Afi Escuela

Ecosistema Hadoop

Máster en Data Science y Big Data

Jorge López-Malla Matute
jlmalla@geoblink.com

Febrero 2022

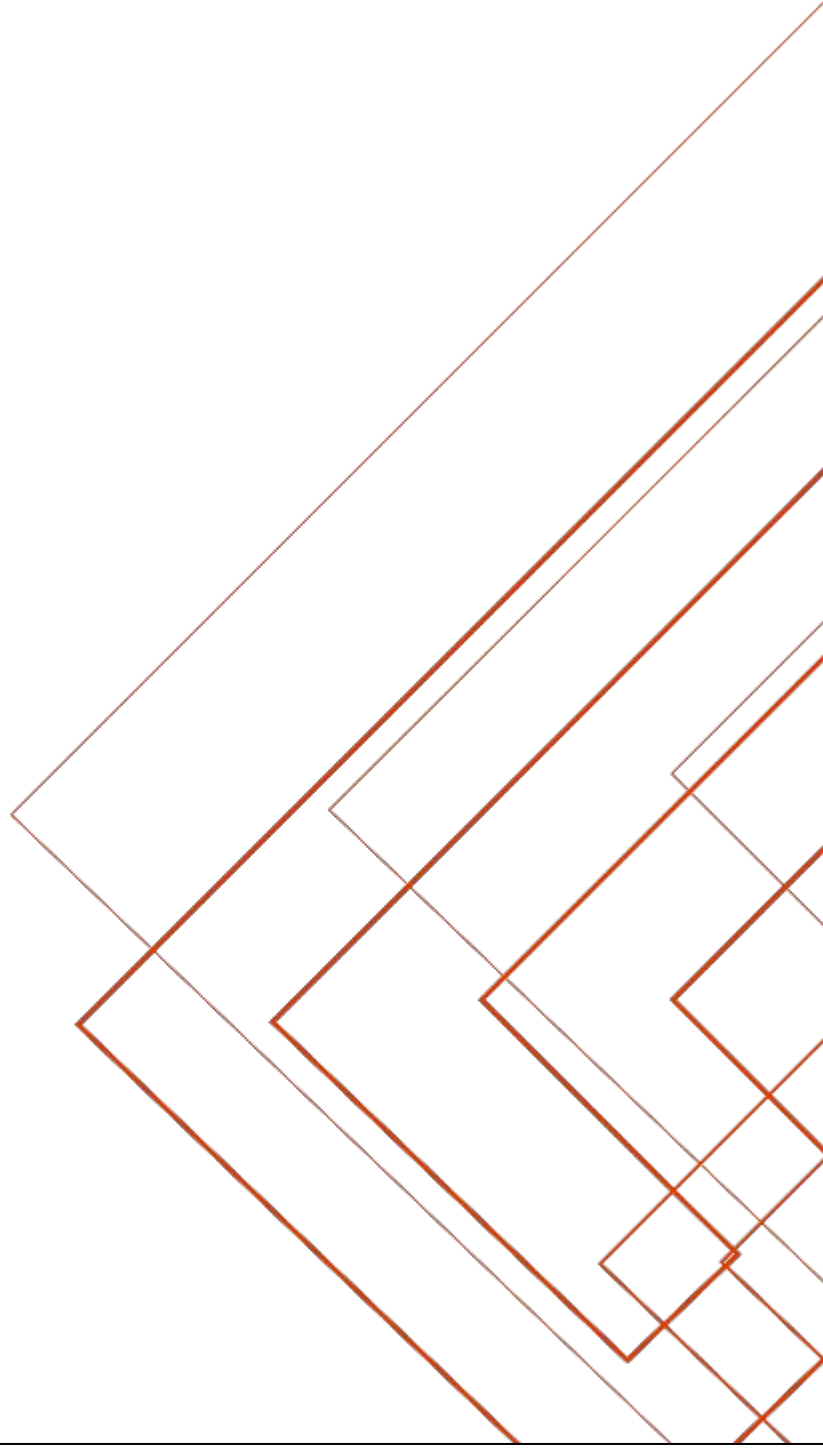
Presentación

- Jorge López-Malla Matute
- Puesto actual:
  - Senior Data Engineer en Geoblink
- Experiencia docente:
  - Profesor en diversos Masters de Big Data durante los últimos 6 años
  - Profesor de Tecnologías Masivas en ICAI
- Años trabajados en Big Data: 9 años

Índice

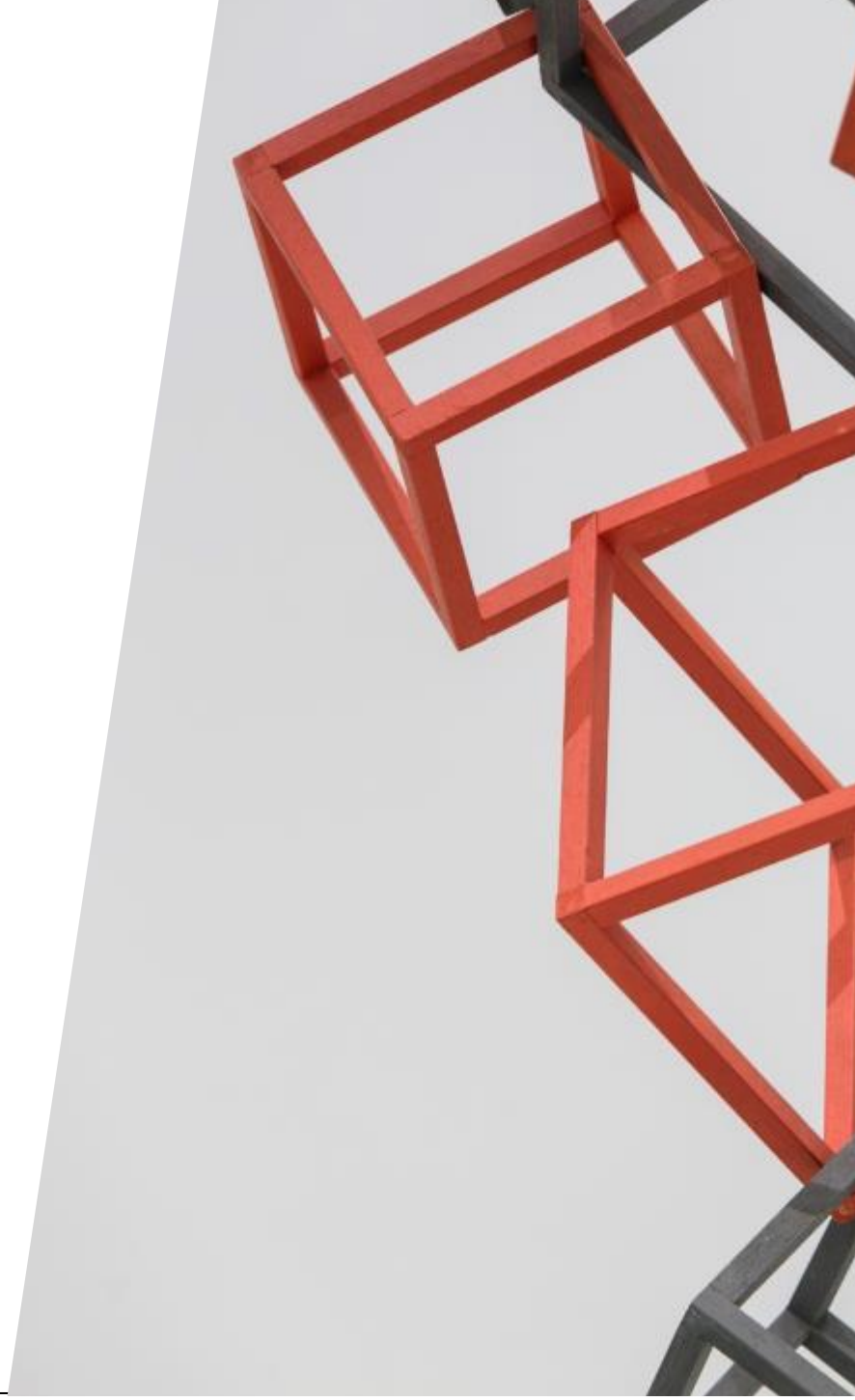
Contenido

1. Componentes
    - Apache Hive
    - Apache HBase



Componentes

Apache Hive



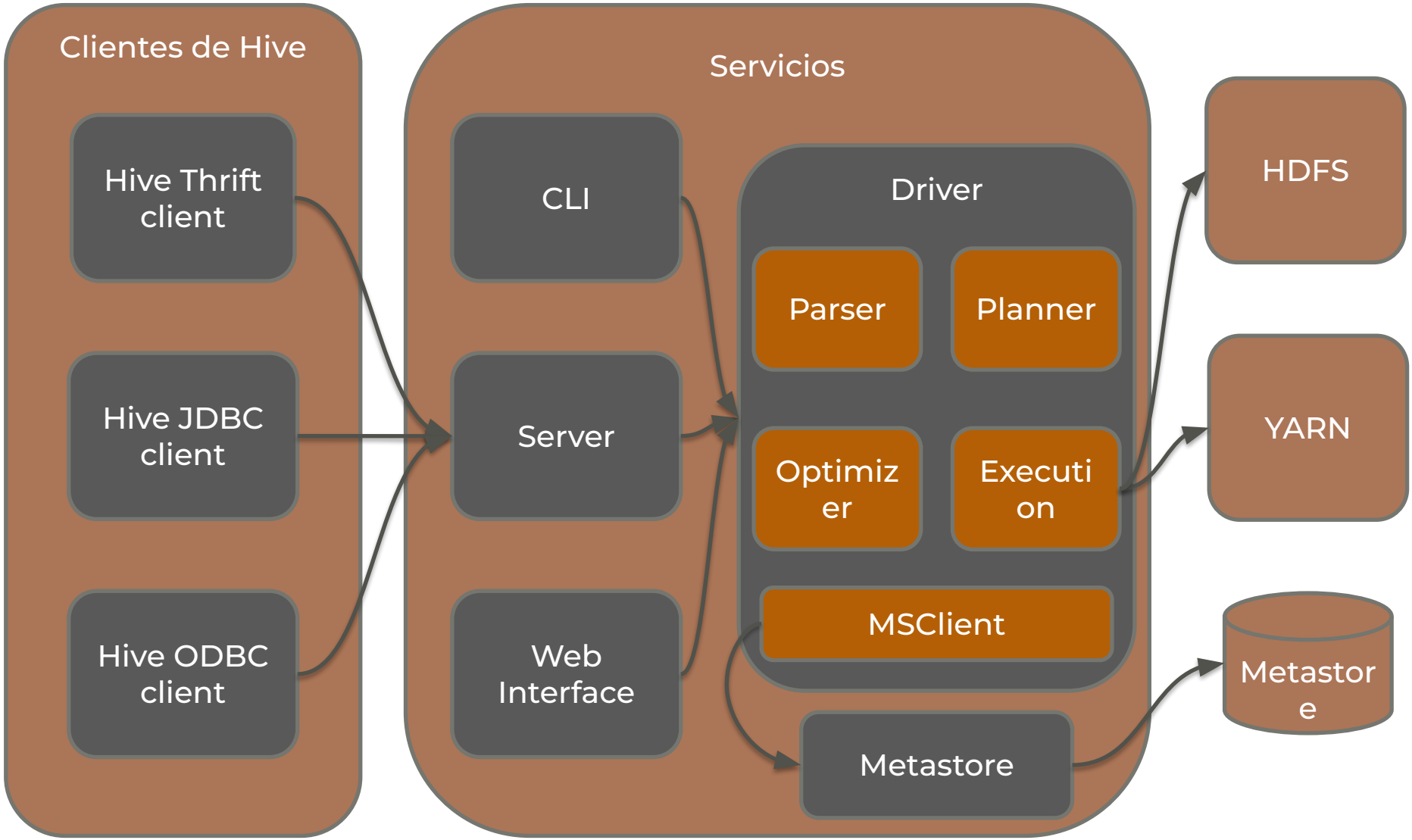
Hive- Introducción

- En el inicio **Map & Reduce** sólo estaba enfocado al desarrollo
- Con el tiempo más casos de uso se fueron desarrollando y se vio la necesidad de lenguajes de scripting
- El primer proyecto de Hadoop en scripting fue **Apache Pig** (2008)
- Los usuarios tenían que seguir aprendiendo otro lenguaje
- No podía conectarse con otros servicios externos con facilidad como lo hubiera hecho el SQL

Hive- Introducción

- En 2010 Facebook libera un proyecto que convierte sentencias SQL en trabajos de Map&Reduce
- Se libera Apache Hive y se extiende su uso
- Al ser un intérprete de SQL permite que herramientas que ya usaban SQL pudieran procesar cantidades masivas de datos
- Hive además permite la compartición de tablas de datos entre distintos procesos de una manera sencilla
- Los proyectos posteriores de procesamiento distribuido toman Hive como referencia

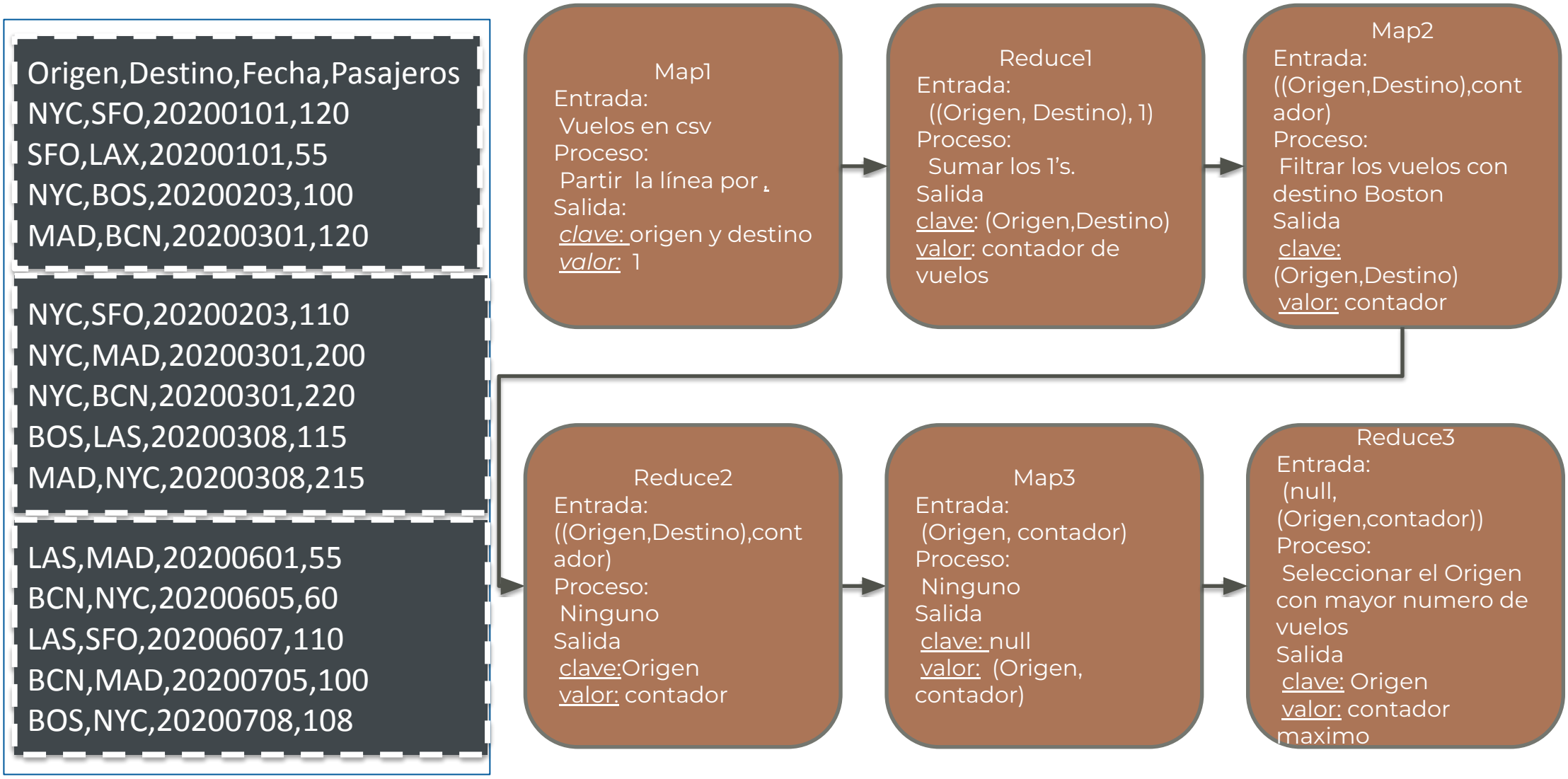
Hive- Arquitectura



Hive- Funcionamiento

- **Hive** interpreta sentencias SQL en un motor de procesamiento distribuido
- **Hive** necesita de una base de datos de metadatos
- En ella se guardan los metadatos de las tablas
- Las tablas tienen que almacenarse en un almacenamiento compatible con el motor de procesamiento
- **Hive** interpreta la secuencia y la convierte en trabajo distribuido más eficiente

Hive- Funcionamiento

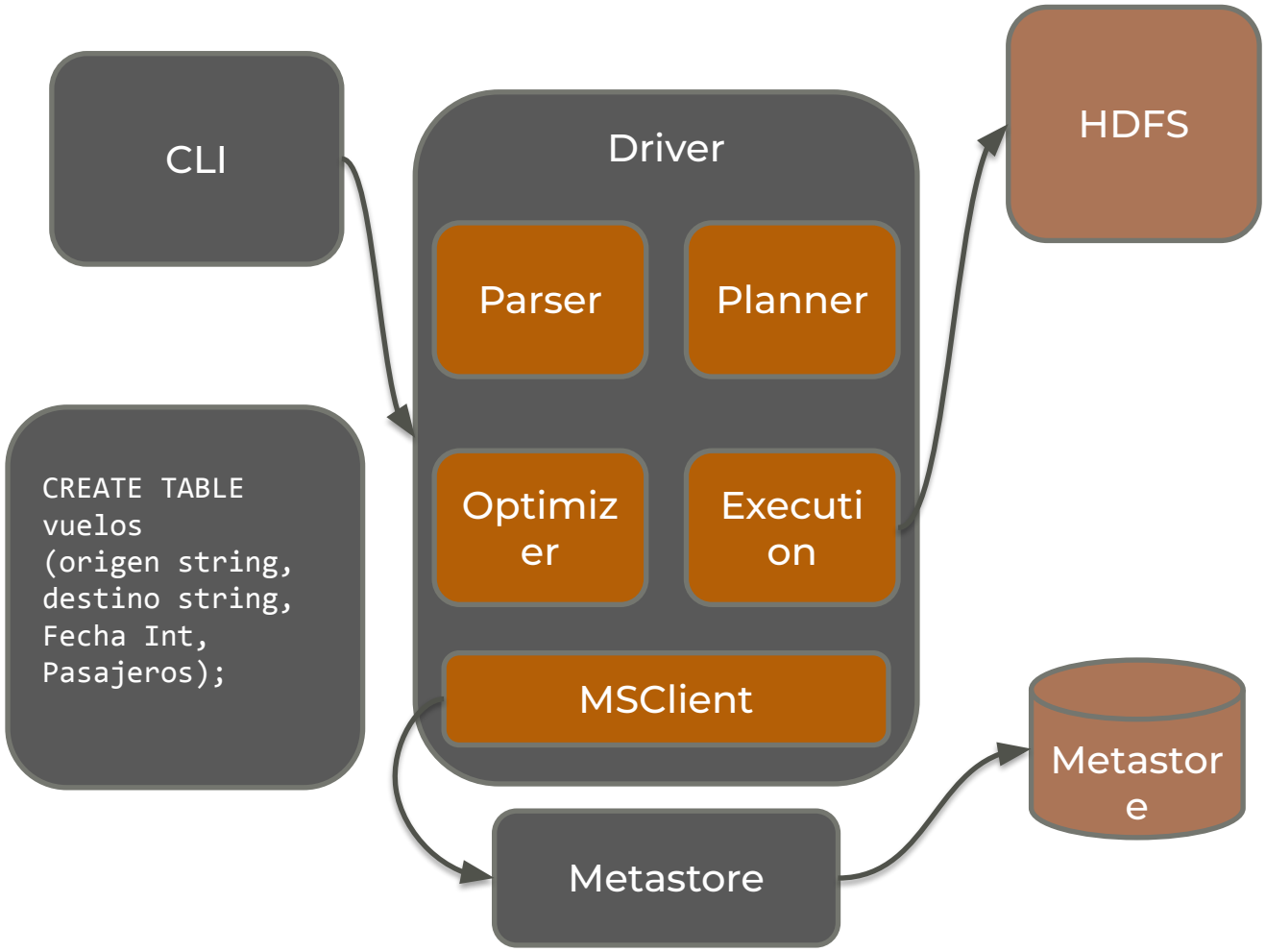


Hive- Funcionamiento

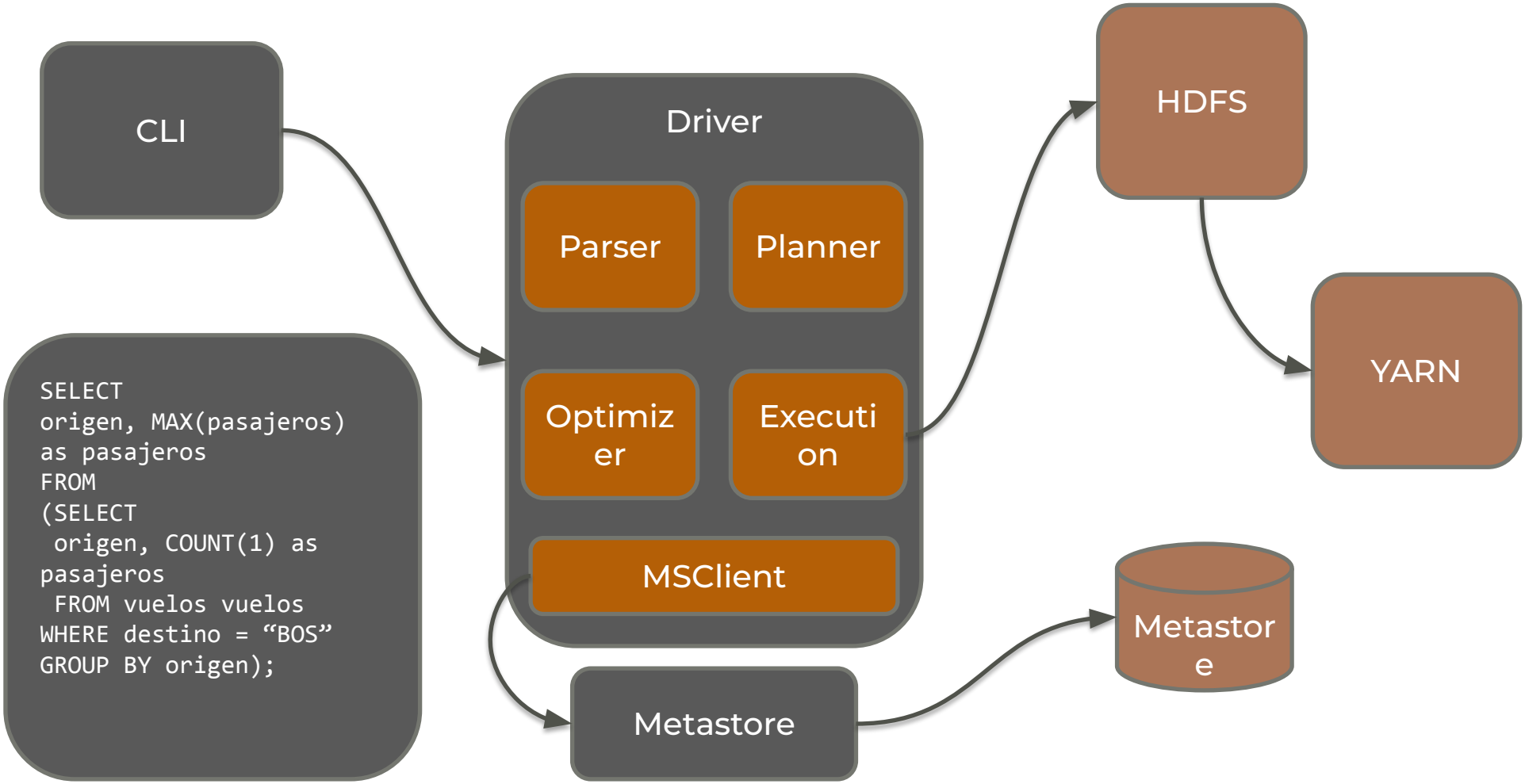
```
Origen,Destino,Fecha,Pasajeros
NYC,SFO,20200101,120
SFO,LAX,20200101,55
NYC,BOS,20200203,100
MAD,BCN,20200301,120
```

```
NYC,SFO,20200203,110
NYC,MAD,20200301,200
NYC,BCN,20200301,220
BOS,LAS,20200308,115
MAD,NYC,20200308,215
```

```
LAS,MAD,20200601,55
BCN,NYC,20200605,60
LAS,SFO,20200607,110
BCN,MAD,20200705,100
BOS,NYC,20200708,108
```



Hive- Funcionamiento



Hive- Funcionamiento

```
SELECT
origen, MAX(pasajeros) as
pasajeros
FROM
(SELECT
 origen, COUNT(1) as pasajeros
 FROM vuelos vuelos
WHERE destino = “BOS”
GROUP BY origen);
```

Map1
Entrada:
tabla vuelos en csv
Proceso:
filtrar los vuelos con destino BOS
Salida:
clave: origen
valor: 1

Reduce1
Entrada:
(Origen, [1’s])
Proceso:
Sumar los 1’s
Salida
clave: Origen
valor: contador

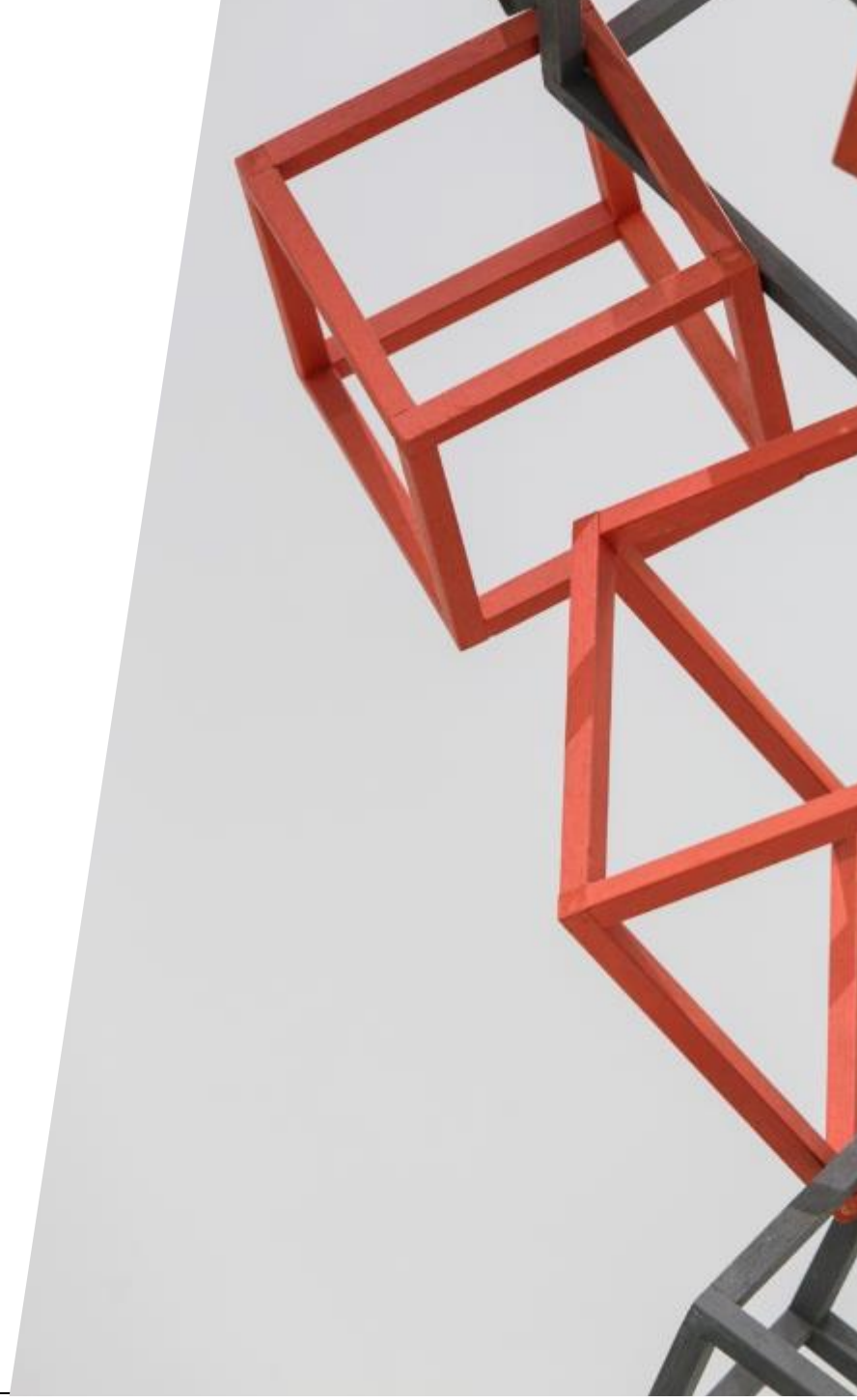
Map2
Entrada:
(Origen, contador)
Proceso:
Ninguno
Salida
clave: Null
valor: (Origen, contador)

Reduce2
Entrada:
(Null,[(Origen, contador)])
Proceso:
Obtener el máximo
Salida
clave:Origen
valor:

Hive- Introducción

- La solución de **Hive** da la solución óptima mirando el planificador
- El planificador de queries no entiende de datos ni de escalabilidad
- Puede hacer algún tipo de optimización usando el almacenamiento
- ¿Y si los vuelos con destino a Boston suman tanto como para llenar la partición más grande de nuestro sistema?
- Se podría solucionar con tablas intermedias

Apache Hbase



HBase - Introducción

- Map & Reduce procesa grandes cantidades de datos en un tiempo aceptable
- Hive nos proporciona un lenguaje SQL
- Si juntamos las dos tecnologías, ¿tendríamos una Base de Datos relacional con todas ventajas del Big Data?
- La respuesta es NO
- Juntando una tecnología de cómputo que admita cantidades masivas de datos y un proyecto que nos permita hacer SQL **NO** nos permite hacer queries en tiempo *online* "sólo" facilita el procesamiento

HBase - Introducción

- A raíz de las nuevas formas de almacenar se empiezan a pensar nuevas formas de consultar esa información
- El principal problema que pretende resolver es el acceso aleatorio a unos datos concretos
- Para ello usa HDFS como sistema de ficheros subyacente y una estrategia **Columnar**
- Con ello se consigue un acceso “rápido” a datos aleatorios en colecciones masivas de información
- Posteriormente nacen tecnologías de almacenamiento basadas en su misma filosofía supliendo algunas de sus carencias (Apache Cassandra)

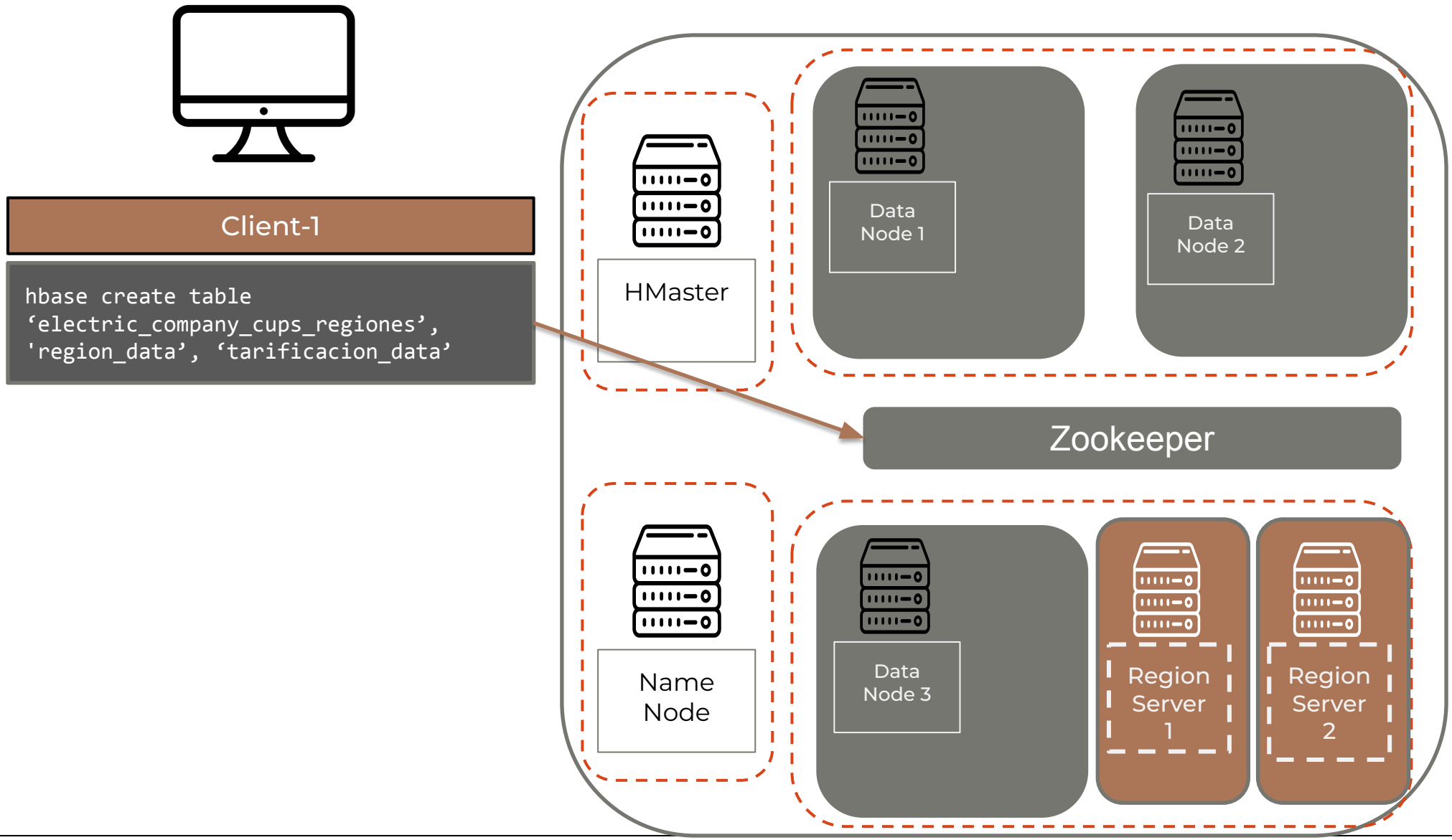
HBase - Introducción

- A raíz de las nuevas formas de almacenar se empiezan a pensar nuevas formas de consultar esa información
- El principal problema que pretende resolver es el acceso aleatorio a unos datos concretos
- Para ello usa HDFS como sistema de ficheros subyacente y una estrategia **Columnar**
- Con ello se consigue un acceso “rápido” a datos aleatorios en colecciones masivas de información
- Posteriormente nacen tecnologías de almacenamiento basadas en su misma filosofía supliendo algunas de sus carencias (Apache Cassandra)

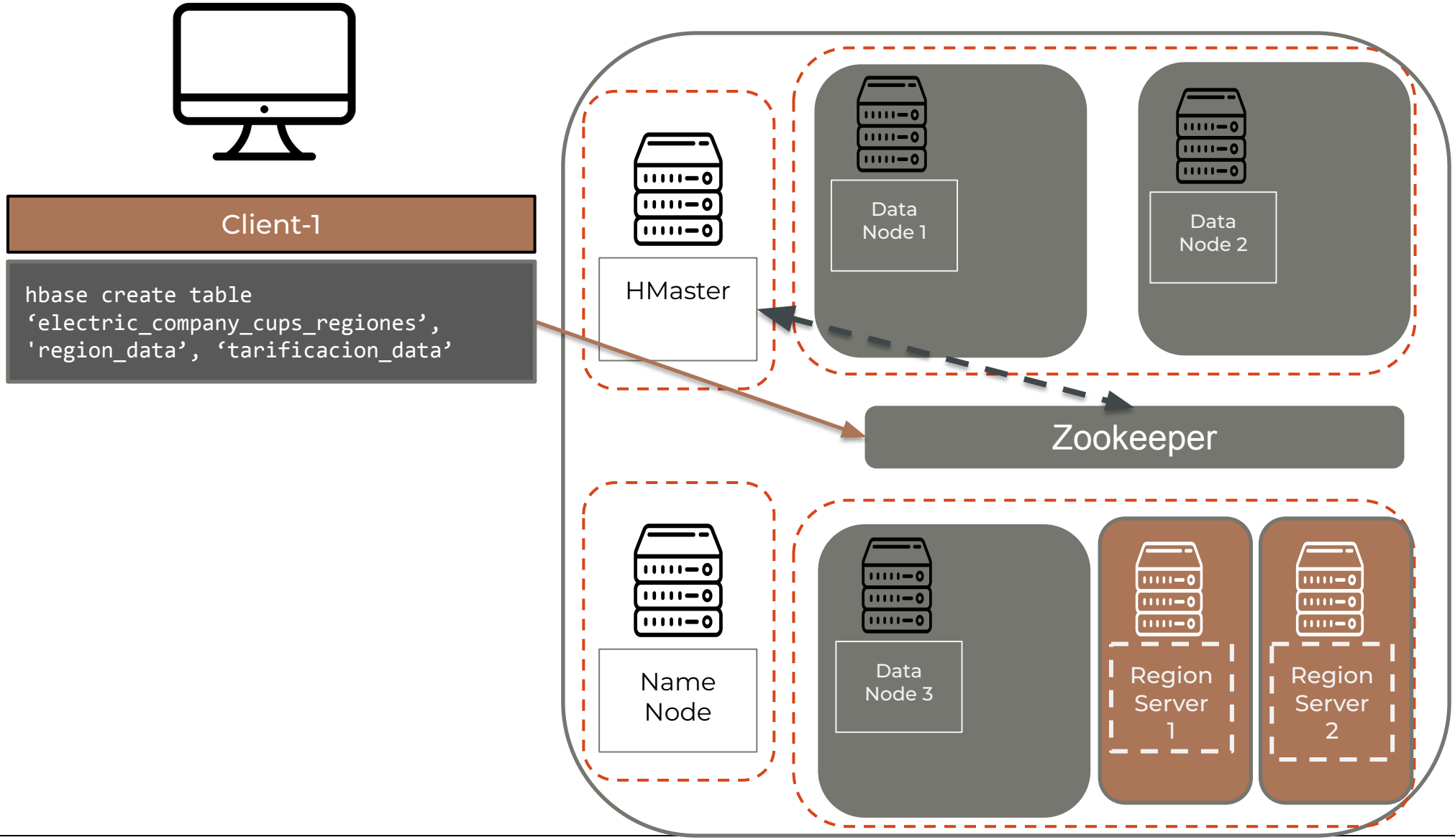
HBase - Terminología

- Hbase tiene la siguiente terminología:
  - **Table**: Conjunto de **rows**
  - **Row:** Conjunto de datos agrupados en columnas. Se compone de una **rowKey** y varias **column families**
  - **RowKey:** Column de una **row** que hace única a la misma**.**
  - **ColumnFamily:** Serie de Columns de una row. Su estrucutra no se tiene que compartir entre distintas **rows** de una misma **table**

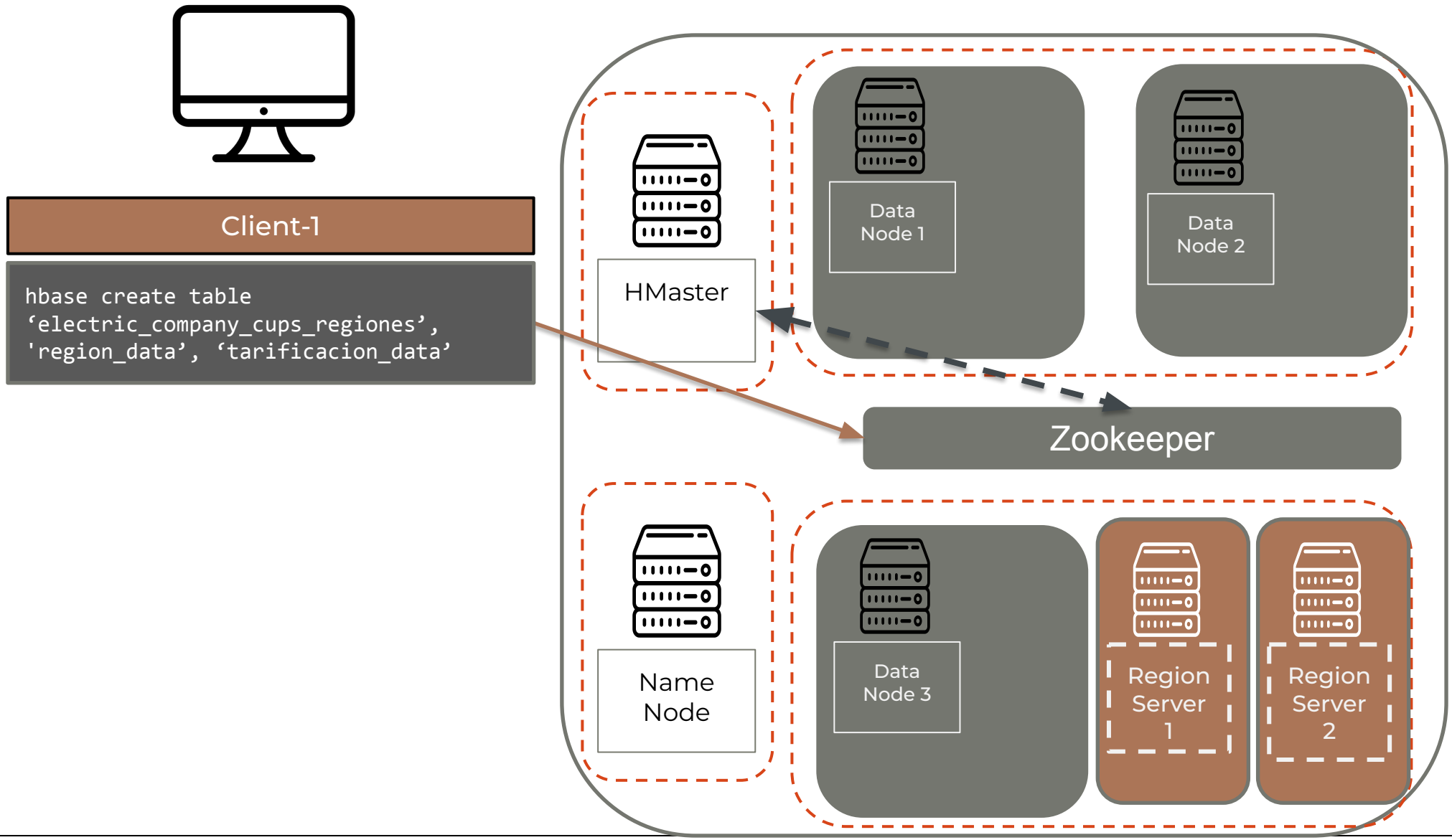
HBase Funcionamiento



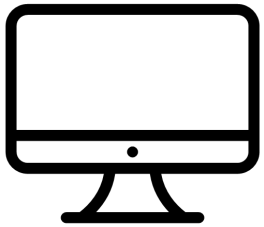
HBase Funcionamiento



HBase Funcionamiento



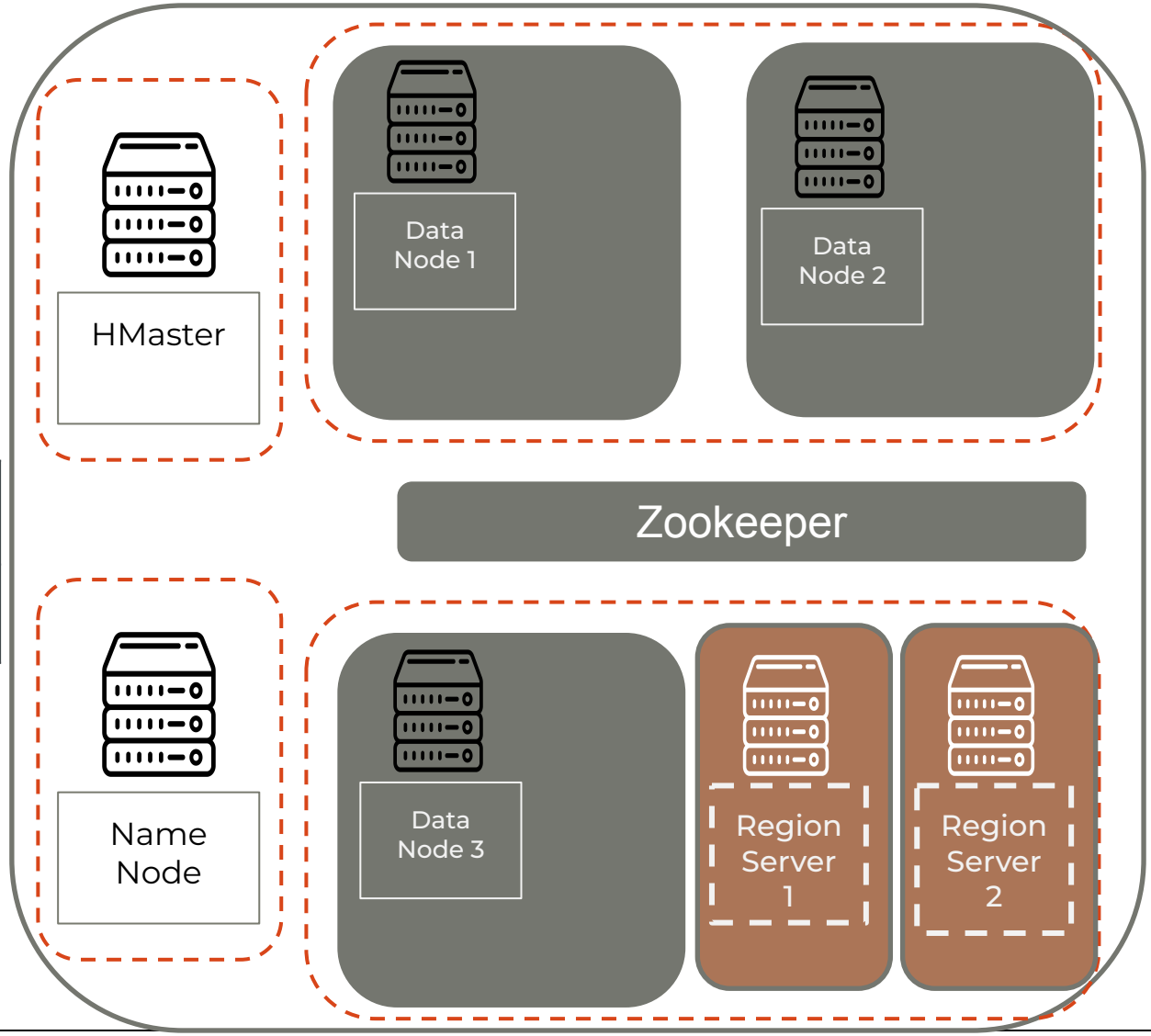
HBase Funcionamiento



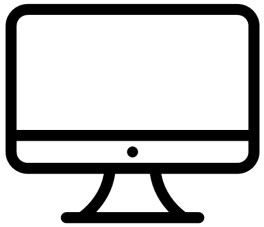
Client-1

```
hbase create table
‘electric_company_cups_regiones’,
'region_data’, ‘tarificacion_data’
```

Row	Region_data	Tarificacion_data	timestamp



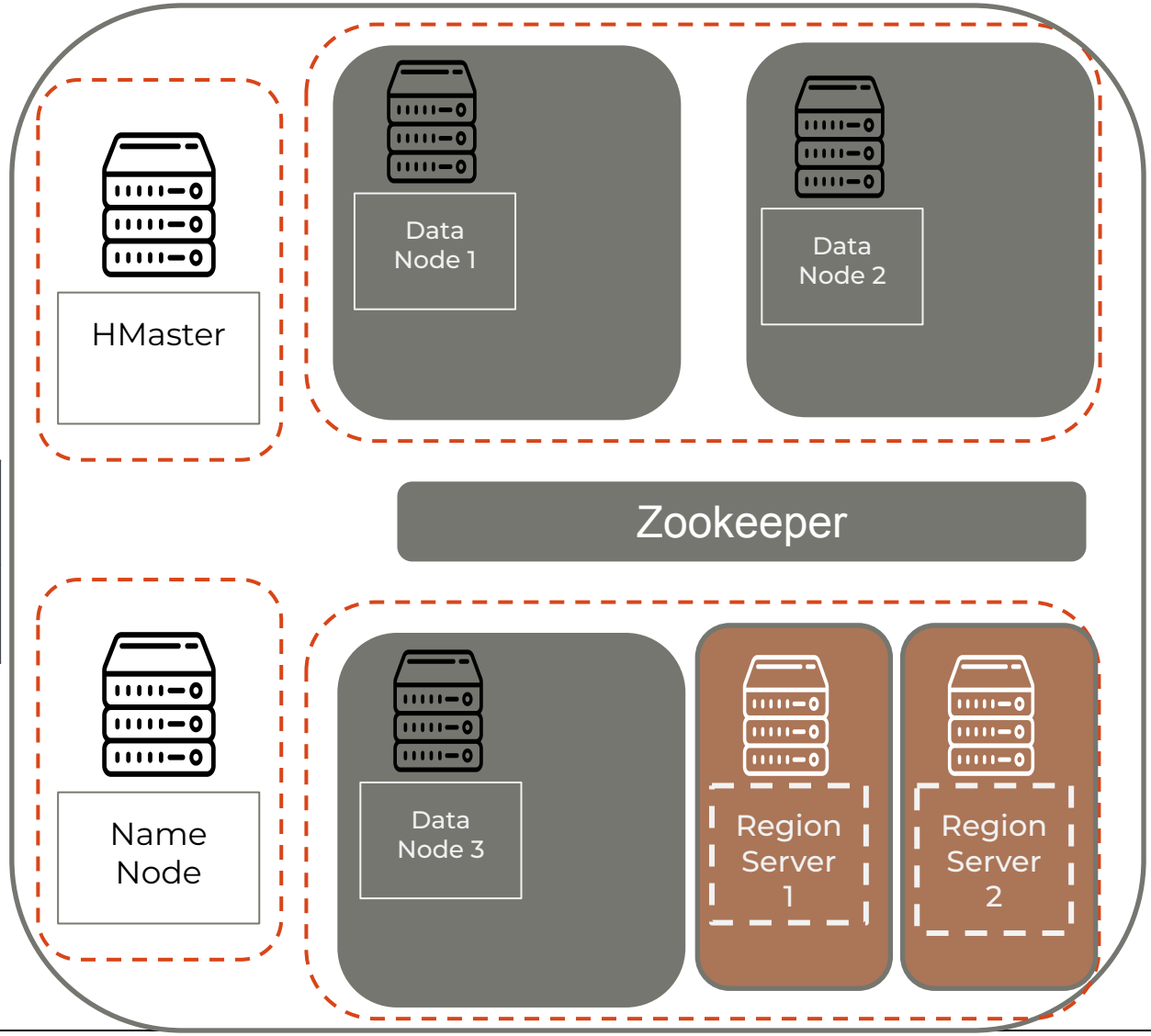
HBase Funcionamiento



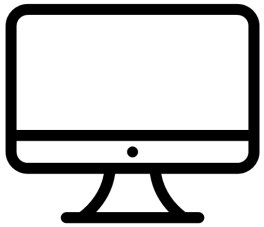
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, 'region_data:valor_region’
‘Colmenar Viejo’
```

Row	Region_data	Tarificacion_data	timestamp



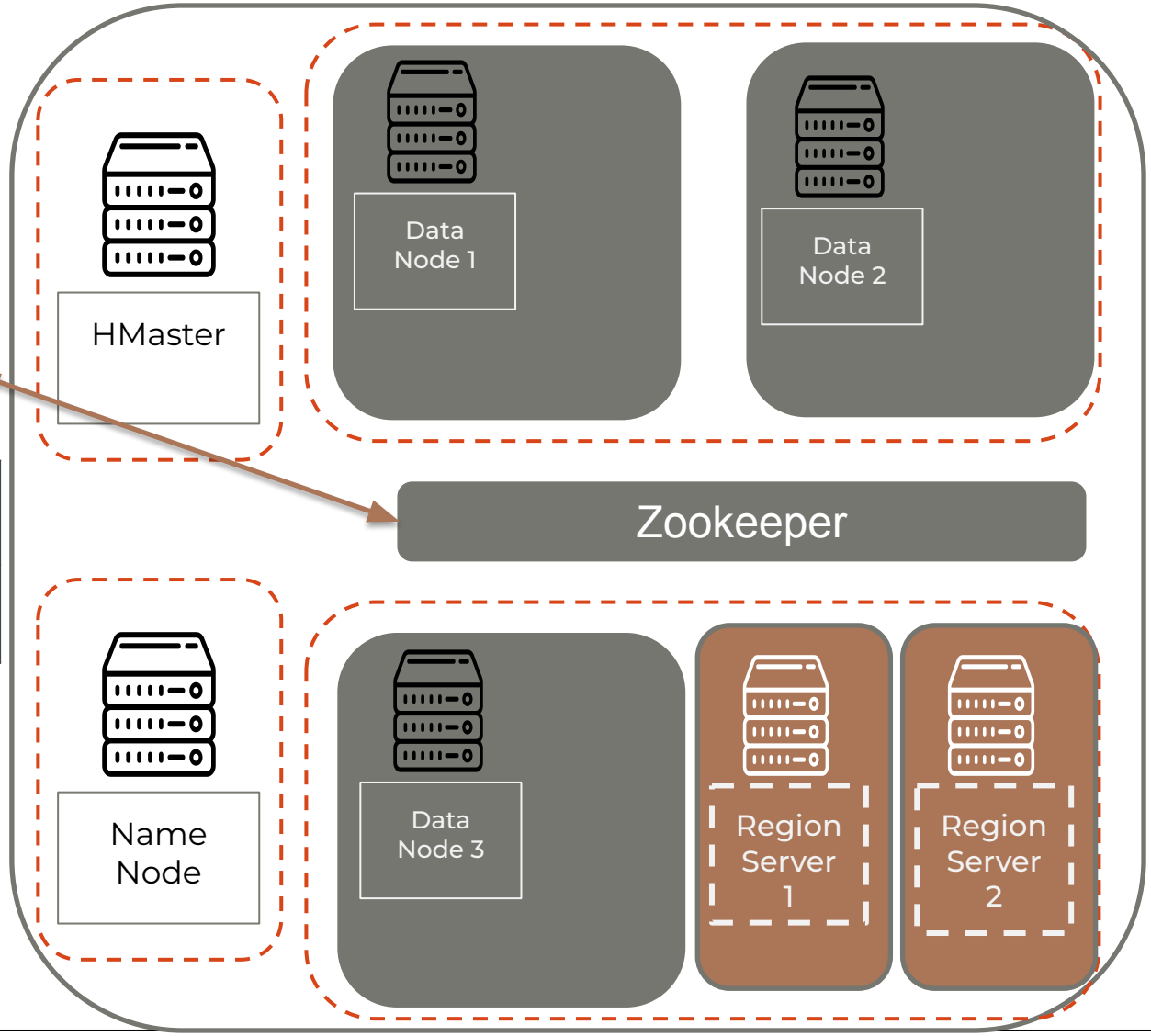
HBase Funcionamiento



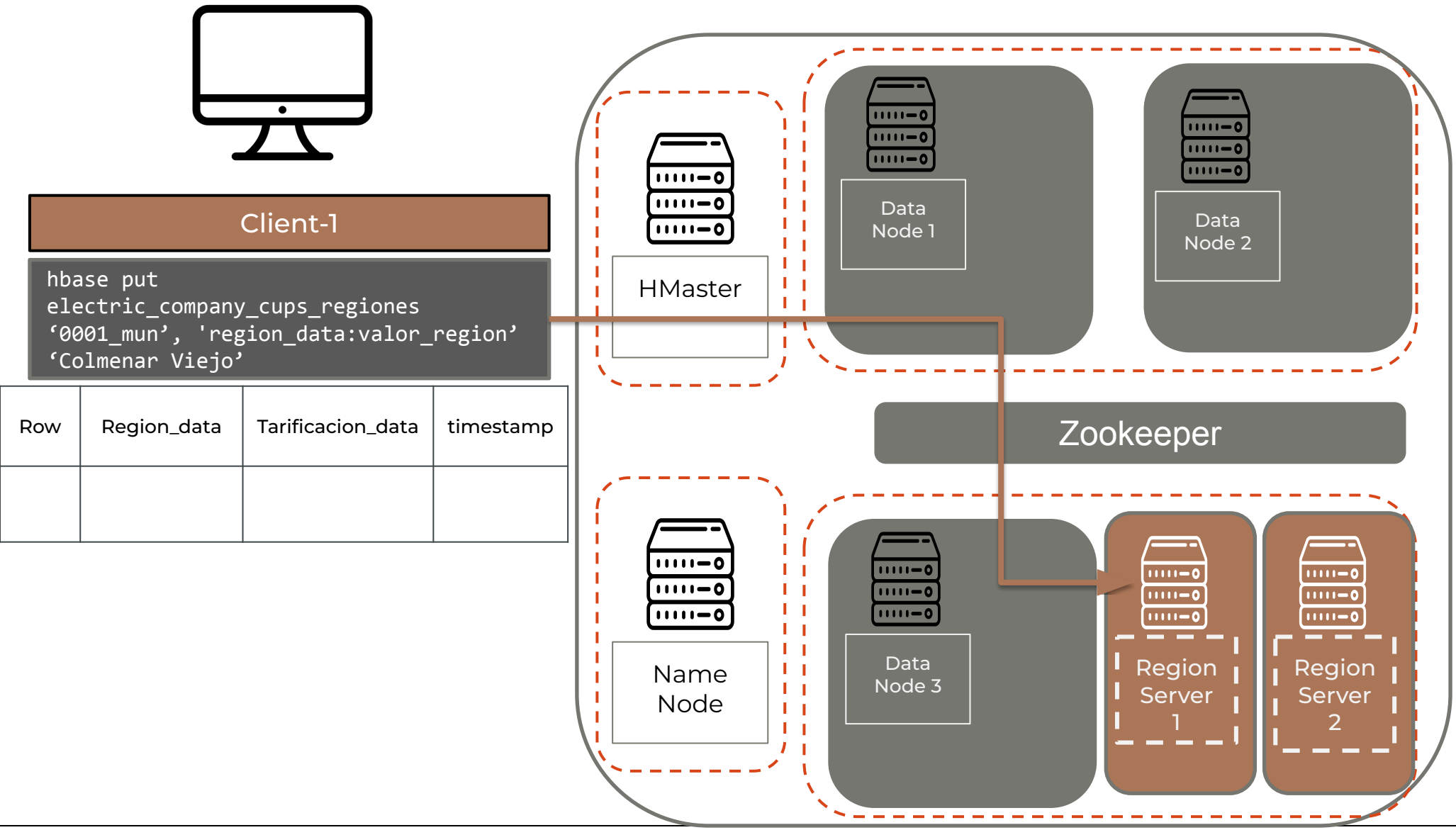
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, 'region_data:valor_region’
‘Colmenar Viejo’
```

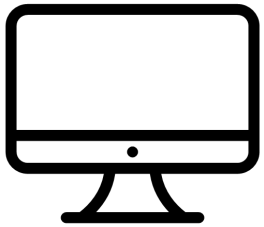
Row	Region_data	Tarificacion_data	timestamp



HBase Funcionamiento



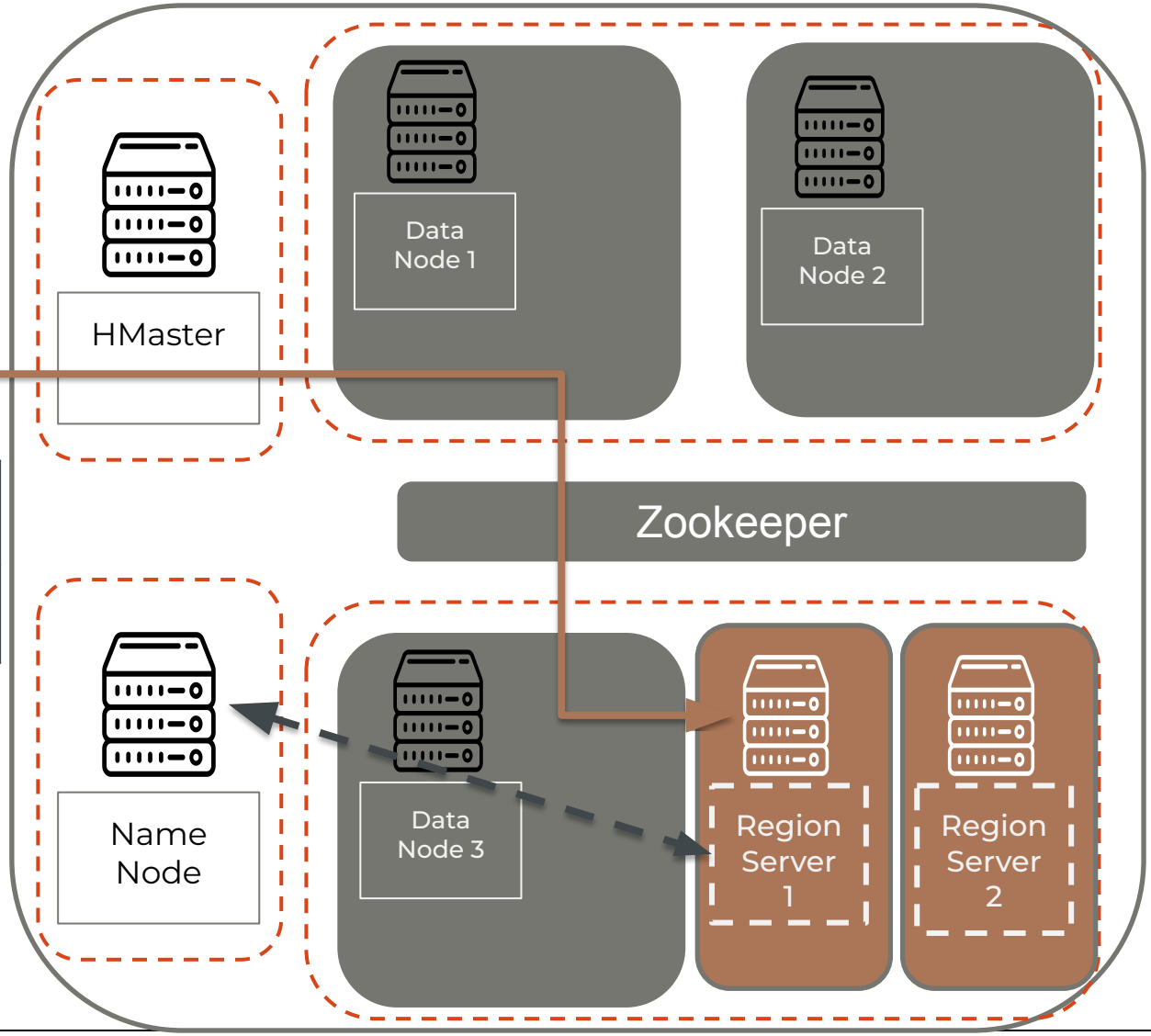
HBase Funcionamiento



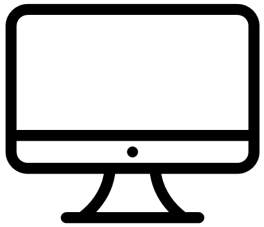
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, 'region_data:valor_region’
‘Colmenar Viejo’
```

Row	Region_data	Tarificacion_data	timestamp



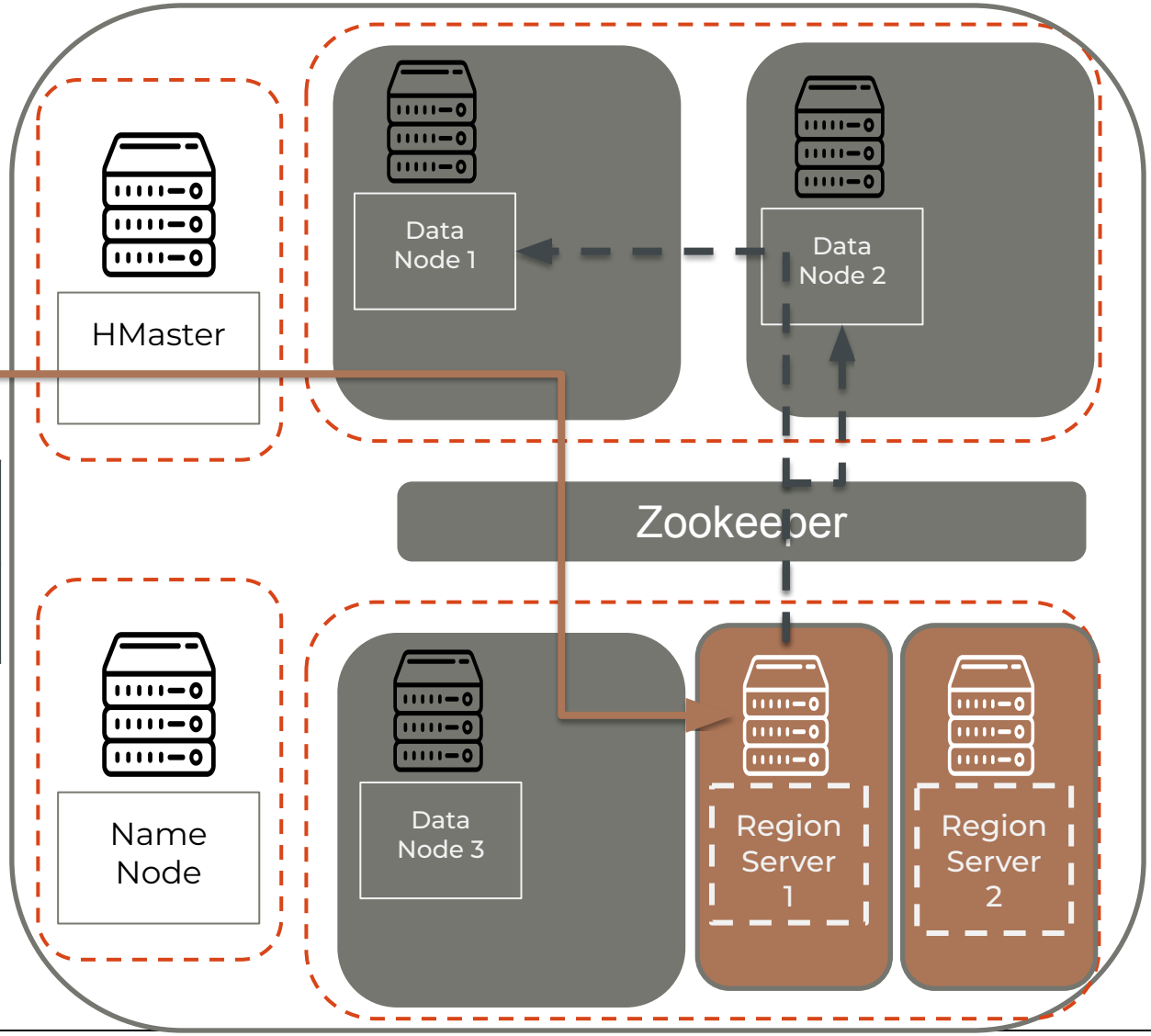
HBase Funcionamiento



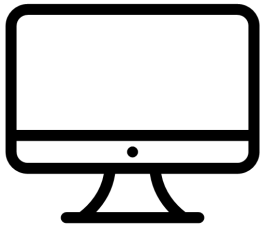
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, 'region_data:valor_region’
‘Colmenar Viejo’
```

Row	Region_data	Tarificacion_data	timestamp



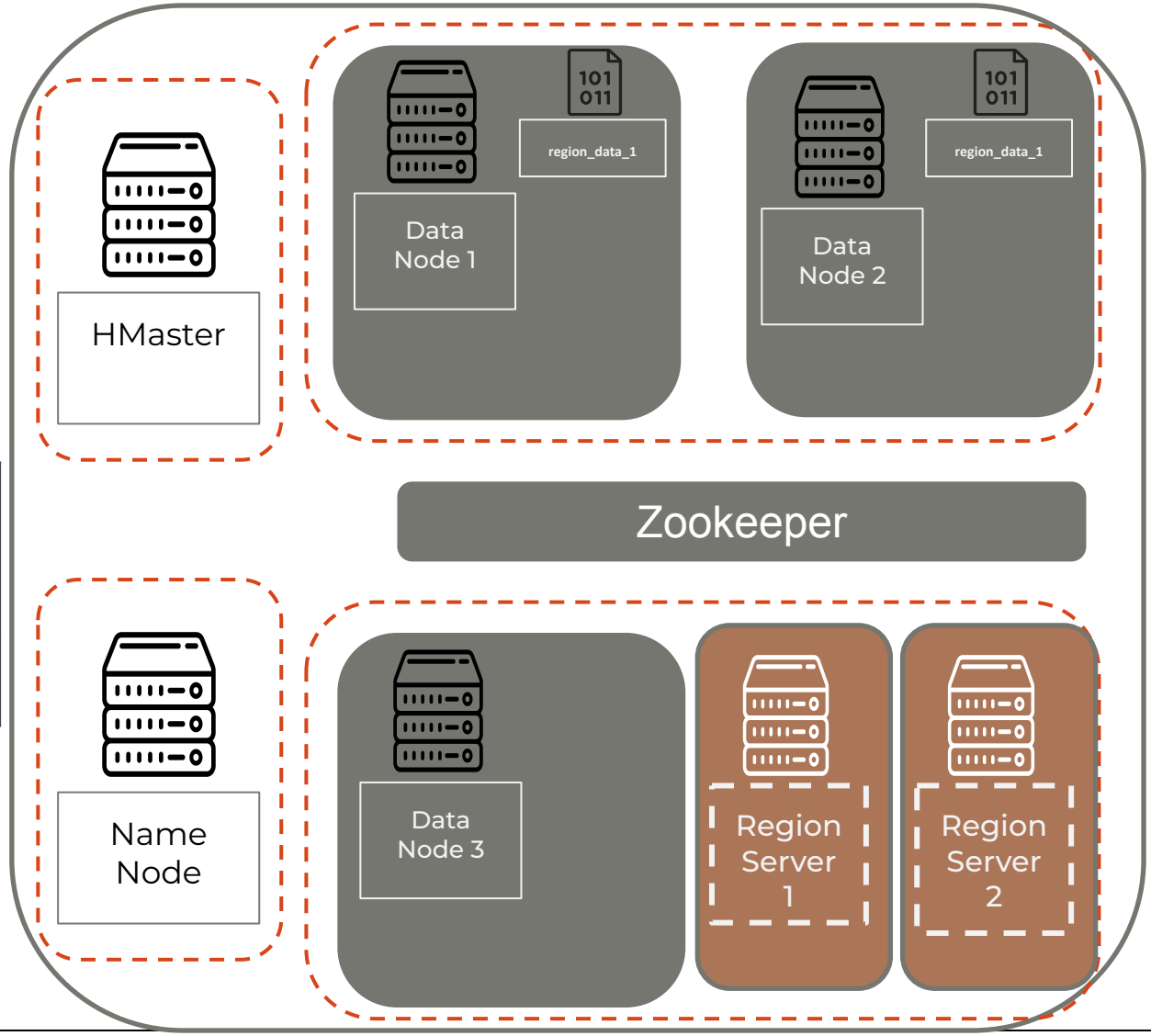
HBase Funcionamiento



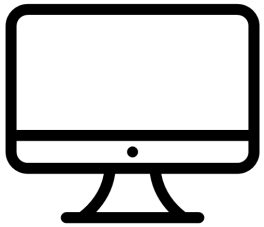
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, 'region_data:valor_region’
‘Colmenar Viejo’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000



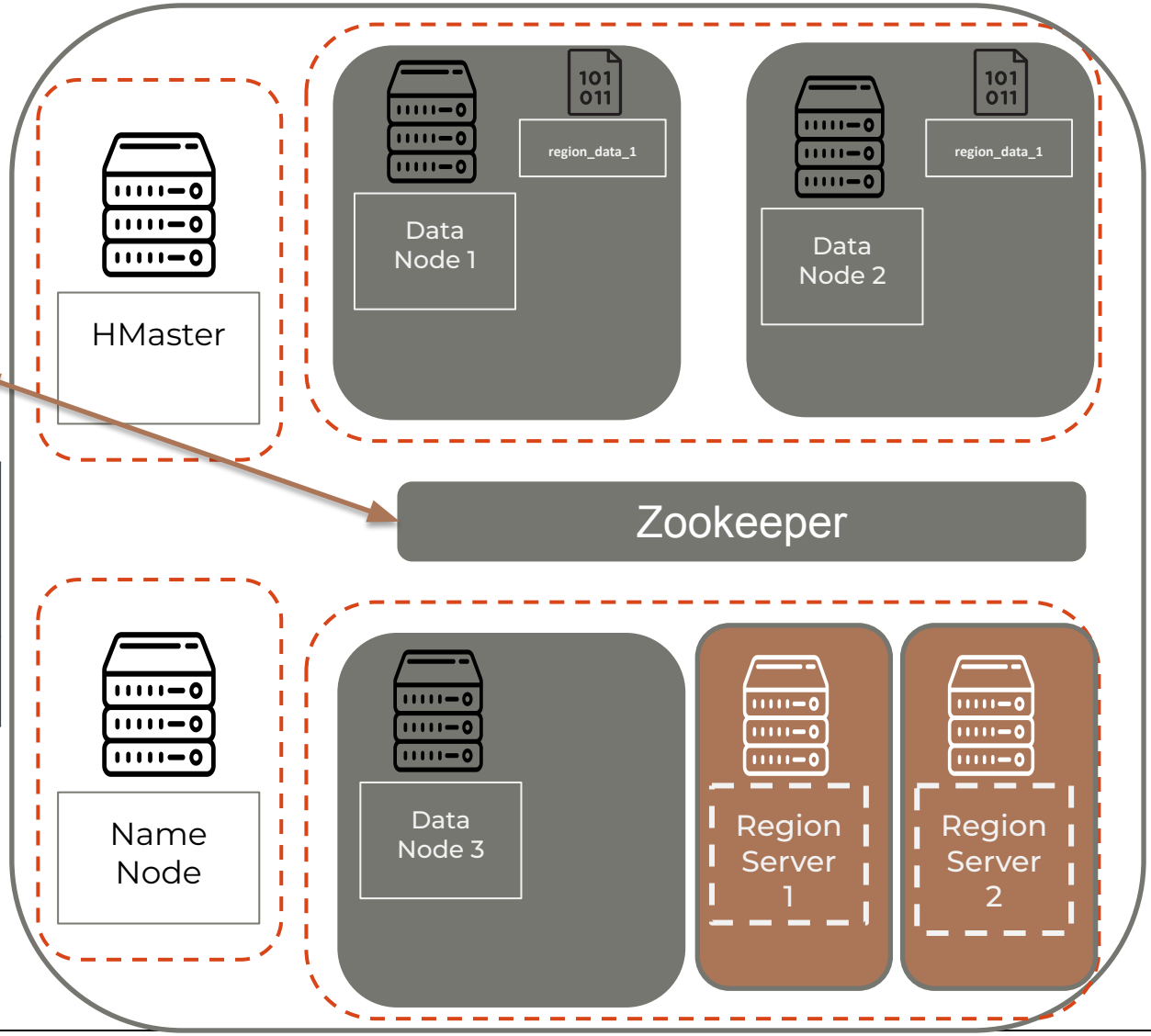
HBase Funcionamiento



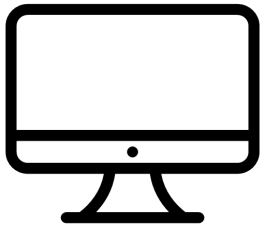
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data’:’valle
10.34’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000



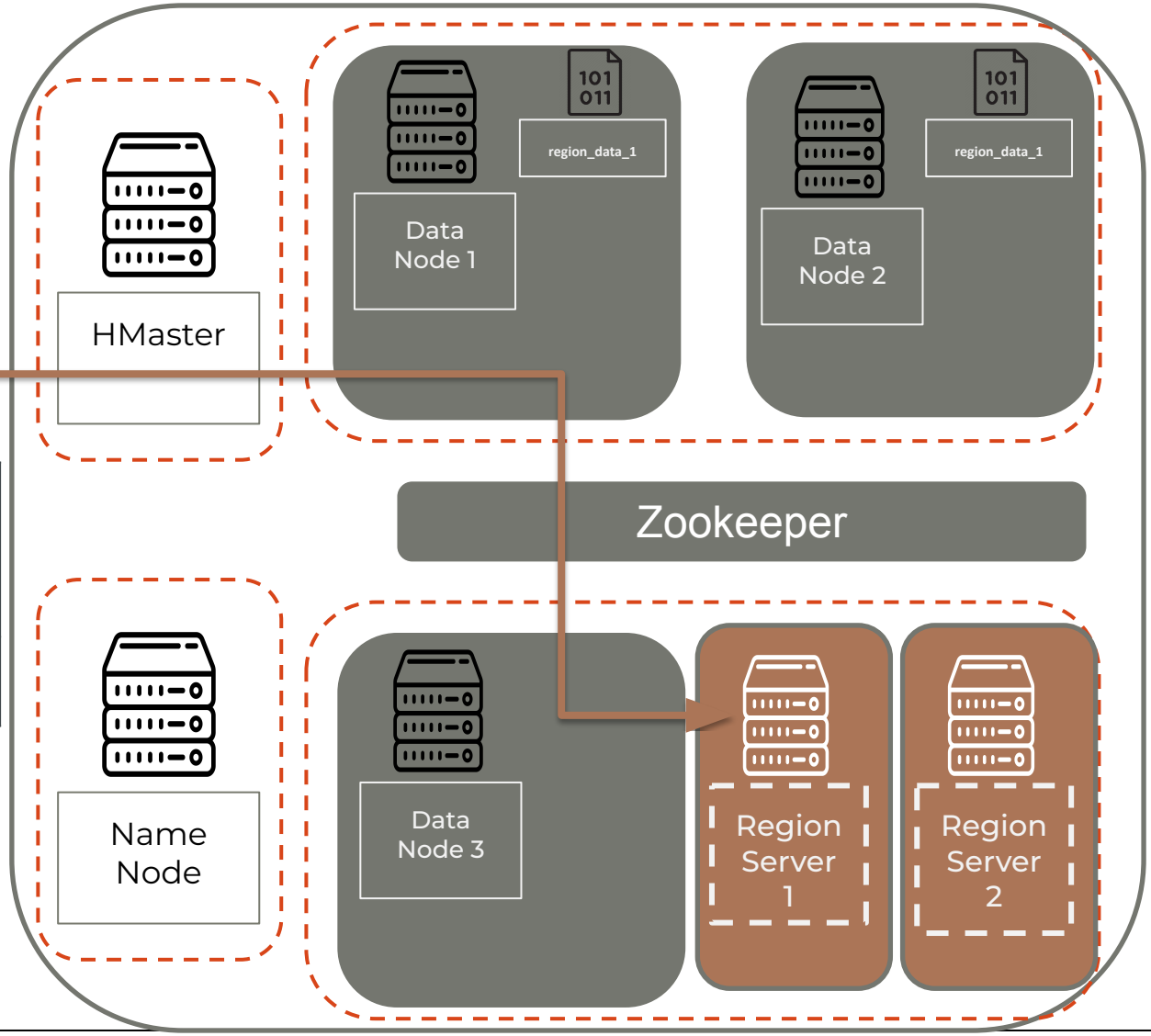
HBase Funcionamiento



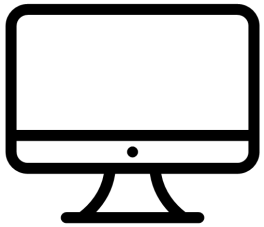
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data’:’valle
10.34’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000



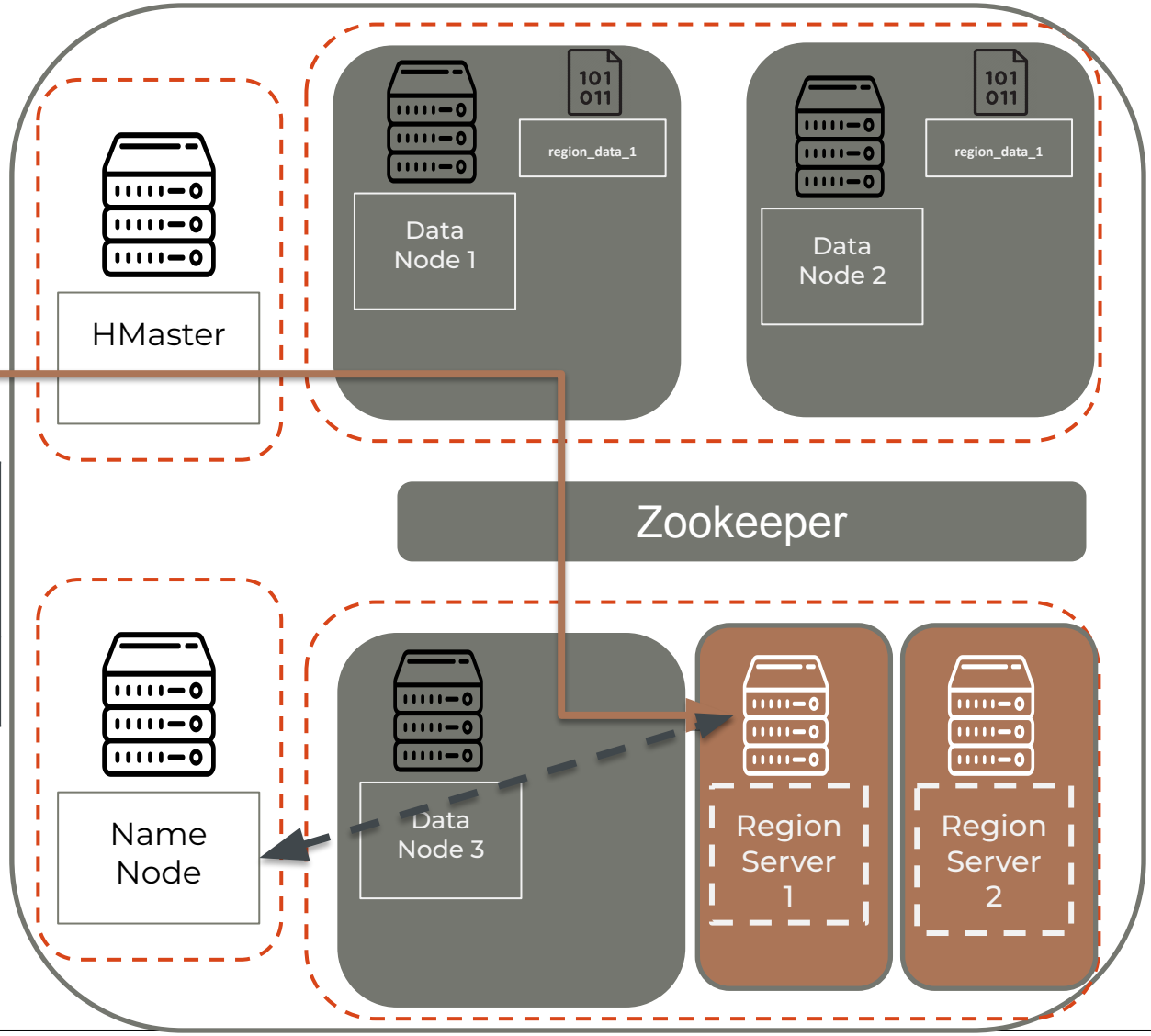
HBase Funcionamiento



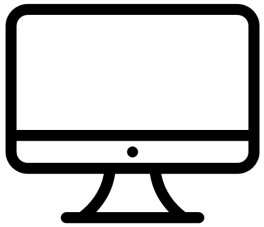
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data’:’valle
10.34’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000



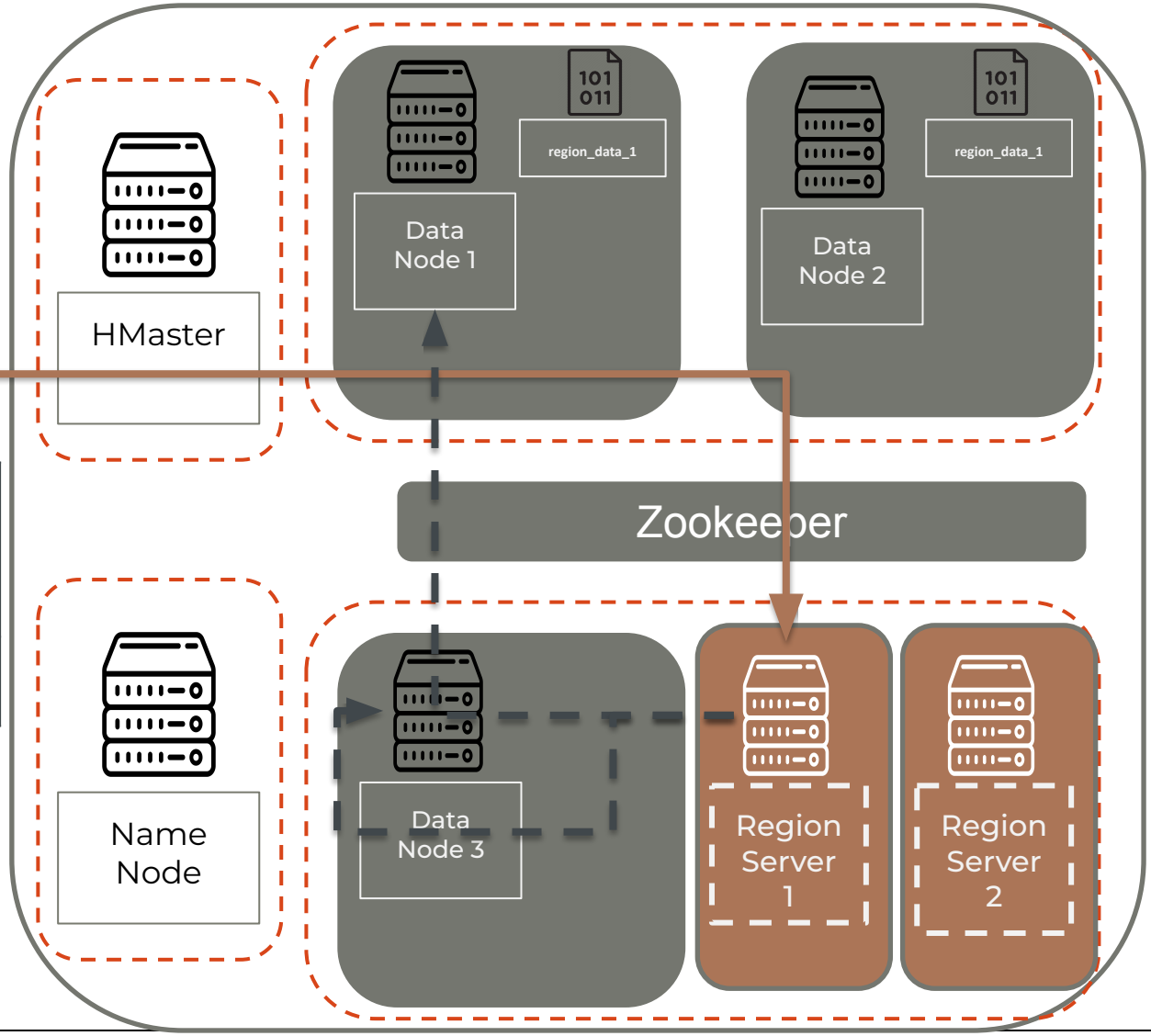
HBase Funcionamiento



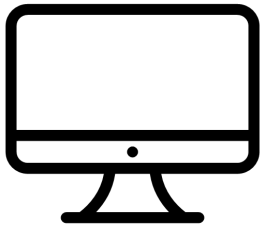
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data’:’valle
10.34’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000



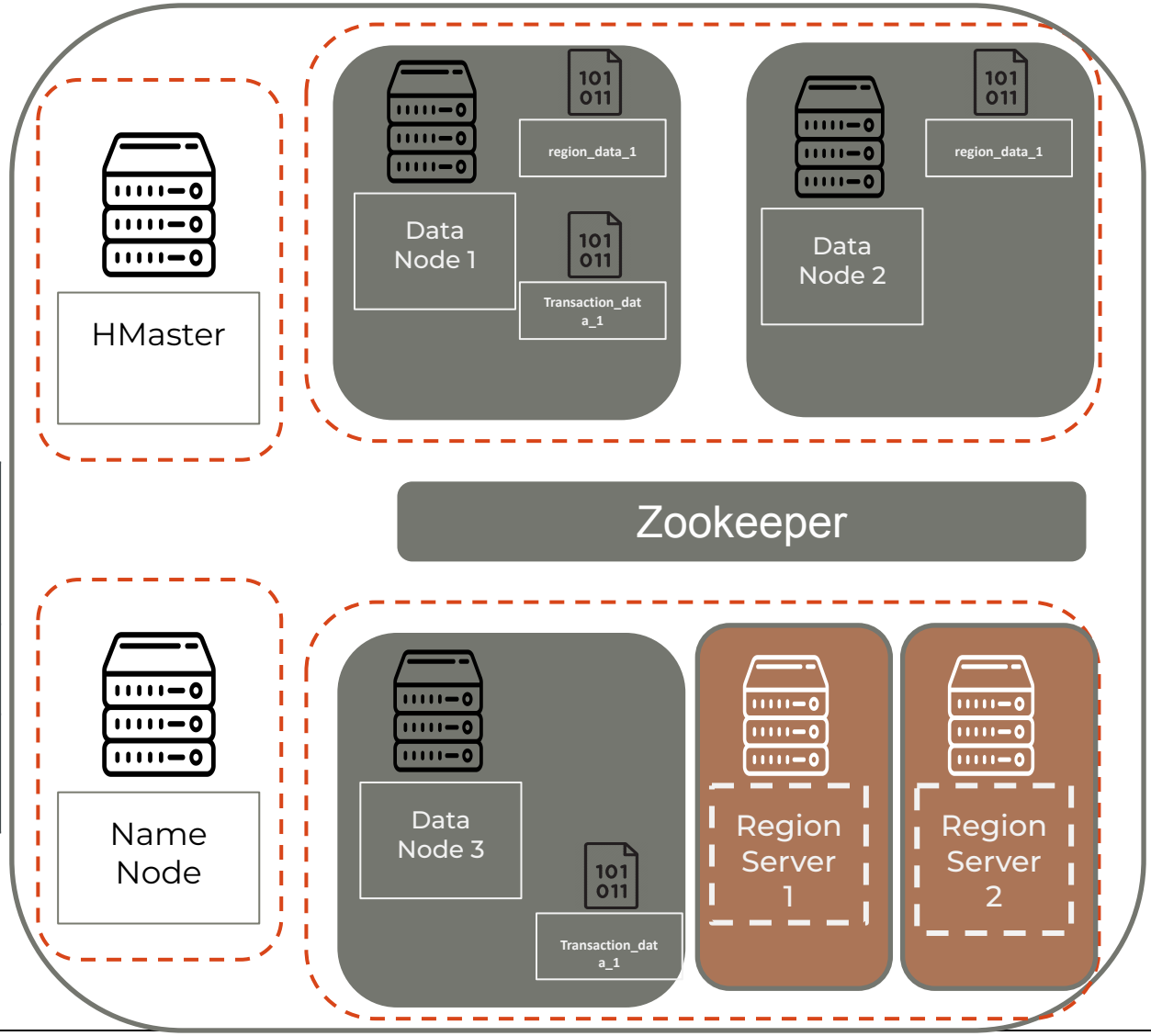
HBase Funcionamiento



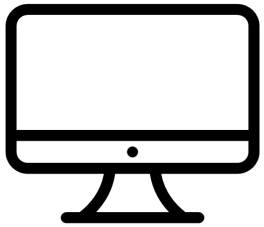
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data’:’valle
10.34’
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000
0001_mun		10.34	1631952400000



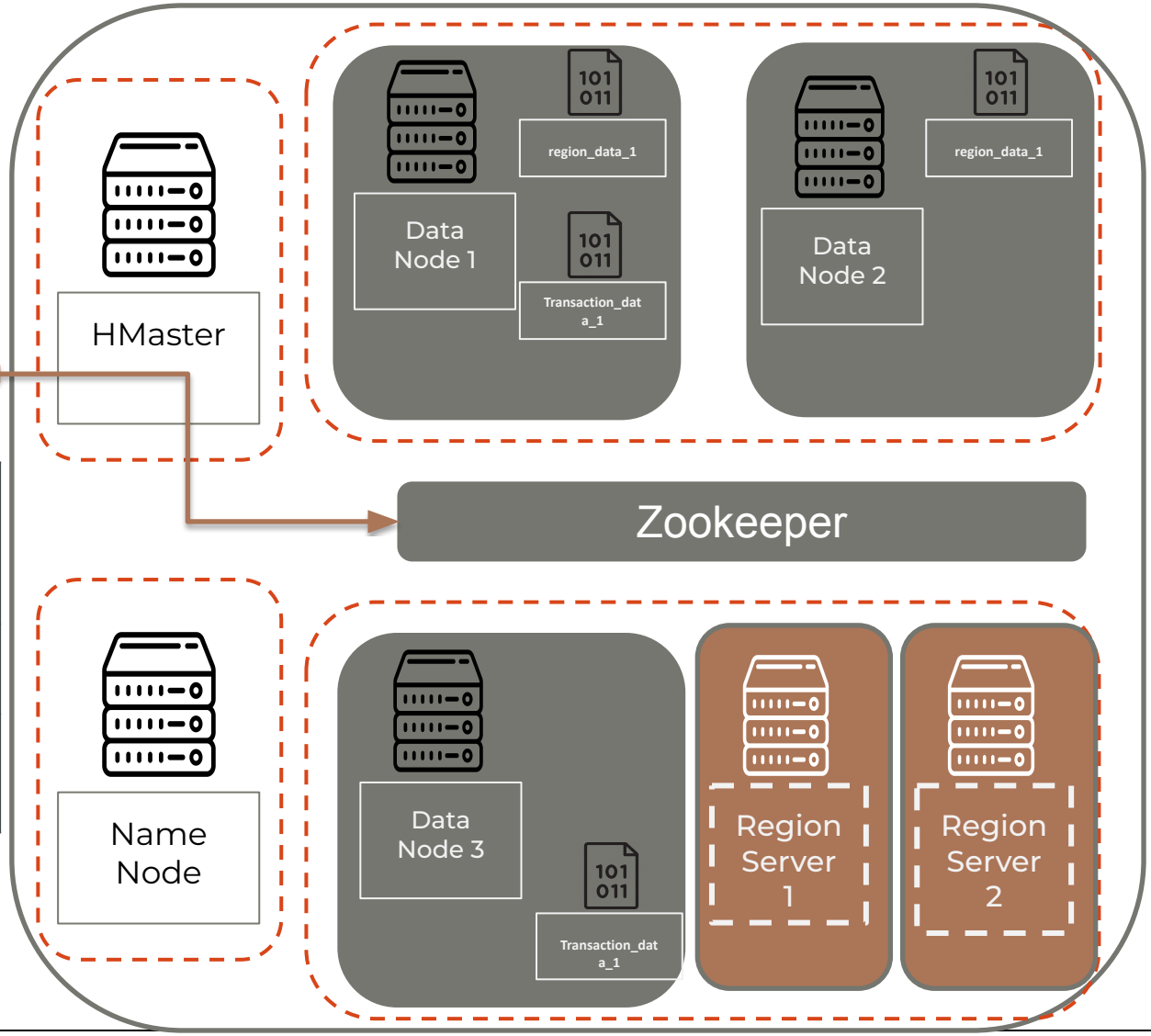
HBase Funcionamiento



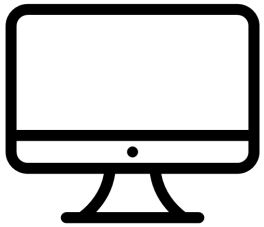
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data:pico‘
‘11.34‘
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000
0001_mun		10.34	1631952400000



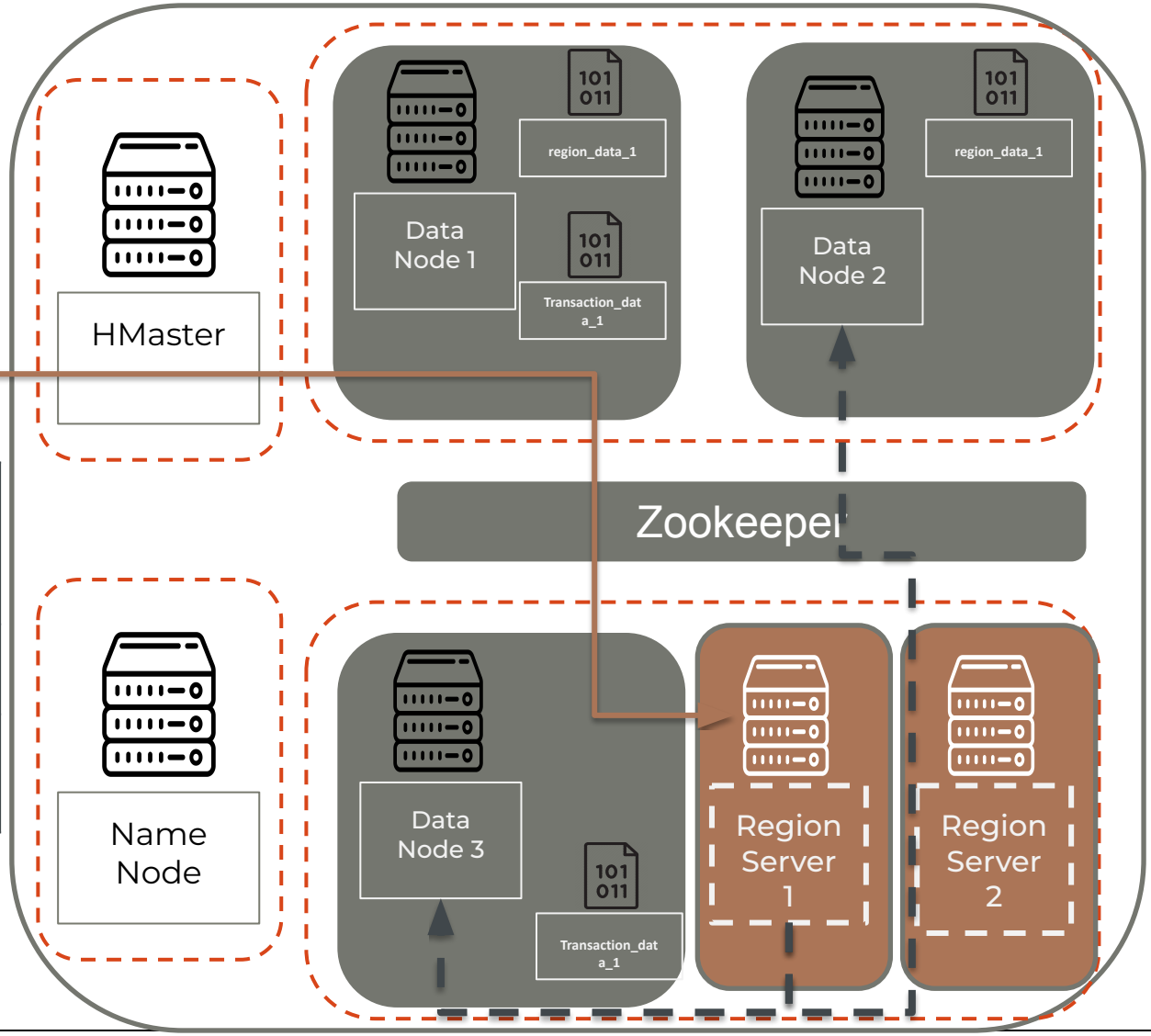
HBase Funcionamiento



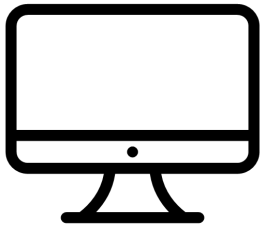
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data:pico‘
‘11.34‘
```

Row	Region_data	Tarificacion_data	timestamp
	valor_region		
0001_mun	Colmenar Viejo		1631952266000
0001_mun		10.34	1631952400000



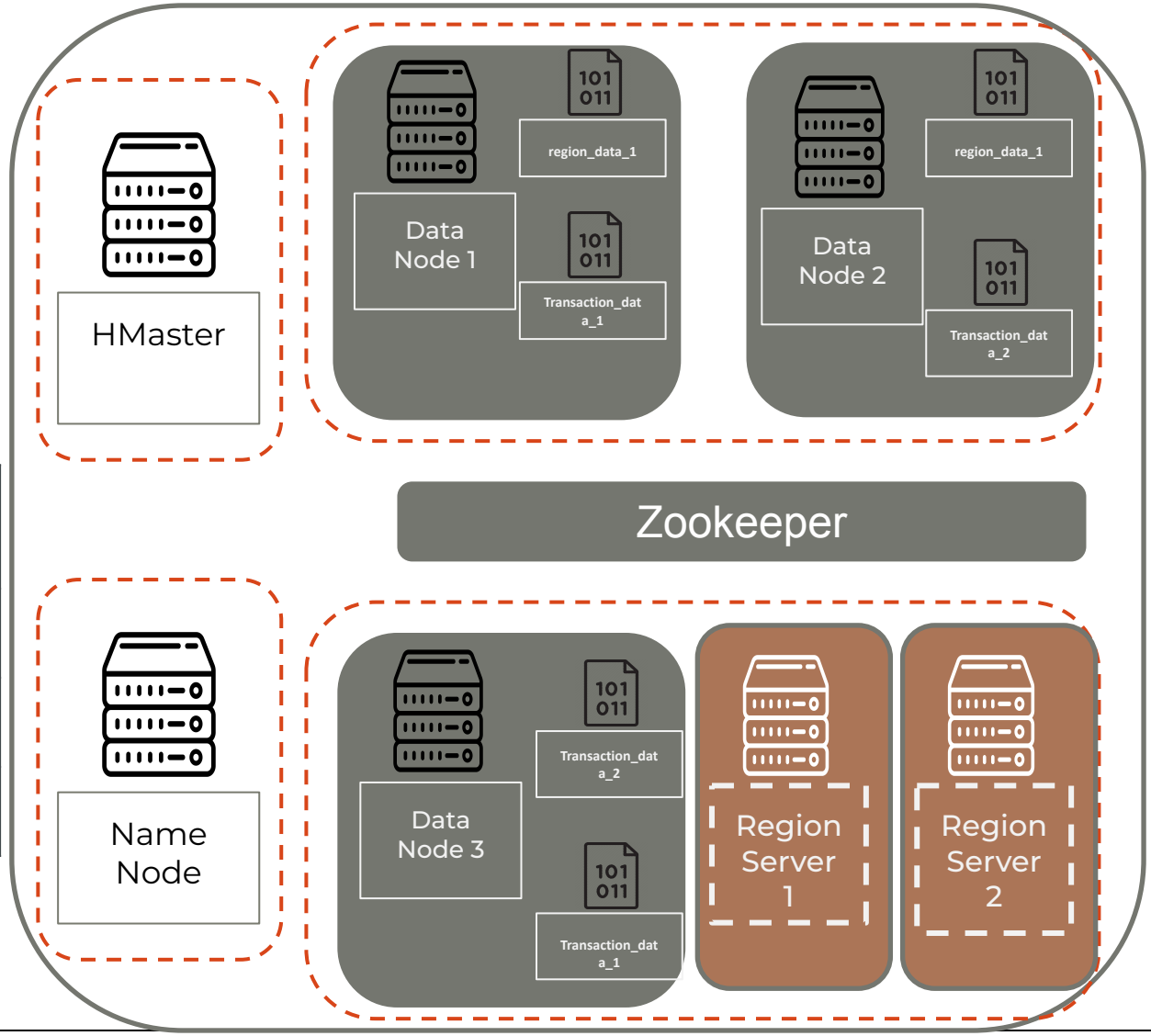
HBase Funcionamiento



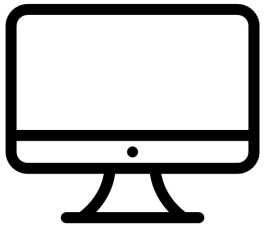
Client-1

```
hbase put
electric_company_cups_regiones
‘0001_mun’, ‘Tarificacion_data:pico‘
‘11.34‘
```

Row	Region_data	Tarificacion_data		timestamp
	valor_region	valle	pico	
0001_mun	Colmenar Viejo			1631952266000
0001_mun		10.34		1631952400000
0001_mun			11.34	1631952500000



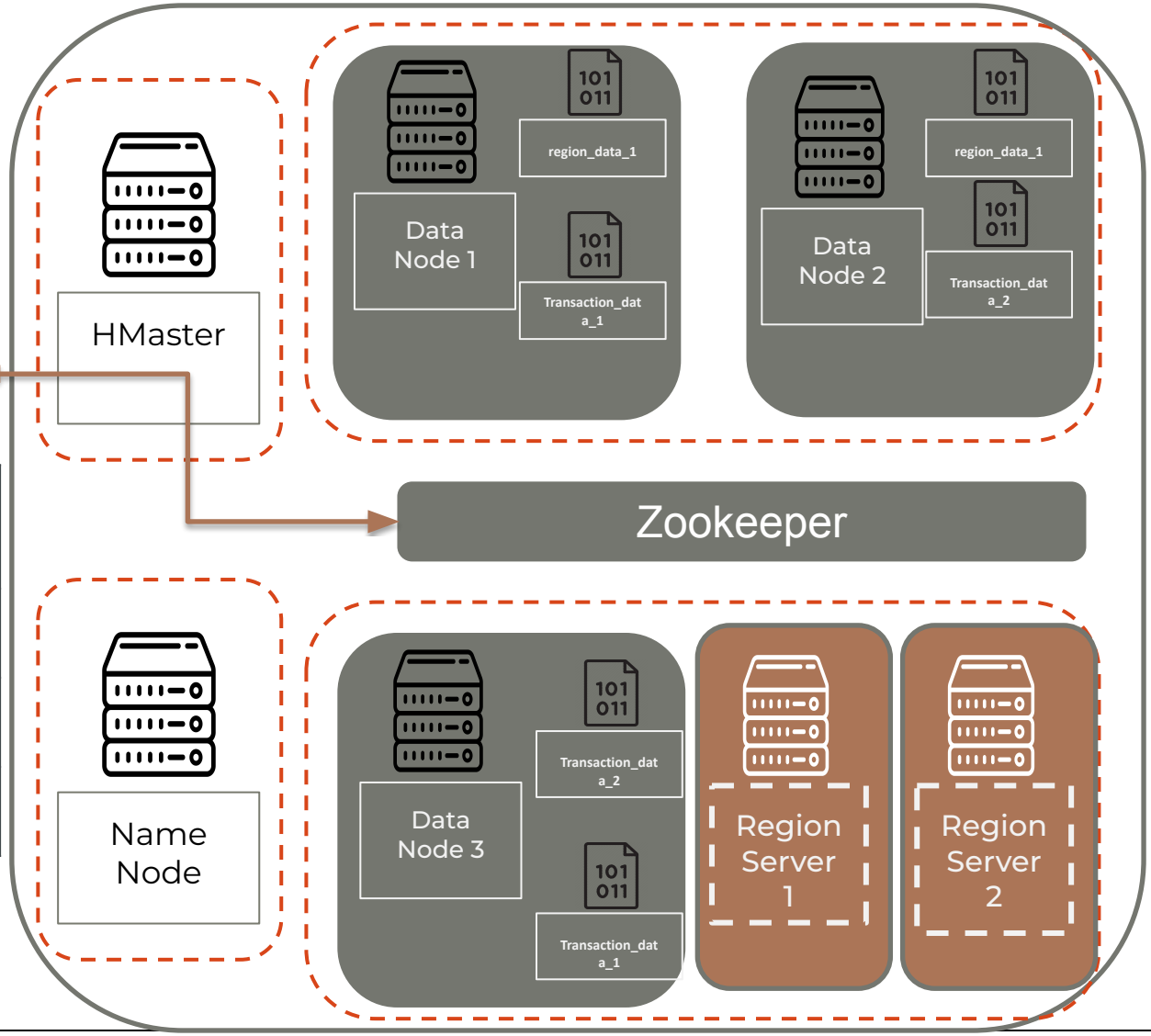
HBase Funcionamiento



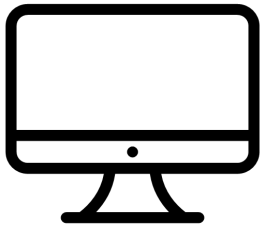
Client-1

```
hbase put
electric_company_cups_regiones
‘0002_mun’, region_data:valor_region
‘Alcalá’
```

Row	Region_data	Tarificacion_data		timestamp
	valor_region	valle	pico	
0001_mun	Colmenar Viejo			1631952266000
0001_mun		10.34		1631952400000
0001_mun			11.34	1631952500000



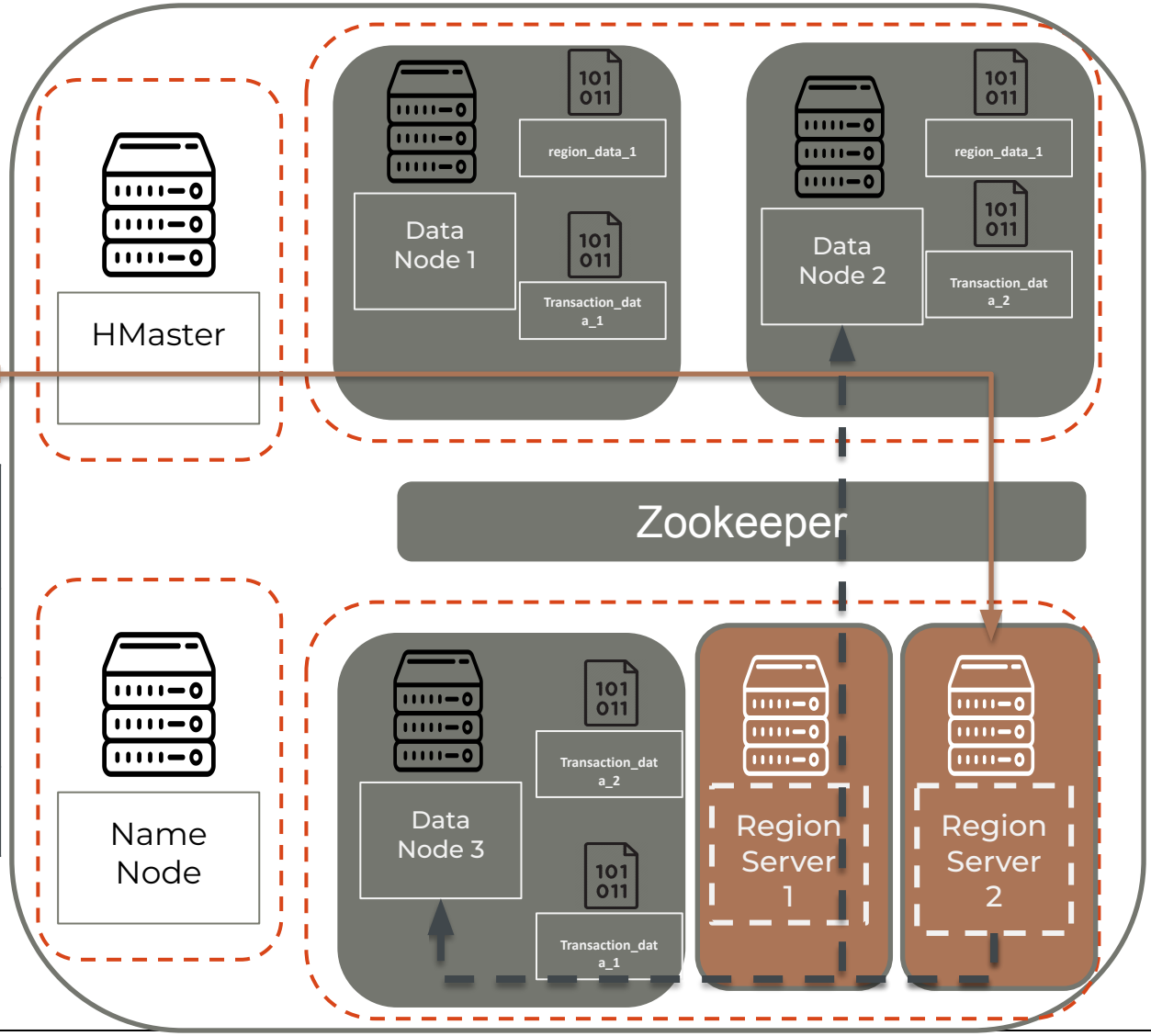
HBase Funcionamiento



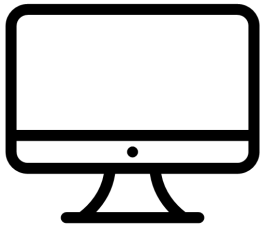
Client-1

```
hbase put
electric_company_cups_regiones
‘0002_mun’, region_data:valor_region
‘Alcalá’
```

Row	Region_data	Tarificacion_data		timestamp
	valor_region	valle	pico	
0001_mun	Colmenar Viejo			1631952266000
0001_mun		10.34		1631952400000
0001_mun			11.34	1631952500000



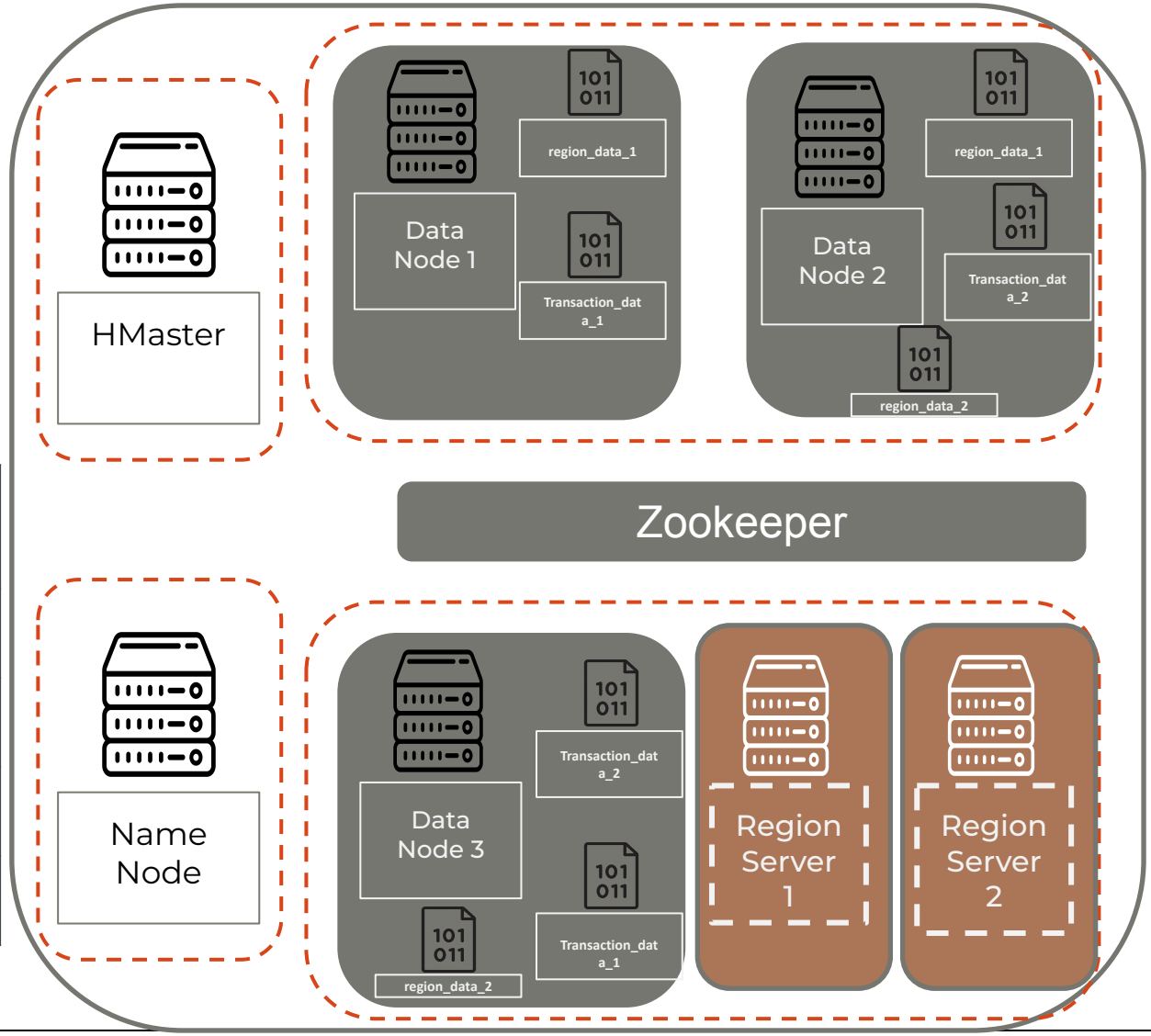
HBase Funcionamiento



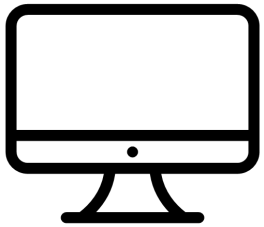
Client-1

```
hbase put
electric_company_cups_regiones
‘0002_mun’, region_data:valor_region
‘Alcalá’
```

Row	Region_data	Tarificacion_data		timestamp
	valor_region	valle	pico	
0001_mun	Colmenar Viejo			1631952266000
0001_mun		10.34		1631952400000
0001_mun			11.34	1631952500000
0002_mun	Acalá			1631952600000



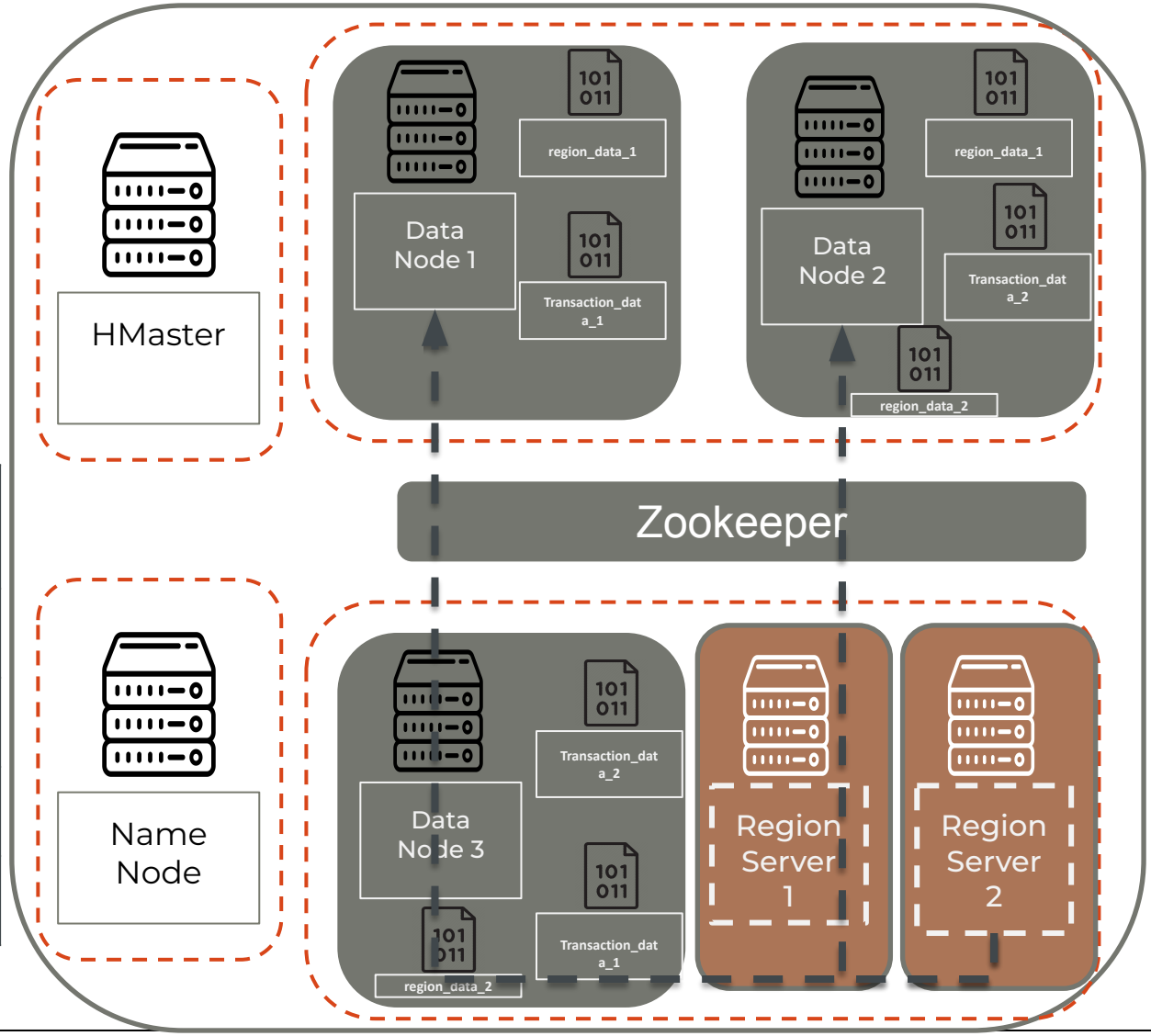
HBase Funcionamiento



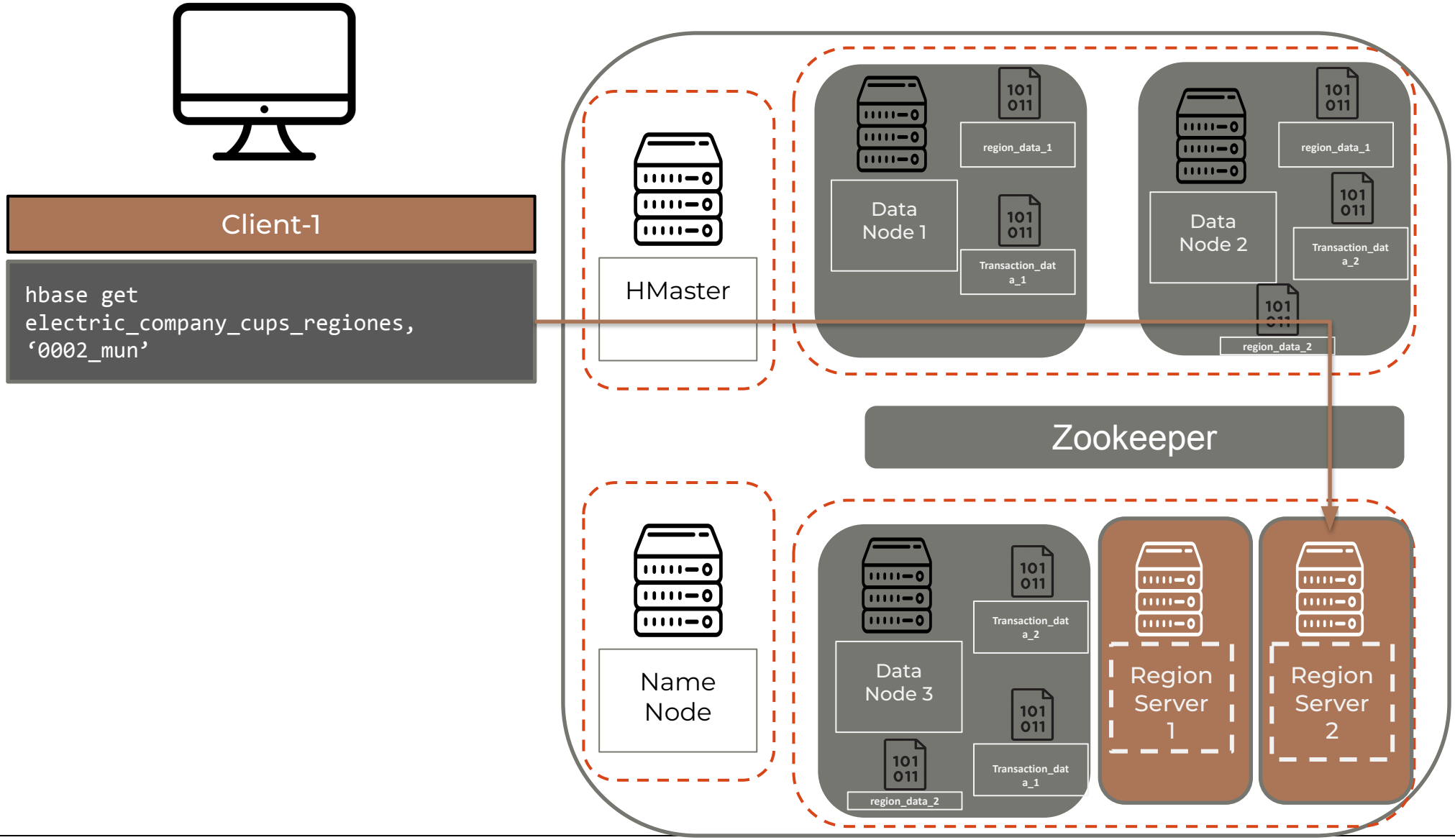
Client-1

```
hbase put
electric_company_cups_regiones
‘0002_mun’, region_data:valor_region
‘Alcalá’
```

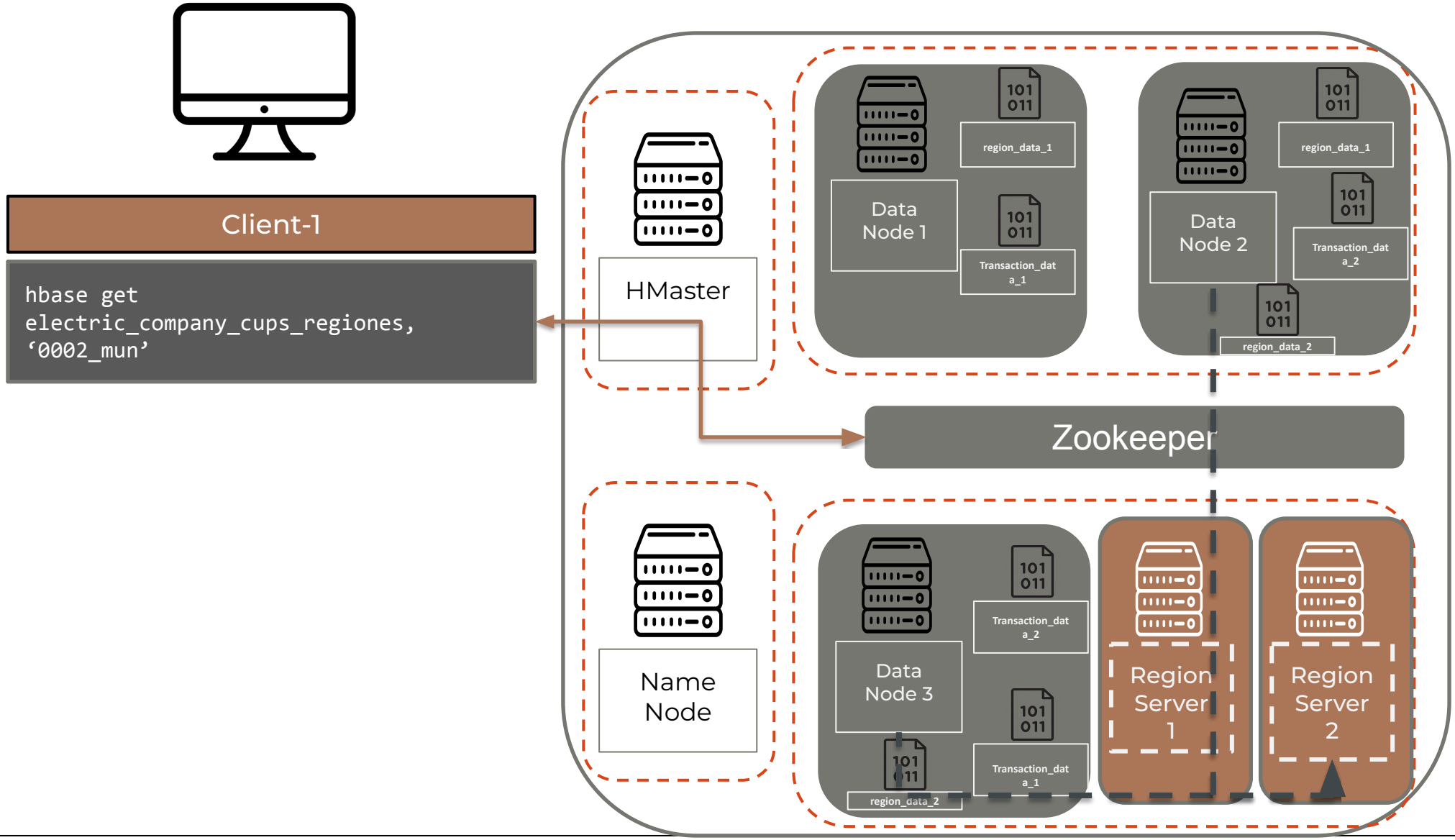
Row	Region_data	Tarificacion_data		timestamp
	valor_region	valle	pico	
0001_mun	Colmenar Viejo			1631952266000
0001_mun		10.34		1631952400000
0001_mun			11.34	1631952500000
0002_mun	Acalá			1631952600000



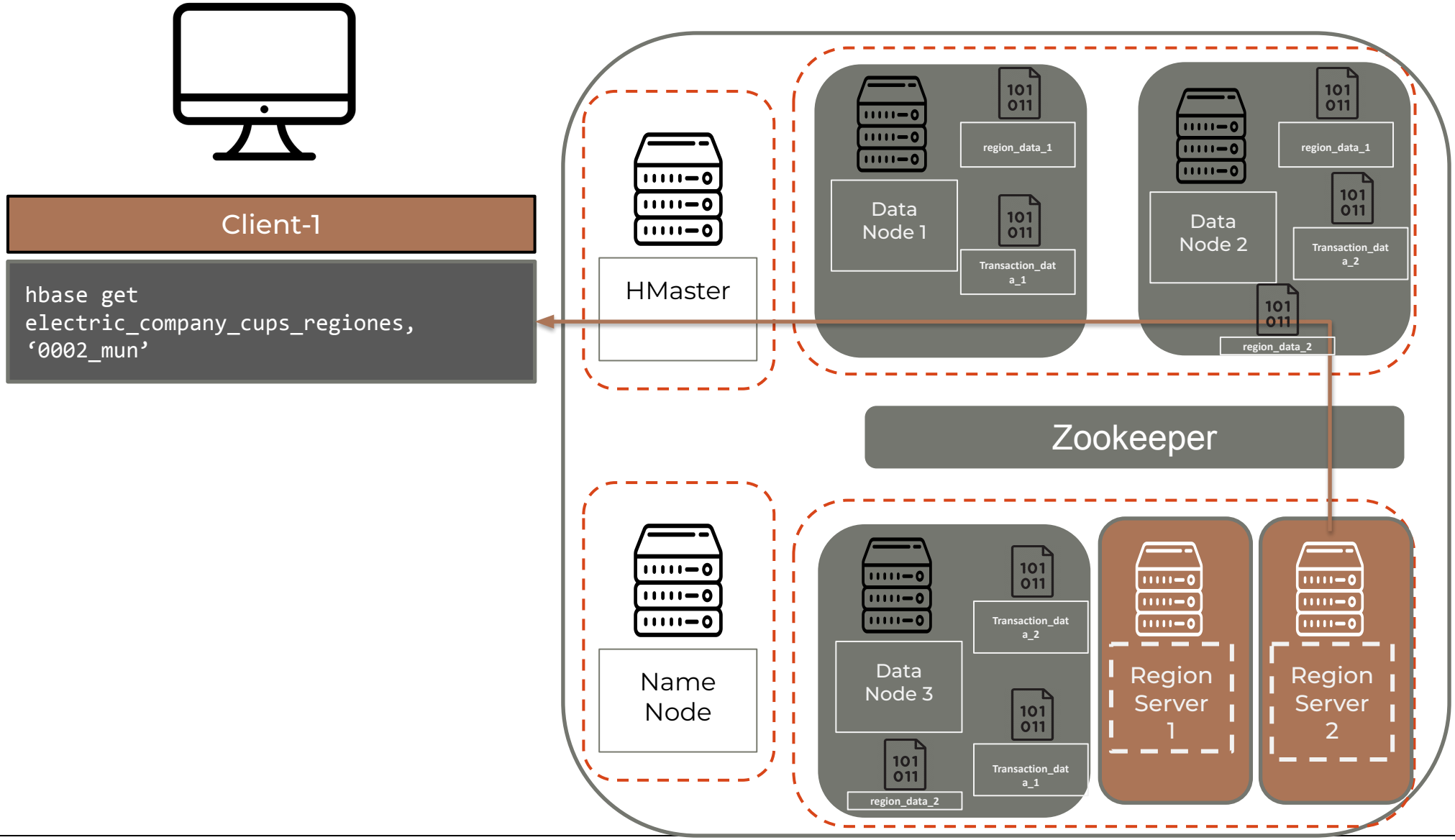
HBase Funcionamiento



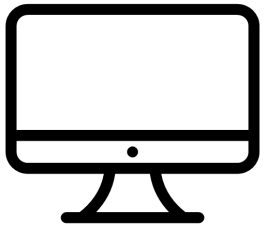
HBase Funcionamiento



HBase Funcionamiento

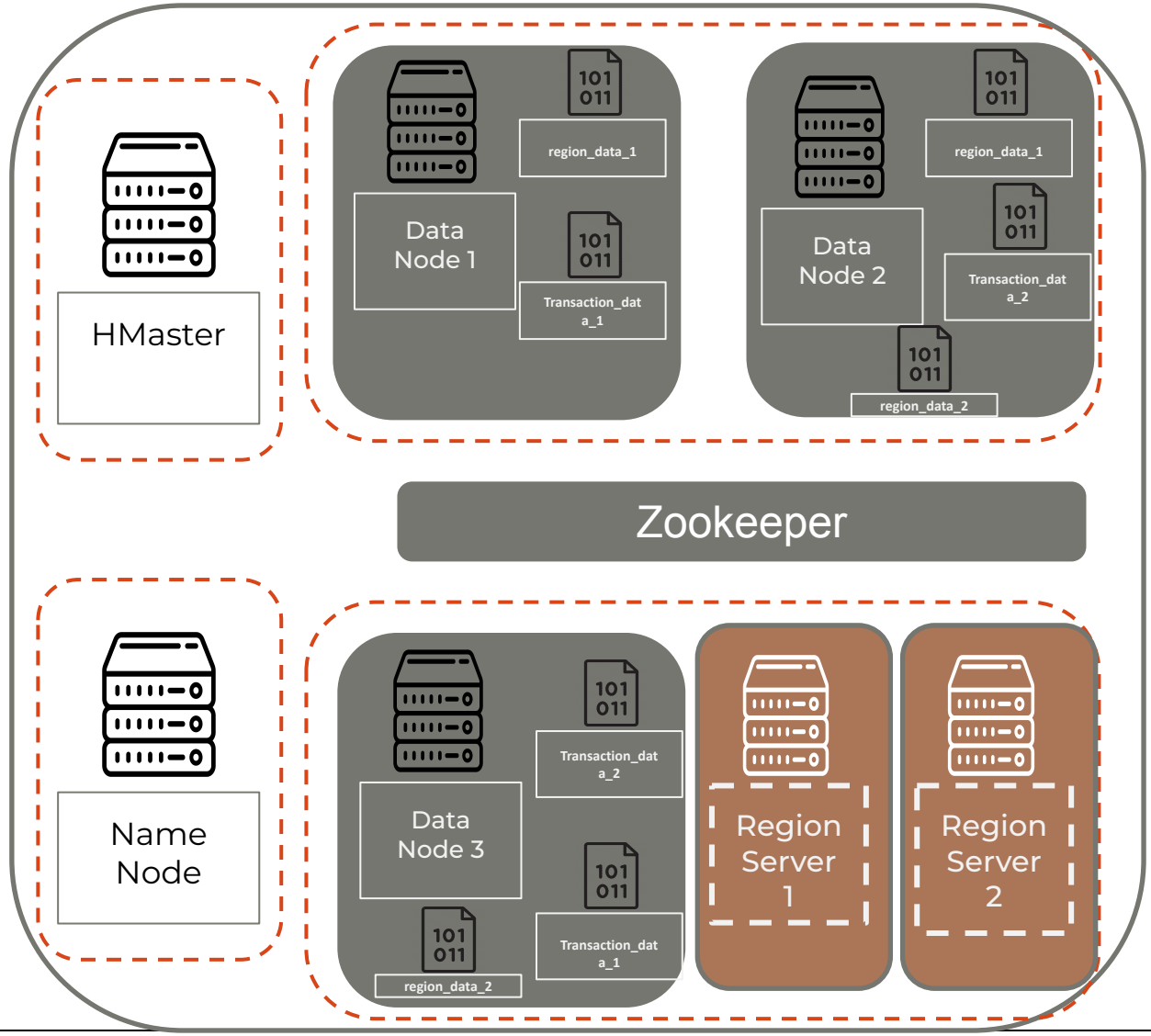


HBase Funcionamiento

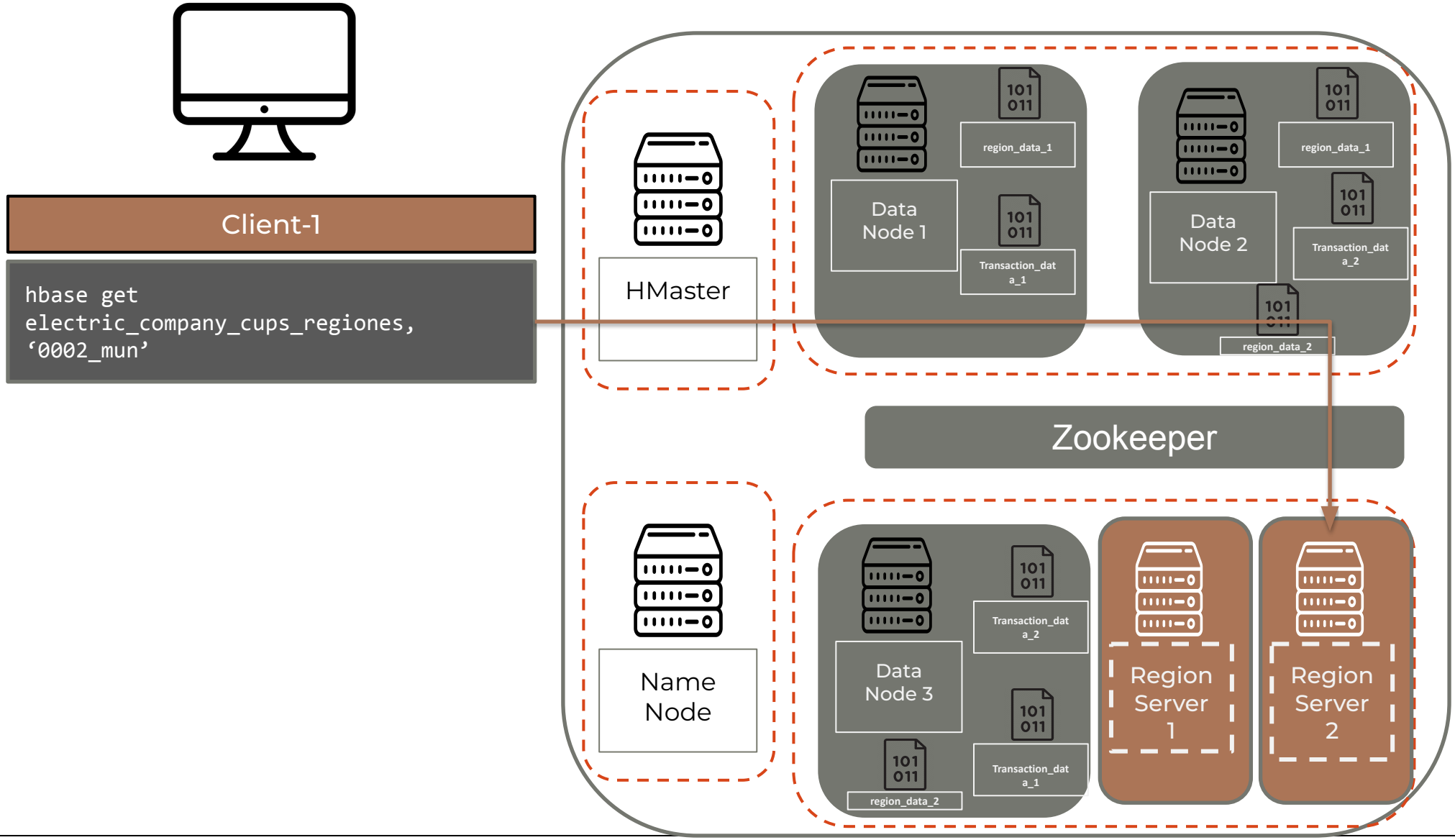


Client-1

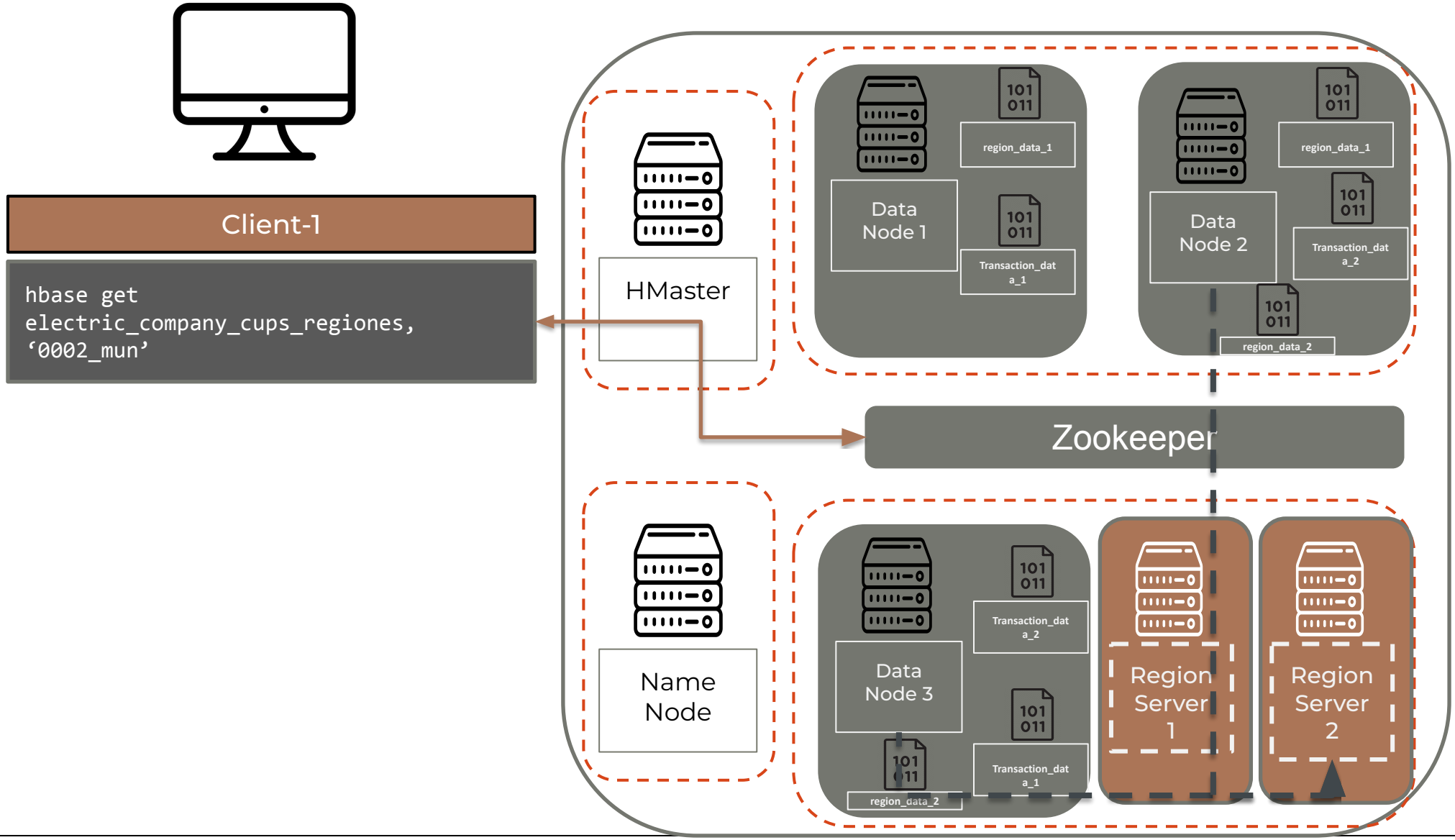
```
hbase get
electric_company_cups_regiones,
‘0002_mun’

COLUMN                          CELL
region_data : valor_region
timestamp=1631952266000,
value=’Colmenar Viejo’
Tarificacion_data:pico
 timestamp=1631952500000, value=11.34
Tarificacion_data:valle
 timestamp=1631954000000, value=100.34
```



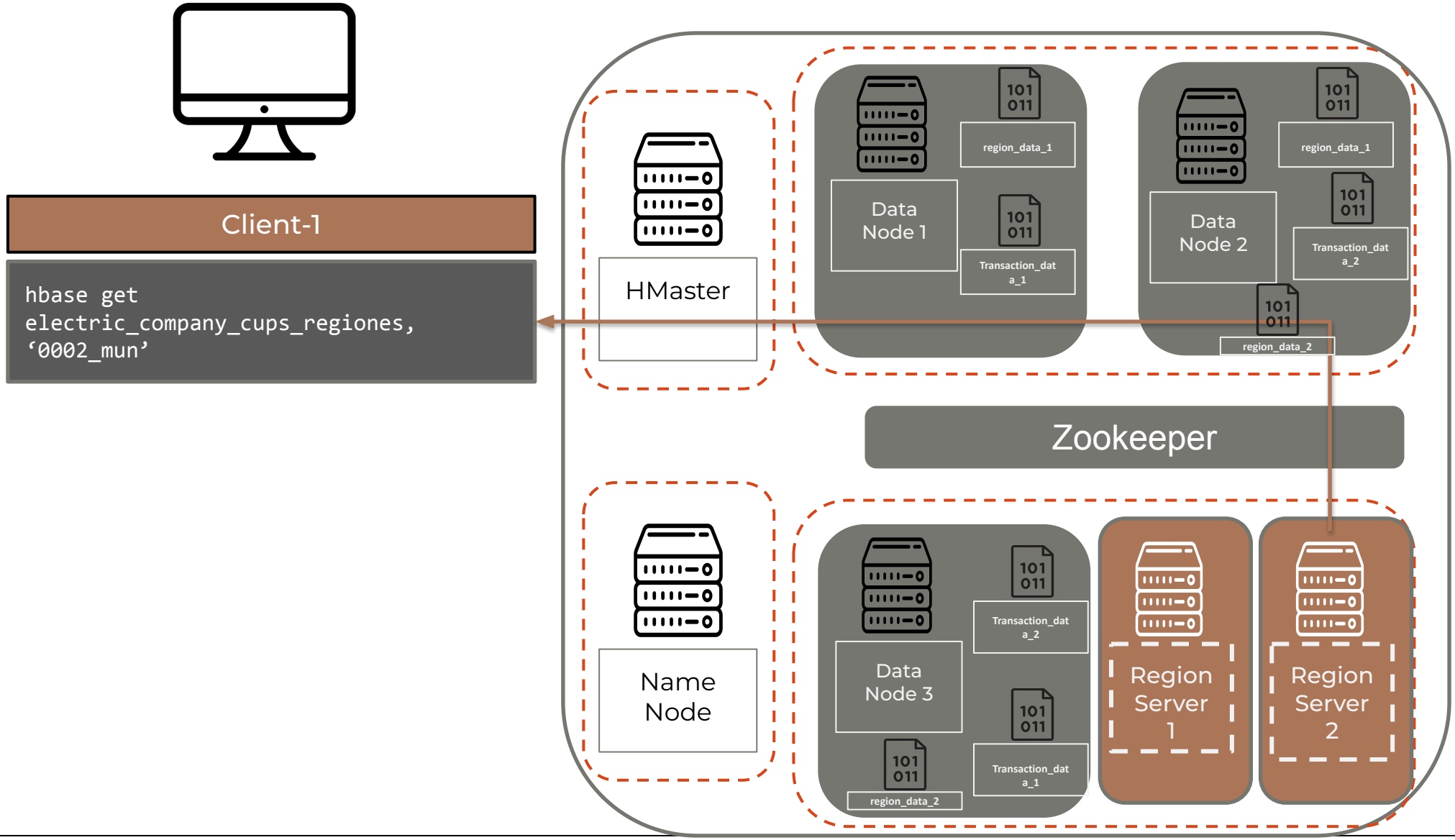
HBase Funcionamiento



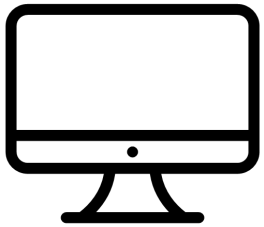
HBase Funcionamiento



HBase Funcionamiento

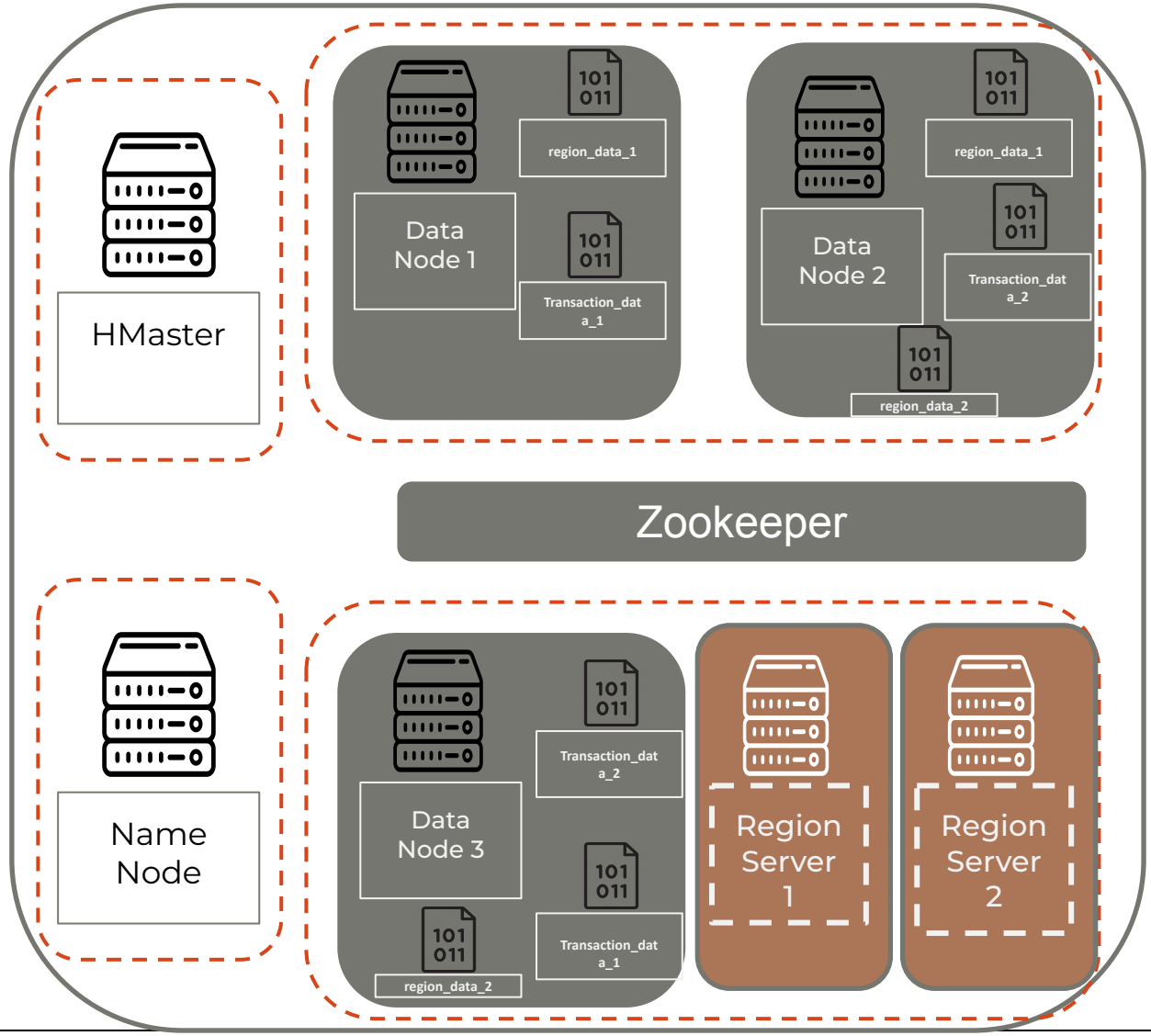


HBase Funcionamiento

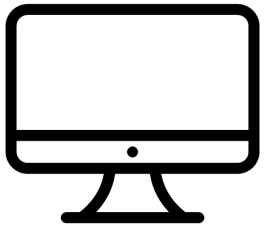


Client-1

```
hbase scan
electric_company_cups_regiones
```

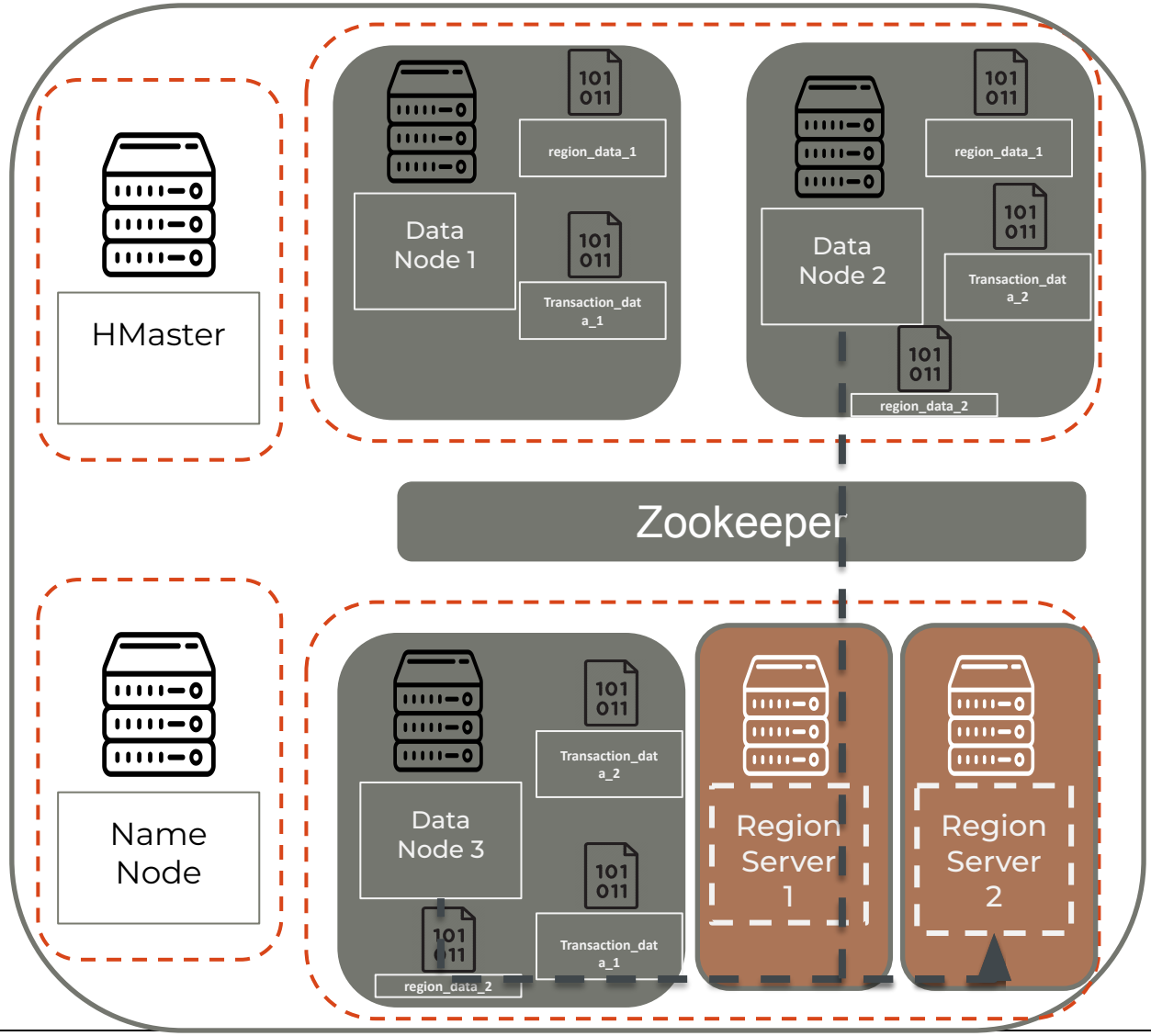


HBase Funcionamiento

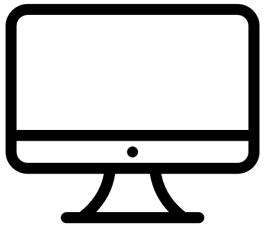


Client-1

```
hbase scan
electric_company_cups_regiones
```

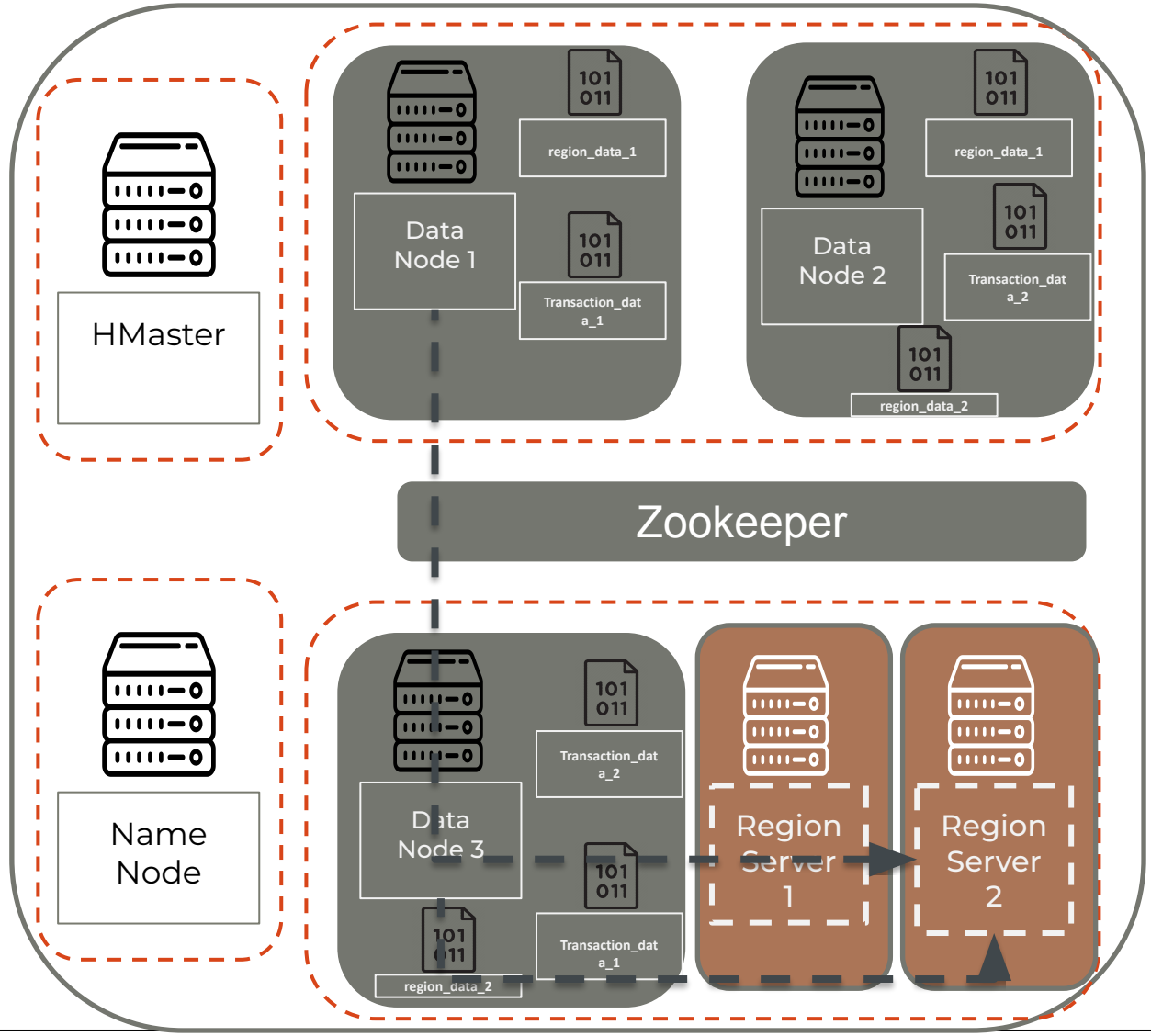


HBase Funcionamiento

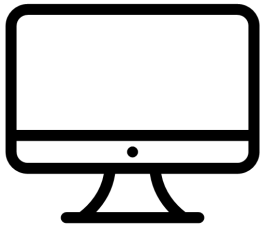


Client-1

```
hbase scan
electric_company_cups_regiones
```

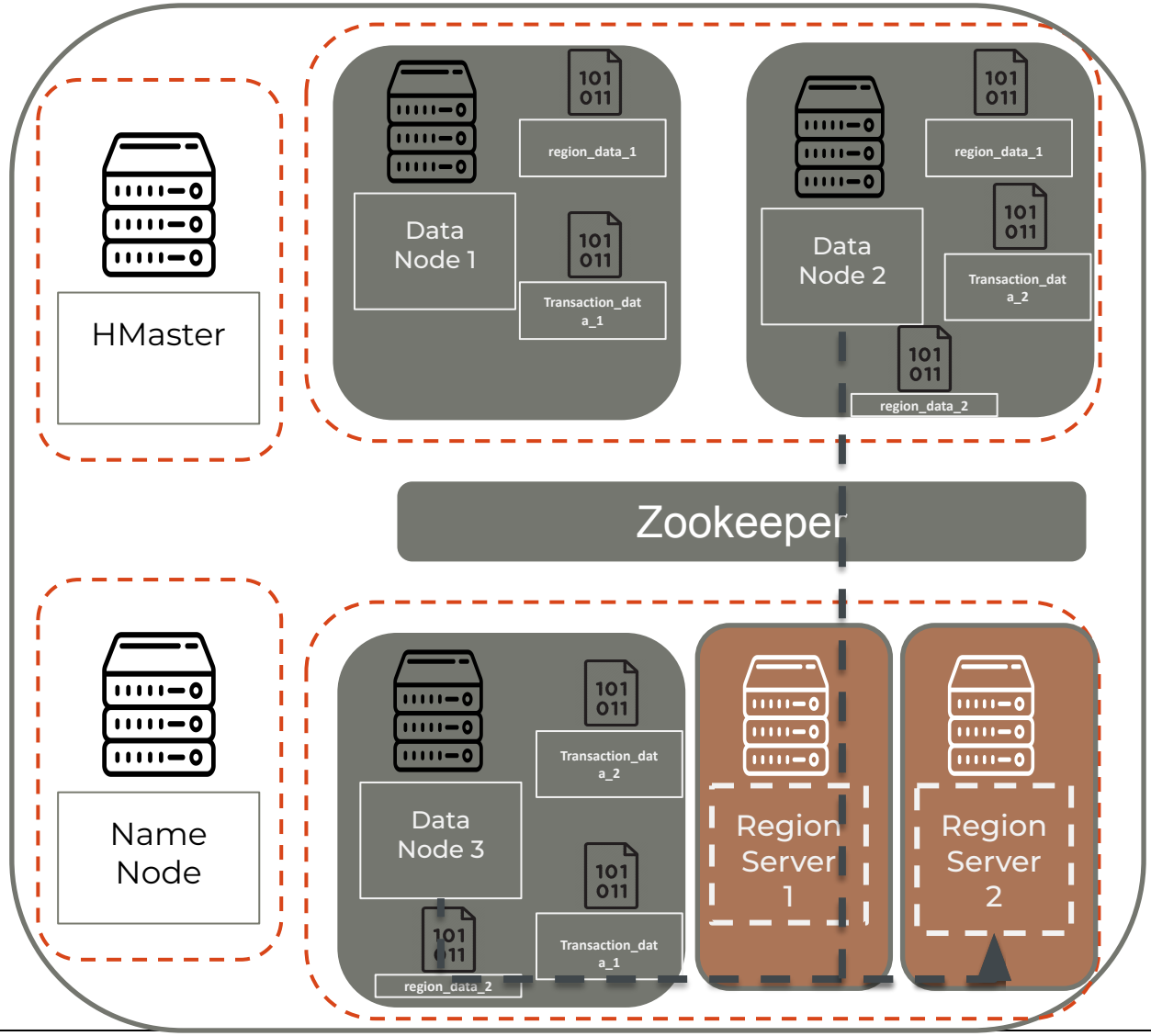


HBase Funcionamiento

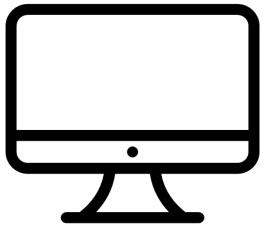


Client-1

```
hbase scan
electric_company_cups_regiones
```

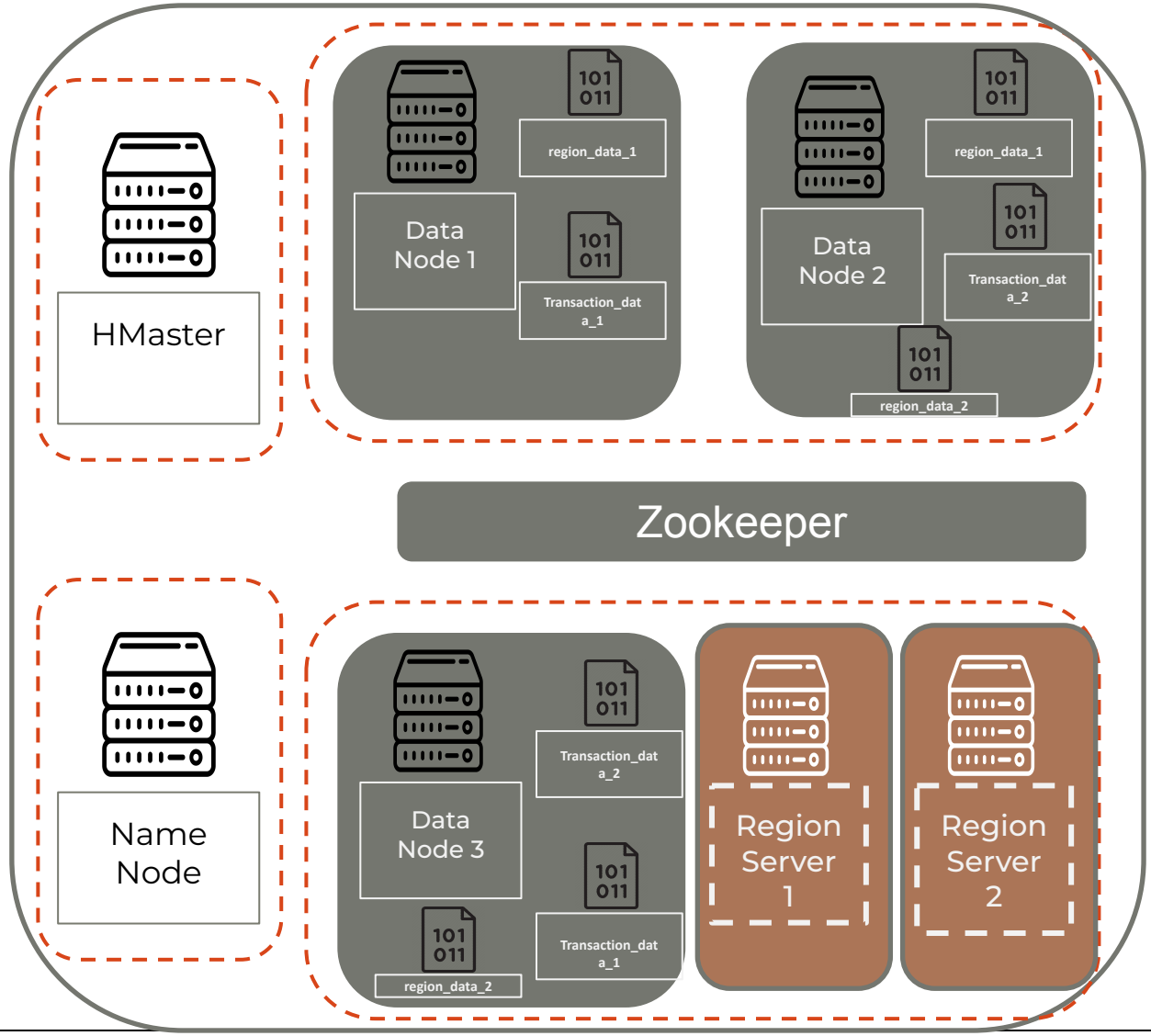


HBase Funcionamiento



Client-1

```
hbase scan
electric_company_cups_regiones

                ROW COLUMN+CELL
            '0001-municipio'
        region_data : valor_region
          timestamp=1631952266000,
          value='Colmenar Viejo'
            '0001-municipio'
        Tarificacion_data:pico
    timestamp=1631952500000, value=11.34
            '0001-municipio'
        Tarificacion_data:valle
    timestamp=1631954000000, value=100.34
            '0002-municipio'
        region_data : valor_region
          timestamp=1631952600000,
          value='Alcalá'
```



HBase - Introducción

- A raíz de las nuevas formas de almacenar se empiezan a pensar nuevas formas de consultar esa información
- El principal problema que pretende resolver es el acceso aleatorio a unos datos concretos
- Para ello usa HDFS como sistema de ficheros subyacente y una estrategia **Columnar**
- Con ello se consigue un acceso “rápido” a datos aleatorios en colecciones masivas de información
- Posteriormente nacen tecnologías de almacenamiento basadas en su misma filosofía supliendo algunas de sus carencias (Apache Cassandra)

Afi Escuela