

Nagios[®]

Advanced Monitoring Topics

NETWAYS **Nagios**
Konferenz 

September 22nd, 2006

Ethan Galstad
nagios@nagios.org

Advanced Monitoring

Topics:

- Optimizations for large installations
- Templates optimization
- Event broker API
- NDOUtils addon
- Security considerations

Optimizing Nagios in Large Installations

Large Install Optimization

Large installation tweaks:

- `use_large_installation_tweaks=[0/1]`
- Checks `fork()` only once instead of twice
- Different child process (zombie) cleanup
- Different memory handling (cleanup left to OS)
- No summary macros (too much CPU overhead)

Large Install Optimization

Faster startup options:

- Nagios 3 can now (re)start much faster
- Faster restarts when adding new hosts/services
- Speedups come from:
 - Pre-processing / pre-caching object config
 - Skipping circular path detection

Fast Startup Options

Startup times:

- Object config processing times
 - Read
 - Resolve
 - Duplicate
 - Inherit
 - etc.
- Config verification times
 - Object relationships
 - Circular path tests
- Speedup estimates displayed with -s (scheduling test)

Fast Startup Options

Object config processing speedups:

- Significant time can be saved if we pre-process the configuration and cache it
- In this case there is a 50% potential time savings!

Object Config Source: Object config files

OBJECT CONFIG PROCESSING TIMES (* = Potential for precache savings)

```
-----  
Read:                0.149578 sec  
Resolve:             0.029014 sec *  
Recomb Contactgroups: 0.000078 sec *  
Recomb Hostgroups:  0.000156 sec *  
Dup Services:       0.007755 sec *  
Recomb Servicegroups: 0.003474 sec *  
Duplicate:          0.377729 sec *  
Inherit:            0.001304 sec *  
Recomb Contacts:    0.008773 sec *  
Sort:               0.035468 sec *  
Register:           0.233234 sec  
Free:               0.006038 sec  
=====
```

TOTAL:	0.861292 sec
Est Precache Savings:	0.472442 sec *

Fast Startup Options

Config verification speedups:

- Significant time can be saved if we skip circular tests (host paths, dependencies, etc.)
- Beware before doing this!
- In this case there is a 50% potential time savings!

```
CONFIG VERIFICATION TIMES
-----
Object Relationships: 0.148683 sec
Circular Paths:      4.491282 sec
Misc:                0.003238 sec
=====
TOTAL:               4.643203 sec
```

With both speedups...

- Startup time initially = 5.50 seconds
- Startup time with both speedups = **0.54 seconds!**

Fast Startup Options

Usage:

- Verify config normally, cache object configuration (-p)

./nagios -vp <config>

- If no errors, start Nagios using cached objects (-u) and skip object path verification (-x)

./nagios -dux <config>

- Before (soft) restarting Nagios in the future, re-verify and re-cache object config

./nagios -vp <config>

Host Check Optimization

Common:

- Optimize host checks
 - More retries (max check attempts)
 - Shorter timeouts
 - Faster ping (check_fping)

Nagios 2.x:

- Use scheduled host checks sparingly (performed serially)

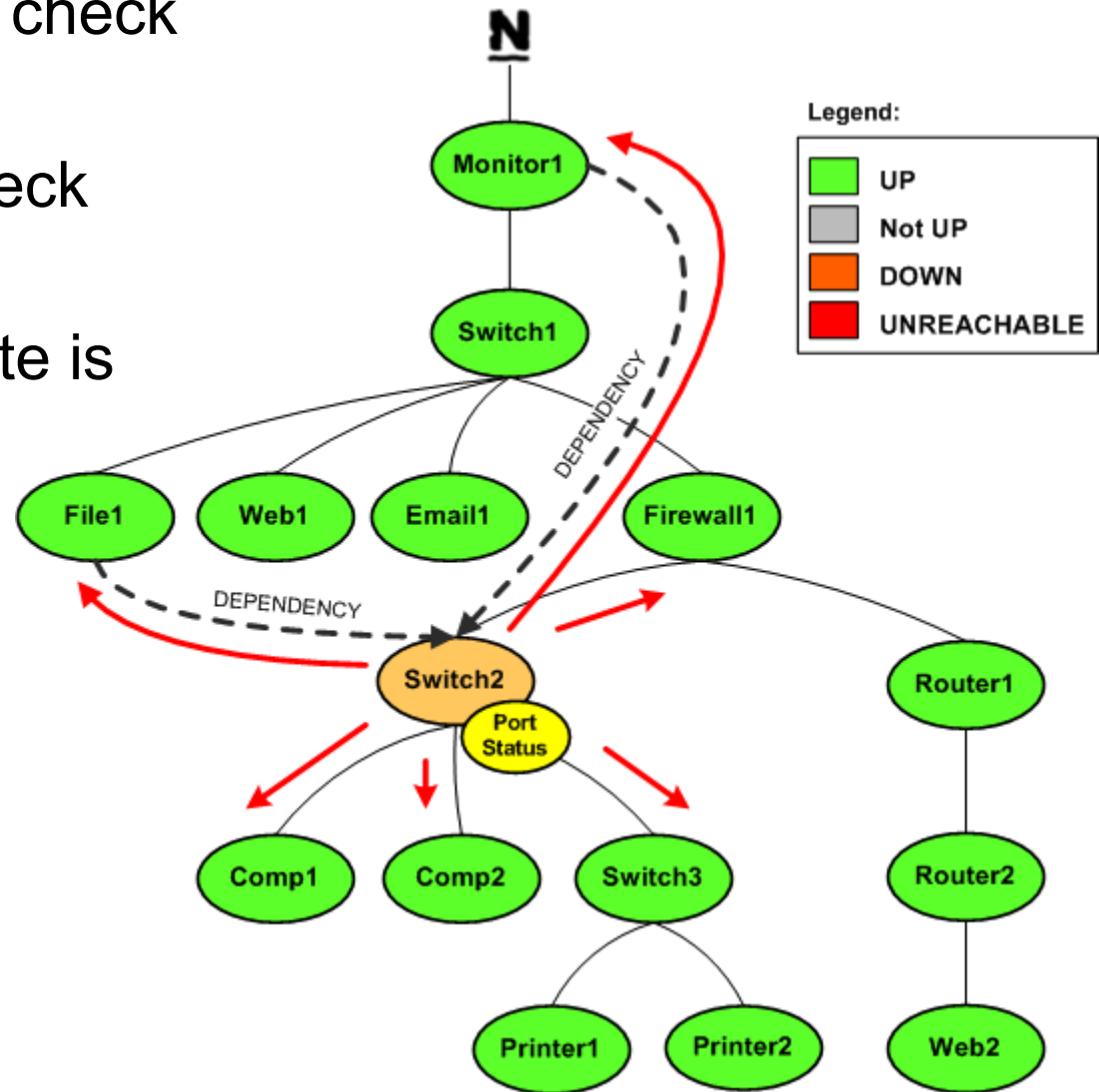
Nagios 3.x:

- Scheduled host checks improve performance (caching)
- Max check attempts should be > 1 (parallelization)

Host Check Optimization

Nagios 3.x Logic:

- Serial host checks if max check attempts = 1
- Parallel checks if max check attempts > 1
- Accurate current host state is important



External Commands

Large installs:

- Distributed setups have a lot of passive checks
- Named pipe may become a bottleneck (4-8K limit)

New command (Nagios 3.x):

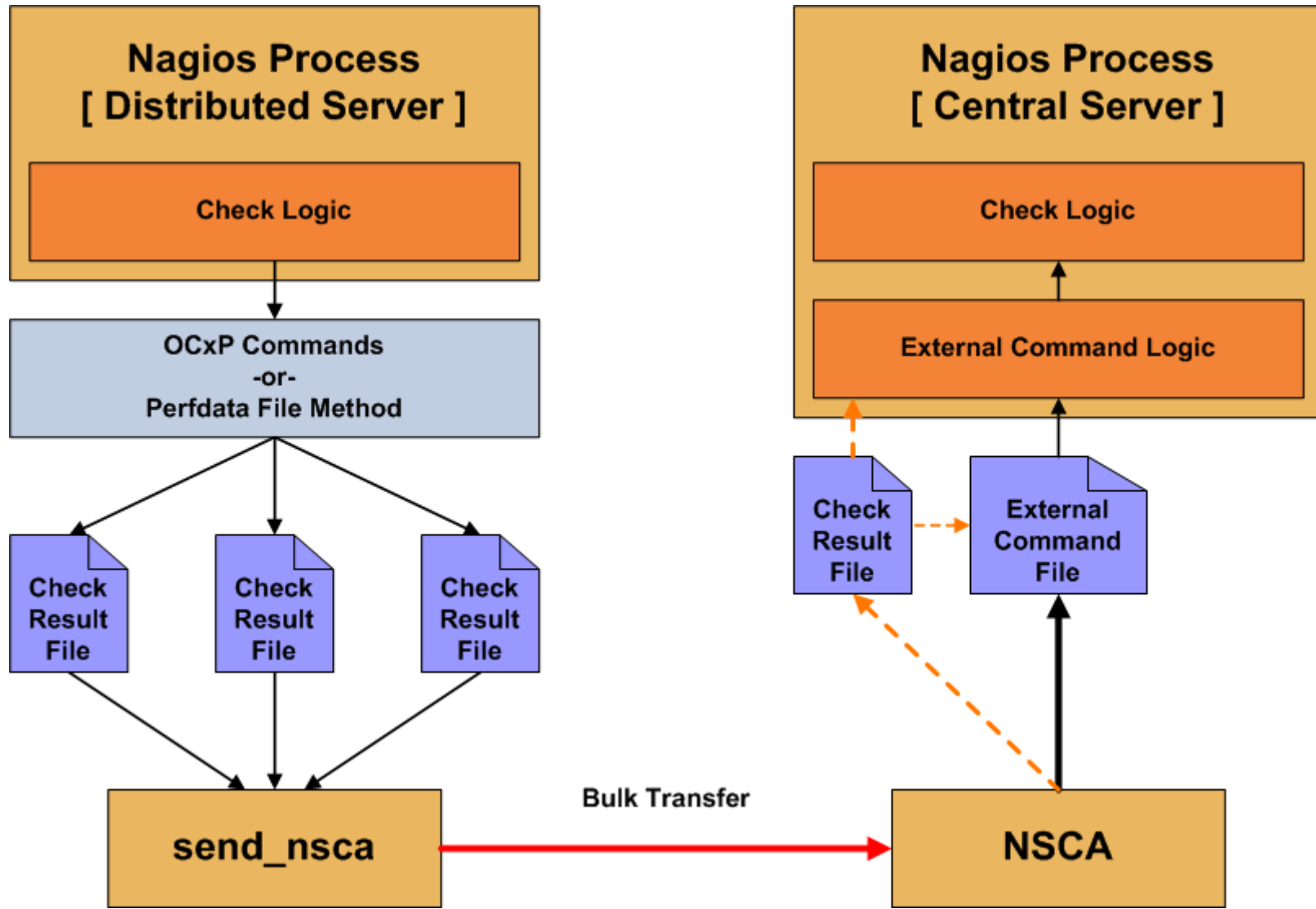
- `PROCESS_FILE <filename>;<delete_file>`
- With one command Nagios will process all commands found in a specified file
- File can be left alone or deleted after processing
- Useful for regularly scripted actions
- Bulk check transfer optimization in NSCA (future)

Distributed Monitoring

Large installs:

- Performance data + passive checks
 - Use either OCSP/OCHP commands or perfdata commands – not both
- Bulk transfers of passive checks to central server
 - Write passive check info to file
 - Every x minutes, files are passed to send_nsca in bulk
 - Less overhead on NSCA daemon and distributed server
 - Bulk processing of passive checks with new external command (future NSCA mod)

Distributed Monitoring



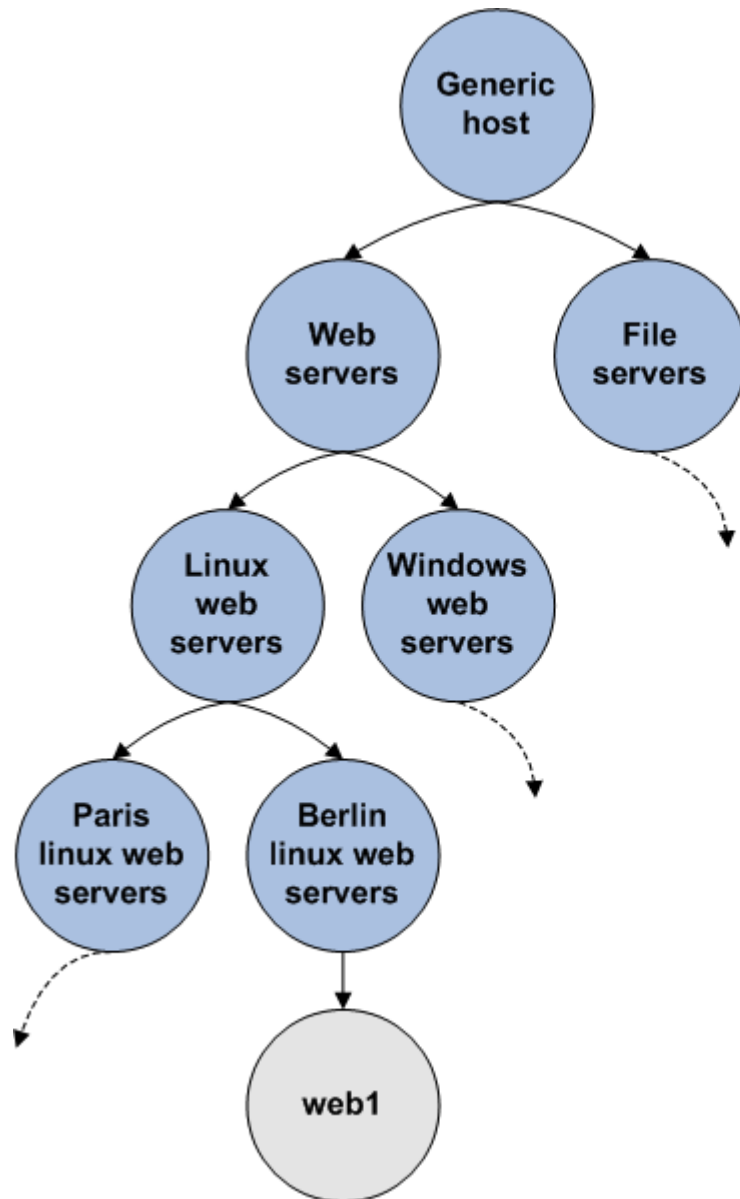
Template Optimization

Template Optimization

Best practices:

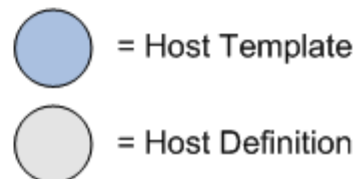
- Create multiple templates for each object type
 - Servers
 - Routers
 - Switches
 - Printers
 - etc...
- Top level templates should be the most generic
- Bottom level templates are the most specific

Template Optimization



Benefits:

- Config is easier to manage
- Easier to change default values for specific classes of hosts and services

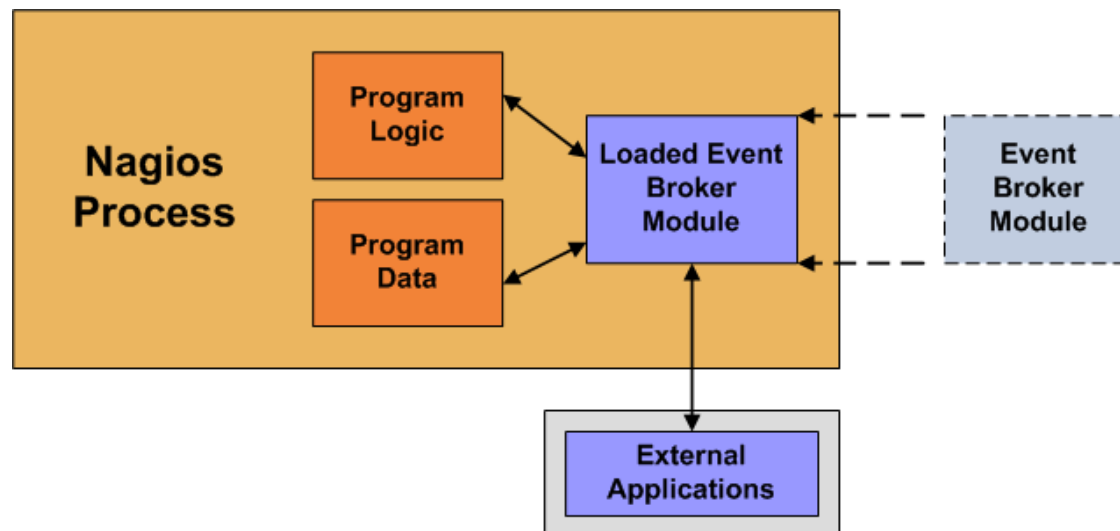


Nagios Event Broker

Event Broker

Overview:

- Incorporates third-party code (modules) into Nagios daemon
- Allows modules to:
 - Trap and handle monitoring events
 - Modify, process, and extract data
 - Incorporate data and commands from external apps



Event Broker

Status:

- Working in Nagios 2.x, minor changes for Nagios 3.x
- Modules must be recompiled for new Nagios versions
- Documentation:
 - Docs by Bob Ingraham (2.x)
 - <http://www.nagios.org/developerinfo/>
- Examples:
 - NDOUtils addon

NDOUtils Addon

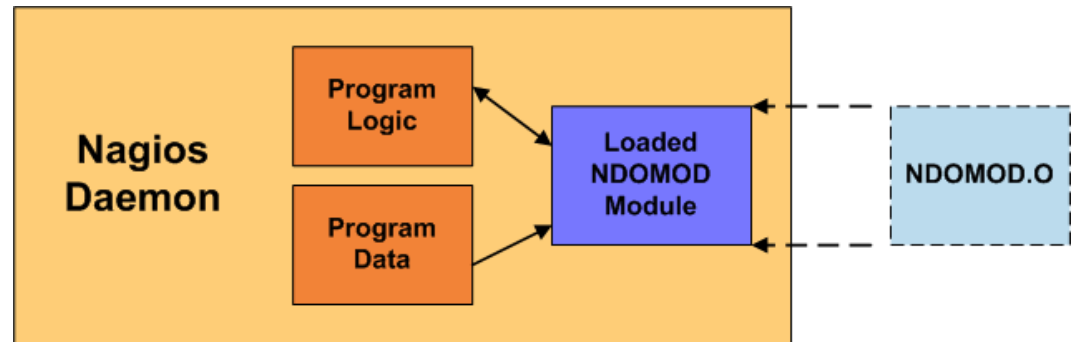
Overview:

- Event broker module and daemon
- Nagios config, status, and event information stored in DB
- Ability to import historical NetSaint/Nagios logs
- MySQL supported now, Postgres in future
- Current log, status files unaffected
- Works with both Nagios 2.x and 3.x
- Currently in development – bugs to fix, docs to write

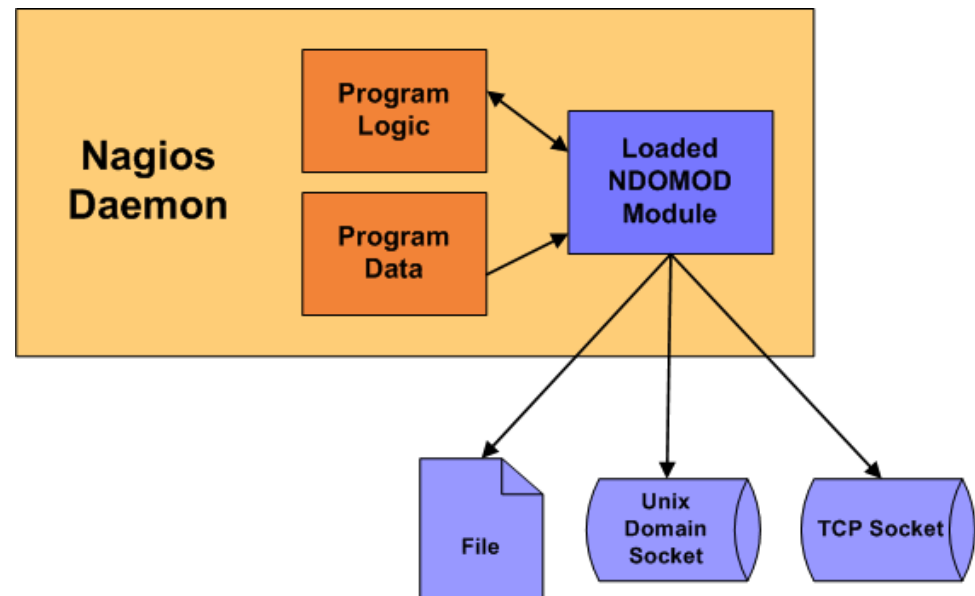
NDOUtils Addon

How it works:

- NDOMOD module is loaded into Nagios



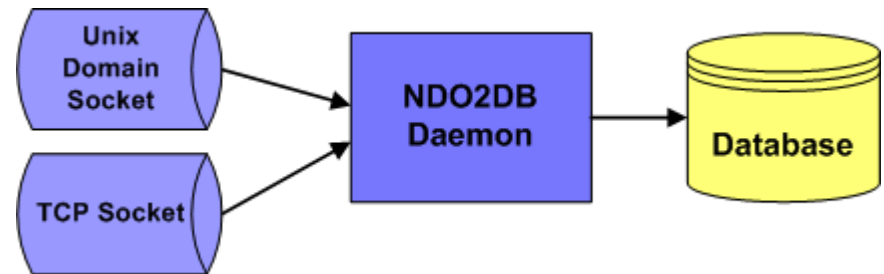
- Data is sent to file, TCP or UNIX domain socket



NDOUtils Addon

How it works:

- NDO2DB daemon reads data from a TCP or UNIX domain socket

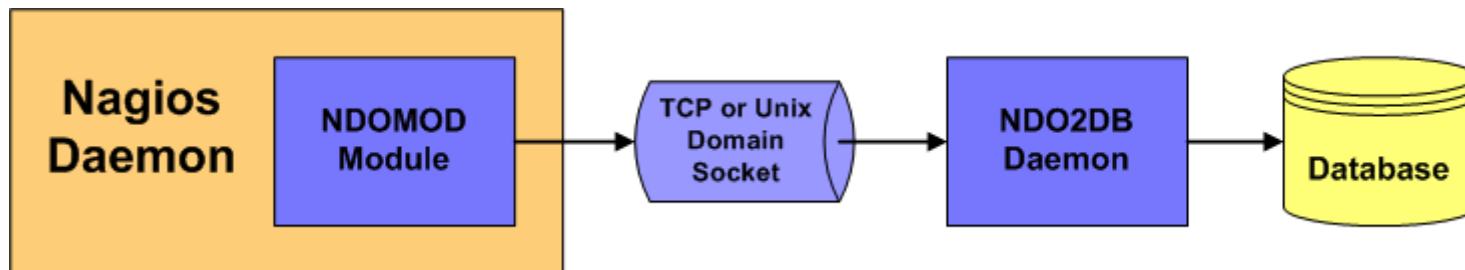


- Data is processed and stored in a DB

NDOUtils Addon

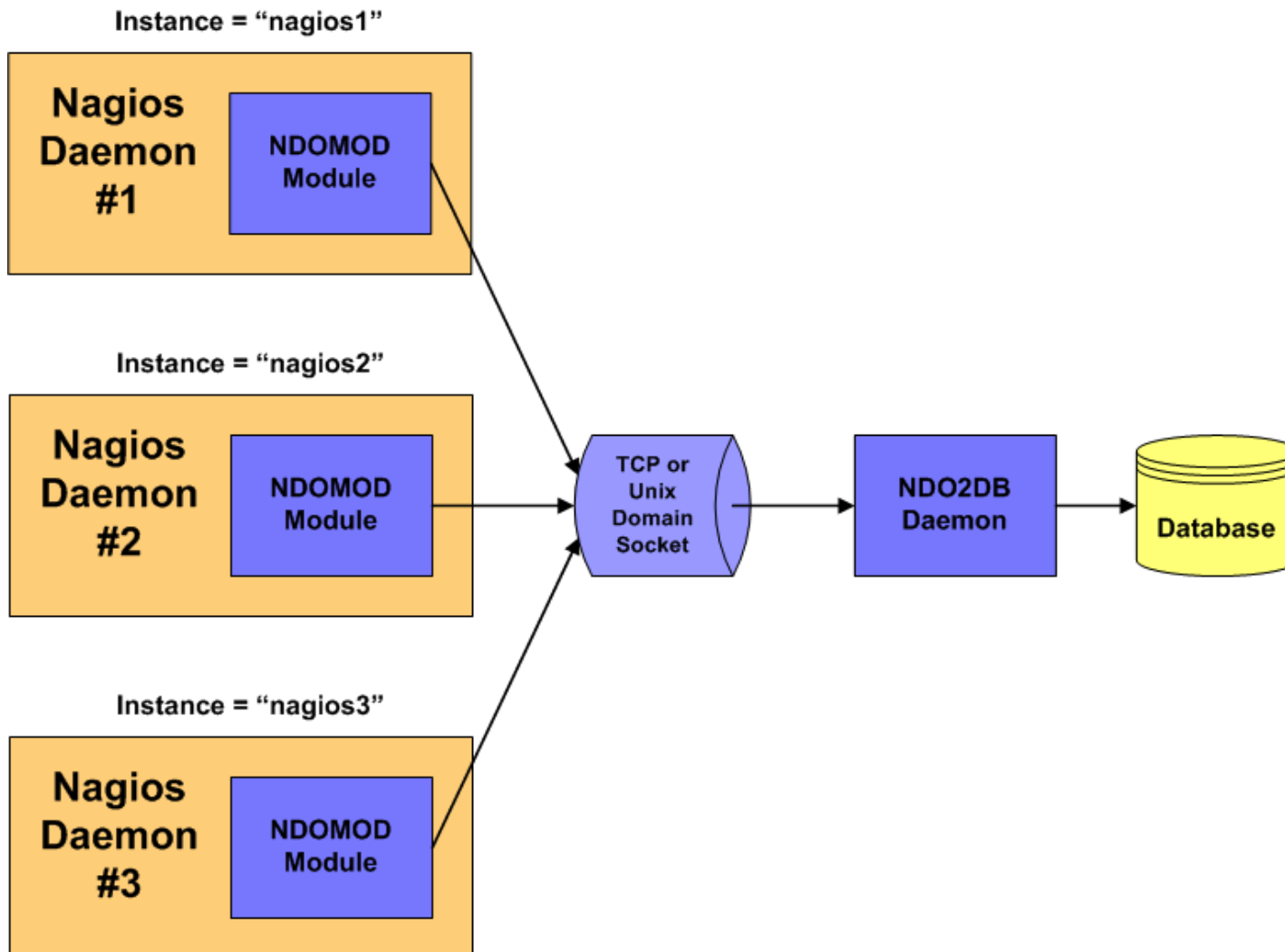
Simple Use:

- Data passes over TCP or Unix domain socket
- Daemon processes data and stores in DB



Multi-Instance Support

Support for multiple Nagios instances:



Security Considerations

Security Considerations

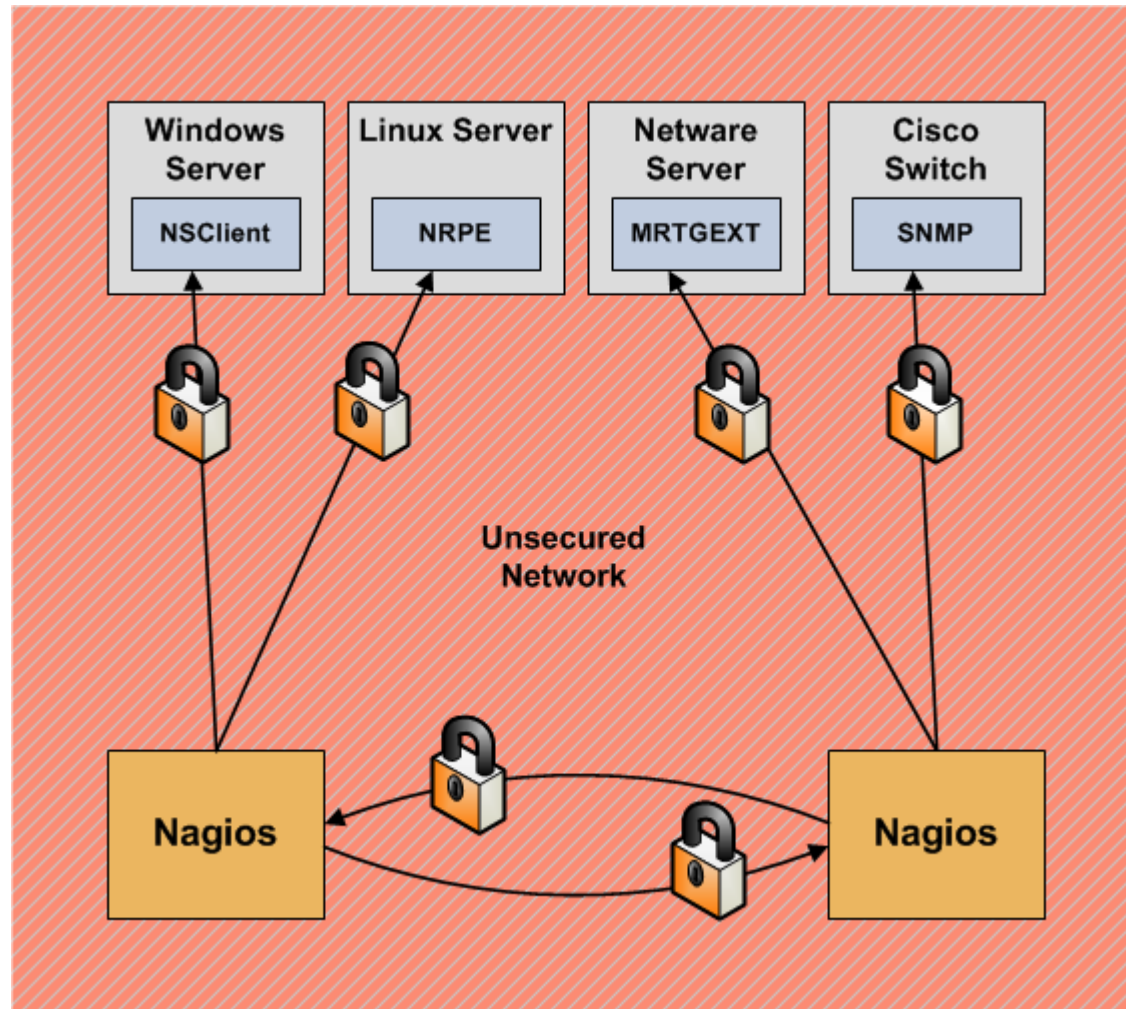
Remember:

- Monitoring solutions should enhance your security, not weaken it!
- Monitoring can introduce holes in your security perimeter
- Use caution when designing and deploying a monitoring solution



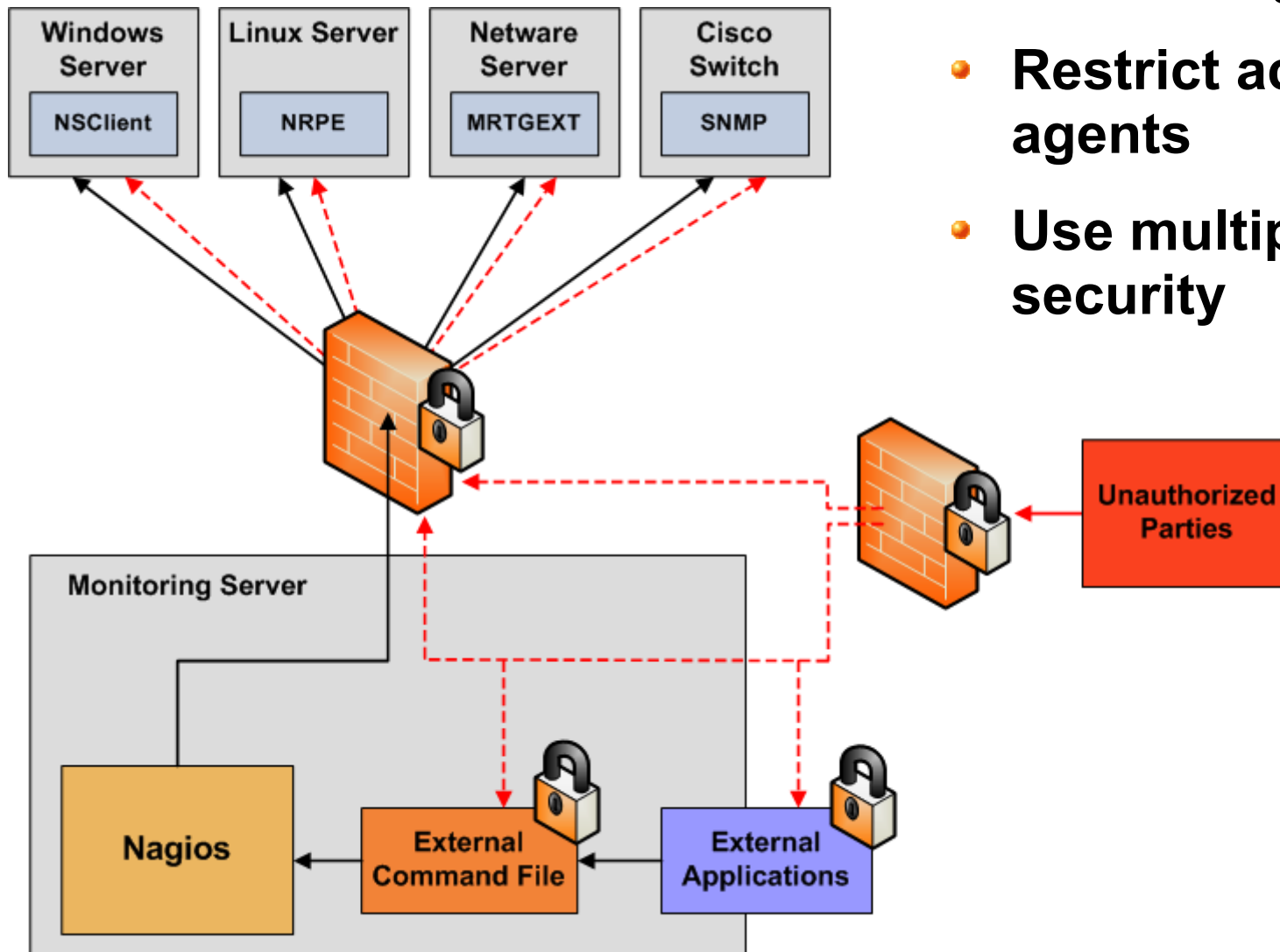
Security Considerations

- Secure communication channels wherever possible...



Security Considerations

- Lock down your monitoring server
- Restrict access to remote agents
- Use multiple layers of security



Questions?



Ethan Galstad
nagios@nagios.org