Local Management of Trust Anchors for the RPKI

Stephen Kent
Chief Scientist – Information Security
BBN Technologies

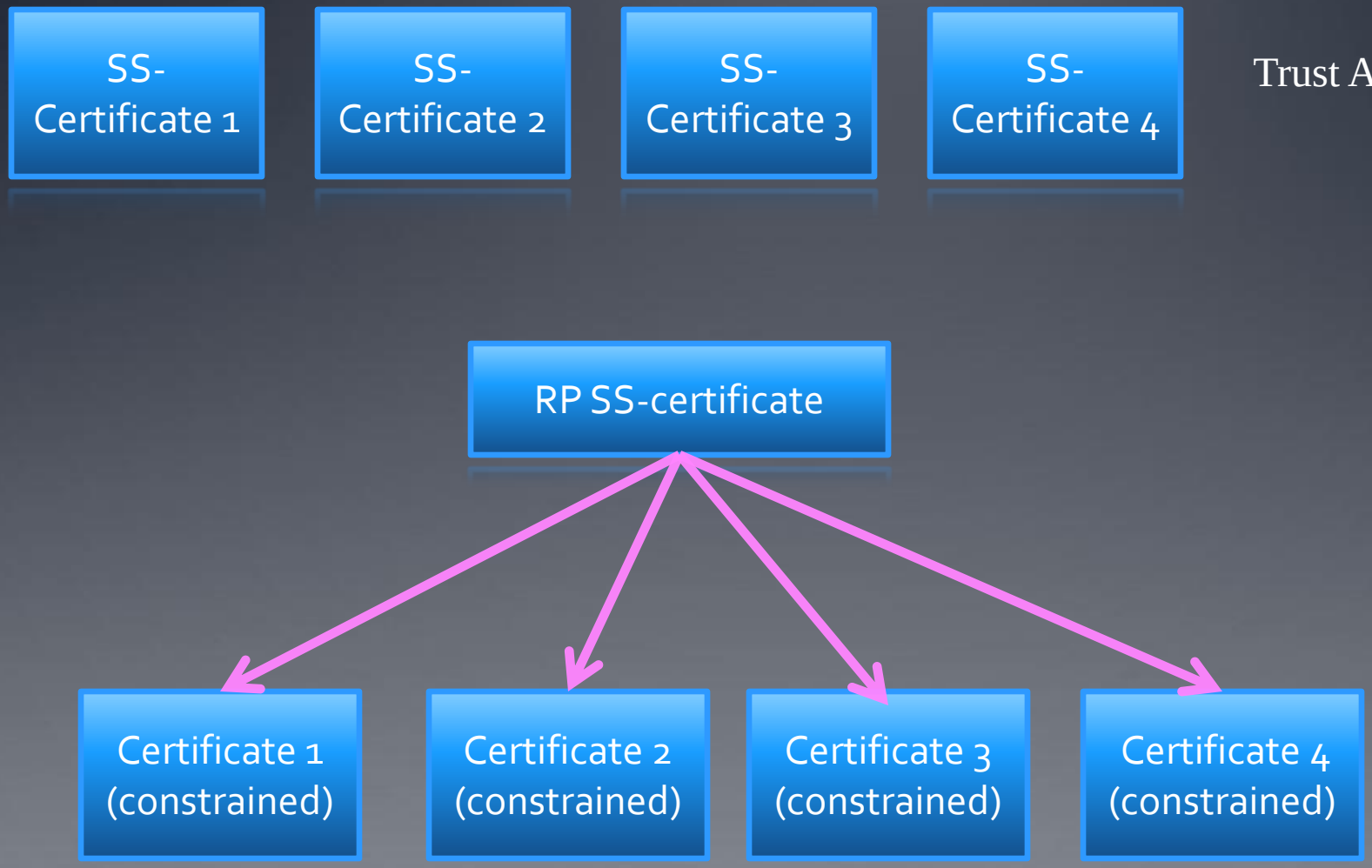
Local TA Management

- A trust anchor (TA) is a public key and associated data used as the starting point for certificate path validation
- Often a TA is represented by a self-signed certificate, but standards not not require 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!

Another TA Management Issue

- As per RFC 5280, normal certificate path processing does not make use of any extensions bound to a TA (e.g., in a self-signed certificate)
- Thus extensions that can be used to constrain the scope of a TA, e.g., Name, Policy, and Basic Constraints (path length) extensions are ignored
- Most self-signed certificates proffered as TAs don’t contain constraining extensions anyway
- There is an easy fix: make the RP the ONLY TA it recognizes, and re-issue all self-signed certificates under that TA, adding constraints as needed!

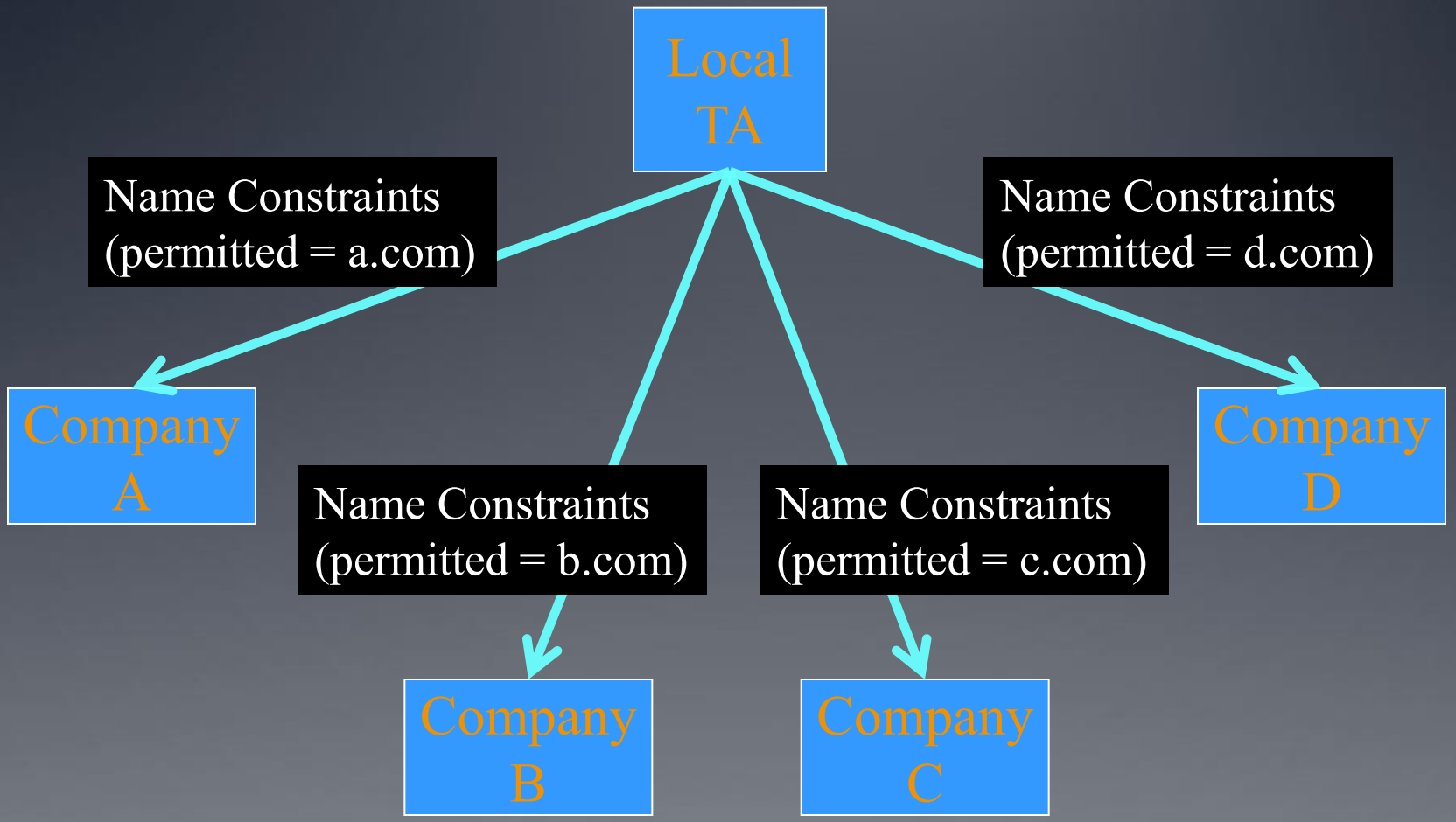
Local TA Example



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, even though one or more “default” TAs will be available
- In the RPKI, a TA must include a public key, a subject name, and RFC 3779 extensions, at a minimum
- An RP should be able to manage TAs locally
  - To allow use of RFC 1918 address space for (local) routing
  - To reflect local security decisions about which default TA(s) and CA to trust, while still maintaining compatibility with RFC 3779 certificate processing

Using Name Constraints



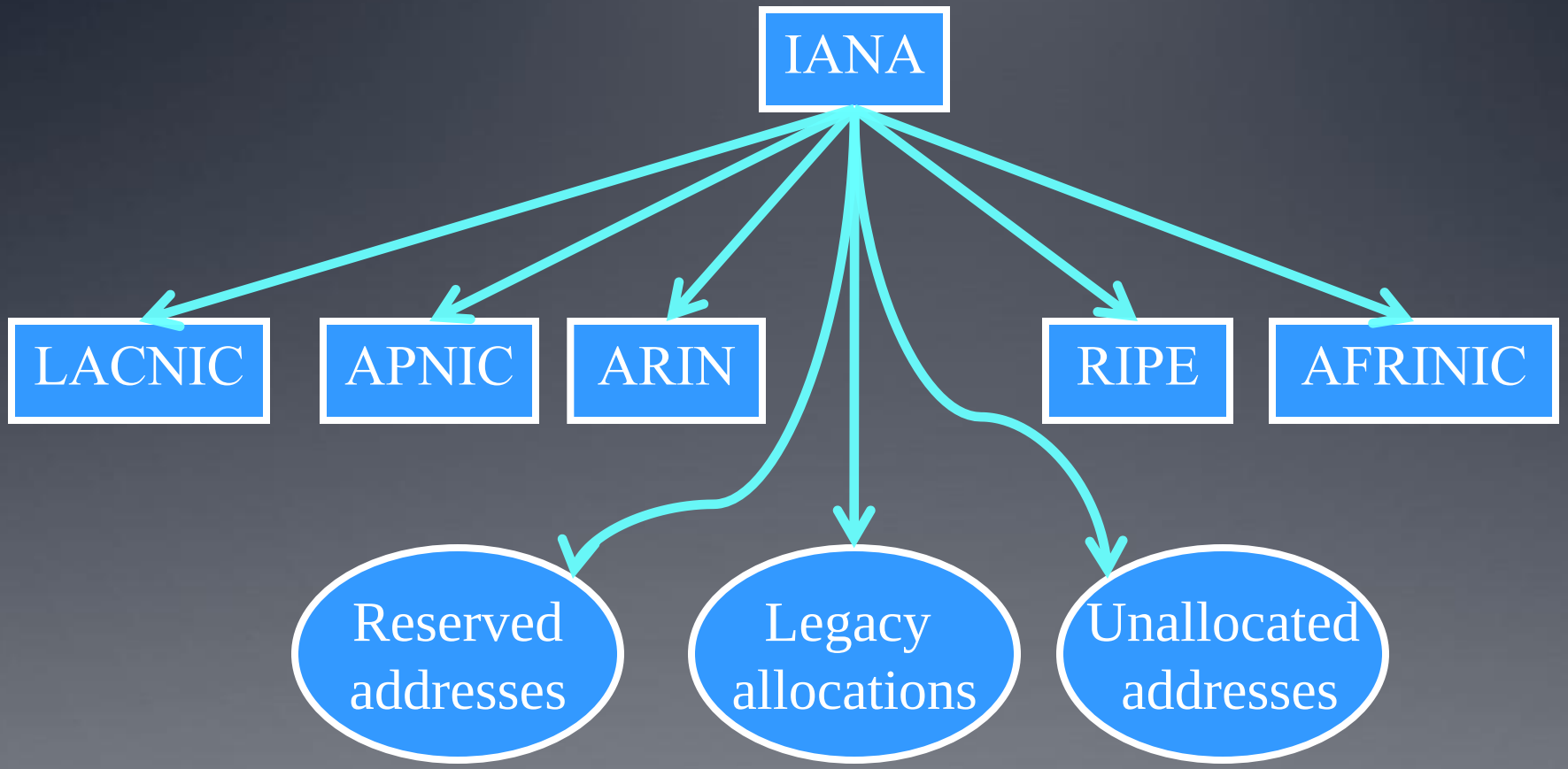
The 1-TA model & the RPKI

- We can apply the local TA management model described previously in the RPKI
- The RP imports default TAs (or any other CAs in the RPKI repository system) and re-issues certificates for them under its own TA
- The RP can thus override the RPKI nominal hierarchy, (paralleling the allocation hierarchy) as represented in the RPKI repository system
- To enable RPs to act in this way a sophisticated tool is needed

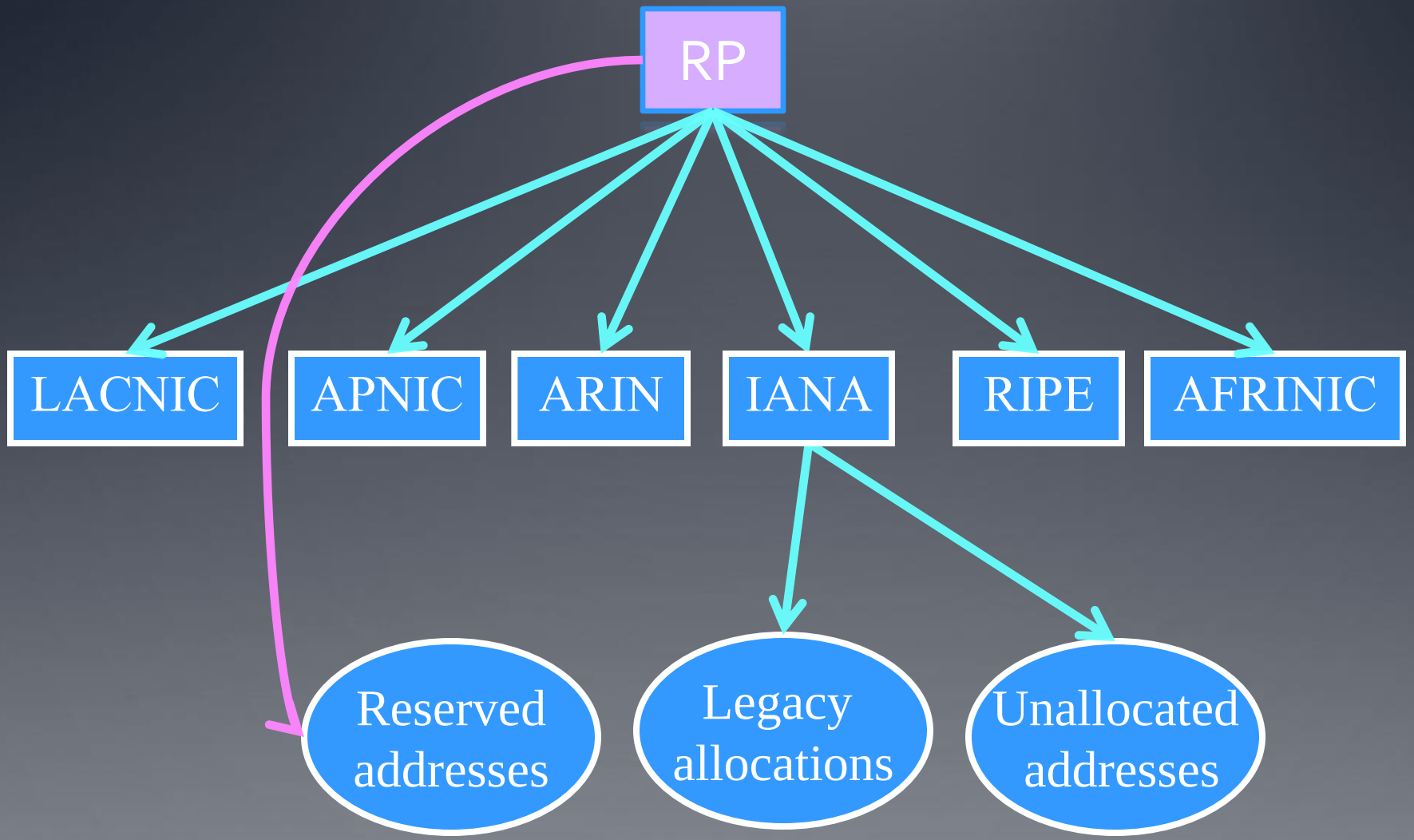
Making this work in the RPKI

- An RP will need to be able to create new certificates, often with modified RFC 3779 extensions
- To make this work
  - The self-signed RP certificate (local TA) must contain RFC 3779 extensions encompassing all addresses and all ASNs
  - The RP re-issues targeted certificates with new 3779 extensions to override the RPKI tree
    - Delete overlapping 3779 data as needed
    - Re-issue targeted certificates under the RP TA
    - Re-issue certificates for ancestors of re-parented certificates under the RP TA

An RPKI TA Example



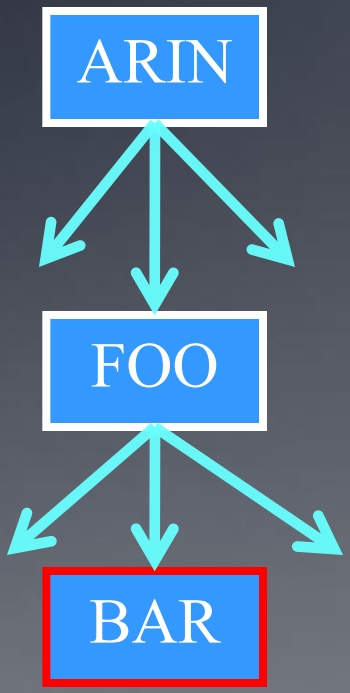
RPKI with Local Control



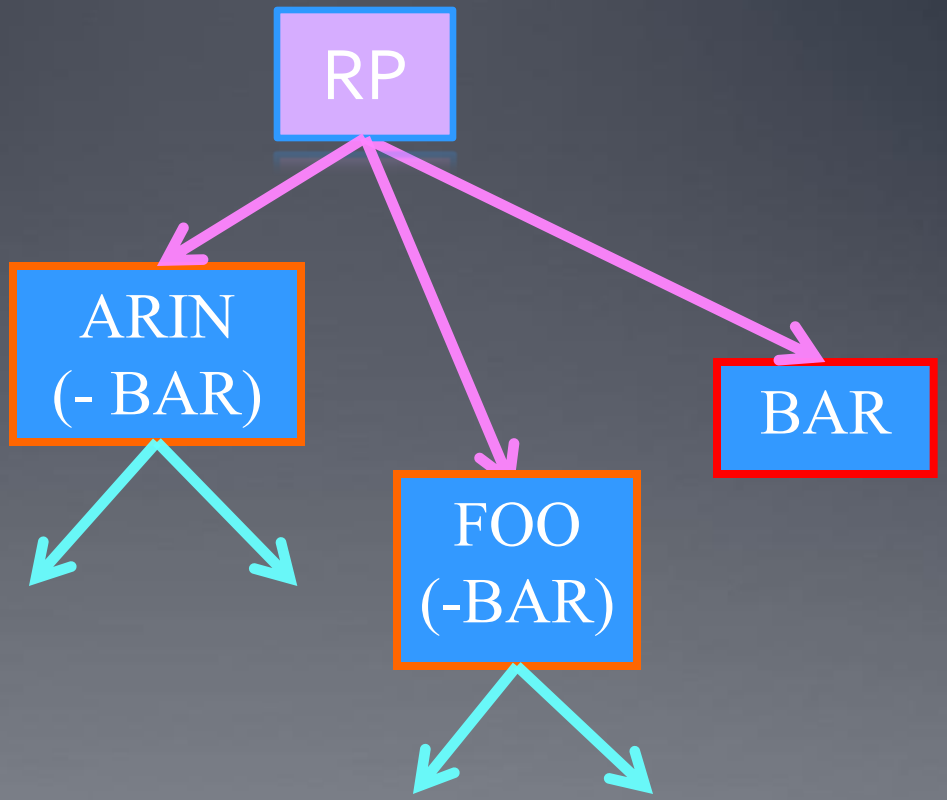
(RP wants to make use of 10/8 for local routing)

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)

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 somewhat harder to manage if you want to create direct links to many CAs, especially at lower tiers in the hierarchy
- BBN is preparing an I-D for the SIDR WG, describing how to do this in more detail, before the next IETF meeting (March 2010)

Questions?