

# Application Snapshots Using Consistency Groups

Ravikumar Alluboyina

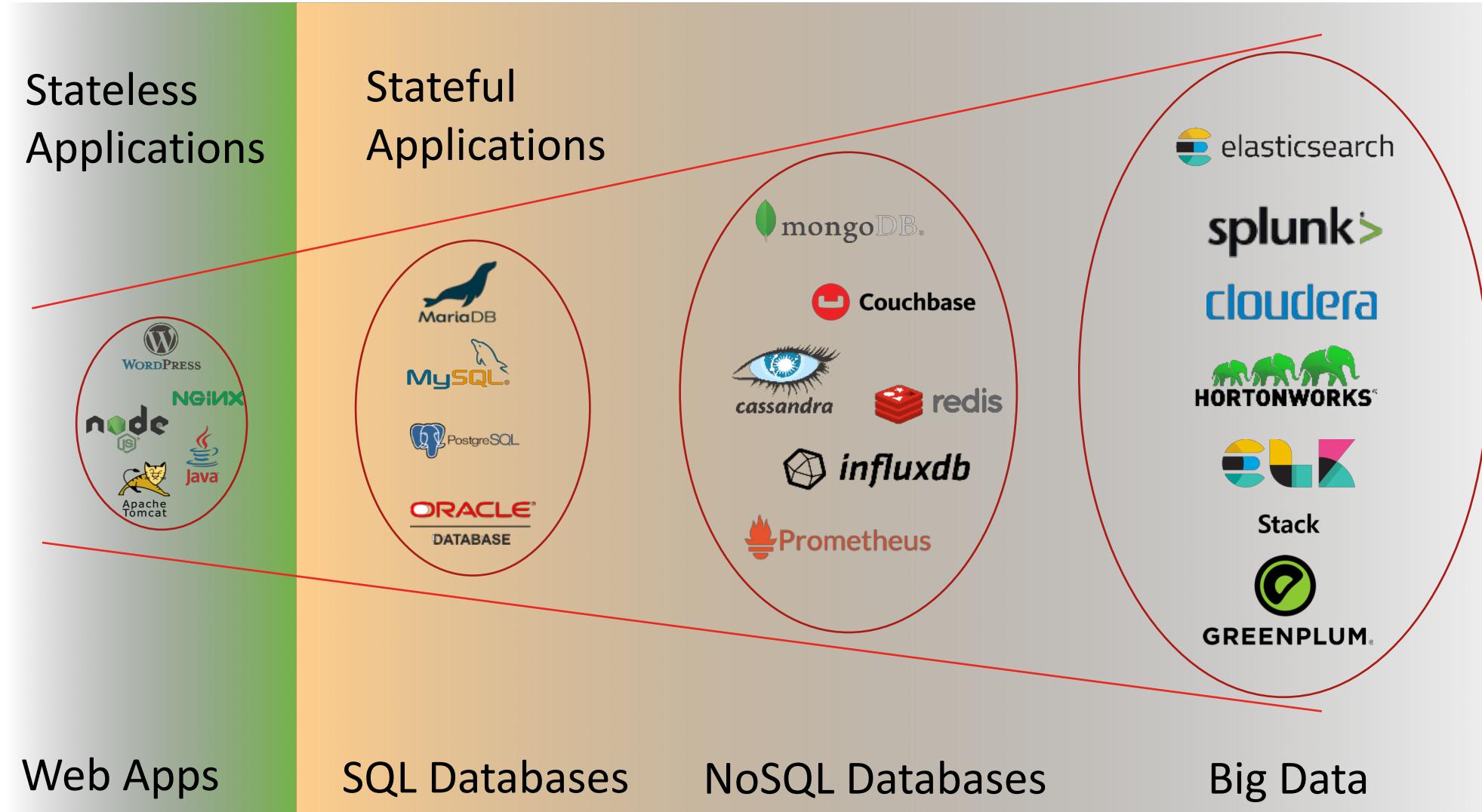
Senior Product Architect, Robin.io  
<https://get.robin.io>



# Agenda

- › Kubernetes and Application landscape
- › Databases and their IO Patterns
- › Application Snapshots and challenges
- › Consistency Groups
- › Q &A

# Spectrum of Applications

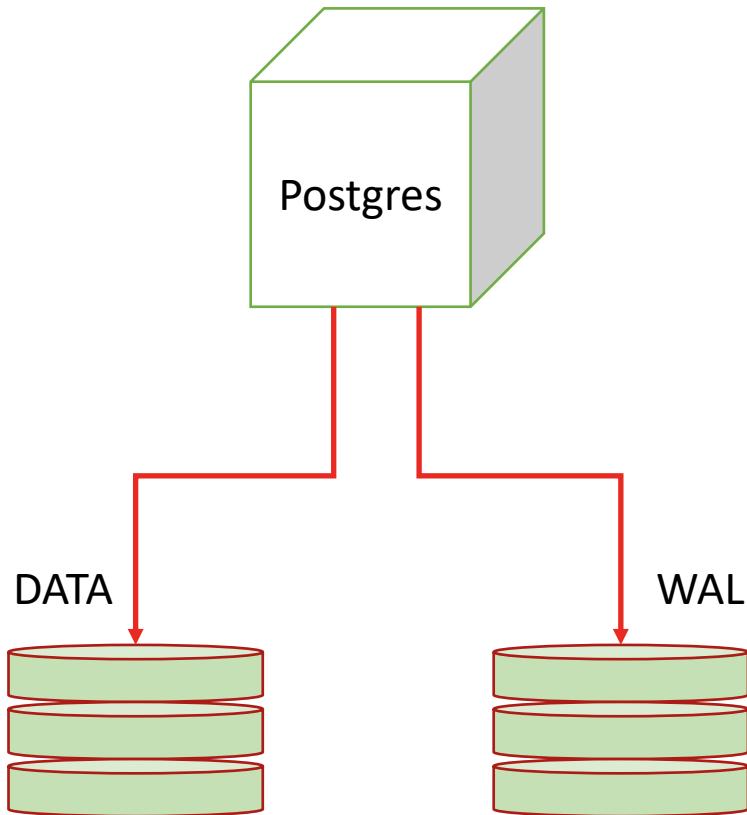


# Databases and the IO Patterns

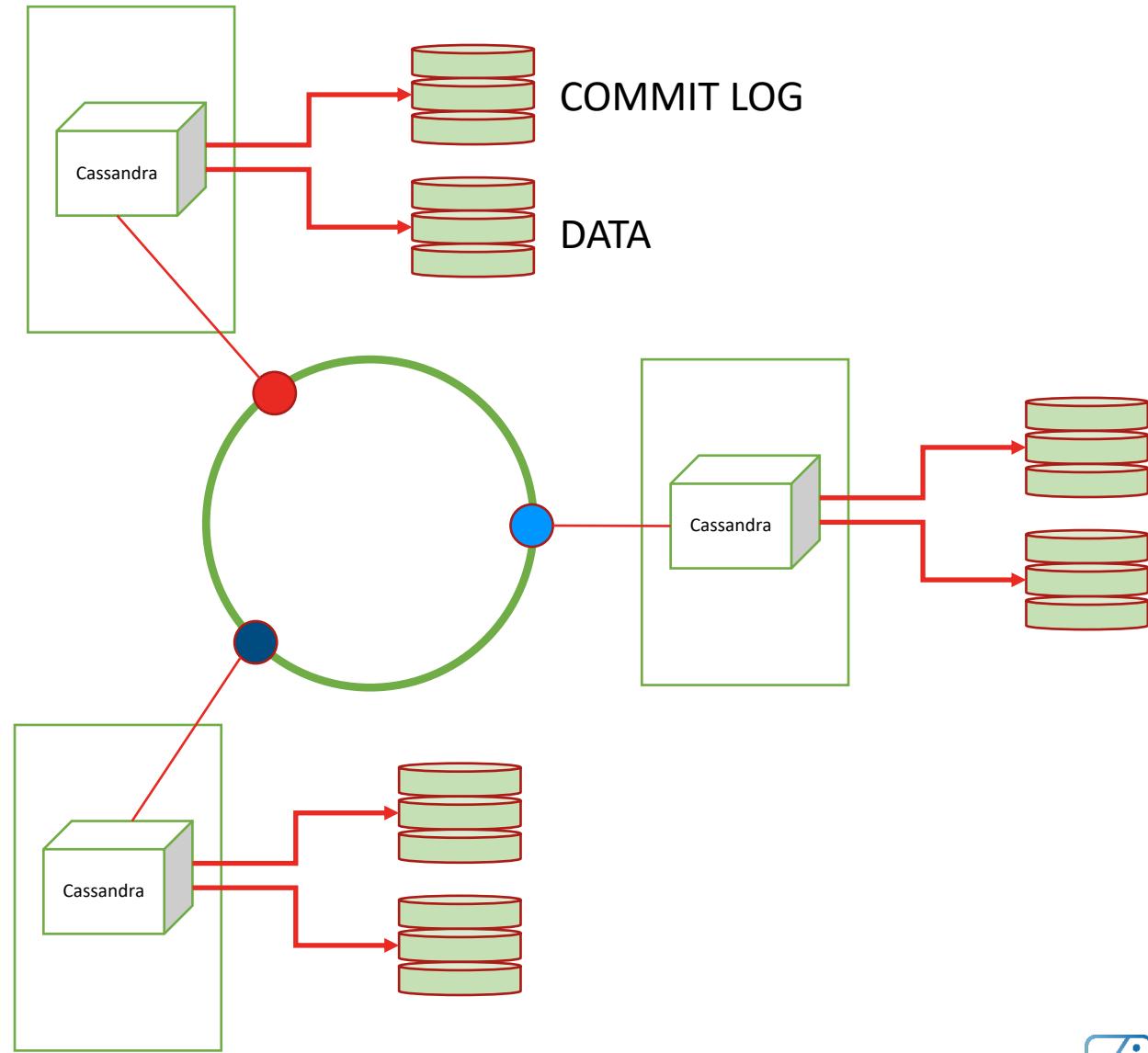
# Databases

- › SQL databases
  - › Complete ACID Compliance
  - › Mostly monolithic architectures
  - › Row oriented databases
  - › OLTP workloads (Transactional workloads)
  - › Standardized SQL interface
- › NoSQL databases
  - › Tunable consistency
  - › Tunable durability guarantees
  - › Scaleout architectures
  - › Column, Key-value or Document databases
  - › OLTP / OLAP (Mixed workloads)
  - › Non-standard client interfaces (CQL, Mongo Client, Redis Client etc)

# Database deployment Models

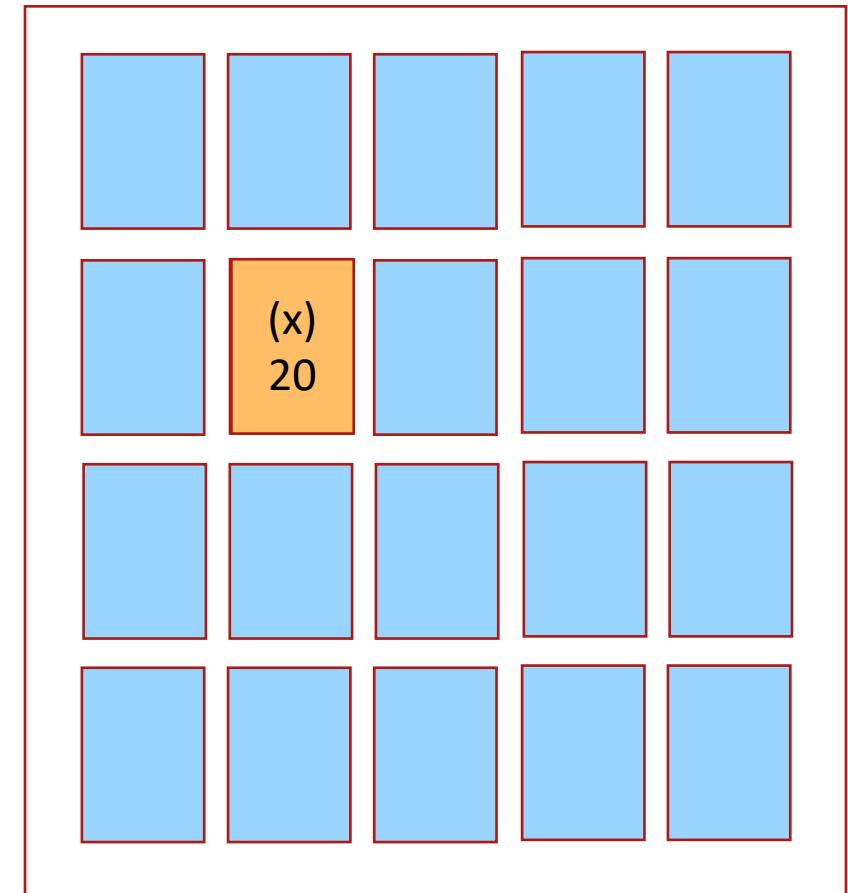
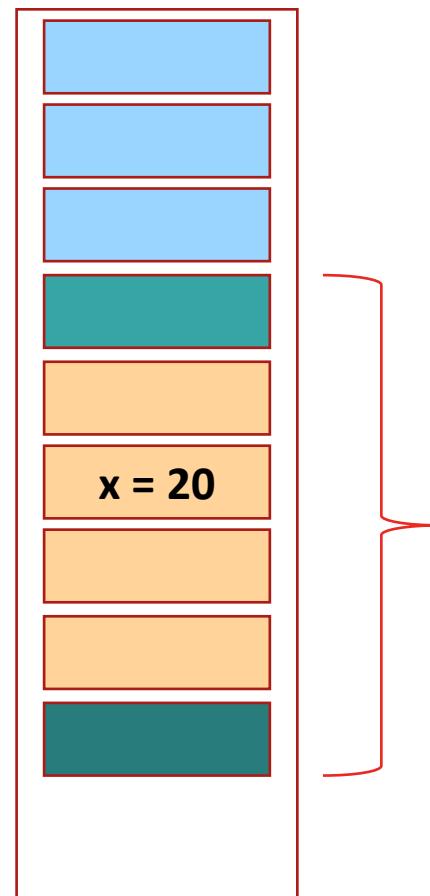


|



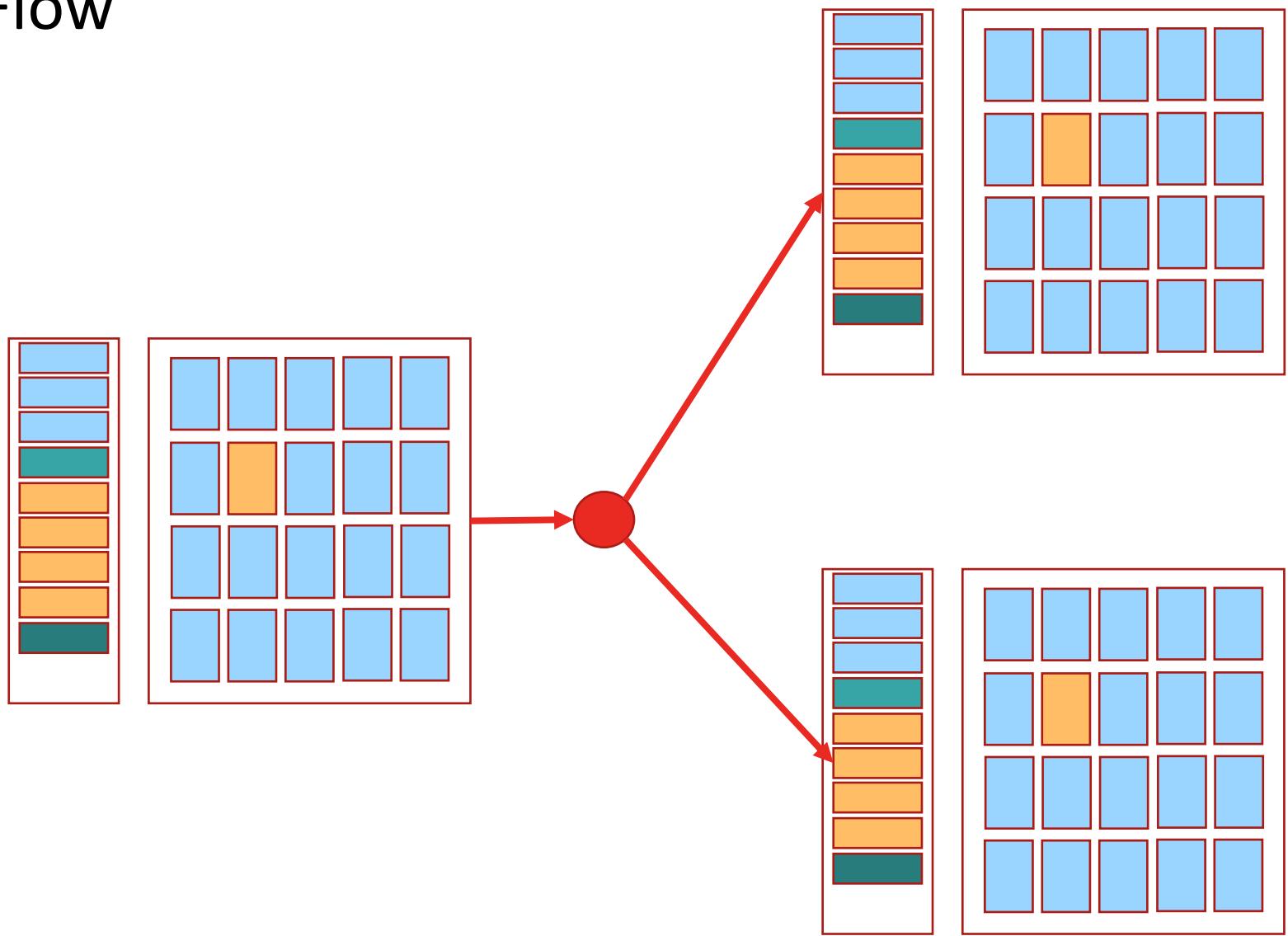
# SQL Database IO Flow

Start TXN →  
.....  
Update  $x=20$  →  
.....  
.....  
COMMIT →



# NoSQL Database IO Flow

Start (\*TXN) →  
.....  
Update x=10 →  
.....  
.....  
COMMIT (TIMEOUT) →



# Design Assumptions

# Design Assumptions

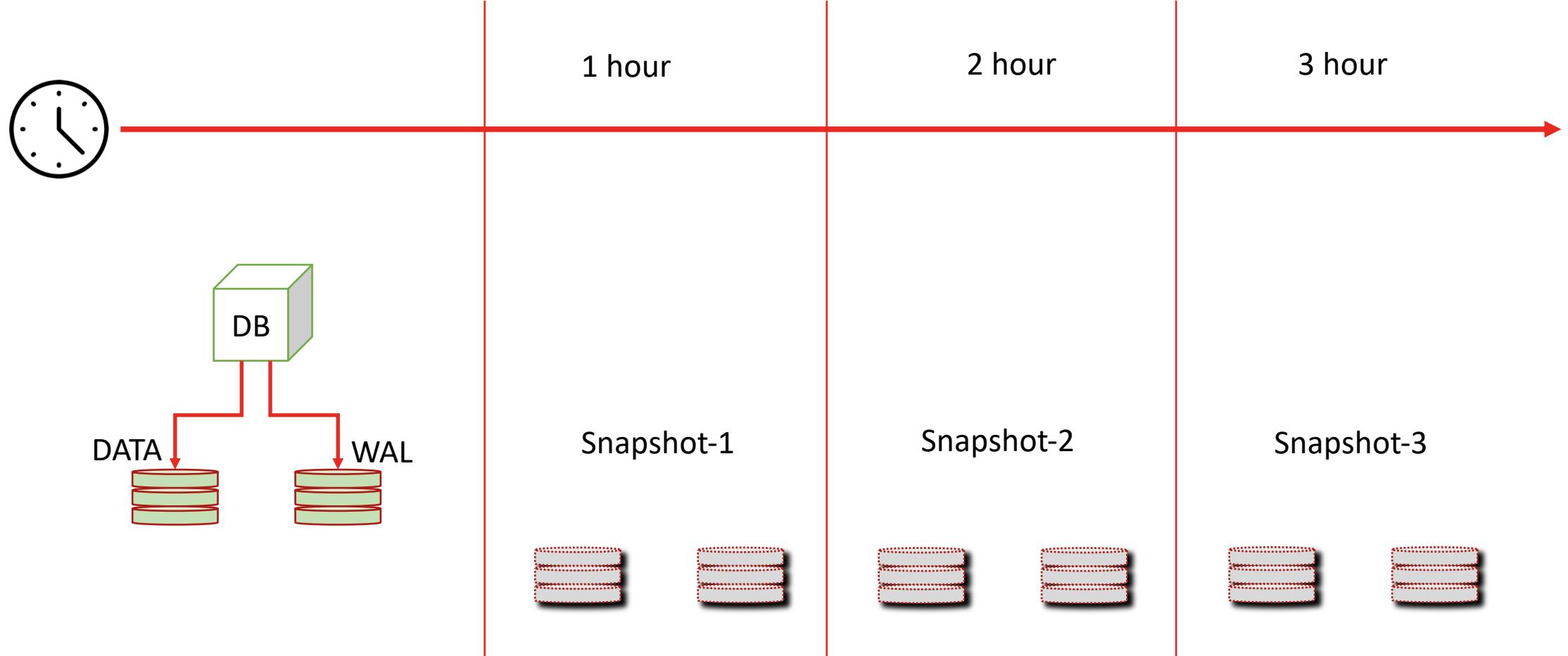
- › WAL and Data files reside on separate disks
- › If there are multiple WAL files, the WALS reside on different disks (Oracle ASM disks)
- › WAL is an append-only workload
- › **Crash Consistency is assumed**
- › **NoSQL Databases assumes storage for partitions go to different hardware**

# IO Path optimizations

- › WAL (Write ahead logging), UNDO log, REDO log
- › WAL are **append-only**, persistent circular buffers
- › Dirty Page management
- › Double write optimization
- › Parallel xlog writes

**Application design assumptions influence Data Protection Strategy**

# Application Data Protection



# The Loop

```
for each volume{  
    snapshot(volume)  
}
```

```
for each container{  
    for each volume{  
        snapshot(volume)  
    }  
}
```

# The Dangerous Loop

```
for each volume {  
    snapshot(volume)  
}
```

T0: Snapshot (WAL)

Process Scheduled out

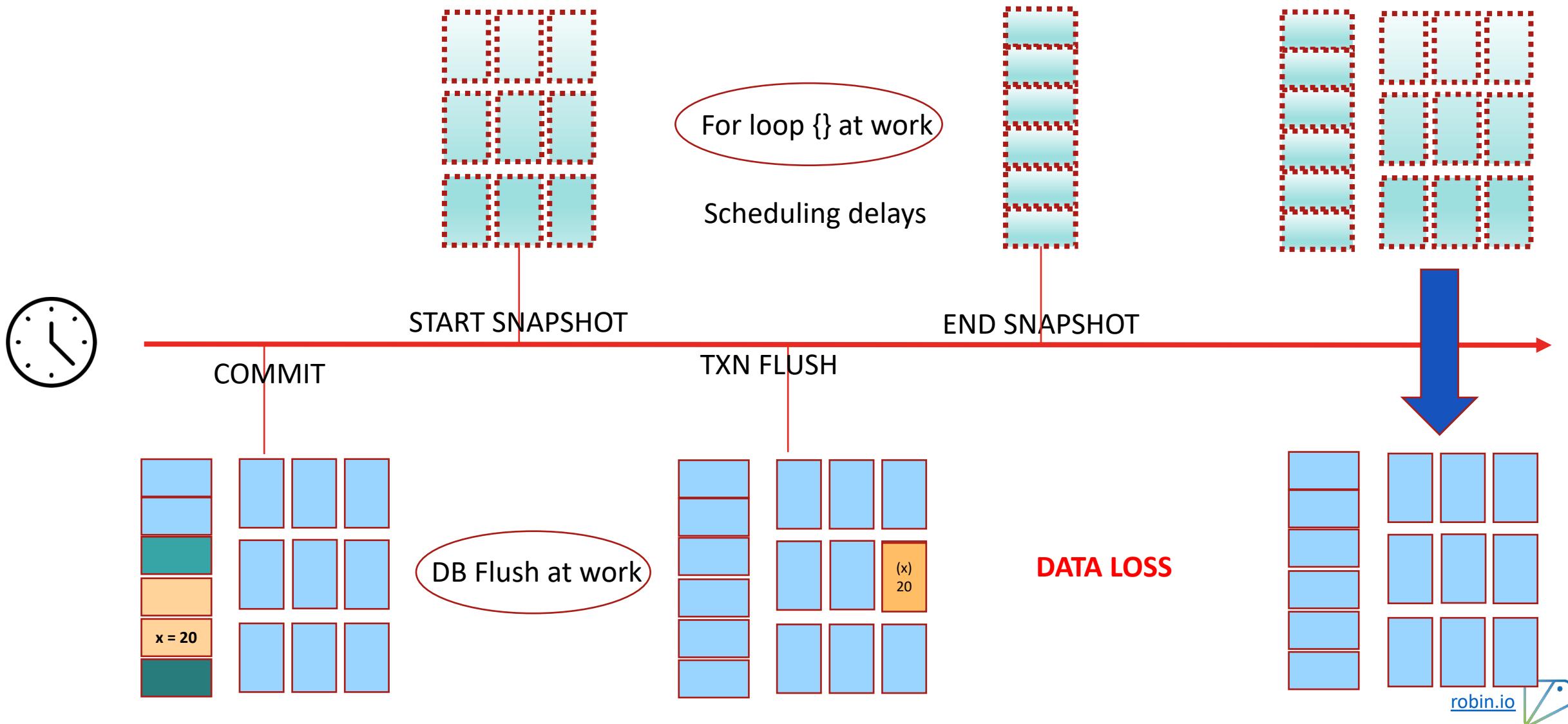
T1: Snapshot (DATA)

T0: Snapshot (DATA)

Process Scheduled out

T1: Snapshot (WAL)

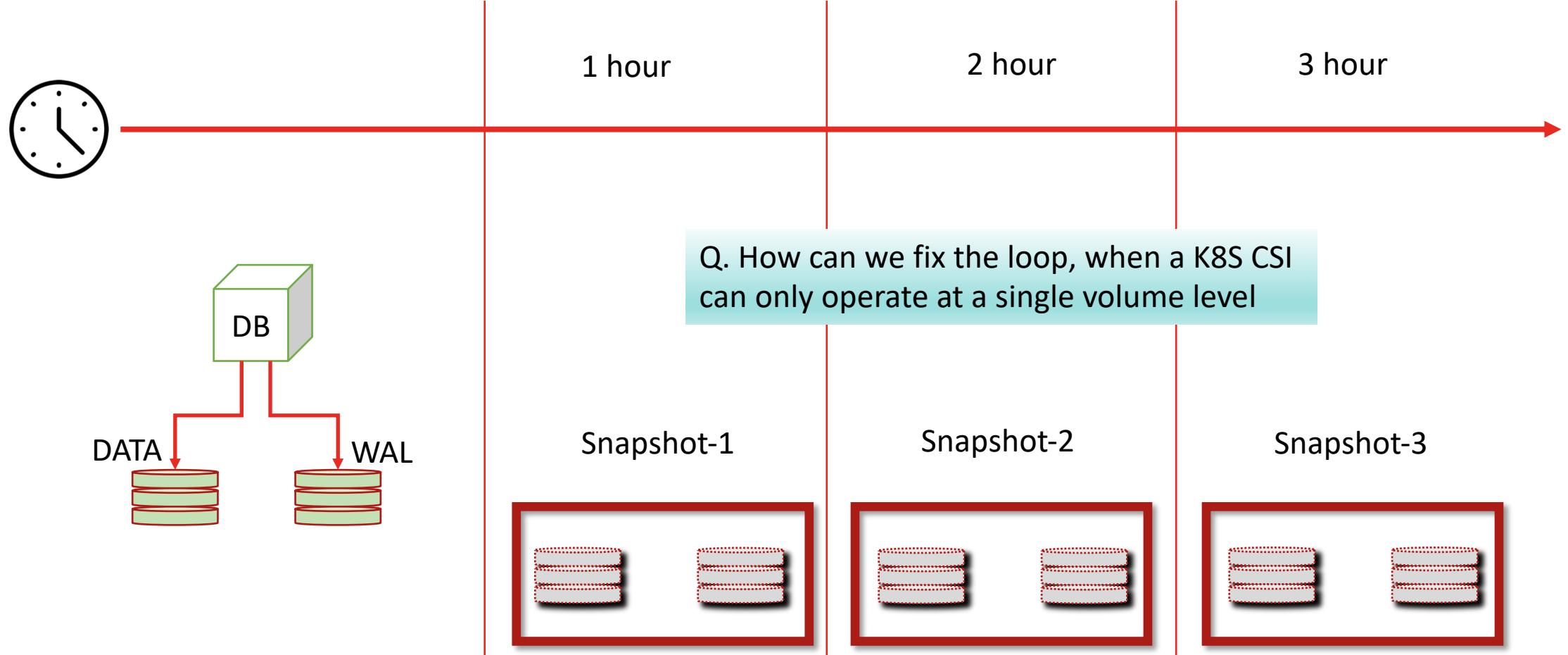
# Data Race



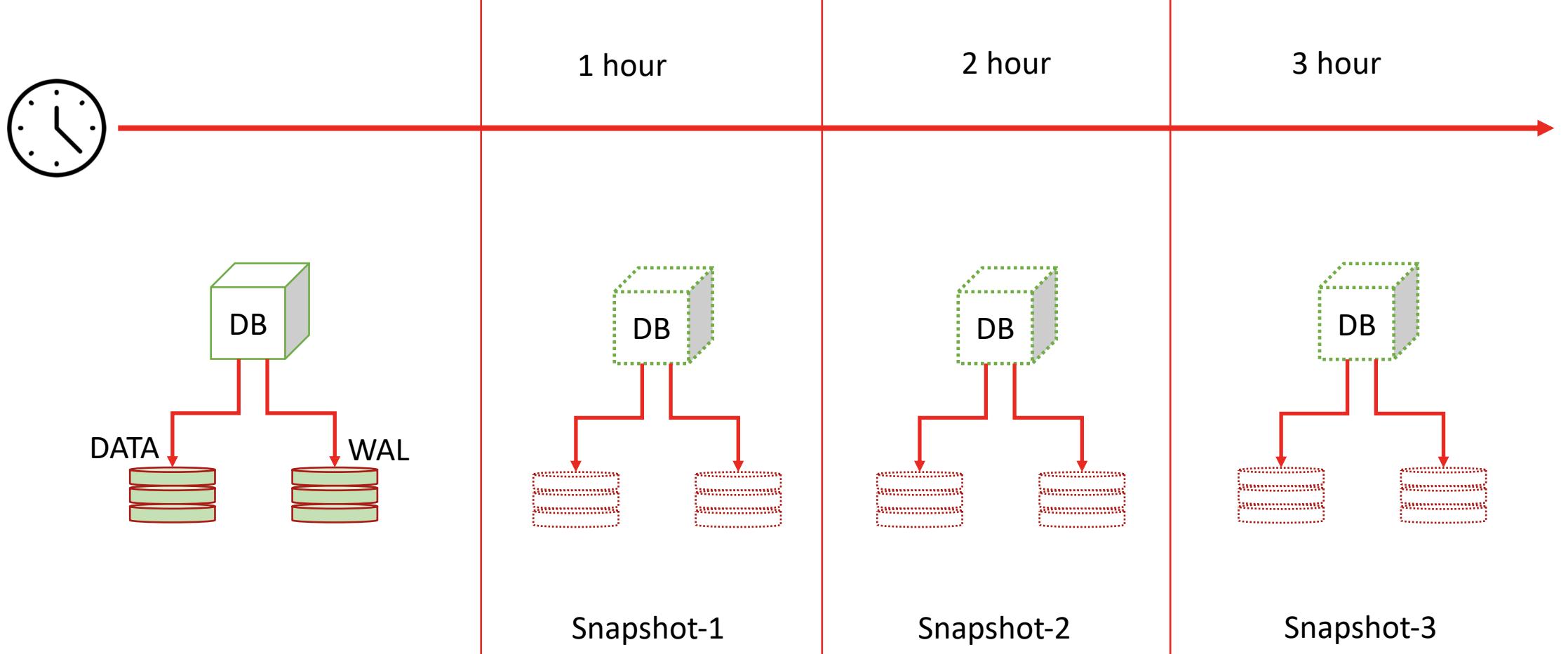
# Consistency Groups

- › Group of volumes that function as a Unit
- › Group on which all the life-cycle operations work
- › Group that maintains write-order
- › Group that will preserve crash-consistency semantics

# Application Data Protection



# Application Snapshots



# Protect an entire Application, not just Storage Volumes

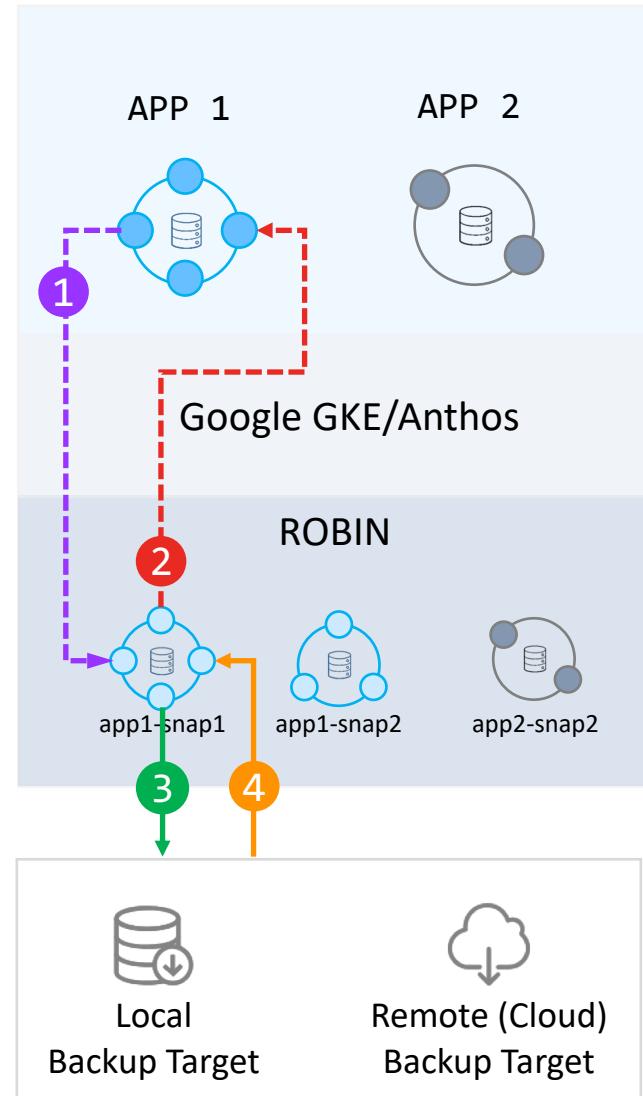
- 1 Maintain periodic checkpoints of your entire app with data

```
$ robin snapshot app1 snap1
```

1	DATA	PersistentVolumeClaims
2	CONFIG	ConfigMap, Secret, Labels, ...
3	METADATA	Pods, StatefulSets, Services, ...

- 2 Rollback entire app+data to healthy state to recover from corruptions or user errors

```
$ robin rollback snap1 app1
```



- 3 Backup entire app+data as into external backup targets

```
$ robin backup snap1 target
```

- 4 Restore entire app+data to healthy state from catastrophic hardware and datacenter failures

```
$ robin restore target snap1
```

› ROBIN Backups are fully self-contained

› Entire app resources can be restored in the same or different data center or cloud even if the source is completely destroyed

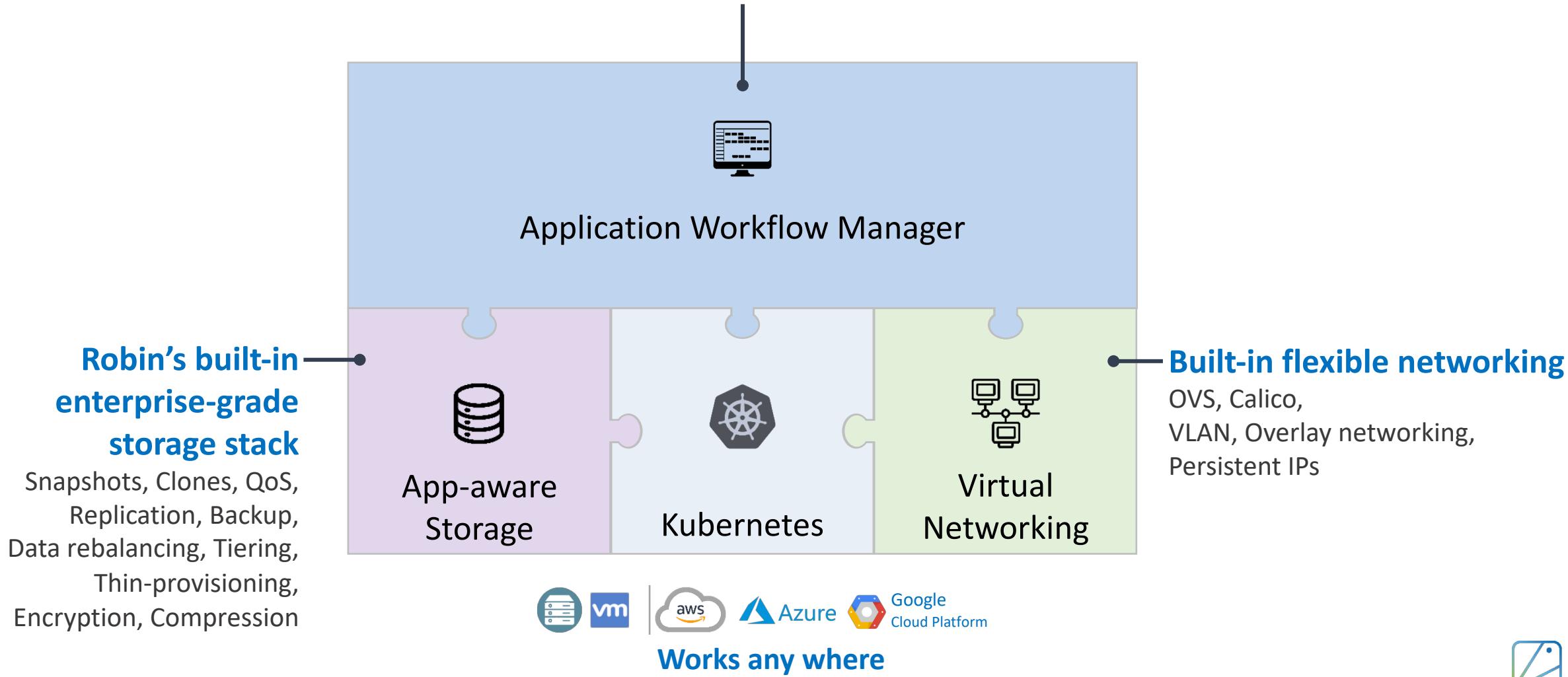
[robin.io](https://robin.io)



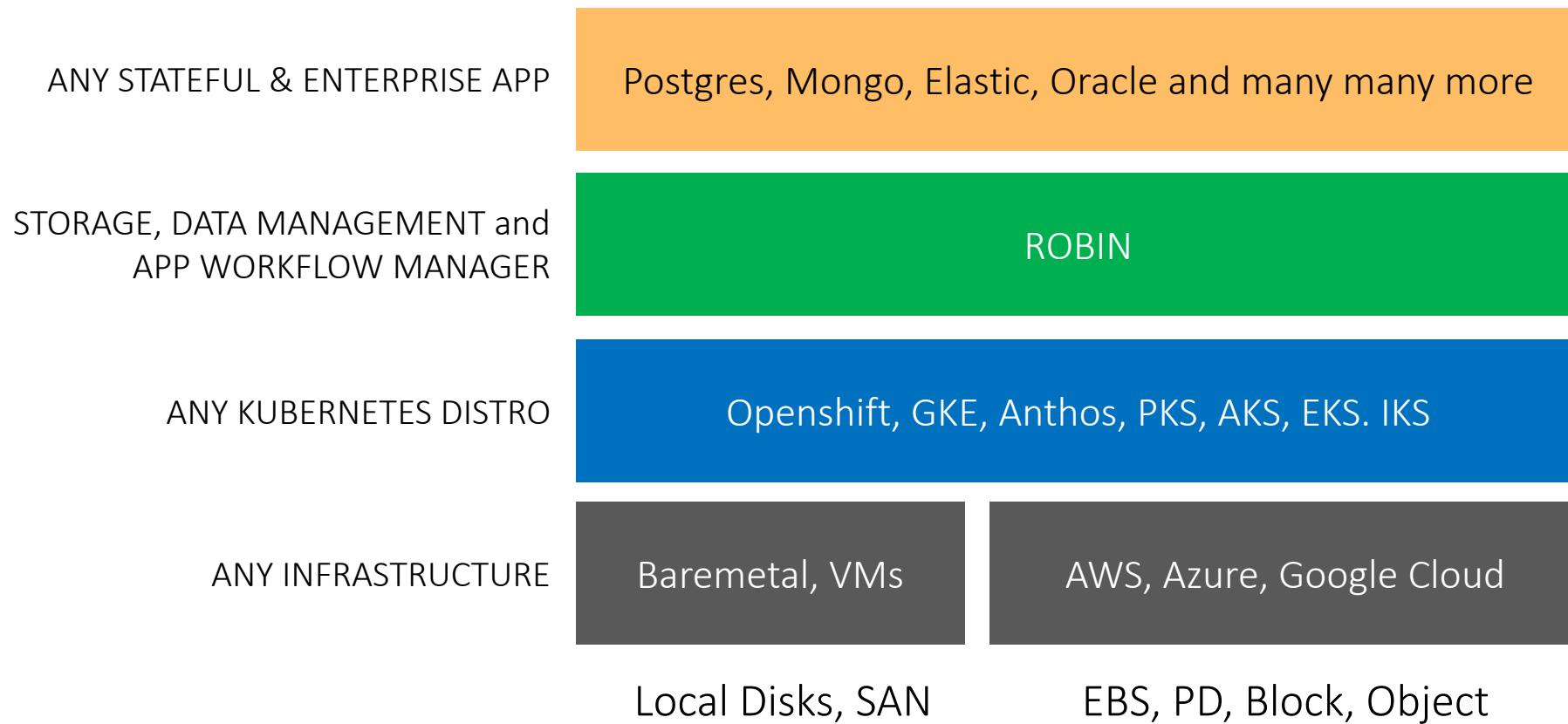
# Robin Architecture Overview

**1-click application Deploy, Snapshot, Clone, Scale, Upgrade, Backup**

Application workflows configure Kubernetes, Storage & Networking



# How Robin helps you run stateful apps on Kubernetes



**ROBIN software allows you run complex Big Data and Databases on Kubernetes**  
**(Storage + Networking + Application Workflow Management + Kubernetes)**

## DEPLOYMENT PROOF POINTS

**11 billion security events ingested and analyzed a day**  
**(Elasticsearch, Logstash, Kibana, Kafka)**

**6 petabytes under active management in a single ROBIN cluster**  
**(Cloudera, Impala, Kafka, Druid)**

**400 Oracle RAC databases managed by a single ROBIN cluster**  
**(Oracle, Oracle RAC)**

# Product Demos

## Demo Series 1

ROBIN's Kubernetes-native Storage and Data Management stack to deploy and manage day-2 operations of Stateful Apps

[Snapshot and Rollback](#)

[Clone](#)

[Backup](#)

[Cloud-portability](#)

## Demo Series 2

ROBIN's Super-Operator for Enterprise Apps

[1-click Deploy, Cloud-Sync and Cloud-Motion](#)



# [get.robin.io](https://get.robin.io) Portal

- › [get.robin.io](https://get.robin.io): Download Robin Storage for free
- › [docs.robin.io](https://docs.robin.io): Read product documentation
- › [slack.robin.io](https://slack.robin.io): Interact with Robin engineering

The screenshot shows the get.robin.io portal homepage. At the top, there's a navigation bar with the ROBIN logo, DOWNLOAD, ACTIVATE, MY LICENSES, DOCS, SLACK, and a user profile icon. Below the navigation, a main heading reads "Supercharge Kubernetes to Run Postgres" with a subtitle "1-click Deploy, Scale, Snapshot, Clone, Backup, Migrate". A blue button says "Download and experience for free". Below this, a large image of a laptop screen displays a grid of application icons, including cloudera, druid, hortonworks, mapr, spark, cassandra, datastax, elasticsearch, elk, kafka, mongo, centos, nginx, redhat, ubuntu, db2, mariadb, mysql, oracle, oracle-rac, saphana, sqserver, weblogic, and couchdb. At the bottom, a section titled "What is ROBIN?" explains that ROBIN extends the agility, efficiency, and portability of Kubernetes to all applications, even complex Big Data, Databases, AI/ML, and custom apps, on any infrastructure, on-premise, hybrid cloud or multi-cloud.

# ROBIN.IO

Supercharge Kubernetes to Deliver Big Data and Databases as-a-Service

1-click Deploy, Scale, Snapshot, Clone, Upgrade, Backup, Migrate

