Monitoring with

Koen de Jonge

About me

- *Koen de Jonge*
- *Using Linux and Apache since 1994*
- *Using Nagios since 1999*
- *Using and contributing to WebGUI since 2001*
- *Co-founder of ProcoliX (PlainBlack hosting partner)*

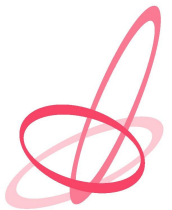
Overview

Presentation based on Ethan Galstad's presentation at FOSDEM 2005

- **Presentation (40 minutes)**
  - Background and overview
  - In-depth information on specific topics
  - Do's and dont's
- **Question & Answer (10 Minutes)**

What is Nagios?

- Open Source (GPL) network monitoring application
- Runs on Linux, FreeBSD, Solaris, etc.
- Monitors hosts and services on your network
- Provides you with an overview of the status of your systems
- Notifies you know when things are go wrong
- Allows you to resolve problems faster
- Provides availability reports for SLAs, etc

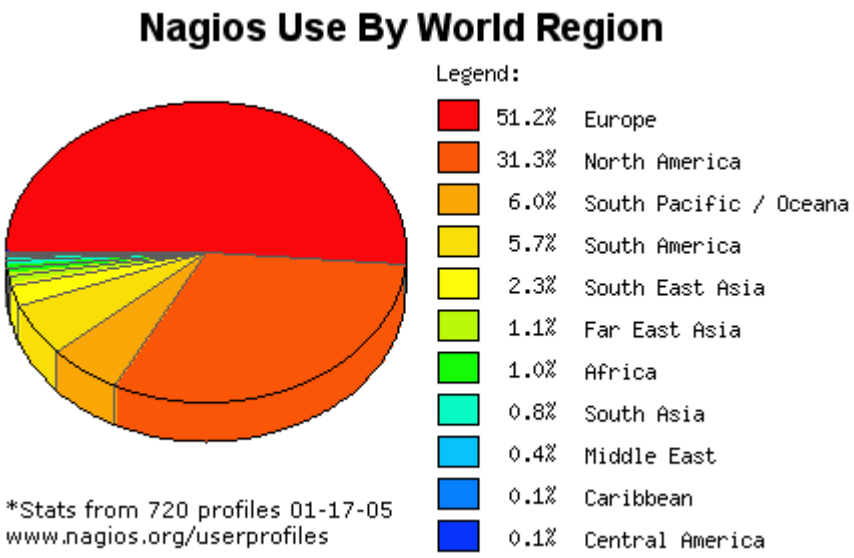


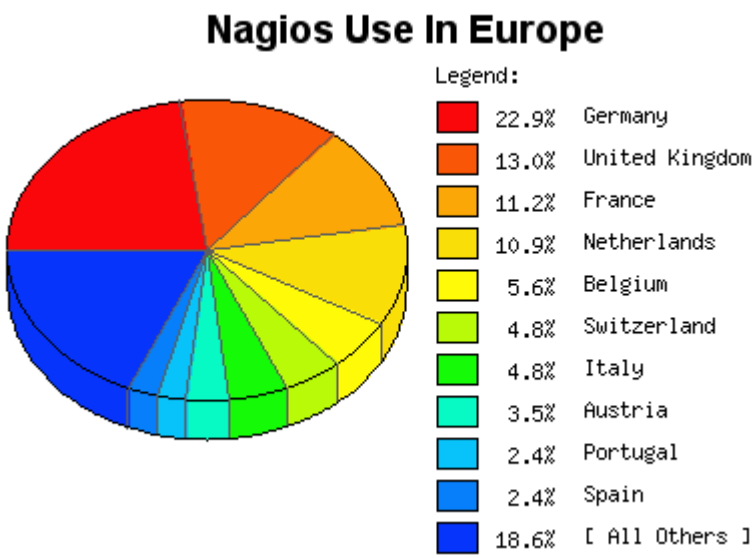
How it started

- Development started in January 1999
- Wanted to provide monitoring services to local businesses
- No suitable (cost- and feature-wise) commercial or free solution was available
- Originally designed to monitor small LAN (~20 servers)
- Initially anticipated a small user base (~20 users)

Where it is now

- Six years and nine major versions later
- Top installations monitor thousands of hosts and services
- Over 150,000 downloads of latest stable version (1.2)





The name

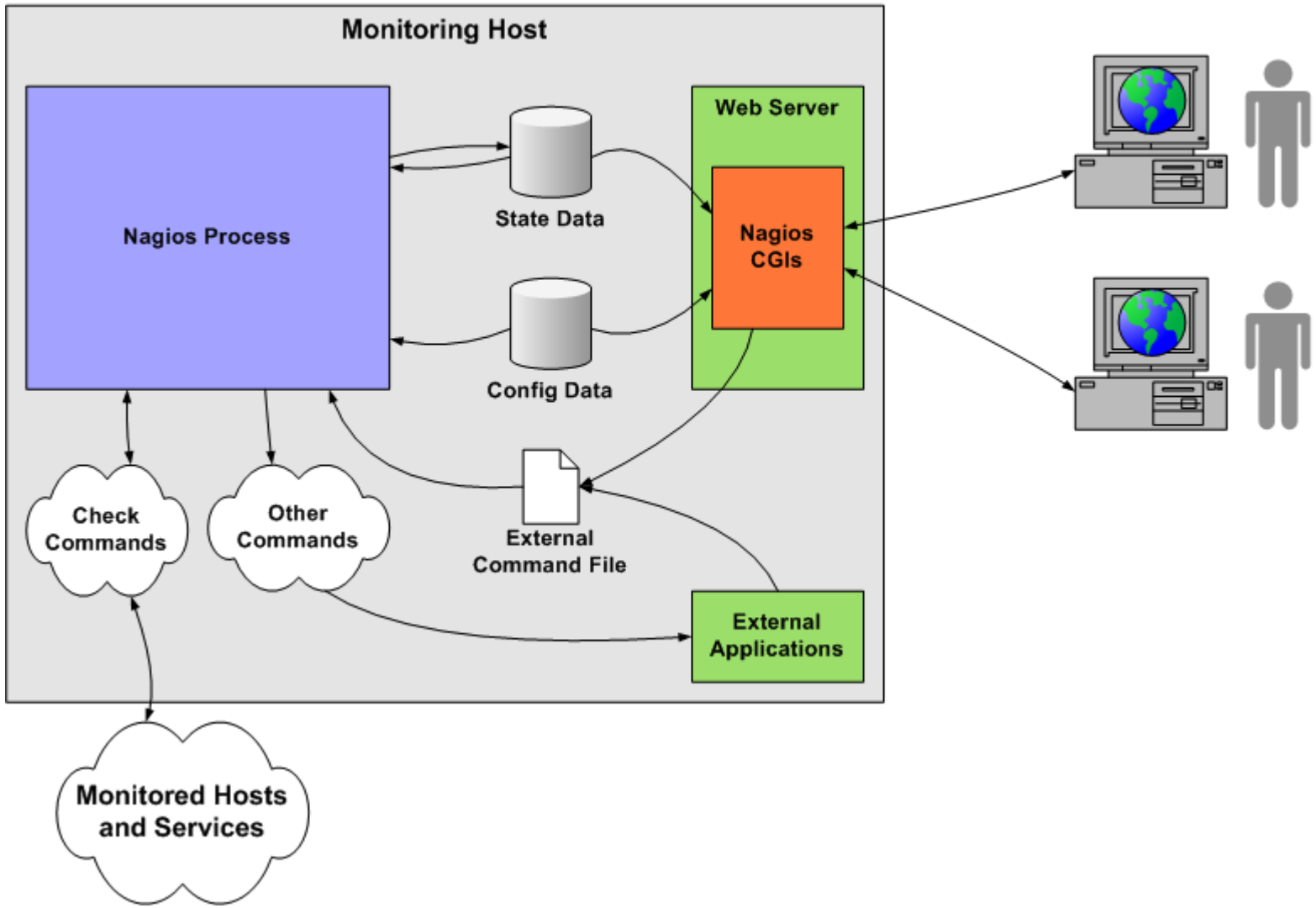
- Originally released under the name “NetSaint” (netsaint.org)
- Name changed to “Nagios” in 2001 for trademark reasons
- N.A.G.I.O.S. = “Nagios Ain't Gonna Insist On Sainthood”
- Trademark registration for Nagios®
- I pronouce it as “nah-ghee-ose”
- Correct pronunciation: however you'd like

Nagios Overview

Design Overview

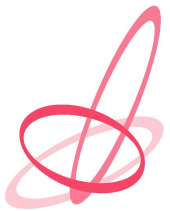
- Designed in a modular fashion
- Daemon contains the monitoring logic and coordinates things
- CGIs allow users to view status information through web browser
- External applications handle the low-level monitoring work
- External commands can be triggered to manage alerts, state changes, and monitoring information
- Can be integrated with 3rd-party applications fairly easily

Nagios Overview



Nagios Overview

- **What can be monitored?**
  - Servers, workstations, printers, routers,etc.
  - More genericly, anything that:
    - has or is associated with an address of some kind
    - is “reachable”
  - Nagios doesn't know or care about network protocols or addresses
  - Not limited to monitoring standard network equipment and services
- **Monitoring subjects/targets**
  - Hosts
  - Services

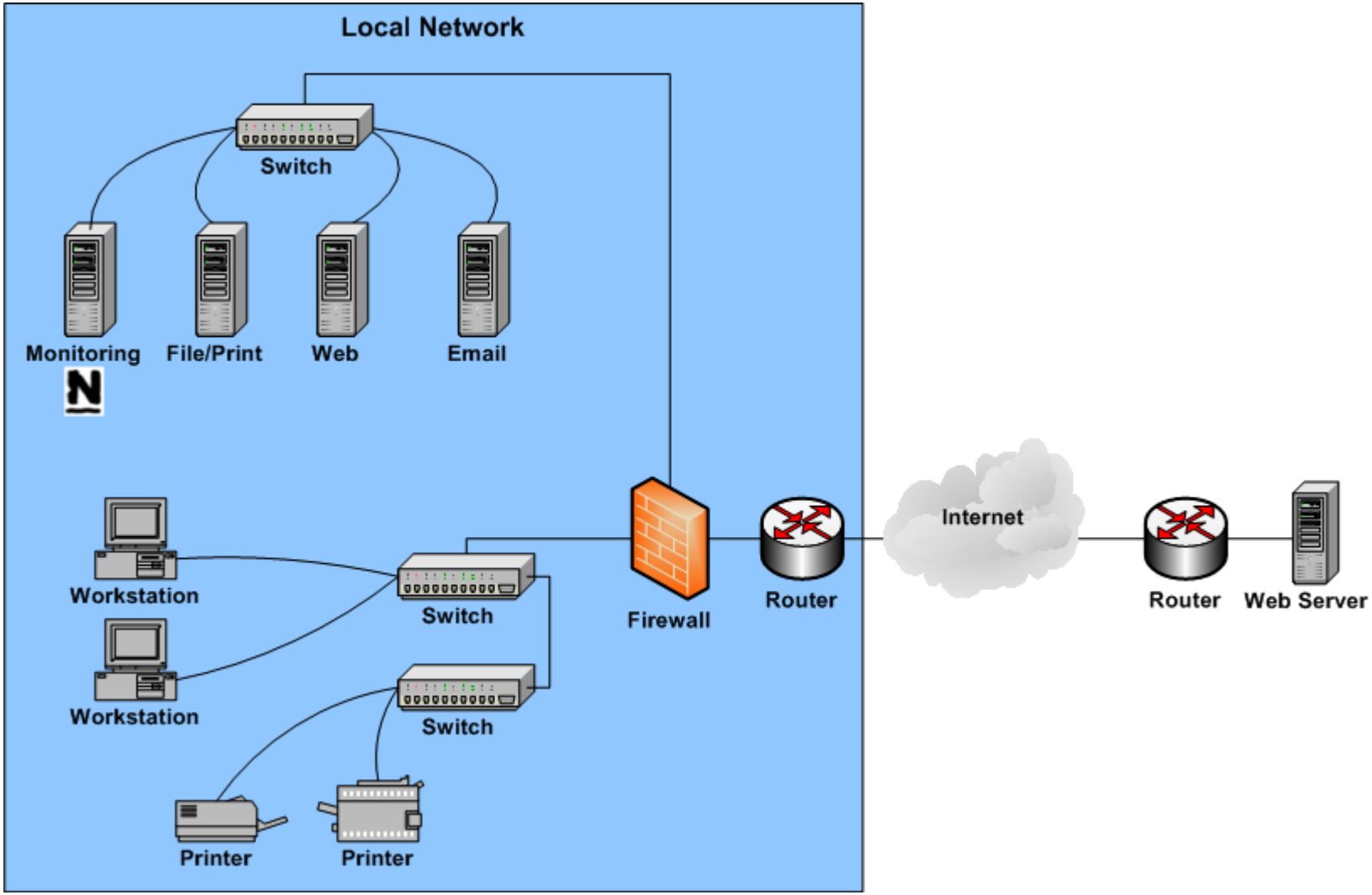


Nagios Overview

- **Hosts**
  - Usually physical objects (server, switches, routers, printers, etc.)
  - Can have parent/child relationships with other hosts
  - Provide one or more services
- **Services**
  - Things associated with or provided by a host
  - Tangible services (e.g. disk usage, printer toner supply)
  - Intangible services (e.g. HTTP, SMTP, IMAP, POP3, DNS)

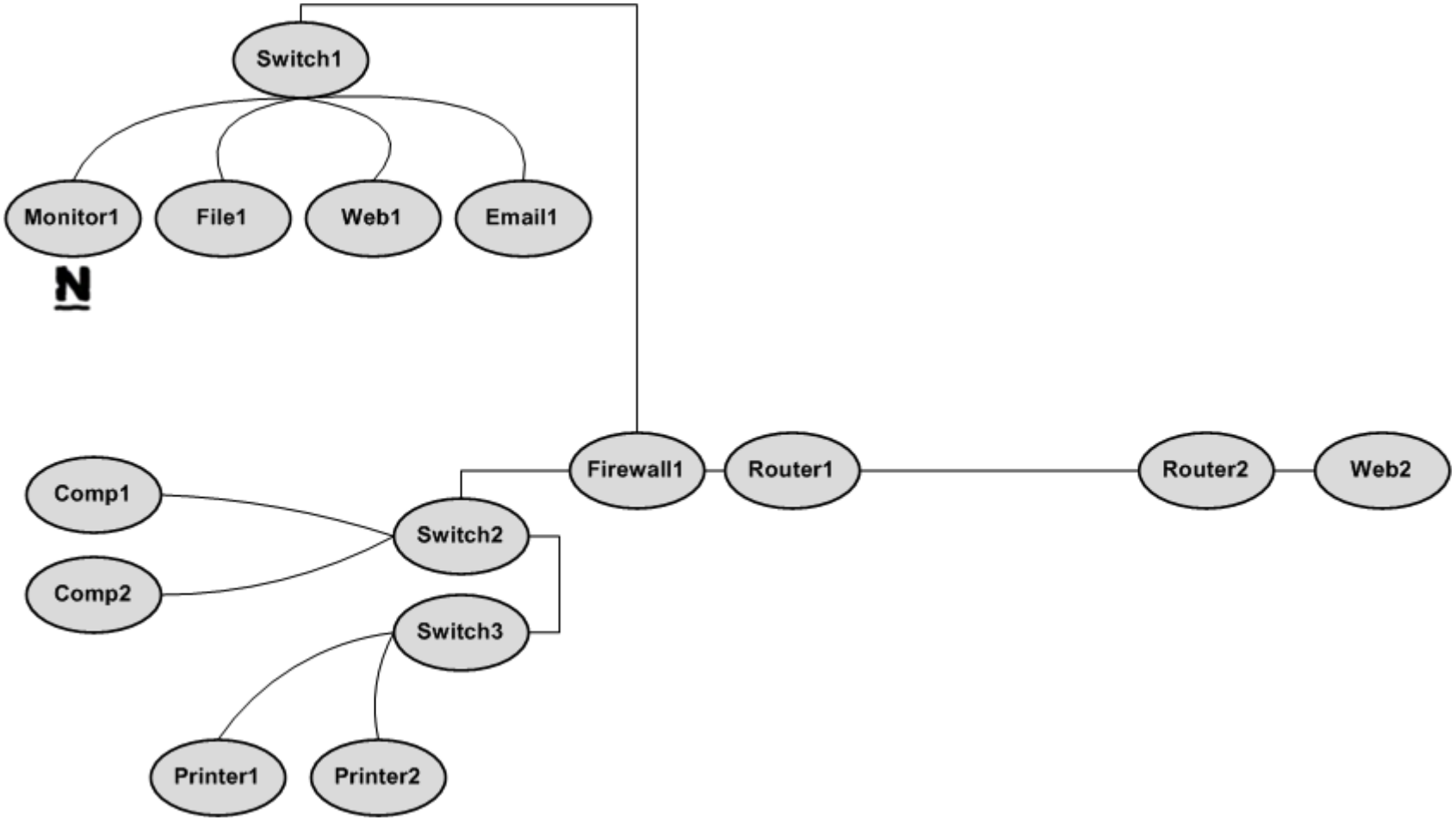
Nagios Overview

- *Example: Small Network Layout*



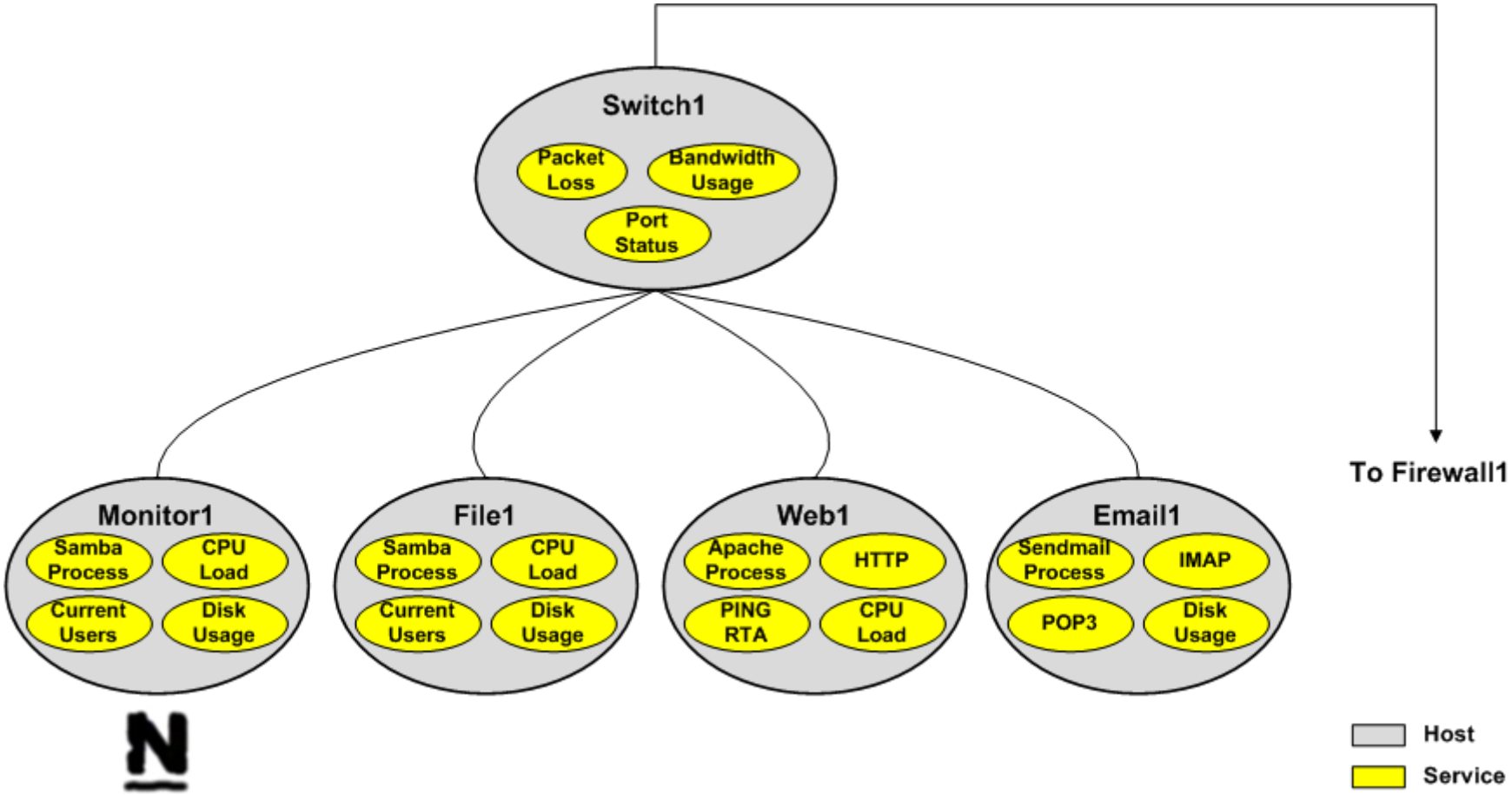
Nagios Overview

- *Hosts as viewed by Nagios*



Nagios Overview

- Services as viewed by Nagios

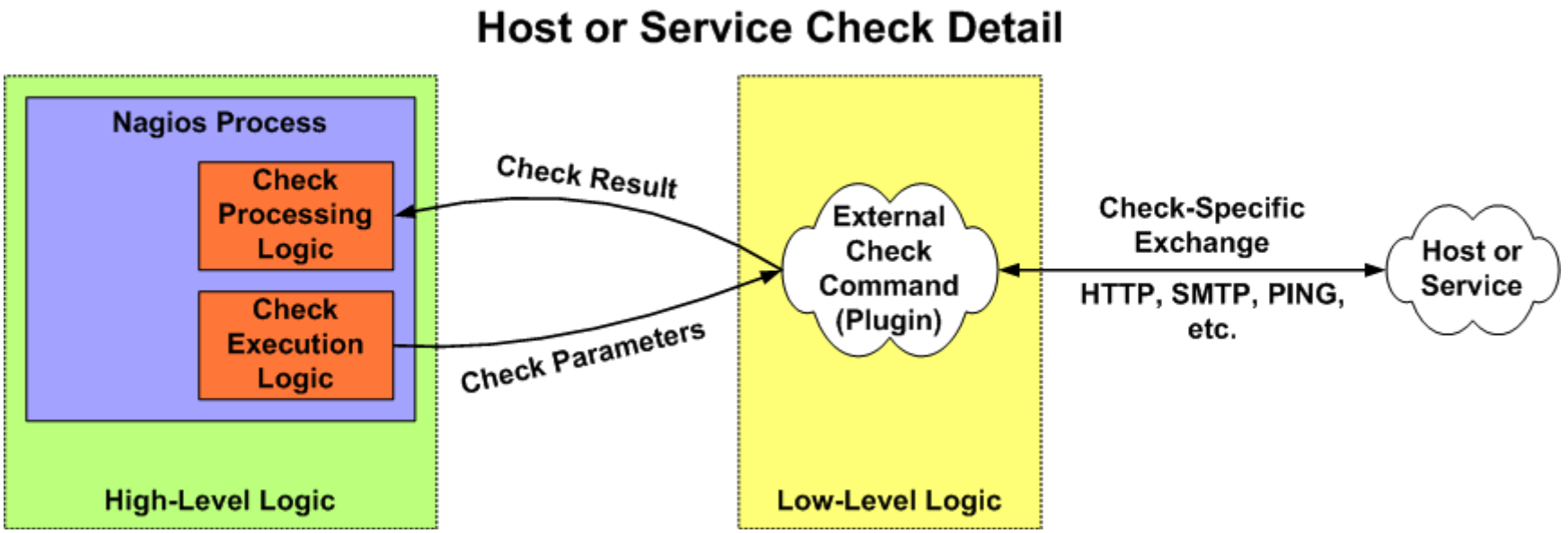


Nagios Plugins

Nagios Plugins

- ***How are hosts and services monitored?***
  - *Nagios doesn't understand network addresses, protocols, or services*
  - *Nagios passes information about what needs to be checked to external commands (plugins)*
  - *Plugins perform the actual checks of hosts and services and return information back to Nagios*
- ***What are plugins?***
  - *Shell or perl scripts, executables, etc.*
  - *Understand network addresses, protocols, services, etc.*

Nagios Plugins

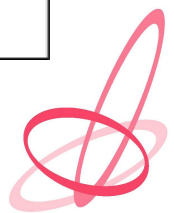


Nagios Plugins

- ***What information do plugins return?***
- *Text output*
  - *Shown to users via the web interface, in notifications, etc.*
- *Return code*
  - *This is what Nagios uses to determine the status of things*
  - *Four possible return codes used to indicate different states*
    - 0 = OK
    - 1 = WARNING
    - 2 = CRITICAL
    - 3 = UNKNOWN

Example Plugin Output:

```
[nagios@monitor ~]# /usr/local/nagios/libexec/check_ping -H www.google.com -w 40,40% -c 100,80%
PING WARNING - Packet loss = 0%, RTA = 48.31 ms
[nagios@monitor ~]# echo $?
1
[nagios@monitor ~]#
```

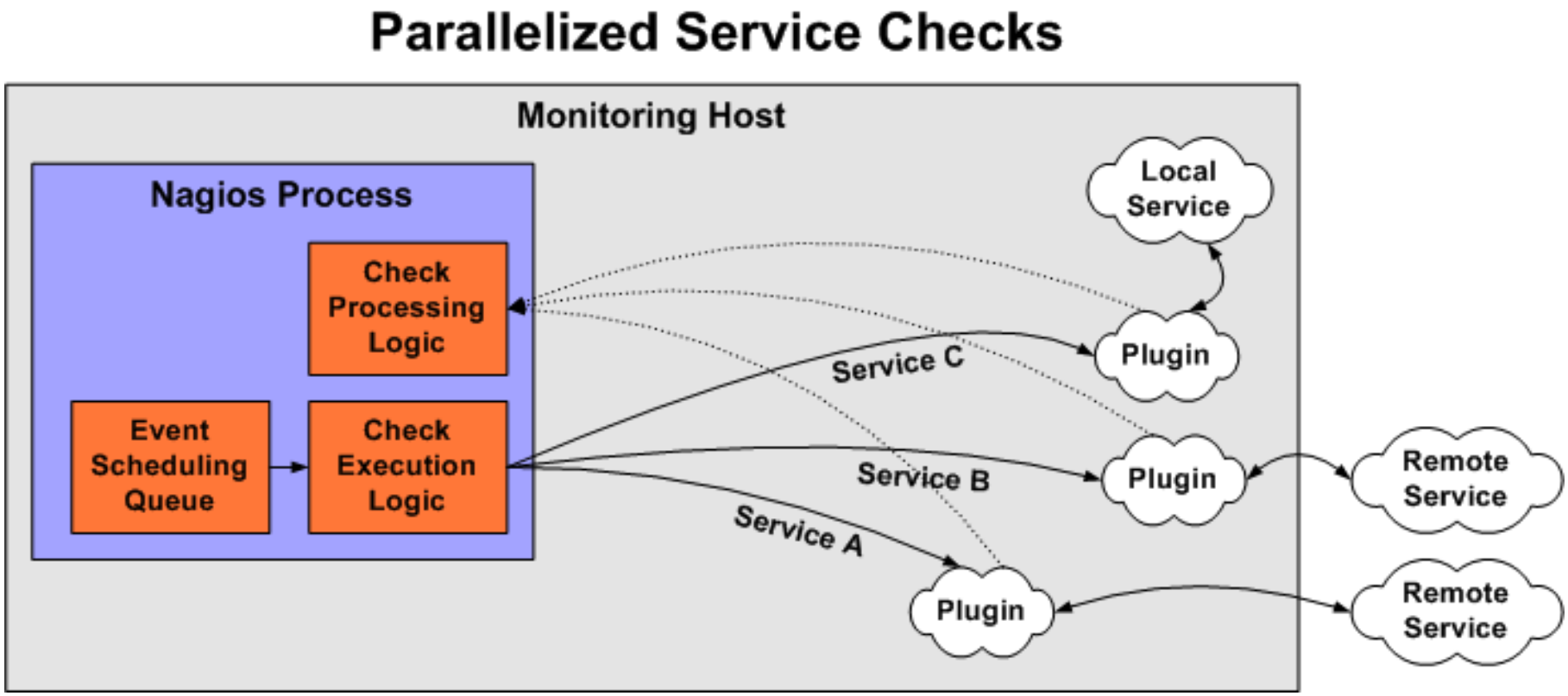
Service Checks

Service Checks

- ***Services***
  - *Services are the main reason we monitor things*
  - *If a host offers no useful services, why would we monitor it?*

- ***Service checks***
  - *Checks are performed at regularly scheduled intervals using plugins*
  - *Multiple service checks can be run in parallel*
  - *Four possible states: OK, WARNING, CRITICAL, and UNKNOWN*
  - *You can monitor anything you can write a plugin for:*
    - Simple: HTTP, IMAP, DNS, disk usage, etc.
    - More complicated: Web transactions, mail loop detection, etc

Service Checks



Host Checks

Host Checks

- ***Hosts***
  - *"Containers" for services*
- ***Host Checks***
  - *As with services, checks are performed using plugins*
  - *Checks are performed on-demand after service state changes*
    - Critical decision point: is the host or the service the real problem?
    - Need timely and accurate information on host status
  - *Three possible states: UP, DOWN, and UNREACHABLE*
  - *Can trigger a route verification*

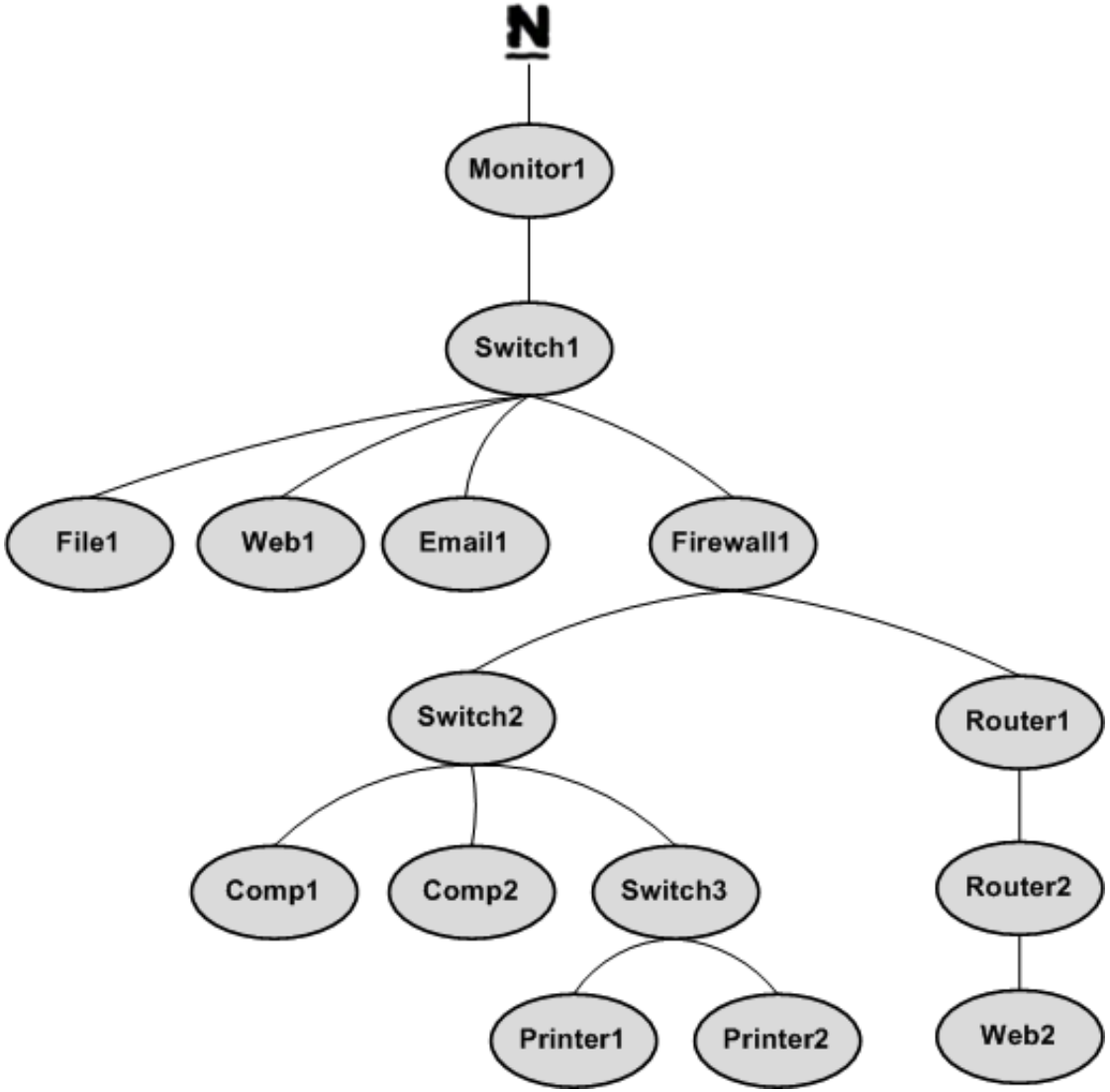
Route Verification

Host Route Verification

- ***Why might a host not be UP?***
  - *The host is DOWN*
  - *The route to the host is blocked by one or more other hosts (UNREACHABLE)*
- ***Route verification***
  - *Determines whether hosts are DOWN or UNREACHABLE*
  - *Can be very time-intensive if network problems are widespread*
- ***Why is it useful?***
  - *Helpful in determining the "real" cause of widespread problems*
  - *Different host states may be acted upon differently by processing and notification logic*

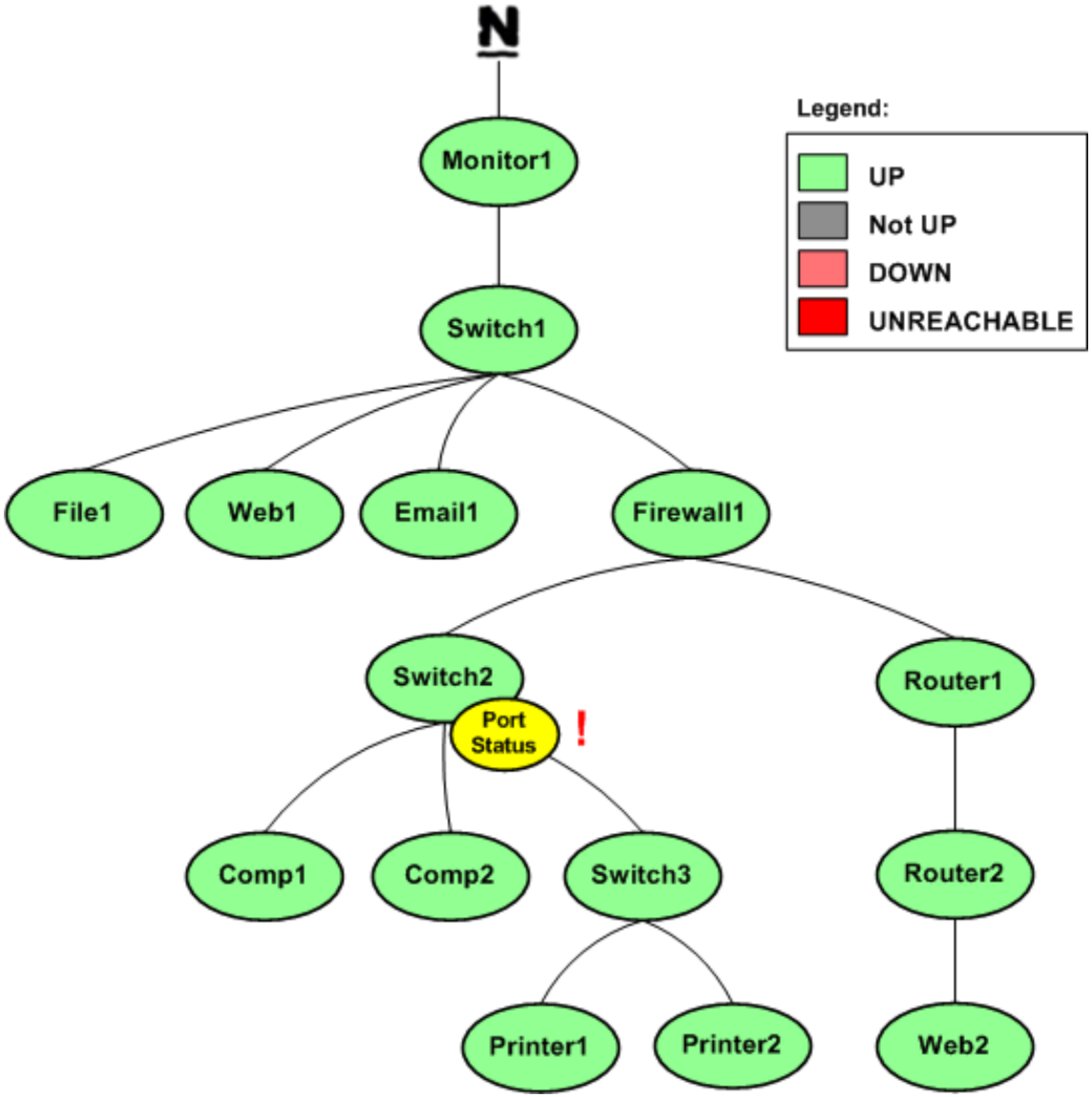
Host Route Verification

- *Logical host relationships - The world according to Nagios*



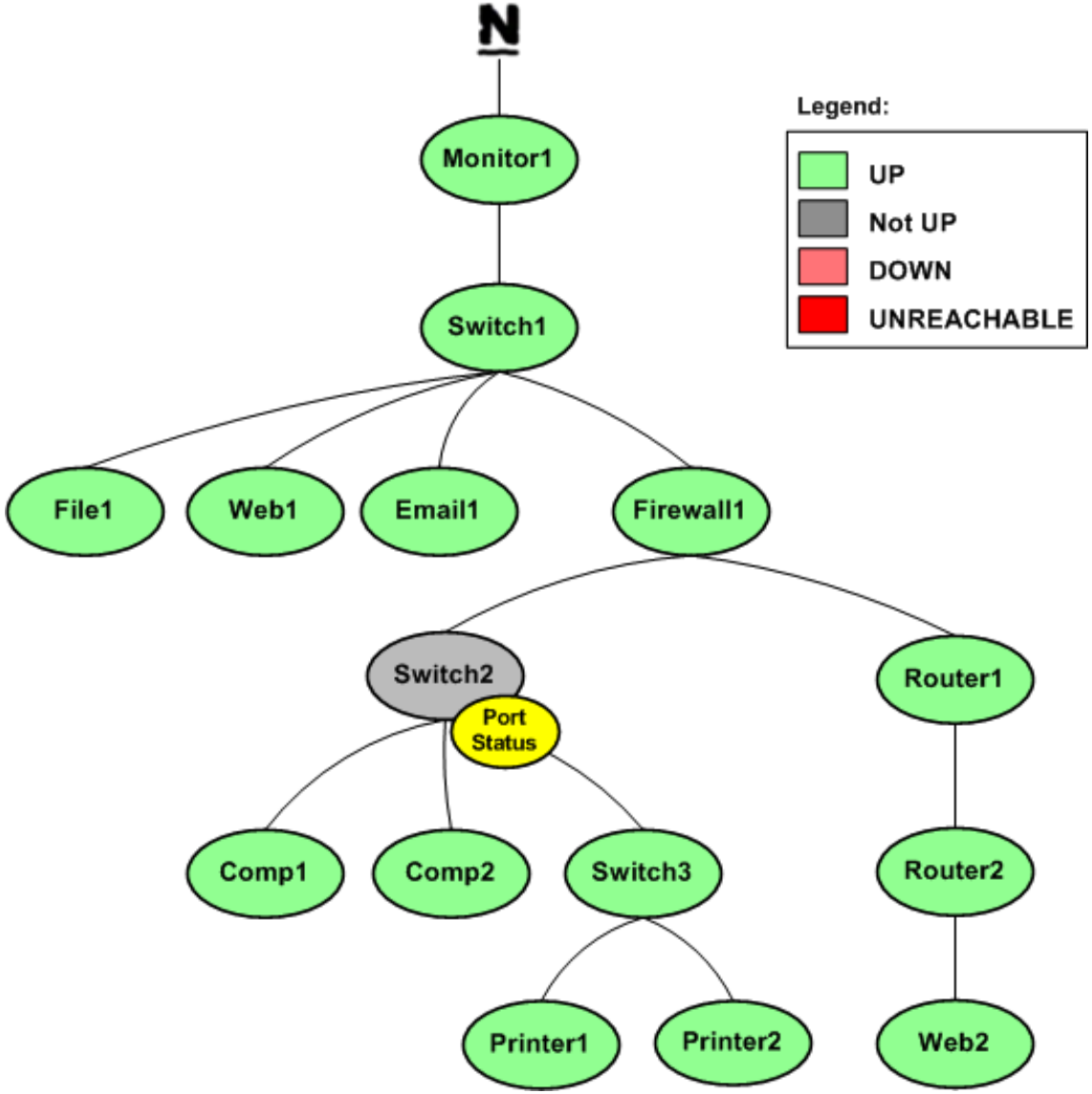
Host Route Verification

- *A problem is detected with 'Port Status' service on Switch2.*



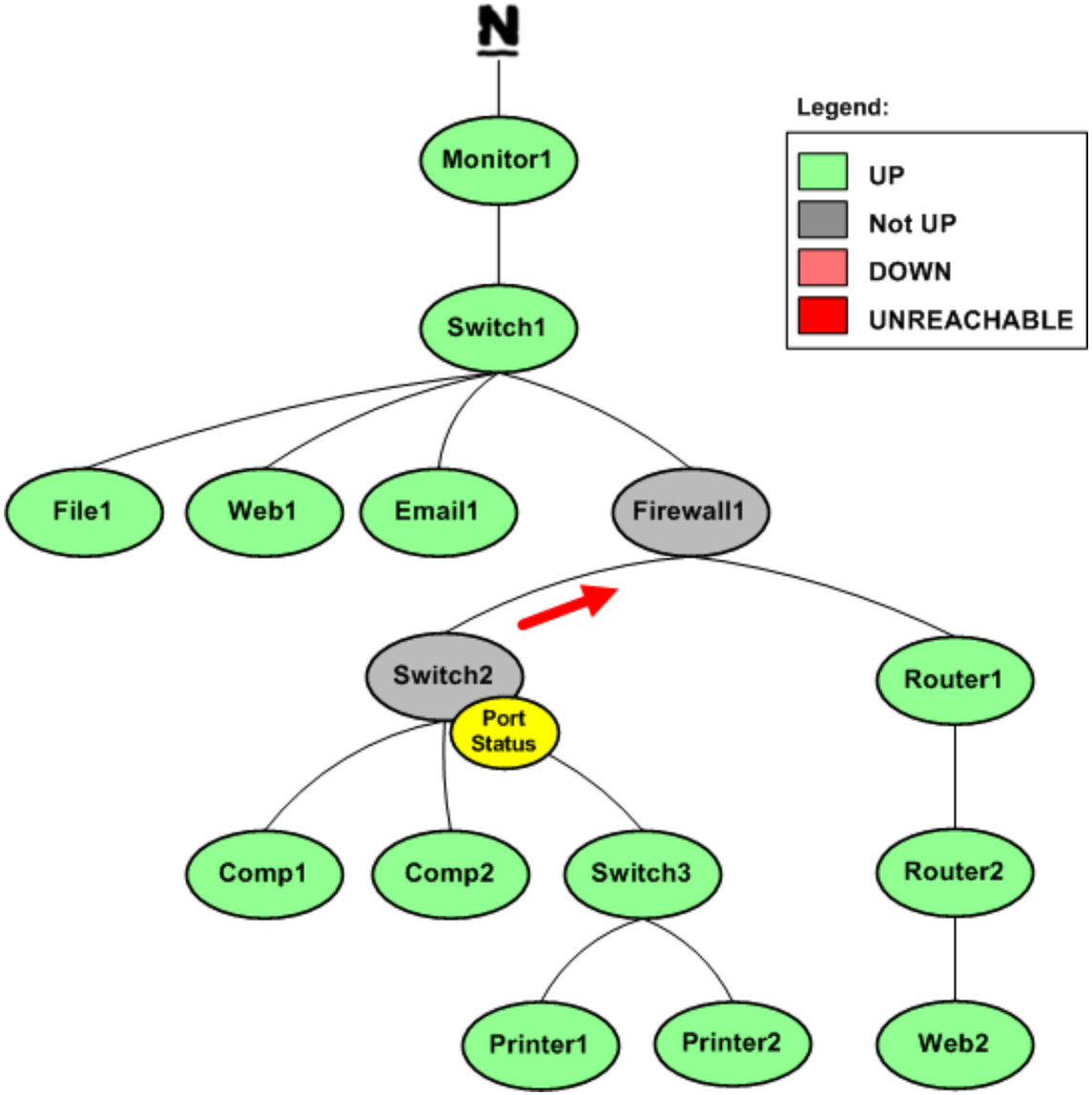
Host Route Verification

- *Switch2 is checked for problems and found to NOT be UP…*



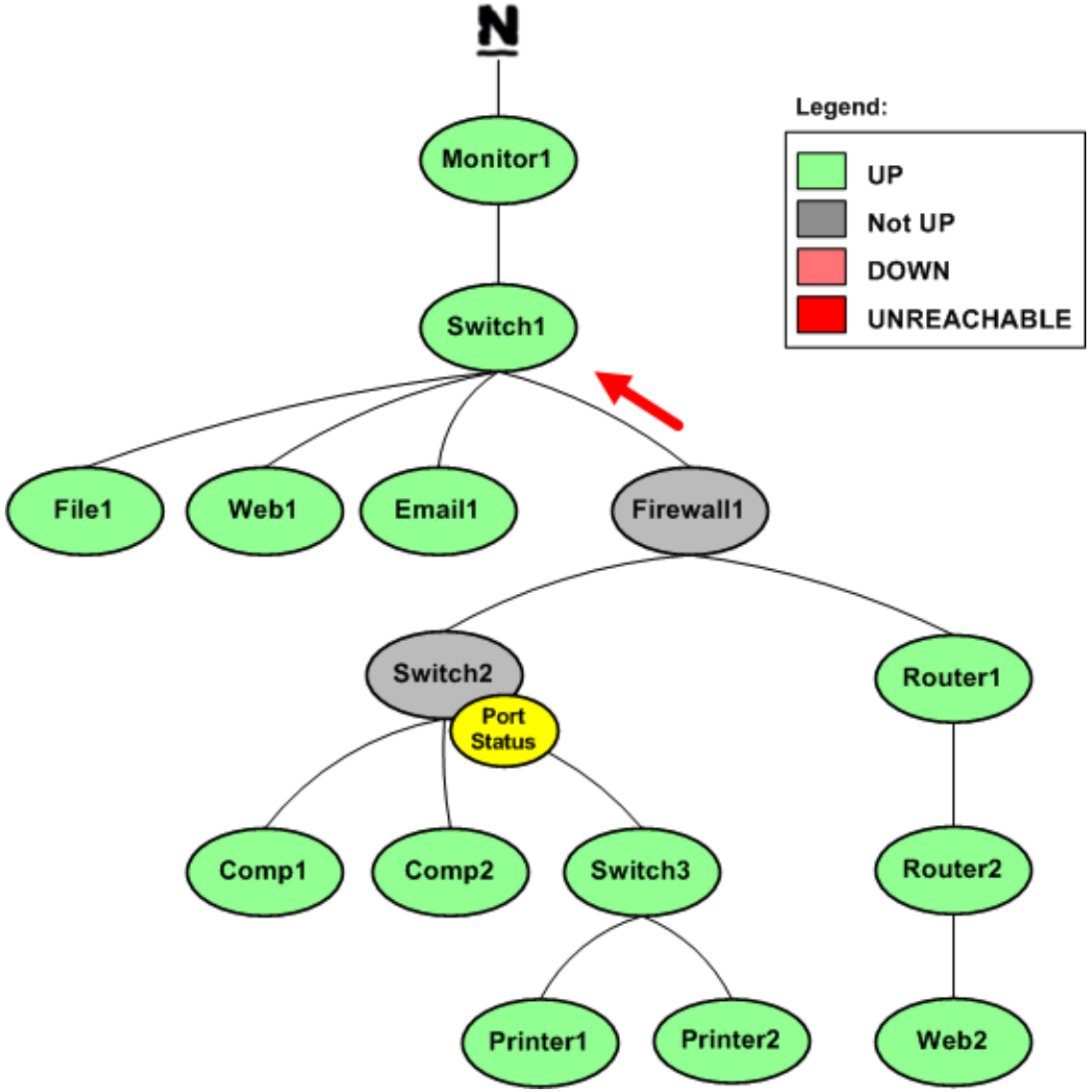
Host Route Verification

- *Check is propagated upstream to Firewall1, which is also NOT UP…*



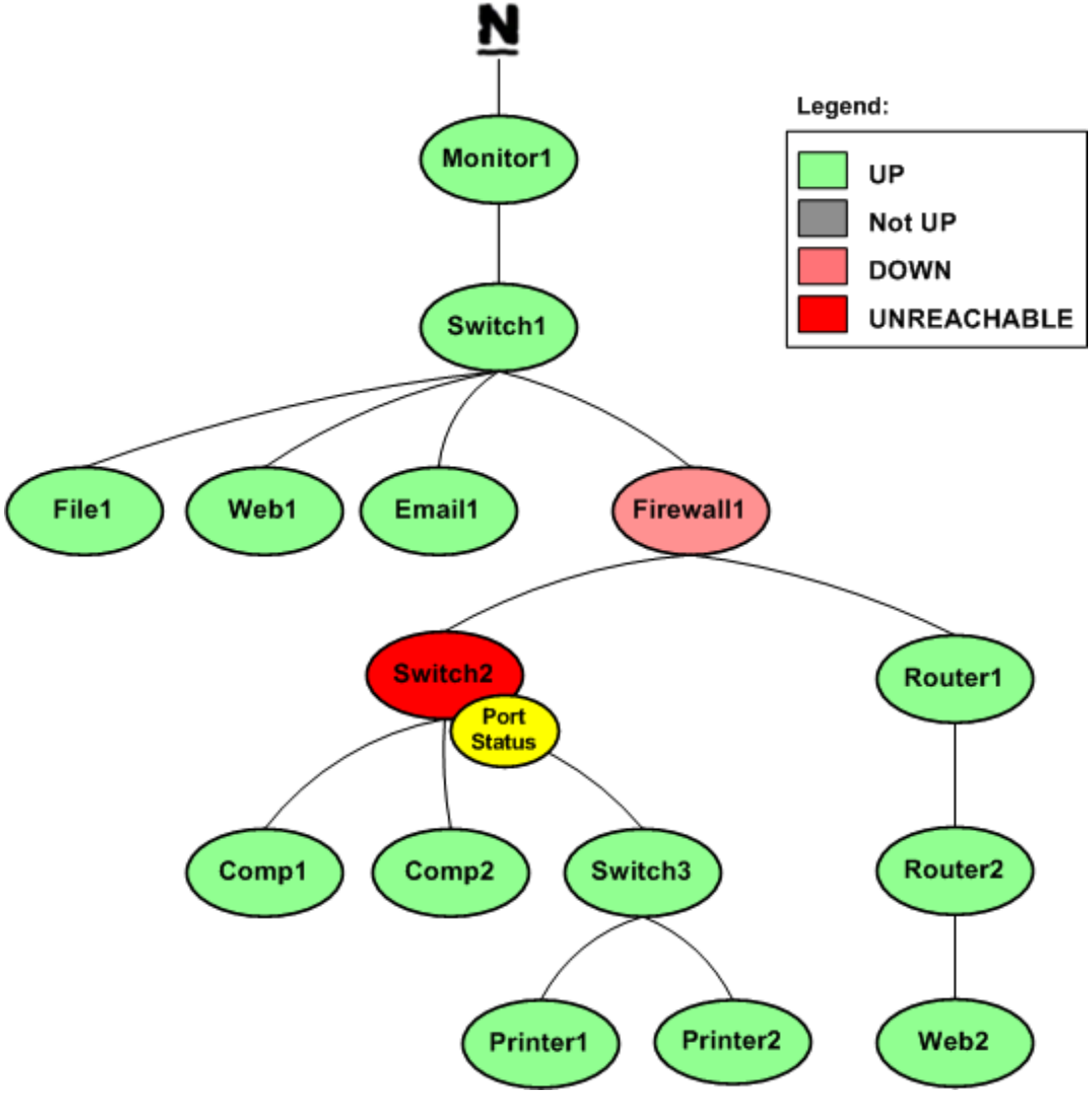
Host Route Verification

- *Check is propagated upstream to Switch1, which IS found to be UP*



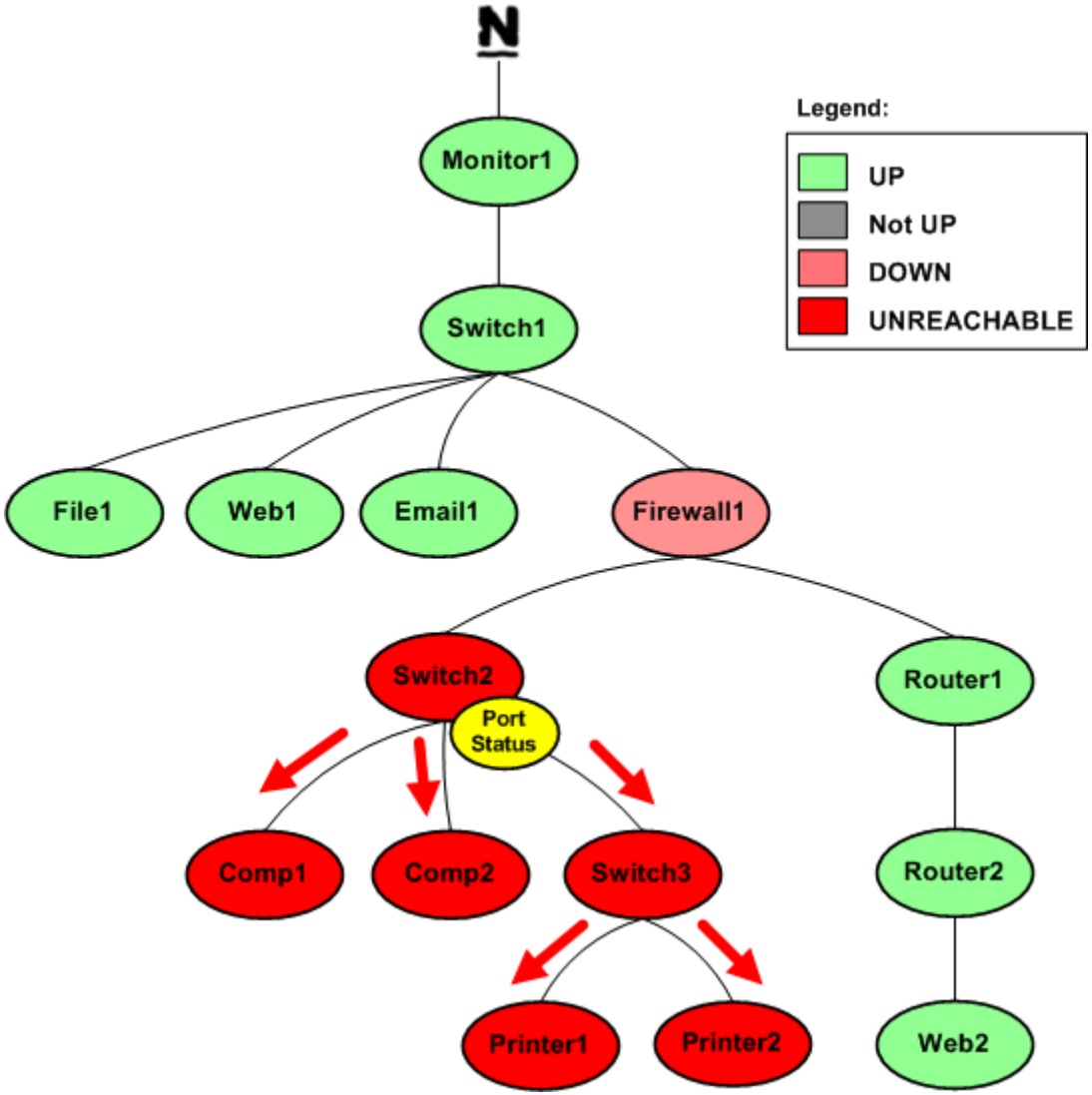
Host Route Verification

- *Reachability of Switch2 can now be determined - Firewall1 is DOWN and Switch2 is UNREACHABLE.*



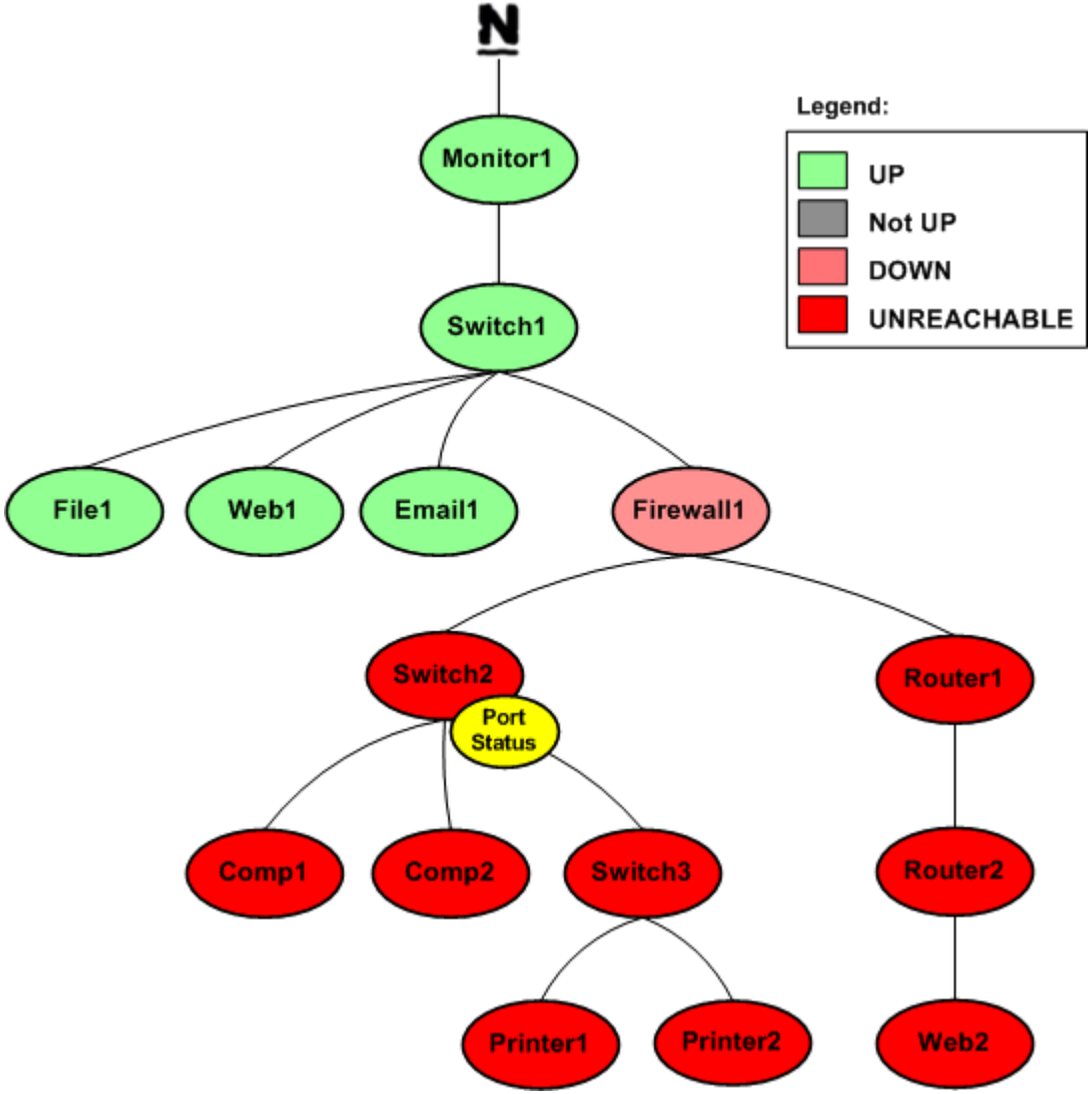
Host Route Verification

- *Checks are propagated to children of SWITCH2, which are found to be UNREACHABLE*



Host Route Verification

- *Other children of Firewall1 are checked later, found to be UNREACHABLE*



External Application Interfaces

External Application Interfaces

- External Commands
- Passive Checks
- Event Handlers (not discussed)
- OCSP/OCHP Commands (not discussed)
- Performance Data (not discussed)

External Commands

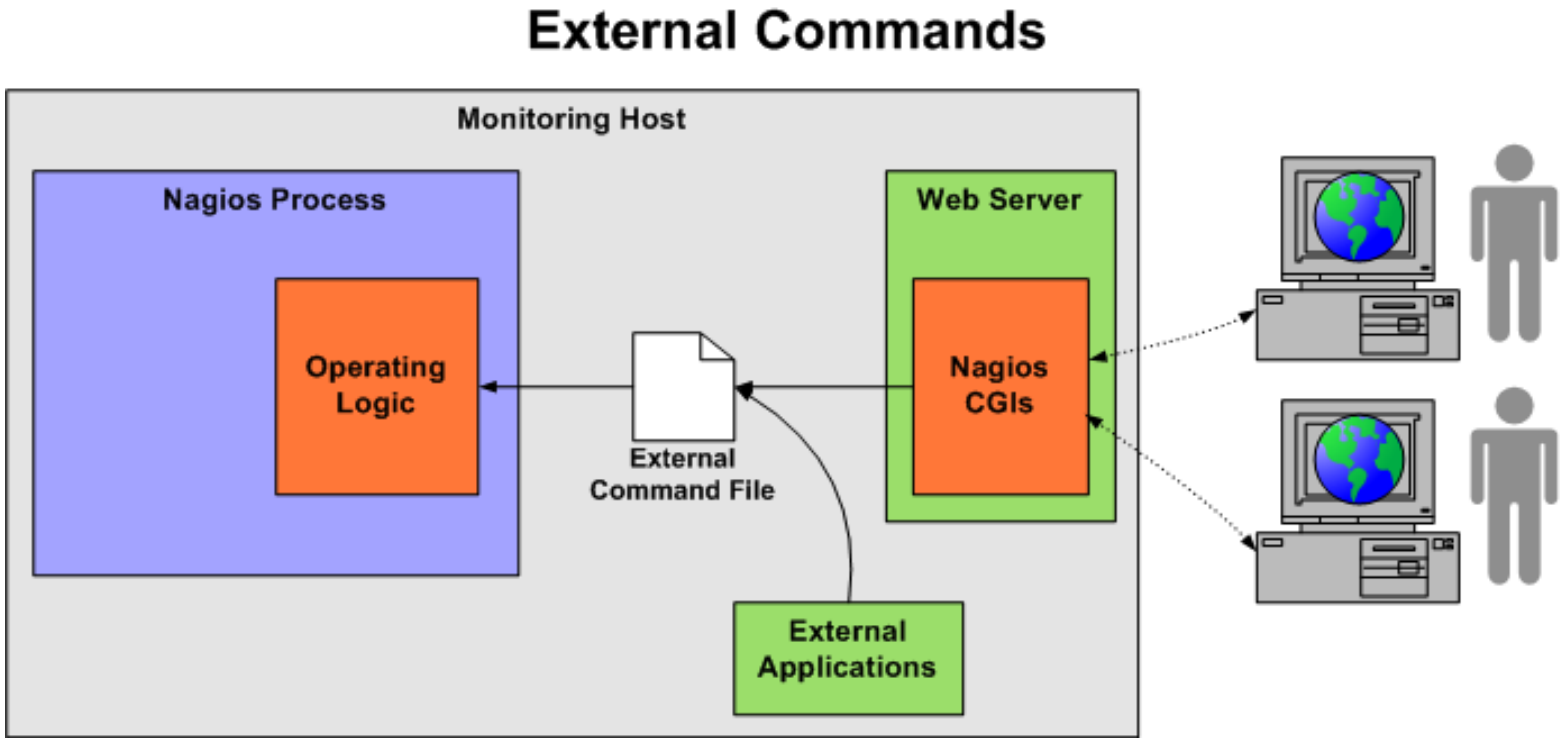
External Commands

- Nagios can accept control commands and data from third-party applications through the external command interface

- **What can external commands be used for?**
  - Controlling aspects of the monitoring and processing logic
    - Disabling/enabling host and service checks
    - Disabling/enabling notifications
    - etc.
  - Submitting host and service check results

External Commands

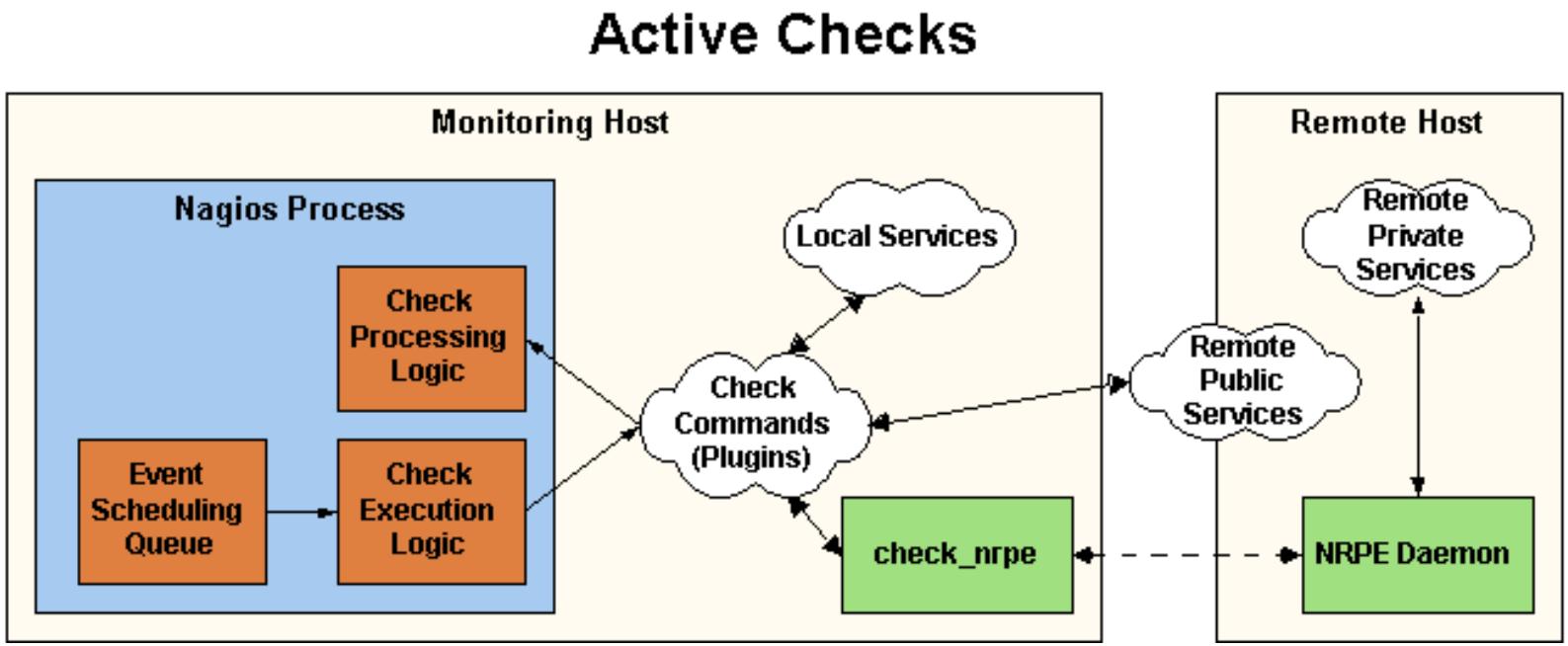
- **How are commands submitted?**
  - Nagios creates a named pipe (FIFO) at startup, reads the file at regular intervals and processes any new commands that are found
  - CGIs and 3rd-party applications submit commands to Nagios by writing to the external command file



Passive Checks

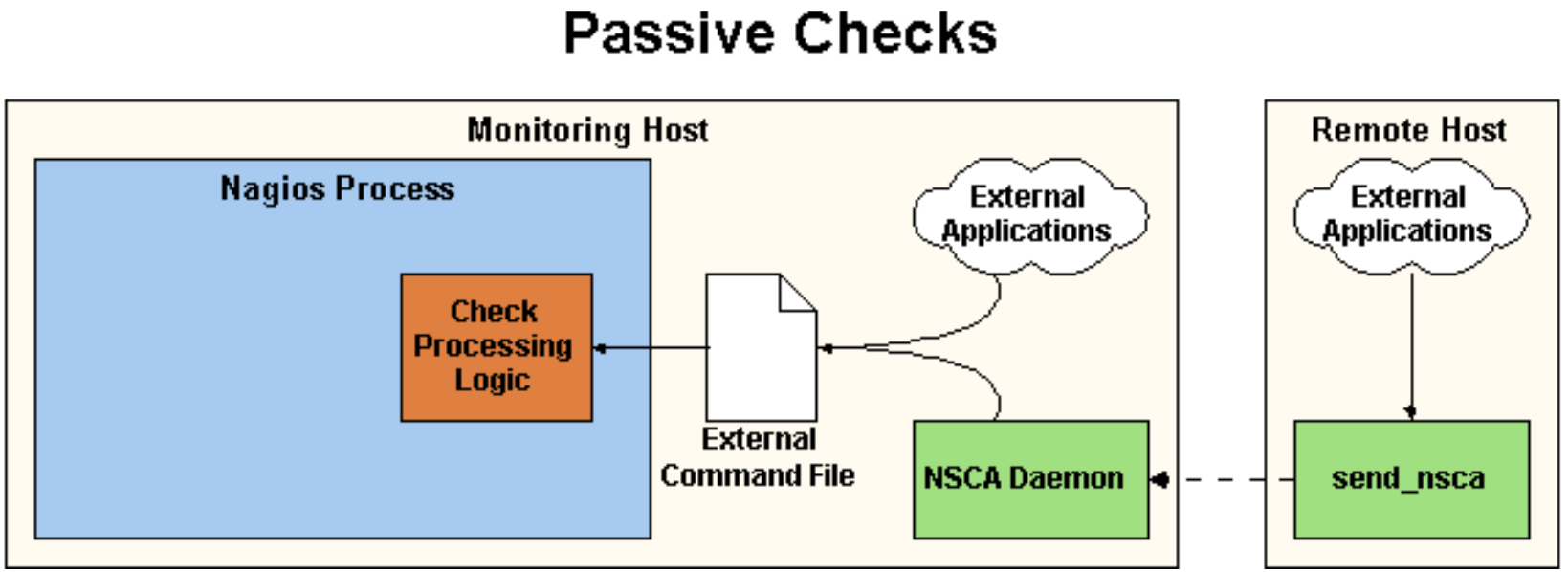
Passive Checks

- **Methods for checking hosts and services**
  - Actively
  - Passively
- **Active checks**
  - Synchronous in nature
  - Scheduled and initiated by Nagios



Passive Checks

- **Passive checks**
  - Performed by external applications
  - Check results are submitted through the external command file
  - Asynchronous in nature
  - Results are processed using the same logic as active checks
  - Ideal for incorporating security alerts



Do's and Dont's

Do's and Dont's

First of all:

- Nagios is a program, not a person
- Nagios cannot replace disciplined support staff
- Nagios is supposed to help your support staff, not bother them
- Nagios can be your best friend in maintainance

Do's and Dont's

Do's:

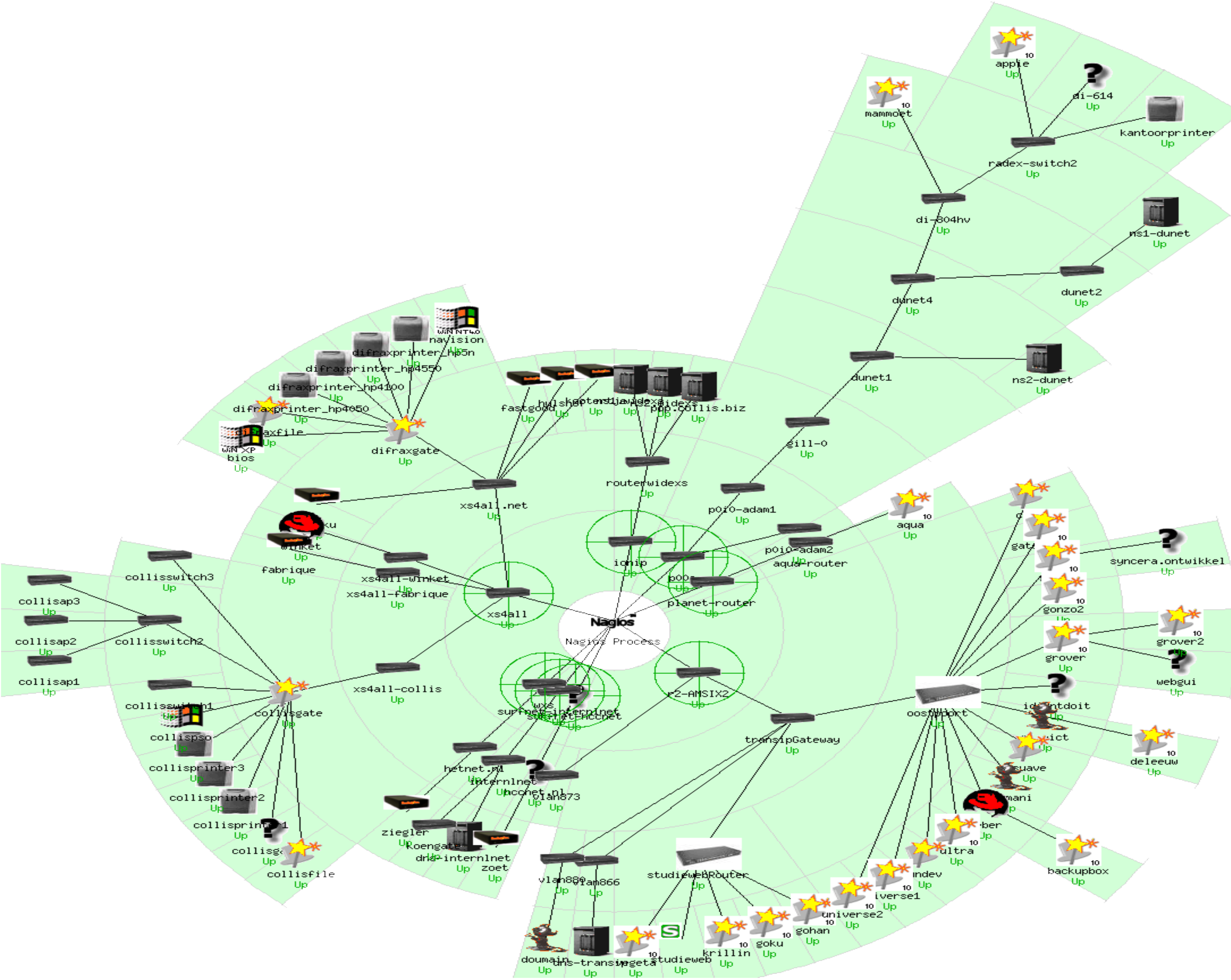
- Use Nagios to check everything you used to check manually
- Create checks for everything that might cause serious problems
- Create checks for everything that is easy to check
- Make sure your list of problems is always empty
- Create time periods
(you don't care about printers in the middle of the night)
- Check less important services less often
- Use a dedicated machine for Nagios

Do's and Dont's

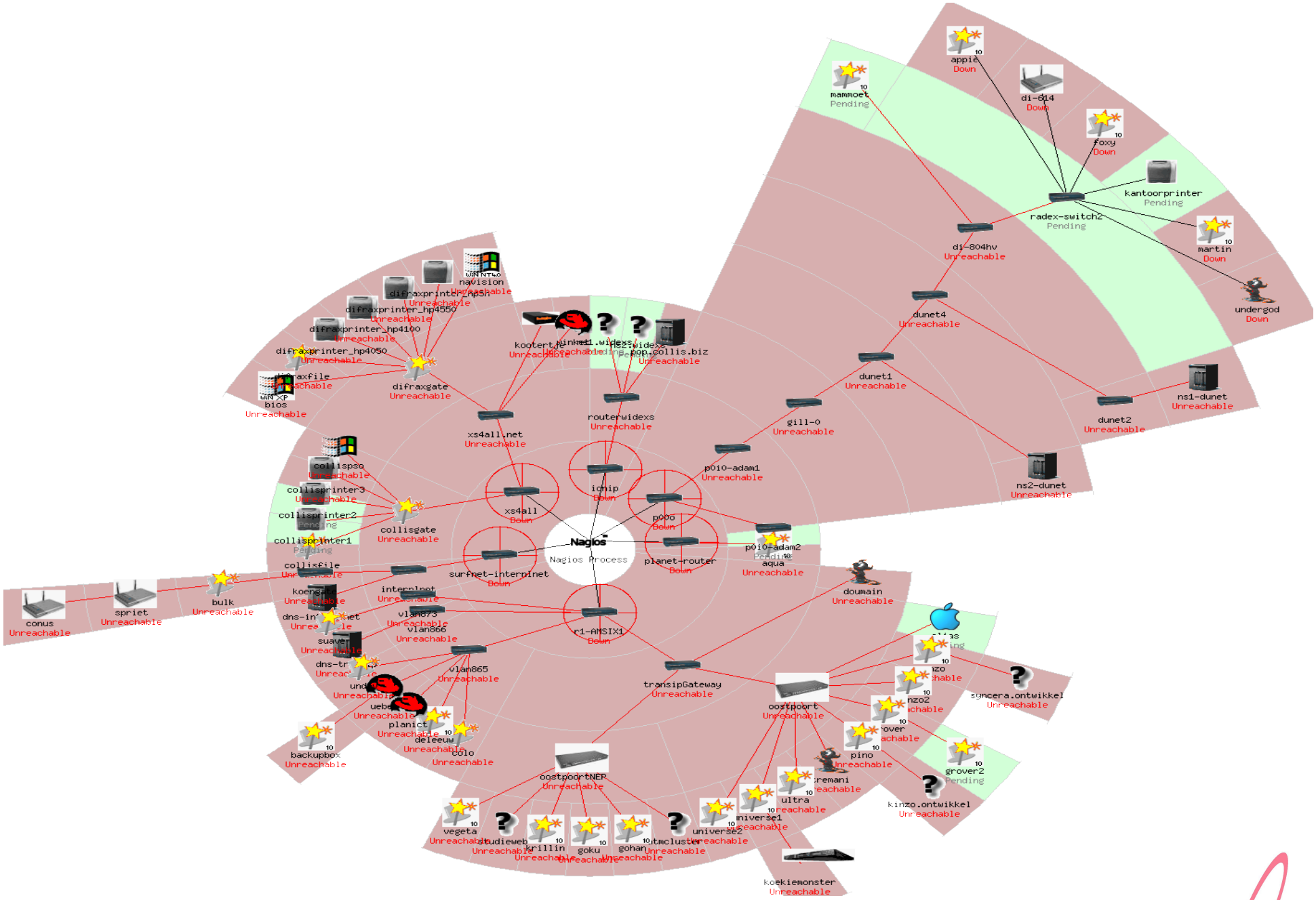
Do not:

- Use Nagios to check the discipline of your staff
- Leave problems 'dangling', ACKNOWLEDGE them!
- Create 'progressing' checks, always use hard limits
- Check hosts or services you have no control over
- Think you can configure Nagios without reading the documentation
- Trust Nagios blindly, check the checks periodically

And Finally...


apple
di-614
mammoet
kantoorprinter
radex-switch2
di-804hv
ns1-dunet
dunet4
dunet2
dunet1
ns2-dunet
gill-o
p0i0-adam1
aqua
p0i0-adam2
aqua-router
syncera.ontwikkel
gonzo2
grover2
grover
webgui
oostpoort
deleeuw
backupbox
ultra
universe2
gohan
goku
krillin
studieweb
studiewebRouter
transipGateway
vlan880
vlan866
doumain
r2-AMSIX2
planet-router
routerwidexs
pop.collis.biz
fastgood
navision
difraxprinter_hp5n
difraxprinter_hp4550
difraxprinter_hp4100
difraxprinter_hp4050
difraxgate
bios
xs4all.net
fabrique
xs4all-winket
xs4all-fabrique
xs4all
Nagios
Nagios Process
collisswitch3
collisap3
collisap2
collisswitch2
collisap1
xs4all-collis
collisgate
collisprinter3
collisprinter2
collisfile
hetnet.nl
vlan873
ziegler
koengate
dns-internlnet
zoet

Procolix


Nagios
Nagios Process
appie
Down
mammoet
Pending
di-614
Down
foxy
Down
kantoorprinter
Pending
radex-switch2
Pending
martin
Down
undergod
Down
di-804hv
Unreachable
dunet4
Unreachable
dunet1
Unreachable
dunet2
Unreachable
ns1-dunet
Unreachable
ns2-dunet
Unreachable
gill-0
Unreachable
p0i0-adam1
Unreachable
p0i0-adam2
aqua
Unreachable
routerwidexs
Unreachable
pop.collis.biz
Unreachable
xs4all.net
Unreachable
ipmip
Down
xs4all
Down
p0i0
Down
planet-router
Down
surfnet-internet
Down
r1-AMSIX1
Down
transipGateway
Unreachable
doumain
Unreachable
oostpoort
Unreachable
oostpoortNEP
Unreachable
collisgate
Unreachable
collisfile
collisprinter1
collisprinter2
collisprinter3
collisp50
difraxprinter_hp4050
difraxprinter_hp4100
difraxprinter_hp4550
difraxprinter_hp5
difraxgate
Unreachable
bios
Unreachable
navision
Unreachable
conus
Unreachable
spriet
Unreachable
bulk
Unreachable
koenraad
Unreachable
internet
Unreachable
vlan873
Unreachable
vlan866
Unreachable
vlan865
Unreachable
suaver
Unreachable
planict
Unreachable
deleeuw
Unreachable
colo
Unreachable
backupbox
Unreachable
pino
Unreachable
ultra
Unreachable
universe1
Unreachable
universe2
Unreachable
utmcluster
Unreachable
gohan
Unreachable
goku
Unreachable
krillin
Unreachable
studieweb
Unreachable
vegeta
Unreachable
grover2
Pending
kinzo.ontwikkel
Unreachable
syncera.ontwikkel
Unreachable
koekiemonster
Unreachable

References

For more information on Nagios look here:

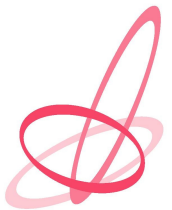
- http://www.nagios.org
- http://nagiosplug.sourceforge.net

For performance data and trend analyses look here

- http://www.cacti.net
- http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/

For information on Koen de Jonge

- http://www.procolix.com
- http://www.gevat.nl (dutch)



Questions?