

Libevent

- Abstraktionslayer für die “schnellen” Event-Mechanismen (epoll, kqueue, /dev/poll) (!= “standard” event loop)
- “*libevent* is meant to replace the event loop found in event driven network servers. An application just needs to call `event_dispatch()` and then add or remove events dynamically without having to change the event loop.” (<http://libevent.org/>)
- Grundlage für z.B. Memcached, Node.js, Chromium, Tor, ttdnsd
- Immer noch sehr verbose:
 - <https://github.com/libevent/libevent/blob/master/sample/hello-world.c> 141 Zeilen
 - <https://github.com/libevent/libevent/blob/master/sample/http-server.c> 415 Zeilen

Twisted Python

- “Abgehangenes” Framework (Python) für Netzwerkprogrammierung
- Alt: http://onlamp.com/pub/a/python/2004/01/15/twisted_intro.html
 - “Twisted is an attempt to build a framework capable of supporting the needs of modern network applications [...] It provides:
 - Multiple levels of abstraction
 - Cross platform (support while still allowing platform-specific code)

- "Batteries included." (→ Verweis: später)
- Separation of concerns: Protocol developer / Application developer / Authorization developer / Deployment
- <http://mumak.net/stuff/twisted-intro.html> In Twisted:
 - transport: read/write bytes from/to the wire
 - protocol: state machine that responds to events like `dataReceived`, `connectionMade`, `connectionLost`
 - protocol factory: bind protocol objects to their transports on new connections
- `twisted.protocols`: These implementations do not enforce policies (e.g., the FTP code is not limited to serving files from the filesystem), allowing maximum developer flexibility. The supported protocols include HTTP, FTP, DNS, SMTP, POP3, IMAP, SSH, SIP, Jabber, TOC and Oscar (AIM), MSN, IRC, NNTP, and others.
- Twisted also provides higher-level frameworks
 - `twisted.web` package
 - `twisted.names`, a DNS server framework.
- `twisted.internet`: networking event loop
 - Reactor + Deferreds (+ Generators)
 - "Callbacks can be messy"
 - Add error handling?
 - Pass the result around?
 - Cancel the original request?
 - Consistent API?

Beispiele:

- <http://convergence.io/>
“An agile, distributed, and secure strategy for replacing Certificate Authorities”
- “Programming” a caching DNS server in Twisted: 0 Zeilen
 - `$ sudo twisted -n dns --recursive -cache`
- DNS Server mit DHT Backend (79 Zeilen)
<http://stackoverflow.com/questions/4399512/python-dns-server-with-custom-backend>
- Mehr Code angucken?
 - `upperserver.py`
 - `proxy3_exercise_2a.py`

Gute Tutorials:

- <http://twistedmatrix.com/documents/current/core/howto/tutorial/index.html>
 - Twisted from Scratch, or The Evolution of Finger.
- http://www.youtube.com/watch?v=GuNwewBU_Hw
 - Video zweier Sessions auf der EuroPython 2012 (Session 1 Server, Session 2 Client)

Setup auf Ubuntu:

```
sudo apt-get install python-virtualenv python-pip
virtualenv py_env
source py_env/bin/activate
pip install twisted
pip install watchdog
watchmedo auto-restart python upperserver.py
```
