

# LVSって不便だよね

- Amazon S3のような分散システムだとLBが容易にボトルネックになり得る
- さくらVPSのような環境で使えない
- ディザスタリカバリやろうと思ったらどうすればいいのか

何とかならないっすかね？

# そこで PrimWatch

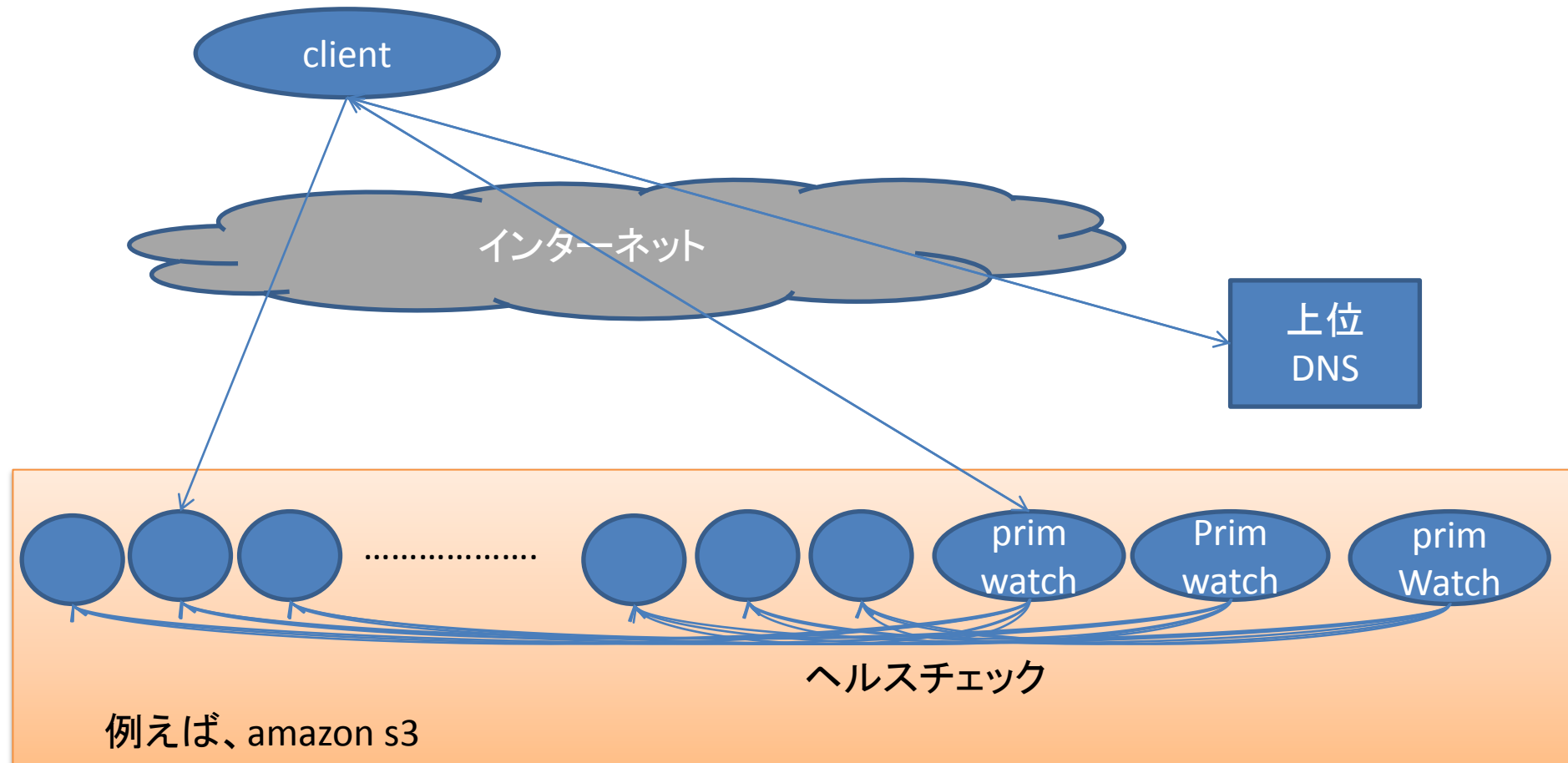
- ホストのヘルスチェックの結果とDNSを連携させることでLVS相当のことを可能にする

# ところで

## 外部とインターフェイスを持つDNS達

- powerDNS
  - <http://www.powerdns.com/content/home-powerdns.html>
  - DNSSECもサポートしててかなり高機能
  - Pipeline backend (pipe)
  - リクエストのソースアドレス判定も可能
- primDNS
  - <https://github.com/ebisawa/primdns>
  - シンプルで高速
  - External engine (pipe)
- gdns
  - <https://github.com/blblack/gdnsd>
  - シンプル
  - Cのプラグインなので書きづらい

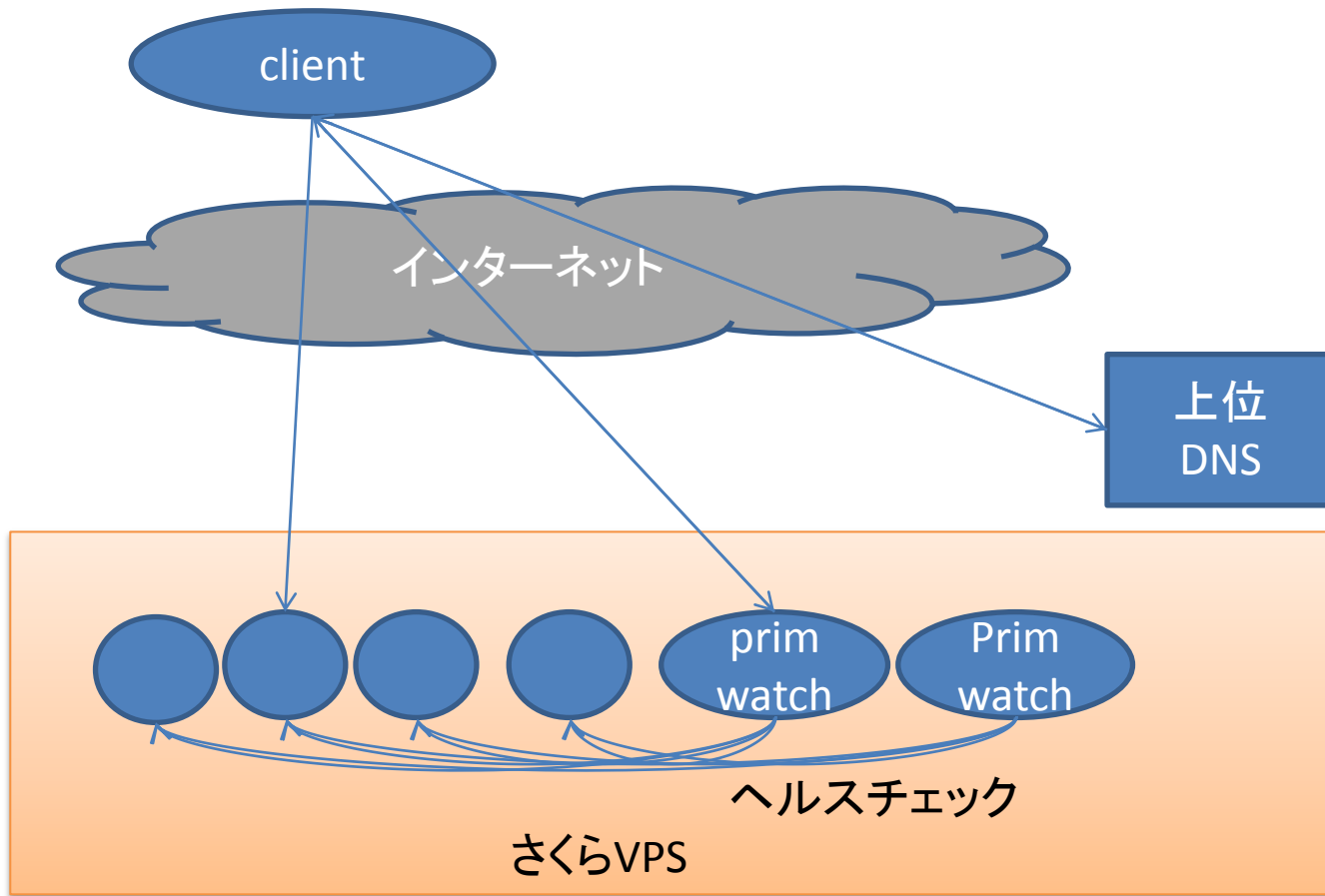
# Use case 分散システム



- LVSによる帯域、集中的なボトルネックがない
- ホスト管理の簡略化

# Use case

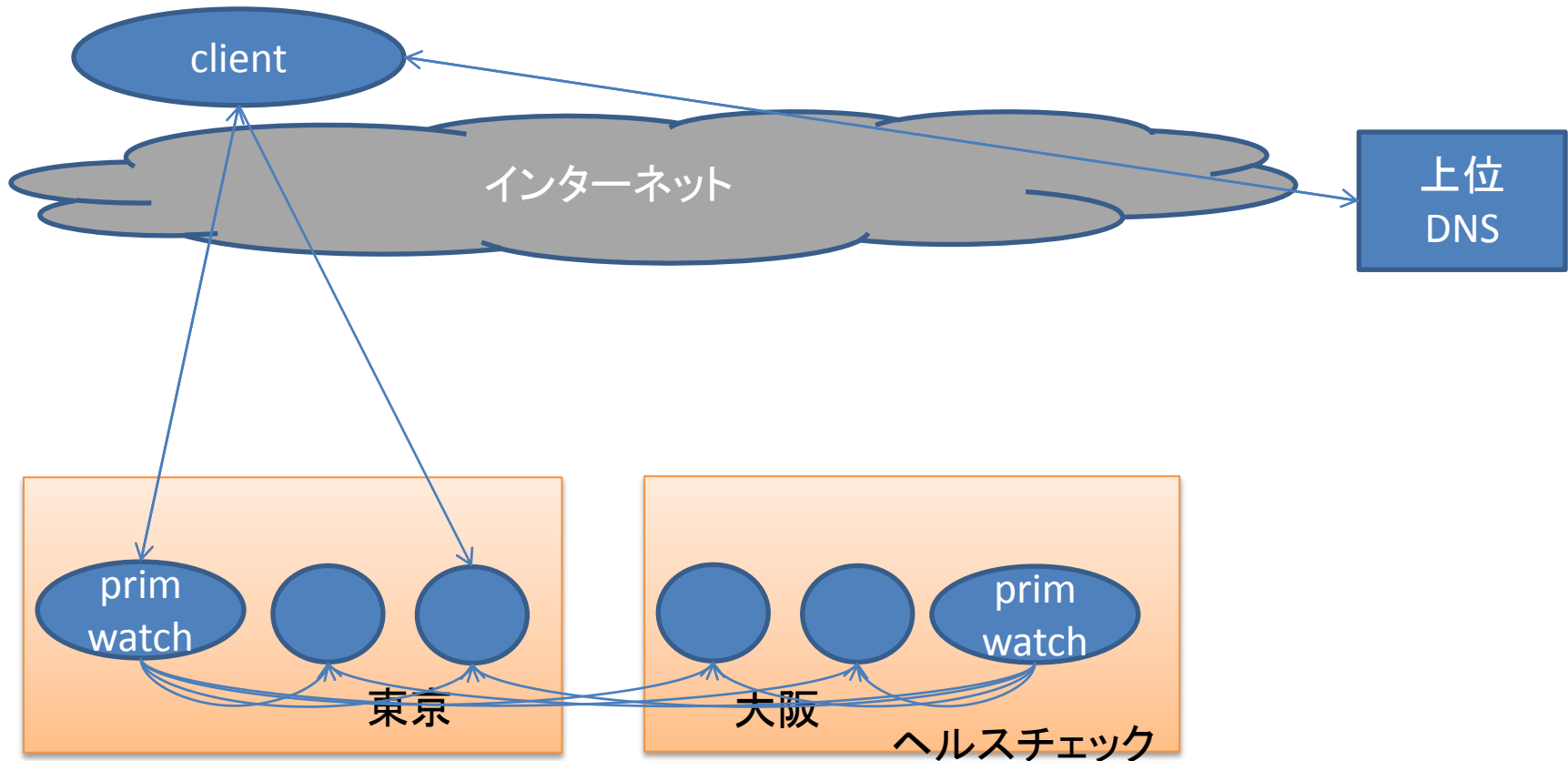
## LVSが使えない環境



- LVsが使えない環境でも冗長化を実現できる

# Use cas

## ディザスタリカバリ対応

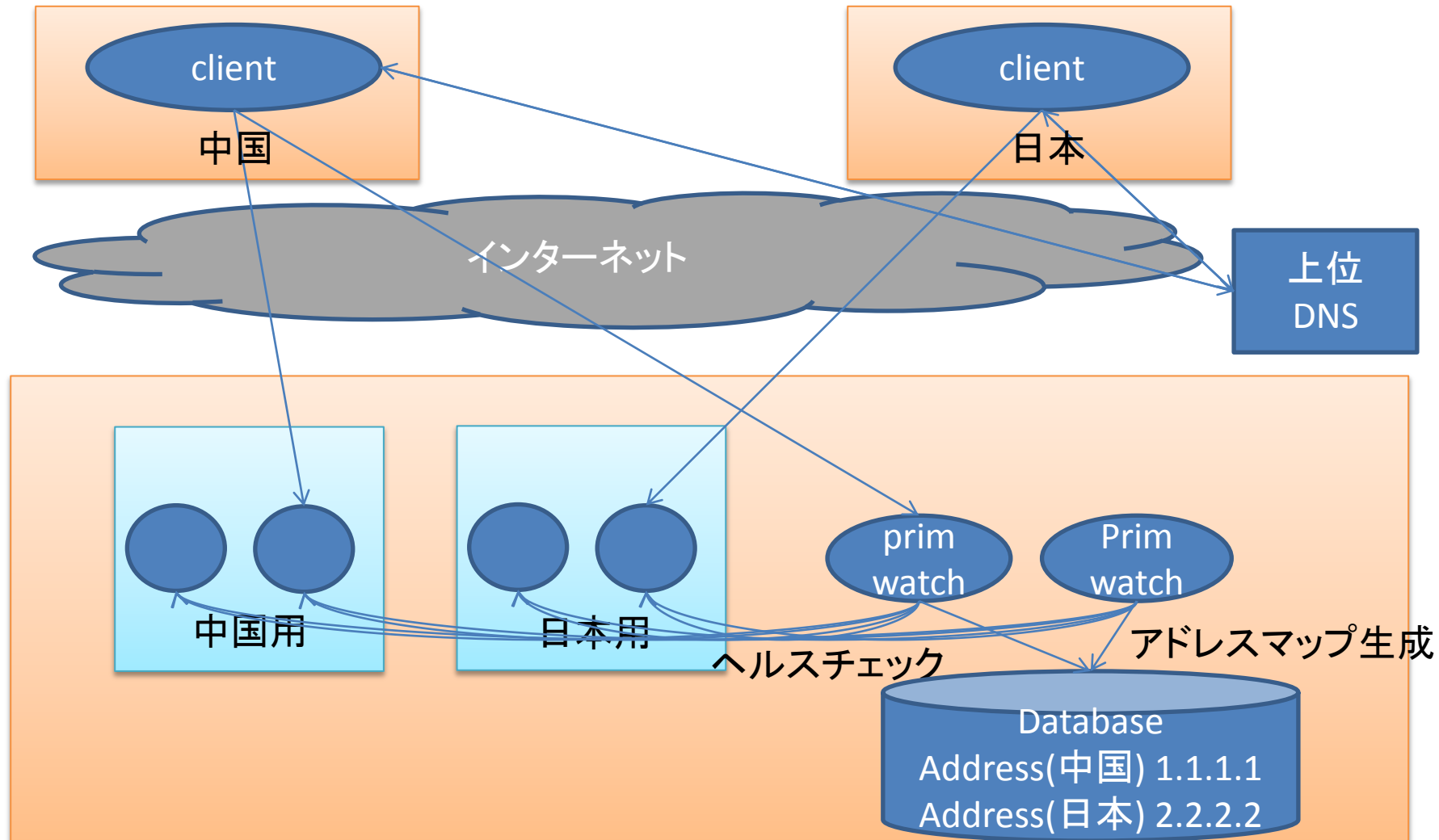


- Priorityにより基本東京、東京が大地震で死んだら大阪



# Use case

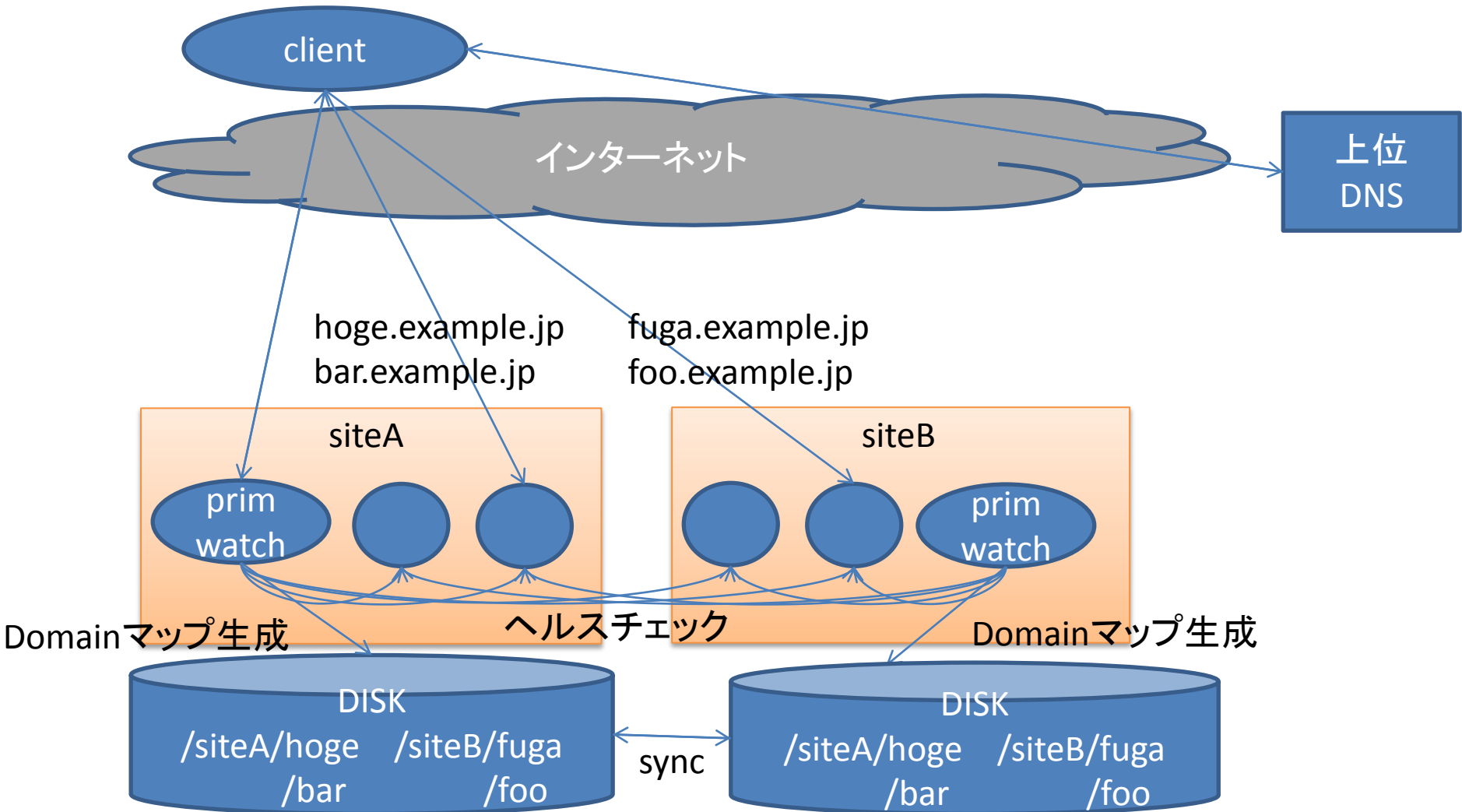
## 国によってわける



- 中国からのリクエストは中国用へ、日本からのリクエストは日本用へ

# Use case

## ドメインによって切り替える



- ドメインにより切り替えを行うことが可能

# というわけで 要件

- ヘルスチェックによりダウンしたホストのアドレスはDNSレコードとして返さない
- リクエストされたドメインによって、返すDNSレコードを切り替えたい
- リクエストのソースアドレスによって、返すDNSレコードを切り替えたい
- 返すレコードをラウンドロビンすることでロードバランスしたい
- 返すレコードをプライオリティで制御したい
- 手動でプライオリティの変更がしたい
  - (運用要件: down/up後、手動で元に戻したい)

# PrimWatchの コンポーネント

- PrimWatch accessa
  - DNSからプロセスforkされるプログラム
  - primDNSモードとpowerDNSモードがある
  - primWatch Serverとmmapでメモリ共有
  - C言語実装
- PrimWatch server
  - コントロールモードとサーバーモードがある
    - コントロールモードでプライオリティの変更、状態参照、設定再読み込みが可能
  - メモリ共有用と監視用スレッドを持つ
  - PrimWatch accessaと現在の情報を共有する
  - 以下の3つのスクリプトから情報を収集する
    - DomainMap script
      - リクエストされたドメインによって返すレコードを切り替えるための情報を返す
    - AddressMap script
      - リクエスト元のアドレスを見て返すレコードを切り返えるための情報を返す
    - healthCheck script
      - ヘルスチェックによりレコードを有効/無効にするための情報を返す
  - HealthCheck scriptのみリファレンス実装として用意する
  - C言語実装

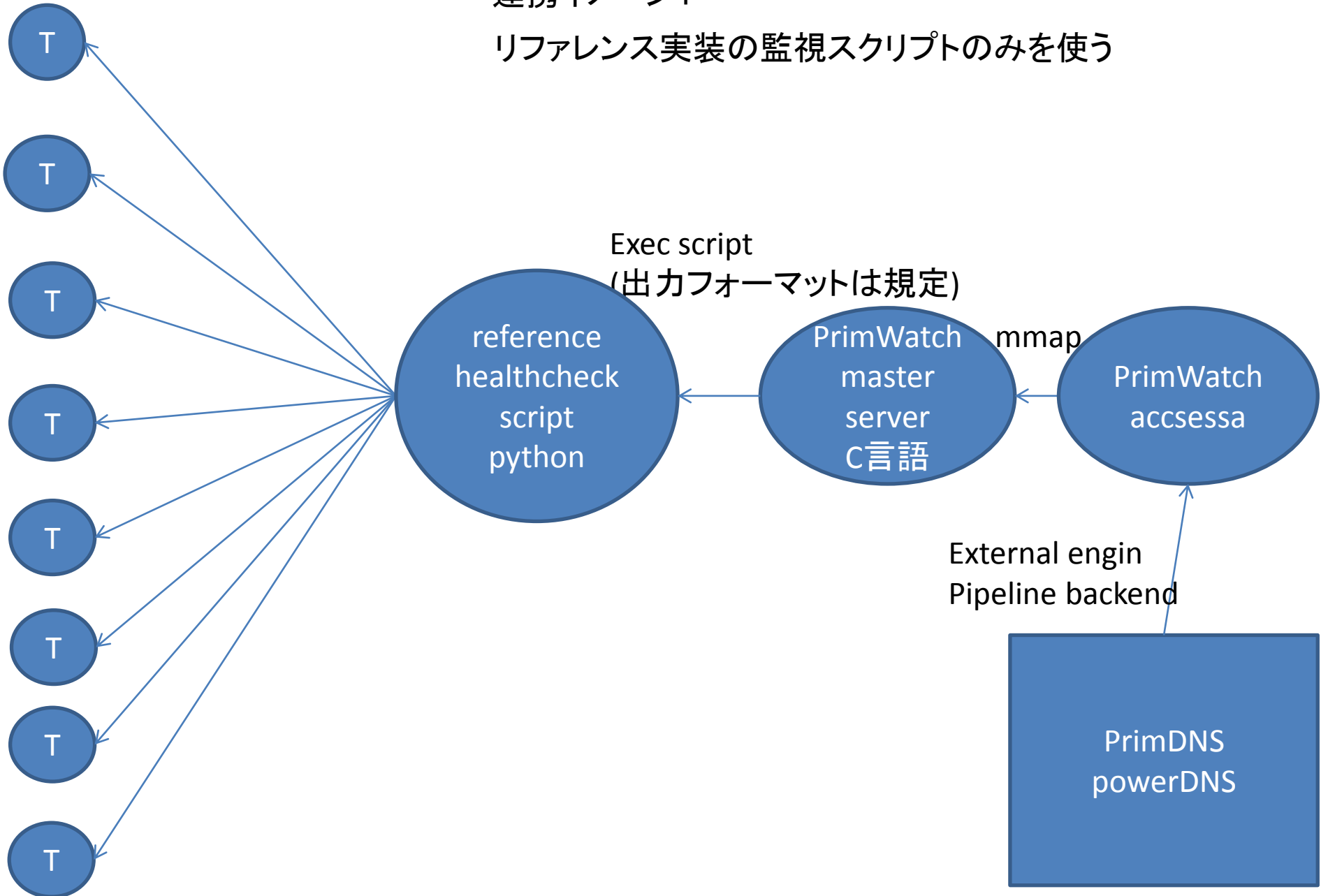
# HealthCheck scriptについて

- TCPポート監視
  - レスポンスの正規表現マッチング
- HTTP監視
  - OKステータスコード複数指定可能
- ICMP監視
- 複数台に対しての監視が可能
- Pythonスクリプト

監視対象

連携イメージ1

リファレンス実装の監視スクリプトのみを使う



監視対象

連携イメージ2  
既存のシステムと組み合わせて使う

