Yocto Layers and Device Profiles

July *11, 2017*

LINUX FOUNDATION

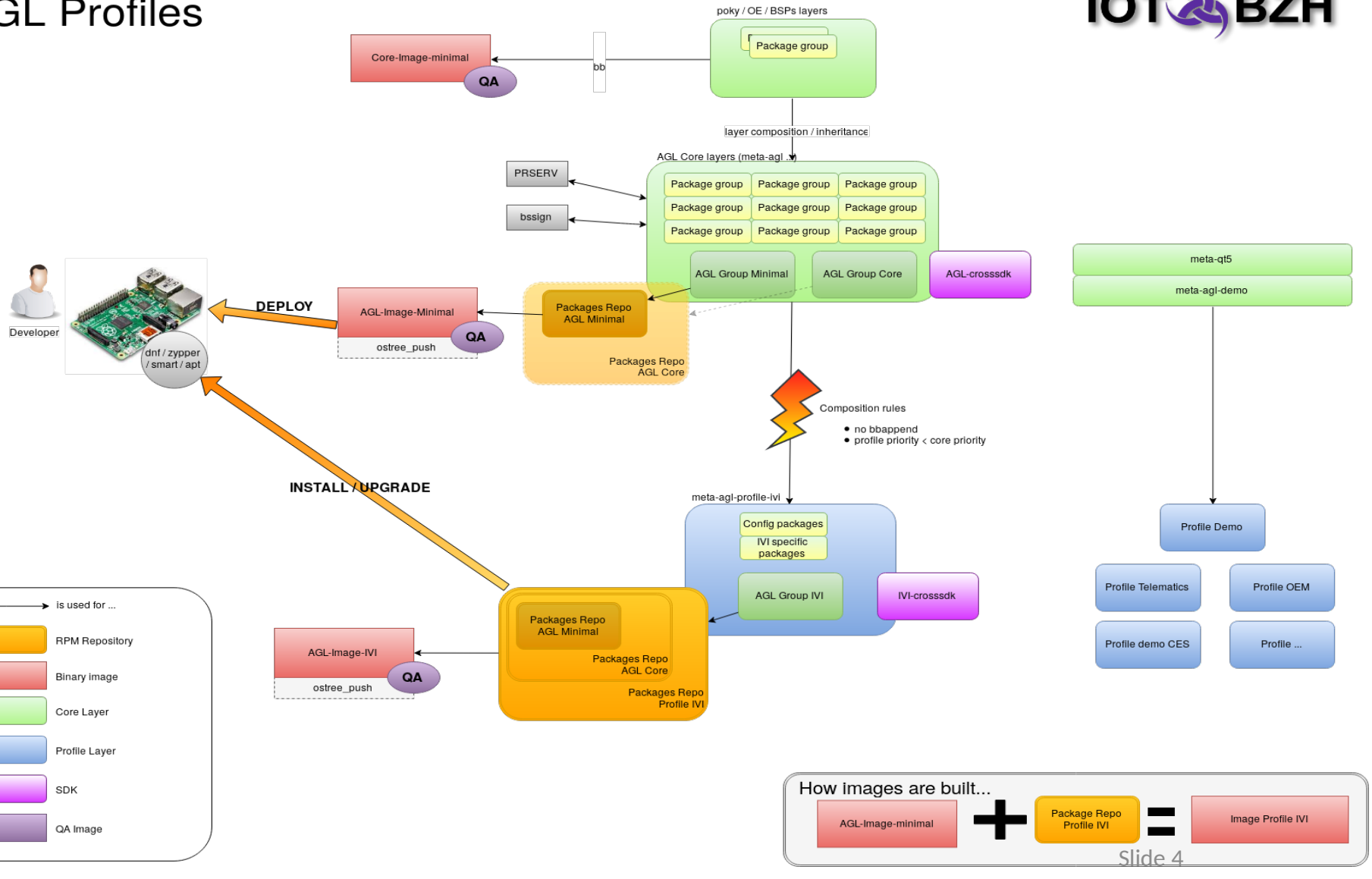
AUTOMOTIVE GRADE LINUX

AGL PROFILES

Prio discussions on Profiles

- Shared doc:
  - https://docs.google.com/document/d/1UFs_f7Cdom5F6GIemRuF_lk_kPIvR-Fk52jeL8ZL0Lw/edit
- Shared drawing:
  - https://www.draw.io/#G0B_w9btsPGBLvZW5mU3JjVklMYkk

AGL Profiles



Generic requirements for profiles

A profile needs to fullfill / provide / contain :

- a superset of the core
- only bbappends (!)
- profile priority < core priority
- options aka DISTRO_FEATURES:
  - **debug build**, *hypervisor*, **qa**
  - *min. capabilities defined for above*

What profiles should we have ?

Envisioned / proposed profiles

- core
- headless / telematics
- ivi (due to sdk needs, spins for -qt5 / -gtk)
- demo

AGL "core" profile

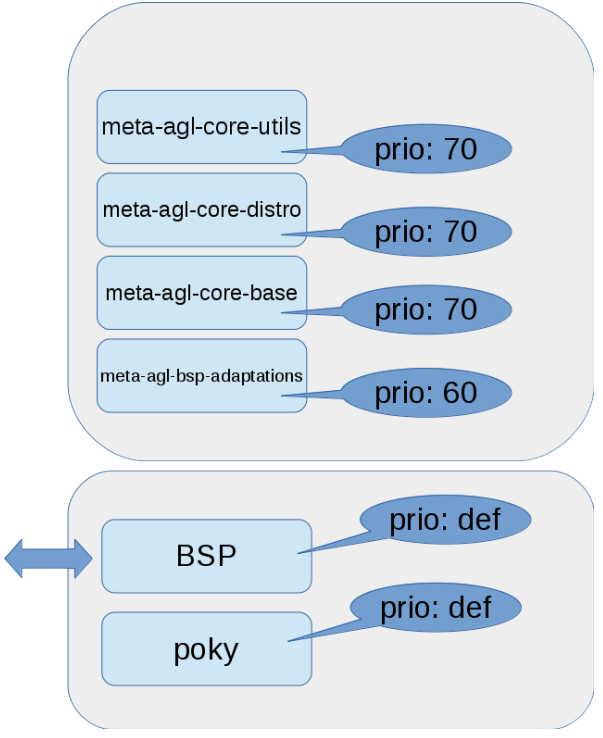
A "core" profile needs to contain/provide:

- headless base system
- AppFW
- security, smack, (secureboot)
- connectivity (at least one, e.g. ethernet)
- sota,update mechanism, package manager

- Also part – but supplied as installable wgt files are:
- ***platform-level binders - like signalling / can*** ***← (supplied as wgt)***

- More specific requirements can be:
  - minimal kernel version or Yocto/AGL features or config fragments

- In yocto terms: core-image-minimal + ^a^b^o^v^e^



AGL "telematics" profile

A "telematics" profile needs to contain/provide:

- ***V2C*** ***← (supplied as wgt)***
- ***Dashboard / Remote control API*** ***← (wgt)***
- ***specific high-level APIs ?*** ***← (supplied as wgt)***
- ***specific connectivity ?*** ***← (supplied as wgt)***
- agl 'core' + ^^^^^

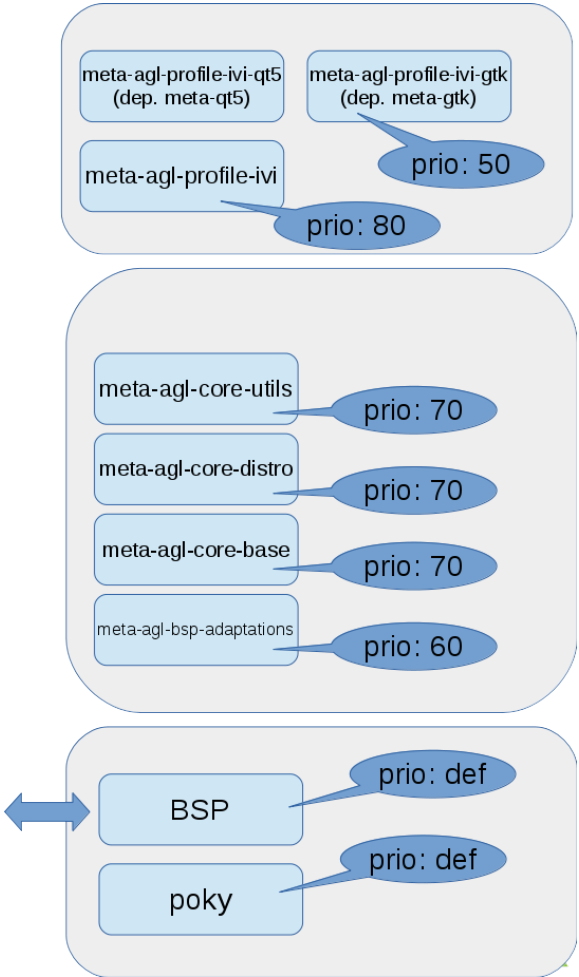
- → same platform as core (all extra is .wgt)

  (yes, we need to talk about kernel/driver differences)

AGL "ivi" profile

A "ivi" profile needs to contain/provide:

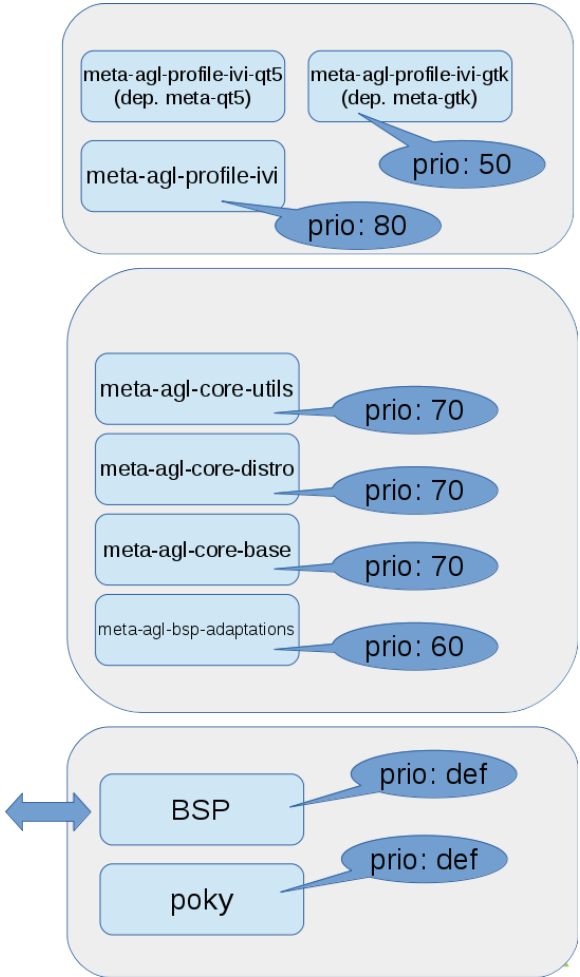
- gfx / wayland + AGL shell protocol (e.g. xdg)
- ***audio / multimedia*** ← ***(supplied as wgt)***
- ***identity*** ← ***(supplied as wgt)***
- ***webview (browser)*** ← ***(supplied as wgt)***
- ***(high-level) application APIs (e.g. geolocation, ...)*** ← ***(supplied as wgt)***
- check with SPEC 1.0 for more req
- → Platform with gfx-stack + wayland, extras all in .wgt



AGL "ivi-qt5" profile (or pkggroup)

A superset of "AGL ivi" profile to build SDK with needed headers. Contains:

- spin of IVI profile++
- qt5 headers for SDK

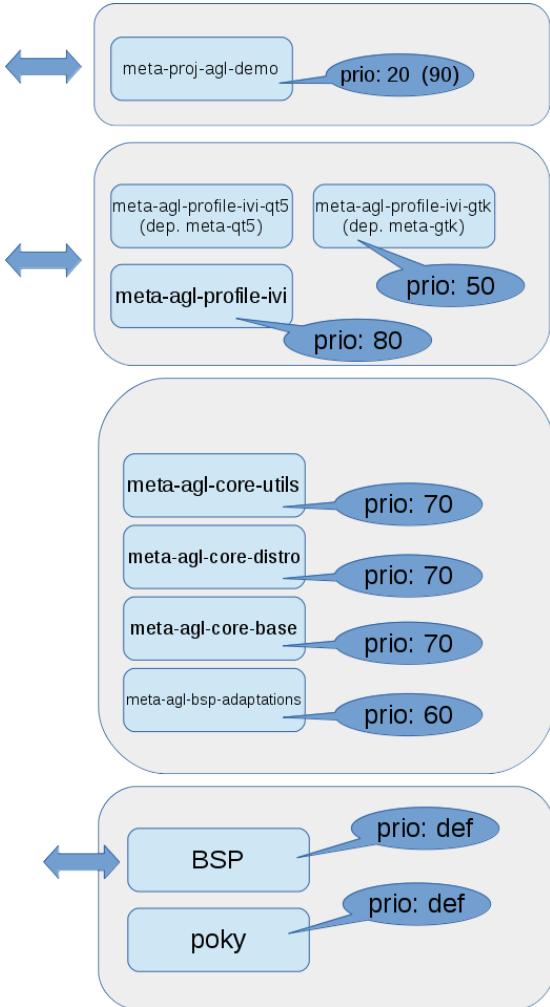


AGL "demo" **Project**

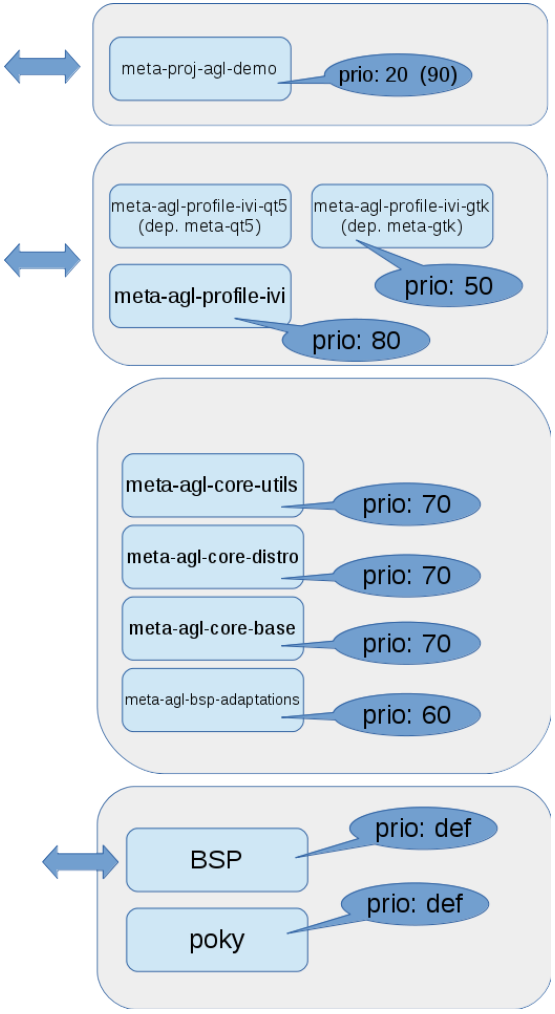
Our "AGL-demo" **Project** contains:

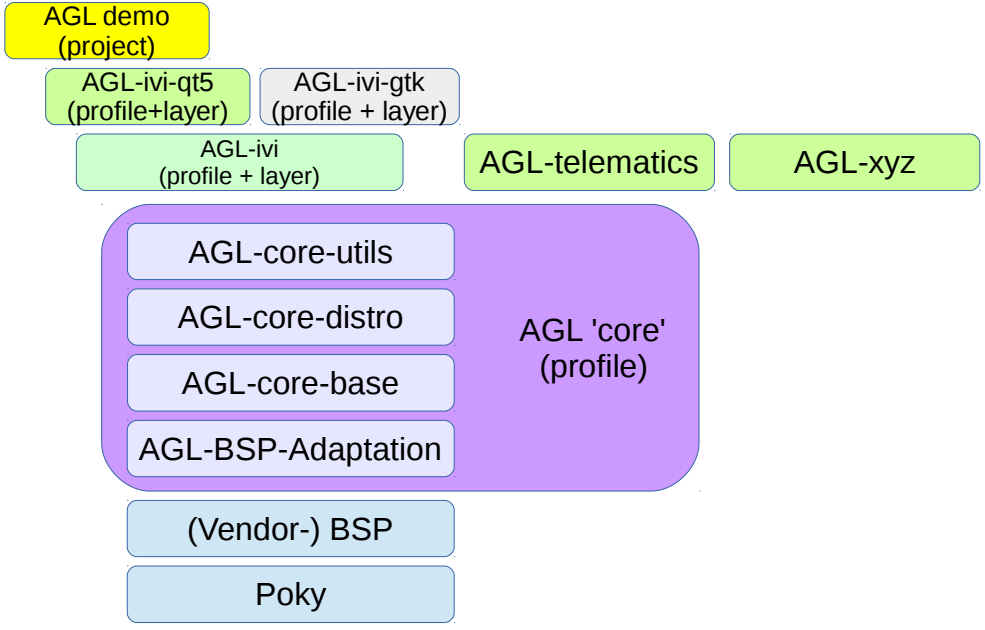
- spin of IVI-qt5 profile++
- ***reference apps (→wgt)***

- A project is a specific instance/spin of a profile



layers / profiles / projects overview





Challenge – NxM SDK

We should only have (at best)

- one SDK per architecture (= 3-4)

We might end-up with

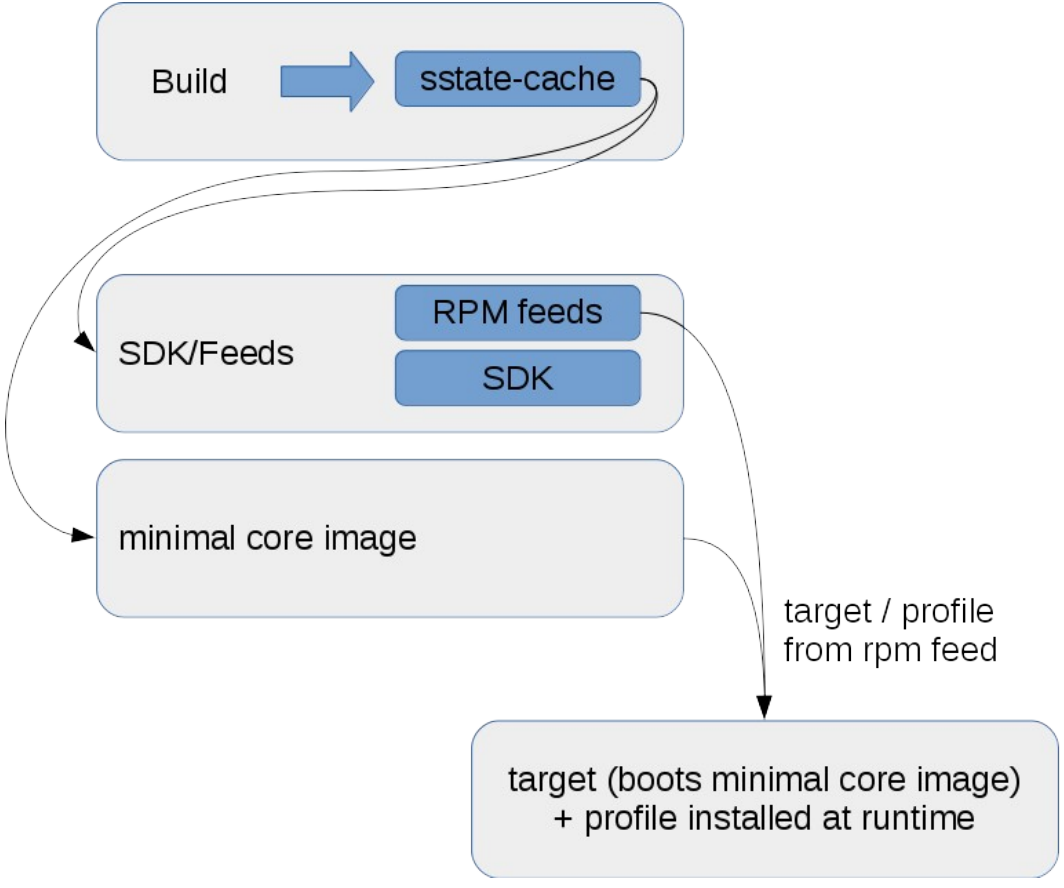
- one SDK per architecture and per profile
  (= 3-4 * X)
- For CES → Demo profile SDK (+ core SDK)
- middle term:
  - more flexible+scaling mechanism for SDK
  - Gaps identified: SDK needs to produce RPMs and be able to install additional -dev packages built with a matching SDK

Challenge – NxM SDK

- Plan:
  - simple "core" SDK for headless (no gfx/qt5/gtk)
  - ivi-qt5 SDK for AGL reference demo
    (core + meta-agl-profile-ivi + meta-qt5 + meta-agl-profile-ivi-qt)
- Options
  - e.g. ivi-gtk SDK
  - e.g. telematics or ADAS SDK
    (with extra libraries / drivers on-top of 'core')

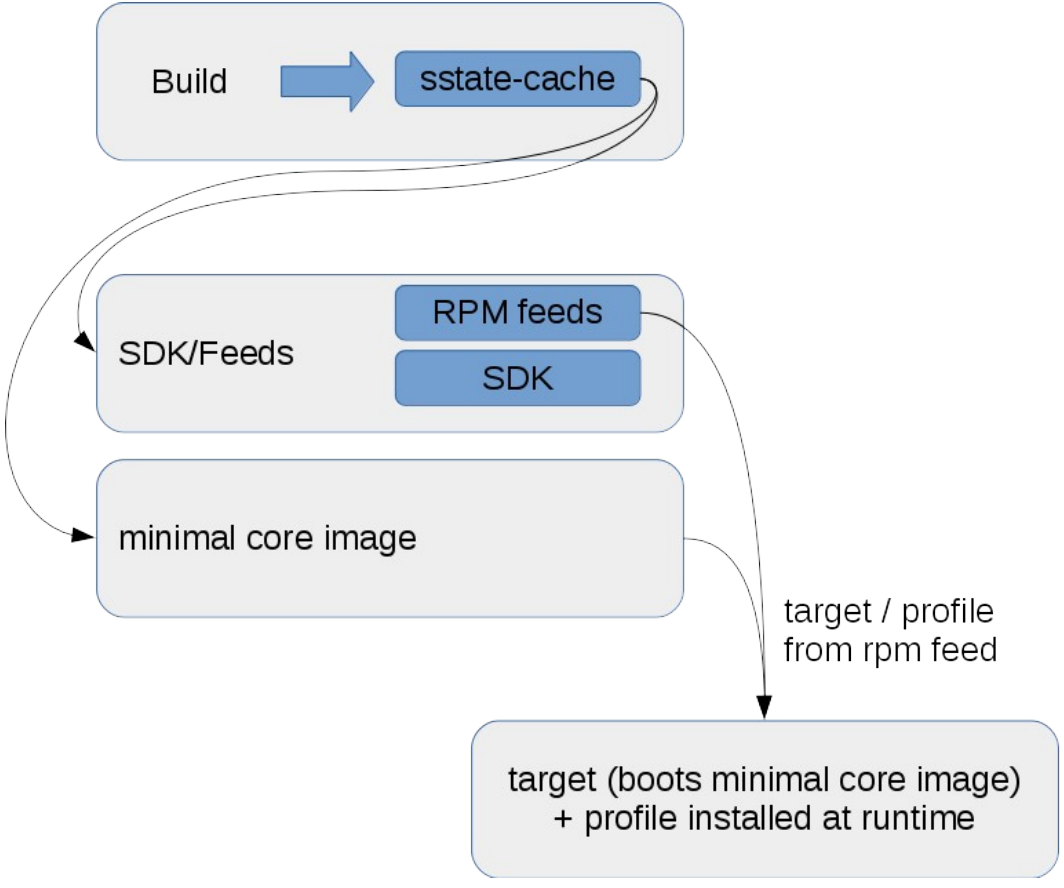
BINARY / RPM FEED

rpm feed for updates/images/SDK



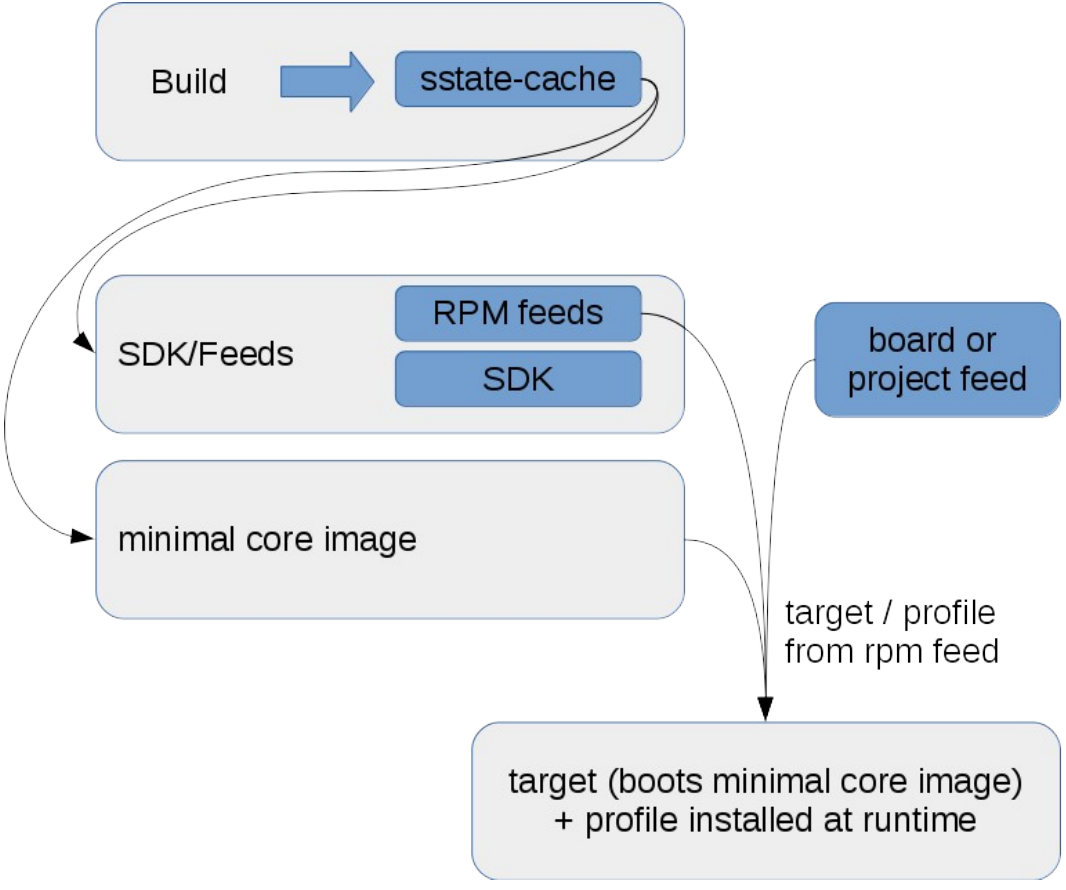
- single / locked sstate-cache
- used to generate rpm feed + sdk out os same artifacts
- targets pulls profile from feed

rpm feed for updates/images/SDK



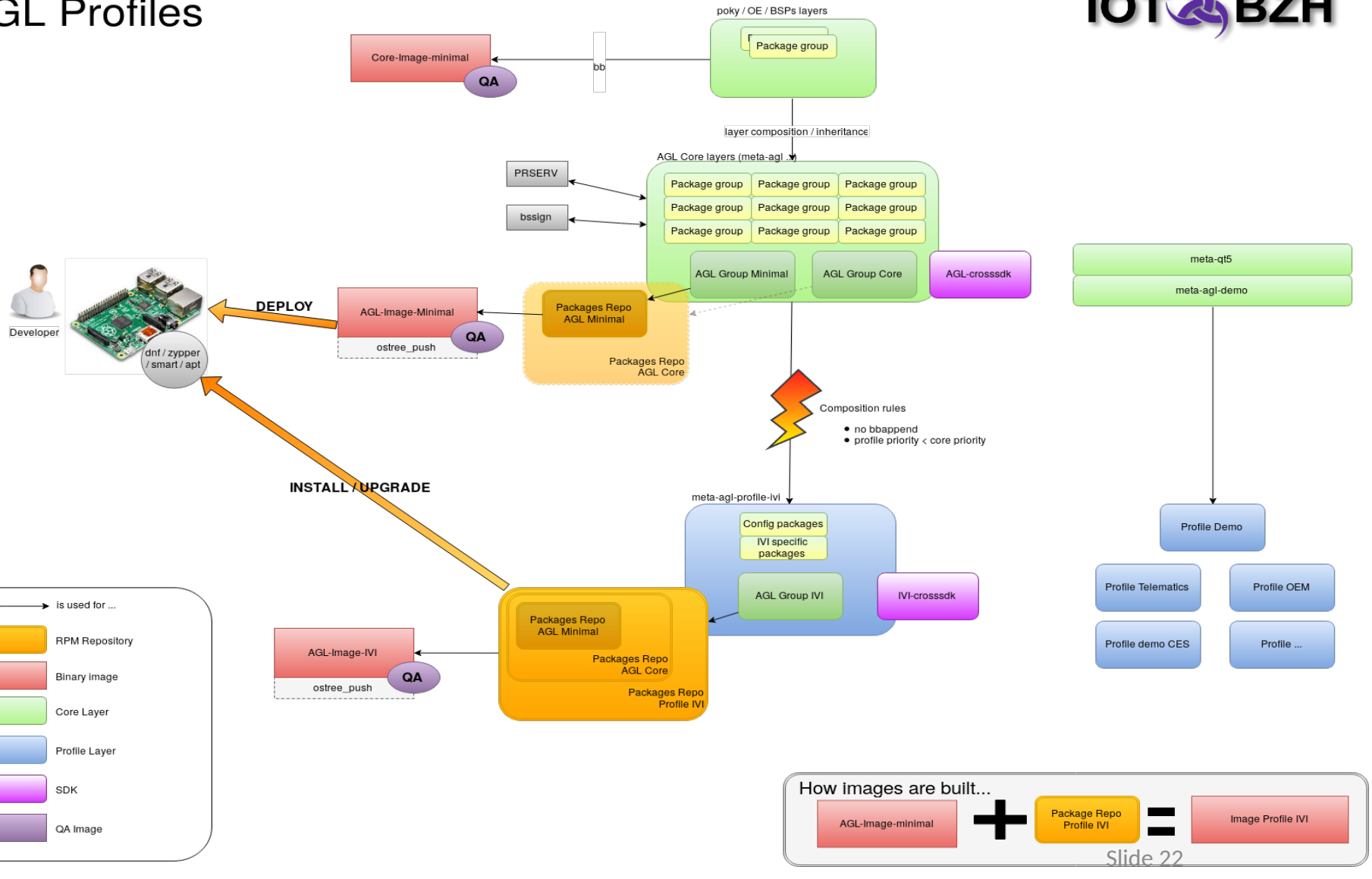
- 3 architectures:
  - generic-x86-64
  - generic-aarch64
  - generic-arm (**high**/medium)

rpm feed for updates/images/SDK



- filesystem / feed is generic !!!
- project specifics can be an additional feed (SDK needs ability to generated that!)

AGL Profiles



AI'S

AI's

- SPEC-675 - Rework packagegroups based on profiles (based on Layer F2F) - pending
- SPEC-676 - Define/document staging process and requirements for inclusion to "AGL core" – not started
- SPEC-677 - POC for signature lock – POC, see github
- SPEC-678 - POC for rpm generation out of locked signatures – investigating

- RPM signing – yes, but only 'local' signer support in yocto. Thus secret key on builder required atm. Better single purpose remote signer box.
- RPM generation out of SDK – investigating (aka run rpmbuild within SDK)
- SDK generation out of locked sstate-cache – investigation eSDK