Open Source Software

"EAS for Android" & mainlining update

Linaro Connect Vancouver 2018

Dietmar Eggemann

20-September-2018

EAS for Android - updated development process

Previous EAS development

- Quarterly releases r1.3, r1.4, r1.5 etc.
- EAS product codeline in Android Common Kernel (ACK)
- Development ran behind LTS kernel

2018- EAS for Android

- Yearly major release aligned with LTS
- Some EAS building blocks in mainline kernel
- **Minimal delta** from mainline kernel (including EAS building blocks on tip/sched/core and LKML)
- Arm's development/testing process closely aligned with Linux mainline

arm

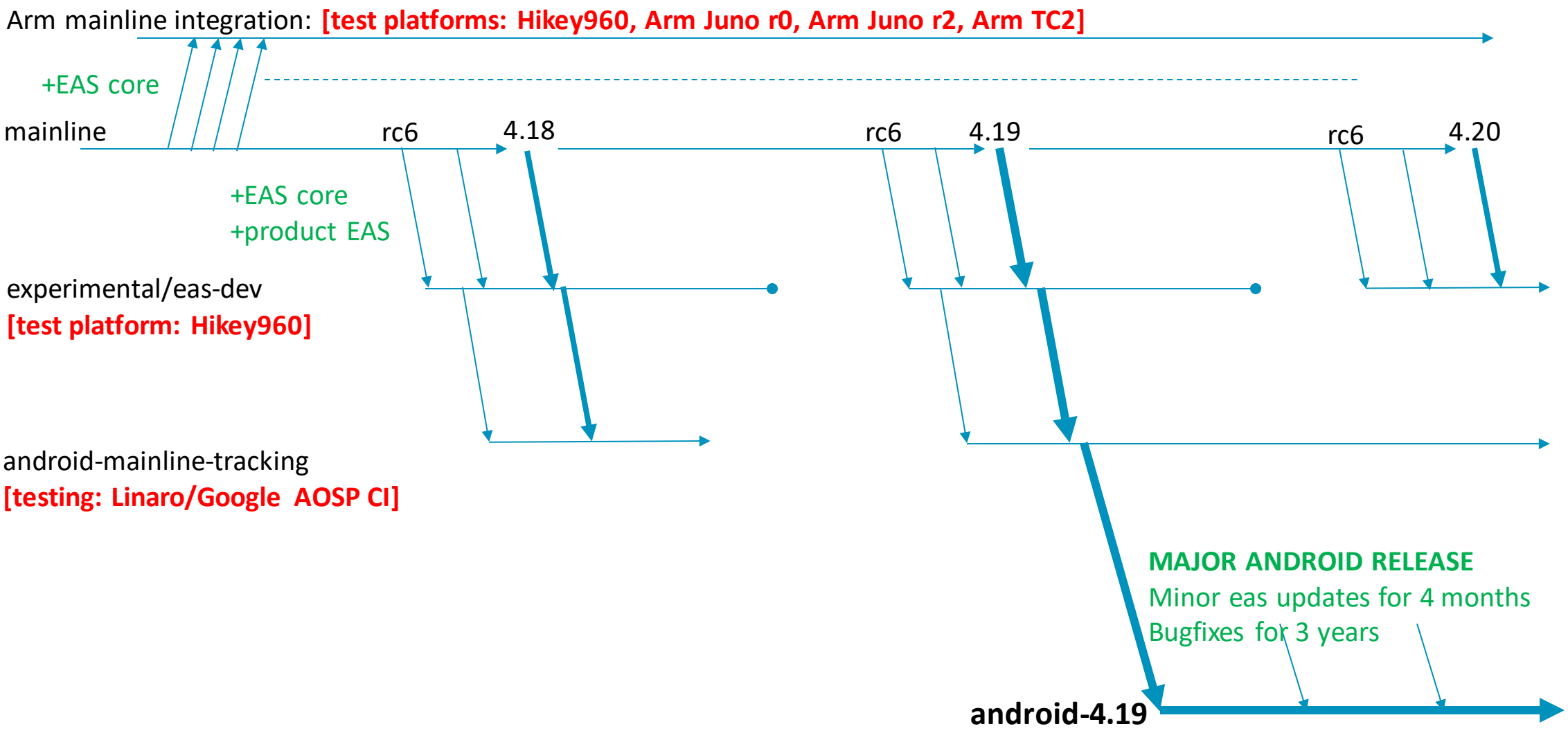
EAS for Android - improved release process

Much closer alignment with mainline Linux

- Goal: earliest creation of Android Common Kernel (ACK) android-X.XX kernel version containing product-quality EAS
- **Single release per year** for ACK
- Longterm goal: minimize differences between LTS kernel an ACK
- Further EAS updates to each android-X.XX release are minimal => easy to catch up
- Development on **android.googlesource.com/kernel/common.git experimental/eas-dev**, tracks latest mainline kernel closely

Fully open workflow – ACK patches discussed on LKML and eas-dev mailing list

EAS for Android - improved release process



arm

EAS Update – Upstream Linux Kernel – Building Blocks

EAS Building Block	Status HKG18	Status YVR18
Frequency and CPU Invariant Engines (FIE/CIE)	merged in v4.15	merged in v4.15
Idle CPU PELT update (Remote status update) (Linaro)	tip/sched/core	merged in v4.17
Util Est	tip/sched/core	merged in v4.17
Misfit Task	v2 on LKML	tip/sched/core (v4.20)
Dynamic Topology Flag Detection	in development	tip/sched/core (v4.20)
Energy Aware Scheduling (Energy Model)	v1 on LKML	v7 on LKML (possibly v4.21)
Util Clamping	v1 on LKML	v4 on LKML

EAS Update – Upstream Linux Kernel – Building Blocks

- Not all features got upstream in LTS – but all have made excellent progress
  - Note: Expected LTS was 4.20 but 4.19 was late and 4.20 is now too far into Dec 30 to be an LTS
  - **Misfit task** & **Dynamic Topology Flag Detection** are queued in tip/sched/core
  - **Energy Aware Scheduling (Energy Model) (v7)** is still being discussed on LKML – positive feedback from Rafael Wysocki
- Will get better idea of merge time for **Util Clamping (v4)** at LPC 18
  - Previous consensus on design is holding, implementation and review ongoing on LKML

EAS Update – Upstream Linux Kernel – Additional Topics

- Further reduce the delta between mainline and Android Common Kernel
  - Example: Per scheduler class utilization signal
    - [PATCH v7 00/11] **track CPU utilization** (Vincent Guittot, Linaro) (v4.19)
  - Example: One (running) task per CPU
    - [PATCH] sched/fair: **fix 1 task per CPU** (Vincent Guittot, Linaro) (LKML)
  - Max. Frequency Capping -> Introduce Thermal Pressure
    - [PATCH 0/6] **Introduce Thermal Pressure** (Thara Gopinath, Linaro)

EAS Update – Upstream Linux Kernel – Integration & Testing

- Biweekly EAS integration branch
  - Based on git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git sched/core
  - Platform support for Juno r0, r2, Hikey960 (64bit) and TC2 (32bit)
  - Published at https://developer.arm.com/open-source/energy-aware-scheduling/eas-mainline-development

- Biweekly Regression testing on all supported platforms
  - Based on Lisa Test Framework
    1. EAS task placement and latency tests (Generic)
    2. PELT Load Tracking tests (Load-Tracking)
    3. Asymmetric CPU capacity load balancing tests (Misfit)
  - Published at https://developer.arm.com/open-source/energy-aware-scheduling/eas-mainline-development (Juno r0)

arm

EAS Update – Upstream Linux Kernel – Integration & Testing

https://developer.arm.com/open-source/energy-aware-scheduling/eas-mainline-development

1. Latest source code
2. Current integration changes
3. EAS main features
4. Latest integration test results from Juno r0

Home / Linux and Open Source / Energy Aware Scheduling / EAS Mainline Development

EAS Mainline Development

Overview | GNU Toolchain | Energy Aware Scheduling | Intelligent Power Allocation

EAS Development for Mainline Linux

EAS development for mainline Linux takes place on LKML and Linaro eas-dev mailing lists where proposed patches are discussed with the community. Arm now provides a pre-integrated version of latest EAS patches for mainline, for upstream development use.

EAS mainline is a moving integration branch tracking tip:sched/core and adding the latest versions of EAS related patch sets targeted at the mainline kernel. The patches may already be under discussion on relevant open source mailing lists or waiting here while their dependencies get resolved and merged upstream.

The integration branch comes as is with synthetic test case testing only. The intention is to have a single branch containing the full EAS picture for mainline Linux kernel developers interested in tracking progress. This EAS mainline integration branch is not suitable for product development. For product, please use the AOSP common kernel.

Source code

EAS mainline integration branches are made available regularly on:

git://linux-arm.org/linux-power.git

The releases are tagged with the date e.g. _20180622 so old integrations are available for comparison

Latest: integration_20180914

Tree: eas/next/integration

Base: eas/next/integration_base

New changes:

1. [PATCH 0/4] sched/topology: Set SD_ASYM_CPUCAPACITY flag automatically by Morten Rasmussen on git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git sched/core (current Base).
2. [PATCHv4 00/12] sched/fair: Migrate 'misfit' tasks on asymmetric capacity systems by Morten Rasmussen on git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git sched/core (current Base).

Main features:

1. Per-cpu Energy Model and Energy Aware Scheduling.
2. Misfit task, i.e. forcing migration of running tasks that do not fit on the CPU they are currently running on.
3. Runtime scheduler domain flag detection.
4. Util(ization) clamping, i.e. a mechanism which allows to "clamp" (i.e. filter) the utilization generated by RT and FAIR task within a range defined from user-space.
5. Periodic load balance improvements.
6. Tracepoints.

Platform support:

1. Arm Juno r0
2. Arm Juno r2
3. Hikey960
4. Arm TC2

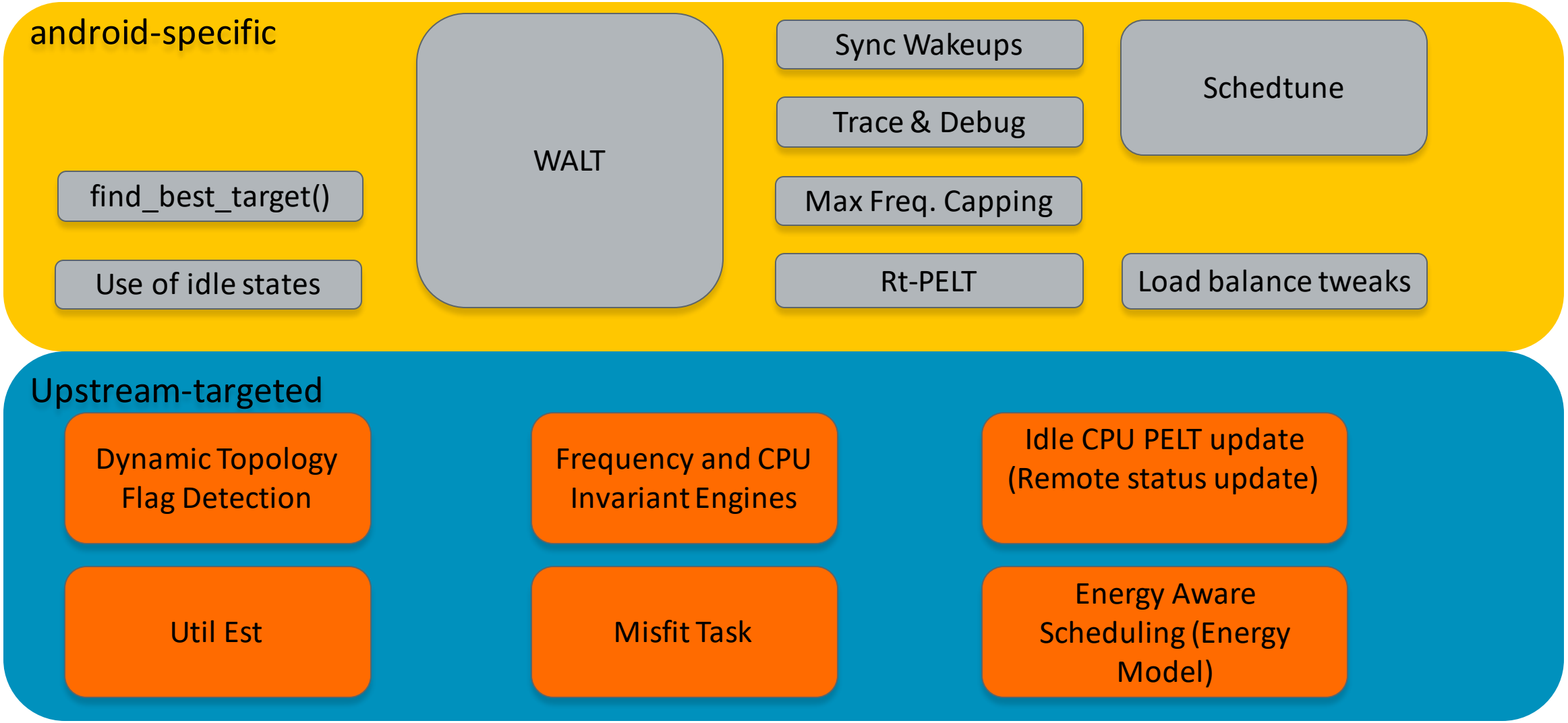
Next work items:

Work on getting a 100% pass-rate on all platforms (Arm Juno r0/r2, Hikey960 and Arm TC2).

Test Results (integration 20180914)

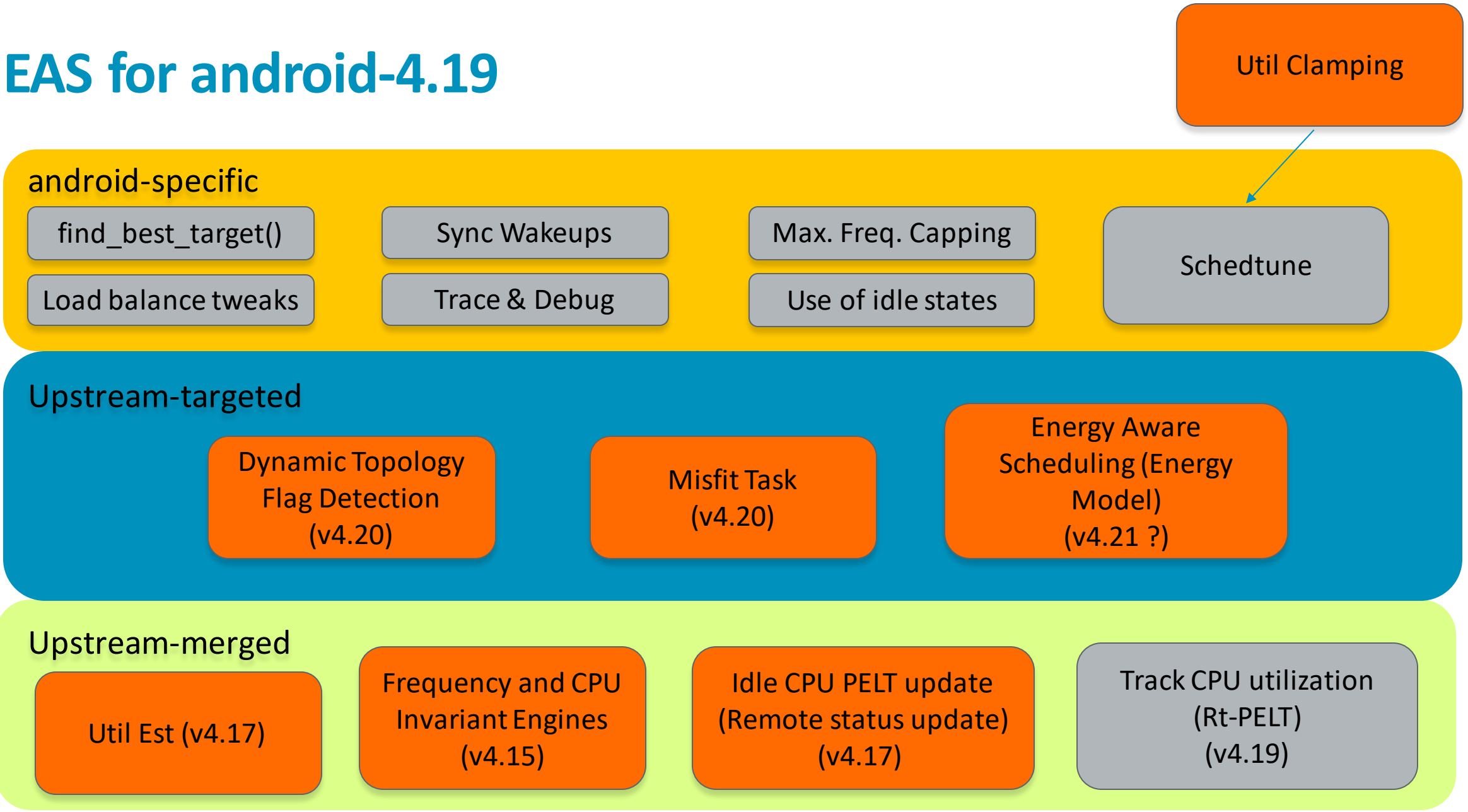
arm

EAS in android-4.14



arm

EAS for android-4.19



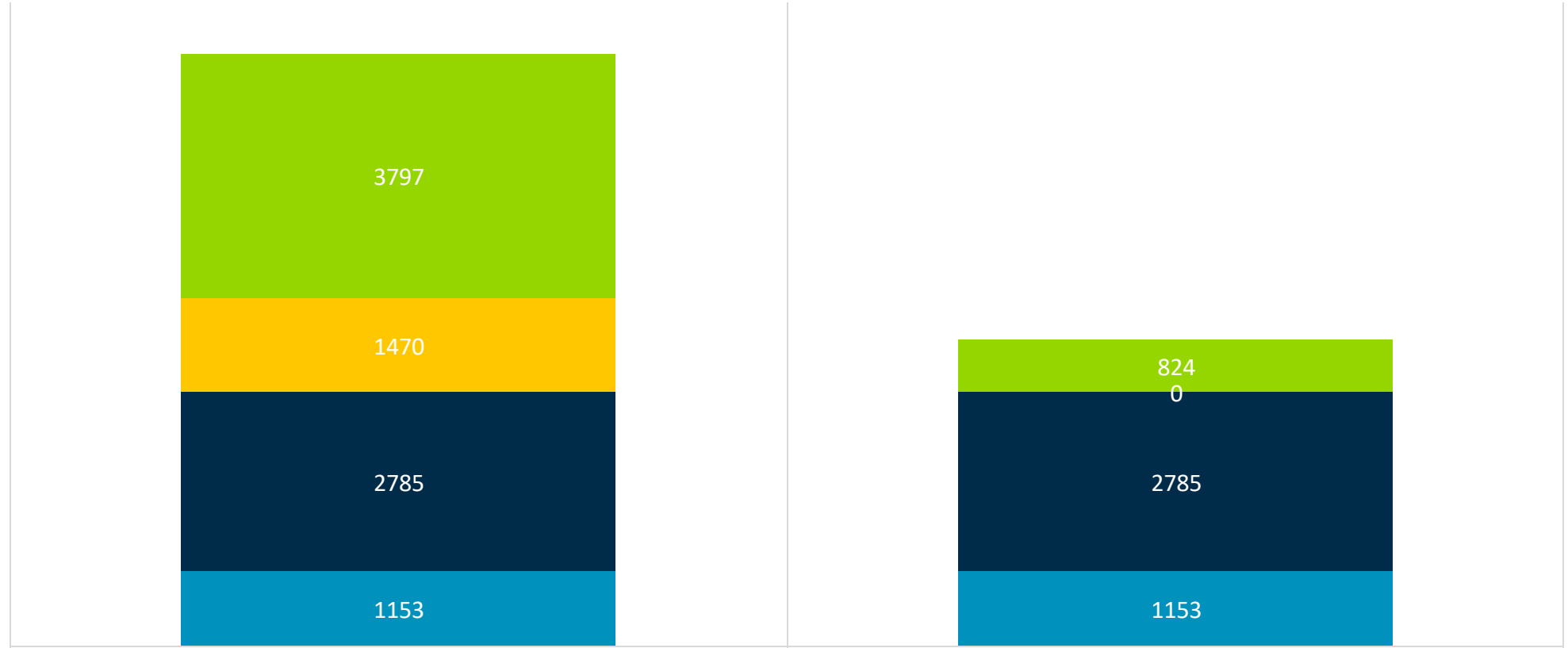
arm

EAS – product code size reduction v4.14 -> v4.19

EAS Building Block		android-4.14	android-4.19
Frequency and CPU Invariant Engines (FIE/CIE)	v4.15	+ 1053 / - 15	-
Idle CPU PELT update (Remote status update) (Linaro)	v4.17	+ 677 / - 447	-
Util Est	v4.17	+ 265 / - 20	-
Misfit Task	v4.20	+ 168 / - 32	+ 168 / - 32
Dynamic Topology Flag Detection	v4.20	+ 78 / - 9	+ 78 / - 9
Energy Aware Scheduling (Energy Model)	v4.21 (?)	+ 1149 / - 80	+ 1149 / - 80
(Schedtune /) Util Clamping	?	+ 1688 / -14	+ 1688 / -14
Track CPU utilization (Rt-PELT) (Linaro)	v4.19	+ 635 / - 429	-

arm

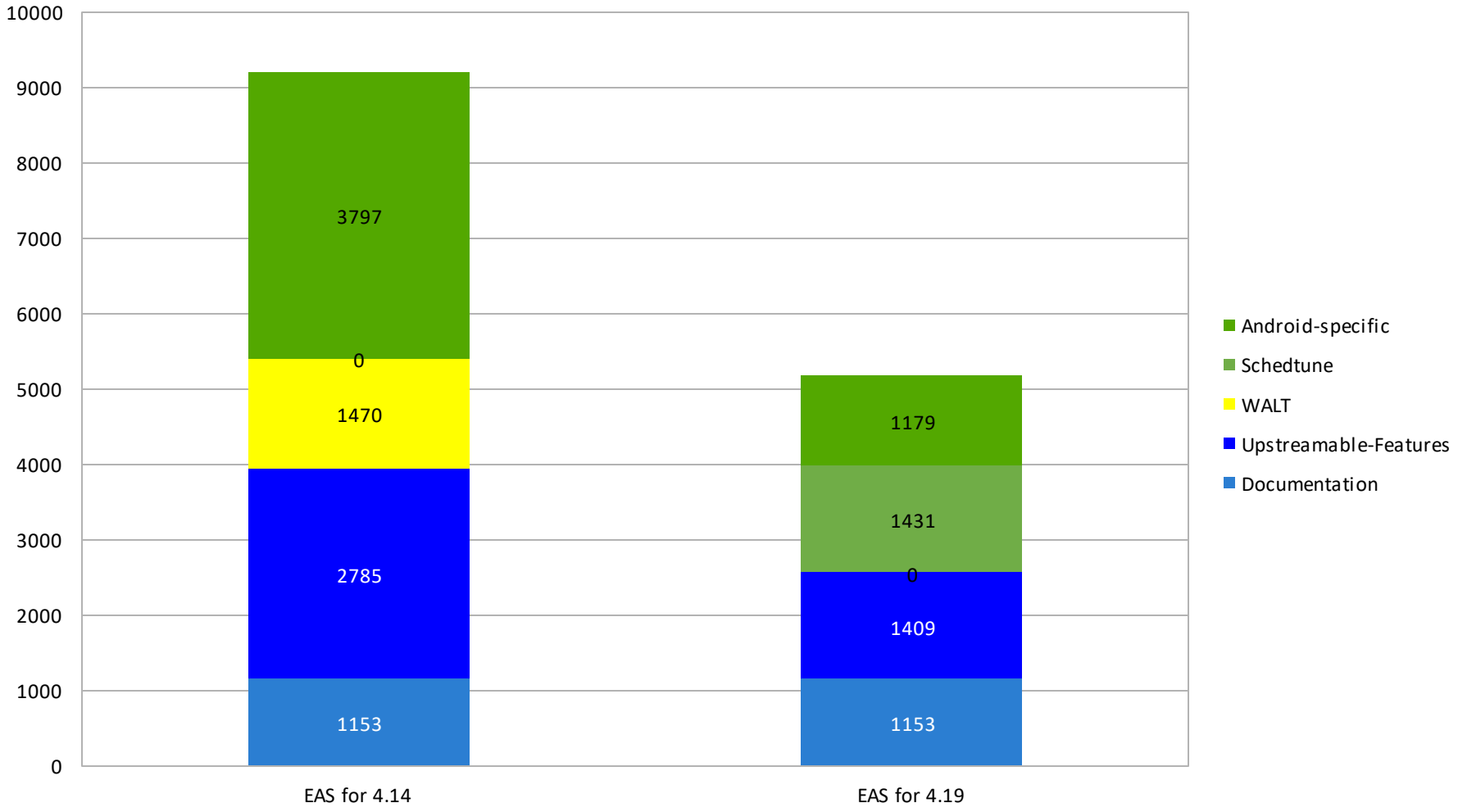
EAS in AOSP (HKG18 Recap)



	EAS Code Size by Category	Target EAS Code Size by Category
Android-specific	3797	824
WALT	1470	0
Upstreamable Features	2785	2785
Documentation	1153	1153

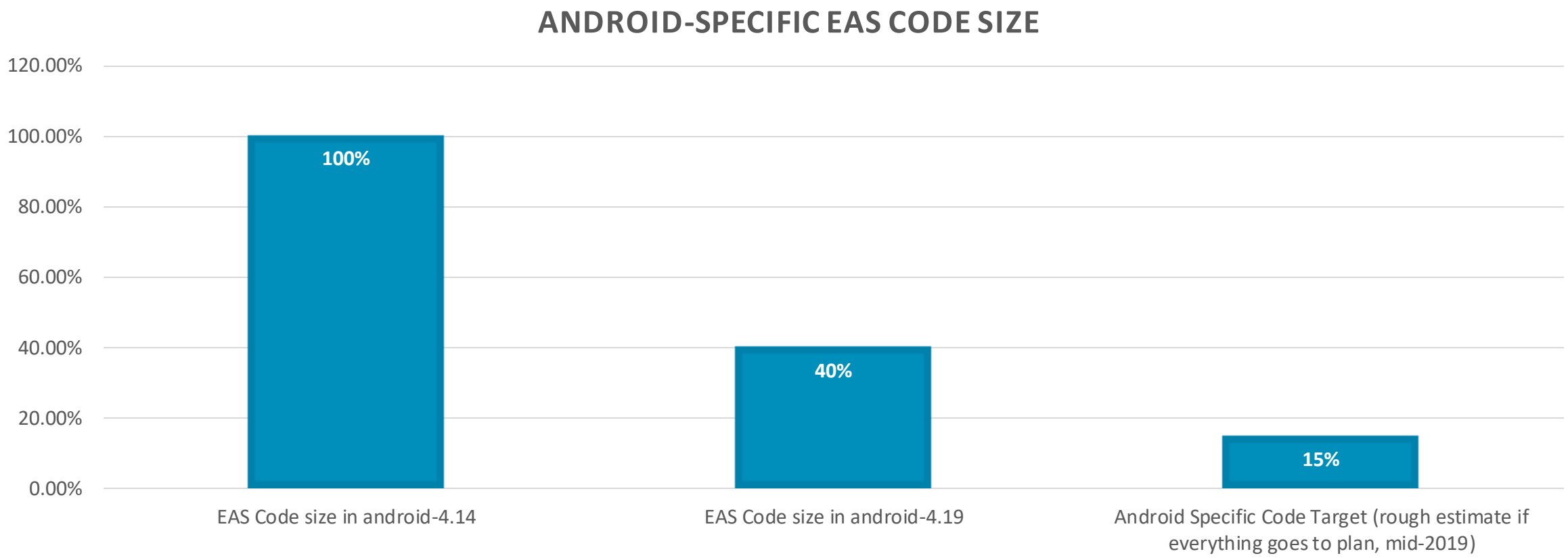
arm

Progress since HKG2018



arm

EAS Size Targets



arm

Thanks.

Any questions ?

arm