OpenDNSSECチュートリアル

@DNSOPS.JP BoF(2009.11.24)

NRIセキュアテクノロジーズ株式会社
エンタープライズセキュリティサービス部
中島智広(nakashima@nri-secure.co.jp)

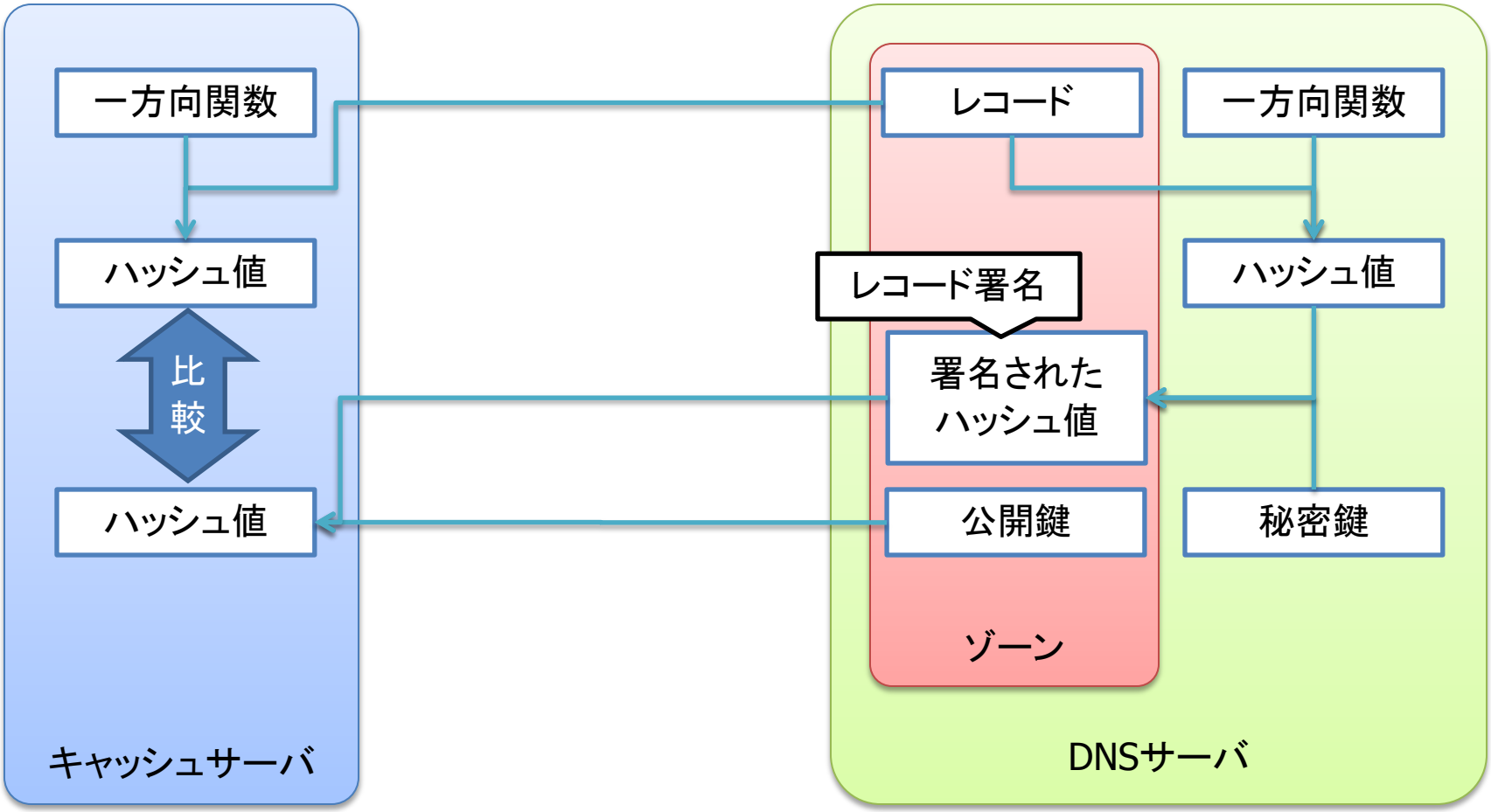
はじめに

- 概要
  - 本発表は煩雑なDNSSEC運用を楽にするための
    ソフトウェアであるOpenDNSSECの概要と導入方法を
    まとめたものです
- おことわり
  - 内容には十分配慮していますが、断片的な情報に基
    づいているため誤りを含んでいる可能性があります
    また、ソフトウェアが現在のところβ版であるため、
    正式版では仕様が変更される可能性があります

発表のながれ

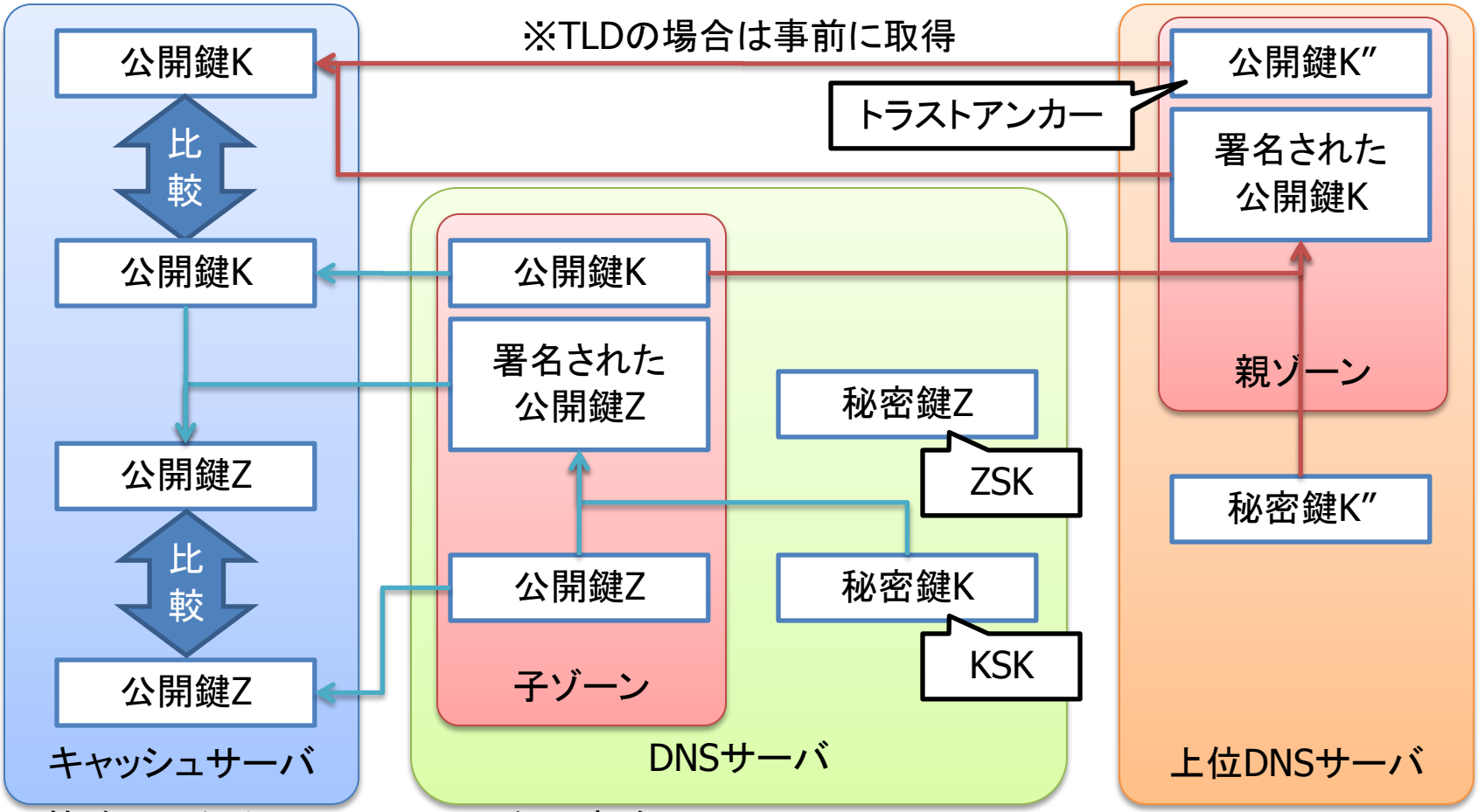
- DNSSECのおさらいと運用における課題の再認識
- OpenDNSSECの概要
- OpenDNSSECの導入チュートリアル

DNSSECのメカニズム(レコードの検証)



レコードから生成したハッシュ値と
レコード署名を公開鍵で復号したハッシュ値を比較し検証する

DNSSECのメカニズム(DS方式)



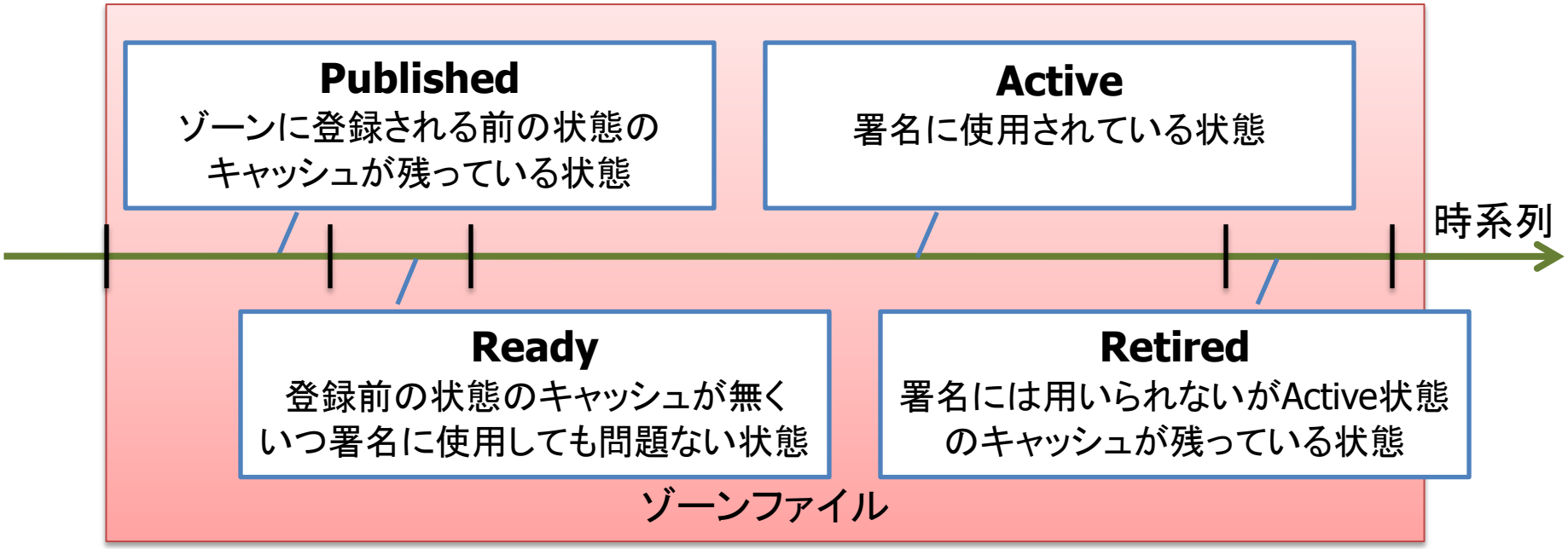
※簡略化のためハッシュ化の過程は省略

DNSサーバのゾーン署名用の公開鍵が真正であることを
別の鍵ペアと上位DNSサーバの鍵ペアを用いて検証する

鍵更新タイミングでの考慮(その1)

NRI SECURE TECHNOLOGIES

- 時系列による鍵の状態遷移



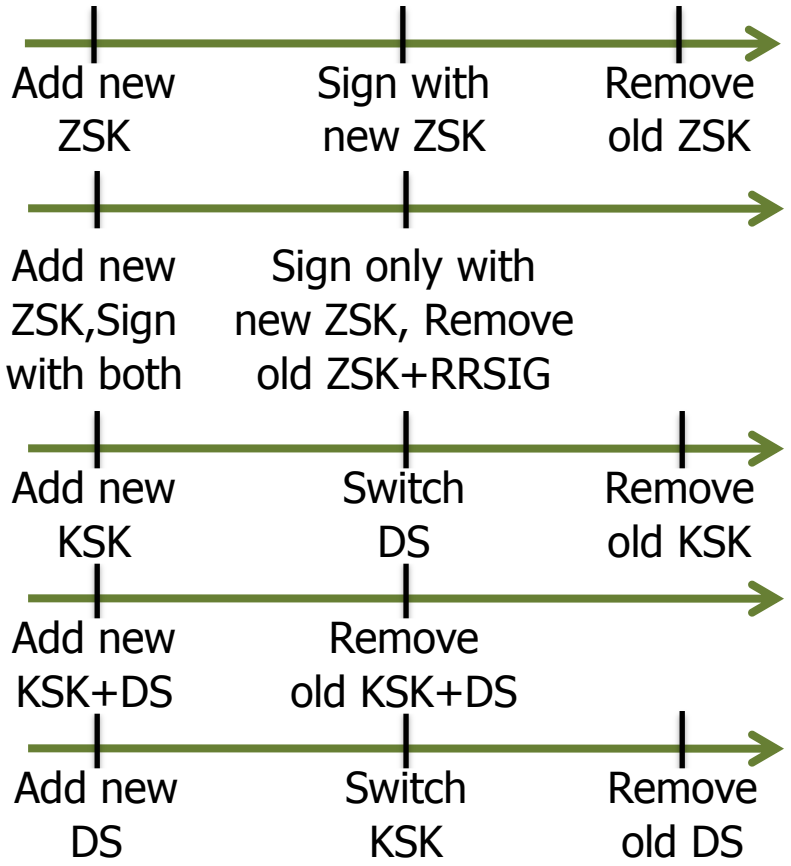
[参考] **DNSSEC Key Timing Considerations**
(draft-morris-dnsop-dnssec-key-timing-01.txt)

キャッシュサーバやリゾルバでキャッシュが保持されるため
鍵更新の際にはTTLとキャッシュ状態の考慮が必要

鍵更新タイミングでの考慮(その2)

NRI SECURE TECHNOLOGIES

- 更新方法のバリエーション
  - ZSK鍵更新時
    - Pre-publication
    - Double Signature
  - KSK鍵更新時
    - Double KSK
    - Double Rrset
    - Double DS



[参考]DNSSEC Key Timing, John A Dickinson, Hohan Ihrenm, Stephen Morris@IETF76

DNSSEC運用の課題

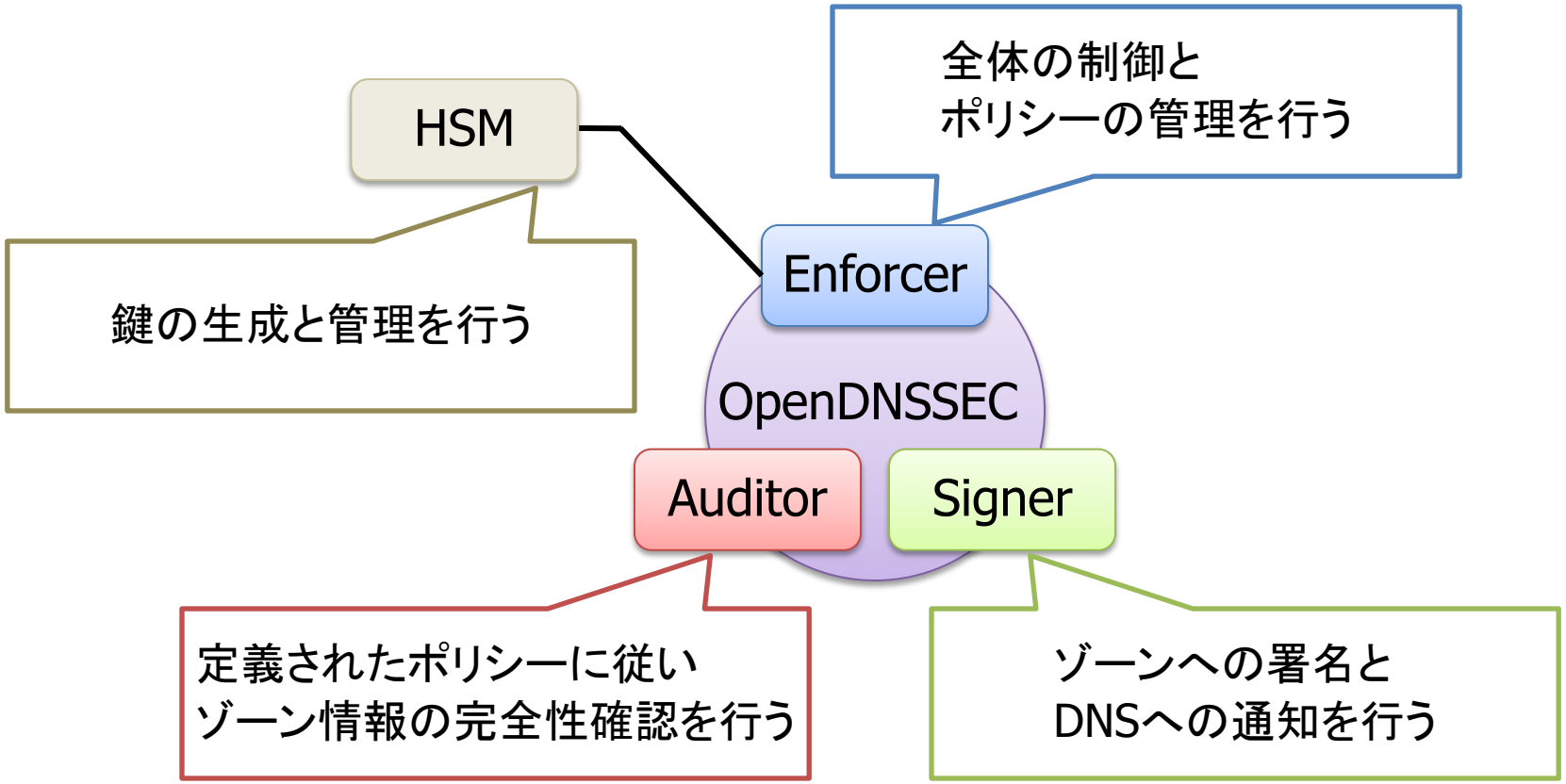
- 2種類の鍵をゾーン毎に管理しなければならない
- 鍵と署名の更新をしなければならない
  - レコードを登録・修正・削除した場合
  - 署名の有効期限が近づいた場合
  - 定期的もしくは緊急に鍵を交換する場合
- 鍵更新時には様々な考慮をしなければならない
- リゾルバで署名の検証が有効の場合
  署名が正しくないと名前解決に失敗する
- 広く普及し正しく運用されなければ意味がない

手順が煩雑かつミスオペレーションが致命的であることから
広く普及するためにはこれら課題の解決策が必要

OpenDNSSECの概要

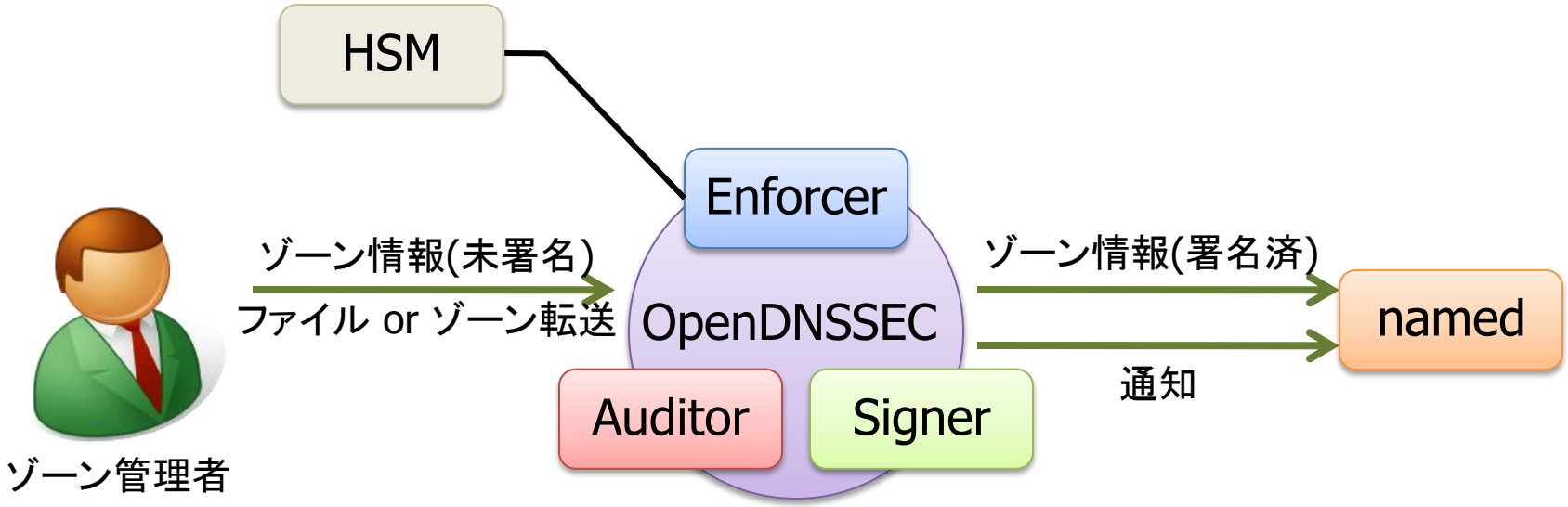
- 煩雑なDNSSEC運用を楽にするためのソフトウェア
  - DNSSEC運用の全プロセスを管理し
    人間の手による作業を削減することでミスを防止
    - 鍵更新とゾーン署名の自動化
    - 署名プロセスとゾーン情報の完全性チェック
    - 手動更新の支援
  - HSM(Hardware Security Module)の利用を前提とし
    鍵の安全性とパフォーマンスを両立
    ※ソフトウェア版HSMもあわせて提供
  - 現在はβ版を公開中(Ver.1.0.0b7)
    正式版は2010年リリース予定
  - オープンソース(BSDライセンス)
  - nominet, NLnet Labs, .se,
    SURFnet, Kirei, John A Dickinsonなどが参加

OpenDNSSECの構成



OpenDNSSECの3つの機能とHSMが連動して
適切なDNSSEC運用を実現する

OpenDNSSECの利用イメージ



ゾーン管理者はDNSSECの鍵や署名を意識することなく
これまでと変わらず未署名のゾーンを管理すればよい

デモ

1. OpenDNSSECの動作デモ
2. OpenDNSSECを用いた手動での署名と鍵の更新
3. さらにもう一度鍵更新を試みた場合の動作
4. Auditorとnamed-checkzoneの違い

導入までの流れ

1. 必要なモジュールのインストール
2. OpenDNSSEC/SoftHSMのインストール
3. OpenDNSSEC/SoftHSMの設定
4. OpenDNSSECの起動

ハードウェア版HSMを用いる場合はSoftHSMは導入不要

インストール

- OpenDNSSEC/SoftHSM
  1. アーカイブファイルをダウンロードして展開
  2. ./configure; make; make install
- 必要モジュール
  - ldns
  - libxml2, libxml2-dev, libxml2-utils
  - ruby, rubygems, dnsruby, libopenssl-ruby
  - sqlite3, libsqlite3, libsqlite3-dev
  - python, python-4suite-xml

必要とされるモジュールのバージョンが新しく
OS標準のパッケージではそろわないものが多いため面倒

設定ファイルの概要

- conf.xml
  - OpenDNSSEC全体の設定
- kasp.xml
  - 鍵と署名のポリシー設定
- zonelist.xml
  - ゾーンとゾーンファイルの設定
- zonefetch.xml
  - ゾーン情報をゾーン転送で取得する場合の設定(未実装)
- softhsm.conf
  - SoftHSMのリポジトリを設定

※設定ファイル中の時間の記述はISO8601形式を使用

conf.xml(一部抜粋)

```
<RepositoryList><Repository name="softHSM">
                    <Module>/usr/local/lib/libsofthsm.so</Module>
                    <TokenLabel>OpenDNSSEC</TokenLabel>
                    <PIN>1234</PIN>
</Repository></RepositoryList>
<Enforcer>
      <Datastore><SQLite>/var/opendnssec/kasp.db</SQLite></Datastore>
       <Interval>PT60S</Interval>
 </Enforcer>(省略)
 <Signer>(省略)
       <NotifyCommand>/usr/local/bind9.7/sbin/rndc -k
/var/named/chroot/etc/rndc.key reload %zone</NotifyCommand>
</Signer>
```

- 環境に合わせて全体の設定を記述
  - HSMを利用するための設定
  - 動作間隔
  - ゾーン情報変更時のnamedへの通知コマンド

kasp.xml(一部抜粋)

```
<Signatures>
      <Resign>PT2H</Resign>
      <Refresh>P3D</Refresh>(省略)
</Signatures>
<Keys> (省略) </Keys>
<Denial><NSEC3>(省略)<Hash>
          <Algorithm>1</Algorithm> (省略)
 </Hash></NSEC3>(省略)</Denial>
<Zone>(省略)
       <SOA>
          <Serial>datecounter</Serial>
       </SOA> (省略)
</Zone>
```

- 環境に合わせて鍵と署名のポリシーを記述
  - 鍵や署名の時間パラメータ
  - 暗号やハッシュのアルゴリズム
  - SOAレコードのSerialのインクリメントルール

zonelist.xml(一部抜粋)

```
<Zonelist>
        <Zone name="example.jp”>
                <Policy>default</Policy>
<SignerConfiguration>/var/opendnssec/signconf/example.jp.xml</Signer
Configurration>
                <Adapters>
                        <Input>
                                <File>/var/opendnssec/unsigned/example.jp</File>
                        </Input>
                        <Output>
                                <File>/var/opendnssec/signed/example.jp</File>
                        </Output>
                </Adapters>
        </Zone>
</ZoneList>
```

- 直接編集せず付属ユーティリティ使用を推奨
  - ods-ksmutil zone add -z example.jp -p default

※SignerConfigurationは初期設定段階では存在しない

softhsm.conf

```
# softHSM configuration file
#
0:/var/softhsm/slot0.db
```

- スロットの設定
  ※ 設定段階ではdbファイルは存在しない

初期設定

- SoftHSMの初期設定
  - softhsm --init-token --slot 0 --label ”OpenDNSSEC”
    ※softhsm.conf, conf.xmlとあわせる
- OpenDNSSECの初期設定(DBへの設定反映)
  - ods-ksmutil setup
    ※更新はods-ksmutil update

使用方法

- OpenDNSSECデーモンの起動/停止
  - ods-control start /stop
- 署名の手動更新
  - ods-signer sign example.jp
- 鍵の状態を確認
  - ods-ksmutil key list
- 鍵の手動更新
  - ods-ksmutil key rollover -z example.jp --keytype zsk
  - ods-ksmutil key rollover -p default --keytype ksk
- 動作はログ出力で確認

まとめ

- OpenDNSSECで実現できること
  - DNSSEC運用の自動化
  - ミスオペレーションの防止
  - ポリシーに則った適切なDNSSEC運用
- OpenDNSSEC導入に必要なこと
  - 鍵と署名のポリシーの定義
  - ゾーン情報とゾーンファイルの対応の定義
- DNSSEC運用における残課題
  - 親ゾーンへのKSK公開鍵情報(DS)の登録

BINDのツール群を組み合わせた運用も一つの方法ですが
OpenDNSSECを導入してみるのはいかがでしょうか？

参考情報

- 公式Web(Wiki, Mailinglist Archive)
  - http://www.opendnssec.org
- DNSSEC Key Timing
  - http://www.ietf.org/proceedings/09nov/slides/dnsop-8.pdf
- DNSSEC Key Timing Considerations
  - http://tools.ietf.org/id/draft-morris-dnsop-dnssec-key-timing-01.txt