Dnsconfd
system integrated DNS cache

tkorbar@redhat.com, pemensik@redhat.com

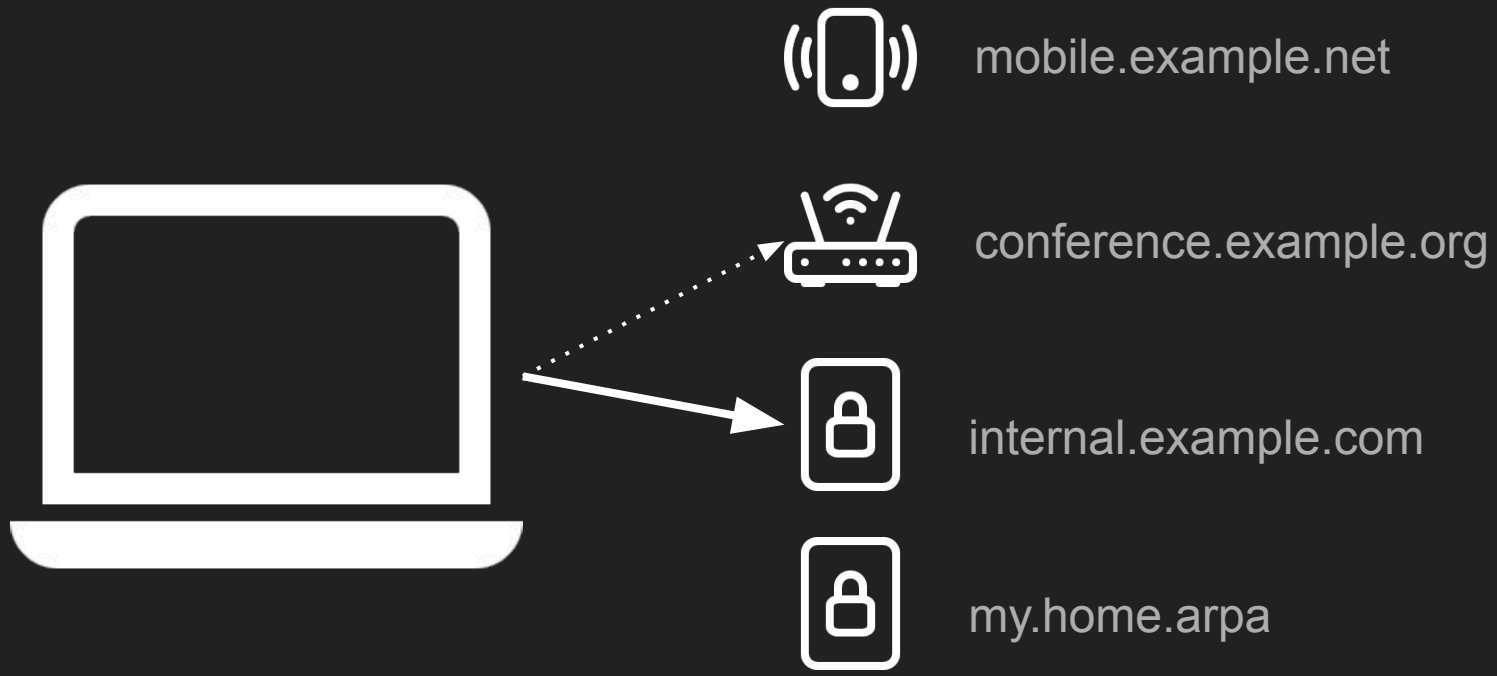
Motivation

- Provide easily extendable DNS cache manager
- Unify configuration of different services
- Allow split-DNS configuration of services
- Full integration with system autoconfiguration
- Configurable behavior
- Avoid known regressions
- Do **not** reinvent the wheel, use existing implementations
- Provide ability to setup DNS over TLS

Fedora 33 change: systemd-resolved

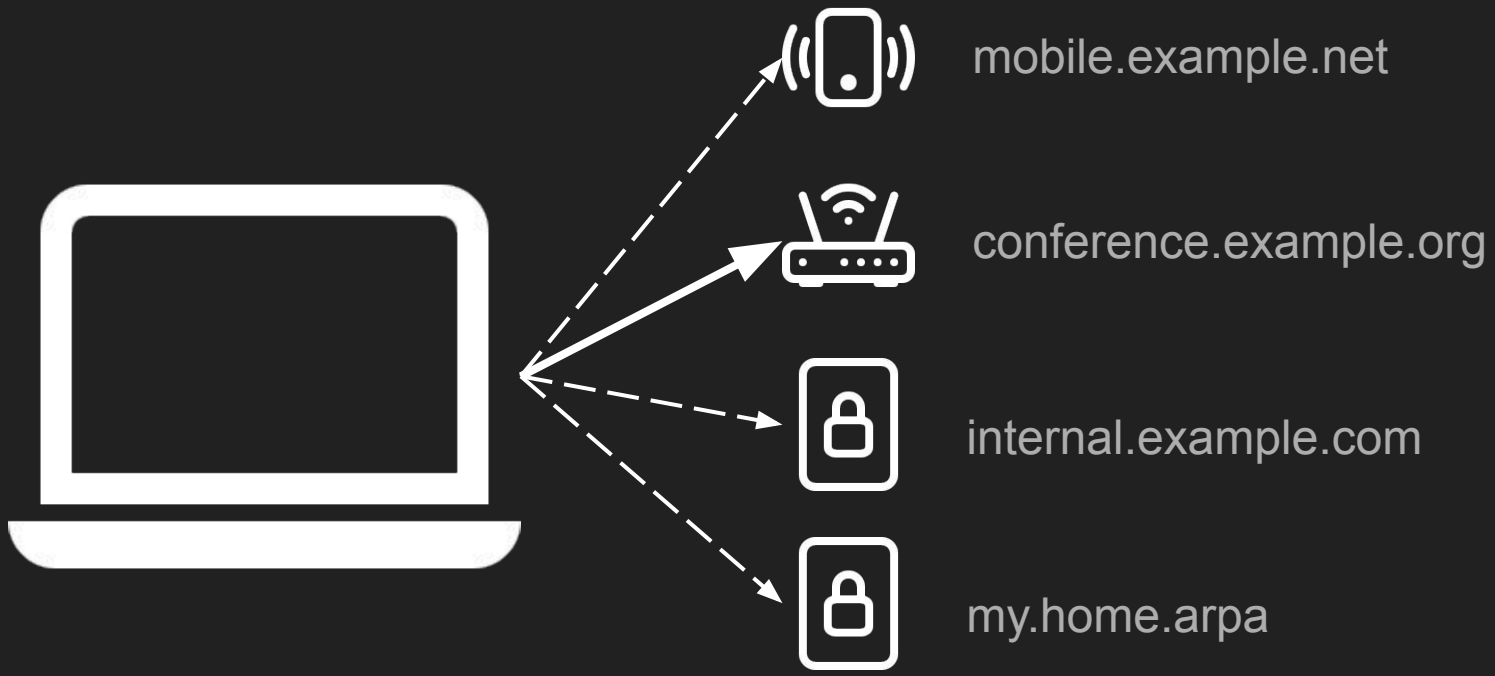
- https://fedoraproject.org/wiki/Changes/systemd-resolved
- DNS cache become enabled by default on both Workstation and Server
- Introduced split-DNS ability
  - Multiple simultaneous VPN connections
  - Global DNS server(s), but reach selected local domain name(s) using servers provided by network
- DNS over TLS ability, not yet enabled by default
- Excellent configuration presentation by resolvectl command
- Well documented DBus interface for both configuration changes and name resolution
- https://systemd.io/RESOLVED-VPNS/

How works default route approach?



Icons from icons.getbootstrap.com

How works split-DNS approach?



Icons from icons.getbootstrap.com

Fedora 33 change: systemd-resolved regressions

- Breaks and prevents usage of DNSSEC
  - EDNS0 DO bit set is never included in forwarded messages
  - systemd#4621, systemd#19227
- Enabled LLMNR does not forward single label names to DNS
  - dig ns com returns NXDOMAIN
  - dig ns github.com returns positive answer
  - Even on Server edition
  - systemd#16059, systemd#23622
- Servfail response may still contain answer
  - Even with DNSSEC=yes, which correctly fails
  - dig +short dnssec-failed.org still prints the address
  - systemd#24827

Lessons learned

- We want split-DNS functionality and DNS over TLS
- Previous alternatives offered poor user frontends (dnsmasq in NM)
- Systemd people have undeniable expertise in service integration
  - But lack expertise in DNS protocol area
- DNS resolvers people have undeniable expertise in DNS protocol
  - But lack expertise in system integration (and with DBus)
- Integration with existing DNS caches is missing
  - Required functionality is already present
  - Both basic and advanced features are available, but configuration syntax varies a lot
  - Nice frontend for DNS configuration information is missing

What is needed for Split-DNS?

- Receiving queries on single unchanging address (localhost)
- Ability to configure different domains forwarded to different set of servers
- Ability to reconfigure without stopping the service

Open source resolvers available

Resolver	Split-DNS	DNS over TLS	DBus	Reconfiguration tool
dnsmasq	✅	❌	✅	DBus, SIGHUP
Bind 9.18	✅	❌	❌	rndc
Bind 9.19+	✅	✅	❌	rndc
Unbound	✅	✅	❌	unbound-control
Knot resolver	✅	✅	❌	kresc
PDNS recursor	✅	dnsdist?	❌	rec_control, web
dnsdist	✅	✅	❌	dnsdist, web?
systemd-resolved	✅	✅	✅	DBus, resolvectl

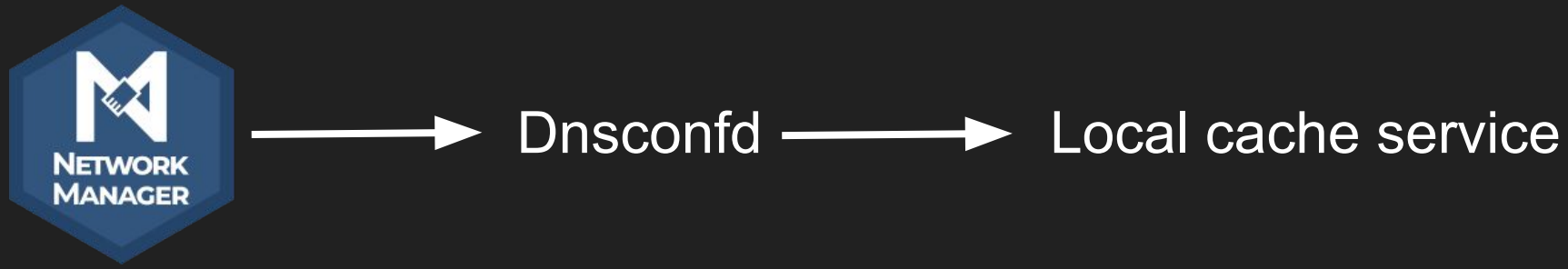
Our approach

- Reuse what already exists if possible
- Provide just user frontend and components coordination
  - Do not try to handle DNS queries ourselves, existing implementations do that well already
  - Almost every open source resolver has required features already
- Verify this idea might work with prototype written in Python 3
- Just single thread
- Set our /etc/resolv.conf only when running, restore it on service stop
- Use standalone daemon, because a lot of VPNs might not use NM
  - Openvpn, wireguard, libreswan may want custom domains redirection too
  - Third party VPN providers as well
  - Virtualization like libvirt might want to delegate name subtree to separate daemon (dnsmasq)
- Dnsconfd handles configuration per interface
  - cache receives just domain names and addresses

Our approach #2

- Configuration should be done in Network Manager
  - We should be just implementation detail for most services
- Relatively small specialized module is needed for specific cache configuration
  - We plan to support unbound, bind9 and dnsmasq
- Provide backward compatibility for services calling systemd-resolved directly
  - But do not plan to reimplement nss-resolve plugin for glibc

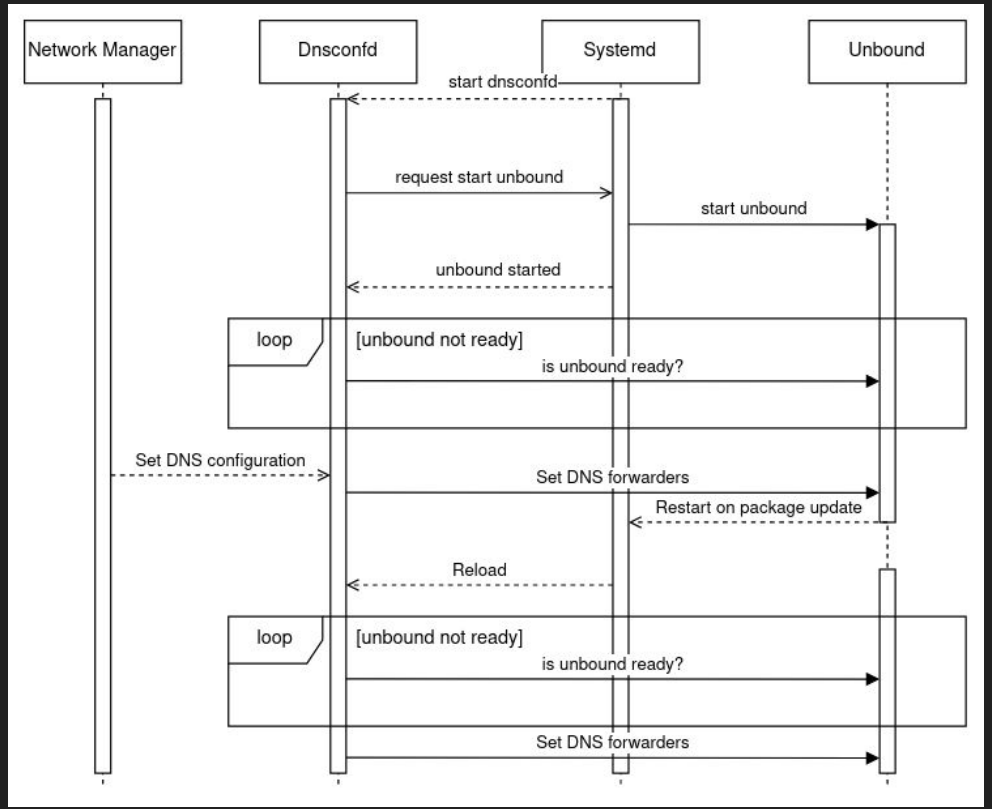
Flow of configuration



System integration

- Dnsconfd uses existing systemd services
- Inherits systemd-resolved DBUS API
- Uses default service configuration
- Watches for service status changes

Dnsconfd life cycle



Issues that we encountered

- Resolv.conf wars
- Subprocess vs system service
- Is unbound truly up?
- Update only updated zones
- DBus is not as simple as we thought it would be

Propose new behaviour

```
rlPhaseStartTest
    sleep 2
    rlRun "podman exec $dnsconfd_cid nmcli connection mod eth0 ipv4.gateway '' ipv4.addr '' ipv4.method auto"
        \ 0 "Setting eth0 to autoconfiguration"
    sleep 2
    rlRun "podman exec $dnsconfd_cid dnsconfd --dbus-name=$DBUS_NAME status > status1"
        \ 0 "Getting status of dnsconfd"
    rlAssertNotDiffer status1 $ORIG_DIR/expected_status.json
    rlRun "podman exec $dnsconfd_cid getent hosts first-address.test.com | grep 192.168.6.3"
        \ 0 "Verifying correct address resolution"
    rlRun "podman exec $dnsconfd_cid getent hosts second-address.test.com | grep 192.168.6.4"
        \ 0 "Verifying correct address resolution"
    rlRun "podman exec $dnsconfd_cid getent hosts second-address | grep 192.168.6.4"
        \ 0 "Verifying correct address resolution"
rlPhaseEnd
```

What is working already

- Split DNS configuration received from Network Manager
- /etc/resolv.conf temporary change
  - Restores resolution when dnsconfd stopped
- Unbound support only
- Implementation of selected DBus interfaces of systemd-resolved
- We re-use NM systemd-resolved DNS plugin
  - But plan to make it just optional
- No DNS over TLS support yet

Planned features

- Add detailed configuration of behaviour
- Provide working DNS over TLS
- Support alternative caches
- Have auto-configuration of DoT (DDR/DNR)
- Support multiple caches at the same time
- Support also DNS over HTTPS, DNS over QUIC
- Have auto-configuration of DNSSEC
- Rewrite into Rust?

Thanks for your attention!

https://github.com/InfrastructureServices/dnsconfd

On Fedora Rawhide:

dnf install dnsconfd

Contacts:

- Petr Menšík <pemensik@redhat.com>
- Tomáš Korbař <tkorbar@redhat.com>