初心者のためのDNS運用入門
- トラブルとその解決のポイント -

2013年7月19日

DNS Summer Days 2013

株式会社日本レジストリサービス（JPRS）

水野 貴史

講師自己紹介

- 氏名：水野 貴史（みずの たかふみ）
- 生年月日：1988年3月3日（25歳）
- 所属：株式会社日本レジストリサービス（JPRS） システム部
- Unix歴：8年目（FreeBSD、OS Xを中心に）
- 職歴：
  - 2013年4月 JPRS入社
  - 2013年4月～6月 新人研修
  - 2013年7月

  → DNS Summer Days 2013講師（←**New!**）

本セミナーの概要と対象

- ツールの紹介と使い方
  - コマンドラインツールとWebサービス

  → トラブルシューティングについて、具体例を挙げながら解説

- 対象
  - DNSサーバーをこれから運用される方
  - DNSサーバーの運用を始めて間もない初学技術者の方

そして、

- 初学技術者ではない方々の知識のおさらい
- 社内セミナーの資料などに活用できる内容

としても活用可能なものをめざします

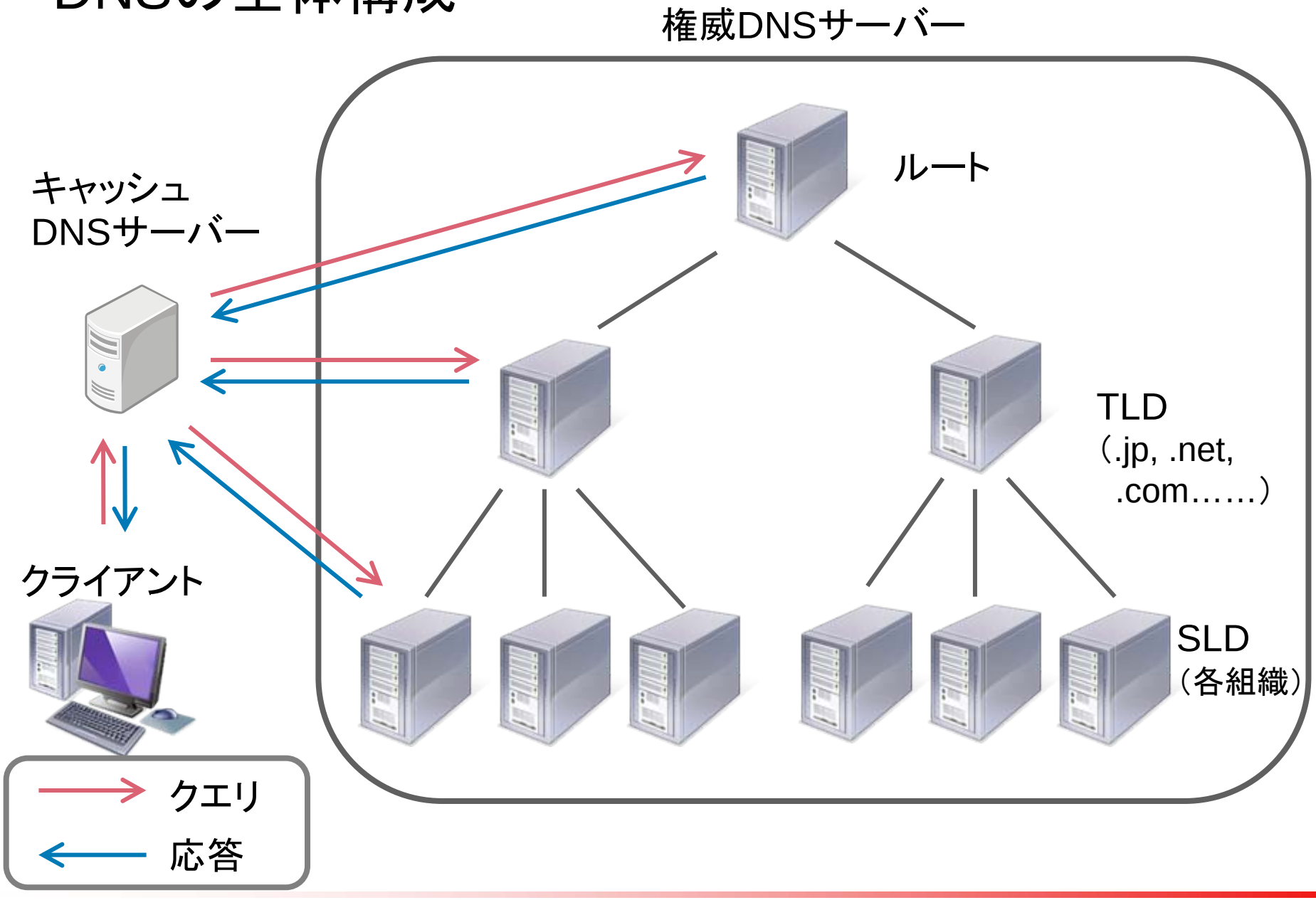
本日の内容

- DNSの基礎知識とトラブルシューティングの基本（おさらい）
  - DNSの全体構成
  - 区別すべき2種類の問い合わせ
  - トラブルシューティングの基本
- 道具の使い方
  - コマンドラインツールの使い方
  - Webサービスの紹介
- よくあるトラブル事例とトラブルシューティング
  - 設定がうまくいかない
  - 名前が引けない
  - 名前を引くのに時間がかかる

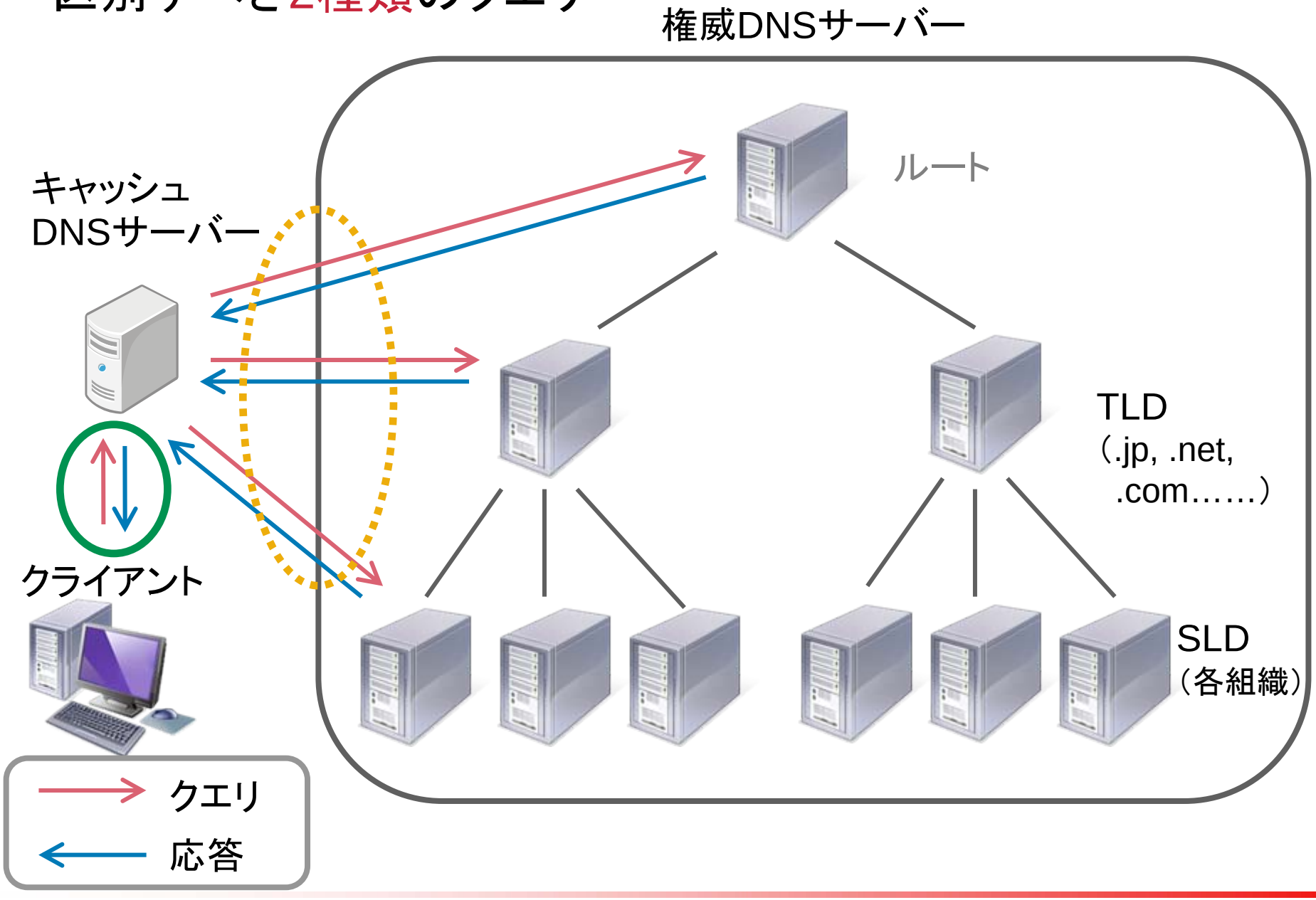
まずは、おさらいとして……

1. DNSの基礎知識と トラブルシューティングの基本

DNSの全体構成



区別すべき2種類のクエリ

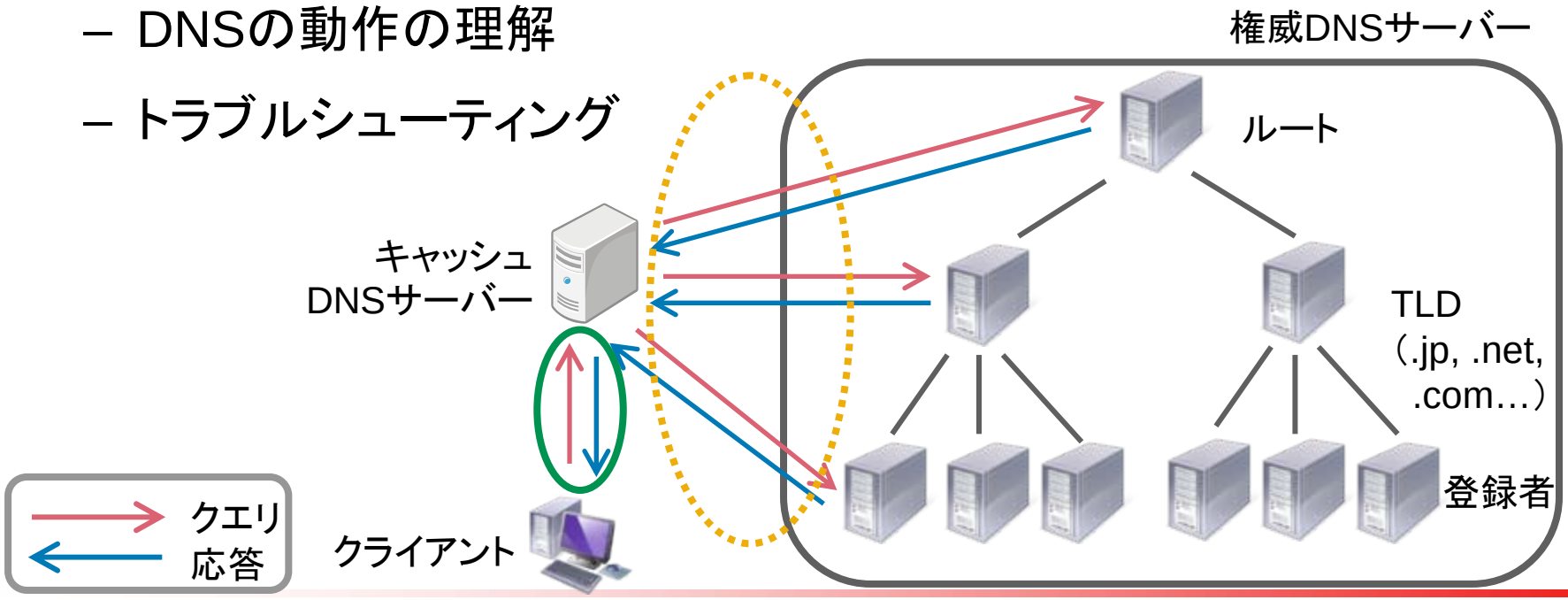


区別すべき2種類のクエリ

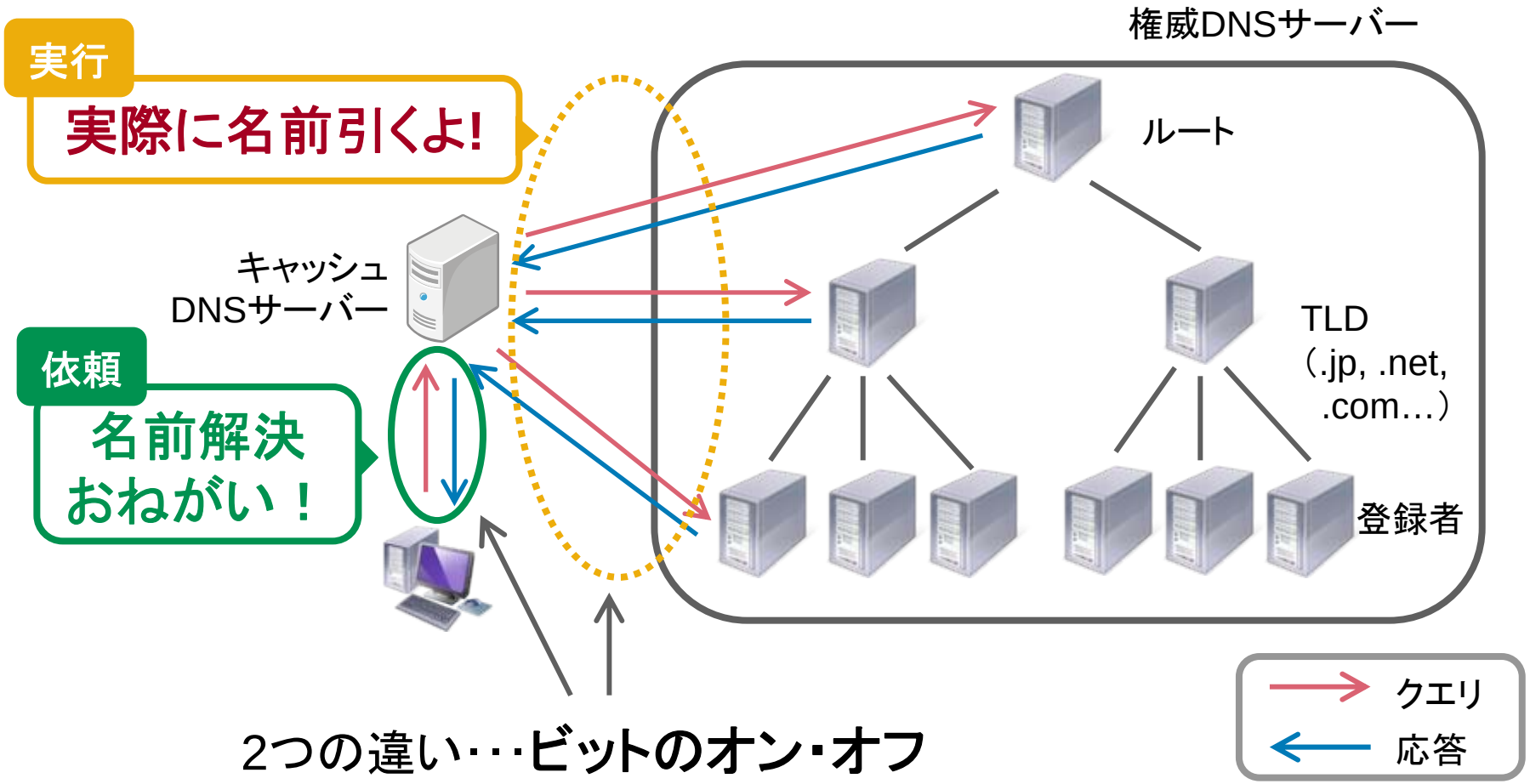
- クライアントからキャッシュDNSサーバーへのクエリ
- キャッシュDNSサーバーから権威DNSサーバーへのクエリ

→ この2種類のクエリを明確に区別することがすべての基本

- DNSの動作の理解
- トラブルシューティング

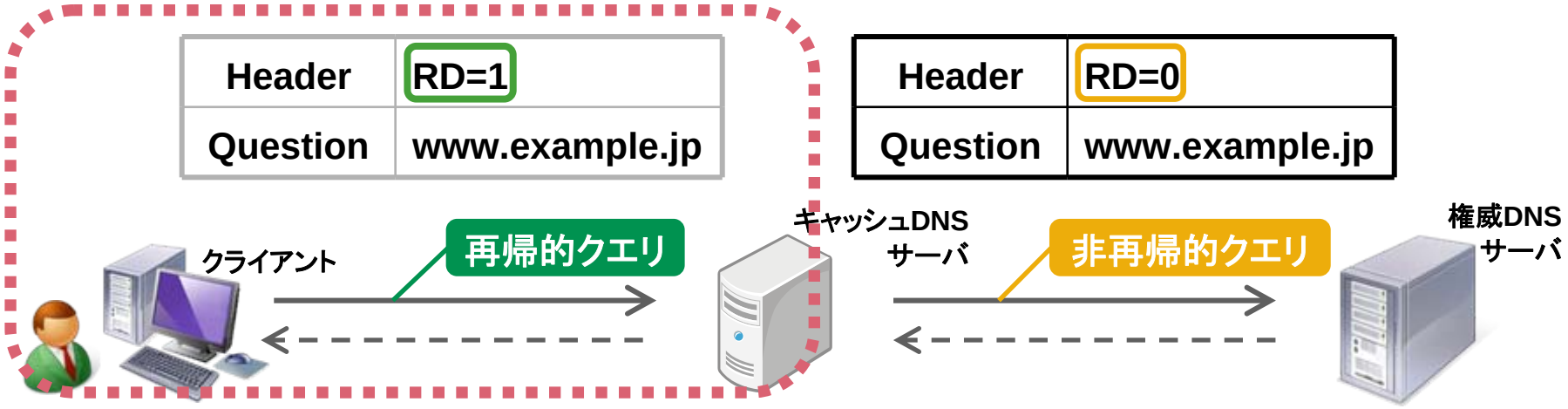


区別すべき2種類のクエリ



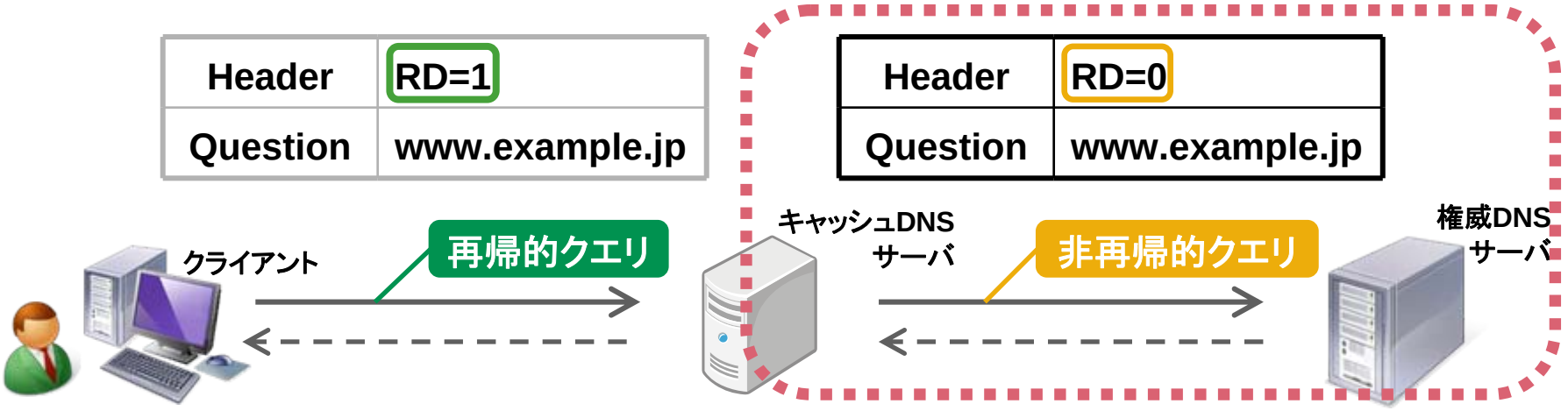
- ✓ 区別しないと、調査の際、問題の切り分けができない
  - ✓どの部分が問題か？どの部分を調べているのか？

再帰的クエリ（recursive query）



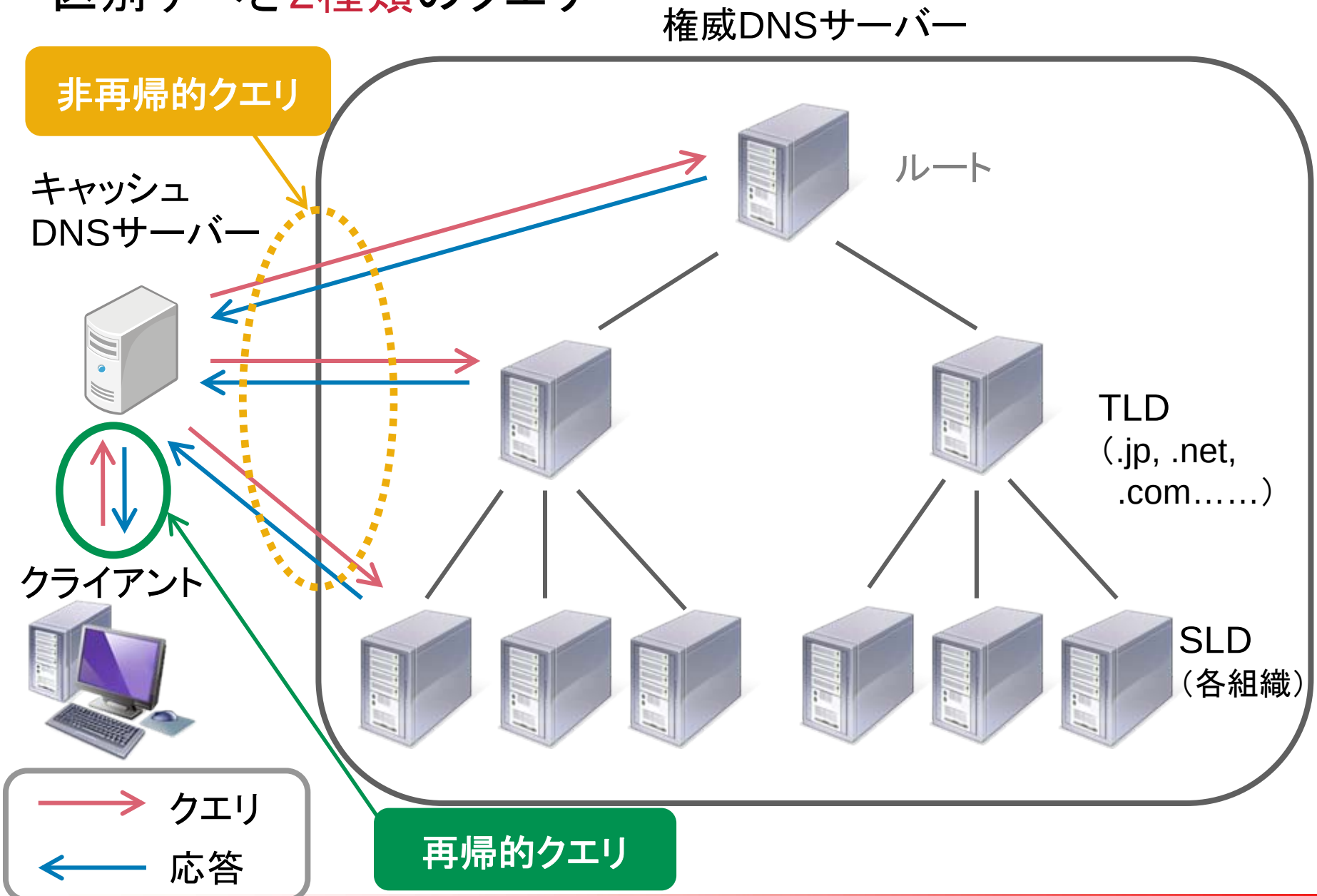
- クライアントからキャッシュDNSサーバーへのクエリ
- クエリ中のRDビットがセットされている
- クライアントはRDビットをセットしたクエリを送信することにより、キャッシュDNSサーバーに階層構造をたどらせる
  - これを名前解決要求という

非再帰的クエリ（non-recursive query）



- キャッシュDNSサーバーから権威DNSサーバーへのクエリ
  - クエリ中のRDビットがセット**されていない**
- クライアントからの名前解決要求によって発生
- 再帰的クエリと**同じ内容**をRDビットをクリアしたうえで送信

区別すべき2種類のクエリ



トラブルシューティングの基本

- **Where?** - 原因はどこか？
  - 手元のキャッシュDNSサーバーか？
  - 権威DNSサーバーのいずれかか？
  - 各サーバーまでのネットワークか？
- **How?** - どこをどう調べればよいか？
  - どんなツールやWebサービスを使えばよいか？
- 調査の際には「再帰的クエリ」と「非再帰的クエリ」を明確に区別すべき
  - 調査対象がキャッシュDNSサーバーか？権威DNSサーバーか？
- それぞれのサーバーに合った形での調査が必要
  - digコマンドのオプションなど → 以降で詳しく説明します

トラブル解決に役立つ

2. 道具の使い方

調査の基本—どのコマンドを使うべきか？

- DNSサーバにリクエストを送り、調査する
  - リクエストに関するパラメータを細かく調整して、応答を調査する
  - 基本はコマンドラインツール
- nslookup コマンド……は使うべきでない
  - クエリの細かいパラメータが指定不可
  - 応答のフラグやセクションの情報を得ることができない
- では、何を使うか？
  - digコマンド、drillコマンド

digコマンドとdrillコマンド

こちら

- dig コマンド
  - BIND 9 に付属するコマンド
  - コマンド例：
    - `$ dig␣+dnssec␣@192.0.2.53␣example.jp.␣SOA`
- drill コマンド
  - Unboundで用いられているライブラリ「ldns」に付属するコマンド
  - コマンド例：
    - `$ drill␣-D␣example.jp.␣@192.0.2.53␣SOA`

今日はdigコマンドを用いた解説をします

- nslookup

```
$ nslookup jprs.co.jp
Server:         192.0.2.12
Address:        192.0.2.12 #53

Non-authoritative answer:
Name:   jprs.co.jp
Address: 202.11.16.167
```

- dig

```
$ dig jprs.co.jp

; <<>> DiG 9.9.2-P2 <<>> jprs.co.jp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERO5:;;jprs.co.jp
```

dig コマンドが使える環境

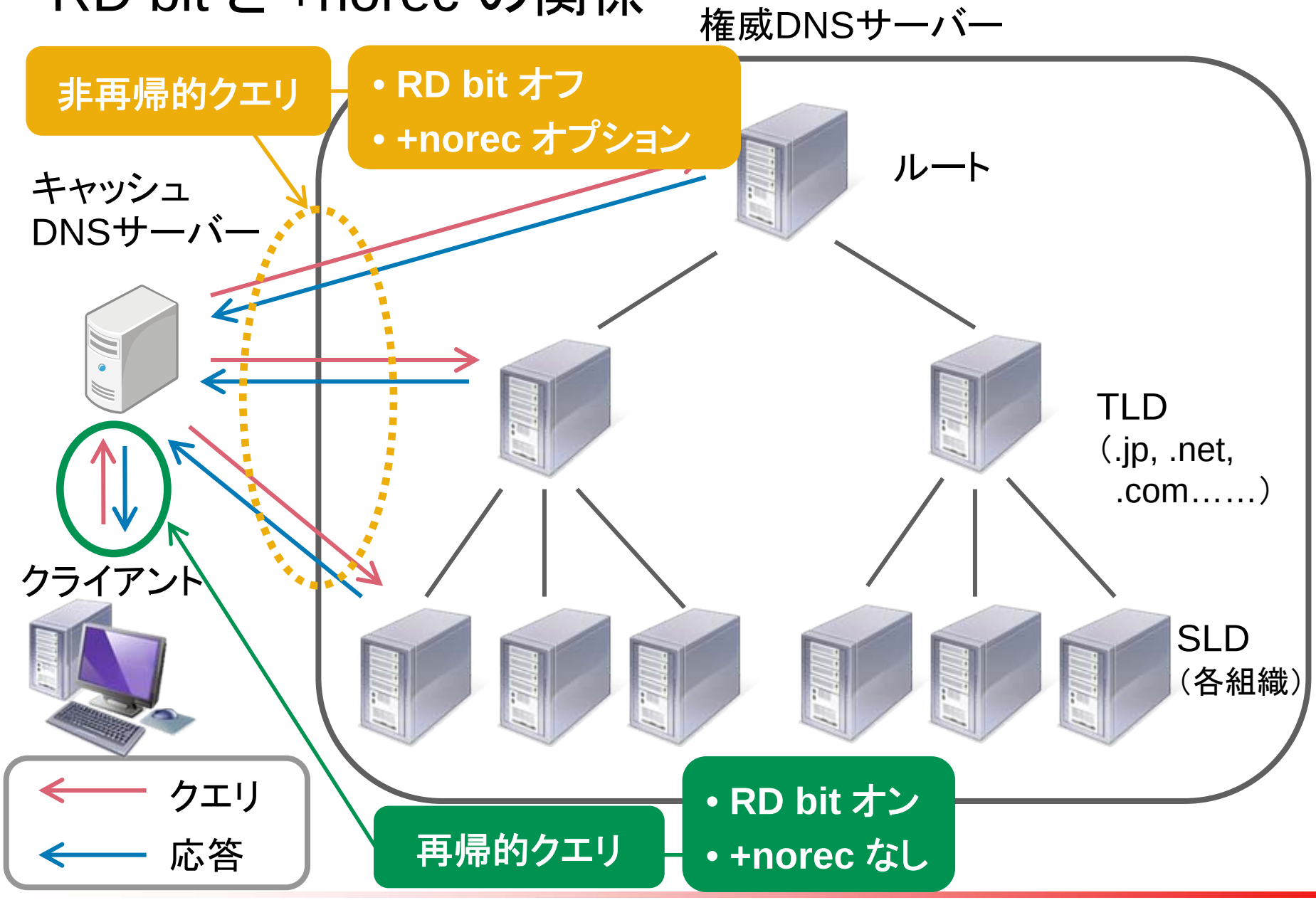
- Unix系OS
  - ほとんどの環境で標準添付
  - OS Xにも標準添付
- Windows
  - Windows版BIND 9のバイナリキットに含まれている
  - 開発元のISCが無償で公開

dig コマンド – 使い方

```
$ dig␣+dnssec␣@192.0.2.53␣example.jp.␣SOA
```

- 重要なオプション
  - RD bit
    - オン　= 階層構造をたどって　　= +recurse　　または +rec
    - オフ　= 持ってる情報を教えて　= +norecurse　または +norec
- RD bit = Recursion Desired bit
  - サーバーに対して「 DNSの階層構造をたどって！」と伝えるために、クライアント側でセット
  - digコマンドやdrillコマンドでは**デフォルトでオン**
  - 権威DNSサーバーに対してリクエストを送信する場合には、オフにしておくこと

RD bit と +norec の関係



dig コマンド – 使い方

```
$ dig␣+dnssec␣@192.0.2.53␣example.jp.␣SOA
```

- DNSSEC関連オプション
  - DO bit: +dnssec（オン）　+nodnssec（オフ）
- DO bit = DNSSEC Ok bit
  - クライアントが設定するbit
  - 「こちらは DNSSEC 関連のレコードを受信する準備がある」ことを通知

dig コマンド – 出力の読み方 (1/7)

```
$ dig +norec @ns1.jprs.jp jprs.jp

; <<>> DiG 9.9.2-P2 <<>> +norec @ns1.jprs.jp jprs.jp
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34174
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 5

;; QUESTION SECTION:
;jprs.jp.                       IN      A

;; ANSWER SECTION:
jprs.jp.                86400   IN      A       202.11.16.167

;; AUTHORITY SECTION:
jprs.jp.                86400   IN      NS      ns2.jprs.jp.
jprs.jp.                86400   IN      NS      ns3.jprs.jp.
jprs.jp.                86400   IN      NS      ns1.jprs.jp.

;; ADDITIONAL SECTION:
ns1.jprs.jp.            86400   IN      A       202.11.16.49
ns1.jprs.jp.            86400   IN      AAAA    2001:df0:8::a153
ns2.jprs.jp.            86400   IN      A       202.11.16.59
ns2.jprs.jp.            86400   IN      AAAA    2001:df0:8::a253
ns3.jprs.jp.            86400   IN      A       61.200.83.204

;; Query time: 1 msec
;; SERVER: 203.0.113.12#53(203.0.113.12)
;; WHEN: Thu May 02 15:20:20 2013
;; MSG SIZE  rcvd: 199
```

特に注目

ヘッダー

Question

Answer

Authority

Additional

応答時間・サイズなど

dig コマンド – 出力の読み方 (2/7)

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34174
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 5
```

- ヘッダの内容
  - 各セクションに関する情報やステータス、フラグなどを格納
- 主な status（RCODE: 応答コード）
  - NOERROR　正常な応答（該当するタイプがない場合も含む）
  - FORMERR　DNSメッセージのフォーマットが不正
  - SERVFAIL　DNSサーバーの異常
  - NXDOMAIN　リクエストされた名前が存在しない
  - REFUSED　リクエストが拒否された

dig コマンド – 出力の読み方 (3/7)

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34174
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 5
```

- 注目すべき主な flags（ヘッダ等に含まれるビット）
  - qr: 応答であることを示す（Query / Response)
    - リクエストではオフになっている
  - aa: 権威ある応答であることを示す(Authoritative Answer)
    - 通常、問い合わせたゾーンの権威DNSサーバーからの応答はオン
    - 他のDNSサーバーに委任していることを示す応答ではオフ
  - ra: 再帰検索要求が処理可能なことを示す（Recursion Available)
    - 通常、キャッシュDNSサーバーからの応答ではオン
  - tc: 応答の一部が切り捨てられたことを示す（TrunCation）
    - TCPに切り替えて（TCPフォールバック）再度問い合わせる
    - digコマンドは動的にTCPフォールバックする（「+ignore オプション」で抑制できる）

dig コマンド – 出力の読み方 (4/7)

```
;; QUESTION SECTION:
;jprs.jp.                        IN      A
```

- Question セクションの内容
  - 問い合わせた内容がそのままコピーされている

```
$ dig +norec @ns1.jprs.jp jprs.jp
```

dig コマンド – 出力の読み方 (5/7)

```
;; ANSWER SECTION:
jprs.jp.                86400   IN      A       202.11.16.167
```

- Answerセクション
  - 問い合わせた内容に対応するリソースレコード（RR）が格納される
  - 問い合わせた名前やタイプが存在しない場合や、
    他のDNSサーバーにゾーンが委任されている場合は空

dig コマンド – 出力の読み方 (6/7)

```
;; AUTHORITY SECTION:
jprs.jp.                86400   IN      NS      ns2.jprs.jp.
jprs.jp.                86400   IN      NS      ns3.jprs.jp.
jprs.jp.                86400   IN      NS      ns1.jprs.jp.
```

- Authorityセクション
  - 権威を持っているDNSサーバーの情報を格納
  - 問い合わせたタイプが存在しないことを示す場合、SOA RRが格納される

dig コマンド – 出力の読み方 (7/7)

```
;; ADDITIONAL SECTION:
ns1.jprs.jp.            86400   IN      A       202.11.16.49
ns1.jprs.jp.            86400   IN      AAAA    2001:df0:8::a153
ns2.jprs.jp.            86400   IN      A       202.11.16.59
ns2.jprs.jp.            86400   IN      AAAA    2001:df0:8::a253
ns3.jprs.jp.            86400   IN      A       61.200.83.204
```

- Additionalセクション
  - 付加的な情報が格納される

→Authorityセクションに含まれるDNSサーバーのA、AAAA RRなど

調査に使えるWebサービス

- DNSの設定などを、GUIで可視化・チェック可能

  ここでは2種類のツールを紹介します（この他にもあります）

→ DNSViz

  – DNSSECの可視化ツール

→ dnscheck.jp

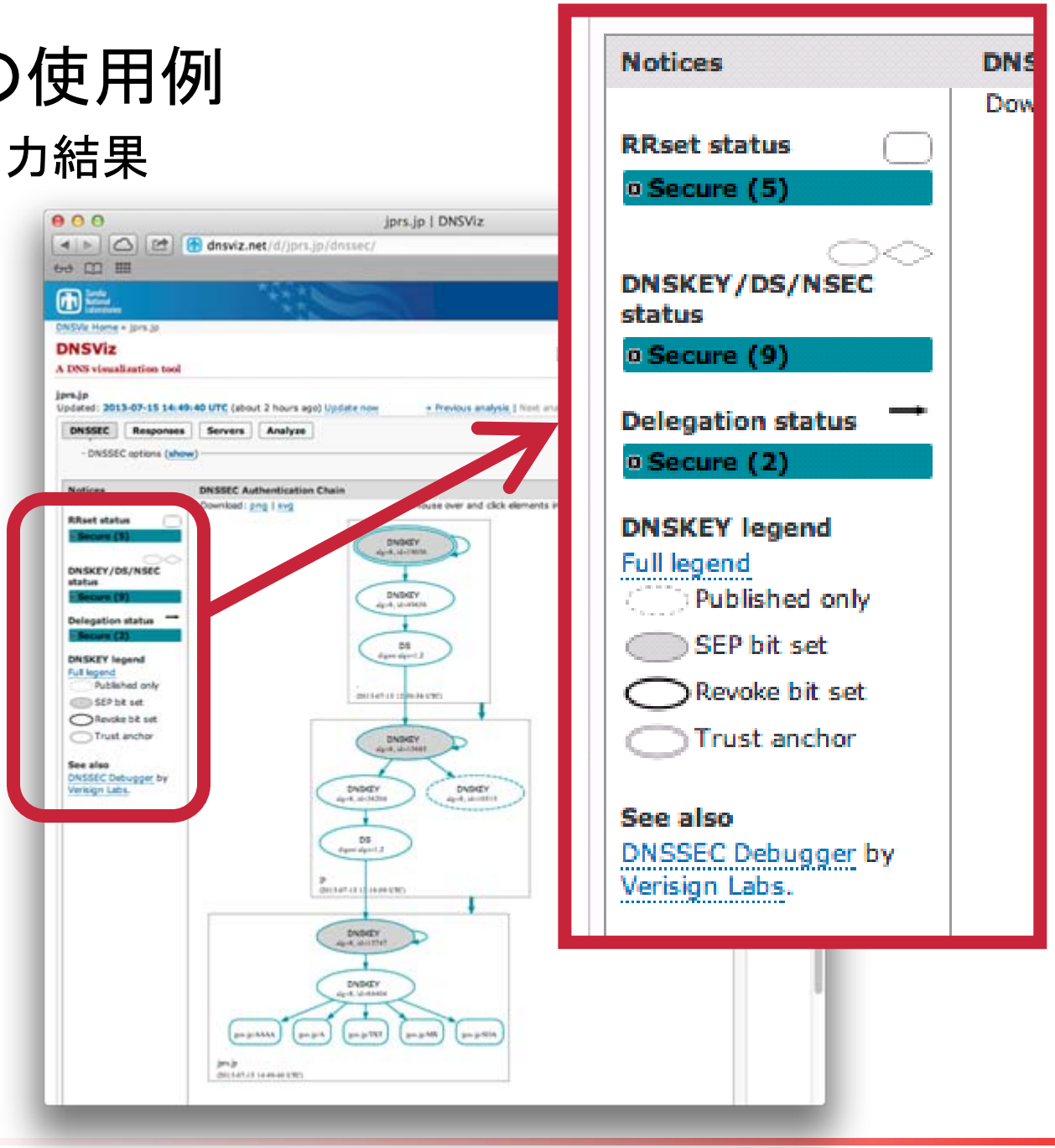
  – DNSの設定チェックツール（JPRS提供）

  – 今現在の設定の確認

  – これからしようと思っている設定

DNSVizの使用例

「jprs.jp」の出力結果



dnscheck.jpの使用例

「jprs.jp」の出力結果

■チェック結果詳細

1.ドメイン名に対するチェック結果

値	重要度	チェック結果
JPRS.JP	OK	

2.各ホスト名に対するチェック結果

値	重要度	チェック結果
ns1.jprs.jp	OK	
202.11.16.49		
ns1.jprs.co.jp.		

困った！どうしてこうなる？

3. DNSトラブル事例

今日紹介するトラブル事例

A) 名前が引けない
   1. DNSサーバーがダウンしている
   2. CNAMEの循環

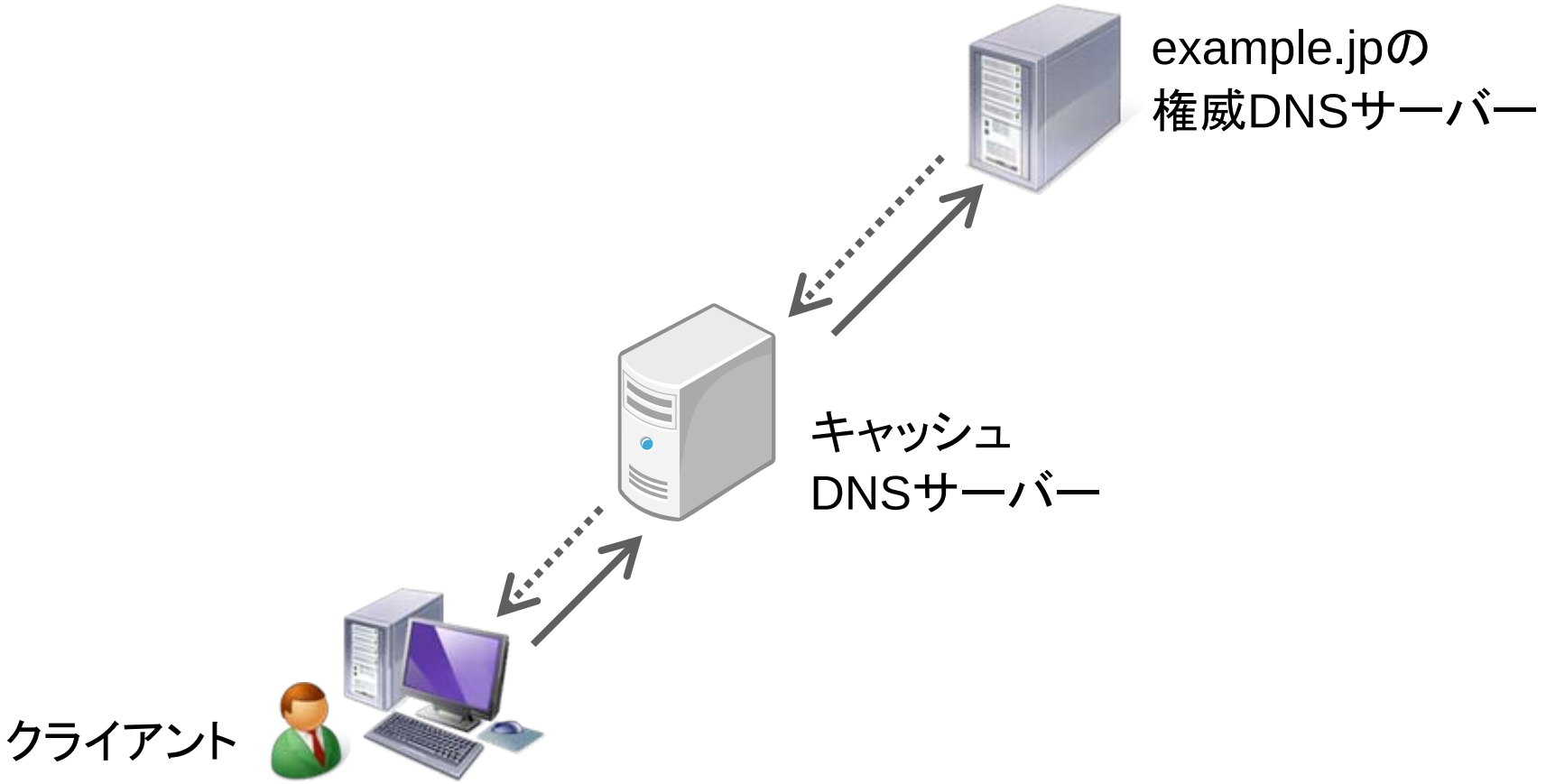
B) 名前を引くのに時間が掛かる
   1. TCPフォールバック
   2. 権威DNSサーバーの一部がダウンしている

C) 設定を間違えた
   1. ゾーン転送がうまくいかない
      1. マスタサーバーにDNSが稼動していない
      2. マスタサーバー側のファイヤーウォールでブロックされている場合
      3. マスタサーバー側でゾーン転送が許可されていない場合
   2. ピリオドを忘れた

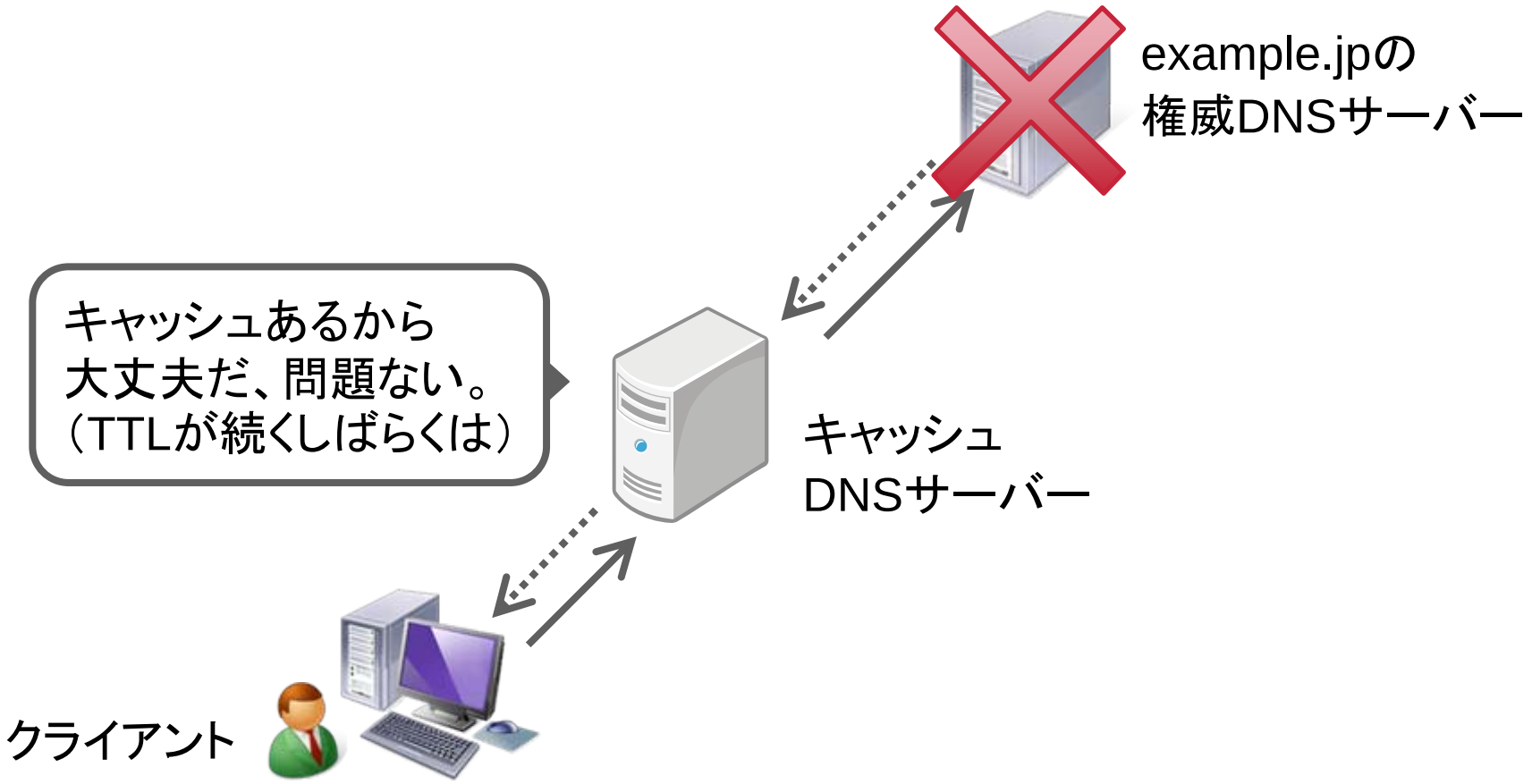
DNSトラブル事例

- A.名前が引けない

1. DNSサーバーがダウンしている

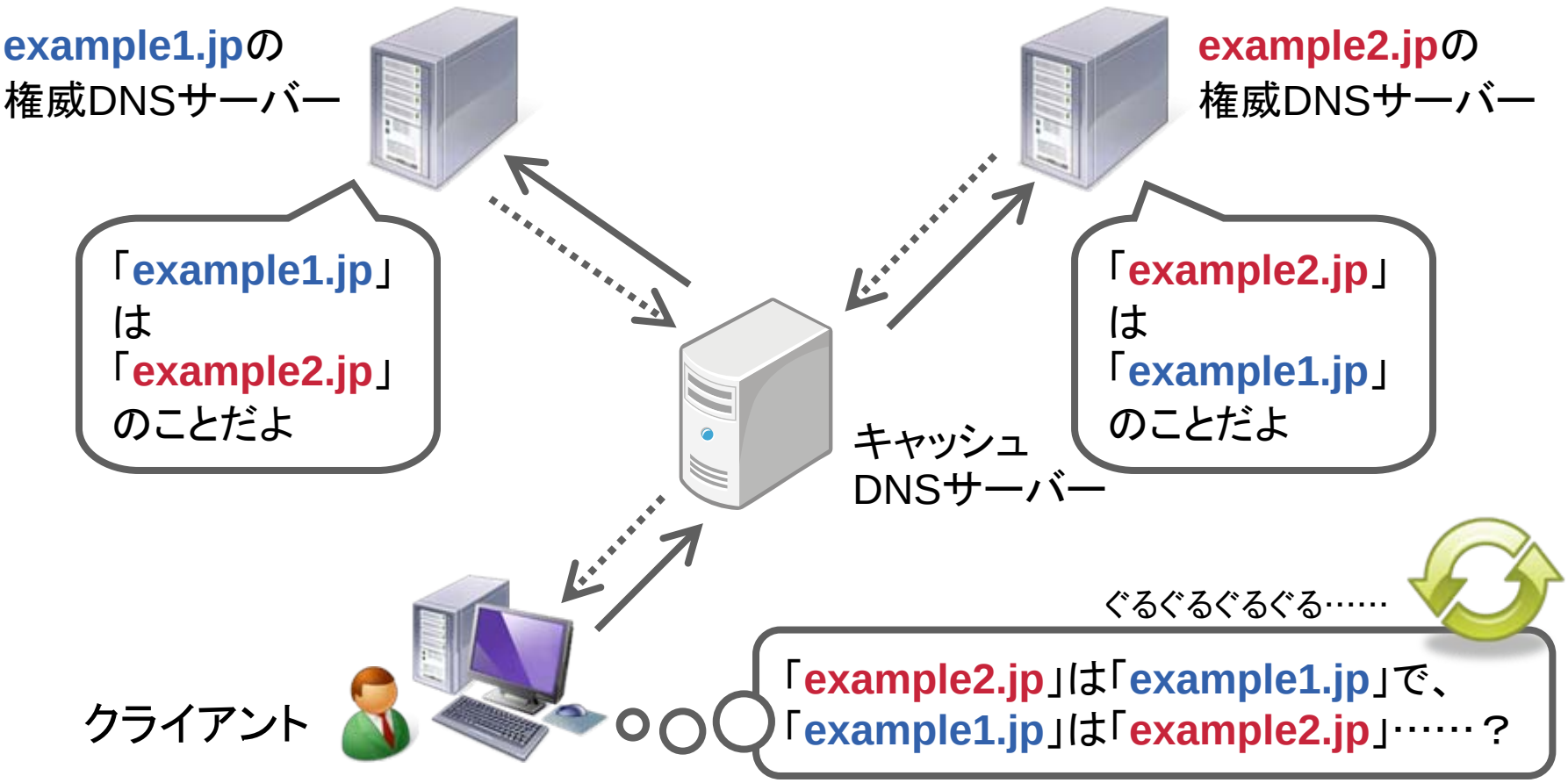


1. DNSサーバーがダウンしている



キャッシュDNSサーバのキャッシュで、気づくのが遅れることも……

2. CNAME の循環



アプリケーションによってはエラーが出たり、そのまま固まったり……

2. CNAME の循環 - dig の実行結果

```
$ dig cname.a.example. @127.0.0.1

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> cname.a.example. @127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20338
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;cname.a.example.               IN      A

;; ANSWER SECTION:
cname.a.example.        15      IN      CNAME   cname.b.example.
cname.b.example.        15      IN      CNAME   cname.a.example.

;; Query time: 15 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Jul 18 20:40:32 2013
;; MSG SIZE  rcvd: 69
```

DNSトラブル事例

- B.名前を引くのに時間が掛かる

1. TCPフォールバック

- DNS の 512byte の壁
  - 応答はできるだけ 512byte 以下に収め、UDP 一発で送信できるのがよい
  - 近頃のトレンド：応答サイズの増大
    - IPv6、DNSSEC、spam対策（SPF情報：TXTレコード）

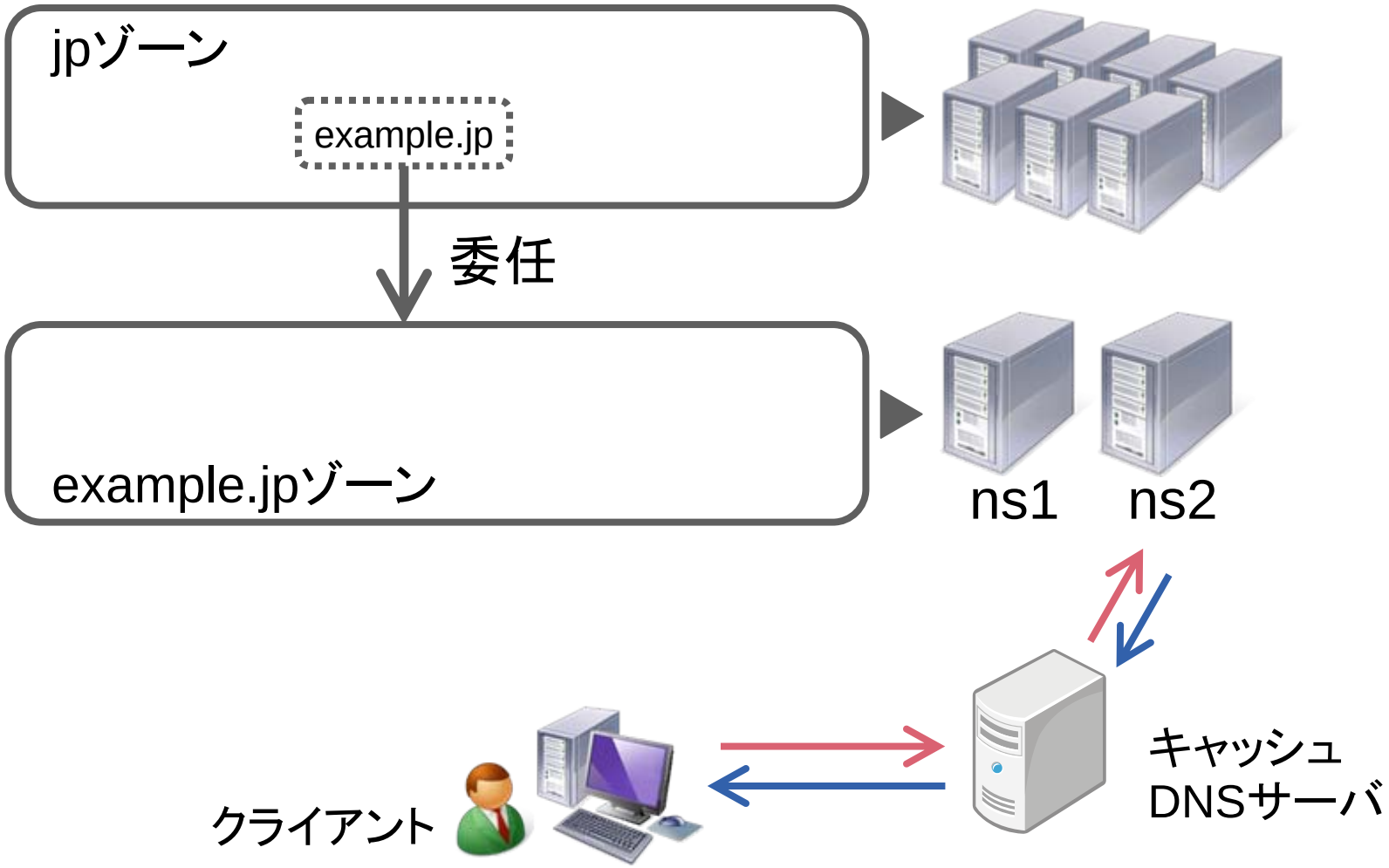
- どうなる？
  - 最初に UDP で問い合わせて、512byte に収まらないことが分かったら TCP で再度問い合わせる
    - udp での問い合わせで tc ビットがオンになっている

→再問い合わせの分遅くなる

  - 最近は「EDNS0」という仕組みが使われる
    - 本資料では省略

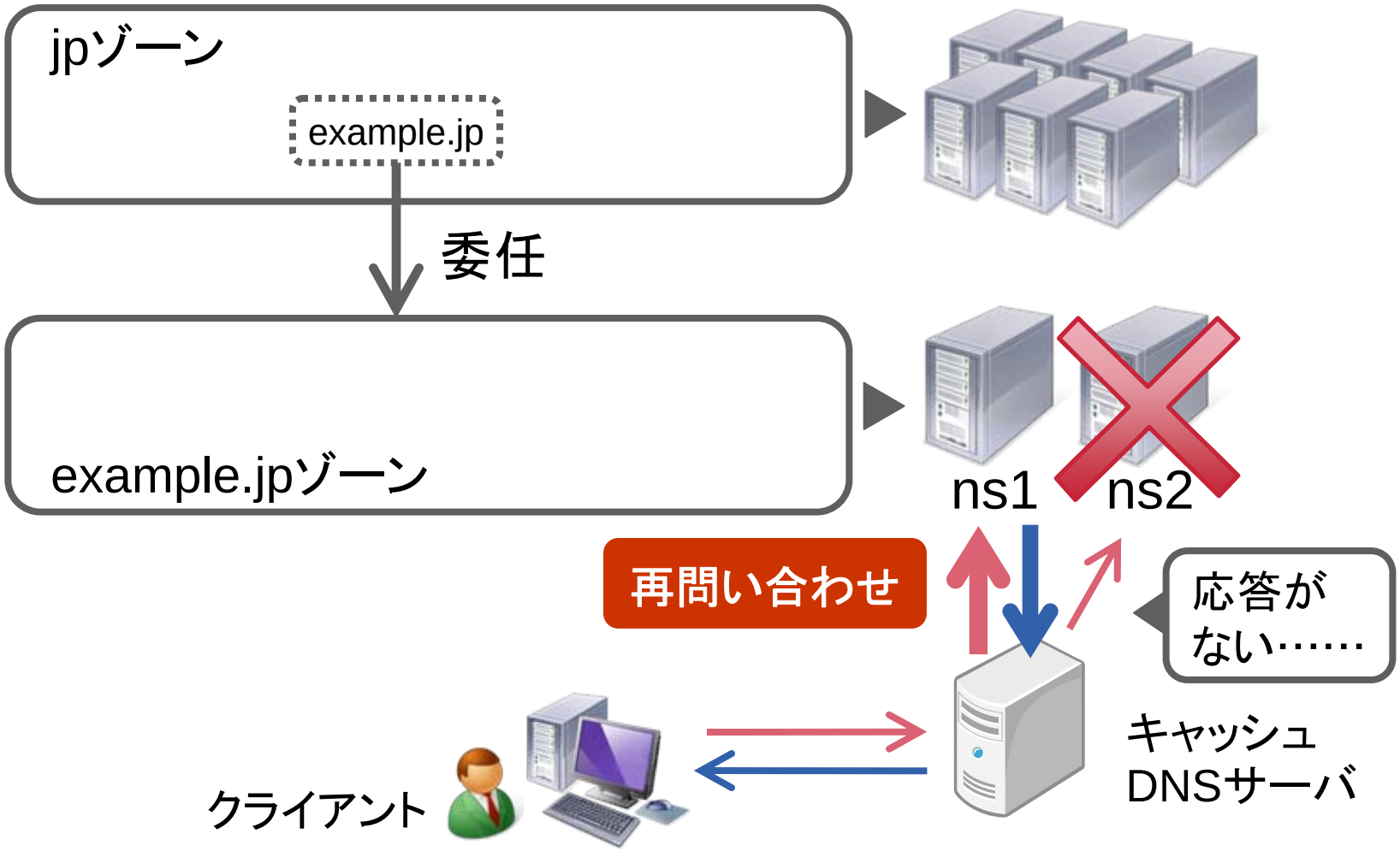
2. 権威DNSサーバーの一部がダウンしている (1/2)

✓ 通常の場合



2. 権威DNSサーバーの一部がダウンしている (2/2)

✓DNSサーバーの一部がダウンしている場合



2. 権威DNSサーバーの一部がダウンしている (2/2)

✓DNSサーバーの一部がダウンしている場合

- キャッシュサーバに一度キャッシュされてしまえば、遅延は発生しない
  - 遅延が発生するのは、キャッシュされていないときの問い合わせ
- 今回の例の場合、ns1にいきなり問い合わせに行ったら、遅延は発生しない
  - 権威 DNS サーバーの選択に、プライマリやセカンダリという概念は**ない**
  - どの権威DNSサーバーに問い合わせに行くかは、**ロシアンルーレット**のようなもの

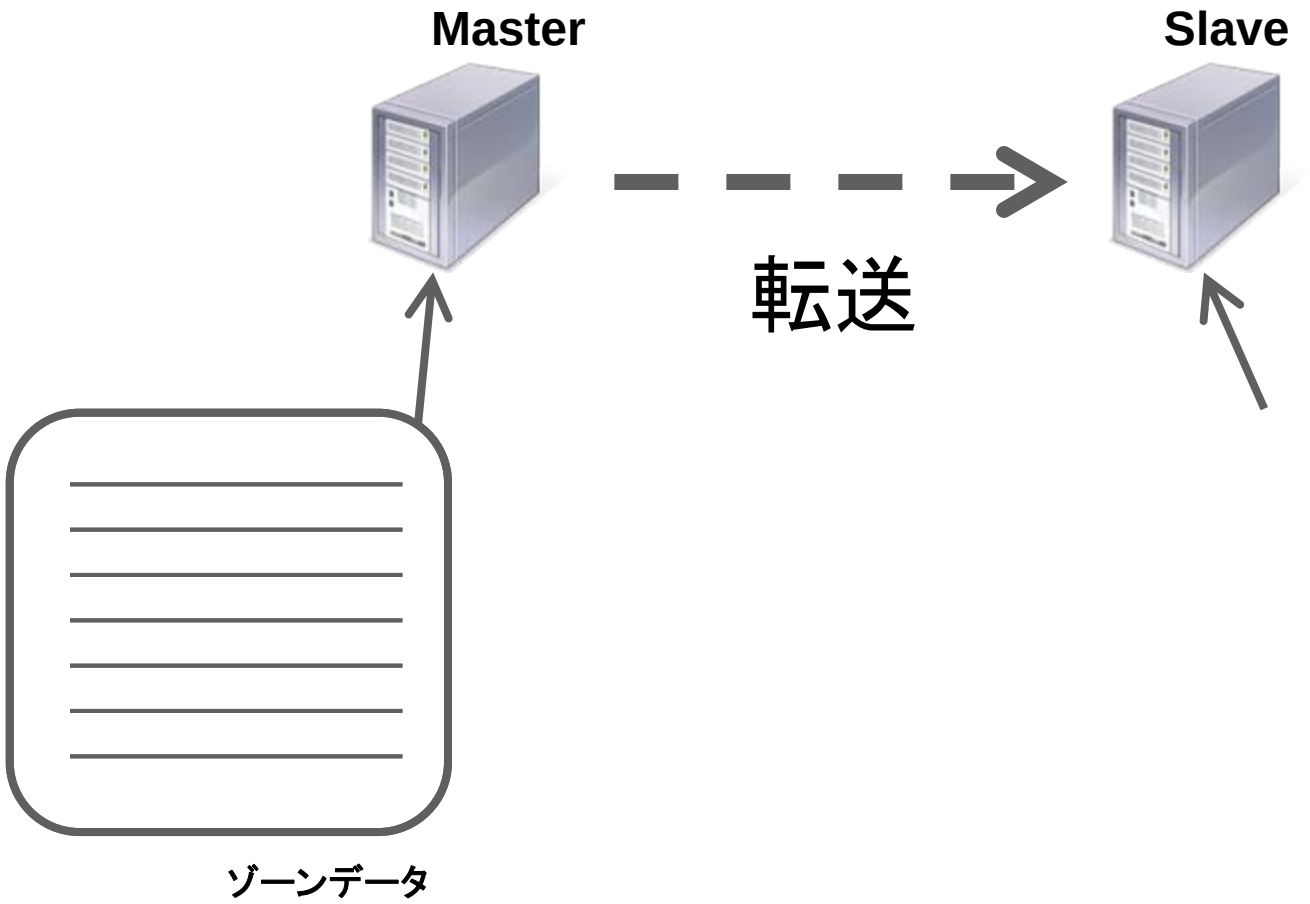
気づくのが遅れることも……

DNSトラブル事例

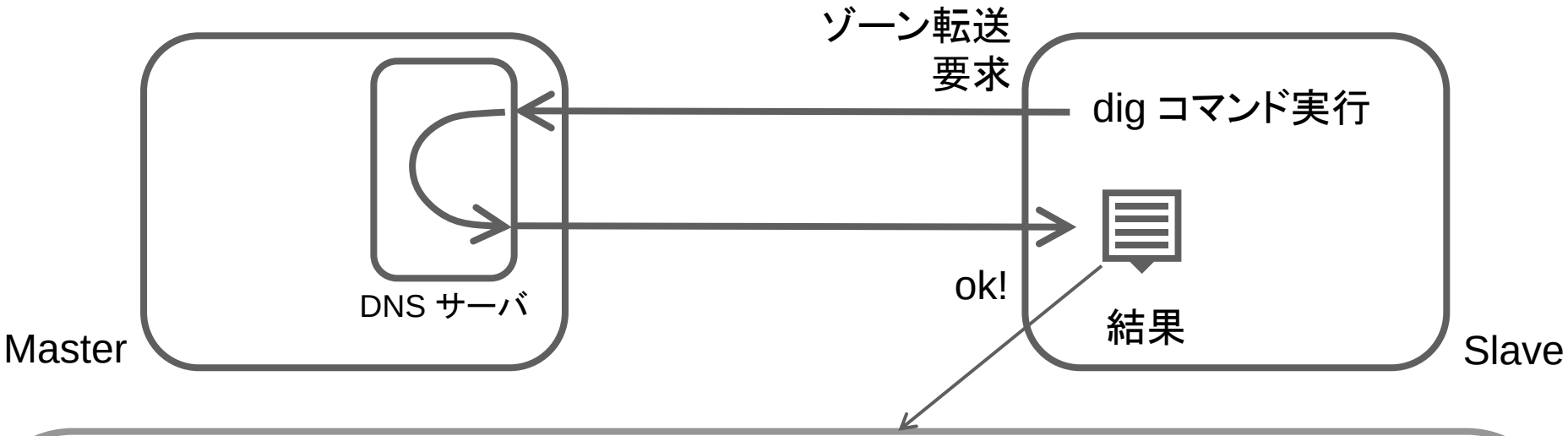
- C. 設定を間違えた

1. ゾーン転送がうまくいかない

- ゾーン転送とは……



1. ゾーン転送がうまくいかない – 正常な例




```
$ dig +norec @(マスタ) example.jp. AXFR

; <<>> DiG 9.8.1-P1 <<>> +norec @(マスタ) example.jp. AXFR
; (1 server found)
;; global options: +cmd
example.jp.             10800   IN      SOA   (中略)
example.jp.             10800   IN      NS      ns1.example.jp.
(中略)
example.jp.             10800   IN      SOA     ns1.example.jp. root.example.jp. (中略)
;; Query time: 1 msec
;; SERVER: (マスタ)#53((マスタ))
;; WHEN: Fri Jul 12 17:56:17 2013
;; XFR size: 31 records (messages 1, bytes 3380)
```

1. ゾーン転送がうまくいかない – よくある原因

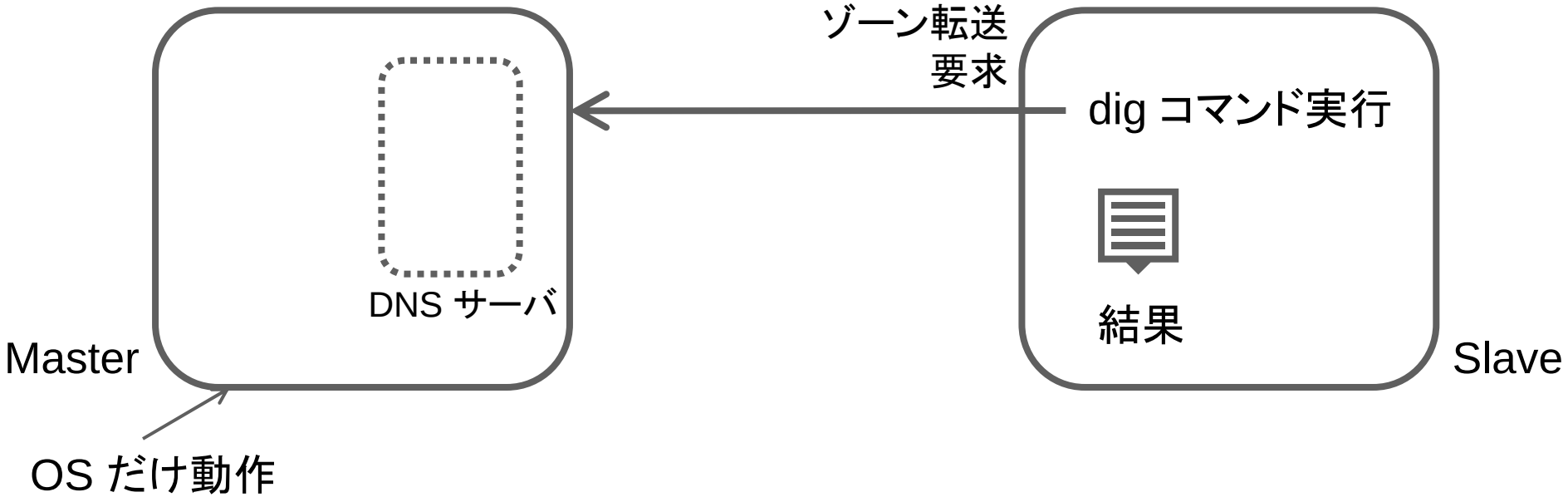
- 原因
  - TCP 53番ポートがフィルタされている？
  - ゾーン転送の設定を間違っている？
  - あるいは他の何か？

どう切り分ける……？

- 調査法
  - dig コマンドを使う
  - コマンド例
    - `$ dig␣+norec␣@(マスタ)␣example.jp␣axfr`

1. ゾーン転送がうまくいかない – 調査と具体例

1. マスタサーバーでDNSが稼動していない場合

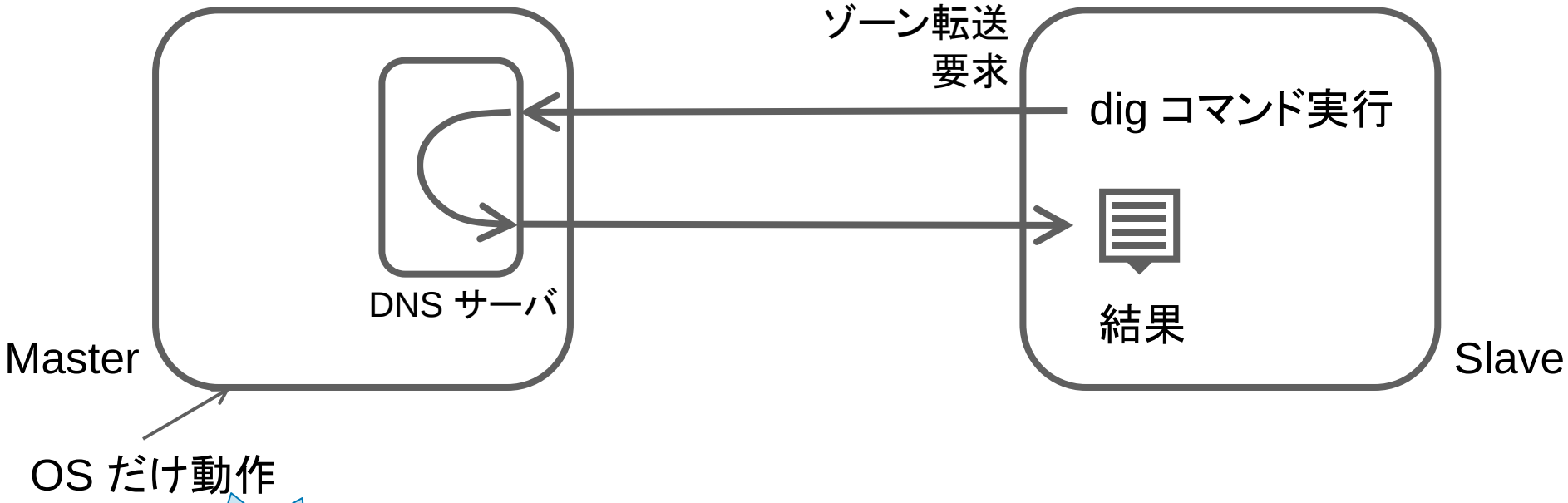


実行結果例

```
$ dig +norec @(マスタ) example.jp axfr
;; Connection to 203.0.113.8 #53(203.0.113.8)
        for example.jp failed: connection refused.
```

1. ゾーン転送がうまくいかない – 調査と具体例

1. マスタサーバーでDNSが稼動していない場合



対応

DNS サーバを立ち上げ直す

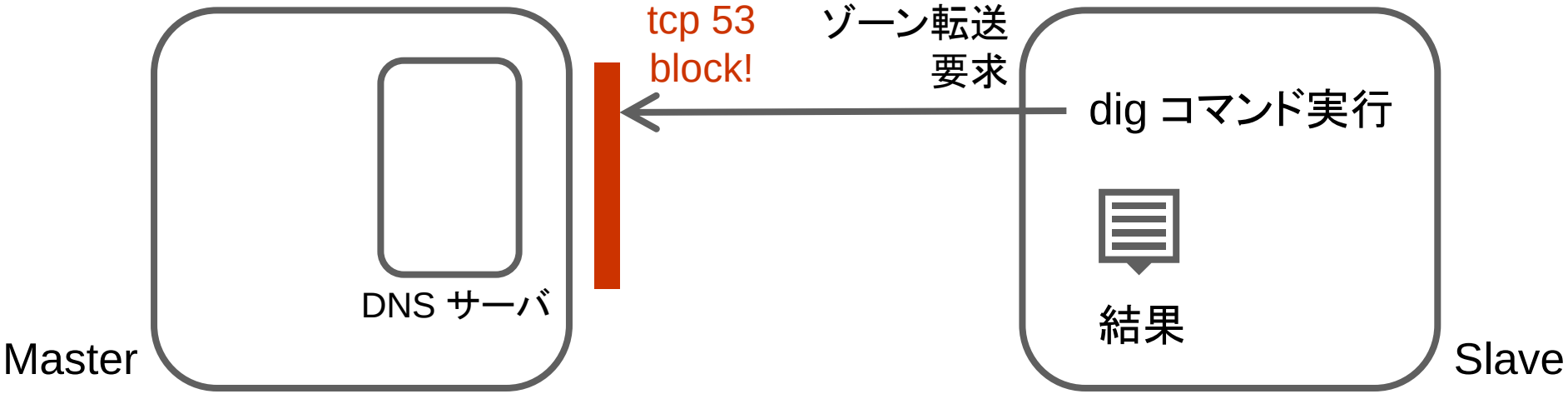
→ 実は気づかないうちに落ちていたのかも……

→ **必ず**原因究明を並行してすすめること

→ サーバーのログのチェックなど……

1. ゾーン転送がうまくいかない – 調査と具体例

2.マスタサーバー側のファイヤーウォールでブロックされている場合



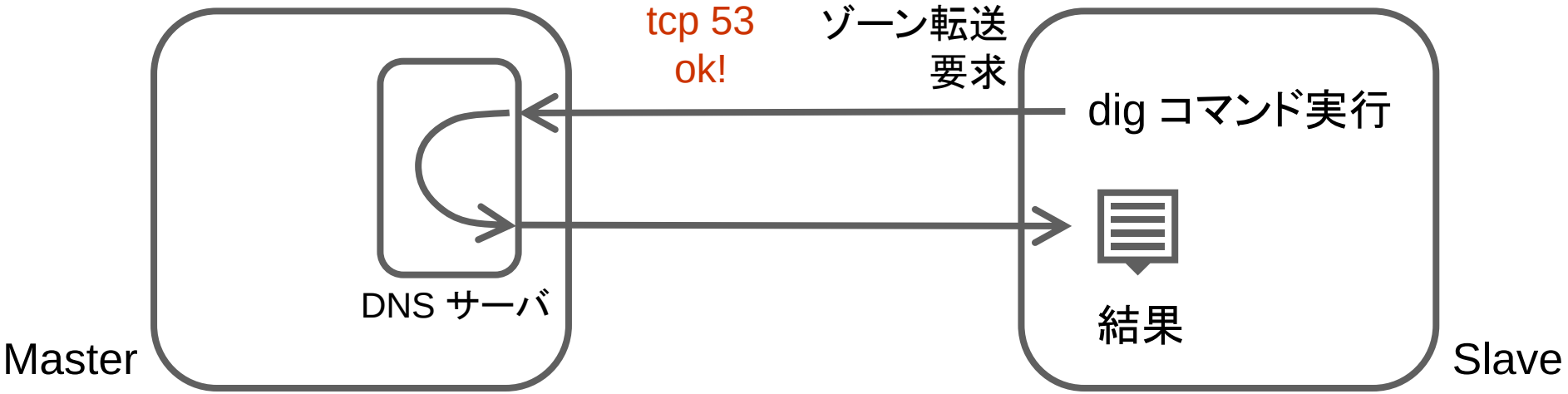
実行結果例

```
$ dig +norec @(マスタ) example.jp axfr

; <<>> DiG 9.9.2-P2 <<>> +norec @203.119.1.1 jprs.co.jp axfr
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

1. ゾーン転送がうまくいかない – 調査と具体例

2.マスタサーバー側のファイヤーウォールでブロックされている場合



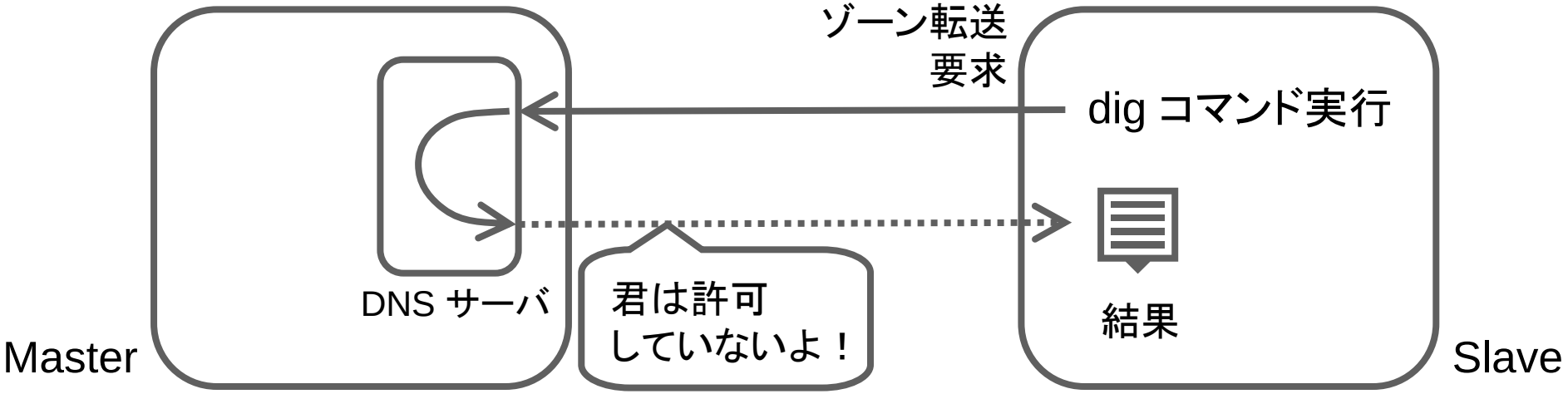
実行結果例

対応

- ✓ TCP 53番ポートを許可する
  - → UDP だけの許可かも……？
  - → そもそも**DNSサーバーでは TCP 53番のオープンが必要！**

1. ゾーン転送がうまくいかない – 調査と具体例

3. マスタサーバー側でゾーン転送が許可されていない場合



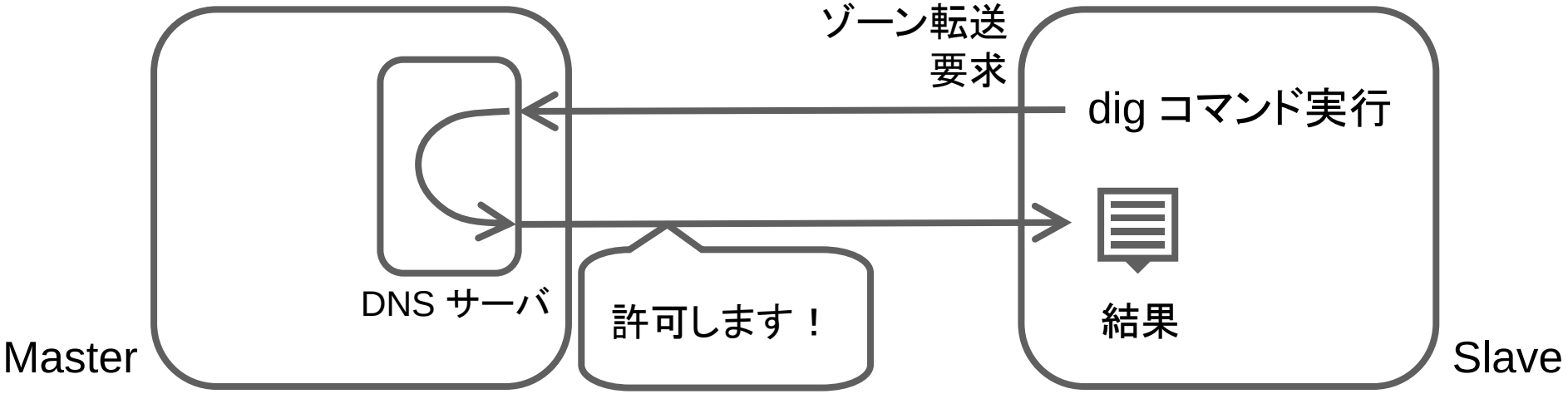
実行結果例

```
$ dig +norec @(マスタ) example.jp axfr

; <<>> DiG 9.9.2-P2 <<>> +norec @203.119.1.1 jprs.co.jp axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

1. ゾーン転送がうまくいかない – 調査と具体例

3. マスタサーバー側でゾーン転送が許可されていない場合



対応

✓ ゾーン転送の設定を見直す

→ 許可ホストの設定を間違えているかも……

2. ピリオドを忘れた – [出題編]

```
$ORIGIN          a.example.
$TTL             86400
@                IN      SOA     ns1.a.example.  root.localhost. (
                                 1047
                                 604800
                                 86400
                                 2419200
                                 3600
                                 )

                 IN      NS      ns1.a.example.
                 IN      MX      10 mail.a.example

ns1.a.example.   IN      A       192.0.2.54
ns1.a.example.   IN      A       2001:db8:53::53
mail.a.example.  IN      A       192.0.2.57
mail.a.example.  IN      AAAA    2001:db8:53::25
www.a.example.   IN      A       192.0.2.58
mail.a.example.  IN      AAAA    2001:db8:53::80
```

2. ピリオドを忘れた – [回答編]

```
$ORIGIN          a.example.
$TTL             86400
@                IN      SOA     ns1.a.example.  root.localhost. (
                                 1047
                                 604800
                                 86400
                                 2419200
                                 3600
                                 )

                 IN      NS      ns1.a.example.
                 IN      MX      10 mail.a.example.

ns1.a.example.   IN      A       192.0.2.54
ns1.a.example.   IN      A       2001:db8:53::53
mail.a.example.  IN      A       192.0.2.57
mail.a.example.  IN      AAAA    2001:db8:53::25
www.a.example.   IN      A       192.0.2.58
mail.a.example.  IN      AAAA    2001:db8:53::80
```

2. ピリオドを忘れた – [回答編] ～dig の場合～

```
$ dig a.example. MX @127.0.0.1

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> a.example. MX @127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8642
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;a.example.                     IN

;; ANSWER SECTION:
a.example.              15      IN      MX      10 mail.a.example.a.example.

;; AUTHORITY SECTION:
a.example.              8       IN      NS      ns1.a.example.

;; ADDITIONAL SECTION:
ns1.a.example.          8       IN      A       192.0.2.54

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Jul 18 20:47:26 2013
;; MSG SIZE  rcvd: 92
```

mail.a.example.a.example.

まとめ

- どこを調べているのか？を理解しよう
  - 再帰問い合わせ？非再帰問い合わせ？
- 道具の使いかたを知ろう
  - dig は友達
    - nslookupはやめよう
    - Windowsでも動く！
    - @でDNSサーバを指定、+norec オプション
  - 便利なWebサービス
    - DNS可視化の「DNSViz」
    - エラーチェックの「dnscheck.jp」
- よくあるトラブル事例
  - まずはログを確認！
  - TCPの53番ポート確認！
  - ファイヤーウォール確認！
  - CNAME 確認！
  - ピリオド確認！

Q&A