

## Anaconda network configuration:

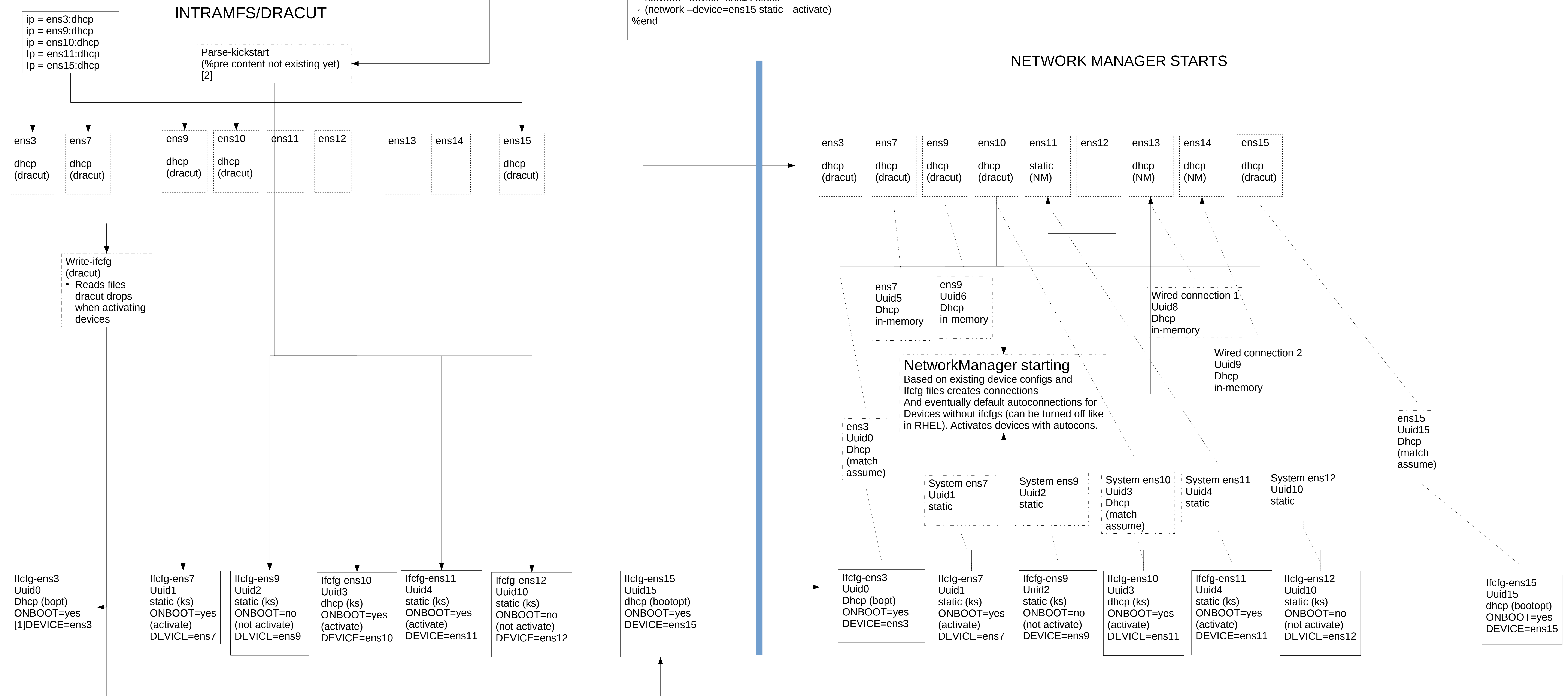
Stages:

- 1) initramfs/dracut: boot options, kickstart → ifcfgs moved to Anaconda root
- 2) Anaconda initialize: settle started NM
- 3) Anaconda initialize : kickstart %pre, activating kickstart config
- 4) Anaconda GUI/TUI
- 5) Anaconda post install tweaks and config – ifcfg: ONBOOT, dracut boot options for network storage

- Configuration in dracut: pxe (installer image), kickstart, updates image, DUD? VNC?
- Devices supported in installer (ethernet, infiniband, wifi, bond, team, bridge, vlan)
- Persistent configuration (ifcfg) of installer environment and target system is the same.
  - It is possible to have another device configuration in installation environment but only via boot opts or default autoconnections.
- One persistent configuration (connection → ifcfg) per device. All the other connections should be only active connections of devices.
- All ifcfgs copied to target system (ifcfg files, dhclient files, static routes configuration)
- Usually working with NM connection (add/update/activate) but sometimes directly with ifcfg: ONBOOT, update/create data.network, create dracut boot opts
- Bootloader options for root on network storage (iscsi)
- Limited support in TUI, virtual devices: show, but can't configure
- GUI: nm-c-e <the con\_uid>, adding virtual device: nm-c-e team
- Initialization – sync with NM, waiting for connecting NM (dhcp) – not very robust but seems to work so far
- GUI can handle added devices (usb) to some extent (added\_device callback), we don't support it officially
- Special handling:
  - lbt
  - S390 – another config file (sysfs, ccw.conf created in s390's init?)
  - Wireless support – GUI
- Ks tests for most of the stuff
- Device specification (--device) and device binding (DEVICE/HWADDR/both)
- Device renaming ifname=eth0:XX:XX:XX:XX:XX

```
network --device ens7 --bootproto static --ip 192.168.124.200 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2 --activate --onboot=no
network --device 52:54:00:5a:29:50 --bootproto static --ip 192.168.124.201 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2
network --device ens10 --bootproto dhcp --activate
network --device ens11 --bootproto static --ip 192.168.124.202 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2 --activate --onboot=no
network --device ens12 --bootproto static --ip 192.168.124.203 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2
%include /tmp/ksinclude
%pre
echo "network --device ens14 --bootproto static --ip 192.168.124.204 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2" >> /tmp/ksinclude
echo "network --device ens15 --bootproto static --ip 192.168.124.205 --netmask 255.255.255.0 --gateway 192.168.124.255 --nameserver 10.34.39.2" >> /tmp/ksinclude
%end
```

```
network --device=ens7 static --activate --onboot=no
network --device=11:11:11:11:11:11 static
network --device=ens10 dhcp --activate
network --device=ens11 static --activate --onboot=no
network --device=ens12 static
%pre
-- network --device=ens14 static
-- (network --device=ens15 static --activate)
%end
```

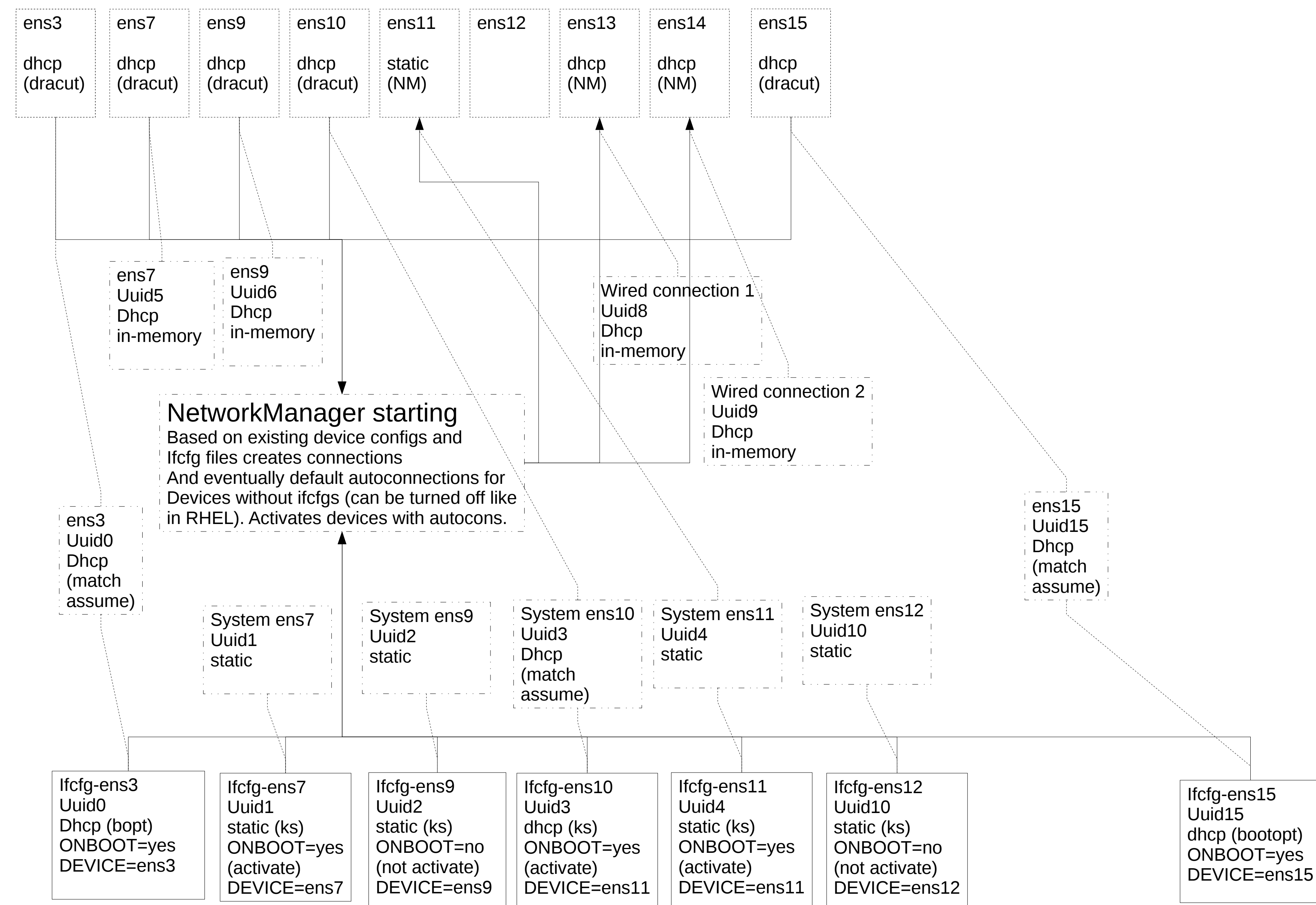


/run/initramfs/state/etc/sysconfig/network-scripts/

[1] For --bindto=mac there is HWADDR=XX:XX:XX:XX:XX:XX  
 For ifname=eth0:XX:XX:XX:XX:XX there is both DEVICE and HWADDR  
 [2] sometimes it is more complicated, parse-kickstart creates ip= boot options from kickstart for dracut and those are applied already in initramfs. I think this works only if dracut is not fetched over network.

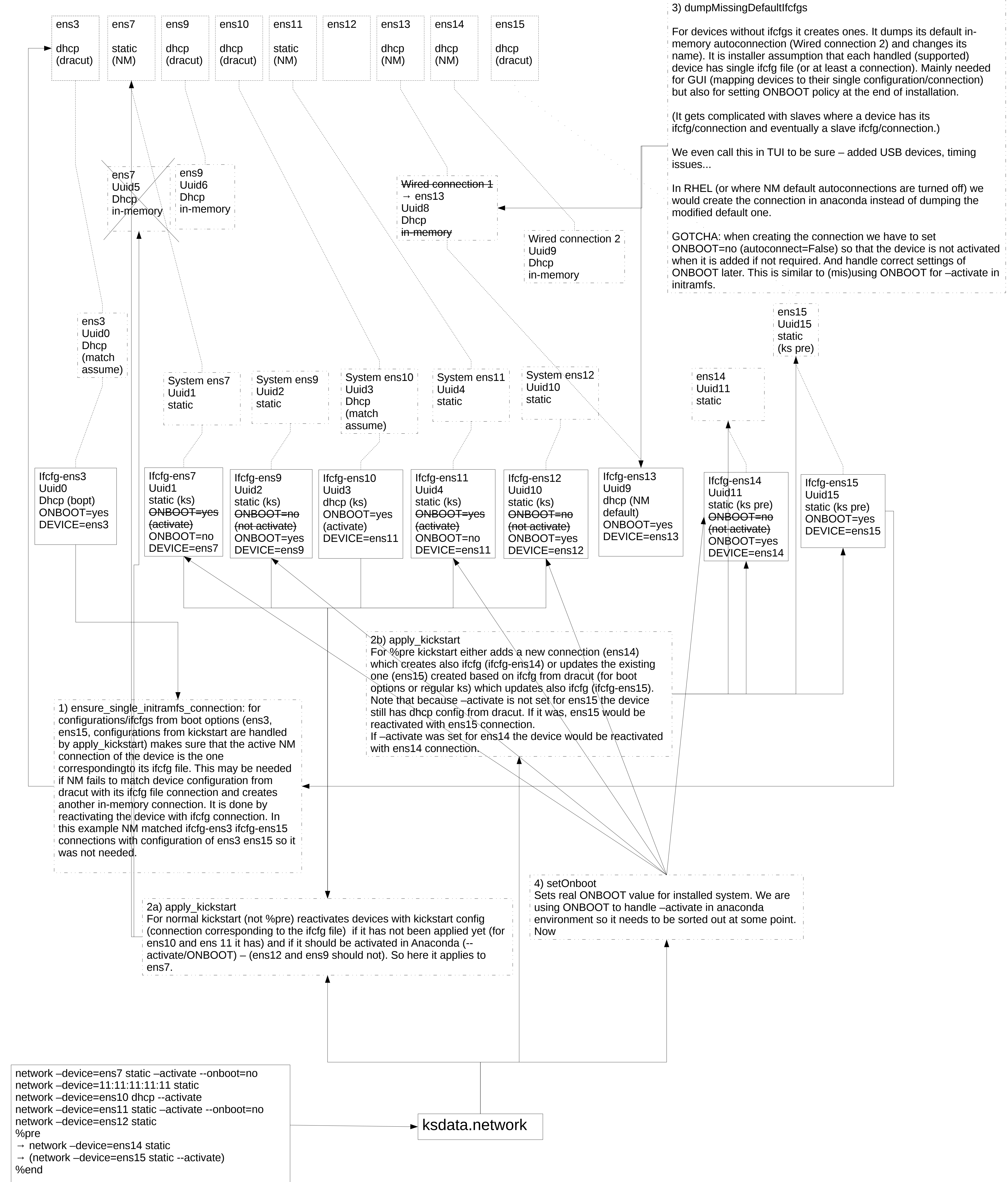


## NETWORK MANAGER STARTS



## ANACONDA - NetworkInitialize

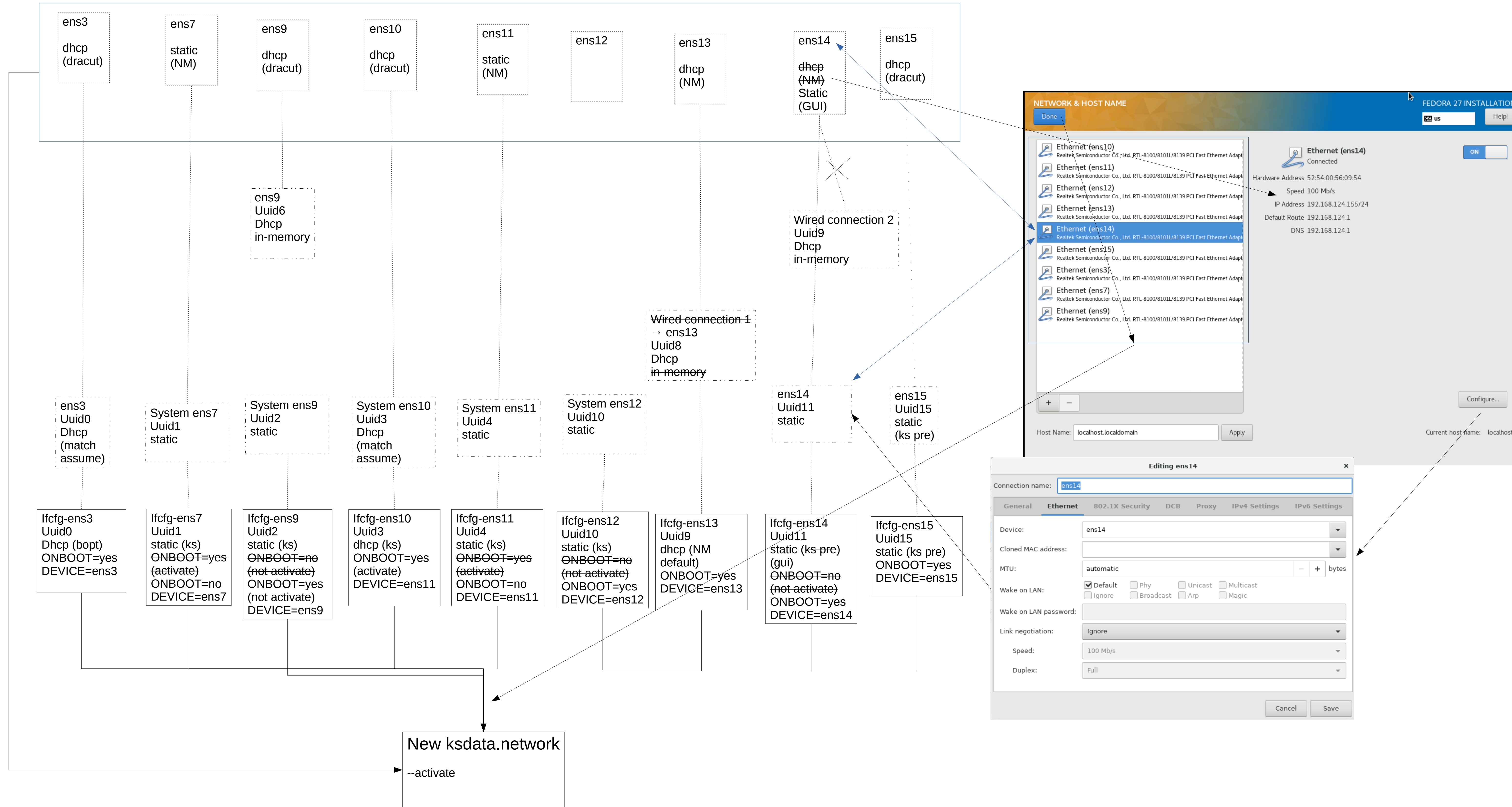
- This all operates only with devices supported by anaconda.
- In most cases ifcfgs are manipulated via NM connections, sometimes directly (setOnboot).





## ANACONDA – GUI

- We display only supported devices (and no slaves) in a list of DeviceConfiguration objects.
- The object maps a device (libnm device object) to its ifcfg connection (libnm connection object) which is the single connection that can be edited with [Configure] (→ nm-c-e connection\_uid).
- Creating the list (DeviceConfiguration objects) and updating it during changes (adding virtual devices) is tricky. For wired devices (eth, infiniband) the device is primary in DeviceConfiguration. For virtual devices (bond, vlan, team, bridge) it is the connection which is primary/persistent (and can be added/removed with [+/-]). Virtual devices are ephemeral.
- UI shows current device configuration which can be different from persistent ifcfg connection (ens14, ens15, ens9). It can even have its own different (in-memory) connection (ens14 from NM start or ens9 from dracut) active. The in-memory connection will be replaced by the persistent (ifcfg) one (but not deleted! Compare with ens7 and ens9) if the device is reactivated by NM (turned [OFF] and [ON] in UI). Or after it is configured because:
- After editing the device is automatically reactivated with edited connection (users requirement)!
- On apply we generate ksdata.network for each DeviceConfiguration in the list. We do it by parsing ifcfg files, not from NM connections. Why? mainly because of ONBOOT? Not sure. Also we set --activate based on whether the device is active now.





# ANACONDA – network.execute → write\_network\_config

