

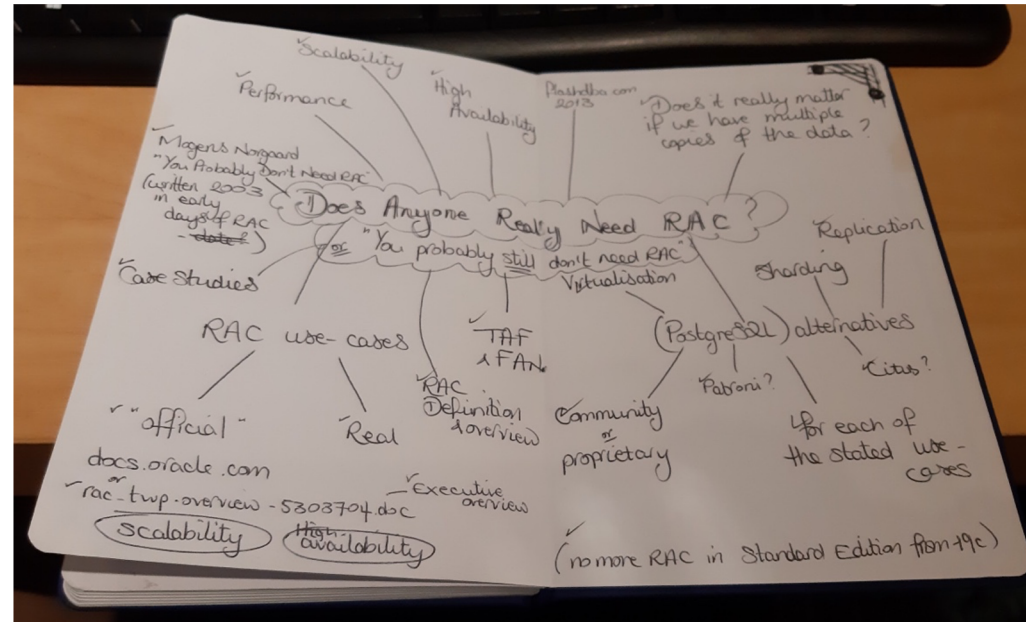
Qui a besoin de RAC ?

Karen Jex | Senior Solutions Architect @ Crunchy Data

juin 2022



Introduction





“If you have a system that needs to be up and running a few seconds after a crash, you probably need RAC. If you cannot buy a big enough system to deliver the CPU power and or memory you crave, you probably need RAC... Otherwise, you probably don't need RAC. Alternatives will usually be cheaper, easier to manage and quite sufficient.”

Morgens Norgaard “You Probably Don't Need RAC.”



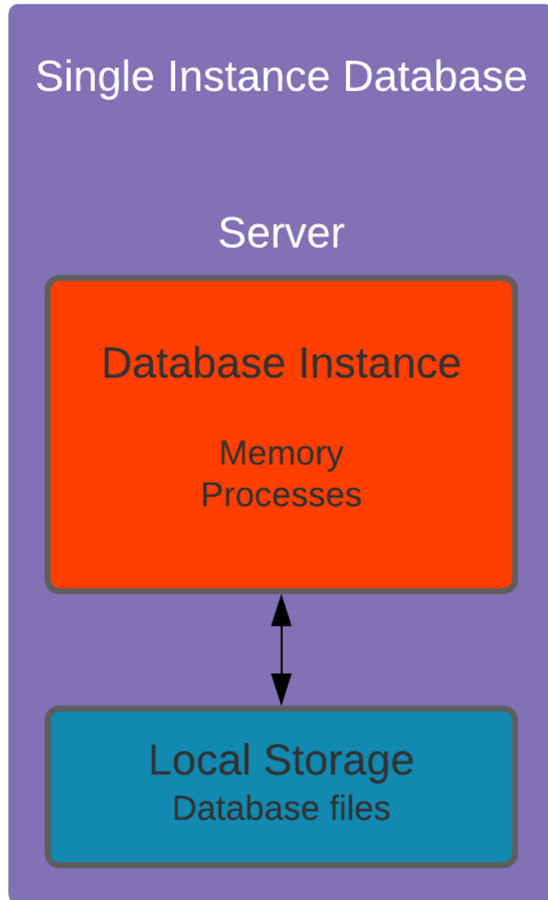
“RAC still has all the hallmarks of something people will want to buy: It increases complexity immensely, it’s expensive, it requires specialists that are increasingly hard to find, there are always excellent alternatives—and it’s pretty much perpetually unstable.”

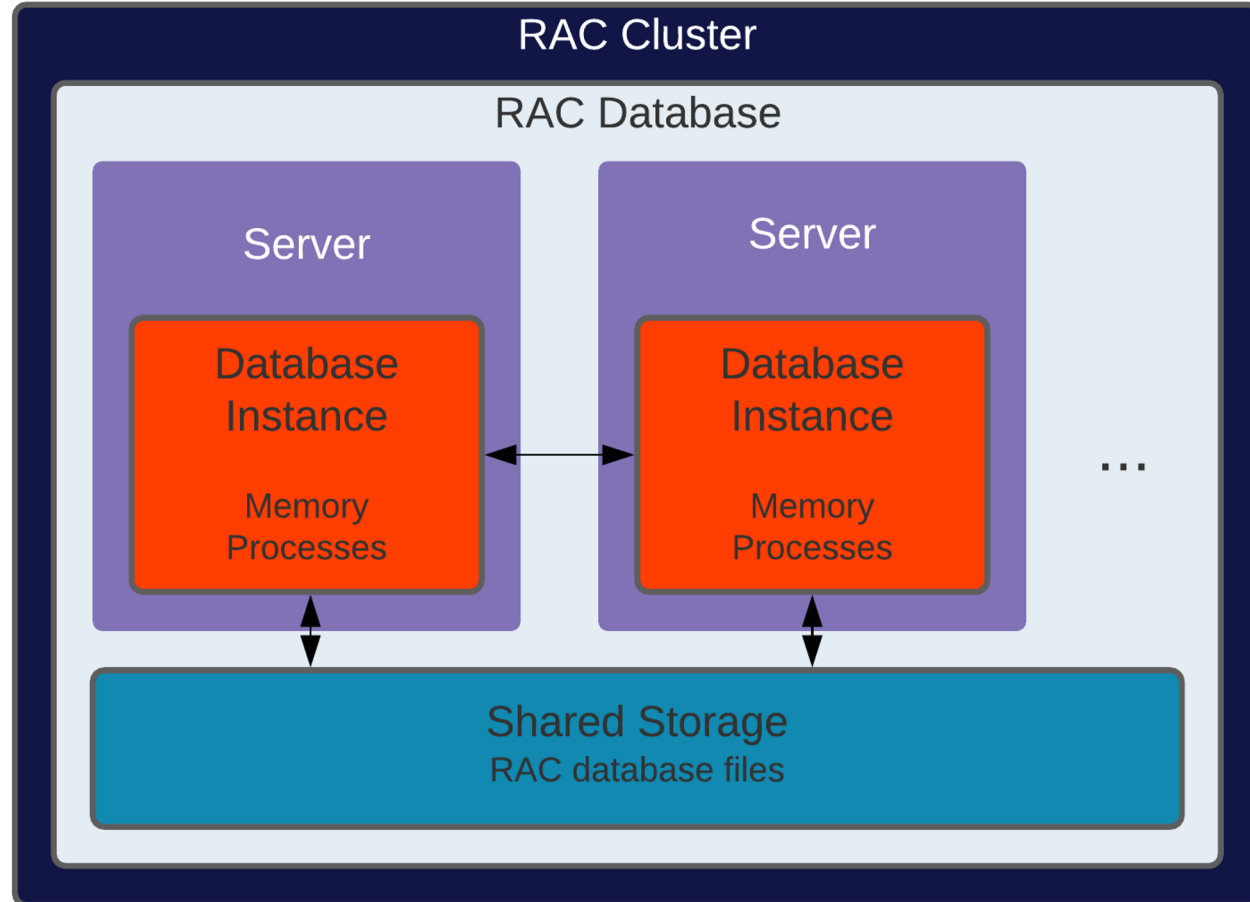
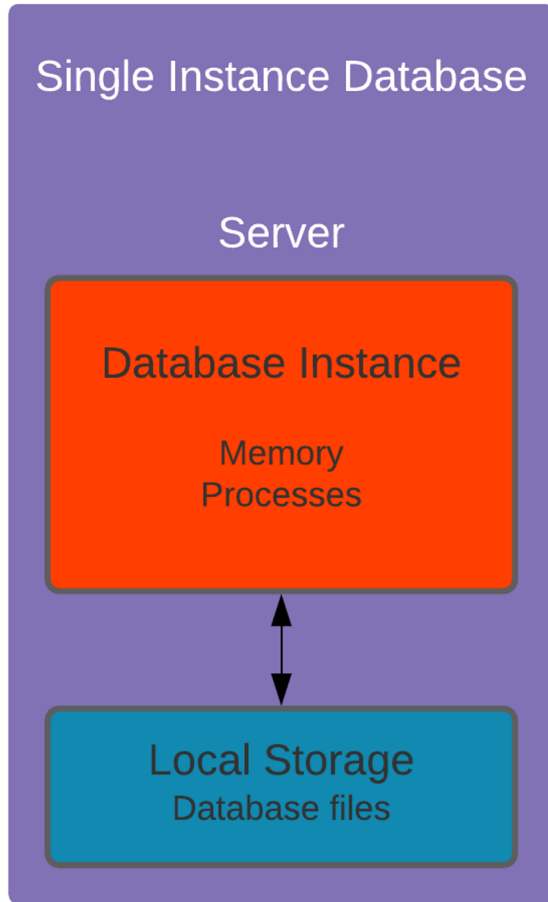
http://www.nocoug.org/Journal/NoCOUG_Journal_201108.pdf



C'est quoi, le RAC?







C'est quoi, le RAC ?

Terminologie RAC

Oracle Automatic Storage Management (ASM)
+ Oracle Clusterware
= Oracle Grid Infrastructure



C'est quoi, le RAC ?

Installation & Configuration

- *Prérequis serveur*
- *Users et groupes OS*
- *Réseau*
- *Stockage partagé*
- *Grid Infrastructure*
- *Clusterware*
- *Logiciel Oracle RAC*
- *Base de Données*



Avantages de RAC



“Oracle Real Application Clusters (RAC) allow customers to run a single Oracle Database across multiple servers in order to maximize availability and enable horizontal scalability, while accessing shared storage.”

“The database spans multiple hardware systems and yet appears as a single unified database to the application.”

<https://www.oracle.com/database/real-application-clusters>



Avantages de RAC

Consolidation



Avantages deRAC

Haute Disponibilité



Haute Disponibilité

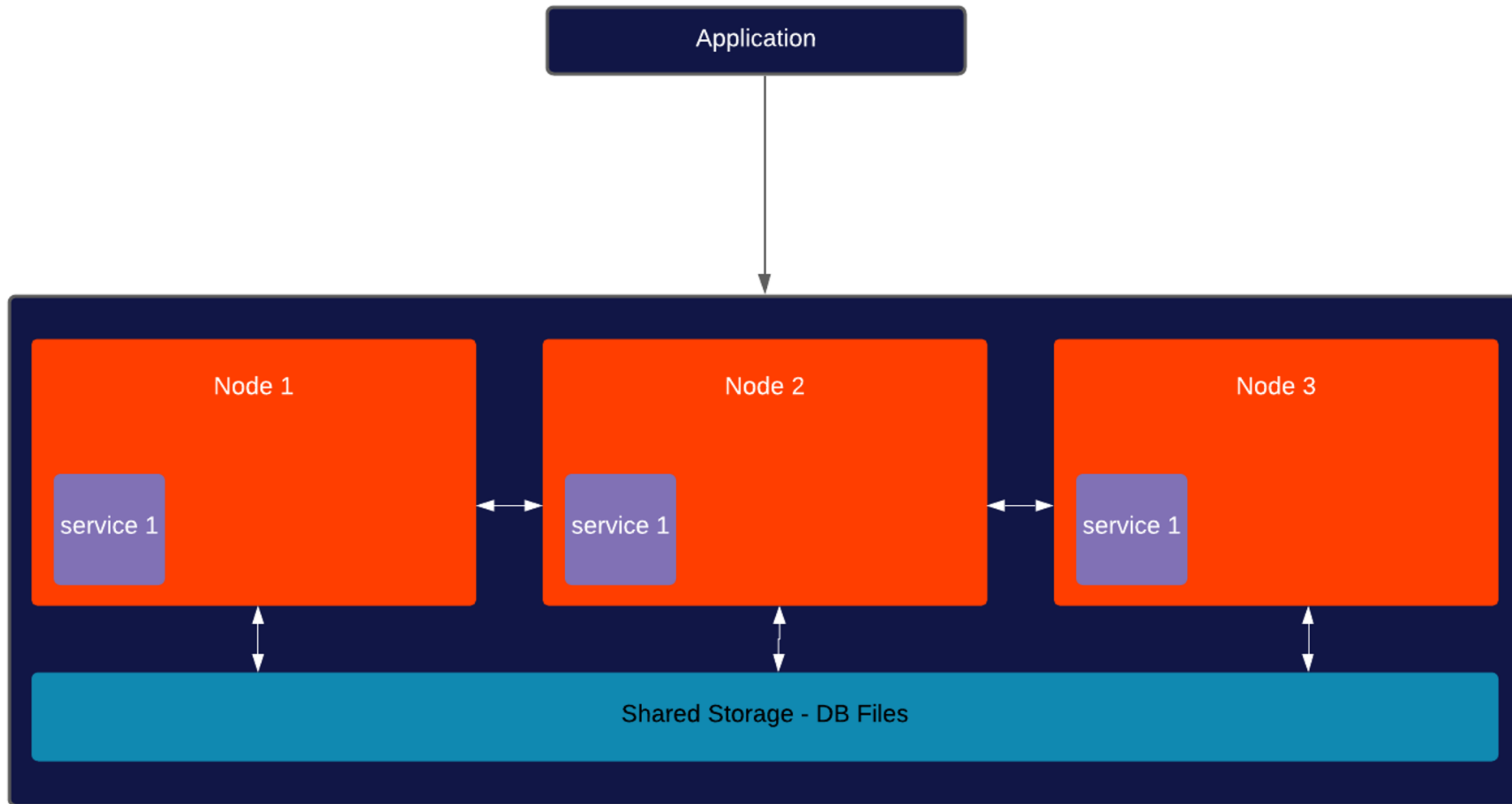
Minimiser l'Indisponibilité Imprévue

En cas de faillite d'une instance ou serveur:

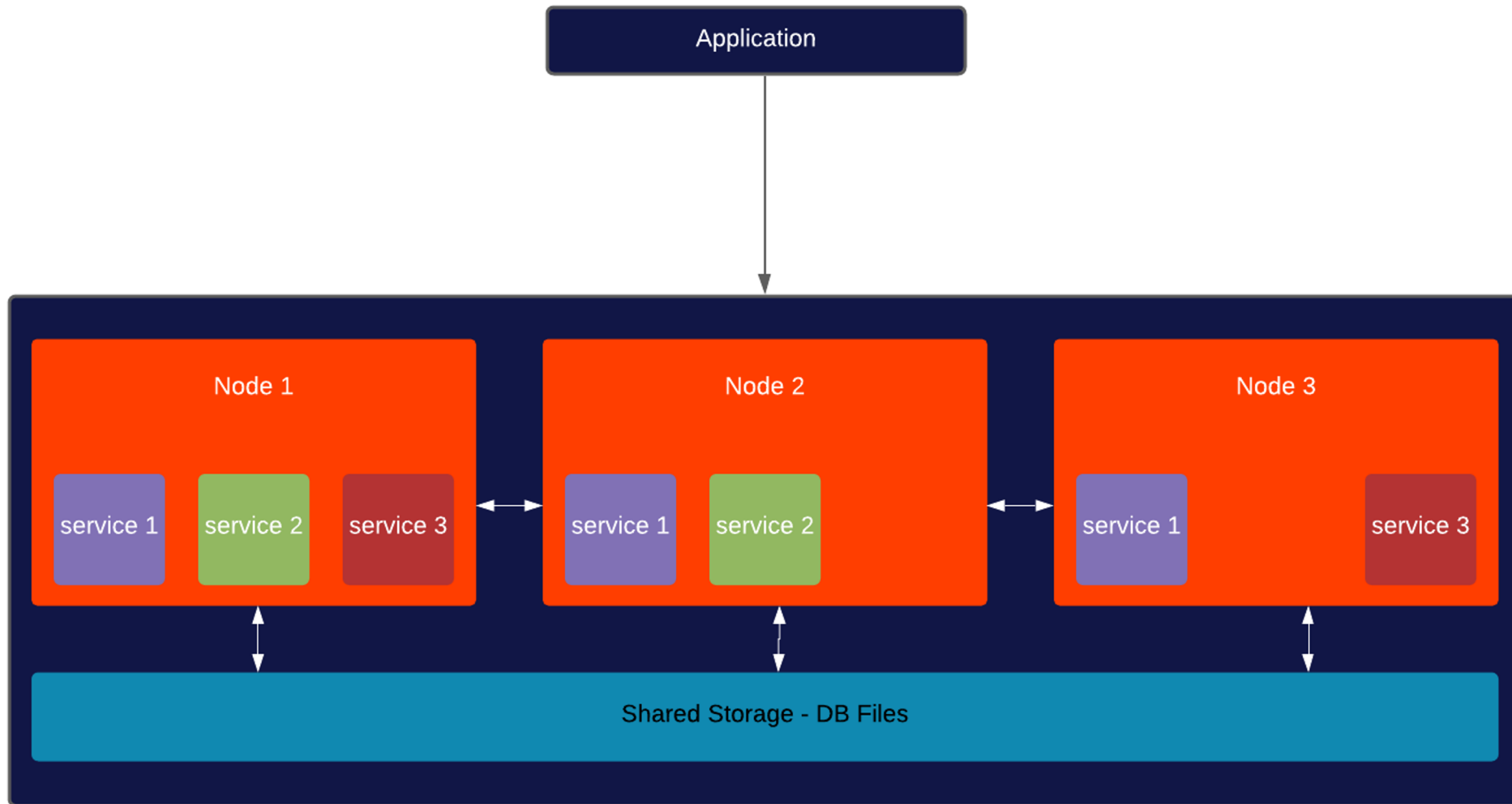
- Clients connectés sont notifiés*
- Nouvelles connexions dirigées vers instances restantes*
- Indisponibilité transparent pour le client*



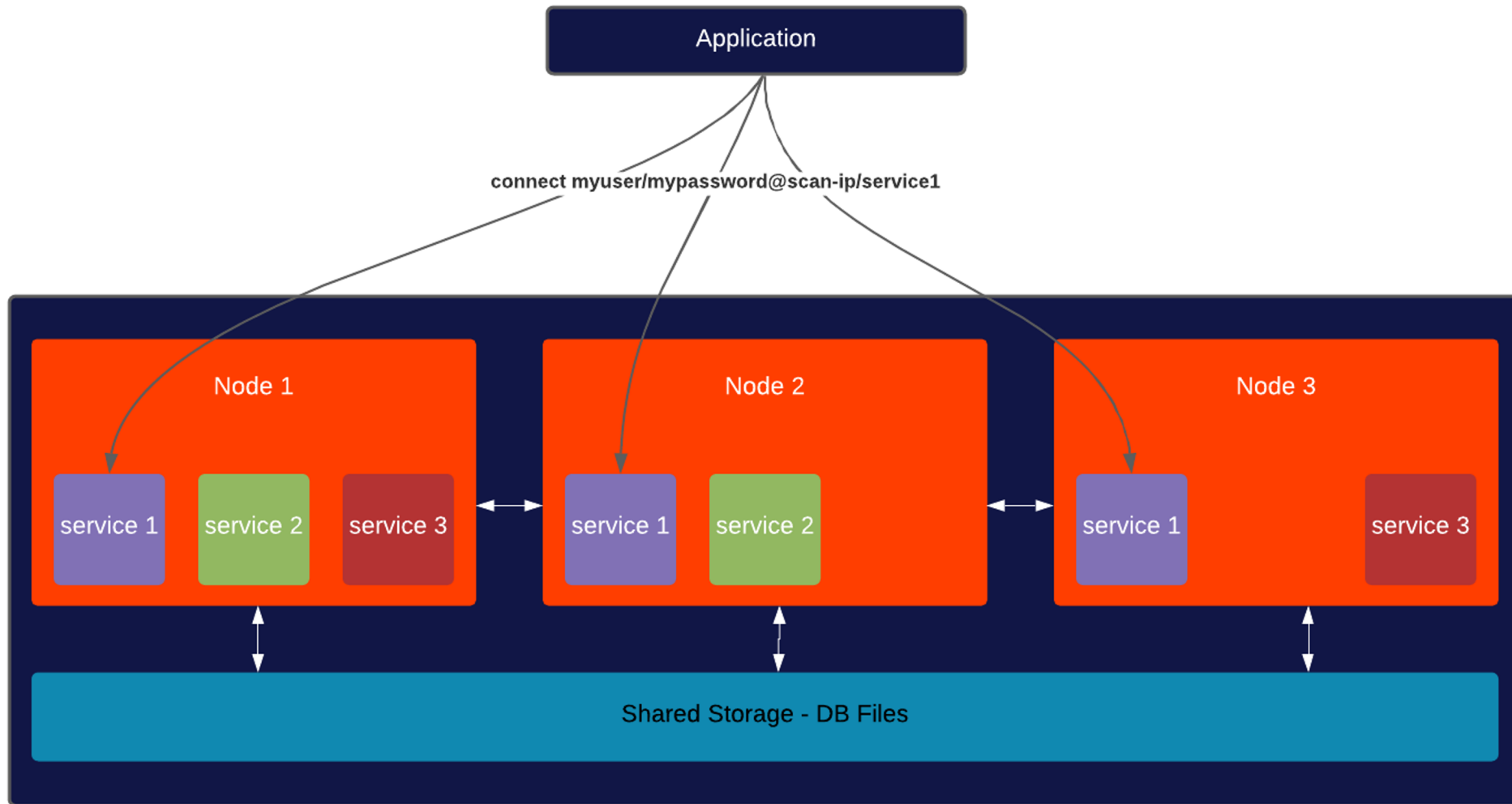
Haute Disponibilité



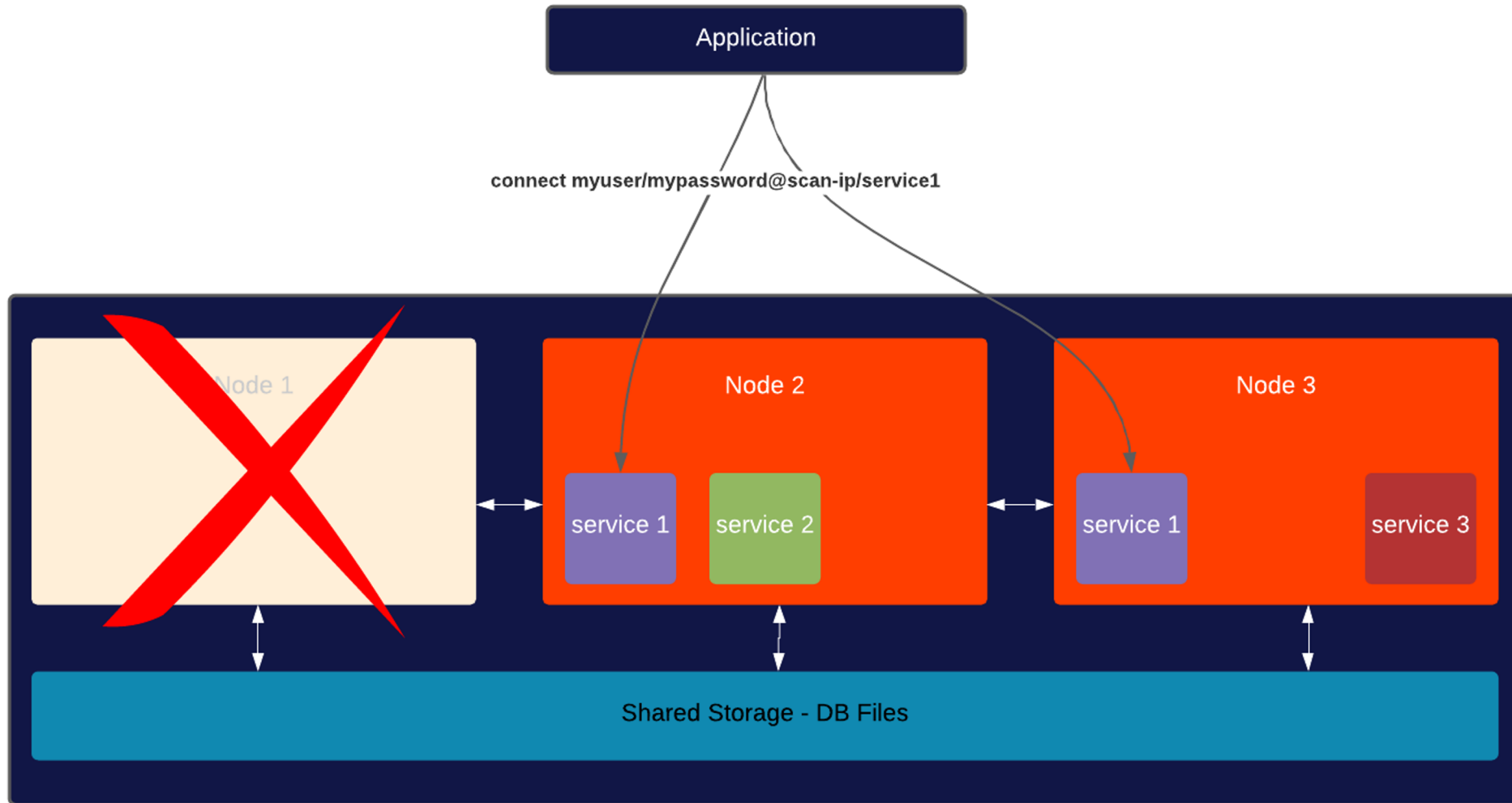
Haute Disponibilité



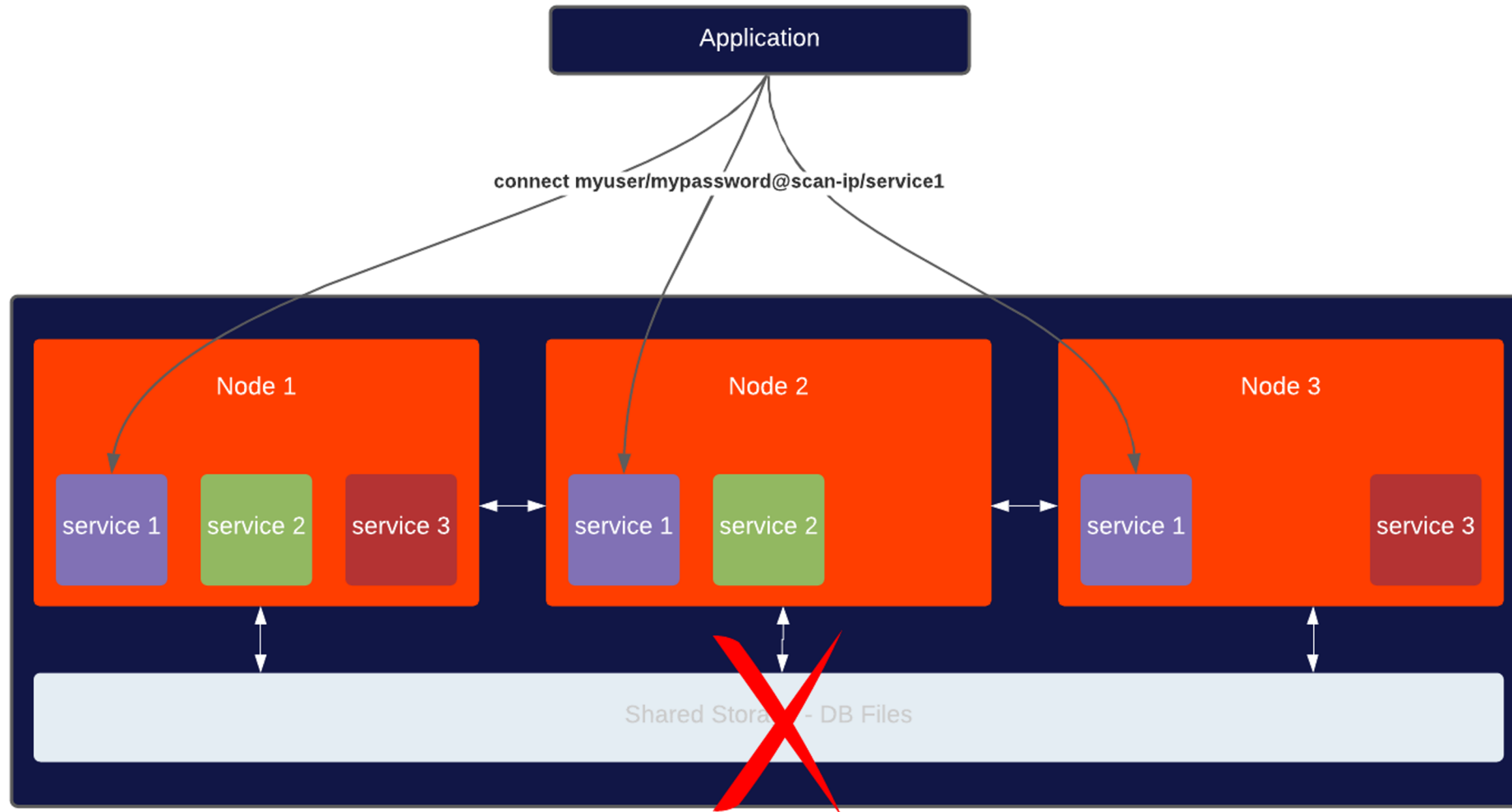
Haute Disponibilité



Haute Disponibilité



Haute Disponibilité



Haute Disponibilité

Minimiser l'Indisponibilité Imprévue

- Groupes de défaillance ASM*
- Oracle Active Data Guard*
- Oracle GoldenGate*
- Sauvegardes*



Haute Disponibiilité

Minimiser l'Indisponibilié Planifiée

- Rolling upgrades*
- Application des patch serveur-par-serveur*

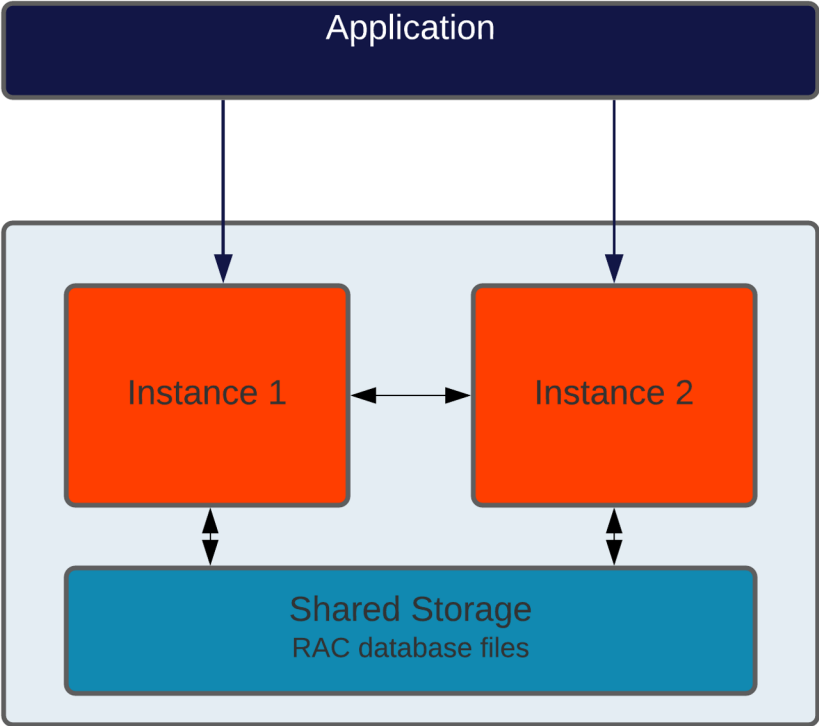


Avantages de RAC

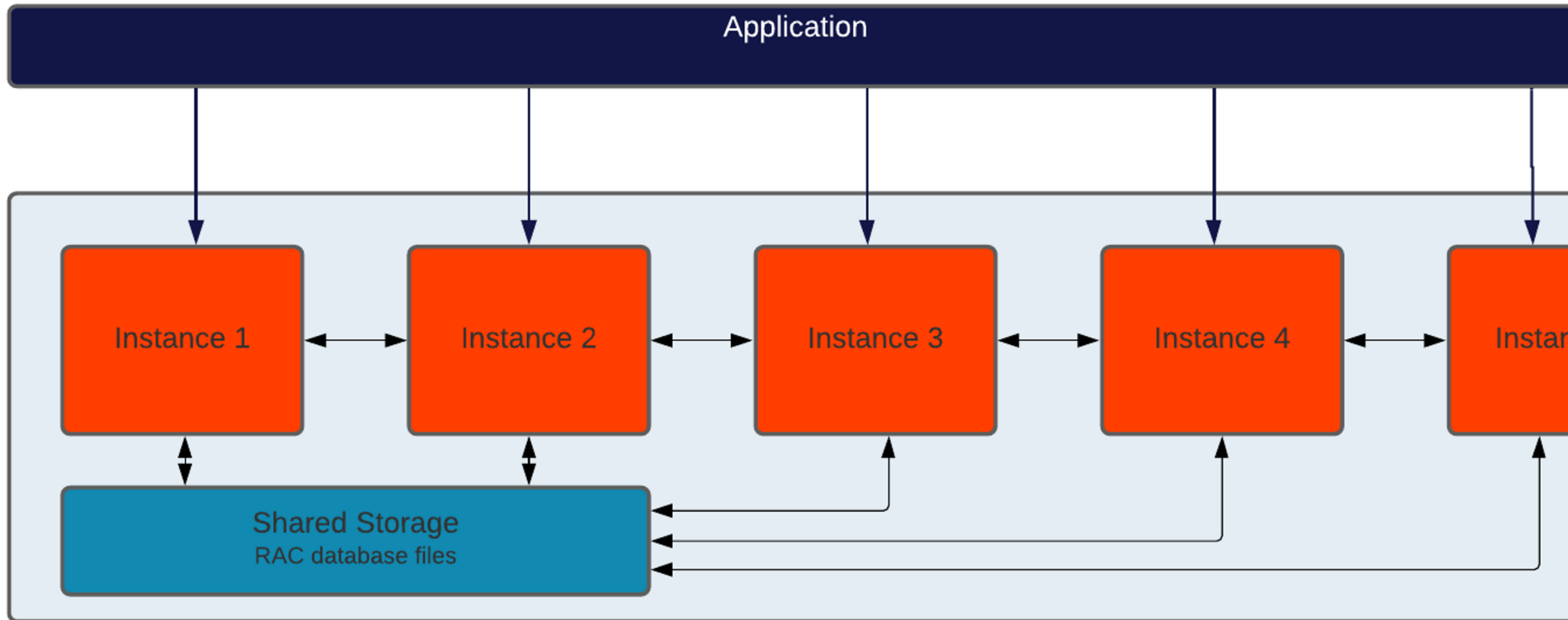
Scalability / Extensibilité



Scalability / Extensibilité



Scalability / Extensibilité



*“Pourquoi on doit absolument,
définitivement utiliser RAC”*



“Pourquoi on doit absolument, définitivement utiliser RAC”

“Nous ne voulons pas plusieurs copies des données”

“Il nous faut zero indisponibilité”

“Il nous faut 5 neufs de disponibilité”

“Nous aurons 10 mille connexions simultanées”

“Il nous faut un scalabilité des écritures”

“Il nous faut 100% de disponibilité pour les lectures”

“Il nous faut un scalabilité des lectures”

“Nous avons 10 mille TPS par need”

“Nous ne voulons pas de point de défaillance unique”



“Pourquoi on doit absolument, définitivement utiliser RAC”

Haute Disponibilité

- zero indisponibilité*
- 5 neufs disponibilité*
- 100% disponibilité pour lectures*
- Pas de point de défaillance unique*



“Pourquoi on doit absolument, définitivement utiliser RAC”

Scalability / Extensibilité

- Extensibilité lectures*
- Extensibilité écritures*
- X mille TPS par nœd*
- X mille connexions simultanées*



“Pourquoi on doit absolument, définitivement utiliser RAC”

Autre

- . Pas de duplication des données*



PostgreSQL “standard”



PostgreSQL “standard”

Disponibilité



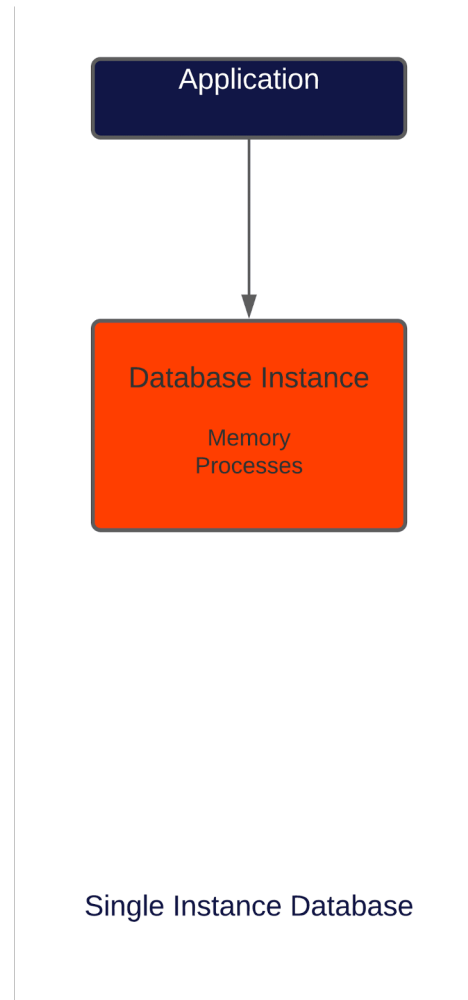
PostgreSQL “standard”: Disponibilité

Application

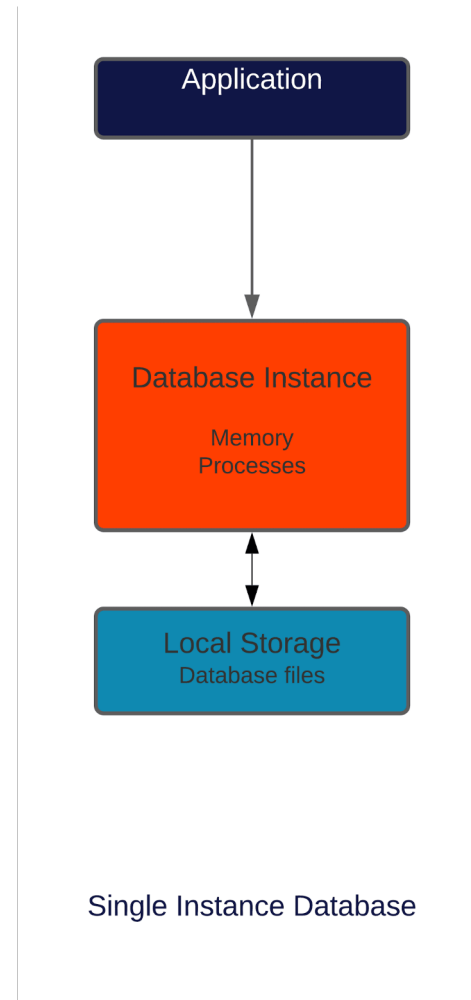
Single Instance Database



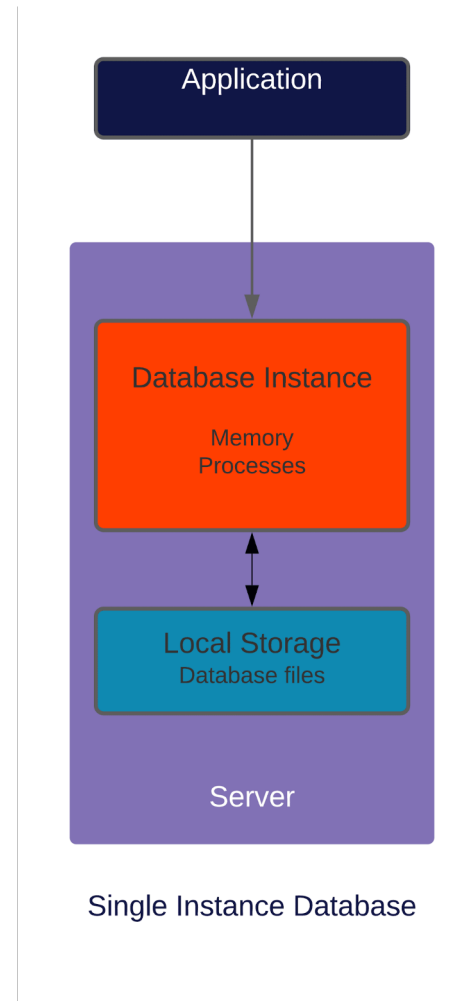
PostgreSQL “standard”: Disponibilité



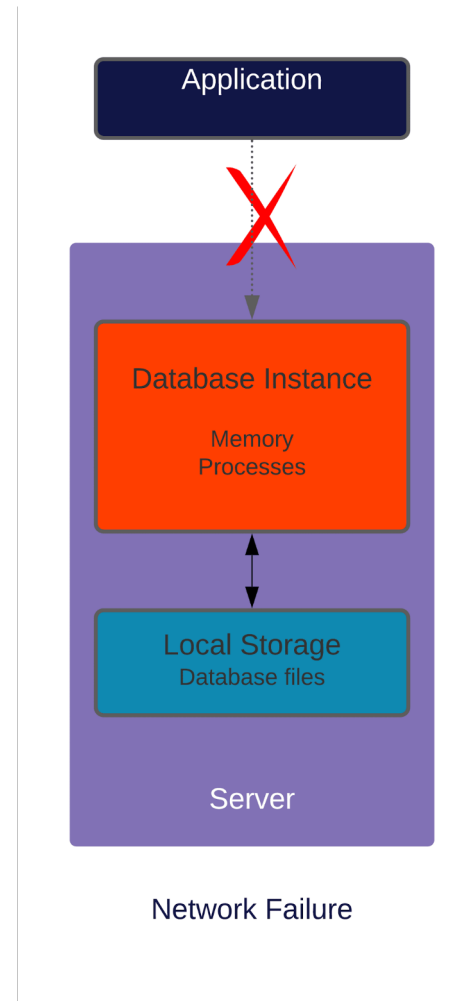
PostgreSQL “standard”: Disponibilité



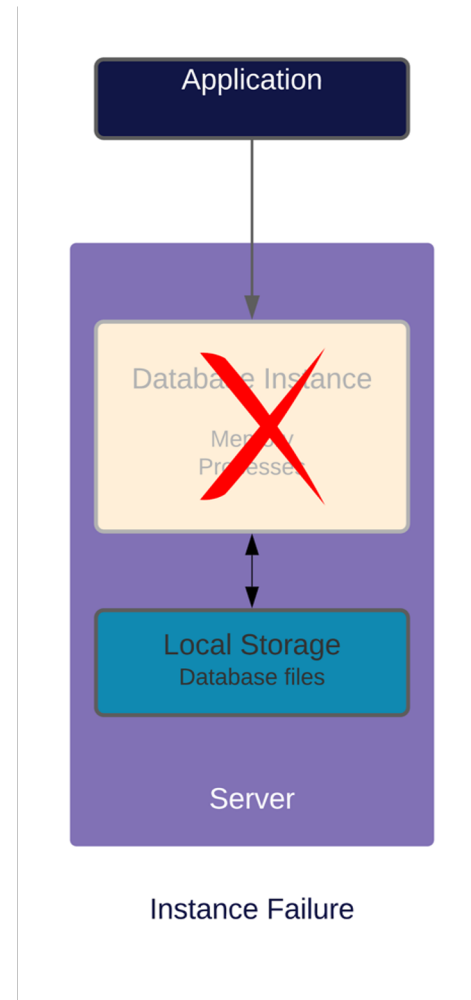
PostgreSQL “standard”: Disponibilité



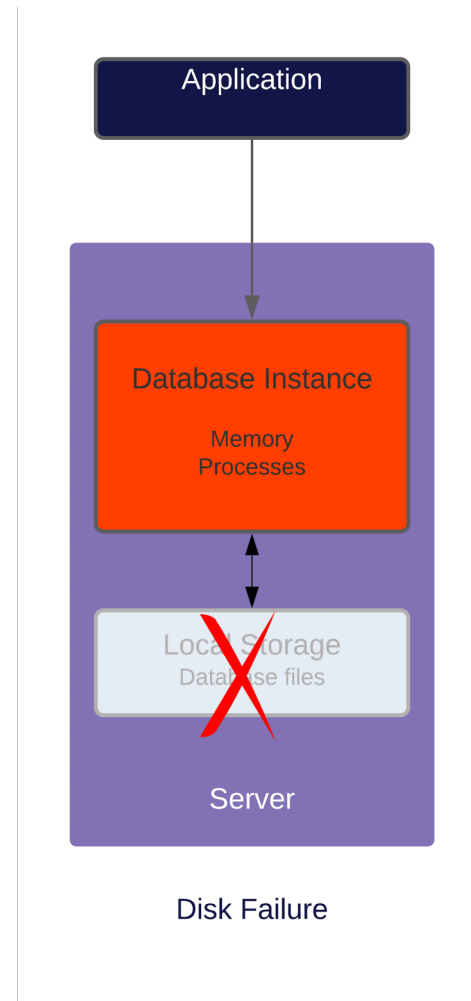
PostgreSQL “standard”: Disponibilité



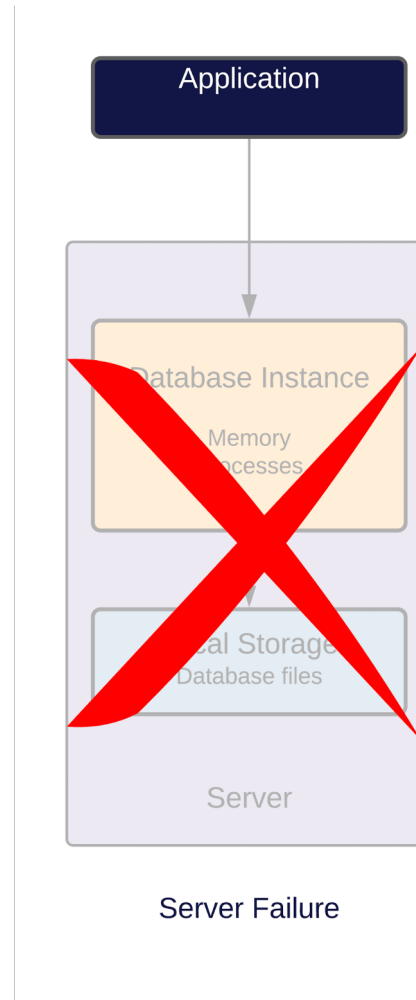
PostgreSQL “standard”: Disponibilité



PostgreSQL “standard”: Disponibilité



PostgreSQL “standard”: Disponibilité





PostgreSQL “standard”

Scalability / Extensibilité



Scalability / Extensibilité

Max Connexions

< 500



Scalability / Extensibilité

Max Connexions

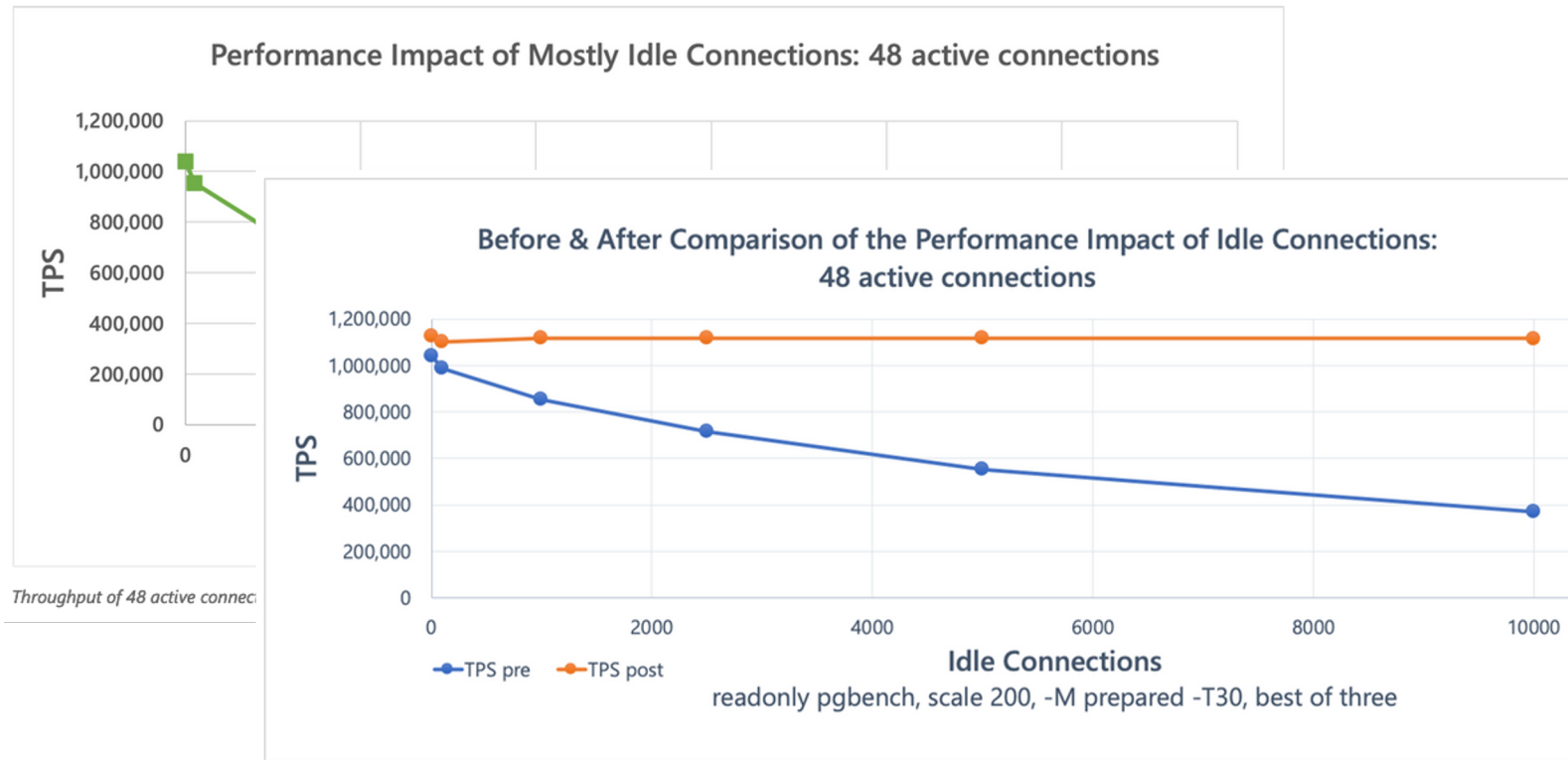


Figure 3: Benchmark result comparison (48 active connections running read-only pgbench, in presence of a variable number of idle connections), showing the effects of the snapshot scalability improvements.

Andres Freund: ["Improving Postgres Connection Scalability: Snapshots"](#)



Scalability / Extensibilité

TPS



PostgreSQL Alternatives au RAC



PostgreSQL Alternatives au RAC

Haute Disponibilité

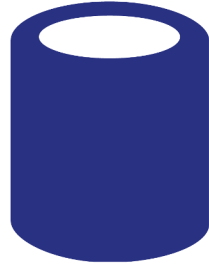


Réplication

Image by [Andrew Martin](#) from [Pixabay](#)



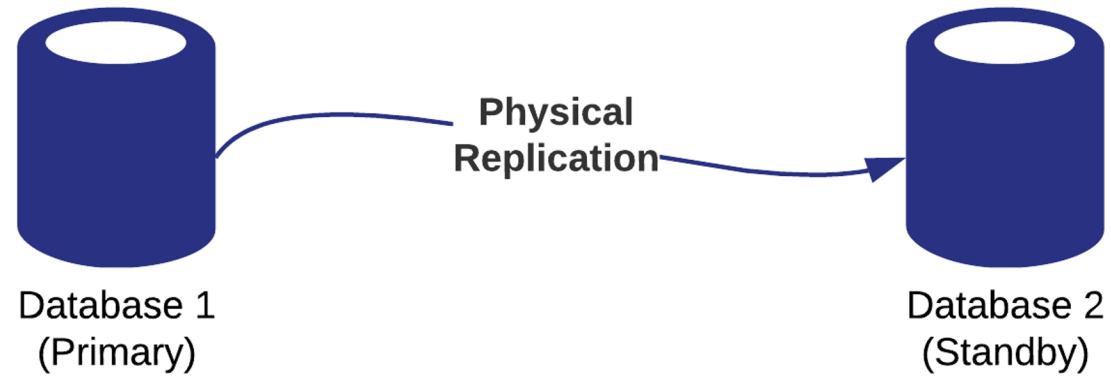
Réplication Physique



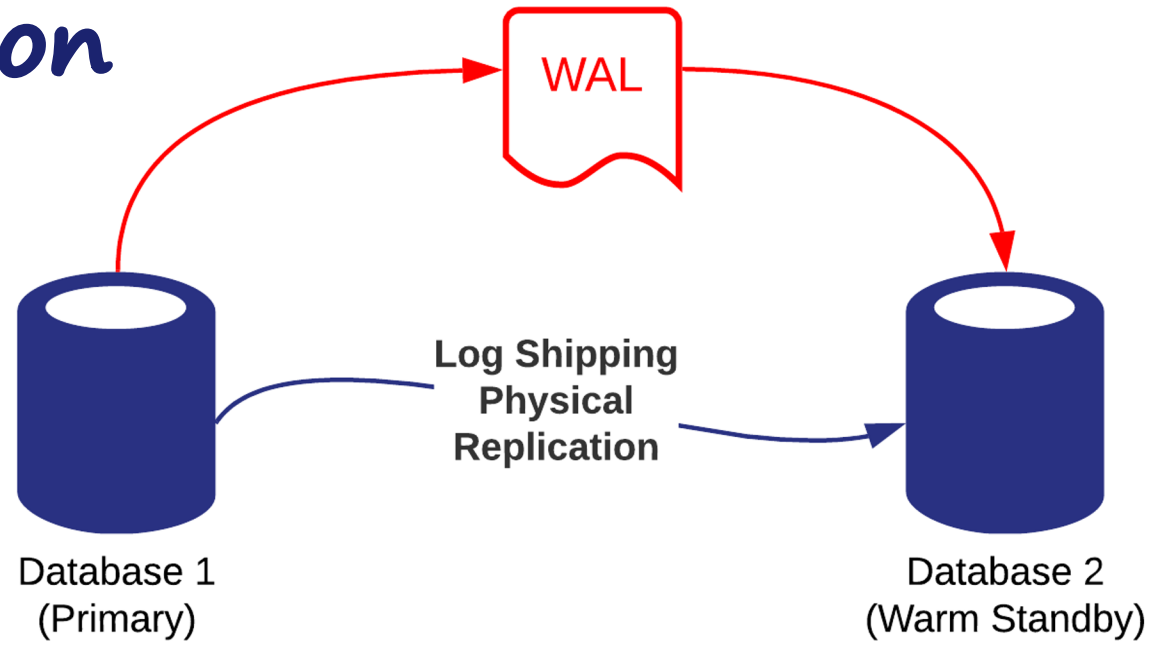
Database 1
(Primary)



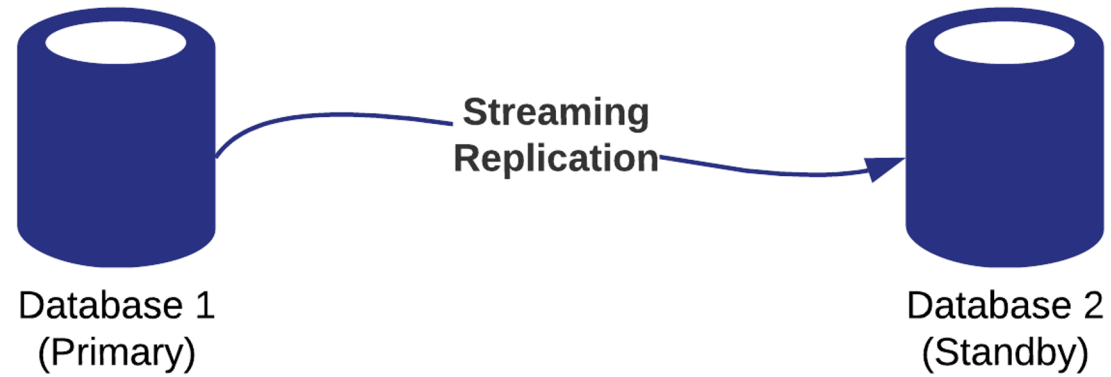
Réplication Physique



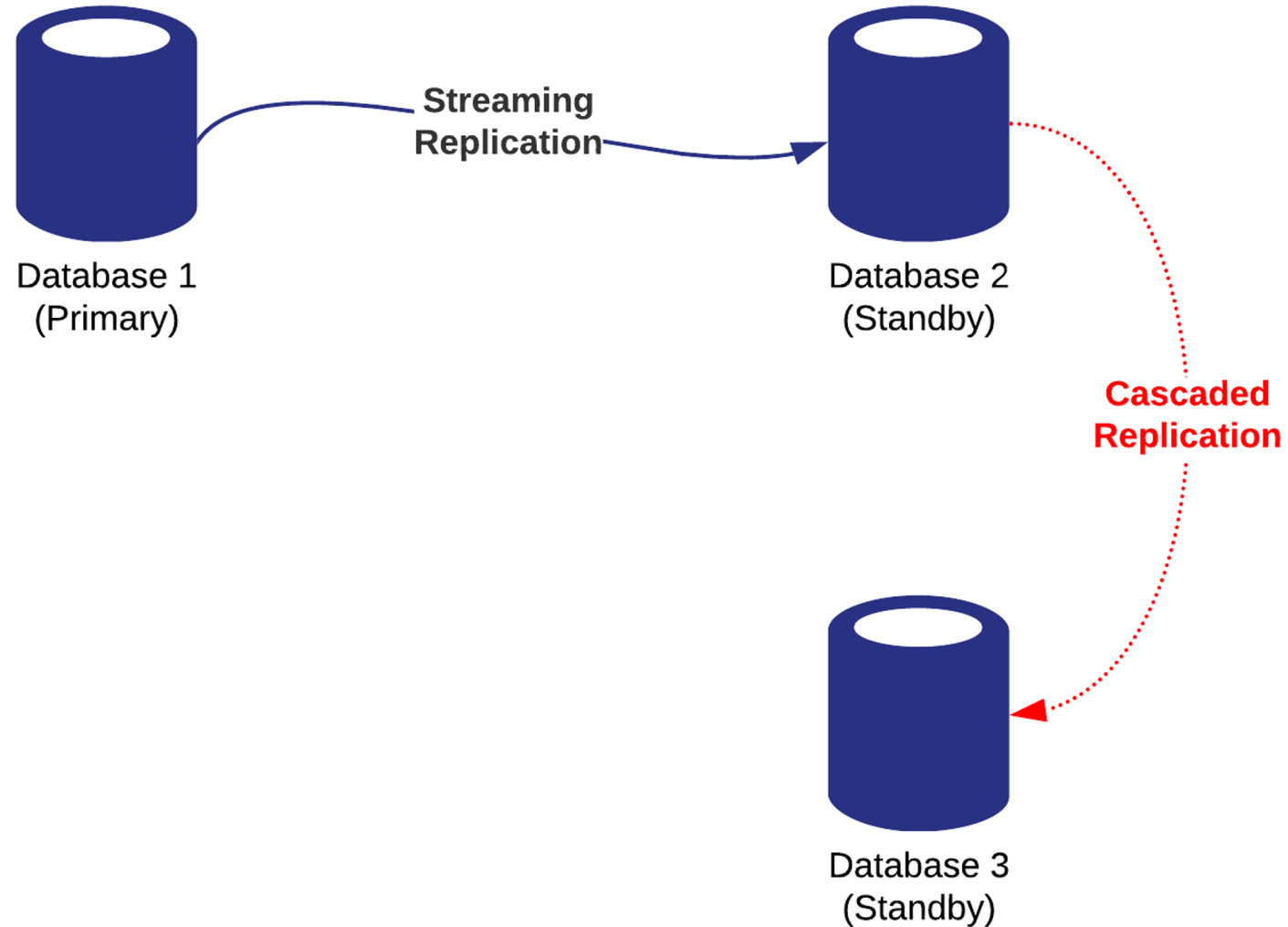
Réplication Physique



Réplication Physique

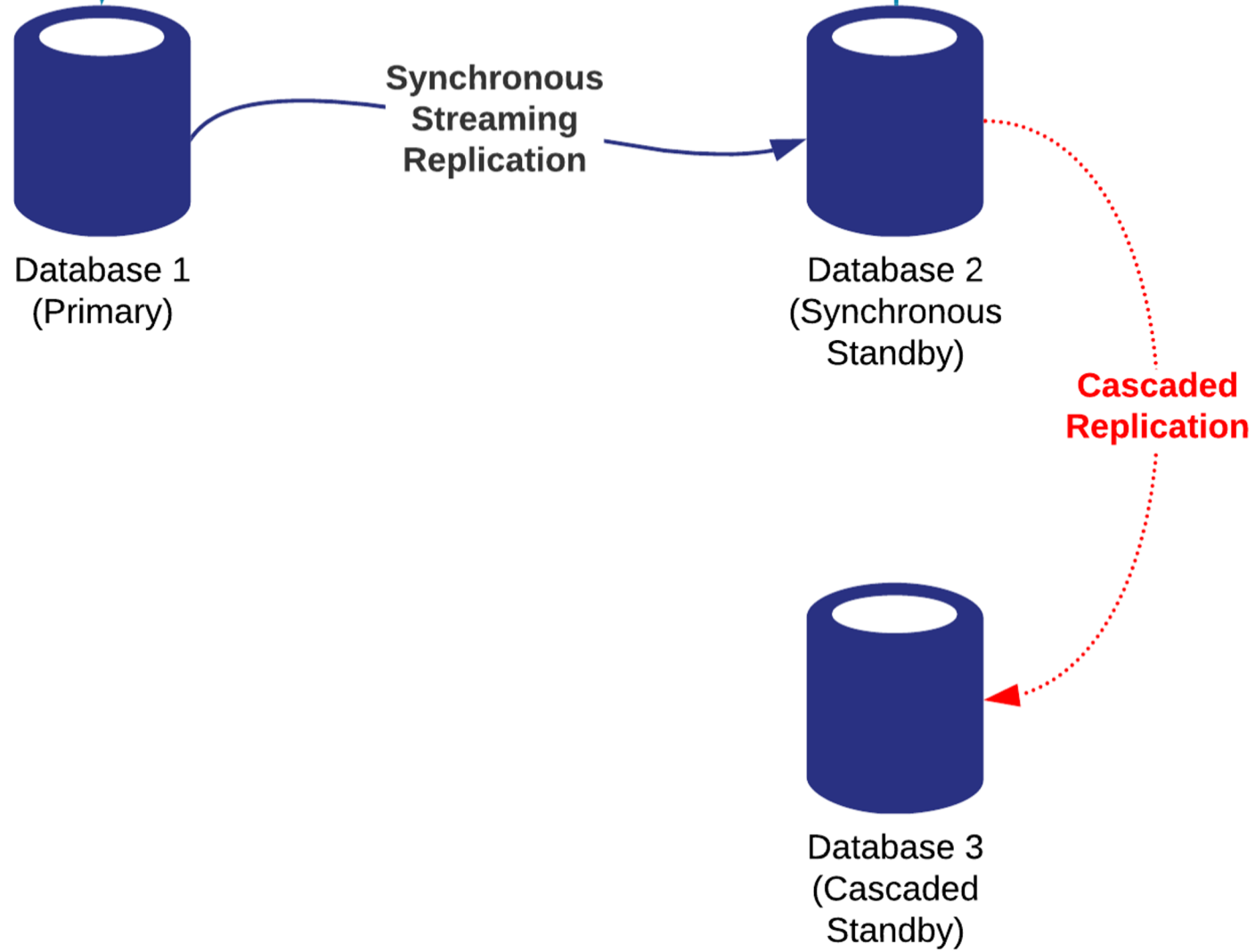


Réplication Physique

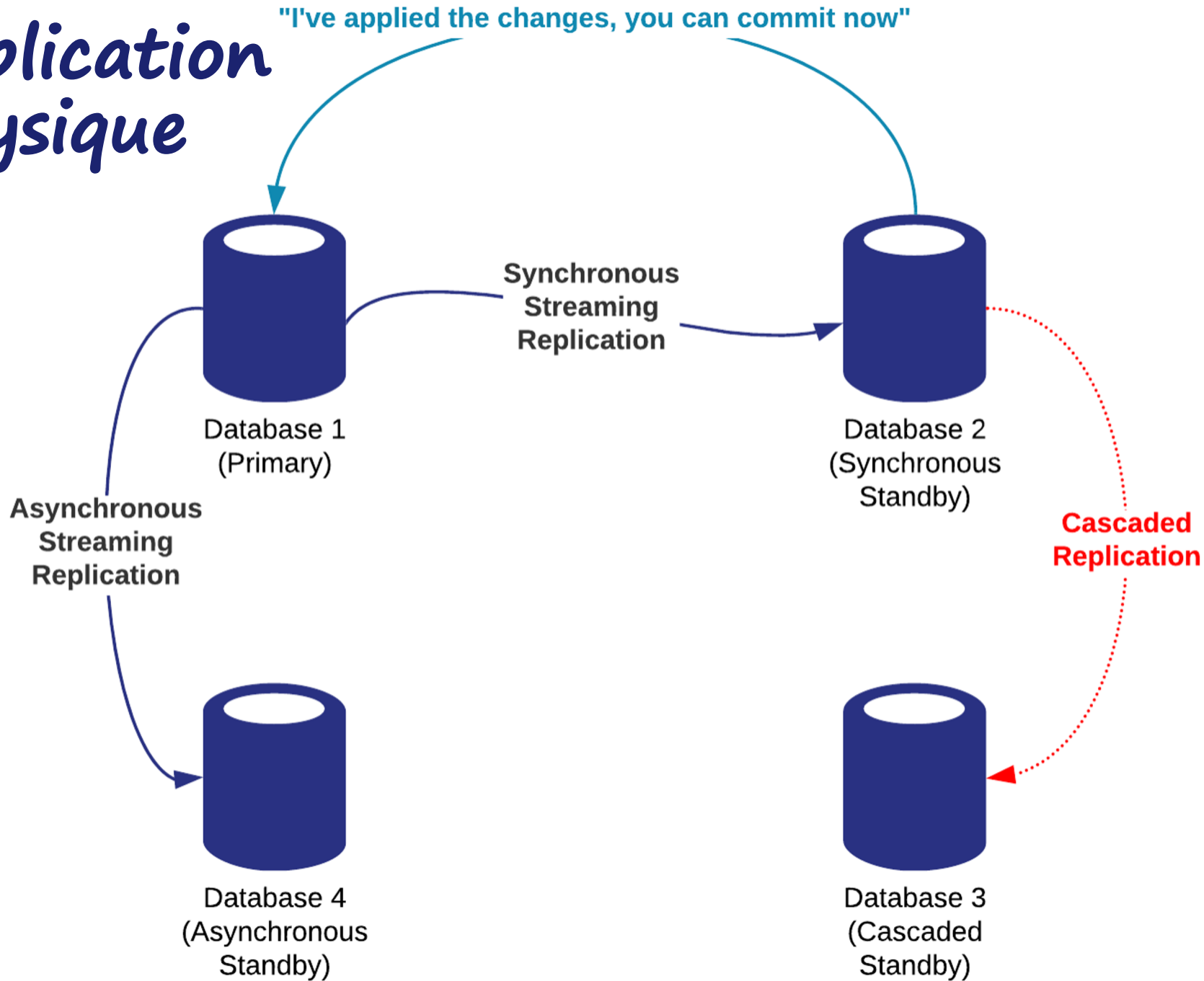


Réplication Physique

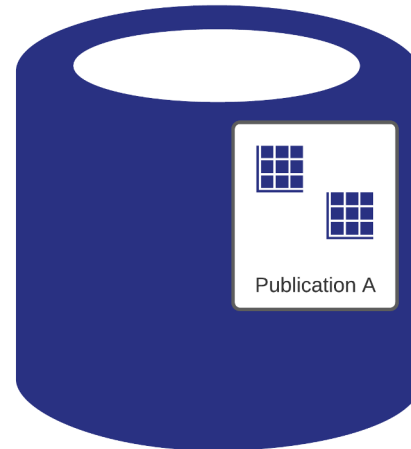
"I've applied the changes, you can commit now"



Réplication Physique



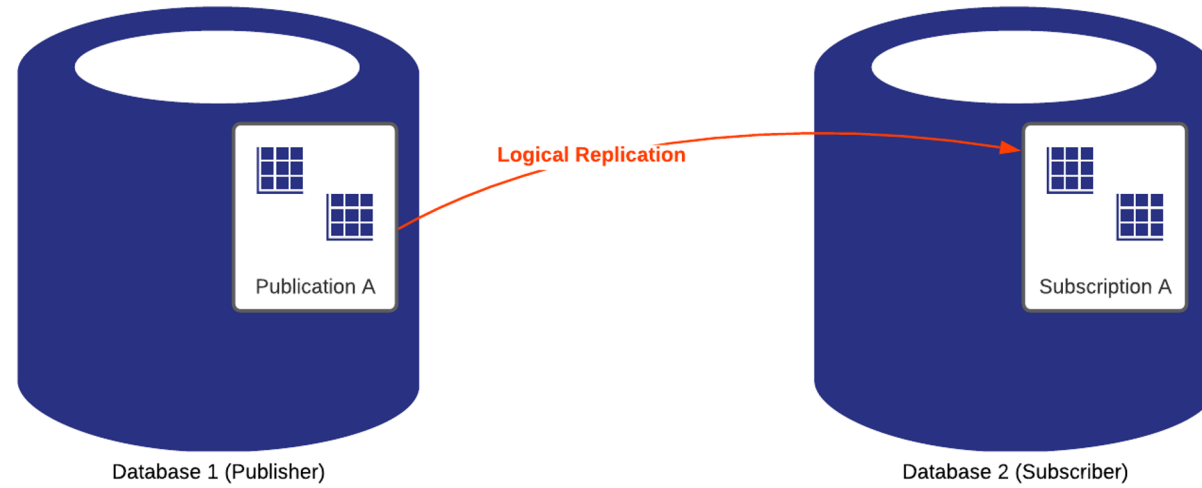
Réplication Logique



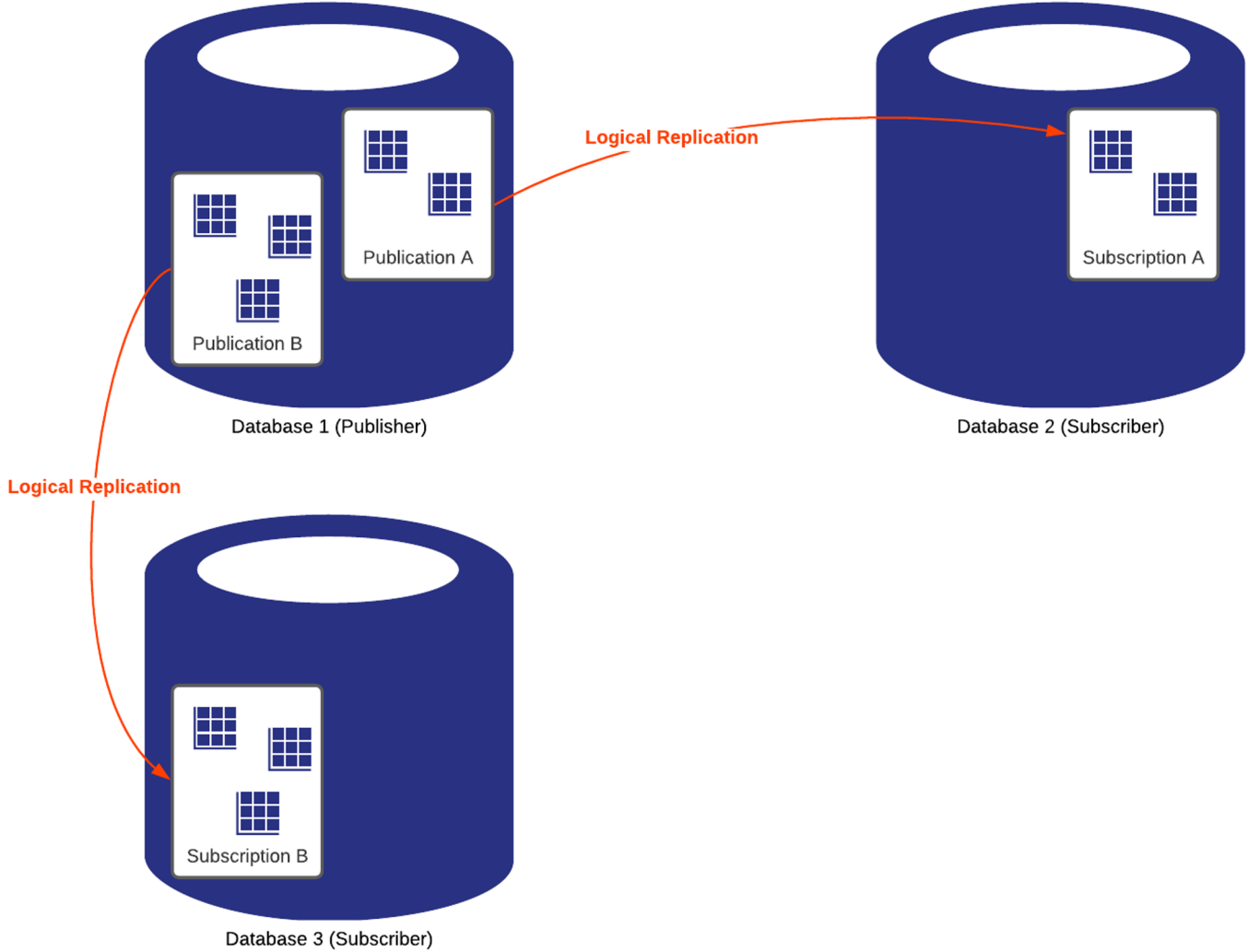
Database 1 (Publisher)



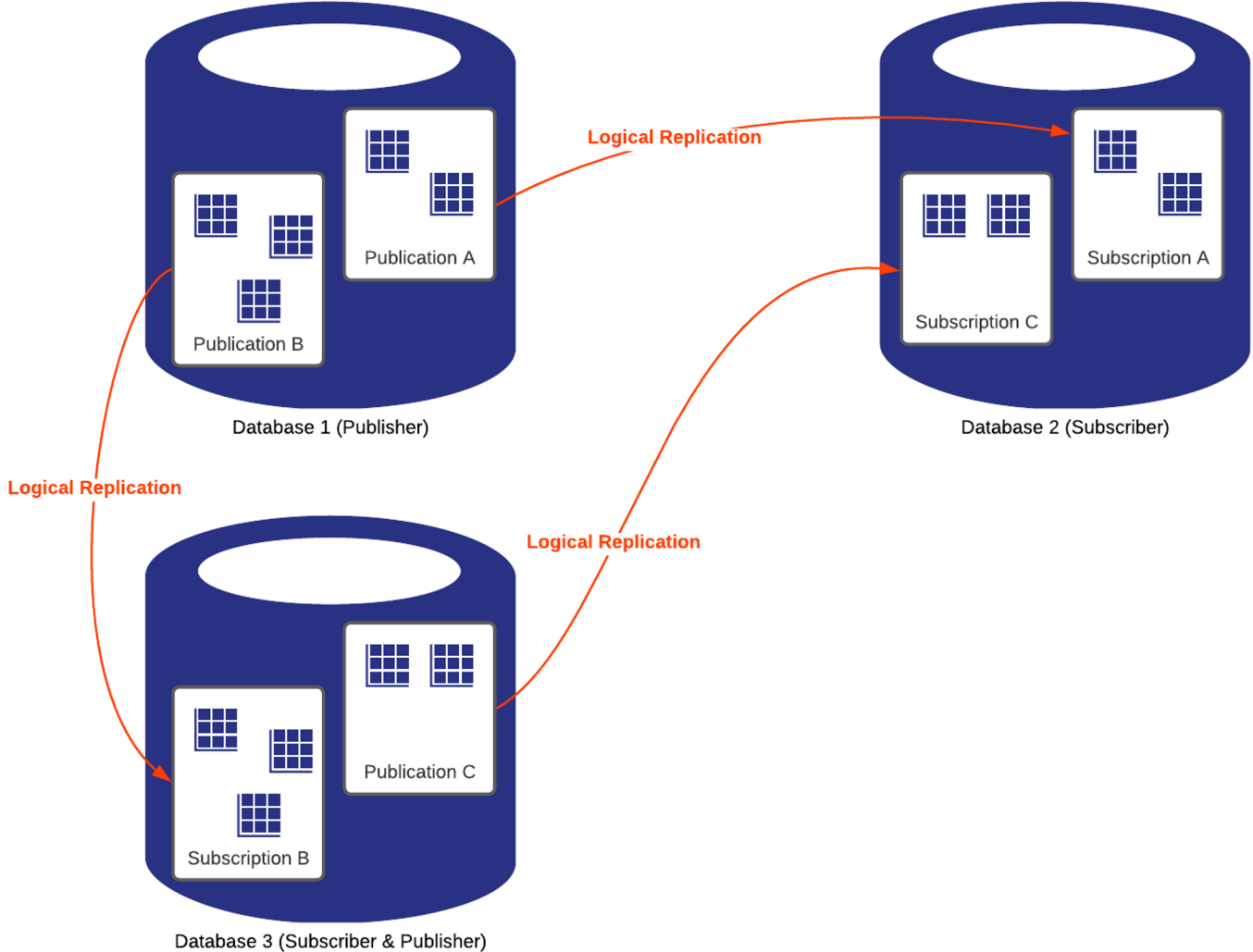
Réplication Logique



Réplication Logique



Réplication Logique



Haute Disponibilité

Failover



Patroni

PostgreSQL Automatic Failover (PAF)

Pacemaker

PostgreSQL Alternatives au RAC

BDR

Haute Disponibilité

stolon

Corosync

EFM

pglookout

Pgpool-II

repmgr



PostgreSQL Alternatives au RAC : Haute Disponibilité

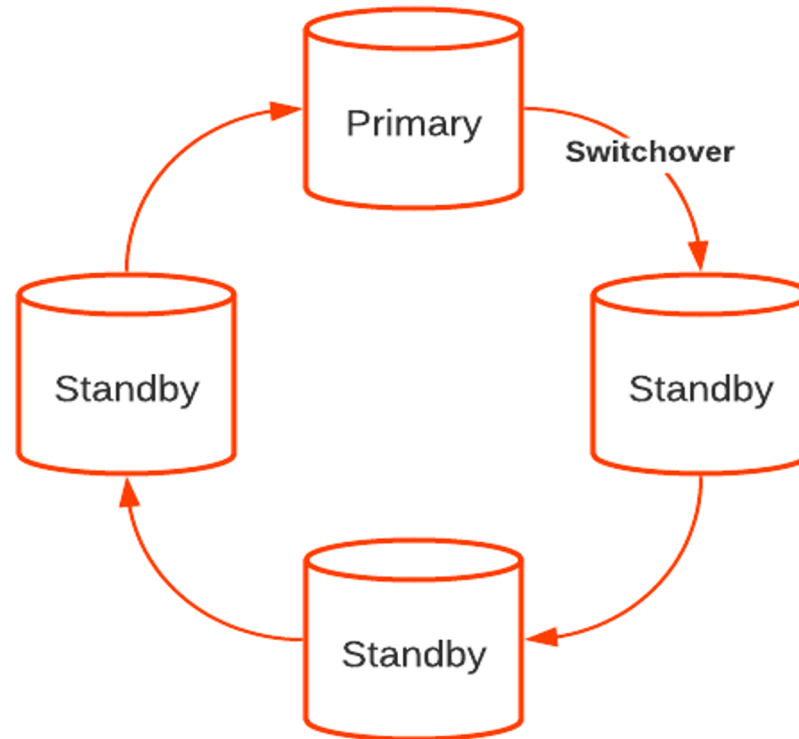
Solution Haute Disponibilité : Caractéristiques

- Virtual IP (VIP) attribuée au primaire actuel
- Monitoring du status des bases dans le cluster
- Action si faillite de la base primaire :
 - Promotion d'un standby en nouvel primaire
 - Eviter le split-brain
- Single tool or individual components



PostgreSQL Alternatives au RAC : Haute Disponibilité

Maintenance Prévû



PostgreSQL Alternatives au RAC : Haute Disponibilité

Composant : Patroni

S'appuie sur:

- Streaming Replication
- Haproxy
- yaml



PostgreSQL Alternatives au RAC : Haute Disponibilité

Framework : Patroni

“Patroni is a template for you to create your own customized, high-availability solution using Python and – for maximum accessibility – a distributed configuration store”

<https://patroni.readthedocs.io/en/latest/>



PostgreSQL Alternatives to RAC: High Availability

Architecture : Patroni & etcd

Patroni

Failover management utility

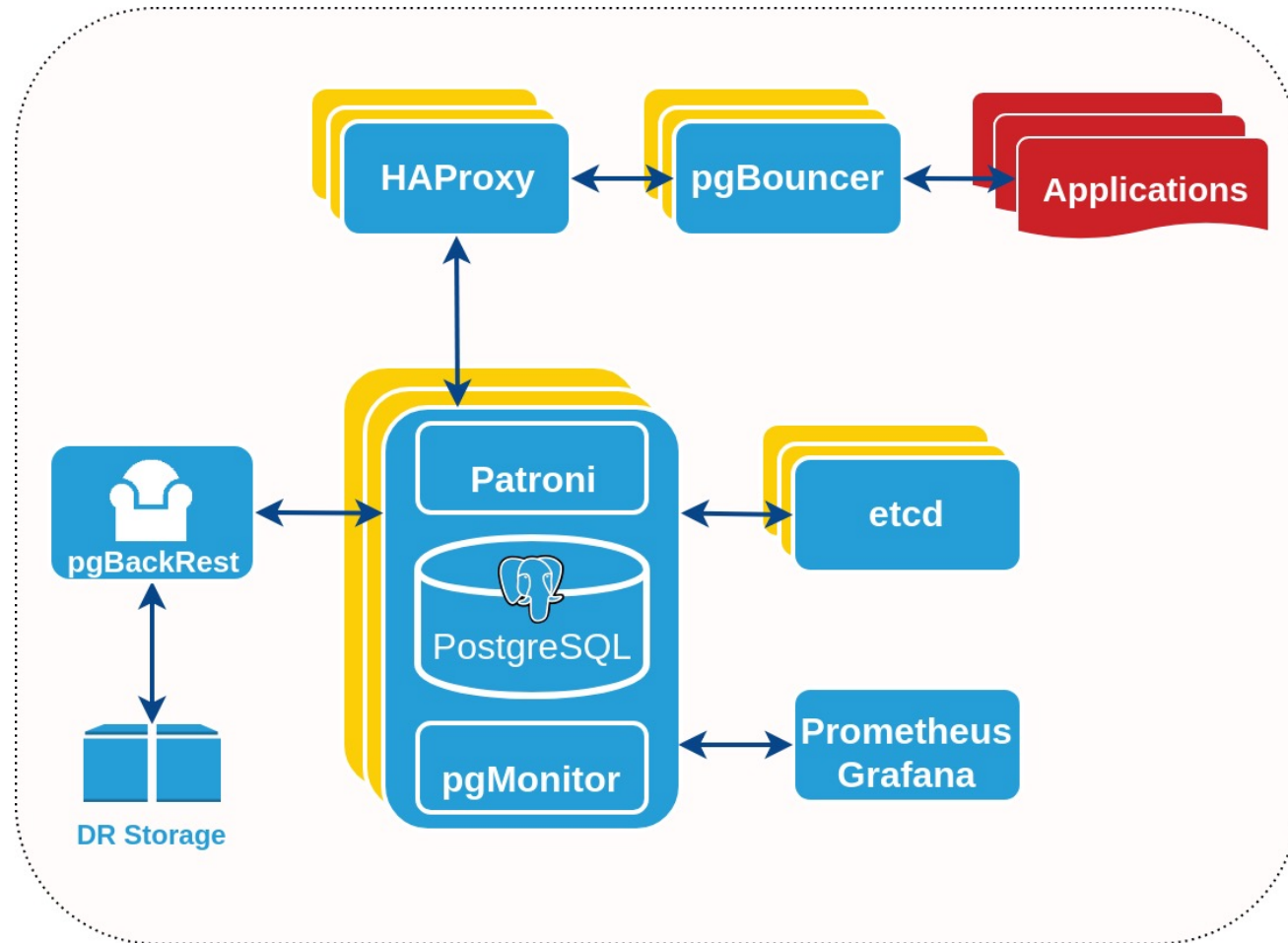
etcd

Distributed configuration store (DCS)



PostgreSQL Alternatives to RAC: High Availability

Architecture : Crunchy HA PostgreSQL



Single Data Center



PostgreSQL Alternatives au RAC : Haute Disponibilité

Framework : PAF (PostgreSQL Automatic Failover)

PAF

Cluster resource agent

Pacemaker

Cluster resource manager

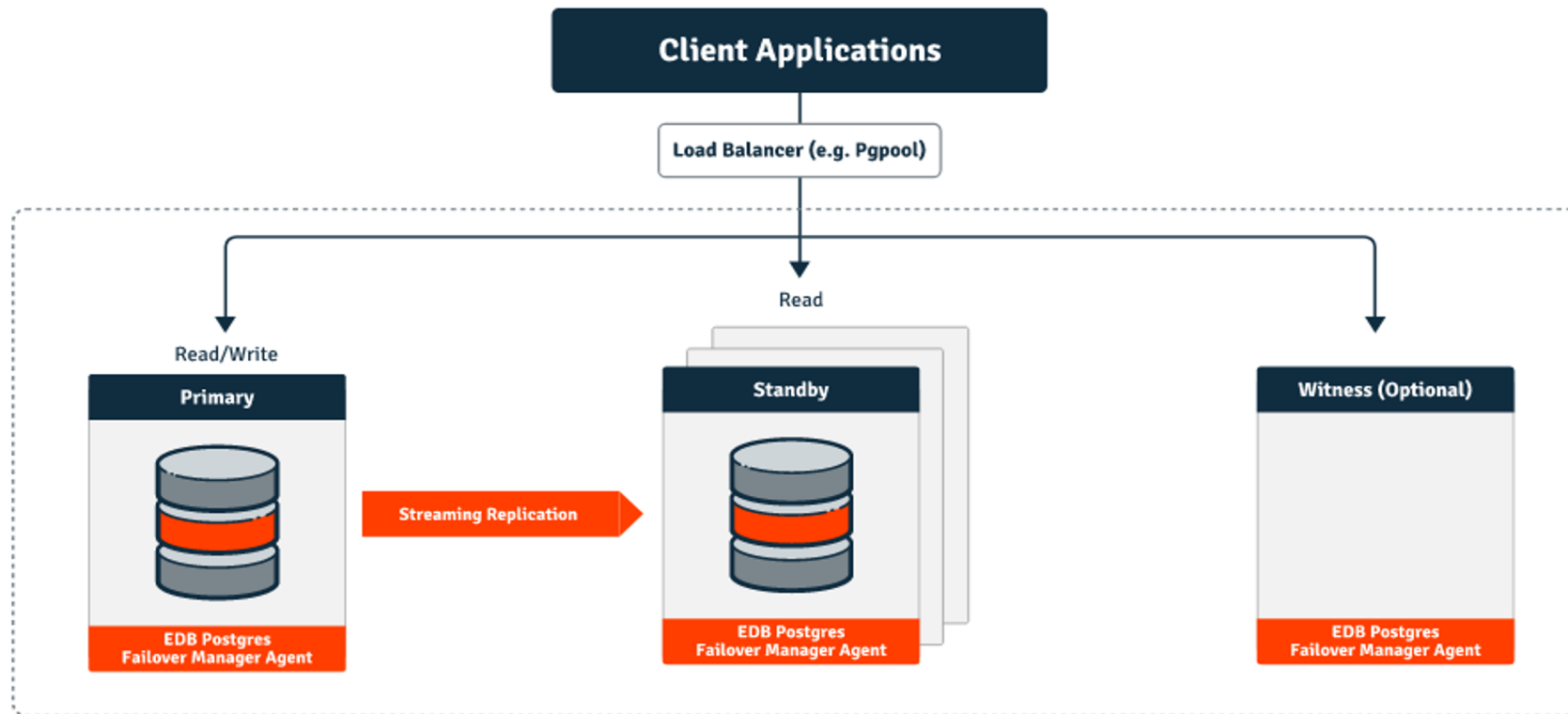
Corosync

Cluster infrastructure



PostgreSQL Alternatives au RAC : Haute Disponibilité

Outil : EFM



PostgreSQL Alternatives au RAC

Scalability / Extensibilité



PostgreSQL Alternatives au RAC

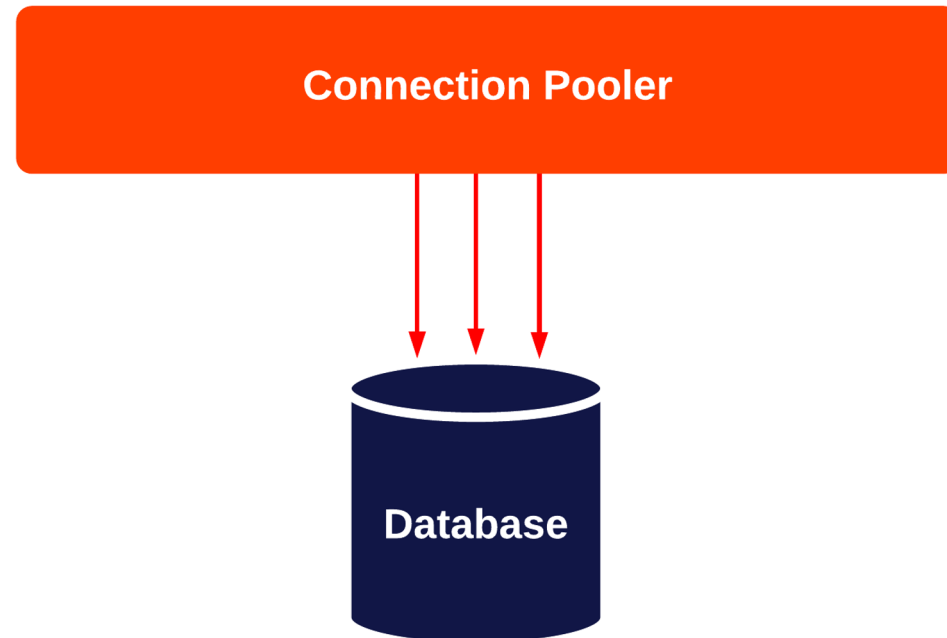
Scalability / Extensibilité

- Pooling des connexions
- Réplicas lecture-seule
- Réplication multi-master
- Load balancing
- Sharding



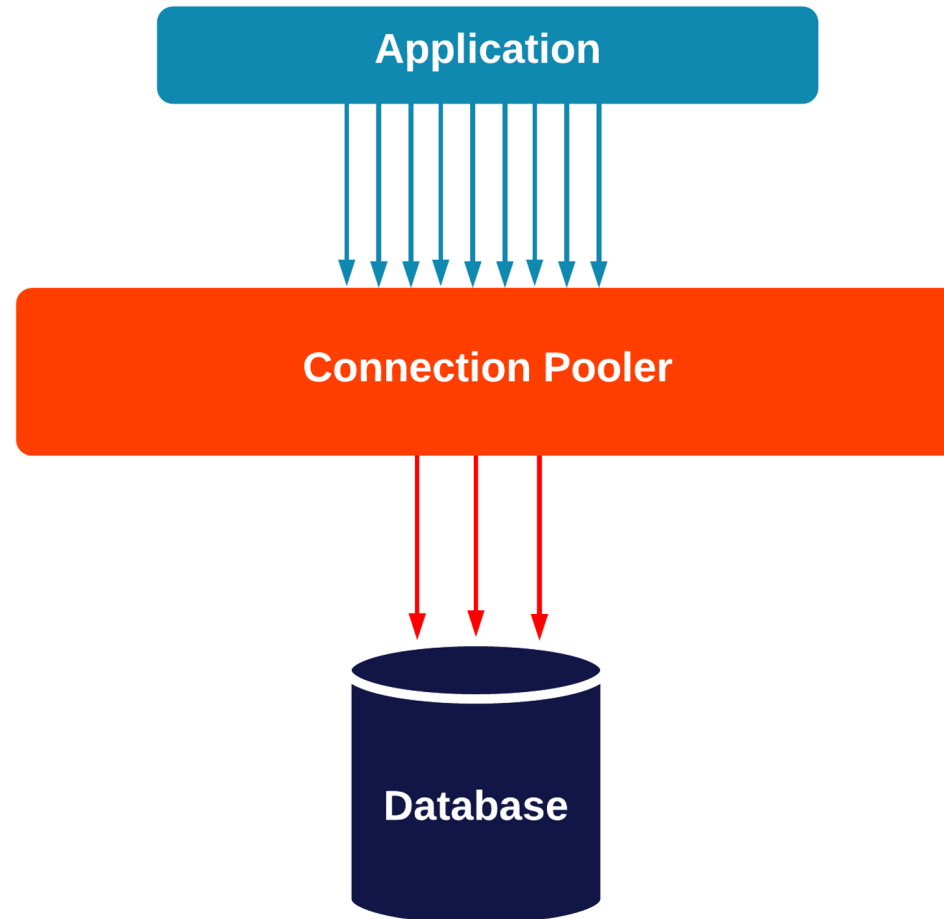
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Pooling des Connexions



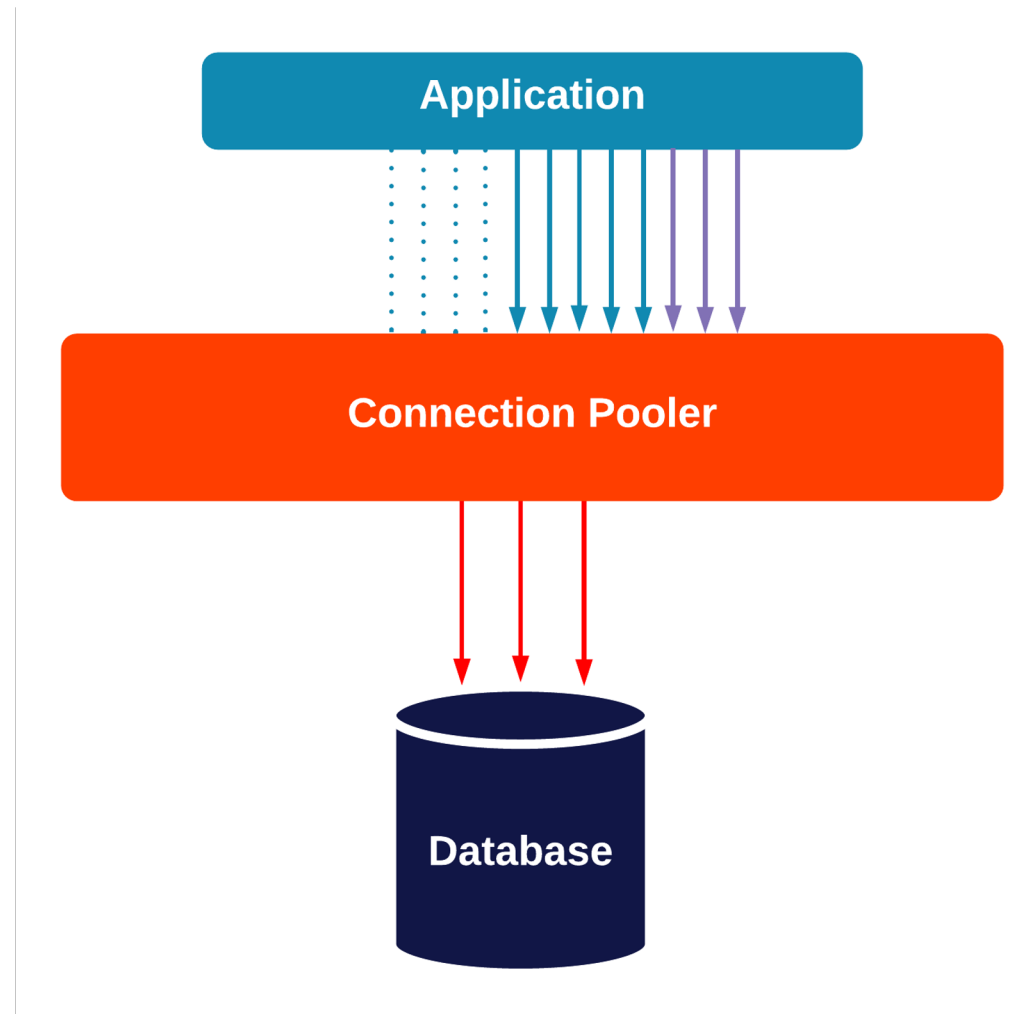
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Pooling des Connexions



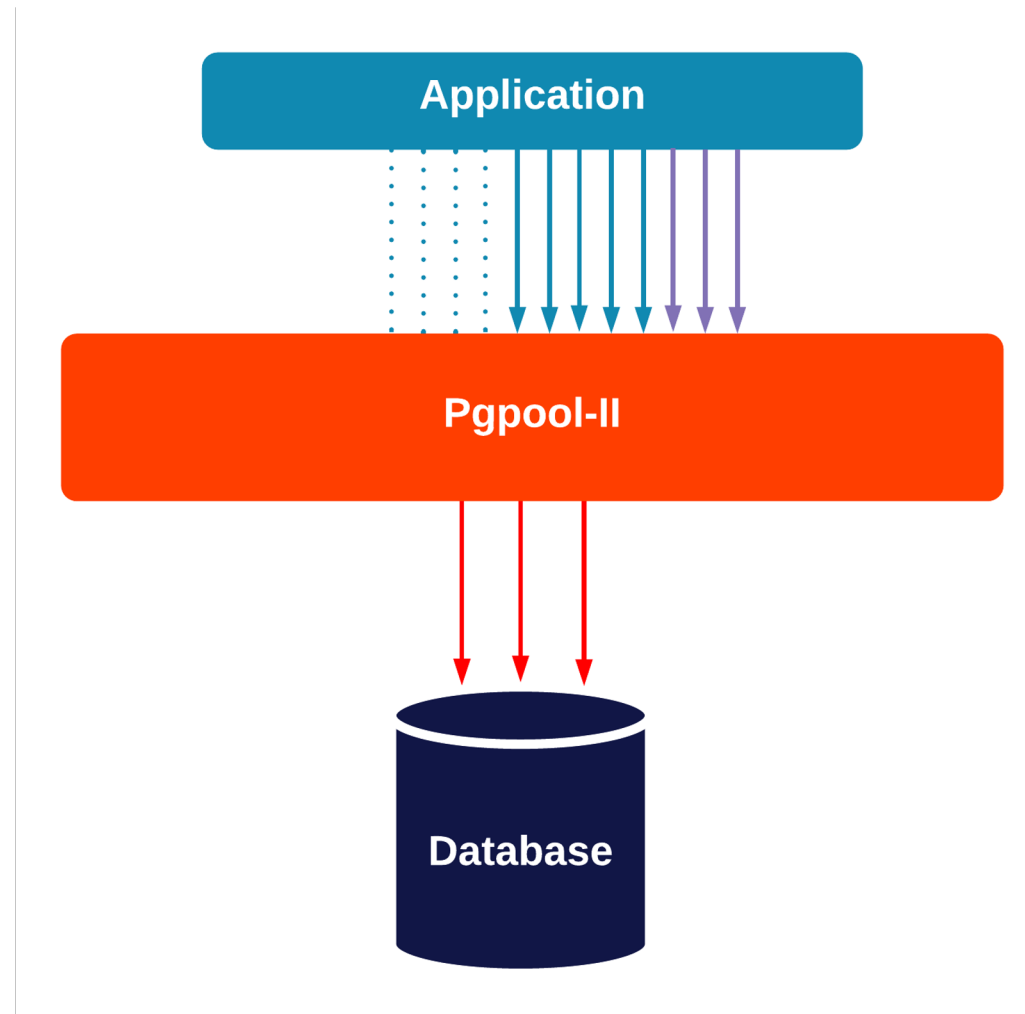
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Pooling des Connexions



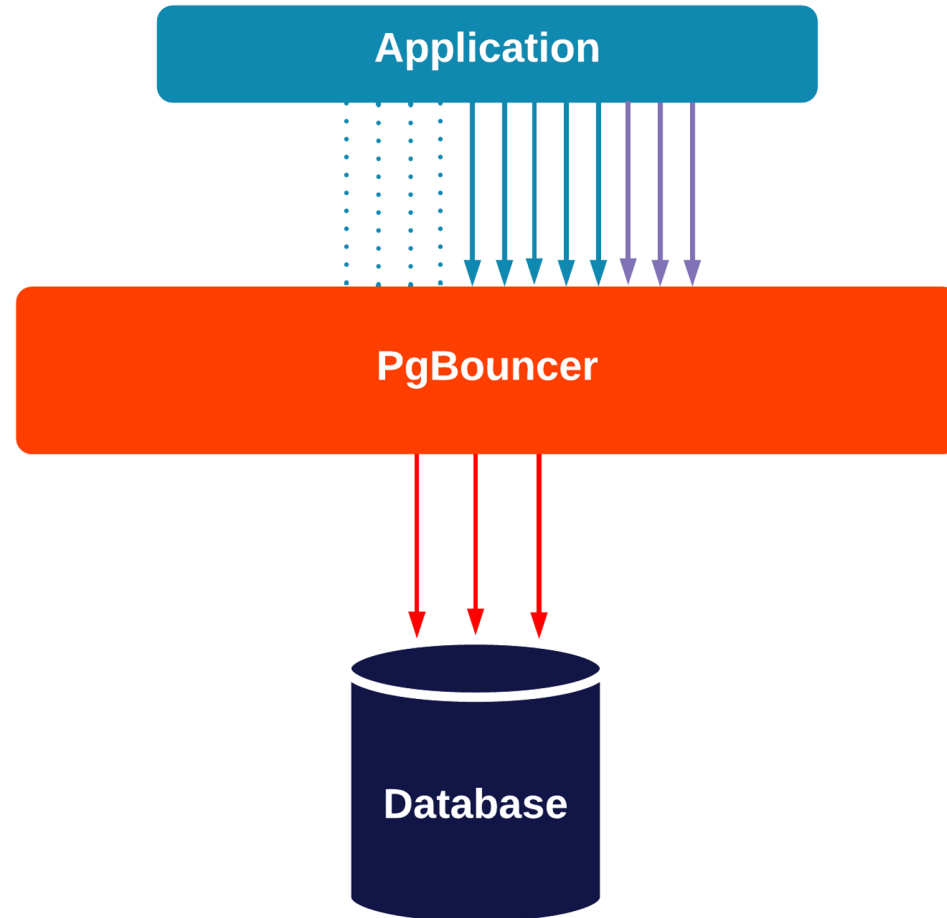
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Pooling des Connexions



PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Pooling des Connexions



PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Répliquas Lecture-Seule

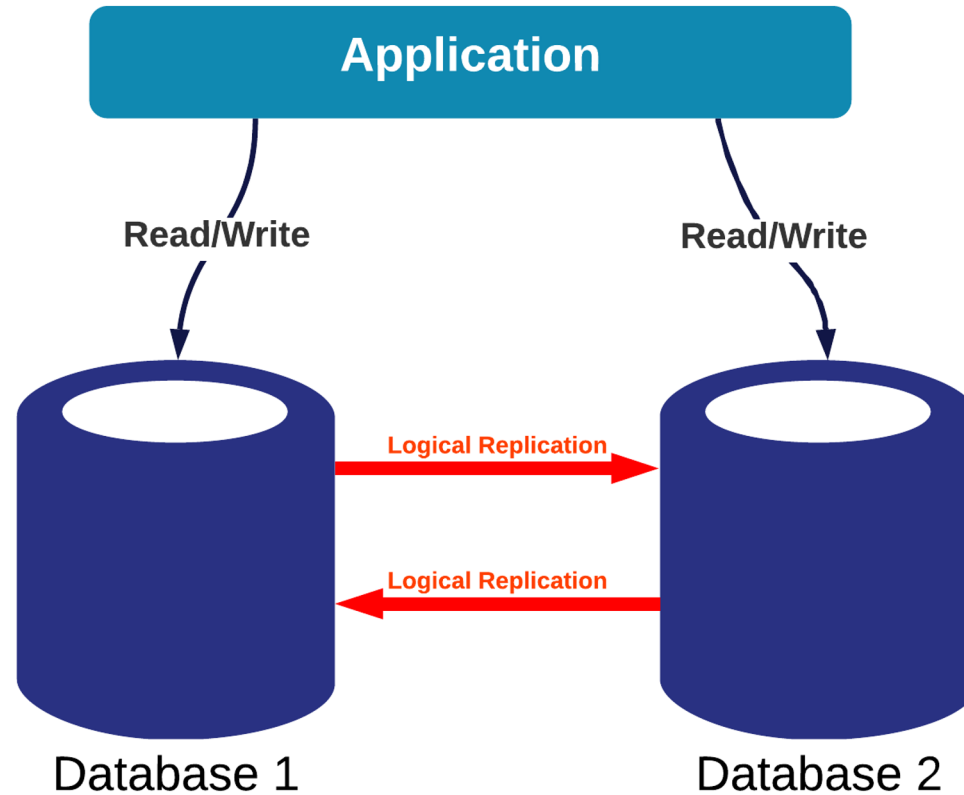


Image by [Free-Photos](#) from [Pixabay](#)



PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Réplication Multi-master



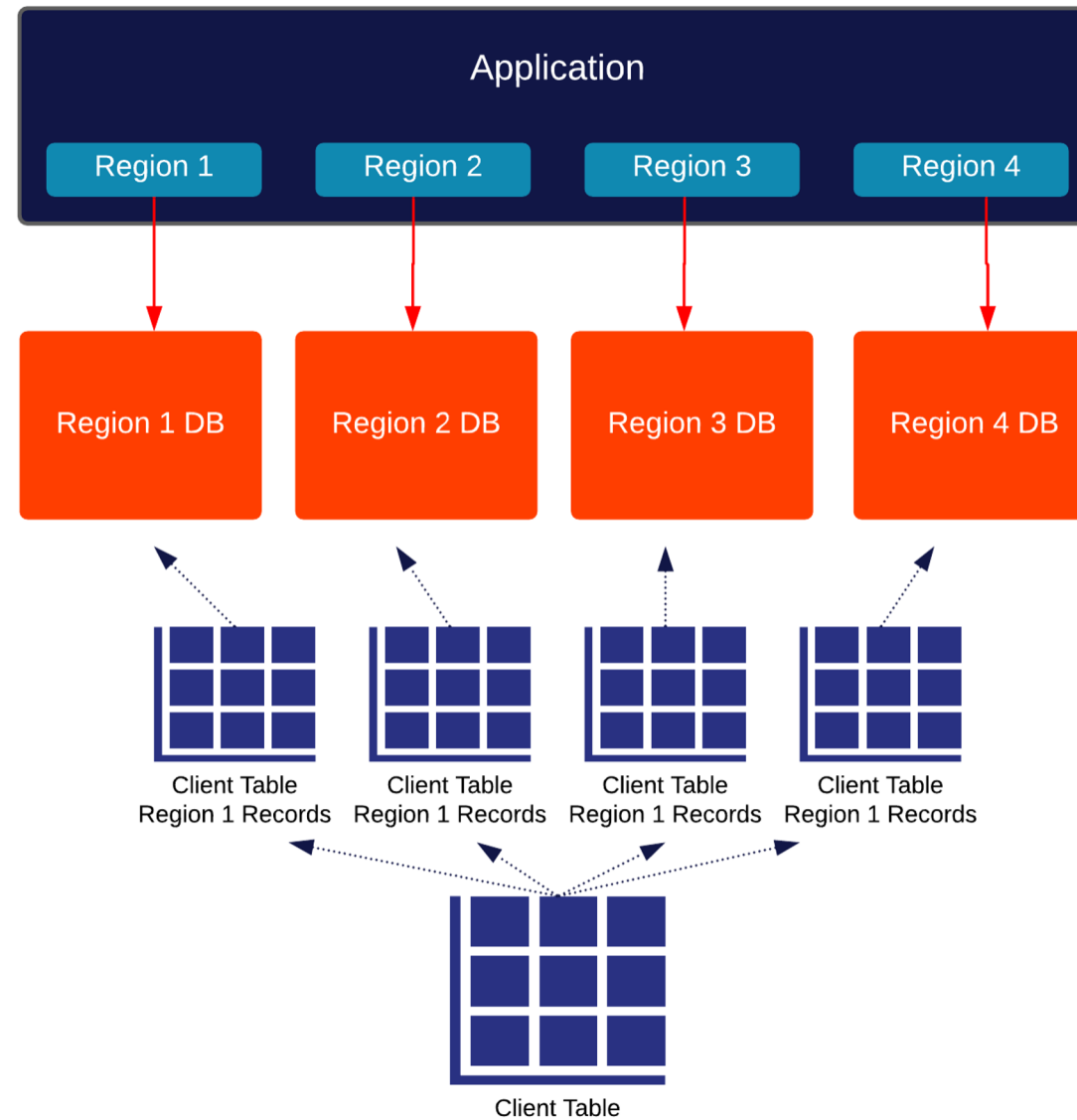
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Load Balancing



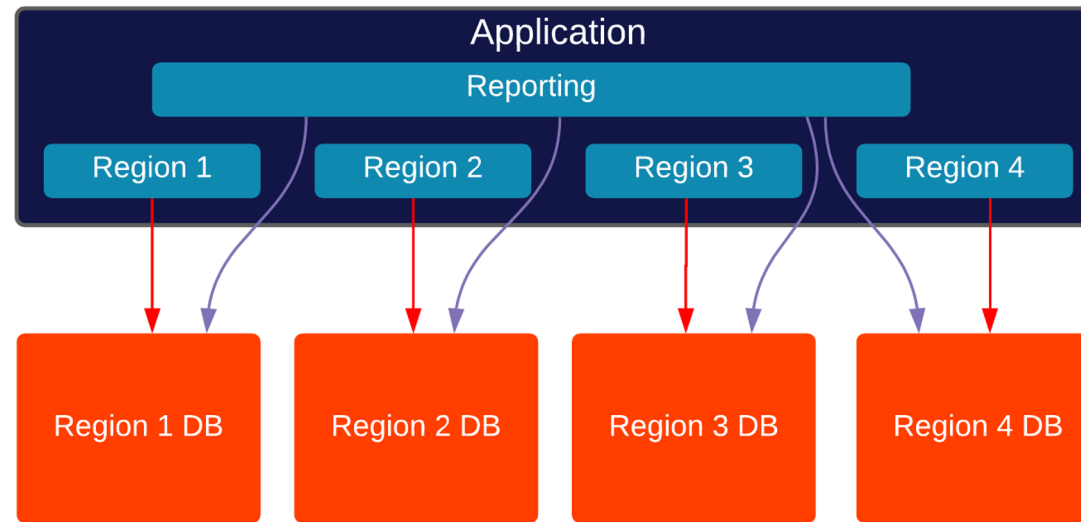
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Sharding



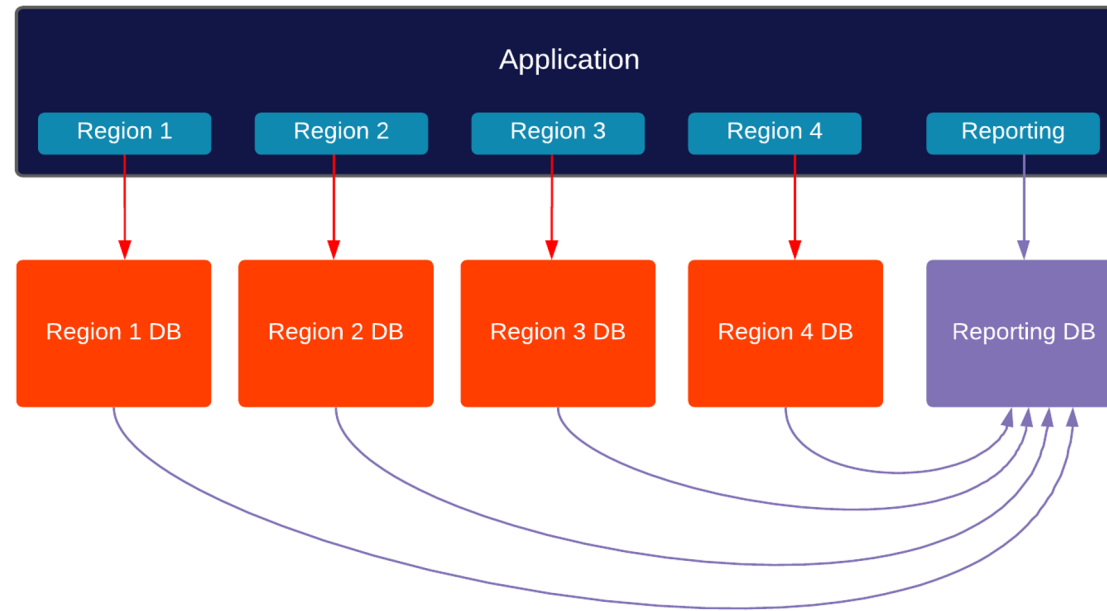
PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Sharding



PostgreSQL Alternatives au RAC : Scalability / Extensibilité

Sharding



*Dessiner une architecture
extensible et hautement disponible*



Dessiner une architecture extensible et hautement disponible

complexité des requêtes SQL

Workload

OLAP vs. OLTP

ratio lectures:écritures

RPO

TPS

on premises vs. cloud

volumétrie BDD

RTO

disponibilité pour écritures

“combien de neufs”

Disponibilité pour lectures



Dessiner une architecture extensible et hautement disponible

System Characteristics/Requirements

Workload

100 TPS
30 connections

50k TPS
20k connections

DB Size

50 GB

20TB

RPO

24 hours

Close to 0

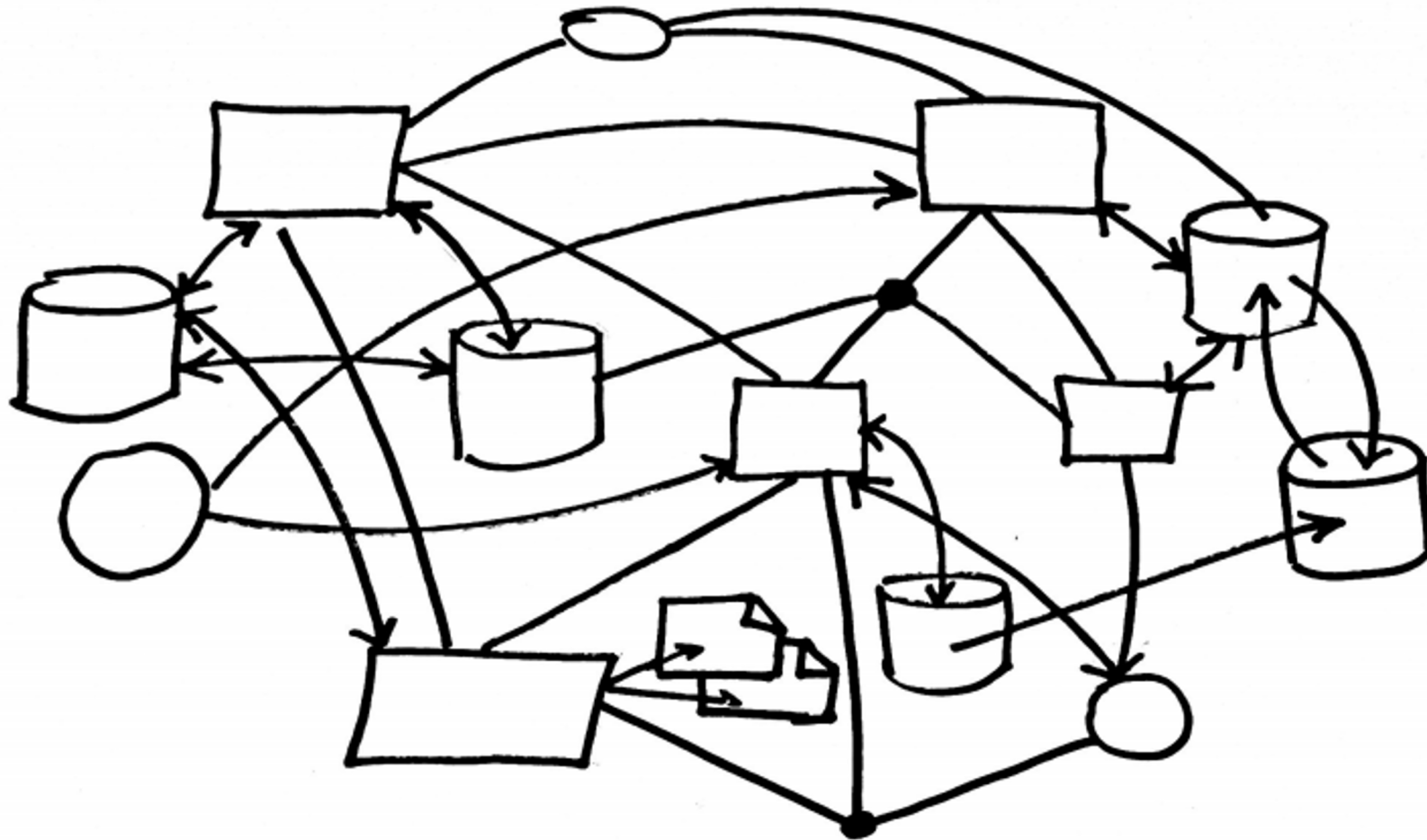
RTO

4 hours

5 seconds



Dessiner une architecture extensible et hautement disponible



Dessiner une architecture extensible et hautement disponible

System Characteristics/Requirements

Workload

100 TPS

30 connections

50k TPS

20k connections

DB Size

50 GB

20TB

RPO

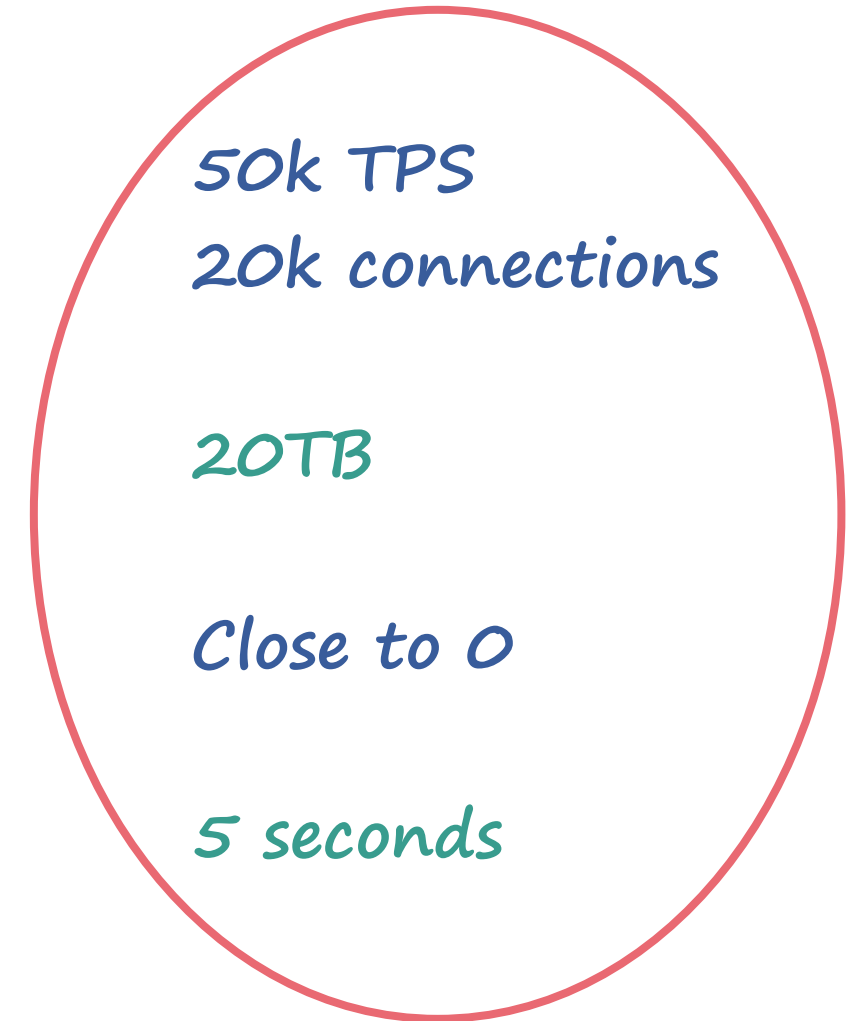
24 hours

Close to 0

RTO

4 hours

5 seconds



*Il est difficile de
Migrer depuis le RAC ?*



Migrer depuis le RAC

Considérations

- Analyse du profil de l'application
- Besoins de l'application
- Avantages vs coûts de la migration



Conclusions



Conclusions

“If you have a system that needs to be up and running a few seconds after a crash, you probably need RAC.”

ou autre architecture haute disponibilité



Conclusions

Vous pouvez probablement acheter ce qu'il vous faut, mais

“If you cannot buy a big enough system to deliver the CPU power and or memory you crave, you probably need RAC.”

ou sharding ou clustering



Conclusions

If you *need to cover your behind* politically in your organisation, you can choose to buy clusters, Oracle, RAC and what have you, and then you can safely say: “We’ve bought the most expensive equipment known to man. *It cannot possibly be our fault* if something goes wrong or the system goes down”

D'accord - peut-être !



Conclusions

“Otherwise, you probably don’t need RAC. Alternatives will usually be cheaper, easier to manage and quite sufficient.”



Merci !

Slides: karenjex.blogspot.com

[@karenhjex](https://twitter.com/karenhjex) | karen.jex@crunchydata.com



Questions ?

