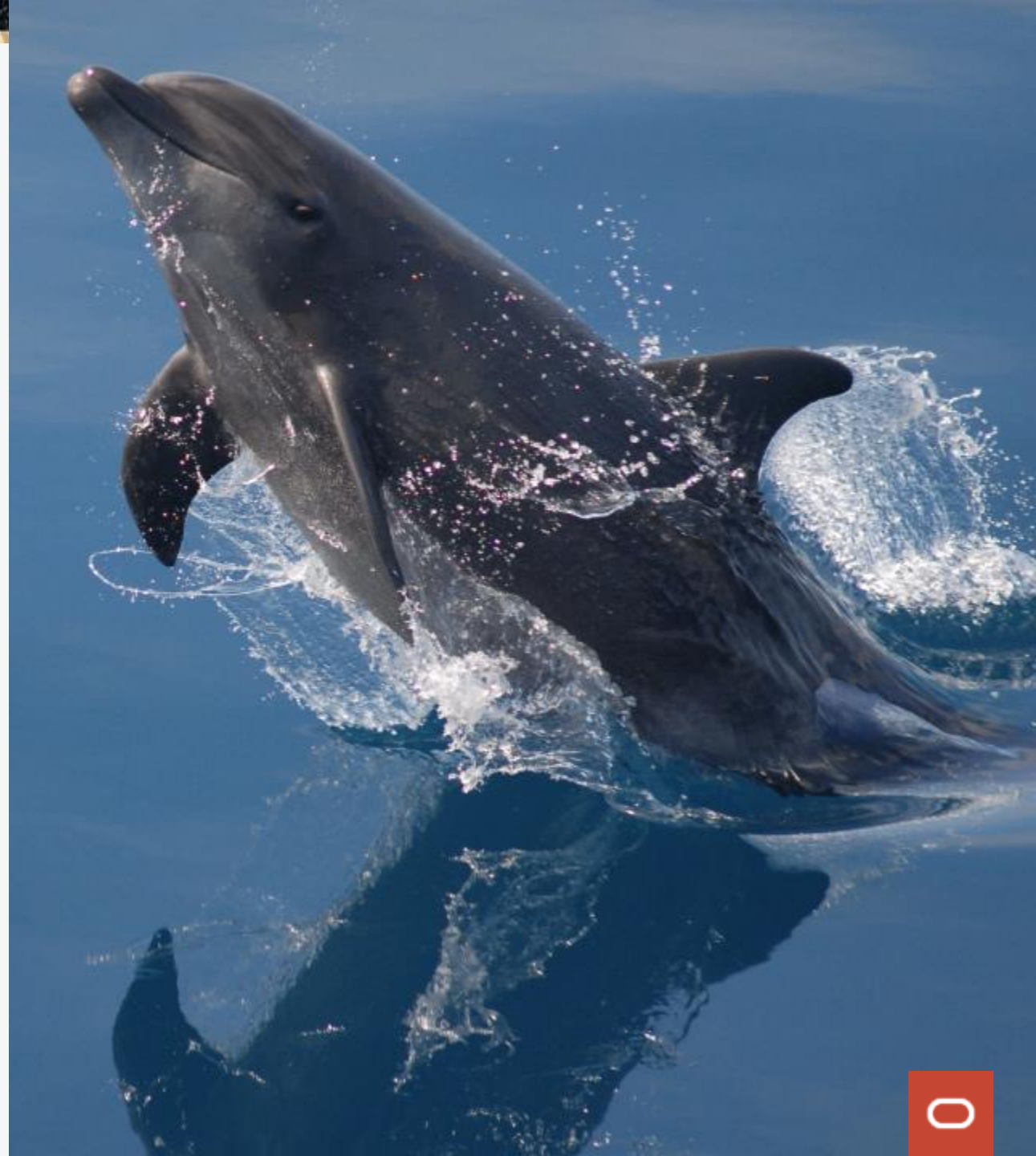
ORACLE

MySQL Enterprise Edition

The Complete Guide

Chandan Kumar
MySQL Solution Architect
MySQL Business Unit



DAY - 02

1. MySQL Advanced Security Features
2. MySQL Replica Set
3. What's new in MySQL 8.0

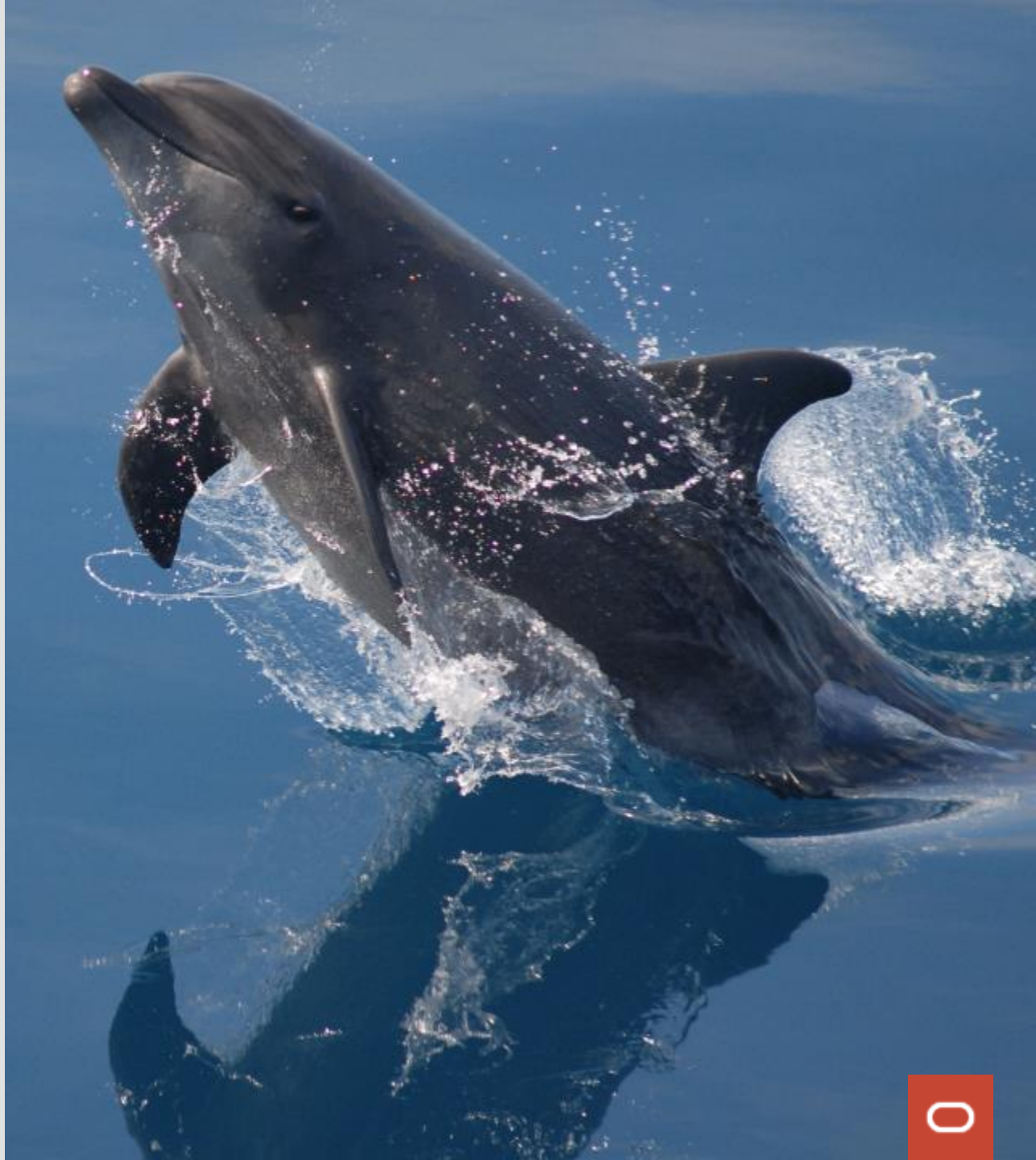
10;30 -5 PM (lunch)

Safe harbor statement

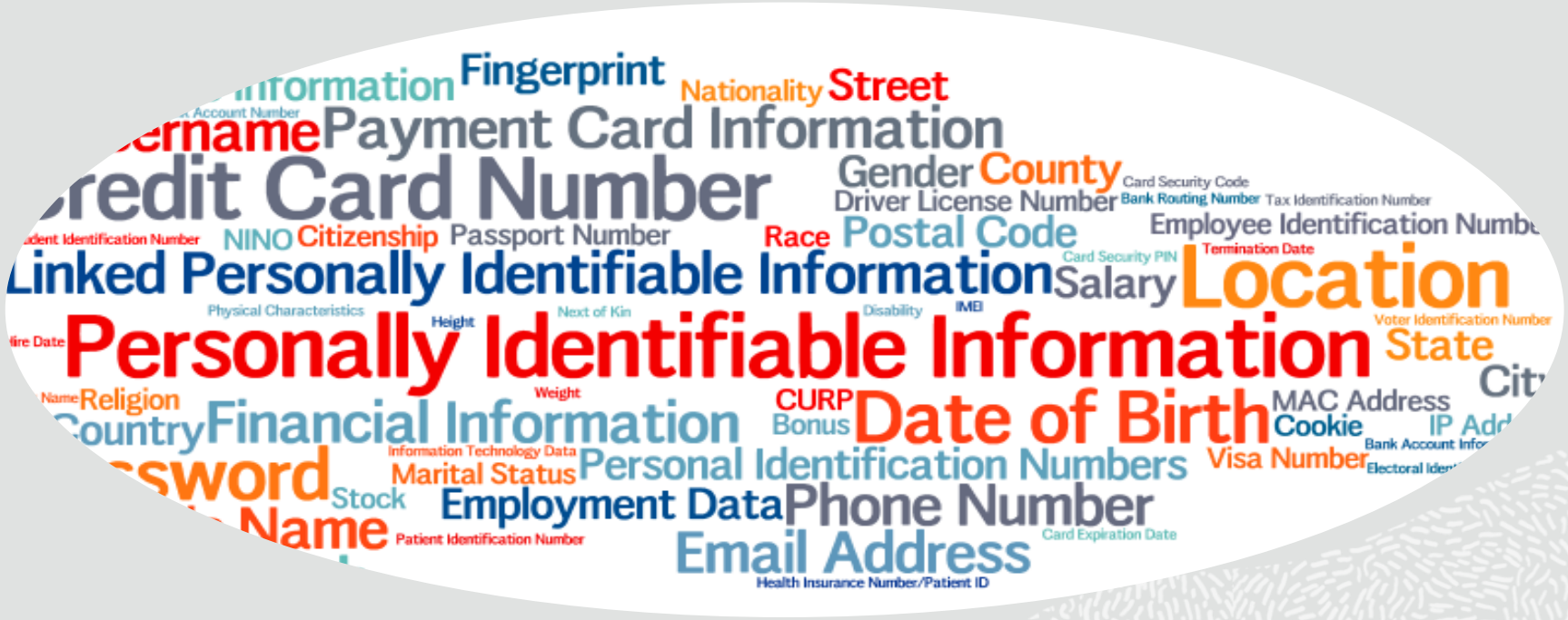
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

MySQL Security Features



Data Your Most Valuable Asset



Data Breaches & Fines

3,800 Breaches, 4 Billion Records Stolen in 1H 2019

A hacker gained access to **100 million** Capital One credit card applications and accounts

British Airways faces record **£183 million** fine for data breach

Marriott discloses massive data breach affecting up to **500 million** guests

Equifax to Pay **$575 million** as Part of Settlement with FTC, CFPB, and States

Regulatory Compliance

Common Requirements

Continuous Monitoring
(Users, Schema, Backups, etc.)

Data Protection
(Encryption, Privileges, Masking, etc.)

Data Retention
(Backups, HA, etc.)

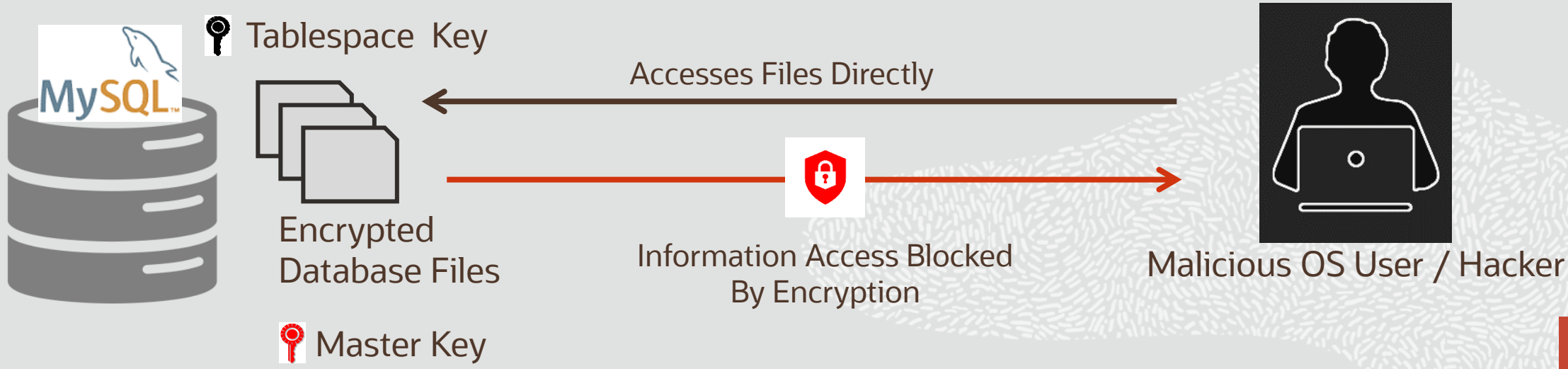
Data Auditing
(User activity, etc.)

MySQL Enterprise Edition SECURITY

- MySQL Enterprise TDE
  - Data-at-Rest Encryption
  - Key Management/Security
- MySQL Enterprise Authentication
  - External Authentication Modules
  - Microsoft AD, Linux PAMs, LDAP
- MySQL Enterprise Encryption
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
  - User Activity Auditing, Regulatory Compliance
- MySQL Data Masking
  - De-identify, Anonymize Sensitive Data
- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- MySQL Enterprise Audit
  - User Activity Auditing, Regulatory Compliance
- MySQL Enterprise Monitor
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords
- MySQL Enterprise Backup
  - Securing Backups, AES 256 encryption

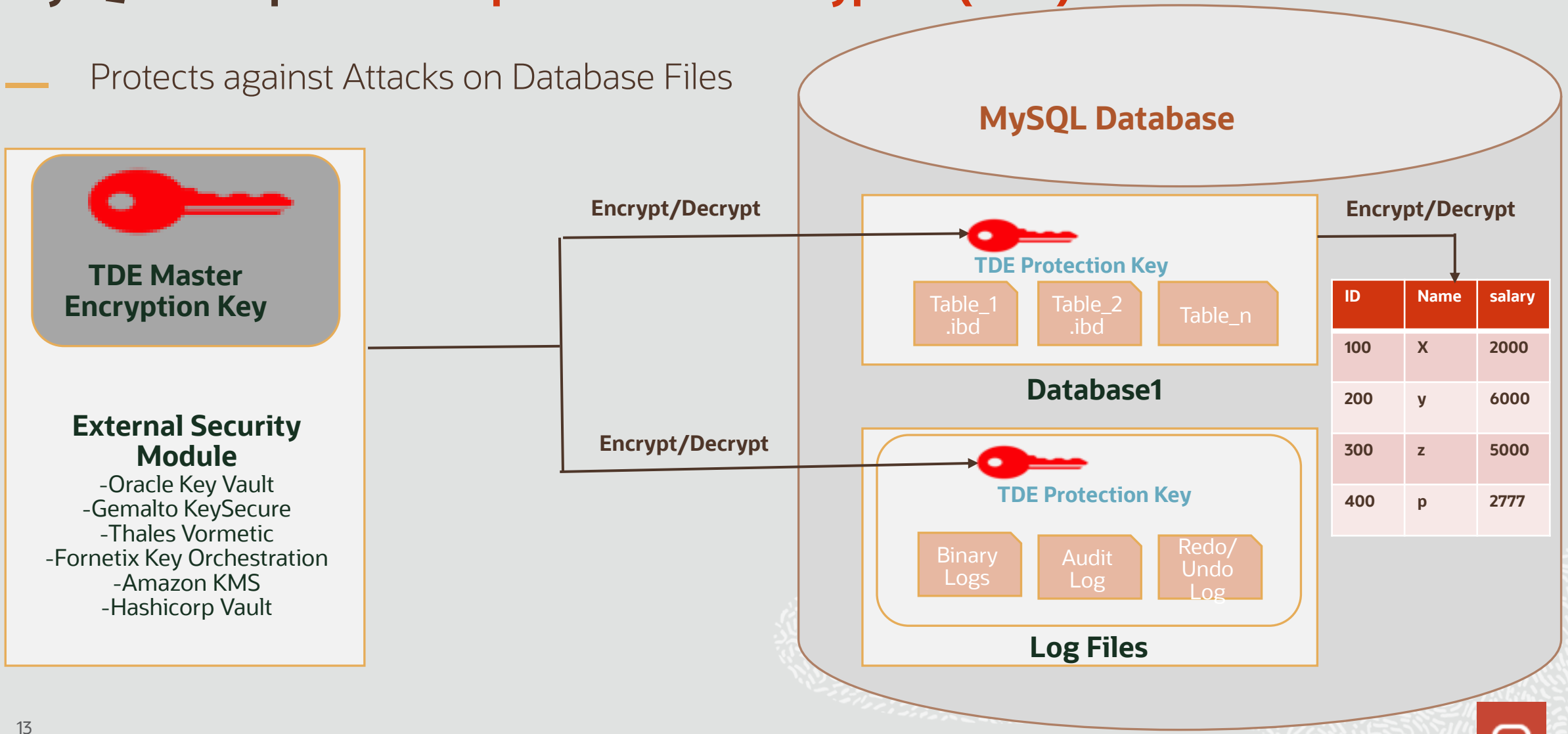
MySQL Enterprise Transparent Data Encryption(TDE)

- Data at Rest Encryption
  - Table spaces, Disks, Storage, OS File system
- Transparent to applications and users
  - No application code, schema or data type changes
- Transparent to DBAs
  - Keys are hidden from DBAs, no configuration changes
- Requires Key Management
  - Protection, rotation, storage, recovery



MySQL Enterprise Transparent Data Encryption(TDE)

Protects against Attacks on Database Files



What about performance after TDE?

- Encryption processing is done at the last stage of the I/O layer.
- A page in the buffer pool is always unencrypted. So once the page is to be read from disk, if it is encrypted, it is unencrypted at I/O layer and then brought into buffer pool.
- All subsequent accesses to that page is fulfilled by buffer pool copy.
- Encryption comes into picture only when page is flushed during which I/O layer encrypts the page before flushing it to the disk.
- That's why there is not a big performance impact of having encryption ON for tablespaces.

MySQL Enterprise Edition SECURITY

- MySQL Enterprise TDE
  - Data-at-Rest Encryption
  - Key Management/Security
- MySQL Enterprise Authentication
  - External Authentication Modules
  - Microsoft AD, Linux PAMs, LDAP
- MySQL Enterprise Encryption (column encryption)
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
  - User Activity Auditing, Regulatory Compliance
- MySQL Data Masking
  - De-identify, Anonymize Sensitive Data
- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- MySQL Enterprise Audit
  - User Activity Auditing, Regulatory Compliance
- MySQL Enterprise Monitor
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords
- MySQL Enterprise Backup
  - Securing Backups, AES 256 encryption

MySQL Enterprise Masking and De-Identification

De-identify, Anonymize Sensitive Data

- Data Masking
  - String Masking, Dictionary Replacement
- Random Data Generators
  - Range based, Payment Card, Email, SSN
- Meet Regulatory Requirements
  - Including GDPR, HIPAA and PCI DSS
- Improve Production, Dev, Test, Env.
  - While Protecting Confidential Data

Employee Table

ID	Last	First	SSN
1111	Sharma	Jay	555-12-5555
1112	Kumar	Ram	444-12-4444

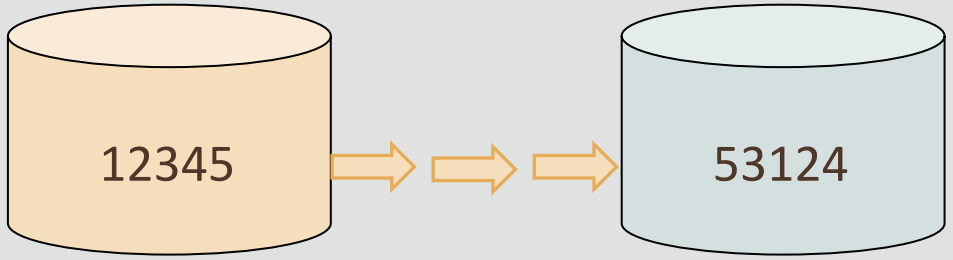
De-identify, Anonymize Sensitive Data

ID	Last	First	SSN
2874	Sharma	Jay	XXX-XX-5555
3281	Kumar	Ram	XXX-XX-4444

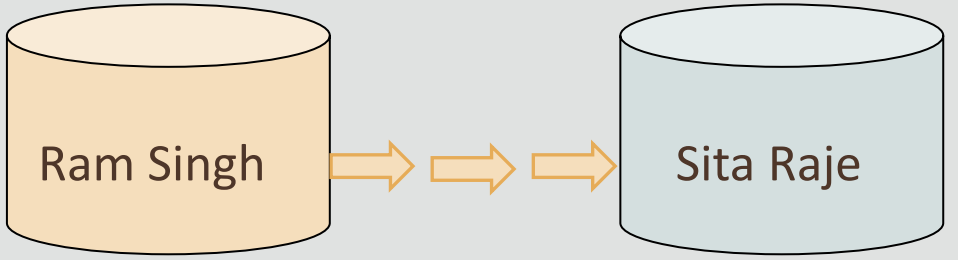
Employee Table Masked View

MySQL Enterprise Masking and De-Identification

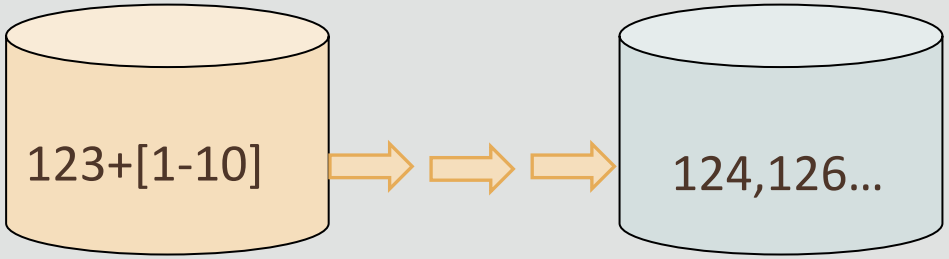
SHUFFLE



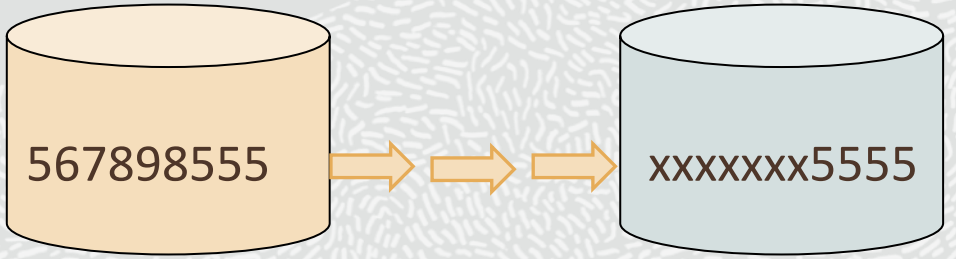
SUBSTITUTION



RANDOM



Masking Out



Data Masking Functions

gen_blacklist
mask_inner
mask_outer
mask_pan
mask_pan_relaxed
mask_ssn
gen_dictionary
gen_rnd_ssn
gen_rnd_us_phone
gen_rnd_email
gen_range
gen_dictionary_load
gen_dictionary_drop

To use Masking functions in app , invoke the functions that are appropriate for the operations you wish to perform.

How to setup Data Masking Functions

In Linux

```
INSTALL PLUGIN data_masking SONAME 'data_masking.so';
CREATE FUNCTION gen_blacklist RETURNS STRING SONAME 'data_masking.so';
CREATE FUNCTION gen_dictionary RETURNS STRING   SONAME 'data_masking.so';
CREATE FUNCTION gen_dictionary_drop RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION gen_dictionary_load RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION gen_range RETURNS INTEGER  SONAME 'data_masking.so';
CREATE FUNCTION gen_rnd_email RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION gen_rnd_pan RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION gen_rnd_ssn RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION gen_rnd_us_phone RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION mask_inner RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION mask_outer RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION mask_pan RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION mask_pan_relaxed RETURNS STRING  SONAME 'data_masking.so';
CREATE FUNCTION mask_ssn RETURNS STRING  SONAME 'data_masking.so';
```

In Windows

```
INSTALL PLUGIN data_masking SONAME 'data_masking.dll';
CREATE FUNCTION gen_blacklist RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_dictionary RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_dictionary_drop RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_dictionary_load RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_range RETURNS INTEGER  SONAME 'data_masking.dll';
CREATE FUNCTION gen_rnd_email RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_rnd_pan RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_rnd_ssn RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION gen_rnd_us_phone RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION mask_inner RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION mask_outer RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION mask_pan RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION mask_pan_relaxed RETURNS STRING  SONAME 'data_masking.dll';
CREATE FUNCTION mask_ssn RETURNS STRING  SONAME 'data_masking.dll';
```

How to Use Data Masking Functions

```
DROP TABLE IF EXISTS Employee;

CREATE TABLE Employee(ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,Empname VARCHAR(20) ,CreditCardNo
CHAR(20) ,EmailID VARCHAR(50) ,SSN CHAR(11));

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Ram
Singh','976654433956345','ram.singh@gmail.com','078-05-1120');

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Krish
Sindhe','76685433956527','krish.sindhe@gmail.com','676-08-9921');

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Sheetal Sharma', '5585495978650',
'sheetal.sharma@gmail.com', '754-97-2143');

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Raj
Kumar','787654433932986','raj.kumar@gmail.com','047-03-7720');

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Deepak Saini','85433956527592',
'deepak.saini@gmail.com','176-81-2111');

INSERT INTO Employee(Empname,CreditCardNo,EmailID,SSN) VALUES('Sumi Shaikh', '2165433959764',
'sheetal.sharma@gmail.com', '475-07-4321');

SELECT * FROM Employee;
```

How to Use Data Masking Functions

```
select Empname ,mask_pan(CreditCardNo) as
CreditCardNo,mask_inner(EmailID,1,4,'*')EmailID,mask_ssn(CONVERT(SSN USING BINARY))SSN  FROM Employee;

###CREATE VIEW on Parent Table, Lets App Call View to see Reports with Masked Data

DROP VIEW IF EXISTS vw_Employee;

CREATE VIEW vw_Employee AS select Empname ,mask_pan(CreditCardNo) as CreditCardNo,
gen_rnd_email()EmailID,mask_ssn(CONVERT(SSN USING BINARY))SSN  FROM Employee;


SELECT * FROM vw_Employee ;

SELECT * FROM vw_Employee WHERE Empname LIKE 'R%';


SELECT * FROM vw_Employee WHERE Empname = 'Sumi Shaikh';
```

How to Use Data Masking Functions

```
SELECT mask_pan(CreditCardNo) CreditCardNo,
       mask_inner(EmailID,1,4,'*')EmailID,
       mask_ssn(CONVERT(SSN USING BINARY))SSN
FROM Employee;
```

```
+------------------+-----------------------------+--------------+
| CreditCardNo     | EmailID                     | SSN          |
+------------------+-----------------------------+--------------+
| XXXXXXXXXXX6345  | r**************.com        | XXX-XX-1120  |
| XXXXXXXXXX6527   | k****************.com      | XXX-XX-9921  |
| XXXXXXXXX8650    | s******************.com    | XXX-XX-2143  |
| XXXXXXXXXXX2986  | r**************.com        | XXX-XX-7720  |
| XXXXXXXXXX7592   | d****************.com      | XXX-XX-2111  |
| XXXXXXXXX9764    | s******************.com    | XXX-XX-4321  |
+------------------+-----------------------------+--------------+
6 rows in set (0.00 sec)
```

MySQL Enterprise Edition SECURITY

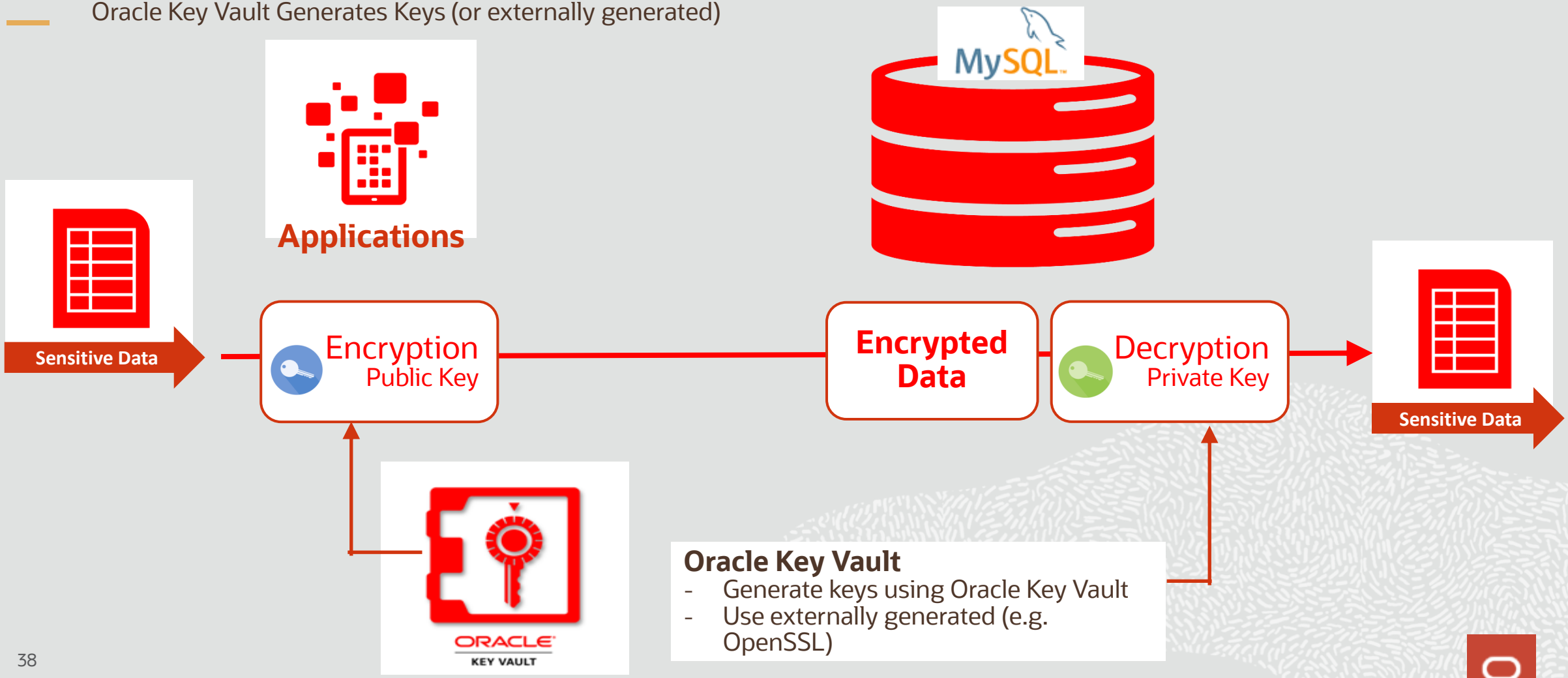
- MySQL Enterprise TDE
  - Data-at-Rest Encryption
  - Key Management/Security
- MySQL Enterprise Authentication
  - External Authentication Modules
  - Microsoft AD, Linux PAMs, LDAP
- MySQL Enterprise Encryption
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
  - User Activity Auditing, Regulatory Compliance
- MySQL Data Masking
  - De-identify, Anonymize Sensitive Data
- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- MySQL Enterprise Audit
  - User Activity Auditing, Regulatory Compliance
- MySQL Enterprise Monitor
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords
- MySQL Enterprise Backup
  - Securing Backups, AES 256 encryption

MySQL Enterprise Encryption

- MySQL encryption libraries
  - Symmetric encryption AES256
  - Public-key / asymmetric cryptography
- Key management
  - Generate public and private keys
  - Key exchange methods: RSA, DSA, DH
- Sign and verify data
  - Cryptographic hashing for digital signing, verification, & validation

MySQL Enterprise Encryption

Oracle Key Vault Generates Keys (or externally generated)



Using MySQL Enterprise Encryptions

```
CREATE FUNCTION asymmetric_decrypt RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION asymmetric_derive RETURNS STRING   SONAME 'openssl_udf.so';
CREATE FUNCTION asymmetric_encrypt RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION asymmetric_sign RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION asymmetric_verify RETURNS INTEGER   SONAME 'openssl_udf.so';
CREATE FUNCTION create_asymmetric_priv_key RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION create_asymmetric_pub_key RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION create_dh_parameters RETURNS STRING  SONAME 'openssl_udf.so';
CREATE FUNCTION create_digest RETURNS STRING  SONAME 'openssl_udf.so';
```

Function	Supported Algorithms
ASYMMETRIC_DECRYPT()	RSA
ASYMMETRIC_DERIVE()	DH
ASYMMETRIC_ENCRYPT()	RSA
ASYMMETRIC_SIGN()	RSA, DSA
ASYMMETRIC_VERIFY()	RSA, DSA
CREATE_ASYMMETRIC_PRIV_KEY()	RSA, DSA, DH
CREATE_ASYMMETRIC_PUB_KEY()	RSA, DSA, DH
CREATE_DH_PARAMETERS()	DH

Using MySQL Enterprise Encryption Functions

Create Public/Private Key

```
-- Encryption algorithm; can be 'DSA' or 'DH' instead
SET @algo = 'RSA';
-- Key length in bits; make larger for stronger keys
SET @key_len = 1024;
-- Create private key
SET @priv = CREATE_ASYMMETRIC_PRIV_KEY(@algo, @key_len);
-- Derive corresponding public key from private key, using same algorithm
SET @pub = CREATE_ASYMMETRIC_PUB_KEY(@algo, @priv);
```

Use Private Key/Public Key to Encrypt Data

```
SET @ciphertext = ASYMMETRIC_ENCRYPT(@algo, 'Secret Data', @priv);
```

```
SET @ciphertext = ASYMMETRIC_ENCRYPT(@algo, 'Secret Data', @pub);
```

Use Public Key/Private Key to Decrypt Data

```
SET @cleartext = ASYMMETRIC_DECRYPT(@algo, @ciphertext, @pub);
```

```
SET @cleartext = ASYMMETRIC_DECRYPT(@algo, @ciphertext, @priv);
```

Using MySQL Enterprise Encryption

```
mysql> select @cleartext, @ciphertext;
+-------------+-----------------------------------------------------------------------------------------------
------------------------------------+
| @cleartext  | @ciphertext                                                                                  |
+-------------+-----------------------------------------------------------------------------------------------
------------------------------------+
| Secret Data | m¨?`Õe*?zZ]t7–¬¾Oæˆà–ËÐK ó-¾ÄŒEcÇ        %·9u ¨·ú˜j@@úvÿ“®Nuÿö>
.à=h“]Áy“:§çe¼¸Š3|&3¼FŒº!F&*peÚkÄòþVíÈ?>gë~w£0˜œ¡ãûšž×+Íw                                              |
+-------------+-----------------------------------------------------------------------------------------------
------------------------------------+
1 row in set (0.00 sec)
```

MySQL Enterprise Audit

- Out-of-the-box logging of connections, logins, and query
- User defined policies for filtering, and log rotation
- Dynamically enabled, disabled: no server restart
- Send data to a remote server / audit data vault
  - Oracle Audit Vault,Splunk,etc
  - Custom Settings
    - XML and New! JSON audit stream formatting options
    - New! Compression-Reduce audit storage up to 10x.
    - New! Encryption-Audit files can now be encrypted using AES-256

MySQL Enterprise Audit

Policy based-audit function

Admin

```
mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';

mysql> SHOW VARIABLES LIKE 'audit_log%';
+--------------------------+--------------+
| Variable_name            | Value        |
+--------------------------+--------------+
| audit_log_buffer_size    | 1048576      |
| audit_log_file           | audit.log    |
| audit_log_flush          | OFF          |
| audit_log_policy         | ALL          |
| audit_log_rotate_on_size | 1044480      |
| audit_log_strategy       | SYNCHRONOUS  |
+--------------------------+--------------+
```

1. DBA enables Audit plugin

Joe （User）

```
shell> mysql -h joeshost -u joe -p
Enter password: ********

mysql> SELECT * FROM joes_table;
+---------------+--------------+
| FIRST_NAME    | LAST_NAME    |
+---------------+--------------+
| Joe           | User         |
+---------------+--------------+
```

2. User Joe connects and runs a query

3. Joe's connection & query logged

```
<?xml version="1.0" encoding="UTF-8"?>
<AUDIT>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:12"
    NAME="Audit"
    SERVER_ID="1"
    VERSION="1"
    STARTUP_OPTIONS="--port=3306"
    OS_VERSION="i686-Linux"
    MYSQL_VERSION="5.5.28-debug-log"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:41"
    NAME="Connect"
    CONNECTION_ID="1"                    WHO
    STATUS="0"
    USER="joe"
    PRIV_USER="root"
    OS_LOGIN=""                          WHERE
    PROXY_USER=""
    HOST="SERVER1"
    IP="127.0.0.1"
    DB="joes_db"/>                       WHEN
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:53:45"
    NAME="Query"
    CONNECTION_ID="1"                    WHAT
    STATUS="0"
    SQLTEXT="SELECT * FROM joes_table;"/>
</AUDIT>
```

MySQL Enterprise Edition SECURITY

- MySQL Enterprise TDE
  - Data-at-Rest Encryption
  - Key Management/Security
- MySQL Enterprise Authentication
  - External Authentication Modules
  - Microsoft AD, Linux PAMs, LDAP
- MySQL Enterprise Encryption
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
  - User Activity Auditing, Regulatory Compliance
- MySQL Data Masking
  - De-identify, Anonymize Sensitive Data
- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- MySQL Enterprise Audit
  - User Activity Auditing, Regulatory Compliance
- MySQL Enterprise Monitor
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords
- MySQL Enterprise Backup
  - Securing Backups, AES 256 encryption

MySQL Enterprise Firewall

- Real Time Protection
  - Queries analyzed and matched against White List
- Blocks SQL Injection Attacks
  - Positive Security Model
- Block Suspicious Traffic
  - Out of Policy Transactions detected & blocked

Transparent

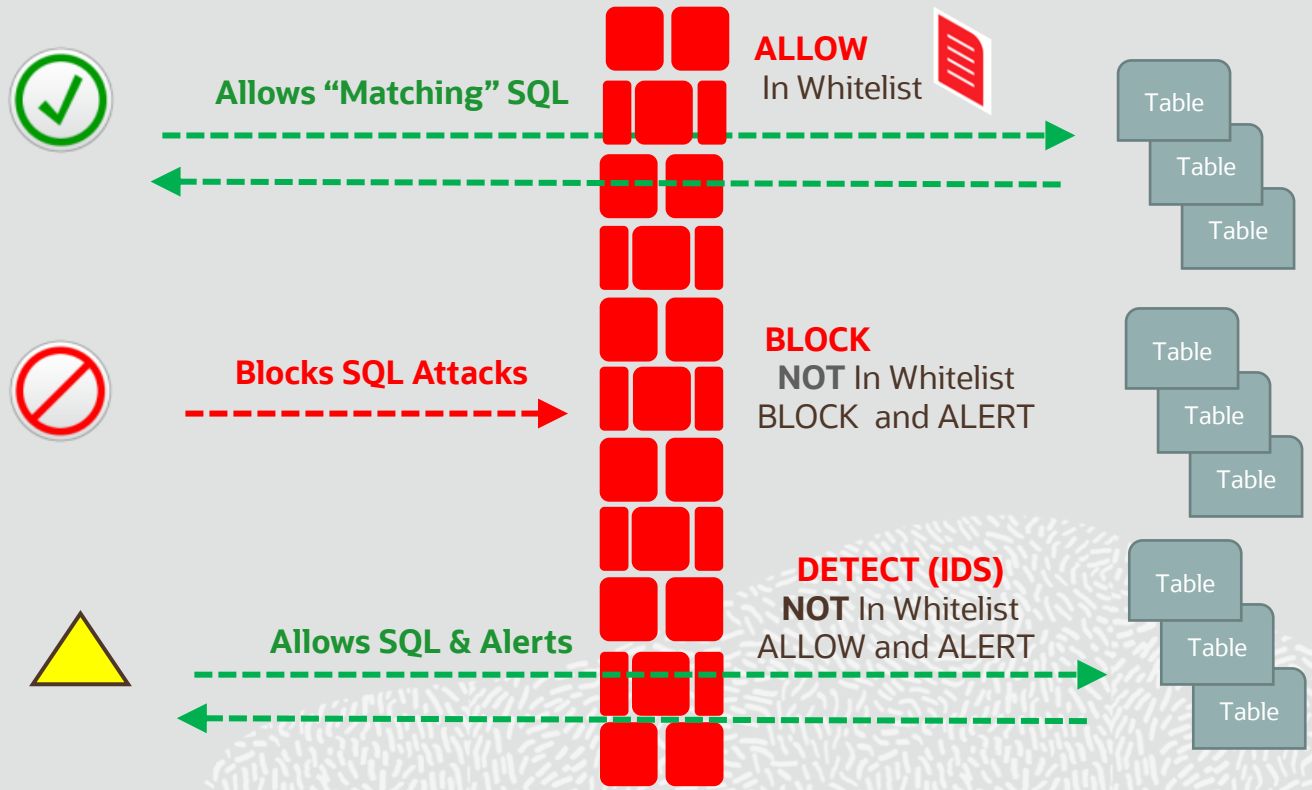
No changes to application required

Learns White List

Automated creation of approved list of SQL command patterns on a per user basis

MySQL Enterprise Firewall : Operating Modes

1. ALLOW – Execute SQL
   - SQL Matches Whitelist
2. BLOCK – Block the request
   - Not in Whitelist
3. DETECT – Execute SQL & Alert
   - Not in Whitelist



MySQL Enterprise Firewall Details

To Install firewall

```
mysql> source /usr/local/mysql/share/mysql/linux_install_firewall.sql
```

-Firewall operation is turned on at a per user level

-Per User States are

```
mysql> SHOW GLOBAL VARIABLES LIKE 'mysql_firewall_mode';
+---------------------+-------+
| Variable_name       | Value |
+---------------------+-------+
| mysql_firewall_mode | ON    |
+---------------------+-------+
1 row in set (0.00 sec)
```

RECORDING

```
call mysql.set_firewall_mode ('fwuser@localhost', 'RECORDING');
```

PROTECTING

```
call mysql.set_firewall_mode ('fwuser@localhost', 'PROTECTING');
```

DETECTING

```
call mysql.set_firewall_mode ('fwuser@localhost', 'DETECTING');
```

OFF

```
call mysql.set_firewall_mode ('fwuser@localhost', 'OFF');
```

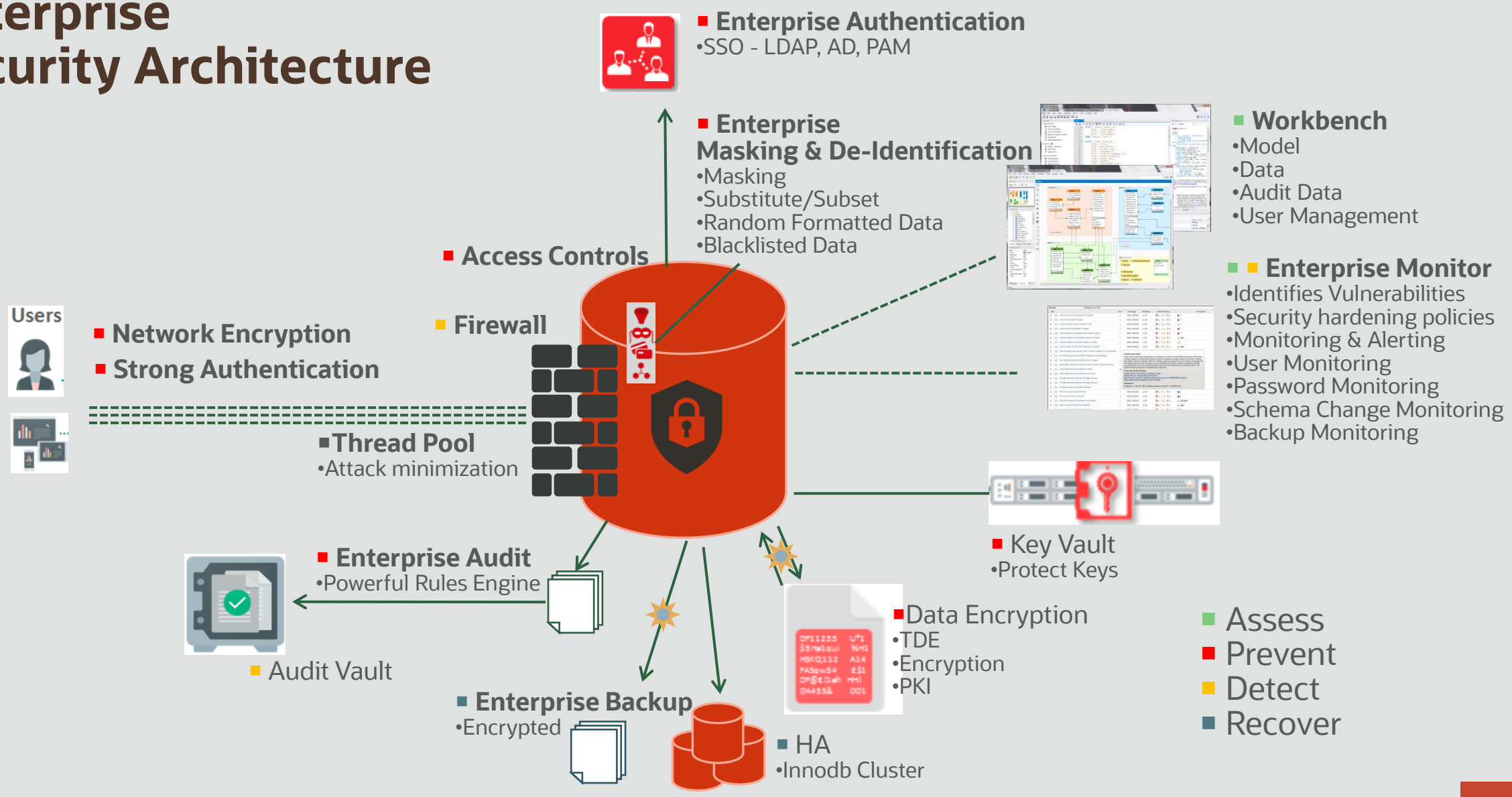
MySQL Enterprise Firewall Example

The client application gets an ERROR

```
mysql> SELECT first_name, last_name FROM customer WHERE customer_id = 1;
ERROR 1045 (28000): Statement was blocked by Firewall
mysql> SHOW DATABASES;
ERROR 1045 (28000): Statement was blocked by Firewall
mysql> TRUNCATE TABLE mysql.user;
ERROR 1045 (28000): Statement was blocked by Firewall
```

Reported to the Error Log

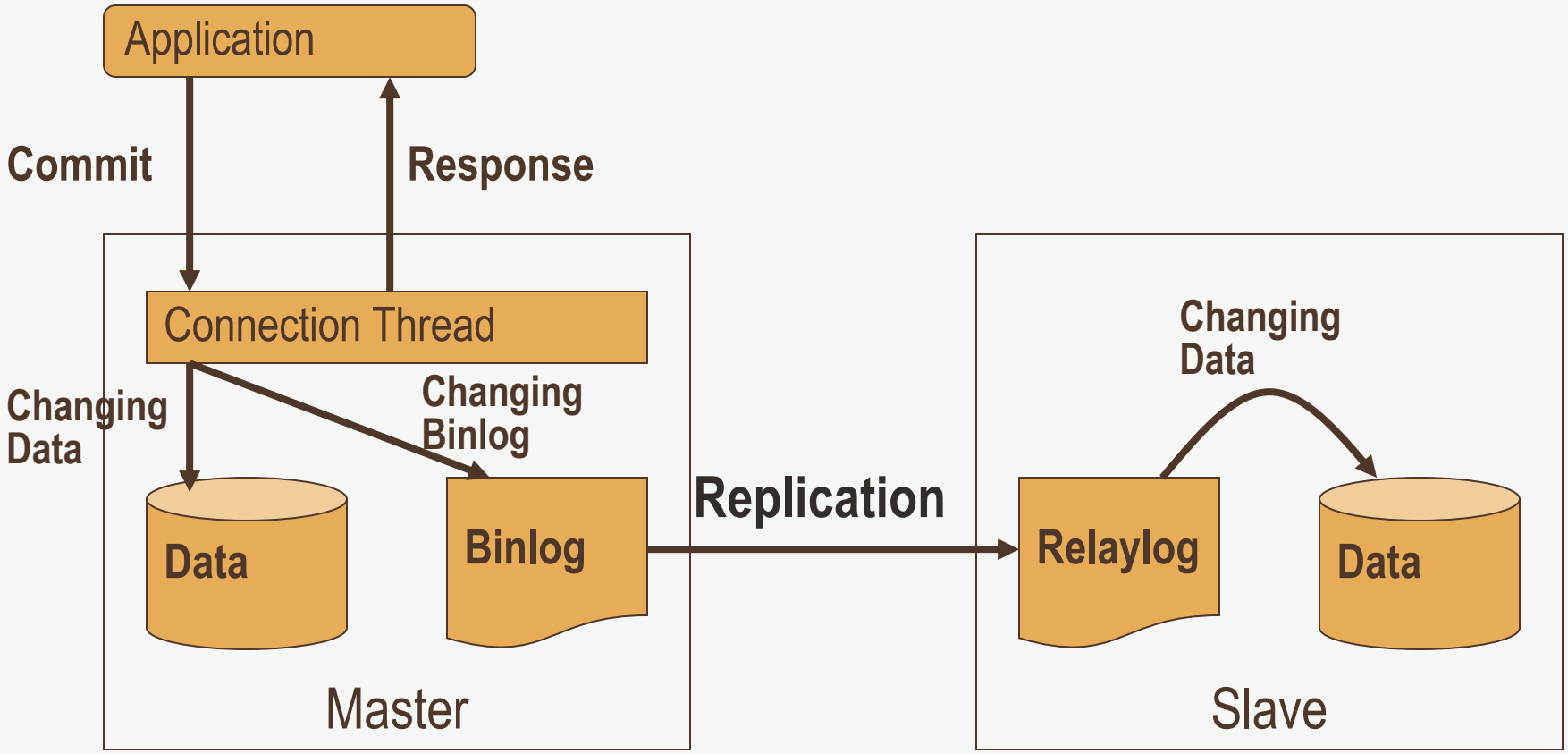
Enterprise Security Architecture



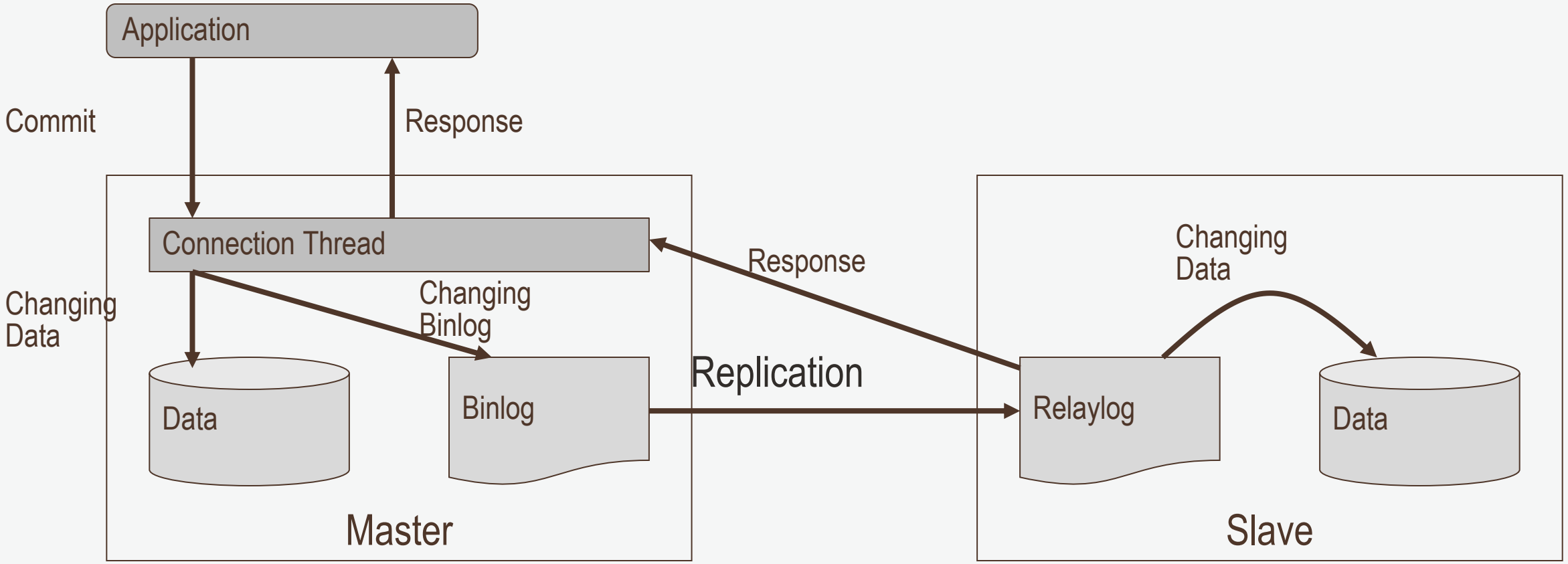
MySQL InnoDB Cluster

High Availability - Out of the Box

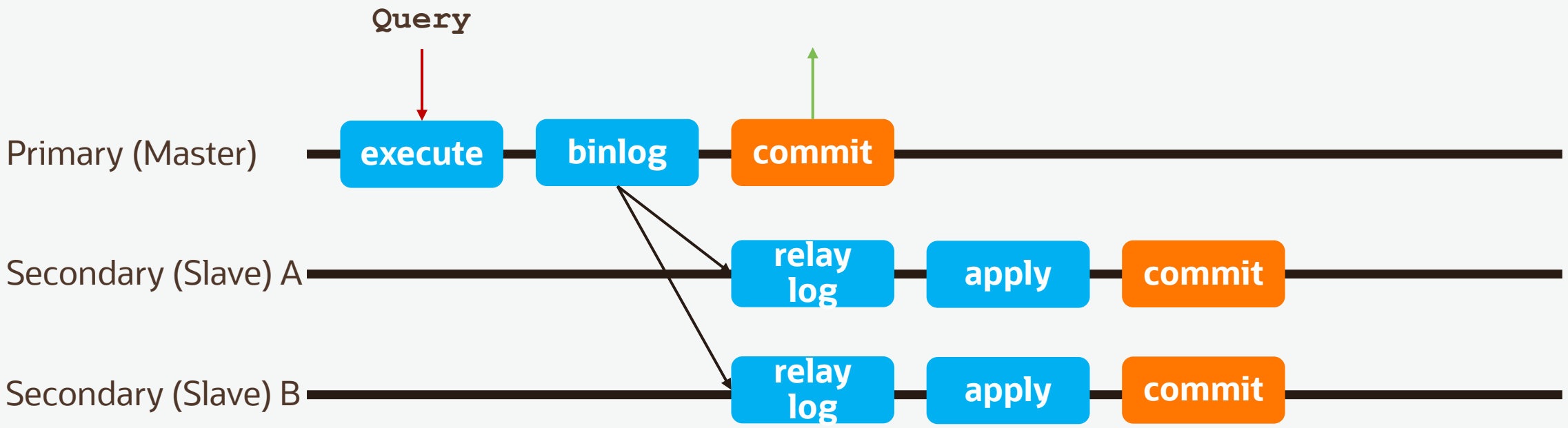
In the beggining there was Asynchronous Replication



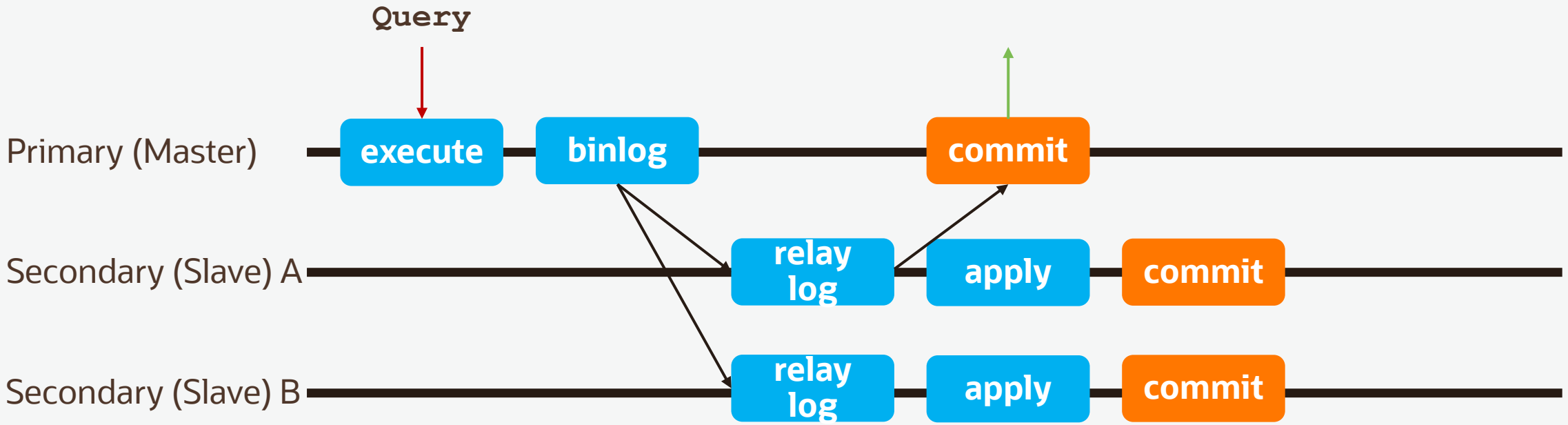
Semi-synchronous Replication



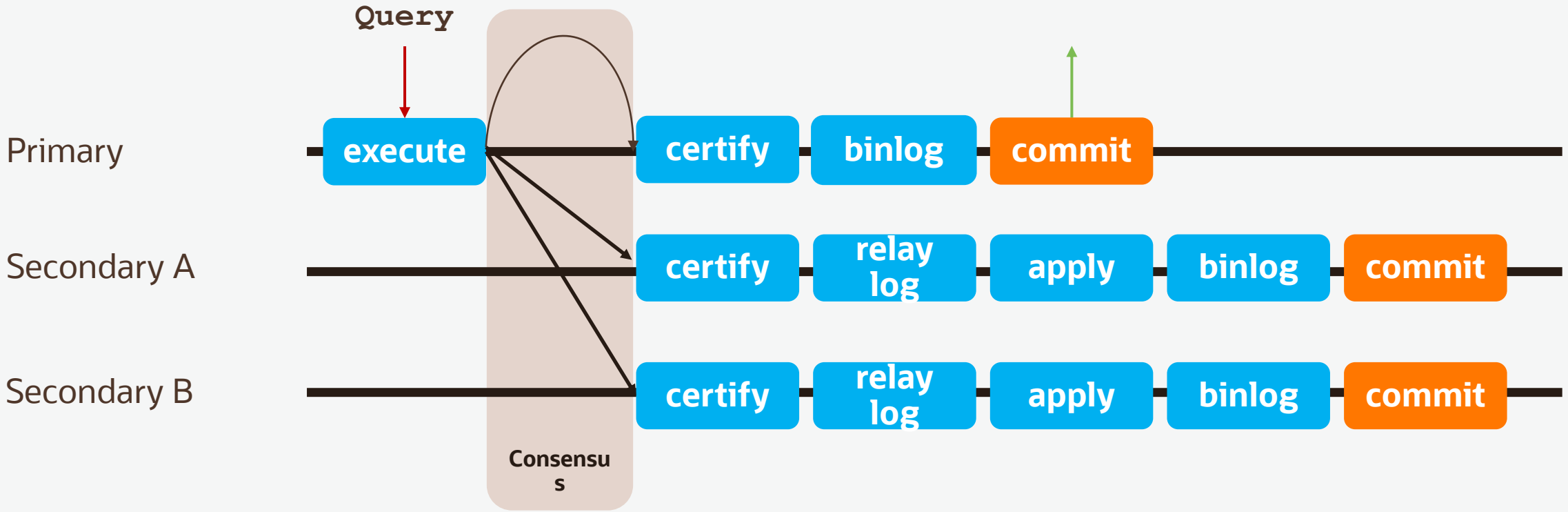
Replication Technologies: Native Replication



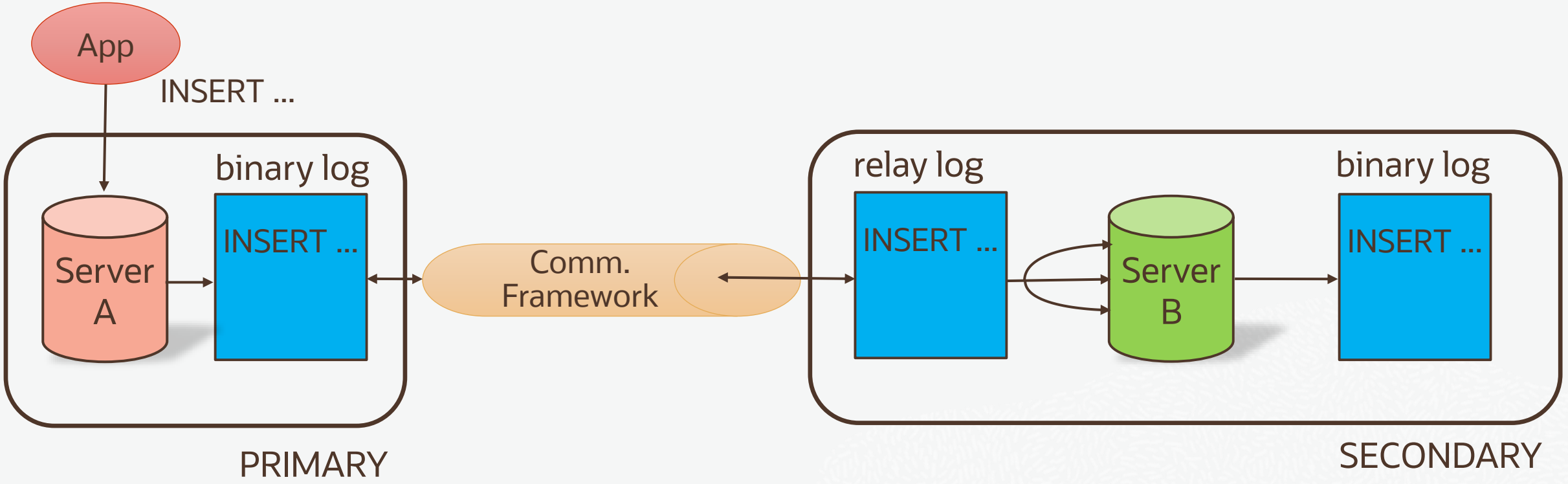
Replication Technologies: Semi-Synchronous Replication



Replication Technologies: Group Replication



MySQL Database Replication: Overview



MySQL Database Replication: Binary Log

Logical replication log recording master changes (binary log).

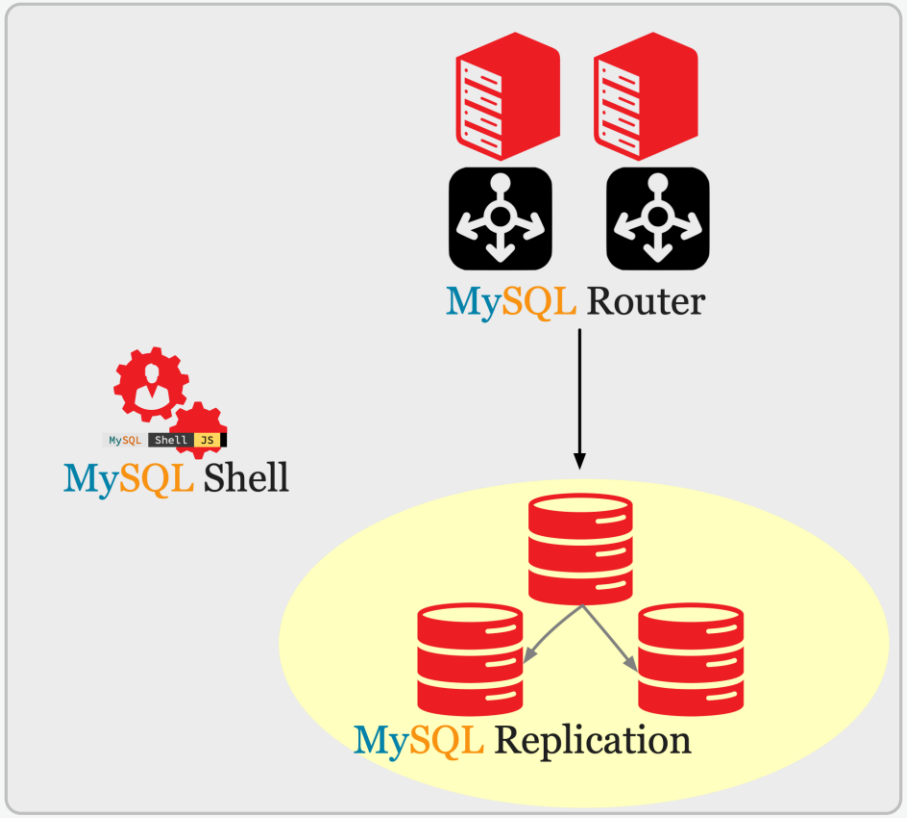
Row or statement based format (may be intermixed).

Each transaction is split into groups of events.

Control events: Rotate, Format Description, Gtid, and more.

Layout of the Binary Log.

MySQL InnoDB ReplicaSets (8.0.19)



- **Asynchronous Replication Architecture**
  - (manual) Switchover & Failover
  - (asynchronous) Read Scaleout
  - Simple Replication architecture
- **MySQL Shell** Configuring, Adding, Removing members
- **MySQL Router** to route application traffic
- **InnoDB CLONE** to automatically provision members, fully integrated in InnoDB

InnoDB ReplicaSets Features

Before

- Restore a backup to provision a member
- Configure Replication Users and Configure Replication
- Manually configuring, adding removing servers in MySQL Router or alternatives
- Manually or relying on external tools to make topology changes
- Use additional monitoring tool log in on all machines to check topology

Now

- ✓ Automatically provisioning new members: InnoDB Clone
- ✓ MySQL Shell Automatically configures users & Replication
- ✓ Integrated MySQL Router load balancing
- ✓ Automatic Router Bootstrapping – no config
- ✓ Router is stateless, adapts to topology changes
- ✓ Easy to use manual switchover/failover

InnoDB ReplicaSets Requirements and Limitations

- **MySQL 8 (set persist)**
- **GTID**
- **Only manual failover**
  - This is good! Data consistency!
- **No multi-primary as such topology cannot guarantee data consistency**
  - No data reconciliation
  - No conflict handling
- **All secondary members replicate from primary**
  - Single tiered replication support

MySQL ReplicaSet – “Hands-On“

Configure MySQL instances for ReplicaSet usage

```
mysql-js> dba.configureReplicaSetInstance("root@servers:3306")
```

Create ReplicaSet

```
mysql-js> \c root@server1:3306
mysql-js> rs = dba.createReplicaSet(“myreplicaset")
```

Check Status

```
mysql-js> rs.status()
```

Add Instances to ReplicaSet

```
mysql-js> rs.addInstance("root@server2:3306")
mysql-js> rs.addInstance("root@server3:3306")
```

Switch-Over

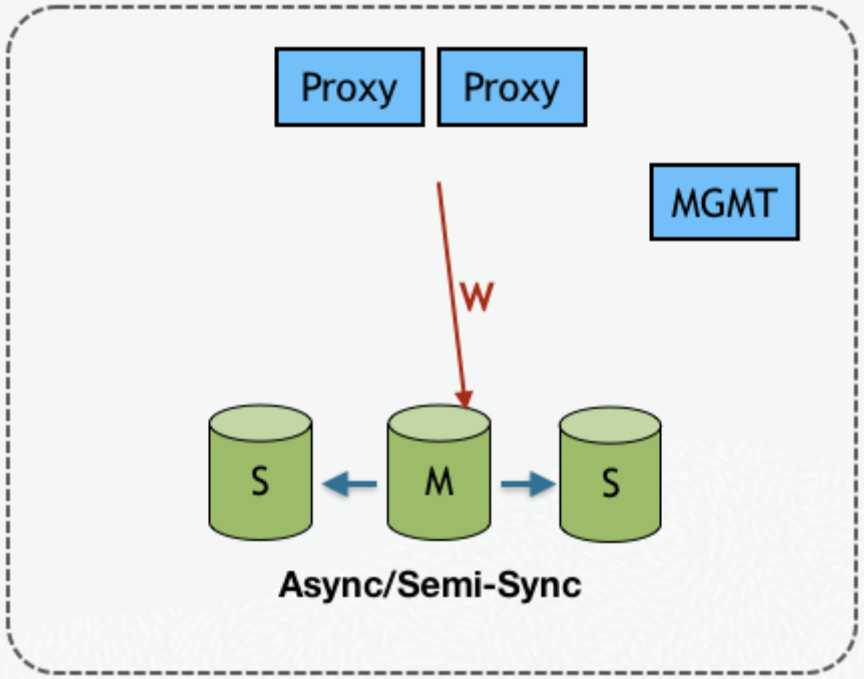
```
mysql-js> rs.setPrimaryInstance("server3:3306")
```

ReplicaSet Manual Failover only!

Having external monitoring processes decide failover can cause a lot of false positives.

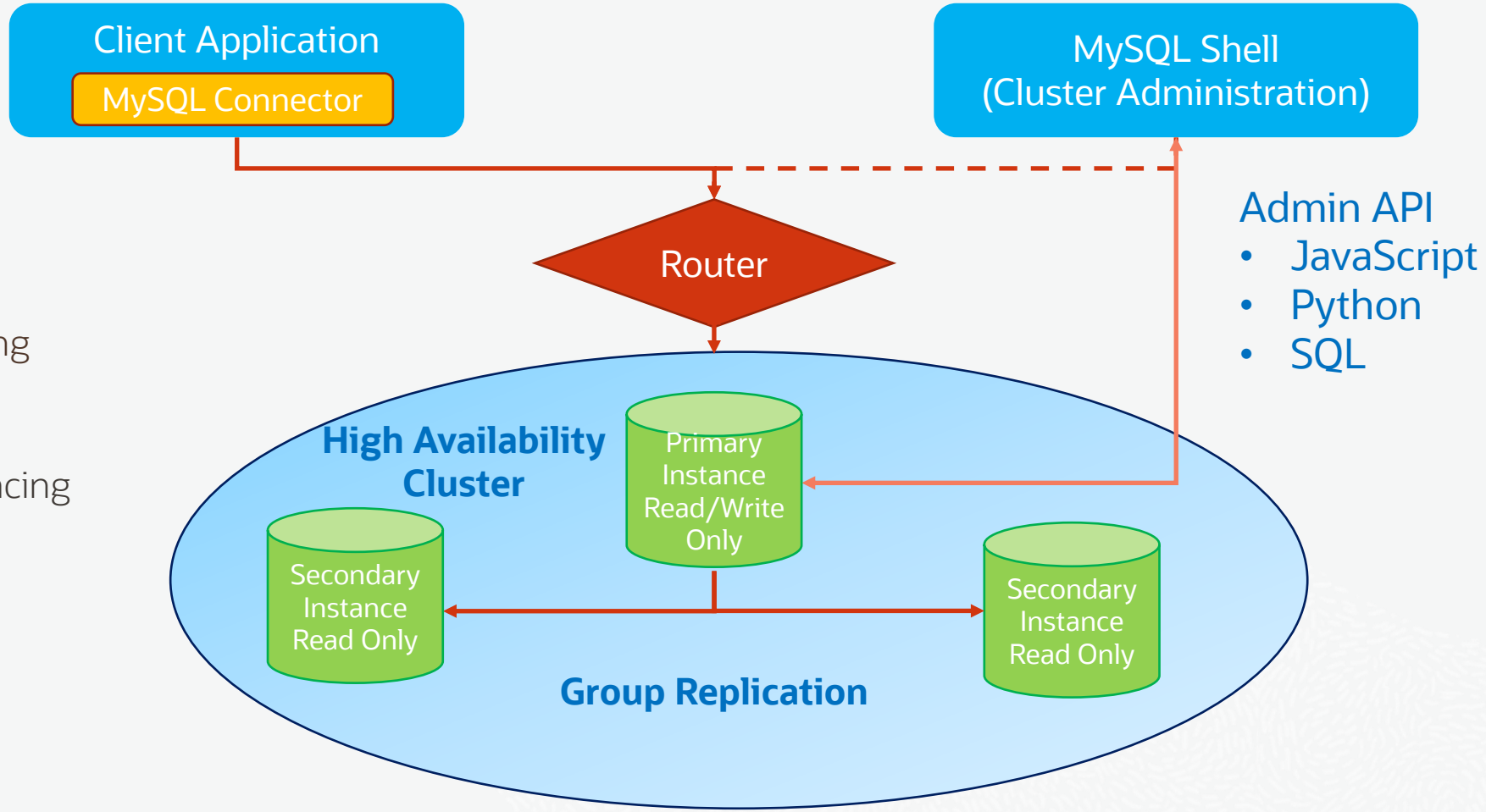
- If the external tool has issues: even bigger issues
- Split brain issues
- Majority of production deployments are configured with Manual Failover, which increases Uptime!

When automatic failover or semi-sync is needed:
Use MySQL InnoDB Cluster!



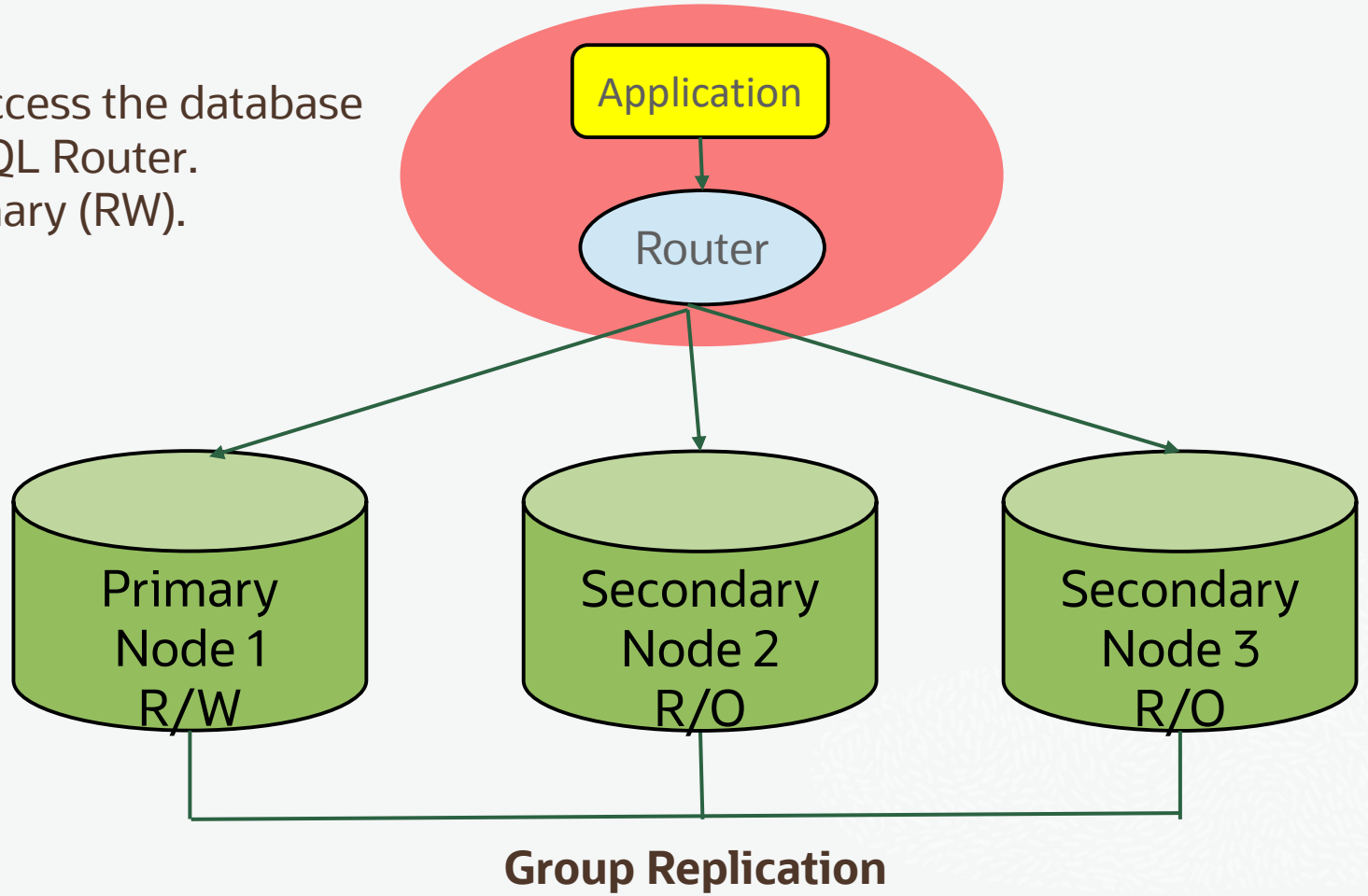
MySQL InnoDB Cluster - Architecture >>Min 03 , Max >>09

- MySQL Group Replication
  - High Availability
  - Elastic, Fault Tolerant, Self Healing
- MySQL Router
  - Connection Routing, Load Balancing
- MySQL Shell
  - Easy Setup & Administration



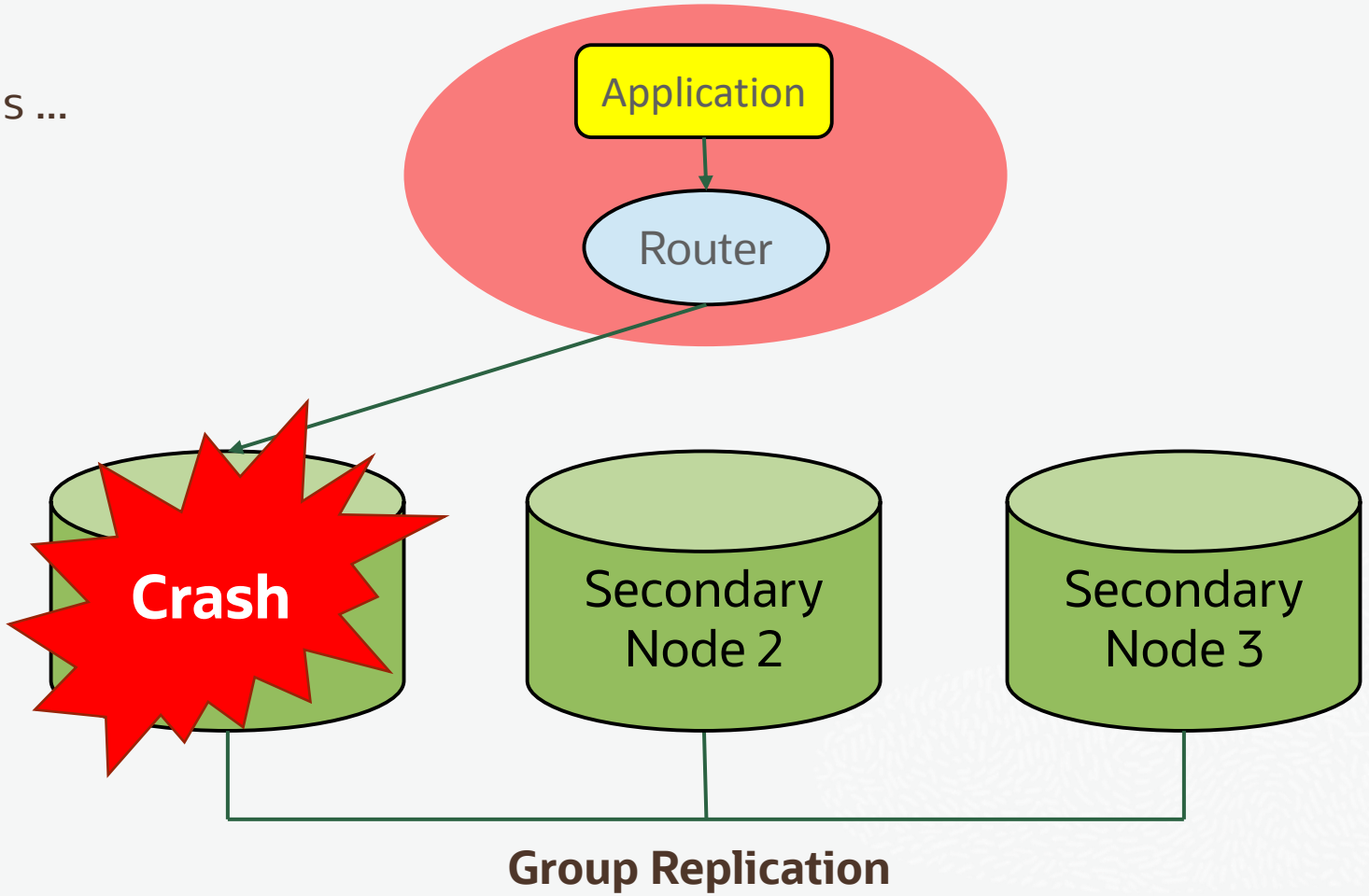
InnoDB Cluster: Application access

Application access the database through MySQL Router.
Node 1 is Primary (RW).



InnoDB Cluster: Failure detection

Crash happens ...

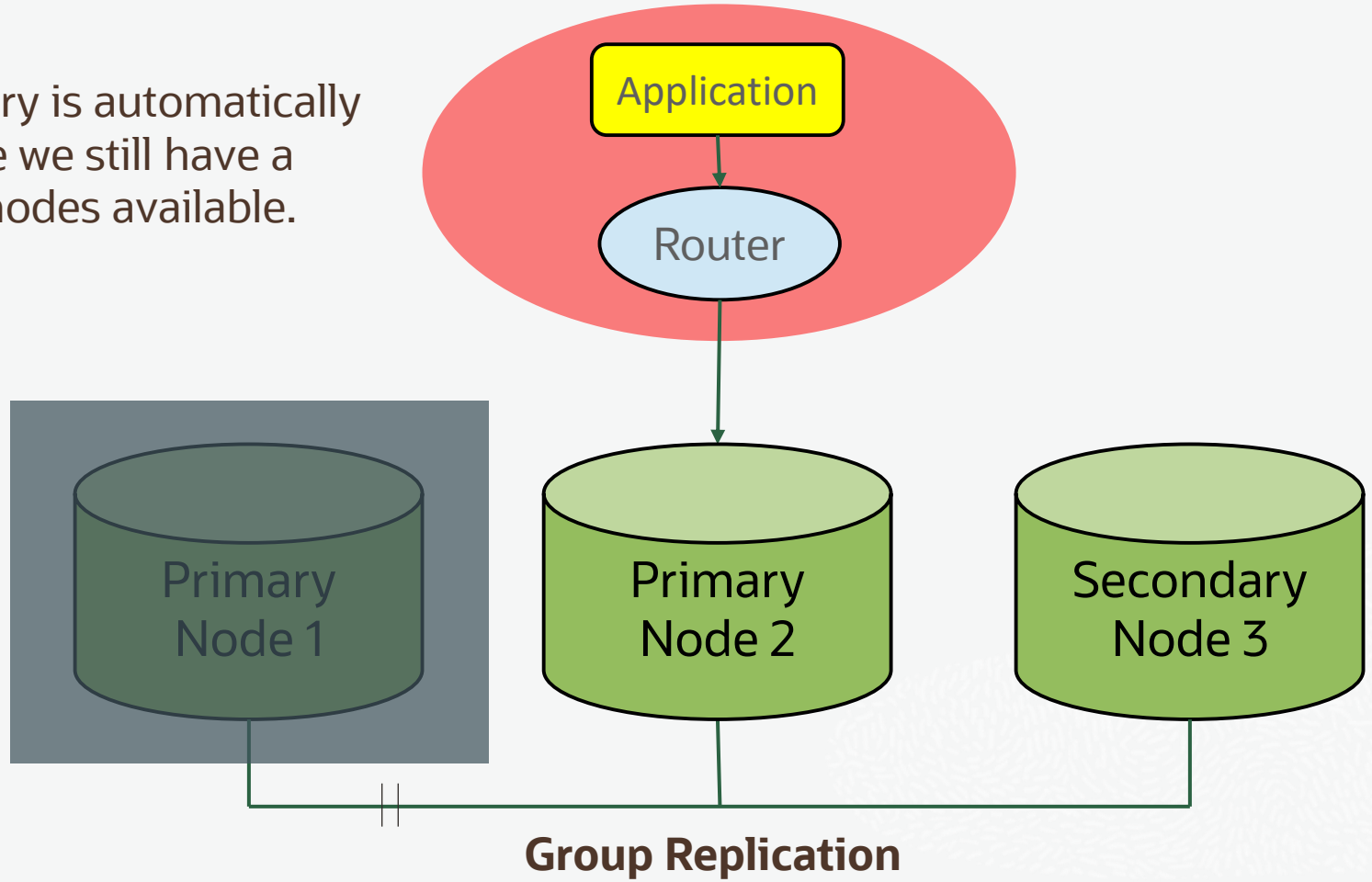


InnoDB Cluster: Failover PRIMARY role to Node 2

Min no of nodes ~ 03

Max no of nodes ~ 09

A new primary is automatically elected since we still have a majority of nodes available.



MySQL Group Replication: Deploy Modes

- Multi-Primary Mode
  - Every server in the group is allowed to execute transactions at any time, even transactions that change state (RW transactions)
  - Risk of more aborted transactions
  - Send DDL operations to only one node in the cluster
- Single-Primary Mode (Default)
  - The group has a single primary server that is set to read-write mode. All the other members in the group are set to read-only mode.
  - No need to modify application.

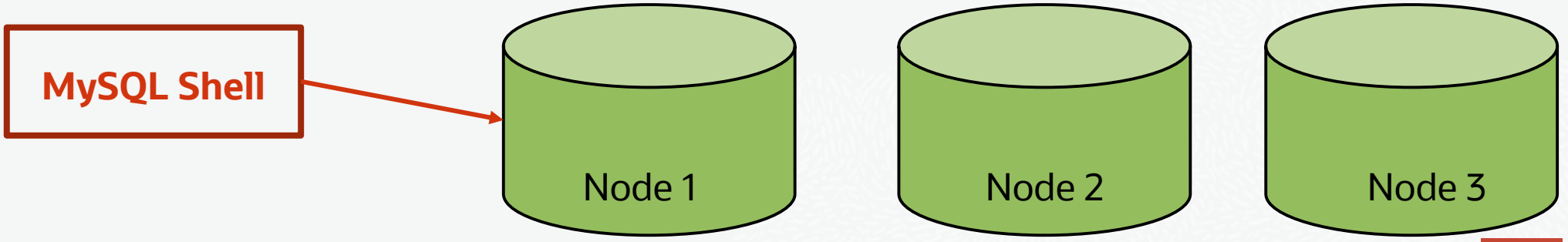
MySQL Shell : Setup InnoDB Cluster easily

- Connect to one server and create cluster with this member

```
MySQL  JS > \connect admin@10.0.0.75
MySQL  10.0.0.75:33060+ ssl  JS > Cluster = dba.createCluster("testcluster")
```

- Add the other two members to the cluster

```
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.76:3306")
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.77:3306")
```



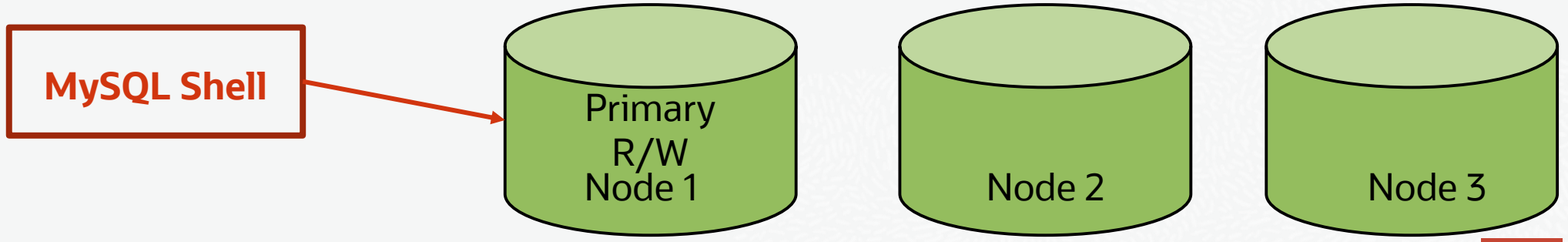
MySQL Shell : Setup InnoDB Cluster easily

- Connect to one server and create cluster with this member

```
MySQL  JS > \connect admin@10.0.0.75
MySQL  10.0.0.75:33060+ ssl  JS > Cluster = dba.createCluster("testcluster")
```

- Add the other two members to the cluster

```
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.76:3306")
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.77:3306")
```



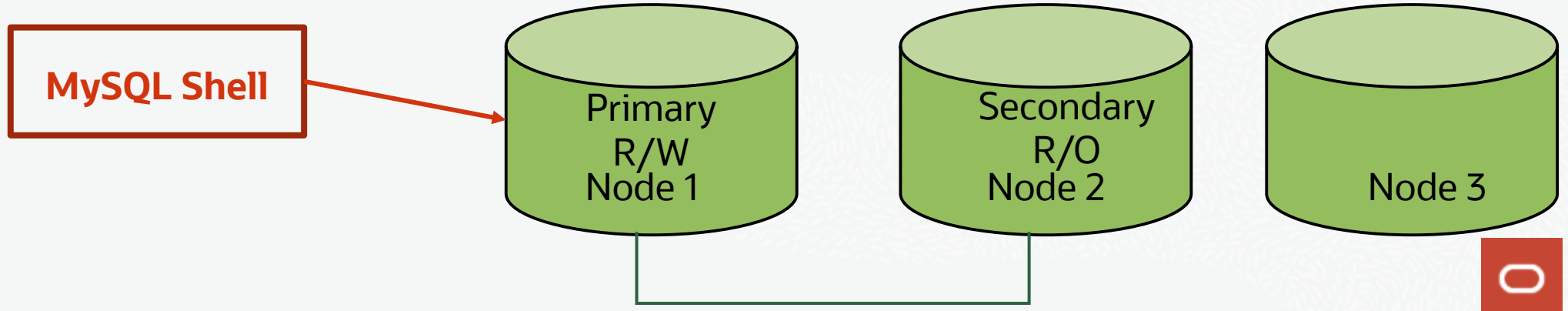
MySQL Shell : Setup InnoDB Cluster easily

- Connect to one server and create cluster with this member

```
MySQL  JS > \connect admin@10.0.0.75
MySQL  10.0.0.75:33060+ ssl  JS > Cluster = dba.createCluster("testcluster")
```

- Add the other two members to the cluster

```
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.76:3306")
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.77:3306")
```



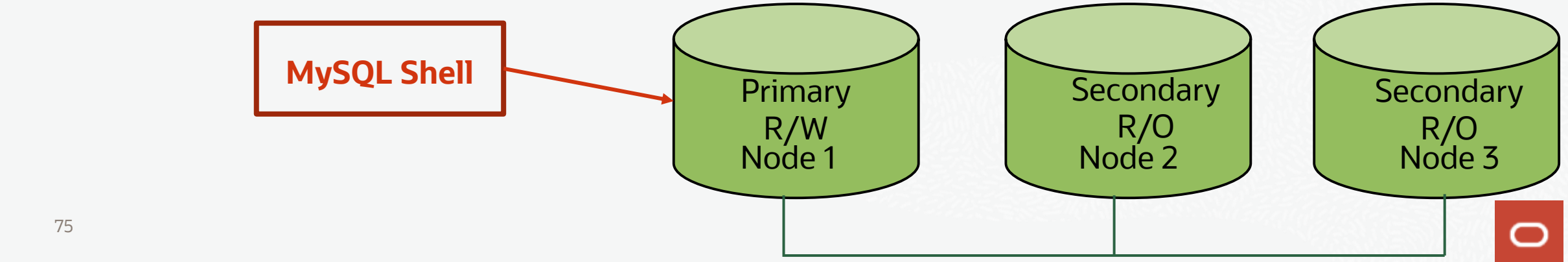
MySQL Shell : Setup InnoDB Cluster easily

- Connect to one server and create cluster with this member

```
MySQL  JS > \connect admin@10.0.0.75
MySQL  10.0.0.75:33060+ ssl  JS > Cluster = dba.createCluster("testcluster")
```

- Add the other two members to the cluster

```
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.76:3306")
MySQL  10.0.0.75:33060+ ssl  JS > Cluster.addInstance("admin@10.0.0.77:3306")
```



Check the status of the cluster

```
MySQL 10.0.0.75:33060+ ssl JS > Cluster.status()
```


```
"clusterName": "testCluster",
"defaultReplicaSet": {
    "name": "default",
    "primary": "student102-serverb:3307",
    "ssl": "REQUIRED",
    "status": "OK",
    "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
    "topology": {
        "student102-servera:3307": {
            "address": "student102-servera:3307",
            "mode": "R/O",
            "readReplicas": {},
            "replicationLag": null,
            "role": "HA",
            "status": "ONLINE",
            "version": "8.0.18"
        },
        "student102-serverb:3307": {
            "address": "student102-serverb:3307",
            "mode": "R/W",
            "readReplicas": {},
            "replicationLag": null,
            "role": "HA",
            "status": "ONLINE",
            "version": "8.0.18"
        },
        "student102-serverb:3317": {
            "address": "student102-serverb:3317",
            "mode": "R/O",
            "readReplicas": {},
            "replicationLag": null,
            "role": "HA",
            "status": "ONLINE",
            "version": "8.0.18"
        }
    },
    "topologyMode": "Single-Primary"
```

Backported to 5.7.22

Highly Efficient Replication Applier

Write set parallelization

WRITESET dependency tracking allows **applying** a **single threaded** workload in **parallel**.

- Delivers the **best throughput** of the three dependency trackers, at **any** concurrency level.

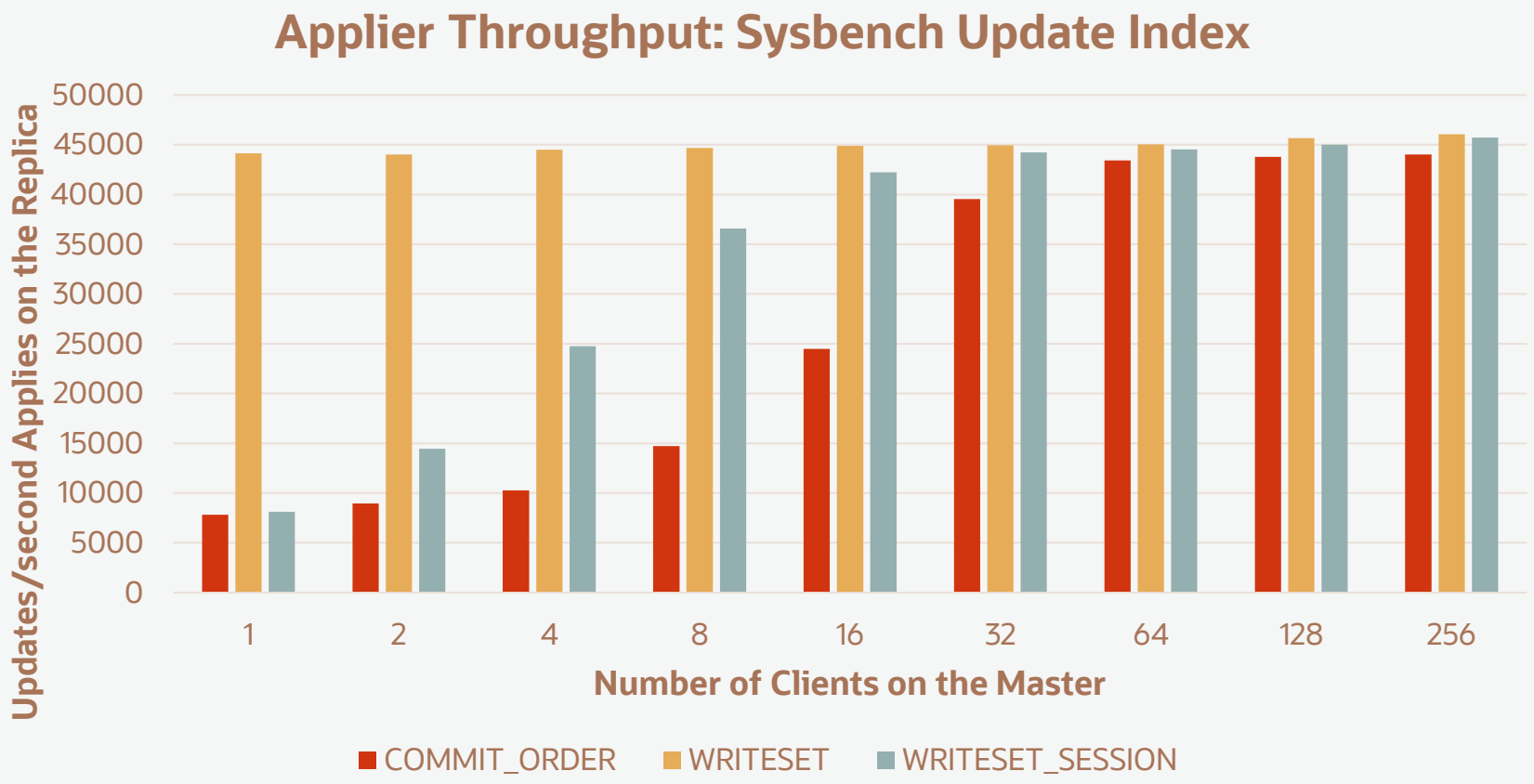
WRITESET_SESSION in addition to writesets **tracks sessions dependencies as well**. Two transactions executed on the same session are always scheduled in execution order on replica servers.

Fast Group Replication recovery – time to catch up.

Backported to 5.7.22

Highly Efficient Replication Applier

Write set parallelization



ORACLE