Generic Kernel Image
Linux Plumbers Conference 2021
Android Microconference
GKI Architecture

- Android Framework
- HAL Implementation
- Generic Kernel
- GKI Modules
- Stable Interfaces
- Unstable Interfaces
- Stable Kernel Module Interface (KMI)
- Vendor Modules
- AOSP
GKI Deployment

Android 11 (2020) -- 5.4
- Devices continue to ship with device kernel
- Android compliance testing must be run with GKI kernel in addition to the product kernel

Android 12 (2021) -- 5.10
- Devices ship with GKI boot image
- Regular binary releases available publicly on ci.android.com
- All testing done with GKI
  - Tests check for *certified* image
Main Difference vs Upstream: Out-of-Tree Modules

- Vendor Hooks to modify default behavior of Generic Kernel
  - *Regular* hooks are the same as TRACEPOINTS
  - *Restricted* hooks can execute handlers in non-atomic context but cannot be detached
- Additional exports via `EXPORT_SYMBOL_GPL()` to enable vendor hooks
- Kconfig options to enable hidden configs on behalf of out-of-tree modules

In *android12-5.10* kernel:
- 194 *Regular* hooks and 107 *Restricted* hooks
Partner Value-Added Features in Generic Kernel

Many partner value-adds can be implemented using Vendor Hooks

○ In Core Kernel:

```c
int ret = 1;
trace_android_vh_my_feature(&ret, &task);
if (!ret) return;
```

○ In Vendor Module:

○ init:

```
rc = register_trace_android_vh_my_feature(my_feature, NULL);
```

○ handler:

```
void my_feature(int *ret, struct task_struct *task) { *ret=0; }
```

Complex cases may require the actual code in ACK

○ Code potentially impacts all vendors
The Future: Increased Alignment with Upstream

● 2020-2022: Accumulating ecosystem technical debt in Android Common Kernels (android12-5.10 and android-mainline)
  ○ Features that were in Vendor or OEM kernels are now
    ■ Upstreamed, OR
    ■ Isolated into vendor modules (possibly via hooks), OR
    ■ Merged into Android Common Kernel
  ○ Debt is visible for android-mainline as quilt series: https://android.googlesource.com/kernel/common-patches/

● 2023-2024: Reducing Technical Debt
  ○ *Upstream First* Development model for new features
  ○ Work toward upstreaming all out-of-tree patches in Android Common Kernels