

Variations on PostgreSQL Replication

Boriss Mejías
Holistic System Software Engineer
Air Guitar Player

3 February 2021



Variations on PostgreSQL Replication

Boriss Mejías
Solution Architect
Air Guitar Player

3 February 2021



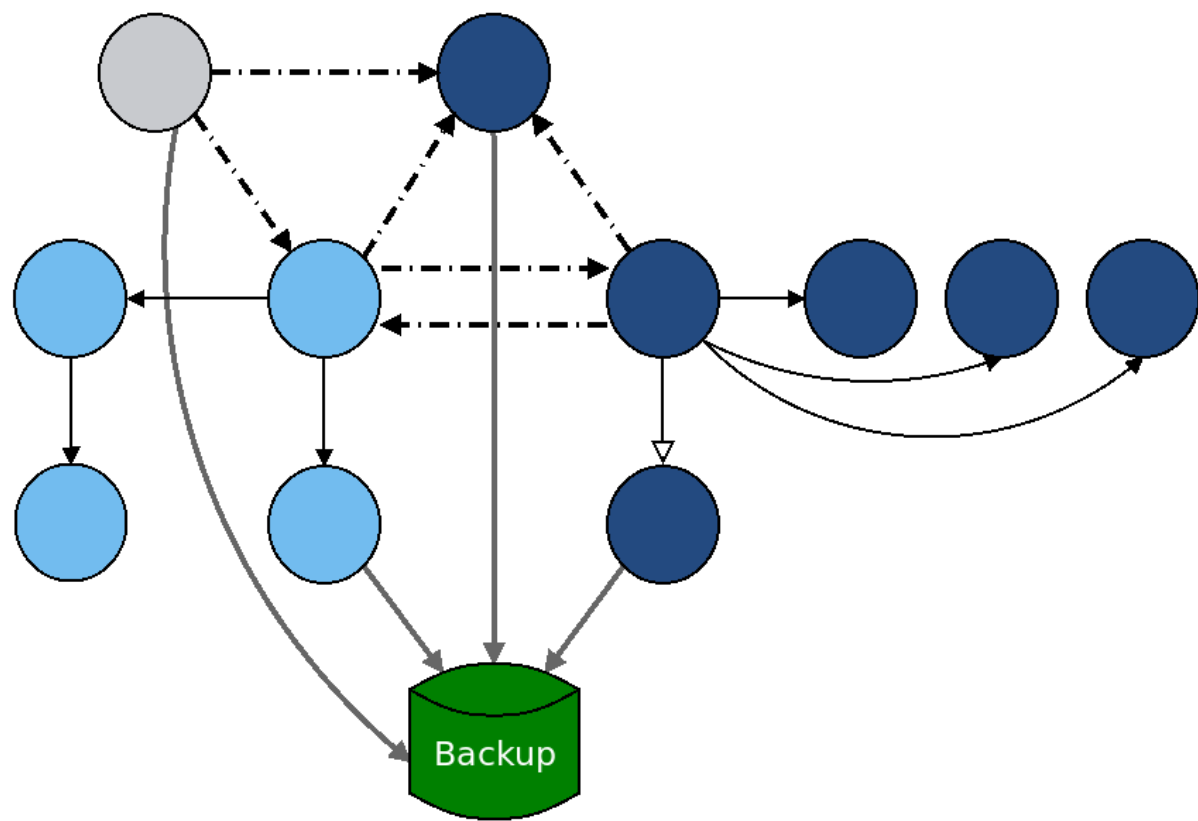
Why this Talk?



Open Source Music



By Kiko Loureiro
<https://gt.kikoloureiro.com/open-source/>



Variations from the roots

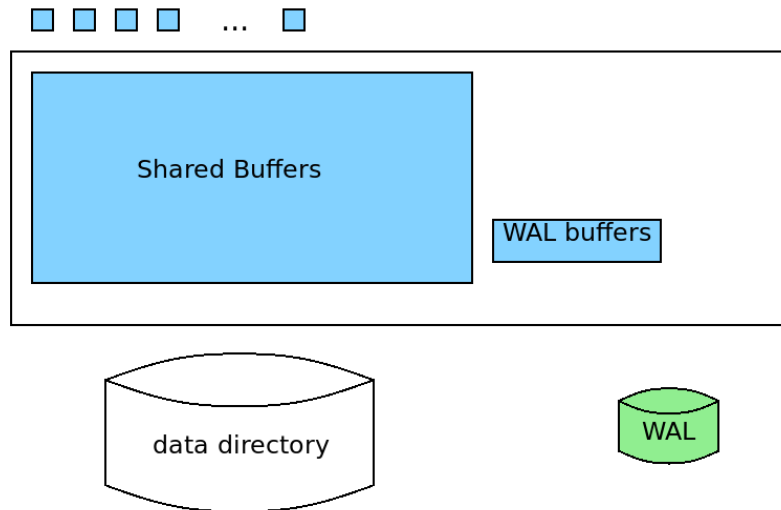
Heavy Metal has its roots in Blues & Rock

Variations from the roots

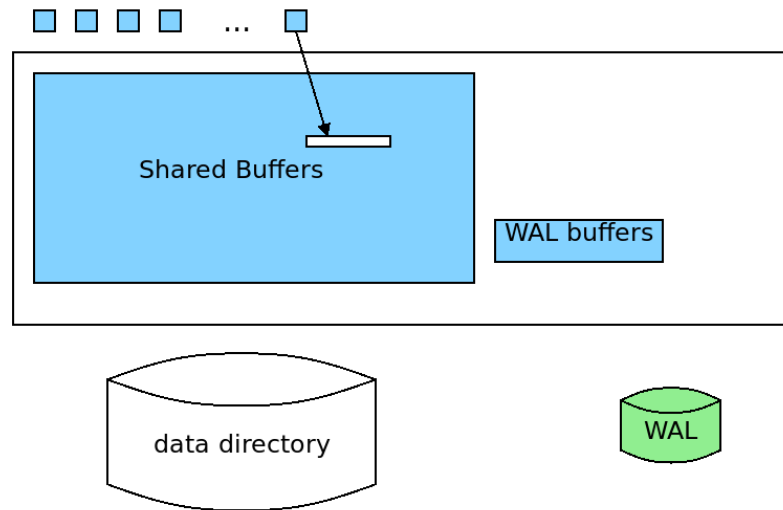
Heavy Metal has its roots in Blues & Rock

Replication has its roots in Backup & Recovery

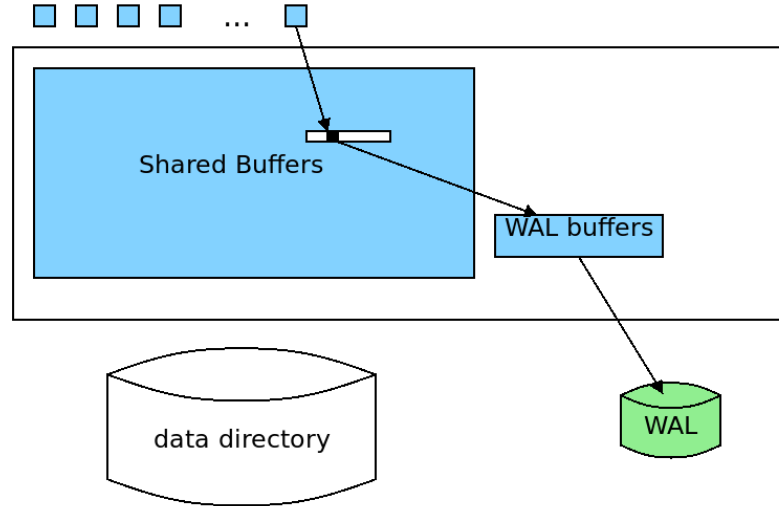
PostgreSQL resilient to crash



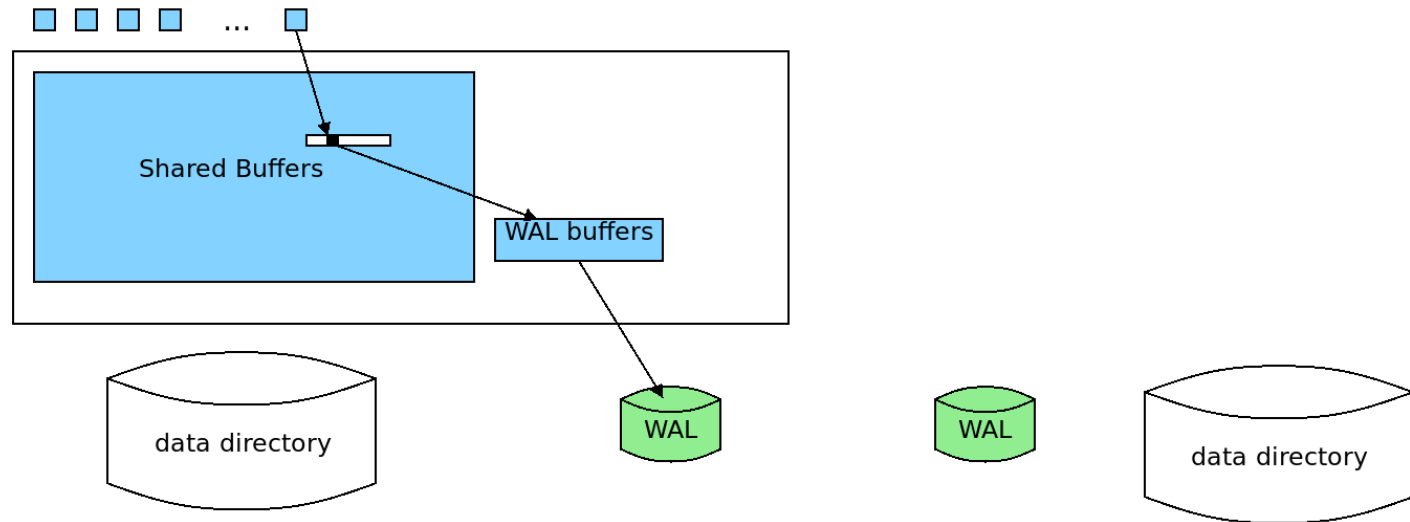
PostgreSQL resilient to crash



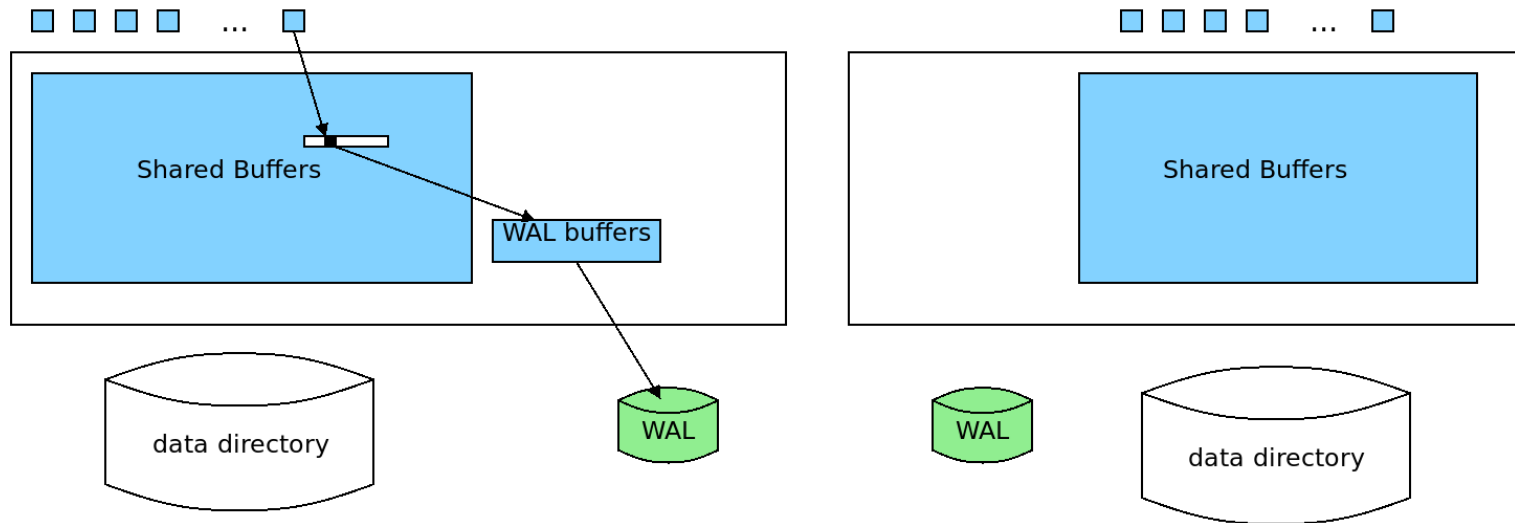
PostgreSQL resilient to crash



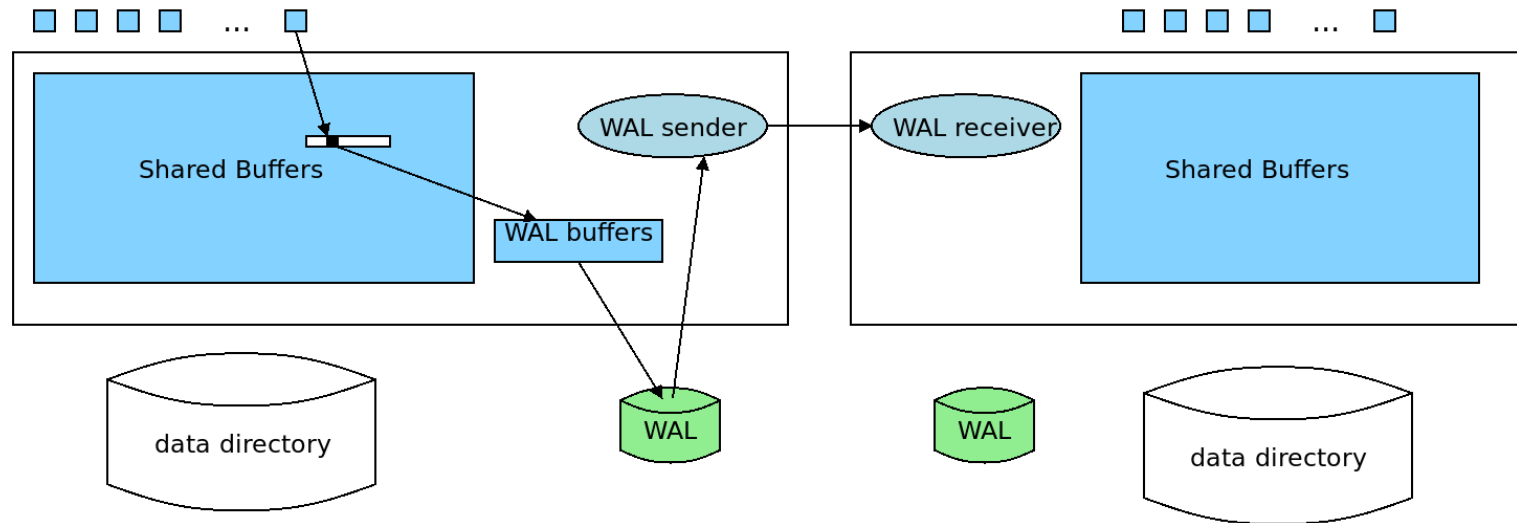
Physical backup



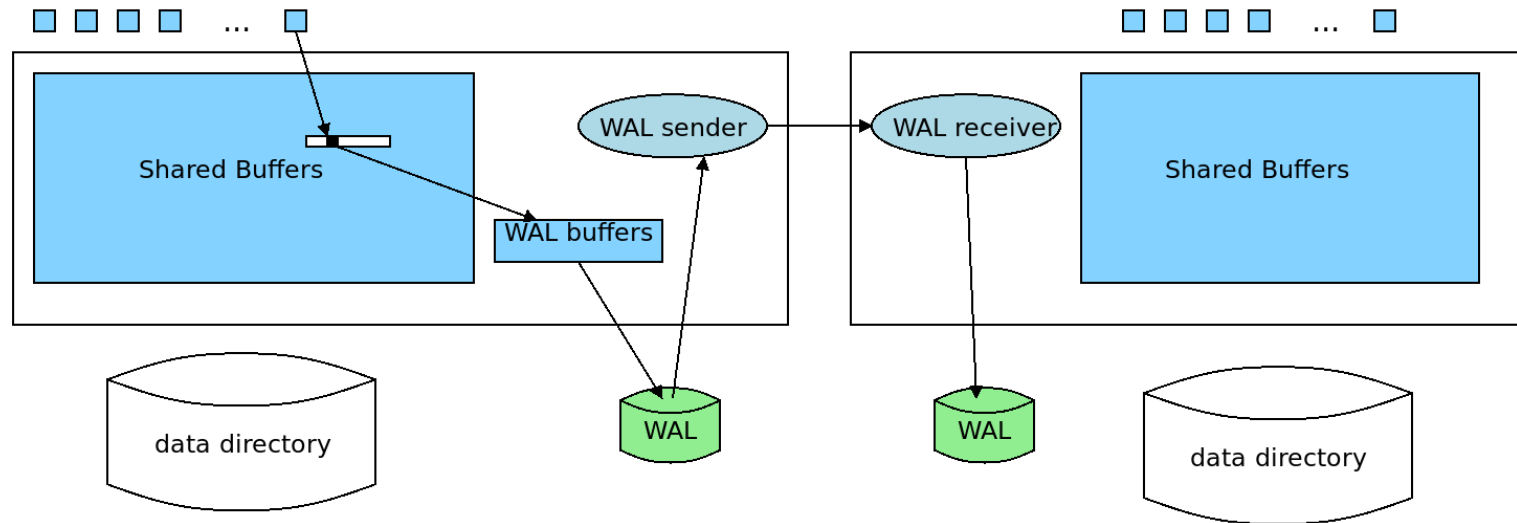
Physical backup and recovery



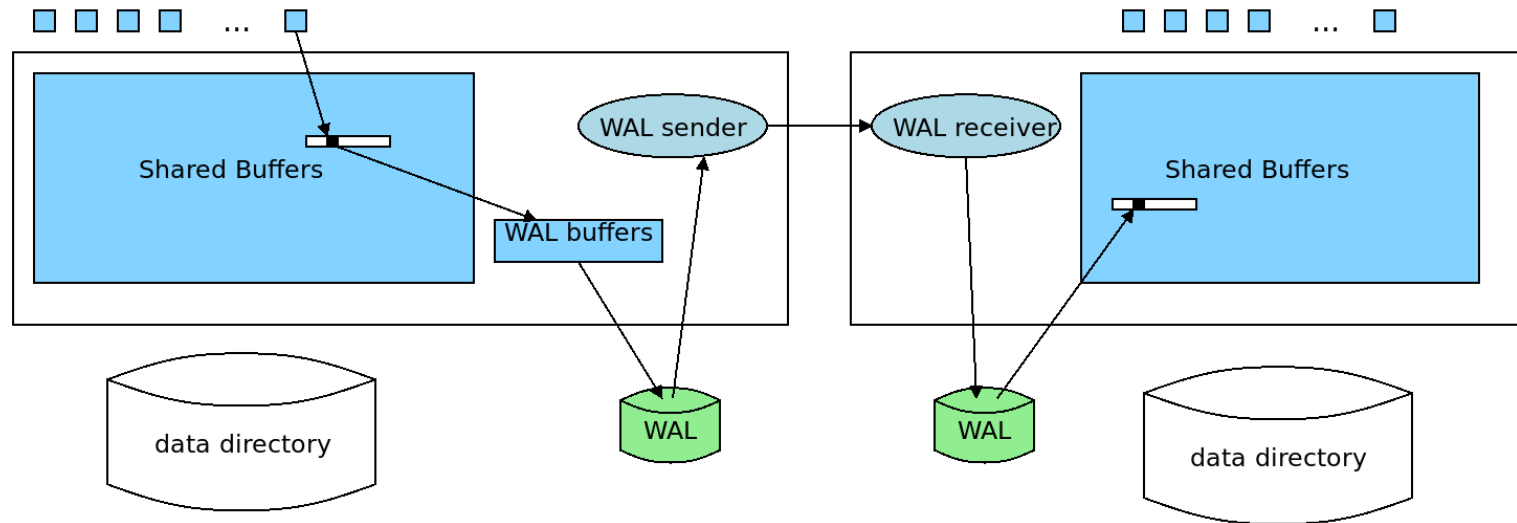
Physical streaming replication



Physical streaming replication



Read replica



Is it master or replica?

On master

```
wacken=# select pg_is_in_recovery();  
pg_is_in_recovery  
-----  
f
```

On replica

```
wacken=# select pg_is_in_recovery();  
pg_is_in_recovery  
-----  
t
```

Can you hear me !?

Can you hear me !?

```
wacken=# select * from pg_stat_replication;
```

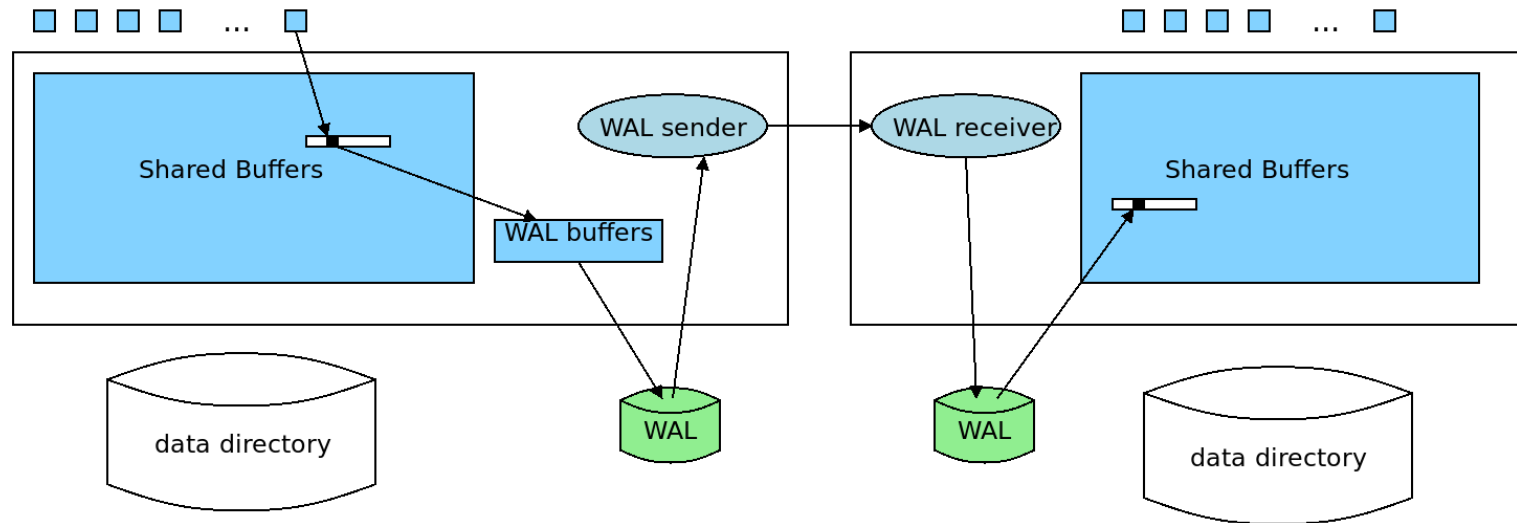
```
-[ RECORD 1 ]-----+-----
```

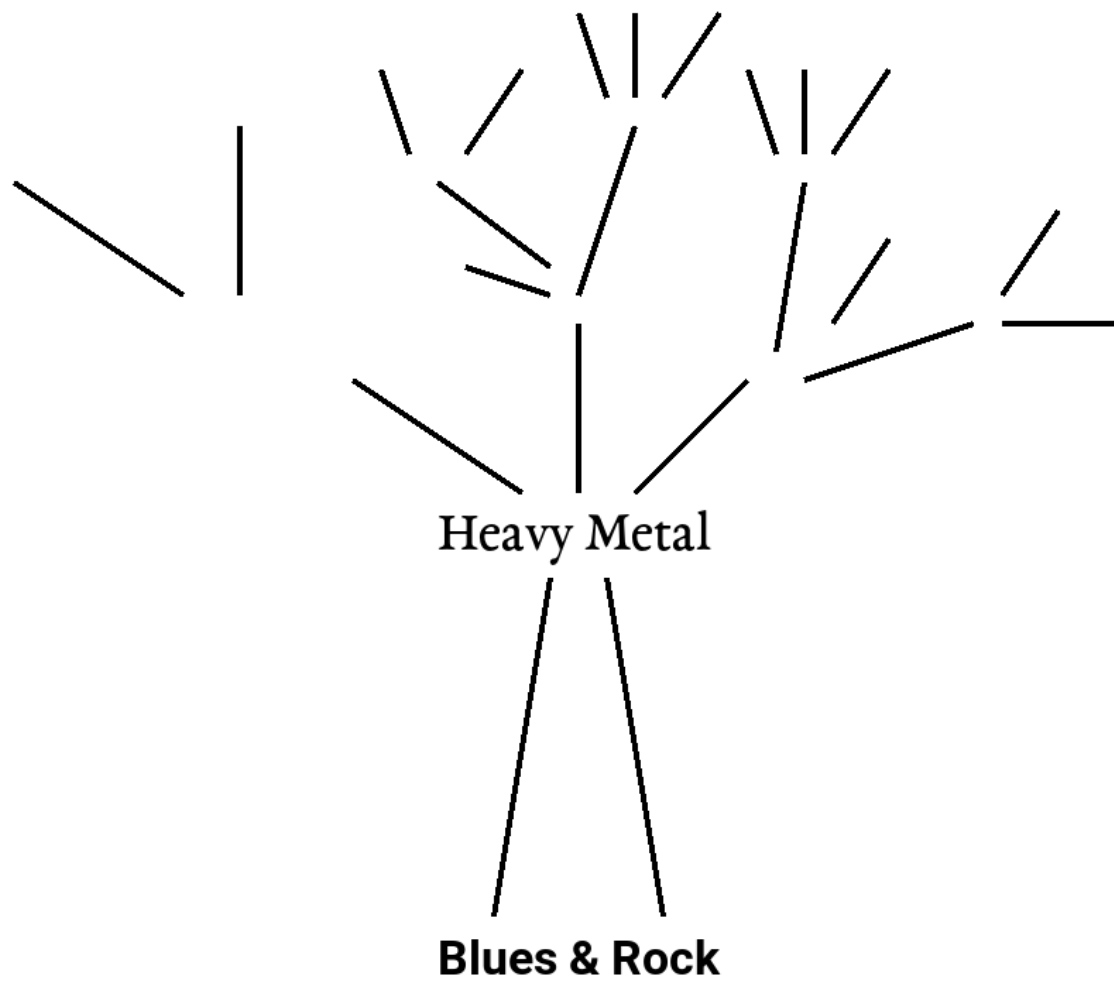
pid	14284
application_name	metal
client_addr	172.17.0.3
state	streaming
sent_lsn	0/3049320
write_lsn	0/3049320
flush_lsn	0/3049320
replay_lsn	0/3049320
write_lag	00:00:00.000628
flush_lag	00:00:00.001741
replay_lag	00:00:00.00224
sync_priority	0
sync_state	async

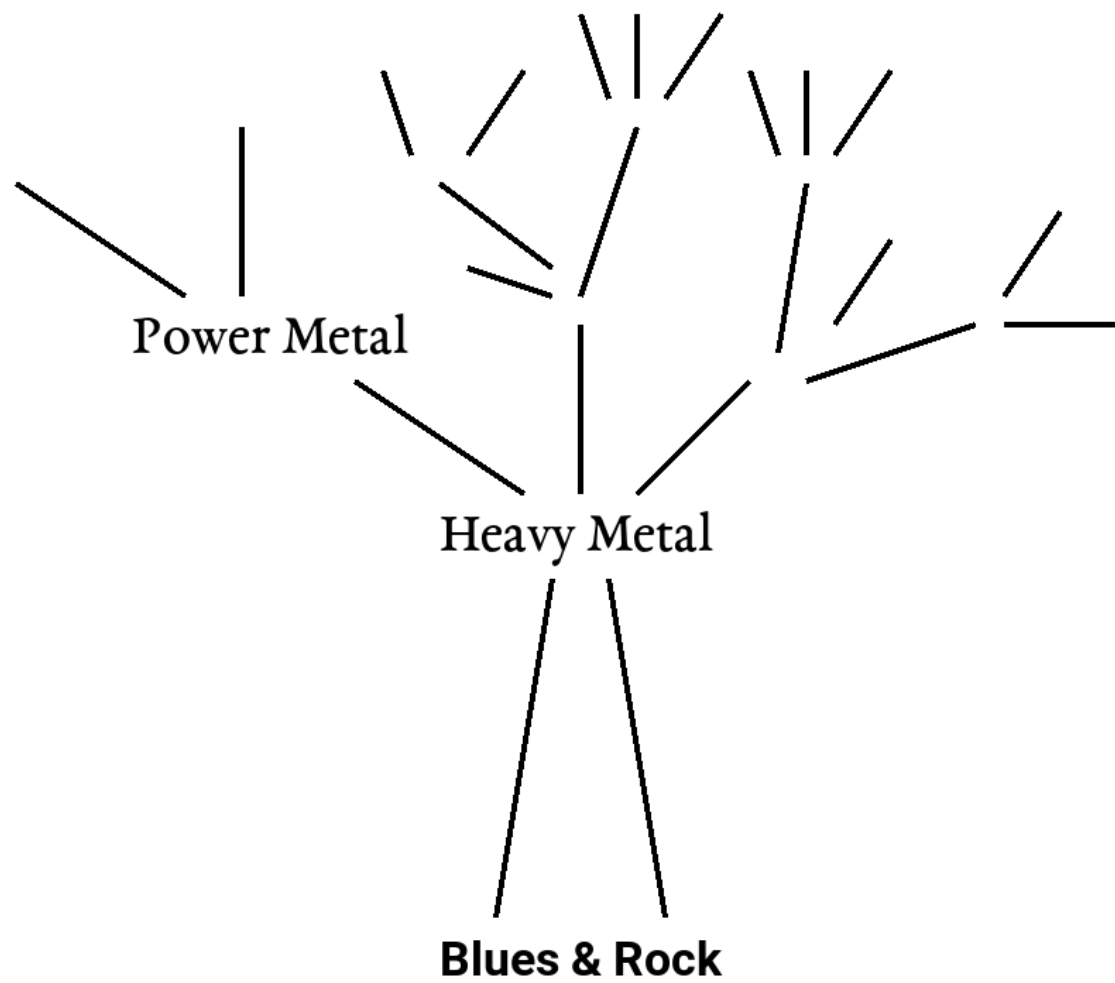
Can you hear me !?

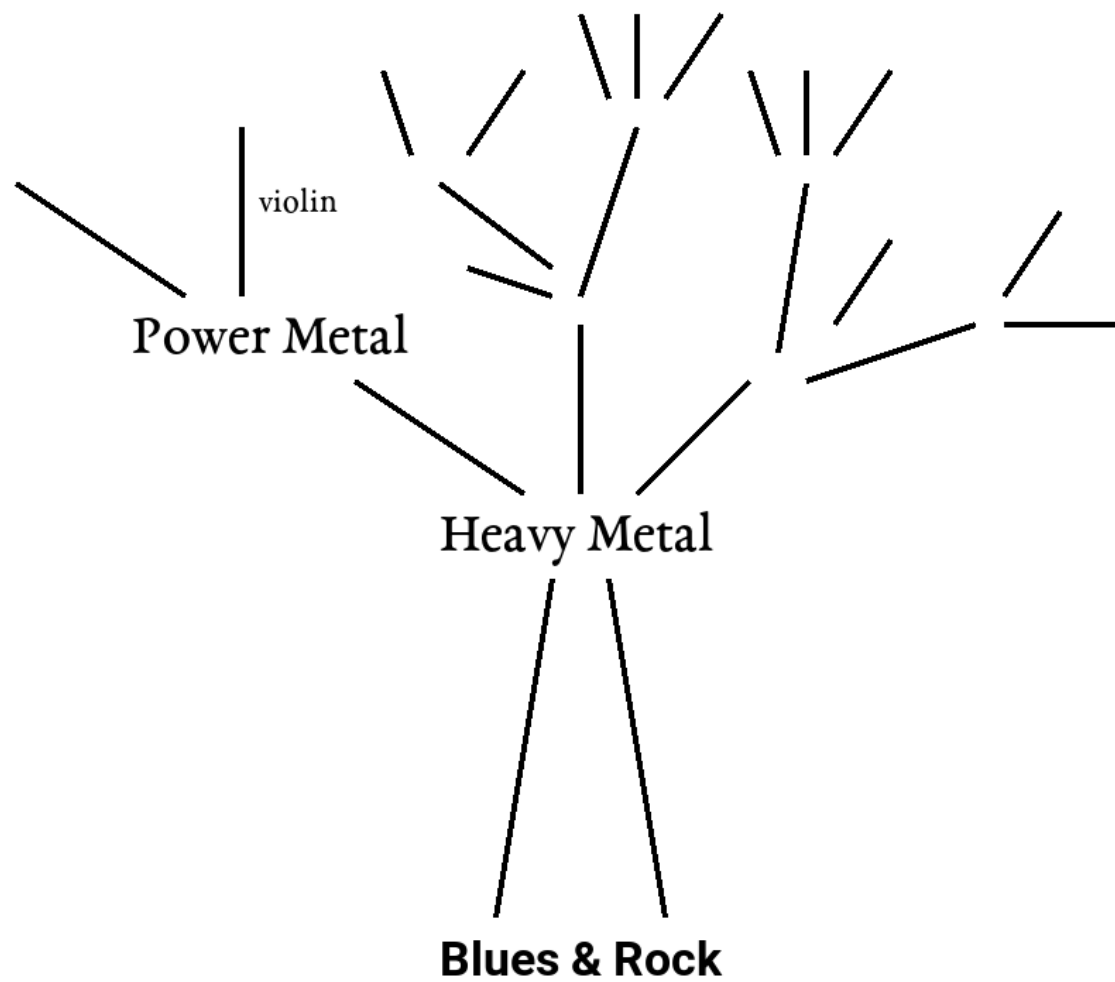
```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid           | 14284
application_name | metal
client_addr    | 172.17.0.3
state         | streaming
sent_lsn       | 0/3049320
write_lsn      | 0/3049320
flush_lsn      | 0/3049320
replay_lsn     | 0/3049320
write_lag      | 00:00:00.000628
flush_lag      | 00:00:00.001741
replay_lag     | 00:00:00.00224
sync_priority  | 0
sync_state     | async
```

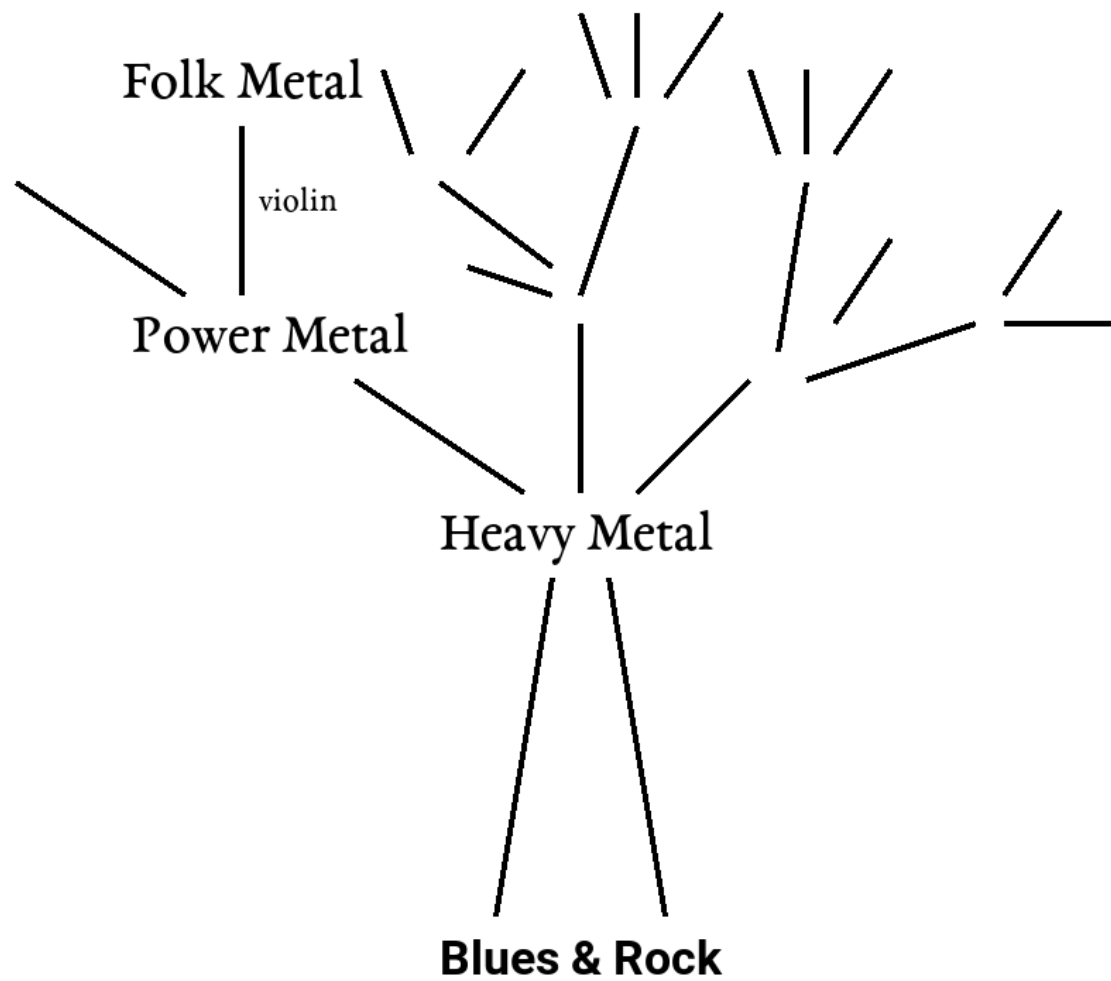
Physical streaming replication

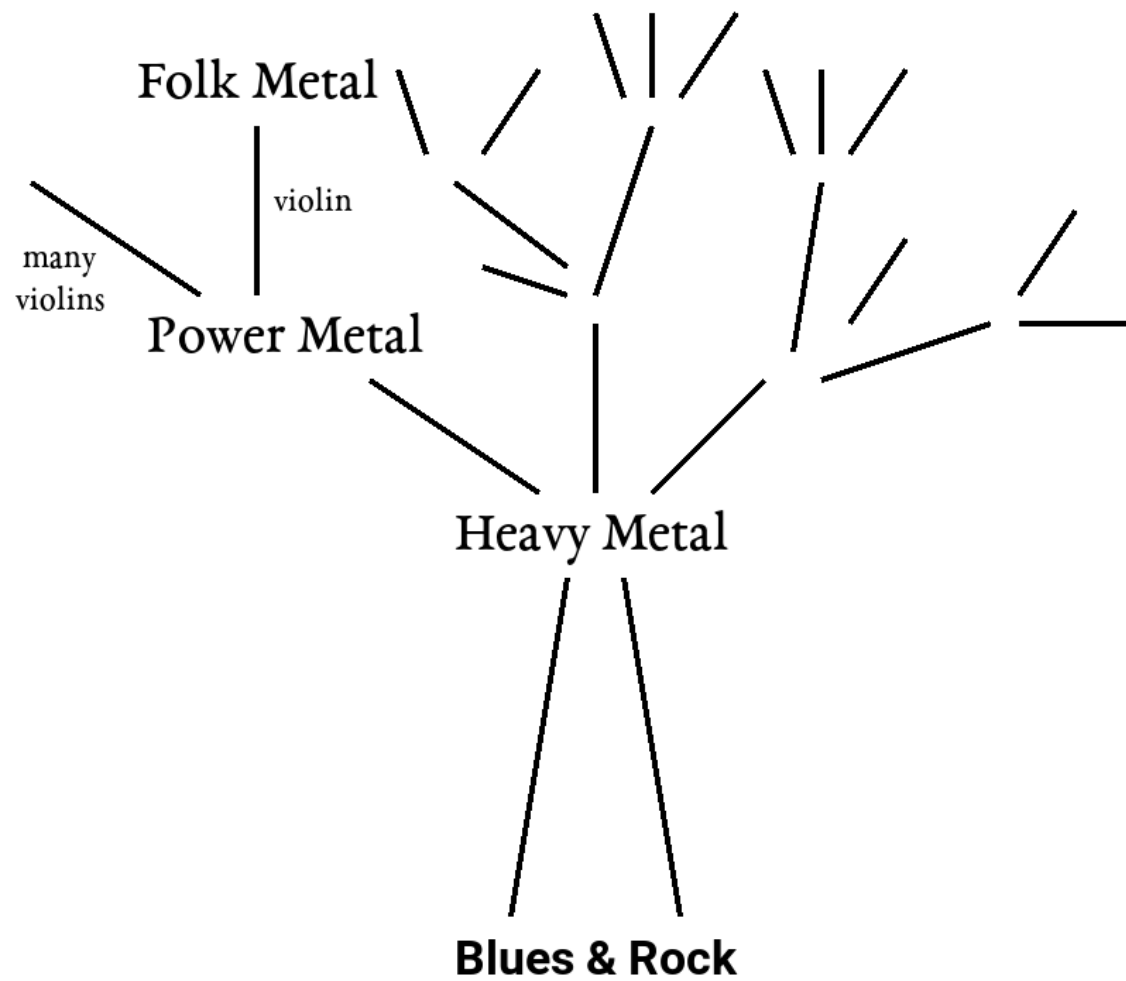


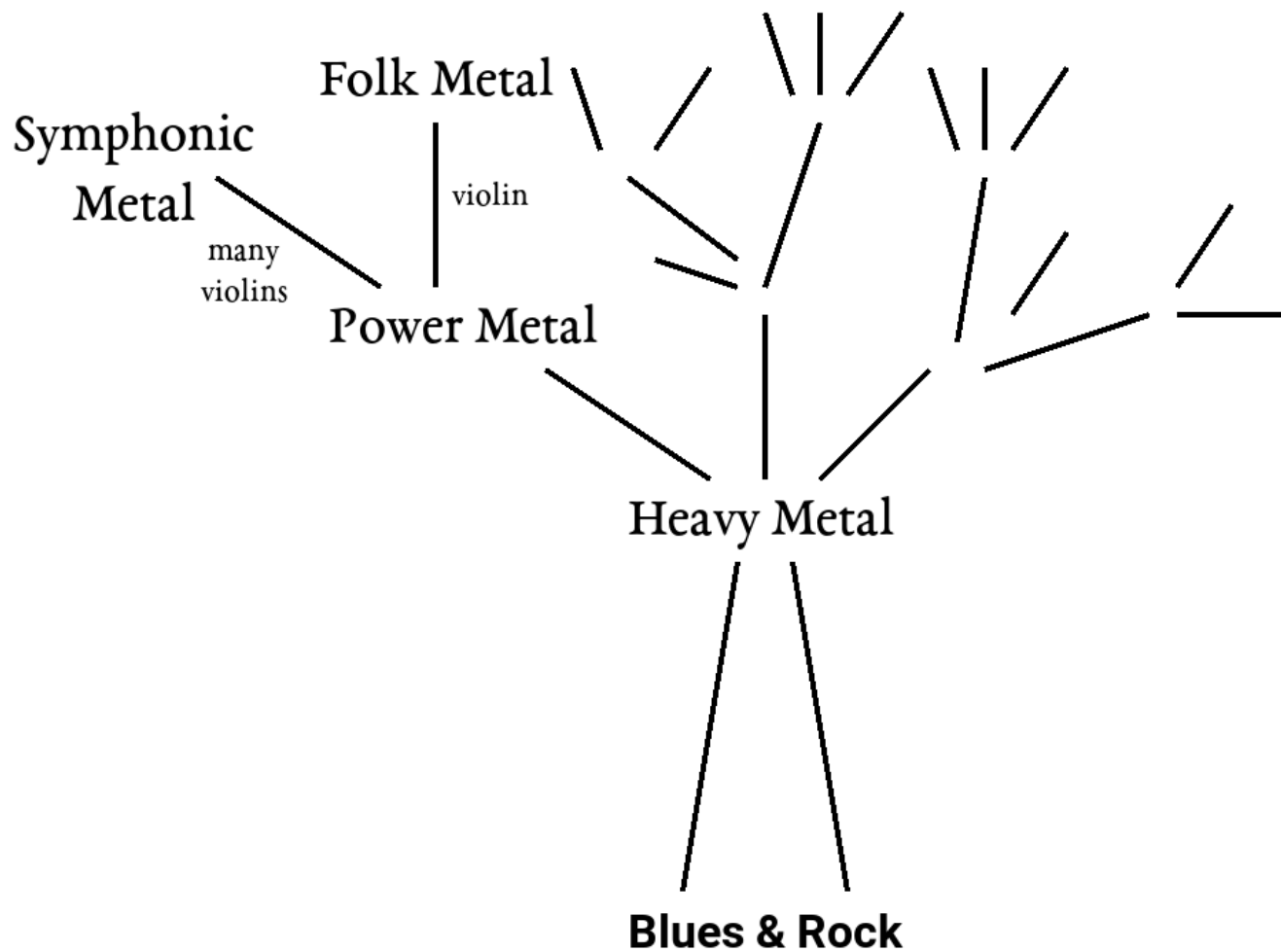


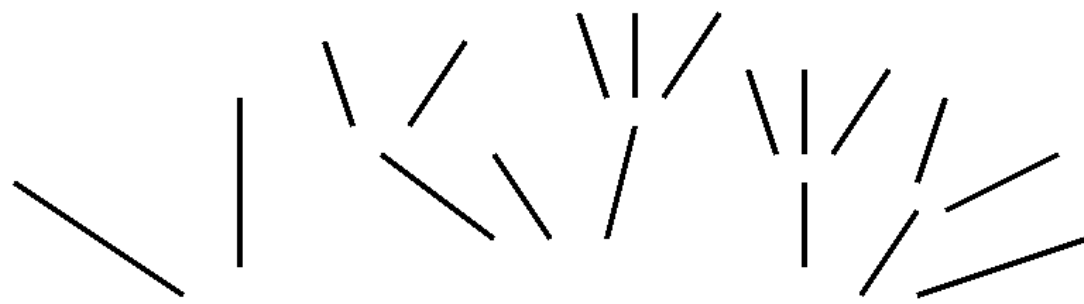






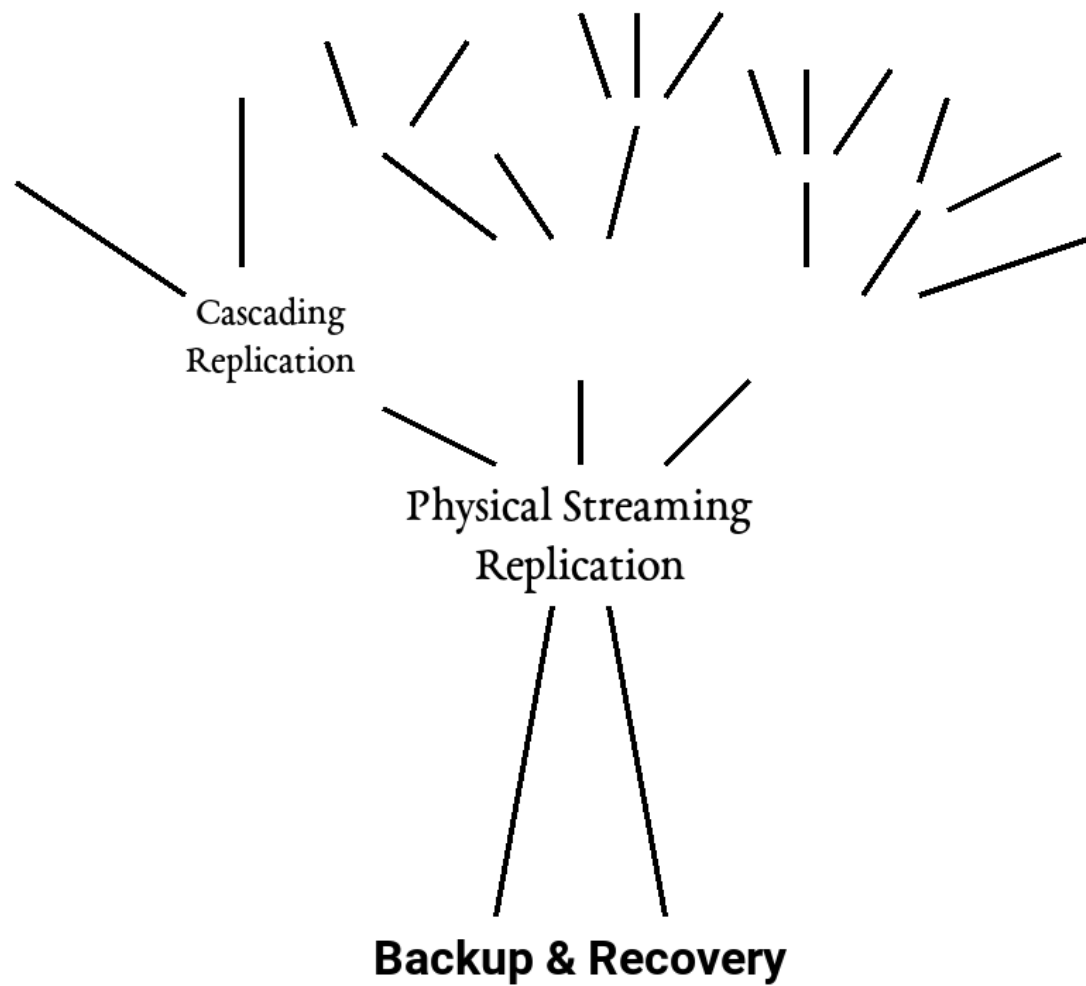


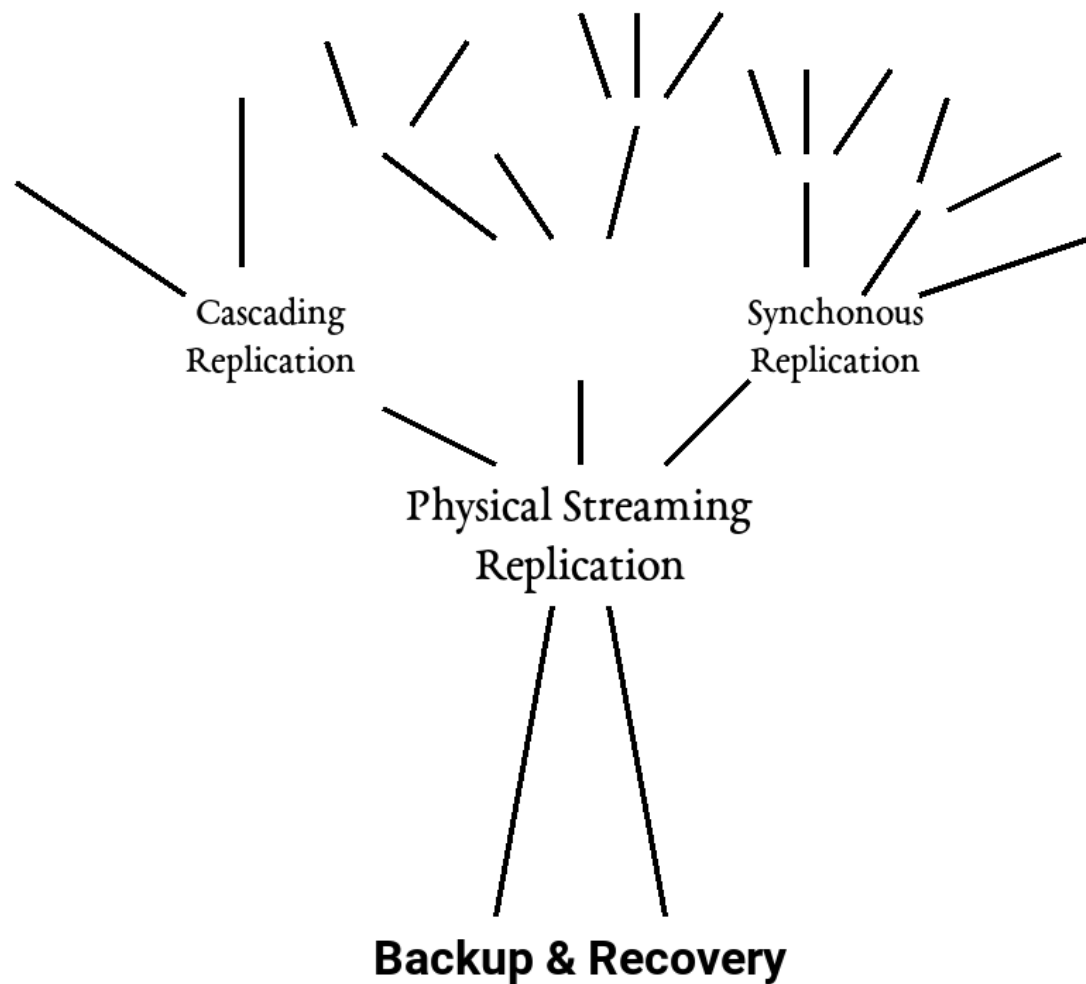


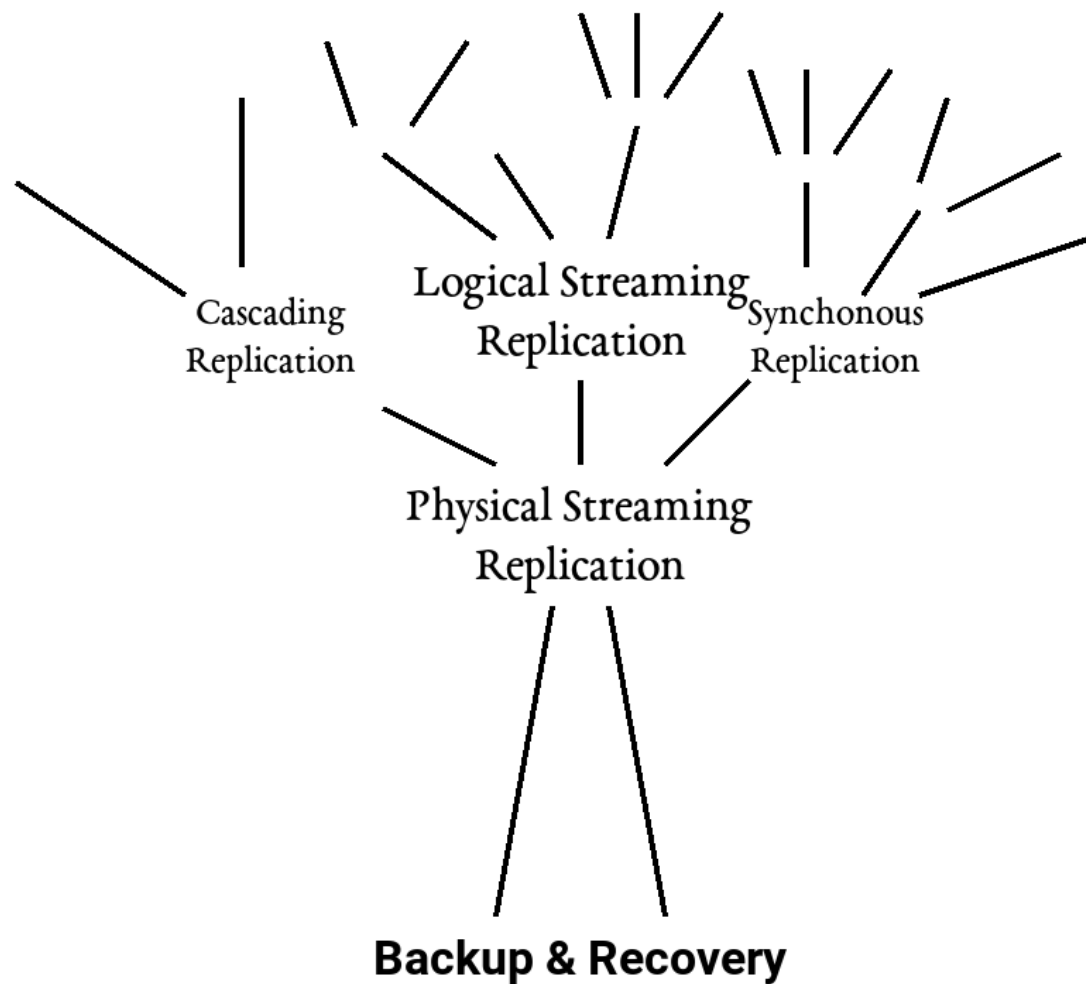


Physical Streaming
Replication

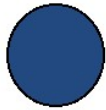
Backup & Recovery



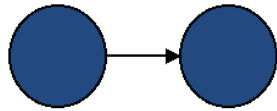




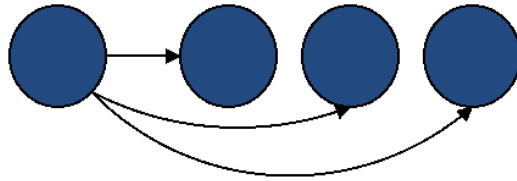
Asynchronous replication



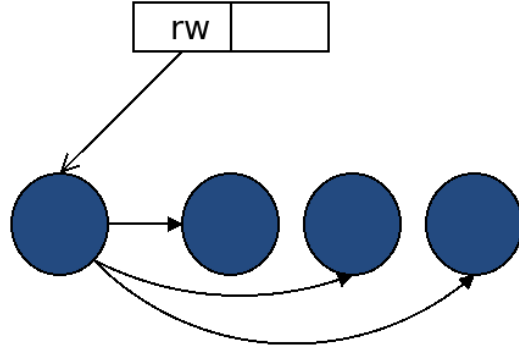
Asynchronous replication



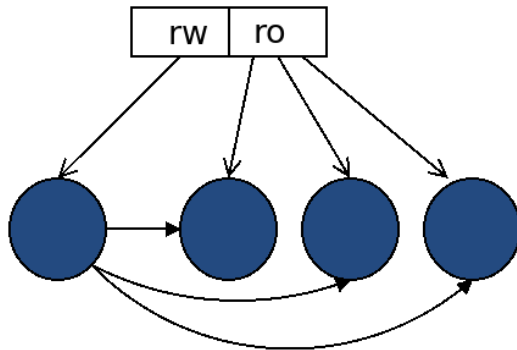
Multiple replicas



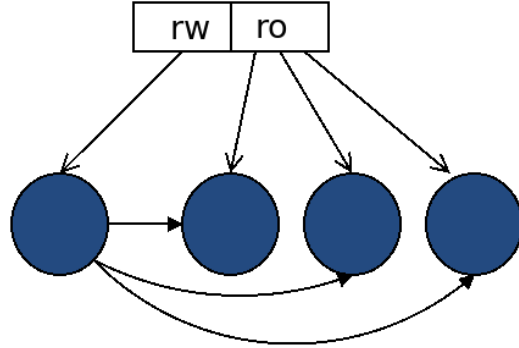
Read-Writes to the master



Read-only to the replicas

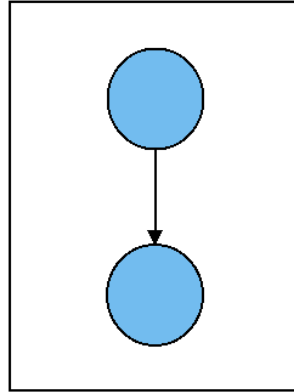


Horizontal read scalability

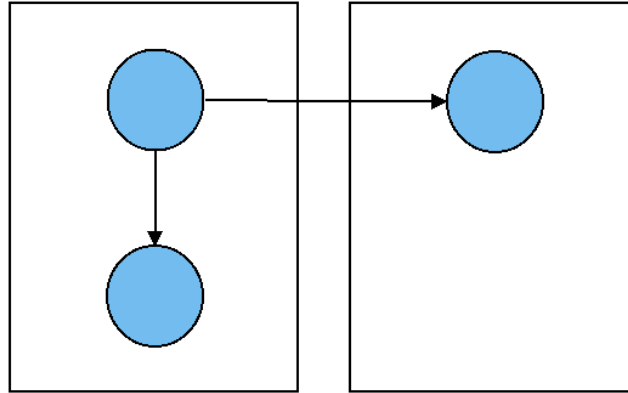


Robustness

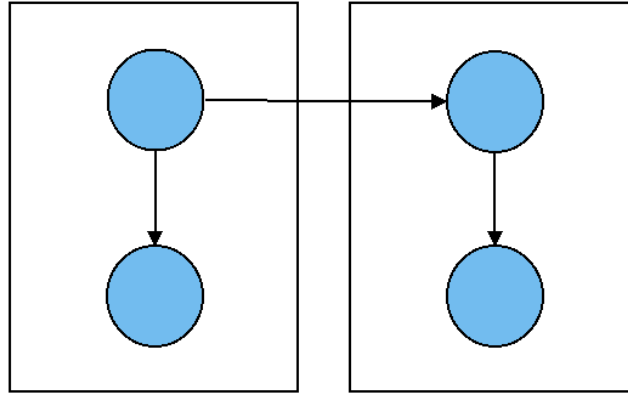
Robustness



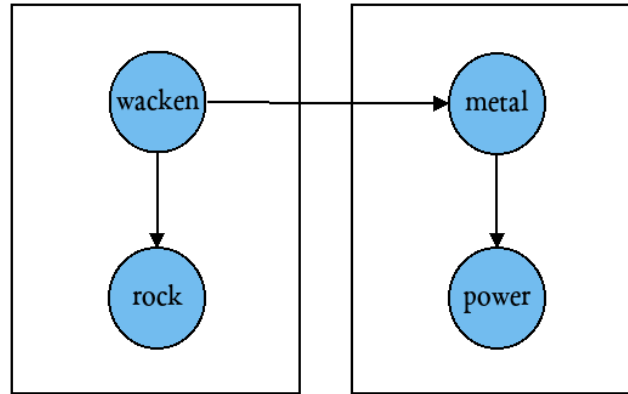
Robustness with two locations



Cascading replication



Cascading replication



Such a fine line between cascading...

wacken is primary

at metal node

```
primary_conninfo='user=repmgr host=wacken port=5432 application_name=metal'
```

at power node

```
primary_conninfo='user=repmgr host=metal port=5432 application_name=rock'
```

Such a fine line between cascading...

wacken is primary

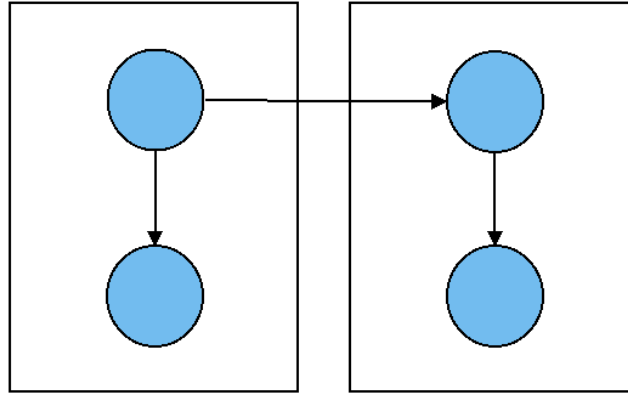
at metal node

```
primary_conninfo='user=repmgr host=wacken port=5432 application_name=metal'
```

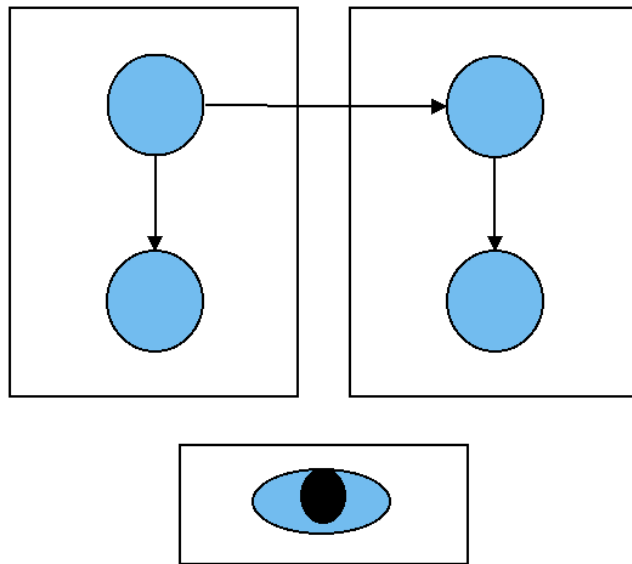
at power node

```
primary_conninfo='user=repmgr host=metal port=5432 application_name=rock'
```

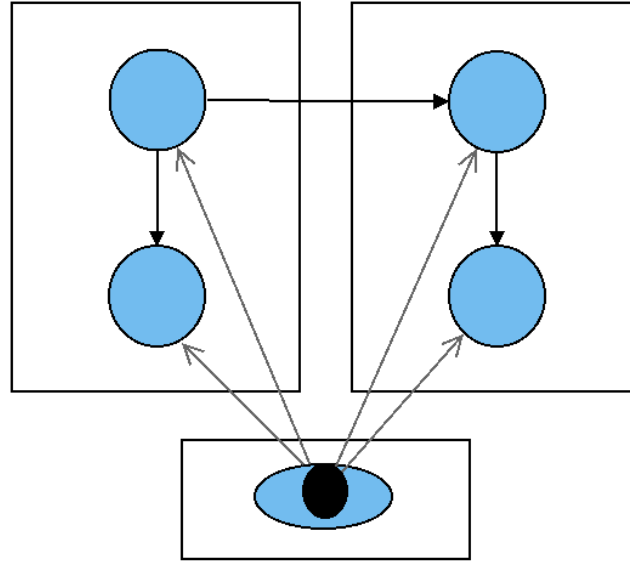
Cascading replication



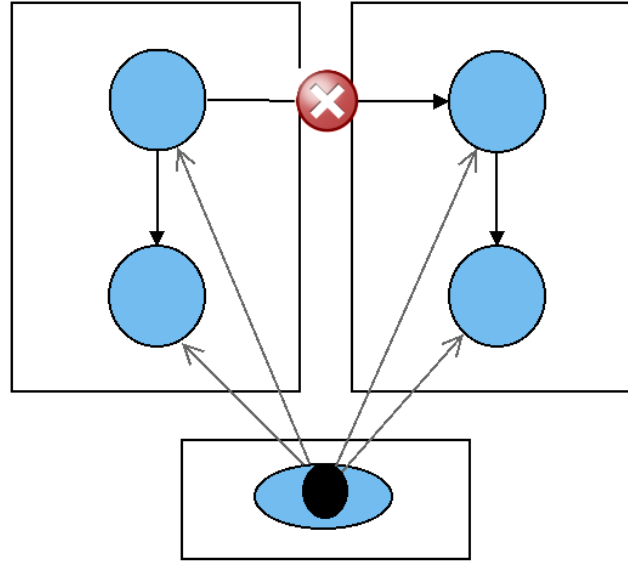
Add a witness



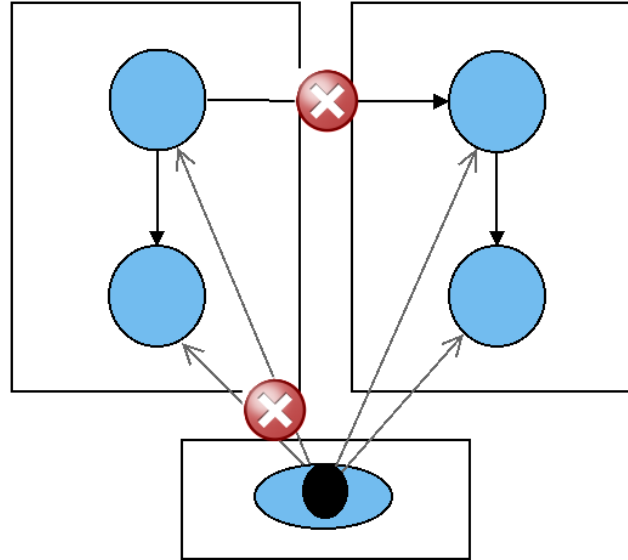
Automated failover



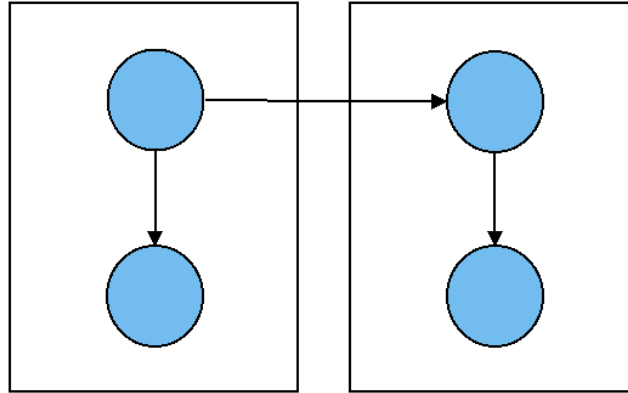
Automated failover – network or node failure



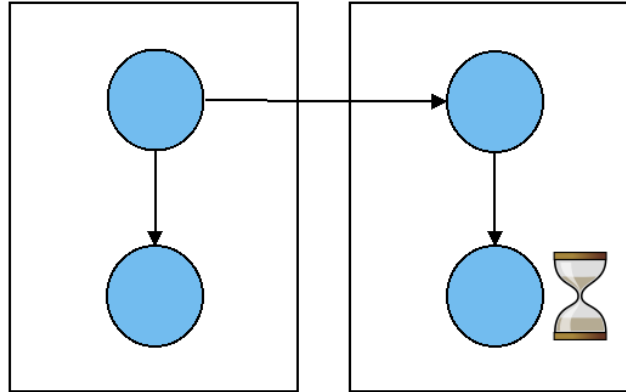
Automated failover – network isolation



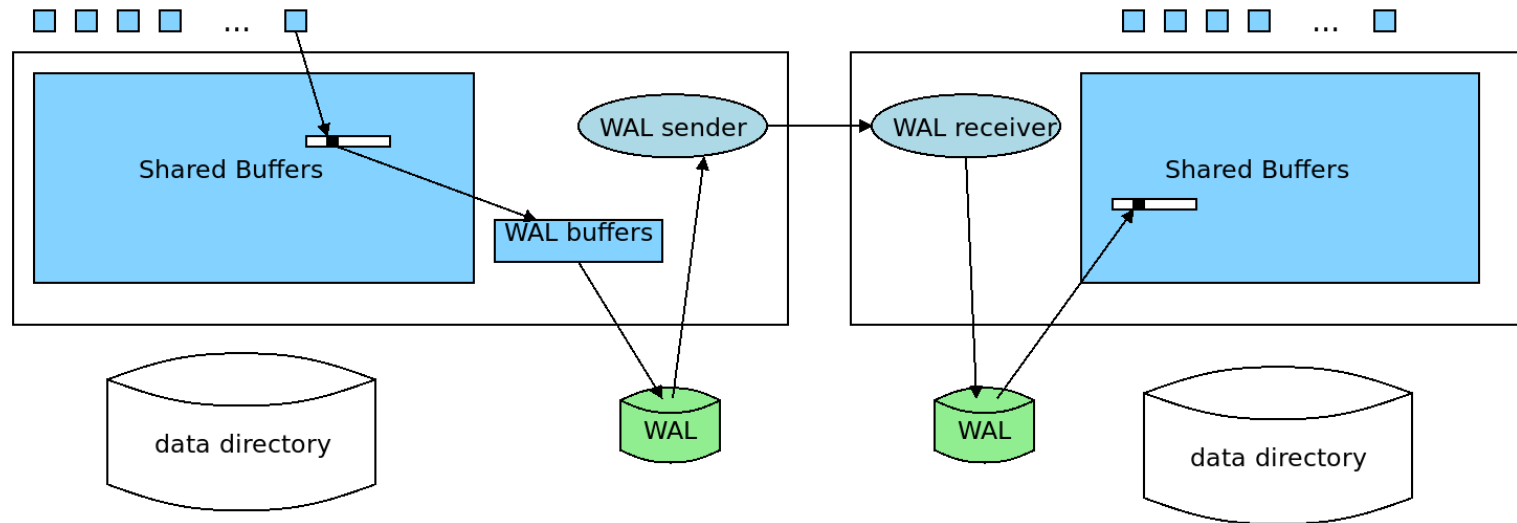
Cascading replication



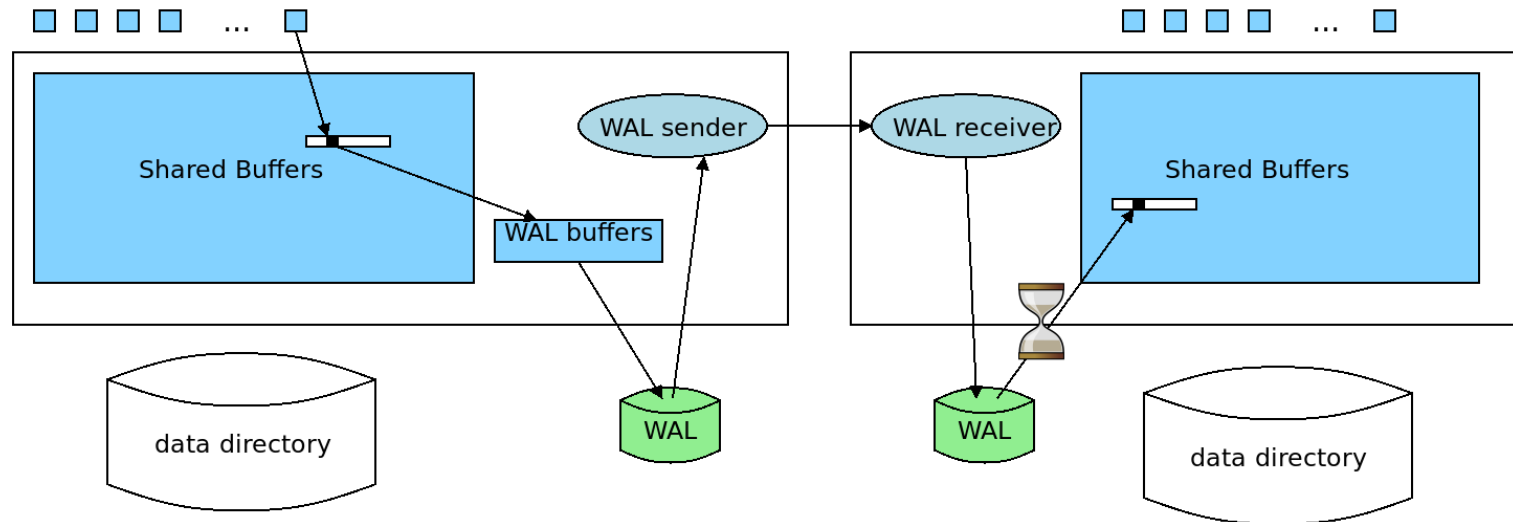
Protect against human error



Physical streaming replication



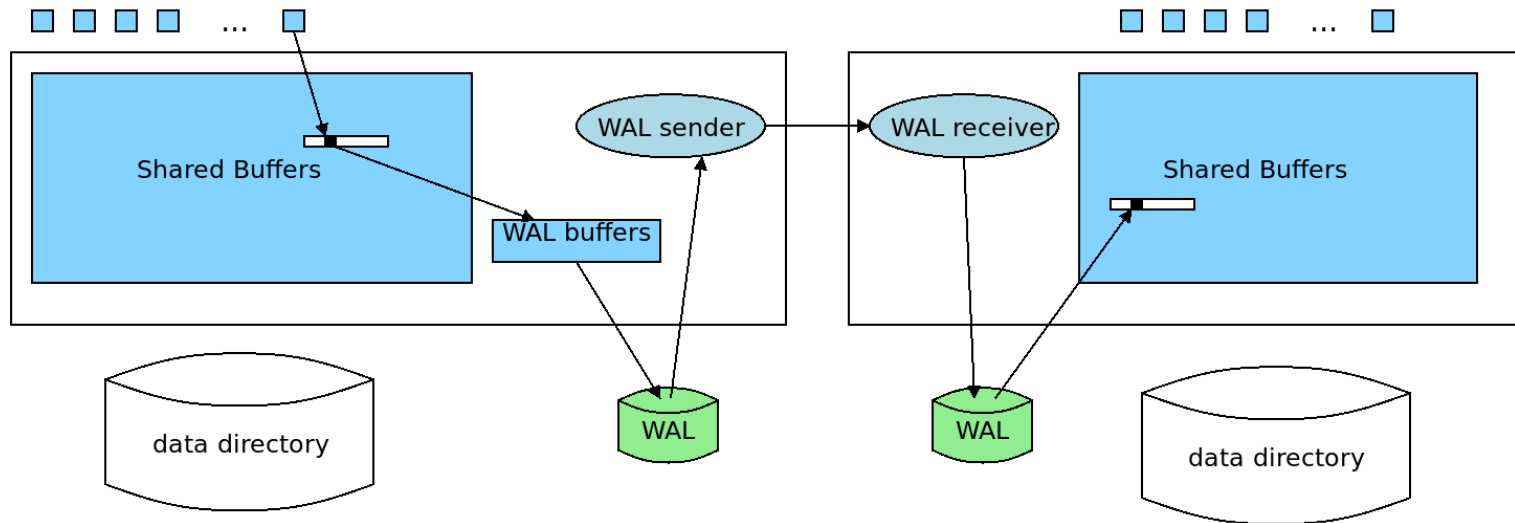
Delayed physical streaming replication



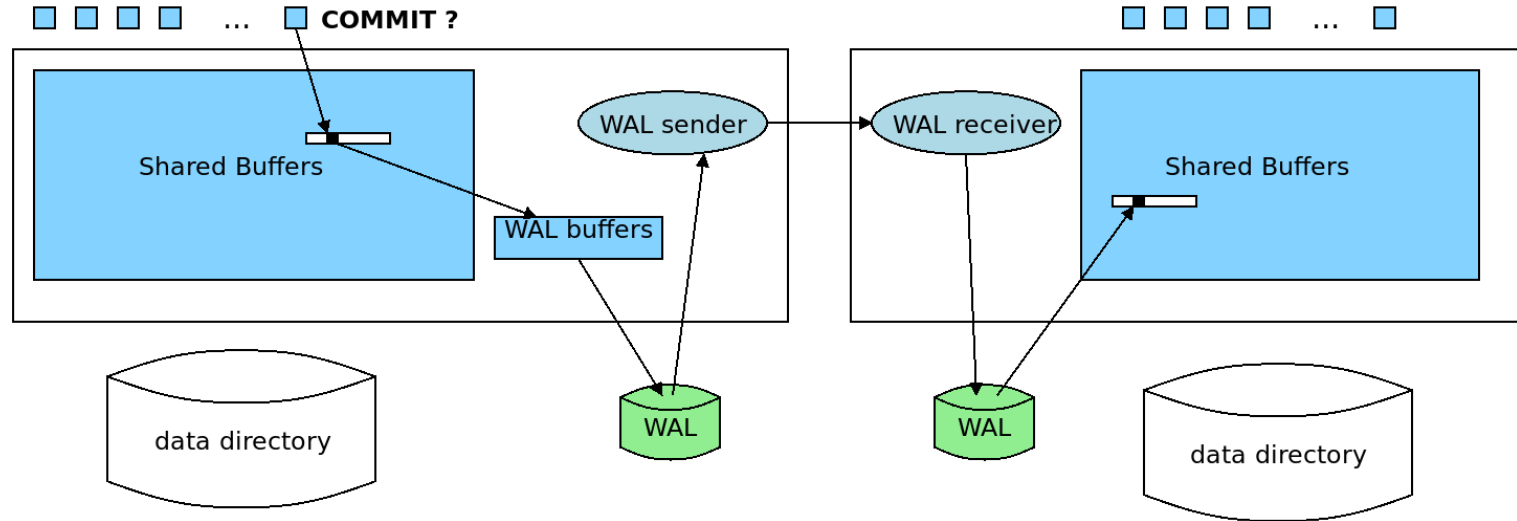
Reduce data loss

Reduce data loss → synchronous replication

Reduce data loss → synchronous replication



Reduce data loss → synchronous replication



Lag really matters now

```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 14284
application_name    | metal
client_addr         | 172.17.0.3
state               | streaming
sent_lsn            | 0/3049320
write_lsn           | 0/3049320
flush_lsn           | 0/3049320
replay_lsn          | 0/3049320
write_lag           | 00:00:00.000628
flush_lag           | 00:00:00.001741
replay_lag          | 00:00:00.00224
sync_priority       | 1
sync_state          | sync
```

Who is replicating

```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 14284
application_name   | metal          synchronous_standby_names
client_addr        | 172.17.0.3
state              | streaming
sent_lsn           | 0/3049320
write_lsn          | 0/3049320
flush_lsn          | 0/3049320
replay_lsn         | 0/3049320
write_lag          | 00:00:00.000628
flush_lag          | 00:00:00.001741
replay_lag         | 00:00:00.00224
sync_priority      | 1
sync_state         | sync
```

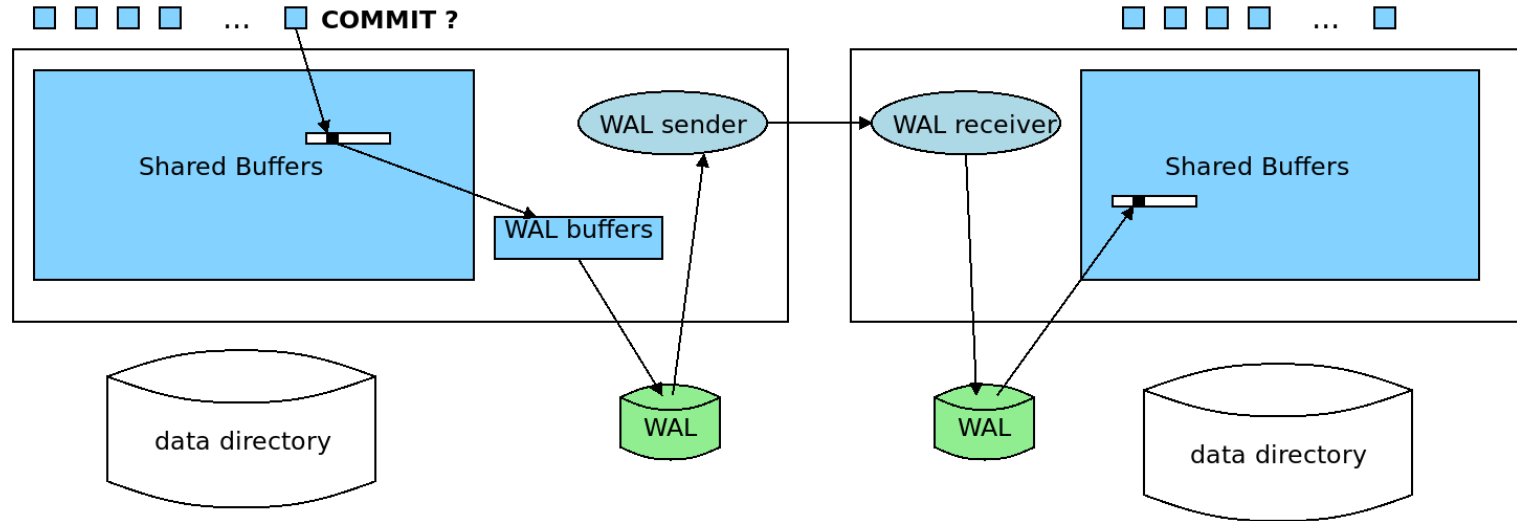
Who is replicating?

```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 14284
application_name   | metal                synchronous_standby_names
client_addr        | 172.17.0.3
state              | streaming
sent_lsn           | 0/3049320
write_lsn          | 0/3049320
flush_lsn          | 0/3049320
replay_lsn         | 0/3049320
write_lag          | 00:00:00.000628
flush_lag          | 00:00:00.001741
replay_lag         | 00:00:00.00224
sync_priority      | 1
sync_state         | sync
```

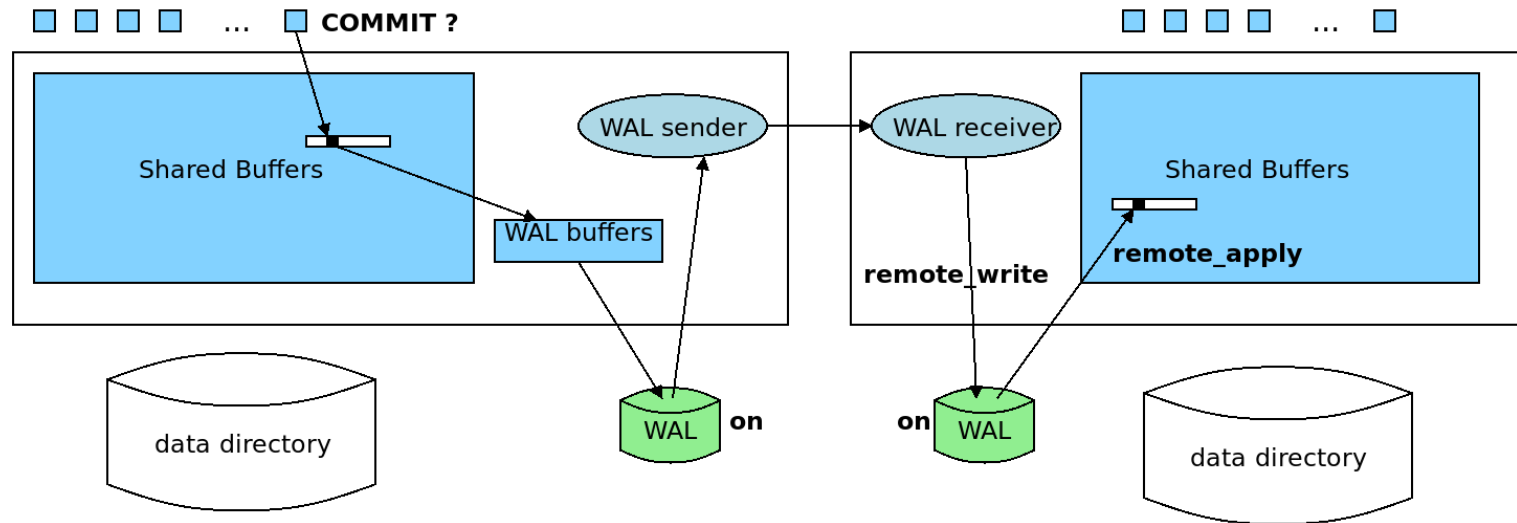
When is synchronously replicated?

```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 14284
application_name   | metal          synchronous_standby_names
client_addr        | 172.17.0.3
state              | streaming
sent_lsn           | 0/3049320
write_lsn          | 0/3049320      remote_write
flush_lsn          | 0/3049320      on
replay_lsn         | 0/3049320      remote_apply
write_lag          | 00:00:00.000628
flush_lag          | 00:00:00.001741
replay_lag         | 00:00:00.00224
sync_priority      | 1
sync_state         | sync
```

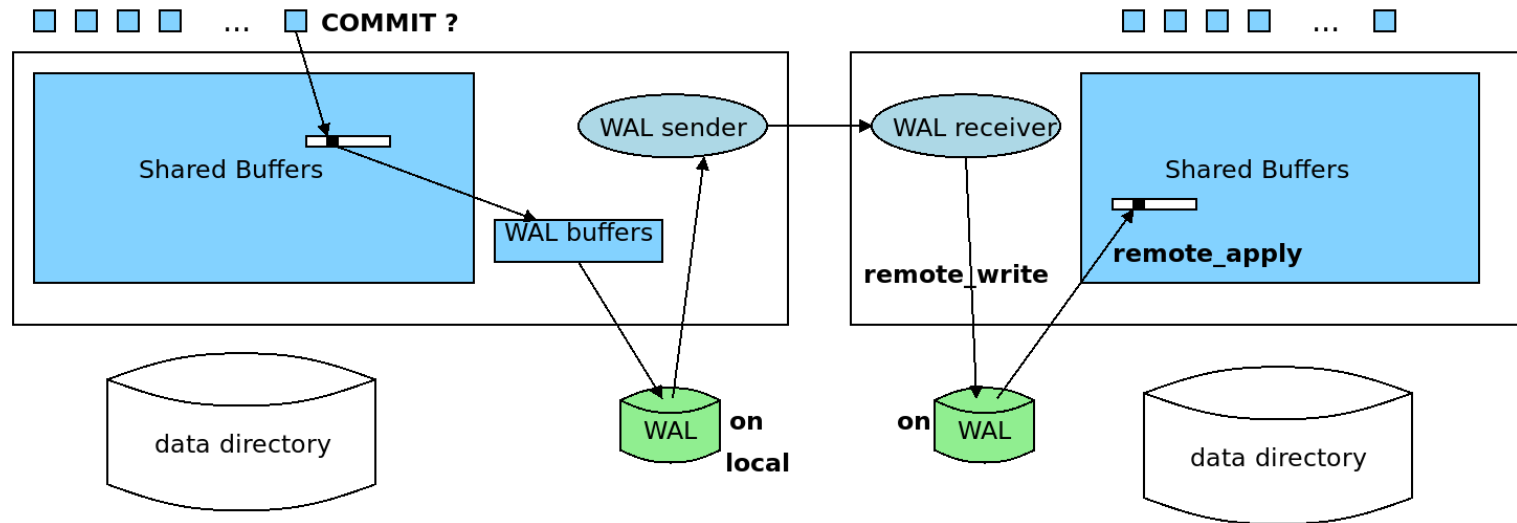
Reduce data loss → synchronous replication



Reduce data loss → synchronous replication



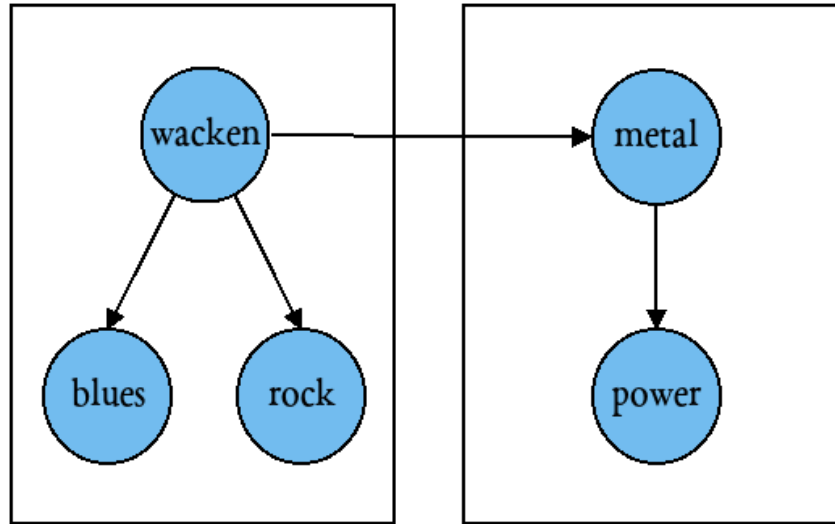
Reduce data loss → synchronous replication



When is synchronously replicated?

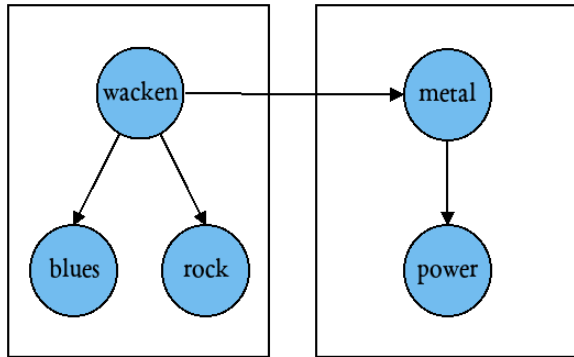
```
wacken=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 14284
application_name   | metal          synchronous_standby_names
client_addr        | 172.17.0.3
state              | streaming
sent_lsn           | 0/3049320
write_lsn           | 0/3049320      remote_write
flush_lsn          | 0/3049320      on
replay_lsn         | 0/3049320      remote_apply
write_lag          | 00:00:00.000628
flush_lag          | 00:00:00.001741
replay_lag         | 00:00:00.00224
sync_priority       | 1
sync_state         | sync
```

Synchronous standby names



Synchronous standby names

```
synchronous_standby_names = ...
```



```
'FIRST 1 (metal, rock, blues)'
```

```
'FIRST 2 (rock, blues, metal)'
```

```
'ANY 2 (rock, blues, metal)'
```

Different places to be synchronous

```
synchronous_commit = ...
```

```
remote_write
```

```
on
```

```
remote_apply
```

Different places to be synchronous

```
synchronous_commit = ...
```

```
remote_write
```

```
on
```

```
remote_apply
```

```
local
```

Synchronous when it's important

Synchronous when it's important

```
synchronous_standby_names = 'ANY 1 (metal, rock, blues)'  
synchronous_commit = local
```


Synchronous when it's important

```
synchronous_standby_names = 'ANY 1 (metal, rock, blues)'  
synchronous_commit = local
```

```
wacken=# SET synchronous_commit = remote_apply;  
SET
```

Synchronous when it's important

```
synchronous_standby_names = 'ANY 1 (metal, rock, blues)'  
synchronous_commit = local
```

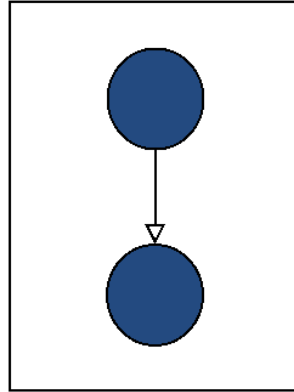
```
wacken=# SET synchronous_commit = remote_apply;  
SET  
wacken=# INSERT INTO orders (id, customer, item, quantity)  
VALUES (2267709, 666, 'beer', 18);  
INSERT 0 1
```

Synchronous when it's important

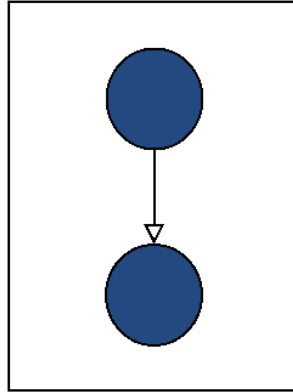
```
synchronous_standby_names = 'ANY 1 (metal, rock, blues)'  
synchronous_commit = local
```

```
wacken=# SET synchronous_commit = remote_apply;  
SET  
wacken=# INSERT INTO orders (id, customer, item, quantity)  
VALUES (2267709, 666, 'beer', 18);  
INSERT 0 1  
wacken=# RESET synchronous_commit ;  
RESET
```

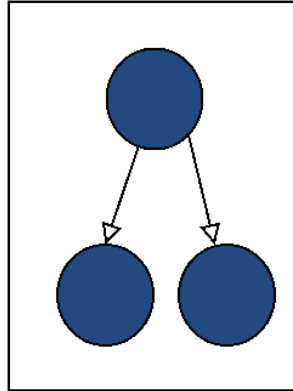
Synchronous Replication



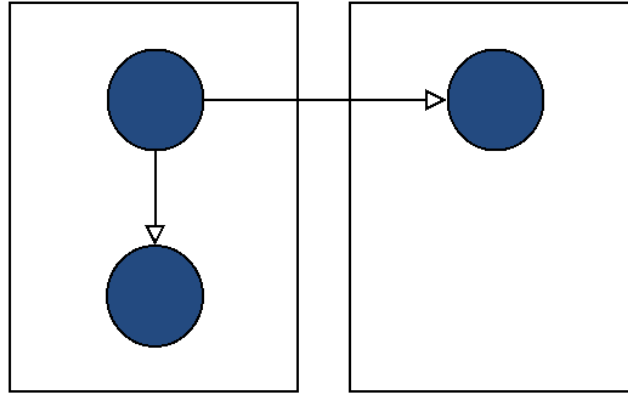
Synchronous Replication - Fragile



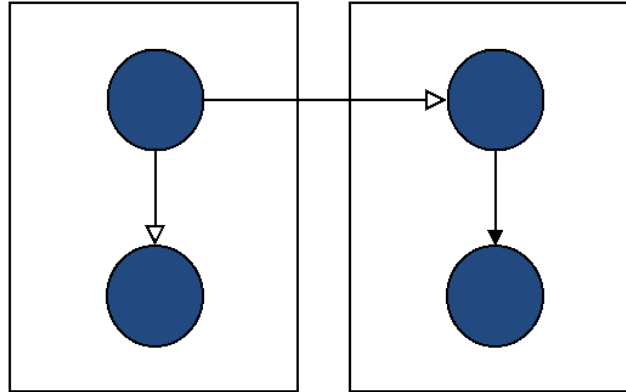
Synchronous Replication - OK



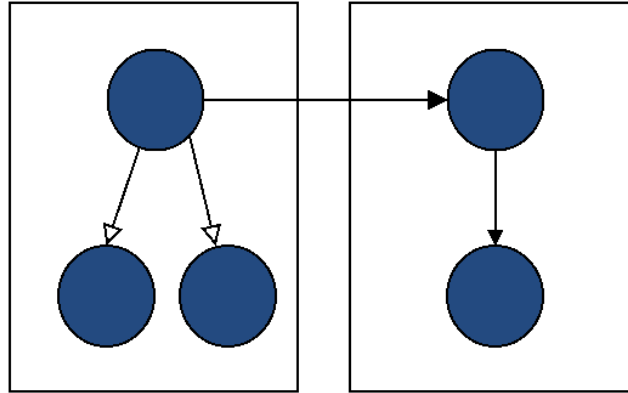
Synchronous Replication – 2 DC



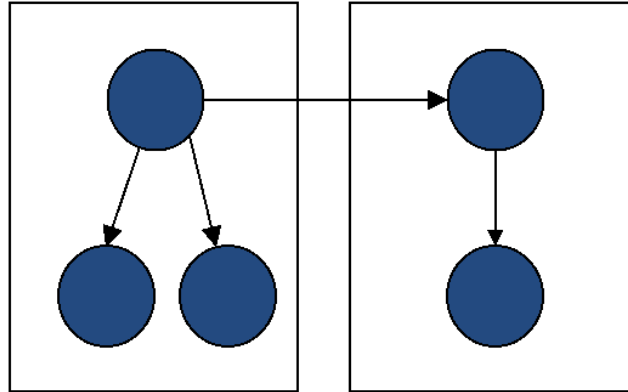
Synchronous Replication – 2 DC



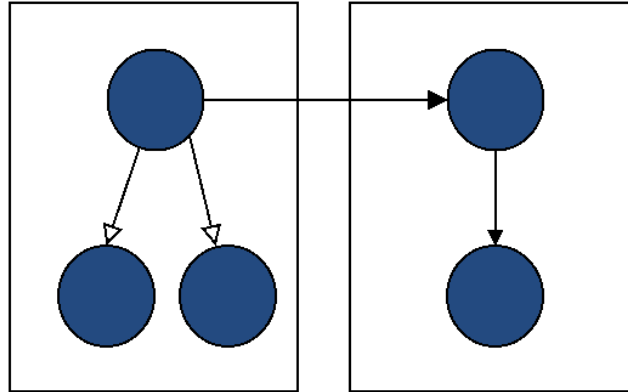
Synchronous Replication - Robust



Dynamic asynchronous / synchronous

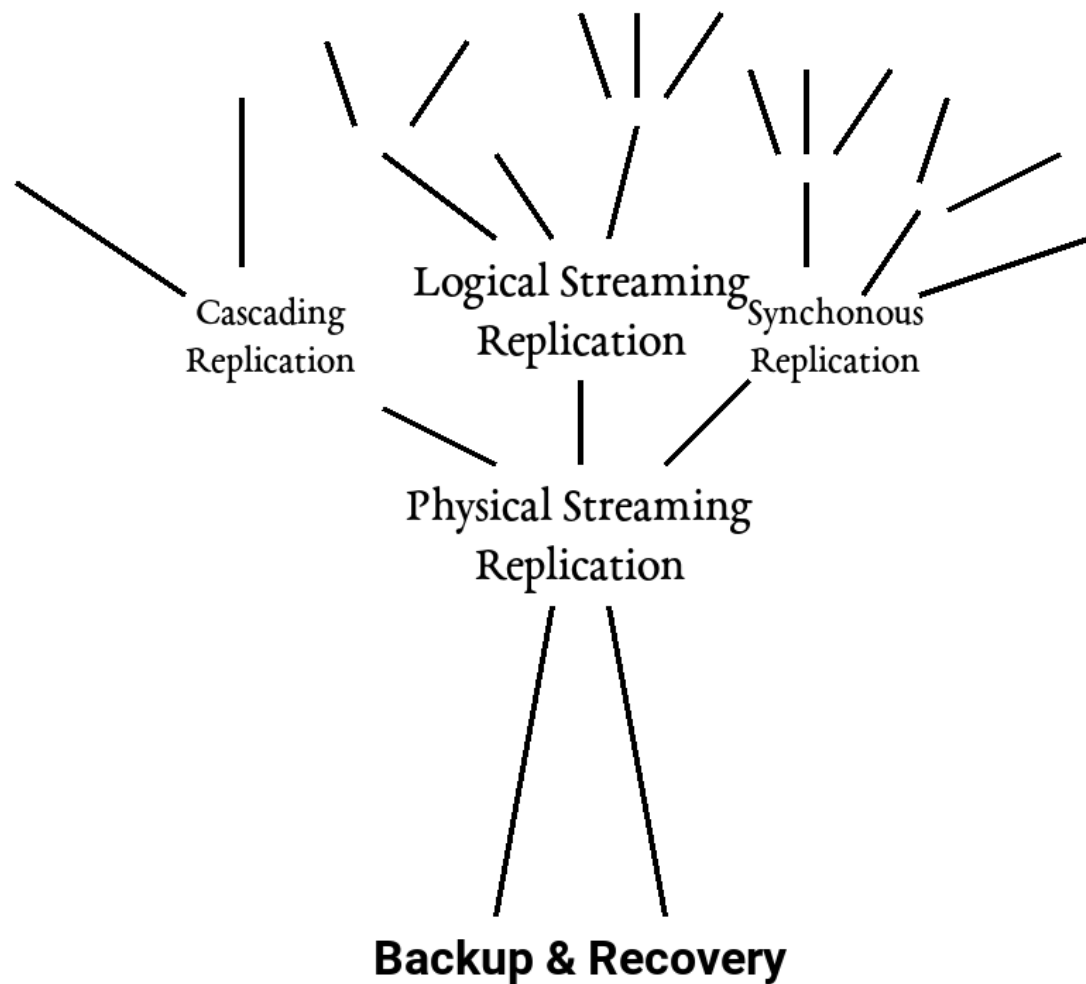


Dynamic asynchronous / synchronous

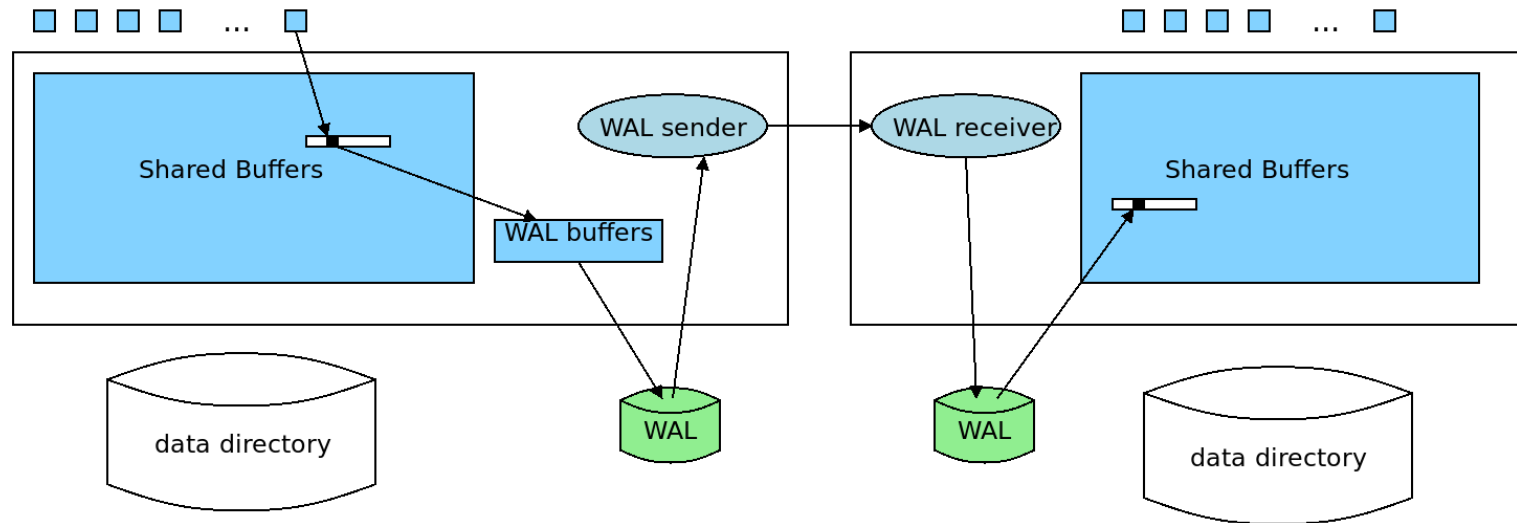


Intermezzo

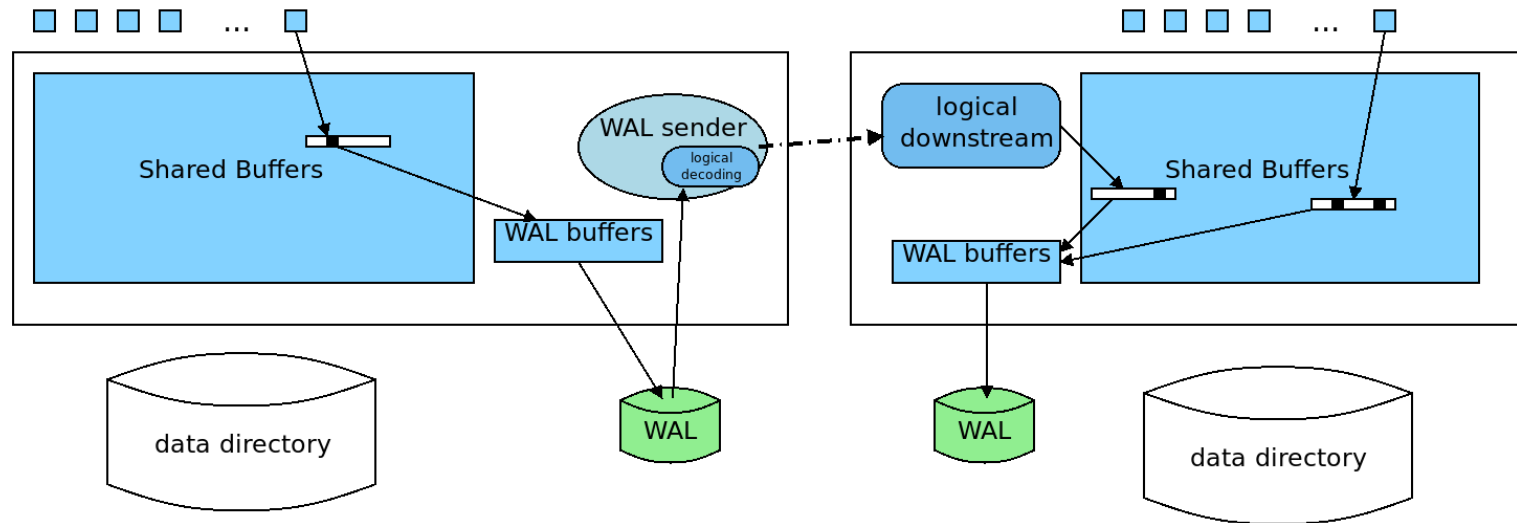




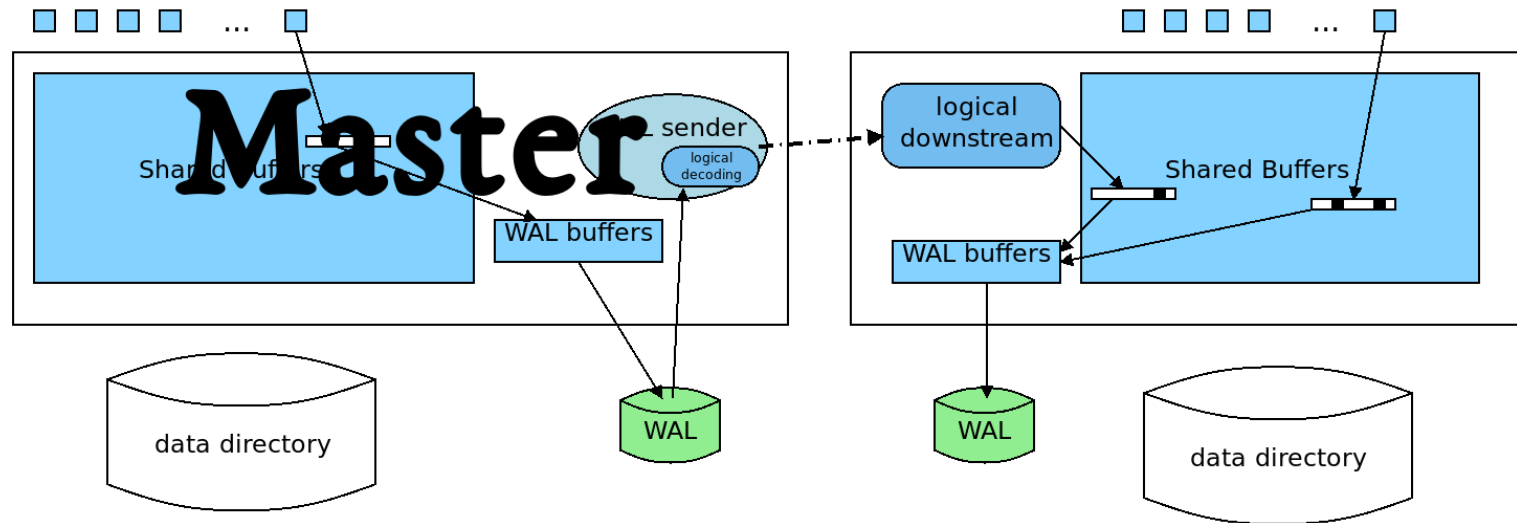
Physical streaming replication



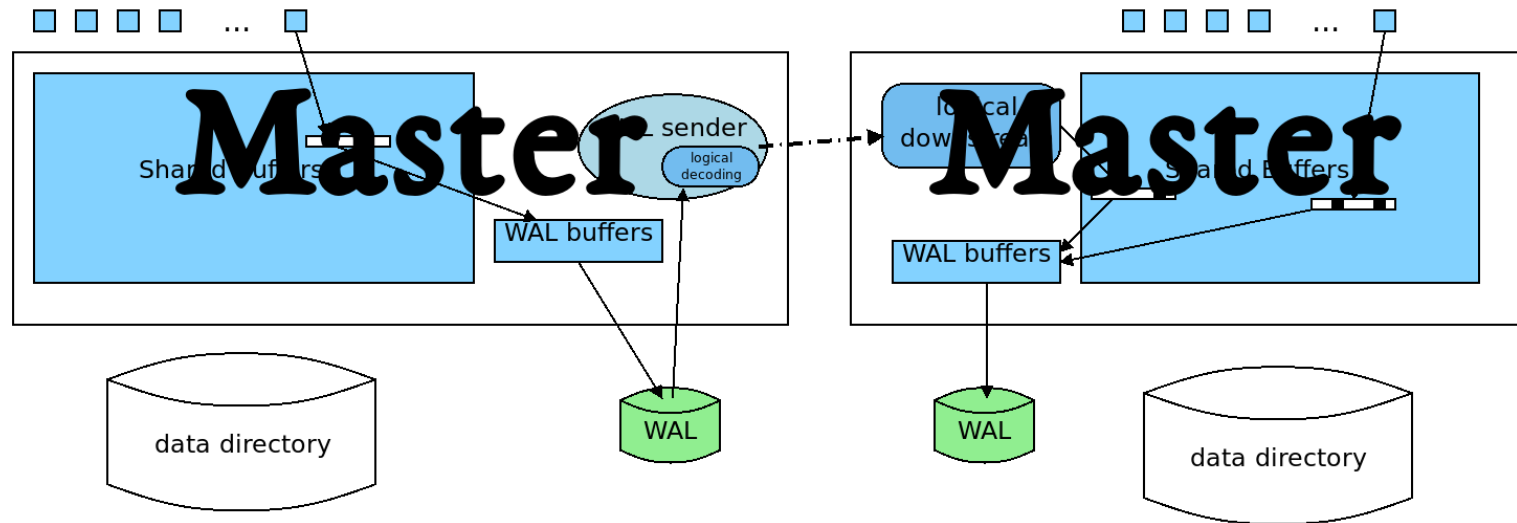
Logical streaming replication



Logical streaming replication



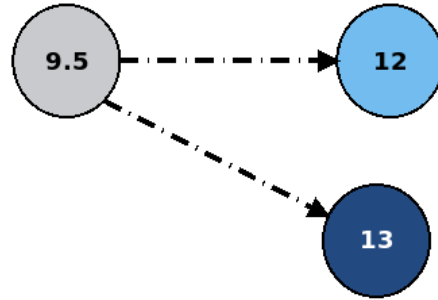
Logical streaming replication



Upgrade major version

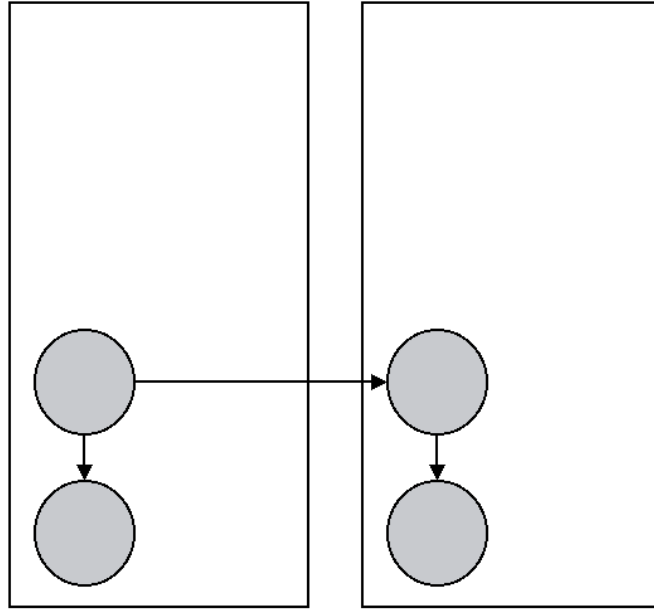


Upgrade major version and split

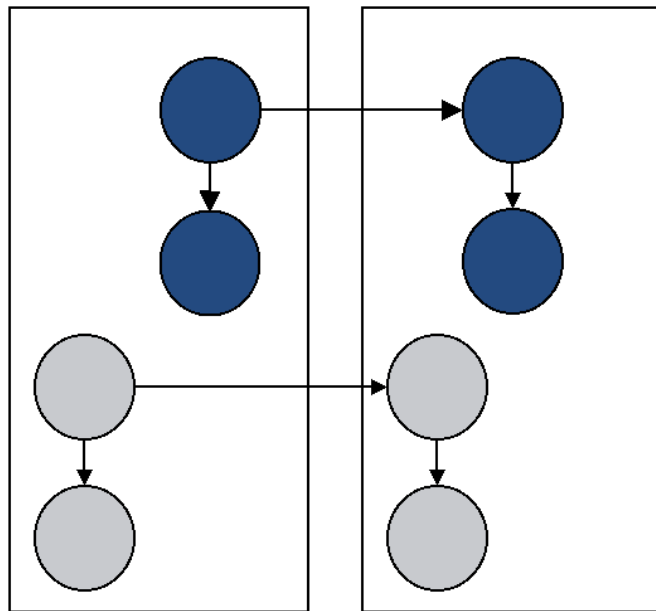


Upgrade full cluster

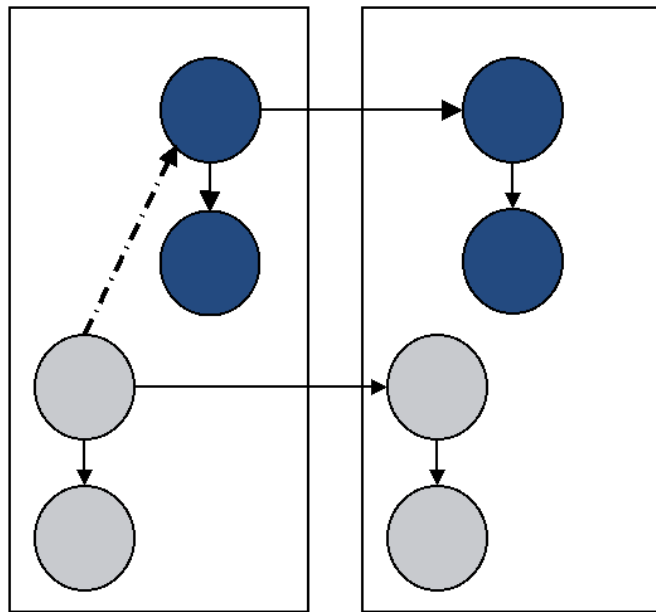
Upgrade full cluster



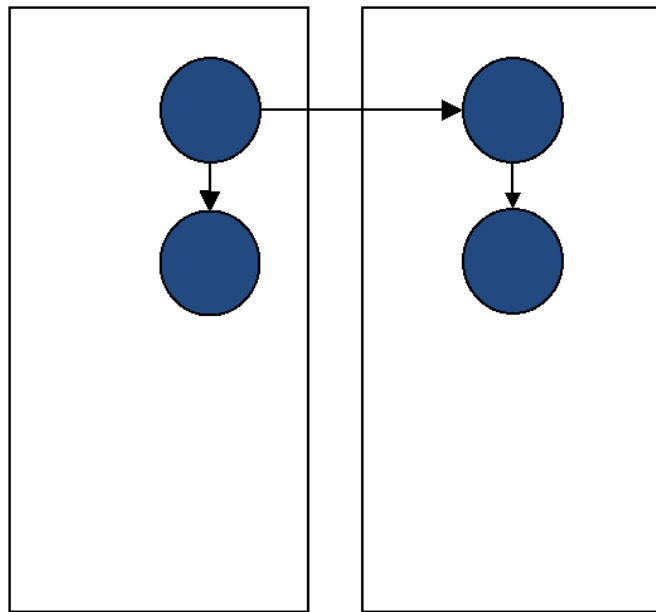
Upgrade full cluster



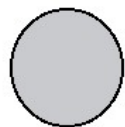
Upgrade full cluster

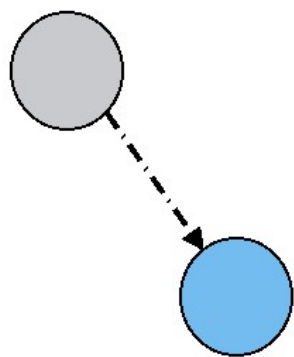


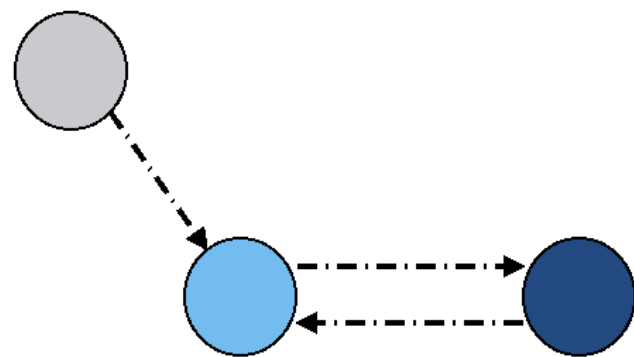
Upgrade full cluster

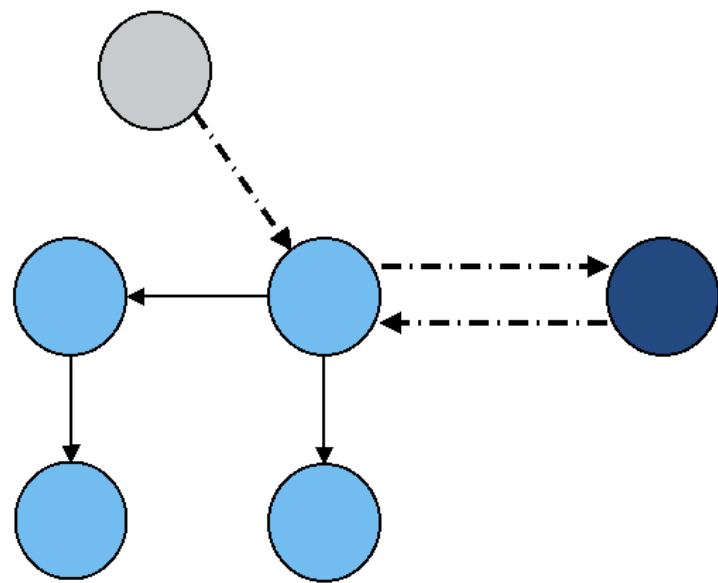


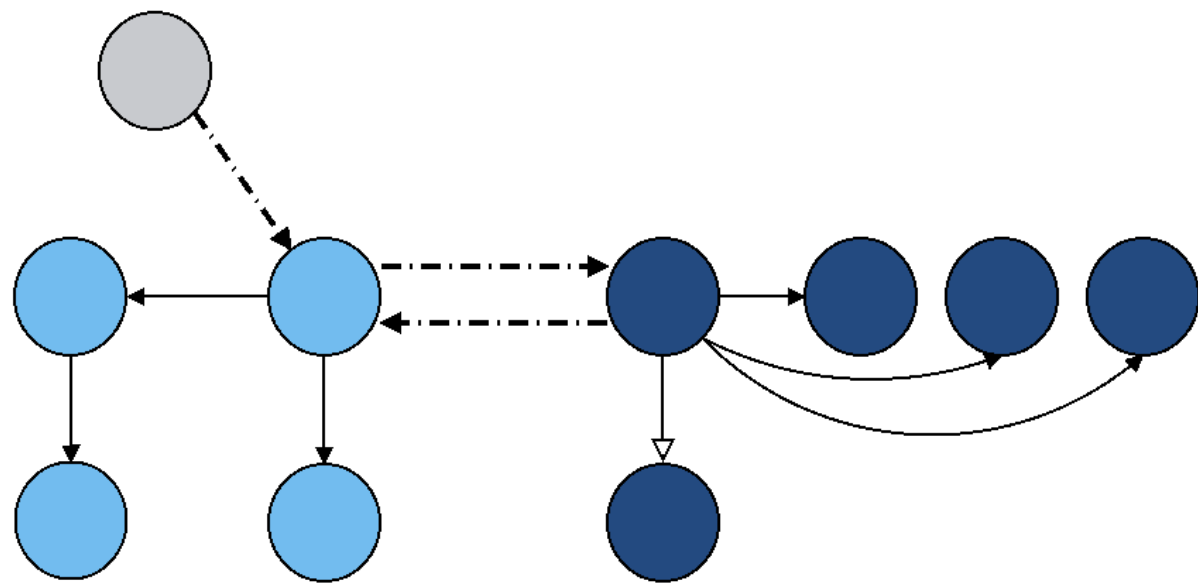
Progressive Metal

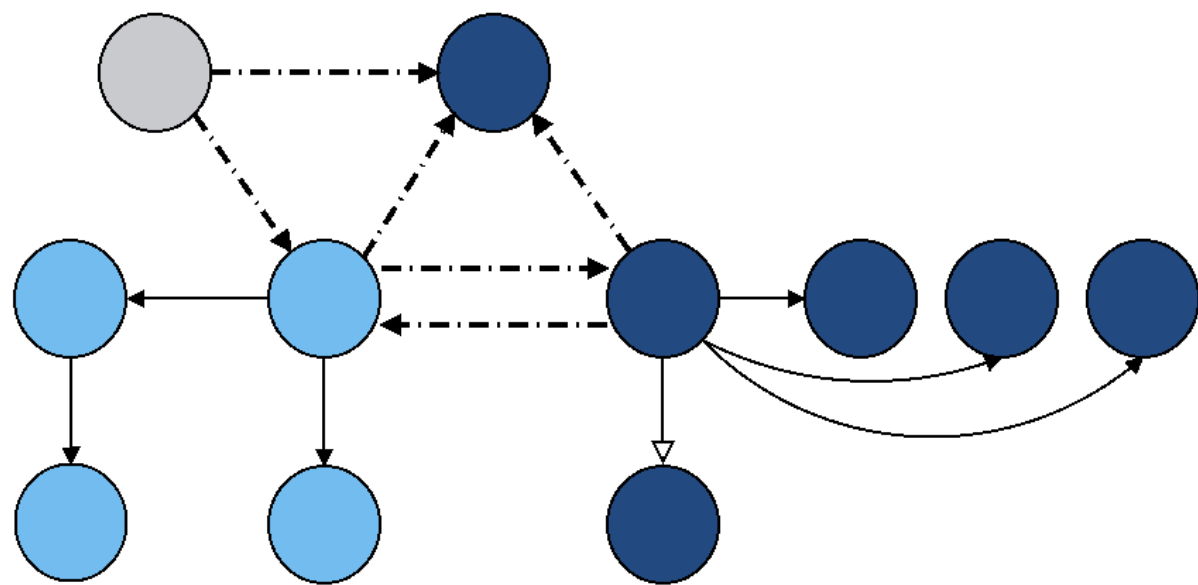


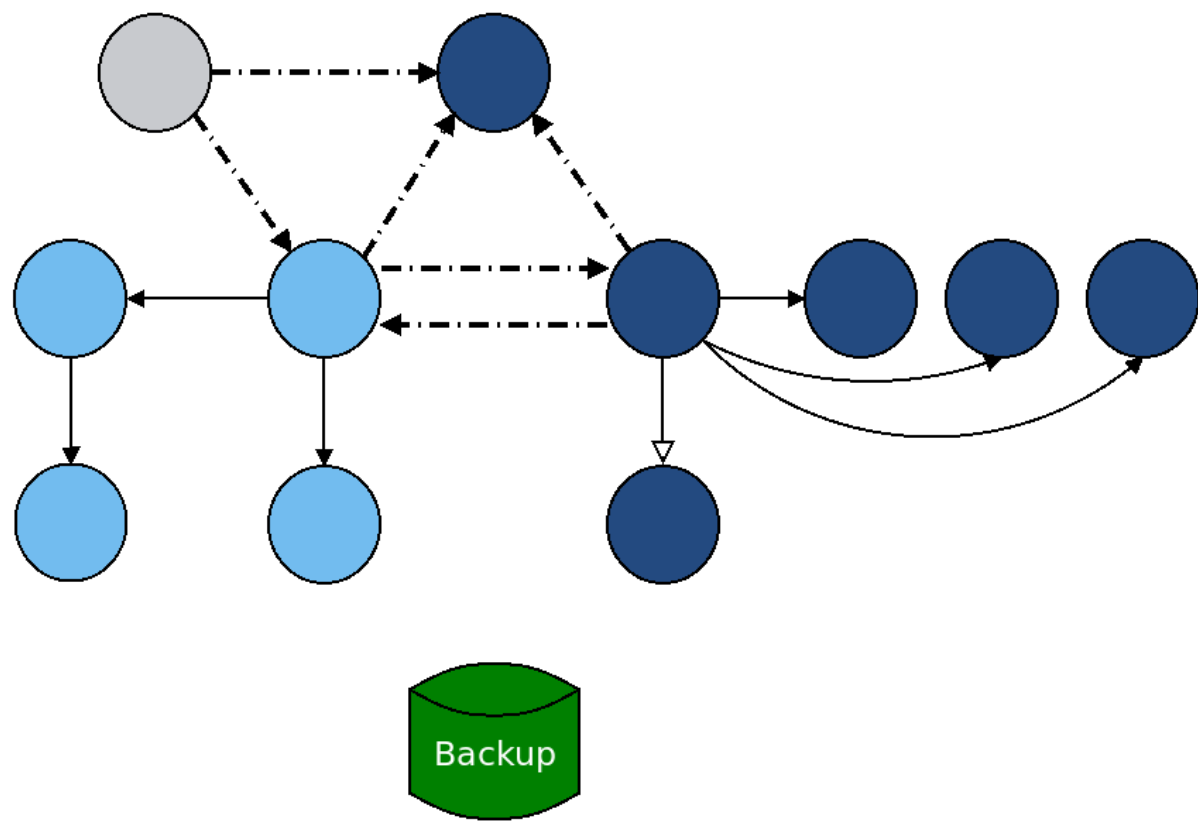


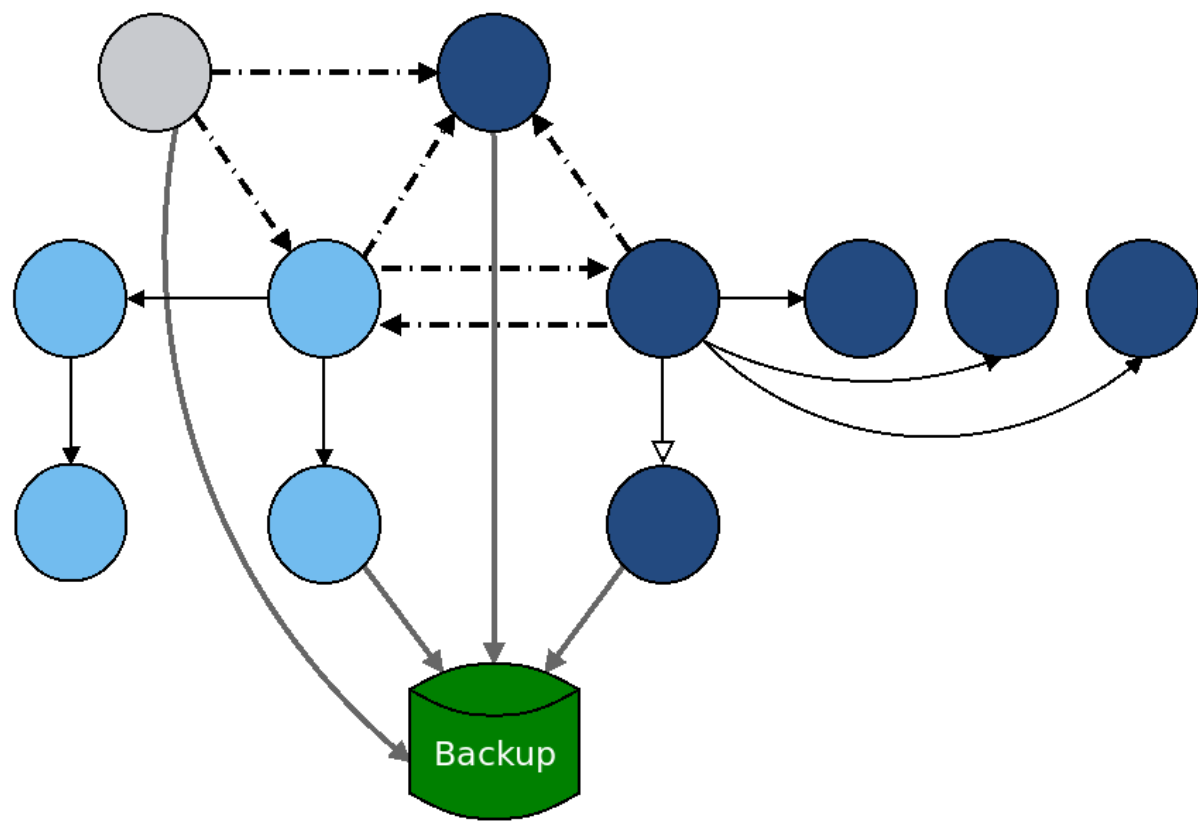




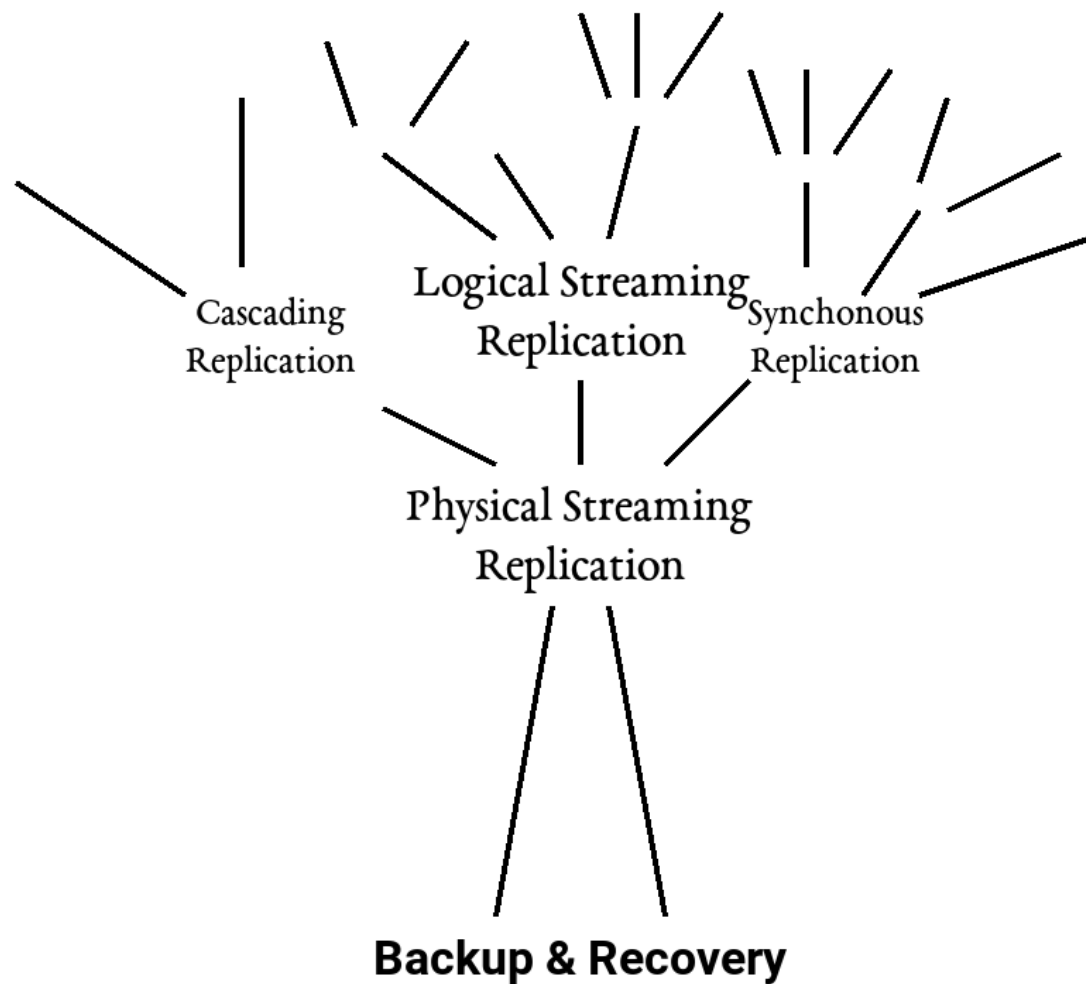








Conclusion



Open Source Community

It brings us a higher sense of community and enhances our creativity and possibilities. So why not take this concept to our art, our music?

- Kiko Loureiro

Variations on PostgreSQL Replication

Thank you

boriss.mejias@enterprisedb.com
@tchorix

