Resource Transfer Protocol and Transfer Authorization Object (TAO)

Author: Edric Barnes <ebarnes@bbn.com>
Presenter: David Mandelberg <dmandelb@bbn.com>
draft-barnes-sidr-tao-00
IETF 89

Background

- Geoff Huston pointed out that transferring resources is somewhat complicated
- Steve Kent suggested a way to automate resource transfers without modifying RFC3779 semantics
- Edric Barnes fleshed out Steve's idea: draft-barnes-sidr-tao-00

Context

- This is not a replacement for the procedural activities associated with INR transfer
- It is a way to help automate RPKI repository maintenance
- It makes use of two new message types based on the up/down protocol (RFC 6492), and a new CMS signed object

Terminology: Actors

- Source: CA that wants to transfer INRs
- Recipient: CA that wants to receive INRs from the source
- Swing point: lowest common ancestor of the source and recipient
- Source path: source, and all its ancestors up to and including the swing point
- Recipient path: recipient, and all its ancestors up to and including the swing point

Terminology: Transfer Types

- Live: INRs are in use during the transfer, so the source and recipient must simultaneously hold the INRs for an overlap period
- Unused: no overlap period is required

High-level Overview

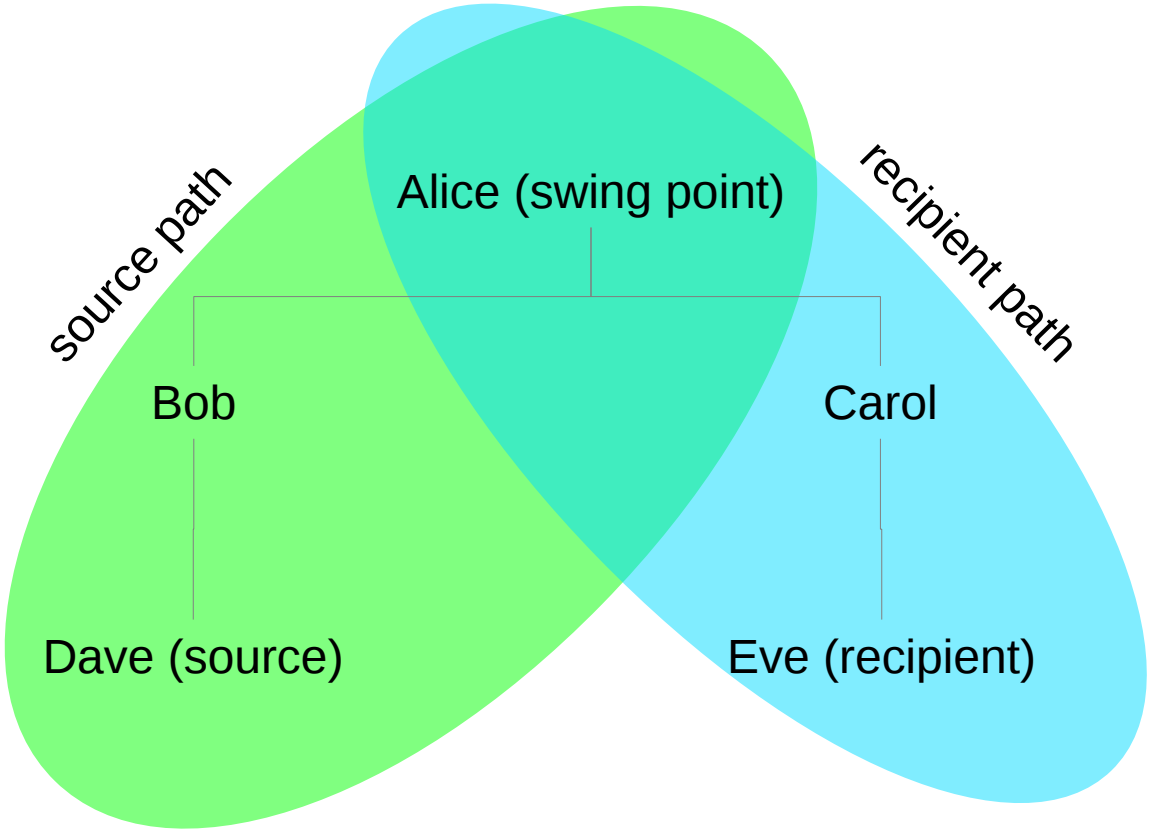
1. Source publishes a Transfer Authorization Object (TAO) and informs the recipient
2. Source and recipient independently pursue transfer by each sending transfer_request messages to their own ancestors, recursively
3. Swing point receives transfer_requests from both paths and transfers the resources.
   - Swing point issues a certificate with the transferred INRs down the recipient path
   - CAs between the swing point and the recipient do the same
   - CAs between the swing point and the source relay success and, for a live INR transfer, await the end of the transfer period to remove the INRs from the source path

- Note: any CA along the path can reject the transfer.

Protocol Requirements

- Swing point exists
- Source and recipient don't re-key during transfer
- Source has not sub-delegated the resources being transferred
- Recipient already has a CA certificate
- Source is not an ancestor of recipient, or vice versa (already covered by existing procedures)

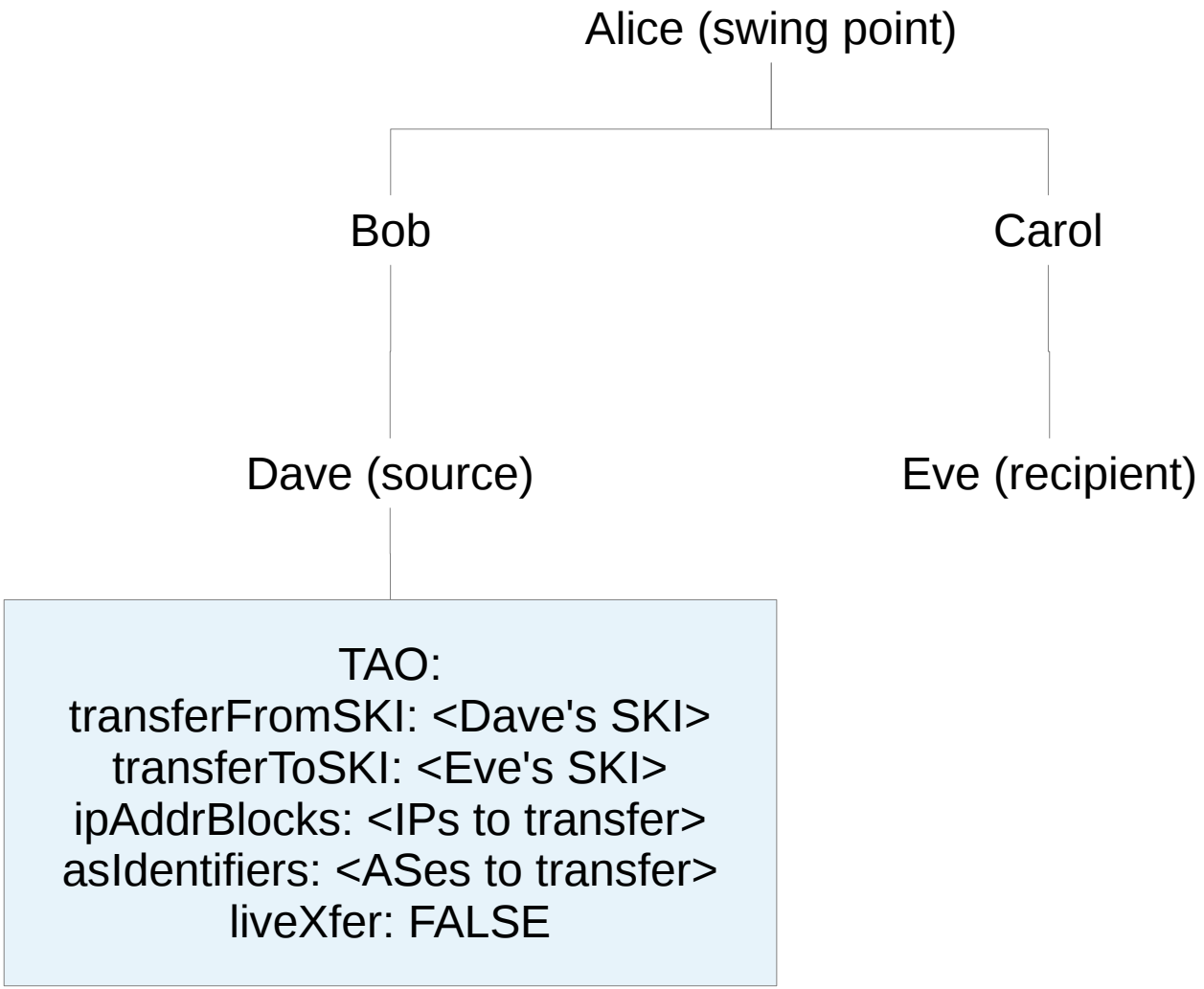
Scenario


source path
recipient path
Alice (swing point)
Bob
Carol
Dave (source)
Eve (recipient)

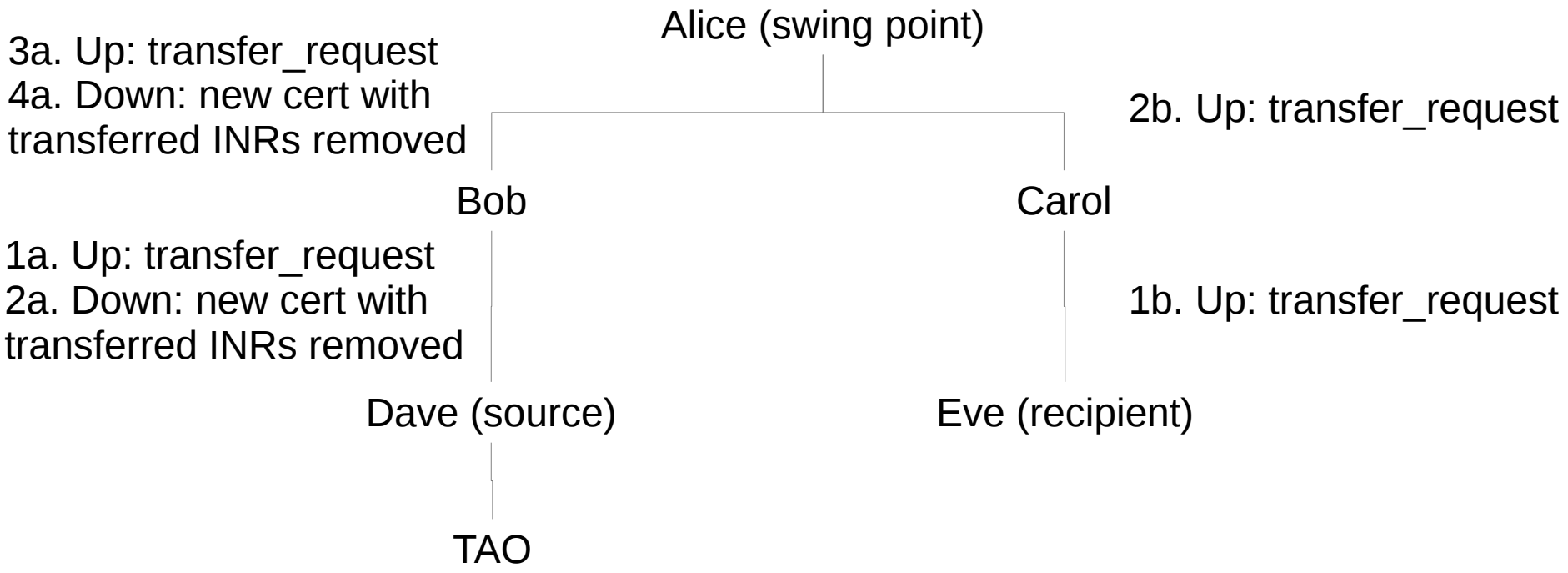
TAO

- A TAO is a CMS signed object conforming to RFC 6488
- The eContent is ASN.1 with the following fields:
  - transferFromSKI: SKI of the source
  - transferToSKI: SKI of the recipient
  - ipAddrBlocks, asIdentifiers: INRs to transfer
  - liveXfer: TRUE for a live INR transfer, FALSE for an unused INR transfer
  - overlapPeriod: minimum number of seconds that the source and recipient must both hold the INRs (live INR transfer only). If the recipient does not receive the INRs before the TAO's notAfter minus the overlapPeriod, the transfer is canceled.

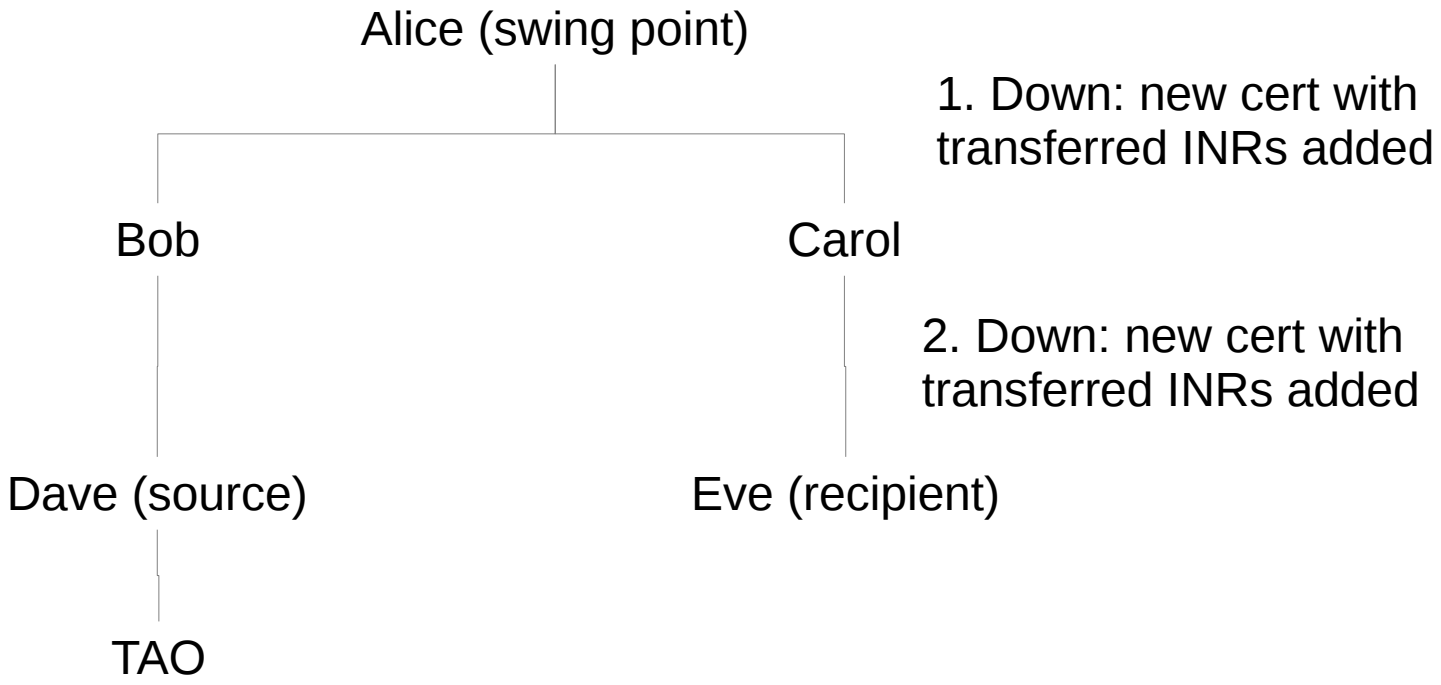
Unused INR Transfer Example: Start



Unused INR Transfer Example: Simultaneous transfer_requests

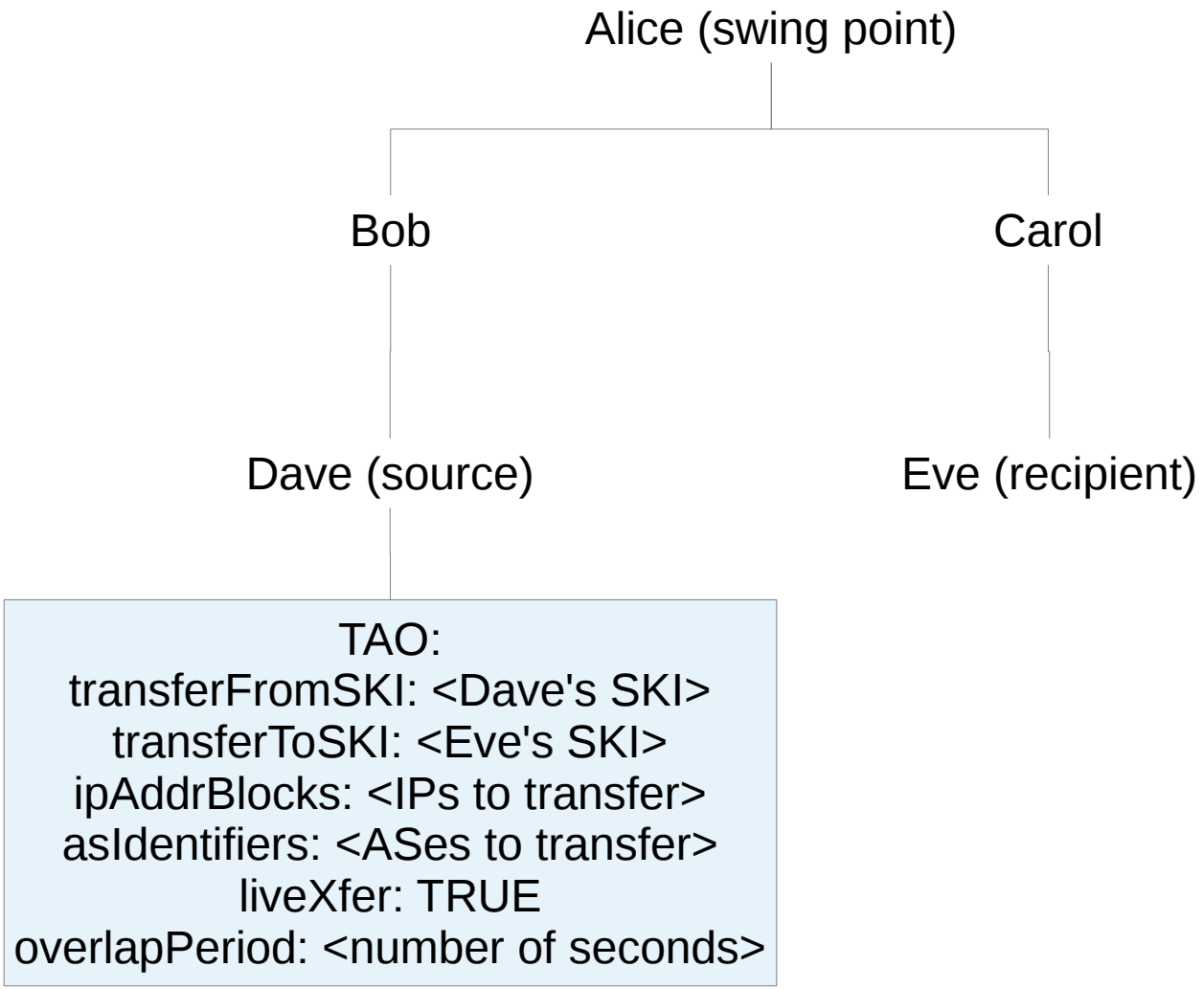


Unused INR Transfer Example: Recipient receives INRs

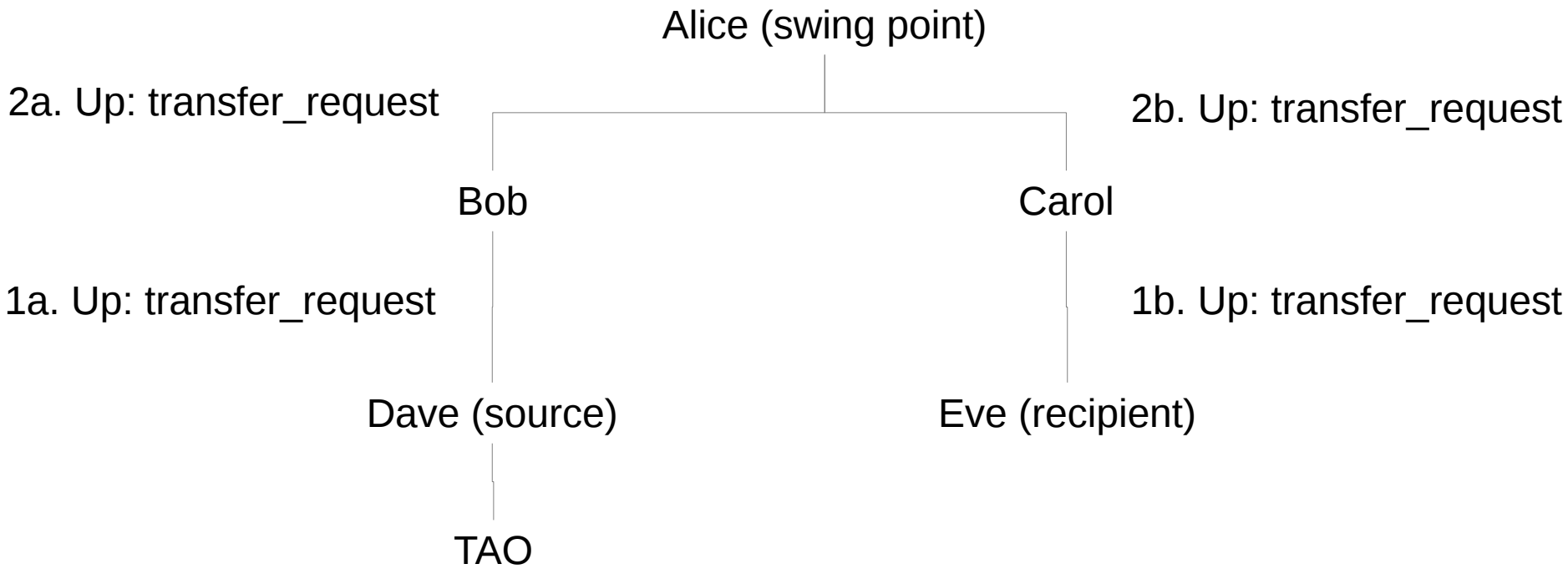


We're done!

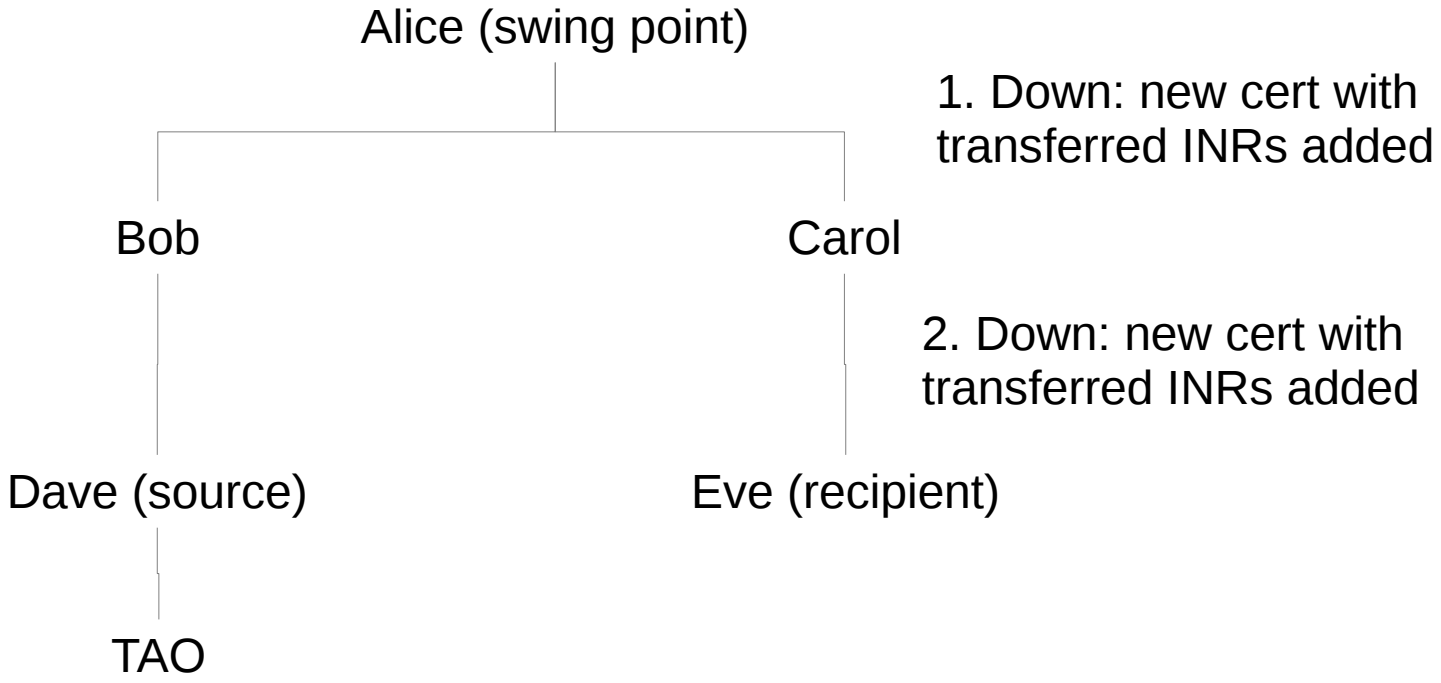
Live INR Transfer Example: Start



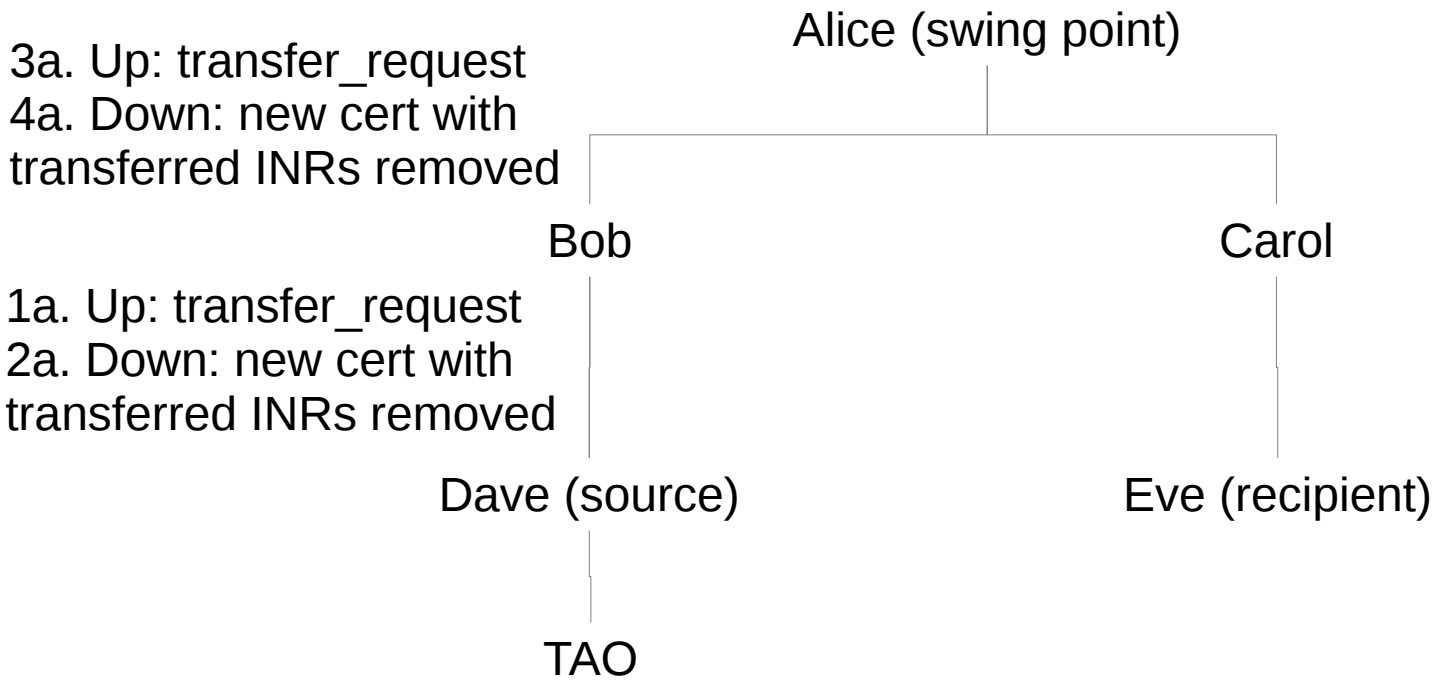
Live INR Transfer Example: Simultaneous transfer_requests



Live INR Transfer Example: Recipient receives INRs (before TAO's notAfter - overlapPeriod)



Live INR Transfer Example: Source relinquishes INRs (after TAO expires)



Recap

- Unused INR transfer:
  - Source path relinquishes INRs while recipient path pass transfer_requests up the path
  - Swing point receives two transfer_requests, and passes the INRs down the recipient path
- Live INR transfer:
  - Source and recipient paths both send transfer_requests up the paths
  - Swing point receives two transfer_requests, and passes the INRs down the recipient path
  - Source waits for the TAO to expire, then the source initiates relinquishment of the INRs

Questions?