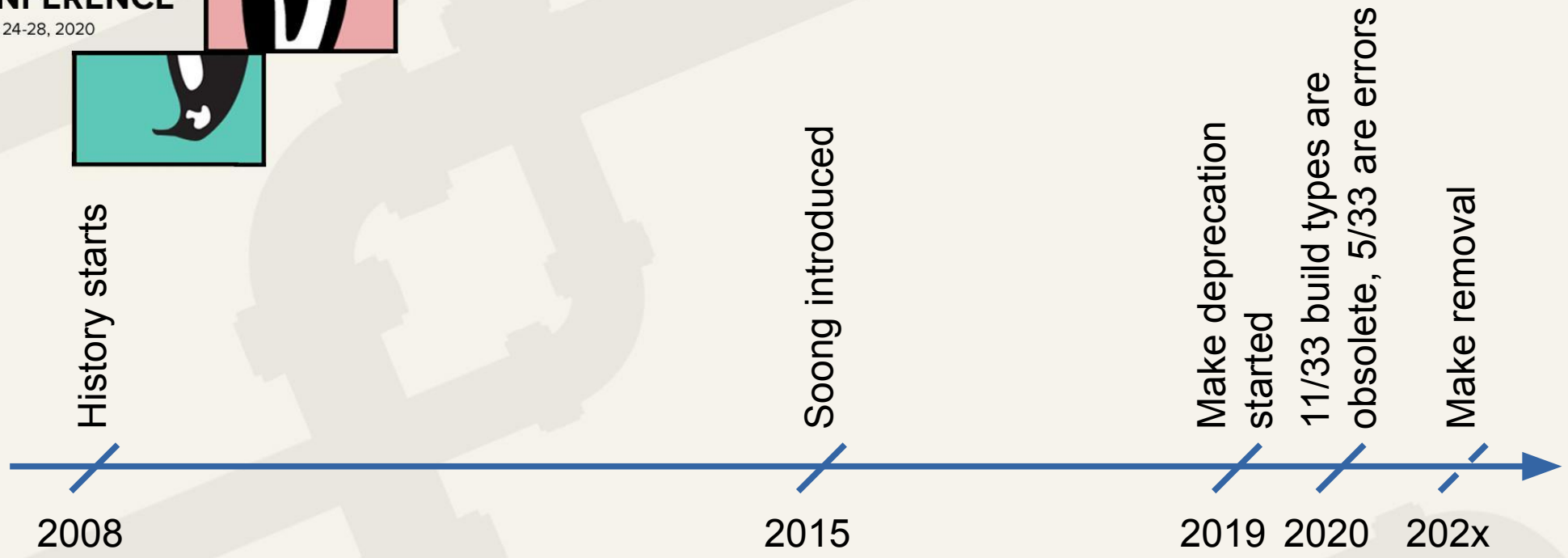
Integrating open source packages into AOSP

Laurent Pinchart (Ideas on Board Oy)
laurent.pinchart@ideasonboard.com

Karim Yaghmour (Opersys Inc.)
karim.yaghmour@opersys.com

John Stultz (Linaro)
john.stultz@linaro.org

Android Build System History



Make-Based Builds

- Based on GNU make with **TONS** of Android rules
- Android.mk in addition of package's native build system (may share code with makefiles)
- Mostly invoke Android rules, but anything is possible (even wrapping the native build)

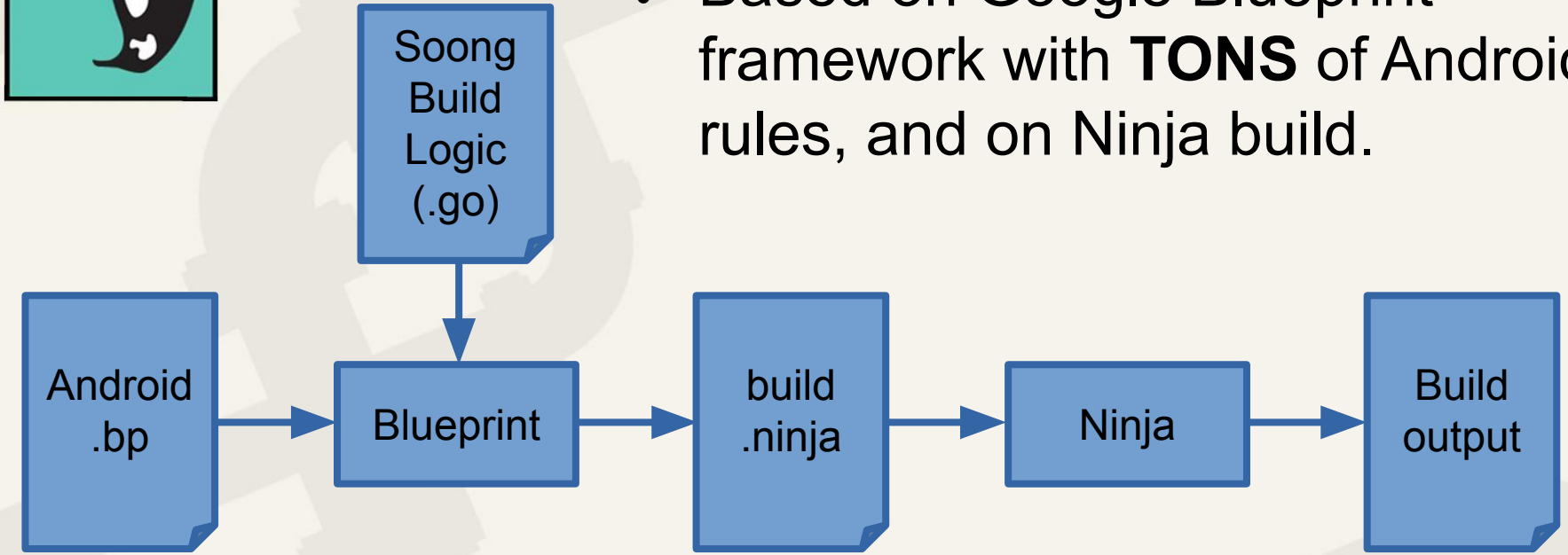
Flexible ☺ / Proven Tools ☺ / Code duplication ☹ / Unpredictable ☹ / Slow ☹

Soong Build System

“The build logic is written in Go using the blueprint framework. Build logic receives module definitions parsed into Go structures using reflection and produces build rules. The build rules are collected by blueprint and written to a ninja build file.”

Soong Build System

- Based on Google Blueprint framework with **TONS** of Android rules, and on Ninja build.



Soong-Based Builds

- Android.bp in addition of package's native build system (no code sharing)
- Extensible with custom build rules in Go
- No easy way to wrap the native build (and will likely not be accepted in AOSP)
- Lack of bp version handling (no fwd/backwards compatibility)

Fast ☺ / Predictable ☺ / Code duplication ☹ / Custom rules in Go ☹

Open-Source Packages

- Wide variety of build systems, some of which share the philosophy of Soong (GYP, GN, Meson, …)
- Wide variety of build time dependencies
- Traditionally packaged in distributions, as opposed to all build in one go

Direction Mesa is likely going

Upstream is pushing to drop Android.mk, require Mesa to be built separately out of AOSP and include Mesa via **binary blobs** (similar to proprietary graphics blobs).

Not unlike how the Linux Kernel is handled.

Eases Mesa maintenance ☺ / Avoids build duplication ☺ / Separate build envs ☹ / Pain for testing ☹

What can be improved?

- Wrapping (some of the) native build systems with custom Go rules? e.g. Meson frontend in Go
- List of approved build time dependencies, maintained with the community? e.g. supported Python packages
- Other ideas?

What else can be improved?

- Helping building packages out-of-tree with their native build system? i.e. standard glue to point Meson to AOSP toolchain, headers and libs
- A manifest for each Android release to list recommended versions of out-of-tree packages?
- Other ideas?