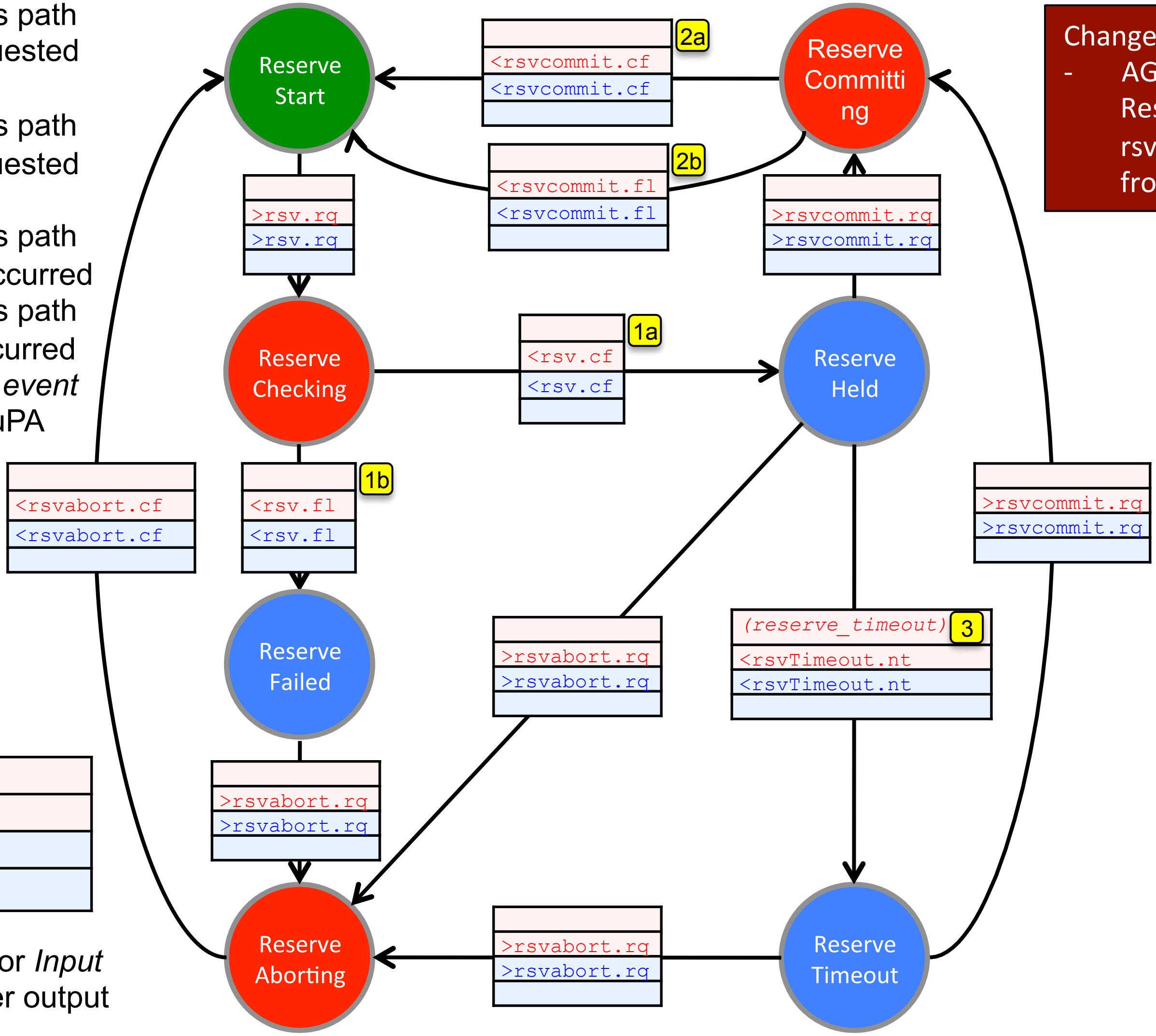
Motivation for Changes in RSM (CSv2.1)

- The current CSv2.0 RSM does not allow the AG to reflect that a uPA has transitioned to a *ReserveTimeout* state
  - When both the AG and uPA are in the *ReserveHeld* state and a uPA expires the resource hold, **only the uPA will transition to the *ReserveTimeout* state**
  - The *rsvTimeout.nt* notification generated by the timed-out uPA is passed up the tree to the uRA, but does not alter the state of any AG along the upstream control-plane path
  - Queries (especially by 3rd party applications) to any AG along control-plane path will see the reservation state in *ReserveHeld*

RSM: Reservation State Machine (CS v2.1)

Notes:
1a: uPA transits this path if the resource requested is available
1b: uPA transits this path if the resource requested is not available
2a: uPA transits this path if no timeout has occurred
2b: uPA transits this path if a timeout has occurred
3: *reserve_timeout event* can only occur on uPA



Changes from CS v2.0

- AG can now transition to ReserveTimeout on receipt of rsvTimeout.nt notification from uPA

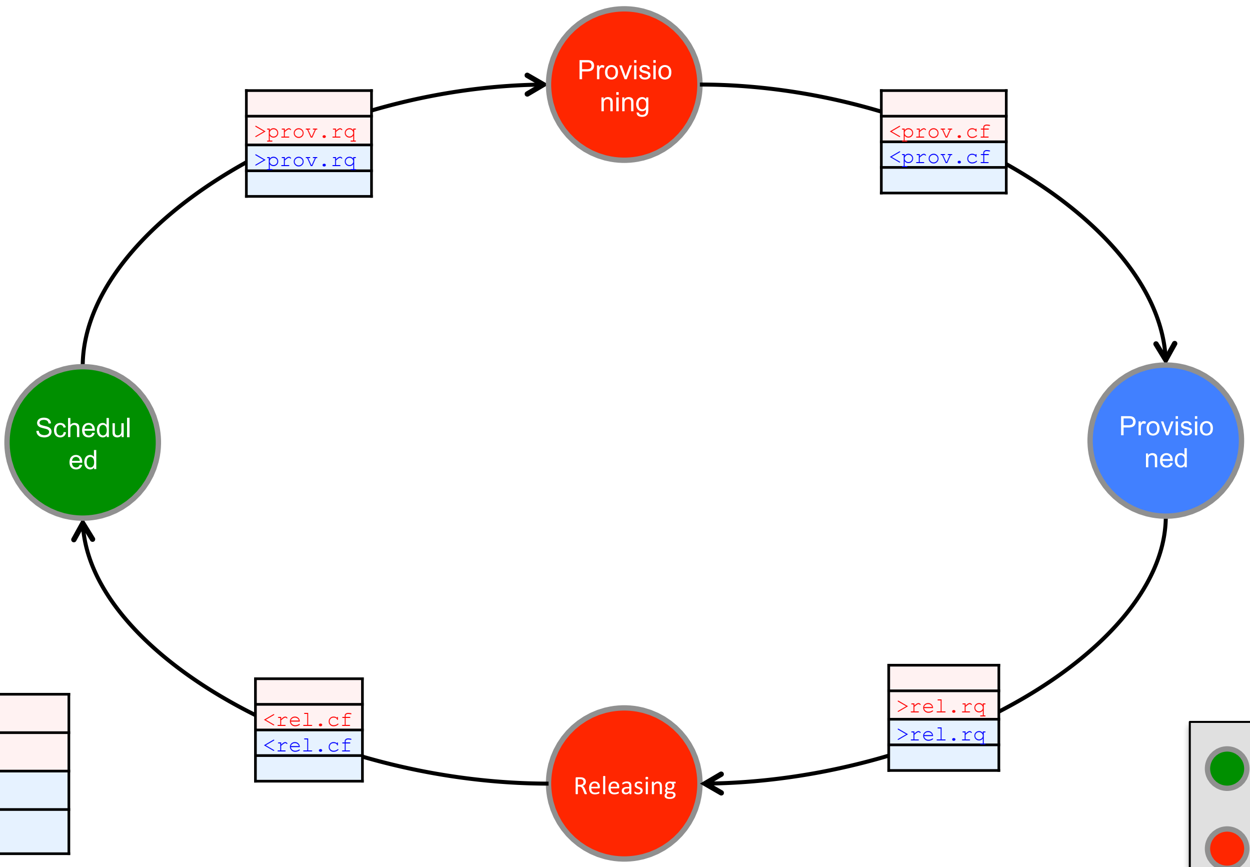
Input event+
Input message+
Output message
Output event

+Either *Input event* or *Input message* can trigger output (logical disjunction)
“>” = downstream message
“<“ = upstream message

● Initial State
● Transitional States
● Stable States

PSM: Provisioning Lifecycle

No changes from CS v2.0



Input event[+]

Input message[+]

Output message

Output event

[+]Either *Input event* or *Input message* can trigger output (logical disjunction)
“>” = downstream message
“<“ = upstream message

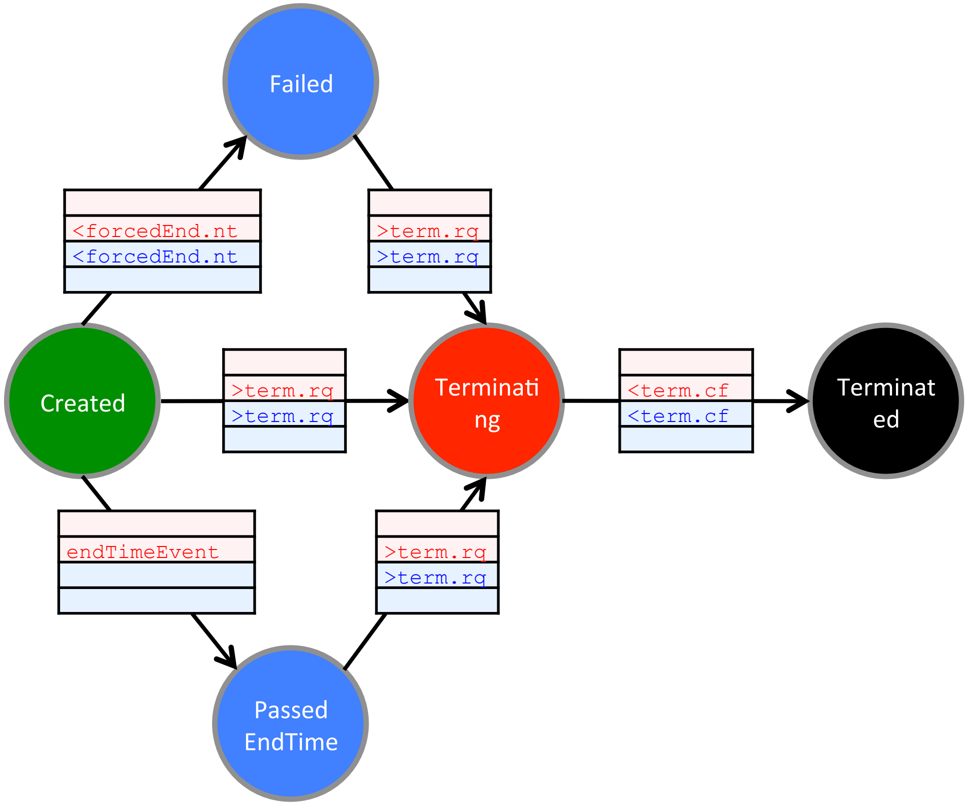
🟢 Initial State

🔴 Transitional States

🔵 Stable States

LSM: Termination Sequence

Changes from CS v2.0
- forceEnd changed to forceEnd.nt



Input event[+]
Input message[+]
Output message
Output event

[+] Either *Input event* or *Input message* can trigger output (logical disjunction)
“>” = downstream message
“<“ = upstream message

● Initial State
● Transitional State
● Stable States
● Final State