JAIPA セキュリティ部会　DNSSEC勉強会

DNSSEC 技術と運用

株式会社ブロードバンドタワー
事業開発グループ
大本　貴

BroadBand Tower, Inc.

- DNSSEC、その前に。

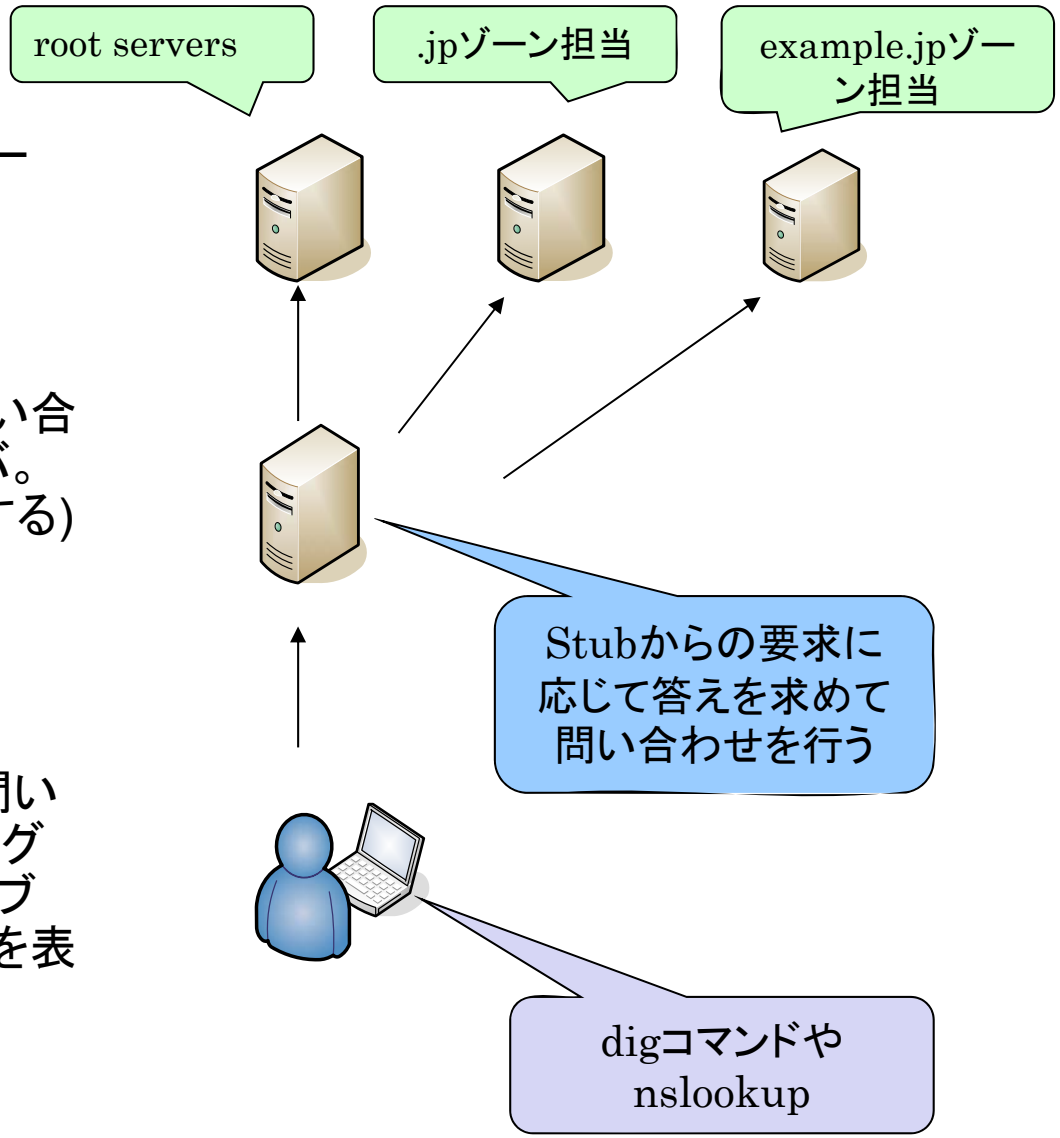
DNS(Domain Name System)について簡潔な用語定義

BroadBand Tower, Inc.

- Zone ・・・ ゾーン。名前解決を行うデータの集まり。
- RR ・・・ リソースレコード。
  ゾーンファイルに登録されたデータ資源。
- Query ・・・ クエリ。owner(ホスト名やIP address)を
  keyにして RRを索くということ。
- Authority ・・・ 権威。コンテンツ(名前空間)について管理
- Delegation ・・・ 権限委譲。コンテンツ(名前空間)の
  一部の管理を下位ゾーンに委譲すること。
- Recursive query ・・・ 再帰検索。(後ほど説明)

DNSサーバ色々

- authoritative server
  - 権威サーバ(コンテンツサーバ)。ゾーン管理している
- recursive server
  - リクエストに応じて再帰問い合わせを行うリゾルバサーバ。(キャッシュサーバも兼務する)
- stub resolver
  - リカーシブサーバに再帰問い合わせを行うリゾルバプログラム。基本的にはリカーシブサーバから得た回答内容を表示するだけ。



- DNSSEC(DNS Security Extensions) の技術概要

そもそもなんでDNSSECが必要?

BroadBand Tower, Inc.

・毒入れアタック(ポイゾニング)

問い合わせ情報に対して第三者が偽装パケットにより偽のレコードをユーザに送りつける

→結果

ウィルス仕込み済みwebサイトへ誘導。

メールを正しい相手に送信できない。

(しかもエンドユーザは気が付きにくい。)

- インターネットの基盤であるDNSの信頼性が揺らいでいる。→何を信じればよいのか。

何が主たる原因か?

BroadBand Tower, Inc.

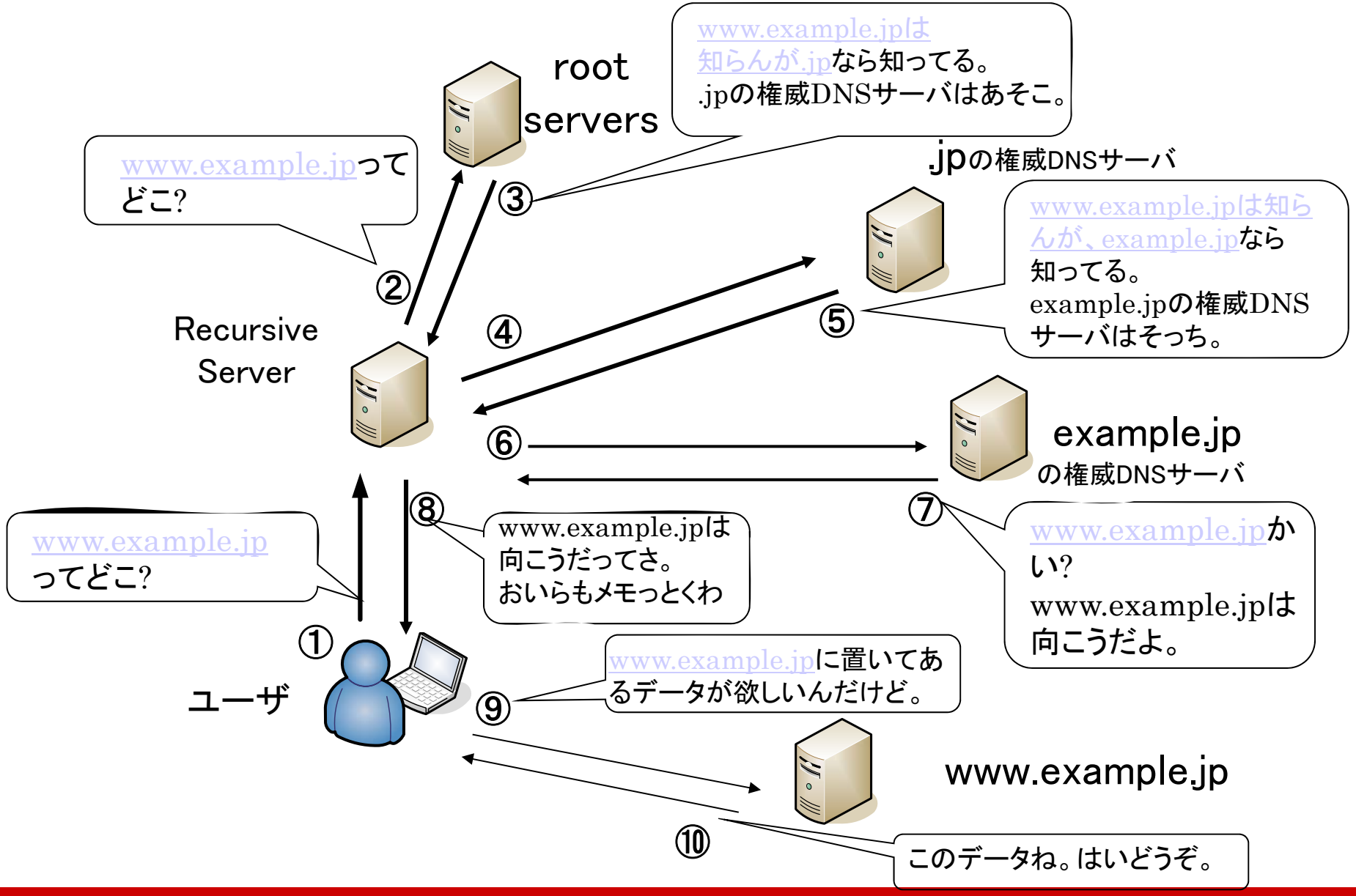
クエリの送受信をUDPでやりとりしているから。

→UDPでこぼれたらTCP fallbackで対応

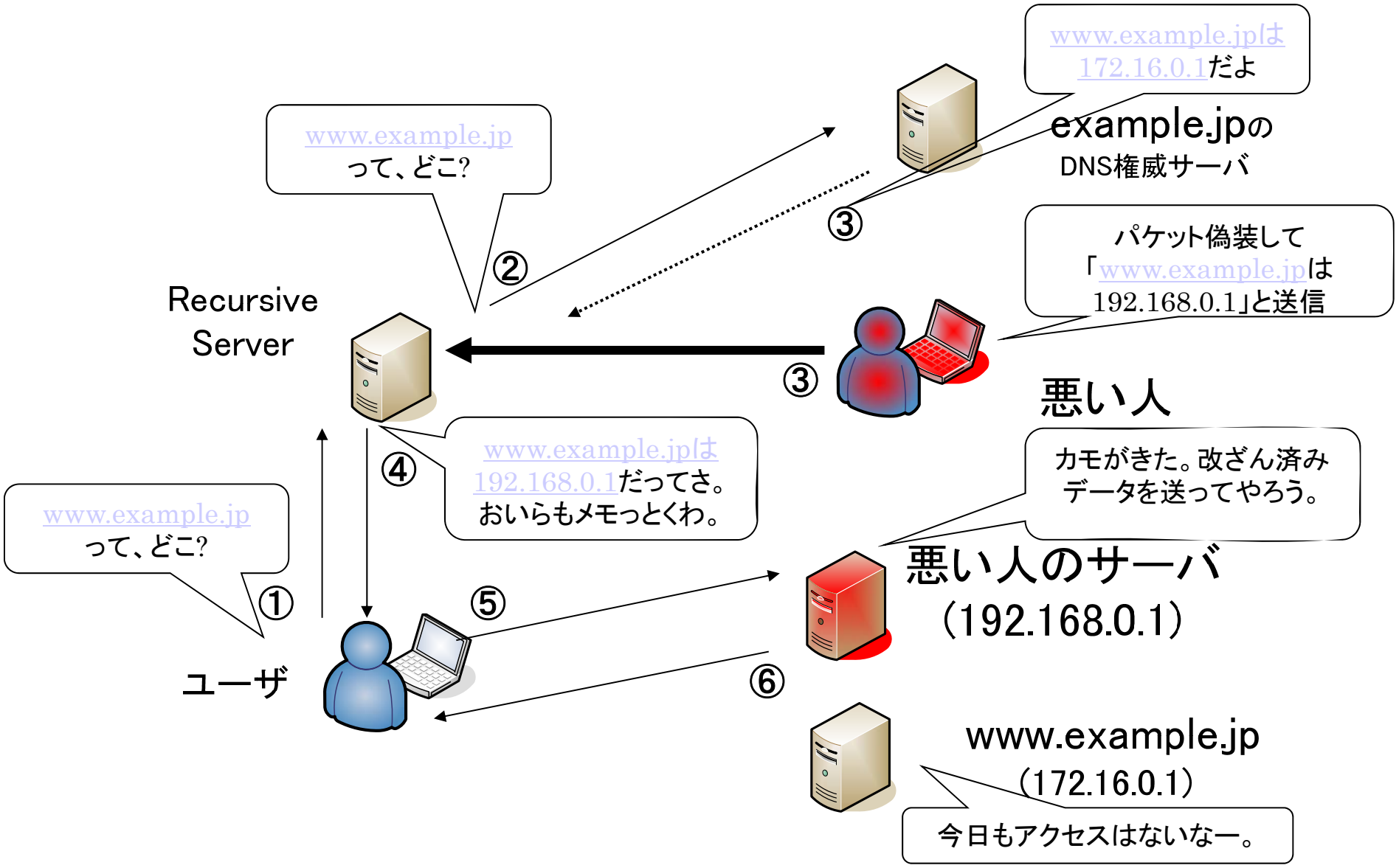
→そもそも最初からTCPで良いのでは?

→ルートゾーンなどで膨大なリクエストを短時間で処理するためにはUDPで処理を軽くしていく必要がある。

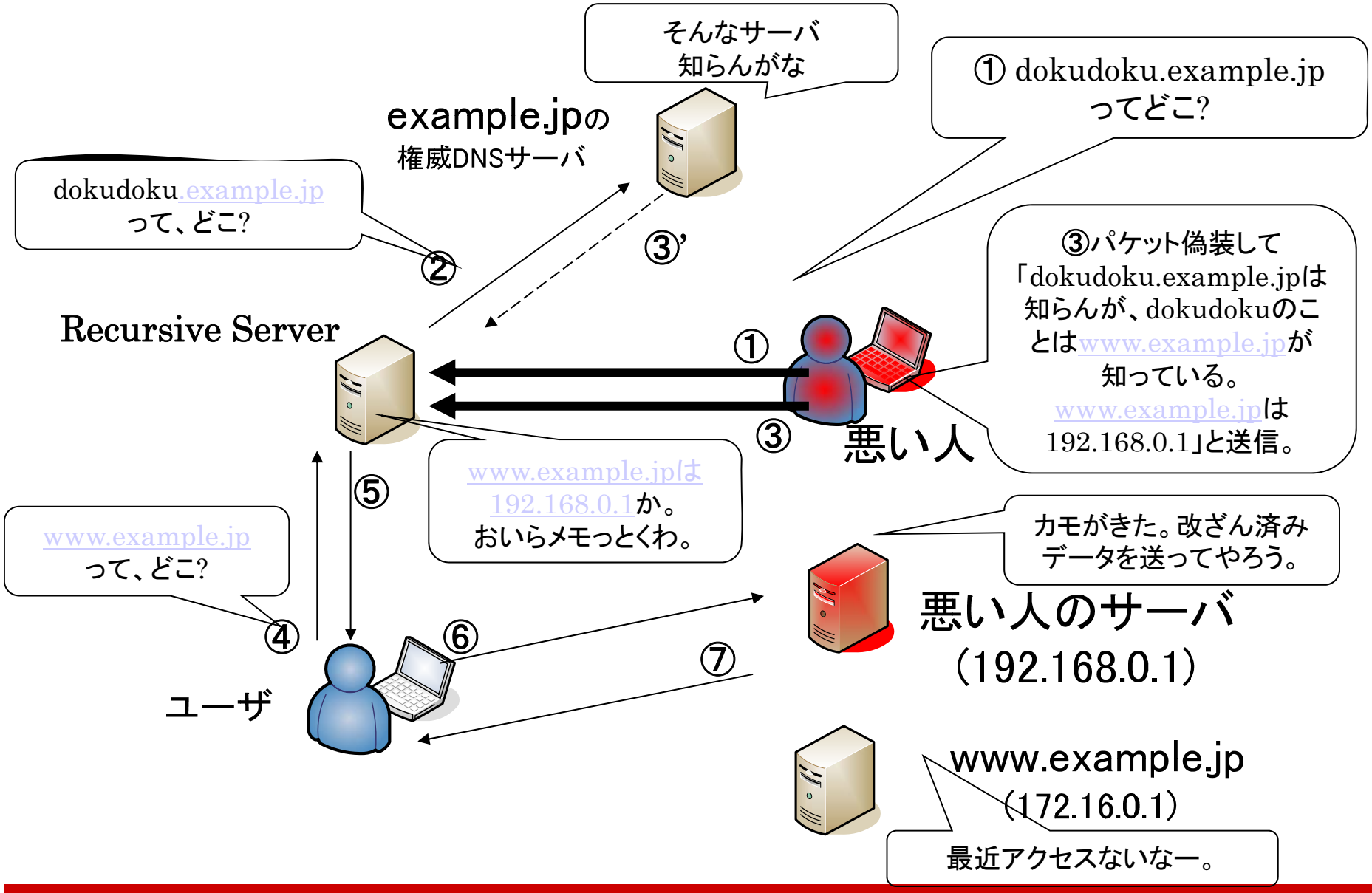
DNSの名前解決の流れ



キャッシュポイゾニングの概要

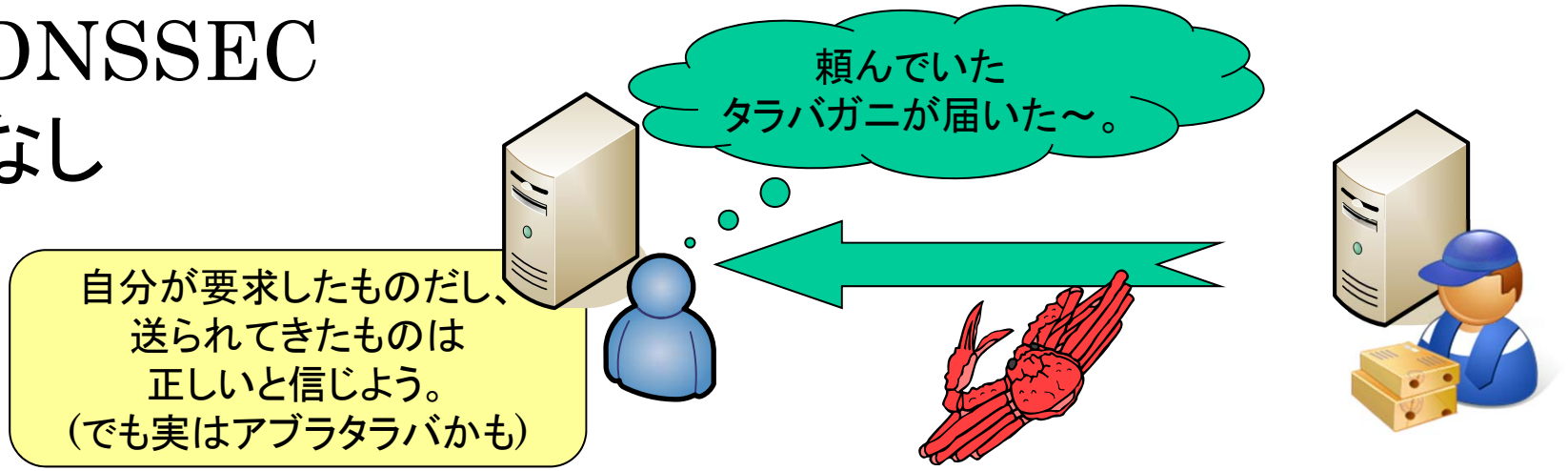


カミンスキーアタックの概要

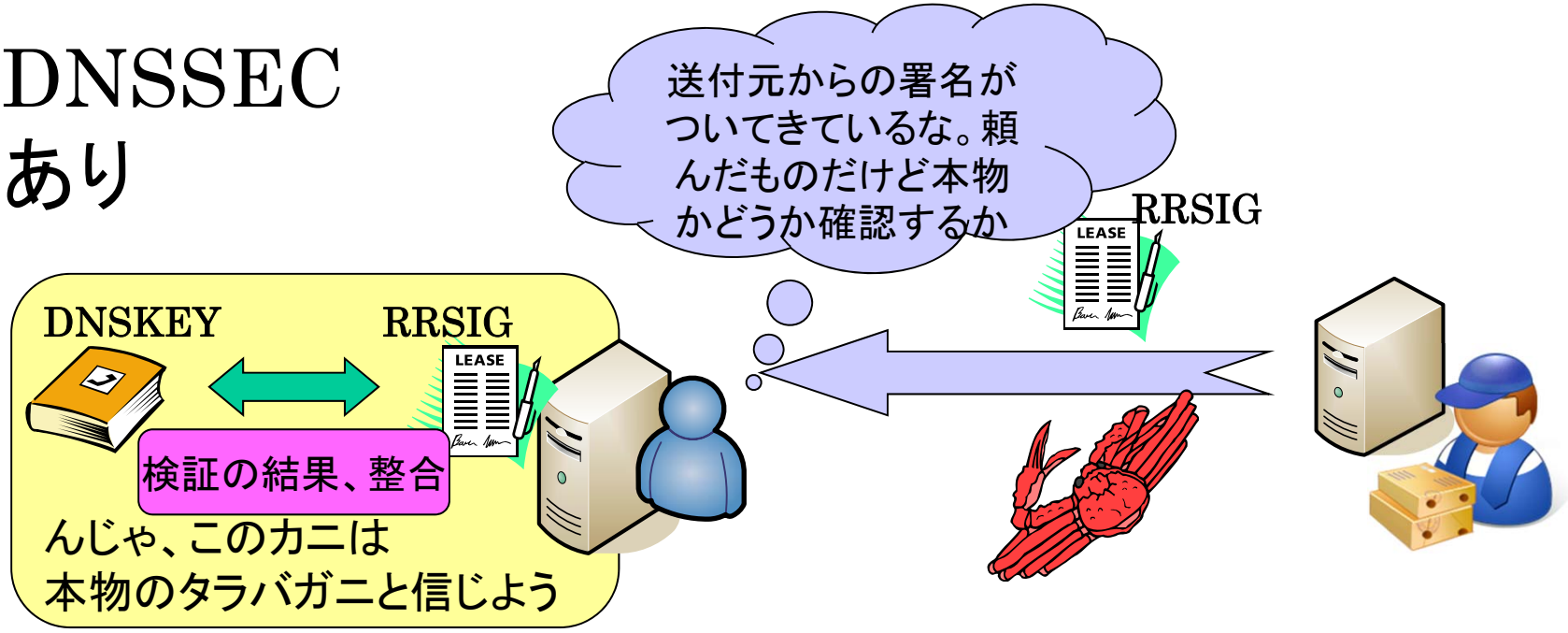


DNSSEC、簡単にするとこんなイメージ?

DNSSEC なし



DNSSEC あり



・DNSSEC、誰に負担があるの？

- ドメイン名登録者(独自ドメイン取得している・したい人)
  - DNSSEC化への判断。
    (自分の管理しているドメインの信頼性。
    上位ゾーンが対応していないならDLV使うか)
- DNSサーバ管理者(ISP、ホスティング事業者 etc.)
  - 権威DNSサーバ
    - 管理しているゾーンのDNSSEC対応、
      鍵の定期的なロールオーバなどが業務に追加。
  - リゾルバサーバ(キャッシュDNSサーバ)
    - キャッシュサーバへトラストアンカーの導入。
- レジストラ
  - 鍵を上位ゾーンへ登録する仲介処理。
- レジストリ
  - 下位ゾーンからの登録申請に対してDS登録処理。
    DS申請を受け付けるシステムの構築。
- ユーザサポート担当者
  - 障害時の切り分けのためのノウハウ習得。

DNSSEC、導入するとどうなるの？

- 勝ち得るもの

DNSの信頼性を高めることができる。真正性を担保できる。

- 試練

めんどくさい。
  - ロールオーバーとか鍵の管理とか再署名とか。
  - 信頼の連鎖を形成するための第三者との連携。

管理コストの増加
  - 作業自体の単純増加、チェックするポイントの増加。

ハイスペックなハードの必要性(特にキャッシュサーバの)
  - 署名の検証処理が加わることで必要スペックも増加。

SERVFAILによるトラブルへの懸念
  - 設定ミスなどで、検証に失敗する状況になると、
    該当ドメインの名前解決しない→障害に繋がる

顧客へのサポートに分岐が発生。(ISPなど)
  - 問題がDNSSECかDNSなのかの切り分けが生じる。
  - WindowsのnslookupではdigのようにSERVFAIL判定を判断できない。
    →Windows用のdigプログラム導入してもらうことに？

DNSSECに対応できる環境にするには？

BroadBand Tower, Inc.

- DNSSEC対応DNSサーバ
  - BINDのバージョン9.3以降 (権威サーバ・キャッシュサーバ)
    - ただし、最近だと9.7.1-p1にDNSSECにバグがあるので使わないように。
    - DNSSECだけではなくDNSのバグもあるので9.7.1-p2以降推奨。
    - 9.7からはSmart SigningというDNSSEC用の仕組みが機能追加。
  - NSD 3.x(権威サーバ)
  - Unbound 1.x(キャッシュサーバ)
  - Windows2008serverはまだ実装不十分。(NSEC3未実装)
  - この資料内の例ではBIND9.7.xを設定事例として紹介します。
- EDNS0対応とTCP fallbackへの対応
  - キャッシュサーバが対応してても・・・

    qmailは、512byte 問題が存在。

    Christopher K. Davis氏が作成したpatch1.03
    (oversize DNS packets patch)を導入することが必須。

    http://www.ckdhr.com/ckd/qmail-103.patch

    ※IPv6で既に問題が顕在化しているので対応済みかも。

DNSSECのために新規に定義されたレコード

BroadBand Tower, Inc.

- DNSKEY ・・・・ DNSSECの公開鍵情報レコード
  - ZSK(Zone Signing Key)
    ・・・ RR(Resource Records)Setを署名する鍵
  - KSK(Key Signing Key)
    ・・・ TYPEがDNSKEYのRRSetを署名する鍵

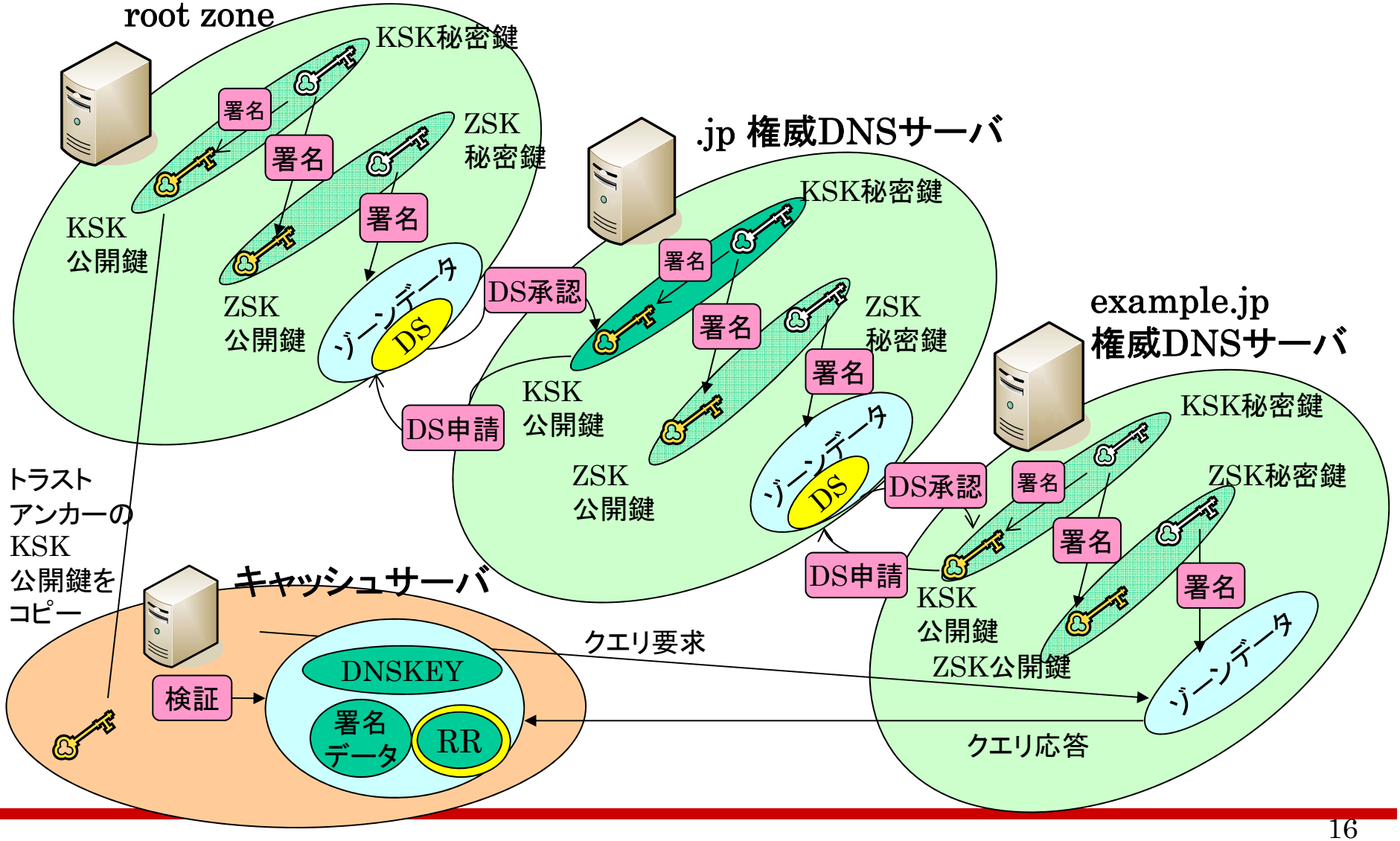
- RRSIG(Resource Record Signature) ・・・・・
  ゾーンファイル内の各レコードの署名を表現する
  レコード。(RRSIG自身には署名しない。)

- DS(Delegation Signer) ・・・・
  子ゾーンのKSKを元に生成されたハッシュ。
  上位DNS権威サーバ(親)のゾーンに登録してもらい、
  親ゾーンのZSKにより署名してもらう。

- NSEC & NSEC3& NSEC3PARAM(後ほど説明)

DNSSECの真正性担保の仕組み

- 「信頼の連鎖」により、レコードの信頼性を担保



鍵生成

- なんで鍵は二種類あるの?
  - 鍵をZSKとKSKを分けて管理することによって、鍵サイズや暗号強度を、役割に合わせたパラメータで運用できるメリットがある。
  - KSKは親ゾーンとの連携が必須になるため、鍵交換する頻度は少なくしたい。
- 鍵の諸パラメータは、ふつう どーする?
  - 鍵の暗号化アルゴリズムはRFC4641的にはRSA/SHA-256が推奨されている。
  - 鍵サイズは1024bit以上を推奨。(KSK・ZSK)
    (ルートゾーンのKSKでは2048bitが適用されている)
    ZSKはKSKよりも暗号強度が弱くとも良いが更新頻度でカバーしていく。

鍵生成の実際

ZSK…. dnssec-keygen -a RSASHA256 -b 1024 ゾーン名
KSK…. dnssec-keygen -a RSASHA256 -b 2048 -f ksk ゾーン名

```
# cat Kohmo.to.+008+60799.key
; This is a key-signing key, keyid 60799, for ohmo.to.
; Created: 20100913141843 (Mon Sep 13 23:18:43 2010)
; Publish: 20100913141843 (Mon Sep 13 23:18:43 2010)
; Activate: 20100913141843 (Mon Sep 13 23:18:43 2010)
ohmo.to. IN DNSKEY 257 3 8 AwEAAdLEIOzWiWrHtOEdc60BH4v4Qh6O8JHCO6cNwi5LsF
nLTJAsc4AV CFV5YG131CIHPAfM1hGnBe4qRnjGj+uA7hIDPD/CsKOd9zaFk8LFYiLb…
```

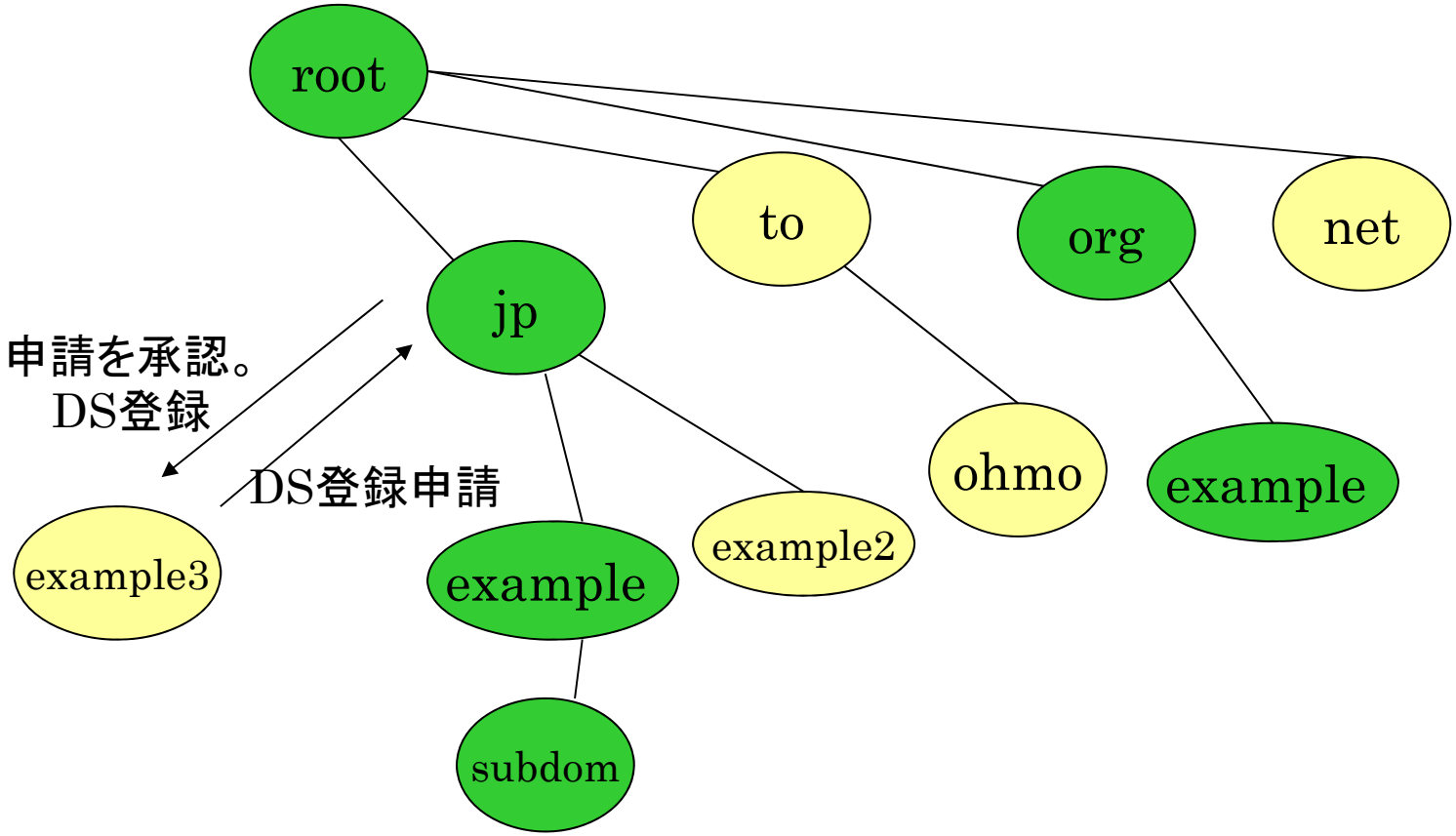
```
#cat Kohmo.to.+008+48543.key
; This is a zone-signing key, keyid 48543, for ohmo.to.
; Created: 20100913141801 (Mon Sep 13 23:18:01 2010)
; Publish: 20100913141801 (Mon Sep 13 23:18:01 2010)
; Activate: 20100913141801 (Mon Sep 13 23:18:01 2010)
ohmo.to. IN DNSKEY 256 3 8 AwEAAcPn4Oj7xRAYMRZ2IkFQmRi5S9wNy7lNkDxMijyB2f
d3aNzw5i1G l3YsG3FHBpgCvbmj3pMkpCIyVi/l74xt2MLF5jAeQKtYdvP2HUjwIsEl….
```

```
#cat Kohmo.to.+008+48543.private
Private-key-format: v1.3
Algorithm: 8 (RSASHA256)
Modulus:
w+fg6PvFEBgxFnYiQVCZGLlL3A3LuU2QPEyKPIHZ93do3PDmLUaXdiwbcUcGmAK……
```

- 実際の処理は後半に。

上位ゾーンへのDSレコード登録

- KSKを上位ゾーンに署名してもらうことで信頼の連鎖を形成させる。



署名と「信頼の連鎖」構築の実際

- DSゾーン申請前提
  - named.confに署名したいゾーンが設定されていること。
  - KSKの鍵生成とZSKの鍵を生成済み。

- 署名してみる。

dnssec-signzone –S example3.jp. (←対象ゾーン名)

  - 処理が正常終了すると、dsset-example3.jp(対象ゾーン名)というDSSetファイルと、署名済みゾーンファイル「example3.jp.signed(対象ゾーン名+.signed)」が生成される。DSSetファイル内に記載された2行のうち、いずれかを上位ゾーンに登録申請する。

```
example3.jp.   IN DS 60799 8 1 D736F20FB259279A4A5FFCEB45536C5CAD33EC78
example3.jp.   IN DS 60799 8 2
    EBC75A18FAEDA255DCD9148418BFCDCF95D6BD6E367EB469EA7BA83A 713CF50F
```

1=SHA1
2=SHA256

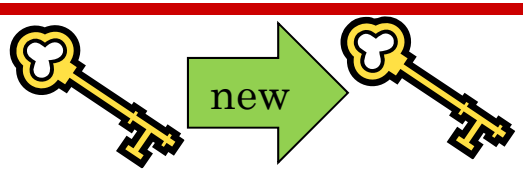
署名の運用 (有効期限)

- 署名には有効期限が設けられている。
  - 有効期限が切れると署名が有効ではない→名前の検証に失敗する→SERVFAILの反応→名前解決できない。→メールは届けられない。webも見れない。

- かといって有効期限を長くしすぎていると期間内に鍵がクラックされる可能性も・・・。
- 有効期限が決められているからサーバ内の時計がずれていると・・・。

- KSKの有効期限は長く、鍵の暗号強度を高く、
  ZSKの有効期限は短く、暗号強度は多少緩め、
  というのが推奨されている。
  - RFC4641的にはKSKは13ヶ月に1度のタイミングでの交換を提案している。
  - ZSKは1ヶ月に1度程度を目安(dnssec-signzoneはオプションなしだと30日で設定される。)

鍵のロールオーバー

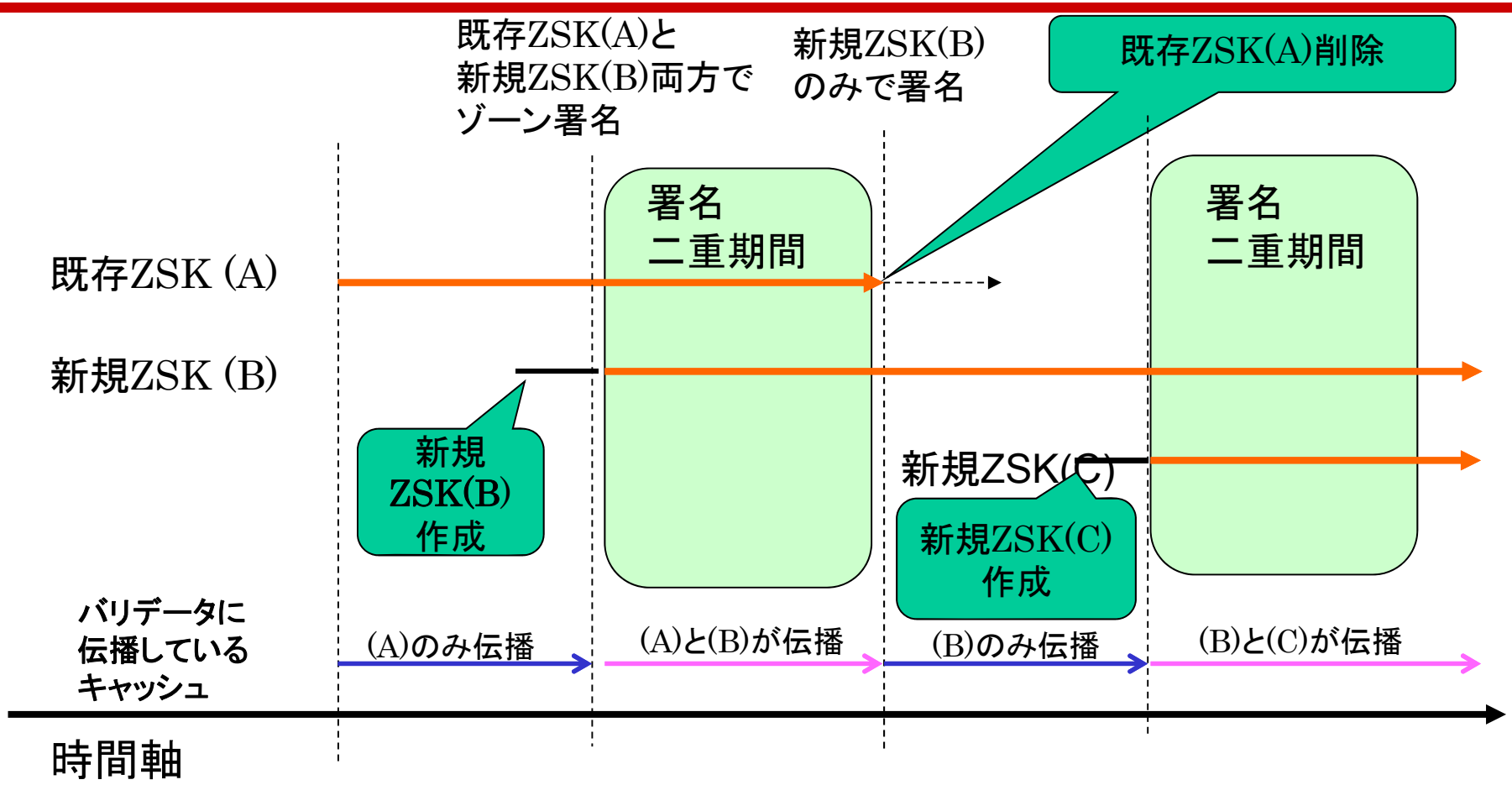


ロールオーバ手法としては下記の手法がある。

ZSK
- Double signatures (二重署名方式)
- Pre-publication (事前公開方式)

KSK
- Double signatures (二重署名方式)
- Double-DS (二重DS方式)
- Double-RRSet (二重RRSet方式)

ZSK / Double signatures （二重署名方式）

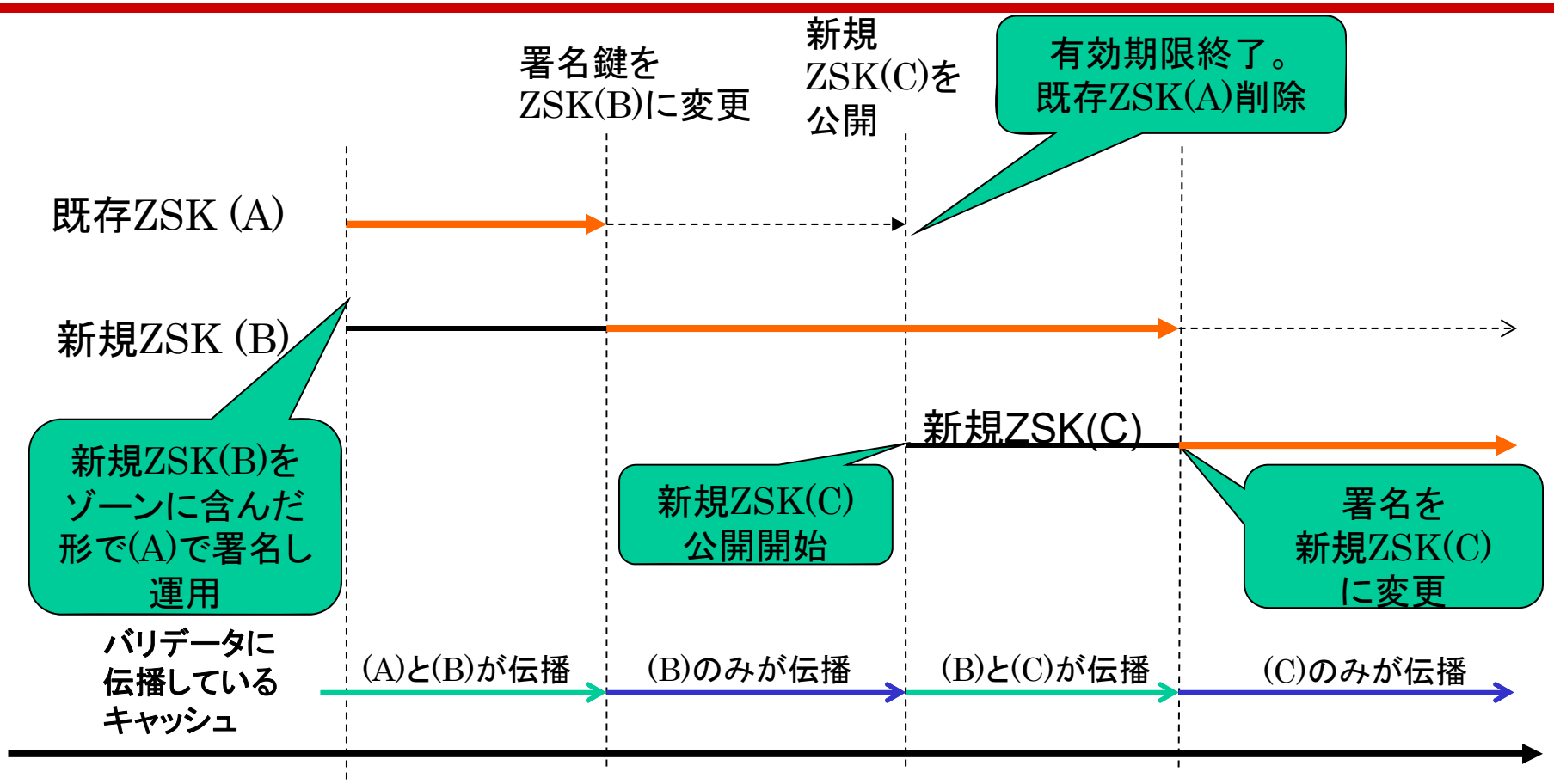


active

rolled

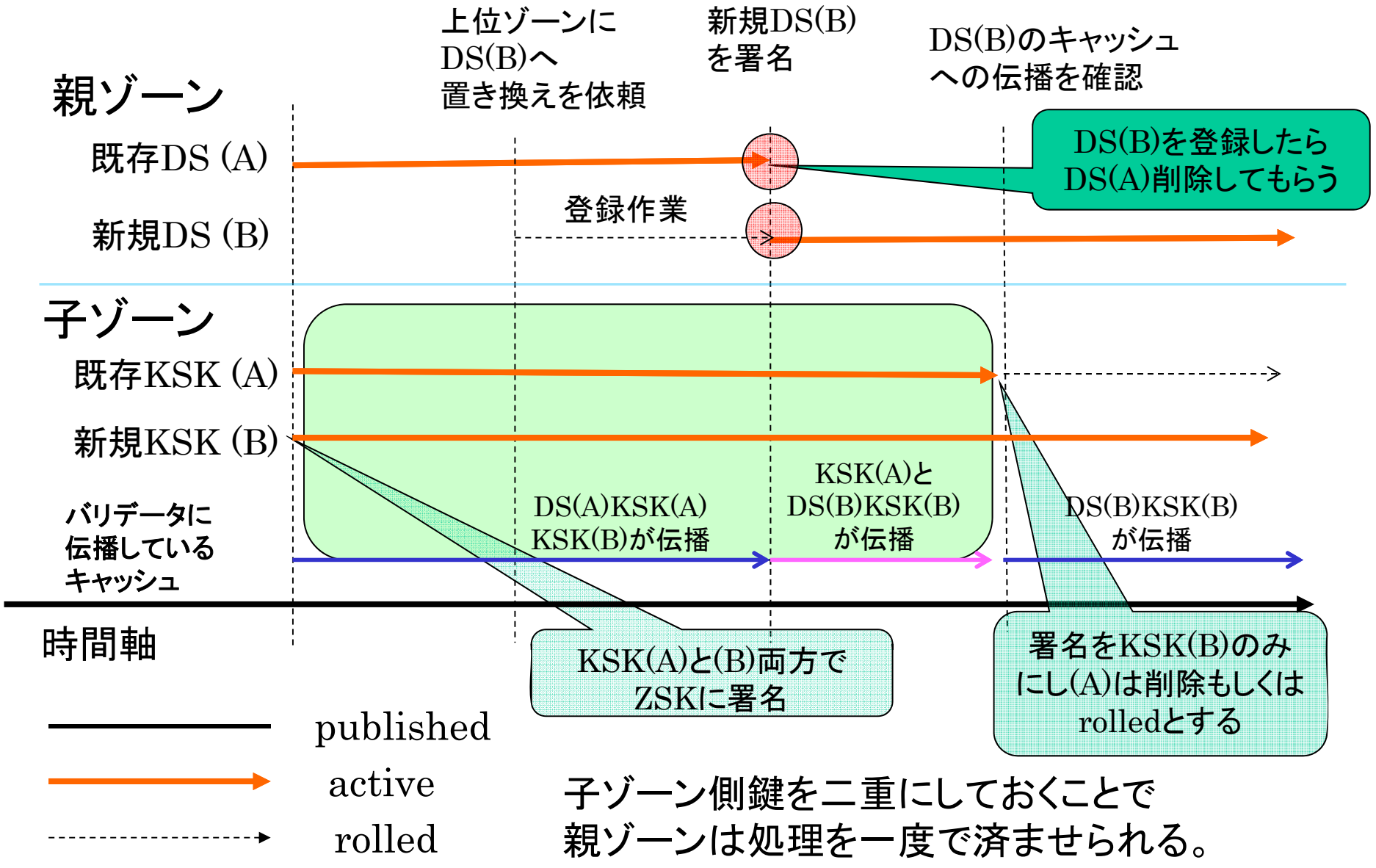
既存と次期DNSKEYと両方で署名し、伝播を待つ。充分に伝播が行き渡っているならば、DNSKEYをいつ切り替えても問題が出ないように考慮した方式。

ZSK / Pre-publication(事前公開方式)



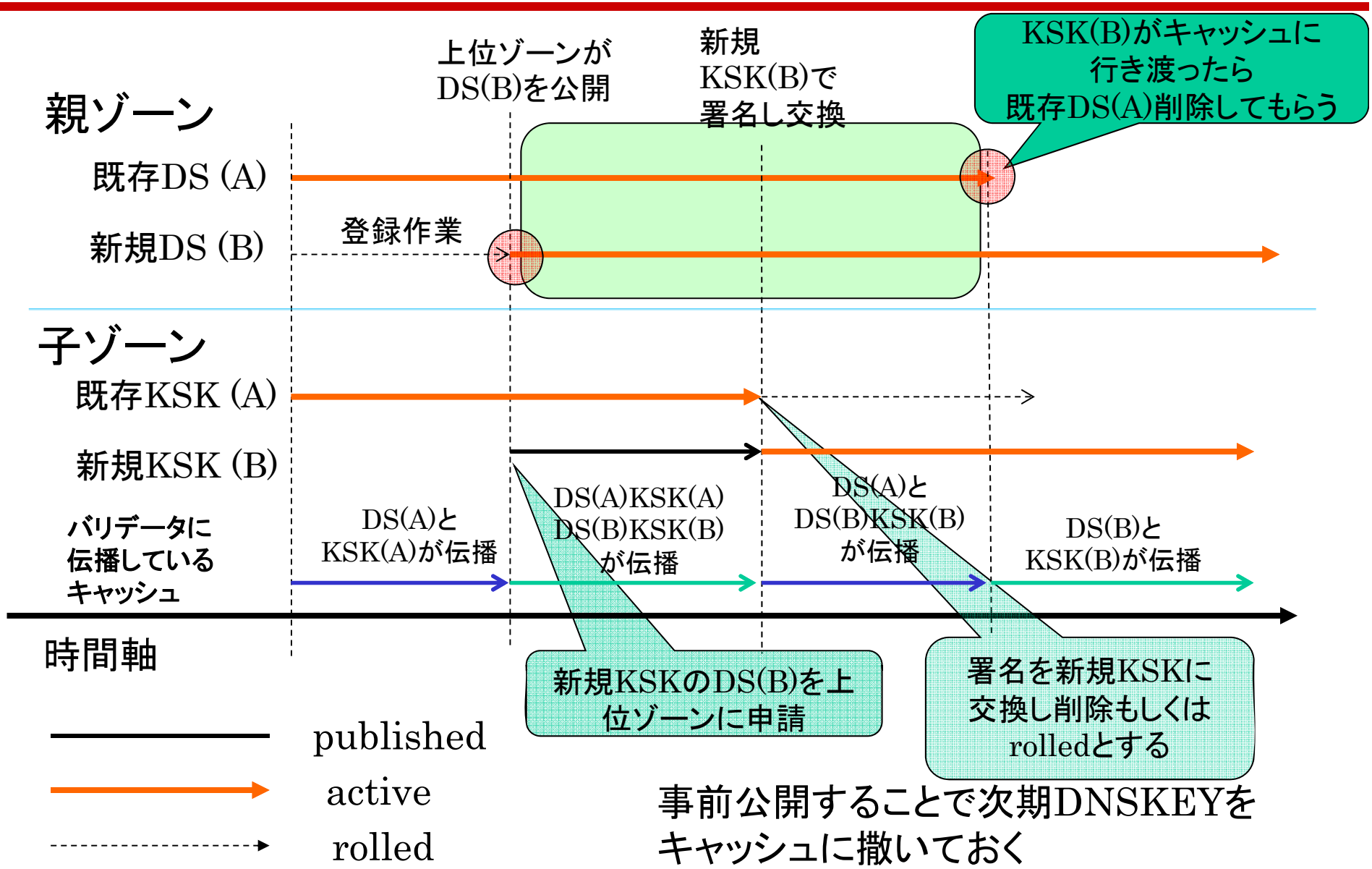
事前公開することで次期DNSKEYを
キャッシュに撒いておける。
これにより署名を切り替えても問題ない
よう考慮した方式

KSK / Double signatures(二重署名方式)



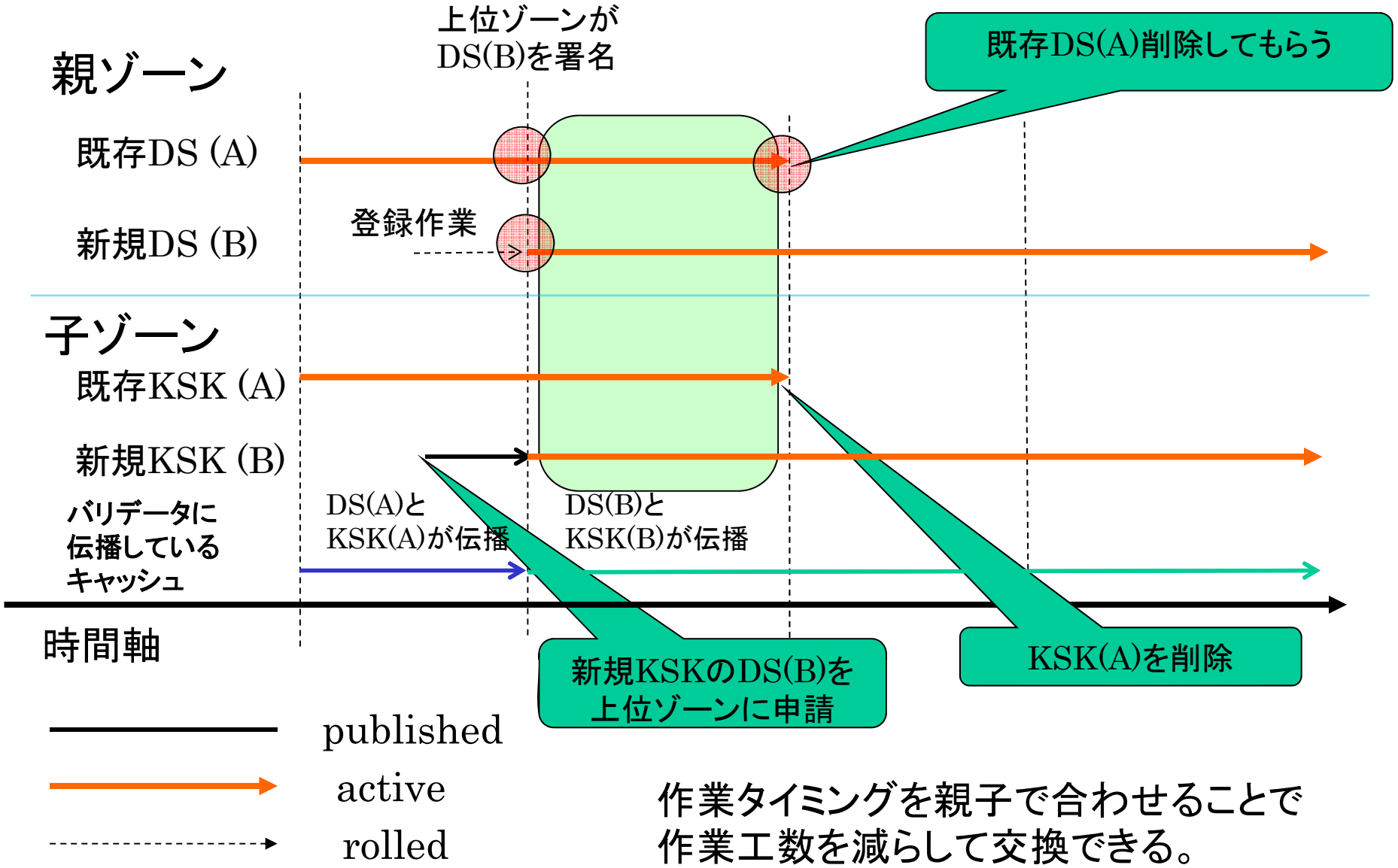
子ゾーン側鍵を二重にしておくことで
親ゾーンは処理を一度で済ませられる。

KSK / Double-DS(二重DS方式)



事前公開することで次期DNSKEYを
キャッシュに撒いておく

KSK / Double RRSet (二重RRSet方式)



作業タイミングを親子で合わせることで
作業工数を減らして交換できる。

鍵のロールオーバーポリシー

- 危殆化して初めて交換するのか定期的に交換するのかの考え方が二種類ある
- 特にKSKの場合は親ゾーン管理者との連携作業になるのでロールオーバーのポリシーは良く考えなくてはならない。

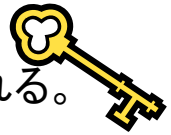
- 危殆化して初めての場合
  - 定期的作業の頻度軽減
  - 作業フローを手が忘れてしまう。

- 定期的に交換する場合
  - 人的作業コストの発生
  - KSKの場合、親ゾーン管理者と都度やりとりする必要が出てくる。
    (親ゾーン管理者の作業タイミングの調整や
    待ち時間等も発生。)
  - 定期的に行うので作業をルーチン化できる

トラストアンカー[Secure Entry Point(SEP)]の設定

BroadBand Tower, Inc.

- バリデート(検証を行う)サーバに、トラストアンカーとなっているKSKをコピーしてきて設定する。
  KSKはDNSSECに対応しているゾーンは公開している。
- # dig *ドメイン名* DNSKEY +norec @対象DNS権威サーバで表示される。
  (実際はdig . DNSKEY +norec @m.root-servers.net推奨)



```
ohmo.to.        1718   IN     DNSKEY  257 3 5
                              AwEAAcOSWzyHSLylvhuKGHn38BMYhK9E6l1yzIerWB+Kt2goS1GasMFq
                              yjkySWpAORwcSZHI3N871KAQGOdUBinGruEkQMCXIslJlI/HM+.....

ohmo.to.        1718   IN     DNSKEY  256 3 5
                              AwEAAb+lrnM0/xl1WnxFCpg4uAOmyhjNvUT+Dx2564w8RK0gmRQmMkuC
                              3UndYy5K71XZHbZizA5MwKbwx89h6Z/4IERo0LFkBYEVOin5oxxX ……..
```

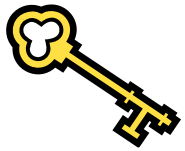
- named.conflに以下を設定すると、トラストアンカーが署名している下位ゾーンについては検証を行う。

  # trusted-keys { };でトラストアンカーのKSKを登録。
  (bind9.7からRFC5011に準拠したmanaged-keysステートメントが実装された)

```
managed-keys{
        "." initial-key 257 3 8 " hoge1234hoge5678…..hogehoge";
                };

options{
        dnssec-enable yes;          # bind9.5以降ではdefaultでyesなので必要ない
        dnssec-validation yes;    #バリデータとして動作させるか
        }
```

トラストアンカーの注意点



1. Validator(バリデータ) は誰がやるの?
   - リゾルバ(キャッシュ)サーバ
   - 仕様としてはエンドのクライアント側でもバリデータとして設定可能
     - ただし、現段階ではエンド側で対応できるのはこれからというところ。

2. トラストアンカーとなるKSKをコピーしてくる手法の信頼性
   - 前頁で紹介した方法だとKSKのキャッシュがすでに汚染していたら・・・。
     - 取得したKSKからDSレコードを生成し(dnssec-dsfromkey) 、一方でHTTPSで取得したWEB上のDSレコードをPGP鍵を利用して真正性を検証、2つを突き合わせてみる。
       https://data.iana.org/root-anchors/root-anchors.xml

3. 親ドメインと子ドメイン間のDNSSEC対応非対応の選択
   - 署名する単位はゾーン単位だから必要なら別ゾーンに小分けする必要がある。(権威サーバ)
     - 下位ドメインで先にDNSSECに対応したいドメインがあればゾーンを分ければよい。

4. トラストアンカー設定上での注意点
   - BINDではtrusted-keysにDSを設定するのは×
     DNSKEYじゃないとだめ。
   - unboundではDSでもDNSKEYでも両方OK

NSEC & NSEC3

BroadBand Tower, Inc.

- NSECってなに?
  - 登録していないホスト名についての不在証明レコード
    - 検証した際に「そのホスト名のレコードは存在しない」ということを証明する。

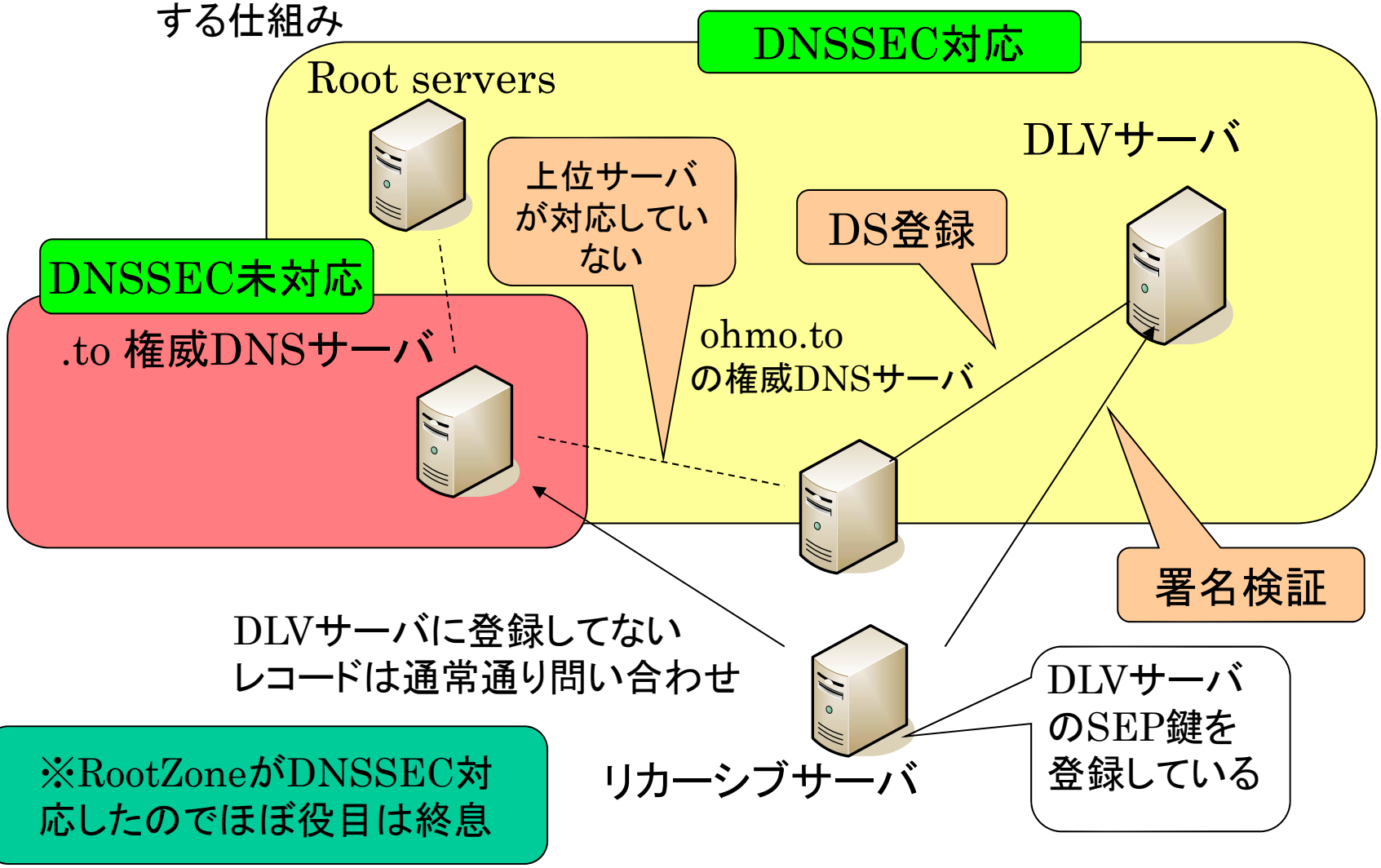
- zonewalkによる露出の危険性(NSEC)
  - NSECの場合、zonewalkをすることで登録しているホスト名が意図せずに露出してしまう。
    - そのゾーンで管理しているホスト名を問い合わせた際に、アルファベット順に次に登録されているレコードが表示される。
    - ohmori.example.jpの次の登録レコードはohmoto.example.jpと表示
      - それってohmo(ri～toの)間には登録ホストがない事の証明でもある。

- →zonewalk対策となるNSEC3が登場

- NSEC3では、FQDNを一方向にハッシュ化して、ハッシュ値をてがかりに不在証明。
  - →ただし、NSECに較べても、キャッシュサーバの処理に結構な負荷がかかる
    Janog26 民田さん@JPRSの報告参照
    http://www.janog.gr.jp/meeting/janog26/doc/post-dnssec-min.pdf

DLV(DNSSEC Lookaside Validation)(参考)

- 通常のドメインツリーとは別にDNSSEC専用の問い合わせ先を用意し、root zoneや上位TLDが対応していなくとも署名の検証を可能とする仕組み



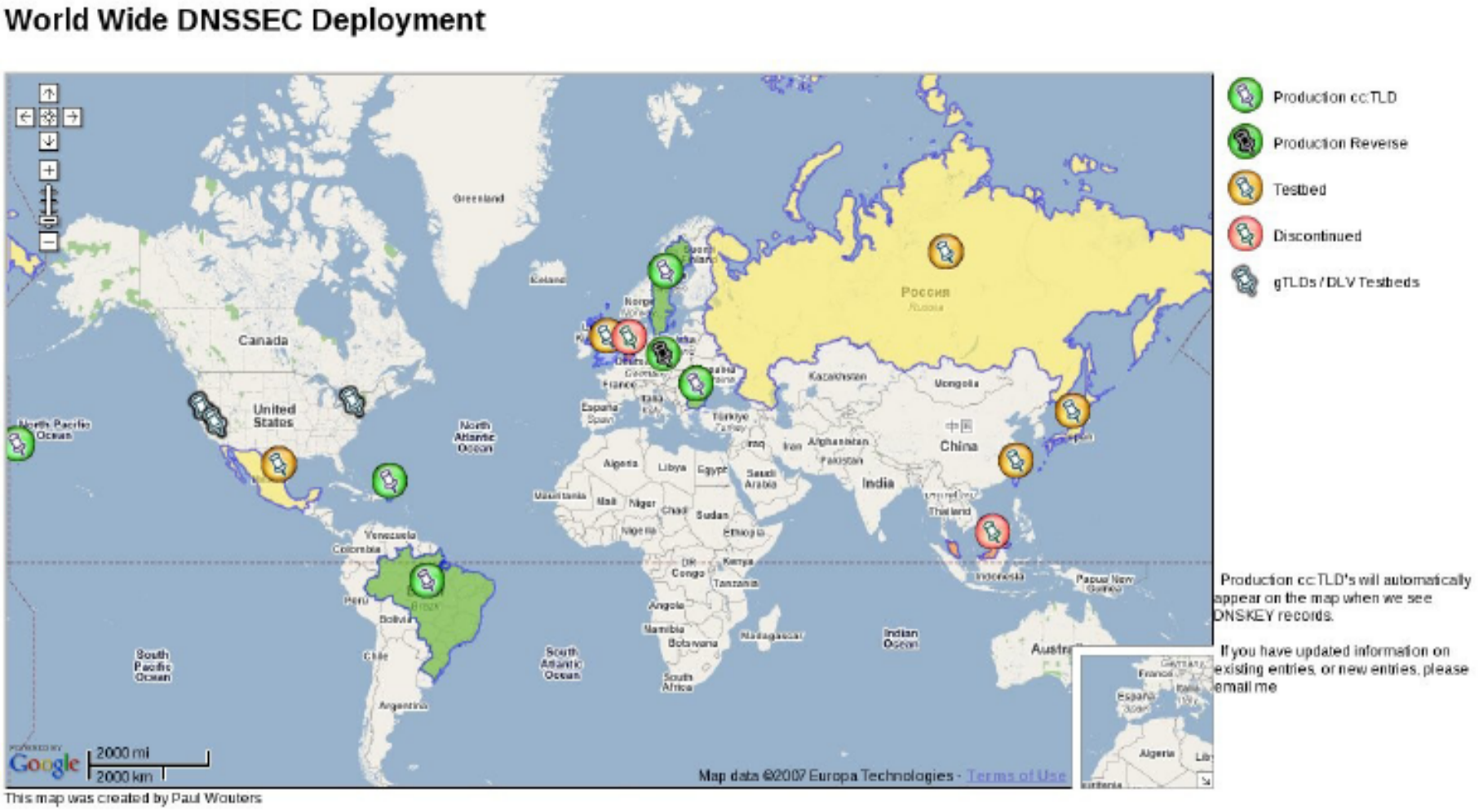
OpenDNSSEC(参考)

BroadBand Tower, Inc.

- 覚えることも多くて運用するにもかなりの慣れが必要そう・・・
- 海外のレポートでもDNSSECに対しての知識がまだエンジニアに浸透していない(対応できるエンジニアが少ない)のが悩みの一つとされている。http://www.afilias.info/webfm_send/119
- ということで、OpenDNSSECというツールがあります。
  http://www.opendnssec.org/
  (2010年9月現在ver1.12が公開中)
  - 鍵更新とゾーン署名の自動化や、ゾーンの完全性チェック、ソフトウェアHSMも含んだツール

各TLDの対応状況@2007年

BroadBand Tower, Inc.



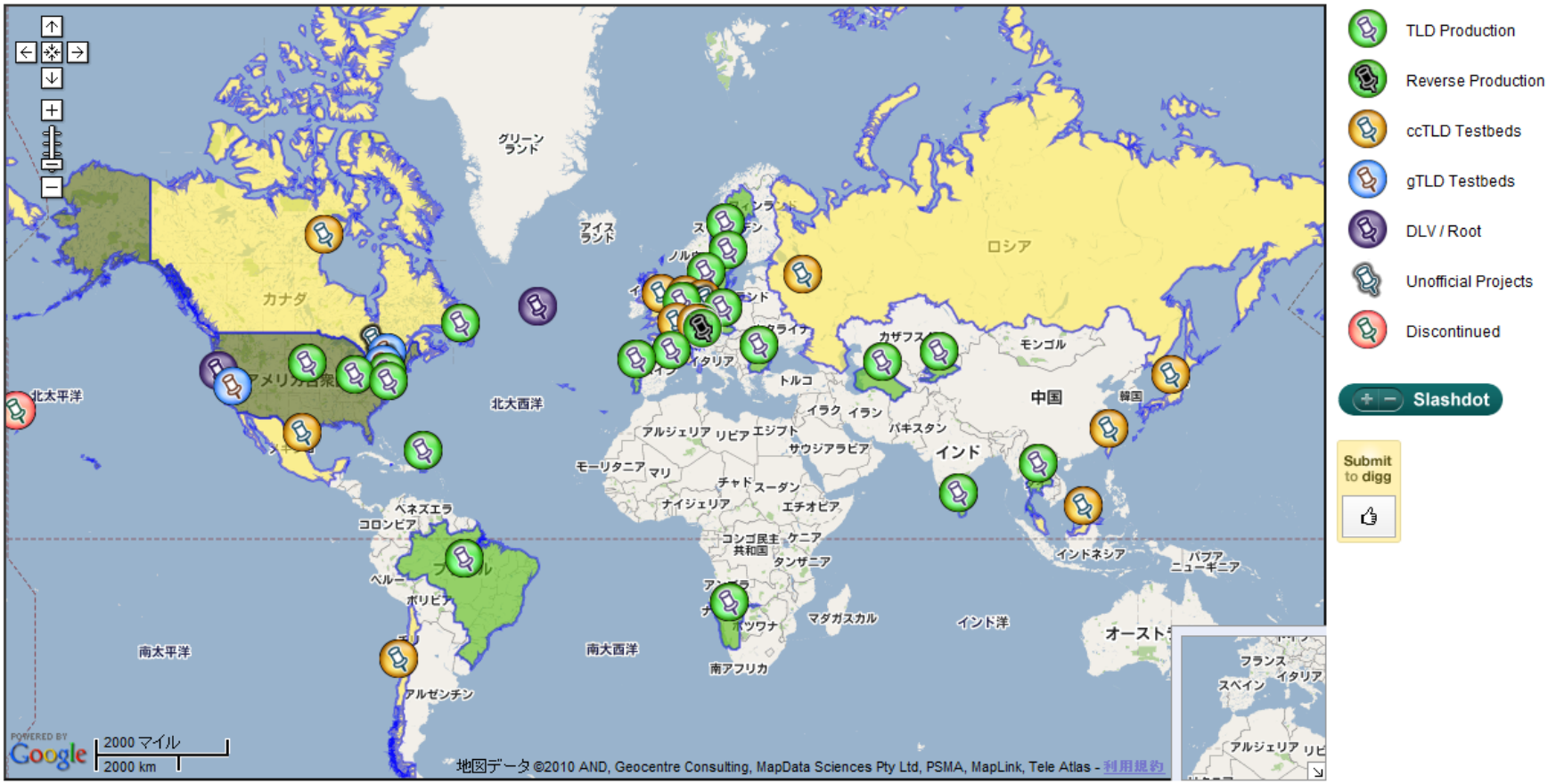
Paul Wouters氏資料 http://www.xelerance.com/talks/sector/Sector2007DNSSEC.pdf より

各TLDの対応状況2010年現在

BroadBand Tower, Inc.

World Wide DNSSEC Deployment

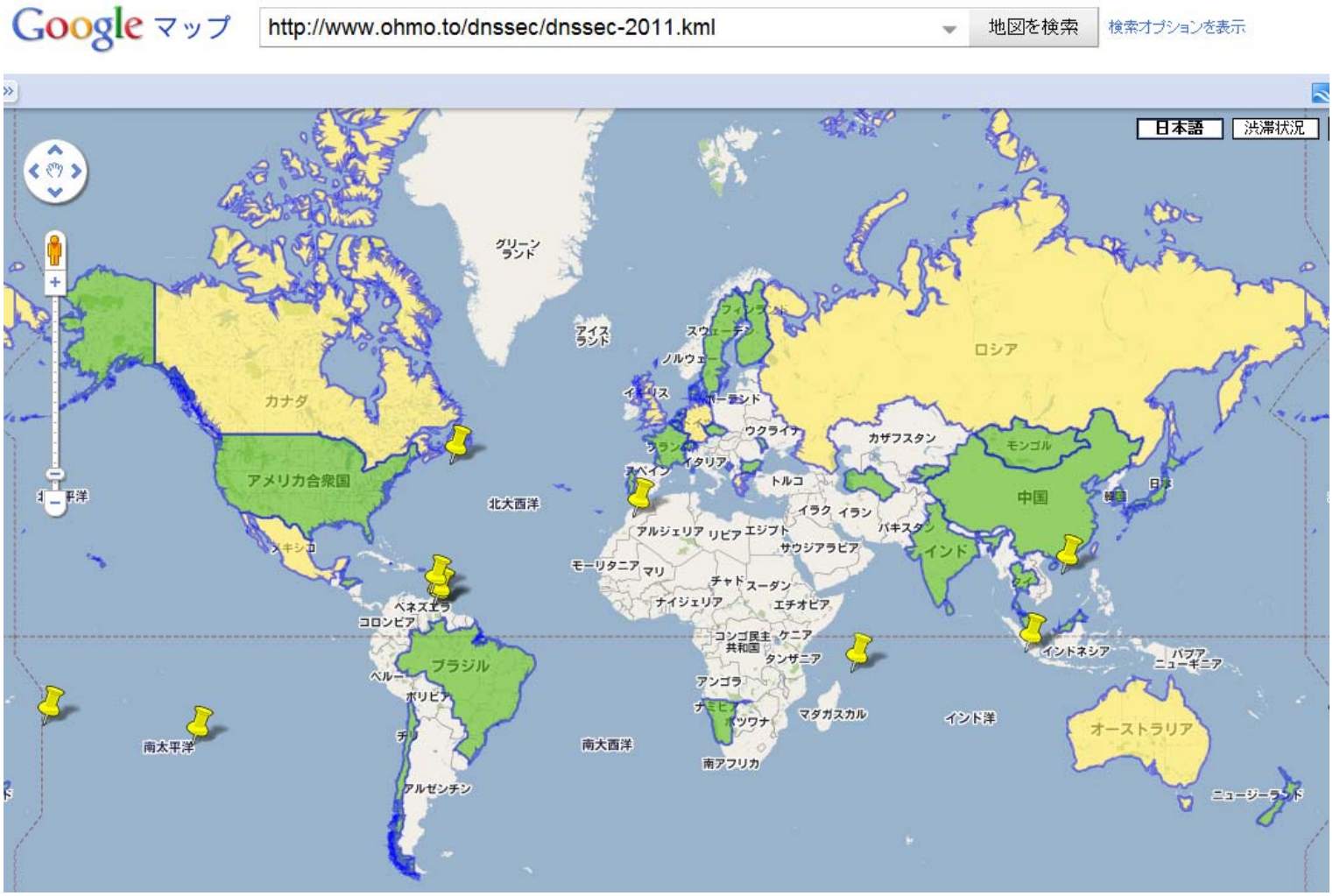
See also DNSSEC Theory and World Wide Deployment by Paul Wouters, November 21, 2007, SecTor



This map was created by Paul Wouters

Paul Wouters氏のweb公開資料 http://www.xelerance.com/dnssec/ より

各TLD対応2011年初(現在告知分からの想定)



- 詳しいステータスはJPNICの是枝さんがまとめた資料が参考になります。
- http://dnssec.jp/wp-content/uploads/2010/07/20100721-tld_dnssec_deployment-koreeda.pdf

技術概要編 まとめ

- 今現在も危険性は潜在しており、国内外問わず、日々DNSSEC対応が進んでいる。
- 本格的なDNSSEC運用に向けて、負荷を軽減するために色んな機能が開発されてきているので敷居は少しずつ低くなってきている。
- DNSSECの導入には負担や課題も多いが、インターネットの基盤を維持するため、導入は進めるべき。