

# Multiple E2T Instances Design

## E2Manager DB

### Models

#### NodebInfo (existing, protobuf)

We shall add a new property to the NodebInfo struct,  
*AssociatedE2TInstanceAddress string*

#### E2TInstance (new, json)

*Address string*  
*AssociatedRanList []string*  
*KeepAliveTimetamp int64*  
*State E2TInstanceState*

#### E2TInstanceState (new, enum)

`type E2TInstanceState string`  
`Active E2TInstanceState = "ACTIVE"`  
`ToBeDeleted E2TInstanceState = "TO_BE_DELETED"`

#### E2TInstanceResponse (new, json)

*e2tAddress string*  
*ranNames []string*

### E2M DB

New keys:

key	value	Description
E2TInstance:<Address>	<i>E2TInstance JSON</i>	E2T instance is saved here by address key, and contains it's associated RANs  Calculation of the chosen E2T instance will be the E2TInstance with the minimum RAN count
E2TAddresses	<i>[]string JSON</i>	List of E2T Addresses

### Rnib Reader

New methods:

*GetE2TInstance(address string) (\*E2TInstance, error)*  
*GetE2TInstances(addresses []string) ([]\*E2TInstances, error)*  
*GetE2TAddresses() ([]string, error)*

### Rnib Writer

New methods:

*SaveE2TInstance(e2tInstance \*E2TInstance) error*  
*RemoveE2TInstance(address string) error*  
*SaveE2TAddresses([]string) error*

### struct E2TInstancesManager

*method SelectE2TInstance() (e2tAddress string, error)*

**method GetE2TInstance(e2tAddress string) (\*E2TInstance, error)**

**method GetE2TInstances() ([]\*E2TInstance, error)**

**method GetE2TAddresses() ([]string, error)**

**method AddE2TInstance(e2tAddress string) error**

1. Create \*E2TInstance{e2tAddress, AssociatedRanList} and set E2TInstance:<Address>

**lock -**

2. Get E2TAddresses

3. Add e2tAddress to E2TAddresses key (add to list)

4. Save E2TAddresses

**unlock**

**method RemoveE2TInstance(e2tAddress string) error**

**lock -**

1. Delete E2TInstance key from DB

2. Get E2TAddresses

3. Delete e2tAddress from E2TAddresses key

4. Save E2TAddresses

**unlock**

**method AddRanToInstance(ranName, e2tAddress ) error**

**lock -**

1. Get E2TInstance key

2. Add ranName to its AssociatedRanList

3. Save E2TInstance key

**unlock**

**method SetE2tInstanceState(e2tAddress, currState, newState) error**

**lock -**

1. Get E2TInstance key

2. If E2TInstance.state = currState

2.1. Set state

2.2. Save E2TInstance key

**unlock**

**method RemoveRanFromInstance(ranName, e2tAddress ) error**

**lock -**

1. Get E2Instance key

2. Remove ranName from its AssociatedRanList

3. Save E2Instance key

**unlock**

**method ClearRanFromInstances() error**

**lock -**

1. Get E2T Addresses
2. Get E2T Instances (addresses)
  - 2.1. Foreach E2T Instance
    - 2.1.1. Clear E2TInstance *AssociatedRanList*
    - 2.1.2. Save E2TInstance key

**unlock**

*method ResetKeepAliveTimestamp(address, count) error*

**lock -**

1. Get E2TInstance key
2. If E2TInstance.state = ACTIVE
  - 2.1. Set KeepAliveTimestamp = now
  - 2.2. Save E2TInstance key

**unlock**

## **struct E2tAssociationManager**

*method AssociateRan(ranName, e2tAddress) error*

*method DissociateRan(ranName, e2tAddress) error*

*method RemoveE2tInstance(e2tAddress, ranNameListToBeDissociated []string, ranAssocList []\*E2tInstance) error*

## **struct E2TKeepAliveWorker**

*method Execute()*

## **struct E2TKeepAliveResponseHandler**

*method Handle()*

## **struct E2TShutdownManager**

*method Shutdown(fqdn)*

## **struct e2tController**

*method getAllE2tInstances()*

## **Configuration**

Add keep-alive timestamp interval to helm-chart

## **E2M Init process**

Set all ACTIVE E2tInstances' timestamp to now()

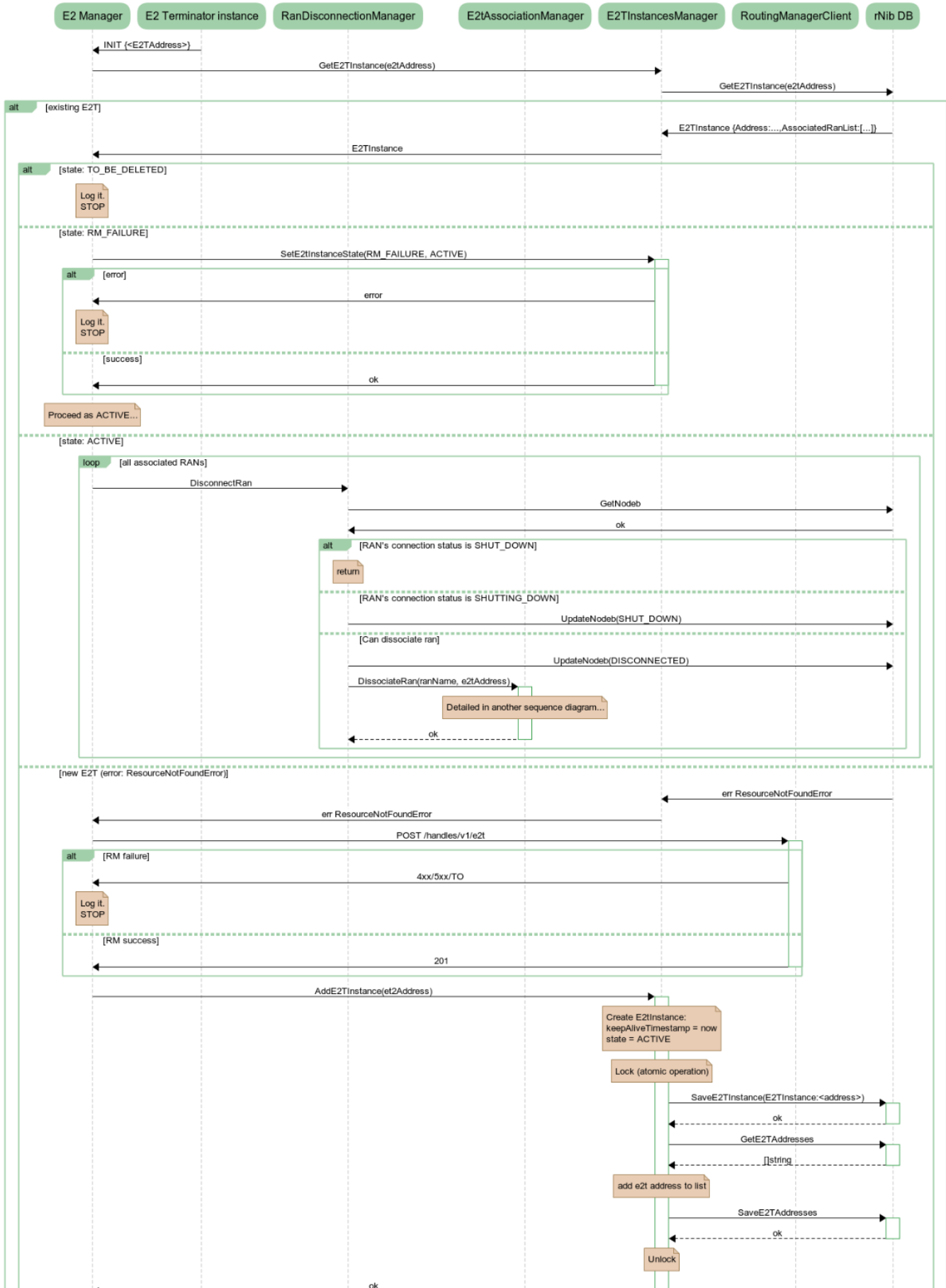
## **E2T Init Flow**

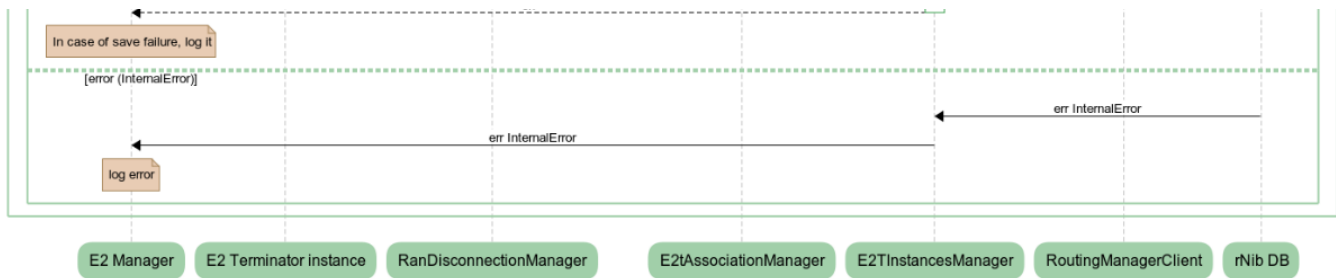
RMR INIT message

E2T sends an INIT message to E2M.  
In the current implementation, message is EMPTY.  
Now, it should contain the instance's **Address**.

Sequence Diagram

# E2T Init flow



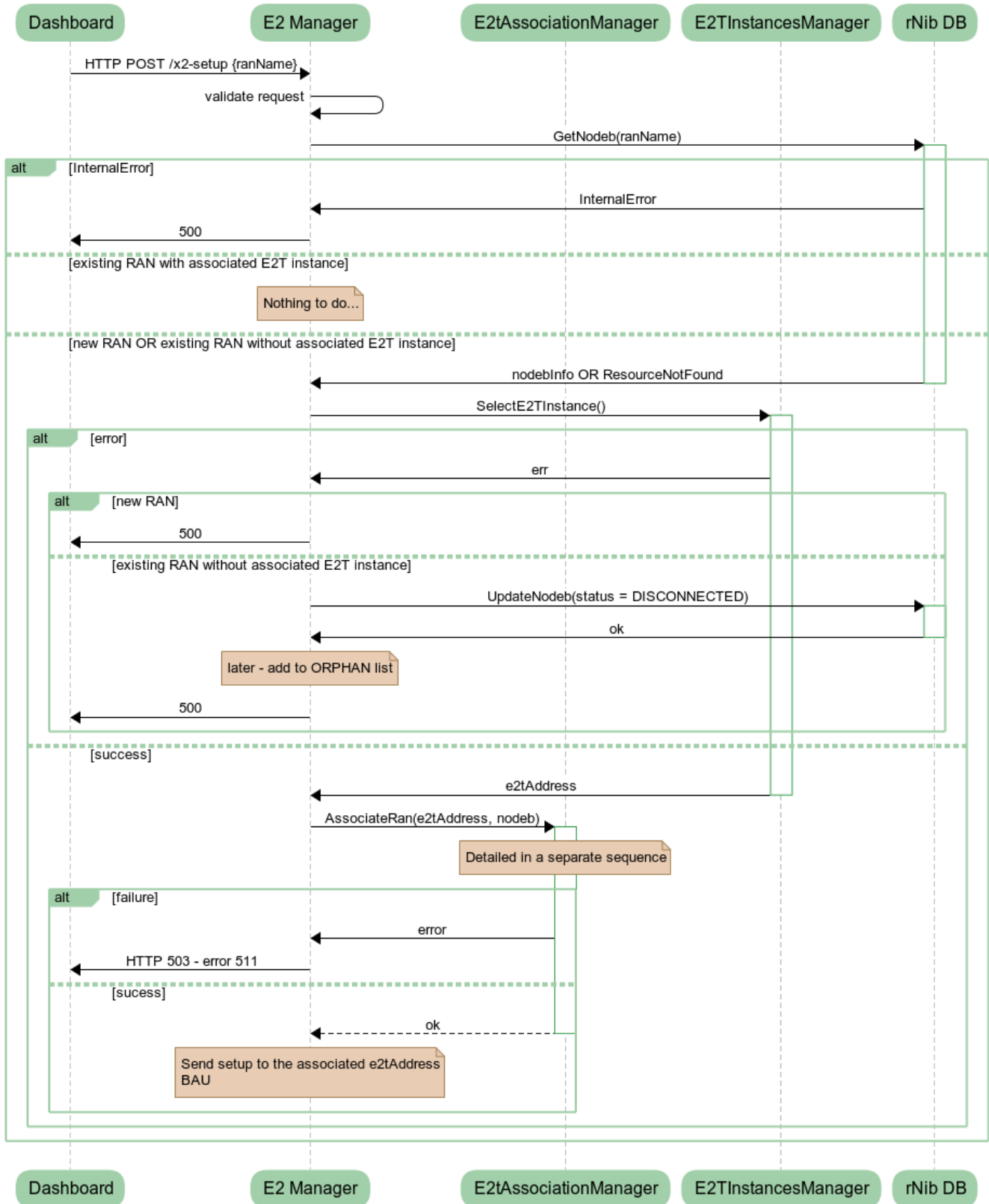


E2T Init flow.txt

## Setup Flow

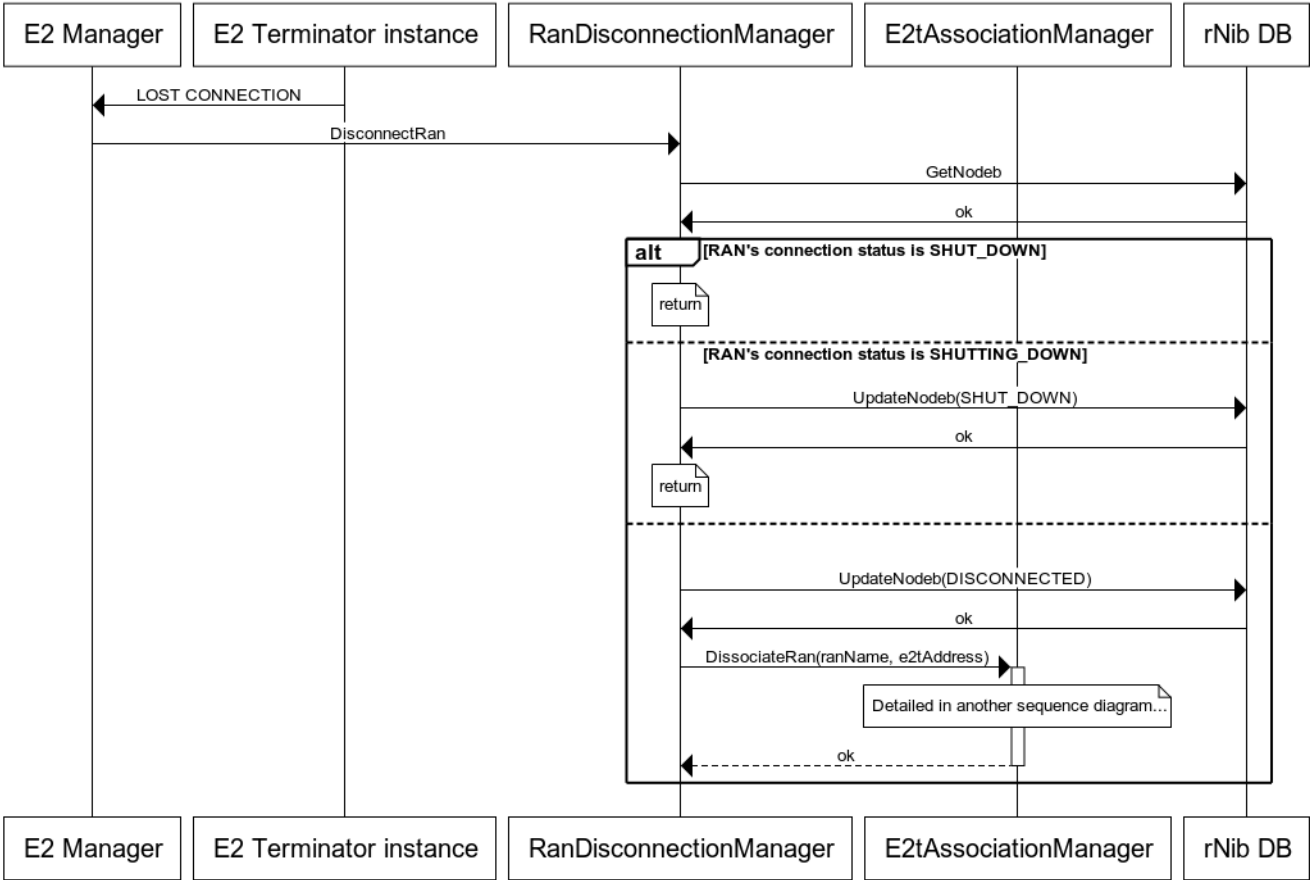
Sequence Diagram

## Setup Flow (from Dashboard)



## Lost Connection Flow

Lost Connection flow

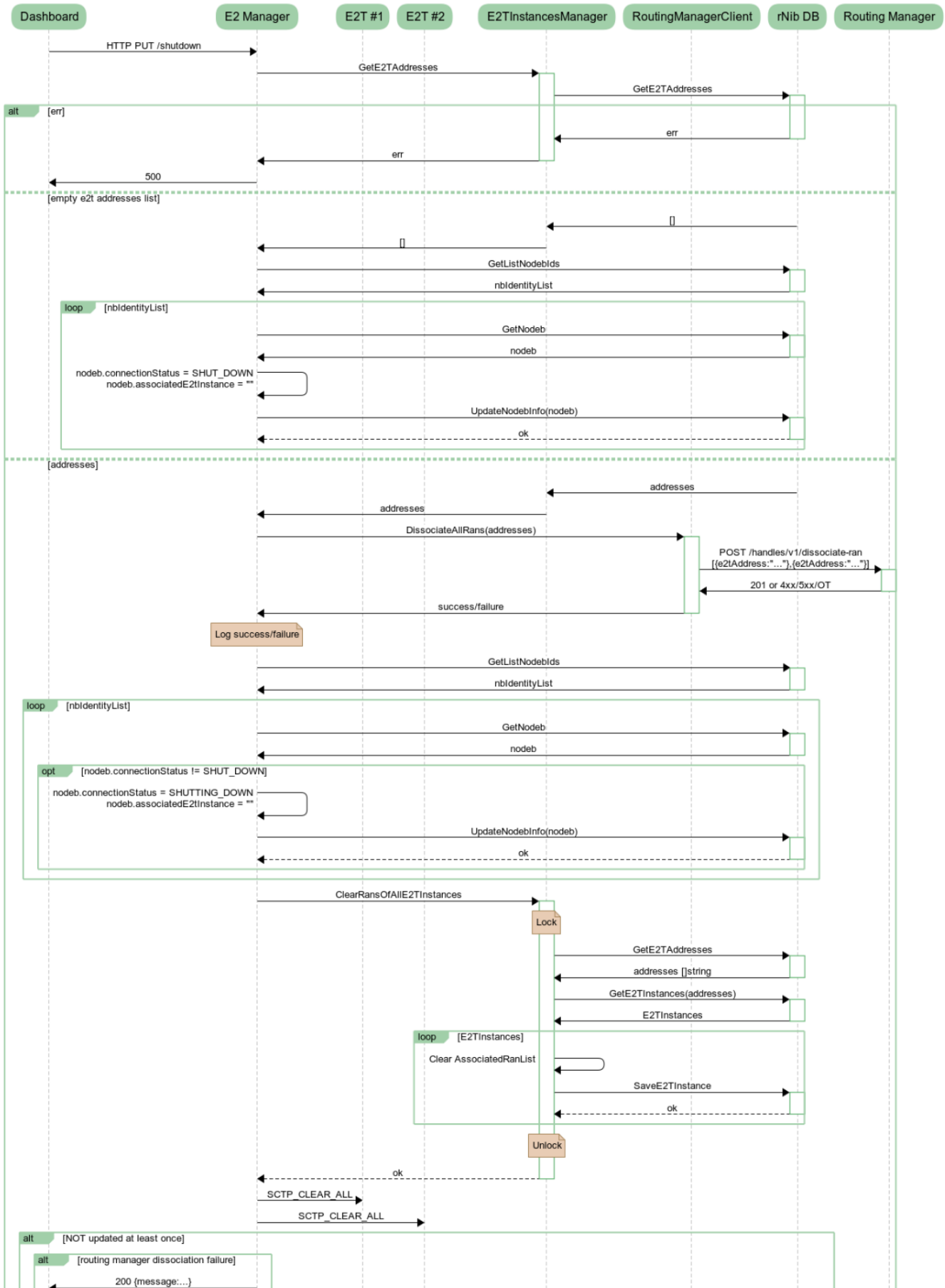


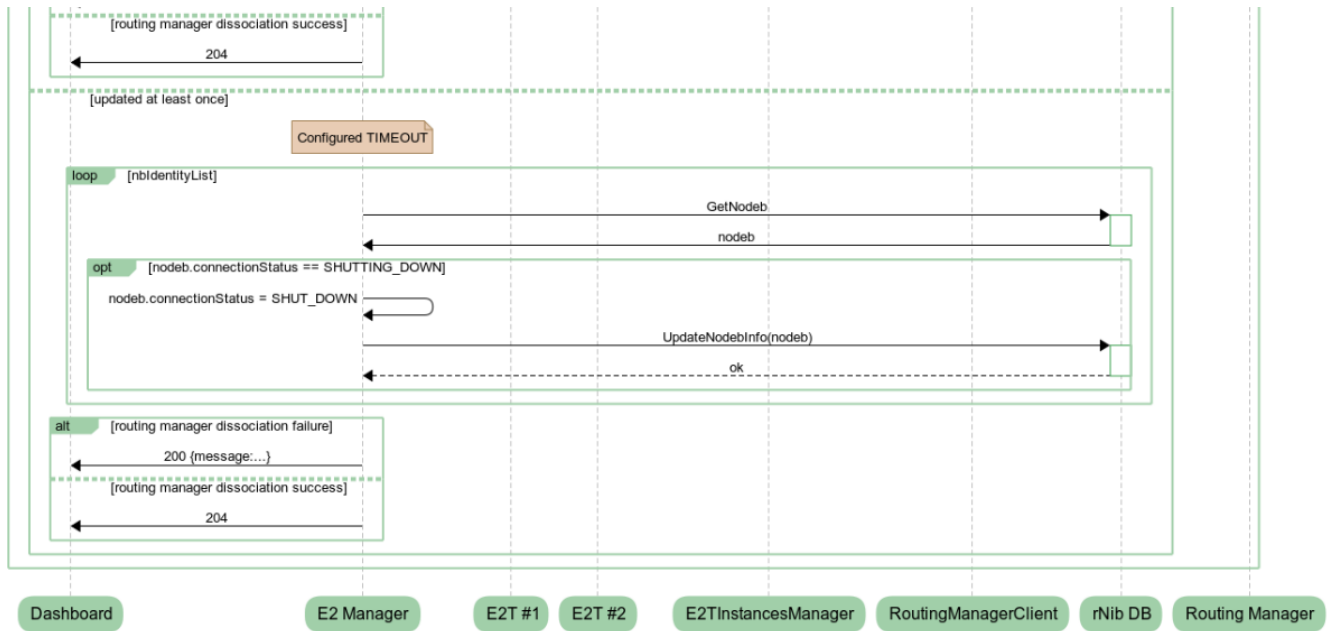
Lost Connection flow.txt

Red Button Flow



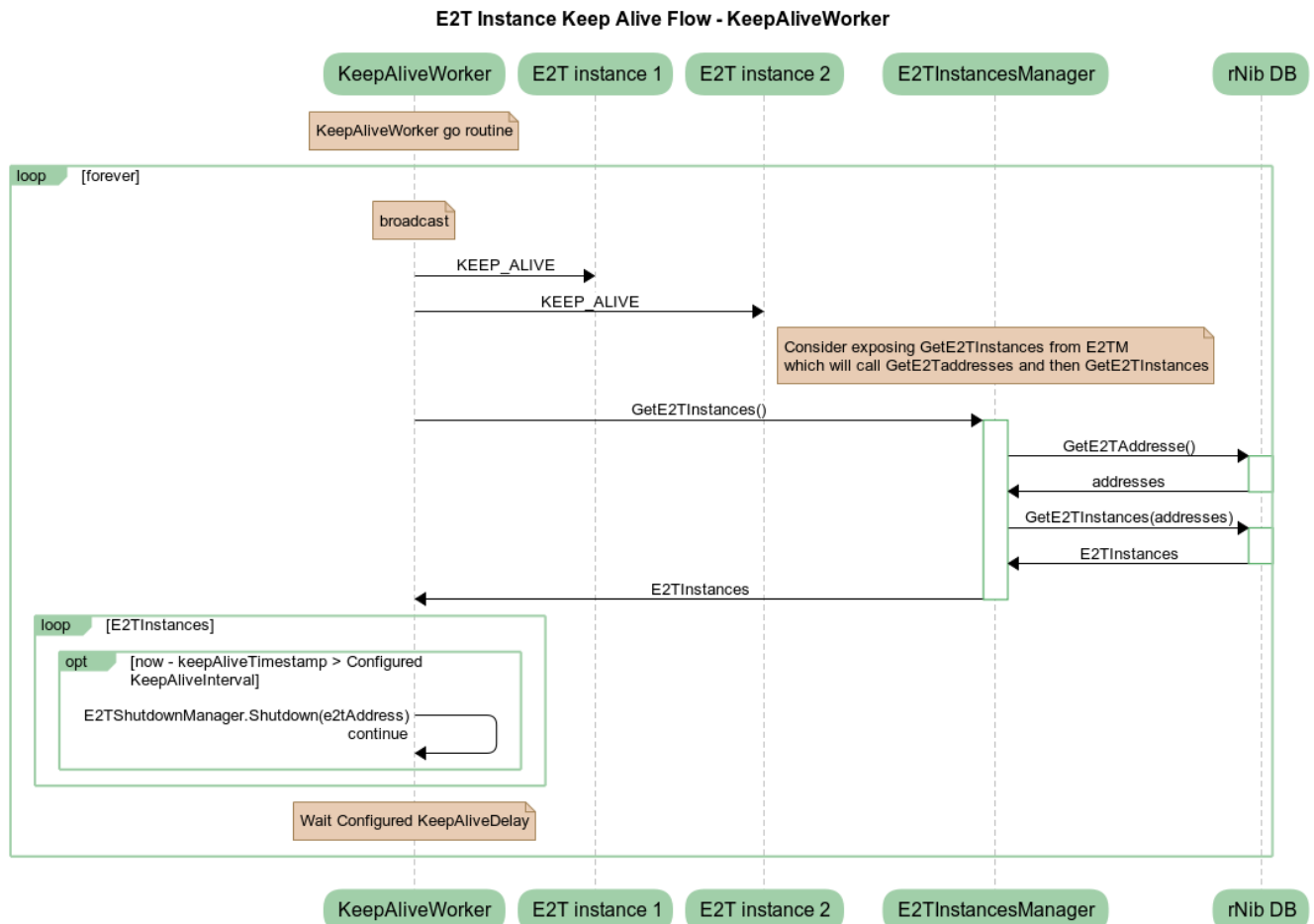
# Red Button flow





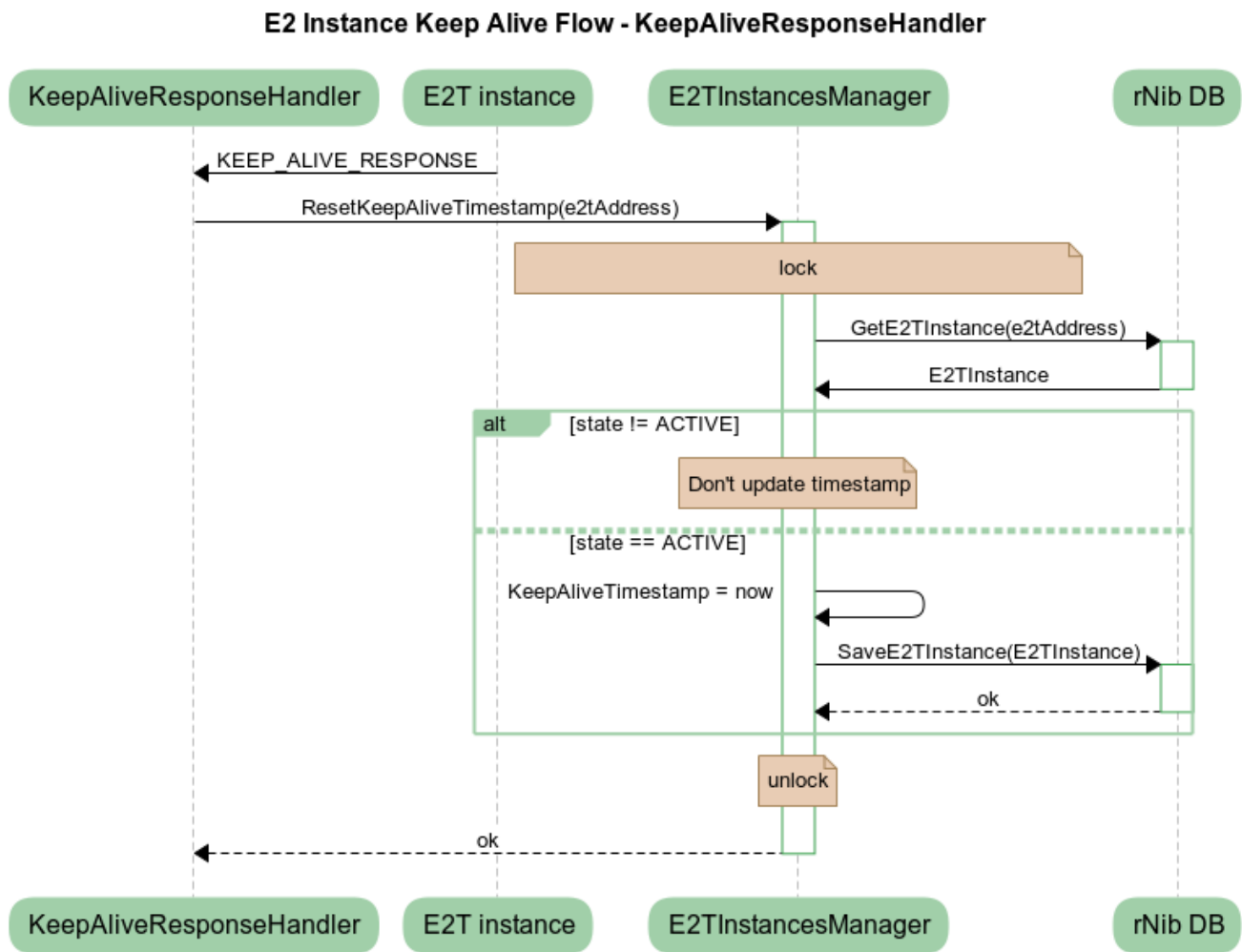
## E2T Instance Keep Alive Flow - KeepAliveWorker

Sequence Diagram



E2T Instance Keep Alive Flow - KeepAliveResponseHandler

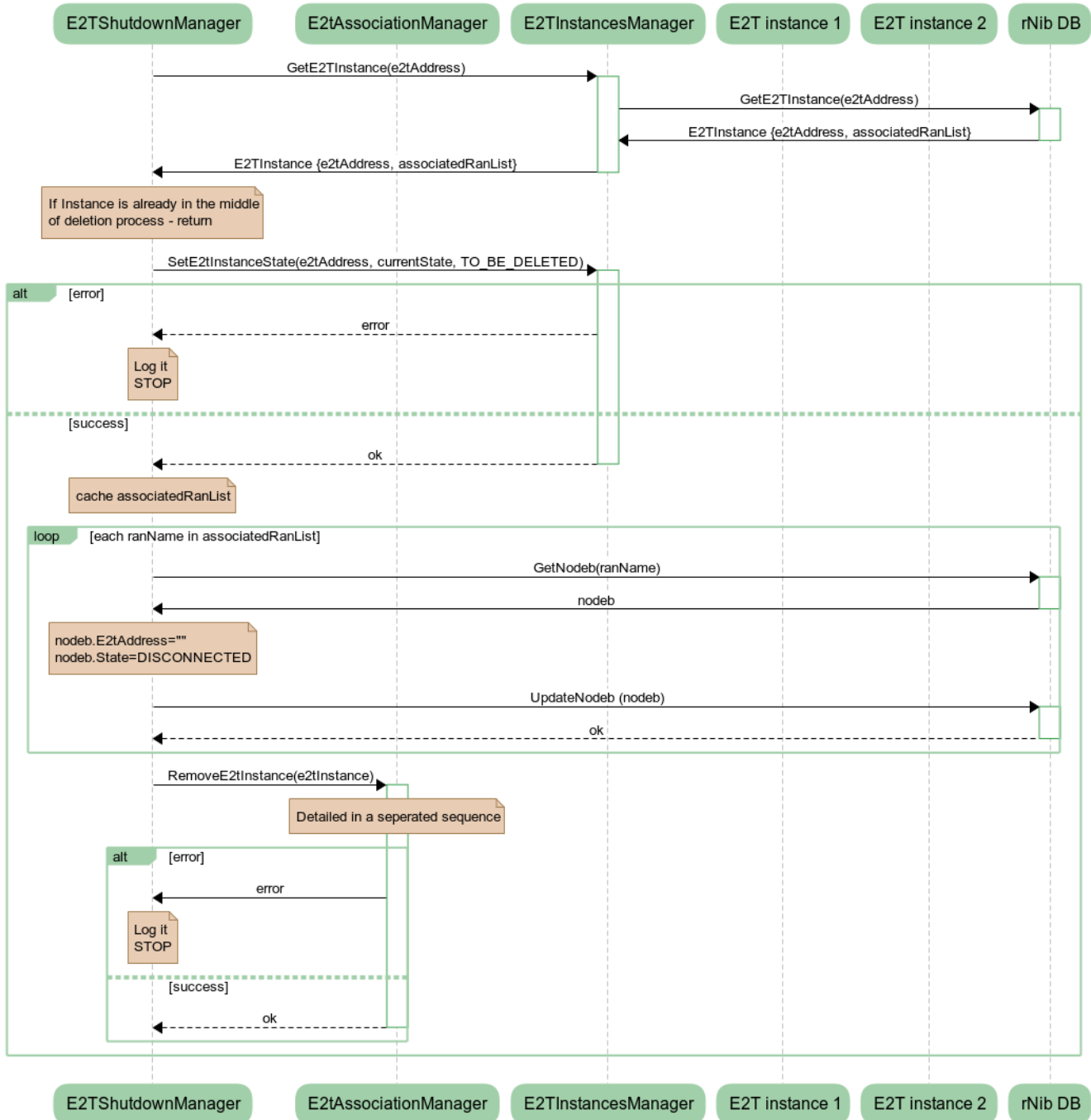
Sequence Diagram



E2T Instance Keep Alive Flow - E2TShutdownManager

Sequence Diagram

## E2 Instance Keep Alive Flow - E2TShutdownManager

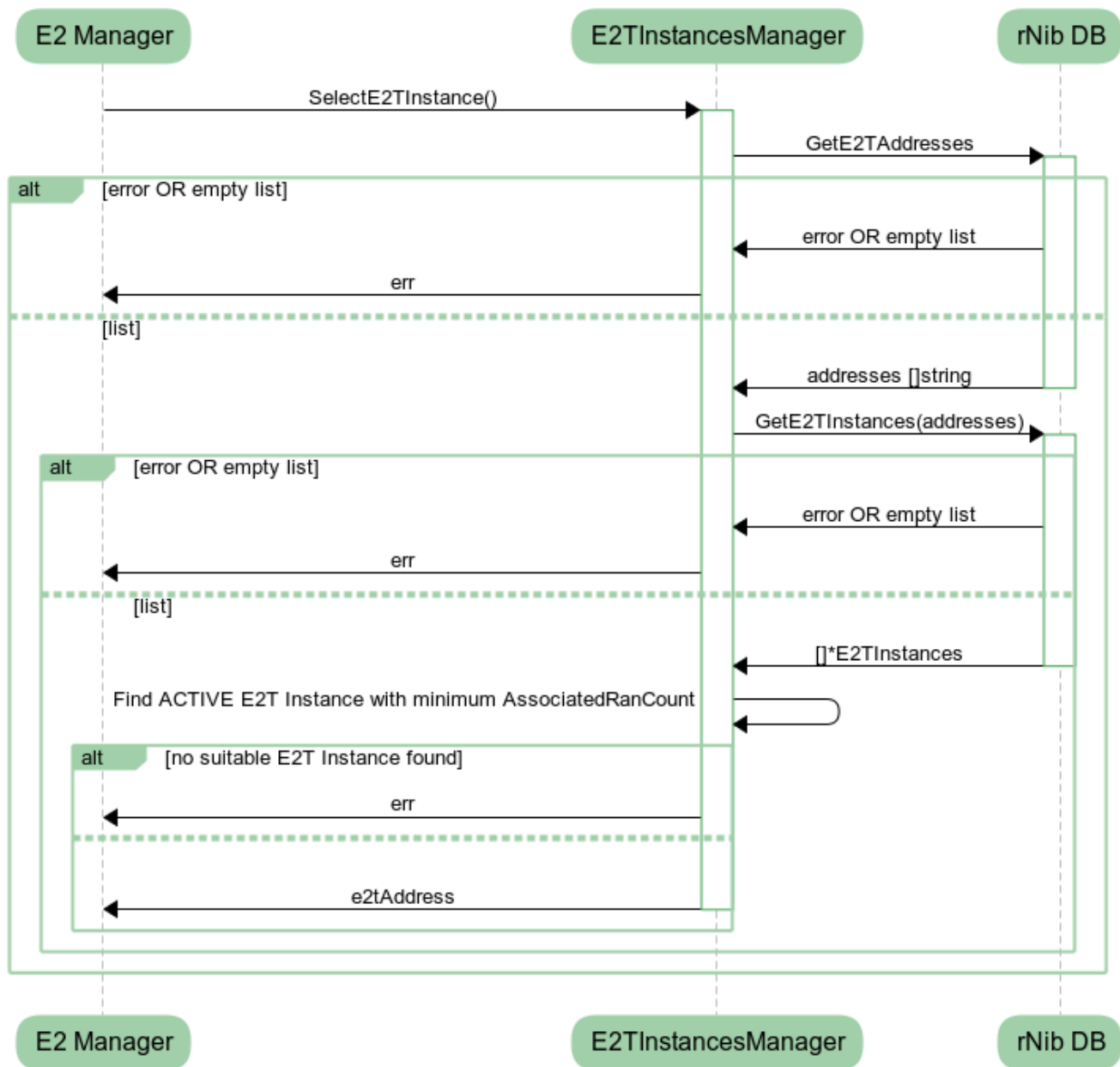




E2 Instance Keep...nManager (6).txt

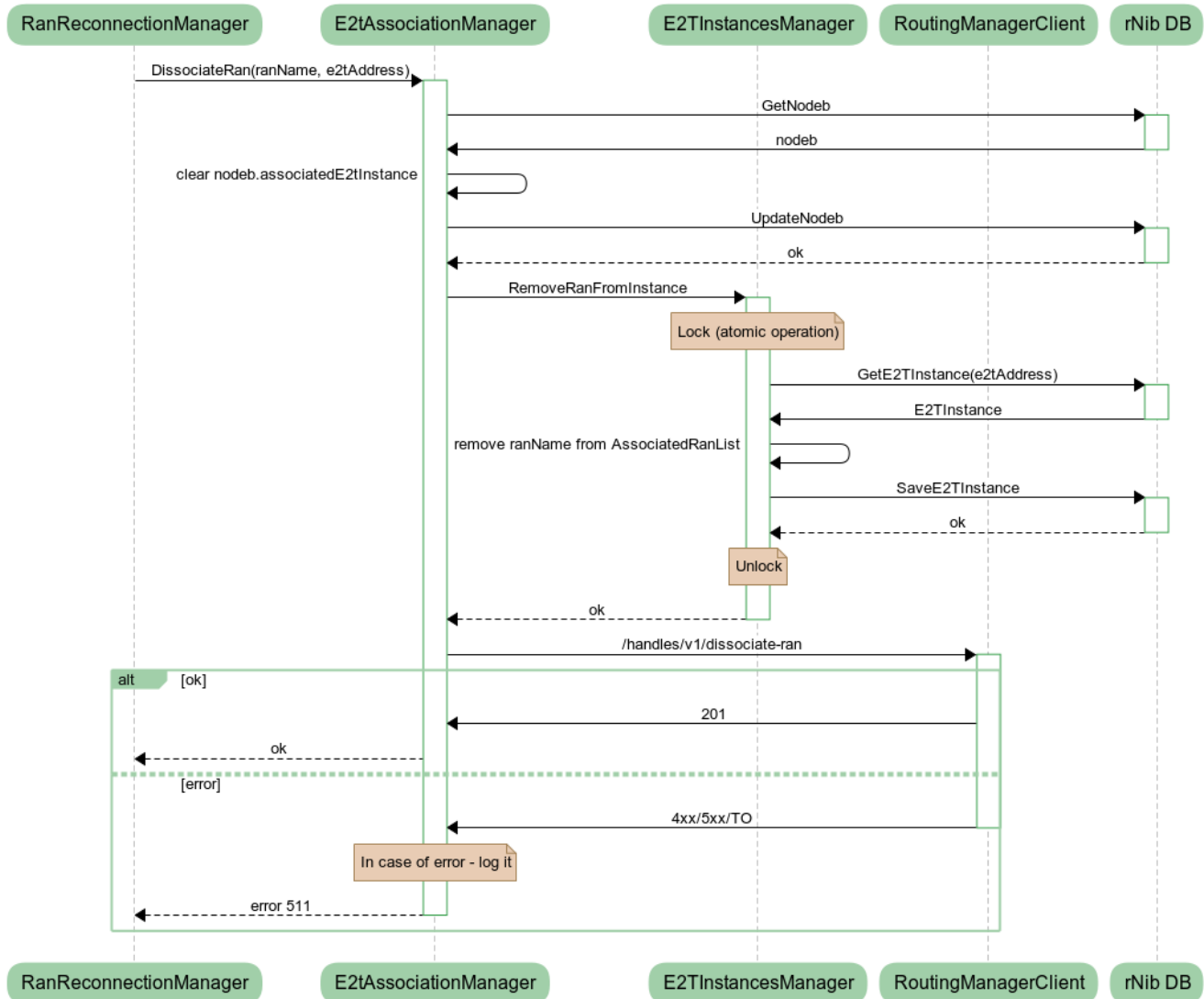
Selection of E2T Instance

## Selection of E2tInstance



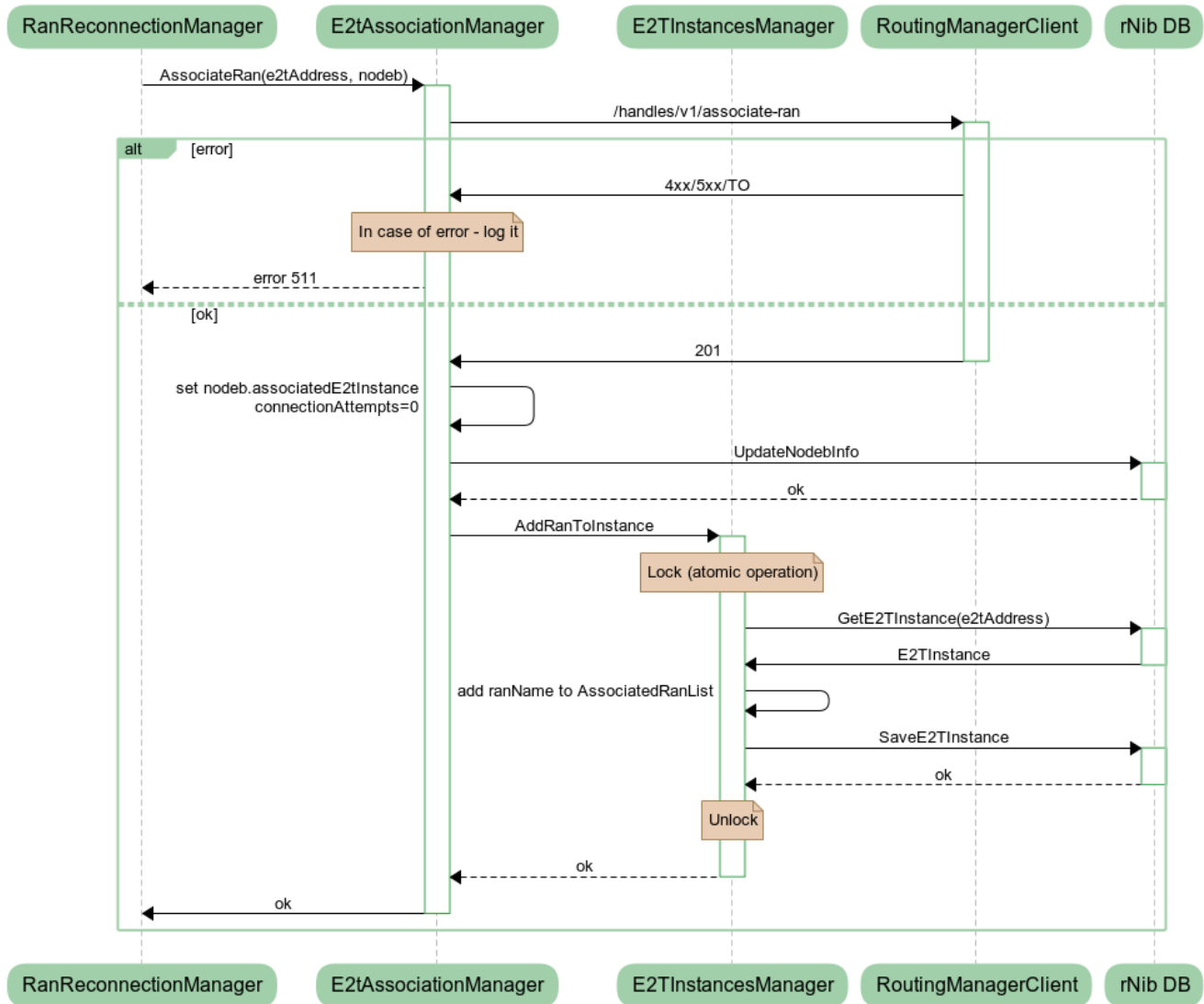
Dissociate RAN

## E2tInstanceManager DissociateRan



Associate RAN

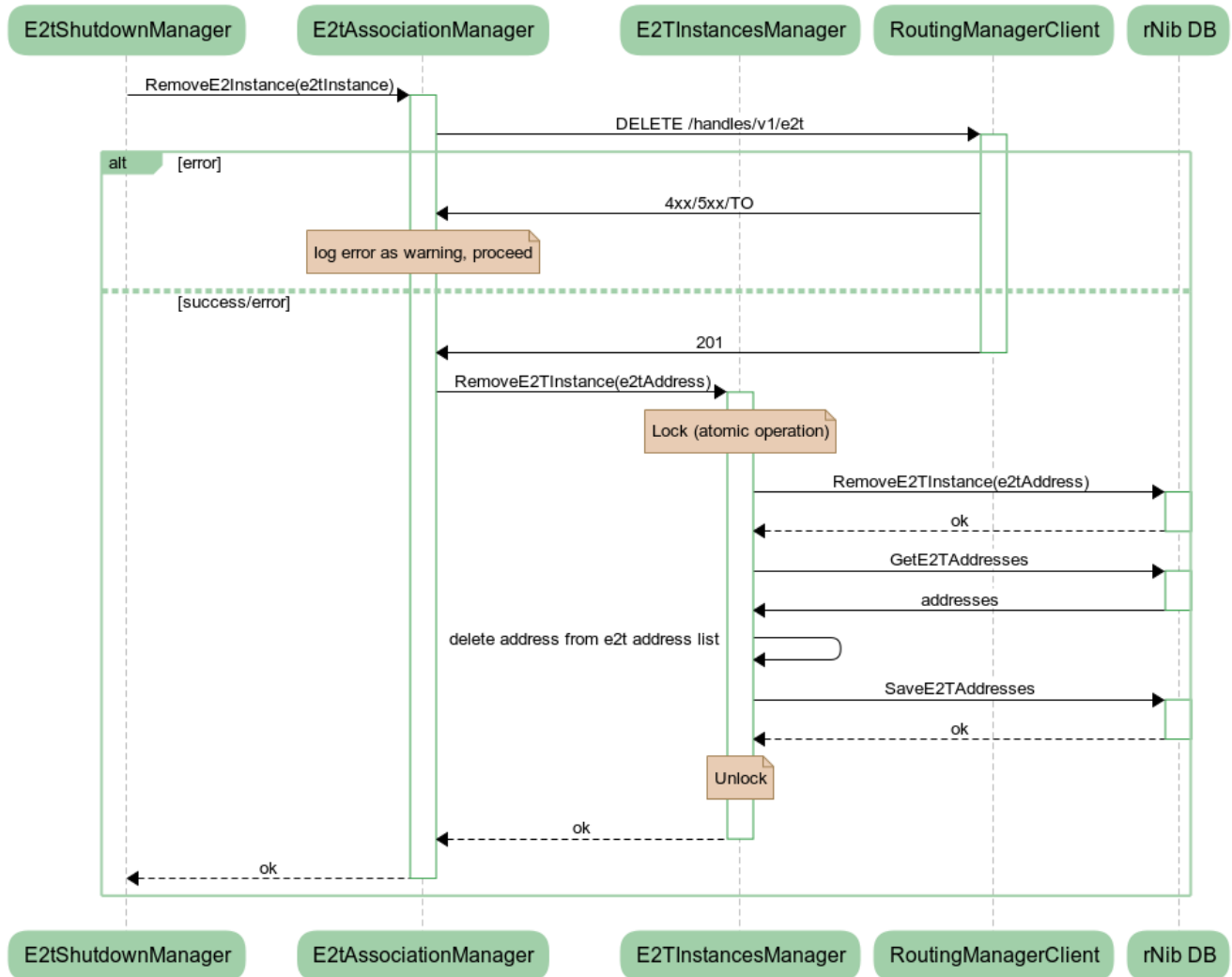
## E2tInstanceManager AssociateRan



Remove E2T Instance

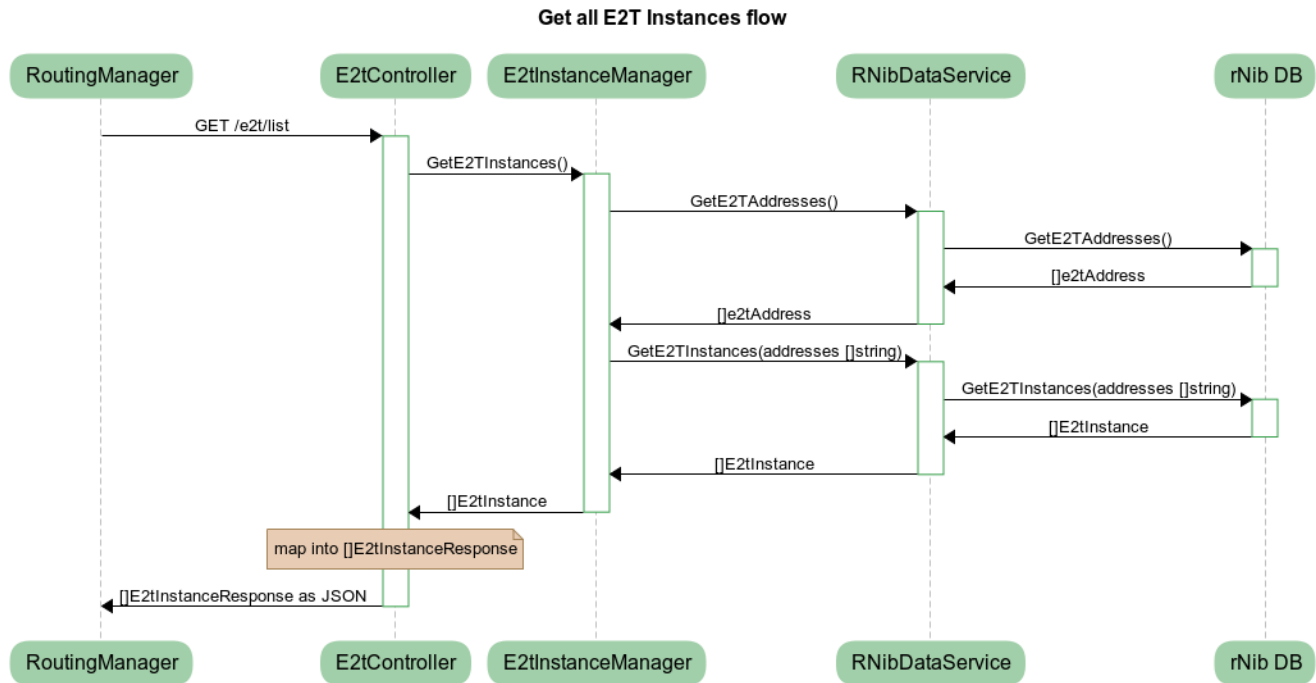


### Remove E2T Instance flow



Remove E2T Instance (3).txt

## Get all E2T Instances



## Set E2T Instance state

## Set E2T Instance state

