

JANOG56 in MATSUE

Day1 2025年7月30日(水) 16:00～17:00

ECS(RFC 7871)を实际使おうと思うと困ること

～商用利用を前提とした検証結果の共有～

※権威DNSサーバー編



株式会社Jストリーム 高見澤信弘

自己紹介

- ▶ 名前：高見澤信弘
- ▶ 出身地：山形県天童市
- ▶ 所属：株式会社 J ストリーム (AS24253)
 - 新卒で J ストリームへ入社
 - エンジニアリング推進室 & プロダクト企画部（アーキテクト）
- ▶ お仕事
 - CDN(Content Delivery Network)の企画、構築
 - ネットワーク企画
- ▶ 好きなもの
 - ロードバランサー → お家にBIG-IP
 - おうち19インチラック勢
- 活動
 - IPoE協議会 IPv6地理情報共有推進委員会 幹事
 - 海賊版対策実務者意見交換会 海賊版対策技術検証チーム(WG) メンバー



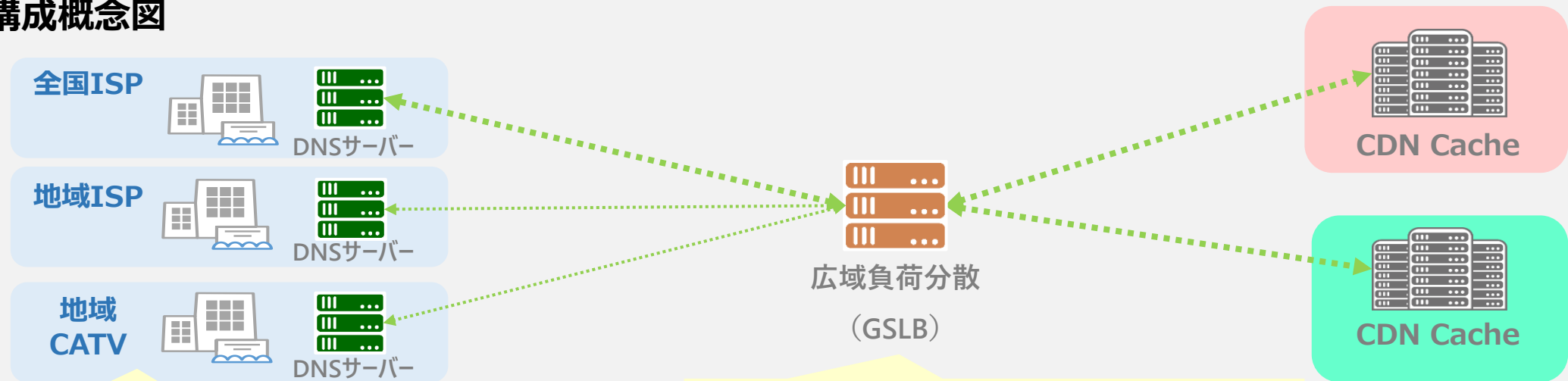
- ▶ コンテンツ配信する側から見たDNSサーバーとECS
- ▶ ISPとCDNのよりよい関係は？
 - ECSのメリット(1)：地域でのトラフィック交換
 - ECSのメリット(2)：キャッシュDNSサーバーのアウトソーシング
- ▶ 検証の概要
- ▶ 実装例と特徴
 - Gdnsd
 - F5 BIG-IP DNS(旧GTM)
- ▶ 検証結果のまとめ
 - 名前解決動作の詳細
 - 権威・キャッシュDNSサーバーの実装の組み合わせで困る例

コンテンツ配信する側から見たDNSサーバーとECS

▶ 求める役割：広域負荷分散(Global Server Load Balancing)

- 複数のサーバーへの負荷分散
- サーバーやデータセンターの死活監視
- トラフィックコントロール：キャッシュDNSサーバーのIPを使った応答

構成概念図



地域ISP/CATVにおいては、ユーザーの近くにキャッシュDNSサーバーを置くことで、GSLB側で適切な地域判定ができる

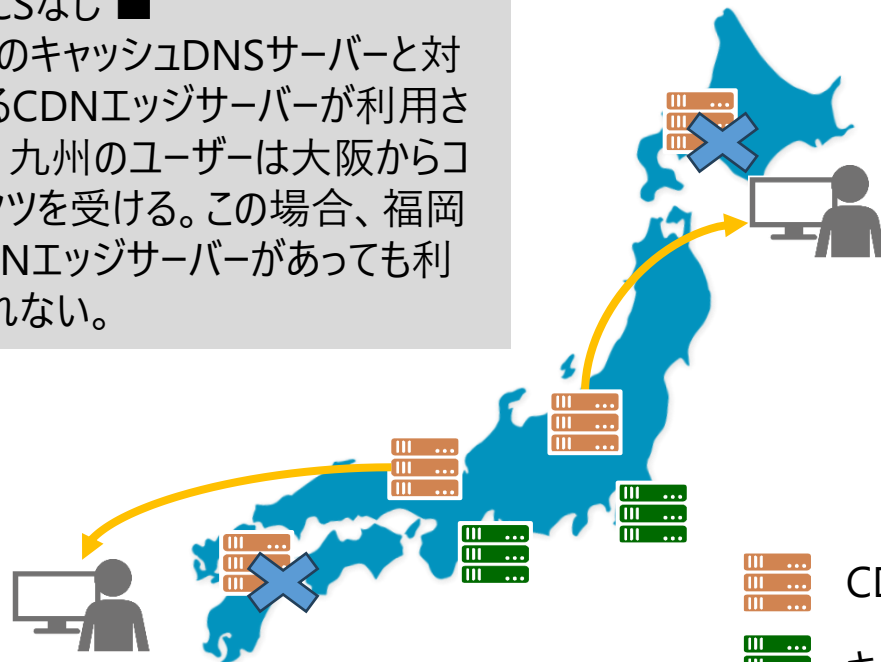
DNSサーバーのIPアドレスに加えて、ECSに含まれるIPアドレスでもコントロールができるようになる = より細かい制御が可能となる

ISPとしてのECSのメリット(1)

- ▶ 地域内でのトラフィックの折り返しが可能となる：全国ISPを想定
 - CDN事業者が地域にもCDNエッジサーバーを分散配置している場合
 - 東京・大阪だけでなく、九州や北海道などのPOPでトラフィック交換が可能
→ 地域分散に対する最適化

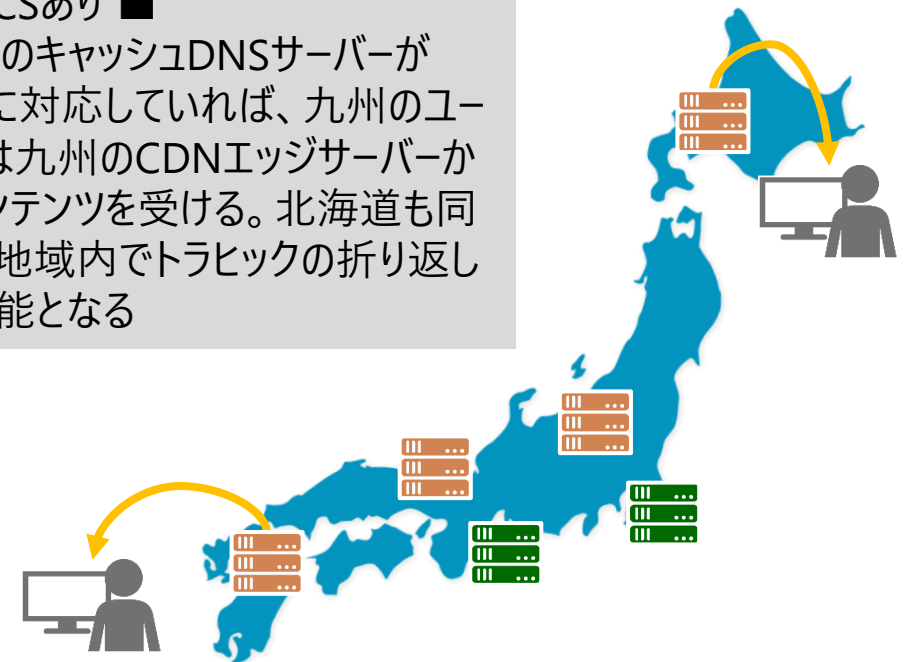
■ ECSなし ■

東阪のキャッシュDNSサーバーと対になるCDNエッジサーバーが利用される。九州のユーザーは大阪からコンテンツを受ける。この場合、福岡にCDNエッジサーバーがあっても利用されない。



■ ECSあり ■

東阪のキャッシュDNSサーバーがECSに対応していれば、九州のユーザーは九州のCDNエッジサーバーからコンテンツを受ける。北海道も同様に地域内でトラフィックの折り返しが可能となる



ISPとしてのECSのメリット(2)

▶ キャッシュDNSサーバーのアウトソーシング時の最適化

- エンドユーザーが参照するキャッシュDNSサーバーが別のネットワークにある場合
- CDN事業者はキャッシュDNSサーバーのIPアドレスを元に配信する
エッジサーバーを決める
- トランジットからトラフィックが流れてくるなど、最適化されない場合がある
- イメージとしてはECSを使ってISPを判別させることができる

地域ISP/CATVにおいては、ユーザーの近くに
キャッシュDNSサーバーを自社で置いていただくこ
とで、GSLB側で適切な地域判定ができる



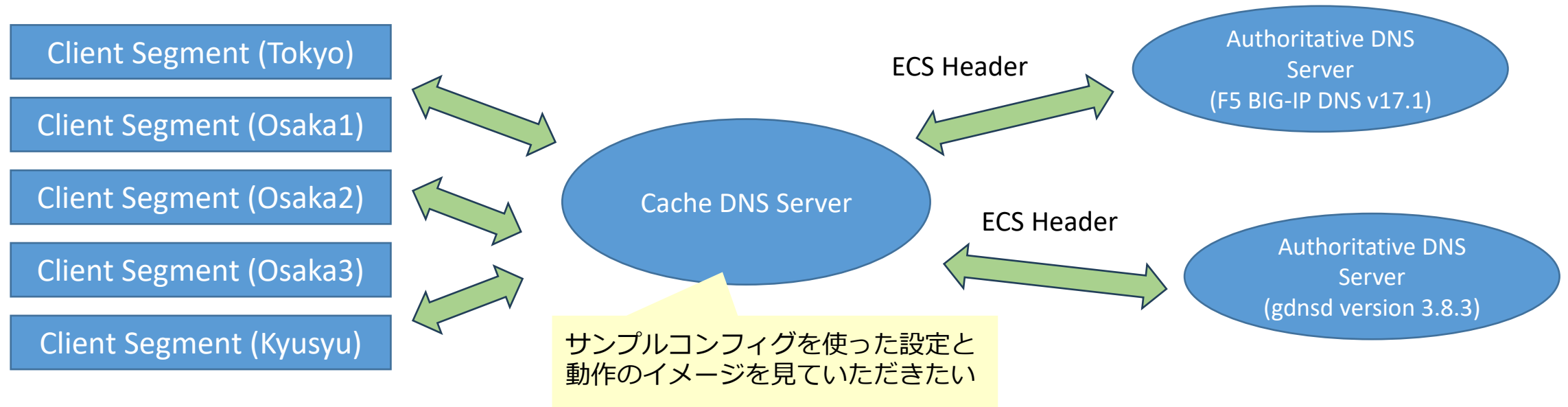
CDNエッジサーバー



キャッシュDNSサーバー



システム構成概要：名前解決動作について

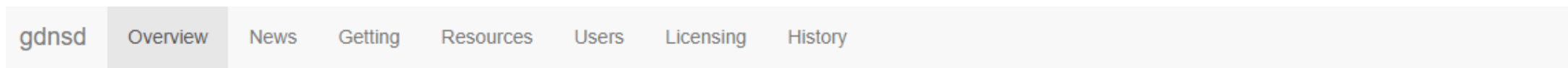


想定地域	ECS(Subnet)	Anser(ServerIP)
大阪1:Osaka1	240b:10:200::/41	2001:db8:10::1/128
大阪2:Osaka2	240b:10:280::/41	2001:db8:10::2/128
大阪3:Osaka3	240b:10:300::/40	2001:db8:10::5/128
九州:Kyusyu	240b:10:400::/39	2001:db8:20::1/128
それ以外:Tokyo	::/0 (default)	2001:db8::1/128

オープンソース実装 : gdnssd

▶ gdnssd : <https://gdnssd.org/>

- 権威DNSサーバー専用のオープンソースなDNSサーバーの実装
- ヘルスチェックを使った監視や応答の切り替えなども可能
- ジオロケーションデータを参照するモジュールもあり、こちらを利用した



gdnssd is an Authoritative-only DNS server. The initial g stands for Geographic, as gdnssd offers a plugin system for geographic (or other sorts of) balancing, redirection, and service-state-conscious failover.

gdnssd is written in C, and uses pthreads with libev and liburcu to attain very high performance, low latency service. It does not offer any form of caching or recursive service, and does not support DNSSEC. There's a strong focus on making the code efficient, lean, and resilient. The code has a decent regression testsuite with full branch coverage on the core packet parsing and generation code, and some scripted QA tools for e.g. valgrind validation, clang-analyzer, etc.

The geographically-aware features also support the EDNS Client Subnet spec from [RFC 7871](#) for receiving more-precise network location information from intermediate shared caches.

Primary source repo on Github: <https://github.com/gdnssd/gdnssd/>

Latest News

- 2024-09-19 - Version 3.8.3 Released

RFC 7871に準拠したECSをサポートする旨の記載がある

実際のコンフィグ : gdnssd

▶ 設定としては

- Conf : ロードバランスなどの設定
- Zone : confファイルで設定したリソース名をどのリソースレコードに紐づけるかを指定

```
# cat zones/example.com
$TTL 86400
```

```
@ SOA ns1 hostmaster (
    2016030300 ; serial
    7200 ; refresh
    30M ; retry
    3D ; expire
    10 ; ncache
)
```

```
@ NS ns1
ns1 A 192.168.1.69
```

```
www 10 DYNA geoip!prod_www
```

右のコンフィグを
wwwに紐づける

各subnetの
宛先を定義

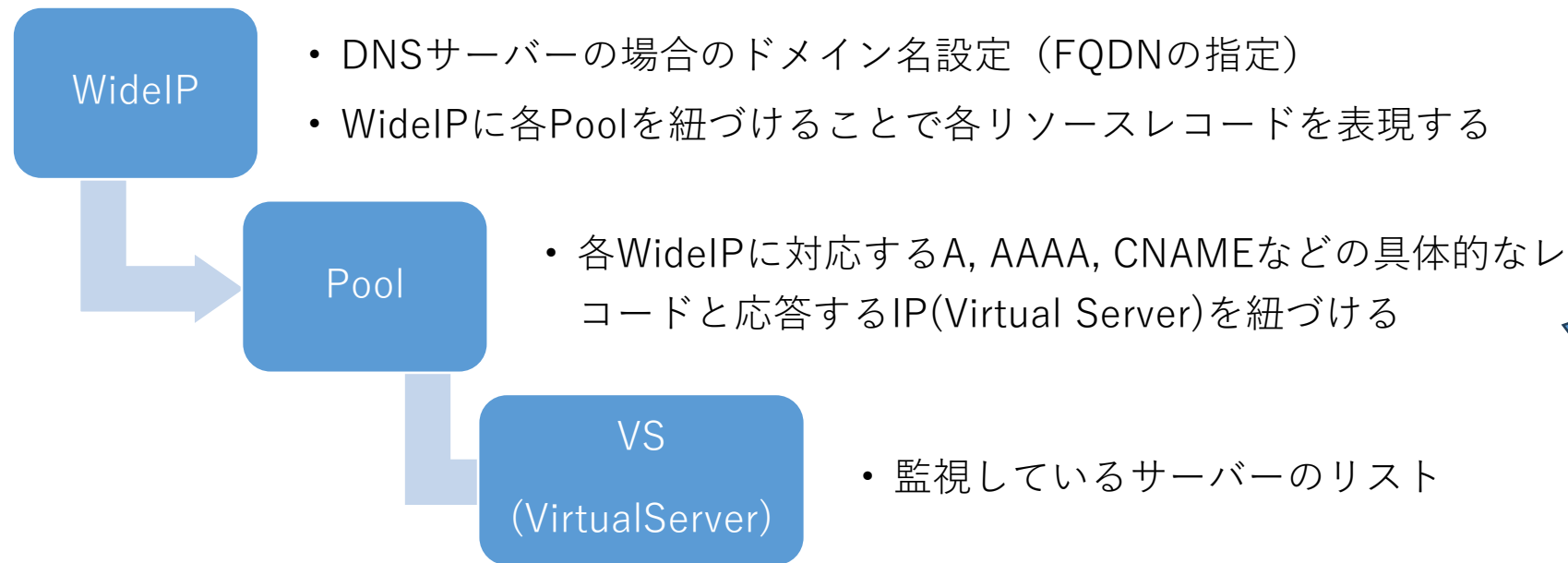
各宛先の回答(IP
address)を定義

```
options => {
    edns_client_subnet => true
}
plugins => {
    geoip => {
        maps => {
            my_prod_map => {
                datacenters => [ dc-01, dc-02, dc-03, dc-04, default ],
                nets => {
                    240b:0010:0200::/41 => [ dc-01 ],
                    240b:0010:0280::/41 => [ dc-02 ],
                    240b:0010:0300::/40 => [ dc-03 ],
                    240b:0010:0400::/39 => [ dc-04 ],
                    ::/0 => [ default ],
                }
            }
        }
    }
}
resources => {
    prod_www => {
        map => my_prod_map
        dcmmap => {
            dc-01 => [2001:db8:10::1 ],
            dc-02 => [2001:db8:10::2 ],
            dc-03 => [2001:db8:10::5 ],
            dc-04 => [2001:db8:20::1 ],
            default => [2001:db8::1 ],
        }
    }
}
}
```

gdnssd : GdnssdPluginGeoip,
<https://github.com/gdnssd/gdnssd/wiki/GdnssdPluginGeoip>
(Visited:2025-06-15)

アプリケーション : f5 BIG-IP DNS

- ▶ GSLBとしては老舗 : 2000年代初頭には使われていたと思われる
- ▶ SLBとGSLBを連携させた監視やロードバランシングが可能
- ▶ 設定は以下のように階層化されている



Pool内に複数のサーバーを設定し、その中からECSが指定するPrefix情報をキーとして、どのサーバーを選択させるか設定した。
A,AAAAだけでなく、CNAMEも設定可能

- ▶ 全体の設定として、ECSを使ったロードバランスを行うように設定する
- ▶ Subnetごとに応答するPool（この場合はIPアドレス）を設定する
 - ロードバランスメソッドとして topology という機能を利用する
 - IPアドレスではなくRegion(Pool)やCNAMEレコードも設定が可能
- ▶ これ以外にFQDN設定や監視設定などの設定が必要

f5 DevCentral Community, Using Client Subnet in DNS Requests, <https://community.f5.com/kb/technicalarticles/using-client-subnet-in-dns-requests/282196> (Visited:2025-06-15)

```
gtm global-settings load-balancing {  
    topology-prefer-edns0-client-subnet enabled  
}  
gtm topology ldns: subnet 240b:0010:0200::/41 server: subnet 2001:db8:10::1/128 {  
    order 1  
    score 100  
}  
gtm topology ldns: subnet 240b:0010:0280::/41 server: subnet 2001:db8:10::2/128 {  
    order 2  
    score 100  
}  
Gtm topology ldns: subnet 240b:0010:0300::/40 server: subnet 2001:db8:10::5/128 {  
    order 3  
    score 100  
}  
gtm topology ldns: subnet 240b:0010:0400::/39 server: subnet 2001:db8:20::1/128 {  
    order 4  
    score 100  
}  
gtm topology ldns: subnet ::/0 server: subnet 2001:db8::1/128 {  
    order 13  
    score 10  
}
```

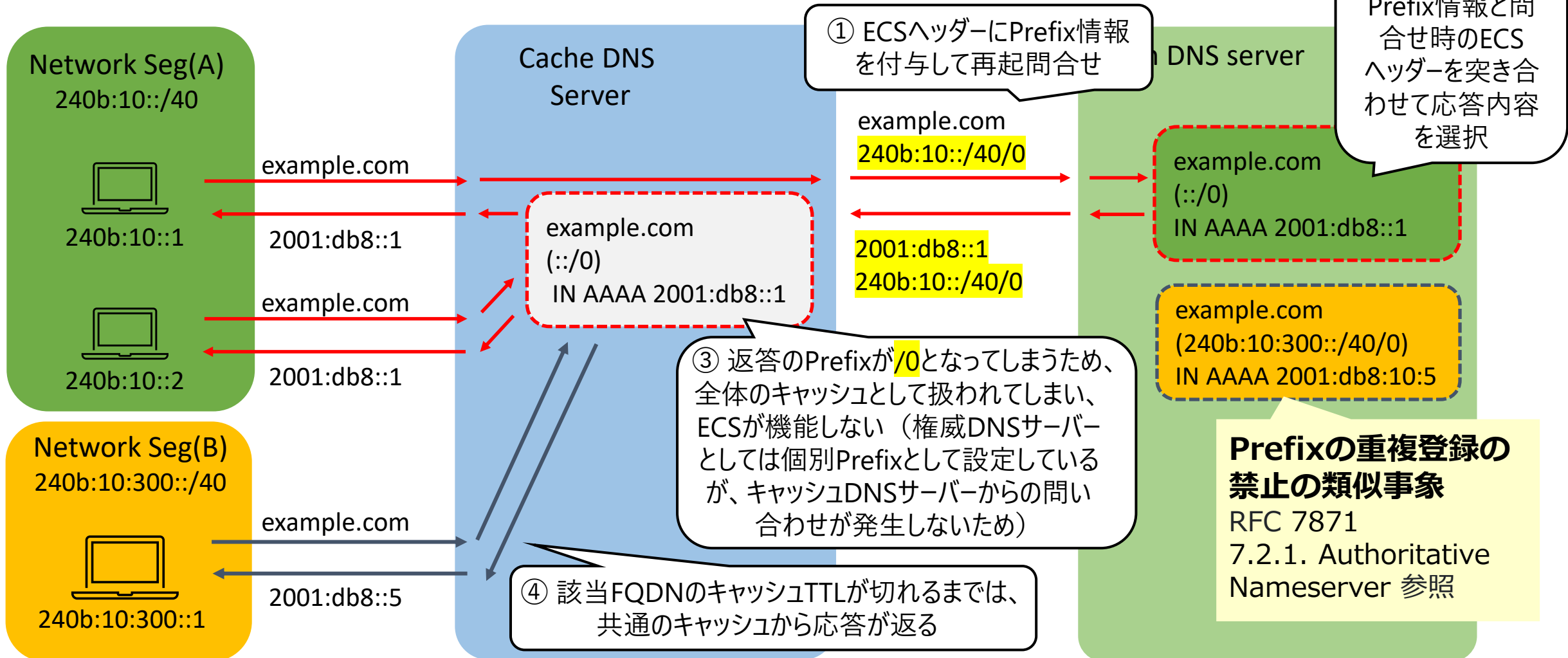
名前解決動作についての詳細

No.	ECS有無	キャッシュDNSサーバーが リクエストするPrefix	権威サーバー設定	応答内容(AAAA / Scope Subnet)	
1	なし	240b:0010:0000::/40/0	Prefix : ::/0 (default) AAAA : 2001:db8::1	2001:db8::1	240b:0010:0000::/40/39
2	あり	240b:0010:0200::/40/0	Prefix : 240b:0010:0200::/41 AAAA : 2001:db8:10::1	2001:db8:10::1	240b:0010:0200::/40/41
3			Prefix : 240b:0010:0280::/41 AAAA : 2001:db8:10::2	2001:db8:10::1	240b:0010:0200::/40/41
4		240b:0010:0300::/40/0	Prefix : 240b:0010:0300::/40 AAAA : 2001:db8:10::5	2001:db8:10::5	240b:0010:0300::/40/40
5		240b:0010:0400::/40/0	Prefix : 240b:0010:0400::/39 AAAA : 2001:db8:20::1	2001:db8:20::1	240b:0010:0400::/40/39
6		240b:0010:0500::/40/0	AAAA : 2001:db8:20::1	2001:db8:20::1	240b:0010:0400::/40/39

- ▶ No.1はScope Subnetが/0で返ってくると予想していたが、結果は/39で返ってきた
 - 次の設定済みPrefixが240b:0010:0200::/40から始まるので、その中で一番大きいsubnet maskを返答している
 - 権威DNSサーバーの実装によっては/0が応答される場合もあり、実装による差異が見られる
- ▶ No.2,3は権威DNSサーバー側で/41と設定しても、キャッシュDNSサーバー側のprefix設定が/40となっているので、キャッシュDNSサーバーからのリクエストが/40 でしか流れてこない
 - 権威DNSサーバー側で/41に個別に設定しても回答が使われない(2001:db8:10::2が使われない)

権威・キャッシュDNSサーバーの実装の組み合わせで困る例

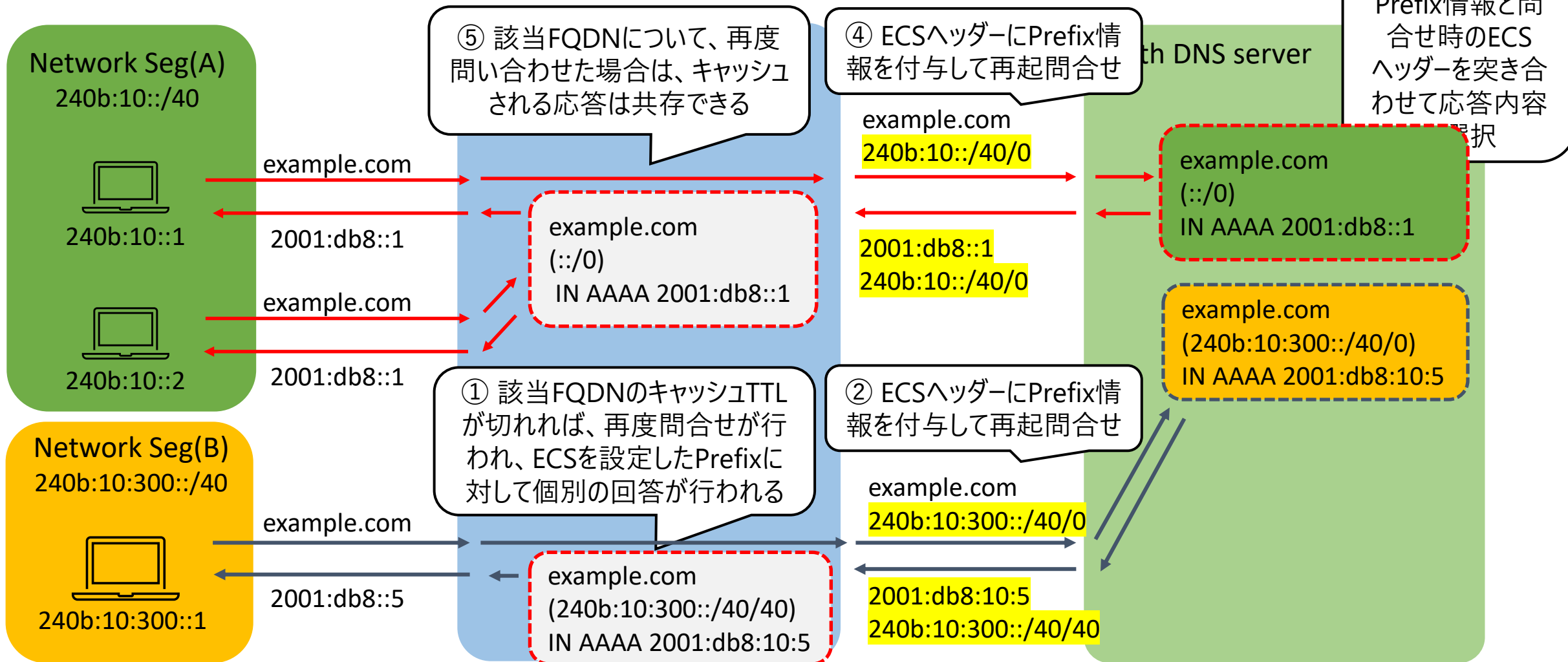
▶ 問合せのタイミングによってECSが有効にならない場合があります厄介



権威・キャッシュDNSサーバーの実装の組み合わせで困る例

(動いてしまう例)

▶ 同じ設定でも問い合わせの順序によっては動いてしまう場合もある



- ▶ ECSを使っている方いらっしゃいますか？
- ▶ ECSが普及しないのはなぜか？
 - キャッシュDNSサーバーのリソース消費を事前に“予測”するのはかなり難しい
 - キャッシュTTLとキャッシュできる量によってキャッシュヒット率にも影響が出てくる
- ▶ 安全にECSを有効にするためのアイデア
 - Prefix毎に試験（答え合わせ）ができる環境があると導入しやすい？
 - まずは顔が見える相手とトライアルする？
- ▶ ECSを使っても最適化するためには、キャッシュと権威の連携が必要
 - キャッシュDNSサーバーと権威DNSサーバーの運営者が連携するための枠組み
 - ECSの対象とするドメイン名、付与するSubnetの大きさなど
- ▶ どの程度トラフィックコントロールしたい（してほしい）ですか？（ISP側のご意見を）