RPKI Validation - Revisited

draft-huston-rpki-validation-01.txt

Geoff Huston
George Michaelson
APNIC

Motivation

- The current model of validation, when applied to the RIR model of resource distribution, has lead us to an intricate system of multiple certificates with complex transitional states with a high degree of fragility
- The consequences of INR inconsistencies at points that are high in the RPKI hierarchy include the potential for catastrophically large routing failures though unintentional certificate invalidation
- Demanding persistent absolute perfection from our certificate management systems has its elements of risk
- Is it possible to think about removing some aspects of this complexity within the RPKI framework, and also reducing the scope of consequential damage of certificate INR mismatch?

RPKI Validation

RFC3779:

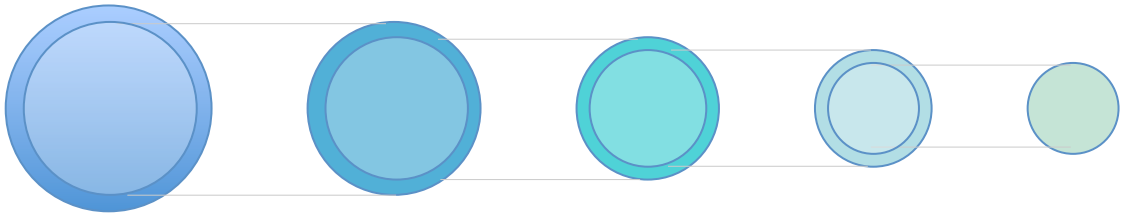
For a certificate to be “valid”:

the certificate must satisfy a number of criteria,

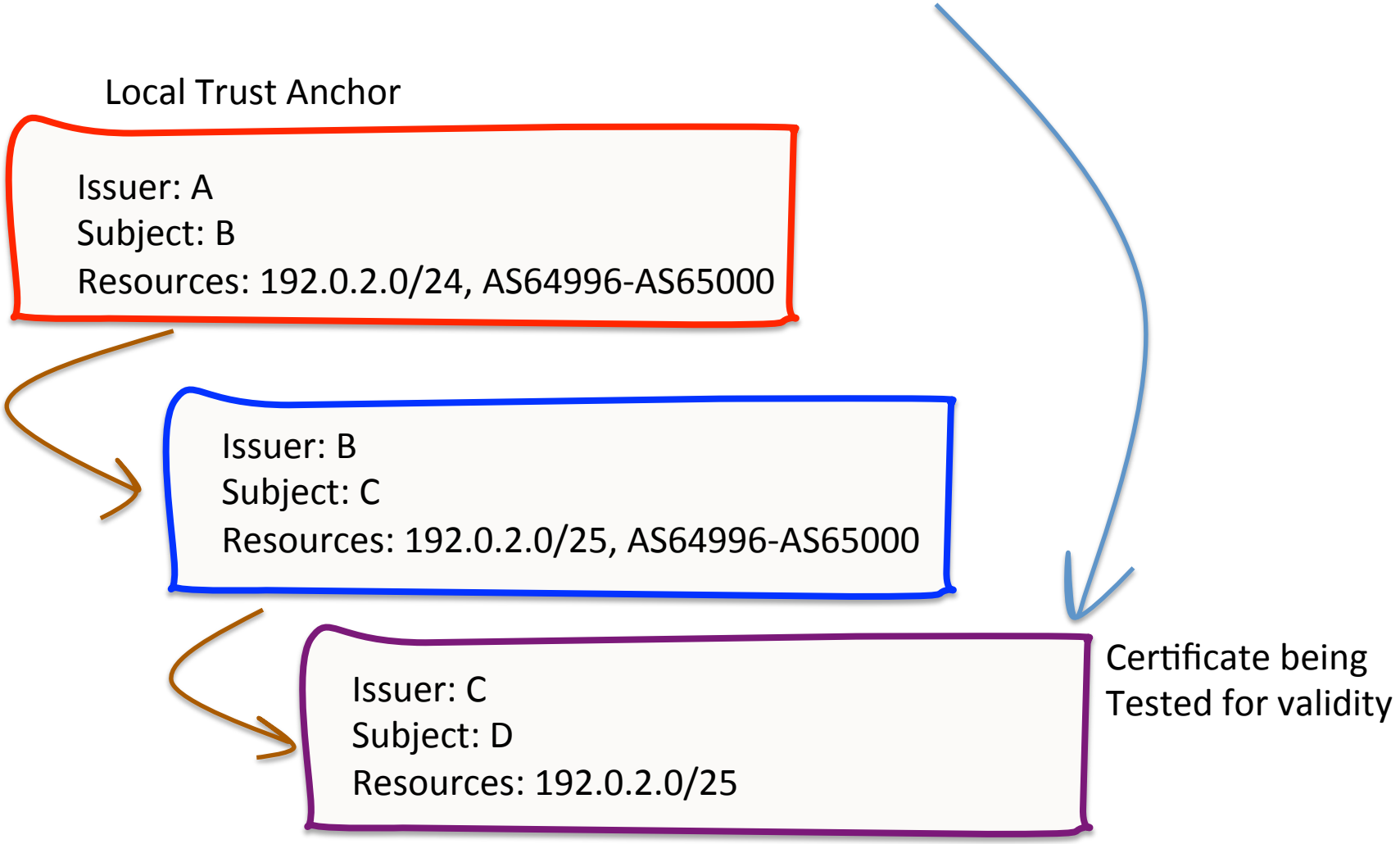
Syntax correctness, validity dates, etc

and there must exist an ordered sequence of certificates (1..n) where:

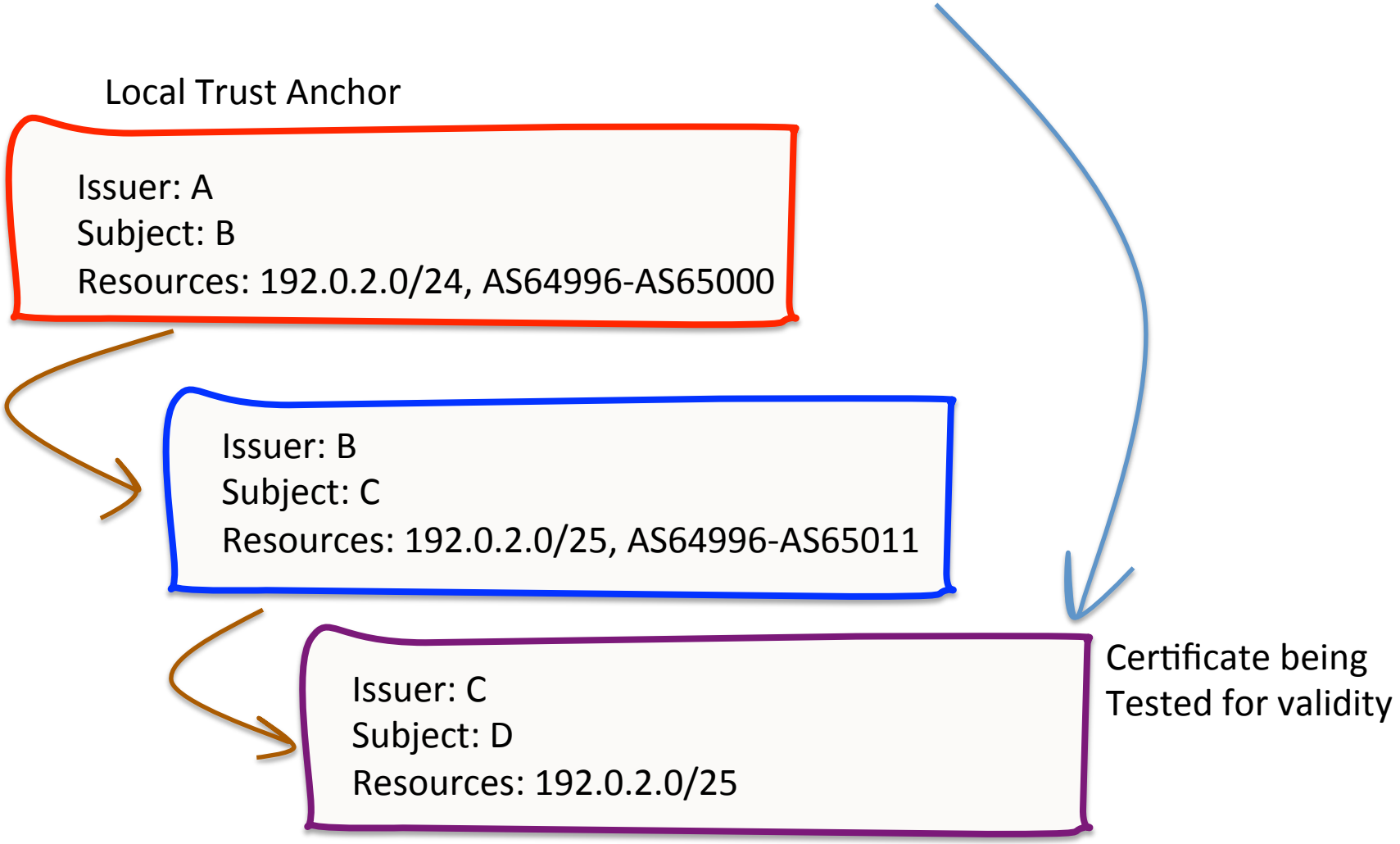
- Certificate 1 is issued by a trust anchor
- Certificate x’s Subject Name value matches Certificate x+1’s Issuer Name value
- The resources in the INR extensions of Certificate x+1 must be “subsumed” by the INR extensions listed in Certificate x
- Certificate ‘n’ is the certificate to be validated
- Certificates 1 through n-1 are also “valid” according to this same criteria



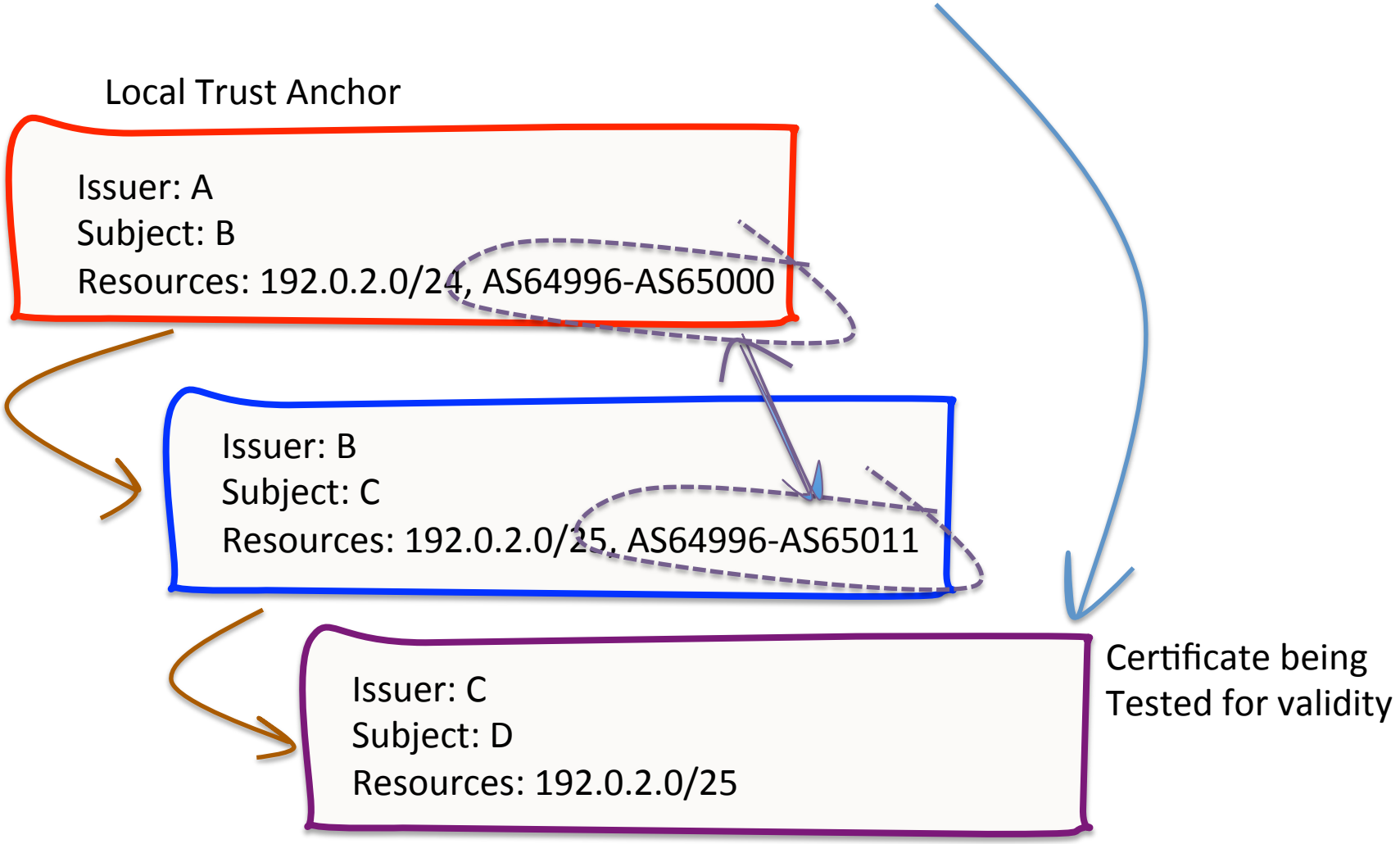
This is Valid



This is not Valid



This is not Valid



Why is this?

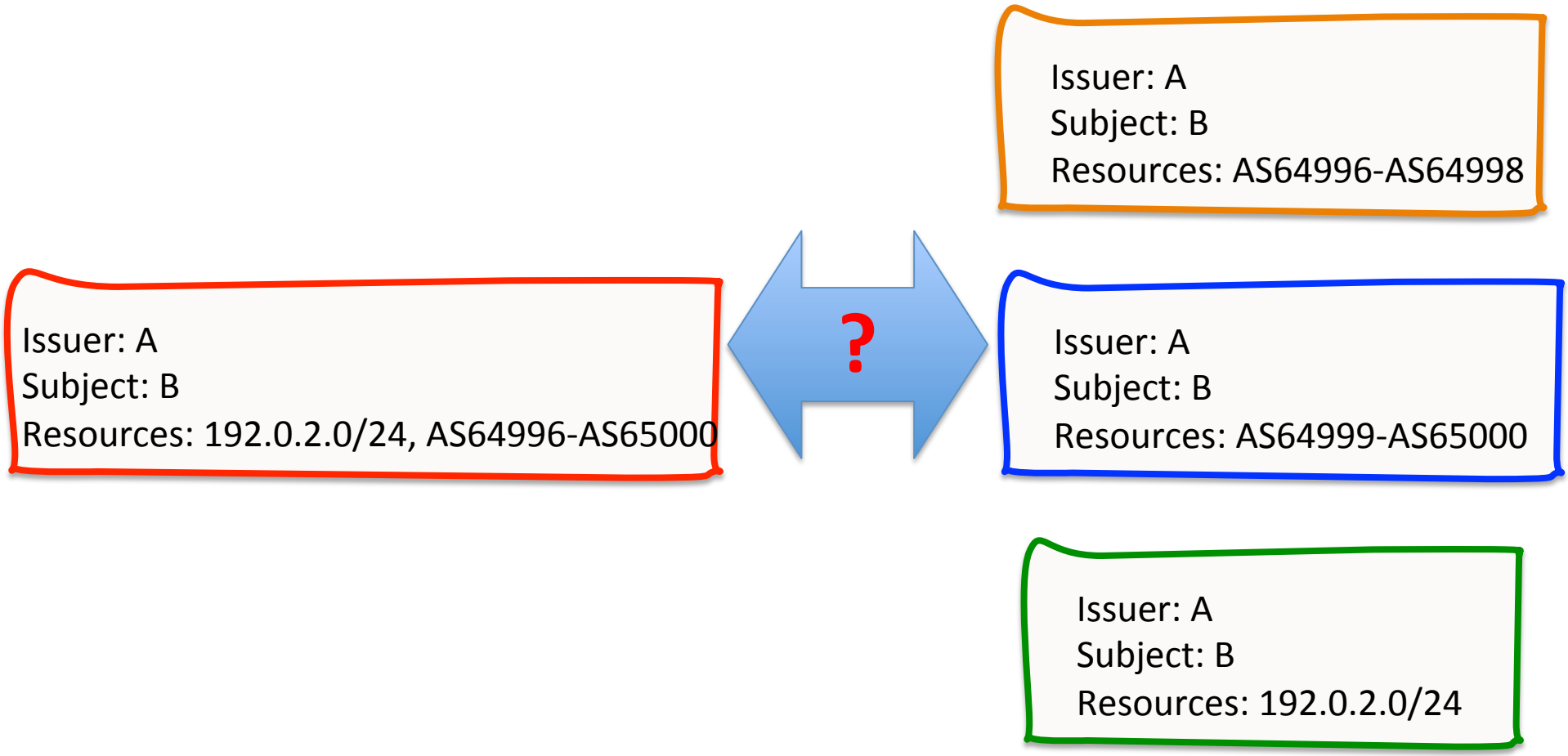
We looking at the entire collection of resources in a certificate as an immutable blob when we think about validation

Why are we doing this?

Are there alternative perspectives?

The Semantics of an RPKI Certificate

What is the semantic difference between a single certificate and a collection of certificates with common crypto and control values?



The Semantics of an RPKI certificate

What's critical in terms of the Resources contained in the RPKI cert?

- Is it the COLLECTION of resources that's critical?
- Or the ENUMERATION of the resources contained in that collection?

Lets explore the implications of asserting that it's the the enumeration of the individual resources contained in the certificate that are critical here, not the particular collection of resources

The Semantics of an RPKI Certificate

These two certificate sets represent the same information

Issuer: A
Subject: B
Resources: AS6496-AS6498

Issuer: A
Subject: B
Resources: 192.0.2.0/24, AS6496-AS6500

=

Issuer: A
Subject: B
Resources: AS64499-AS6500

Issuer: A
Subject: B
Resources: 192.0.2.0/24

So what?

The RFC3779 definition of certificate validation has its operational consequences

Treating the collection of resources as an immutable aggregate has caused considerable complexity and fragility in the RPKI

Examples:

- Holders of resources that have different allocation paths require multiple certificates
- Changes in resource holdings require careful synchronization of certificate issuance actions between distinct actors
- Mismatch in collections between parent and child certificates invalidates the entire child hierarchy

An Alternative Approach

- Treat each RPKI certificate as a separable set of resources with a common crypto bundle
- Re-phrase the RP's validation approach to validate as many of the resources contained in the certificate's INR extension as possible, with a given set of TAs
- For each certificate generate a set of resources which is the union of all resources that can be validated via this certificate by the RP using the set of the RP's chosen TAs

An Alternative Validity question

Replace:

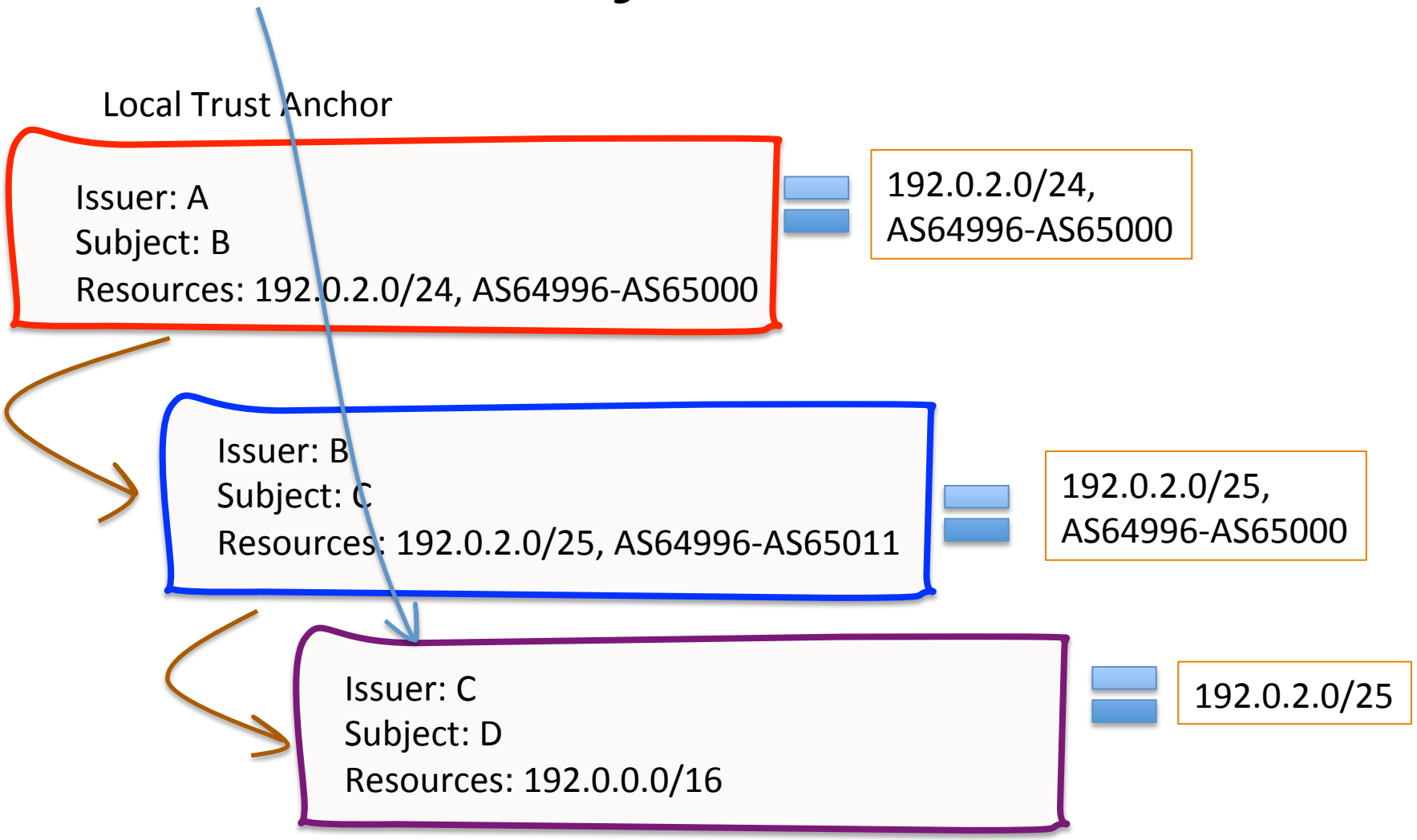
"*Is this certificate valid for the entire collection of resources listed in the certificate?*"

with:

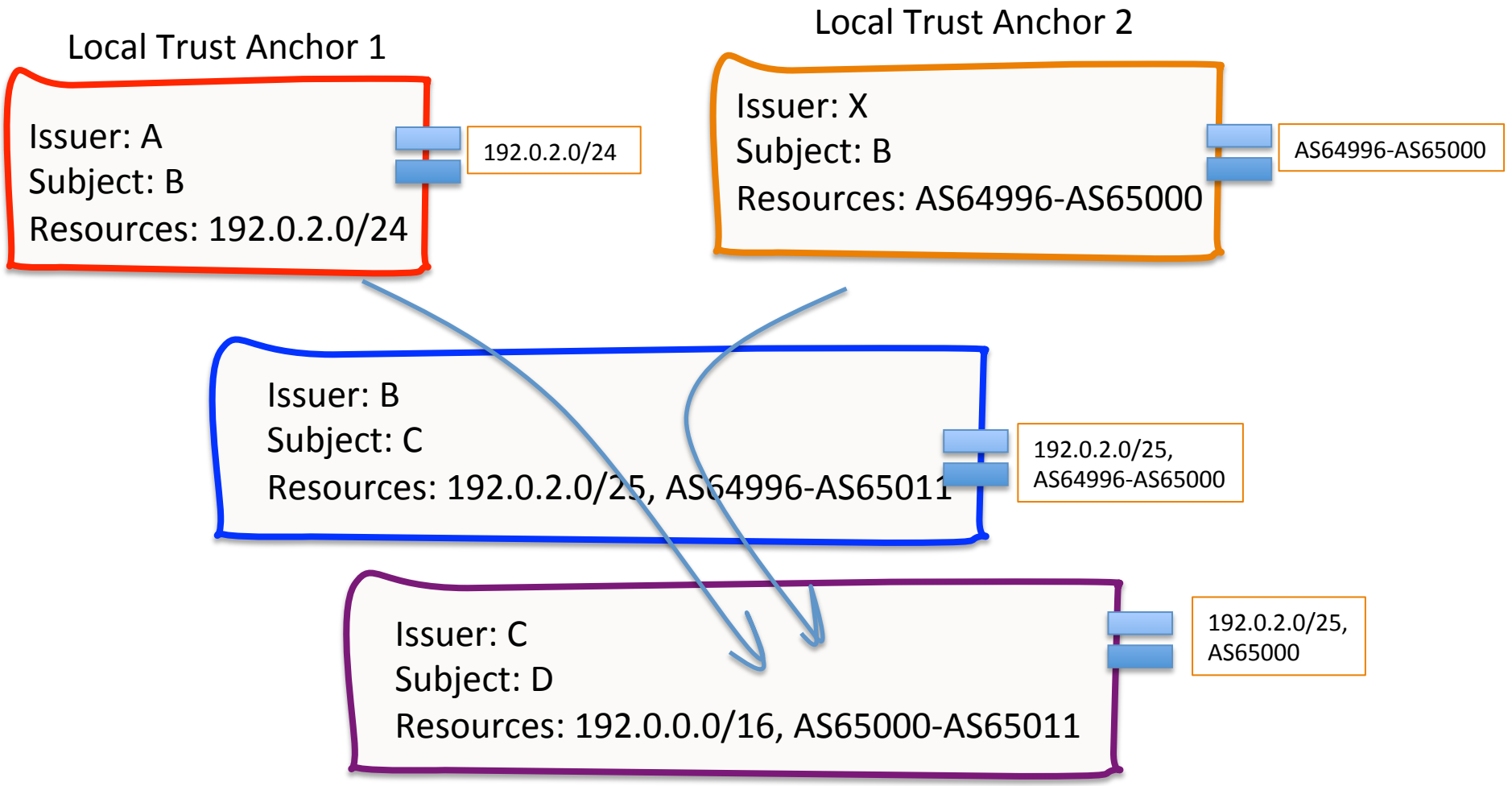
"***For which resources*** *is this certificate valid?*"

i.e. associate a computed set of resources with a certificate such that these are the resources that the RP can validate using the RP's chosen set of TAs

This is Valid *for 192.0.2.0/25*



An example using Multiple TAs



What else changes?

Not much:

- A ROA is valid if the certificate used to sign the ROA is valid *for the resources listed in the ROA*
- Similar refinements can be used in other cases of RPKI certificate use
- The semantics of certificate issuance are unaltered: CAs only issue certificates based on their records of the resources held by the subordinate entity
- The top-down local cache validation function is consistent with the current approach to local cache management

A revised Local Cache Management Approach

- Perform top-down local cache construction
- Add a data object to the local cache of each certificate
  - This object holds the intersection of the resources listed in the associated certificate and the resources in the data object associated with the “parent” certificate
- Use the resources in the associated data object instead of the resources listed in the certificate in all cases where “resources certified by this certificate” are used

This Alternate RPKI Validation Model

This alternative approach:

For a certificate to be "valid" for a given Internet Number Resource:

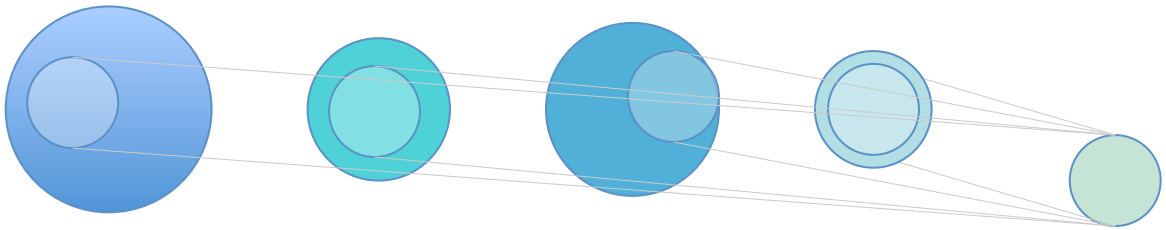
the certificate must satisfy a number of criteria,

Syntax correctness, validity dates, etc

and there must exist an ordered sequence of certificates (1..n)

where:

- Certificate 1 is issued by a trust anchor
- Certificate x's Subject Name value matches Certificate x+1's Issuer Name value
- The resources in the INR extensions of Certificate x must "subsume" the given Internet Number Resource
- Certificate 'n' is the certificate to be validated
- Certificates 1 through n-1 are also "valid" according to this same criteria



So what?

This approach provides a higher degree of robustness to the RPKI system and can simplify the mapping of multiple allocation registries into equivalent certificates

- Examples:
  - Holders of resources can use a single certificate to describe the entirety of their resource holdings (or not, as it would be effectively a choice available to the resource holder)
  - Changes in resource holdings would need not be synchronized across CAs, as the only aspect of potential disruption is the resource that is being moved
  - Local Trust Anchors could be used to refer to specific resource sets without additional support mechanisms