

# User Space Filesysteme mit FUSE

Mirko Dziadzka  
mirko.dziadzka@gmail.com

MUCOSUG 2012-01-19

# Wer bin ich?

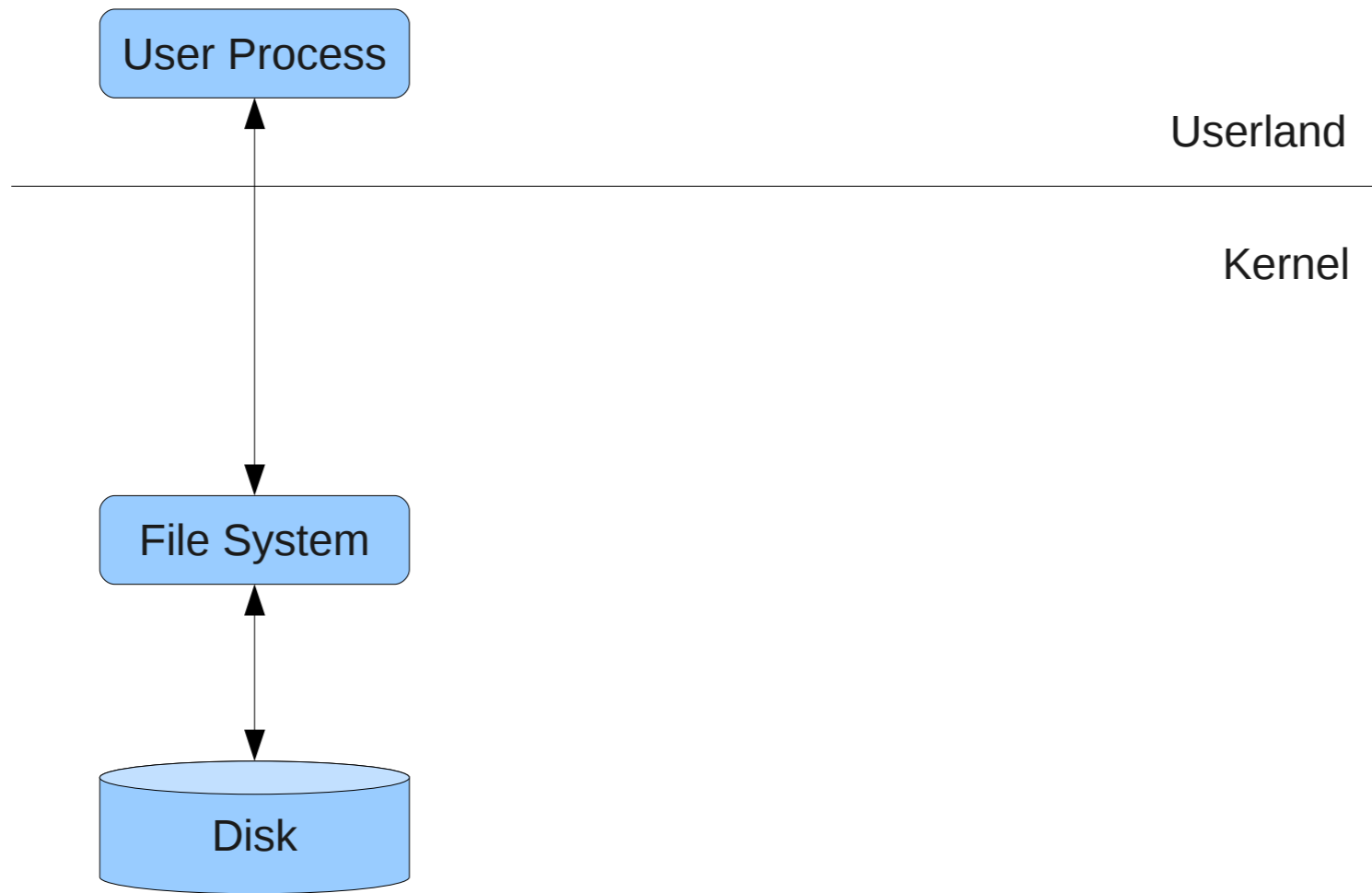
- Studium Mathe / Informatik
- Seit 15+ Jahren im Umfeld
  - Architektur - Entwicklung - Betrieb
  - Unix - Internet - Security - Scalability
- Zur Zeit bei Riverbed / Regensburg
- Web Application Firewall

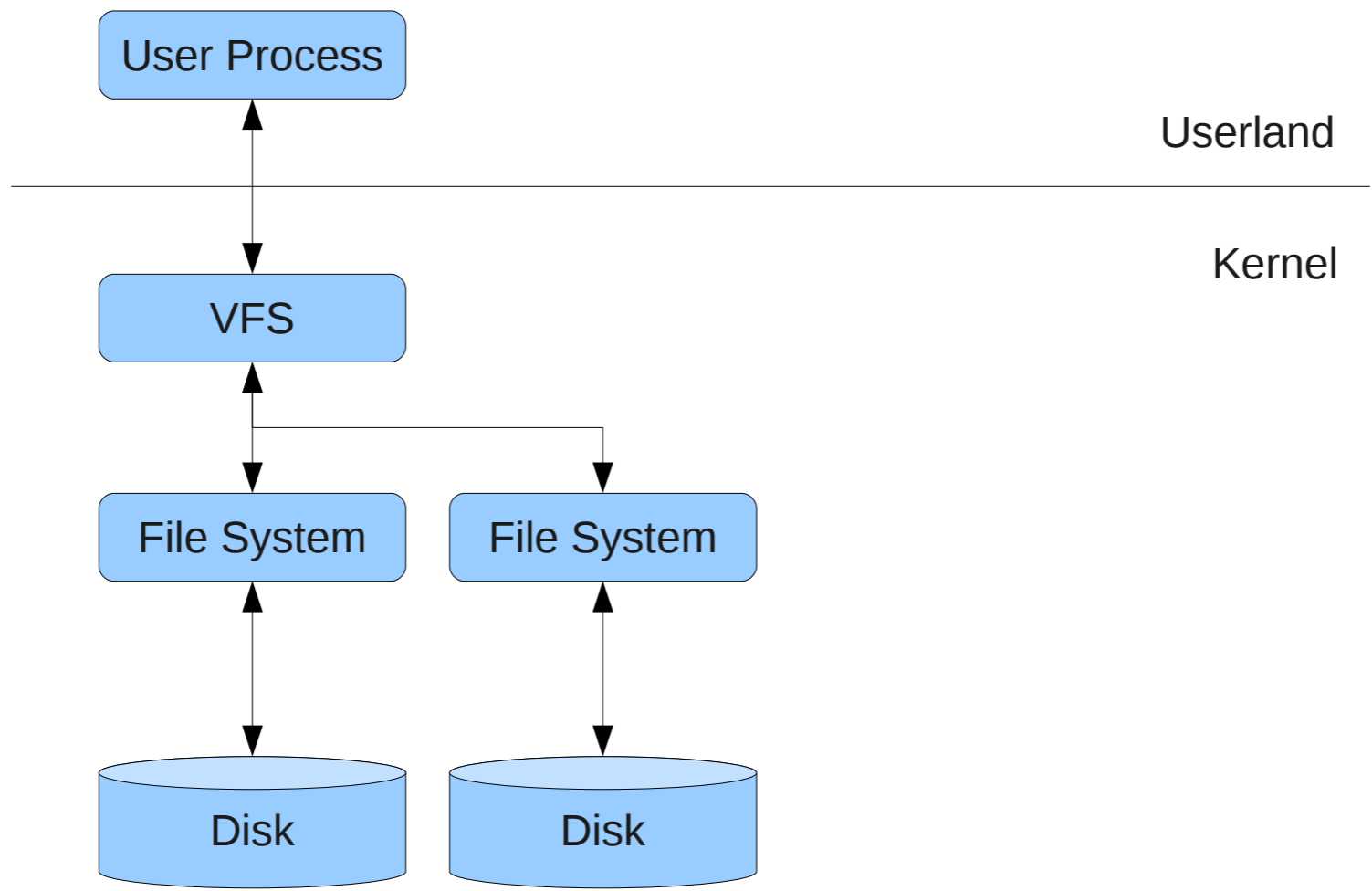
# Worum geht's heute?

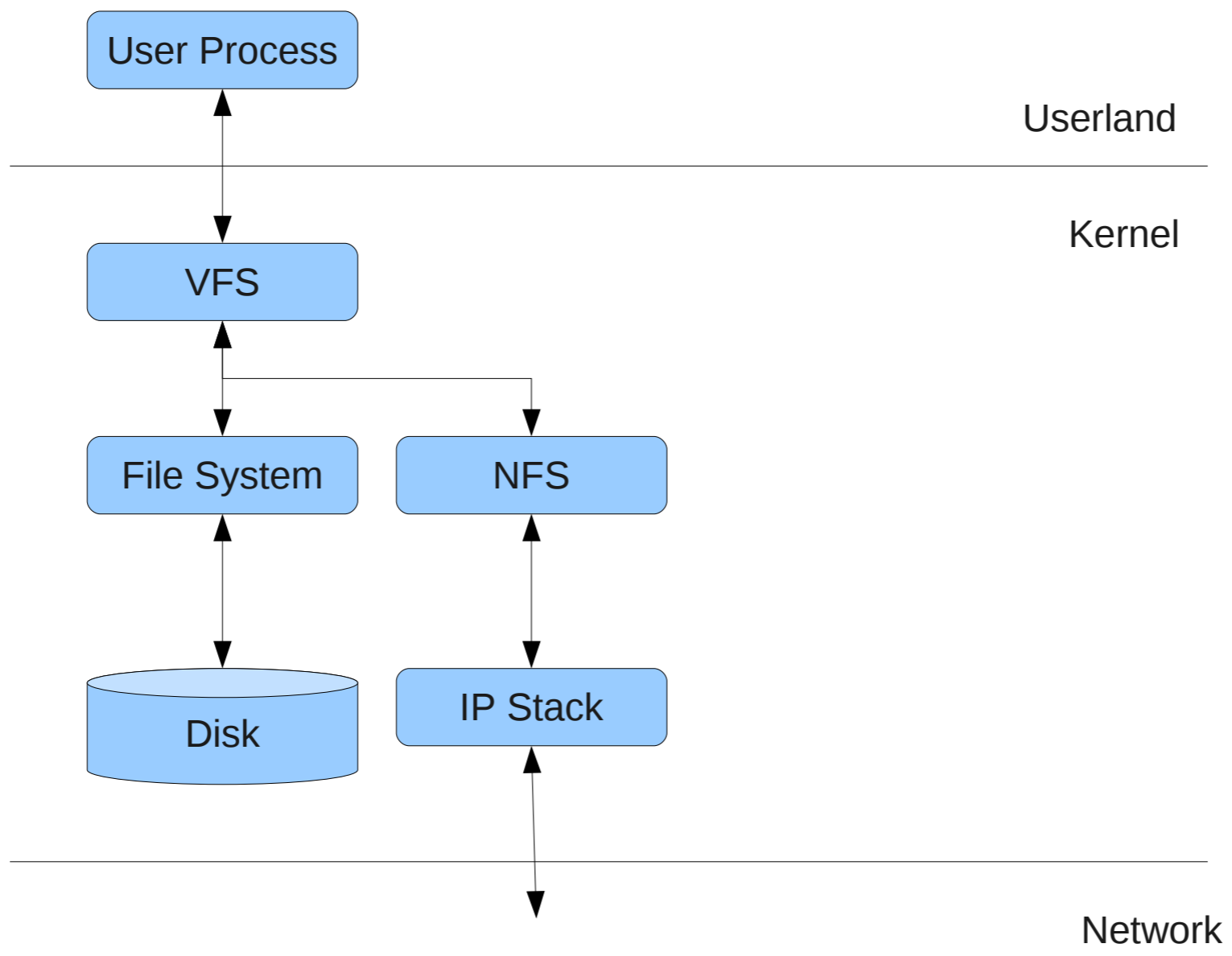
- User Space Filesystem mit Fuse
- Wo kommt das her?
- Warum will man das?
- Wo sind die Grenzen
- Wie mache ich mir ein Filesystem mit Python

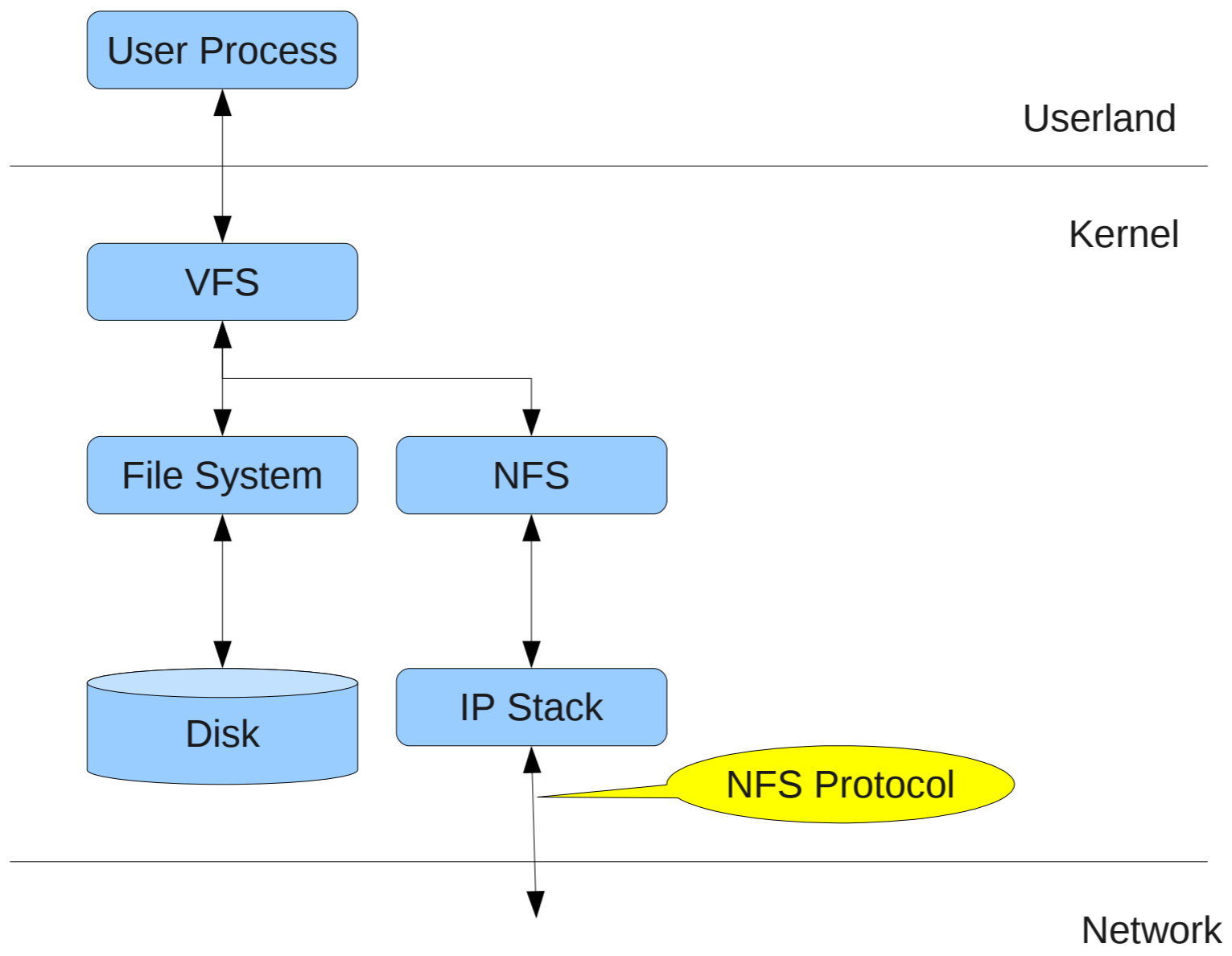
# Geschichte

Eine sehr subjektive Sicht



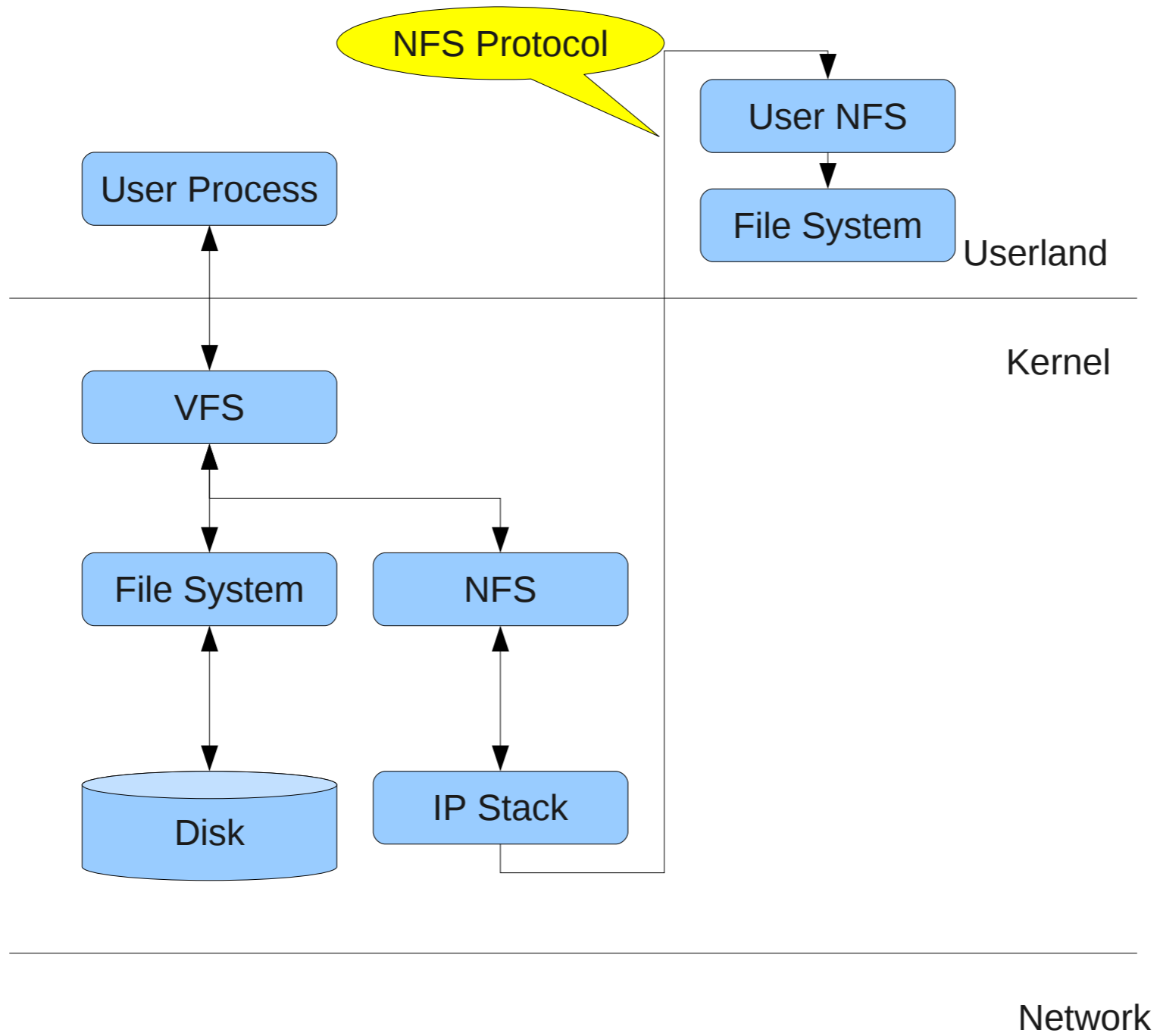








Wo ein Protokoll ist,  
kann man es auch nutzen

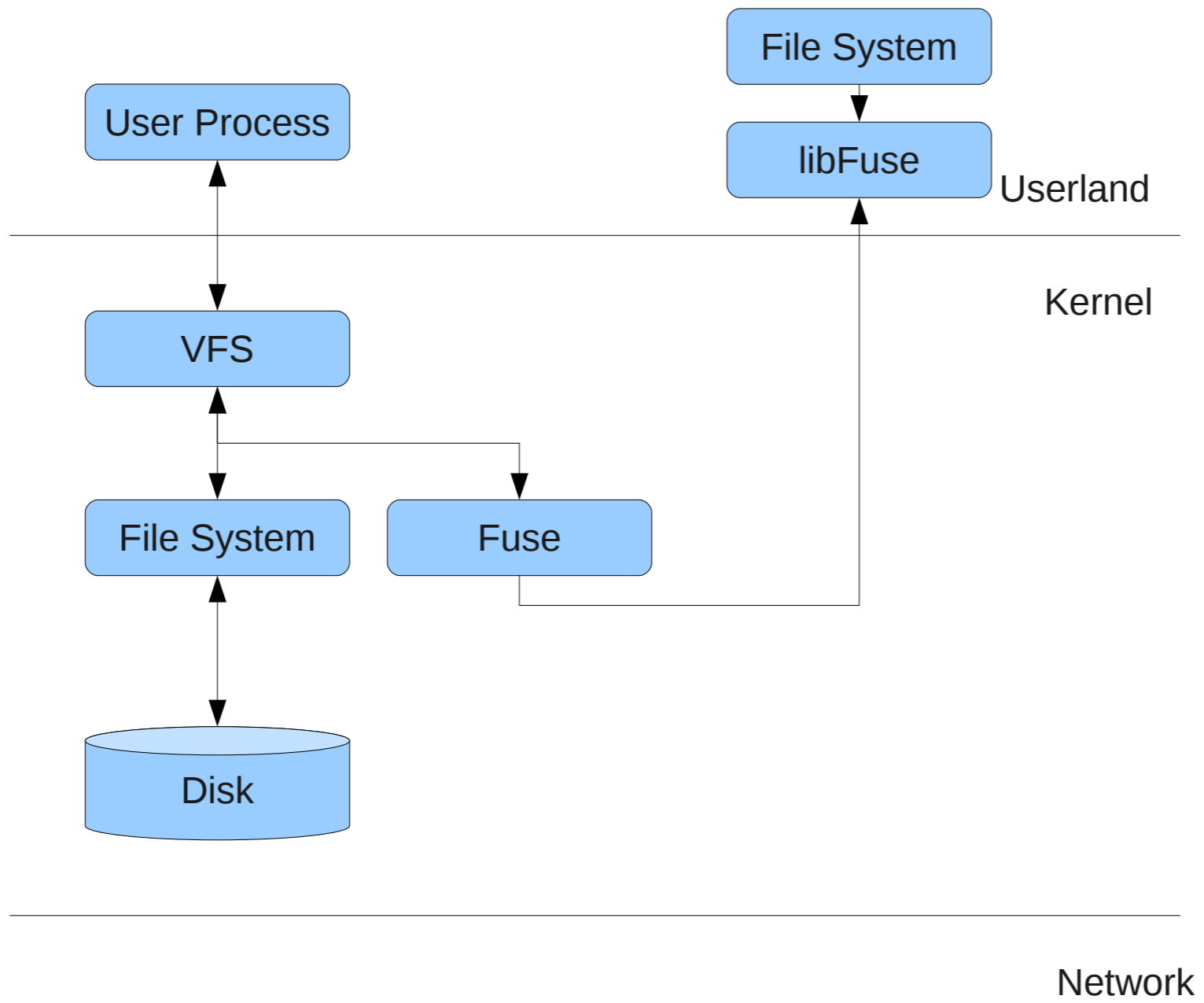


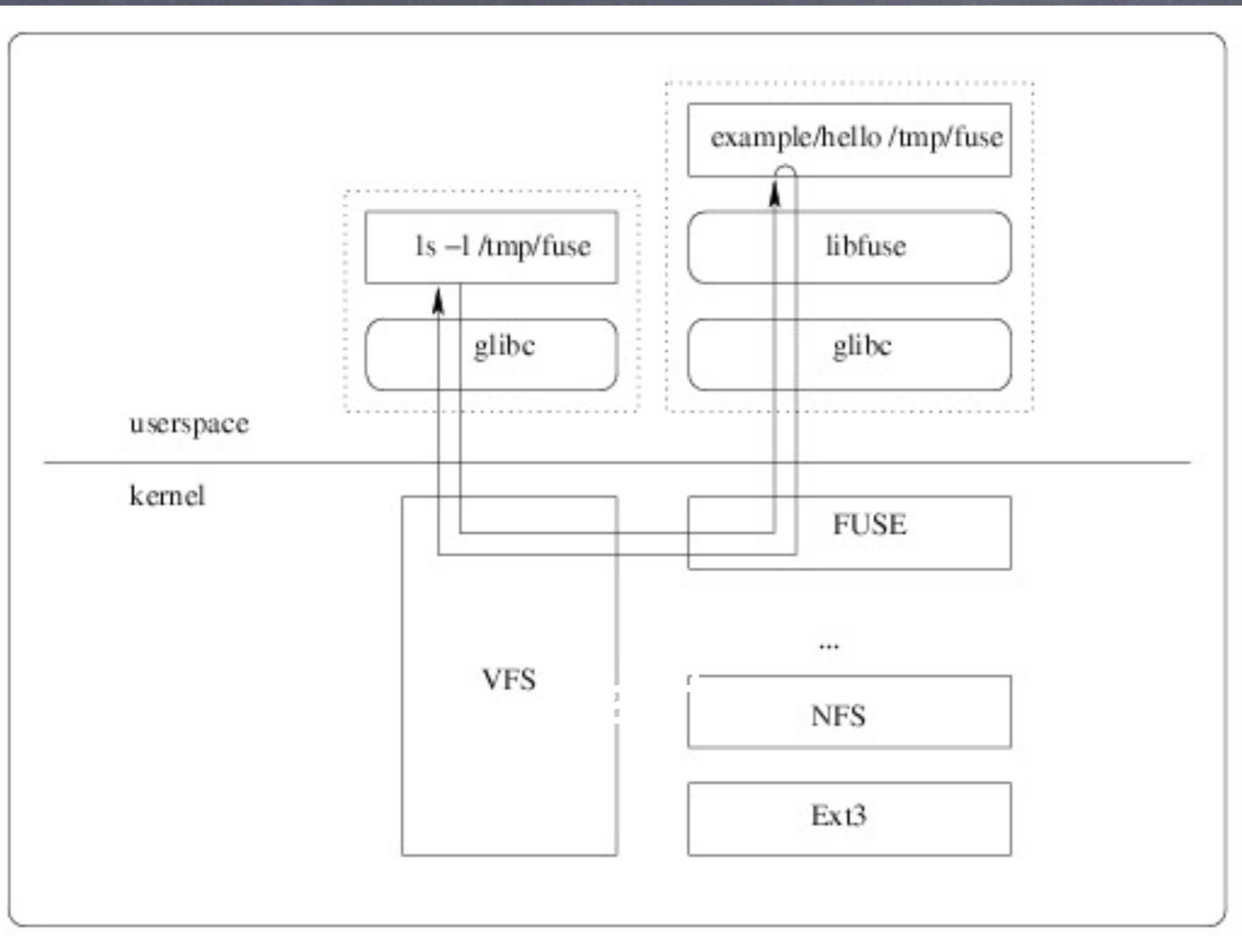
# NFSv2 als User Space FS

- `open()` erzeugt aus einem Namen eine Filehandle
- Filehandle ist unbegrenzt gültig.
- Server braucht mapping von Filehandle -> Fileobjekt
- Die Welt passt nicht in 32bit

# Was neues muss her

- Die Linux Welt wollte User Space Filesysteme
- LUFS
- FUSE (hat sich durchgesetzt)





<http://fuse.sourceforge.net/>

# Nur Linux?

- Linux
- FreeBSD / NetBSD
- Mac OS X (fuse4x, macfuse, ...)
- OpenSolaris
- Windows
- GNU/Hurd

# Vorteile

- läuft im User Space
  - kann also alle libs nutzen
- File Systeme sind portabel
- Beliebige Storage Backends möglich
  - Netzerk, Datenbank, Prozess, Filesystem
- gut für schnelle Prototyping



# Nachteile

Performance

# Anwendungen

- Praktisches
  - encfs
  - sshfs
- Netzwerkstorage / Backup / Sync
  - Wuala

# Anwendungen

- Portables
  - ZFS für Linux
  - ext3 für MacOS
- Exotisches
  - gmailfs
  - imapfs

# Fertiges

Liste aller viele Fuse Filesysteme auf

<http://fuse.sourceforge.net/>

# Praxis

<http://mirko.dziadzka.de/Vortrag/mucosug-fuse-20120119/>