Uncomplicated monitoring for small environments

Ben Fuhrmannek <ben@fuhrmannek.de>

About me

Professional Life

- Ben Fuhrmannek
- Computer Scientist
- Software Development:
  Perl, Erlang, Python, C, Java, Tcl...
- 6 Years of Information Security
- SektionEins → Web- & Mobile Security, Infrastructure Security, Secure Development, Architecture review, Training

Spare time
(IT related)

- eventphone → Telephony software + VoIP installations for hacker events

- Open Source development (see next slide)
- Amateur radio operator (DH4BE)

Other Tcl-Projects

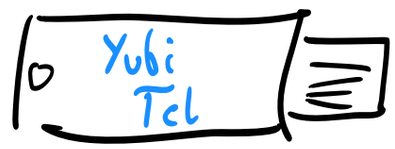
- apachesubst:
  Trivial template system for apache httpd configuration files
  https://github.com/bef/apachesubst

- yate-tcl: Tcl Library and Applications for
  the Yate Telephony Engine
  https://github.com/bef/yate-tcl

- debrepo: debian repository creator
  https://github.com/bef/debrepo

- tcl-escpos: Tcl library for ESC/POS compatible receipt printers
  http://code.google.com/p/tcl-escpos/

- yubi-tcl: Yubikey-compatible validation server and client
  for OTP validation
  http://code.google.com/p/yubi-tcl/



TOC

- Idea / Motivation
- QMON Architecture Overview
- Code: Tcl 8.5 compatibility
- Code: Command Execution
- sqlite Backend
- Status Output (CGI + CLI)
- INI-style Configuration
- Time for Questions + Suggestions

QMON: Motivation

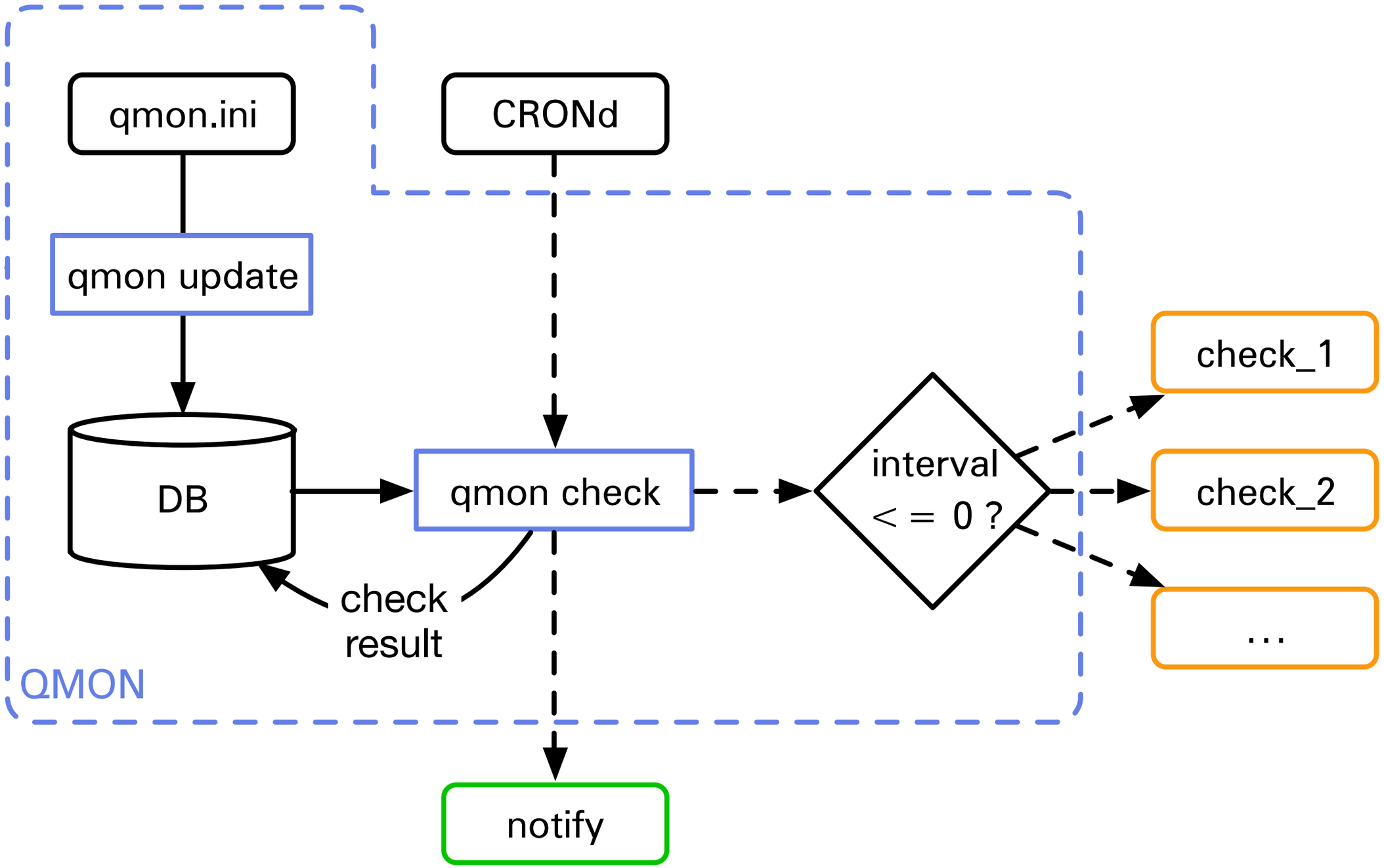
other open source monitoring systems:

- too big and complicated
- difficult to configure (web interface, users, roles, huge databases, ...)
- too many features
  → security concerns

instead I wanted:

- quick protocol-specific checks
- not to reinvent the wheel
  → nagios check compatibility
- easy file based configuration
  → for me 'easy' means "vim foo.ini" and be done with it in no time at all
- alert notification
- lightweight application
  → should be able to run on Raspberry PI or equiv.
- no daemon → cron based invocation
- nice overview page (optional)

QMON Architecture Overview



Tcl 8.5 compatibility (close enough)

```
if {[info command lmap] eq ""} {
   proc lmap {varlist list body} {
      set i 0
      set varlist2 {}
      foreach var $varlist {
         upvar 1 $var var[incr i]
         lappend varlist2 var$i
      }
      set res {}
      foreach $varlist2 $list {
         lappend res [uplevel 1 $body]
      }
      set res
   }
}
```

Command Execution

```
set code 0
try {
    set fd [open "| $cmd"]
    set output [read $fd]
    close $fd
} trap {CHILDSTATUS} {errmsg eropts} {
    lassign $::errorCode - pid code
} on error {errmsg eropts} {
    ## e.g. command not found
    set code 3
    set output $errmsg
}

switch $code {
    0 {set status ok}
    1 {set status warning}
    2 {set status critical}
    3 {set status unknown}
}
```

Backend: sqlite

```
CREATE TABLE IF NOT EXISTS checks (
   name TEXT PRIMARY KEY,
   cmd TEXT,
   interval NUMERIC,
   enabled INTEGER,
   host TEXT,
   desc TEXT,
   status TEXT,
   output TEXT,
   perfdata TEXT,
   last_check NUMERIC
);
```

Status output (CLI)

```
$ ./qmon status
* [     ok] na/na_ssh                          2014-01-22 16:33:43
SSH OK - OpenSSH_6.0p1 Debian-4 (protocol 2.0) |
time=0.060938s;;;0.000000;10.000000
* [critical] unknown/foo_critical              2014-01-22 16:33:43
CRITICAL
  |
* [     ok] unknown/foo_ok                     2014-01-22 16:33:43
OK
  |
* [unknown] unknown/foo_unknown               2014-01-22 16:33:43
UNKNOWN
  |
* [warning] unknown/foo_warning               2014-01-22 16:33:43
WARNING
  |
* [     ok] unknown/fuhrmannek.de_http         2014-01-22 16:33:42
HTTP OK: HTTP/1.1 200 OK - 2999 bytes in 0.142 second response
time | time=0.141518s;;;0.000000 size=2999B;;;0
```

note the colour output

Status output (CGI)

QMON Status

Service Detail

ok: 3 | warning: 1 | critical: 1 | unknown: 1 | new: 0

Host	Status	Description	Last Check	Interval	Output
na	ok	SSH	2014-01-22 16:33:43	3600s	SSH OK - OpenSSH_6.0p1 Debian-4 (protocol 2.0)
unknown	critical	foo_critical	2014-01-22 16:33:43	3600s	CRITICAL
unknown	ok	foo_ok	2014-01-22 16:33:43	3600s	OK
unknown	unknown	foo_unknown	2014-01-22 16:33:43	3600s	UNKNOWN
unknown	warning	foo_warning	2014-01-22 16:33:43	3600s	WARNING
unknown	ok	fuhrmannek.de_http	2014-01-22 16:33:42	3600s	HTTP OK: HTTP/1.1 200 OK - 2999 bytes in 0.142 second response time

INI-Style Configuration (Example)

```
[alpha]
type=host
hostname=alpha.foo.bar
ip=192.168.2.1
desc=first host

[alpha_ssh]
type=check
host=alpha
desc=SSH
cmd=check_ssh -H $cfg(alpha.hostname) -4
; enabled=0
```

INI-Style Configuration w/ Redundancy

```
[alpha]
type=host
hostname=alpha.foo.bar
ip=192.168.2.1

[bravo]
type=host
hostname=bravo.foo.bar
ip=192.168.2.2

[charlie]
type=host
hostname=charlie.foo.bar
ip=192.168.2.3

;...
```

INI-Style Configuration w/ Templates

```
#template stdhost %HOSTNAME% %IP%
[%HOSTNAME%]
type=host
hostname=%HOSTNAME%.foo.bar
ip=%IP%
#end template

#use stdhost alpha 192.168.2.1
#use stdhost bravo 192.168.2.2
#use stdhost charlie 192.168.2.3
```

```
[alpha]
type=host
hostname=alpha.foo.bar
ip=192.168.2.1

[bravo]
type=host
hostname=bravo.foo.bar
ip=192.168.2.2

[charlie]
type=host
hostname=charlie.foo.bar
ip=192.168.2.3

;...
```

Questions? Suggestions?

QMON -

Uncomplicated monitoring for small environments

https://github.com/bef/qmon