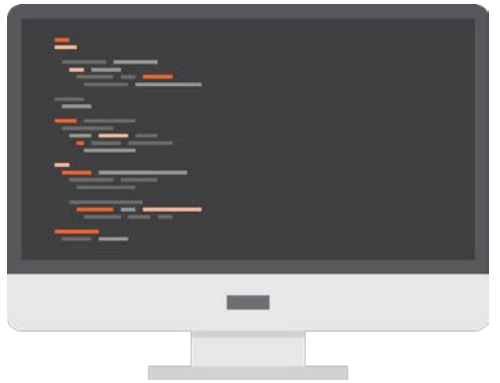
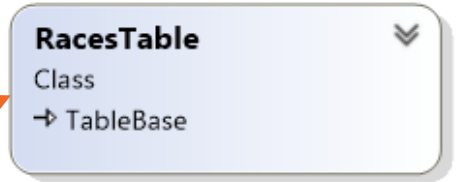
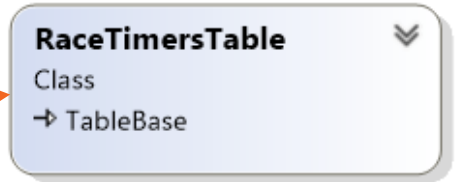
HBase Deep Dive

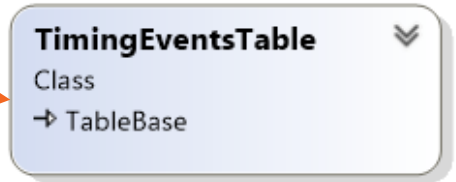
Elton Stoneman

@EltonStoneman | blog.sixeyed.com









Table

Region 1

Region 2

CF1

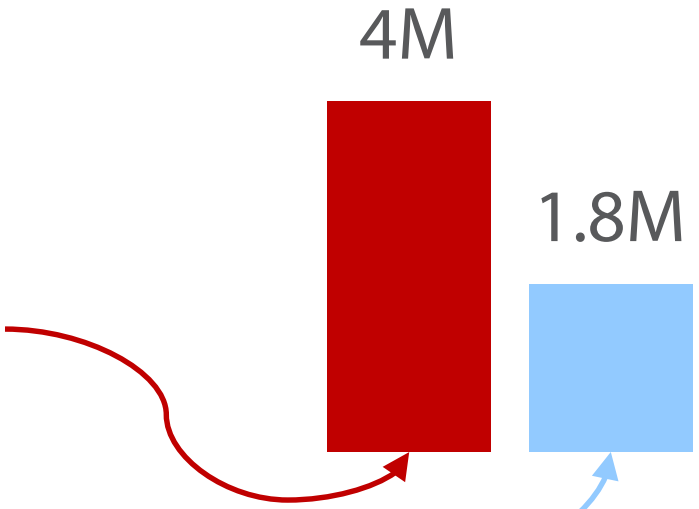
CF2

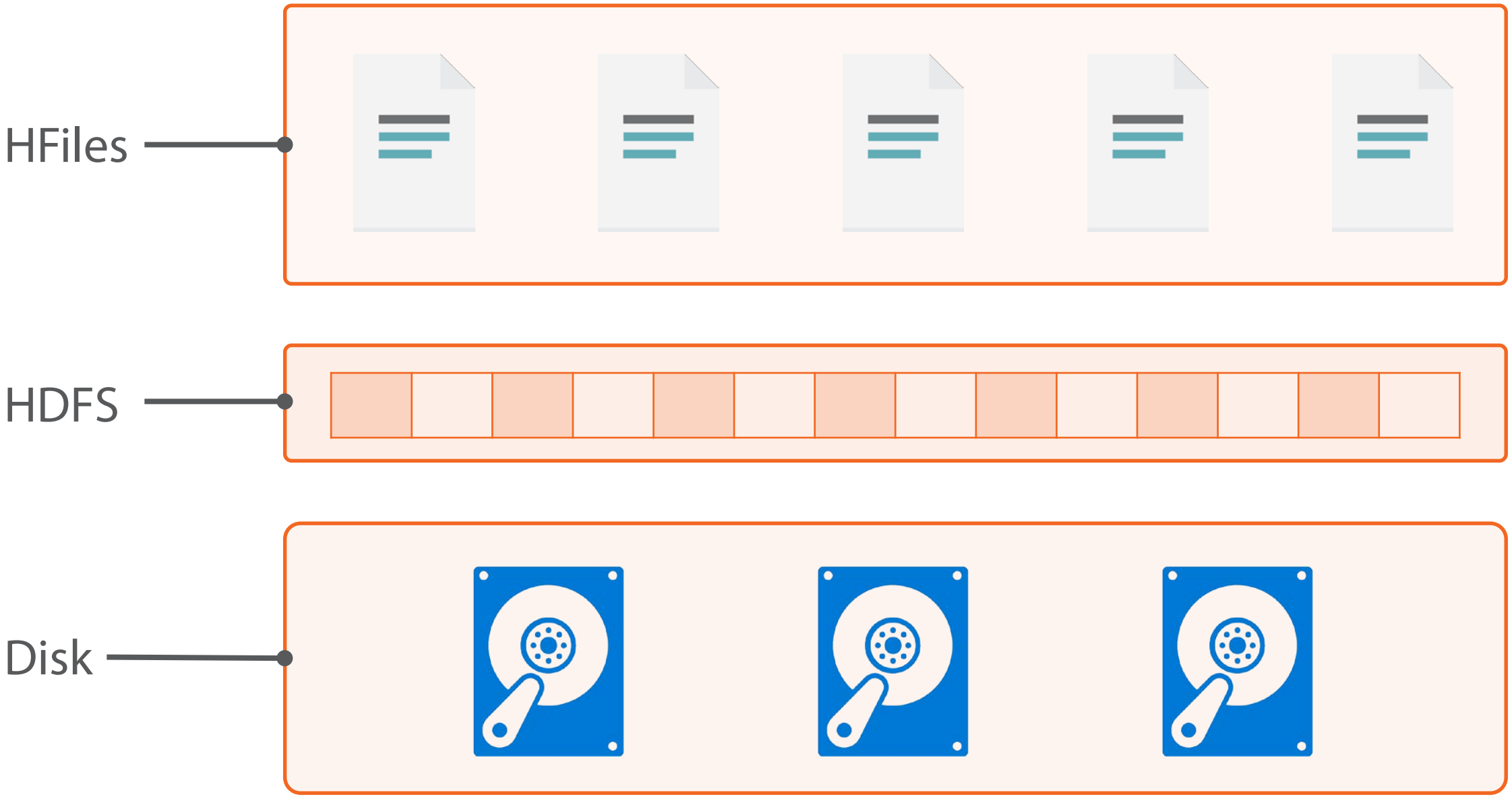
CF1

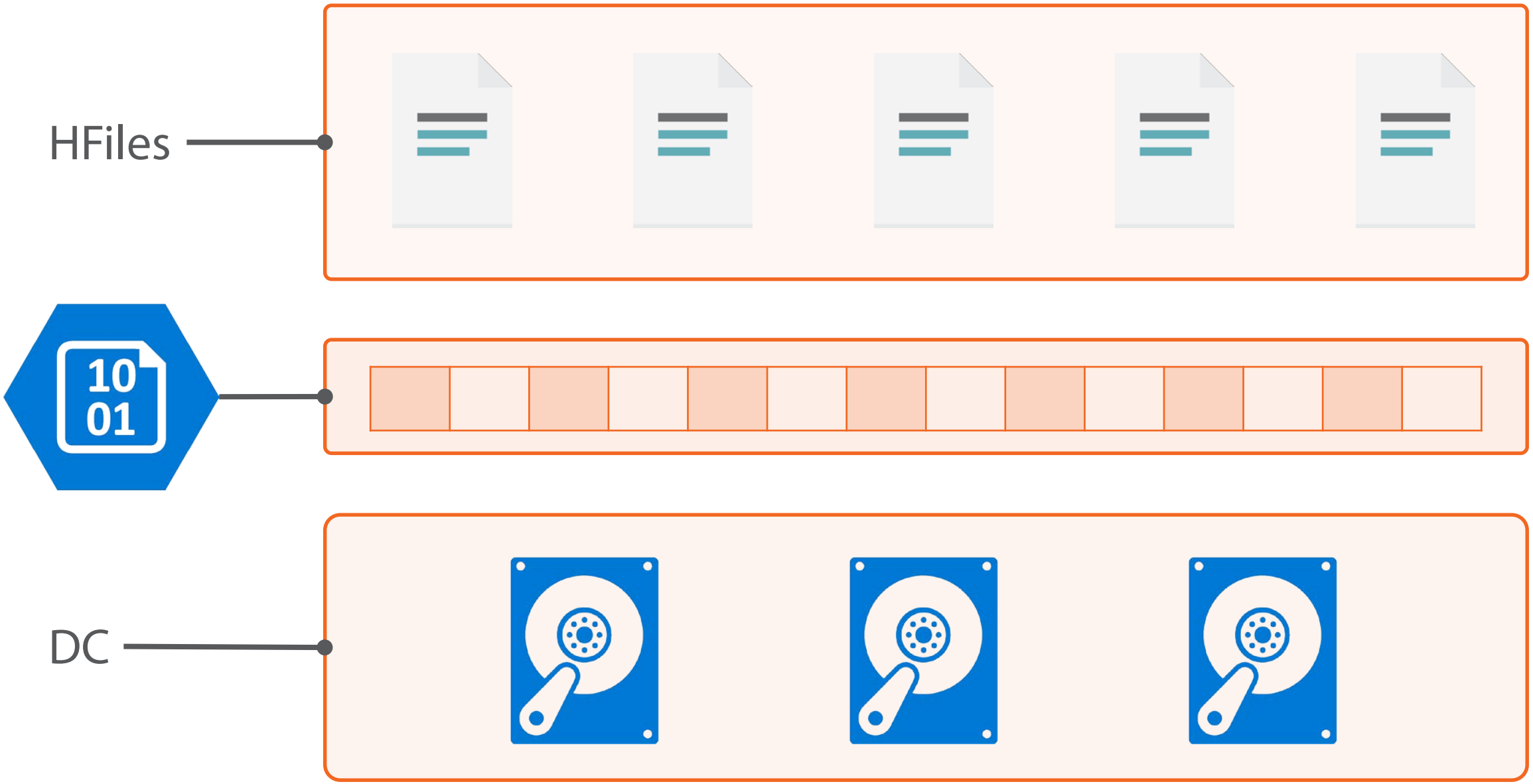
CF2

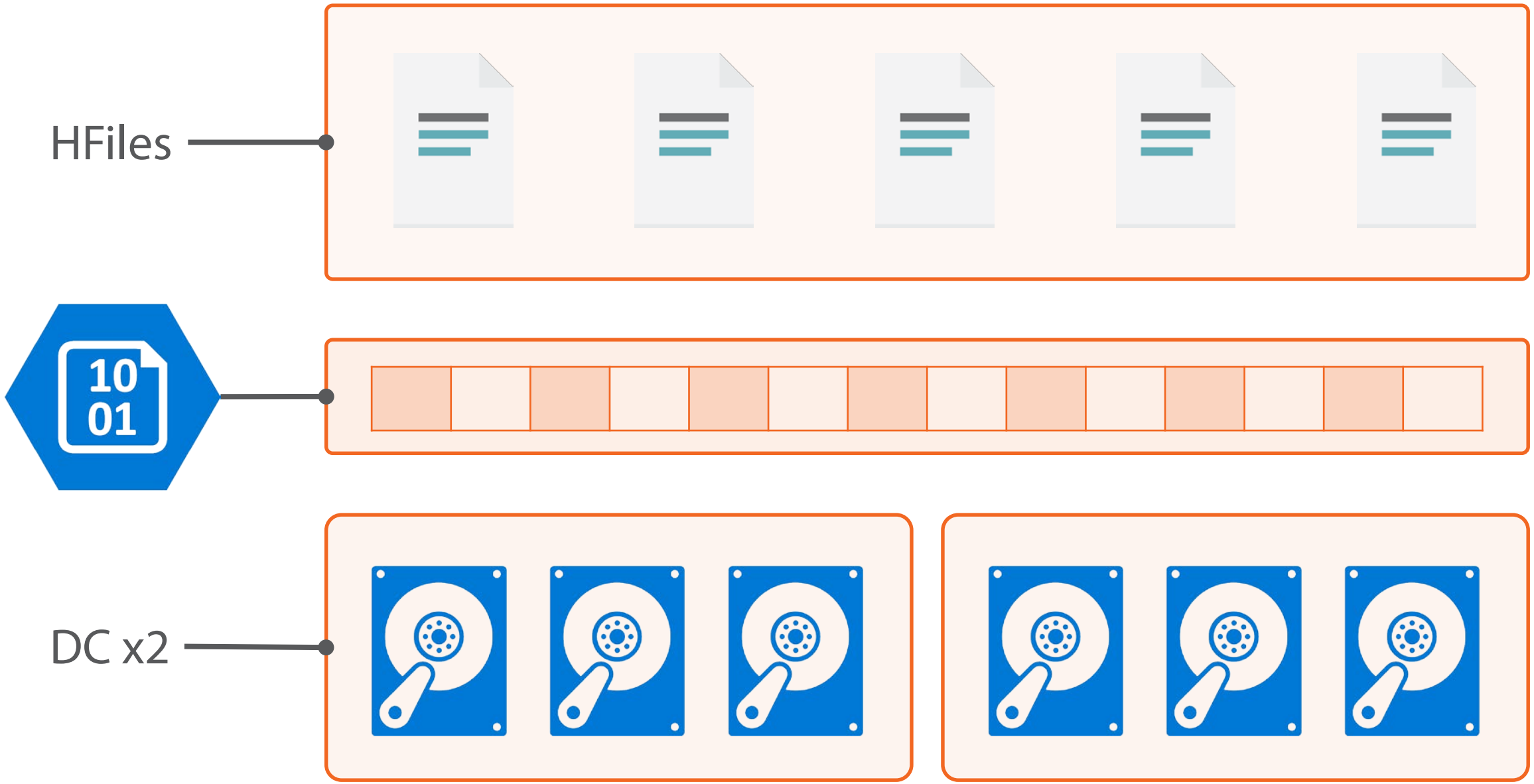
Node	Regions	Req/sec
workernode0	3	630
workernode1	3	1,205
workernode2	3	685
workernode3	3	403

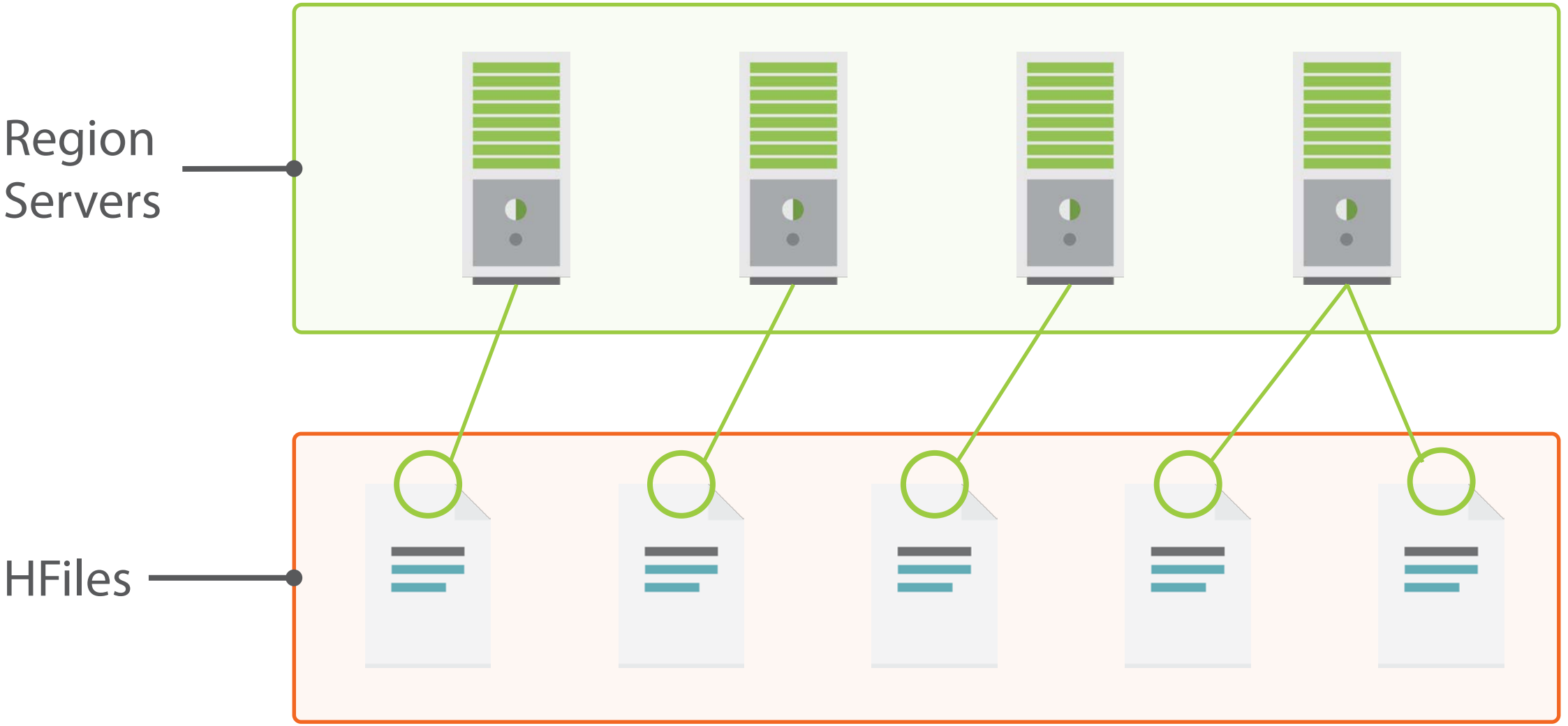
APACHE HBASE

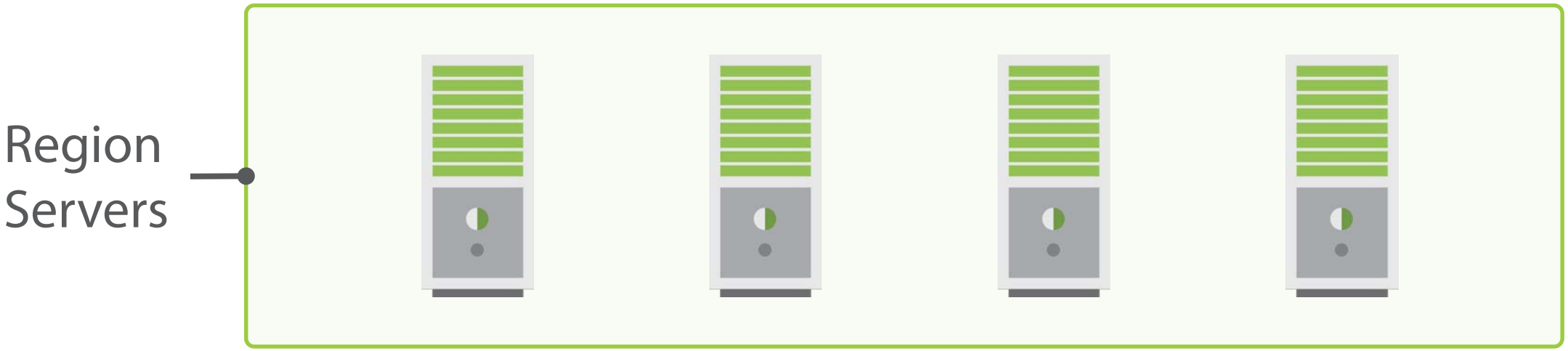


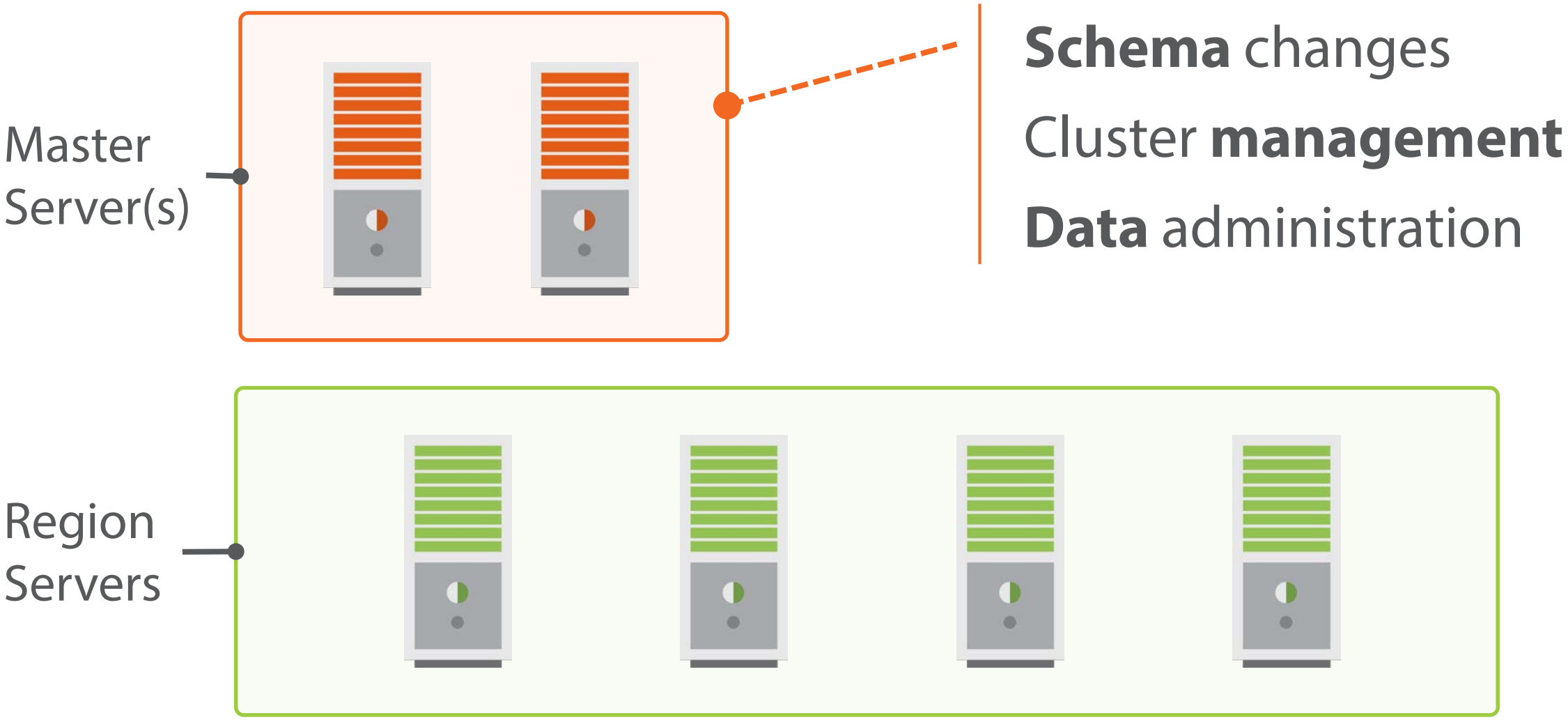

HFiles
HDFS
Disk

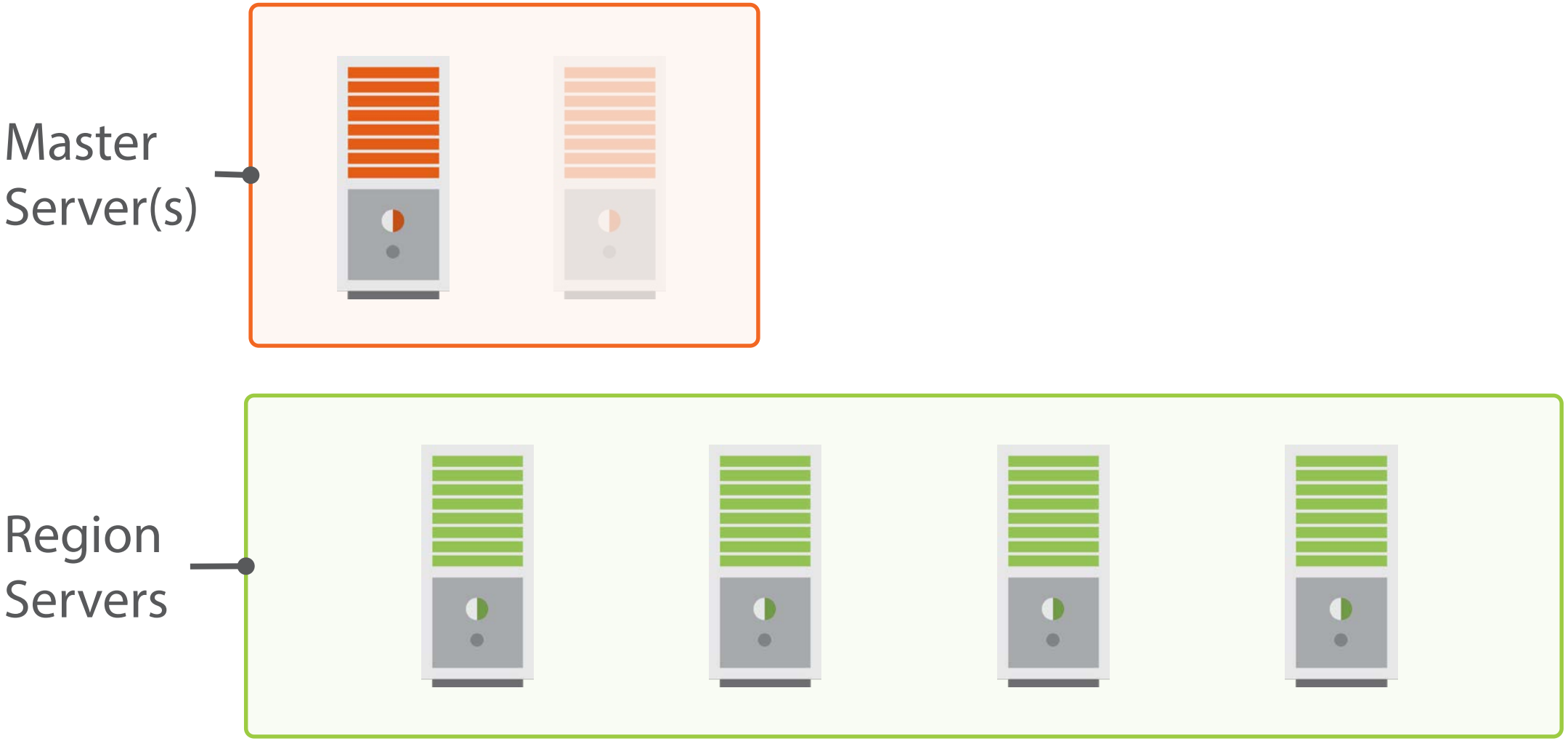


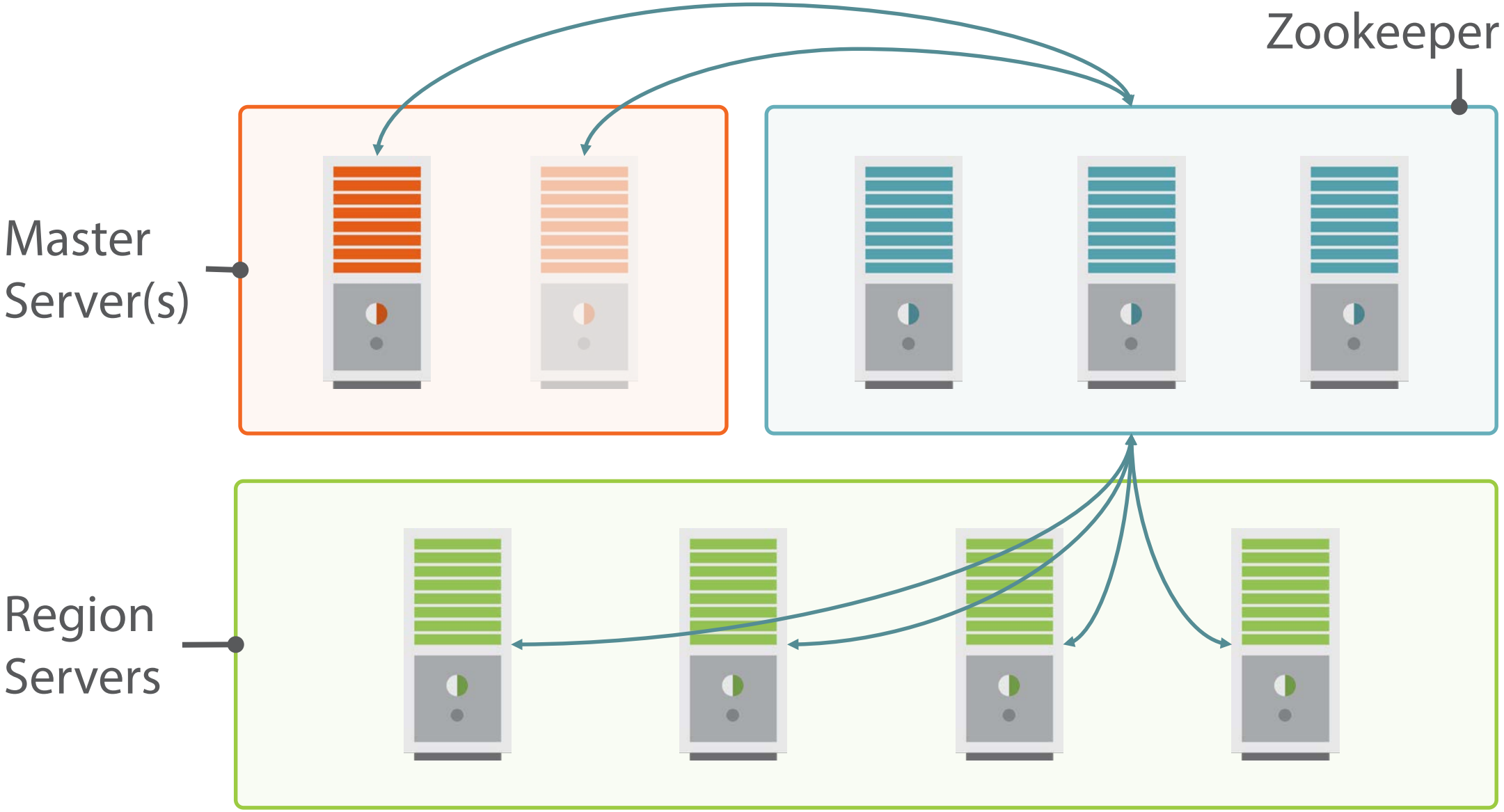

HFiles
10
01
DC x2

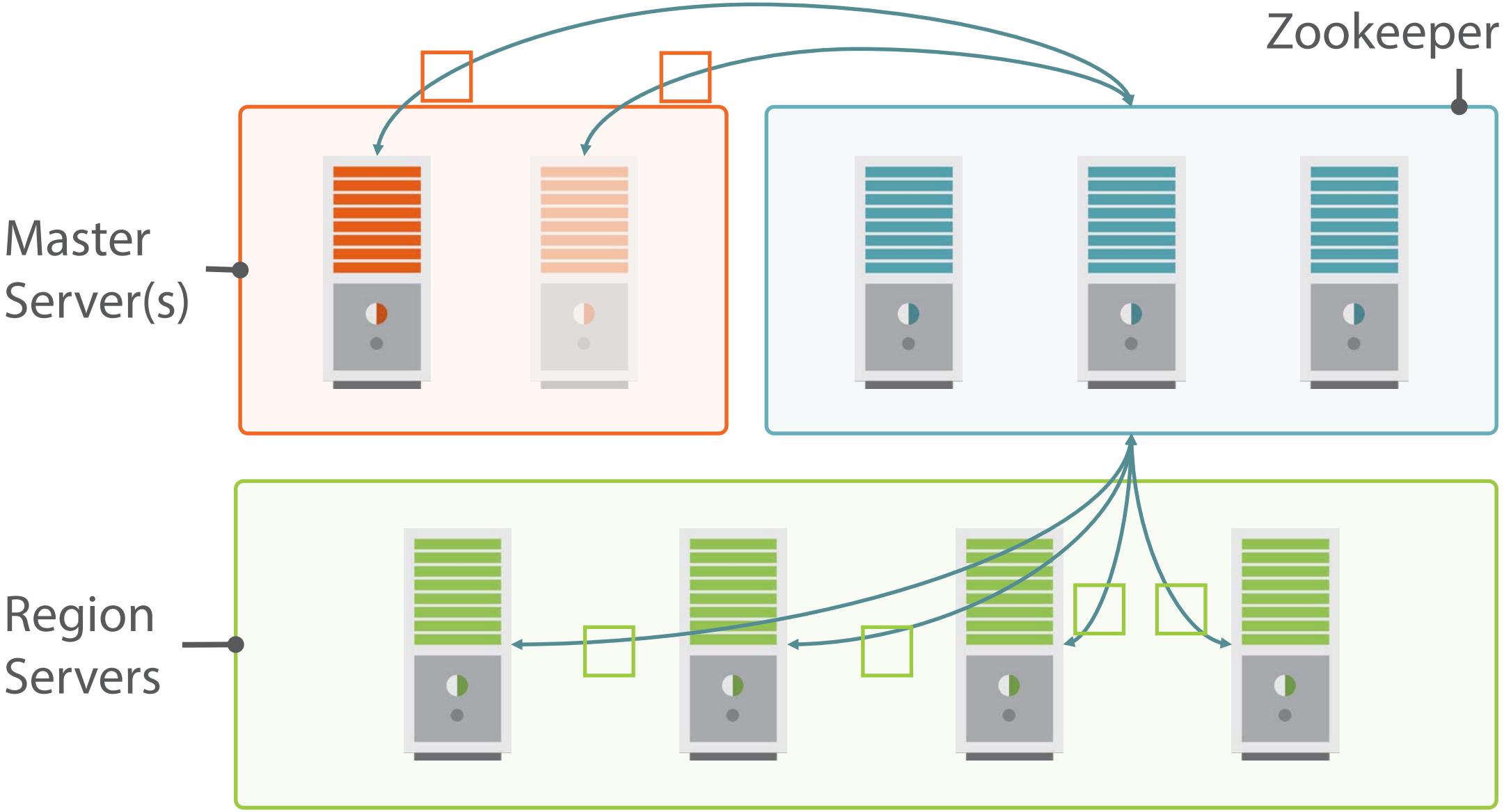


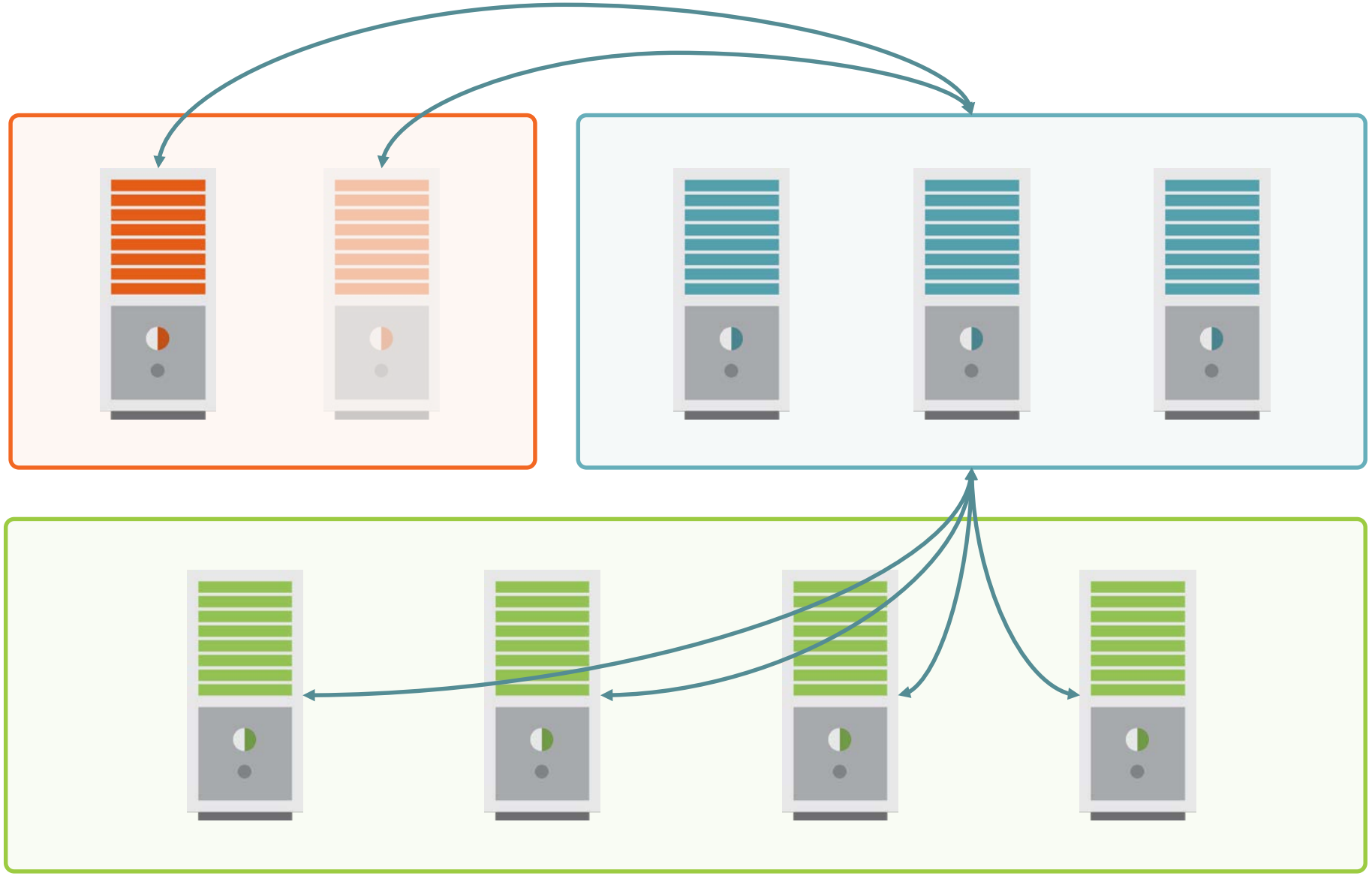


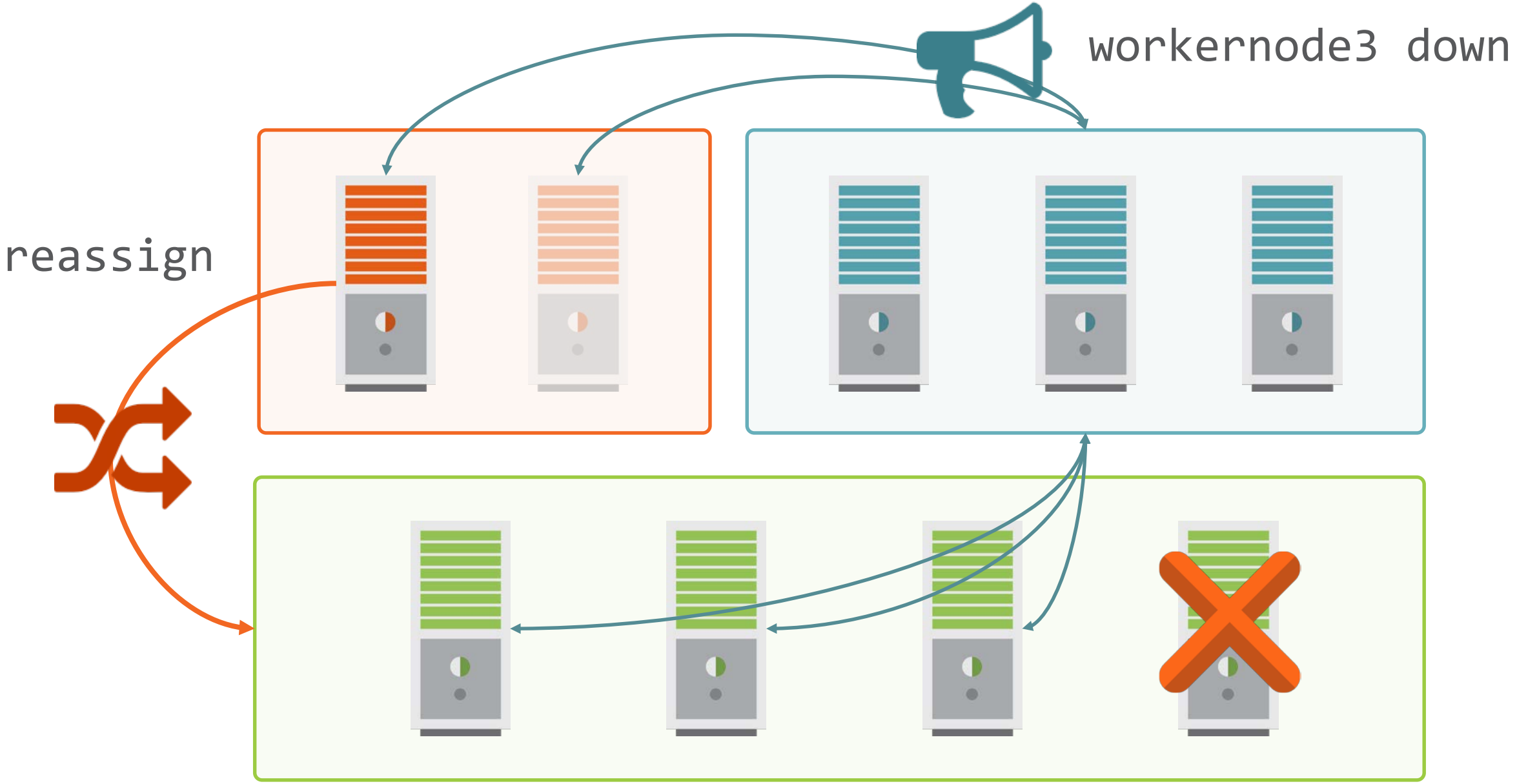


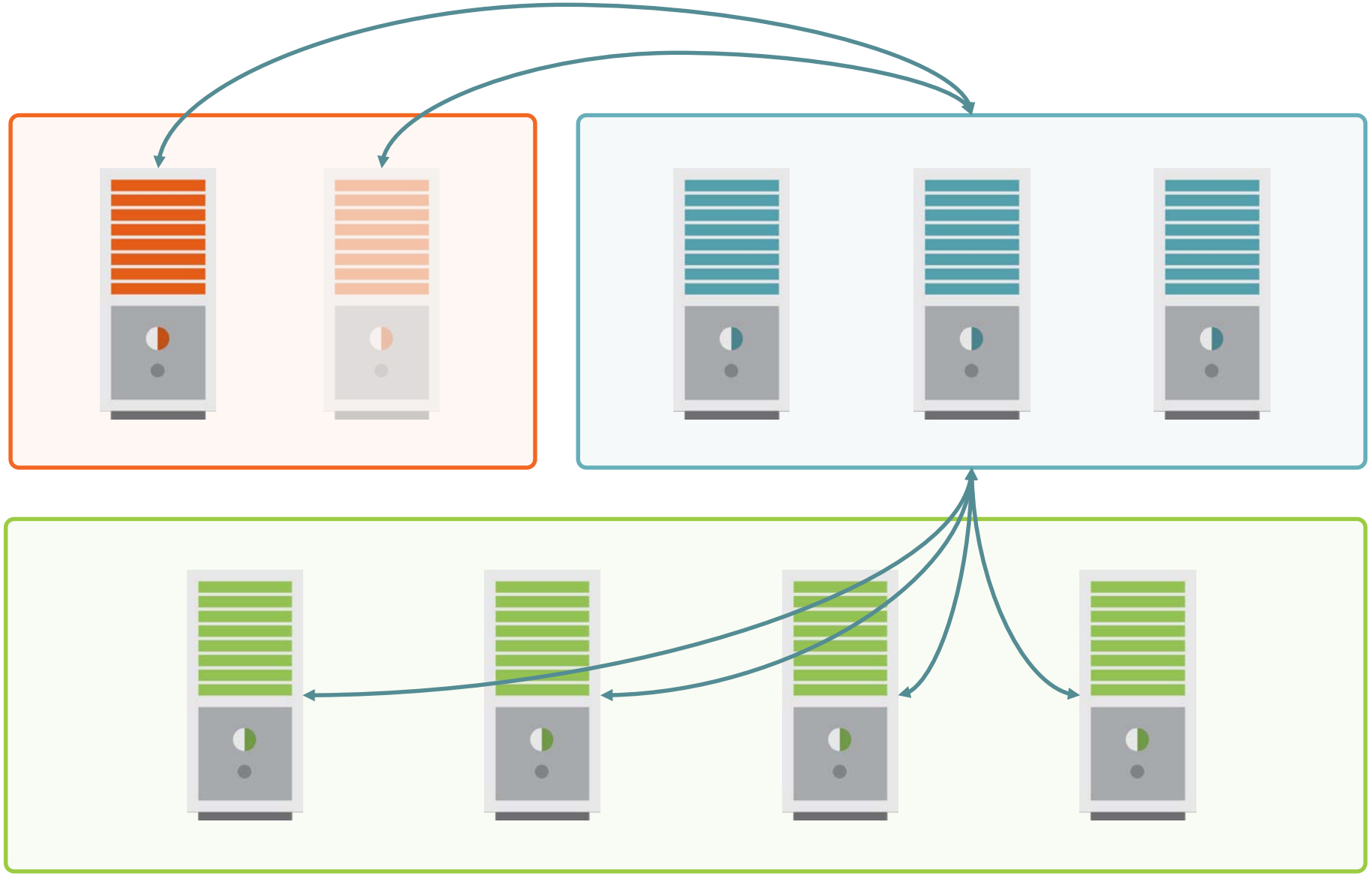


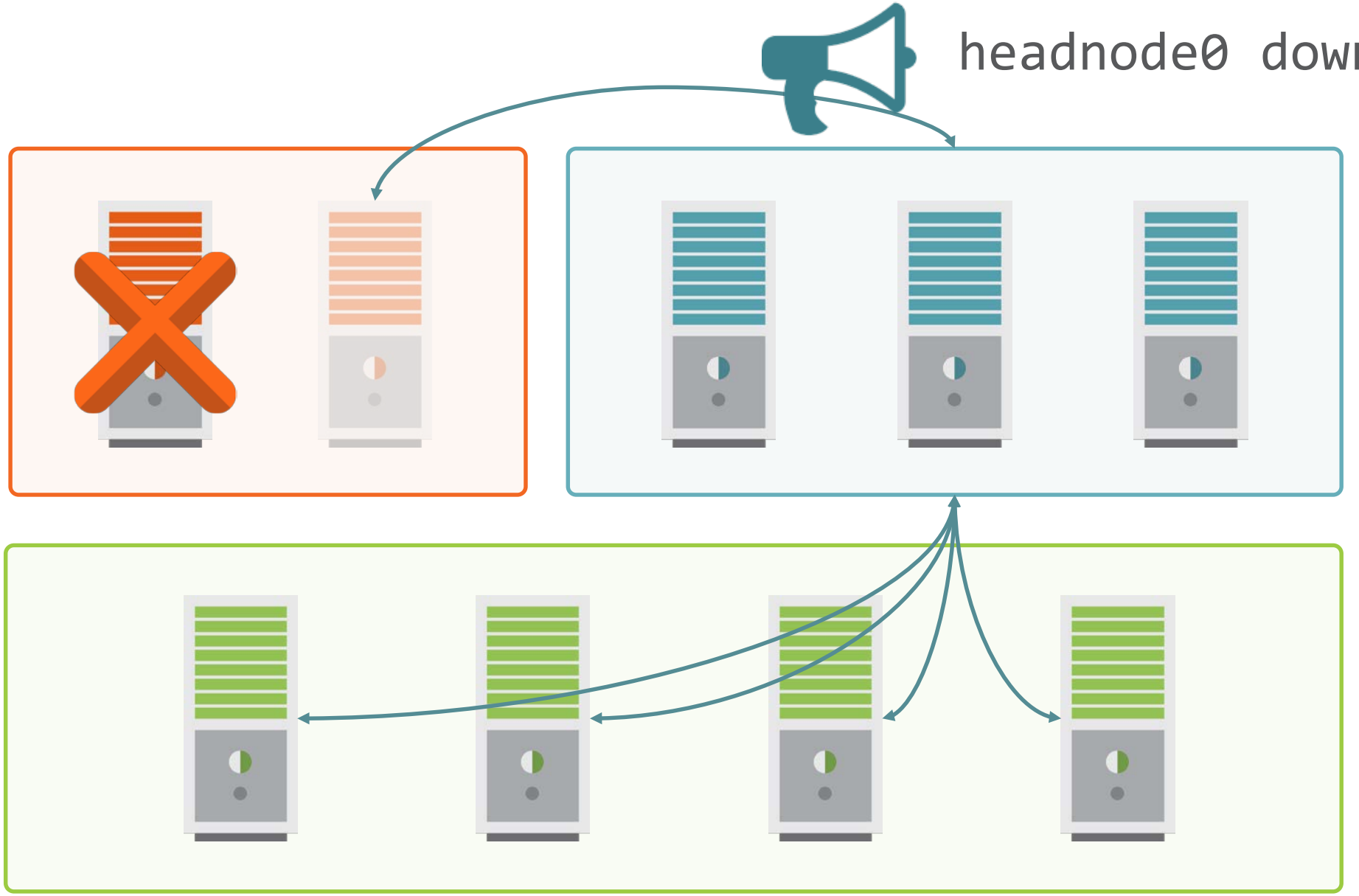


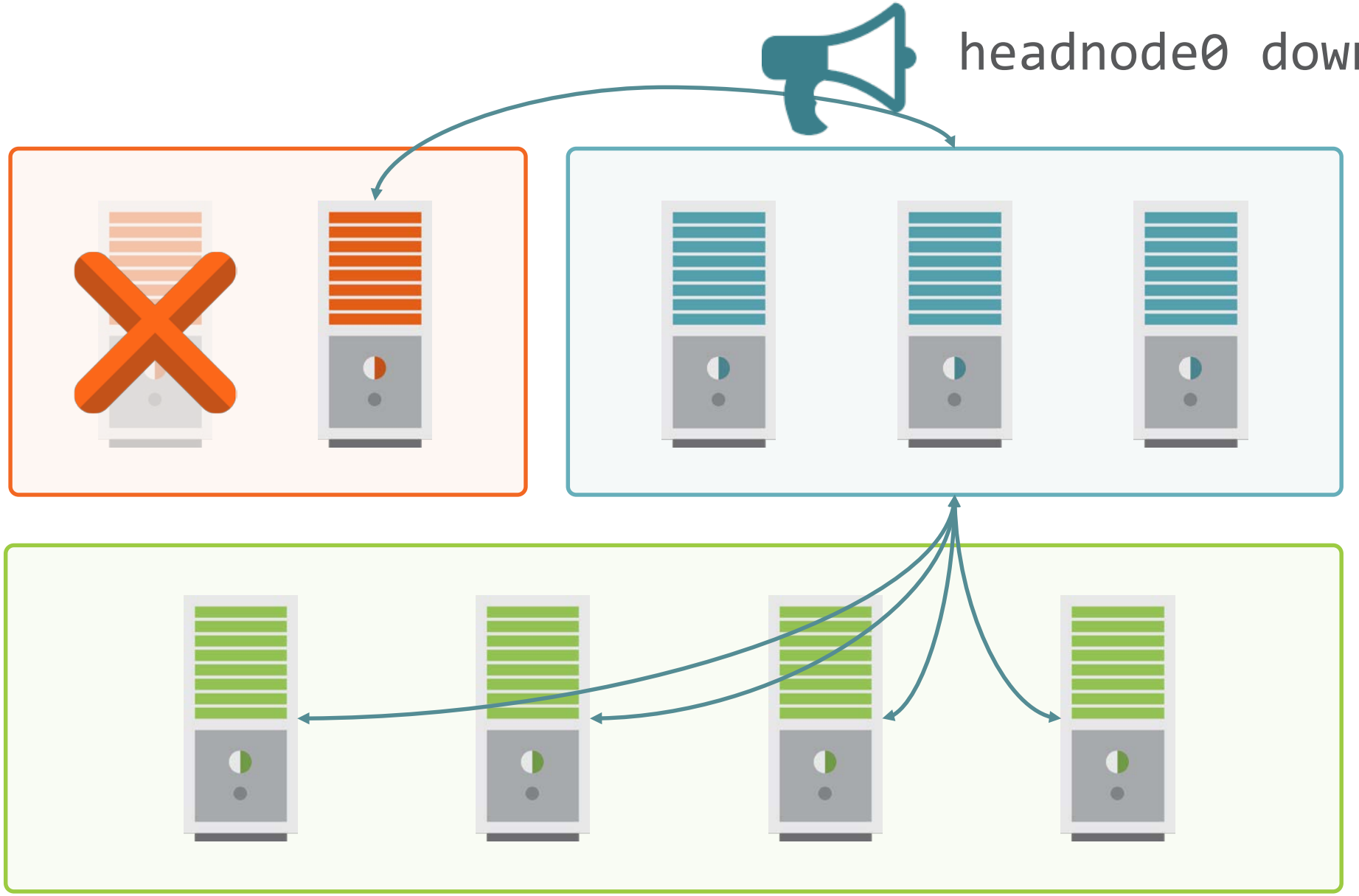






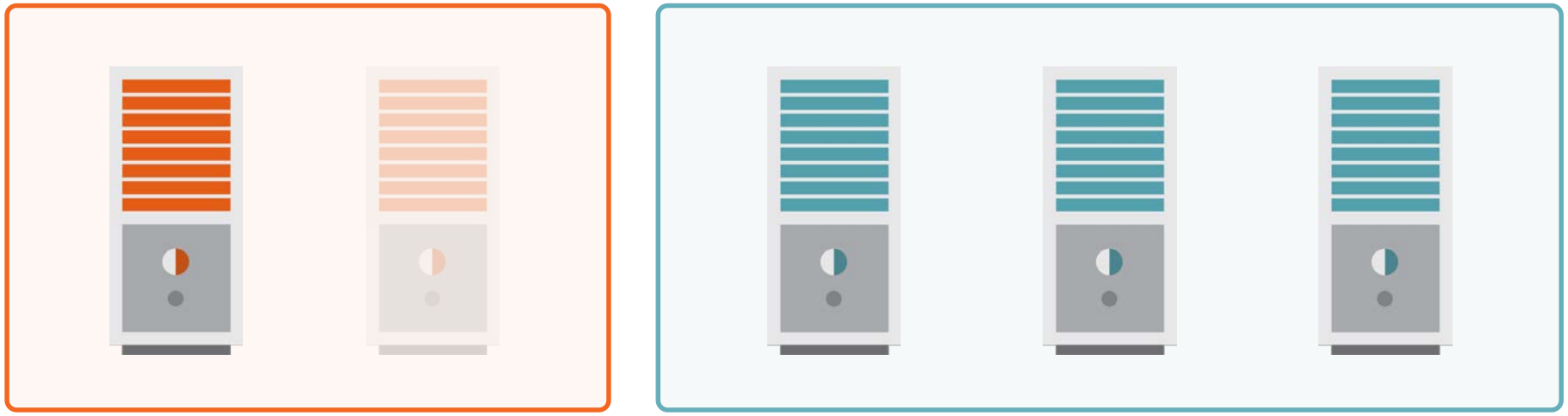




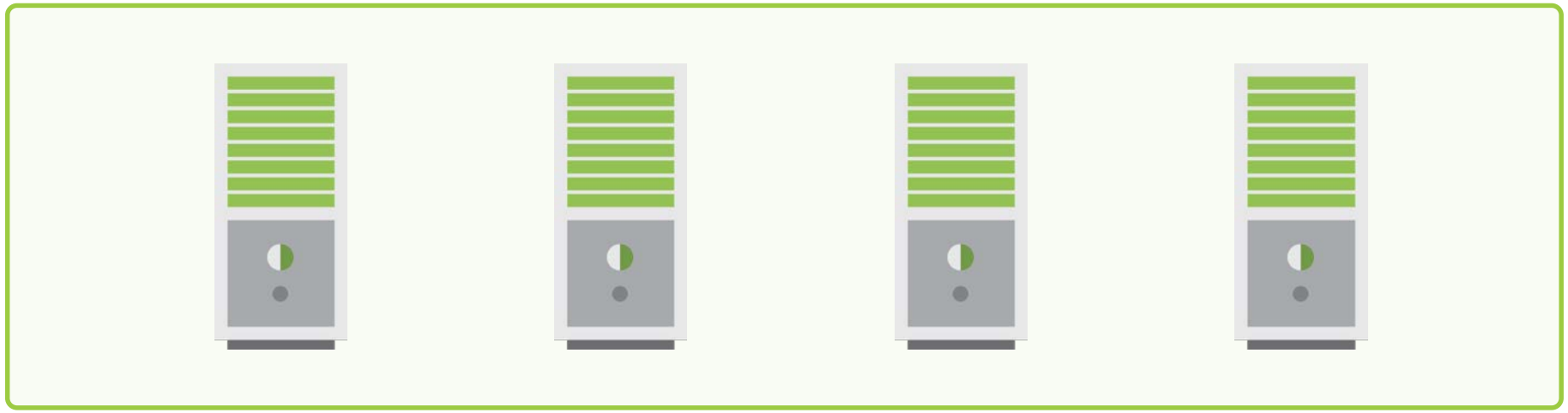


3x Zookeeper

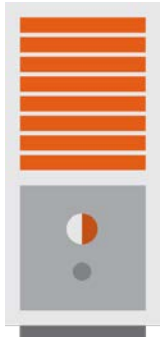
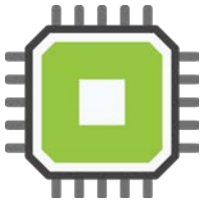
2x Master



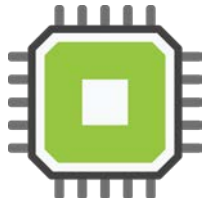
4x Region



36 cores

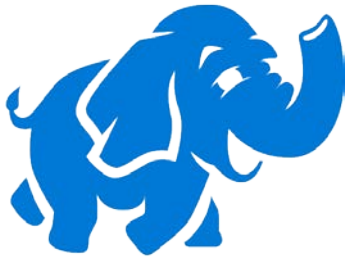
 x4	 x2	 x3
 4 cores	4 cores	4 cores
 14 GB	7 GB	7 GB
 200 GB **SSD**	285 GB	285 GB

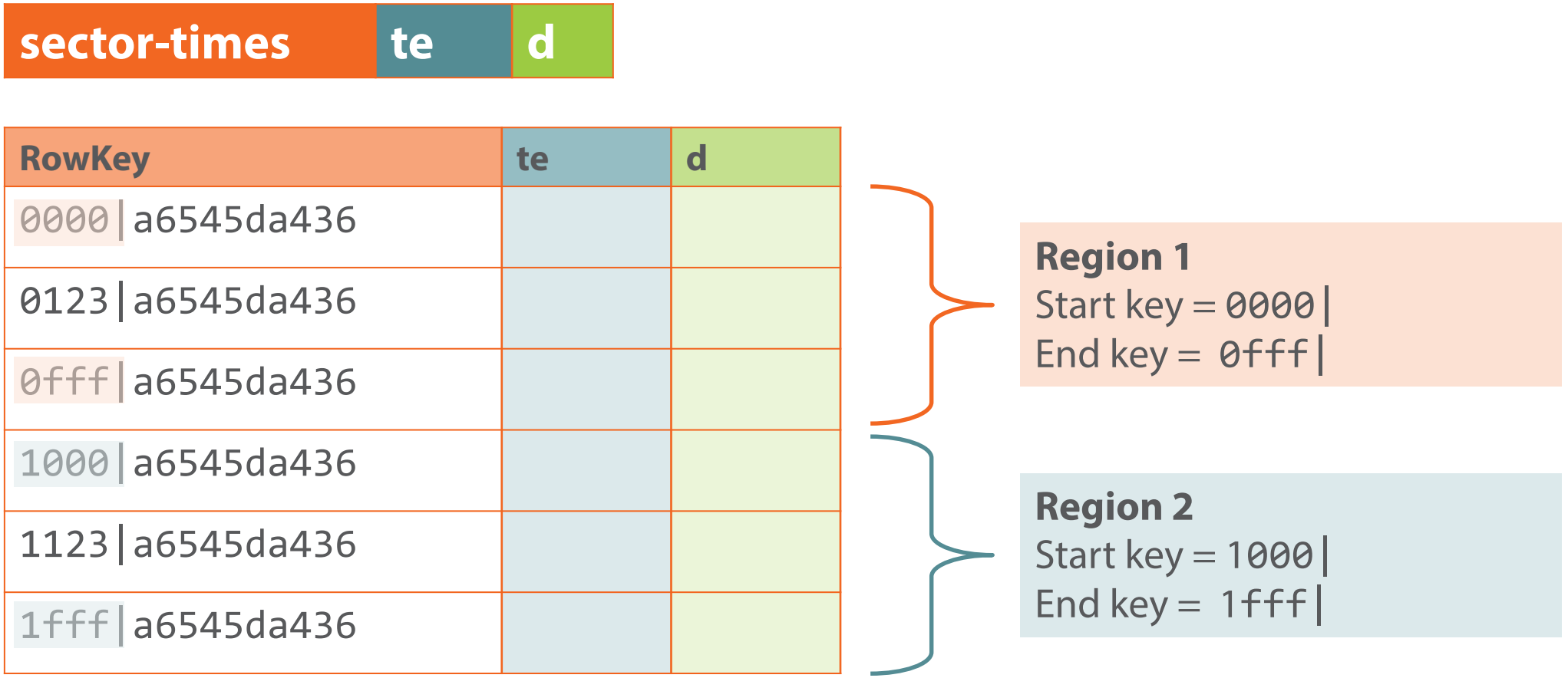
27 cores

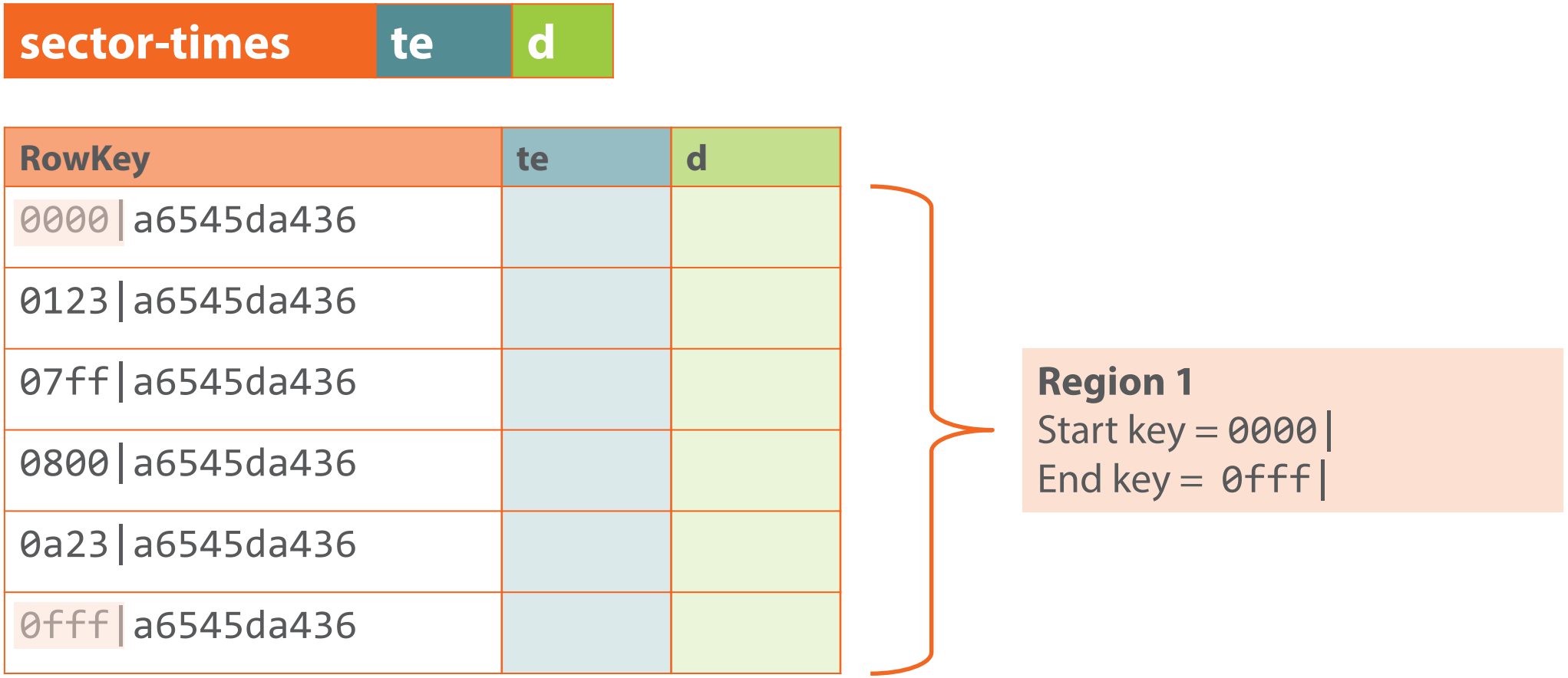
	x4	x2	x3
	4 cores	4 cores	1 core
	7 GB	7 GB	3.5 GB
	285 GB	285 GB	50 GB **SSD**

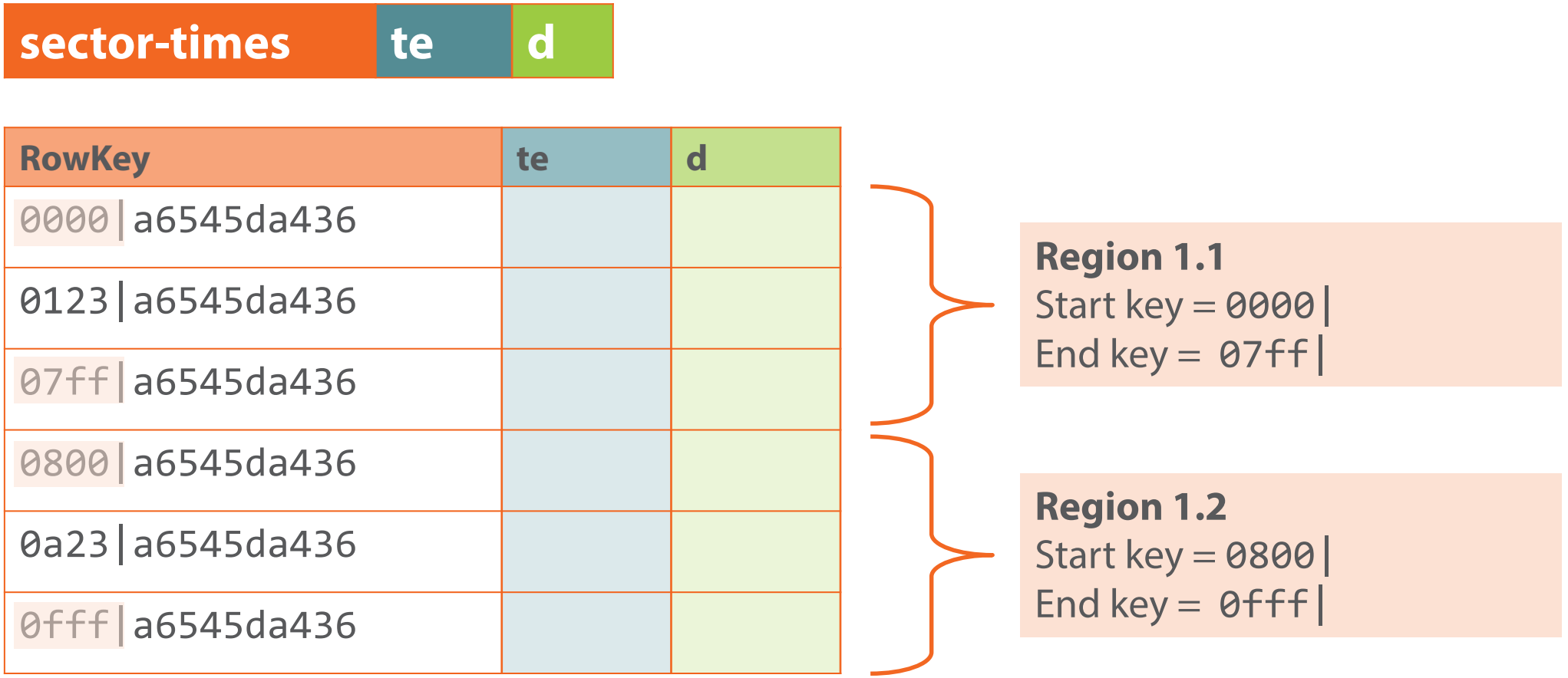
cluster-01

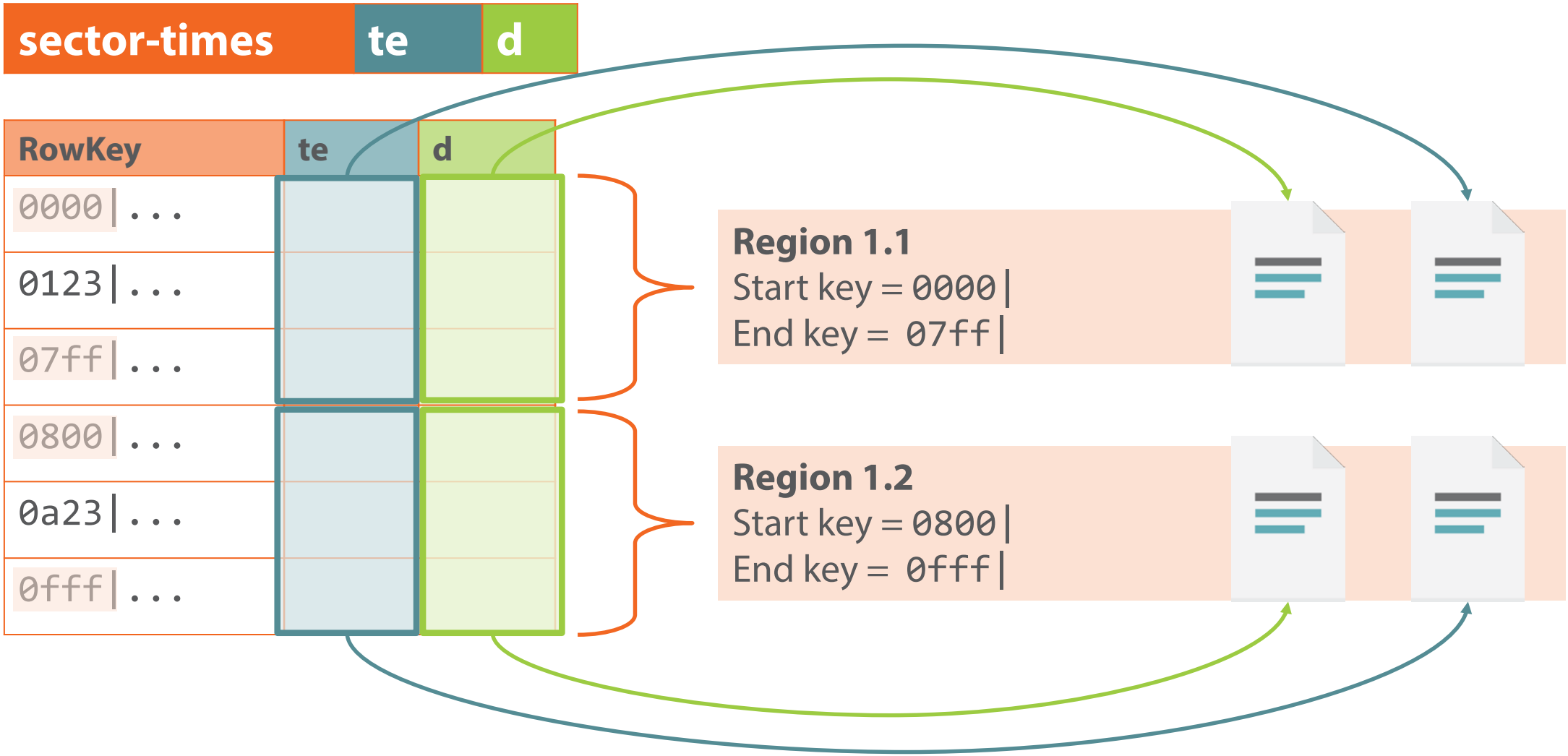
cluster-02



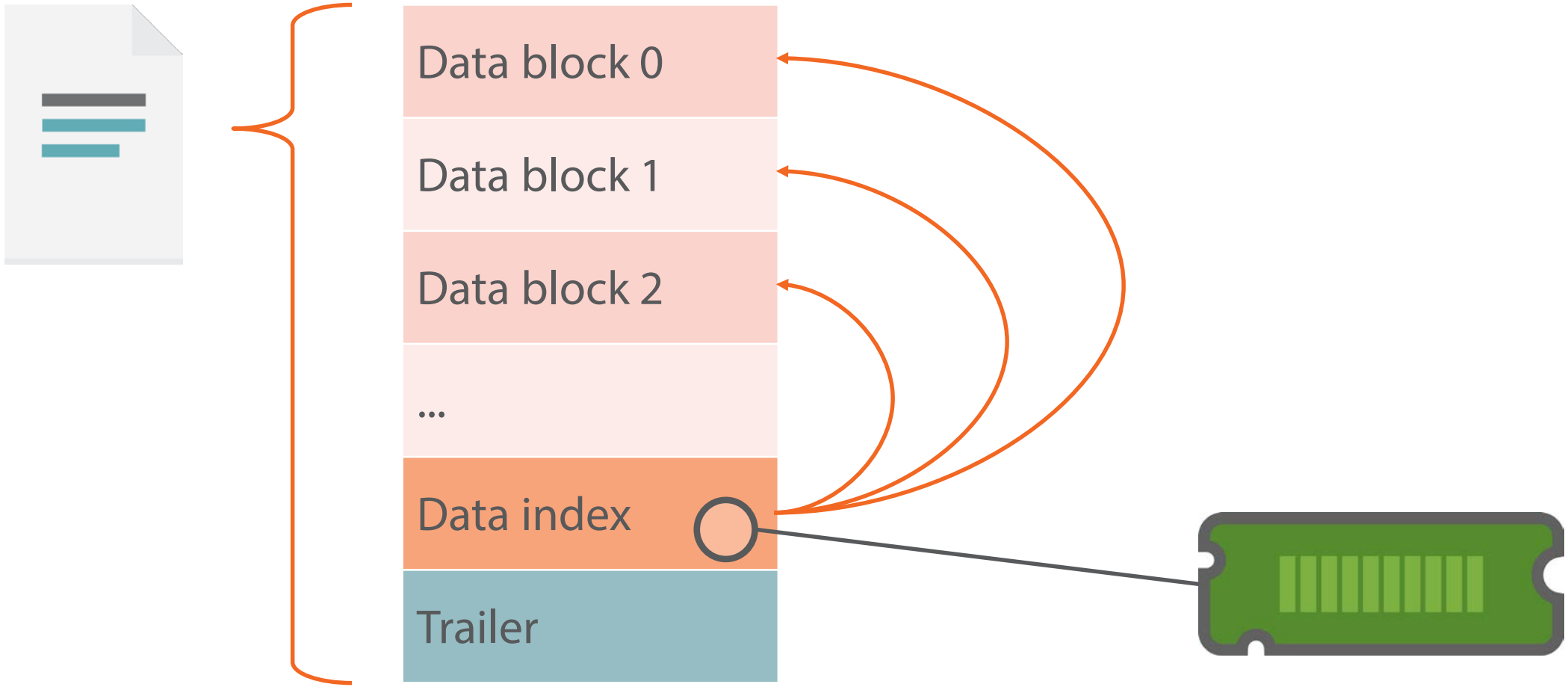




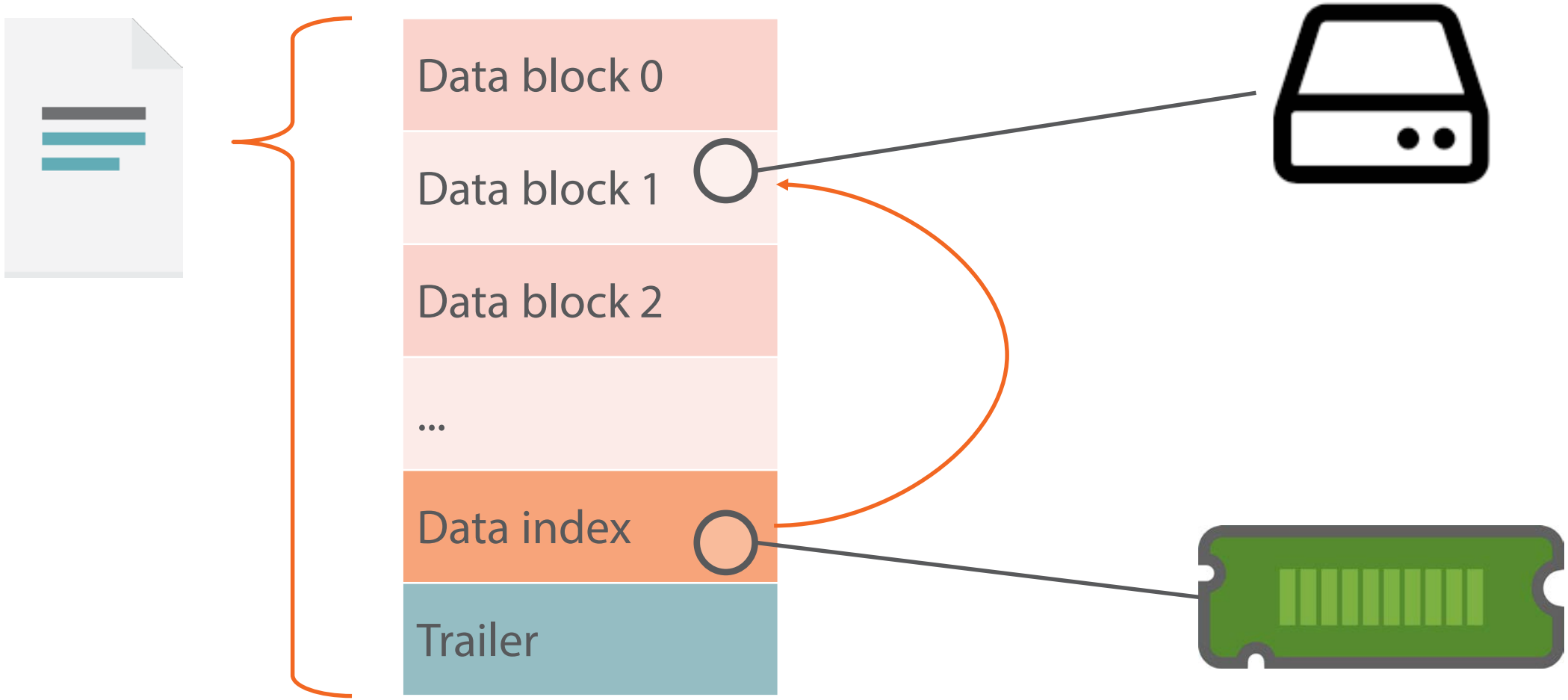


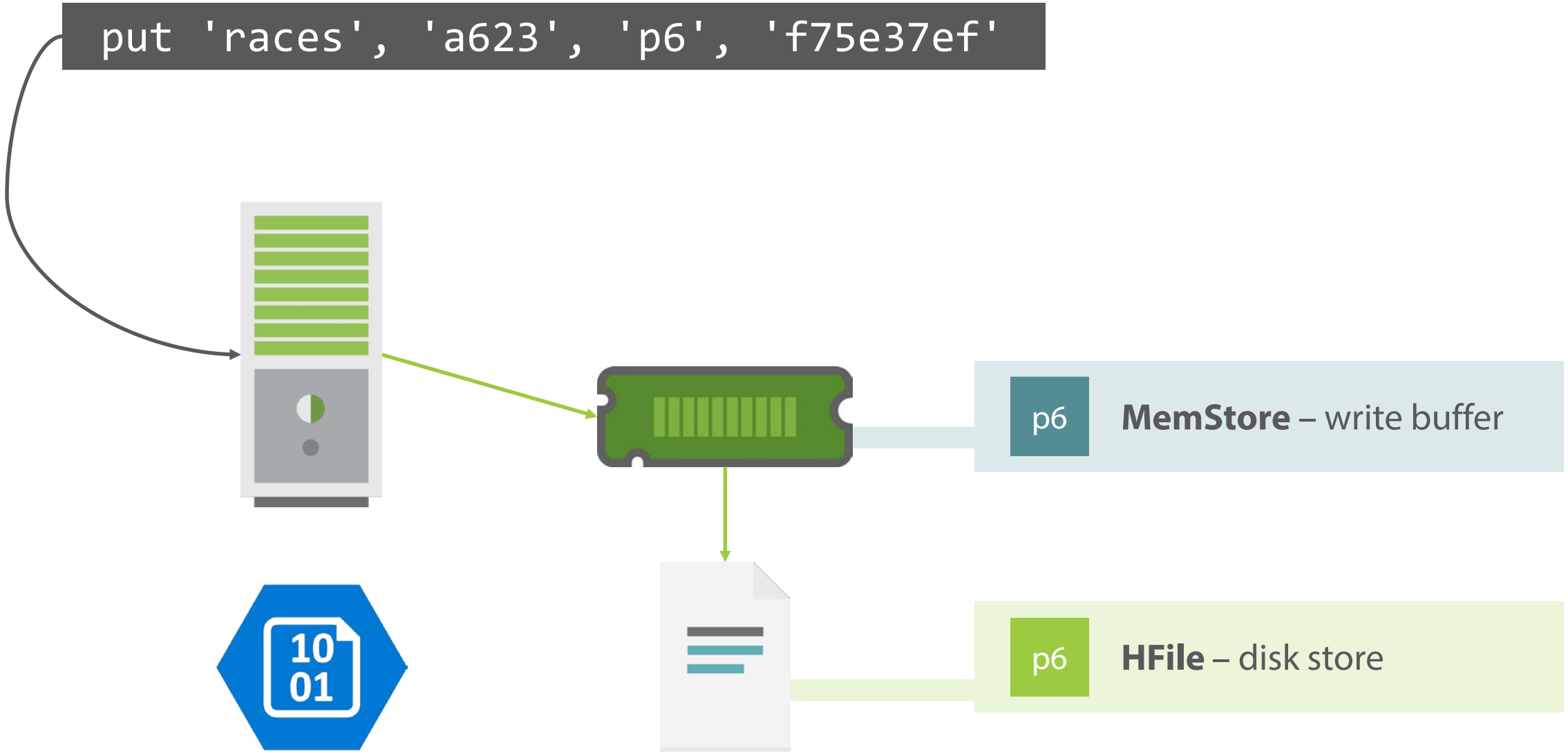


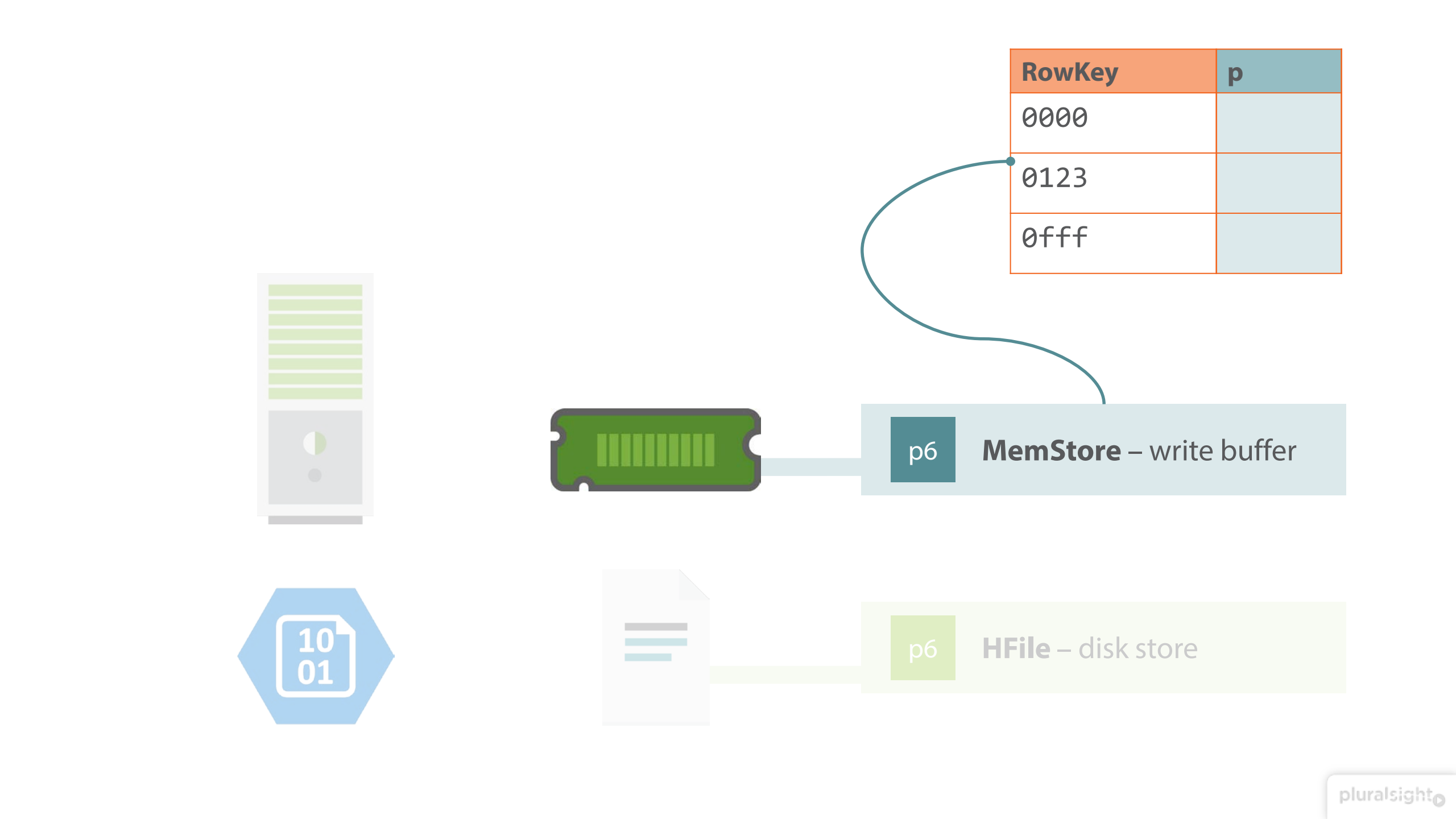
/sector-times/region-1.1/te/hfile-1

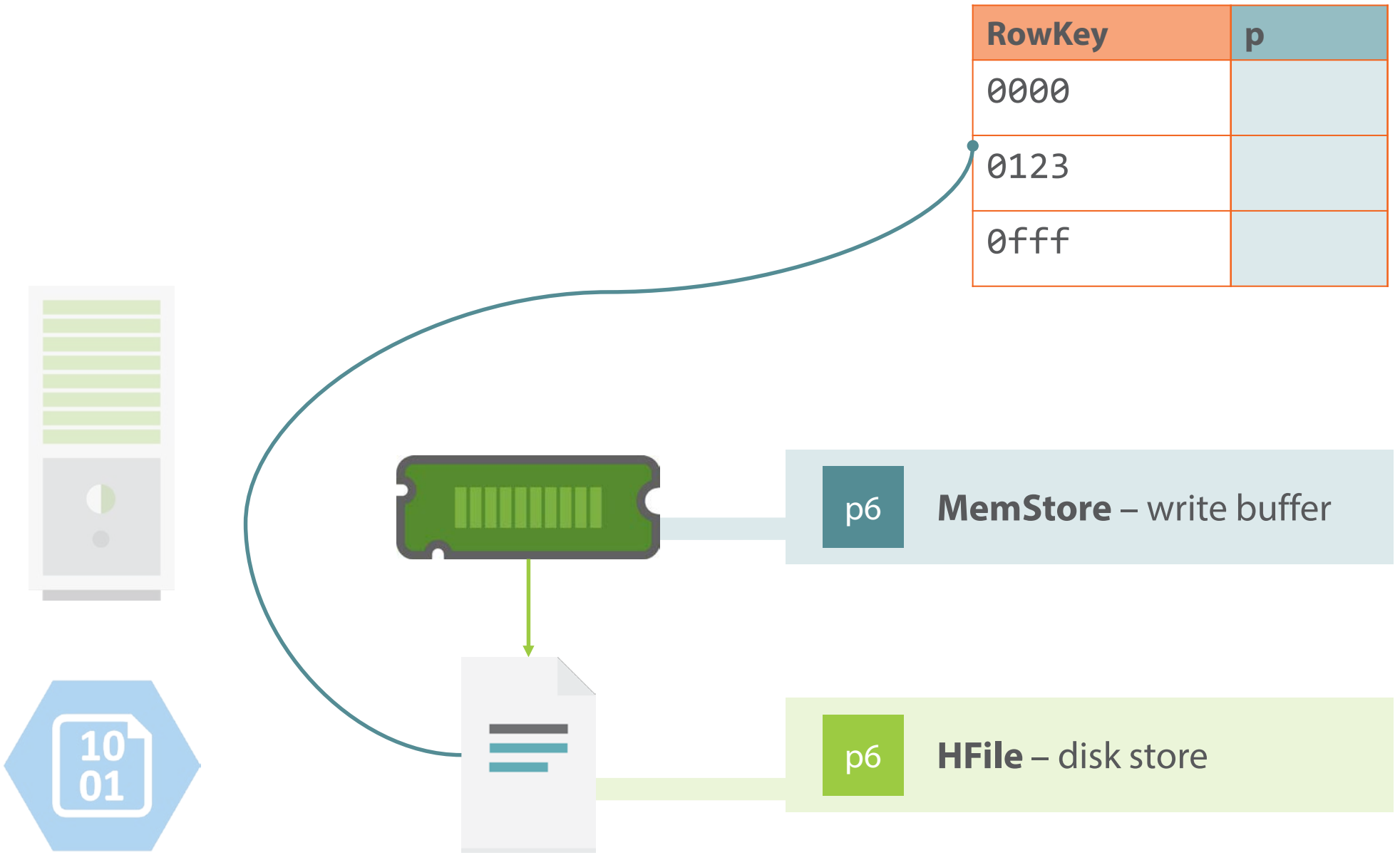


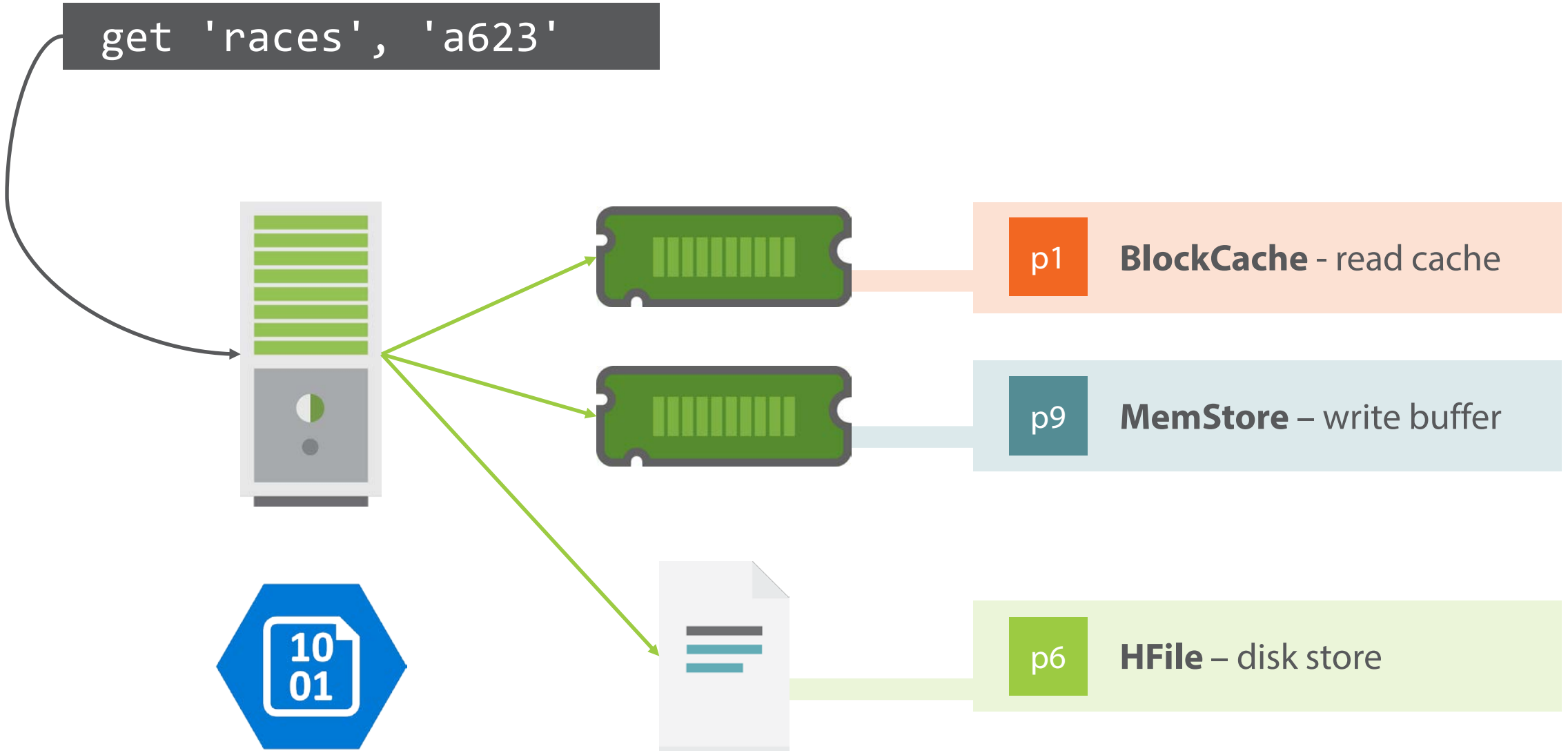
/sector-times/region-1.1/te/hfile-1

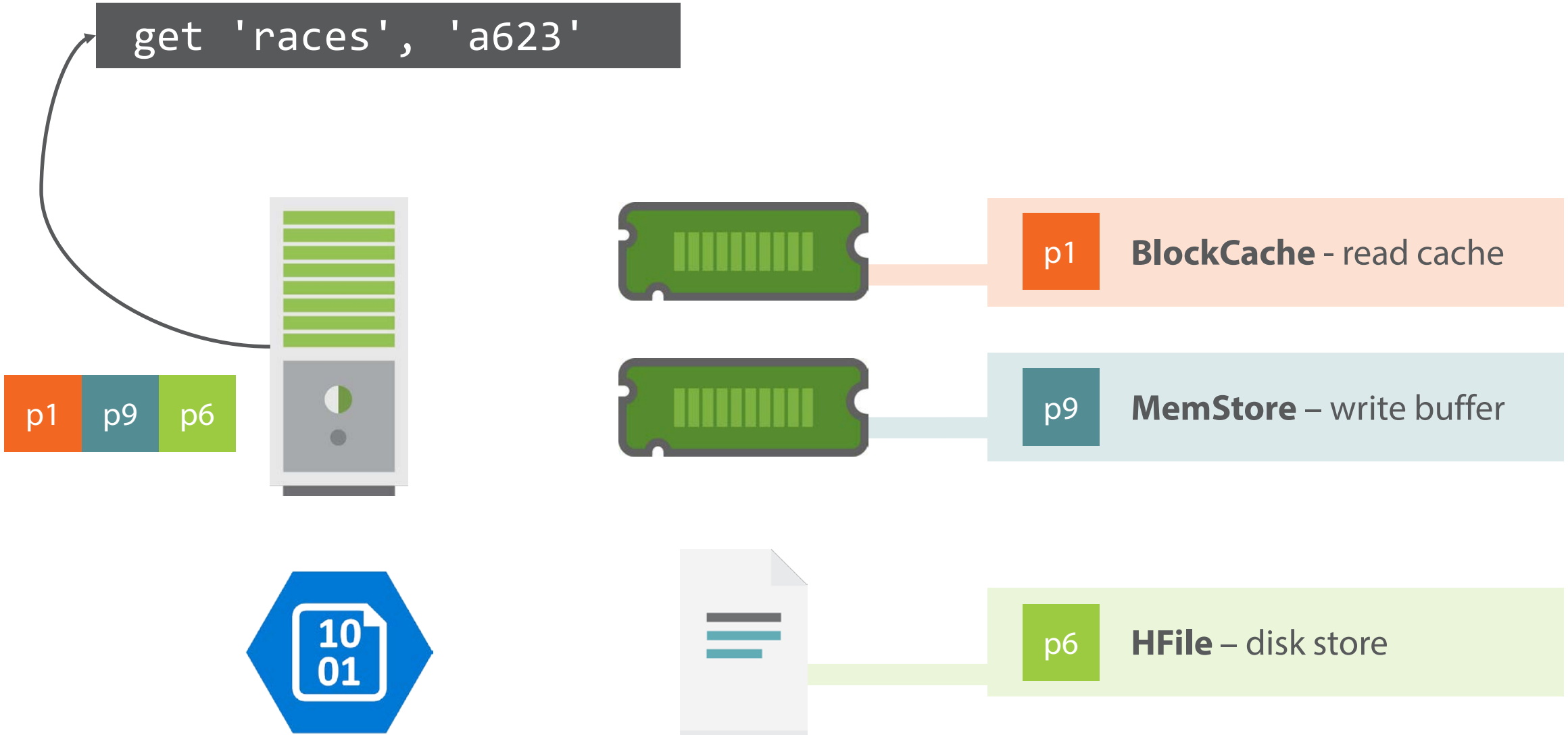


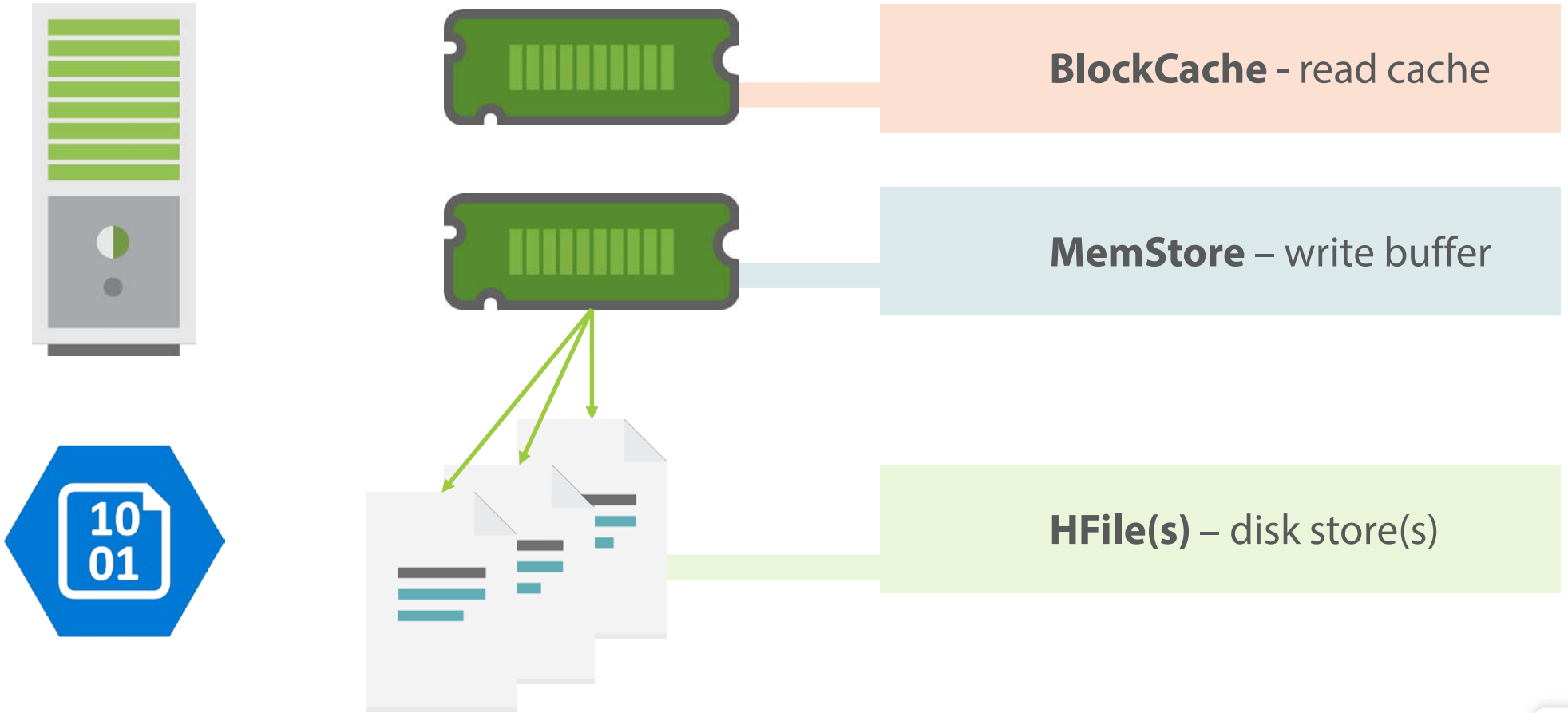


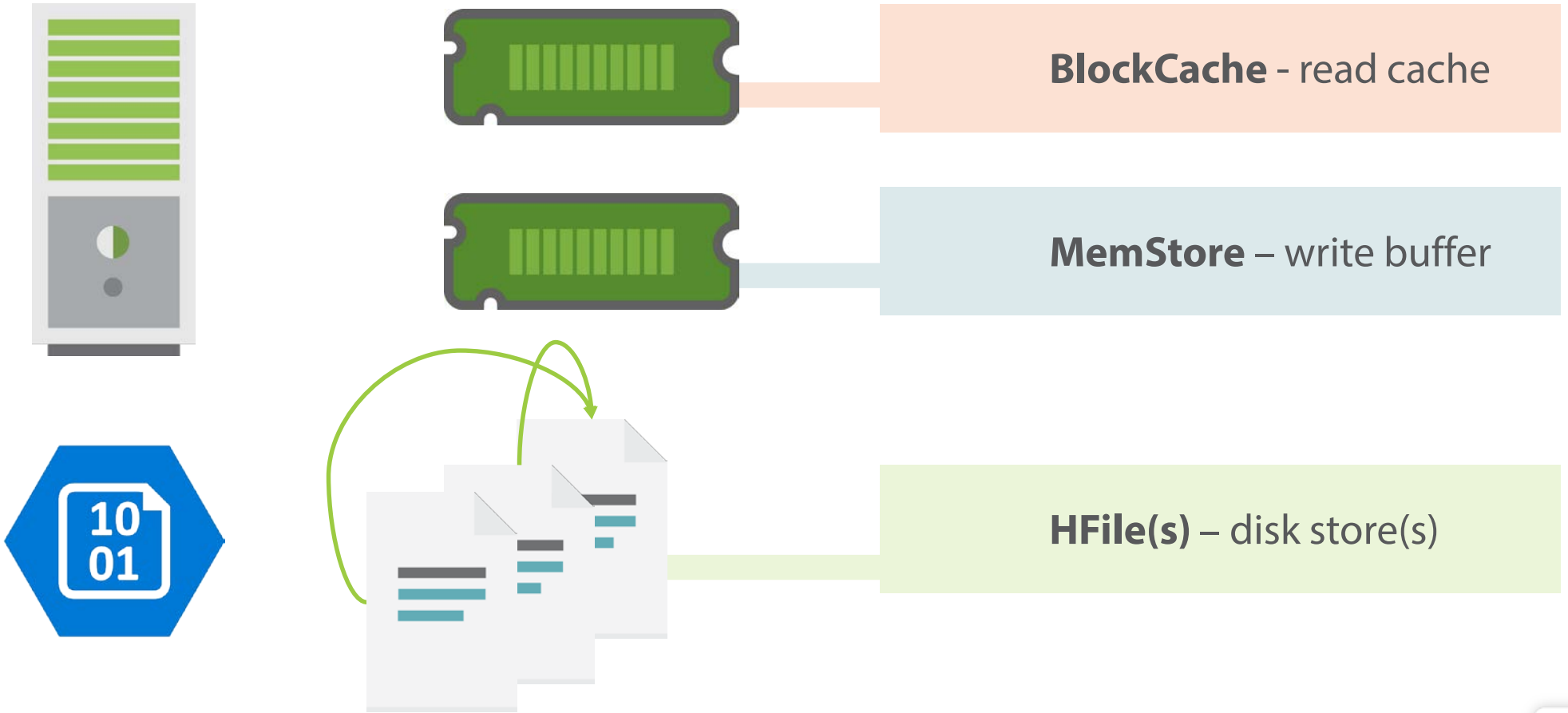

RowKey
p
0000
0123
0fff
p6
MemStore – write buffer
10
01
p6
HFile – disk store

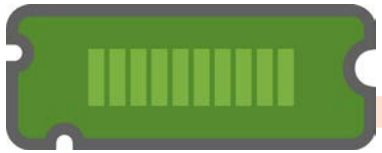




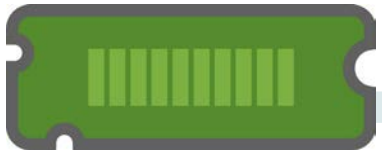






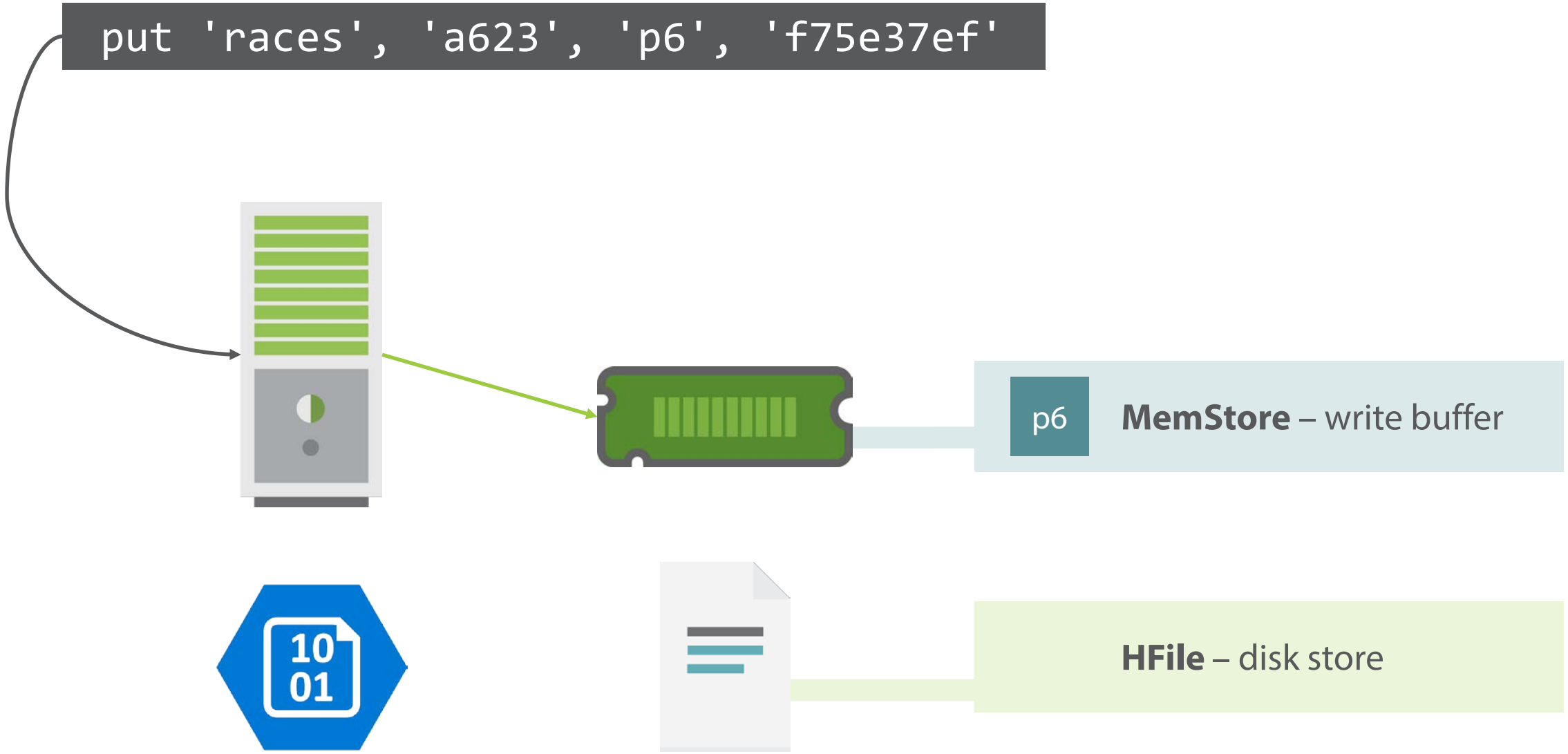


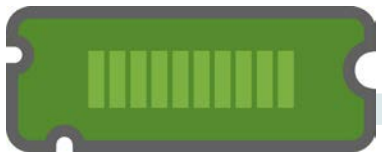
BlockCache - read cache



MemStore – write buffer

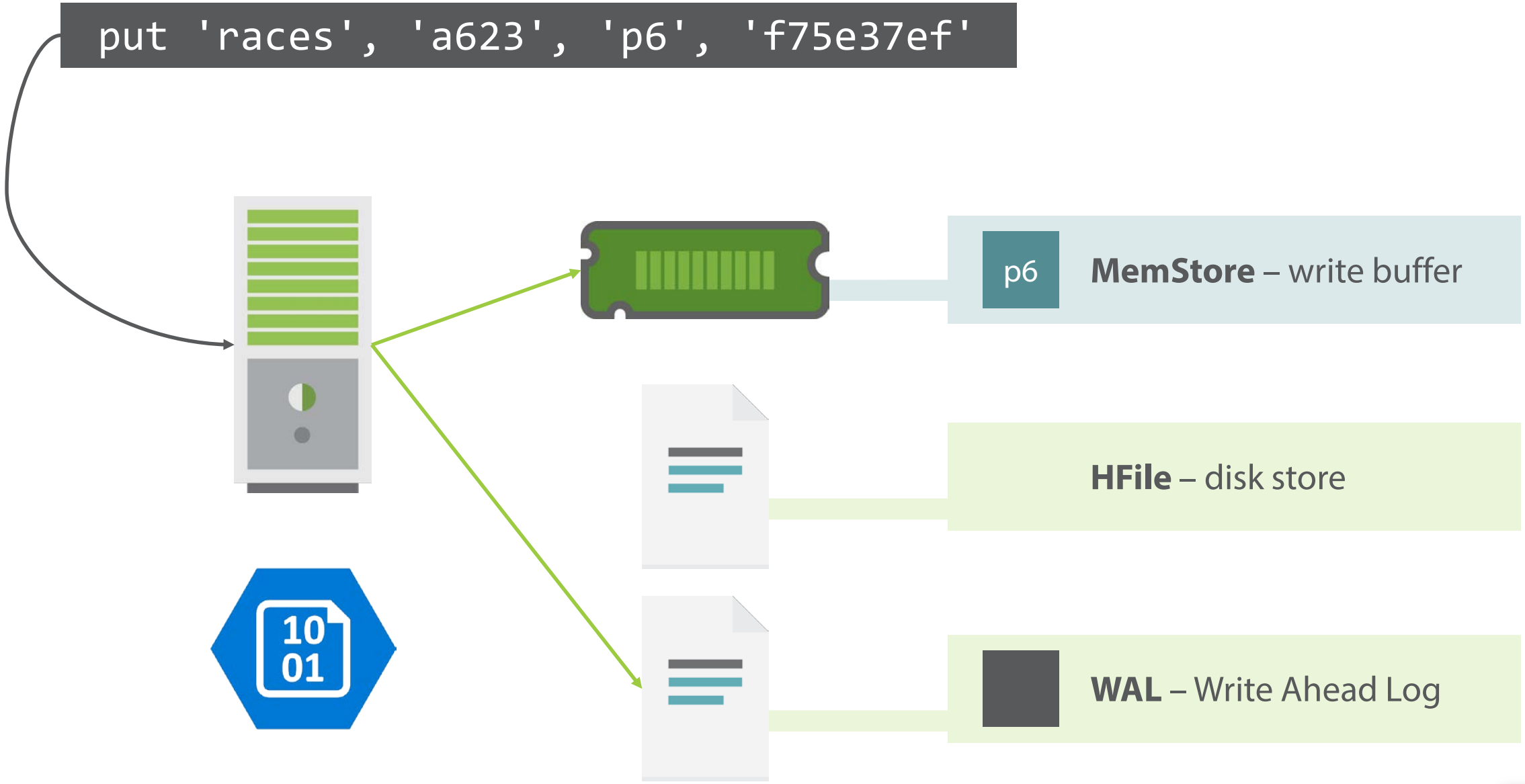
HFile(s) – disk store(s)

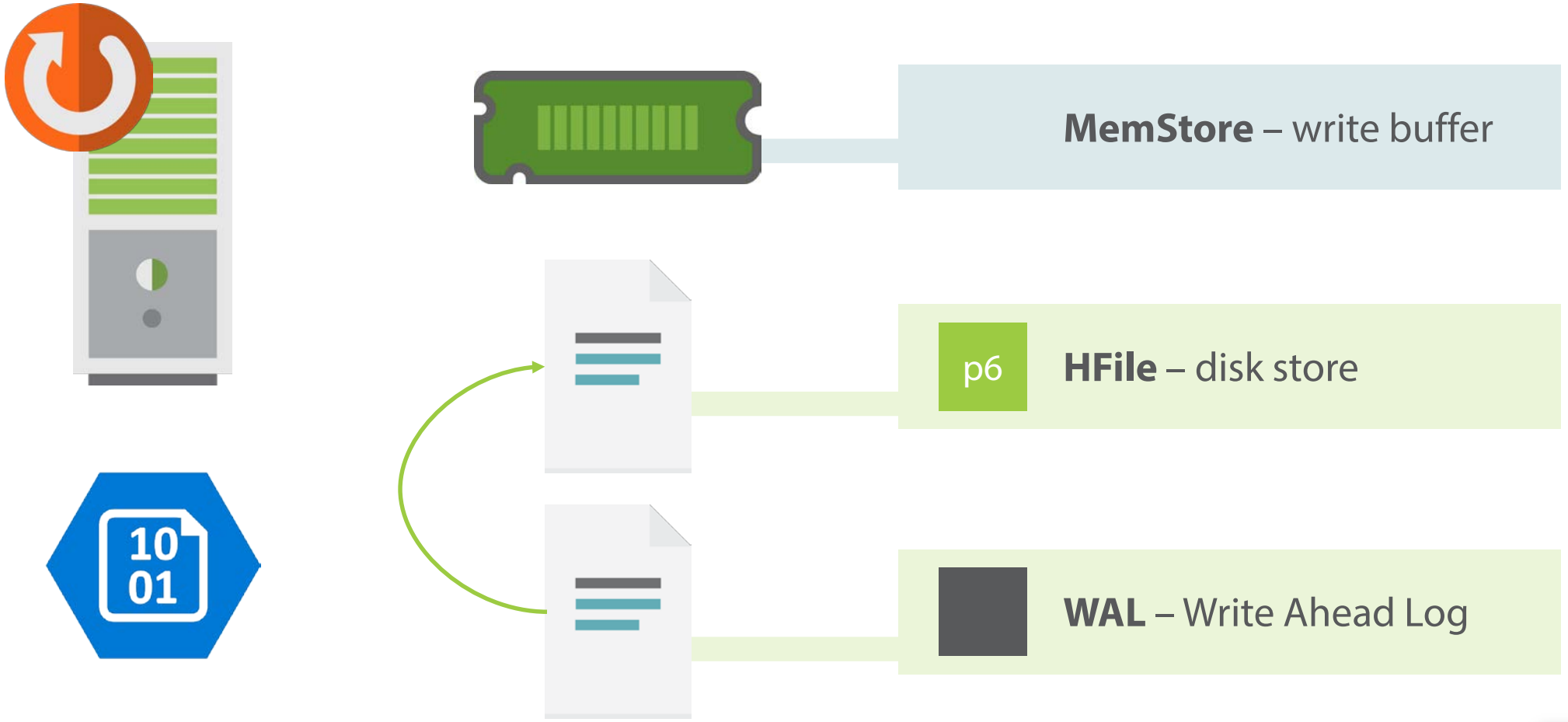

put 'races', 'a623', 'p6', 'f75e37ef'
10
01
p6
MemStore – write buffer
HFile – disk store

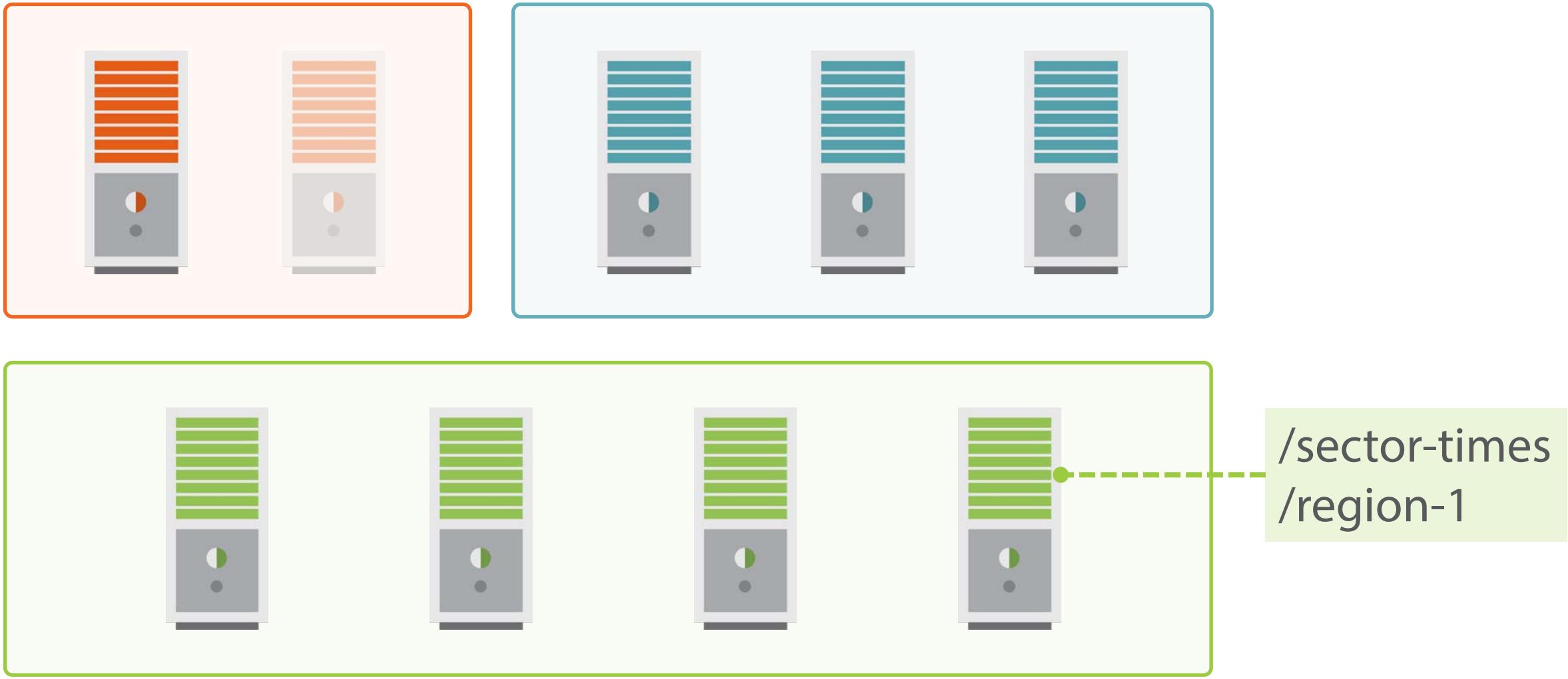


p6 **MemStore** – write buffer

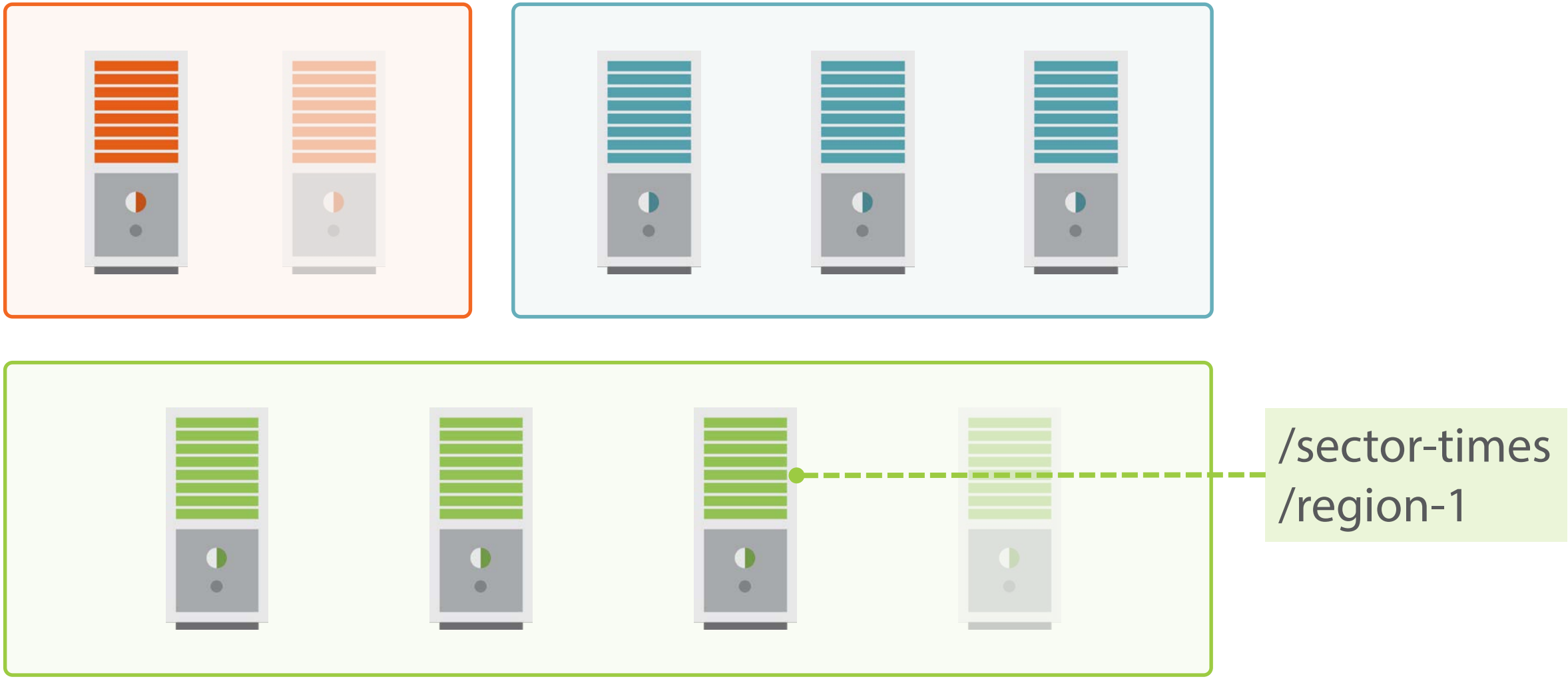
HFile – disk store

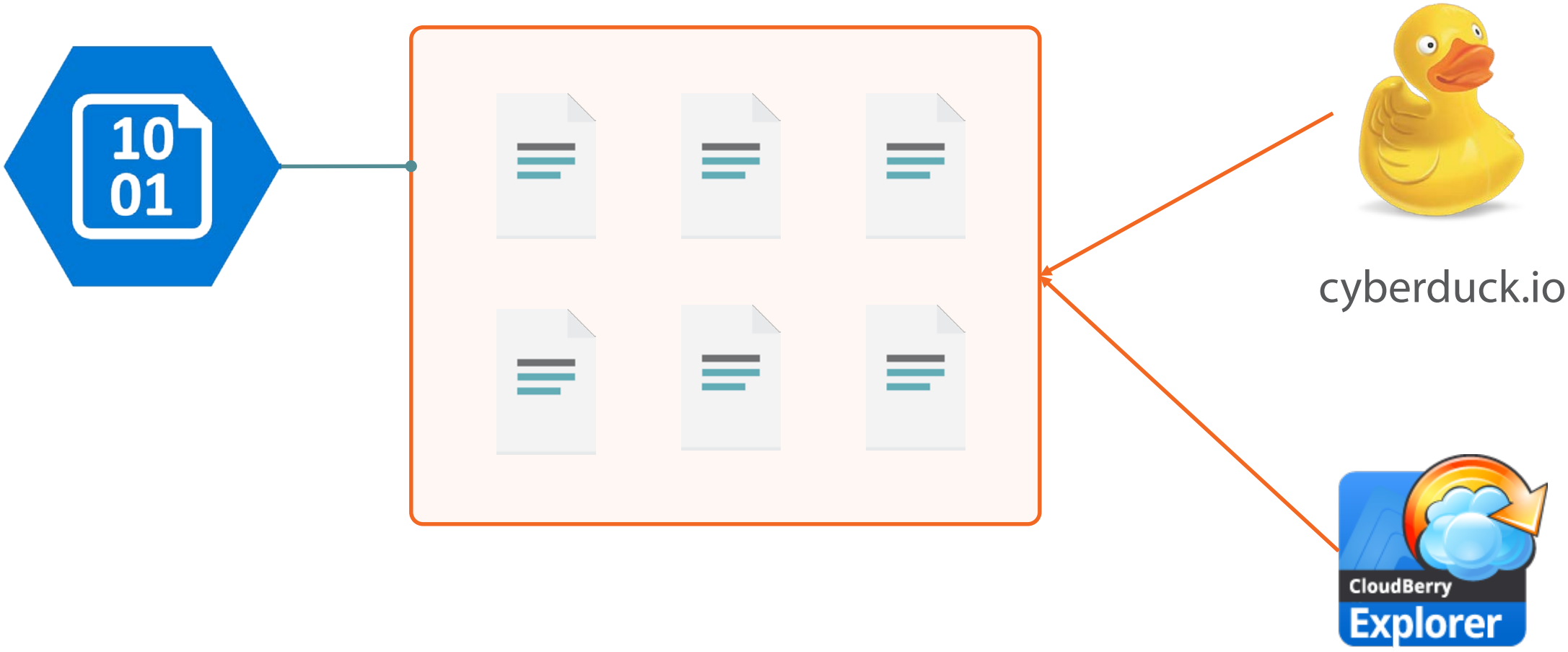






/sector-times
/region-1



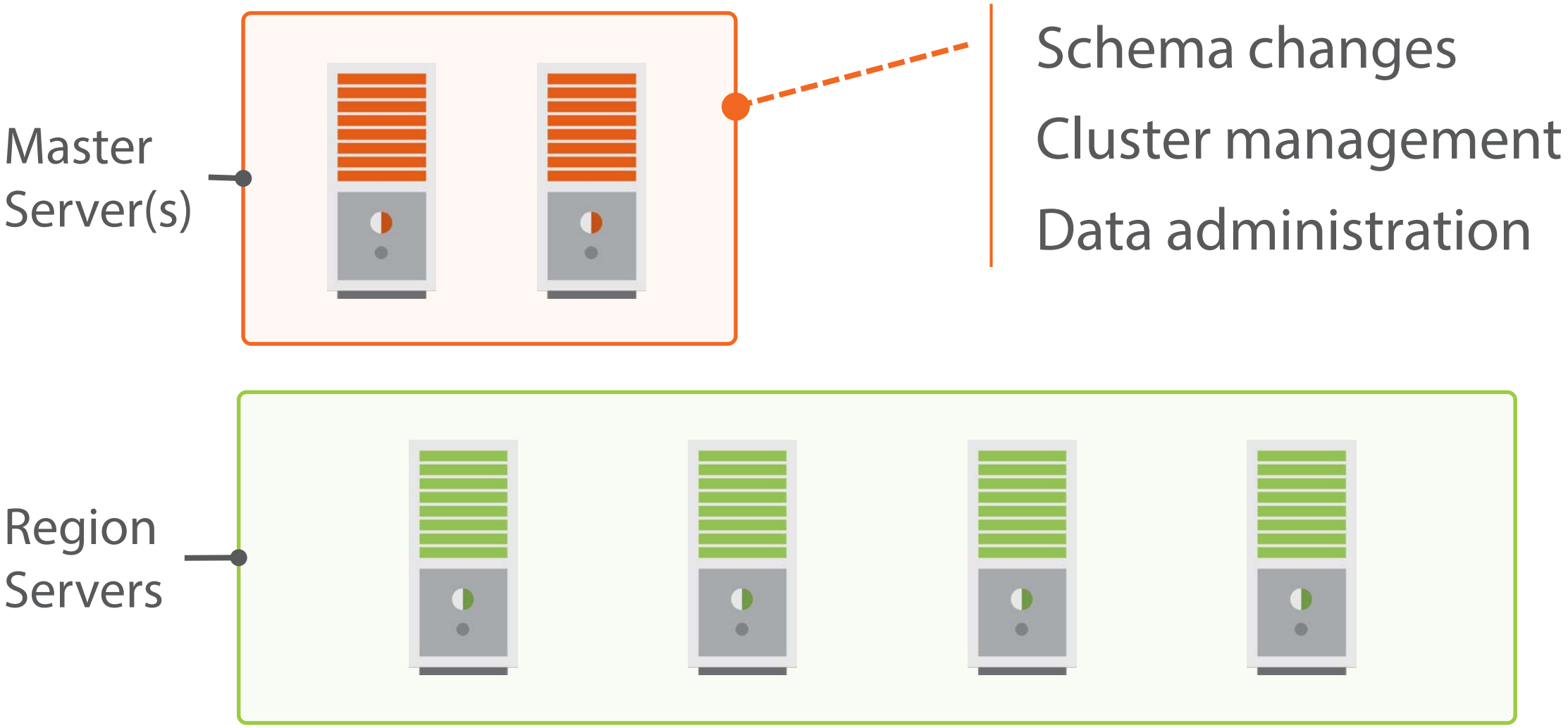


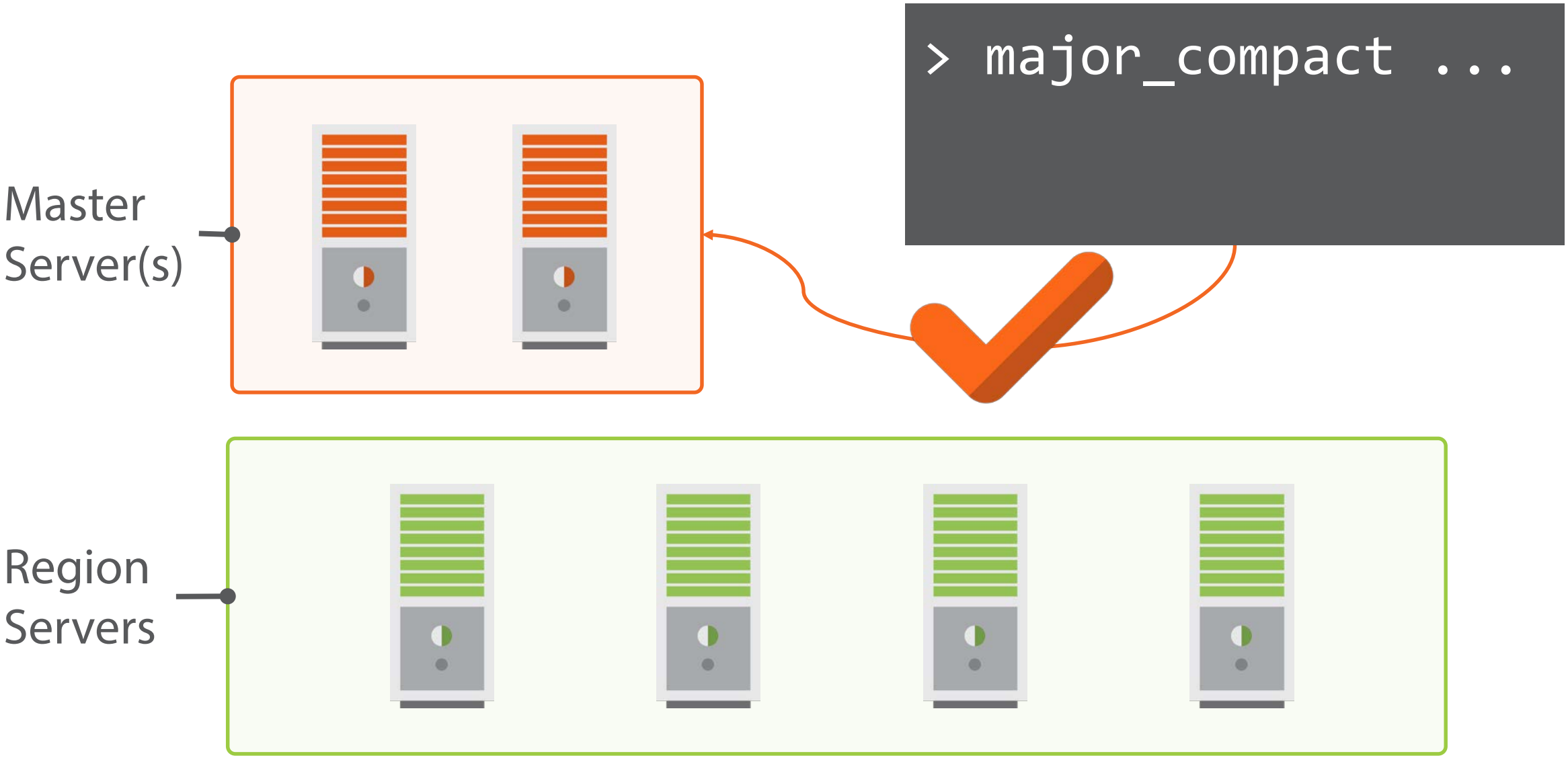
Demo: HBase File Storage

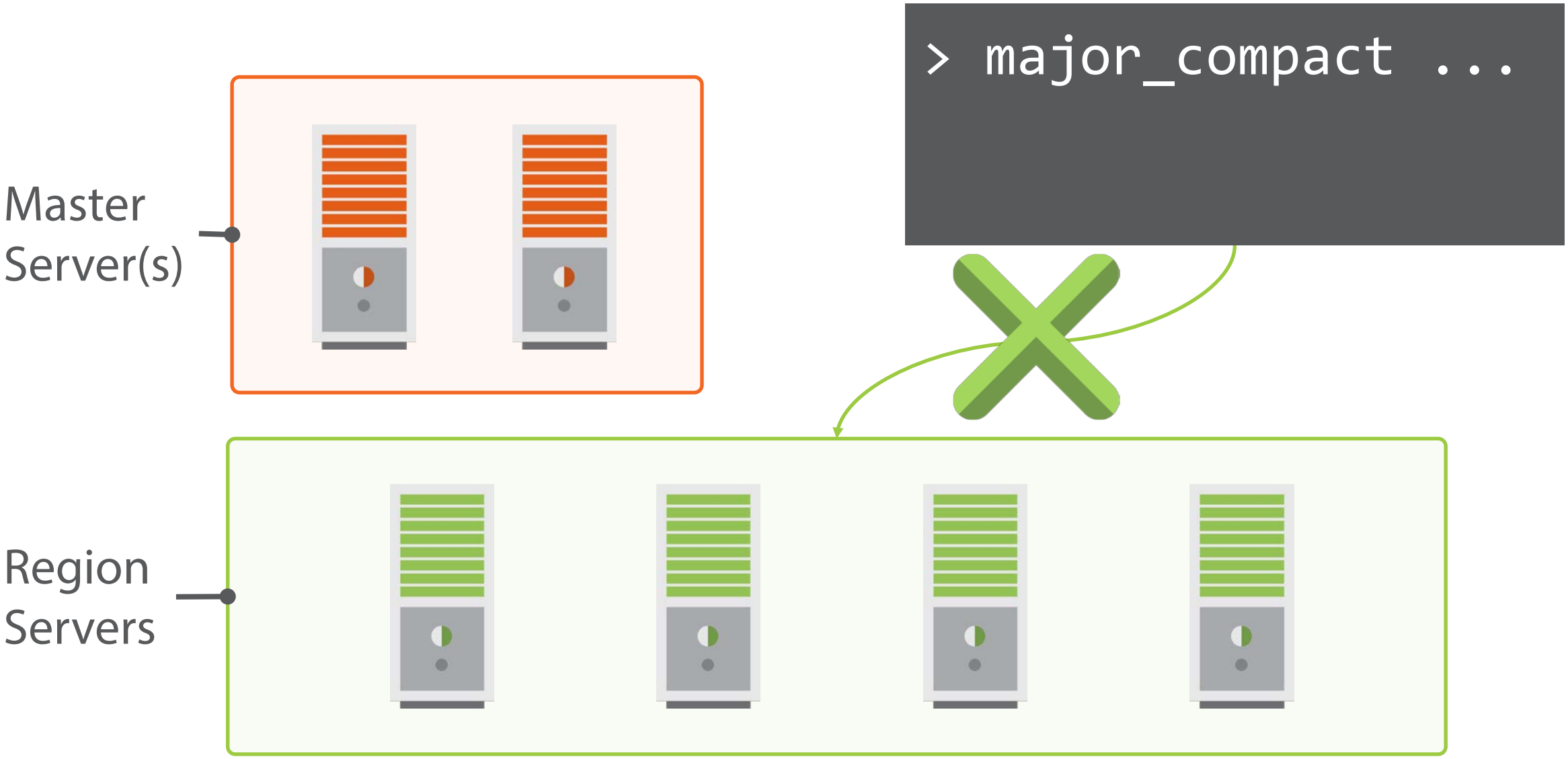
Azure Portal

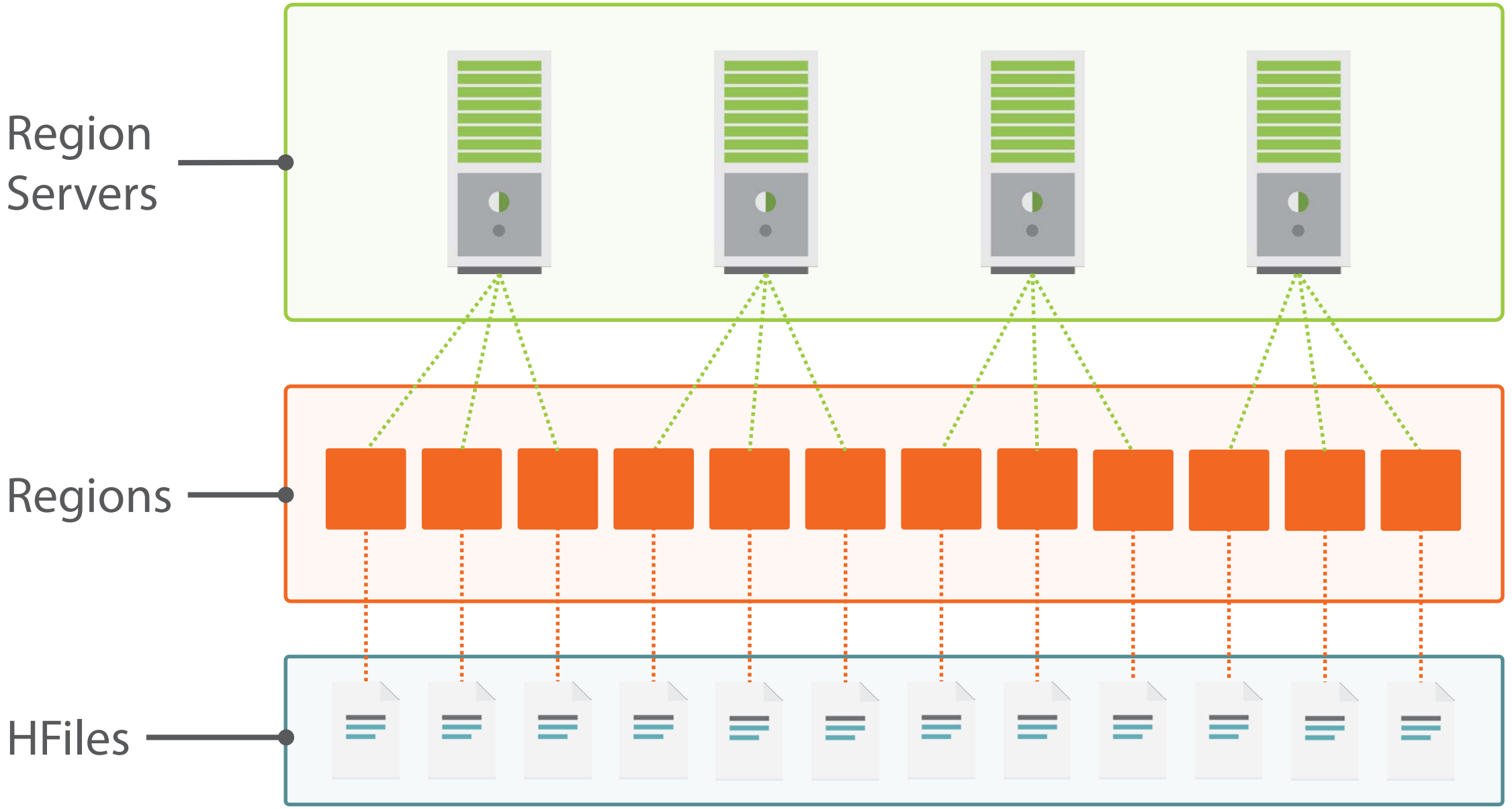
Folder Structure

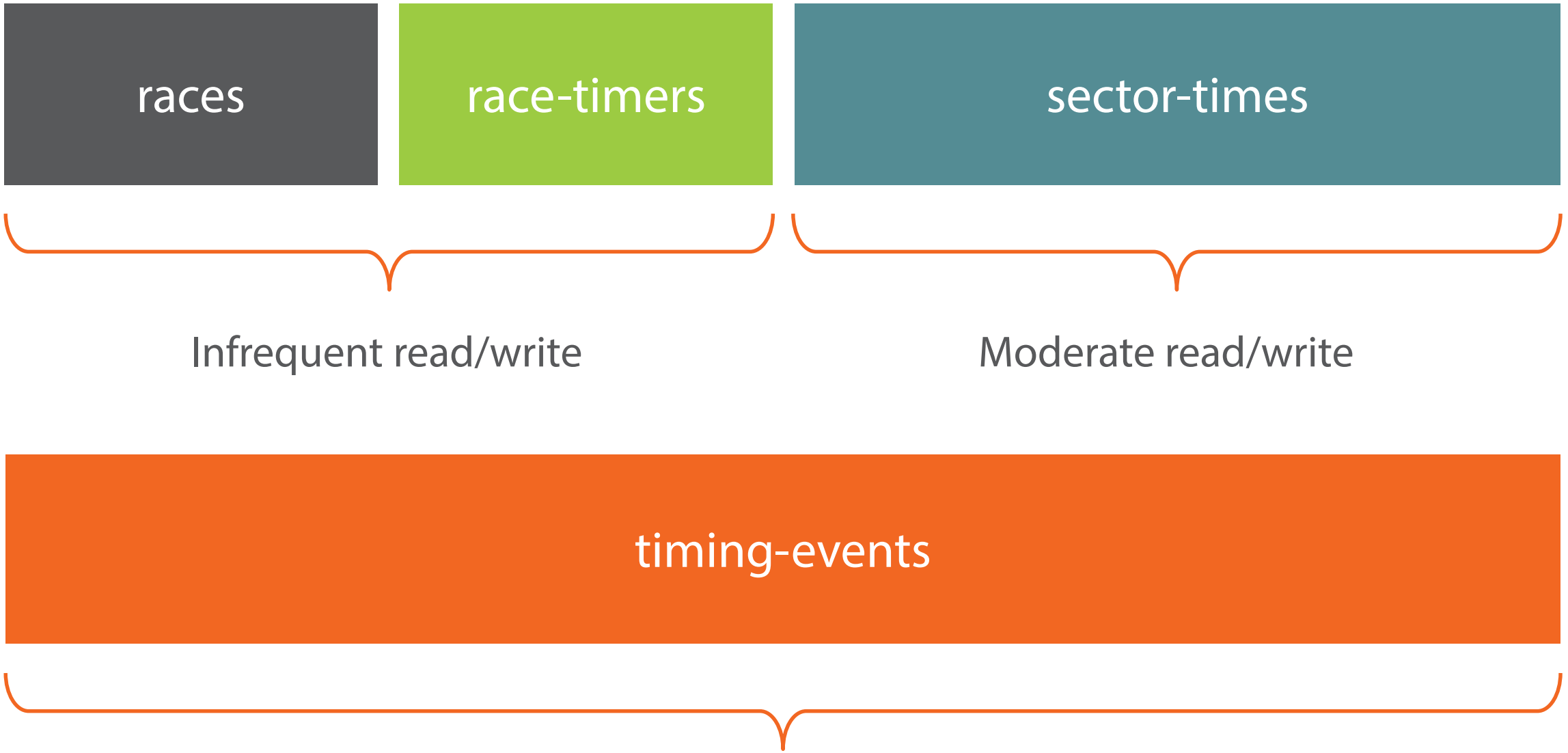
HFile Contents

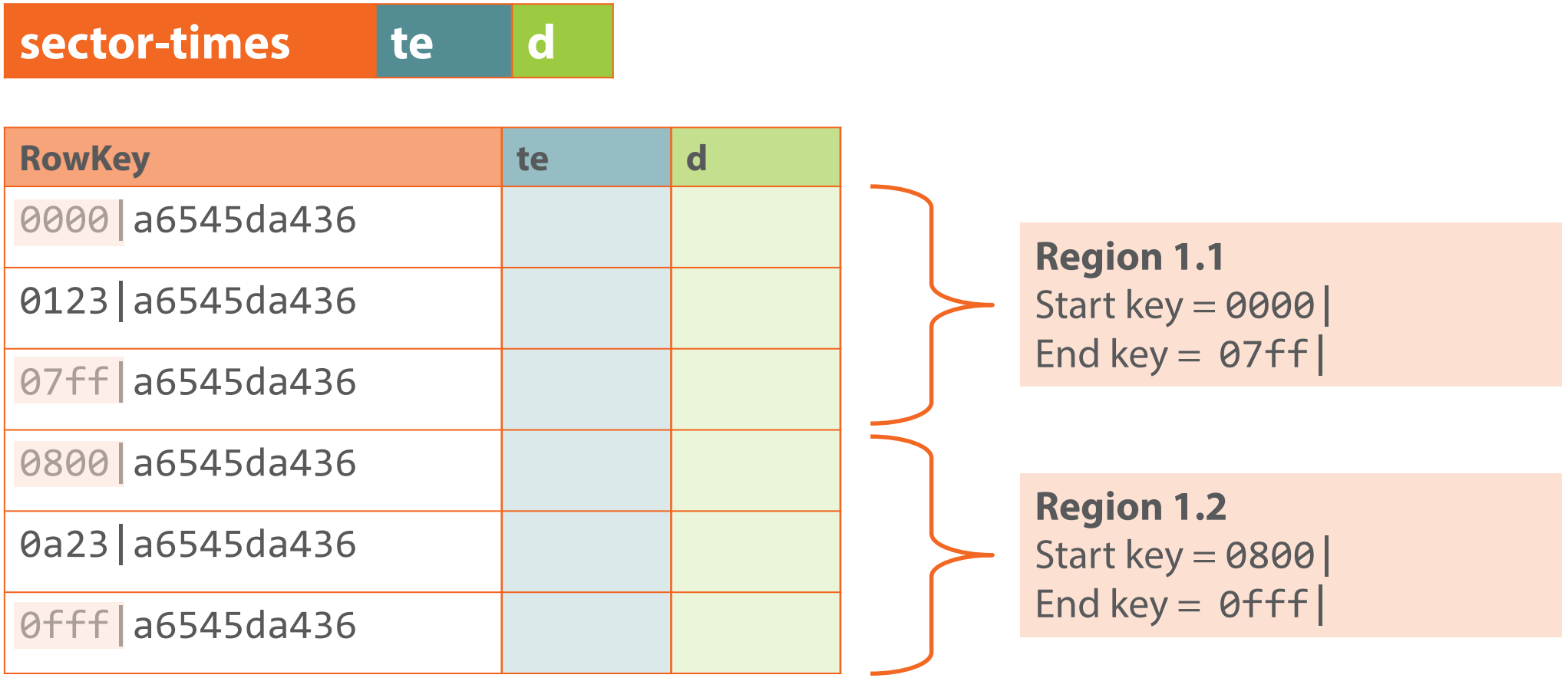




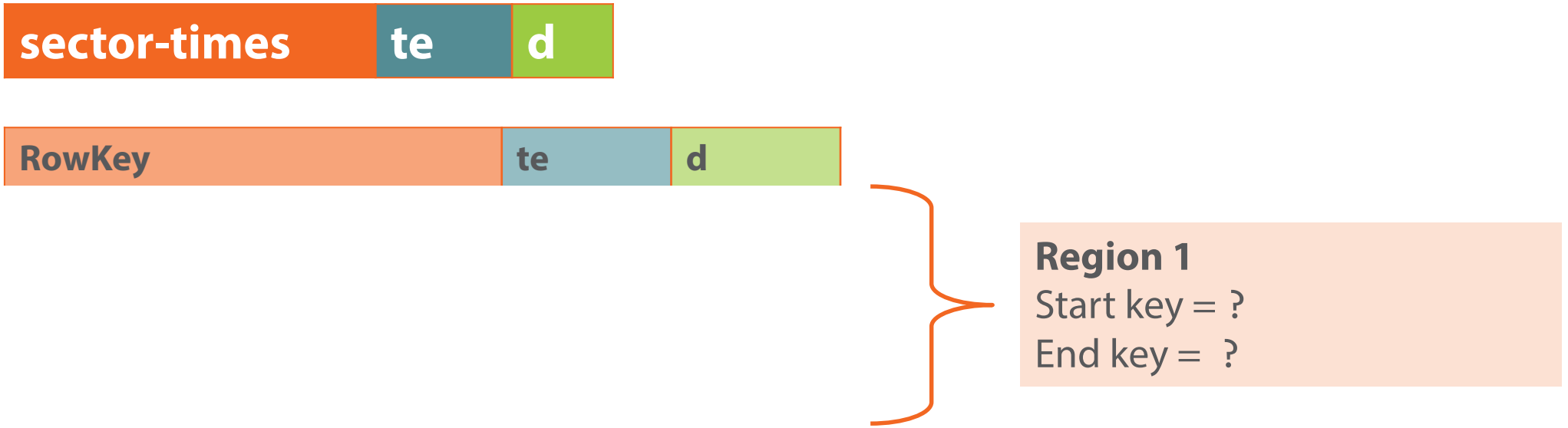


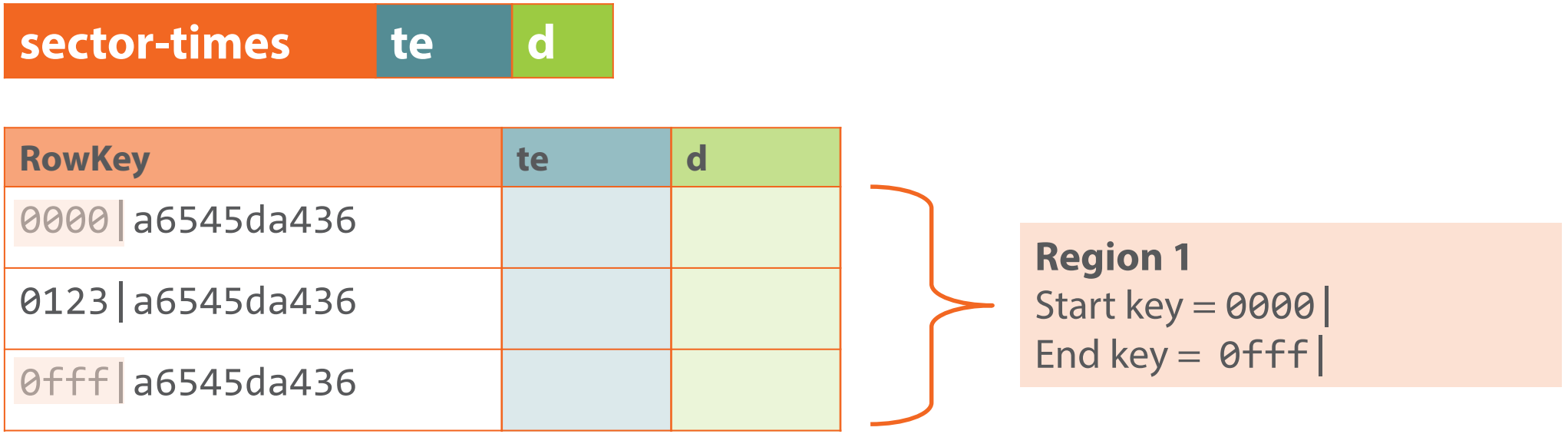


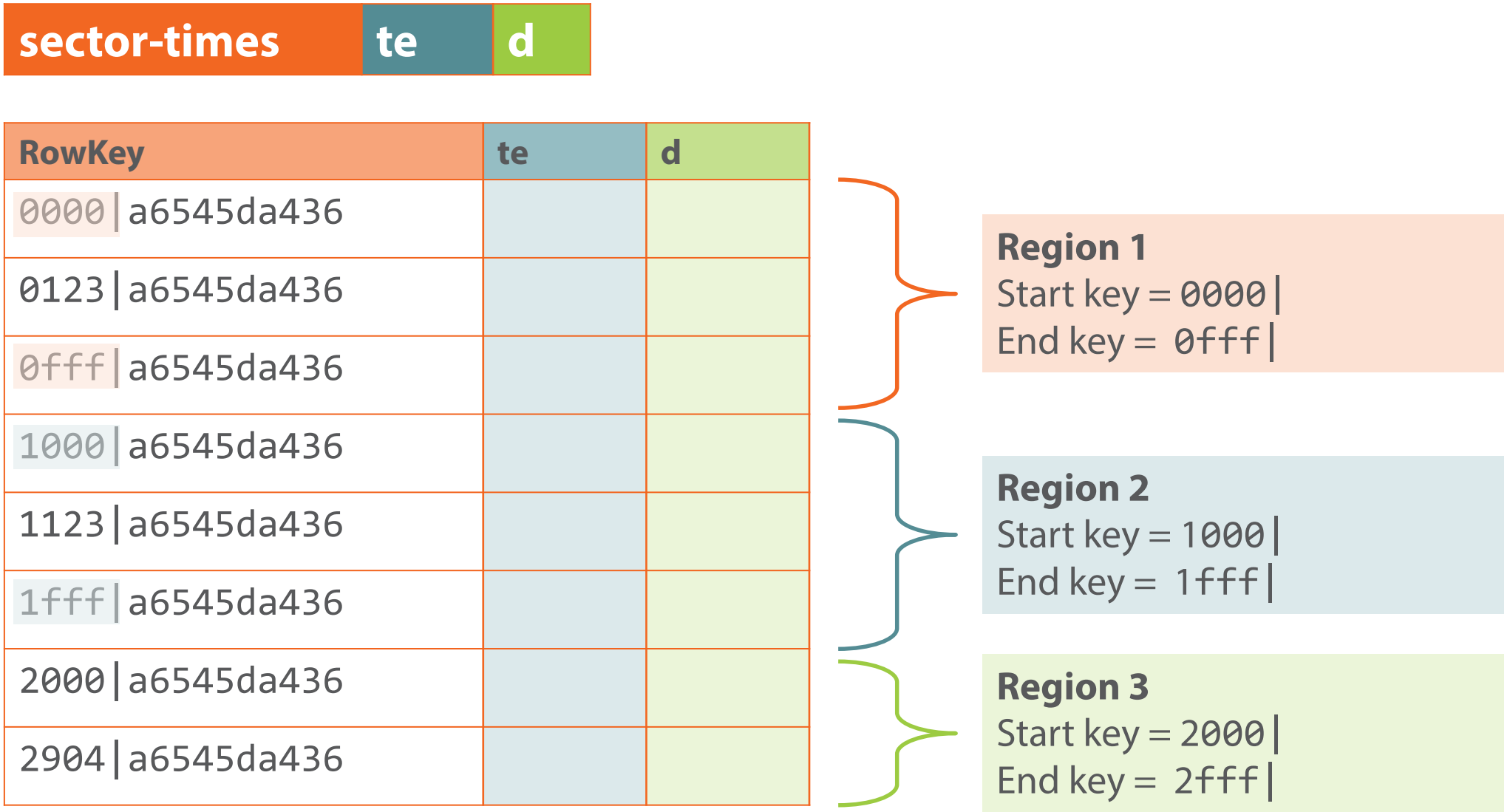




Region split at **256MB**





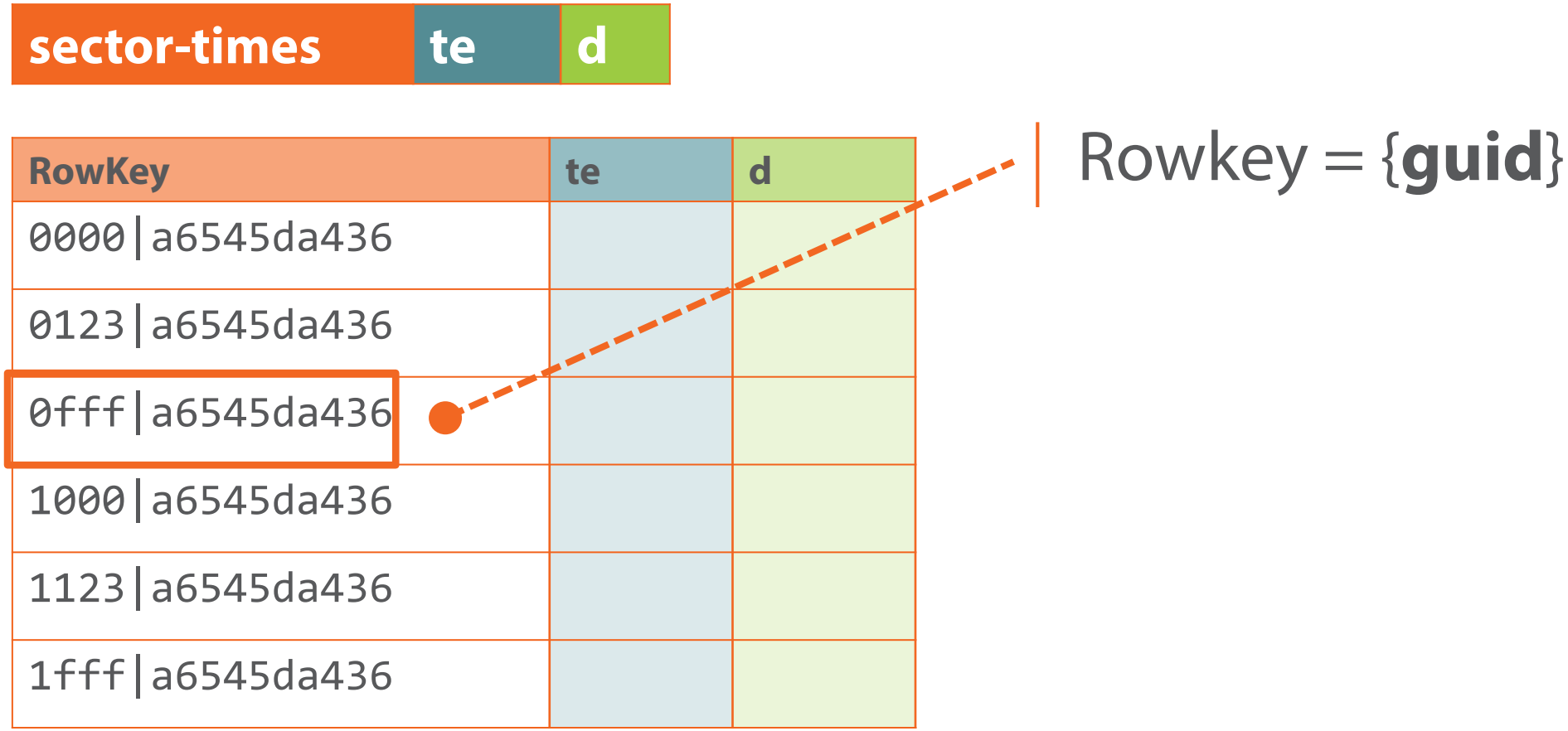


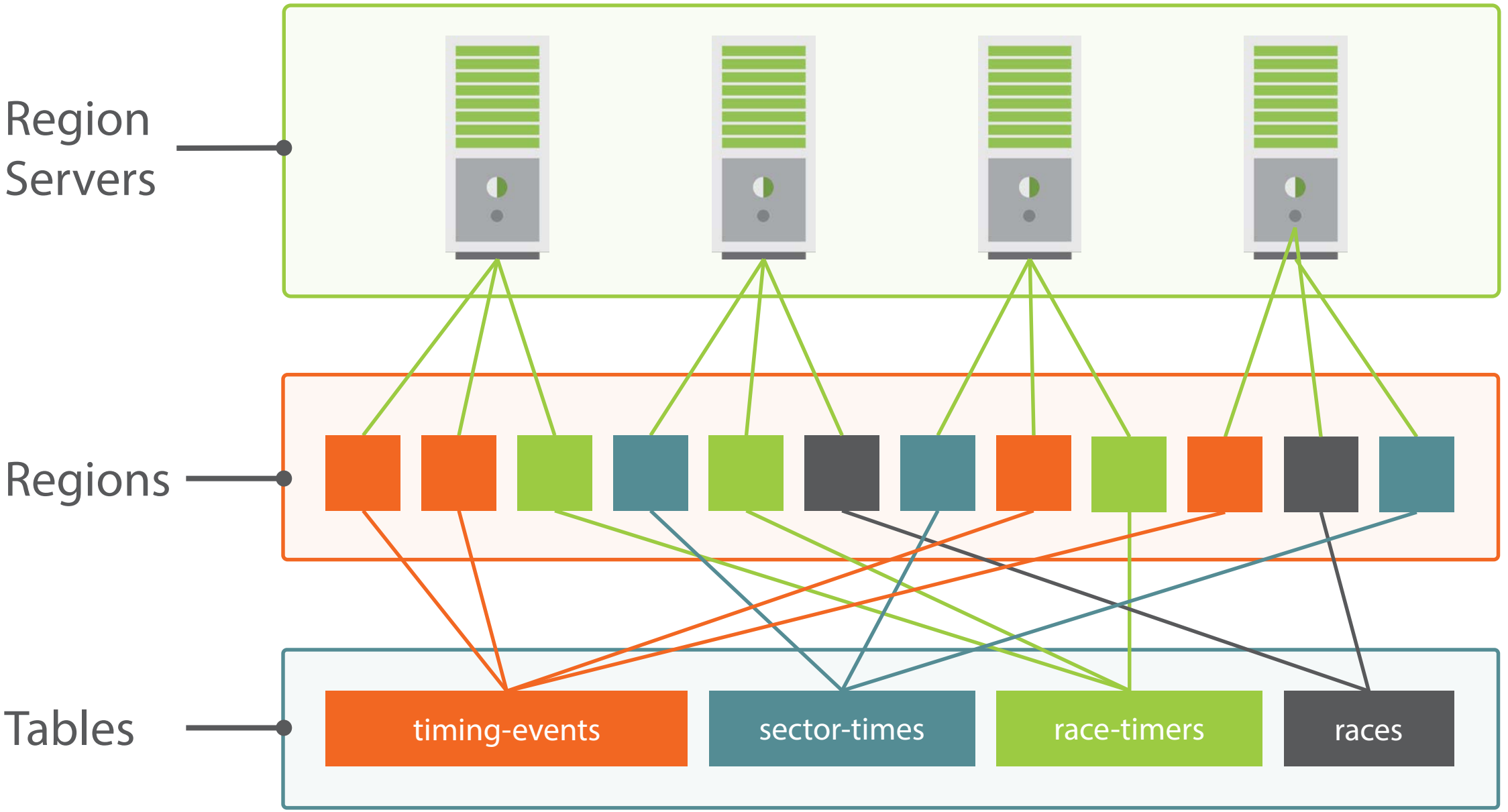
Demo: Regions and Servers

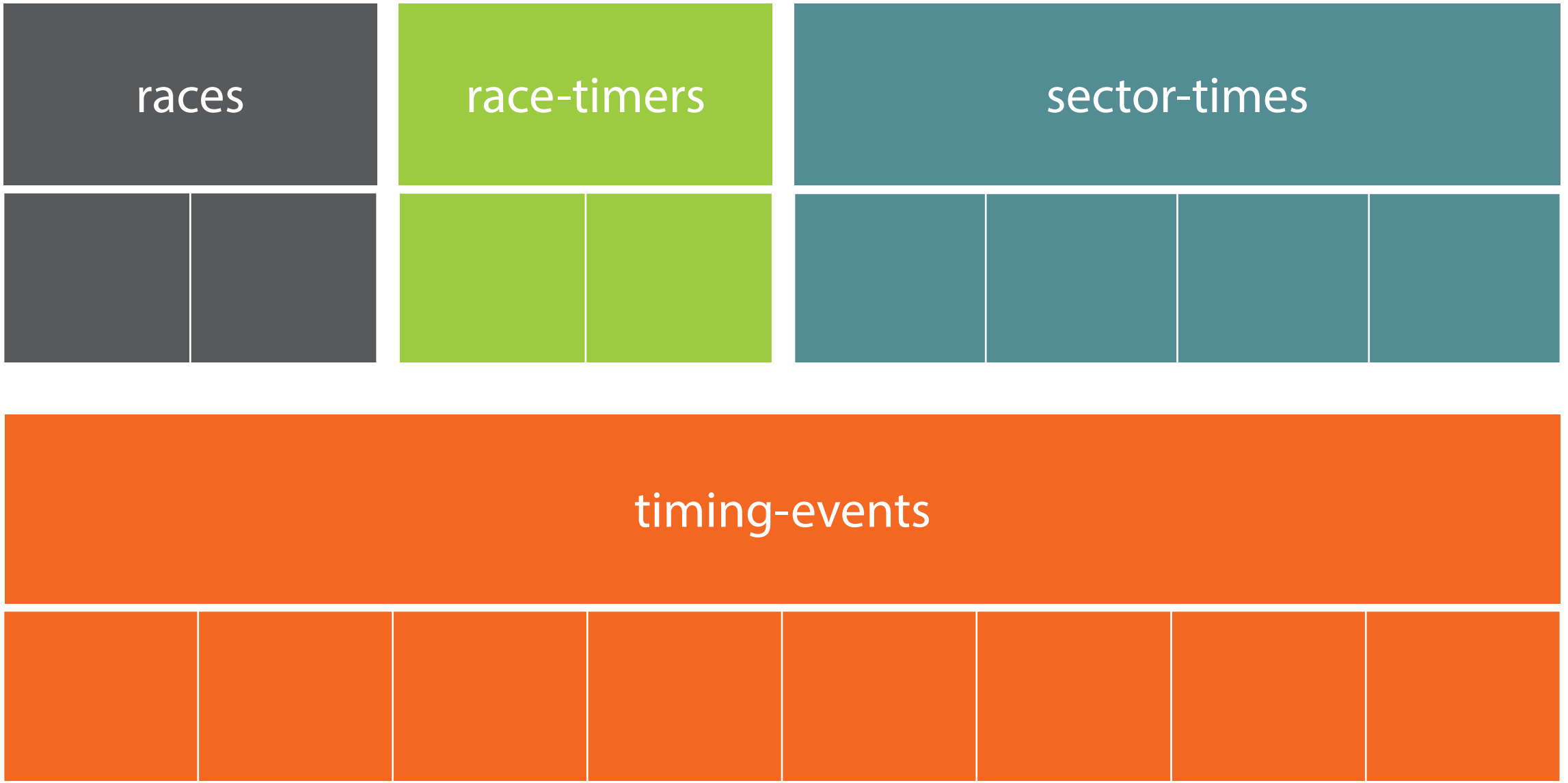
Region Splits

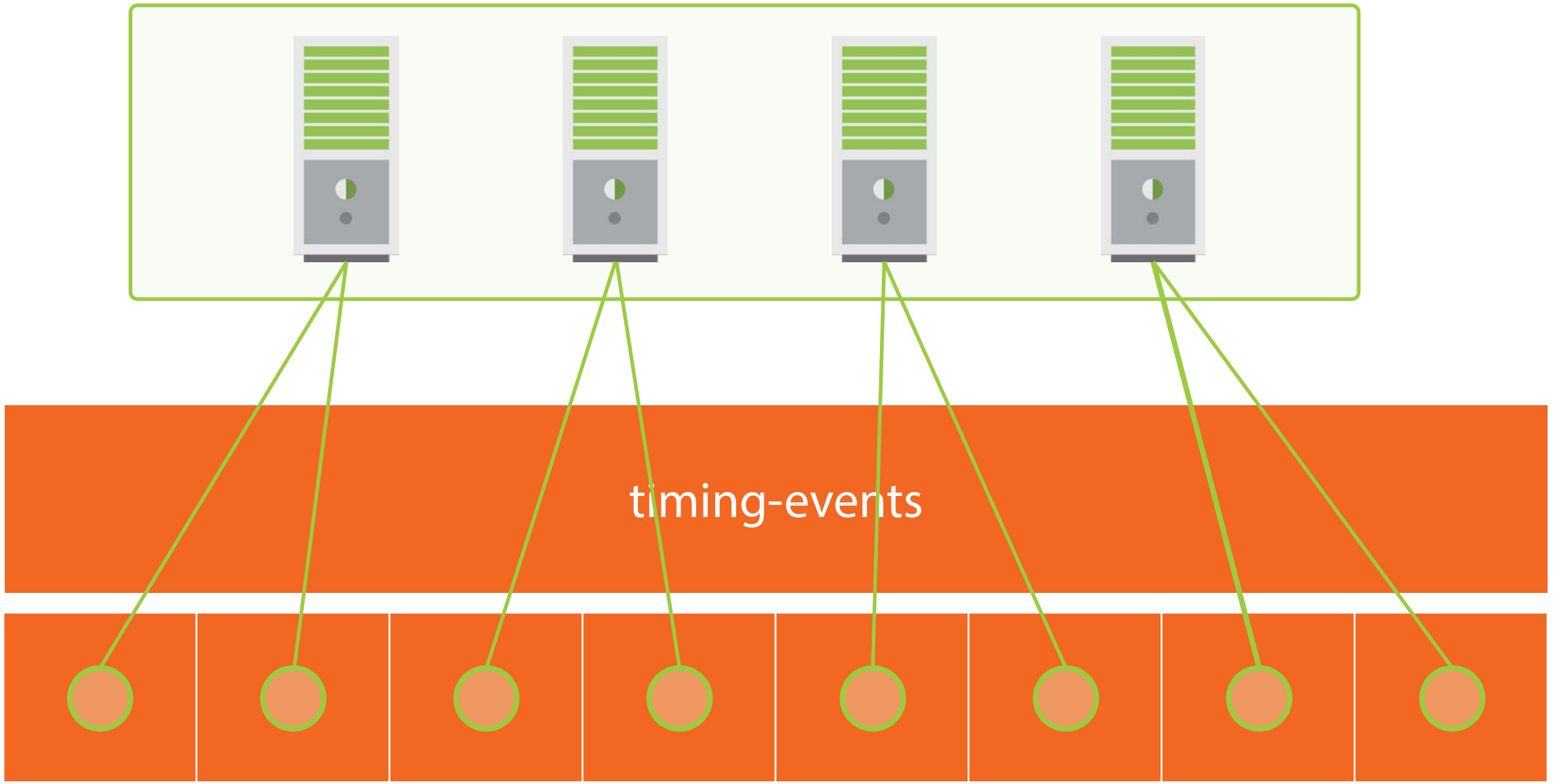
Region Distribution

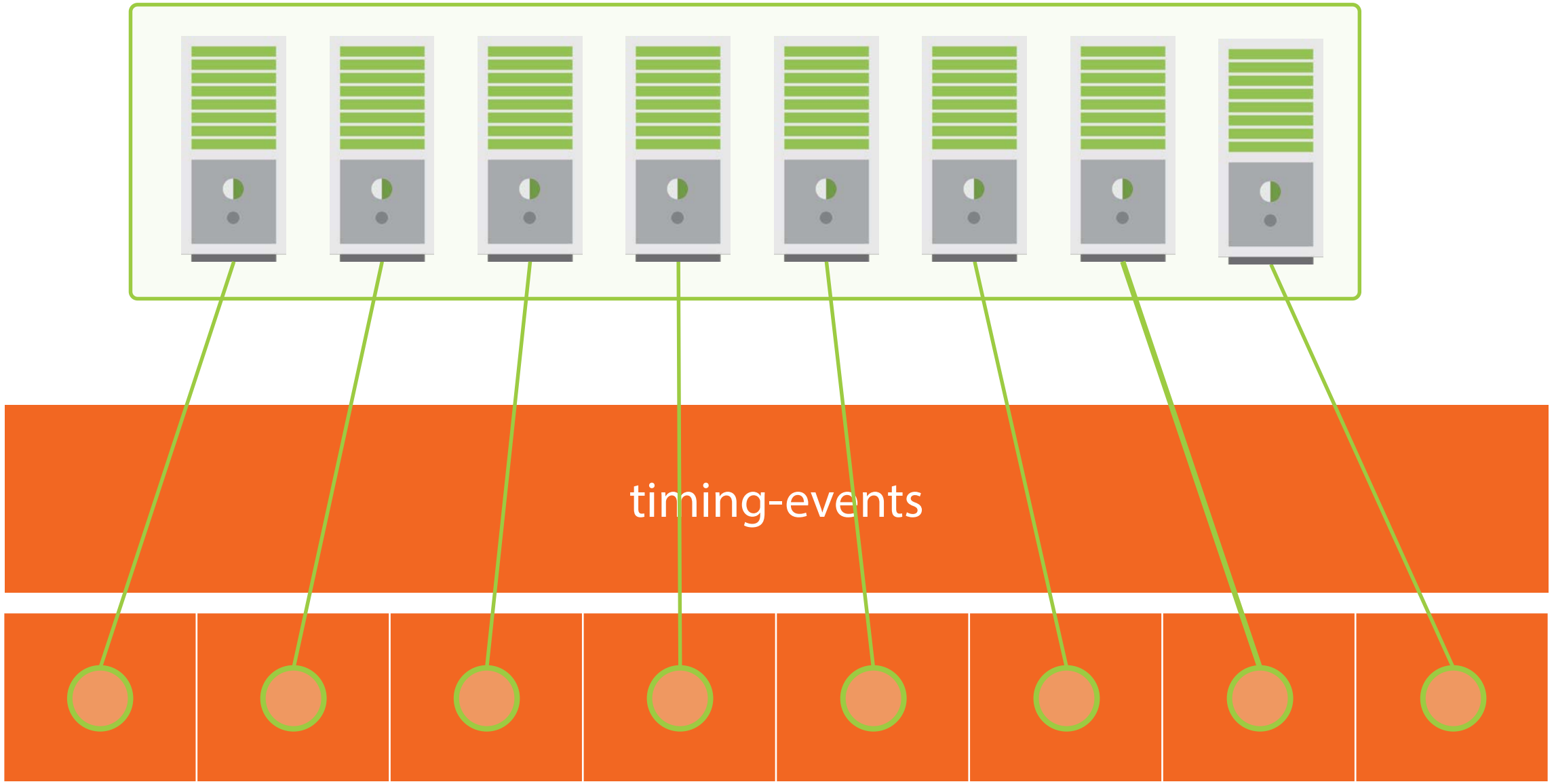
Region Server Utilisation

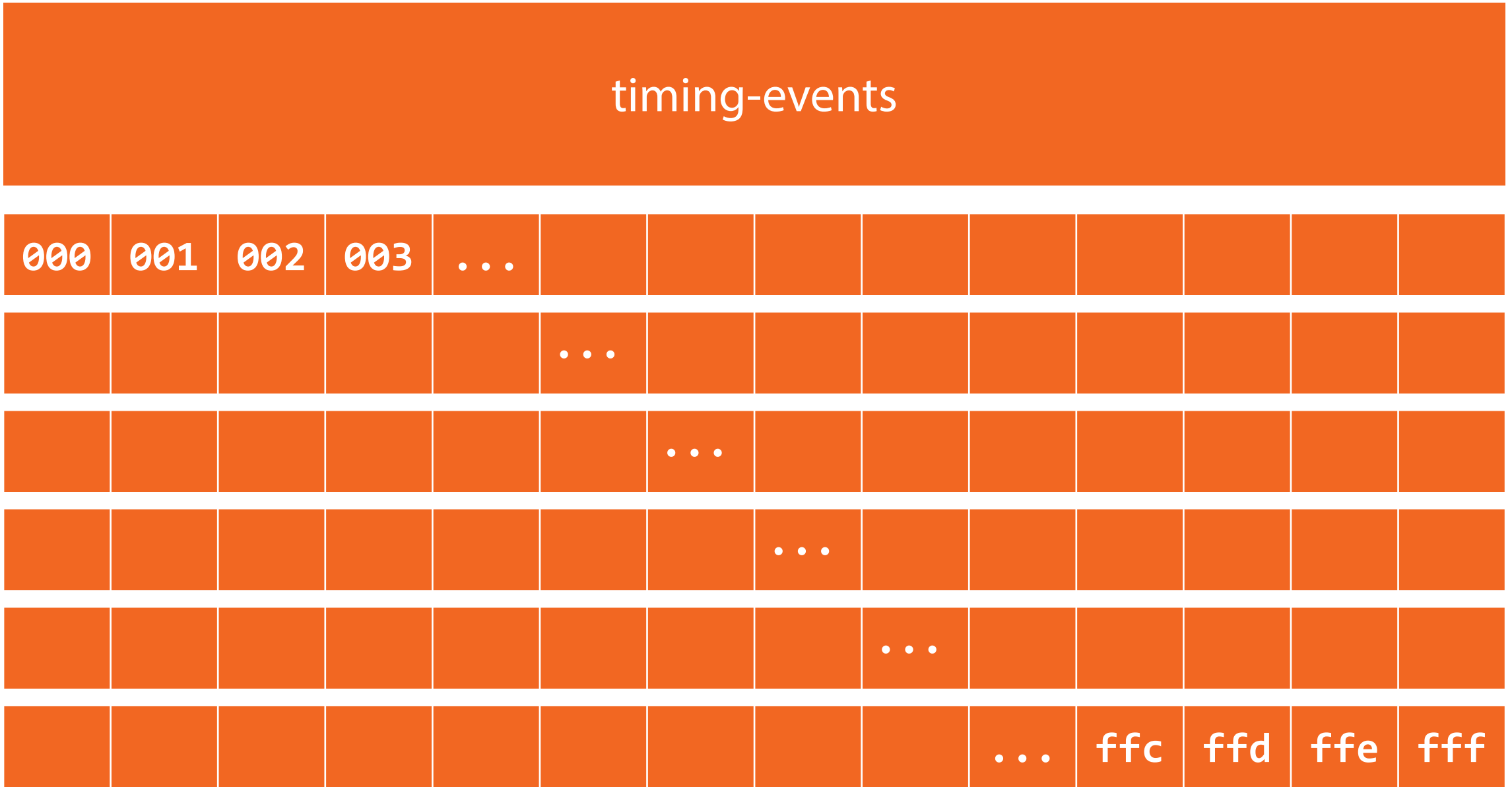


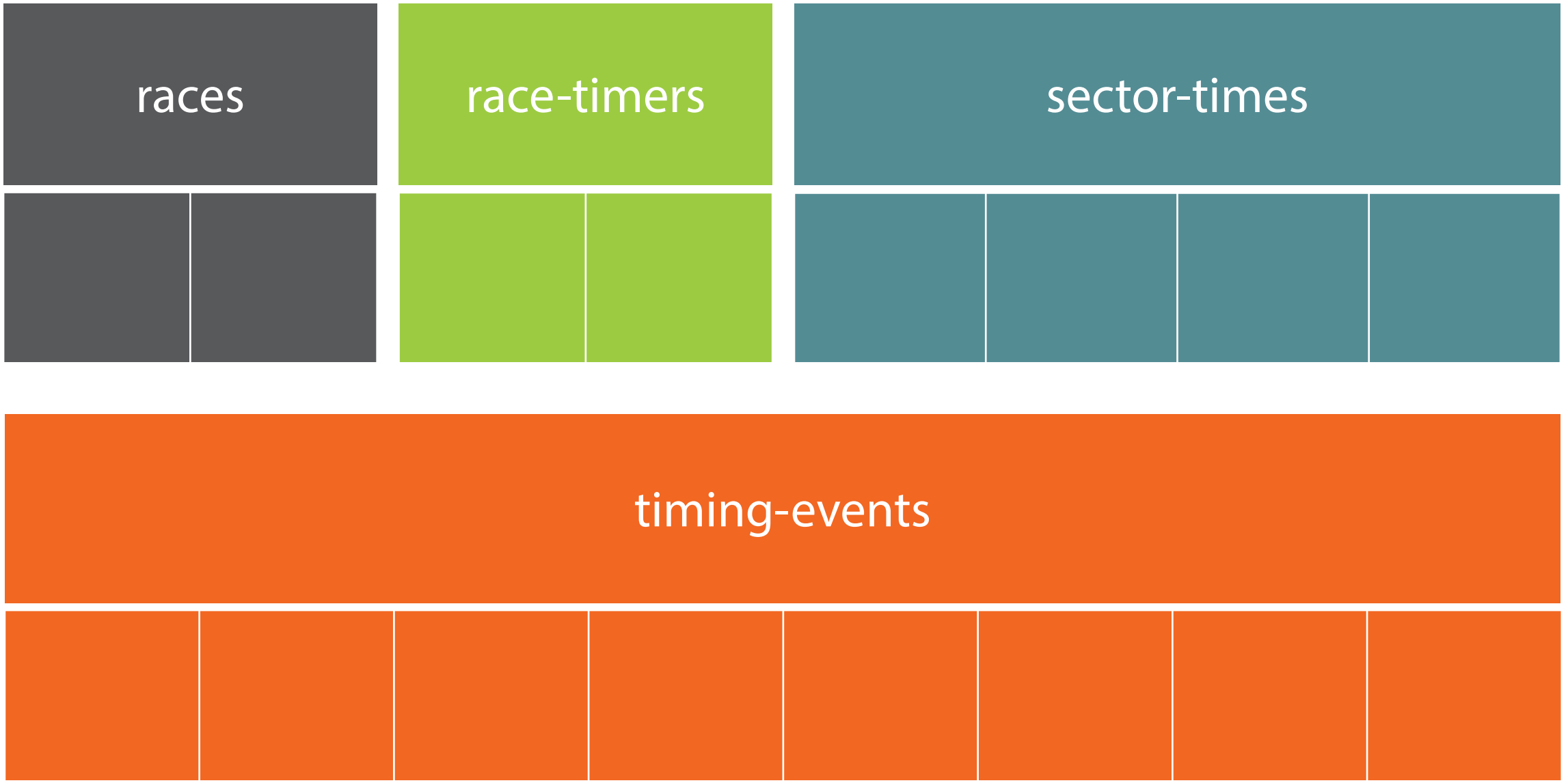

Region Servers
Regions
Tables
timing-events
sector-times
race-timers
races








timing-events
000
001
002
003
...
...
...
...
...
...
ffc
ffd
ffe
fff

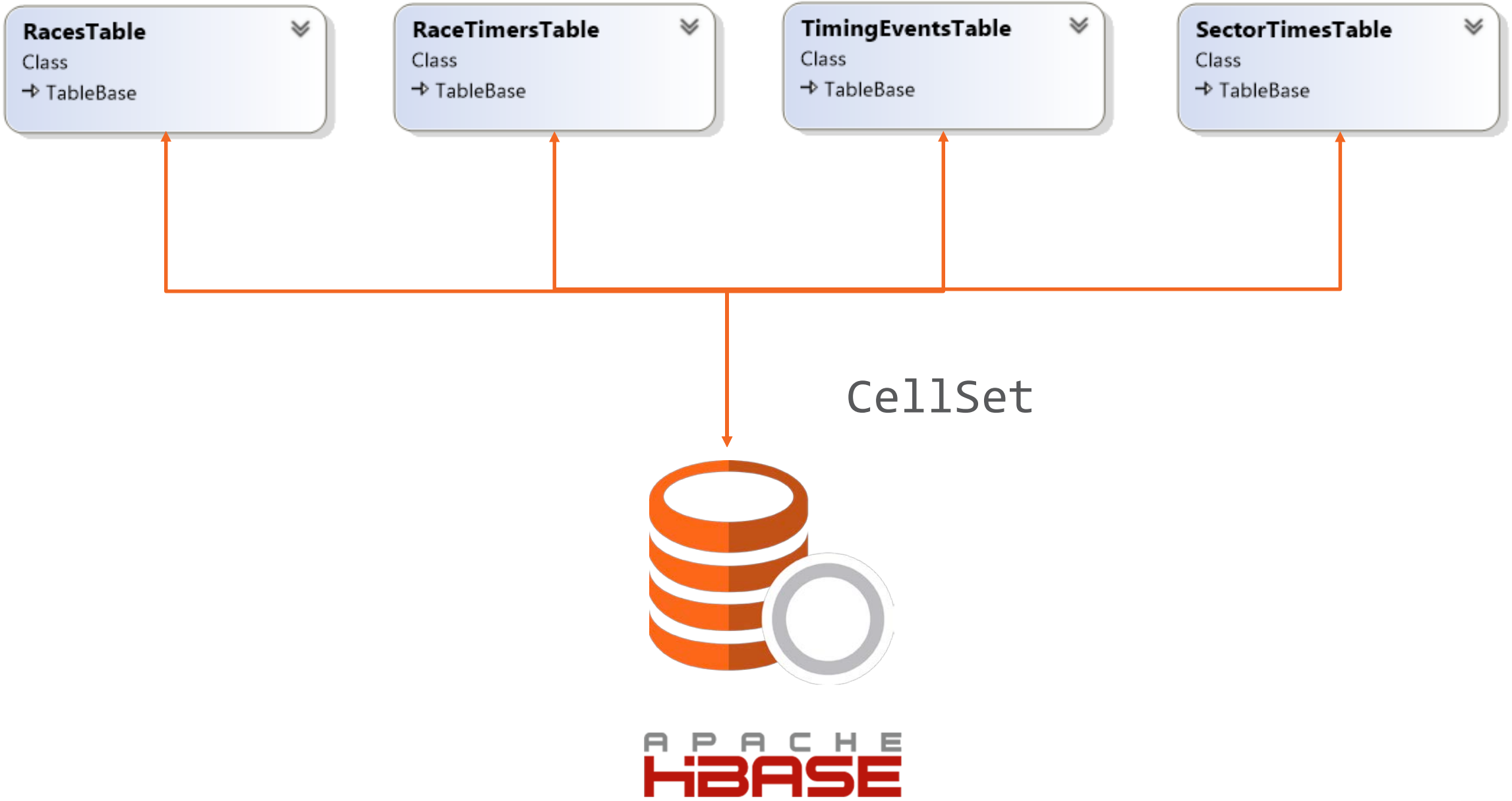


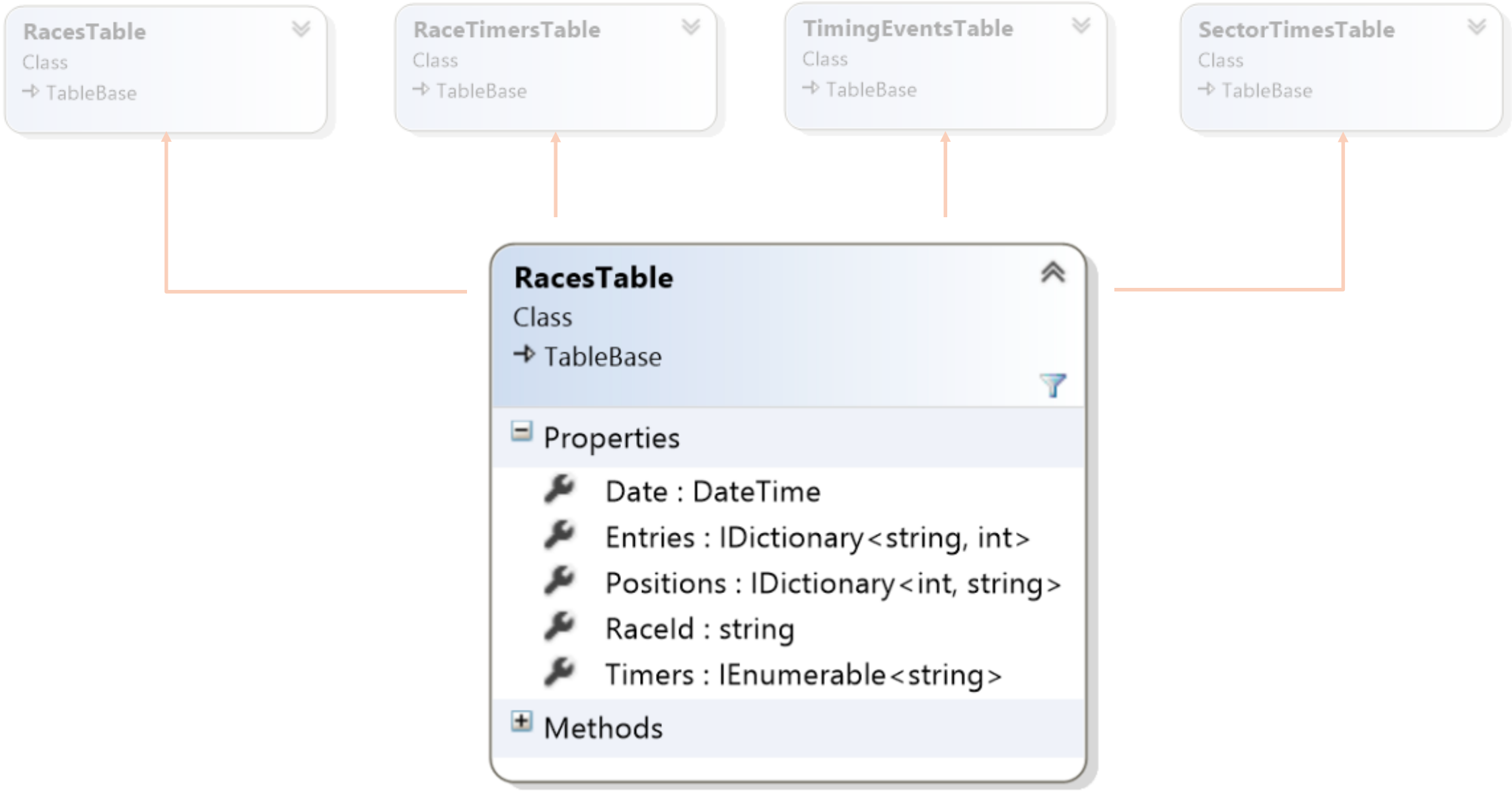
Demo: HBase and .NET

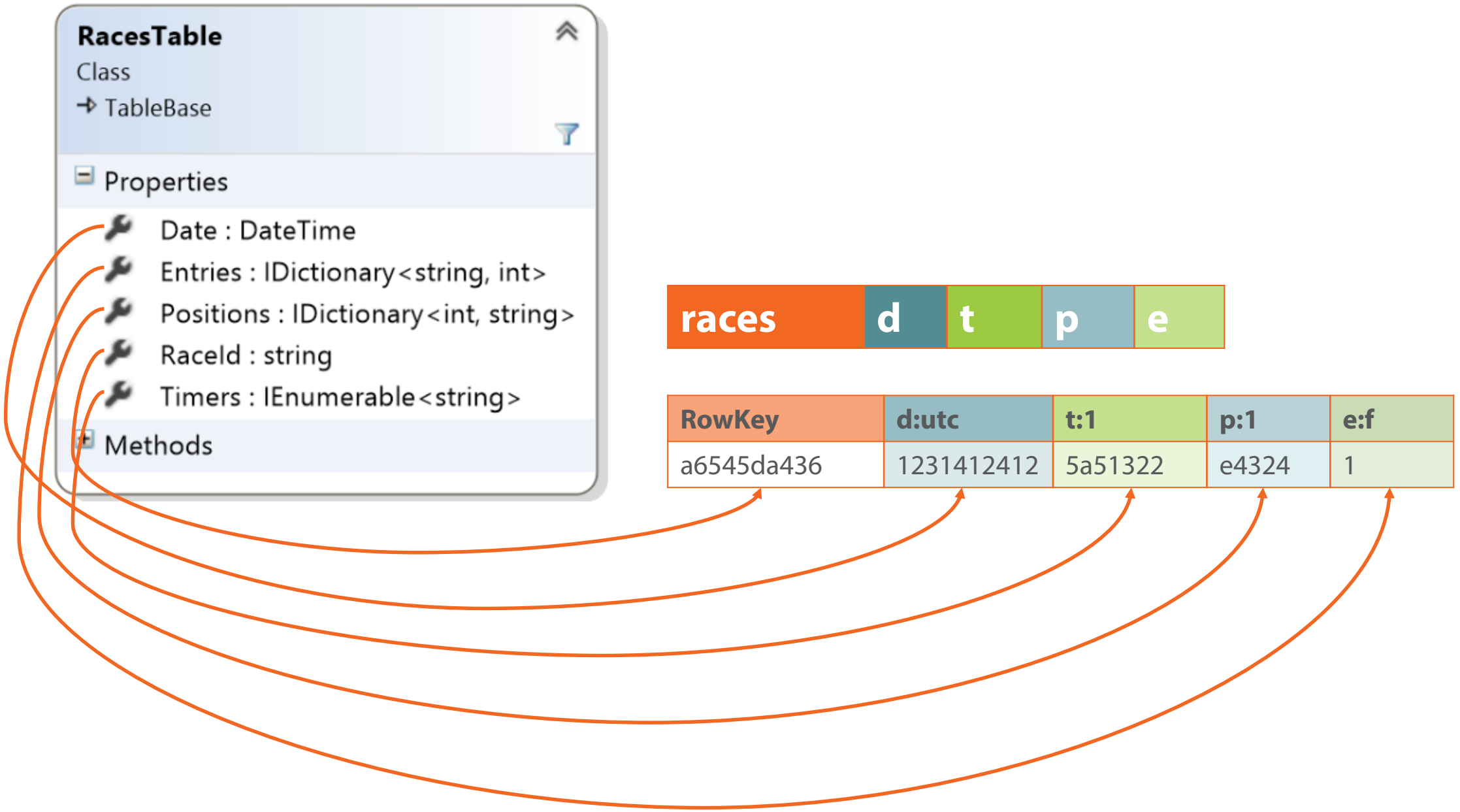
Best Practices

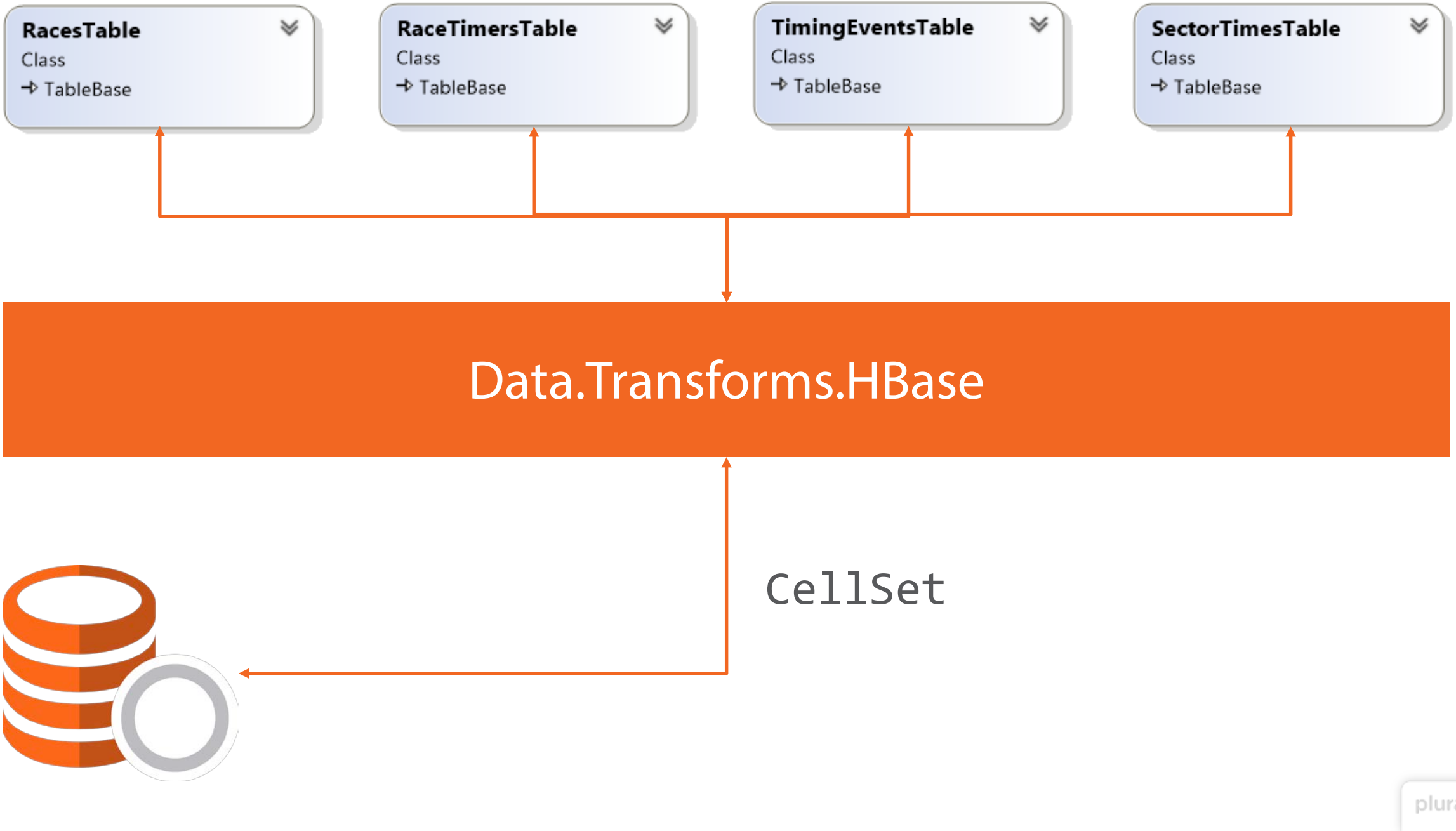
Modelling Schemas

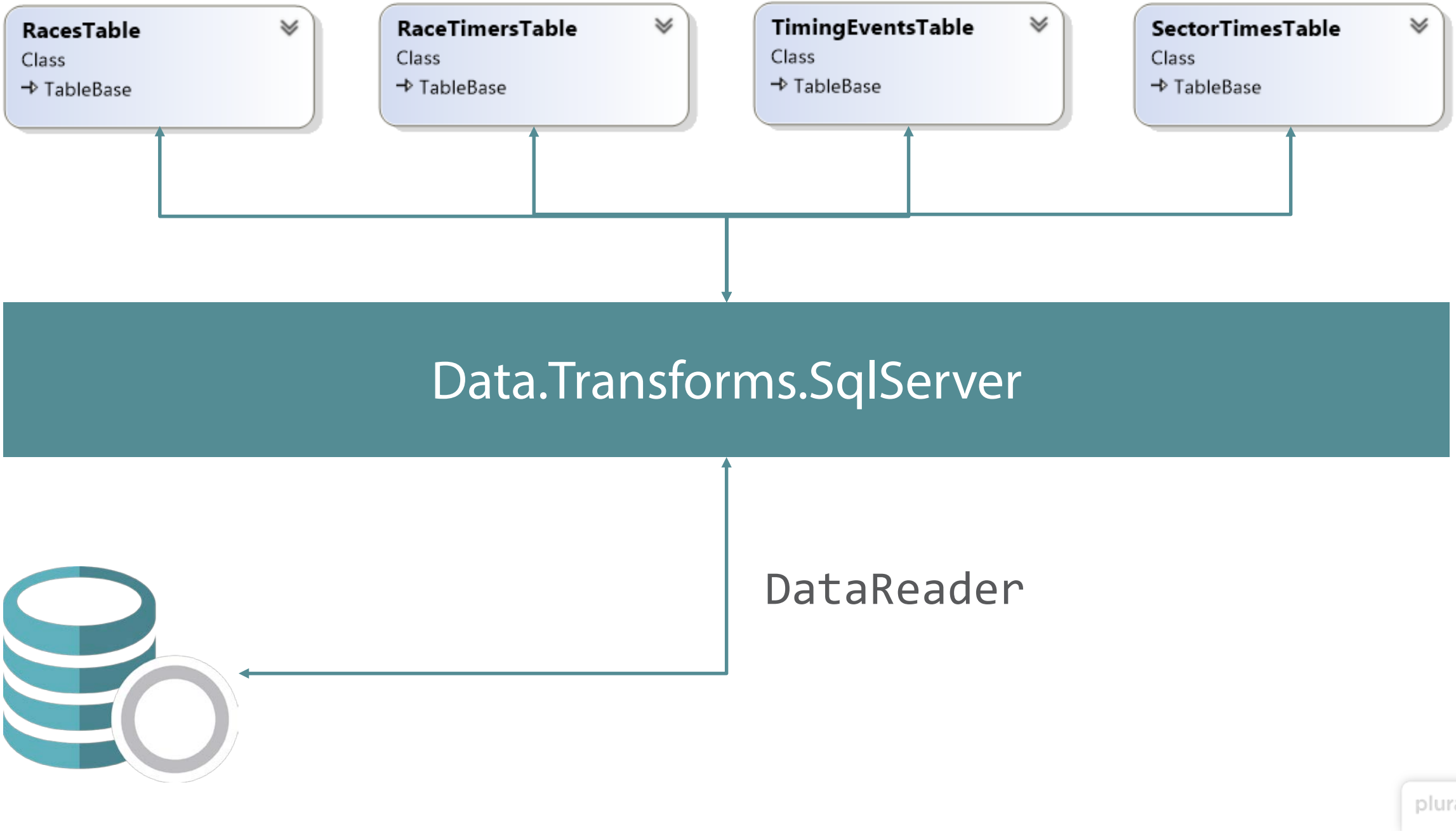
Modelling Access Patterns

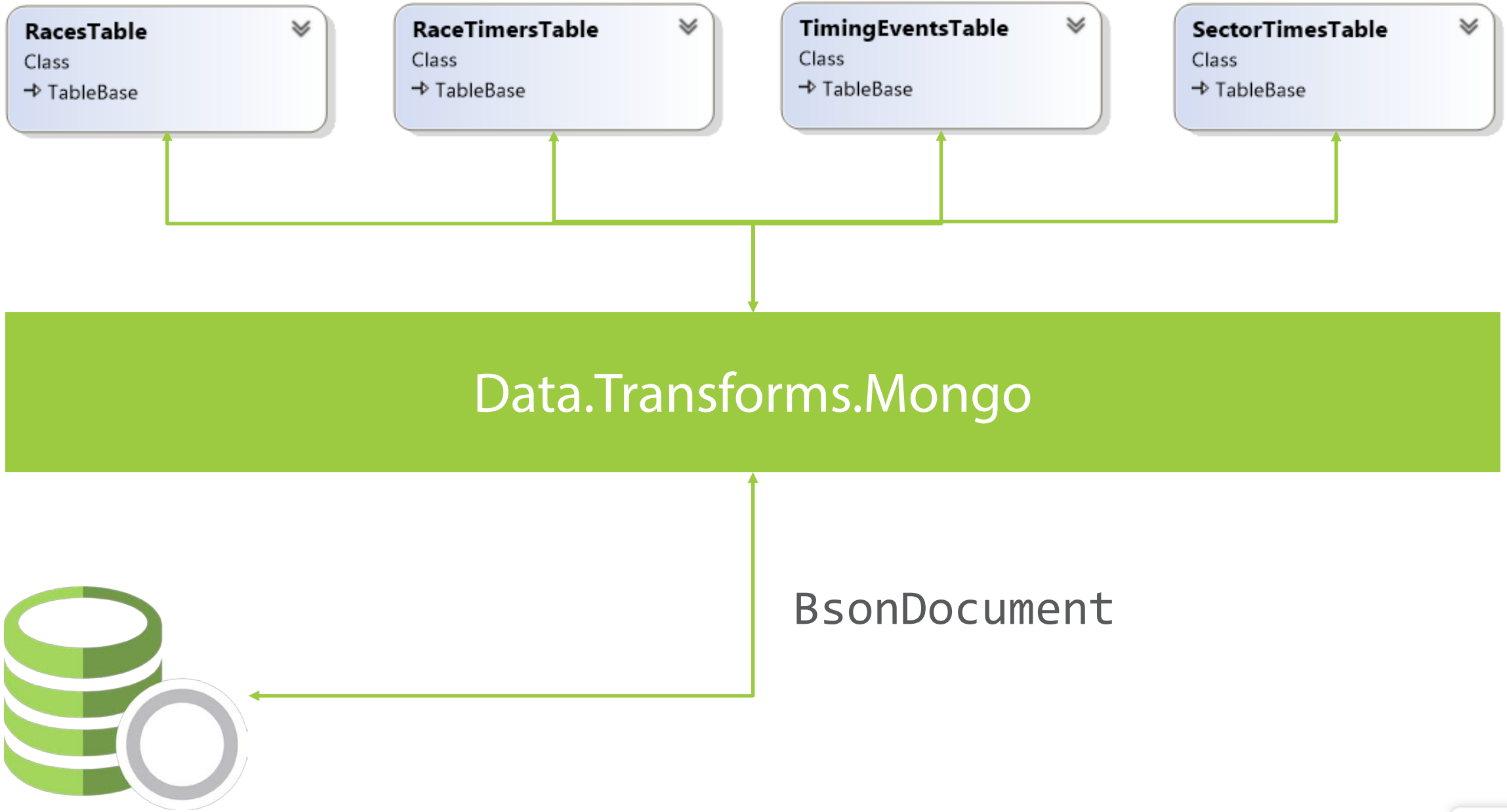


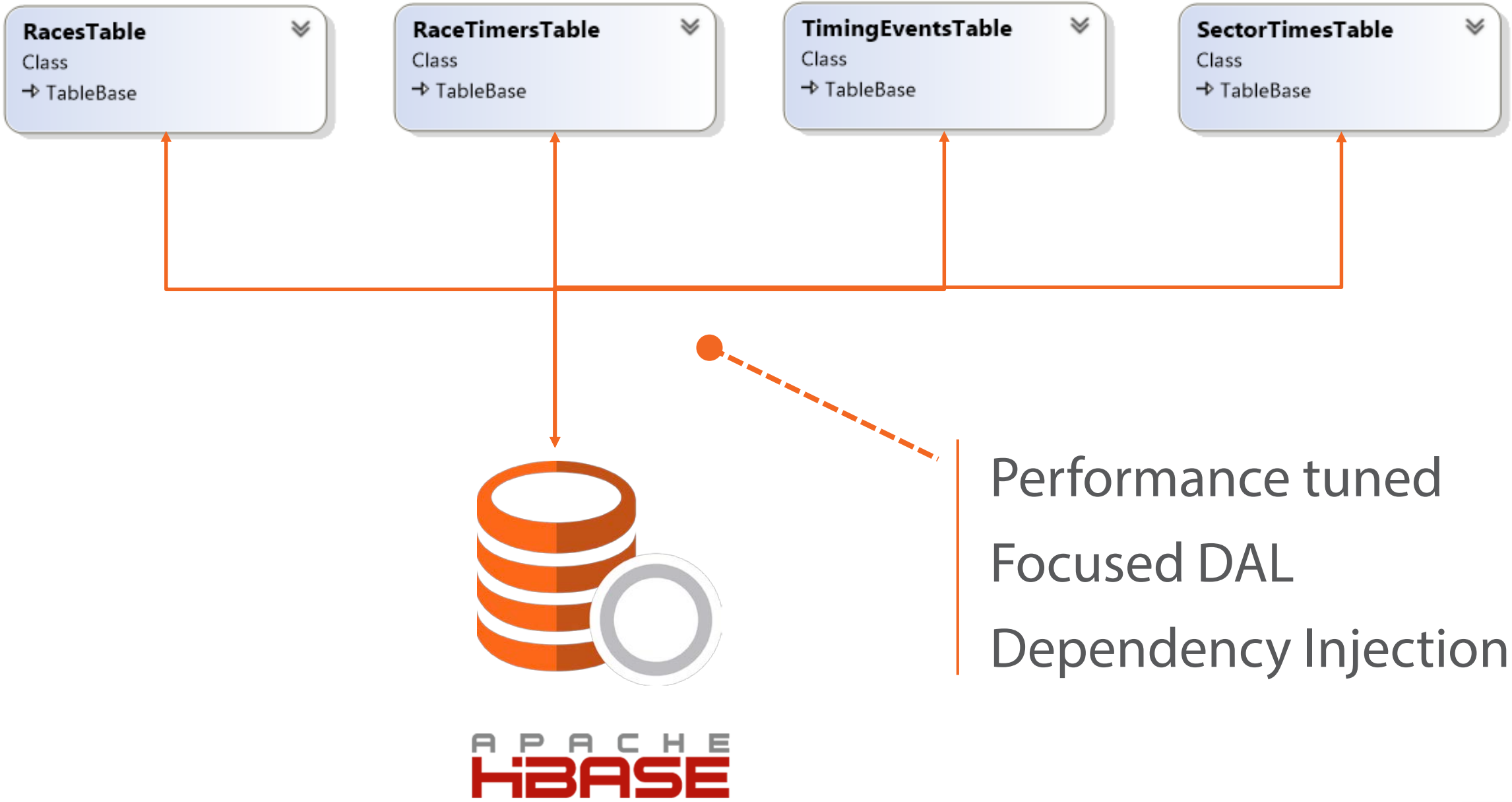



RacesTable
Class
TableBase
Properties
Date : DateTime
Entries : IDictionary<string, int>
Positions : IDictionary<int, string>
RaceId : string
Timers : IEnumerable<string>
Methods
races
d
t
p
e
RowKey
d:utc
t:1
p:1
e:f
a6545da436
1231412412
5a51322
e4324
1


RacesTable
Class
TableBase
RaceTimersTable
Class
TableBase
TimingEventsTable
Class
TableBase
SectorTimesTable
Class
TableBase
Data.Transforms.HBase
CellSet


RacesTable
Class
TableBase
RaceTimersTable
Class
TableBase
TimingEventsTable
Class
TableBase
SectorTimesTable
Class
TableBase
Data.Transforms.SqlServer
DataReader



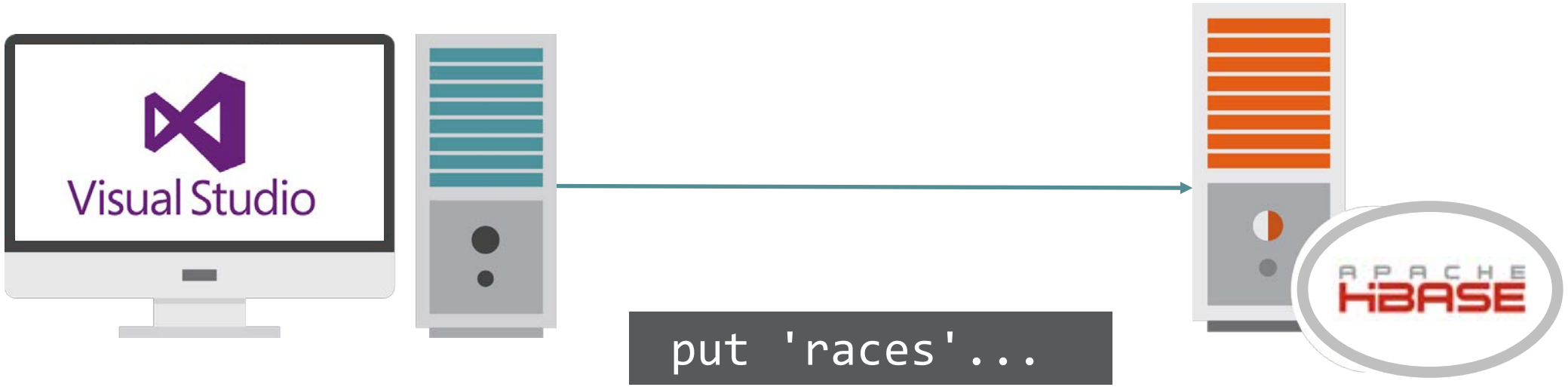


Demo: Automated Testing

Integration Tests

HBase in Docker

Mocking Stargate




```
put 'races'...
```

```
get 'races'...
```

DAL **logic**


```
Assert [duration < 1sec]
```

Mock< APACHE HBASE >

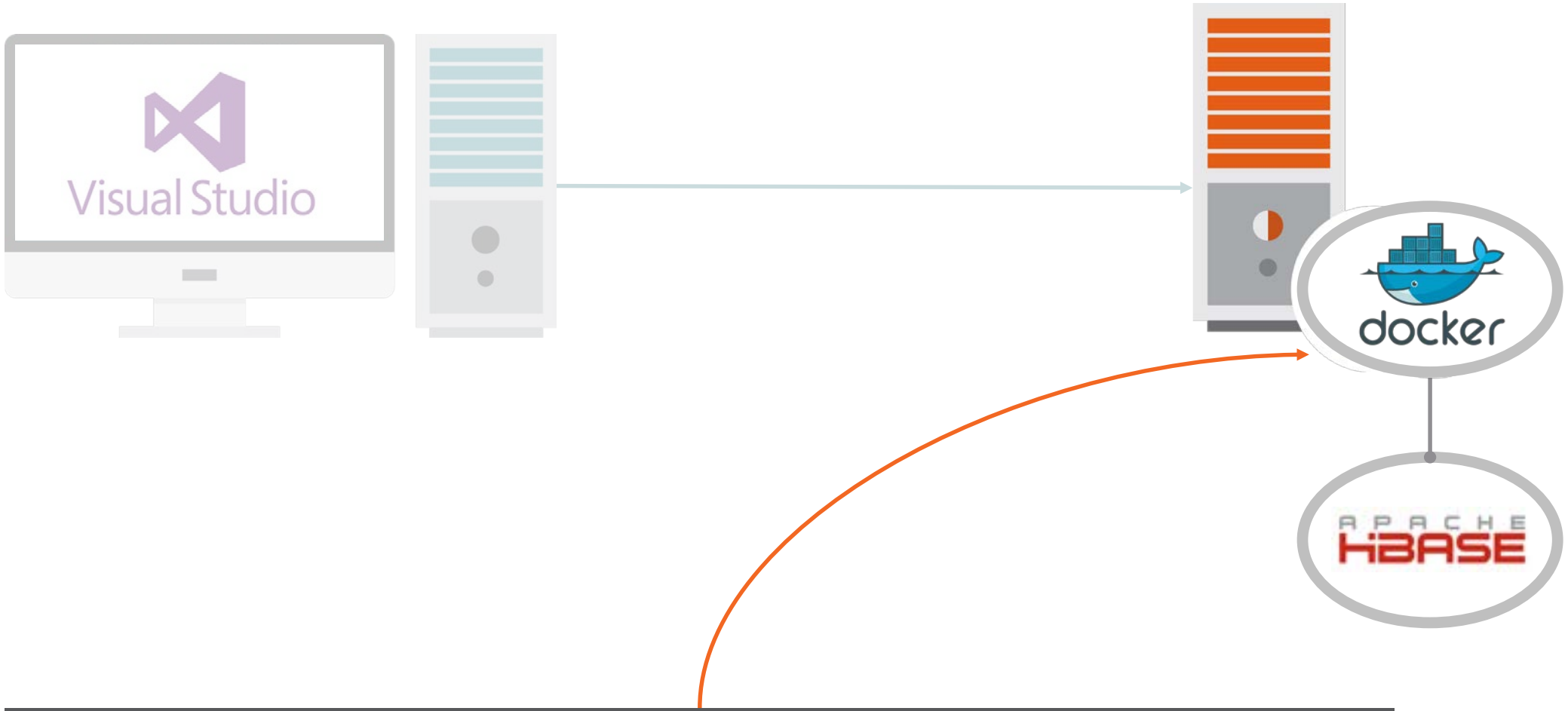
```
put 'races'...
```

```
get 'races'...
```

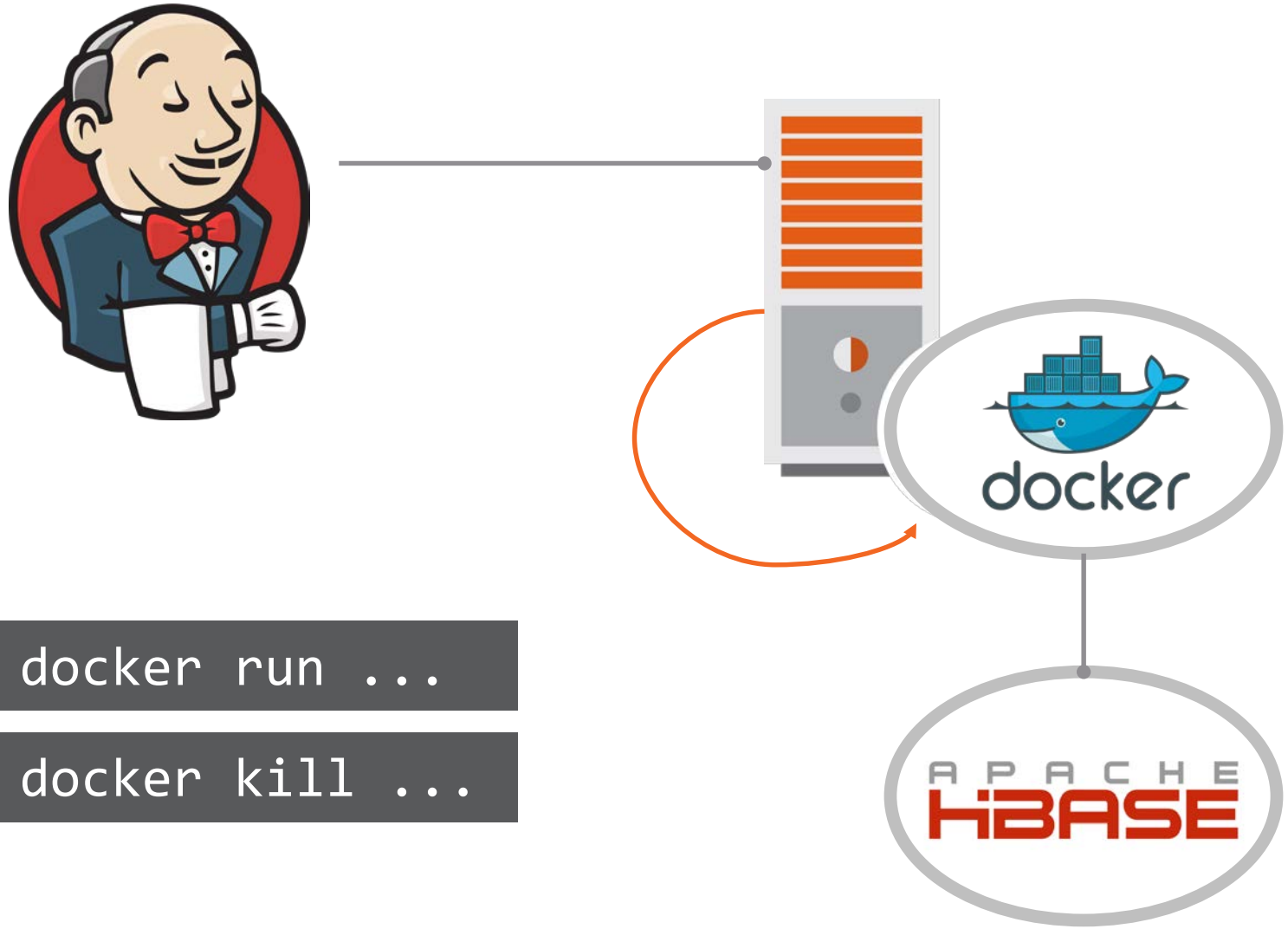
DAL **logic**

DI setup

App **config**

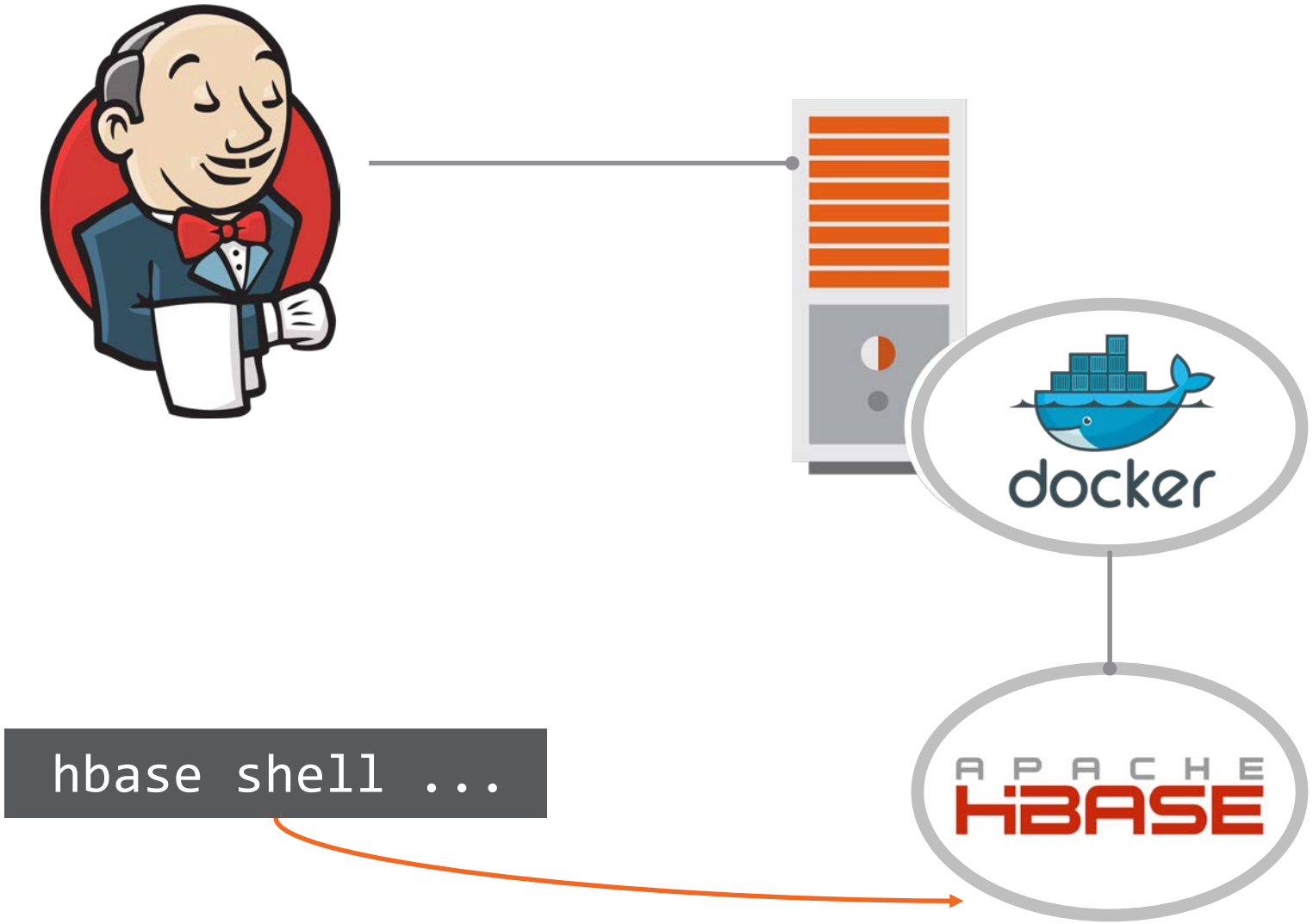



```
docker run -d -p 8080:8080 sixeyed/hbase-stargate
```




```
docker run ...
```

```
docker kill ...
```

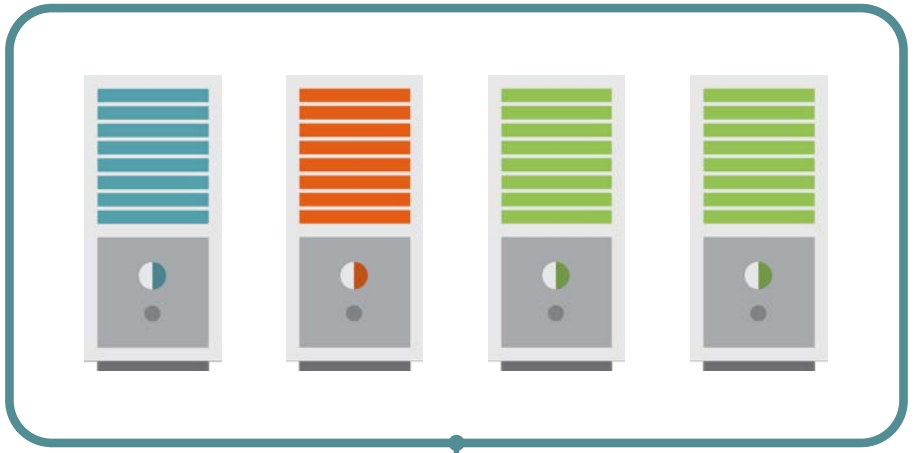


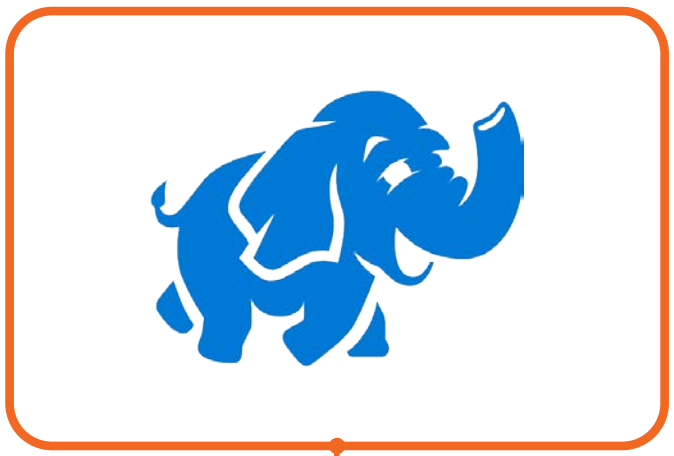

```
hbase shell ...
```

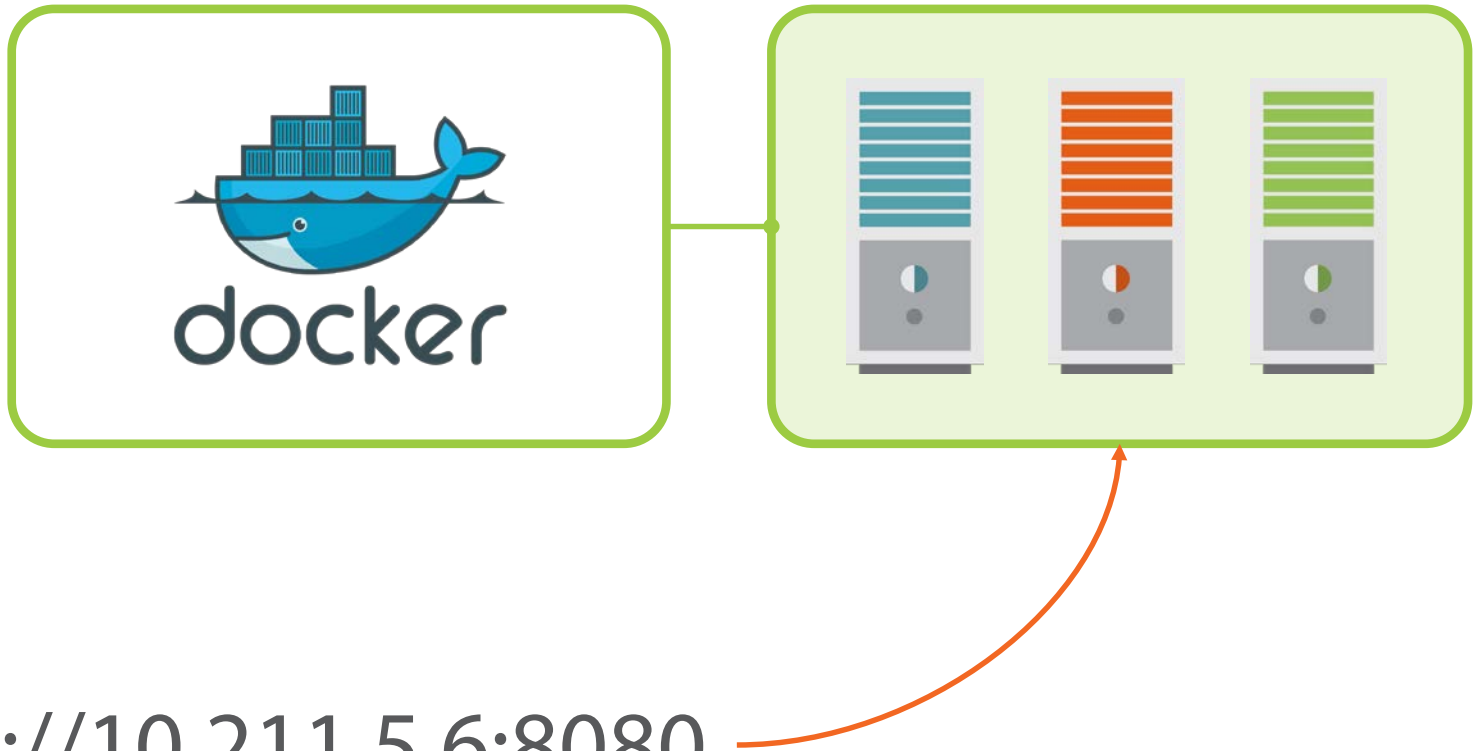
dev/build

test

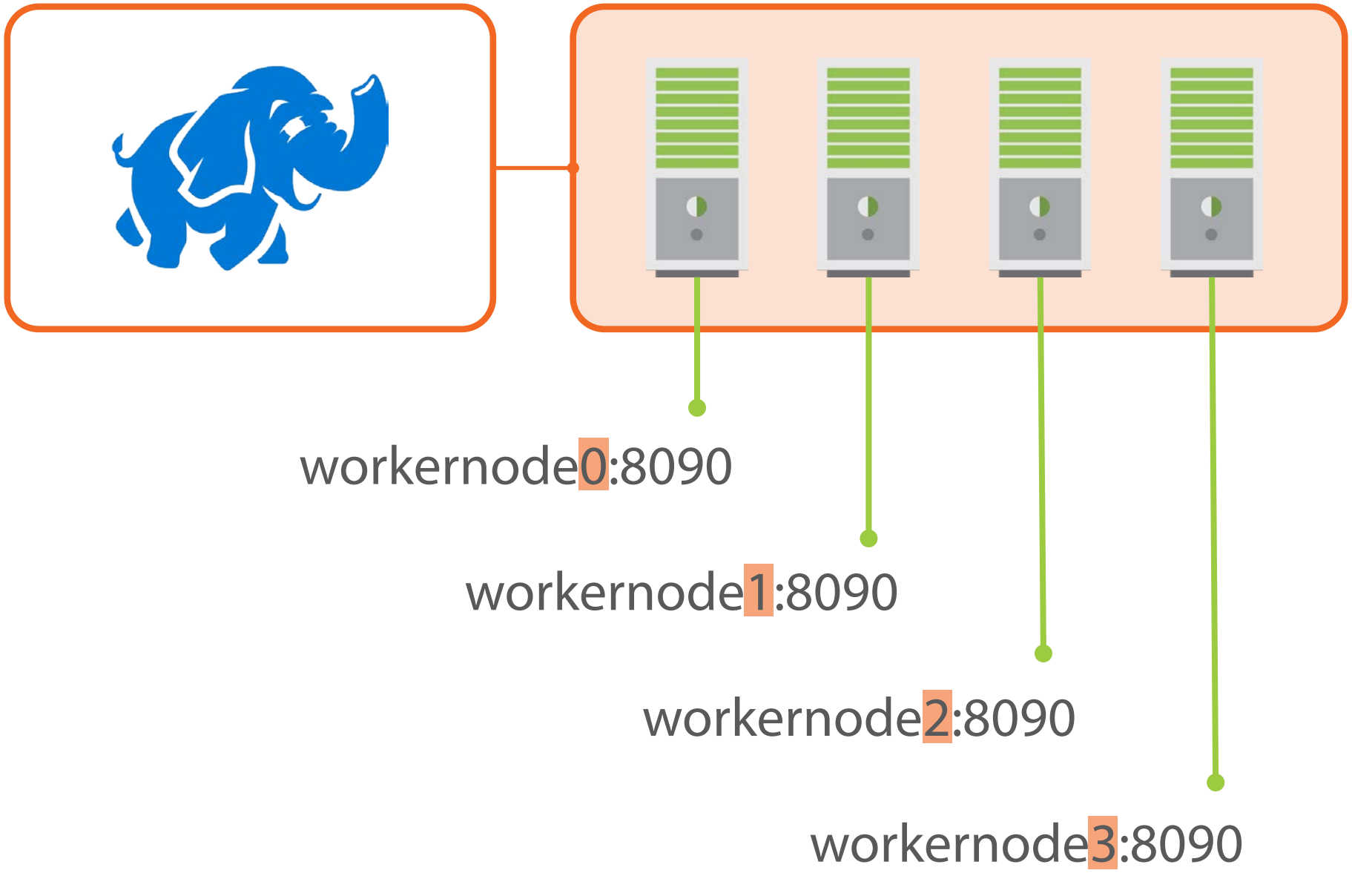
production

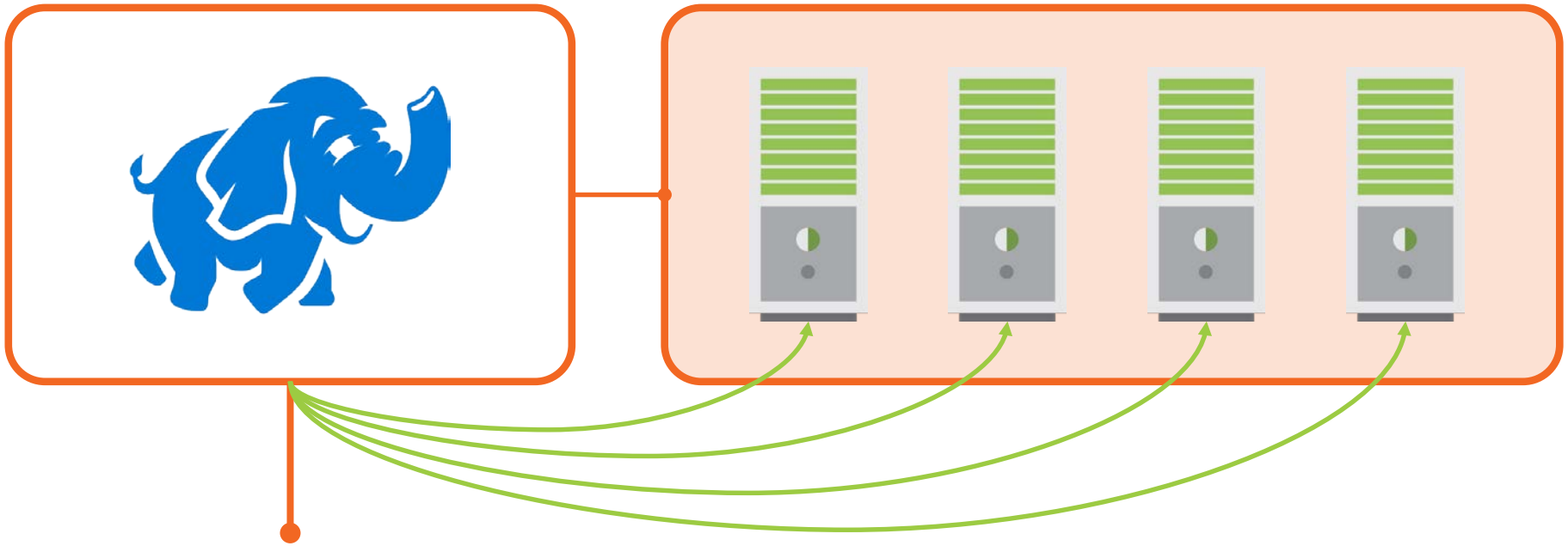




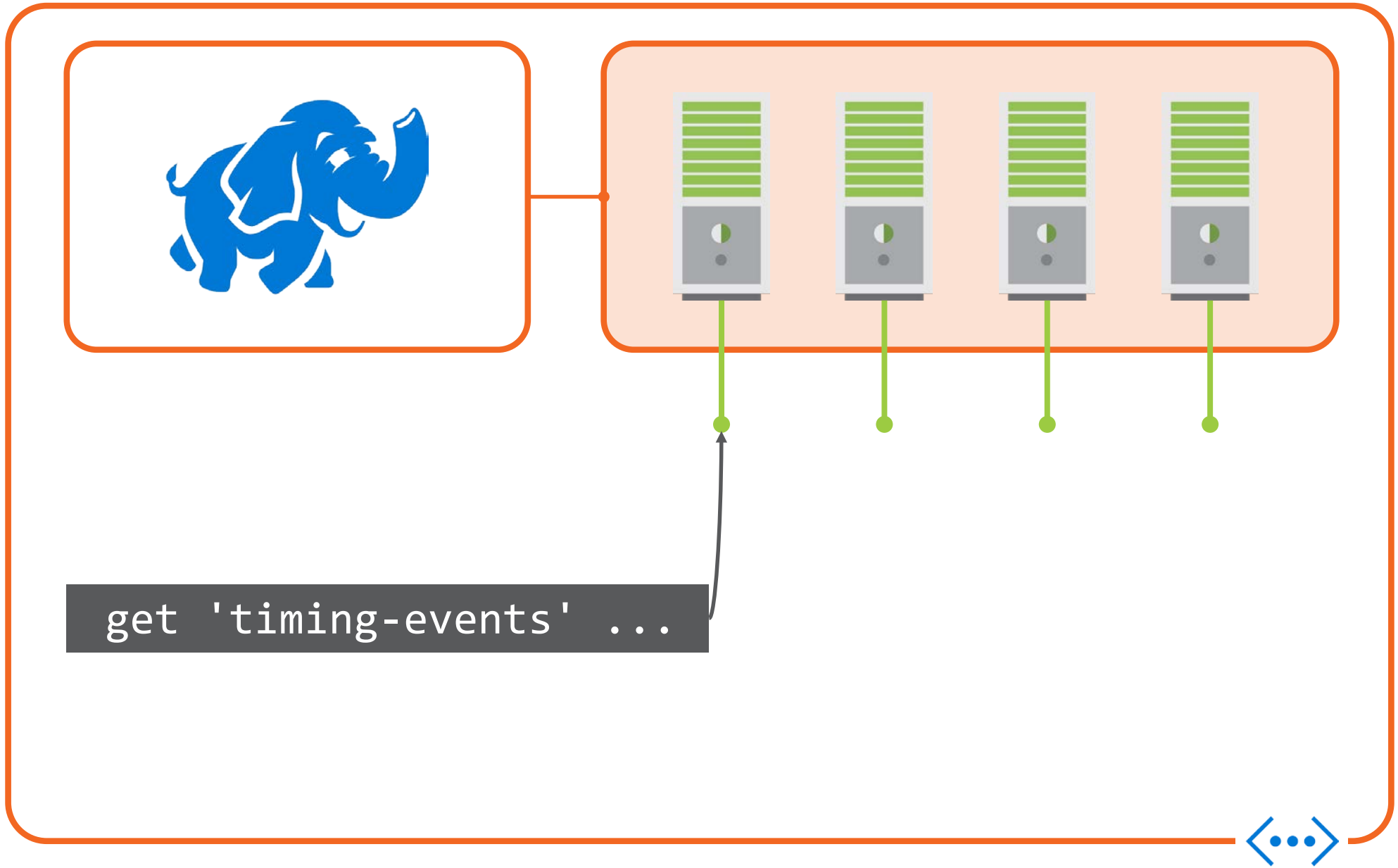


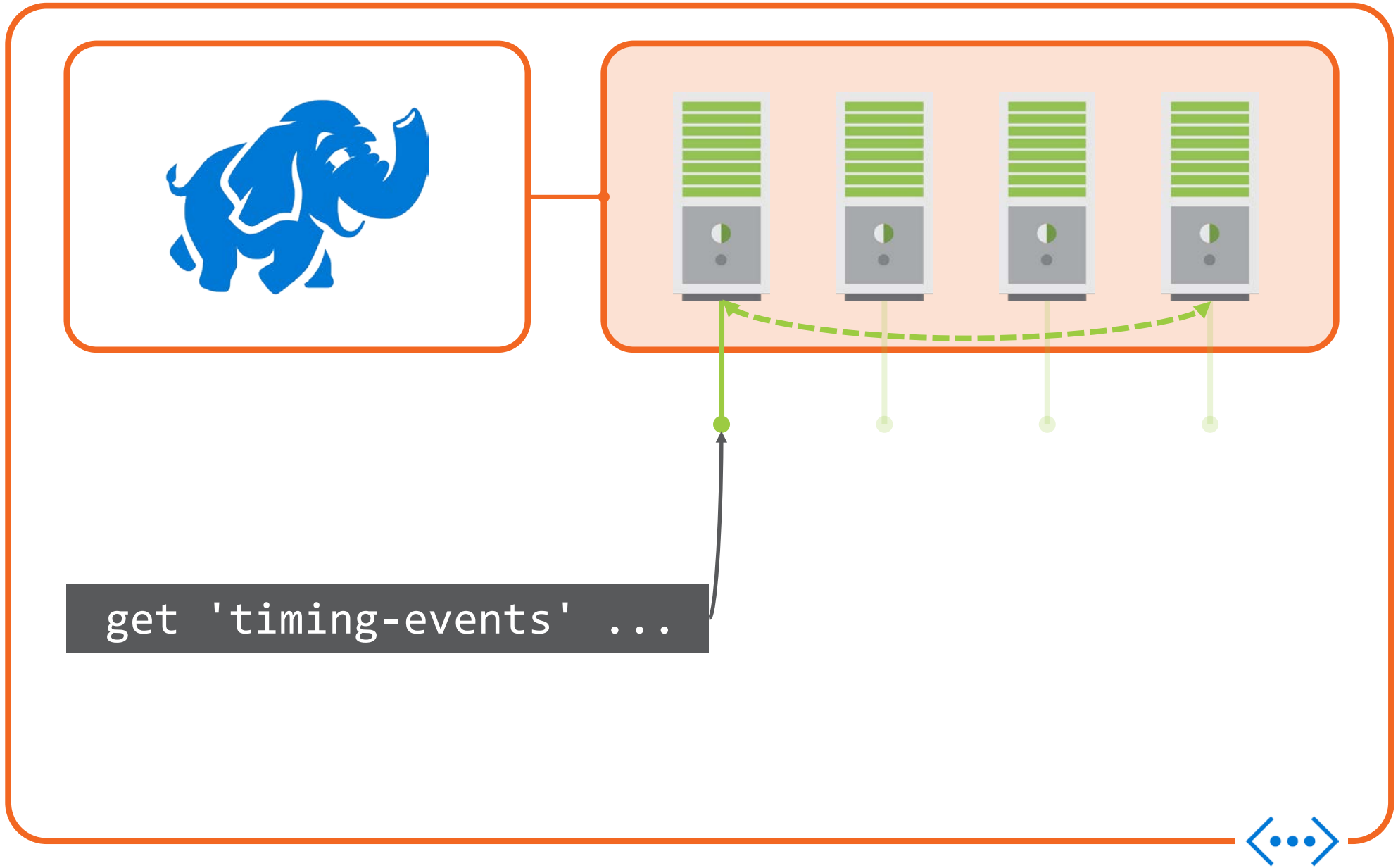
http://<?>:8080

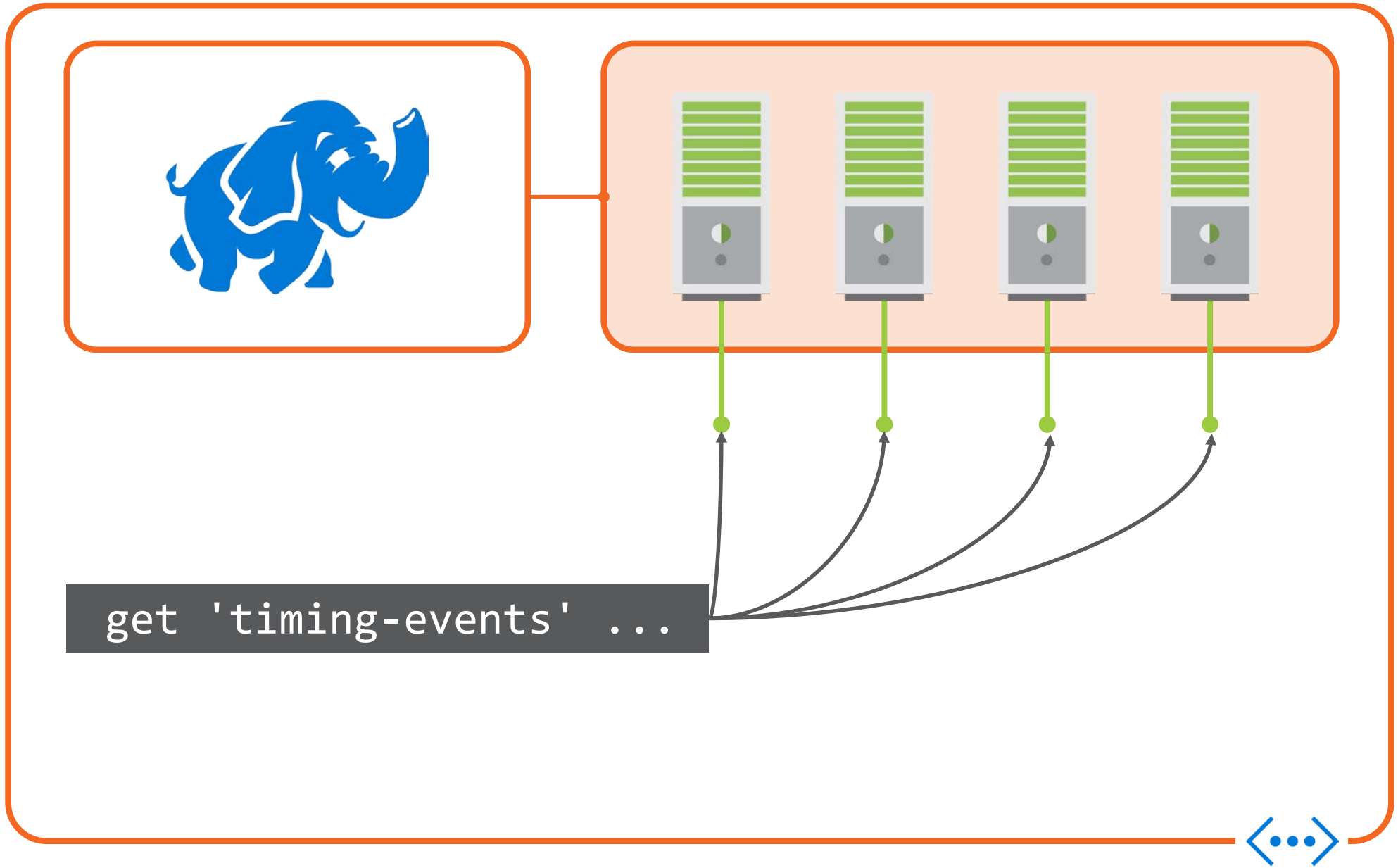


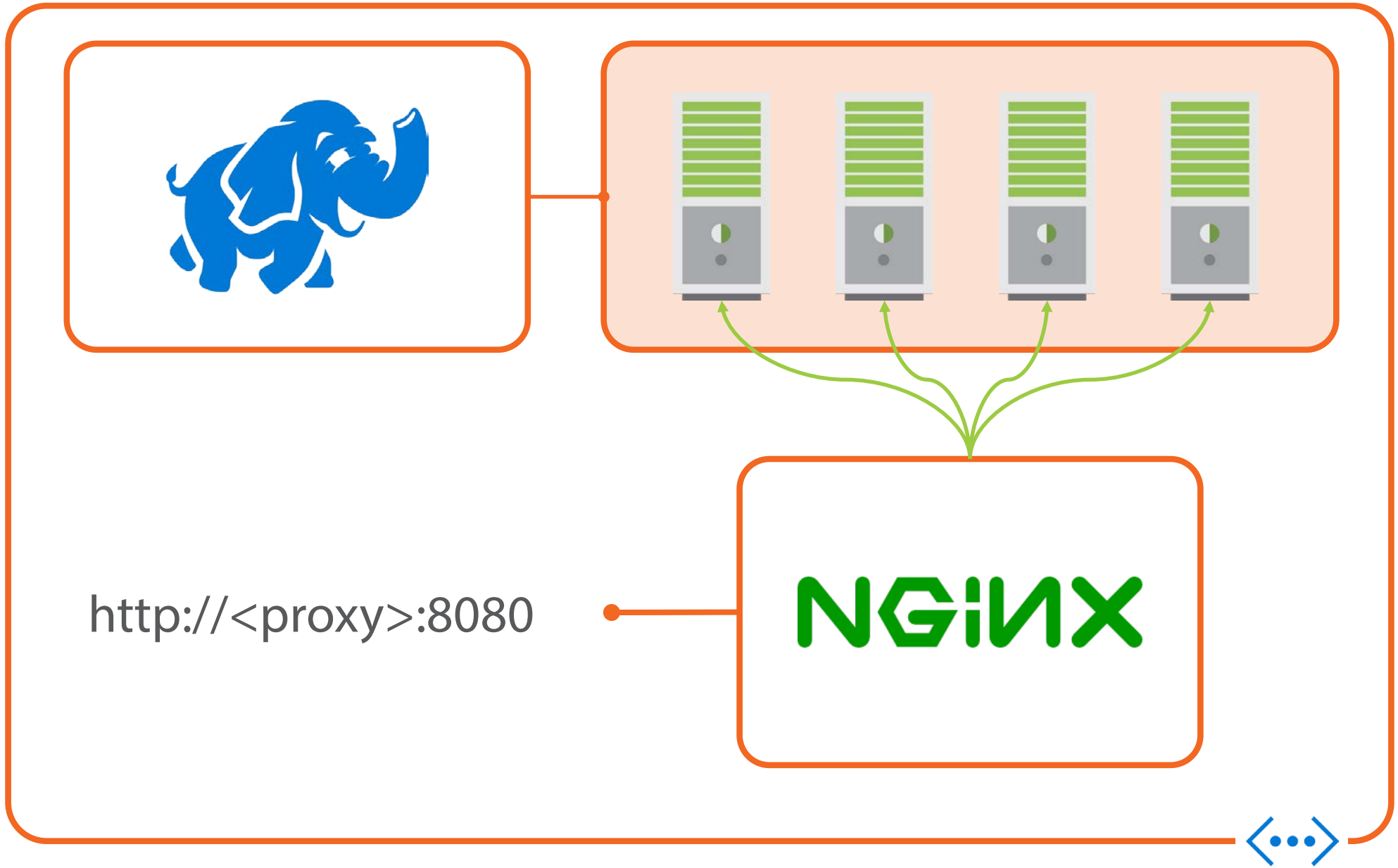


https://<cluster>:443/hbaserest



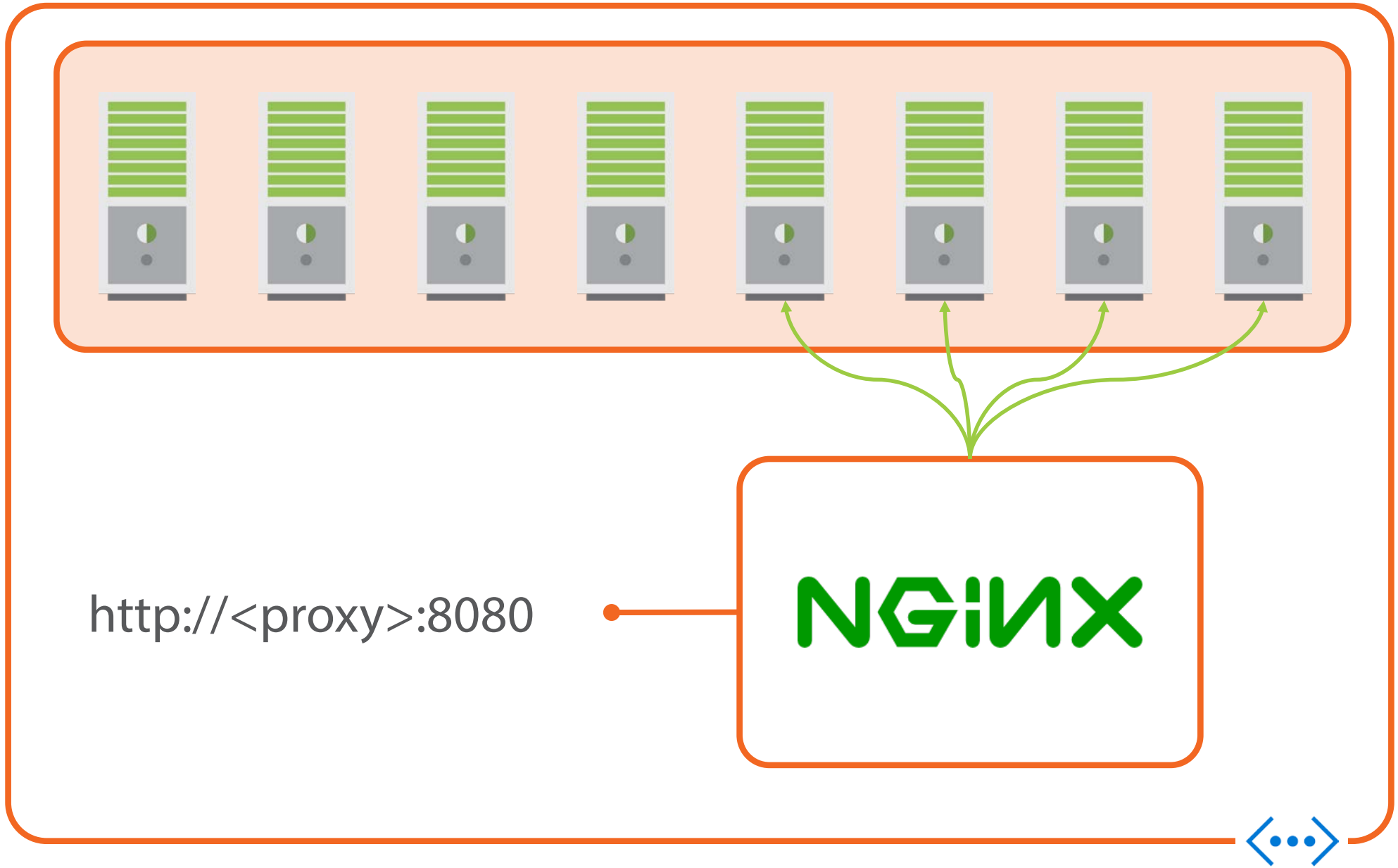


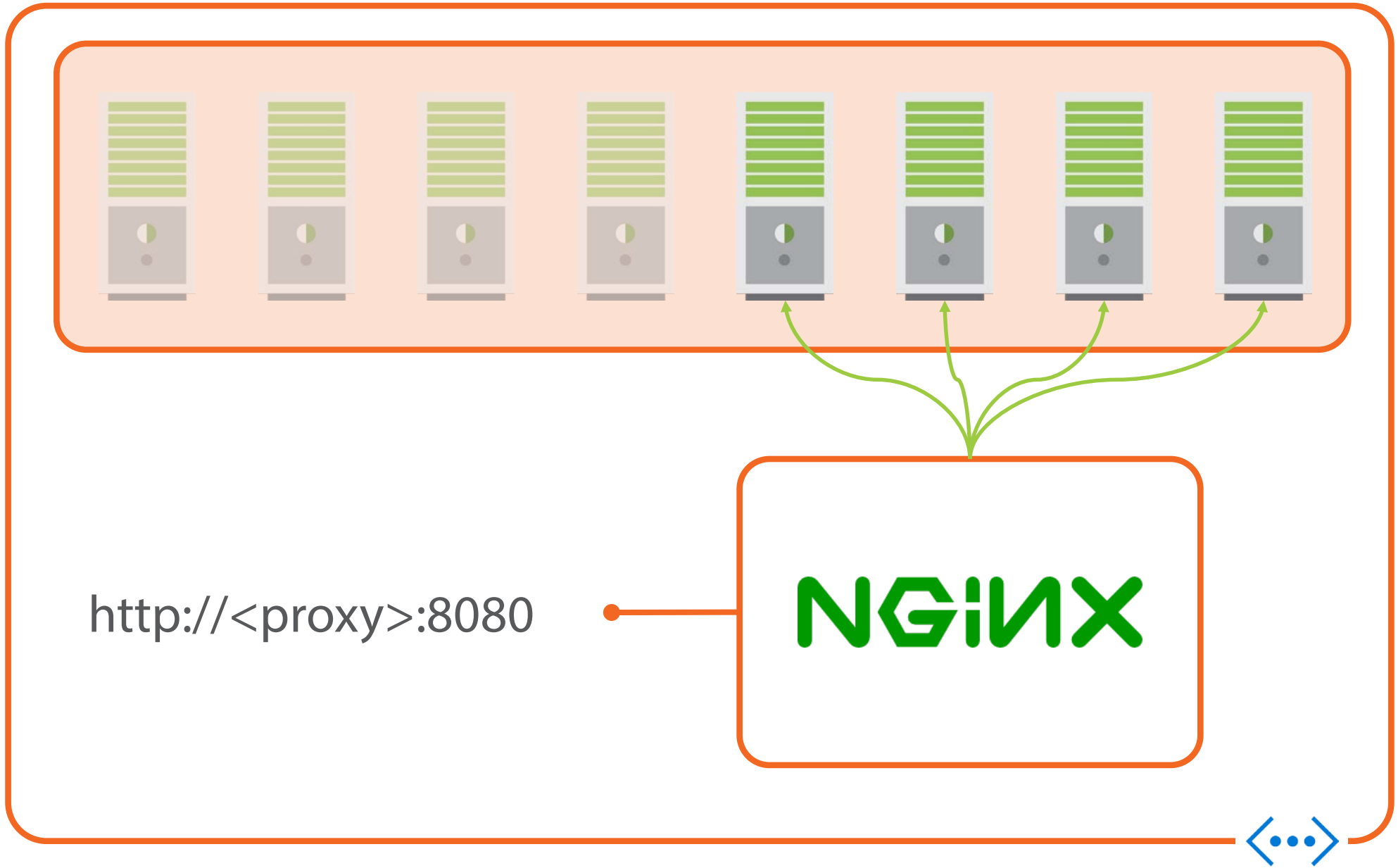



http://<proxy>:8080
NGINX

Nginx Reverse Proxy Config

```
upstream stargate {
    server workernode0.my-hbase.f7.internal.cloudapp.net:8090;
    ...
    server workernode3.my-hbase.f7.internal.cloudapp.net:8090;
}
server {
    listen 8080 default_server;
    server_name my-hbase-proxy;
    location / {
        proxy_pass http://stargate;
    }
}
```



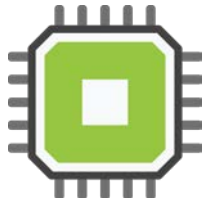


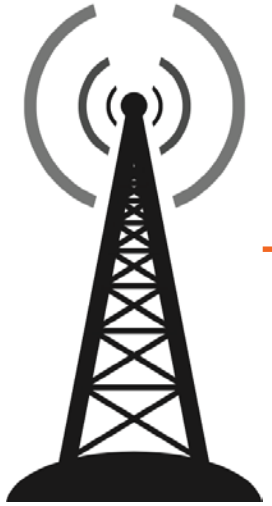
Demo: Stargate Proxy

Nginx VM

Reverse Proxy

Load Balancing

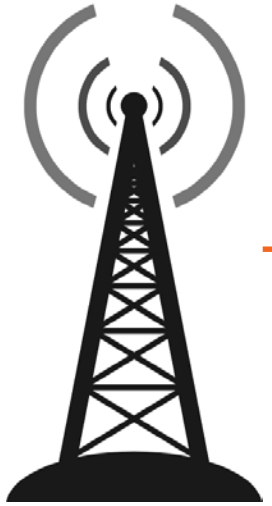
	x4	x2	x3	x1
	4 cores	4 cores		
	14 GB	7 GB		
	200 GB **SSD**	285 GB		



Event Processing

APACHE HBASE

? events/racer | **?** events/second | **~800** requests/second



Event Processing

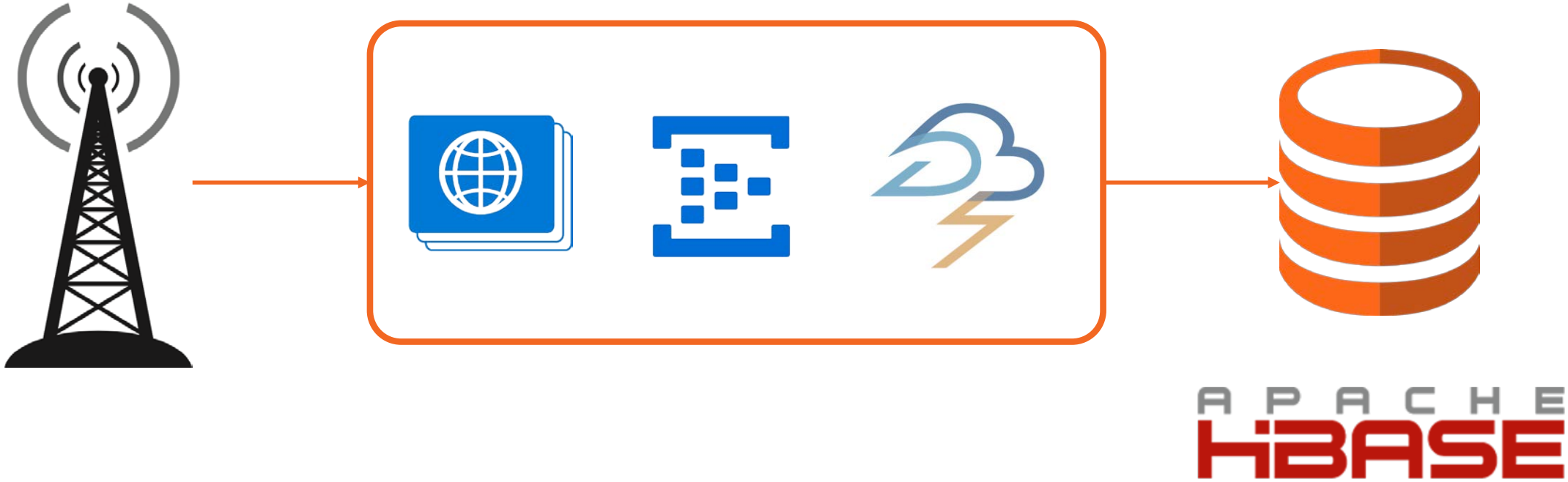
APACHE HBASE

100 events/racer

3K events/second

~800 requests/second

HBase **bottleneck**

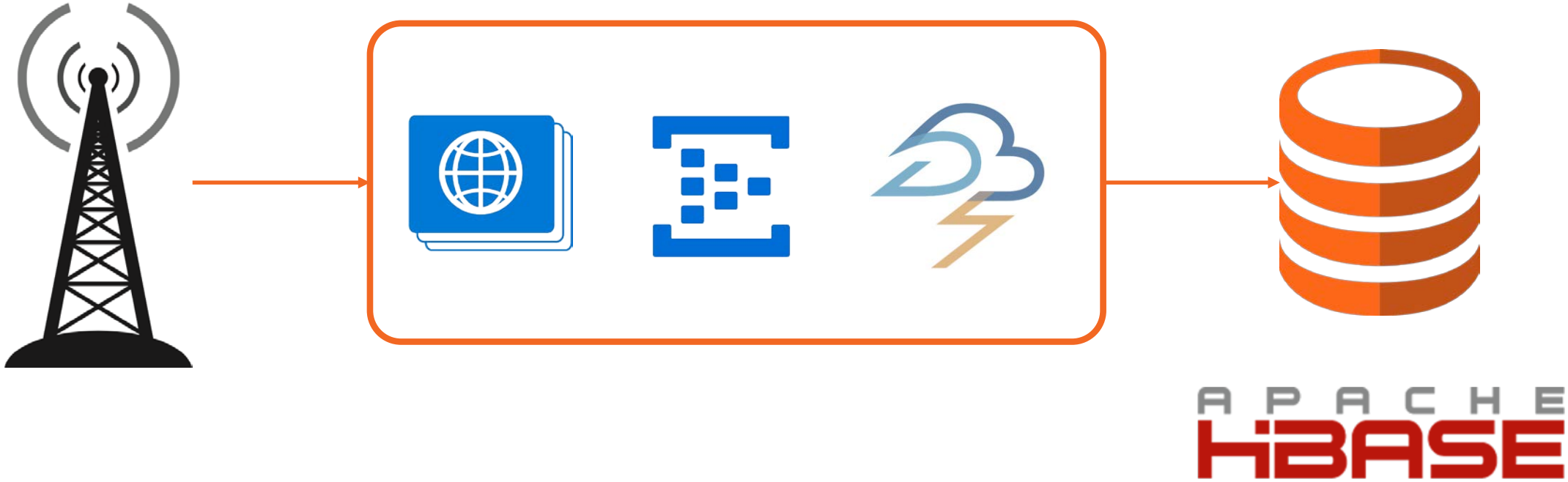


100 events/racer | **3K** events/second | **~800** requests/second

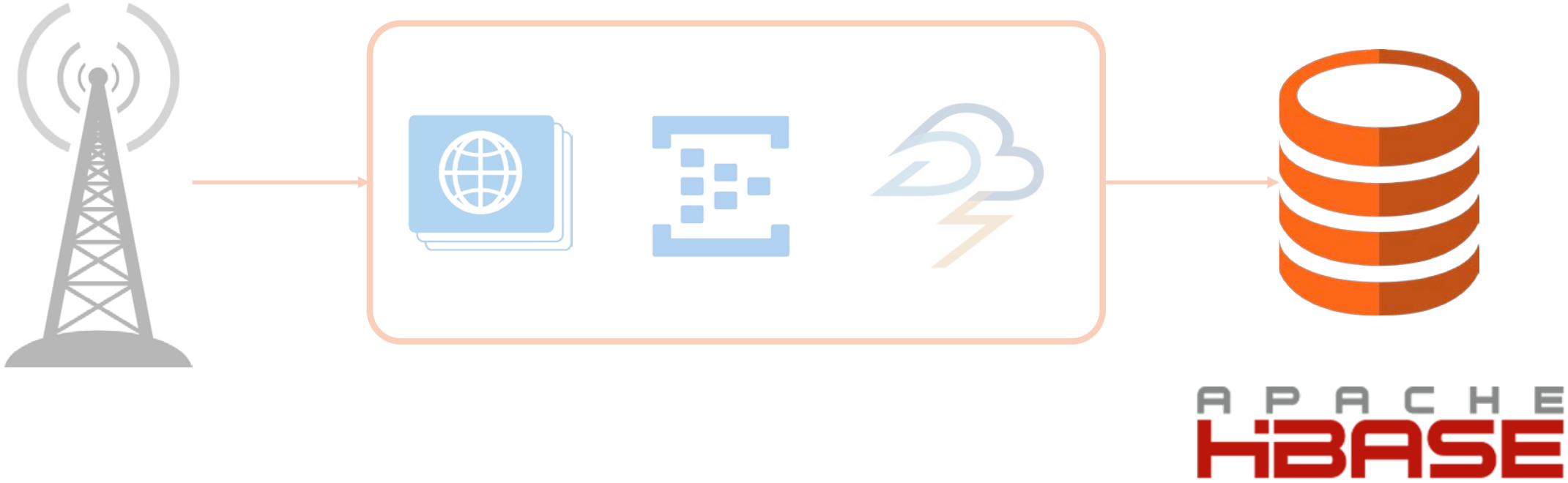
Self-regulating

Event Hub queue

Storm processing



15K events | Pull **~750** events/second | **~800** requests/second



15K events

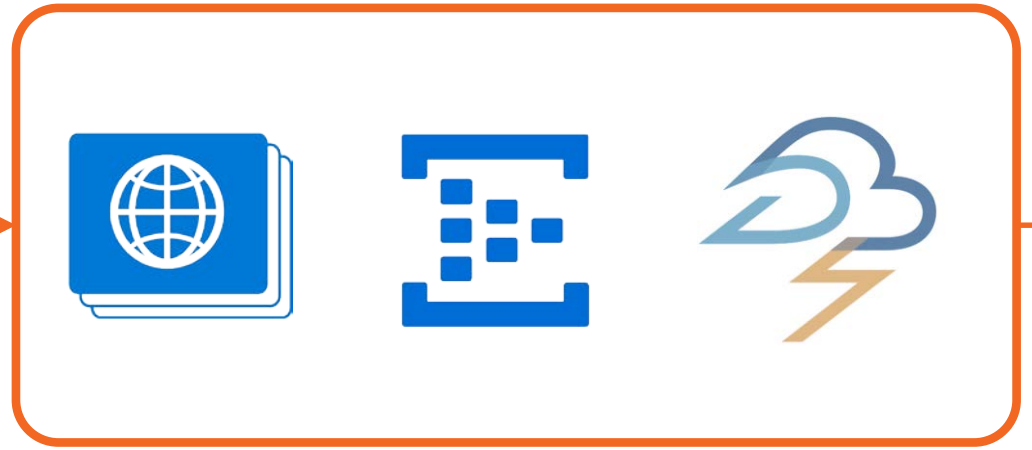
Pull **~750** events/second

~800 requests/second

```
put 'timing-events' ...

get/put 'sector-times' ...
```

~15.5K requests

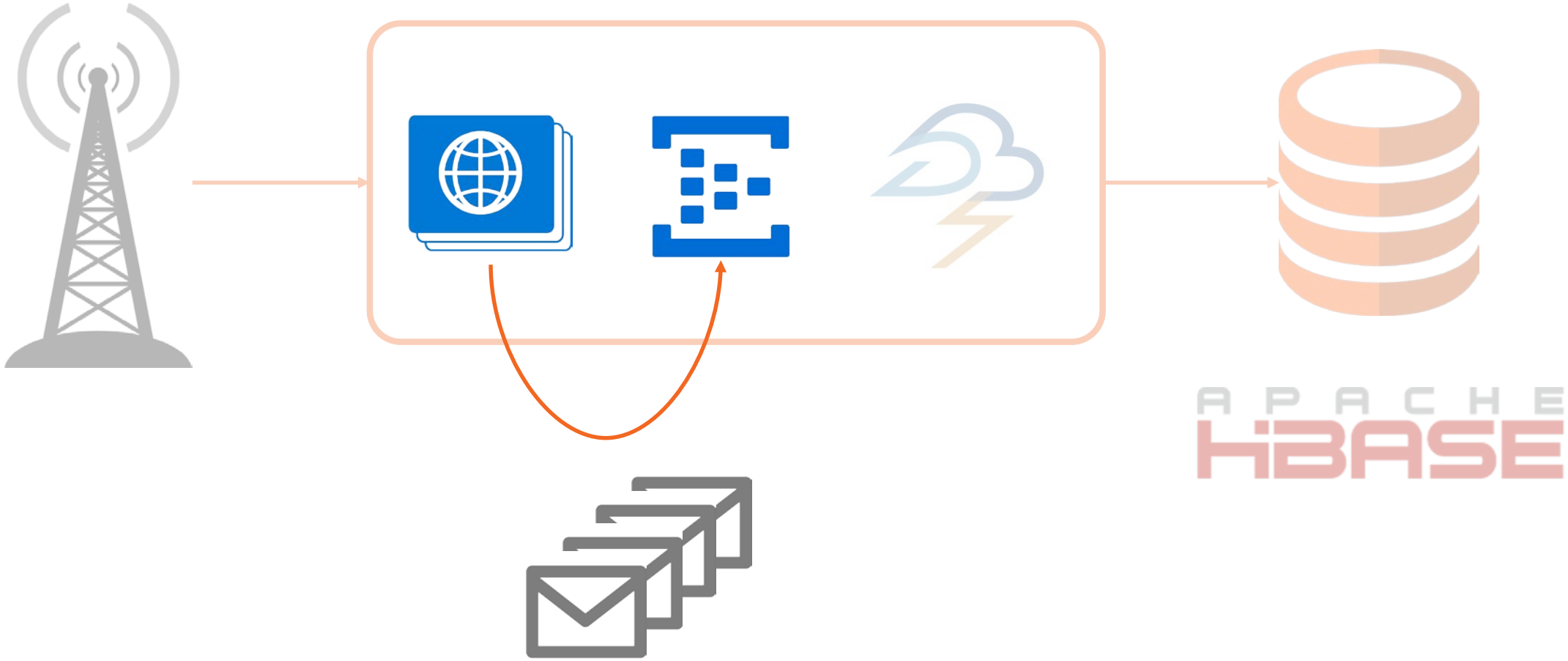


APACHE HBASE

15K events | Pull **~750** events/second | **~800** requests/second

T+5 seconds → **T+20** seconds

>20 seconds



Real World Big Data in Azure

Elton Stoneman

@EltonStoneman | blog.sixeyed.com

100K events

300 seconds

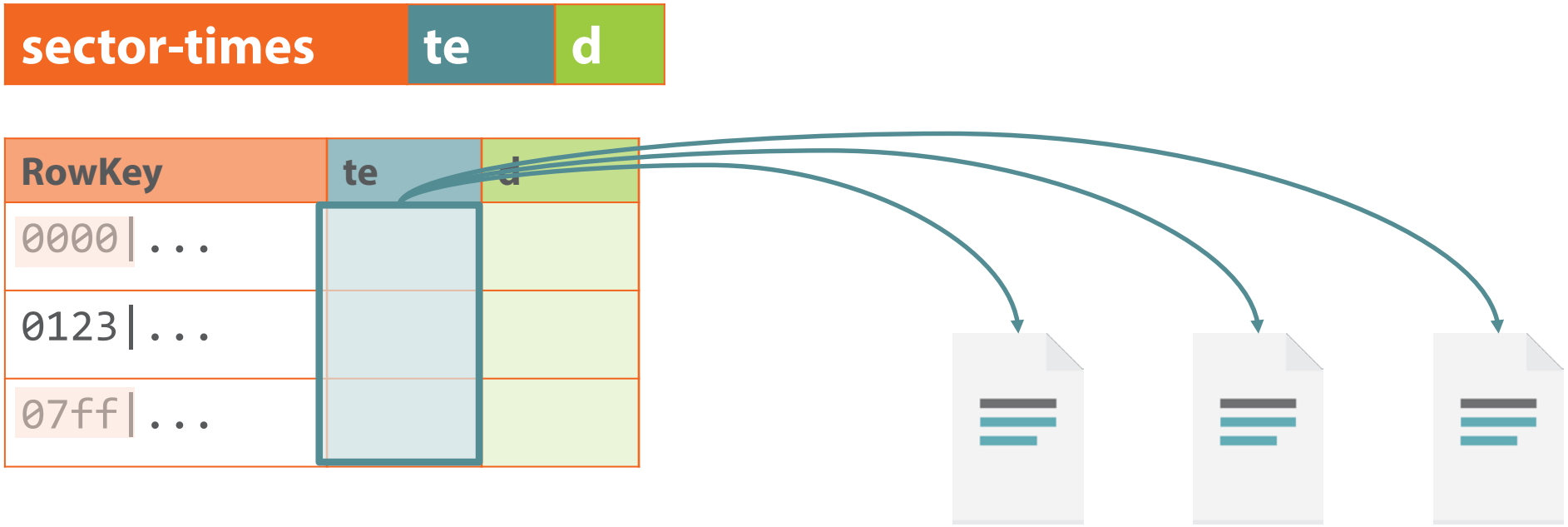
~333 events/second

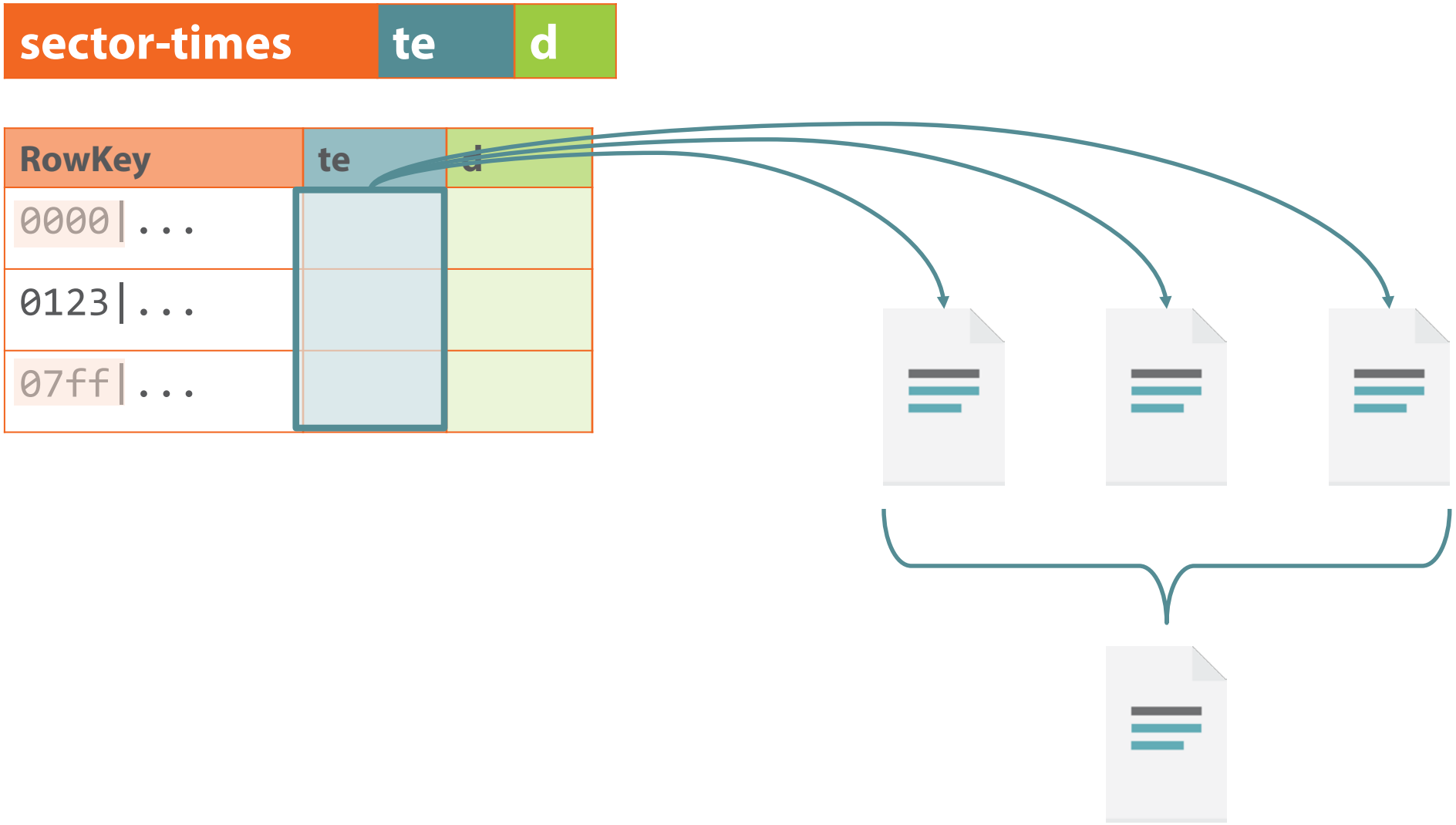
Demo: Scaling HBase

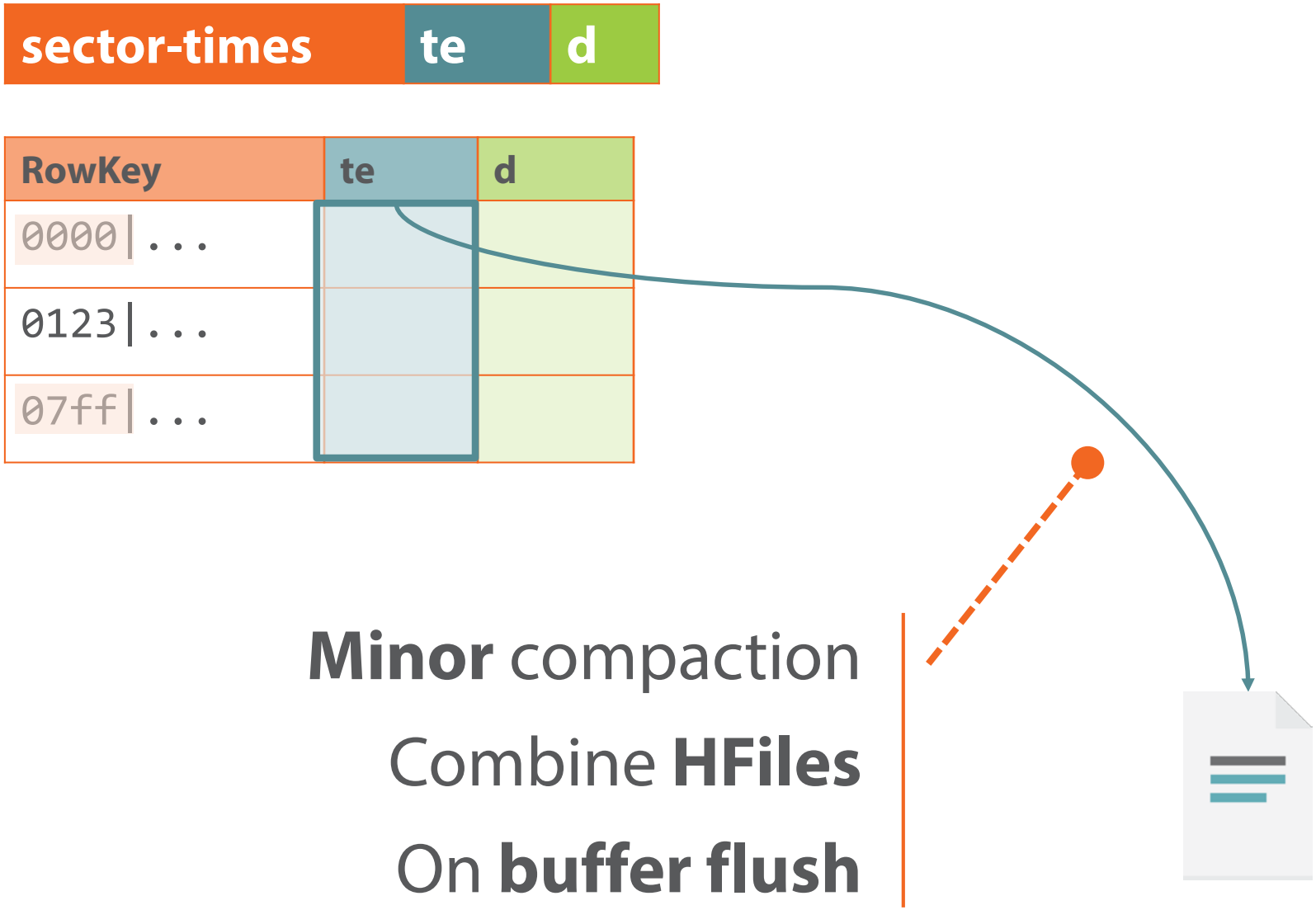
Scale Up Cluster

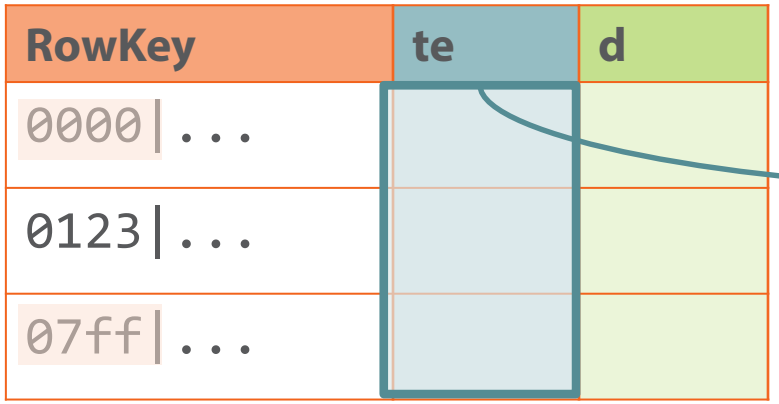
Run Major Compaction

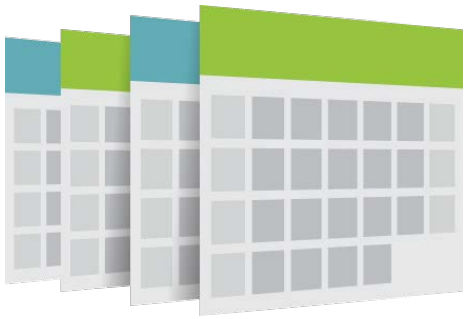
Performance Test







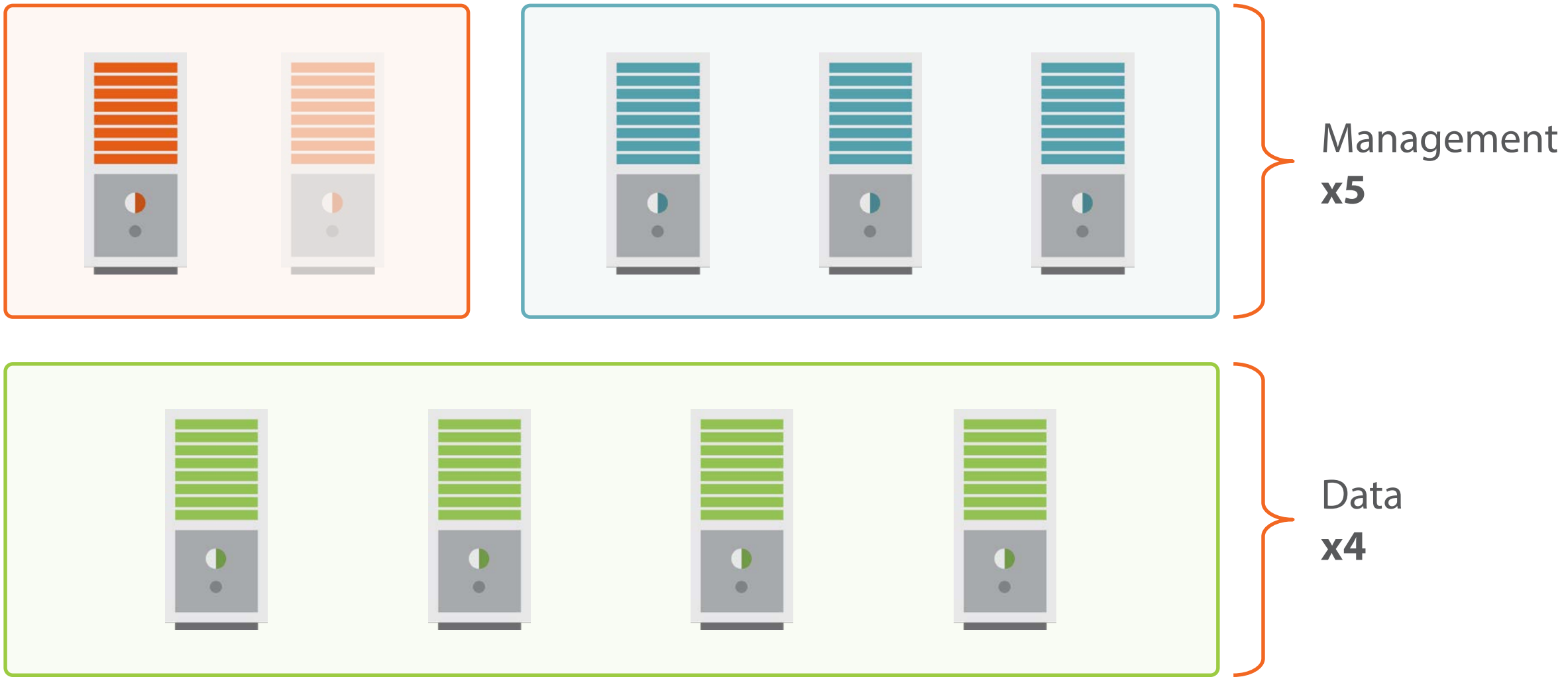




Major compaction

Combine **HFiles**

Split large regions

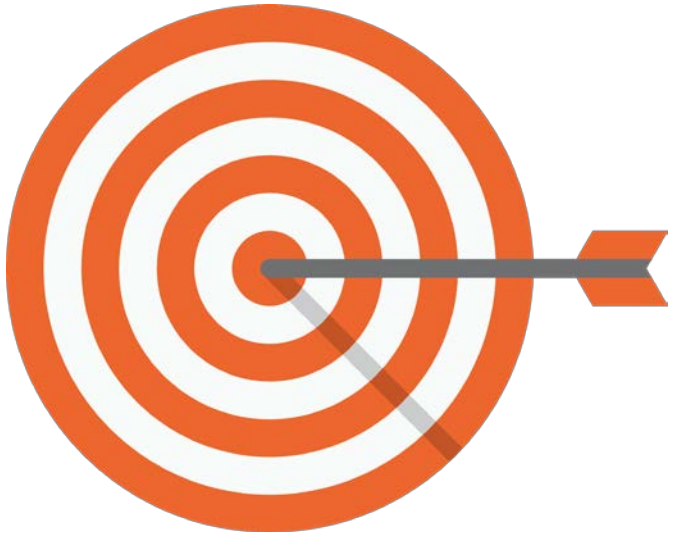


Region Servers = **44%** of compute

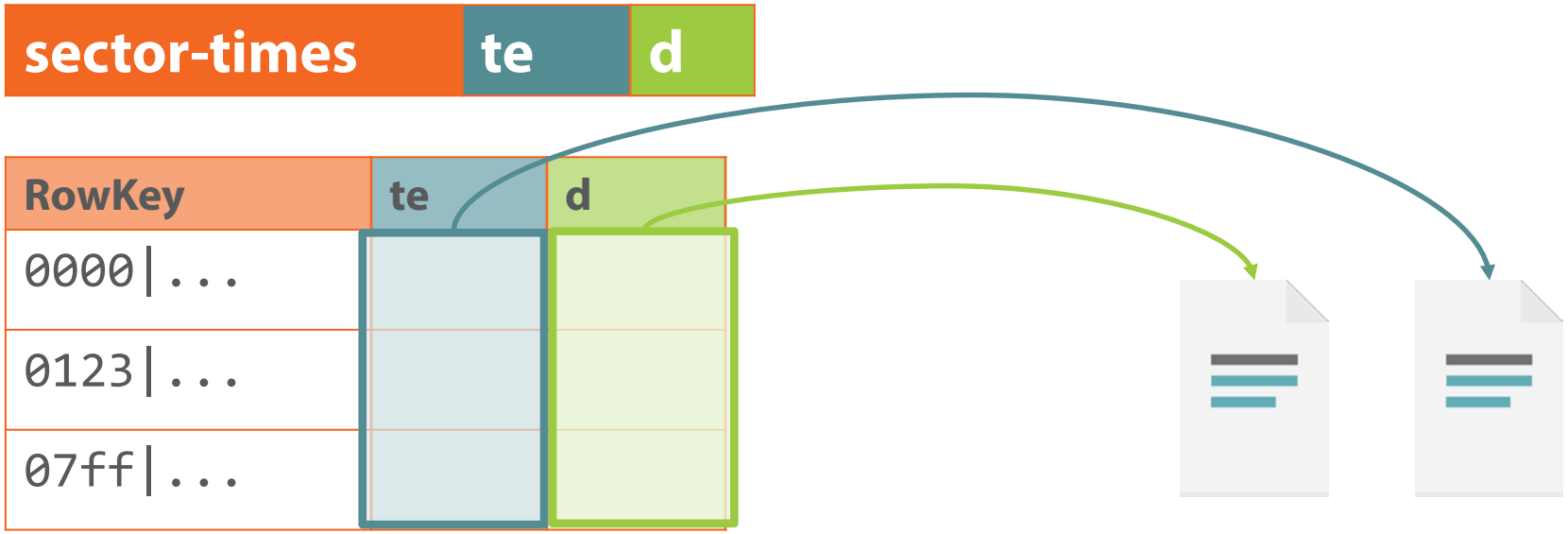
Region Servers = **62%** of compute

Region Servers = **76%** of compute

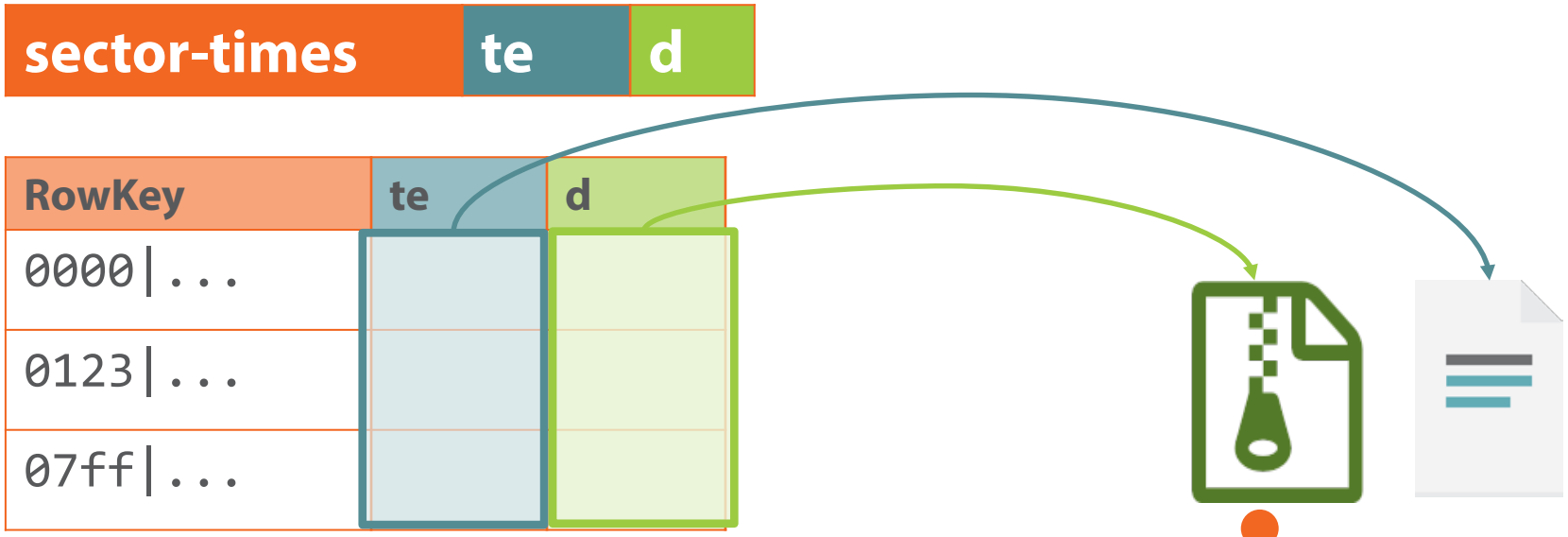
Module Goals



- Cluster Setup & Region Servers
- Regions & HFiles
- Load Balancing & Scaling
- .NET Best Practices
- HBase Administration



HFile **compression**



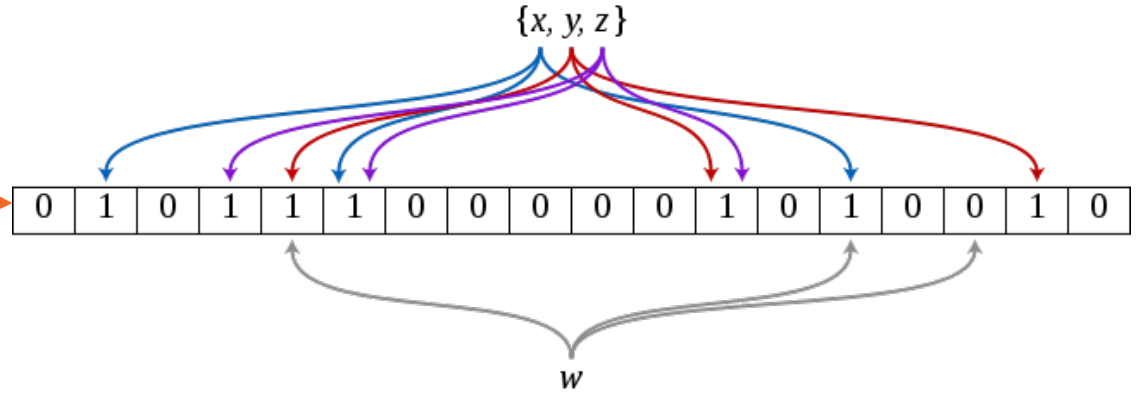
Column Family compressed

GZip, LZO or Snappy

Less disk **I/O**

sector-times	te	d

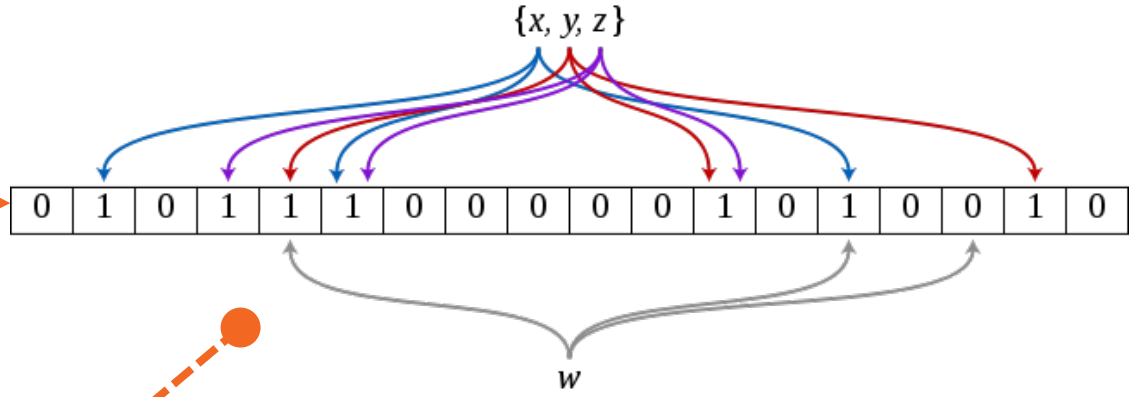
RowKey	te	d
0000|...		
0123|...		
07ff|...		



Bloom filters

sector-times	te	d

RowKey	te	d
0000|...		
0123|...		
07ff|...		

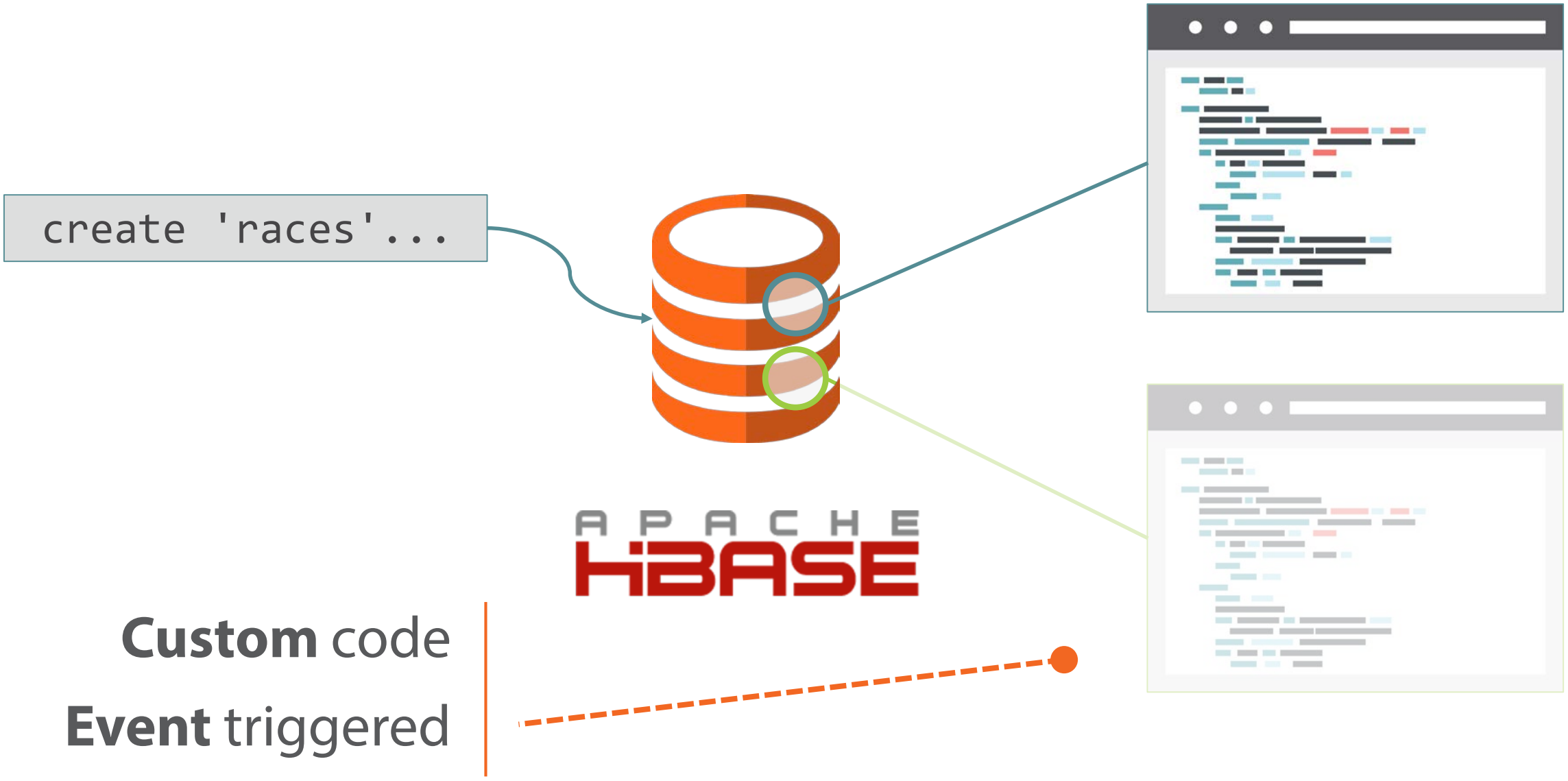


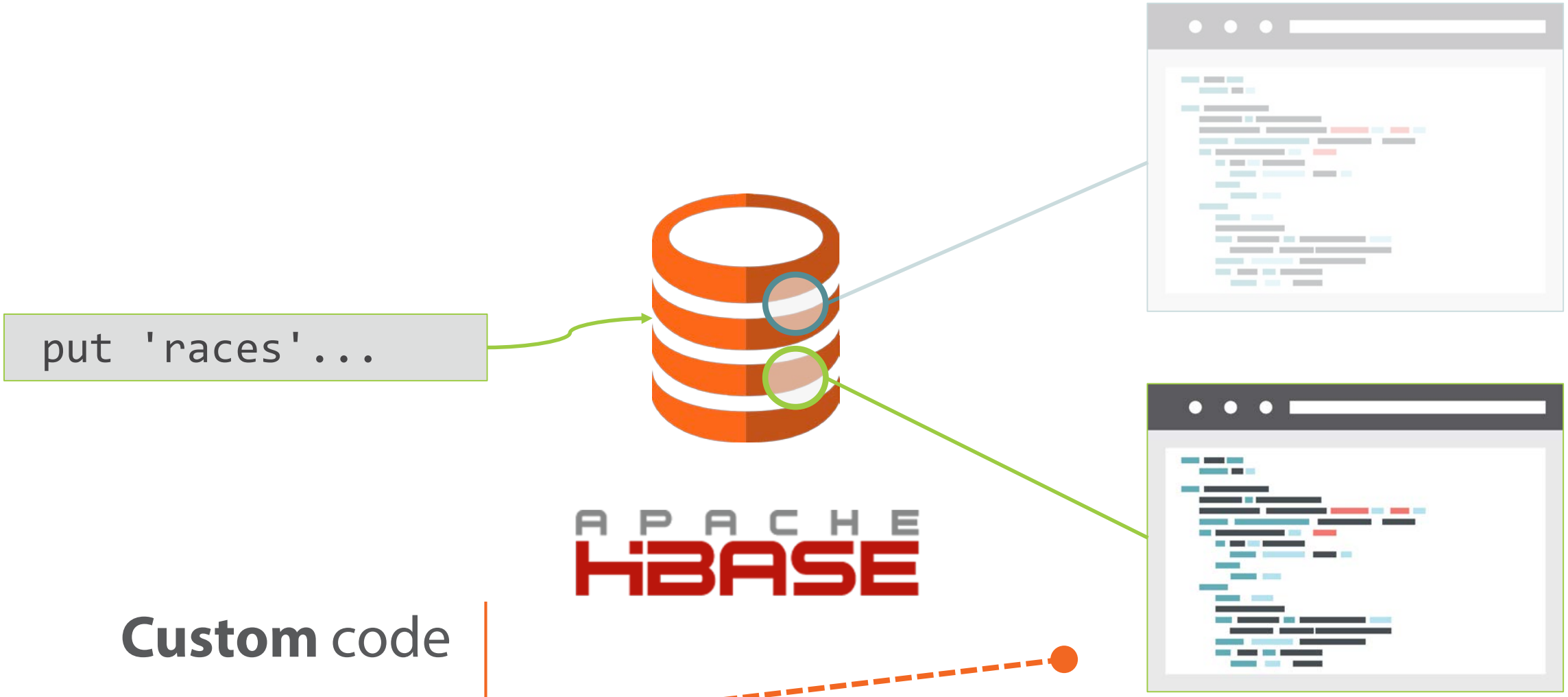
Row or **column** level

Additional indexing

Less disk **I/O**

Coprocessors

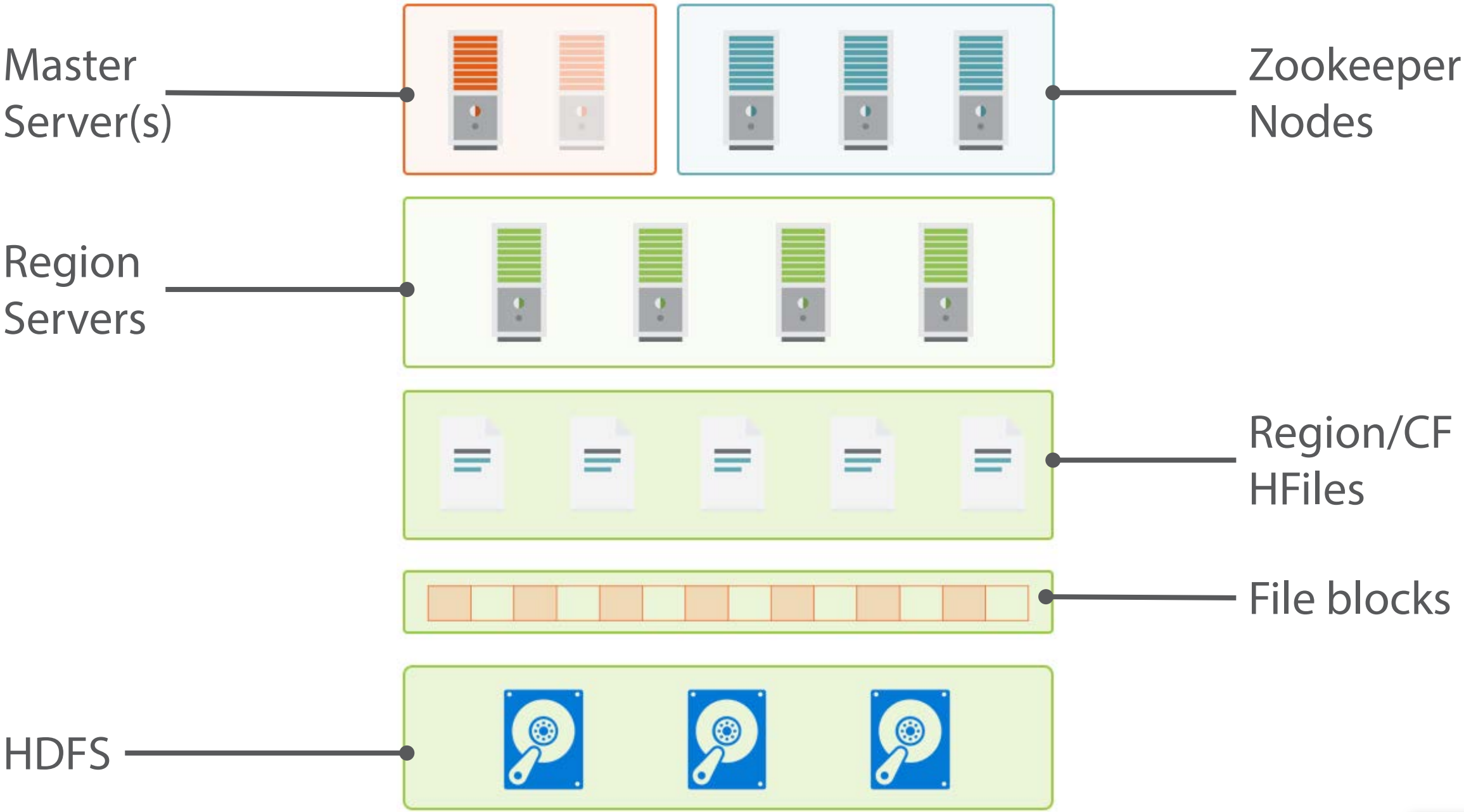

create 'races'...
APACHE
HBASE
Custom code
Event triggered

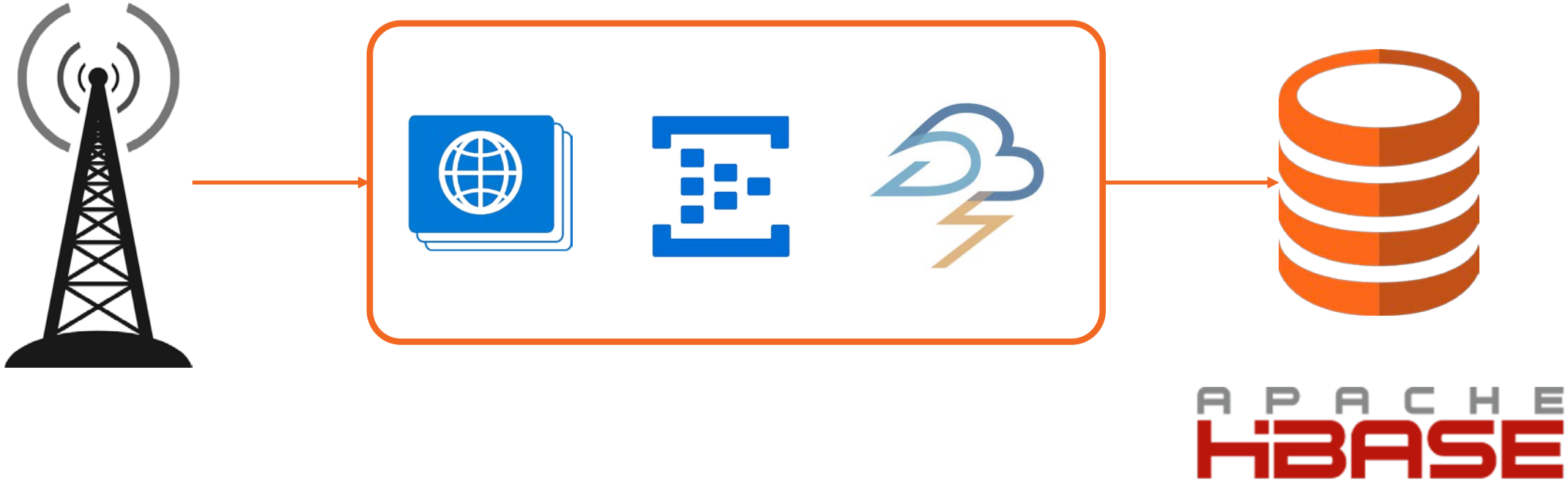


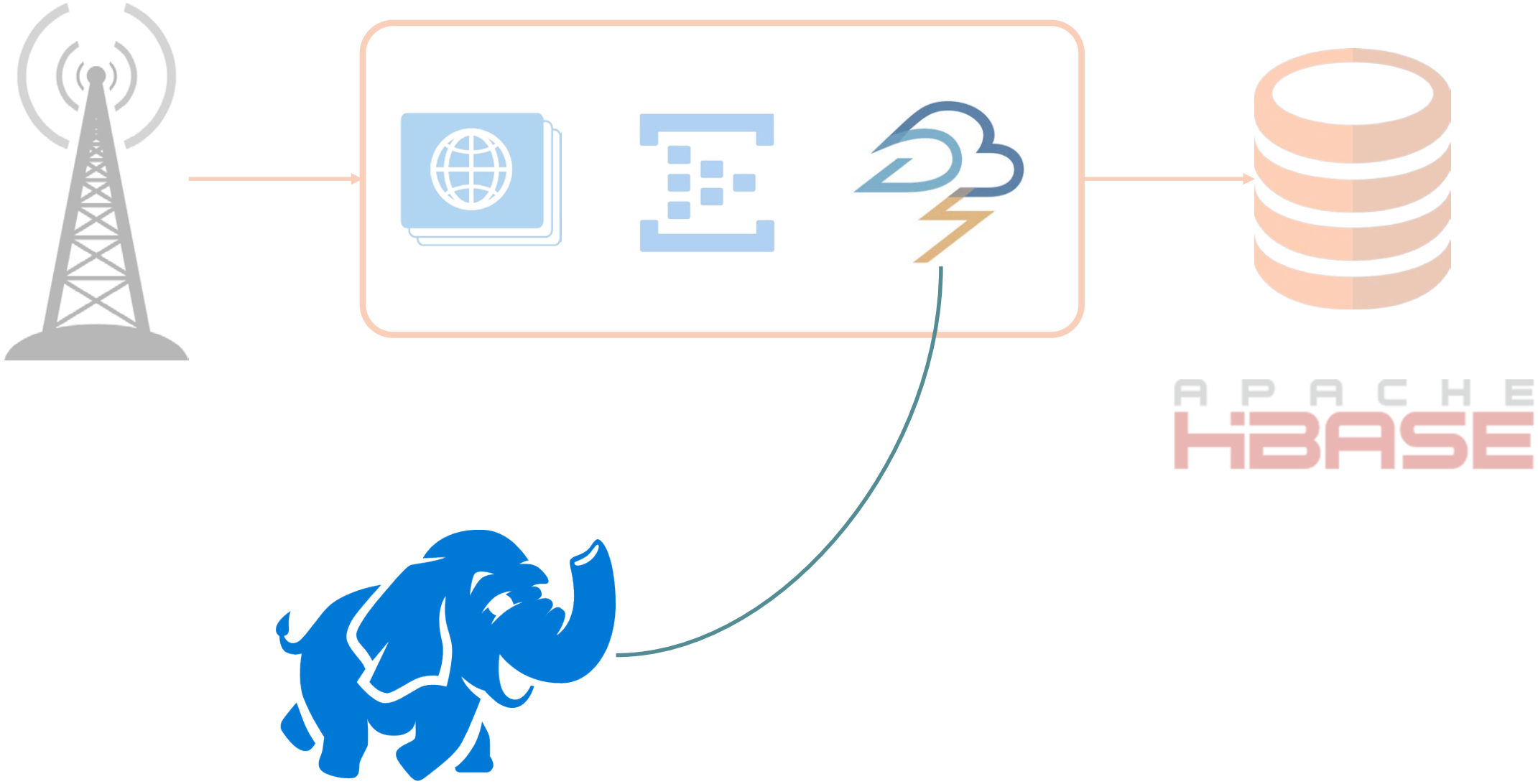
Custom code

Event triggered

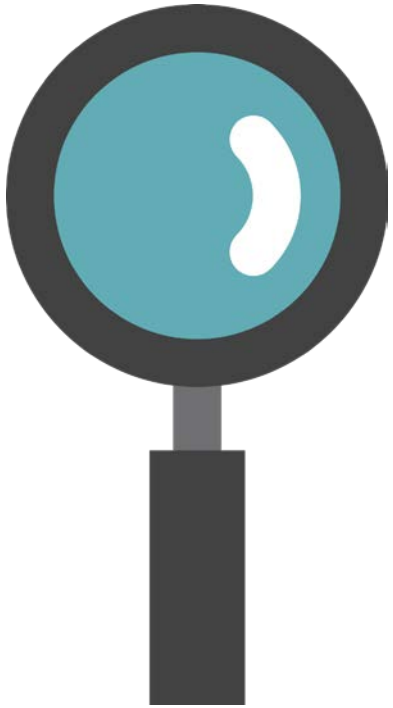
Runs on **event node**


Master
Server(s)
Zookeeper
Nodes
Region
Servers
Region/CF
HFiles
File blocks
HDFS





Coming Next



Event processing

Real-time

Storm on HDInsight

Azure Event Hubs