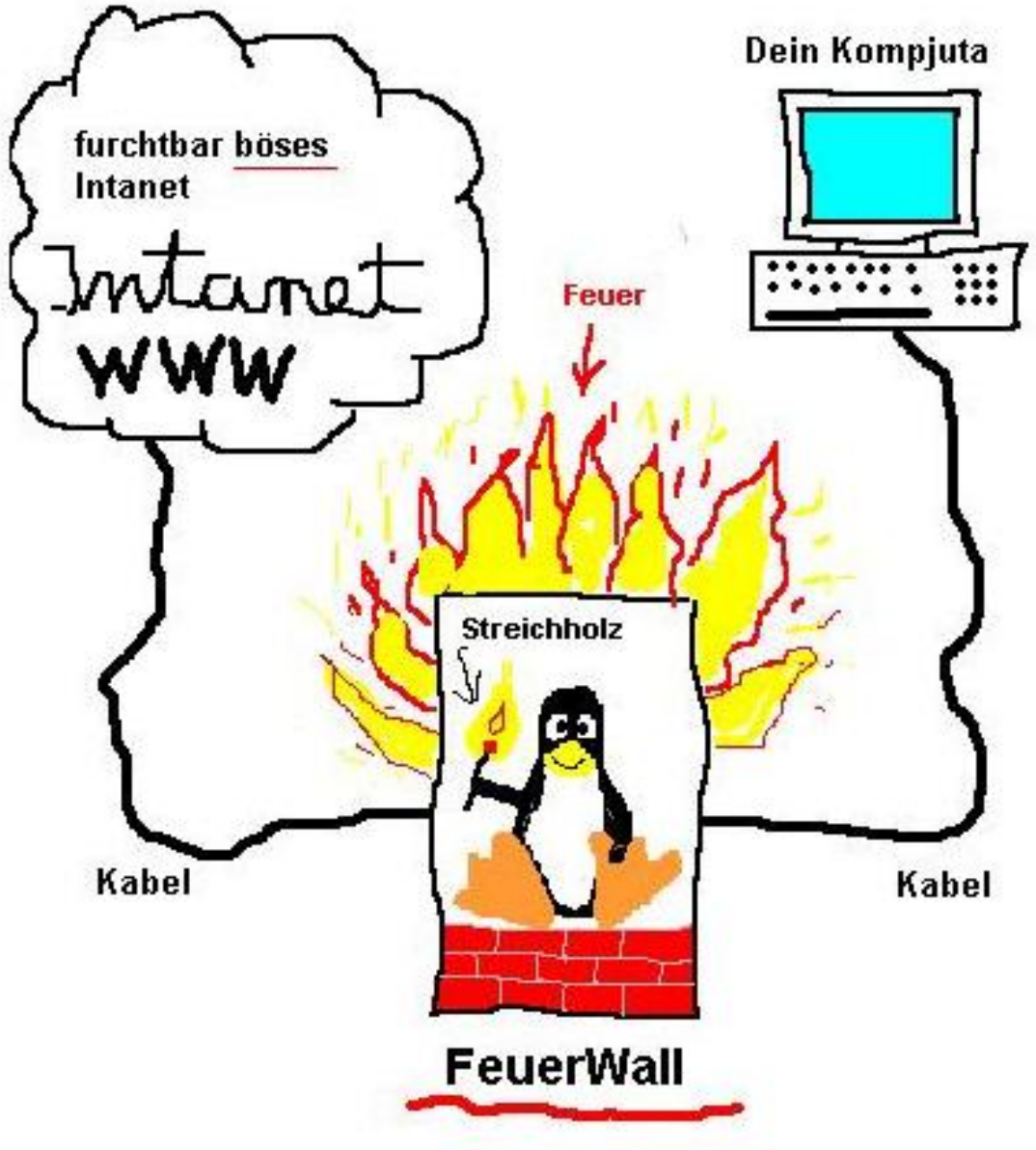
Firewall



Agenda

- Was ist eine Firewall?
- Aufgaben eines Firewall-Systems
- Elemente eines Firewall-Systems
- Firewall-Regeln
- Regeln mit **`iptables`**
- Routing und NAT
- Beispiel
- Stateful Packet Inspection

Windows Firewall NAT-Tabelle

Was ist eine Firewall?

⇒ schützt ein internes Netz vor Angriffen aus einem externen Netz

⇒ elektronische Brandschutzmauer

⇒ elektronischer Pförtner

Aufgaben eines Firewall-Systems

⇒ abschotten bestimmter Bereiche
⇒ sicherer, besonders bewachter Übergang zwischen Netzen
⇒ definierter Zugang durch Identifikation und Authentifikation
⇒ kontrollieren der ein- und ausgehenden Daten
⇒ protokollieren aller ein- und ausgehenden Daten und Ereignisse

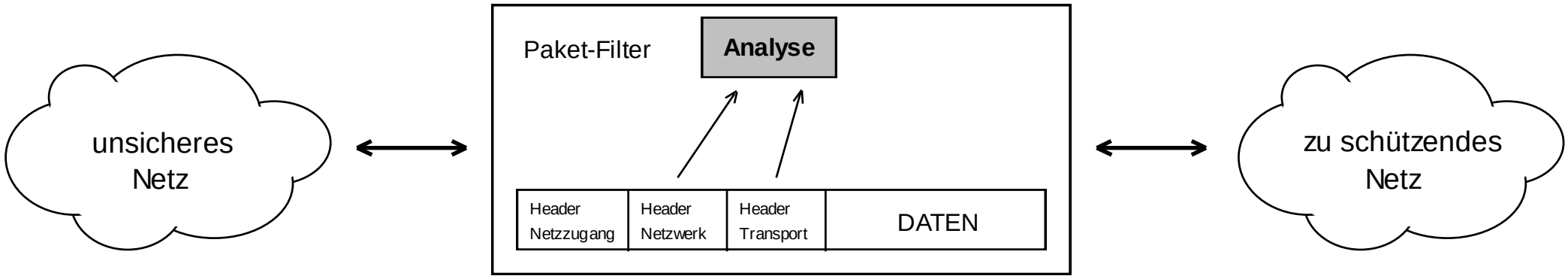
Keine Firewall ohne Sicherheitskonzept!

Elemente eines Firewall-Systems

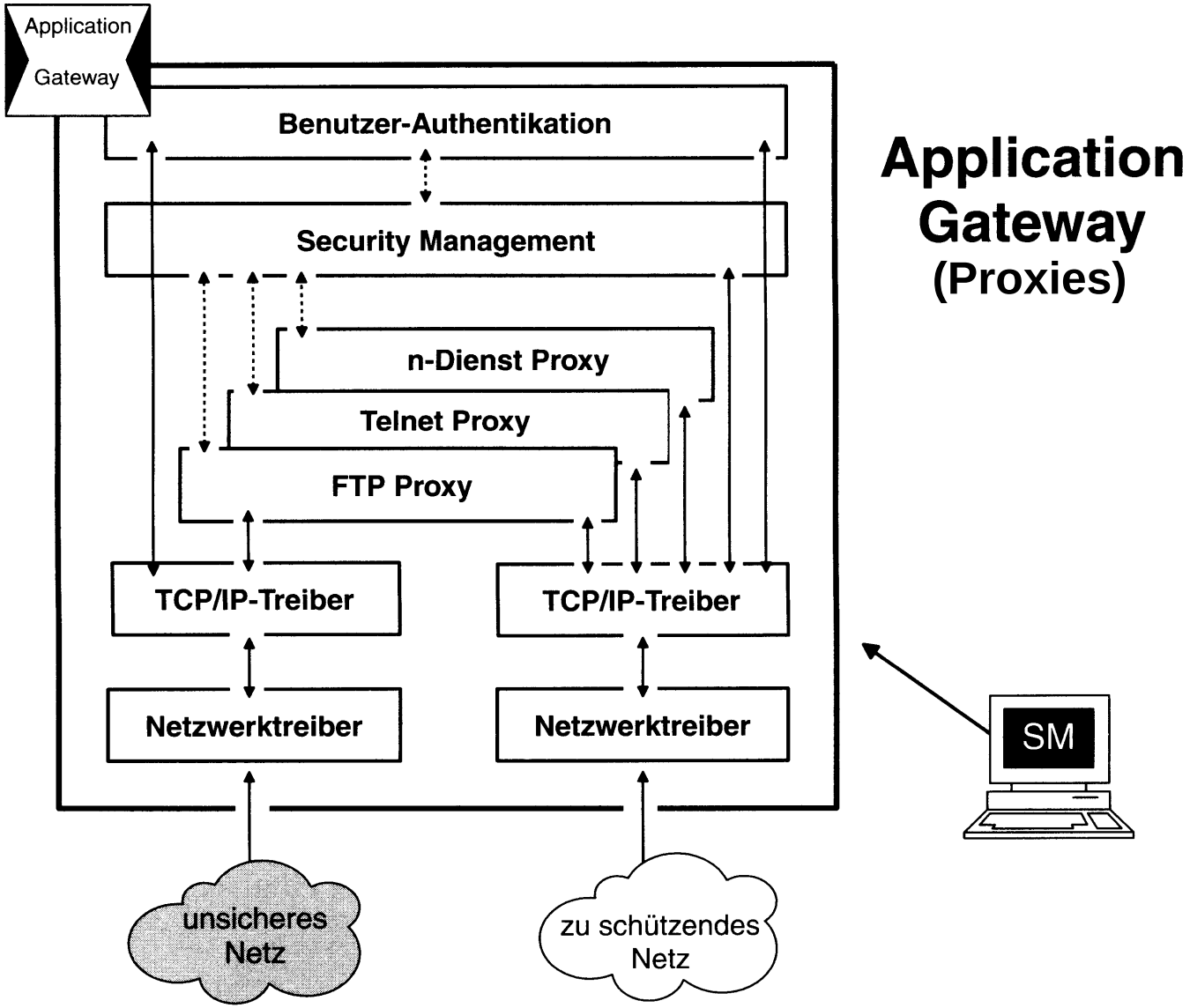
⇒ Paket-Filter

⇒ Application-Gateway (Proxies)

Paket-Filter



Analysiert nur die Header bis Layer 4

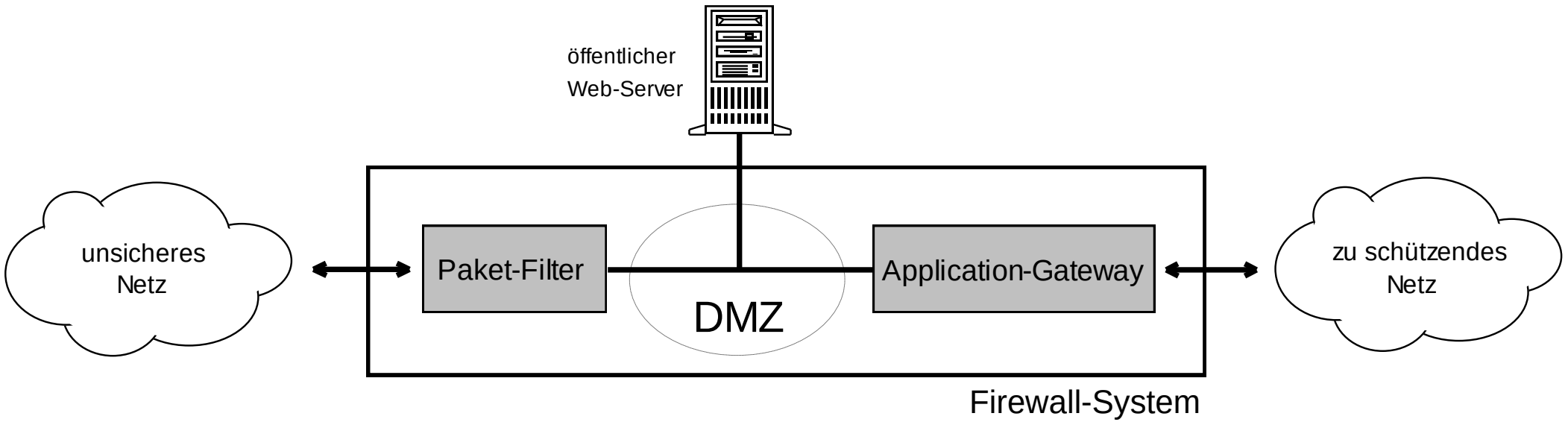


Application Gateway (Proxies)

Analysiert alle Header und Inhalte bis Layer 7

Proxy = Stellvertreter

Beispiel: Firewall-System für mittleren Schutzbedarf



DMZ - demilitarisierte Zone

Firewall-Regeln

definieren

⇒ welche Pakete die Firewall vom LAN akzeptiert

⇒ welche Pakete sie ins LAN ausgibt

⇒ welche Pakete vom Internet angenommen werden

⇒ welche Pakete ins Internet geschickt werden

⇒ .

Prüf-Kriterien sind

⇒ Richtung des Datenflusses

⇒ IP-Adressen von Absender und Empfänger

⇒ Transportprotokoll

⇒ Portnummern (Quell- bzw. Ziel-Port)

⇒ ...

Firewall-Regeln

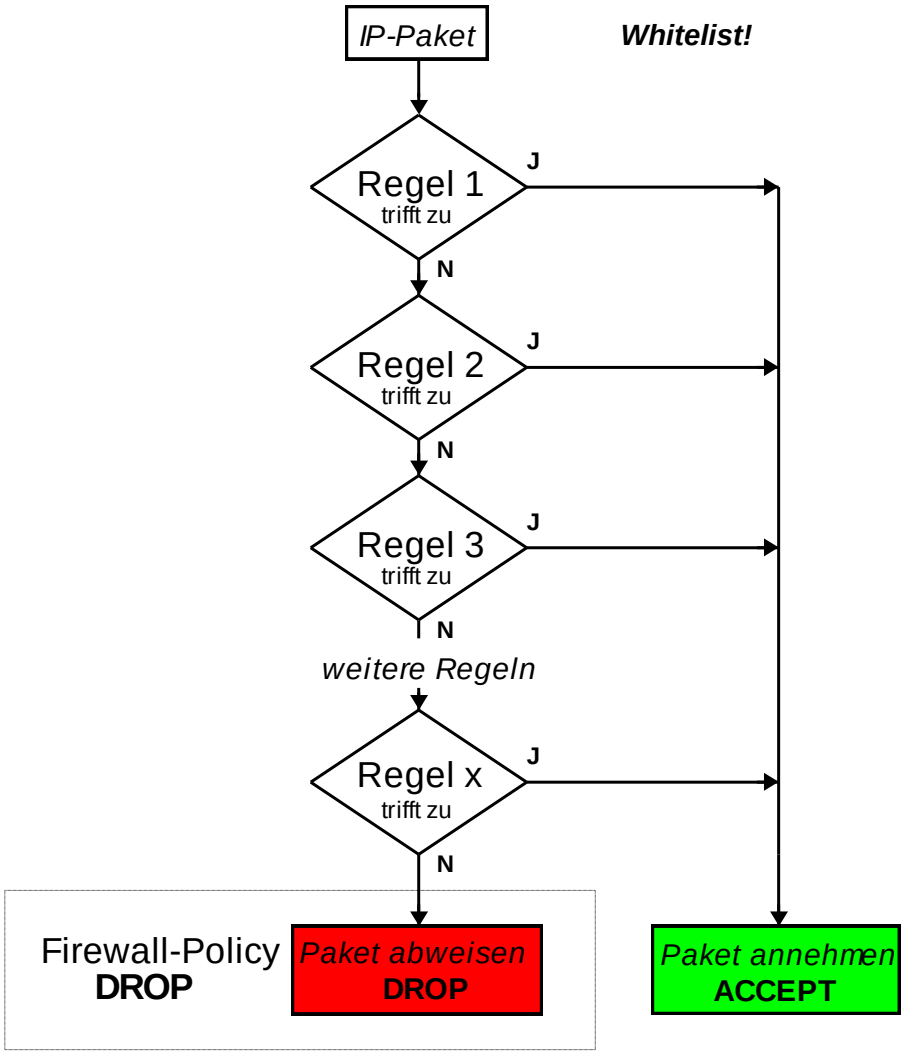
Strategie bei der Regelerstellung:

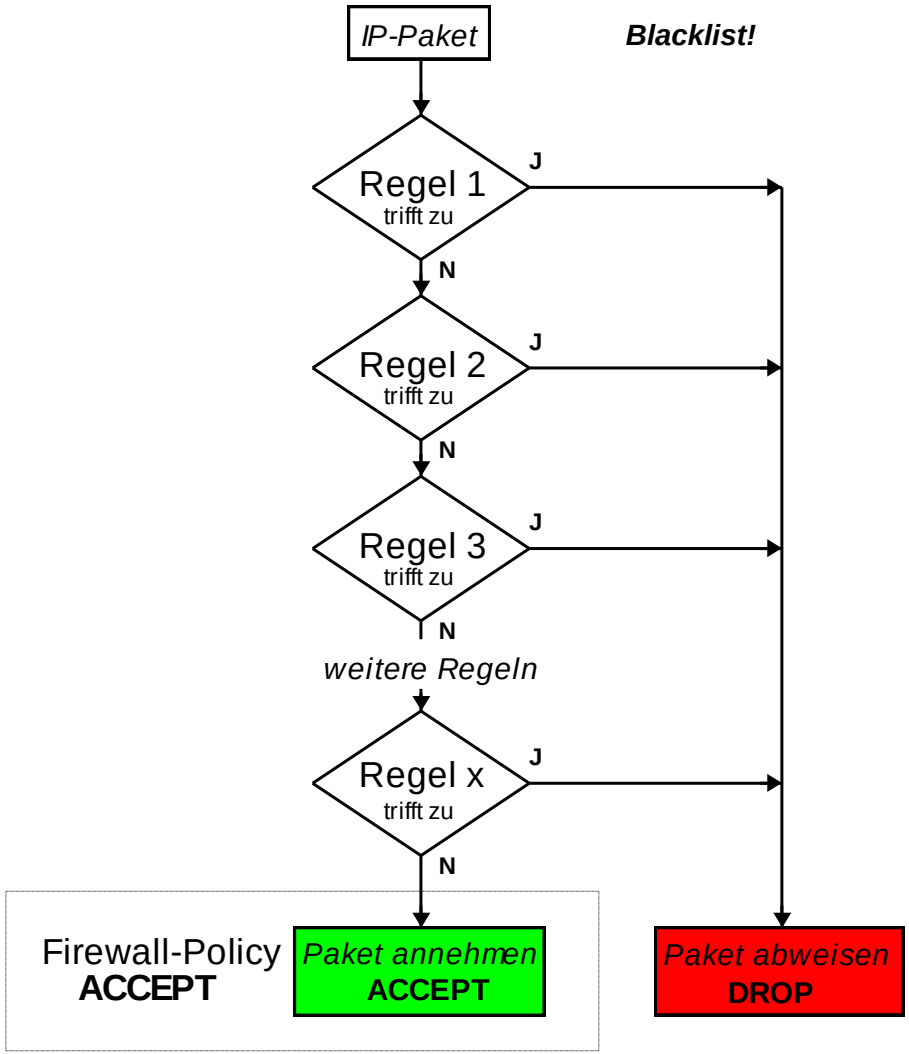
⇒ zuerst jeglichen Verkehr verbieten

⇒ dann nur das Benötigte (IP, Protokolle, Dienste) freigeben

Das schrittweise Abarbeiten der Regeln bedingt,
dass spezielle Regeln stets **vor** allgemeineren stehen müssen,
denn wenn die allgemeinere Regel zutrifft,
wird die spezielle nicht mehr erreicht (*first match*)!

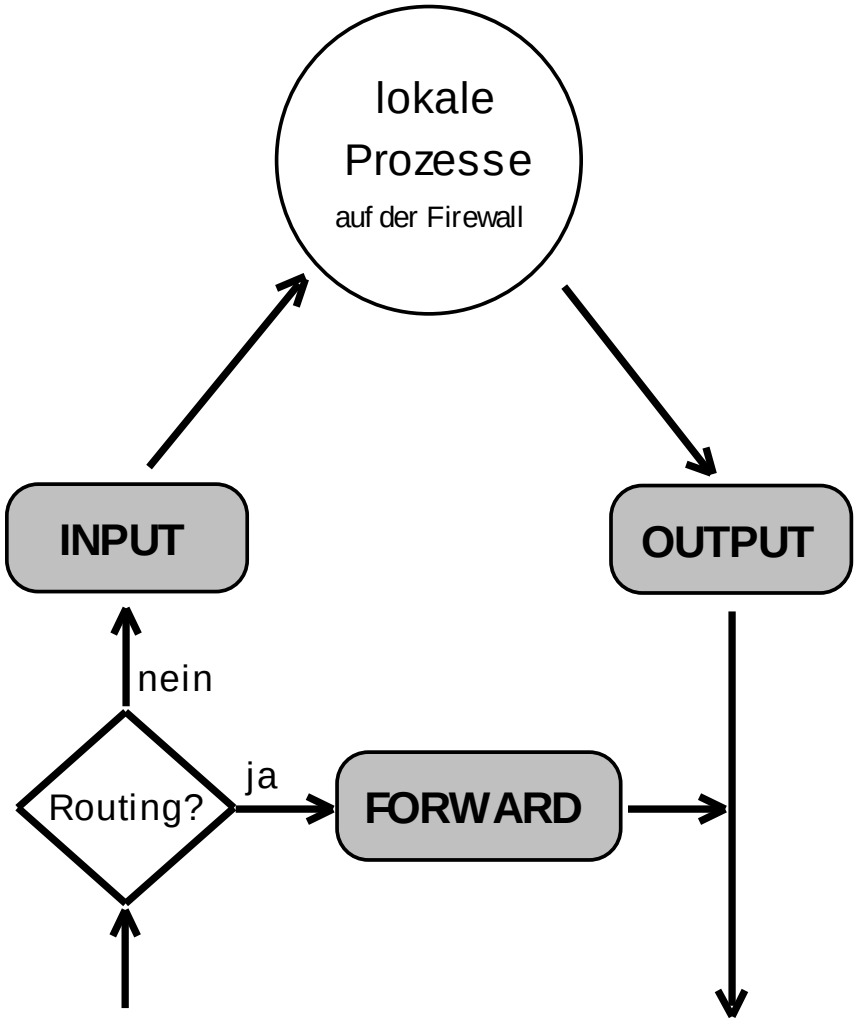
Firewall-Policy





Policy (*engl. Richtlinie, Strategie*): Standardaktion, die Pakete behandelt, auf die keine Regel zutrifft

Regel-Ketten im Linux-Kernel (*chains*)



⇒ **INPUT**

⇒ **OUTPUT**

⇒ **FORWARD**

weitere ...

- PREROUTING, POSTROUTING, NAT
- benutzerdefinierte Regel-Ketten

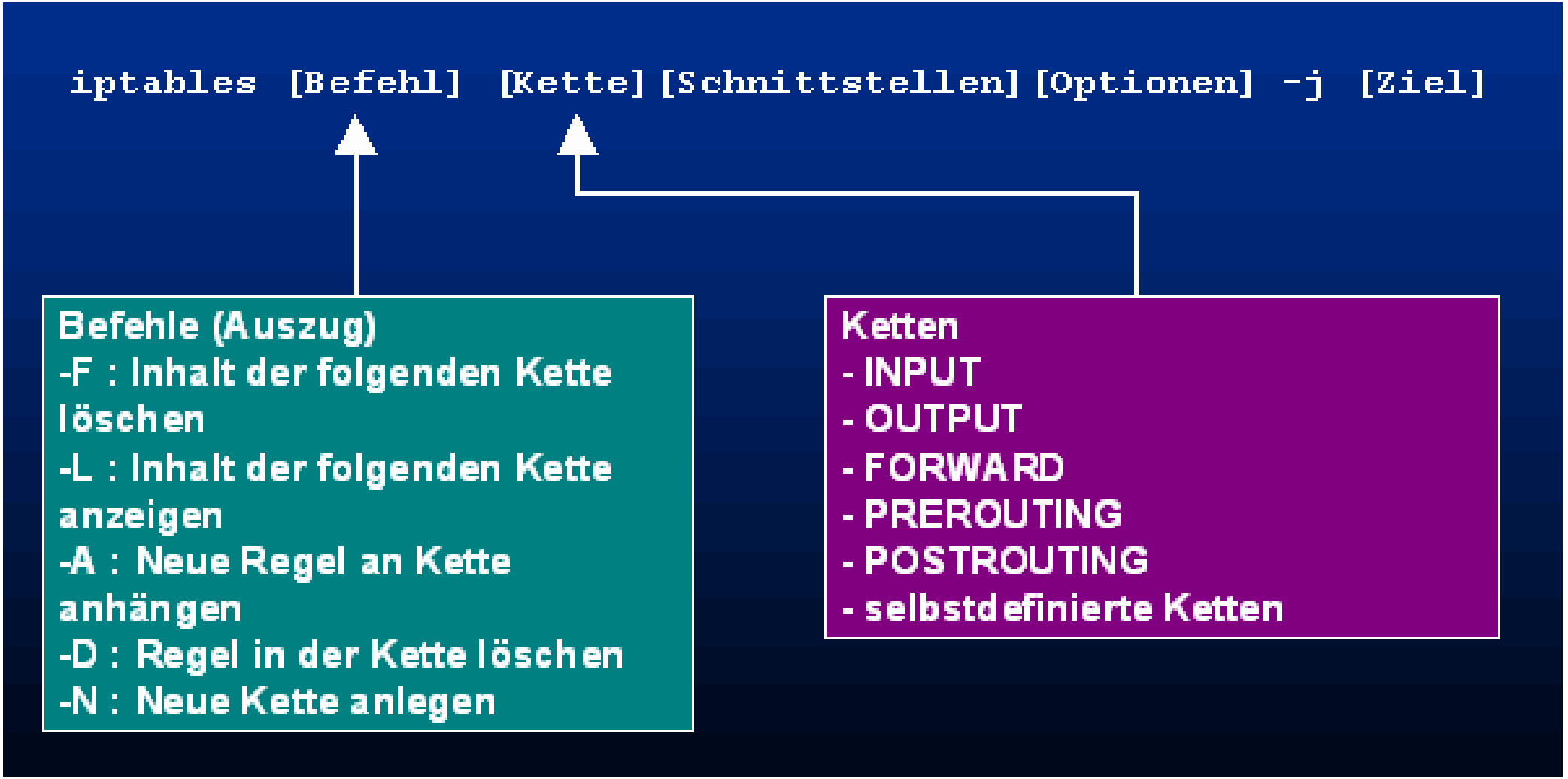
Auszuführende Aktionen z.B.:

- ACCEPT
- DROP
- REJECT
- MASQUERADING
- LOG

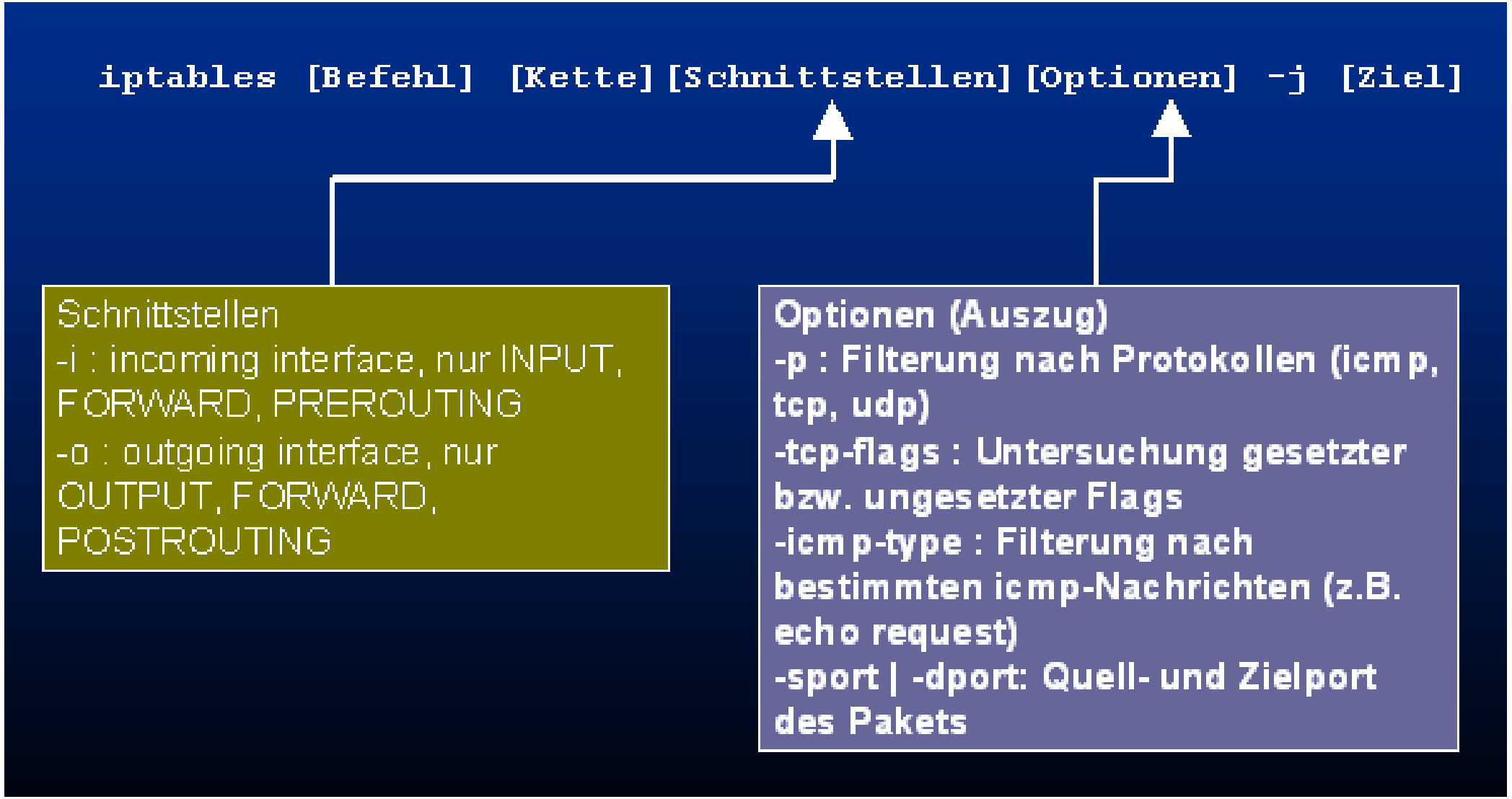
Achtung:

Wird in einer Regelliste keine zutreffende Regel gefunden,
so wird ihre Standard-Aktion (*policy*) ausgeführt!

Firewall Regeln mit `iptables` (1)



Firewall Regeln mit `iptables` (2)



Firewall Regeln mit `iptables` (3)

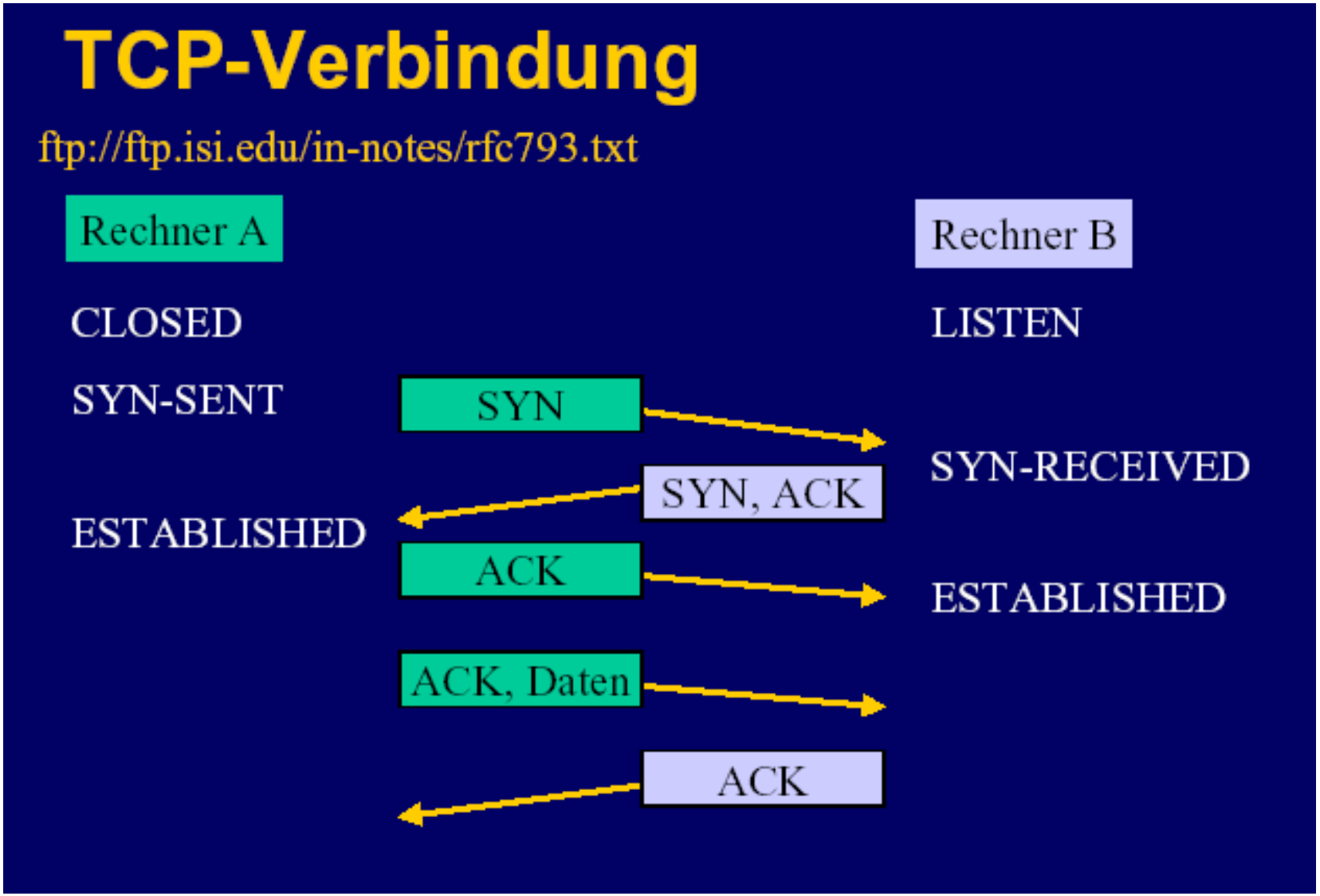
```
iptables [Befehl] [Kette] [Schnittstellen] [Optionen] -j [Ziel]
```

Ziele
DROP : Paket wird verworfen
ACCEPT : Paket wird weitergesendet
REJECT : wie DROP, Absender erhält ICMP-Fehlermeldung
LOG : Paket wird mitprotokolliert
RETURN : Durchlaufen der aktuellen Kette wird beendet
SNAT / DNAT : Quell-/Zieladresse wird geändert
MASQUERADE : wird für Verbindungen mit dynamischen IP-Adressen verwendet. Weiterleitung an IP-Adresse derjenigen Schnittstelle, über die das Paket das System verläßt; dabei gleichzeitig SNAT.

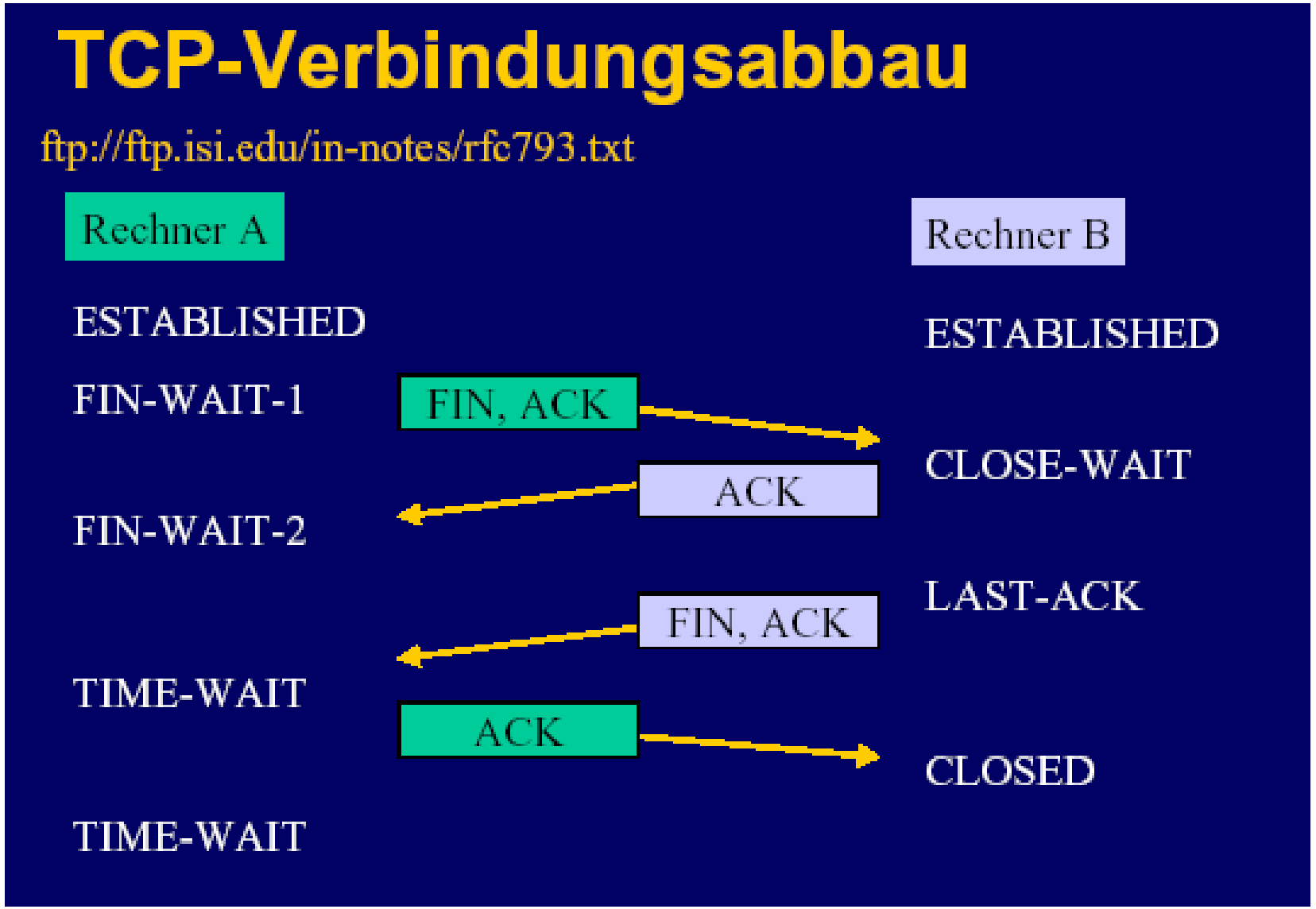
Firewall Regeln mit `iptables` (4) - Beispiele

```
iptables -P FORWARD -j DROP

iptables -A OUTPUT -d 194.94.249.128 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

iptables -A INPUT -p tcp -s 194.200.19.23 --dport smtp -j DROP

iptables -A FORWARD -s 194.94.249.2 -j meine_chain

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -o ippp0 -i eth0 -p tcp \
--dport 80 --sport 1024: -j ACCEPT

iptables -t nat -A POSTROUTING -j MASQUERADE

iptables -t nat -A POSTROUTING -j DNAT --to 194.94.249.2
```

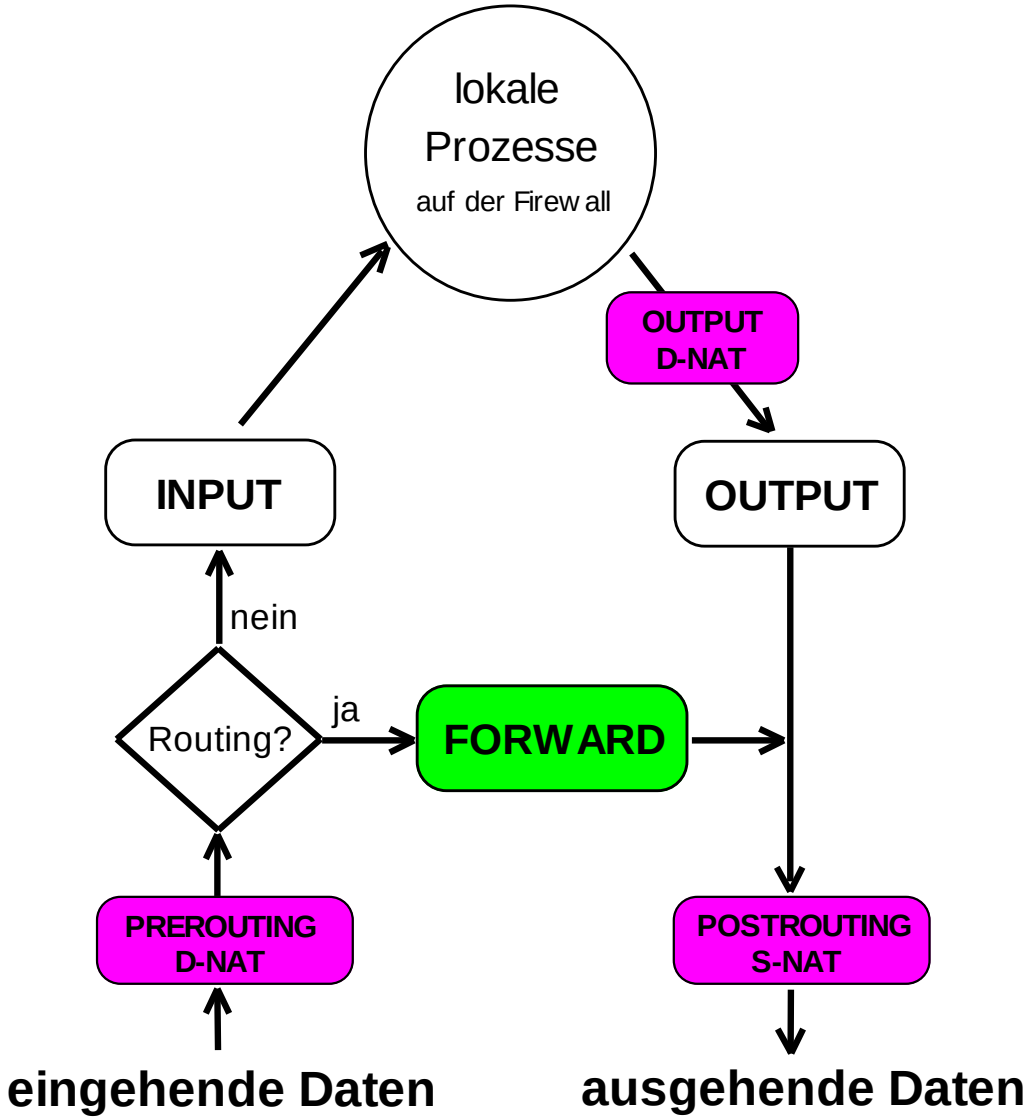
Wdh: TCP-Verbindungsaufbau



Wdh: TCP-Verbindungsabau



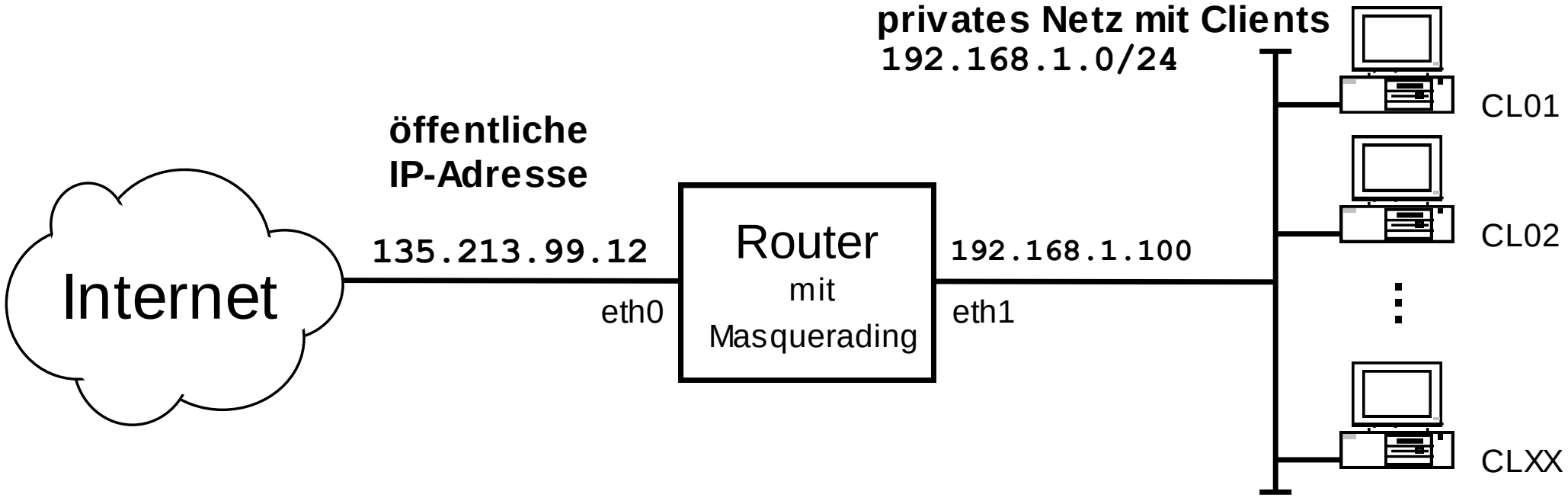
Regel-Ketten für Routing und NAT (Network Address Translation)



Source-NAT (S-NAT) bzw. Masquerading

S-NAT: Die Absenderadresse (*Source Address*) wird vom Router durch eine andere Adresse ersetzt.

Beispiel: Anschluss eines privaten Netzwerkes an das Internet über eine einzige offizielle IP-Adresse.



bei statischen öffentlichen IP-Adressen:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 135.213.99.12
```

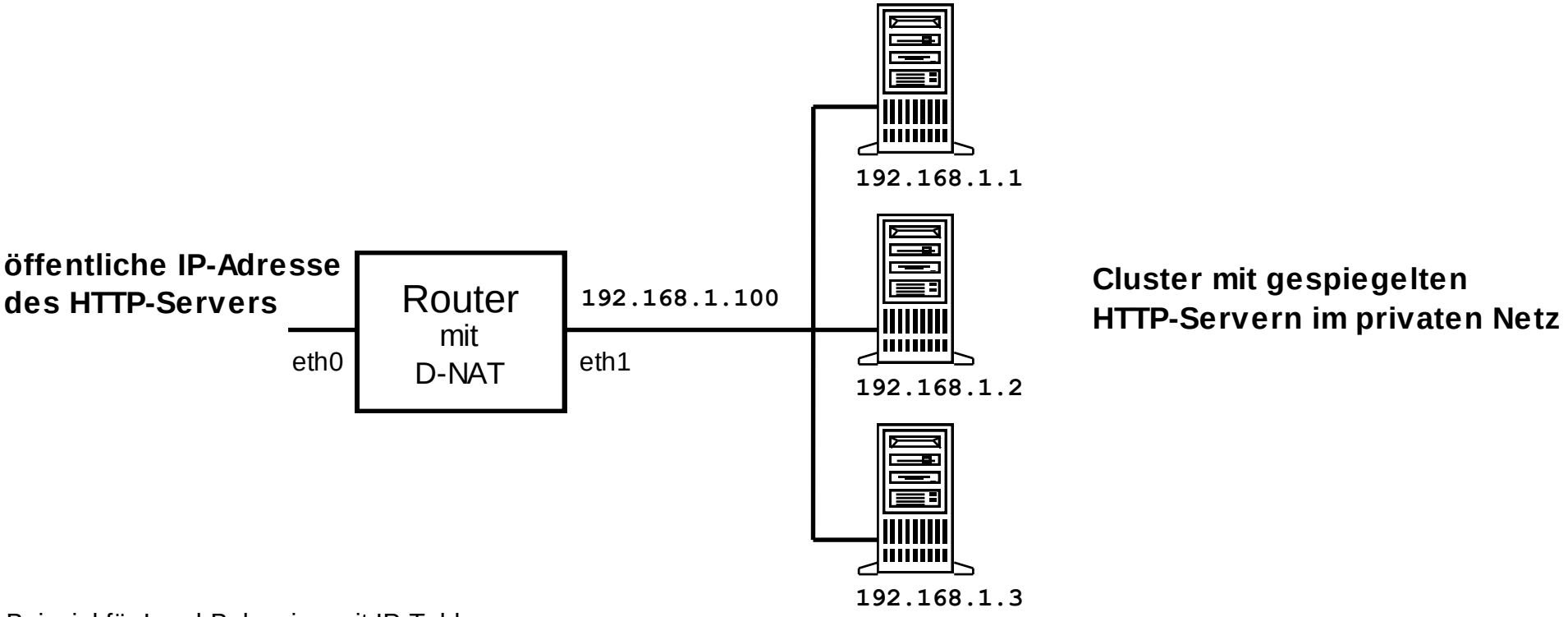
bei dynamischen öffentlichen IP-Adressen (DSL/Modem):

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Destination-NAT (D-NAT)

D-NAT: Die Empfängeradresse (*Destination Address*) wird vom Router durch eine andere Adresse ersetzt.

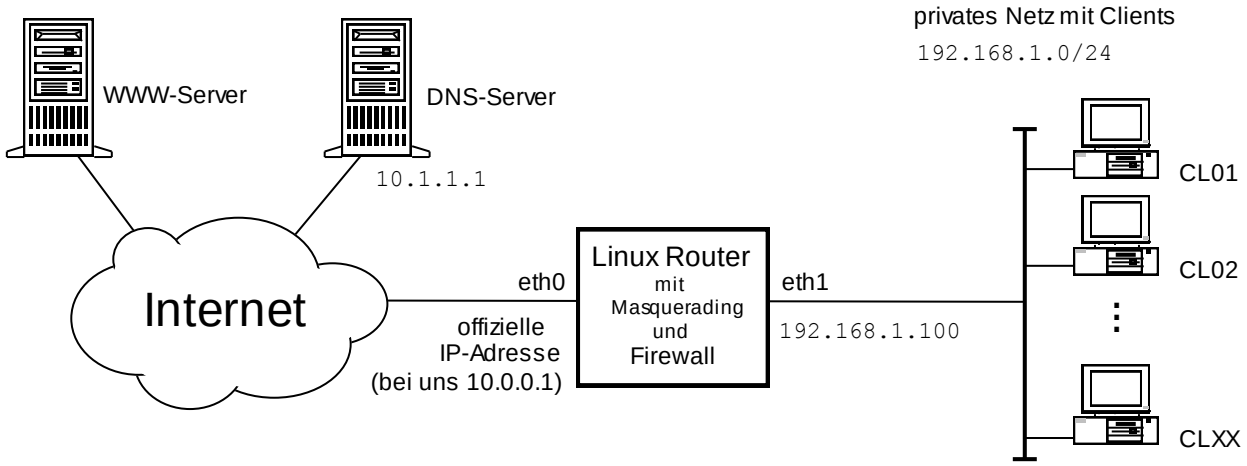
Beispiel: Anstatt eines Webservers stehen nach einem Router, der D-Nat durchführt, ein Cluster von Servern, die nacheinander bedient werden (*Load Balancing*).



Beispiel für Load-Balancing mit IP-Tables

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
                              -j DNAT --to-destination 192.168.1.1-192.168.1.3
```

Regeln zu einer Beispiel-Firewall (1)




```
iptables -P INPUT   DROP
iptables -P OUTPUT  DROP
iptables -P FORWARD DROP
```

Default-Policies auf Abweisen aller Pakete setzen

```
iptables -F
iptables -F -t nat
iptables -X
```

alle vorhandenden und benutzerdefinierten Regelsätze löschen

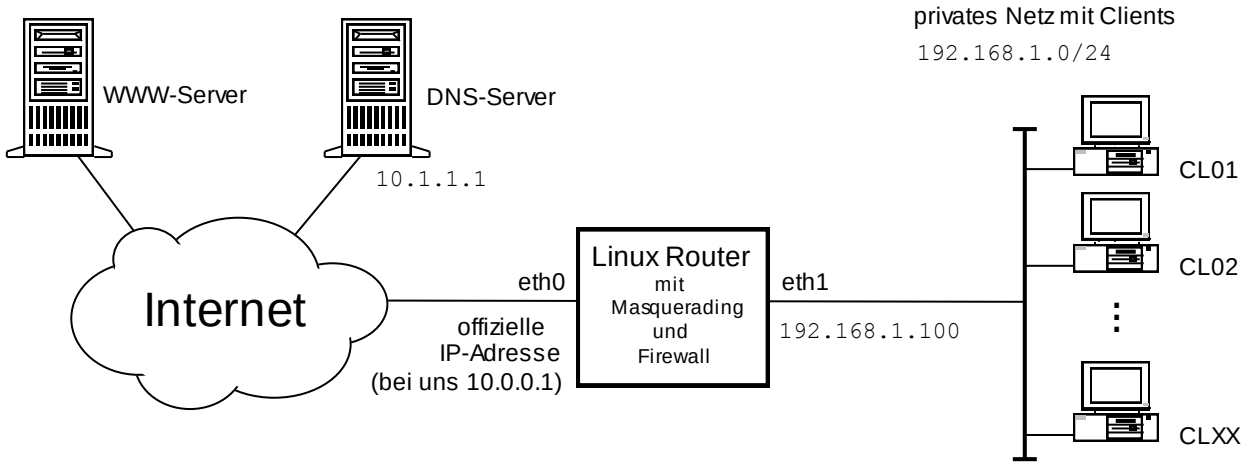
```
iptables -A INPUT  -i eth1 -s 192.168.1.0/24 -j ACCEPT
iptables -A OUTPUT -o eth1 -d 192.168.1.0/24 -j ACCEPT
```

von innen den Zugriff auf die Firewall erlauben

```
iptables -A FORWARD -i eth0 -s 192.168.1.0/24 -j DROP
```

noch einmal explizit Pakete, die von außen kommen, aber eine interne Quell-IP-Adresse besitzen, blocken (Anti-Spoofing)

Regeln zu einer Beispiel-Firewall (2)



ausgehenden Ping erlauben

```
iptables -A FORWARD -o eth0 -p icmp --icmp-type echo-request -j ACCEPT
iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply   -j ACCEPT
```

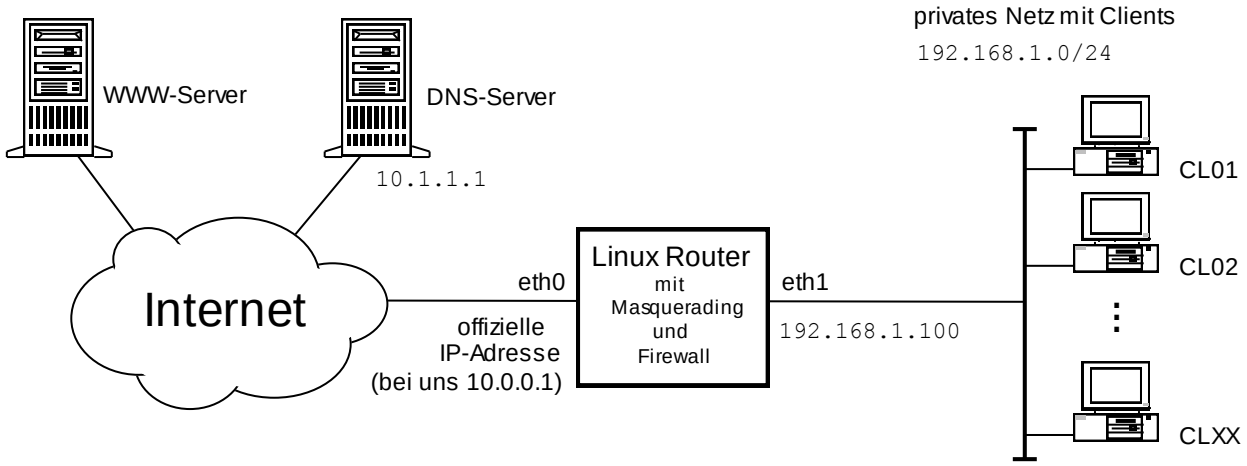
HTTP-Zugriff nach außen erlauben

```
iptables -A FORWARD -o eth0 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -p tcp --sport 80 -j ACCEPT
```

DNS-Anfragen (UDP) an 10.1.1.1 erlauben

```
iptables -A FORWARD -o eth0 -p udp -d 10.1.1.1 --dport 53 -j ACCEPT
iptables -A FORWARD -i eth0 -p udp -s 10.1.1.1 --sport 53 -j ACCEPT
```

Regeln zu einer Beispiel-Firewall (3)



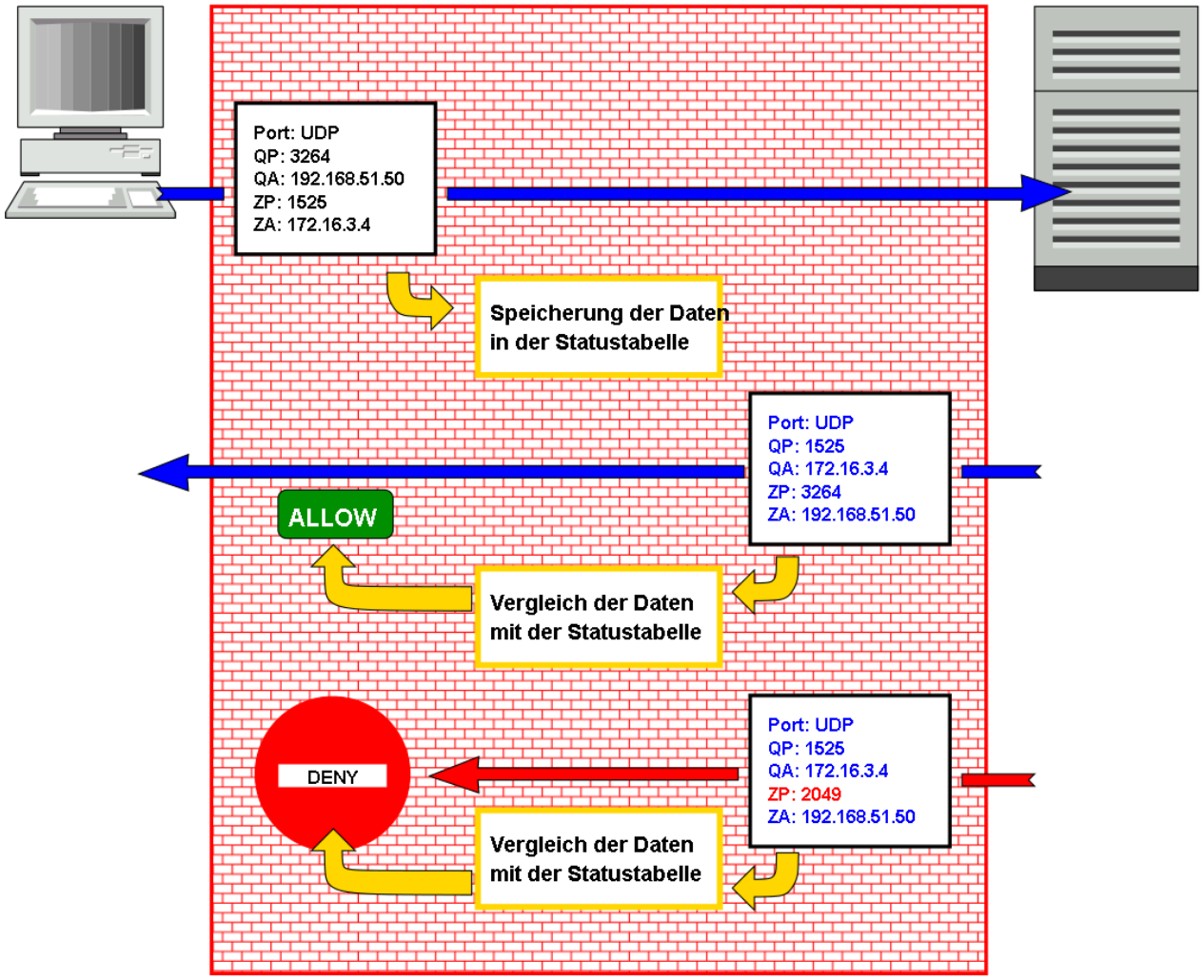
Maskieren der Source-Adresse der Clients mit der Adresse der äußeren Netzwerkkarte (MASQUERADING bzw. S-NAT)

```
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.1.0/24 -j MASQUERADE
```

schließlich das IP-Forwarding im Kernel erlauben

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Stateful Packet Inspection (SPI)



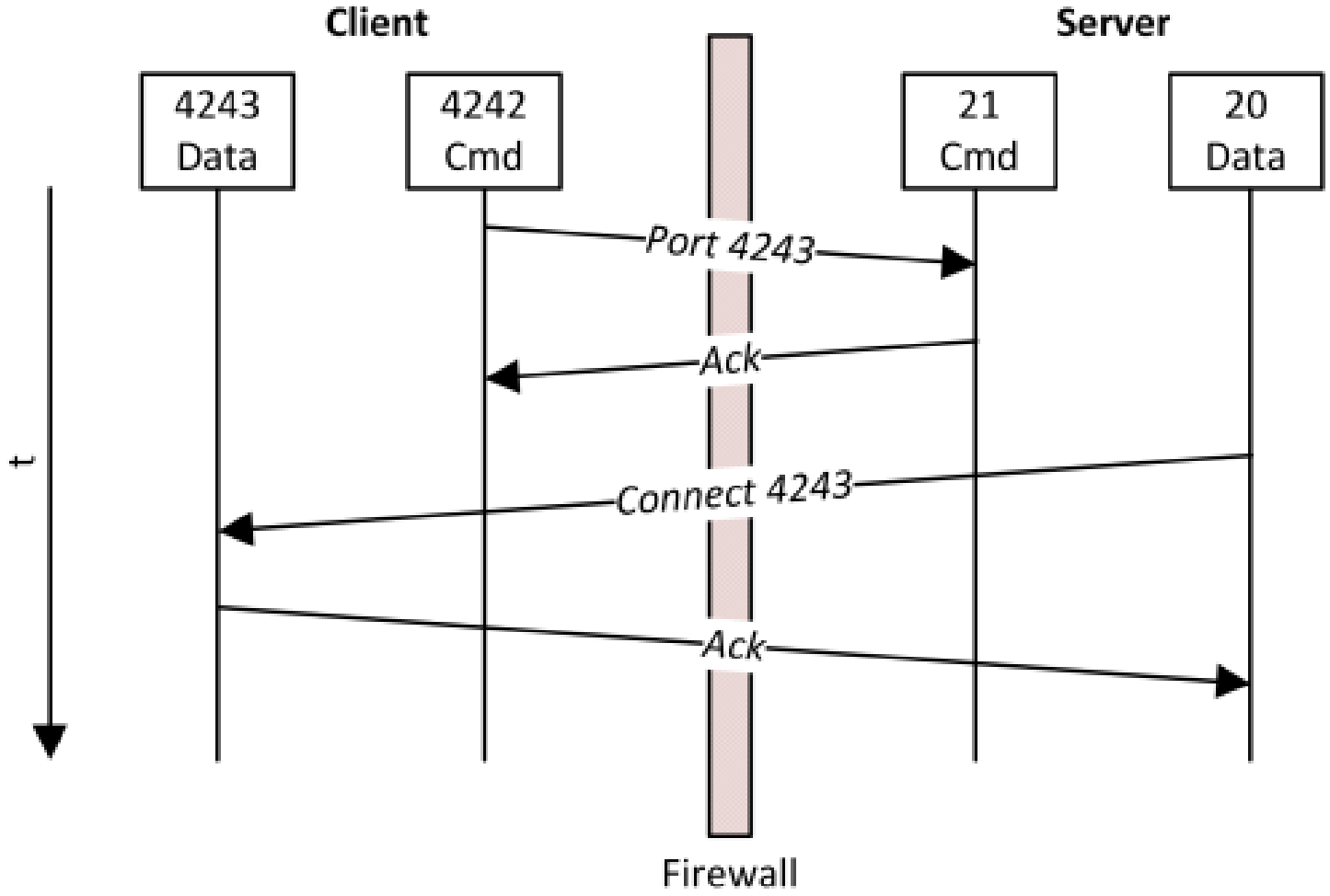
dynamischer Paketfilter, der jedes Datenpaket einer bestimmten aktiven Session zuordnet

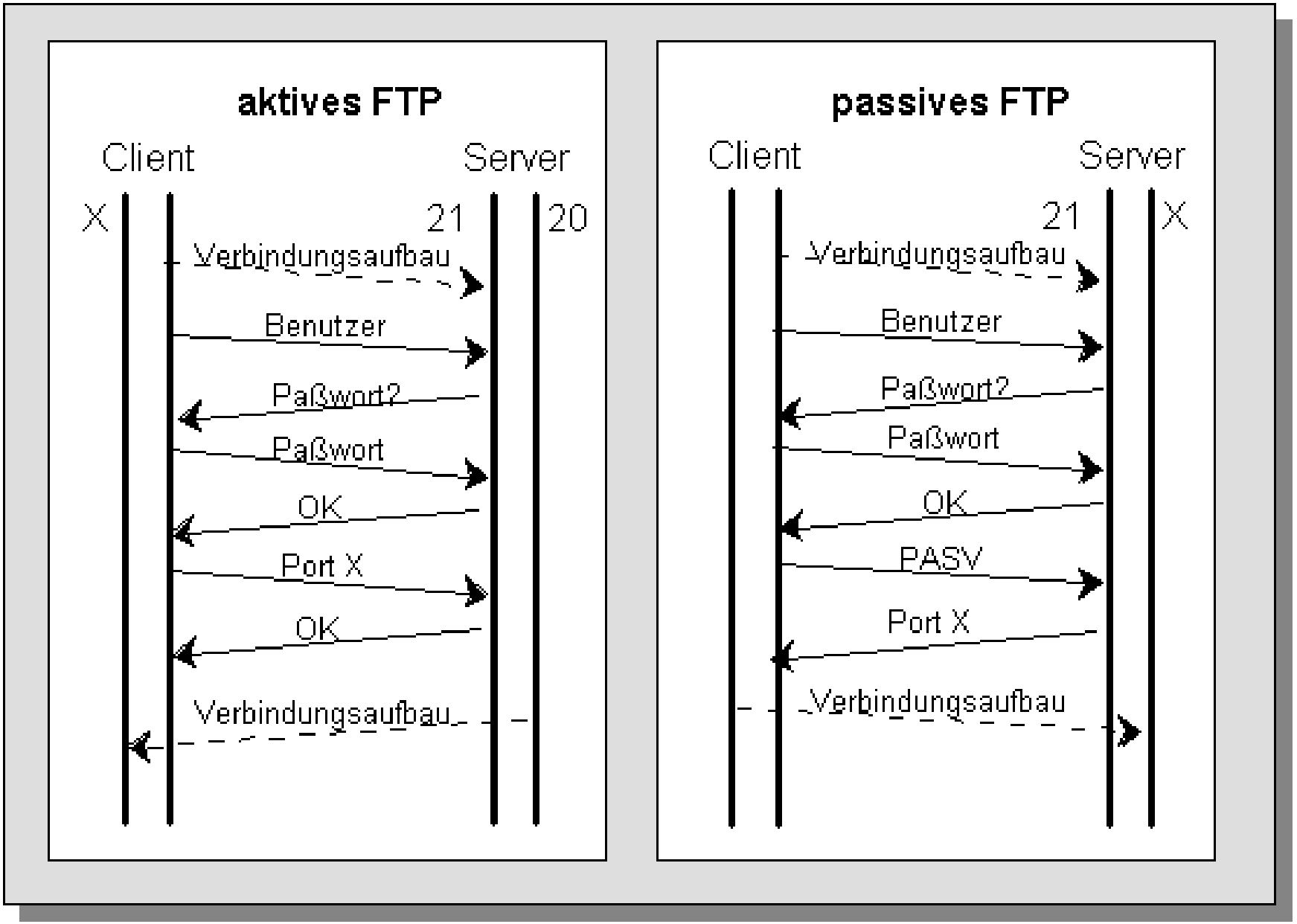
Quelle
de.wikipedia.org/wiki/Stateful_Packet_Inspection

Beispiel: `iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT`

Stateful Packet Inspection (SPI)

Beispiel: aktives FTP




aktives FTP
Client
Server
X
21
20
Verbindungsaufbau
Benutzer
Paßwort?
Paßwort
OK
Port X
OK
Verbindungsaufbau
passives FTP
Client
Server
21
X
Verbindungsaufbau
Benutzer
Paßwort?
Paßwort
OK
PASV
Port X
Verbindungsaufbau

alle Regeln löschen:

```
iptables -P OUTPUT ACCEPT ; iptables -P INPUT ACCEPT; iptables -F ; iptables -L
```

Windows-Firewall (aus: Top 10: Windows Firewall Netsh Commands)

*immer **eine** Zeile!*

alle Regeln anzeigen

```
netsh advfirewall firewall show rule name=all
```

eingehendes ICMP (Ping) erlauben

```
netsh advfirewall firewall add rule name="All ICMP V4"
dir=in action=allow protocol=icmpv4
```

eingehendes ICMP (Ping) verbieten

```
netsh advfirewall firewall add rule name="All ICMP V4"
dir=in action=block protocol=icmpv4
```

Port 1433/TCP eingehend für den MS SQL-Server öffnen

```
netsh advfirewall firewall add rule name="Open SQL Server Port 1433"
dir=in action=allow protocol=TCP localport=1433
```

Regel für MS SQL-Server löschen

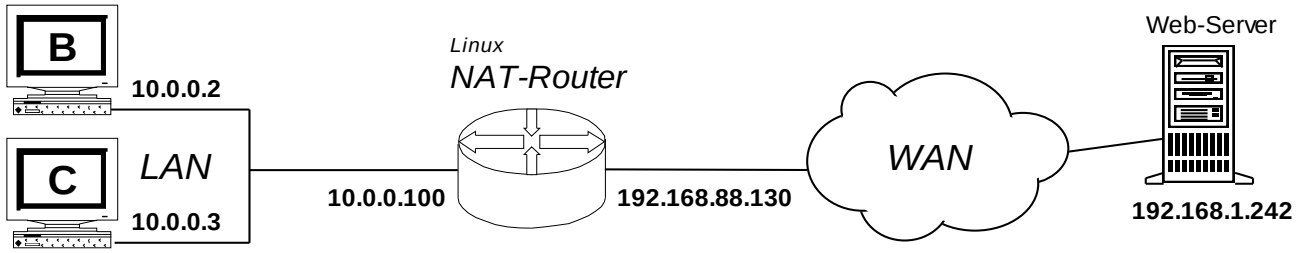
```
netsh advfirewall firewall delete rule name="Open SQL Server Port 1433"
protocol=tcp localport=1433
```

einem Programm Netzwerkzugriff gewähren

```
netsh advfirewall firewall add rule name="Trojaner erlauben"
dir=in action=allow program="C:\Program Files\ungefährlichesProgramm\trojan.exe"
```

auch: Windows Firewall mit erweiterter Sicherheitsverwaltung mit Windows PowerShell

NAT-Tabelle bei einem NAT-Router (Linux)



ausgegeben auf dem NAT-Router mit **`netstat-nat`**

```
root@ubuntu:~# netstat-nat -nN
Proto NATed Address                 NAT-host Address                Destination Address          State
tcp   10.0.0.2:56331          <---  192.168.88.130:56331            192.168.1.242:80             ESTABLISHED
tcp   10.0.0.2:56332          <---  192.168.88.130:56332            192.168.1.242:80             ESTABLISHED
tcp   10.0.0.2:56333                192.168.88.130:56333            192.168.1.242:80             ESTABLISHED
tcp   10.0.0.3:56325                192.168.88.130:56325            192.168.1.242:80             CLOSE
tcp   10.0.0.3:56331          <---  192.168.88.130:1024     <---     192.168.1.242:80             CLOSE
tcp   10.0.0.3:56319                192.168.88.130:56319    ?       192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56328                192.168.88.130:56328            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56332          <---  192.168.88.130:1025     <---     192.168.1.242:80             CLOSE
tcp   10.0.0.3:56326                192.168.88.130:56326            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56330                192.168.88.130:56330            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56329                192.168.88.130:56329            192.168.1.242:80             CLOSE
tcp   10.0.0.3:56323                192.168.88.130:56323            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56327                192.168.88.130:56327            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56318                192.168.88.130:56318            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56324                192.168.88.130:56324            192.168.1.242:80             SYN_SENT
tcp   10.0.0.3:56316                192.168.88.130:56316            192.168.1.242:80             SYN_SENT
root@ubuntu:~#
```

1. Schritt: auf 10.0.0.2: **`ncat 192.168.1.242 80`**
   dann wurde mit **netstat** der benutzte Client-Port ermittelt (z.B. 56331)

2. Schritt auf 10.0.0.3: **`ncat -p 56331 192.168.1.242 80`**
   hier wird **ncat** gezwungen, genau diesen Client-Port 56331 auch zu benutzen --> dann ersetzt der NAT-Router den Client-Port

bei Netfilter/IPtables befindet sich die NAT-Tabelle in `/proc/net/ip_conntrack` bzw. `/proc/net/nf_conntrack`