MariaDB ColumnStore Scalability and Transactions

thinQ

there is a better way
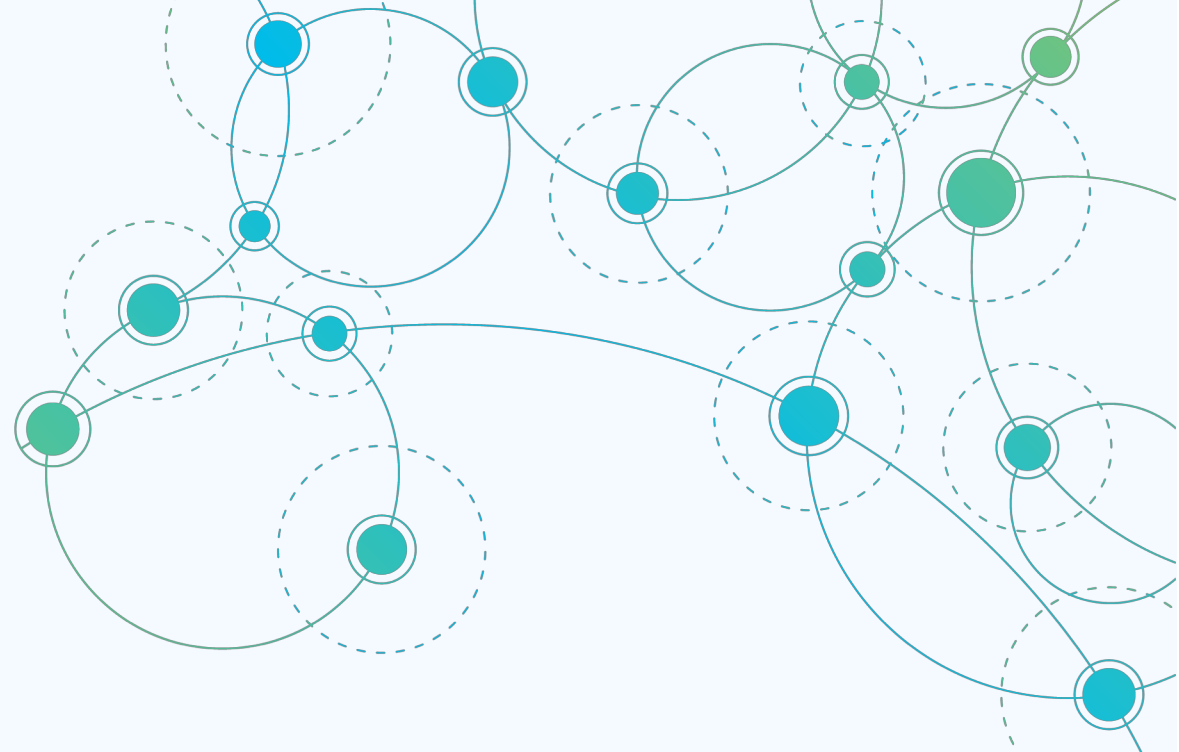
Headquartered in Raleigh — part of North Carolina's booming Research Triangle — **thinQ** is a cloud-based software company that provides Communications Platform as a Service (CPaaS) solutions for the telecommunications industry

Outline

1. Scalability beyond many DBRoots

2. Transaction management for High Availability streaming

Scaling Up MariaDB ColumnStore

MariaDB MaxScale
(Proxy)

User Modules
(SQL & Aggregation)

Performance Modules
(Storage and Access Primitives)
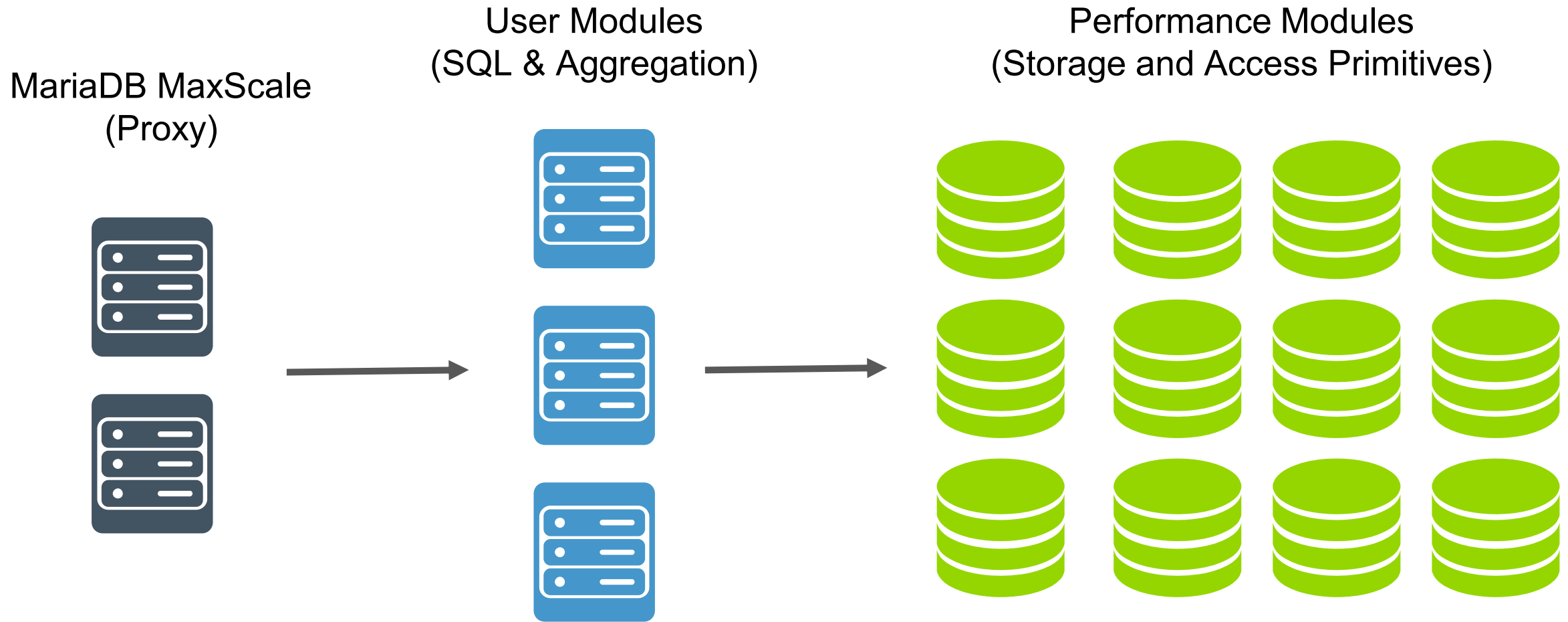
Picture from Thomas Boyd

Scaling Up Analytics: Case for a Graceful Degradation

- Consider a ColumnStore system scaled up beyond ten DBRoot nodes
  - With one DBRoot node down, you still want to use the remaining 90% of data

MariaDB MaxScale
(Proxy)

User Modules
(SQL & Aggregation)

Performance Modules
(Storage and Access Primitives)
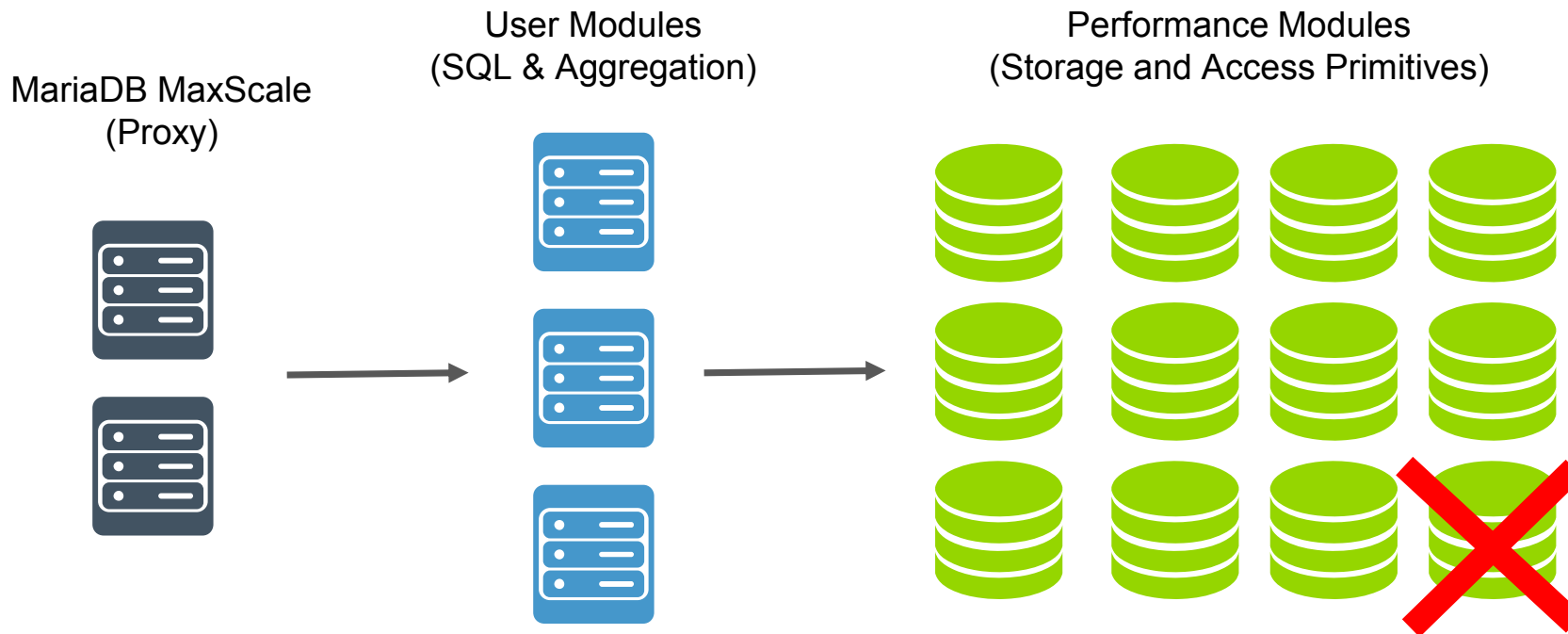
- Because your statistical findings from the 90% of data will practically be the same as those derived from all 100%

thinQ

What happens when one PM/DBRoot is down?

Let's find out what happens using ColumnStore demo

Picture from Thomas Boyd
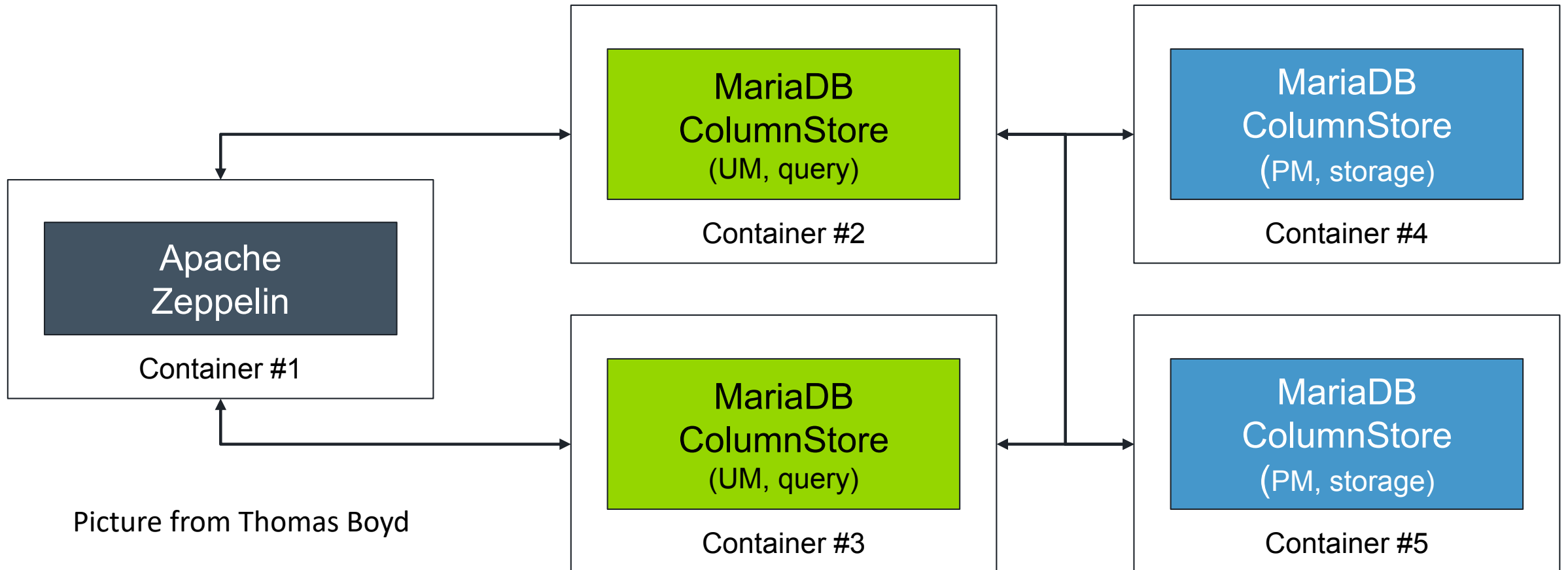
Using ColumnStore Demo without PM2/DBRoot node

```
docker stop columnstore_zeppelin_pm2_1
```

select count(*) from books;

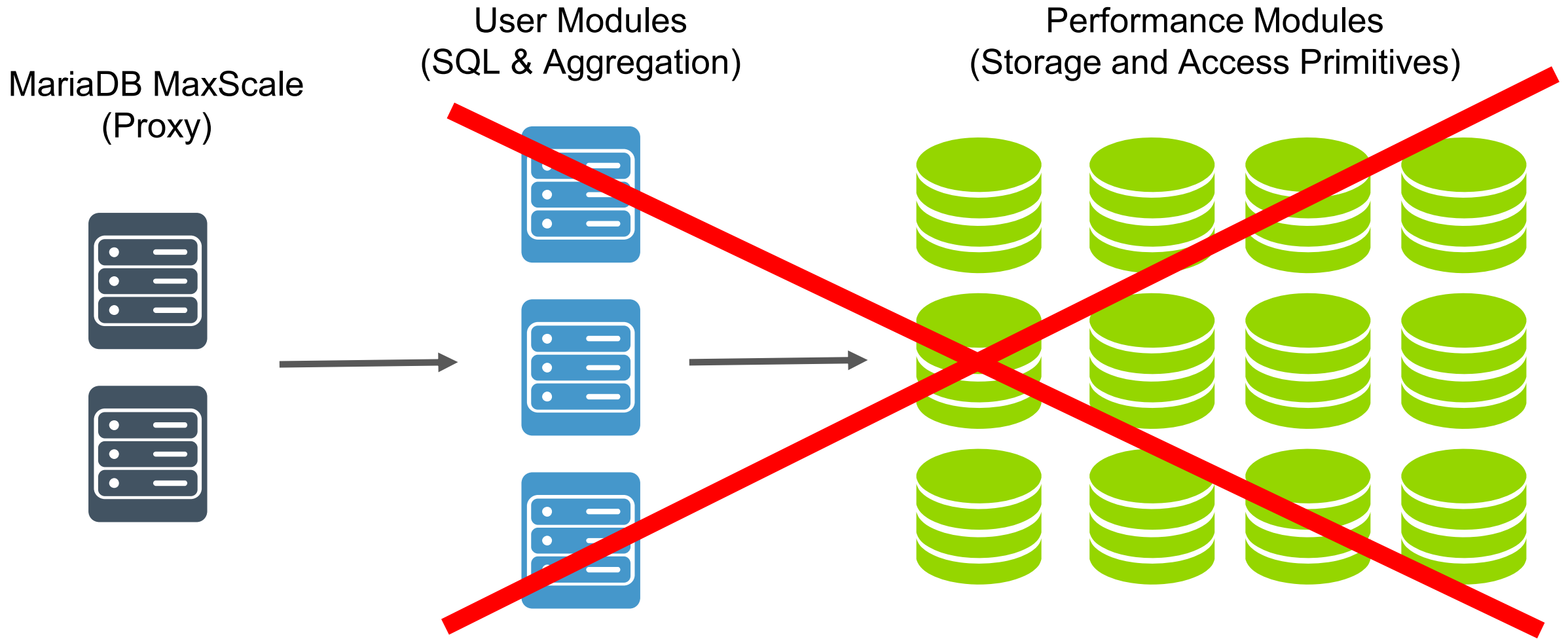ERROR 1815 (HY000): Internal error: InetStreamSocket::readToMagic: Remote is closed

```
Component        Status                      Last Status Change
---------        ------                      ------------------
System           ACTIVE                      Thu Feb 14 21:31:38 2019

Module um1       FAILED                      Thu Feb 14 21:31:29 2019
Module um2       ACTIVE                      Thu Feb 14 21:31:21 2019
Module pm1       ACTIVE                      Thu Feb 14 21:13:19 2019
Module pm2       AUTO_DISABLED/DEGRADED      Thu Feb 14 21:29:29 2019
```

select count(*) from books;

ERROR 1815 (HY000): Internal error: st: 0 TupleBPS::sendPrimitiveMessages() caught an exception: IDB-2034: At least one DBRoot required for that query is offline.

thinQ

Single DBRoot failure could make whole system unusable

MariaDB MaxScale
(Proxy)

User Modules
(SQL & Aggregation)

Performance Modules
(Storage and Access Primitives)

- Moreover, with more than ten DBRoots, the probability of having one down increases

Attempt to delete missing DBRoot from the Extent Map

```
editem -h

-p dbr         delete all extents on dbroot dbr
```

```
/usr/local/mariadb/columnstore/bin/editem -p 2
/data/buildbot/bb-worker/centos7/mariadb-columnstore-engine/tools/editem/editem.cpp@756: assertion 'rc == 0' failed
terminate called after throwing an instance of 'logging::IDBExcept'
  what():  IDB-2035: An internal error occurred.  Check the error log file & contact support.
Aborted
```

```
tail /var/log/mariadb/columnstore/err.log
Feb 14 21:30:49 0cd631a37c0c controllernode[35475]: 49.549174 |0|0|0| C 29 CAL0000: DBRM Controller: network error distributing command to worker 1
Feb 14 21:30:49 0cd631a37c0c controllernode[35475]: 49.550630 |0|0|0| C 29 CAL0000: DBRM Controller: undo(): warning, could not contact worker number 1
Feb 14 21:30:49 0cd631a37c0c controllernode[35475]: 49.550706 |0|0|0| C 29 CAL0000: DBRM Controller: Caught network error.  Sending command 43, length 5.  Setting read-only mode.
Feb 14 21:30:49 0cd631a37c0c Calpont[35568]: 49.554098 |0|0|0| E 00 CAL0000: /data/buildbot/bb-worker/centos7/mariadb-columnstore-engine/tools/editem/editem.cpp@756: assertion 'rc == 0' failed
```

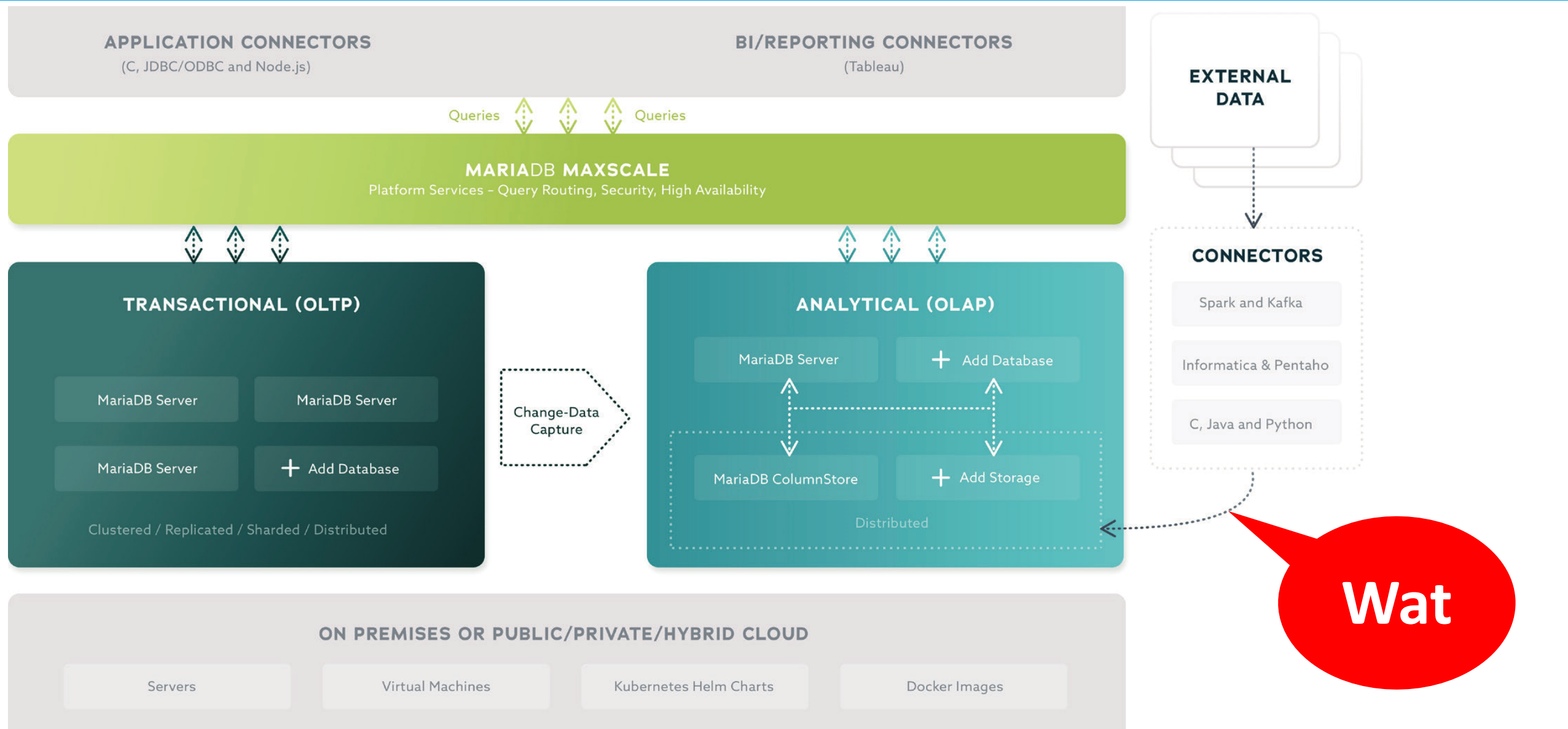https://github.com/mariadb-corporation/mariadb-columnstore-engine/blob/master/tools/editem/editem.cpp#L749

https://github.com/mariadb-corporation/mariadb-columnstore-engine/blob/master/versioning/BRM/extentmap.cpp#L5335

```
/usr/local/mariadb/columnstore/bin/editem -d
Col OID = 3071, NumExtents = 2, width = 4
656384 - 660479 (4096) min: 1, max: 5, seqNum: 1, state: valid, fbo: 0, DBRoot: 1, part#: 0, seg#: 0, HWM: 494; status: avail
795648 - 799743 (4096) min: 1, max: 5, seqNum: 1, state: valid, fbo: 0, DBRoot: 2, part#: 0, seg#: 1, HWM: 492; status: avail
```
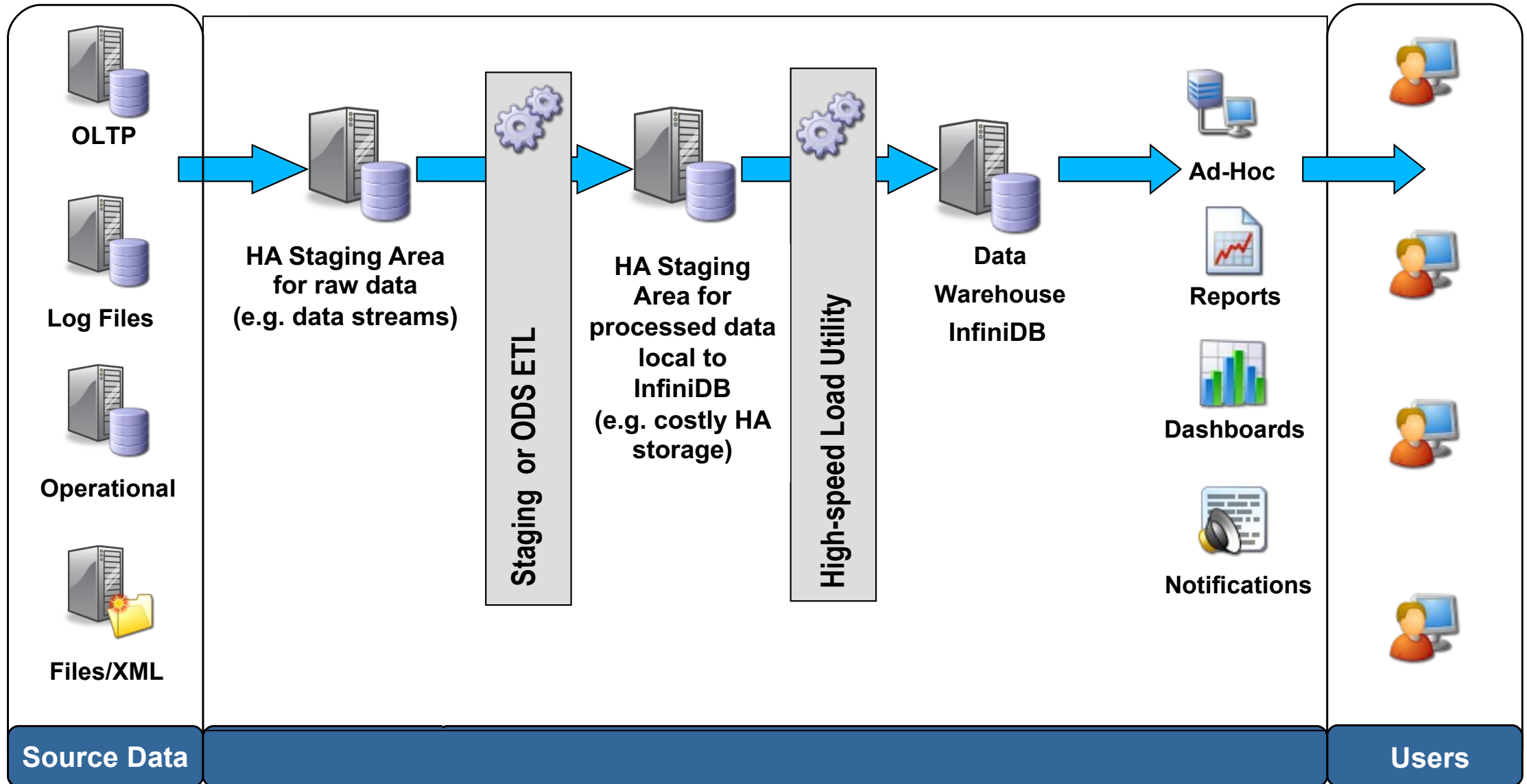
thinQ
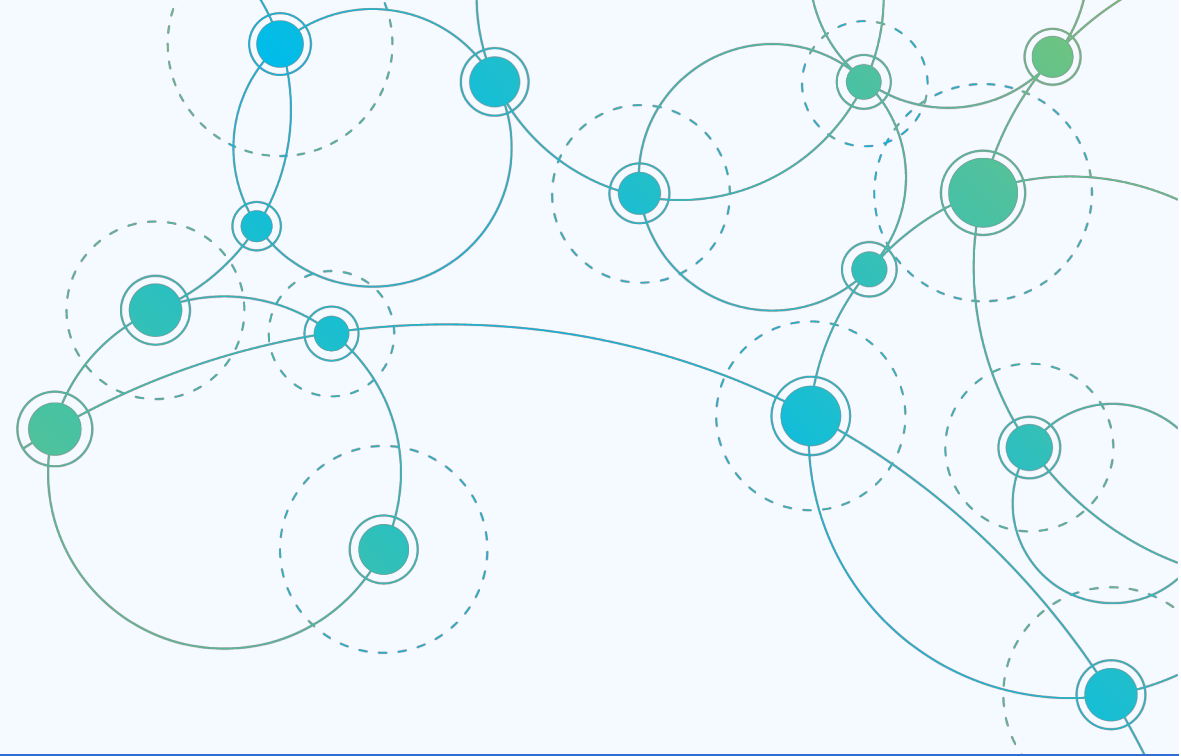
Could editem be extended to mask the DBRoot, which is down temporarily?

High Availability with ColumnStore Bulk Write SDK

Legacy ColumnStore Data Processing Pipeline

OLTP

Log Files

Operational

Files/XML

HA Staging Area for raw data (e.g. data streams)

Staging or ODS ETL

HA Staging Area for processed data local to InfiniDB (e.g. costly HA storage)

High-speed Load Utility

Data Warehouse InfiniDB

Ad-Hoc

Reports

Dashboards

Notifications

Source Data

Users

New ColumnStore Data Processing Pipeline
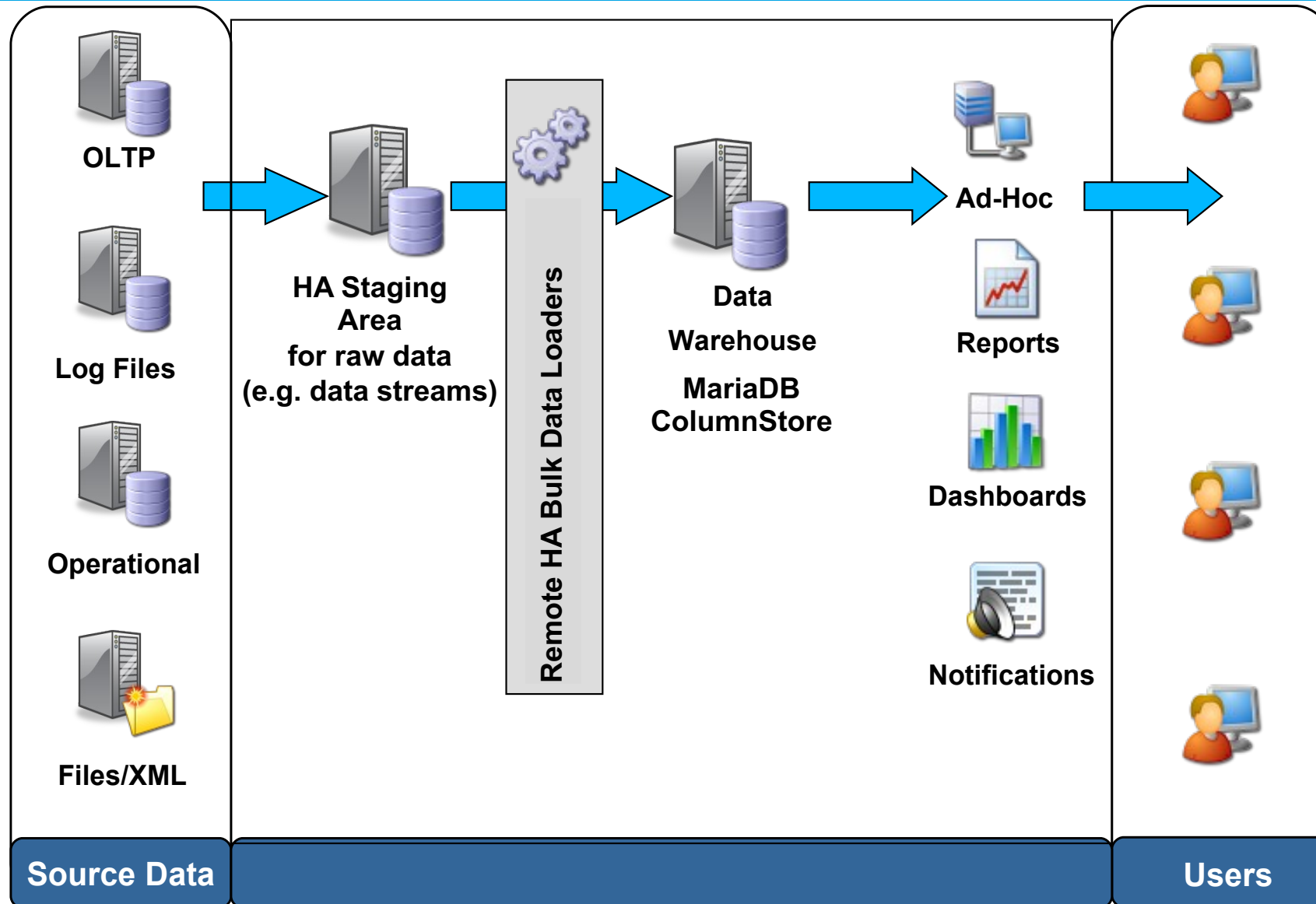
- Improvements in data processing pipeline provided by the **remote** mcsimport

Source Data

OLTP

Log Files

Operational

Files/XML

HA Staging Area for raw data (e.g. data streams)

Remote HA Bulk Data Loaders

Data Warehouse MariaDB ColumnStore

Ad-Hoc

Reports

Dashboards

Notifications

Users

thinQ

High Availability with ColumnStore Bulk Write SDK

- By their nature, data streaming applications run continuously
  - Redundant applications could increase data streaming uptime, since if one application fails, a second application would still be running
- How do you implement HA/failover between data streaming applications using bulk write SDK remotely?
  - MariaDB developers provided functions to view and clear table locks remotely
- **In contrast, MariaDB Server rolls back transaction upon client failure**
  - Perhaps the MariaDB Platform X3 may implement a similar behavior for ColumnStore
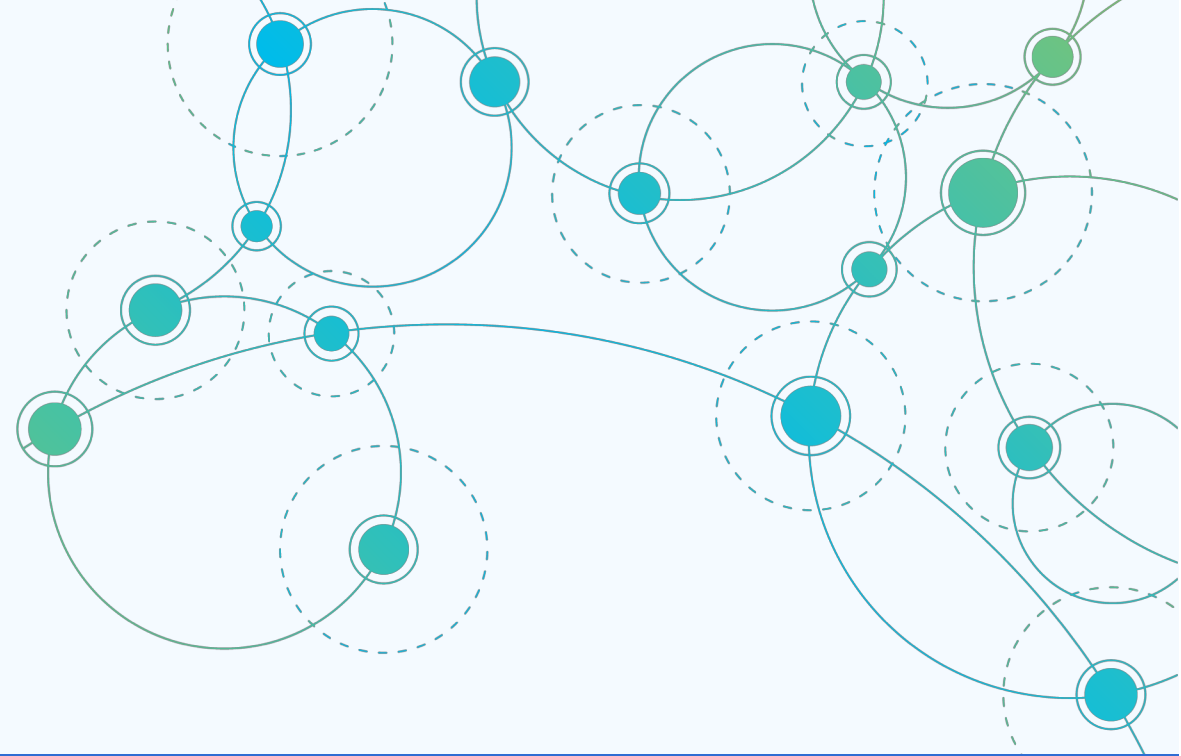
thinQ

From client connection to physical storage

Deep Dive: InnoDB Transactions and Write Paths

From the client connection to physical storage

Marko Mäkelä, Lead Developer InnoDB
Michaël de Groot, MariaDB Consultant

Can MariaDB Platform X3 implement ColumnStore transaction management for streaming clients similar to the MariaDB Server client transaction management?

Summary: Requests to ColumnStore Developers

- Extend editem to mask DBRoot that is down temporarily
  - This would scale ColumnStore further
- Integrate ColumnStore transaction management with MariaDB Server transaction management
  - This would simplify HA for remote data streaming clients

thinQ