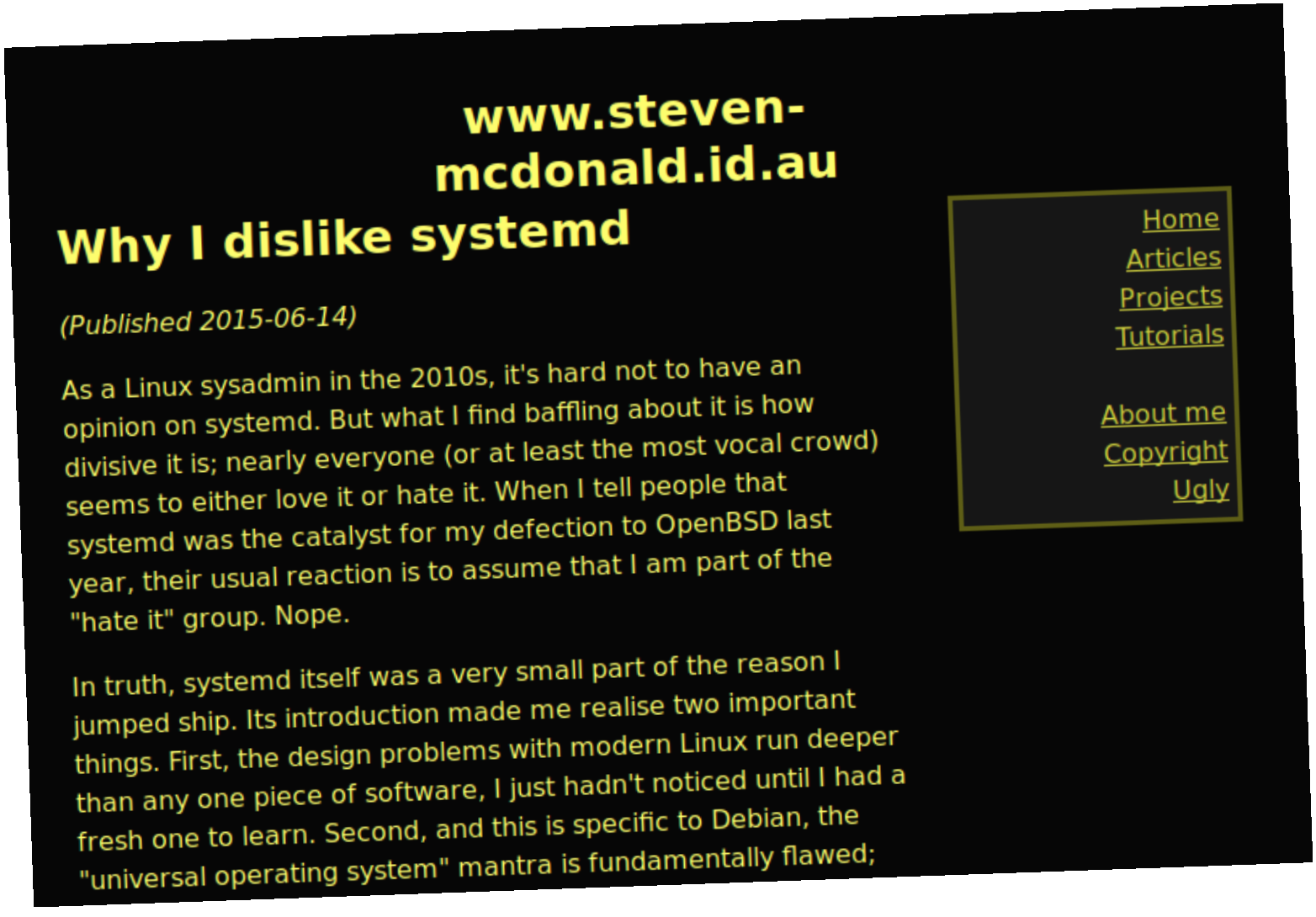
19 November 2014

Debian führt Systemd ein [colourized]

[17], [11], [5], [10]



www.steven-mcdonald.id.au

Why I dislike systemd

(Published 2015-06-14)

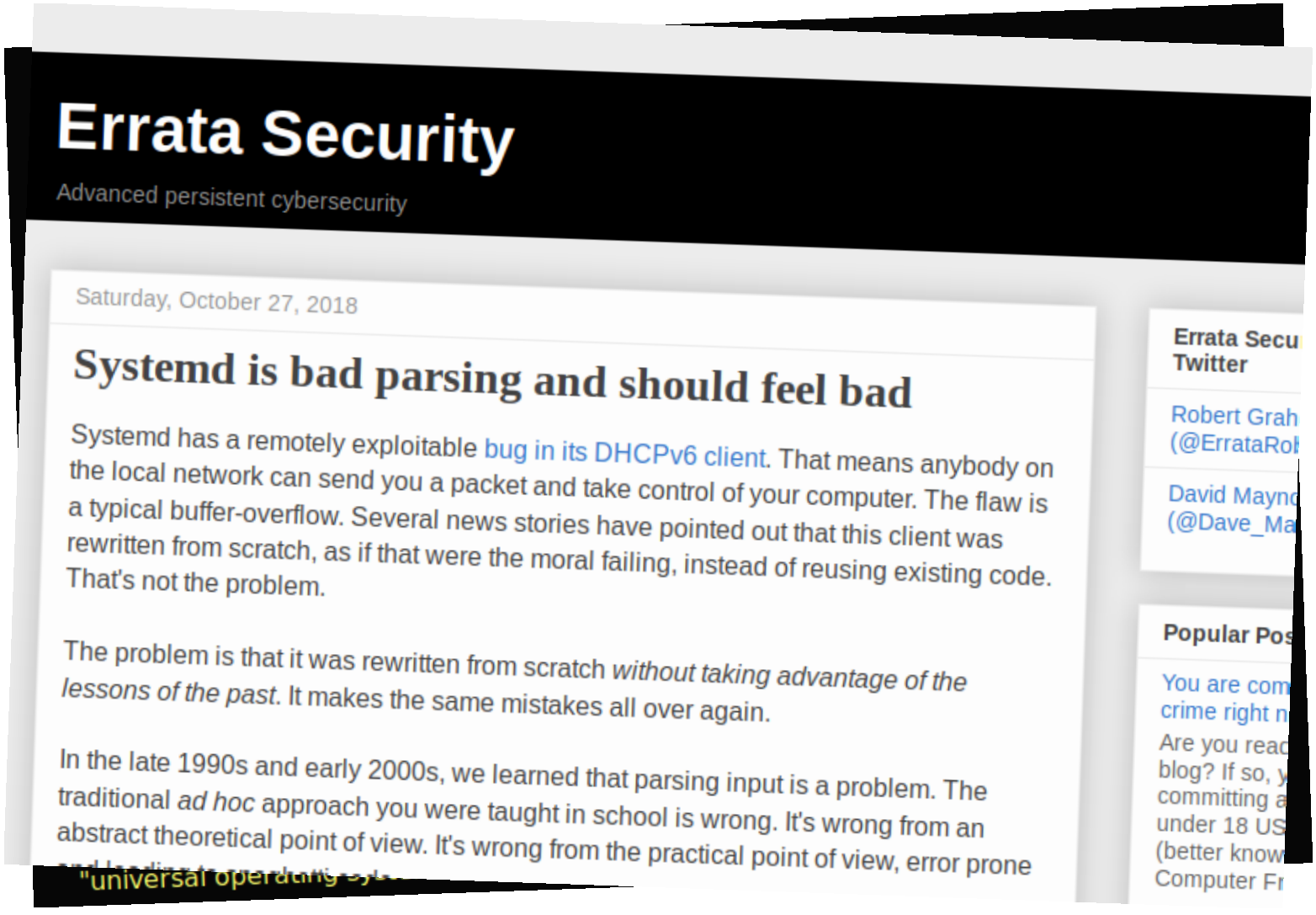
As a Linux sysadmin in the 2010s, it's hard not to have an opinion on systemd. But what I find baffling about it is how divisive it is; nearly everyone (or at least the most vocal crowd) seems to either love it or hate it. When I tell people that systemd was the catalyst for my defection to OpenBSD last year, their usual reaction is to assume that I am part of the "hate it" group. Nope.

In truth, systemd itself was a very small part of the reason I jumped ship. Its introduction made me realise two important things. First, the design problems with modern Linux run deeper than any one piece of software, I just hadn't noticed until I had a fresh one to learn. Second, and this is specific to Debian, the "universal operating system" mantra is fundamentally flawed;

Home
Articles
Projects
Tutorials

About me
Copyright
Ugly

[20],



Errata Security

Advanced persistent cybersecurity

Saturday, October 27, 2018

Systemd is bad parsing and should feel bad

Systemd has a remotely exploitable bug in its DHCPv6 client. That means anybody on the local network can send you a packet and take control of your computer. The flaw is a typical buffer-overflow. Several news stories have pointed out that this client was rewritten from scratch, as if that were the moral failing, instead of reusing existing code. That's not the problem.

The problem is that it was rewritten from scratch *without taking advantage of the lessons of the past*. It makes the same mistakes all over again.

In the late 1990s and early 2000s, we learned that parsing input is a problem. The traditional *ad hoc* approach you were taught in school is wrong. It's wrong from an abstract theoretical point of view. It's wrong from the practical point of view, error prone

Errata Secu
Twitter

Robert Grah
(@ErrataRob

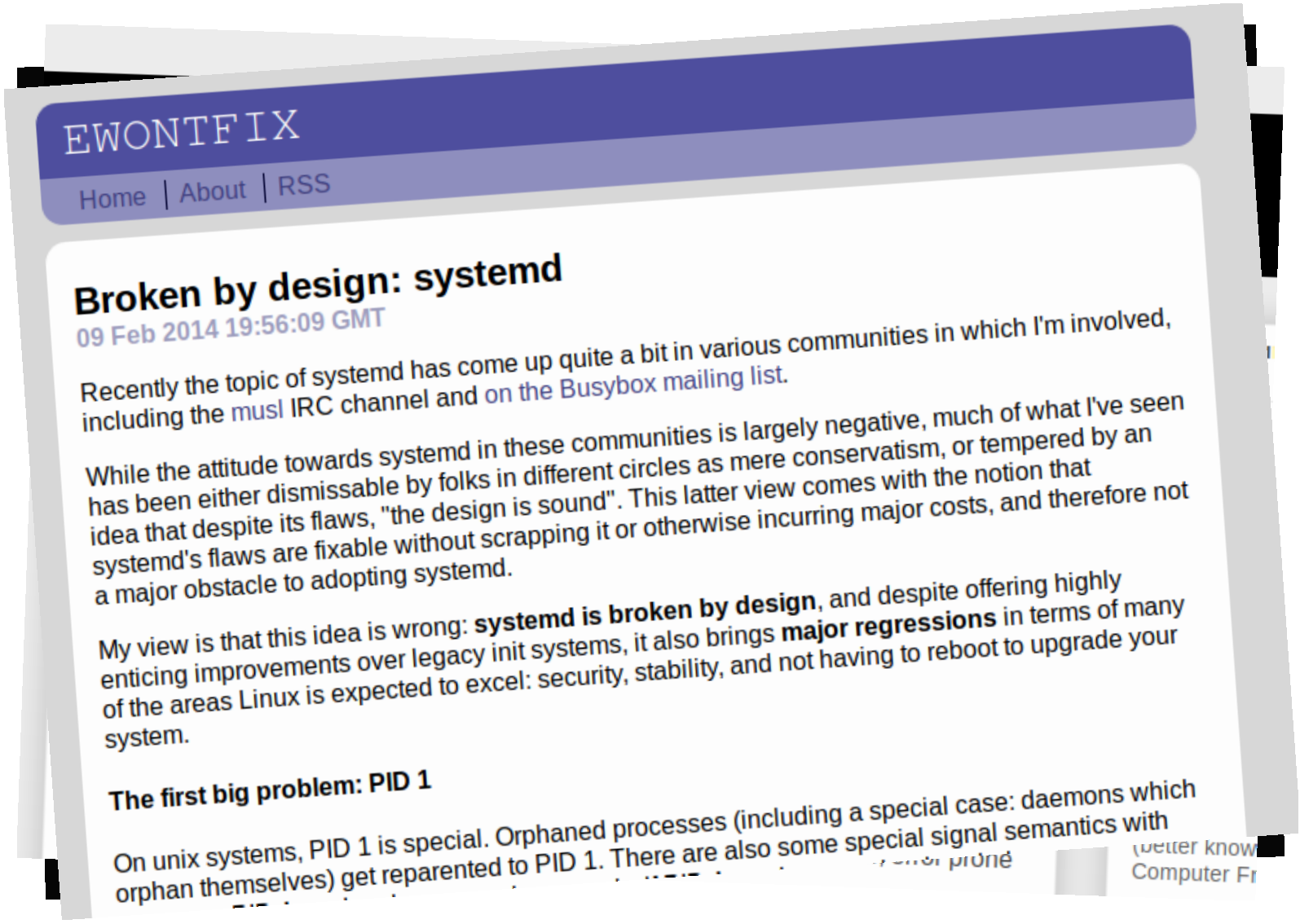
David Mayn
(@Dave_Ma

Popular Pos

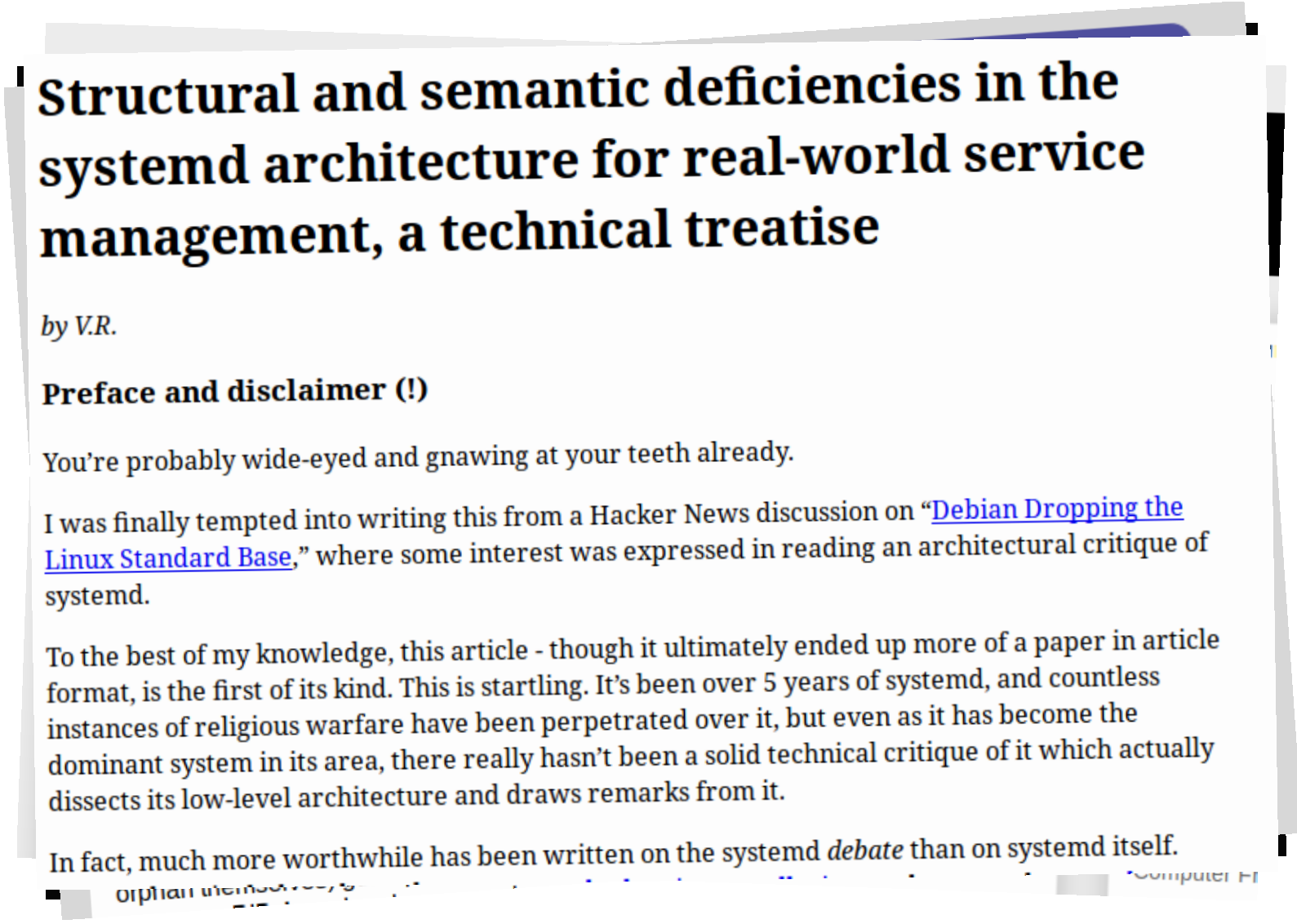
You are com
crime right n

Are you reac
blog? If so, y
committing a
under 18 US
(better know
Computer Fr

[20], [14],


EWONTFIX
Home | About | RSS
Broken by design: systemd
09 Feb 2014 19:56:09 GMT
Recently the topic of systemd has come up quite a bit in various communities in which I'm involved, including the musl IRC channel and on the Busybox mailing list.
While the attitude towards systemd in these communities is largely negative, much of what I've seen has been either dismissable by folks in different circles as mere conservatism, or tempered by an idea that despite its flaws, "the design is sound". This latter view comes with the notion that systemd's flaws are fixable without scrapping it or otherwise incurring major costs, and therefore not a major obstacle to adopting systemd.
My view is that this idea is wrong: systemd is broken by design, and despite offering highly enticing improvements over legacy init systems, it also brings major regressions in terms of many of the areas Linux is expected to excel: security, stability, and not having to reboot to upgrade your system.
The first big problem: PID 1
On unix systems, PID 1 is special. Orphaned processes (including a special case: daemons which orphan themselves) get reparented to PID 1. There are also some special signal semantics with

[20], [14], [3],



Structural and semantic deficiencies in the systemd architecture for real-world service management, a technical treatise

by V.R.

Preface and disclaimer (!)

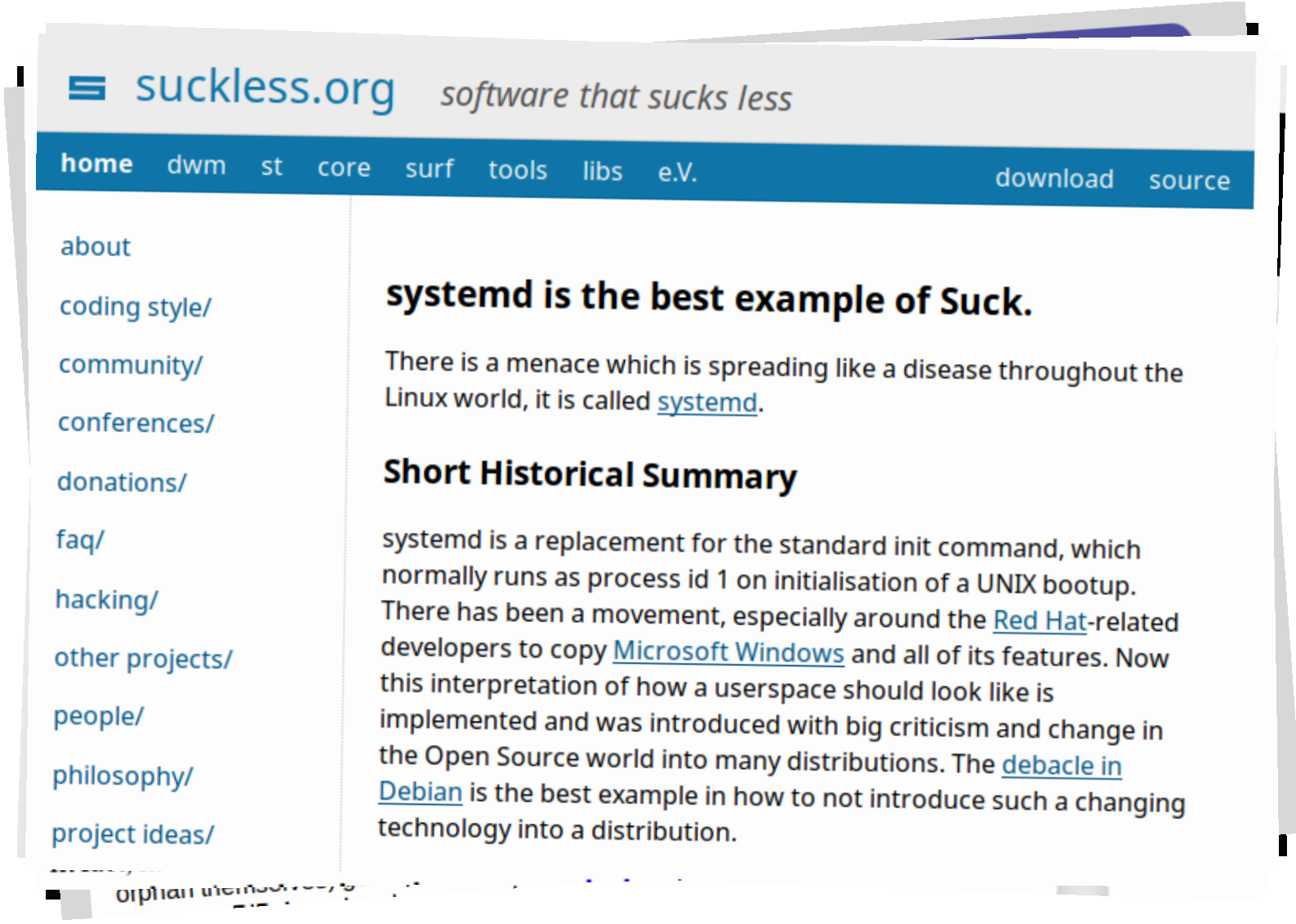
You're probably wide-eyed and gnawing at your teeth already.

I was finally tempted into writing this from a Hacker News discussion on "Debian Dropping the Linux Standard Base," where some interest was expressed in reading an architectural critique of systemd.

To the best of my knowledge, this article - though it ultimately ended up more of a paper in article format, is the first of its kind. This is startling. It's been over 5 years of systemd, and countless instances of religious warfare have been perpetrated over it, but even as it has become the dominant system in its area, there really hasn't been a solid technical critique of it which actually dissects its low-level architecture and draws remarks from it.

In fact, much more worthwhile has been written on the systemd *debate* than on systemd itself.

[20], [14], [3], [23],


suckless.org
software that sucks less
home
dwm
st
core
surf
tools
libs
e.V.
download
source
about
coding style/
community/
conferences/
donations/
faq/
hacking/
other projects/
people/
philosophy/
project ideas/
systemd is the best example of Suck.
There is a menace which is spreading like a disease throughout the Linux world, it is called systemd.
Short Historical Summary
systemd is a replacement for the standard init command, which normally runs as process id 1 on initialisation of a UNIX bootup. There has been a movement, especially around the Red Hat-related developers to copy Microsoft Windows and all of its features. Now this interpretation of how a userspace should look like is implemented and was introduced with big criticism and change in the Open Source world into many distributions. The debacle in Debian is the best example in how to not introduce such a changing technology into a distribution.

[20], [14], [3], [23], [12],

NOV 20TH, 2014 | COMMENTS

Systemd-redux

I figured it was about time for a followup on my systemd post. I've been meaning to do it for a while but time hasn't allowed.

The end of Linux

Some people wrongly characterized this as some sort of hyperbole. It was not. Systemd *IS* changing what we know as Linux today. It remains to be seen if this is a good or bad thing but Linux is becoming something different than it was.

Linux is in for a rough few years

I do honestly believe this will end up being the start of a rocky period for Linux.

orphan

About my Blogger blog

I'm currently in the process of migrating content from my blogger blog into Octopress. As such, this all seems pretty threadbare. If you're curious, you can get to it here

Recent Posts

So you wanna go on-prem do ya

The Flaw In All Things

Review: Site Reliability Engineering

how is rundeck formed

aws api gateway for fun and profit

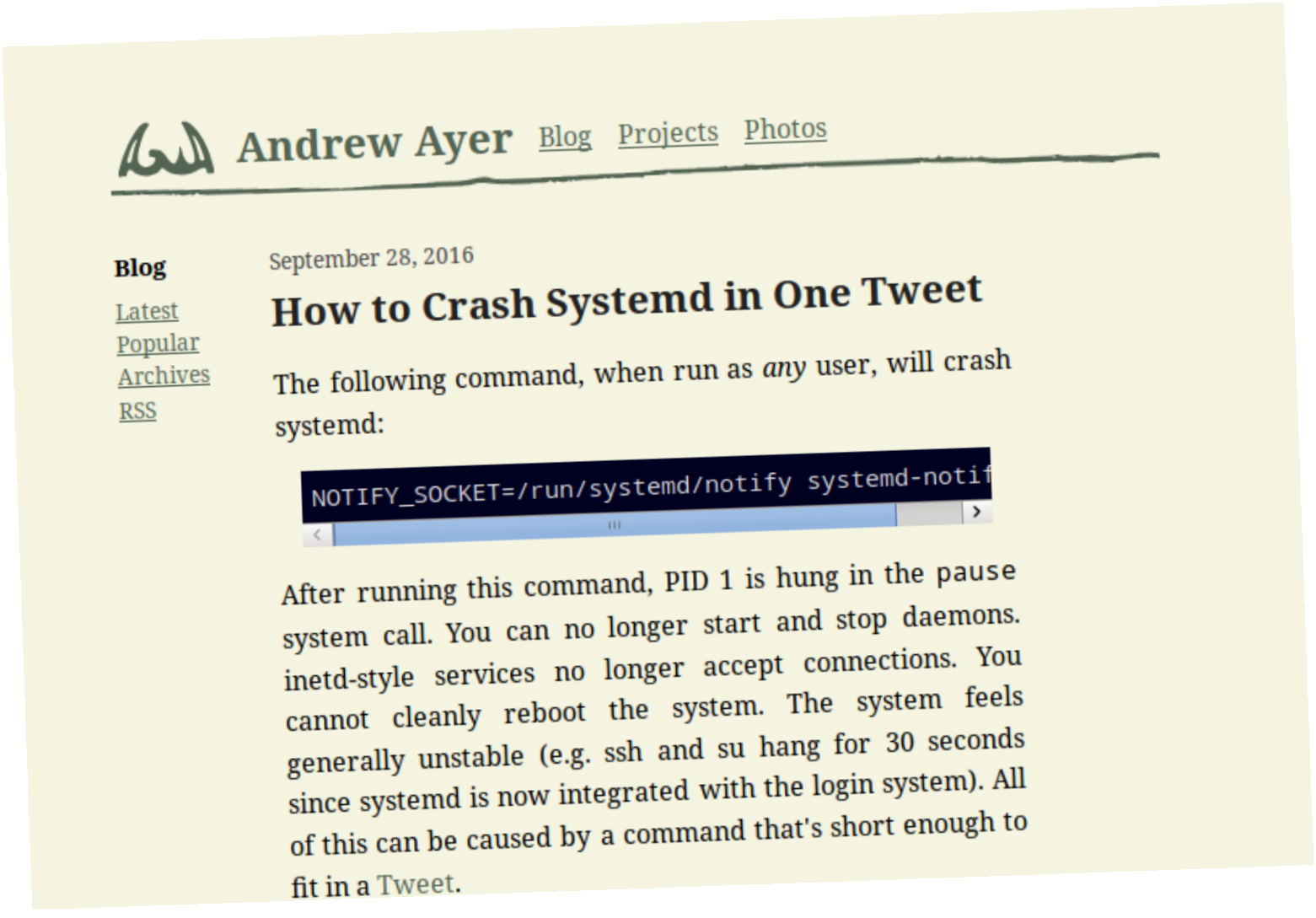
Github Repos

Status updating...

@lusis on Github

Latest Tweets

[20], [14], [3], [23], [12], [31]



Andrew Ayer Blog Projects Photos

Blog

Latest
Popular
Archives
RSS

September 28, 2016

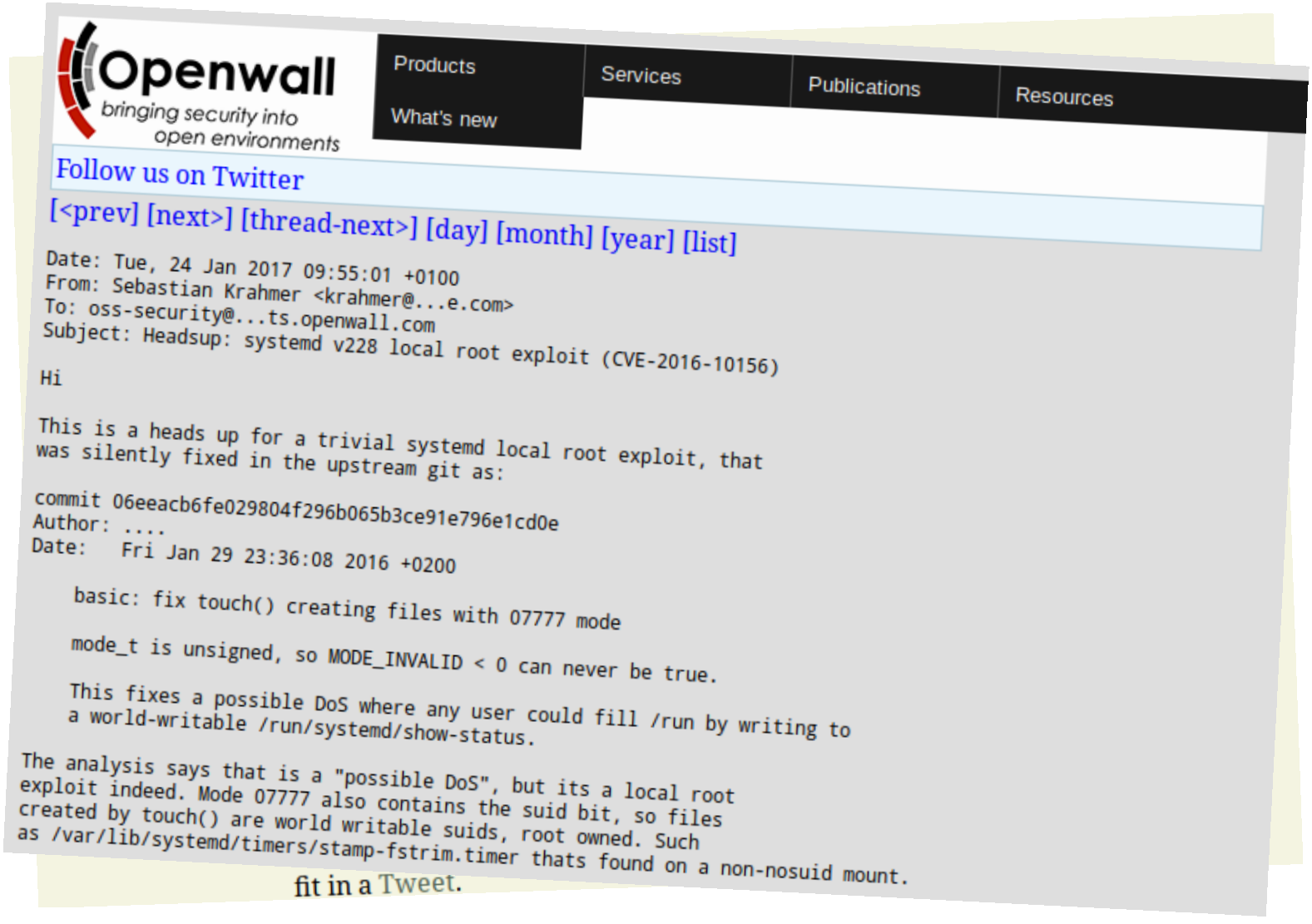
How to Crash Systemd in One Tweet

The following command, when run as *any* user, will crash systemd:

```
NOTIFY_SOCKET=/run/systemd/notify systemd-notif
```

After running this command, PID 1 is hung in the `pause` system call. You can no longer start and stop daemons. inetd-style services no longer accept connections. You cannot cleanly reboot the system. The system feels generally unstable (e.g. ssh and su hang for 30 seconds since systemd is now integrated with the login system). All of this can be caused by a command that's short enough to fit in a Tweet.

[6],



Openwall
bringing security into open environments

Products | Services | Publications | Resources
What's new

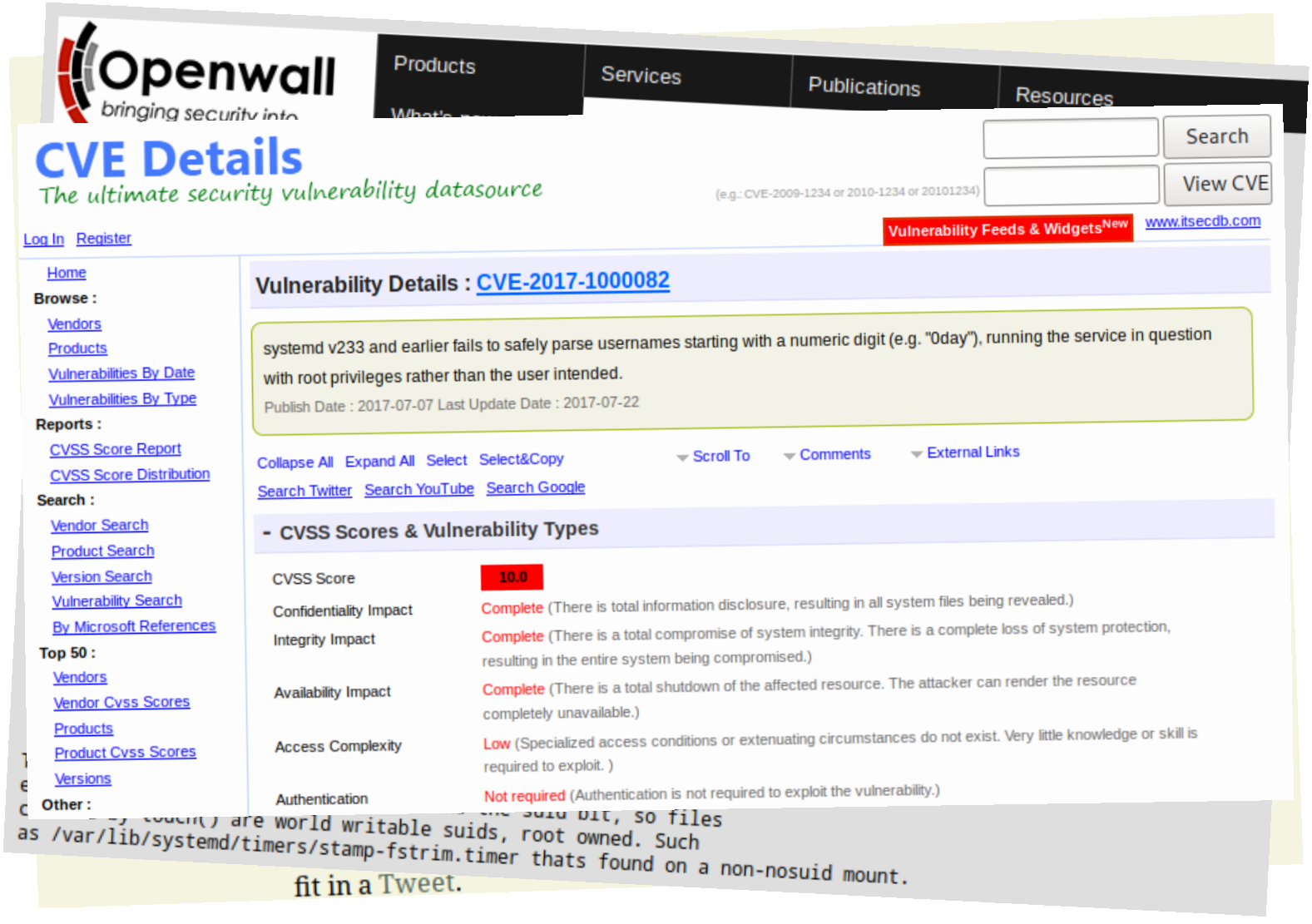
Follow us on Twitter

[<prev] [next>] [thread-next>] [day] [month] [year] [list]

```
Date: Tue, 24 Jan 2017 09:55:01 +0100
From: Sebastian Krahmer <krahmer@...e.com>
To: oss-security@...ts.openwall.com
Subject: Headsup: systemd v228 local root exploit (CVE-2016-10156)

Hi

This is a heads up for a trivial systemd local root exploit, that
was silently fixed in the upstream git as:

commit 06eeacb6fe029804f296b065b3ce91e796e1cd0e
Author: ....
Date:   Fri Jan 29 23:36:08 2016 +0200

    basic: fix touch() creating files with 07777 mode

    mode_t is unsigned, so MODE_INVALID < 0 can never be true.

    This fixes a possible DoS where any user could fill /run by writing to
    a world-writable /run/systemd/show-status.

The analysis says that is a "possible DoS", but its a local root
exploit indeed. Mode 07777 also contains the suid bit, so files
created by touch() are world writable suids, root owned. Such
as /var/lib/systemd/timers/stamp-fstrim.timer thats found on a non-nosuid mount.
```

fit in a Tweet.

[6], [19],


Openwall
bringing security into
Products
Services
Publications
Resources
CVE Details
The ultimate security vulnerability datasource
(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)
Search
View CVE
Log In
Register
Vulnerability Feeds & Widgets New
www.itsecdb.com
Home
Browse :
Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type
Reports :
CVSS Score Report
CVSS Score Distribution
Search :
Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References
Top 50 :
Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions
Other :
Vulnerability Details : CVE-2017-1000082
systemd v233 and earlier fails to safely parse usernames starting with a numeric digit (e.g. "0day"), running the service in question with root privileges rather than the user intended.
Publish Date : 2017-07-07 Last Update Date : 2017-07-22
Collapse All
Expand All
Select
Select&Copy
Scroll To
Comments
External Links
Search Twitter
Search YouTube
Search Google
- CVSS Scores & Vulnerability Types
CVSS Score
10.0
Confidentiality Impact
Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact
Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact
Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity
Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication
Not required (Authentication is not required to exploit the vulnerability.)
as /var/lib/systemd/timers/stamp-fstrim.timer thats found on a non-nosuid mount.
fit in a Tweet.

[6], [19], [9]

... und es wird doch genutzt.

Also was ist `systemd`?

Moritz Müller Philip Kaludercic

2019-11-11 — PASST

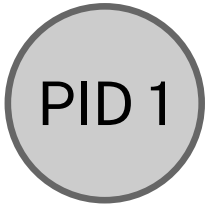
Teil I

Problem

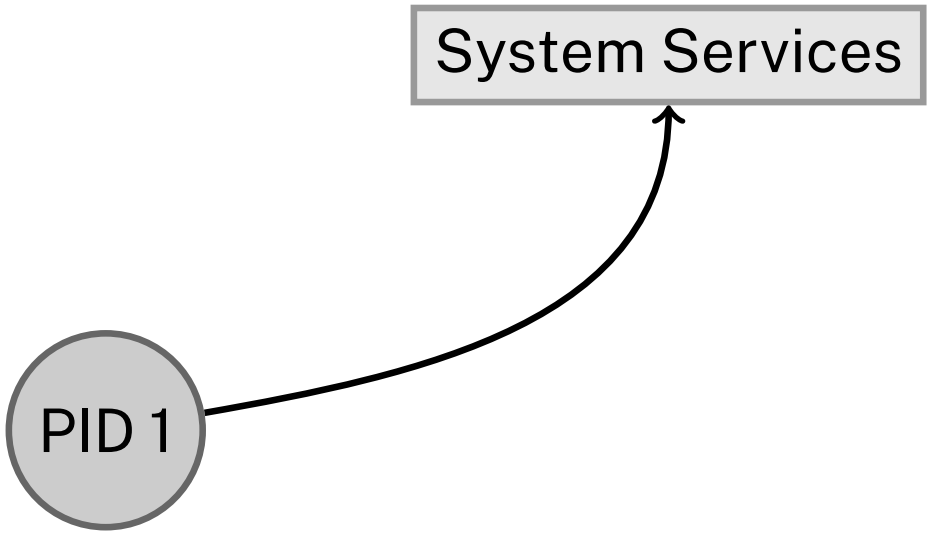
Was ist PID 1?[24][21]

nichts

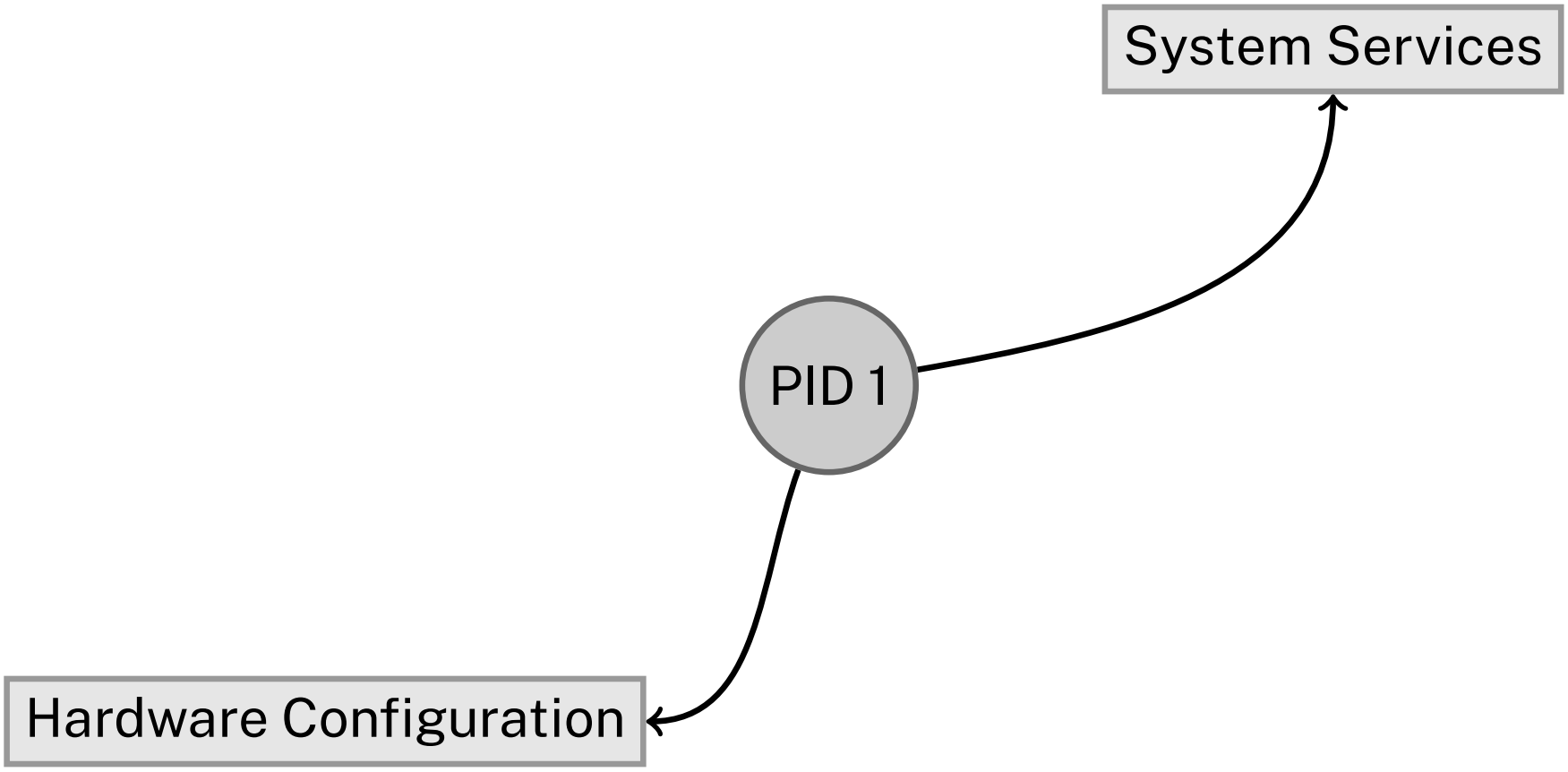
Was ist PID 1?[24][21]



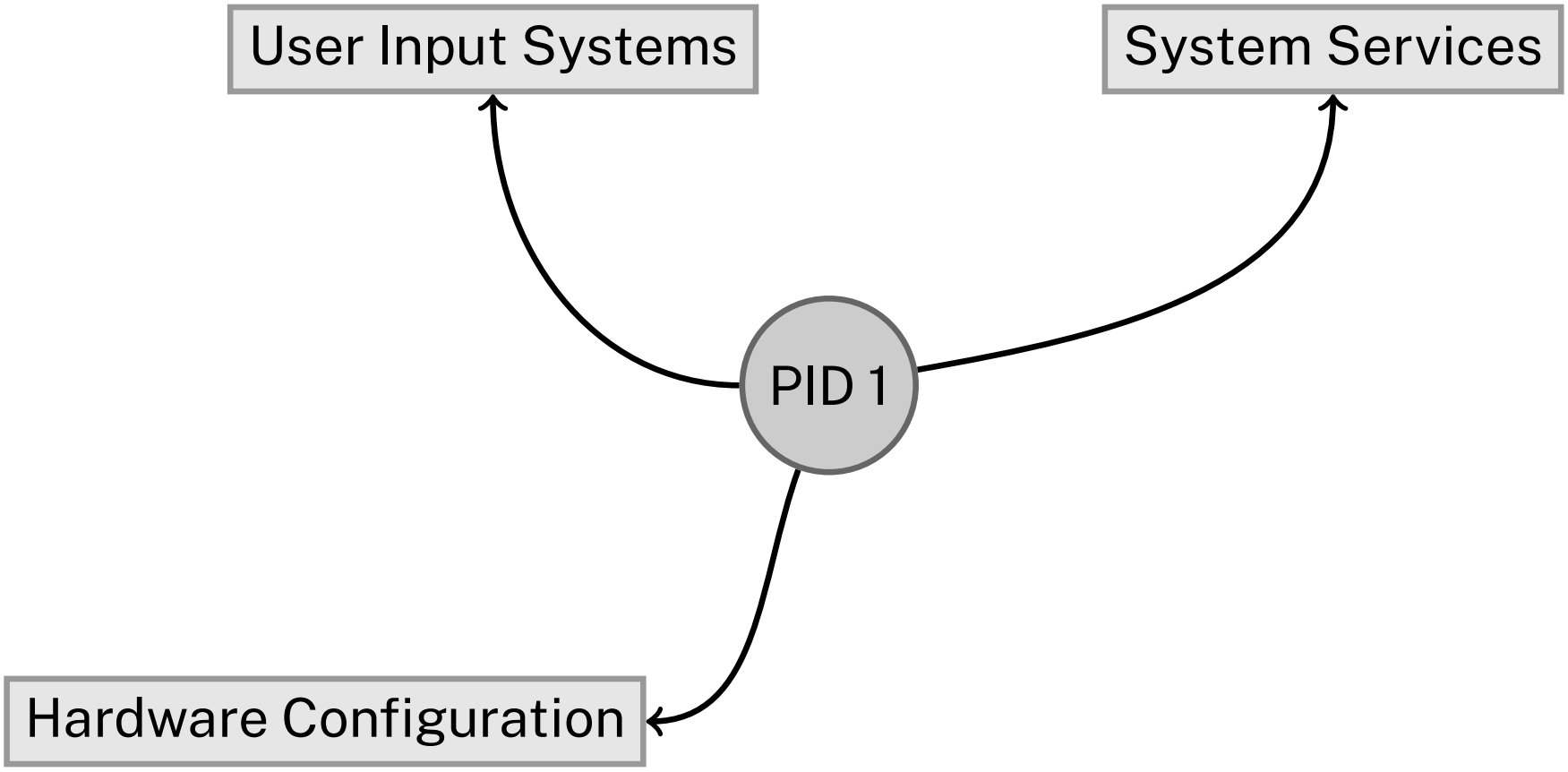
Was ist PID 1?[24][21]



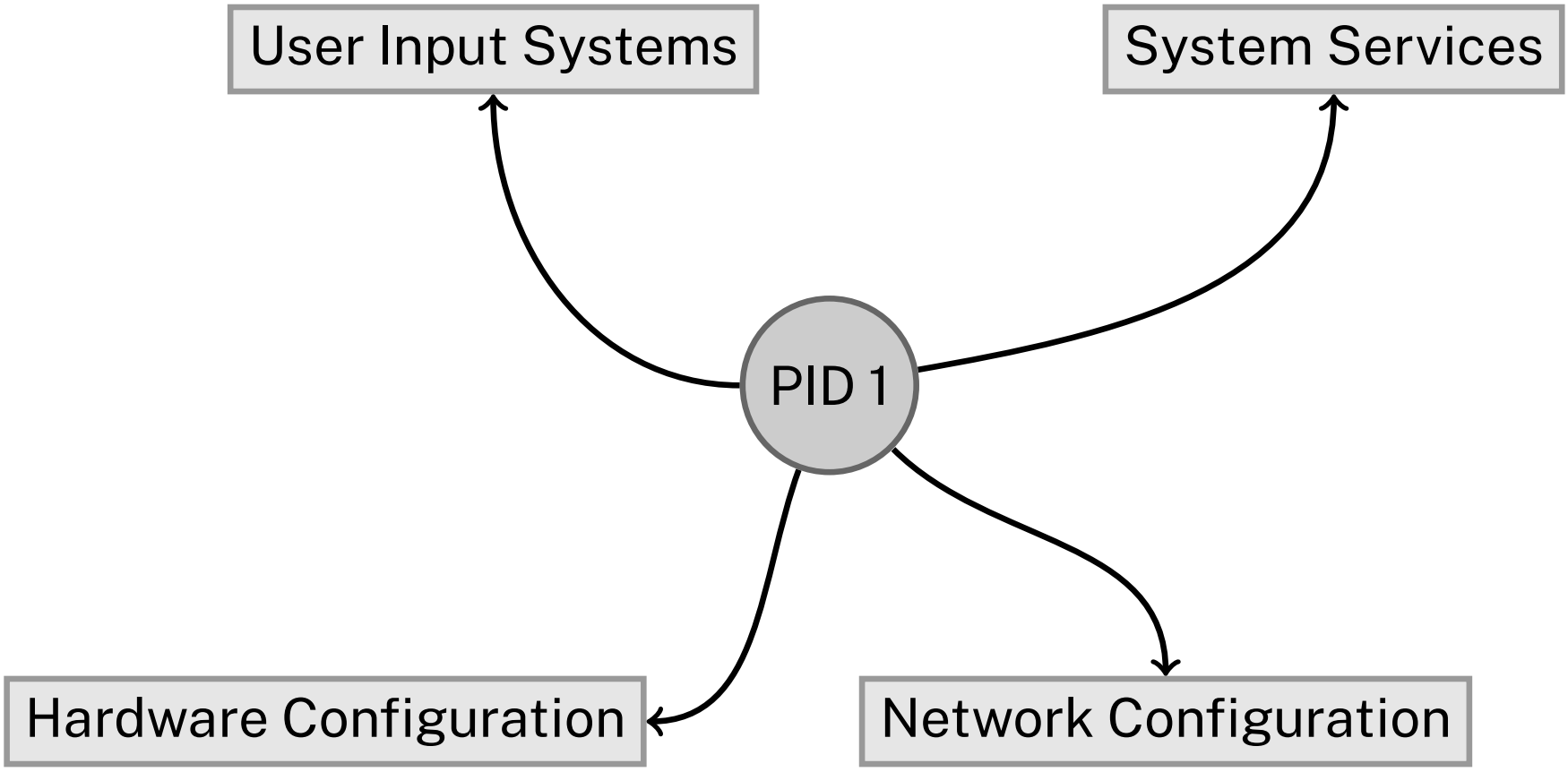
Was ist PID 1?[24][21]



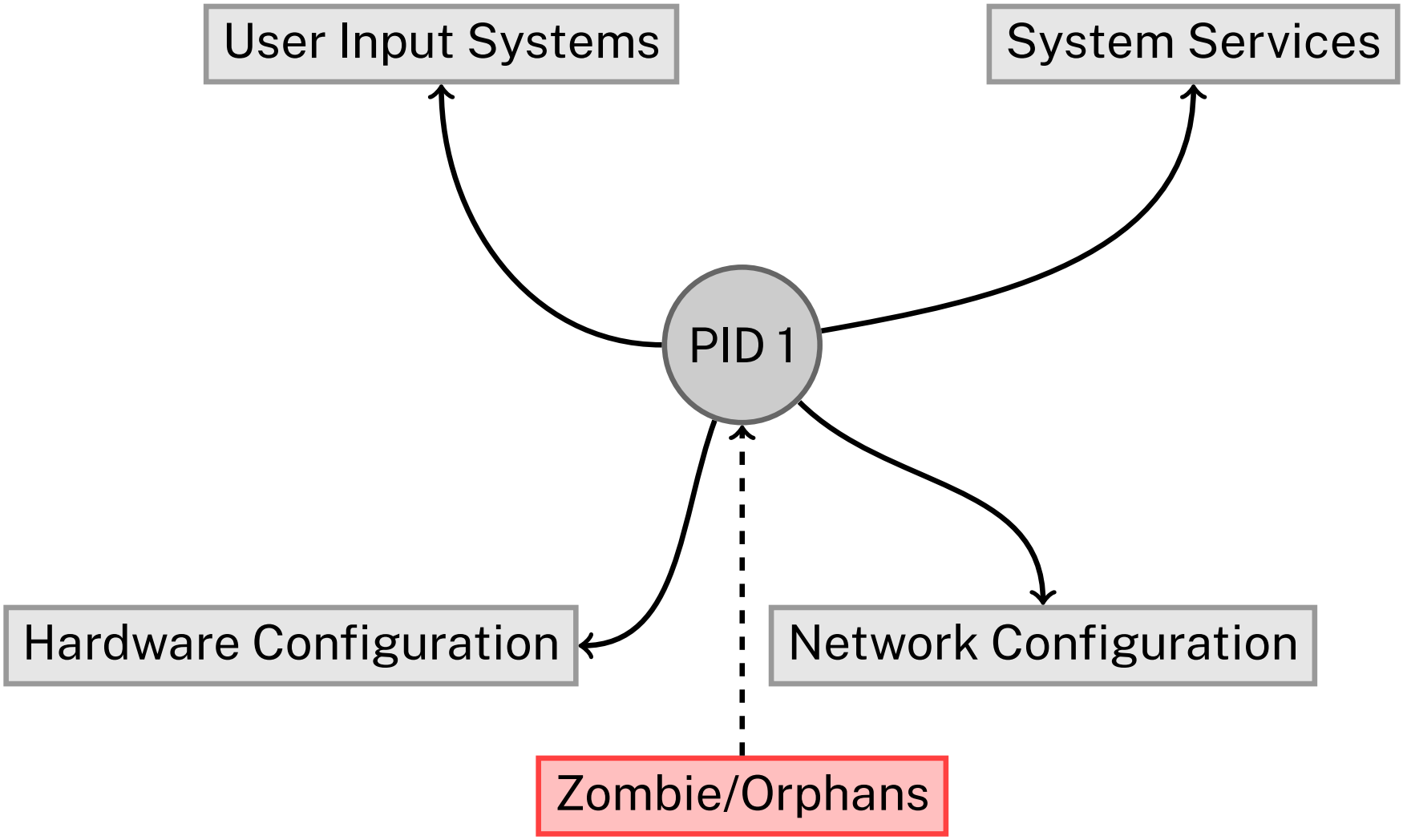
Was ist PID 1?[24][21]



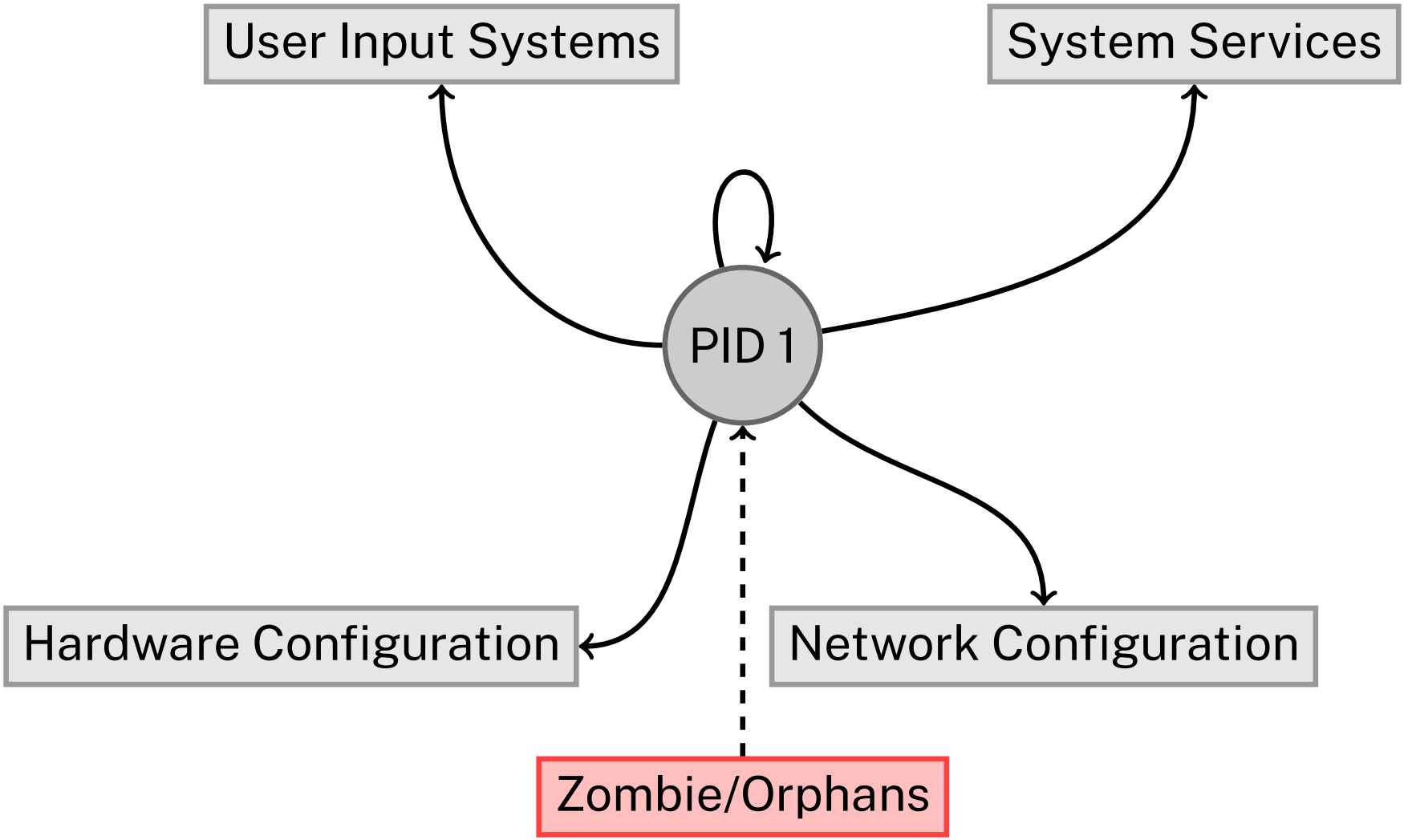
Was ist PID 1?[24][21]



Was ist PID 1?[24][21]



Was ist PID 1?[24][21]



Minimales Program[24]

```
int main() {
    int status;
    const char *shell[] = { "/bin/sh", NULL };
    if (!fork())
        execve(shell[0], shell, NULL);
    while (1)
        wait(&status);
}
```

In “init(1)”

```
  sys intrp
  jms init1   " Fork the first child
              " connected to ttyin/ttyout
  jms init2   " Fork the second child
              " connected to keyboard/display
1:
  sys rmes    " Wait for a child to exit
  sad pid1
    jmp 1f    " It was child 1, so jump to
              " 1f and restart it
  sad pid2
  ...
login:
  -1
  sys intrp
  sys open; password; 0 " Open the passwd file
  lac d1
  sys write; m1; m1s    " Write "\nlogin:" on
                        " the terminal
  jms rline             " and read the user's
                        " username
  ...
```

- PDP-7 Unix hatte nur einen Assembler[1]

In “`init(1)`”

Aus `init(8)` in Unix-7:

> *“When init comes up multiuser, it invokes a shell, with* ***input taken from the file /etc/rc****. This command file performs housekeeping like removing temporary files, mounting file systems, and starting daemons.”*

- PDP-7 Unix hatte nur einen Assembler[1]
- Wird später auf ein Shellscript ausgelagert[2]

In “init(1)”

```
#!/bin/bash

start() { ... }
stop() { ... }

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        ;;
  *)
        exit 1
esac
```

- PDP-7 Unix hatte nur einen Assembler[1]
- Wird später auf ein Shellscript ausgelagert[2]
- `/etc/rc` wird mit der Zeit erweitert[21]

“Shell scripts tend to be **slow**, needlessly **hard to read**, very **verbose** and **fragile**.”

— *Lennart Poettering*[22], 2010

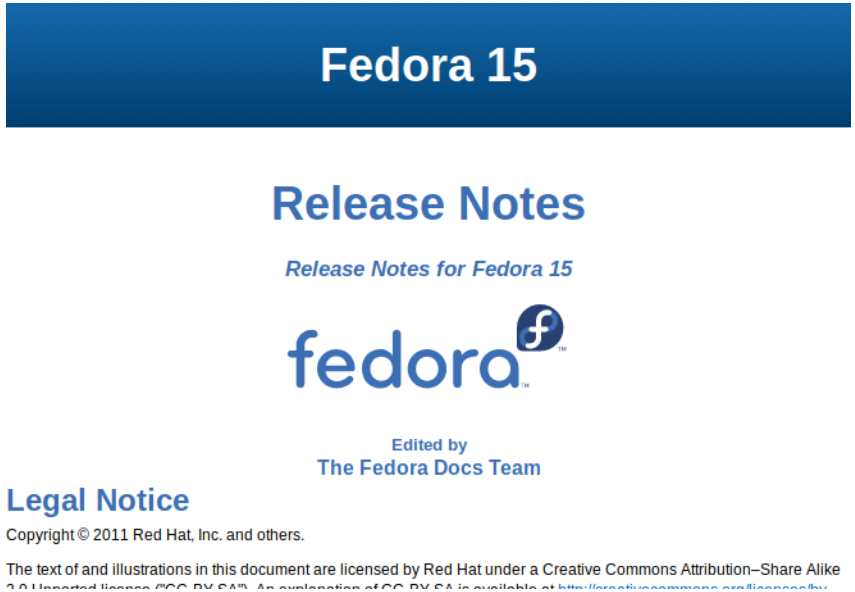
Wer?

2010 Lennart Pottering (Red Hat) will `init(8)` überdenken[21][16]

Wer?

2010 Lennart Pottering (Red Hat) will `init(8)` überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetz[4][13]



Fedora 15

Release Notes

Release Notes for Fedora 15

fedora

Edited by
The Fedora Docs Team

Legal Notice

Copyright © 2011 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike

Wer?

2010 Lennart Pottering (Red Hat) will `init(8)` überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetz[4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf `freedesktop.org`[25]

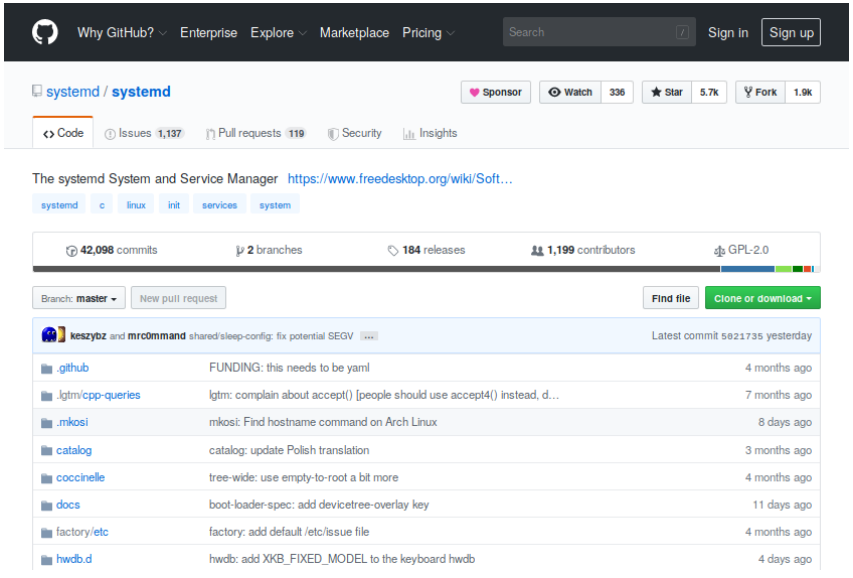
Wer?

2010 Lennart Pottering (Red Hat) will `init(8)` überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetz[4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf `freedesktop.org`[25]

2015- Entwicklung wird auf **GitHub** verlagert[15]



`systemd/systemd`

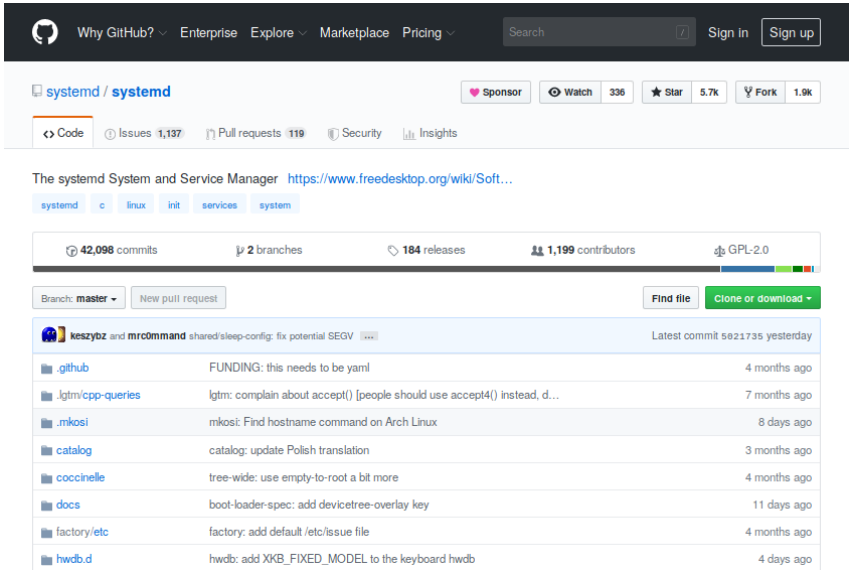
Wer?

2010 Lennart Pottering (Red Hat) will `init(8)` überdenken[21][16]

2011 Obwohl zunächst zögernd[7], wird systemd in Fedora umgesetz[4][13]

2010- Das Projekt wird organisiert mittels Mailingslisten[30], IRC[8], auf `freedesktop.org`[25]

2015- Entwicklung wird auf **GitHub** verlagert[15] … alte Strukturen bleiben aber immernoch aktiv.



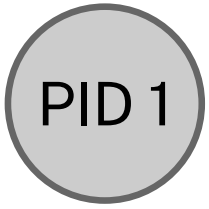
`systemd/systemd`

Teil II

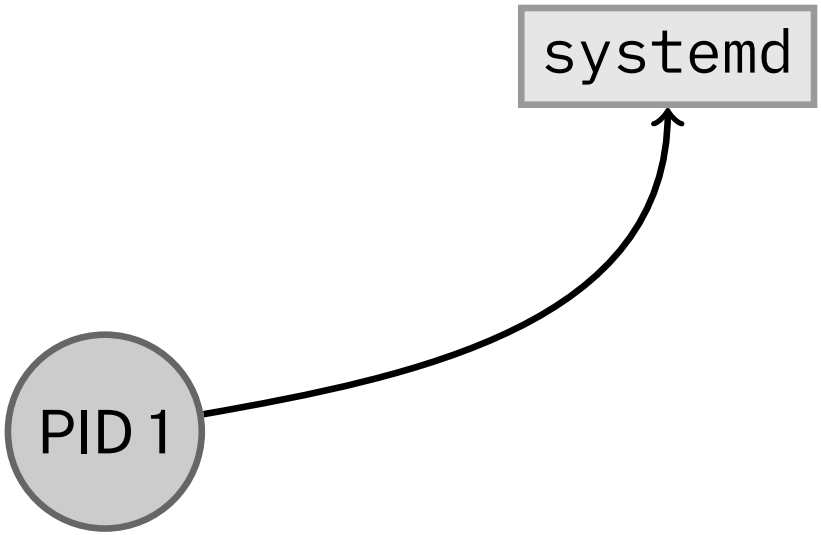
Lösung?

Komponenten von systemd[29][28][27][26]

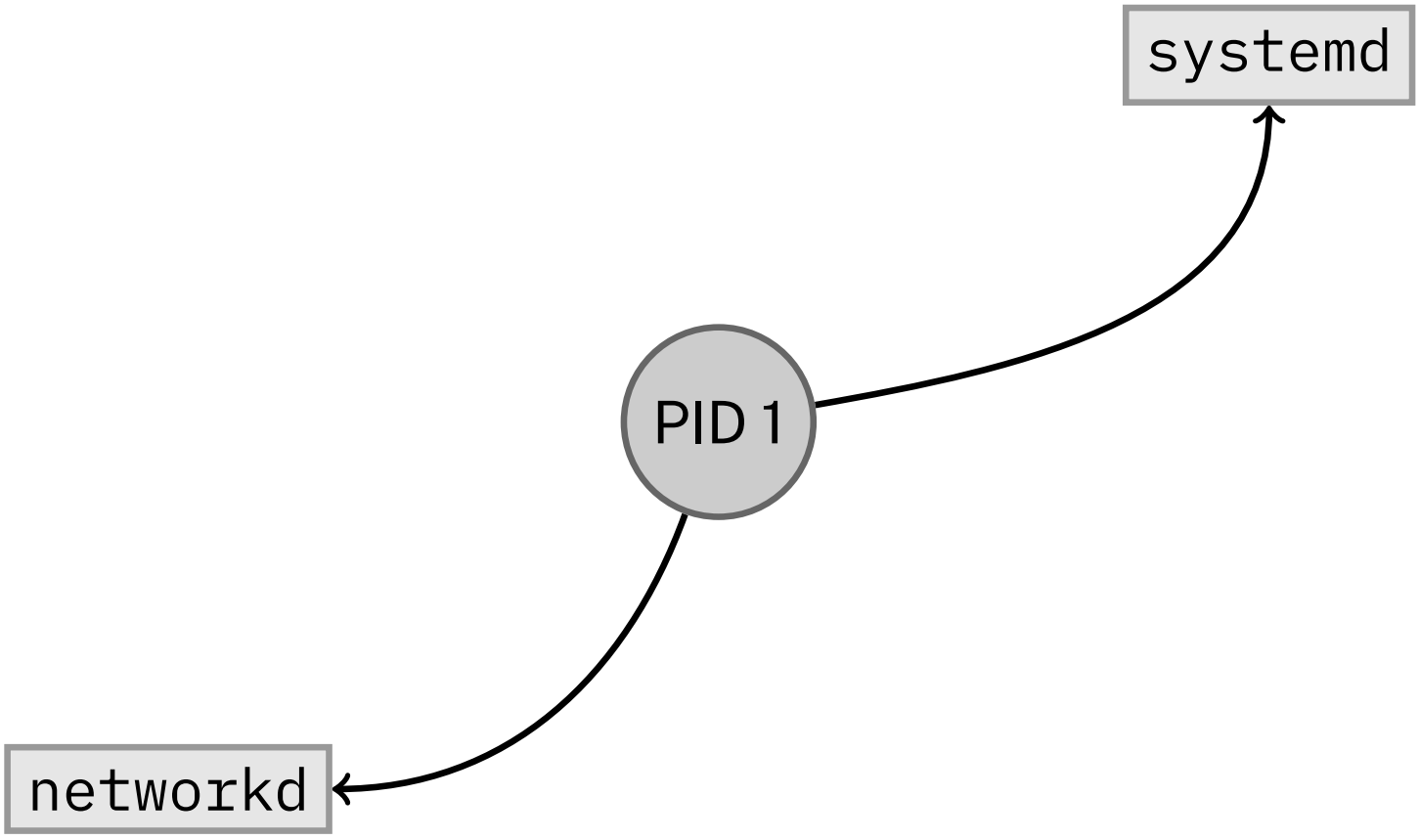
Komponenten von systemd[29][28][27][26]



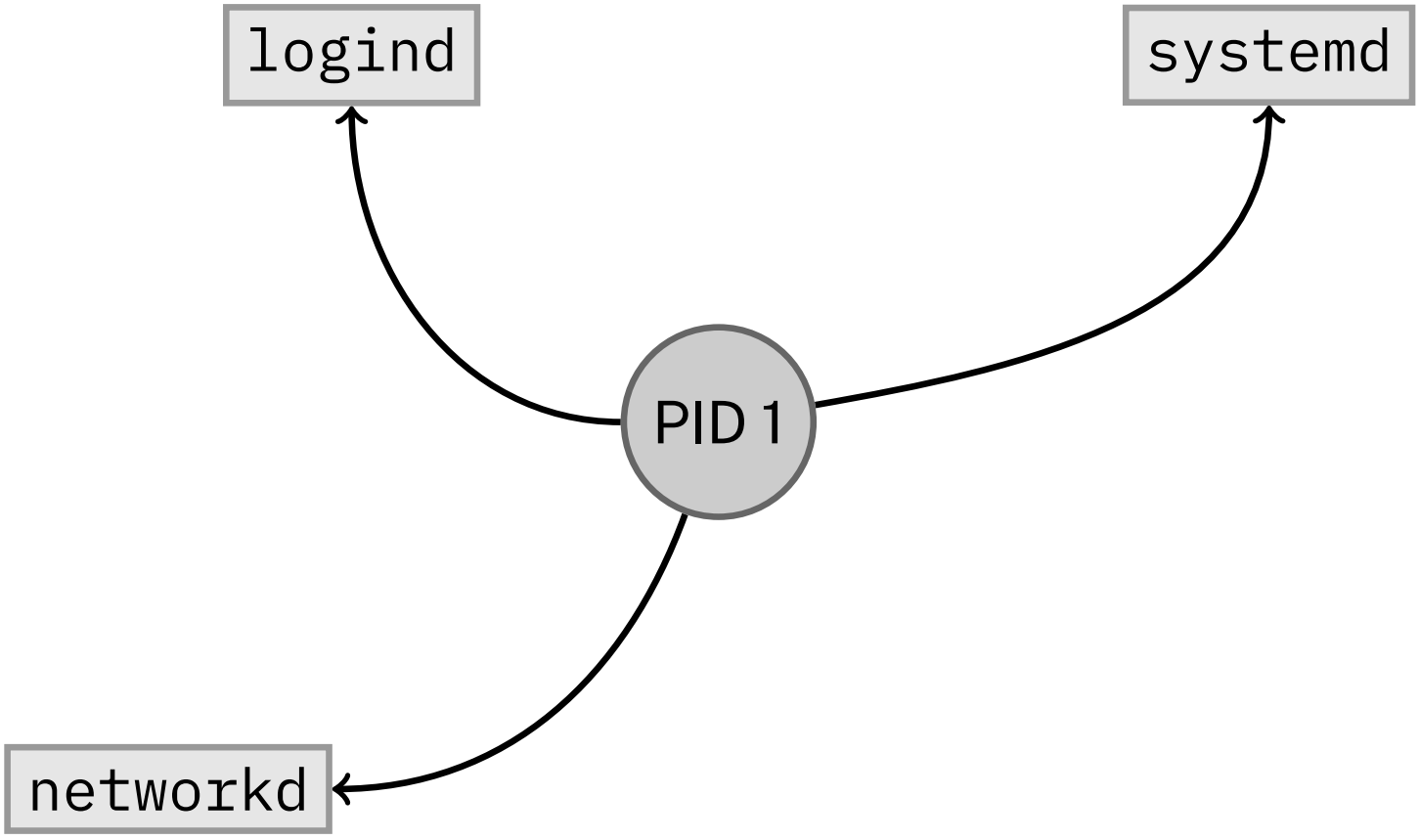
Komponenten von systemd[29][28][27][26]



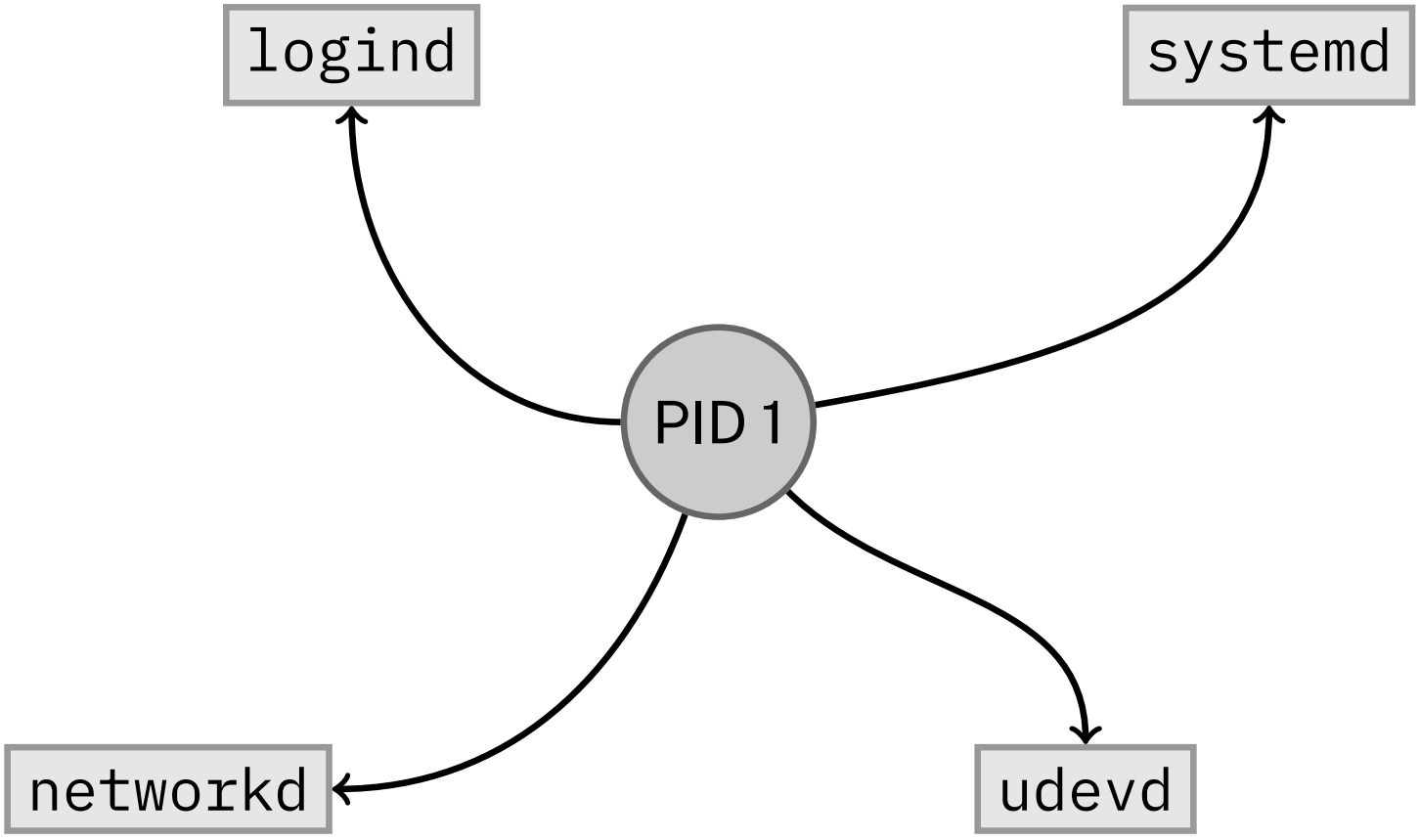
Komponenten von systemd[29][28][27][26]



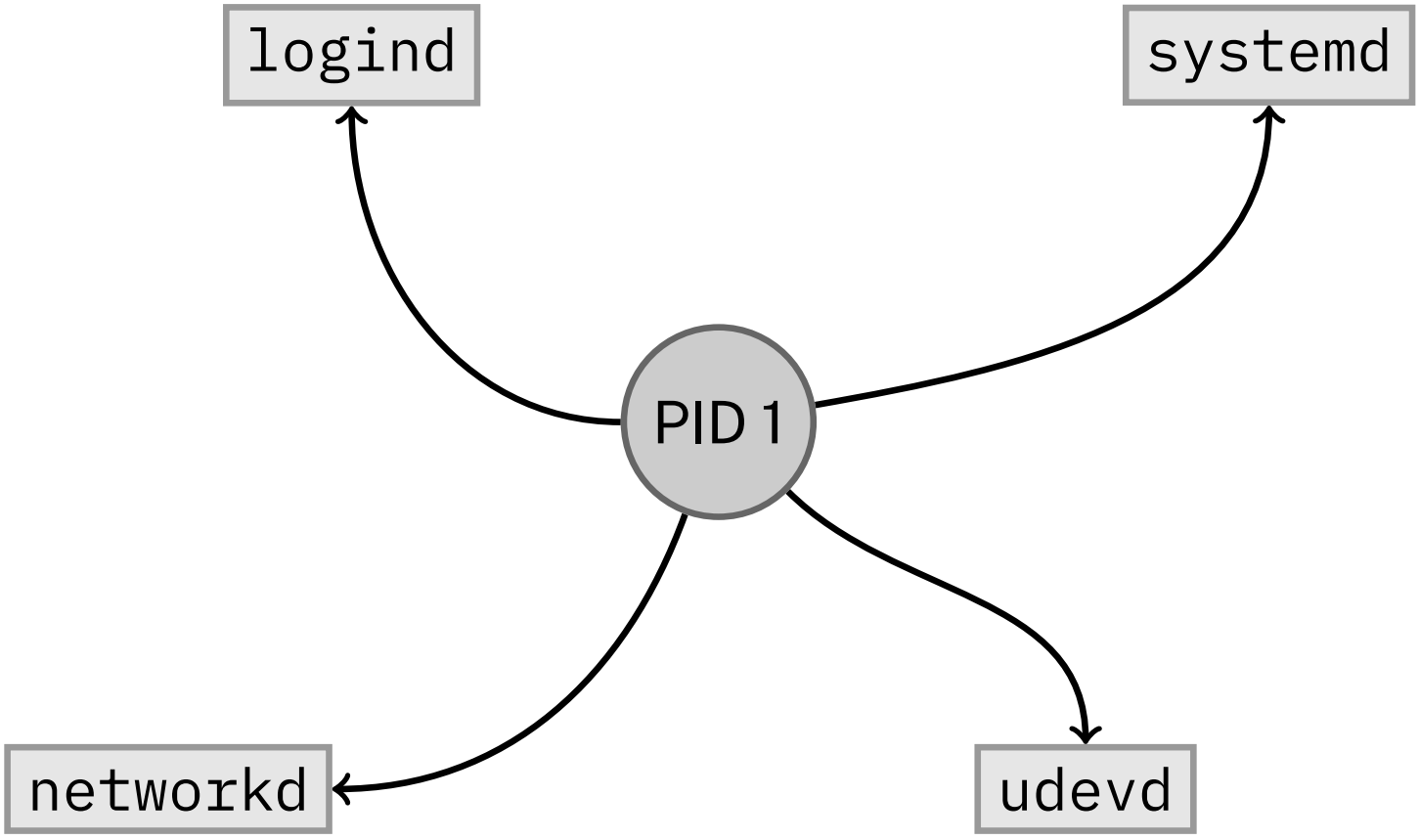
Komponenten von systemd[29][28][27][26]



Komponenten von systemd[29][28][27][26]

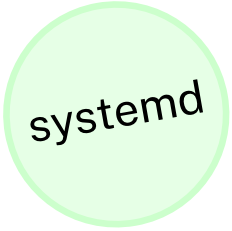


Komponenten von systemd[29][28][27][26]

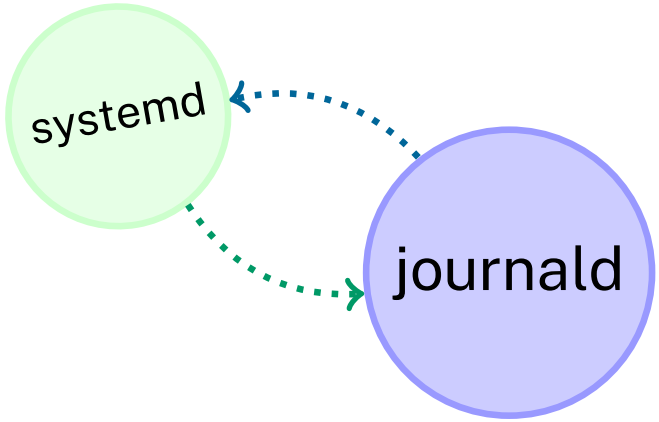


... 69 Binaries

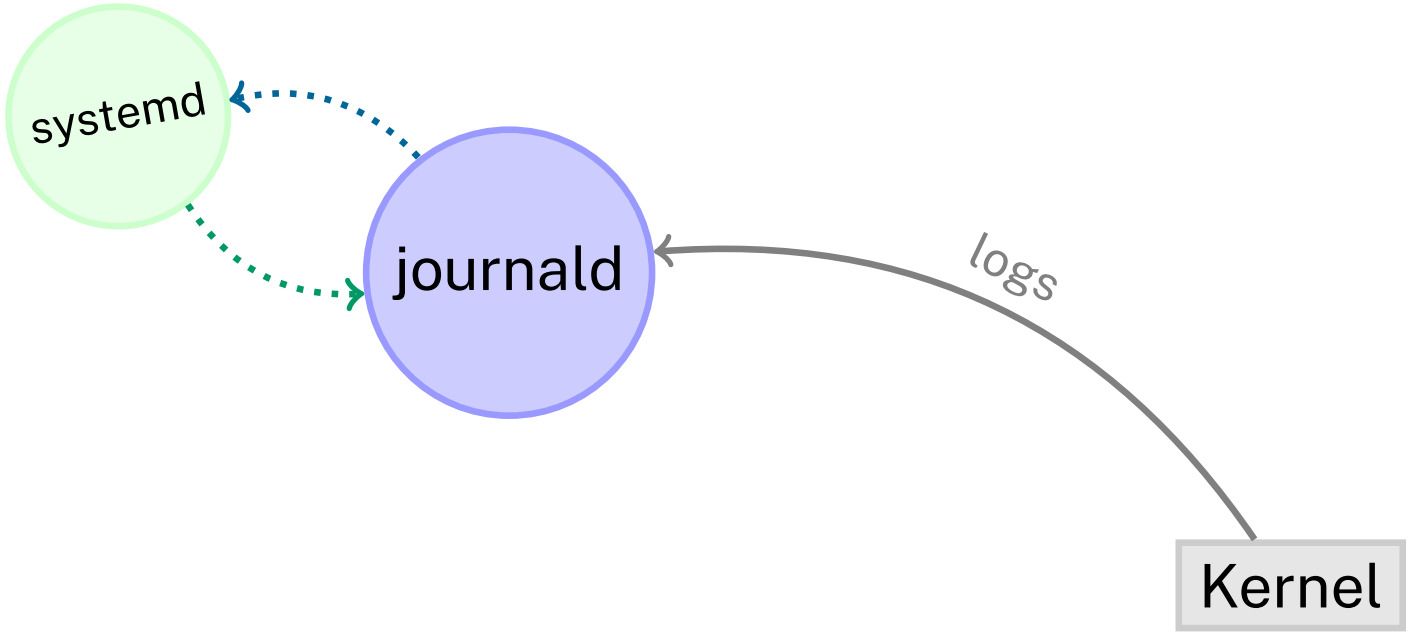
Beispiel: `journald`[26]



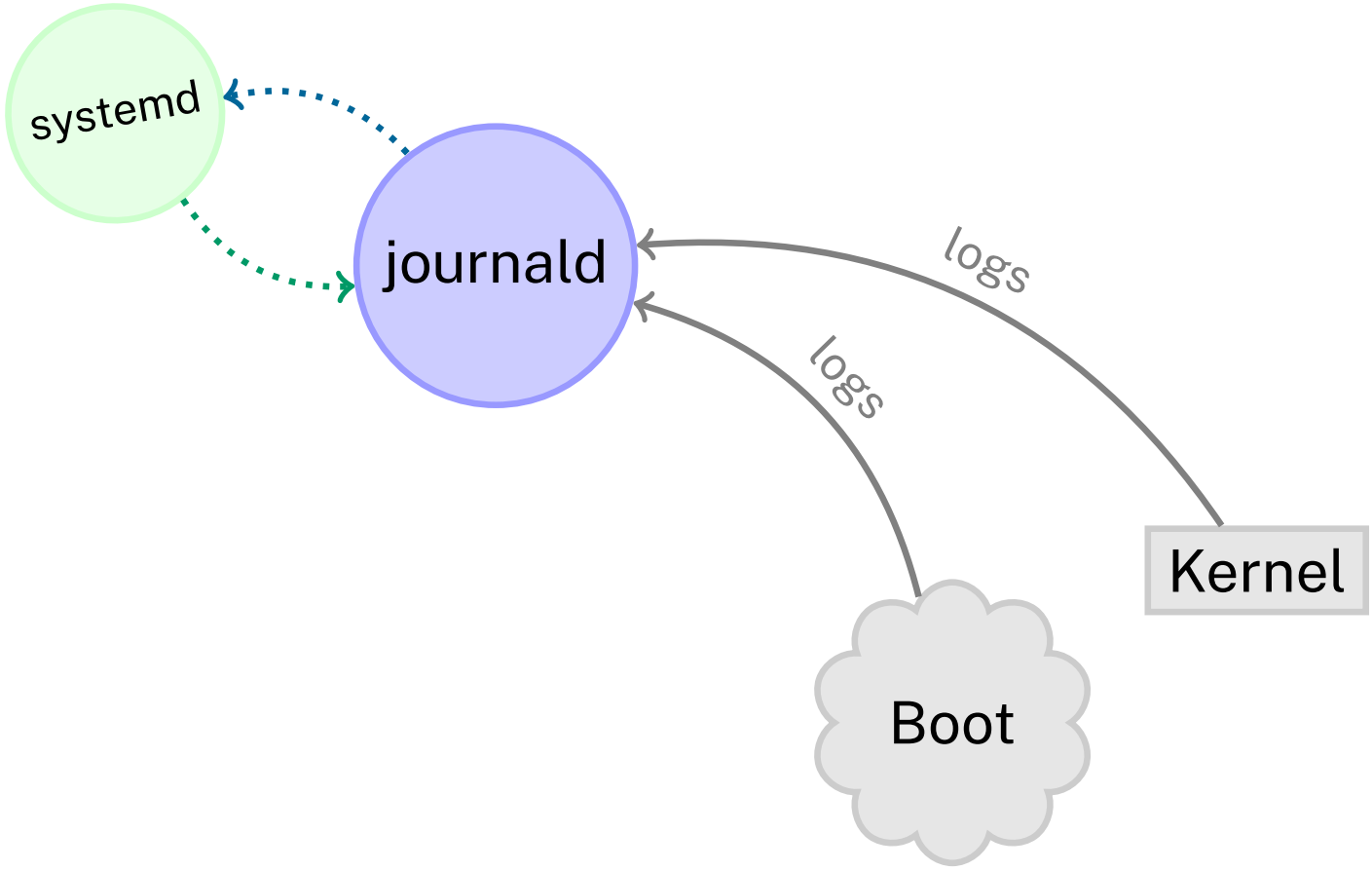
Beispiel: journald[26]



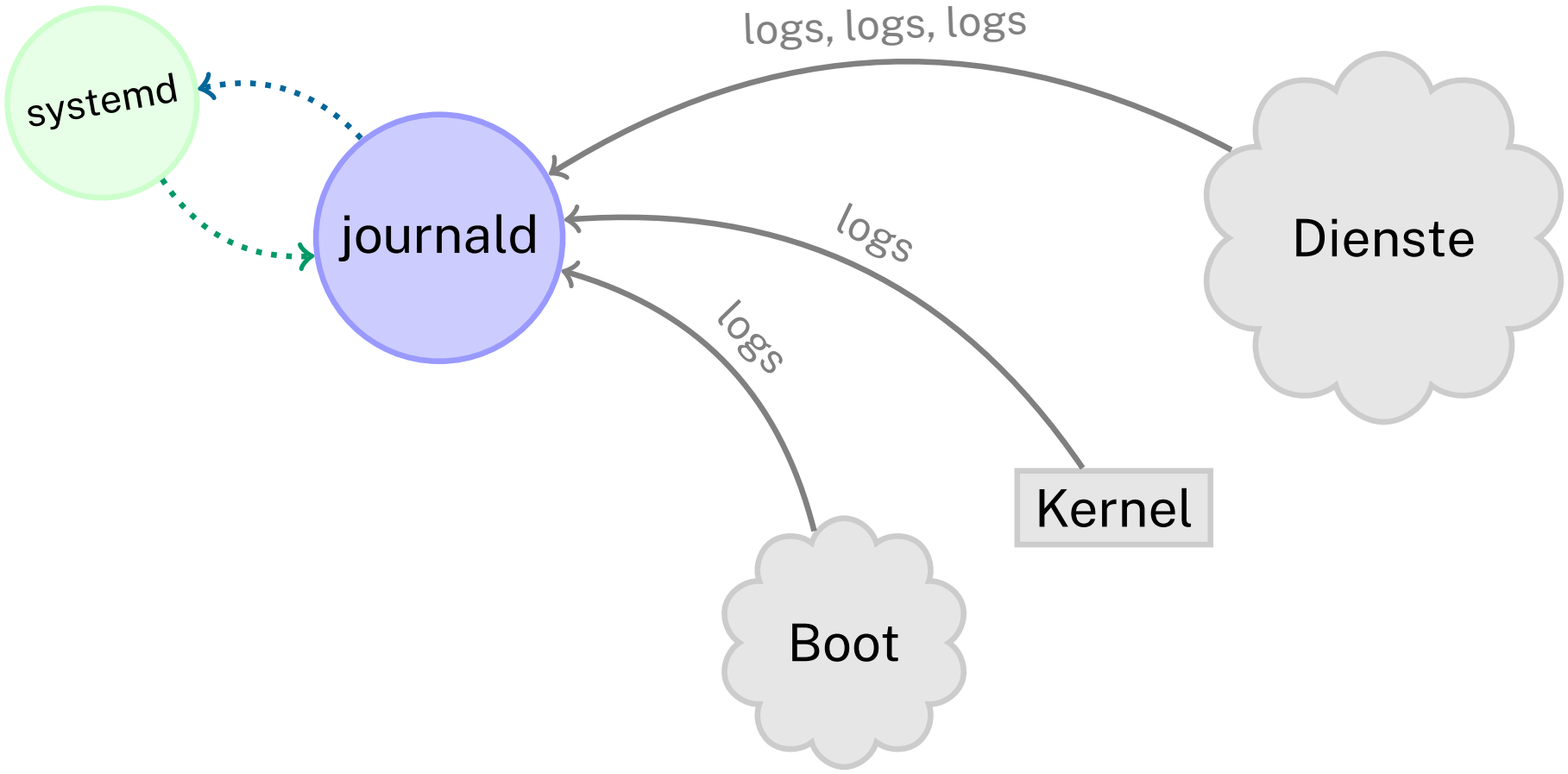
Beispiel: journald[26]



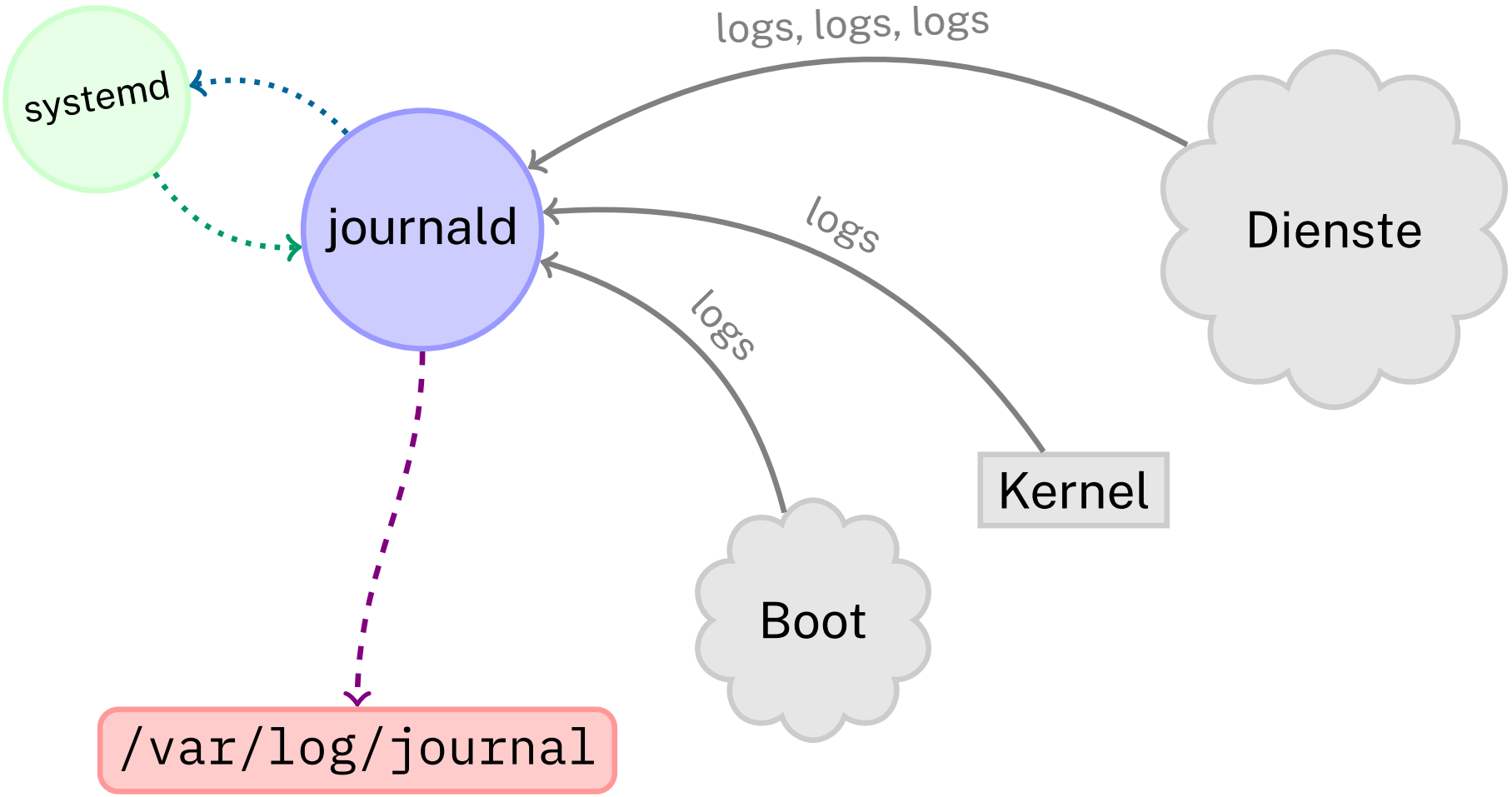
Beispiel: `journald`[26]



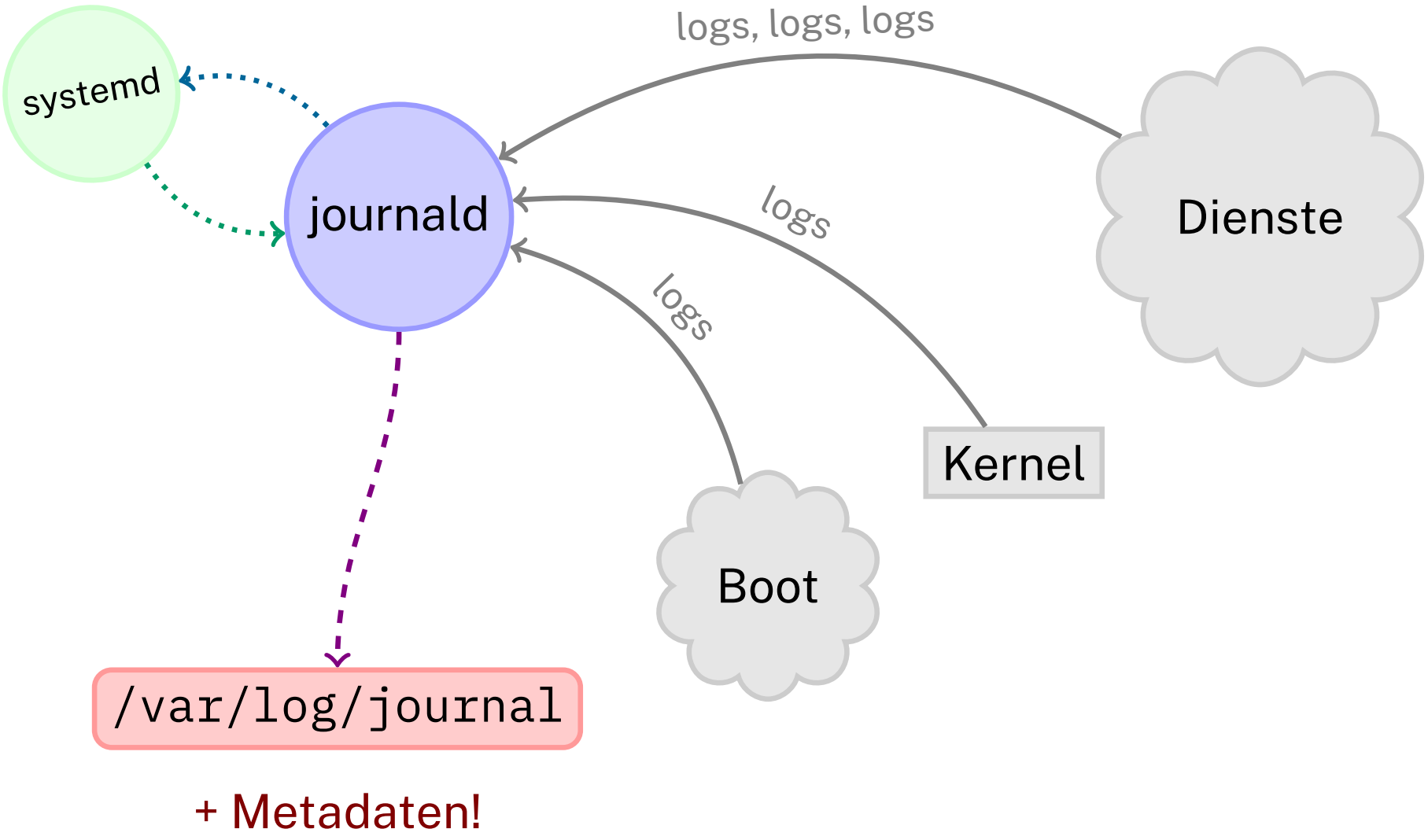
Beispiel: journald[26]



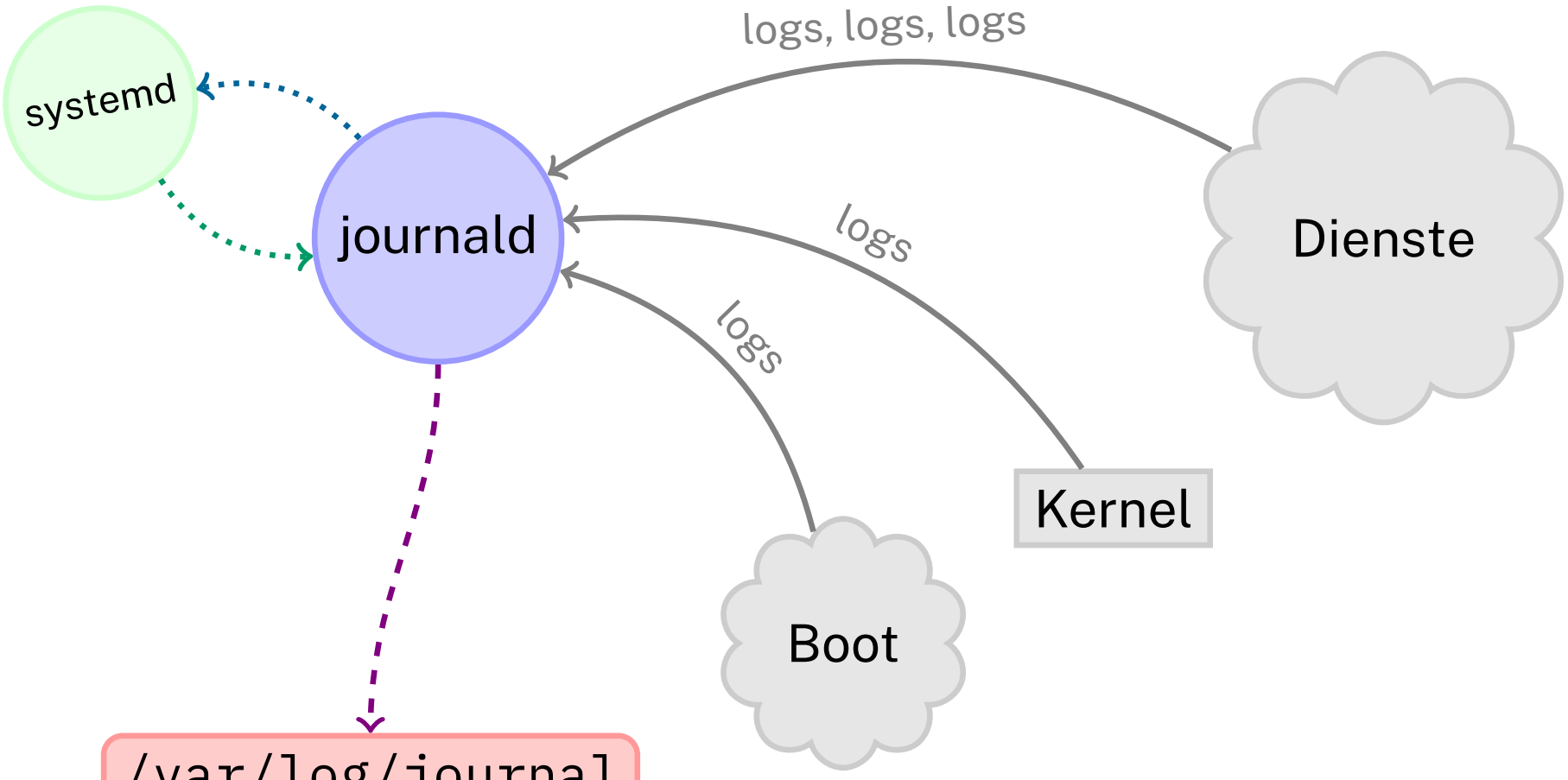
Beispiel: journald[26]



Beispiel: `journald`[26]



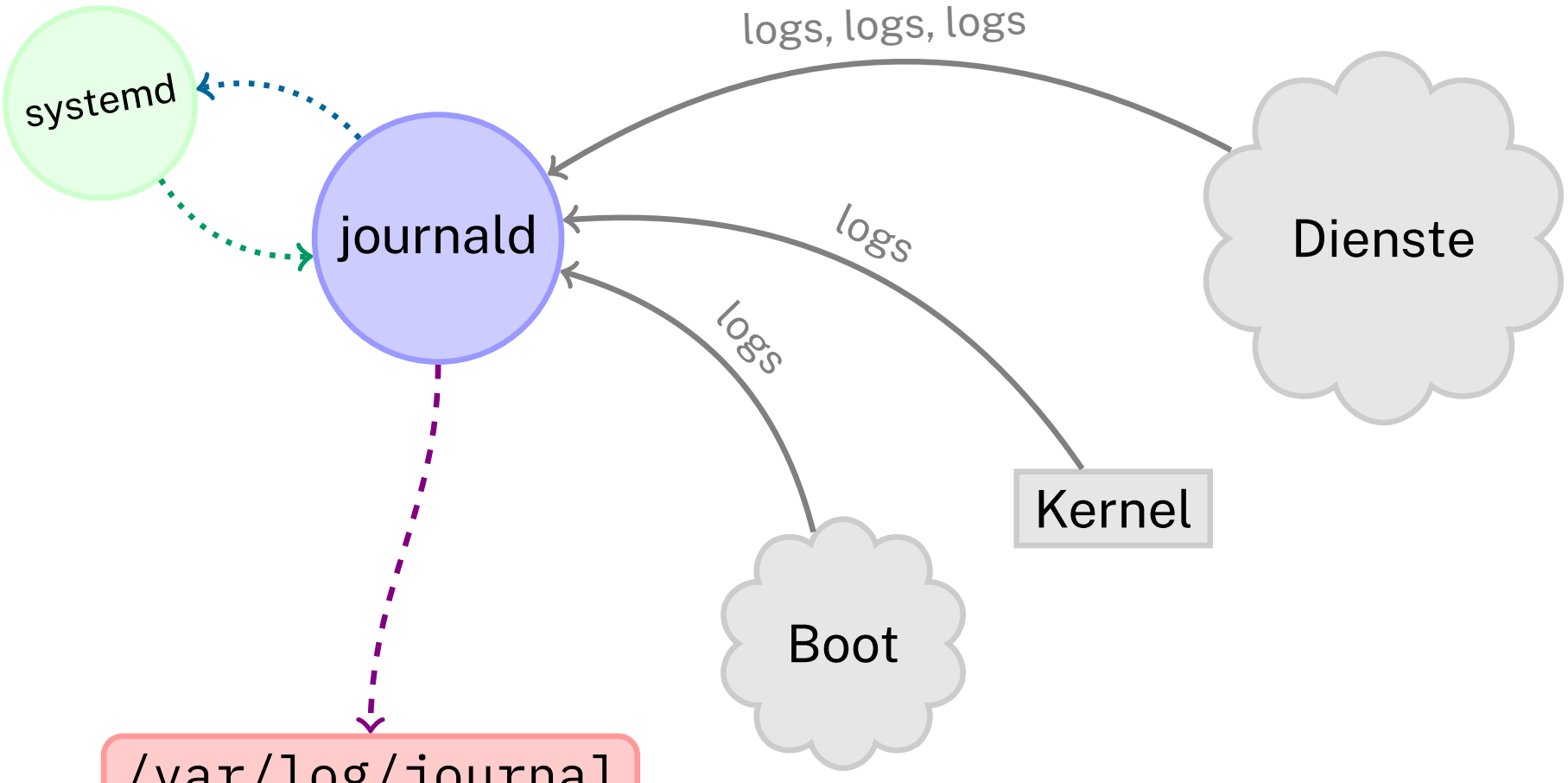
Beispiel: `journald`[26]



+ Metadaten!

... in einem Binärformat

Beispiel: `journald`[26]

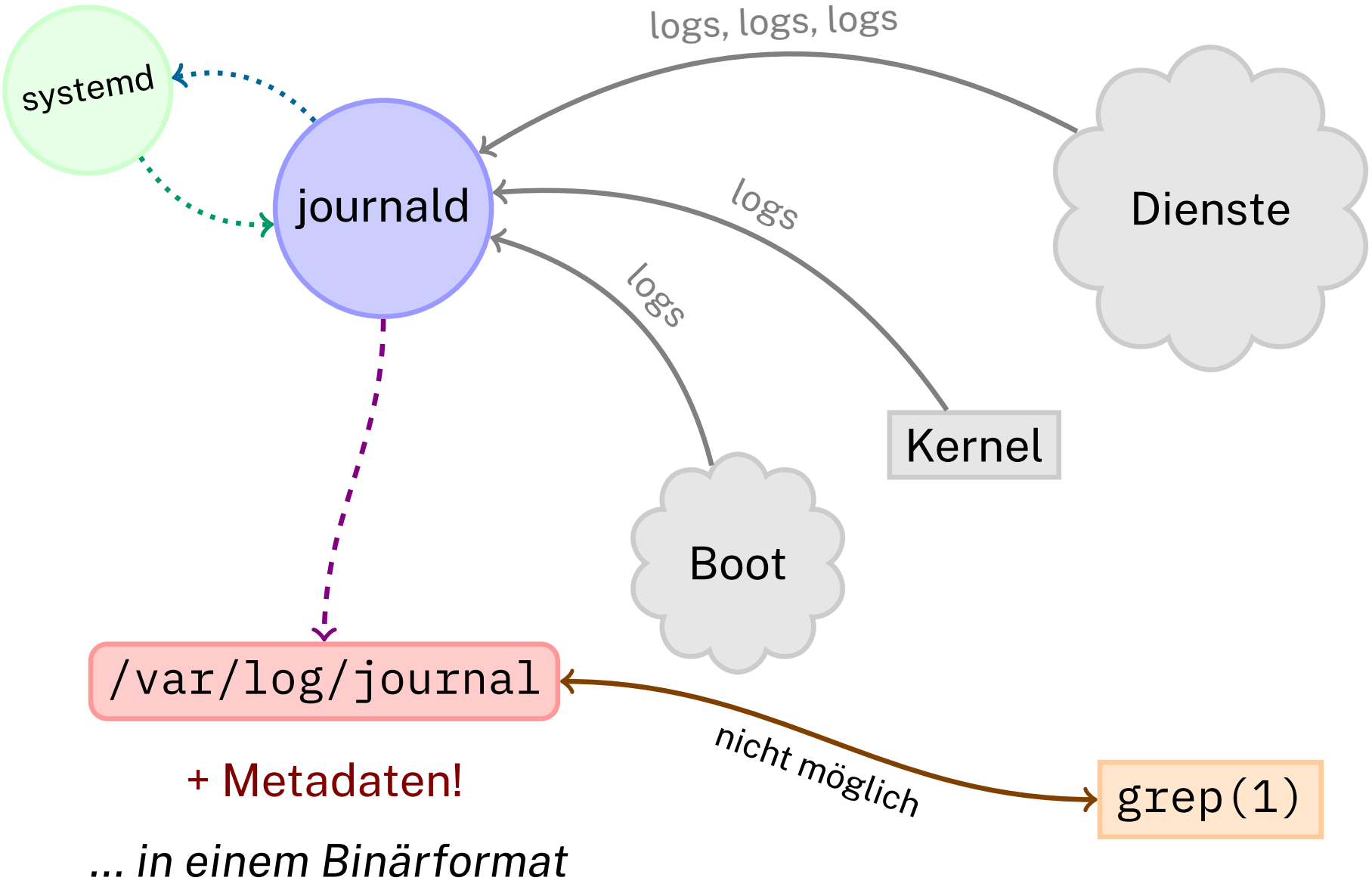


+ Metadaten!

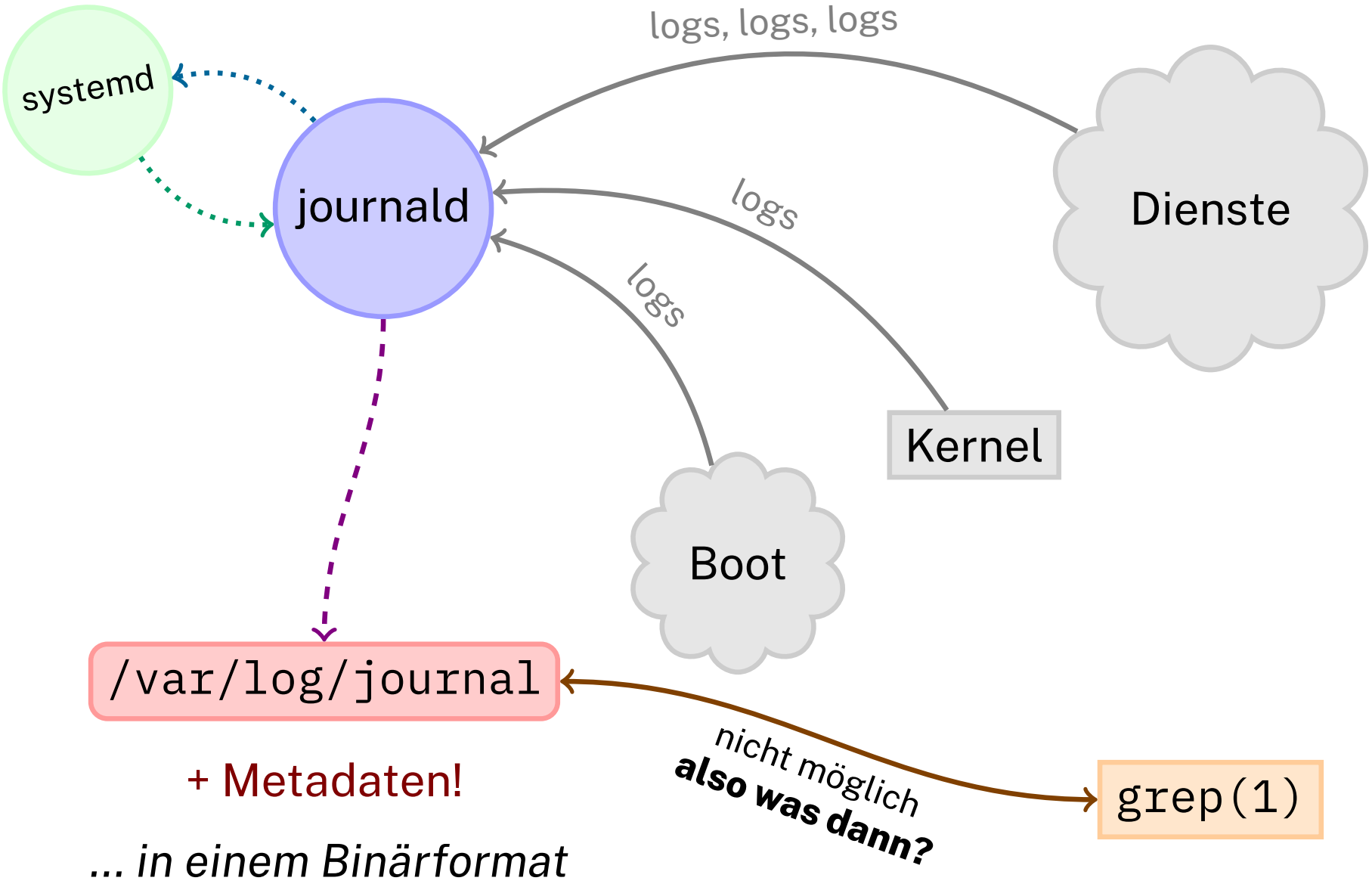
... in einem Binärformat

`grep(1)`

Beispiel: journald[26]



Beispiel: journald[26]



Schnittstelle: journalctl

```
$ journalctl
Nov 04 14:56:46 faui04p sshd[32571]: Re
Nov 04 14:56:46 faui04p sshd[32571]: Di
Nov 04 14:56:55 faui04p dbus-daemon[326
Nov 04 14:56:55 faui04p dbus-daemon[326
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:56 faui04p dbus-daemon[326
...
```

Schnittstelle: `journalctl` … zu viel

```
$ journalctl
Nov 04 14:56:46 faui04p sshd[32571]: Re
Nov 04 14:56:46 faui04p sshd[32571]: Di
Nov 04 14:56:55 faui04p dbus-daemon[326
Nov 04 14:56:55 faui04p dbus-daemon[326
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:55 faui04p org.a11y.Bus[32
Nov 04 14:56:56 faui04p dbus-daemon[326
...
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

Auf den vorigen Boot beschränken?

```
$ journalctl -b -1
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

Auf den vorigen Boot beschränken?

```
$ journalctl -b -1
```

Nur Fehler?

```
$ journalctl -p err
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

Auf den vorigen Boot beschränken?

```
$ journalctl -b -1
```

Nur Fehler?

```
$ journalctl -p err
```

Zeitlich beschränkt?

```
$ journalctl --since=2019-11-10
             --until="2019-11-11 23:59"
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

Auf den vorigen Boot beschränken?

```
$ journalctl -b -1
```

Nur Fehler?

```
$ journalctl -p err
```

Zeitlich beschränkt?

```
$ journalctl --since=2019-11-10
             --until="2019-11-11 23:59"
```

Live-Log vom Apache-Prozess?

```
$ journalctl -f -u apache
```

schlauer und genauer[18]

Auf den Boot beschränken?

```
$ journalctl -b
```

Auf den vorigen Boot beschränken?

```
$ journalctl -b -1
```

Nur Fehler?

```
$ journalctl -p err
```

Zeitlich beschränkt?

```
$ journalctl --since=2019-11-10
             --until="2019-11-11 23:59"
```

Live-Log vom Apache-Prozess?

```
$ journalctl -f -u apache
```

Nach beliebigen Metadaten gefiltert?

```
$ journalctl _UID=70 _COMM=pulseaudio
```

Vielen Dank für die Aufmerksamkeit!

Anregungen zur Diskussion:

- Versucht systemd zu viel auf einmal?
- Hat systemd es geschafft, die Linux-Welt zu standardisieren?
- Persönliche Erfahrungen?

Referenzen I

[1] Jan. 1970. URL: https://minnie.tuhs.org/cgi-bin/utree.pl?file=PDP7-Unix/cmd/init.s (besucht am 09. 11. 2019).

[2] Jan. 1979. URL: http://man.openbsd.org/UNIX-7/init (besucht am 09. 11. 2019).

[3] Rich Felker (dalias). *Broken by design: systemd*. Feb. 2014. URL: https://ewontfix.com/14/ (besucht am 10. 11. 2019).

[4] *3. Changes in Fedora for System Administrators*. URL: https://docs.fedoraproject.org/en-US/Fedora/15/html/Release_Notes/sect-Release_Notes-Changes_for_SysAdmin.html#sect-RelNotes-Boot (besucht am 09. 11. 2019).

Referenzen II

[5] Don Armstrong. *Bug#727708: call for votes on default Linux init system for jessie*. Feb. 2014. URL: https://lists.debian.org/debian-ctte/2014/02/msg00294.html (besucht am 09. 11. 2019).

[6] Andrew Ayer. *How to Crash Systemd in One Tweet*. Sep. 2016. URL: https://www.agwa.name/blog/post/how_to_crash_systemd_in_one_tweet (besucht am 10. 11. 2019).

[7] Konstantin Bläsi. *The Tragedy of systemd*. Feb. 2019. URL: https://www.youtube.com/watch?v=ImfU4Q58zrg (besucht am 09. 11. 2019).

Referenzen III

[8] BrandonPhilips. *freedesktop.org - Software/systemd*. Juli 2010. URL: http://web.archive.org/web/20100728210058/https://www.freedesktop.org/wiki/Software/systemd/ (besucht am 09.11.2019).

[9] *CVE-2018-15686*. Okt. 2018. URL: https://www.cvedetails.com/cve/CVE-2018-15686/ (besucht am 10.11.2019).

[10] *Debate Init System To Use*. URL: https://wiki.debian.org/Debate/initsystem.

[11] *Debian committee members vote for systemd as init system*. Feb. 2014. URL: https://news.ycombinator.com/item?id=7203364 (besucht am 09.11.2019).

Referenzen IV

[12] Suckless e.V. *Structural and semantic deficiencies in the systemd architecture for real-world service management, a technical treatise*. URL: http://suckless.org/sucks/systemd/ (besucht am 10. 11. 2019).

[13] *Fedora 14 talking points*. Sep. 2016. URL: https://fedoraproject.org/wiki/Fedora_14_talking_points#Systemd (besucht am 09. 11. 2019).

[14] Robert Graham. *Systemd is bad parsing and should feel bad*. Okt. 2018. URL: https://blog.erratasec.com/2018/10/systemd-is-bad-parsing-and-should-feel.html (besucht am 10. 11. 2019).

Referenzen V

[15] David Herrmann. *[systemd-devel] [ANNOUNCE] Git development moved to github*. Juni 2015. URL: https://lists.freedesktop.org/archives/systemd-devel/2015-June/032652.html (besucht am 09. 11. 2019).

[16] *Interview mit Lennart Poettering, Entwickler Systemd*. Mai 2011. URL: https://video.golem.de/oss/4823/interview-mit-lennart-poettering-entwickler-systemd.html (besucht am 09. 11. 2019).

[17] Ian Jackson. *General Resolution: init system coupling*. Okt. 2014. URL: https://www.debian.org/vote/2014/vote_003 (besucht am 09. 11. 2019).

[18] *JOURNALCTL(1): journalctl - Query the systemd journal*.

Referenzen VI

[19] Sebastian Krahmer. *Headsup: systemd v228 local root exploit*. Jan. 2017. URL: https://www.openwall.com/lists/oss-security/2017/01/24/4 (besucht am 10. 11. 2019).

[20] Steven McDonald. *Why I dislike systemd*. Juni 2015. URL: http://www.steven-mcdonald.id.au/articles/systemd.shtml (besucht am 10. 11. 2019).

[21] Lennart Poettering. *Rethinking PID 1*. Apr. 2010. URL: http://0pointer.net/blog/projects/systemd.html (besucht am 09. 11. 2019).

Referenzen VII

[22] Lennart Poettering. *systemd for Administrators, Part III*. Okt. 2010. URL: http://0pointer.de/blog/projects/systemd-for-admins-3.html (besucht am 09.11.2019).

[23] C J Silverio. *Structural and semantic deficiencies in the systemd architecture for real-world service management, a technical treatise*. Okt. 2015. URL: https://blog.darknedgy.net/technology/2015/10/11/0/ (besucht am 10.11.2019).

[24] *So You Want To Write A Linux Userland*. Mai 2015. URL: http://tilde.town/~elly/userland.txt (besucht am 09.11.2019).

Referenzen VIII

[25] *Systemd Bugzilla*. URL: https://web.archive.org/web/20131011183817/https://bugs.freedesktop.org/enter_bug.cgi?product=systemd (besucht am 09.11.2019).

[26] *systemd-journald.service(8): systemd-journald.service, systemd-journald.socket, systemd-journald-dev-log.socket, systemd-journald-audit.socket, systemd-journald - Journal service*.

[27] *systemd-logind.service(8): systemd-logind.service, systemd-logind - Login manager*.

[28] *systemd-networkd.service(8): systemd-networkd.service, systemd-networkd - Network manager*.

Referenzen IX

[29] *systemd(1): systemd, init - systemd system and service manager*.

[30] *The systemd-devel Archives*. URL: https://lists.freedesktop.org/archives/systemd-devel/ (besucht am 09. 11. 2019).

[31] John E. Vincent. *Systemd-redux*. Nov. 2014. URL: http://blog.lusis.org/blog/2014/11/20/systemd-redux/ (besucht am 10. 11. 2019).