Vom Client zum Server

Der Verbindungsaufbau im Detail

MARTIN BERGER

DOAG 2013

Martin Berger

ORACLE DBA seit 2000

(ORACLE Support & Kunden)


http://berxblog.blogspot.com


@martinberx


martin.a.berger@gmail.com

"*it depends*"

Einschränkungen

- Linux
  - eventuell andere UNIX derivate
  - kein Windows

- 11.2
  - kein 12c / threads!

Vom Client zum Server

- bequeath connection
  - Instanz Startup
  - prelim
- Listener
- SCAN Listener

bequeath connection

sqlplus “/ as sysdba”

```
Connected to an idle instance.


SQL> startup
ORACLE instance started
```

```
Starting ORACLE instance (normal)
...
Completed: ALTER DATABASE OPEN
```

Instanz startup

Instanz

ORACLE_HOME
&
ORACLE_SID

Shared memory
&
Semaphoren

```
sysresv

IPC Resources for ORACLE_SID "VAX1" :
Shared Memory:
ID              KEY
97124396        0x00000000
97157165        0x00000000
97189934        0x3b695db0
Semaphores:
ID              KEY
235077710       0x02b6f2d4
235110479       0x02b6f2d5
235143248       0x02b6f2d6
235176017       0x02b6f2d7
235208786       0x02b6f2d8
Oracle Instance alive for sid "VAX1"
```

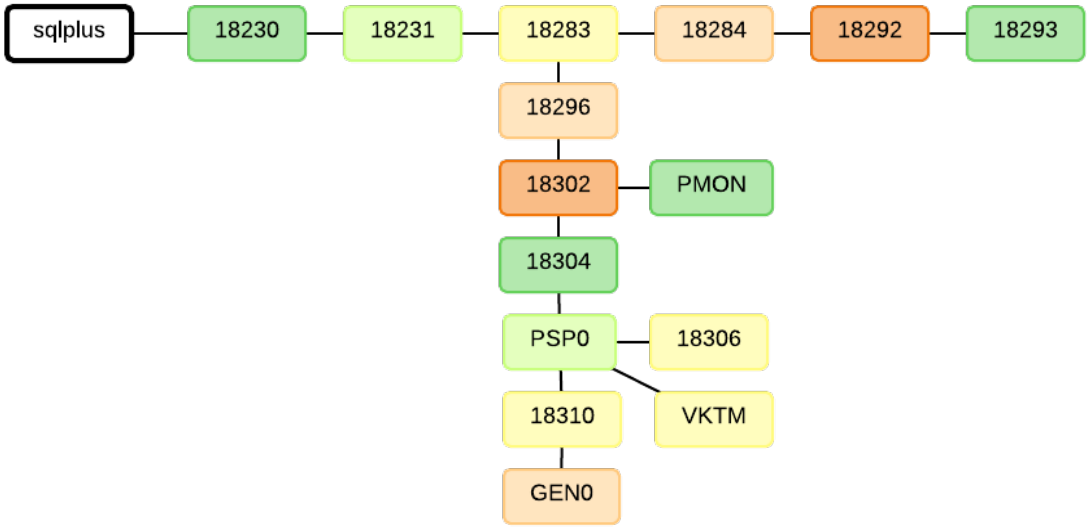
Instanz startup

strace

```
18230 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2b4943c40b80) = 18231
18231 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2afb7e669ec0) = 18283
18283 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2afb7e669ec0) = 18284
18284 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2ab005560ec0) = 18292
18292 clone( <unfinished ...>
18292 <... clone resumed> child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x2ab005560ec0) = 18293
18231 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2afb7e669ec0) = 18296
18231 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2afb7e669ec0) = 18302
18302 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2afb7e669ec0) = 18303
18231 clone( <unfinished ...>
18231 <... clone resumed> child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x2afb7e669ec0) = 18304
18304 clone( <unfinished ...>
18304 <... clone resumed> child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x2afb7e669ec0) = 18305
18305 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2b9d095f1ec0) = 18306
18306 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2b9d095f1ec0) = 18307
18307 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2b4038d2cec0) = 18308
18308 clone( <unfinished ...>
```

Instanz startup

strace



„PSP0 (process spawner) spawns Oracle processes.“

Instanz startup

Konsequenzen

Alle Kernelprozesse Erben von ursprünglichem sqlplus

- Umgebungsvariablen
- Systemlimits (ulimit)
- user/groups

Zu identifizieren in /proc

- /proc/<pid>/environ
- /proc/<pid>/limits
- /proc/<pid>/status

(leider teilw. nur als root lesbar)

Instanz startup

“`sqlplus /`” bei laufender Instanz

- muss sich “nur” zu existierenden shared memory & semaphoren verbinden.
- hat eigene Umgebungsvariablen, limits, groups, etc

bequeath connection

“`sqlplus /`” bei laufender Instanz

- muss sich “nur” zu existierenden shared memory & semaphoren verbinden.
- hat eigene Umgebungsvariablen, limits, groups, etc

prelim

```
sqlplus /prelim „/ as sysdba“
```

- Verbindet sich zu shared memory & semaphoren
- keine session!
- auch sonst keine registrierung in der SGA (z.b. v$process)
- keine SQL statemens
- aber z.b. oradebug:

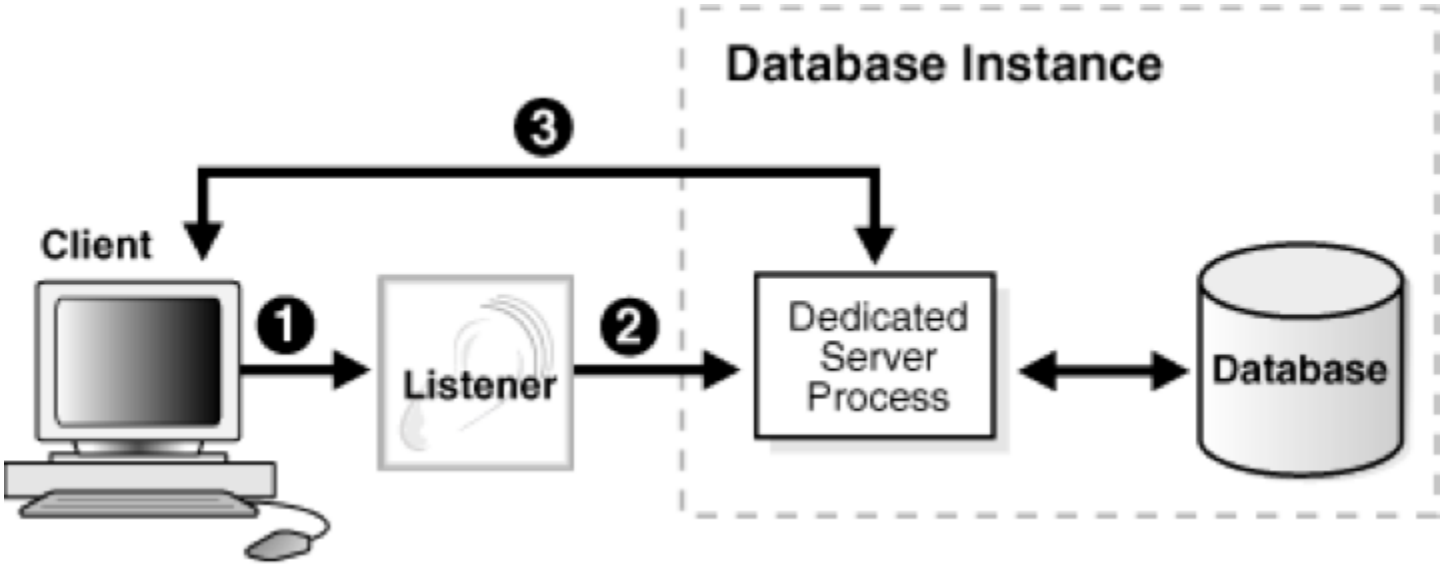
```
ORADEBUG SETOSPID OS_PID
ORADEBUG DUMP HANGANALYZE 3
```

bequeath connection

Fragen ?

Listener

Beschreibung von Oracle:



Listener

strace

[pid 2979] clone(Process 27028 attached

child_stackm0, flags=CLONE_CHILD_CLEARTIDICLONE_CHILD_SETTIDISIGCHLD, child tidptr=0x2aedd9914680) = 27028

[pid 2979] wait4(27028, Process 2979 suspended <unfinished ...>

[pid 27028] clone(Process 27029 attached (waiting for parent)

Process 27029 resumed (parent 27028 ready) child stackm0, flagsg•CLONE_CHILD CLEARTIDICLONE_CHILD_SETTIDISIGCHLD, child_tidptrmOx2aedd9914b80) = 27029

[pid 27028] exit_group(0) =? Process 2979 resumed Process 27028 detached

[pid 2979] <... wait4 resumed> [{WIFEXITED(s) && WEXITSTATUS(s) == 0)), 0, NULL) m 27028

[pid 27029] close(15 <unfinished ...> [

pid 2979] --- SIGULD (Child exited) @ 0 (0) ---

[pid 27029] c... close resumed>) m 0

[pid 2979] close(14 <unfinished ...>

[pid 27029] close(16 <unfinished ...>

[pid 2979] <... close resumed>) = 0

[pid 27029] <... close resumed>) = 0

[pid 2979] close(17) m 0

[pid 27029] execve(“/appl/oracle/product/rdbms_112022_a/bin/oracle", [“oracleTTT051", "(LOCAL=NO)"], [14, 109 vars */]) = 0

Listener

strace - details

```
[pid 2979] clone(Process 27028 attached
child_stackm0, flags=CLONE_CHILD_CLEARTIDICLONE_CHILD_SET
[pid 2979] wait4(27028, Process 2979 suspended <unfinished ...>
[pid 27028] clone(Process 27029 attached (waiting for parent)
Process 27029 resumed (parent 27028 ready) child stackm0, flagsg•CL(
child_tidptrmOx2aedd9914b80) = 27029
[pid 27028] exit_group(0) =? Process 2979 resumed Process 27028 det
[pid 2979] <... wait4 resumed> [{WIFEXITED(s) && WEXITSTATUS(s) =
[pid 27029] close(15 <unfinished ...> [
pid 2979] --- SIGULD (Child exited) @ 0 (0) ---
[pid 27029] c... close resumed> ) m 0
[pid 2979] close(14 <unfinished ...>
[pid 27029] close(16 <unfinished ...>
[pid 2979] <... close resumed> ) = 0
[pid 27029] <... close resumed> ) = 0
[pid 2979] close(17) m 0
[pid 27029] execve("/appl/oracle/product/rdbms_112022_a/bin/oracle", ['
```

ein neuer Prozess wird erzeugt.

mit vollkommen neuem Programm binaries

execve() executes the program pointed to by *filename*. *filename* must be either a binary executable, or a script starting with a line of the form:

execve() does not return on success, and the text, data, bss, and stack of the calling process are overwritten by that of the program loaded.

By default, file descriptors remain open across an **execve**()

Listener

Fragen ?

SCAN Listener

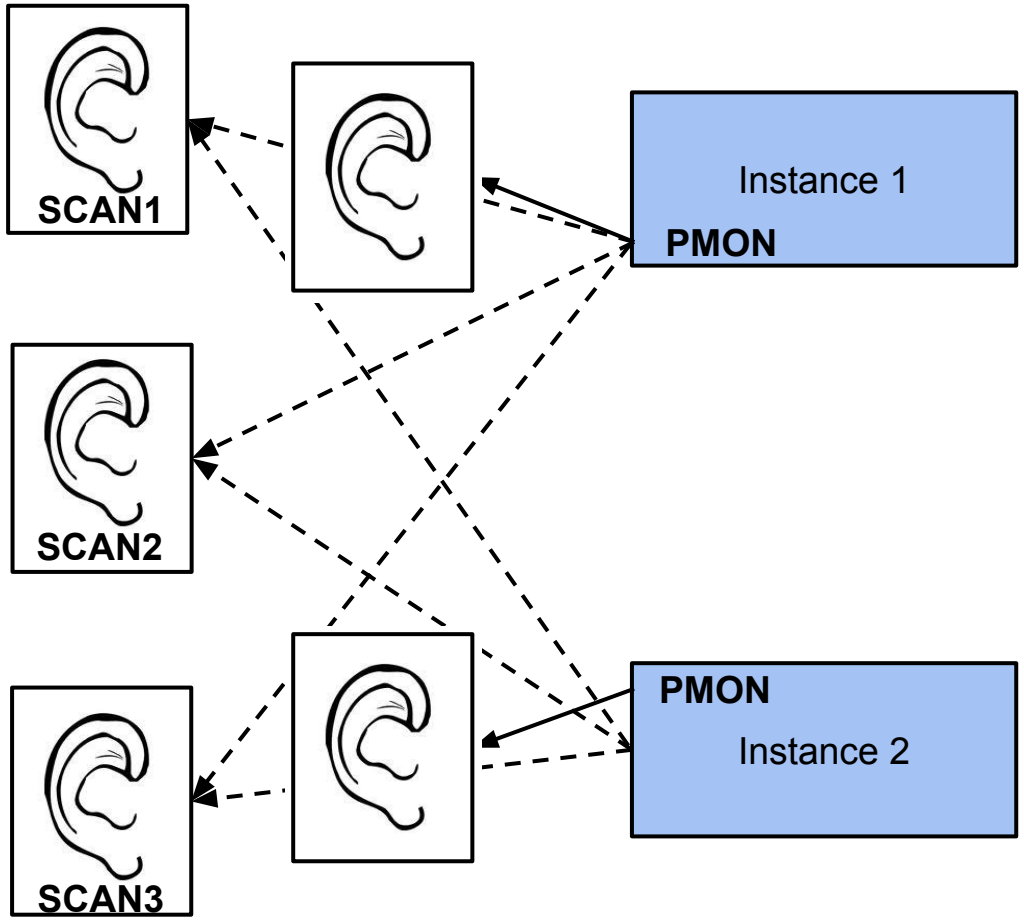
„Single Client Access Name“ (SCAN)

- Vereinfacht die Client Konfiguration
- nur einen HOSTS Eintrag
- Dynamisches Verändern der Cluster Nodes

funktioniert über die bekannten Parameter `local_listener` und `remote_listener`

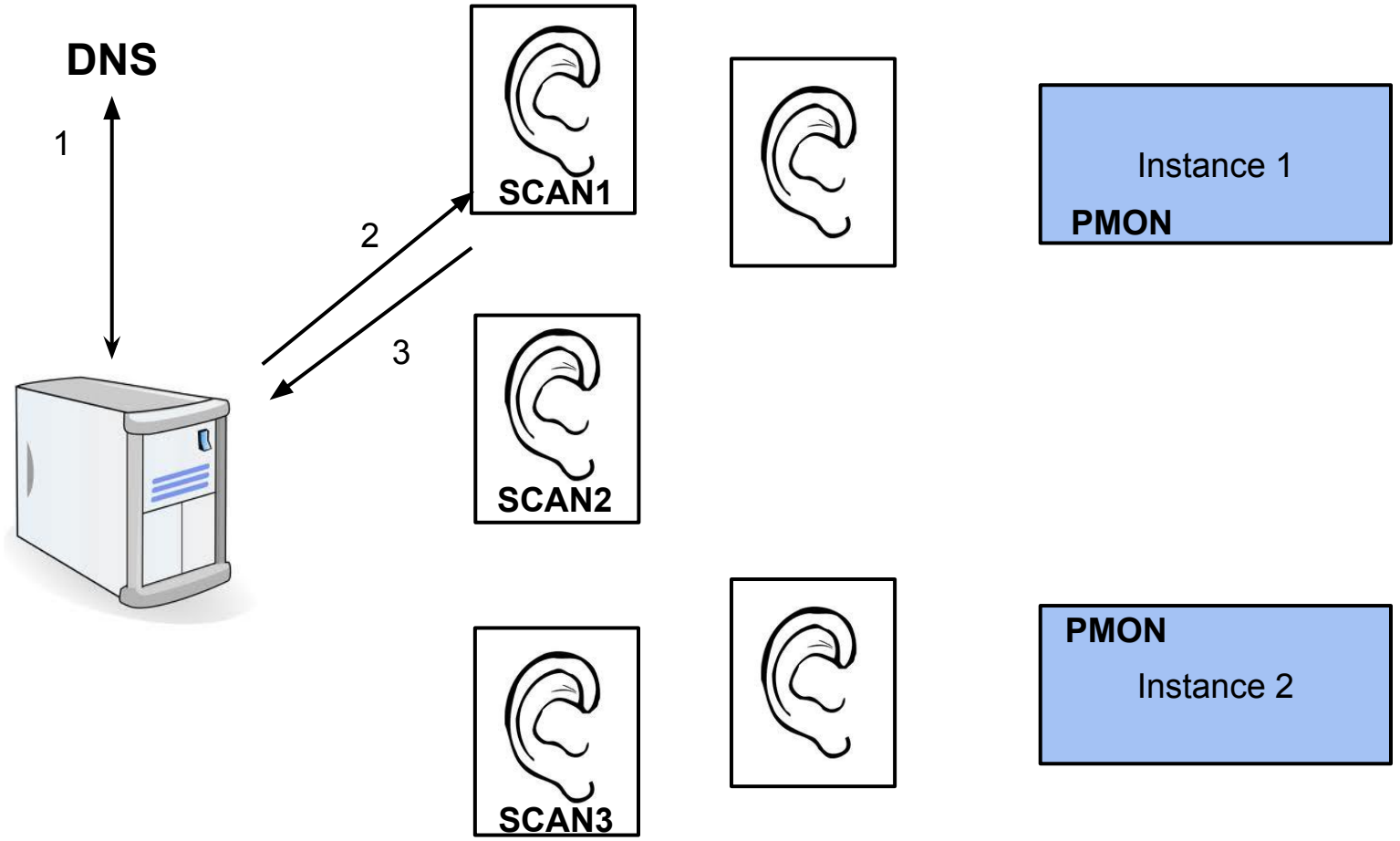
SCAN Listener

pmon registriert die Instanz bei den Listenern



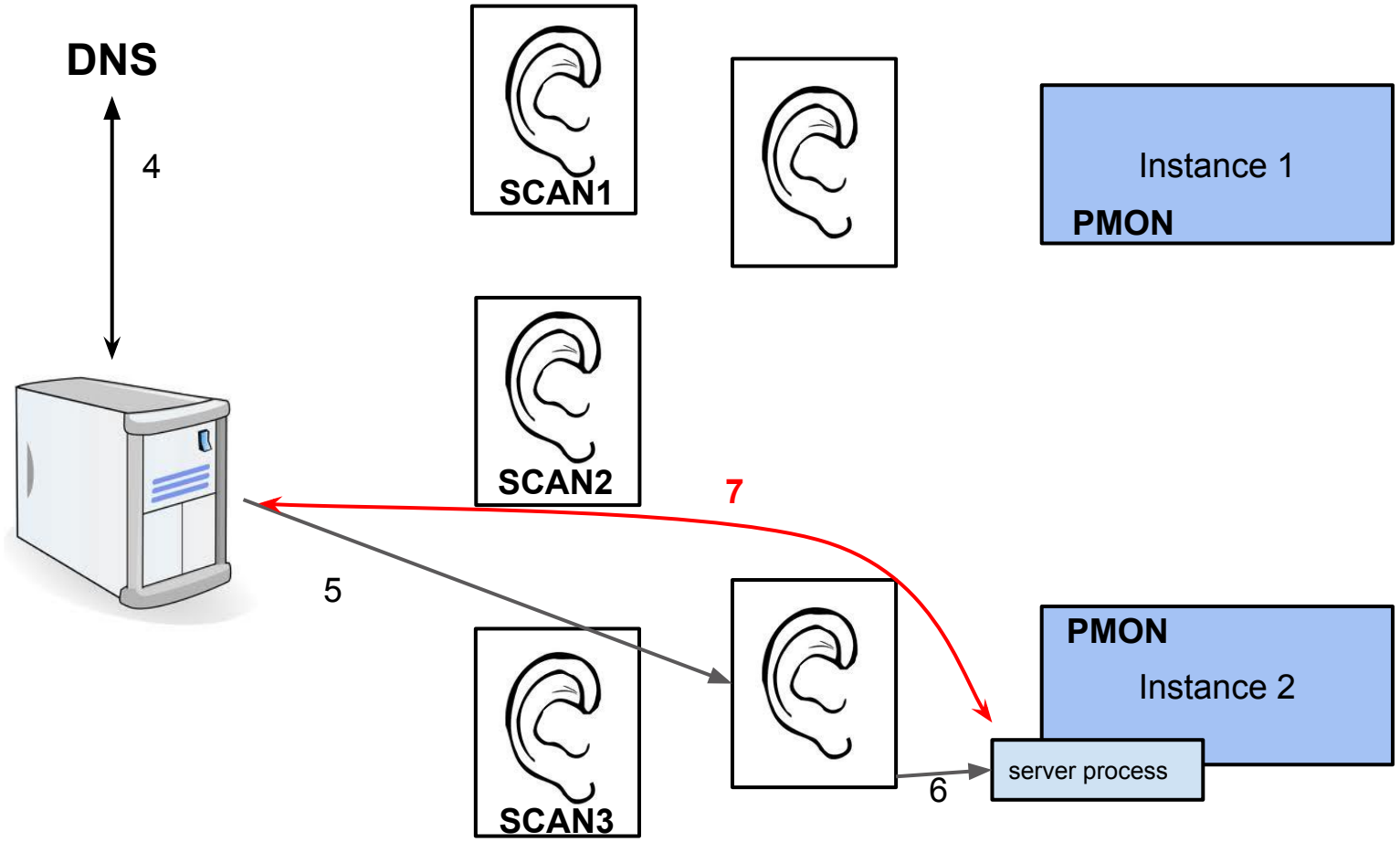
SCAN Listener

client verbindet sich zu SCAN Listener



SCAN Listener

client verbindet sich zu VIP Listener



SCAN Listener

Fragen ?

Vom Client zum Server

- bequeath connection
  - Instanz Startup
  - prelim
- Listener
- SCAN Listener