TechTrain 2021

Genode

Фреймворк для
создания
операционных систем

Сергей Платонов

План

В чём проблема?

Микроядро

Пример

Архитектура

Инструменты

Disclaimer

Disclaimer

не разработчик ОС

Disclaimer

не разработчик ОС

применение: embedded

Disclaimer

не разработчик ОС

применение: embedded

сравнение с linux

Disclaimer

не разработчик ОС

применение: embedded

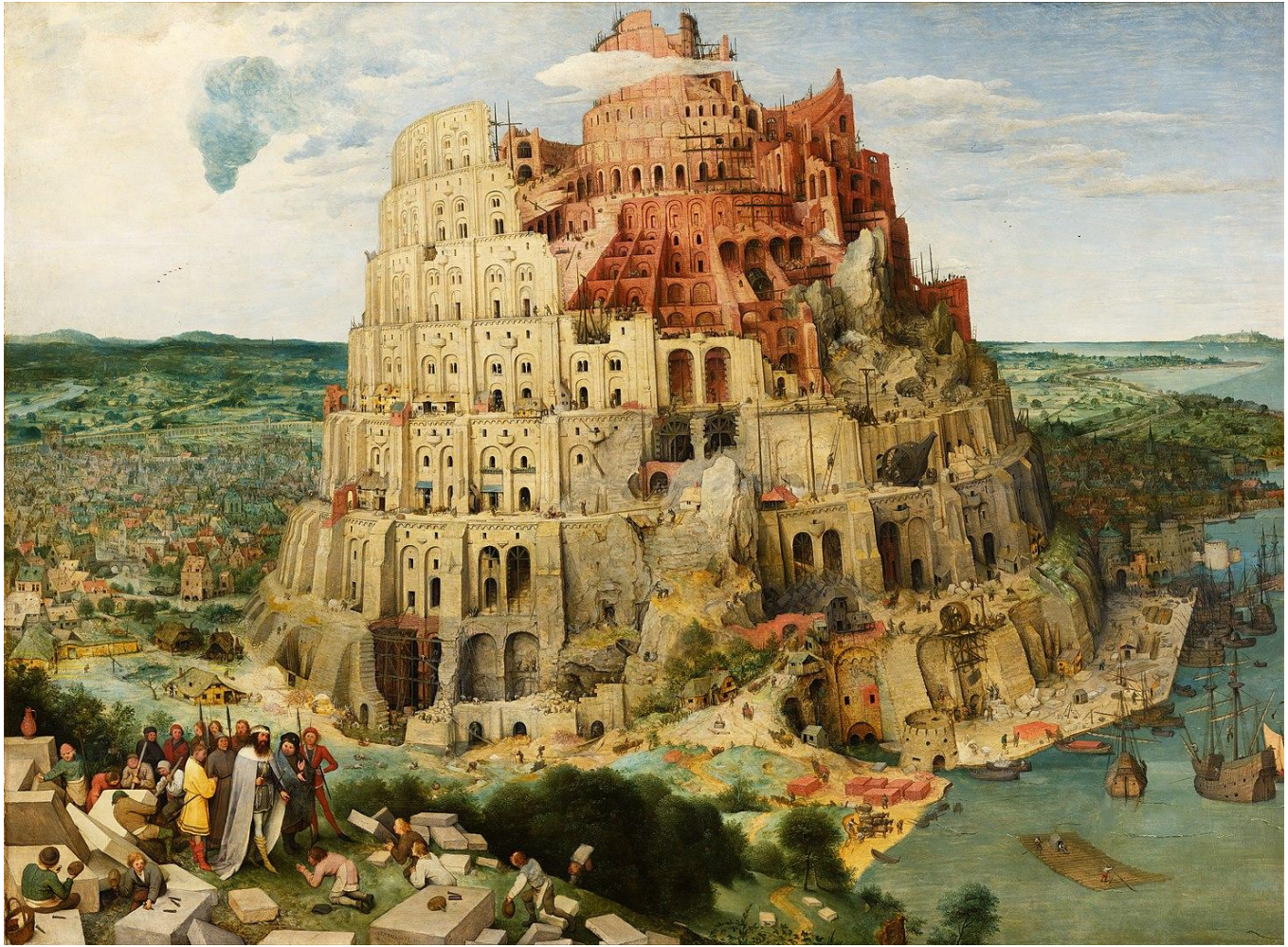
сравнение с linux

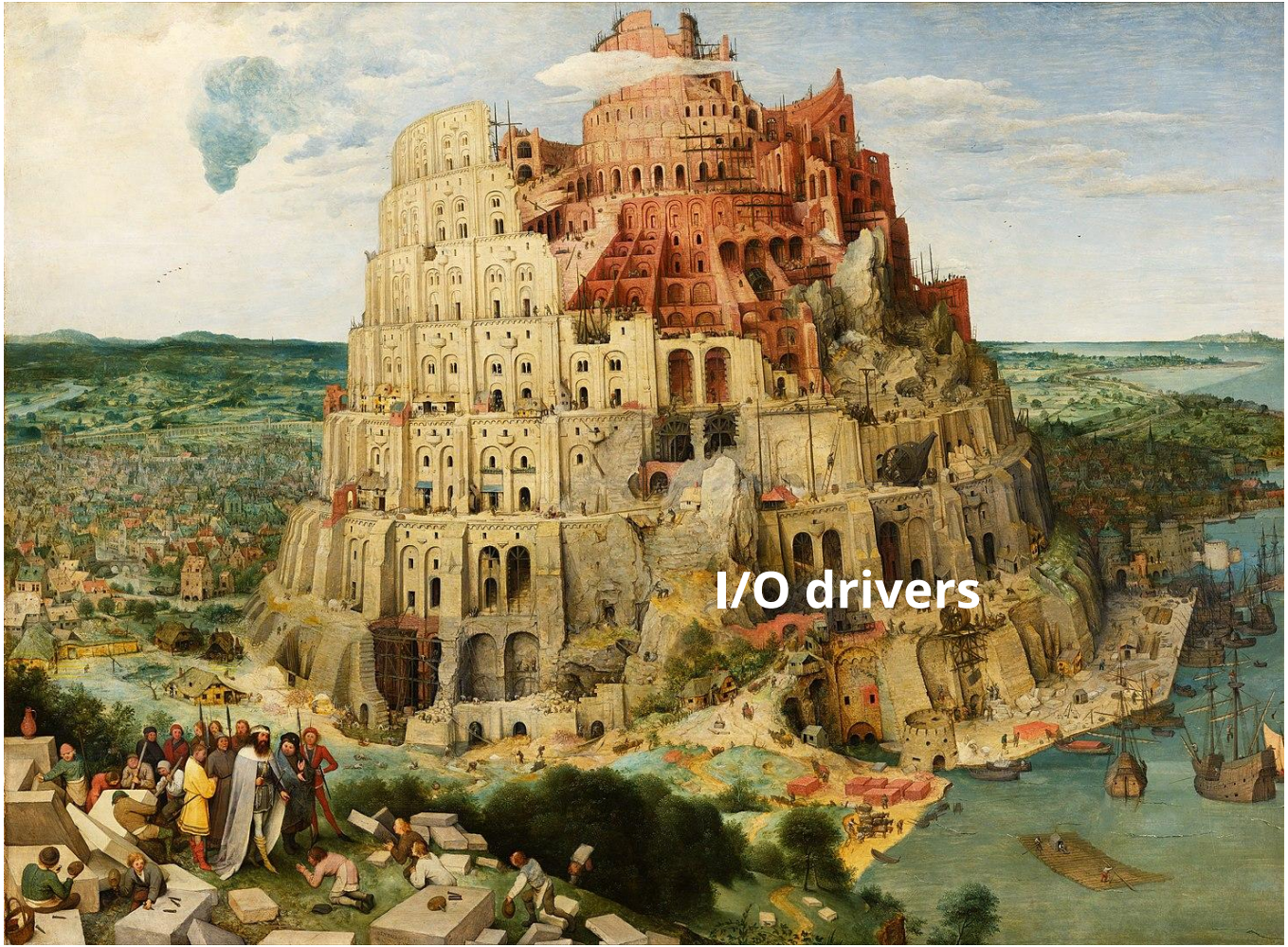
это не введение в Genode

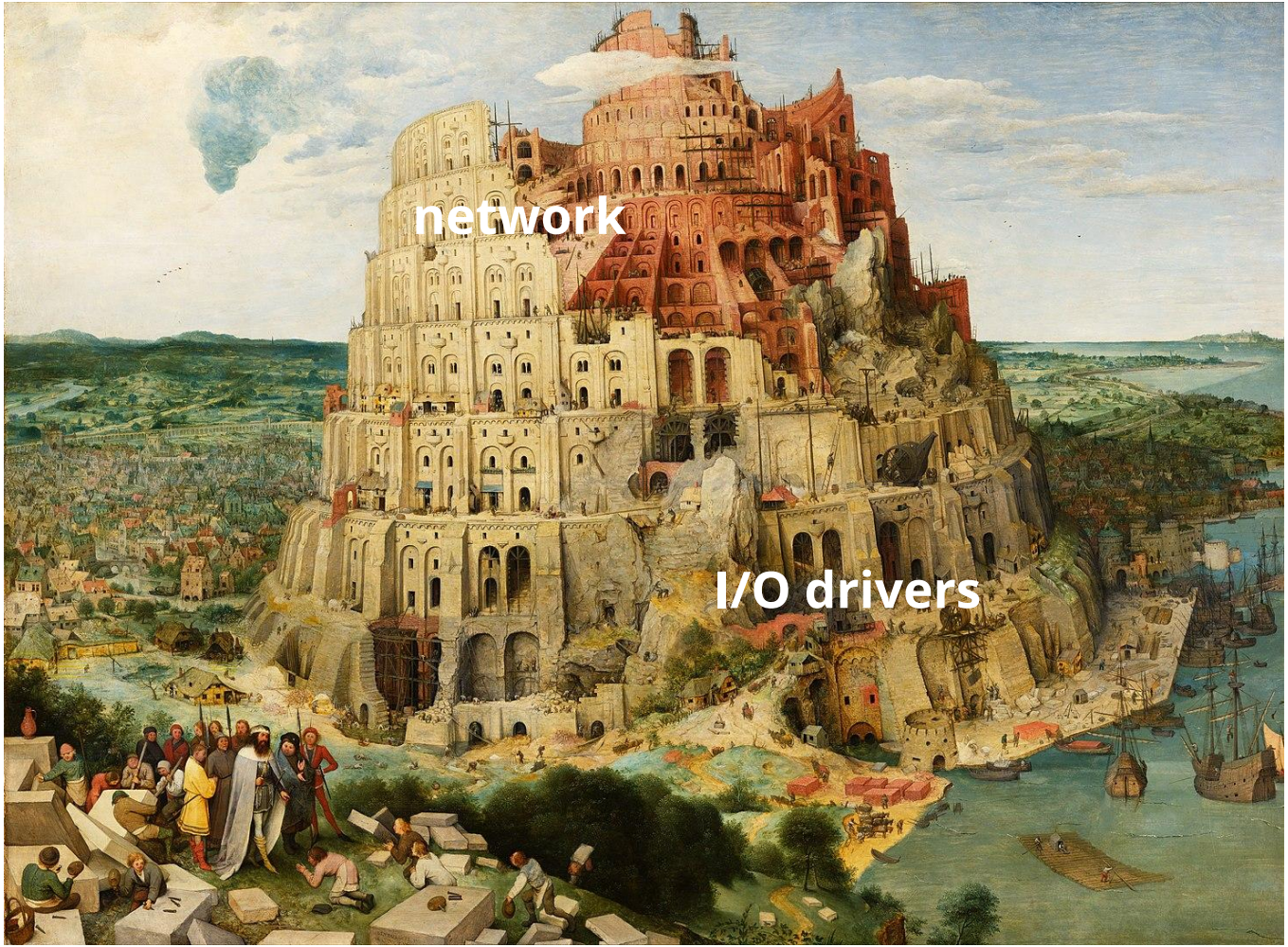
Цель доклада

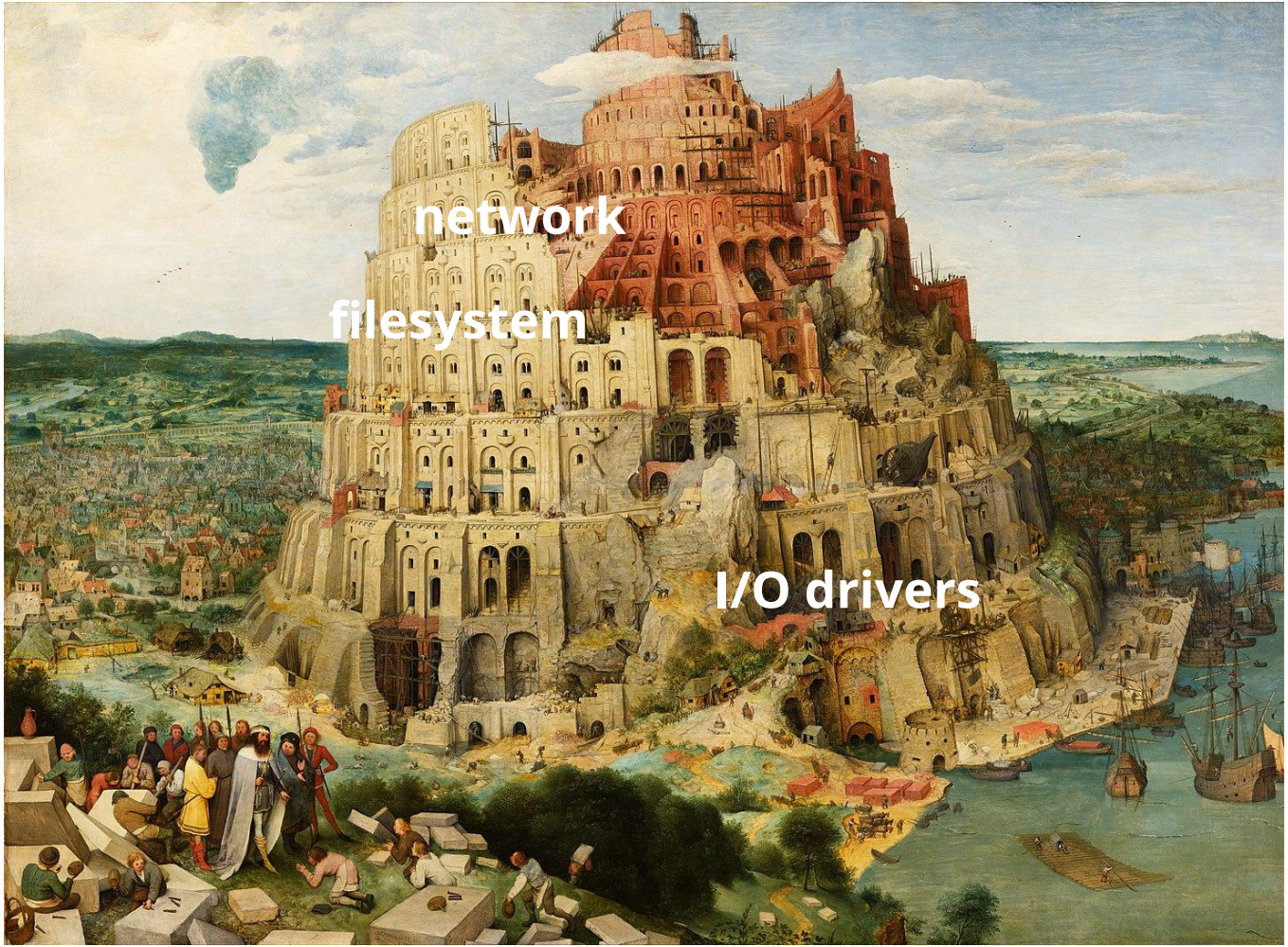
знакомство с Genode

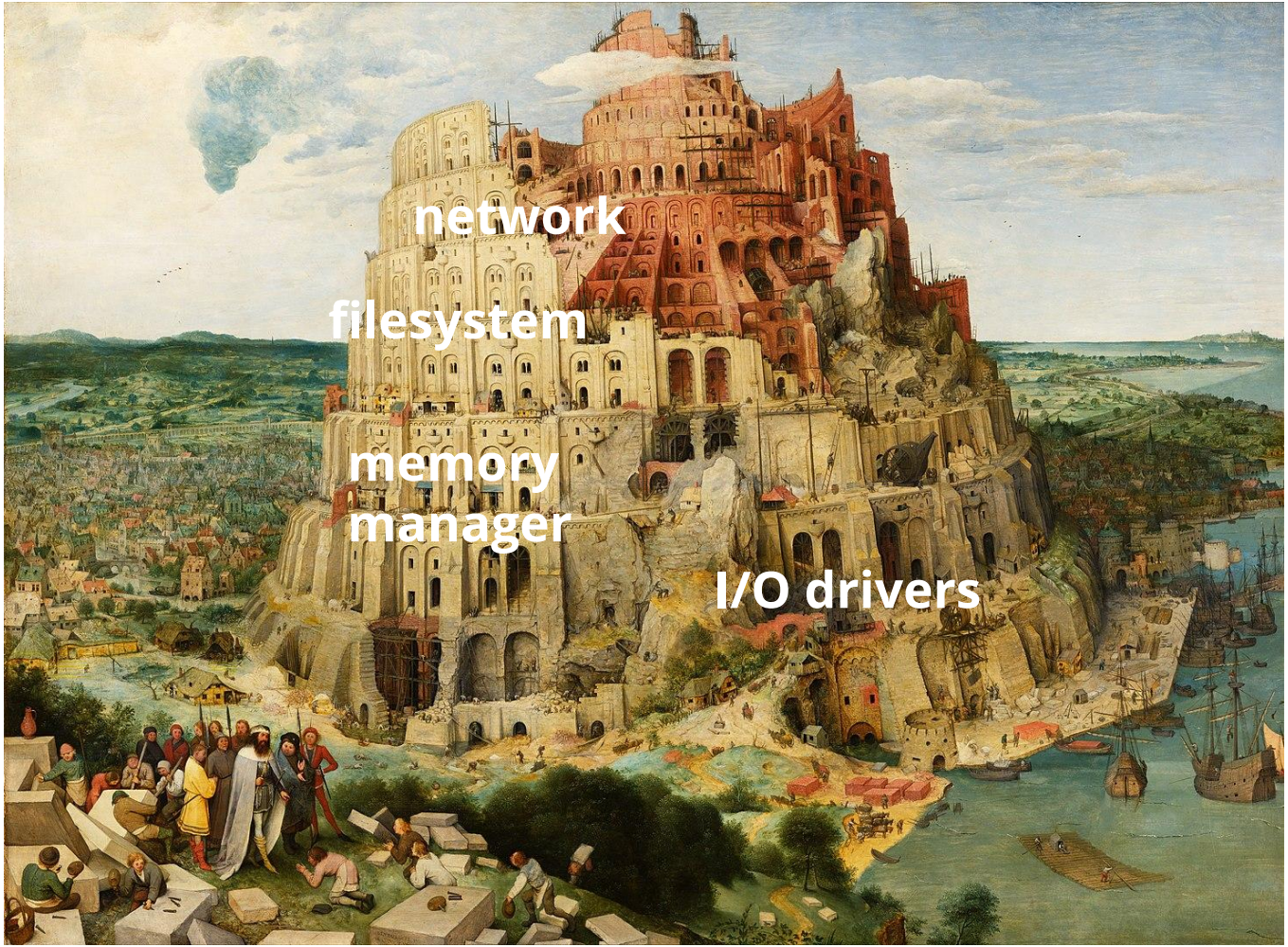
простота конфигурации

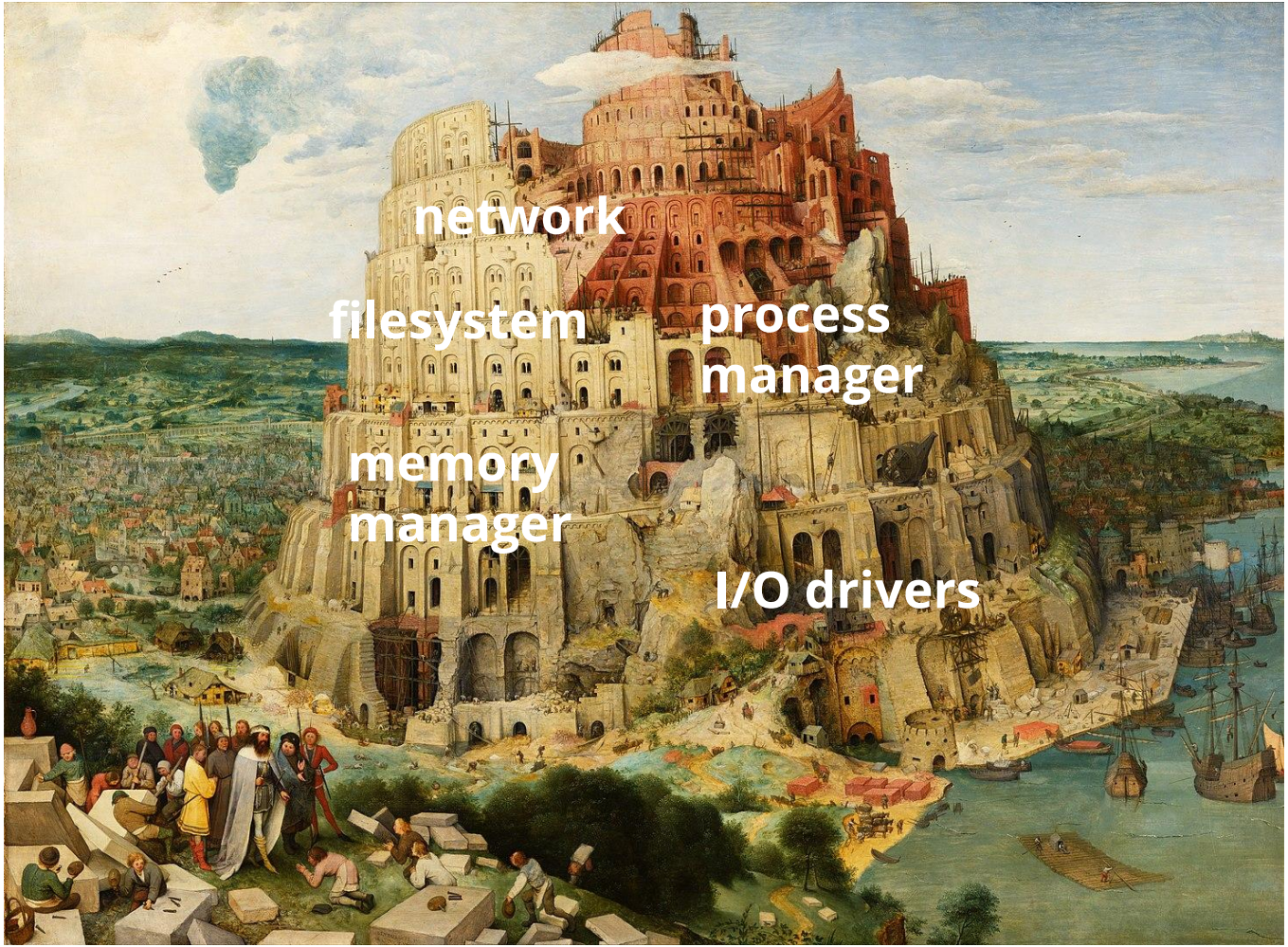



I/O drivers


network
I/O drivers


network
filesystem
I/O drivers


network
filesystem
memory
manager
I/O drivers


network
filesystem
process manager
memory manager
I/O drivers

Genode

Genode

микроядро

Genode

микроядро

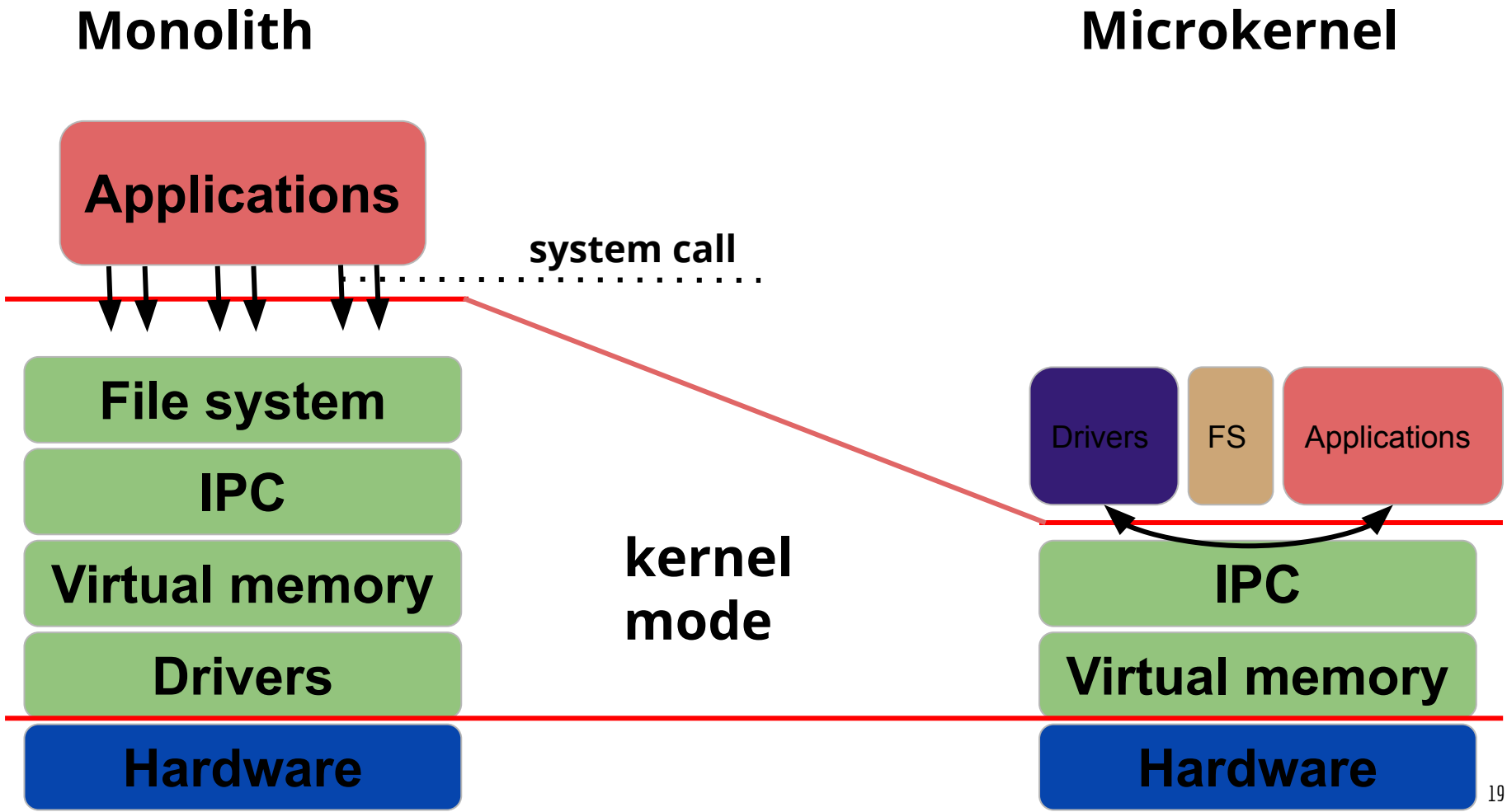
capability-based модель безопасности

Genode

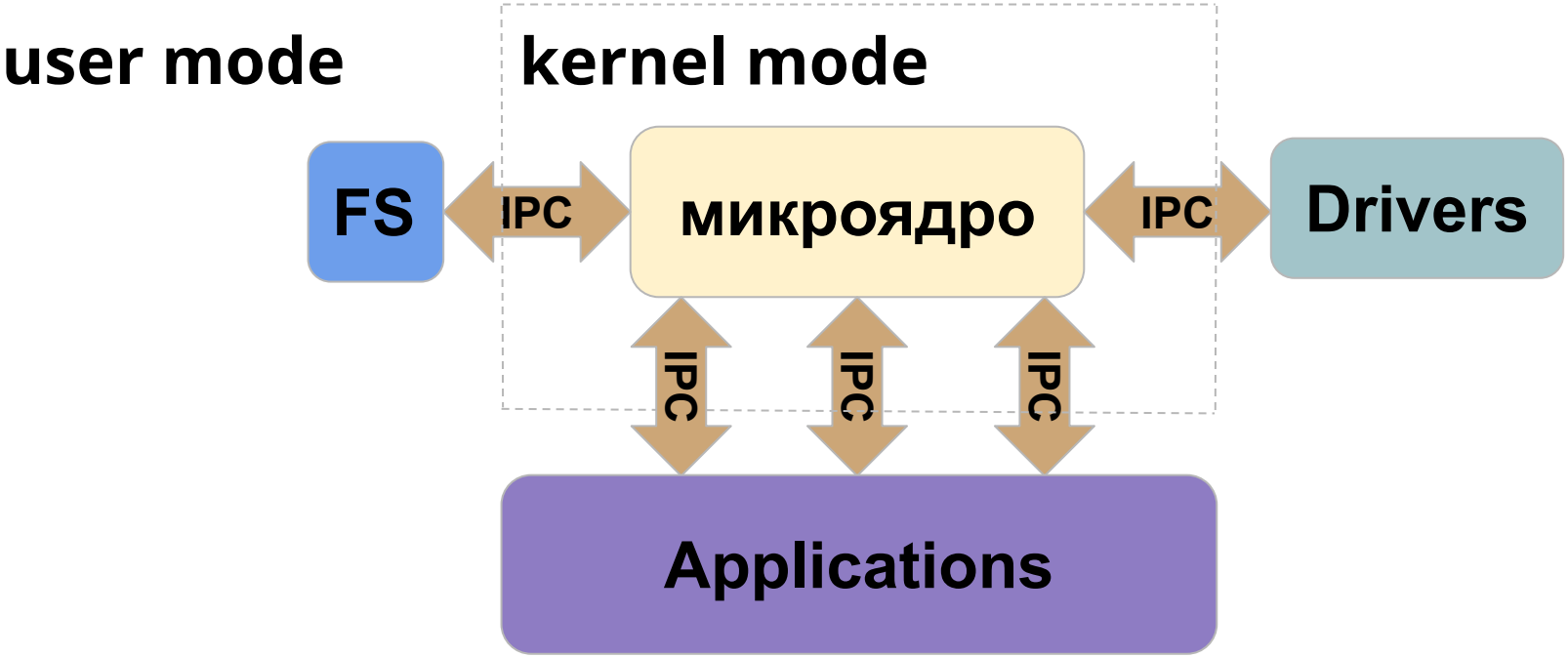
микроядро

capability-based модель безопасности

объём доверенного кода (trusted code base)


Monolith
Microkernel
Applications
system call
File system
IPC
Virtual memory
Drivers
Hardware
kernel mode
Drivers
FS
Applications
IPC
Virtual memory
Hardware

Микроядро


user mode
kernel mode
FS
IPC
микроядро
IPC
Drivers
IPC
IPC
IPC
Applications

Genode: доступные ядра

nova

sel4

L4/Fiasco

base-hw

linux

Genode: структура

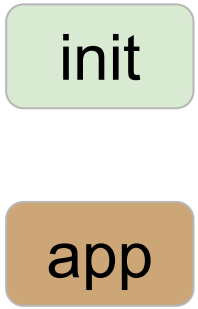
компонент — базовый строительный блок

Genode: структура

компонент — базовый строительный блок

Genode: структура

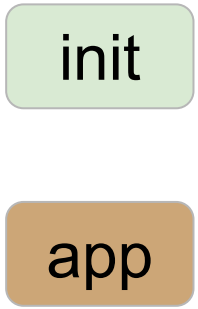
компонент — базовый строительный блок



Genode: структура

компонент — базовый строительный блок

отношение “родитель — потомок”

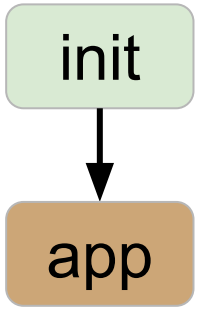


Genode: структура

компонент — базовый строительный блок

отношение “родитель — потомок”

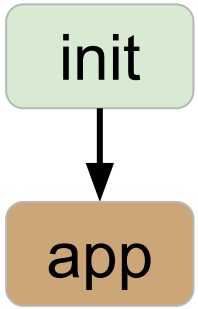
дерево компонентов



Genode: структура

компонент — базовый строительный блок

отношение “родитель — потомок”



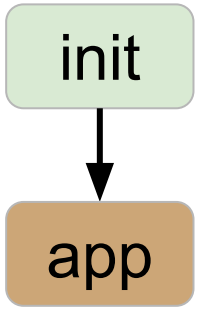
Genode: структура

компонент — базовый строительный блок

отношение “родитель — потомок”

дерево компонентов

сервисы и сессии



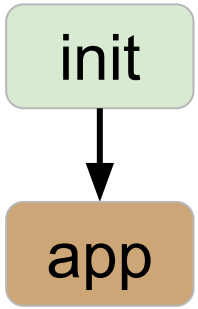
Genode: структура

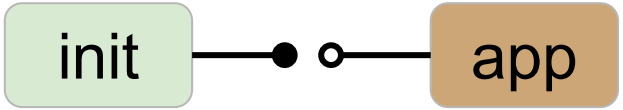
компонент — базовый строительный блок

отношение “родитель — потомок”

дерево компонентов

сервисы и сессии





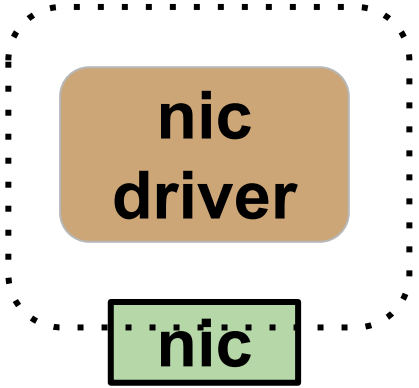
Genode: компоненты

у каждого компонента свой protection domain

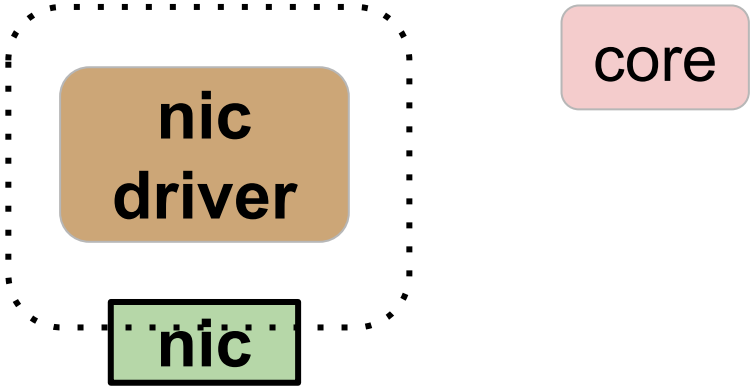
компоненты могут предоставлять сервисы

компоненты взаимодействуют через IPC

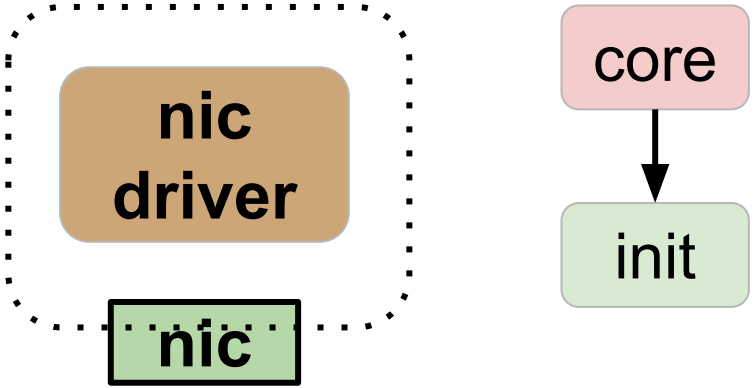
Genode: сервисы



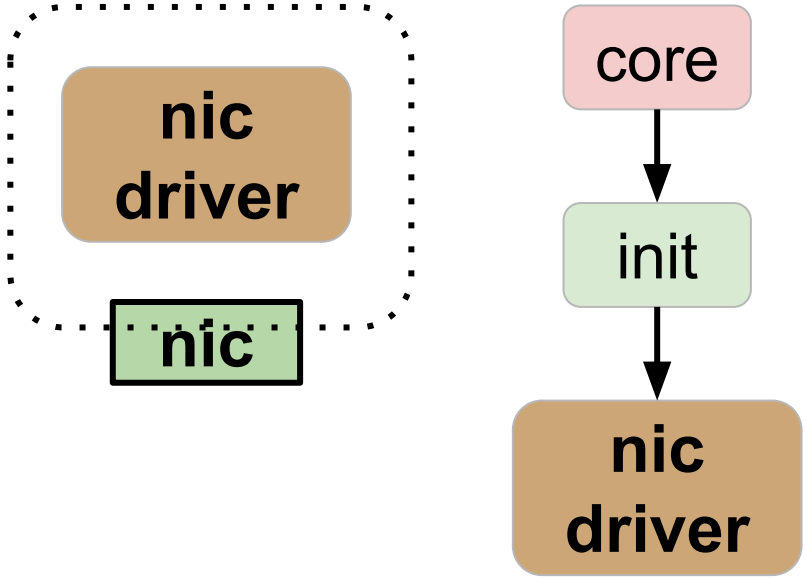
Genode: сервисы



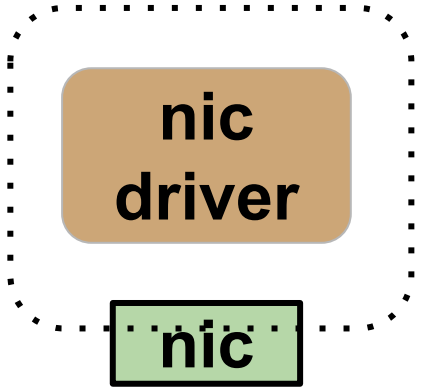
Genode: сервисы

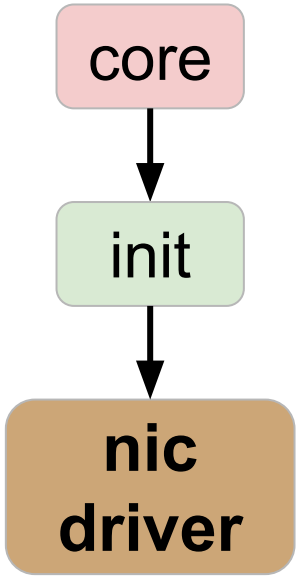


Genode: сервисы

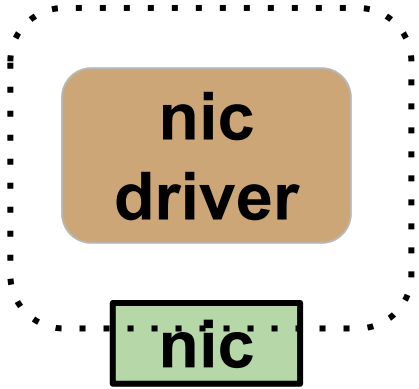


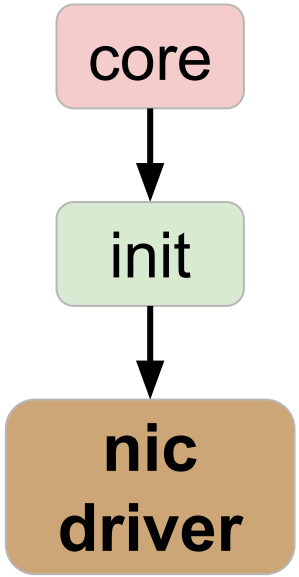
Genode: сервисы



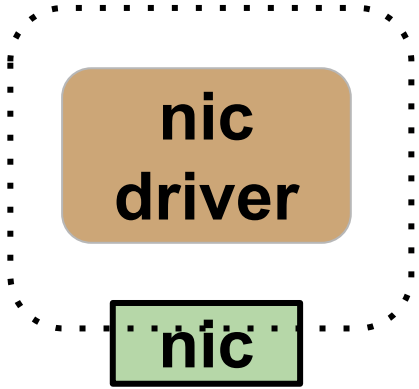


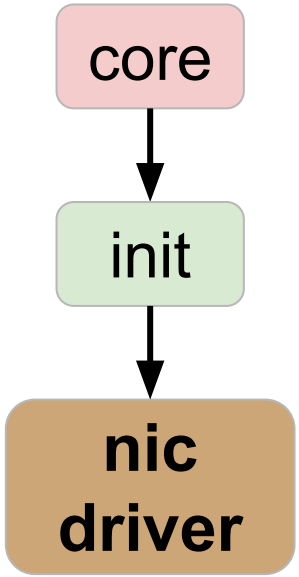
Genode: сервисы

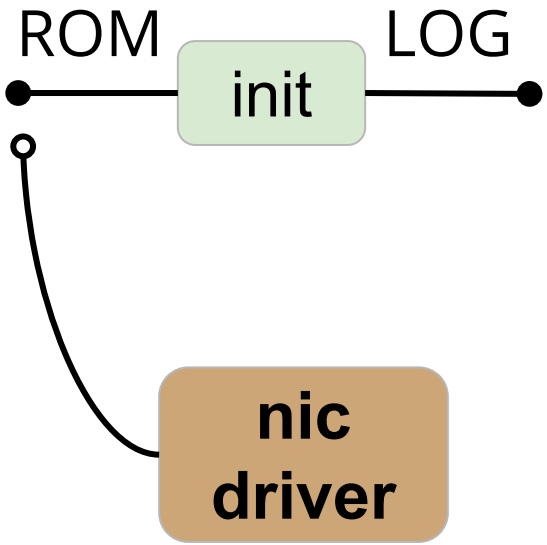




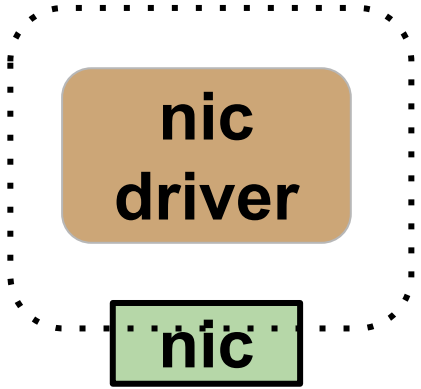
Genode: сервисы

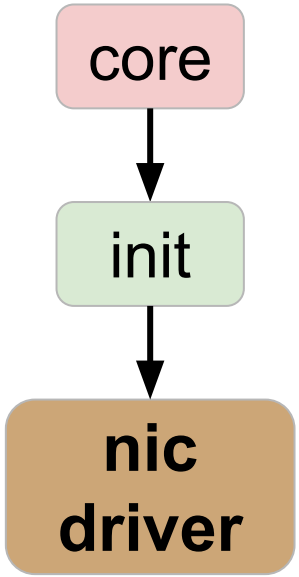


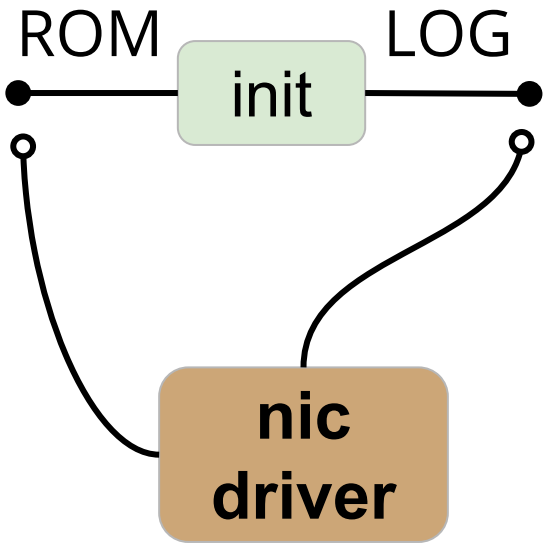




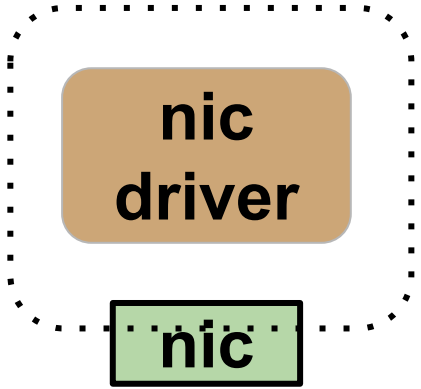
Genode: сервисы

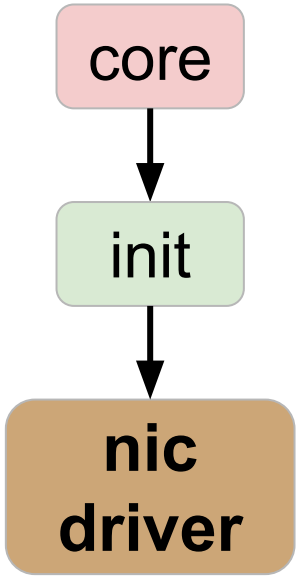


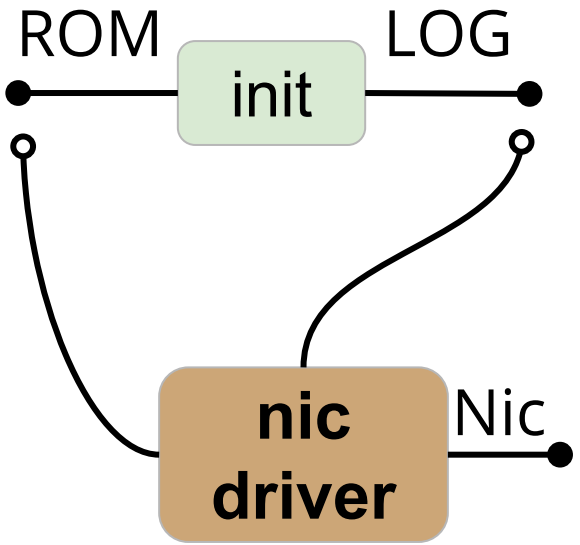




Genode: сервисы







MQTT

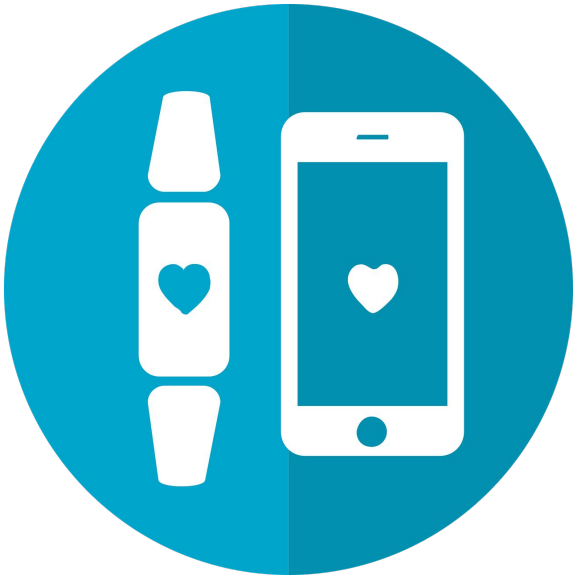
Message Queue Telemetry Transport

- queue
- lightweight
- publish-subscribe
- IoT

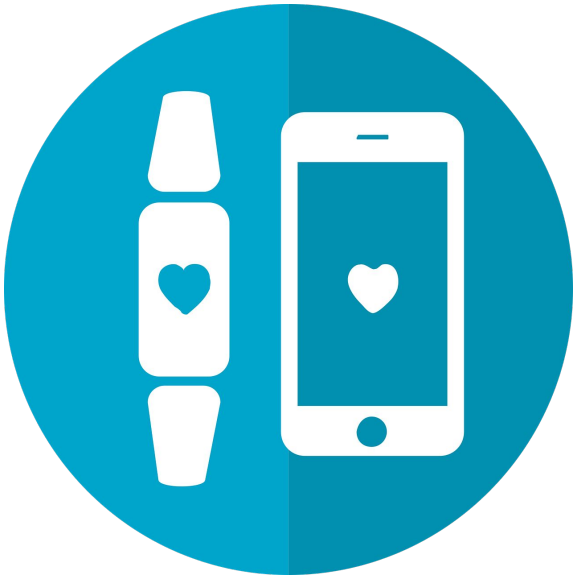
MQTT

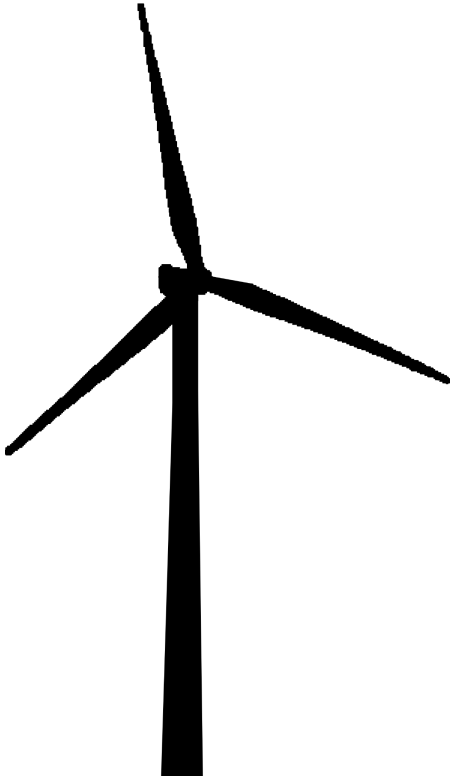
MQTT

MQTT

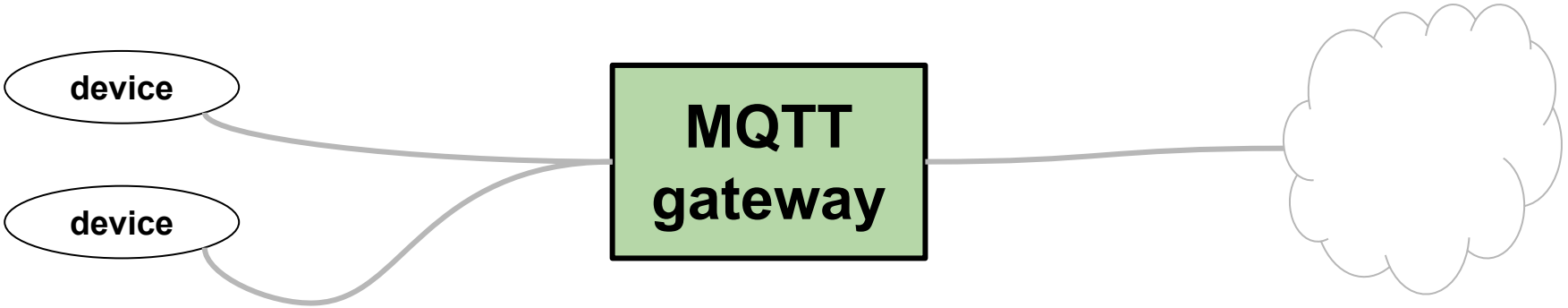


MQTT

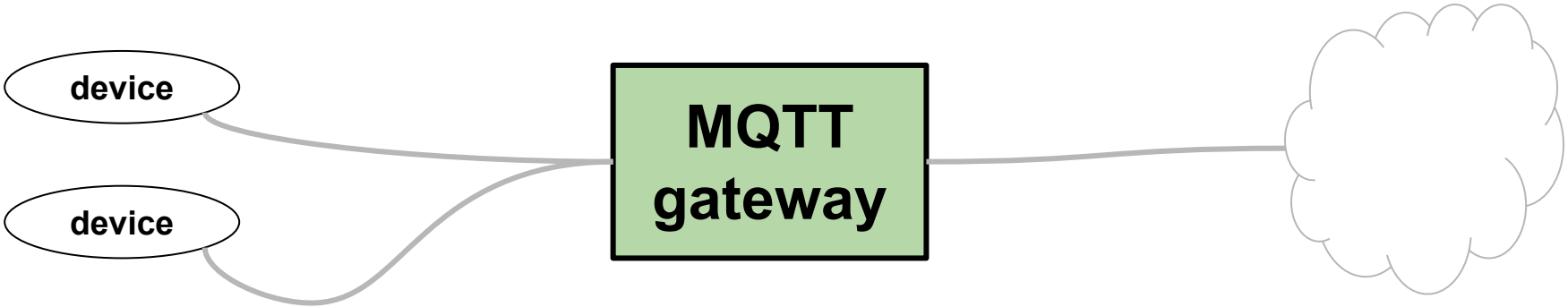




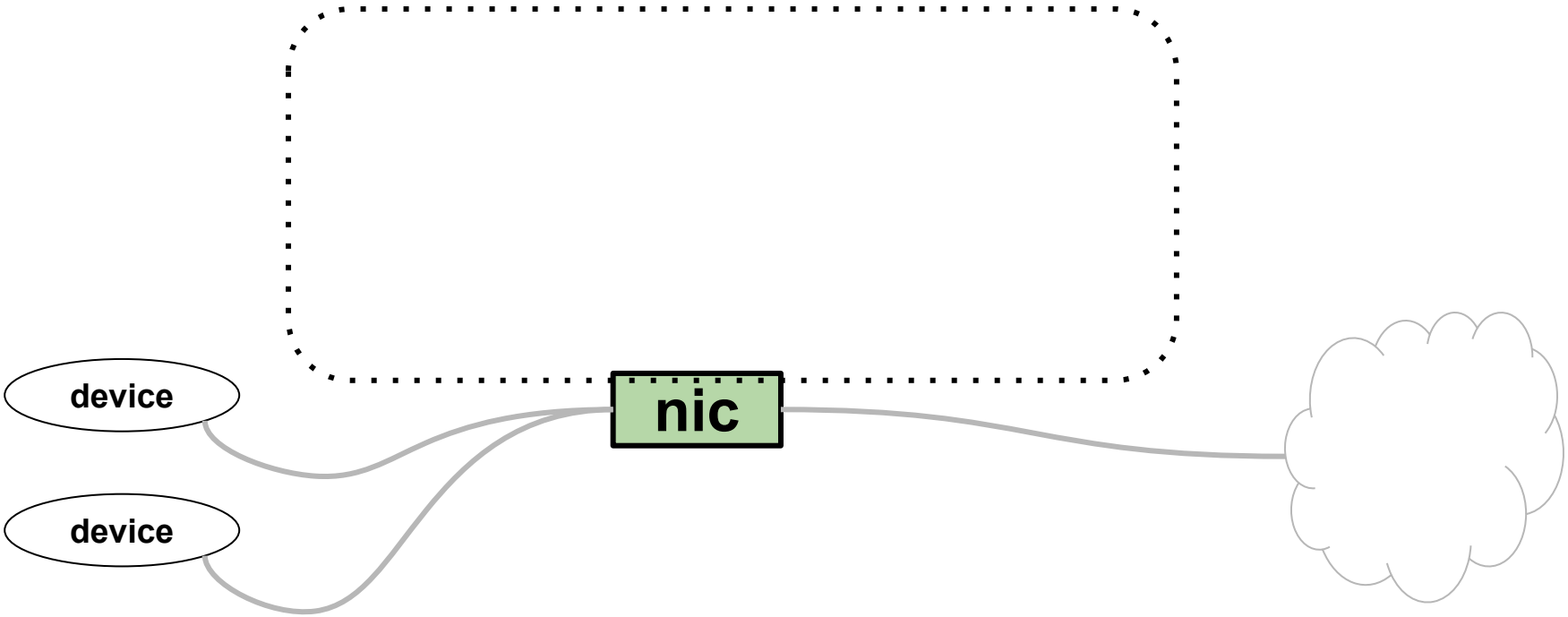
MQTT gateway



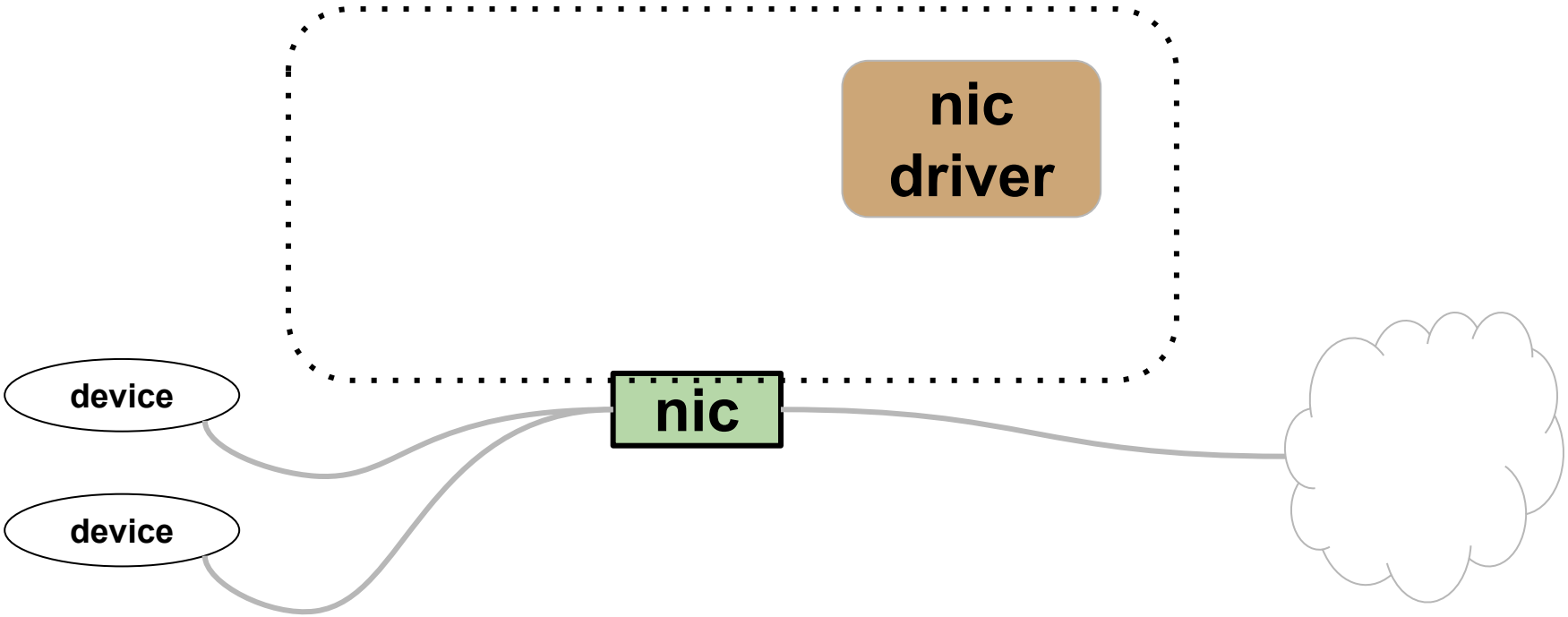
MQTT gateway



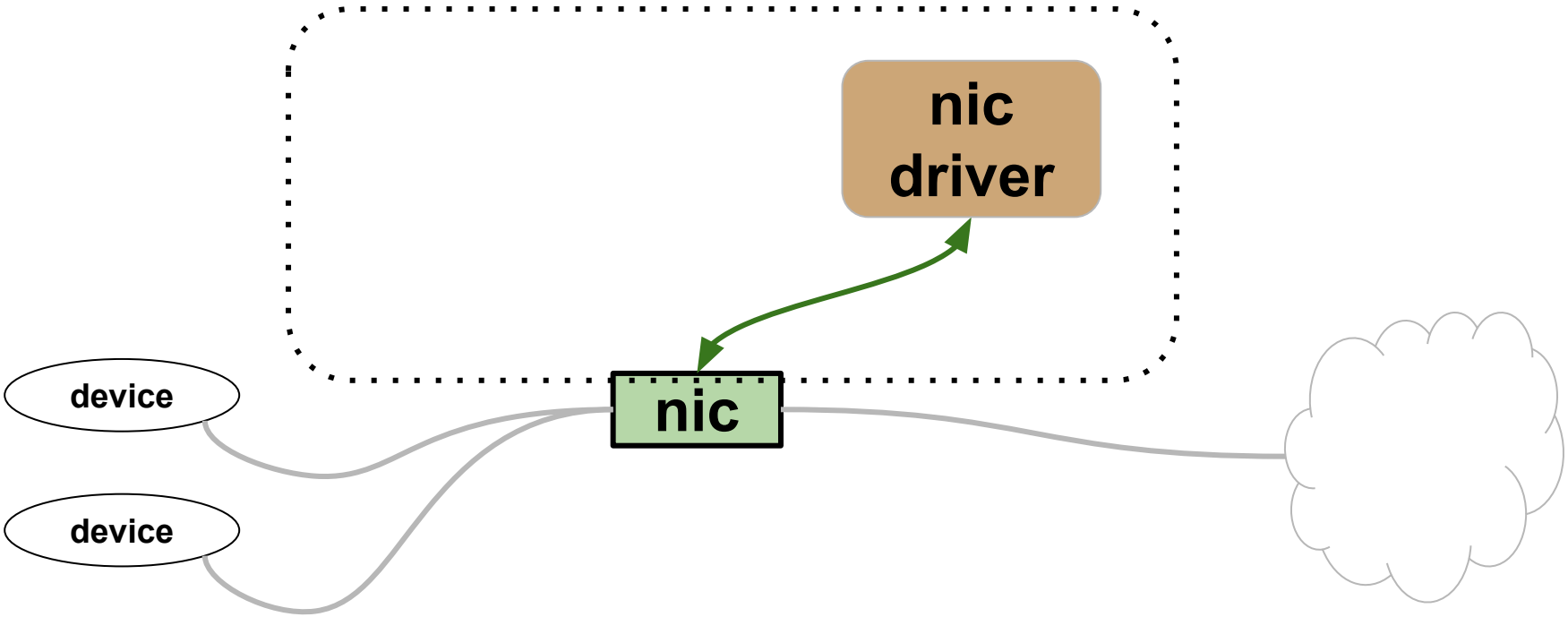
MQTT gateway



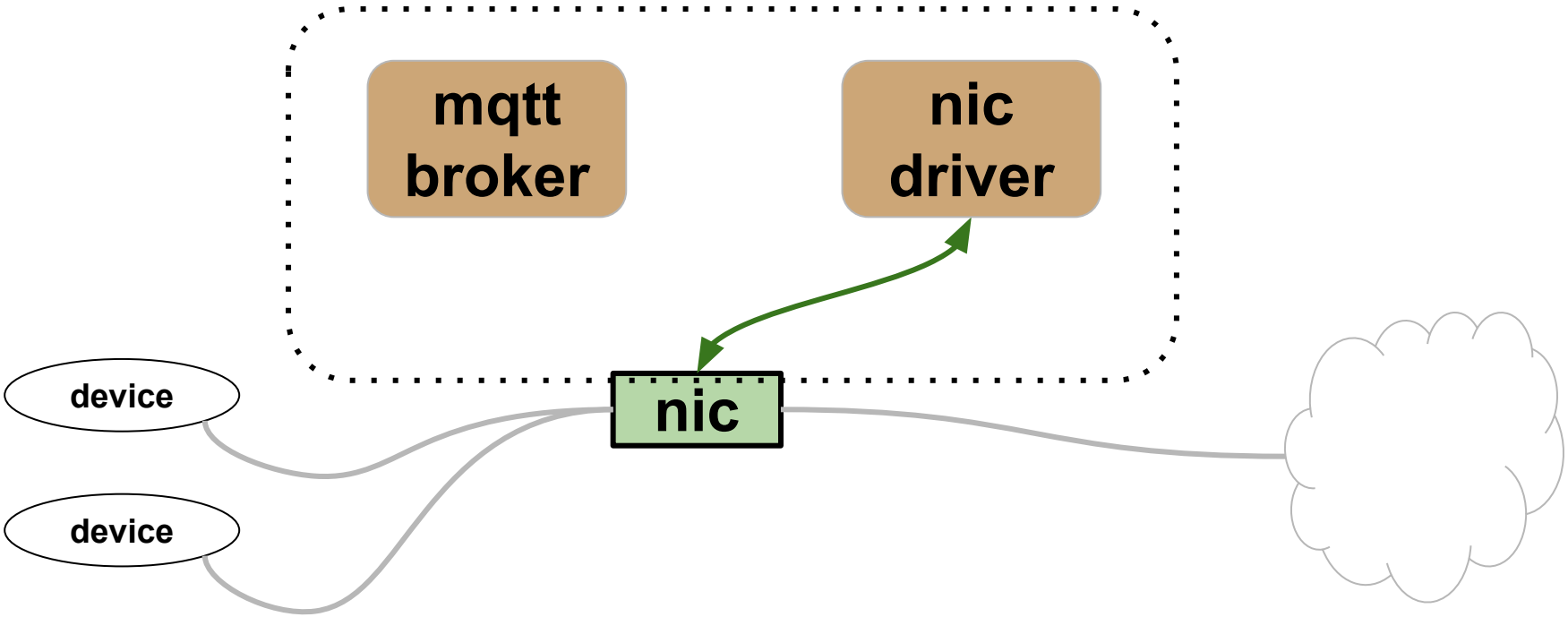
MQTT gateway



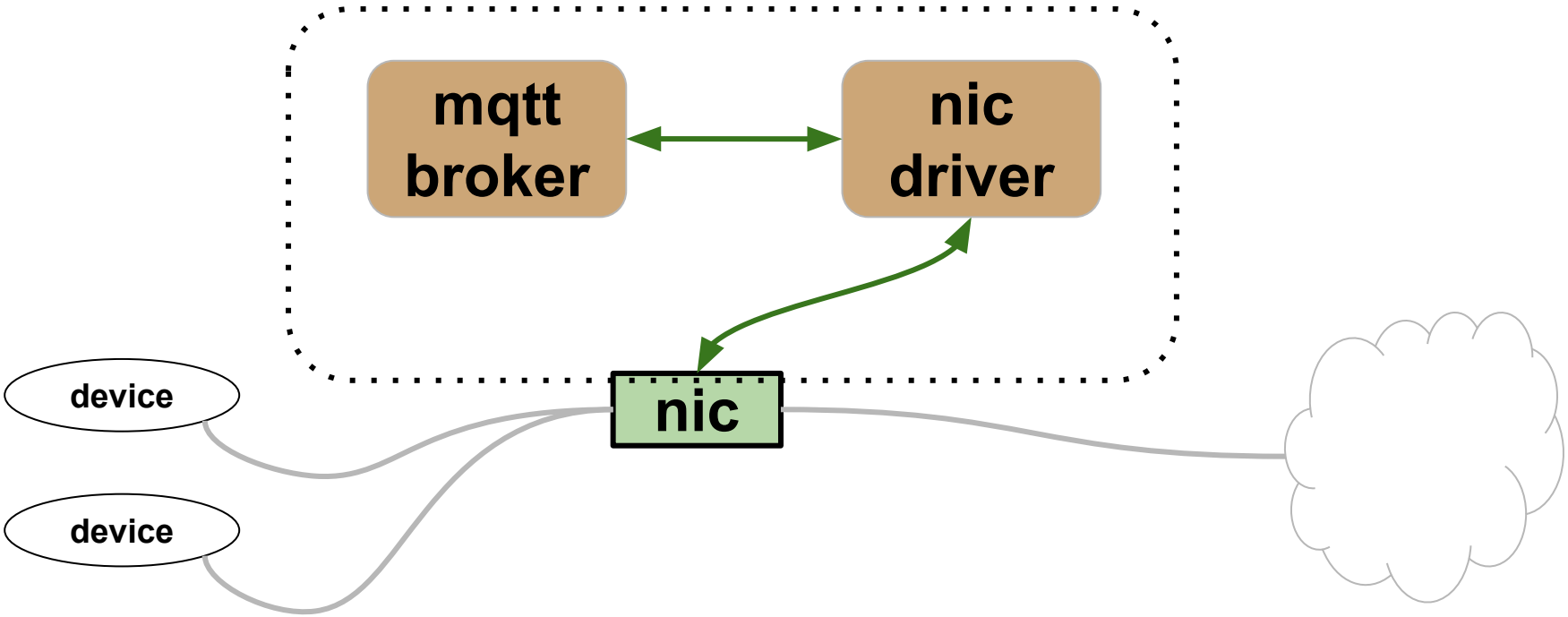
MQTT gateway



MQTT gateway



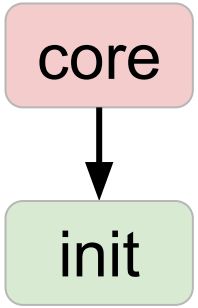
MQTT gateway



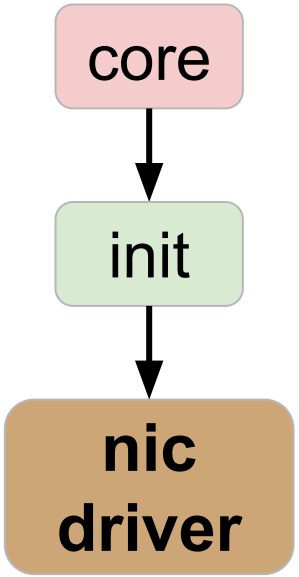
Genode: сервисы

core

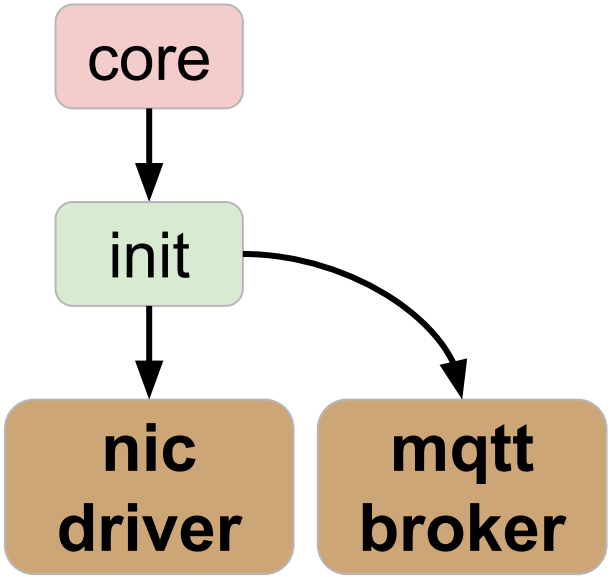
Genode: сервисы



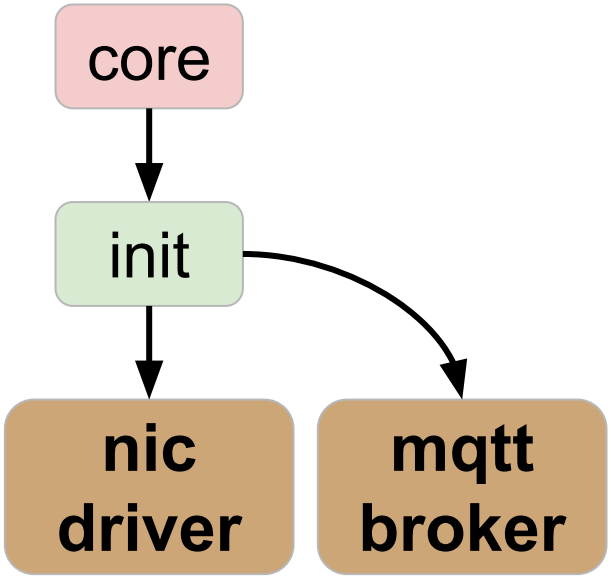
Genode: сервисы



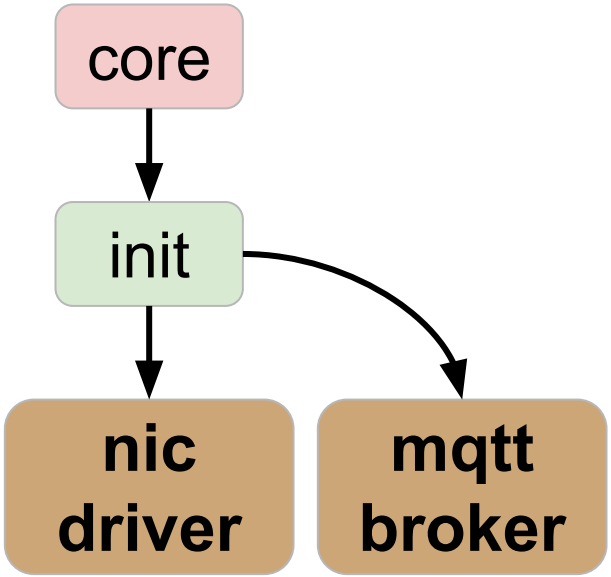
Genode: сервисы

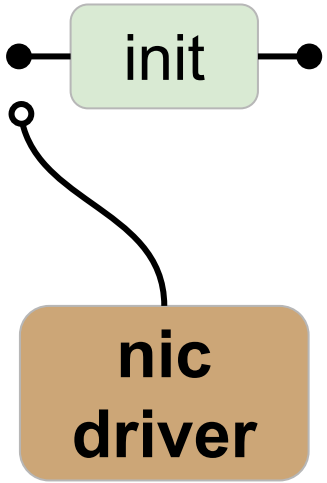


Genode: сервисы

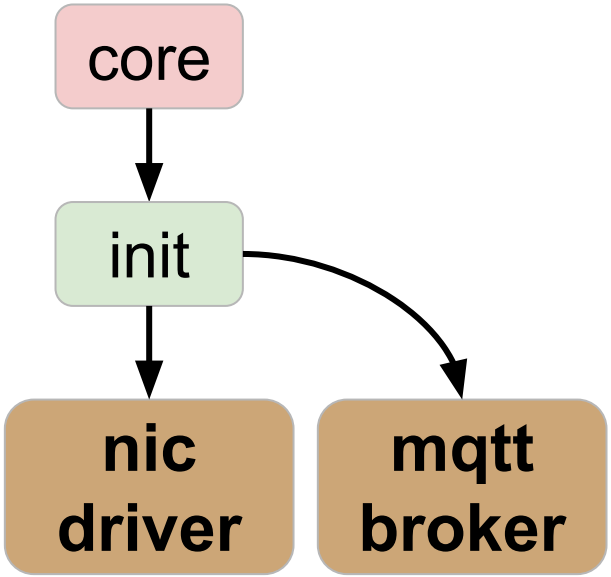


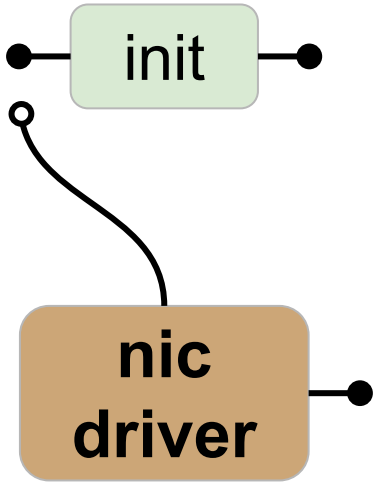
Genode: сервисы



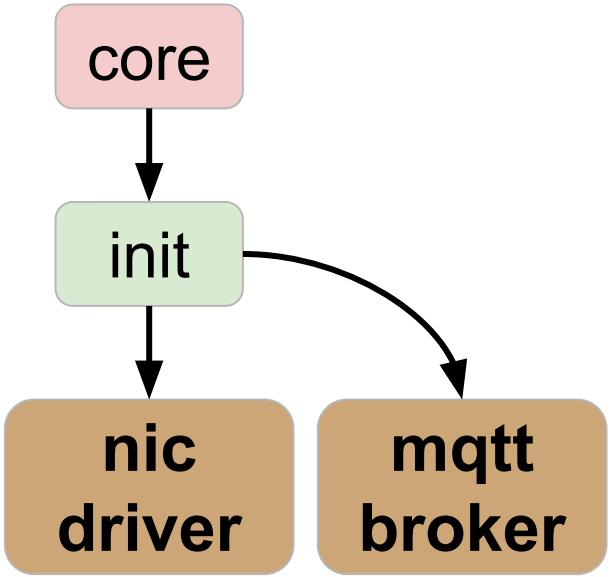


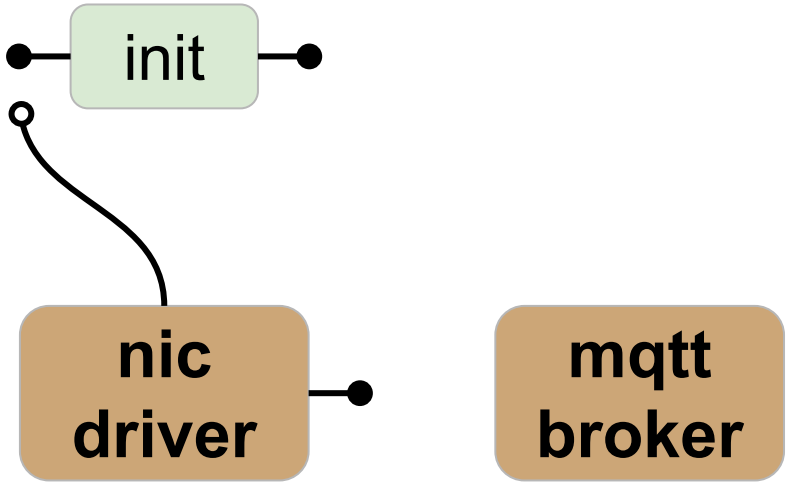
Genode: сервисы



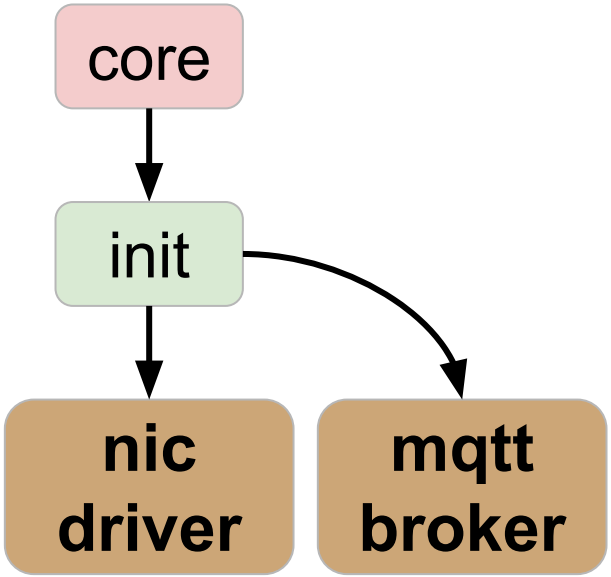


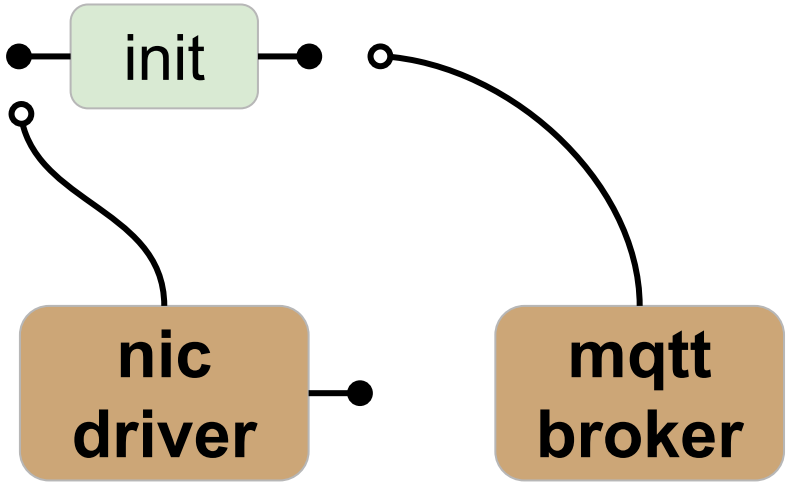
Genode: сервисы



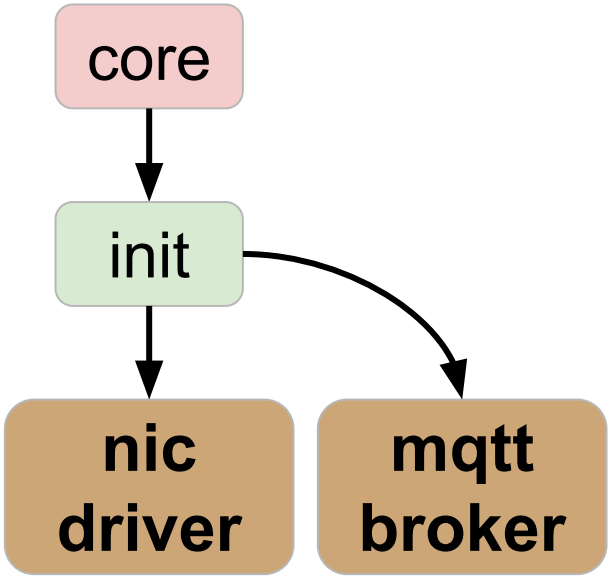


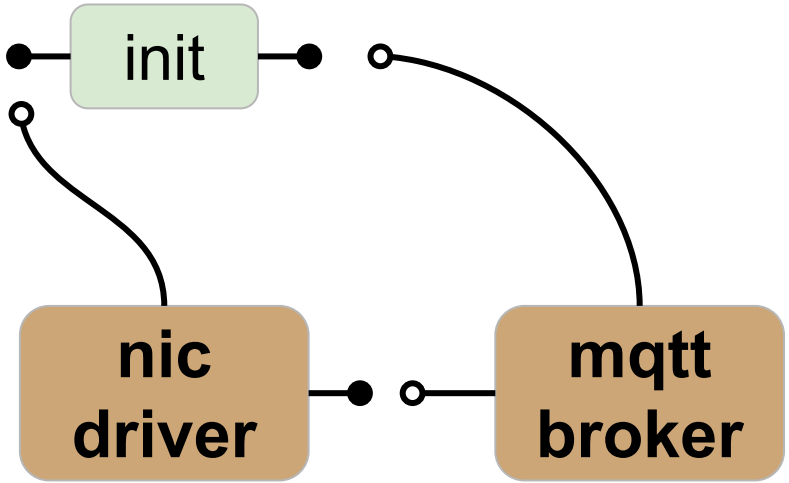
Genode: сервисы



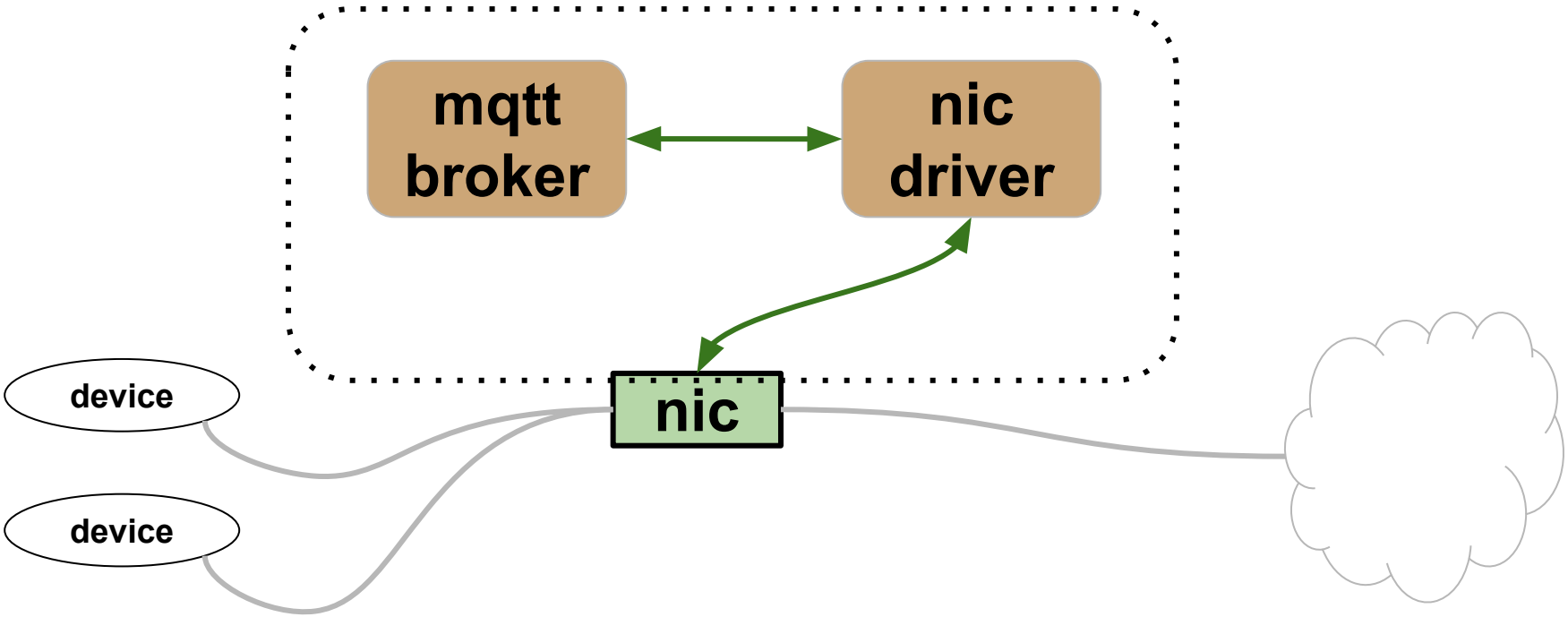


Genode: сервисы

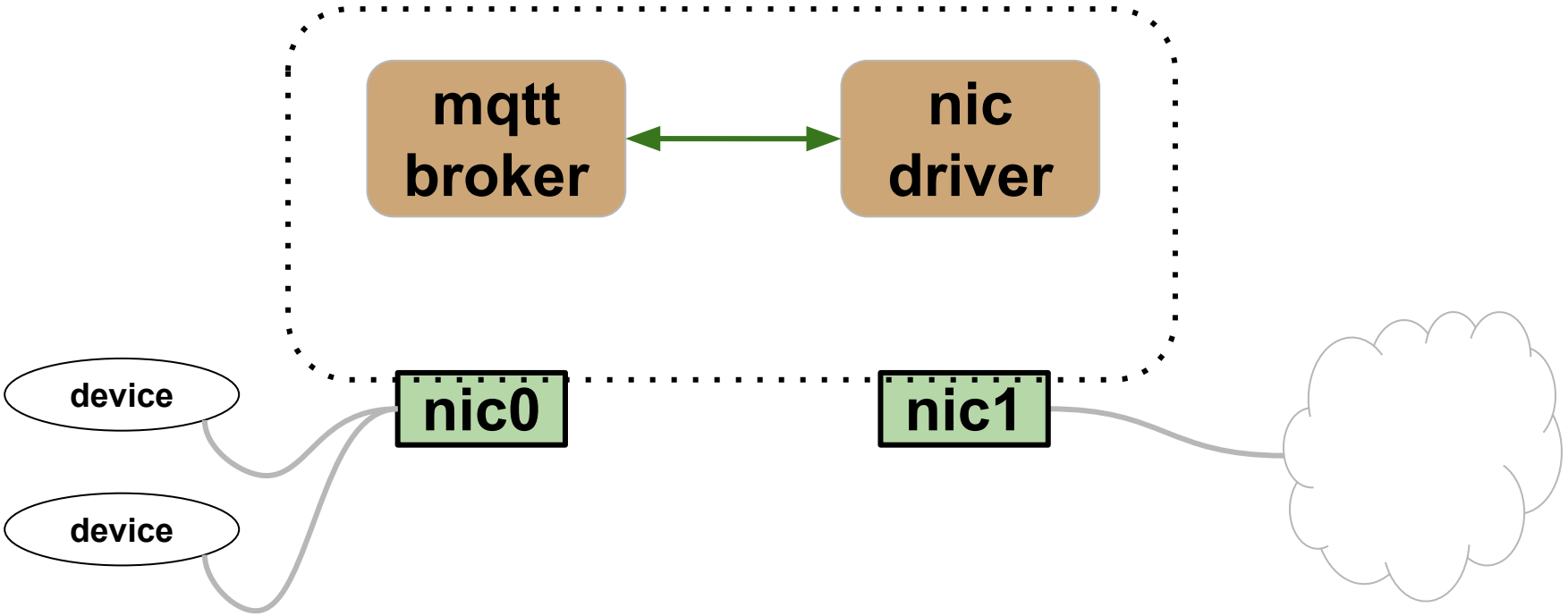




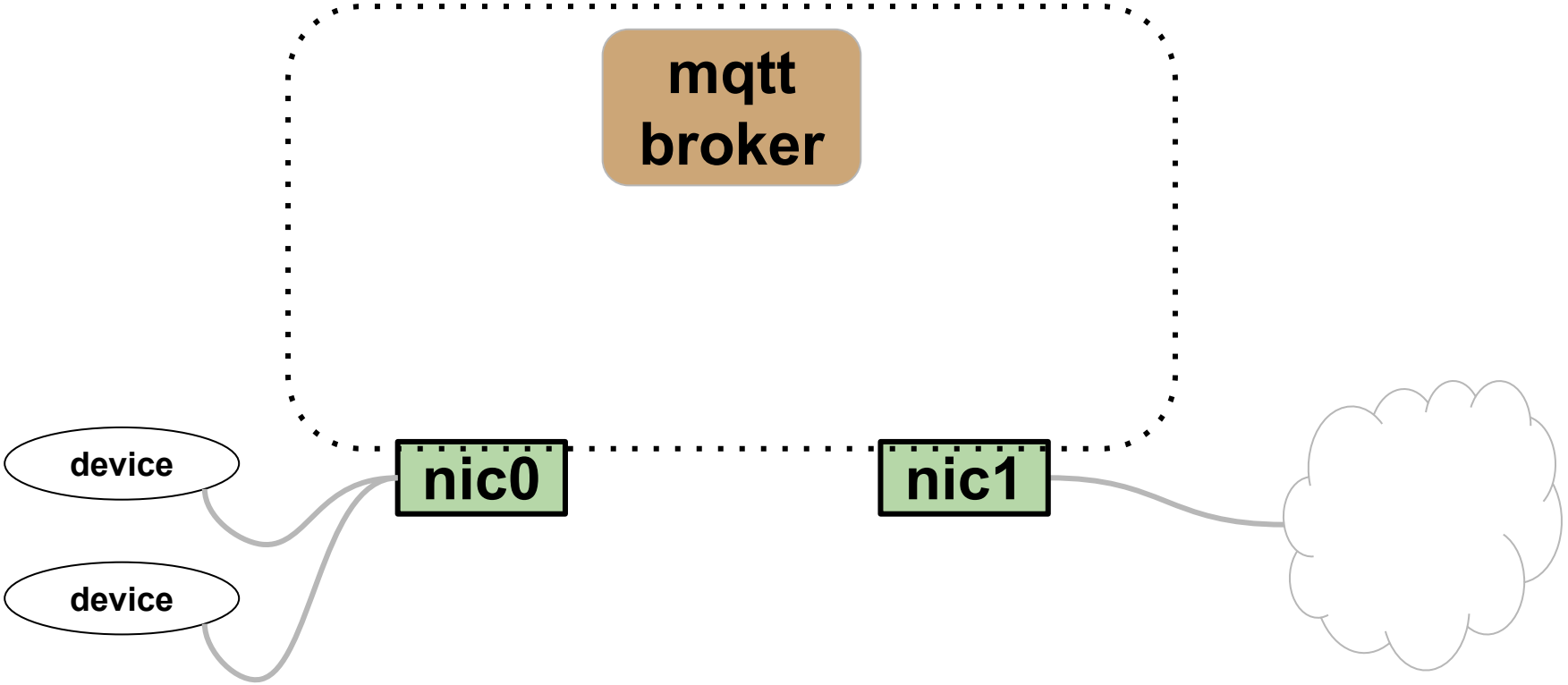
MQTT gateway



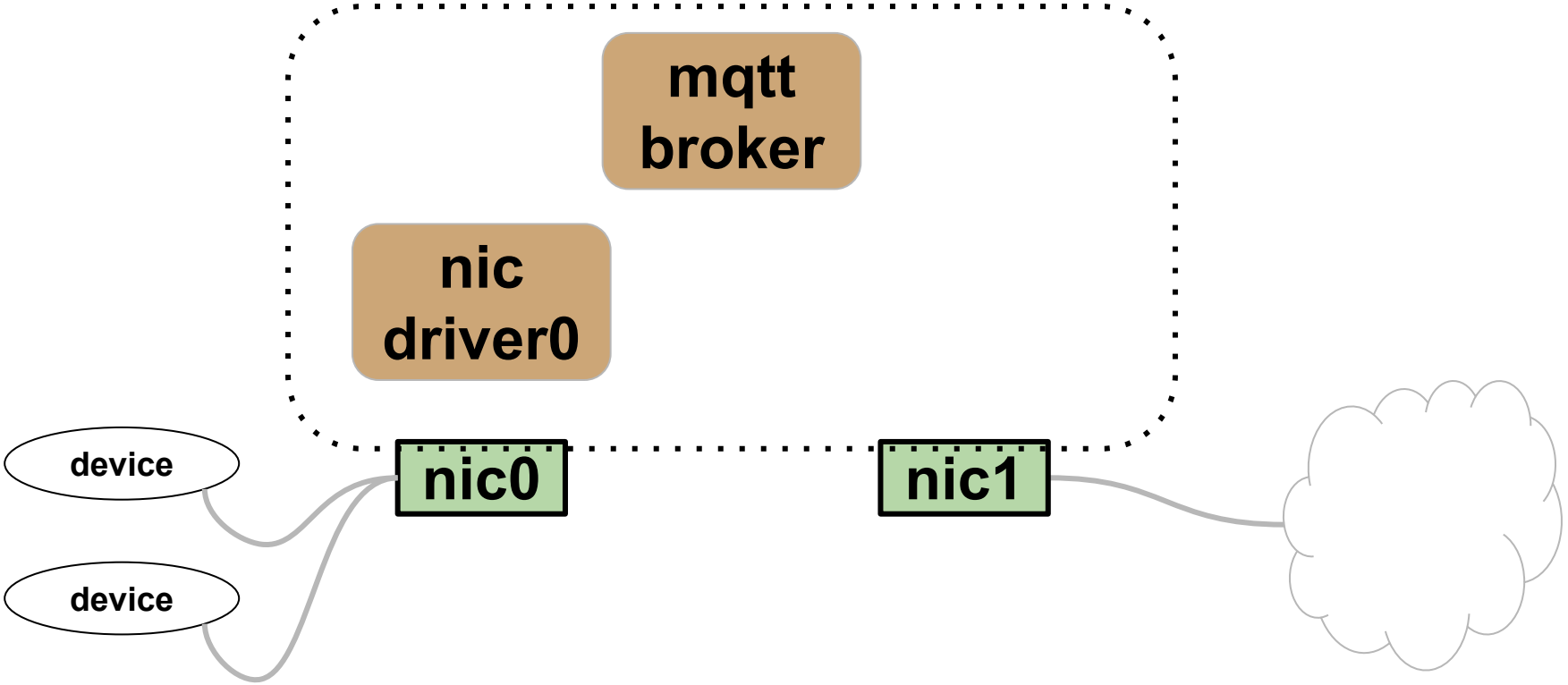
MQTT gateway



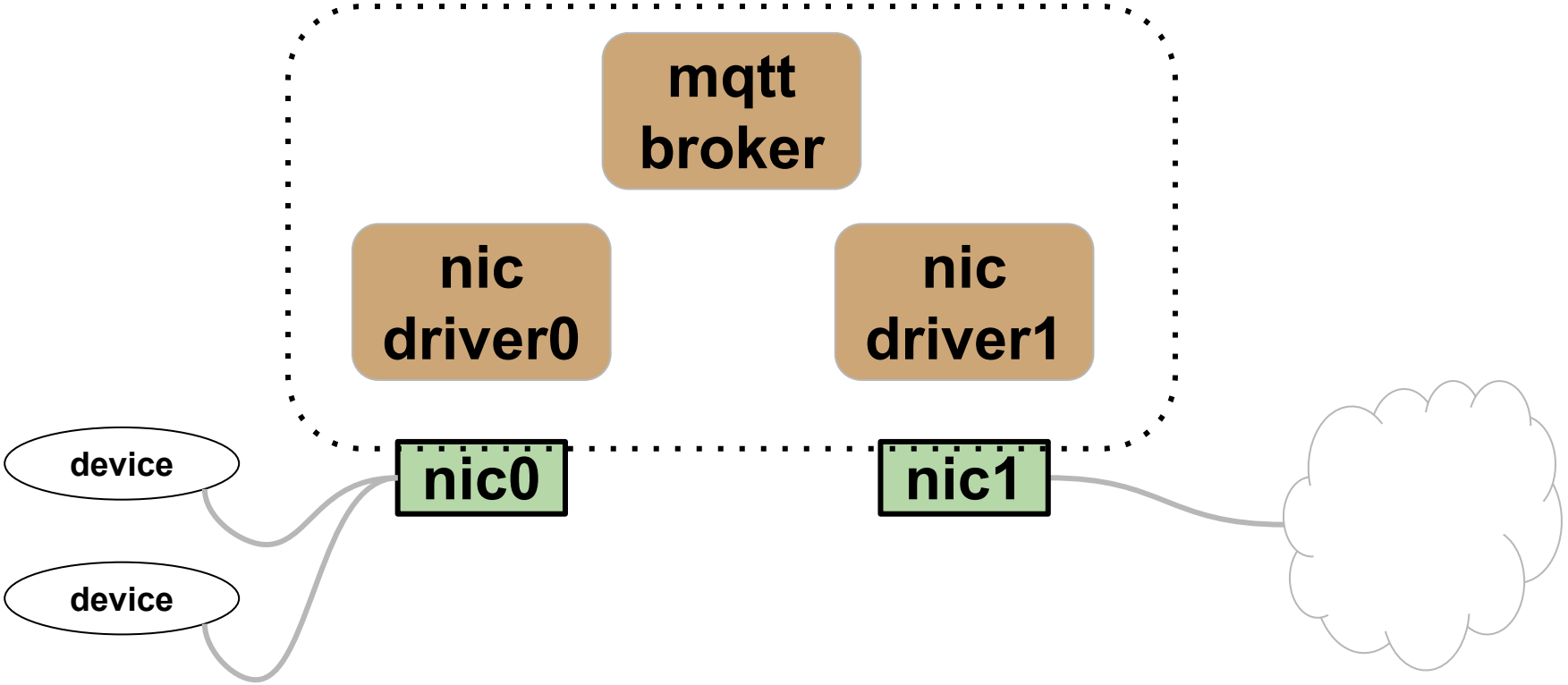
MQTT gateway



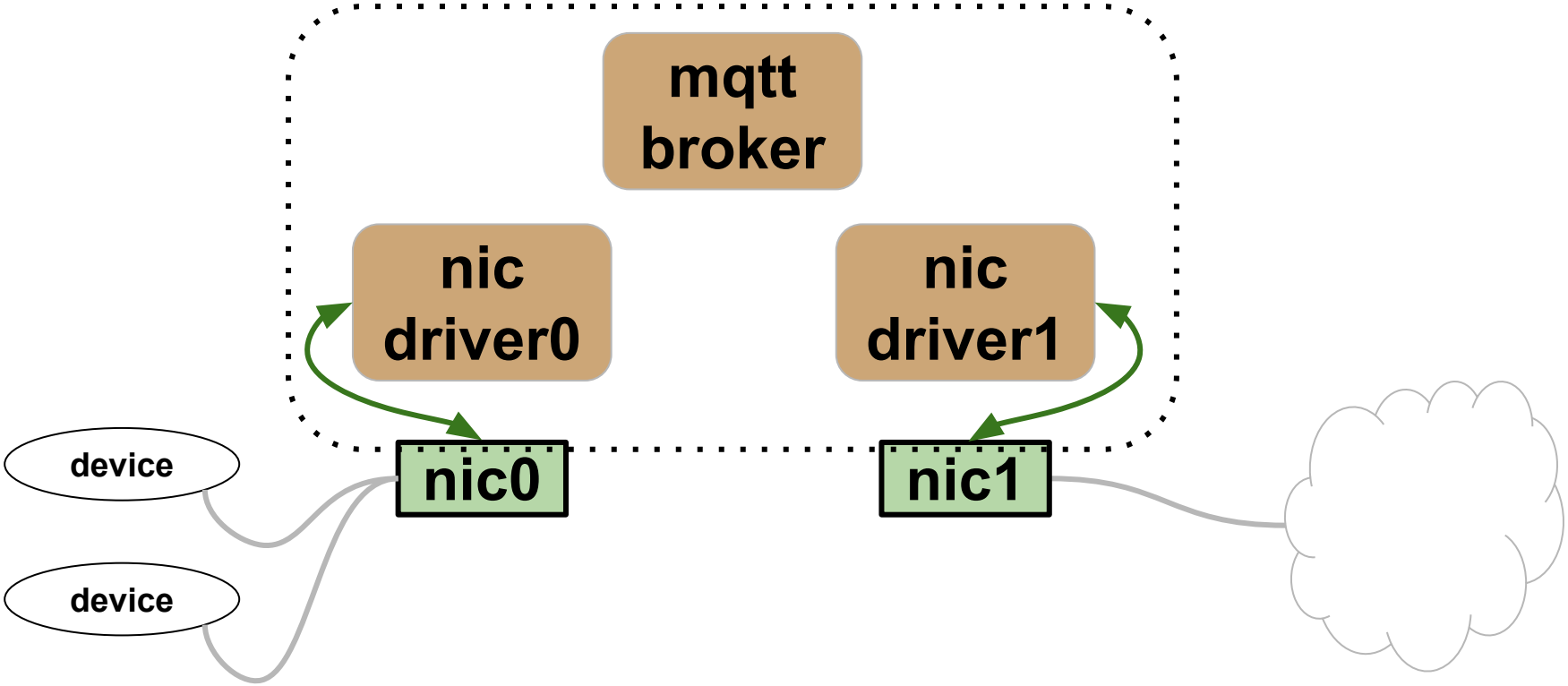
MQTT gateway



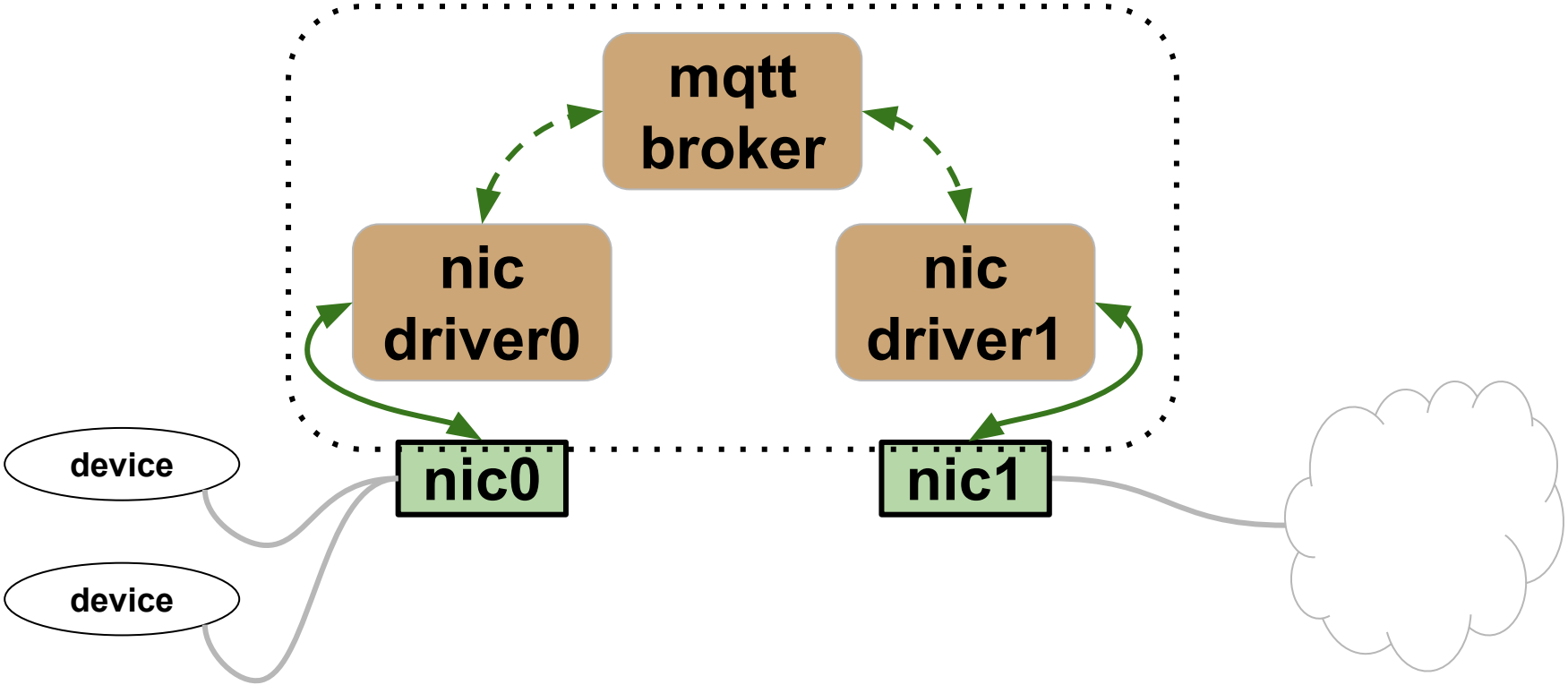
MQTT gateway



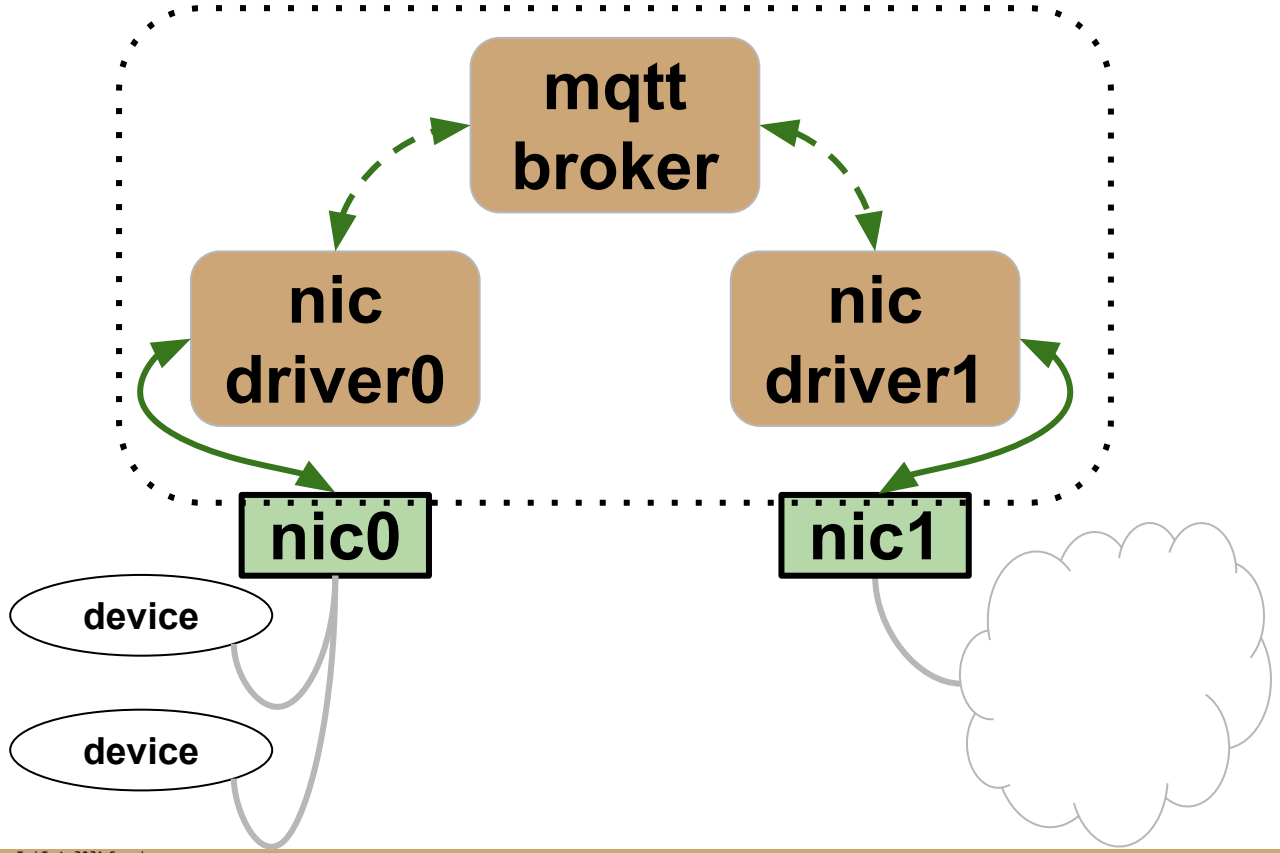
MQTT gateway



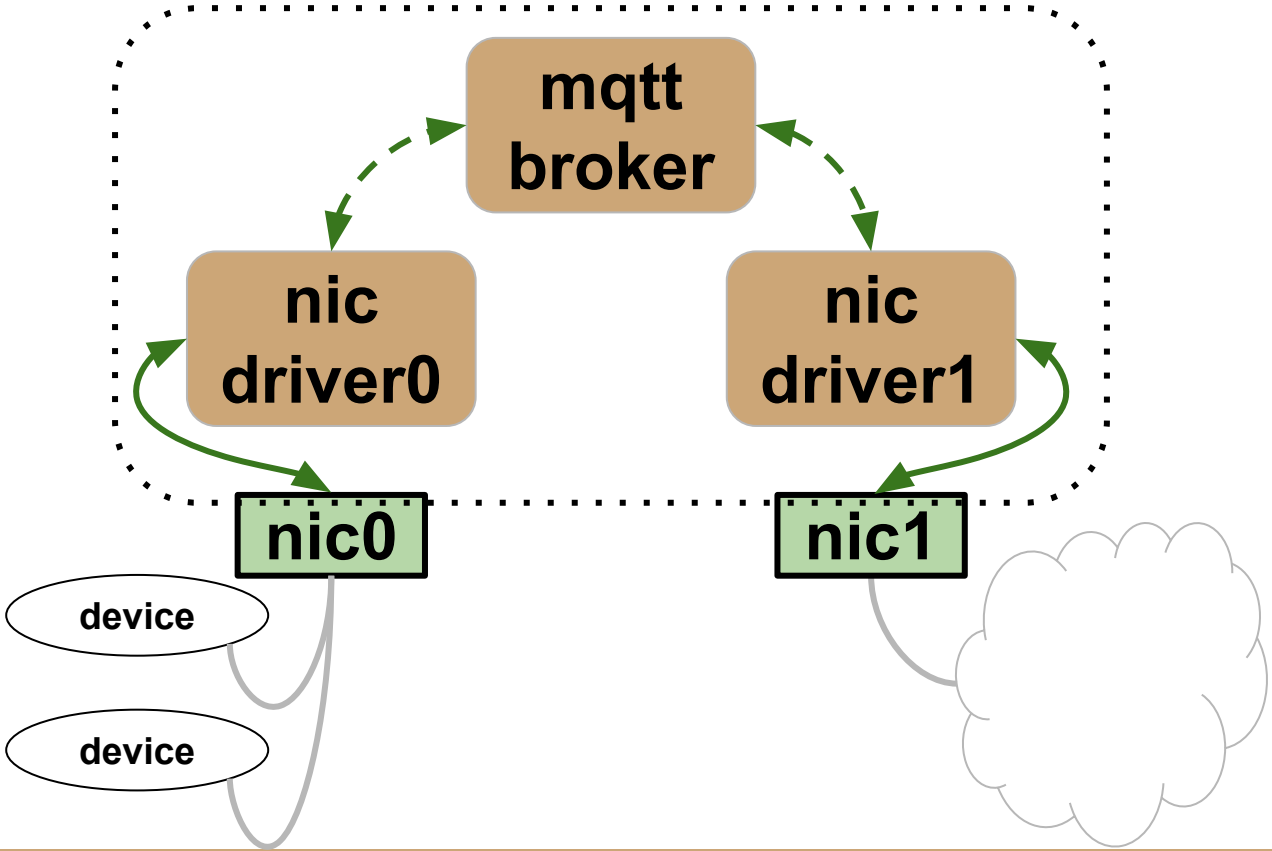
MQTT gateway



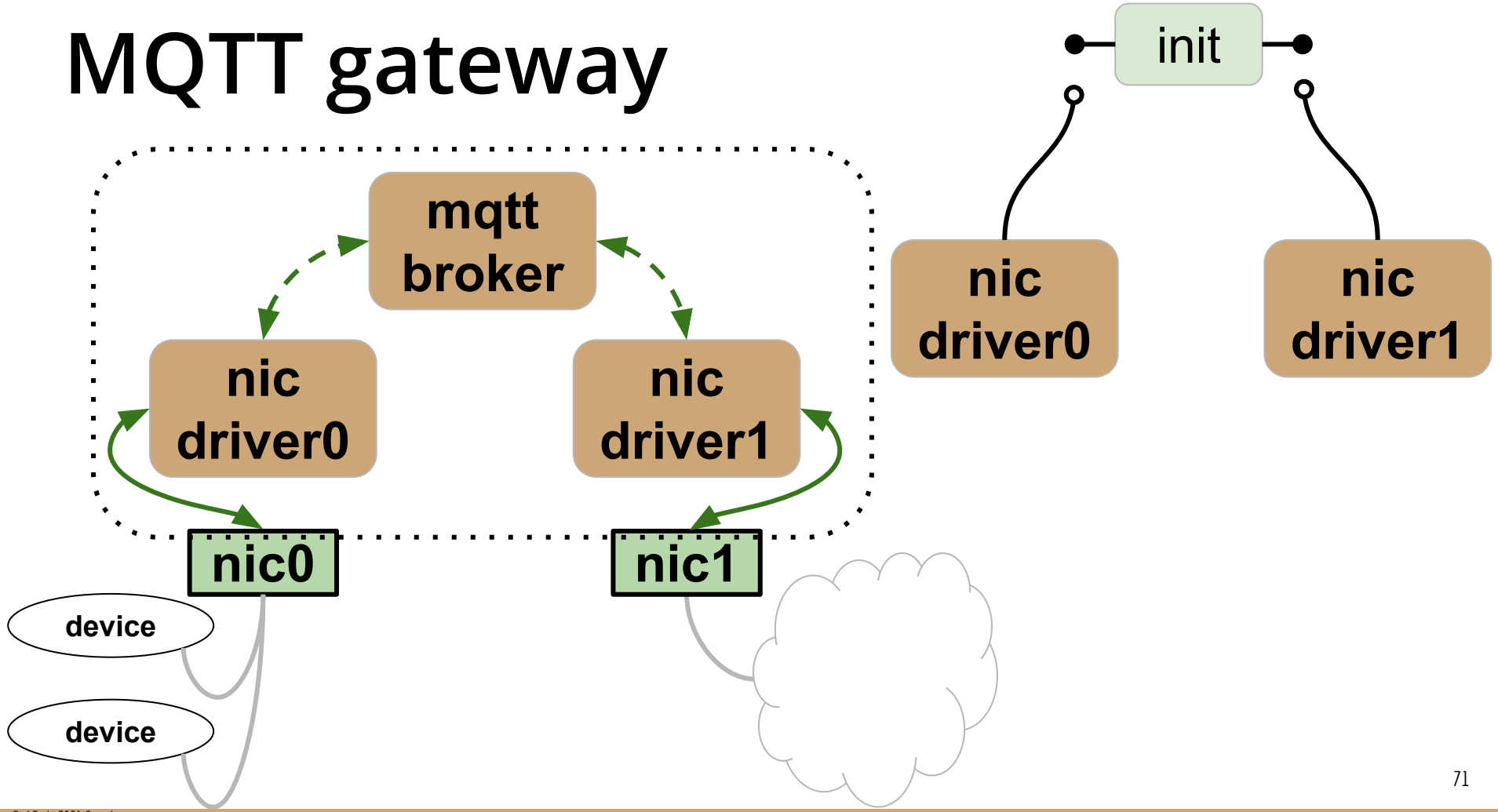
MQTT gateway



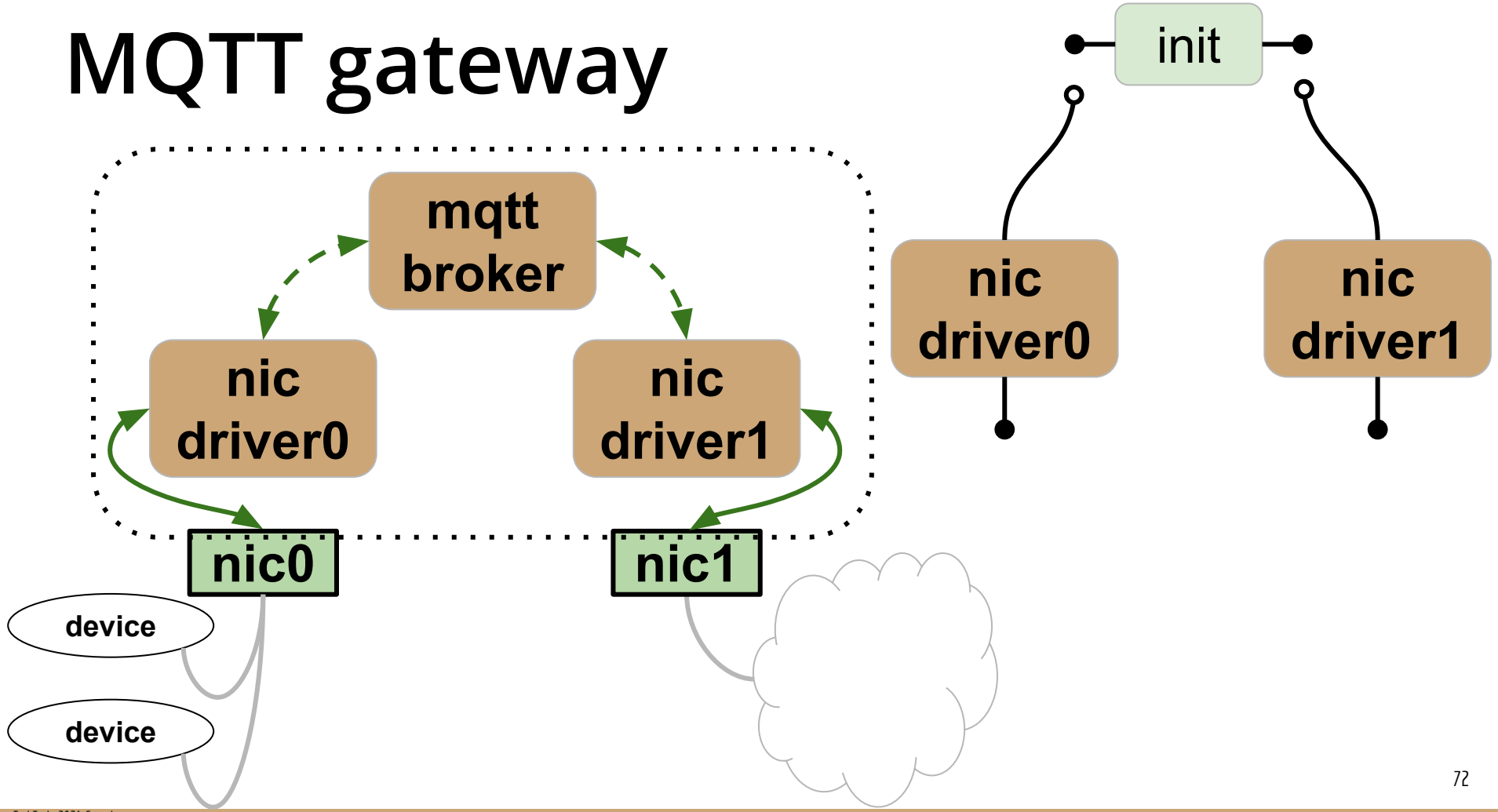
MQTT gateway



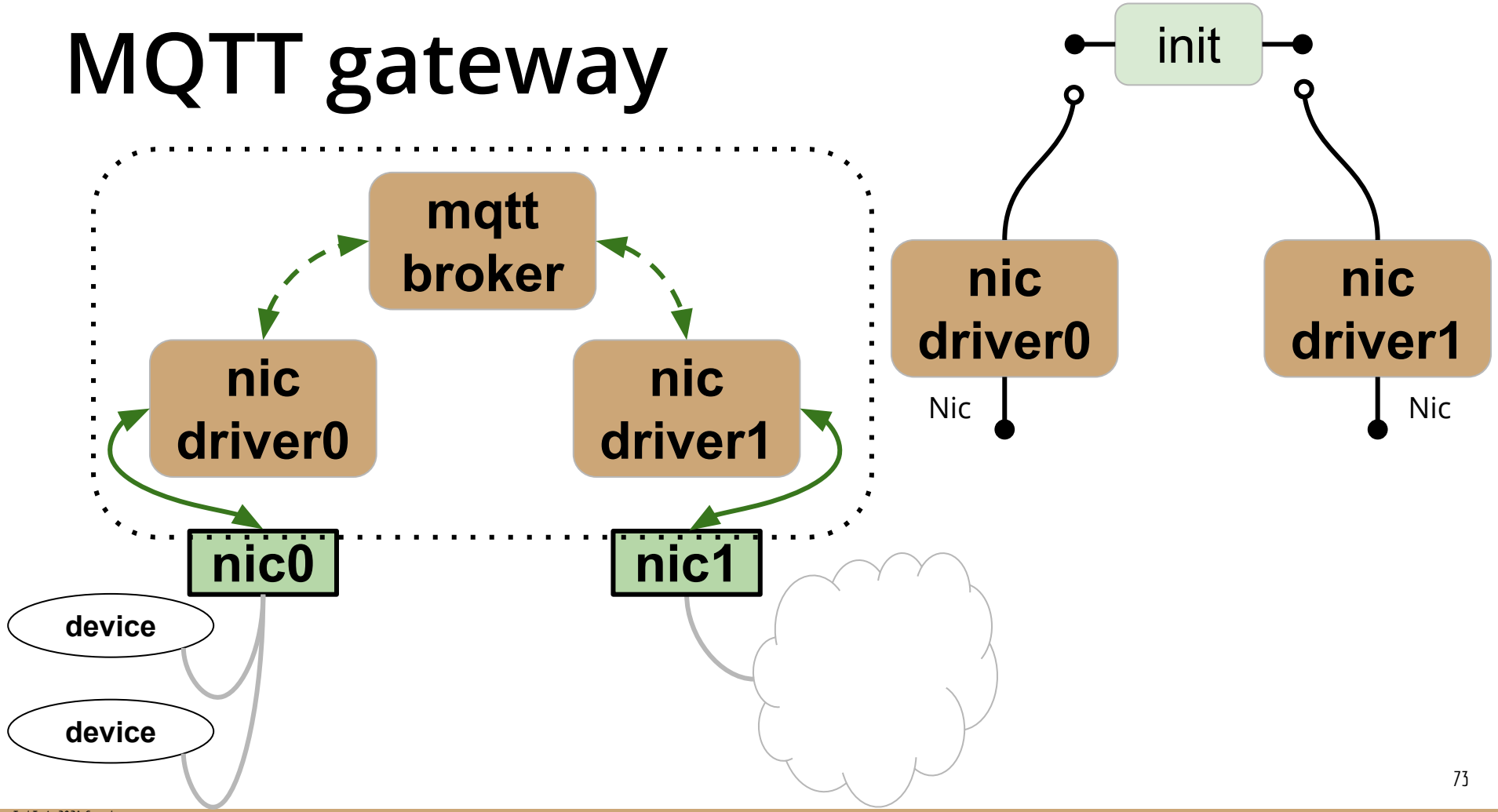
MQTT gateway



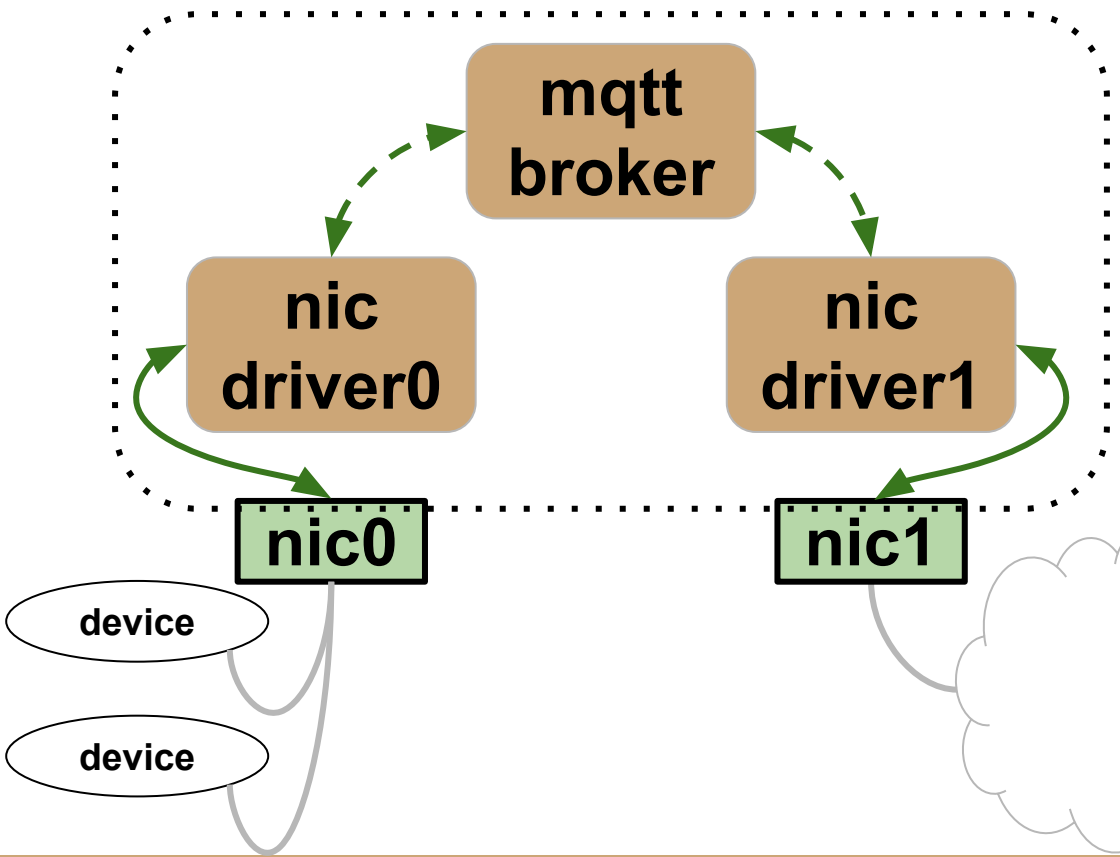
MQTT gateway

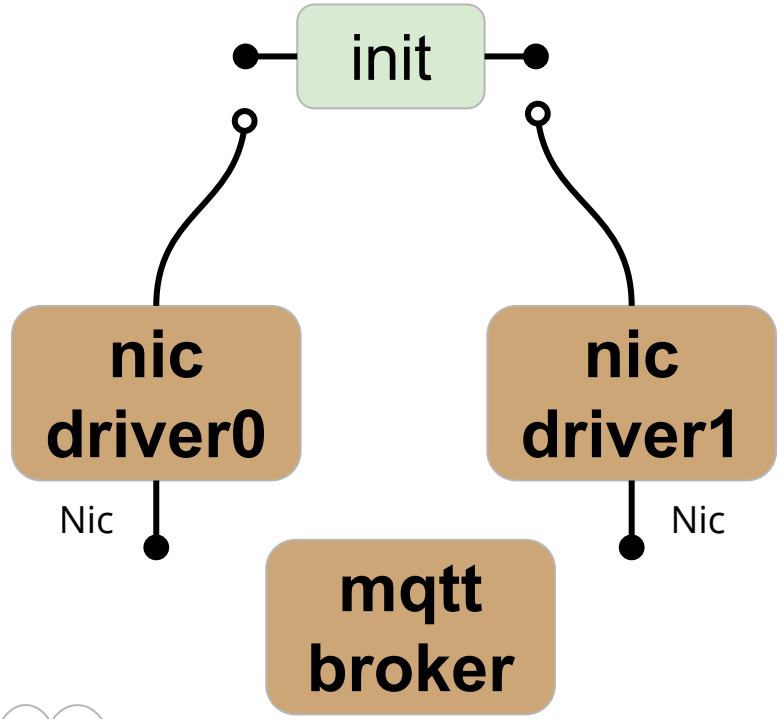


MQTT gateway

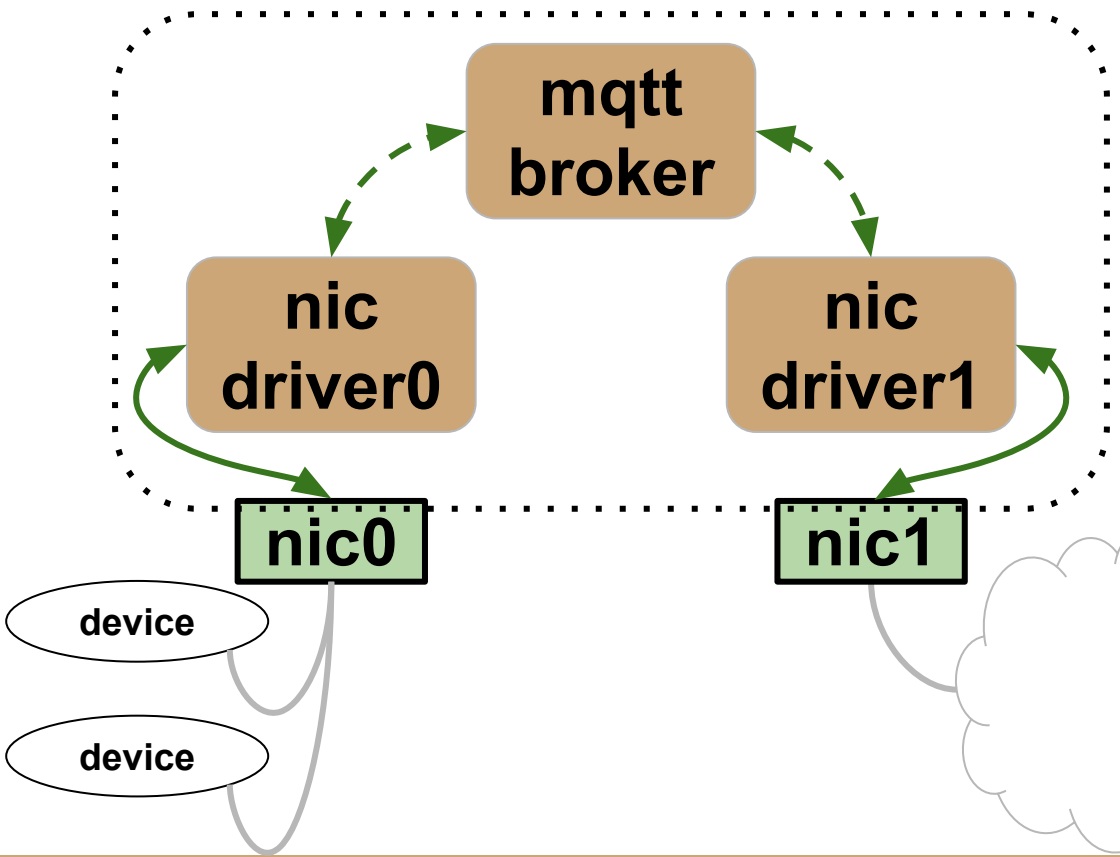


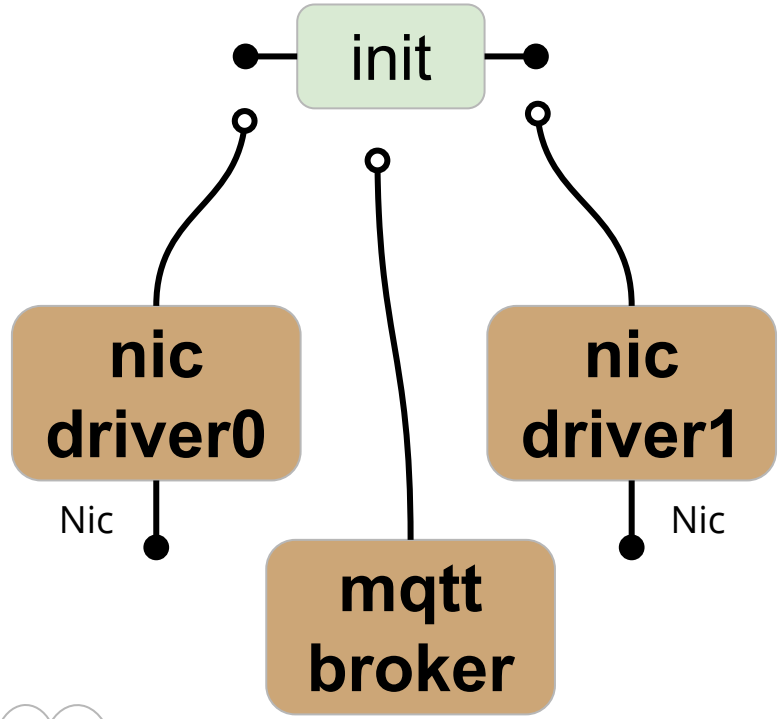
MQTT gateway



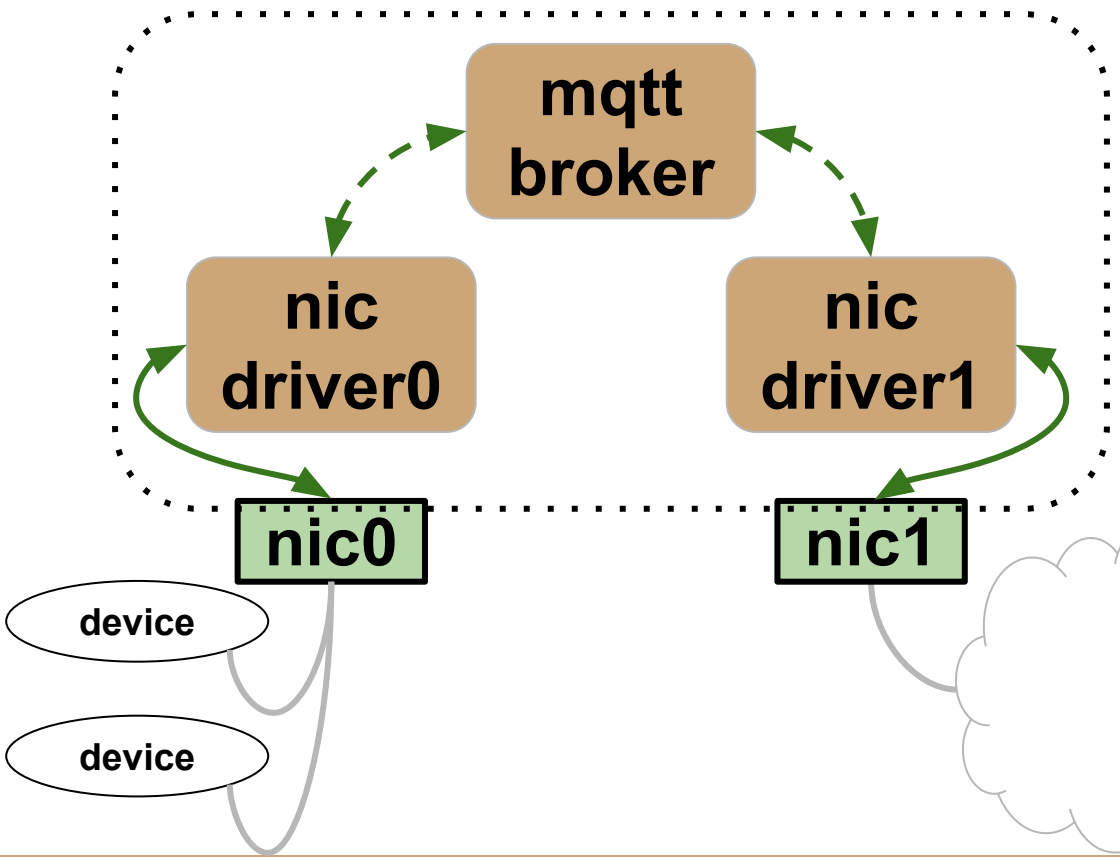


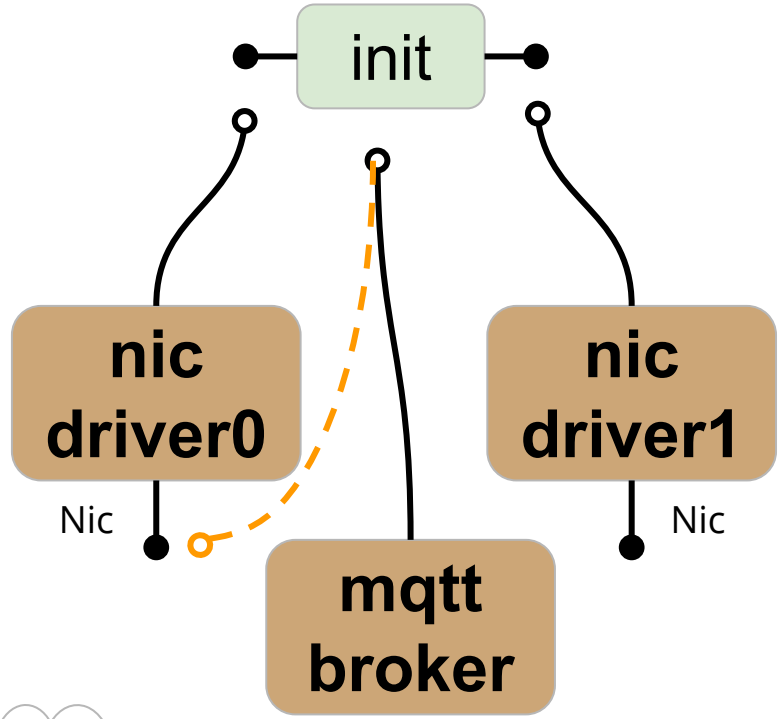
MQTT gateway



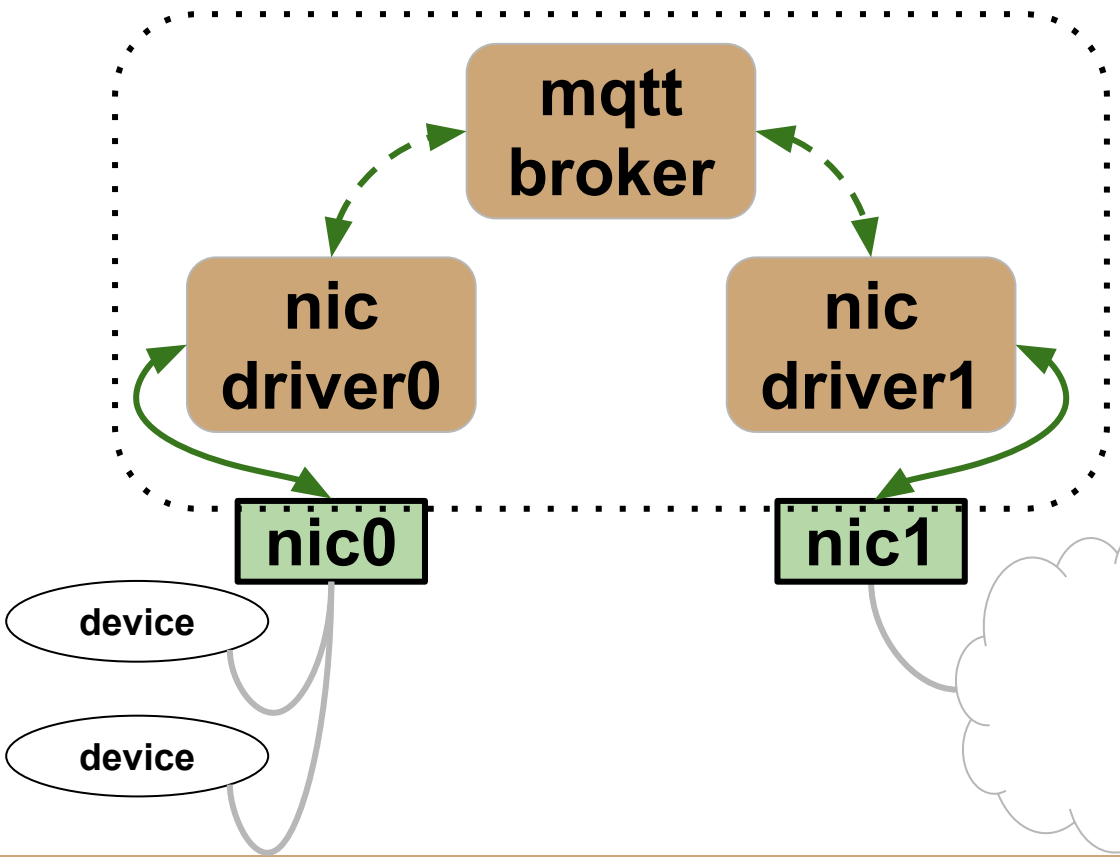


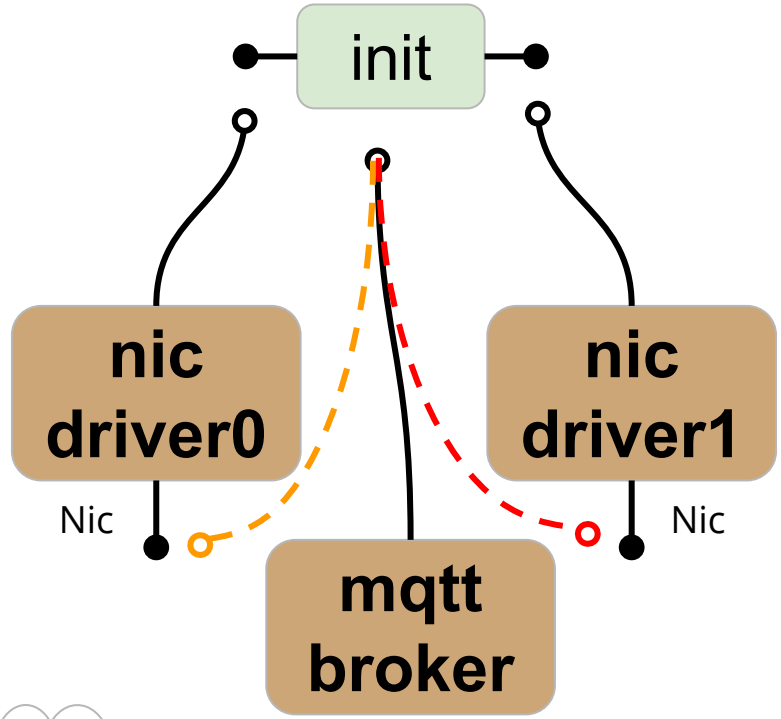
MQTT gateway



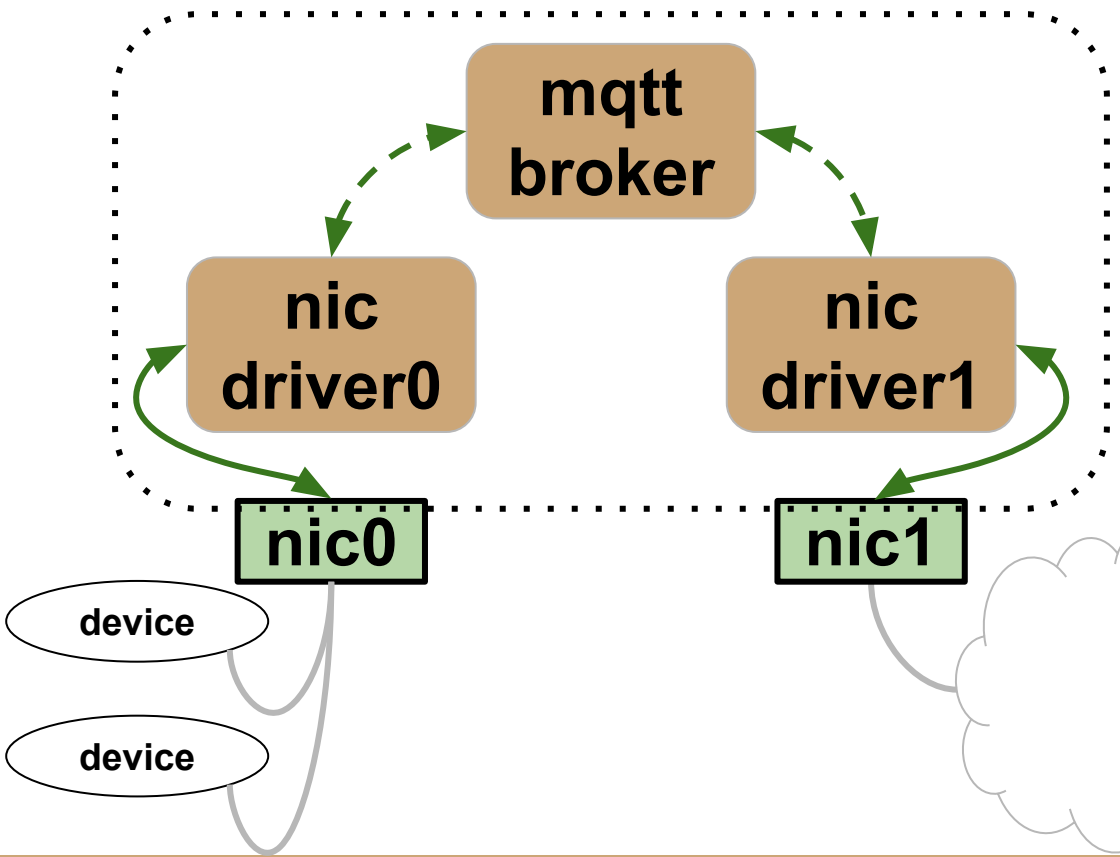


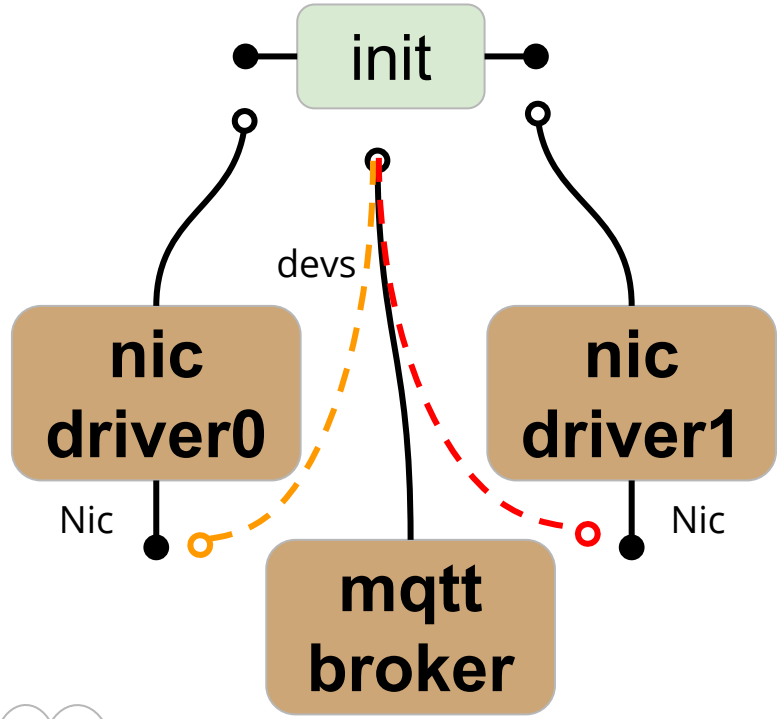
MQTT gateway



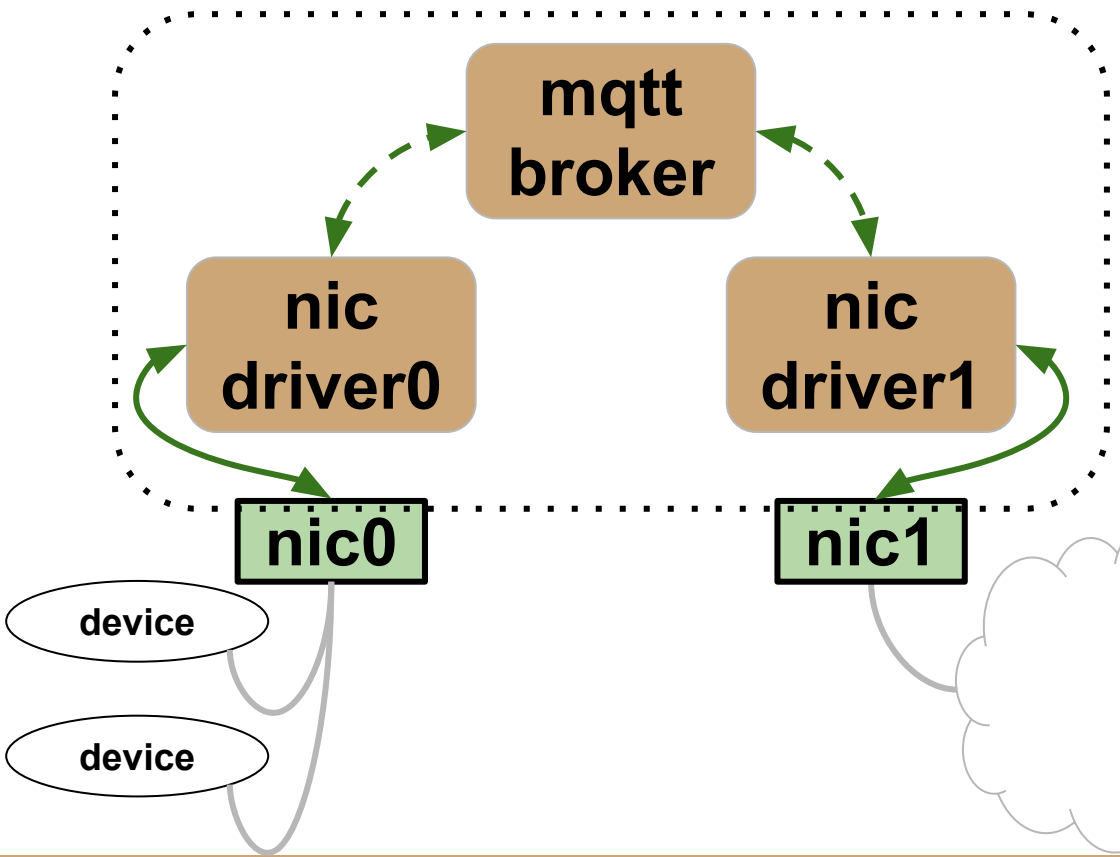


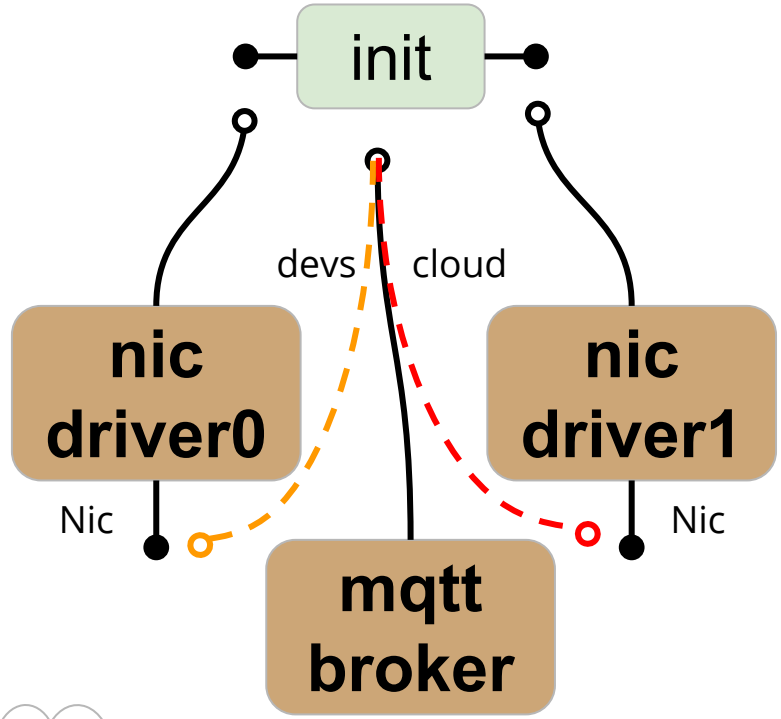
MQTT gateway



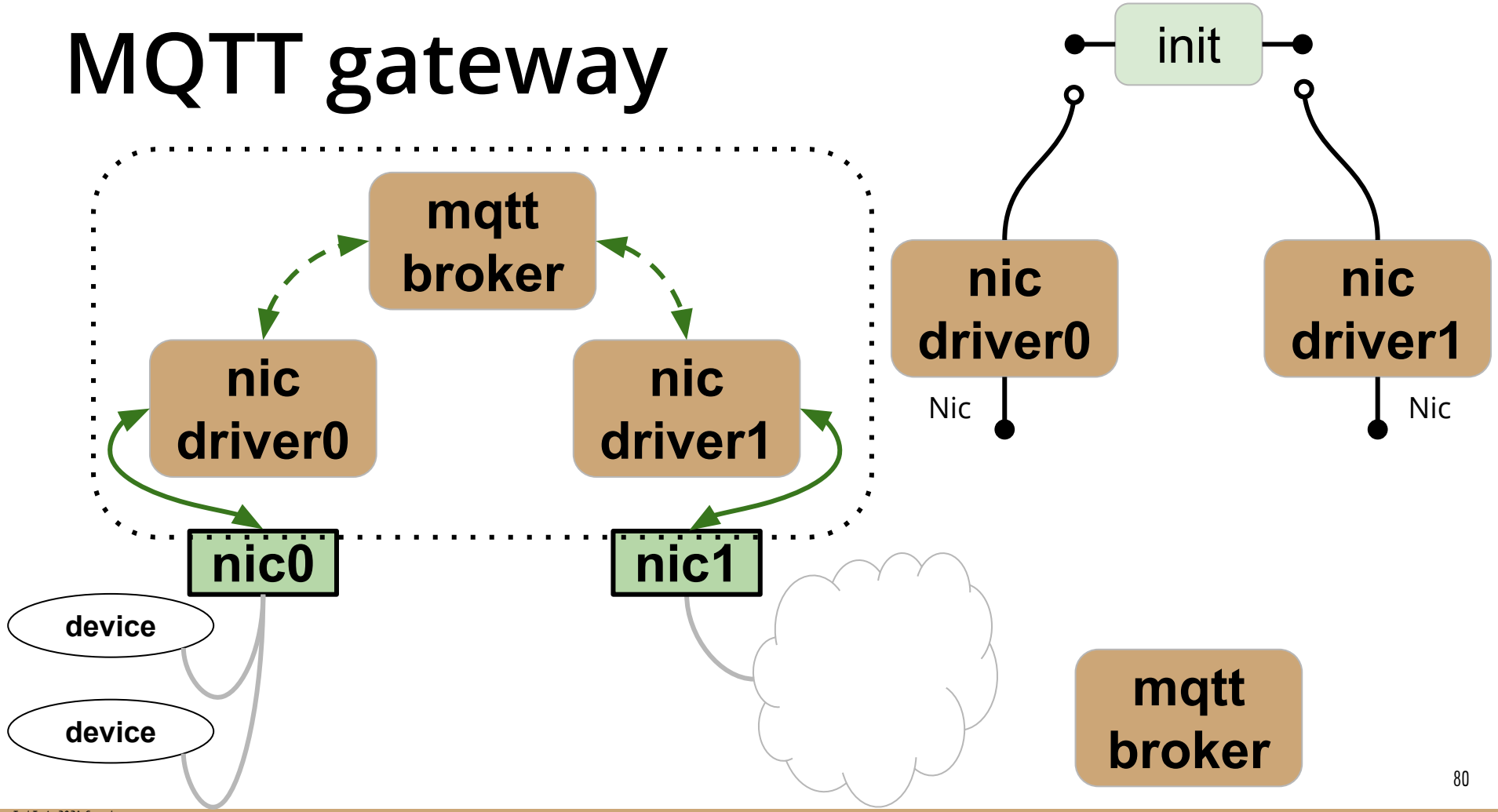


MQTT gateway

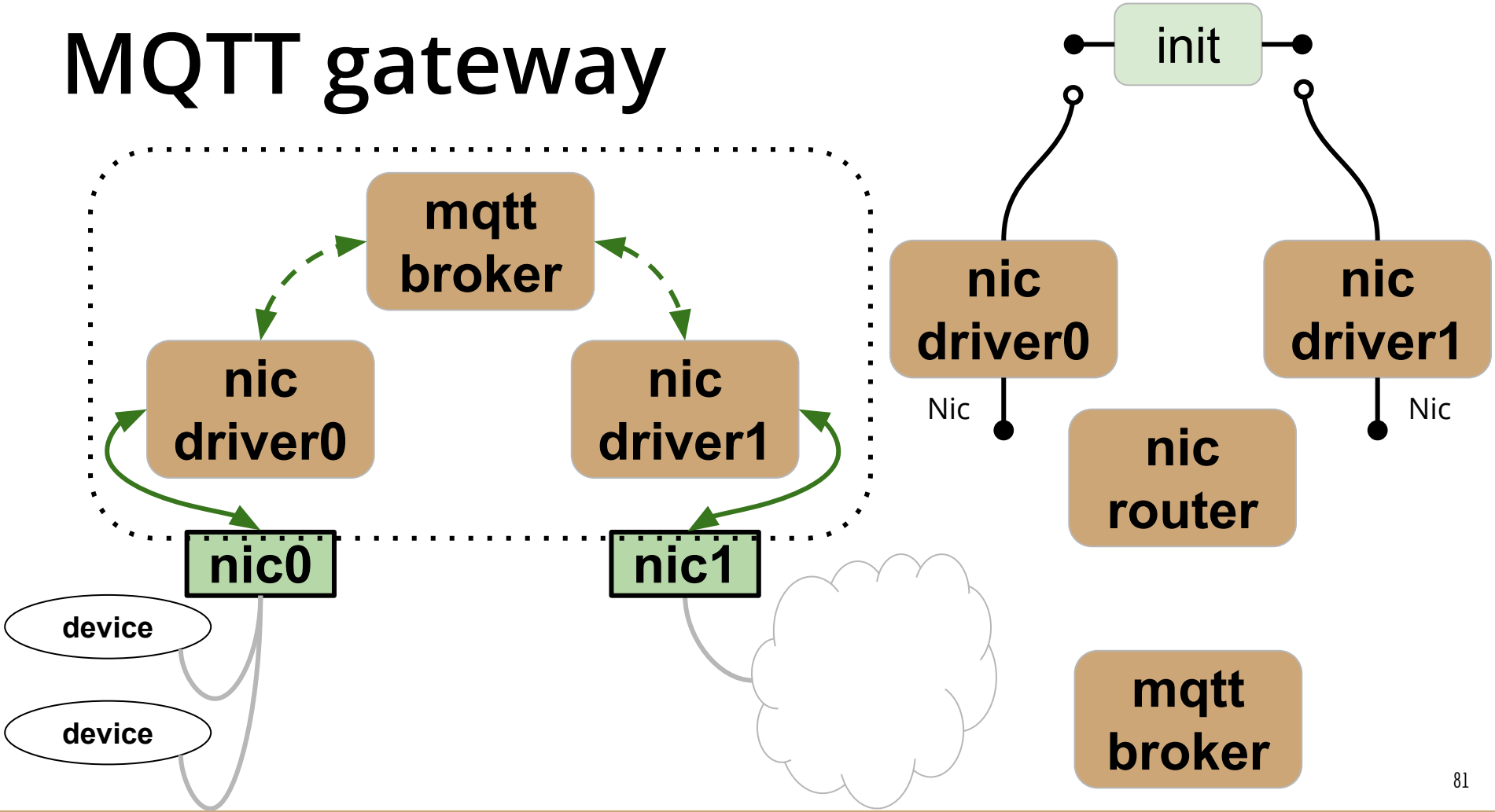




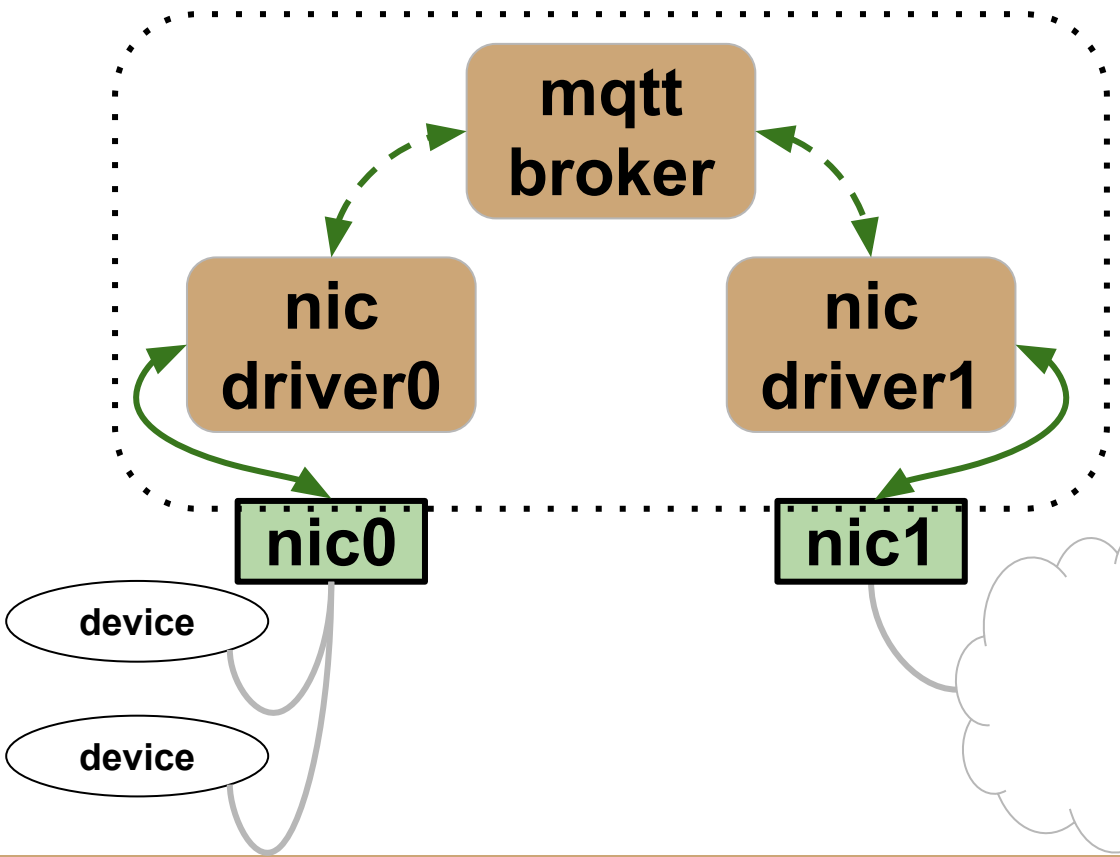
MQTT gateway

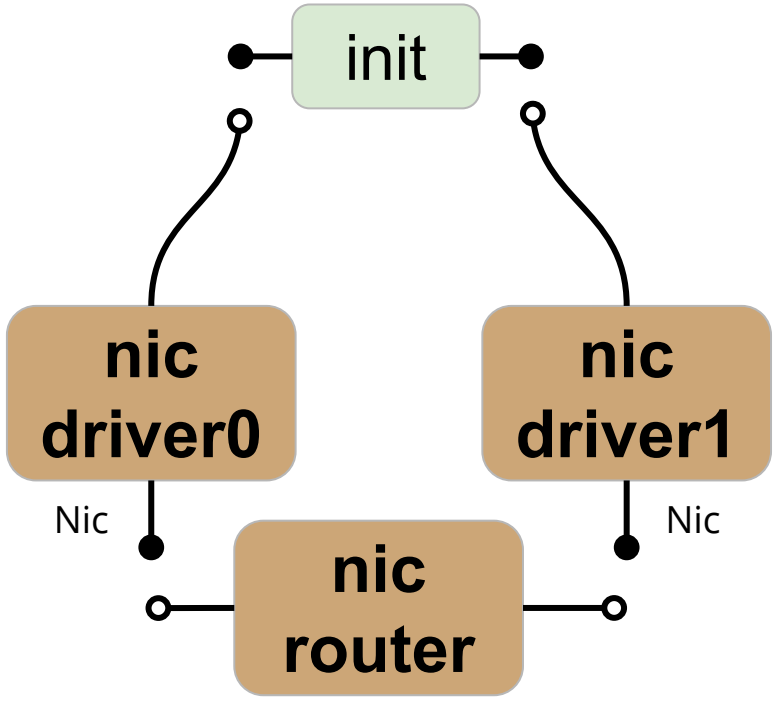


MQTT gateway



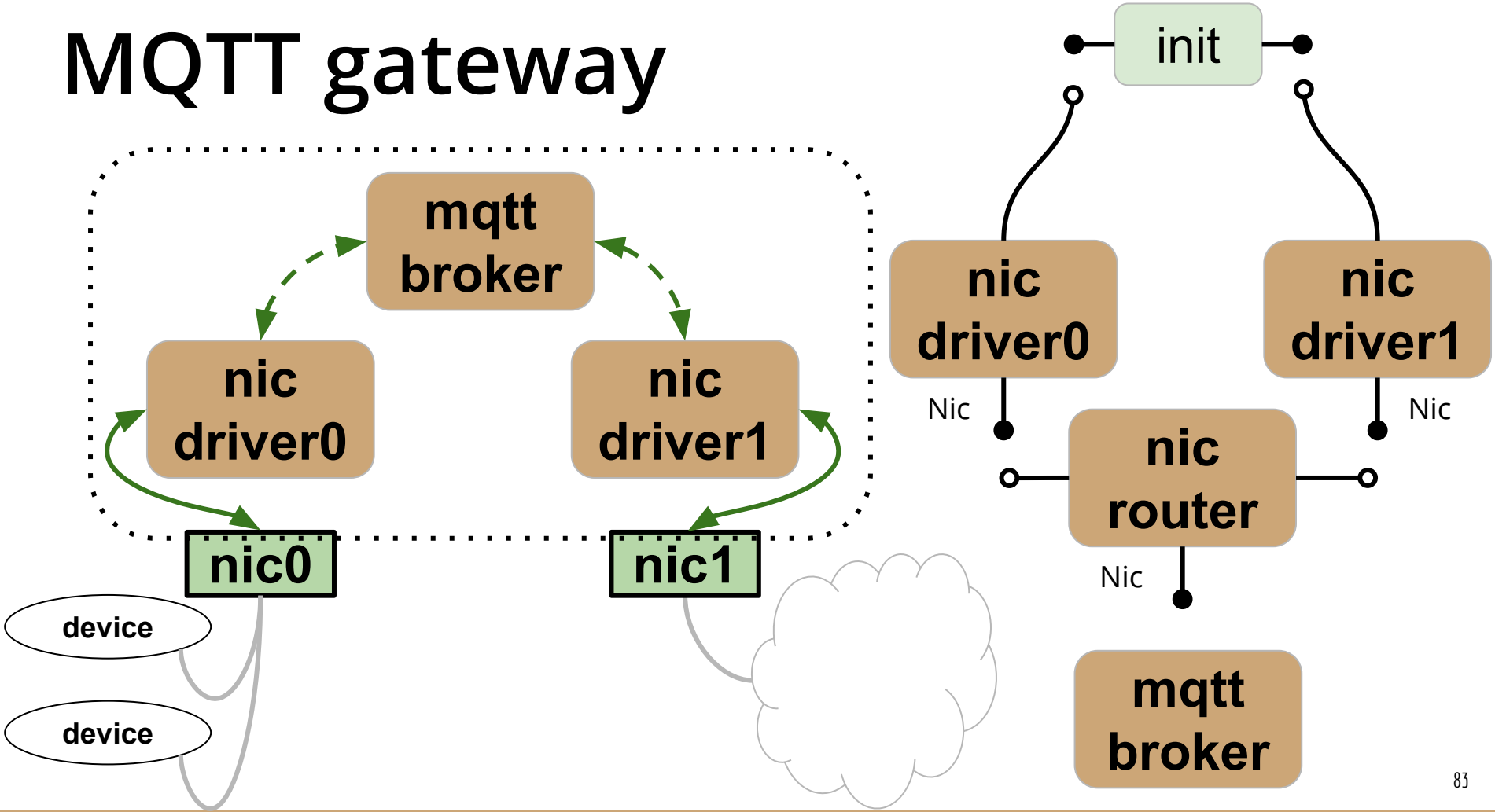
MQTT gateway



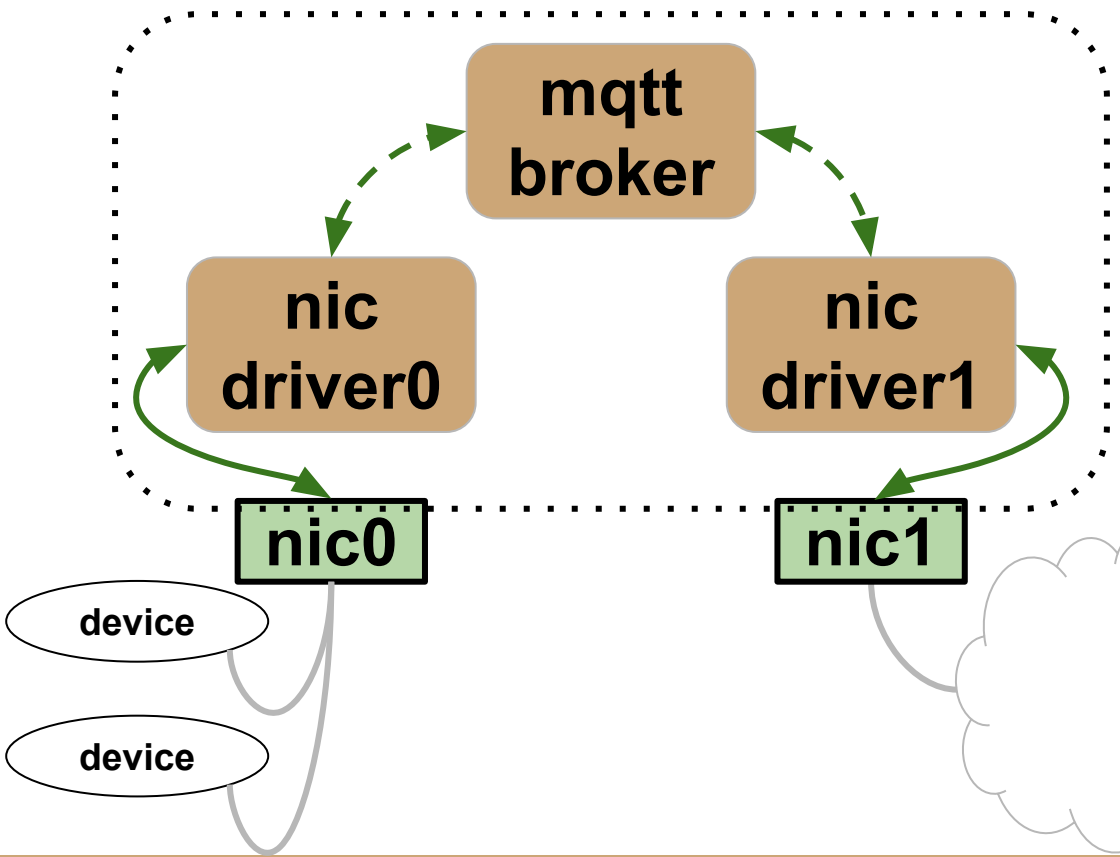


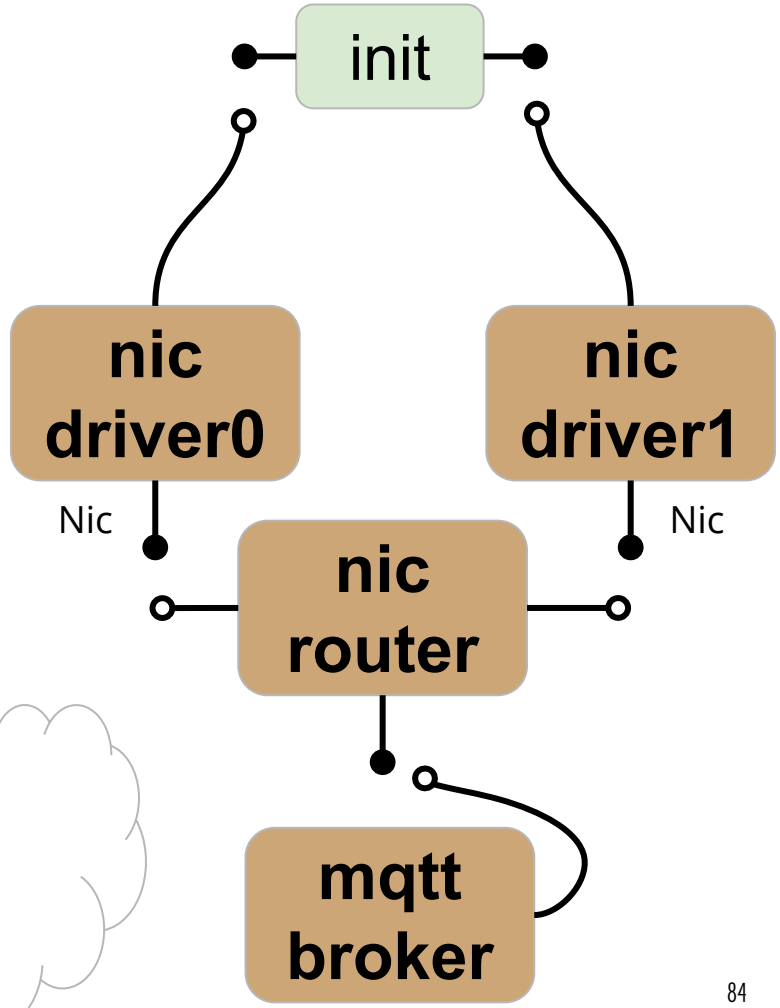
mqtt
broker

MQTT gateway

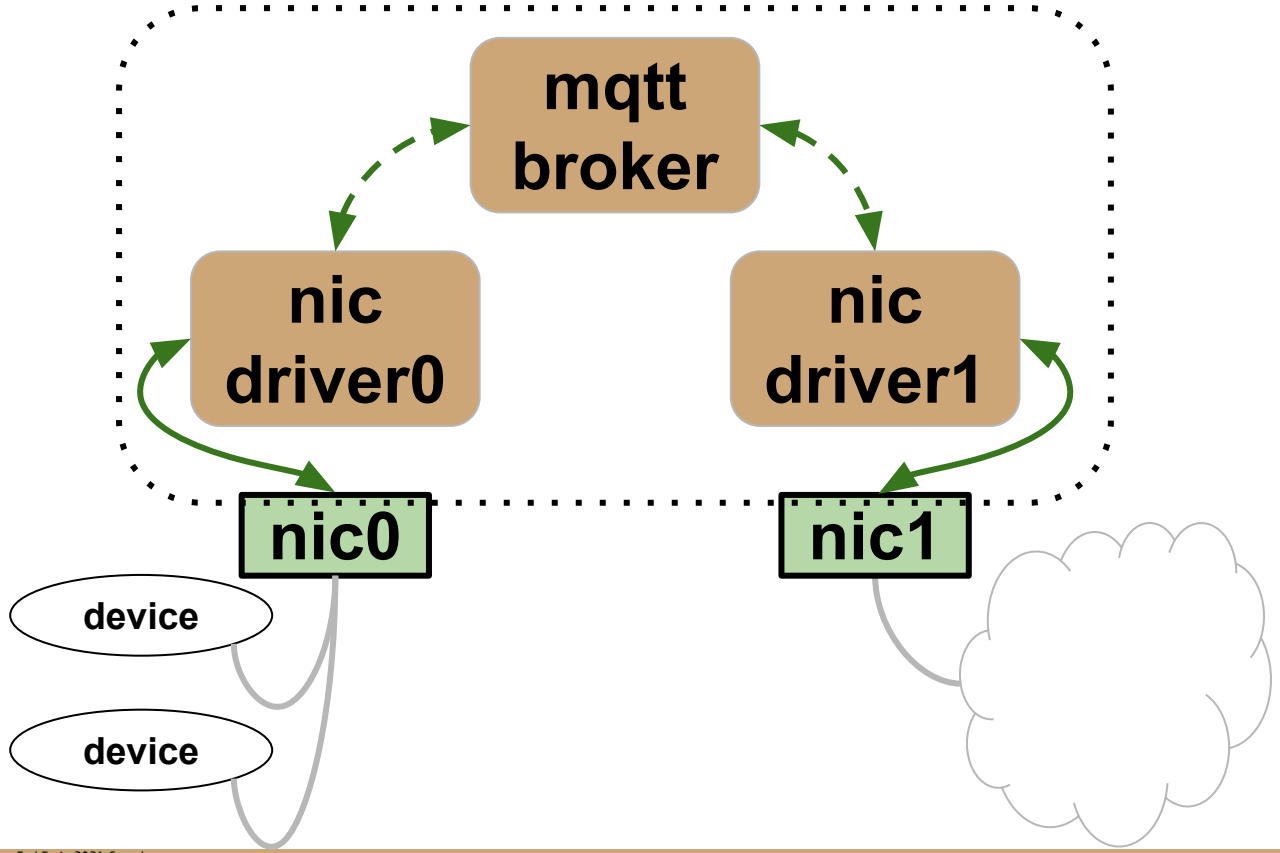


MQTT gateway

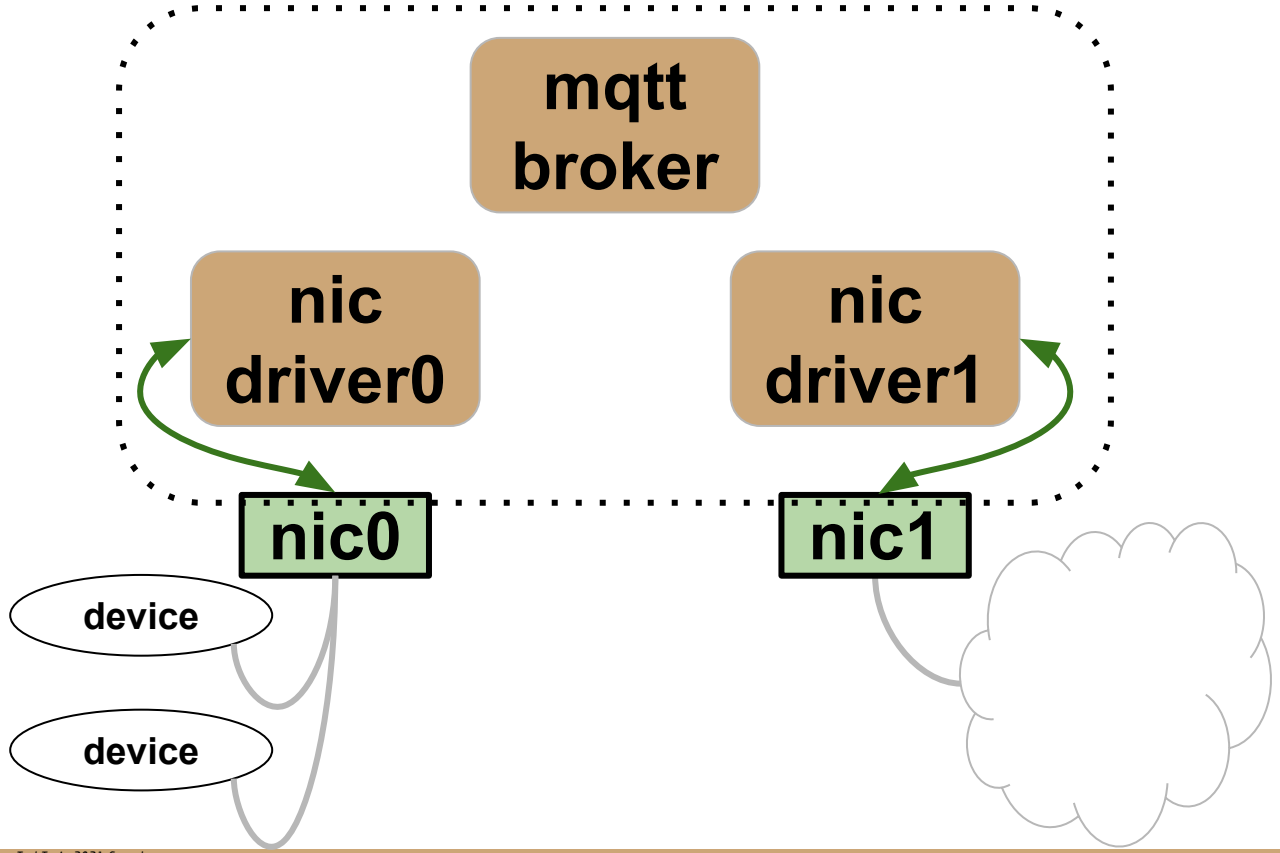




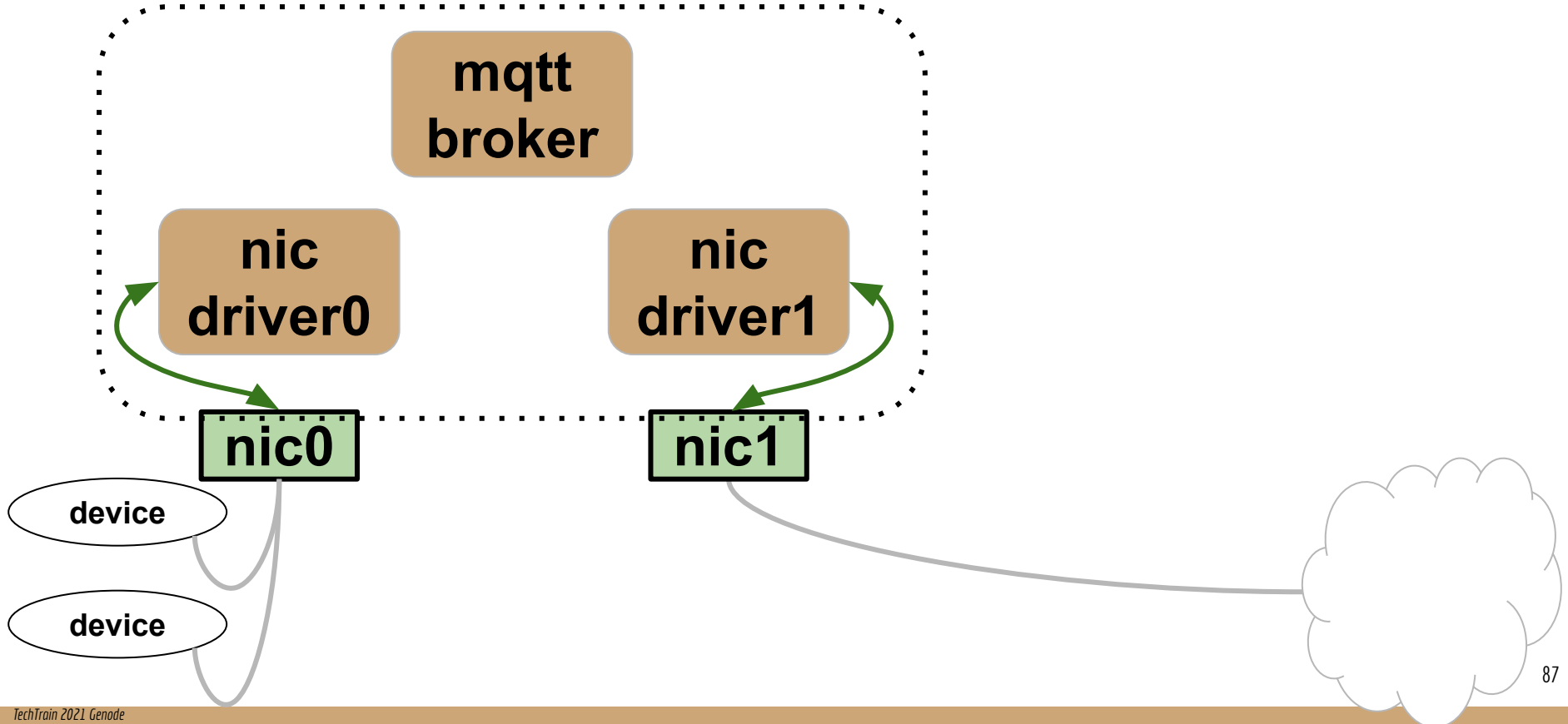
MQTT gateway



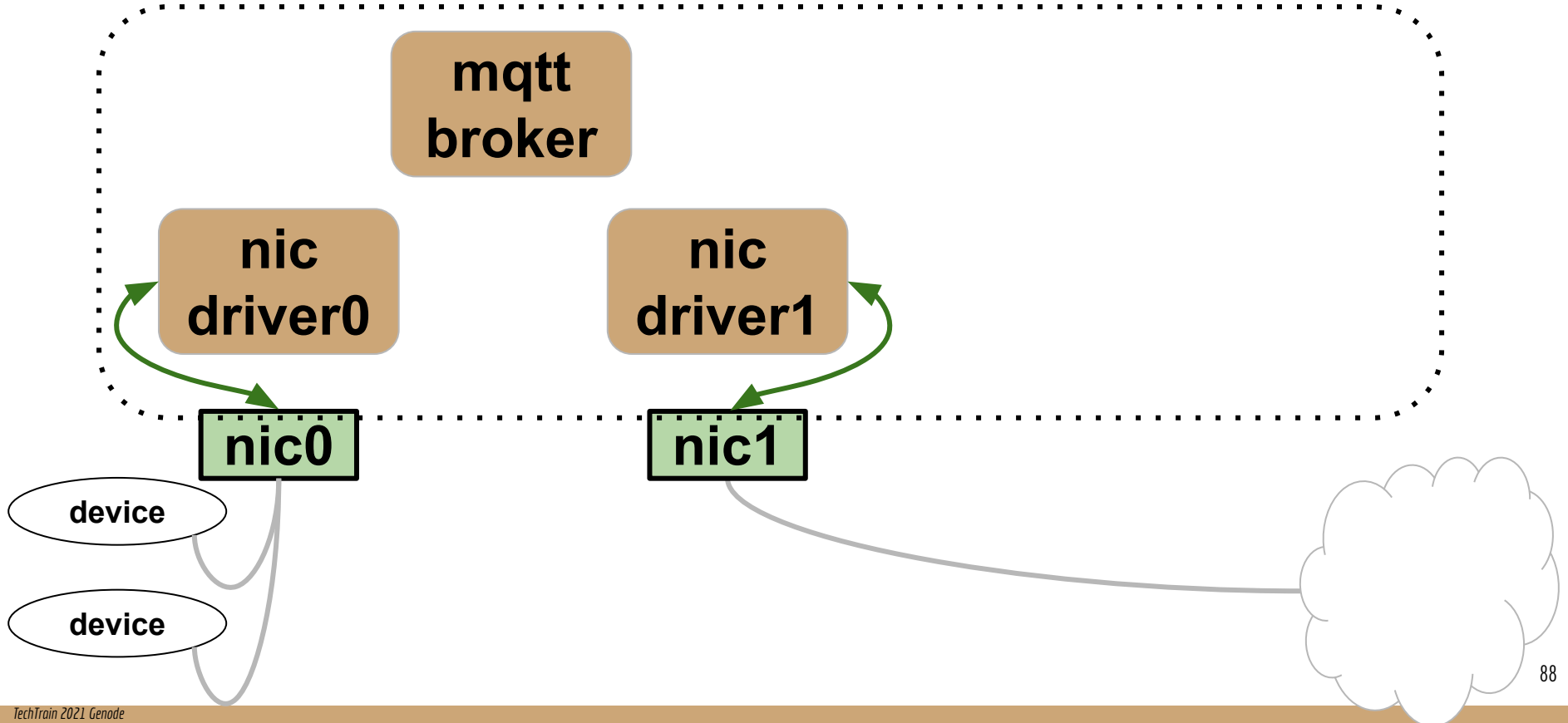
MQTT gateway



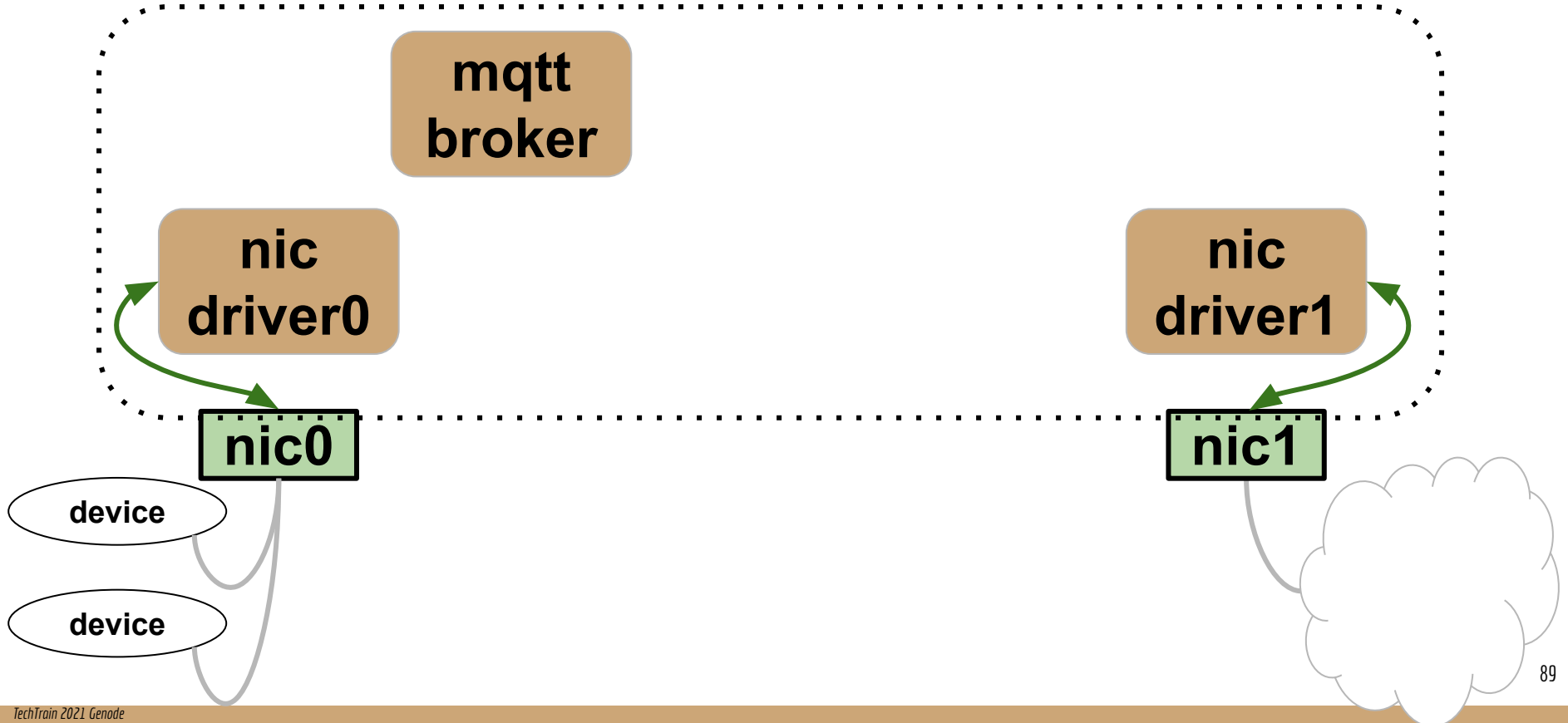
MQTT gateway



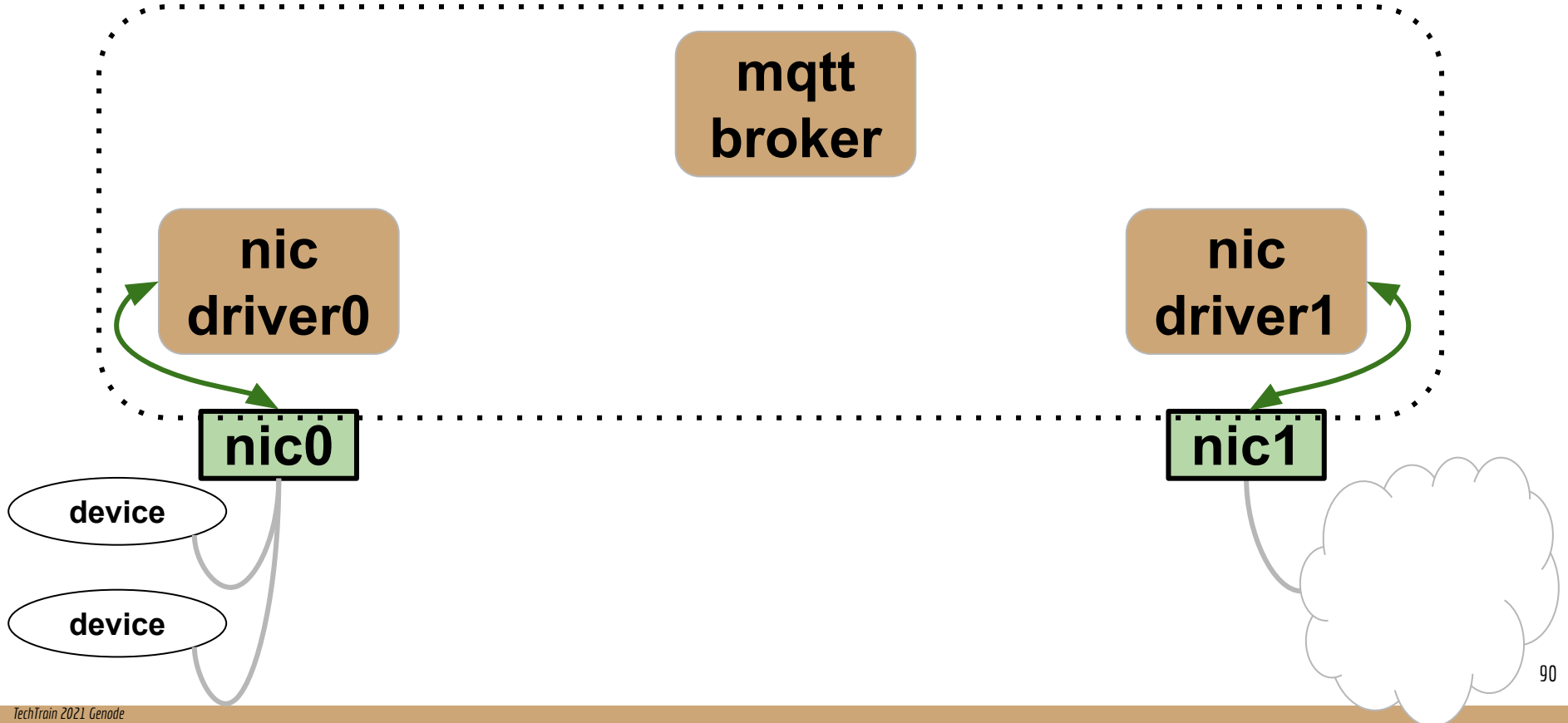
MQTT gateway



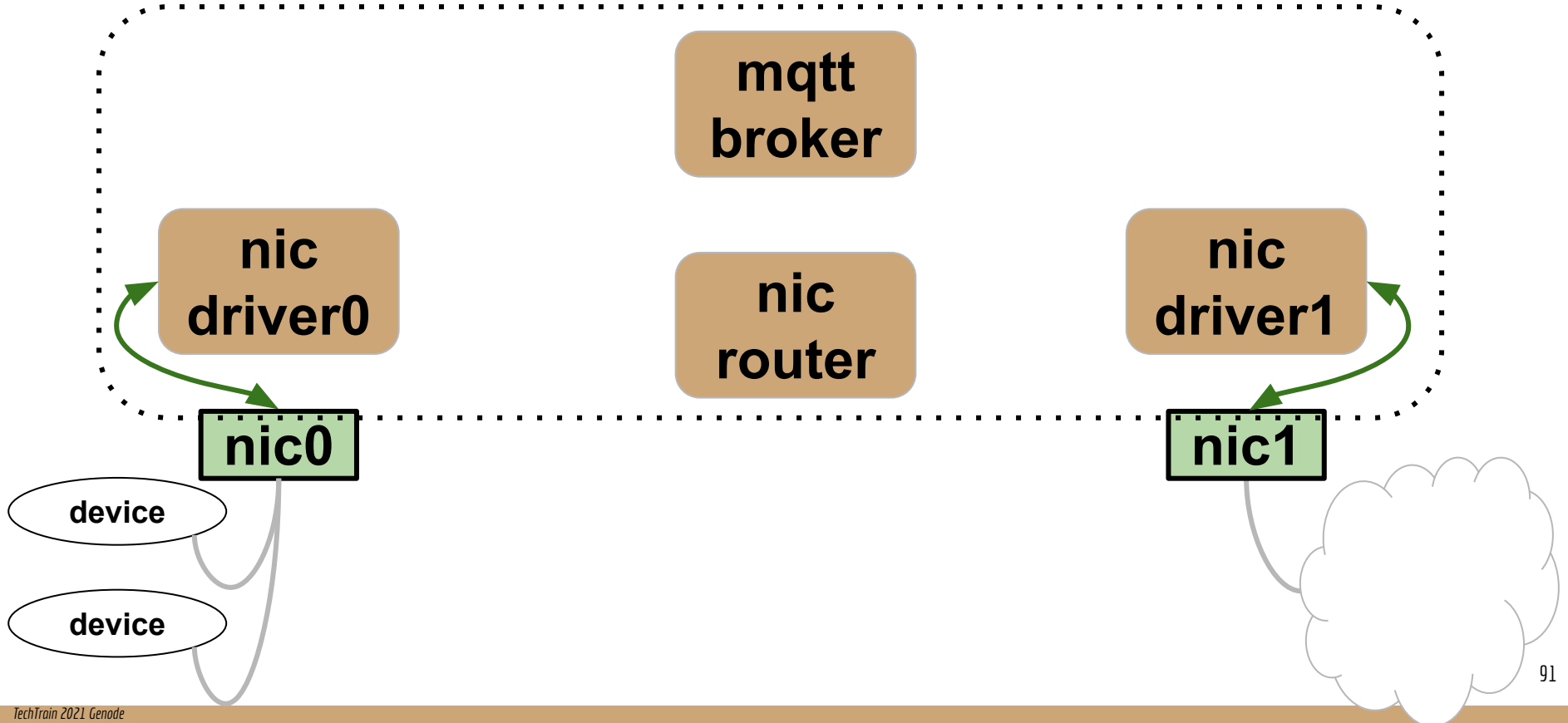
MQTT gateway



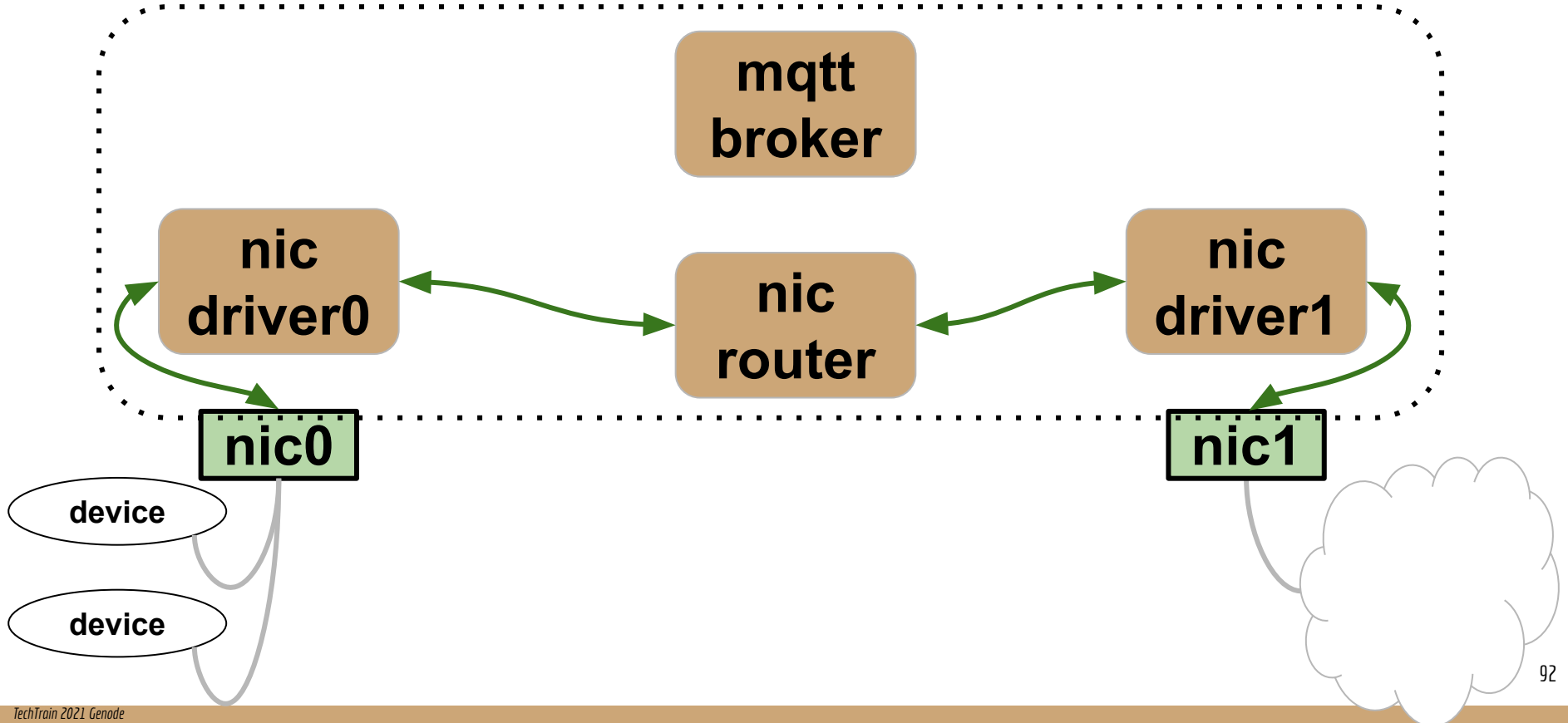
MQTT gateway



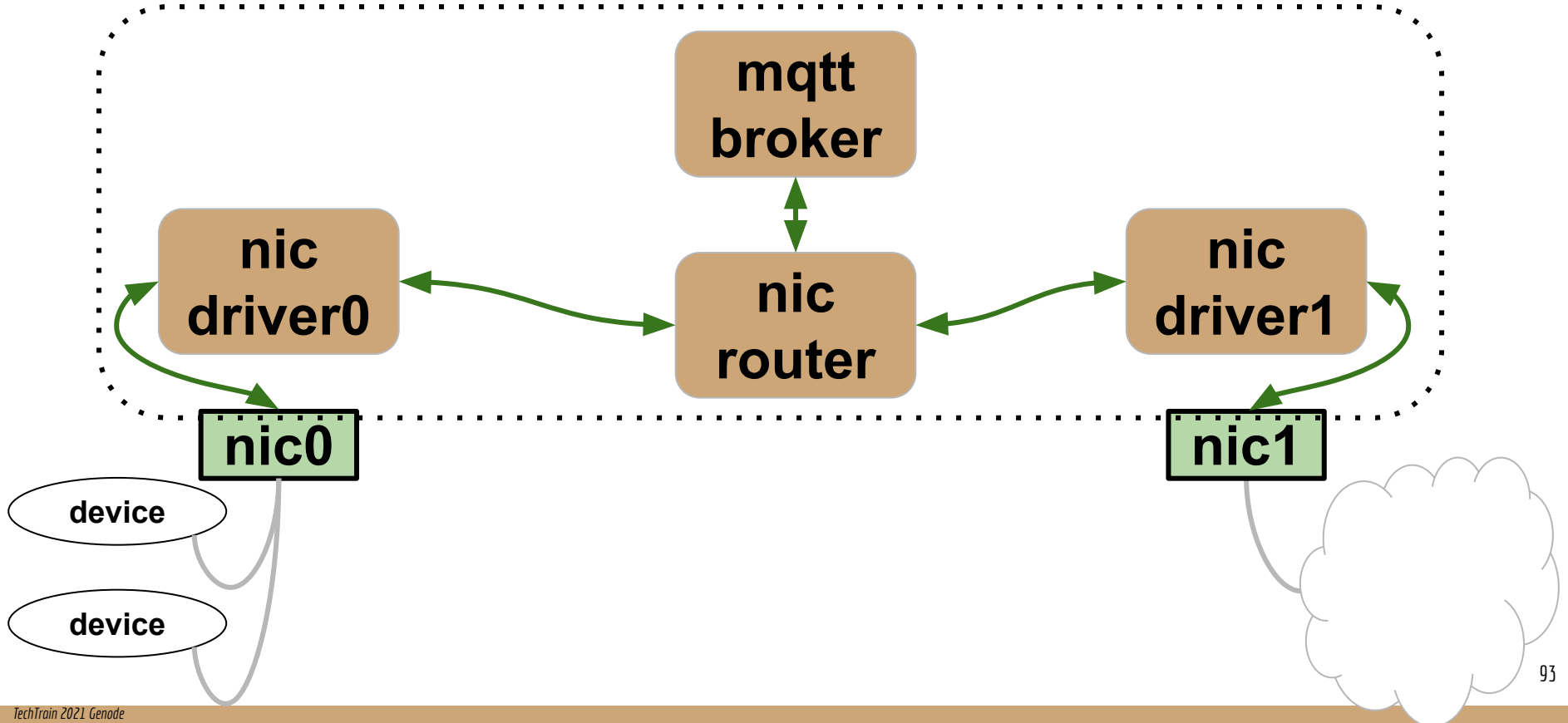
MQTT gateway



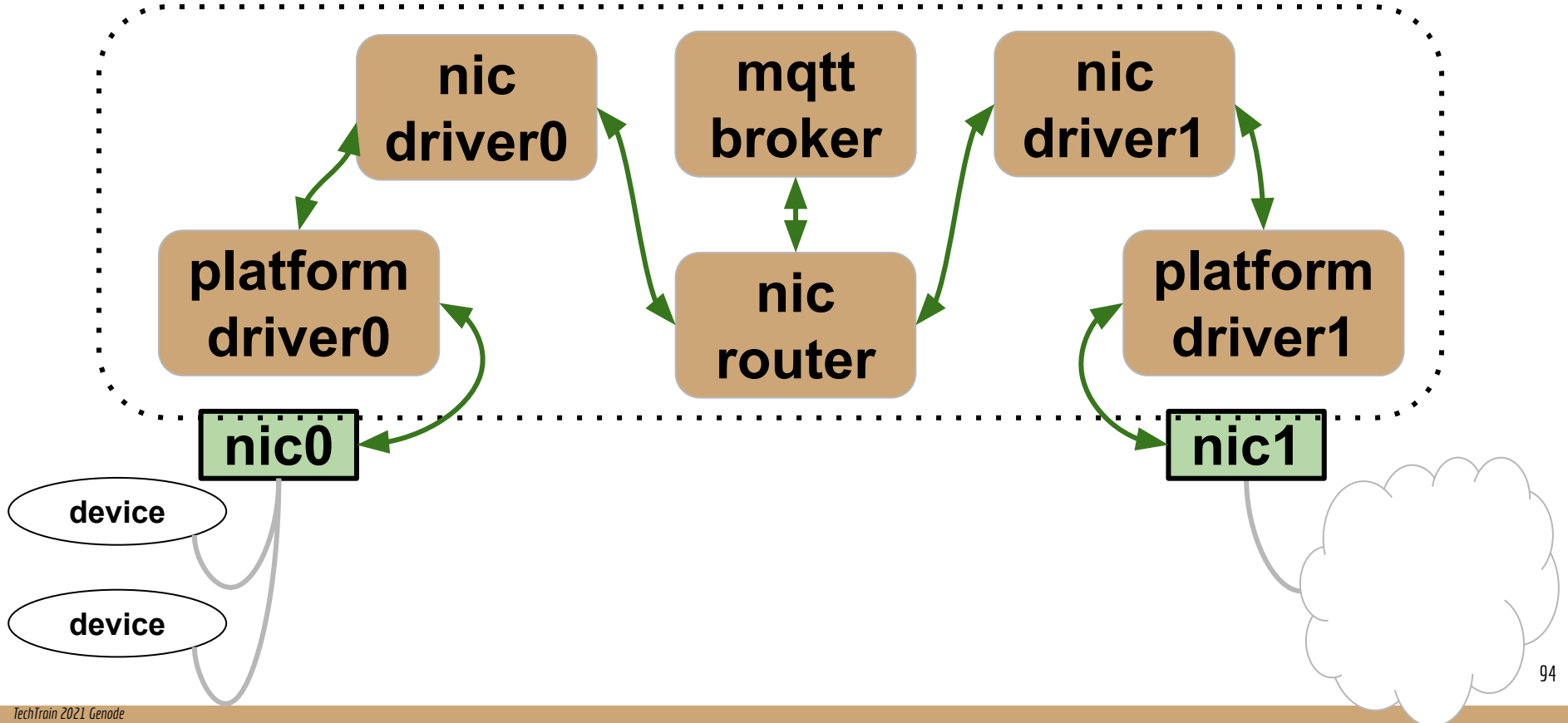
MQTT gateway



MQTT gateway



MQTT gateway base-hw



Genode: конфигурация init

```
1  <config>
2    <parent-provides>
3        <service name="LOG"/>
4        <service name="PD"/>
5        <service name="CPU"/>
6        <service name="ROM"/>
7    </parent-provides>
8    <default-route>
9        <any-service> <parent/> </any-service>
10   </default-route>
11  <start name="app"> </start>
12 </config>
```

Genode: конфигурация init

```
1  <config>
2    <parent-provides>
3        <service name="LOG"/>
4        <service name="PD"/>
5        <service name="CPU"/>
6        <service name="ROM"/>
7    </parent-provides>
8    <default-route>
9        <any-service> <parent/> </any-service>
10   </default-route>
11  <start name="app"> </start>
12 </config>
```

Сервисы, предоставляемые родителем

Genode: конфигурация init

```
1  <config>
2    <parent-provides>
3        <service name="LOG"/>
4        <service name="PD"/>
5        <service name="CPU"/>
6        <service name="ROM"/>
7    </parent-provides>
8    <default-route>
9        <any-service> <parent/> </any-service>
10   </default-route>
11   <start name="app"> </start>
12 </config>
```

Сервисы, предоставляемые родителем

Где искать сервисы по умолчанию

Genode: конфигурация init

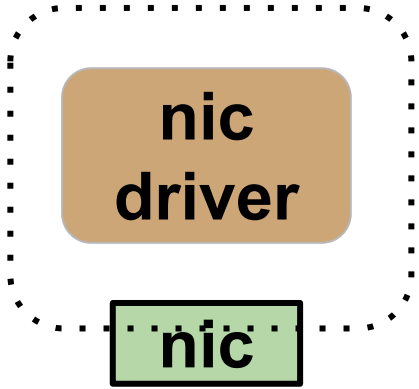
```
1  <config>
2    <parent-provides>
3        <service name="LOG"/>
4        <service name="PD"/>
5        <service name="CPU"/>
6        <service name="ROM"/>
7    </parent-provides>
8    <default-route>
9        <any-service> <parent/> </any-service>
10   </default-route>
11  <start name="app"> </start>
12 </config>
```

Сервисы, предоставляемые родителем

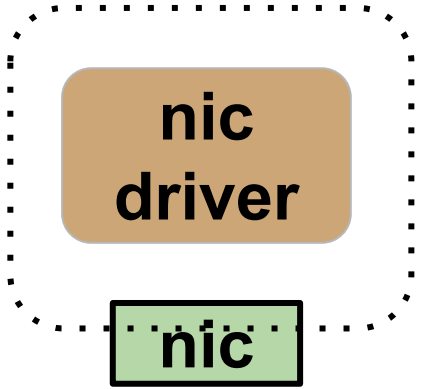
Где искать сервисы по умолчанию

Приложение

Genode: конфигурация init

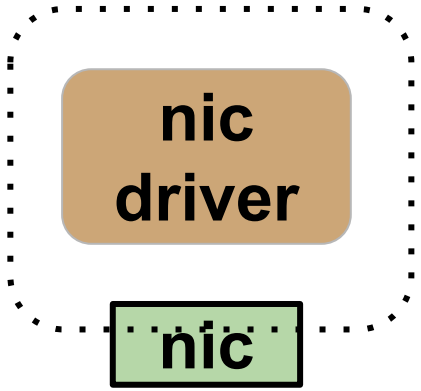


Genode: nic_driver linux




```
1  <start name="linux_nic_drv" ld="no">
2    <resource name="RAM" quantum="4M"/>
3    <config tap="tap0" mac="02:00:00:00:00:00"/>
4  </start>
```

Genode: nic_driver base-hw




```
1   <start name="fec_nic_drv" caps="150">
2       <provides>
3           <service name="Nic"/>
4       </provides>
5       <resource name="RAM" quantum="16M"/>
6       <config mac="02:00:00:00:00:00"/>
7       <route>
8           <any-service> <parent/> <any-child/>
9           </any-service>
10      </route>
11  </start>
```

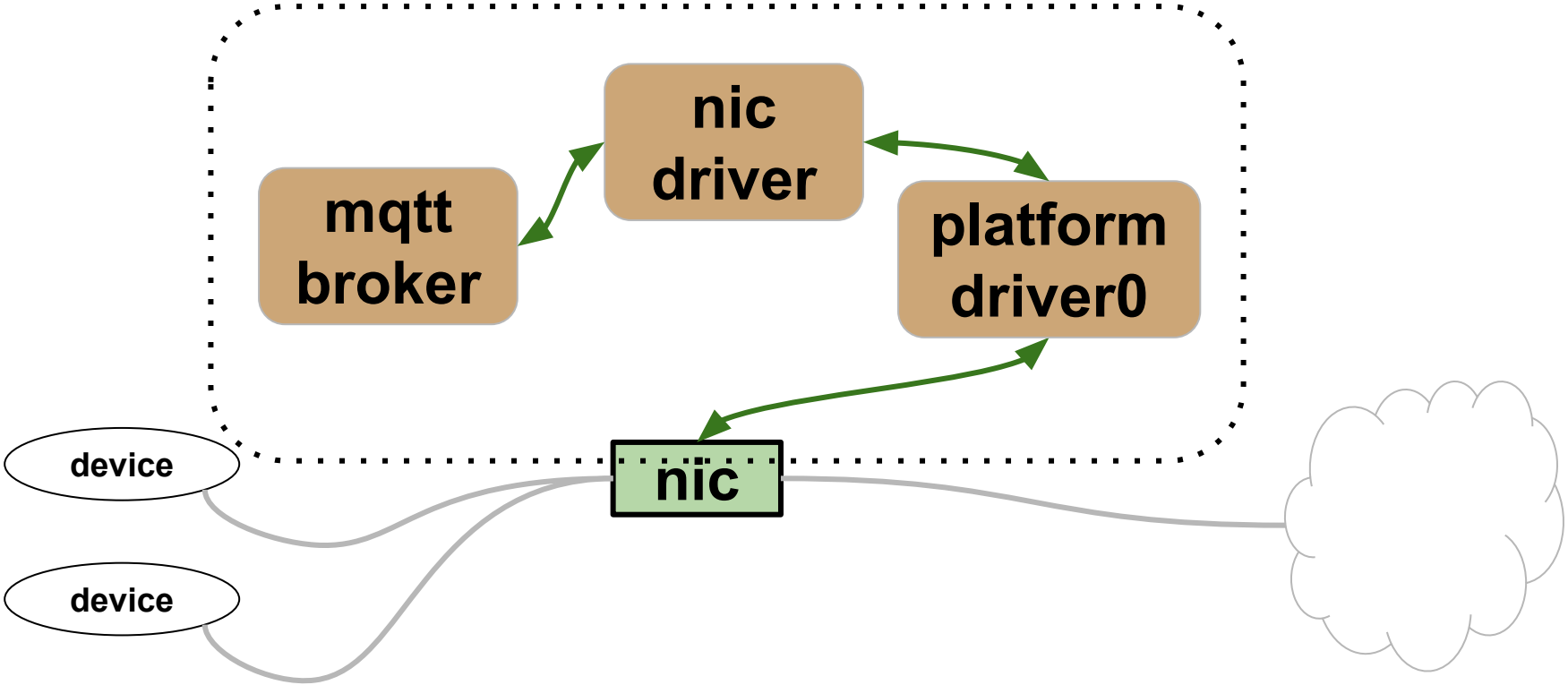
Genode: platform_drv

```
1  <start name="platform_drv" caps="350" managing_system="yes">
2    <resource name="RAM" quantum="4M"/>
3    <provides><service name="Platform"/></provides>
4    <config>
5      <device name="fec" type="fsl,imx6sx-fec">
6        <io_mem   address="0x60ff0000" size="0x10000"/>
7        <irq      number="140"/>
8      </device>
9      <policy label="fec_nic_drv -> " info="yes">
10       <device name="fec"/>
11     </policy>
12   </config>
13 </start>
```

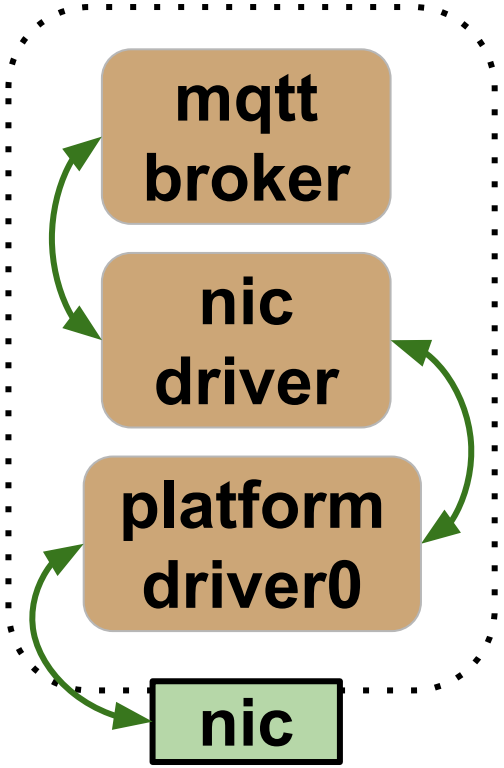
Genode: конфигурация init

```
1  <start name="platform_drv" caps="350" managing_system="yes">
2    <resource name="RAM" quantum="4M"/>
3    <provides><service name="Platform"/></provides>
4    <config>
5      <device name="fec" type="fsl,imx6sx-fec">
6        <io_mem   address="0x60ff0000" size="0x10000"/>
7        <irq      number="140"/>
8      </device>
9      <policy label="fec_nic_drv -> " info="yes">
10       <device name="fec"/>
11     </policy>
12   </config>
13 </start>
```

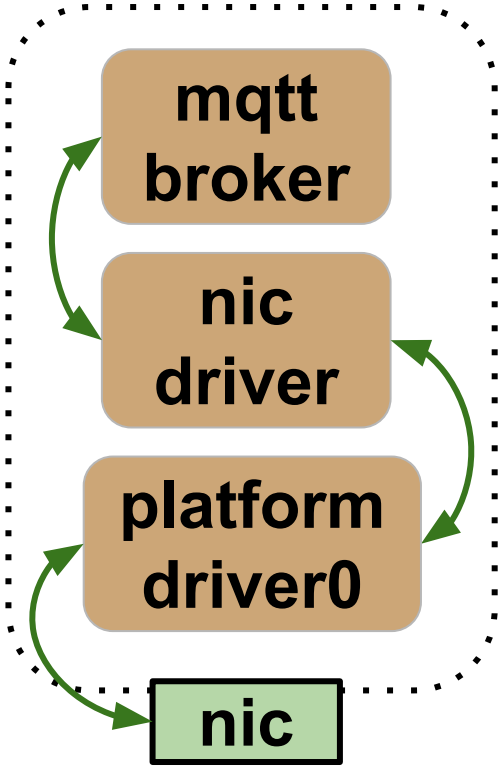
MQTT gateway



Genode: конфигурация init

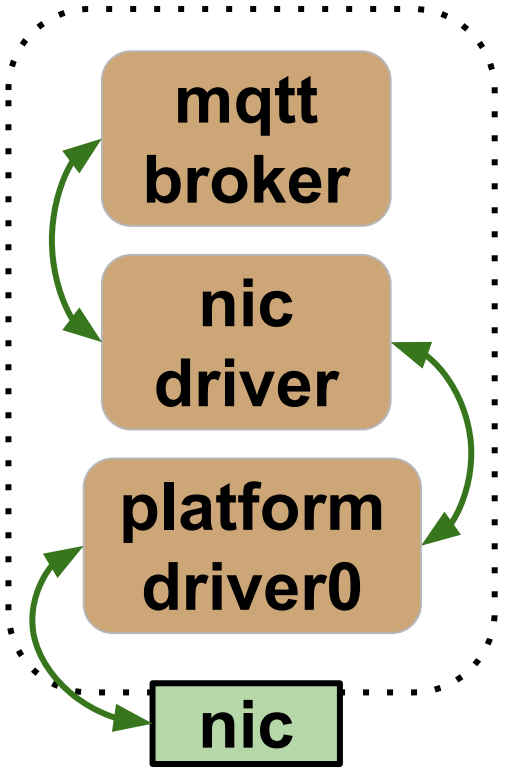


Genode: конфигурация init



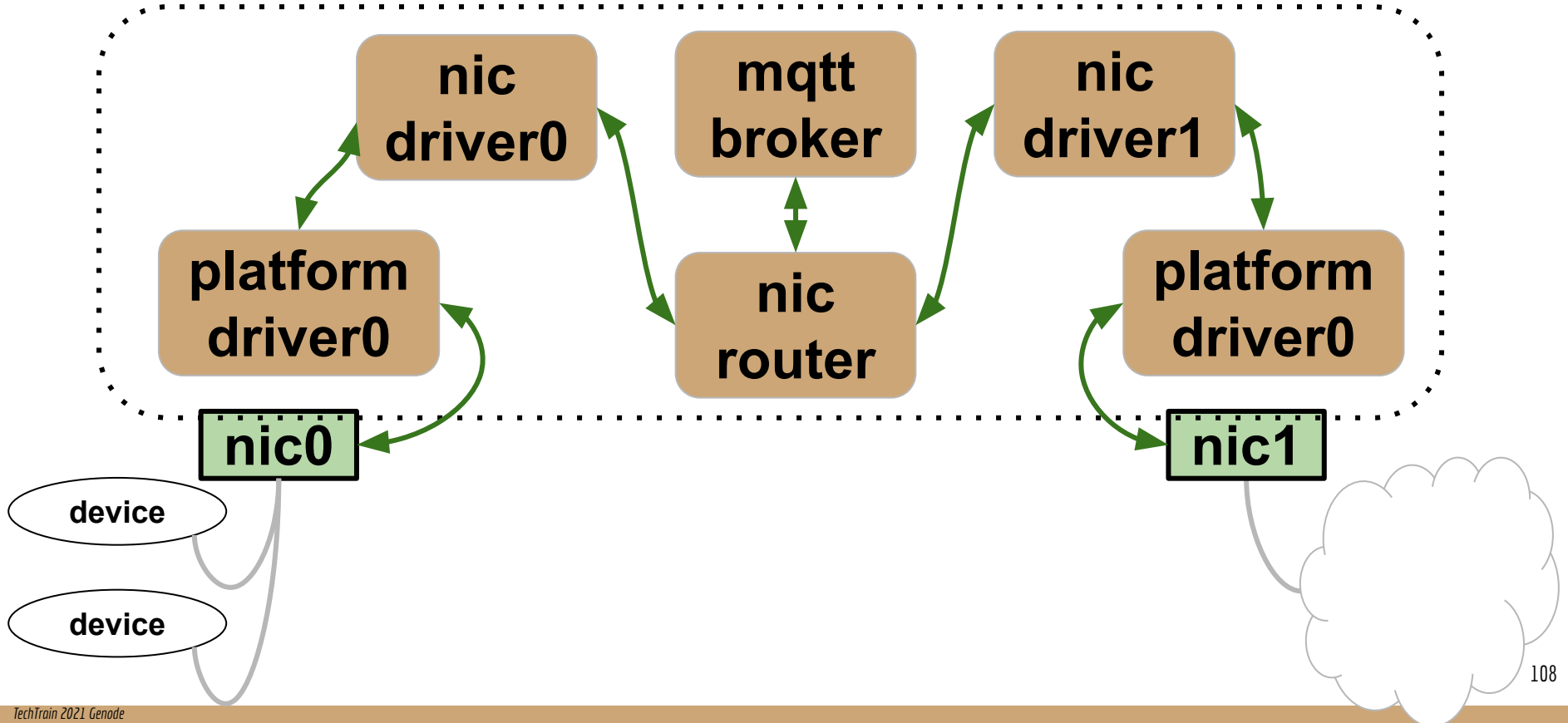

```
<start name="mqtt_broker" caps="200">
 <resource name="RAM" quantum="8M"/>
 <config>
   <arg value="mqtt_broker"/>
   <arg value="/etc/mqtt_broker.conf"/>
   <libc stdin="/dev/null" stdout="/dev/log" stderr="/dev/log" socket="/dev/socket"/>
   <vfs>
     <dir name="dev">
       <dir name="socket">
         <lwip dhcp="yes"/>
       </dir>
       <log/> <null/>
     </dir>
     <dir name="etc">
       <inline name="mqtt_broker.conf">
listener 1883
user root
password_file /etc/password
      </inline>
     </dir>
   </vfs>
 </config>
</start>
```

Genode: конфигурация init




```
4 <arg value="mqtt_broker"/>
5 <arg value="/etc/mqtt_broker.conf"/>
6 <libc stdin="/dev/null" stdout="/dev/log"
        stderr="/dev/log" socket="/dev/socket"/>
7  <vfs>
8    <dir name="dev">
9     <dir name="socket"><lwip dhcp="yes"/> </dir>
10   </dir>
11   <dir name="etc">
12     <inline name="mqtt_broker.conf"></inline>
20   </dir>
21 </vfs>
```

MQTT gateway base-hw



Genode: nic_router

```
<start name= "nic_router"  caps= "400" >
    <resource name=  "RAM"  quantum= "16M" />
    <provides>
        <service name=  "Nic" />
        <service name=  "Uplink" />
    </provides>
    <config verbose_domain_state=  "no"
            verbose_packets=  "no"
            verbose_packet_drop=  "no" >

        <domain name=  "uplink"  interface= "192.168.1.222/24"  gateway= "192.168.1.1" >
            <nat domain=  "server"  tcp-ports= "100"  />
            <nat domain=  "client"  tcp-ports= "100"  />
            <tcp-forward port=  "1883"  domain= "server"  to= "10.10.10.2" />
        </domain>

        <domain name=  "server"  interface= "10.10.10.1/24" >
            <tcp dst=  "10.10.20.0/24" > <permit-any domain=  "client" /> </tcp>
            <tcp dst=  "0.0.0.0/0" > <permit-any domain=  "uplink" /> </tcp>
        </domain>
        <domain name=  "client"  interface= "10.10.20.1/24" >
            <tcp dst=  "10.10.10.0/24" > <permit-any domain=  "server" /> </tcp>
            <tcp dst=  "0.0.0.0/0" > <permit-any domain=  "uplink" /> </tcp>
        </domain>
        <policy label=  "mqtt_bridge -> lwip"  domain= "server" />
        <policy label=  "ifm_mqtt -> lwip"     domain= "client" />
        <policy label=  "nic -> "              domain=  "uplink" />
    </config>
    <route>
        <any-service> <parent/> <any-child/> </any-service>
    </route>
</start>
```

Genode: nic_router

```
17 <domain name="server" interface="10.10.10.1/24">
18   <tcp dst="10.10.20.0/24"> <permit-any domain="client"/> </tcp>
19   <tcp dst="0.0.0.0/0"> <permit-any domain="uplink"/> </tcp>
20 </domain>
21 <domain name="client" interface="10.10.20.1/24">
22   <tcp dst="10.10.10.0/24"> <permit-any domain="server"/> </tcp>
23   <tcp dst="0.0.0.0/0"> <permit-any domain="uplink"/> </tcp>
24 </domain>
```

Genode: nic_router

```
11 <domain name="uplink" interface="192.168.1.222/24"
           gateway="192.168.1.1">
12   <nat domain="server" tcp-ports="100" />
13   <nat domain="client" tcp-ports="100" />
14   <tcp-forward port="1883" domain="server" to="10.10.10.2"/>
15 </domain>

25 <policy label="mqtt_broker -> lwip" domain="server"/>
26 <policy label="mqtt_bridge -> lwip" domain="client"/>
27 <policy label="nic -> "             domain="uplink"/>
```

Genode: runscripts

для конфигурации системы — XML

для сборки и запуска образа — Tcl

для сборки пакетов и запуска скриптов — Make

mqtt_gateway.run

```
build \
 "core init app/platform_drv app/nic_router app/mqtt_broker"

create_boot_directory
install_config {...}
build_boot_image \
   "core ld.lib.so libc init platform_drv nic_router
mqtt_broker"
```

Genode: make run

make run/mqtt_gateway KERNEL=base-hw

https://genode.org/

https://github.com/genodelabs

http://genodians.org/

users@lists.genode.org

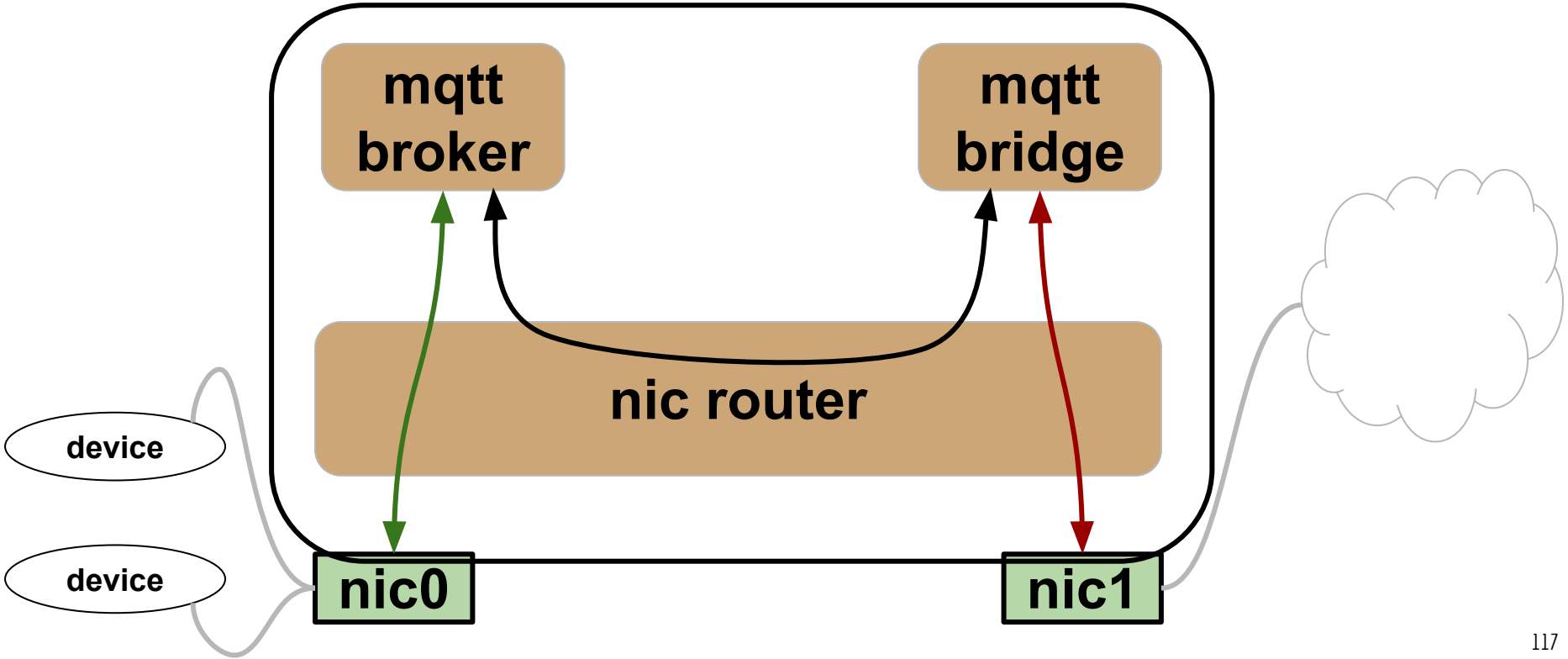
https://www.reddit.com/r/genode/

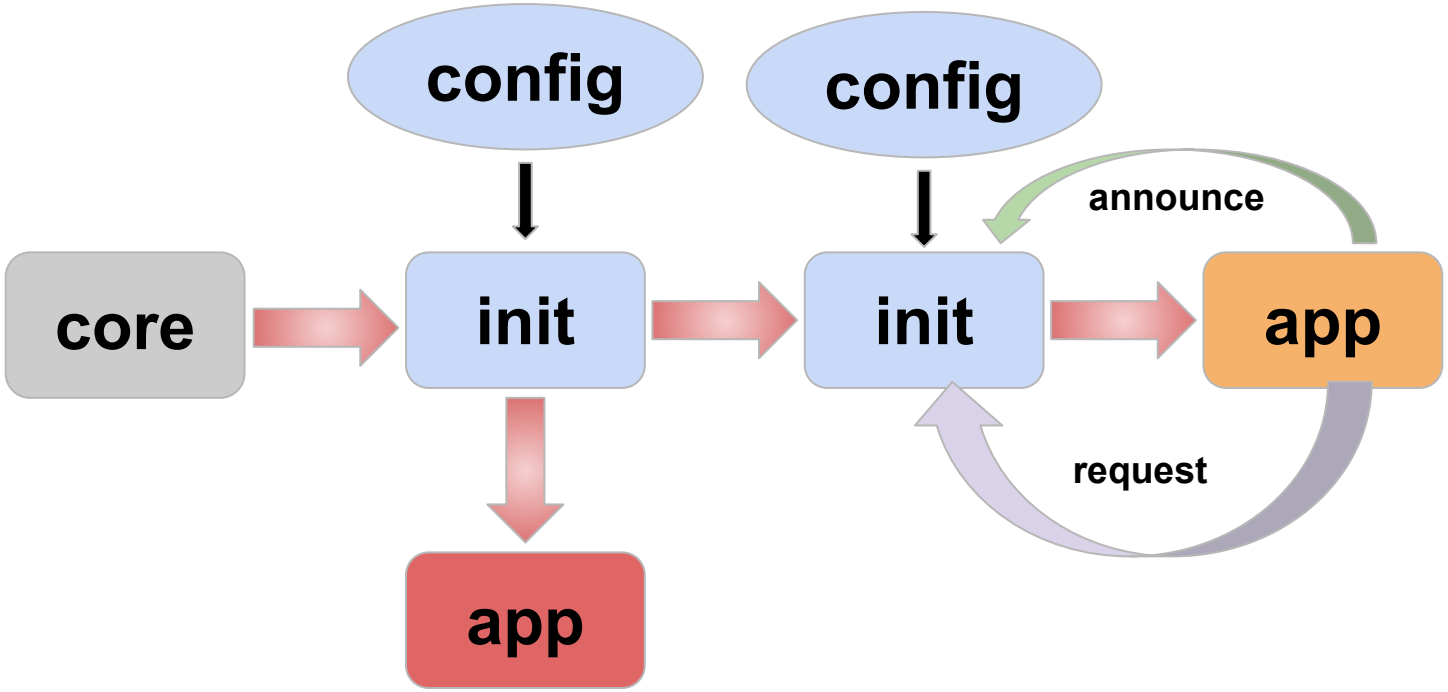
https://genode.org/download/sculpt

Спасибо за внимание!

https://twitter.com/sermp
https://t.me/sermp
https://github.com/sergey-platonov

MQTT gateway




config
config
announce
core
init
init
app
request
app

Capability-based security

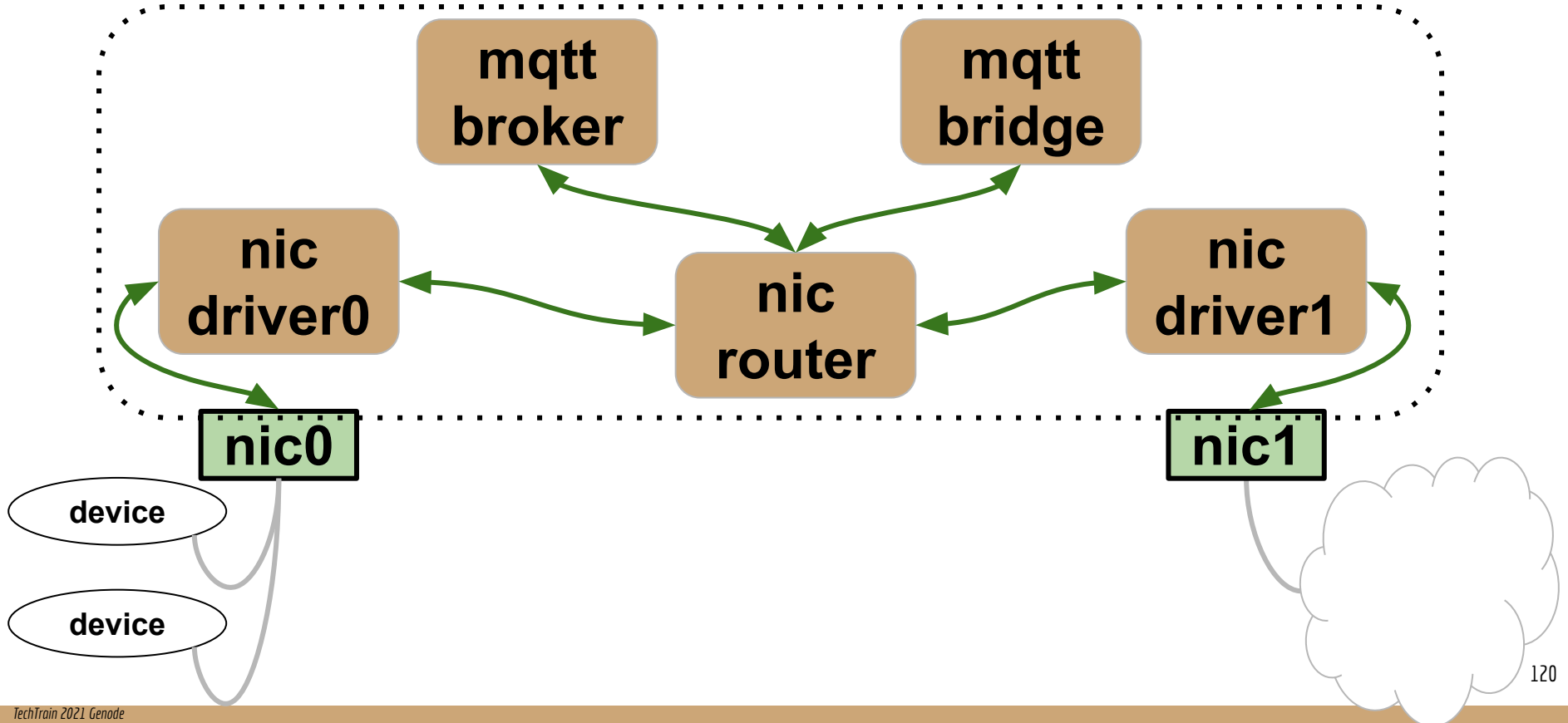
фиксированный набор токенов

токены нужны для доступа к объектам

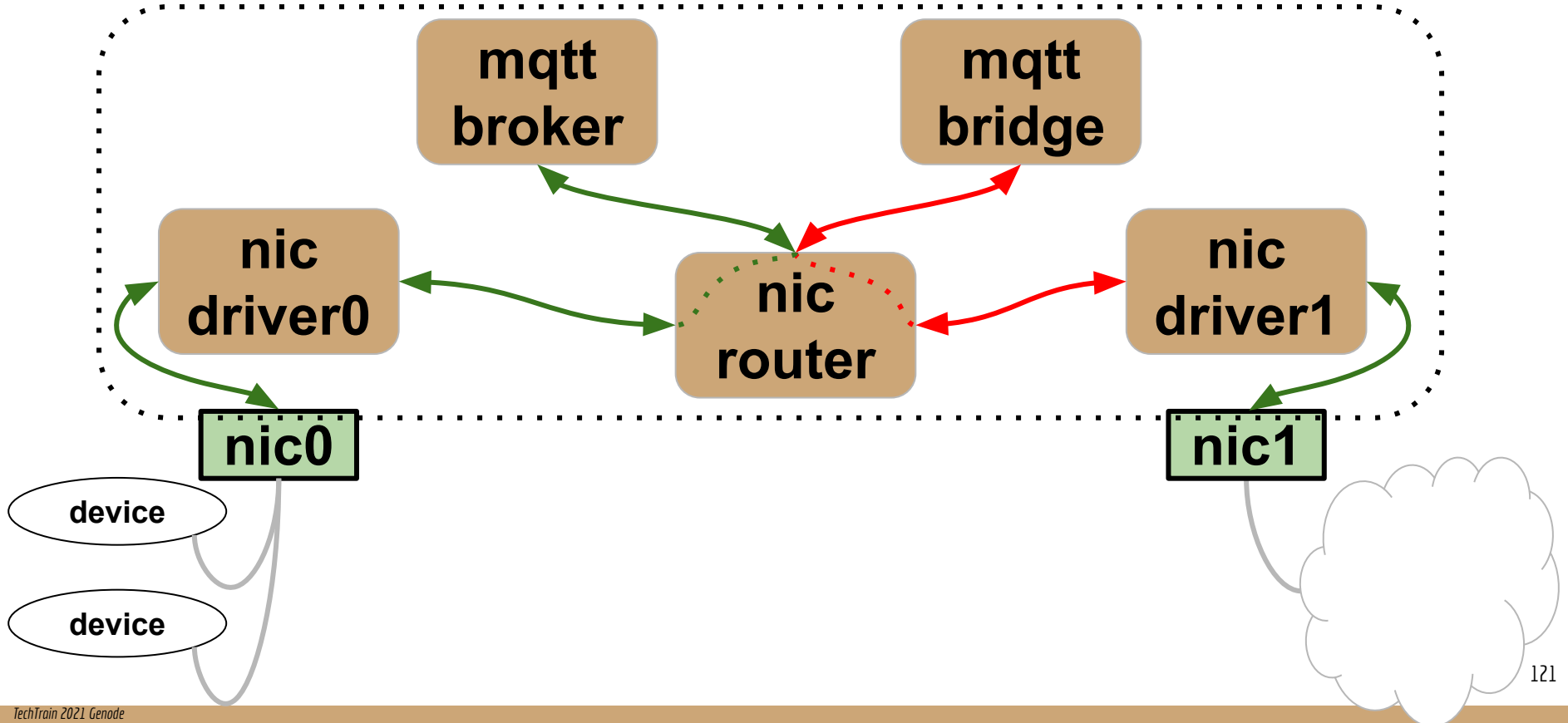
токены можно передавать потомкам

токены можно передавать серверам

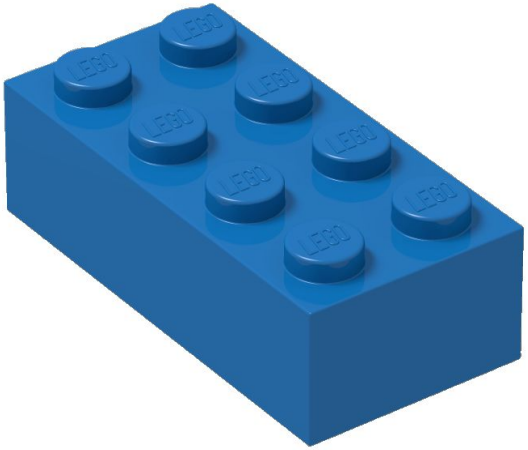
MQTT gateway

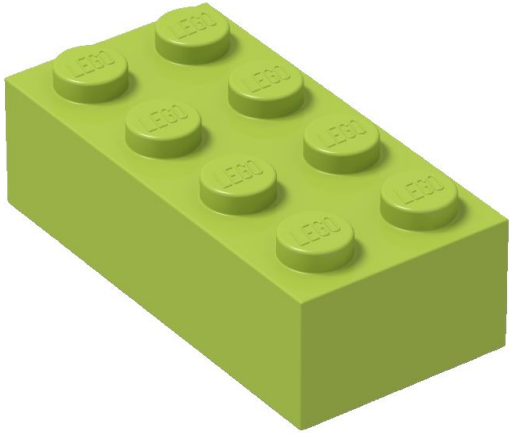


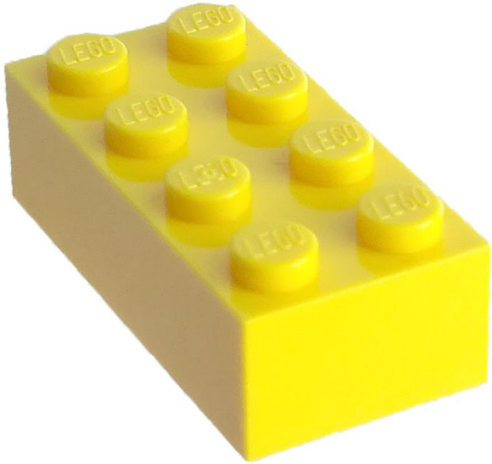
MQTT gateway



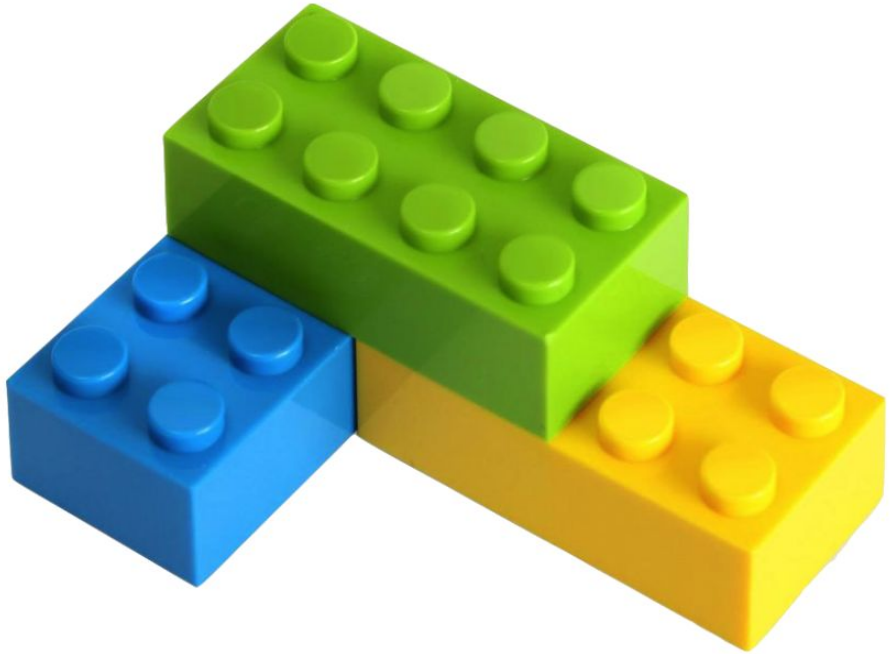
Genode: архитектура



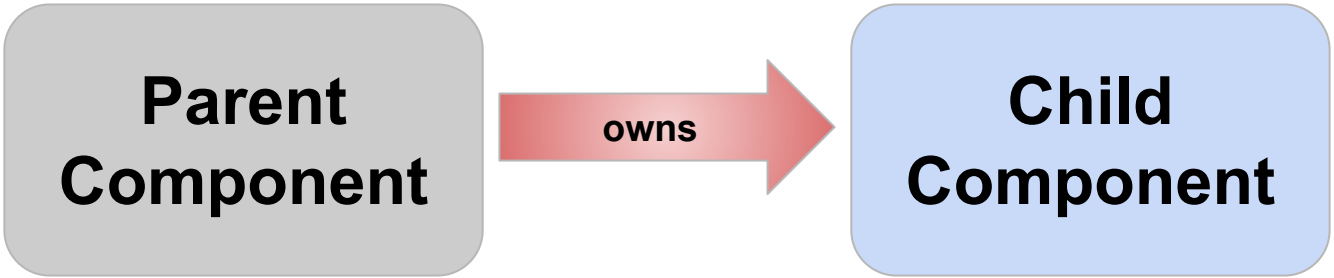




Genode: архитектура



Genode: архитектура



Genode: архитектура