



# QEMU's device model qdev: Where do we go from here?

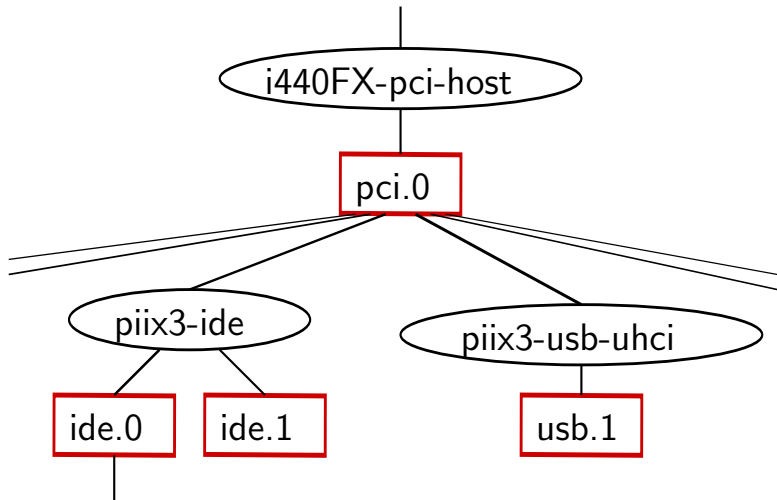
Markus Armbruster  
armbru@redhat.com  
KVM Forum 2011

## What's qdev?

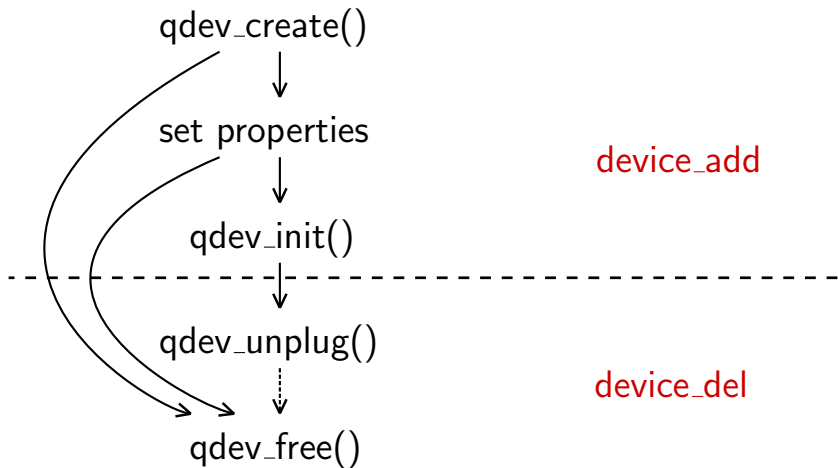
It's QEMU's **Device Model Abstraction**:

- Tree of **devices** connected by **buses**
- Device-independent configuration & control
- Devices implement (bus-specific) API

## A device tree fragment



## Device-independent configuration & control



## Devices implement (bus-specific) API

```
struct SCSIDeviceInfo {
    [...]
    int (*init)(SCSIDevice *dev);
    void (*destroy)(SCSIDevice *s);
    SCSIRequest *(*alloc_req)(SCSIDevice *s, [...]);
    void (*free_req)(SCSIRequest *req);
    int32_t (*send_command)(SCSIRequest *req,
                           uint8_t *buf);
    void (*read_data)(SCSIRequest *req);
    void (*write_data)(SCSIRequest *req);
    void (*cancel_io)(SCSIRequest *req);
    [...]
};
```

## qdev is a success!

- > 250 devices qdevified
- One way to **configure** them: `-device`
- One way to **hot-plug** them: `device_add`
- One way to **hot-unplug** them: `device_del`
- Configuration file: `-readconfig`
- qdevs are **discoverable**

Yay qdev!



## Woe qdev!



- Missing things
- Stupid things
- Difficult things



## qdev is incomplete!



- Still more qdevification
- Anemic buses
- Better IRQ/GPIO binding
- Saner override of default devs
- Canonical dev address
- ...

**qdev is a mess!**



## Oook! Oook! Oook!

 Documentation? What documentation?

 Tree node names are FUBAR

 Many hacks break `-device`

## Oook! Ouch! Oook!



No taste in naming

PCISState, ISADevice, i2c\_slave



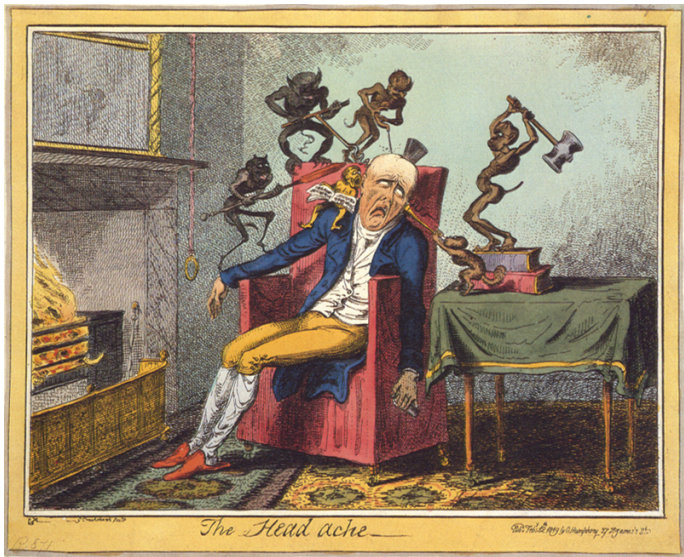
No taste in semantics either

isa\_create(), usb\_create(), i2c\_create\_slave()



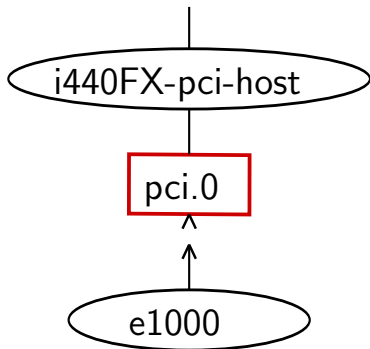
Monkey no talk, monkey hack!

# qdev is hard!



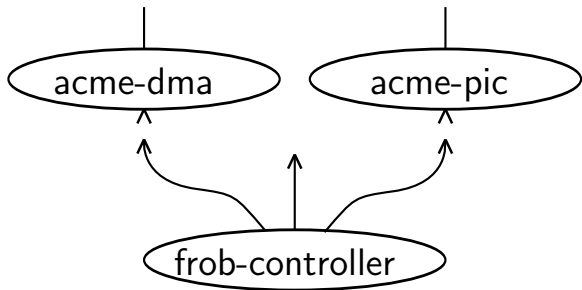
## Connecting devices

- Tree node: just **one parent**
- **Okay** for plugging into a “real” bus



## Connecting devices

Not okay for plugging into many “things”

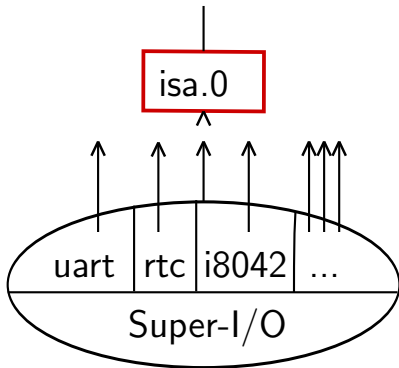


Look ma, no bus!

No -device either...

## Connecting devices

Not okay for plugging many times





## Composing devices

UART can be

- a standalone device on ISA bus
- a component of a Super I/O chip

Need **a way to compose** devices from components

## Observations & Ideas

- Tree is an oversimplification
- Bus is useful, but not everything's a bus
- Generalize: devices have **plugs & sockets**
- Leads to **directed graph**

## Machine description file



- All config must be data
- User plugs in more devices
- Which might be composed
- You think a single file will do?

## qdev needs love!



Promising invention,  
needs work for general use



Questions?