30分で学ぶDNSの基礎の基礎
～DNSをこれから勉強する人のために～

2014年9月27日
SECCON 2014 長野大会 DNS Security Challenge
株式会社日本レジストリサービス（JPRS）
森下 泰宏（Yasuhiro Orange Morishita）
@OrangeMorishita

自己紹介

- 氏名：森下 泰宏（もりした やすひろ）
- 勤務先：（株）日本レジストリサービス
- 肩書：広報宣伝室 技術広報担当
- 主な業務内容：ドメイン名・DNSに関する技術情報をわかりやすく伝える
- 最近は「重複をお許しくださいの人」と呼ばれることが多いです
  - 私がMLに出す注意喚起の書き出しに由来している…ようです

で、「JPRS」ってそもそも何者？

- 正式名：株式会社日本レジストリサービス
  - Japan Registry Services Co. Ltd.
  - 略称：JPRS（ジェーピーアールエス）
- 「IT企業」と呼ばれるものの一つ
- トップレベルドメイン（TLD）「jp」のレジストリを担当している

JPRSはjpの「レジストリ」

- レジストリの役割
  - どのドメイン名をどの組織・個人が使っているか、という情報をきちんと管理する
  - インターネットでドメイン名を使えるようにするための、DNSと呼ばれる仕組みの一部分を管理運用する
    - 具体的にはそのドメイン名の「権威DNSサーバー」
- 「jp」は「日本」に割り当てられている
  - 国コードトップレベルドメイン：ccTLD
  - 例：seccon.jp, jprs.co.jp, nagano-u.ac.jp, ...
- JPRSはccTLD「jp」のレジストリとしてそれを管理し、インターネットで使えるようにしている会社

これから話す内容

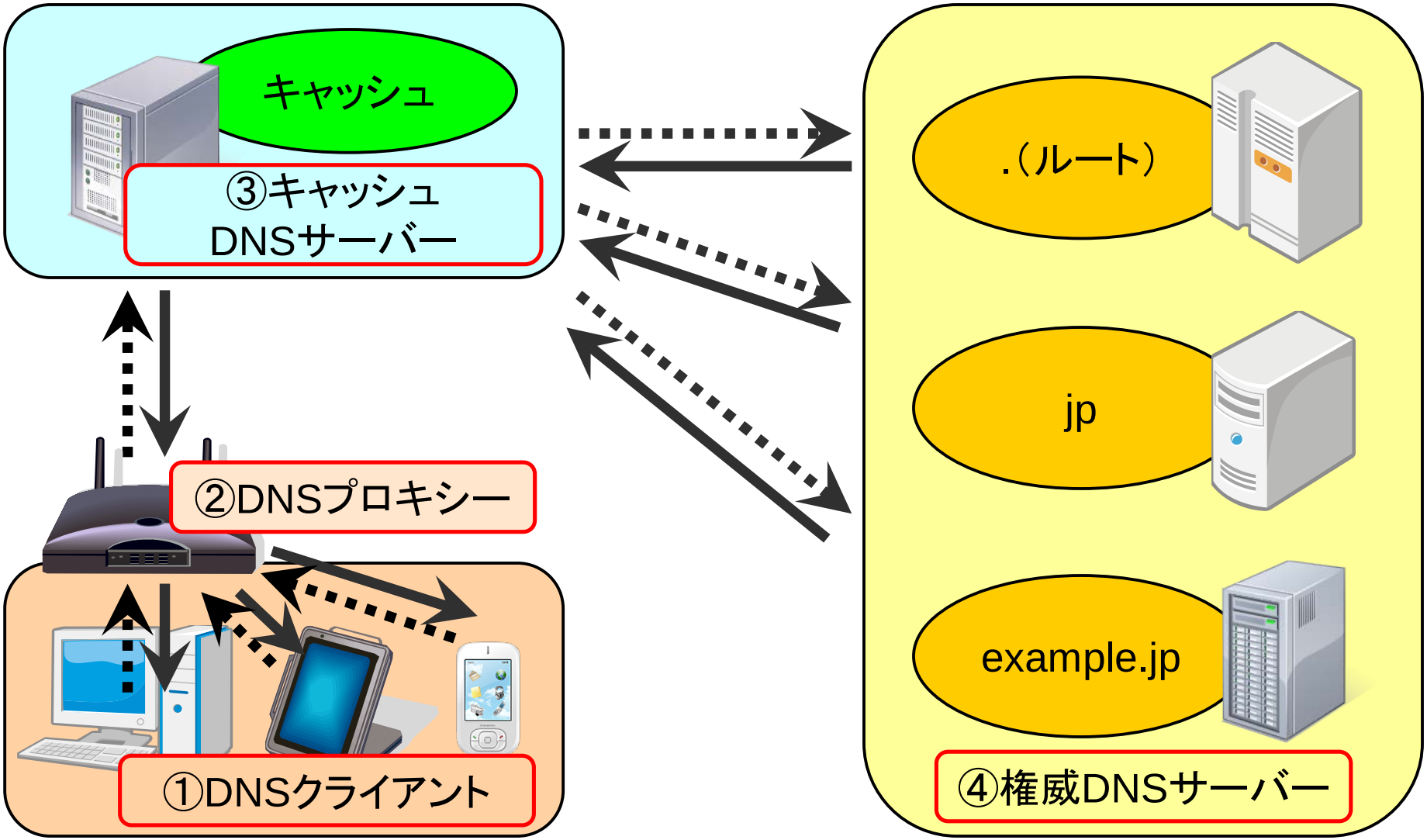
1. DNSの構成要素とその役割
2. DNSを勉強するにあたっての注意点
   - 初学者が特にはまりやすい4つのポイント

DNSをこれから勉強しようと思っている人が、
まず頭に入れておくべきことを簡単に解説します

注意：DNSが30分で学べる、というわけではありません

1. DNSの構成要素とその役割

DNSの構成要素（4種類の登場人物）



登場人物（構成要素）とその役割

名前解決を実行する

③キャッシュ
DNSサーバー

名前解決の依頼・応答を
中継する（ない場合もある）

②DNSプロキシー

名前解決を依頼する

①DNSクライアント

.（ルート）

jp

実データ（各ゾーン）を管理する

④権威DNSサーバー

※名前解決：ネットワーク上の機器に付けられた名前から、その機器に割り当てられているアドレスを求めること

まとめ：各構成要素の役割とその実例

① DNSクライアント：名前解決を依頼する
- 実例：Webブラウザー、各アプリなど

② DNSプロキシー：名前解決の依頼・応答を中継する
- 実例：ホームルーターなど
- DNSプロキシーが存在しない場合もある

③ キャッシュDNSサーバー：名前解決を実行する
- 実例：Google Public DNS

④ 権威DNSサーバー：実データ（各ゾーン）を管理する
- 実例：Amazon Route 53
- DNSでは管理するそれぞれの単位のことをゾーンと呼ぶ

これらの構成要素とその役割を把握することが
DNSの理解への第一歩

2. DNSを勉強するにあたっての注意点

初学者が特にはまりやすい4つのポイント

初学者が特にはまりやすいポイント

- 以下の4つに特に注意
  - ① 名称の不統一
  - ② 二種類のDNSサーバー
  - ③ 二種類の問い合わせ
  - ④ 兼用可能な実装
- これら4つは、初学者が特にはまりやすいポイント
  - かつ、一度はまると抜け出しにくい
  - そのため、初学者でなくても・・・

注意点①：名称の不統一

- 各構成要素を示す用語が統一されていない
- 文献や著者の違いなどにより、それぞれの構成要素がいろいろな名称で呼ばれている
  - 技術者や専門家間においても違いや使い分けがある
    - 何に注目するかによる使い分けなど
- 混乱を招きやすく、理解の妨げになりやすい
- 日本語だけでなく、英語でも統一されていない

そのため、どの名称がどの構成要素（機能）を指しているかの把握が重要

使われている名称の例

- DNSクライアント：スタブリゾルバーなど
- DNSプロキシー：DNSフォワーダーなど
- キャッシュDNSサーバー：フルリゾルバー、フルサービスリゾルバー、参照サーバーなど
- 権威DNSサーバー：DNSコンテンツサーバー、権威ネームサーバー、ゾーンサーバーなど

自分が使う名称・相手が使っている名称に
気を遣うようになったら一歩前進

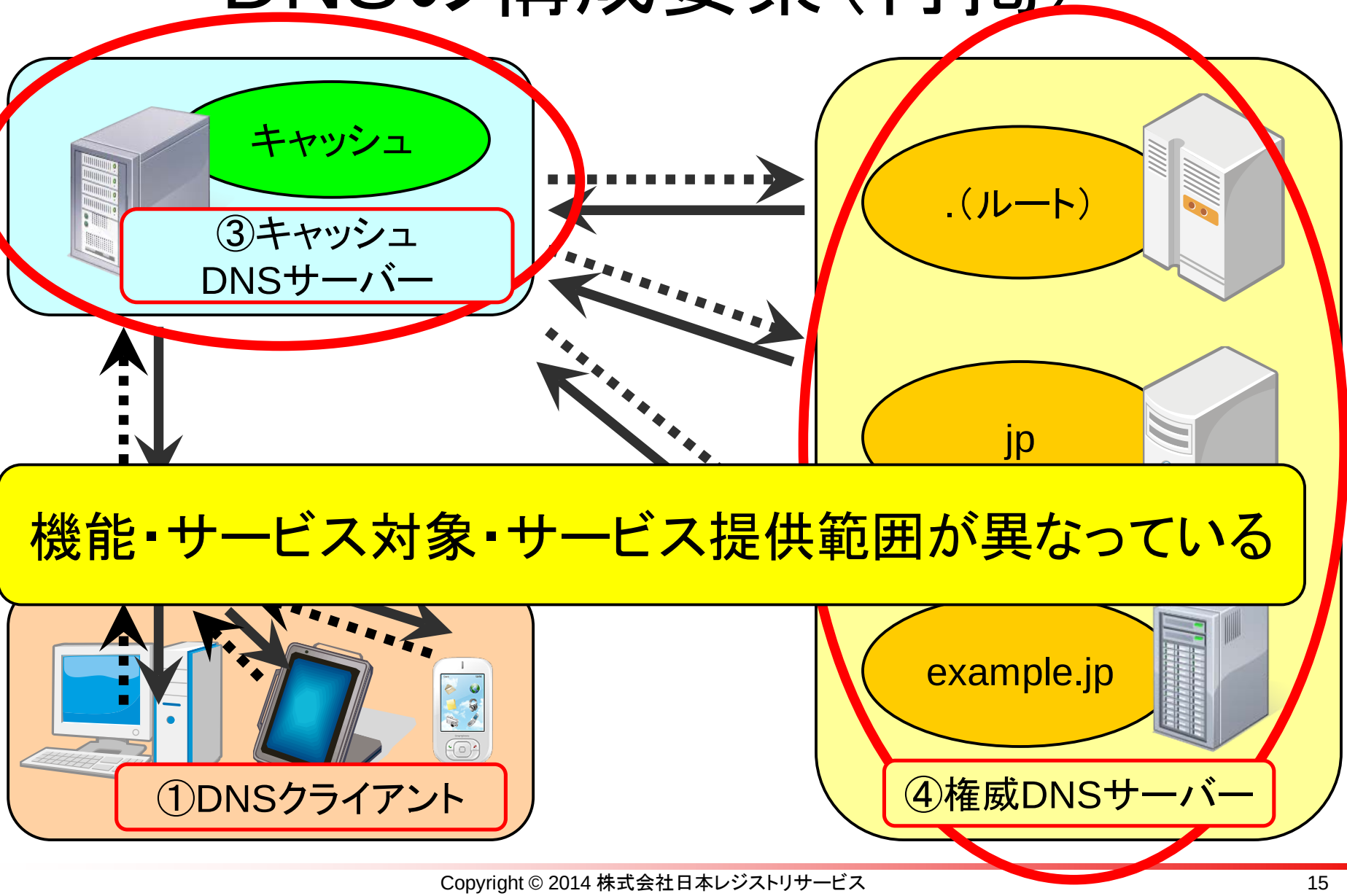
事後資料で更新：（あるいは双方を指しているのか）の記述を追加

注意点②：二種類のDNSサーバー

- DNSサービスを提供するサーバーは二種類ある
  - キャッシュDNSサーバーと権威DNSサーバー
- これらのサーバーは機能、サービス対象、サービス提供範囲が異なっているにもかかわらず、いずれも「DNSサーバー」と呼ばれている
- 「DNSサーバー」という用語が単独で使われた場合、そのどちらを指しているのか（あるいは双方を指しているのか）を常に意識・把握しておく必要がある

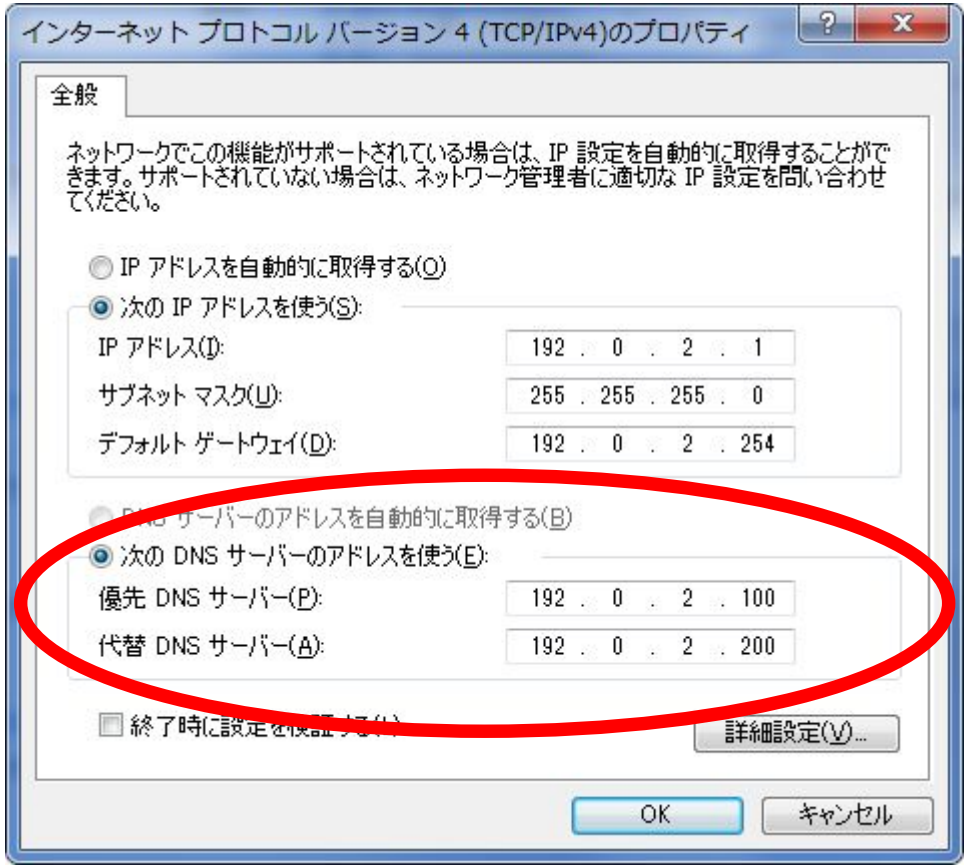
作成者の理解不足・説明不足などにより、
それが判然としない文書も数多く存在する

DNSの構成要素（再掲）



事後資料で更新:(DNSプロキシーの場合もある)の記述を追加

Q:どちらのDNSサーバーを指している?



例1:WindowsのTCP/IPのプロパティ

キャッシュDNSサーバー
(DNSプロキシーの場合もある)

例2:TechNet「DNSサーバーを計画する」

権威DNSサーバー

まとめ：二種類のDNSサーバー

	キャッシュ DNSサーバー	権威 DNSサーバー
機能	階層構造をたどり ドメイン名を検索する	階層構造を構成し ドメイン名を管理する
サービス対象	ISPや組織などの 利用者 （DNSクライアントや DNSプロキシー）	インターネット上の キャッシュ DNSサーバー
サービス提供範囲	通常はISP内や 組織内に限定	インターネット 全体

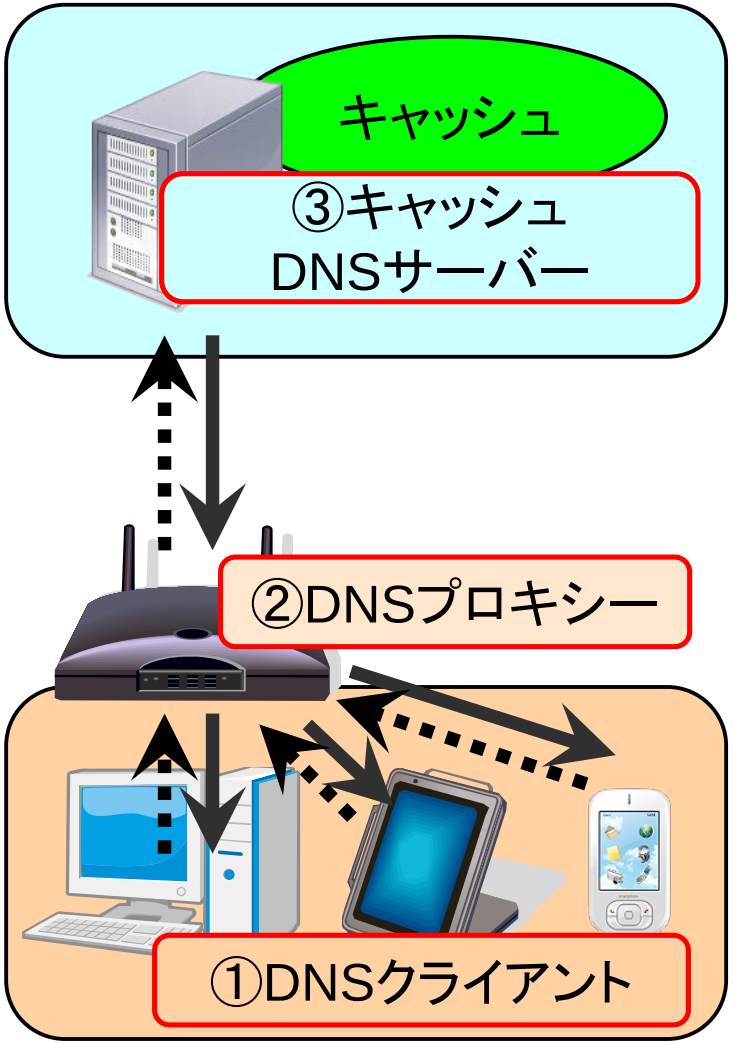
双方を示す場合を除き「DNSサーバー」を単独で使わず、「キャッシュDNSサーバー」「権威DNSサーバー」を使い分けるのがおすすめ

注意点③：二種類の問い合わせ

- DNSには、役割の異なる二種類の「問い合わせ」が存在する
  - 再帰問い合わせと非再帰問い合わせ
- これらの問い合わせは機能・動作が異なっており、明確に区別して考える必要がある

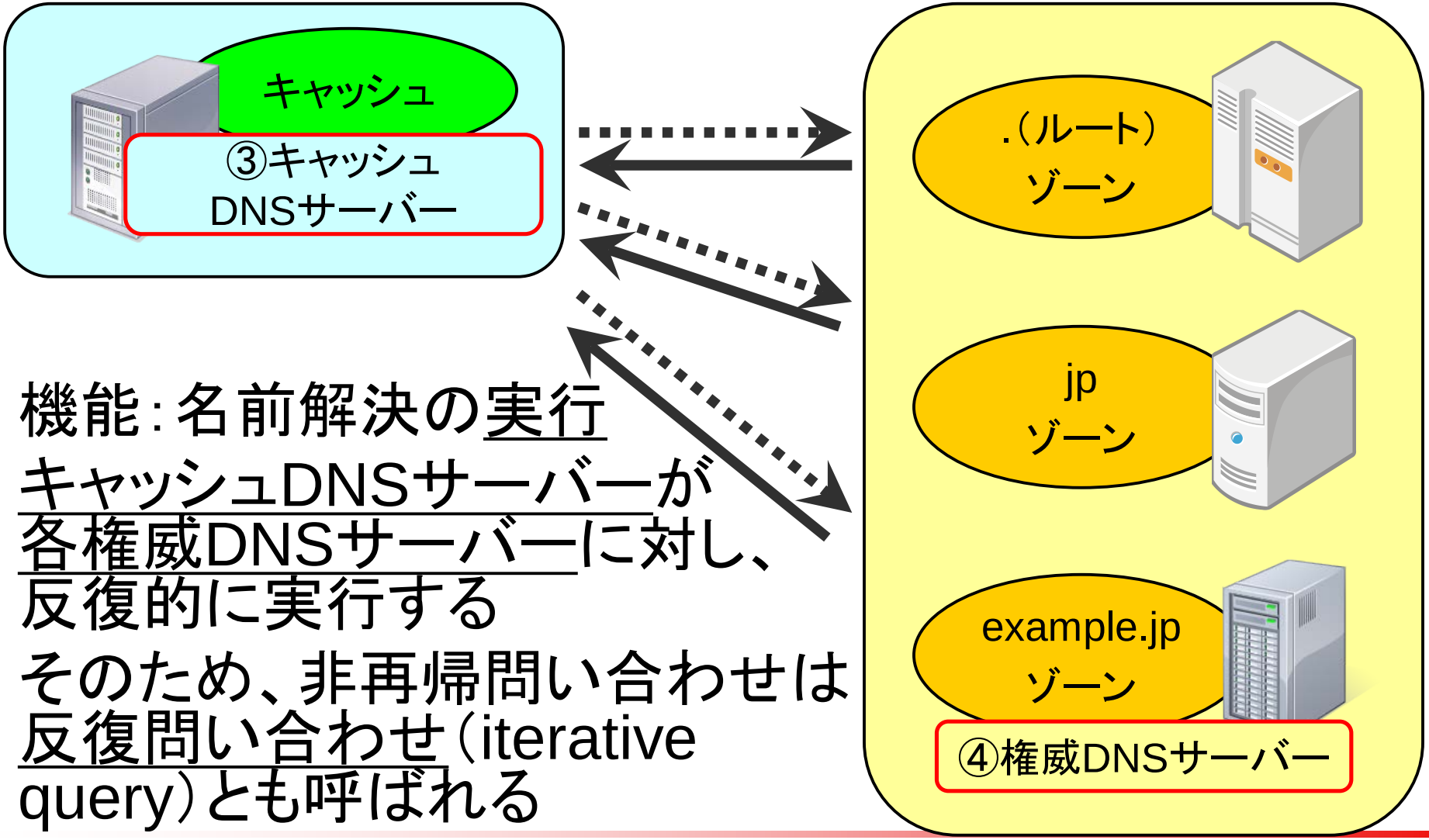
そうしなければ、名前解決の仕組みを理解できない

再帰問い合わせ（recursive query）



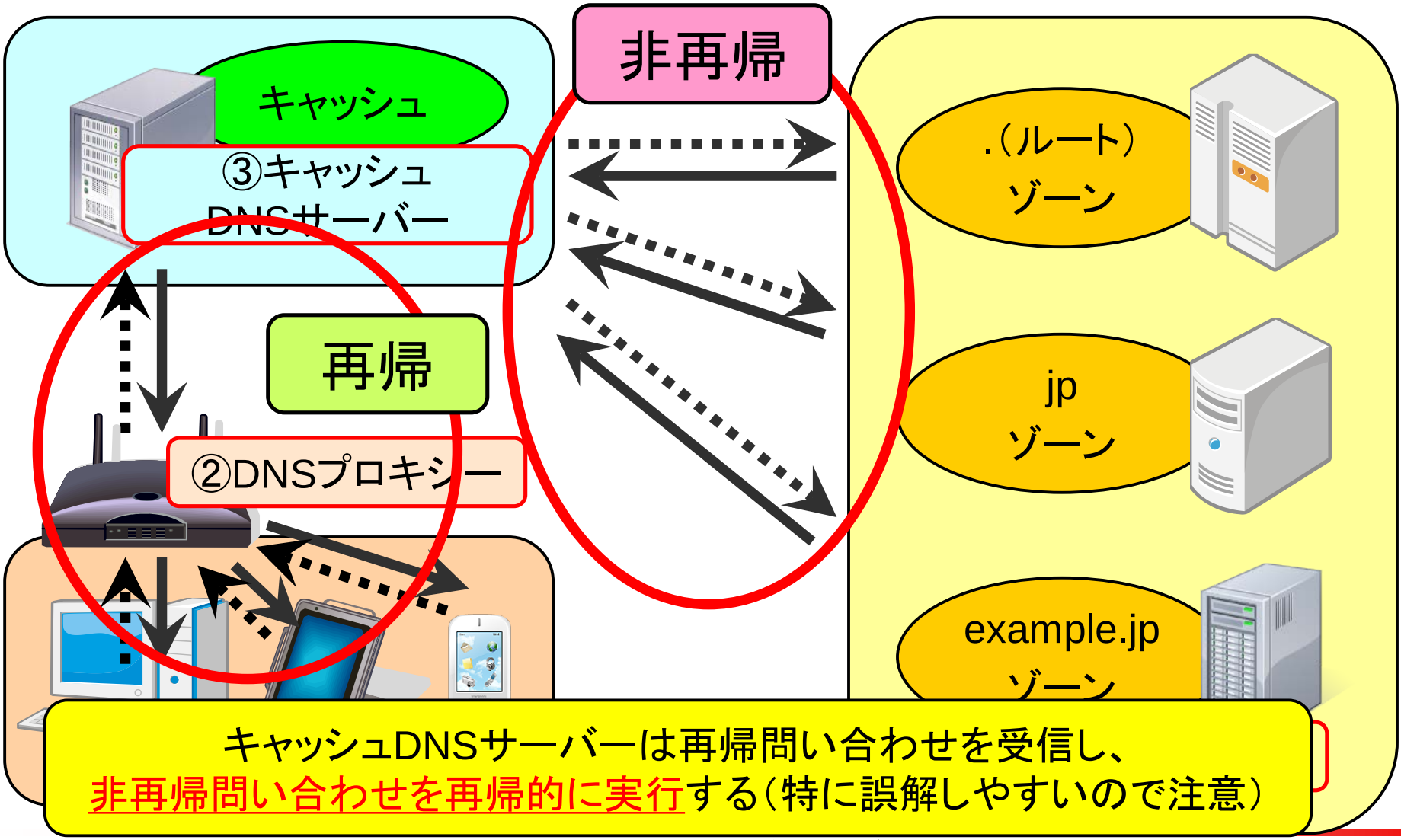
- 機能：名前解決の依頼
- DNSクライアントやDNSプロキシーがキャッシュDNSサーバーに対し、必要に応じて実行する
- DNSクライアントやDNSプロキシーは再帰問い合わせによって、キャッシュDNSサーバーに名前解決を要求する

非再帰問い合わせ（non-recursive query）



- 機能：名前解決の実行
- キャッシュDNSサーバーが各権威DNSサーバーに対し、反復的に実行する
- そのため、非再帰問い合わせは反復問い合わせ（iterative query）とも呼ばれる

おさらい：どちらが再帰問い合わせで どちらが非再帰問い合わせ？



キャッシュDNSサーバーは再帰問い合わせを受信し、
非再帰問い合わせを再帰的に実行する（特に誤解しやすいので注意）

digコマンドによる使い分け

- 再帰問い合わせ
  - 問い合わせ先がキャッシュDNSサーバーである場合に使用
  - digコマンドをオプションを付けずに実行する

  ＜実行例＞`dig www.jprs.jp a @8.8.8.8`

  Google Public DNS
- 非再帰問い合わせ（反復問い合わせ）
  - 問い合わせ先が権威DNSサーバーである場合に使用
  - digコマンドに+norecオプションを付けて実行する

  ＜実行例＞`dig +norec www.jprs.jp a @a.dns.jp`

  JP DNSサーバー

digコマンドの+norecオプションの使い方を覚えて、二種類の問い合わせを適切に使い分けられるようになったら「初心者卒業」

事後資料で新規追加

drillコマンドによる使い分け

- BINDが標準添付されなくなったシステムではdigコマンドに替え、drillコマンドが提供されている場合がある
  - 例：最新版のFreeBSDやArch Linuxなど
  - digコマンドはBINDの一部として開発されている
- 再帰問い合わせ
  - drillコマンドをオプションを付けずに実行する

Google Public DNS

＜実行例＞`drill www.jprs.jp a @8.8.8.8`

- 非再帰問い合わせ（反復問い合わせ）
  - drillコマンドに-o rdオプションを付けて実行する

JP DNSサーバー

＜実行例＞`drill -o rd www.jprs.jp a @a.dns.jp`

デフォルトでは-o RD（RDビットをセット）

事後資料で更新：drillオプションの項目を追加

まとめ：二種類の問い合わせ

	再帰問い合わせ	非再帰問い合わせ （反復問い合わせ）
機能	名前解決を 依頼（要求）する	名前解決を実行する
RDビット	RD=1（セット）	RD=0（クリア）
問い合わせ元	DNSクライアント・ DNSプロキシー	キャッシュDNSサーバー
問い合わせ先	キャッシュDNSサーバー	権威DNSサーバー
実行形態	必要に応じて 能動的に実行	再帰問い合わせを 受けて実行
digオプション	なし（デフォルトで+rec）	+norec
drillオプション	なし（デフォルトで-o RD）	-o rd

キャッシュDNSサーバーと権威DNSサーバーを
兼用していると、理解しづらくなるので注意（次で説明）

注意点④：兼用可能な実装

- キャッシュDNSサーバーと権威DNSサーバーを一つのプログラムで兼用可能な実装（BIND）が存在している
- かつ、BINDはデフォルトで双方の機能が有効
  - そのため、双方の機能が兼用されている場合がある
    - 特に「過去の経緯」を抱えているサーバーに多い
  - 兼用していなくても、双方の機能が有効になって（しまって）いる場合がある

これらを兼用することはさまざまな問題の原因となりうるため、
機能分離と適切な機能制限を強く推奨

機能分離・制限を強く推奨する理由

- DNSの動作・各構成要素の理解促進
  - DNSの理解のためには機能分離が必須と言ってよい
- セキュリティ上のリスクの回避
  - オープンリゾルバーになりやすい
    - 例：権威DNSサーバーにおいて、キャッシュDNSサーバーの機能も有効になってしまっている場合
  - キャッシュポイズニング攻撃を受けやすい
    - 例：オープンリゾルバーになってしまっている場合
  - 脆弱性の影響を受けやすい
    - 例：使っていない機能が有効になっている場合

※オープンリゾルバー：必要なアクセスコントロールが実施されておらず、インターネット上のどこからの名前解決要求であっても実行してしまう状態のDNSサーバー

機能分離・制限を強く推奨する理由（続き）

- 設定・運用コストの軽減・トラブルの防止
  - 兼用していると・・・
    - 障害発生時の原因切り分けが面倒
    - トラブルシューティングが面倒
- 将来の移行
  - BIND以外の多くの実装では機能分離されている
  - 開発元のISCも機能分離・制限を推奨している

今日のまとめ

今日のまとめ①

1. DNSの構成要素とその役割
   - ① DNSクライアント：名前解決を依頼
   - ② DNSプロキシー：名前解決の依頼・応答を中継
   - ③ キャッシュDNSサーバー：名前解決を実行
   - ④ 権威DNSサーバー：実データ（ゾーン）を管理

これらの構成要素とその役割を把握することが
DNSの理解への第一歩

事後資料で更新：drillの記述を追加

今日のまとめ②

2. DNSを勉強するにあたっての注意点
（初学者が特にはまりやすい4つのポイント）

① 名称の不統一

名前に気を遣って一歩前進

② 二種類のDNSサーバー

どちらを示しているのかを常に明確に

③ 二種類の問い合わせ

digの+norec／drillの-o rdを適切に使えれば初心者卒業

④ 兼用可能な実装（BIND）

DNSの理解のためにもキャッシュと権威は分け、
それぞれの機能のみを有効にすべし

Q & A