

OGF NSI CS State Machine Delft v4 with P2PC Modify

July 31, 2012

Chin Guok chin@es.net

Henrik Thostrup Jensen htj@nordu.net

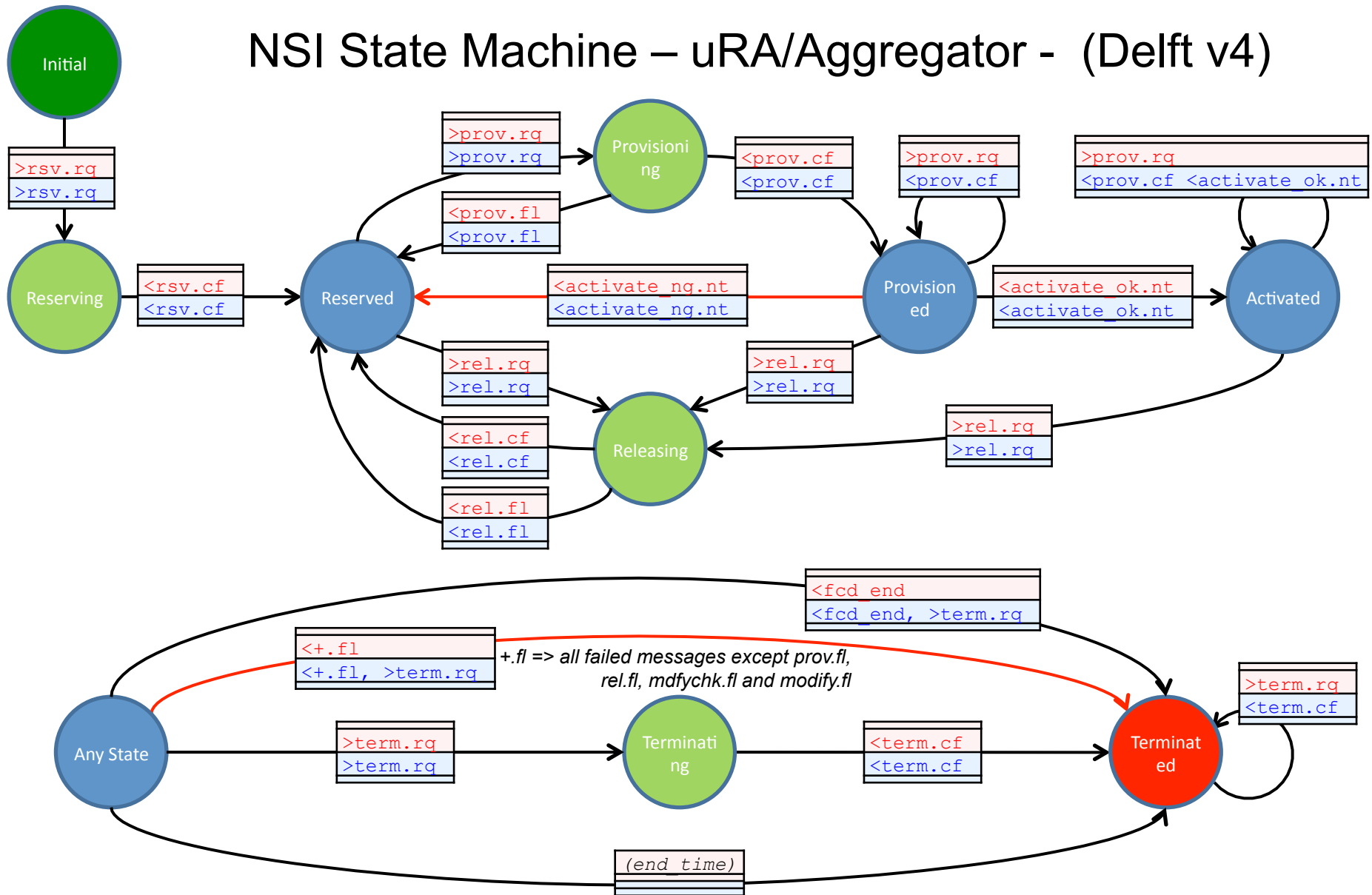
John MacAuley john.macauley@surfnet.nl

Tomohiro Kudoh t.kudoh@aist.go.jp

Summary

- This proposal introduces modify function by just adding operations/states for modify
 - Pseudo-2PC, which has been used in the reservation sequence of the original SM, is used in modify too.
 - The original SM has not been changed at all (slide 3 and 5 are unchanged). Just added operations/states for modify (slide 4 and 6)

NSI State Machine – uRA/Aggregator - (Delft v4)

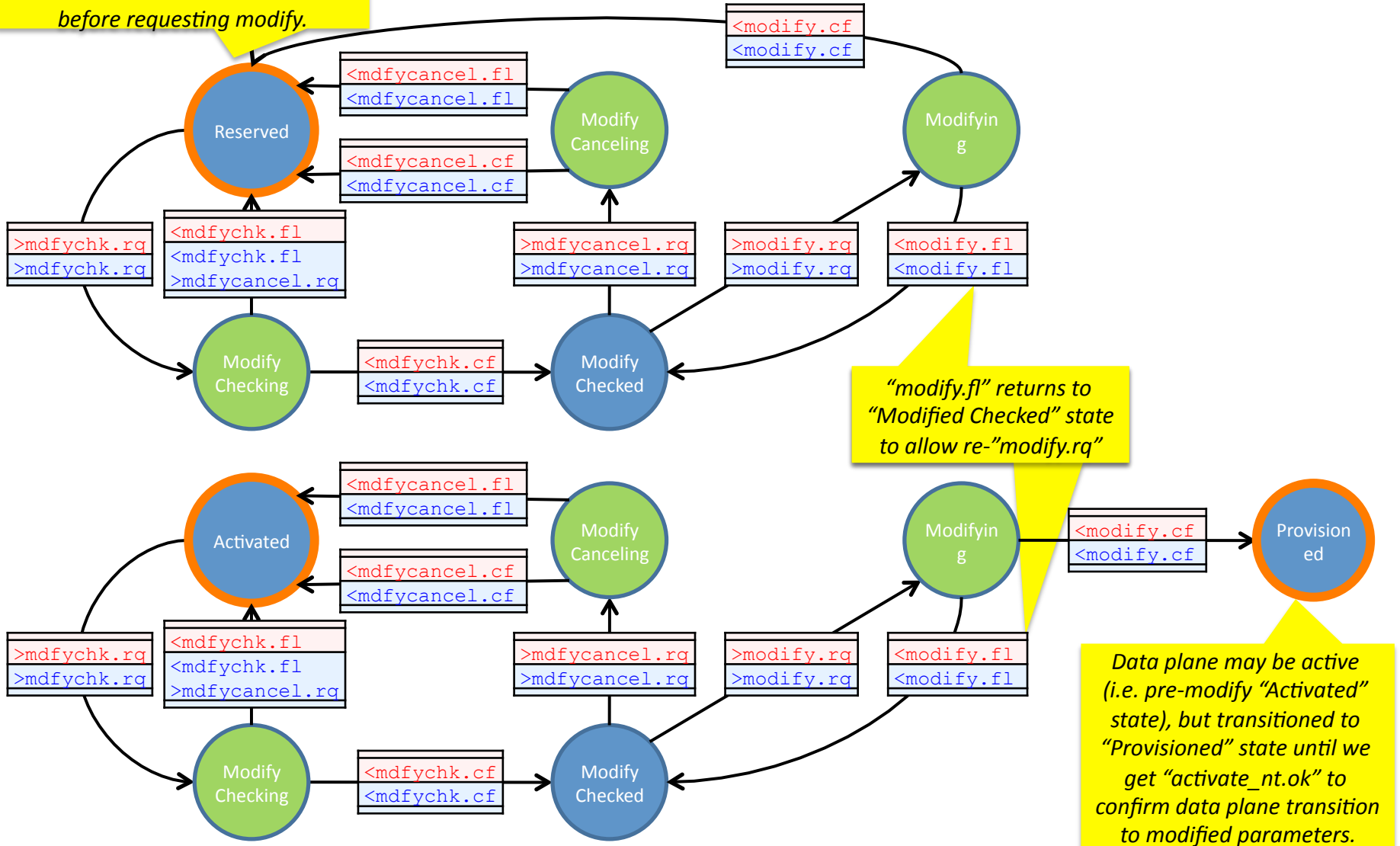


● Initial State
 ● Transitional States
 ● Stable States
 ● Final State

NB: Requests received in this state is queued and processed only when it transitions to a Stable State.

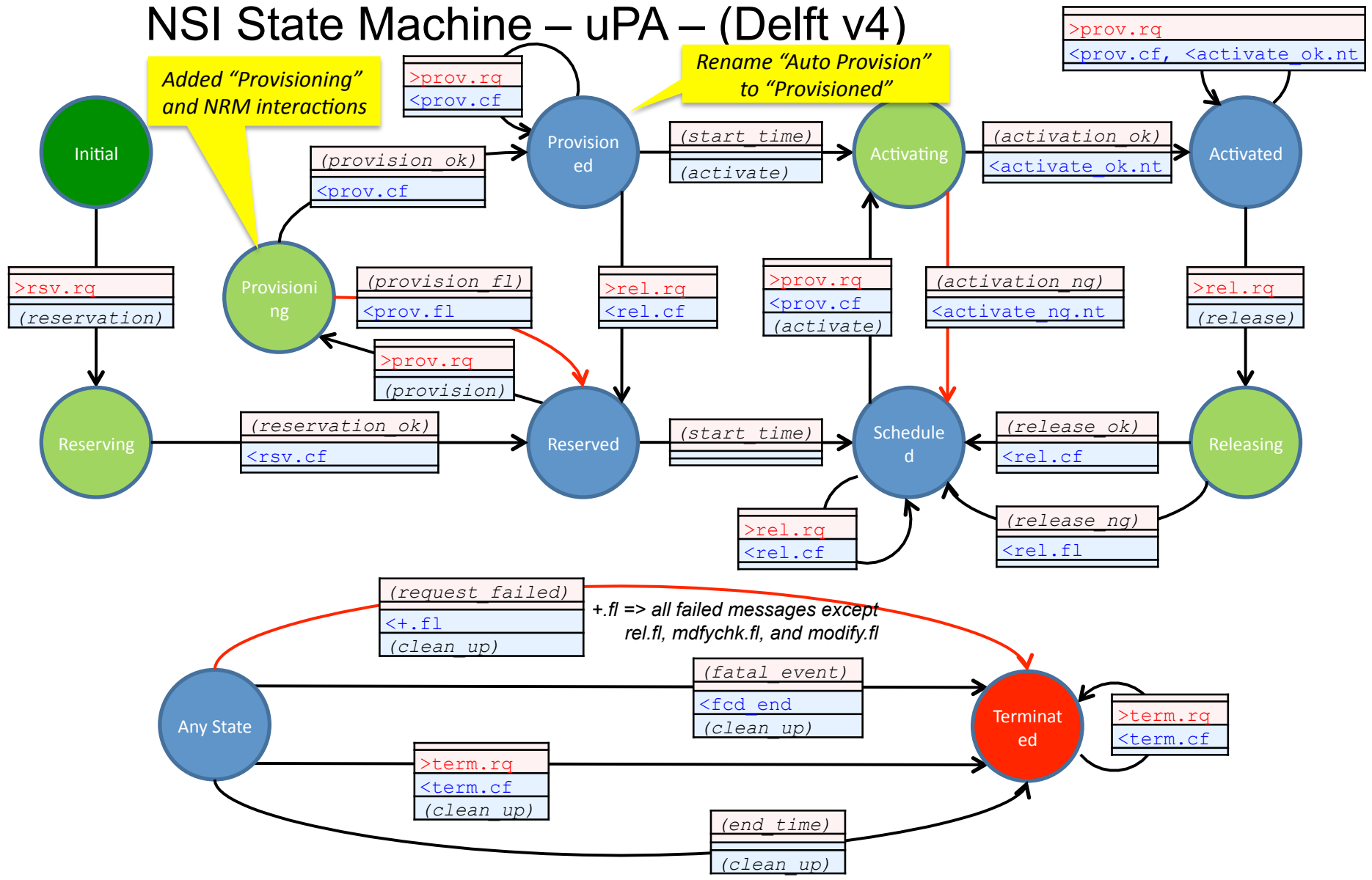
NSI State Machine – uRA/Aggregator - (Addendum for Modify)

If base SM is in "Provisioned" state, user must release (i.e. ">Rel.rq") and return to "Reserved" state before requesting modify.



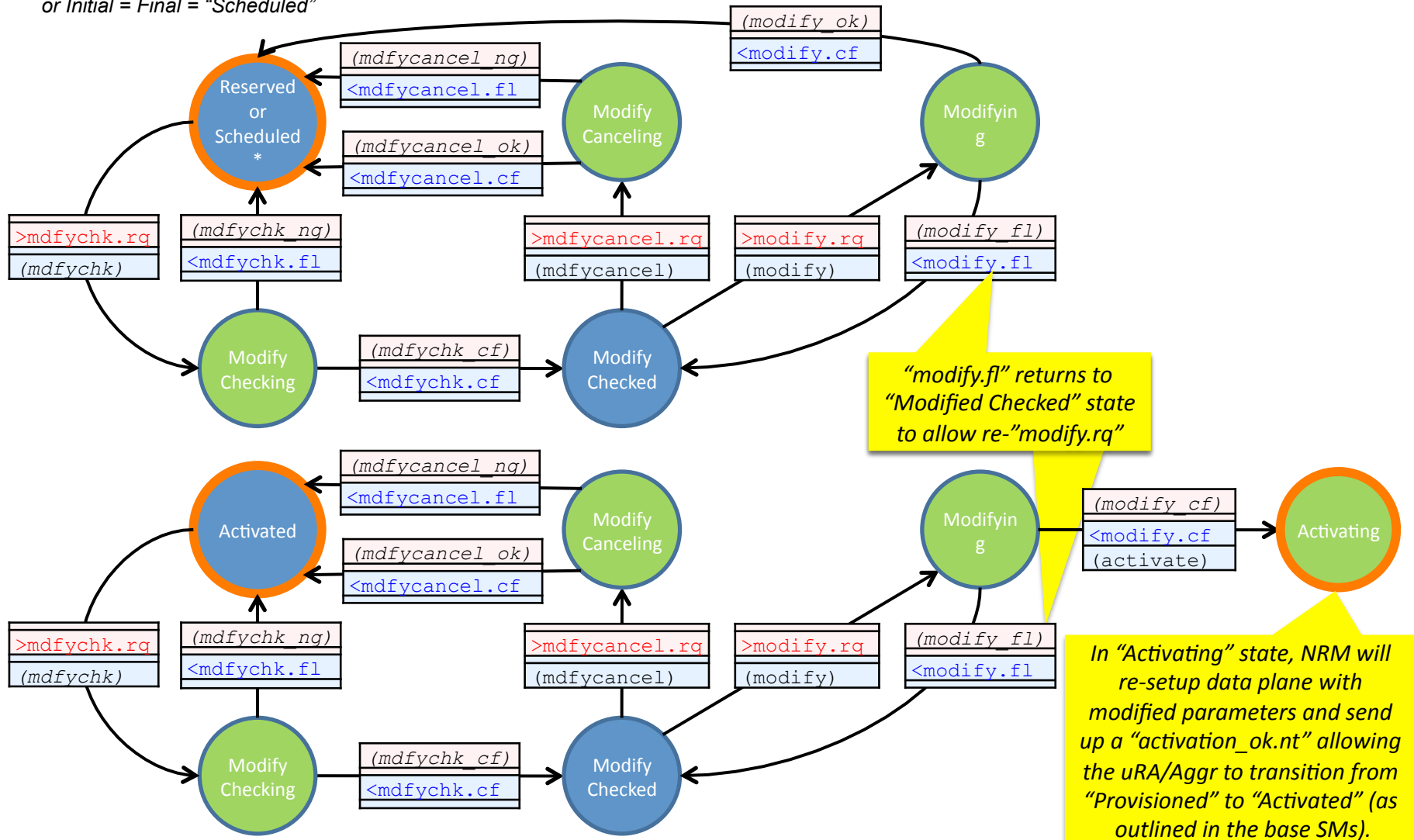
○ Base SM states
NB: Refers to states in the corresponding (i.e. uRA/Aggr, or uPA) base state machines.

NSI State Machine – uPA – (Delft v4)



NSI State Machine – uPA – (Addendum for Modify)

*Initial and final states are identical,
e.g. Initial = Final = "Reserved",
or Initial = Final = "Scheduled"



○ Base SM states
NB: Refers to states in the corresponding (i.e. uRA/Aggr, or uPA) base state machines.