



DORNERWORKS

seL4 Mainlining

Experiences, challenges,
and solutions

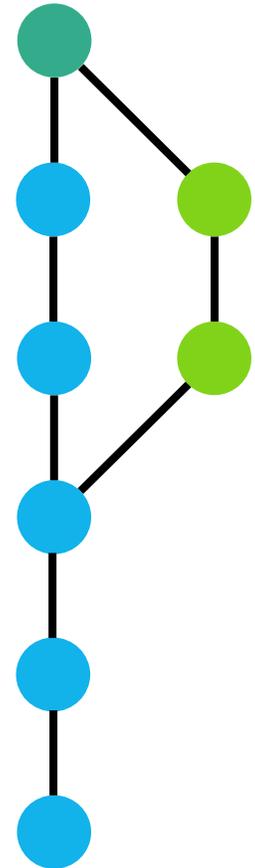
Mainlining Discussion

- Progress and adoption of an open-source project is fueled by community contributions
- Goals
 - Give an experience report of contributing to seL4 source with a focus on successes, challenges, and current status
 - Foster discussion on:
 - Other experiences
 - Improving process



Successes

- Upstreaming new platform support
 - Xilinx Zynq UltraScale+ MPSoC (Zcu102, Ultra96)
 - NXP i.MX8
 - Cubietruck (Allwinner A20)
- ARMv8 virtualization support
- 64-bit x86 virtualization support
- Other smaller features and bug fixes in both kernel-space and user-space



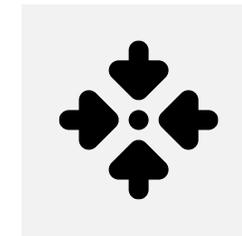


Challenges

Our main challenges come about due to prohibitive processes and restrictions of the real world



Customers



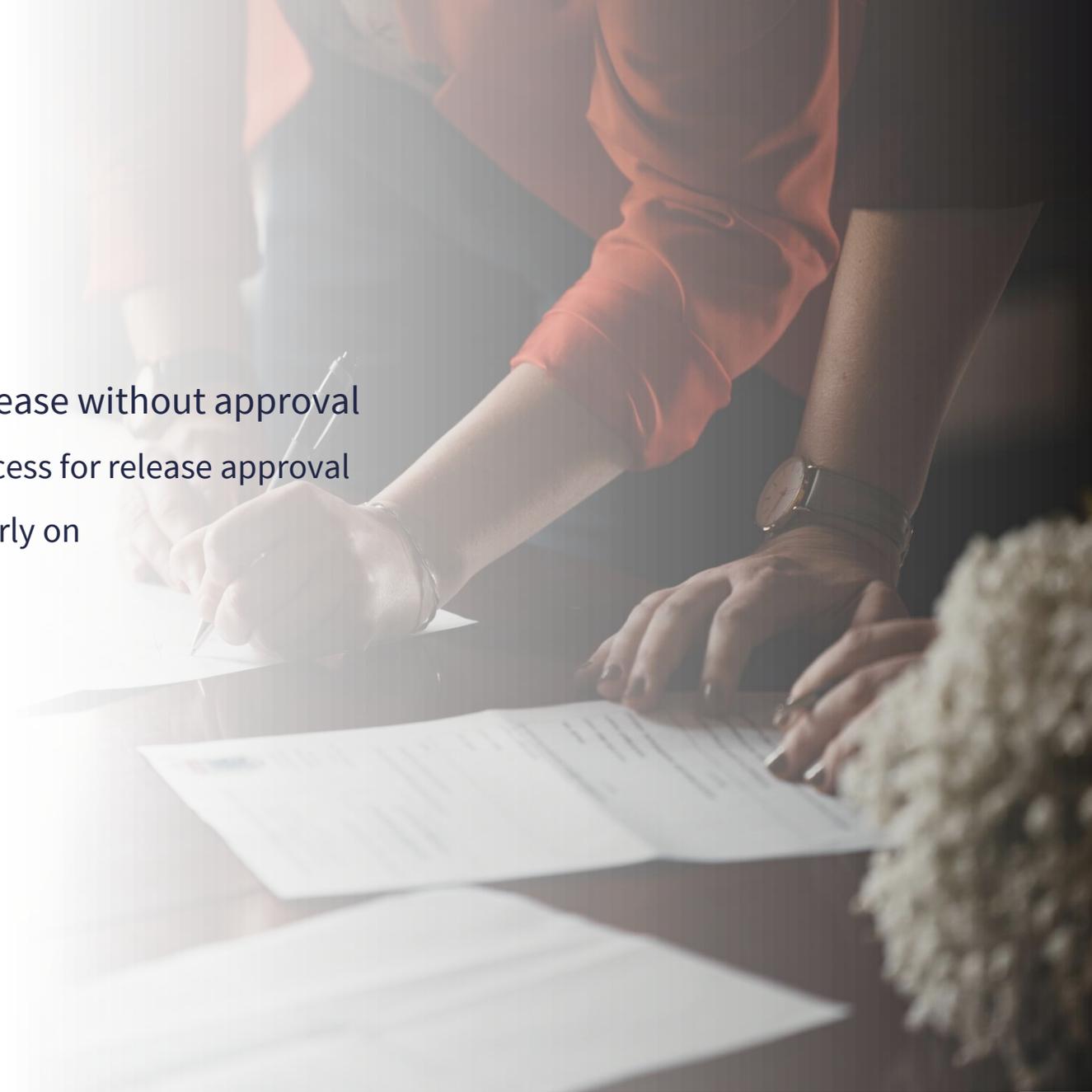
Internal



External

Customer Challenges

- We are beholden to our customers
- Some customers default to export controlled or no release without approval
 - These cases require following the customer's specific process for release approval
 - Prohibits us from collaborating with the foundation early on
 - Can take months!
- Customer IP
 - Can limit the specific features we can release



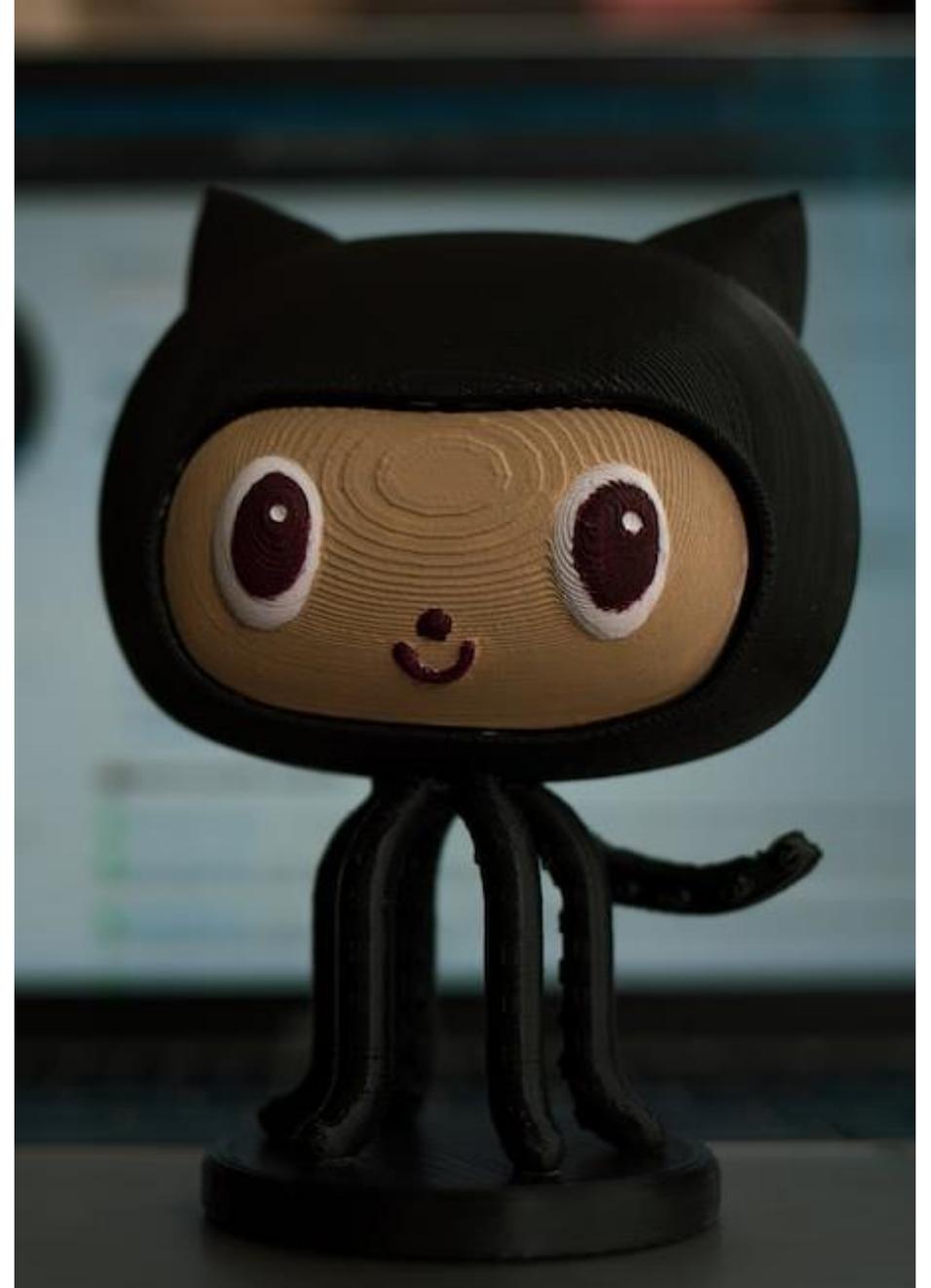
Internal Challenges

- Open sourcing is often our lowest priority for a project
- Tend to open @ completion of project instead of @ completion of each feature
 - Sometimes this makes more sense when we need to follow customer processes, but overall may make it more difficult
 - Further out of sync
 - Bigger patches
 - Fully completing a new feature before any discussion with community
- Not always adding new tests alongside new code

DORNERWORKS

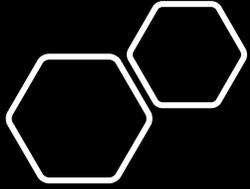
External Challenges

- Has been a lack in response time for PRs
 - This has gotten much better with the introduction of the Foundation
- Denials for dependent patches
 - Ex: zynqmp support in camkes-vm because SMC calls needed to be enabled
- RFCs falling off the radar



Current state

- Due to the combination of all these challenges
 - DornerWorks has **~100 commits** that we want to open source, but are having trouble getting past roadblocks
 - Necessary to periodically rebase all these changes on top of mainline to stay up to date
 - Hoping to push RFCs forward and break up our commits into smaller, easily integrated patches
 - Does take some effort to do this
- Trying to improve our internal issues as I have most control over those
 - Some changes are hard to make without being more synchronized with the mainlines



Discussion Ideas

- Thoughts on external improvements?
- Thoughts on how contributors can get involved with maintainers sooner?
 - Maybe baking RFC submission into contracts at the beginning of projects?
- Have others faced similar issues? Solved them?