RPKI routing WG報告
～RPKIルーティングを試す会～

JANOG32 事前公開資料

RPKIルーティングを試す会とは

- WGについて
  - RPKI(Resource PKI)を使ったルーティングに関わる現在の実装状況とそして今後どのような運用の形になっていくかをテーマとするWGです。主にRPKIの実装を動かしてみることを通じて構造や運用のポイントなどについて情報共有することを目標とします。
- 期間
  - 2013年1月～2013年7月
- 予定される成果物
  - 実装の現状と今後の課題に関する報告
- チェア
  - 吉田友哉、木村泰司
- メーリングリスト
  - https://www.janog.gr.jp/mailman/listinfo/rpki-routing-wg/

活動しました

	日時	やったこと
RPKIハッカソン	2013年1月21日（水）@IIJ 2013年2月20日（水）@JPNIC	ツール（RPKI Tools）とBGPルーター実装の設定に向けたハッカソン
RPKIハンズオン	2013年4月26日（金）in ENOG20 2013年5月27日（木）in 第4回電力系勉強会	RPKIを使ったOrigin Validationの体験ハンズオン
（おまけ）RPKIハッカソン/BoF in APNIC	2013年2月27日（水）in APNIC35@シンガポール	AP地域におけるRPKIの勉強会＋課題整理

活動成果として

- RPKIを構築し使うためのツールの現状
  - RPKI Toolsを使ってJPNICのRPKIの認証局システムを設定してみました
  - JPNICとしてのRPKIの構築から、ユーザサイド（BGPルーターでの参照）までの一連の動作は確認できた。ただし、、
- 今後の課題と論点
  - RPKIを使う業務（ROAの発行など）はGUIを使う分にはできそう
  - ASのオペレーターとIPアドレスの担当者の業務をどう結びつけるのか、IRRの登録業務と同時にやらないといけないのか
  - IXにおけるRPKIの運用はどうなるのか、パンチングホールは扱えるのか

    などの課題が見えてきた。。一体どういうことなのか。

JANOG32ミーティング

RPKIを試す会報告
～ルーティングへの活用 概要偏～

一般社団法人日本ネットワークインフォメーションセンター

岡田 雅之＜okadams@nic. ad. jp＞

ハッカソン→ハンズオン ？

- ハッカソン

・参加者と開発者が一緒に構築・ビルドしながら問題や利用者視点の課題を共有し、参加者はシスの知識を取得し、開発者は未知のトラブル把握や課題を把握、解決につなげる

・参加に関するのハードルが高く、なかなか利用者としての議論まで到達が困難

平易にRPKIを体験してもらい、体験を通じて感じたこと・課題を議論するためにハンズオン形式とした。

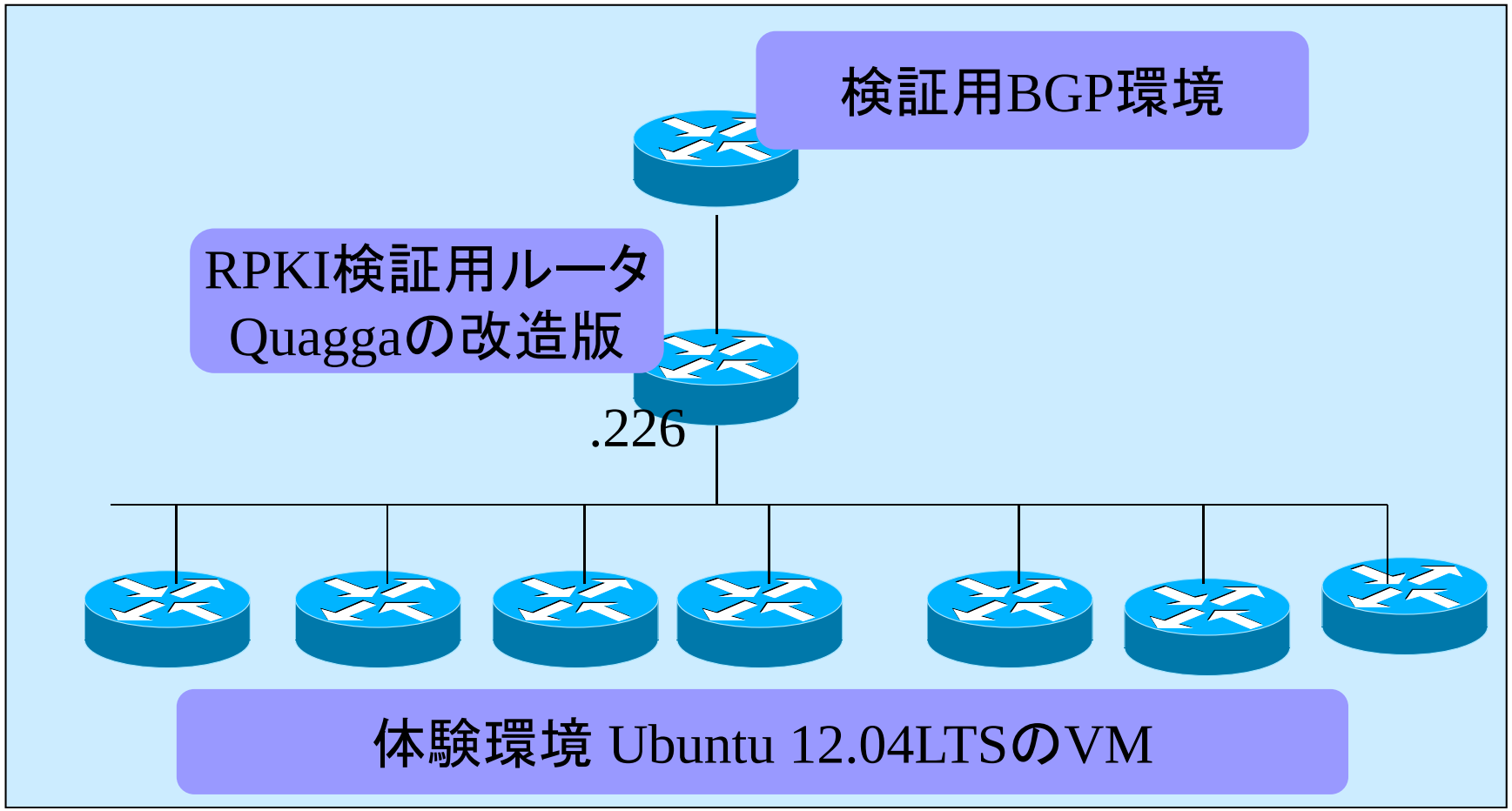
これまでに行ったハンズオンの概要

- 発行したROAの活用を体験

・ROAの収穫とROAの中身の確認
・ルータに食わせるまでの加工過程
・ルータでのROAはどのように見えるのか

実体験を通じて、RPKIとルーティングの関係を感じてもらい、問題点や改善すべき点の議論へつなげる

ハンズオン環境の概要



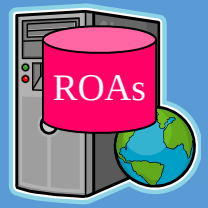
参加者はVMにインストールされたRPKIツールとQuaggaRPKI拡張版を利用

RPKIフレームワークとハンズオン環境

証明書公開場所

証明書発行組織（ｘIR,xNIC)

ROA利用者

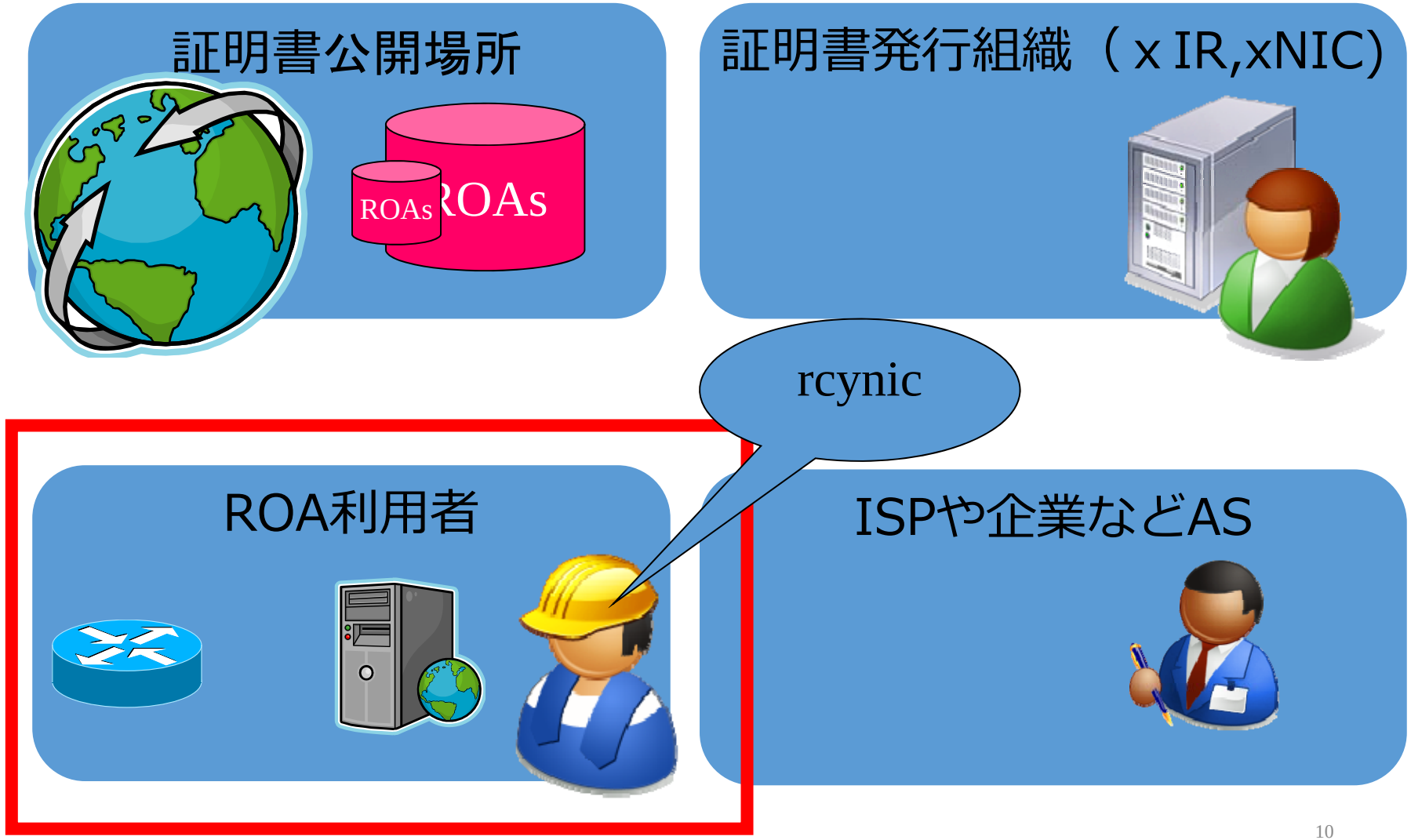


ISPや企業などAS

ルータの利用しやすい形への加工過程

1. ROAの収集
   - 有効なROAを収集
   - 無効なROA(証明書の期限切れ、IPアドレス範囲外)除外
2. ROAの加工
   - ROAからルータが必要とする情報を抽出
     - IP PrefixとAS番号を取り出す
   - 署名関連の情報などルータが必要としない情報をカット
3. ROAを参照したルーティング
   - ROAキャッシュを利用したルーティング
   - RPKI-RTRプロトコルによるルータとキャッシュ間通信

工程１ ROAの収集


証明書公開場所
ROAs
ROAs
証明書発行組織（ｘIR,xNIC)
rcynic
ROA利用者
ISPや企業などAS

工程１　ＲＯＡの収集

1.　RPKIコマンド・rcynicの実行

```
$ sudo -s
$ cd /var/rcynic
$ rcynic –c /var/rcynic/etc/rcynic.conf –l log_debug –j 1
```

rcynic: rsyncのラッパー、rsyncをしながら証明書の有効性の検証もします。
（気合をいれれば人力でも可能）

工程１　ＲＯＡの収集

1.　取得したROAの確認（≒rsyncで収集）

```
cd /var/rcynic/data
$ find
./authenticated
./authenticated.2013-04-25T03:21:24Z
./authenticated.2013-04-24T13:31:36Z
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03.cer
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/root.crl
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02/2
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02/2/
psc6rX9nKjhG4bcPfphMiJX-zKE.mft
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02/2/
psc6rX9nKjhG4bcPfphMiJX-zKE.crl
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02/2
/LcbRde5eNd9uTzS3ki3YzDDydJU.gbr
./authenticated.2013-04-24T13:31:36Z/rpki02.nic.ad.jp/publication/JPNIC03/JPNIC-OFFICE-02/2/
zWT4gjPUOgKwE13Zse_rEgOunEE.roa
./authenticated.2013-04-24T13:31:
```

有効な情報がauthenticatedディレクトリに入る。
有効でない情報はunauthenticatedディレクトリへ。

工程1 ＲＯＡの収集

1. ROAのファイル

```
cd /var/rcynic/data
$ cd hogehoge
./apsc6rX9nKjhG4bcPfphMiJX-zKE.mft
./psc6rX9nKjhG4bcPfphMiJX-zKE.crl
./zWT4gjPU0gKwE13Zse_rEg0unEE.roa
```

.mft：マニフェストファイル：ユーザの発行した証明書の一覧。（ヌケモレ防止）
.crl：証明書破棄リスト
.roa：ROAの実体

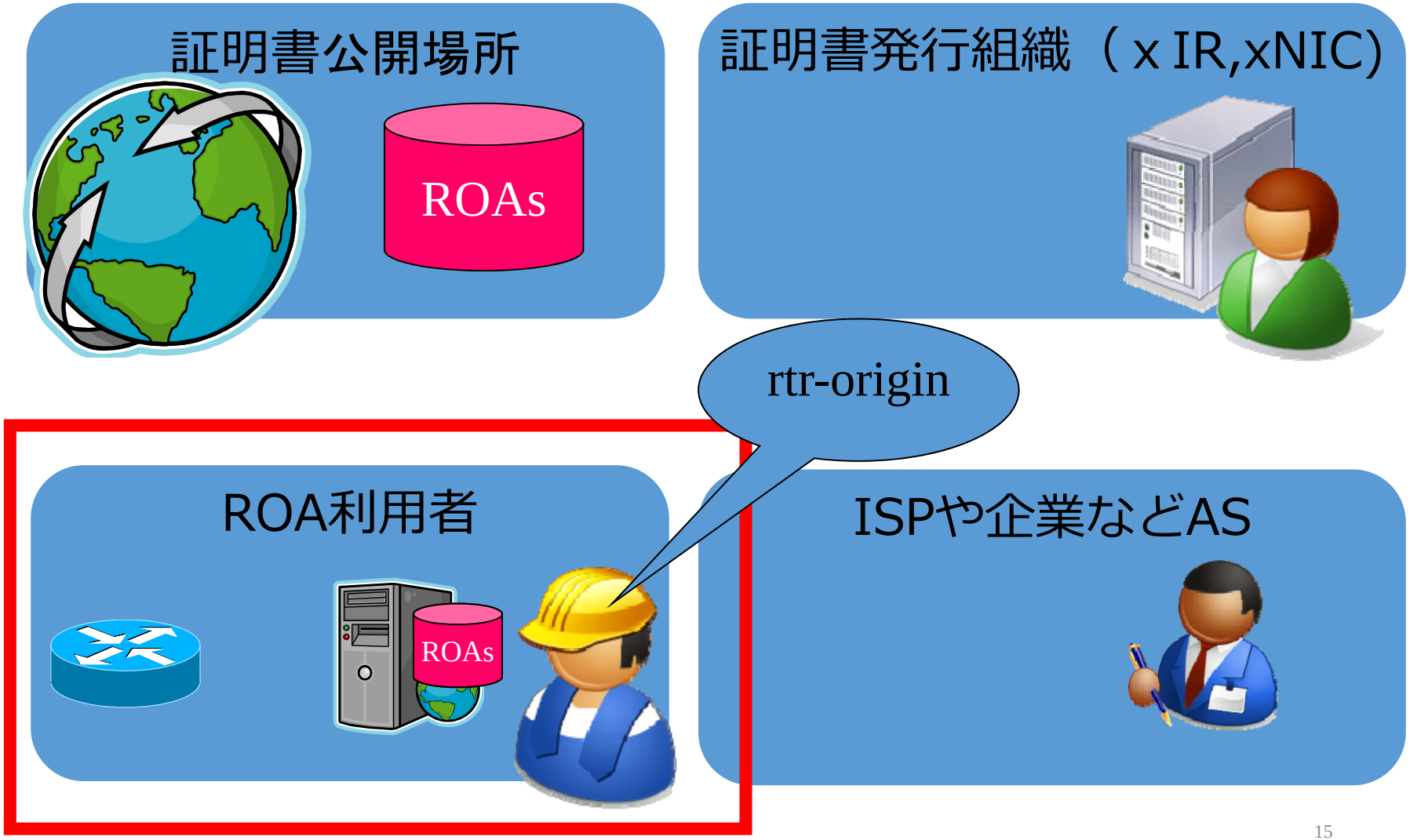
工程1 ＲＯＡの収集

1. ROAの中身

```
cd /var/rcynic/data/authenticated/rpki01.nic.ad.jp/repository/JPNIC02…
$ print_roa zWT4gjPU0gKwE13Zse_rEg0unEE.roa
Certificates:   1
CRLs:           0
SignerId[0]:    0f:9f:bb:7c:2a:35:6d:19:30:35:14:6b:d8:b9:3b:2f:cb:31:bd:
d4 [Matches certificate 0] [signingTime(U) 130424134202Z]
eContentType:   1.2.840.113549.1.9.16.1.24
version:        0 [Defaulted]
asID:           2515
 addressFamily: 1
     IPaddress: 202.12.30.0/24-25
```

ROAの中はIPアドレス、AS番号、最長マスク長などが入っている。

工程２ ROAの加工


証明書公開場所
ROAs
証明書発行組織（ｘIR,xNIC)
rtr-origin
ROA利用者
ROAs
ISPや企業などAS

工程２　ＲＯＡの加工

1.　ルータの必要な情報へ加工

```
$ cd /var/rpki-rtr
$ rtr-origin --cronjob /var/rcynic/data/authenticated
```

rtr-originはキャッシュ作成とキャッシュ応答、の二つの異なる機能を持つ
cronjobはキャッシュ生成を行う。
カレントディレクトリにキャッシュファイルを生成し、デーモンへ合図を送る
引数でキャッシュを生成するROAの対象を選択（unauthenticateを、、、）

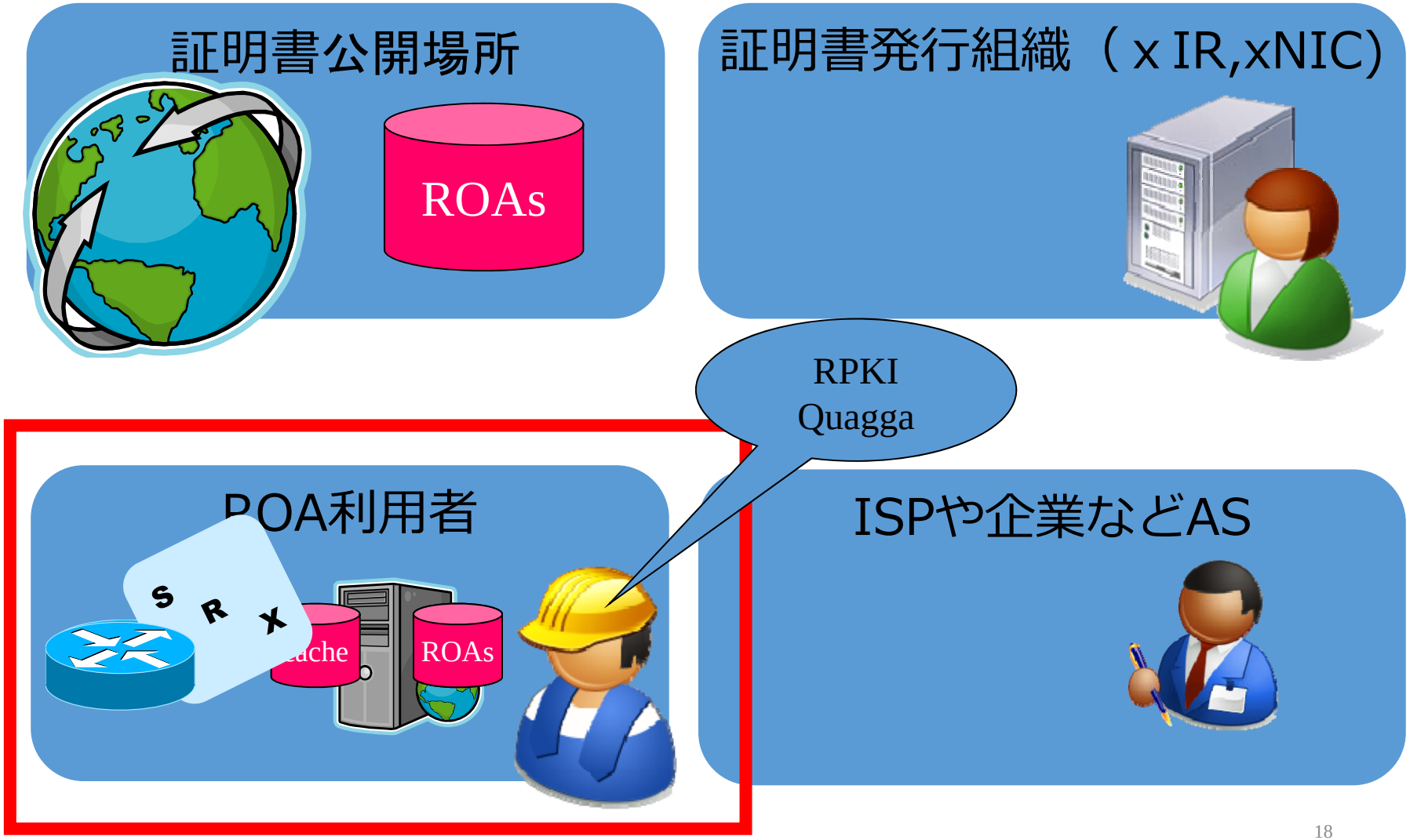
工程２　ＲＯＡの加工

1.　加工した内容の確認

```
$ rtr-origin --client tcp localhost 42420
2013-04-25 04:54:05 rtr-origin/client[896]: +    2518  192.47.117.0/24-24
 00:04:00:00:00:00:00:14:01:18:18:00:C0:2F:75:00:00:00:09:D6
2013-04-25 04:54:05 rtr-origin/client[896]: +   45674  192.47.117.0/24-24
 00:04:00:00:00:00:00:14:01:18:18:00:C0:2F:75:00:00:00:B2:6A
2013-04-25 04:54:05 rtr-origin/client[896]: +    2515  202.12.30.0/24-24
00:04:00:00:00:00:00:14:01:18:18:00:CA:0C:1E:00:00:00:09:D3
2013-04-25 04:54:05 rtr-origin/client[896]: [end_of_data, serial #1366793945 nonce 62065]
```

rtr-origin clientモードはデーモンにキャッシュの内容を問い合わせる
(実はxinetdからrtr-originデーモンモードを起動し、キャッシュファイルを表示)

工程３ ROAの活用


証明書公開場所
ROAs
証明書発行組織（ｘIR,xNIC)
RPKI
Quagga
ROA利用者
S
R
X
ache
ROAs
ISPや企業などAS

工程３　ＲＯＡの活用

```
$ cd /usr/local/sbin/
$ ./bgpd –f ../etc/bgpd.conf &
```

Quagga SRXがすでにインストールされ起動できます。

工程３　ＲＯＡの活用

```
bgpd# sh ip bgp
BGP table version is 0, local router ID is 192.41.192.249
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Validation:    v - valid, n - notfound, i - invalid, ? - undefined
SRx Status:    I - route ignored, D - SRx evaluation deactivated
SRxVal Format: validation result  (origin validation, path validation)
Origin codes: i - IGP, e - EGP, ? - incomplete

   Ident     SRxVal SRxLP Status Network          Next Hop            Metric  LocPrf Weight Path
*> 576815A8 n(n,-)  + 100,      198.1.0.0/21     192.41.192.226             100s      0 65001 2515 2497 69
*> EC7A62E2 n(n,-)  + 100,      198.1.16.0/21    192.41.192.226             100s      0 65001 2515 2497  i
*> E2D7A444 n(n,-)  + 100,      198.1.24.0/23    192.41.192.226             100s      0 65001 2515 2497 i
*> 26095C25 n(n,-)  + 100,      198.1.28.0/22    192.41.192.226             100s      0 65001 2515 2497 i
*> 6454466E n(n,-)  + 100,      198.1.32.0/20    192.41.192.226             100s      0 65001 2515 2497 i
*> 1A01CA5C n(n,-)  + 100,      198.1.32.0       192.41.192.226             100s      0 65001 2515 2497 6 i
```

SRxValがValidation結果です。
n:notfound v:valid i:invalidとなります。(n,-)は将来への予約です。

工程３　ＲＯＡの活用

```
*> 29C78D0D n(n,-) +100, 202.12.29.0 192.41.192.226      100s      0 65001 2515 i
*> DE83681B v(v,-) +200, 202.12.30.0 192.41.192.226      200s      0 65001 2515 i
*> FBF4BE57 n(n,-) +100,202.12.31.0 192.41.192.226      100s      0 65001 2515 i
```

202.12.30.0/24はAS2515のROAが存在するため V となっています。
先ほどのConfigではValidationStatus VのときLPを＋200するとしています。

議論：なぜRPKIが必要であるのか？

- IRRではカバーできない点
  - アドレスホルダーとRouteオブジェクトの関係
    - 残念ながらRADbなどは任意のIP Prefixを登録可能
  - Routeオブジェクトの欠落などの検証が不可
    - RouteやASの抜け漏れの検知手段が存在しない
  - Routeオブジェクトやさまざまなオブジェクトの改ざんが可能
    - ミラーや保管時の改ざん検知の仕組みが存在しない
- そもそもIRRは緩いシステム
  - 信じるも信じないも参照者・登録者の意思
  - システム全体の可容性もルーティングシステムと比較し低い
- さて、RPKIではどのように改善されるのであろう