



Thunderbolt 3 and GNU/Linux

FOSDEM 2018

Christian Kellner, PhD
Desktop Hardware Enablement
04/02/2018

What is this  THUNDERBOLT™, anyway?

“The USB-C that does it all”

*Intel**

* <https://thunderbolttechnology.net/>

Thunderbolt 3 – Overview

- USB type C connector (one port to confuse them all)
- 40 Gb/s
- 4 PCI Express (Gen 3) lanes
- 8 DisplayPort (1.2) lanes
- Native USB 3.1
- Daisy-chain up to 6 devices
- Up to 100 W for charging, 15W for devices

- Networking, external Graphic
- Docks, docks, docks



Thunderbolt 3 – Connection Modes

USB ONLY

Active when USB devices are plugged in.
Behaves as a normal USB-C 3.1 port.



DISPLAYPORT ONLY

Switch pins of USB-C into DP alternate mode. TB will act as a router for DP data from GFX to USB-C port



DP & USB MULTI-FUNCTION

One high-speed pair is used for DP.

The other high-speed pair is used for USB 3.1

THUNDERBOLT 3

All 4 high speeds links active (at 10/20 Gbps).
max 4 PCIe Gen 3 lanes
max 2 DisplayPort links

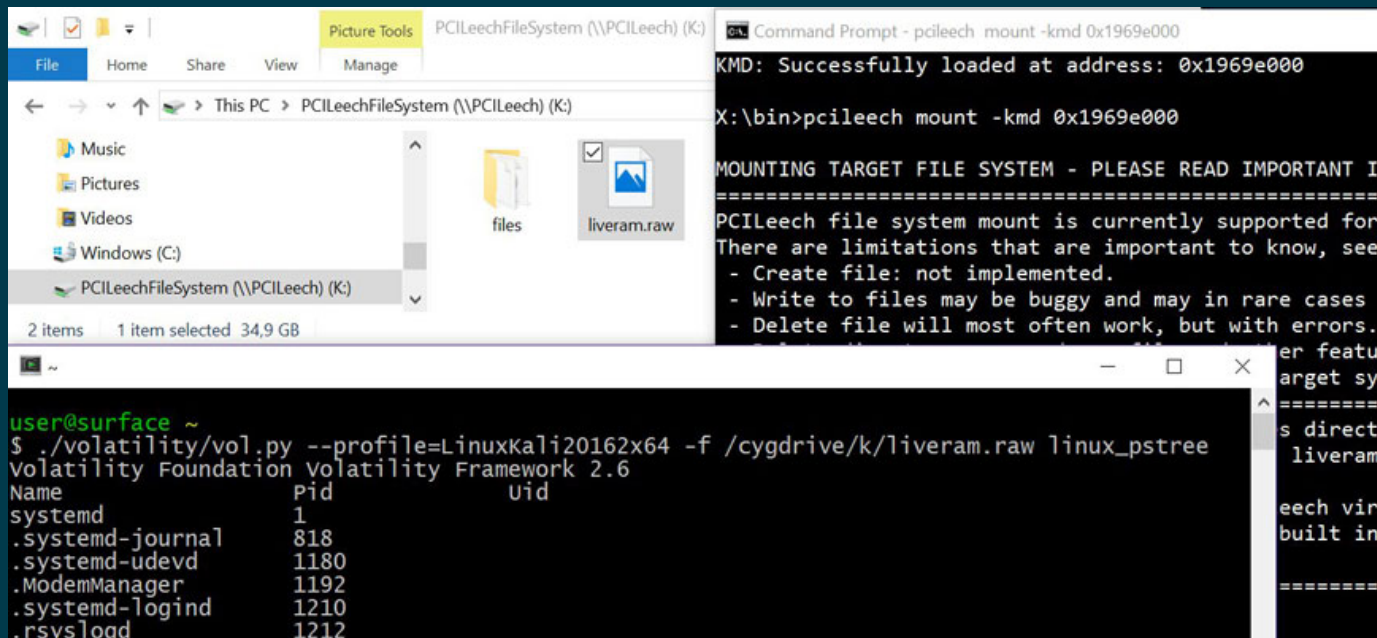


POWER DELIVERY & CHARGING

THUNDERBOLT NETWORKING

Thunderbolt – Security ???

Thunderbolt is PCIe → DMA → DMA attacks



The screenshot shows a Windows File Explorer window displaying a mounted PCIleech file system. The address bar shows 'This PC > PCIleechFileSystem (\\PCIleech) (K:)'. The file list contains two items: 'files' and 'liveram.raw'. Below the File Explorer, a Windows Command Prompt window is open, showing the execution of the 'pcileech mount' command. The output indicates that the kernel module was successfully loaded at address 0x1969e000 and that the file system is being mounted. A warning message is displayed, stating that the PCIleech file system mount is currently supported for limited operations: 'Create file: not implemented', 'Write to files may be buggy and may in rare cases cause data corruption', and 'Delete file will most often work, but with errors.'

```
user@surface ~
$ ./volatility/vol.py --profile=LinuxKali20162x64 -f /cygdrive/k/liveram.raw linux_pstree
Volatility Foundation Volatility Framework 2.6
Name          Pid      Uid
systemd       1
.systemd-journal 818
.systemd-udev 1180
.ModemManager 1192
.systemd-logind 1210
.rsyslogd     1212
```

<https://github.com/ufrisk/pcileech>

Thunderbolt 3 – Security Modes

NONE

No Security. Doh.

All devices are authorized by default.

DP ONLY

Display Port only.
You guessed right.

USER

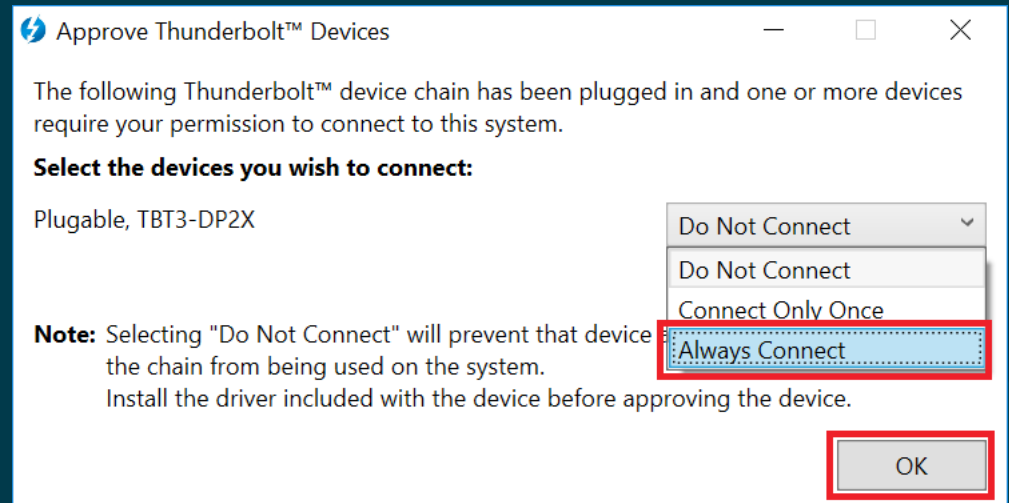
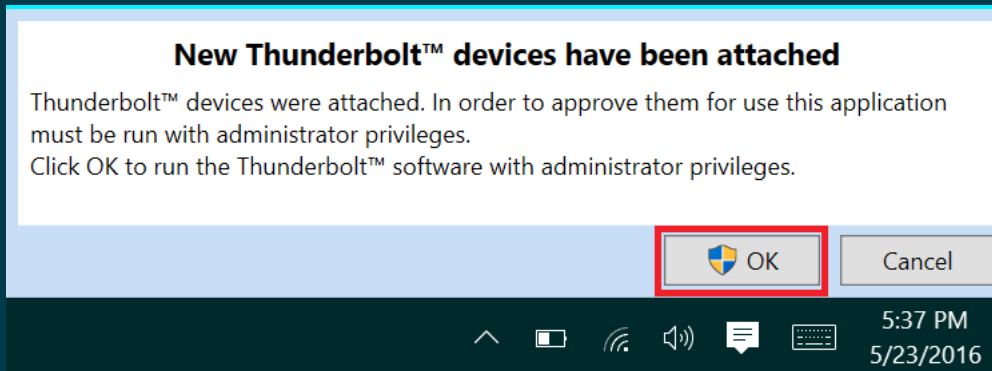
Thunderbolt devices need to be authorized. Only then are PCIe lanes activated.

SECURE

Thunderbolt devices need to be authorized. Their identity can be verified via a key.

Thunderbolt 3 – Security Modes

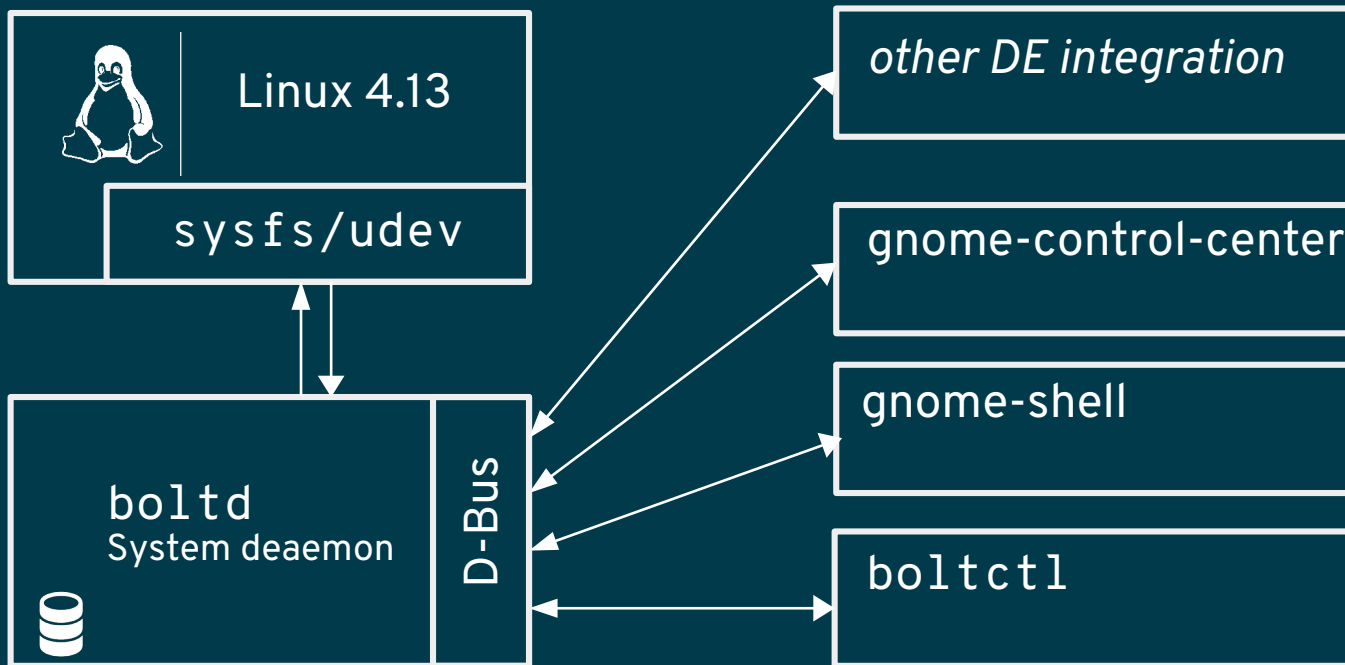
In the land of the dialogs no we are not doing that.



Thunderbolt and GNU/Linux

Thunderbolt & GNU/Linux

Overview



Kernel Interface

Linux kernel 4.13+ provide a sysfs interface

```
/sys/bus/thunderbolt/  
├── devices  
│   ├── domain0 → 0-0/ security subsystem@ uevent [...]  
│   ├── 0-0 → 0-1/ authorized device device_name vendor_name unique_id [...]  
│   ├── 0-1 → 0-301/ authorized [...] key [...] unique_id  
│   └── 0-301 → [...] nvm_active2/ nvm_non_active2/ nvm_version nvm_authenticate
```

```
# echo 1 > /sys/bus/thunderbolt/devices/0-1/authorized
```

```
# key=$(openssl rand -hex 32)
```

```
# echo $key > /sys/bus/thunderbolt/devices/0-1/key
```

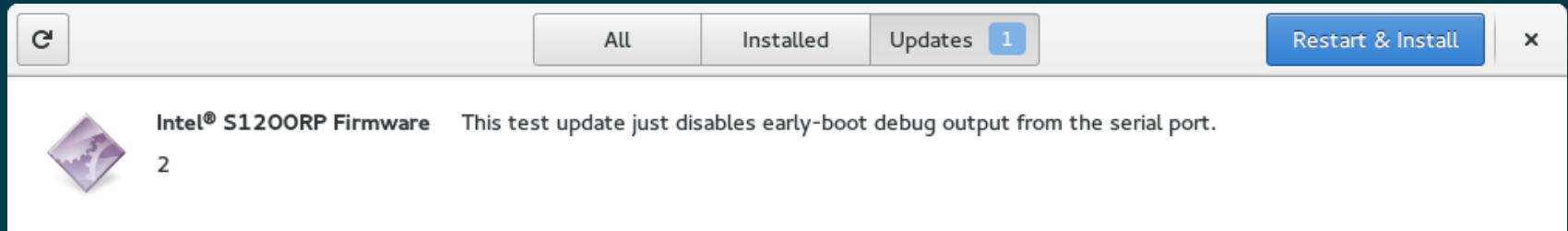
```
# echo 1 > /sys/bus/thunderbolt/devices/0-1/authorized
```

```
# echo $key > /sys/bus/thunderbolt/devices/0-1/key
```

```
# echo 2 > /sys/bus/thunderbolt/devices/0-1/authorized
```

Thunderbolt firmware updates

fwupd & Linux Vendor Firmware Service (LVFS)



```
# get current version
nvm_version

# write new firmware to
nvm_non_active2/nvmem

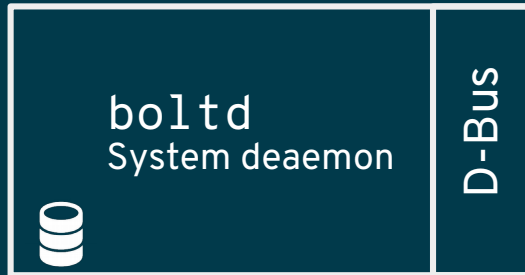
# start updating
nvm_authenticate
```



* <https://fwupd.org/>

Thunderbolt & GNU/Linux

boltd



- System daemon, activated on demand
- D-Bus API to manage devices, signal device “changes”
- Authorize, enroll (authorize and store)
- Polkit to secure the D-Bus API
- Device “database” of previously enrolled devices and their policy
- *Paranoid* (now *fortify*) mode
- Needs a *policy agent* to do the initial authorization, enrollment

bolt

D-Bus API: manager interface

The screenshot displays a D-Bus API viewer for the `org.freedesktop.bolt` service. The address is `unix:path=/var/run/dbus/system_bus_socket` and the name is `org.freedesktop.bolt`. The object path is `/org/freedesktop/bolt`. The interface `org.freedesktop.bolt1.Manager` is expanded, showing the following details:

- Interfaces:**
 - `org.freedesktop.DBus.Introspectable`
 - `org.freedesktop.DBus.Peer`
 - `org.freedesktop.DBus.Properties`
 - `org.freedesktop.bolt1.Manager`
- Methods:**
 - `DeviceByUid (String uid) ⇨ (Object Path device)`
 - `EnrollDevice (String uid, UInt32 policy, UInt32 flags) ⇨ (Object Path device)`
 - `ForgetDevice (String uid) ⇨ ()`
 - `ListDevices () ⇨ (Array of [Object Path] devices)`
- Properties:**
 - `Boolean Fortify (read / write)`
 - `Boolean Probing (read)`
 - `UInt32 DefaultPolicy (read)`
 - `UInt32 Version (read)`
- Signals:**
 - `DeviceAdded (Object Path)`
 - `DeviceRemoved (Object Path)`

boltd

D-Bus API: manager interface

```
▼ /org/freedesktop/bolt/devices/c9030000_0091_8718_a2f6_f2e5ec4...
└─ Interfaces
  ├── org.freedesktop.DBus.Introspectable
  ├── org.freedesktop.DBus.Peer
  ├── org.freedesktop.DBus.Properties
  └─ org.freedesktop.bolt1.Device
    ├── Methods
    │   └─ Authorize (UInt32 flags) → ()
    └─ Properties
        ├── Boolean Stored (read)
        ├── String Name (read)
        ├── String Parent (read)
        ├── String SysfsPath (read)
        ├── String Uid (read)
        ├── String Vendor (read)
        ├── UInt32 Key (read)
        ├── UInt32 Policy (read)
        ├── UInt32 Security (read)
        ├── UInt32 Status (read)
        └─ UInt32 Type (read)
  ► /org/freedesktop/bolt/devices/de030000_00b1_9f08_a21e_f3e516...
```

boltctl

cli interface

```
1/1 ▾ + [f] [m] Bolt
1: gicmo@hanada [Code/src/bolt] ▾
λ ~/C/s/bolt → boltctl
○ Dell Thunderbolt Dock
├─ uuid:      c9030000-0091-8718-a2f6-
├─ vendor:    Dell
├─ status:    disconnected
├─ stored:    yes
│   └─ policy: auto
│      └─ key:  yes
● T470s
├─ uuid:      dc010000-00a2-a088-2059-
├─ vendor:    Lenovo
├─ status:    authorized
│   └─ security: secure
└─ stored:    no
● ThinkPad Thunderbolt 3 Dock
├─ uuid:      d8030000-0060-5708-23a1-
├─ vendor:    Lenovo
├─ status:    authorized
│   └─ security: secure
```

```
1/1 ▾ + [f] [m] Tilix: Default
1: gicmo@hanada [~] ▾
λ ~ → boltctl --help
Usage:
boltctl [OPTION...] [COMMAND]

Commands:
authorize      Authorize a device
enroll         Authorize and store a device in the database
forget         Remove a stored device from the database
info           Show information about a device
list           List connected and stored devices
monitor        Listen and print changes

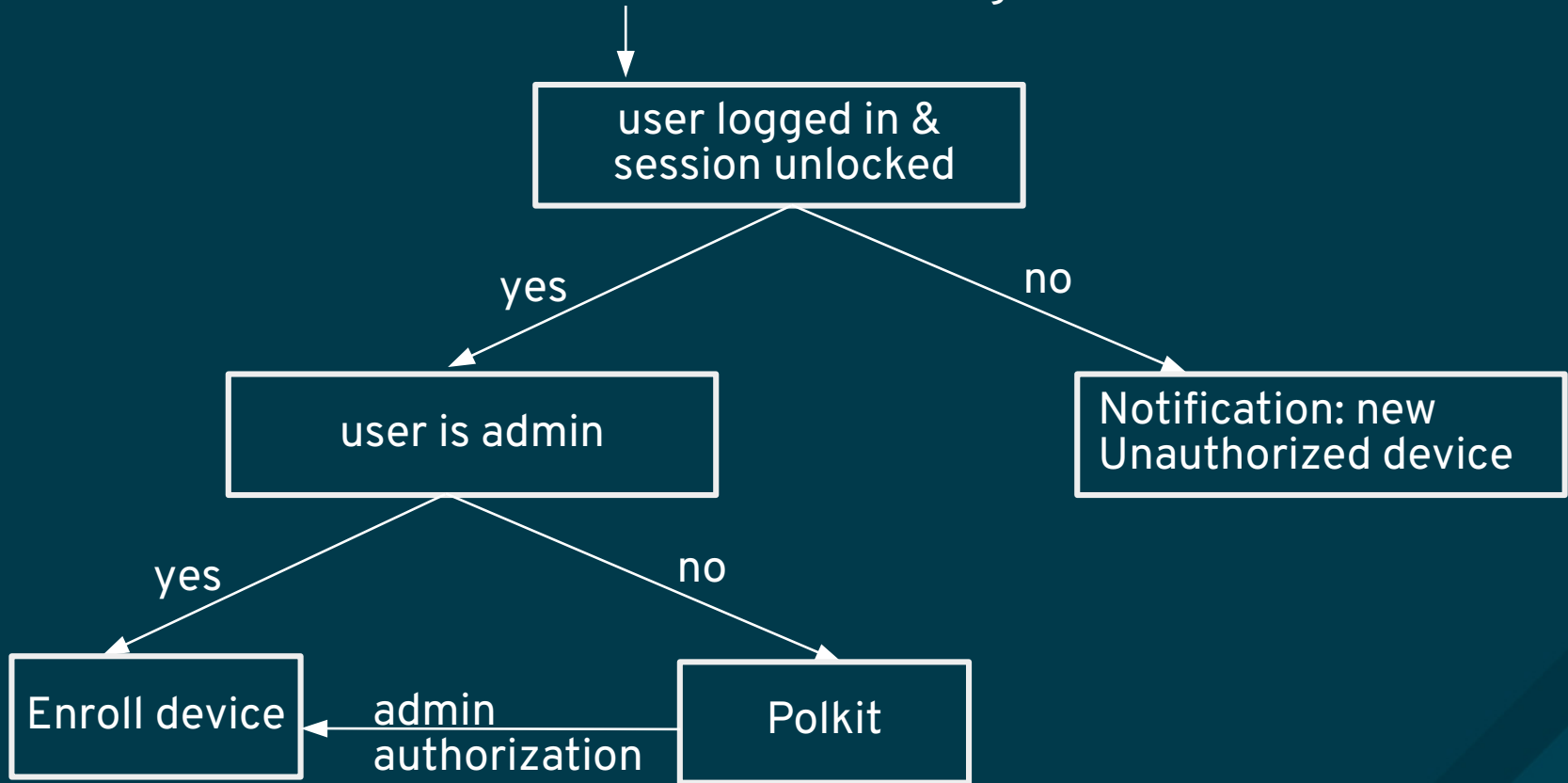
Help Options:
-h, --help     Show help options

λ ~ →
```


gnome-shell

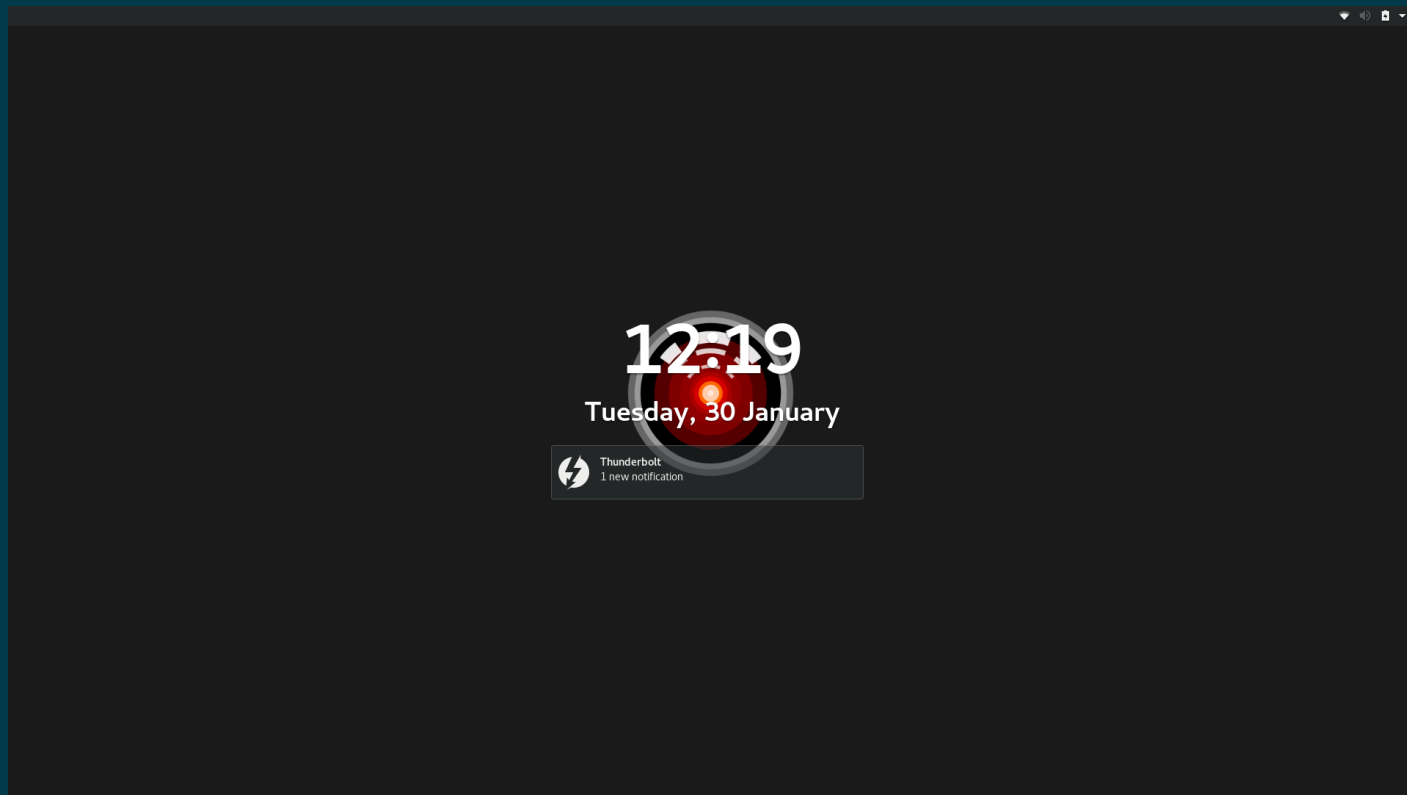
Acts as a *policy agent*

Listen to “device-added” D-Bus signal from bolt d



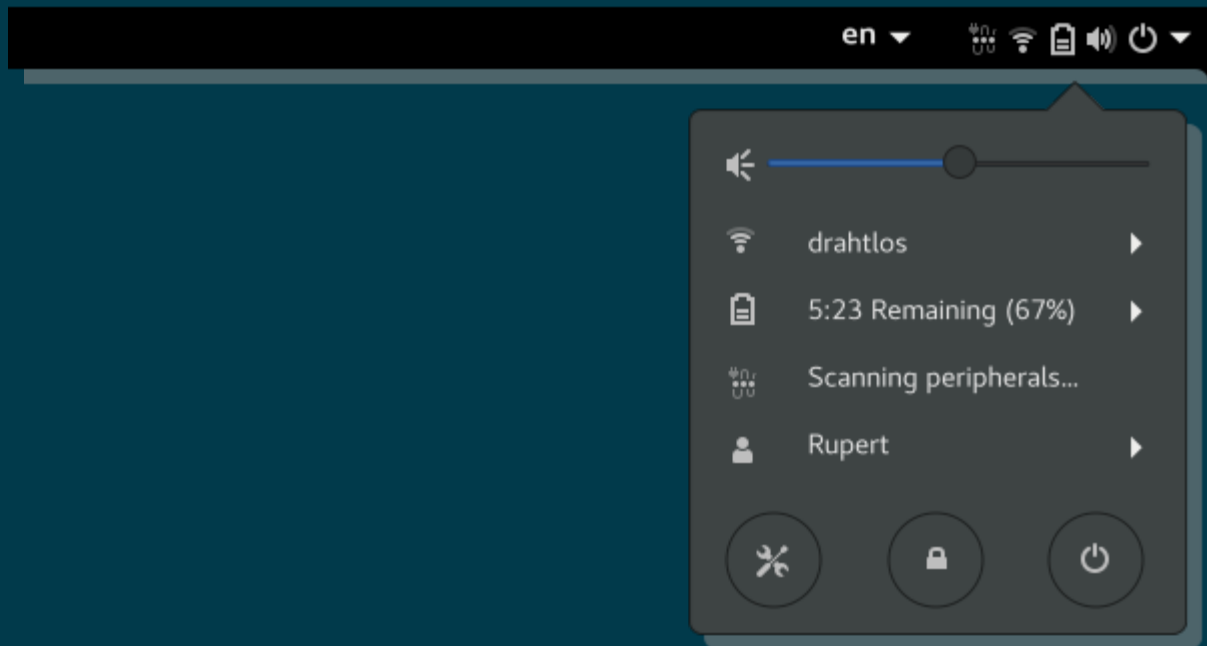
gnome-shell

Acts as a *policy agent*



gnome-shell

provide UI feedback about thunderbolt bus activity



gnome-control-center

manage devices, provide feedback

The screenshot shows the GNOME Control Center interface for managing Thunderbolt devices. The left sidebar lists various device categories, with 'Thunderbolt' selected. The main panel displays a list of Thunderbolt devices:

| Device Name | Status |
|--|---------------|
| Lenovo ThinkPad Thunderbolt 3 Dock Device was authorized for the first time. | Authorized |
| Dell Thunderbolt Dock Device will be securely authorized when connected. | Not connected |

A detailed dialog box for the 'Lenovo ThinkPad Thunderbolt 3 Dock' is shown, displaying the following information:

- Status: Authorized
- Stored: Yes
- Policy: authorize on connect (auto)
- Key State: key present
- Details:
 - UUID: d8030000-0060-5708-23a1-0e3109d16a1c
 - Parent: dc010000-00a2-a088-2059-ac440f92d31c

A 'Forget Device' button is visible at the bottom right of the dialog box.

gnome-control-center

manage devices, provide feedback

The screenshot displays the GNOME Control Center interface. On the left, a sidebar lists various device categories: Printers, Keyboard, Mouse & Touchpad, Displays, Removable Media, Wacom Tablet, Colour, and Thunderbolt. The 'Thunderbolt' category is selected and highlighted in blue. The main content area is titled 'Thunderbolt' and features a warning icon and the heading 'Authorization issues'. Below this, a message states: 'One or more thunderbolt devices are connected but not authorized and therefore will not work properly. [Learn more](#)'. A table titled 'Thunderbolt devices' lists two devices: 'Lenovo ThinkPad Thunderbolt 3 Dock' (status: Connected, with a warning icon and the note 'The device needs authorization. Please re-plug the device.') and 'Dell Thunderbolt Dock' (status: Not connected, with the note 'Device will be securely authorized when connected.'). In the foreground, a smaller window titled 'ThinkPad Thunderbolt 3 Dock' is open, showing a blue notification bar with a warning icon and the text: 'This device is connected but not authorized. Re-plug the device to authorize it. [Learn more](#)'. Below the notification, the device name 'Lenovo ThinkPad Thunderbolt 3 Dock' is displayed, along with its status 'Connected' and 'Stored No'. A 'Details:' link is also visible.



THANK YOU

 github.com/gicmo/bolt

christian.kellner.me