

# Systems Advanced II

Samenvatting - Tristan Reynders - 2023

<b>Kernel</b>	<b>4</b>
Linux Virtual Memory	4
Memory Usage	4
Linux Kernel	4
Features en Services - Linux Kernel	5
Linux Kernel Organization	5
User Space en User Processes	5
CPU Privilege modes	6
Linux Virtual File System	6
Process Management	7
Linux processen	7
Types of Linux Processes	7
Process States & Signals	8
fork() en exec()	8
Shell Command Execution	8
Process Management - context switching	8
The Linux Scheduler	9
Linux threads	9
Advantages threads over processes	9
Interrupts in Linux	10
Sockets	10
<b>Routing</b>	<b>11</b>
Terminologie	11
<b>Firewall</b>	<b>12</b>
Functies van een Firewall	12
iptables en netfilter	12
nftables	13
Packet Processing	13
Tables	14
Chains	14
iptables: Tables and Chains	15
Rules	16
Special Targets	16
Stateless vs stateful firewall	16
NAT masquerading met conntrack table	17
Default Packet-Filtering Policy	17
<b>NFS</b>	<b>18</b>
Network File System (NFS)	18

NFS Architecture	18
NFS Networking	18
Quorum	18
Load Balancers	19
State, Stateful en Determinism	19
Finite State Machine (FSM)	20
REST - REpresentational State Transfer	21
REST Components	21
Session-based authentication	22
Token-based authenticatie	22
Load Balancer	22
Typische features	22
Sticky Session	23
Reverse proxy	23
Reverse proxies - typical features	23
Load Balancing - OSI layers	24
Load Balancing Algoritmes	24
<b>eBPF</b>	<b>25</b>
Belang	25
eBPF networking	25
<b>Systemd</b>	<b>26</b>
Issues met traditionele init systemen	26
Advantages systemd	26
Disadvantages systemd	26
Features van systemd	26
Message Bus communication models	28
D-Bus (Desktop Bus)	28
Systemd Units	29
Systemd timers	29
Systemd targets	30
Praktische targets	30
Runlevel	30
Systemd logs	30
<b>Booting linux</b>	<b>31</b>
UEFI	31
Enhancing Bootloader Functionality and Flexibility	32
Linux Boot Loader Stage	33
Software Licenses	34
Open-Source Software	34
Software license types	34
GPL (GNU General Public License)	35
GPLv2	35
GPLv3	35
GPLv2 vs. GPLv3	35

<b>Containers</b>	<b>36</b>
Container	36
Containers in de linux kernel	36
Container Image	37
Docker runs in user space	37
Open Container Initiative (OCI)	38
<b>Kubernetes</b>	<b>39</b>
Definitie	39
Problems solved door Kubernetes	39
Monolithic Applications	39
Microservices Architectuur	39
Features	39
Terminologie	40
The Declarative Model	40
Kubernetes Service	40

# Kernel

## Linux Virtual Memory

- Wanneer een linux programma memory allocate bestaat deze memory initieel nog niet, het is een entry in een table in het OS.
- Wanneer het programma de memory wilt gebruiken wordt deze gevonden en gebruikt.

## Memory Usage

- Hoeveel virtual memory er in totaal gebruikt wordt
- Hoeveel actual memory of “resident” memory er gebruikt wordt gelimiteerd tot het systeems resident RAM capacity (+ swapping to disk).

## Linux Kernel

- Basis component van het Linux OS
- Verantwoordelijk voor:
  - Het managen van system resources
  - Aanbieden van low-level services aan andere componenten van het OS
  - Controle over hardware devices
- Typisch beschreven als monolithisch
  - Alle system-level services zitten in een executable file
  - Voordelen:
    - Verbeterde performance
    - Versimpeld systeem management
- De kernel omvat ook een modulair design
  - Sommige features kunnen als loadable kernel modules gecompileerd worden
  - Modules kunnen dynamisch ingeladen of uitgeladen worden tijdens runtime
  - Betere flexibiliteit en customisatie mogelijkheden
    - Experimenteren met nieuwe features zonder het te commiten naar de main kernel

## Features en Services - Linux Kernel

- **Process management**
  - Managen van running processes
  - Allocaten van resources
  - Schemen van CPU time
- **Memory management**
  - Verantwoordelijk voor het managen van allocation en deallocation van system memory
  - Verantwoordelijk voor het implementeren van virtual memory
- **File system management**
  - Zorgt voor de file system interface voor het managen van meerdere file systemen
- **Device management**
  - Controle over hardware apparaten (Disks, network adapters, IO devices)
- **Network management**
  - Networking stack

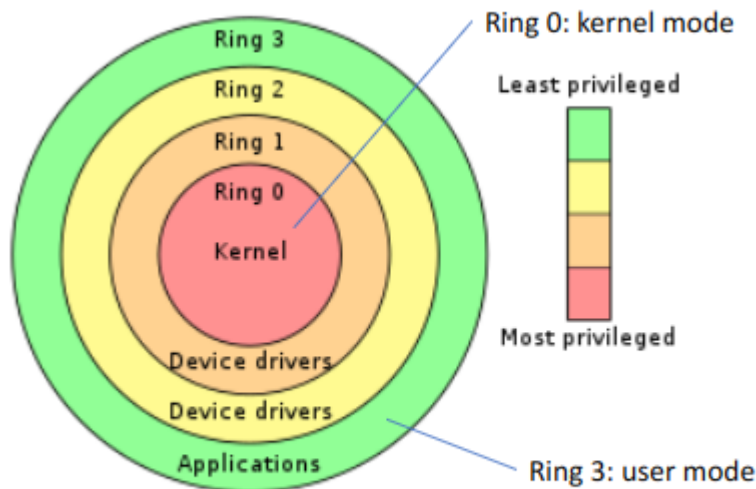
## Linux Kernel Organization

- Software in memory dat de CPU verteld waar deze moet kijken voor zijn volgende taak.
- Handelt als tussenpersoon tussen hardware en applicaties
- (User) processen, managed door de kernel -> user space

## User Space en User Processes

- De kernel runned in kernel mode, de user processen in user mode
- **User space**
  - User mode geeft enkel toegang tot een kleine subset van memory en safe CPU operations → Makkelijk op te ruimen door de kernel als er iets fout gaat.
- **Kernel space**
  - Code die runned in kernel mode heeft toegang tot de CPU en main memory
  - Krachtig maar gevaarlijke rechten → Kan makkelijk de kernel corrupten en het hele systeem laten crashen
  - Memory dat enkel de kernel kan gebruiken

## CPU Privilege modes



*CPU Privilege rings for the Intel x86*

- Veel moderne CPU architecturen zoals ARM en x86 maken gebruik van deze ring protection.
- Ring 0
  - Hoogste autoriteit en kan direct aan alle resources
- Ring 3
  - Enkel toegang tot bepaalde resources
  - Moet door system calls privileged resources aanspreken
- Geeft de CPU een manier om tussen bepaalde levels van privilege te switchen tijdens runtime

## Linux Virtual File System

- Geeft een unified view van het filesystem aan apps en kernel
- File systemen gemount op specifieke mount points in file system hierarchy
- Cached recente files/directories in memory voor een performance boost
- Support voor vele file systeem types (local, network, special)
  - **procfs**
    - Virtuele view van het running systeem
    - Geeft toegang tot systeeminfo en config parameters
  - **sysfs**
    - Virtuele view van het systeem zijn hardware devices, drivers
    - Geeft toegang tot device attributes en settings

## Process Management

- Starten, pauzeren, voortzetten, schedulen en het termineren van processen
- Elk proces gebruikt de CPU voor een fractie van een seconde, vervolgens gebruikt een ander proces de CPU voor een fractie van een seconde en zo voort..  
→ **Context switch**
- **Time slice**
  - Elk stukje tijd geeft een proces voldoende tijd voor serieuze bewerkingen
- **System calls**
  - Voor het managen van processen → Het aanmaken van nieuwe processen, prioriteit van een proces aan te passen, het termineren van processen

## Linux processen

- Een instantie van een programma dat uitgevoerd wordt
- Elk proces heeft zijn eigen memory space, program code en execution context, program counter, stack en andere registers geïncorporeerd.
- Processen kunnen communiceren met elkaar door middel van interprocess communication (IPC) mechanismen
  - Pipes, sockets, shared memory segments
- Kunnen worden aangemaakt door middel van de **fork** system call
  - Maakt een nieuw proces door het huidige proces te dupliceren en krijgt een ander PID toegewezen
  - Het nieuwe proces noemt het child process en heeft zijn eigen uniek process ID (PID)
- Kunnen ook worden aangemaakt door middel van een **exec** system call
  - Het huidige proces wordt vervangen door een nieuw programma of clone
  - Creeert een nieuw proces met shared memory en andere resources

## Types of Linux Processes

- **Init Process**
  - Het eerste proces dat gestart wordt wanneer het linux systeem boot
  - Parent proces van alle andere processen
- **Parent en Child Processen**
  - Elk user proces heeft een parent proces in het systeem
  - Meeste commandos hebben shell als parent
- **Orphan Processen**
  - Wanneer een child process gestopt wordt, wordt het parent proces geupdate door het SIGCHLD signaal
  - Wanneer het parent process afgesloten wordt voor het child process wordt het child process een orphan process met het init process als parent
- **Zombie Processen**
  - Een proces wat afgesloten is maar toch nog getoond wordt in de proces table
- **Daemon Processen**
  - Systeem gerelateerde background processen die runnen met root rechten en wachten op requests van andere processen

## Process States & Signals

- Signals zijn een vorm van interprocess communication gebruikt om een process te notifiëren over een event of conditie
- Kunnen verzonden worden door een andere proces, de kernel of door het proces zelf
- Elk signaal heeft een uniek nummer
  - SIGTERM: termination
  - SIGKILL: immediate termination
- Wanneer een proces een signaal ontvangt kan deze het signaal negeren, handelen of een default actie uitvoeren die bij het signaal hoort

## fork() en exec()

- Fork
  - Clone operatie
  - neemt het huidige proces (parent process) en cloneert dit naar een nieuw proces met een unieke PID
  - Alles wordt mee gekopieerd
    - stack, heap, file descriptors →
    - standard input, output, error
- Exec
  - Het huidige proces vervangen door een image van een andere proces

## Shell Command Execution

- Proces dat op user input wacht
- Launched een bash interpreter en het zorgt voor een environment voor de commands om te runnen
- Wanneer je een command uitvoert wordt de shell geforked naar een child proces
- Daarna zal dit child proces een exec uitvoeren om het huidige proces om te zetten in de juiste shell process image

## Process Management - context switching

1. CPU onderbreekt het huidige proces gebaseerd op een interne timer, switched naar kernel mode en geeft de controle over aan de kernel
2. De kernel legt de huidige status van de cpu en memory vast, deze is nodig voor het vorige proces terug te kunnen opstarten
3. De kernel voert taken uit die eventueel omhoog komen tijdens de vorige time slice
4. De kernel is nu klaar voor een volgend proces te laten runnen, de kernel overloopt een lijst met processen die klaar staan en kiest er een om uit te voeren
5. De kernel bereidt de memory voor en daarna de cpu voor het nieuwe proces
6. De kernel geeft hoe lang de time slice voor het nieuwe proces is door aan de CPU
7. De kernel switched de CPU terug naar user mode en geeft de controle aan de CPU om het nieuwe proces uit te voeren



## The Linux Scheduler

- Verantwoordelijk voor welk proces als volgende te runnen
- Gebruikt process priorities, CPU verbruik, process states en scheduling classes om geïnformeerde keuzes te maken
- Elk proces heeft een priority value tussen -20 (Hoogste prioriteit) en 19 (Laagste prioriteit)
- Priority based scheduling algorithm en andere factors
- Is in staat om priorities dynamisch te wijzigen op basis van system load of andere factoren
- Completely fair scheduler is meestal de default
  - Fairness algoritme om zeker te zijn dat elk proces evenveel CPU time verkrijgen

## Linux threads

- Lightweight execution contexts dat memory en code delen binnen een parent proces
  - In een browser kan elke tab een aparte thread zijn
- Elke thread heeft zijn eigen stack en program counter, maar elke thread binnen een proces deelt dezelfde heap en global variables
- Zorgen ervoor dat een programma meerdere taken simultaan kan uitvoeren
- Kunnen communiceren met elkaar via shared memory of andere IPC mechanismen
- Kunnen hun acties synchroniseren met mutexes, semaphores of ander synchronisatie primitives

## Advantages threads over processes

- **Responsiveness**
  - Wanneer een thread klaar is kan de output direct verkregen worden voor de andere klaar zijn
- **Snellere context switch**
  - Process context switching heeft meer CPU overhead nodig dan bij threads
- **Effectief verbruik van multiprocessor systeem**
  - Wanneer er meerdere threads zijn binnen een proces kunnen deze verdeeld worden over meerdere processors
- **Resource sharing**
  - Code, data en files kunnen makkelijk gedeeld worden binnen threads
- **Communicatie**
  - Is makkelijker bij threads omdat ze common address space delen

## Interrupts in Linux

- Signaal gegeven door hardware, processor of software wanneer een process directe aandacht nodig heeft
- Zorgen dat de CPU switched naar interrupt handler
  - Software routine die een specifieke taak, zoals het lezen van data uit een hardware device of het notifiëren van de kernel van een user input event
- Worden gemanaged door de Interrupt Service Routine (ISR) en de Interrupt Request (IRQ)
  - De ISR is ingeroepen om de interrupt te handelen en de bijbehorende IRQ nummer wordt hier aan door gegeven
- Kernel supports interrupt coalescing dat ervoor zorgt dat veel interrupts samen kunnen genomen worden wat de overhead vermindert

## Sockets

- Een soort van file descriptor dat processen toelaat om te communiceren met elkaar over een netwerk of tussen processen op hetzelfde systeem
- Door het aanmaken van named contact points en het aanbieden van een gestandaardiseerde manier om data te versturen en te ontvangen
- Worden aangemaakt of gemanaged door system calls
  - `socket()`, `bind()`, `listen()`, `accept()`, `connect()`, `send()` en `recv()`
- Gebruikt voor verschillende netwerkprotocollen zoals TCP/IP, UDP, ...
- Wordt vaak gebruikt voor client-server applicaties
  - De server maakt een socket, hangt het aan een poort en wacht op de client die contact legt met de socket
  - De client maakt een socket en probeert dan een connectie te leggen met de server socket
  - Wanneer de connectie tot stand is gebracht kan data overgebracht worden
- Practically
  - Een socket geconnecteerd tot een netwerk is aangemaakt aan beide kanten van de communicatie
  - Elke socket heeft een specifiek adres (Ip address + port number)

# Routing

## Terminologie

- **Routing**
  - Een pad selecteren voor netwerkverkeer in een netwerk of tussen meerdere netwerken
- **Packet Forwarding**
  - Vervoer tussen network packets van een netwerk interface naar een andere
- **Routing tables**
  - Bezitten een record voor routes naar verschillende netwerk destinations
- **IP routing**
  - Neemt aan dat netwerkadressen gestructureerd zijn en dat gelijkaardige adressen bij elkaar horen in een netwerk. Gestructureerde adressen laten een eenmalige routing table entry toe om een route naar een groep devices te representeren.
- **NAT**
  - Networking Address Translation
  - Techniek om IP-adressen te vertalen tussen verschillende netwerken
  - Private IPs omzetten naar een public IP

# Firewall

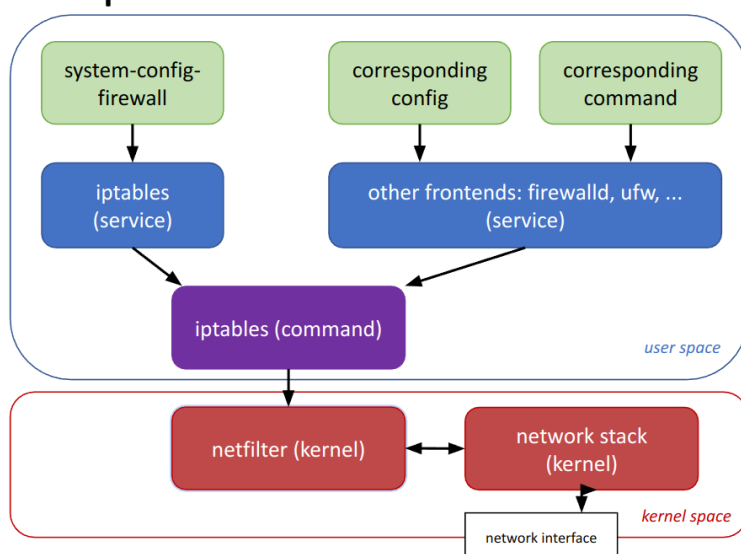
## Functies van een Firewall

- Beschermt een host systeem van unauthorized connecties
  - Blocked connecties naar services die niet public horen te zijn
  - Restrictie van connecties naar bepaalde IP ranges
- Herschrijft packet headers naar route packets tussen netwerken
  - Laat de machine toe om te fungeren als network router
  - Meeste consumer "router" devices zijn kleine computers die een firewall runnen met routing capabilities
- Geavanceerde firewall functies:
  - NAT
  - Quality of Service

## iptables en netfilter

- iptables is een user space interface naar de linux kernels netfilter systeem
  - Oplijsten van de content van een packet filter ruleset
  - Toevoegen, verwijderen of aanpassen van regels in een packet filter ruleset
  - Oplijsten, zeroing per per-rule counters in een packet filter ruleset
- netfilter implementeert een firewall en routing capabilities in de kernel wat er voor zorgt dat een linux machine zich kan gedragen als een firewall en/of router
  - Stateless en stateful packet filtering
  - NAT
  - Flexibele en extensieve infrastructuur
  - Meerdere API lagen voor 3rd party extensies

## iptables Architecture



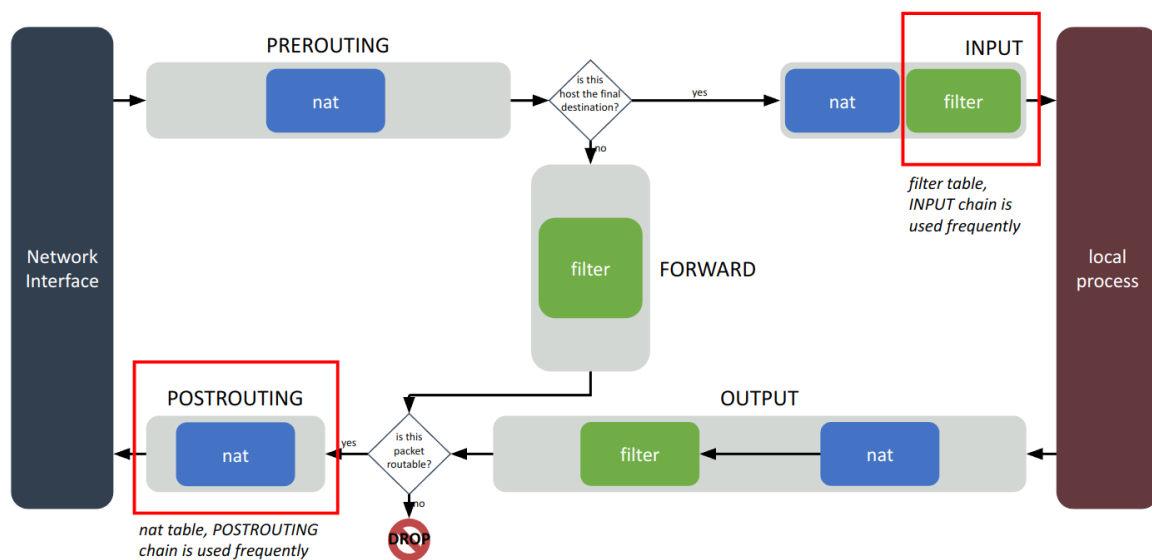
## nftables

- Gemaakt door het netfilter project
- Als tegenhanger voor iptables
- Default voor ubuntu

## Packet Processing

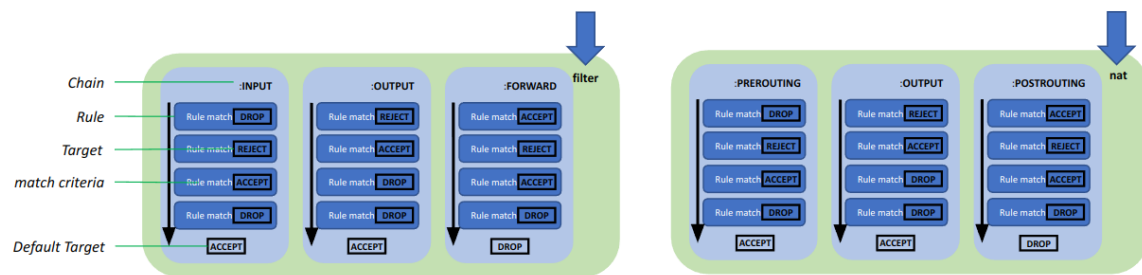
- Wanneer een packet binnenkomt in het netwerk herkent netfilter component in de kernel dit en groepeerde de packets in streams of flows
  - Stateful Packet Inspection (SPI)
    - Analyseren van packet headers en contents om connecties te tracken
- Elk pakket in een stream is geprocesst door de firewall met performance optimalizaties
  - Bijvoorbeeld: NAT rules zijn enkel bepaald voor het eerste pakket in een stream, de volgende pakketjes krijgen dezelfde processing

## Netfilter packet traversal - overview



## Tables

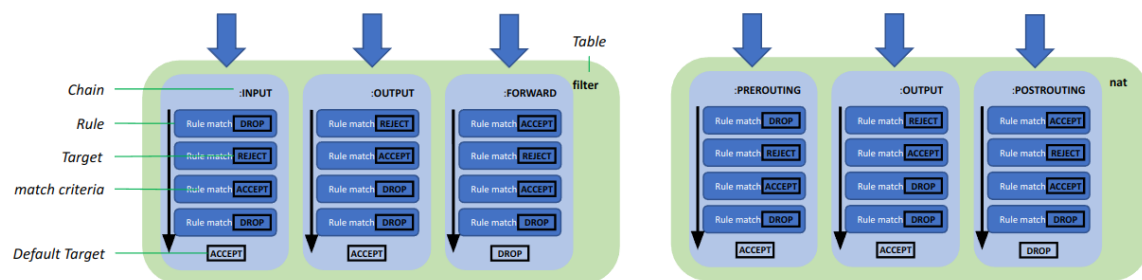
## Tables



- Data structuur dat een aantal chains bevat, er zijn meerdere tables
- default table voor iptables is "filter"

## Chains

## Chains



- Wanneer het bepaald is dat het een packet is wordt deze gematched tegen een bepaalde table, daarna gematched tegen rules die bij deze table zijn chains horen
  - Binnen een chain start een pakket van boven en wordt gematched rule-by-rule
  - Wanneer een match gevonden is jumped processing meestal naar een andere chain of target
- Wanneer een pakket aan het einde van een chain geraakt wordt de default policy voor deze chain toegepast

## iptables: Tables and Chains

- Zorgen voor statefulness van de firewall

Table	filter	nat	<i>rarely used</i> mangle	conntrack <i>tracks incoming and outgoing packets across the flow</i>	<i>rarely used</i> raw <i>provides opt-out of connection tracking with 'NOTRACK'</i>
PREROUTING chain		PREROUTING	PREROUTING	PREROUTING	PREROUTING
INPUT chain	INPUT	INPUT	INPUT		
FORWARD chain	FORWARD		FORWARD		
OUTPUT chain	OUTPUT	OUTPUT	OUTPUT	OUTPUT	OUTPUT
POSTROUTING chain		POSTROUTING	POSTROUTING		

← processing order

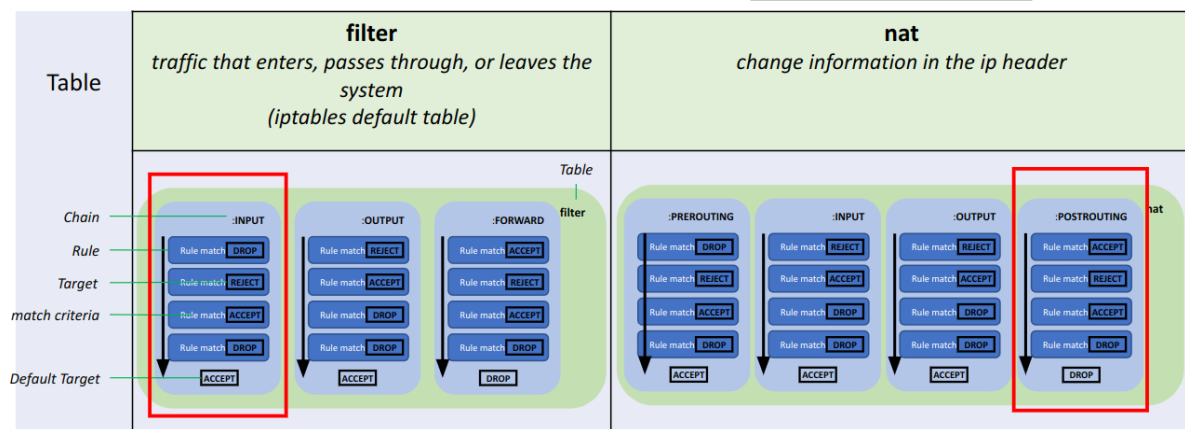
decisions to let traffic through or not ('filtering')

how and when are the source and destination of packages changed

Table	filter <i>traffic that enters, passes through, or leaves the system (iptables default table)</i>	nat <i>change information in the ip header</i>
Chains	<b>INPUT</b> <i>Handles packets destined for local sockets</i> <b>FORWARD</b> <i>Handles packets routed through this host to another destination</i> <b>OUTPUT</b> <i>Handles locally generated packets</i>	<b>PREROUTING</b> <i>Alters packets as they arrive from the network interface</i> <b>OUTPUT</b> <i>Alters locally-generated packets before routing them</i> <b>POSTROUTING</b> <i>Alters packets as they are about to be sent out from the host, <b>after</b> routing decision has been taken</i>

decisions to let traffic through or not ('filtering')

how and when are the source and destination of packages changed



## Rules

- Processen gebeurt in iptables door het matchen van packets aan rules
  - Als een pakket een rule matched, wordt de rule toegepast
  - Wanneer er geen matching properties gegeven zijn wordt de regel altijd toegepast
- Rules specificeren meestal een target waarnaar het proces springt als de rule matched

## Special Targets

- **ACCEPT**
  - Pakket wordt geaccepteerd en gaat verder in het proces door de volgende table
- **DROP**
  - Pakket wordt gediscard
  - Geen response naar de verzender
- **REJECT**
  - DROP maar stuurt een error message pakket terug naar de verzender
    - Handig voor sommige protocollen als TCP
    - Niet handig voor security reasons
      - Erkent het bestaan van de server

## Stateless vs stateful firewall

- Stateless firewall
  - Packet filtering
  - Meestal enkel in de network layer (OSI)
    - Soms protocollen in een hogere laag
- Stateful firewall
  - Alles wat een stateless firewall doet
  - Tracked ook wanneer een pakket al eens gezien is in een sessie
    - Voegt access policies toe aan packets op basis wat al gezien is geweest voor een bepaalde connectie
- netfilter/iptables
  - Zijn stateful
  - Dit door conntrack table



## NAT masquerading met conntrack table

- **Connection tracking (conntrack)**
  - Belangrijke kernel feature gebruikt door linux machines om TCP connections die binnenkomen en buitengaan bij te houden
- Zorgt ervoor dat pakketjes dat ge-NAT zijn, verzonden kunnen worden naar de exacte interne machine die de connectie tot stand bracht
- **conntrack table**
  - Table waarin de complete UDP/TCP connectie status gestored is
    - IPs
    - Protocols
    - Poortnummers
    - Status van de connectie
- Gebruikt bij NAT en Stateful firewalls

## Default Packet-Filtering Policy

- Een firewall is een device om access control policies te implementeren
- 2 basis manier om de default policy te configureren
  - **Default DENY** en specifieke pakketten toelaten
    - Meest aanbevolen
    - **Veilig**
    - Je moet het communicatie protocol voor elke service die je wil toelaten begrijpen
    - Meer werk
  - **Default ACCEPT** en specifieke pakketten weigeren
    - Gemakkelijker om initieel op te zetten
    - Je moet er aan denken om elke service te blokkeren tegen misbruik
    - Configuratie en onderhoud is meer werk
    - **Minder veilig**

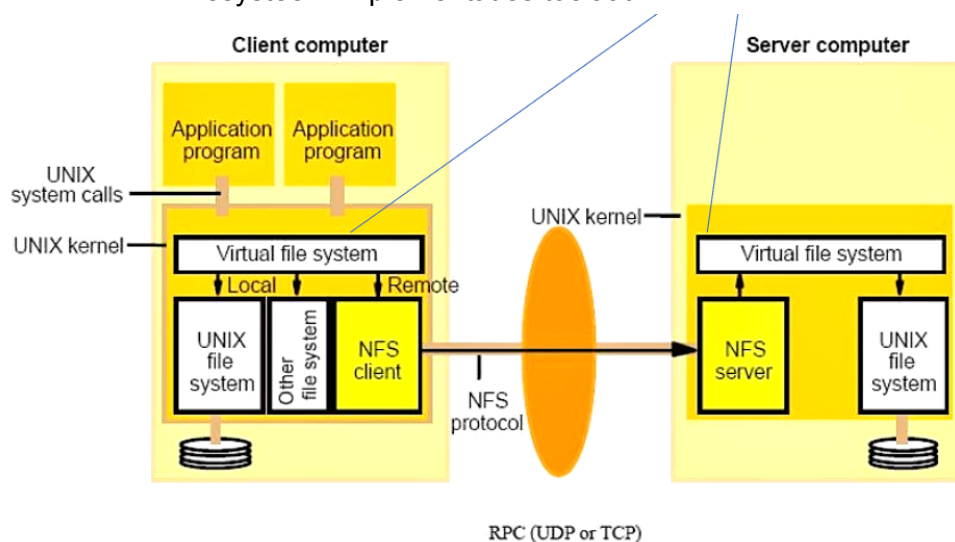
# NFS

## Network File System (NFS)

- Protocol voor gedistribueerd file systeem
- Client kan via een netwerk toegang krijgen tot remote file systemen via een local mount point

## NFS Architecture

- Linux virtual file system (vfs)
  - Software layer in de kernel dat de filesystem interface biedt aan user space applicaties
  - Biedt ook een abstractielaag aan binnen de kernel die verschillende filesystem implementaties toelaat



## NFS Networking

- NFSv3
  - UDP
  - Sneller
  - Makkelijker om op te zetten
- NFSv4
  - TCP
  - Betrouwbaarder
  - Meer security → Kerberos support
  - File locking support
  - Complex

## Quorum

- meer dan de helft van de nodes vormt een quorum of absolute meerderheid. De cluster is available zolang de available nodes quorum hebben.

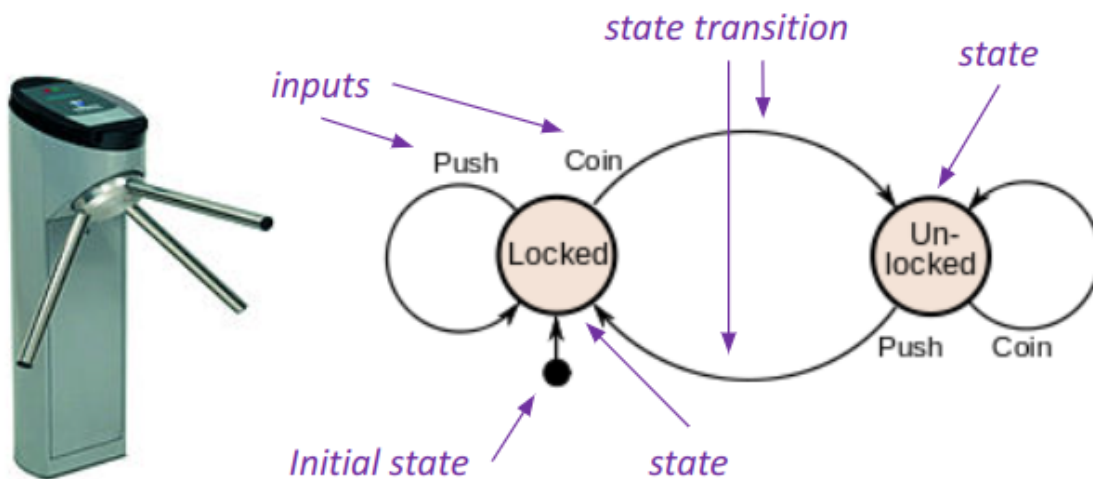
# Load Balancers

## State, Stateful en Determinism

- **Stateful systems**
  - Designed om voorgaande events of user interacties te onthouden
  - De onthouden informatie noemt de state van het systeem
  - bv. TCP
- **Stateless systems**
  - Hebben geen state
  - bv. HTTP en REST
- **Determinism**
  - De mogelijkheid van een systeem om dezelfde output te produceren wanneer de conditie gelijk is
  - Een algoritme dat altijd dezelfde output genereert met dezelfde input

## Finite State Machine (FSM)

- Abstracte machine gebruikt om algoritmes, systemen of protocollen te illustreren
- Kan van de ene state naar de andere veranderen als antwoord op bepaalde inputs
- Defined door:
  - Een lijst van states
  - Initial state
  - De input die elke transitie triggered



*State diagram for a turnstile*

Current State	Input	Next State	Output
<b>Locked</b>	coin	Unlocked	Ontgrendelt het tourdraaihekje, zodat de klant erdoor kan.
	push	Locked	None
<b>Unlocked</b>	coin	Unlocked	None
	push	Locked	Wanneer de klant is doorgedrongen, sluit het draaihekje

*State transition diagram for a turnstile*

# REST - REpresentational State Transfer

- Gebruikt een subset van HTTP
- Vaak gebruikt voor interactieve webapplicaties
  - Offert web resources in textueel formaat
    - Laat toe om gelezen en aangepast te worden met behulp van een stateless protocol
  - Antwoord op requests naar een URI van een resource met een payload in HTML, XML, JSON of andere formaten
  - Zorgt voor interoperabiliteit tussen webservices
  - Makkelijk schaalbaar omdat het stateless is
- Voordelen
  - Splitst client en server verantwoordelijkheden
  - Verbeterd zichtbaarheid, betrouwbaarheid en schaalbaarheid van de applicatie
  - Vermindert koppeling tussen systemen
- Limitaties
  - Gebrek aan standaardisatie voor error handling
  - Potentiële security kwetsbaarheden wanneer het niet goed is geïmplementeerd

## REST Components

- Resources - **endpoint**
  - documenten, afbeeldingen, videos, informatie
- Request Verbs - **method**
  - HTTP methods
    - GET (read)
    - POST (write)
    - PUT (update)
    - DELETE
- **Request Headers**
  - Bijkomende instructies in de request
- **Request Body**
  - Bij een POST of UPDATE wordt er data verzonden met de request
- **Response Status codes**
  - HTTP codes die bij de response komen
    - 200: OK
    - 403: FORBIDDEN

The diagram shows a curl command with four labels pointing to its parts: 'method' points to '-X POST', 'endpoint' points to 'https://requestb.in/1ix963n1', 'headers' points to '-H "Content-Type: application/json"', and 'body' points to the JSON payload.

```
curl -X POST https://requestb.in/1ix963n1 \
-H "Content-Type: application/json" \
-d '{
  "property1": "value1",
  "property2": "value2"
}'
```

## Session-based authentication

- Stateful
  1. De server maakt een sessie aan voor de user nadat deze inlogt
  2. De session ID wordt bewaard in een cookie in de browser van de user
  3. Zolang de user ingelogd blijft wordt de cookie mee verzonden met elke request naar de server
  4. De server vergelijkt de session ID van de cookie met de sessie informatie die bewaard is in memory en verstuurd een gepaste response

## Token-based authenticatie

- Stateless
- Veel RESTful webservices gebruiken tokens zoals JSON Web Tokens (JWT) voor authenticatie in plaats van sessies
  1. De server genereert een token met een secret en verstuurd het naar de client nadat deze geauthenticeerd is
  2. De client bewaard de token lokaal en voegt het toe aan de header bij elke request
  3. De server valideert de token met elke client request en verstuurd een gepaste response gebaseerd op de tokens geldigheid

## Load Balancer

- Een apparaat of software die taken verdeeld over een groep resources om het proces sneller te laten verlopen
- Biedt een single internet service aan van meerdere servers, ook wel server farm genoemd
- Vaak gebruikt in HTTP request management
  - Veel requests simultaan behandelen

## Typische features

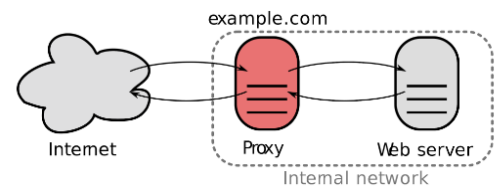
- Asymmetrische load
  - Een grotere portie workload geven aan een specifieke server
- Priority activation
  - Wanneer available servers onder een treshold vallen of de load te groot wordt nieuwe machines opspinnen
- Health checking
  - Check op de applicatie layer als failed servers verwijderen van de pool
- Session persistence
  - sticky sessions

## Sticky Session

- Feature van laag 7 load balancers
- Inkomende requests voor een bepaalde sessie altijd naar dezelfde server sturen
- Voor webapplicaties die nood hebben aan persistente sessies
- Kunnen voor oneven load zorgen
- Op te lossen door gebruik te maken van stateless services

## Reverse proxy

- Service dat voor een of meerdere servers zit, die requests accepteert voor resources die op de server staan
- Voor de POV van de client is de proxy de webserver
- Perfect punt voor traffic control en performance optimalisatie omdat al het verkeer hier langs passeert
- Single point of connection



## Reverse proxies - typical features

- Compression
  - Comprimeren van server responses voordat deze naar de client gaan
  - Verminderd nodige bandbreedte
  - Versneld transport snelheid
- SSL termination
  - Door het decrypteren van inkomende requests en het encrypten van server responses maakt de proxy resources vrij op de backend server
- Caching
  - Wanneer gelijkaardige requests verstuurd worden kan de proxy een antwoord terug geven zonder dat de backend hier aan tussenkomst

A **reverse proxy** accepts a request from a client, forwards it to a backend origin server that can handle it, and then sends the server's response back to the client. A reverse proxy is often used in combination with a load balancer.

- **HTTP caching / web acceleration** - stores static content so that some requests don't have to go to the servers
- **Load balancing** – distributes requests across a backend pool of servers based on certain algorithms (see later slide)
- **URL rewriting** – rewrites the URL in each incoming request to match the internal location of the requested resource
- **Security** hides the existence and characteristics of origin servers and the internal network
- **HTTP security** – hides HTTP error pages, removes server identification headers from HTTP responses, and encrypts cookies
- **Application firewall** – protects against denial-of-service (DoS) and distributed denial-of-service attacks (DDoS)
- **TLS acceleration**: encrypts/decrypts SSL (with or without special hardware) on the proxy
- **Client authentication** - authenticates users via authentication sources before allowing them to proceed
- **HTTP compression** – reduces the size of HTTP objects using gzip compression, available in all modern web browsers
- **TCP buffering / spoon feeding** - buffers the server response and feeds it in small portions to slow clients so that the origin server becomes available faster
- **Rate limiting** - responds with status code 429 (Too Many Requests) if a client has made too many requests within a certain time frame
- **Logging**

## Load Balancing - OSI layers

- Layer 4 ( Transport)
  - TCP/UDP packets
  - Routing requests
  - Kunnen schalen om grote aantallen requests op te vangen
- Layer 7 (Applicatie)
  - Applicatie load balancers
  - Routing beslissingen gebaseerd op de content van applicatie verkeer
  - Goede keus voor gecontaineriseerde microservices-based architectuur

## Load Balancing Algoritmes

- Round Robin
  - Beurt om beurt
  - Handig wanneer elke server dezelfde resources heeft
- Least Connections
  - Minst actieve connecties
- IP Hash
  - Gebruikt de clients IP om te bepalen welke server
  - Elke client wordt elke keer naar dezelfde server gestuurd
- Random
- Weighted Round Robin
  - Beurt om beurt maar elke server heeft een gewicht
- Least Response Time
  - Verkeer gaat naar de server die het snelst reageert



# eBPF

- Extended Berkeley Packet Filter
- Technologie die het toe laat om custom code te schrijven dat dynamisch de manier waarop de Kernel zich gedraagt aan te passen
- Platform om security, observability en networking tools op te bouwen
- Geëvolueerd van BPF wat gemaakt was voor efficiënte packet filtering
- Kan gebruikt worden om elk deel van de Kernel en user space programma's te instrumenteren

## Belang

- Voegt nieuwe functionaliteit toe aan de kernel zonder kernel modules die werken zonder de kernel te rebooten of opnieuw te compilen
- Kan overal inhaken op de kernel
- eBPF programma's zijn sandboxed
- Veel gebruikt in cloud native omgevingen

## eBPF networking

- XDP (eXpress Data Path)
  - high-performance data path gebruikt om network packets te versturen en te ontvangen aan hoge snelheden door de meerderheid van de OS networking stack over te slaan

# Systemd

- open-source software suite dat voor systeem componenten zorgt
- Service configuratie en gedrag over linux distributies
- Biedt een systeem en service manager die als PID 1 runned en de rest van van het systeem start
- Biedt vervanging aan verschillende daemons en utilities
  - Device management
  - Login management
  - Network connection management
  - Event logging

## Issues met traditionele init systemen

- Slow boot times
- Service dependencies
  - Het managen van deze dependencies kon complex en error-prone zijn leidend tot instabiliteit
- Limited logging en monitoring
  - Moeilijk om problemen te diagnosticeren en te troubleshooten
- Inconsistent system behavior
  - Verschillende linux distros hadden verschillende init systemen
- Limited security features
  - Moeilijk om systeemprocessen en resources te isoleren

## Advantages systemd

- Faster boot times
  - Parallelized service startup en dependency management
- Centralized en unified management van system services, sessions en devices
- Verbeterde security
  - Process sandboxing en user isolation
- Extensive logging

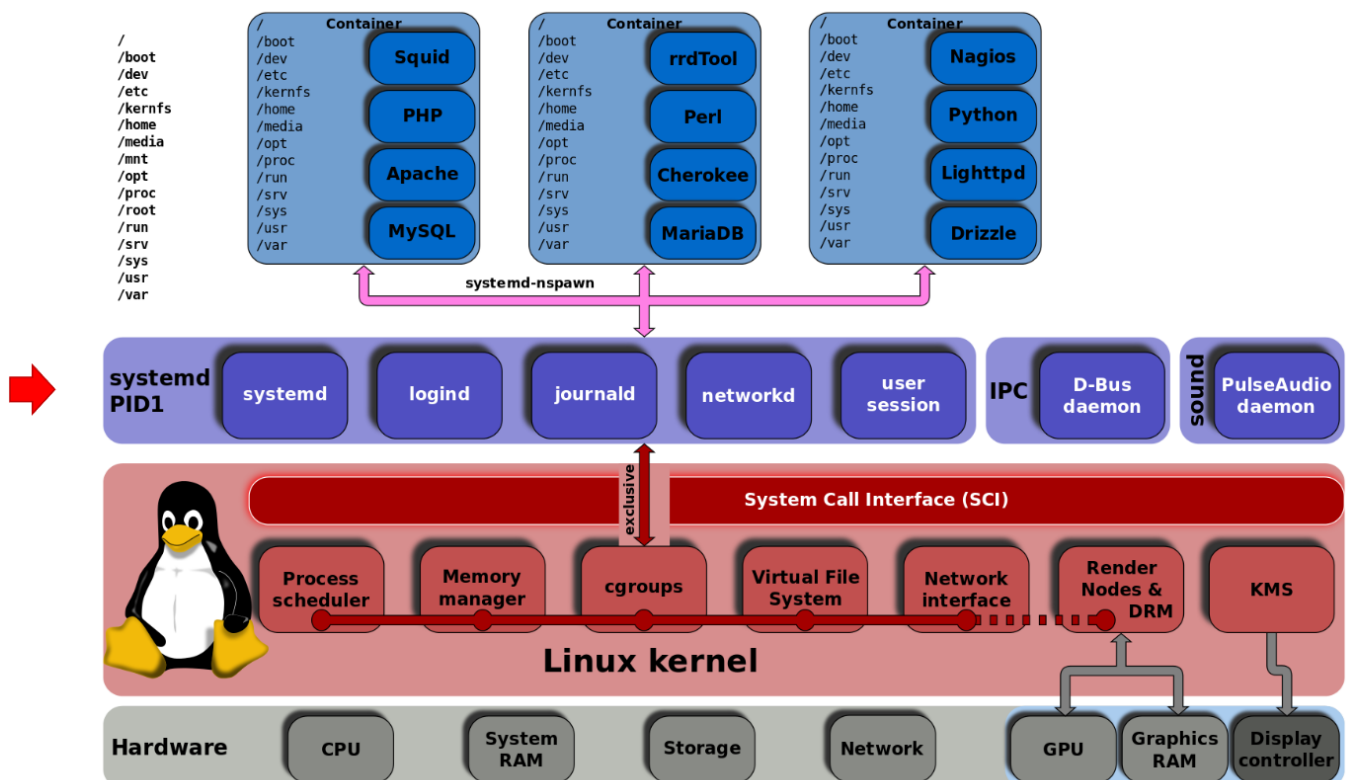
## Disadvantages systemd

- Complex, steep learning curve
- Concerns over system stability en reliability
  - systemd failures en bugs
- Beperkte support voor non-linux platformen

## Features van systemd

- Aggressive parallelization capabilities
- socket en D-Bus activatie voor het starten van services
- on-demand starten van daemons
- Houd alle processen bij door het gebruik van linux cgroups

- cgroups, control groups is een kernel feature dat accounts limiteert en de resources isoleert
- Houd mount en automount points bij
- Elaborate transactional dependency-based service control logic
- Logging daemon (journald)
- Controle over basis systeemconfiguratie zoals hostname, date, locale
- Houd een lijst bij van
  - logged-in users
  - running containers
  - VMs
  - system accounts
  - runtime directories en settings
- Daemons voor het handelen van simpele network configuration, network time synchronization, log forwarding en name resolution



## Message Bus communication models

- Messages op de bus hebben een gedefinieerde structuur
- busses kunnen 2 modes voor het interchanging van messages hebben
  - One-to-one Request-Response
    - synchroon
  - Pub/Sub (Publish/Subscribe) pattern
    - Asynchroon
    - Publishers communiceren met subscribers asynchroon door het broadcasten van events als messages
    - Messages worden gegroepeerd per topic
    - Subscribers kunnen subscriben op topics en verkrijgen zo alle messages voordat topic

## D-Bus (Desktop Bus)

- Simpele efficiënte manier voor applicaties en systeem componenten om messages en signalen te delen
  - Laat communicatie tussen meerdere processen toe
  - IPC running gelijktijdig op dezelfde machine
- Gebruikt door desktop environments
- open source referentie implementatie libdbus dat het wire-protocol implementeerd door gebruikt te maken van linux sockets
- Speciaal daemon proces dbus-daemon speelt de bus role waar dan de overige processen naar connecteren
- Meerdere bussen
  - Single system bus
    - Toegankelijk voor alle users en processen
    - Biedt toegang tot system services
  - Session bus
    - Voor elke user 1
    - Biedt desktop services voor user applicaties in dezelfde desktop session

## Systemd Units

- Configuratie files dat systeem resources en services defineert managed door de systemd init system
- Meerdere types van units, elk voor een specifieke use case
- Meest voorkomende types
  - Service units
    - define system services en daemons
  - Target units
    - define dependencies en service groepen die samen gestart of gestopt worden
  - Timer units
    - define scheduled tasks/jobs
  - Device units
    - define hardware devices en hun attributen
  - Mount units
    - define file system mounts
  - Socket units
    - define network sockets

## Systemd timers

- Timers en cronjobs worden gebruikt om taken te plannen en te automatiseren
  - systemd timers zijn flexibeler en krachtiger
  - cronjobs simpeler
- systemd heeft zijn eigen timer systeem alternatief voor het traditioneel cron systeem
- systemd timers zijn units die aangeven wanneer een bepaalde taak moet runnen
- Gelijkaardig aan services maar in plaats van het runnen van een service triggeren ze een actie of commando
- Timers kunnen geconfigureerd worden om eenmalig, herhaaldelijk, tijdens specifieke intervallen, tijden te runnen

## Systemd targets

- Collectie van systeem services en resources dat een specifieke system state moeten bereiken
  - multi-user mode of graphical user interface mode
- Groep van services die gelijktijdig gestart en gestopt moeten worden
- Target units kunnen dependencies hebben op andere units
  - Maakt het makkelijker om service startup/shutdown te managen
- Gelijkaardig aan runlevels in traditionele init systemen maar flexibeler en krachtiger
- Types
  - Basic targets
    - Fundamentele targets, inclusief boot, shutdown, rescue en emergency targets
  - Service targets
    - Starten of stoppen van een specifieke groep van services
  - Slice targets
    - Gebruikt voor het groeperen van processen gebaseerd op hun resource gebruik
  - Multi-user targets
    - Providen een volledig systeem met een graphical user interface
  - Andere
    - network, remote-fs, time-sync targets

## Praktische targets

- graphical.target
  - Repreenteert de runlevel waar het systeem de GUI runned
- multi-user.target
  - Repreenteert een runlevel waar het systeem in een niet graphical mode met networking enabled
- rescue.target
  - Repreenteert een runlevel voor het uitvoeren van system maintenance en recovery taken in single-user mode
- reboot.target
  - Repreenteert een runlevel voor het uitzetten en rebooten van het systeem

## Runlevel

- Define de state van een systeem op verschillende punten in tijd zoals tijdens het boot process of tijdens shutdown

## Systemd logs

- journald is een gecentraliseerd log systeem voor snelle, efficiënte en betrouwbare logging van events en messages
  - Log messages in binary formaat voor snellere en efficiënte filtering
  - Automatische rotatie en compressie om disk space te besparen

# Booting linux

## UEFI

- Unified Extensible Firmware Interface
- Low-level software dat start bij het booten van je PC
- UEFI kan wat BIOS kan +
  - Booten van een disk groter dan 2TB door middel van GPT
  - De user een GUI geven
  - Extra security, platform independence, consistentie en performance
- UEFI Compatibility Support Module (CSM)
  - Laat een EFI-based computer toe om BIOS-mode boot loaders te gebruiken
- Secure Boot
  - Enforces signature checking van het boot proces
  - Wanneer het systeem start checkt de firmware elke signature, als alles goed is start het systeem op
    - Protectie tegen rootkits, unauthorized software updates, ...
- GUID Partition Table (GPT)
  - Vervangt Master Boot Record (MBR) partition scheme
    - > 2TB en >4 partities (tot 128)
- Consistent variables en services
- Modular en Extensible
  - Modules kunnen worden toegevoegd, verwijderd of ge update
- Improved Boot Performance

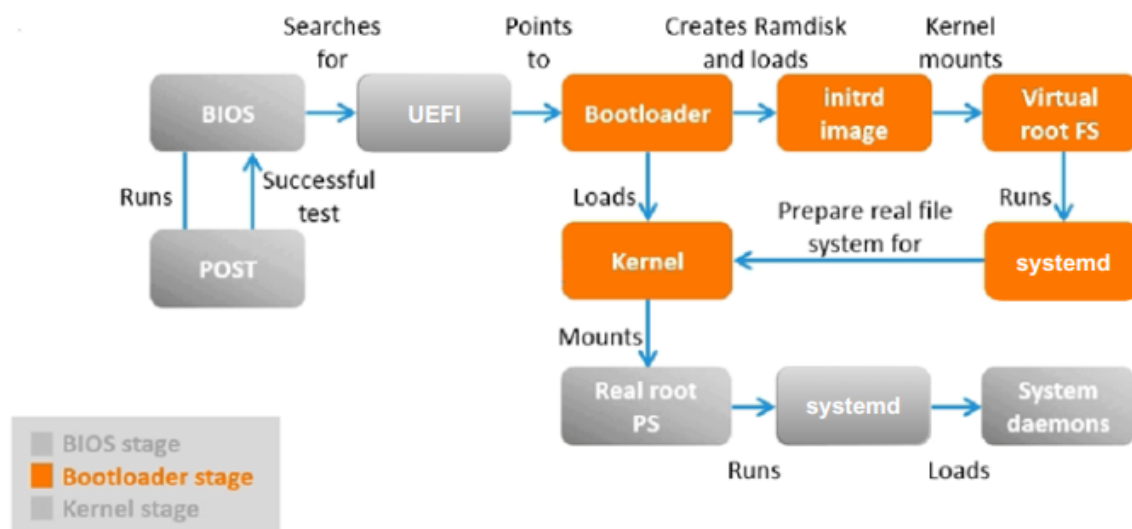
# Enhancing Bootloader Functionality and Flexibility

- **EFI executables**
  - Definieert door UEFI specification
  - Common format
    - Versimpeld het schrijven van bootloaders voor UEFI systemen
  - Type binary file gebruikt in het UEFI boot proces
  - Designed om direct uit te voeren door de UEFI firmware
  - Gedragen zich als initial boot loaders
  - common examples
    - **BOOTx64.EFI**
      - Common fallback bootloader voor x86-64 systemen
    - **GRUBx64.EFI**
      - GRUB bootloader
      - Common voor linux distros
      - Verantwoordelijk voor het laden van de linux kernel en initial ramdisk (initrd) en het starten van het bootproces
    - **Windows Boot Manager (bootmgfw.efi)**
- **EFI system partitions**
  - bevat EFI executables
  - klein, dedicated FAT-formatted partitie
- **UEFI boot manager**
  - Firmware policy
  - Laat gebruikers toe om boot menu entries aan te passen
  - Verantwoordelijk voor key tasks:
    - Maintaining Boot Entries
      - Lijst van boot entries die verschillende boot opties representeren met informatie en plaats van de loader
    - Defining Boot Order
      - Specificeert de volgorde in welke boot entry uitgevoerd worden
    - Executing Boot Entries
      - De bootloader volgt de boot order een voor een en gaat door naar de andere als een boot entry failed
    - Handling Fallback Mechanisms
      - Wanneer een boot entry niet available is kan de boot manager kijken voor een boot loader in een vooraf gedefinieerde plek
    - BIOS Compatibility
      - De boot manager kan ook boot entries handlen die BIOS compatibility mode triggeren zodat devices of OS kunnen booten met legacy BIOS



## Linux Boot Loader Stage

- The boot loader zal de gebruiker een lijst met menu entries geven die elk een OS of kernel versie voorstellen
- Wanneer je een optie selecteert om linux te starten, laad het en decompressed de linux kernel in memory
  - systemd-boot leest de instellingen van EDI partitie en laad de kernel de initial RAM disk (initrd) in memory
  - De kernel begint met het uitvoeren en initialiseren van verschillende subsystemen zoals memory management, process scheduling, device drivers en meer
  - Wanneer de kernel voldoende geïnitiliseerd is zoekt het voor het init proces, dit is het eerste user-space proces dat uitgevoerd wordt door de kernel








# Software Licenses

## Open-Source Software

- Type van software waarbij de code publiek toegankelijk is en veranderd en uitgebracht kan worden door iedereen
- Community driven development
- Software kan aangepast worden naar personal preference
- Vaak uitgebracht onder open-source licenses

## Software license types

- Software licenses duiden aan hoe software gebruikt mag worden, hoe het mag worden aangepast en hoe het mag worden uitgebracht
- **Copyright license**
  - Default type of license
  - Exclusieve rechten, creator dictates
- **Copyleft license**
  - Open-source license
  - Forks moeten onder dezelfde licentie uitgebracht worden
- **Permissive license**
  - Vrij gebruik, modificatie en distributie
- **Creative Commons**
  - Geen specifieke software license
  - Veel gebruikt voor creative works
    - Afbeeldingen, videos, audio

 MIT	 Copyright	 Copyleft	 Permissive	 Creative Commons
What is a user allowed to do with the code?	What creator dictates	What user wants under certain rules	What user wants with a few restrictions	What user wants without restrictions
Clause of the use	As creator dictates	Derivative work must be attributed to creator, open-source and copyleft	Derivative work must be attributed to a creator	Derivative work must be attributed to a creator
Source code	As creator dictates	Must be open	Don't have to be open	No specific terms about the distribution of source code
Is creator liable for bugs?	✓ YES	✓ YES	✗ NO	✗ NO
Re-licensing	As creator dictates	Derivative work cannot be released as proprietary software	Derivative work can be released under another license or as proprietary software	Derivative work can be released under another license or as proprietary software
Commercial restrictions	As creator dictates	Permitted	Permitted	Permitted

## GPL (GNU General Public License)

- Veel gebruikte open-source license
- Copyleft license

### GPLv2

- Verder gebouwde en verduidelijkte versie van GPLv1
- Provisions voor patent protection en internationalisatie
- Linux kernel

### GPLv3

- Verder gebouwd op GPLv2
- Nieuwe provisions voor digital rights management, patent protection en software patents
- Veel kritiek

### GPLv2 vs. GPLv3

- **v2** wordt beschouwd als meer permissive dan **v3**, deze wordt meer restrictive beschouwd
- GPLv2 software kan niet overgaan naar GPLv3

# Containers

## Container

- Portable package voor apps
  - Alle componenten en dependencies zitten in een image en kan worden hergebruikt
- Geïsoleerd proces
- Ephemeral
  - Limited lifecycle
- Immutable
  - Verandert niet
- Stateless
  - Houd geen state bij
- Waarom?
  - Portability
  - Agility
    - Shipping
    - DevOps
  - Scalability

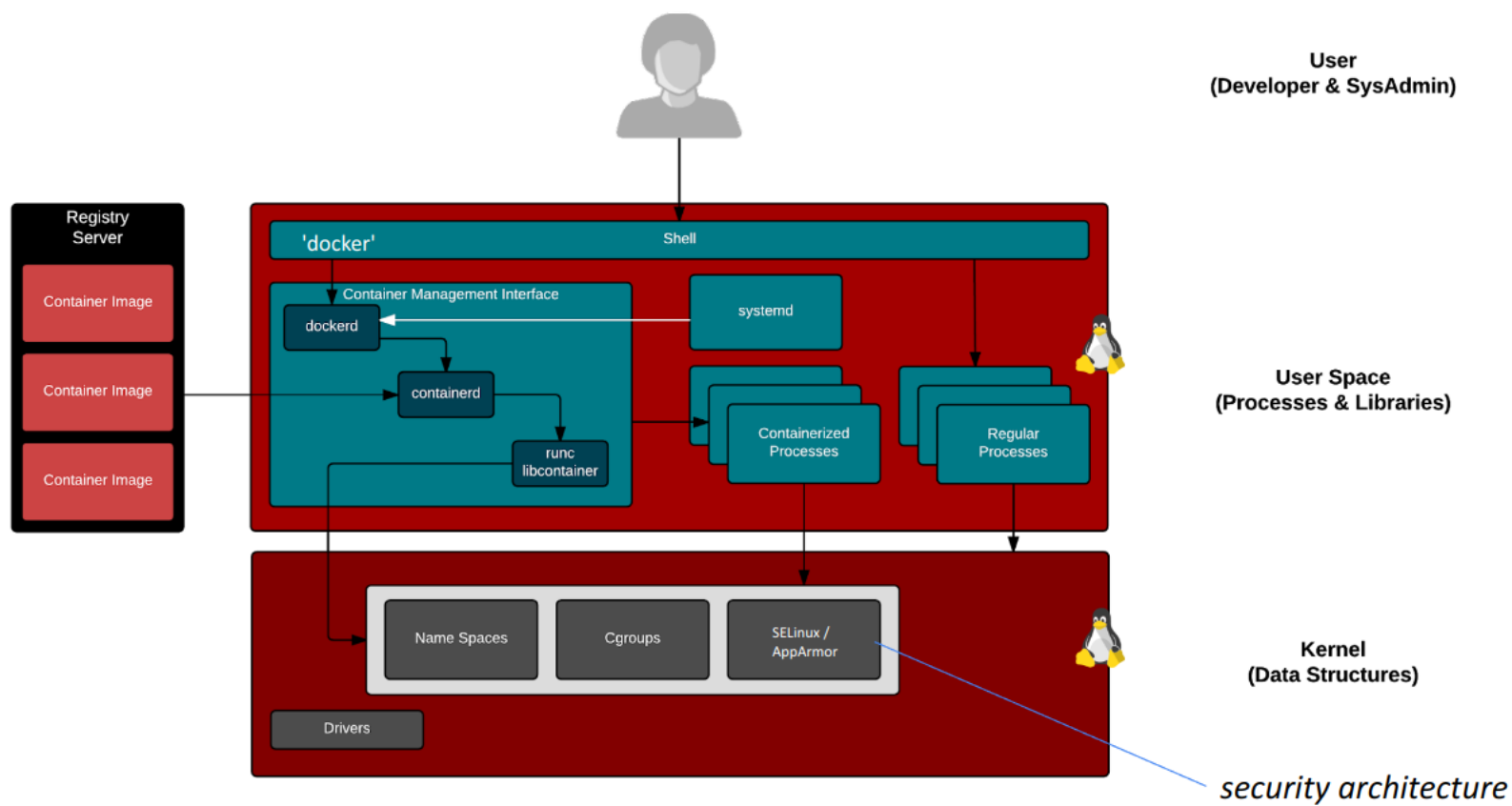
## Containers in de linux kernel

- Een container is een proces
- Linux kernel support
  - Process namespace
    - pid: process isolation
    - net: managing network interfaces
    - ipc: IPC resources
    - mnt: managing file system mount points
    - uts: isolating kernel en version identifiers
  - cgroups
    - capabilities
    - resource limits
  - union file system

# Container Image

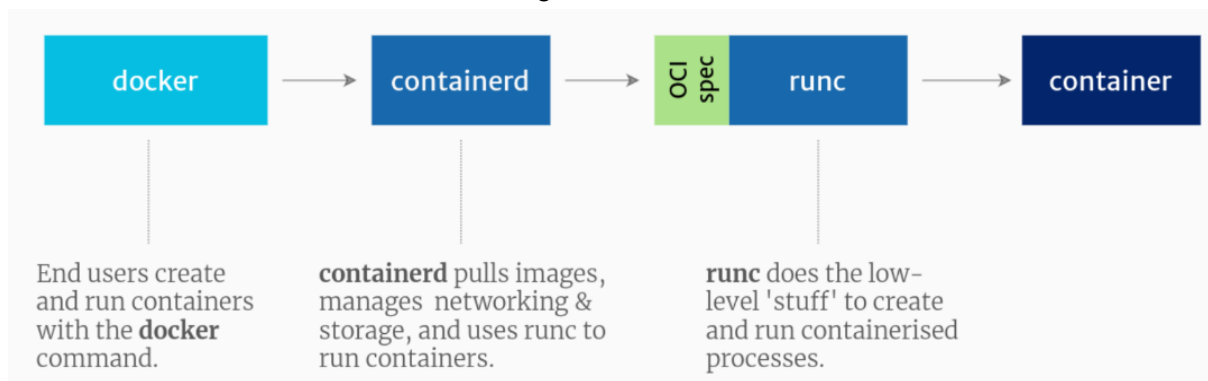
- Een file
- hiërarchische, dependent layers
  - app
  - sshd
  - ubuntu
  - scratch
- Kunnen gedeeld worden
- Offered als een union file system
- Een container is de runtime instance van een container image

## Docker runs in user space



# Open Container Initiative (OCI)

- Lightweight project voor het doel om industrie standaarden te maken rond container formats en runtime
  - OCI Runtime Specification (runtime-spec)
    - Outlines hoe een file system bundle te runnen dat is uitgepakt op een disk
    - Download een OCI image en unpack die image in een OCI Runtime file system bundle
    - De OCI Runtime Bundle wordt gerunnen door een OCI runtime: runC
    - Docker gebruikt runC om containers te runnen
      - runC gebruikt libcontainer om low-level container management te implementeren
  - OCI Image Format Specification (image-spec)
    - Docker images



# Kubernetes

## Definitie

- Open-source container orchestration tool
- Helpt bij het managen van containerized applications
- Cloud Native Computing Foundation

## Problems solved door Kubernetes

- Trend van monolieten naar Microservices
- Populariteit van container gebruik
- Honderden containers managen

## Monolithic Applications

- Problemen
  - Code maintainability
  - Opnieuw deployen van de applicatie bij een kleine update
  - Scaling problematisch

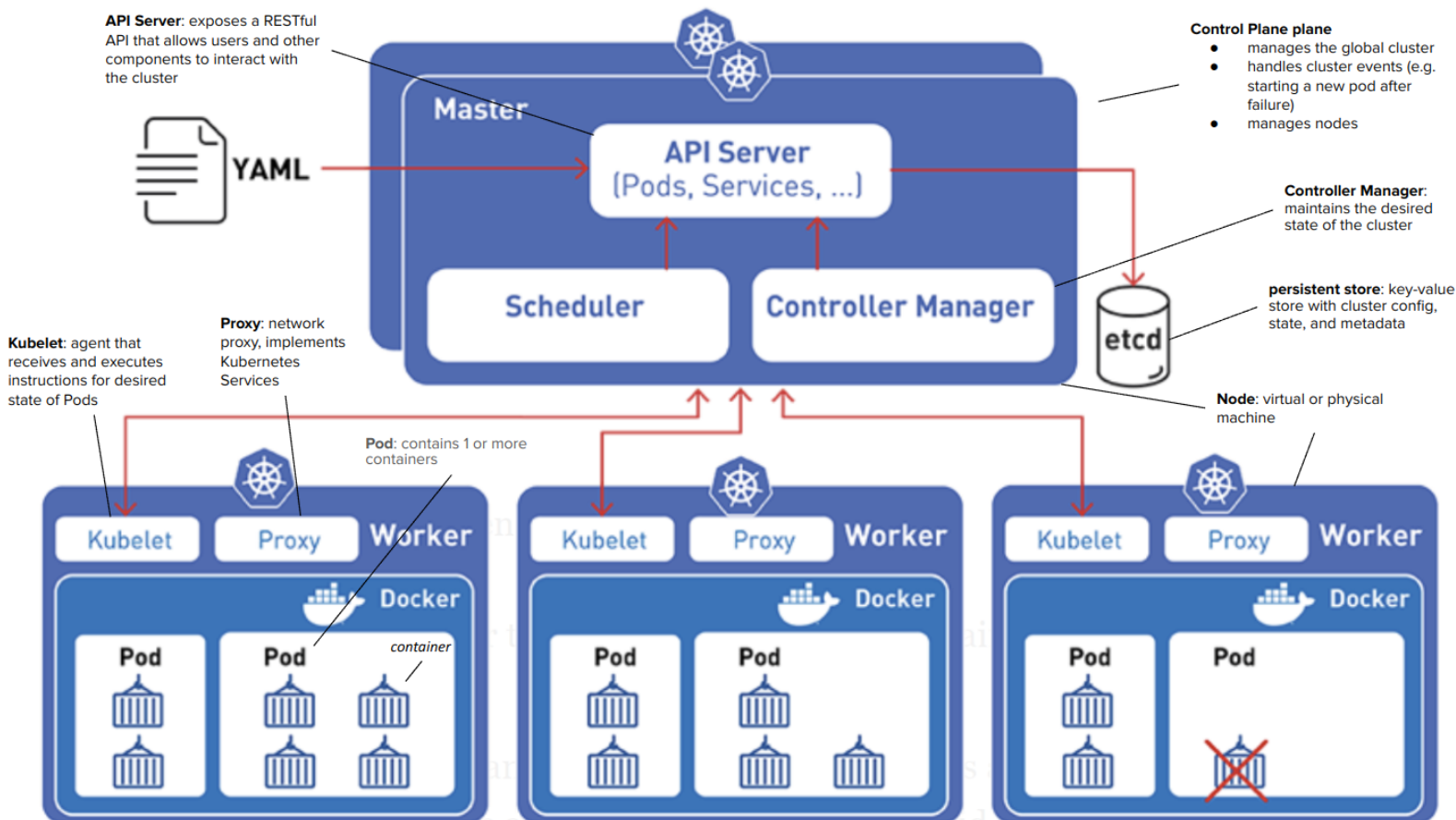
## Microservices Architectuur

- Structureert een applicatie als een collectie van services
  - Highly maintainable
  - Los van elkaar deploybaar
  - Zorgt voor snelle, frequente en reliable delivery van grote complexe applicaties
  - Handig voor Agile Scrum proces

## Features

- Service discovery en load balancing
- Storage orchestration
- Automatische rollouts en rollbacks
- Je kan de desired state opvragen en aanpassen
- Automatic bin packing
- Self-healing
- Secret en configuration management

# Terminologie



## The Declarative Model

- Beschrijf de end state die je wil bereiken → desired state, in een manifest file
- Wanneer de observed state verschilt van de desired state zal kubernetes acties ondernemen om de desired state te bereiken

## Kubernetes Service

- Abstracte manier om een applicatie te exposen die op een paar pods aan het runnen is als een network service
- Met kubernetes moet de applicatie niet aangepast te worden om een onbekende service discovery mechanismen te gebruiken
- Kubernetes geeft Pods hun eigen IP en DNS naam en kan hier tussen load balancen