



# Kubernetes: Edge vs. Level Triggered Logic

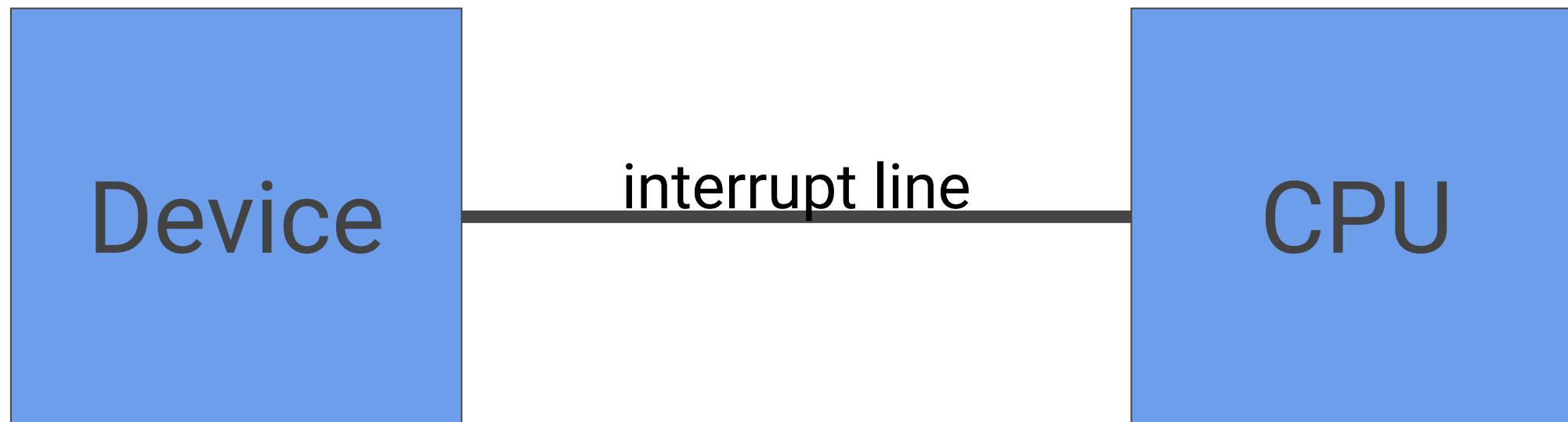


Tim Hockin <[thockin@google.com](mailto:thockin@google.com)>  
Principal Software Engineer  
[@thockin](#)

Once upon a time there was hardware

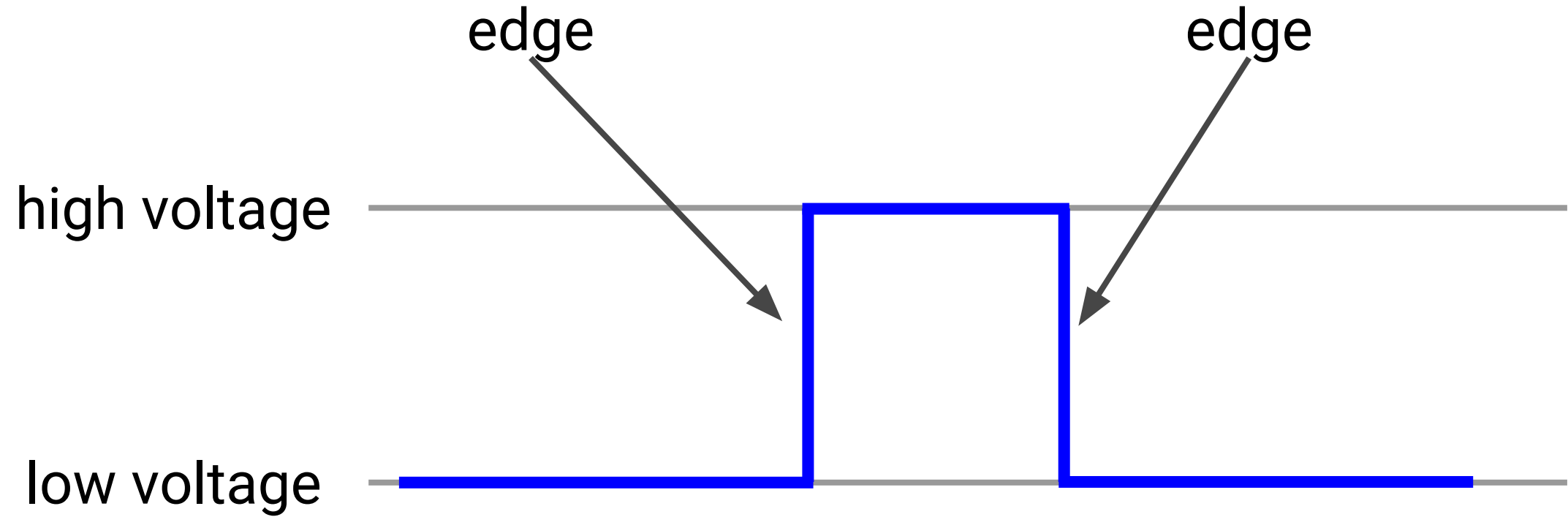
When the hardware needed attention, it would signal the CPU == “interrupt”

Interrupts used be a literal wire from the device to the CPU



The device can change the voltage on the wire

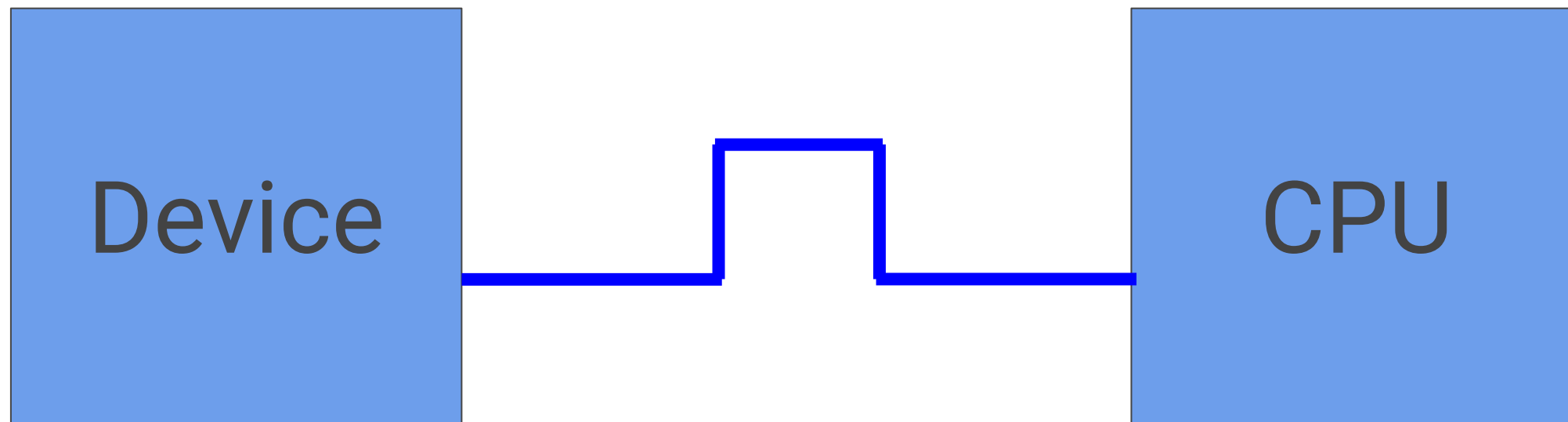
The CPU can detect changes in voltage



Simple: the device “pulses” the interrupt line

The CPU reacts

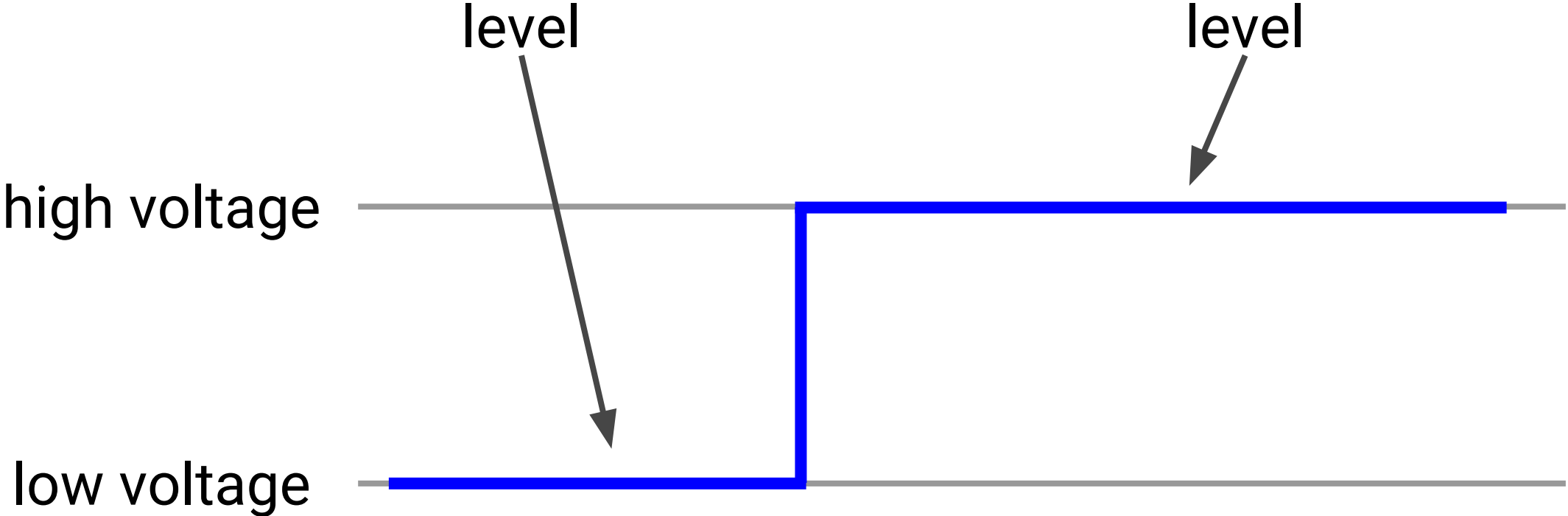
This is “edge triggered”



Problem: what happens if the CPU doesn't detect a pulse?

The interrupt doesn't get serviced!



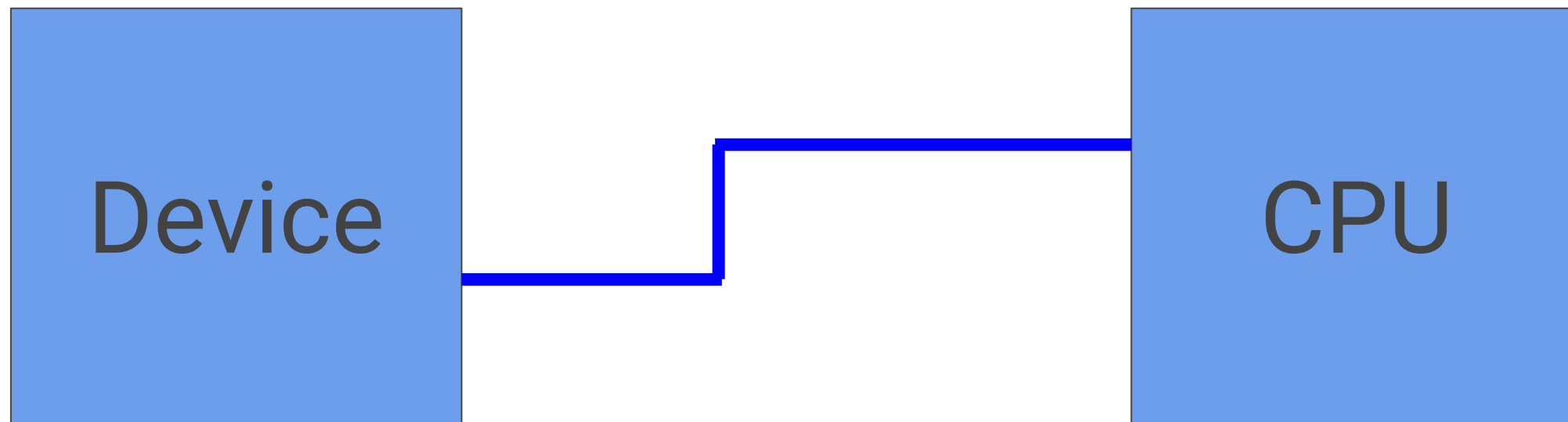


The device can change the voltage and keep it there until the interrupt is serviced

The CPU can't miss edges

The device can “stack” interrupt reasons - leave the voltage until all reasons are ACK'ed

This is “level triggered”



# What does this mean for Kubernetes?

State is more useful than events

Level-driven software is about asserting state

Clients can check and re-check state at any time

This is the heart of Kubernetes' controller model