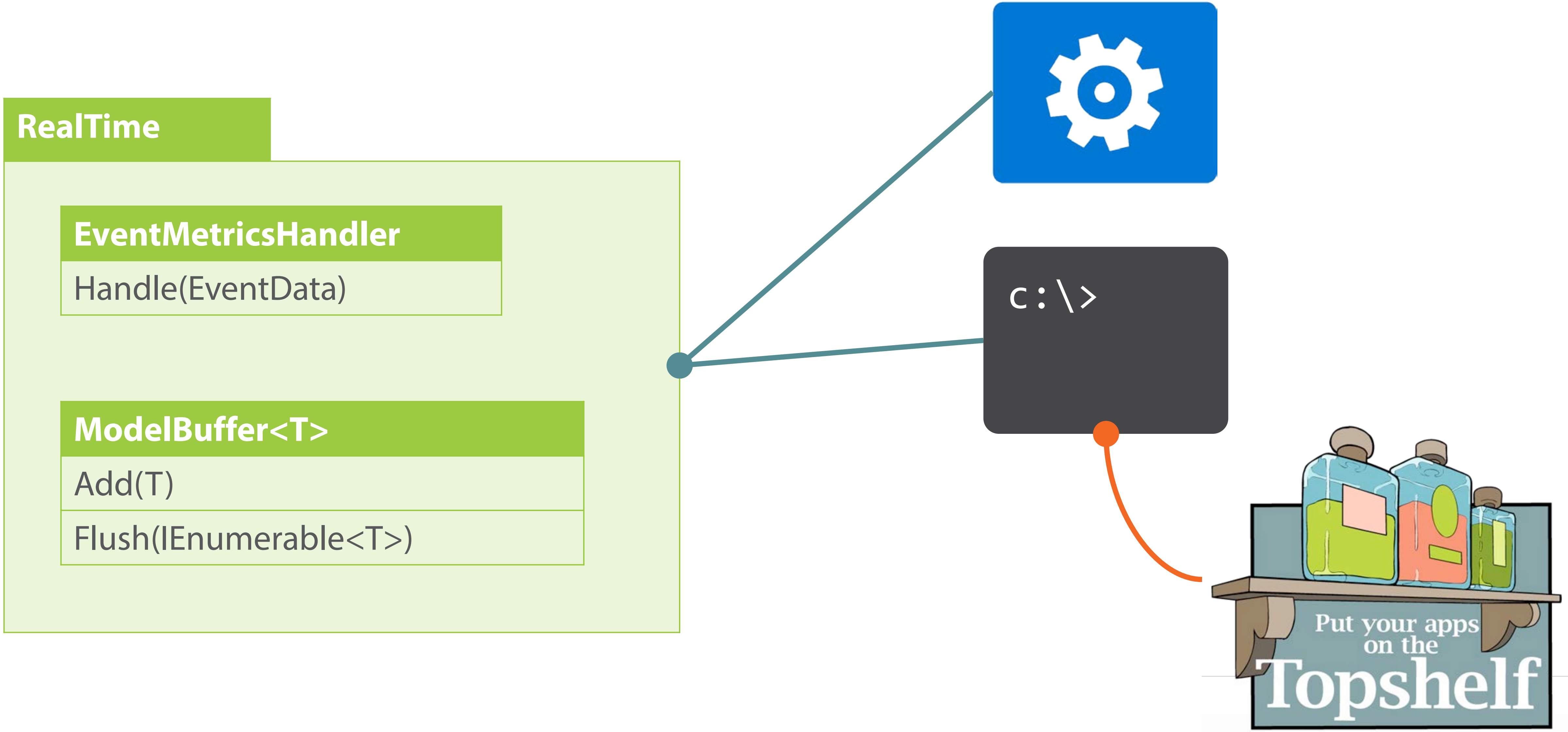
WorkerRole

Register dependencies with Unity

```
container.RegisterType<EventReceiver>(
     new InjectionConstructor(
     new ResolvedParameter<IEventProcessorFactory>(),
     eventHubName, consumerGroupName,
     eventHubConnectionString, checkpointConnectionString));
_receiver = container.Resolve<EventReceiver>();
```

EventReceiver

From Unity with dependency chain resolved



RealTime

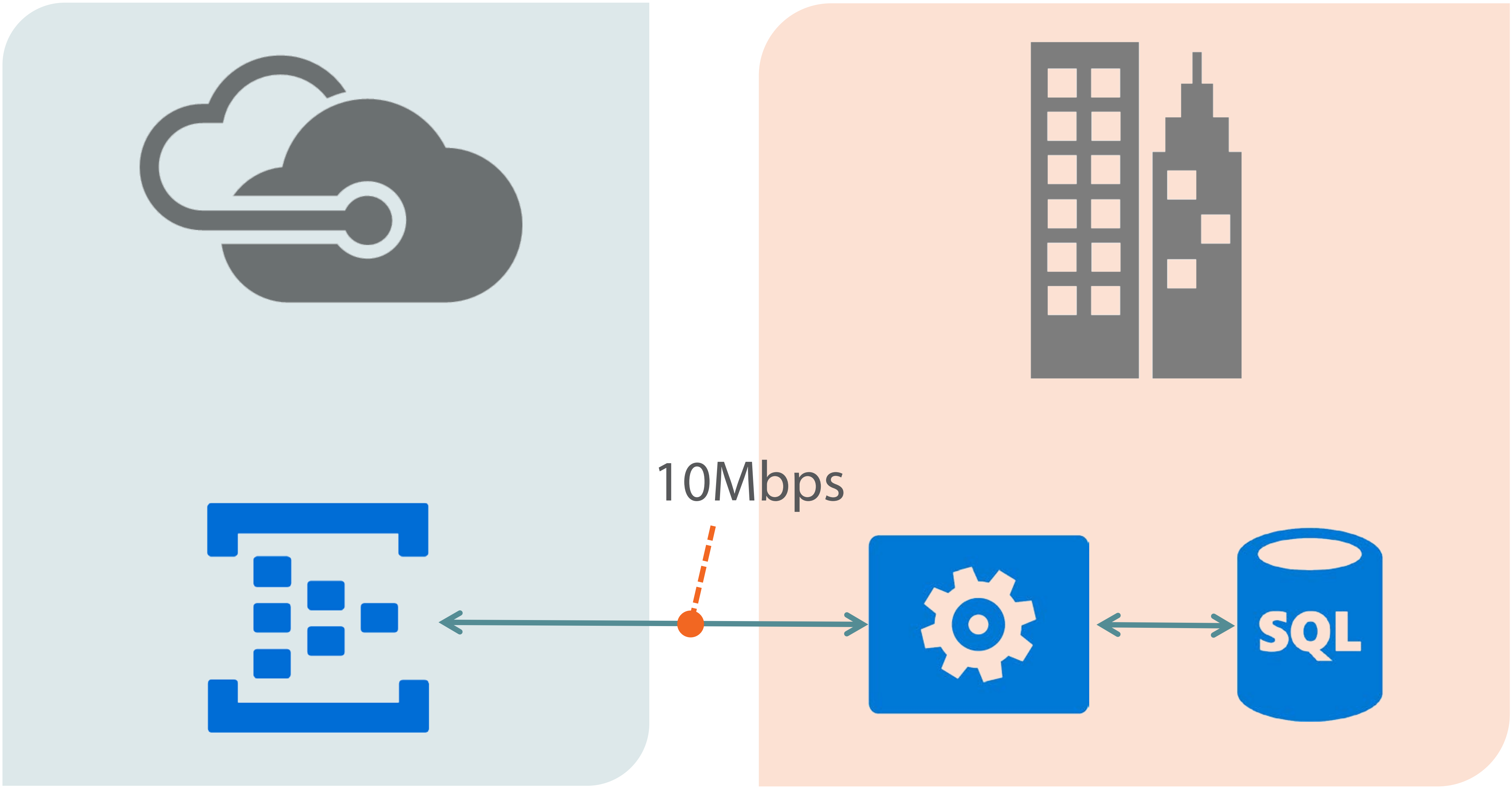
EventMetricsHandler

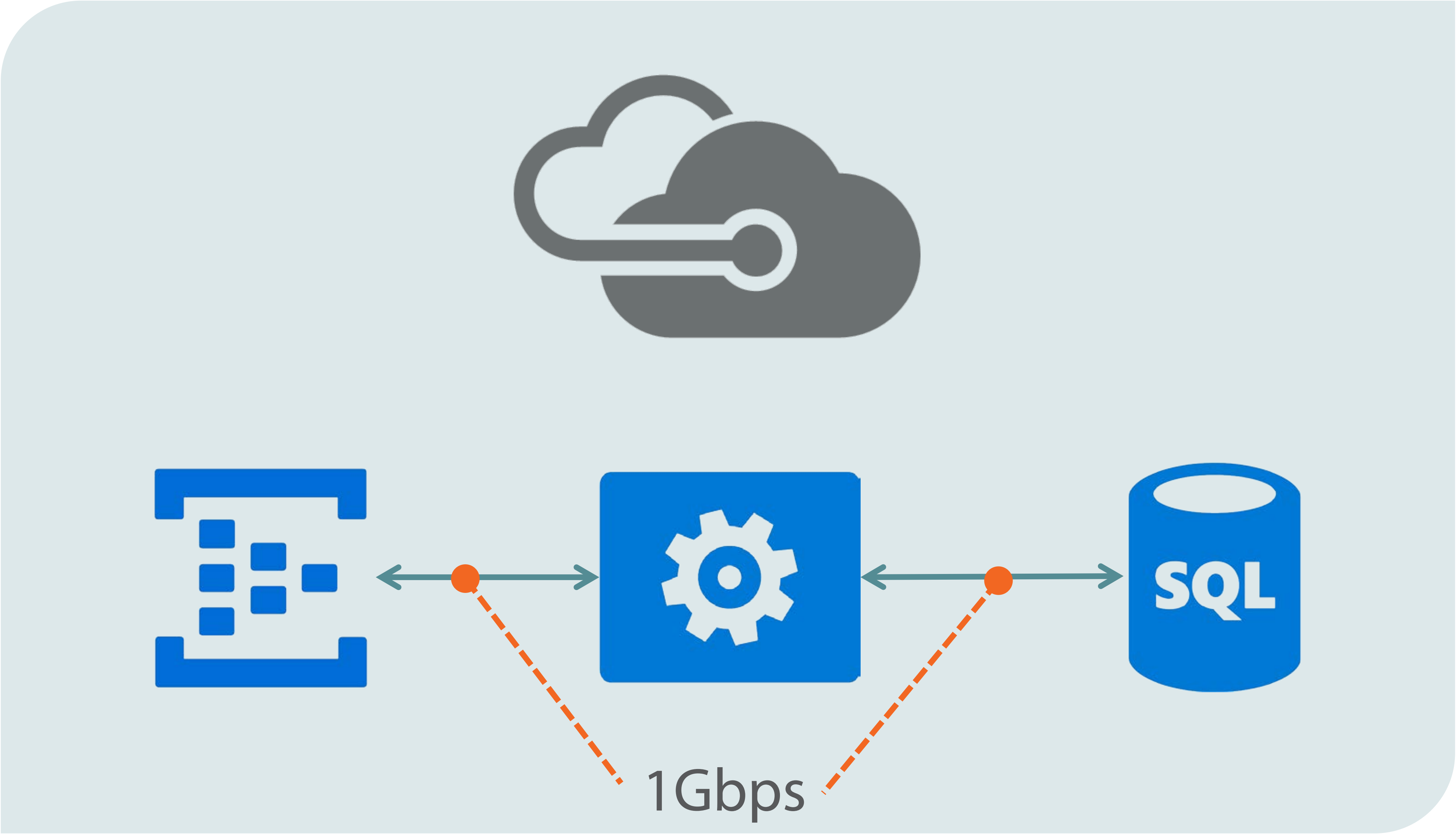
Handle(EventData)

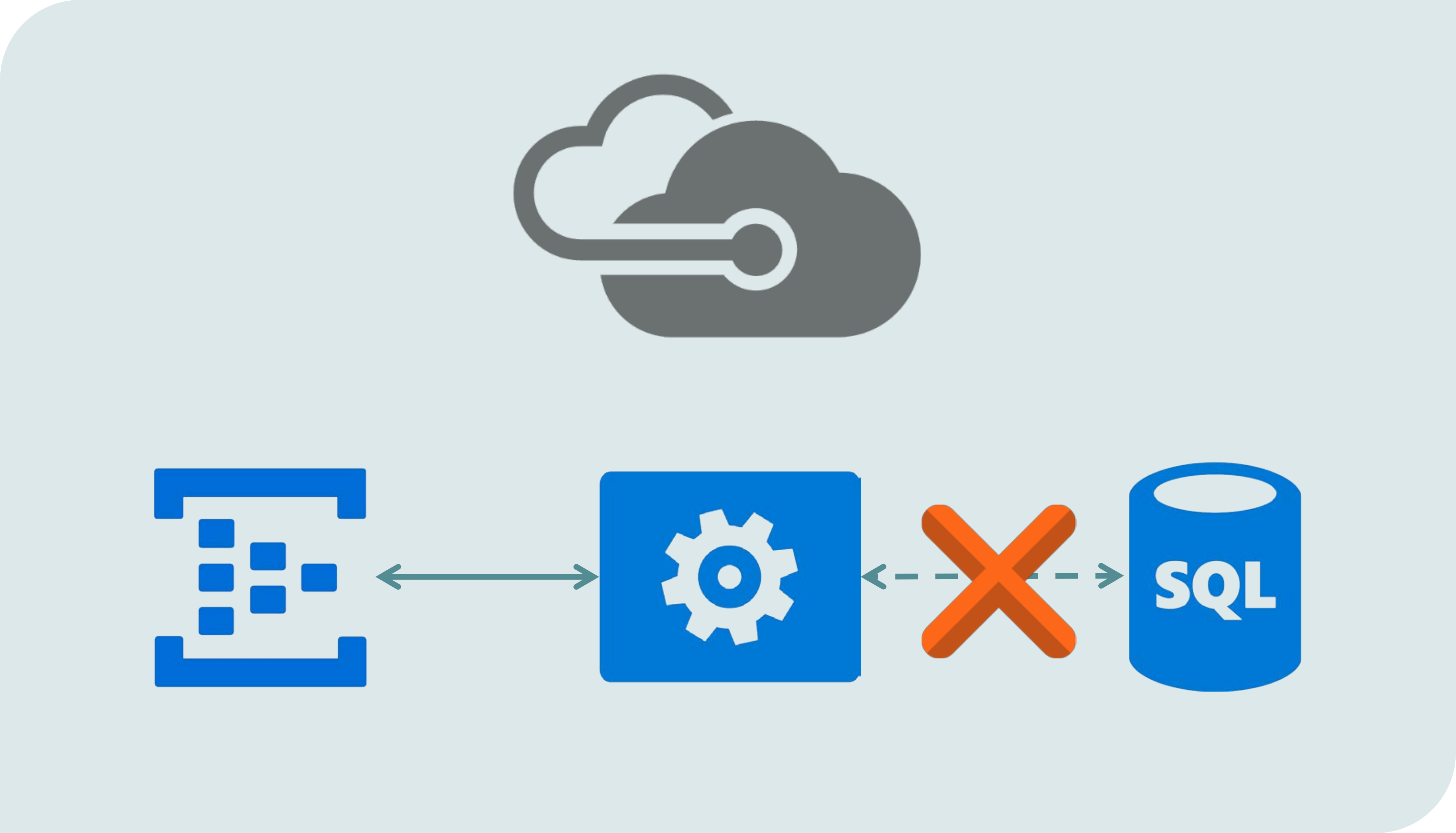
ModelBuffer<T>

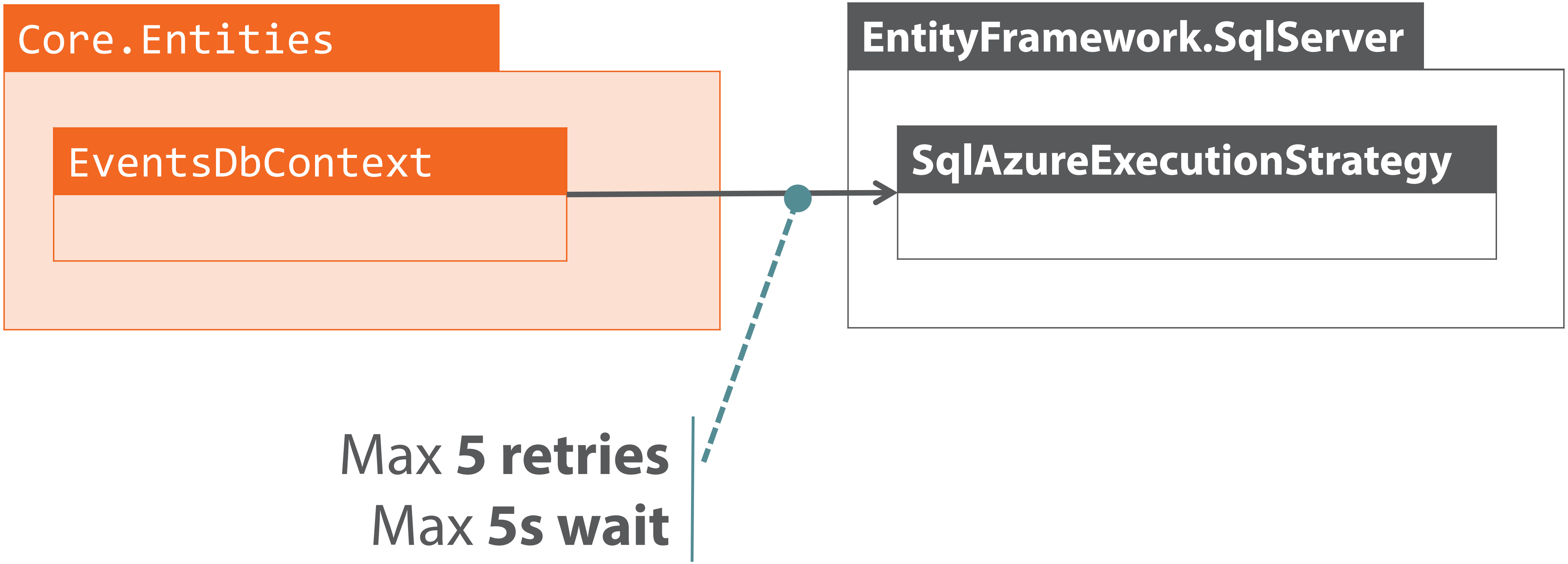
Add(T)

Flush(IEnumerable<T>)


10Mbps
SQL

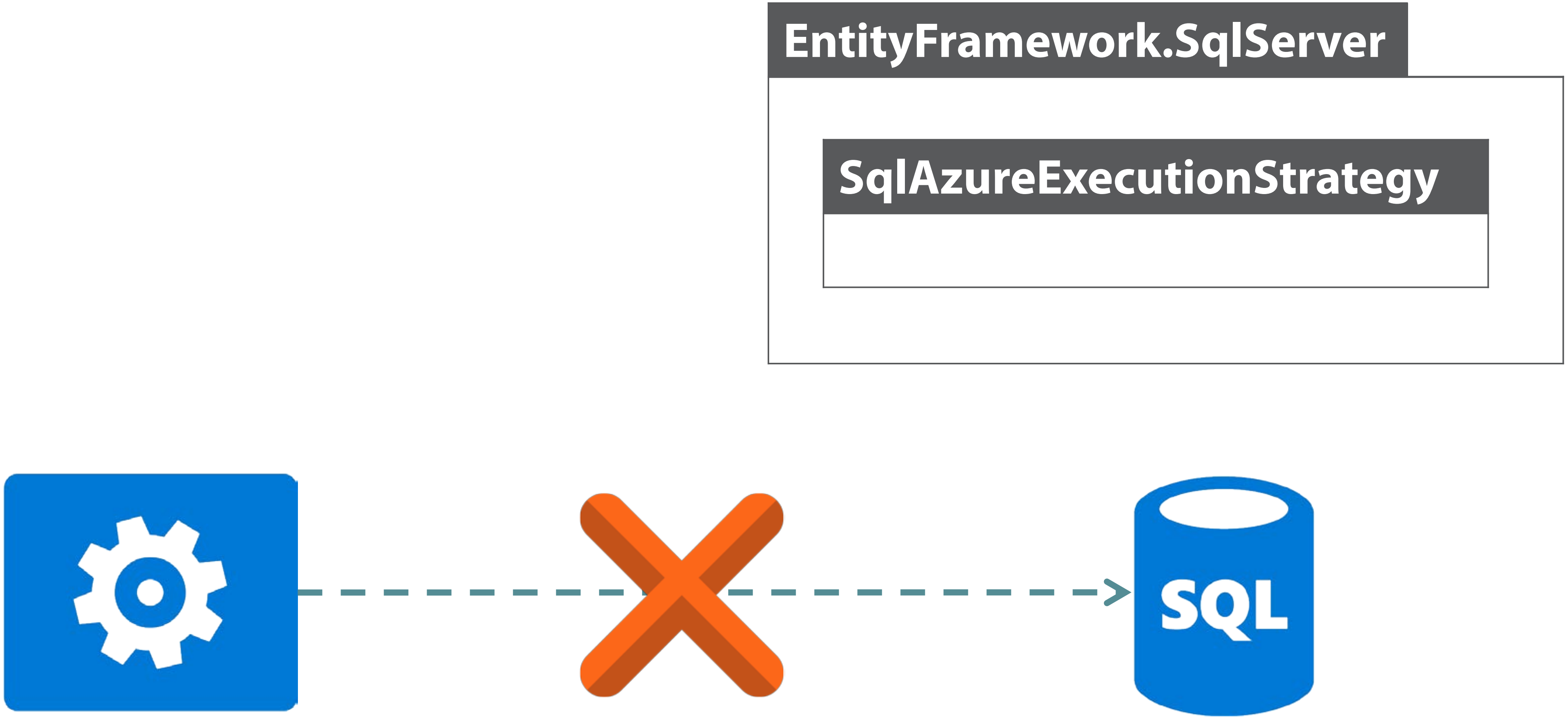

SQL
1Gbps


SQL



Max **5 retries**

Max **5s wait**

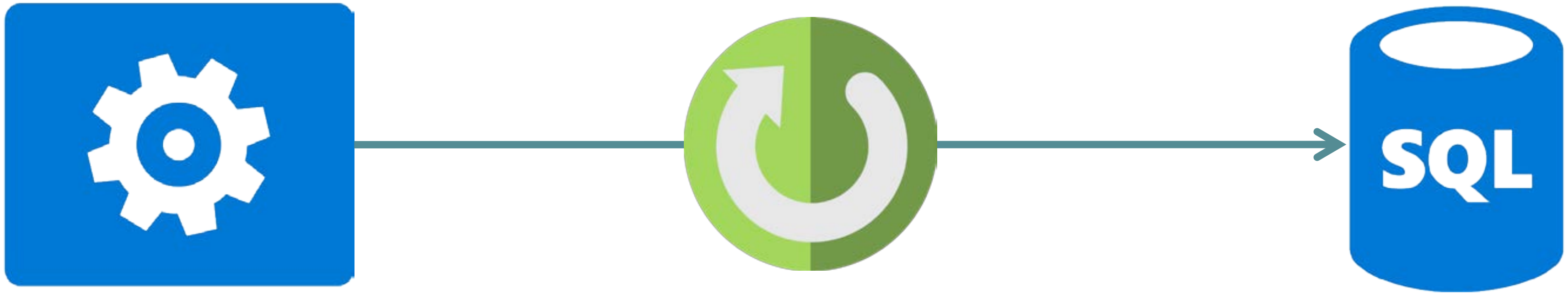


EntityFramework.SqlServer

SqlAzureExecutionStrategy

EntityFramework.SqlServer

SqlAzureExecutionStrategy



Demo: Running in Azure

Using SqlAzureExecutionStrategy

Script SQL Azure Deployment

Run RealTime worker in Azure

```
public EventsDbConfiguration()
{
   SetExecutionStrategy("System.Data.SqlClient", () =>
new SqlAzureExecutionStrategy(10, TimeSpan.FromSeconds(5)));
}
```

EventsDbConfiguration

Inherits DbConfiguration sets SqlAzureExecutionStrategy

```
<entityFramework codeConfigurationType=
"Telemetry.Entities.EventsDbConfiguration,Telemetry.Entities"
>
```

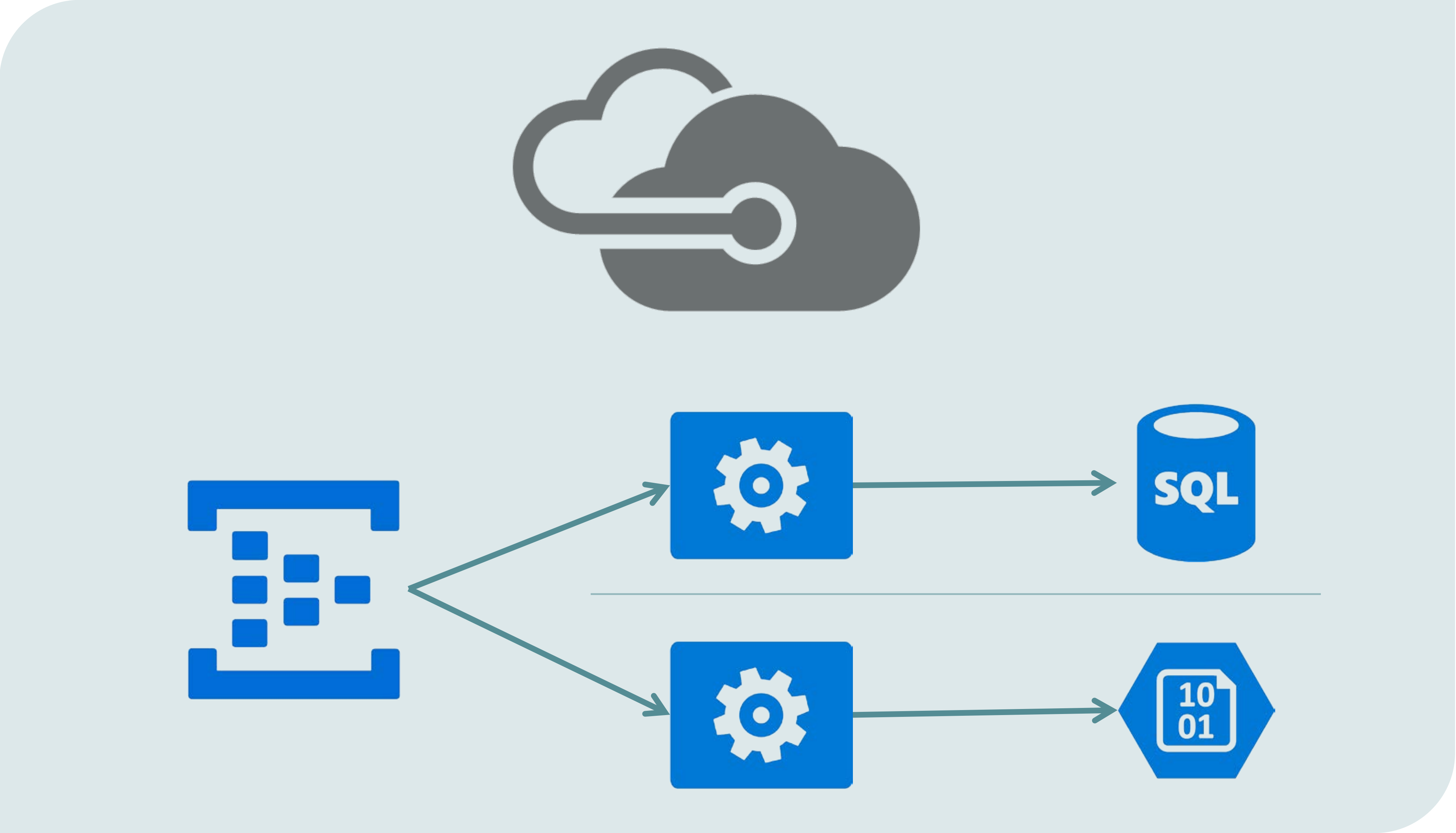
App.config

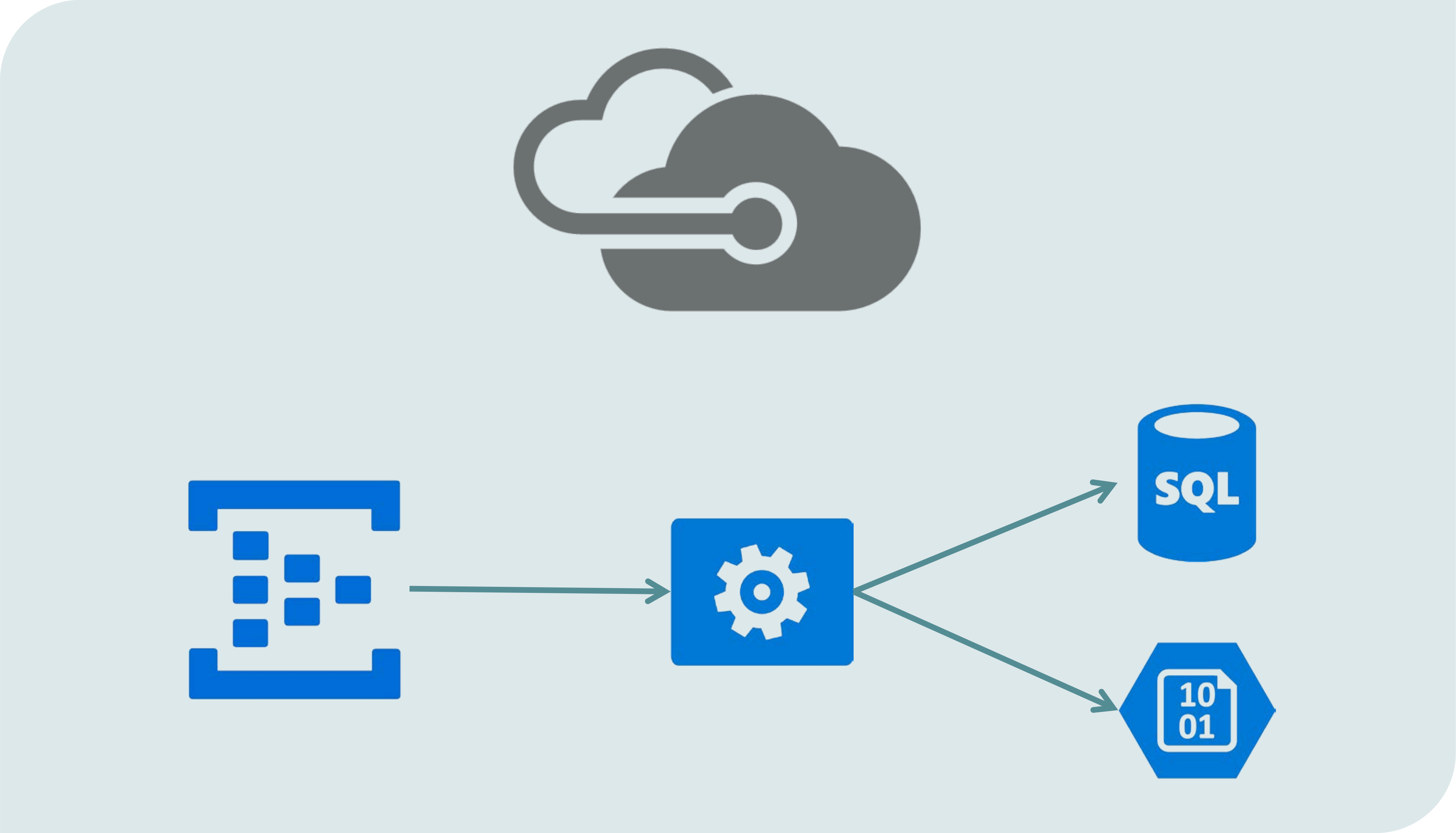
Specify codeConfigurationType for entityFramewrok

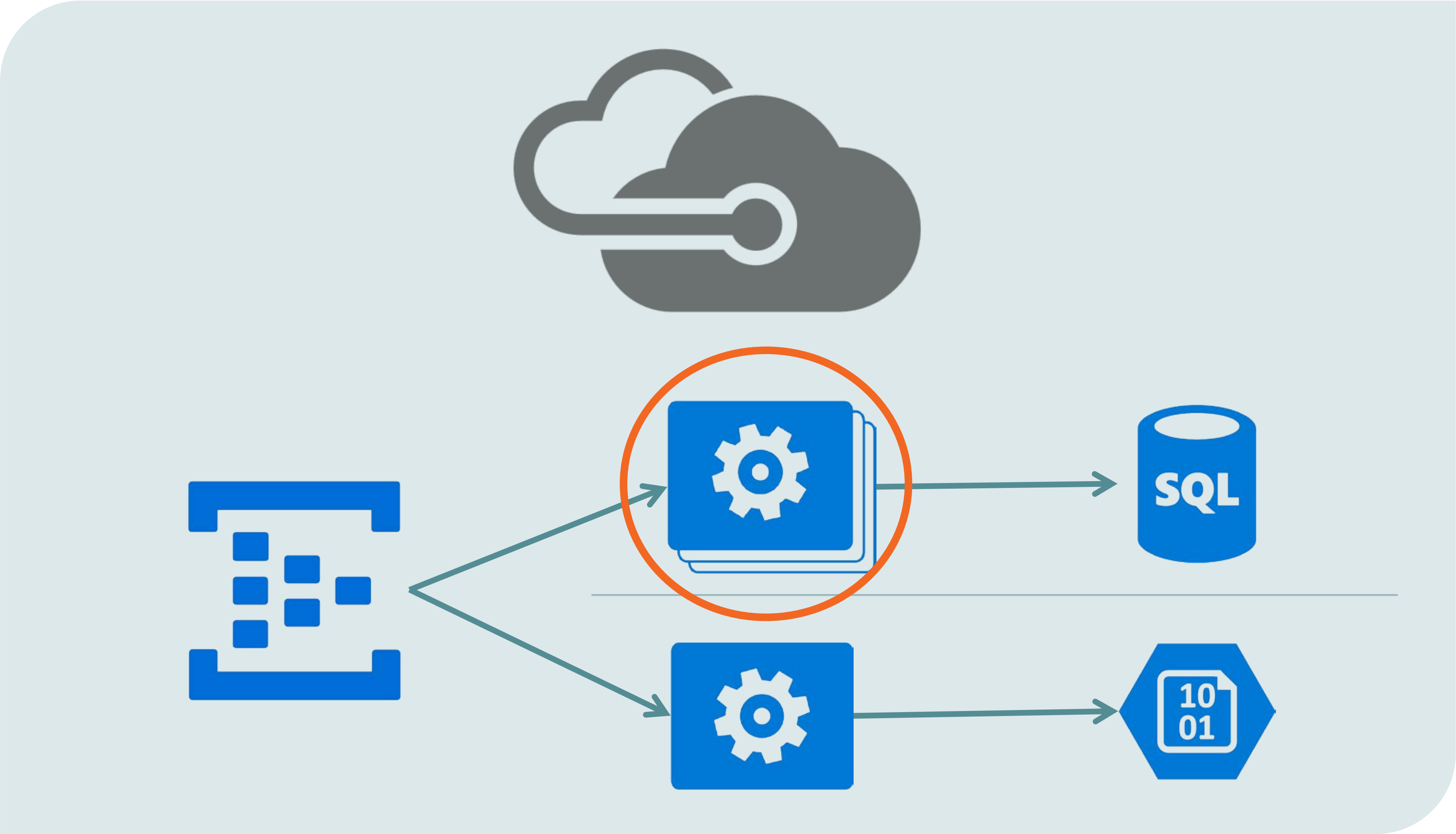
```
New-AzureSqlDatabaseServer -AdministratorLogin $AdminUser ...

New-AzureSqlDatabase -DatabaseName 'eventsdb-prd' ...

& "...\SqlPackage.exe" /SourceFile:"$DacPacPath"
  /TargetConnectionString:"$connectionString" /Action:Publish
```

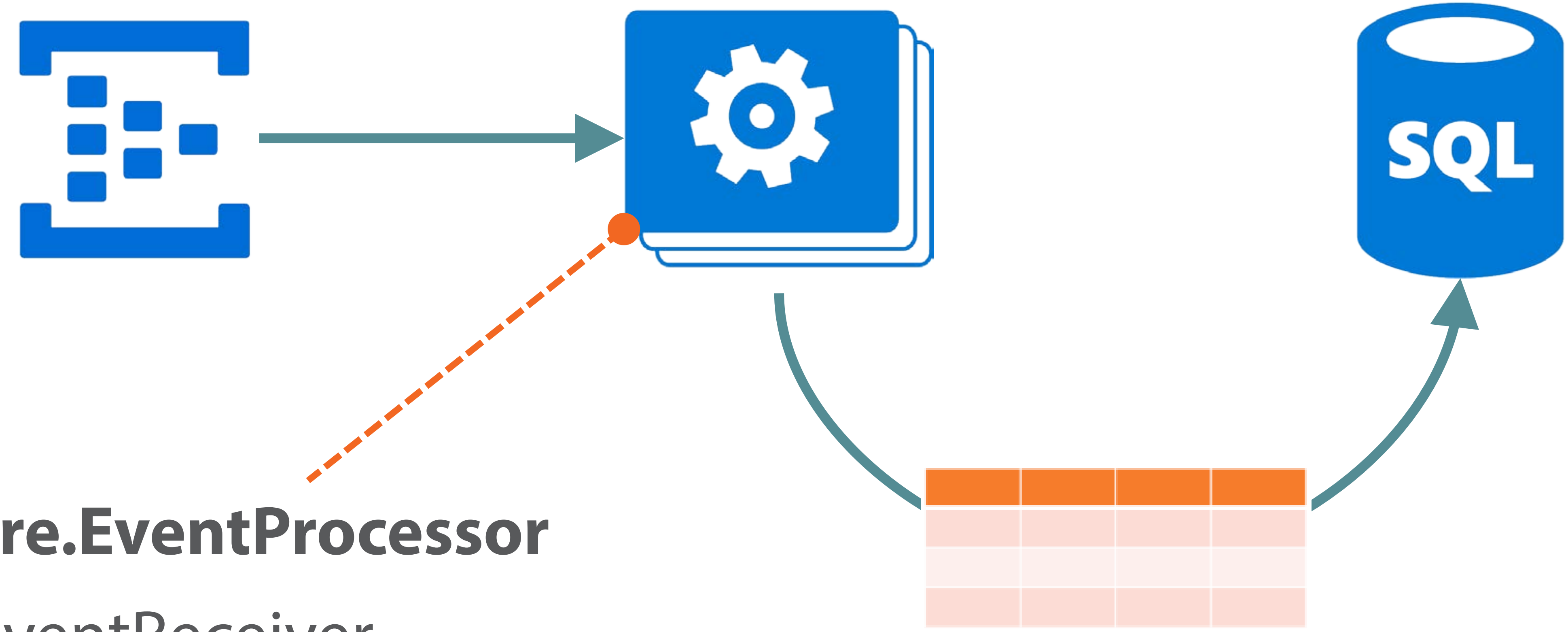
Publish to SQL Azure

PowerShell Azure cmdlets and SqlPackage tool


SQL
10
01

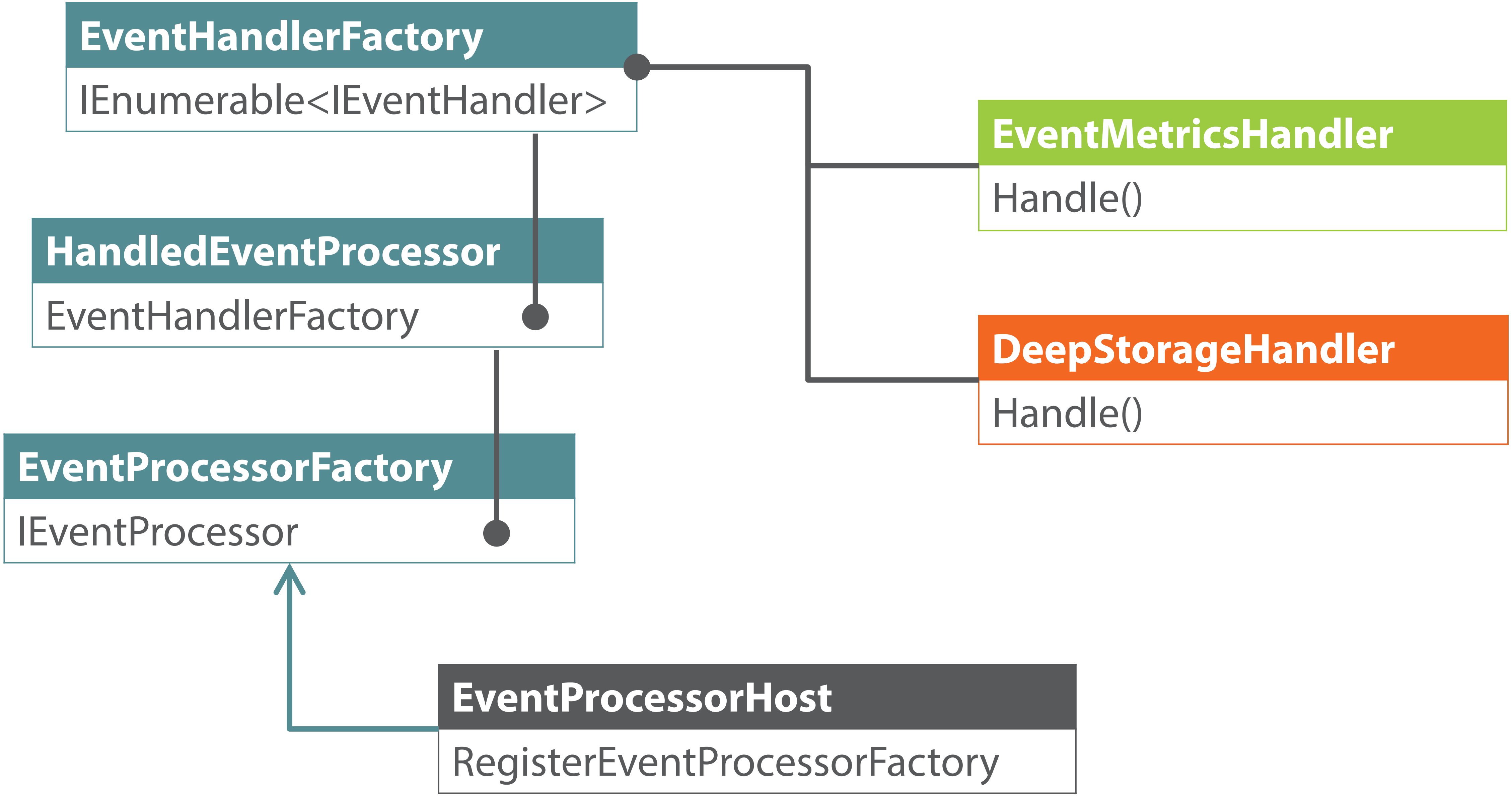

SQL
10
01


SQL
10
01



Core.EventProcessor

- EventReceiver
- HandledEventProcessor
- IEventHandler



EventHandlerFactory

IEnumerable<IEventHandler>

EventMetricsHandler

Handle()

HandledEventProcessor

EventHandlerFactory

DeepStorageHandler

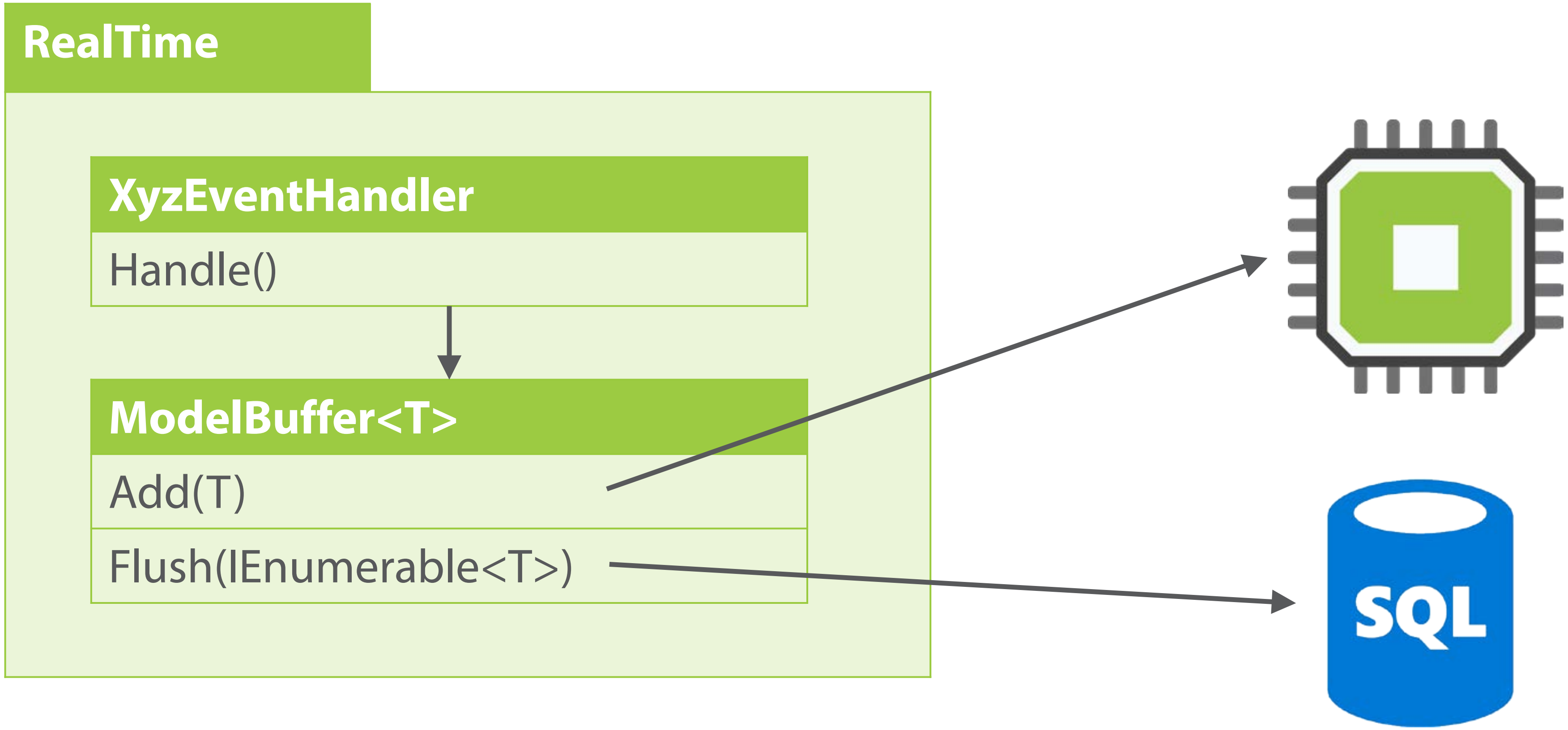
Handle()

EventProcessorFactory

IEventProcessor

EventProcessorHost

RegisterEventProcessorFactory



RealTime

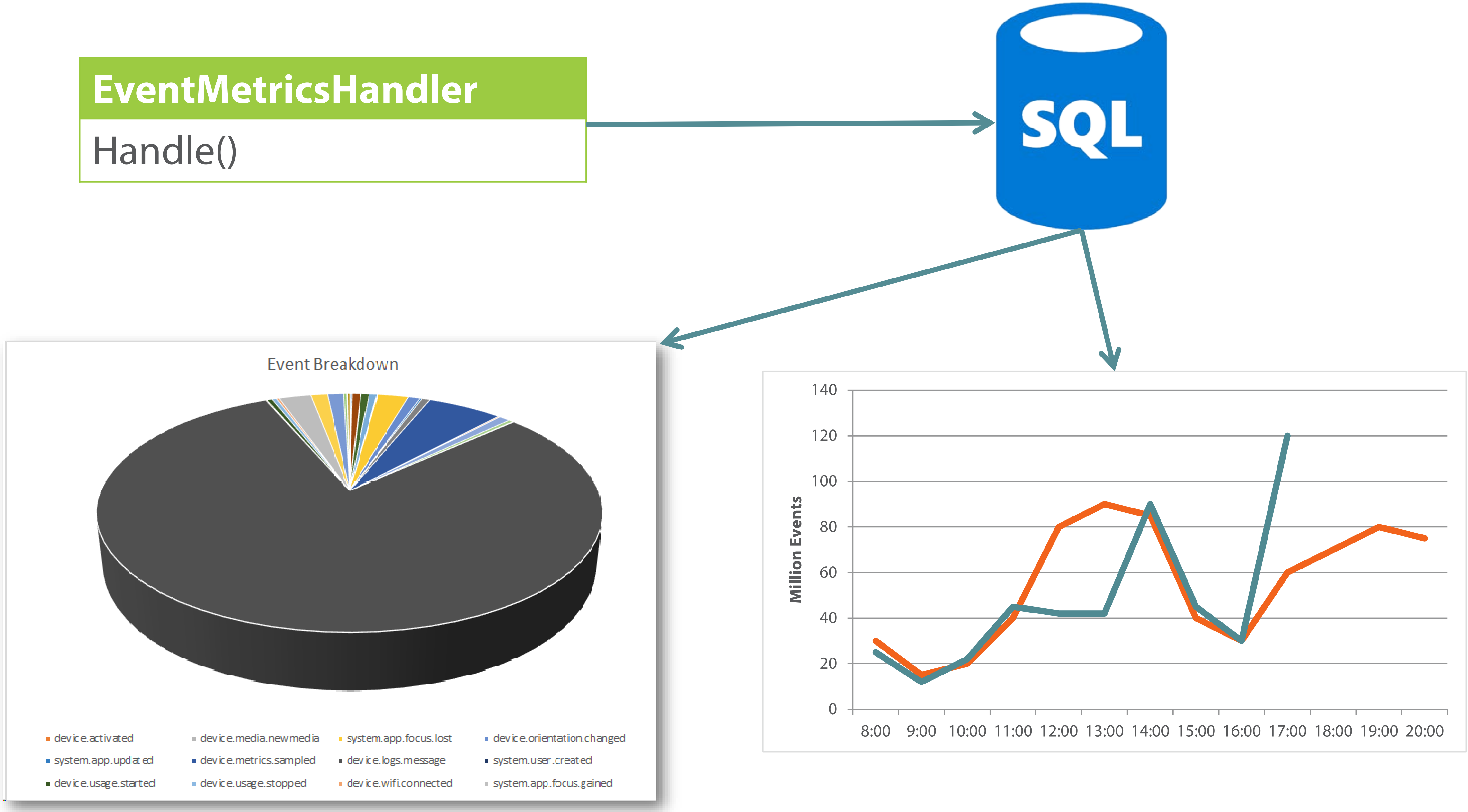
XyzEventHandler

Handle()

ModelBuffer<T>

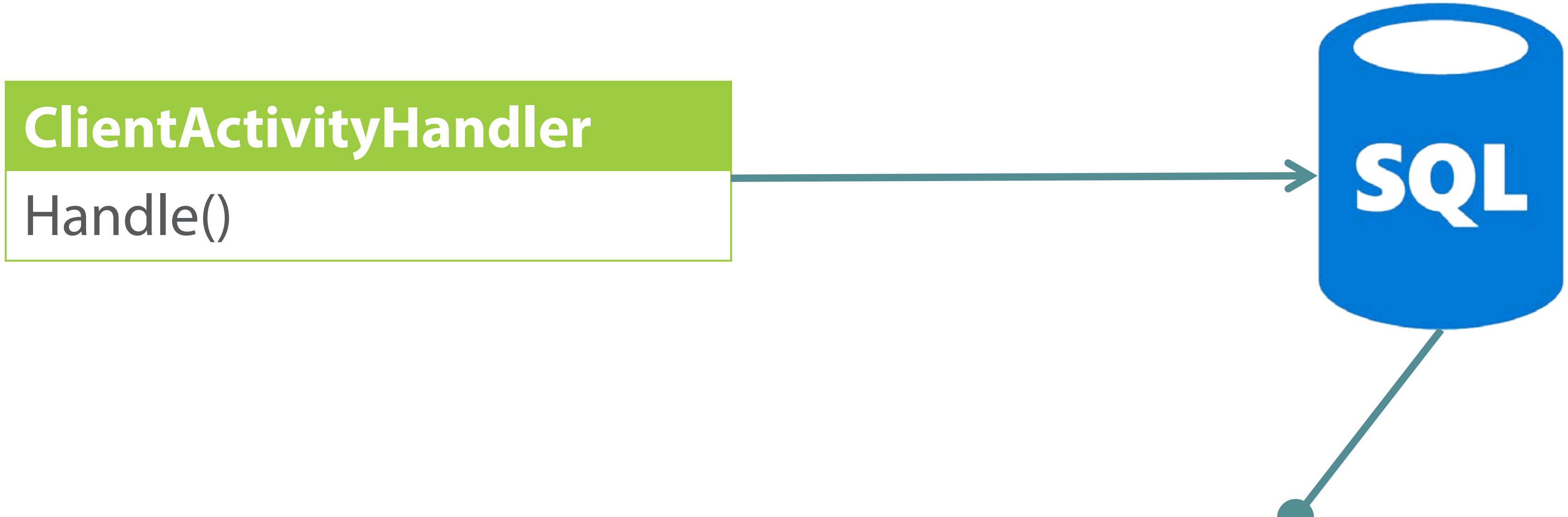
Add(T)

Flush(IEnumerable<T>)

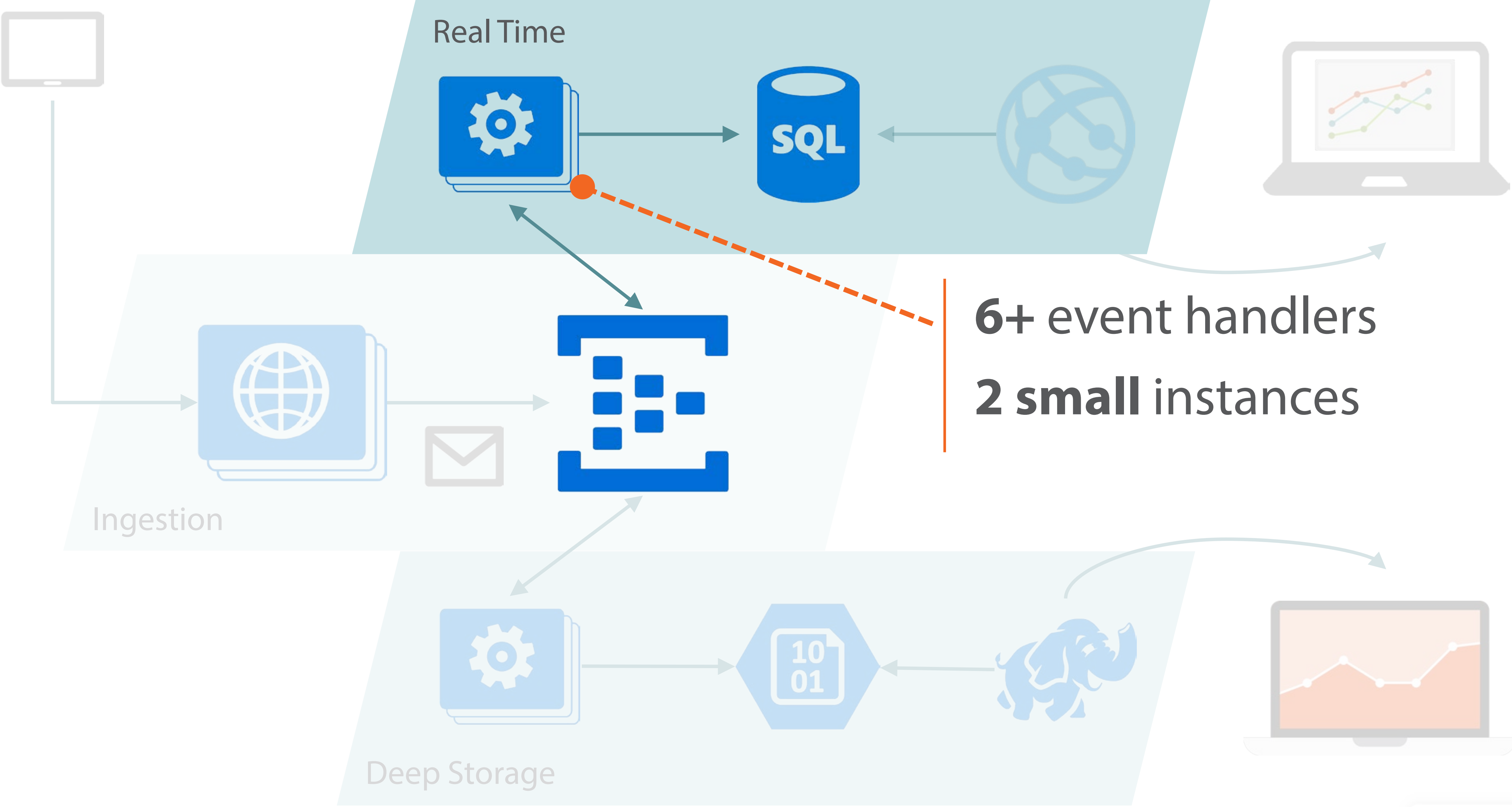


EventMetricsHandler

Handle()

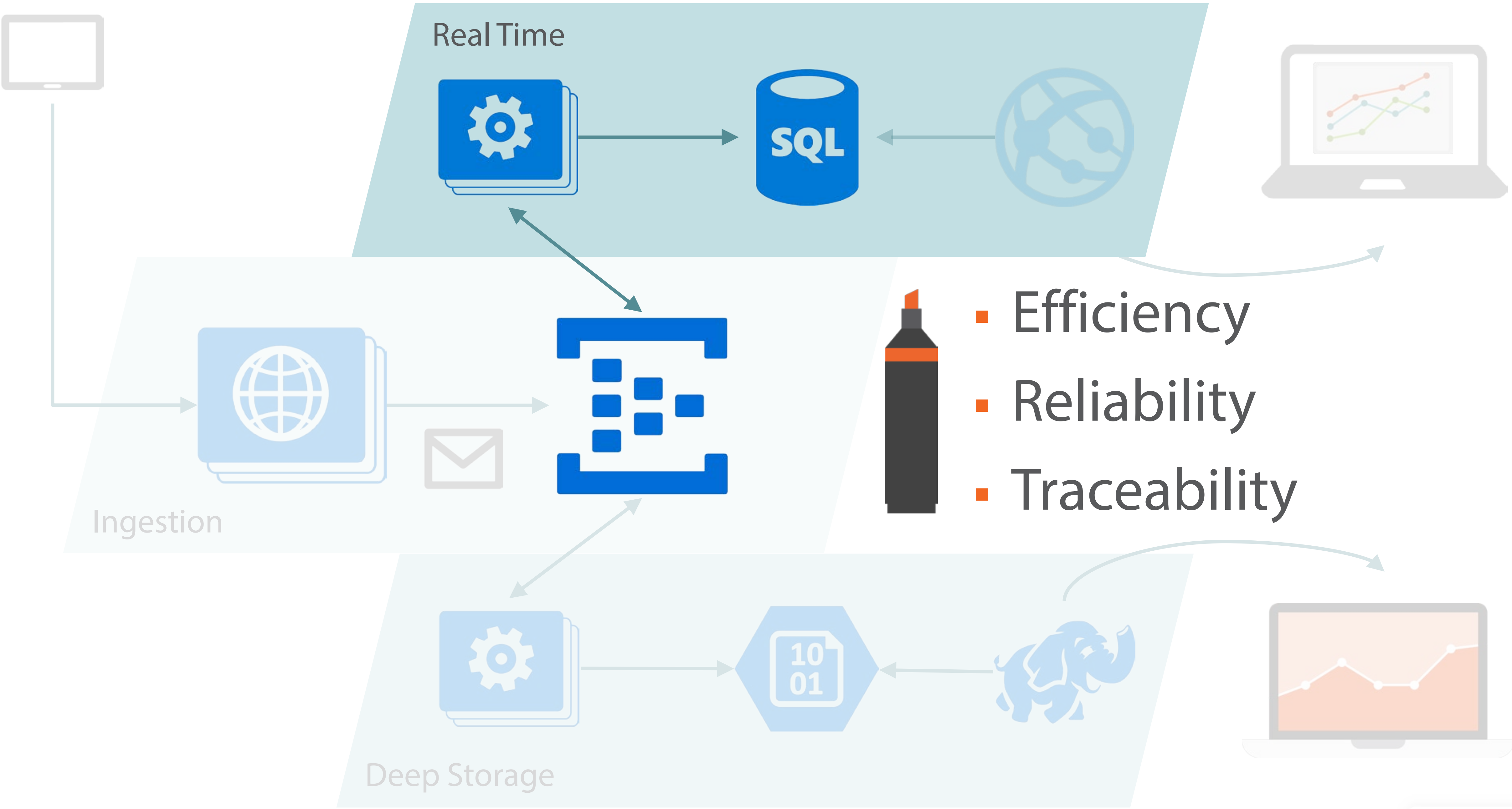


DeviceId	TotalUsage	UsageThisWeek	UsageThisMonth	UsageThisYear
a6b56d5b64	6,791	520	1,224	6,791
cb4375f2aa	11,203	1,604	4,376	9,543
bace33211e	102	102	102	102
dff1344fee	11	1	10	11

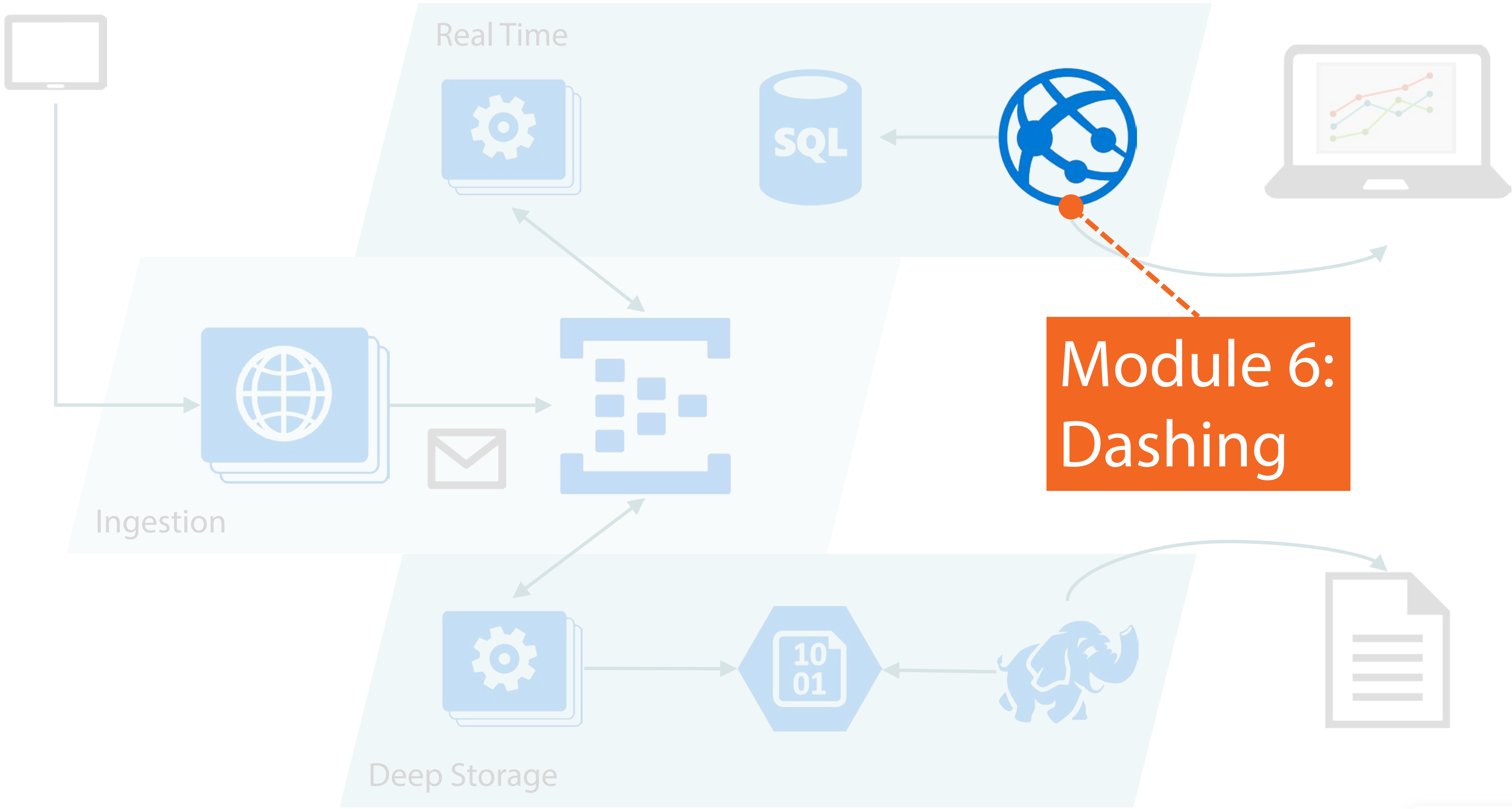


6+ event handlers

2 small instances



- Efficiency
- Reliability
- Traceability



Module 6: Dashing