

L3冗長のはなし

株式会社 Jストリーム

ネットワークインフラ部 ネットワークインフラ課

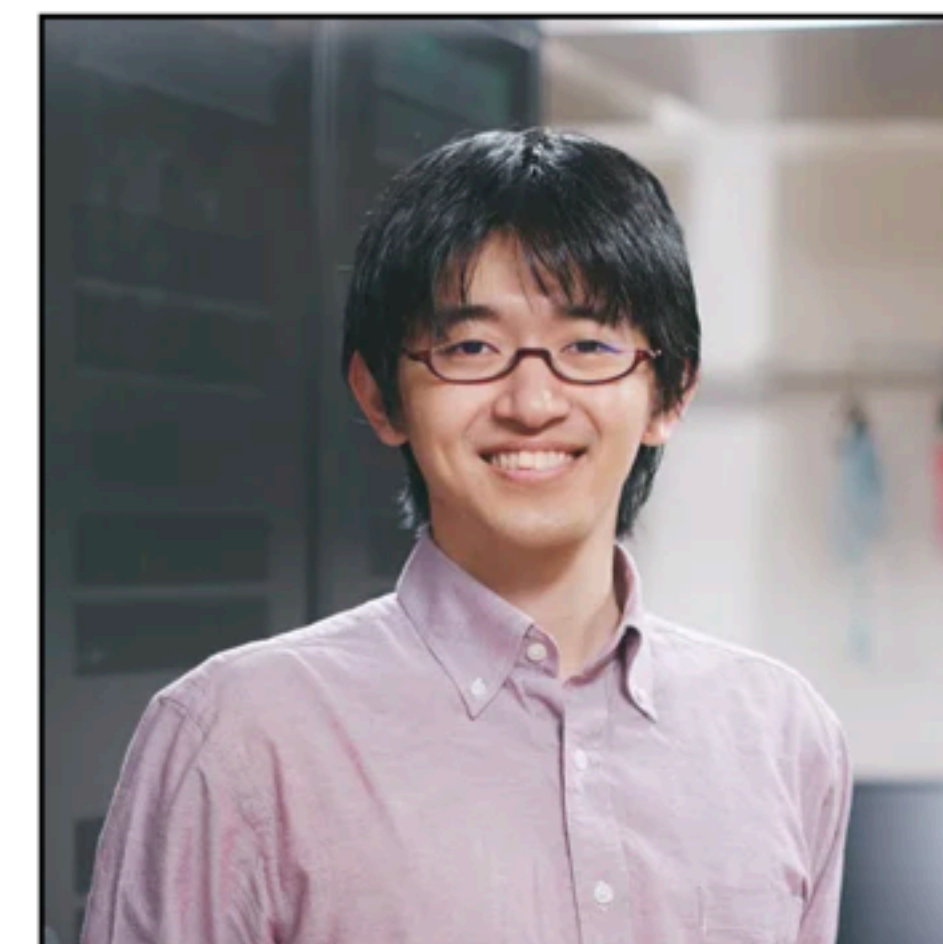
今井宏謙

2024-05-11

株式会社Jストリーム ネットワークインフラ部 ネットワークインフラ課

今井 宏謙

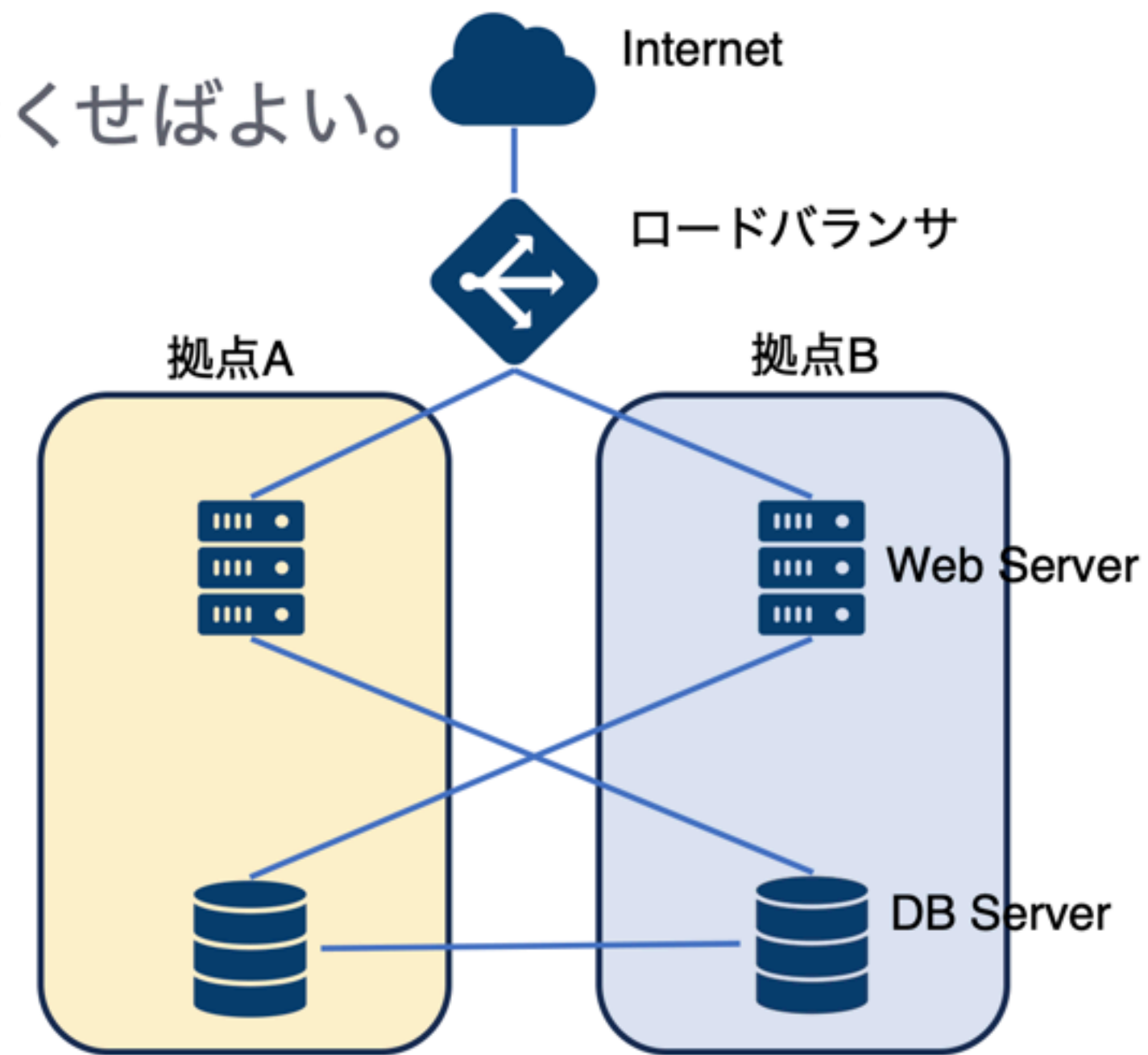
- 2020年度入社、新卒5年目
- CDNの基盤ネットワーク・サーバインフラの構築・運用
- JANOG参加は10回目ぐらい
 - 初参加はJANOG39@金沢に若者支援で参加しました
- JANOG54 BAKUCHIKU参加してます
 - 会場Wi-Fi、問題なく提供出来てるといいな



- この構成について説明します

パターン5 Active-Standby (Zabbix-HA)

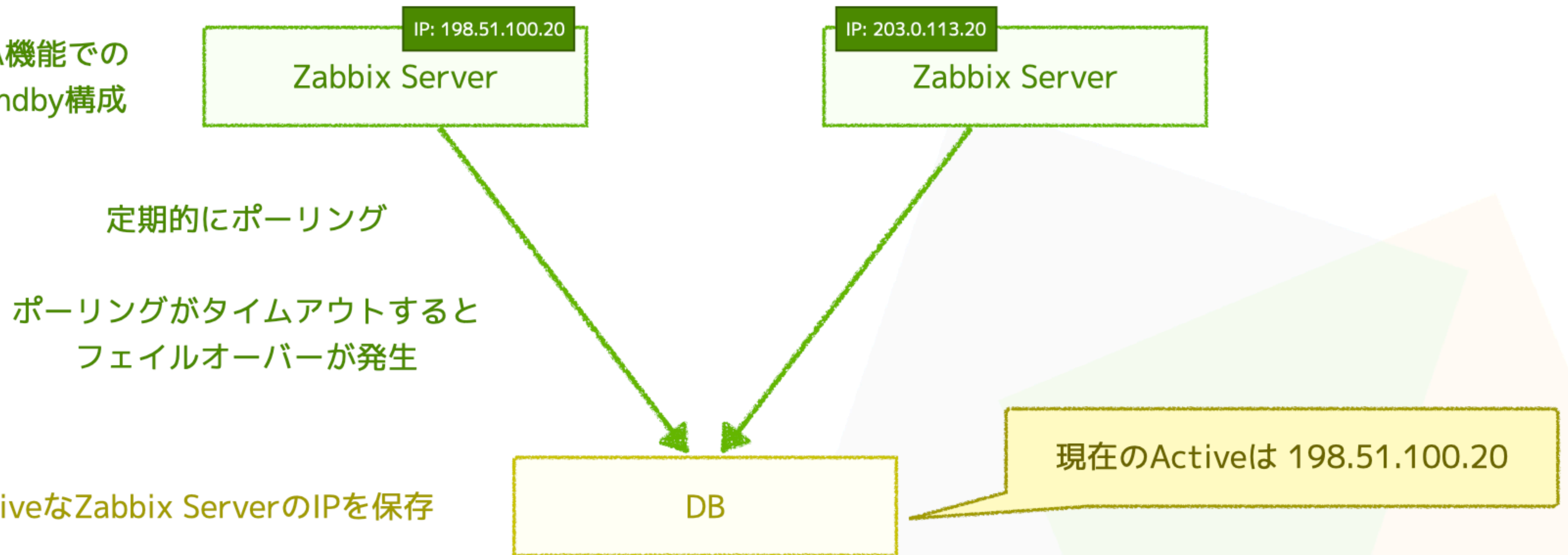
- パターン3を拡張
- VIPの問題があるのであれば、VIPをなくせばよい。
- 実現方法を考えてみた。



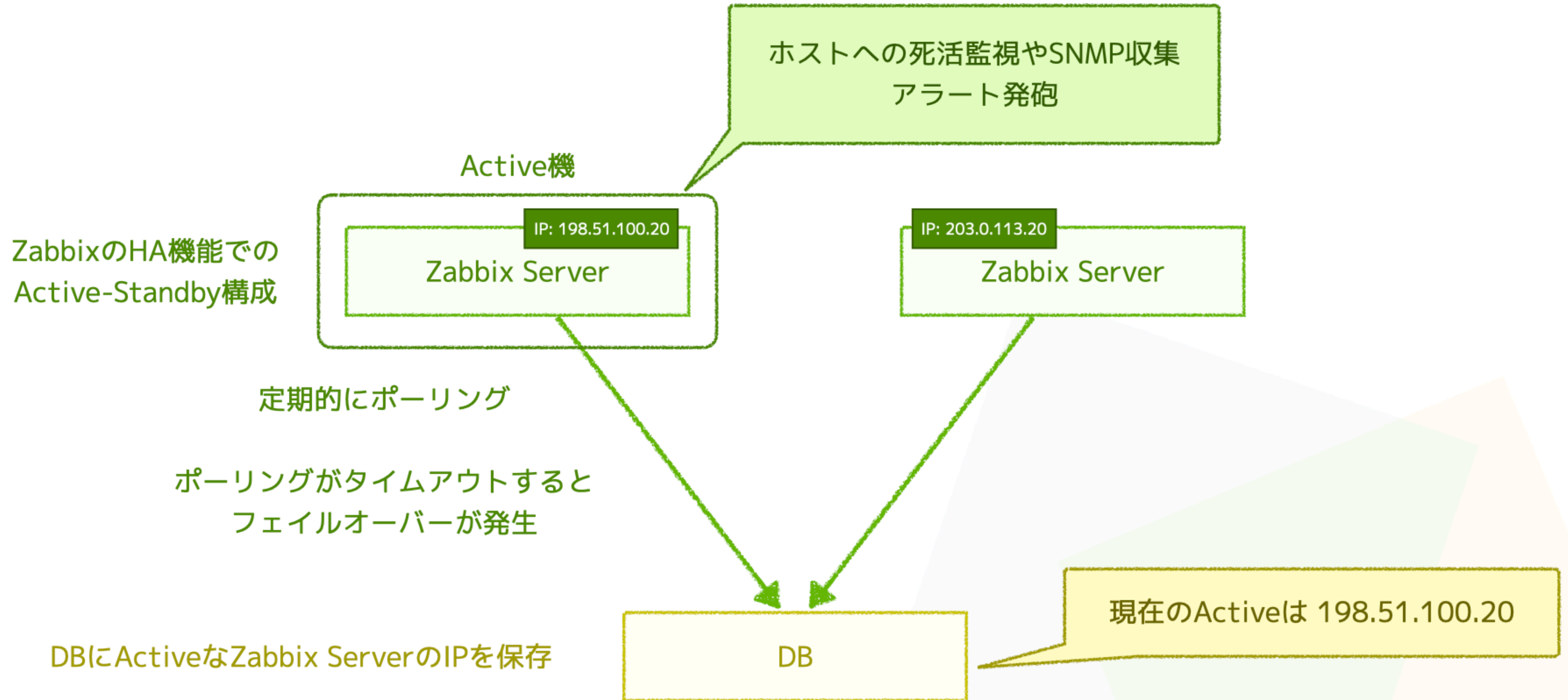
ZabbixのHA機能での
Active-Standby構成

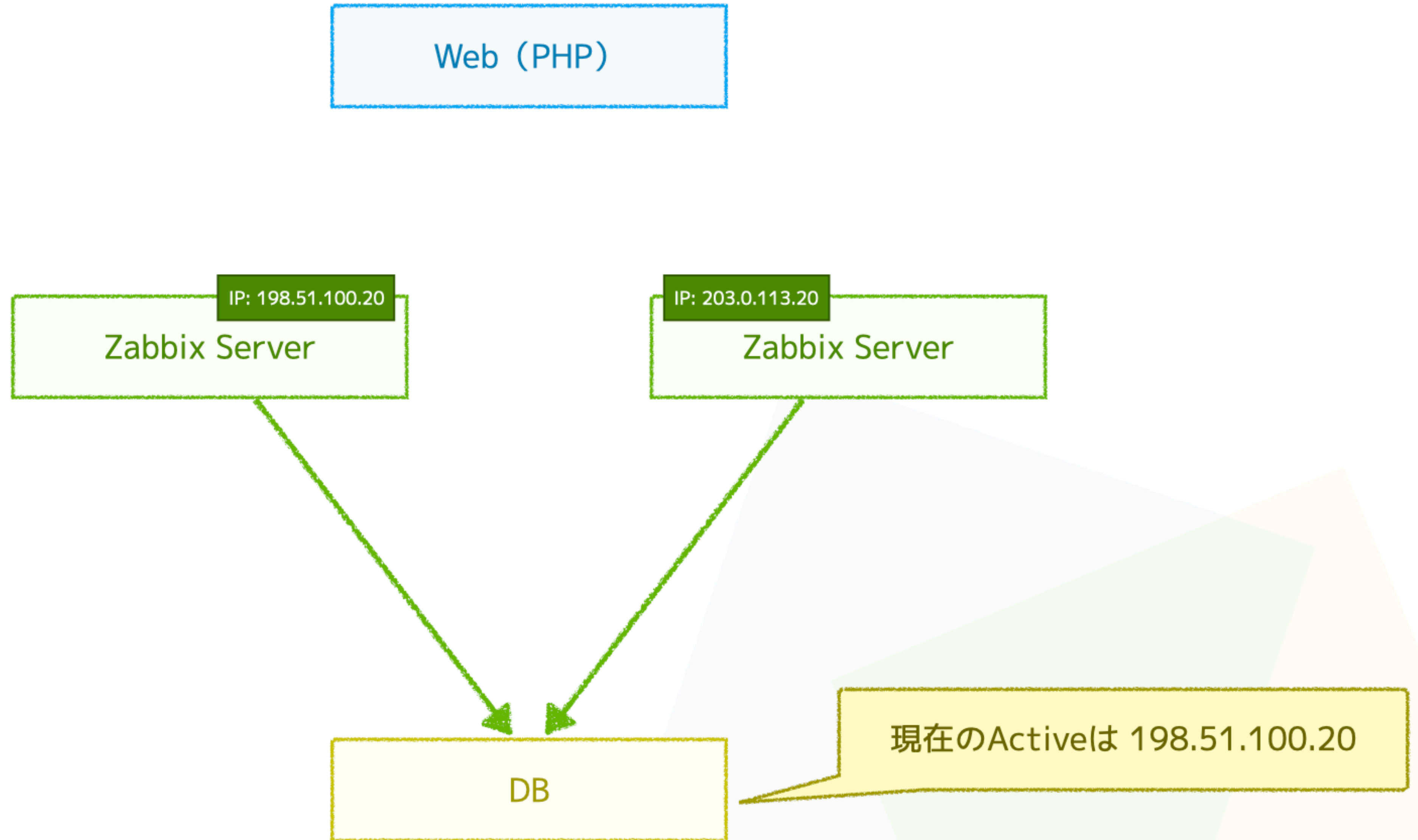


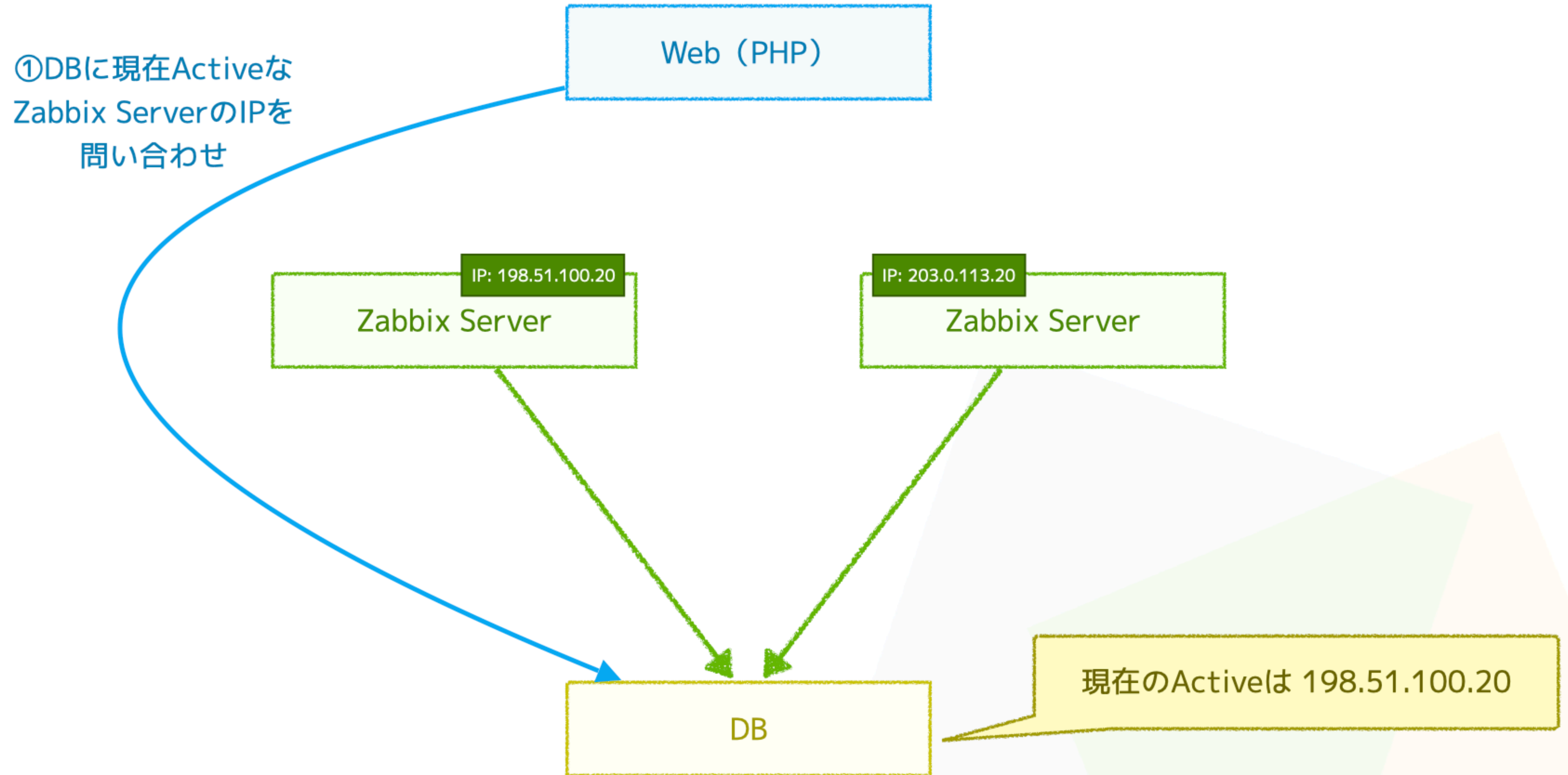
ZabbixのHA機能での
Active-Standby構成

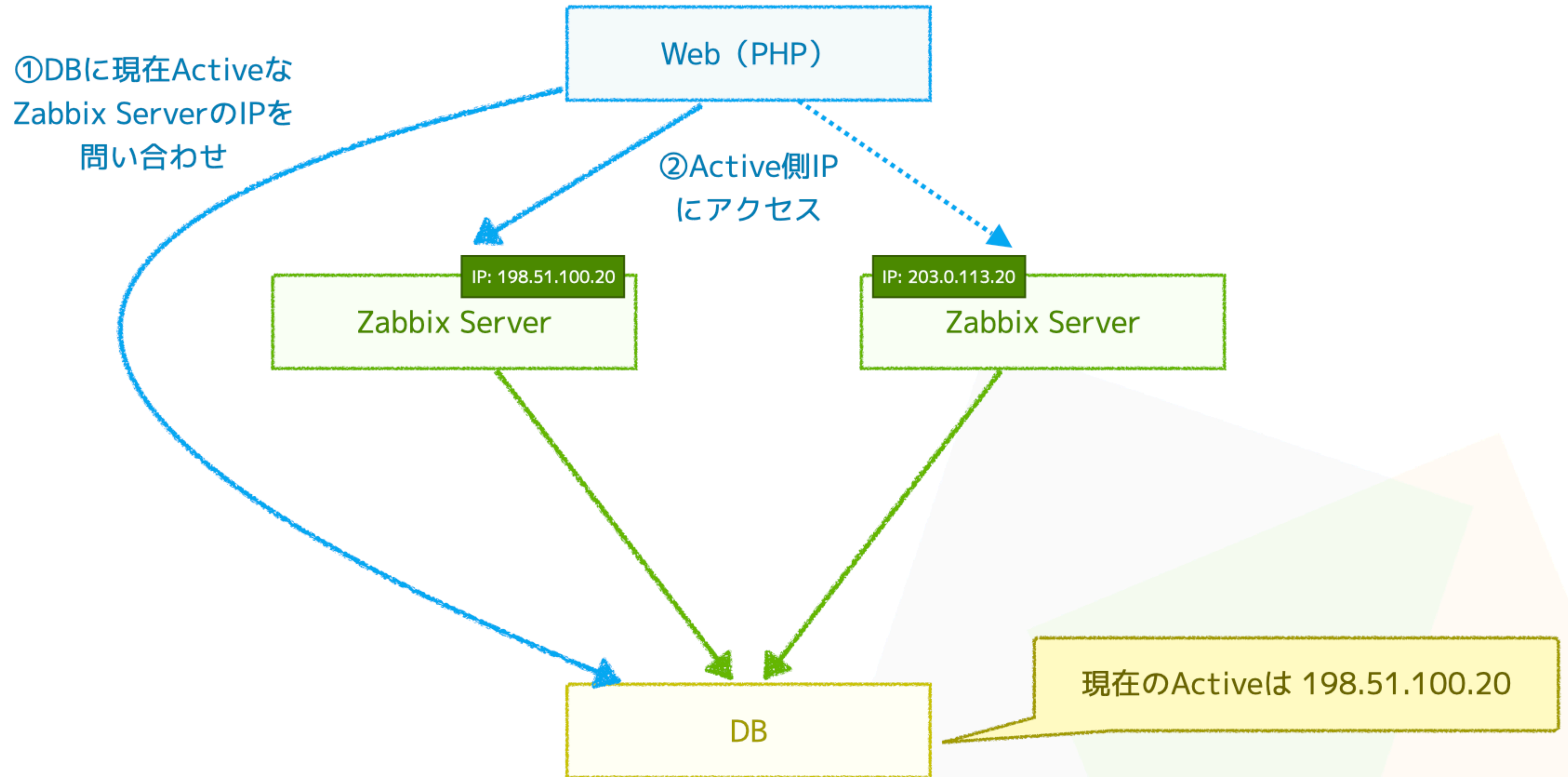


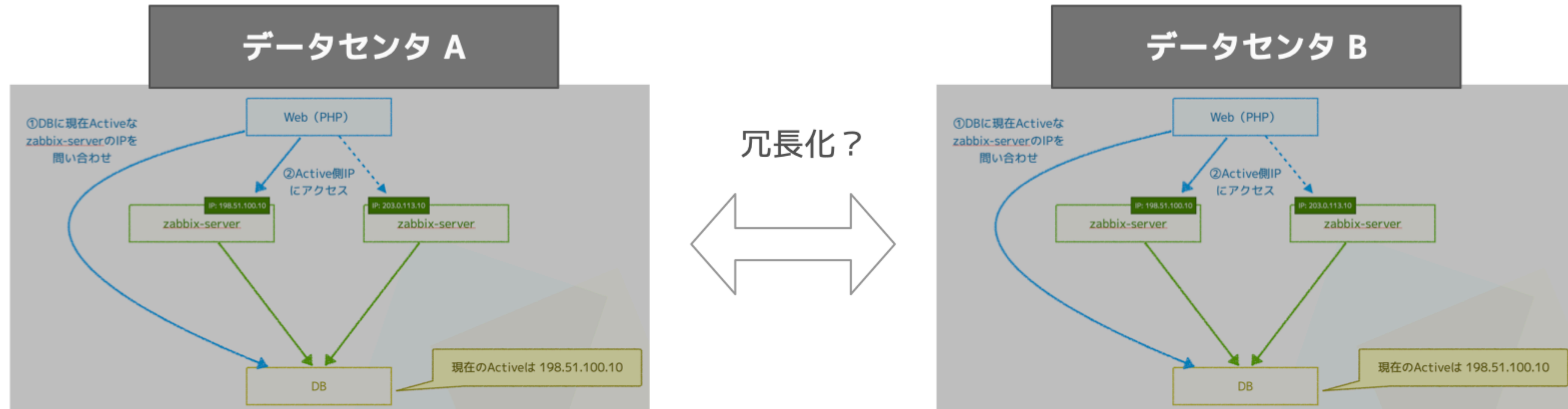
DBにActiveなZabbix ServerのIPを保存



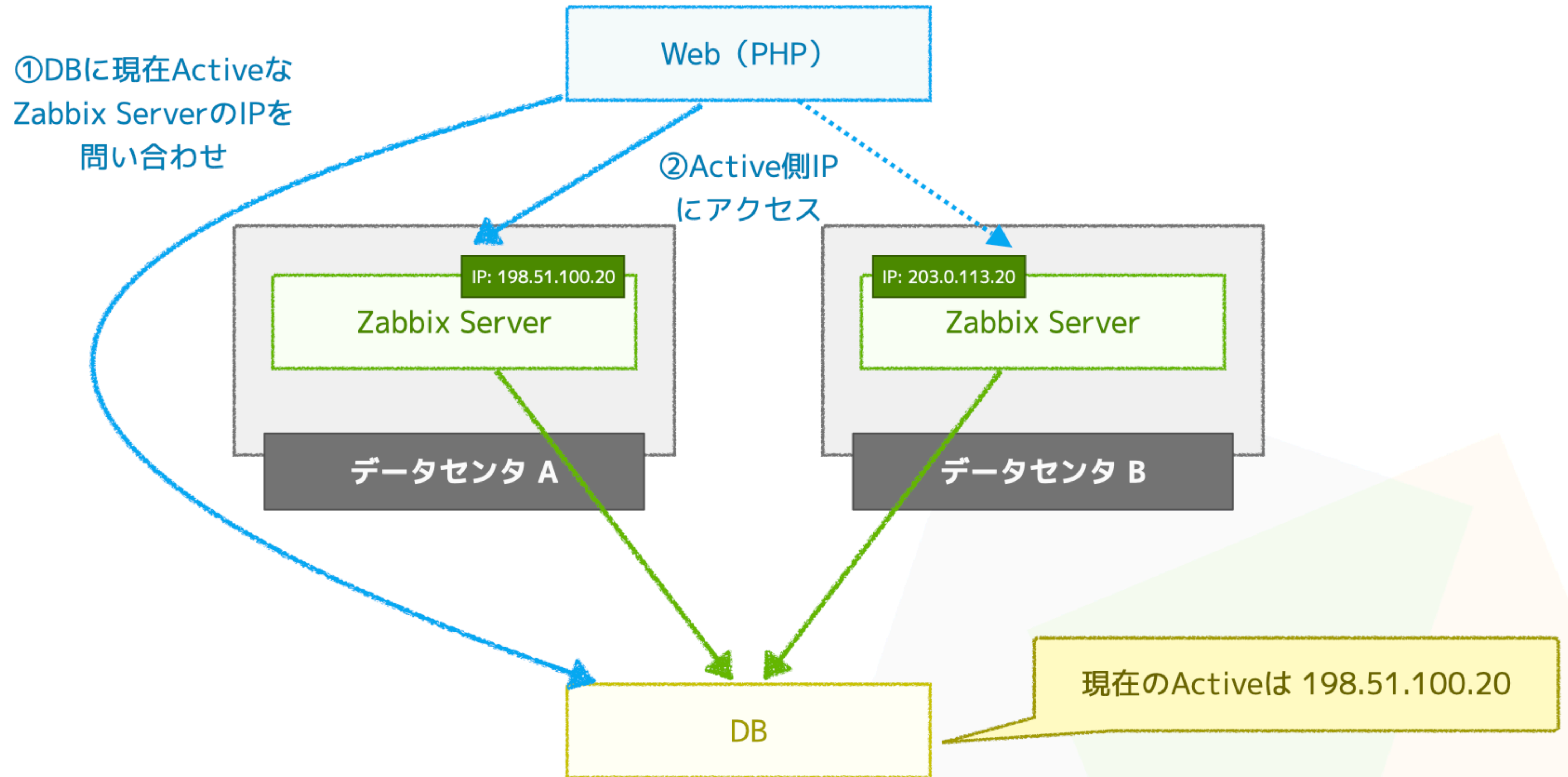


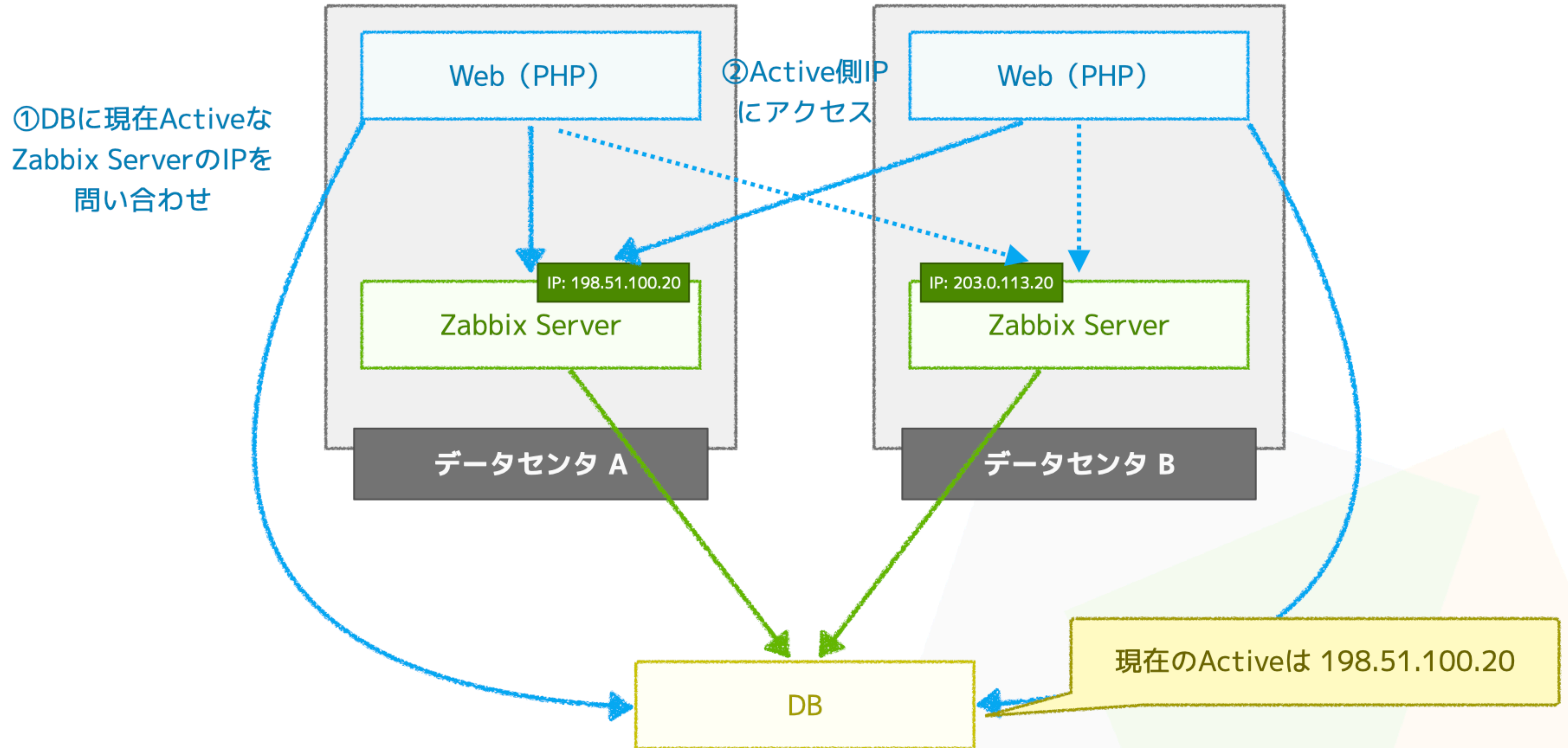


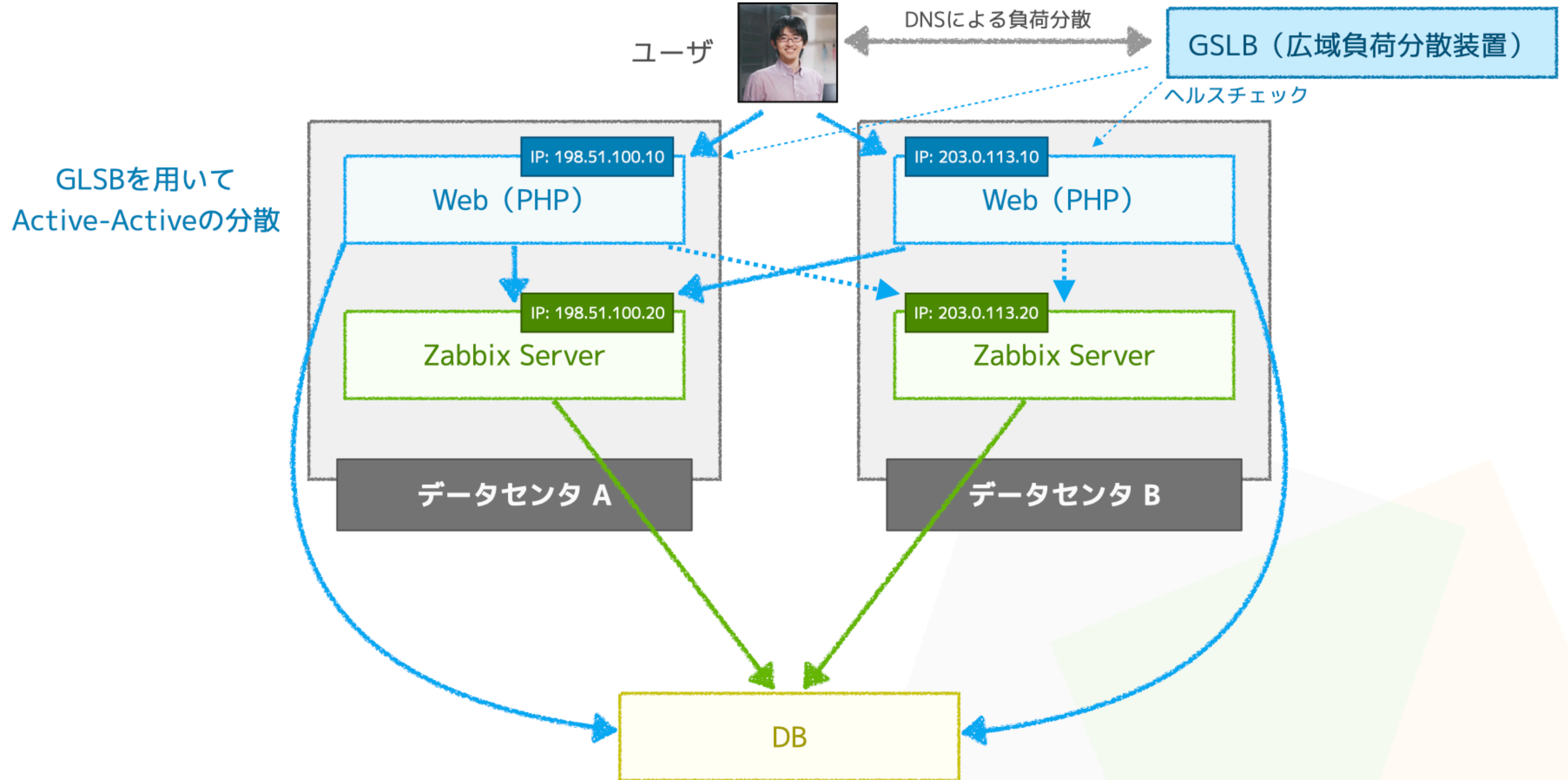


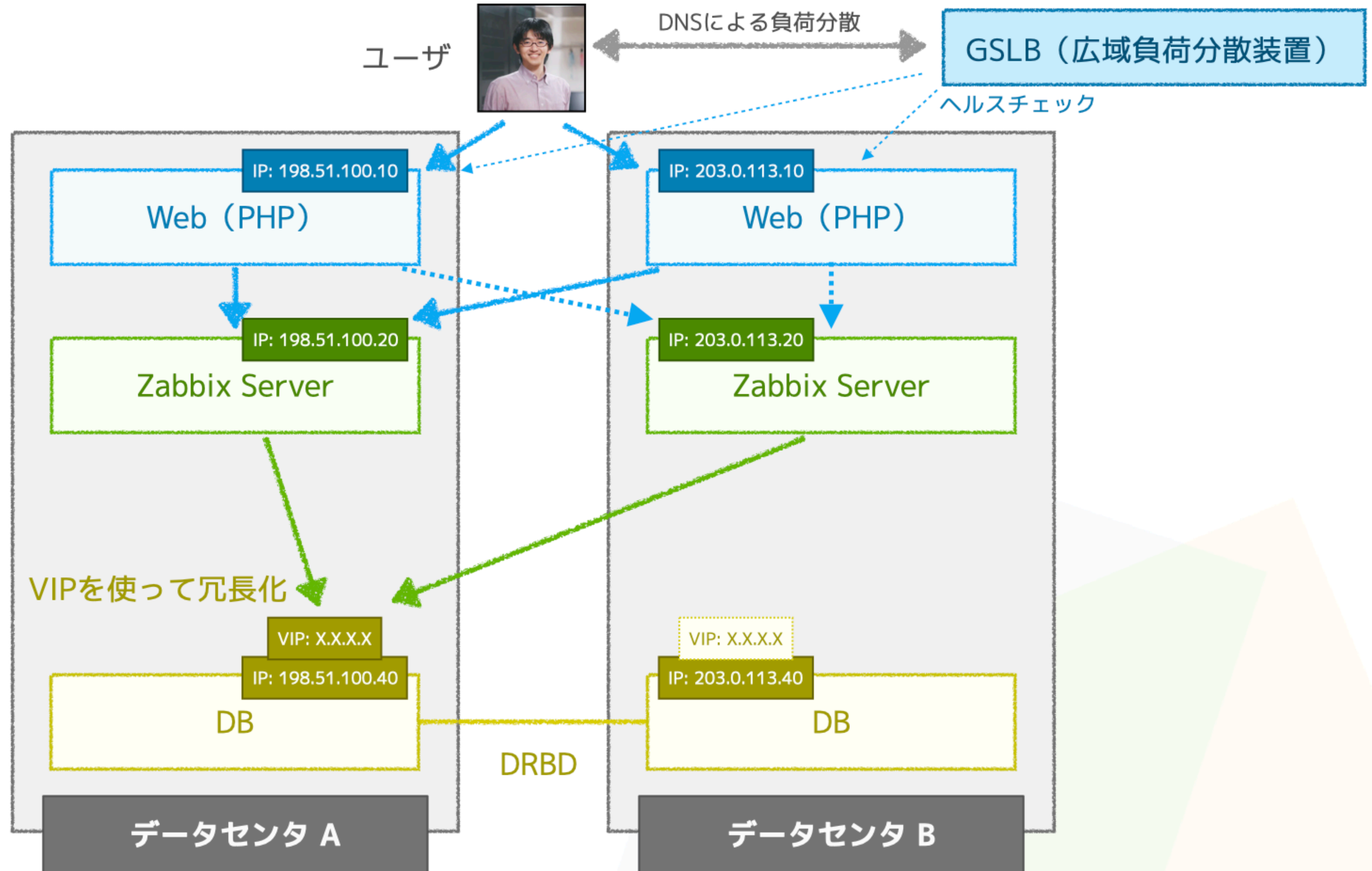


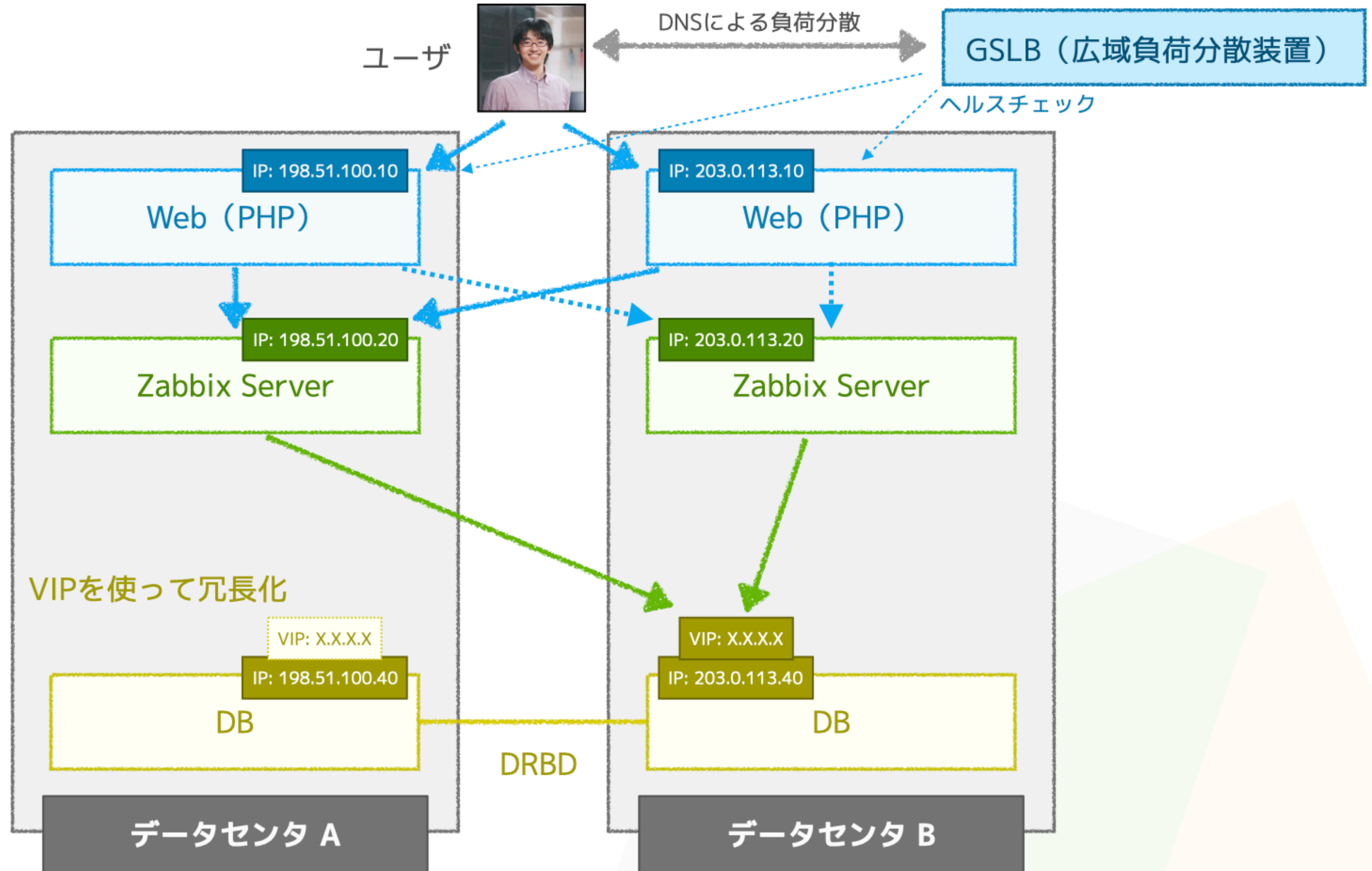
- 拠点を跨いだ冗長を行うには、一工夫必要そう
- Zabbixのパートナー企業様の資料によるとHAの方法はあるが、L2ネットワーク内で考えられているものが多く、拠点をまたぐのは難しそう
 - DC冗長を構築するドキュメントは乏しい、、、
 - DB周りはZabbix公式でドキュメントが見当たらなかった

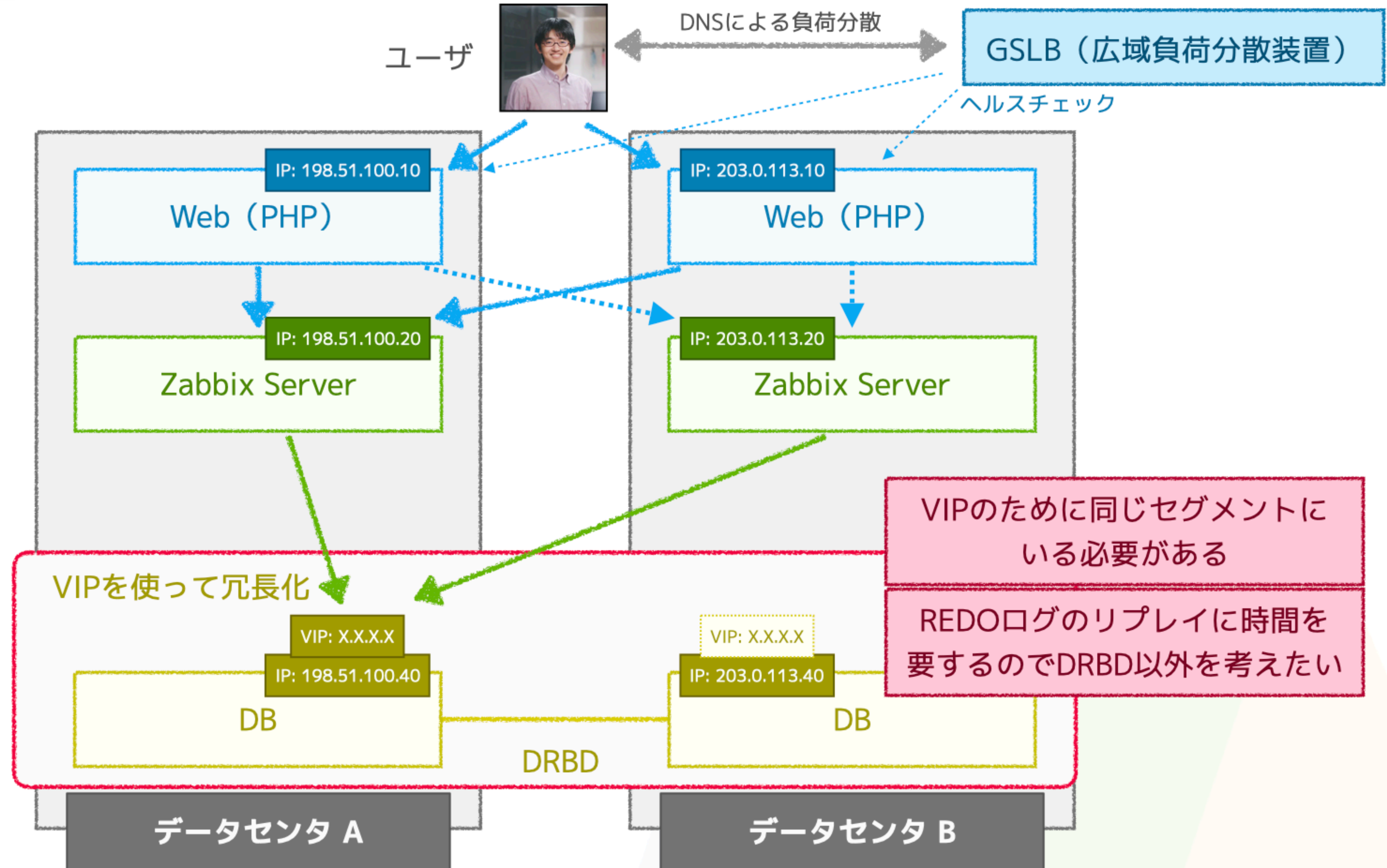


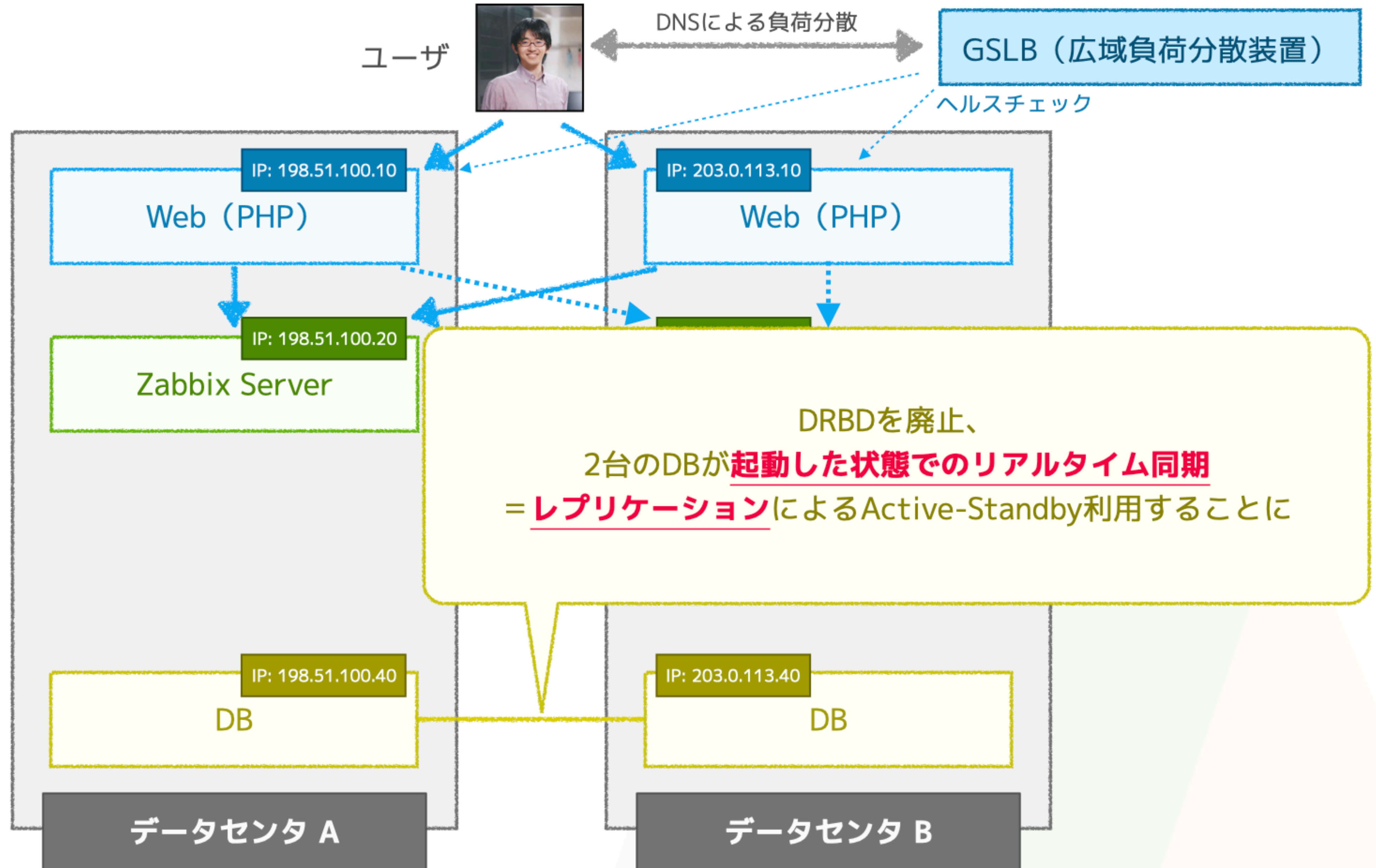


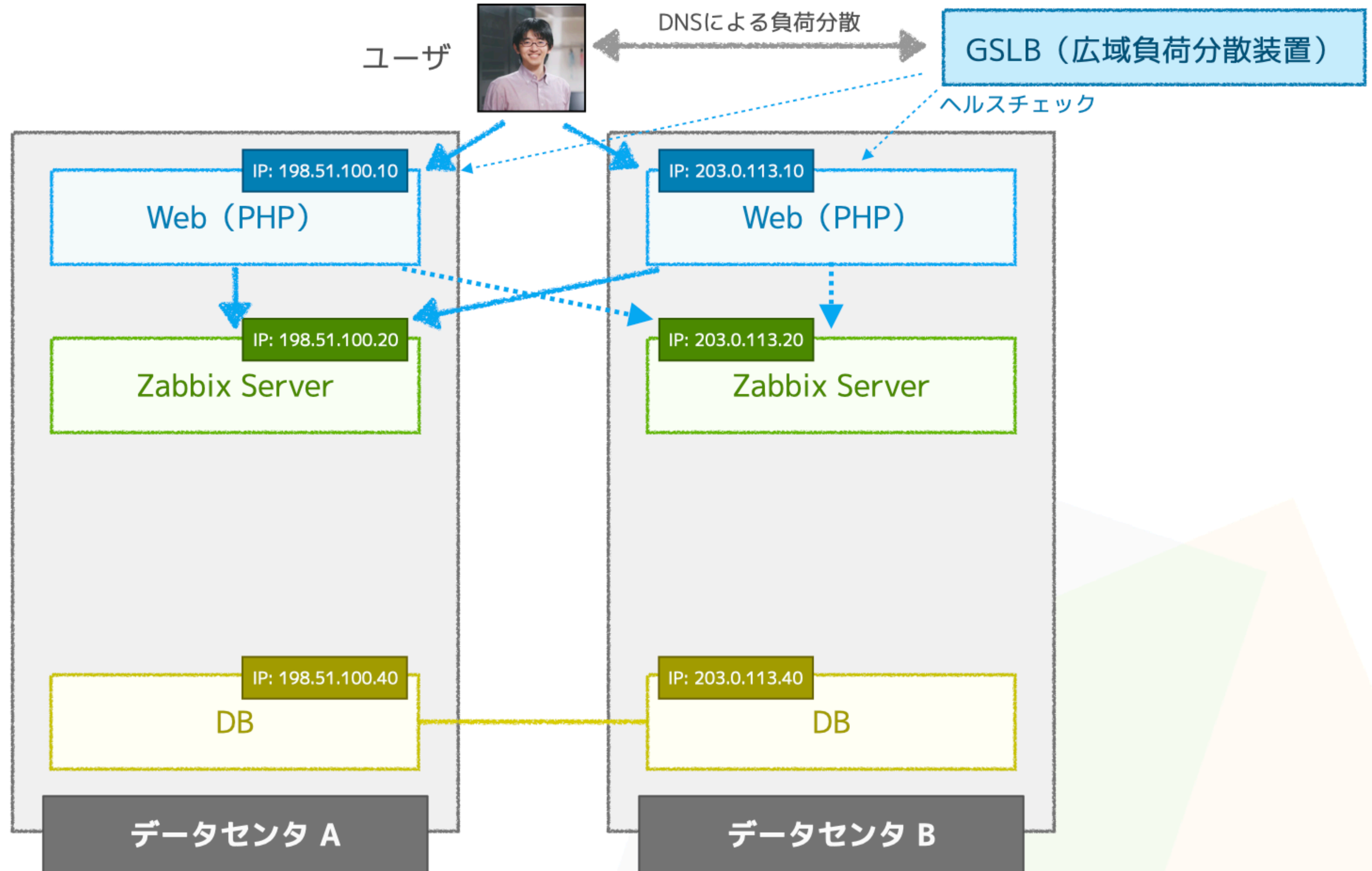


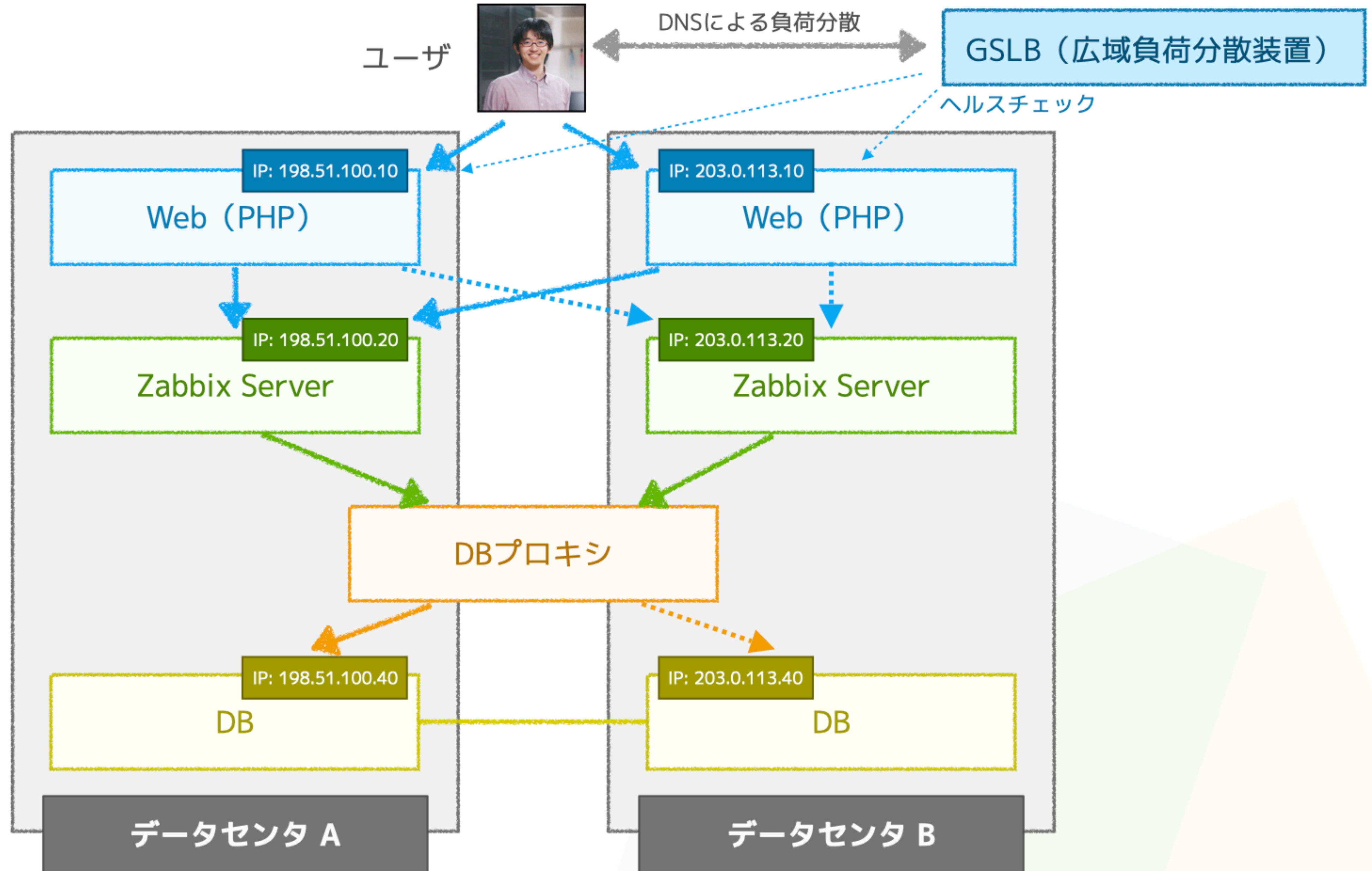


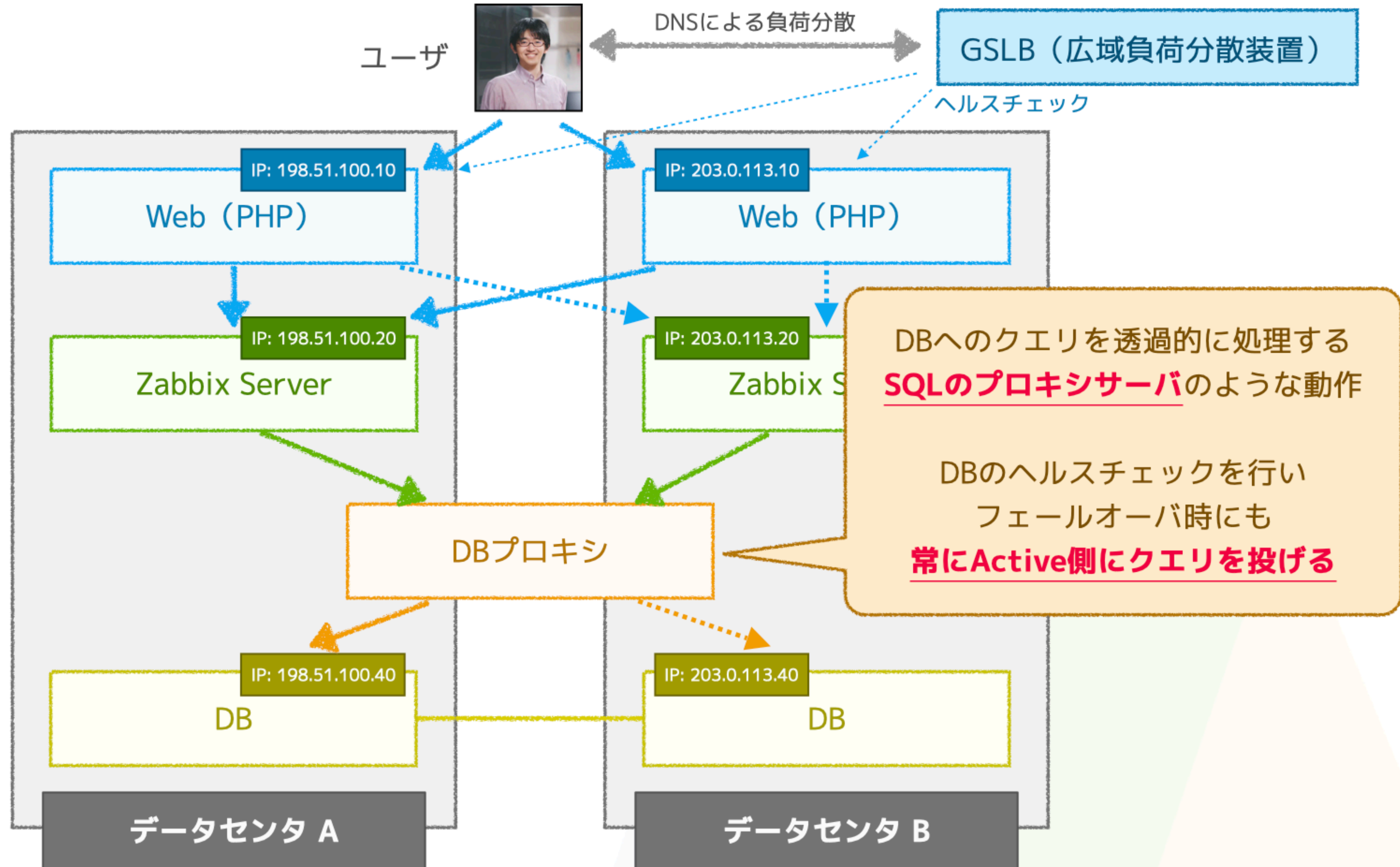


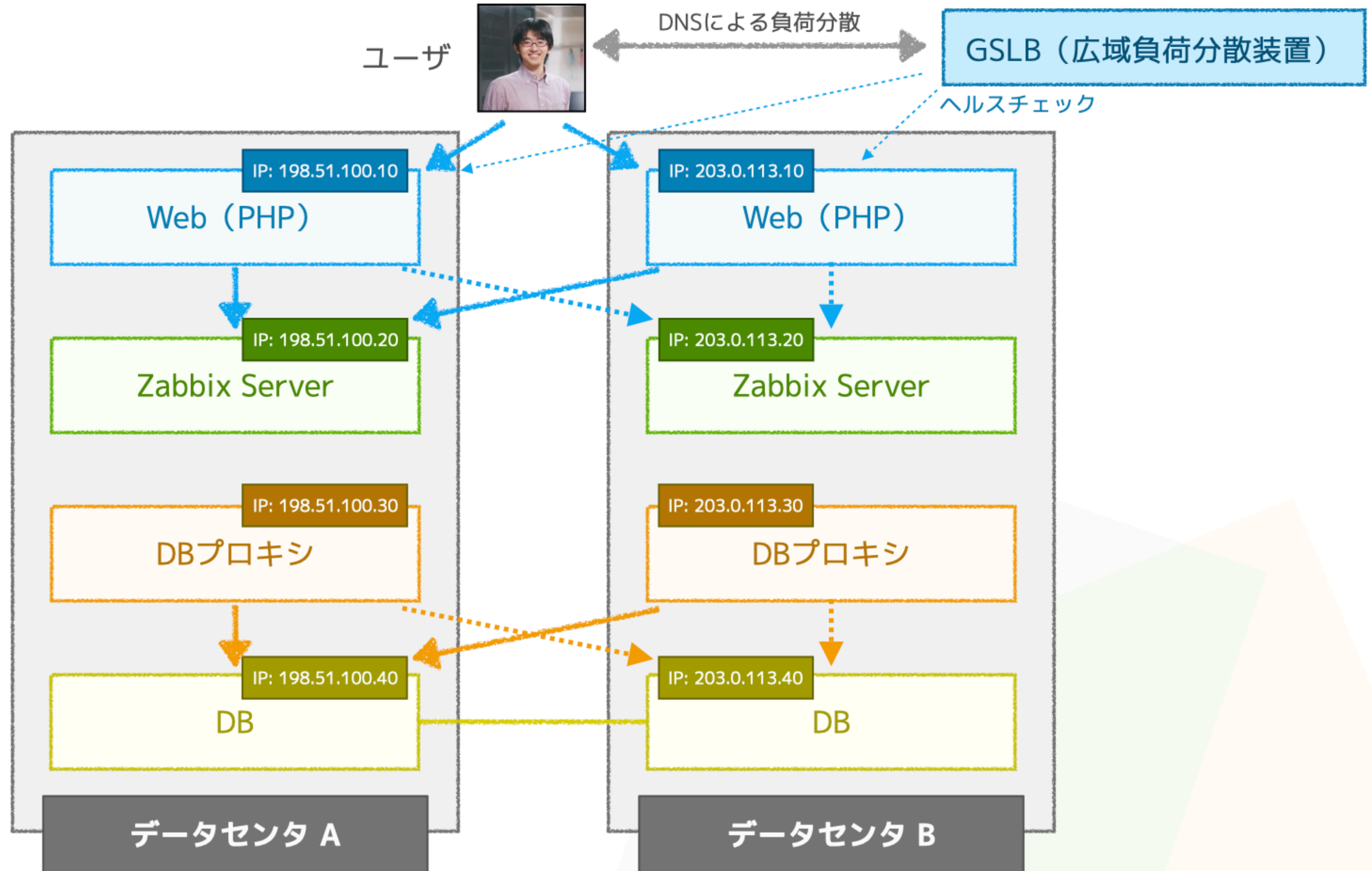


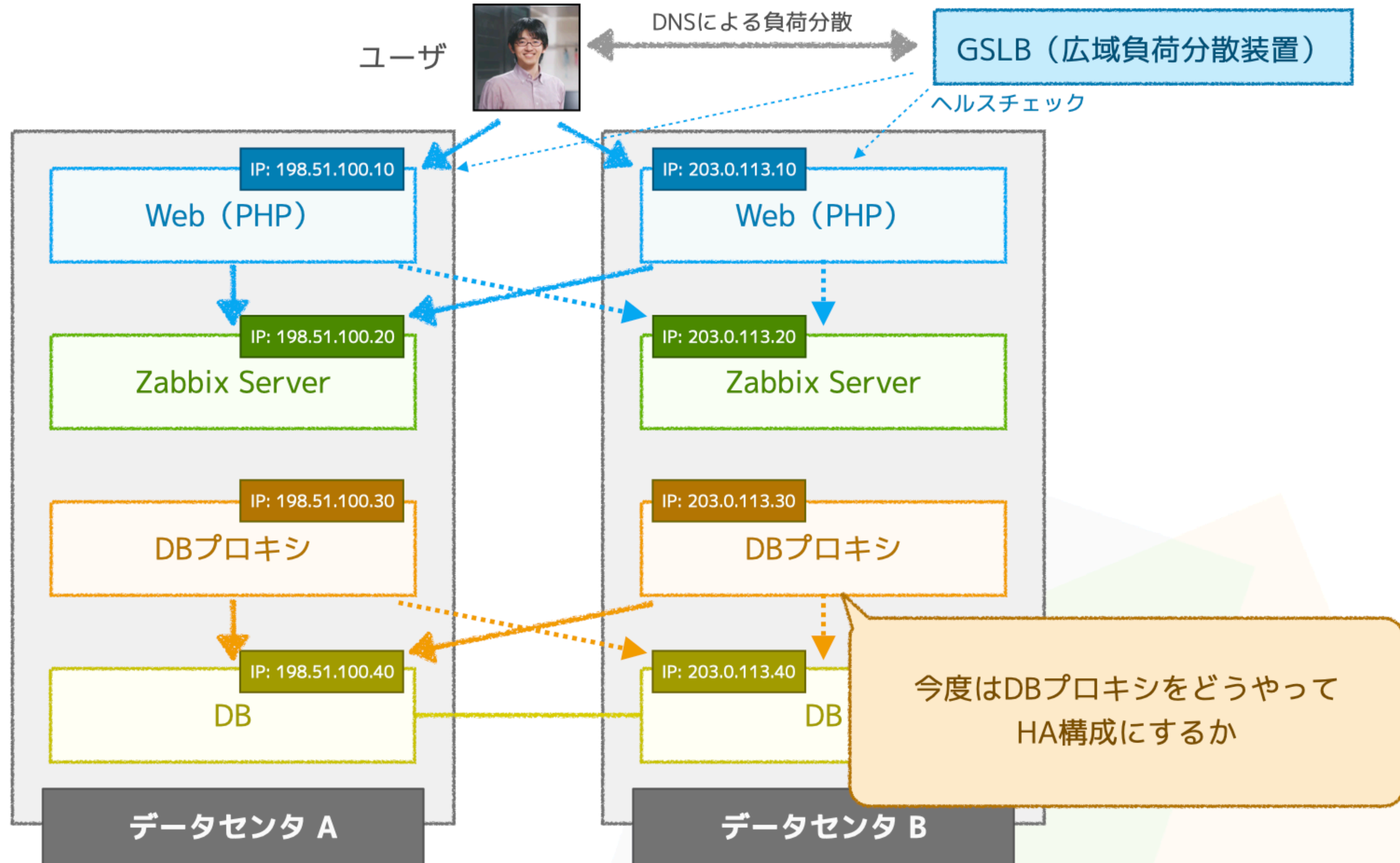


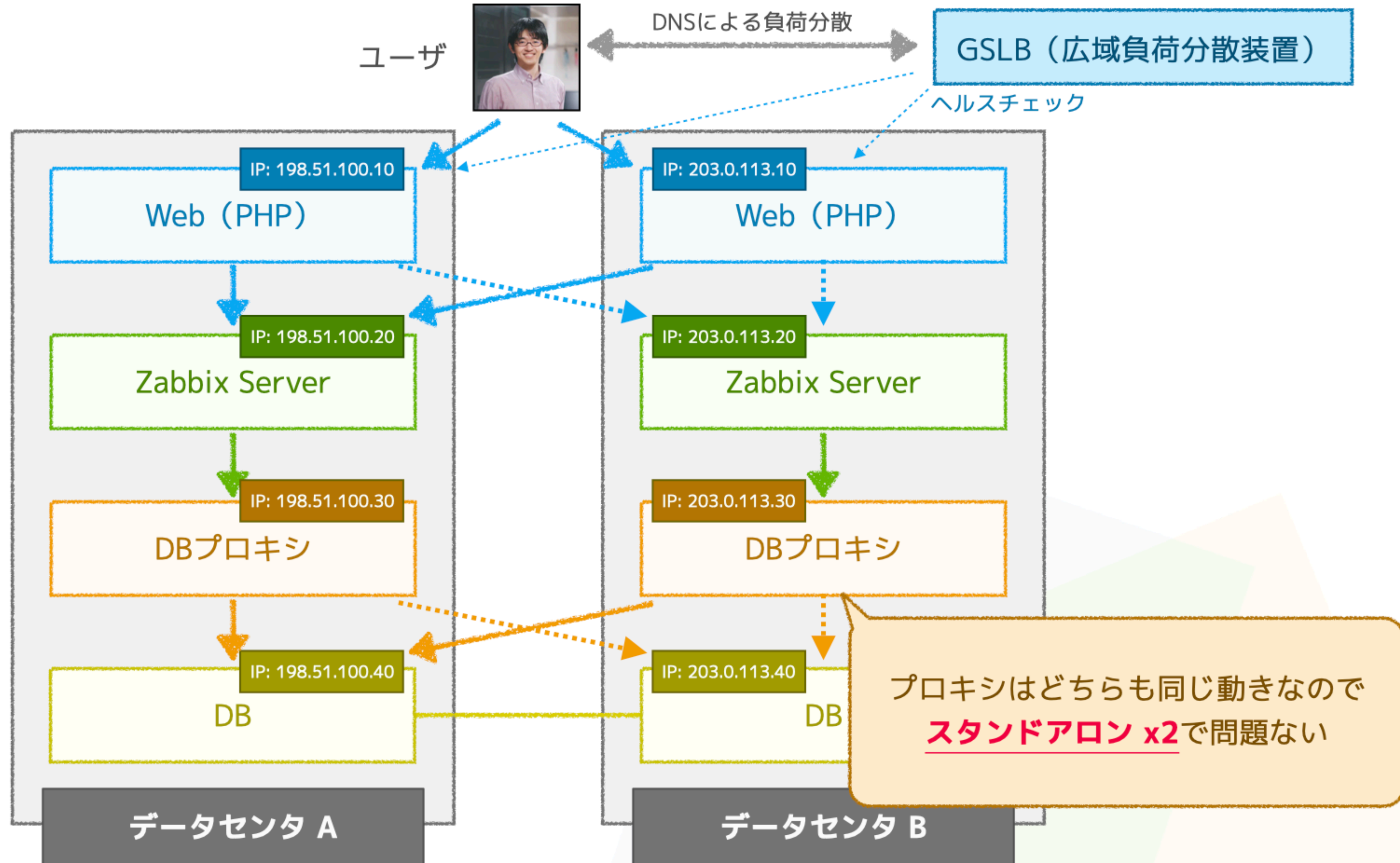


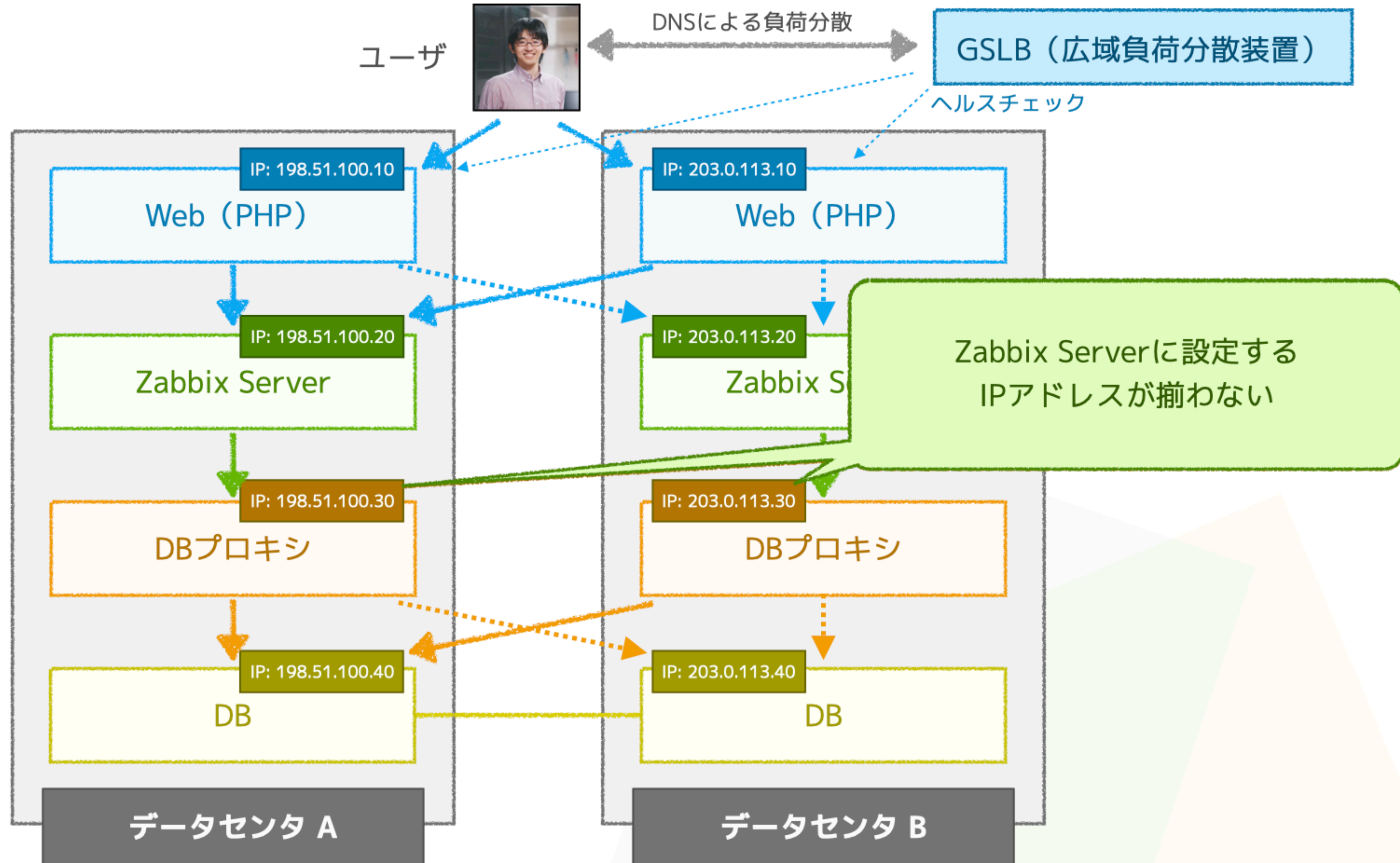


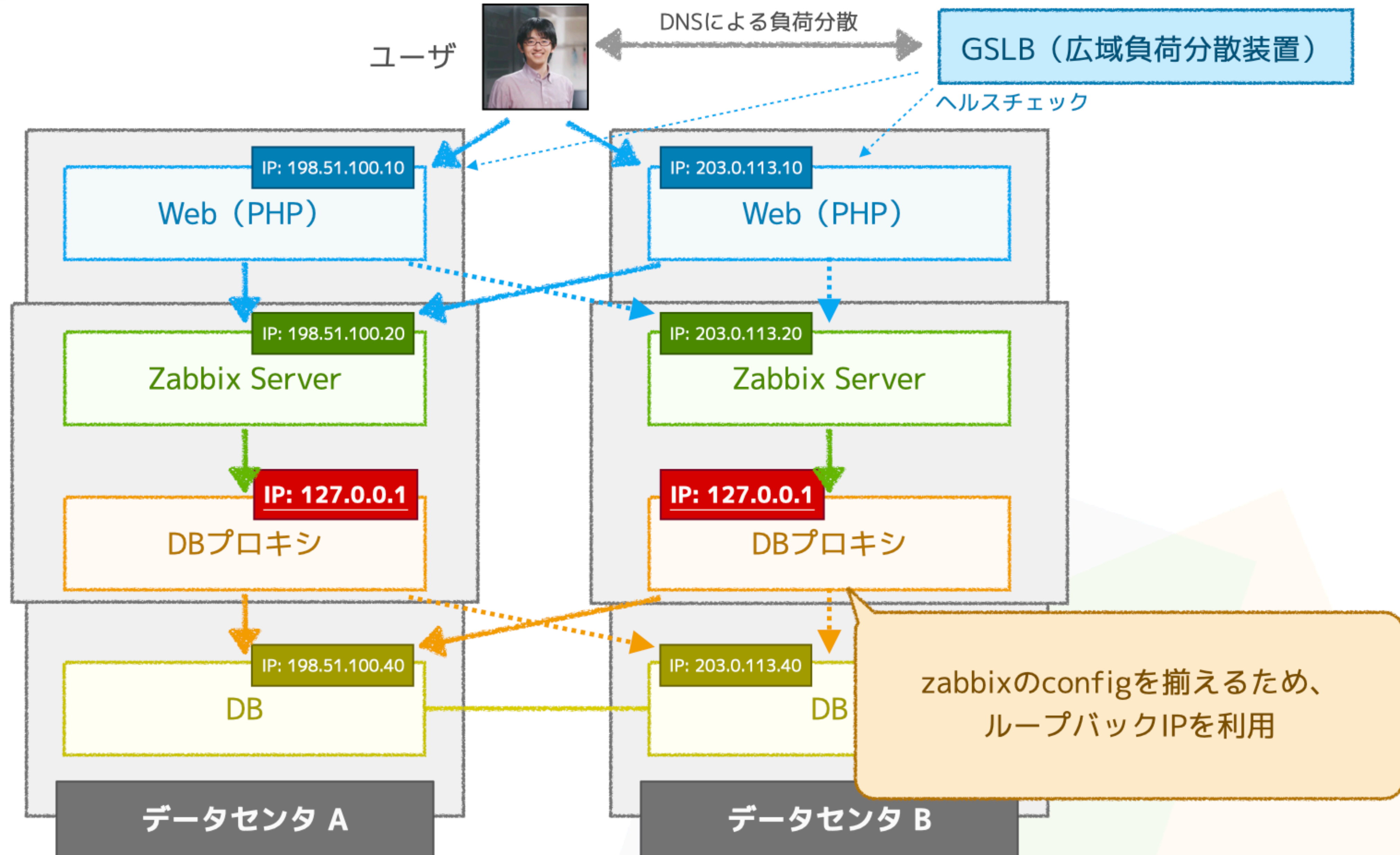








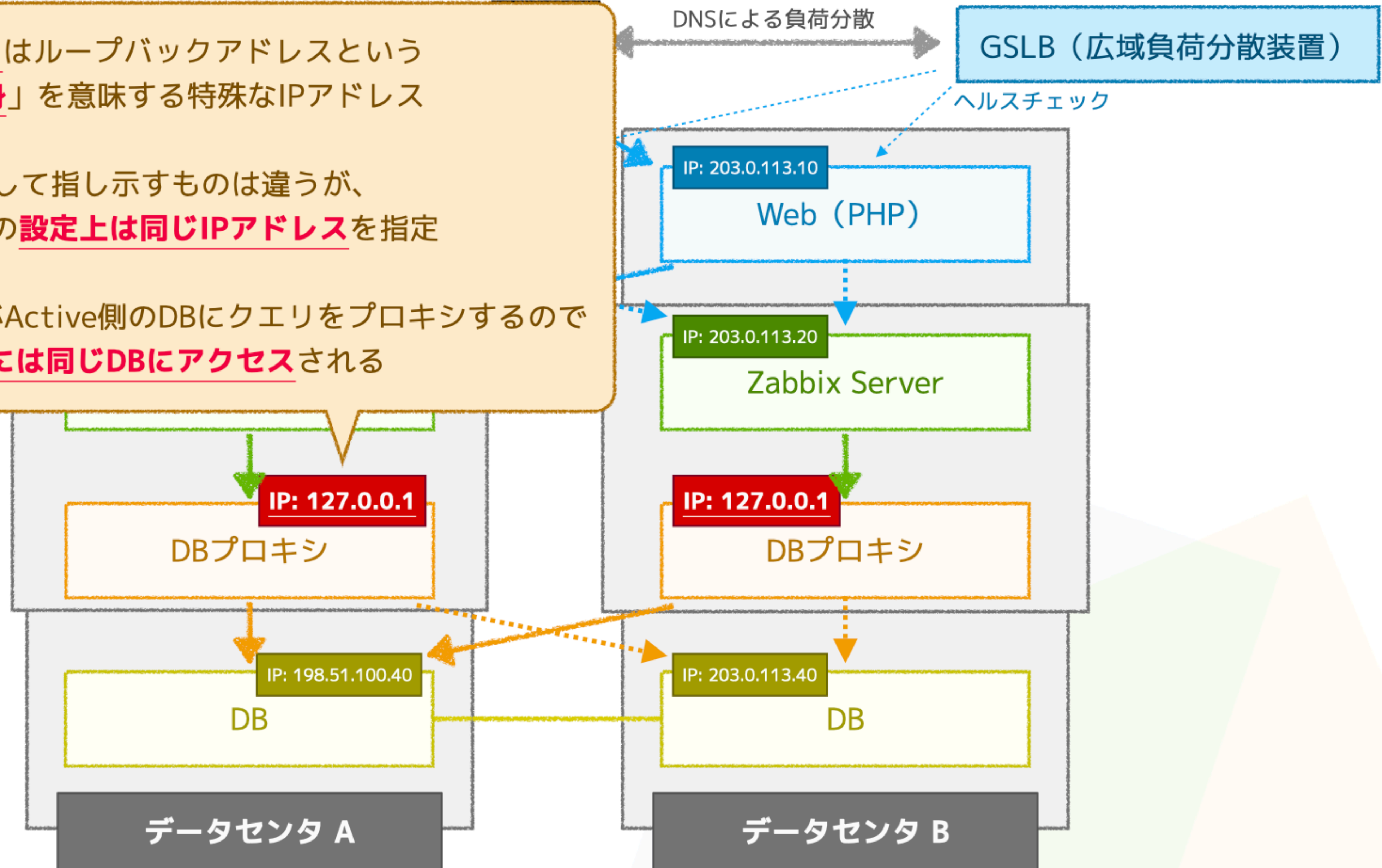




127.0.0.1はループバックアドレスという
「**自分自身**」を意味する特殊なIPアドレス

実体として指し示すものは違うが、
Zabbix HAの**設定上は同じIPアドレス**を指定

最終的にはPgpool-IIがActive側のDBにクエリをプロキシするので
最終的には同じDBにアクセスされる



127.0.0.1はループバックアドレスという
「自分自身」を意味する特殊なIPアドレス

実体として指し示すものは違うが、
Zabbix HAの設定上は同じIPアドレスを指定

最終的にはPgpool-IIがActive側のDBにクエリをプロキシするので
最終的には同じDBにアクセスされる

DNSによる負荷分散

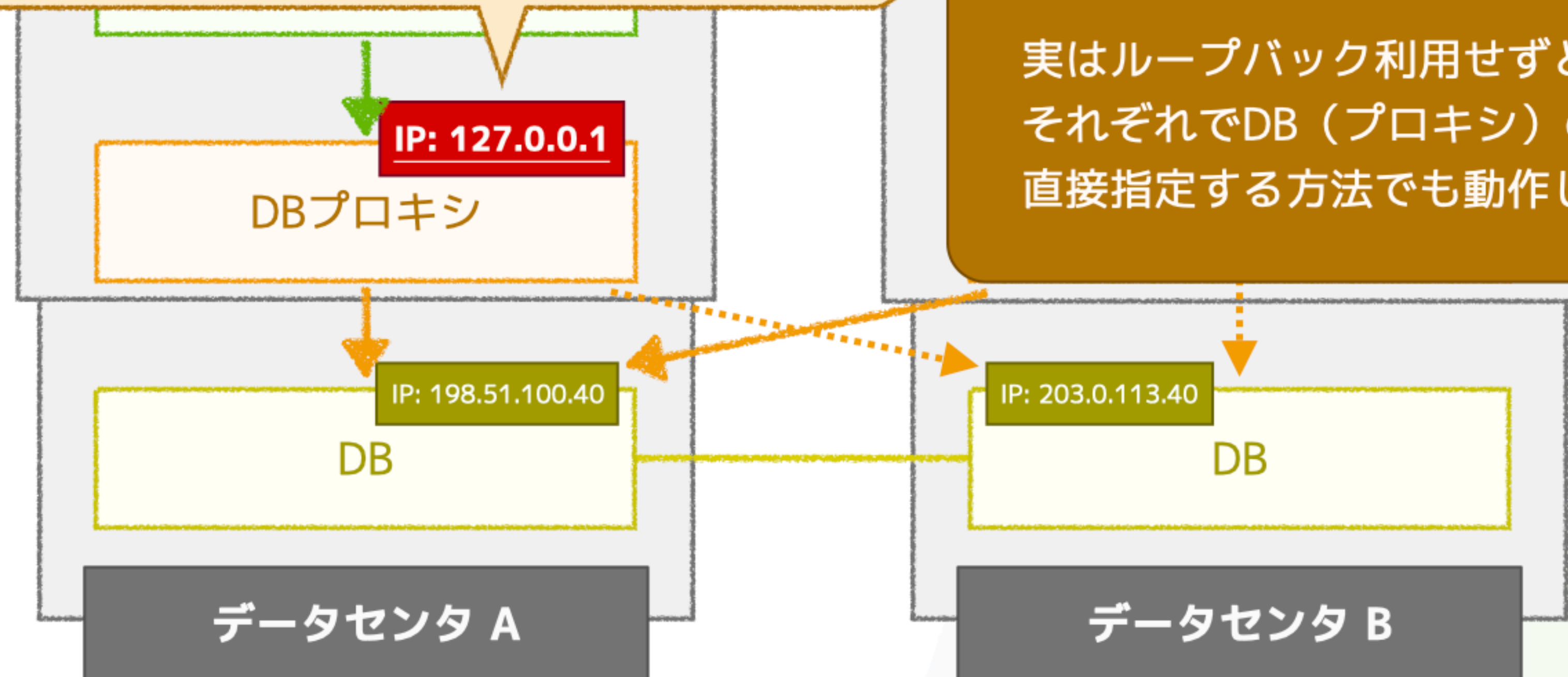
GSLB（広域負荷分散装置）

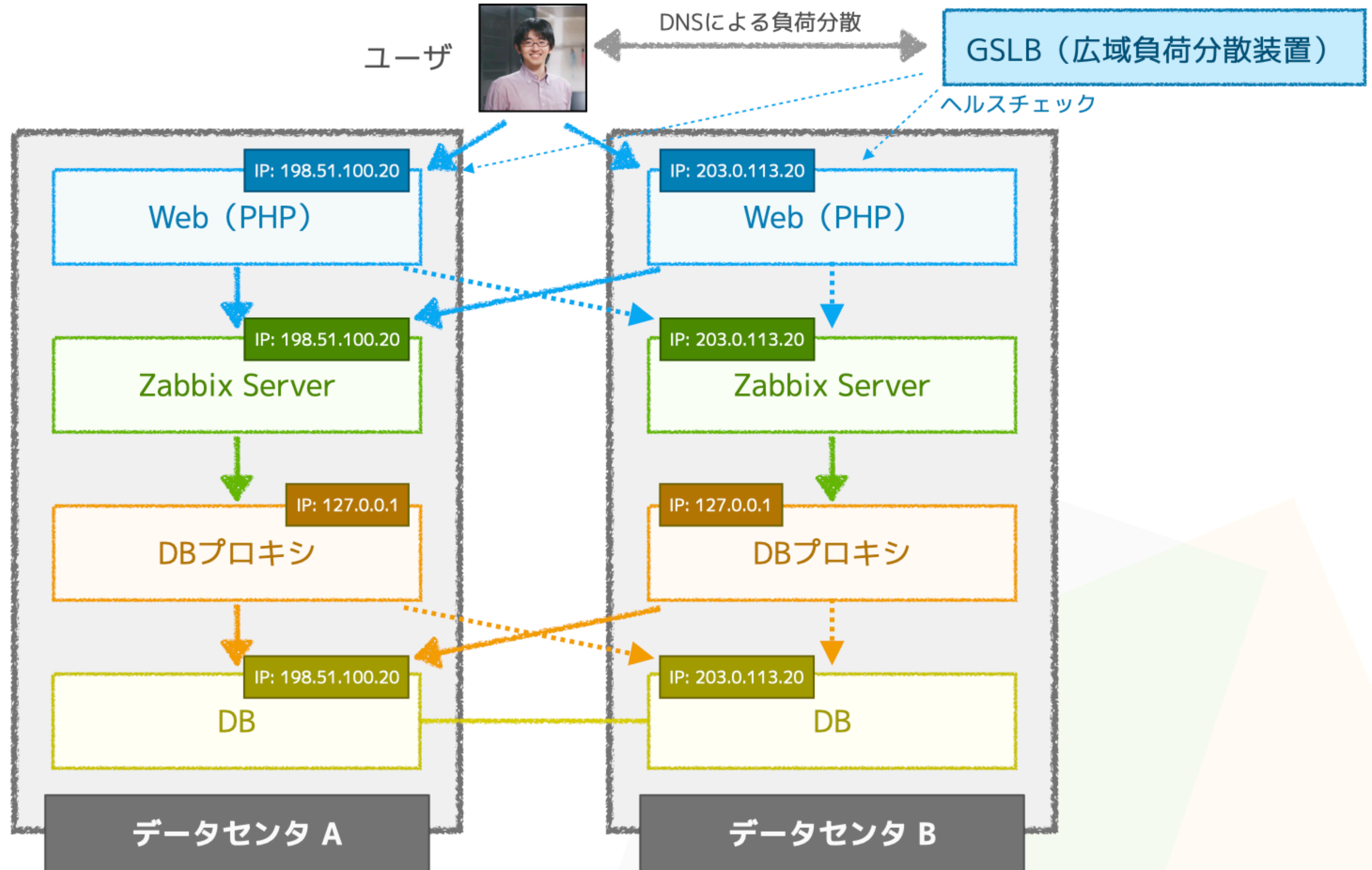
ヘルスチェック

構築当時、公式ドキュメントのHAの項の
“uses the same database
（同じデータベースを使用）”

の文言を、「DB設定には同じ値を指定」と
取り違えていました、、、

実はループバック利用せずとも、
それぞれでDB（プロキシ）のIPを
直接指定する方法でも動作しました





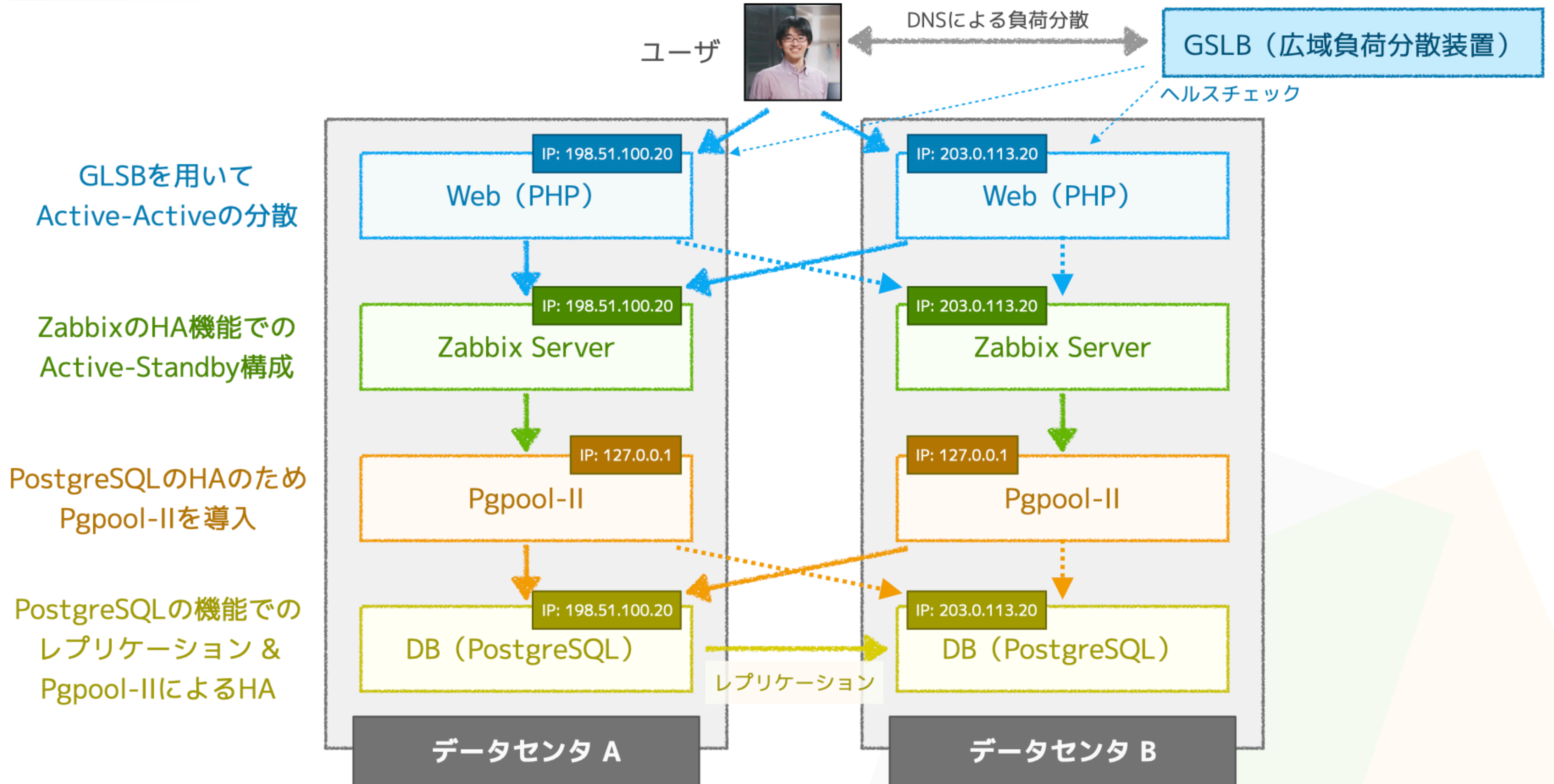
- MySQL
 - 既存のzabbix環境で利用していた
 - DBプロキシにはProxySQLというツールを利用
- PostgreSQL
 - 社内の他のインフラ系ツールで利用している
 - DBプロキシにはPgpool-IIというツールを利用

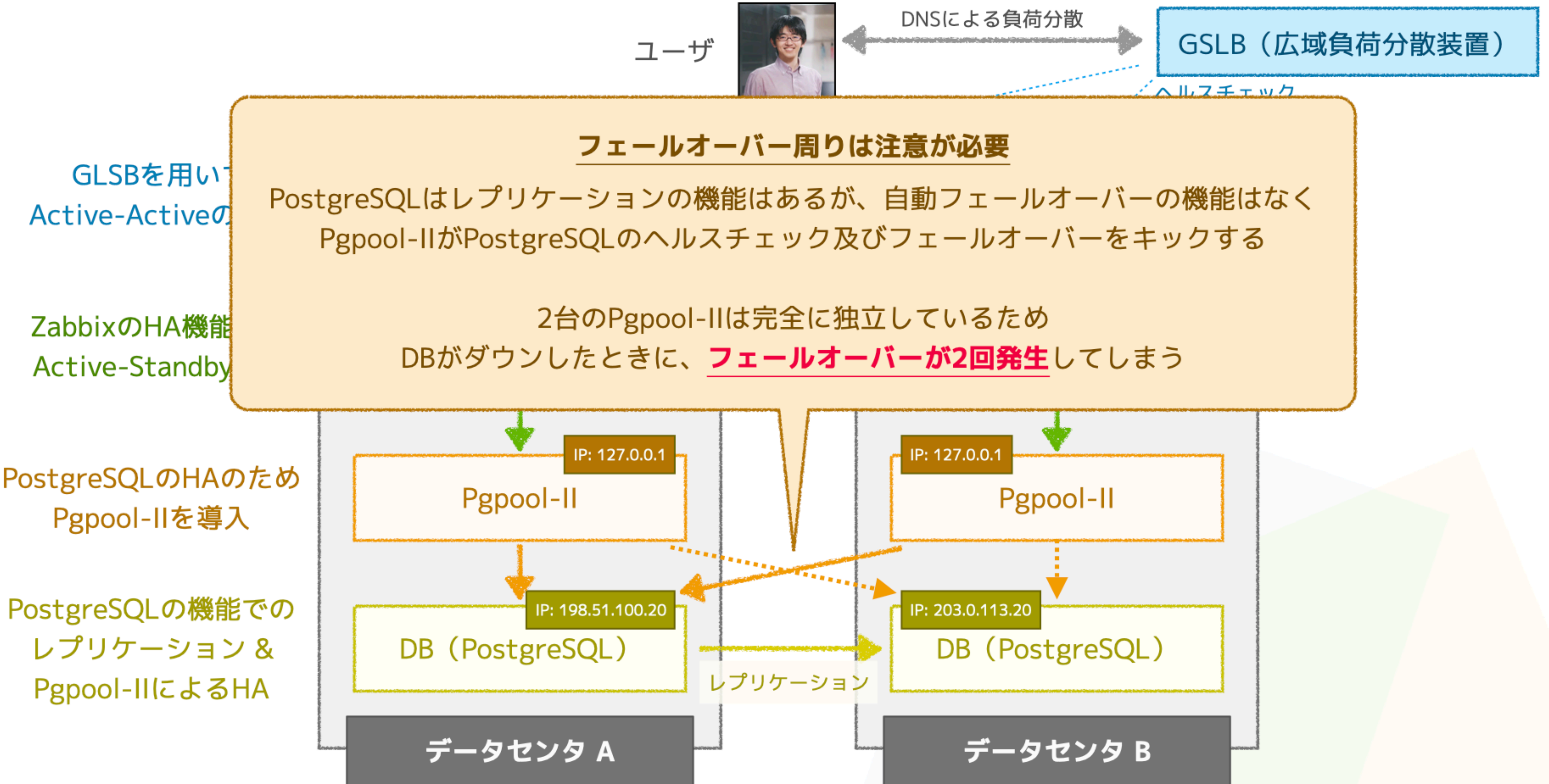
本業はネットワークエンジニアリングのため、
DB固有のトラブルにハマりたくない

ノウハウを再利用できるほうが嬉しい



PostgreSQL（ストリーミングレプリケーション）
+
Pgpool-II





- Pgpool-IIが定期的にバックエンドのpostgresqlにヘルスチェックを行う
- ヘルスチェックが失敗するとフェールオーバースクリプトがキックされる
 - ヘルスチェックが失敗したノードをdownとしてマークし、プロキシ先から除外
 - Standbyだったノードを昇格させ、Activeとして振る舞うようにさせる
- 同期レプリケーションのため、Standby側もダウンした際には切り離しが必要



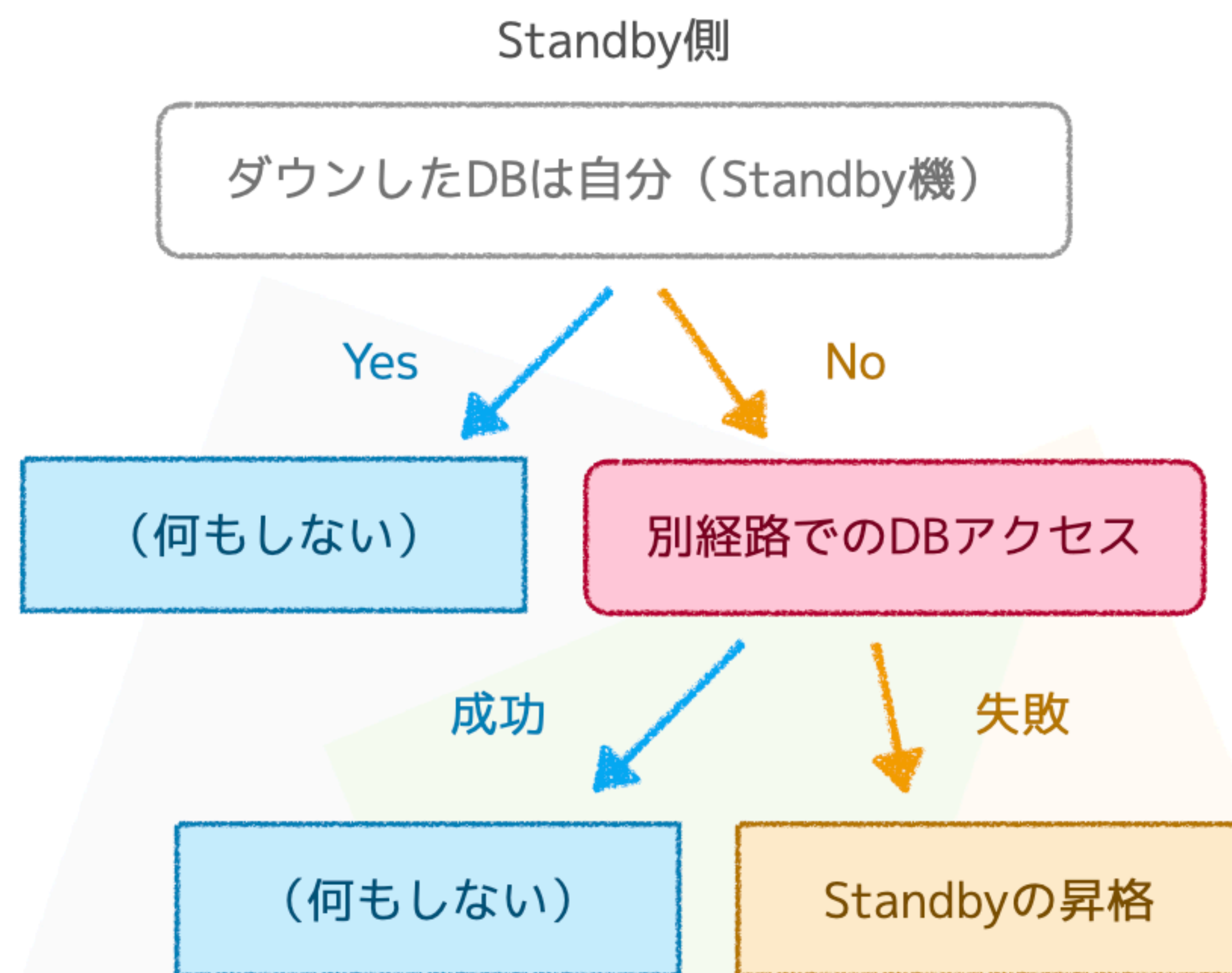
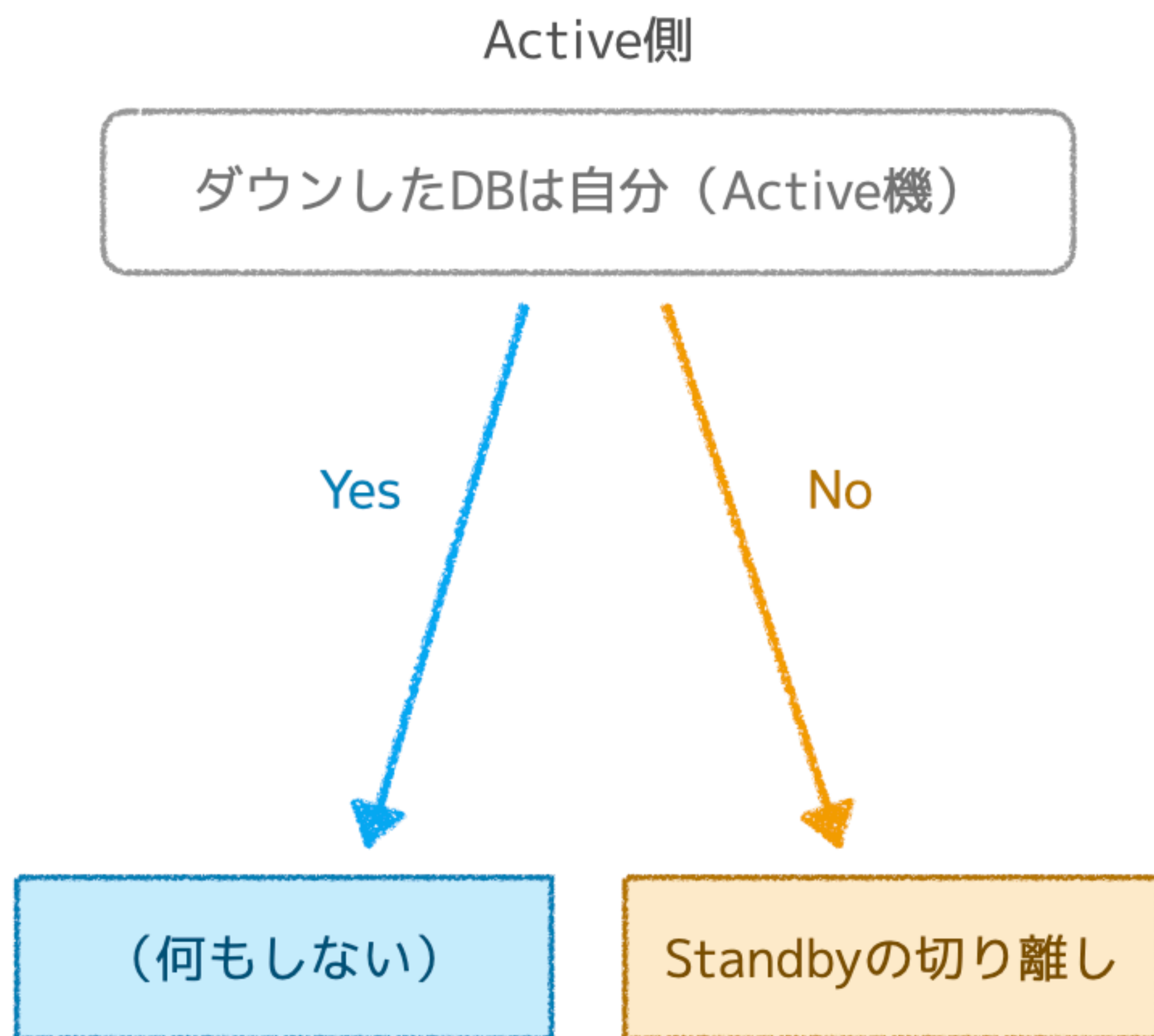
普通にフェールオーバーを仕込むと、、？

- フェールオーバーがダブる
- L3で冗長化すると経路上のNW障害によるヘルスチェック失敗がある
 - フェールオーバーを早まると、スプリットブレーンが発生してしまう

障害のパターン分けを行い、必要に合わせてフェールオーバーをキックするように

- Active側/Standby側のサーバ全体がハング
- Active側/Standby側のpgpoolがハング
- Active側/Standby側のpostgresqlがハング
- Active-Standby間のネットワーク疎通がダウン

- 各ホストのPgpool-IIは、同一ホスト上のPostgreSQLのフェールオーバーのみ実施
- Active・Standbyそれぞれで独立してフェールオーバーをキック



- 珍しい（想定されていないだろう）構成のため、何が正解であるかがわからない
 - 本当に求めているものができたか検証を繰り返し行うひつようがある
 - 要件定義をしっかりと行い、1つずつテストを実施した
- Pgpool-IIのフェールオーバー
 - これも想定されていないだろう構成のため起こり得るパターンを列挙して1つずつテスト
- 今後のアップデート
 - アップデートによりこの構成だと上手くいかない機能が出てきてしまうのではないか
 - 都度検証が必要
 - それはそれで楽しいからヨシ？