クラウド環境でサービス断しない Act/Sbyクラスタを設計してみた

KDDI株式会社　宮崎 亮輔
KDDI株式会社　横山 真悟

Tomorrow, Together KDDI
おもしろいほうの未来へ。 au

お断り

本講演資料は、

・講演者の見解を述べたものであり、会社としての見解・業界としての見解ではありません。
・スクリーンショット等にて引用させていただいたものは2024/9/24現在の情報です。
　資料同ページ内に引用元URLを掲載しております。
・著作権フリーもしくは無償利用可能なイラストを一部で利用しております。

アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

アジェンダ

1	**講演者紹介**
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

講演者紹介

宮崎 亮輔

KDDI株式会社
コア技術統括本部
エンジニアリング推進本部

業務	auひかり系システムの開発/運用
趣味	野球観戦 自宅サーバ (電気代が最近の悩み)

横山 真悟

KDDI株式会社
コア技術統括本部
エンジニアリング推進本部

業務	法人お客様向けシステムの開発/運用
趣味	大谷選手応援 ソフトテニス

アジェンダ

1	講演者紹介
2	**KDDI「手の内化」の紹介**
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

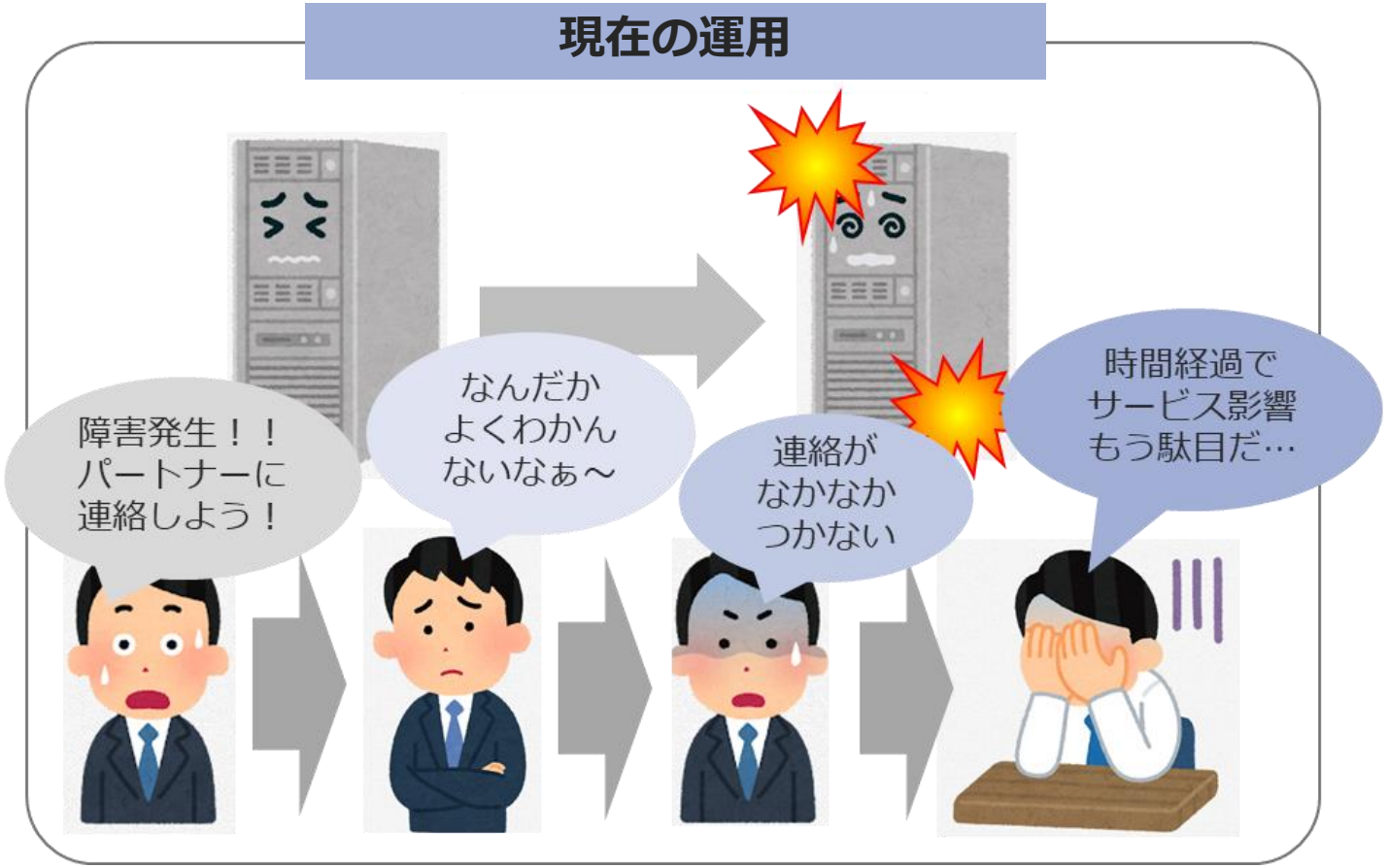
つなぐことが私たちの使命
24時間365日、通信を守る

KDDIの目指す姿

障害をゼロにはできない

すぐに復旧することが重要

すぐに復旧したいが・・・



設備の理解が浅いと・・・

障害が長期化し
サービス影響が出るケースも

目指す運用

設備を自分達で理解できている

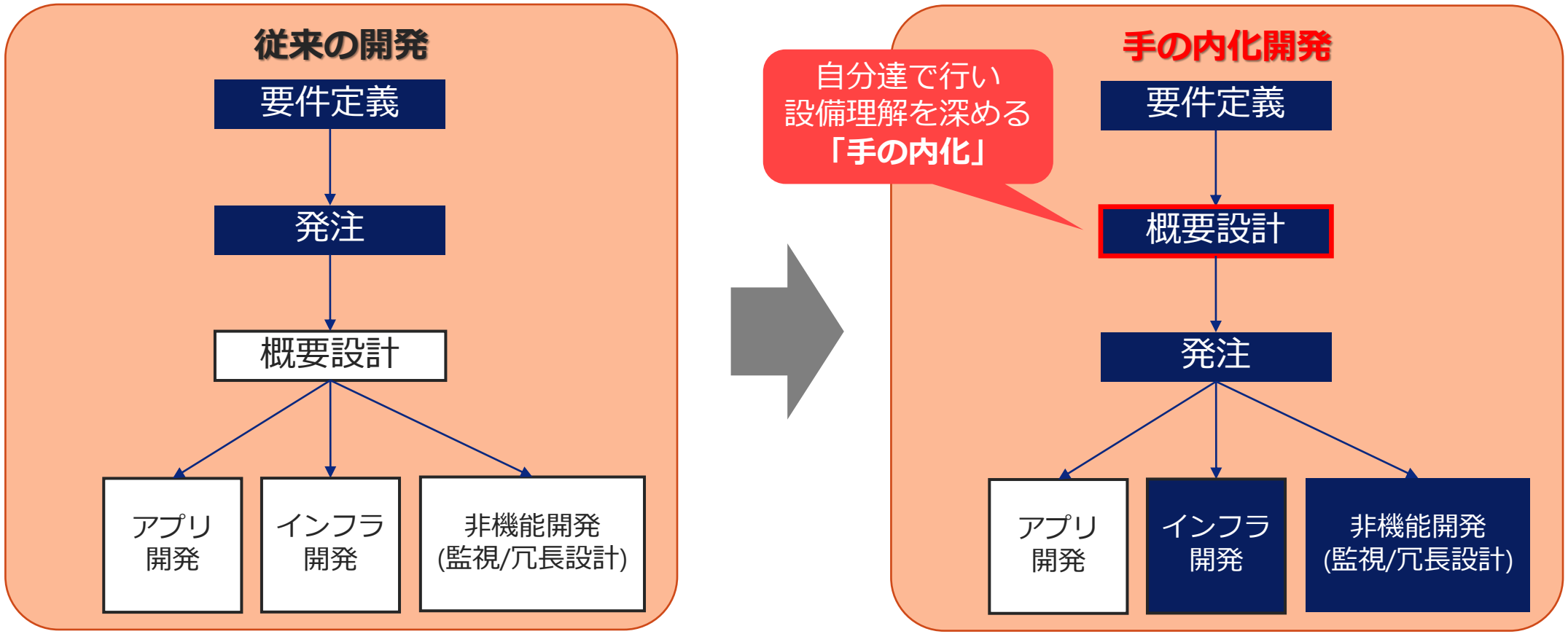
すぐに障害復旧できる！

目指す運用を実現するために

「手の内化」で設備を理解する

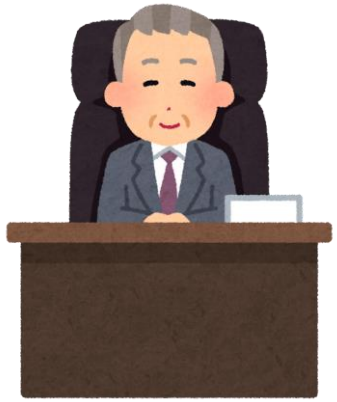
: KDDI開発

: パートナー様開発



そうは言っても・・・

・KDDIは数百以上の設備を保有しており、限られた人数で各システムを開発している
・1システム開発チームで**概要設計を行うノウハウが不足している**



（えらい人）

これからは**「手の内化」**だから！
みんな頑張れ～

オンプレばかりやってきて
クラウド設計なんて
初めてだゾ・・・

冗長設計って
どう考えればいいの？

この要件だと
どんな**データベース設計**が
最適なの？

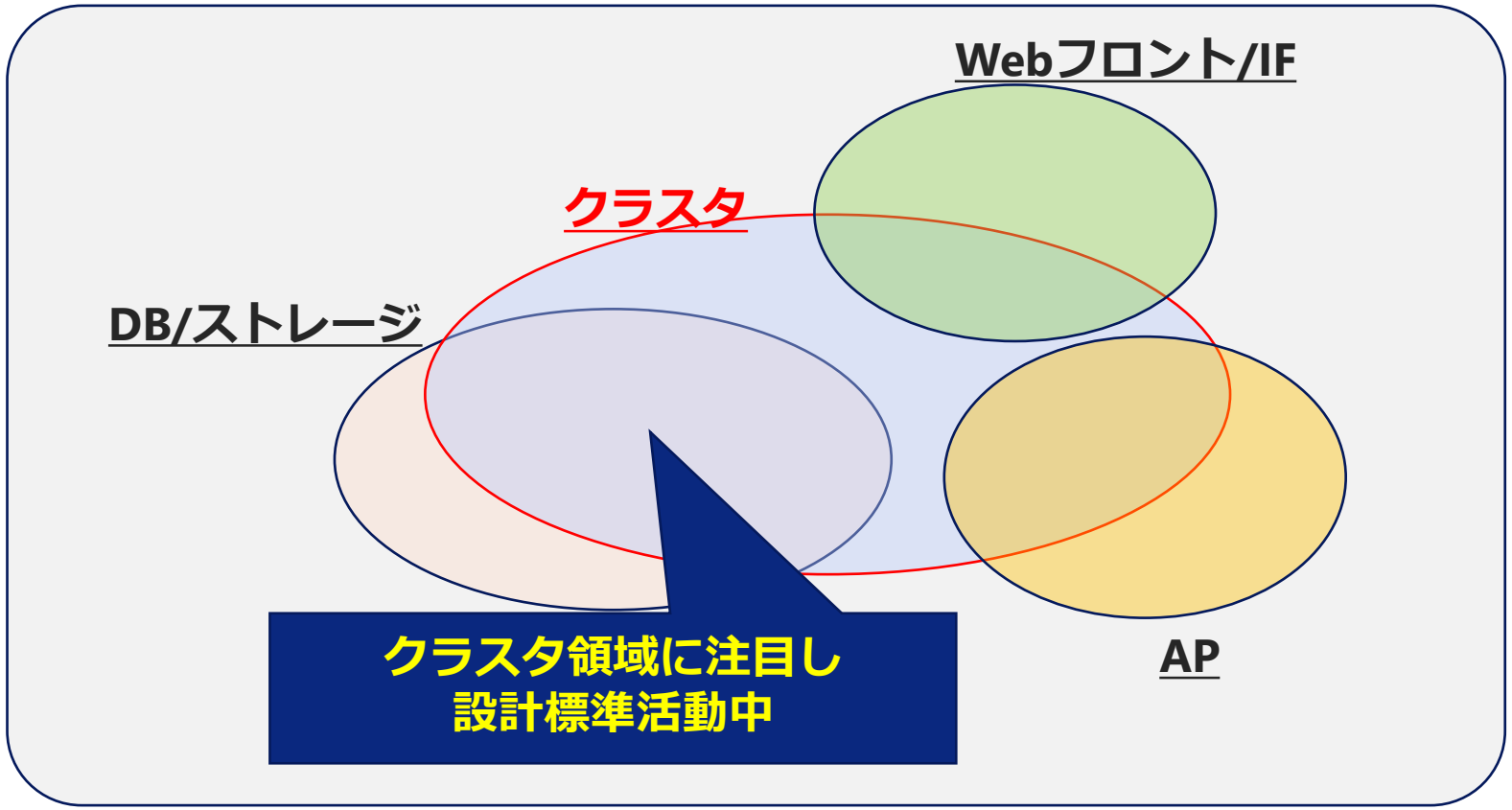
アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	**設計標準活動の紹介**
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

設計標準活動

・システムごとに異なる**設計を標準化（ベストプラクティスを作成）**し、サービスの早期復旧を図る
・Web/AP/DBの信頼性に関わる**クラスタ**を、第一の標準化ターゲットとして選定

レジリエンス関連の設計要素



※ AWS、インフラ等の別領域の標準化チームも活動中！

活動内容

① 【設計標準】クラスタ設計の標準化

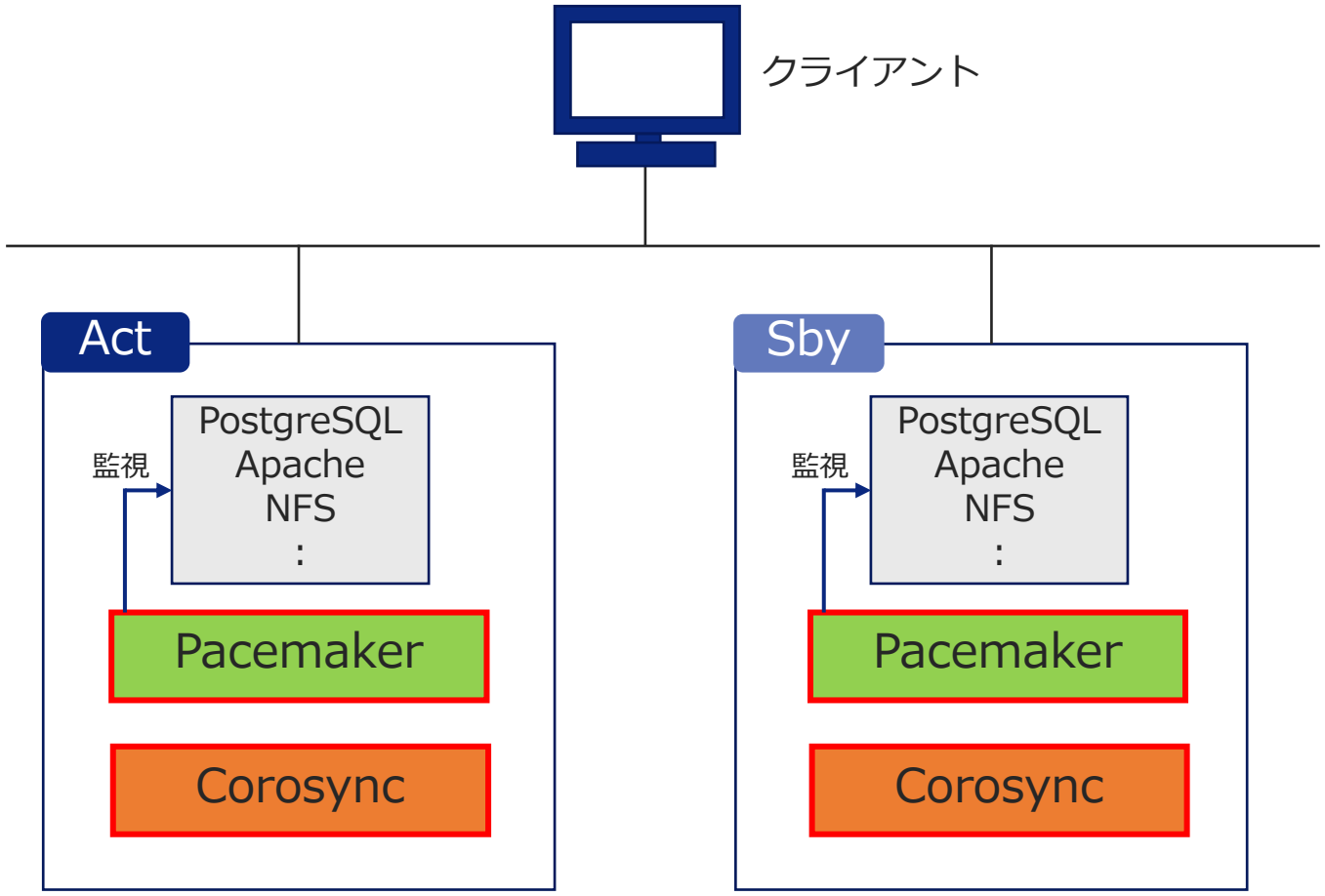
② 【手の内化運用】監視ツール・復旧手順の標準化

設計標準のご説明 の前に…

アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	**前提知識①：Pacemaker/Corosync**
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

HAクラスタを実現するポピュラーなOSS



Pacemaker/Corosyncの役割

- HAクラスタOSSのデファクトスタンダード
- Pacemaker:各サーバのリソース監視とHA設定
- Corosync:サーバ間HA通信

できること

- ディスクやDBなどのリソースそれぞれを
  リソースエージェントで**状態監視**

- 各リソースの状態変化(停止・起動等)の際、
  他サーバーや他プロセスの状態等を加味して
  どのようにフェールオーバーするかを細かく制御可能

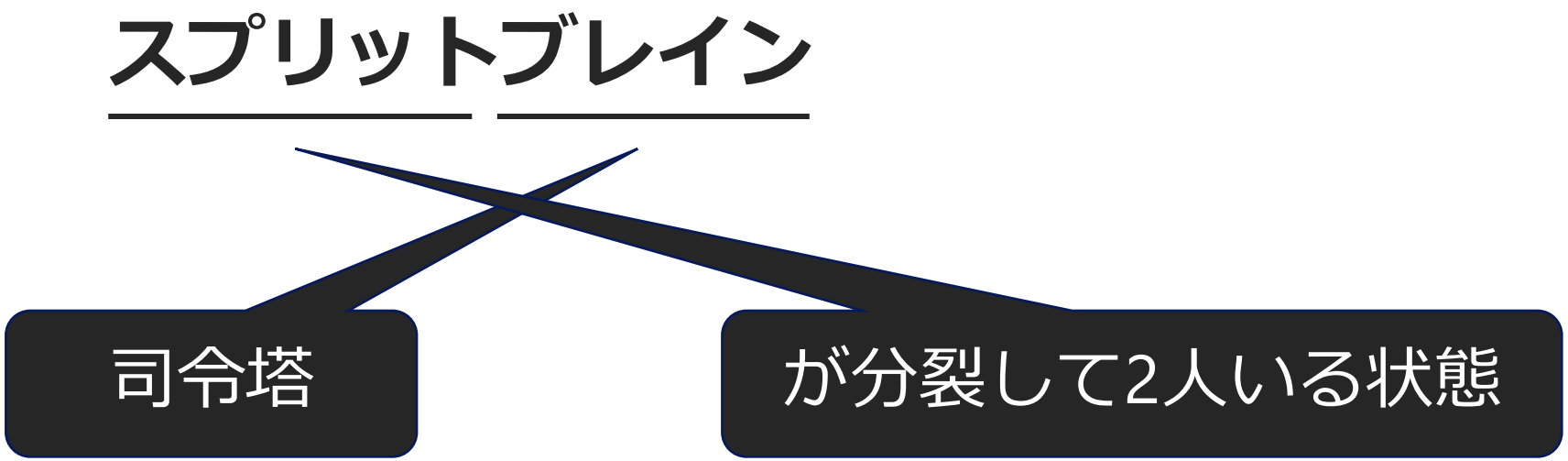
公式応援キャラクターもいるよ！

Linux-HA Japanプロジェクトのこれまでとこれから
https://www.slideshare.net/ksk_ha/linuxha-japan

アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	**前提知識②：スプリットブレイン**
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

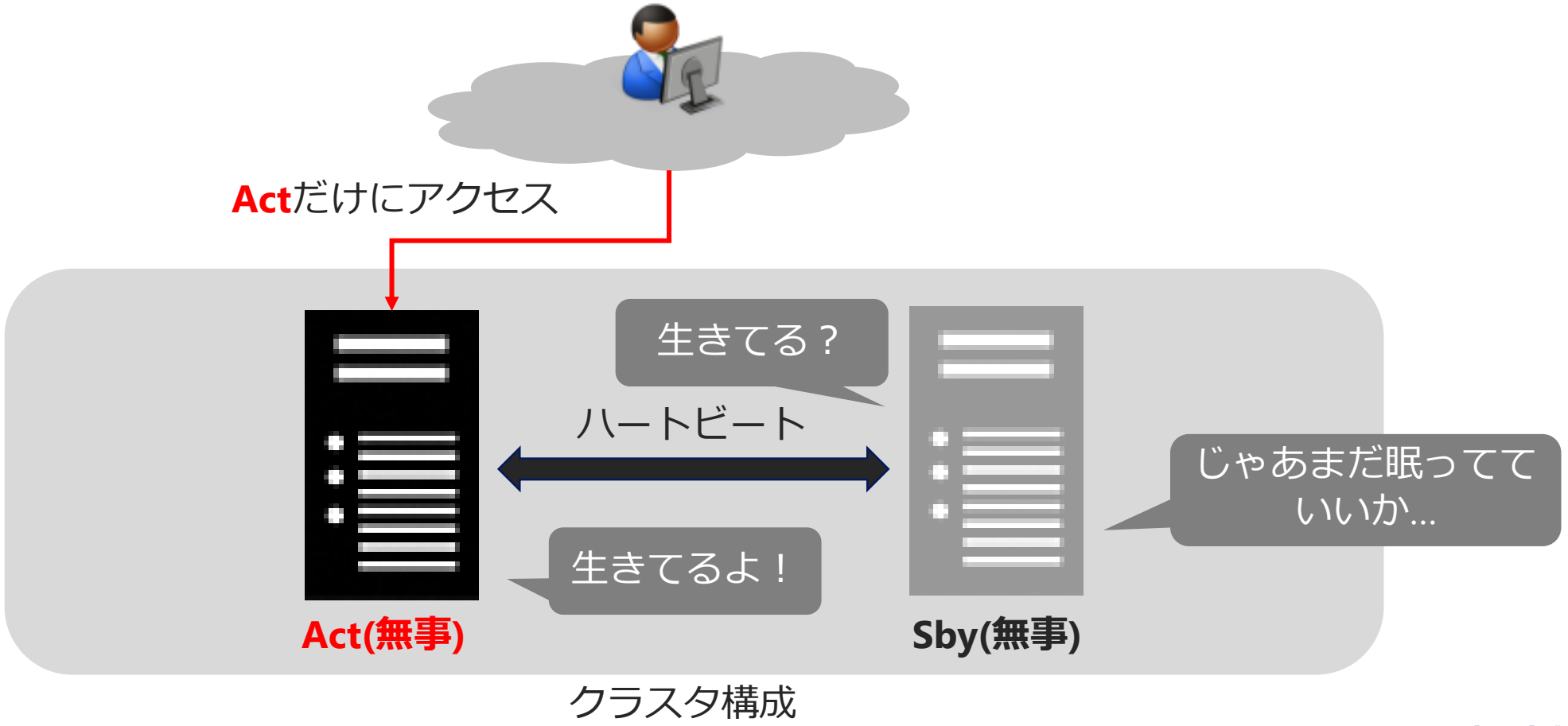
スプリットブレインとは



⇒システムが混乱してしまう

具体的にどういう状態？

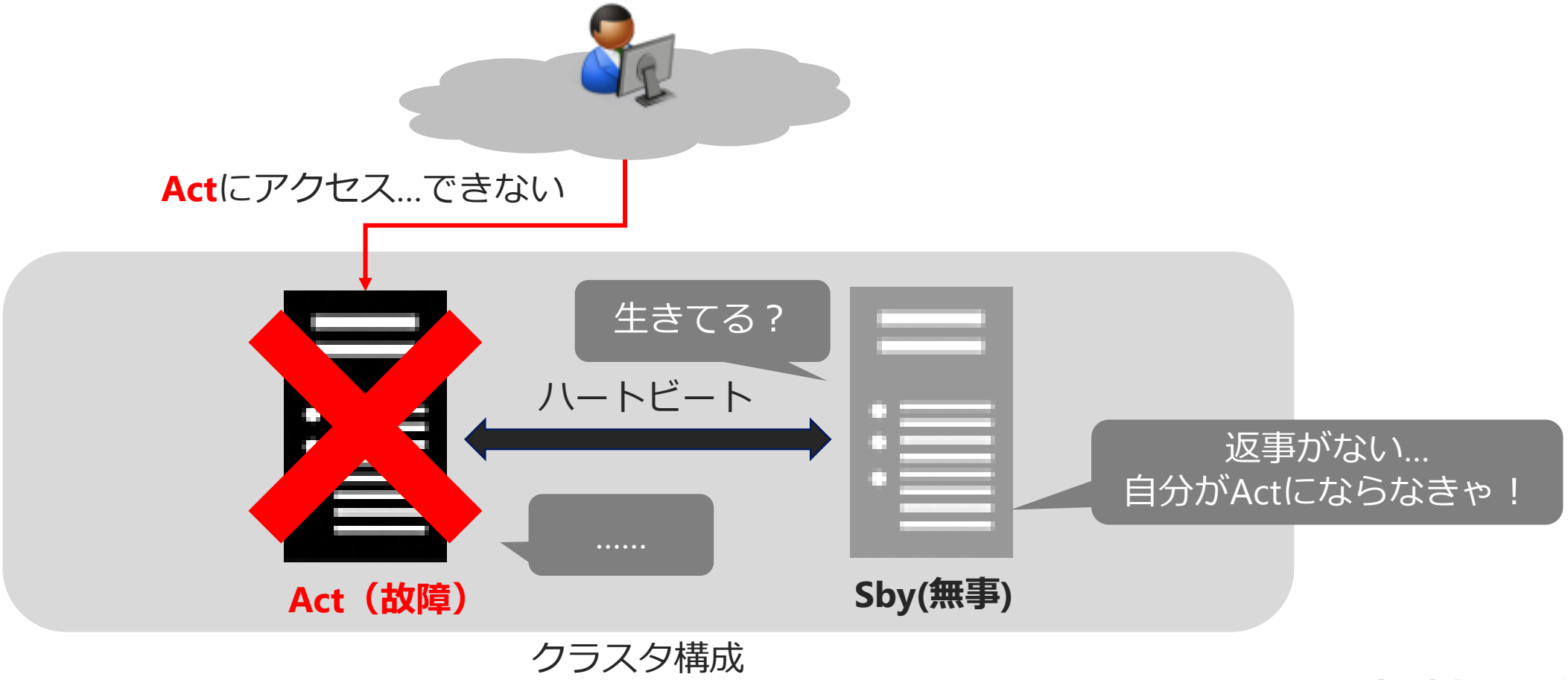
まず、一般的なAct/Sbyのサーバ2台構成のイメージは下記



クラスタ構成

具体的にどういう状態？

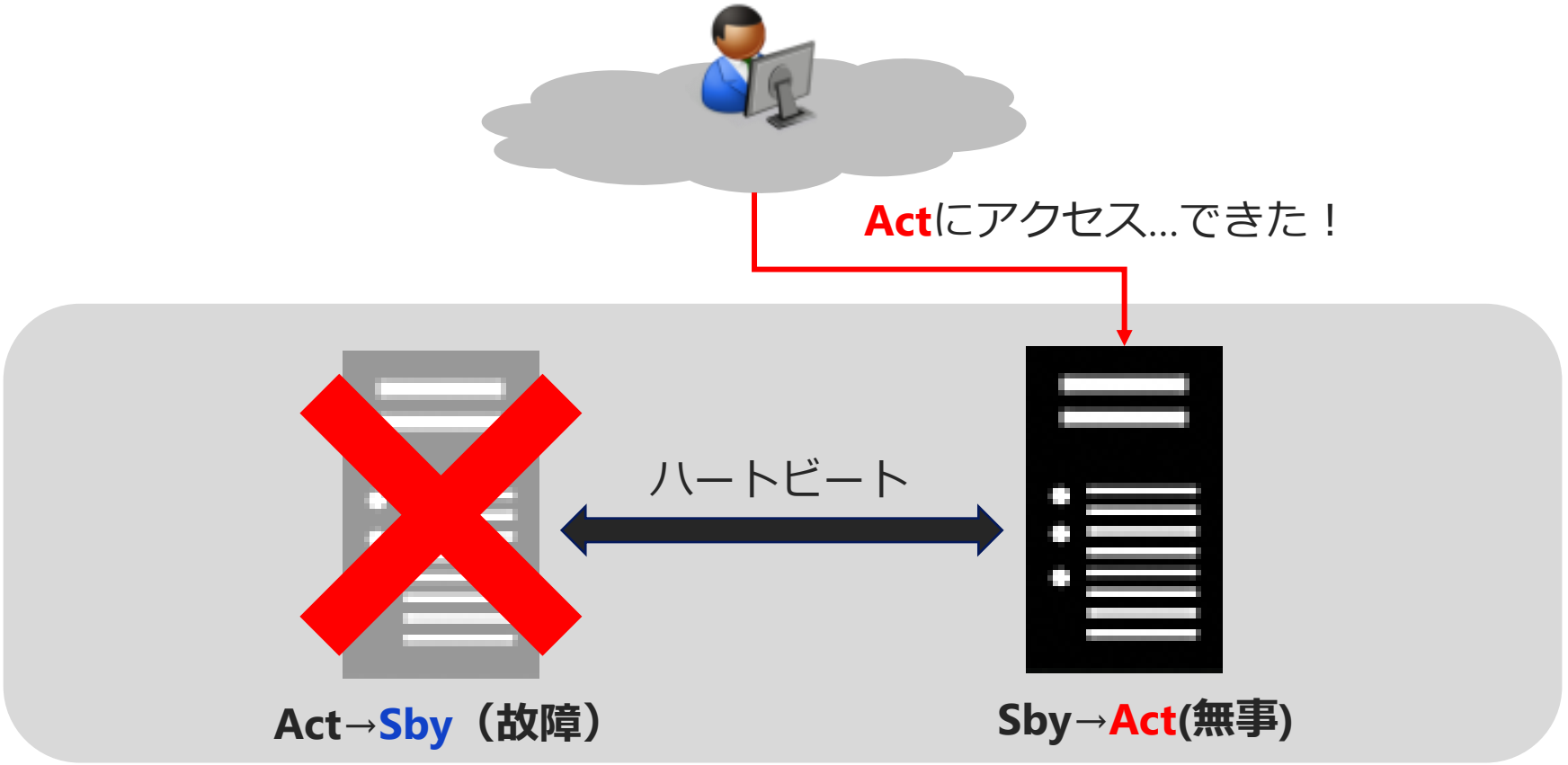
生死確認で**返事がない場合**にはSbyがActに昇格



クラスタ構成

具体的にどういう状態？

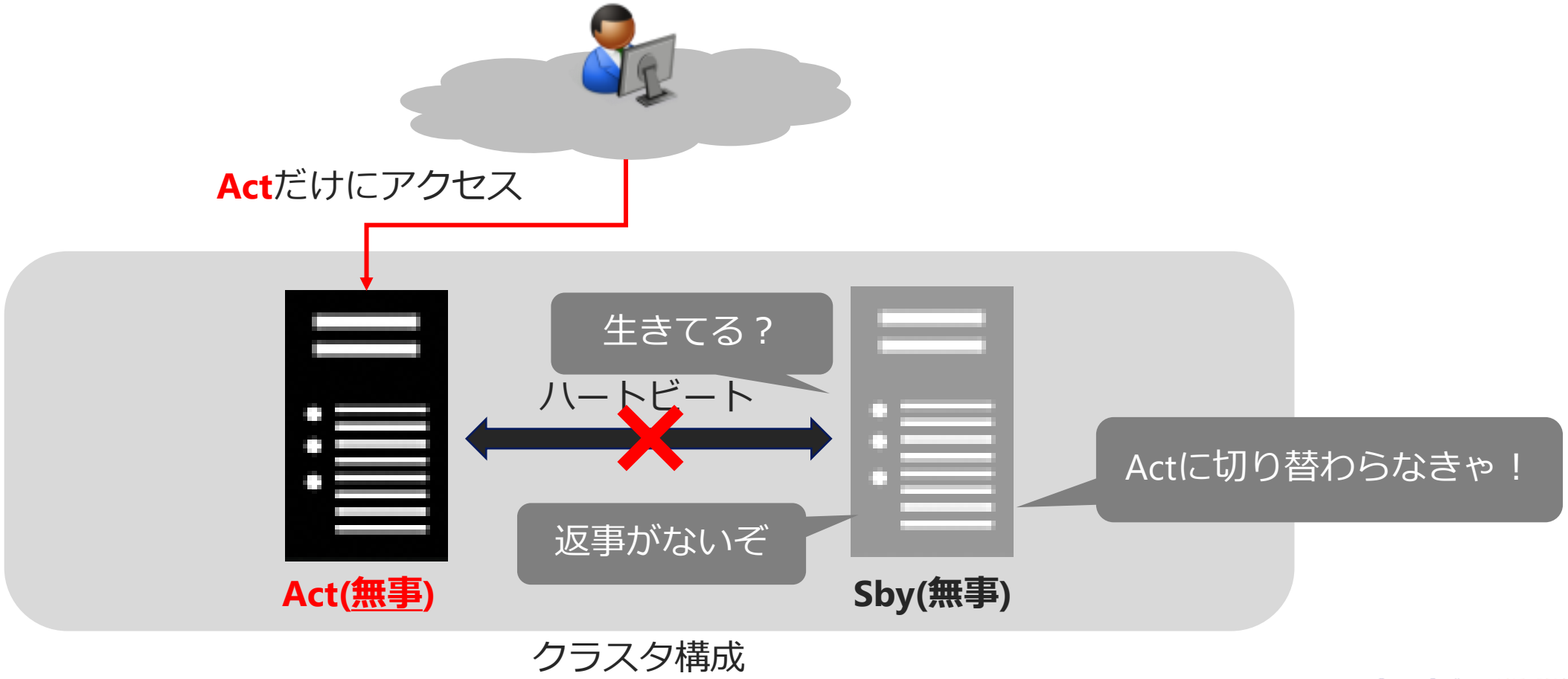
少ないサービス断時間で復旧



クラスタ構成

具体的にどういう状態？

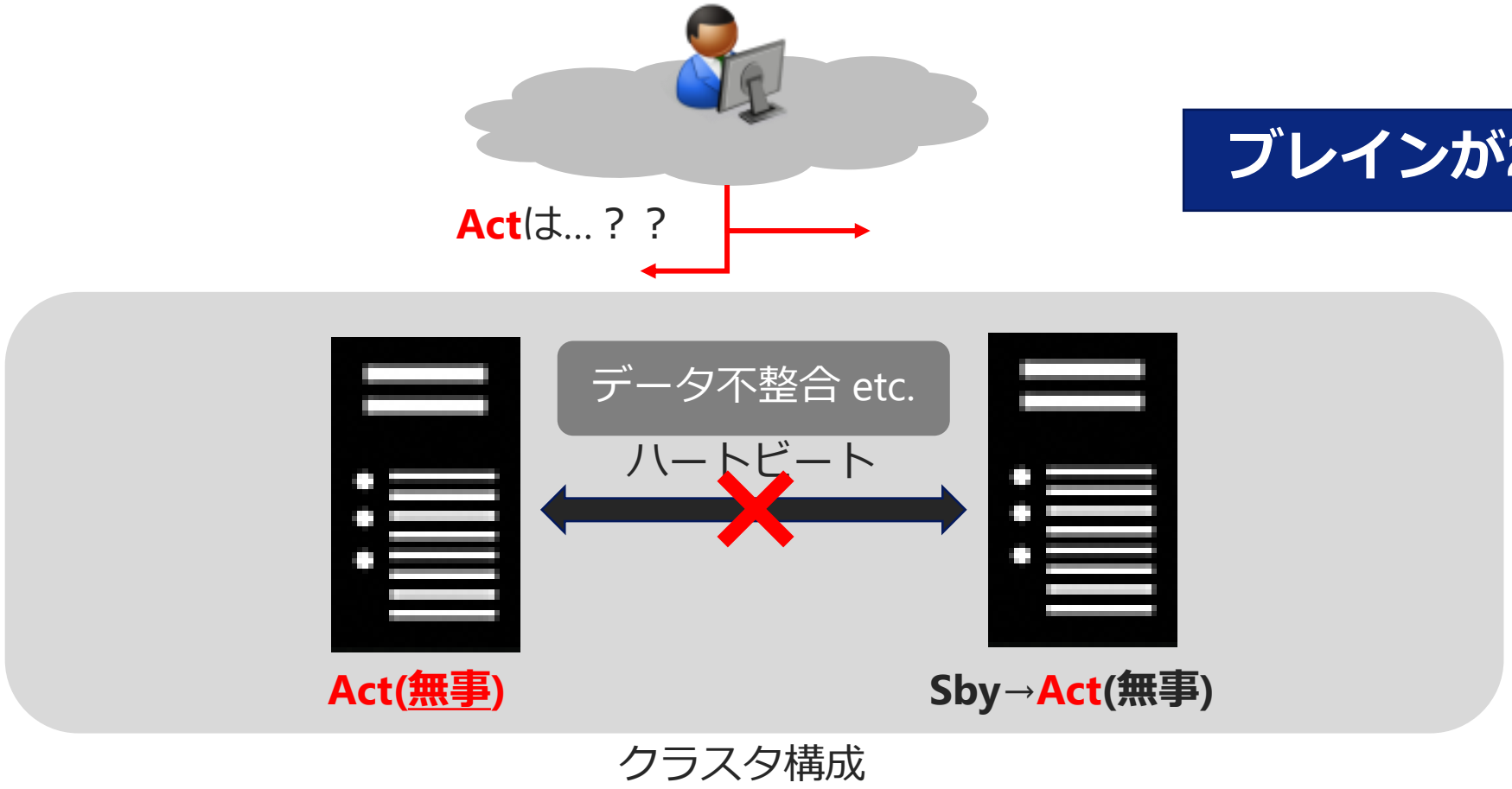
問題点：生死確認**だけ**NGだったら？



クラスタ構成

具体的にどういう状態？

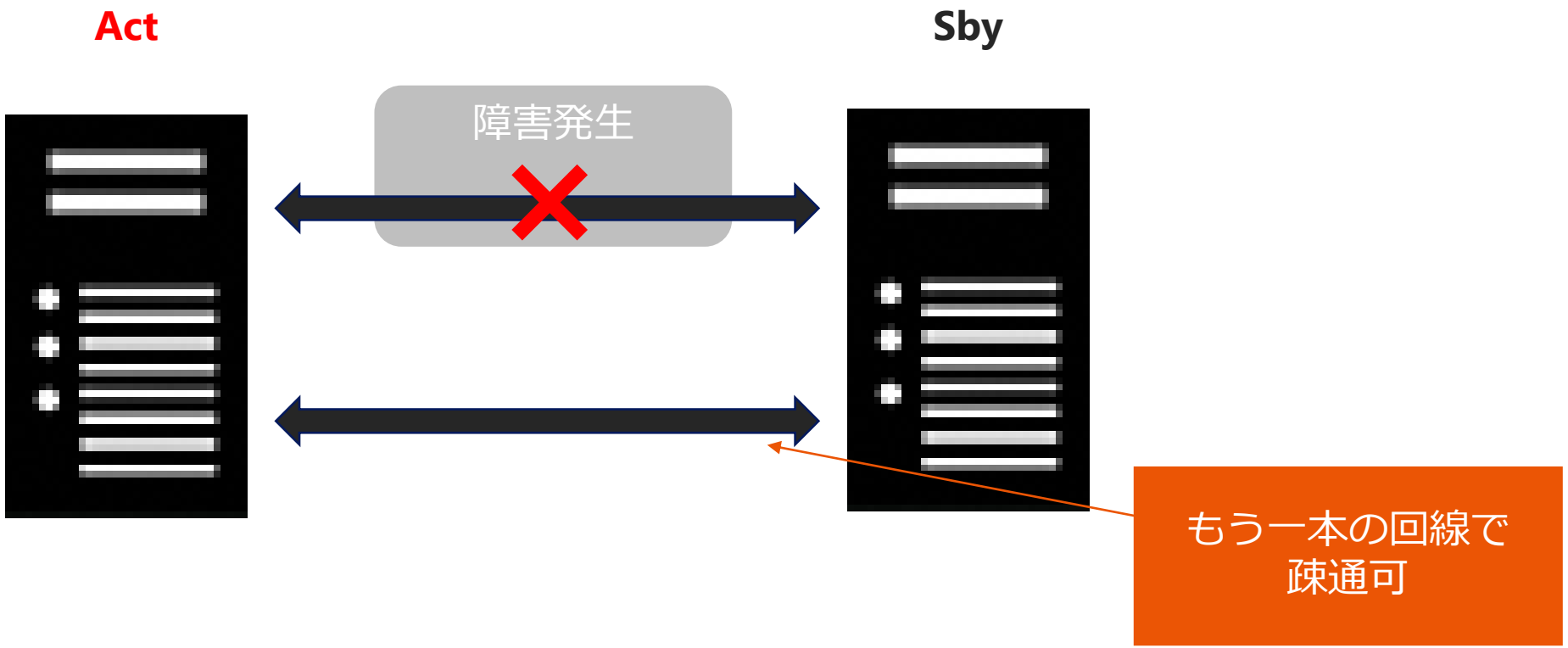
問題点：生死確認の線だけ切れたら？



クラスタ構成

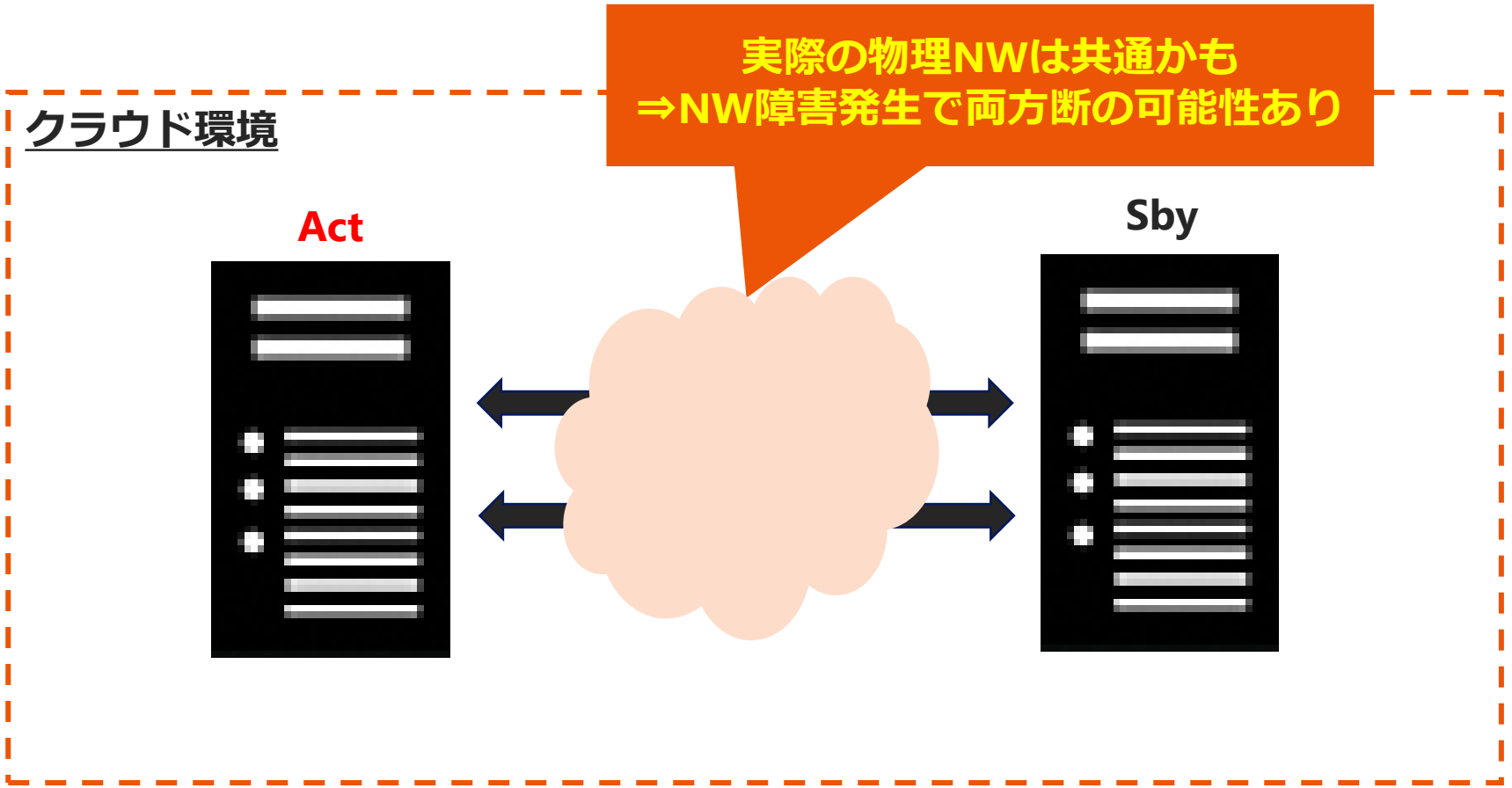
オンプレでのスプリットブレイン対策

ハートビート回線を多重化



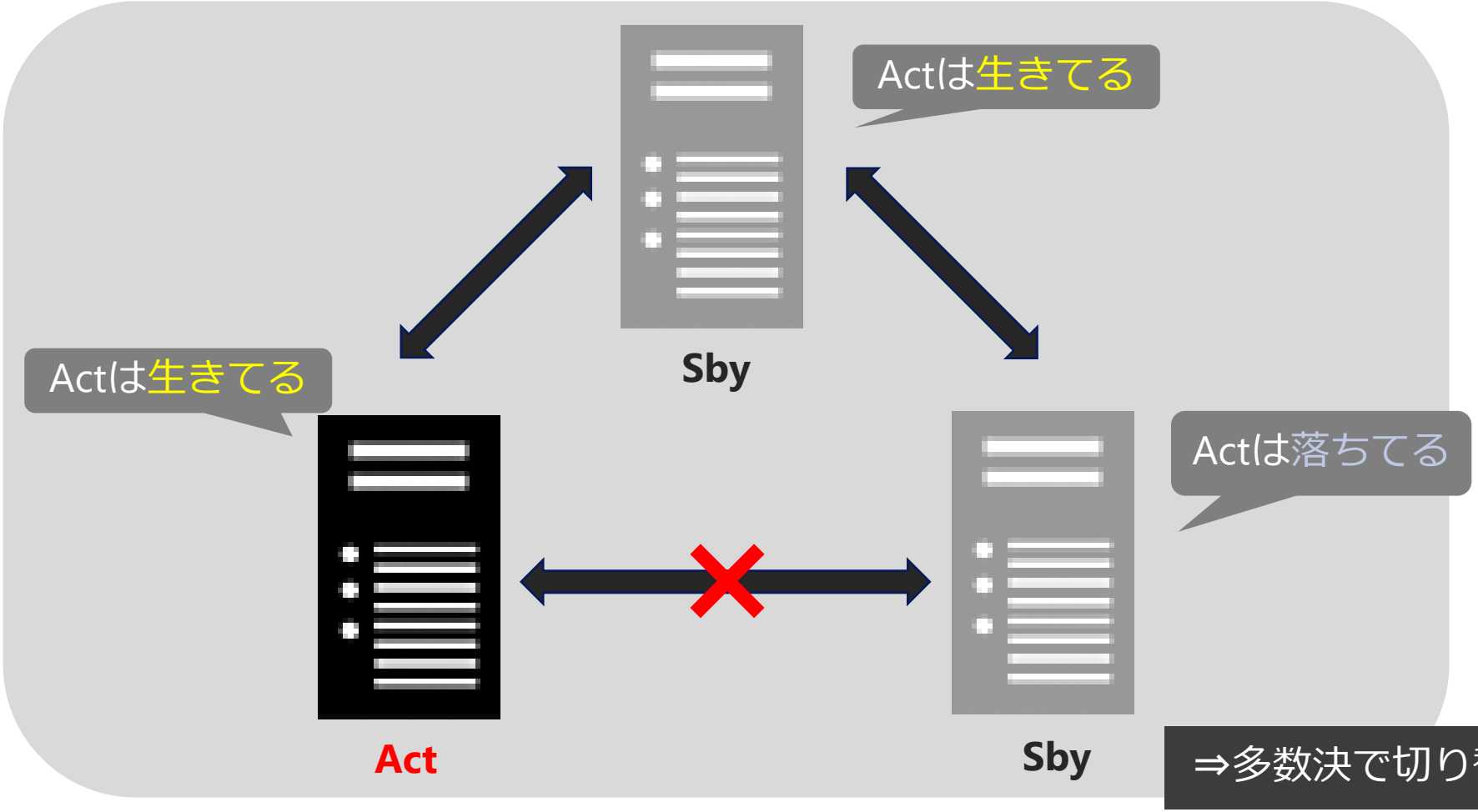
クラウド環境でのスプリットブレイン対策

クラウド環境の裏の**物理NW構成を意識**する必要あり



その他のスプリットブレイン対策： 3台(奇数台)の多数決

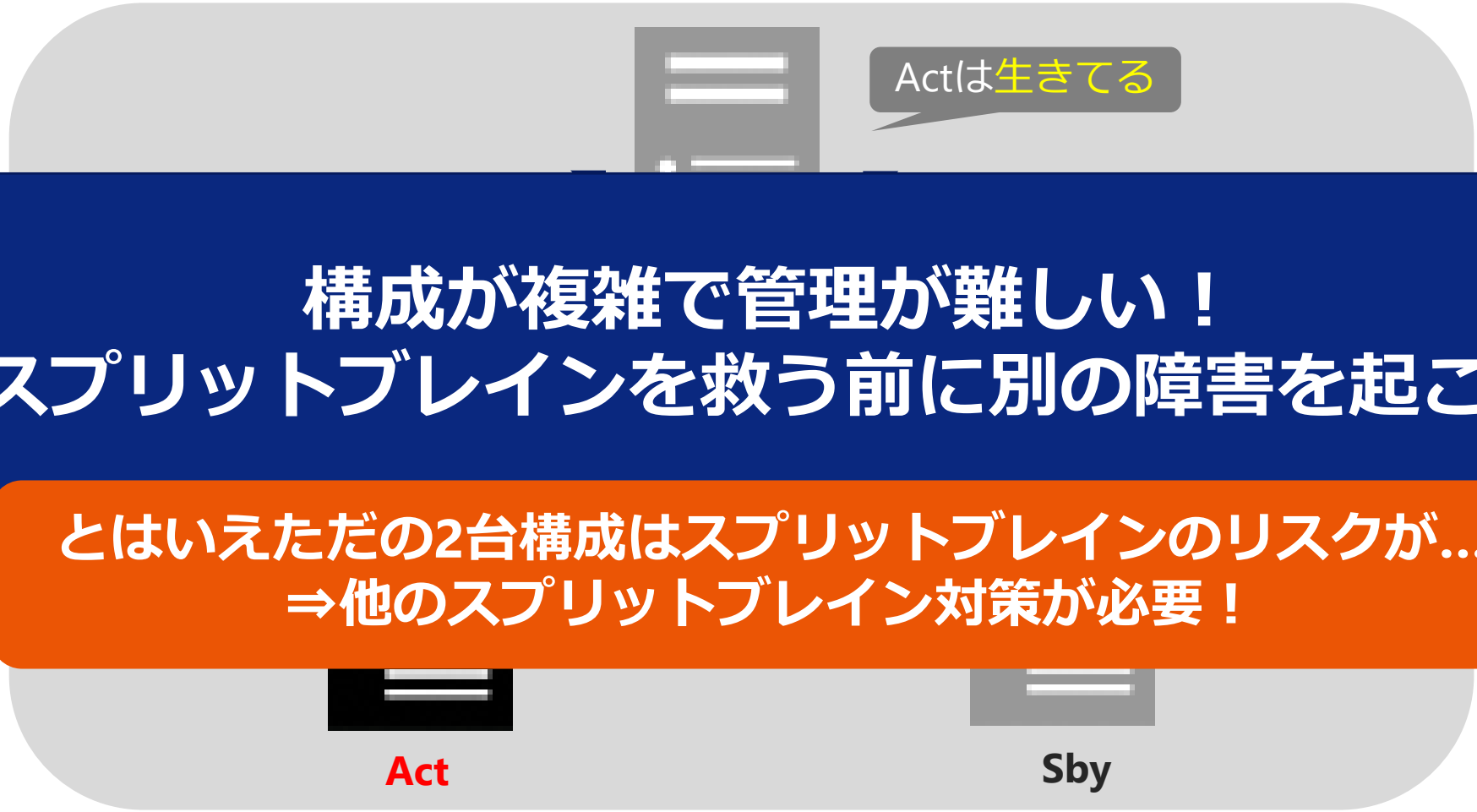
3台の多数決構成



クラスタ構成

その他のスプリットブレイン対策： 3台(奇数台)の多数決

3台の多数決構成



構成が複雑で管理が難しい！
(スプリットブレインを救う前に別の障害を起こす)

とはいえただの2台構成はスプリットブレインのリスクが…
⇒他のスプリットブレイン対策が必要！

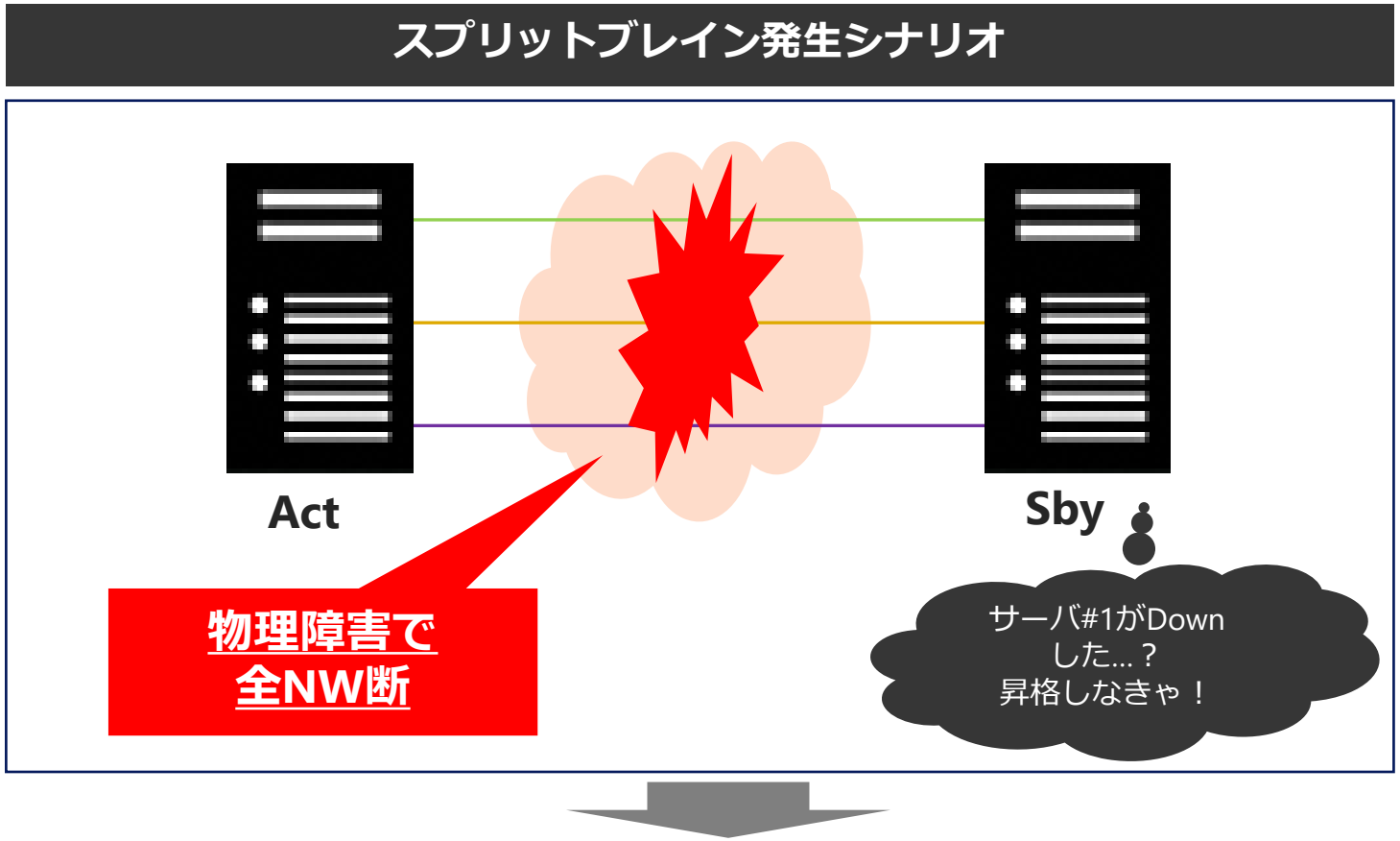
クラスタ構成

アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	**私達の提案する標準クラスタ設計**
7	スプリットブレイン発生時の復旧方法
8	まとめ

クラウドでのスプリットブレイン発生シナリオ

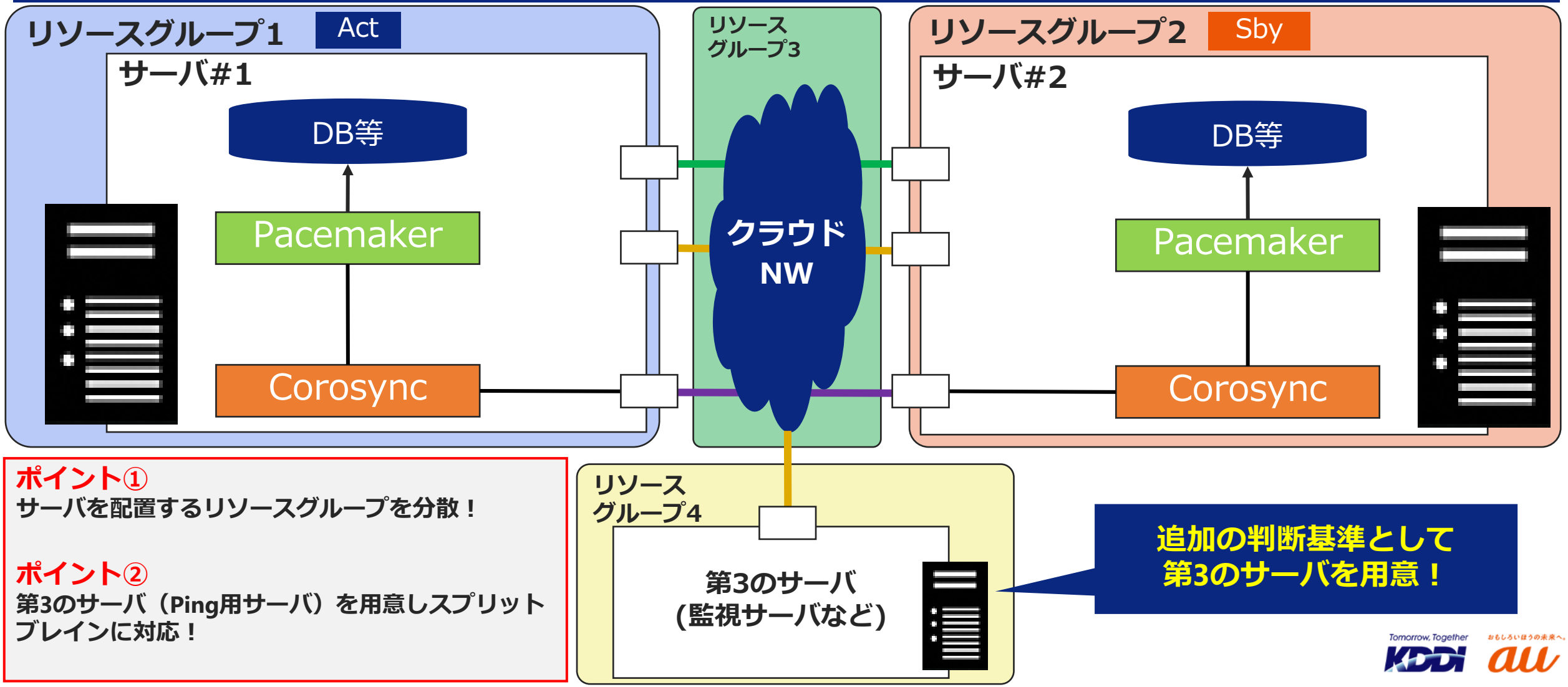
物理障害により全ネットワークの疎通が取れなくなり、SbyサーバがActに昇格する



Act-Sby間ハートビート以外の判断基準が必要！

提案方式

クラウドが提供するリソースグループ（同時には壊れない物理的な範囲）ごとに
独立したスプリットブレイン対策を実施する



ポイント①
サーバを配置するリソースグループを分散！

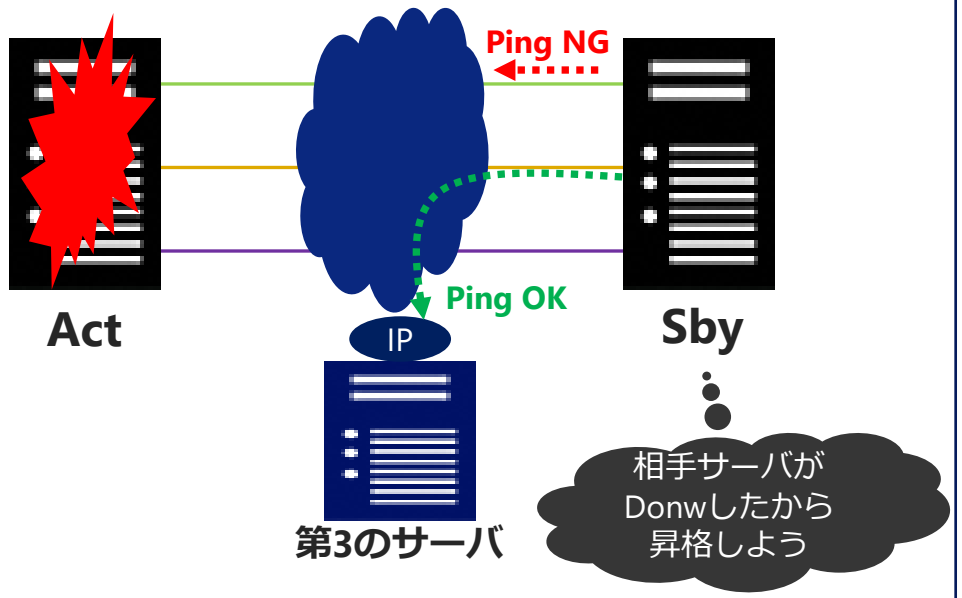
ポイント②
第3のサーバ（Ping用サーバ）を用意しスプリットブレインに対応！

追加の判断基準として
第3のサーバを用意！

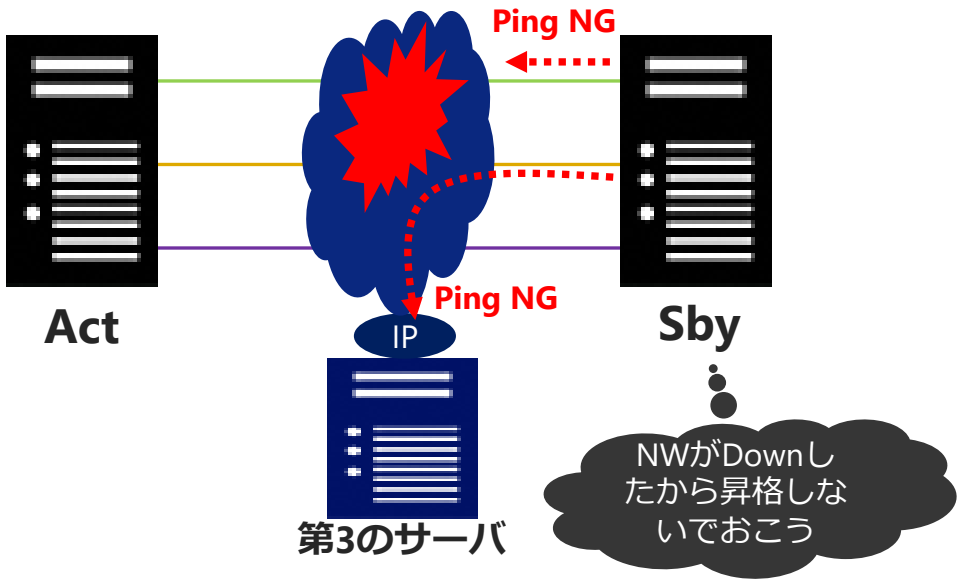
スプリットブレイン対策：第3のサーバをPing監視

相手サーバへのハートビート（Ping監視）に加え、第3のサーバをPing監視する

パターン①：ActサーバがDown



パターン②：ネットワーク全体がDown



Point

- Sbyサーバから別リソースグループ上で稼動している第3のサーバに対してPingを実行する。
- 第3のサーバへのPingNGであればActサーバと疎通出来なくなった原因がネットワークだと判断できるため、SbyサーバはActに昇格させない。

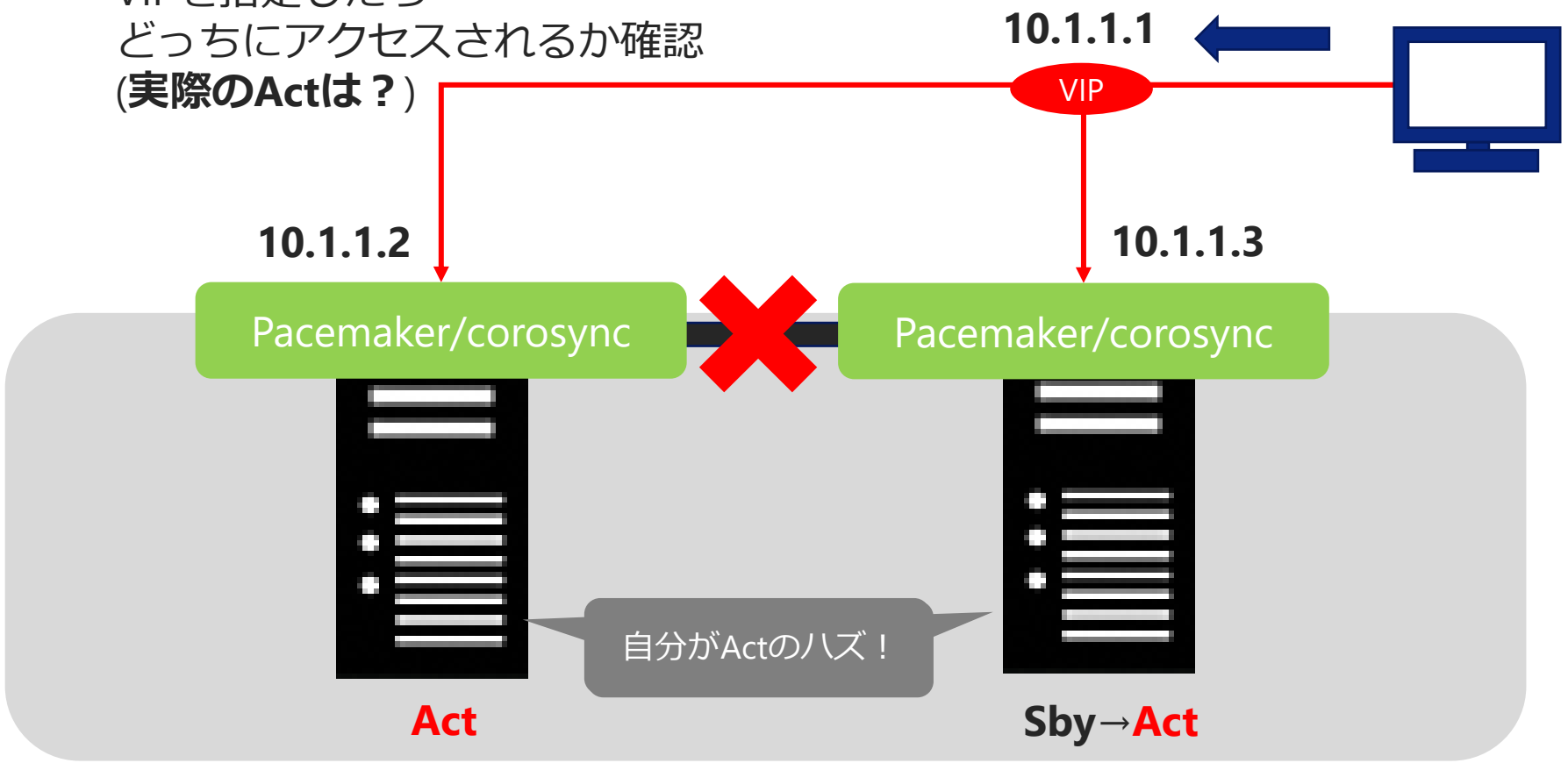
アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

復旧シナリオ①：両Act状態からの復旧（1/2）

スプリットブレイン発生時に両Actを検知(この時点ではサービスは継続)

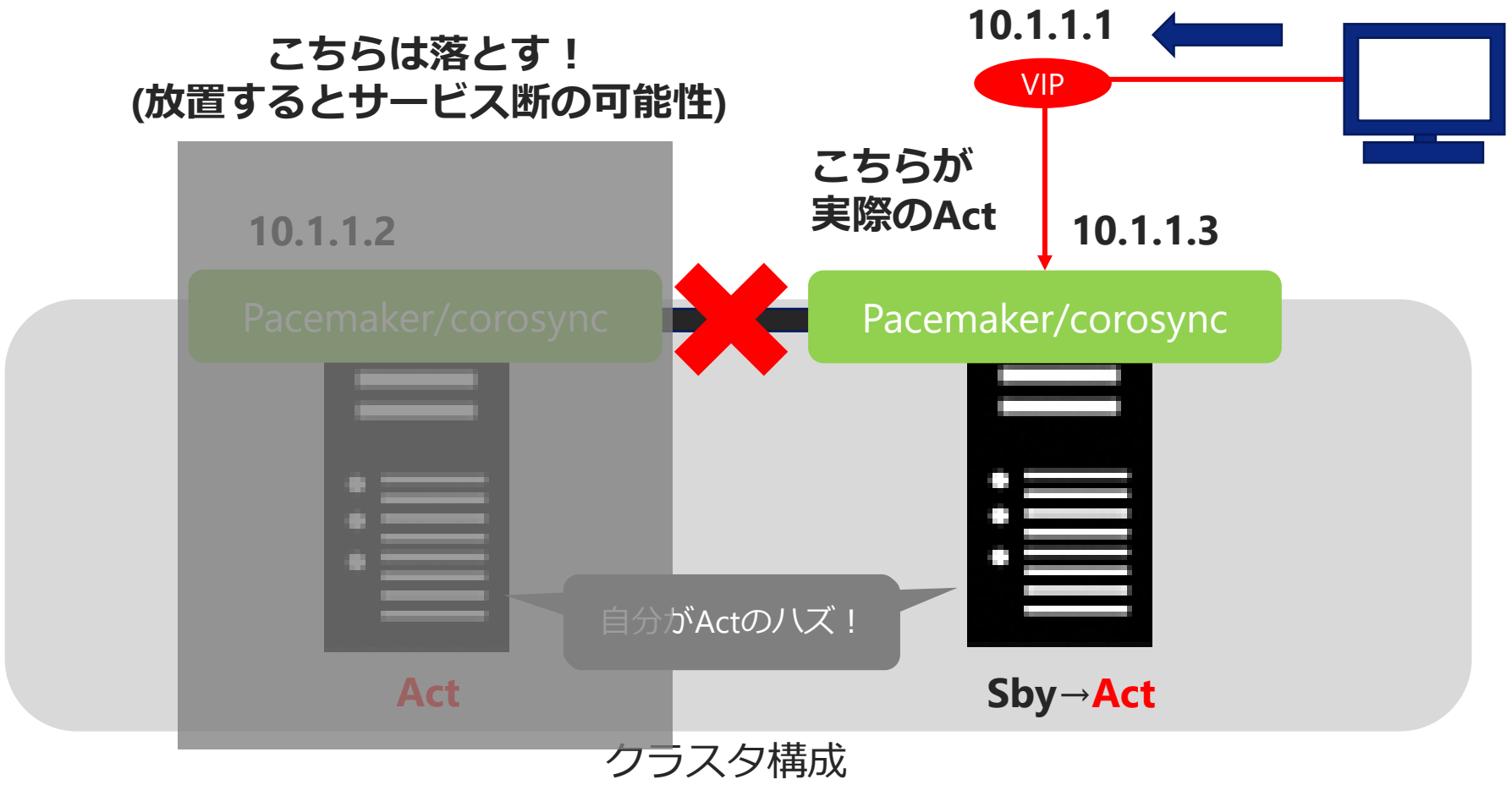
VIPを指定したら
どっちにアクセスされるか確認
(**実際のActは？**)



クラスタ構成

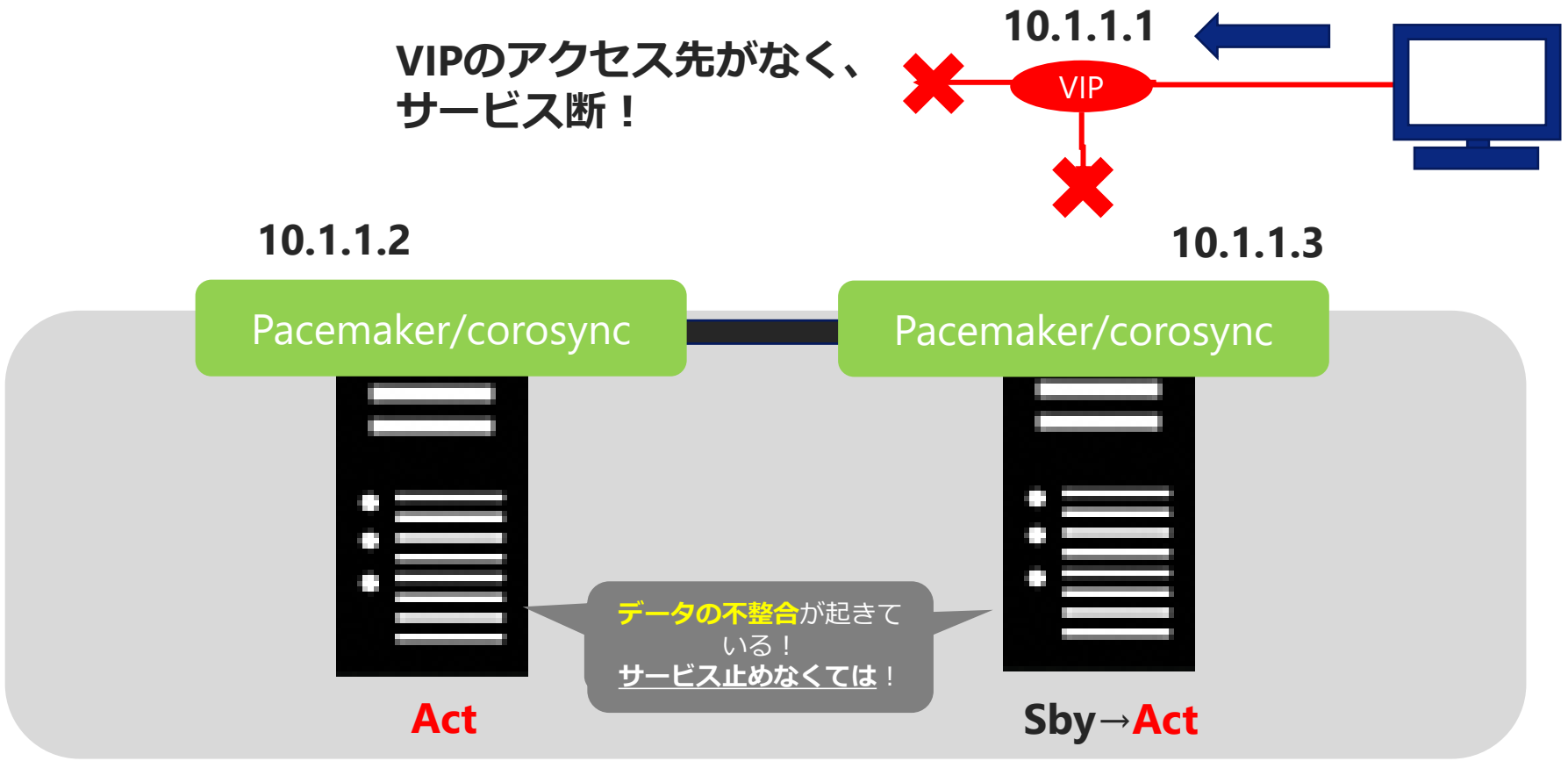
復旧シナリオ①：両Act状態からの復旧（2/2）

復旧方法：実際のActではない方のサーバを落とす！



復旧シナリオ②：両系断状態からの復旧（1/2）

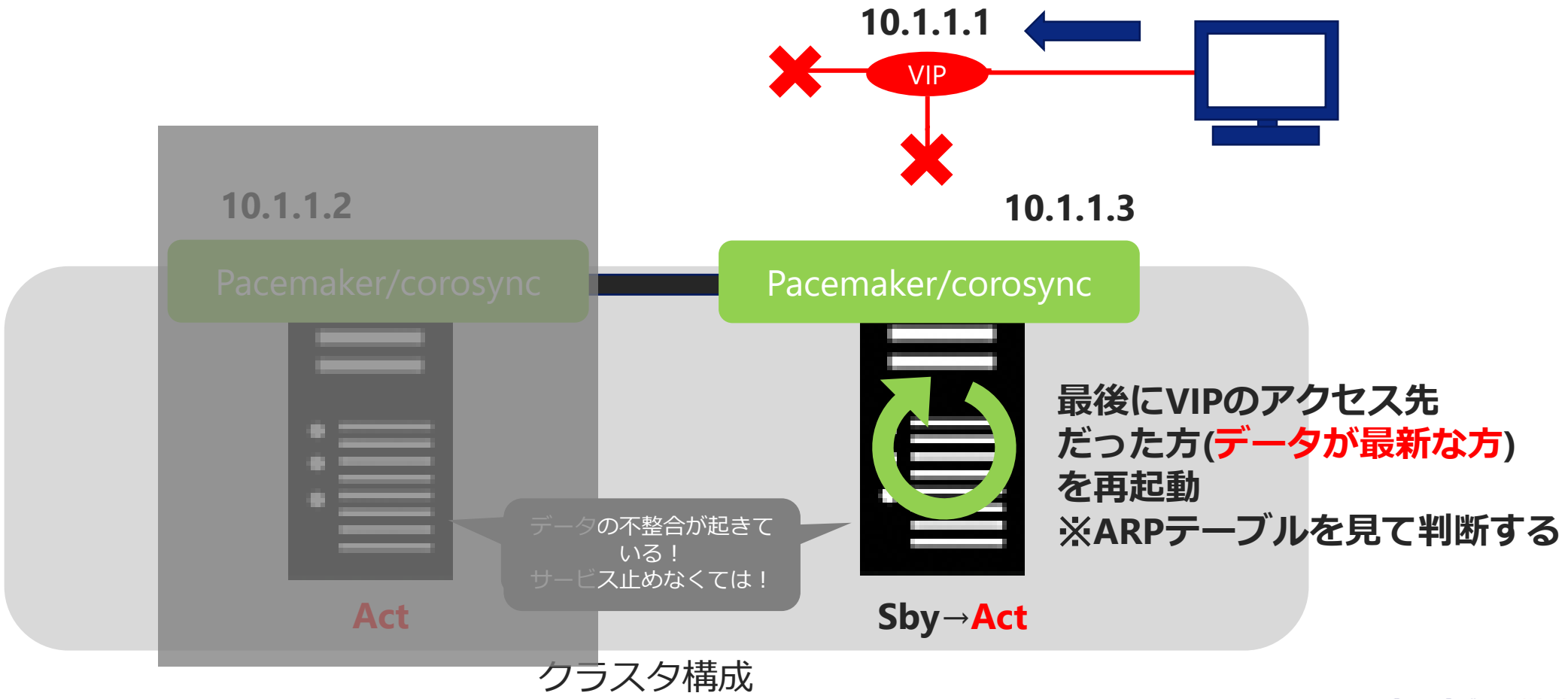
両Act状態からネットワークが復旧した場合、両系断となる



クラスタ構成

復旧シナリオ②：両系断状態からの復旧（2/2）

復旧方法②：最後にVIPのアクセス先だった方(データが最新な方)を再起動



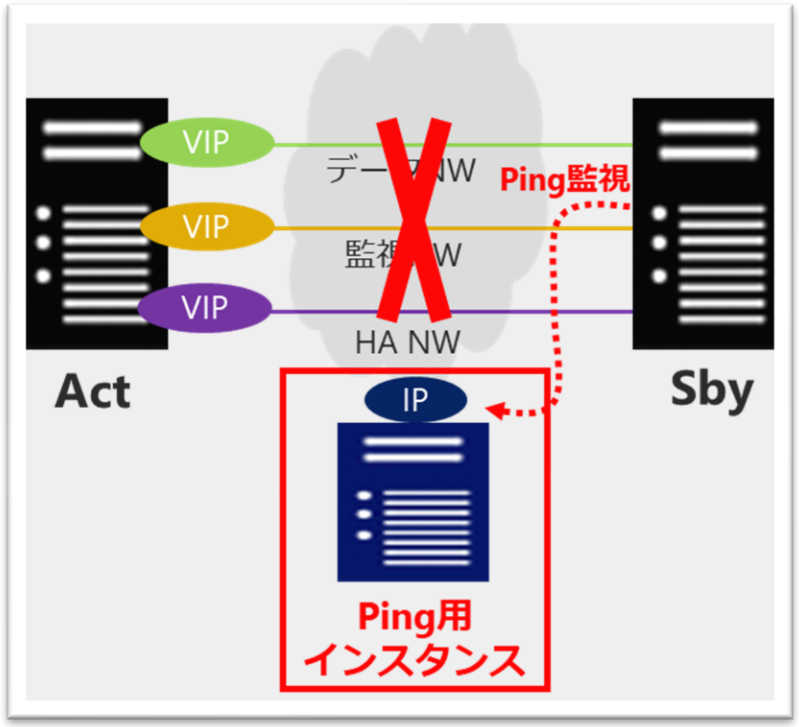
クラスタ構成

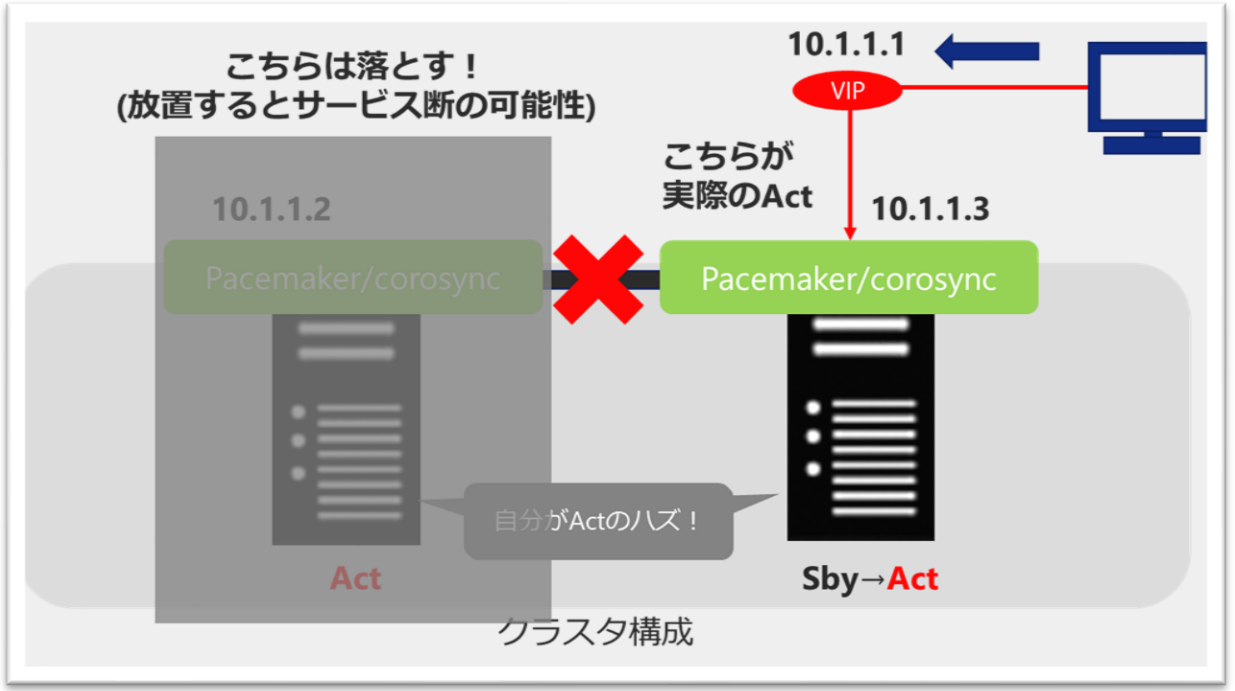
アジェンダ

1	講演者紹介
2	KDDI「手の内化」の紹介
3	設計標準活動の紹介
4	前提知識①：Pacemaker/Corosync
5	前提知識②：スプリットブレイン
6	私達の提案する標準クラスタ設計
7	スプリットブレイン発生時の復旧方法
8	まとめ

まとめ

- クラウド環境で障害が起こりにくいAct/Sbyクラスタ設計を提案
- 障害が起きても早く復旧できる復旧方法を提案





第3のサーバを用意することで
スプリットブレイン予防を提案！

早くスプリットブレインから
復旧する方法を提案！

参考

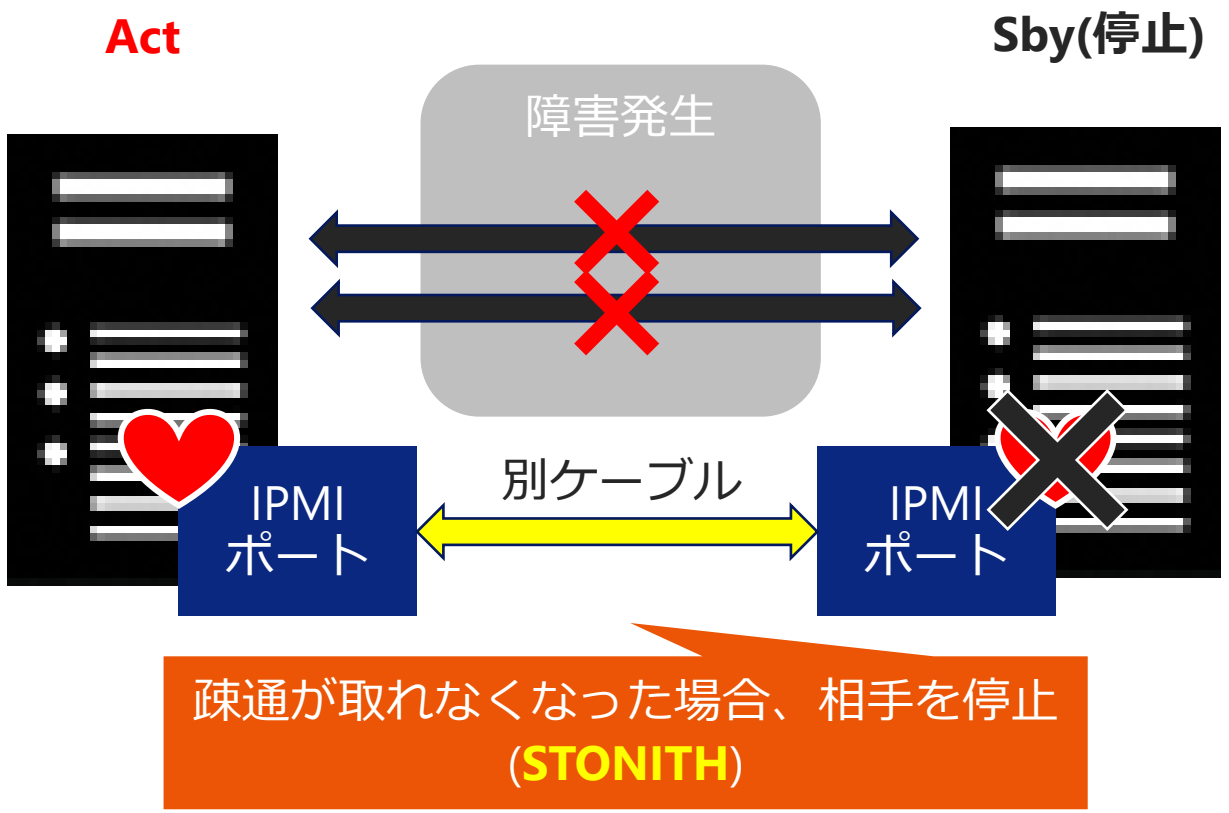
ノード数の違い

「奇数(3)ノードによる多数決」と「2ノードACT/SBY」のメリット/デメリット比較

項目	3ノード	2ノード	説明
スプリットブレイン発生確率低減	○	△	ActサーバがNW障害などで意図せずにSbyサーバがActに昇格する確率を低減する。
障害復旧の容易さ	△	○	クラスタを構成するサーバの数が増えるのでトラブルシューティングは必然的に複雑になる。
システム構成のわかりやすさ	×	○	クラスタを構成するサーバの数が増えるのでシステム構成は必然的に複雑になる。

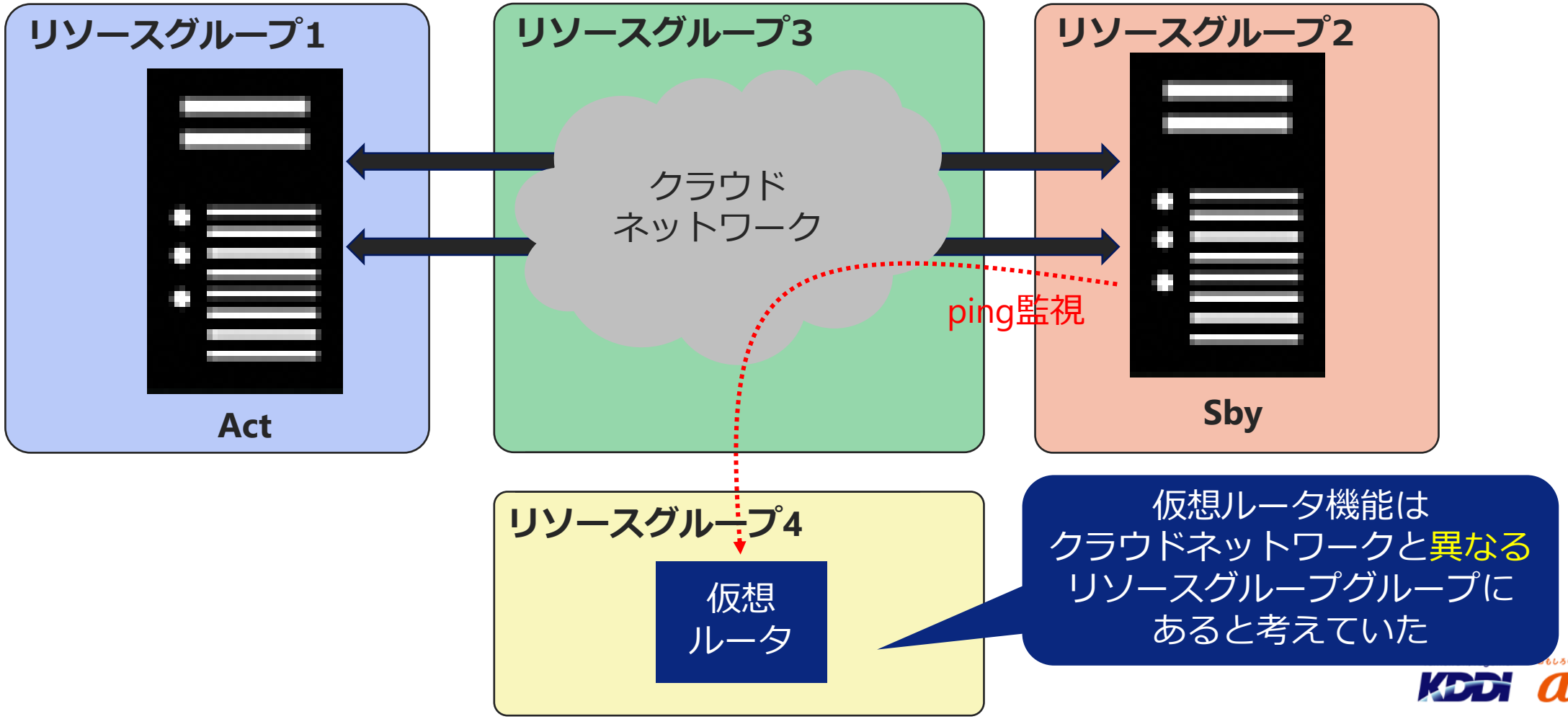
その他のスプリットブレイン対策：STONITH

専用ハード (IPMIポート) による死活監視 (STONITH) で
強制的にスプリットブレイン解消



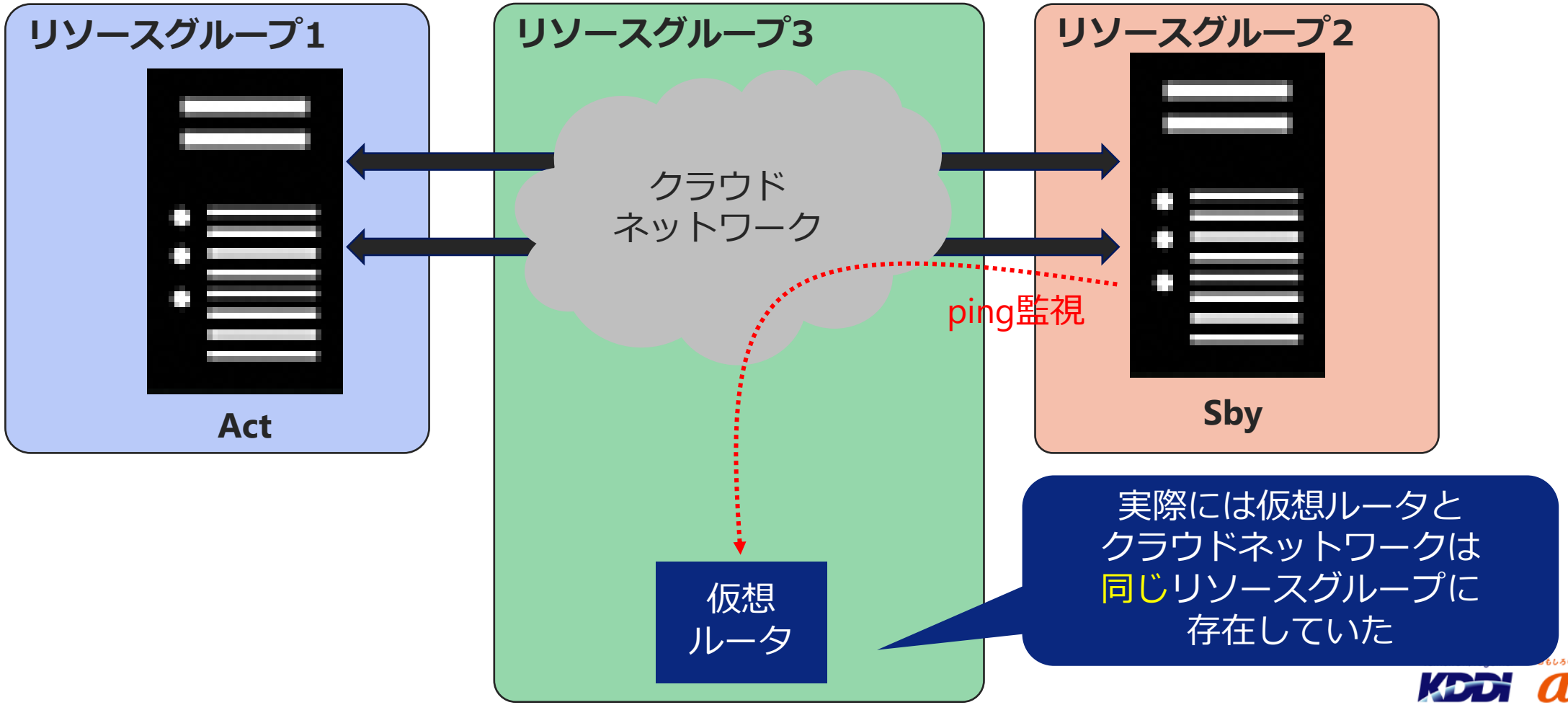
第3の「サーバ」じゃないとダメか？(1/2)

当初はスプリットブレイン対策として
仮想ルータへPing監視を設定していたが・・・



第3の「サーバ」じゃないとダメか？(2/2)

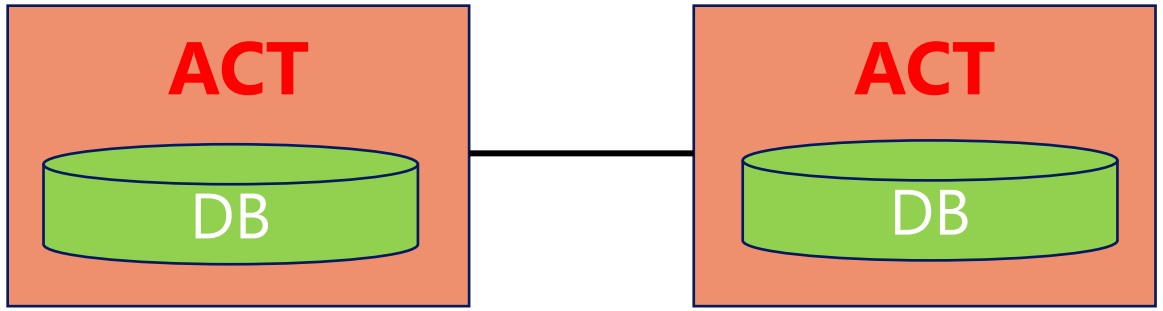
実際にはクラウドネットワークと仮想ルータ機能は
同じリソースグループに存在していた！
→サーバを別のリソースグループに立てる必要あり

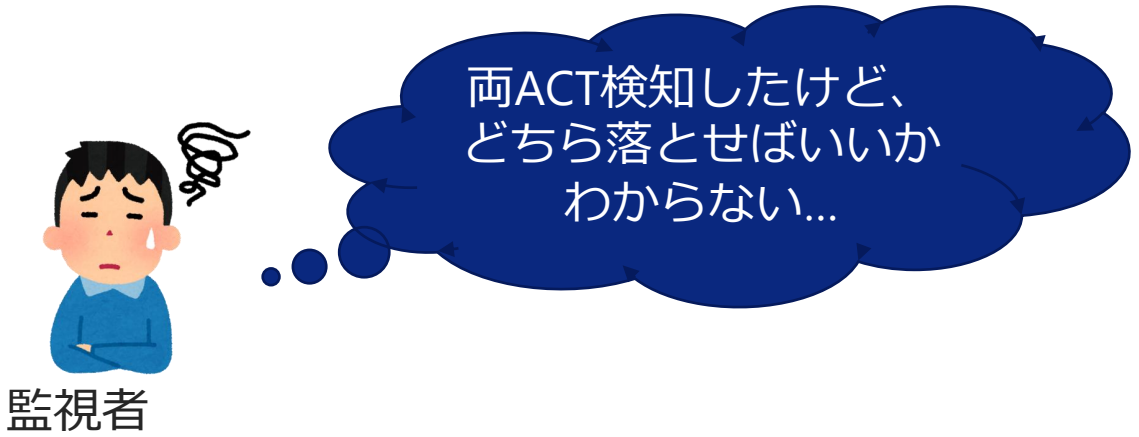


スプリットブレイン復旧の難しさ

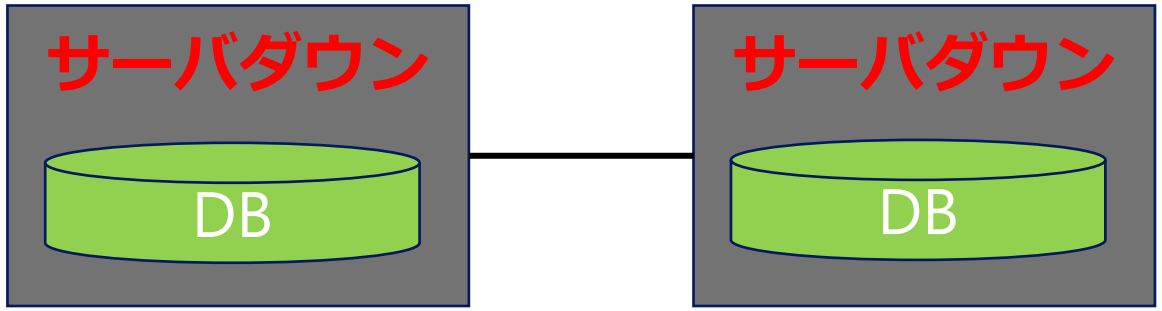
スプリットブレインや両系断が発生すると、監視者では対応が難しい

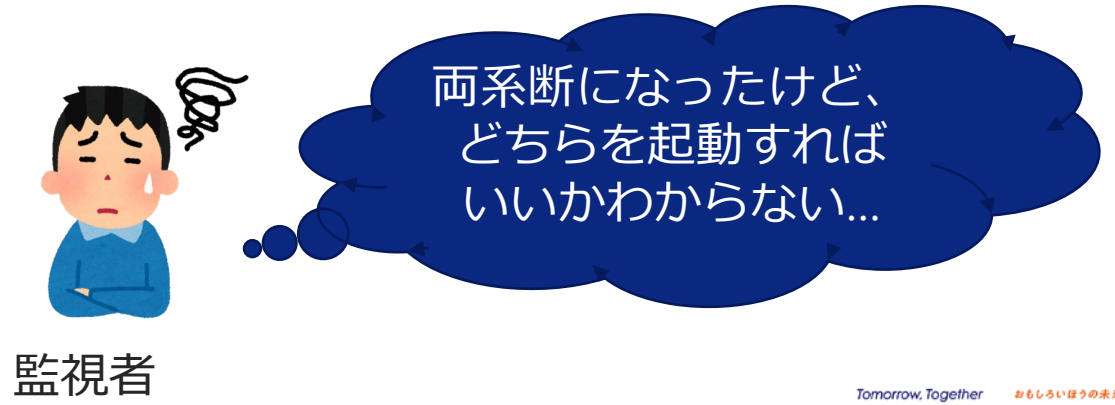
障害ケース　スプリットブレイン





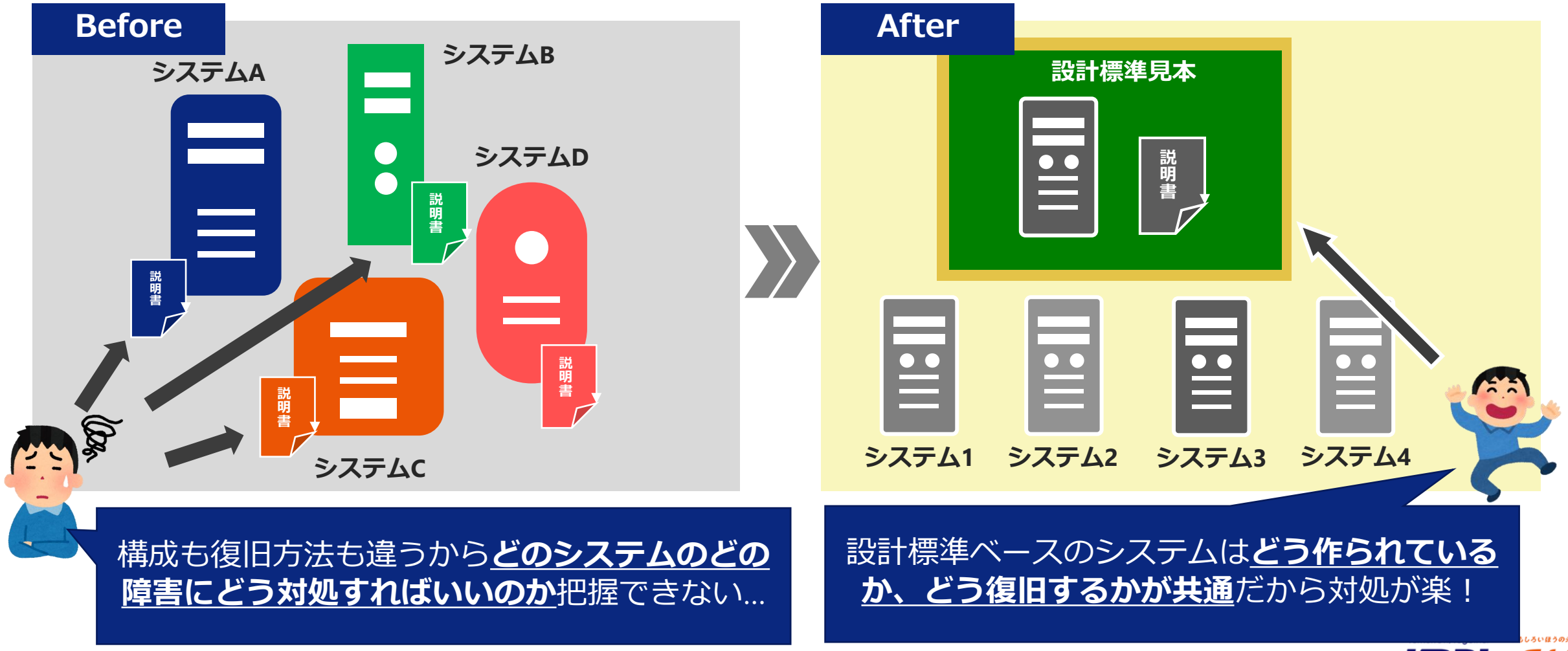
障害ケース　両系断





標準化のメリット

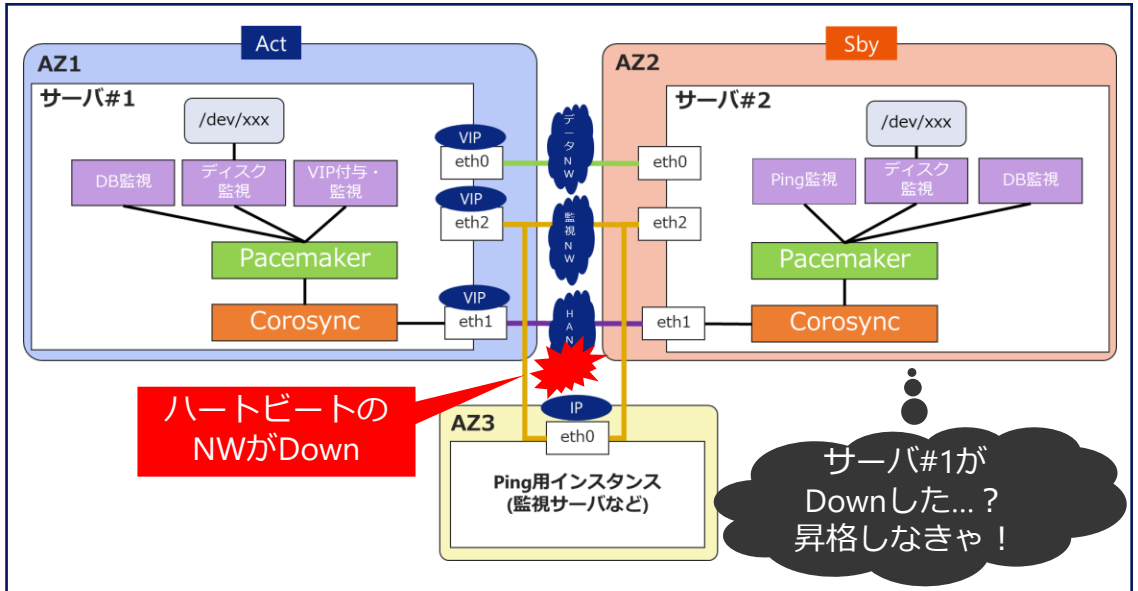
設計標準化することで数百以上の設備もすぐに障害復旧できるようなる！



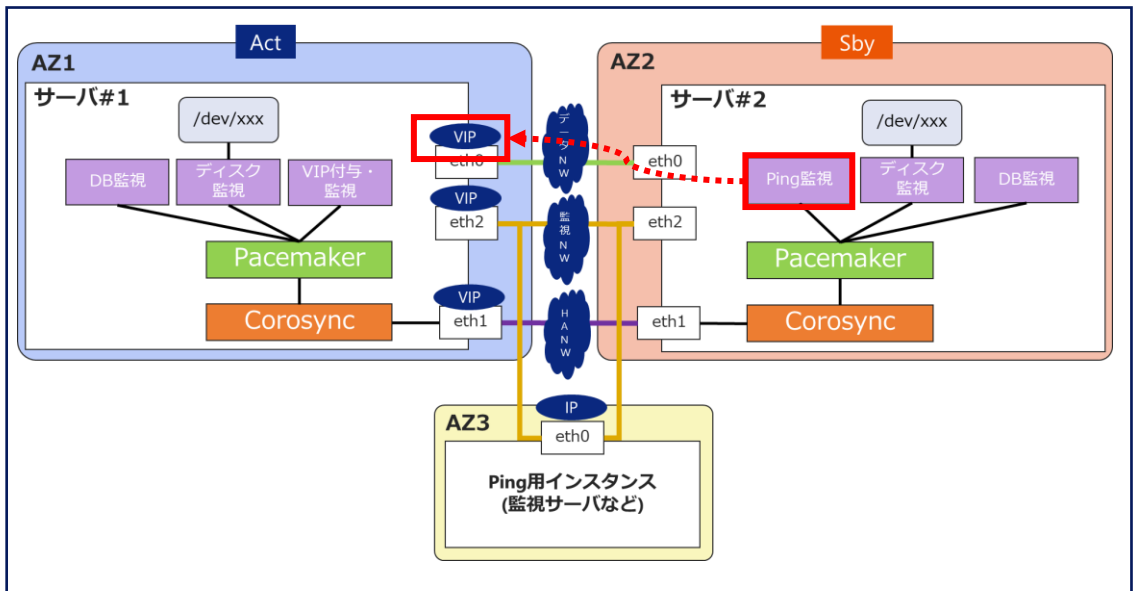
提案方式の具体的な実装①　VIP Ping監視

スプリットブレインの発生シナリオ①
Actサーバが正常に稼動していても、**ハートビートのネットワークが疎通断**になるとSbyサーバがActに昇格する。

スプリットブレイン発生シナリオ①



対策①: VIP Ping監視



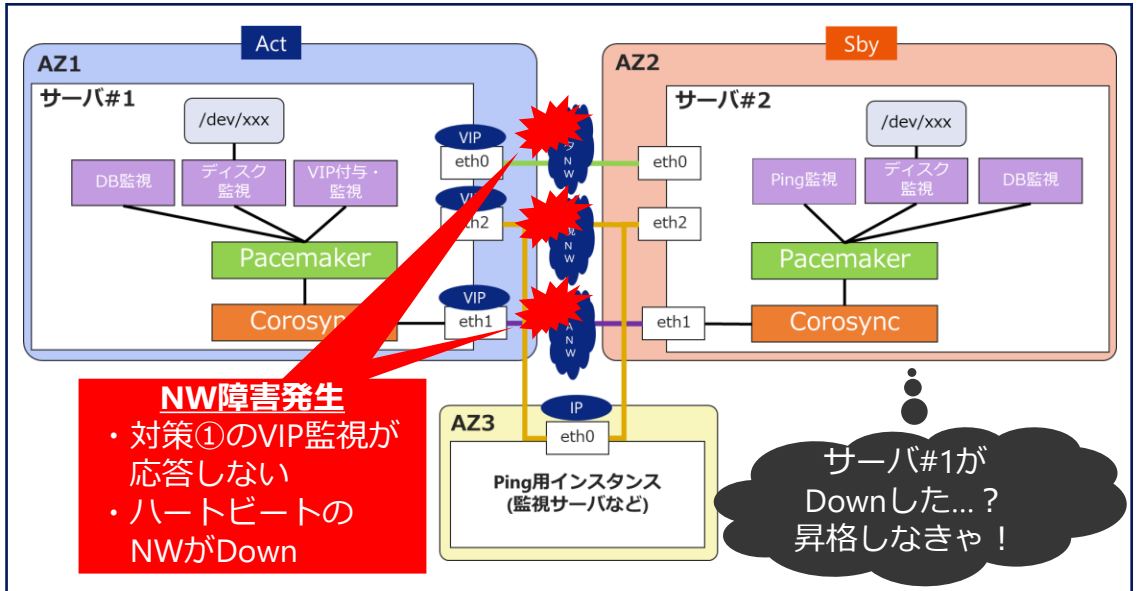
Point

- SbyサーバからActサーバのデータセグメントの**VIP**に対して**Ping**を実行する。
- PingOKであればActサーバのサービスは継続できる状態だと判断できるため、**Sbyサーバは**Actに昇格させない。

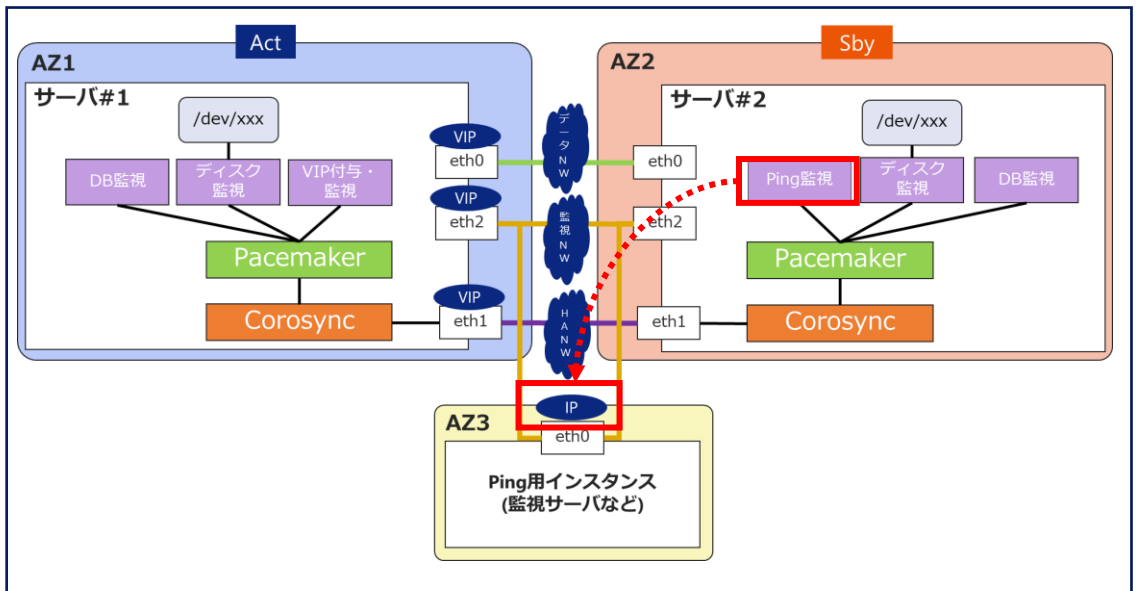
提案方式の具体的な実装②　第3のサーバのPing監視

スプリットブレインの発生シナリオ②
ネットワークの問題等で**前頁対策①(VIP Ping監視)が機能せず**、SbyサーバがActに昇格する。

スプリットブレイン発生シナリオ②



対策②: Ping用インスタンス監視



Point

- Sbyサーバから別リソースグループ上で稼動している**第3のサーバに対してPingを実行**する。
- 第3のサーバへのPingNGであれば**Actサーバと疎通出来なくなった原因がネットワークだと判断**できるため、SbyサーバはActに昇格させない。