Using “kas” to make Yocto manageable

Alan Martinovic - External consultant for Mender.io
YOCTO PROJECT SUMMIT 2022.05
May 17 – 19, 2022
Talk overview - this is about

1. What is kas?
2. kas for simplifying Customer Engineering tools at Mender
3. Conclusion
4. Q&A
Talk overview - this isn’t about

- Why kas is better/worse than git submodules
- Why kas is better/worse than the repo tool
What is kas?
What is kas?

Official definition¹:

*Setup tool for bitbake based projects*

Unofficial definitions:

*Build an image with yocto from a blank slate in less steps.*

*Python wrapper for reducing the nr. of steps in yocto workflow.*

[¹] [kas github](kas.github.com)
What is kas? - Project health [github]

- Few commits a week
- Open to closed PR ratio: 1/23
- Open to closed issue ratio: 1/4
- docs live within the repository with the code
  - rendered on readthedocs
What is kas? - Installation

git clone -b 3.0.2 https://github.com/siemens/kas.git
cd kas
pip install .

- Containerized option also exists (kas-container)
- I only used it natively
What is kas? - Core features

Unofficial definition:
Build an image with yocto from a blank slate in less steps.

- config.yml
  - git clone ...
  - kas checkout
    - layer repos cloned
  - source build directory
  - kas shell
    - bitbake environment
  - kas build
    - bitbake image
    - yocto artifacts
What is kas? - Config file

```yaml
header:
  version: 8

machine: raspberrypi

distro: poky

target:
  - core-IMAGE-base

repos:
  meta-raspberry:
  poky:
    url: https://git.yoctoproject.org/git/poky
    path: layers/poky
    refspec: master
  layers:
  meta:
  meta-poky:
  meta-yocto-base:

meta-openEmbedded:
  url: https://git.openembedded.org/meta-openembedded
  path: layers/meta-openembedded
  refspec: master
  layers:
  meta:
  meta-python:
  meta-networking:
  meta-pi:
  meta-gtk:
    url: https://github.com/meta-gtk/make-gtk
    path: layers/meta-gtk
    refspec: master

bblayers_conf_header:
  format: ${MAKEINFO} -s
  target: "make work and downloads"
  rebuild: "m_make_deferred"

local_conf_header:
  format: ${MAKEINFO} -s
  target: "make work and downloads"
  rebuild: "m_make_deferred"
```

Example from https://github.com/agherzan/meta-raspberry/blob/master/kas-poky-rpi.yml

Miscellaneous

Where to clone layers from

What to add to bblayers.conf

What to add to local.conf
What is kas? - Config file

- config-x.yml
  - Miscellaneous
  - Where to clone layers from
  - What to add to bblayers.conf
  - What to add to local.conf

- config-y.yml
  - Miscellaneous
    - includes:
      - config-x.yml
  - Where to clone layers from
  - What to add to bblayers.conf
  - What to add to local.conf
kas in mender

Customer Engineering
kas at Mender - requirements

- Build multiple images for different HW (rpi3/4, bbb)
  - Diagnose customer issues
  - Test prospect requirements
- Diagnose Yocto issues
kas at Mender - requirements [rephrased]

- Minimal config and command overhead to build images
- `local.conf` doesn’t change much once set
  - I just want to make domain specific changes
- Simple to reach artifacts once they are built
- One step away from the regular bitbake env (if things break)
source set_machine rpi3
# Edit auto.conf to set the dynamic build config
# i.e. MENDER_ARTIFACT_NAME = "version-1"

build_and_gather core-image-minimal
cd artifacts/rpi3/

_kas_shell()

Mender relevant artifacts available here

Go to initialized build dir with all config preset
Why two configs? 
Wouldn’t one do?

Questionable benefits :)
Why two configs?
Wouldn’t one do?

Quick change in local.conf style
Known config format
Simpler README

source set_machine rpi3
build_and_gather core-image-minimal

set-me-once.yml
board_agnostic.yml
rpi3.yml
Kas at Mender - a wrapper around a wrapper

_kas_shell()

Wrapper around “kas shell”

Why another wrapper?!?

To propagate the auto.conf to the build environment

Results depend on a hidden state (ENV vars) instead of explicit parameters

Add an implicit mechanism for extending env with custom config
Add an implicit mechanism for extending env with custom config

Some mender features need multiple lines in local.conf
This allows for simpler feature inclusion
# Clone the repositories
git clone http://git.openembedded.org/meta-openembedded

git clone https://github.com/mendersoftware/meta-mender

git clone https://github.com/agherzan/meta-raspberrypi

git clone https://git.yoctoproject.org/git/poky

source poky/oe-init-build-env

# Please add the following to your local.conf

DL_DIR= "/home/workspace/build_hard/yocto_cache/downloads"

SSTATE_DIR= "/home/workspace/build_hard/yocto_cache/sstate-together"

MENDER_SERVER_URL = "https://hosted.mender.io"

# MENDER_SERVER_URL = "https://staging.hosted.mender.io"

DISTRO_FEATURES_append = " systemd"

VIRTUAL-RUNTIME_init_manager = "systemd"

VIRTUAL-RUNTIME_initscripts = ""

# INHERIT += "rm_work"

IMAGE_LINK_NAME_append = "-${MENDER_ARTIFACT_NAME}"

IMAGE_FEATURES = "ssh-server-openssh allow-empty-password debug-tweaks"

IMAGE_INSTALL_append = "python3"

# Stuff from meta-mender-ce

IMAGE_INSTALL_append = "mender-monitor-crasher-app"

INHERIT = "mender-full"

RPI_USE_U_BOOT = "1"

LICENSE_FLAGS_WHITELIST_append=" commercial_mender-binary-delta"

LICENSE_FLAGS_DISABLE_append=" proprietary_mender-binary-delta"

LICENSE_FLAGS_WHITELIST_append=" commercial_mender-binary-delta"

IMAGE_INSTALL_append = "read-only-rootfs"

IMAGE_INSTALL_append = "mender-binary-delta"

...

# Edit local.conf to set the dynamic build config

MENDER_ARTIFACT_NAME = "version-1"

bitbake core-image-minimal

# Edit auto.conf to set the dynamic build config

# i.e. MENDER_ARTIFACT_NAME = "version-1"

build_and_gather core-image-minimal

cd artifacts/rpi3/

_kas_shell()

Mender relevant artifacts available here

Go to initialized build dir with all config preset
Conclusion
Conclusion

- kas replaces extra commands with config files
- kas config files can be included into each other
  - as config syntax and as cli parameters
- in mender CE kas is used to do the heavy lifting with regards yocto setup
- It significantly reduces the complexity of helper scripts making them maintainable
Q&A

Thank you