OGF NSI CS State Machine Delft v1 (Proposed)

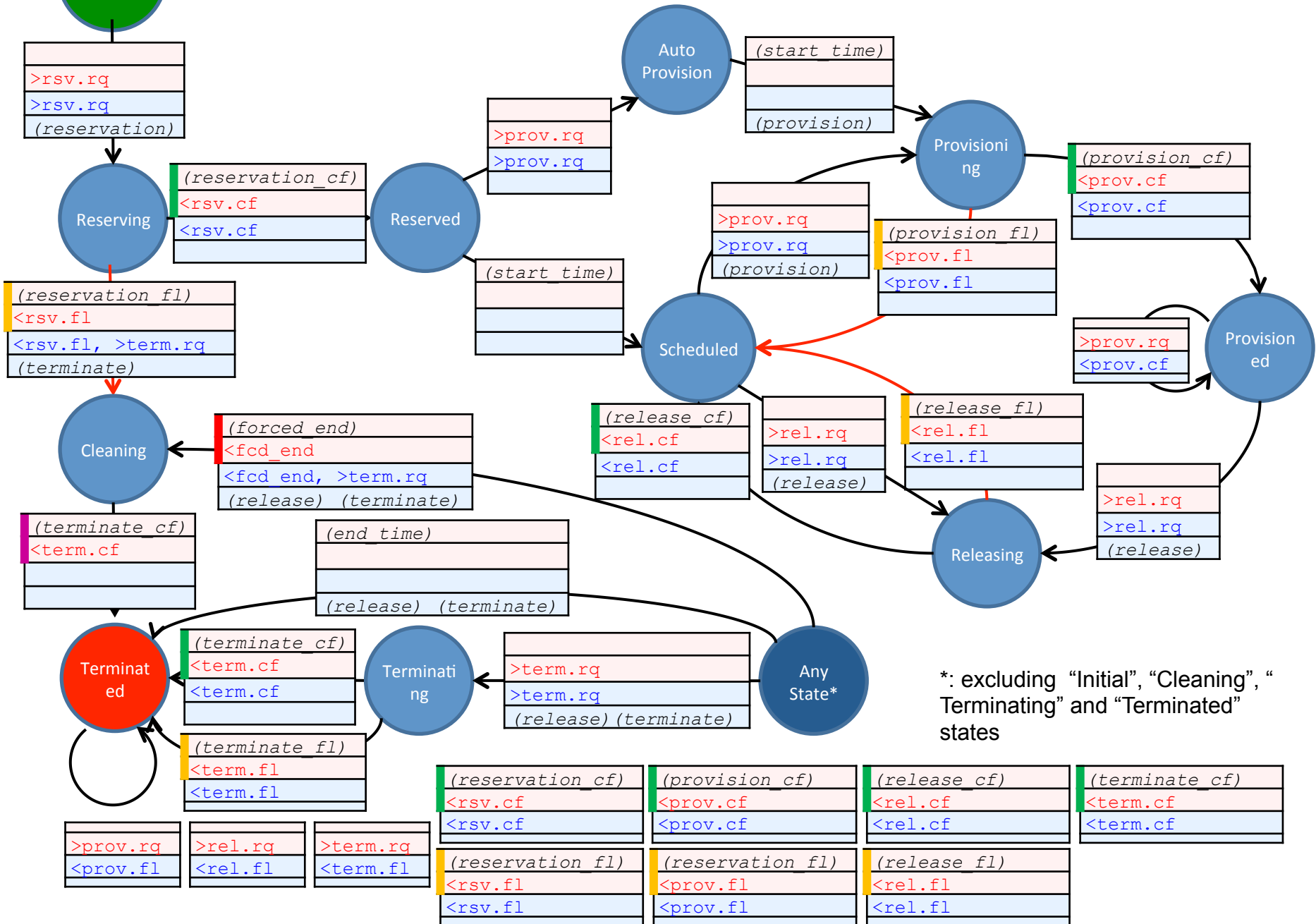OGF35 June 17-19, 2012

Henrik Thostrup Jensen, htj@nordu.net
Chin Guok, chin@es.net
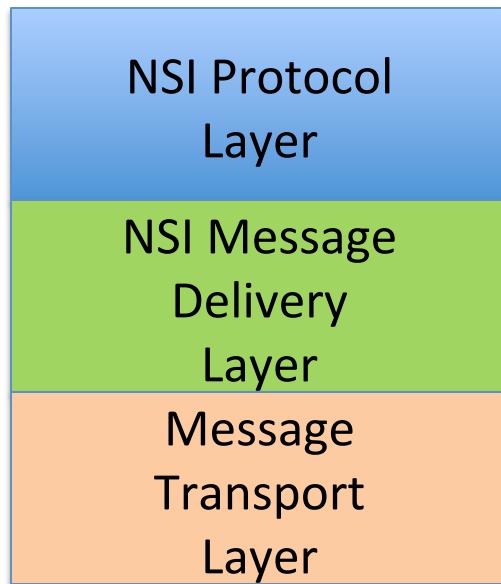**Tomohiro Kudoh, t.kudoh@aist.go.jp**

NSI State Machine (v1.0SC)

*: excluding "Initial", "Cleaning", "Terminating" and "Terminated" states

NSI message delivery layer (MDL)

NSI Protocol Layer

← State Machine works here
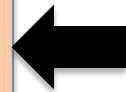
NSI Message Delivery Layer

← New layer which confirms delivery of message to all immediate children including uPA in the same NSA

"ensures" might be too strong a statemen, changed to "confirms"

Message Transport Layer

← Peer-to-peer message delivery
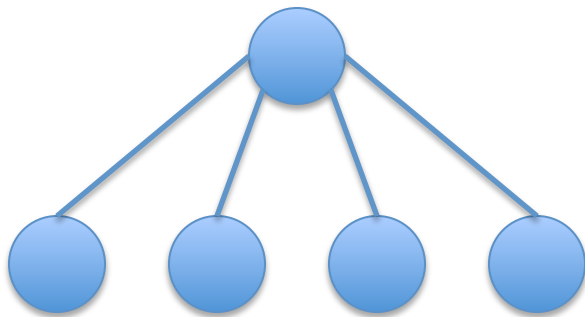
- MDL does
    - Aggregation of replies from children
        - all-ok/one-or-more-failed
    - Timeout/Re-try (as hard as possible)
- If MDL returns "fail", it is fatal.

State machines and MDL, NRM
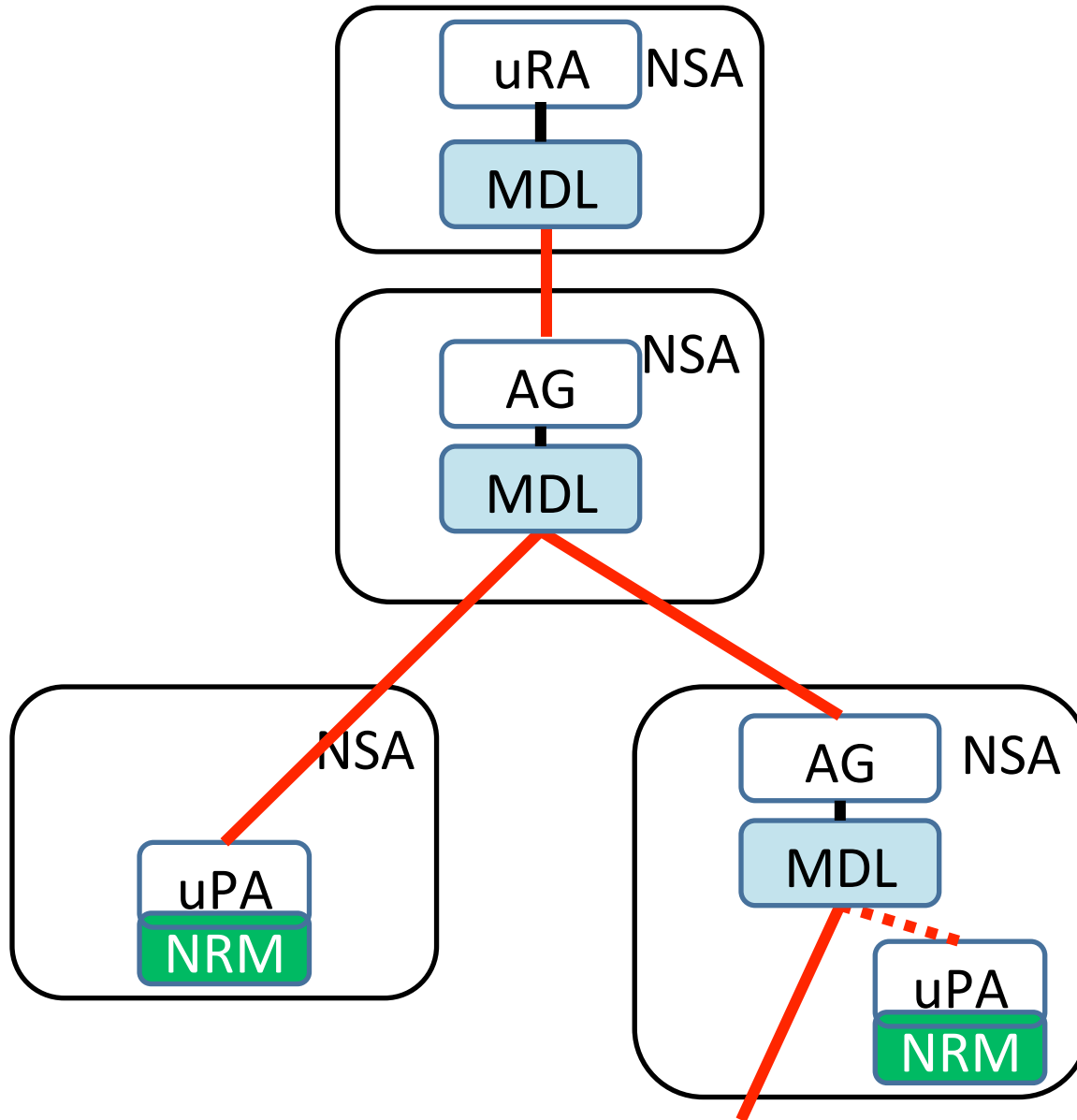
NSI State Machine – uRA/Aggregator - (Oxford v8)

NSI State Machine – uPA - (Oxford v8)

NSI State Machine – uRA/Aggregator - (Oxford v9_AutoStart)

This transition from "Reserved" directly to "Provisioned" is to implicit AutoStart feature
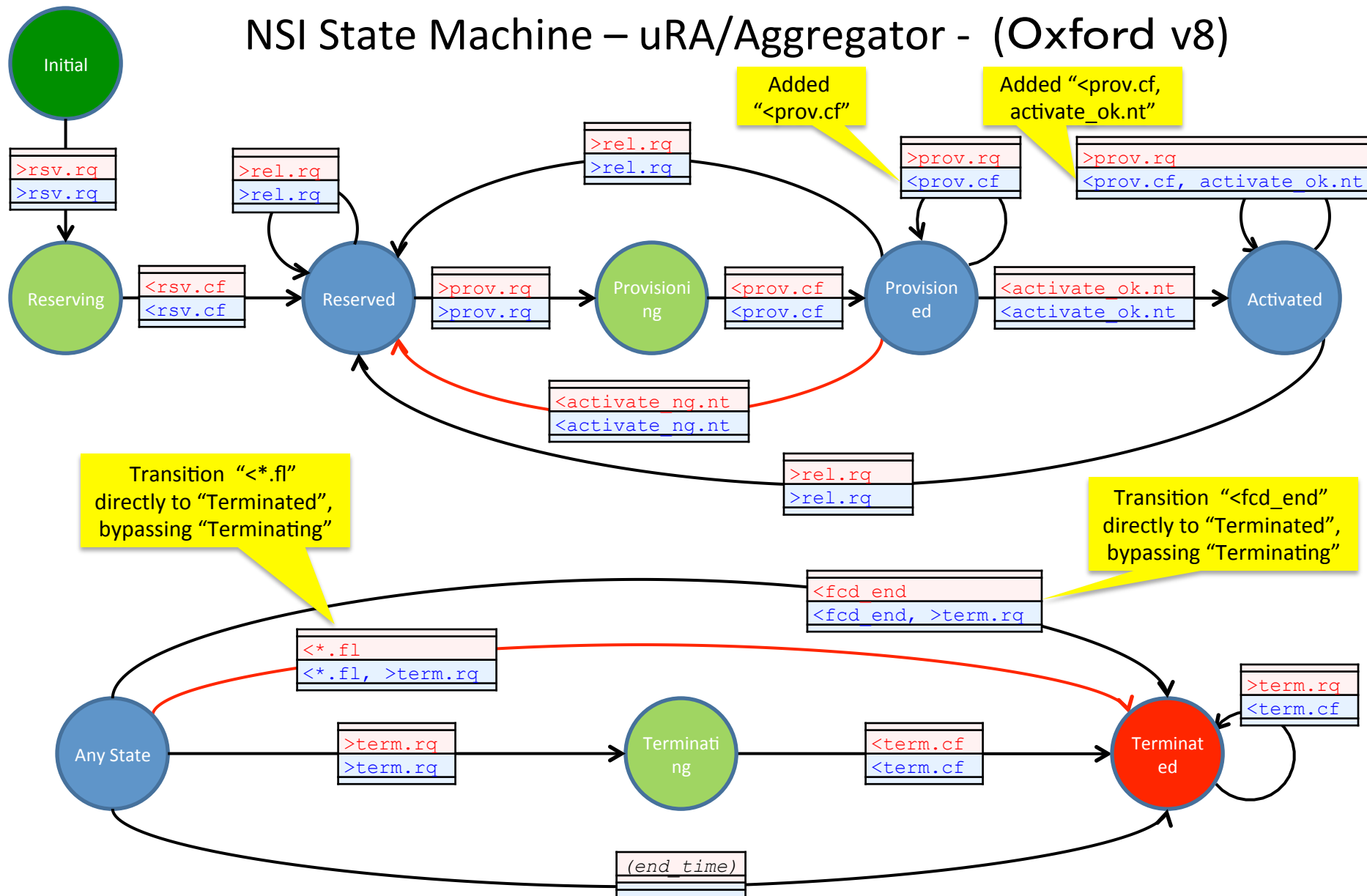
Transition to "Provisioned" only happens when all data planes are setup

Initial

>rsv.rq
>rsv.rq

Reserving

<rsv.cf
<rsv.cf

>prov.rq
>prov.rq

Reserved

<prov.cf
<prov.cf

Provisioning

<provision_ng.nt
<provision_ng.nt

<provision_ok.nt
<provision_ok.nt

<provision_ok.nt
<provision_ok.nt

Provisioning

>prov.rq
<prov.cf

Provisioned

>rel.rq
>rel.rq

<rel.cf
<rel.cf

<fcd_end
<fcd_end, >term.rq

<*.fl
<*.fl, >term.rq

Any State

>term.rq
>term.rq

Terminating

<term.cf
<term.cf

Terminated

>term.rq
<term.cf

(end_time)

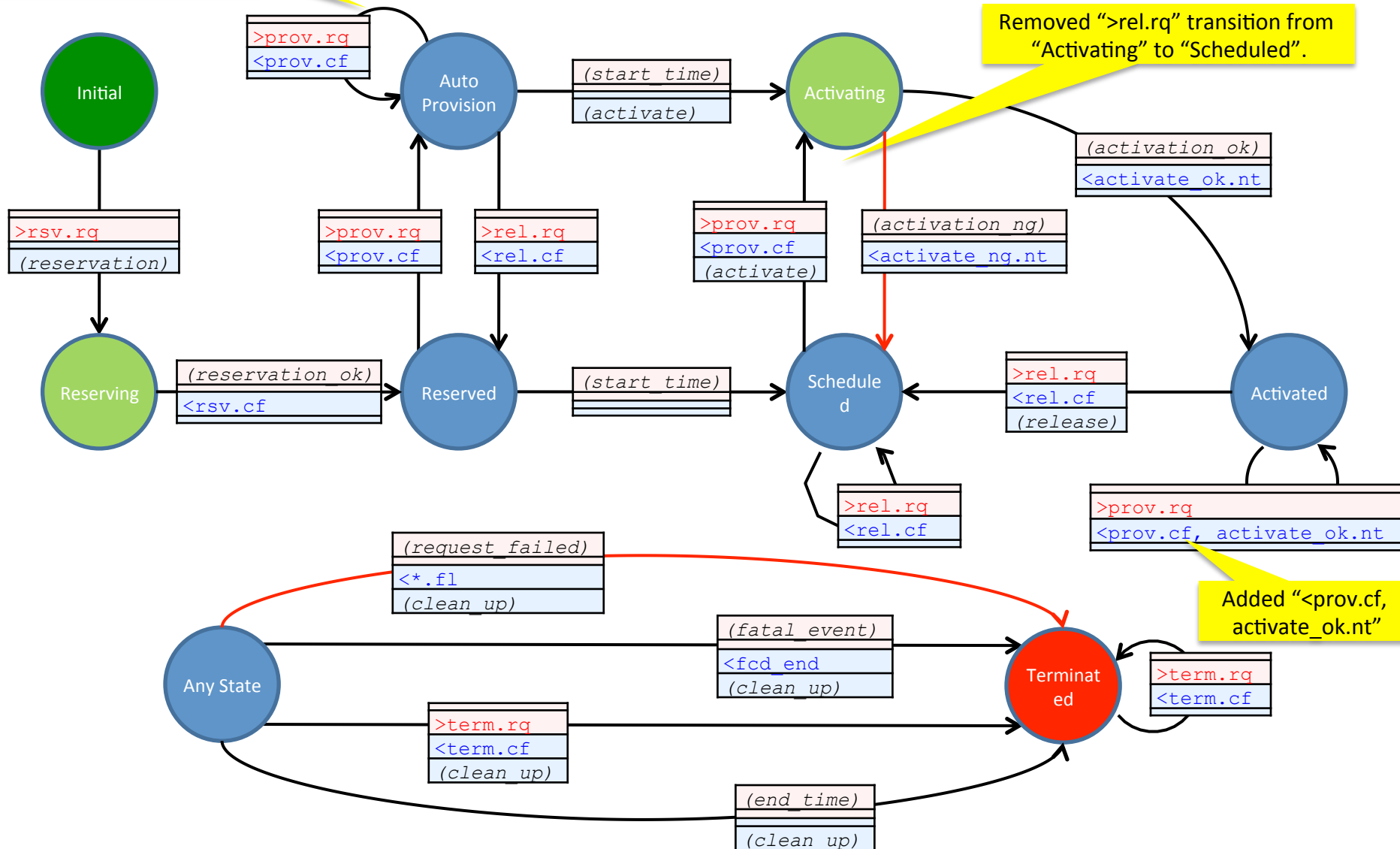Initial State   Transitional States   Stable States   Final State

NB: Messages received in this state is queued and processed only when it transitions to a Stable State.
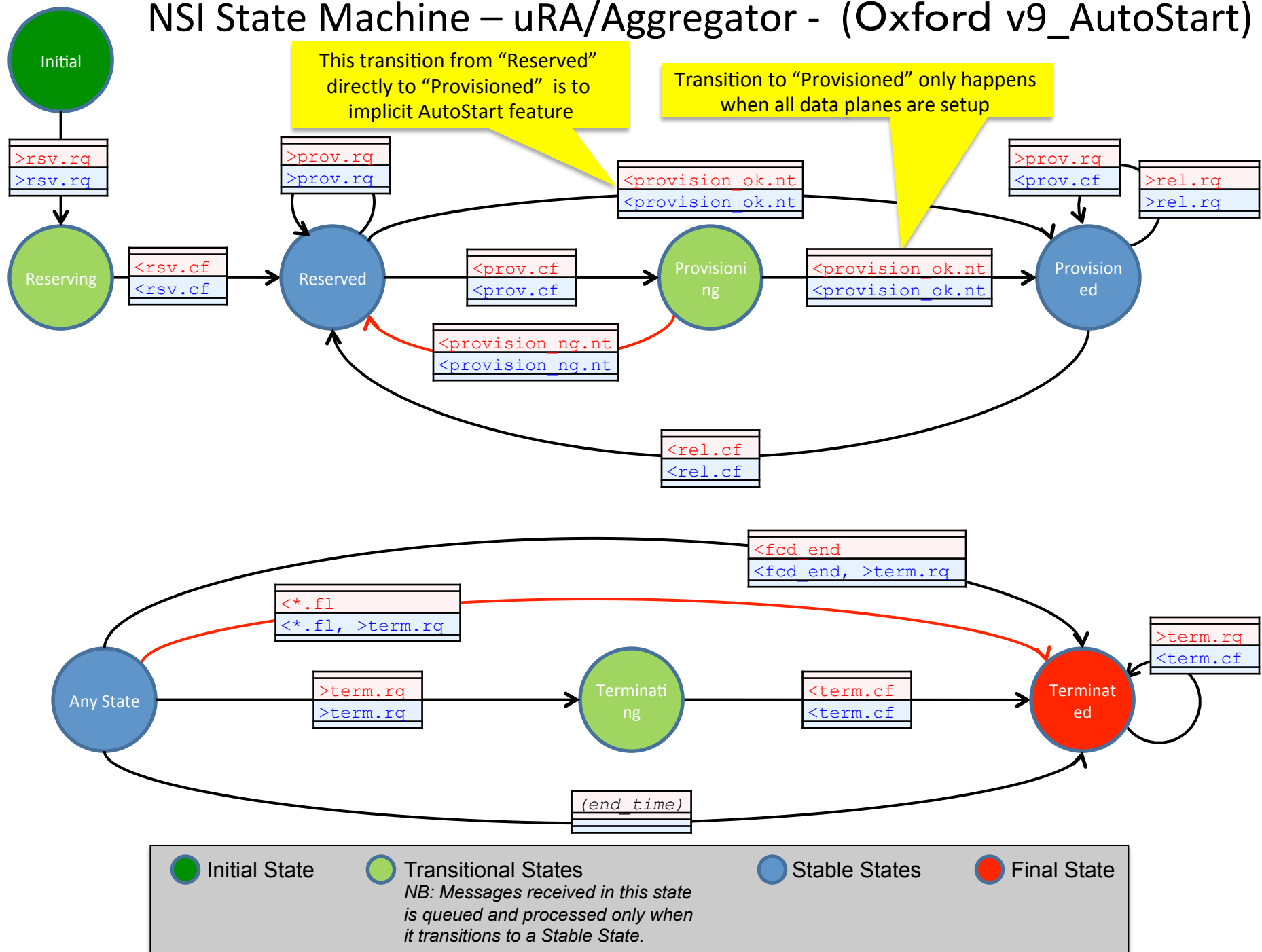
NSI State Machine – uPA - (Oxford v9_AutoStart)
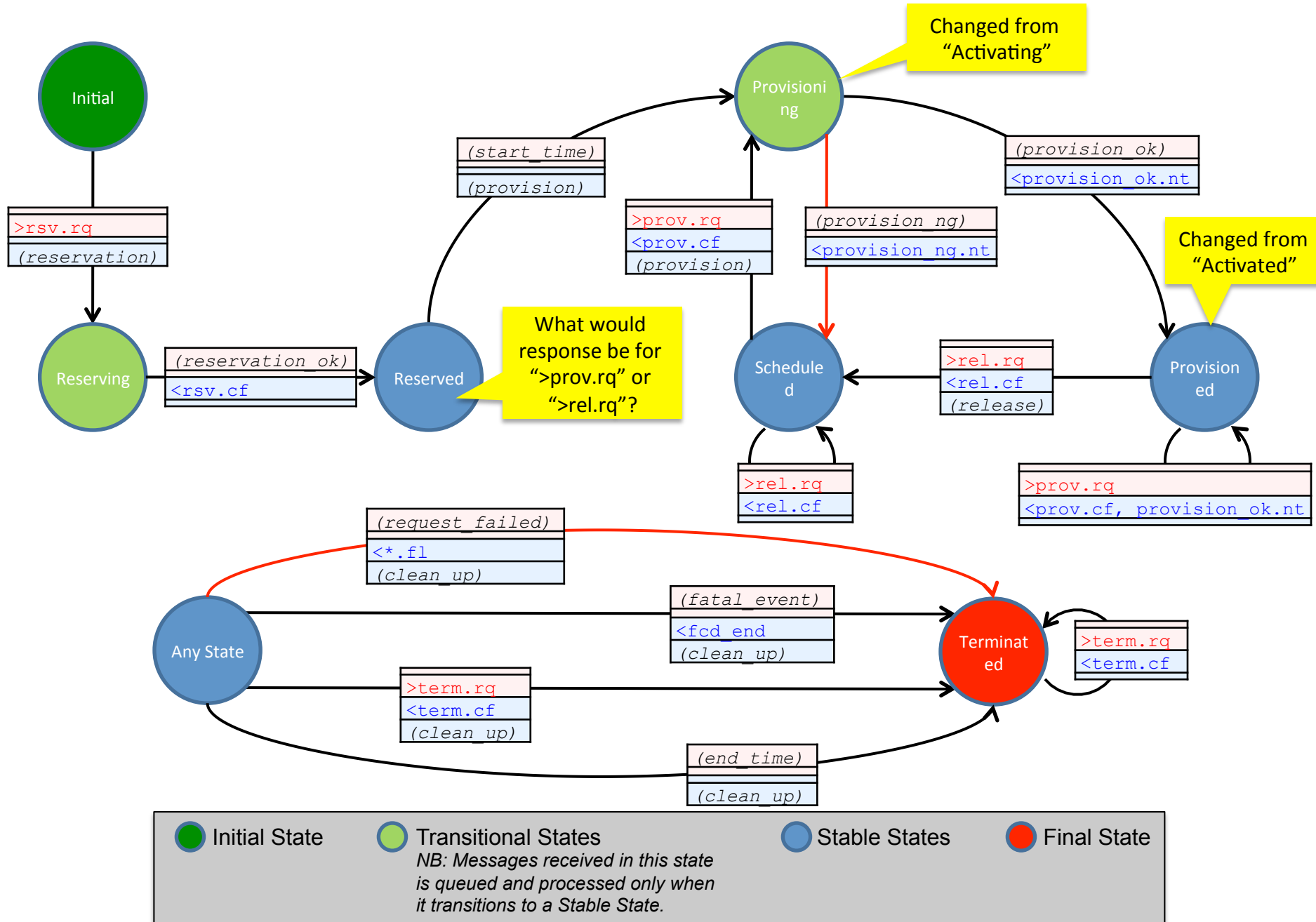
NSI State Machine – uRA/Aggregator - (Oxford v11)
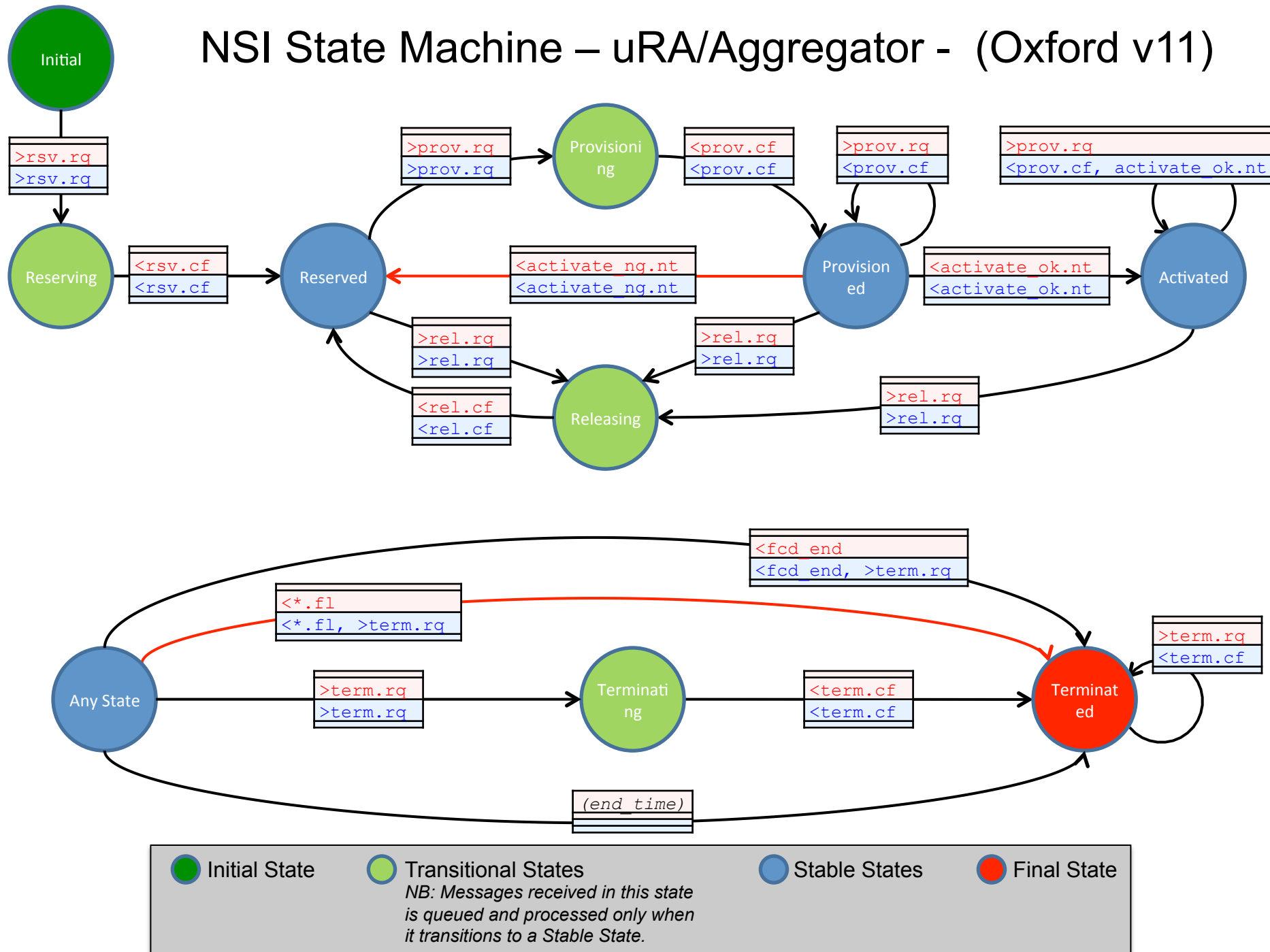
NSI State Machine – uPA - (Oxford v11)

Initial

>rsv.rq
(reservation)

Reserving

(reservation_ok)
<rsv.cf

Reserved

>prov.rq
<prov.cf

Auto Provision

>prov.rq
<prov.cf

(start_time)
(activate)

Activating

(activation_ok)
<activate_ok.nt

Activated

>prov.rq
<prov.cf, activate_ok.nt

>prov.rq
<prov.cf
(activate)

(activation_ng)
<activate_ng.nt

>prov.rq
<prov.cf

>rel.rq
<rel.cf

(start_time)

Scheduled

(release_ok)
<rel.cf

Releasing

>rel.rq
(release)

>rel.rq
<rel.cf

(request_failed)
<*.fl
(clean_up)

Any State

(fatal_event)
<fcd_end
(clean_up)

>term.rq
<term.cf
(clean_up)

(end_time)
(clean_up)

Terminated

>term.rq
<term.cf

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

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

NSI State Machine – uRA/Aggregator - (Delft v2)
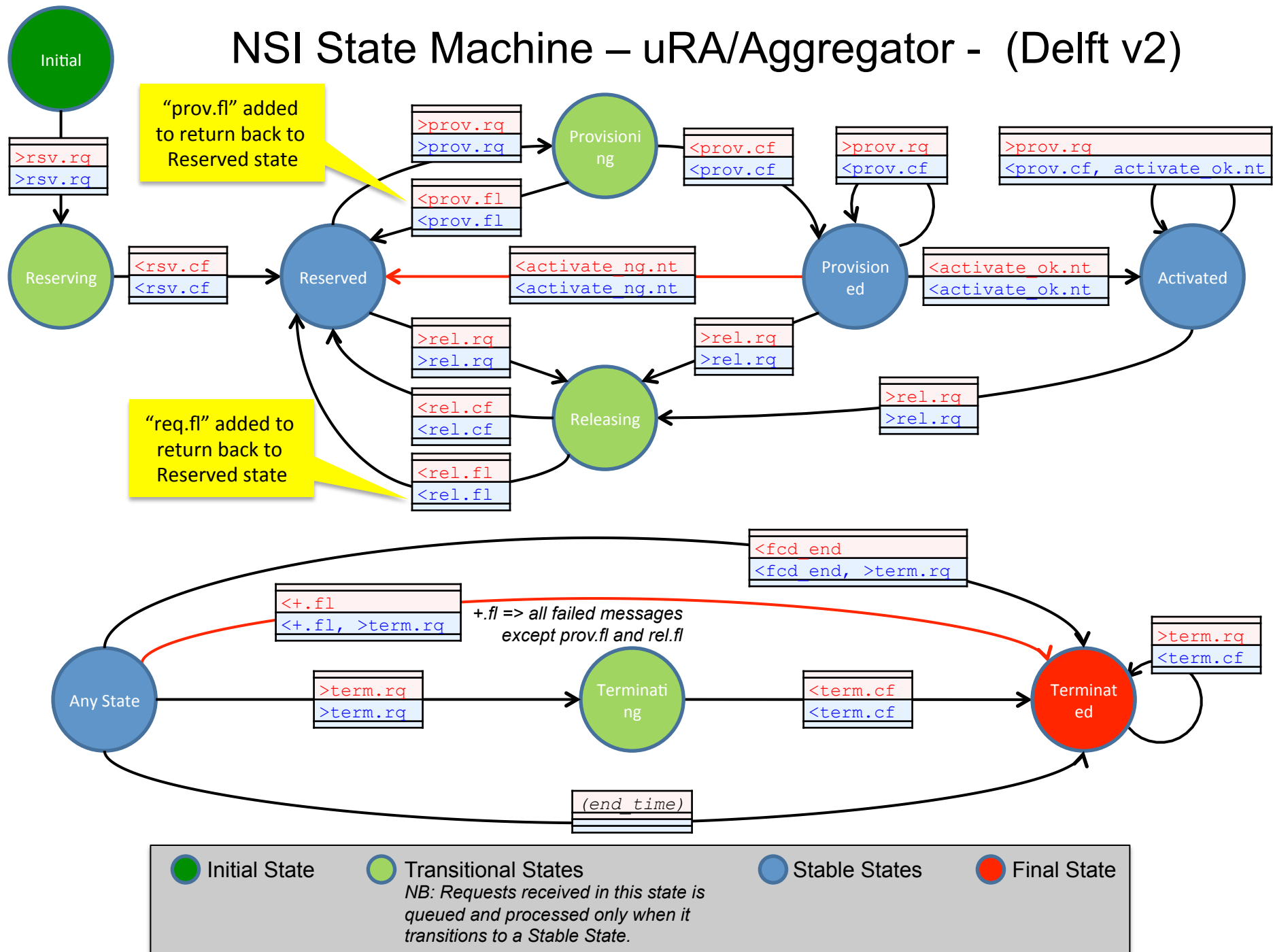
NSI State Machine – uPA – (Delft v2)

Initial → (rsv.rq / (reservation)) → **Reserving**

>prov.rq / <prov.cf — **Auto Provision**

>prov.rq / <prov.cf, <activate_ok.nt — **Activated**

(start_time) / (activate) → **Activating**

(activation_ok) / <activate_ok.nt → **Activated**

(activation_ng) / <activate_ng.nt → **Scheduled**

>rsv.rq / (reservation) → **Reserving**

>prov.rq / <prov.cf — **Reserved**

>rel.rq / <rel.cf

(reservation_ok) / <rsv.cf → **Reserved**

(start_time) → **Scheduled**

>prov.rq / <prov.cf / (activate) → **Scheduled**

>rel.rq / (release) → **Releasing**

(release_ok) / <rel.cf → **Scheduled**

(release_ng) / <rel.fl → **Scheduled**

>rel.rq / <rel.cf → **Scheduled**

NRM release failed transitions to Scheduled. This MAY not reflect actual end-to-end dataplane.

(request_failed) / <+.fl / (clean_up)

+.fl => all failed messages except rel.fl

Any State

(fatal_event) / <fcd_end / (clean_up) → **Terminated**

>term.rq / <term.cf / (clean_up) → **Terminated**

(end_time) / (clean_up) → **Terminated**

>term.rq / <term.cf → **Terminated**

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.