

# PrivCand + Txid Alignment

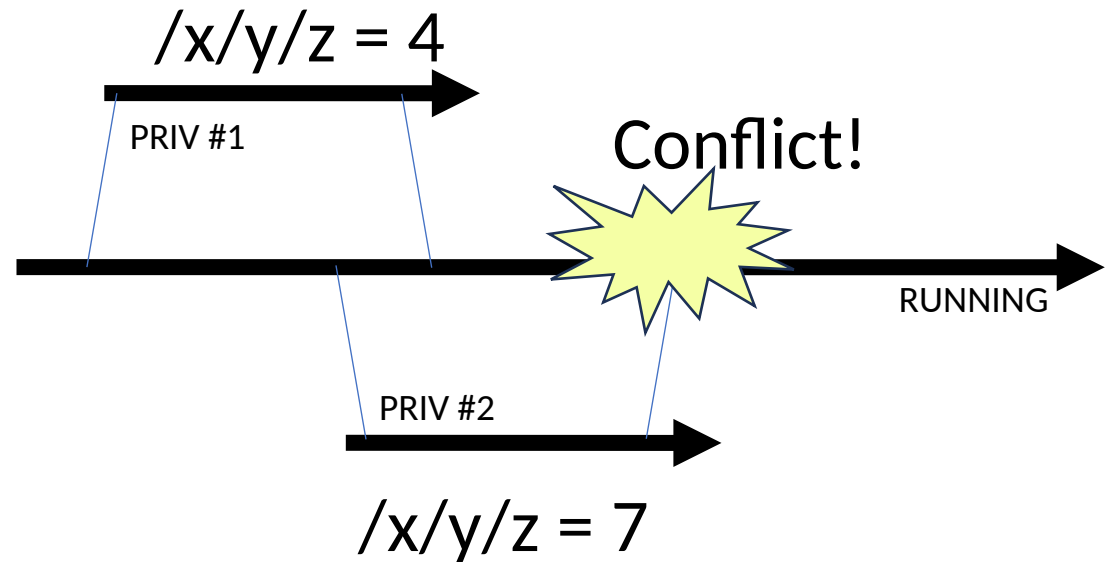
IETF 119, Mar 2024

Jan Lindblad <[jlindbla@cisco.com](mailto:jlindbla@cisco.com)>

# PrivCand + Txid

Complementary functionality,  
even better together — if aligned

Both PrivCand and Txid have a  
"conflict" concept



# Conflict Detection

## PrivCand-02 Detection

- IFF same leaf changed to different values in running and privcand  $\Rightarrow$  conflict

## Txid-03 Detection

- IFF client specifies txid for node (leaf, container, list item) and the same node (or descendants for containers, list items) is changed  $\Rightarrow$  conflict

Gives client control over exactly what changes matter to client, also in parts not changed by client

# Conflict Resolution

PrivCand-02 Resolution modes

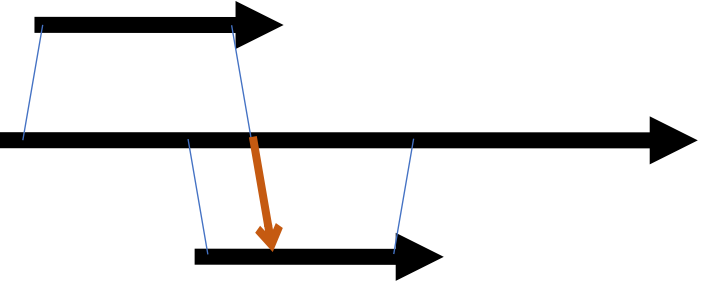
- Ours ("ignore")
- Theirs ("overwrite")
- Stop ("revert-on-conflict")

Single resolution mode applied to all conflicts in privcand at once (atypical way of merging diffs)

Revert-on-conflict is only allowed if server implements privcand as snapshot.

A snapshot implementation is good for many use cases but does not scale well to large configs or high volumes of changes.

# Snapshot or Diff vs. Running



## Snapshot Copy on create

- When a privcand is created, a full copy of running is made

## Snapshot Copy on write

- When a privcand is created, it is empty, backed by running
- On other's updates to running, the privcand is updated to hold original running config

## Diff vs. Running (no copying)

- When a privcand is created, it is empty, backed by running
- On other's updates to running, the privcand is not updated, making the changes show

# PrivCand-02 Proposed Changes

- Allow other conflict detection mechanisms
- Allow revert-on-conflict for all implementation styles

Thank you