Local Management of Trust Anchors for the RPKI

Stephen Kent

BBN Technologies

Local TA Management

- A TA is a public key and associated data used as the starting point for certificate path validation
- It need not be a self-signed certificate (although I am told that OpenSSL requires this format!)
- An underlying assumption in PKI standards is that each relying party selects the trust anchors it will use
- Thus the set of TAs employed by a PKI-enabled application is a local matter
- In practice, few PKI-enabled applications provide users with good tools for managing TAs!

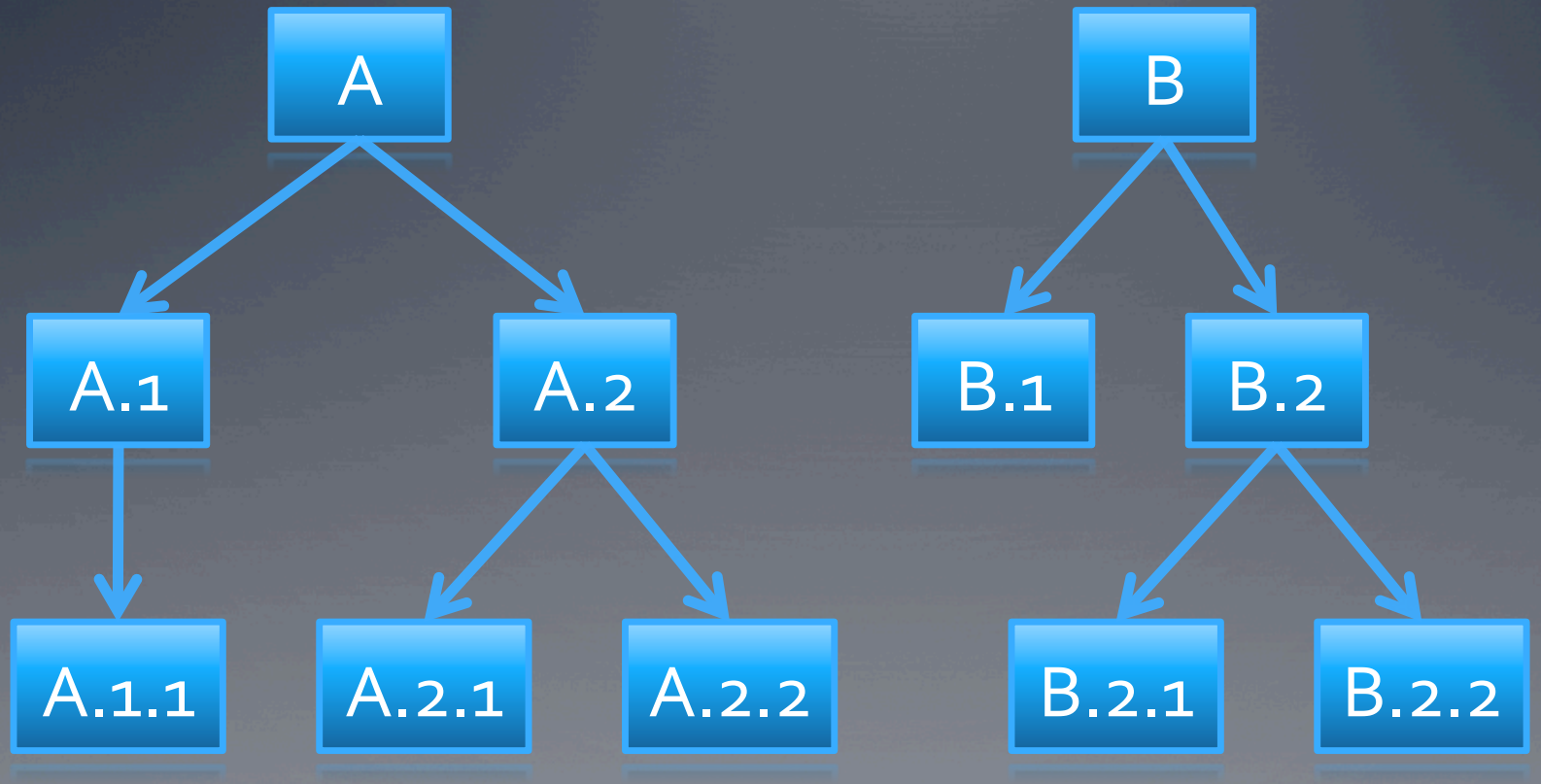
TAs in the RPKI

- The RPKI architecture follows the general PKI model with respect to TAs, i.e., it assumes each relying party (RP) selects its own set of TAs
- In the RPKI, a TA must include a public key, a subject name, and RFC 3779 extensions, at a minimum
- Thus an RP must be able to create compatible TAs
  - To allow use of local address space for (local) routing
  - To reflect local security decisions about TAs, while still maintaining compatibility with RFC 3779 certificate processing
- This motivates creating a tool to help RPs manage TAs

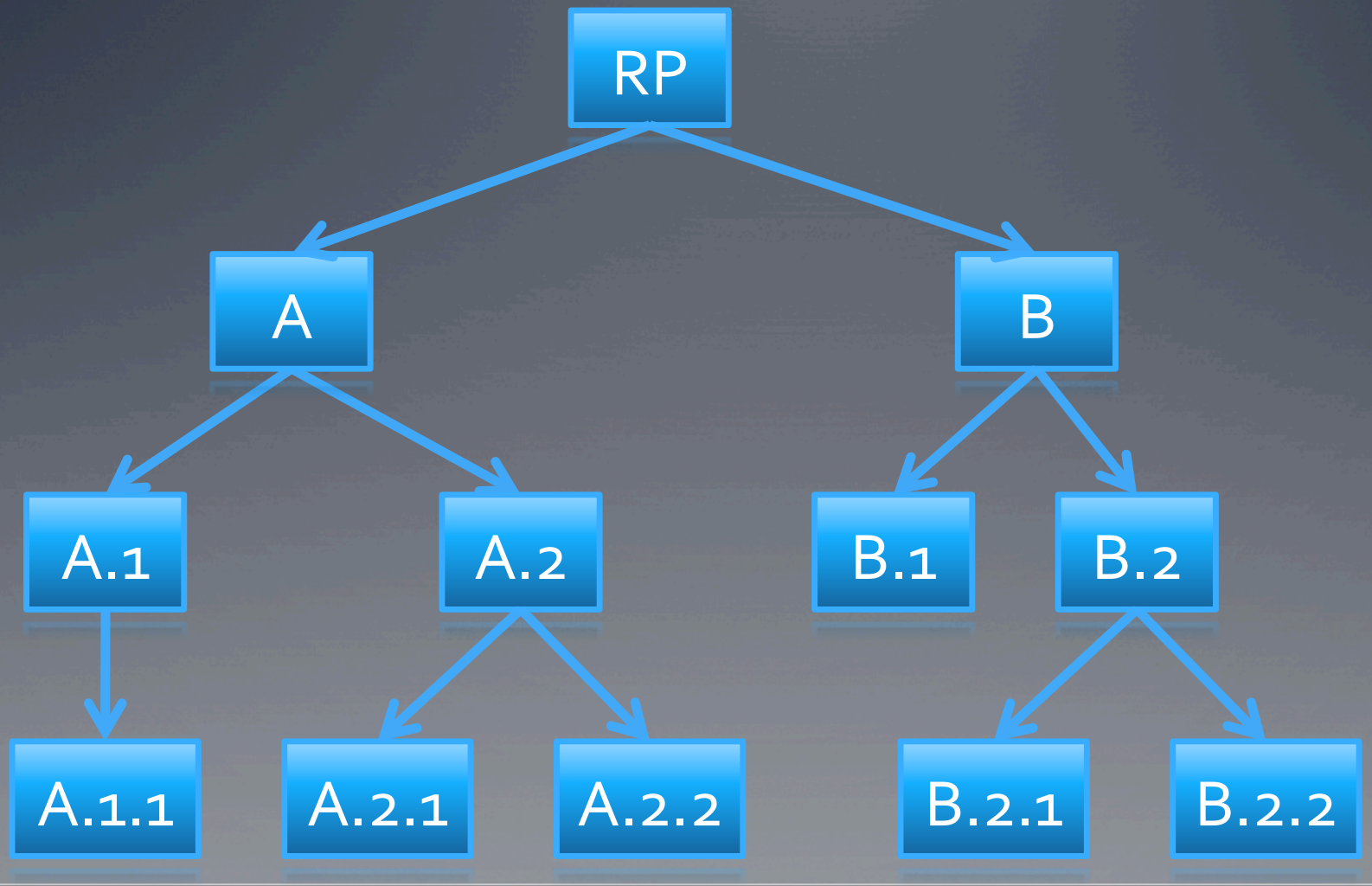
The RP as the TA!

- The next 2 slides show a PKI with two CAs (A and B) that have offered themselves as TAs (to a set of RPs), by issuing self-signed certificates
- In the first slide we see the PKI as perceived by these two CAs (two, singly-rooted trees)
- In the second slide we see the same PKI as viewed by an RP that has acquired the certificates issued by A and B, but has NOT agreed to accept them as TAs per se (e.g., maybe to add constraining extensions)
- It has transformed the PKI by replacing the self-signed certificates with certificates issued under itself as TA

PKI as Advertised by A & B



PKI as Perceived by the RP



What did the RP do?

- Issue a self-signed CA certificate for itself, to act as the only TA for the RP
- Acquire certificates for A & B and verify them
- Extract the subject name, public key and any extensions that are “important” from each certificate
- Modify (or add) important extensions to match the RP’s policy, thus overriding what A or B may have asserted in their self-signed certificates
- Issue new certificates to A and B with the RP as the issuer

A's Certificate: Before and After

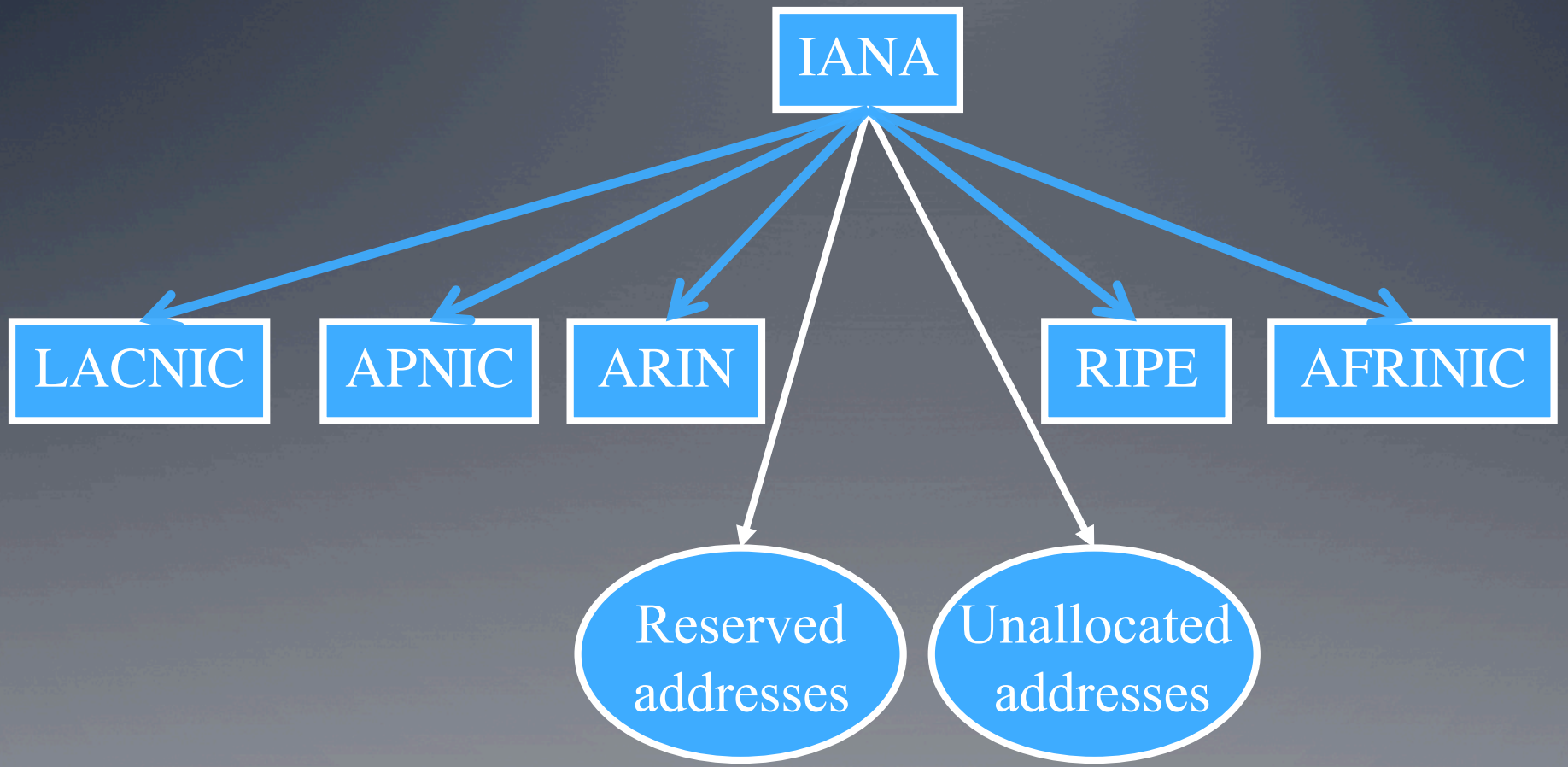
Issuer = A, Subject = A, PK = 123…, signed by A

Issuer = RP, Subject = A, PK = 123…, signed by RP

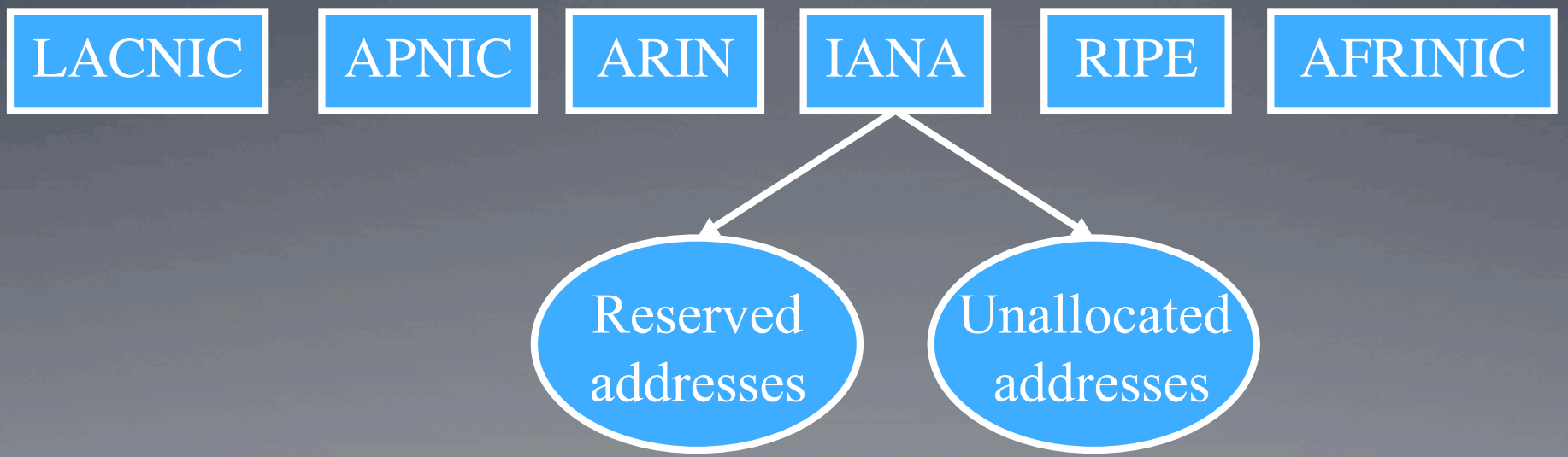
The RPKI Version

- In the RPKI we need to be able to create new certificates, possibly with modified RFC 3779 extensions
- To make this work the RP
  - Self-signed RP certificate must contain RFC 3779 extensions encompassing all addresses and all ASNs
  - Issues new certificates, under the RP's TA, excluding any 3779 extension data that it wants to control directly
  - Re-issues certificates with new 3779 extensions to override the RPKI tree (reissue parent certificates as needed)
    - Delete overlapping 3779 data as needed
    - Re-home targeted certificates under the RP TA

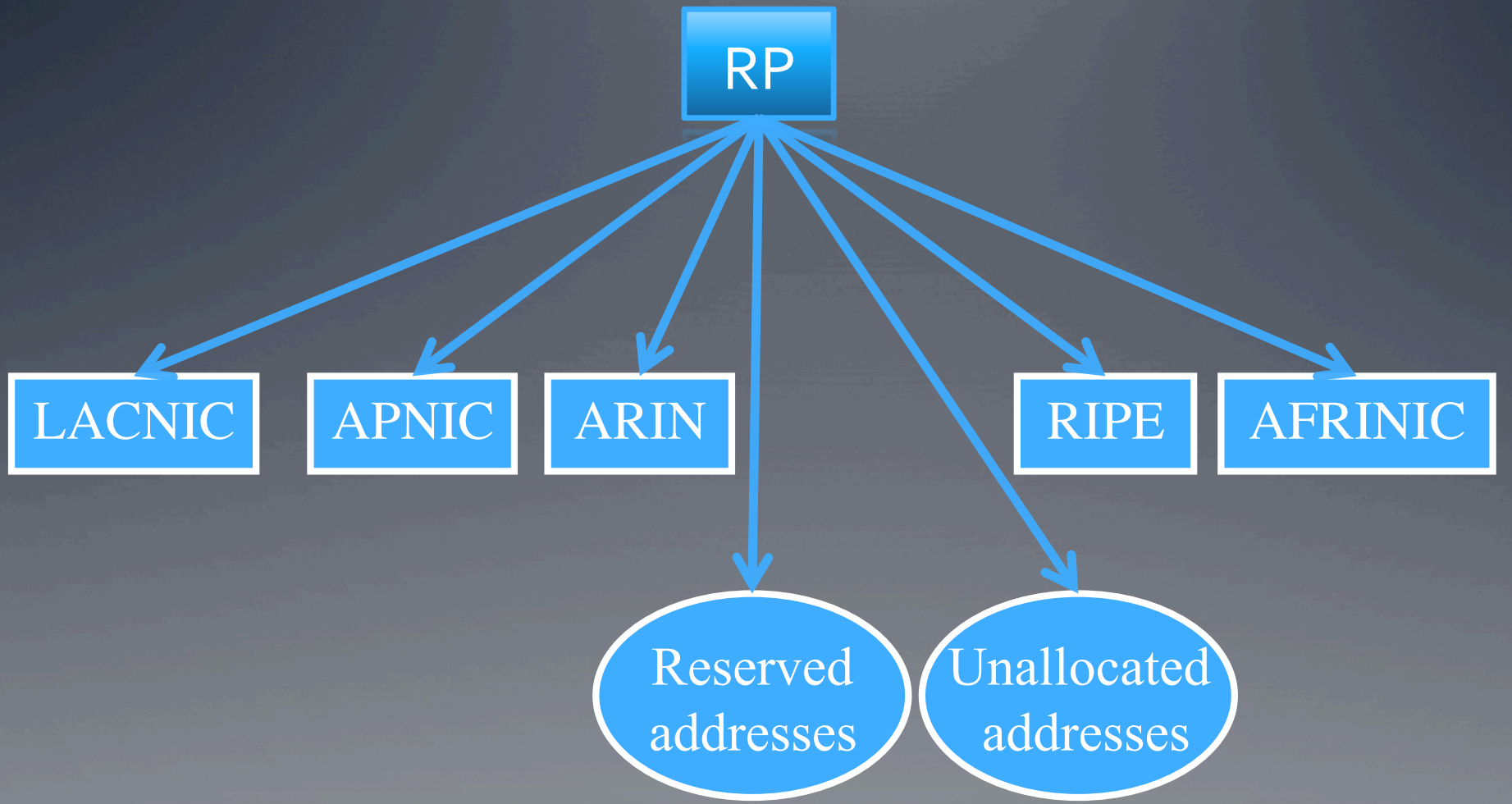
An RPKI TA Example (1/2)



An RPKI TA Example (2/2)

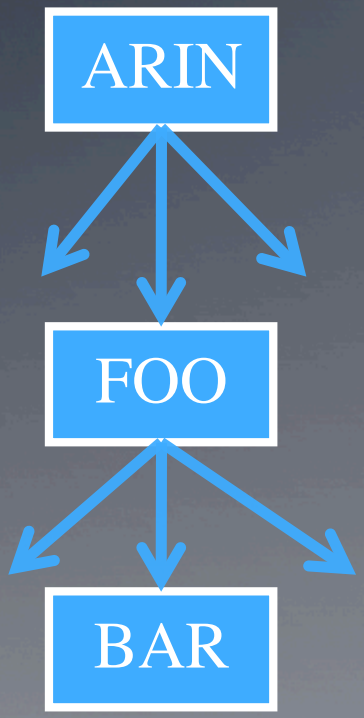


RPKI with Local Control

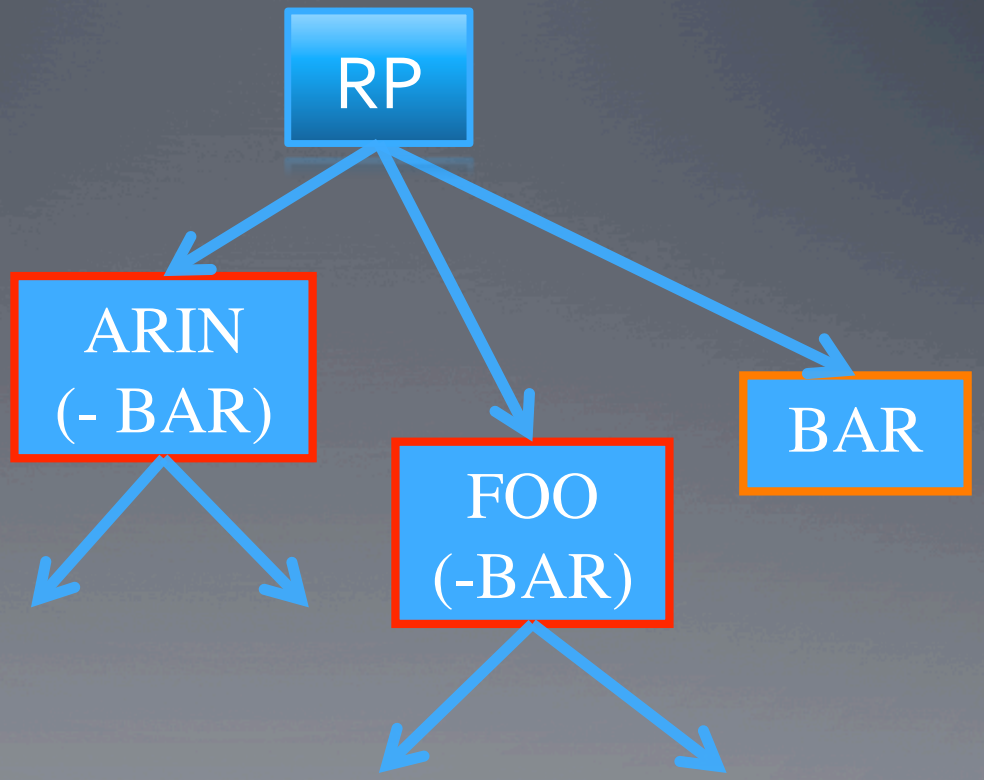


A More Elaborate Example

As offered by ARIN



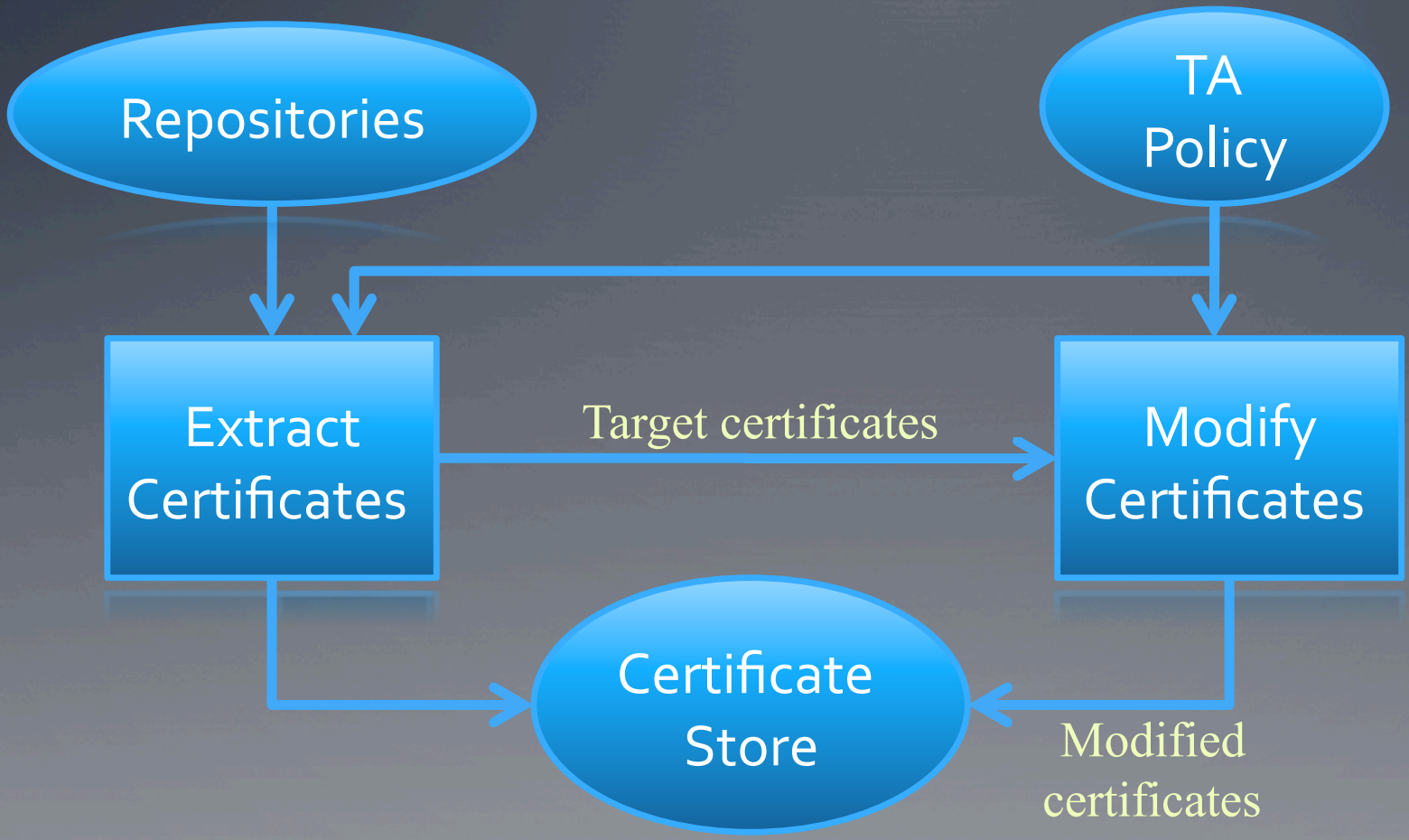
As managed by an RP



What does this do?

- It allows each RP to override the nominal RPKI hierarchy, on a local basis
- It is easy to manage if you want to override resource allocations only for local resources (i.e., your allocations) or IANA “reserved” allocations
- It is harder to manage IF you want to create direct links to many CAs, especially at lower tiers in the hierarchy
- BBN plans to provide open source software that supports this model, and that works with the rest of our RP software

BBN SW Model (revised)



What does this Proposal Do?

- It instantiates an RP as the only TA, a model that offers the ultimate in local policy control
- It enables each RP to import putative TAs, check them against a local policy, and reissue their (self-signed) certificates to match the local policy, as needed
- It allows re-homing selected sub-trees of the RPKI at any tier, at the cost of additional policy specification complexity and more certificate issuance operations
- It allows a local authority to specify a policy and then export the results of applying that policy (to the RPKI) to other RPs that are willing to rely upon that local authority

What Else is Needed?

- We need a good way to express an RP's local policy, to drive certificate re-issuance & re-homing
  - Might specify this policy as a hash of the target certificate's public key (SKI) and the 3779 extensions to be used
  - A good GUI might help
- This proposal does NOT
  - Address how to represent TA info for RIRs
  - Say how to acquire and verify putative TA info
  - Provide details of how to manage the local cache when it is modified by this local policy enforcer, e.g., breaking AIA/SIA links and manifests

What do we Call This?

Multi-Entity Facets of

Internet Resource Trust

ME FIRST

Courtesy of Richard Barnes

Questions?