Configuring and Tuning an Availability Group

Russ Thomas
DATA ARCHITECT

@sqljudo

We've established Always On foundation

We've established AG without Always On

This module is pure DBA

Node

An individual server participating in a cluster

Replica

An individual SQL instance participating in a SQL Server Availability Group

Overview

Mechanism by which replicas stay in synch

- Synchronous mode
- Asynchronous mode
- Transaction log queues

Advanced settings for secondary replicas

- Read-only replicas
- Offloading backups

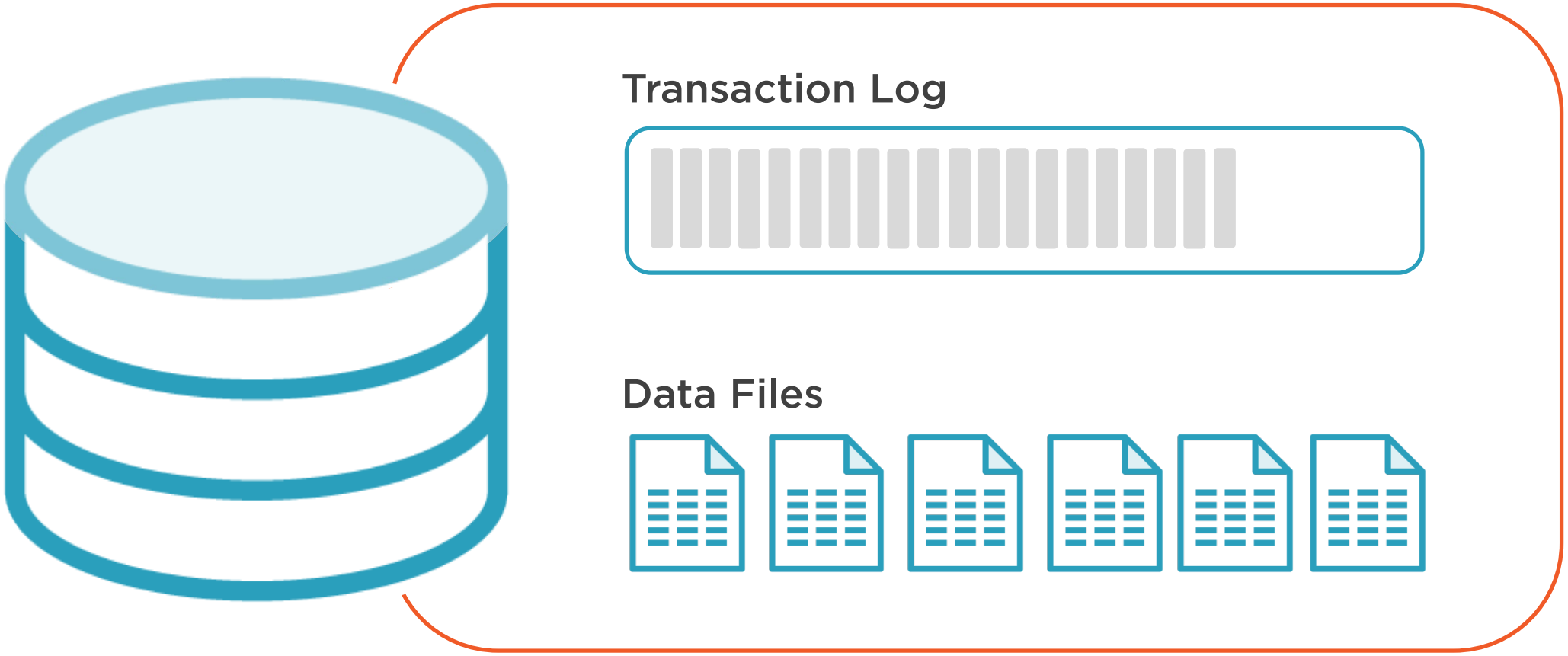
Establishing an AG listener

- Automatic failovers
- Active directory
- Demonstrations

Transaction

Logical unit of work

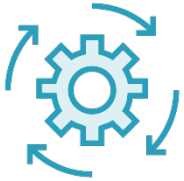
SQL Server Database



Additional Training on Transaction Log

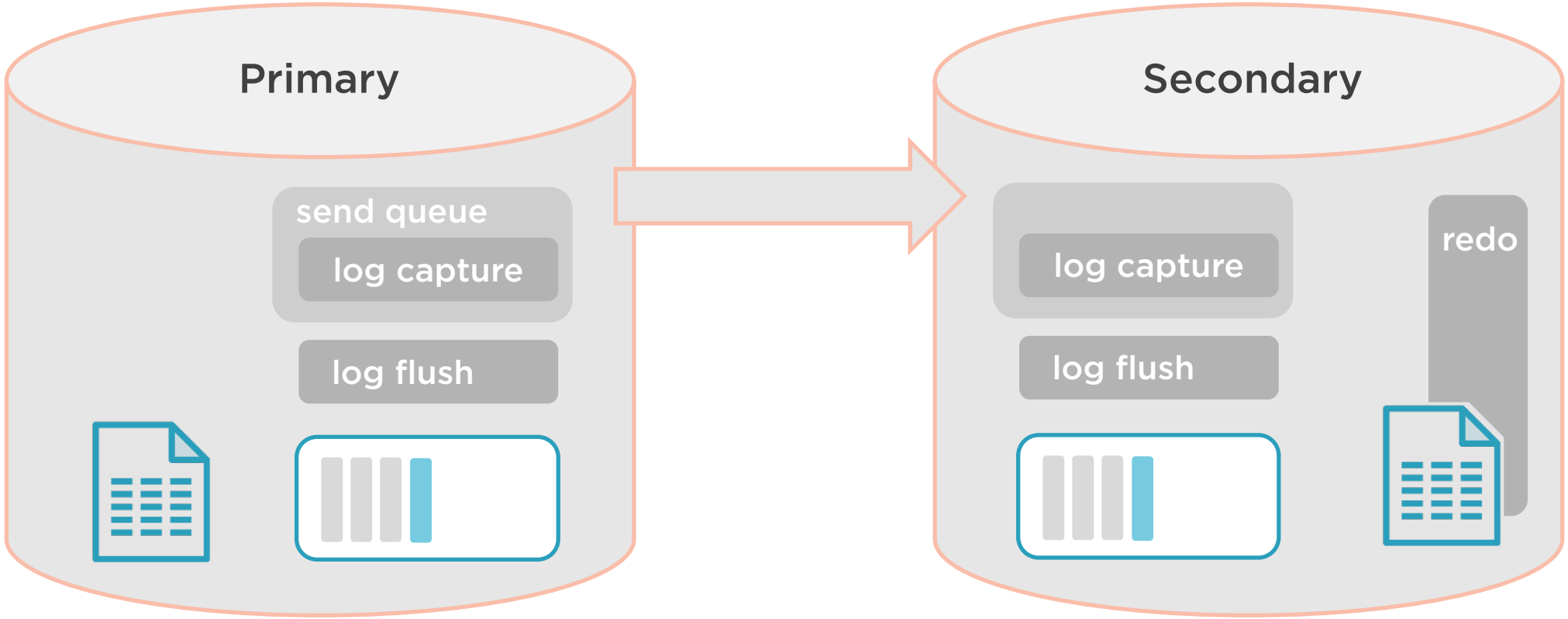
Russ Thomas https://bit.ly/2BUm3Tv

Paul Randal https://bit.ly/2fG5UlJ



Understand ‘REDO’ and ‘UNDO’ functions of the log

Data Synchronization



https://bit.ly/2pyLtOS

Data Synchronization

Synchronous

The transaction must wait for confirmation from all synchronous logs primary to considering it's self committed

No risk of data loss across replicas

Potential for poor performance

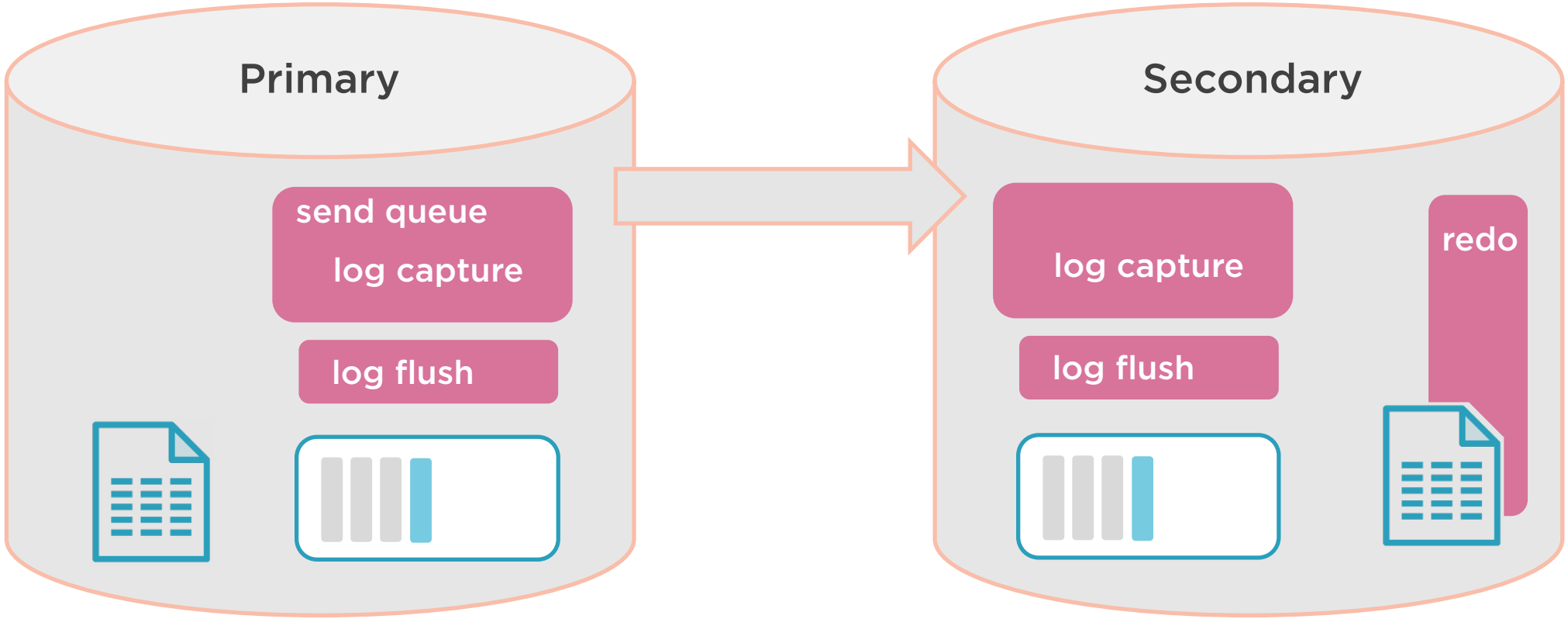
Asynchronous

The transaction does not have to wait for confirmation from the asynchronous replicas before considering it's self committed

Potential for higher performance

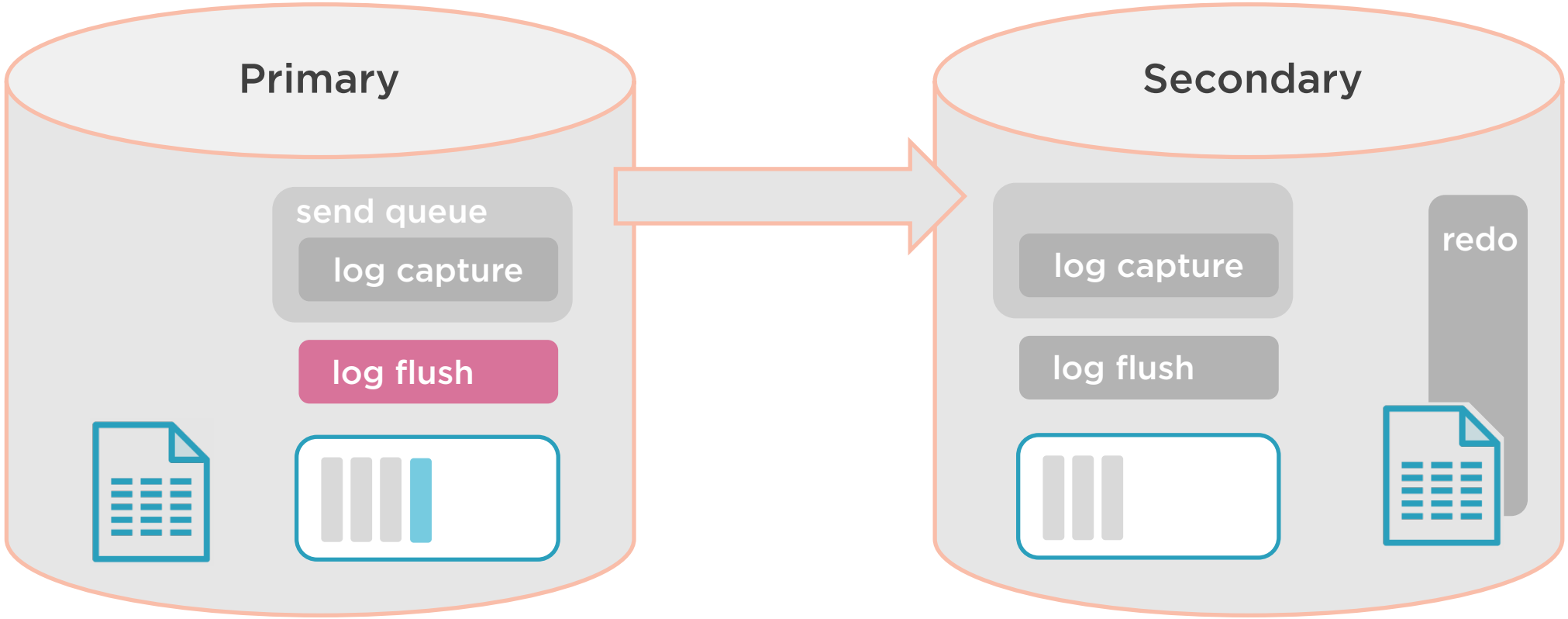
Includes a risk of data loss

Synchronous Means Latency Potential



https://bit.ly/2pyLtOS

Asynchronous Means Data Loss Potential

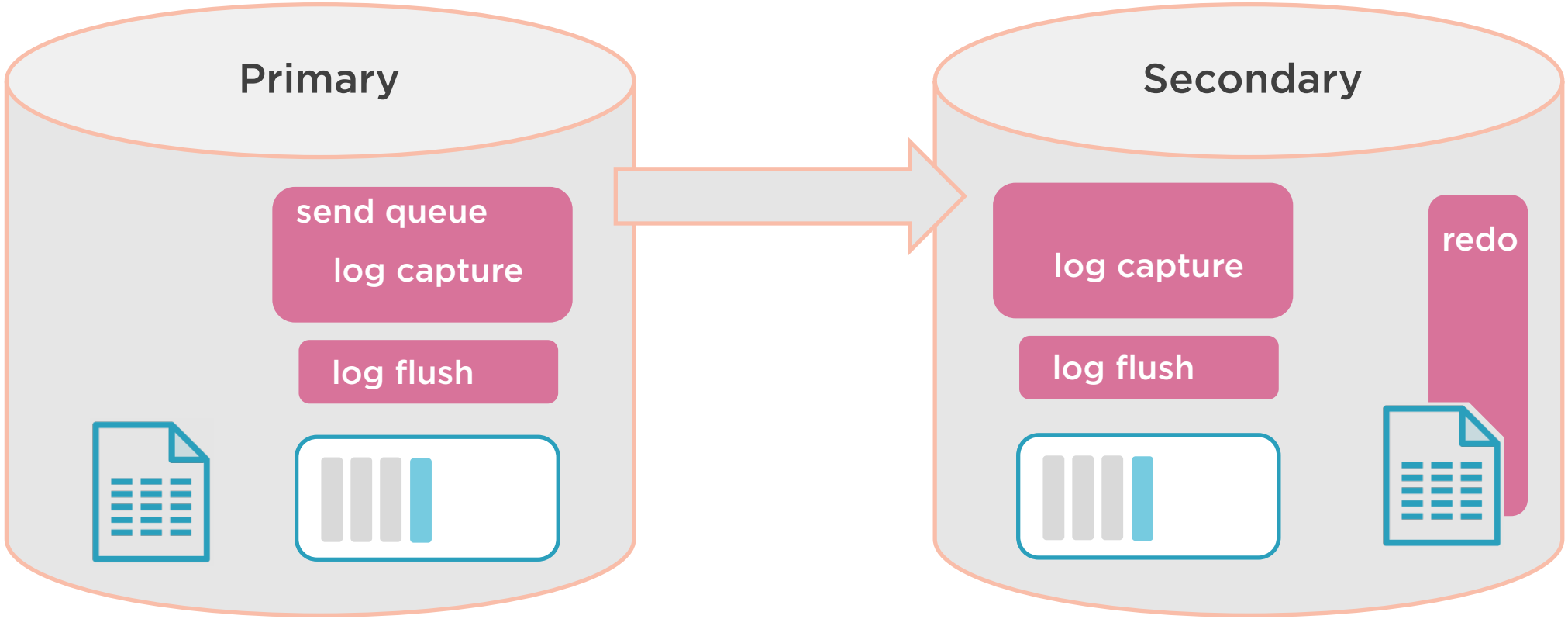


https://bit.ly/2pyLtOS

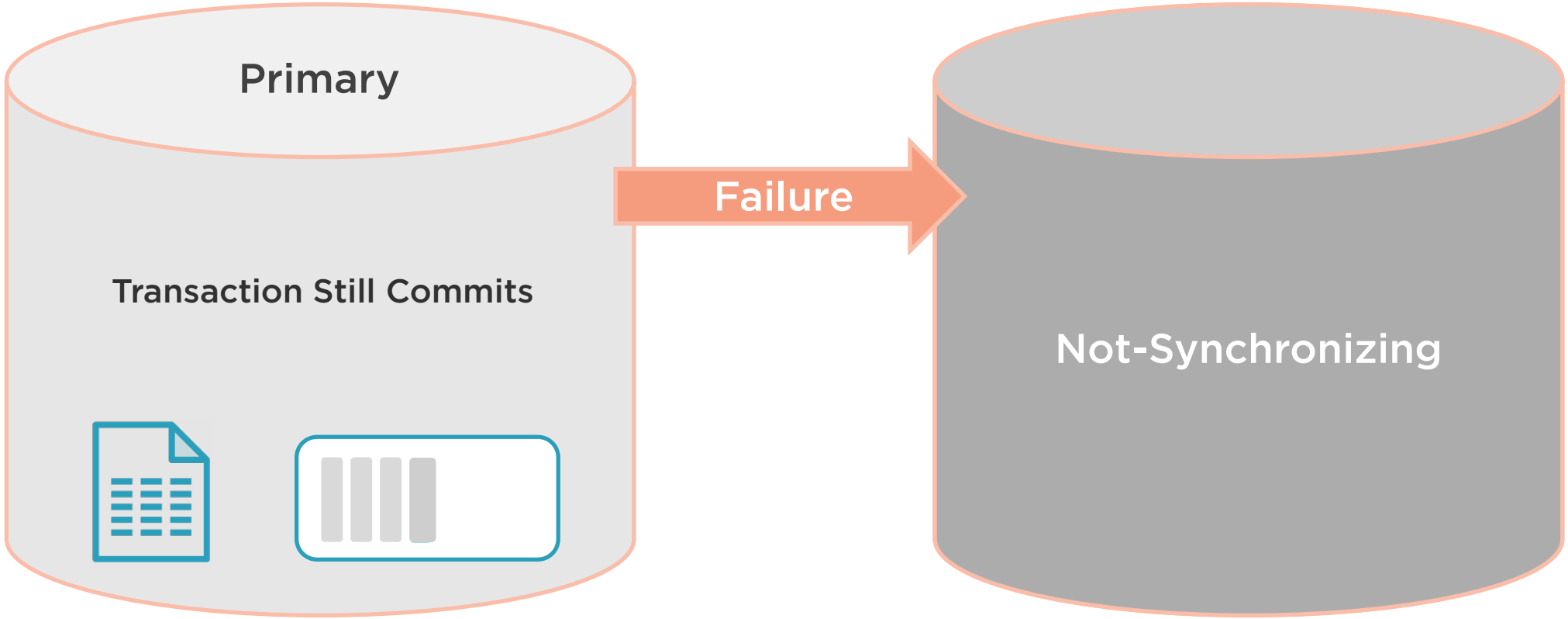
Your monitoring tools should alert on non-standard configurations

Avoid forcing quorum outside of SQL Server Availability Group

Synchronous = Dual Commit



Synchronous / Replica Failure



Failure Condition Levels

Level 1 = SQL Service down

Level 2 = Unable to connect to cluster + timeout reached

Level 3 = Critical SQL error (default setting)

Failure Condition Levels

Level 4 = Moderate SQL error

Level 5 = Any qualifying error

SQL Server flexible failover policy

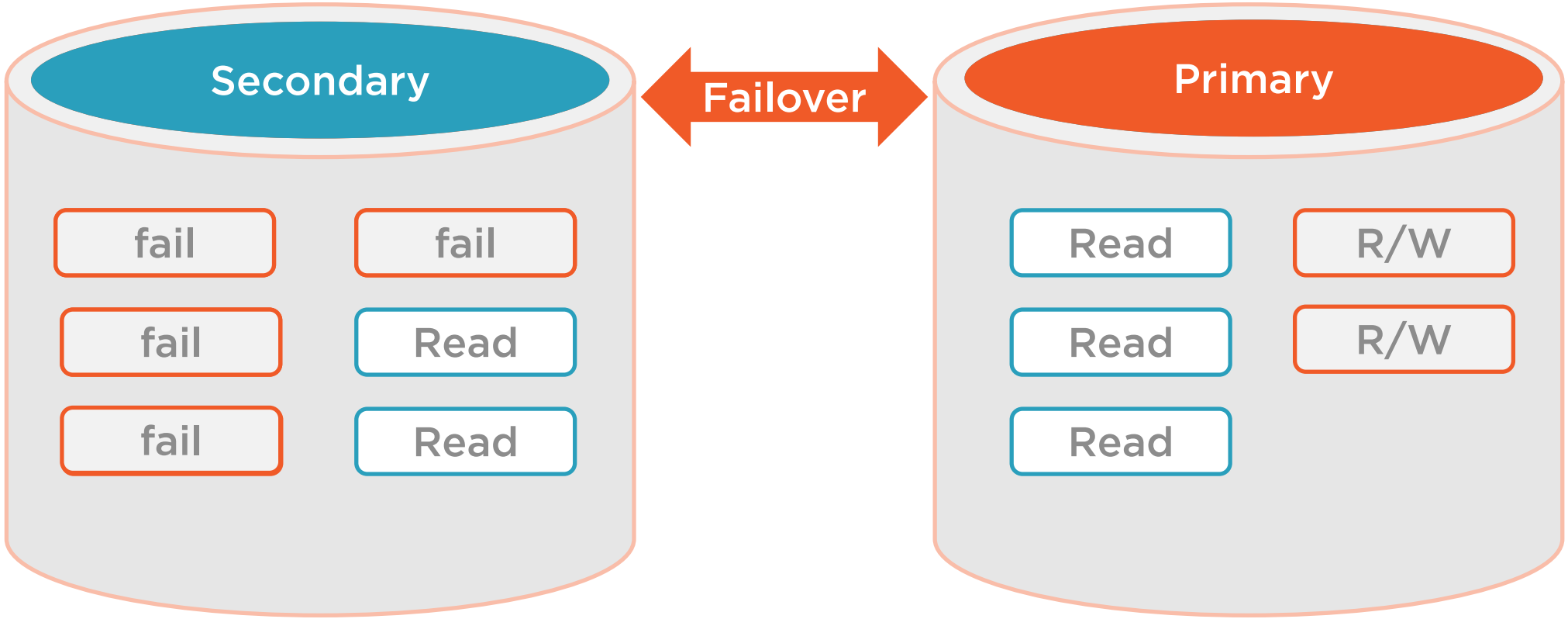
Read-only Secondary

- Read only connections after role change
- Read only limitations
- Asynch versus synch behavior
- Isolation levels

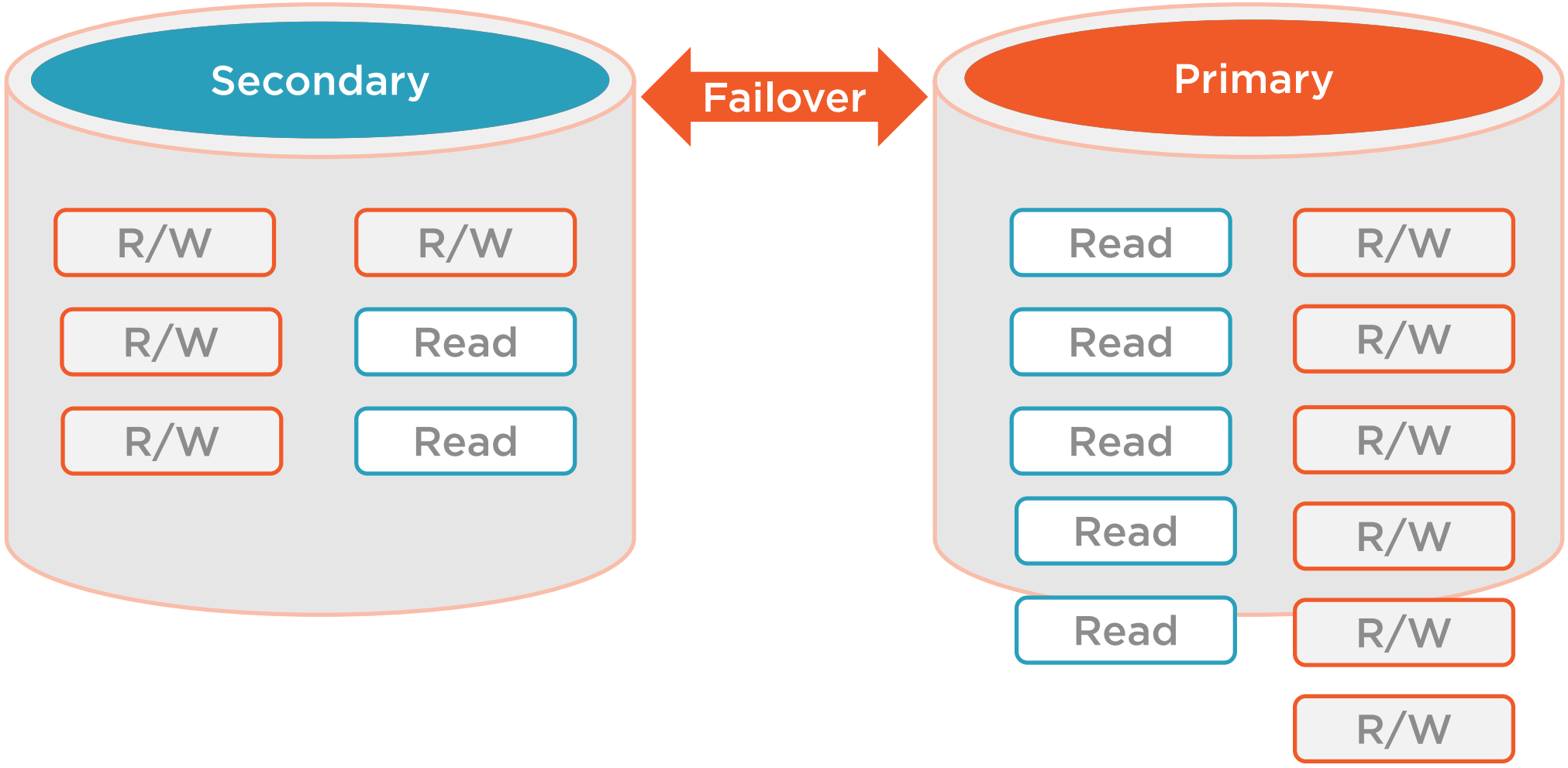
Read-intent Connection String Sample

```
Data Source=AbacosB; Initial Catalog=AbacosCRM;
Integrated Security=True; ApplicationIntent=ReadOnly;
```

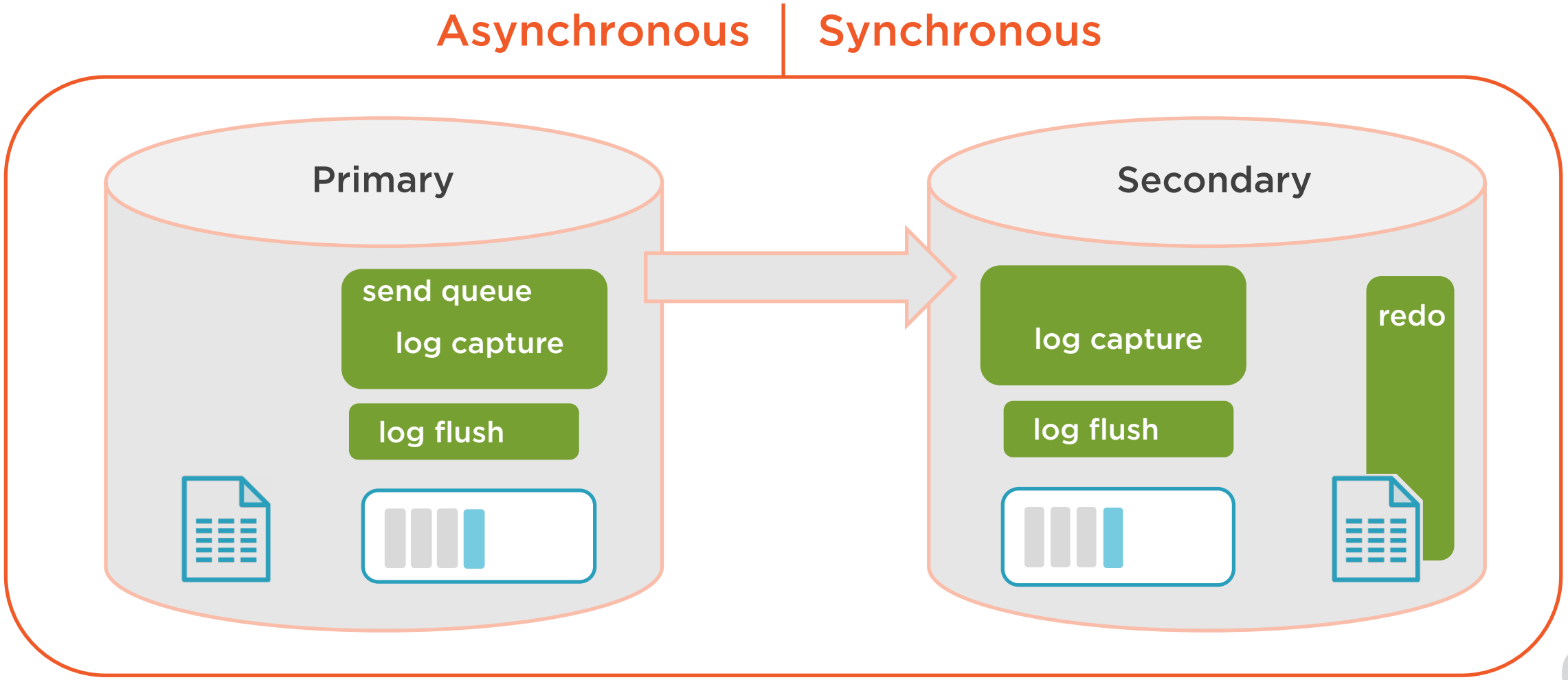
Readable Secondary Considerations



Readable Secondary Considerations



Read Only Secondary



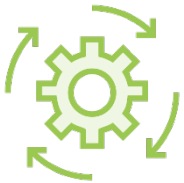
Read Only Secondary Latency

Should be considered near real time for most circumstances

Factors include distance, bandwidth, and server workload

Latency will be between a few milliseconds to potentially minutes

Read Only Secondary Isolation Level



All queries against a readable secondary are snapshot isolation

Keeps transactions from blocking the REDO queue

Query hints are ignored on readable secondaries

The flexibility of readable secondaries becomes much greater with a listener

```
READ_ONLY_ROUTING_LIST=(('BoxA','BoxB'),('BoxC','BoxD'))
```

Read Only Routing List Load Balancing

Read-intent Connection String Sample

```
Data Source=AbacosB; Initial Catalog=AbacosCRM;
Integrated Security=True; ApplicationIntent=ReadOnly;
```

Summary

- Began with default availability group
- Established a multi-subnet cluster
- Established a synchronous HA pair and an asynchronous DR replica
- Configured read-only secondaries
- Configured back-up policy preferring secondary replicas
- Added a listener to route primary and secondary traffic

What Remains

Troubleshooting common issues

Performance tuning

Rights required to build this solution