

# LINEにおけるMySQL運用の現状と バージョンアップを支える仕組み

ITSC Database dept db1 team / 大塚知亮

2020.12.17



LINE

# 概要

- 01 LINE Database室紹介
- 02 MySQL platformの変遷
- 03 バージョンアップを支える仕組み
- 04 運用を支えるツール
- 05 今後の課題

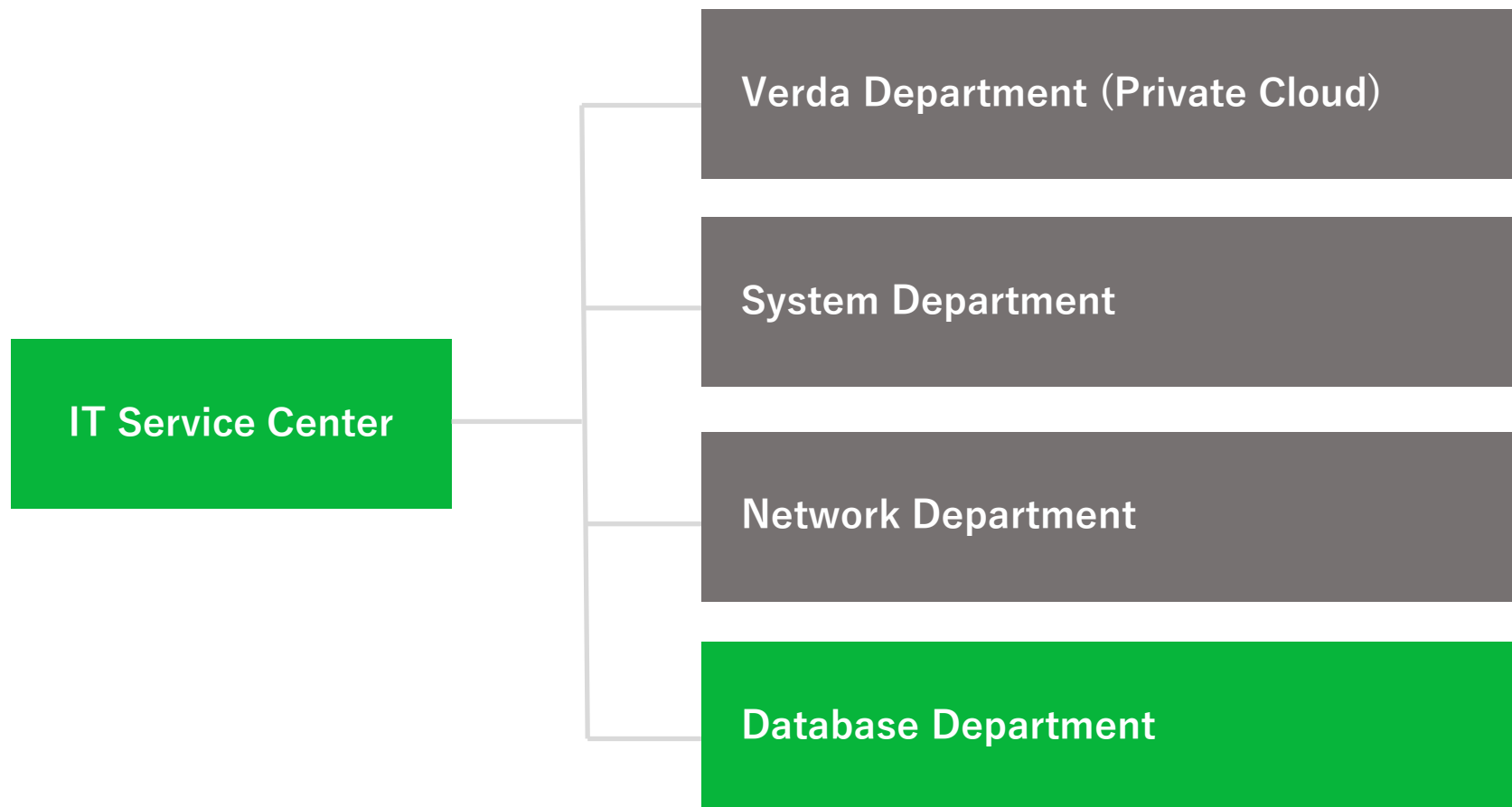
# 登壇者紹介



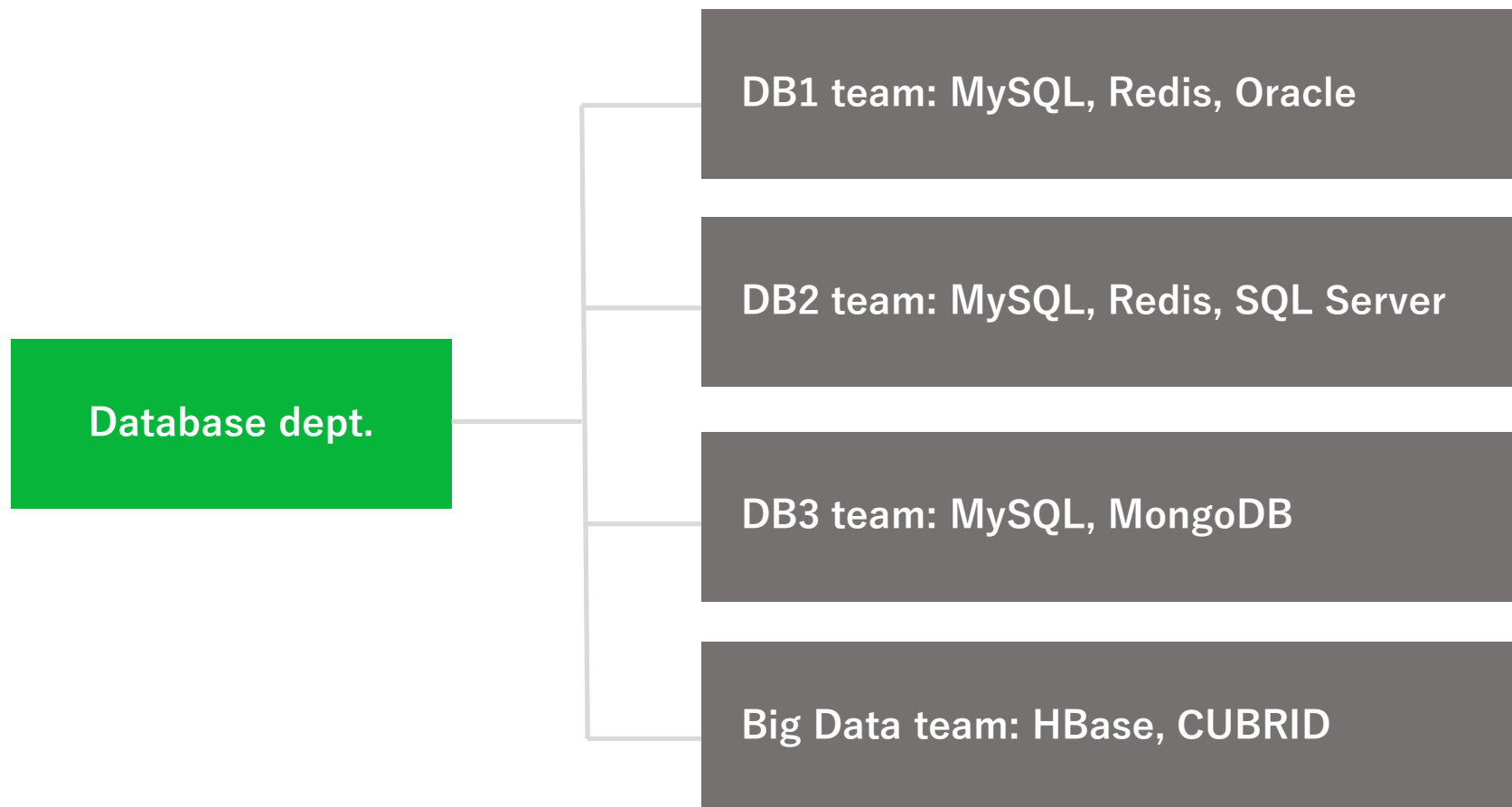
- 大塚 知亮 (@tom\_\_bo)
- IT Service Center, Database dept, DB1 team
- MySQL オペレーションと自動化  
DB管理システムの開発
- 2016: LINE Summer Internship
- 2017: LINE Engineer アルバイト
- 2018: 入社
- 好きなMySQLの機能は準同期レプリケーション！

# Database室紹介

# IT Service Centerの組織図



# Database室構成



# サポートしているDBMS



# DBMS運用の歴史

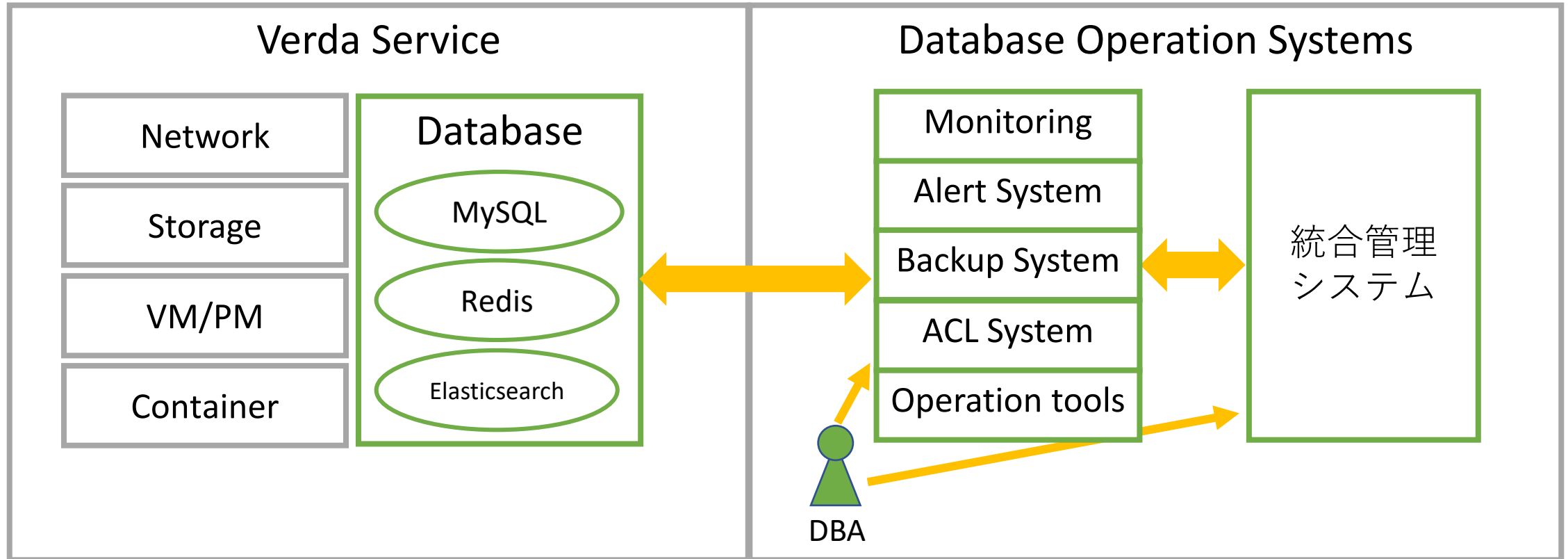




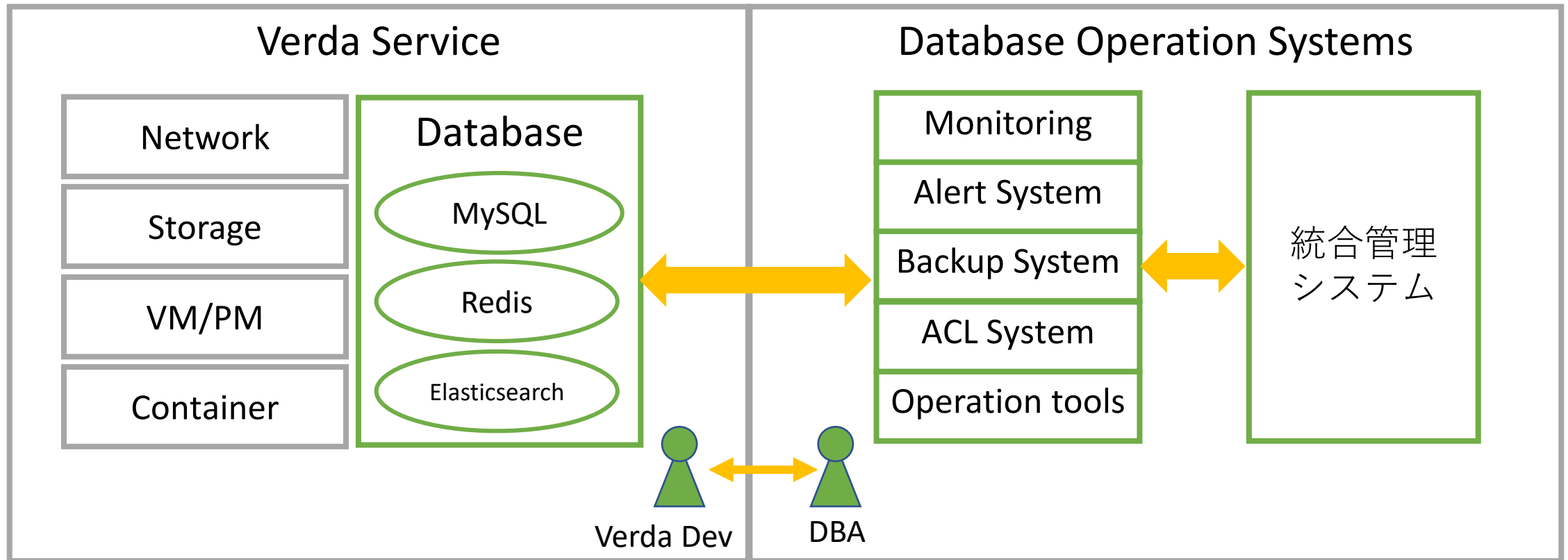
# 運用業務

- スキーマ管理
- ACL管理
- コンフィグ管理
- インデックス管理
- レプリケーショントポロジ管理
- マイグレーション作業
- モニタリング
- アラート対応
- クエリチューニング
- バックアップ・リカバリ
- 新機能・バグ調査
- 運用システムの開発

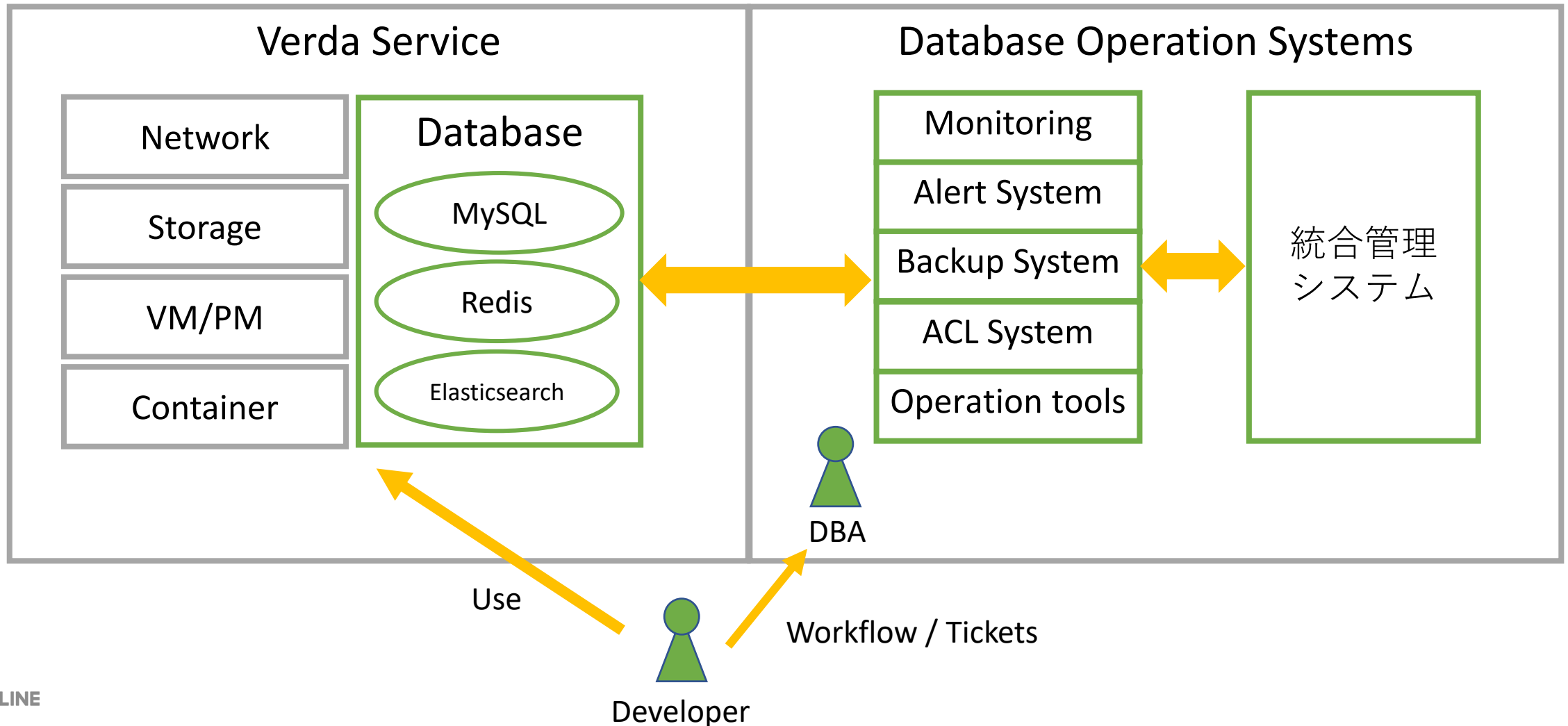
# Database関連システム



# Database関連システム



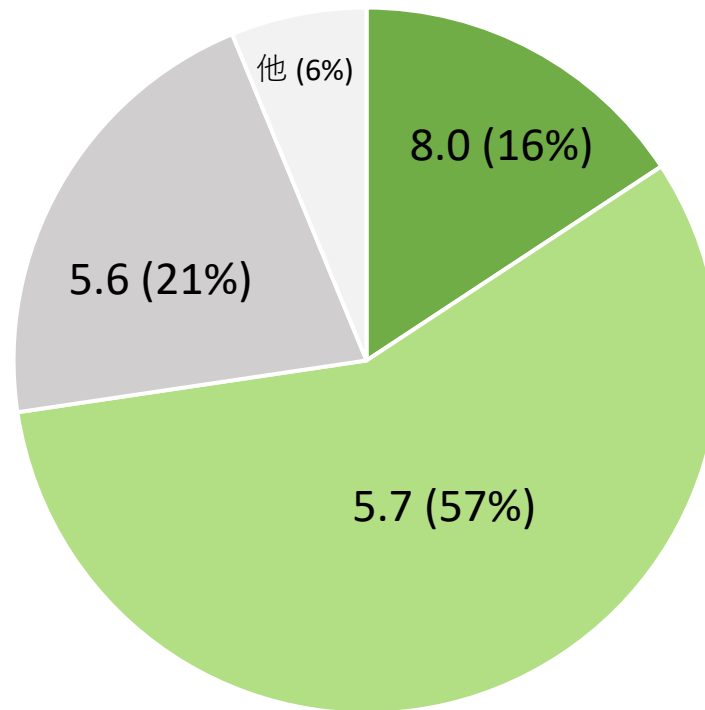
# Database関連システム



# MySQL Platformの変遷

# MySQLのバージョン比率

バージョンごとの台数比



# MySQLの増加



# MySQL Platform以前

- ~ 2017年
- サーバ調達から調整
- 急激なインスタンス増加でサーバスペックの統一が難しい
- Ansibleや各DBAのスクリプトで単純な自動化のみ



# MySQL Platform以前

- 2011 ~ 2017
- サーバ調達から調整
- 急激なインスタンス増加でサーバスペックの統一が難しい
- Ansibleや各DBAのスクリプトで単純な自動化のみ

## [課題]

- 開発者は作業のたびにWFを申請
- WFの対応が自動化されていない
- 担当するDBAによって運用方法が異なる

# MySQL Platform 1.0

- 2017 ~ 2020
- インスタンスタイプの標準化・統一
- 構築作業の自動化
- 単純なWFが自動化システムと連携

# MySQL Platform 1.0

- 2017 ~ 2020
- インスタンスタイプの標準化・統一
- 構築作業の自動化
- 単純なWFが自動化システムと連携

## [課題]

- 増え続けるサービスにDBA作業の限界
- インフラコスト削減のためのマイグレーション作業が頻発
- 規模の拡大によるアラートの増加

# MySQL Platform 2.0

- 2020年リリース (DBS for MySQL)
- 開発者によるセルフマネジメント
- オートヒーリング・トポロジ変更機能を開発者に提供
- 必要に応じてDBAがサポート
- クエリチューニングやIndex設計の社内教育を強化

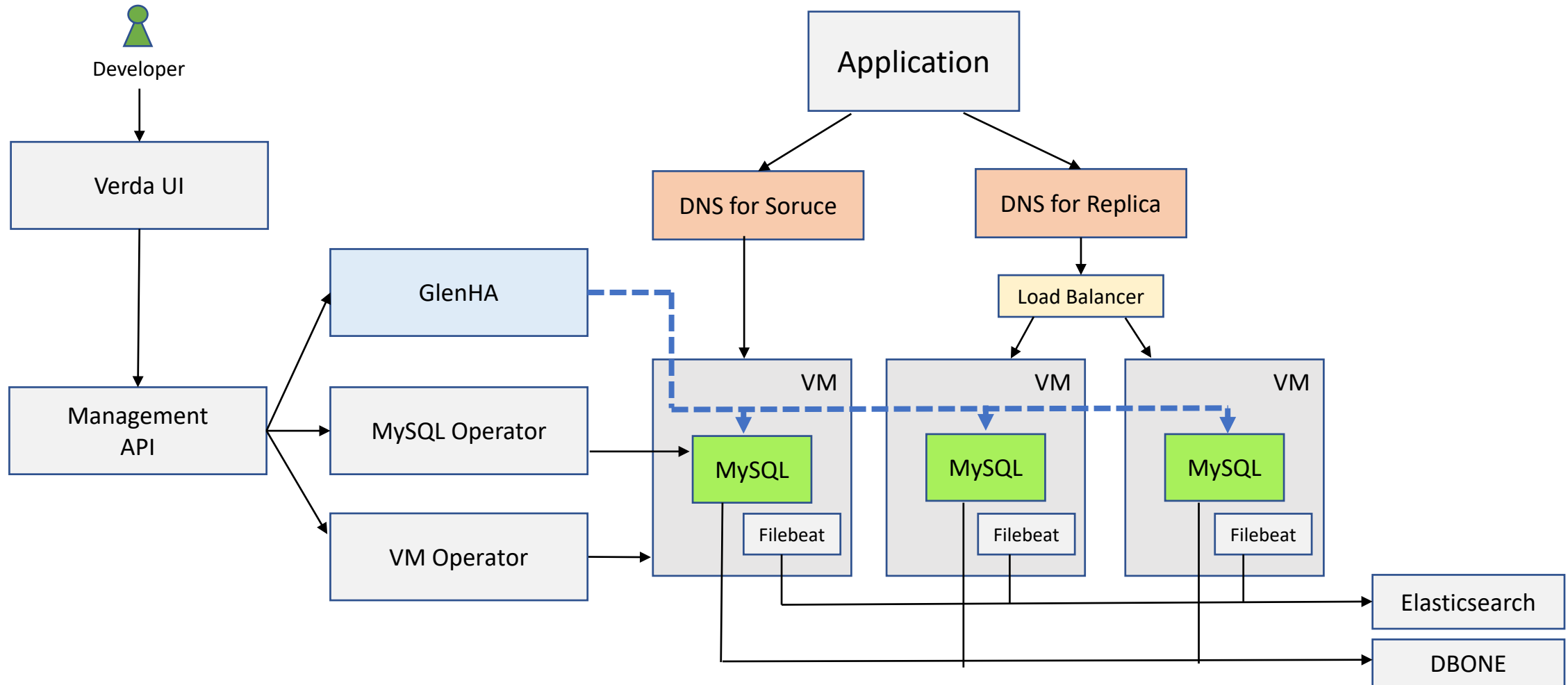
# MySQL Platform 2.0

- 2020年リリース (DBS for MySQL)
- 開発者によるセルフマネジメント
- オートヒーリング・トポロジ変更機能を開発者に提供
- 必要に応じてDBAがサポート
- クエリチューニングやIndex設計の社内教育を強化

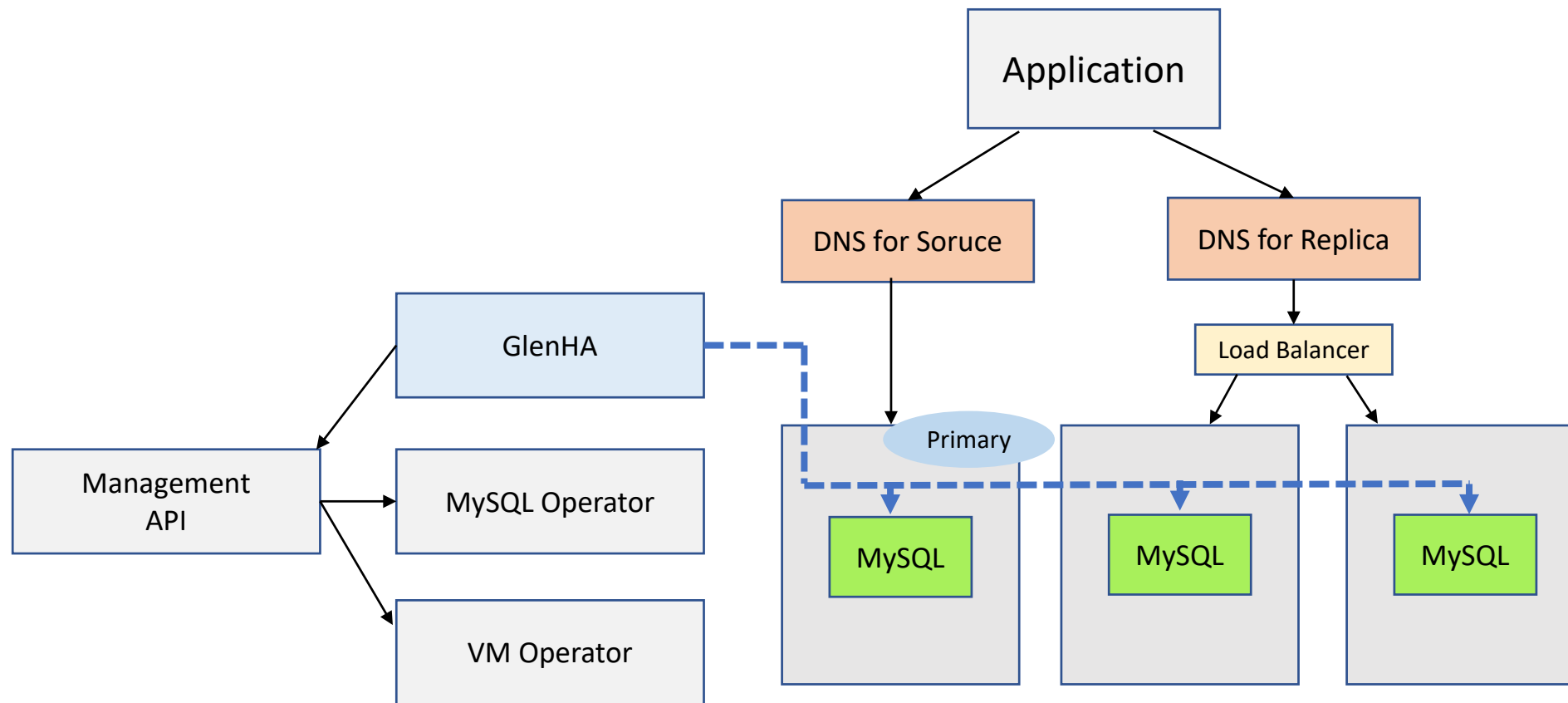
## [GlenDBの開発]

- オートヒーリング・インスタンスタイプ変更を自動化
- トポロジ変更操作、確認がWeb画面から可能
- HA managerをスクラッチで開発

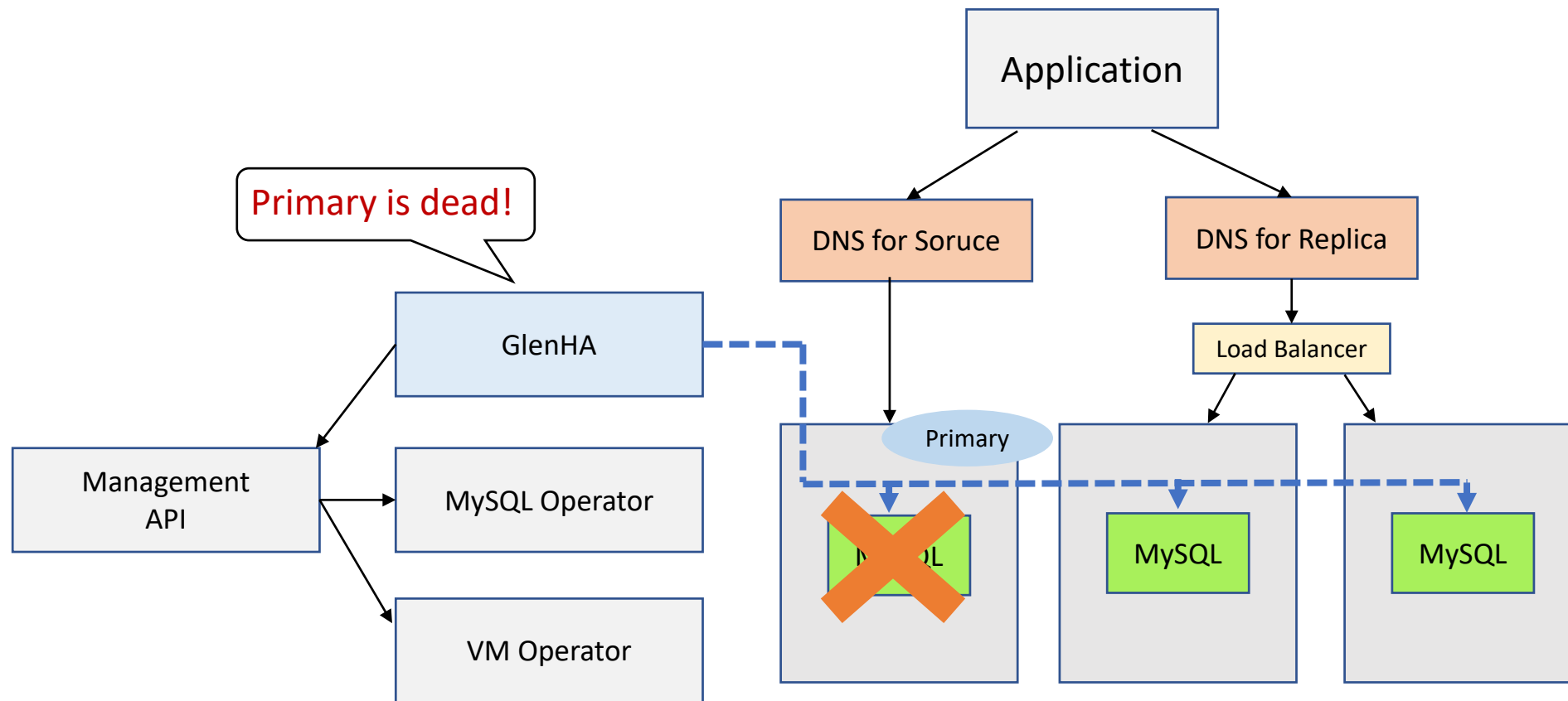
# GlenDB



# Auto Healing

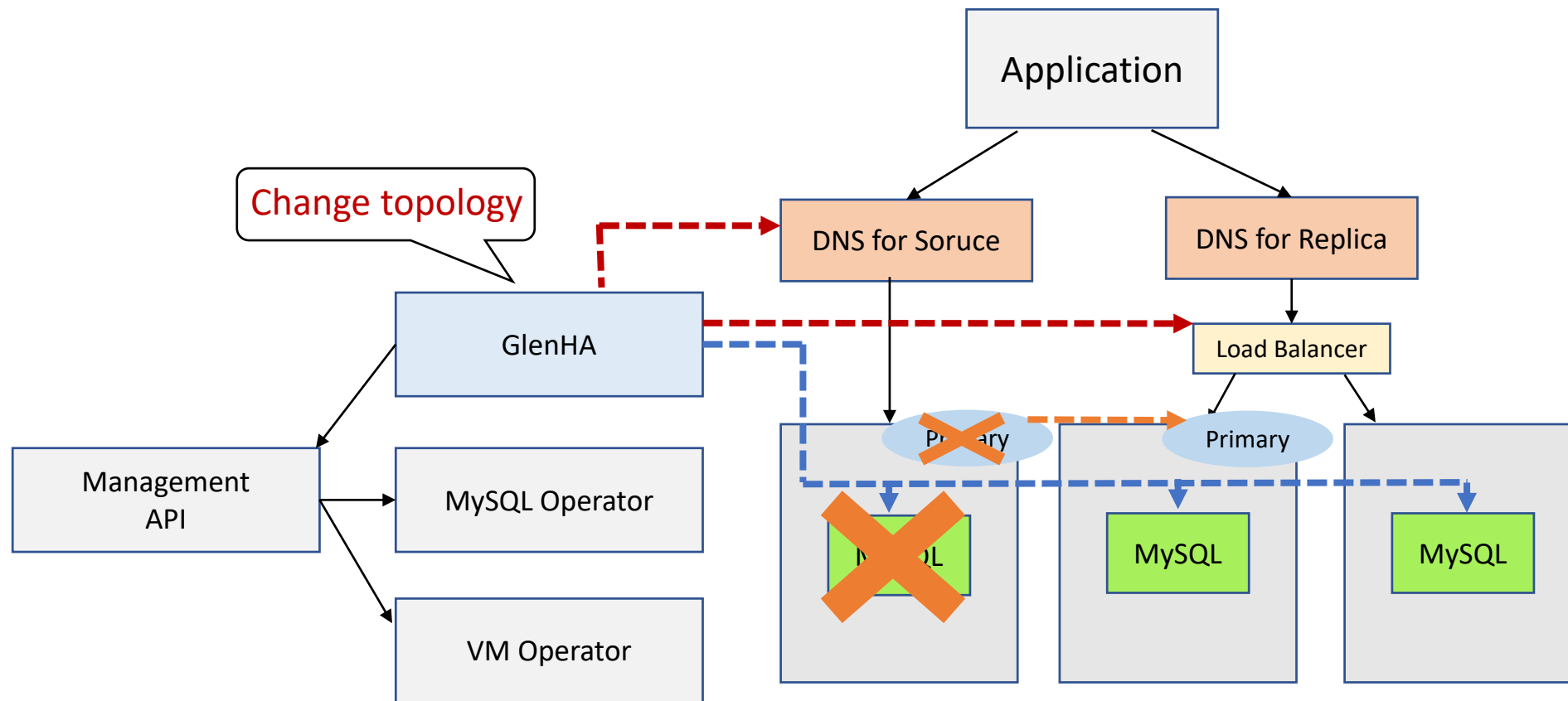


# Auto Healing

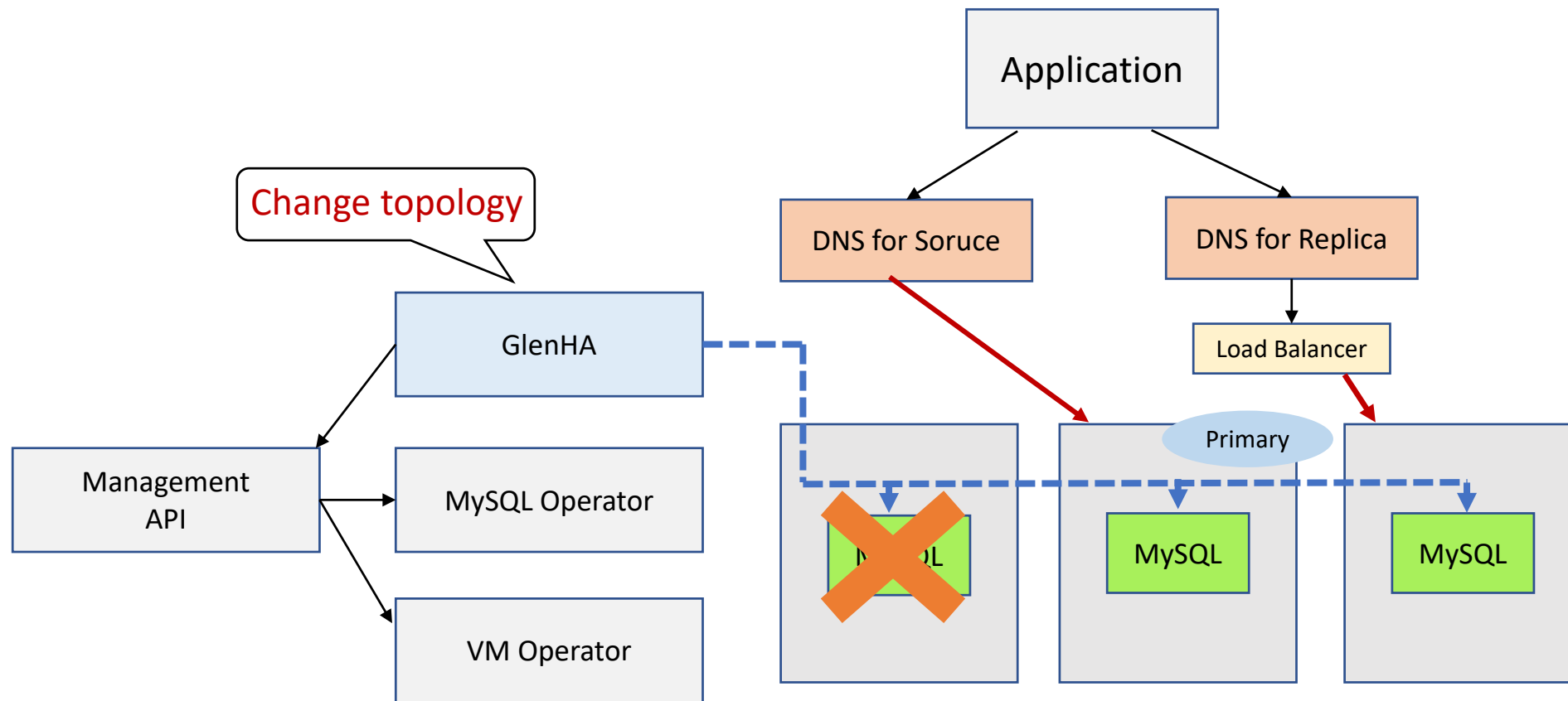




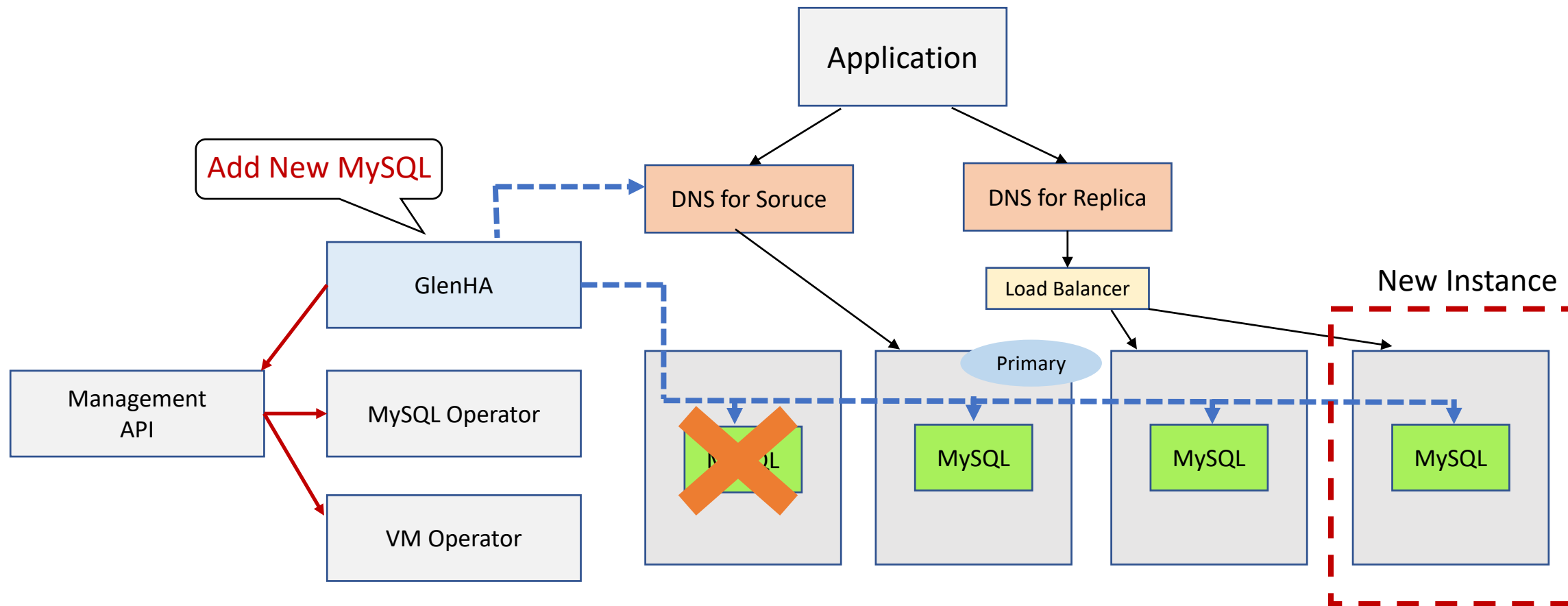
# Auto Healing



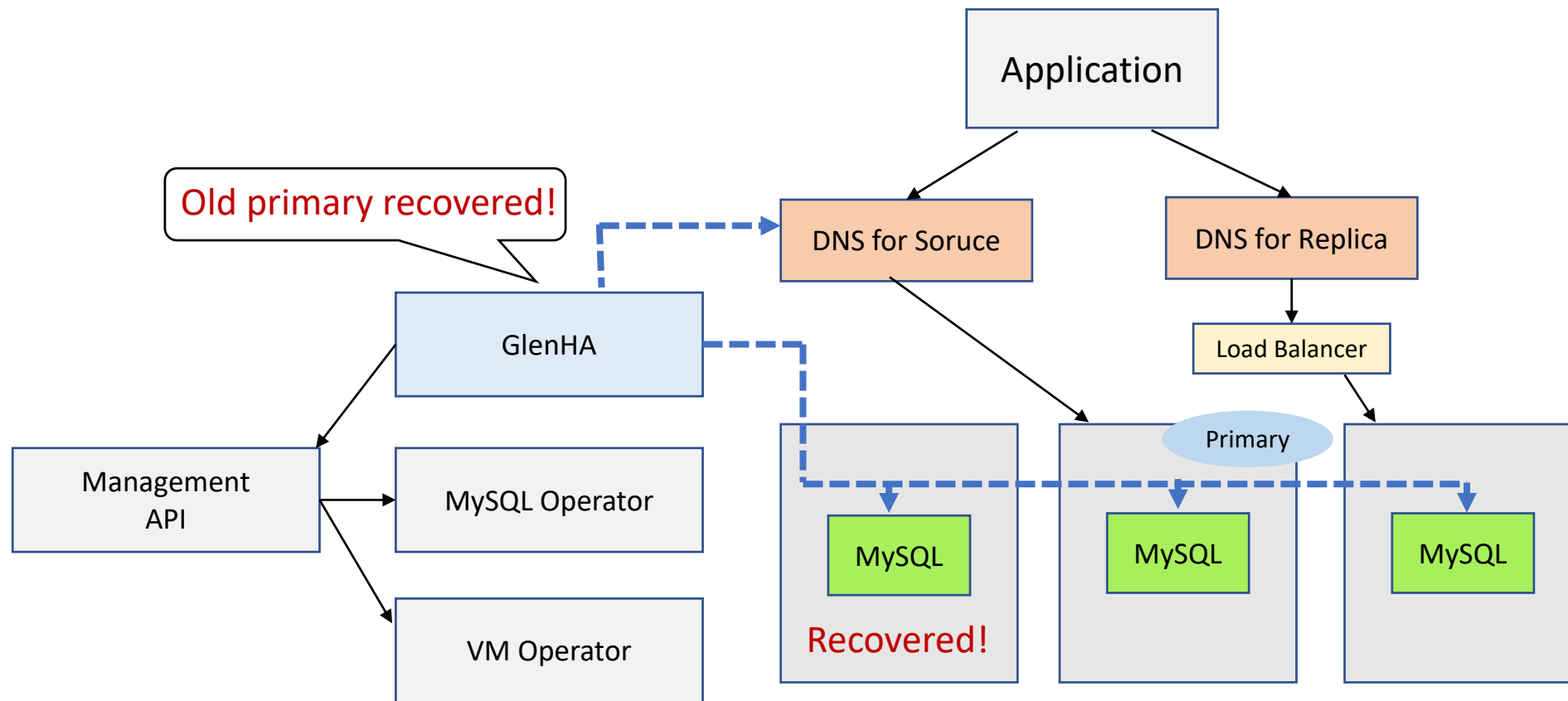
# Auto Healing



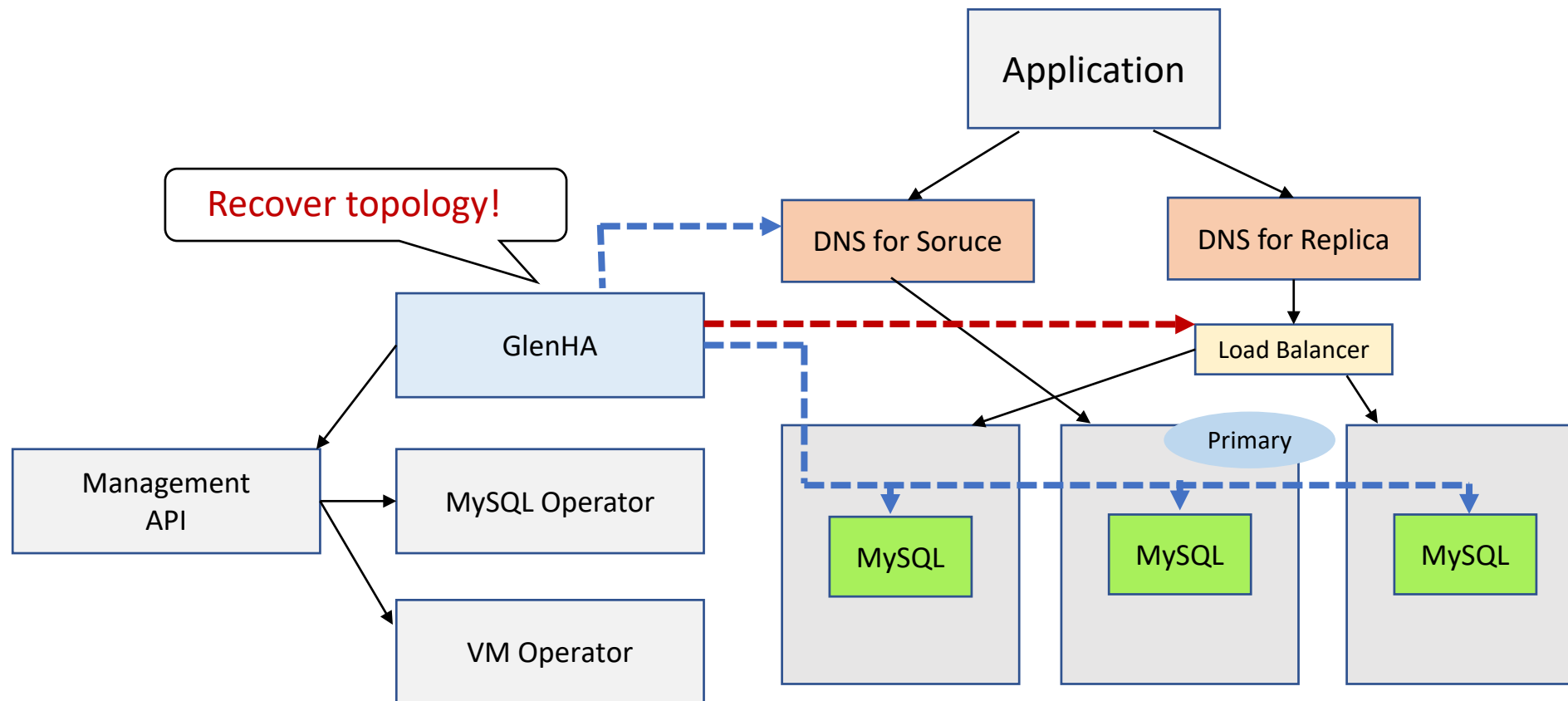
# Auto Healing



# Auto Healing



# Auto Healing



# Sample view

**Verda**<sup>Dev</sup>

Sample DBS ▾

DBS for MySQL ▾

Support ▾

testuser ▾

Announce

Maintenance

VBS

VBS-dev planned maintenance 2020-11-12 14:00~20:00 (UTC+9)

MonDB-API / DBS for MySQL

MySQL Services 

+

DBS for MySQL User Guide ⓘ

Instance name

Reset

+ Create Service

Instance name ▴	Role	Version	Flavor	Status	Time since created	Dev admin	Actions
otsuka-test-1	Primary	8.0.19	4vCPU_8GB_100GB_SSD	Running	4 months, 2 weeks	Otsuka Tomoaki	Actions ▾
otsuka-test-2	Standby replica	8.0.19	4vCPU_8GB_100GB_SSD	Running	2 months, 3 weeks	Otsuka Tomoaki	Actions ▾

10 /page ▾

< 1 >

Failover

Restart instance

Change Dev admin

Scale read replica

Delete service

# バージョンアップを支える仕組み

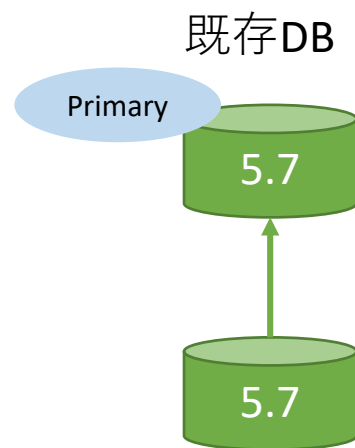
# バージョンアップポリシーの設定

- 運用作業に追われて最近までポリシーがなかった
- サポート期限切れのインスタンスは潜在リスク
- 古いバージョンがあることでサポートや自動化のコストが高い



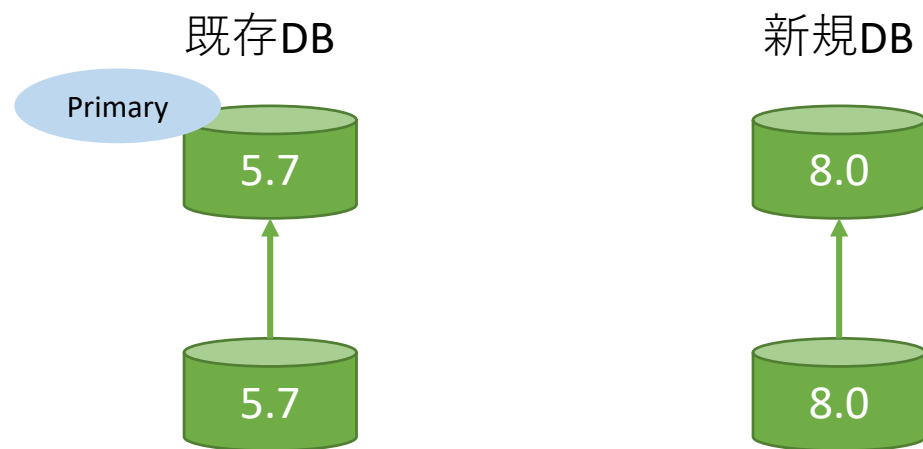
# メジャーバージョンアップの方法

- 論理バックアップを利用して新しいバージョンで構築
- データとACLは別々に移行
- フェイルオーバーにより切り替え



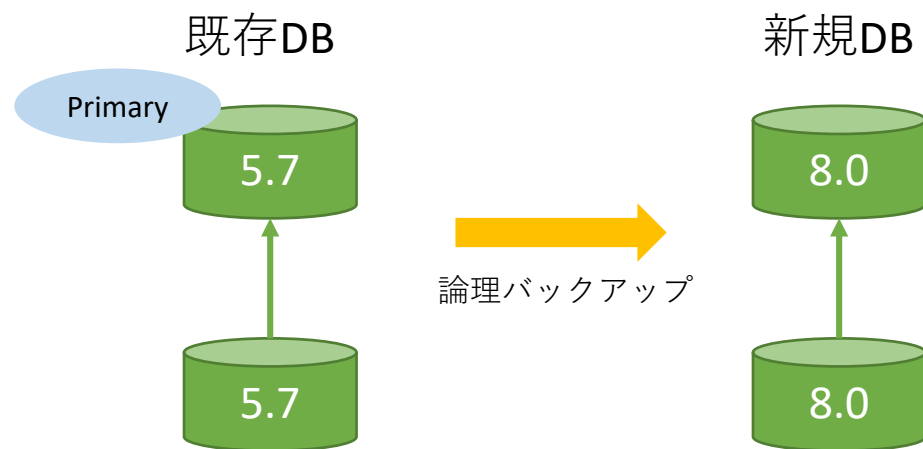
# メジャーバージョンアップの方法

- 論理バックアップを利用して新しいバージョンで構築
- データとACLは別々に移行
- フェイルオーバーにより切り替え



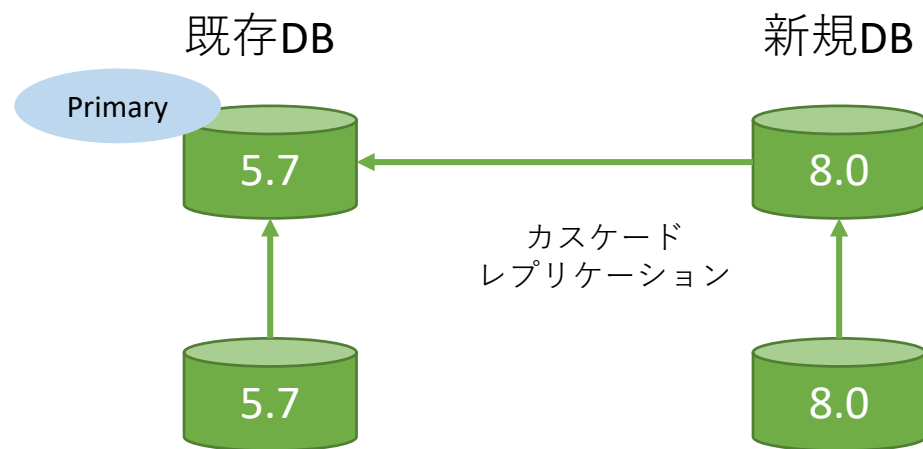
# メジャーバージョンアップの方法

- 論理バックアップを利用して新しいバージョンで構築
- データとACLは別々に移行
- フェイルオーバーにより切り替え



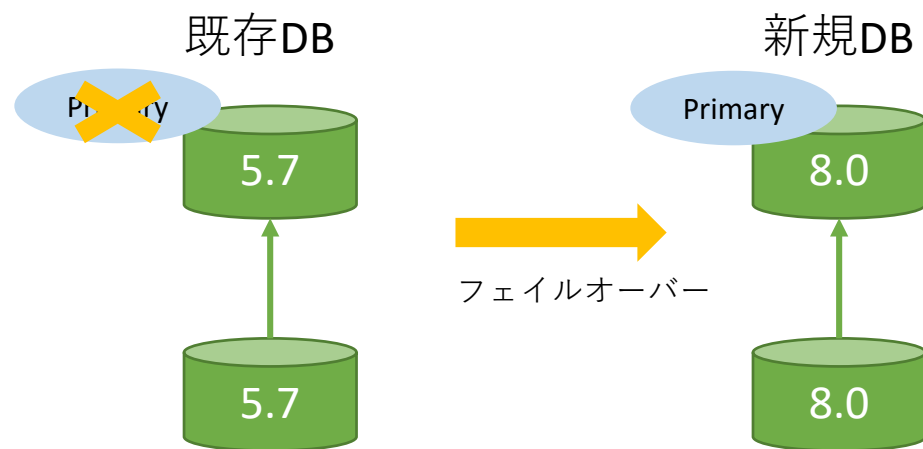
# メジャーバージョンアップの方法

- 論理バックアップを利用して新しいバージョンで構築
- データとACLは別々に移行
- フェイルオーバーにより切り替え



# メジャーバージョンアップの方法

- 論理バックアップを利用して新しいバージョンで構築
- データとACLは別々に移行
- フェイルオーバーにより切り替え



# メジャーバージョンアップの流れ

1. テスト環境でバージョンアップテスト
2. 移行先MySQLの作成
3. データ移行（論理バックアップをリストア）
4. ACL移行
5. データ・メタデータの確認
6. クエリのデグレード確認
7. 移行用レプリケーションの設定
8. 移行先へフェイルオーバー

# アップグレード作業への備え

- 1000台以上、数百のサービスを移行するプロジェクト
- バージョンアップと同時にプラットフォーム移行
- 作業コスト・作業ミスを最小化

# アップグレード作業への備え

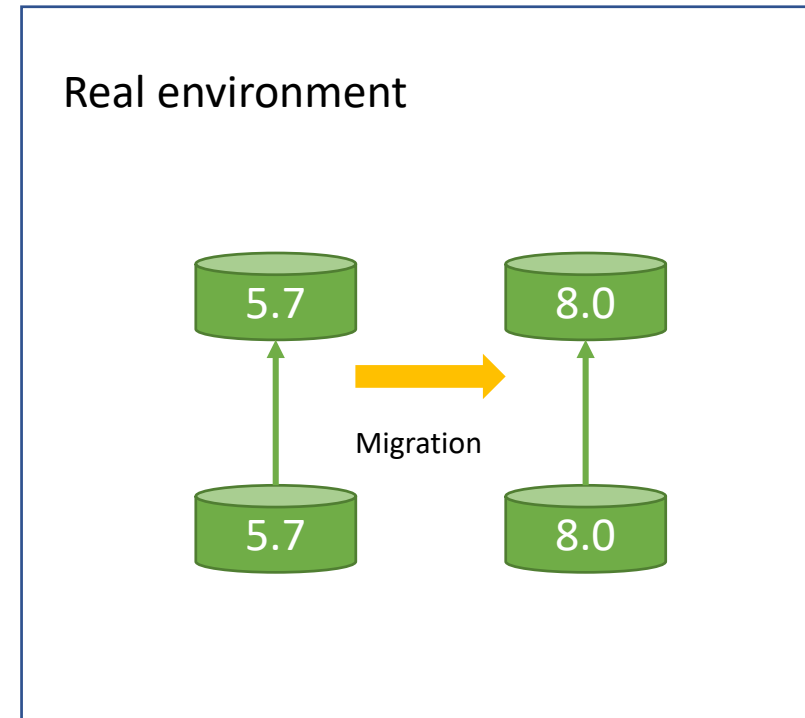
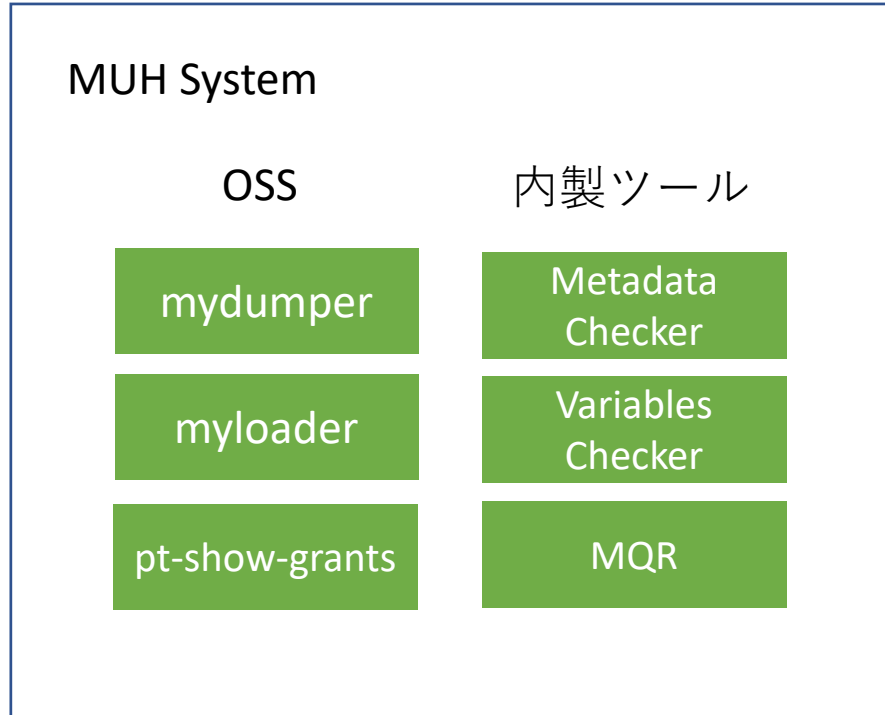
- 1000台以上、数百のサービスを移行するプロジェクト
- バージョンアップと同時にプラットフォーム移行
- 作業コスト・作業ミスを最小化

[MySQL Upgrade Helperの開発]

- マイグレーション作業を自動化
- OSSや既存のツールを組み合わせ高速に開発

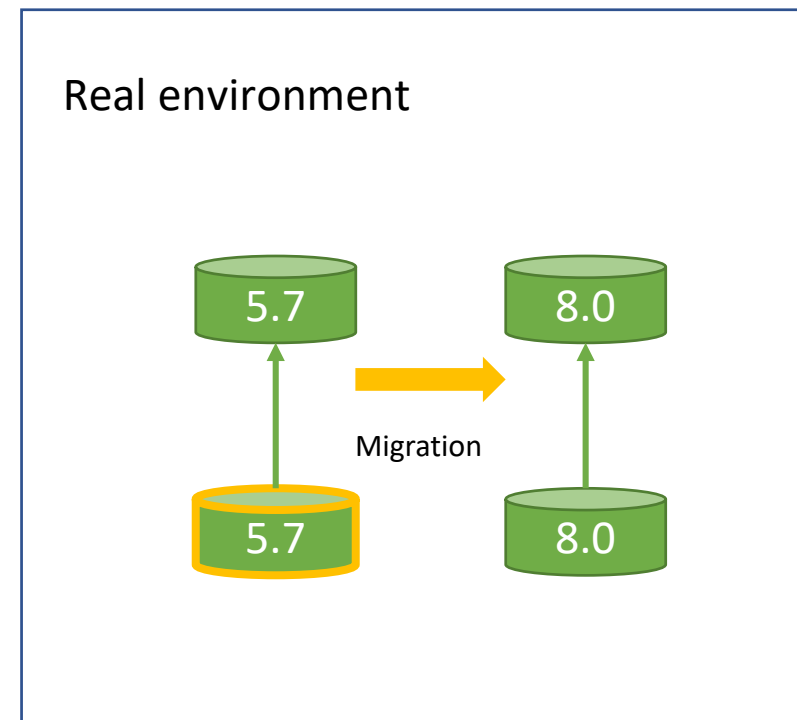
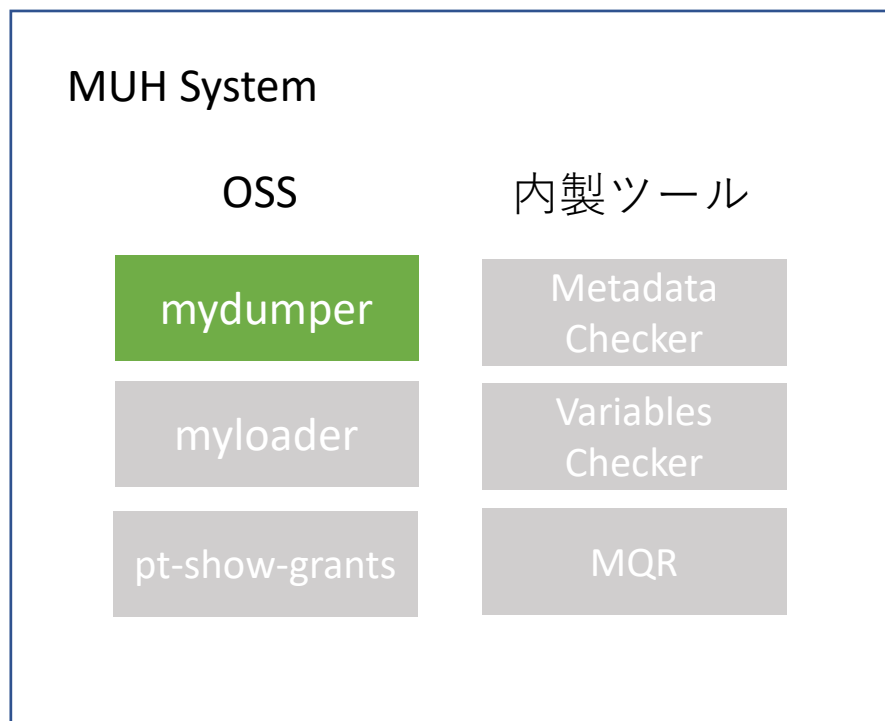


# MUH: MySQL Upgrade Helper



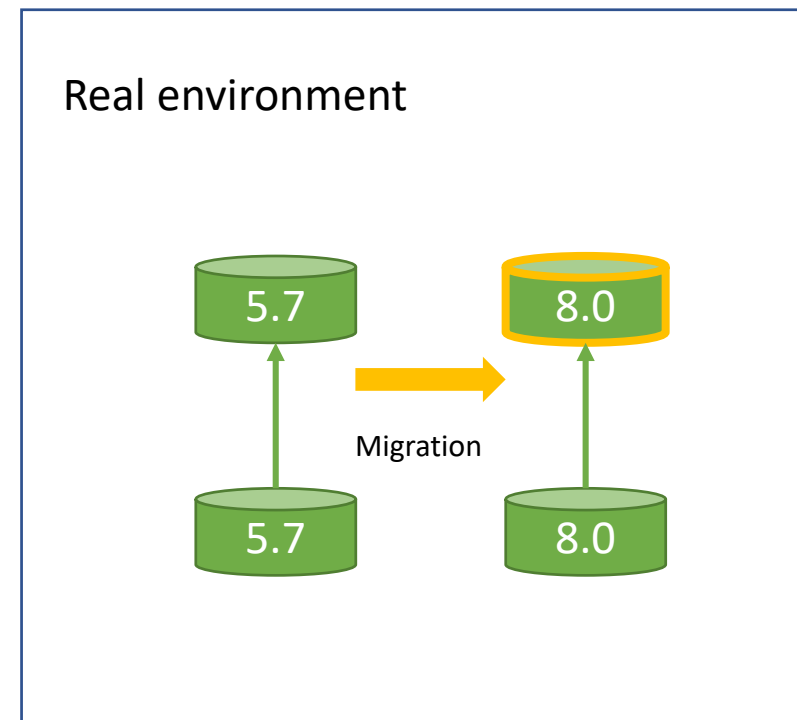
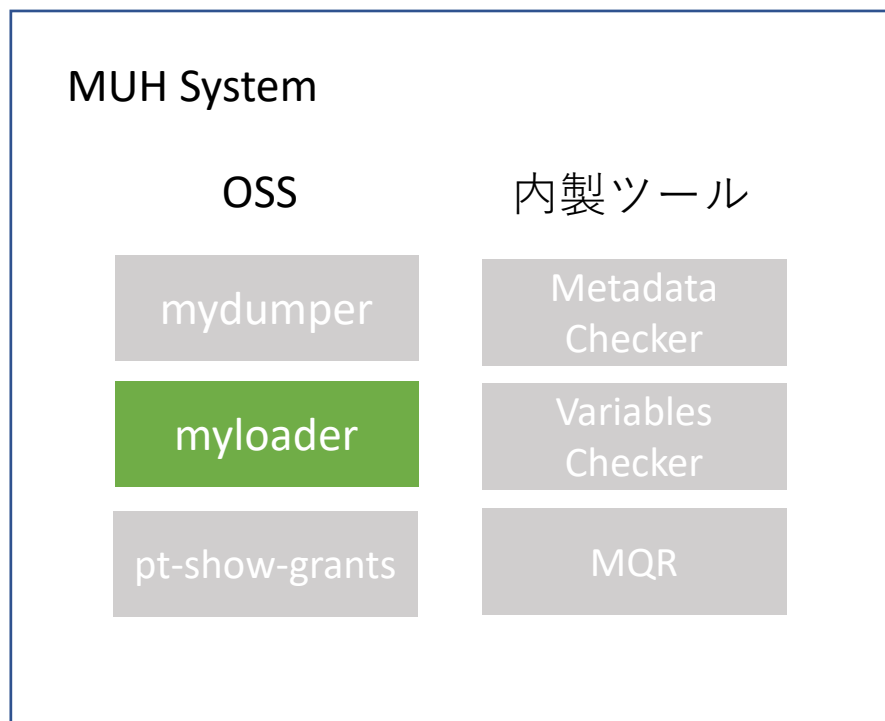
# MUH: MySQL Upgrade Helper

- Replicaからデータをダンプ



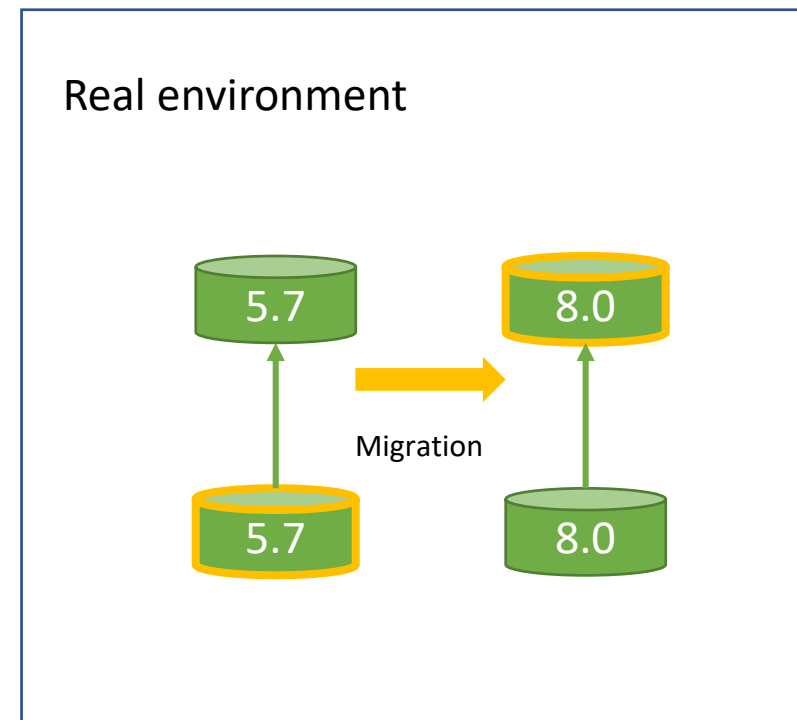
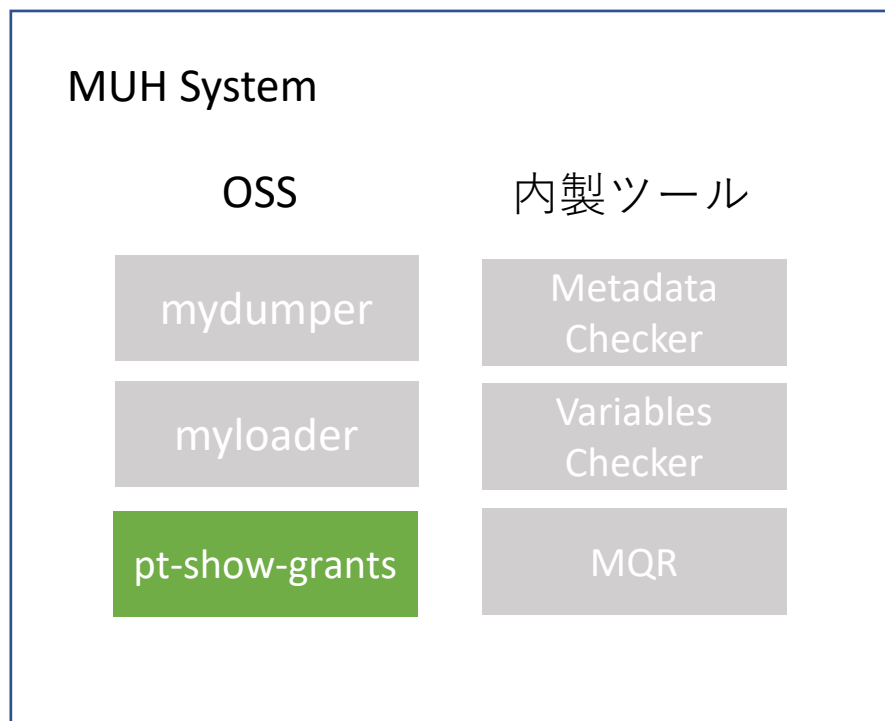
# MUH: MySQL Upgrade Helper

- ダンプしたデータを移行先にロード



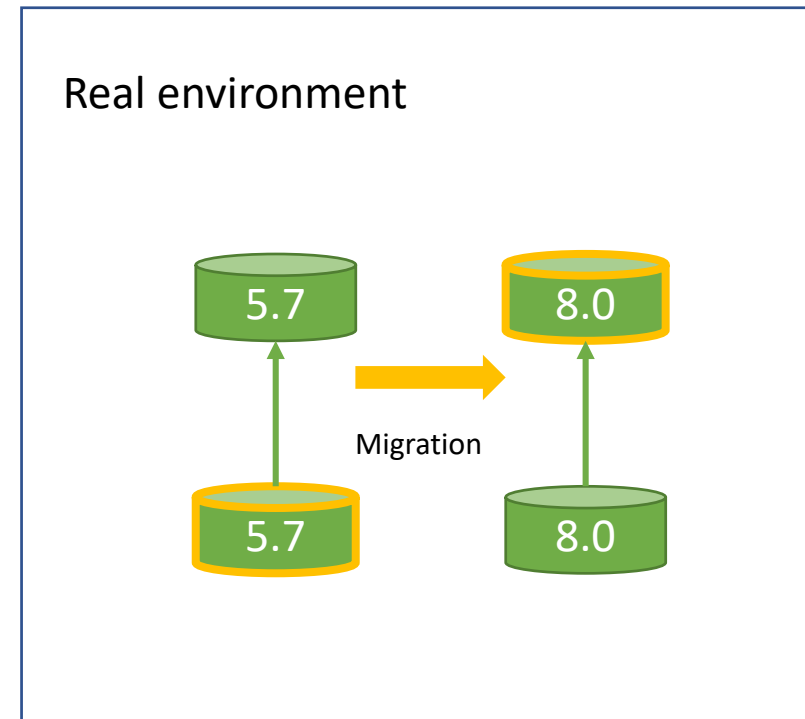
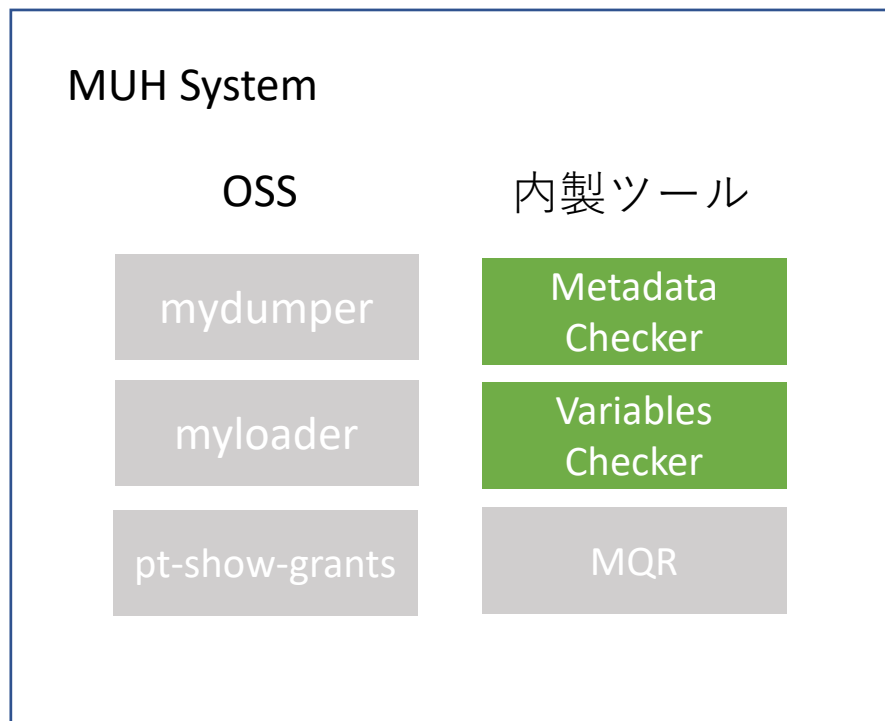
# MUH: MySQL Upgrade Helper

- 移行先が必要なACL情報を確認し移行



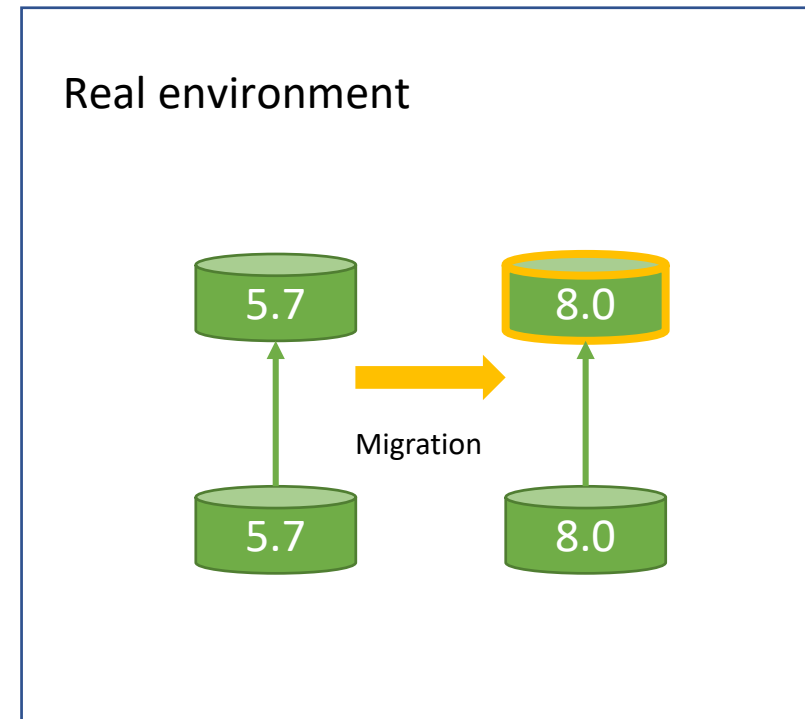
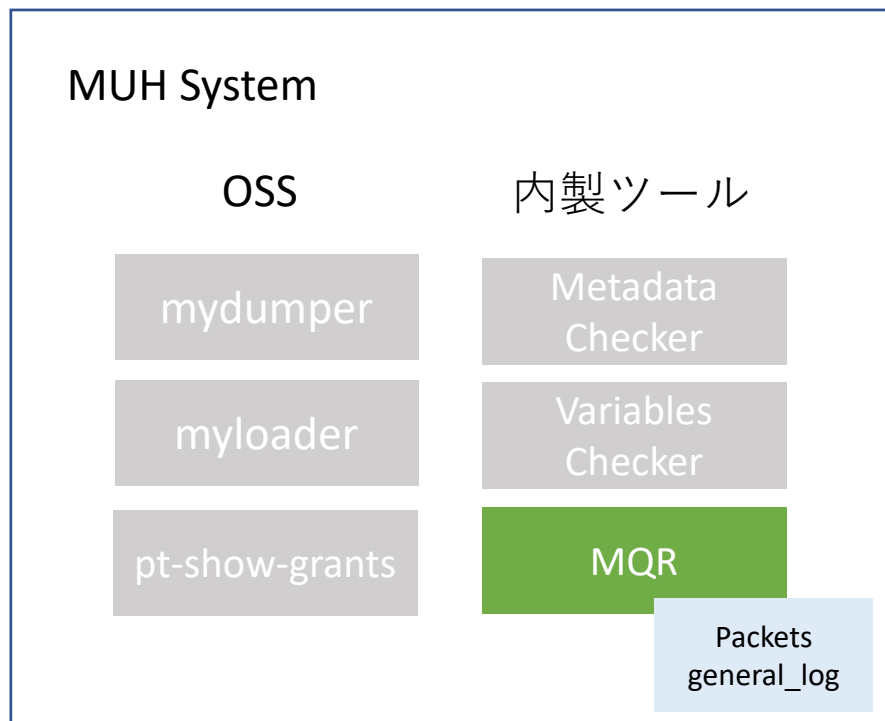
# MUH: MySQL Upgrade Helper

- メタデータ・Variablesの差分を確認



# MUH: MySQL Upgrade Helper

- Network packetsかgeneral\_logを取得して新環境に再現



# MUH: MySQL Upgrade Helper

MUH-Web

RegisterReferenceVerda2.0Feedback

# 41

Source

test-otsuka-source-  
hostname:3306

Target

test-otsuka-target-  
hostname:3306

DBA

1.Otsuka Tomoaki

Time

2020-10-05 18:16:19

Delete

MuhUser

muh\_dummysring

Source

3

Target

3

Delete Users

Variables Difference

Check

User Migration

Add

Check

#	Time	Status	IncludeUser	Comment	View
173	2020-10-05 18:20:36	FINISH	sysbench	Some Grant statement Failed. Check logfile[/mysql/mydumper/173/Failed.log]	HistoryExportLogImportLog

Database Migration

Add

#	Time	Status	Databases	Threads	Migrate	Comment	View
172	2020-10-05 18:18:58	FINISH	sysbench	5 / 5	Schema & Data		HistoryExportLogImportLogReplication

Metadata Difference

Add

#	Time	Status	Databases	MetadataDiff	TableDiff	Error	View
87	2020-10-05 18:21:25	FINISH	sysbench	0	0		No Differences

MQR(Query Replayer)

Add

#	Queued	Start	End	Status	Database	Comment	Function
16	2020-10-15 11:02:16	2020-10-15 11:02:18	2020-10-15 11:02:40	Completed	sysbench		Result
15	2020-10-14 17:20:28	2020-10-14 17:20:28	2020-10-14 17:20:49	Completed	sysbench		Result
10	2020-10-13 16:48:29	2020-10-13 16:48:30	2020-10-13 16:48:52	Completed	sysbench		Result
4	2020-10-08 14:44:37	2020-10-08 14:44:42	2020-10-08 14:45:04	Completed	sysbench		Result

# MQR: MySQL Query Replayer

## ⚡ MQR Result

mqr\_task\_id: # 62 , mqr\_run\_id: # 23

Search:

#	Total(sec)	ExecCnt	Avg(sec)	Max(sec)	FingerPrint	Histogram	Error
1	3.419	13,141	0.000	0.002	SELECT c FROM sbtest1 WHERE id=?	<a href="#">Histogram</a>	0
2	3.377	13,007	0.000	0.002	SELECT c FROM sbtest2 WHERE id=?	<a href="#">Histogram</a>	0
3	2.956	11,390	0.000	0.002	SELECT c FROM sbtest3 WHERE id=?	<a href="#">Histogram</a>	0
4	1.356	1,280	0.001	0.002	SELECT DISTINCT c FROM sbtest2 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
5	1.346	1,281	0.001	0.002	SELECT DISTINCT c FROM sbtest1 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
6	1.266	1,192	0.001	0.003	SELECT DISTINCT c FROM sbtest3 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
7	0.813	1,260	0.001	0.007	SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
8	0.800	1,253	0.001	0.002	SELECT c FROM sbtest2 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
9	0.797	1,240	0.001	0.002	SELECT c FROM sbtest3 WHERE id BETWEEN ? AND ? ORDER BY c	<a href="#">Histogram</a>	0
10	0.654	1,274	0.001	0.004	SELECT c FROM sbtest2 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
11	0.633	1,247	0.001	0.001	SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
12	0.622	1,232	0.001	0.002	SELECT c FROM sbtest3 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
13	0.458	1,293	0.000	0.001	SELECT SUM(k) FROM sbtest1 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
14	0.436	1,227	0.000	0.002	SELECT SUM(k) FROM sbtest2 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
15	0.434	1,233	0.000	0.001	SELECT SUM(k) FROM sbtest3 WHERE id BETWEEN ? AND ?	<a href="#">Histogram</a>	0
16	0.001	1	0.001	0.001	SELECT @@log_bin	<a href="#">Histogram</a>	0
17	0.000	1	0.000	0.000	SELECT @@version	<a href="#">Histogram</a>	0
18	0.000	0	0.000	0.000	SELECT ?, IFNULL(MAX(TIME_TO_SEC(TIMEDIFF(NOW(), trx_started))), ?) AS MAX_RUNTIME FROM INFORMATION_SCH...	<a href="#">Histogram</a>	1
19	0.000	0	0.000	0.000	SELECT name, subsystem, type, comment, count FROM information_schema.innodb_metrics WHERE status = ?	<a href="#">Histogram</a>	1



# MQR: MySQL Query Replayer

## ⚡ MQR Result

mqr\_task\_id: # 62 , mqr\_run\_id: # 23

Search:

#	Total(sec)	ExecCnt	Avg(sec)	Max(sec)	FingerPrint	Histogram	Error
1	3.419	13,141	0.000	0.002	SELECT c FROM sbtest1 WHERE id=?	<a href="#">Histogram</a>	0
2	3.377	13,007	0.000	0.002	SELECT c FROM sbtest2 WHERE id=?	<a href="#">Histogram</a>	0
3	2.956	11,390				<a href="#">Histogram</a>	0
4	1.356	1,280				<a href="#">Histogram</a>	0
5	1.346	1,281				<a href="#">Histogram</a>	0
6	1.266	1,192				<a href="#">Histogram</a>	0
7	0.813	1,260				<a href="#">Histogram</a>	0
8	0.800	1,253				<a href="#">Histogram</a>	0
9	0.797	1,240				<a href="#">Histogram</a>	0
10	0.654	1,274				<a href="#">Histogram</a>	0
11	0.633	1,247				<a href="#">Histogram</a>	0
12	0.622	1,232				<a href="#">Histogram</a>	0
13	0.458	1,293				<a href="#">Histogram</a>	0
14	0.436	1,227				<a href="#">Histogram</a>	0
15	0.434	1,233				<a href="#">Histogram</a>	0
16	0.001	1				<a href="#">Histogram</a>	0
17	0.000	1				<a href="#">Histogram</a>	0
18	0.000	0				<a href="#">Histogram</a>	0
19	0.000	0	0.000	0.000	SELECT name, subsystem, type, comment, count FROM information_schema.innodb_metrics WHERE status = ?	<a href="#">Histogram</a>	1

### # Histogram

Time Interval	Frequency (approx.)
10s+	0
<10s	0
<1s	0
<100ms	500
<10ms	4500
<1ms	13000
<100us	100
<10us	50

# メジャーバージョンアップの流れ

- テスト環境でバージョンアップテスト(dev環境)
- 移行先MySQLの作成
- データ移行 (論理バックアップをリストア)
- ACL移行
- データ・メタデータの確認
- デグレードチェック
- 移行用レプリケーションの設定
- 移行先へフェイルオーバー



# アップグレード時(8.0)の問題例

- mysqldumpコマンドでMySQL 5.7からデータをダンプして8.0にインポート
  - テーブルの照合順序(collation)が変わってしまった
  - 5.7(utf8mb4\_general\_ci) => 8.0(utf8mb4\_0900\_ai\_ci)

Version	character_set_server	collation_server
5.7	utf8mb4	utf8mb4_general_ci
8.0	utf8mb4	utf8mb4_general_ci

表: 問題が発生したときのMySQLの設定値

# アップグレード時(8.0)の問題例

- 8.0で設定可能になった `default_collation_for_utf8mb4` の設定が原因
  - `CREATE TABLE` や `ALTER TABLE` 文などで `CHARACTER SET utf8mb4` 句があって `COLLATION` 句を指定していないときに利用されるデフォルトのcollation

Version	default_collation_for_utf8mb4	comment
5.7	-	暗黙にutf8mb4_general_ci
8.0	utf8mb4_0900_ai_ci	明示的に設定可能

表: default\_collation\_for\_utf8mb4のdefault値

# アップグレード時(8.0)の問題例

Version	character_set_server	collation_server	default_collation_for_utf8mb4
5.7	utf8mb4	utf8mb4_general_ci	(暗黙にutf8mb4_general_ci)
8.0	utf8mb4	utf8mb4_general_ci	utf8mb4_0900_ai_ci

表: 問題が発生したときのMySQLの設定値

```
# at MySQL 5.7
> create table tbl(id int not null);

> SHOW CREATE TABLE tbl\G
***** 1. row *****
      Table: tbl
Create Table: CREATE TABLE `tbl` (
  `id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
1 row in set (0.01 sec)
```

# アップグレード時(8.0)の問題例

Version	character_set_server	collation_server	default_collation_for_utf8mb4
5.7	utf8mb4	utf8mb4_general_ci	(暗黙にutf8mb4_general_ci)
8.0	utf8mb4	utf8mb4_general_ci	utf8mb4_0900_ai_ci

表: 問題が発生したときのMySQLの設定値

```
# at MySQL 8.0 (import dumped sql)
> CREATE TABLE `tbl` ( `id` int(11) NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

> SHOW CREATE TABLE tbl\G
***** 1. row *****

Table: tbl
Create Table: CREATE TABLE `tbl` (
  `id` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```

# アップグレード時(8.0)の問題例

- 対策
  - `default_collation_for_utf8mb4` を適切に設定する
  - 移行後にTableのcollationを変更する
- 余談
  - utf8mb4\_0900\_ai\_ciだと ”びょういん” と “びょういん”  
などが区別できない
  - “MySQL 8.0でもutf8mb4\_general\_ciを使いたい僕らは”  
<https://mita2db.hateblo.jp/entry/2020/12/07/000000>

# オペレーションを支えるツール



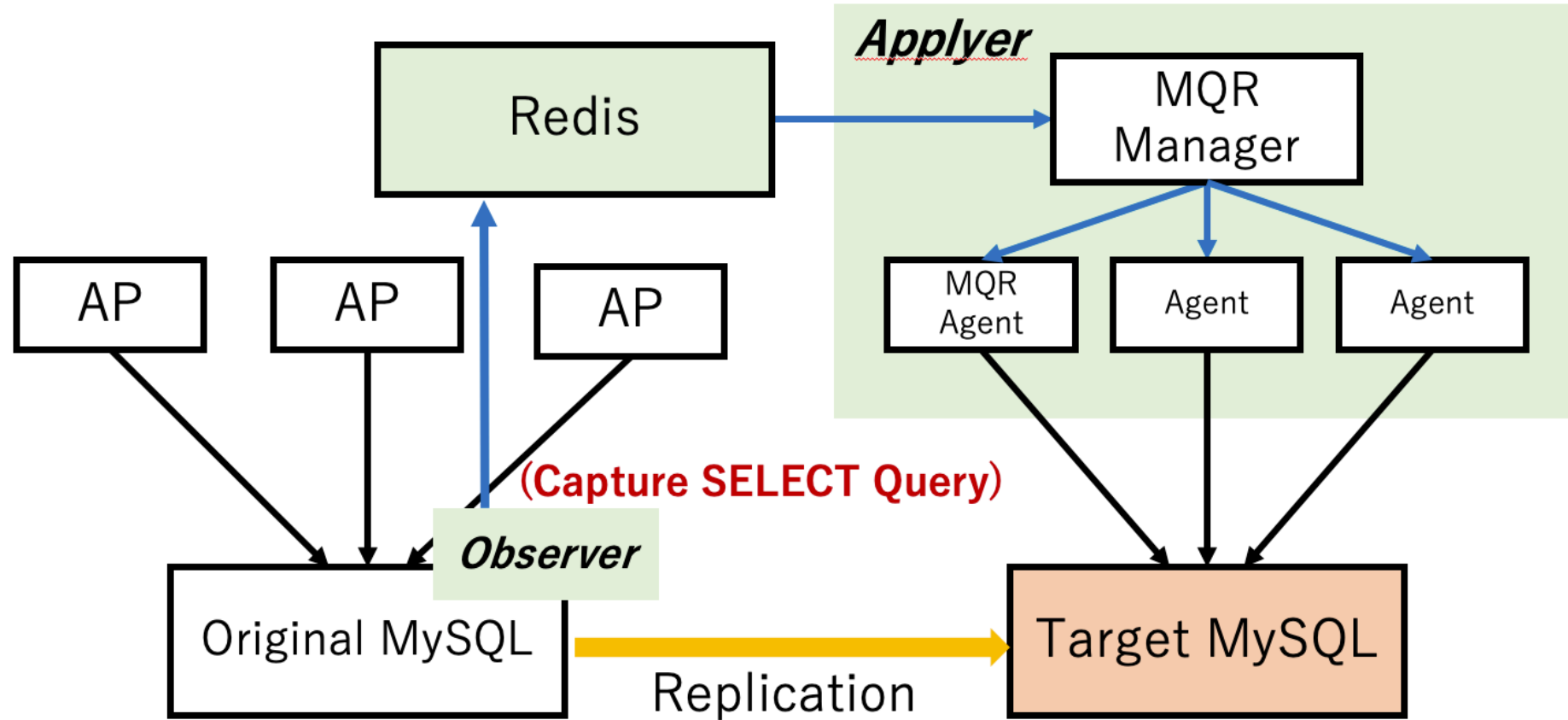
# オペレーションを支えるツール

- mysql\_query\_replayer
  - GitHub: <https://github.com/tom--bo/mysql-query-replayer>
  - ネットワークパケットからクエリを抽出し、別のMySQLにクエリを再現することでデグレードを確認
- myStatusgo
  - GitHub: <https://github.com/kenken0807/myStatusgo>
  - インスタンス情報を1秒間隔で表示するMySQL版topコマンド

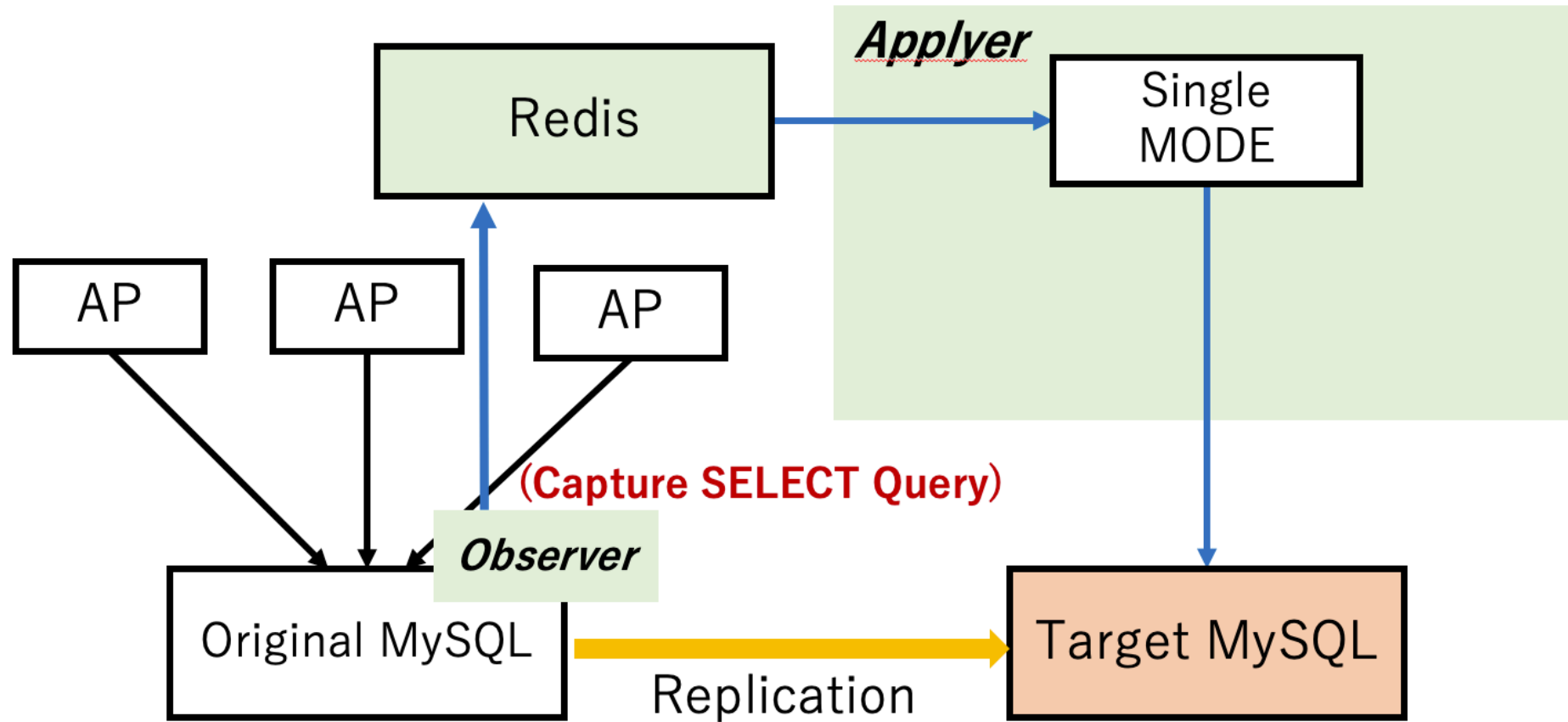
# MQR (MySQL Query Replayer)

- GitHub: <https://github.com/tom--bo/mysql-query-replayer>
- ネットワークパケットからクエリを抽出し、別のMySQLにクエリを再現することでデグレードを確認
- クエリの再現方法によって3つのモードを選択可能
  - Multi Mode: リアルタイムに本番環境の負荷を再現する
  - Single Mode: Queuing MWを使わずにシンプルに動作
  - mpReader: パケットのダンプファイルを引数にコマンドとして実行

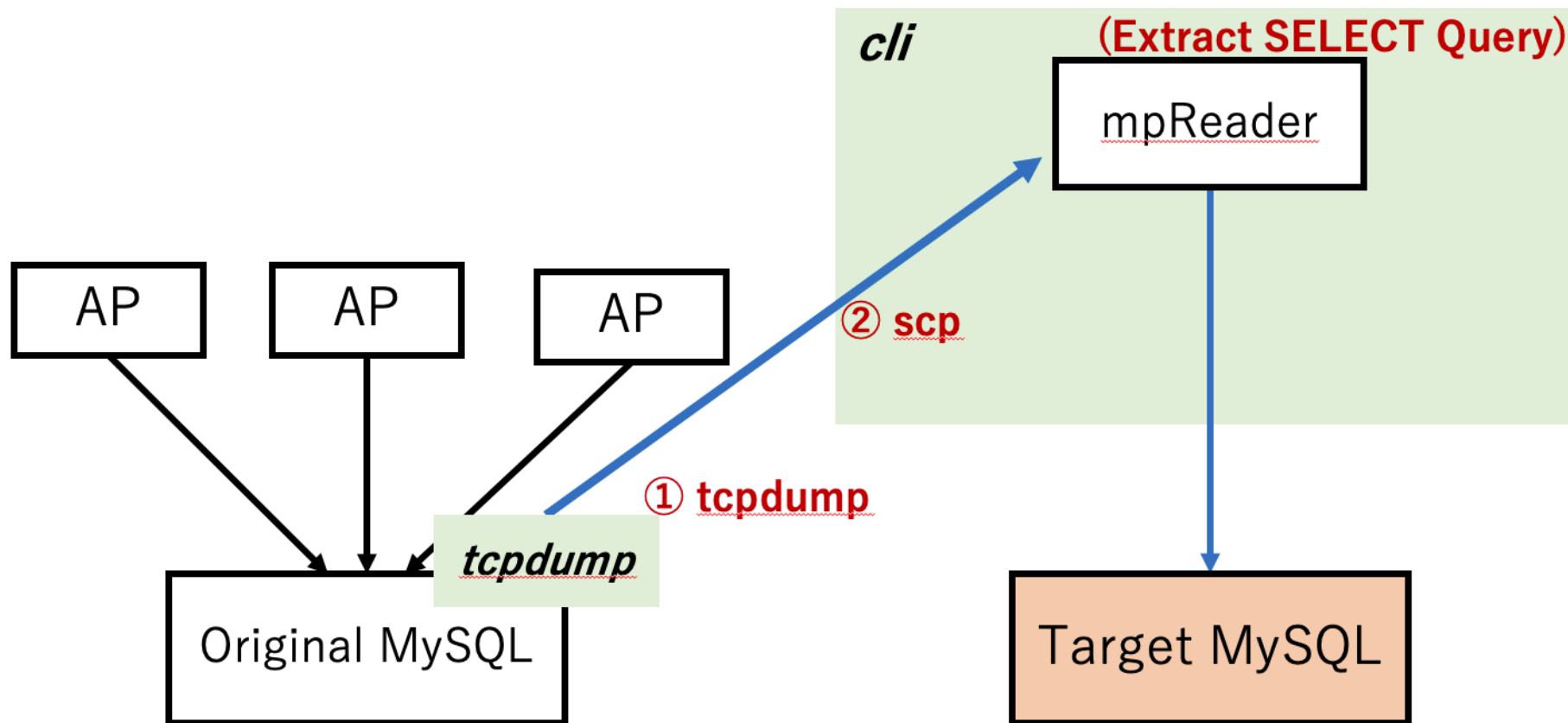
# MQR: MySQL Query Replayer



# MQR: MySQL Query Replayer



# MQR: MySQL Query Replayer



# オペレーションを支えるツール

- myStatusgo
  - GitHub: <https://github.com/kenken0807/myStatusgo>
  - インスタンス情報を1秒間隔で表示するMySQL版topコマンド
  - 以下のような情報を収集しターミナル上に表示
    - SHOW GLOBAL STATUS
    - performance\_schema.threads
    - performance\_schema.statement\_digest
    - performance\_schema.file\_summary\_by\_instance
    - information\_schema.innodb\_trx

# mystatusgo

```
now() : 2019-07-02 21:08:26 [Start : 2019-07-02 21:08:11] [Esc or SpaceBar.Exit]
Mode : F1.Normal F2.OS Metric F3.Threads F4.P_S Info F5.SlaveStatus F6.Handler/InnoDB_Rows F7.InnoDB Lock Info F8.InnoDB Buffer Info F9.Tab
----- QPS -----
Hostname Port Conn Run Abort Select Update Insert Delete Replace Qcache Call QPSAll StmtExc Commit Rolbk Slow Ssum Read Delay IO SQL
-----
Master 3306 5 5 0 6205 886 443 443 0 0 0 7977 0 443 0 0 0 OFF
Slave1 3306 1 1 0 2 886 443 443 0 0 0 1774 0 443 0 0 0 ON 0 Yes Yes

>> Elapsed time : 13.70711ms
```

# mystatusgo

```
now() : 2019-07-02 21:09:58 [Start : 2019-07-02 21:08:11] [Esc or SpaceBar.Exit]
Mode : F1.Normal F2.OS Metric F3.Threads F4.P_S Info F5.SlaveStatus F6.Handler/InnoDB_Rows F7.InnoDB Lock Info F8.InnoDB Buffer Info F9.Table IO Statistic (LEFT[KeyArrowLeft]/RIGHT[KeyArrowRight])
Display per Servers (Select Up[KeyArrowUp] or Down[KeyArrowDown] )
----- performace_schema ----- Statement Digest -----
Hostname Port SQL Avg Max Lost * Schema Ful AvgEm AvgLtcy ExCnt Query
-----
Master 3306 3 269.0us 411.0us 0 * sysbench No 1 120.0us 638 SELECT 'c' FROM 'sbtest3' WHERE 'id' = ?
* sysbench No 1 120.0us 480 SELECT 'c' FROM 'sbtest4' WHERE 'id' = ?
* sysbench No 1 120.0us 460 SELECT 'c' FROM 'sbtest2' WHERE 'id' = ?
* sysbench No 1 121.0us 460 SELECT 'c' FROM 'sbtest1' WHERE 'id' = ?
* sysbench No 1 121.0us 458 SELECT 'c' FROM 'sbtest8' WHERE 'id' = ?
* sysbench No 1 121.0us 450 SELECT 'c' FROM 'sbtest9' WHERE 'id' = ?
* sysbench No 0 1.0ms 442 COMMIT
* sysbench No 1 120.0us 400 SELECT 'c' FROM 'sbtest5' WHERE 'id' = ?
* sysbench No 1 121.0us 400 SELECT 'c' FROM 'sbtest10' WHERE 'id' = ?
* sysbench No 1 121.0us 360 SELECT 'c' FROM 'sbtest6' WHERE 'id' = ?
Slave1 3306 2 236.0us 407.0us 0
>> Elapsed time : 10.250398ms
```



# mystatusgo

```
----- QPS -----
Conn Run Abort Select Update Insert Delete Replace Qcache Call QPSAll StmtExc Commit RoIbk Slow Ssum Read Delay IO SQL usr sys CPU ----- -- Disk -- -- NW ---load-avg--- ----- memory ----- - swap --
12 11 0 10131 1449 724 723 0 0 0 13027 0 725 0 0 0 0 OFF -1

----- InnoDB Rows -----
read inserted updated deleted RdFirst RdKey RdLast RdNxt RdPrev RdRnd RdRndNxt Write Update Delete Prepare Commit Rollback
302017 724 1446 724 0 12604 0 289710 0 46910 50270 49004 1505 724 7238 17360 5

----- InnoDB Buffer Pool Info -----
Total Data Dirty Free DataPage DirtyPage FreePage MiscPag Read RdReq WritReq Writen Create Flush LSN-CK HitRate
3.2G 3.1G 911M 121M 189239 55639 7369 0 1154 202954 34095 453 114 453 -1 99.43/100

----- Thread Info -----
* Cmd State User Host DB Time Query
* Query starting root localhost sysben. 0 COMMIT
* Query starting root localhost sysben. 0 COMMIT
* Query System lock root localhost sysben. 0 SELECT c FROM sbtest3 WHERE id BETWEEN 4993 AND 5092
* Query starting root localhost sysben. 0 COMMIT
* Query init root localhost sysben. 0 DELETE FROM sbtest2 WHERE id=5020

----- performance_schema -----
SQL Avg Max Lost * Schema Ful AvgEm AvgLtcy ExCnt Query
10 2.0ms 7.0ms 0 * sysbenc. No 1 388.0us 2568 SELECT 'c' FROM 'sbtest2' WHERE 'id' = ?
* sysbenc. No 1 449.0us 2401 SELECT 'c' FROM 'sbtest1' WHERE 'id' = ?
* sysbenc. No 1 451.0us 2270 SELECT 'c' FROM 'sbtest3' WHERE 'id' = ?
* sysbenc. No 0 1.0ms 724 COMMIT
* sysbenc. No 1 644.0us 267 DELETE FROM 'sbtest3' WHERE 'id' = ?

----- SLAVE STATUS -----
MasterHost IO SQL Master_Log MasterPos Relay_M_Log Exec_Pos Delay SemiM SemiS ChannelName

(Choose Sort Field by [Tab] key)

----- Table/File IO Statistics -----
* Schema TableName/FileName AllRequest AllLtcy FetchRow FetchLtcy InsRows InsLtcy UpdRows UpdLtcy DelRows DelLtcy ReadReq ReadByte ReadLtcy WriteReq WriteByte WriteLtcy MiscReq MiscLtcy
* sysbenchdb sbtest2 105591 1.0s 103542 296.0ms 231 397.0ms 456 478.0ms 232 83.0ms 1009 16M 222.0ms 113 1.9M 1.0ms 8 940.0us
* sysbenchdb sbtest3 102186 393.0ms 101132 354.0ms 267 11.0ms 519 14.0ms 266 13.0ms 0 0 0.0ns 0 0 0.0ns 2 321.0us
* sysbenchdb sbtest1 99892 397.0ms 98969 364.0ms 225 11.0ms 473 15.0ms 225 6.0ms 0 0 0.0ns 0 0 0.0ns 0 0.0ns
* data ib_logfile1 1219 0.0ns 0 0.0ns 0 0.0ns 0 0.0ns 0 0.0ns 0 0 0.0ns 610 9.1M 29.0ms 609 341.0ms
* logs binary_log 608 0.0ns 0 0.0ns 0 0.0ns 0 0.0ns 0 0.0ns 0 0 0.0ns 608 720k 8.0ms 0 0.0ns

>> Elapsed time : 16.700863ms
```

# 今後の課題

# 今後の課題

- DBAが作業することを前提にシステムを設計してきたが、開発者が低コストで運用できるようなシステムの開発
  - オンラインでのスキーマ変更
  - スケーラブルなアーキテクチャに合うACL管理方法
- 自律したシステムの開発
  - オートヒーリングやオートスケーリング機能の改良
  - HAシステムの改良

# Engineer募集集中!

# まとめ

- 01 LINE Database室紹介
- 02 MySQL platformの変遷
- 03 バージョンアップを支える仕組み
- 04 運用を支えるツール
- 05 今後の課題

**LINE**