

Building a Redundant PostgreSQL Cluster for Zabbix Using Patroni

Thomas Oftring

Rhein-Main Solutions GmbH



ZABBIX '26
CONFERENCE
GERMANY



Wieso, weshalb, warum

... ist doch NUR das Monitoring ...



ZABBIX '26

CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

(...)

*... ist doch NUR das
Monitoring ...*

Blind Spot

Keine Monitoringdaten bei Ausfall
der Infrastruktur zum
Troubleshooting

SLA Berechnungsgrundlage

Datenlücken verfälschen die
Berechnung von SLAs

Datenintegrität & Kontinuität

Datenlücken bei Ausfall der Zabbix
Datenbank

Maintenance Downtime

Monitoring kann nicht ohne
Downtime und Datenlücken
gepatched werden

ZABBIX '26

CONFERENCE

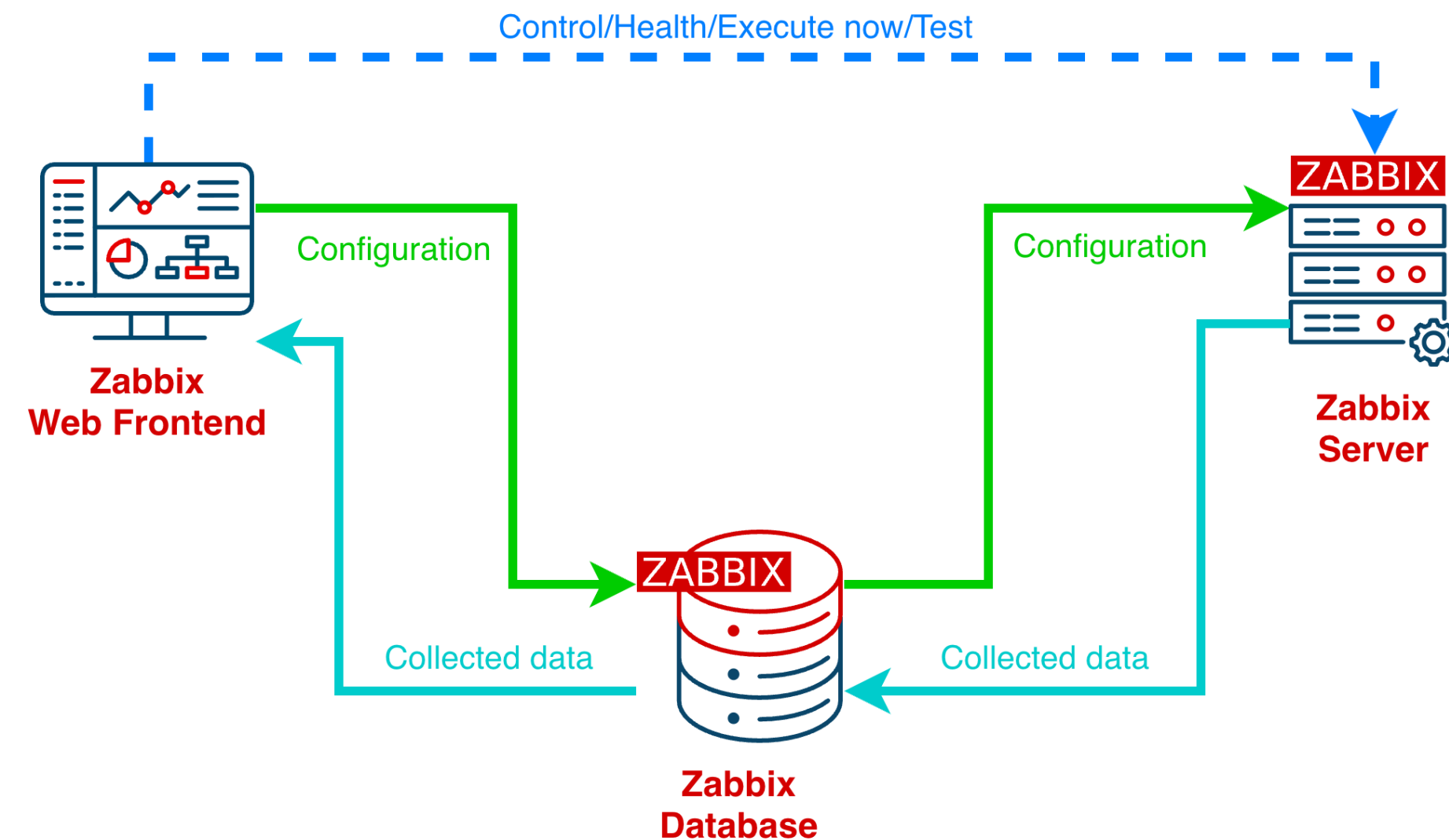
GERMANY

Datenbank HA bei Zabbix

...ist das denn überhaupt notwendig?

 RHEIN-MAIN
SOLUTIONS

Zabbix Single Server



Zabbix Server

- Konfigurationssynchronisation
- Speichern der gesammelten Daten/Trends
- Trigger Evaluierung
- Alerts/Events
- Maintenances Aktivierung/Deaktivierung
- Services/SLA

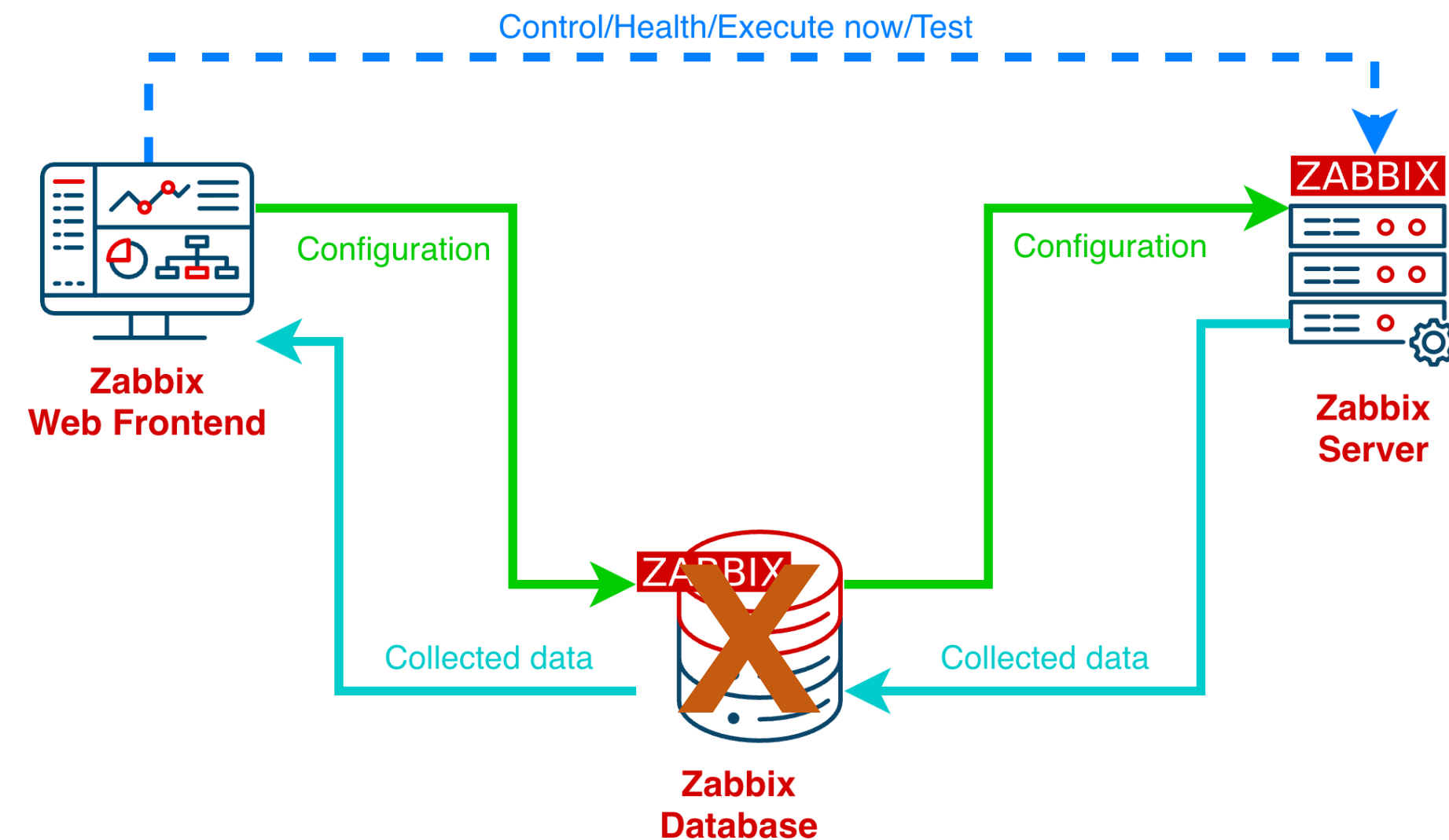
Zabbix Web Frontend

- Visualisierung gesammelter Daten
- Visualisierung von Problemen
- Konfigurationsänderungen
- Bereitstellung der API

Zabbix Proxies (falls vorhanden)

- Konfigurationssynchronisation
- Datensammlung
- Datenpufferung

Zabbix Single Server



Zabbix Server

Konfigurationssynchronisation

Speichern der gesammelten Daten/Trends

Trigger-Evaluierung

Alerts/Events Alarm Datenbank Down (wenn aktiv)

Maintenances-Aktivierung/Deaktivierung

Services/SLA

Zabbix Web Frontend

Visualisierung gesammelter Daten

Visualisierung von Problemen

Konfigurationsänderungen

Bereitstellung der API

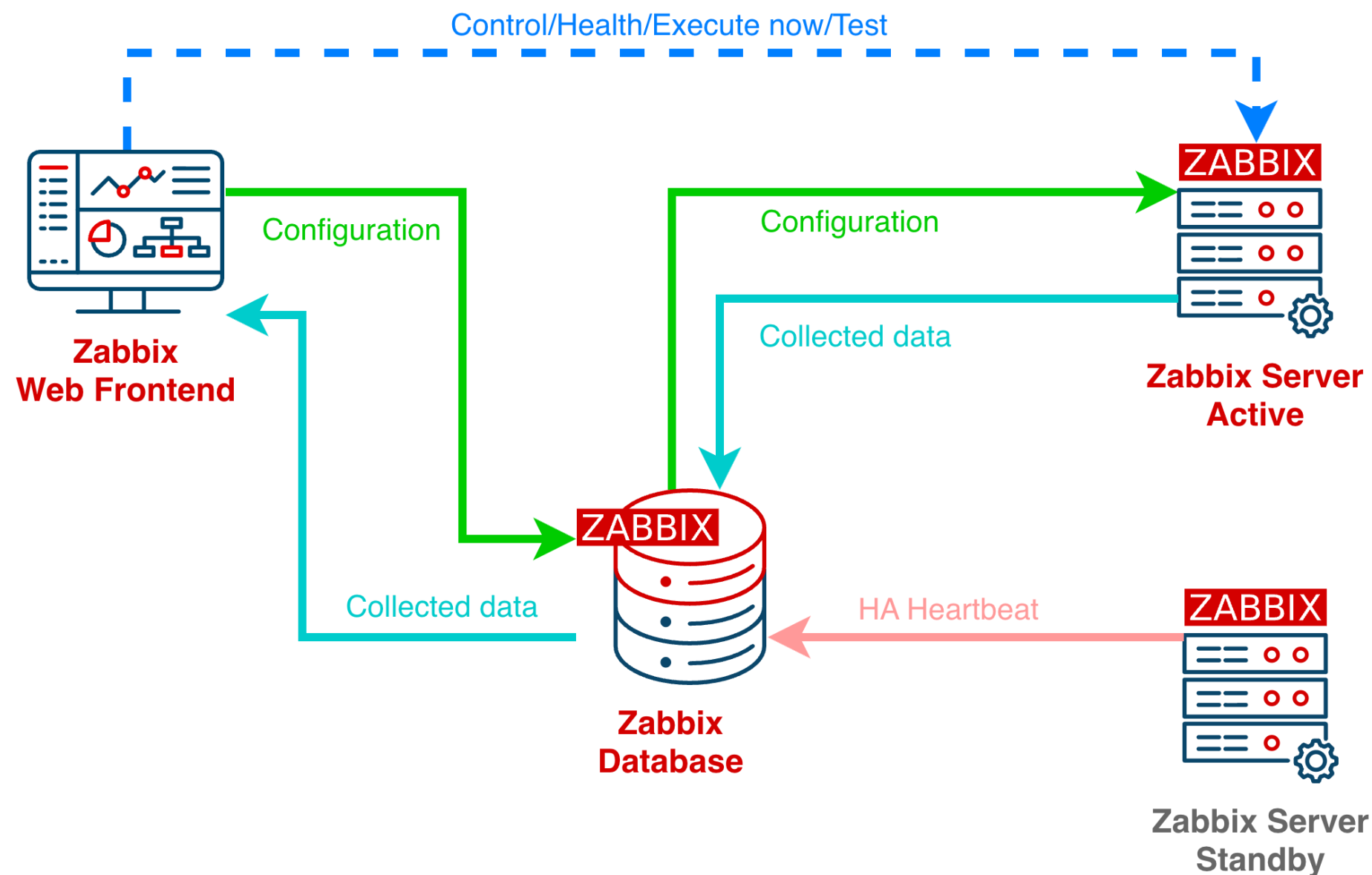
Zabbix Proxies (falls vorhanden)

Konfigurationssynchronisation Letzte Konfiguration wird benutzt

Datensammlung

Datenpufferung

Zabbix Server HA Setup



Zabbix Server

- Konfigurationssynchronisation
- Speichern der gesammelten Daten/Trends
- Trigger Evaluierung
- Alerts/Events
- Maintenances Aktivierung/Deaktivierung
- Services/SLA

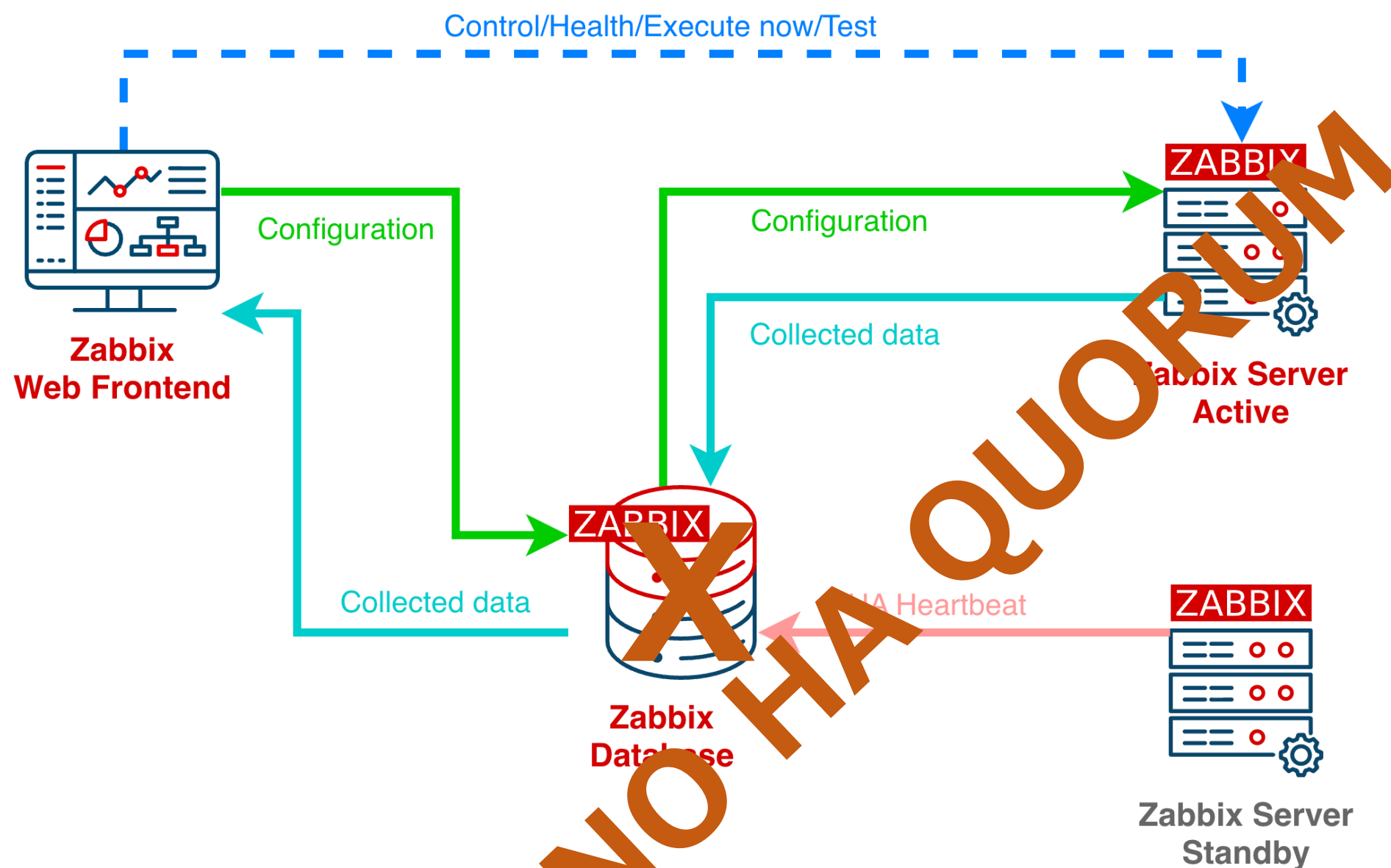
Zabbix Web Frontend

- Visualisierung gesammelter Daten
- Visualisierung von Problemen
- Konfigurationsänderungen
- Bereitstellung der API

Zabbix Proxies (falls vorhanden)

- Konfigurationssynchronisation
- Datensammlung
- Datenpufferung

Zabbix Server HA Setup



Zabbix Server

Konfigurationssynchronisation

Speichern der gesammelten Daten/Trends

Trigger-Evaluierung

Alerts/Events Alarm Datenbank Down (wenn aktiv)

Maintenances-Aktivierung/Deaktivierung

Services/SLA

Zabbix Web Frontend

Visualisierung gesammelter Daten

Visualisierung von Problemen

Konfigurationsänderungen

Bereitstellung der API

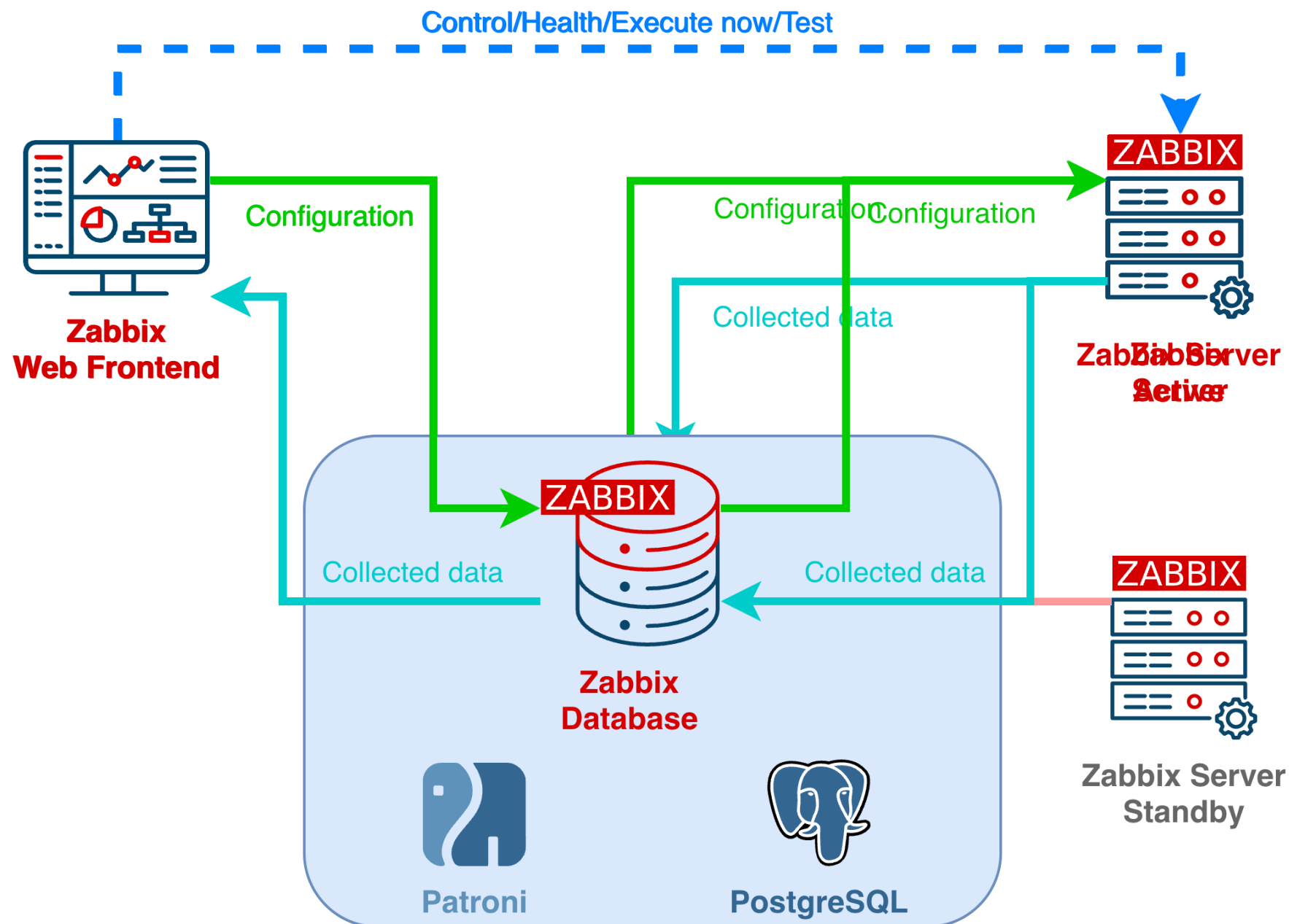
Zabbix Proxies (falls vorhanden)

Konfigurationssynchronisation Letzte Konfiguration wird benutzt

Datensammlung

Datenpufferung

Zabbix Single Server/Zabbix HA



Zabbix Server

- Konfigurationssynchronisation
- Speichern der gesammelten Daten/Trends
- Trigger Evaluierung
- Alerts/Events
- Maintenances Aktivierung/Deaktivierung
- Services/SLA

Zabbix Web Frontend

- Visualisierung gesammelter Daten
- Visualisierung von Problemen
- Konfigurationsänderungen
- Bereitstellung der API

Zabbix Proxies (falls vorhanden)

- Konfigurationssynchronisation
- Datensammlung
- Datenpufferung



Absicherung vor ...

Patroni und PostgreSQL

ZABBIX '26

CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS



Schutz bei Hardwareausfällen/Systemausfällen

Server Ausfall

Hardware Defekt (Bare Metal)

Ausfall VM (HyperVisor)

Stromausfall

Speicherfehler

Storage Ausfall

Ausfall HDD/SSD

Ausfall SAN/NAS

Korruptes Dateisystem

Storage voll (Gefahr bei Überprovisionierung)

Netzwerk Ausfall

Switch/Router Defekt

SFP/Kabel Defekt

Routing Probleme

DNS Probleme

Firewall/DoS Systeme



Schutz bei Softwareproblemen

Betriebssystem Probleme

Kernel Panic

System hängt sich auf

System überlastet

Fehlkonfiguration nach Updates

PostgreSQL Probleme

OOM Kill eines PostgreSQL Prozesses

Segfaults

Zugriffsprobleme auf Dateisystem

Zugriffsprobleme auf Ressourcen

Menschliche Fehler/KI Fehler

Fehlbedienung

Rollout fehlerhafter Konfiguration

Falsche/Fehlerhafte Skripte

Versehentliches Löschen/Aktualisieren von Daten



Schutzmechanismen

Patroni und PostgreSQL

ZABBIX '26

CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS



High Availability & Auto Failover

Patroni als Wächter/Koordinator

Überwacht Leader

Überwacht alle Standby Knoten

Rest API zur Steuerung

Rest API zur Status Überwachung

Auto Failover

Promotion eines Standby bei Ausfall des Leaders

Quorum Bildung über DCS

Betriebssystem Watchdog nutzbar

Fencing integriert in Patroni

Clusterverwaltung in DCS

Getrennt von der Datenbankumgebung

Ermöglicht HA mit nur 2 Datenbankknoten

Quorum wird über DCS gebildet (Consul/etcd/...)



Schutz vor Datenverlust

Patroni

Synchronous Replikation nutzen/fordern

Prüfung und Verwaltung der Replication Slots

Prüfung der WAL Lags

Berücksichtigung der WAL Lags bei Auto Failover

Failover nur auf Standby der Synchron ist

Failover auf Standby mit aktuellsten

Datenstand

Failover nur wenn Maximum Lag nicht
überschritten

Kein Failover auf ausgeschlossene Standby
PostgreSQL

Synchronous Replikation

Bereitstellung WAL (Write Ahead Log)

Bereitstellung Replication Slots

Einspielen der WAL Pakete auf den Standby

Knoten



Schutz der Datenintegrität

Patroni

Single Source of Truth (Konfiguration im DCS)

Leader Locking im DCS

Automatisches Demoting von Leader Knoten

Patroni stellt Resync von Standby Knoten Sicher

Fencing mit Watchdog (Linux-Watchdog)

Replication Lag Check

Maximum Lag Threshold kann berücksichtigt werden

Synchronous Replication kann erzwungen werden

PostgreSQL

Daten Checksummen erkennen I/O Fehler auf Block Level

pg_rewind nutzt diese Checksummen

WAL (Write-Ahead Log) Shipping

Replication Slots



Backup & Recovery

Wasserfall Modell möglich

Leader

Standby mit aktuellem Stand (No Lag)

Standby mit Zeitversatz (z. B. 1 Stunde)

Kann vom AutoFailover ausgeschlossen werden

Schnelles POIT Recovery möglich

Restore/Reinit eines Standby Knoten

Automatischer Restore/Reinit

Manueller Restore/Reinit

Standardmäßig Kombination

pgbasebackup/pg_rewind

Kombinierbar mit pgbackrest und Failback auf Standard

Restore des kompletten Clusters

Kein Schutz in Patroni vorhanden

Einbindung von pgbackrest notwendig

Kann Nativ in Patroni Konfiguration genutzt werden



Architektur Überblick

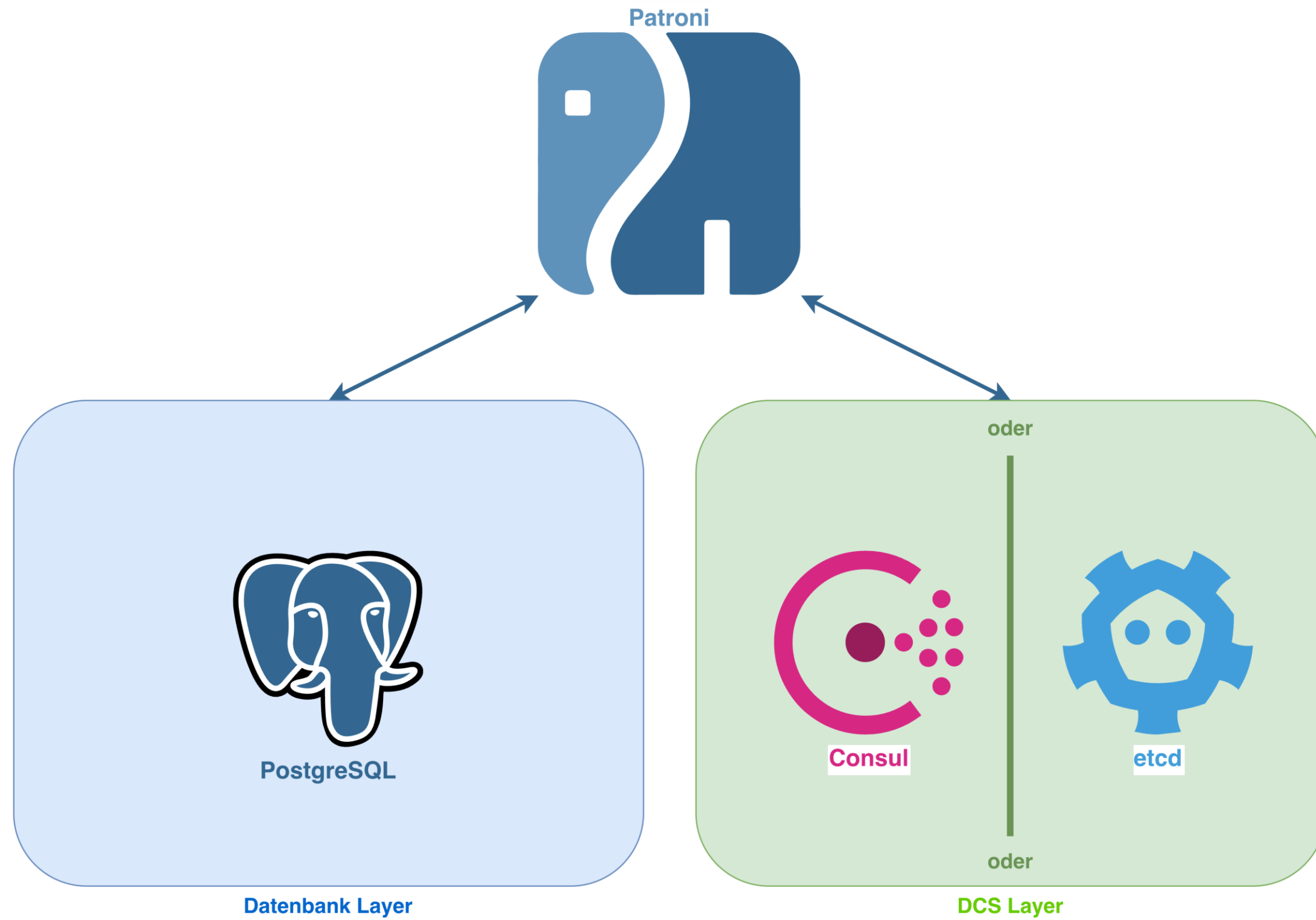
Patroni und PostgreSQL

ZABBIX '26

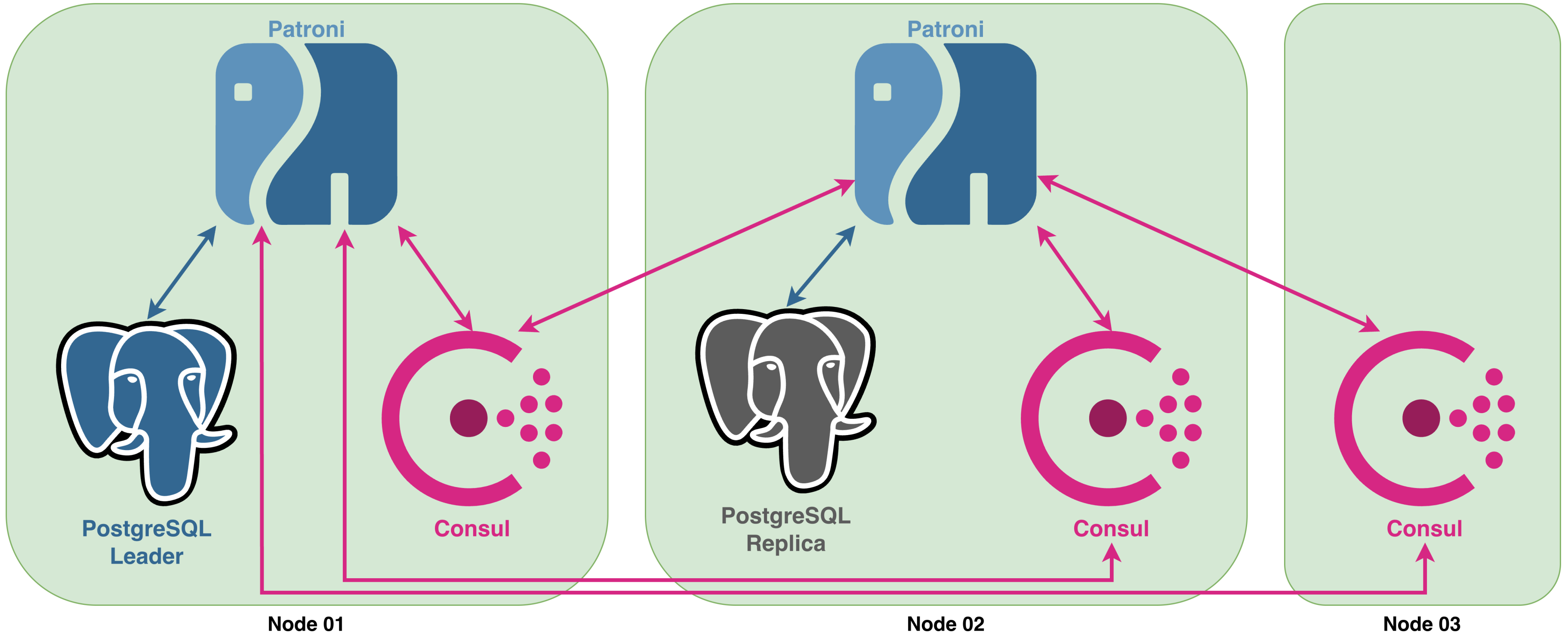
CONFERENCE

GERMANY

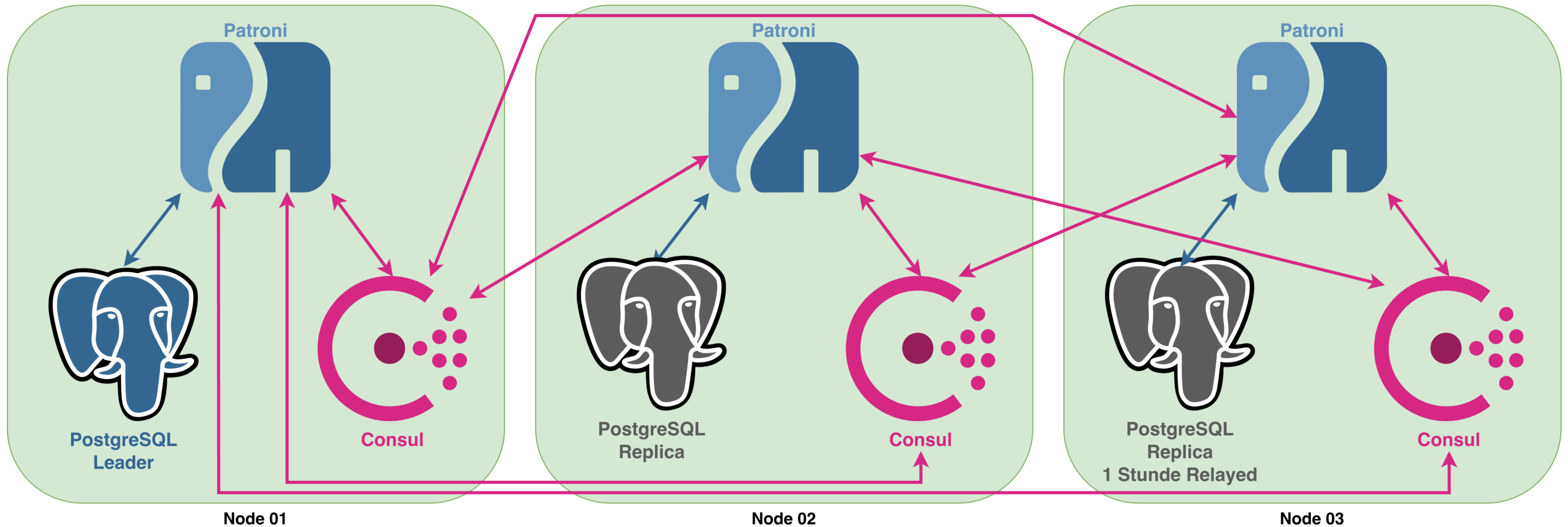
 RHEIN-MAIN
SOLUTIONS



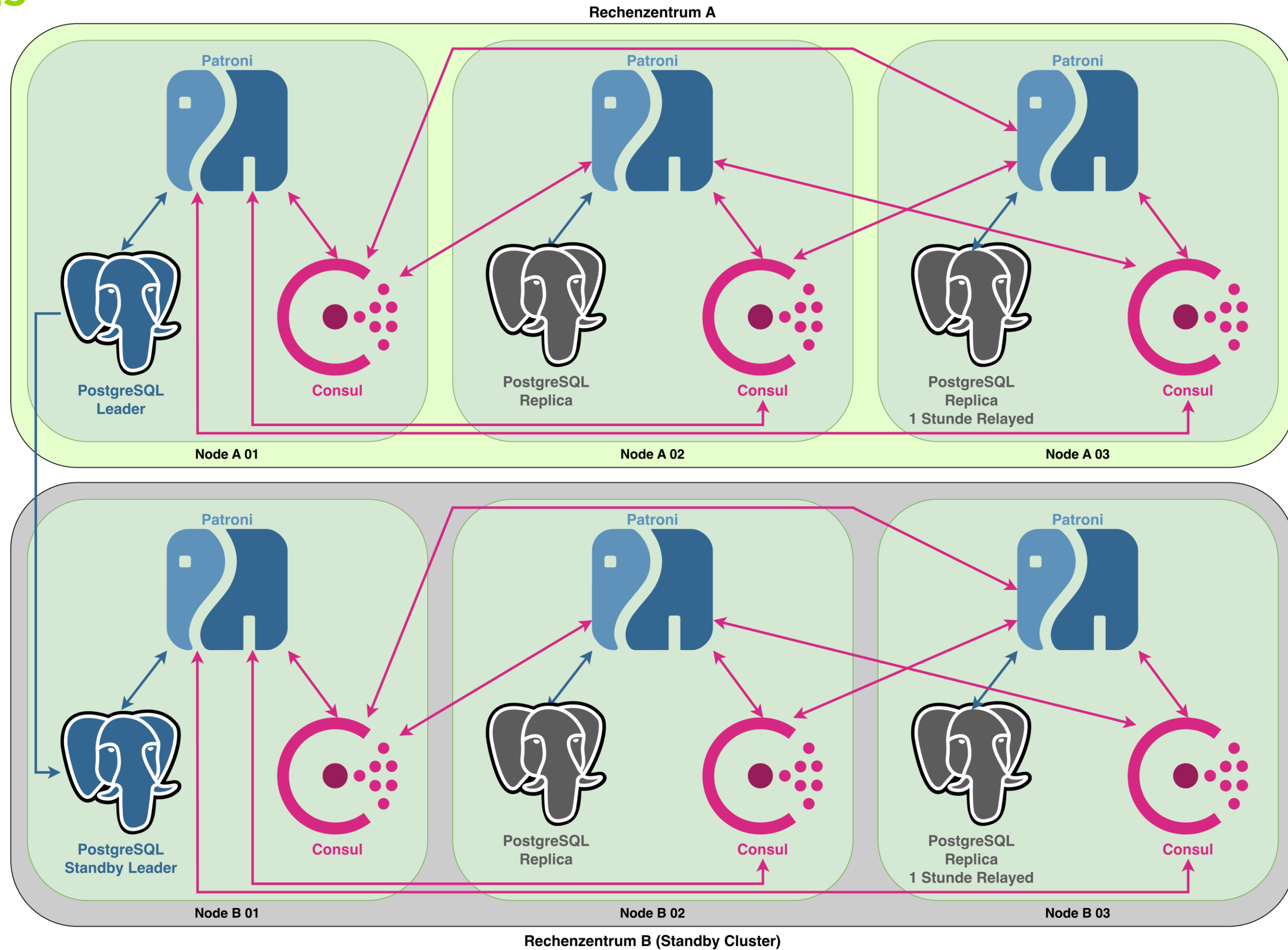
Trennung Datenbank und DCS



Patroni mit 2 PostgreSQL Knoten
Mindest Ausbau



**Patroni mit 3 PostgreSQL Knoten
Replica mit 1 Stunde Zeitversatz**

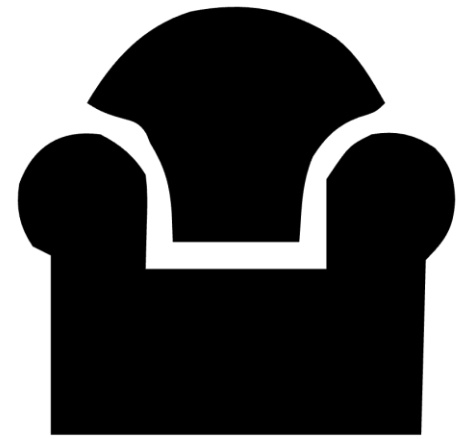


```
patronictl -c /etc/patroni/config.yml list

+ Cluster: datacenter_a_frankfurt (5493148990349154283) -----+
| Member | Host | Role | State | TL | Lag in MB |
+-----+-----+-----+-----+---+-----+
| fradb01 | 172.16.160.201 | Leader | running | 2 | |
| fradb02 | 172.16.160.202 | Replica | streaming | 2 | 0 |
| fradb03 | 172.16.160.203 | Replica | streaming | 2 | 0 |
+-----+-----+-----+-----+---+-----+

+ Cluster: datacenter_b_dublin (5493148990349154283) ---+-----+-----+
| Member | Host | Role | State | TL | Lag in MB |
+-----+-----+-----+-----+---+-----+
| dubdb01 | 172.16.170.201 | Standby Leader | streaming | 2 | |
| dubdb02 | 172.16.170.202 | Replica | streaming | 2 | 0 |
| dubdb03 | 172.16.170.203 | Replica | streaming | 2 | 0 |
+-----+-----+-----+-----+---+-----+
```

Beispiel Status mit Standby Datacenter



Backup mit pgBackRest

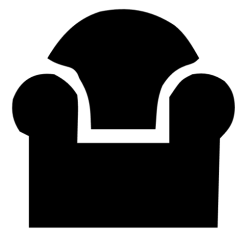
... braucht man das wirklich?

ZABBIX '26

CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS



Was ist pgBackRest



Allgemeine Infos

Moderne Backup Lösung für PostgreSQL

Open Source (MIT License)

Optimiert für große PostgreSQL Umgebungen

Native Integration in PostgreSQL WAL Archivierung

Performance und Skalierbarkeit

Parallele Backup Prozesse (Multi Core)

Parallele Restore Prozesse (Multi Core)

Deduplizierung & Komprimierung

Voll/Inkrementell/Differentiell Backups

Datensicherung

Mehrere gleichzeitige Backup Repositories

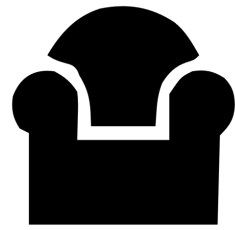
Unterschiedliche Repository Typen

- Lokales Dateisystem

- S3-Kompatibel (AWS S3, Minio, RustFS, Ceph, etc.)

- Azure Blob Storage/Google Cloud Storage

- SETP



Enterprise Features in pgBackRest

Allgemeine Infos

Automatische Backup Bereinigung (Retention Management)

Backup Validierung über Checksummen

Monitoring JSON/Prometheus Output

Sicherheit & Compliance

TLS Verschlüsselung für Netzwerkverbindungen

Data-At-Rest Verschlüsselung der gesicherten Daten (AES-256)

Key Management – Anbindung an externe KMS-Systeme

Unterschiedliche Keys für unterschiedliche Repository

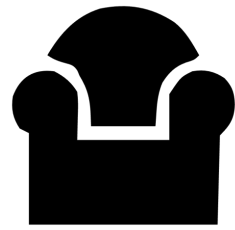
Eingebaute Health Checks zur Backup Validierung

Self Healing für fehlerhafte WAL

Retention Management berücksichtigt Backup Status

PITR (Point in Time Recovery)

Datenintegrität



Vorteile pgBackRest und HA

Integration in HA Architektur

Zentrale Backup Konfiguration (Patroni Bootstrap)

Automatische WAL Archivierung auf allen Knoten

Backup von Replica ohne extra Last auf Leader

Einbindung in PostgreSQL Archiv modus

archive_command

restore-command

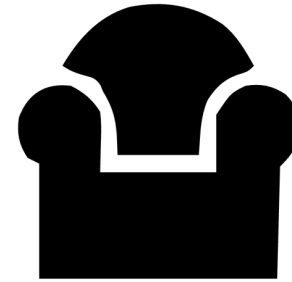
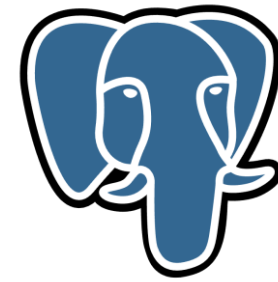
Desaster Recovery für gesamten Cluster

PITR (Point In Time Recovery) für den Cluster

Rebuild von Replicas aus Backup anstatt vom Leader

Rebuild von nur fehlenden/beschädigten Daten

Cross Datacenter Backups möglich



...Demo Time...



ZABBIX '26

CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

```
demo@zgc2026n01:~# sudo -u postgres patronictl -c /etc/patroni/config.yml list
+ Cluster: pg-ha-cluster (7627195140299511853) ----+-----+-----+-----+-----+-----+-----+
| Member                | Host                | Role   | State   | TL | Receive LSN | Lag | Replay LSN | Lag |
+-----+-----+-----+-----+-----+-----+-----+-----+
| zgc2026n01.example.edu | 10.42.57.161        | Leader | running | 4  |              |     |             |     |
| zgc2026n02.example.edu | 10.42.57.162        | Replica | streaming | 4  | 1/293A7958   | 0   | 1/293A7958 | 0   |
```

Beispiel Status mit Standby Datacenter

```

demo@zgc2026n01:~# sudo -u postgres patronictl -c /etc/patroni/config.yml switchover pg-ha-cluster
Current cluster topology
+ Cluster: pg-ha-cluster (7627195140299511853) ----+-----+-----+-----+-----+-----+
| Member                | Host                | Role   | State   | TL | Receive LSN | Lag | Replay LSN | Lag |
+-----+-----+-----+-----+-----+-----+-----+-----+
| zgc2026n01.example.edu | 10.42.57.161        | Leader | running | 4  |              |     |             |     |
| zgc2026n02.example.edu | 10.42.57.162        | Replica | streaming | 4  | 1/2944EAD8   | 0   | 1/2944EAD8   | 0   |
+-----+-----+-----+-----+-----+-----+-----+-----+
Primary [zgc2026n01.example.edu]:
Candidate ['zgc2026n02.example.edu'] []: zgc2026n02.example.edu
When should the switchover take place (e.g. 2026-04-24T18:56 ) [now]: now
Are you sure you want to switchover cluster pg-ha-cluster, demoting current leader zgc2026n01.example.edu? [y/N]: y
2026-04-24 17:56:48.67942 Successfully switched over to "zgc2026n02.example.edu"
+ Cluster: pg-ha-cluster (7627195140299511853) ----+-----+-----+-----+-----+-----+
| Member                | Host                | Role   | State   | TL | Receive LSN | Lag | Replay LSN | Lag |
+-----+-----+-----+-----+-----+-----+-----+-----+
| zgc2026n01.example.edu | 10.42.57.161        | Replica | stopped |    | unknown     |     | unknown     |     |
| zgc2026n02.example.edu | 10.42.57.162        | Leader | running | 4  |              |     |             |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

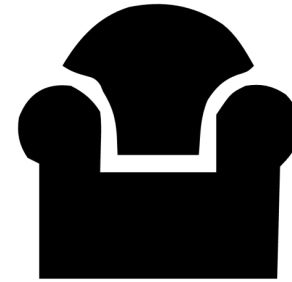
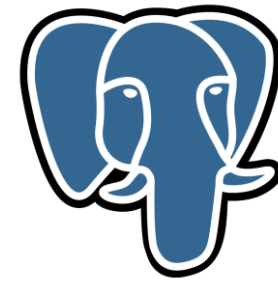
demo@zgc2026n01:~# sudo -u postgres patronictl -c /etc/patroni/config.yml list
+ Cluster: pg-ha-cluster (7627195140299511853) ----+-----+-----+-----+-----+-----+
| Member                | Host                | Role   | State   | TL | Receive LSN | Lag | Replay LSN | Lag |
+-----+-----+-----+-----+-----+-----+-----+-----+
| zgc2026n01.example.edu | 10.42.57.161        | Replica | streaming | 5  | 1/2A00DDA0   | 0   | 1/2A00DDA0   | 0   |
| zgc2026n02.example.edu | 10.42.57.162        | Leader | running  | 5  |              |     |             |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Beispiel Switchover bei 2 PostgreSQL Knoten

```
demo@zgc2026n01:~# sudo -u postgres patronictl -c /etc/patroni/config.yml list
+ Cluster: pg-ha-cluster (7627195140299511853) ----+-----+-----+-----+-----+-----+-----+-----+
| Member                | Host                | Role   | State   | TL | Receive LSN | Lag | Replay LSN | Lag |
+-----+-----+-----+-----+-----+-----+-----+-----+
| zgc2026n01.example.edu | 10.42.57.161 | Leader | running | 4 |              |    |              |    |
| zgc2026n02.example.edu | 10.42.57.162 | Replica | streaming | 4 | 1/293A7958 | 0 | 1/293A7958 | 0 |
| zgc2026n03.example.edu | 10.42.57.163 | Replica | streaming | 4 | 1/293A7958 | 0 | 1/293A7958 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Beispiel Status mit 3 PostgreSQL Knoten



...sind gut konfiguriert

Die deutsche Bahn 
zuverlässiger als:

Als das deutsche Mobilfunknetz 

Die Wettervorhersage 

Murphy: "Anything that can go wrong will go wrong."

Deshalb Monitoring nicht

vergessen!

ZABBIX

ZABBIX '26
CONFERENCE

GERMANY

 **RHEIN-MAIN
SOLUTIONS**

Fragen

Thomas Oftring
Rhein-Main Solutions GmbH



ZABBIX '26
CONFERENCE

GERMANY

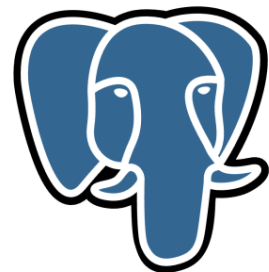


ZABBIX

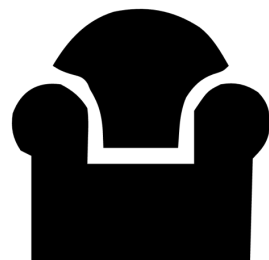
[Zabbix \(AGPLv3 License\)](#)



[patroni \(MIT License\)](#)



[PostgreSQL \(PostgreSQL License\)](#)



[pgBackRest \(MIT License\)](#)

ZABBIX '26

CONFERENCE

GERMANY

 **RHEIN-MAIN
SOLUTIONS**