



open**embedded**

yocto .
PROJECT

Inhoud

Geschiedenis / evolutie van
OpenEmbedded en Yocto

Overzicht van de architectuur en
concepten

Persoonlijke ervaring



Introductie

Leon Woestenberg

Embedded systeem architect

RTOS, Linux, hardware, FPGA

leon@sidebranch.com



Stukje geschiedenis

2002: Chris Larson vind de ROM van de Zaurus slecht.

Hij maakt bouwgereedschap voor een open Linux distributie.



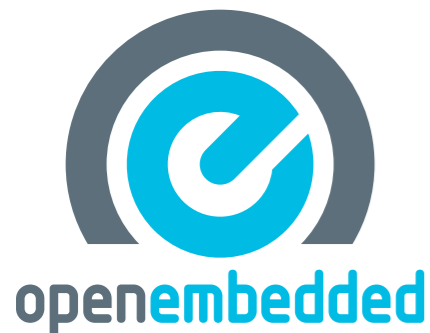
Stukje geschiedenis

2002: OpenZaurus project wordt voor andere apparaten gebruikt.



Stukje geschiedenis

2003: Holger Schurig, Michael Lauer en Chris Larson bedenken een nieuw build systeem: OpenEmbedded.



Stukje geschiedenis

2004: OpenEmbedded splitst het bouw gereedschap 'Bitbake' van de bouw instructies.



Stukje geschiedenis

2005: OpenEmbedded is volwassen.
(tenminste, dat vond ik toen).



Doelen

Build-from-scratch

(versus bestaande distributie
installeren en aanpassen.)



Doelen

Cross-compileren

(versus compileren op de 'target' machine of architectuur zelf.)

Doelen

Deterministisch en correct

Geen vervuiling uit host Linux

(lastig bij cross-compileren).



Doelen

Deterministisch en correct

Geen vervuiling uit host Linux

(lastig bij cross-compileren).

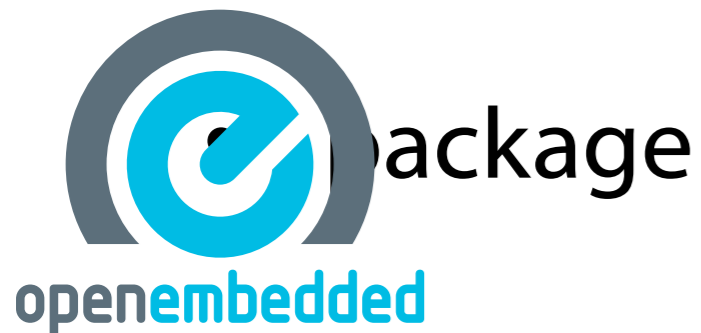
Taken



Software bouw taken

- fetch
- unpack
- patch
- configure
- compile
- install

} 'Tasks';
uitgevoerd
door 'BitBake'



Objectgeëïenteeerd

- basis implementatie voor elke taak (basis klasse)
- taken kunnen worden uitgebreid (prepend/append) of vervangen (override)
- eigen taken kunnen worden ingevoegd

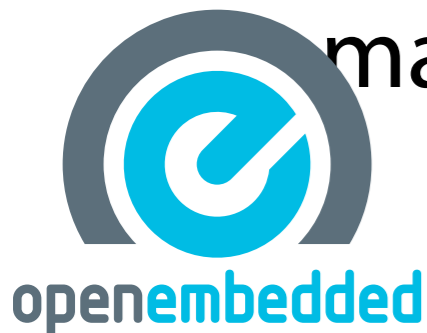


Objectgeëïenteeerd

implementaties voor gangbare standaarden:

- archiefformaten
- versiebeheersystemen
- make systemen
- distributieverpakkingen (packages)

maar zelf een klasse toevoegen kan ook.



Recepten

Recepten

Voor elk software pakket is een BitBake recept wat de bouwtaken beschrijft. Dit zijn shell scripts.

Veelal wordt een bestaande klasse overerfd.

Gevolg: een recept bevat doorgaans weinig scripts, maar vooral shell variabelen die de taken gebruiken tijdens uitvoeren.

Hello World

```
1 DESCRIPTION = "HelloWorld"
2 SECTION = "examples"
3 LICENSE = "GPL"
4
5 SRC_URI = "file:///helloworld.c"
6
7 S = "${WORKDIR}"
8
9 do_compile() {
10     ${CC} ${CFLAGS} ${LDFLAGS} helloworld.c -o helloworld
11 }
12
13 do_install() {
14     install -d ${D}${bindir}
15     install -m 0755 helloworld ${D}${bindir}
```



Configuratie

Vertelt bitbake voor welke MACHINE gebouwd gaat worden en waar het de recepten en klassen moet zoeken.

Meerdere architecturen

host packages

target packages

cross tools



Meerdere packages

Eén recept kan meerdere packages genereren.

Bijvoorbeeld een package voor de shared library en één voor de headers ervan.

Afhankelijkheden

Elk recept kan afhankelijkheden aangeven; het kan pas gebouwd worden zodra een ander package beschikbaar is.

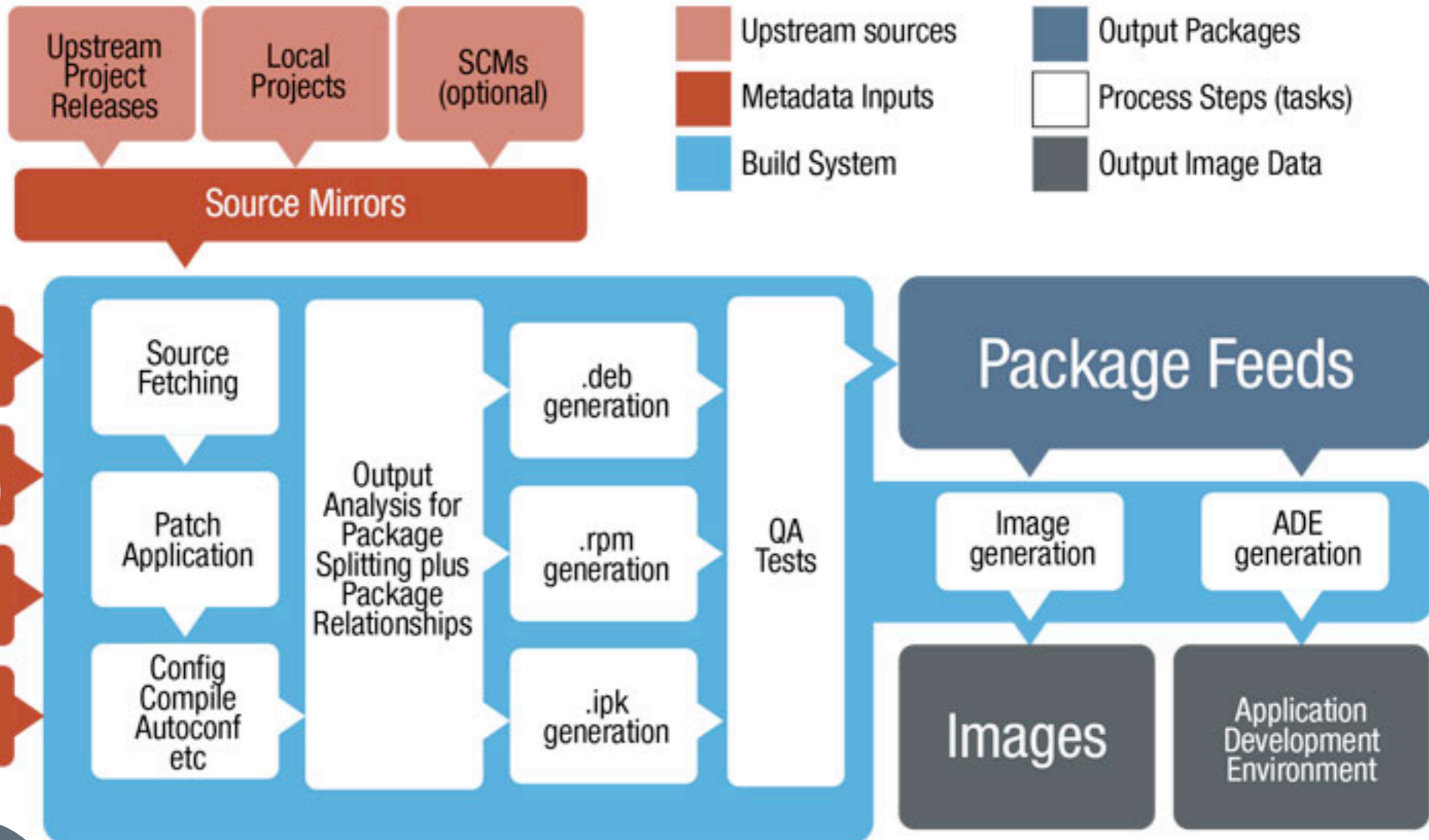
DEPENDS

Images

Veelal wil je niet één pakket bouwen, maar juist een compleet filesystem wat je direct kunt schrijven naar het apparaat.

Start

Bij het uitvoeren van BitBake geef je het gewenste package of image op, en BitBake begint te bouwen.



Bouwen

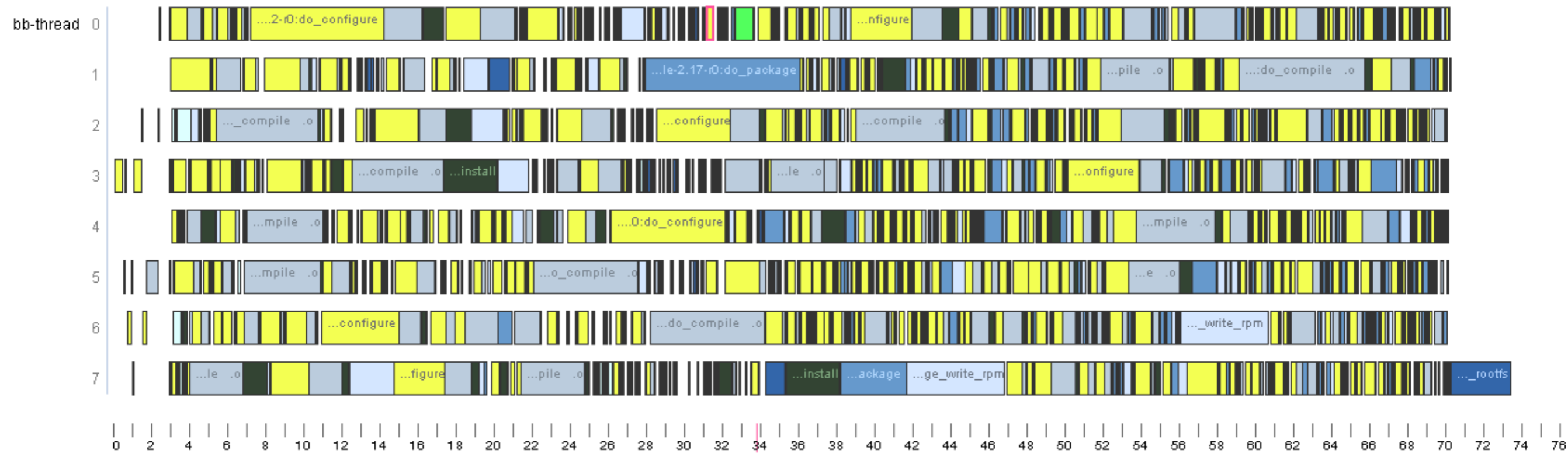
Run Queue

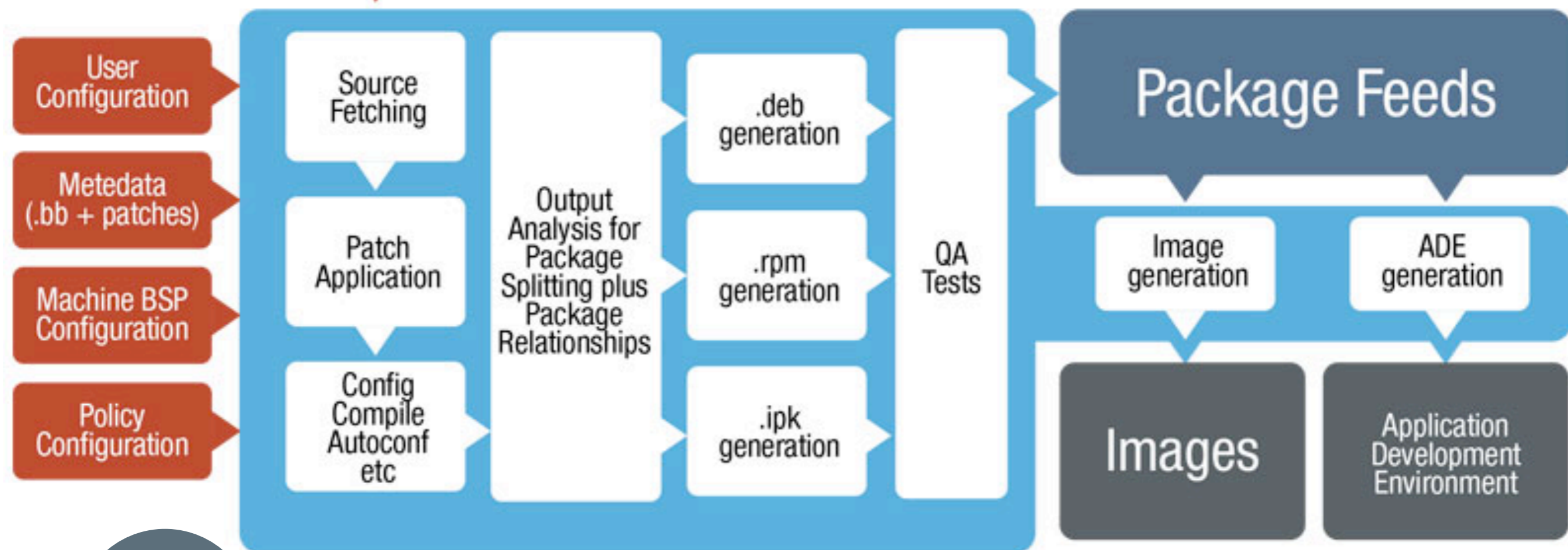
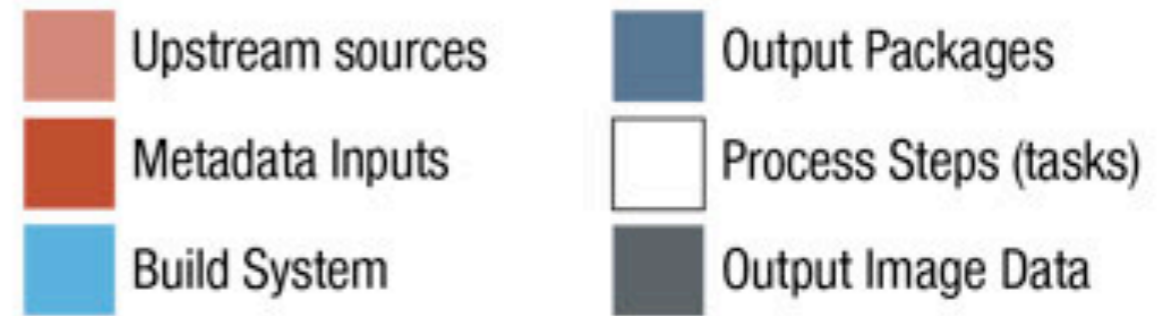
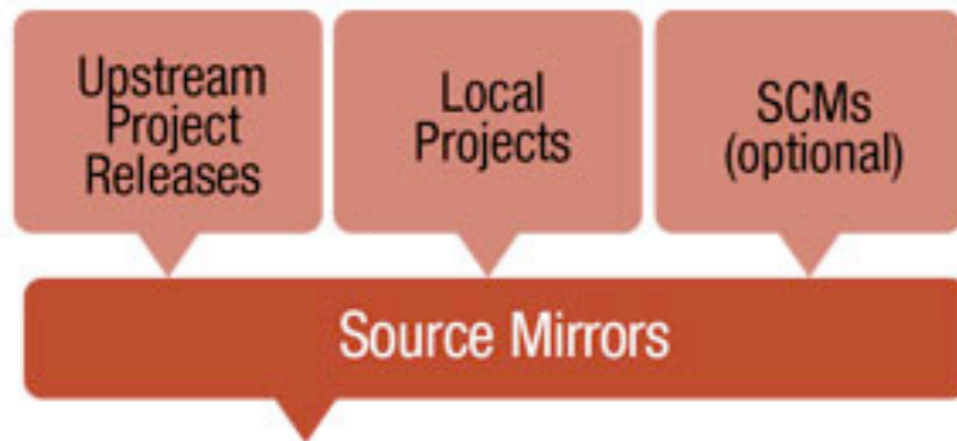
Mede gestuurd door de afhankelijkheden tussen packages, maakt BitBake eerst een afhankelijkhedenboom tussen alle uit te voeren taken.

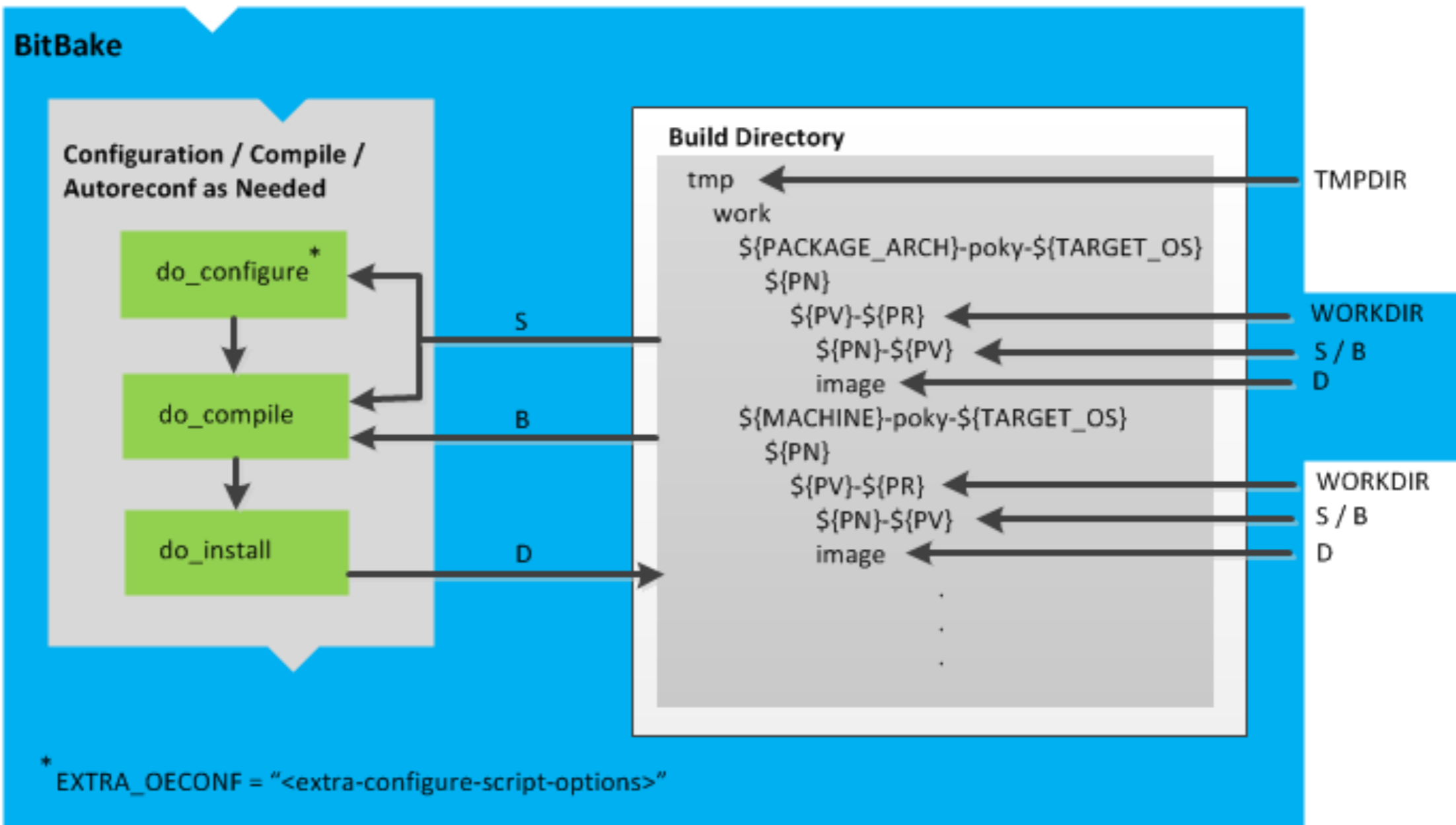
Deze taken worden verdeeld over meerdere threads en cores.

Bouwen op 8 cores

Agents & Jobs



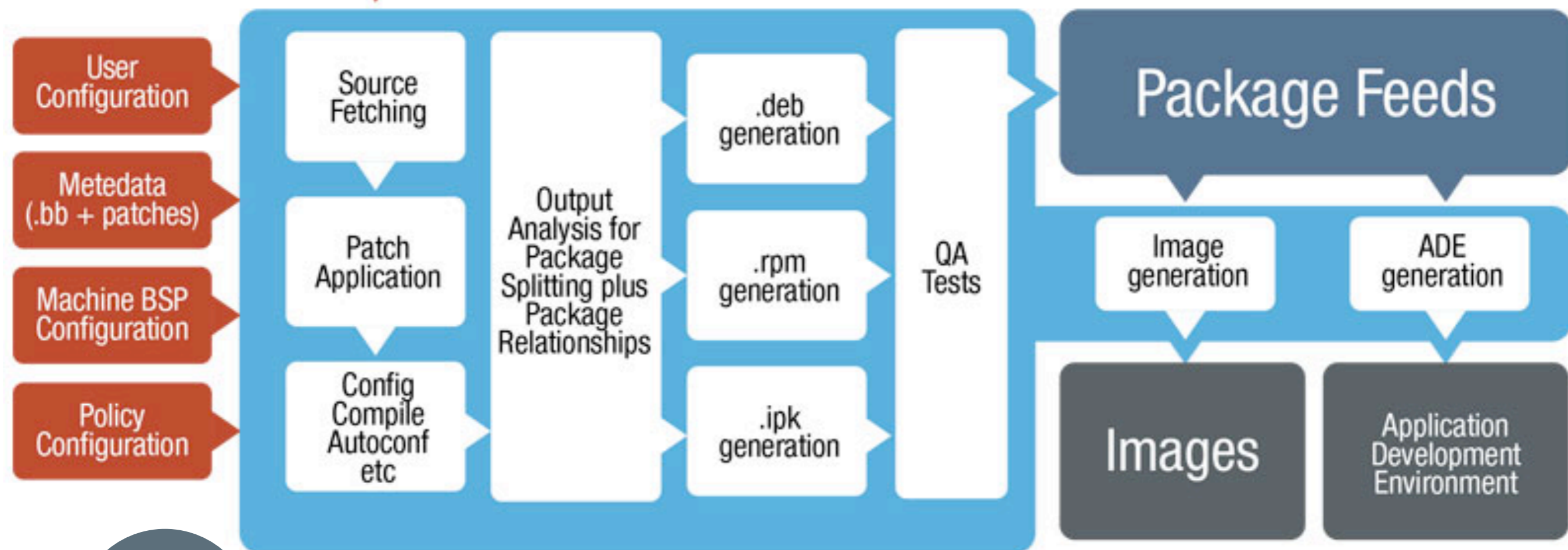
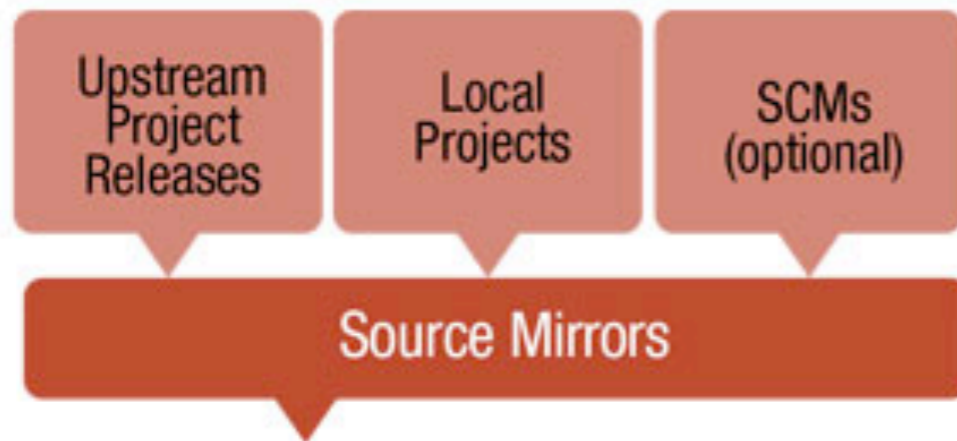




Cache

Checksums om te zien of afhankelijkheden zijn veranderd, bijvoorbeeld door aanpassing recept.

BitBake bouwt alleen de nodige pakketten opnieuw.



SDK's

OpenEmbedded kan een SDK bouwen voor een andere ontwikkelmachine.

Hiermee kunnen software ontwikkelaars hun software compileren voor het embedded target zonder dat ze een complete distributie moeten bouwen.

NATIVESDK

Resultaat

Deploy images

Kopieer bootloader, kernel en filesystem (image) naar de SD kaart, Flash, disk.

En booten maar...

Yocto



Meer geschiedenis

2008: Intel neemt belang in OpenedHand, een Linux ontwikkel club in de UK, die verbeteringen aanbrengt in 'Poky', een fork.

Meer geschiedenis

2010: Linux Foundation stelt het Yocto Project voor op de Embedded Linux conferentie in Cambridge.



Meer geschiedenis

2010: OpenEmbedded Core is een kleine subset van OpenEmbedded (Classic), basis voor Yocto Project, in sync met Poky.



Resultaat

Yocto Project gebruikt OpenEmbedded Core als basis voor een ontwikkelomgeving voor embedded Linux.

Yocto biedt meer high-level tools, zoals voor Eclipse en bouwen vanuit een web browser.



Hoe verder?

Suggesties:

Kies een ontwikkel platform wat wordt ondersteund door OpenEmbedded / Yocto en volg de "Getting Started" Guide.

Of neem deel aan de Yocto Developer Days.



Yocto Developer Day 2014



Einde

Vragen? Suggesties? Ervaringen?

Leon Woestenber

leon@sidebranch.com

