Introducing YugabyteDB Voyager

Accelerate cloud adoption with simple migration.

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

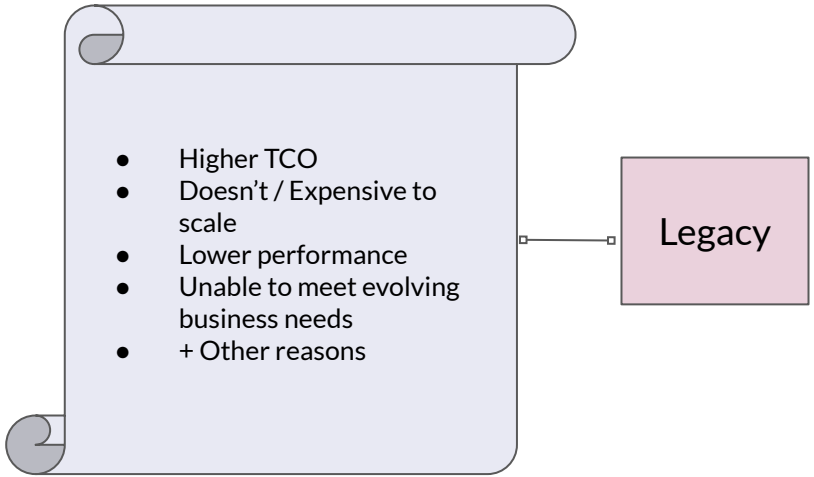
yugabyteDB

YB Voyager

Motivation

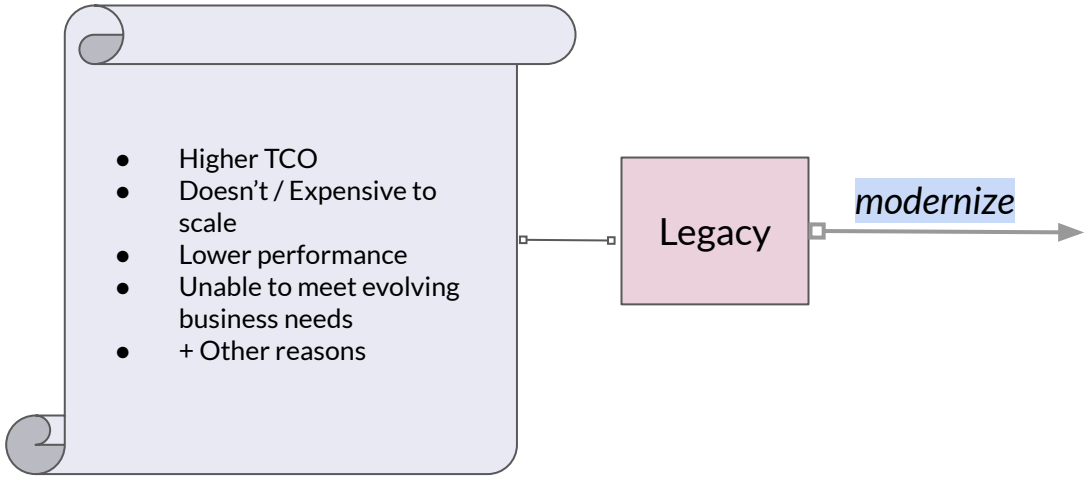
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize



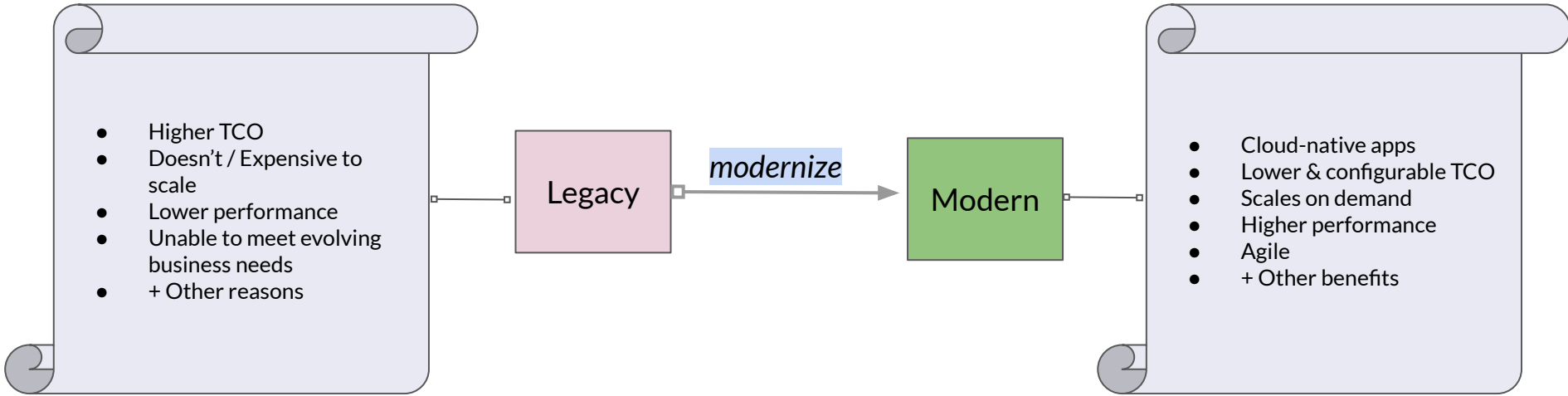
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize



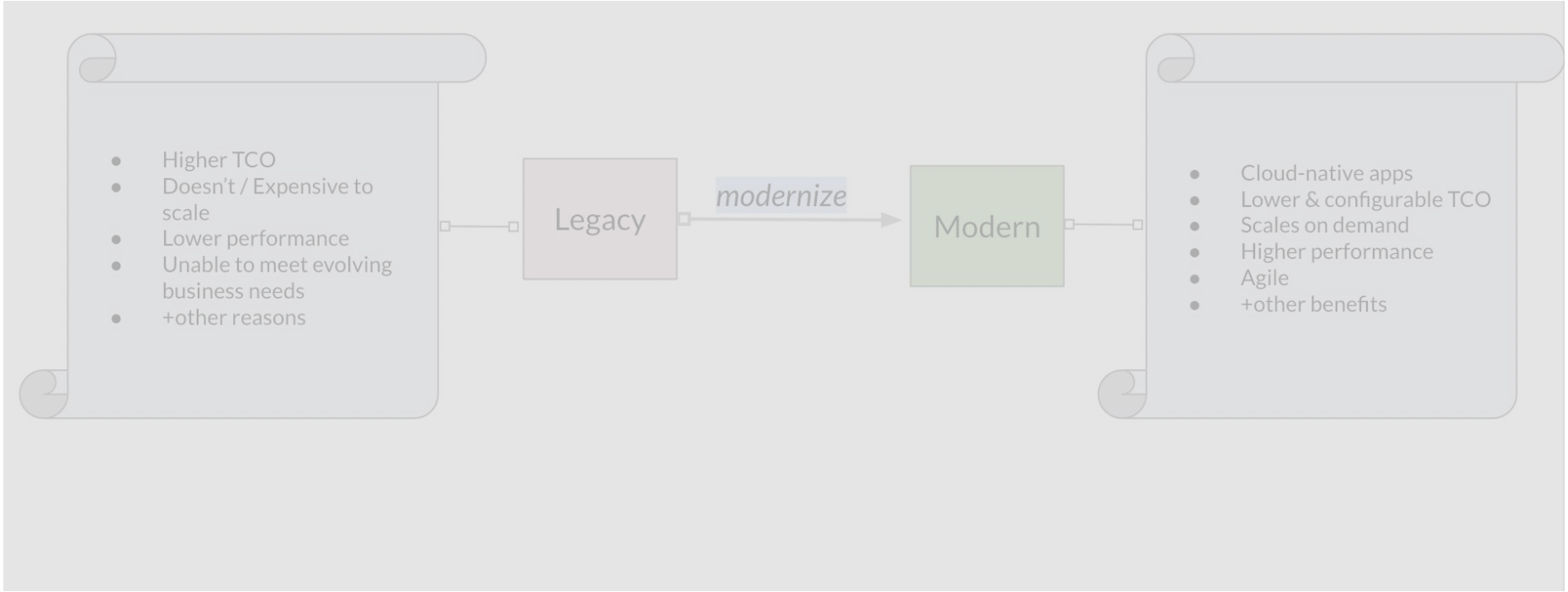
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize



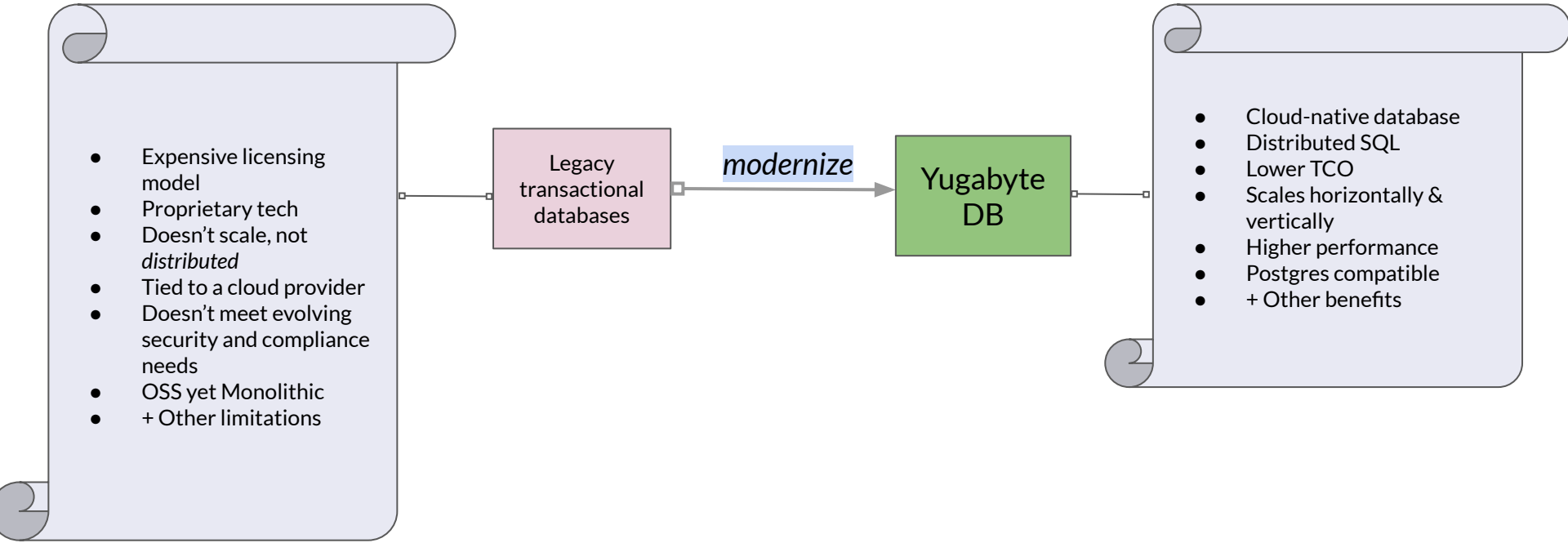
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize your *data*



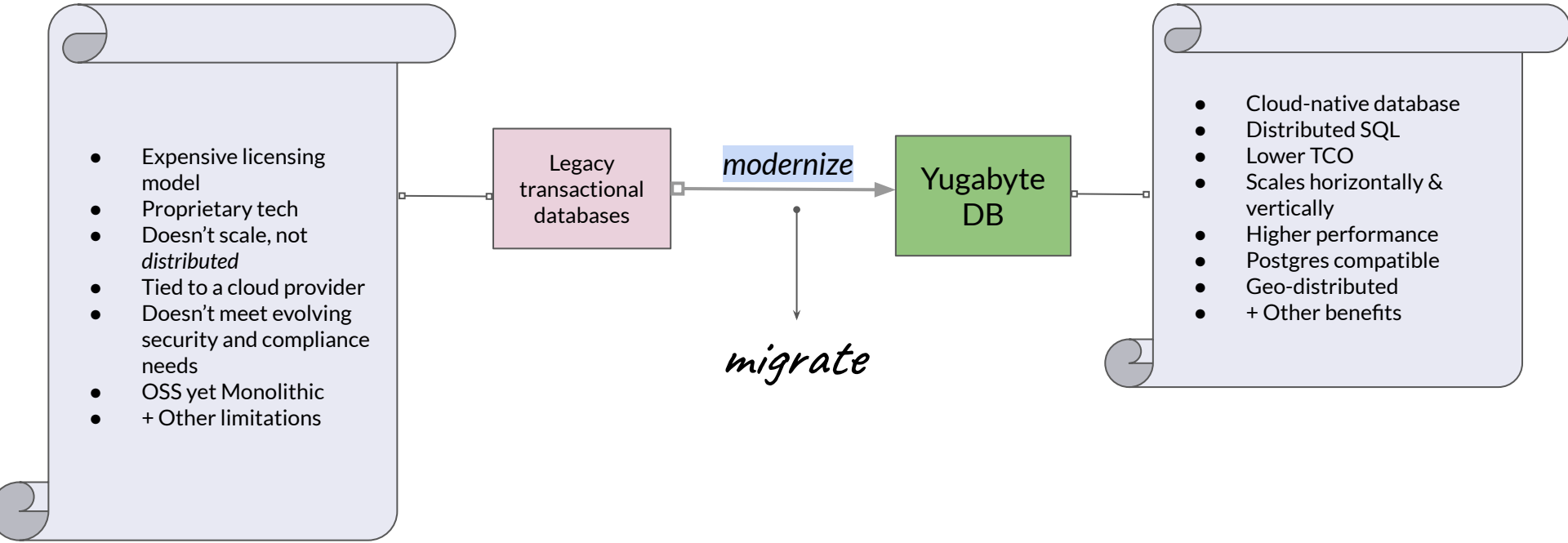
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize your *data*



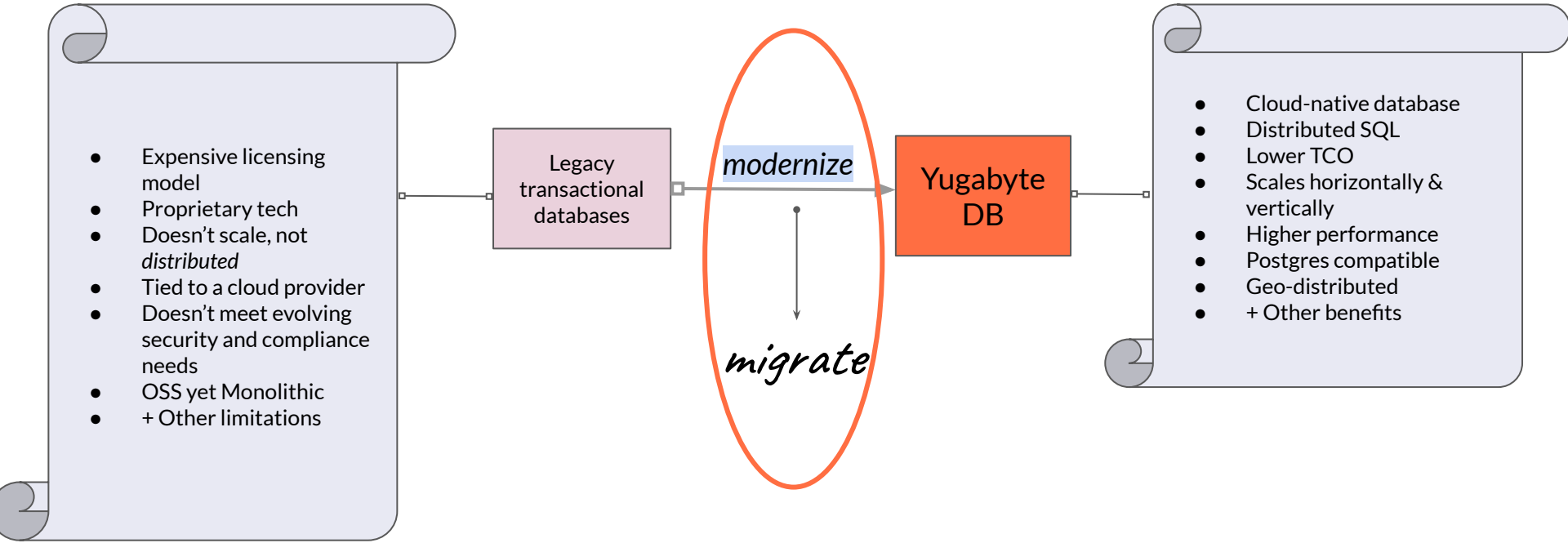
Yugabyte © 2022 – All Rights Reserved

Motivation: Modernize your *data*



Yugabyte © 2022 – All Rights Reserved

Motivation: *Migrate to a CN database to modernize your data*



Yugabyte © 2022 – All Rights Reserved

Data Migration

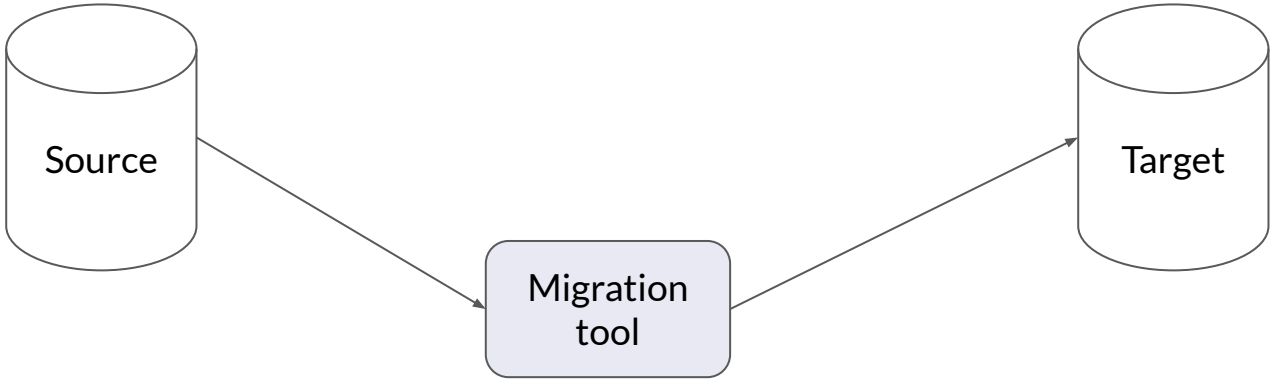
Challenges

Yugabyte © 2022 – All Rights Reserved

Typical *Migration* process

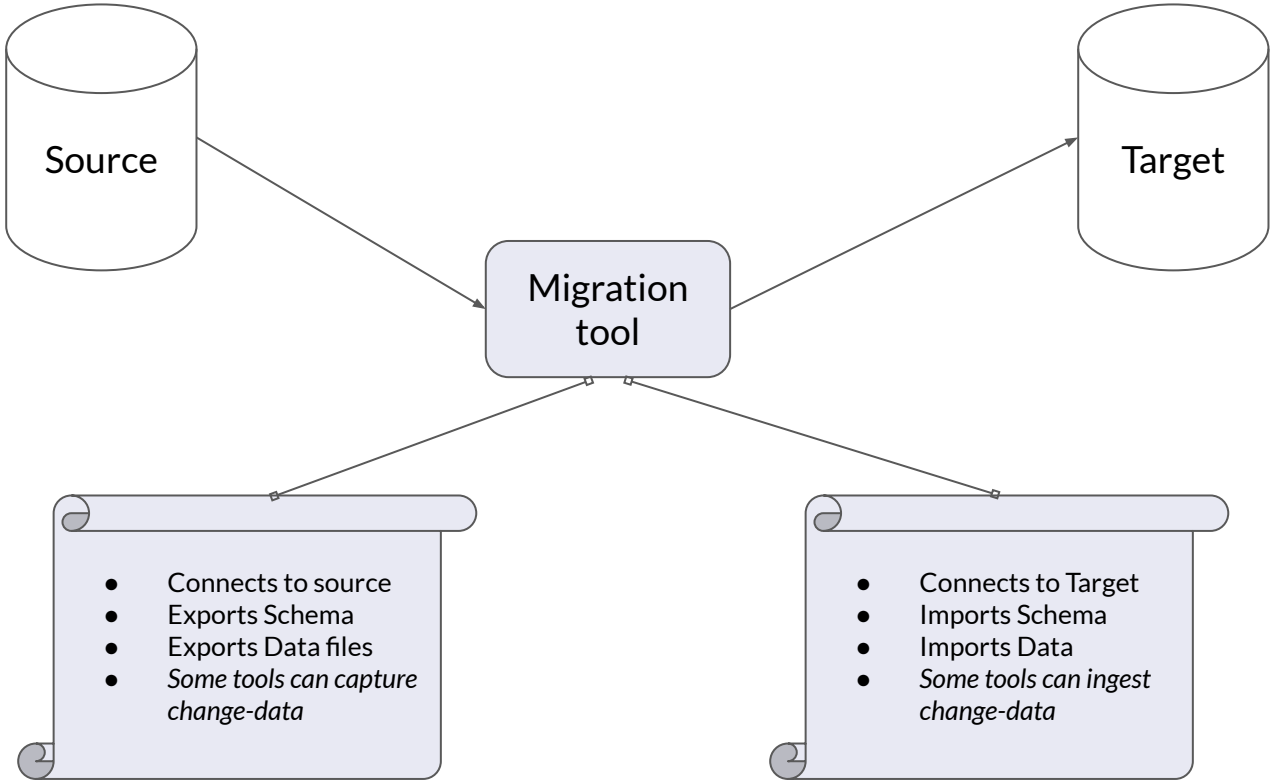
Yugabyte © 2022 – All Rights Reserved

Typical *Migration* process



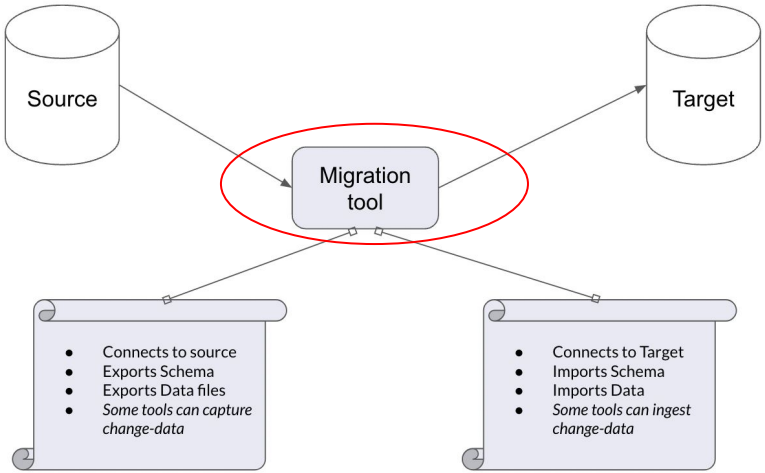
Yugabyte © 2022 – All Rights Reserved

Typical *Migration* process



Yugabyte © 2022 – All Rights Reserved

Typical *Migration* tools - *Atypical* for Distributed SQL



Works just fine when the Target is also a monolithic/single-node database.

Unaware of the strengths of a distributed database and doesn't scale.

Some tools depend on computational clusters to prep data sets to be ingested into a distributed database.

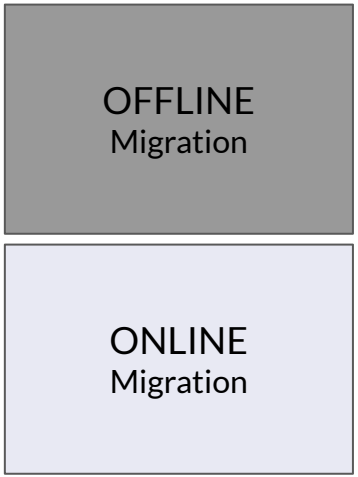
Yugabyte © 2022 – All Rights Reserved

YB Voyager

Requirements: When Migrating to Distributed SQL

Yugabyte © 2022 – All Rights Reserved

YB Voyager: Goals & requirements



Yugabyte © 2022 – All Rights Reserved

YB Voyager: Goals & requirements

OFFLINE Migration	❑ Export source database schema/objects ❑ Generate a report for users to verify ❑ Create schema objects on YB ❑ Bulk load data ❑ Do validation and generate a final report
ONLINE Migration	❑ All from above (OFFLINE mode) ❑ Stream change-data to YB
FAILBACK	❑ Replicate changes from YB to original source DB ❑ Mitigates risk of migrating to cloud

Yugabyte © 2022 – All Rights Reserved

YB Voyager: Goals & requirements

OFFLINE Migration	❑ Export source database schema/objects ❑ Generate a report for users to verify ❑ Create schema objects on YB ❑ Bulk load data ❑ Do validation and generate a final report
ONLINE Migration	❑ All from above (OFFLINE mode) ❑ Stream change-data to YB
FAILBACK	❑ Replicate changes from YB to original source DB ❑ Mitigates risk of migrating to cloud

- ➔ User friendly CLI
- ➔ Chunk the source files
- ➔ Parallel data ingestion based on Target YB cluster config
- ➔ Migrate data to YB from flat files & popular databases like Oracle, MySQL etc
- ➔ Same UX/steps for all supported source systems
- ➔ Safe defaults (with override):
  - ◆ #of connections based on target DB config
  - ◆ Ingest batch size
  - ◆ Disable transaction mode
  - ◆ Upsert mode
- ➔ Idempotent & should restart/resume from failed step

Yugabyte © 2022 – All Rights Reserved

YB Voyager: OFFLINE Migration process flow

1

Prepare
Install Voyager
Prepare source DB
Prepare target DB

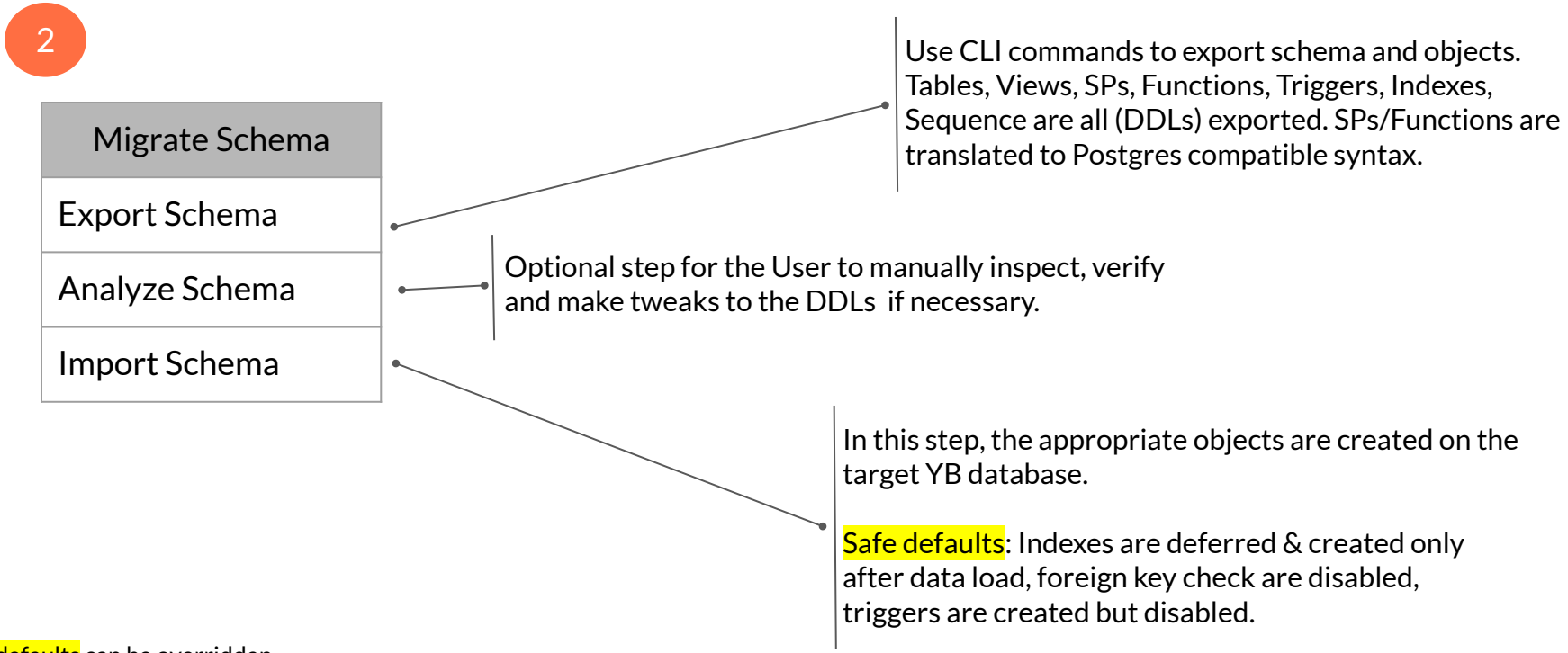
Voyager is installed on a VM with sufficient disk space

Verify connectivity and right level of access to source DB

Verify connectivity and right level of access to target Yugabyte DB

Yugabyte © 2022 – All Rights Reserved

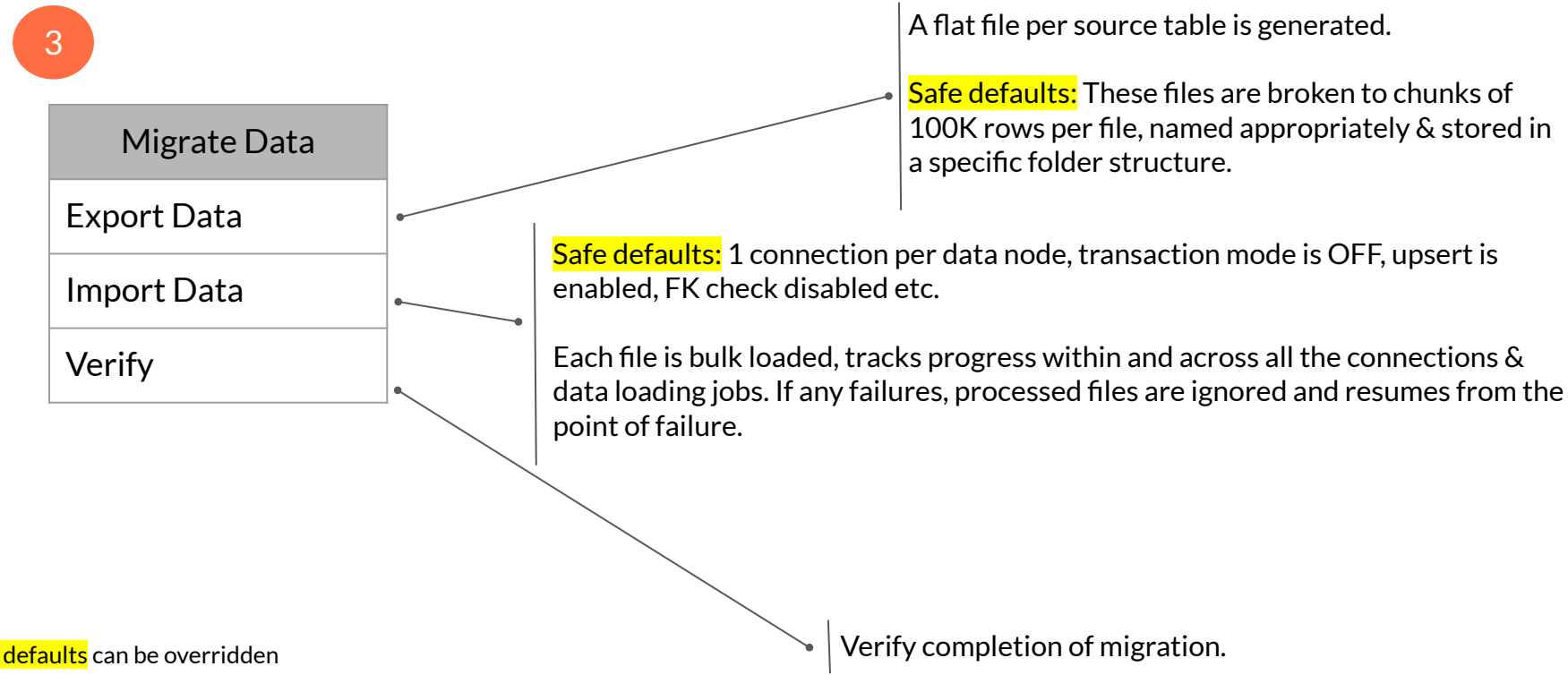
YB Voyager: OFFLINE Migration process flow



*Safe defaults can be overridden

Yugabyte © 2022 – All Rights Reserved

YB Voyager: OFFLINE Migration process flow



*Safe defaults can be overridden

Yugabyte © 2022 – All Rights Reserved

Thank You

YB Voyager Docs/Getting Started:
https://docs.yugabyte.com/preview/migrate/yb-voyager/

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

yugabyteDB