# Local Management of Trust Anchors for the RPKI (document status update)

Stephen Kent

BBN Technologies

# Local TA Management

- I’ve discussion this I-D in previous meetings (since March, 2010!)
- We recently posted the -05 version of the document, but it was just a refresh of the -04 version
- I am making a number of edits to the -05 version, to improve readability, but no substantive changes so far
- The one substantive change I can envision is to allow the global flags defined in the “tags” section to also be asserted on a per-target basis in the “blocks” section

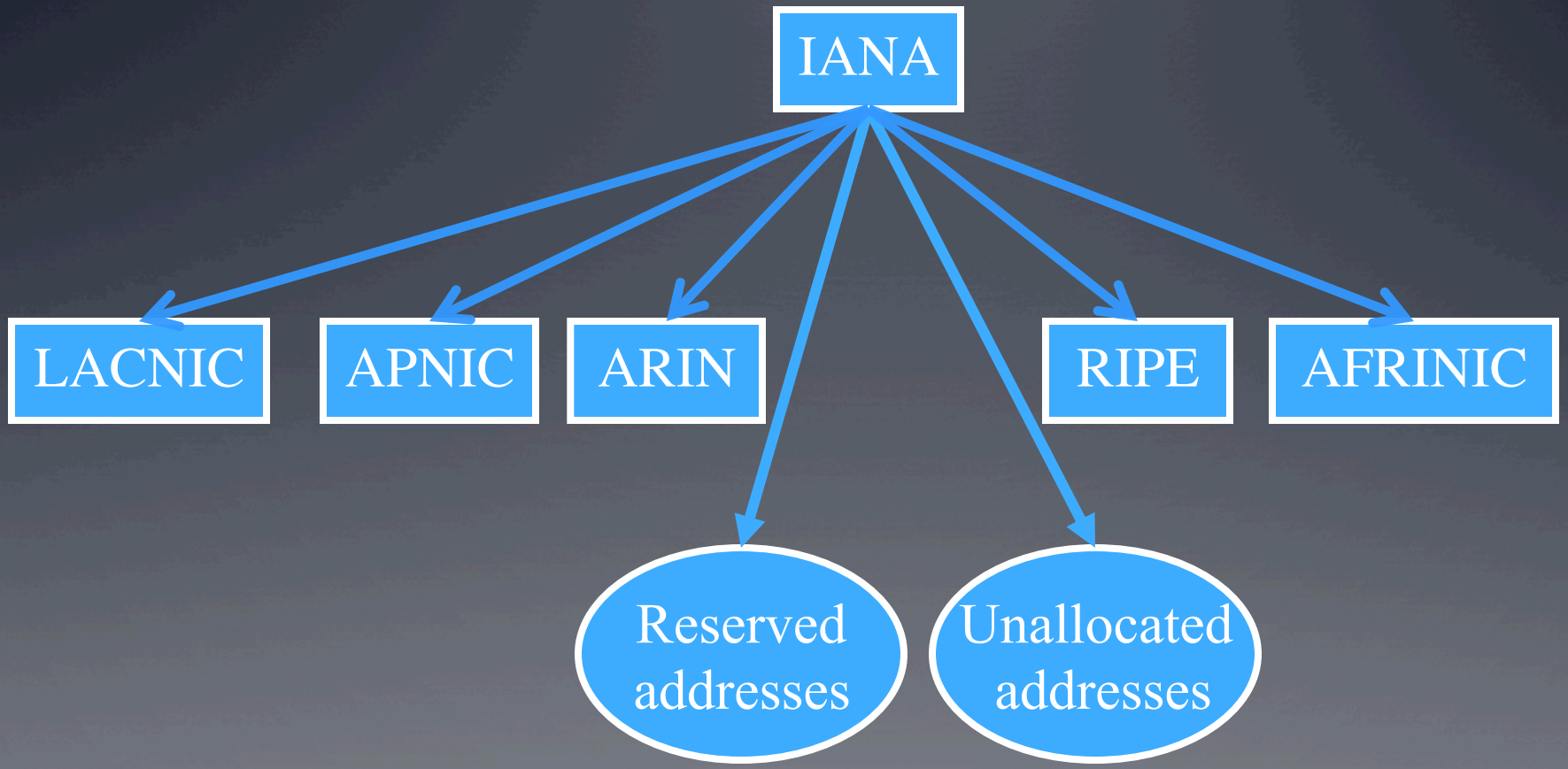
# The Model: The RP is the TA!

- The model we propose calls for each RP to recognize exactly one TA: itself
- The RP imports the putative TAs (typically in the form of self-signed certificates) and re-homes them under itself
- It also allows import of “targeted” certificates (referenced by SKIs) also rehomed under the local TA, using a “constraints” file
- The RP can thus override the RPKI nominal hierarchy, represented in the repository system, as needed

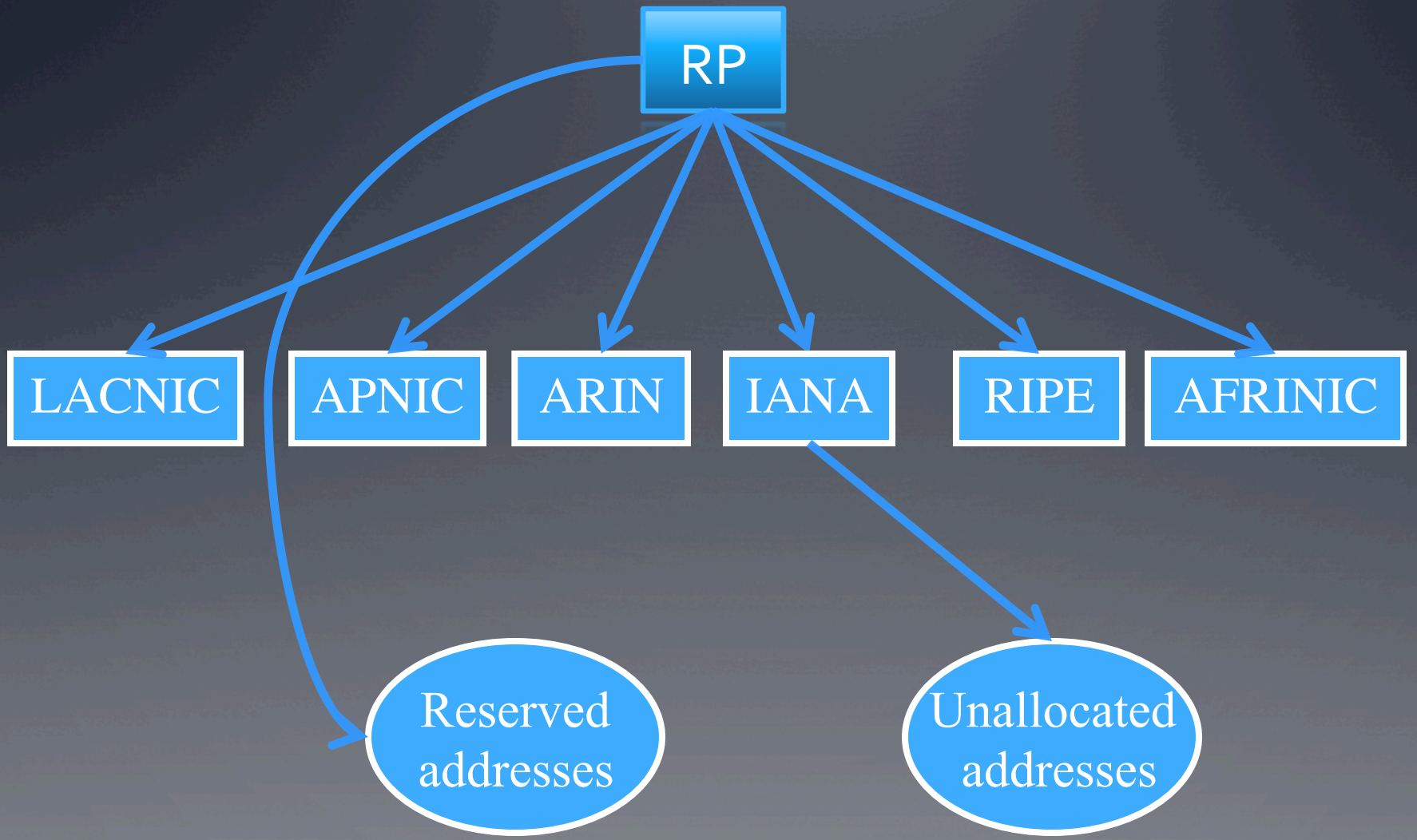
# Making this Work in the RPKI

- An RP must be able to create new certificates, often with modified RFC 3779 extensions
- To make this work
  - The self-signed RP certificate must contain RFC 3779 extensions encompassing all addresses and all ASNs
  - The RP re-issues certificates with new 3779 extensions to override the RPKI tree
    - Delete overlapping 3779 data as needed
    - Re-homing targeted certificates under the RP TA
    - Re-homing ancestors of rehomed certificates under the RP TA
  - The RP can also override certain fields of the re-issued certificate using the constraints file

# An RPKI TA Example



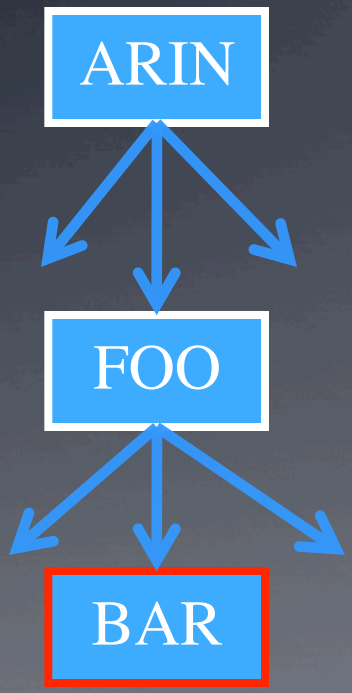
# RPKI with Local Control



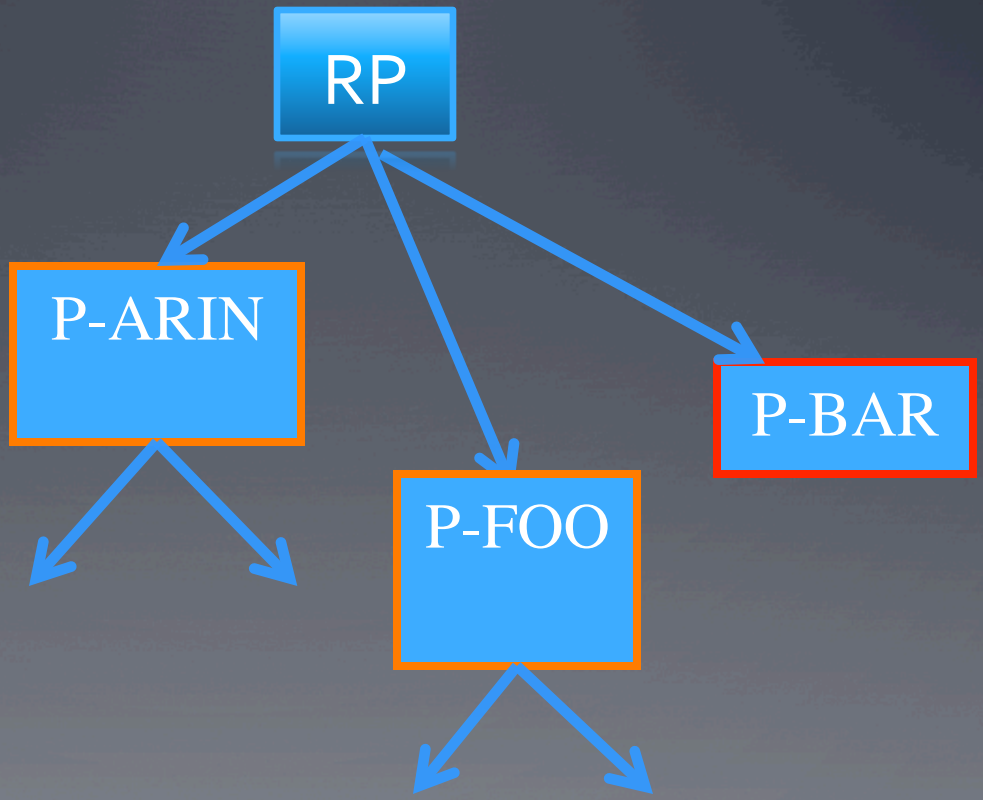

**(RP wants to make use of 10/8 for local routing)**

# 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
  - Proofreading algorithm
- Resource re-writing algorithm
  - Target processing
  - Ancestor processing
  - Tree processing
  - TA re-homing
- Path discovery
- Revocation
- Expiration

# Constraints File

- The RP creates (or imports) a constraints file specifying IP address and AS# resources for target certificates
  - Certificates are specified by SKI
- 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

- The process begins with an optional proofreading algorithm to verify and clean up the constraints file
- The four stages to the algorithm have not changed
  - Target processing
  - Ancestor processing
  - Tree processing
  - TA re-homing
- One change under considerations is to take some global flags from the constraints file and allow them to be applied to individual certificates (e.g., to change validity intervals)

# Processing Order Dependencies?

- What happens if a certificate is processed by more than one stage of the algorithm?
  - Can the resulting “paracertificate” be dependent on the order of the entries in the constraints file?
- An iterative sorting algorithm is applied to the constraint file entries to remove such dependencies
- There is an upper bound on the iteration count to ensure that the algorithm converges
- So long as the maximum path length is less than or equal to this upper bound, no order dependency can occur

# Implications & Resolutions

- This algorithm creates two hierarchies: the original certificate hierarchy and the paracertificate hierarchy
  - The path discovery algorithm prefers the paracertificate hierarchy
  - The original hierarchy and the para-hierarchy are disjoint; revocation of a certificate in one does not affect the other
  - Paracertificates are all issued by the RP, so only the RP can revoke them, while riginal certificates are revoked by their issuers
- Expiration dates can be changed  by the constraints file, overriding original certificate expiration dates

# A Significant Change

- The -05 and prior versions say that the syntax for the constraints file is informative, not normative
- I want to change this to make the syntax mandatory, to enable constraints files generated by different sources to be accepted by all RP software
- The algorithm is still nominal, i.e., any algorithm that yields the same results is OK
- What do folks think?

# A Side Note

- Mark Reynolds, the principle author of this document, and the software engineer, is making available an open-source version of the LTA software for a more general environment
- Island Peak Software will release an LTA plug-in for Firefox as free, open-source software
- I believe that this version focuses on the more general PKI context, with less emphasis on 3779 extensions (which tend to be RPKI-specific)
- Contact Mark (mcr@islandpeaksoftware.com) for more info

# Questions?