委任にまつわるエトセトラ

～と、一つの小さな技術的アプローチ～

2012年9月1日

DNS Summer Days 2012

株式会社日本レジストリサービス（JPRS）

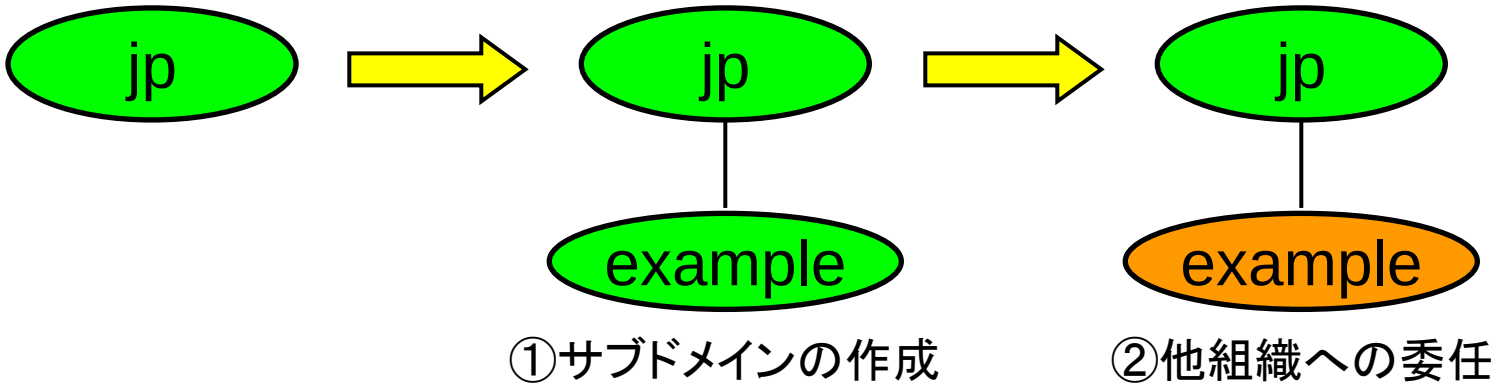
森下 泰宏

内容

- DNSの特徴である「ゾーンに分割された階層構造を持つ名前空間」を形作り、
- いにしえの昔から現在に至るまで、DNSにおけるさまざまな障害の原因や脆弱性（攻撃）の標的にもなった委任（delegation）の概要について改めて解説し、
- 後半では、本セッション終了後のディスカッションにおけるトピックスの中心になるであろう、いわゆる親子同居問題の内容と、
- その問題を解決するための一つの小さな技術的アプローチの内容についてご紹介します

おさらい:そもそも委任とは何か？

- 目的:ドメイン名管理の分散化
  ① 自分が管理するドメイン名にサブドメインを作成し、
  ② そのサブドメインの管理権限を他の組織に任せる
  ことにより実施される



おさらい：ゾーンとは何か？

- 前ページの例では委任により、ドメイン名の管理の範囲がjpとexample.jpの2つに分割されている
- この管理範囲のことをゾーンと呼ぶ
  - jpゾーン
  - example.jpゾーン
- ゾーンという言葉を使った説明例（前ページの例）：
  - jpゾーンがjpゾーンとexample.jpゾーンに分割された
- この説明もよく見る（正しい）：
  - jpゾーンからexample.jpゾーンが切り出された
- いずれにせよ委任により、元のゾーンの管理の範囲はそれ以前よりも小さくなる

ゾーンの分割は委任の成立により発生

- 別の言い方：委任の成立がゾーンの分割の必要条件
- そのため、以下の説明（よく見かける）は、実はあまり適切ではない
  1. jpゾーンをjpゾーンとexample.jpゾーンの2つのゾーンに分割
  2. 分割したゾーンを他組織に委任
- この説明は「サブドメインごとに必ず別ゾーンへの分割が必要」という、誤った認識を持たれる可能性がある
- 以下の説明がより適切
  1. jpゾーン内にサブドメインexample.jpを作成
  2. 作成したサブドメインを他組織に委任
  3. 委任の成立により、jpゾーンがjpゾーンとexample.jpゾーンの2つのゾーンに分割

事例：jpとco.jp

- 現在のjpとco.jpは別のゾーンに分割されておらず、いずれもjpゾーンに属している
  - 2006年の更新間隔短縮の導入に伴い変更
- 誤解を生まない（後付けの）説明：「co.jpはjpと同じJPRSが管理しているので、JPRSではこの説明（再掲）の2.から先を実施していません」

1. jpゾーン内にサブドメインco.jpを作成
2. 作成したサブドメインを他組織に委任
3. 委任の成立により、
   jpゾーンがjpゾーンとco.jpゾーンに分割

「委」「任」「譲」:暗黙に他者を想定？

- 委任の2文字は「委」と「任」
  - つまり、「委ねる」と「任せる」
- delegationのもう一つの訳語「委譲」は「委」と「譲」
  - つまり、「譲る」
- つまり、委任（委譲）という行為は、委任先として暗黙のうちに他者を想定していたのではないかと考えられる
- それを裏付けるように、DNSの名前解決アルゴリズムについて記述したRFC 1034の5.3.3. Algorithmには、
  - if the response contains a better delegation to **other** servers

  という記述がある

「自分への委任」はDNS仕様の想定外？

- 以上の背景から「自分への委任」という行為は、DNSのプロトコル仕様において想定の範囲外だったのではないかと、私は考えている
  - 今度モカペトリスさんに会ったら聞いてみよう・・・
- つまり、DNSにおける親子同居はそもそも仕様外、つまり未定義の設定であり、どのような動作をすべきなのかが明確ではなく、さまざまな問題（バグ）も発生しやすい、のではないだろうか…

ということで、親子同居問題再び

- また、いわゆる「親子同居問題」について考える場合、以下の2つを場合分けして考える必要がある
  - 1）一見正しそうな、親からの正当な委任がある場合
  - 2）一見まずそうな、親からの正当な委任がない場合
- 前述した想定外の動作は、一見正しそうな1）の場合にも影響を及ぼしている
  - 以降で説明

どう動くべきか？

```
$ORIGIN example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
sub  IN NS ns1.example.ne.jp.

$ORIGIN sub.example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
www  IN A 192.0.2.1
```

- example.jpとsub.example.jpの2つが、ISPが運営する1台の権威DNSサーバー、ns1.example.ne.jpにより管理されている
- このサーバーに対し、通常の名前解決によってwww.sub.example.jpのAレコードが問い合わされた場合、どのように動作すべきなのか？

親が入っていない場合

```
$ORIGIN example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
sub  IN NS ns1.example.ne.jp.

$ORIGIN sub.example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
www  IN A 192.0.2.1
```

- 私はwww.sub.example.jpのIPアドレスを知っている
  - 192.0.2.1を返す
- これは当然

親が入っている場合（委任あり）

```
$ORIGIN example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
sub  IN NS ns1.example.ne.jp.

$ORIGIN sub.example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
www  IN A 192.0.2.1
```

- 親ゾーンが入っていても、別ゾーンだから影響はないはず
  - データの管理はゾーンごと
- 私はwww.sub.example.jpのIPアドレスを知っている
  - 192.0.2.1を返す

→問題なさそうだ

- でもこの場合、委任情報を含め、親ゾーンの内容は参照されないのね
  - 何だか釈然としないな・・・

親が入っている場合（委任なし）

```
$ORIGIN example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
sub  IN NS ns1.example.ne.jp.

$ORIGIN sub.example.jp.
@    IN SOA …
     IN NS ns1.example.ne.jp.
www  IN A 192.0.2.1
```

- 親ゾーンが入っていても、別ゾーンだから影響はないはず
  - データの管理はゾーンごと
- 私はwww.sub.example.jpのIPアドレスを知っている
  - 192.0.2.1を返す

→え？本当にそれでいいの？

  - だって私、委任してないって知ってるよ？
- その子は本当に私の子なの？

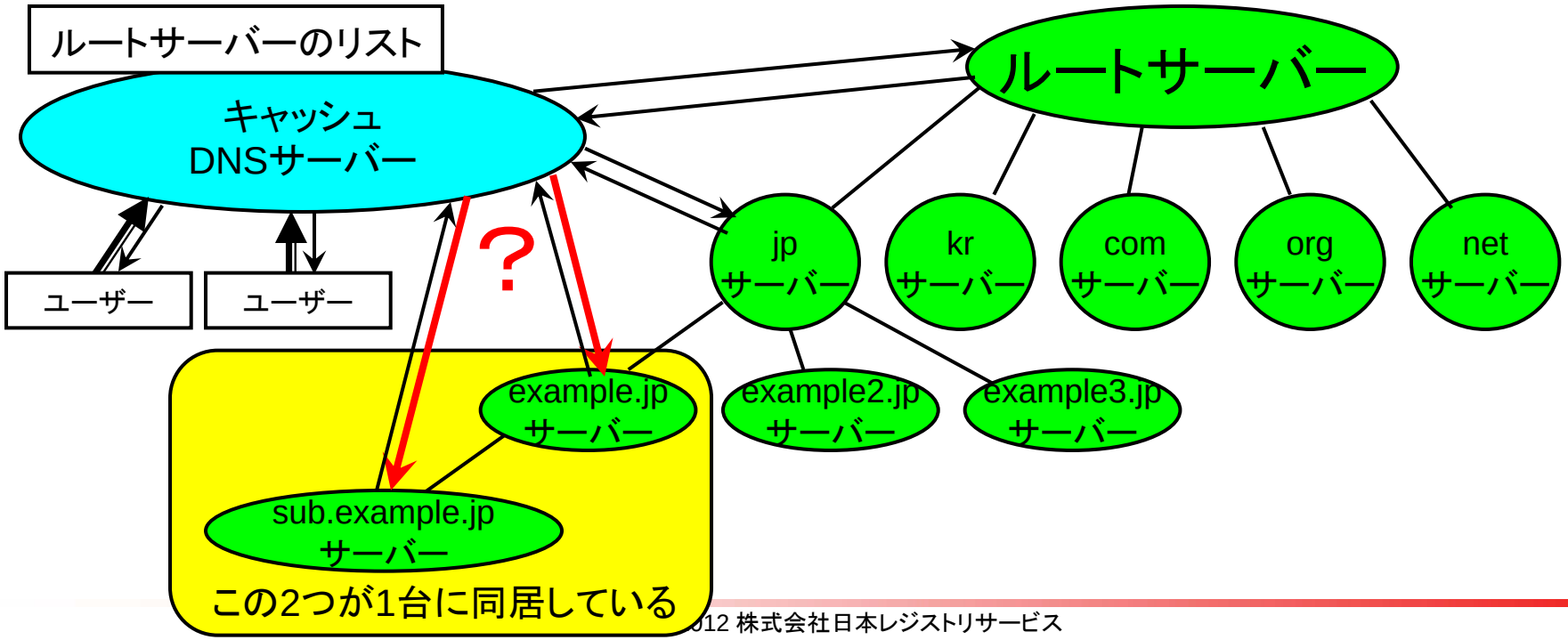
現在の多くの実装の動作

- BIND 9やNSDなど現在の多くの実装では、
  - 「私はwww.sub.example.jpのIPアドレスを知っている → 192.0.2.1を返す」

  という動作をする
- つまり、同居している親から委任があるかどうかは、動作には関係しない
- 委任してないって知ってるのに？
  - うん、確かにそうなんだけど・・・

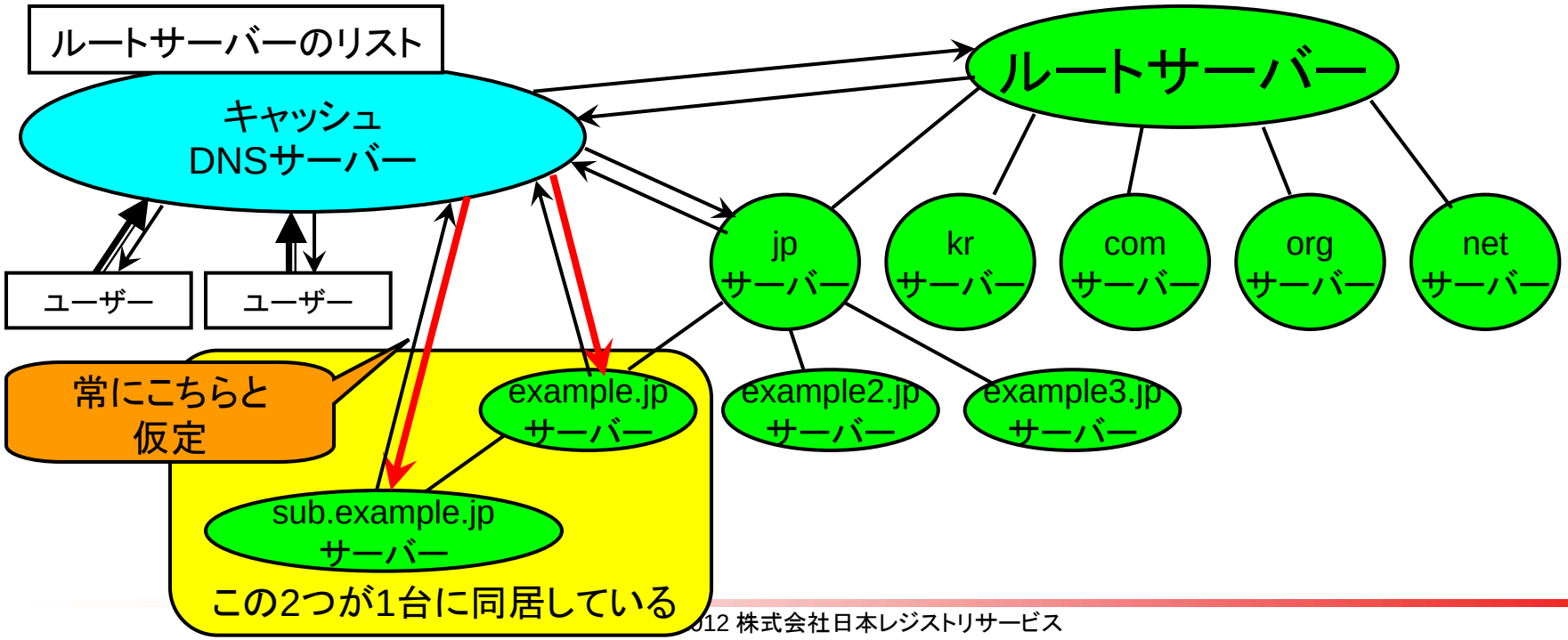
なぜこうなってしまうのか

- example.jpゾーンとsub.example.jpゾーンの権威DNSサーバーが1台に同居している
- そのため、キャッシュDNSサーバーからの問い合わせがexample.jpゾーンの権威DNSサーバーに対するつもりのものなのか、sub.example.jpゾーンの権威DNSサーバーに対するつもりのものなのかを、権威DNSサーバー側から見た場合に判別できない



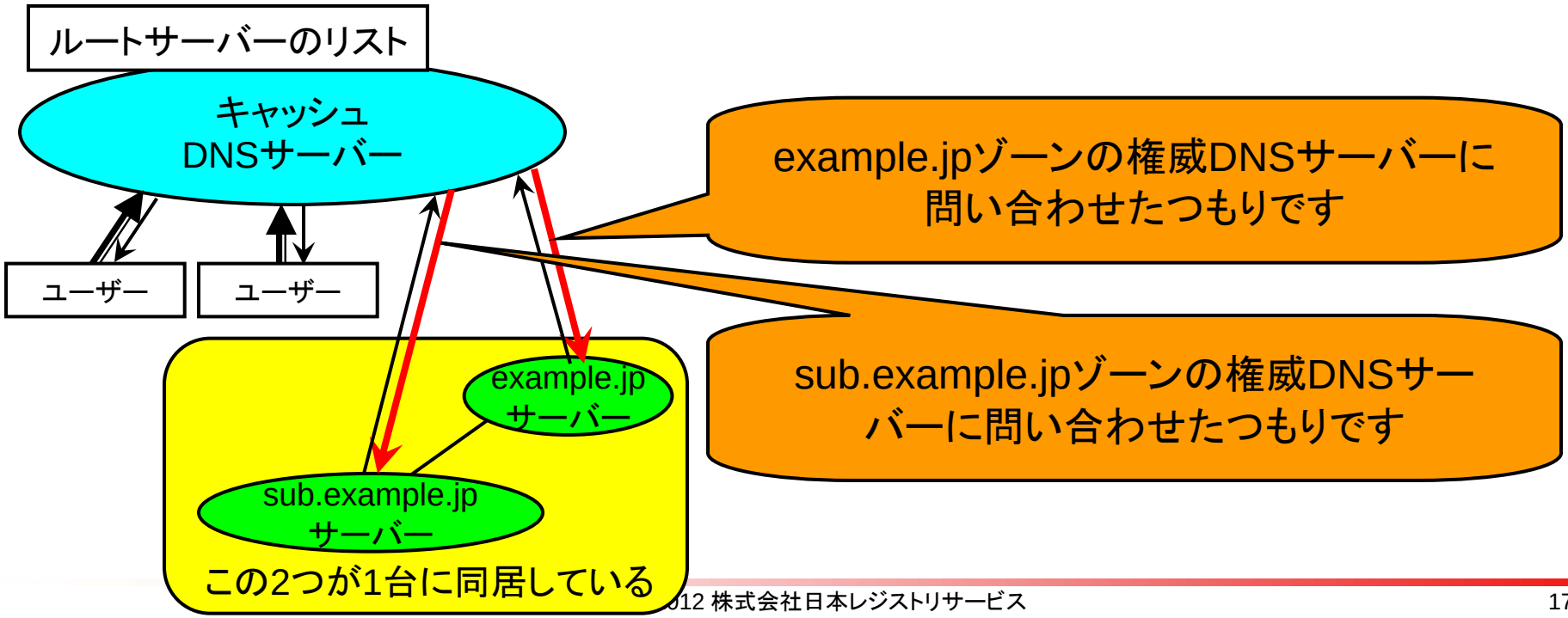
なぜこうなってしまうのか（続き）

- そのため、現在の多くの実装では**より狭い範囲**、つまりsub.example.jpゾーンの権威DNSサーバーに対するものであると**仮定した**応答をするように作られている（ようだ）
  - 「自分の知っているうち**一番詳しいもの**を答える」という動作



小さな技術的アプローチ

- 問い合わせの際に「どのゾーンの権威DNSサーバーに対して問い合わせたつもりなのか」という情報を、キャッシュDNSサーバー側から伝えられるようにすればいいのではないか？
- そうすれば権威DNSサーバー側で、それに応じたしかるべき応答を作成できる



DNS/1.1？

- このアイディアはtwitterのタイムライン上で、東大亮さん（@hdais）と私がほぼ同時に思いついた
  - 東さんは具体的な実装のアイディア（下記）まで・・・
- EDNS0を使って、キャッシュDNSサーバーからの問い合わせパケットに親から得たゾーン情報を追加する
  - EDNS0を使うことで、従来からの問い合わせには従来と同じ応答を返すという、後方互換性も確保できる
- HTTP/1.1の名前ベースのバーチャルホスト拡張に相当
  - Hostヘッダ（例：Host: www.sub.example.jp）
  - HTTPでは共有サーバーの問題をこれで解決
- 親子同居問題を技術的に解決可能
  - ただし、導入の副作用については別途検討が必要

Q&A