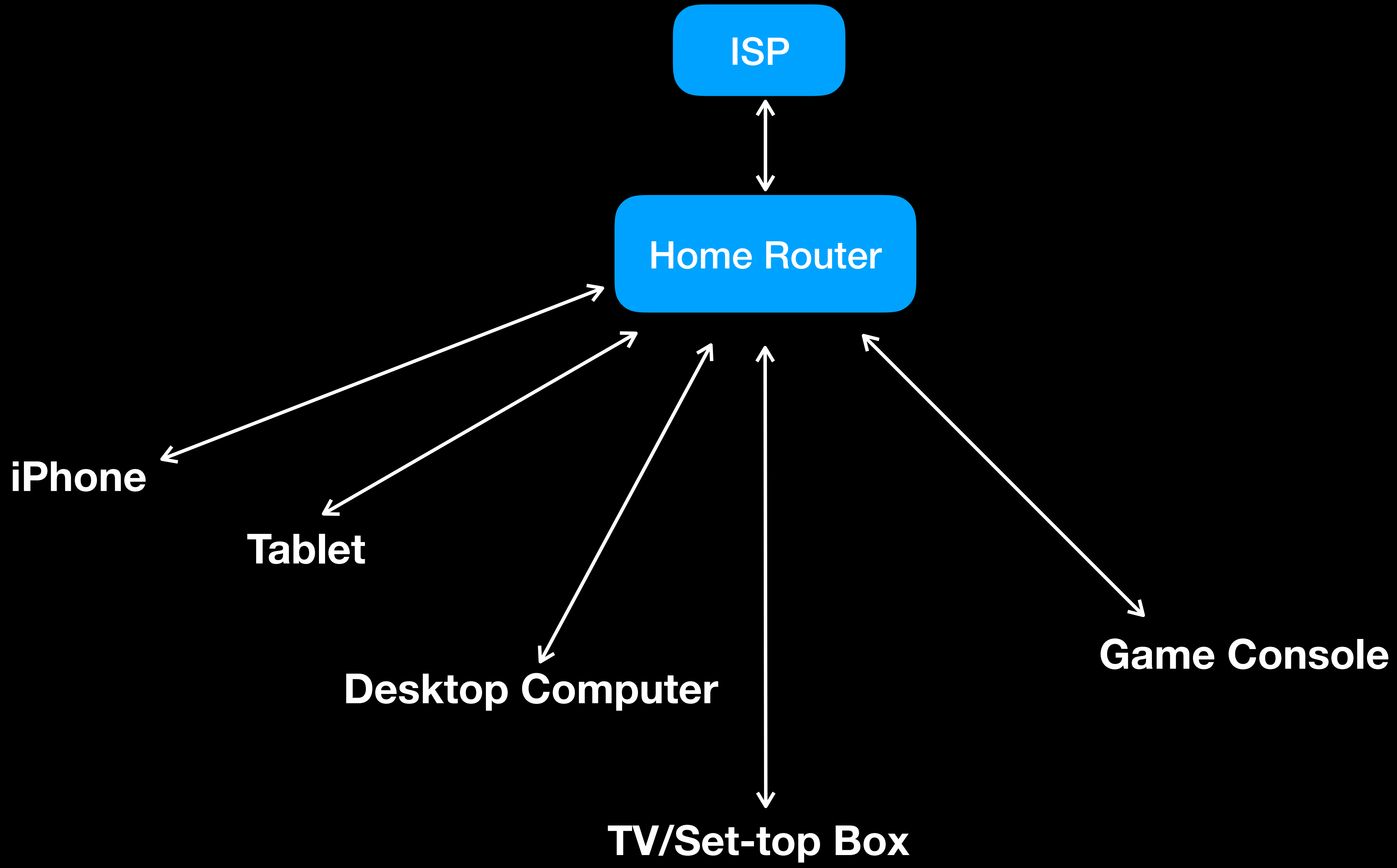
Service Discovery and Trust in a Homenet

Ted Lemon <ted.lemon@nominum.com>

Introduction

- I’m Ted Lemon, I work for Nominum, mostly on forward-looking standards work in the IETF
- The work presented today will be work done in the Homenet working group and DNS Service Discovery working group in the IETF
- I’m the author of the Homenet Naming Architecture, which is a work in progress
- I’m also working with Stuart Cheshire from Apple Computer on improvements to DNS Service Discovery for the homenet
- Some of the work discussed here has not yet been put into working group documents

Home network



ISP

Home Router

iPhone

Tablet

Desktop Computer

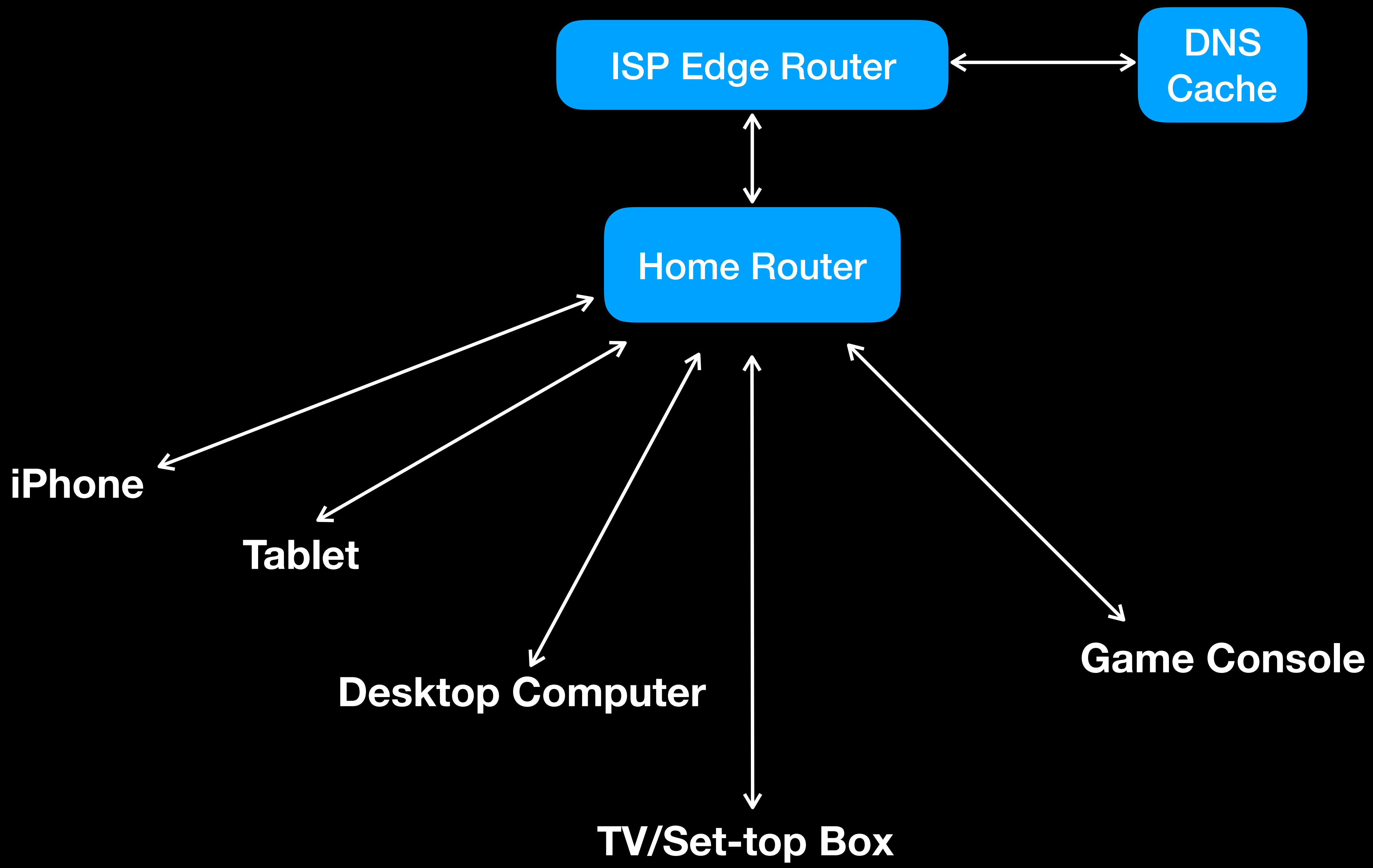
TV/Set-top Box

Game Console

Basic Service Discovery

- Discover Addressing
- Discover Routing
- Discover DNS Server(s)
- Look up services using names and URLS
- This basic pattern is followed by the homenet router and by devices on the homenet
- The key network service required for this to work is DNS

Home network



ISP Edge Router

DNS Cache

Home Router

iPhone

Tablet

Desktop Computer

TV/Set-top Box

Game Console

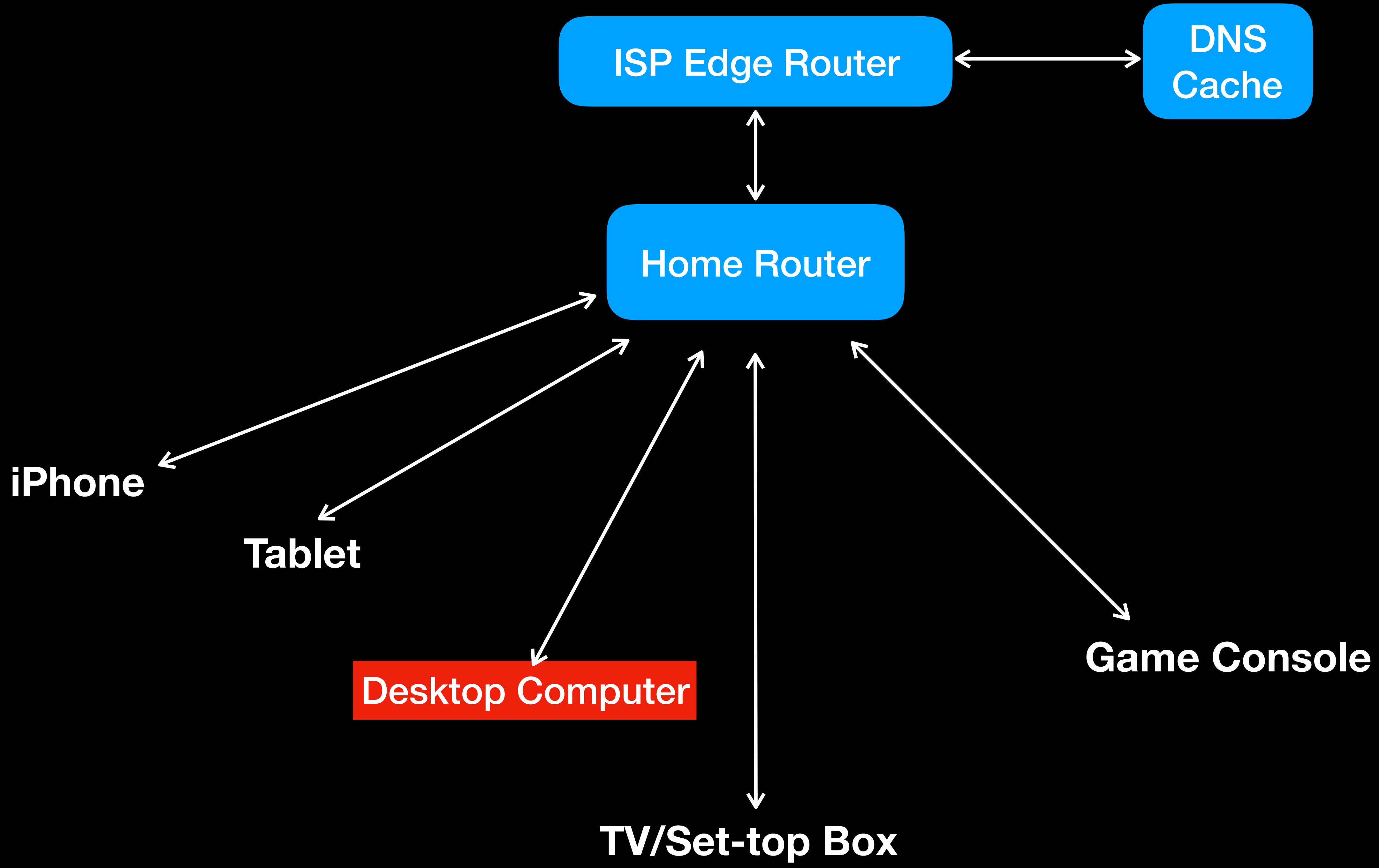
DNS Service

- DNS is how hosts discover IP addresses of services on the Internet

- This includes things like malware servers and botnet command and control servers

- The servers we (Nominum) make use this to detect and block connections to malware servers, and to discover the presence of malware on end-user home networks

Home network



ISP Edge Router

DNS Cache

Home Router

iPhone

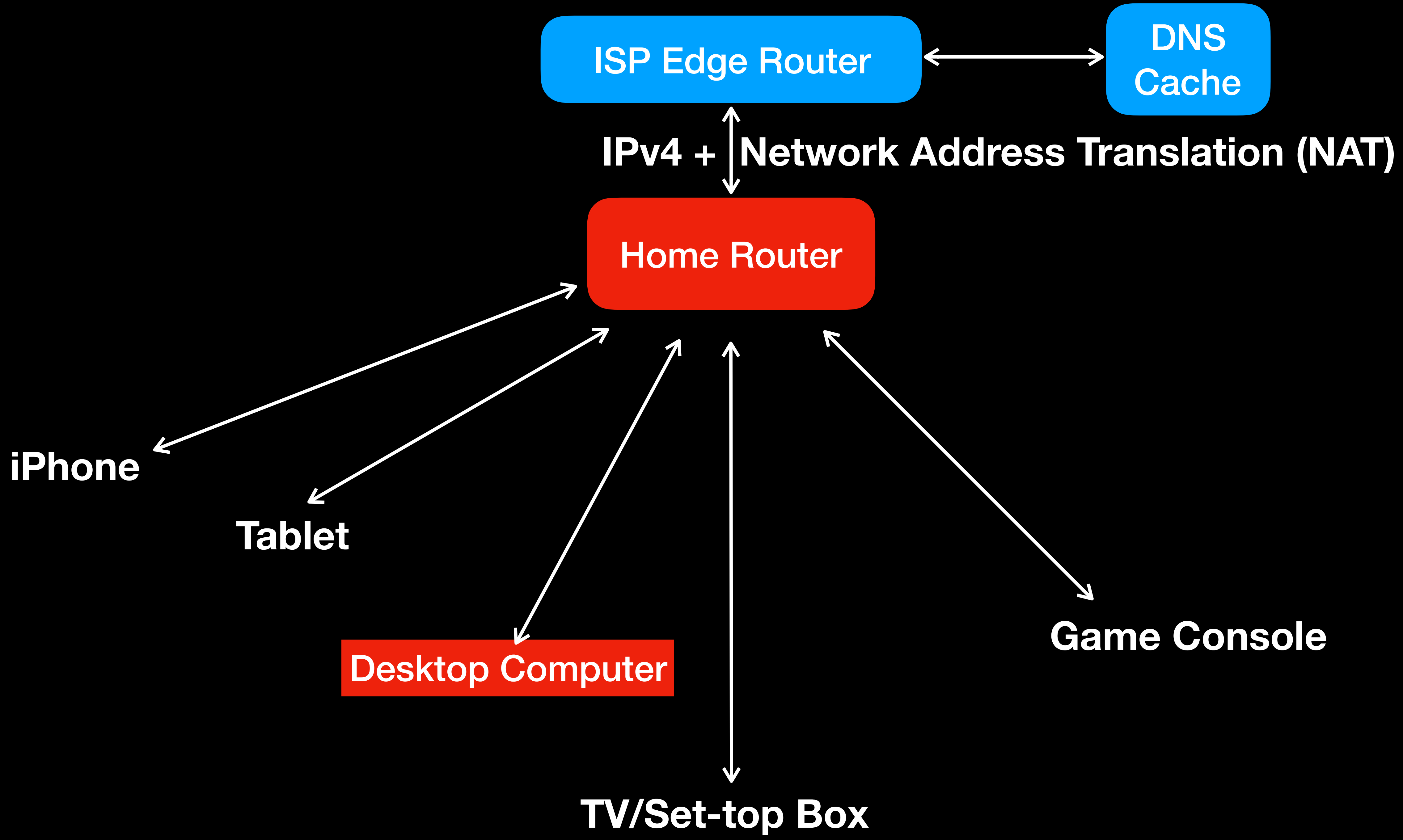
Tablet

Desktop Computer

TV/Set-top Box

Game Console

Home network



ISP Edge Router

DNS Cache

IPv4 + Network Address Translation (NAT)

Home Router

iPhone

Tablet

Desktop Computer

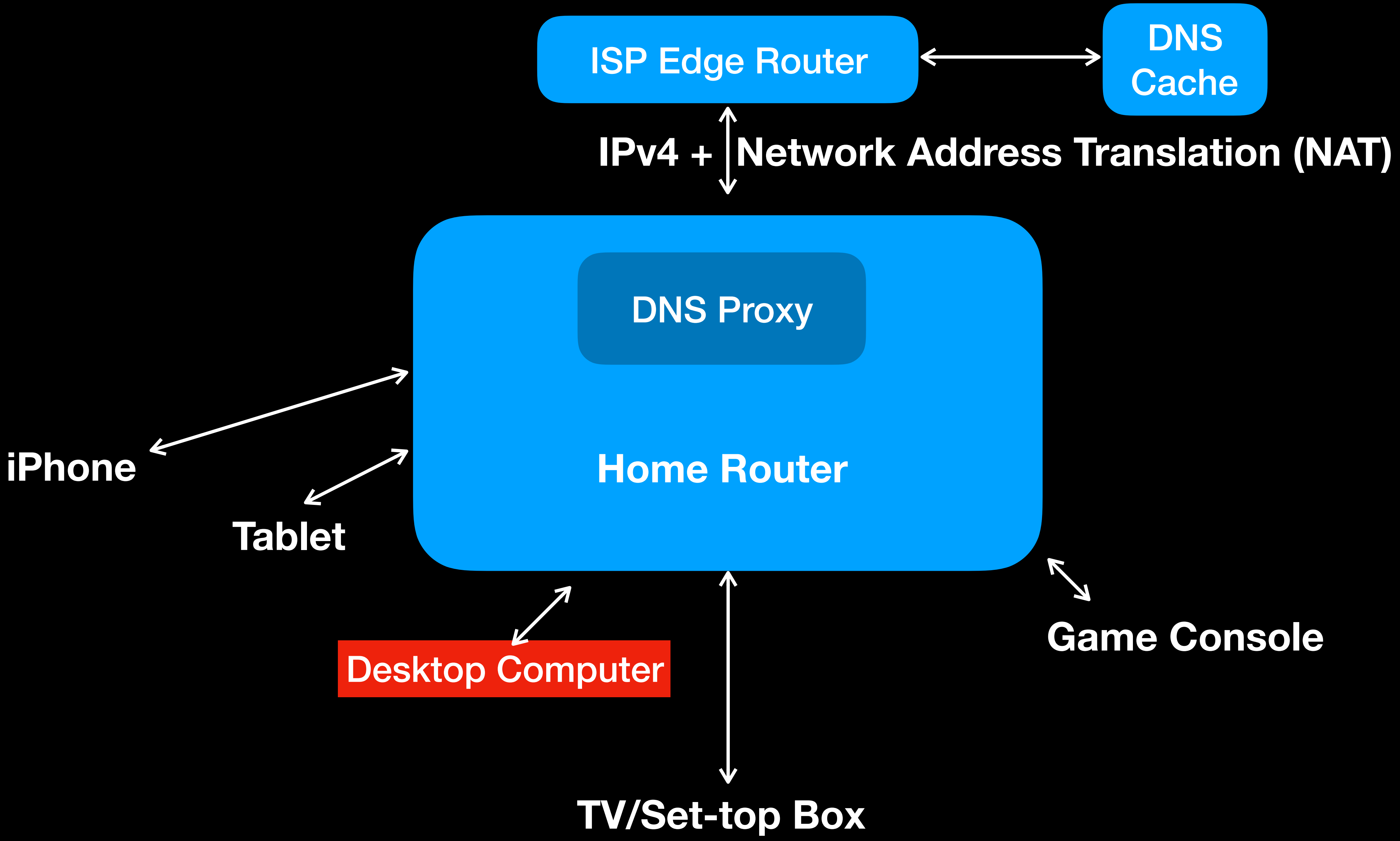
TV/Set-top Box

Game Console

Quarantining with NAT

- A home gateway with NAT has a single IP address
- All queries come from that address
- Can’t quarantine just the infected host
- Solution: put a DNS Proxy in the router that adds identifying information so that we can quarantine the individual host

Home network



ISP Edge Router

DNS Cache

IPv4 + Network Address Translation (NAT)

DNS Proxy

Home Router

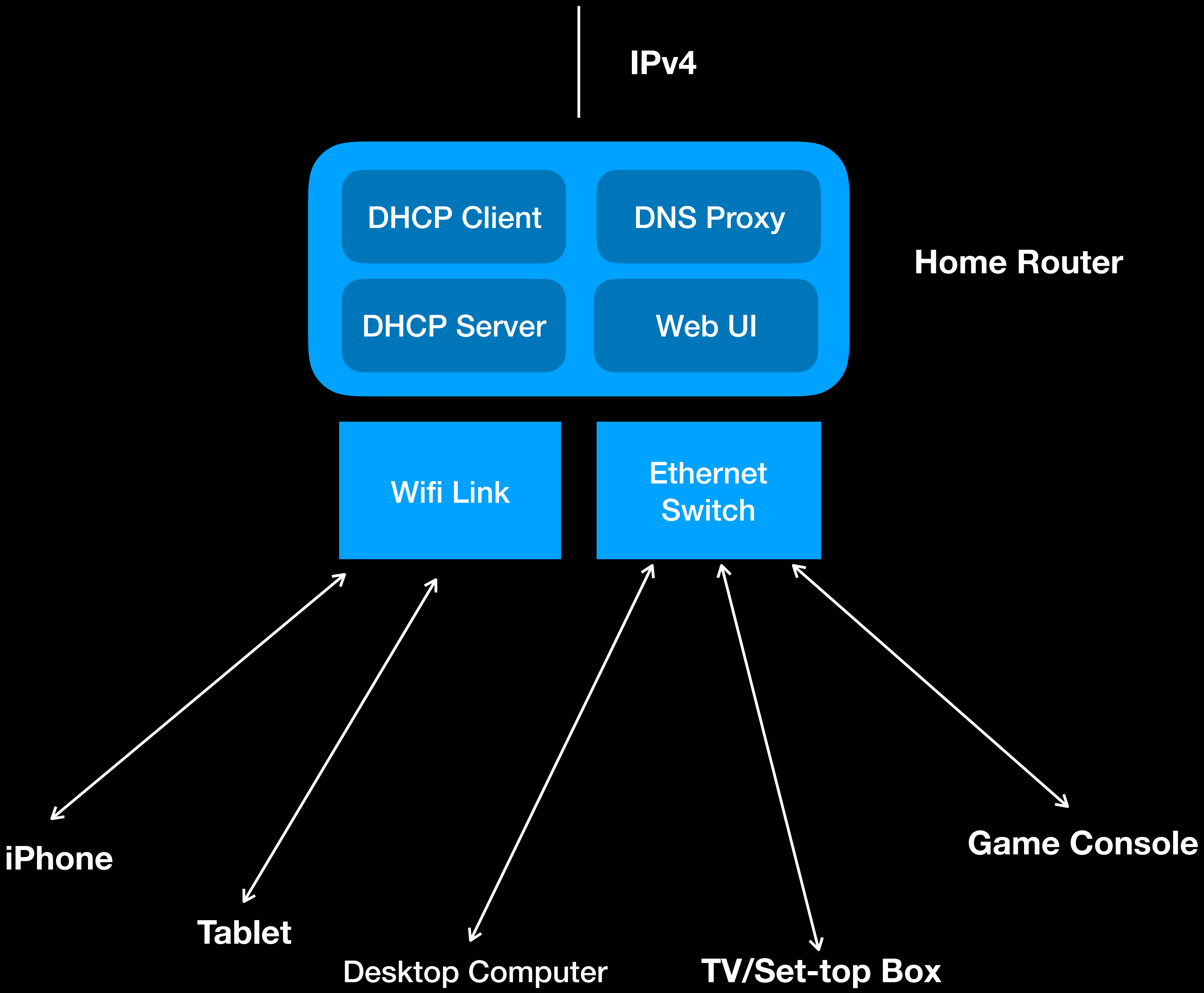
iPhone

Tablet

Desktop Computer

Game Console

TV/Set-top Box



IPv4

Home Router

DHCP Client

DNS Proxy

DHCP Server

Web UI

Wifi Link

Ethernet Switch

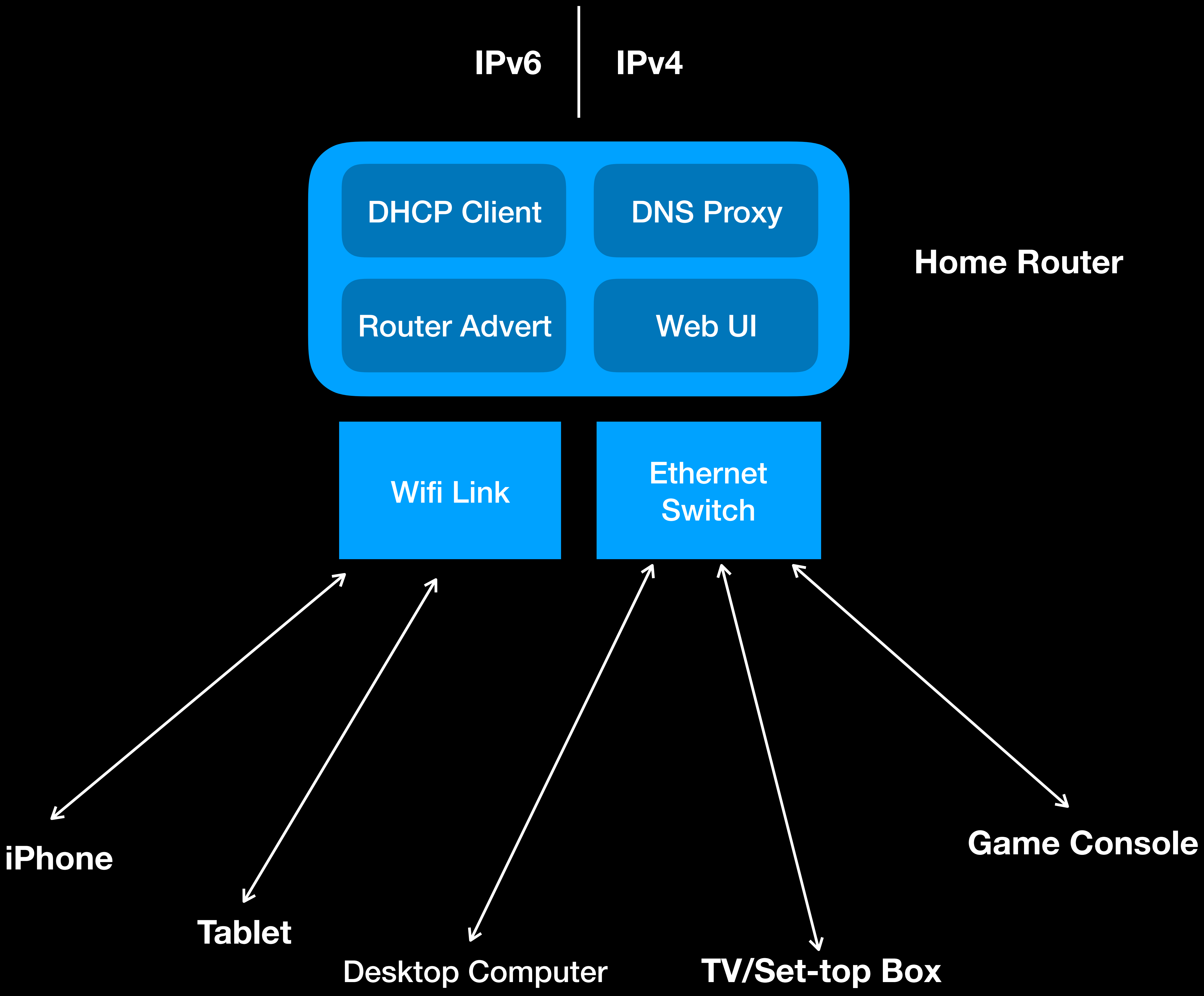
iPhone

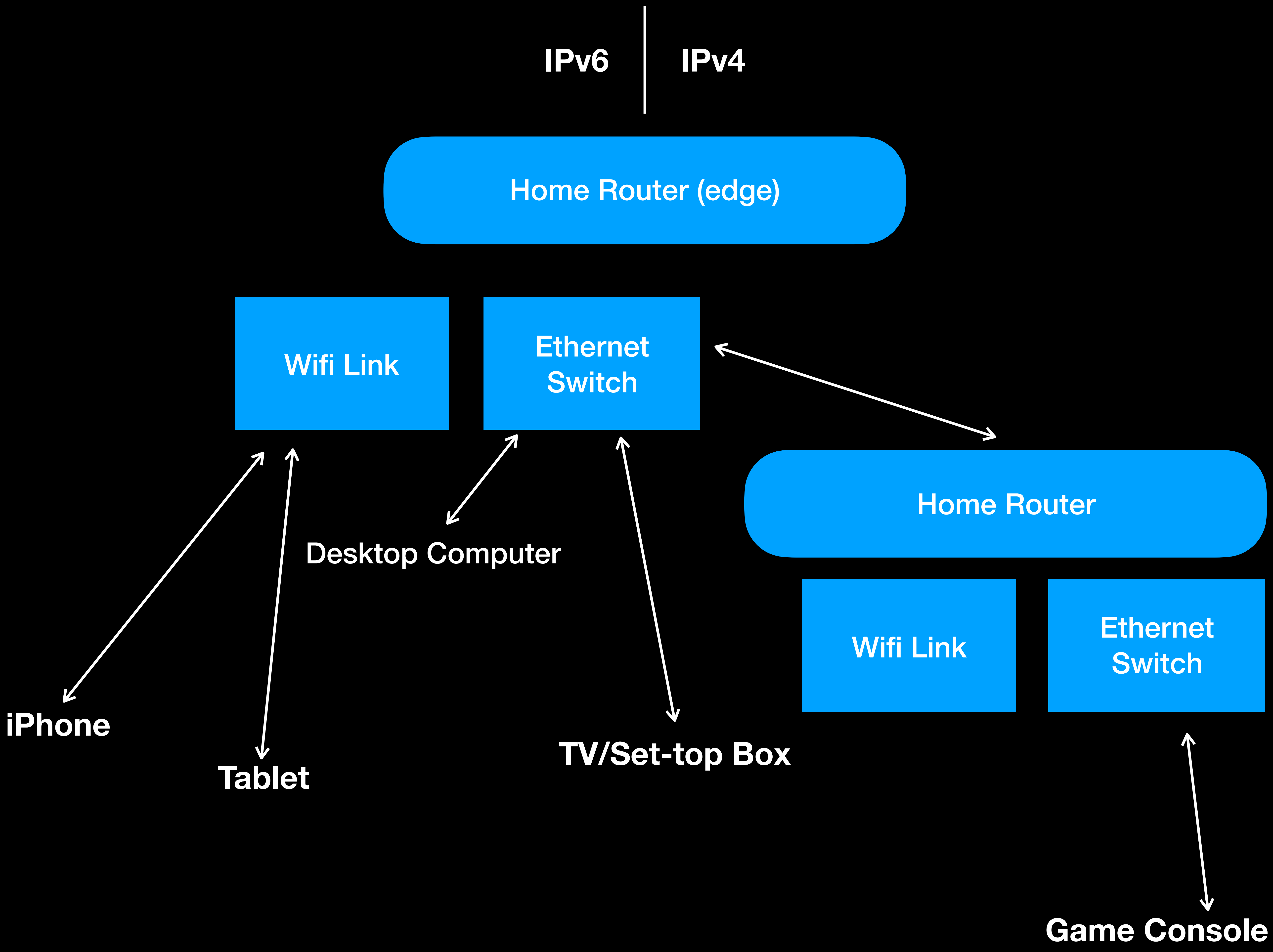
Tablet

Desktop Computer

TV/Set-top Box

Game Console

IPv6
IPv4
DHCP Client
DNS Proxy
Router Advert
Web UI
Home Router
Wifi Link
Ethernet Switch
iPhone
Tablet
Desktop Computer
TV/Set-top Box
Game Console



IPv6

IPv4

Home Router (edge)

Wifi Link

Ethernet Switch

Desktop Computer

Home Router

Wifi Link

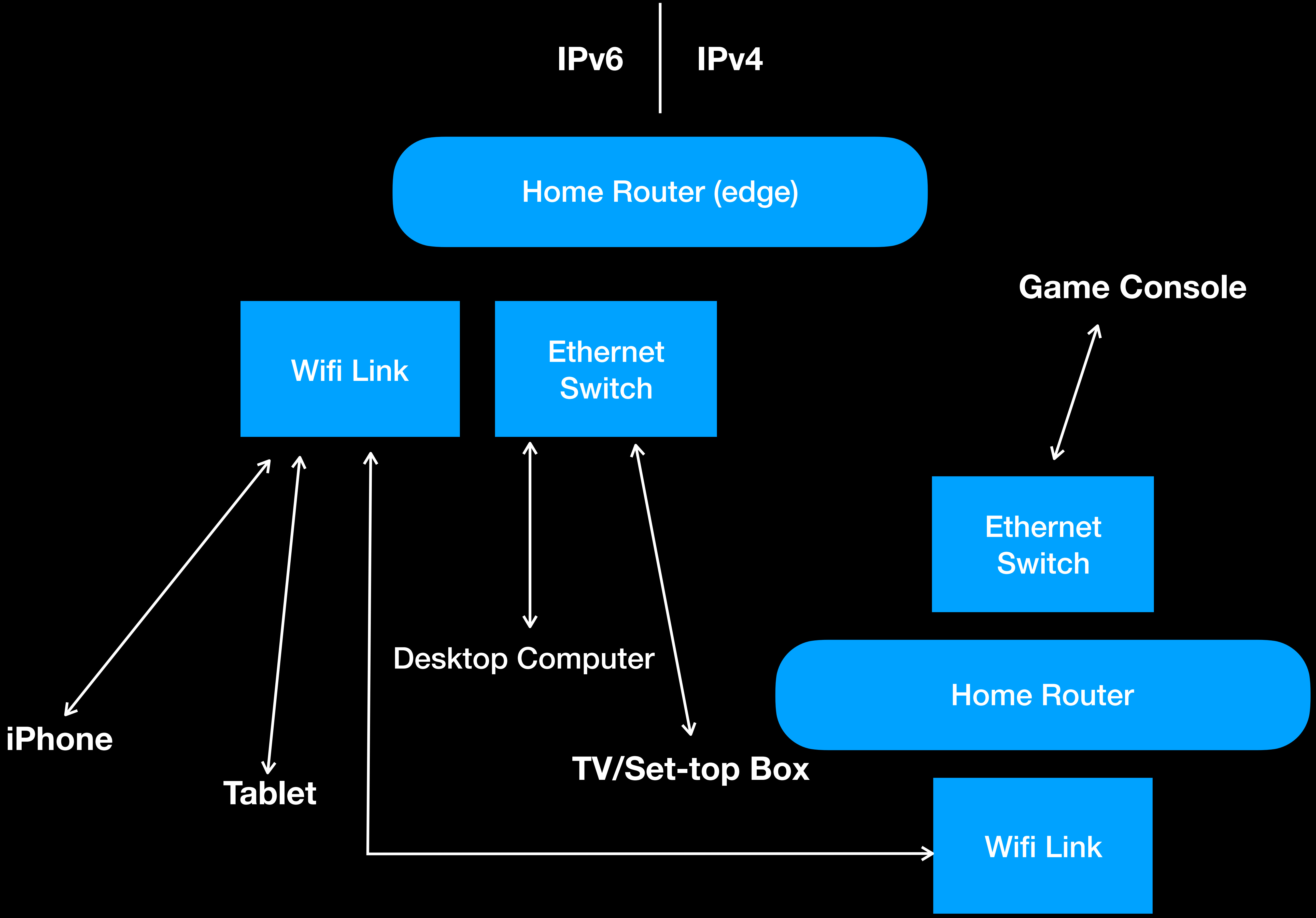
Ethernet Switch

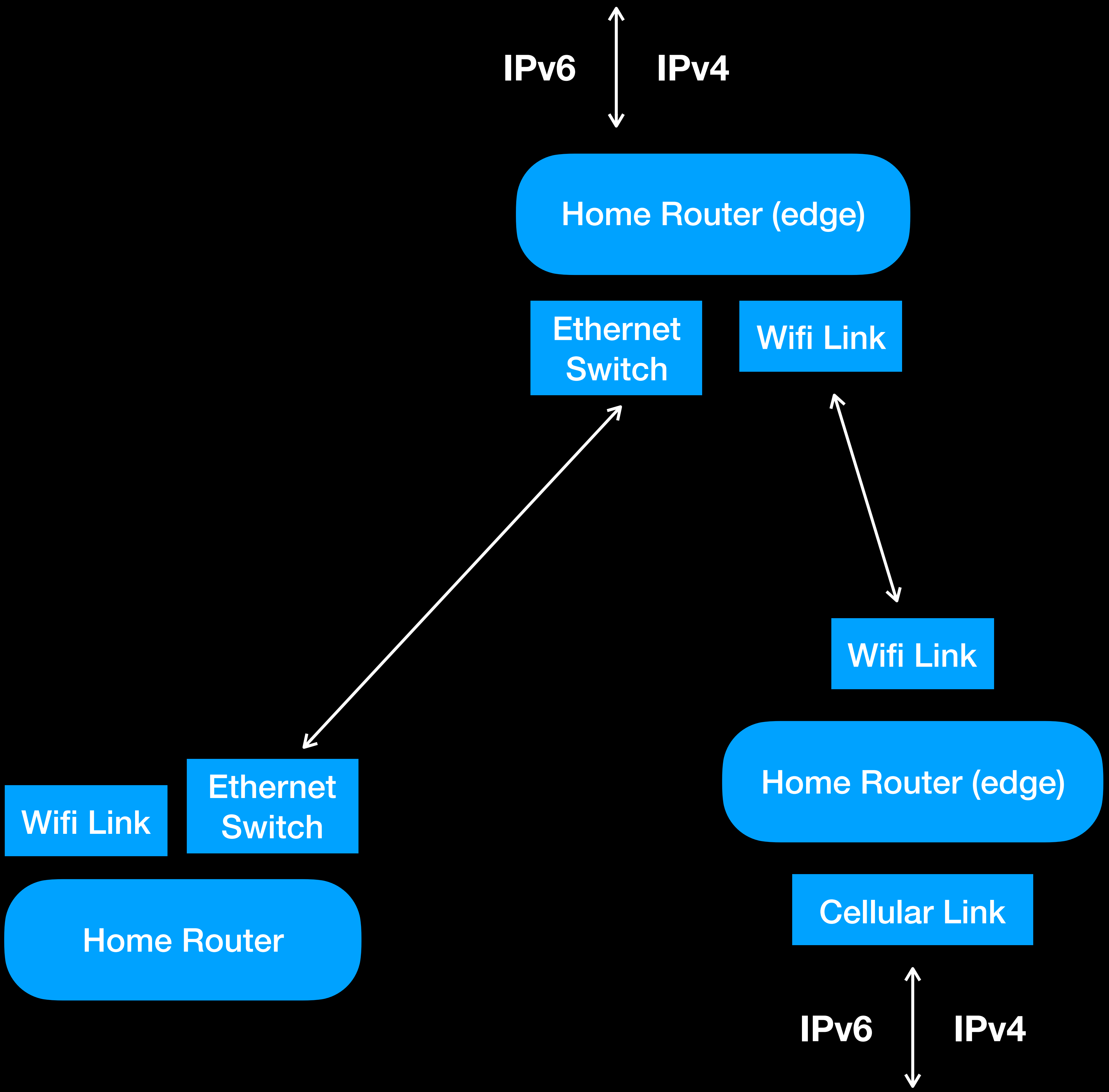
iPhone

Tablet

TV/Set-top Box

Game Console





IPv6

IPv4

Home Router (edge)

Ethernet Switch

Wifi Link

Wifi Link

Home Router (edge)

Cellular Link

IPv6

IPv4

Wifi Link

Ethernet Switch

Home Router

New requirements

- Have a stable address prefix for the home net that works even if the ISP service isn’t available
- Have a prefix from each provider for which a connection exists that can be subdivided to support multiple subnet links
- Route packets to the right ISP, based on the source address chosen by the host (we don’t control)
- Provide service discovery across subnet boundaries
- Configure all of this automatically, with no user intervention
- Support for multiple provisioning domains (RFC 7556)

Protocols

- Routing Protocol: Babel
- Network Management Protocol: HNCP
- Service Discovery Protocol: DNSSD
- 802.11 (SSID) and 802.11i (WPA2 password)
- Homenets are plug and play: plug them together and they start delivering packets and service. No user configuration required.
- Every link in homenet has a separate prefix, and every homenet has a ULA prefix plus zero or more ISP prefixes

Babel

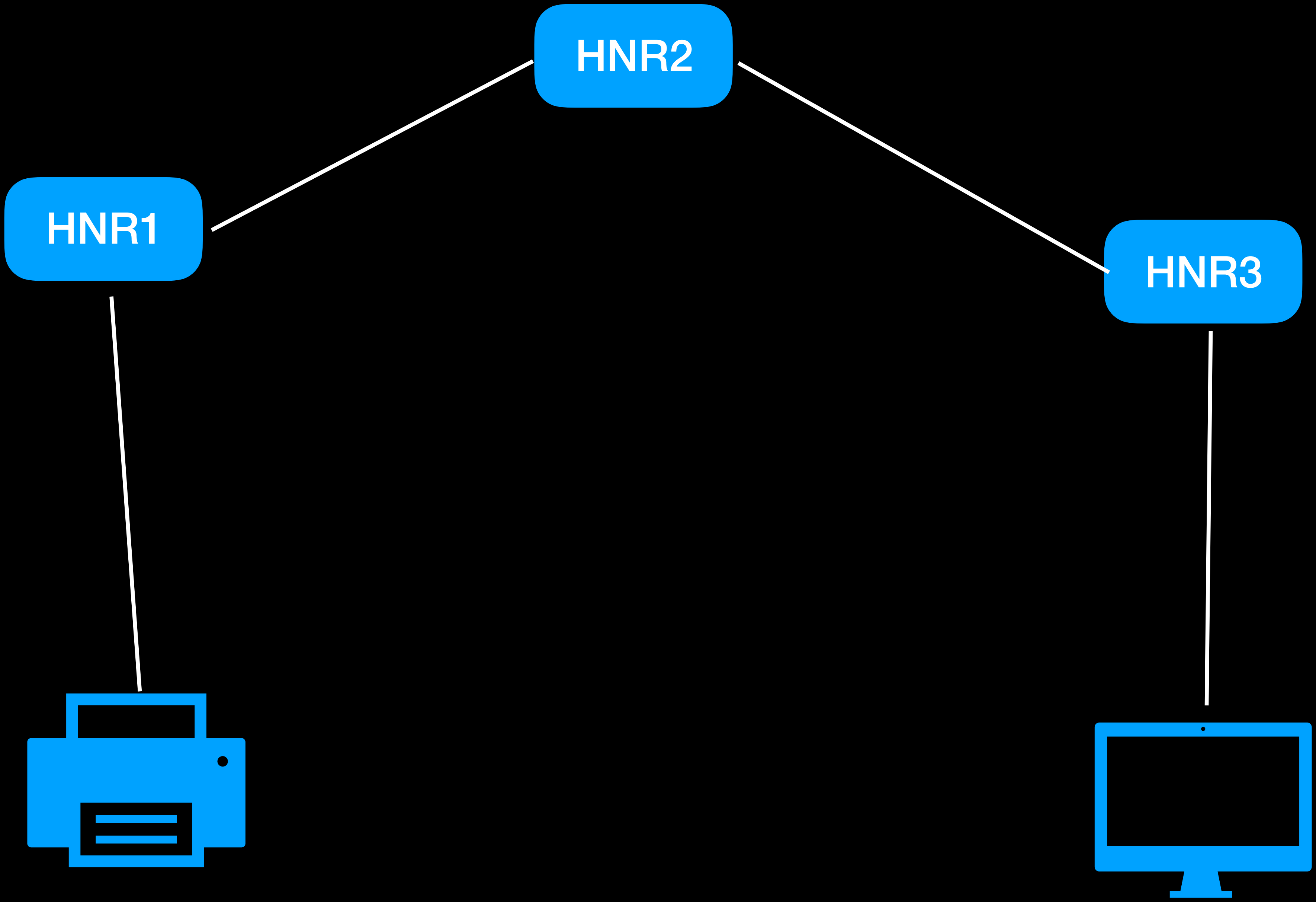
- Good at routing multiply-connected network with links of different quality
- Modified for Homenet to support source-specific routing: whichever prefix a host chooses to connect, that will determine through which ISP that flow is routed.
- Information published by router A may be sent to router B and then consumed by router C
- No security protocol
- Relies on multicast

HNCP

- Flood fill using trickle algorithm
- Identifies network edges
- Identifies internal links
- Identifies routers
- Identifies links between routers
- No encryption, no authentication
- Relies on multicast

DNSSD

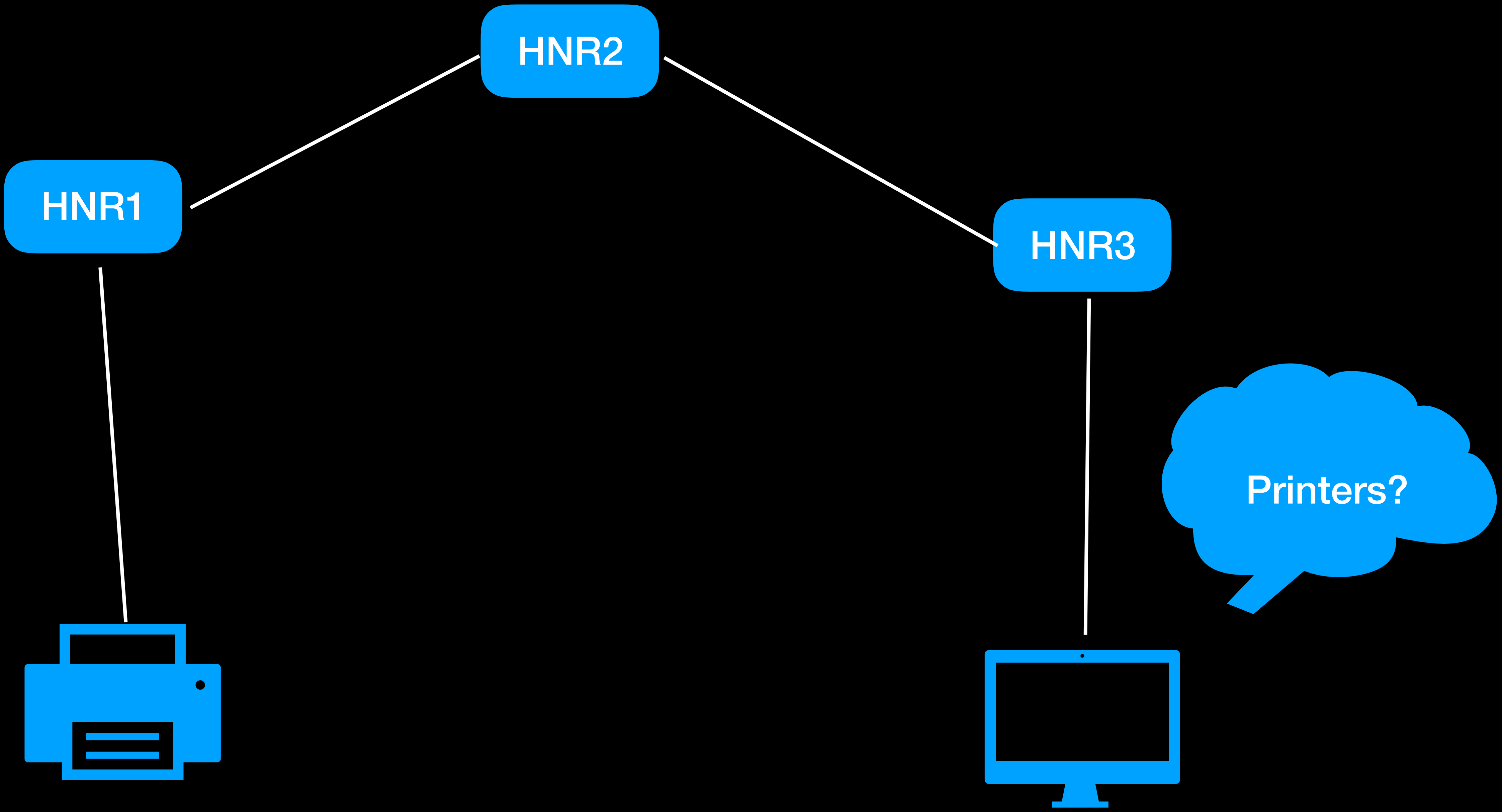
- Uses the DNS protocol
- Uses RFC 6763 service discovery
- Leverages Multicast DNS (RFC 6762)



HNR2

HNR1

HNR3

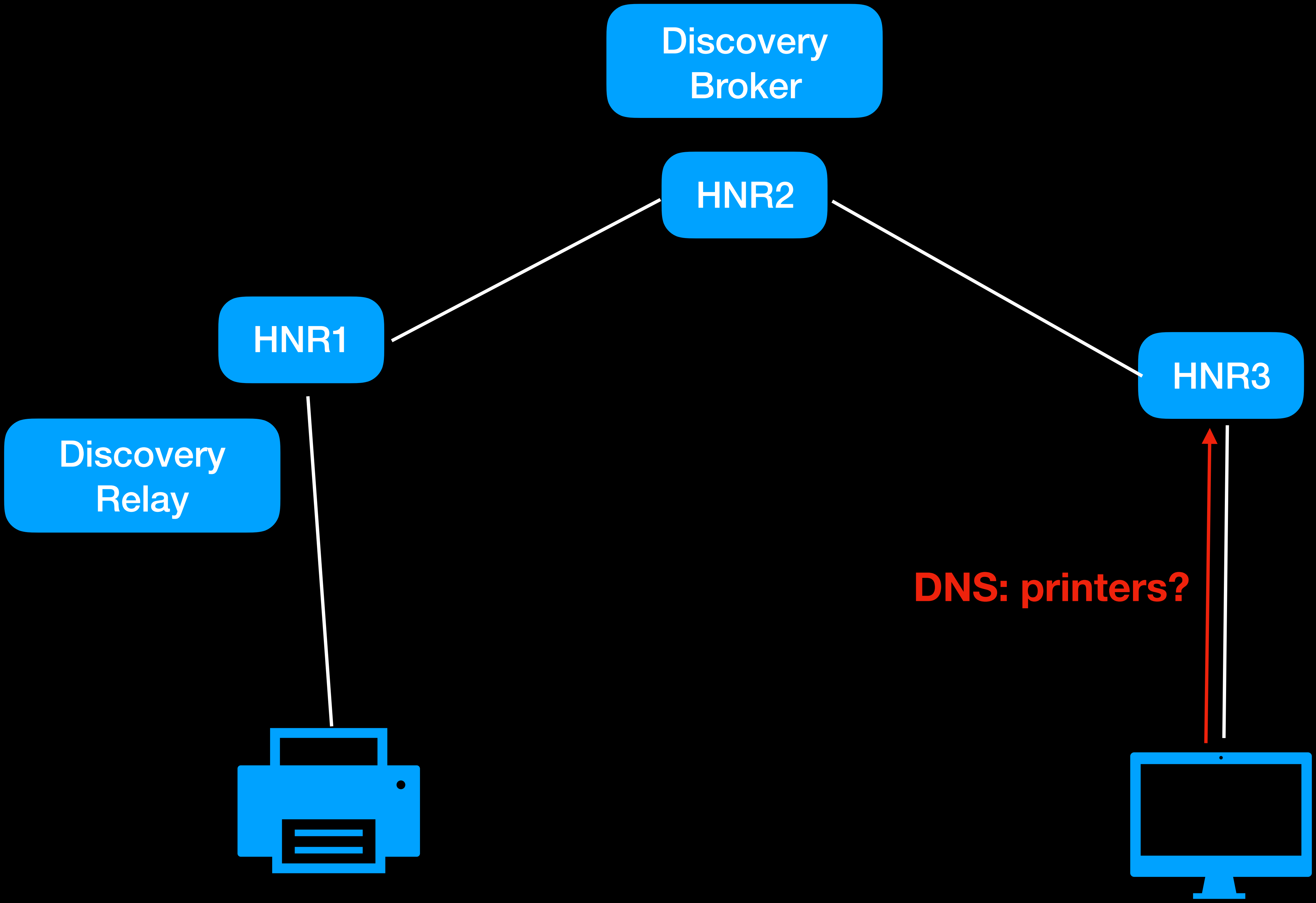


HNR2

HNR1

HNR3

Printers?



Discovery Broker

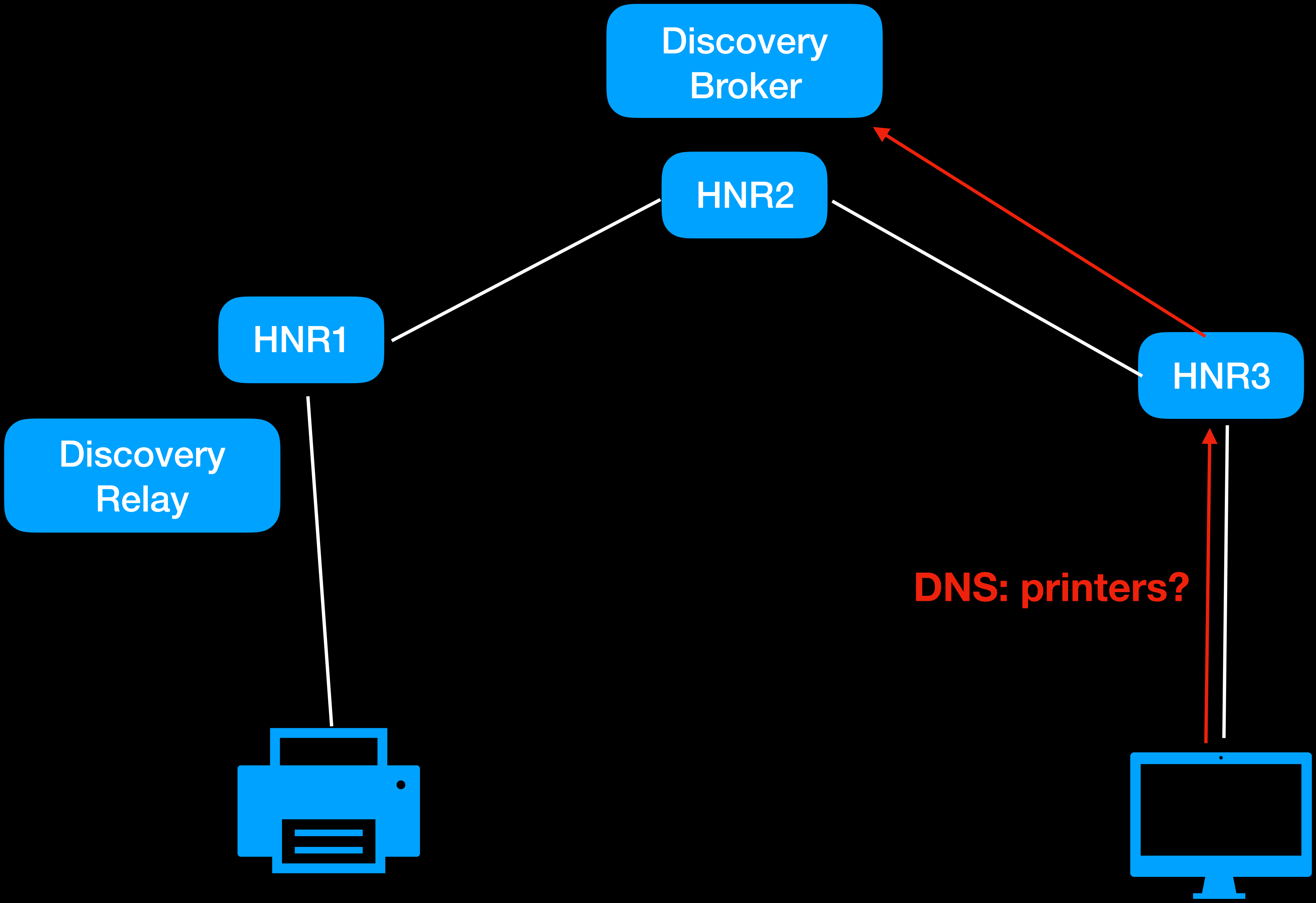
HNR2

HNR1

HNR3

Discovery Relay

DNS: printers?



Discovery Broker

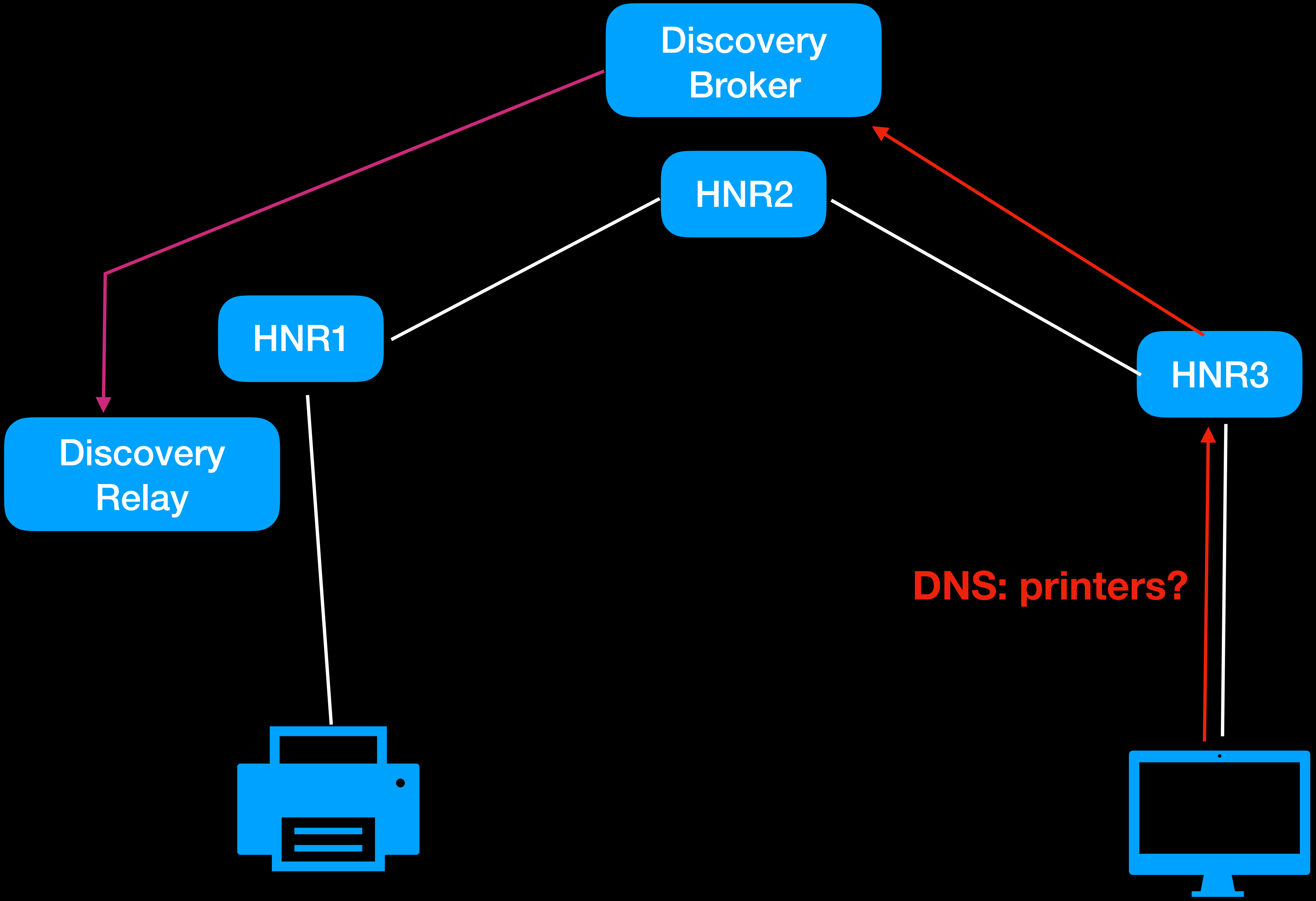
HNR2

HNR1

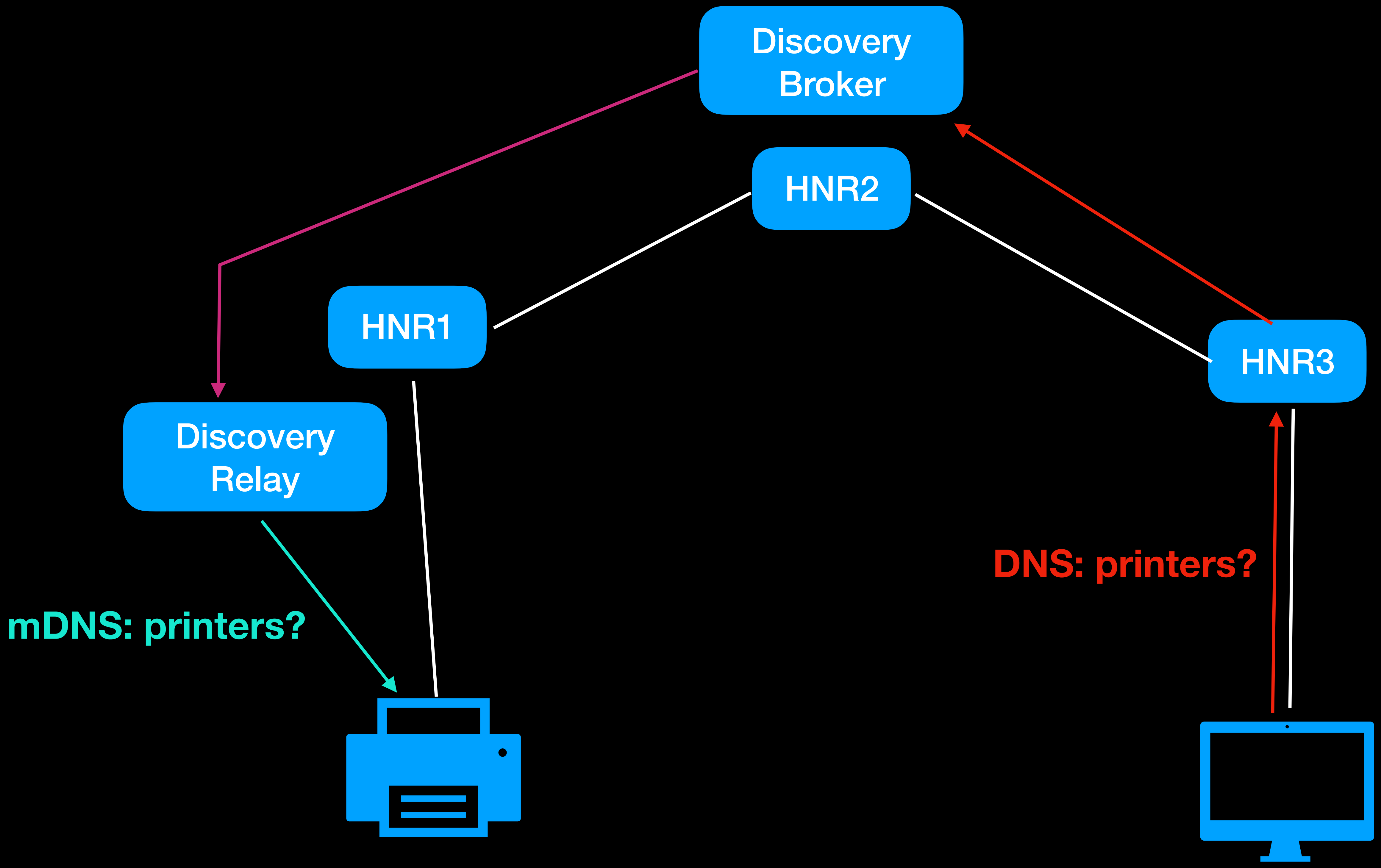
HNR3

Discovery Relay

DNS: printers?



Discovery Broker
HNR2
HNR1
HNR3
Discovery Relay
DNS: printers?



Discovery Broker

HNR2

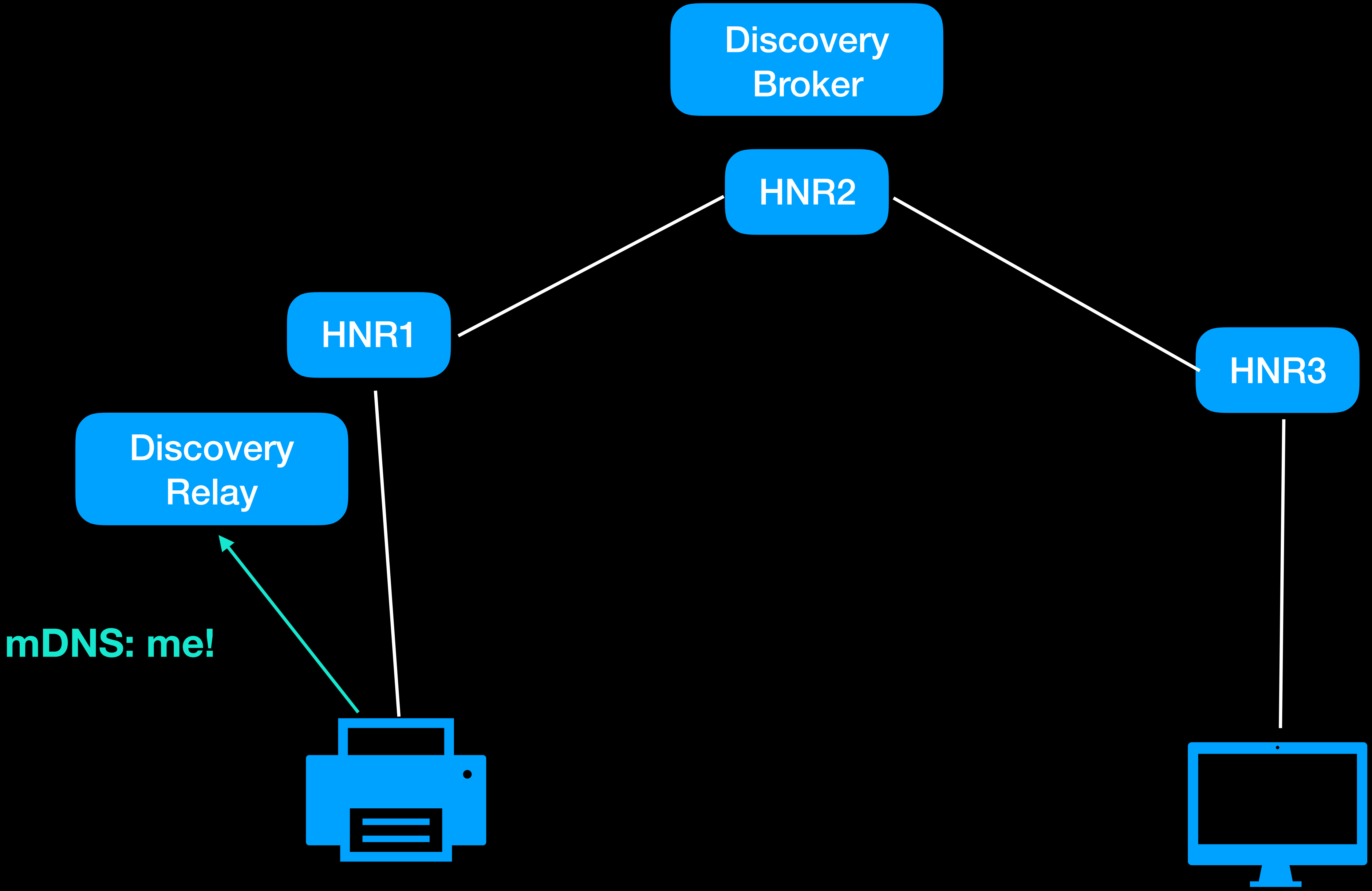
HNR1

HNR3

Discovery Relay

DNS: printers?

mDNS: printers?



Discovery Broker

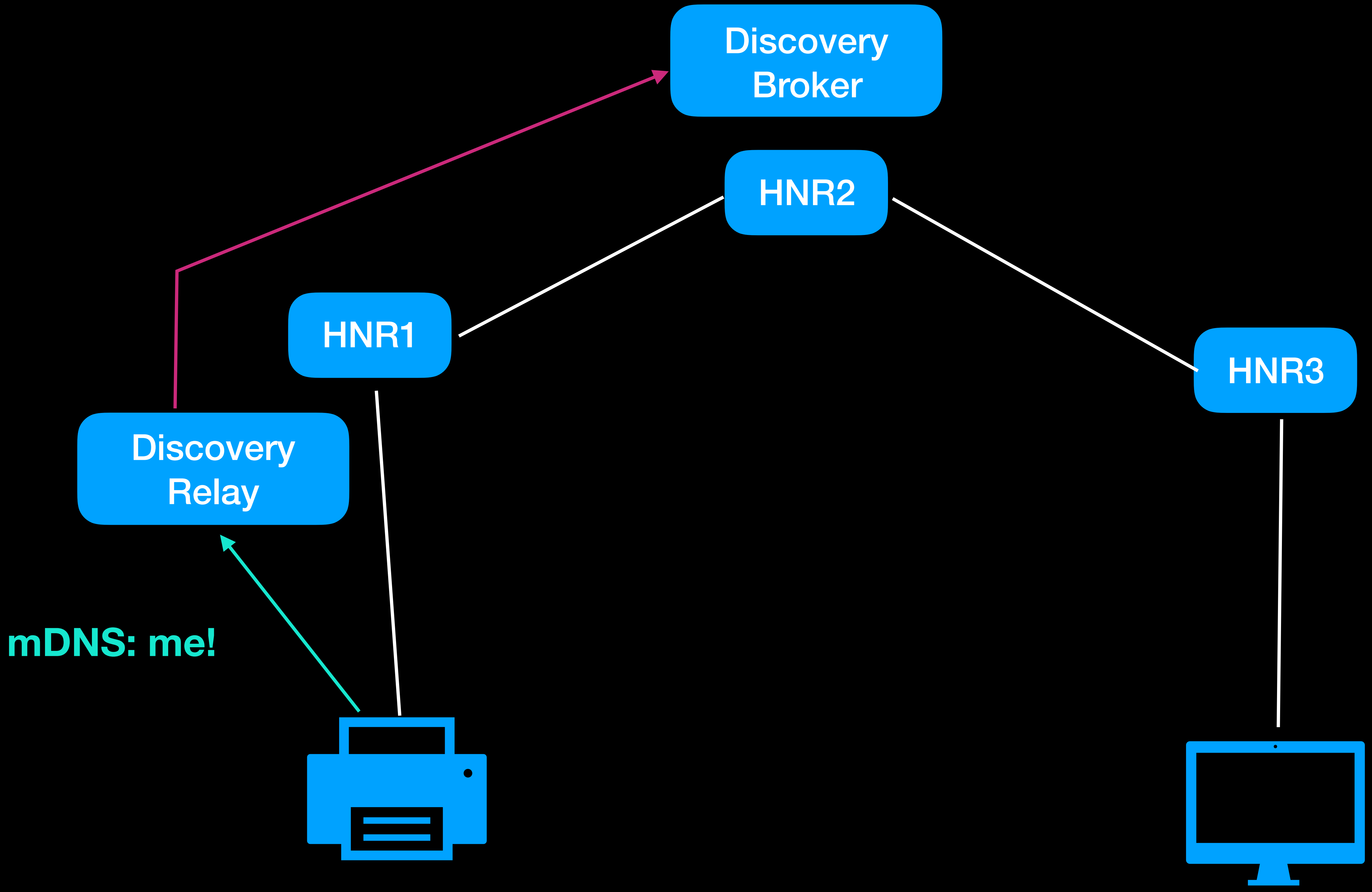
HNR2

HNR1

HNR3

Discovery Relay

mDNS: me!



Discovery Broker

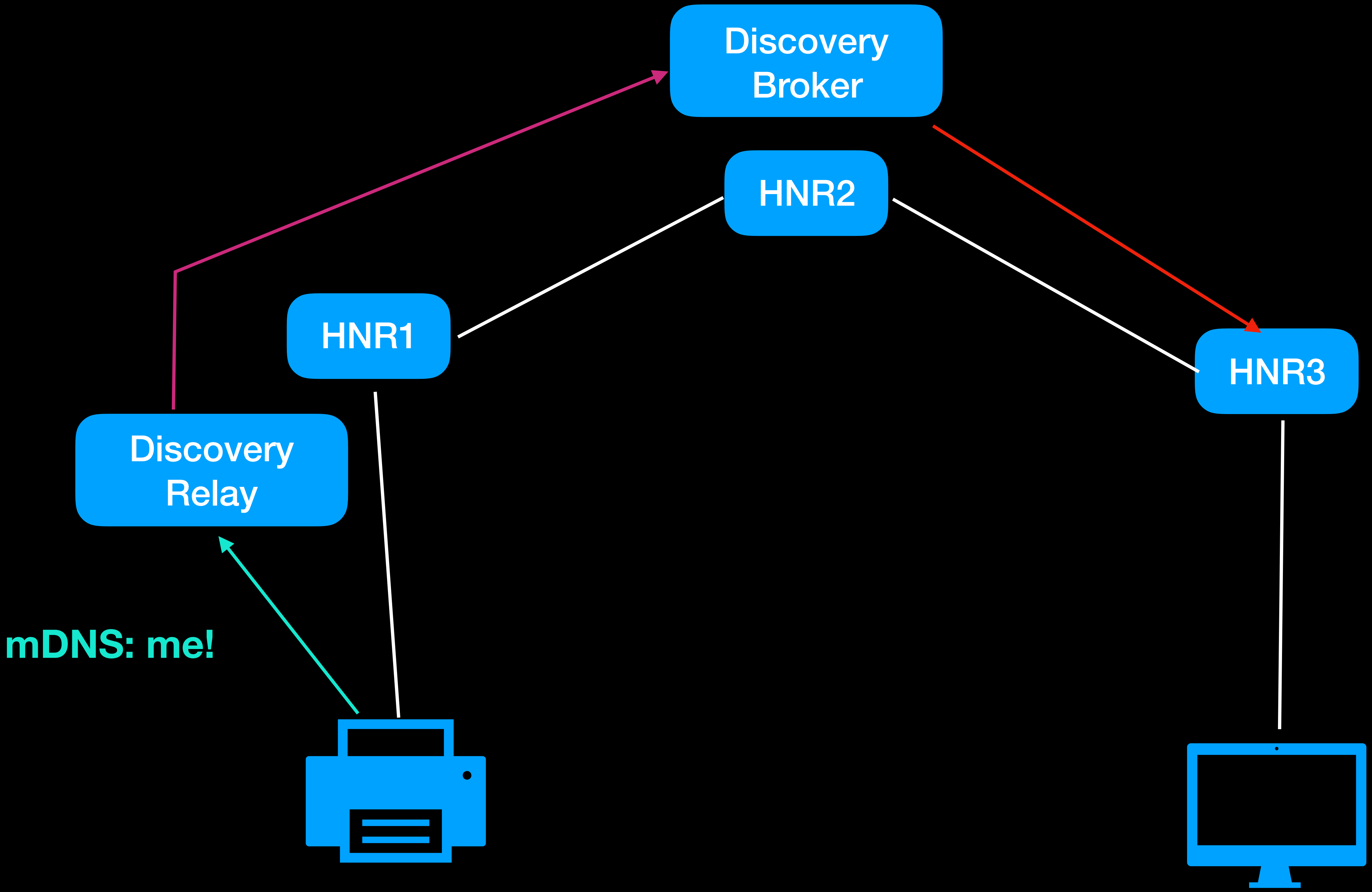
HNR2

HNR1

HNR3

Discovery Relay

mDNS: me!



Discovery Broker

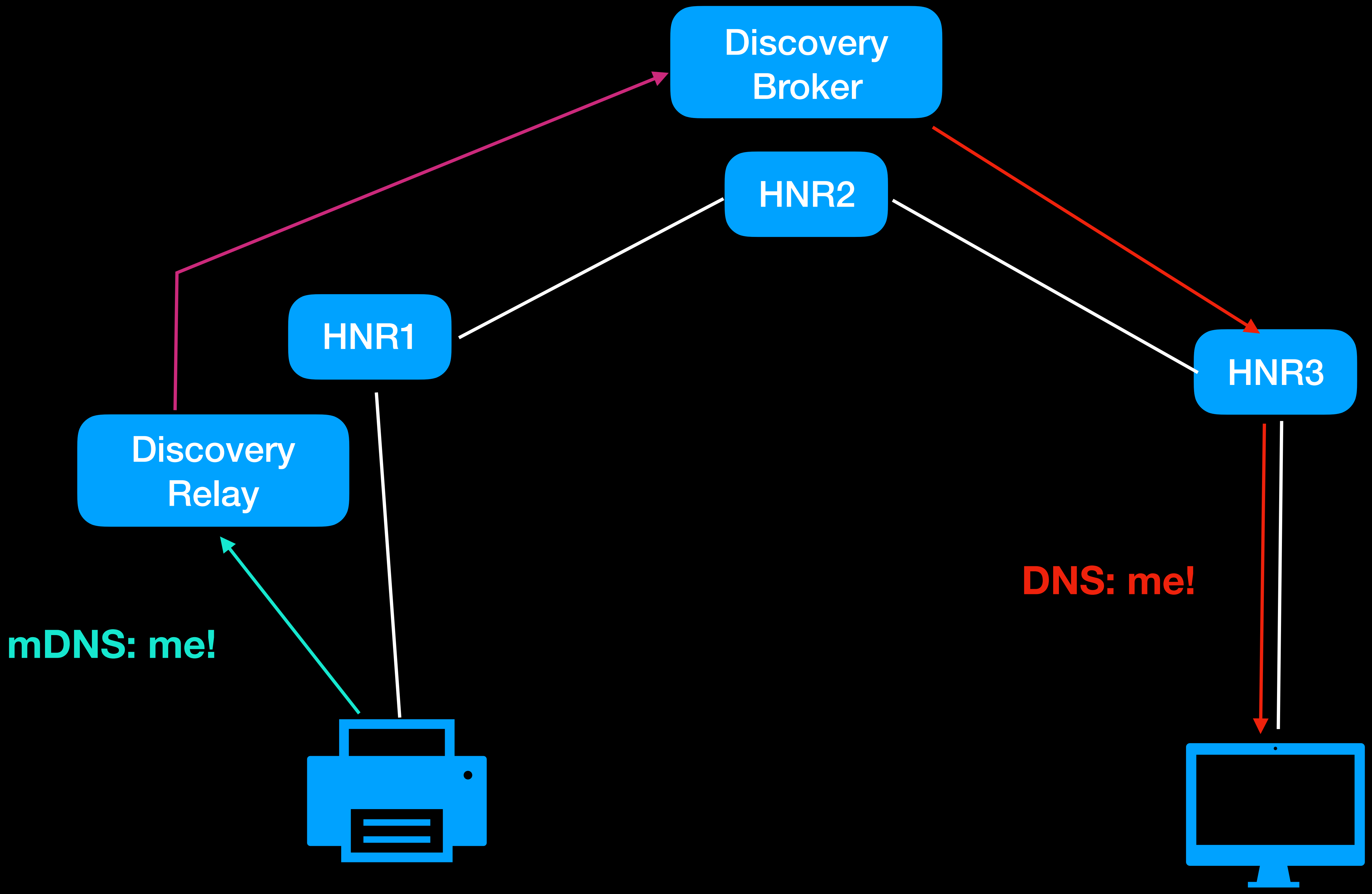
HNR2

HNR1

HNR3

Discovery Relay

mDNS: me!



Discovery Broker

HNR2

HNR1

HNR3

Discovery Relay

mDNS: me!

DNS: me!

DNSSD Security

- Could use DNSSEC for authentication
- But DNSSEC depends on a trust chain from the root and,
- Homenets have no registered domain name
- Options:
    - Register a domain name
    - Provide a special trust root that can be validated by host resolvers
    - No DNSSEC

HNCP/Babel security

- Head in the sand: we don’t need security, link security (WPA2) is enough, anything else is too complicated: do nothing
- Shared secret authentication, with secrets shared in the clear or protected using DTLS, keyed with (hand-wave)
- Lay the groundwork for a secure network now, figure out some of the details as we go along.

What would we need to secure the network?

- Each service provider (example: homenet router) generates a public/private key pair
- Public Key shared to all participants using HNCP, no encryption required because public, but no trust establishment mechanism either
- Now we can generate shared secrets between each router or sign data using public keys
- Public keys can then be used to authenticate DTLS or TLS connections between participants

What about trust?

- Sharing public keys gives us authentication of who holds the key, and encryption if we need it, but does not give us trust (authorization).

- This at least lets us identify a bad actor that’s harming the network and remove it, but it can always generate a new identity.

- We need a way to establish trust for devices that we authenticate with these keys

How might we establish trust?

- Print a key fingerprint on each device, have devices display their fingerprint in the UI along with their public key, have sysadmin compare printed fingerprint to UI display

- Hook devices together with wires (only works for devices with ethernet ports), push a button, and do trust establishment based on security of link plus user signal

- Leap-of-faith over WiFi based on user signal (assume that nobody bad is eavesdropping or MiTMing).

- Etc.   We plan to have another brainstorming session in Singapore (IETF 100).

Who is the sysadmin?

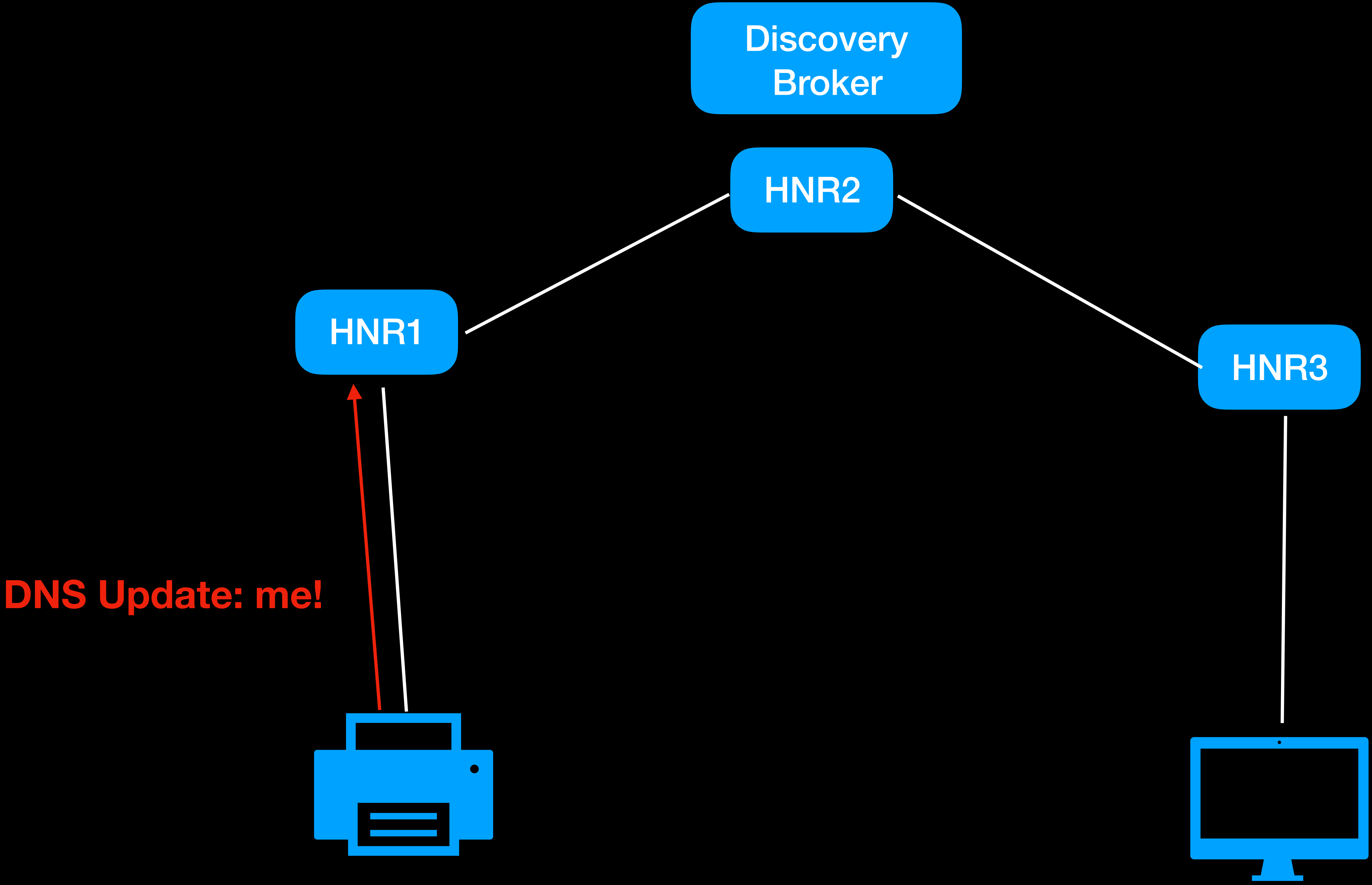
- Did I mention that the operator of this network has no idea what authentication and authorization are? And that the network is supposed to self-configure and self-manage?
- This makes establishing trust really hard
- Best current theory is that a web app running on user's phone with access to the camera could walk user through trust establishment process and display basic network status
- Alternative: ISP manages home network as a service (but how safe is this really)?

What about DNSSD trust establishment?

- A replacement for mDNS.
- Allows services to be discovered on multiple links
- Assumes that most devices will not know about DNSSD, and will just use mDNS
- Most service-providing devices (e.g., printers, set-top boxes, TVs) will not participate in HNCP
- Therefore if trust is to be established, will be done using DNS keys

DNSSD FCFS

- DNSSD service providers claim and protect names using DNS Update (RFC2136) in combination with DNS keys and SIG(0).
- Service publishes name + key, signs updates using SIG(0) with that key
- If the name isn’t taken, it’s claimed and assigned that key; otherwise service has to choose a new name
- Subsequent updates to that name must use the same key, or are rejected.
- Trust established as with mDNS: user chooses service, it works, they trust it.
- Better than mDNS, though: once trust is established, service can’t be spoofed
- DNS keys can also be used for TLS/DTLS if service supports that.



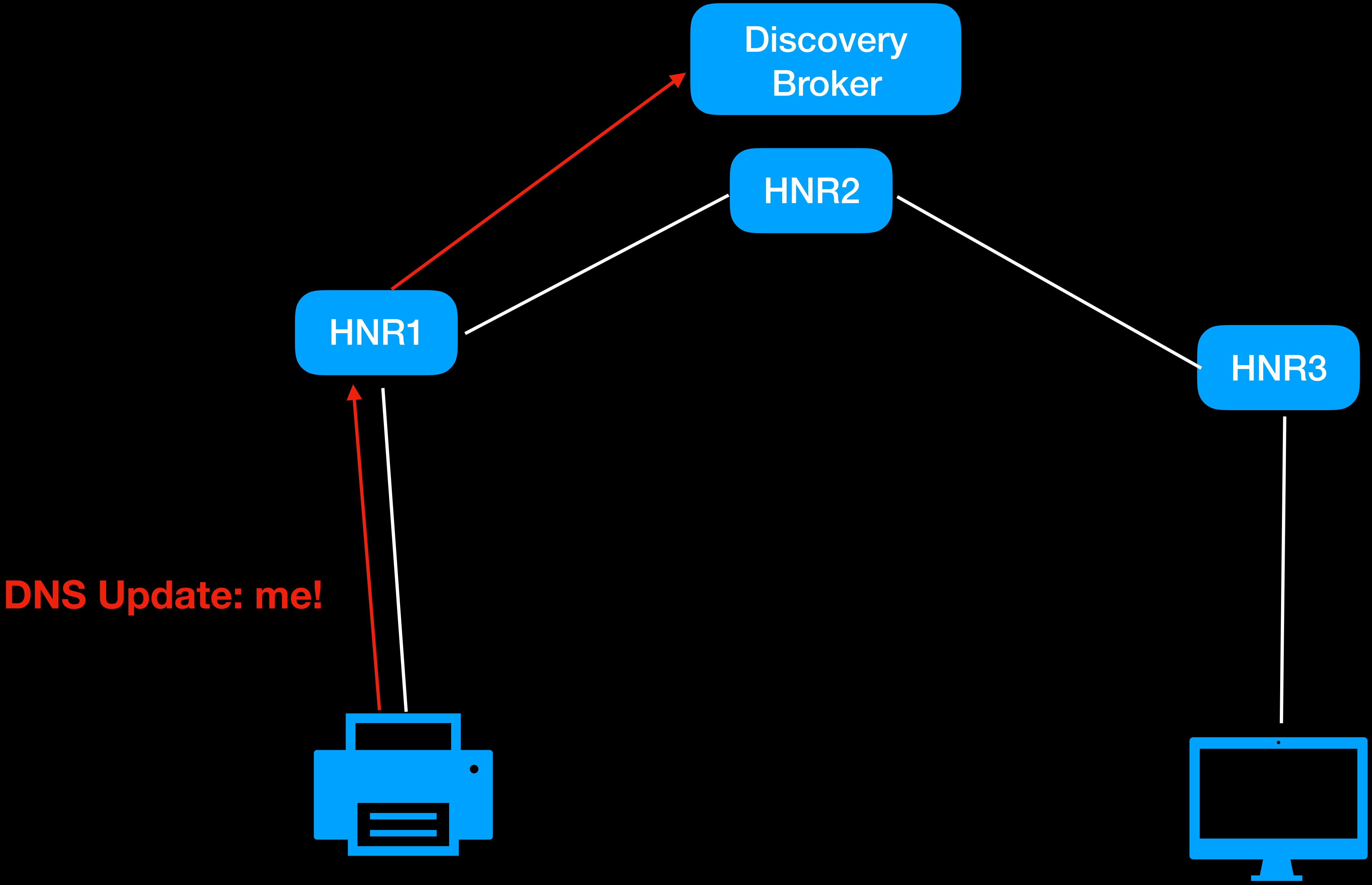
Discovery Broker

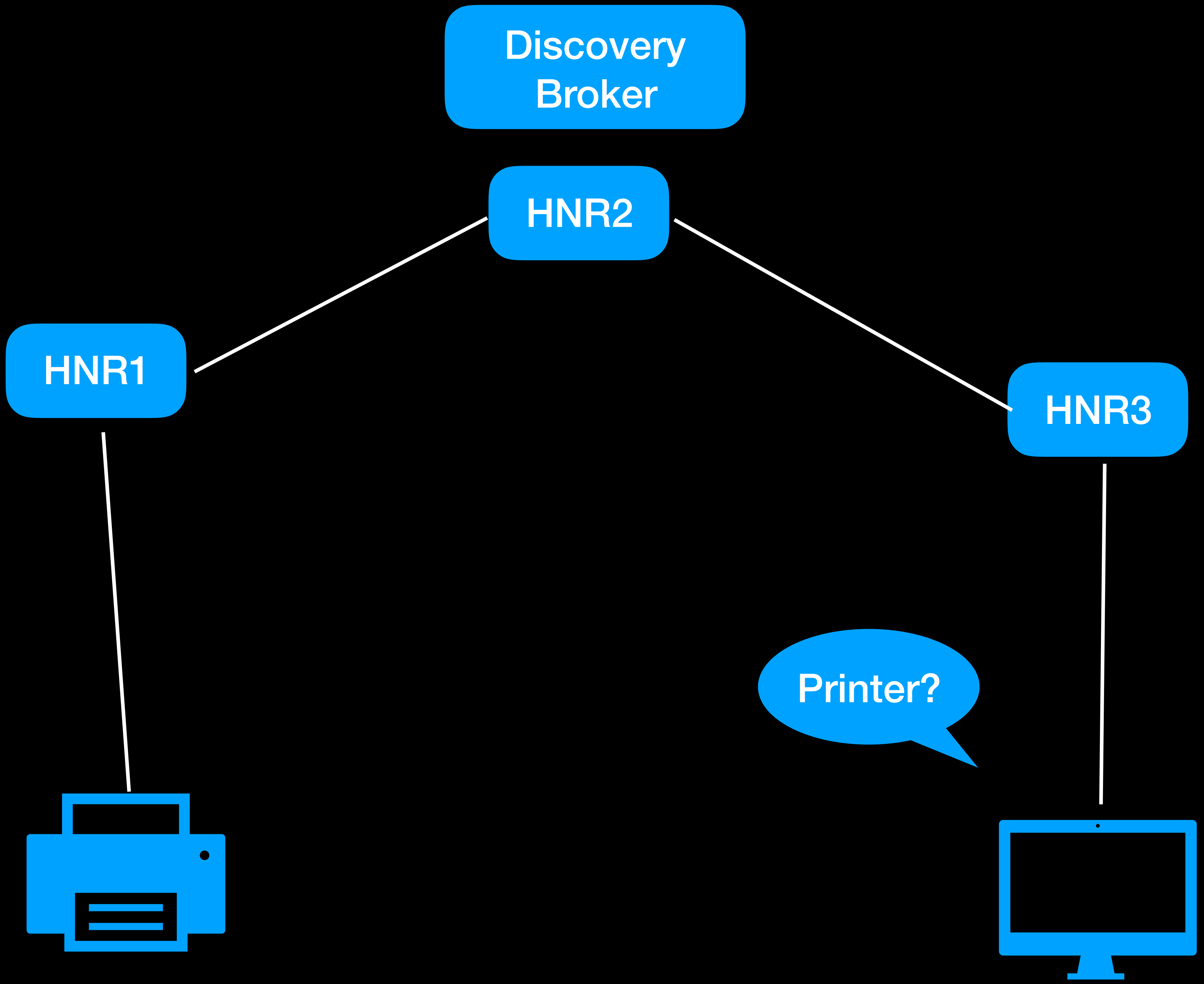
HNR2

HNR1

HNR3

DNS Update: me!


Discovery Broker
HNR2
HNR1
HNR3
DNS Update: me!



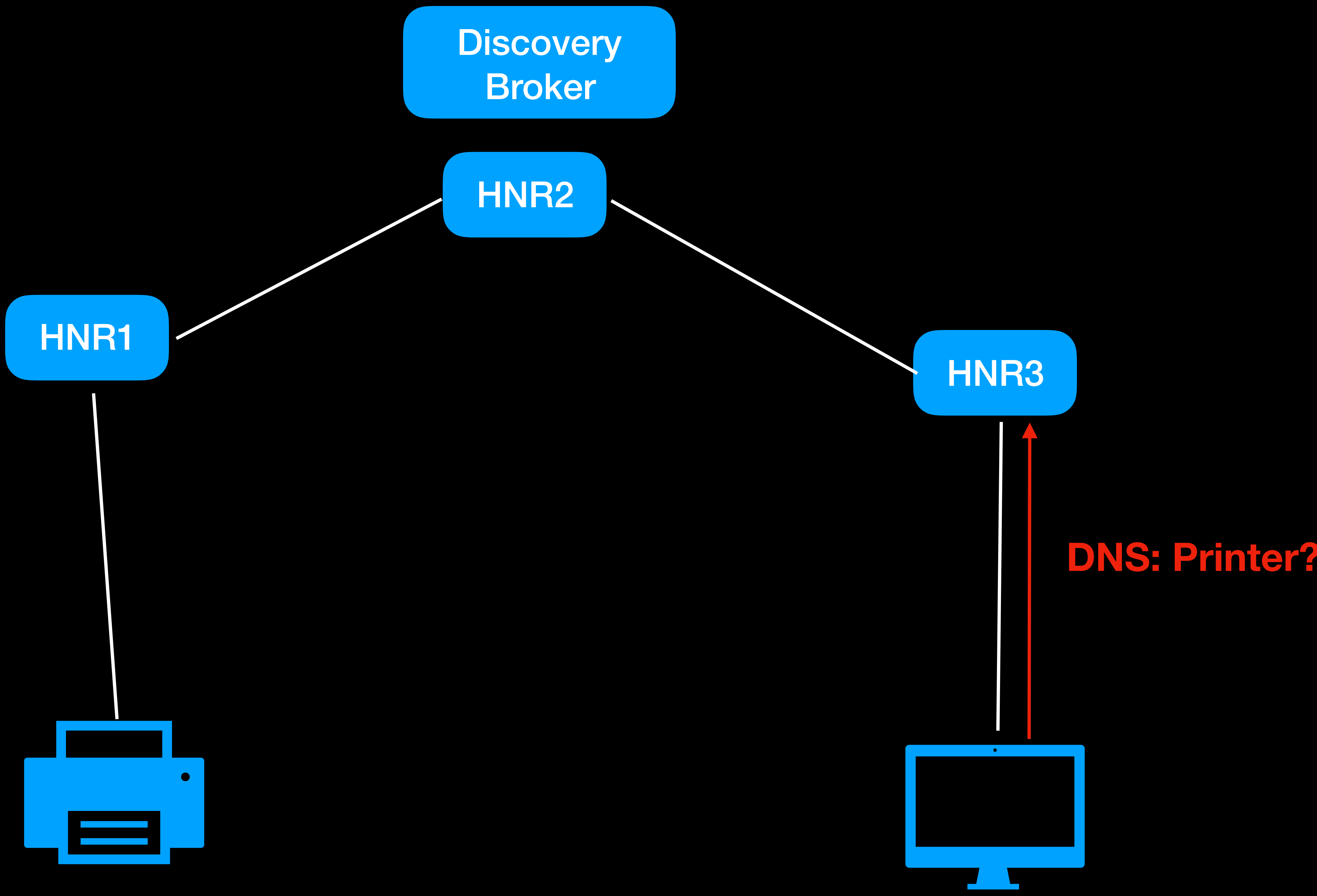
Discovery Broker

HNR2

HNR1

HNR3

Printer?



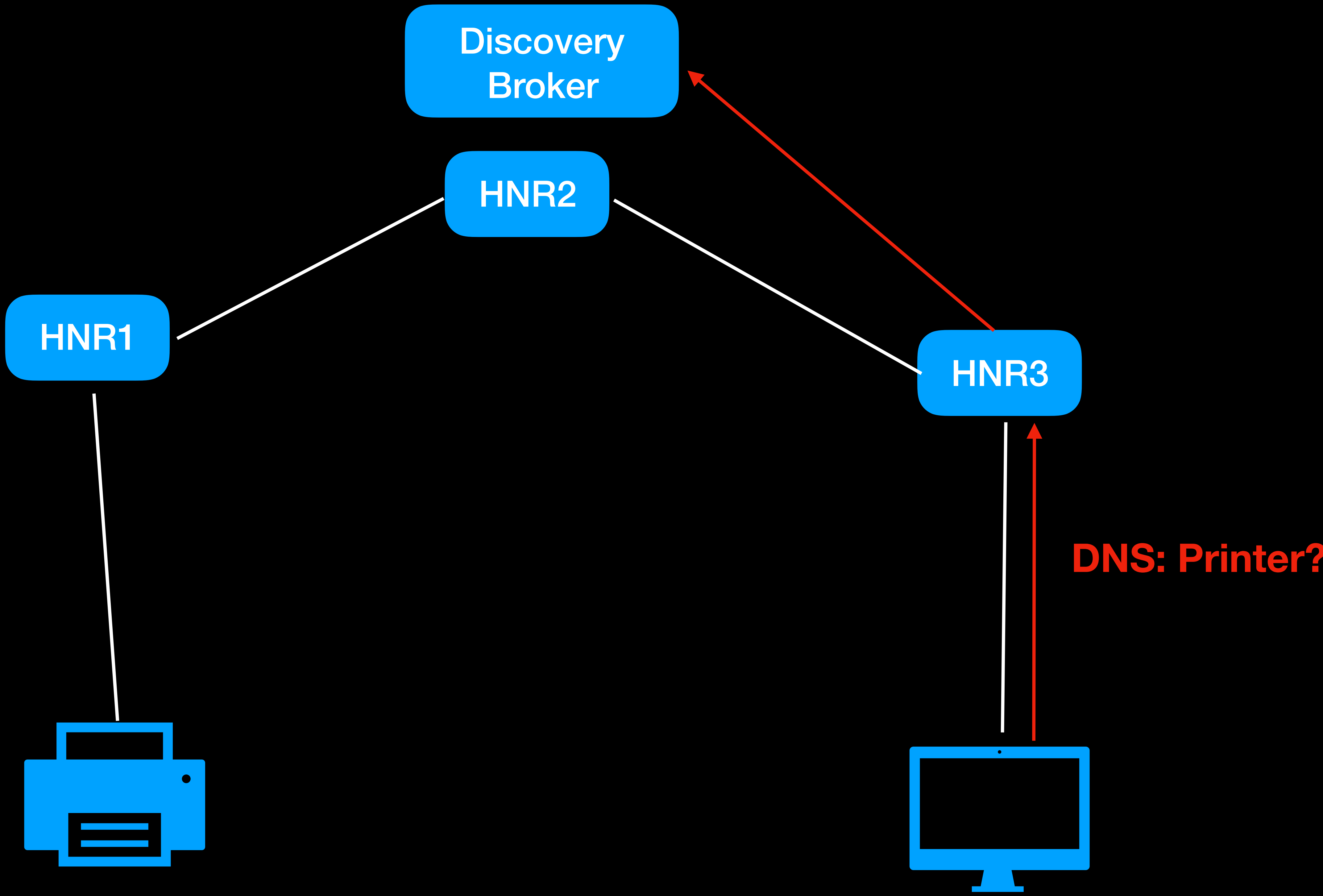
Discovery Broker

HNR2

HNR1

HNR3

DNS: Printer?



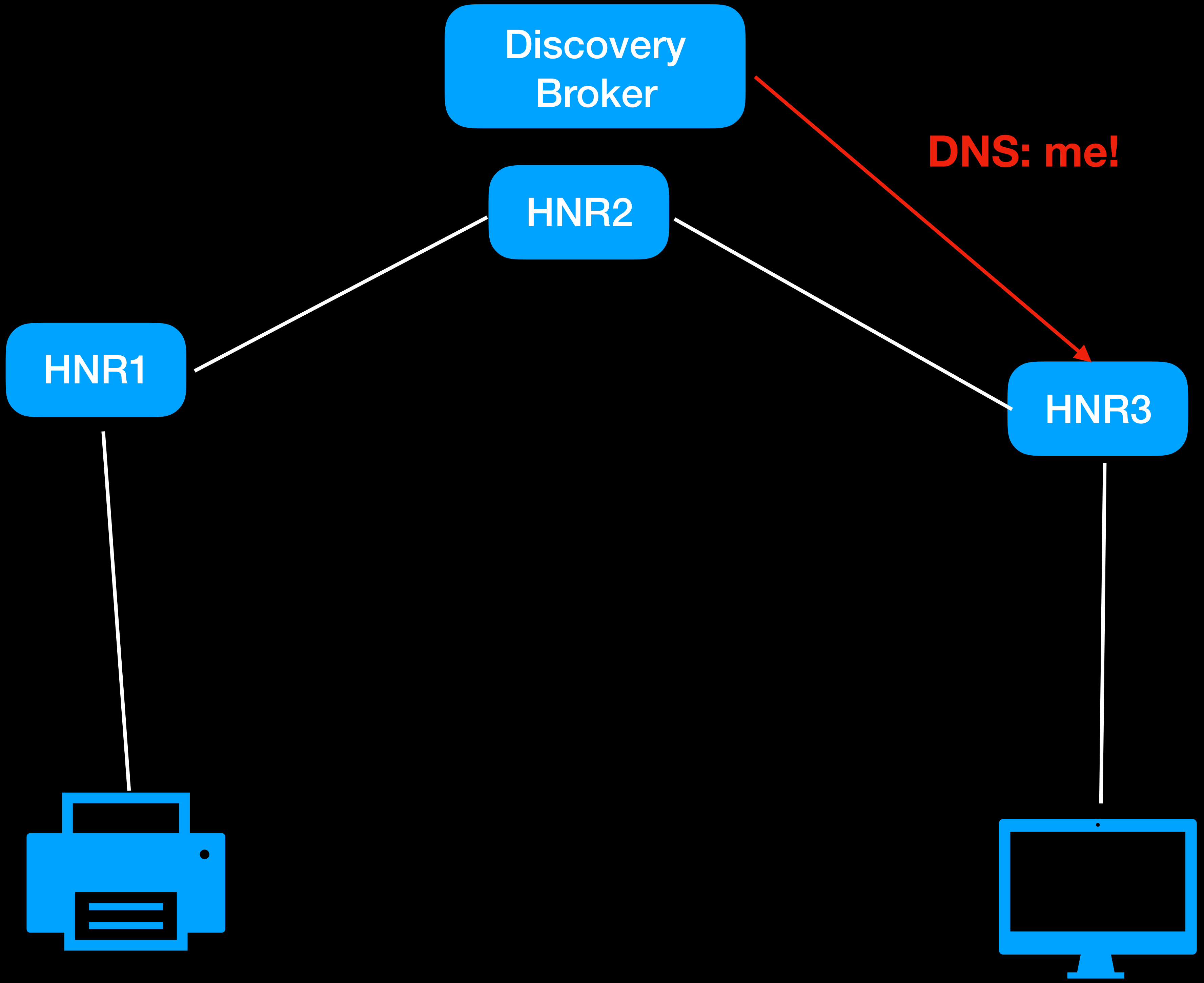
Discovery Broker

HNR2

HNR1

HNR3

DNS: Printer?



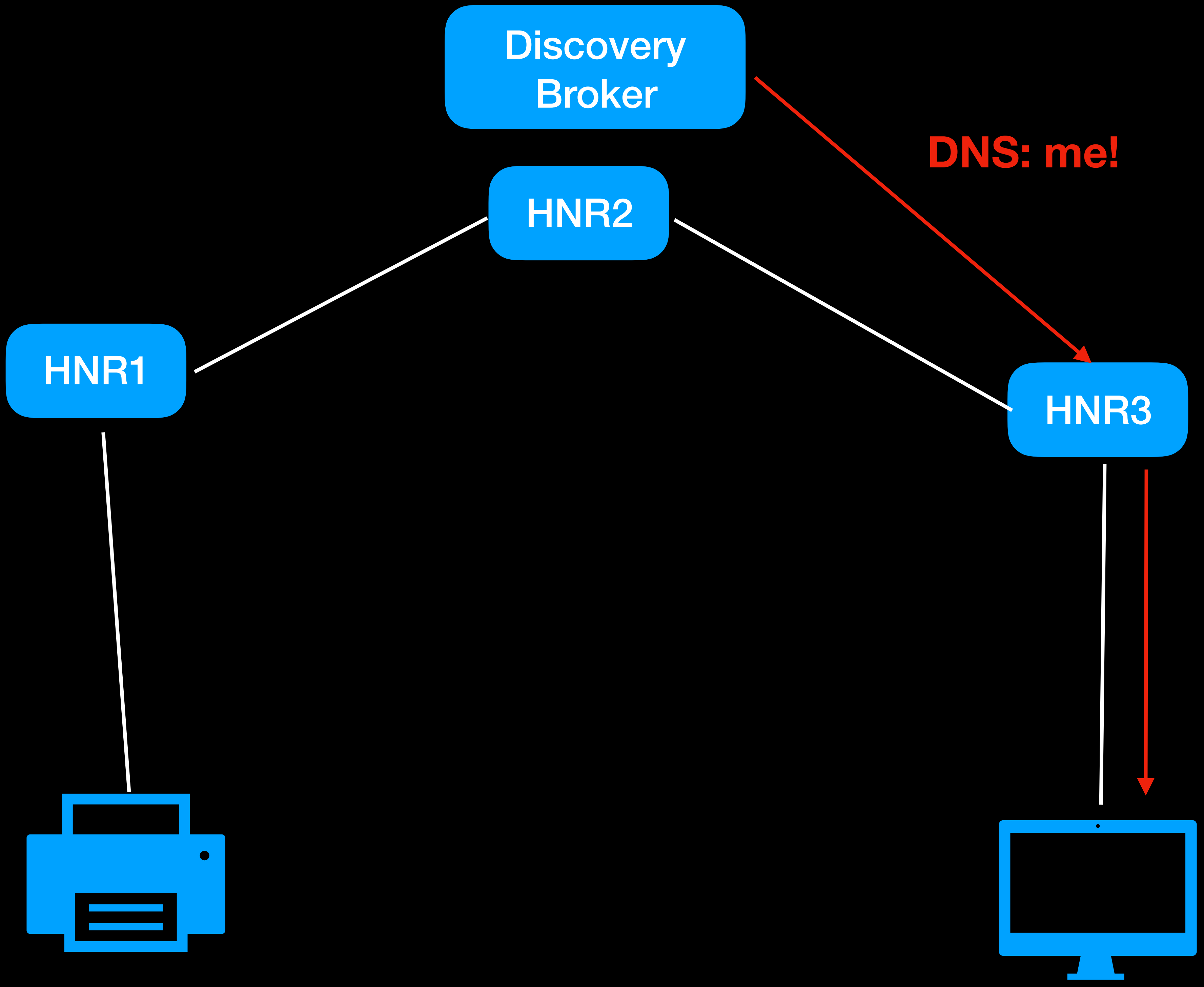
Discovery Broker

DNS: me!

HNR2

HNR1

HNR3


Discovery Broker
DNS: me!
HNR2
HNR1
HNR3

Web UI issues

- Most home routers have a web UI
- Web UI is over http, passwords in the clear
- Without a valid PKI cert, web UI will be seen as insecure by browser when submitting passwords, producing a warning.
- Don’t want to train user to click through warnings.
- We need a cert the browser will accept
- There is no way to get PKI certs, and browsers currently do not accept DNSSEC certs

Solutions

- Forbid web UI, replace with management API (which we would then have to specify in detail).
- Get browser vendors to support DNSSEC/DANE/TLSA as a way to secure TLS sessions
- Make it possible for home networks to get real domain names automatically, then use Let’s Encrypt/ACME to get PKI certs for browsers (chicken and egg problem, though).
- ???

Home networks are ephemeral

- Devices can be unplugged, factory reset, etc.
- Keys can be lost.
- Once trust is established, if devices remember keys, and then those keys are lost, how do we re-establish trust?
- Phone app serves as key store?
- Master key and key revocation protocol?
- This is an open issue—the working group has not yet gamed this out, but it needs to be addressed.

This is a really hard problem

- This is why some homenet people simply throw their hands up and say "why bother?"
- If we did everything we’ve currently envisioned, we still would have gaps.
- But if we do nothing, we’ll have nothing *but* gaps.

Current plan

- Put as many security building blocks in place as possible
- Try to clearly understand how to use them when network is completely unmanaged, sort-of managed and professionally managed
- Try to identify gaps and think about how to address them
- Try not to miss any opportunity to secure one of the protocols used in homenet, but don’t think that encryption=security or authentication=security without considering how trust is established
- Remember that perfect is the enemy of good enough.

References

- Homenet Naming Architecture (draft-tldm-simple-homenet-naming-01)
- RFC 6762 (Multicast DNS)
- RFC 6763 (DNS Service Discovery)
- DNSSD Hybrid Proxy (draft-ietf-dnssd-hybrid)
- RFC 7788 (HNCP)
- Service Registration Protocol for DNS-Based Service Discovery (draft-sctl-service-registration-00)
- Babel (draft-ietf-homenet-babel-profile)
- Multiple Provisioning Domains (RFC 7556)