Private candidates

draft-jgc-netconf-privcand-00

James Cumming

Robert Wills

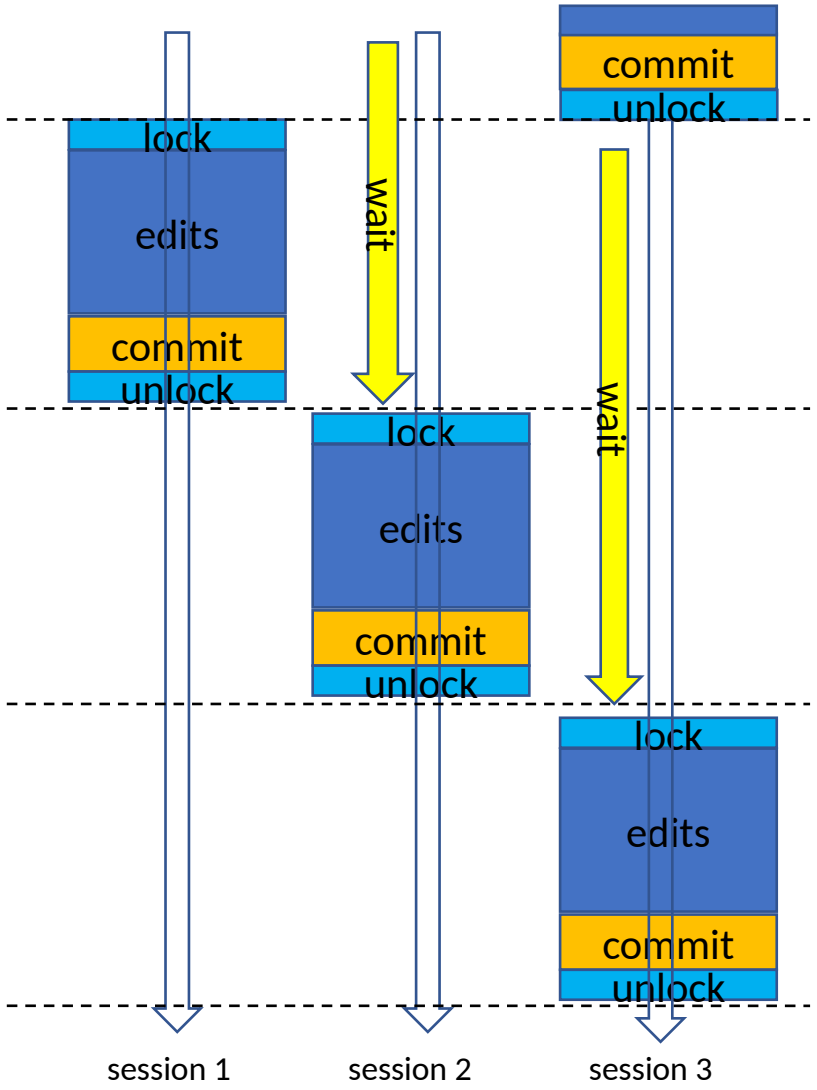
Problem space

- Existing Netconf “candidate” is a shared candidate
  - Multiple clients can make changes to the same datastore before committing
  - Therefore, one client may unwittingly commit another client’s changes
- Existing solution: Locking
  - Only allows one client to make changes at a time
- Many existing devices support the ability for a client to make changes and commit only those changes
  - “Private Candidate”
  - Want to standardize this

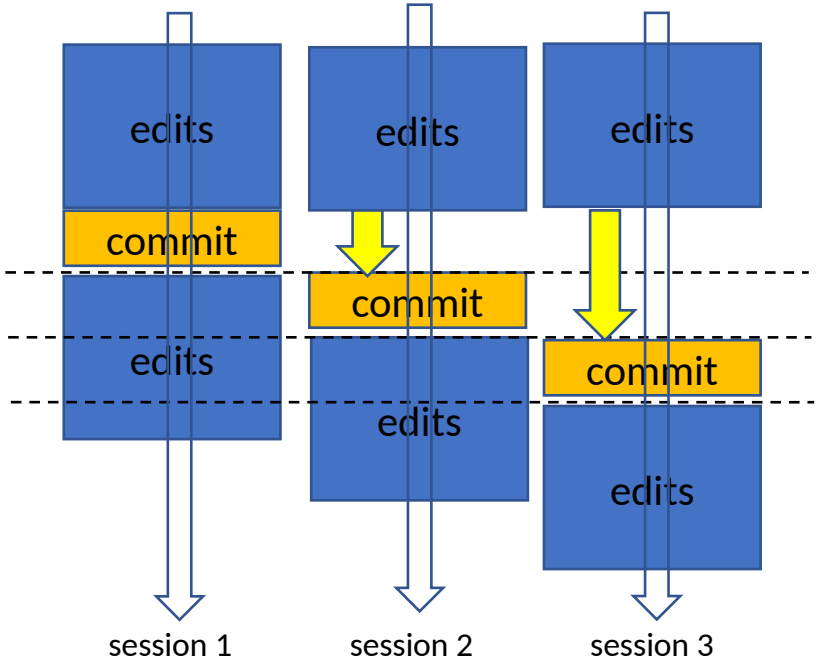
Locking the Global vs Using Private Candidates

Global candidate with locking

- Clients wait for other clients to complete entire update cycle before starting their edits (wait on lock)



Private candidates



- Only commit stage is serialized
- Edits parallelized

High level aims

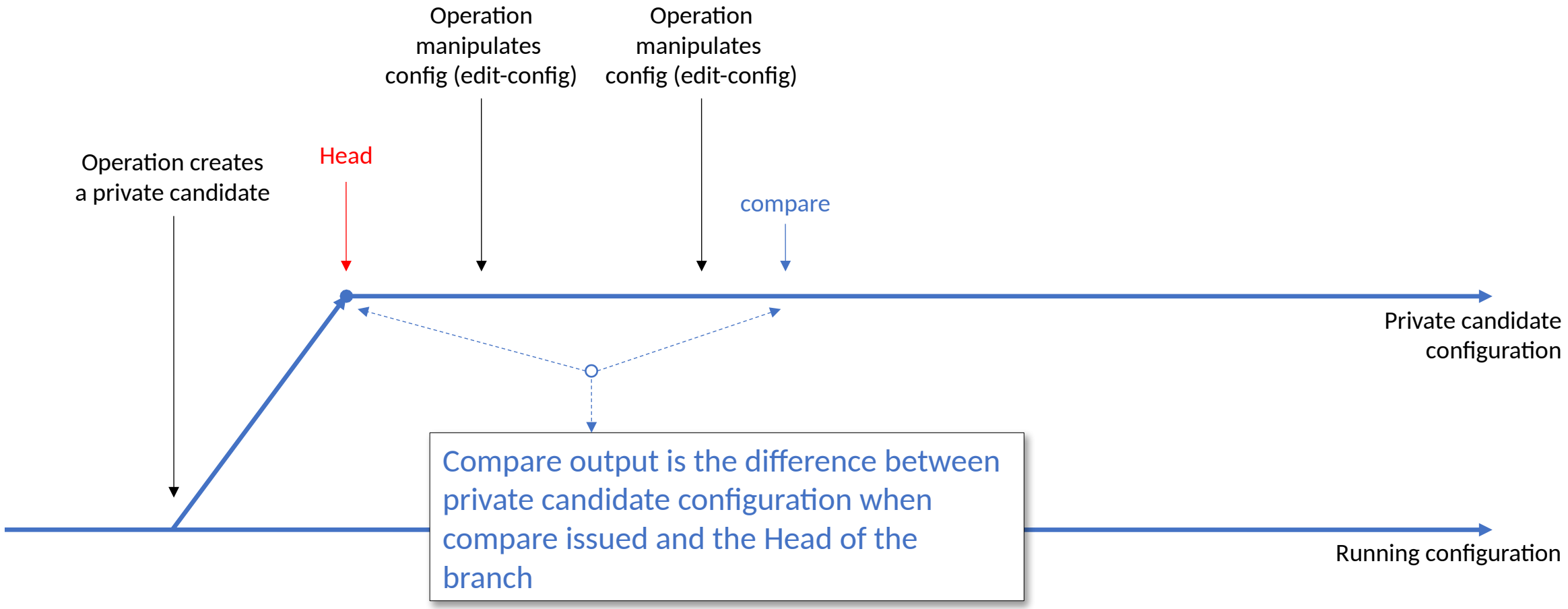
- Define a what a private candidate is
- Define how a private candidate is created and manipulated using NETCONF
- Define how and when conflict resolution is performed

- Is this a problem space the working group is interested in?

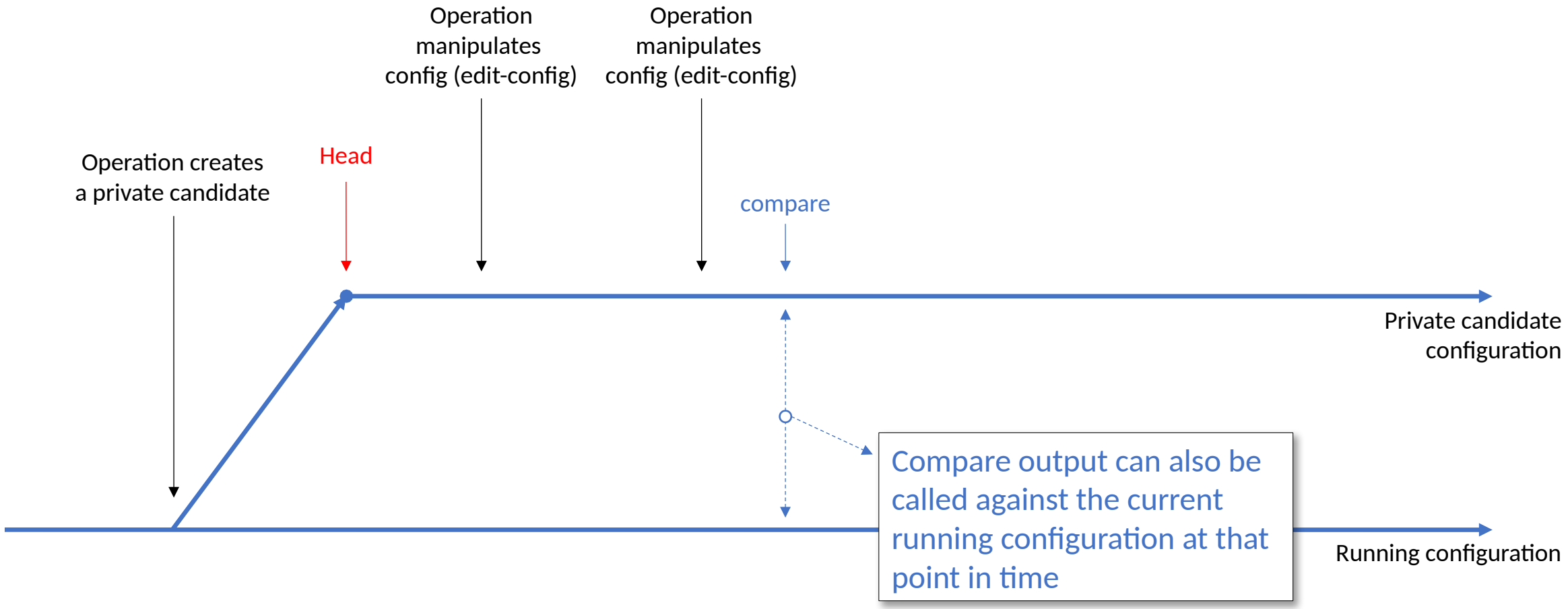
What is a private candidate

- A candidate configuration datastore to which
  - Is not visible to anyone else
  - Is not accessible to anyone else
  - Provides a workspace for a user to stage new configurations
- Created when the first NETCONF operation requires it by branching the running configuration
- Contains a full router configuration including all changes made to it

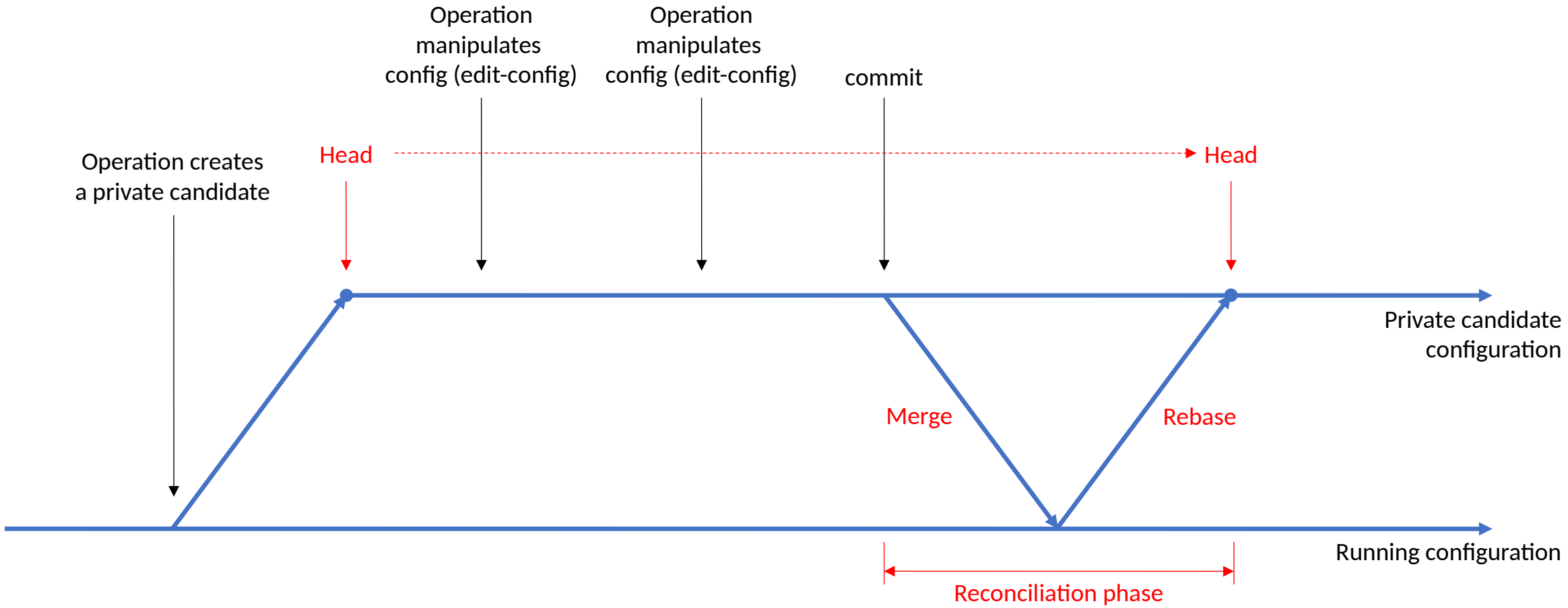
Private candidate conceptual use



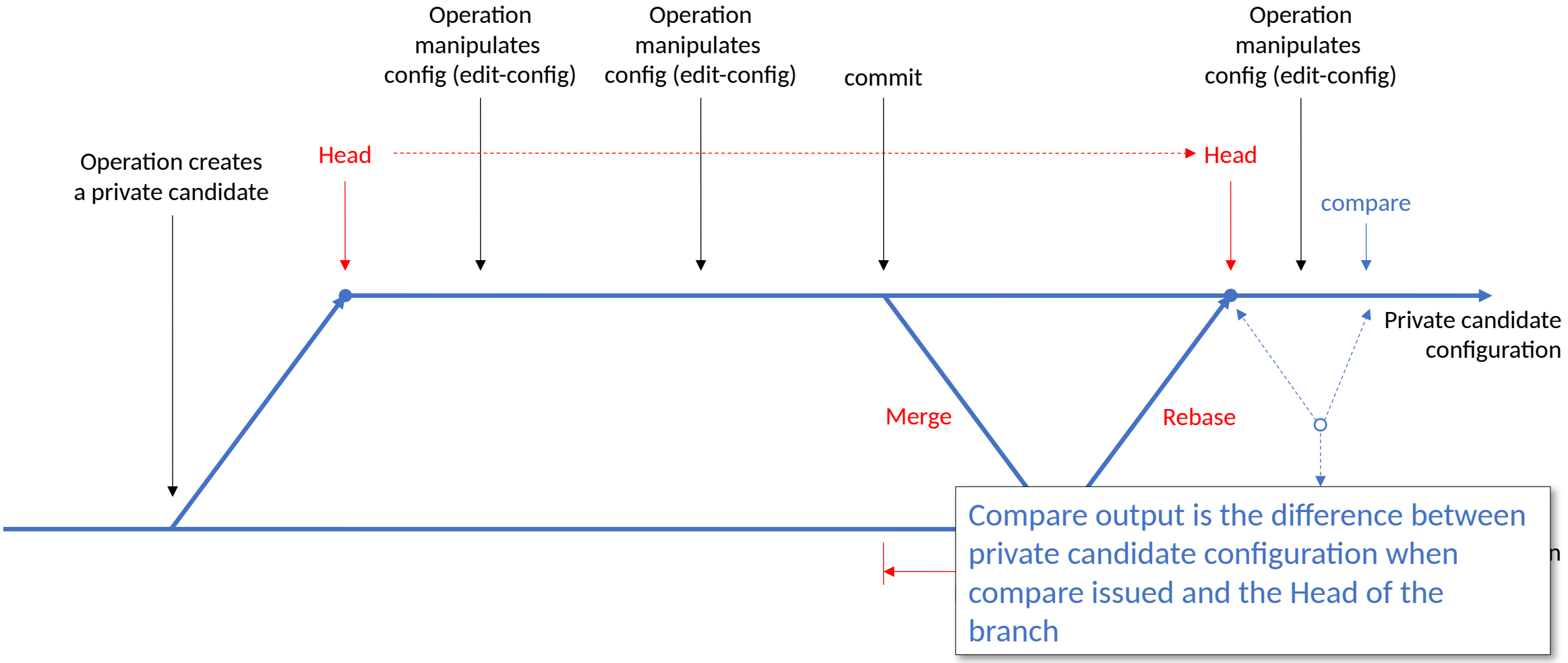
Private candidate conceptual use



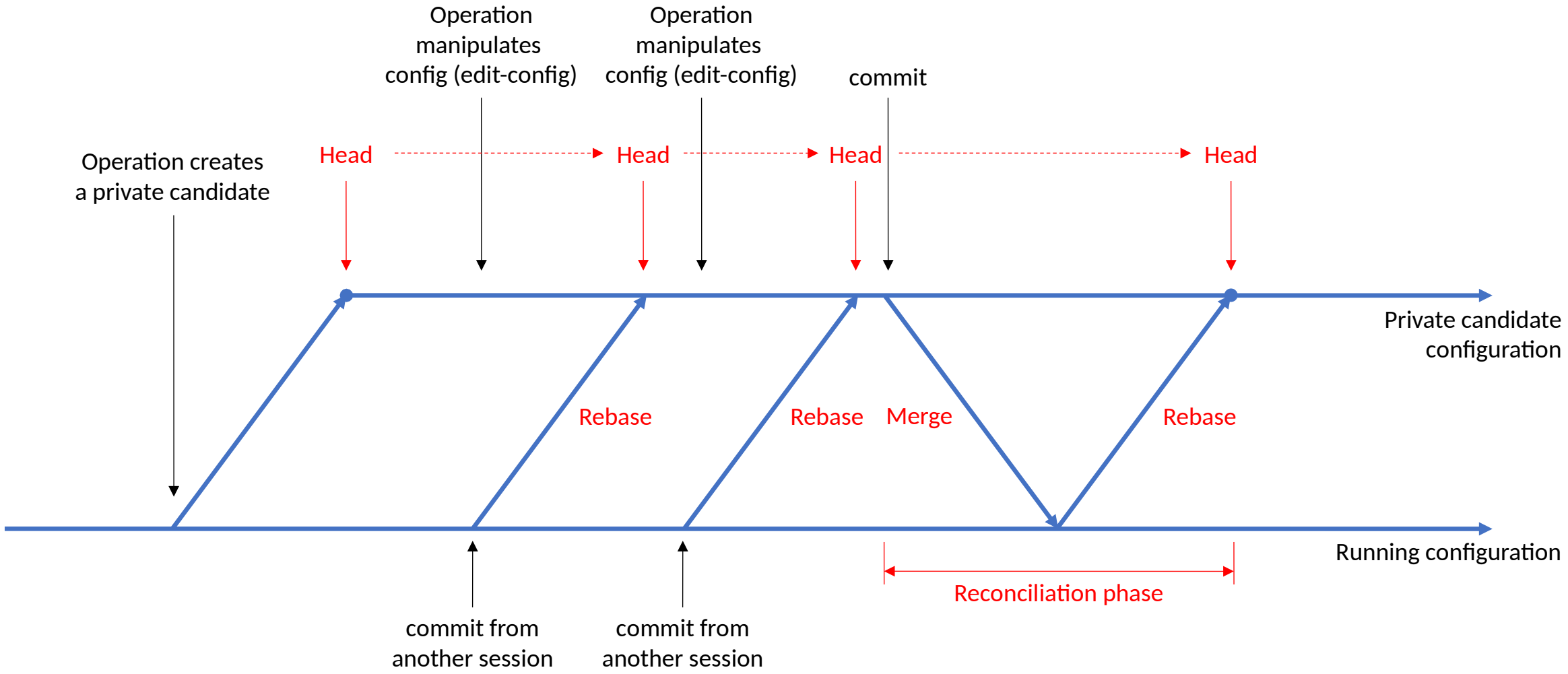
Private candidate conceptual use
(Static branch mode)



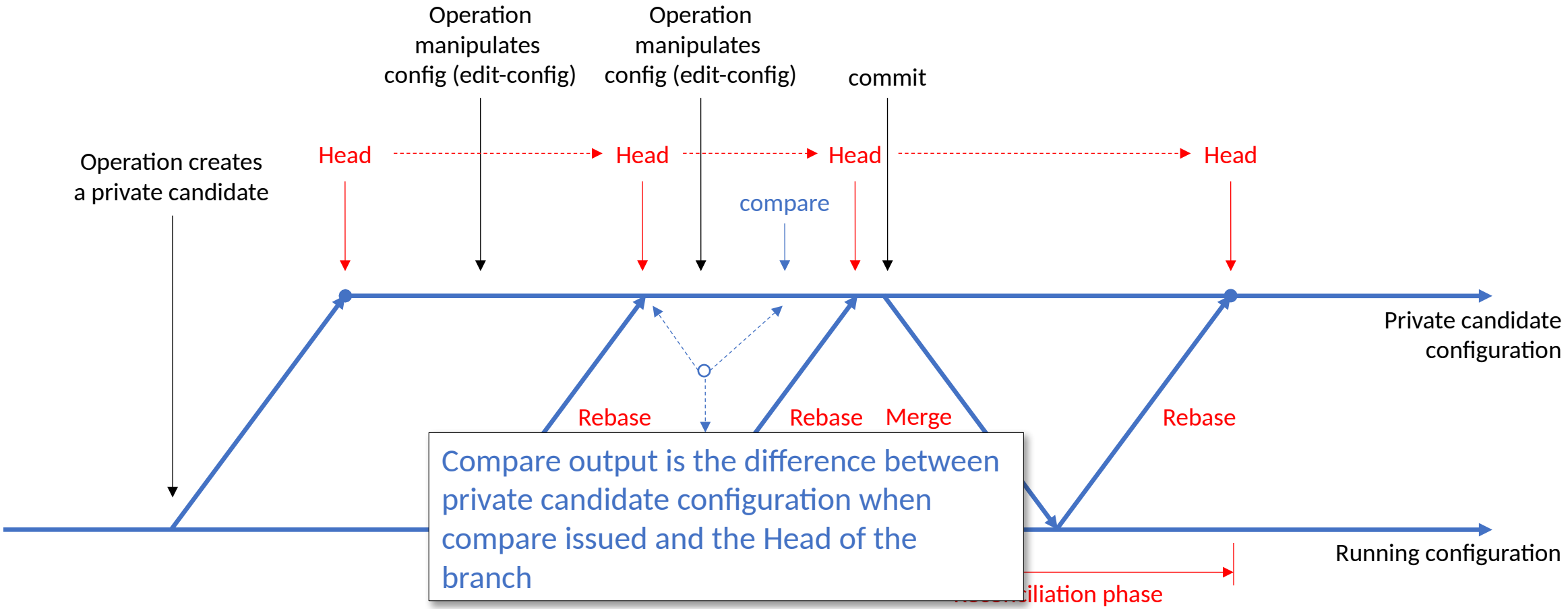
Private candidate conceptual use
(Static branch mode)



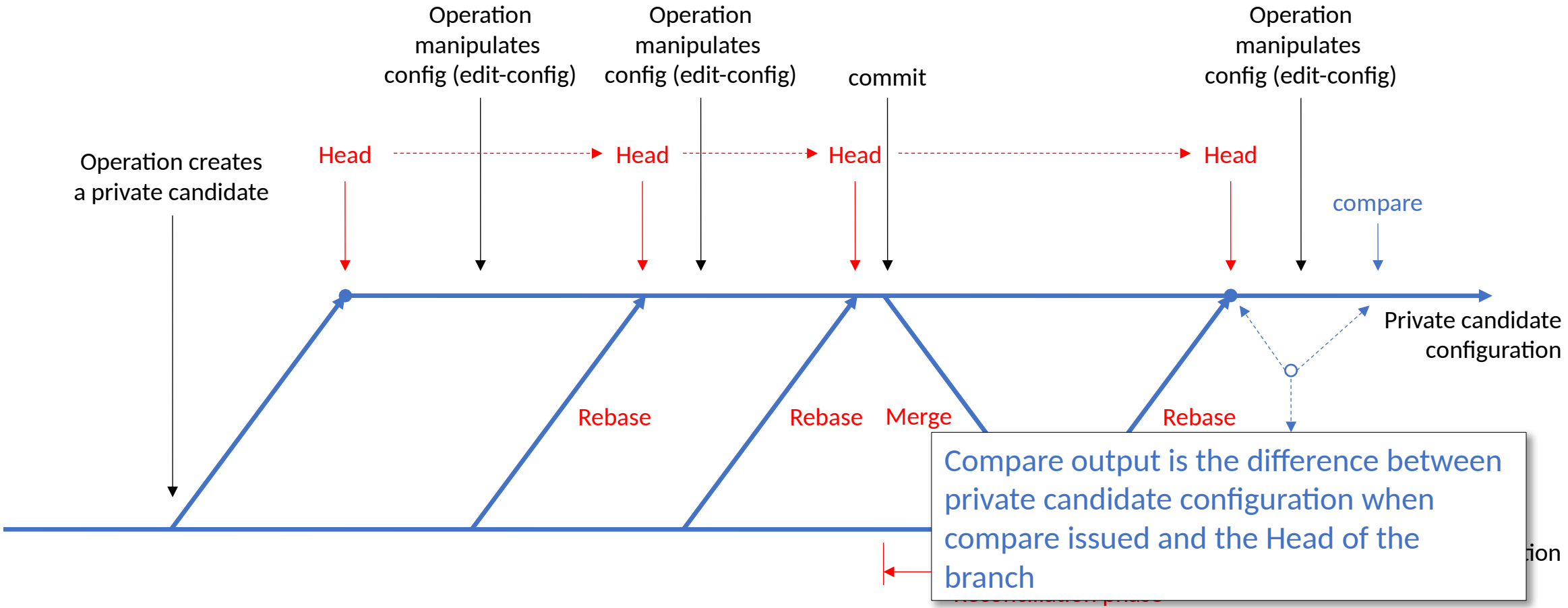
Private candidate conceptual use (Continuous rebase mode)



Private candidate conceptual use (Continuous rebase mode)



Private candidate conceptual use
(Continuous rebase mode)



Interaction methods

- Client capability
  - New “private candidate” capability indicates that the client/server wants their candidate datastore to be a private candidate
  - Easy compatibility with existing Netconf clients
- NMDA private-candidate datastore
  - Define a new NMDA datastore
  - Existing get-data, edit-data operations against this datastore
- Not necessarily mutually exclusive

Collision detection

- Multiple places where collisions could occur:
  - Static branch mode: When committing
  - Continuous rebase mode: When running config changes
- How might collision detection be resolved?
  - An option for ”merge” is an “update” RPC
  - “Force” option
  - Still actively thinking about options here!

Working items

- Reconciliation methods
  - Select one / support both
- Interaction methods
  - Select one / support both
- Conflict resolution