

**OpenACS and EuroTcl 2025**

# NaviServer 5.0 (Release)



Univ.-Prof. Dr. Gustaf Neumann  
Vienna University of Economics and Business  
Information Systems and New Media

JULY 10, 2025



# Overview

## ■ NaviServer 5 is released

- Has the world waited for it?
- Was it necessary?
- NaviServer 4.99.1 ... 4.99.31

v3.0: 2000, v4.0: 2001, v4.99: 2005

## ■ Many Changes ...

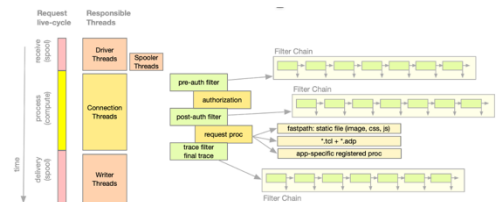
- Most extensive NaviServer release of the last 25 years

522 files changed, 78K insertions, 33K deletions  
Changelog: 18K lines

- Some architectural changes
- Many new features (some previews already presented last year)

## ■ ... such as:

- Tcl9 support
- Many security enhancements
- Improved scalability, flexibility, and configurability
- Substantial cleanup
  - Code
  - Documentation
- New HTML Design



Security by Default

## ■ Security enhancements



- Security by default, esp. peer certificate validation
- Bundled CA certificates and fine-grained validation policies
- Pluggable authorization framework
- Context Constraints
- Many cryptographic improvements, e.g.



- Argon2 password hashing, SCRAM-sha-256, ...
- Passing certificates as file and/or strings
- ...

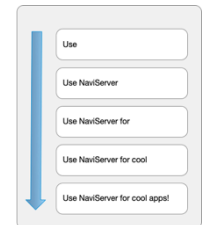


Security by Default

## ■ Other key features



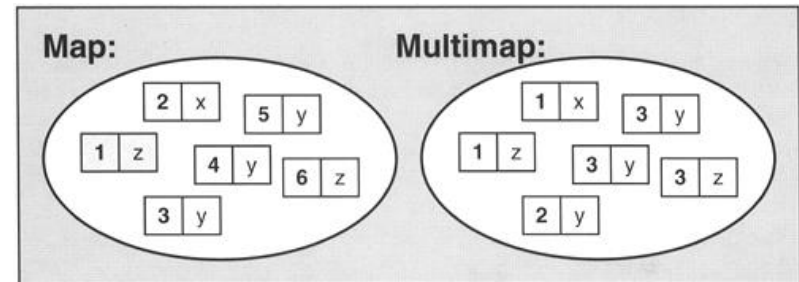
- Optionally case insensitive sets
- Generalized logdir/serverdir handling (mass virtual hosting)
- Reverse/forward proxy in core (with UNIX socket and streaming support)
- Responsive start page UI with dark mode support
- Extended Env-variable-based config (Docker/Kubernetes-ready)
- New commands for developer (`ns_parsehtml`, `ns_percentencode`, `ns_joinurl`, ...,
- Extended introspection: `ns_conn details`, `ns_conn urldict`, `ns_server hosts`, ...
- Revived TclPro debugger



# Extending the power of ns\_sets (1/2)

## ■ What are ns\_sets

- NaviServer data structure for developers
- Technically multimap (a dict is a map)
- ns\_sets have names, provide automated garbage collection, ...



## ■ Used in NaviServer e.g. for

- HTTP header fields
- Configuration values
- SQL tuples
- ...

```
set h [ns_set create headers]
ns_set put $h Set-Cookie a=1
ns_set put $h Set-Cookie b=2
ns_set put $h Content-type text/plain
```

## ■ New Features in NaviServer 5

- **Case insensitivity** (major uses are per definition case-insensitive)
- **Reading statistics** (e.g., was a configuration parameter read? Is this the default?)
- **Performance improvements:** Improved locality and CPU cache hit rate, less locks

# Extending the power of ns\_sets (2/2)

## ■ NaviServer 4.99

- Some sets are per definition case-insensitive
- Case-insensitivity developer burden
- 7 commands starting with "i"
- Error prone, when omitting "i",  
-> Multiple errors in OpenACS and NaviServer

```
set h [ns_conn headers]
set ct [ns_set iget $h Content-Type]
ns_set iupdate $h Content-Length 0
ns_set get $h User-Agent      ;# ERROR
```

**ERROR**

## ■ NaviServer 5

Some sets default case-insensitive

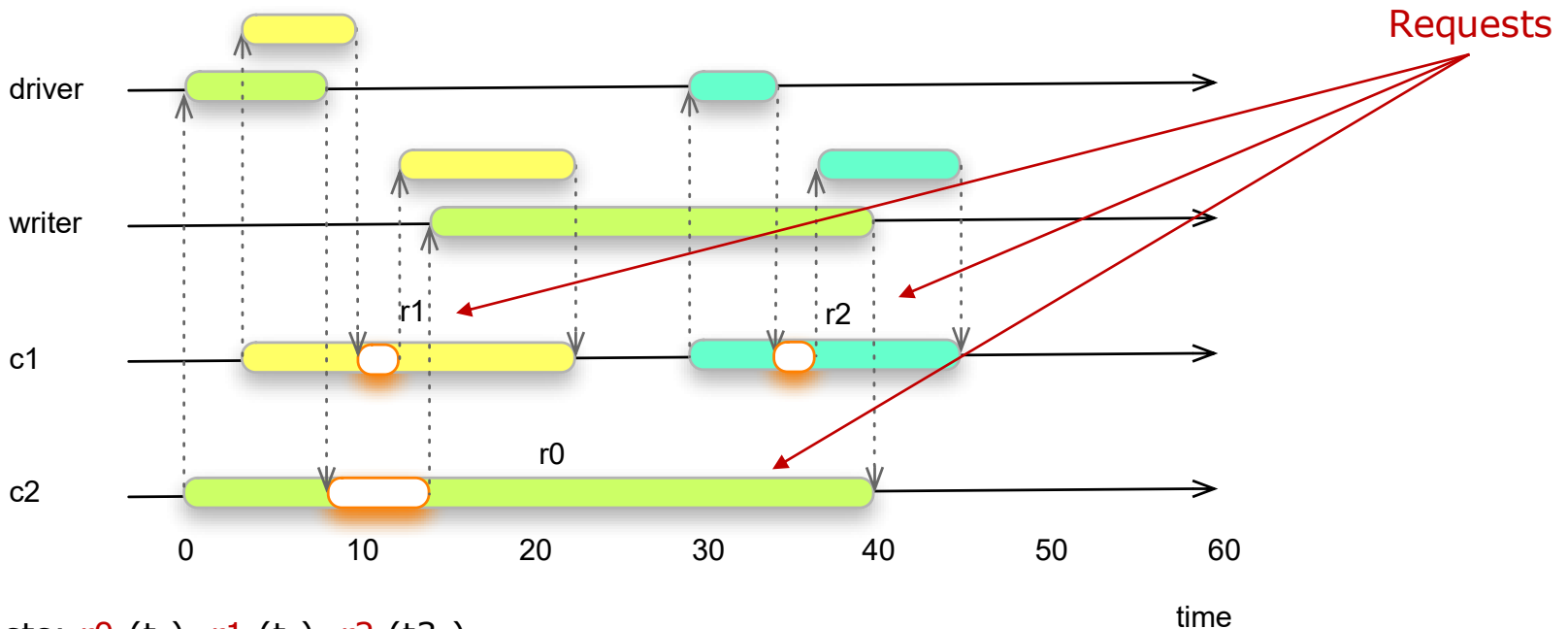
- HTTP header fields
  - Configuration values
  - SQL tuples
- 
- New sets can be defined insensitive with "-nocase flag)
  - Performance impact: key comparison ~20x faster

```
set h [ns_conn headers]
set ct [ns_set get $h Content-Type]
ns_set update $h Content-Length 0
ns_set get $h User-Agent      ;# OK
```

```
set h [ns_set create foo -nocase]
ns_set update -nocase $x foo bar
```

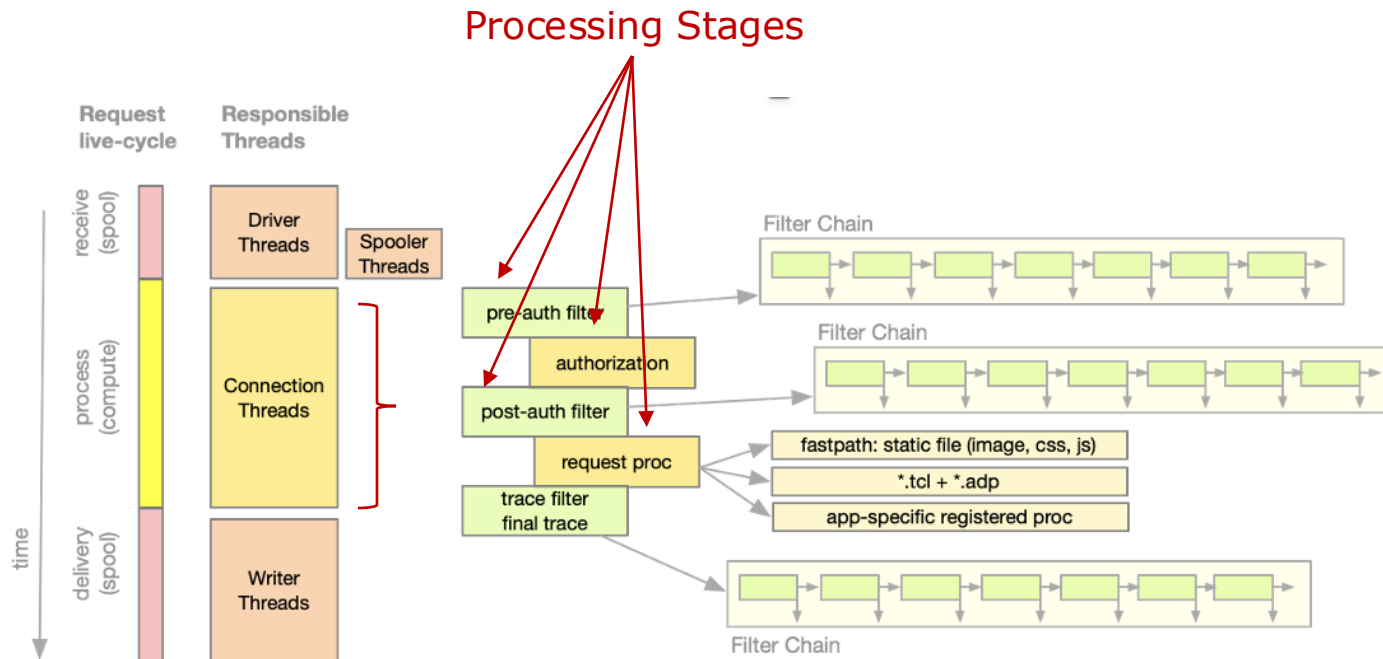


# Handling Concurrent Requests in NaviServer



- Requests: **r0** ( $t_0$ ), **r1** ( $t_5$ ), **r2** ( $t_{30}$ )
- Spooling Threads:** driver, writer shared for many requests  
One spooling thread can handle multiple (hundreds, thousand concurrent connections)
- Connection Threads:** c1, c2, ... exclusive per request  
Only orange parts are actually executed in the connection threads
- By the use of spooling threads
  - r0** uses 5 instead of 40 time-units in connection threads (total instead 9 instead of 70 time-units)
  - Throughput increased, NaviServer can make better use of multiple cores
  - Spooling times depend on client, mitigation for slow write/slow read attacks

# Processing Stages for a Single Request in NaviServer 4.99

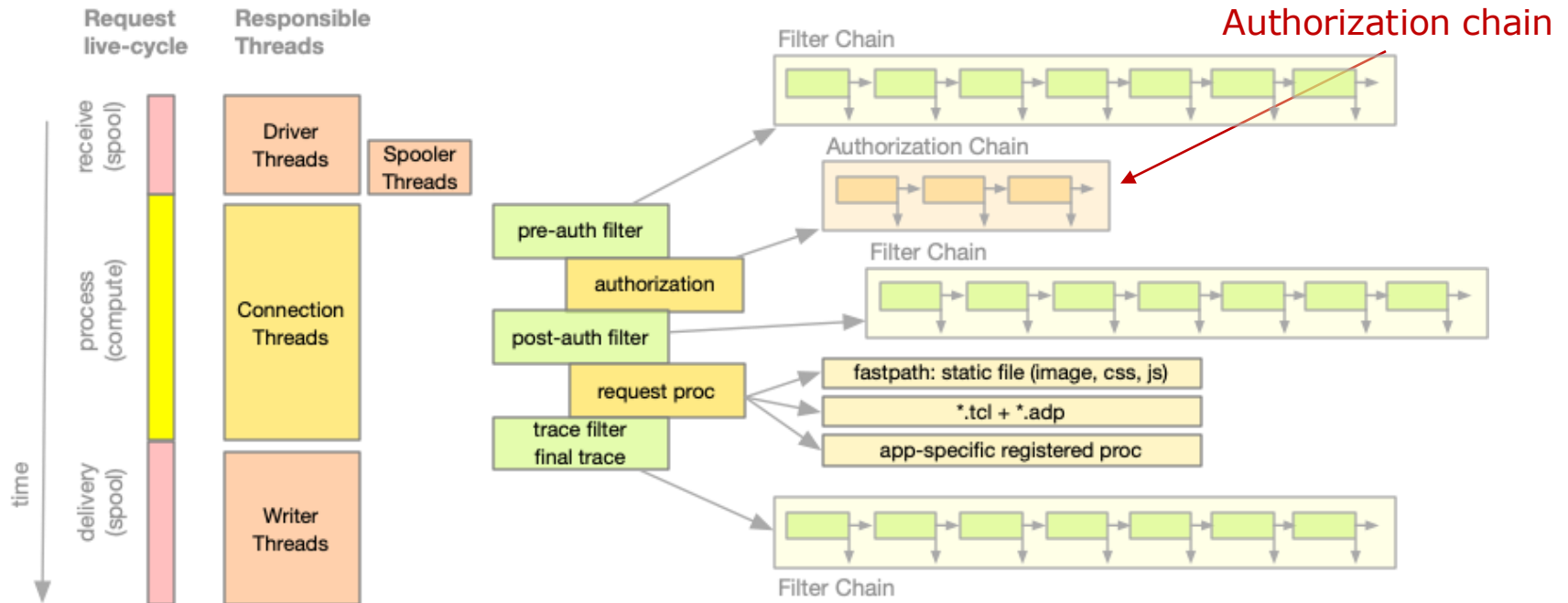


## ■ NaviServer 4.99.\*:

- Registration of **multiple** filters (preauth, postauth, trace) possible
- Registration of a **single** authorization handler
- OpenACS:
  - Not using NaviServer's authorization handler,
  - handling all authorization in filter chains



# Processing Stages for a Single Request in NaviServer 5

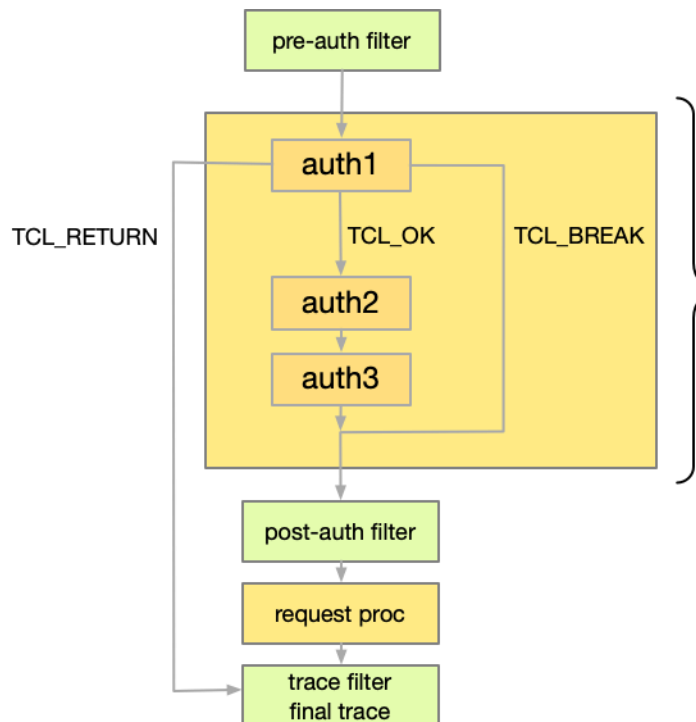


- NaviServer 5
  - Support of **multiple** authorization handlers (authorization chain)
  - Registration (below registration of Tcl authorization callback):

`ns_register_auth ?-authority /value/? ?-first? ?--? request|user /script/ ?/arg .../?`



# Authorization Chain in NaviServer 5



## Result of Authorization Chain

OK - access granted

FORBIDDEN - access denied, no possible retry

UNAUTHORIZED - authentication required or failed, retry possible

- 2 types of results:
  - Signal continuation of processing flow (using TCL\_OK, TCL\_BREAK, TCL\_RETURN)
  - Result of authorization (function result)

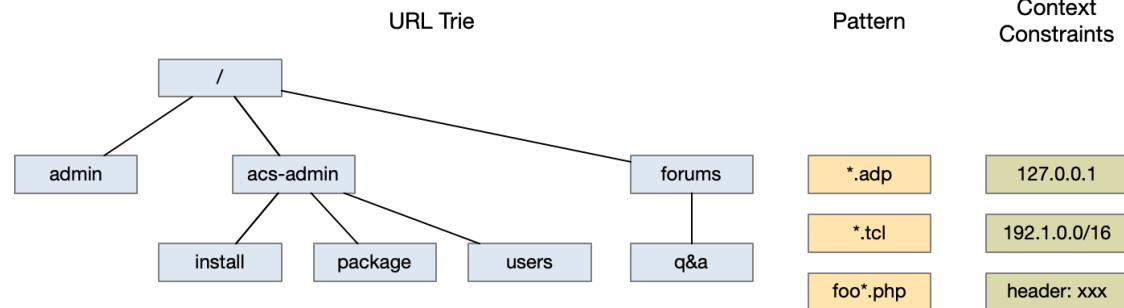
# Context Constraints for URL Tries

## ■ URL Tries in NaviServer 4.99:

- Resolve on **METHOD**, **path** and **pattern** (only last part)
- Used for
  - Request handlers (`ns_register_proc`, `ns_register_adp`, `ns_register_tcl`, ...),
  - mapping of requests to pools
  - `ns_urlspace` command, in modules like `nscgi`, `nsperm`

## ■ Examples:

```
ns_urlspace set /*.foo x
ns_register_tcl GET /*.tcl
ns_register_cgi GET /cgi/*.php
```



## ■ NaviServer 5:

- Resolve on **METHOD**, **path**, **pattern** and **request context** (Context constrains on IP, headers)
- Match inside path possible
- Context constrains can be used whenever URL Trie is used and for `ns_register_filter`  
E.g. allow reverse proxy only for 127.208.0.0/16 requests

```
ns_register_proc -constraints {X-ns-ip 127.208.0.0/16} GET /admin/* { ... }
ns_register_cgi -constraints {X-User *admin*} GET /admin/Panel/*
```

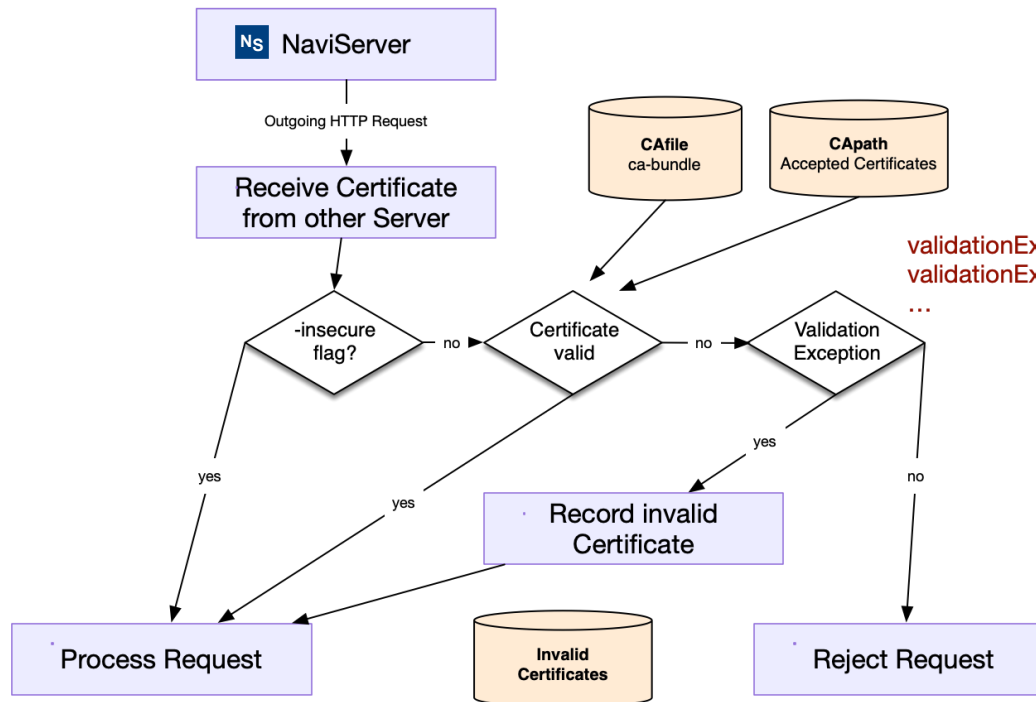
# NaviServer as HTTP Client (1/4)

## Certificate Management

- Use of external Web-Services nowadays increasingly important
- Security by default -> outgoing HTTPS request might fail  
-> handling of certification violations



Security by Default



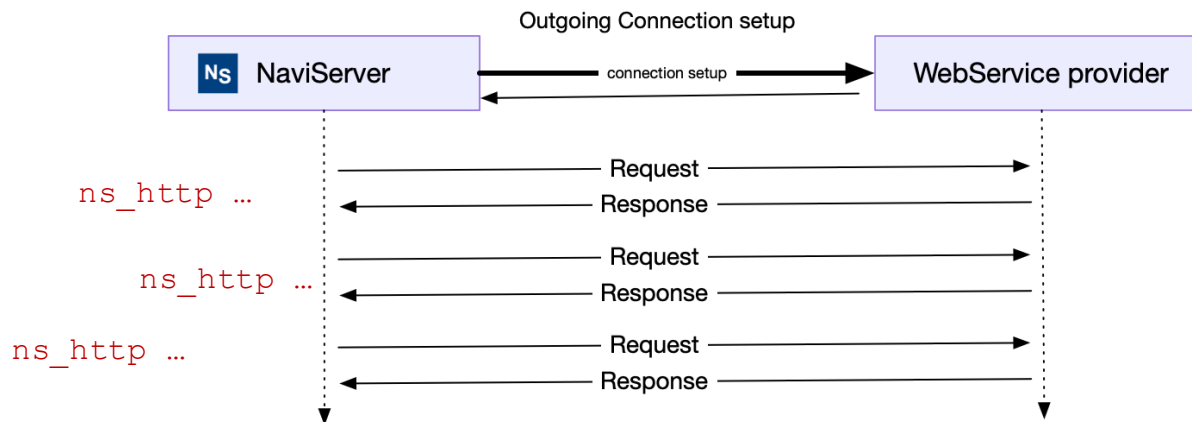
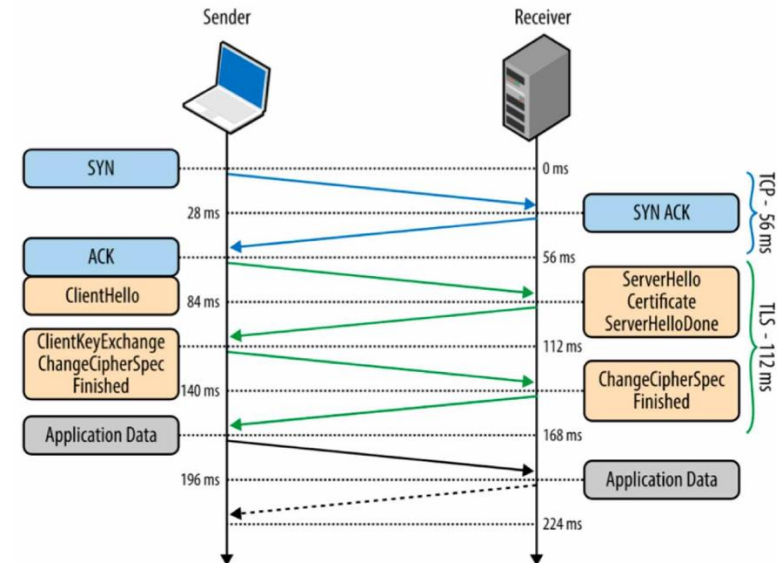
### Sysadmins

- Can define exception rules
- Can review rejected certificates for auditing and manual handling

# NaviServer as HTTP Client (2/4)

## Persistent Connections

- Persistent connections for client requests
  - NaviServer makes multiple requests to the same service
  - Connection setup costs are high
  - But also, using a single connection is not optimal
- NaviServer provides
  - Depending on load multiple concurrent reusable connections
  - Detailed statistics

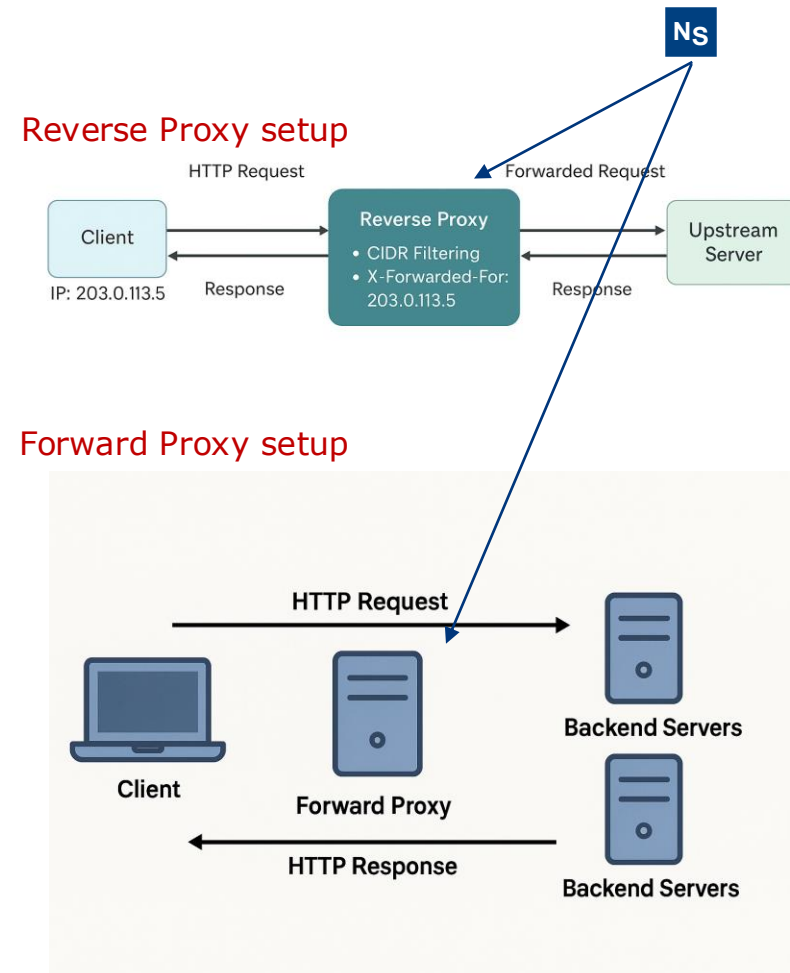


Requests from same or different threads

# NaviServer as HTTP Client (3/4)

## Deployment Scenarios

- Using HTTP client-functionality for
  - Many applications, including now:
    - Reverse proxy
    - Forward proxy
- Make use of HTTP client infrastructure
  - Built-in reverse proxy module with streaming & keep-alive
  - CIDR-based upstream trust & X-Forwarded-For parsing (multiple options)
  - Forward proxy supports HTTP CONNECT (TLS tunneling)
  - Configurable via `ns_http` or `ns_connchan`
  - Fully scriptable
    - E.g. run authorization in NaviServer/OpenACS before forwarding request to upstream server
- Same NaviServer instance can act
  - As reverse proxy for multiple backends
  - As forward proxy
  - As multiple web servers (mass virtual hosting)



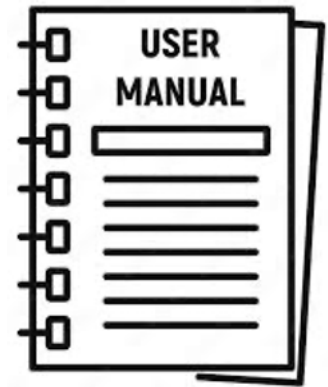
# Substantial Rework of Command Documentaion

## ■ NaviServer 4.99:

- Some commands were not documented
- Some options of commands were not documented
- Some options were documented, that did not exist
- Deprecated commands were used in examples in documentation
- Some commands/subcommands were not included in regression test
- Omissions on full-command-list overview in documentation

## ■ NaviServer 5:

- Generated documentation (using Tcl doctools) is parsed into dicts
- Regression tests are parsed into dicts
- Source code (command registrations) are parsed into dicts
- Compare results -> fixed several hundred problems!



Tested commands not documented

-----> 728 tested commands, 91 deprecated, 655 documented, 0 NOT documented

Documented commands not tested

-----> 655 documented commands, 655 tested, 0 NOT tested

-----> 39 deprecated commands are tested

-----> 0 usage messages differ from doc, 655 are identical

# NaviServer as HTTP Client (4/4)

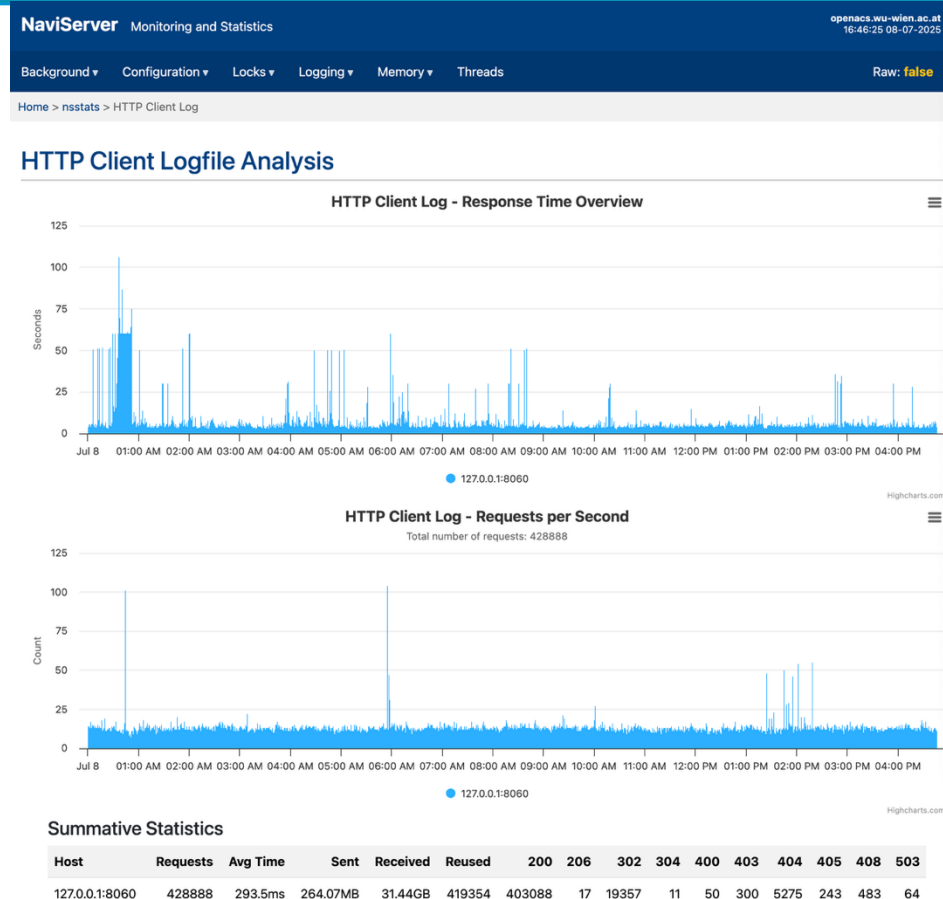
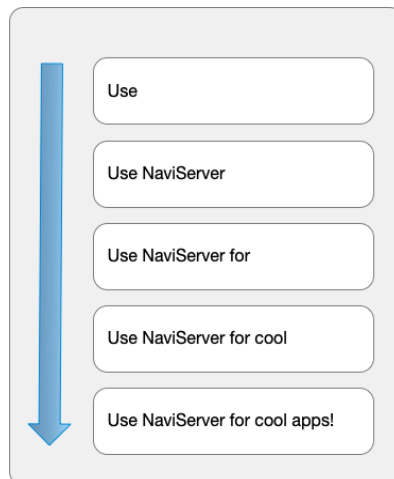
## Response Streaming

- Support for Streaming Client Response processing requests in `ns_http`

- Two new callbacks:
  - `-response_header_callback`
  - `-response_data_callback`

Important for

- Streaming REST calls, where outputs are produced progressively (as used e.g. by generative AI)
- Improving performance and scalability of reverse proxy



July 8 reverse proxy for CVS browser on OpenACS.org:

- Up to 110 revproxy requests per second
- Time cap with 60s
- Avg time 293ms
- 98% reuse, about 0.5 mio requests on this day so far



**WU**  
WIRTSCHAFTS  
UNIVERSITÄT  
WIEN VIENNA  
UNIVERSITY OF  
ECONOMICS  
AND BUSINESS

- NaviServer Programmable Web Server
- # Welcome to NaviServer 5.0.0b1
- Congratulations – your NaviServer instance is up and running on Darwin 24.5.0! This page confirms that your default installation is active.
- ## Getting Started
- To begin using NaviServer, you can:
- Replace this placeholder page with your custom content by configuring the appropriate directory.
  - Review our Documentation for complete setup instructions and feature details.
  - Explore the NaviServer Statistics Module for real-time performance insights and logging analysis.
- ## Current Server Configuration
- **Configuration File:** `/usr/local/na/conf/nsd-config.tcl`
  - **Listening:** `http://0.0.0.0:8080` (Note: Your server may be publicly accessible)
  - **Loaded Modules:** `nscgi, nslog, nspem, nsstats, revproxy`
- ⚠ Security Notice:** For production environments, please replace or secure this default page to protect your server's details.
- ⚠ Security Alert:** The 'nspem' module is installed, but the default system administrator password for 'nsadmin' remains unchanged. Please update the password immediately. [Change Password Now](#)

<h1>NaviServer™ Programmer Web Server</h1>	
<p>[ Keyword Index   Command List   Command List   ]</p>	
<h2>Table Of Contents</h2>	
•	<ul style="list-style-type: none"> <li>NavServer Manual</li> <li>NavServer Core API</li> <li>NavServer CGI Module (nrcgi)</li> <li>NavServer Control Port Module (nrcp)</li> <li>NavServer Database API (nldb)</li> <li>NavServer Access Log API (nrlg)</li> <li>NavServer Permission Module (nrsperm)</li> <li>NavServer Process Proxy Module (nrsproxy)</li> <li>HTTP Network Driver (nrcssd)</li> <li>HTTPS Network Driver (nrssd)</li> <li>NavServer Reverse Proxy Module (nrproxy)</li> </ul>
<h2>NaviServer Manual</h2>	
admin-conf	NavServer Configuration Guide
admin-cfg	NavServer Database Administration Guide
admin-install	NavServer Installation Guide
admin-maintenance	NavServer Maintenance Guide
admin-running	Running NavServer
admin-tuning	NavServer Tuning and Scaling Guide
api-overview	NavServer API Development
cli-drivers	NavServer Database Driver Development Guide

NaviServer Monitoring and Statistics
Middlebrook Pro 201 Installation  
14.36.43.07 07 Jul 2022

[Background](#)
[Configuration](#)
[Locks](#)
[Logging](#)
[Memory](#)
[Threads](#)
**Raw Data**

---

Home > [install](#) > [Process](#)

## Process Information

Key	Value
Host	Middlebrook-Pro-Hit Installation (192.168.1.10 / 8.9.16. ipsec4, Darwin 24.5.0, connected via protocol 0 currenready 127.0.0.1 public 0 failed 0 early 0 early 0 type IPv4 WRITE_RECVING=PROCESSED=REMOVED=ACCEPTED=CAPTURED from client 127.0.0.1
Boot Time	Mon Jul 14 23:42:30 2025
Uptime	7m 0s
Process	77905 ./usr/local/bin/nssm (pid 28.58M vsize 421.94GB fork-time 7ms)
Open Files	67
Home	/usr/local/etc
Configuration	/usr/local/etc/nssm-config.txt
System Log	/usr/local/var/log/nssm
Log Statistics	Notice 114 Warning 1 Error 0 Fail 0 Bug 0 Debug 0 Deprecated 0 Dev 0 Security 1 Debugging(throw) 0 Debug(thread) 0 Debug(lock) 0 Debug(memory) 2 Debug(connection) 0 Debug(resource) 0 Debug(sessions) 0 Debug(internal) 0 Debug(message) 0 Debug(signal) 0
Version	5.0.0.0 [Bug nssmserver-0.000000] [Signal nssmserver-0.000000] [Compiler (17.0.6 clang-1700.1.13.5) 20250515 system "x86_64" with "Optimizations 1" bit 8 & 64]
Build Date	Jul 7 2025 at 14:23:08
Runtime	(info) [My NaviServer Instance]
Driver Info	module http_type request server 0 location 0 address 0.0.0 port 8080 defaulturl 80 protocol Http module ssl increase 30 seconds timeout 0
DB-Pools	Bread http 0 module http received 24 queued 0 partial 11 errors 0

# Improved Configurability

- `ns_configure_variables`

API for conveniently using environment variables in configuration files

- **Motivation**

- Often, configuration files differ only a little
- Useful for:
  - containers, when running e.g. multiple NaviServer instances
  - systemd service files
- Containers are typically configured via environment variables

- **Used in**

- Sample configuration file for plain NaviServer
- Sample configuration file for OpenACS

docker-compose.yml

```
...
environment:
  - NSD_HTTP_PORT=8080
  - NSD_HTTPS_PORT=8443
  - NSD_CERTIFICATE=...
....
```



container

**NS**

NaviServer configuration file:

```
ns_configure_variables -prefix NSD_
→ sets Tcl variable http_port
→ sets Tcl variable https_port
→ sets Tcl variable certificate
....
→ Tcl variable http_port used for ns/module/http/port
→ Tcl variable https_port used for ns/module/https/port
...
```

## ■ NaviServer 5

- Improved security
- Many new features
- Strong backward compatibility
- Major cleanup of API and documentation
- 50 new and extended Tcl commands and subcommands.

Average Code Age: **9.8 years**

based on a line-per-line analysis,  
e.g. 5.11% of the code base is  
unmodified since 2000



... 2000 :	5.11%
... 2001 :	1.60%
... 2002 :	1.54%
... 2003 :	1.44%
... 2004 :	0.14%
... 2005 :	7.30%
... 2006 :	7.84%
... 2007 :	3.28%
... 2008 :	1.84%
... 2009 :	0.21%
... 2010 :	0.10%
... 2011 :	0.12%
... 2012 :	3.06%
... 2013 :	1.60%
... 2014 :	3.60%
... 2015 :	1.88%
... 2016 :	7.21%
... 2017 :	4.25%
... 2018 :	4.70%
... 2019 :	5.40%
... 2020 :	7.17%
... 2021 :	3.62%
... 2022 :	2.92%
... 2023 :	4.25%
... 2024 :	8.90%
... 2025 :	10.93%



VIENNA UNIVERSITY OF  
ECONOMICS AND BUSINESS

**Institute for Information Systems and  
New Media**

Welthandelsplatz 1, 1020 Vienna, Austria

**UNIV.PROF. DR. Gustaf Neumann**

T +43-1-313 36-4671

[gustaf.neumann@wu.ac.at](mailto:gustaf.neumann@wu.ac.at)

[www.wu.ac.at](http://www.wu.ac.at)