

Pgpool-II徹底入門

～クラウド時代の Pgpool-II の活用および
新バージョン 4.2 のご紹介～



2020/11/13 JPUG Conference

SRA OSS, Inc. 日本支社

石井 達夫

彭博

本日のアジェンダ

- 第1部
 - Pgpool-IIとは
 - コンセプト
 - Pgpool-IIの機能
 - 盛り沢山の機能をわかりやすくご紹介
- 第2部
 - Pgpool-IIのユースケース
 - Aurora、Kubernetes での活用
 - Pgpool-II 4.2の新機能
- Q/A



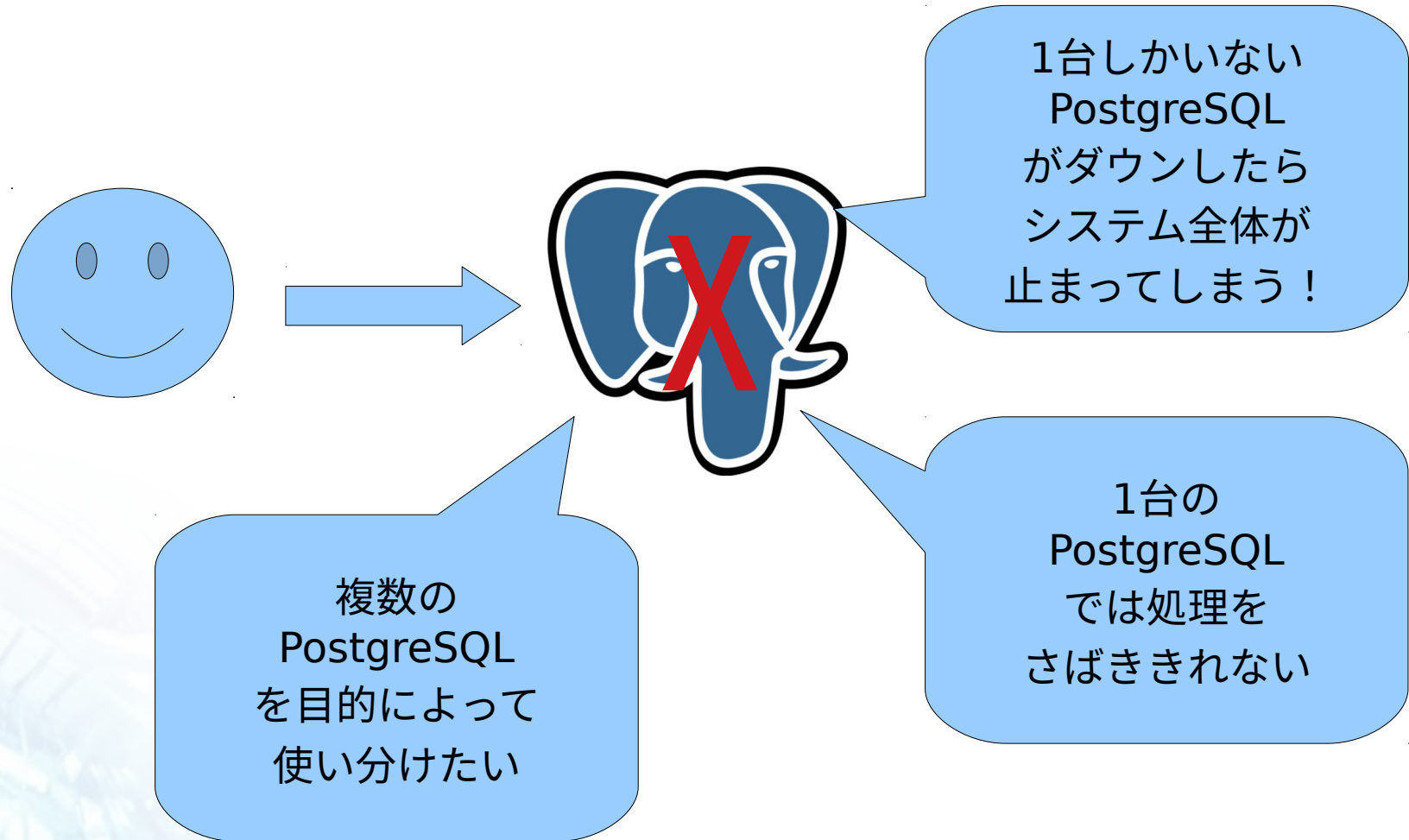
自己紹介

- 石井達夫
 - SRA OSS, Inc.日本支社で経営者兼OSS開発者をしています
 - Pgpool-IIの開発者、PostgreSQLコミッタ
 - 趣味は音楽鑑賞、SF、下手な料理



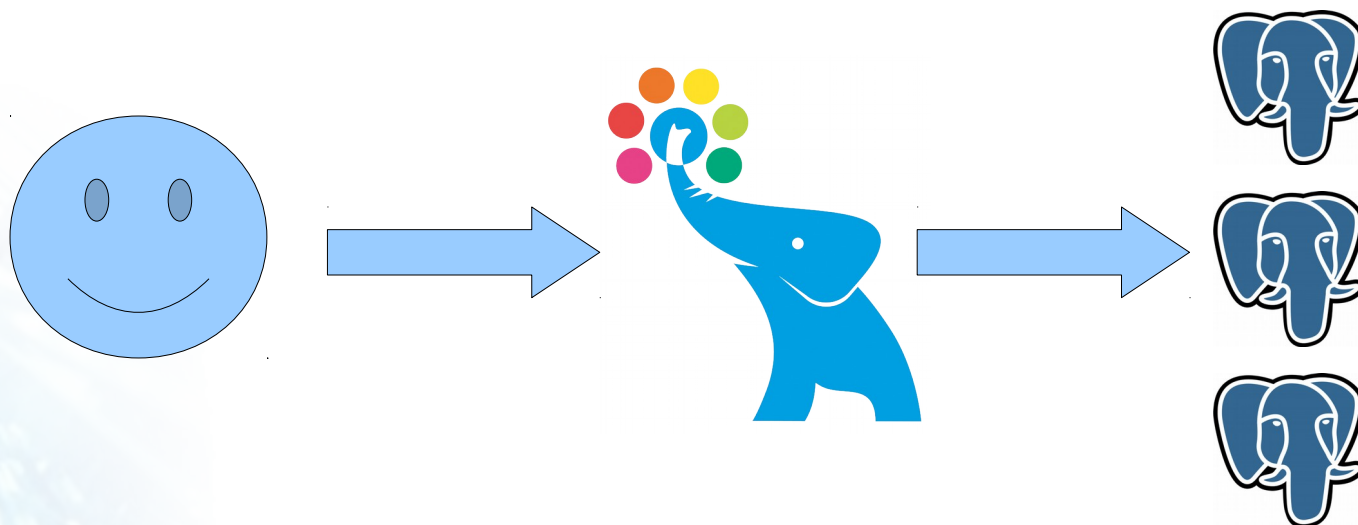
2018年PGConf.EU開催地ポルトガルにて

そもそもなぜクラスタは必要なの？

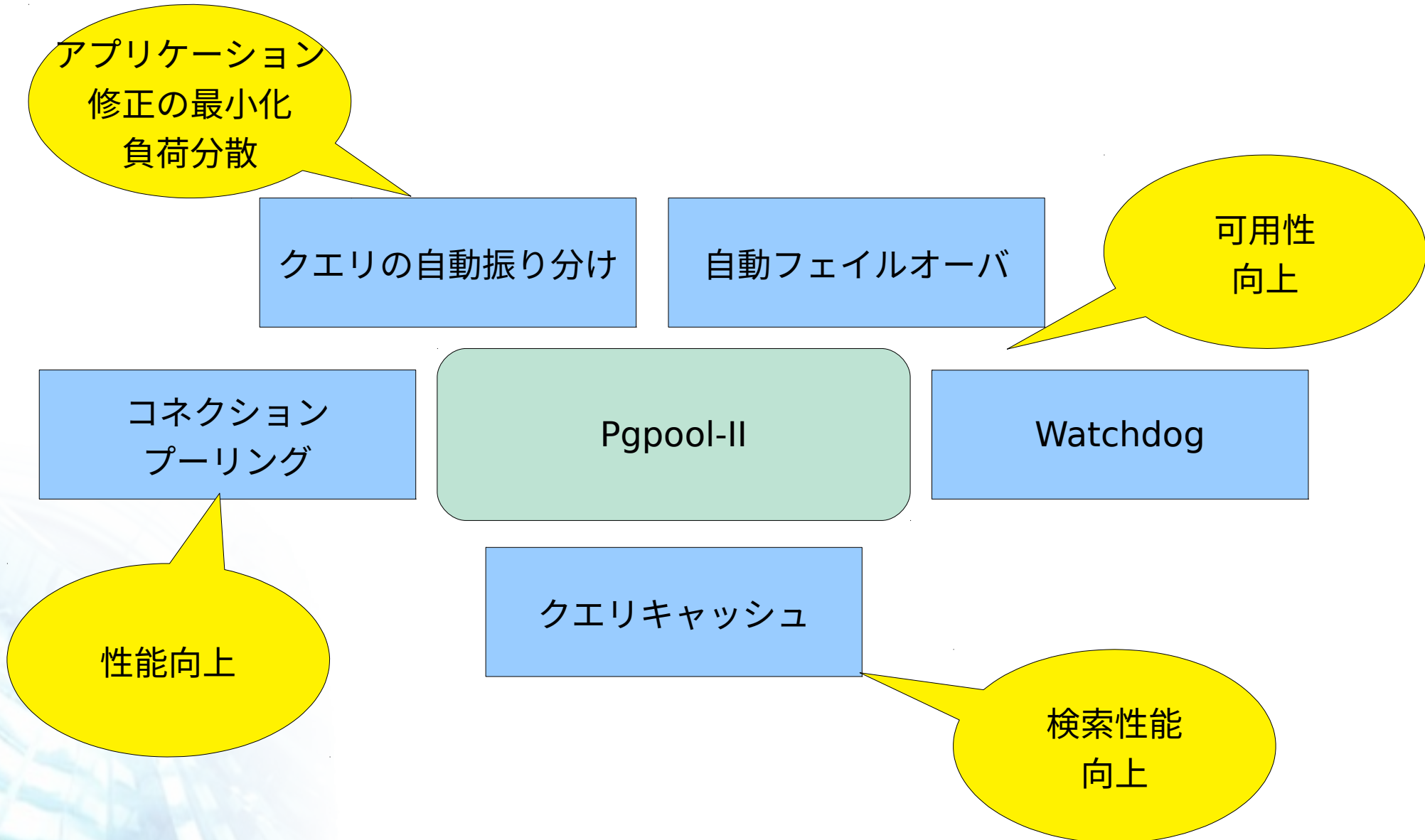


Pgpool-IIのコンセプト

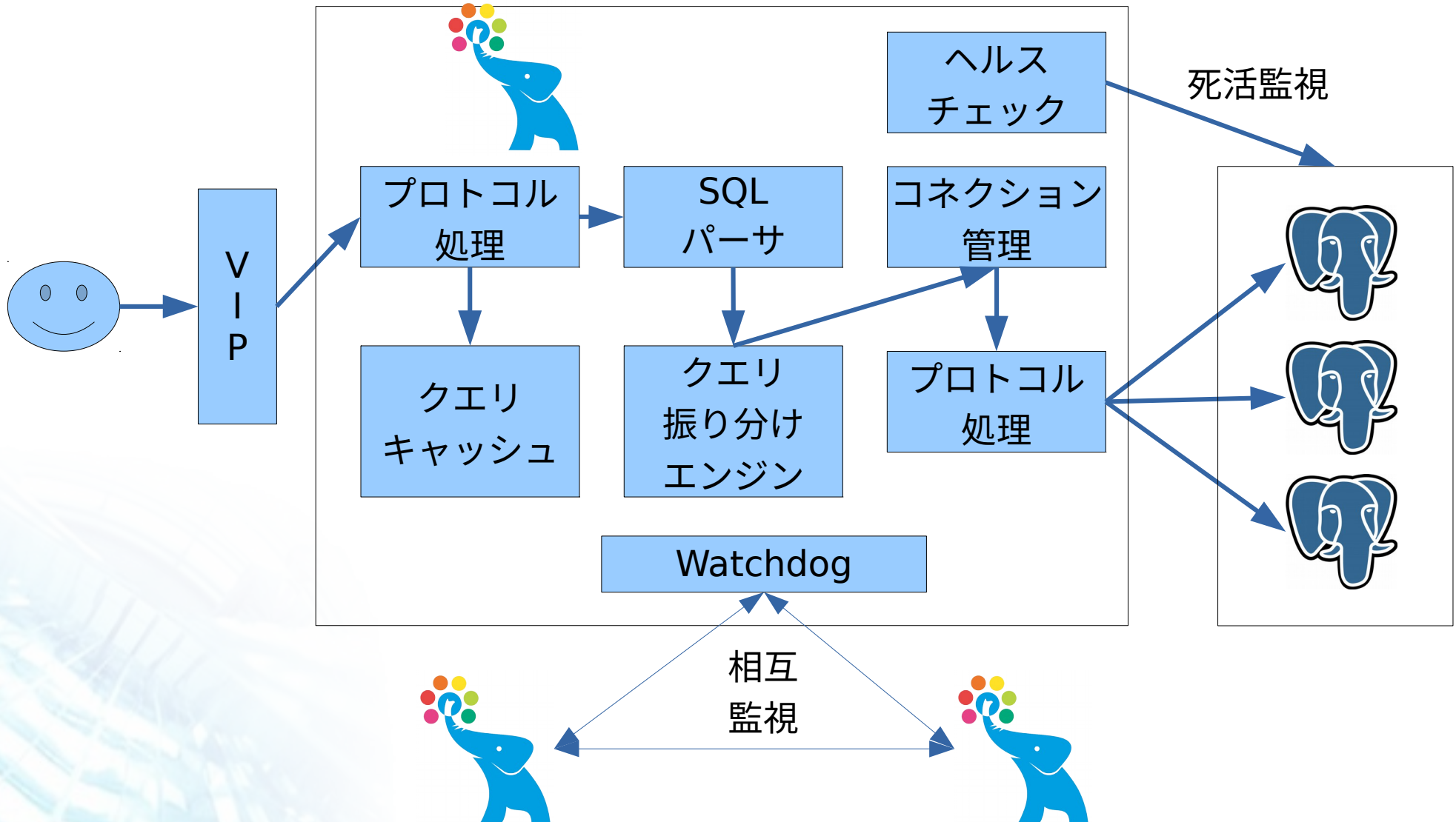
- ✓ ユーザから見て、なるべく複数PostgreSQLによるクラスタを使っていることを意識させたくない
- ✓ シングルPostgreSQL用のアプリケーションがなるべく変更なしに複数PostgreSQLによるクラスタを使う環境に移行できるようにしたい



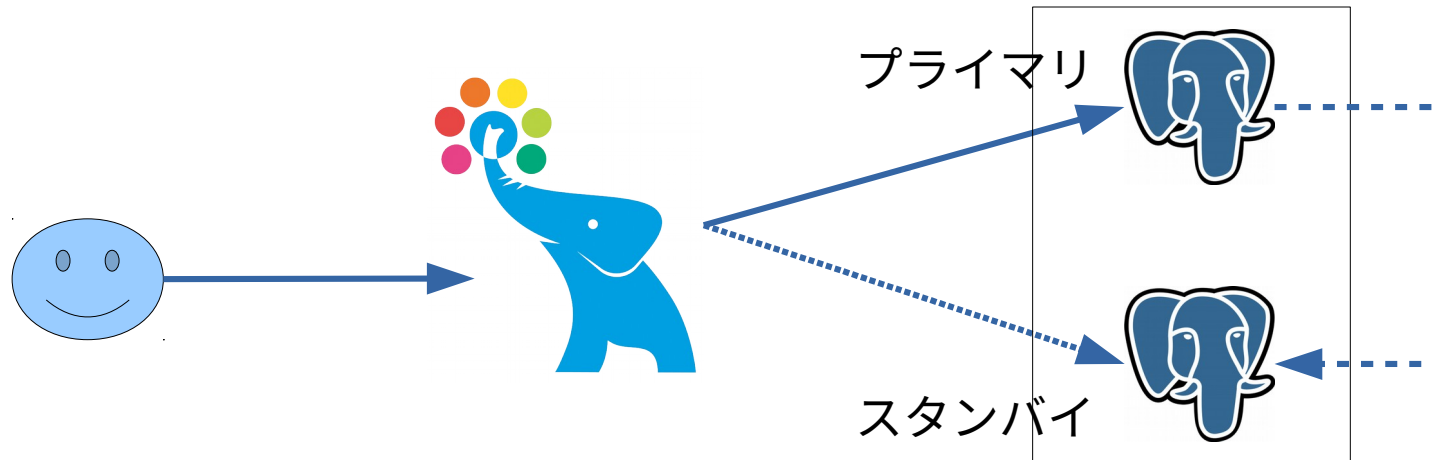
Pgpool-IIの機能



Pgpool-IIの構造

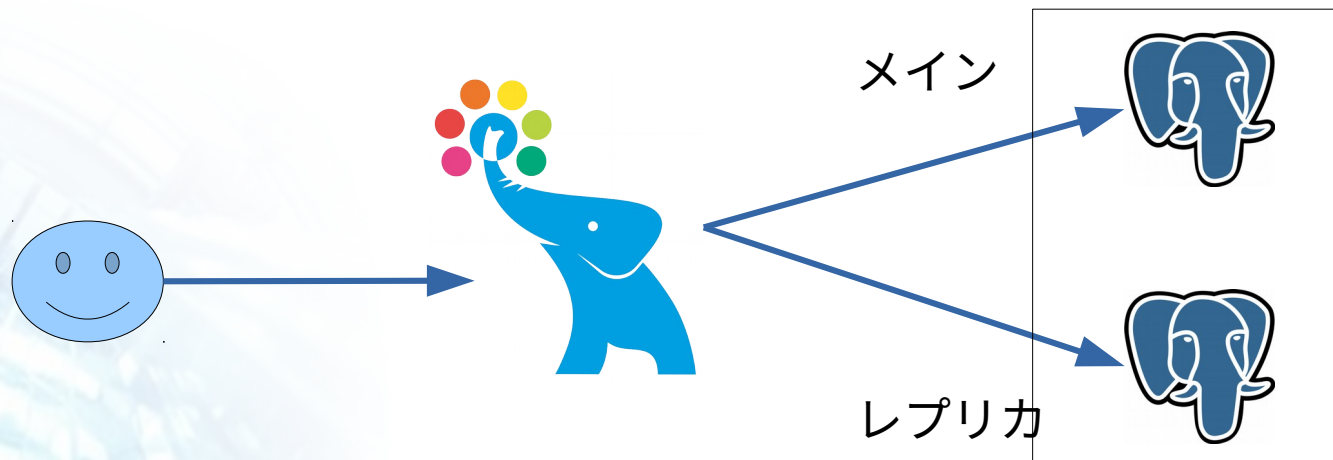


Pgpool-IIの機能: クエリの振り分け



本日主に
取り上げるのは
こちら

ストリーミング
レプリケーション
モード
更新クエリを
プライマリにのみ
投げる



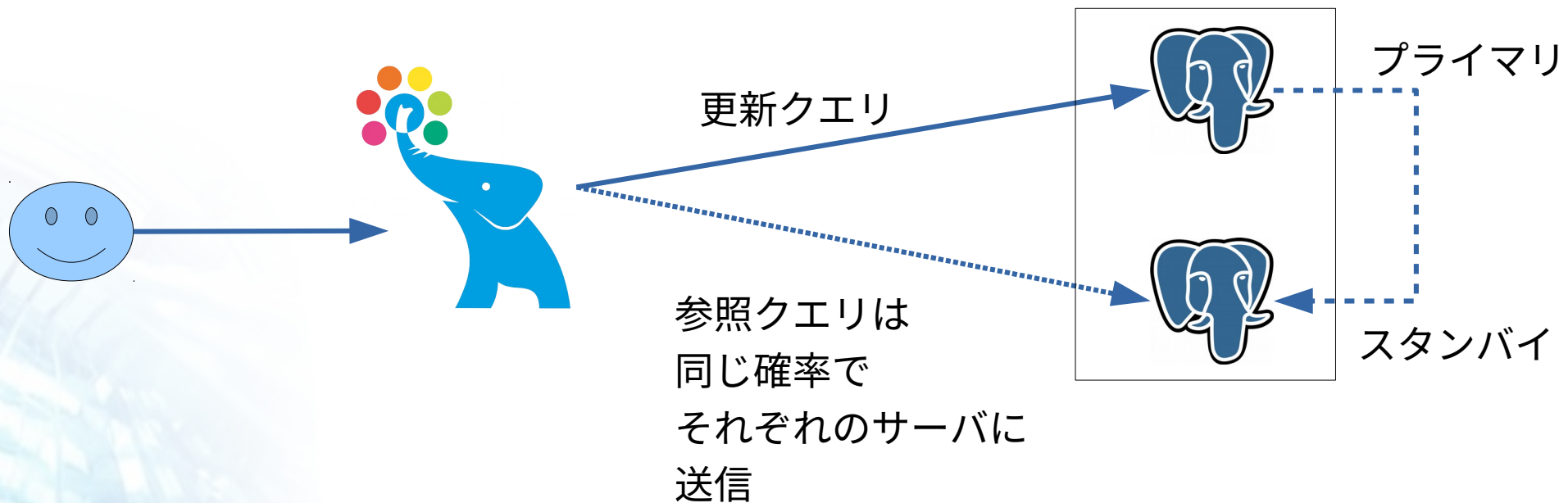
ネイティブ
レプリケーション
モード
更新クエリを
両方に投げる

Pgpool-IIの機能:クエリの振り分け

- たぶんPgpool-IIを使いたい最大の理由がクエリの振り分け
 - スタンバイに投げてはいけないクエリをプライマリにのみ投げる
 - 更新クエリ
 - 強いロック
 - シリアライザブルトランザクション
 - その他
 - それ以外のクエリはスタンバイに投げる
 - クエリ振り分けのメリット
 - シングルPostgreSQL用のアプリケーションをストリーミングレプリケーション環境にそのまま移行できる
 - スタンバイのリソースを有効活用できる
 - プライマリサーバの負担を軽減する

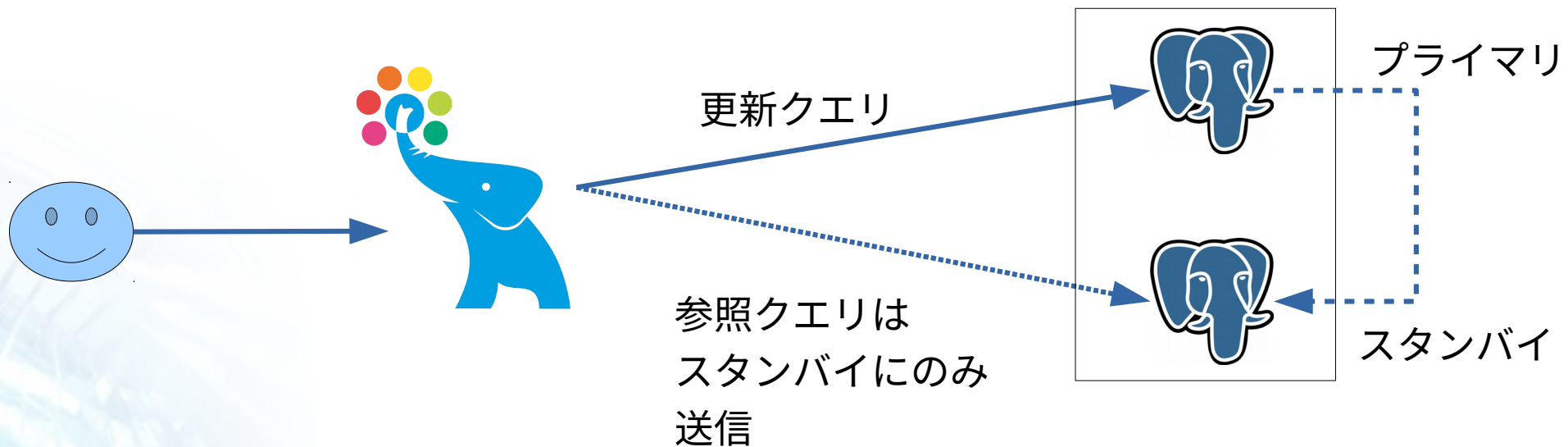
クエリの振り分けの設定

- 負荷分散の重み付けが設定可能
 - `backend_weight0 = 0.5`
 - `backend_weight1 = 0.5`



クエリの振り分けの設定

- プライマリサーバを更新処理に専念させ、負荷を下げる設定の例
 - `backend_weight0 = 0.0`
 - `backend_weight1 = 1.0`

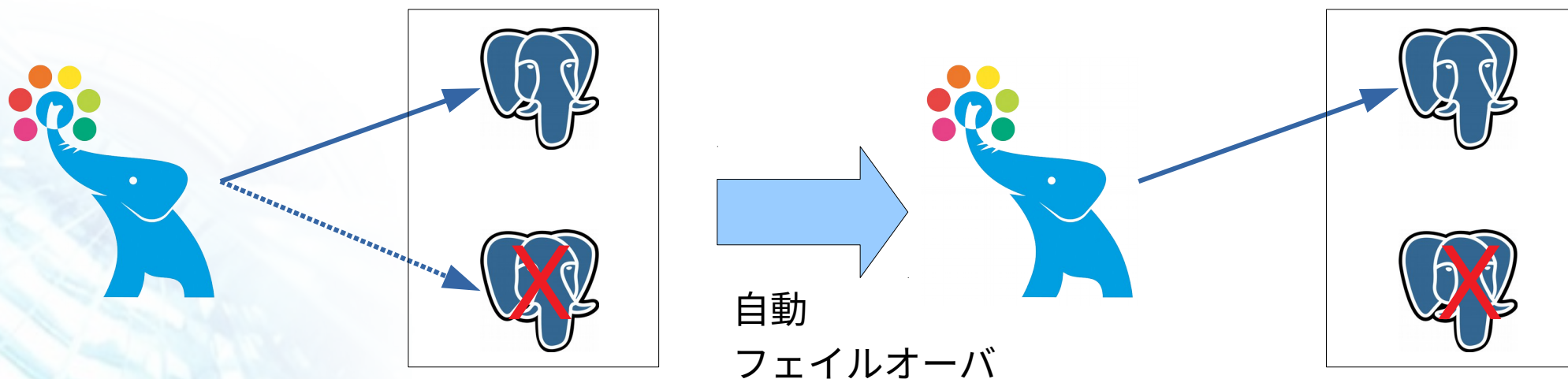


その他のクエリ振り分けの設定

- `statement_level_load_balance`
 - クエリの振り分けをセッション単位ではなく、SQL文単位で決定
- `database_redirect_preference_list`
 - データベース名によってクエリの振り分け先を決定
- `application_redirect_preference_list`
 - アプリケーション名によってクエリの振り分け先を決定
- `disable_load_balance_on_write`
 - 明示的なトランザクション内のクエリの振り分けの挙動を制御

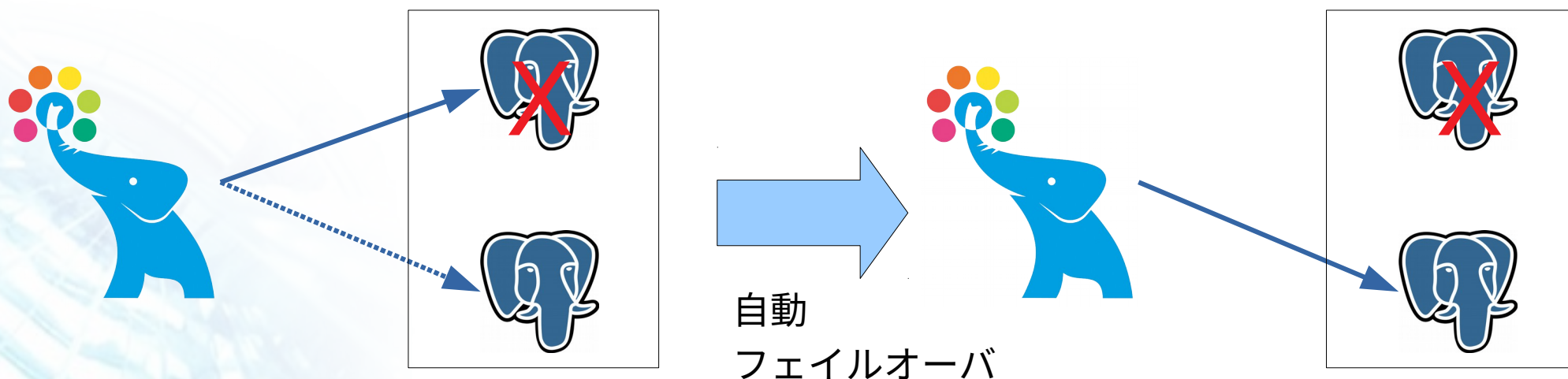
Pgpool-IIの機能：自動フェイルオーバー

- PostgreSQLサーバがダウンしたらそのサーバを切り離し、残りのサーバで処理を継続
- 定期的にヘルスチェックプロセスがPostgreSQLサーバと通信ができるかどうかを確認
- 通信ができなければ自動的にそのサーバを切り離す



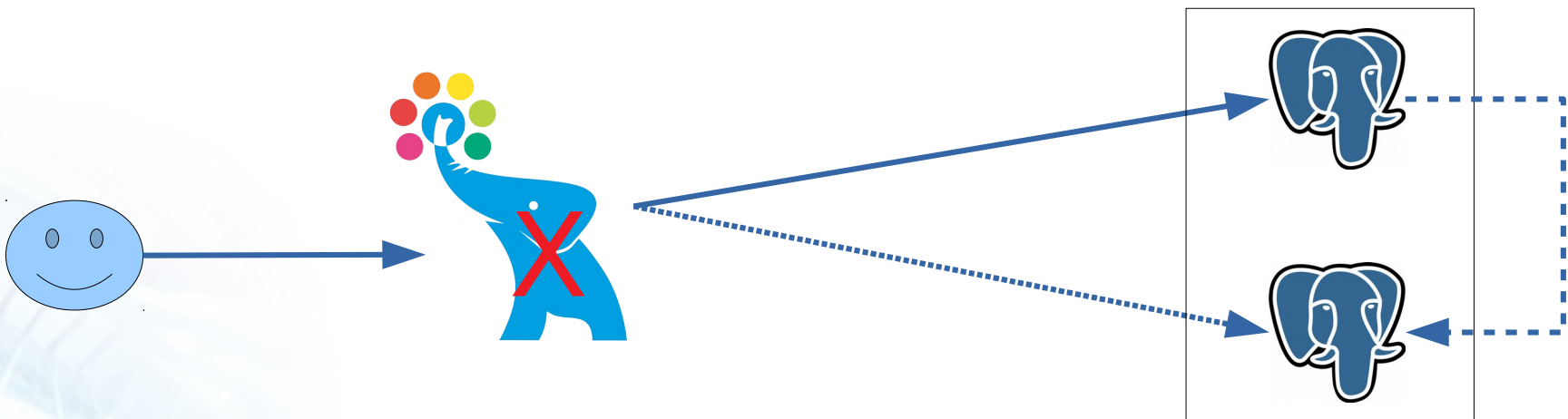
Pgpool-IIの機能：自動フェイルオーバー

- プライマリサーバがダウンした時は、スタンバイサーバを自動昇格して新しいプライマリサーバで処理を続行する
 - 昇格処理を行うスクリプトを指定できる



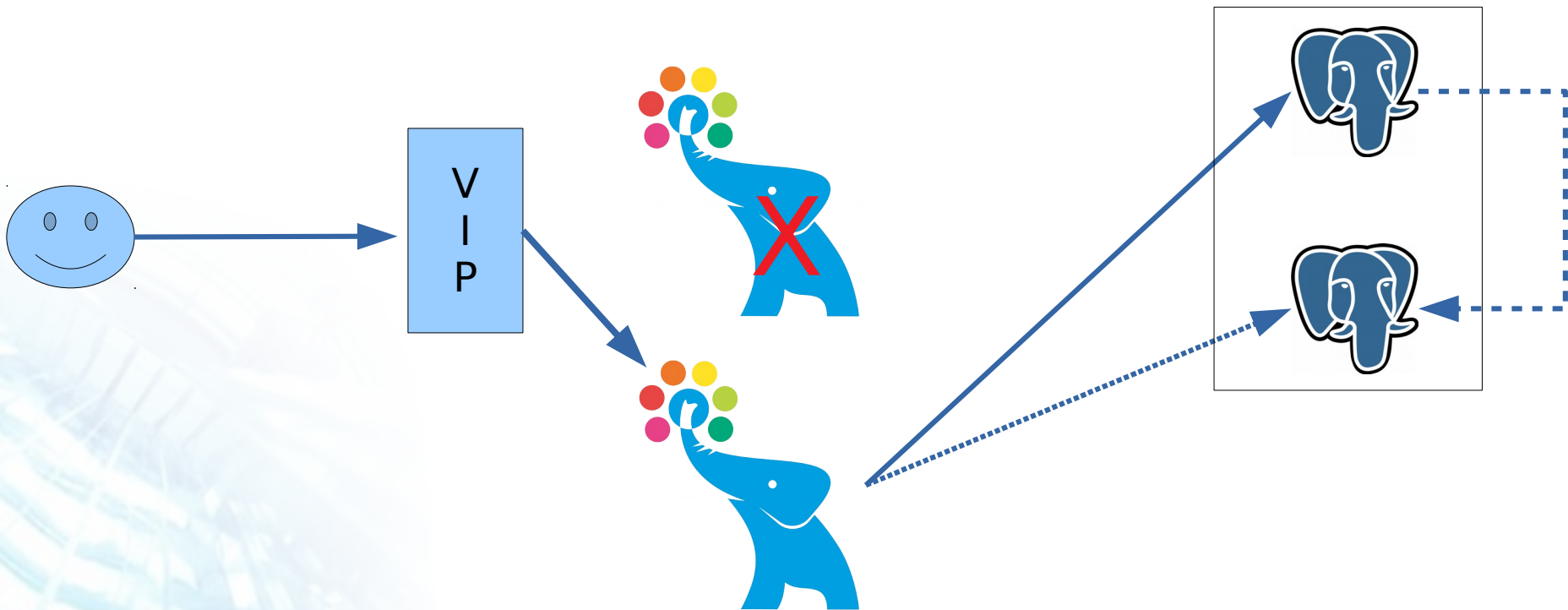
Pgpool-IIの機能: watchdog

- Pgpool-IIがダウンすると、PostgreSQLにアクセスできなくなってしまう(単一障害点、SPOF: Single point of failure)



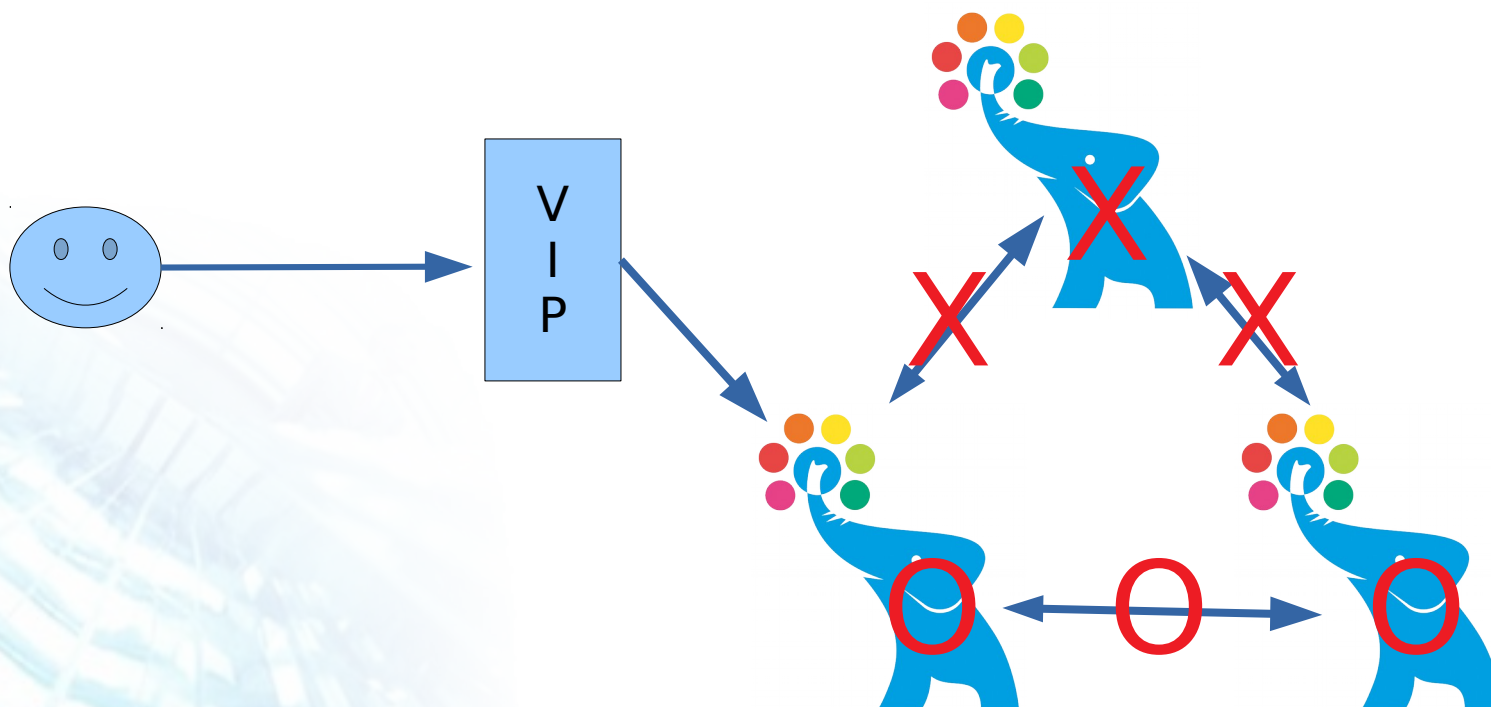
Pgpool-IIの機能: watchdog

- 複数のPgpool-IIを用意してお互いに監視し、相手がダウンしたときには待機系のPgpool-IIを昇格して運用を継続する
- Pgpool-IIへのアクセスはVIP経由にして、クライアント側では同一IPでアクセスし続けられるようにする



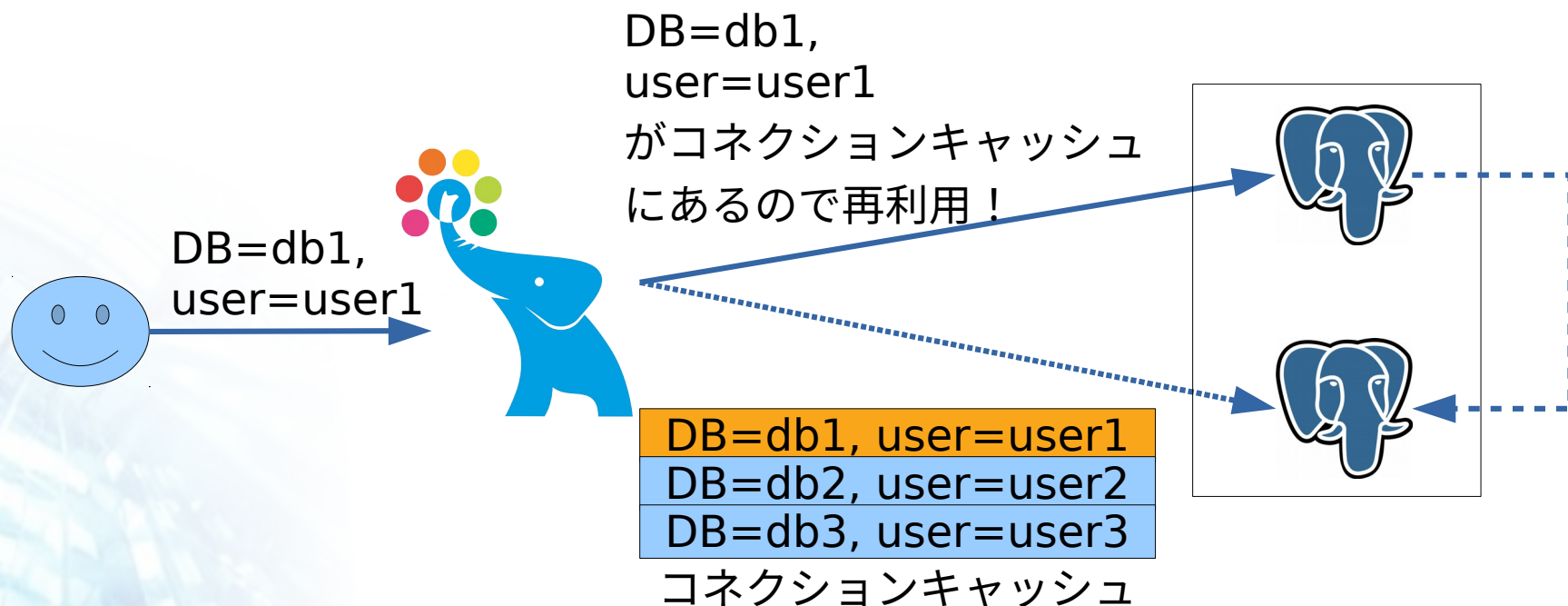
Pgpool-IIの機能: watchdog

- Pgpool-IIがお互いに相手がダウンしていると思ってしまう問題 (Split brain problem)の解決
- Pgpool-IIを奇数台用意することによって、多数決でどれが本当にダウンしているのかを正しく判断できる



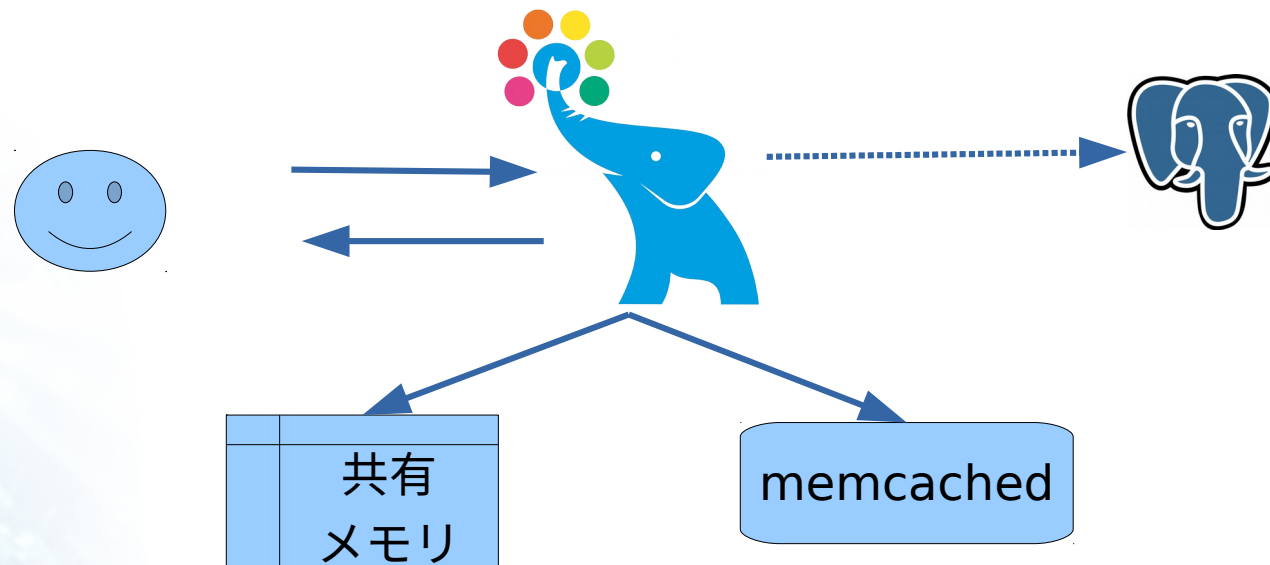
Pgpool-IIの機能： コネクションプーリング

- PostgreSQLに限らず、DBへの接続が時間がかかる
- そこで一度接続したDBへのコネクションを取っておき、条件が合えば再利用する
 - DB名、ユーザ名が同じであること



Pgpool-IIの機能:クエリキャッシュ

- 検索結果を共有メモリ(またはmemcached)に保存し、同じクエリ文字列が送られてきたら2回目以降はキャッシュから結果を返す
 - SQLパーサを経由せず、DBアクセスも発生しないので超高速
- キャッシュはテーブルに変更があったらクリアされる
- 更新の多いシステムには向かない(キャッシュヒット率70%以上での利用を推奨)



クエリキャッシュヒット率の確認

```
show pool_cache;
-[ RECORD 1 ]-----+-----
num_cache_hits      | 891703
num_selects         | 99995
cache_hit_ratio     | 0.90
num_hash_entries    | 131072
used_hash_entries   | 99992
num_cache_entries   | 99992
used_cache_enrties_size | 12482600
free_cache_entries_size | 54626264
fragment_cache_entries_size | 0
```

ここまでのまとめ



- Pgpool-IIのコンセプト、構造、基本機能をご紹介しました
- 他にも豊富な機能が盛り沢山なので、是非マニュアルやブログ記事をご覧ください
- では第2部にバトンタッチします。彭さんよろしくお願ひします