Local Management of Trust Anchors (for the RPKI)

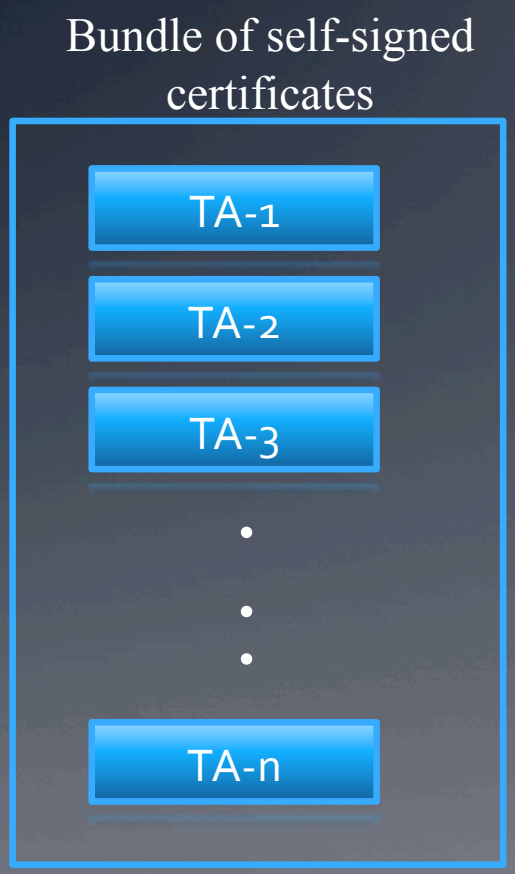
Stephen Kent

BBN Technologies

Local TA Management

- In principle, every RP should be able to locally control the set of TAs that it will employ
- In practice, most PKI applications do not provide good, local TA management capabilities
- The common form of a TA, a self-signed certificate, also limits a user's ability to impose constraints on it and on subordinate certificates
- The mechanisms described here are from a document that is now a WG item in SIDR, to provide a local TA management capability for the Resource PKI (RPKI)

Typical TA Configuration



- In common practice, each self-signed certificate contains no extensions that constrain it
- 5280 path validation algorithm would not impose such constraints, because they appear in a TA
- The “scope” of each TA may overlap, e.g., re name constraints
- A confusing (& dangerous) model for RPs

Our Model: The RP is the TA!

- The model we are pursuing calls for each RP to recognize exactly one TA, itself!
- The RP imports putative TAs (in the form of self-signed certificates) and re-issues them under itself
- In so doing, the RP is empowered to insert constraints
- Because these certificates are now one step below the TA, normal certificate path processing (ala 5280) will impose those constraints (e.g., path length, policy, name, and RFC 3779)

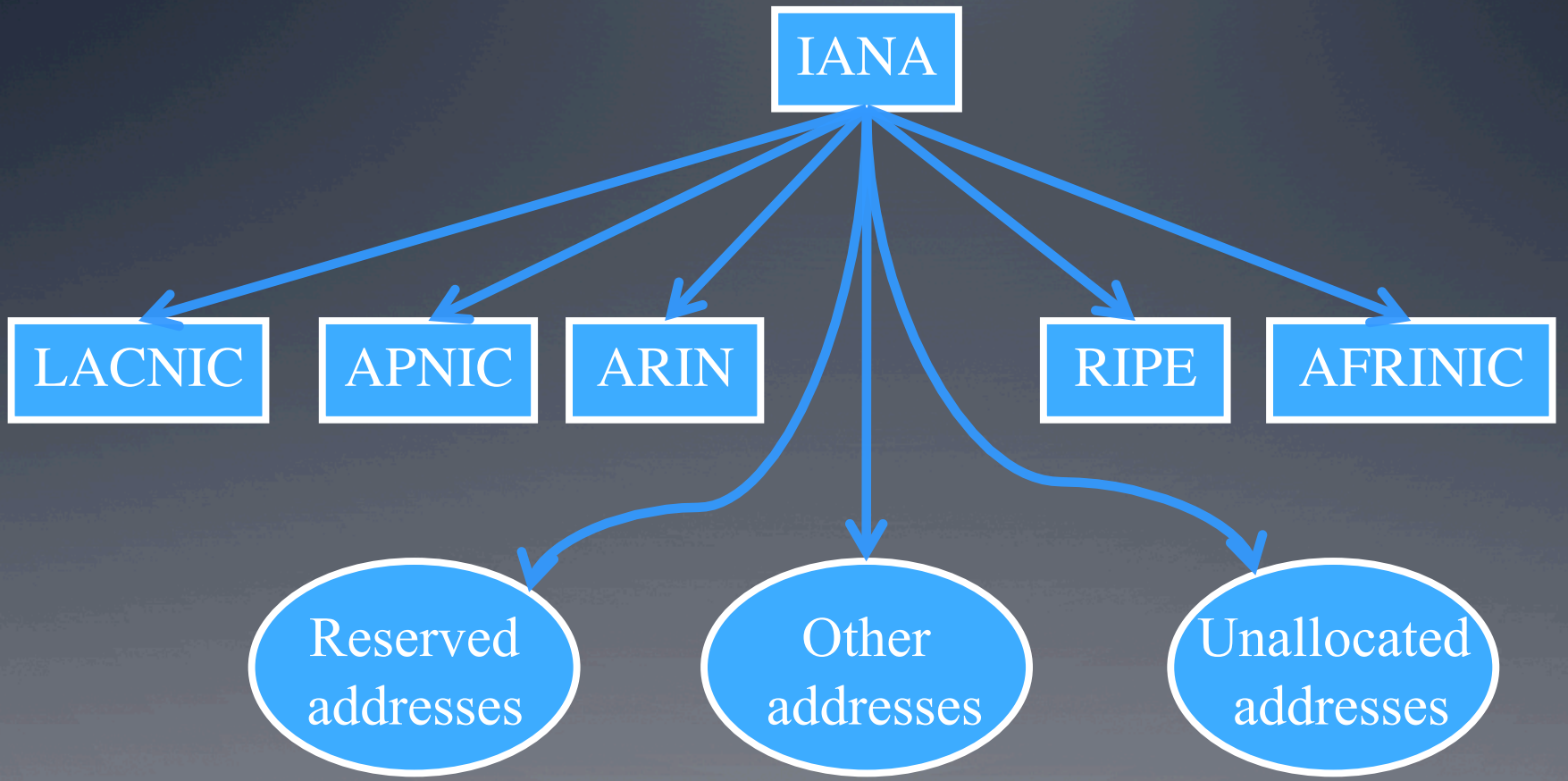
Why the RPKI Needs This

- The RPKI makes use of certificates (CA and EE) that contain Internet Number Resource (INR) extensions, as defined in RFC 3779
- The validation algorithm defined for these extensions requires that a path conform to subset rules, analogous to the name constraints extension, all the way to the TA for the path
- 3779 extensions imply that the RPKI is a hierarchy
- Some RPs may need to override the RPKI nominal hierarchy, e.g., to deal with RFC 1918 addresses, or for security reasons

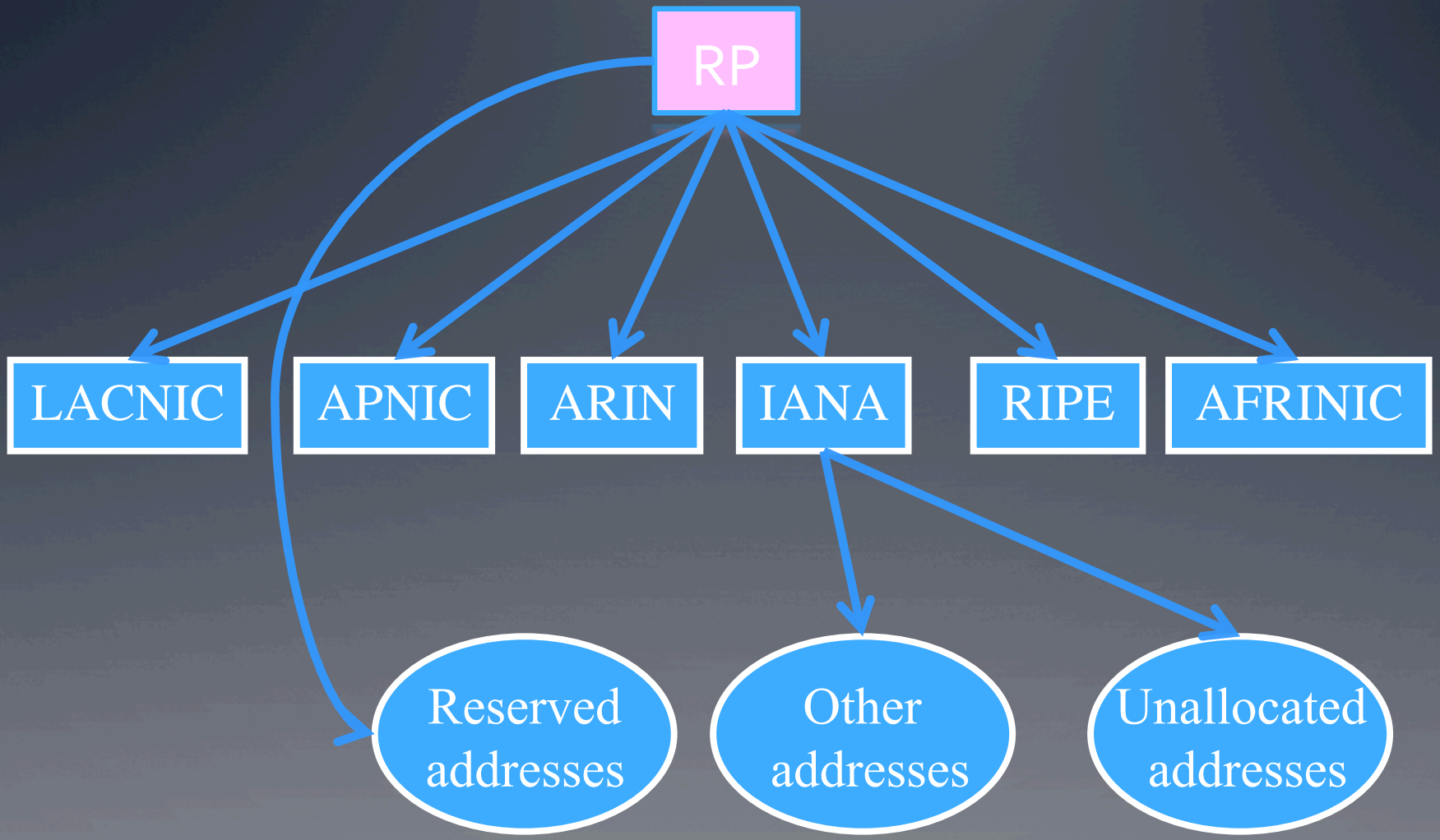
Making this Work in the RPKI

- We will need to be able to create new certificates, often with modified RFC 3779 extensions
- To make this work
  - The RP's TA certificate must contain RFC 3779 extensions encompassing all addresses and all AS#s
  - The RP TA re-issues certificates with new 3779 extensions
    - Delete overlapping 3779 data as needed
    - Re-issuing targeted certificates directly under the RP TA
    - Re-parenting ancestors of re-issued certificates under the RP TA
  - The RP can also override certain fields of the re-issued certificate using a "constraints file"

An RPKI TA Example



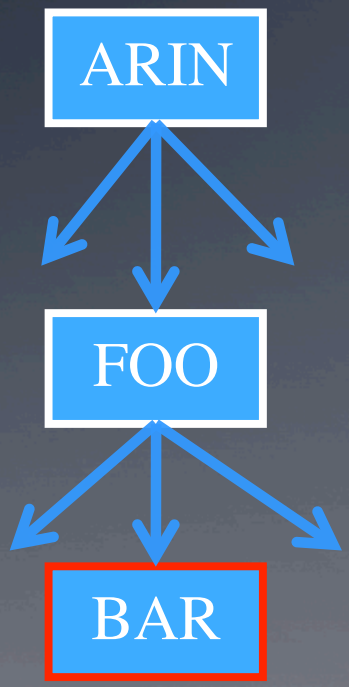
RPKI with Local Control



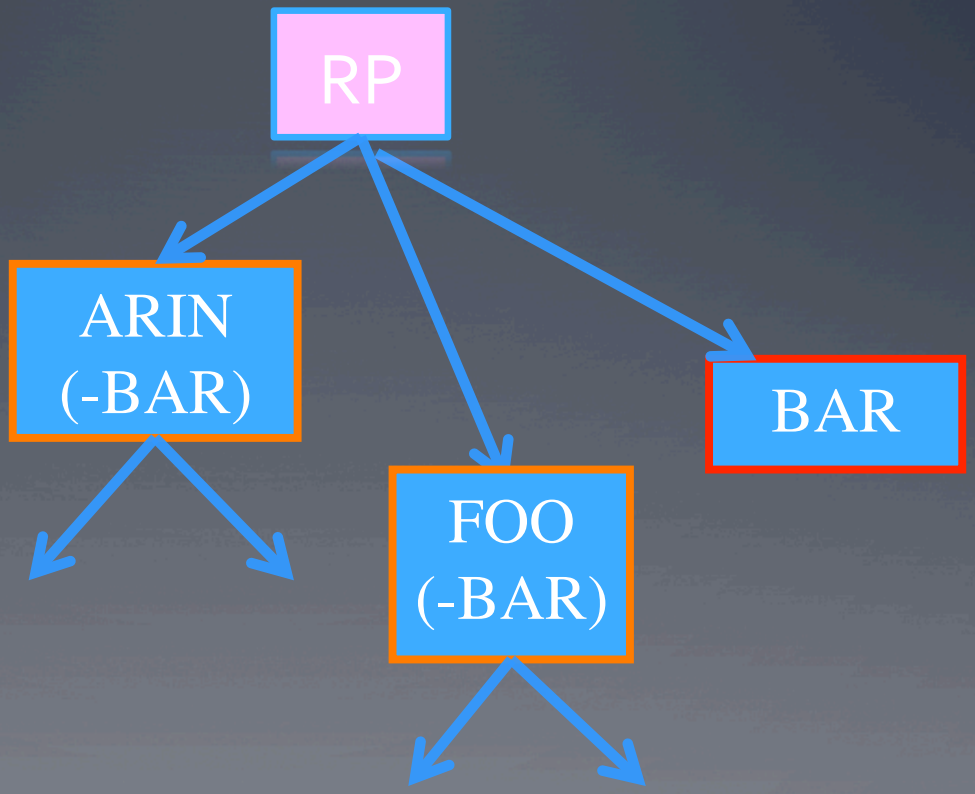
(RP wants to make use of 10/8 for local routing)

A More Detailed Example

As offered by ARIN



As managed by an RP



(RP trusts its own knowledge of BAR's address allocation and does not want any action by ARIN or FOO to override that knowledge)

Elements of the Solution

- Constraints file
- Resource re-writing algorithm
  - Target processing
  - Ancestor processing
  - Tree processing
  - TA re-homing
- Path discovery
- Revocation
- Expiration

Constraints File

- The RP creates (or acquires from an authoritative source) a constraints file specifying IP address space and AS# resources for target certificates
  - Certificates are specified by SKI, thus the constraints file must be updated when the targets rekey
- The constraints file also allows the RP to control rewriting certain fields in the re-issued certificates
  - Validity dates
  - CRLDP
  - AIA
  - Policy Qualifier OID

Resource Rewriting Algorithm

- There are four stages to the algorithm
  - Target processing
    - Certificates that match a given SKI have their resources rewritten to those specified in the constraints file
  - Ancestor processing
    - Ancestors of targets are processed to ensure RFC 3779 rule compliance (remove target certificate resources)
  - Tree processing
    - The entire tree of certificates is searched, and certificates with resources that conflict with any target resources are modified to remove the conflict
  - TA re-issuing
    - All TAs in the original hierarchy are re-issued under the RP's TA

Implications of this Model

- This algorithm creates two parallel hierarchies: the original certificate hierarchy and the para-certificate hierarchy
- There are implications for path discovery, since a certificate can now have an original parent and a para-parent
- There are implications for revocation
- There are also implications for expiration, since the constraints file allows rewriting the validity interval of para-certificates

Path Discovery

- Path discovery prefers the para-certificate hierarchy
- If a certificate has a para-parent, that para-parent will be used to form the certificate path
- If the certificate has only an original parent, but that parent was a target specified in the constraints file, or an ancestor of such a target, then path discovery fails
  - This can occur if the RP has revoked the para-certificate, the original certificate is still present, and the Local TA tool has not yet been run to regenerate the para-certificate
- If the certificate has only an original parent, and the parent is not a target, or the ancestor of a target, path discovery can proceed up the original chain

Revocation

- The original hierarchy and the para-hierarchy are disjoint; revocation of a certificate in one does not affect the other
- Para-certificates are all issued by the RP, so only the RP can revoke them
- Original certificates can still be revoked by their issuers
- Because of the modified path discovery rule, revocation of any para-certificate will cause path discovery to fail until the para-certificate has been replaced or regenerated

Expiration

- The constraints file allows the RP to specify notBefore and notAfter for all para-certificates
  - This is a global rewrite rule, not a per-certificate rewrite rule
- As a result, expiration of the original certificate does not necessarily imply that the corresponding para-certificate expires at the same time
- Expiration of a para-certificate affects path discovery in the same way as revocation of a para-certificate

Questions?