or.. limit collateral damage in case of inconsistencies of resources in the certificate chain
Version -03

- New text and examples, same idea
- Validation algorithm using verified resources for each certificate, rather than listed resources
  - Warnings on overclaims
- Working Group, is the text more clear and are concerns addressed?
Current - all valid

Parent
192.168.0.0/16,
10.0.0.0/8

Child
192.168.2.0/24,
10.0.0.0/24

Grandchild
192.168.2.0/24,
10.0.0.0/24

ROA:
192.168.2.0/24

signs and publishes

signs and publishes

signs and publishes
Current - invalidated

Parent
192.168.0.0/16, 10.0.0.0/8

Child
192.168.2.0/24

Validator rejects
Cert by child
grandchild is invalid

Grandchild
192.168.0.0/24, 10.0.0.0/24

Validator rejects
everything issued
by grandchild

ROA:
192.168.2.0/24

signs and publishes

signs and publishes

signs and publishes
Parent invalidating Grandchild

- **Parent** issues a shrunk certificate to the child
- And the **child** is unaware
- Now **grandchild** is invalidated
Reasons?

- Transfer timing gone wrong
- Parent may have to reissue, can't wait forever until child volunteers to shrink
- Parent made a mistake

Low likelihood, but high impact
Reconsidered - some invalid resource

Parent
192.168.0.0/16, 10.0.0.0/8

Child
192.168.2.0/24

Grandchild
192.168.2.0/24, 10.0.0.0/24

Validator accepts ROA for 192.168.2.0/24 by grandchild

Validator rejects only 10.0.0.0/24 on cert by child

ROA:
192.168.2.0/24

signs and publishes

signs and publishes

signs and publishes
Reconsidered - no valid overclaims

Validator rejects only 10.0.0.0/24 on cert by child

Validator rejects ROA for 10.0.0.0/24

Parent
192.168.0.0/16, 10.0.0.0/8

Child
192.168.2.0/24

Grandchild
192.168.2.0/24, 10.0.0.0/24

ROA:
10.0.0.0/24
Could this introduce new problems?

• Less incentive for child to clean up
  - There are warnings
  - And actually it may not be the child's fault
  - We believe this is better than having invalid

• Parent can revoke only specific resources used by grandchild with less collateral damage for layer-9 reasons
  - But, really, would invalidating all of grandchild stop them?
  - We have a bigger problem if this happens
Why we think this is a good idea

- Limit the impact of inconsistent resources to just those resources
  - Overclaims are never seen as valid
  - Works in running code, not seen warnings in production

➡ Change this to a low likelihood, lowest possible impact problem
Questions