CSC 369: Distributed Computing

Alex Dekhtyar

Day 3: Distributed DBMS in a nutshell


13-109
김치양념 맛있게

Housekeeping

- State of the WaitList
- Roll call?

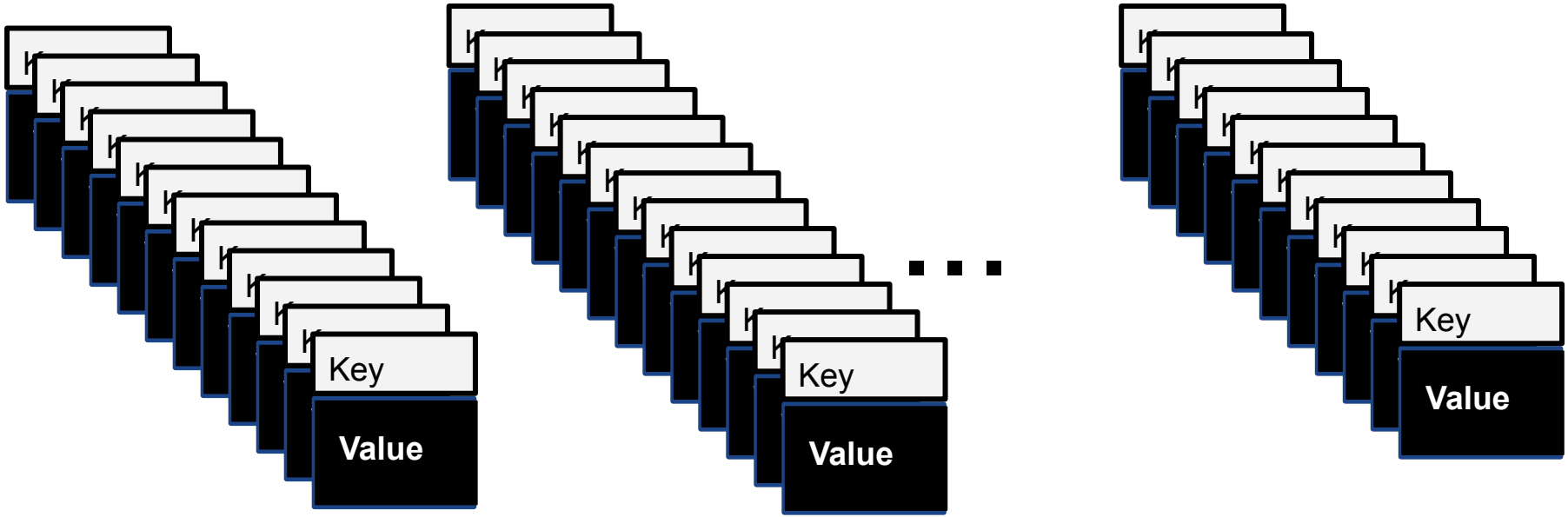
Previously on CSC 369

The “Facebook” Example:

How do they do it?

With Key-Value Stores!

```
SELECT * FROM UserProfiles
WHERE UserID = “bob@yahoo.com
```



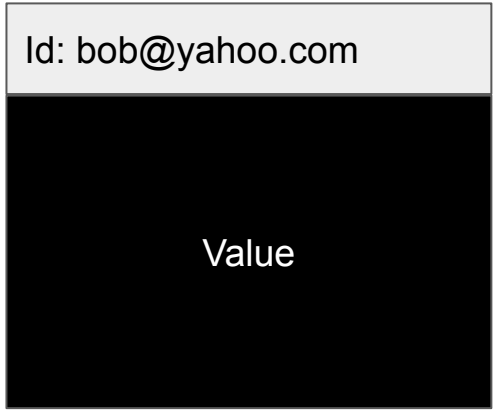
Trade-offs

Relational Databases

Id	bob@yahoo.com
Name	Bob Smith
Status	single
Avatar	bob.jpg
...	...

All data exposed
Powerful queries
Slow

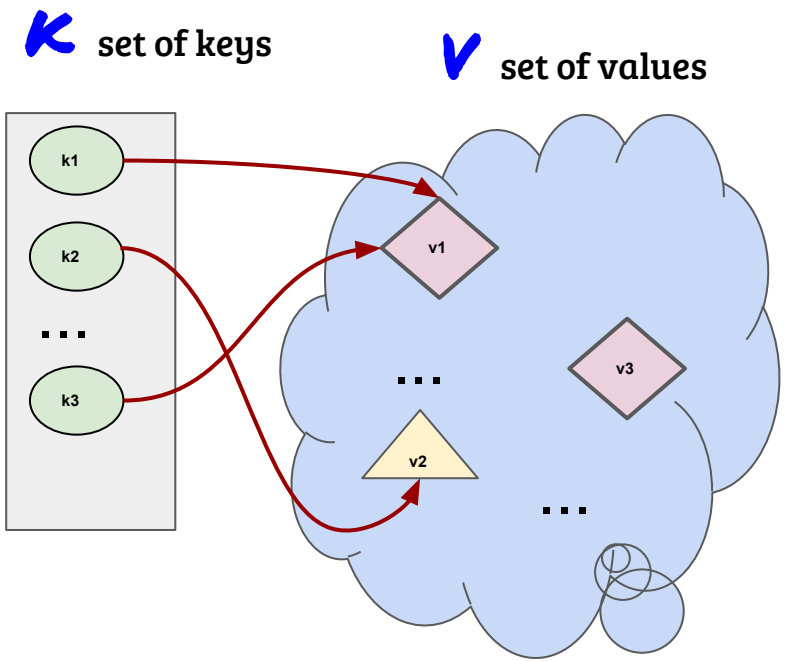
Key Value Stores



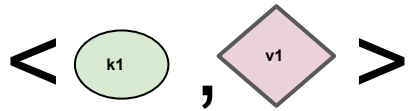
Value is obscured
Only **get(key)**
Lightning fast
Easier to Distribute

Maps

A **map** is another word for **"function"** over a finite domain



Key Value Pair

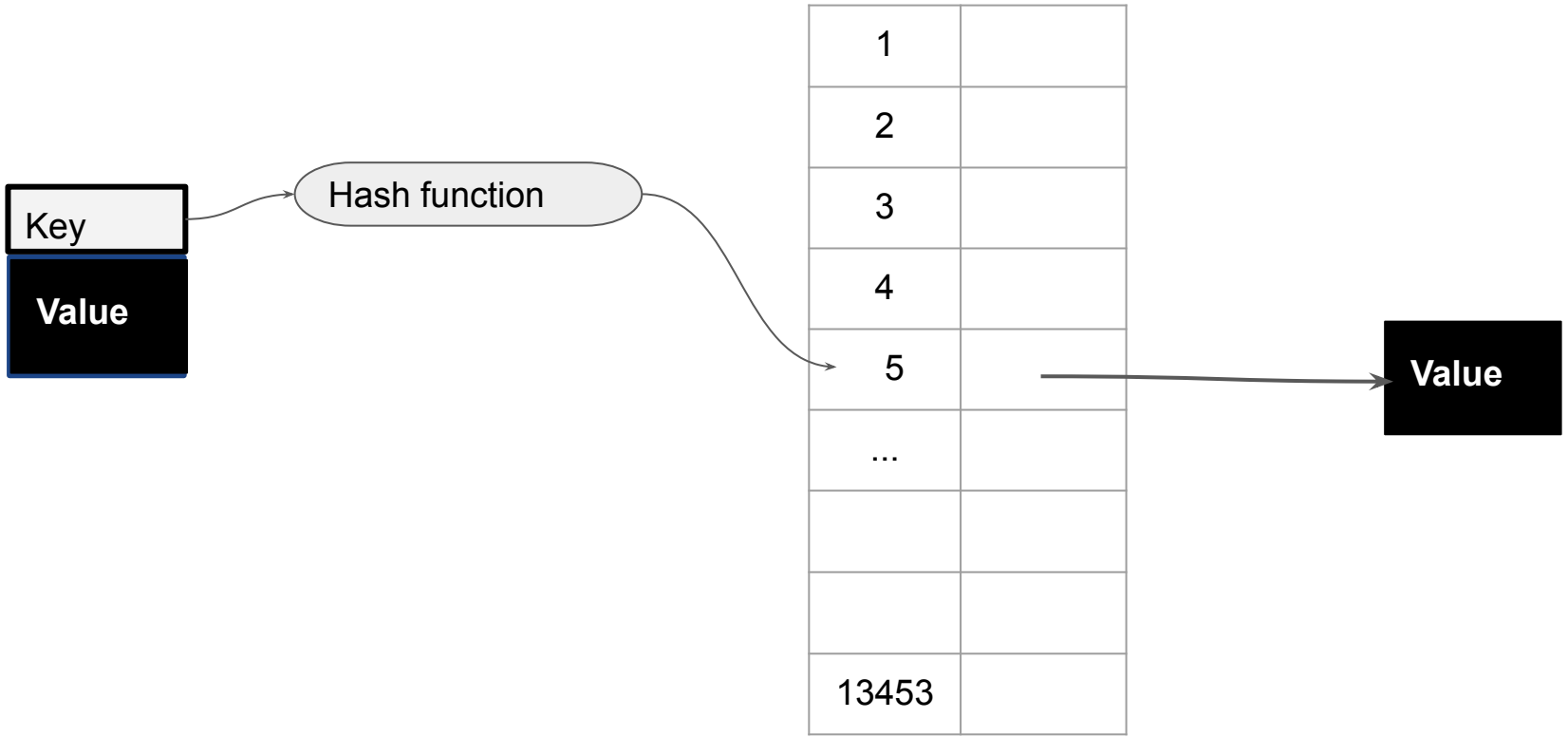


A map is a set of key-value pairs

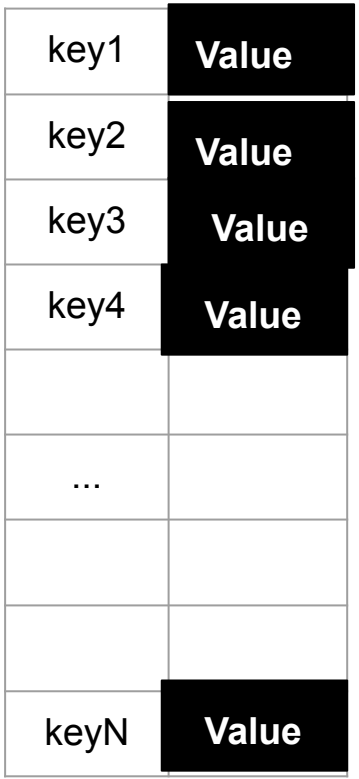
$$\{<k,v> \mid k \text{ in } K, v \text{ in } V\}$$

$$\{<k,M(k)> \mid k \text{ in } K\}$$

Key-Value Store/Map by Hashing

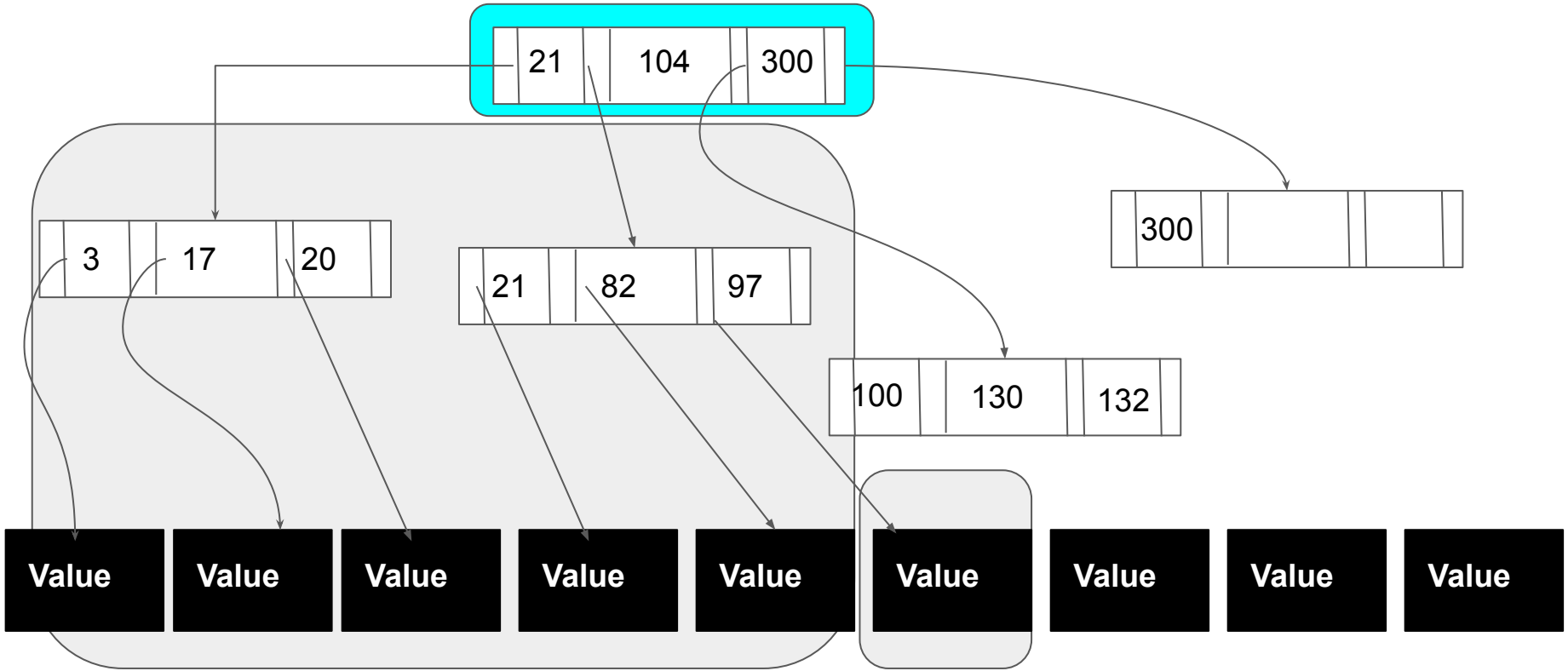


Key-Value Store by By Sorting



Key-Value Store by Via Trees

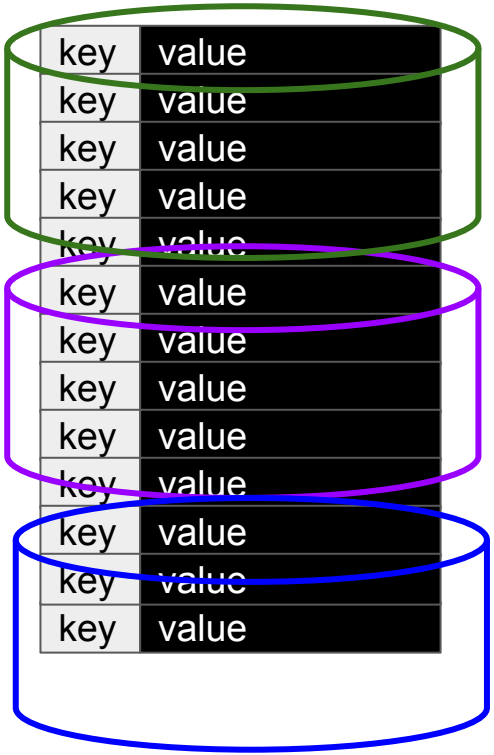
B+trees / **Red**-Black Trees



Distributed Databases In a nutshell

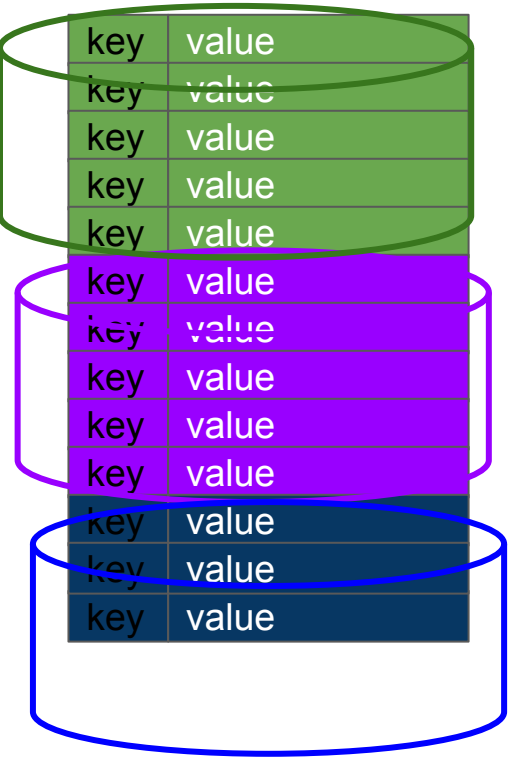
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value
key	value

Distributed Databases In a nutshell



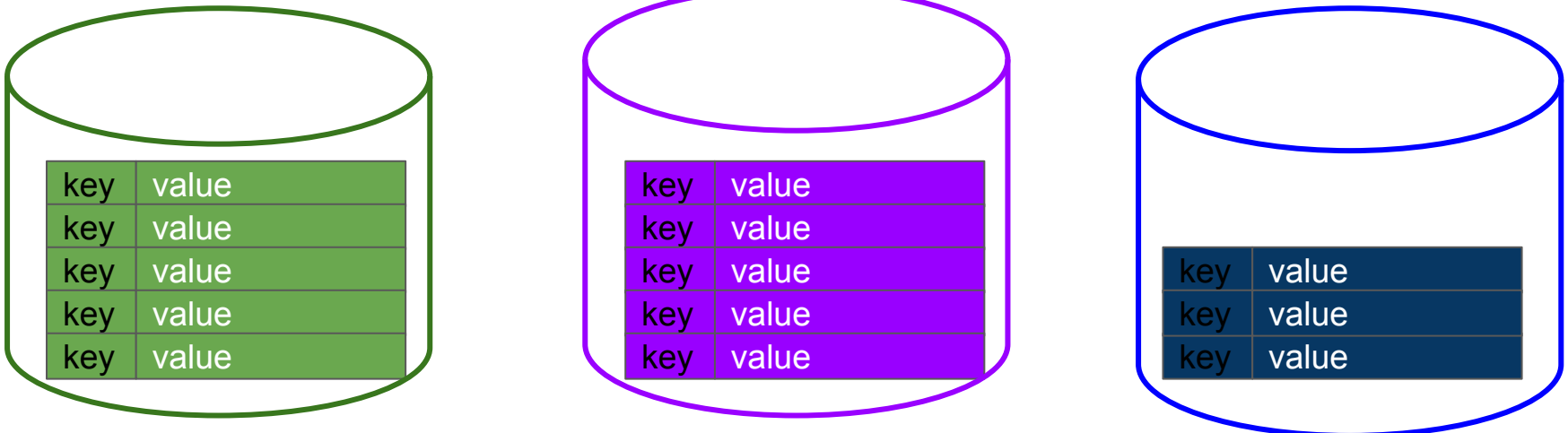
Distributed Databases In a nutshell

Sharding (Partitioning)



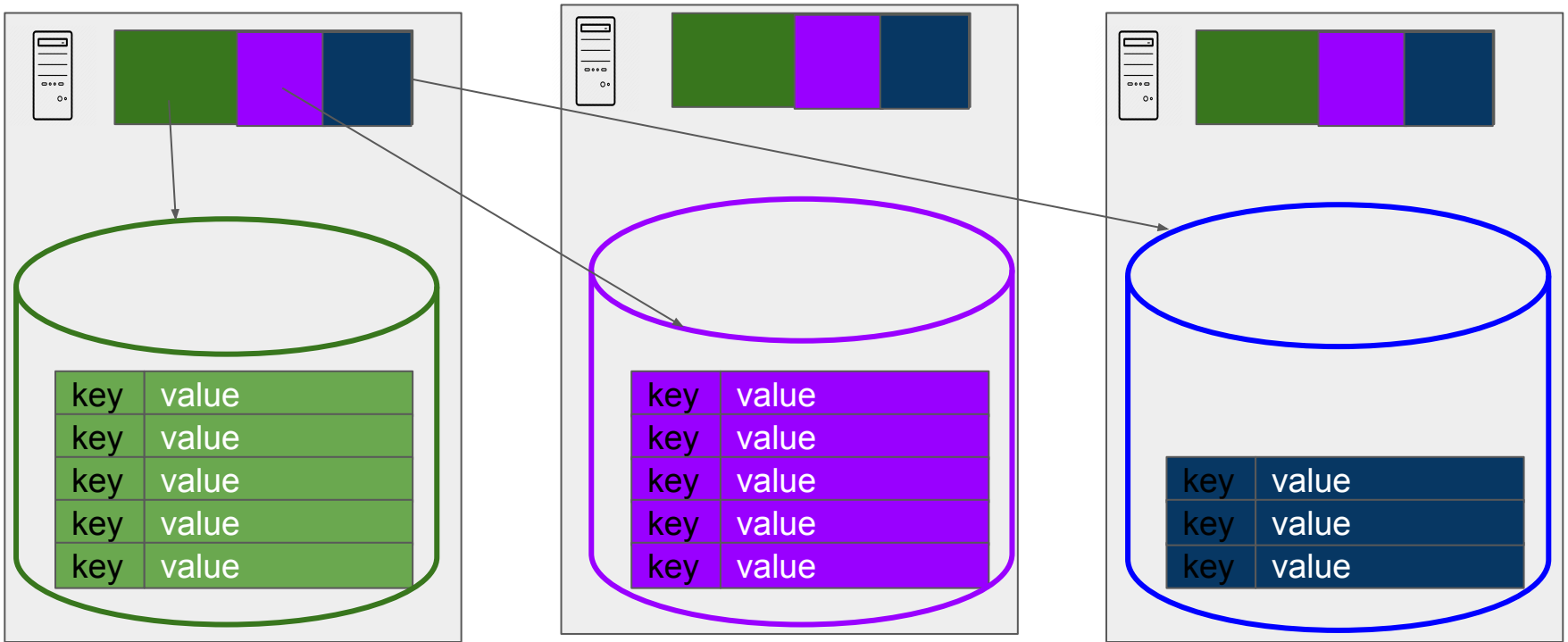
Distributed Databases In a nutshell

Sharding (Partitioning)



Distributed Databases In a nutshell

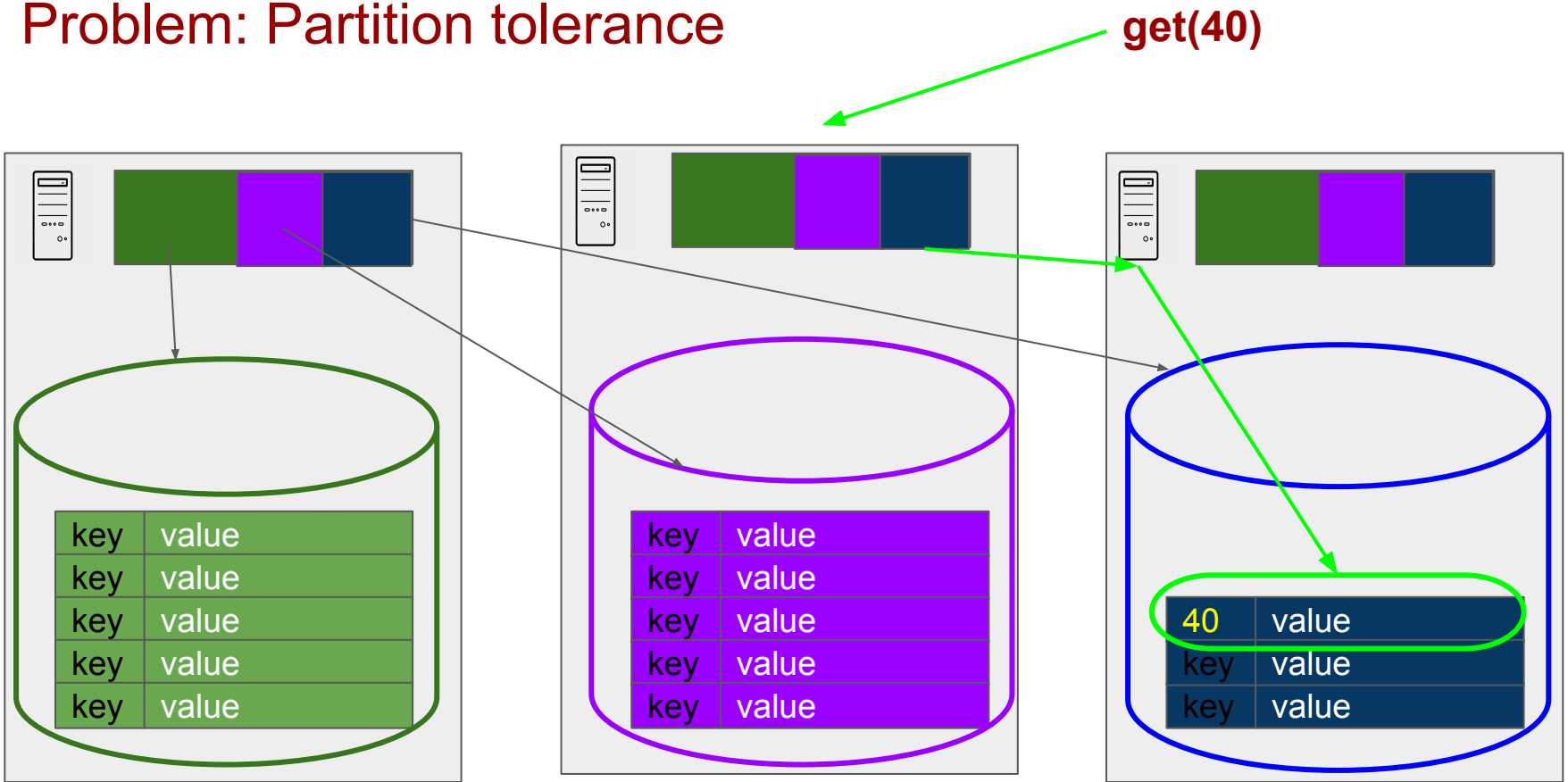
Sharding (Partitioning)



Distributed Databases In a nutshell

Sharding (Partitioning)

Problem: Partition tolerance

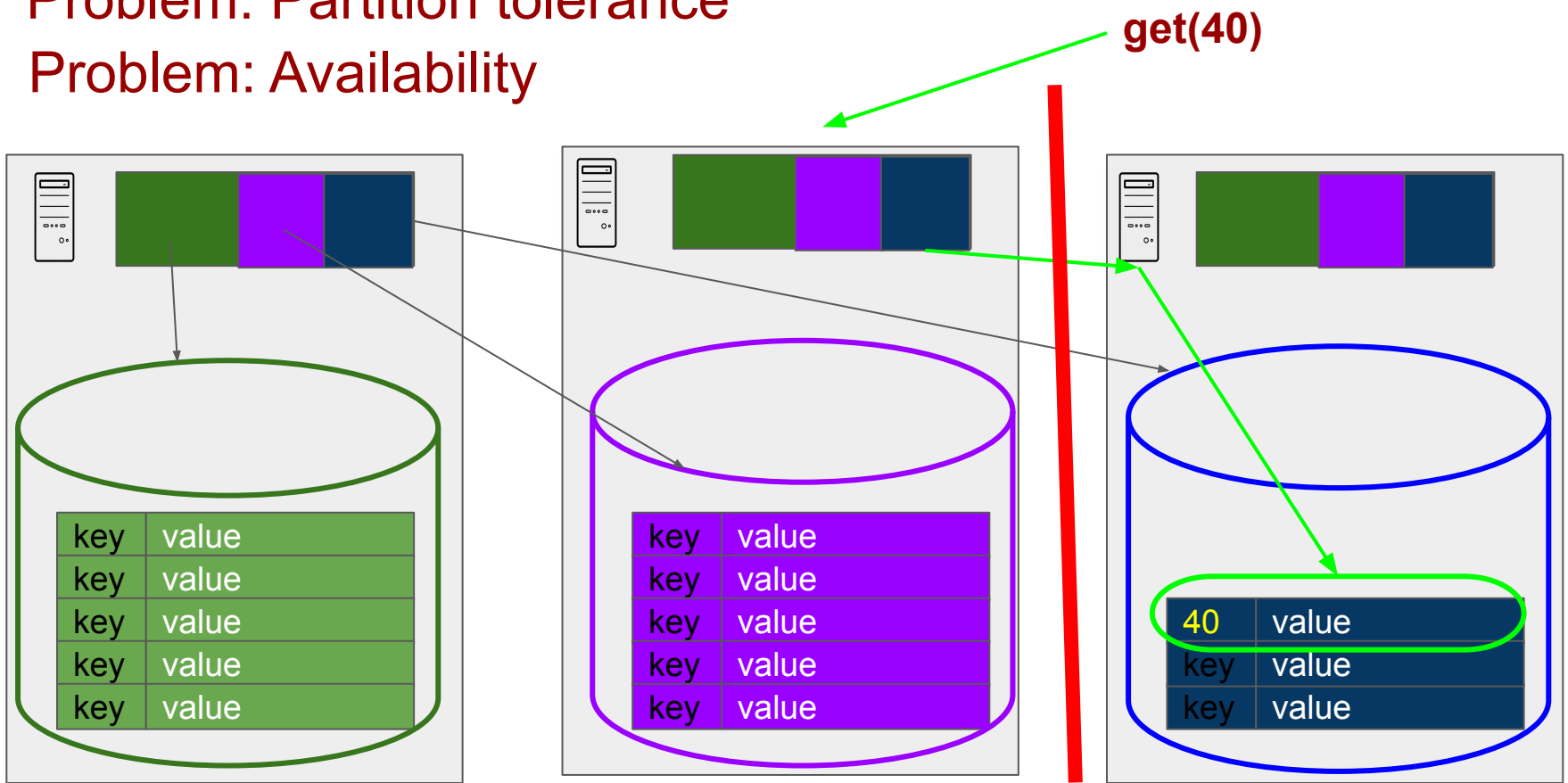


Distributed Databases In a nutshell

Sharding (Partitioning)

Problem: Partition tolerance

Problem: Availability

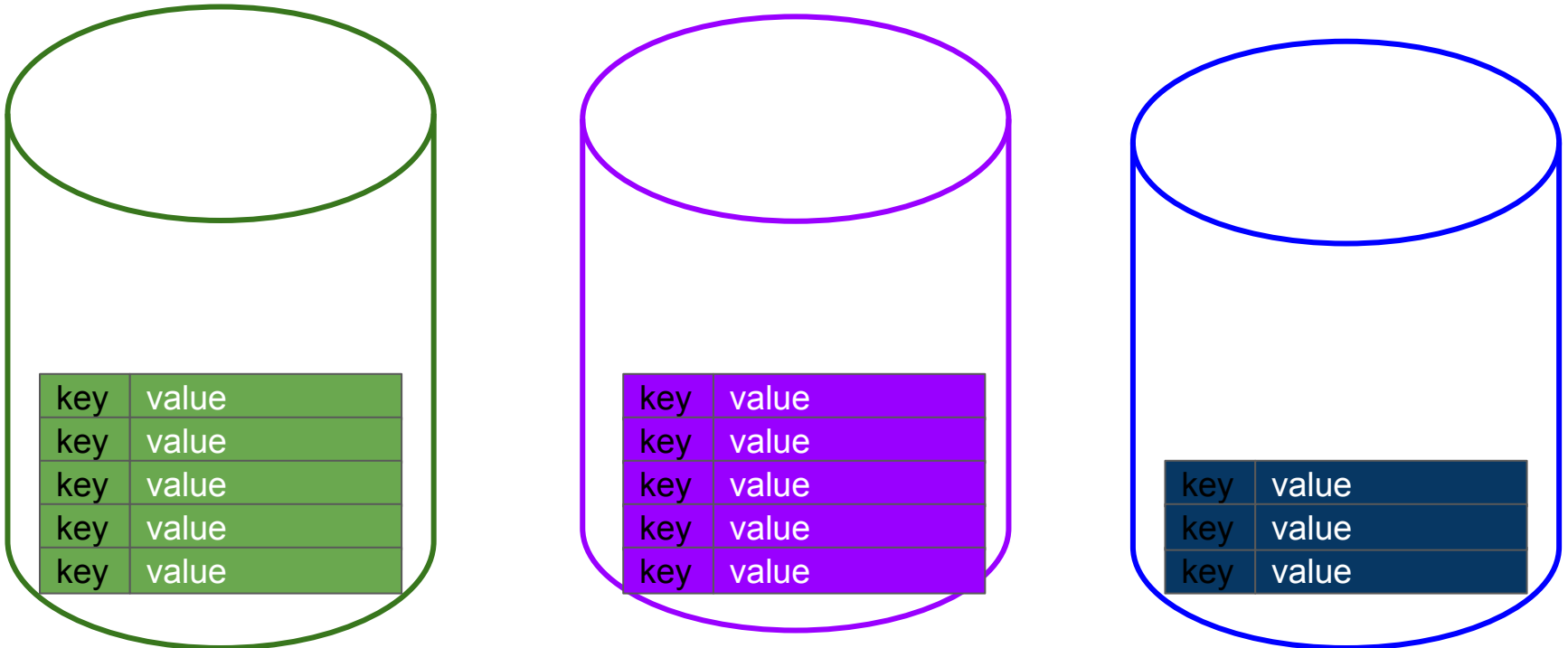


How do we fight Partition Intolerance?

How do we fight Unavailability?

Distributed Databases In a nutshell

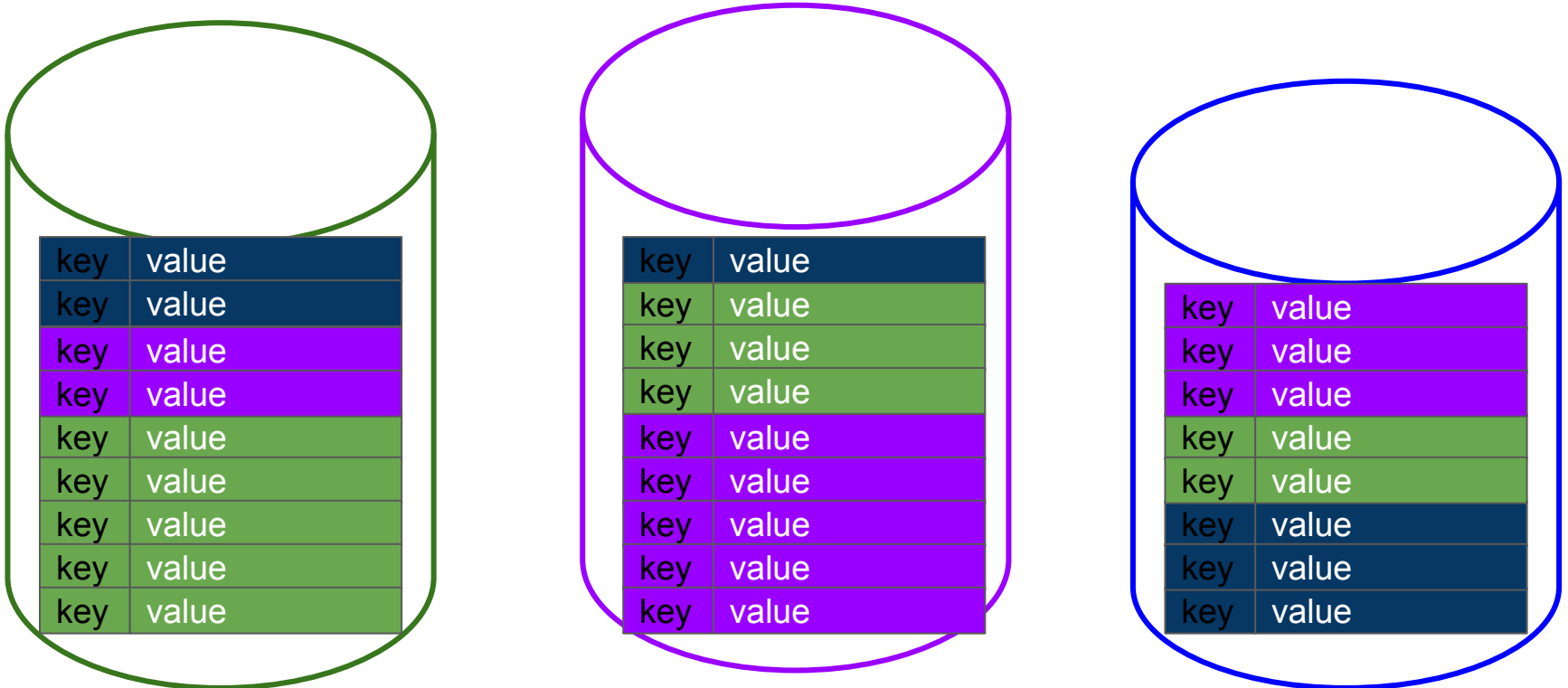
Replication	Sharding (Partitioning)



Distributed Databases In a nutshell

Replication

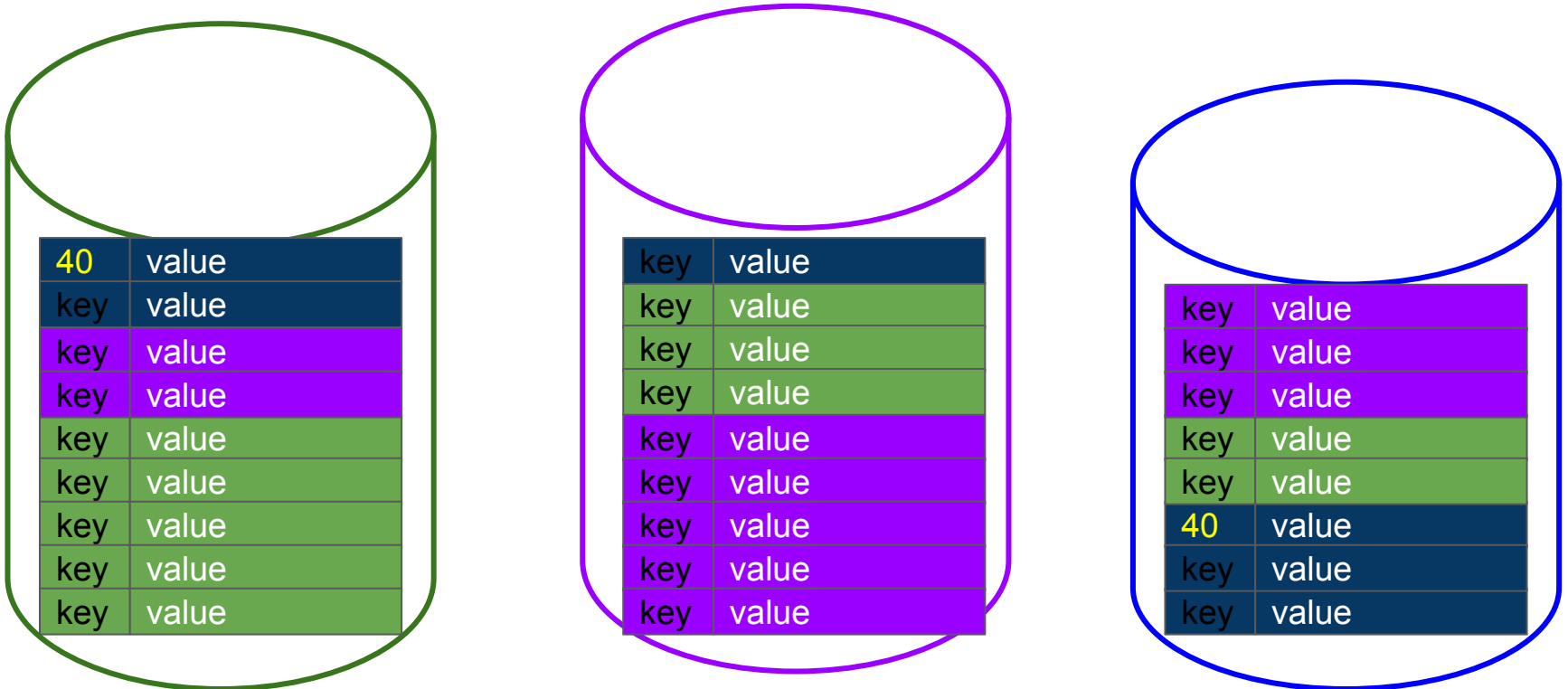
Sharding (Partitioning)



Distributed Databases In a nutshell

Sharding (Partitioning)
Replication

Problem: Consistency

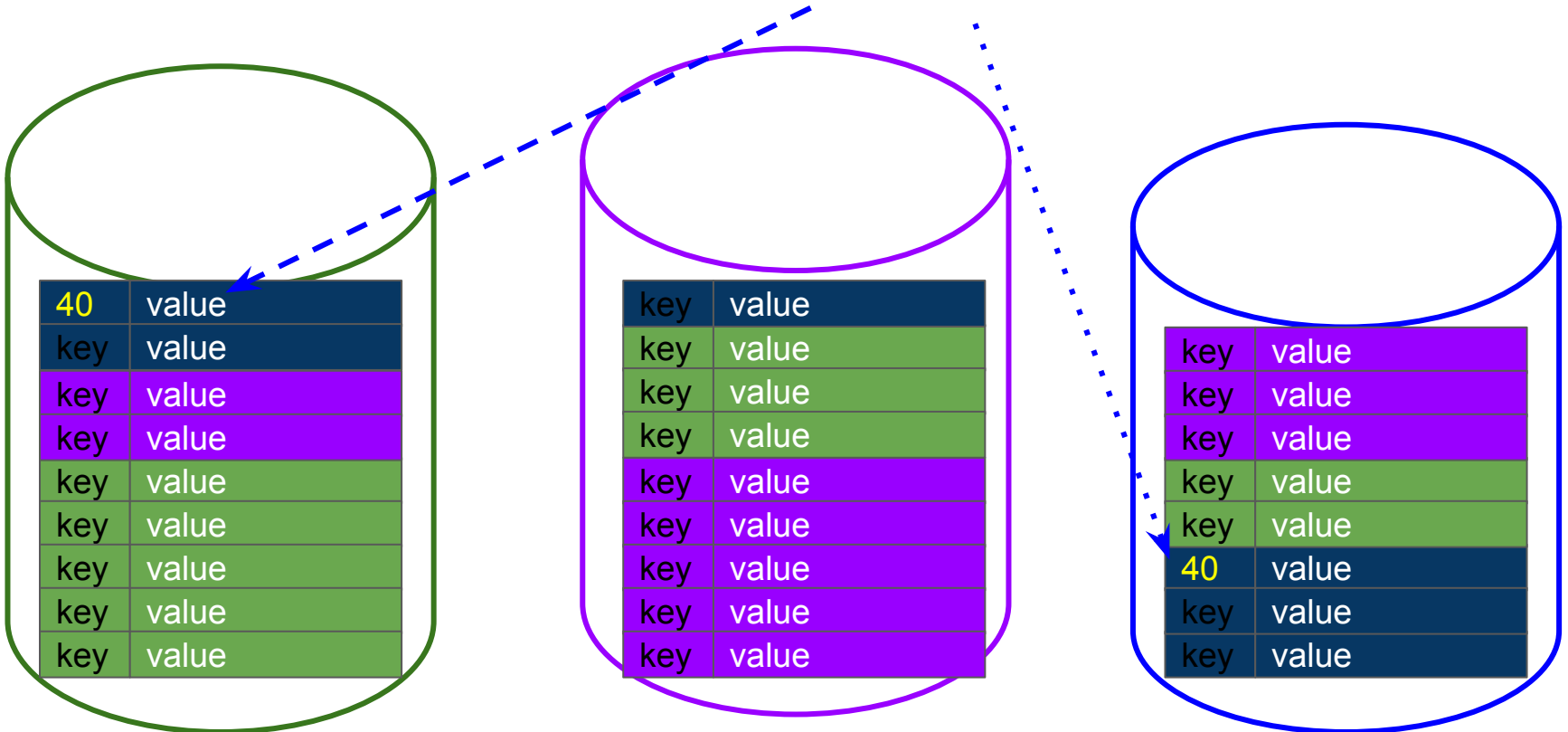


Distributed Databases In a nutshell

Sharding (Partitioning)
Replication

Problem: Consistency

update(40, NewValue)



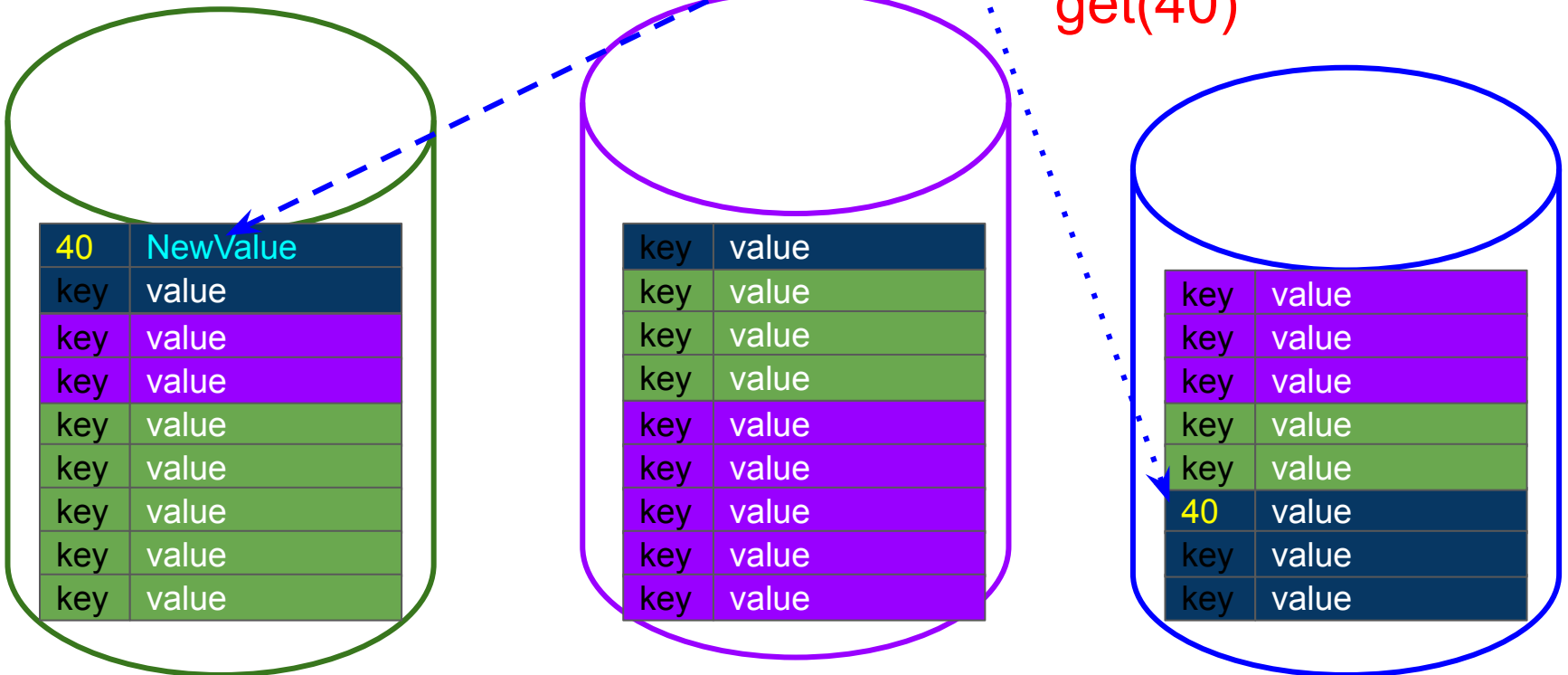
Distributed Databases In a nutshell

Sharding (Partitioning)
Replication

Problem: Consistency

update(40, NewValue)

get(40)



Desired Characteristics

Consistency: each read receives the most recent write

Availability: system returns a response

Partition-tolerance: system continues to operate despite messages being dropped

The CAP Theorem

Consistency: each read receives the most recent write

Availability: system returns a response

Partition-tolerance: system continues to operate despite messages being dropped

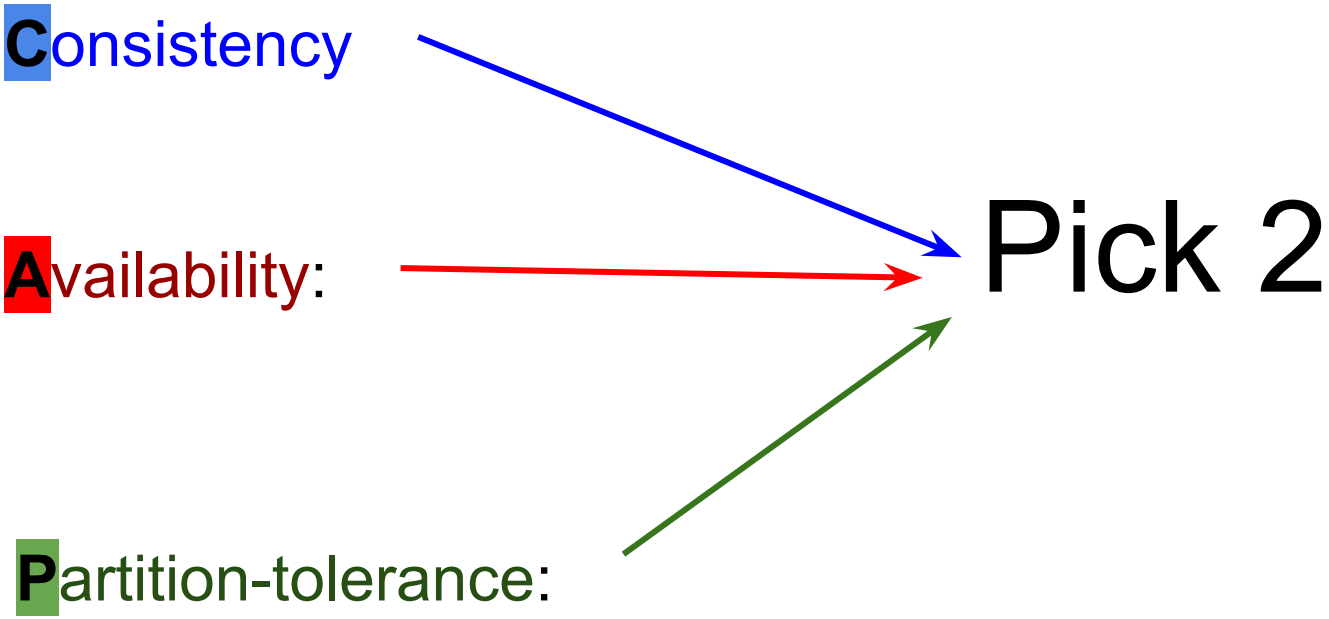
The CAP Theorem

Consistency

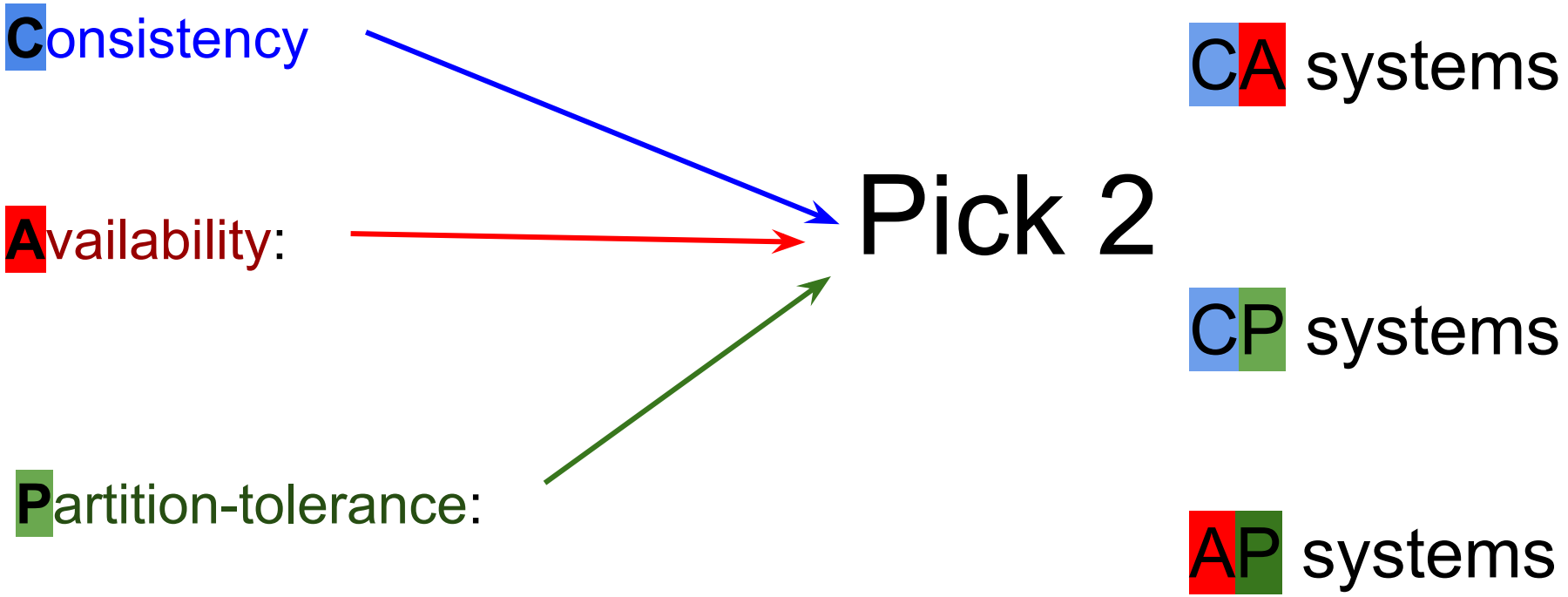
Availability:

Partition-tolerance:

The CAP Theorem



The CAP Theorem



The CAP Theorem

Not really **distributed**

Consistency

Availability:

Partition-tolerance:

Must have

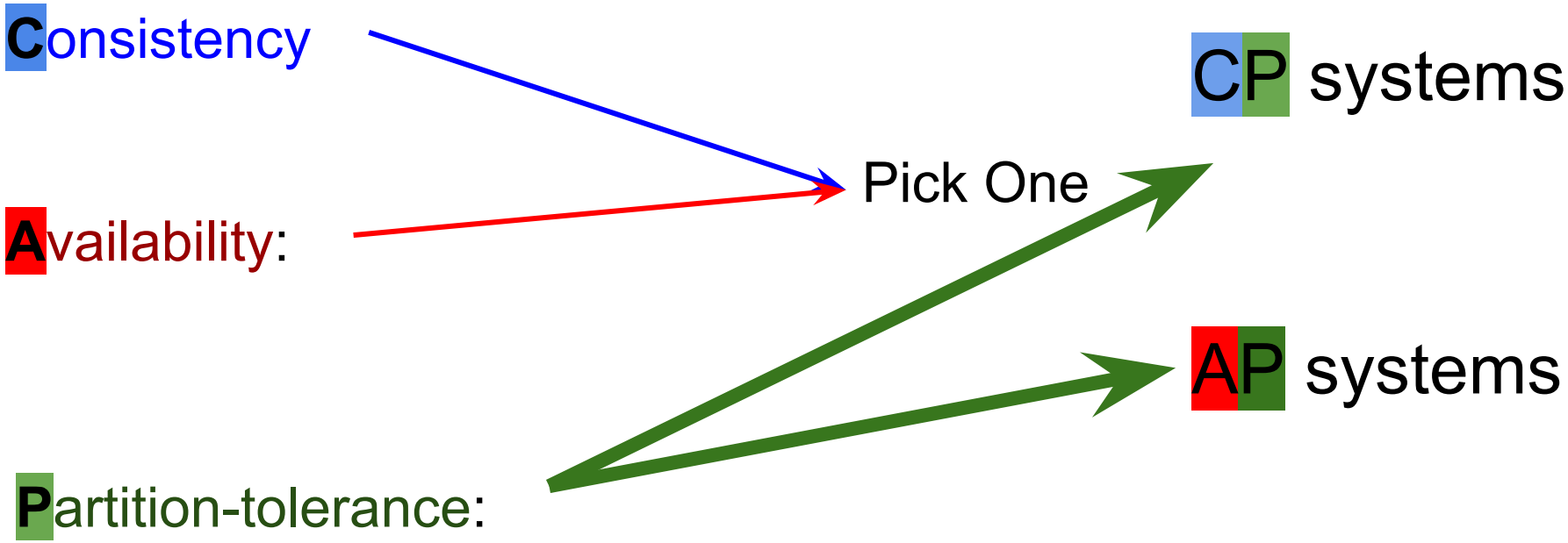
Pick 2

CA systems

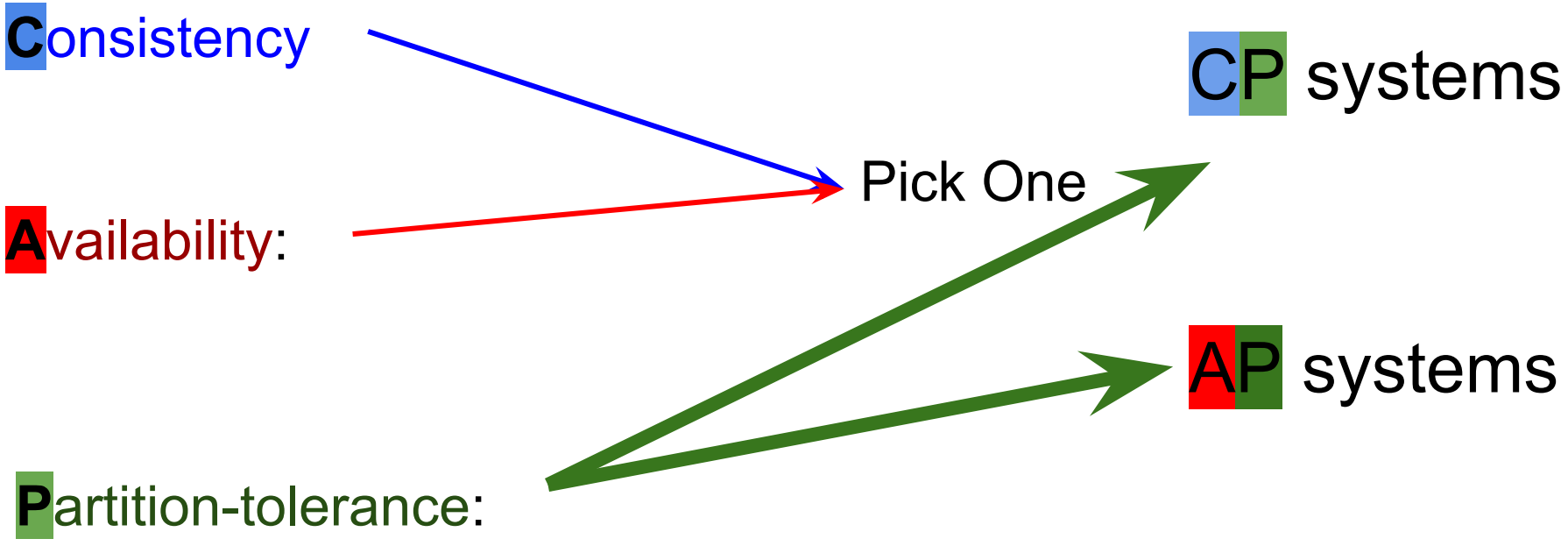
CP systems

AP systems

The CAP Theorem



The CAP Theorem



Partition-tolerant distributed systems can be either consistent or available but not both

The CAP Theorem

Consistency

Availability:

Partition-tolerance:

WHAT'S THE DIFFERENCE?

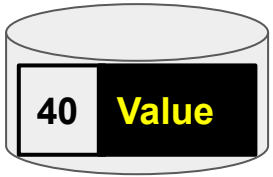
CP systems

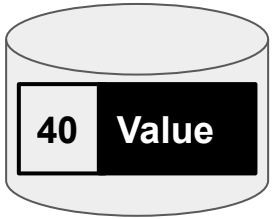
AP systems

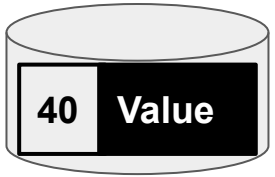
CP system behavior

the system will return an error or a time out if particular information cannot be guaranteed to be up to date due to network partitioning

get(40)

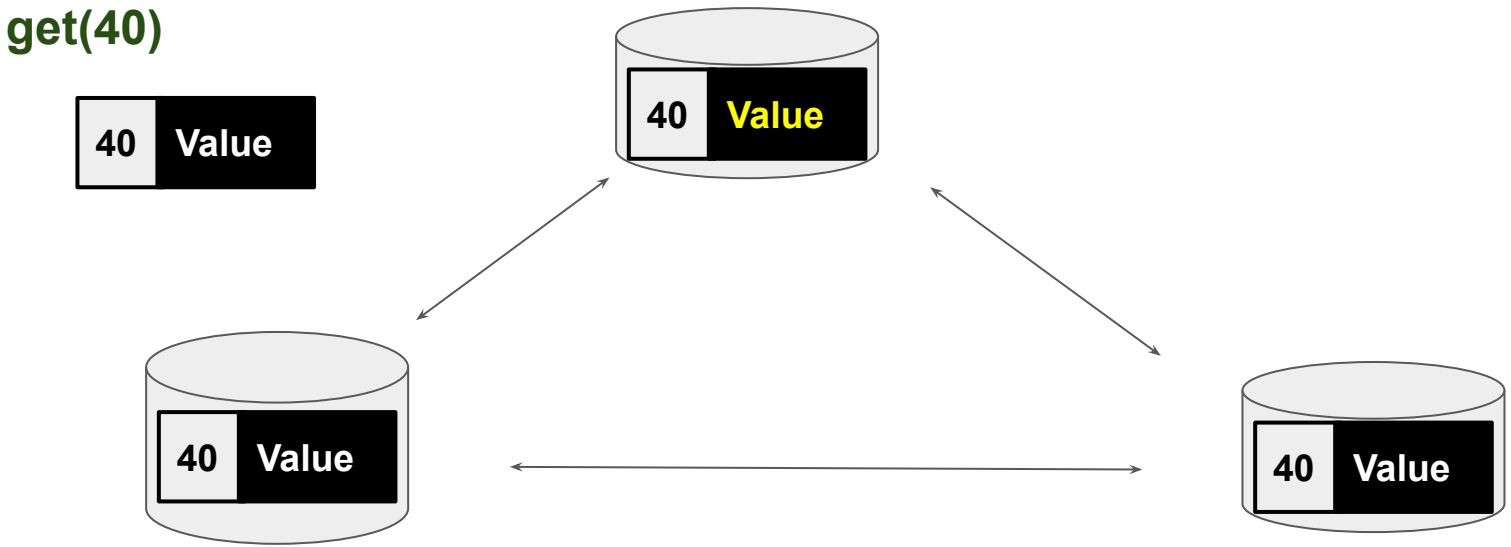






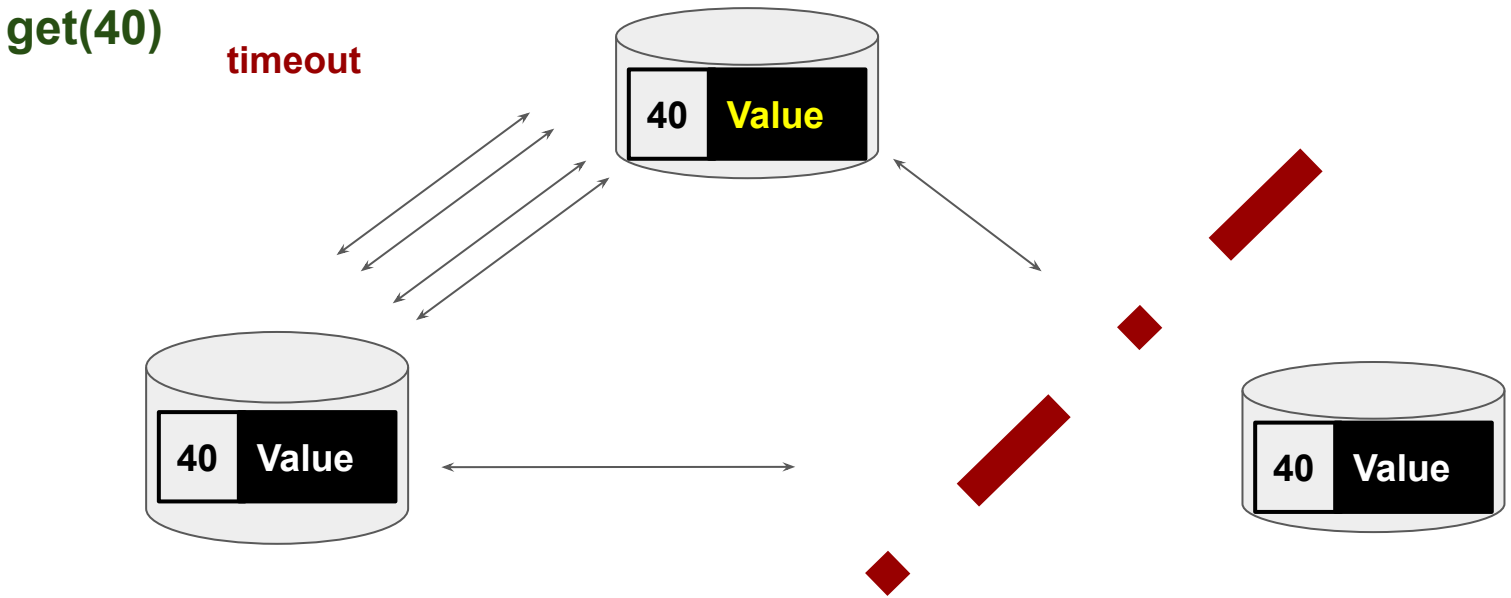
CP system behavior

the system will return an error or a time out if particular information cannot be guaranteed to be up to date due to network partitioning



CP system behavior

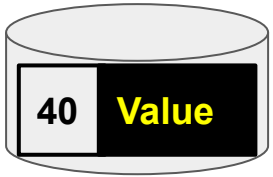
the system will return an error or a time out if particular information cannot be guaranteed to be up to date due to network partitioning

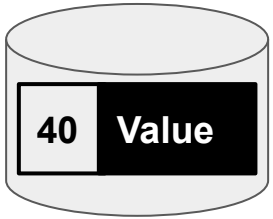


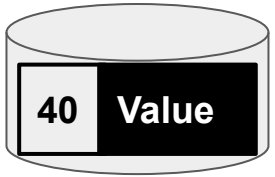
AP system behavior

The system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.

get(40)

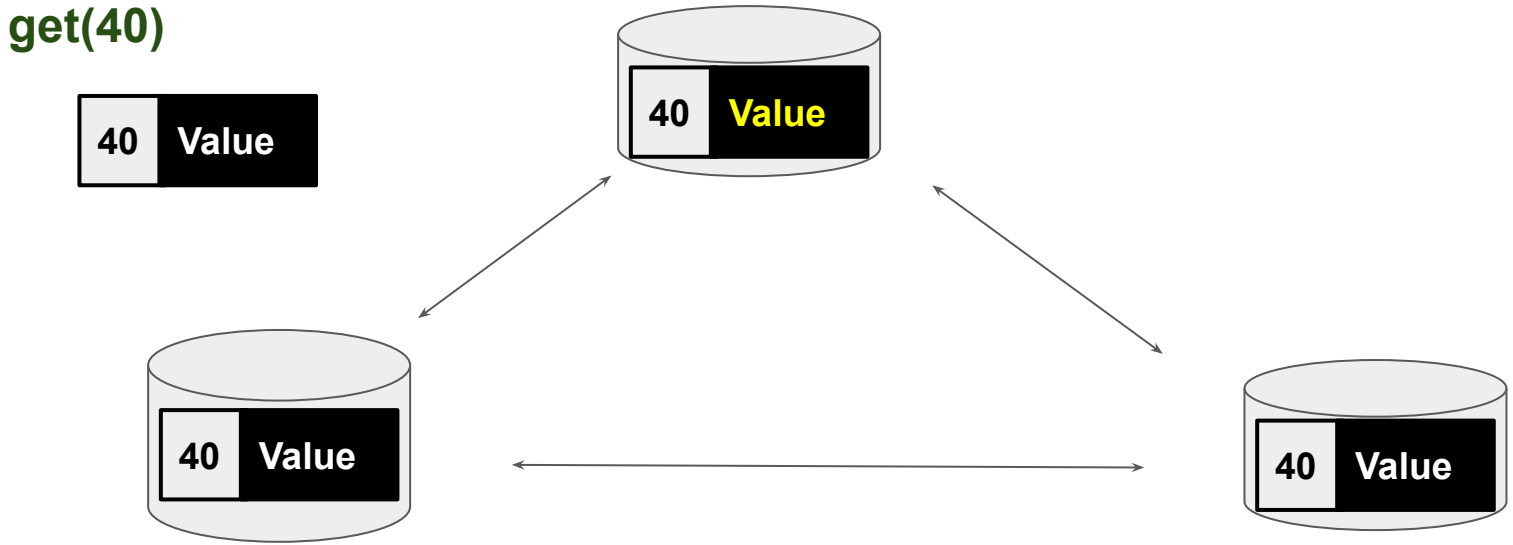






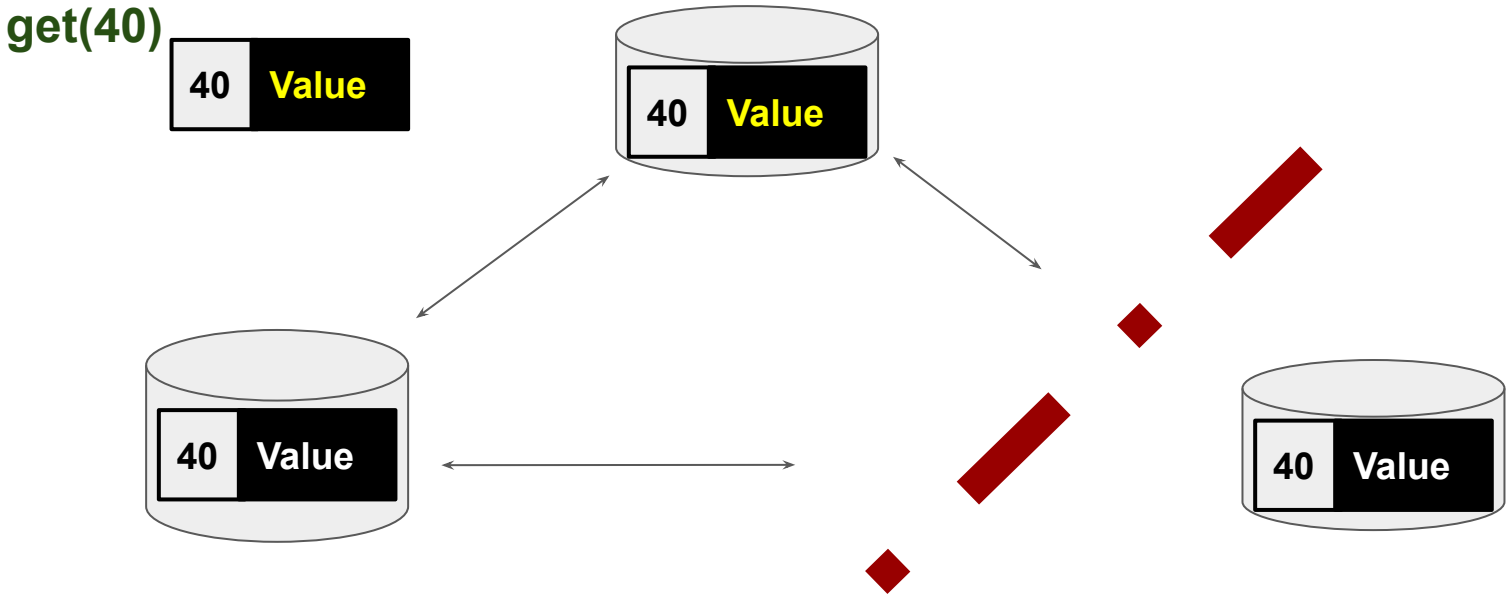
AP system behavior

The system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.



AP system behavior

The system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.



The CAP Theorem

WHAT'S THE DIFFERENCE?

CP systems

- ➔ Mission Critical Applications
- ➔ The "RIGHT" Answer is important

AP systems

- ➔ Web & Social Apps
- ➔ The FAST Answer is important

Partition-tolerance:

Consistency

Availability: