



SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY !

Addis Ababa University
አዲስ አበባ ዩኒቨርሲቲ



Big Data Analytics and Visualization



By: Bezawit Mamo

Girma Moges

Mikias Zelalem

Contents

Introduction to HBase

Data Model

HBase Architecture

HBase read or write

Hbase vs RDMS

Features of HBase

HBase Shell Commands

Introduction to HBase

Back in days, data used be less & was mostly Structured



Structured data



This data could be easily stored in a Relational Database (RDBMS)

Then, internet involved and huge volume of structured and semi-structured data got generated



Semi-structured data



Storing and processing this data on RDBMS became a major problem

The Solution to this NoSQL DB
E.g. HBase, MangosDS, Casandra...

Solution



Semi-structured data



Apache HBASE was the solution for this

What is HBase?



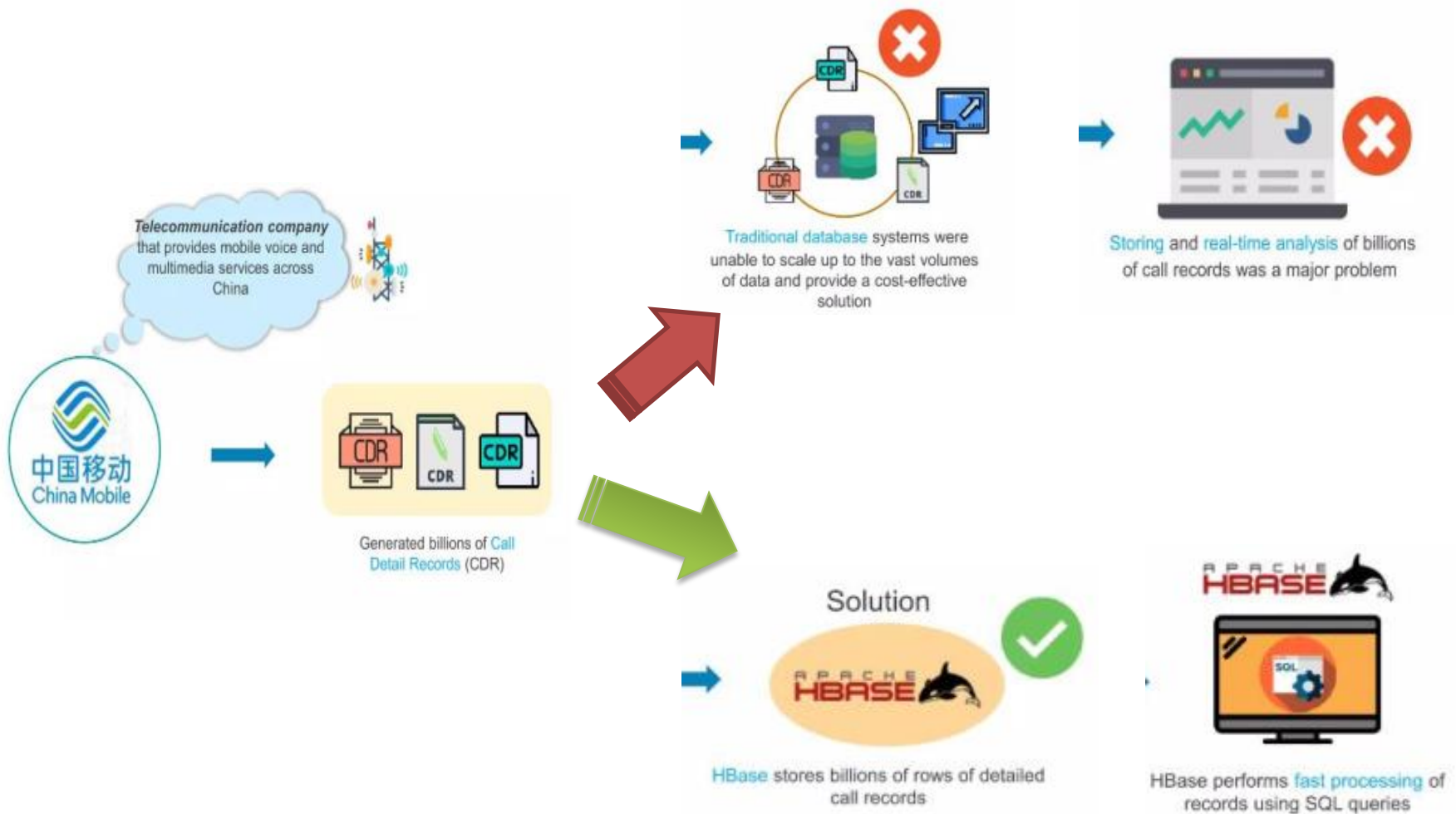
HBase is a column oriented database management system derived from Google NoSQL database [BigTable](#) that runs on top of HDFS

Open source project that is horizontally scalable

It can store huge amount of data in tabular format for extremely fast read writes

Well suited for sparse data set (can contain missing or NA values)

HBase: Real Life Connect

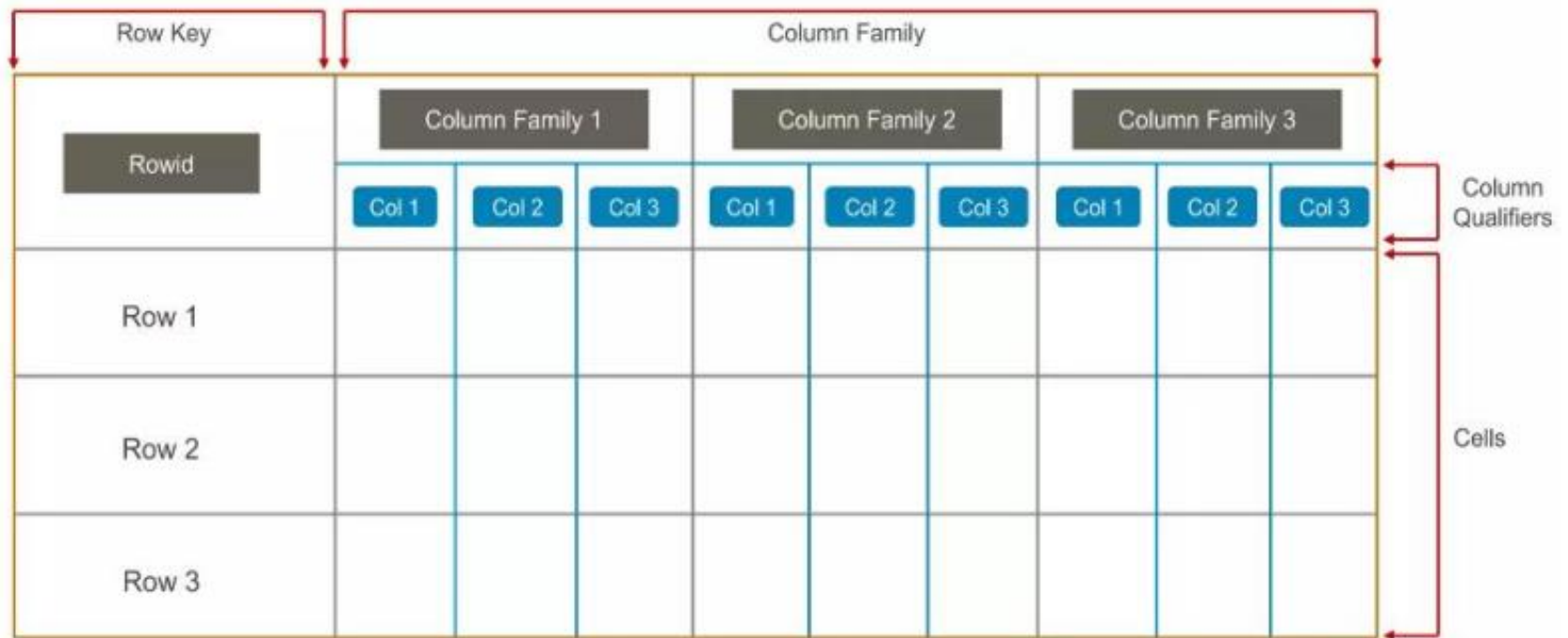


HBase: Real Life Connect

Companies using HBase



Data Model



Data Model: Example

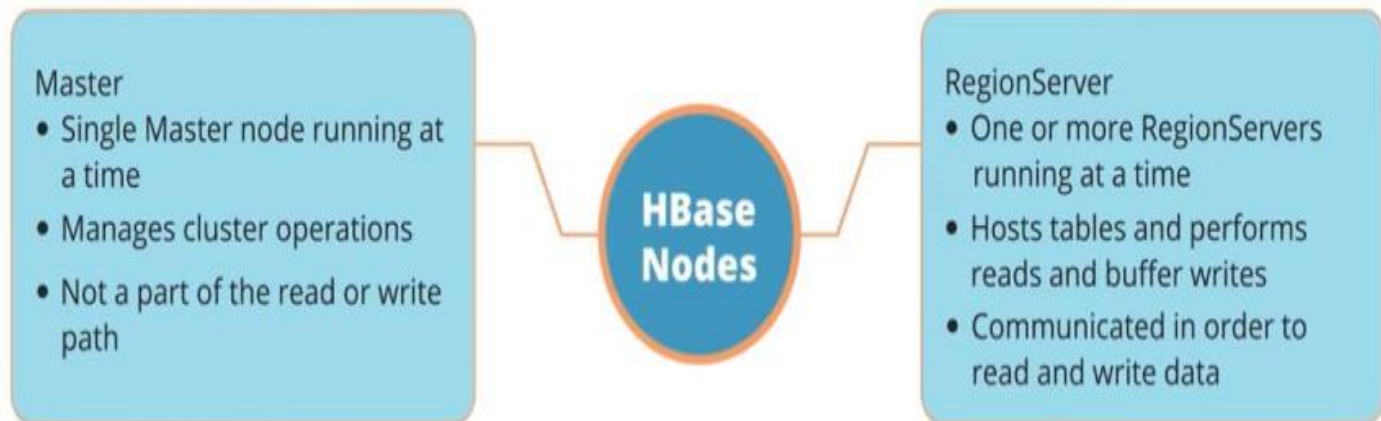
The diagram illustrates a data model table with the following structure and annotations:

- Row Key:** Indicated by a bracket above the first column, which contains the `Rowid` header and `empid` values.
- Column Family:** Indicated by a bracket above the remaining columns, which are grouped into `Personal data` (name, city, age) and `Professional data` (designation, salary).
- Column Qualifiers:** Indicated by arrows pointing to the individual column headers (`name`, `city`, `age`, `designation`, `salary`).
- Cells:** Indicated by a bracket on the right side, pointing to the data rows.

Rowid	Personal data			Professional data	
empid	name	city	age	designation	salary
1	Angela	Chicago	31	Big Data Architect	\$70,000
2	Dwayne	Boston	35	Web Developer	\$65,000
3	David	Seattle	29	Data Analyst	\$55,000

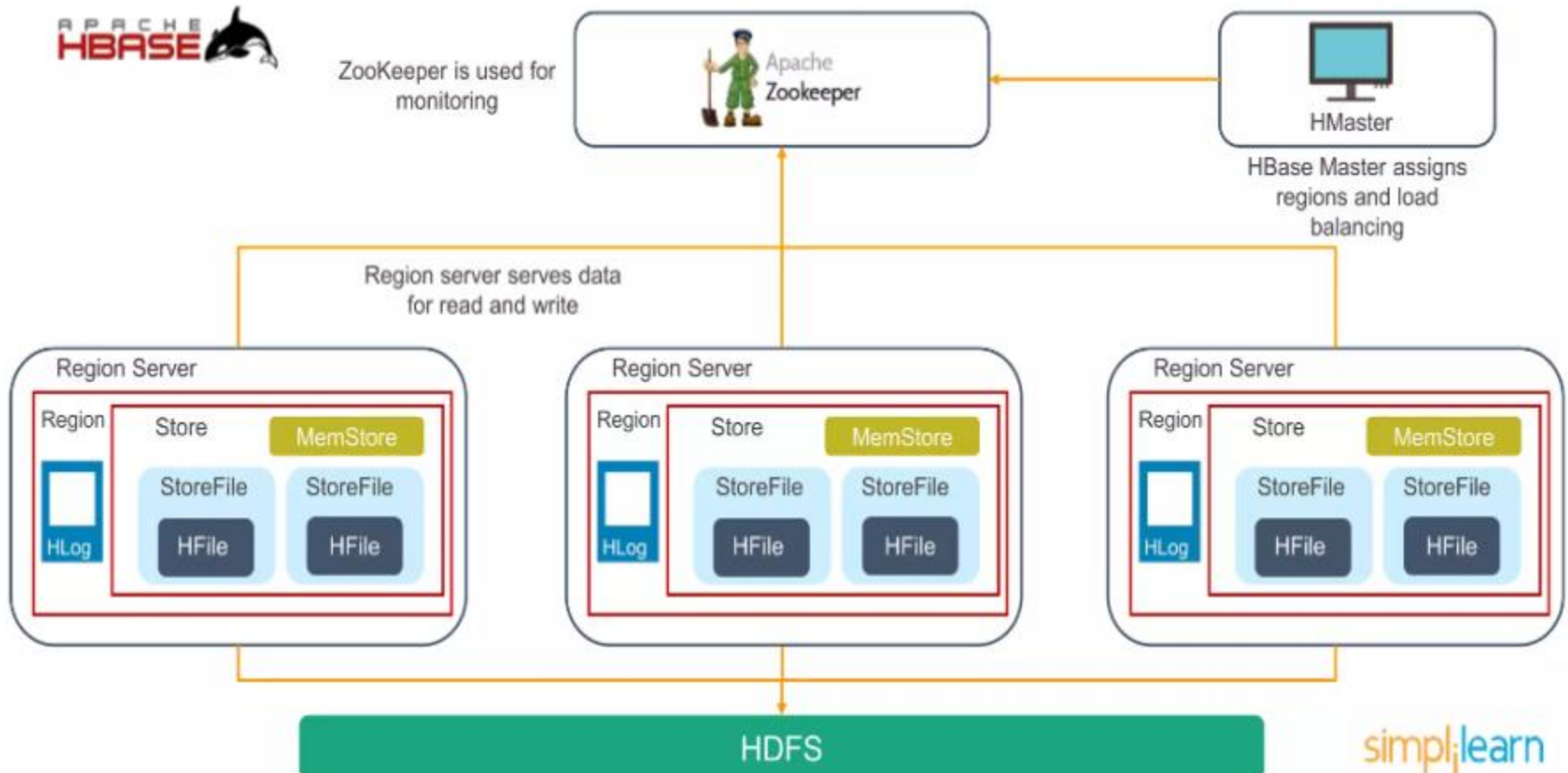
HBase Architecture

Hbase has two type of nodes : **Master** and **Regional Server**



A region in HBase is subset of table's rows. The **Master** node detects the status of **RegionServers** and assign a region to it.

HBase Architecture



HBase Architecture components: Regions



HBase tables are divided horizontally by row key range into "**Regions**"

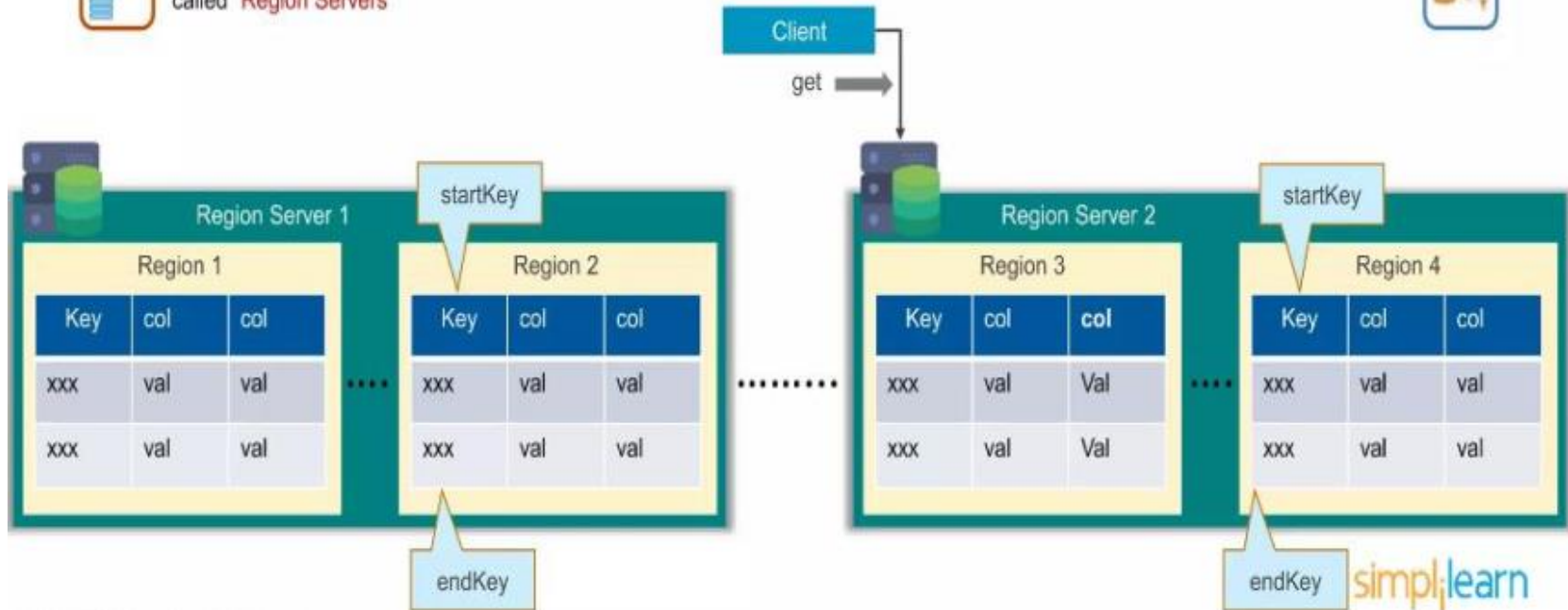


Regions are assigned to the nodes in the cluster, called "**Region Servers**"

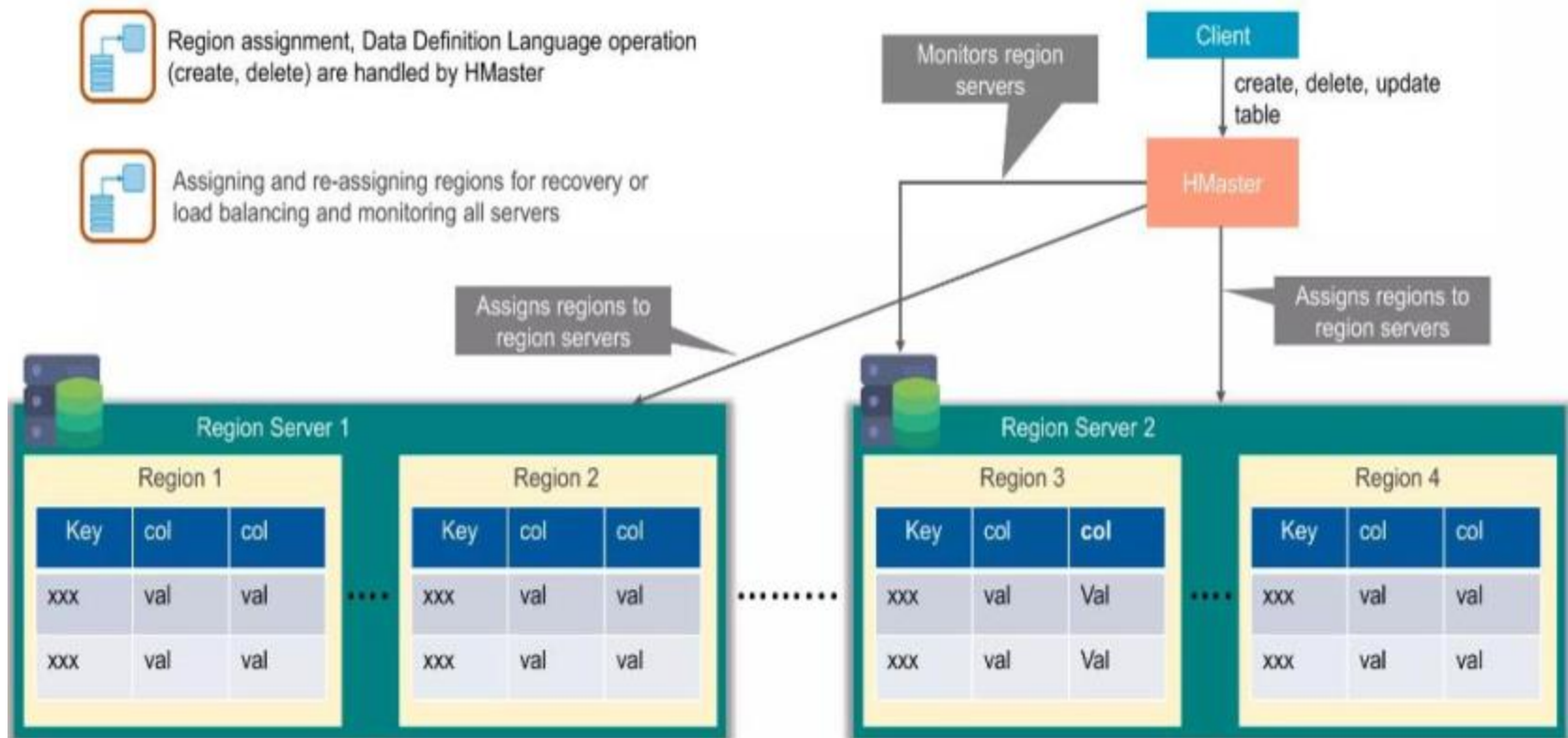
A region contains all rows in the table between the region's **start key** and **end key**



These servers serve data for **read** and **write**

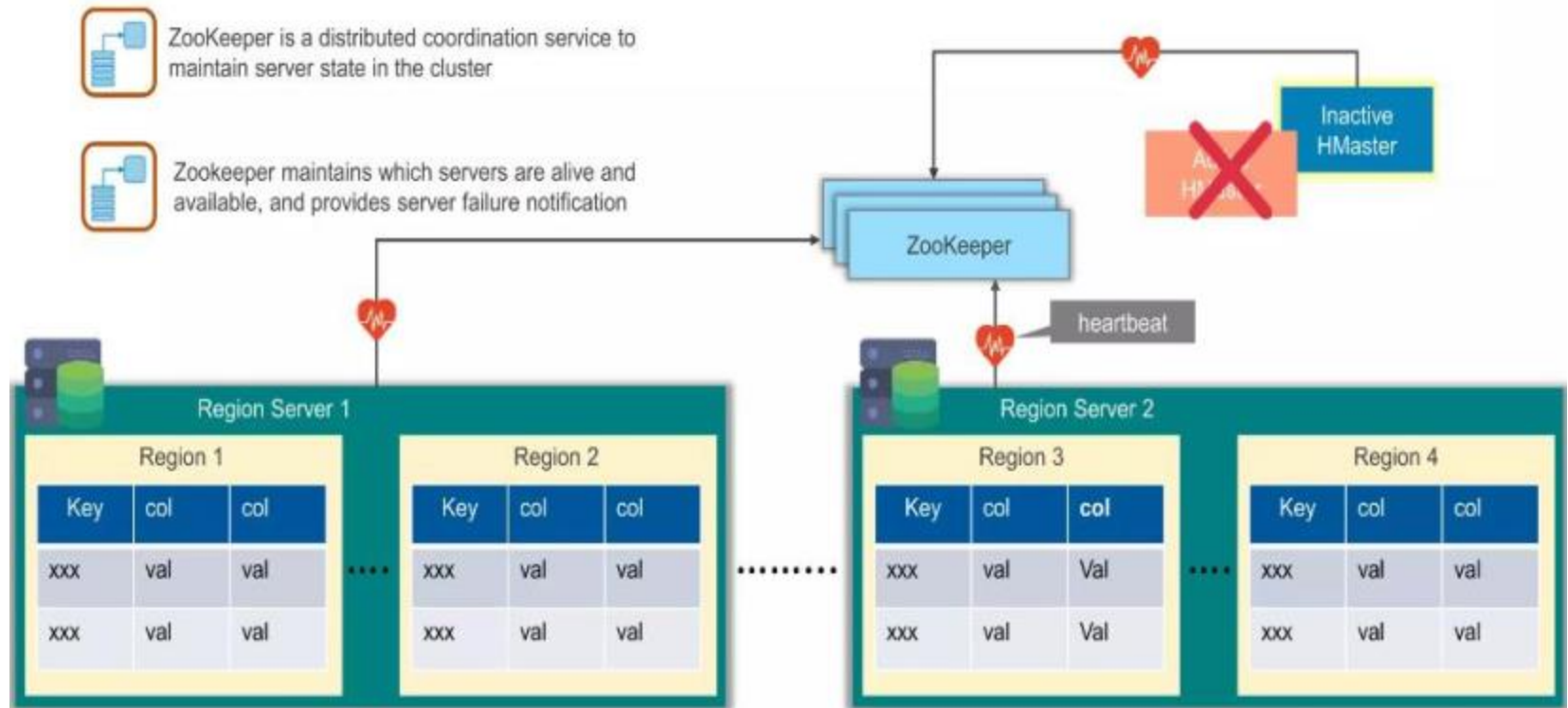


HBase Architecture components: HMaster



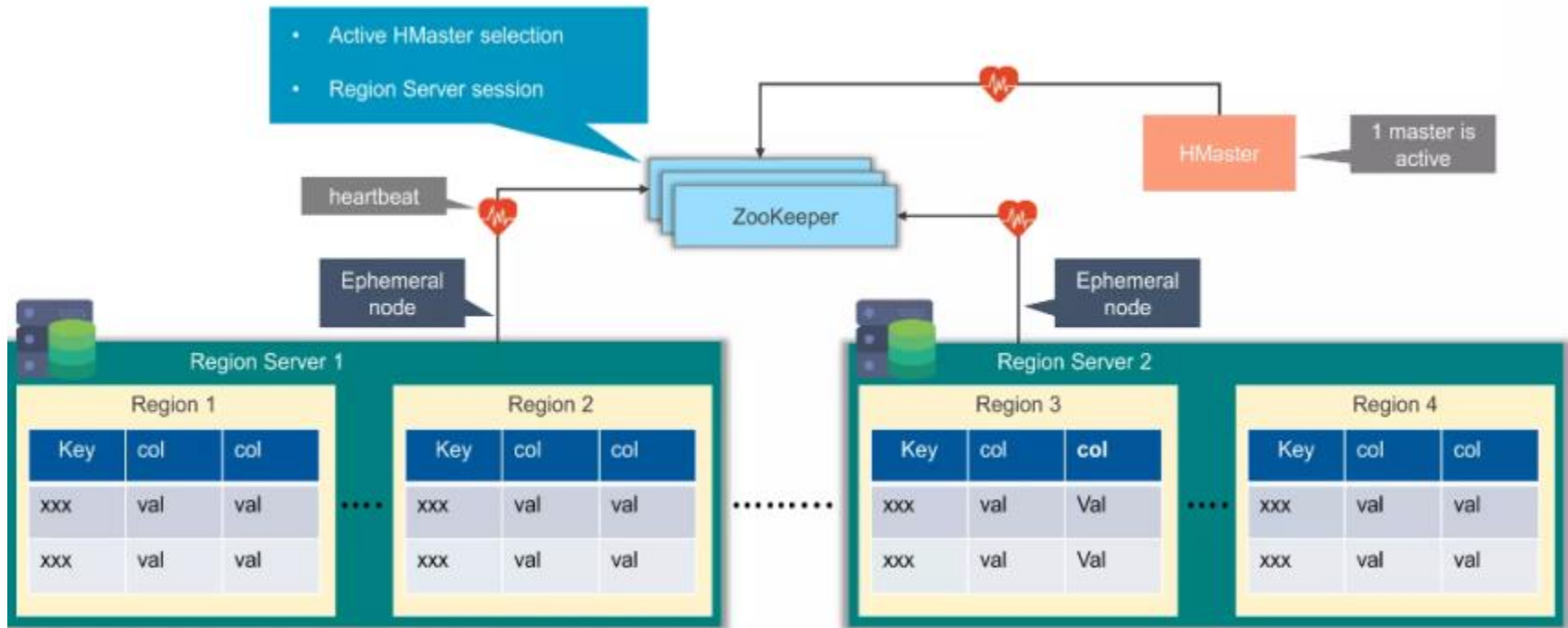
Hbase has a distributed enviroment where **Hmaster** alone is not sufficient to manage everything. Hence, **Zookeeper** was introduced.

HBase Architecture components: Zookeeper



- **Active Hmaster** send a heartbeat signal to Zookeeper indicating its active .
- **Inactive servers** act as a backup. If the active Hmaster fails, it will come to rescuer.
- **Region servers** send their status to zookeeper indicating they are ready for read and write operation

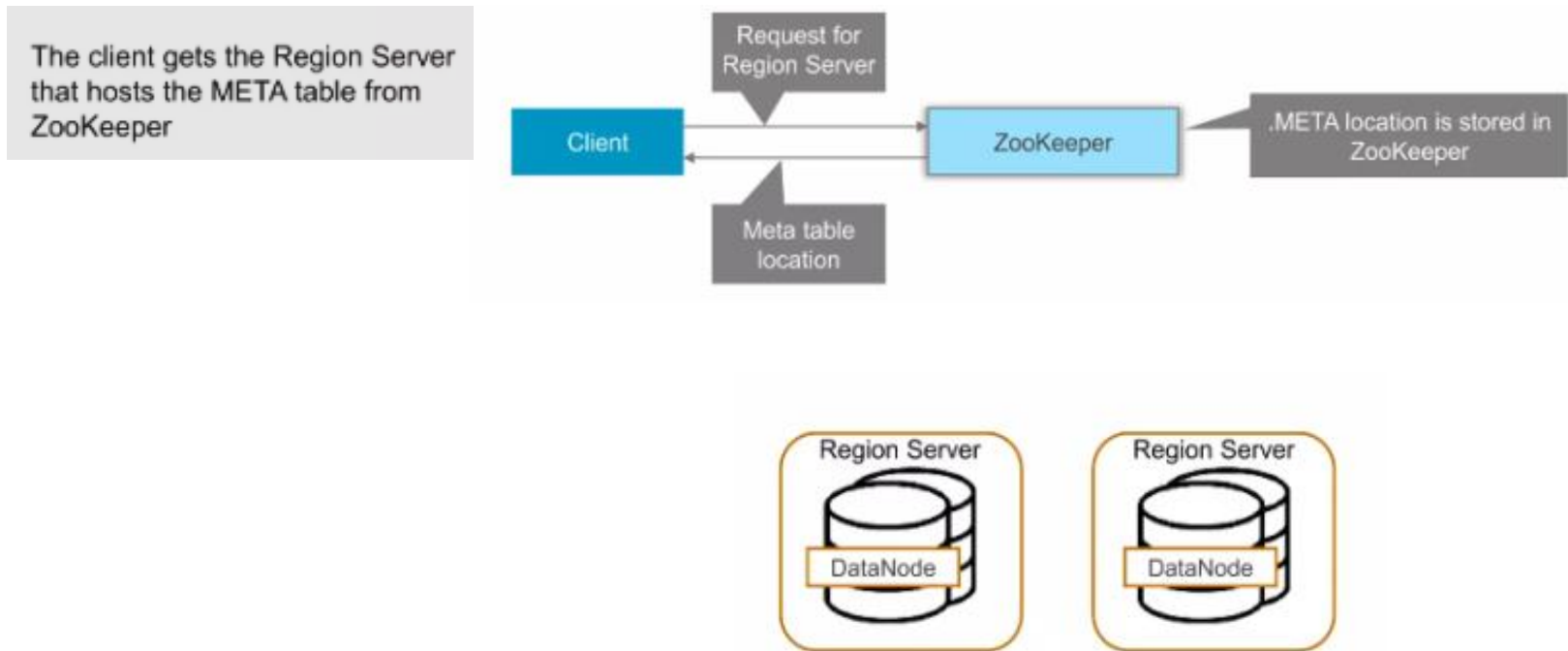
How the components work together?



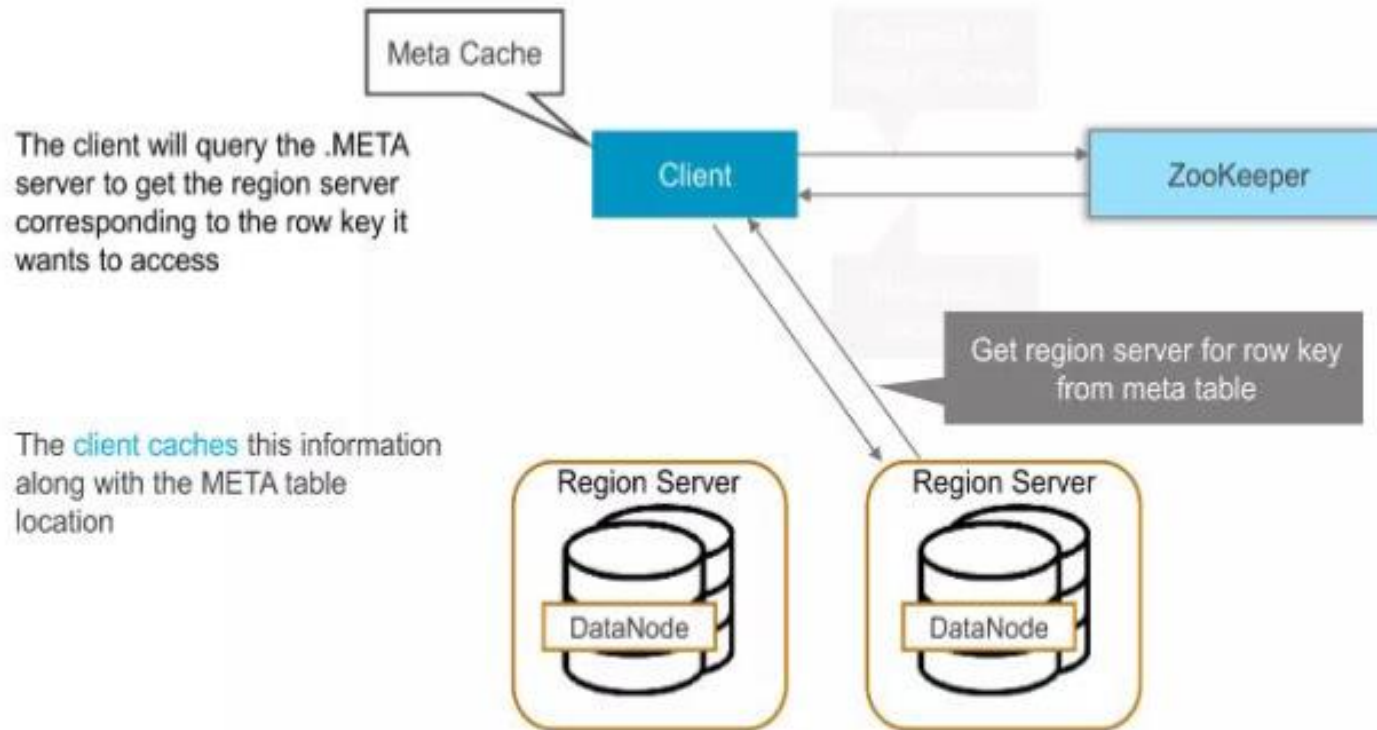
1. Active Hmaster and Region Servers connect with a session to Zookeeper.
2. Zookeeper ephemeral nodes for active sessions via heartbeat to indicate that region servers are up and running

HBase Read or Write

- There is a special Hbase catalog table called **META table**, which holds the location of regions in the cluster.
- **Here is what happened the first time a client read or write data to Hbase**

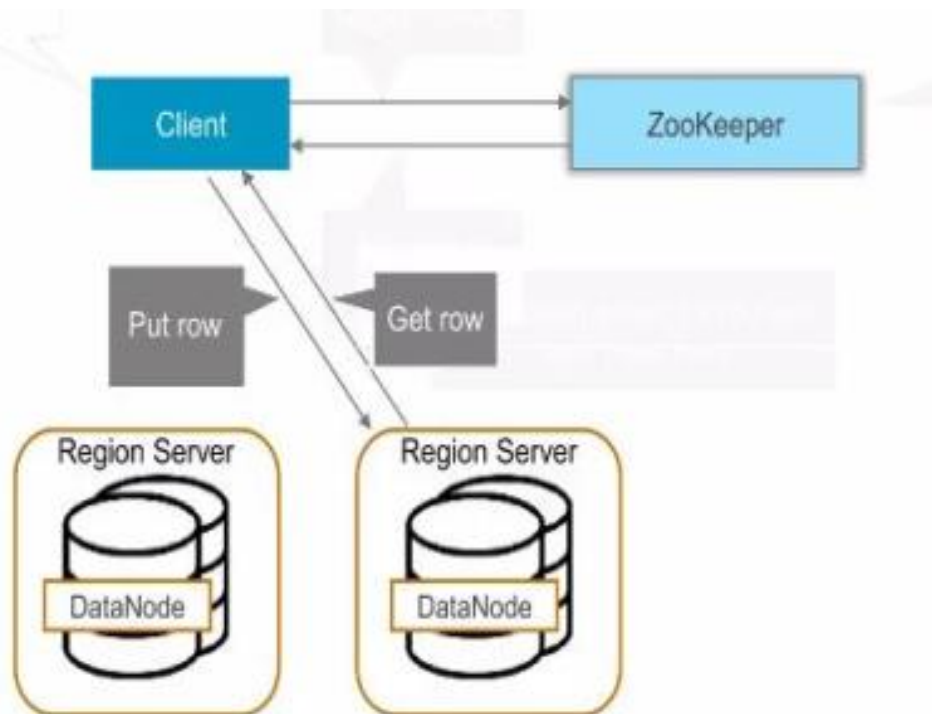


HBase Read or Write



HBase Read or Write

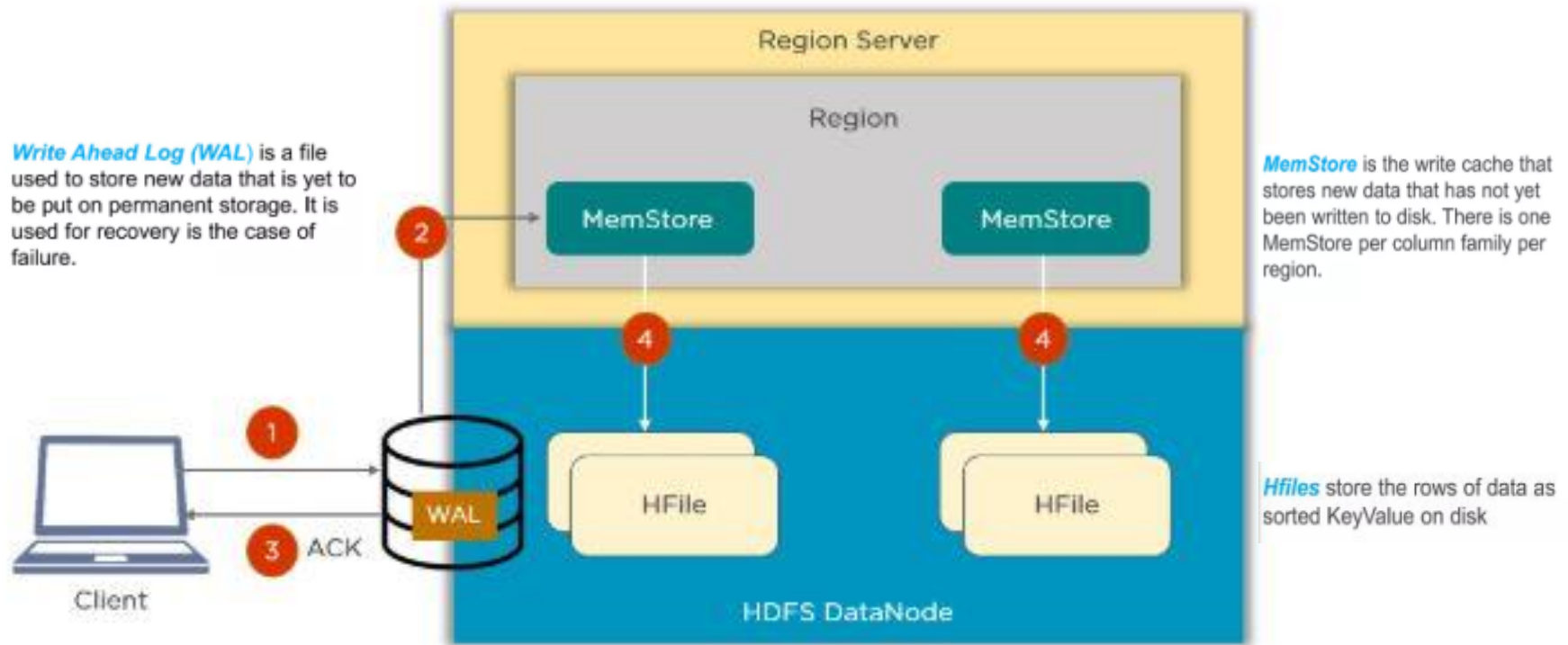
It will get the Row from the corresponding Region Server



HBase Meta Table








HBase Write Mechanism



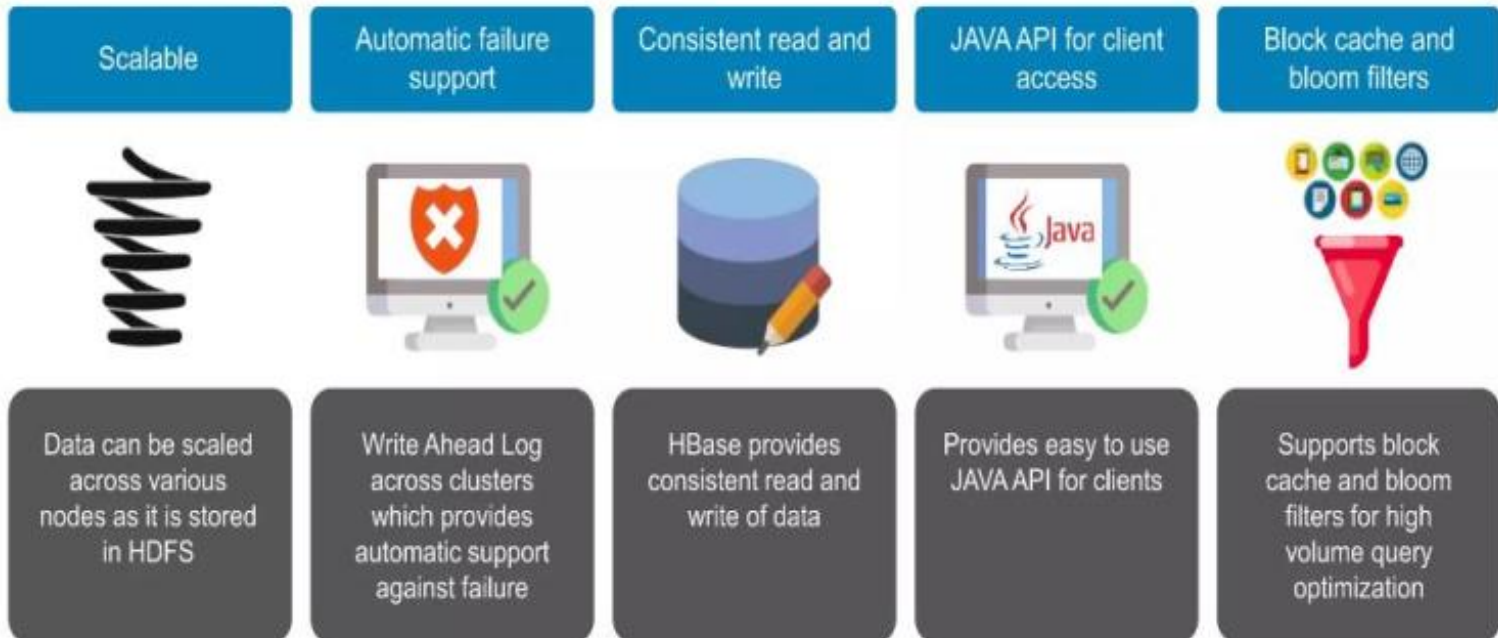
- 1 When client issues a **put** request, it will write the data to the write-ahead log (WAL)
- 2 Once data is written to the WAL, it is then copied to the **MemStore**
- 3 Once the data is placed in MemStore, the client then receives the **acknowledgment**
- 4 When the MemStore reaches the threshold, it dumps or commits the data into a **HFile**

HBase vs RDBMS

The table shows a comparison between HBase and a Relational Database Management System (RDBMS):

HBase		RDBMS
Automatic partitioning		Usually manual and admin-driven partitioning
Scales linearly and automatically with new nodes		Usually scales vertically by adding more hardware resources
Uses commodity hardware		Relies on expensive servers
Has fault tolerance		Fault tolerance may or may not be present
Leverages batch processing with MapReduce distributed processing		Relies on multiple threads or processes rather than MapReduce distributed processing

Features of HBase



HBase Shell Commands

Common commands include, but are not limited to, the following:

Create table. Pass table name from a dictionary of specifications per column family, and a dictionary of table configuration which is optional.

HBase> create 't1', {NAME => 'f1'}, {NAME => 'f2'}, {NAME => 'f3'} HBase> #
The above in shorthand would be the following: HBase> create 't1', 'f1', 'f2', 'f3'

Describe the table named.

HBase> describe 't1'

Start the disabling of the table named.

HBase> disable 't1'

Drop the table named. Table must first be disabled.

HBase> drop 't1'

List all tables in HBase. Optional regular expression parameter can be used to filter the output.

HBase> list

HBase Shell Commands



Count

Counting the number of rows in a table

Delete
Deleting a cell value



Get

Getting the contents of a row or a cell

Put
Putting a cell value



Scan

Scanning a table's values

For more: Reading materials

- https://hbase.apache.org/apache_hbase_reference_guide.pdf
- <https://www.simplilearn.com/tutorials/hadoop-tutorial/hbase>



Thank you!