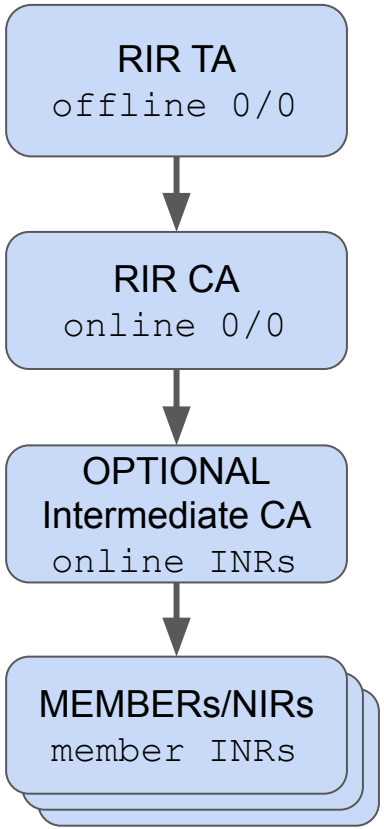
draft-nro-sidrops-ta-constraints-00

Does not replace similarly named

draft-snij-sidrops-constraining-rpki-trust-anchors

But complements it

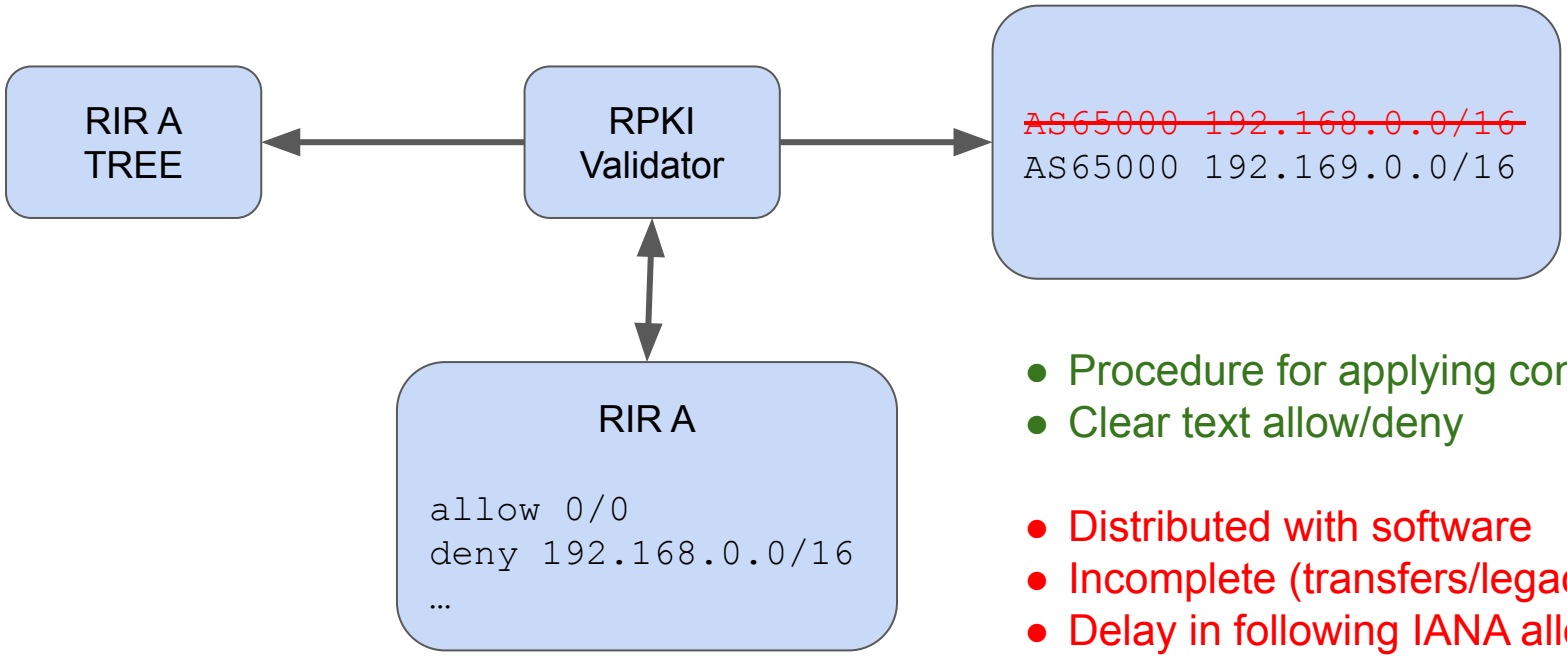
Current Setup / Problem Statement



- **Each** RIR claims 0/0 offline
- Then issues 0/0 online
- RIPE NCC has an intermediate cert (this offers no security, just clarity)
- Then members/NIRs

Each RIR can overclaim and cause e.g. ROV invalids in other regions

draft-snij-sidrops-constraining-rpki-trust-anchors



- Procedure for applying constraints
- Clear text allow/deny

- Distributed with software
- Incomplete (transfers/legacy)
- Delay in following IANA allocations

Can we do better?

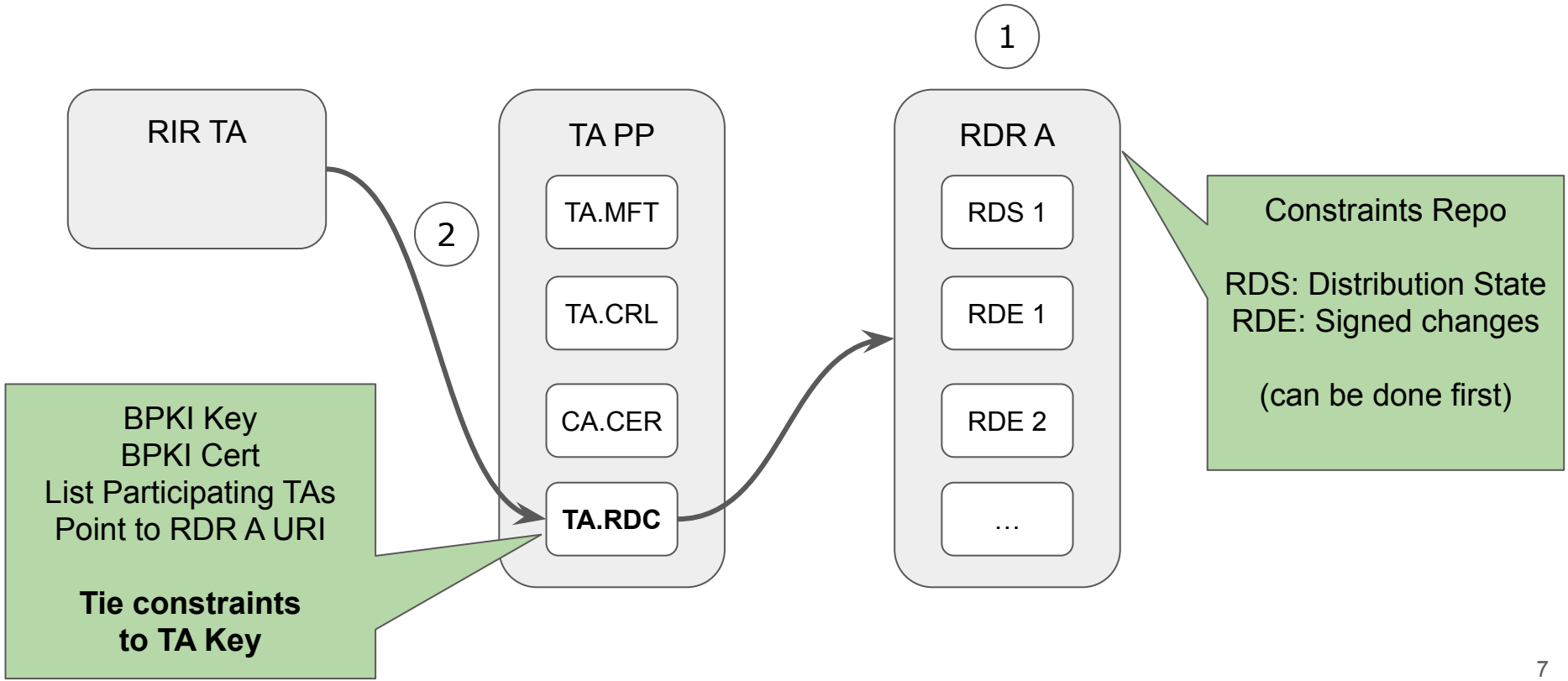
In a nutshell

Problem: any RIR can overclaim INRs causing issues at other RIRs

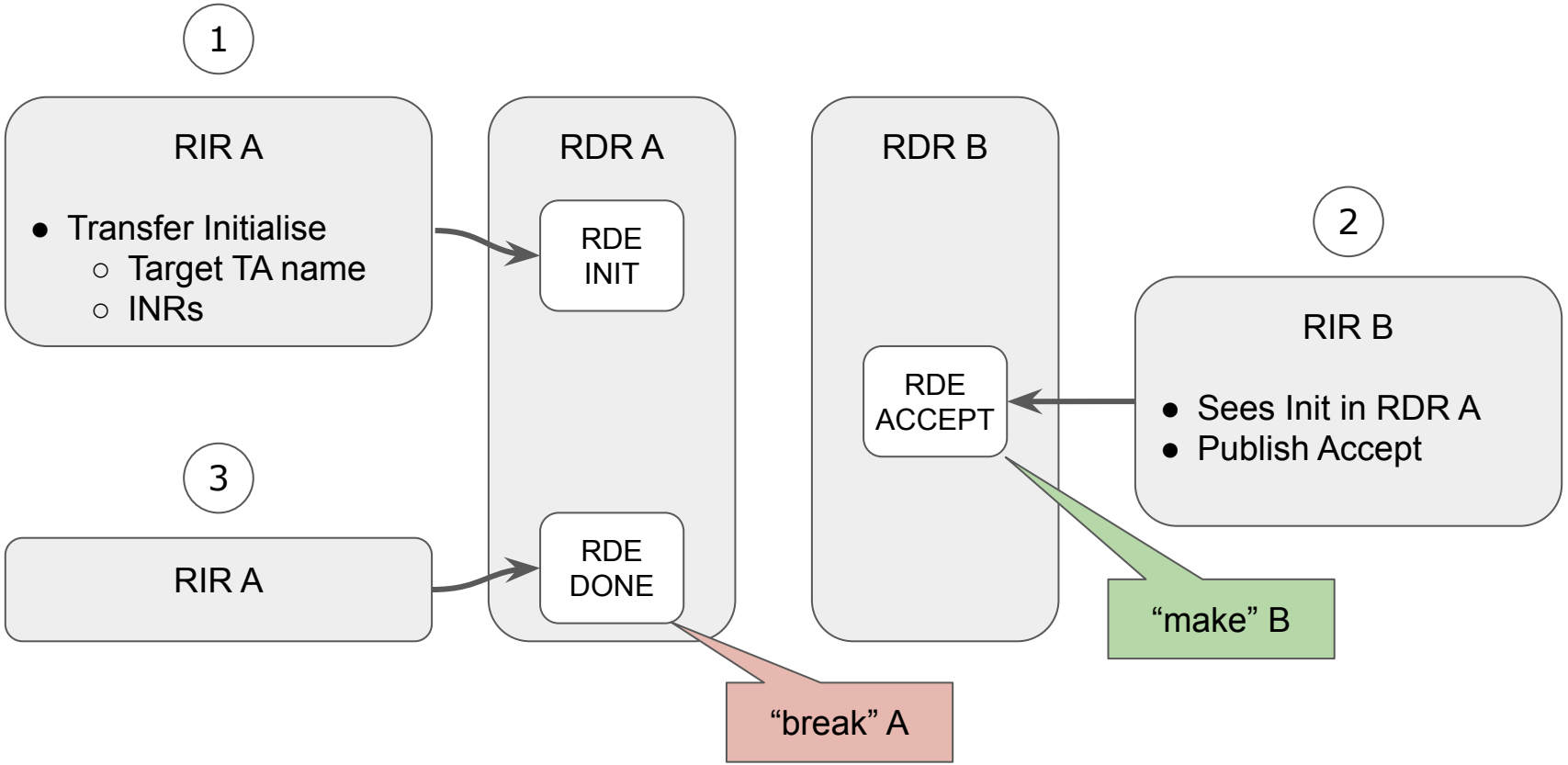
Proposal:

- Sign initial distribution of INRs between the RIRs
- Sign changes after this
- Publish *mostly* outside of the core RPKI repository

30000 Feet View



Use Case: Normal Transfer



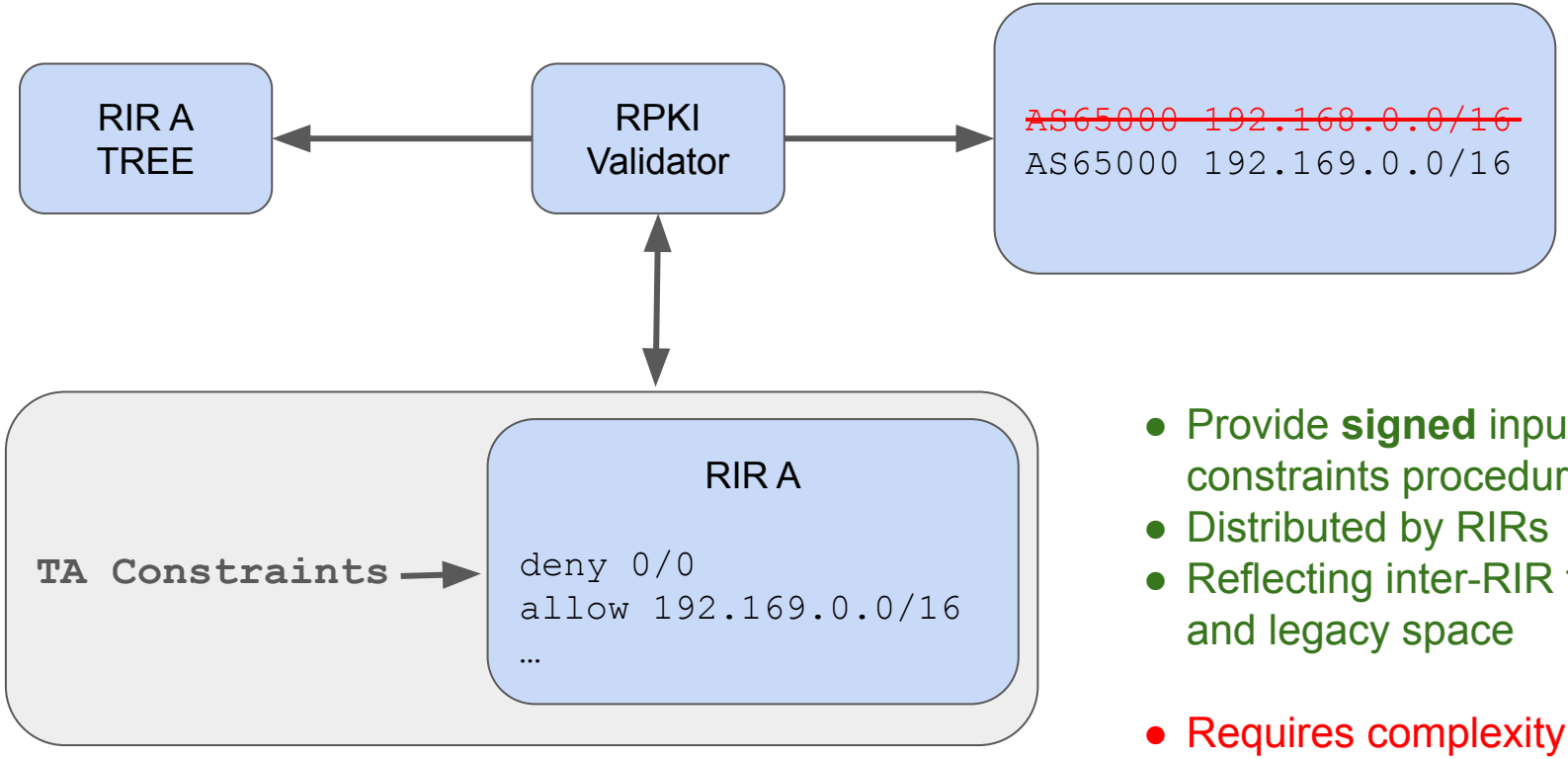
Many Other Cases

- Cancel Transfer
- Include / Exclude (IANA)
- Add / Remove Trust Anchors (ICP-2)
- Missing TALs
- Unreachable Repositories

In the draft… we only have 15 minutes…

How can this be used in future?

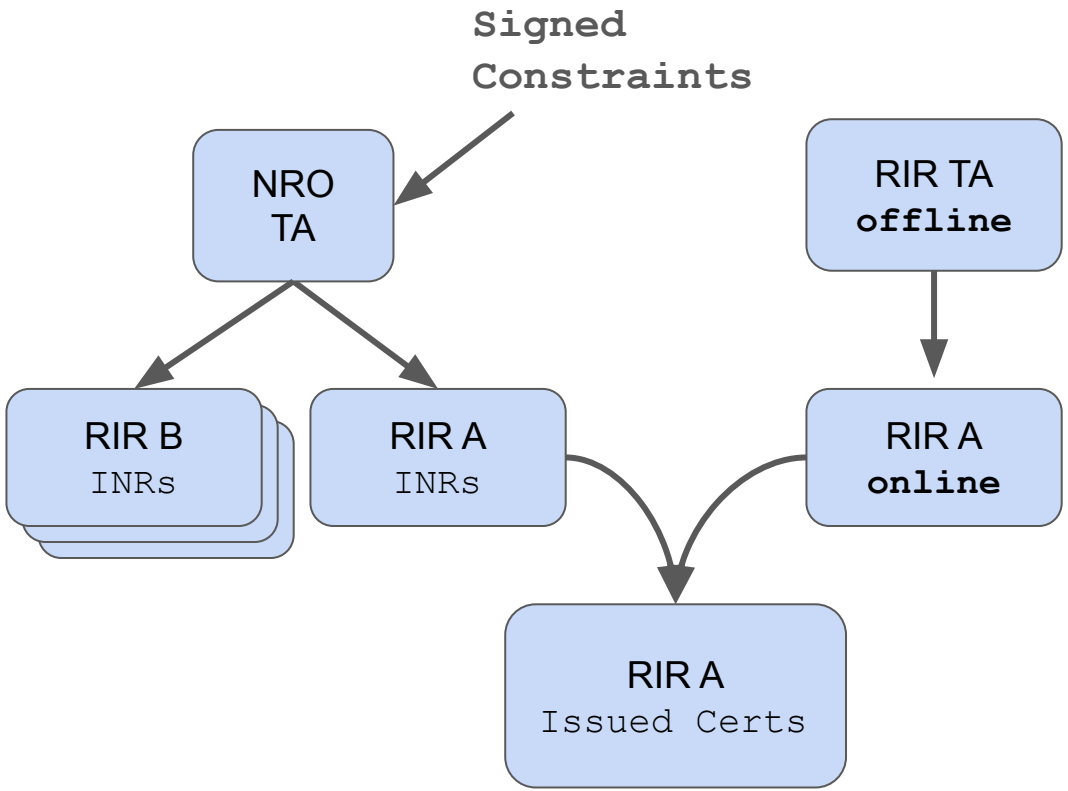
draft-snij-sidrops-constraining-rpki-trust-anchors-00 ++?



- Provide **signed** input to constraints procedure
- Distributed by RIRs
- Reflecting inter-RIR transfer and legacy space

- Requires complexity in RP software

Input to NRO Trust Anchor?



- Synthetic TA signs certificates with the same public Key and SIA URIs as RIR TAs
- Use **signed constraints** to determine INRs for each RIR
- Observers can verify correctness
- RPs can ultimately use this instead of the RIR TAs without the need to understand signed constraints
- Needs operations, signing ceremony, observers, etc..

Questions / Comments?

We are here all week!