RESILIENCE

REALIZED

KubeCon | CloudNativeCon

North America 2021

Vitess: Introduction and New Features

Deepthi Sigireddi - PlanetScale @ATechGirl

Alkin Tezuysal - PlanetScale @ask_dba

Andrew Mason - Slack Corp. @andrew_mason1

Malcolm Akinje - Slack Corp. @LESkidd113

Agenda

- Introduction - *Deepthi Sigireddi*
- VTOP (Vitess Operator) + Demo - *Alkin Tezuysal*
- VTAdmin Overview + Demo - *Andrew Mason*
- VDiff Coordinator Demo - *Malcolm Akinje*

Vitess

A database clustering system for horizontal scaling of MySQL / MariaDB

- CNCF graduated project
- Open source, Apache 2.0 licence
- Contributors from around the community
- Written in Golang

Vitess

@vitessio

Vitess

Cloud-native distributed database
- Runs in Kubernetes

Scalable

Highly available

Durability guarantees

Illusion of “single database”
- Single dedicated connection
- MySQL 5.7 or 8.0
- Compatible with frameworks / ORMs etc.

@vitessio

Vitess serves millions of QPS in production

slack
New Relic
Square
Flipkart
HubSpot
peak
Pinterest
YouTube
nozzle
weave
GitHub
JD.COM 京东
Quiz of Kings
stitchlabs
planetscale

Vitess

@vitessio

Concepts

Keyspace

- Logical database

Shard

- Subset of logical database

Cell

- Failure domain

Vitess Architecture basics

A common replicated database cluster with primary and replicas

Vitess

@vitessio

Vitess Architecture basics

Each MySQL server is assigned a **vttablet**

- A daemon/sidecar
- Controls the **mysqld** process
- Interacts with the **mysqld** server
- Typically on same host as **mysqld**

Vitess

@vitessio

Vitess Architecture basics

In production you have multiple clusters

Vitess

@vitessio

Vitess Architecture basics

User and application traffic is routed via **vtgate**

- A smart, stateless proxy
- Speaks the MySQL protocol
- Impersonates a monolithic MySQL server
- Relays queries to **vttablet**s

Vitess

@vitessio

Vitess Architecture basics

A vitess deployment will run multiple **vtgate** servers for scale out

Vitess

@vitessio

Vitess Architecture basics

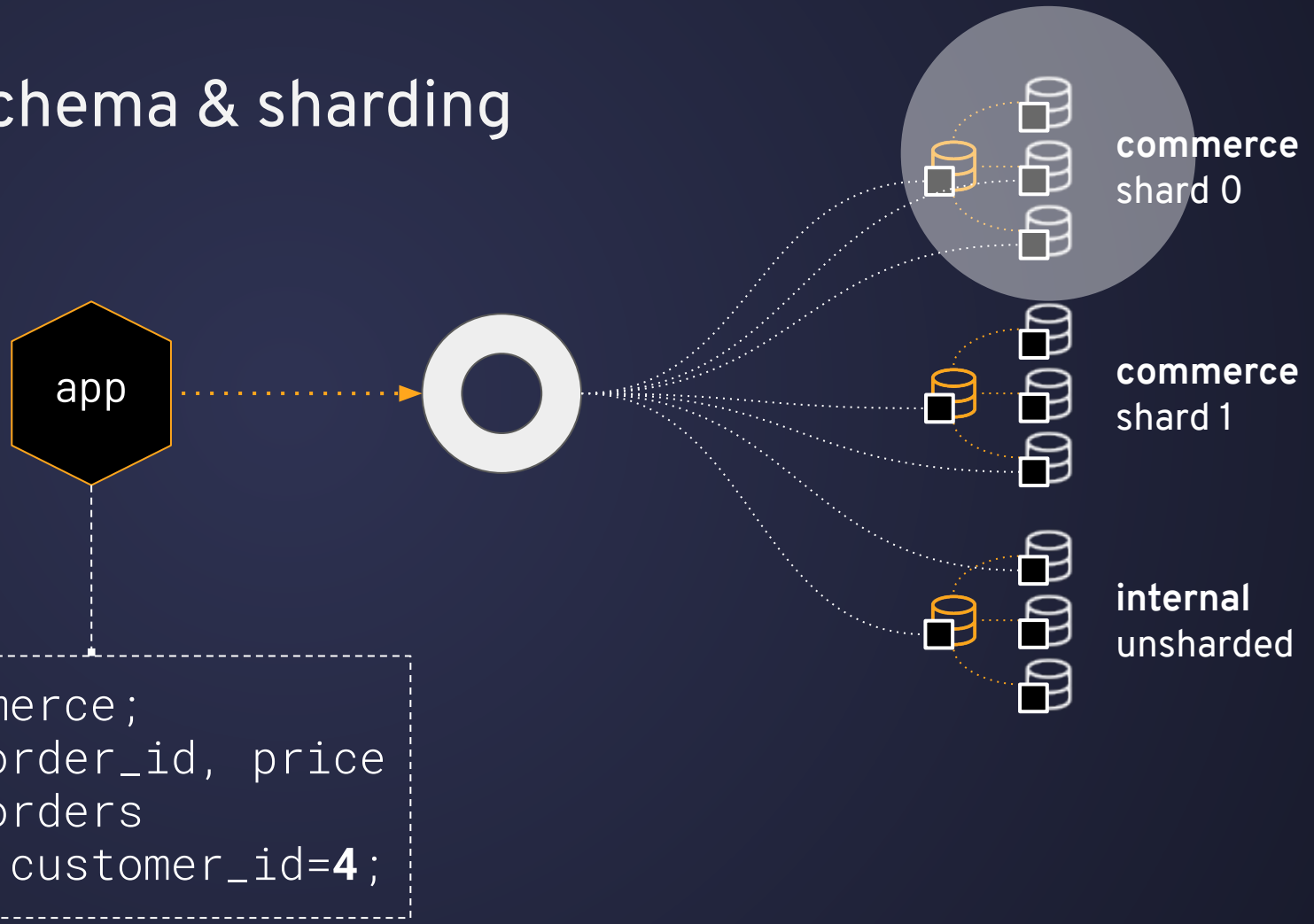
vtgate must transparently route queries to correct clusters, to relevant shards

Vitess

@vitessio

Vitess Architecture basics

Queries route based on schema & sharding scheme



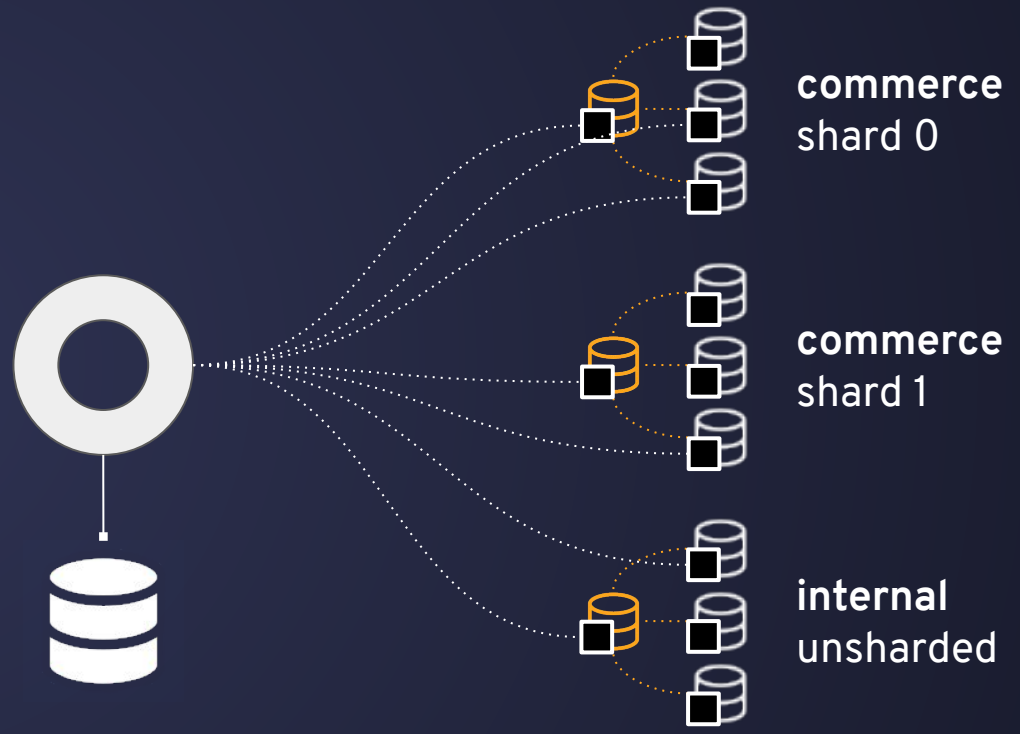
Vitess

@vitessio

Vitess Architecture basics

topo: distributed key/value store

- Stores the state of vitess: schemas, shards, sharding scheme, tablets, roles, etc.
- etcd/zookeeper/kubernetes
- Small dataset, mostly cached by **vtgate**



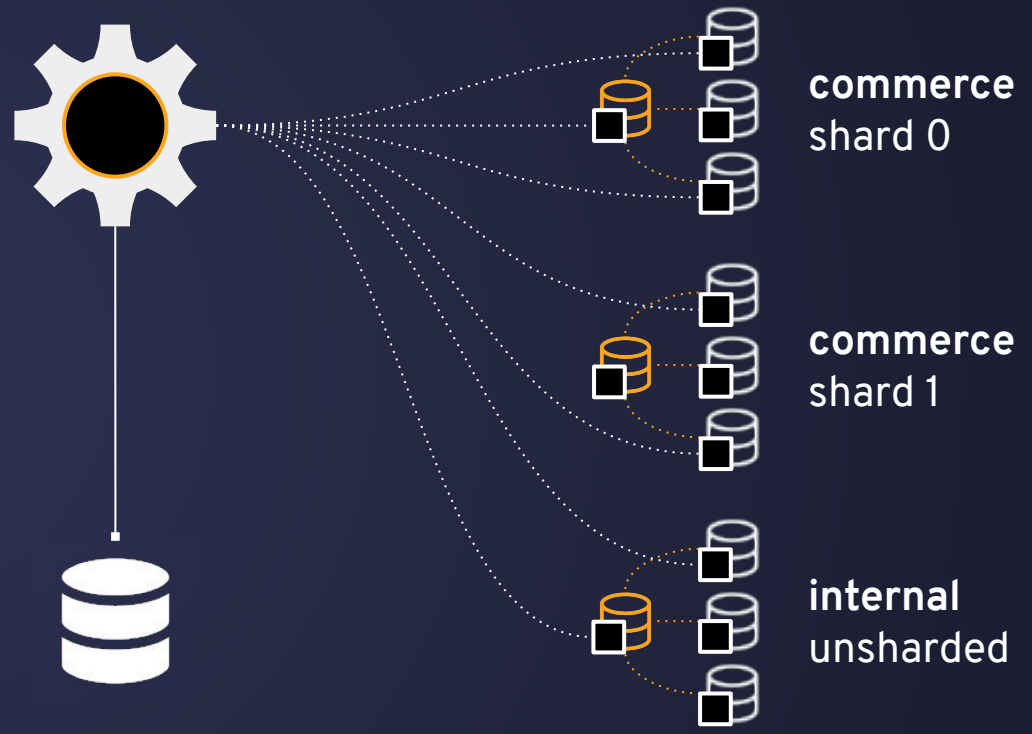
Vitess

@vitessio

Vitess Architecture basics

vtctld: control daemon

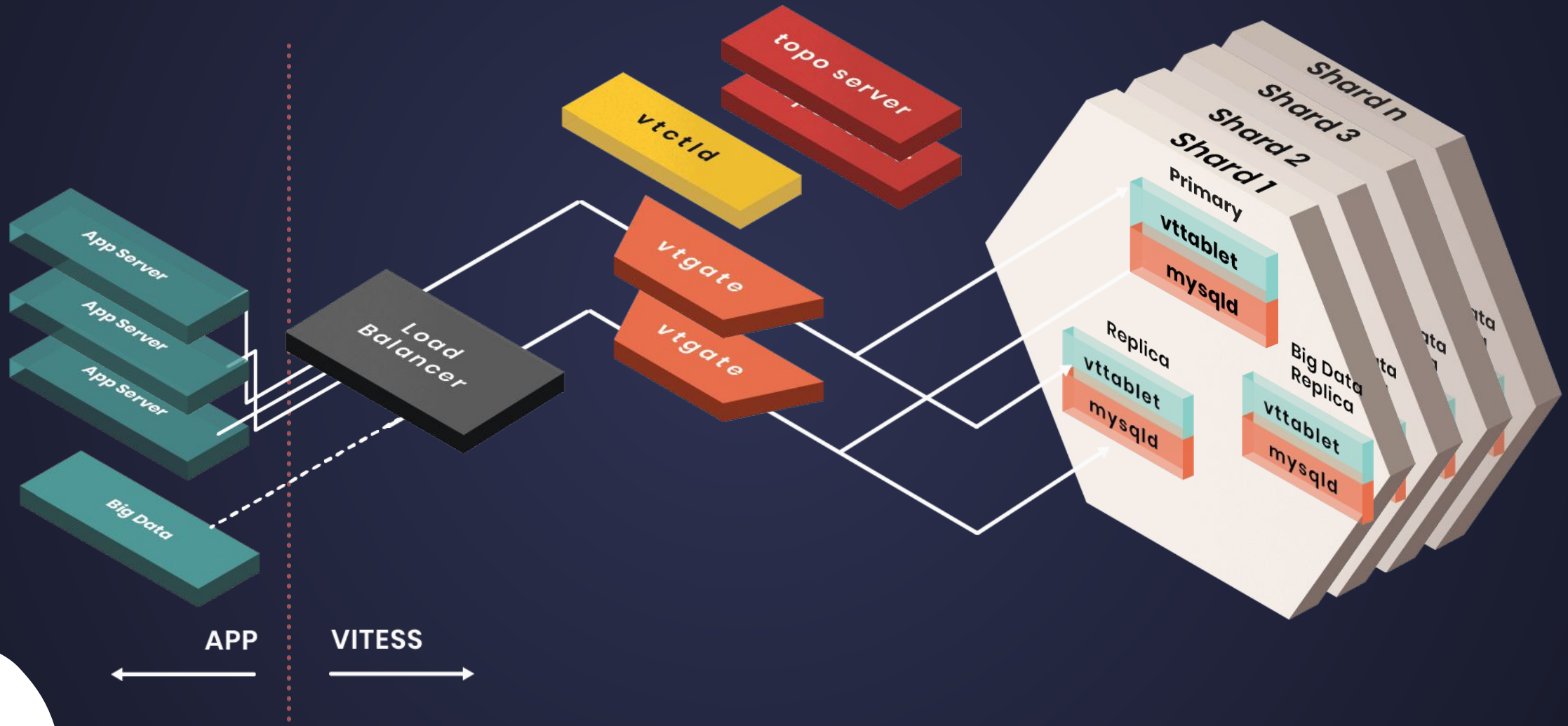
- Runs ad hoc operations
- API server
- Reads/writes **topo**
- Uses locks
- Operates on tablets



Vitess

@vitessio

Vitess Architecture Summary



Vitess

@vitessio

New and Upcoming Features

- 12.0 release GA Oct 26 2021
- Gen4 planner - experimental in 12.0
- Continuous Benchmarking - since 11.0
- Performance Improvements - ongoing
- More supported query constructs - ongoing
- VTAdmin
- Multi-column vindexes
- Automatic failure detection
- Collations
- Distributed transactions

VTOP - Vitess Operator for Kubernetes

Operator Functions and Automation of Vitess functions

Vitess

- Automation of Vitess functions
- Deploy Clusters
- Deploy Shards
- Failover nodes
- Recover pods
- Replicate data cross AZ

The Vitess Operator is open source on GitHub.

Vitess

@vitessio

VTOP - DEMO

Use of Operator

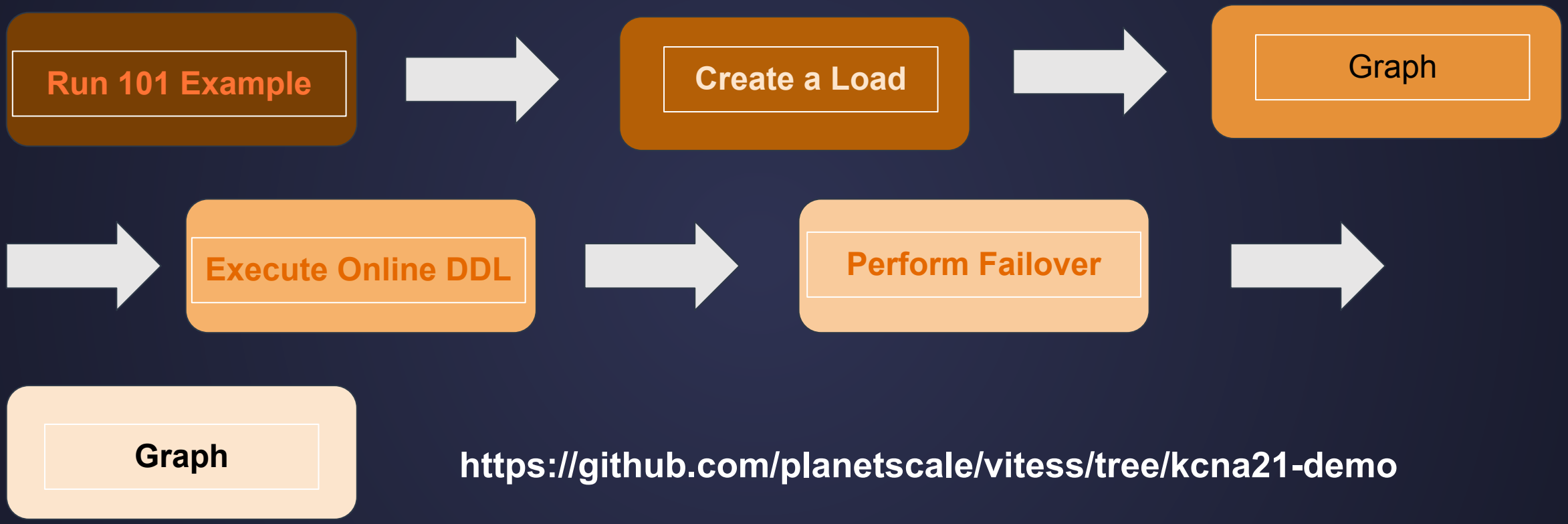
- Deploying a test cluster with Operator (Already deployed)
- Creating Sharding Workflow (Already created)
- Create a database load (Live)
- Failover during OnlineDDL on Sharded cluster (Live)

Vitess

Vitess

@vitessio

VTOP - DEMO Workflow



https://github.com/planetscale/vitess/tree/kcna21-demo

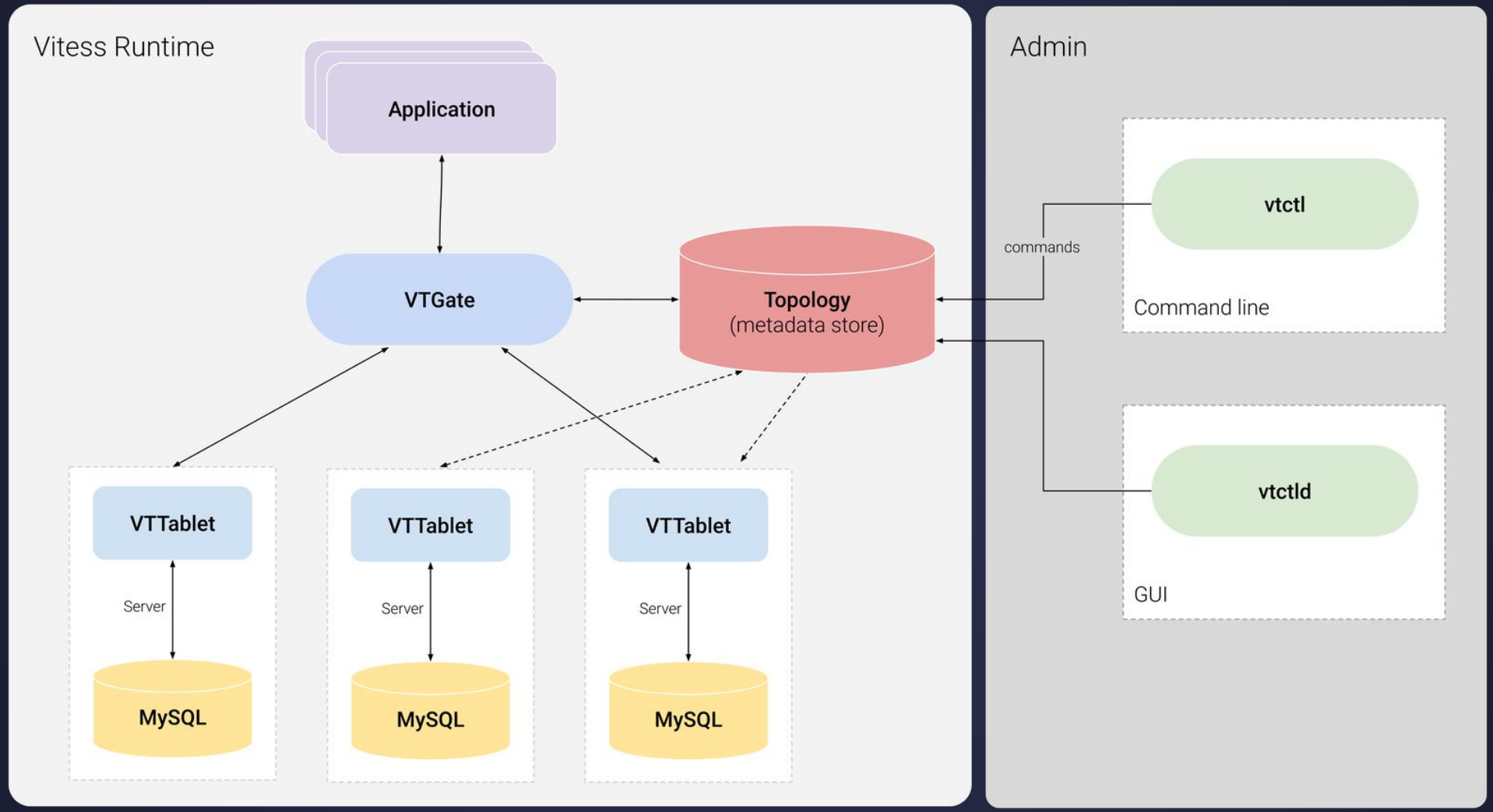
Vitess

@vitessio

VTAdmin - Overview

- Single control plane for many Vitess deployments (“clusters”)
- gRPC/http API and UI components
  - Golang backend with react frontend
- (Eventual) replacement for the existing vtctld UI

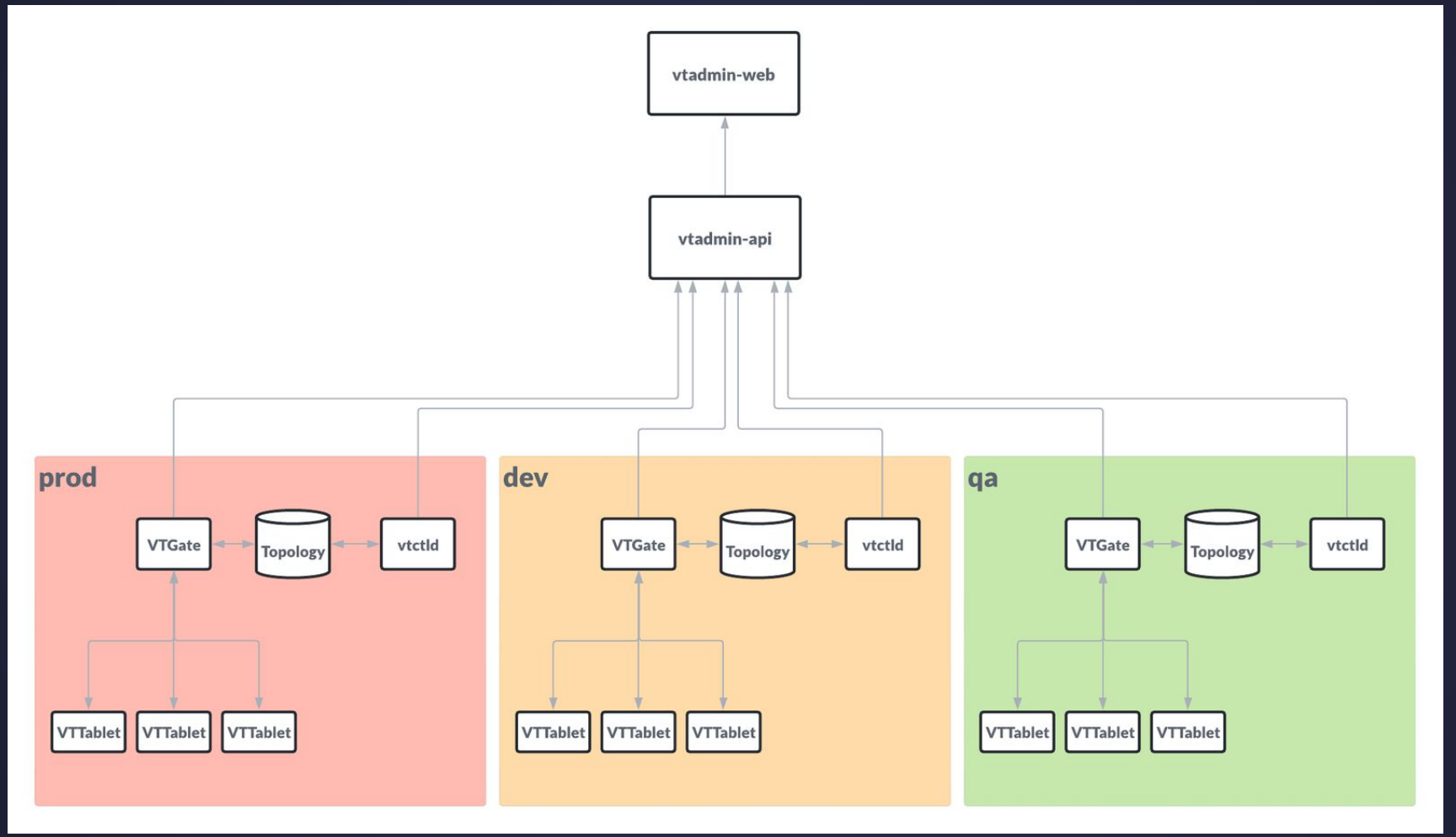
VTAdmin - Architecture (single cluster)



Vitess

@vitessio

VTAdmin - Architecture (multi cluster)



Vitess

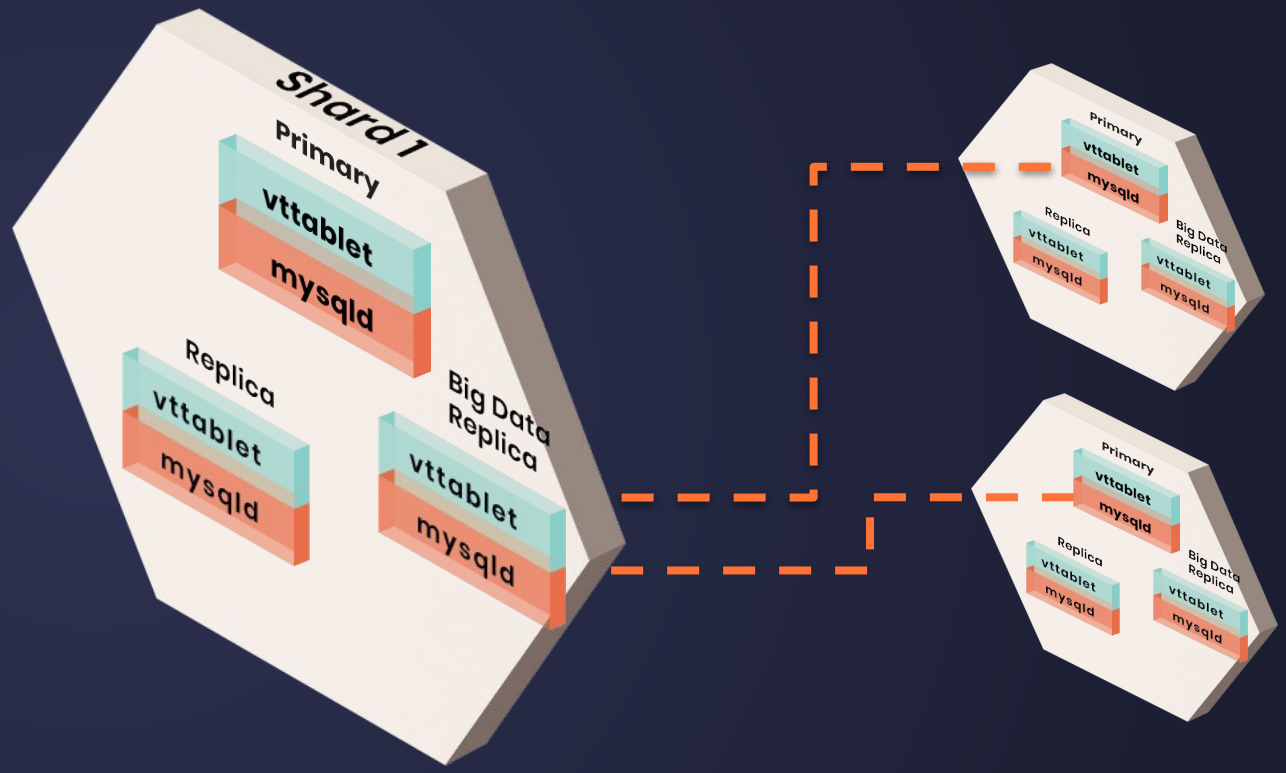
@vitessio

VTAdmin Demo

@vitessio

VDiff

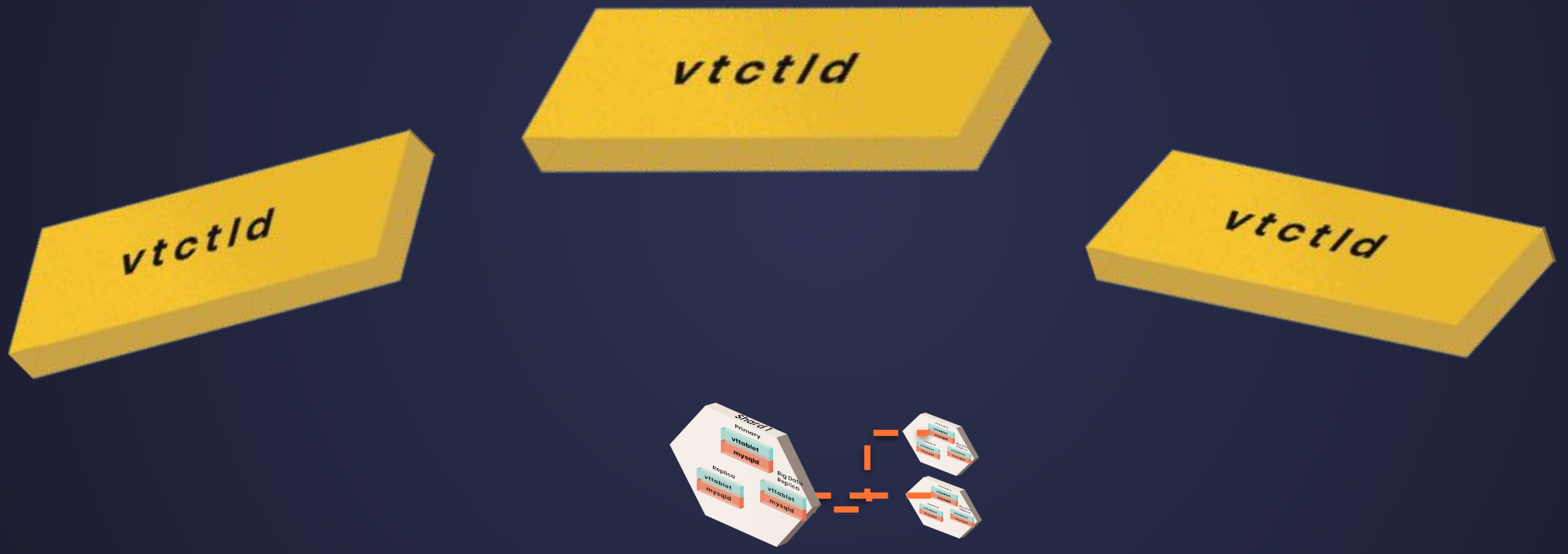
- Runs a diff between the source and target shards
- One VDiff per workflow
- This is a blocking call



Vitess

@vitessio

VDiff
(Work mainly happens in a vtctld)



Vitess

@vitessio

VDiff

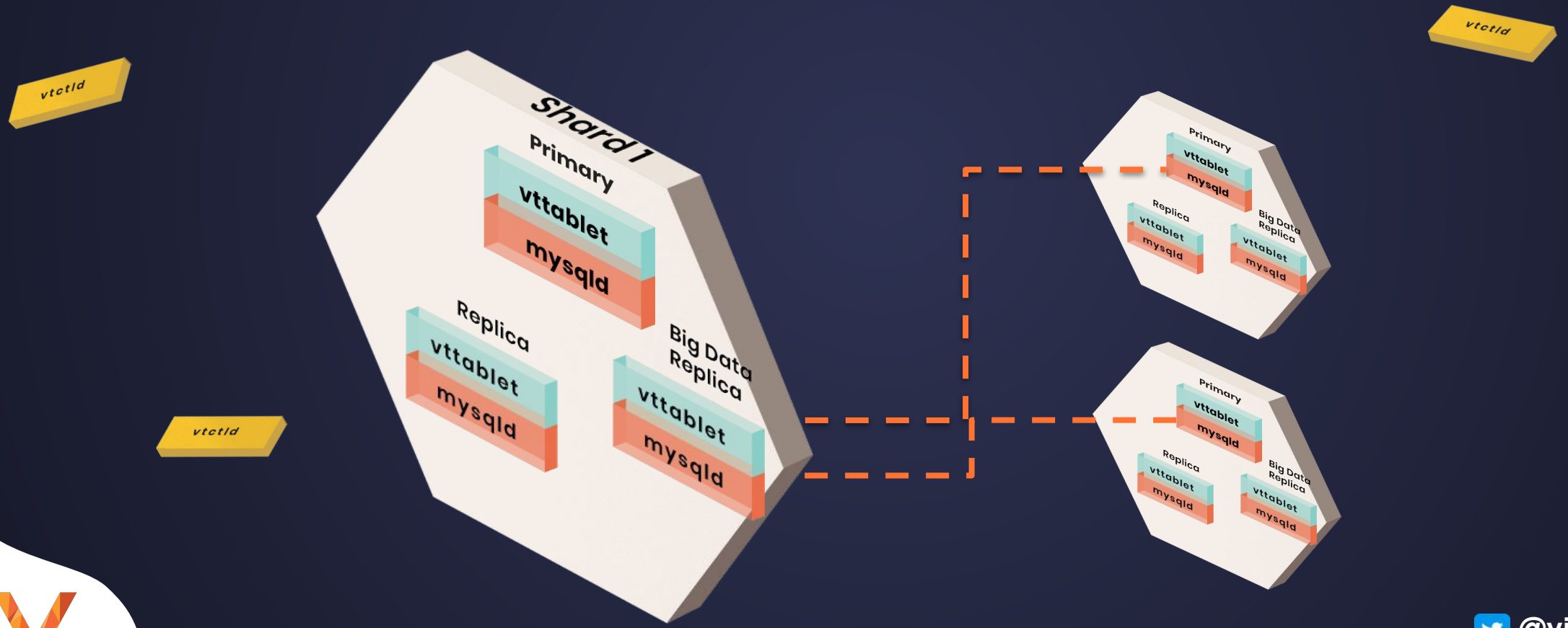
```
makinje@slack-vtctld-1:~$ sudo ./vtops vdiff \
--scell us_east_1 \
--tcell us_east_1 \
--ttype replica \
--ttype rdonly \
--wf keyspace.reshard_28-30 \
--wf keyspace.reshard_50-58 \
--wf keyspace.reshard_58-60
I 0624 14:02:59.636186 12366 vdiff:106 {}       🔎🔎Found Vtctlds🔎🔎
I 0624 14:02:59.636221 12366 vdiff:109 {}       slack-vtctld-1
I 0624 14:02:59.636232 12366 vdiff:109 {}       slack-vtctld-2
I 0624 14:02:59.636240 12366 vdiff:109 {}       slack-vtctld-3
I 0624 14:02:59.636250 12366 vdiff:133 {}       🚜🚜Starting 3 VDiff Workers🚜🚜
I 0624 14:02:59.636259 12366 vdiff:136 {}       VDiffWorker-0
I 0624 14:02:59.636268 12366 vdiff:142 {}       Using: slack-vtctld-1.example.com:15999
I 0624 14:02:59.636284 12366 vdiff:136 {}       VDiffWorker-1
I 0624 14:02:59.636292 12366 vdiff:142 {}       Using: slack-vtctld-2.example.com:15999
I 0624 14:02:59.636320 12366 vdiff:136 {}       VDiffWorker-2
I 0624 14:02:59.636329 12366 vdiff:142 {}       Using: slack-vtctld-3.example.com:15999
I 0624 14:02:59.636346 12366 vdiff:148 {}       📝📝Enqueuing 3 workflows📝📝
I 0624 14:02:59.636355 12366 vdiff:151 {}       keyspace.reshard_28-30
I 0624 14:02:59.636368 12366 vdiff:151 {}       keyspace.reshard_50-58
🚜🚜VDiffWorker-0 picked up keyspace.reshard_28-30🚜🚜
I 0624 14:02:59.636375 12366 vdiff:151 {}       keyspace.reshard_58-60
🚜🚜VDiffWorker-1 picked up keyspace.reshard_50-58🚜🚜
Retries Left: 3
Retries Left: 3
🚜🚜VDiffWorker-2 picked up keyspace.reshard_58-60🚜🚜
Retries Left: 3
🟢🟢 Vdiff for keyspace.reshard_50-58 completed 🟢🟢
🟢🟢 Vdiff for keyspace.reshard_28-30 completed 🟢🟢
🟢🟢 Vdiff for keyspace.reshard_58-60 completed 🟢🟢
```

Vitess

@vitessio

VDiff Coordinator Demo

VDiff Future Goals
(diffs done on tablet layer)



Vitess

@vitessio

Resources

Docs: vitess.io/docs/
Code: github.com/vitessio/vitess
Slack: vitess.slack.com

Thank you!

@vitessio