CHAOS

Paper By: PK Gadepalli, G Peach, G Parmer, J Espy, Z Day
Presentation By: Gregor Peach

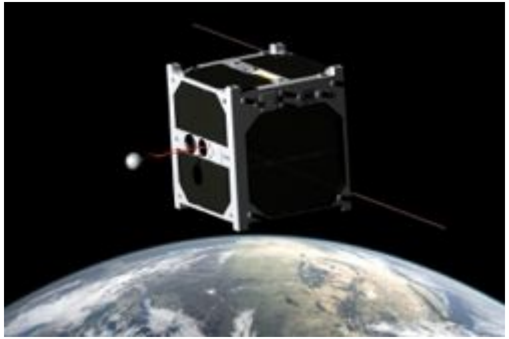
Motivation

Mixed Criticality Systems

- Hear me out, but what if have a system that does

TWO THINGS!!!!

(big wow)



Low and High Criticality

High Criticality

- Important tasks
- Highly tested code

Low Criticality

- Unimportant tasks
- Less tested code

State of the Art: Real-time Systems

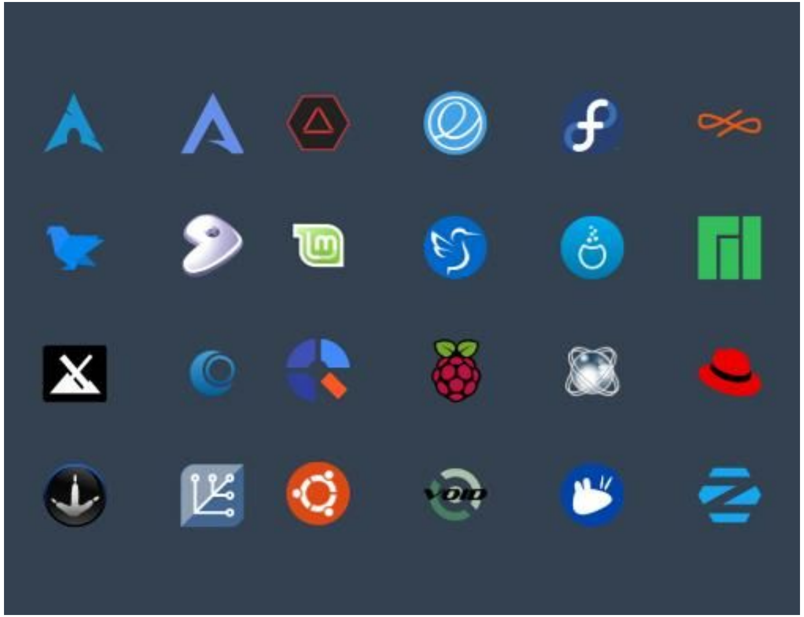
Classic Real-time Operating Systems

- eg. FreeRTOS
- Pros:
  - Predictable execution (real-time)
  - Single core
  - High assurance
- Cons:
  - No multicore
  - No rich POSIX APIs

State of the Art: POSIX

LINUX

- eg. Ubuntu
- Pros:
  - Rich programming environment
  - Good hardware support
  - Real-time patch available
- Cons:
  - Really big and complex
  - Real-time performance terrible
  - Big surface area for attacks



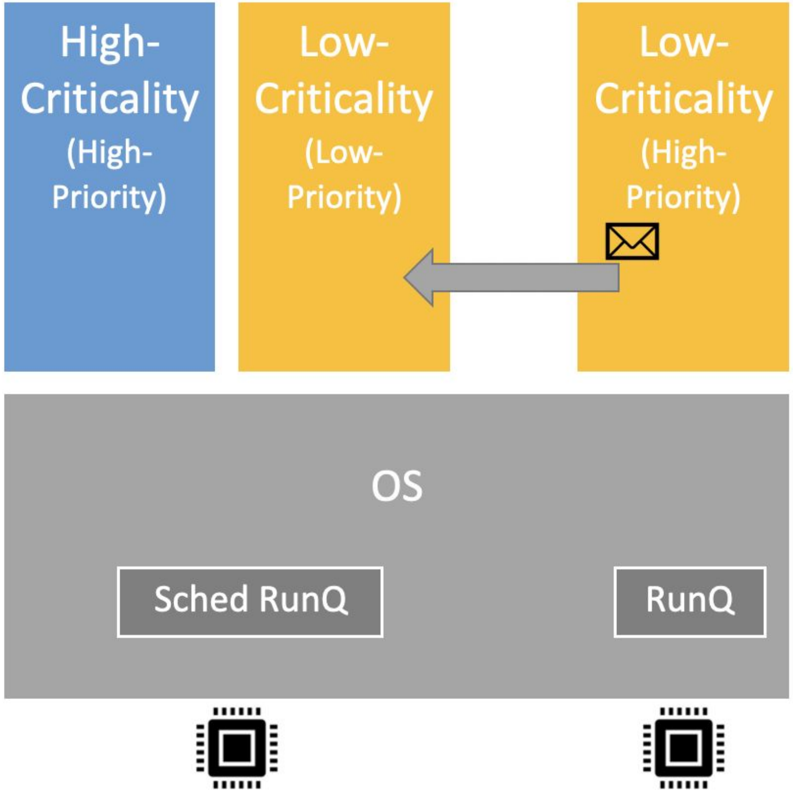
Mixed Criticality Software Lacks a Clear Home

If I have a mixed criticality system, what platform should I build it on?

What are the pros and cons of running it on FreeRTOS?

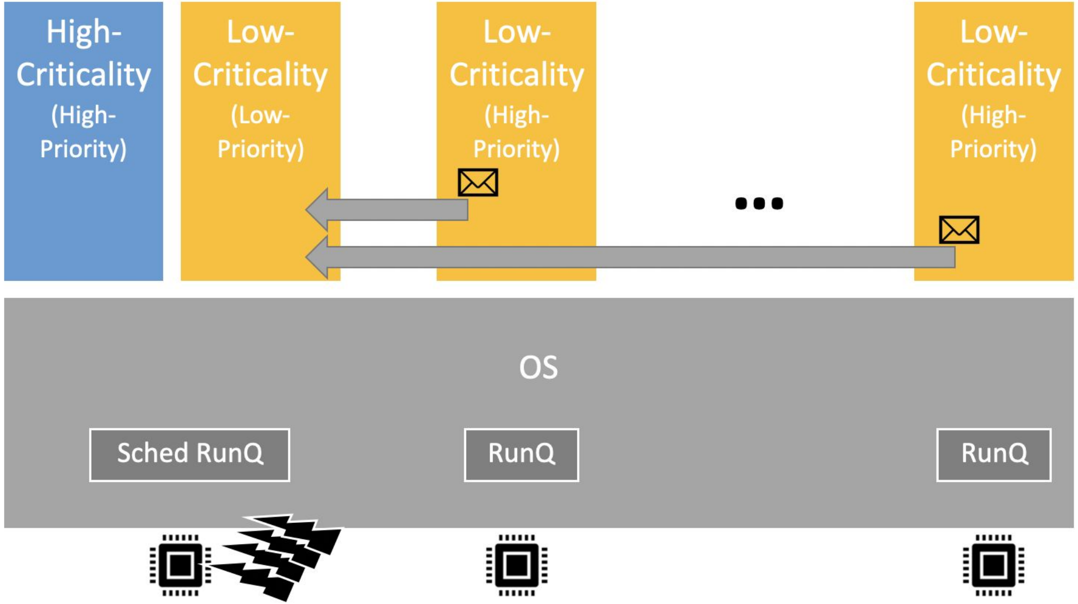
Linux?

This Setup is Unlikely to Work

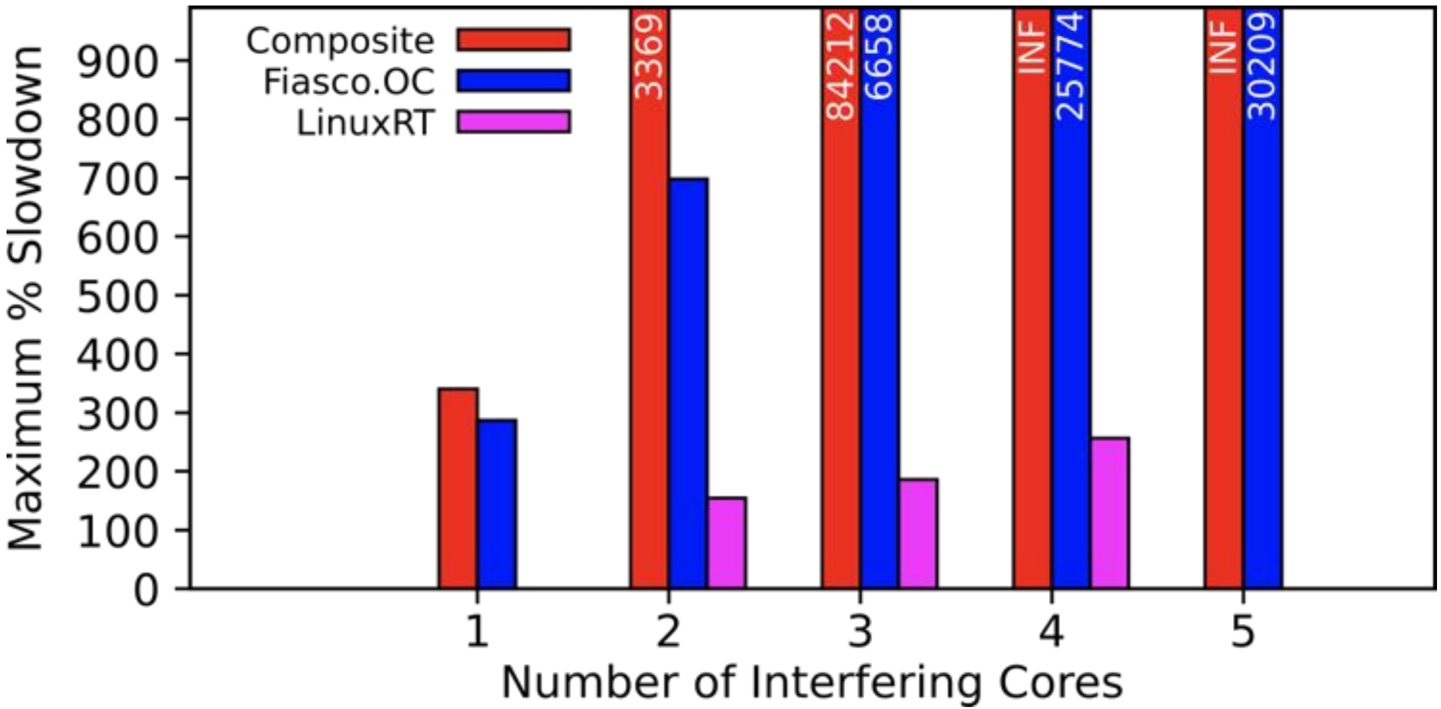


And Even If It Did...

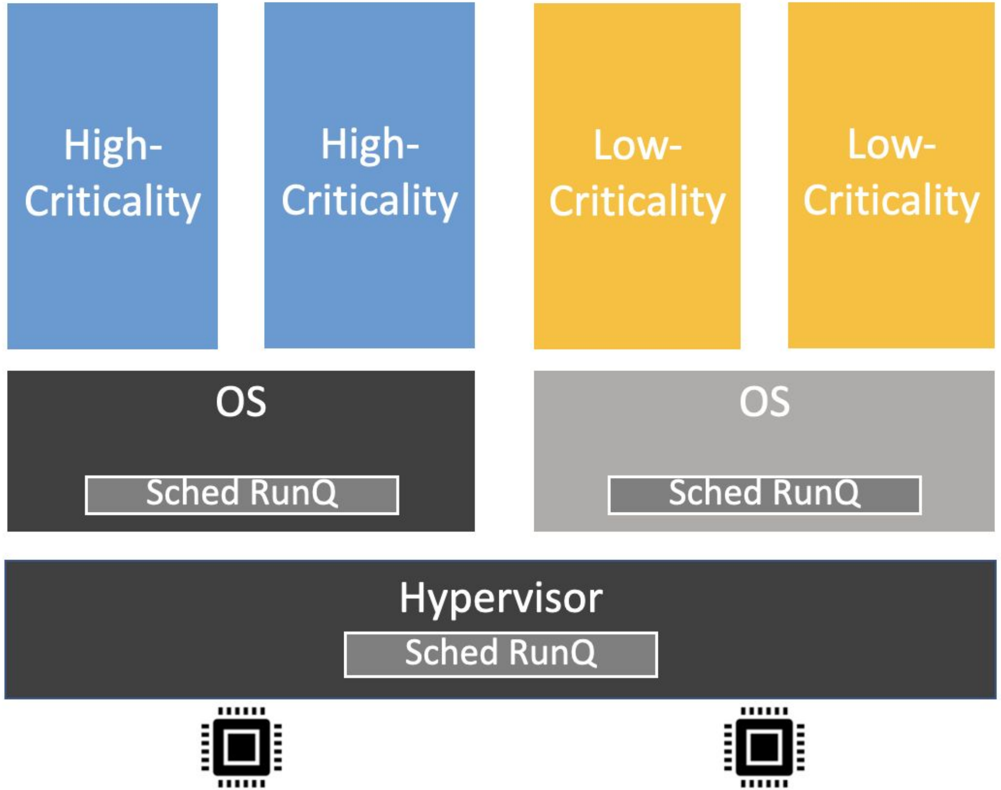
- Naive core sharing has a problem
  - Cross core interrupt denial of service attack


High-Criticality (High-Priority)
Low-Criticality (Low-Priority)
Low-Criticality (High-Priority)
...
Low-Criticality (High-Priority)
OS
Sched RunQ
RunQ
RunQ

Performance Problem!



Solution in Practice: Per Core Virtualization


High-Criticality

High-Criticality

Low-Criticality

Low-Criticality

OS

Sched RunQ

OS

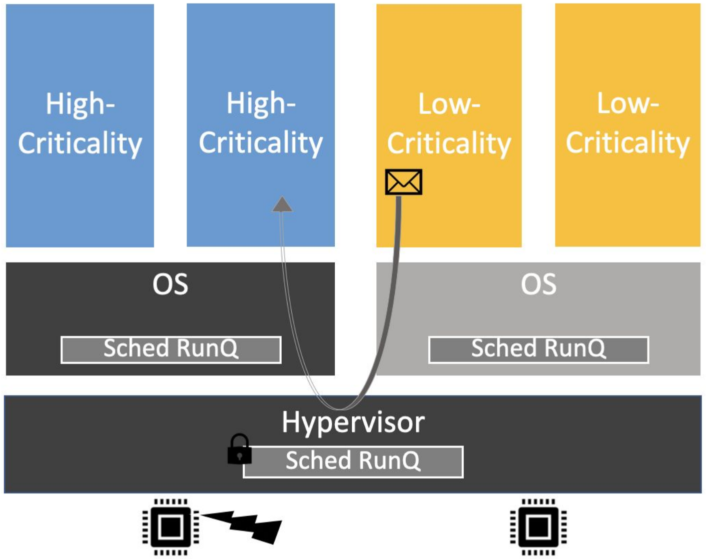
Sched RunQ

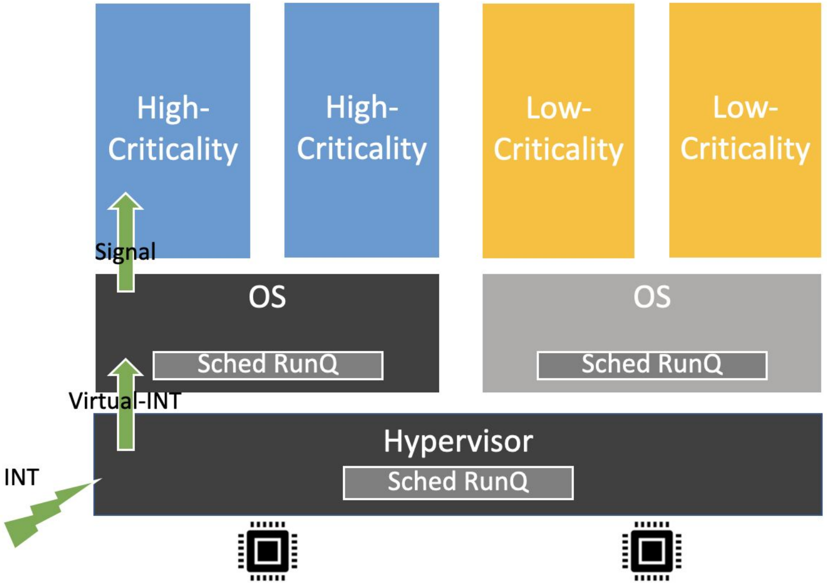
Hypervisor

Sched RunQ

Solution in Practice: Per Core Virtualization

Interrupts handled properly...



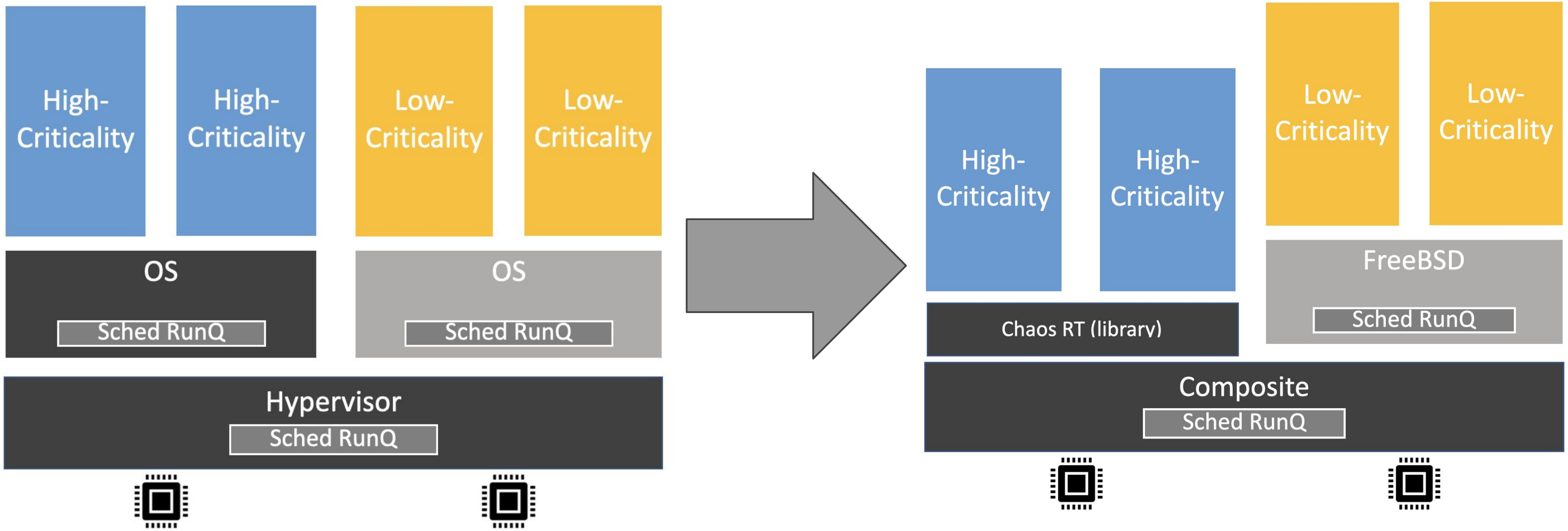


Problems with Per Core Virtualization

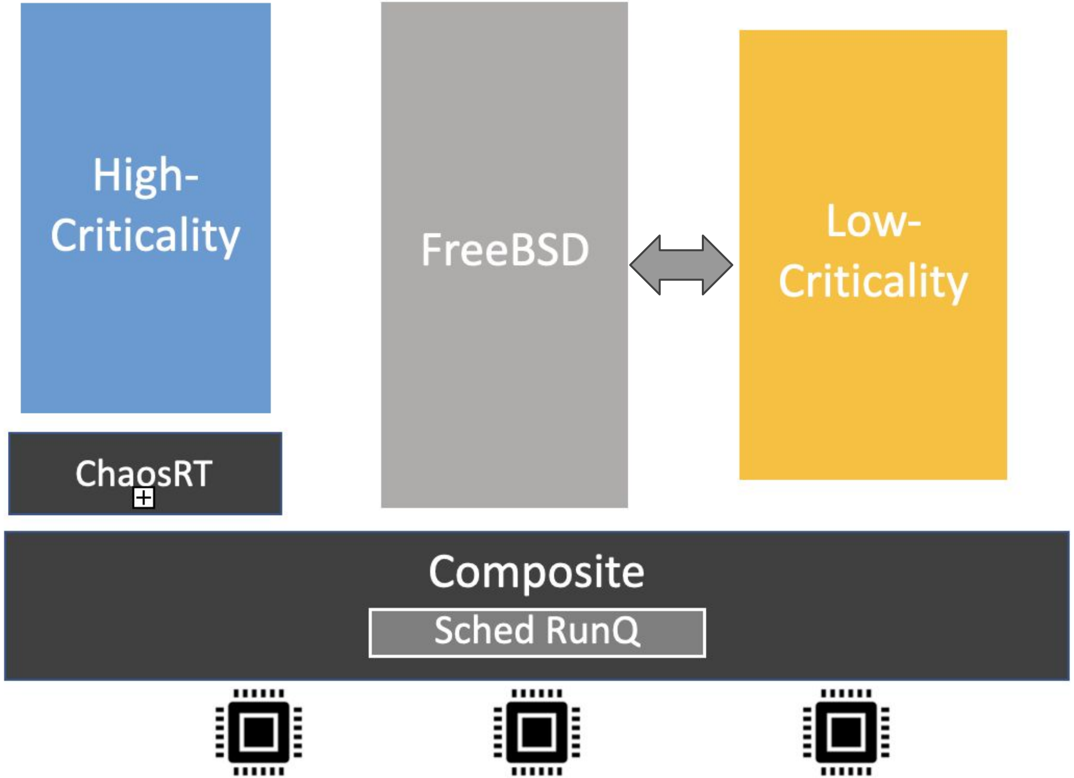
- It’s super slow, big overheads for normal computation
    - Real-time worst case bounds are subject to variance (Hierarchical Overheads)
- Slower to coordinate between high and low assurance

What is CHAOS?

CHAOS: Presenting Devirtualization!!



CHAOS: Presenting Devirtualization!!

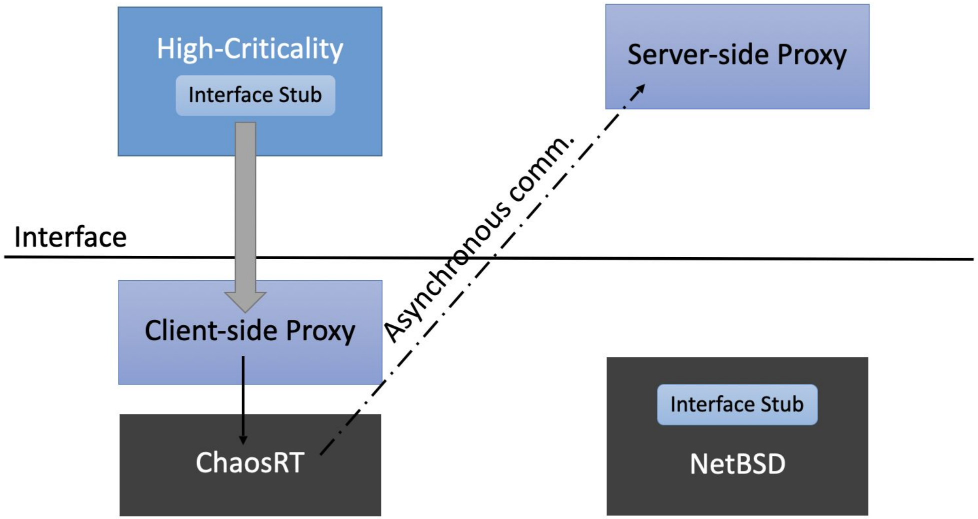


Devirtualization: Get Rid of the Slow VM Layer

- Per Core Virtualization Slow
  - Too many layers of isolation
- Removing it is a win for predictability and performance!
- Works for both low and high criticality code!

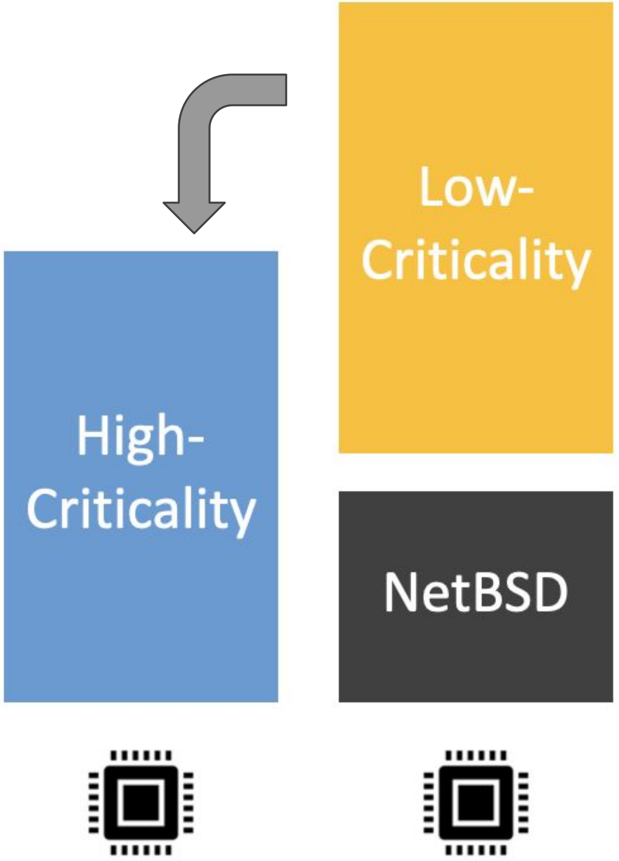
Extension: High Assurance Code uses UNIX APIS

Proxies = mechanism for letting High-Criticality and Low-Criticality processes communicate

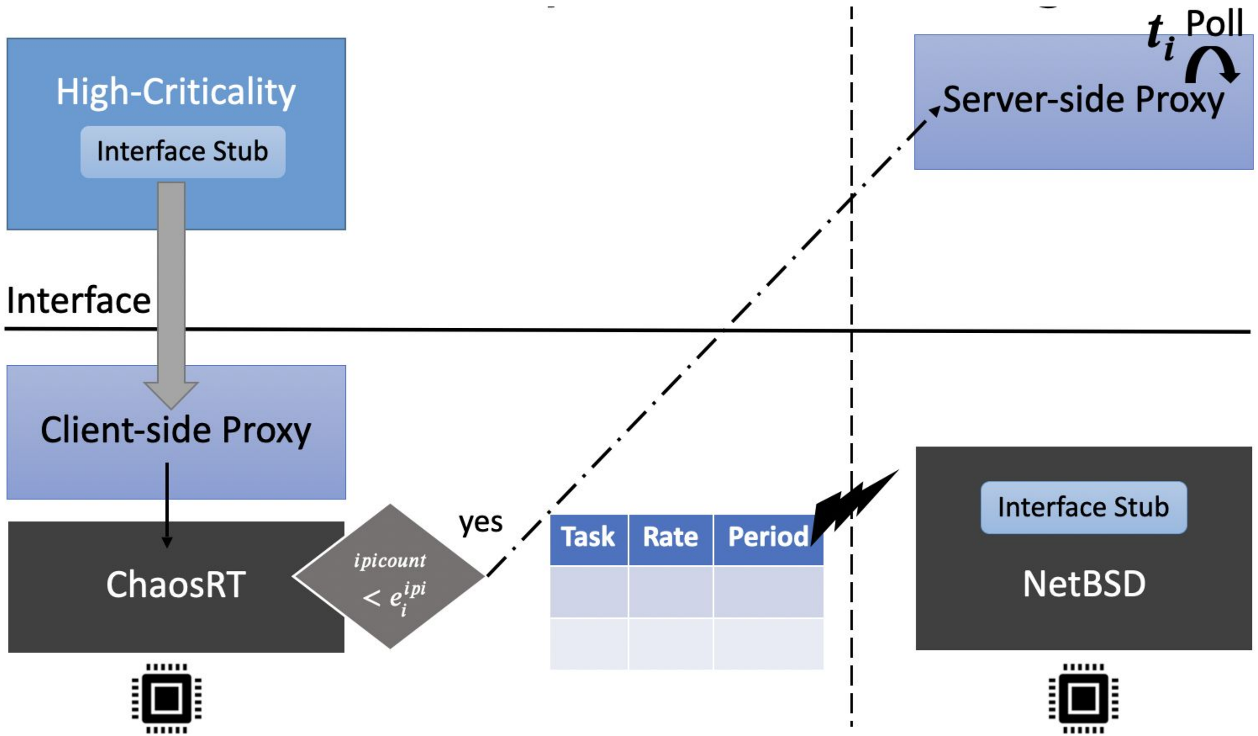


Why Not Talk Directly?

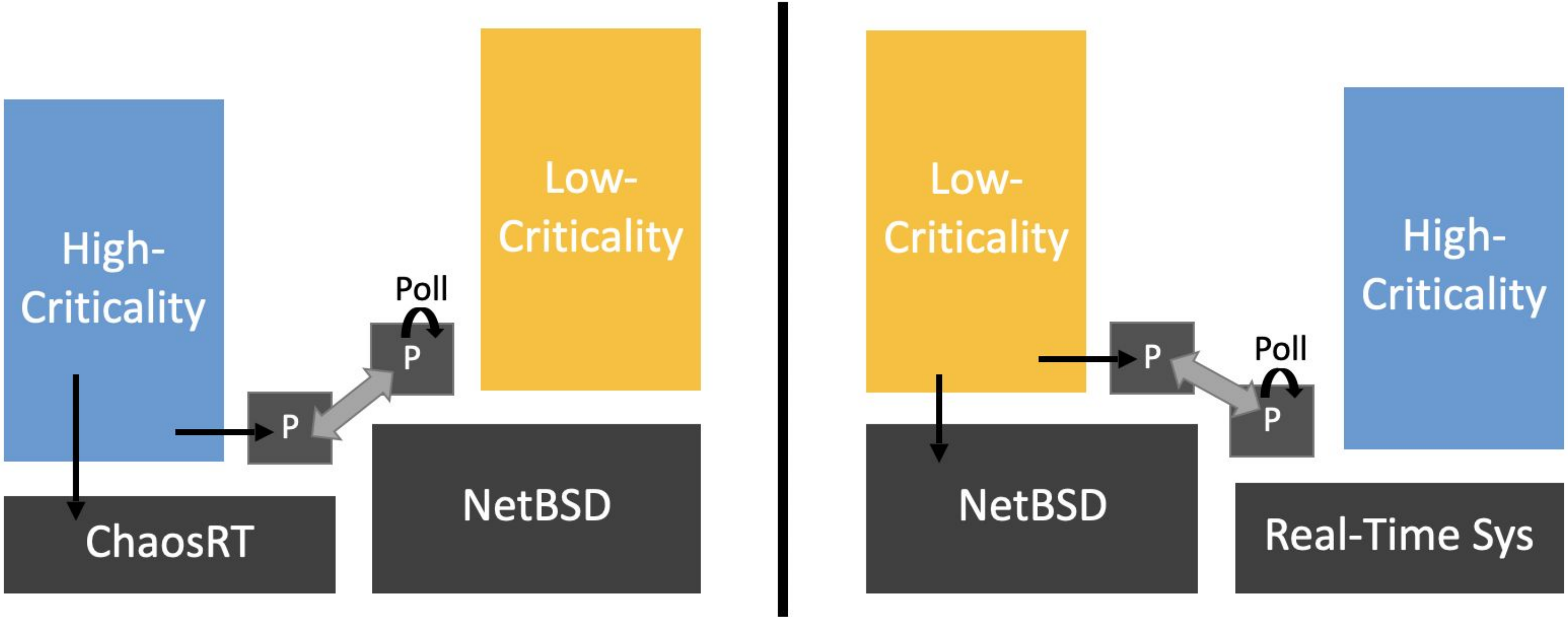
- What might happen here?
- (Recall earlier)



IPI Interference!

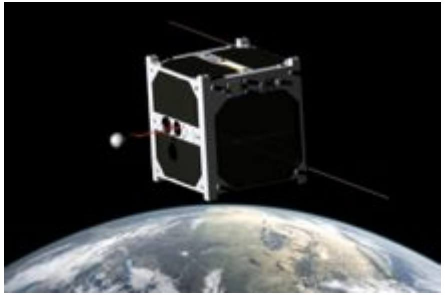


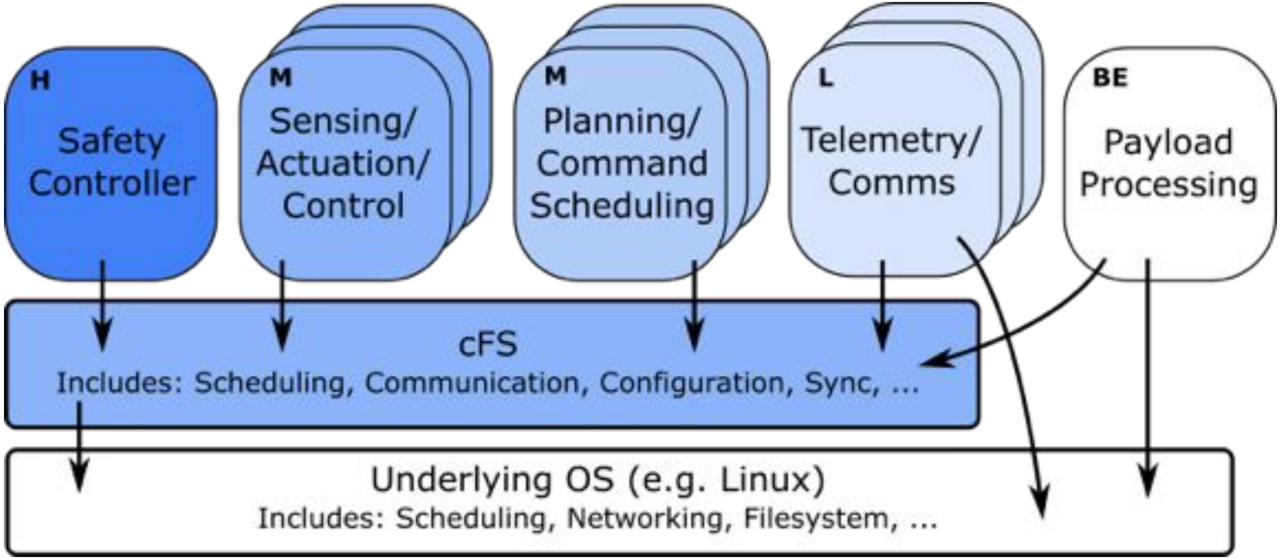
Proxies: Limiting IPI Interference



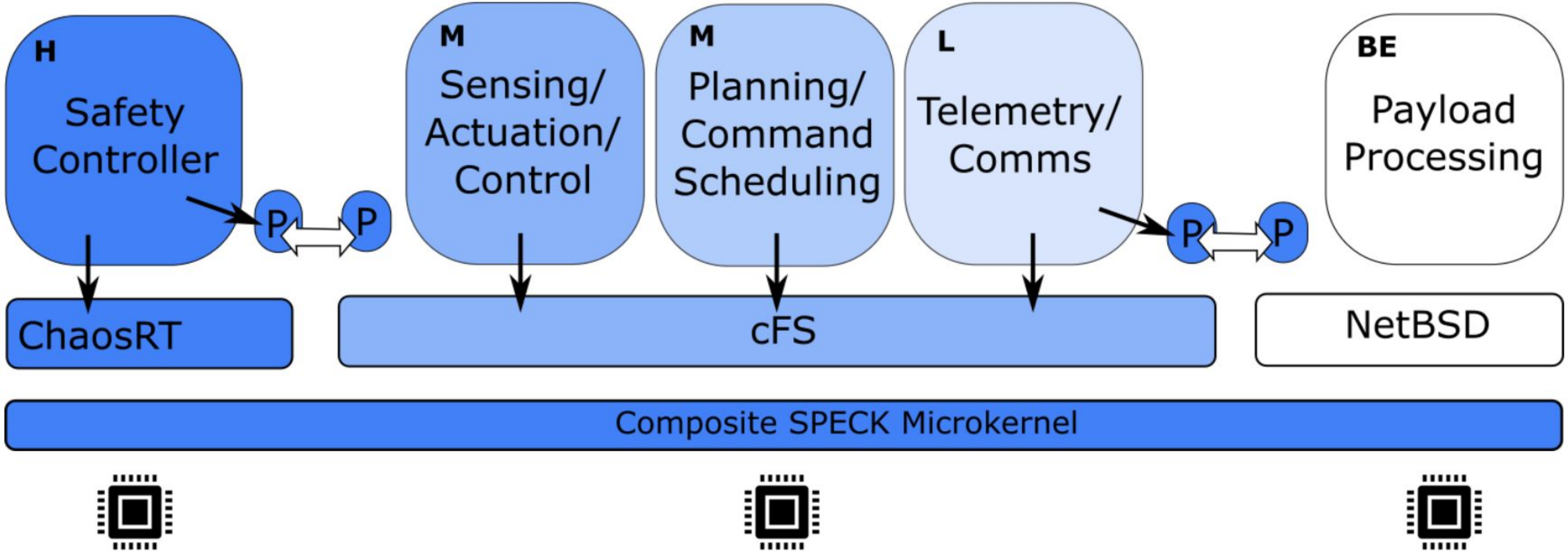
Evaluation

Use Case: The cFE (aka cFS)

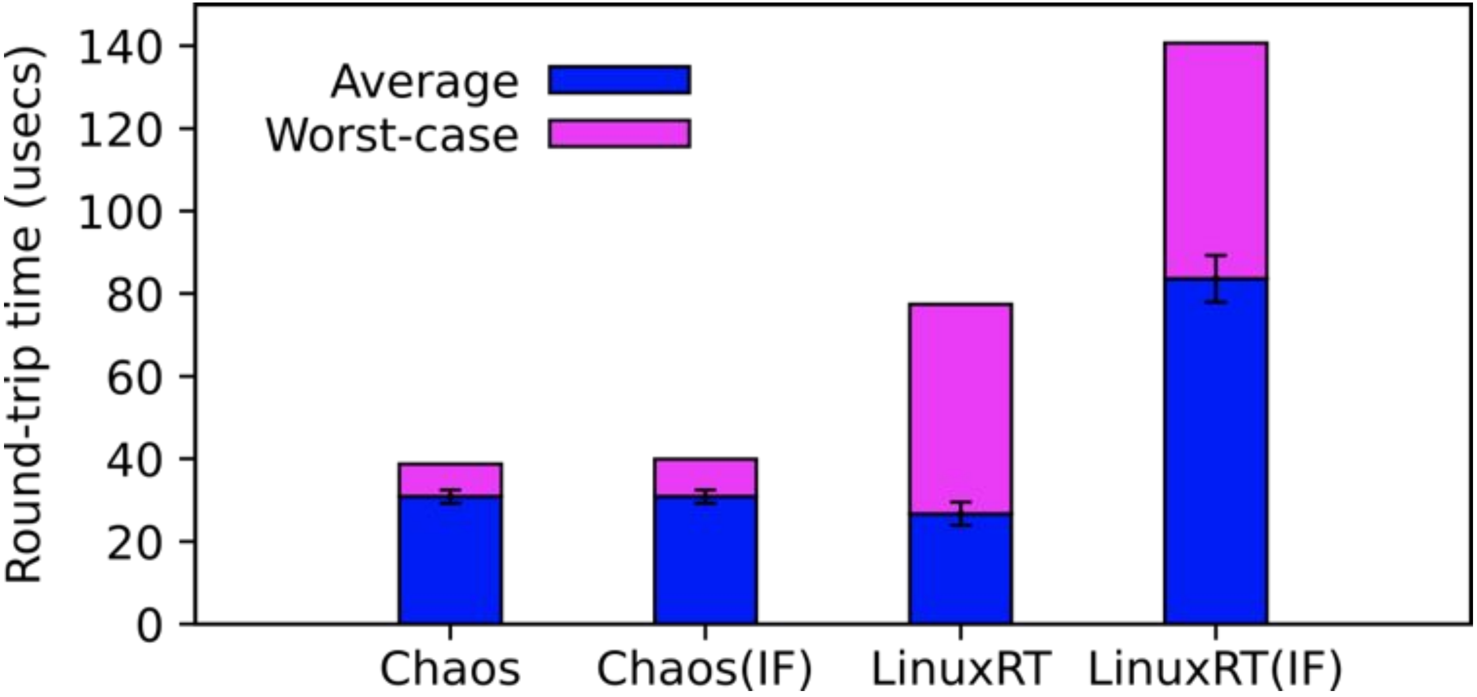



H
Safety Controller
M
Sensing/ Actuation/ Control
M
Planning/ Command Scheduling
L
Telemetry/ Comms
BE
Payload Processing
cFS
Includes: Scheduling, Communication, Configuration, Sync, ...
Underlying OS (e.g. Linux)
Includes: Scheduling, Networking, Filesystem, ...

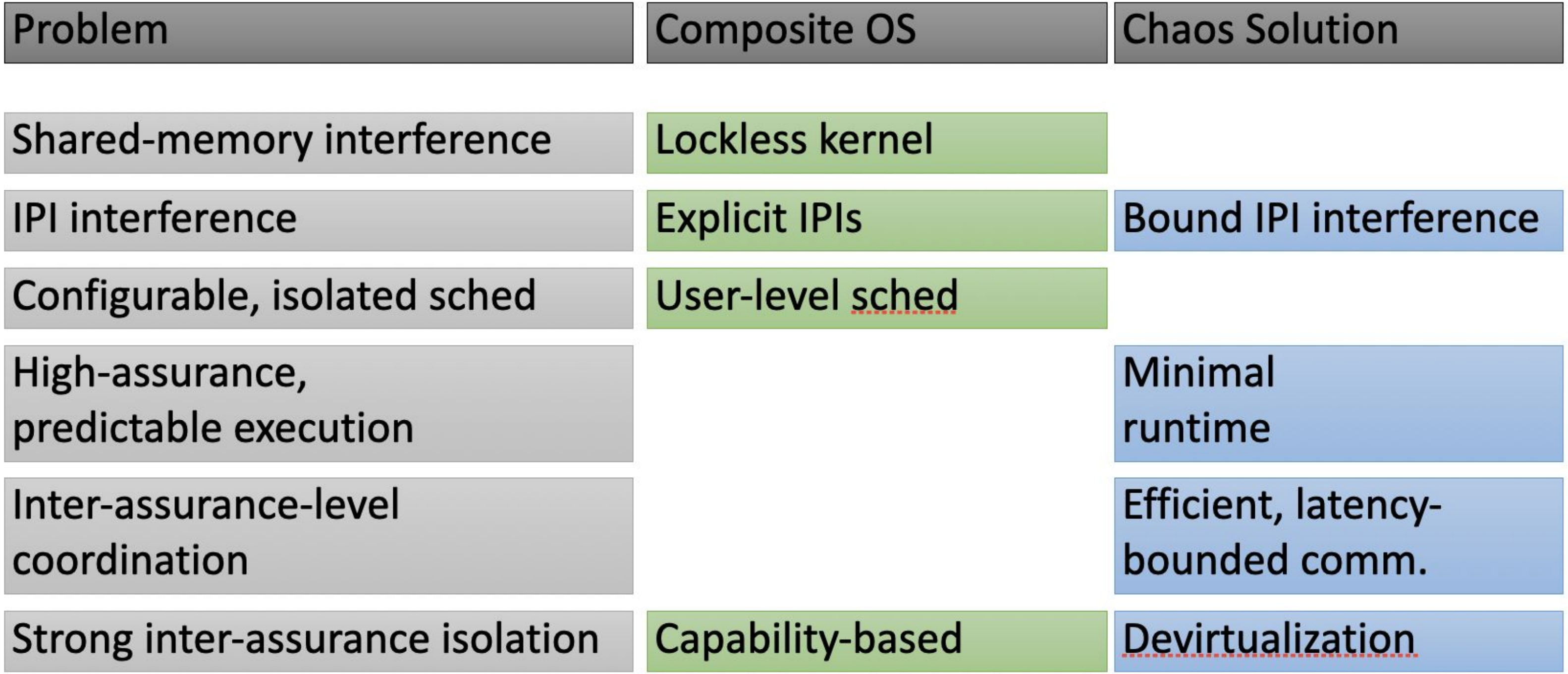
cFE under CHAOS


H
Safety Controller
P
P
M
Sensing/ Actuation/ Control
M
Planning/ Command Scheduling
L
Telemetry/ Comms
P
P
BE
Payload Processing
ChaosRT
cFS
NetBSD
Composite SPECK Microkernel

Worst Case Communication Latency



Summary of Features



Problem	Composite OS	Chaos Solution
Shared-memory interference	Lockless kernel	
IPI interference	Explicit IPIs	Bound IPI interference
Configurable, isolated sched	User-level sched	
High-assurance, predictable execution		Minimal runtime
Inter-assurance-level coordination		Efficient, latency-bounded comm.
Strong inter-assurance isolation	Capability-based	Devirtualization

Critiques

Questions from the Issue

@pcodes: Could this scale to multiple assurance levels (does it even make sense to)?

@anguyen0204: How do TCAPS work?

@rebeccc: IPIs vs shared memory: why not try and resolve the issues of shared memory?

Issues from the Issue

@pcodes: The JpS (jargon per sentence) was too high

   @others: The paper was too dense

@rebeccc: Devirtualization--unclear what this meant in the context of the paper

@Others: The evaluation graphs kinda suck