

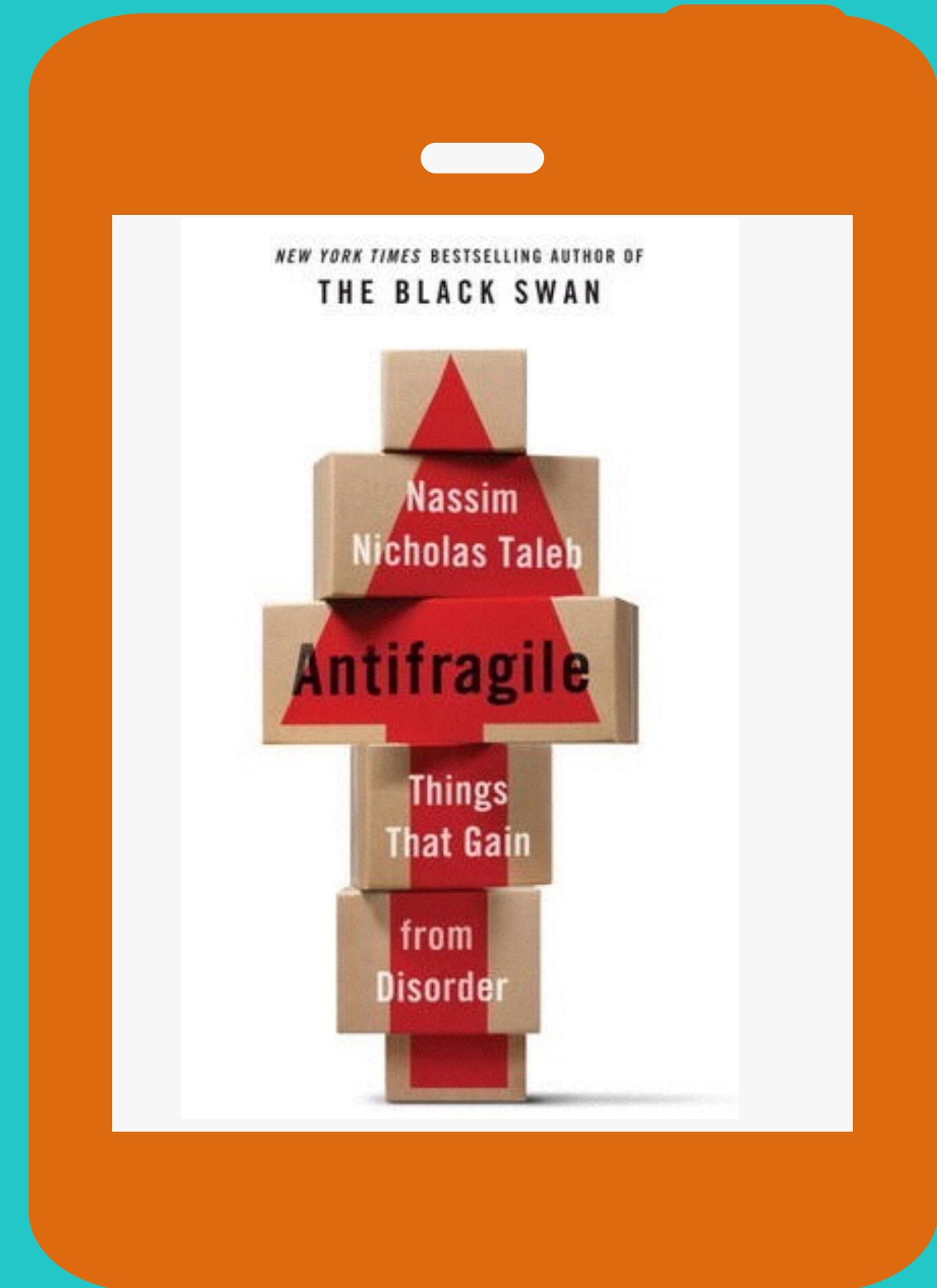
ANTI-FRAGILE CLOUD

ARCHITECTURES

AGIM EMRULI - @AEMRULI - MIMACOM

“Antifragility is beyond resilience or robustness. The resilient resists shocks and stays the same; the antifragile gets better.”

Nasim Nicholas Taleb



FRAGILE

NON-LINEAR
(KONKAV)

POST-TRAUMATIC
SYNDROM

CENTRALIZED

ROBUST

LINEAR

ANTI-FRAGILE

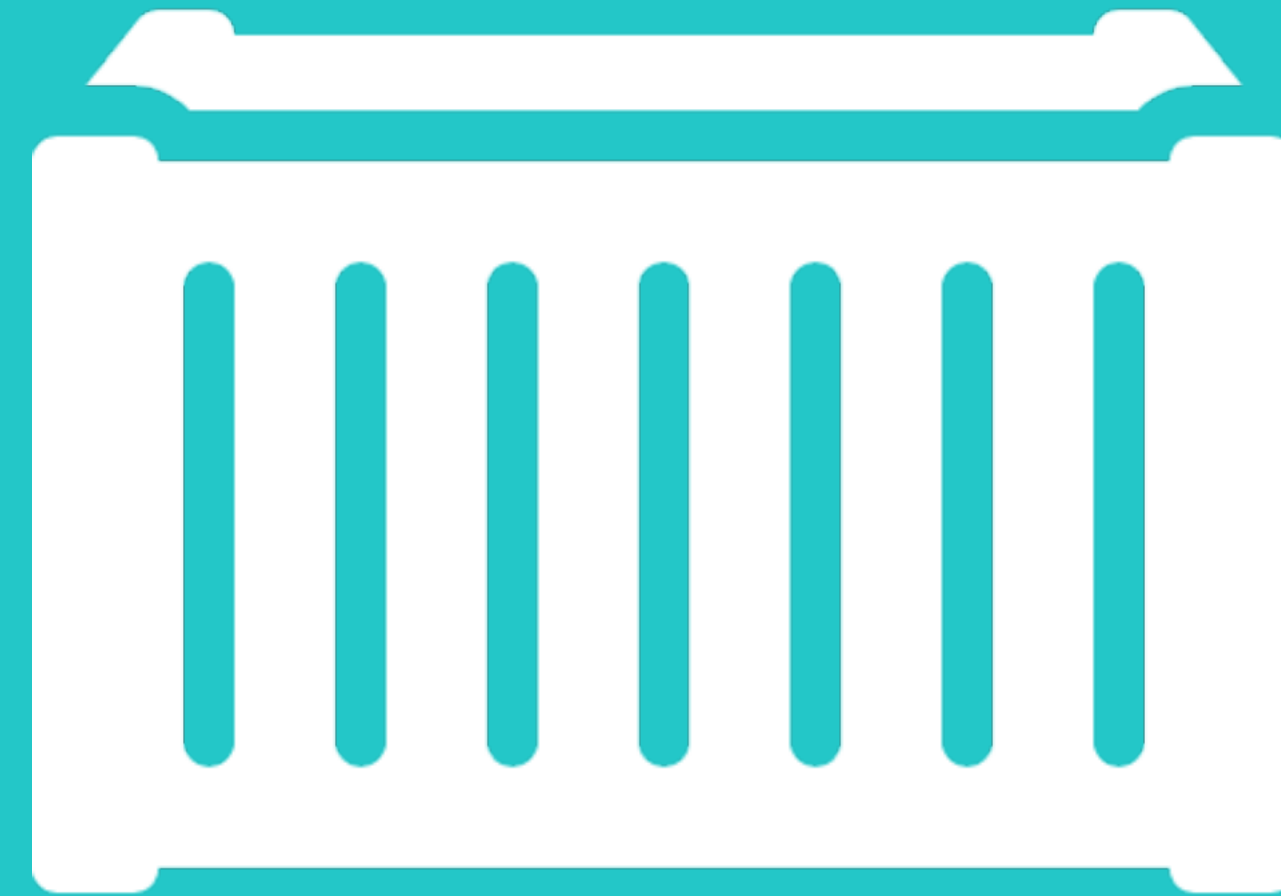
NON-LINEAR
(KONVEX)

POST-TRAUMATIC
GROWTH

DECENTRALIZED



FRAGILE



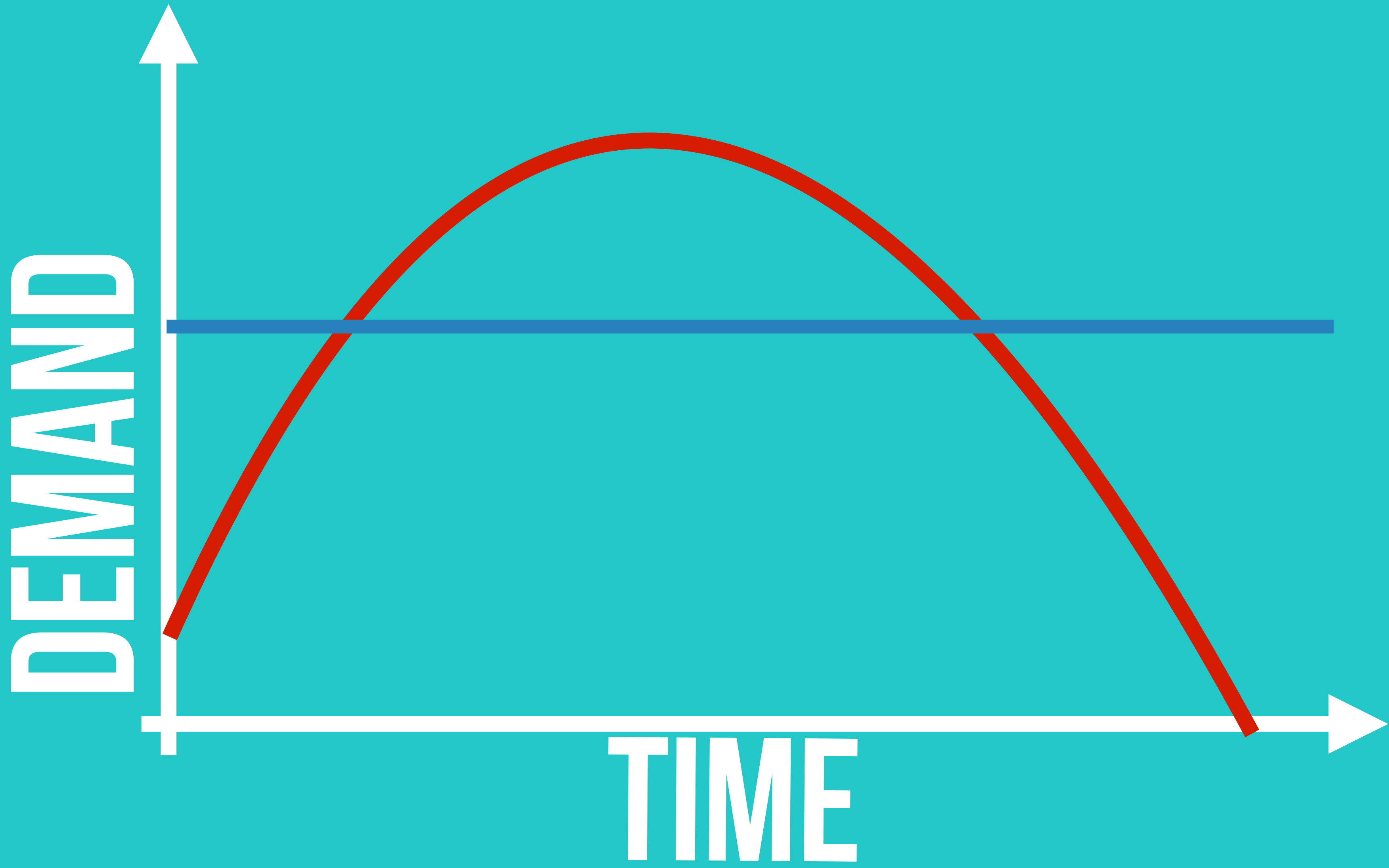
ROBUST



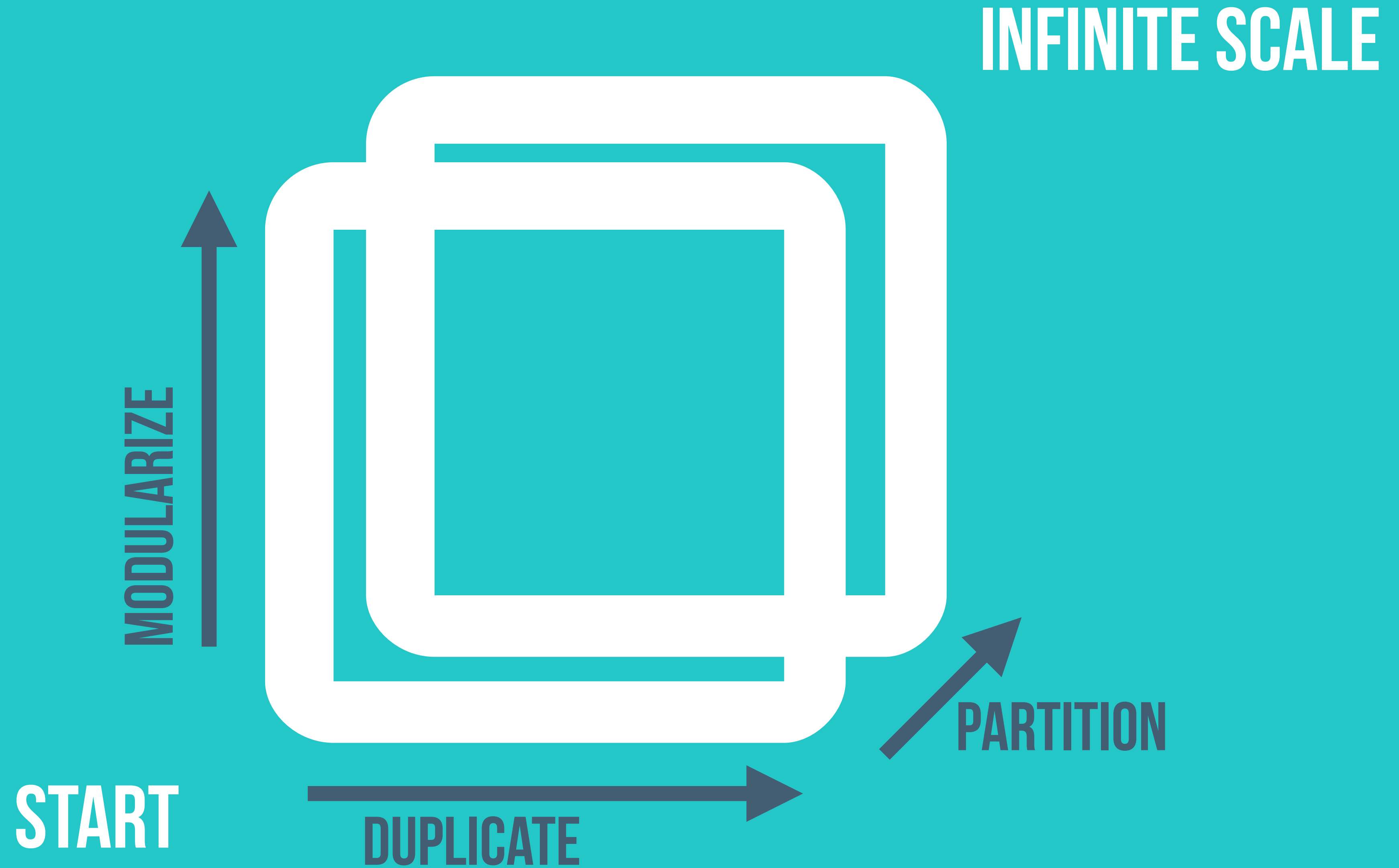
FRAGILE



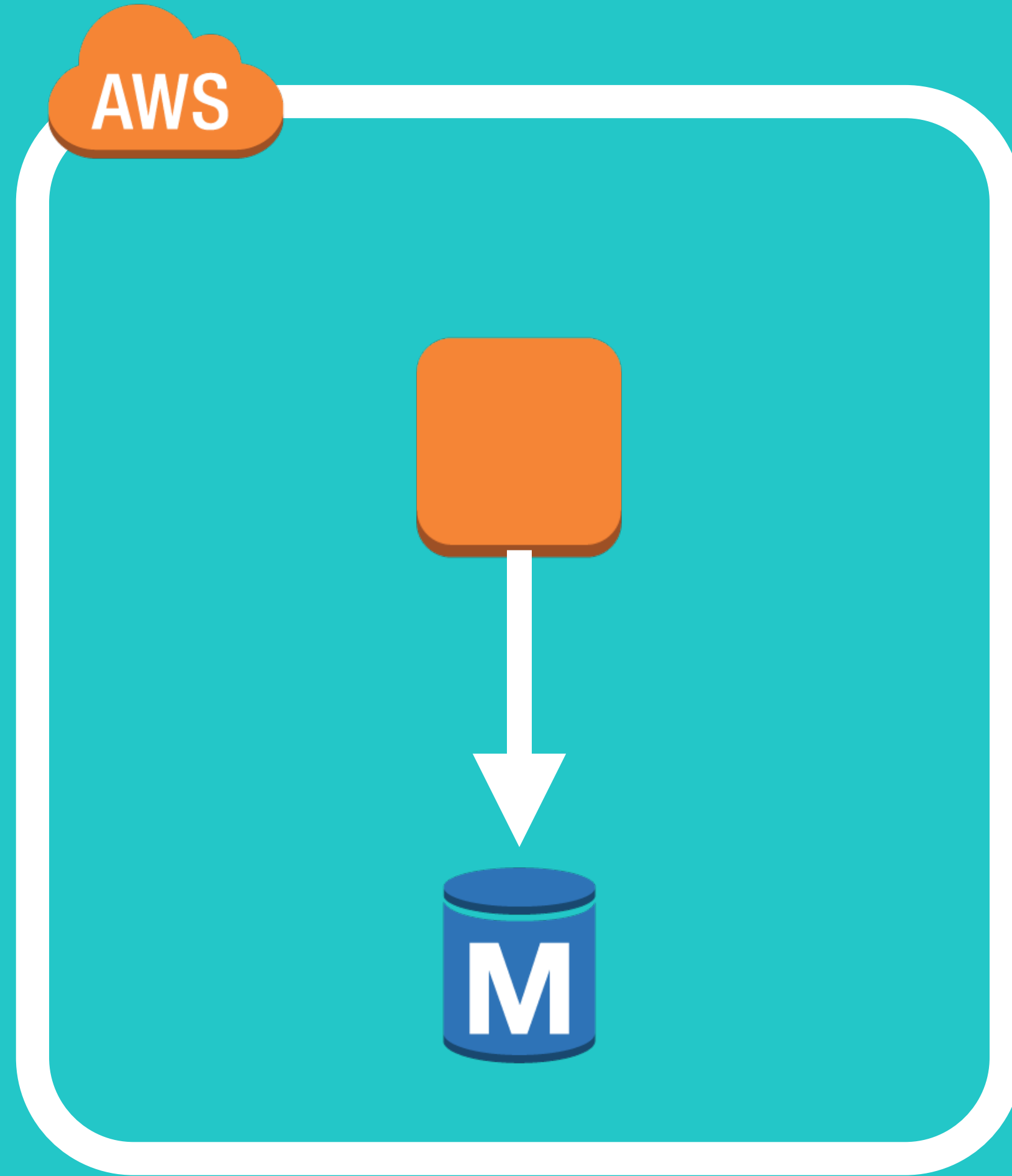
ANTI-FRAGILE



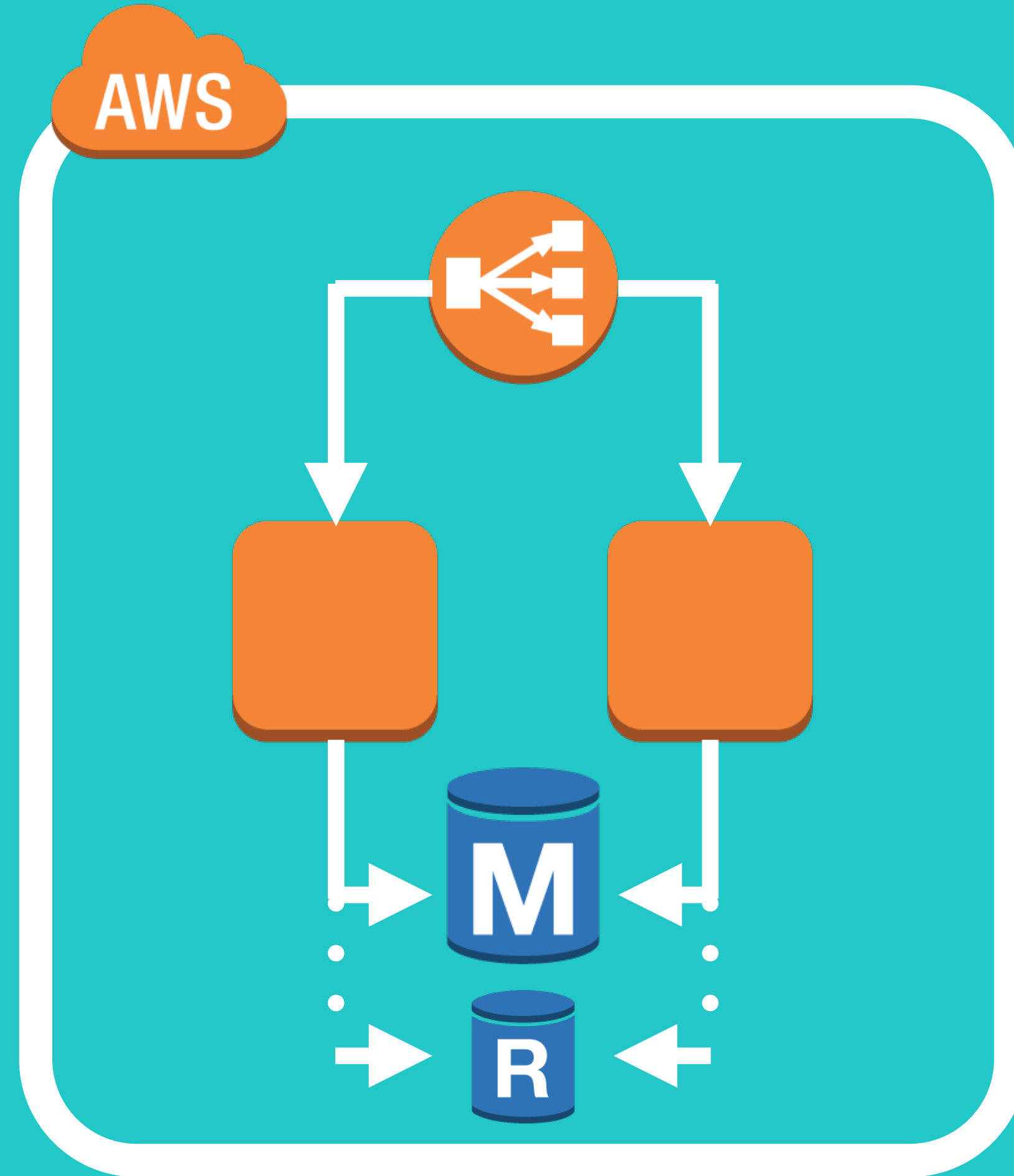
SCALE CUBE

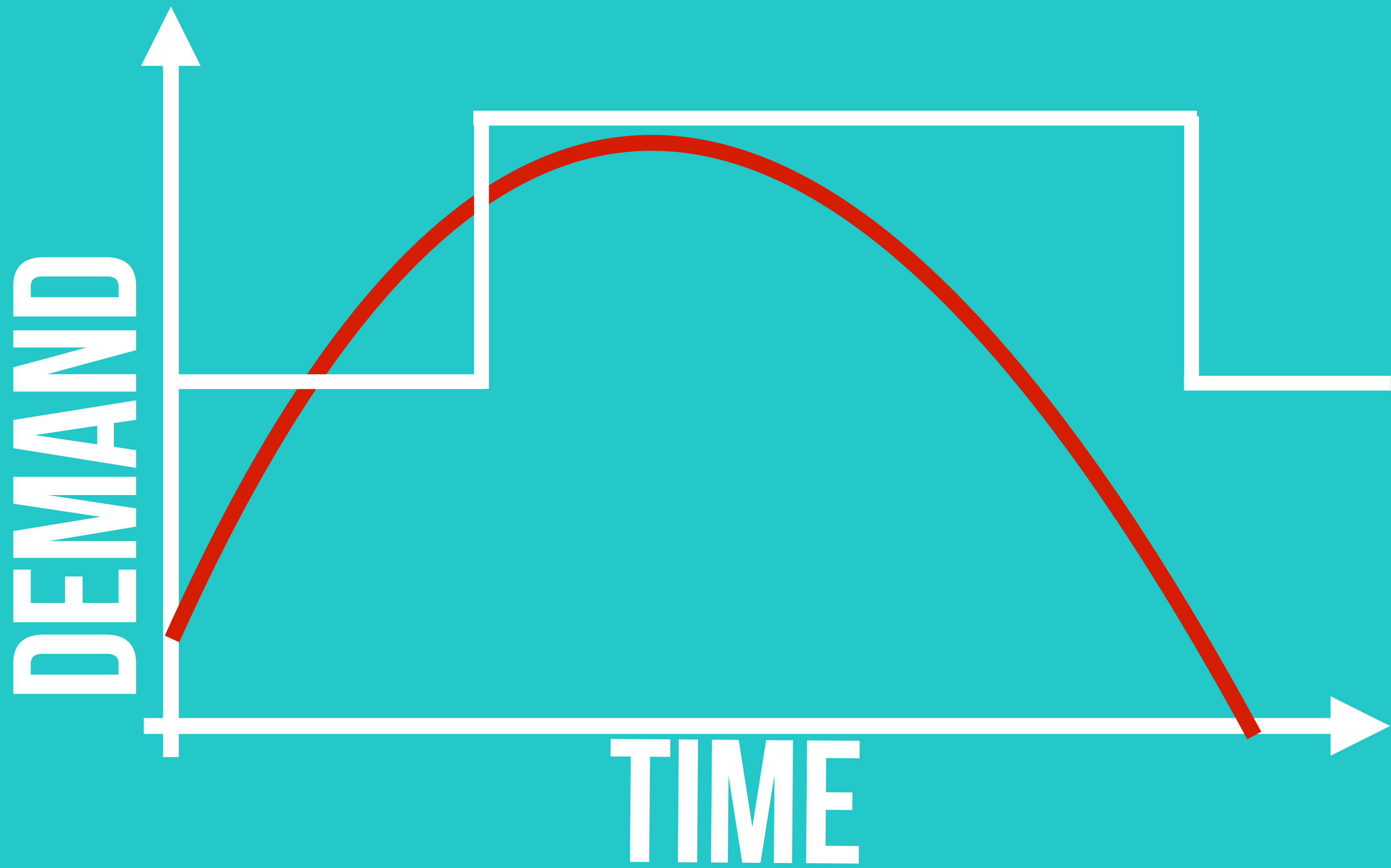


START

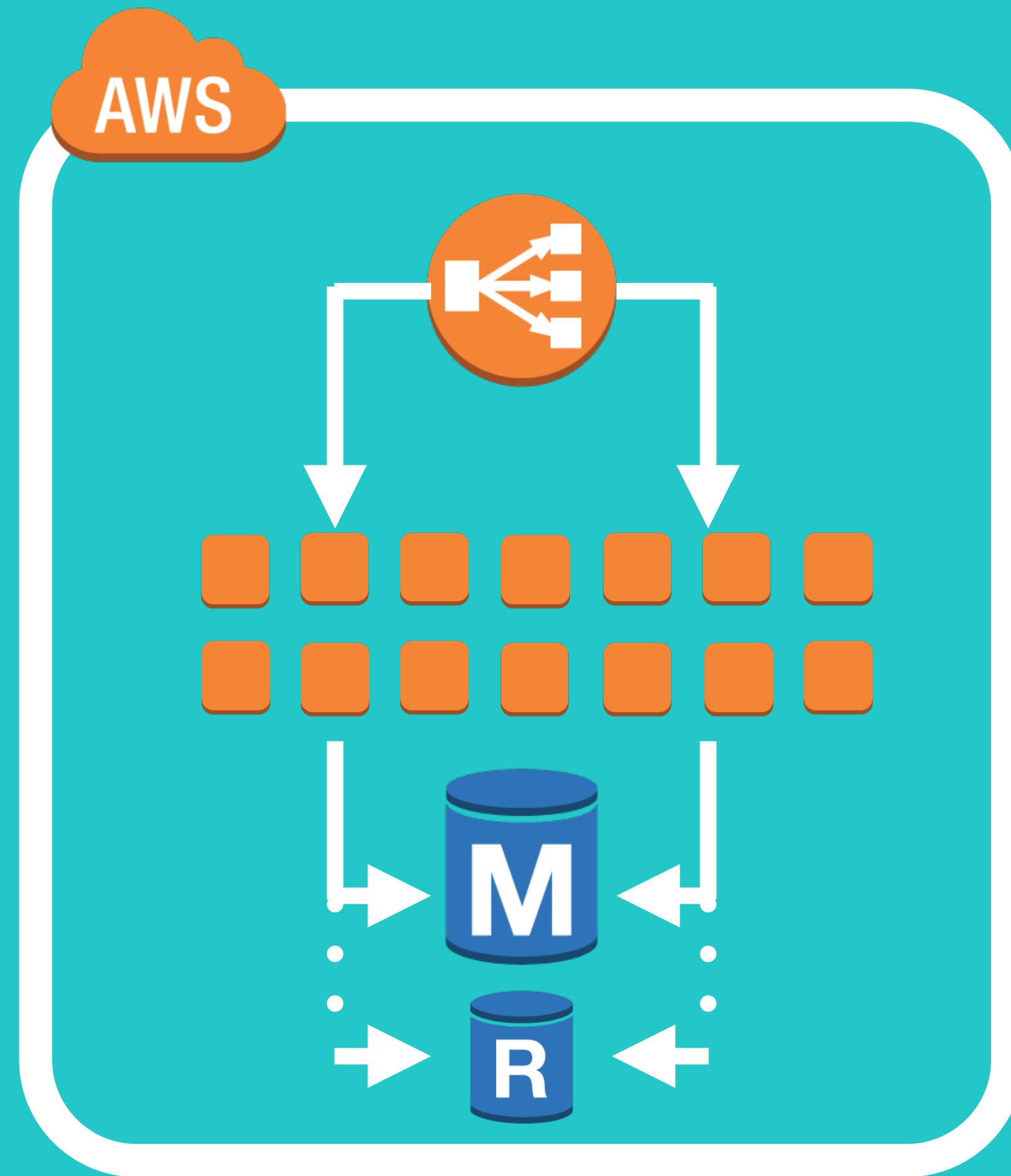


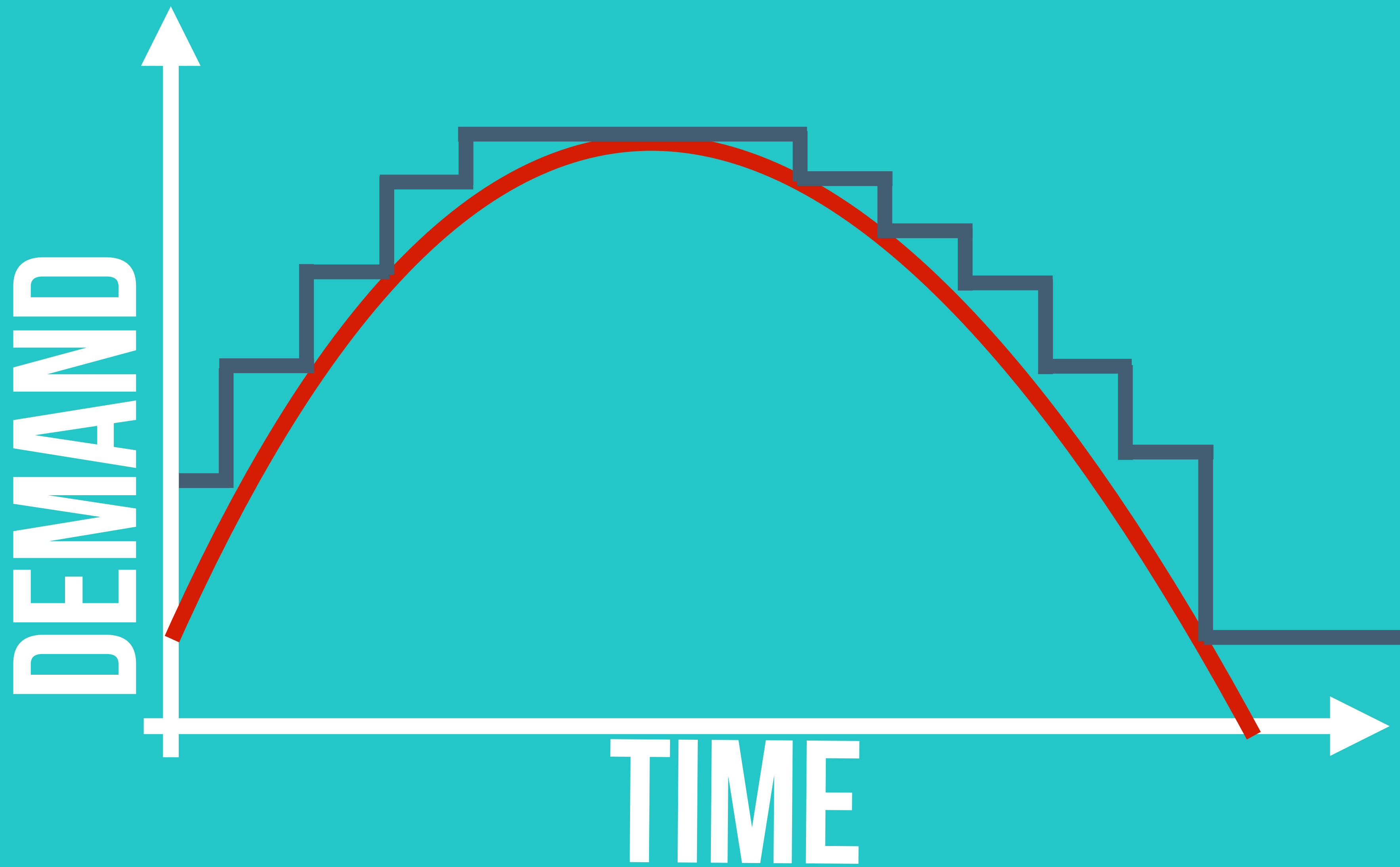
DUPLICATION



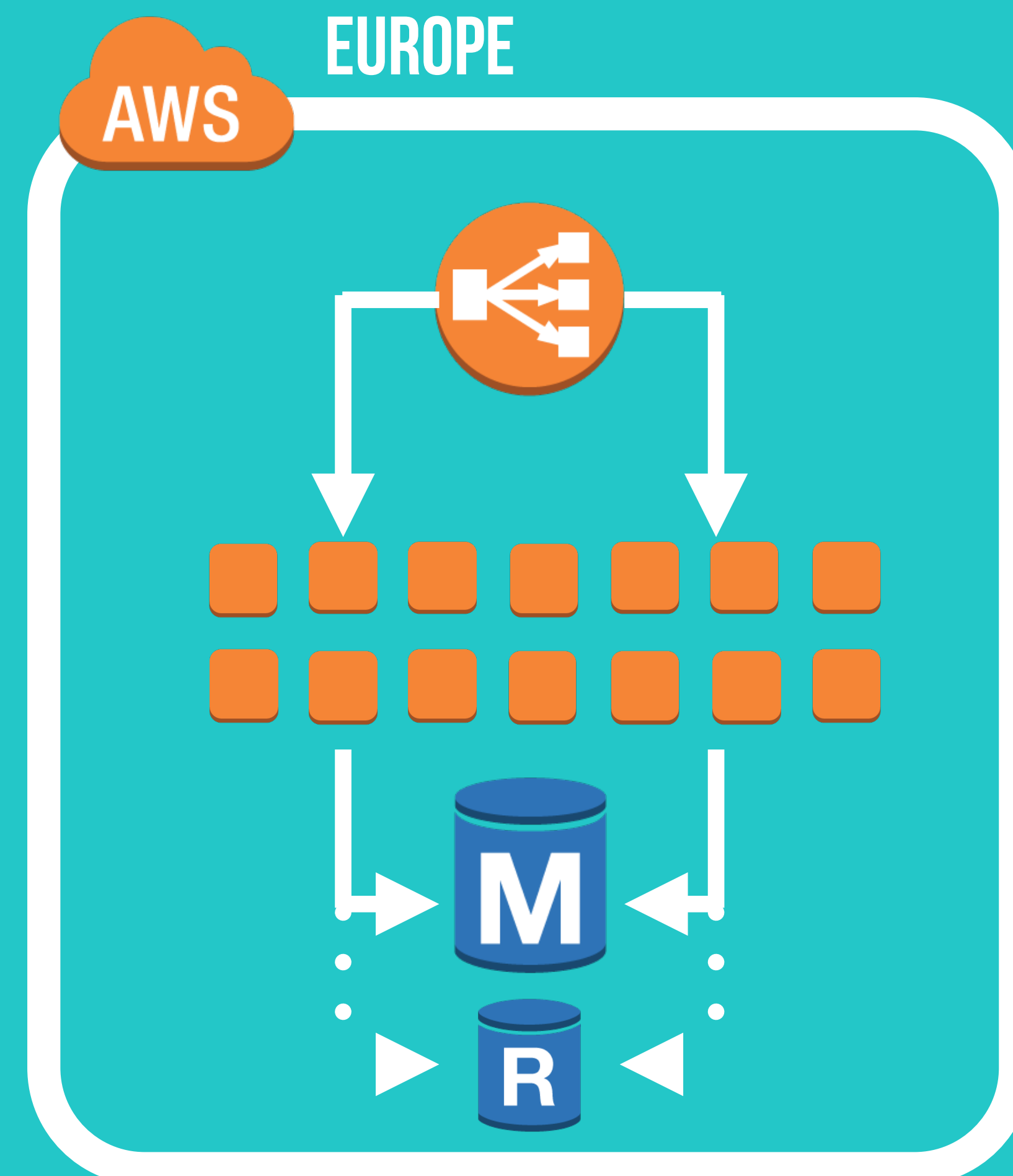
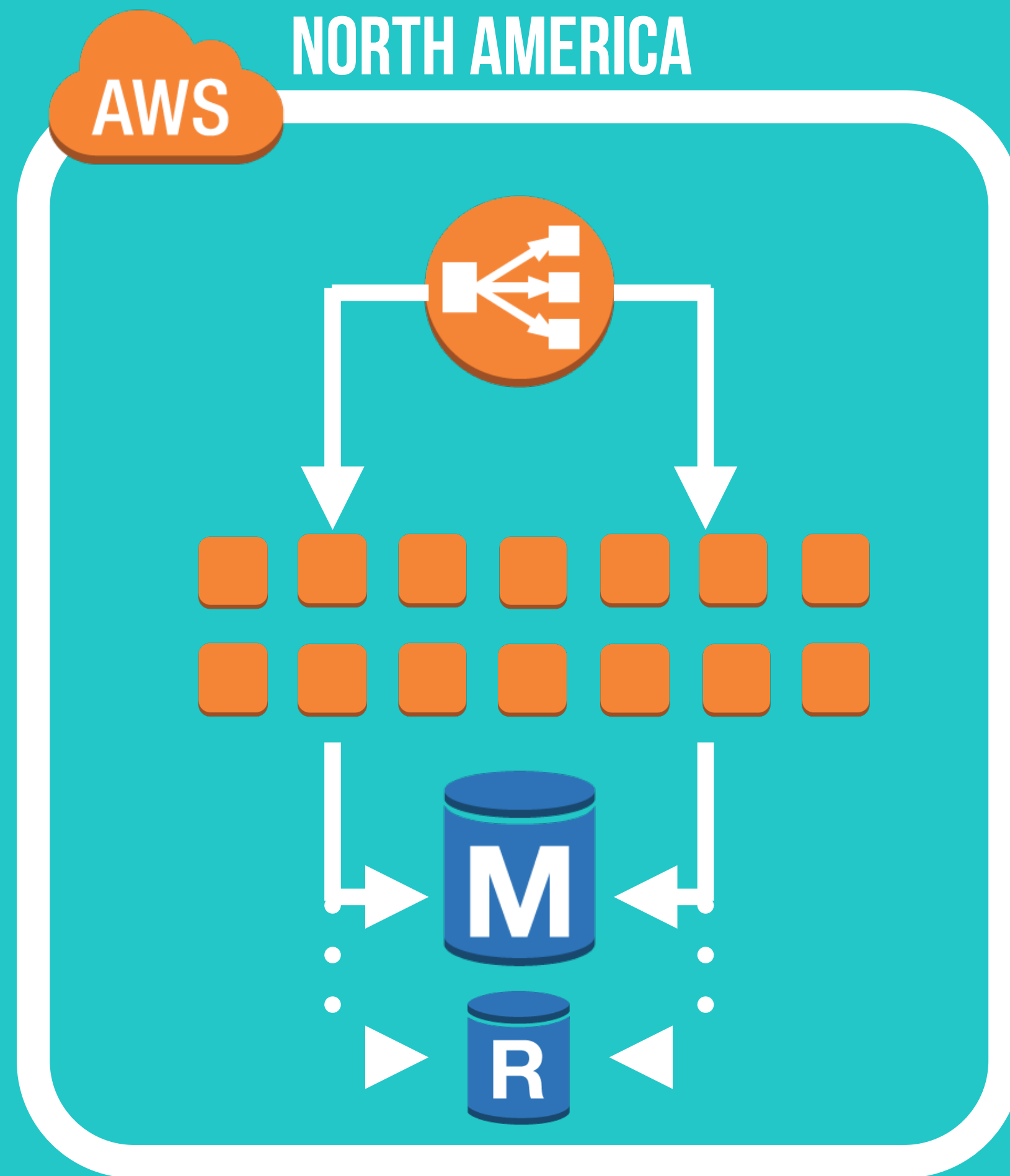


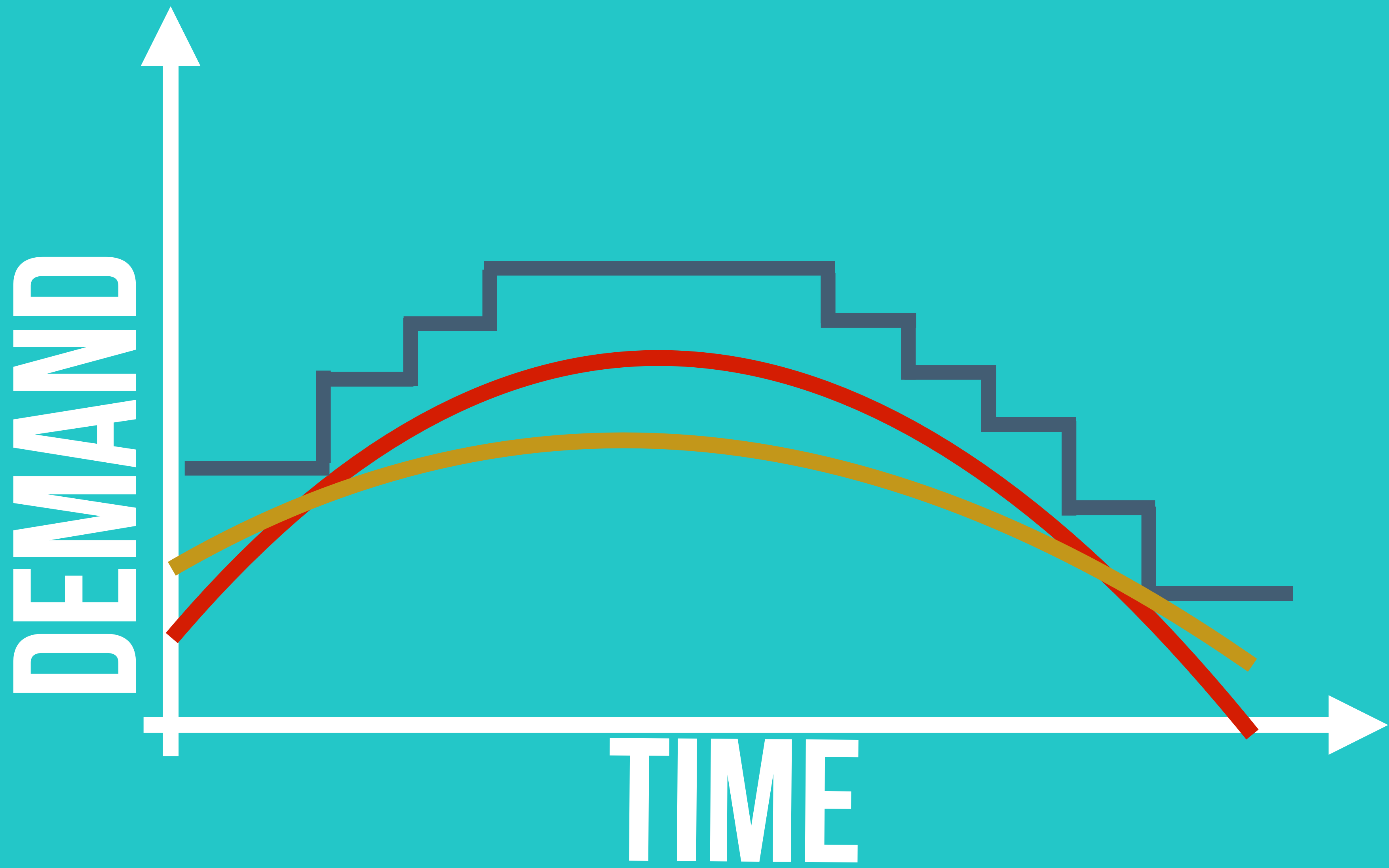
MODULARIZATION





PARTIONING

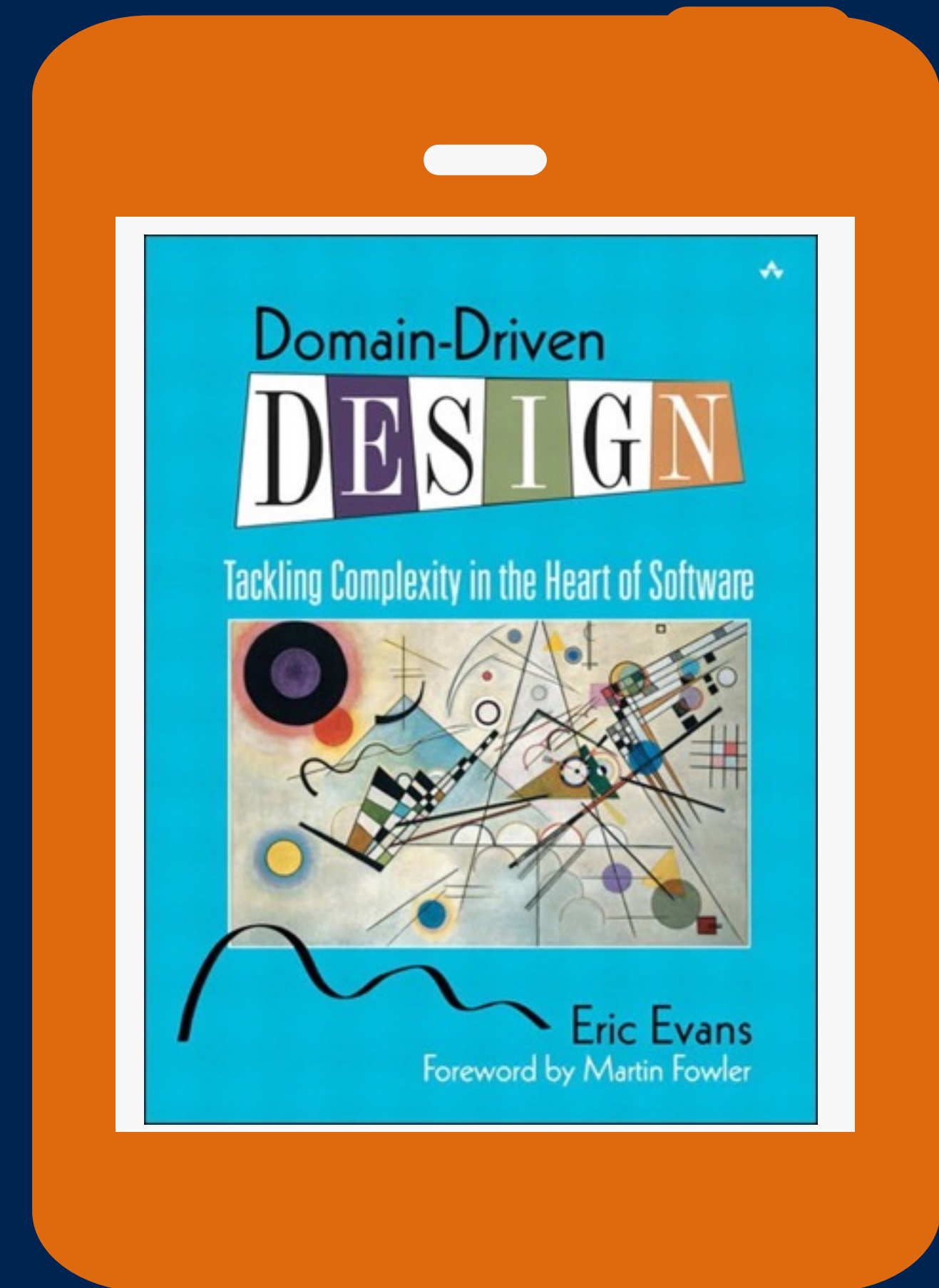


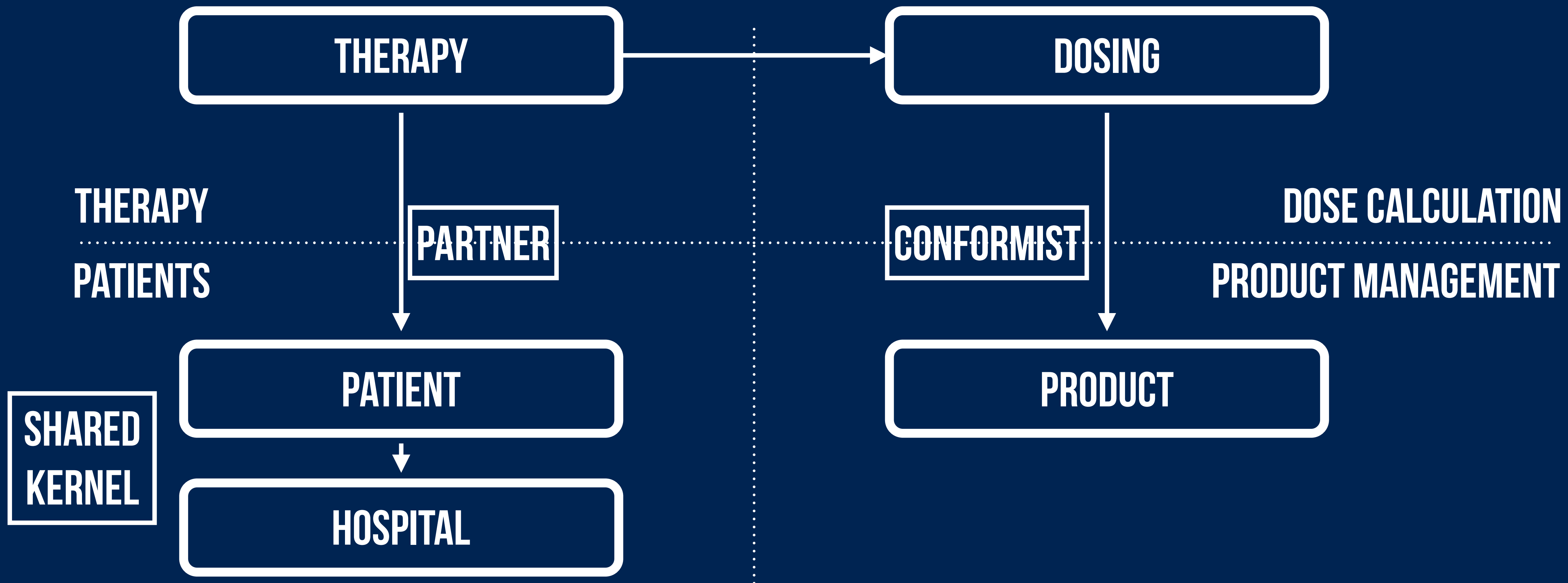


{YOU NAME IT}
SERVICE



A MICROSERVICES ARCHITECTURE IS A SERVICE-ORIENTED ARCHITECTURE COMPOSED OF LOOSELY COUPLED ELEMENTS THAT HAVE **BOUNDED CONTEXTS**





COMPONENT

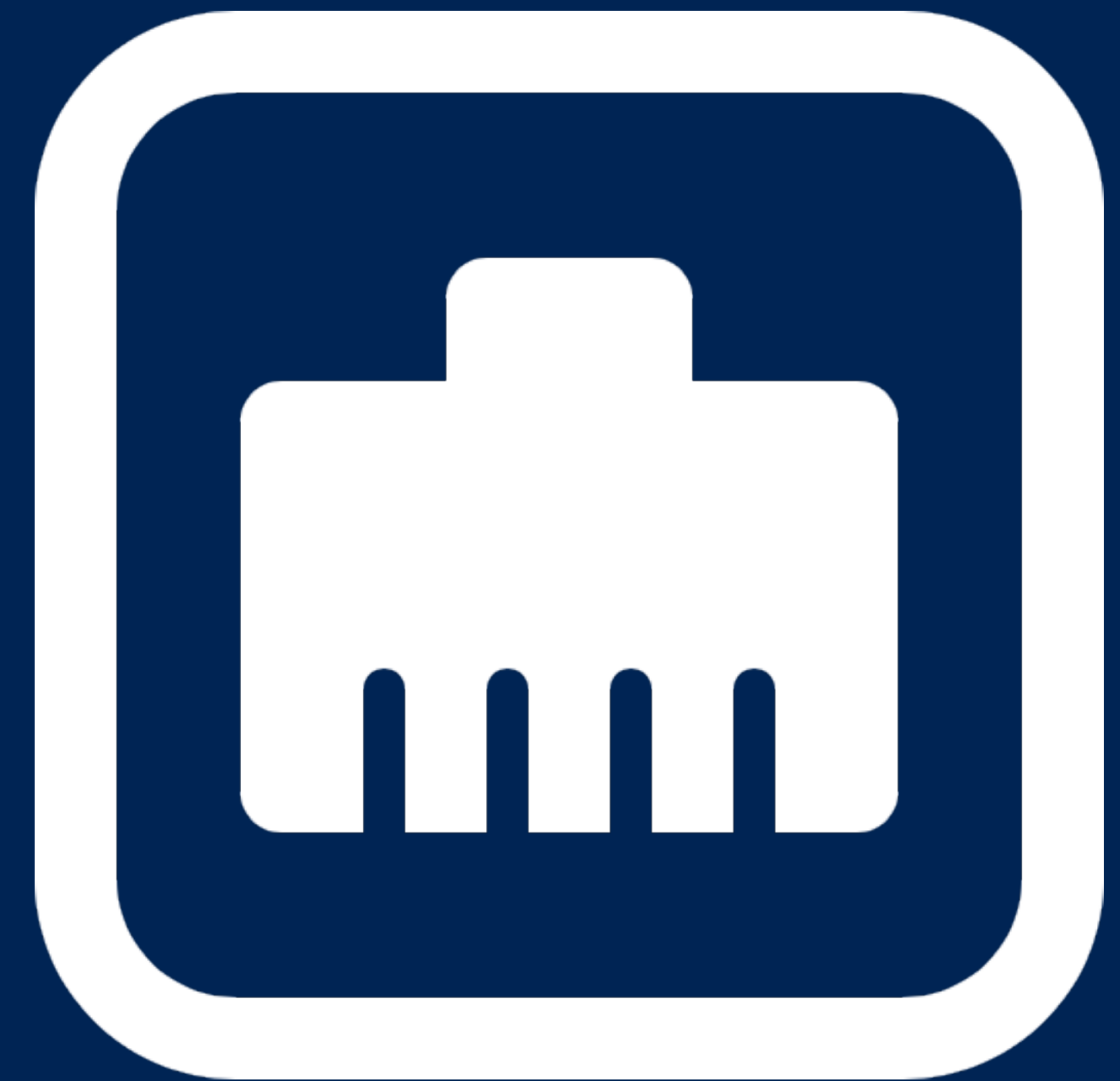
COMPONENT

COMPONENT

COMPONENT



- THE NETWORK IS RELIABLE
- LATENCY IS ZERO
- BANDWIDTH IS INFINITE
- THE NETWORK IS SECURE
- TOPOLOGY DOESN'T CHANGE
- THERE IS ONE ADMINISTRATOR
- TRANSPORT COST IS ZERO
- THE NETWORK IS HOMOGENEOUS



- ✘ THE NETWORK IS RELIABLE
- ✘ LATENCY IS ZERO
- ✘ BANDWIDTH IS INFINITE
- ✘ THE NETWORK IS SECURE
- ✘ TOPOLOGY DOESN'T CHANGE
- ✘ THERE IS ONE ADMINISTRATOR
- ✘ TRANSPORT COST IS ZERO
- ✘ THE NETWORK IS HOMOGENEOUS

THE NETWORK FALLACIES

[HTTP://WWW.RGOARCHITECTS.COM/FILES/FALLACIES.PDF](http://www.rgoarchitects.com/files/fallacies.pdf)

TIMEOUT



```
public class MessageReceiver {  
  
    private Session session;  
    private Destination destination;  
  
    public void doReceive() throws Exception{  
        MessageConsumer consumer =  
            session.createConsumer(destination);  
  
        consumer.receive();  
    }  
}
```

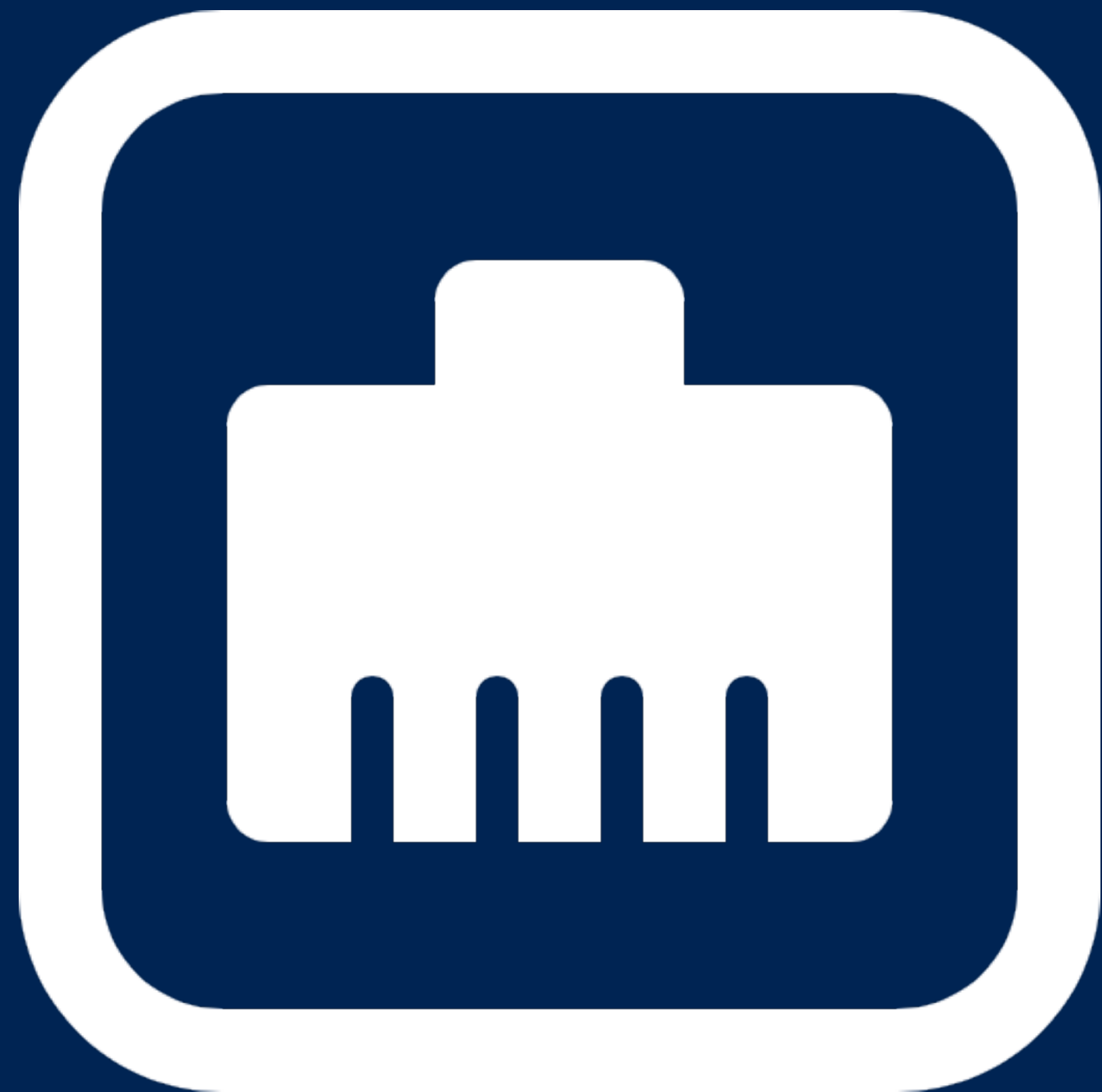
```
public class MessageReceiver {  
    private Session session;  
    private Destination destination;  
  
    public void doReceive() throws Exception{  
        MessageConsumer consumer =  
            session.createConsumer(destination);  
  
        consumer.receive(20L);  
    }  
}
```

```
public class HttpReceiver {  
    public String getResource() throws IOException {  
        URL url = new URL("http://www.google.de");  
        InputStream inputStream = url.openStream();  
        return "...";  
    }  
}
```



```
public class HttpReceiver {  
  
    public String getResource() throws IOException {  
        URL url = new URL("http://www.google.de");  
        URLConnection urlConn = url.openConnection();  
        urlConn.setConnectTimeout(10);  
        urlConn.setReadTimeout(10);  
        InputStream inputStream = url.getInputStream();  
        return "...";  
    }  
  
}
```

```
public class DataSourceConfig {  
    public void DataSource setupDataSource(){  
        BasicDataSource basicDataSource =  
            new BasicDataSource();  
        basicDataSource.setMaxWait(30L);  
    }  
}
```



ADD LATENCY

```
tc qdisc add dev eth0 root latency delay 1000ms 500ms
```

CORRUPT PACKAGES

```
tc qdisc add dev eth0 root netem corrupt 5%
```

DROP PACKAGES

```
tc qdisc add dev eth0 root netem loss 7% 25%
```

BLOCK DNS

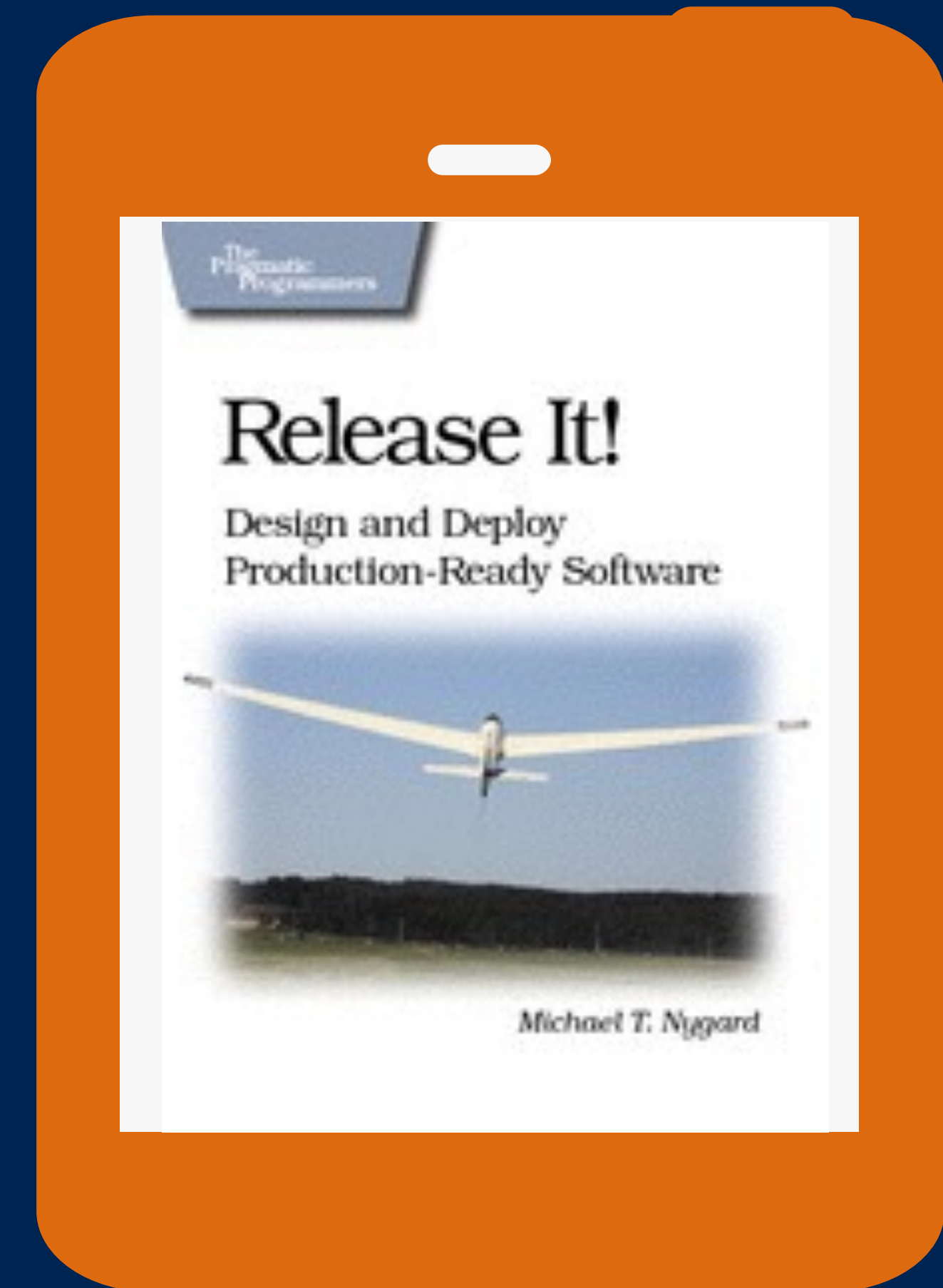
```
iptables -A INPUT -p tcp -m tcp --dport 53 -j DROP
```

PATTERNS

STABILITY

CAPACITY

TRANSPARENCY

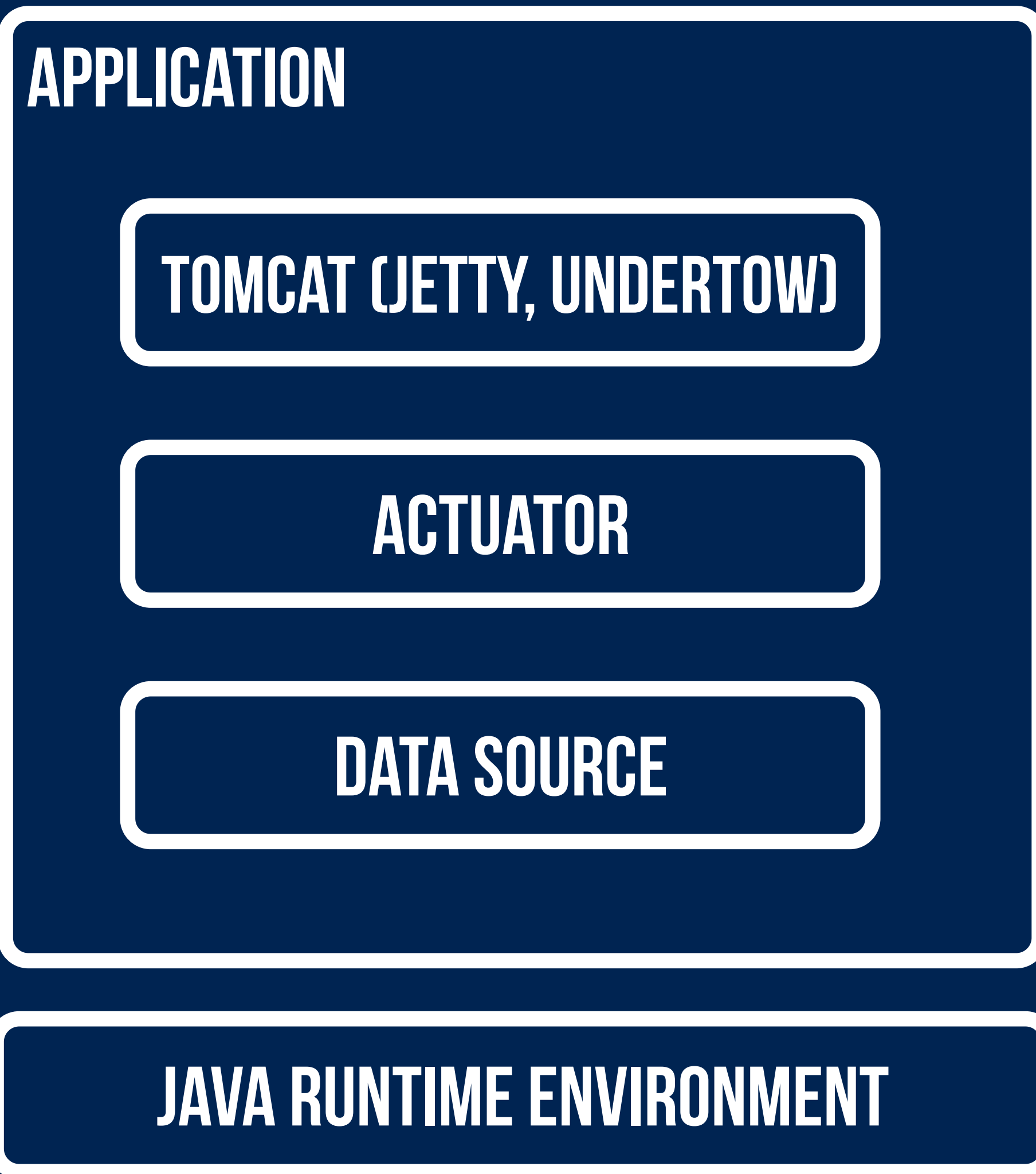


37%
INTERNET
TRAFFIC



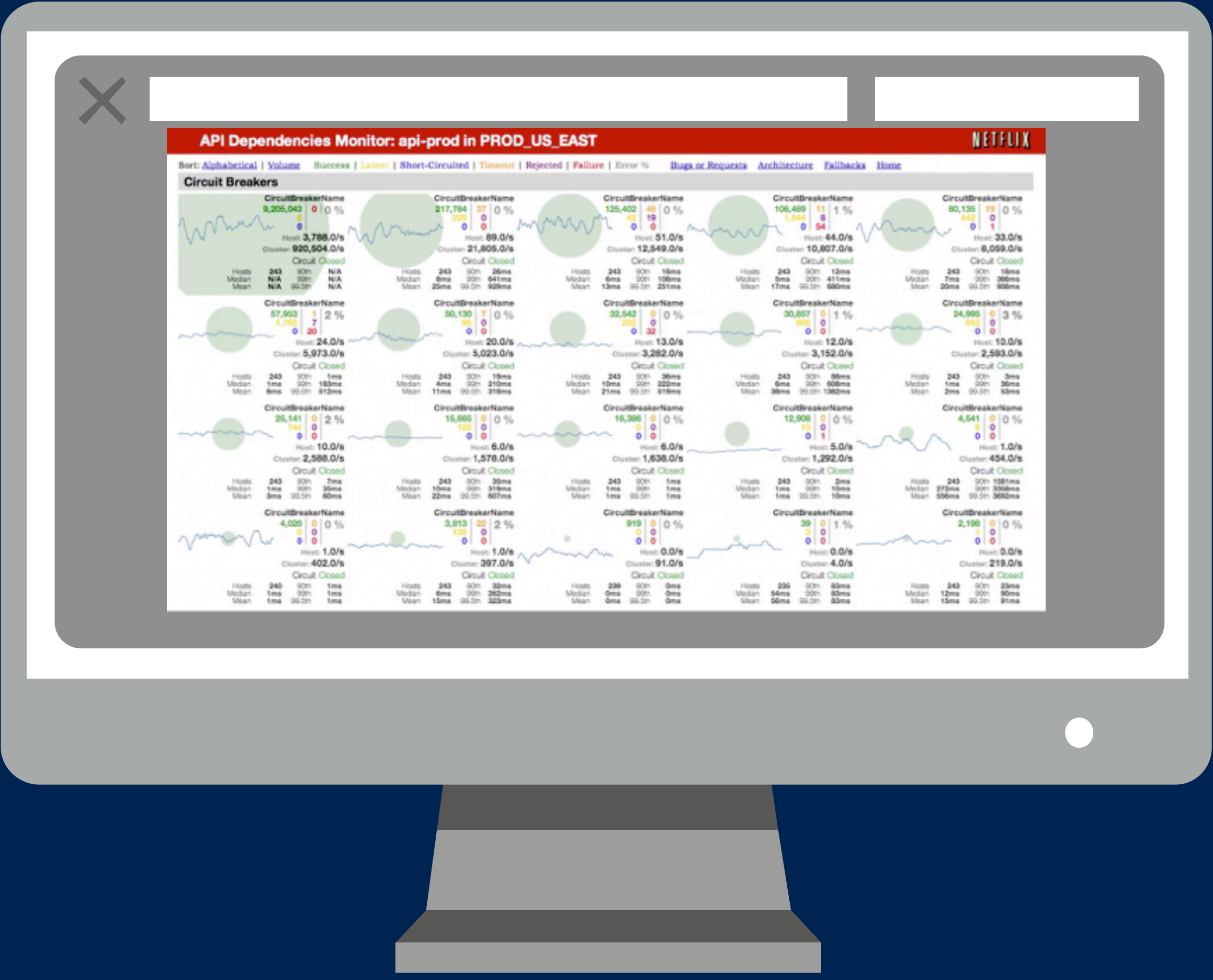
NETFLIX





JAVA -JAR MYAPPLICATION.JAR

CIRCUIT BREAKER



FRAGILE

NO TIMEOUT

ROBUST

TIMEOUT

ANTI-FRAGILE

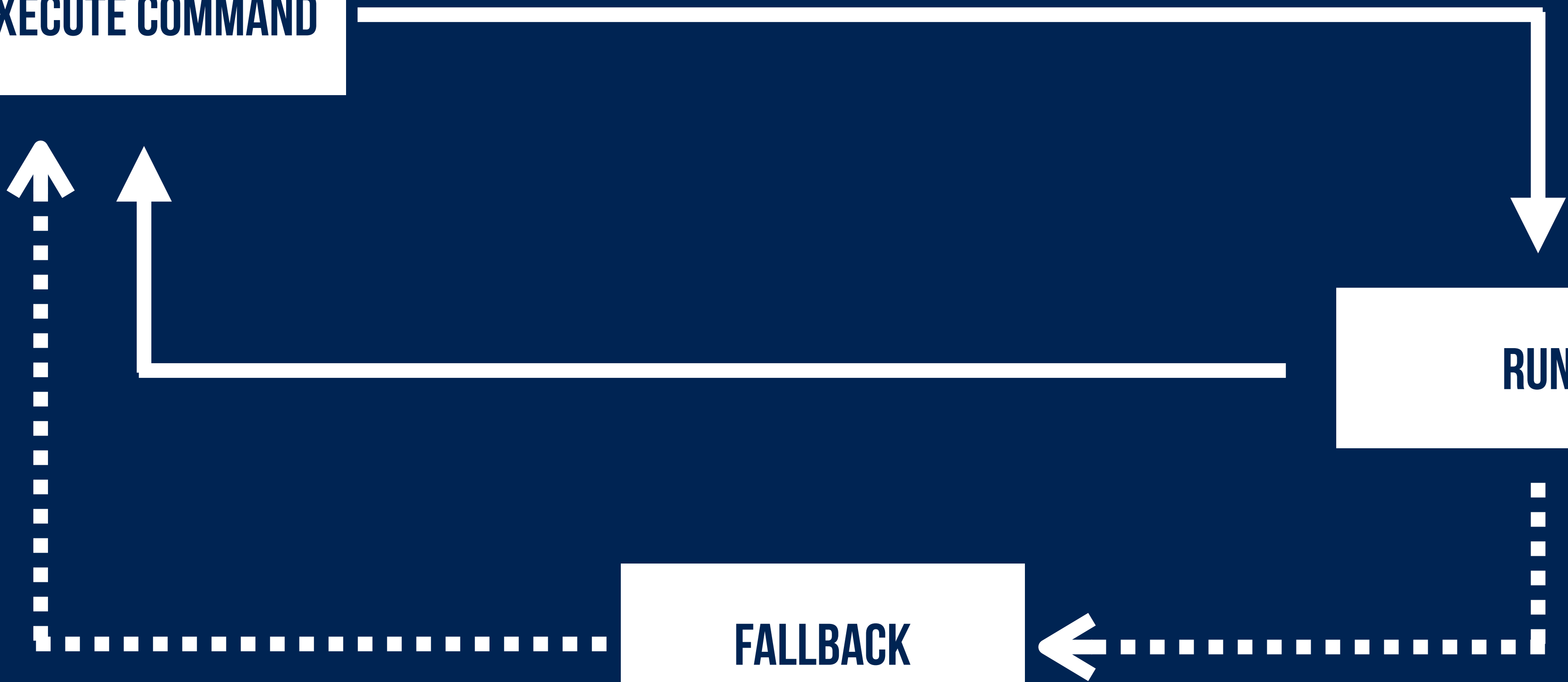
CIRCUIT-BREAKER

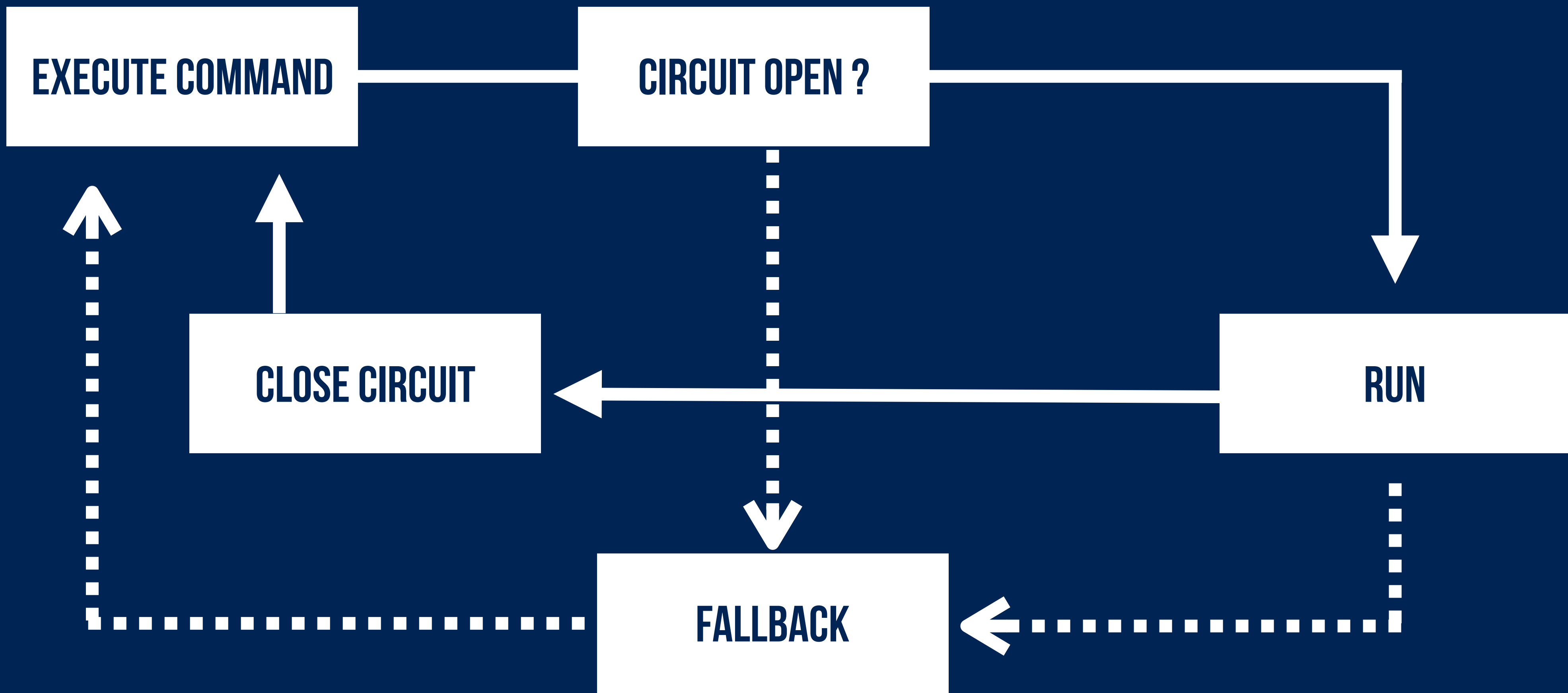
EXECUTE COMMAND

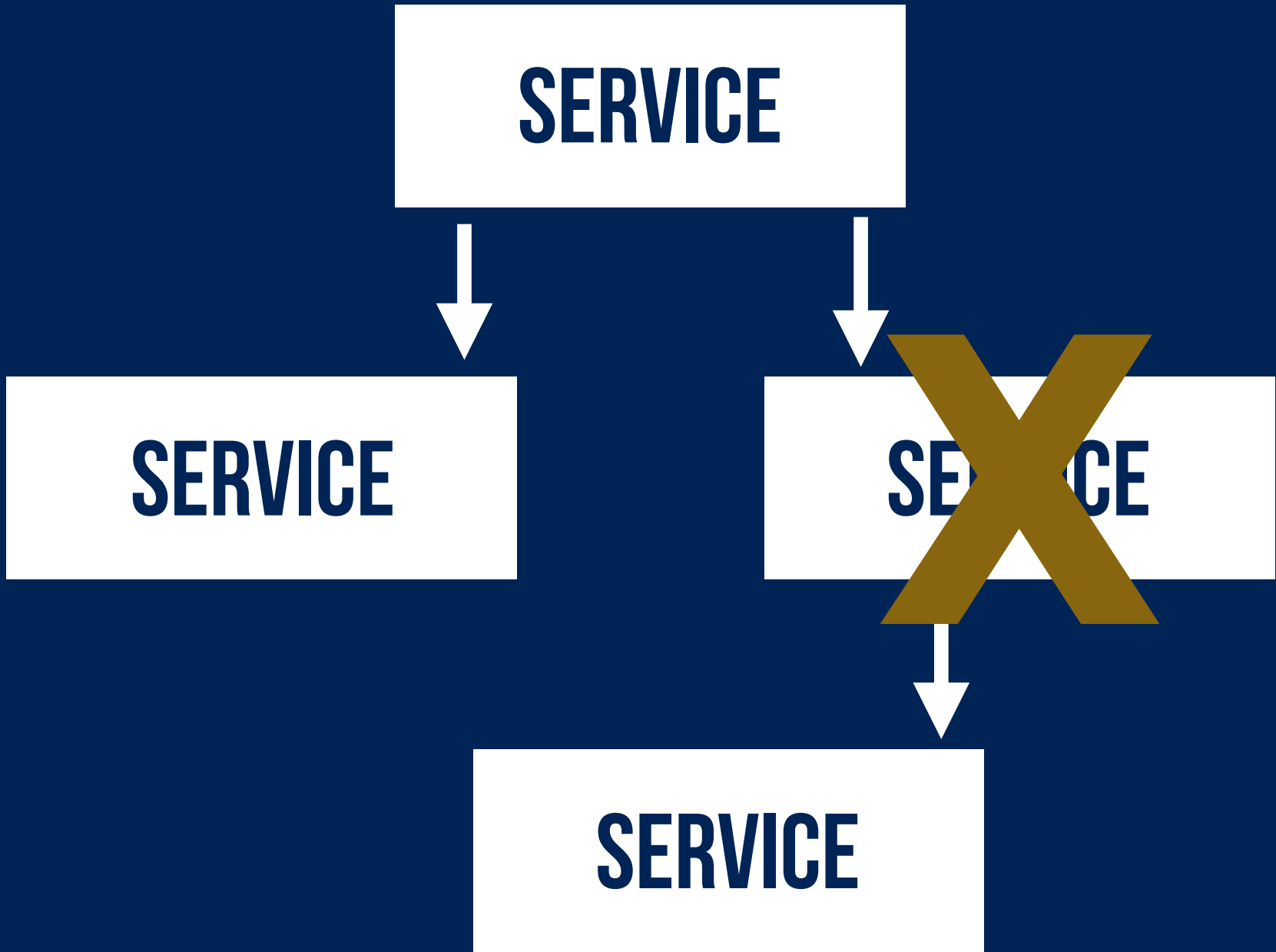
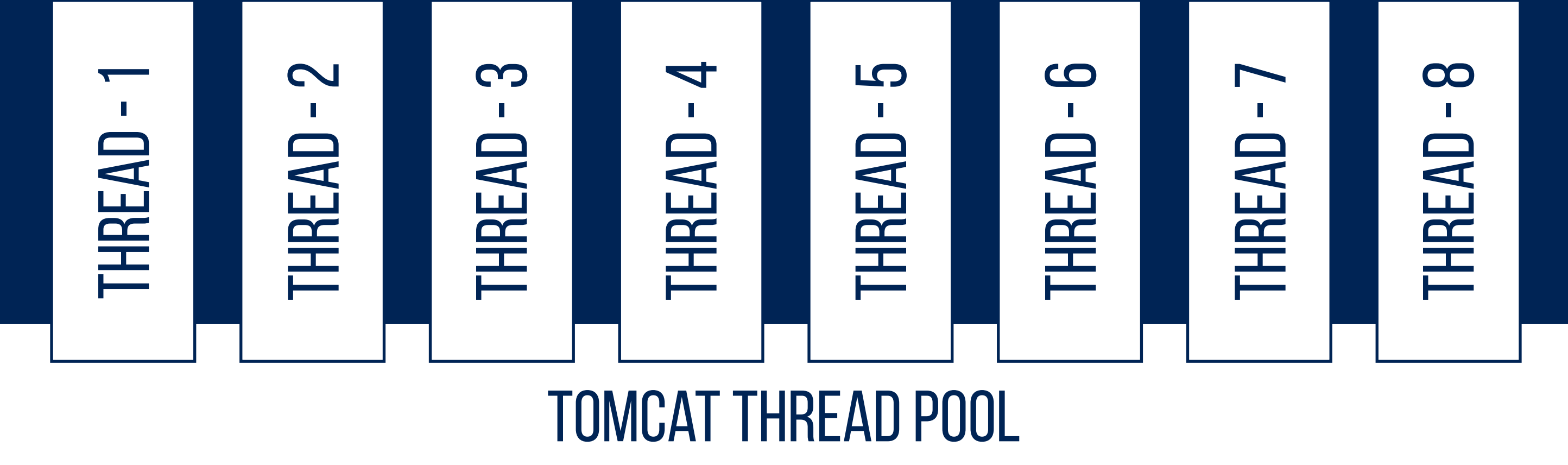


RUN

FALLBACK









COMMAND THREAD



COMMAND THREAD



COMMAND THREAD

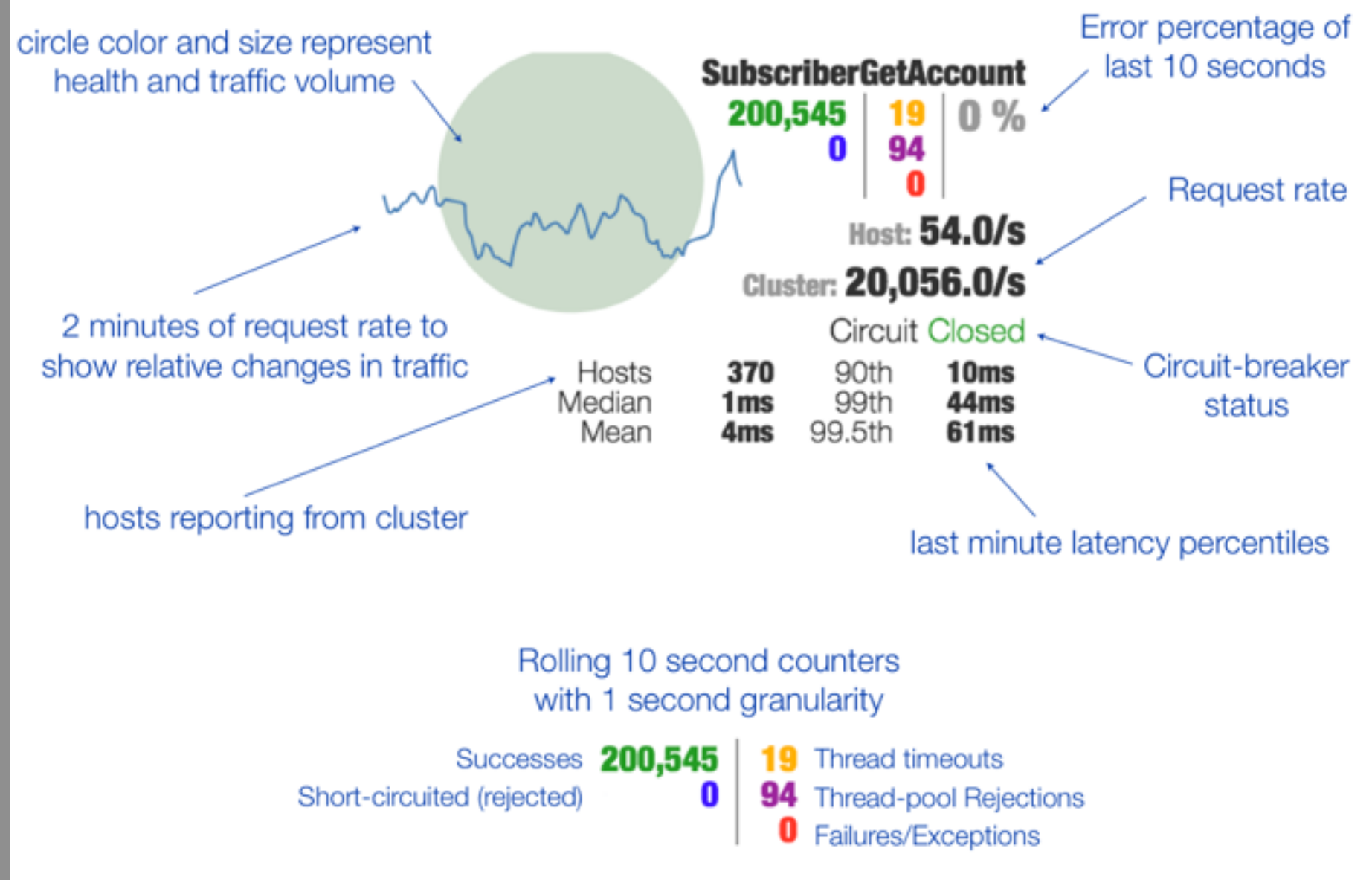
TRACE --- [HTTP-NIO-AUTO-1-EXEC-10] OUTSIDE COMMAND

TRACE --- [HYSTRIX-RESCURRENCYEXCHANGE-10] INSIDE COMMAND



THREAD LOCALS

```
@SpringCloudApplication  
public class SearchGateway {  
  
    @HystrixCommand(fallbackMethod = "fallback")  
    public List<SearchHit> search(String query) {  
        return ...;  
    }  
    public List<SearchHit> fallback() {  
        return Collections.emptyList();  
    }  
}
```



circle color and size represent health and traffic volume

Error percentage of last 10 seconds

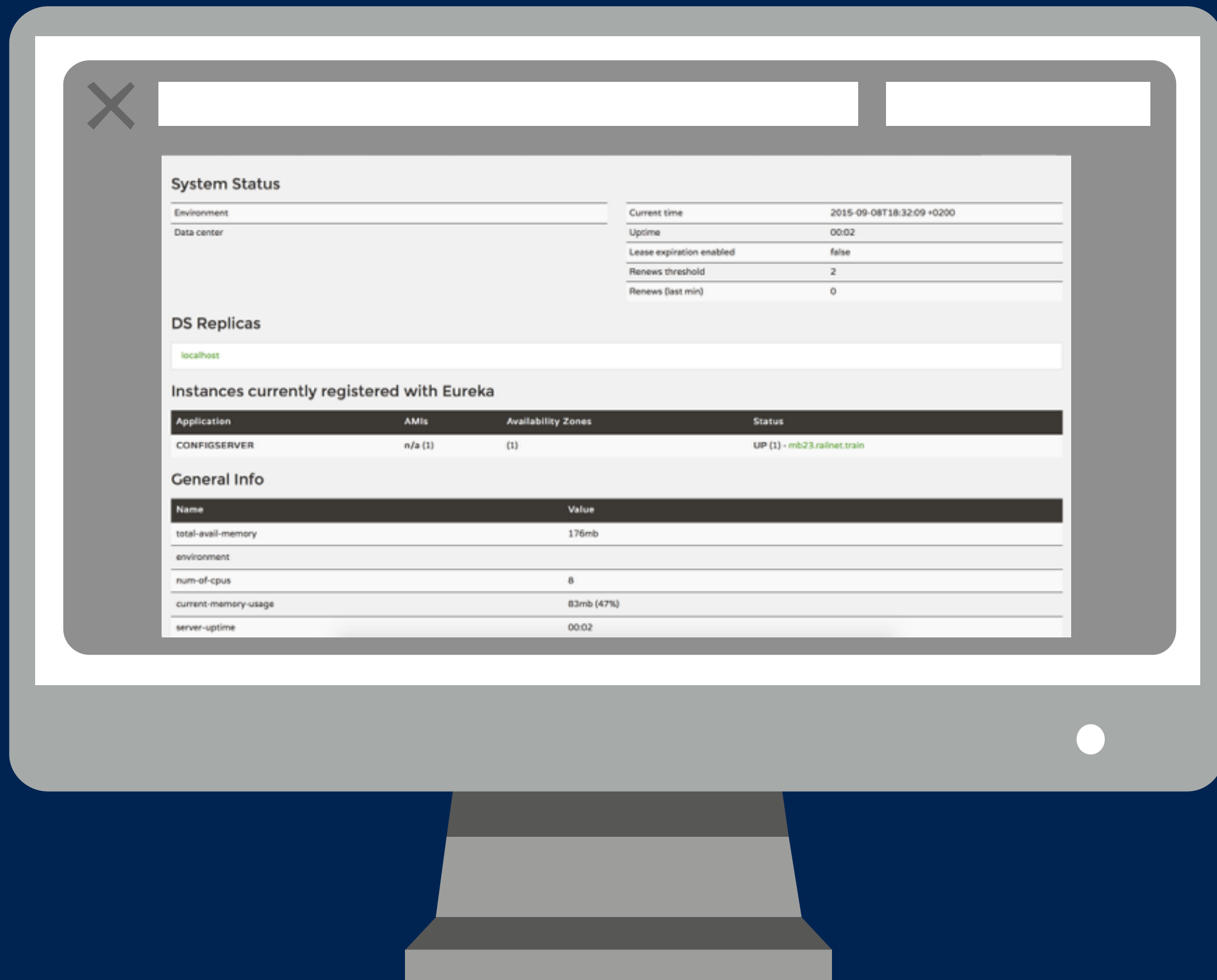
2 minutes of request rate to show relative changes in traffic

Request rate

hosts reporting from cluster

Circuit-breaker status

last minute latency percentiles



SERVICE REGISTRY



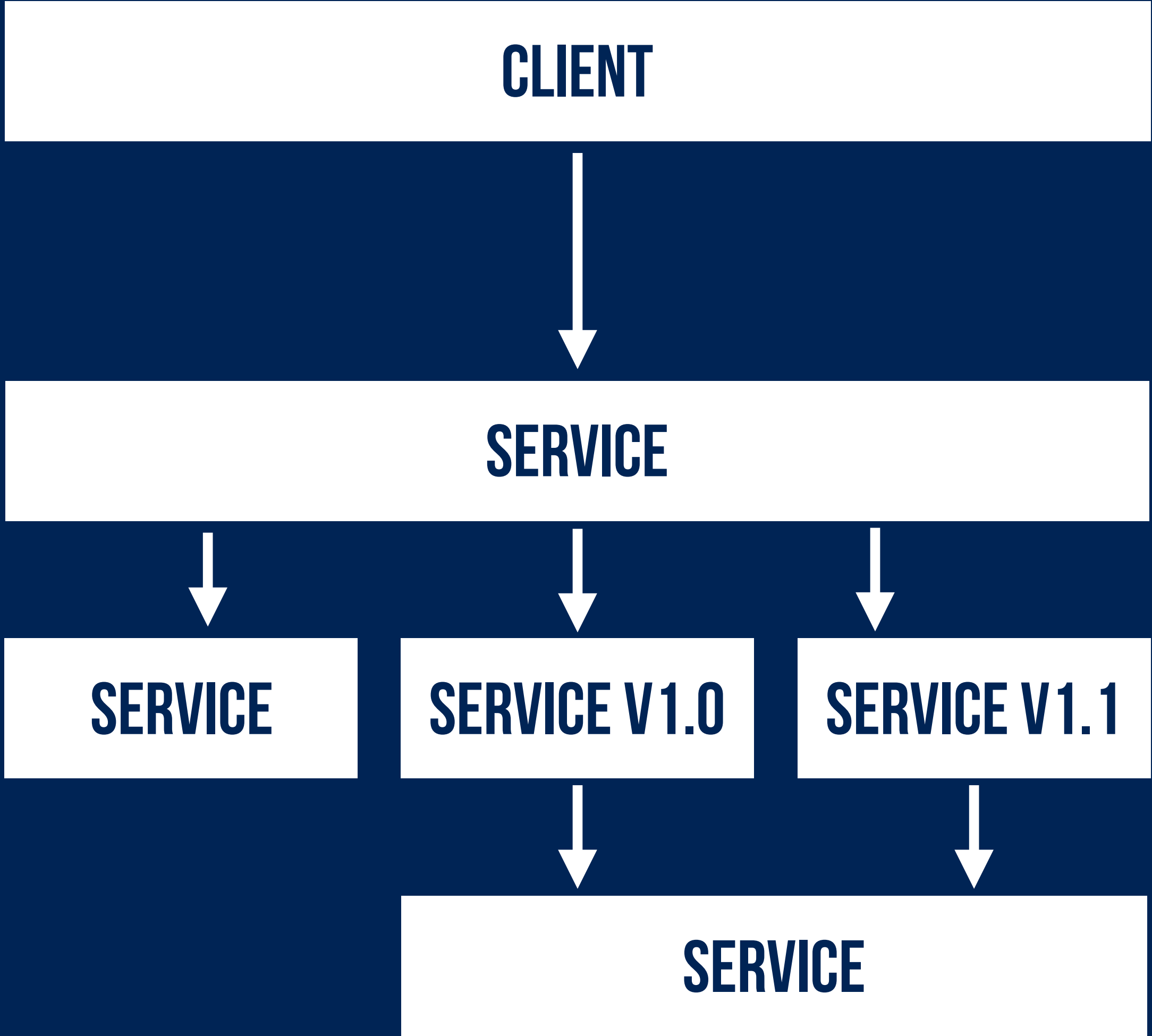
FRAGILE

POINT-TO-POINT

ROBUST

ANTI-FRAGILE

SERVICE-REGISTRY





NETFLIX

EUREKA

**AVAILABILITY
PARTITIONING**



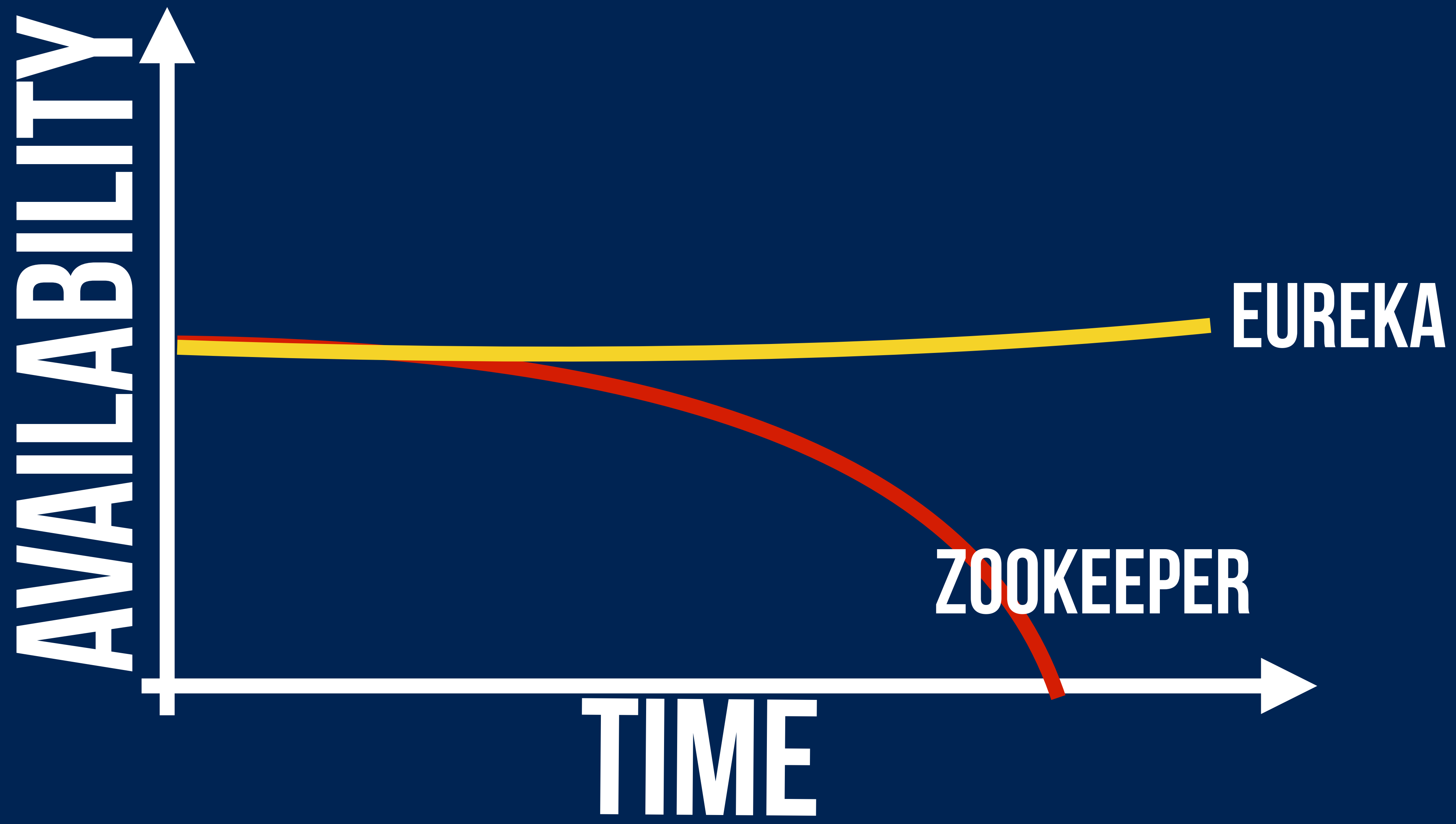
CONSUL

**CONSISTENCY
AVAILABILITY**

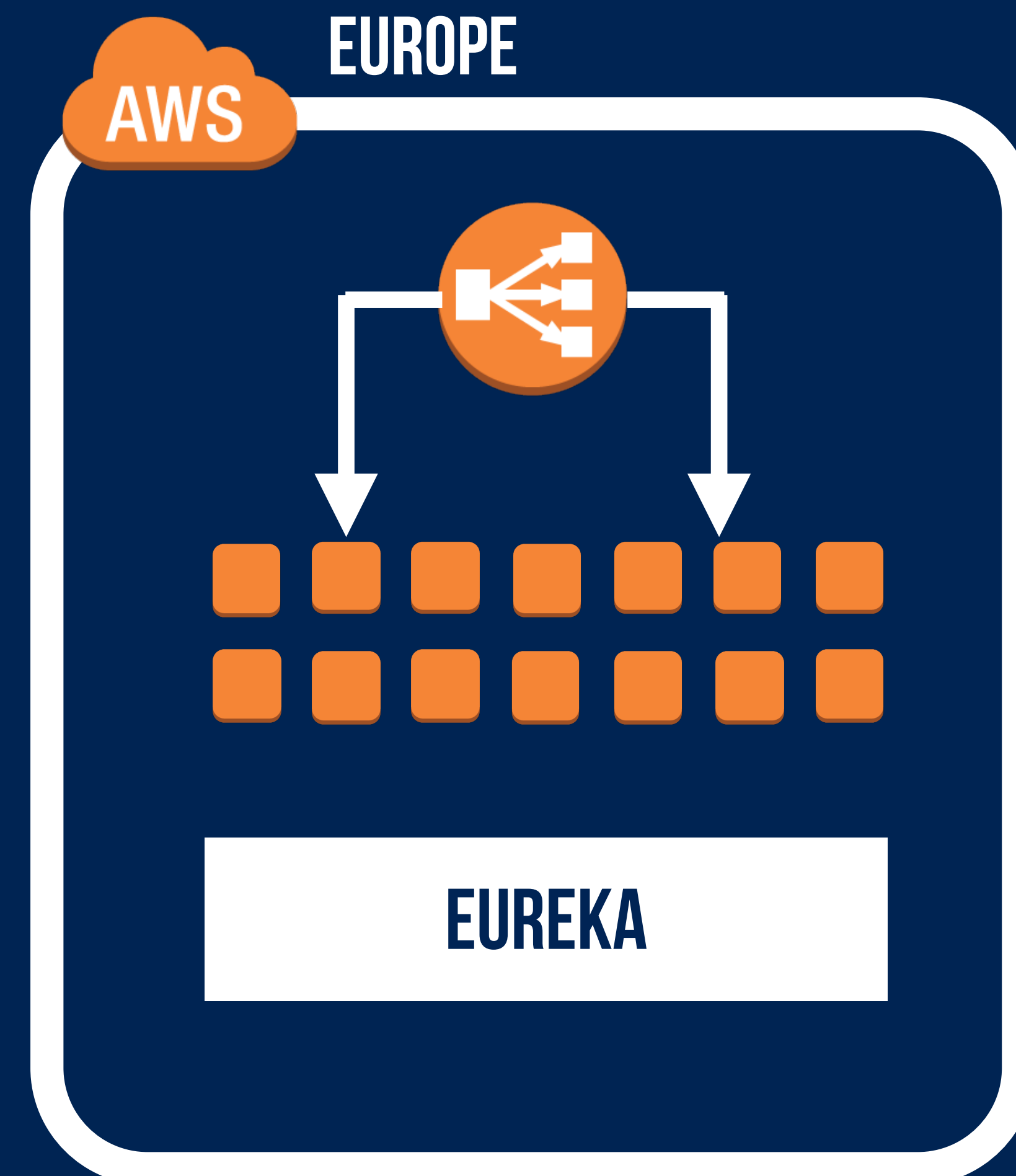
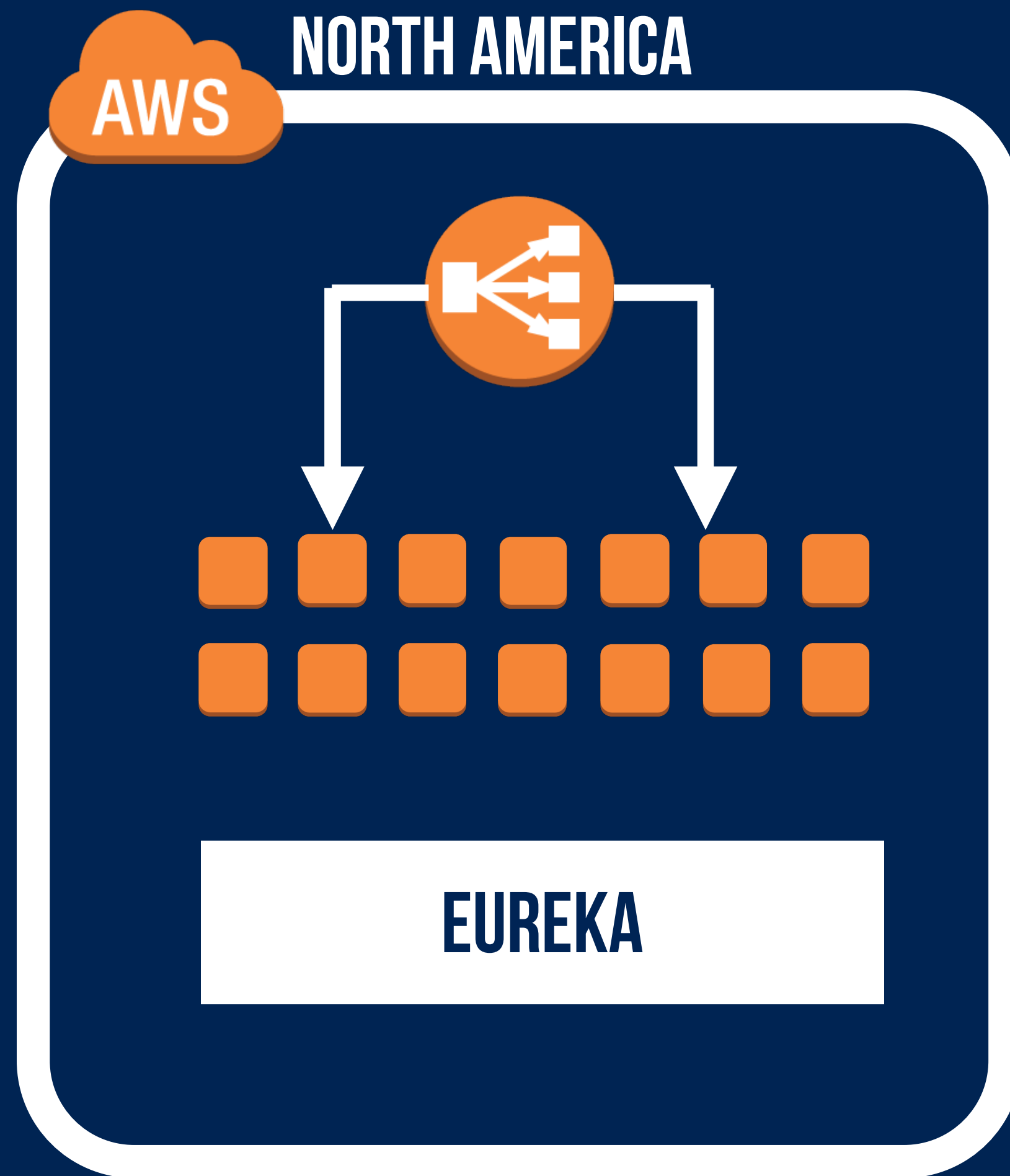


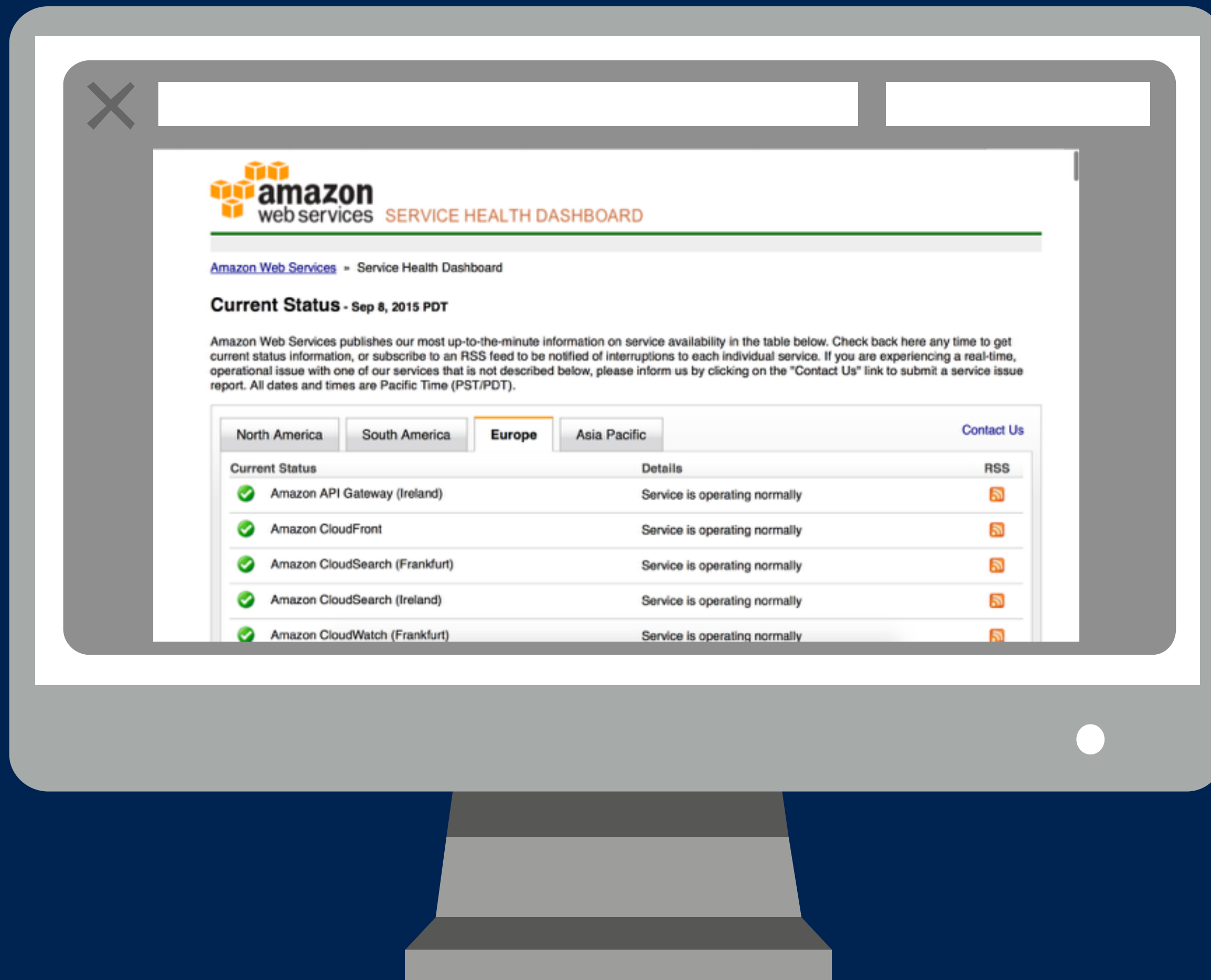
ZOOKEEPER

**CONSISTENCY
AVAILABILITY**

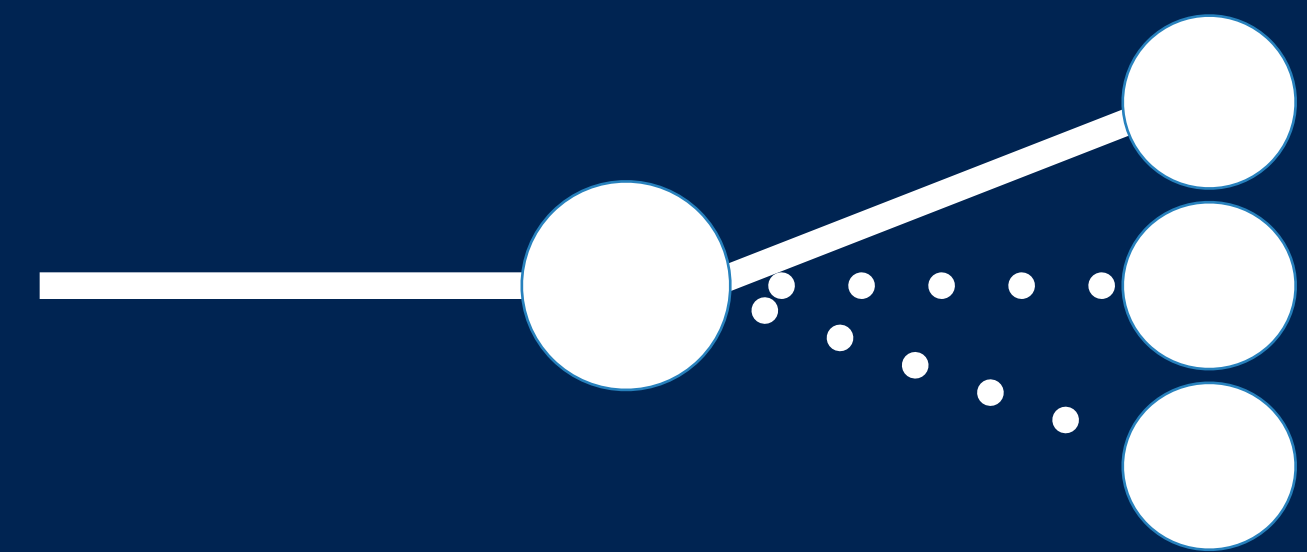


PARTIONING

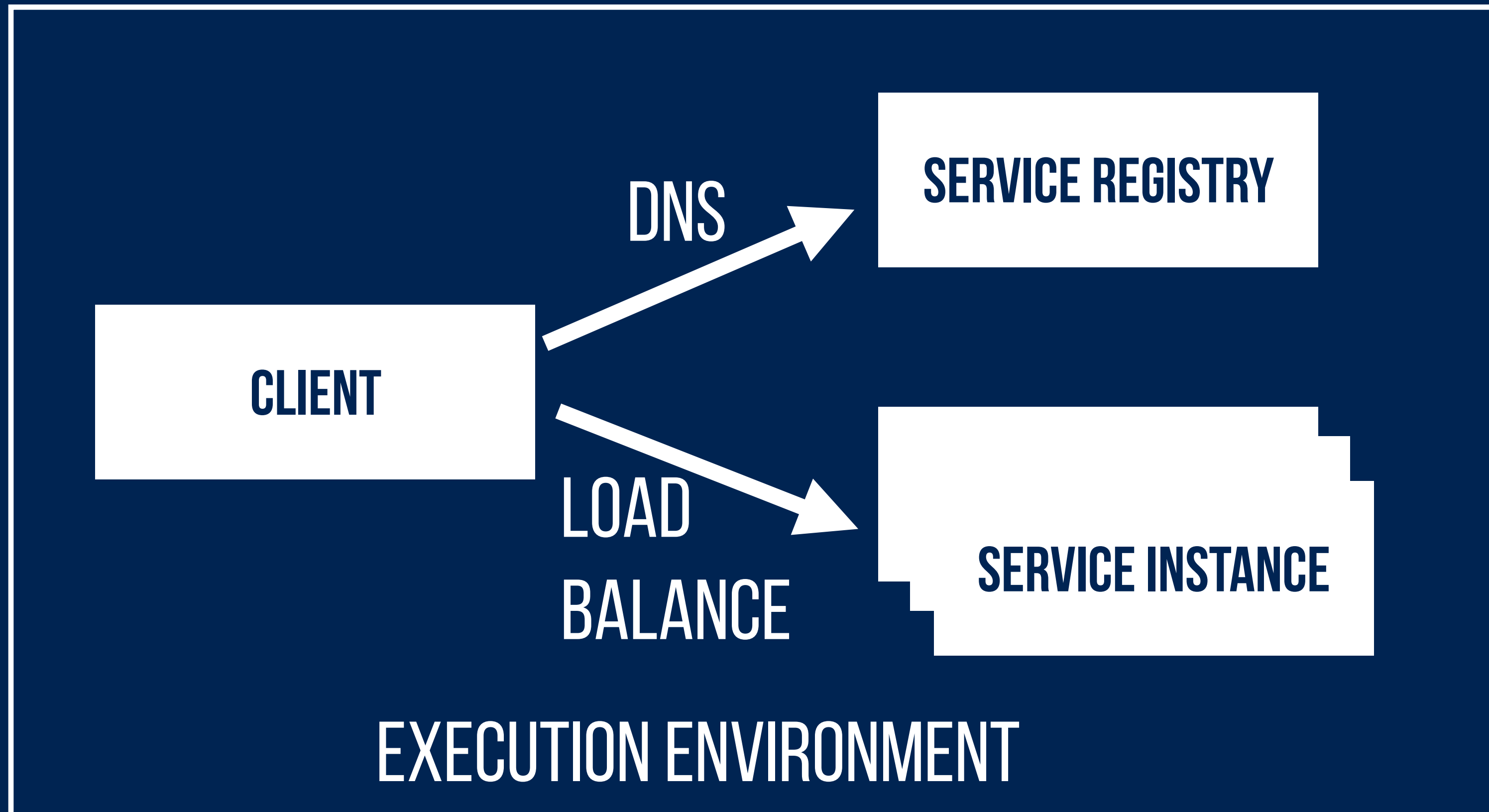




DISCOVERY



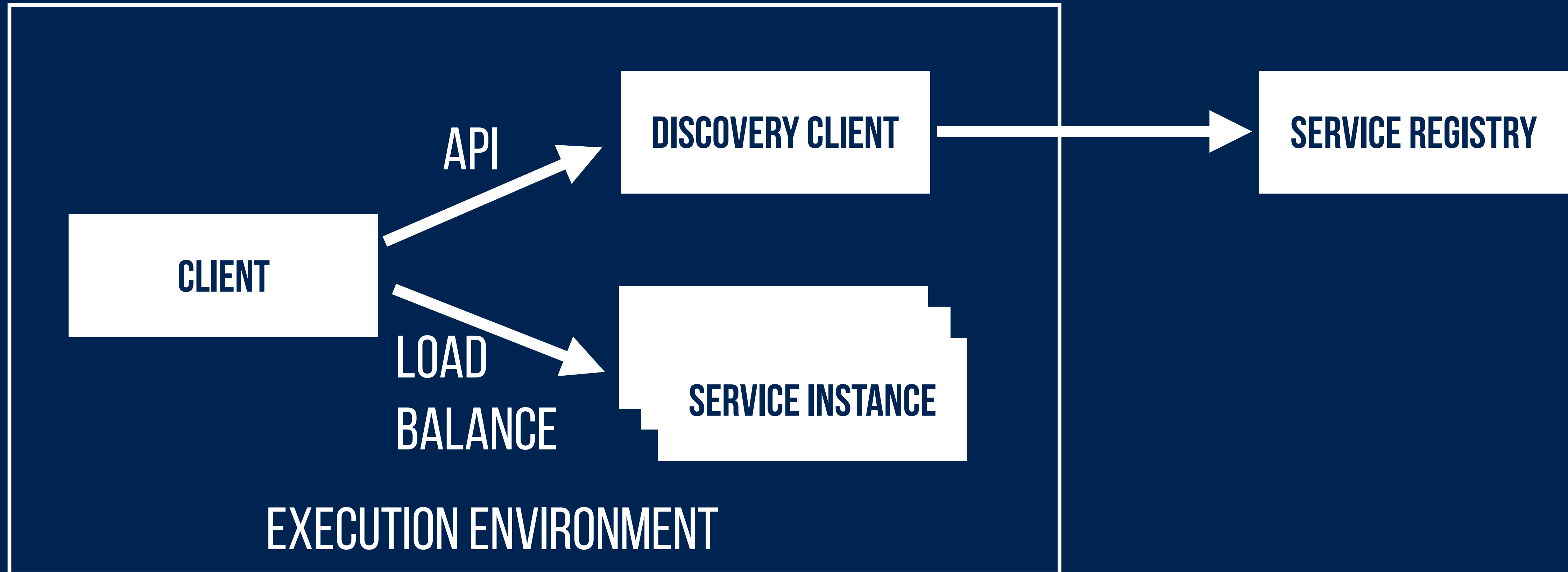
CLIENT DISCOVERY



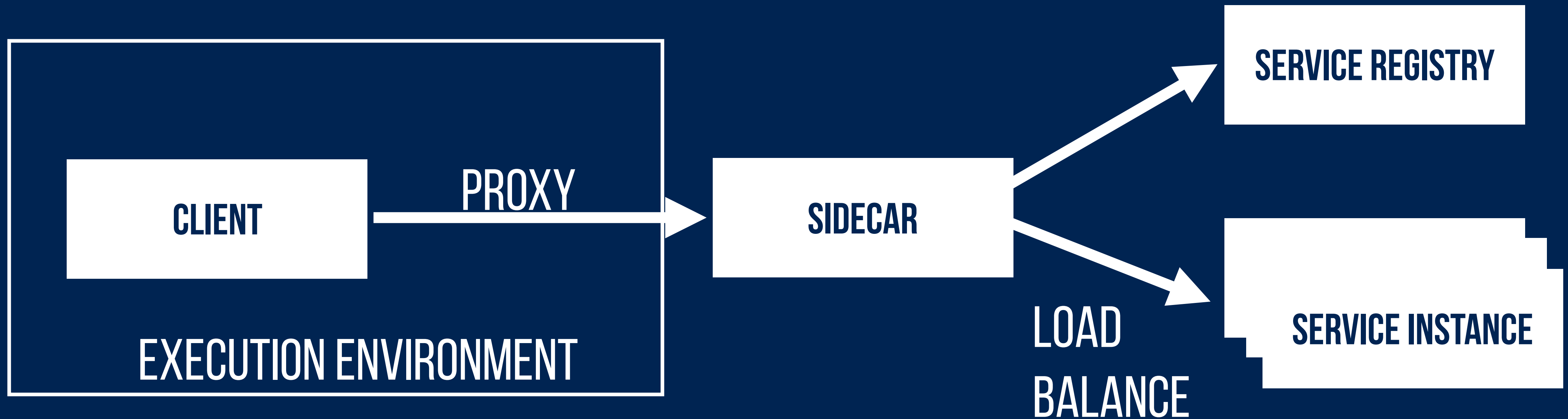
CONSUL

KUBERNETES

CLIENT DISCOVERY



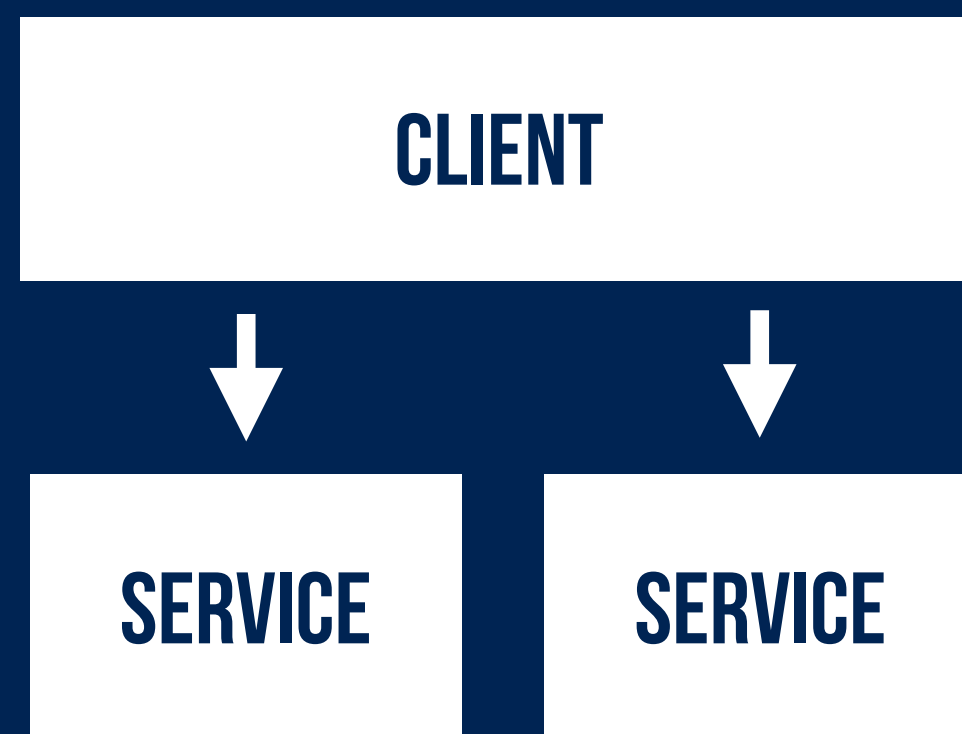
SERVER DISCOVERY



SERVER DISCOVERY

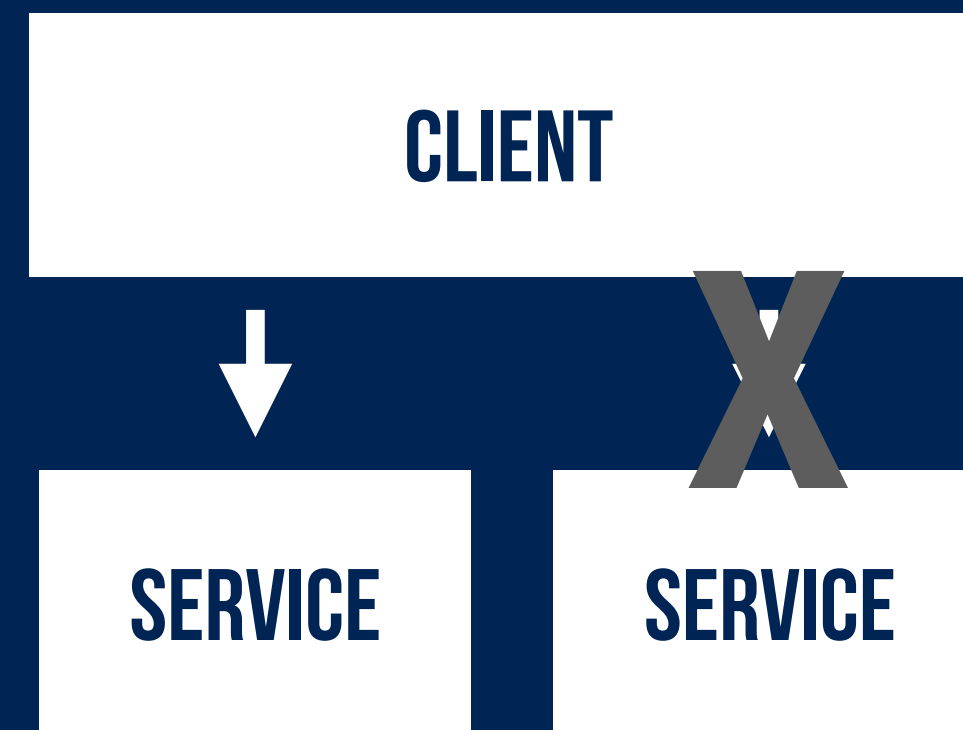


LOAD BALANCING



ROUND ROBIN

LINEAR



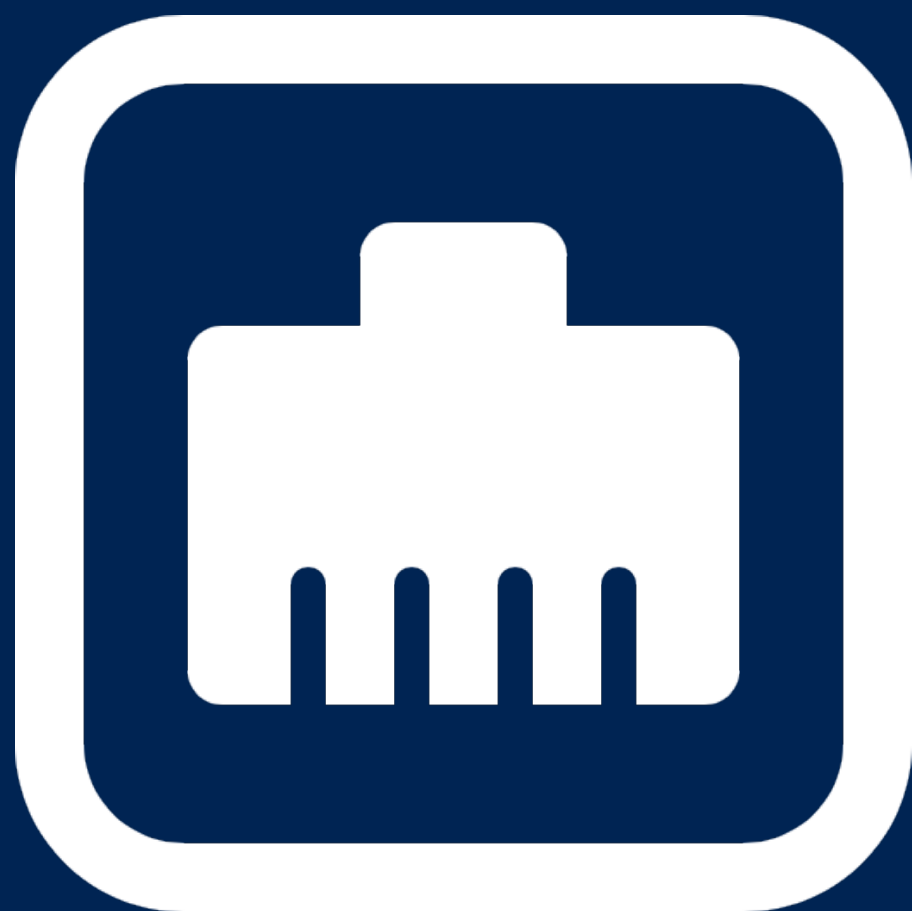
**AVAILABILITY
FILTERING**

NON-LINEAR



**WEIGHTED
RESPONSE TIME**

NON-LINEAR



ADD LATENCY

```
tc qdisc add dev eth0 root latency delay 1000ms 500ms
```

CORRUPT PACKAGES

```
tc qdisc add dev eth0 root netem corrupt 5%
```

DROP PACKAGES

```
tc qdisc add dev eth0 root netem loss 7% 25%
```

BLOCK DNS

```
iptables -A INPUT -p tcp -m tcp --dport 53 -j DROP
```

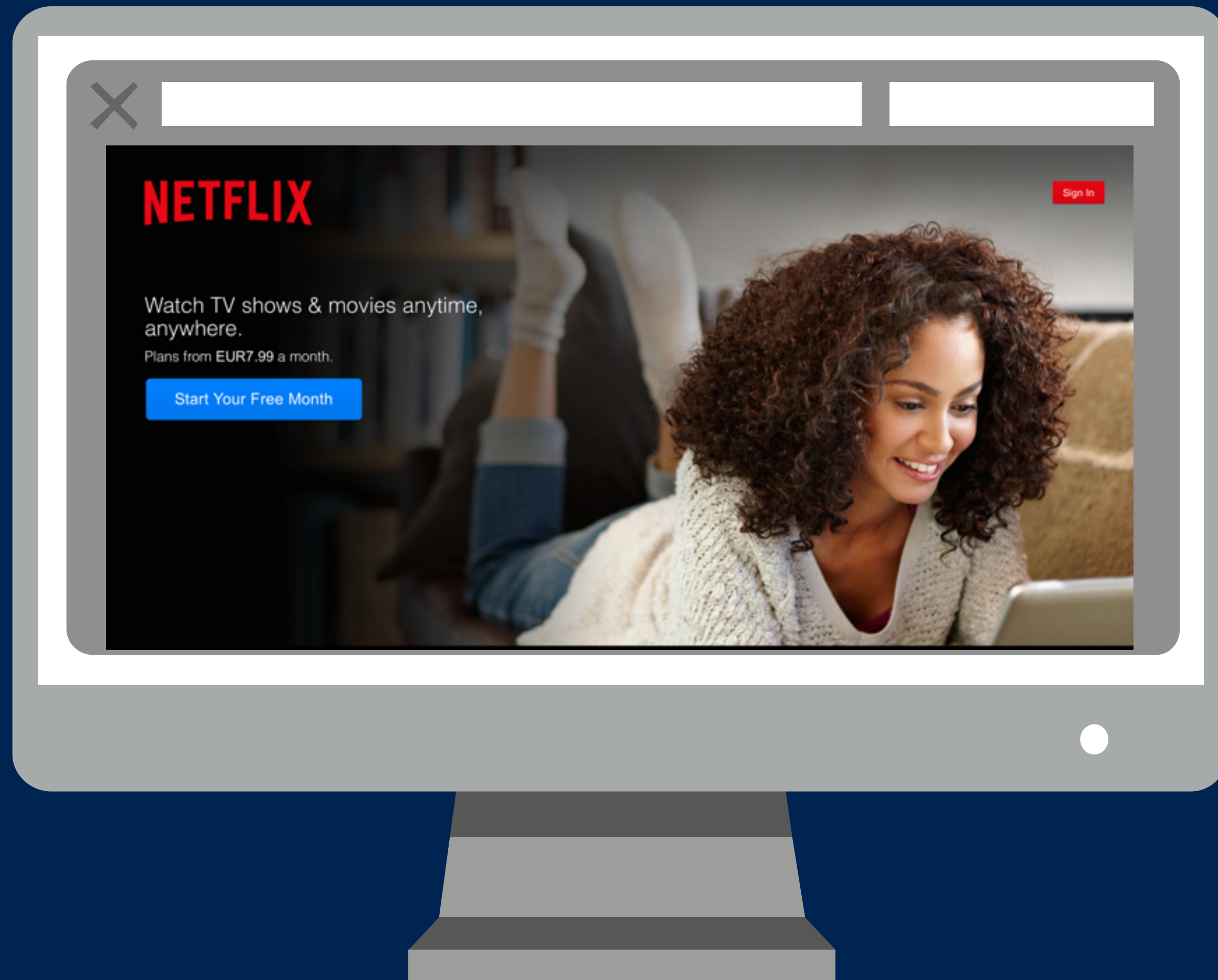


SIMULATE HEAVY IO

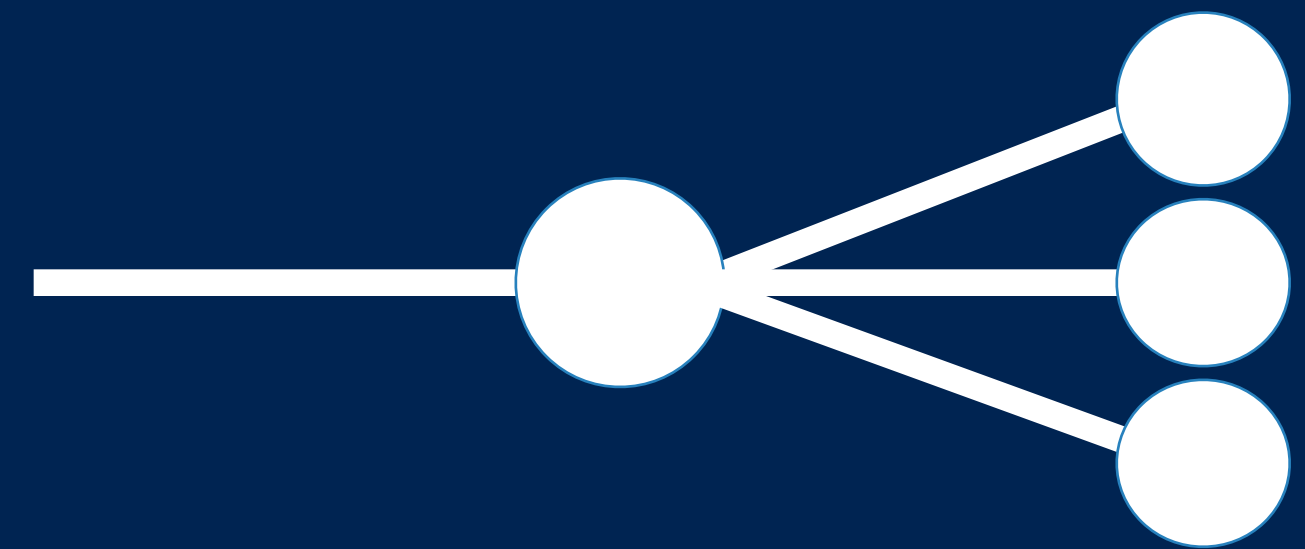
```
while true;  
  
do  
    dd if=/dev/urandom of=/burn bs=1M count=1024 iflag=fullblock  
done
```

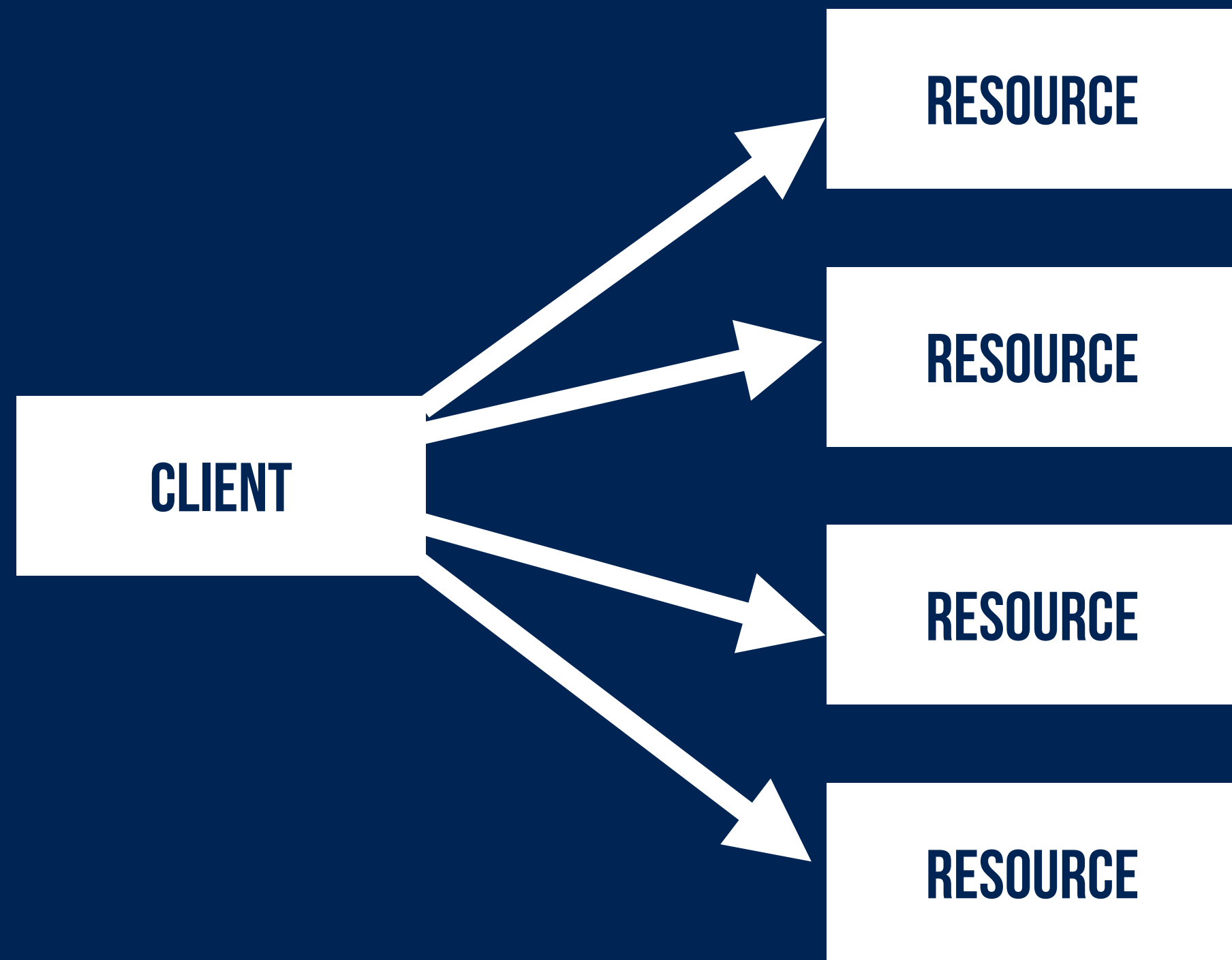
BURN CPU

```
while true;  
do openssl speed;  
done
```

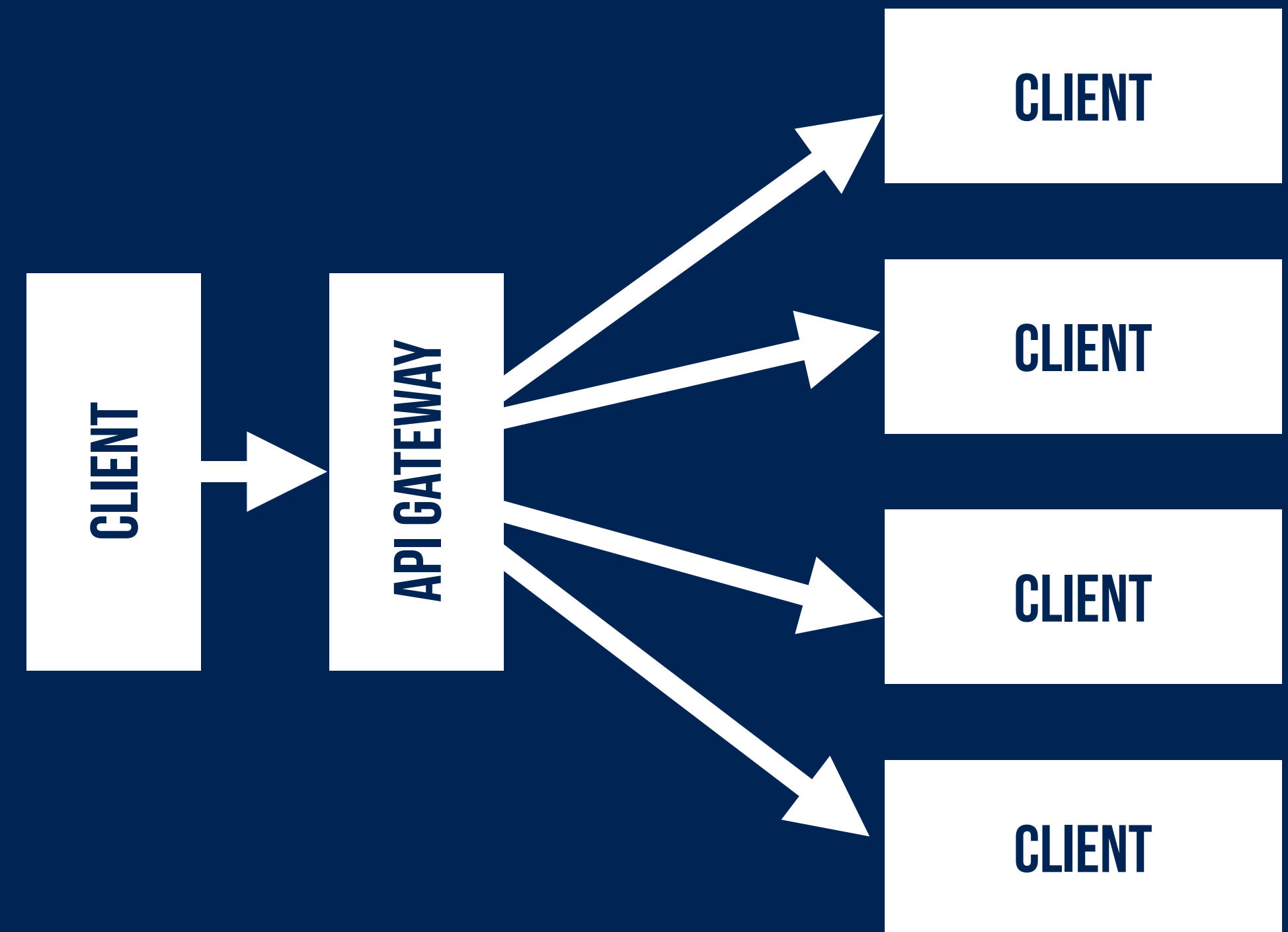


API GATEWAY



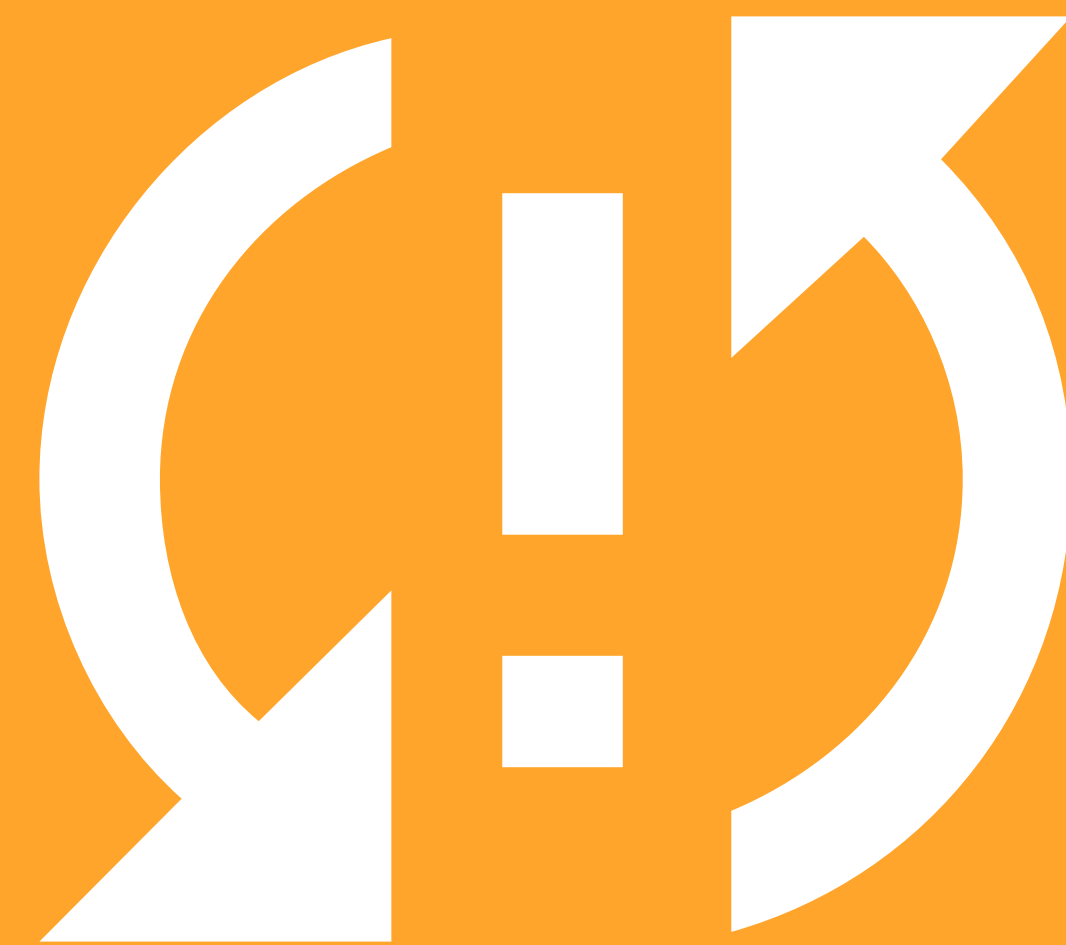


(C OR R1 OR R2 OR R3 OR R4)



(C OR A) AND (R1 OR R2 OR R3 OR R4)

**RUNNING
DISTRIBUTED
ARCHITECTURES**

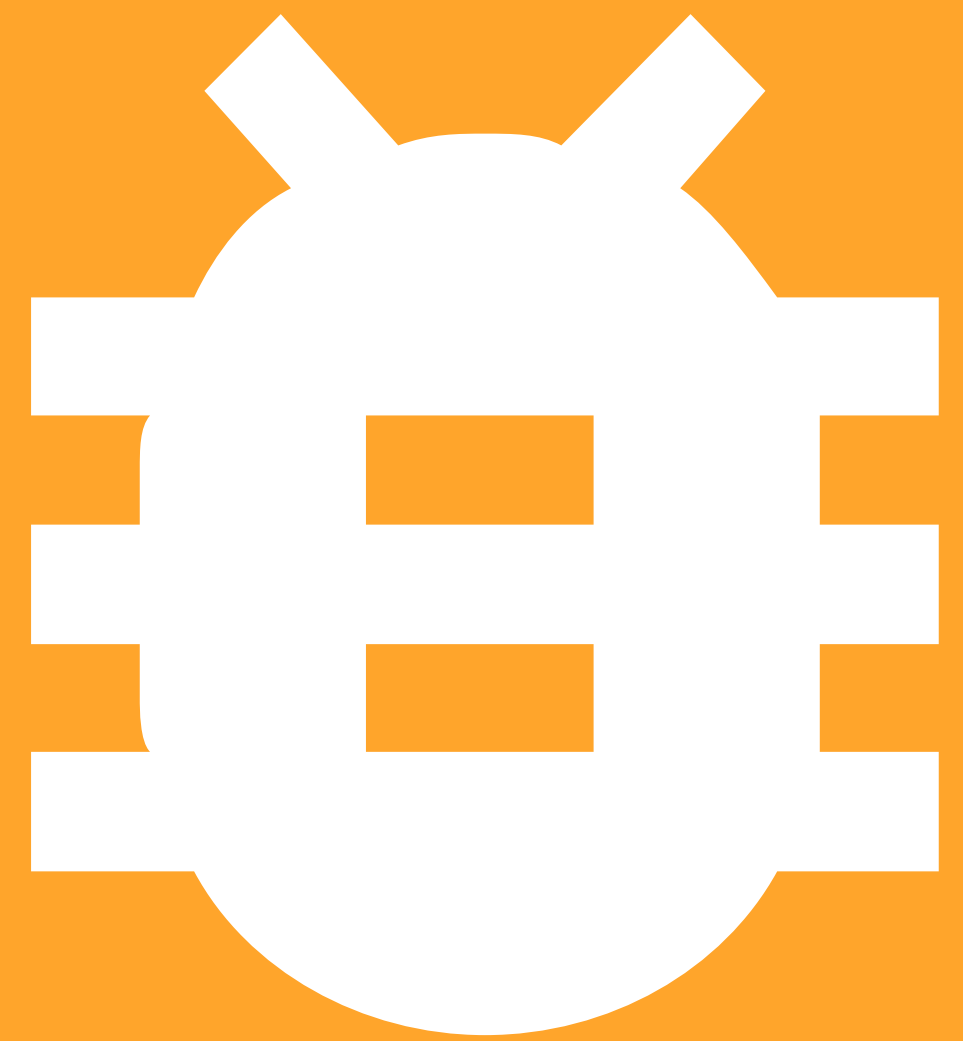




LOAD



PERFORMANCE

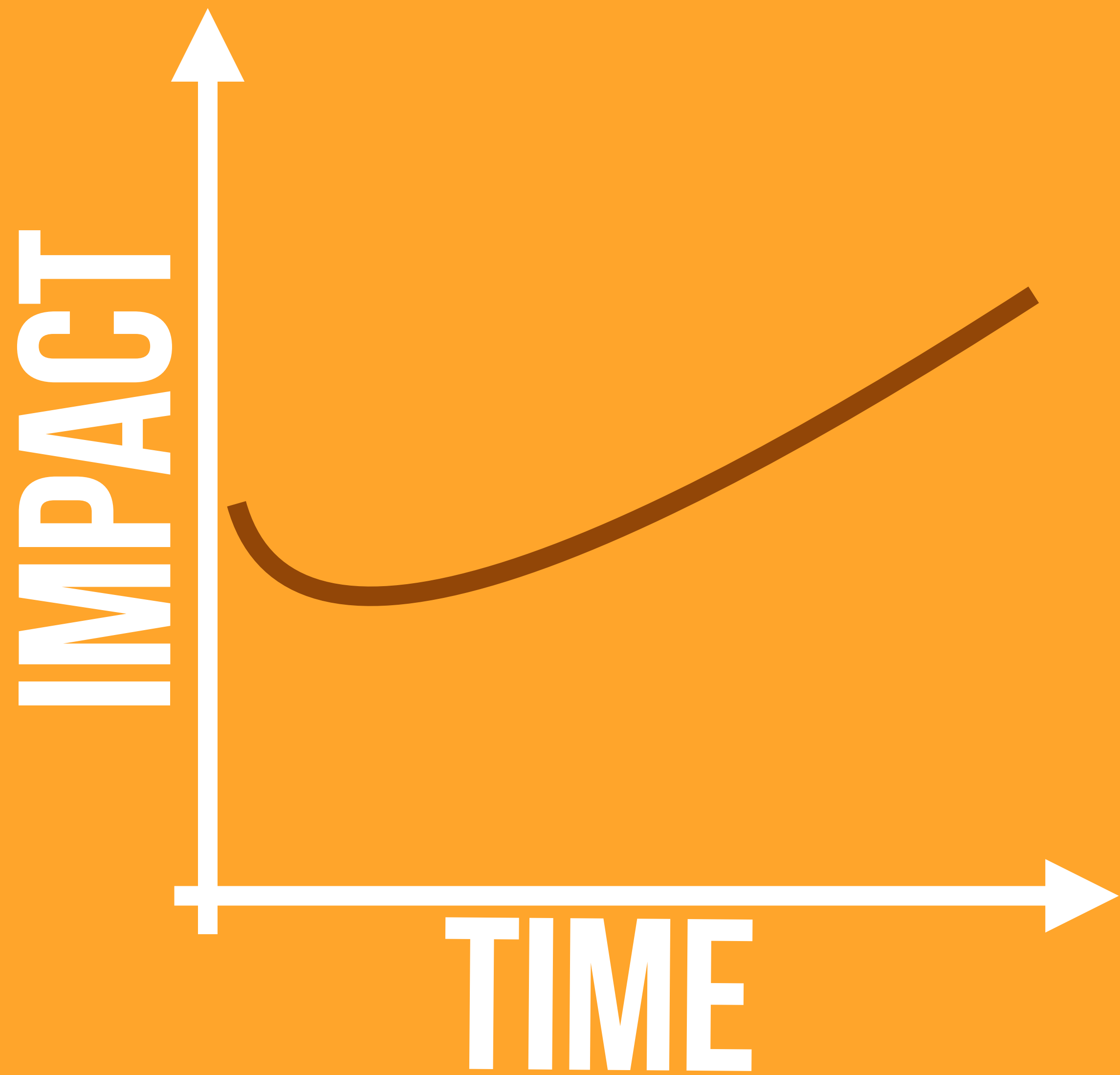


STRESS

ROBUST

AT 12:24 PM PACIFIC TIME ON DECEMBER 24 **NETWORK TRAFFIC** STOPPED ON A FEW ELBS..... AT AROUND 3:30 PM ON DECEMBER 24, NETWORK TRAFFIC STOPPED ON ADDITIONAL ELBSNETFLIX IS DESIGNED TO HANDLE FAILURE OF ALL OR PART OF A SINGLE AVAILABILITY ZONE IN A REGION AS WE RUN ACROSS THREE ZONES AND OPERATE WITH NO LOSS OF FUNCTIONALITY ON TWO. WE ARE **WORKING ON WAYS OF EXTENDING OUR RESILIENCY** TO HANDLE PARTIAL OR COMPLETE REGIONAL OUTAGES.





HORMESIS



ANTI-FRAGILE

ON SUNDAY, AT 2:19AM PDT, THERE WAS A BRIEF **NETWORK DISRUPTION** THAT IMPACTED.....

SO, WHEN THE **NETWORK DISRUPTION** OCCURRED ON SUNDAY MORNING, AND A NUMBER OF STORAGE SERVERS SIMULTANEOUSLY REQUESTED THEIR MEMBERSHIP DATA,.....

BY 2:37AM PDT, THE **ERROR RATE** IN CUSTOMER REQUESTS TO DYNAMODB HAD RISEN FAR BEYOND ANY LEVEL EXPERIENCED IN THE LAST 3 YEARS.....

AFTER SEVERAL FAILED ATTEMPTS AT ADDING CAPACITY, AT 5:06AM PDT, WE DECIDED TO **PAUSE REQUESTS** TO THE METADATA SERVICE.

ANTI-FRAGILE

DESPITE BEING RUN ENTIRELY FROM AWS' CLOUD PLATFORM THE ONLINE STREAMING GIANT NETFLIX REPORTS A QUICK RECOVERY FROM SUNDAY'S DISRUPTION - **DEMONSTRATING THE IMPORTANCE OF ITS APPROACH OF BUILDING CLOUD-BASED SYSTEMS TO "FAIL"**.

AWS:REBOOT

CASSANDRA

2700+ NODES

218 REBOOTED

22 DEAD

THANKS

AGIM EMRULI - MIMACOM

@AEMRULI