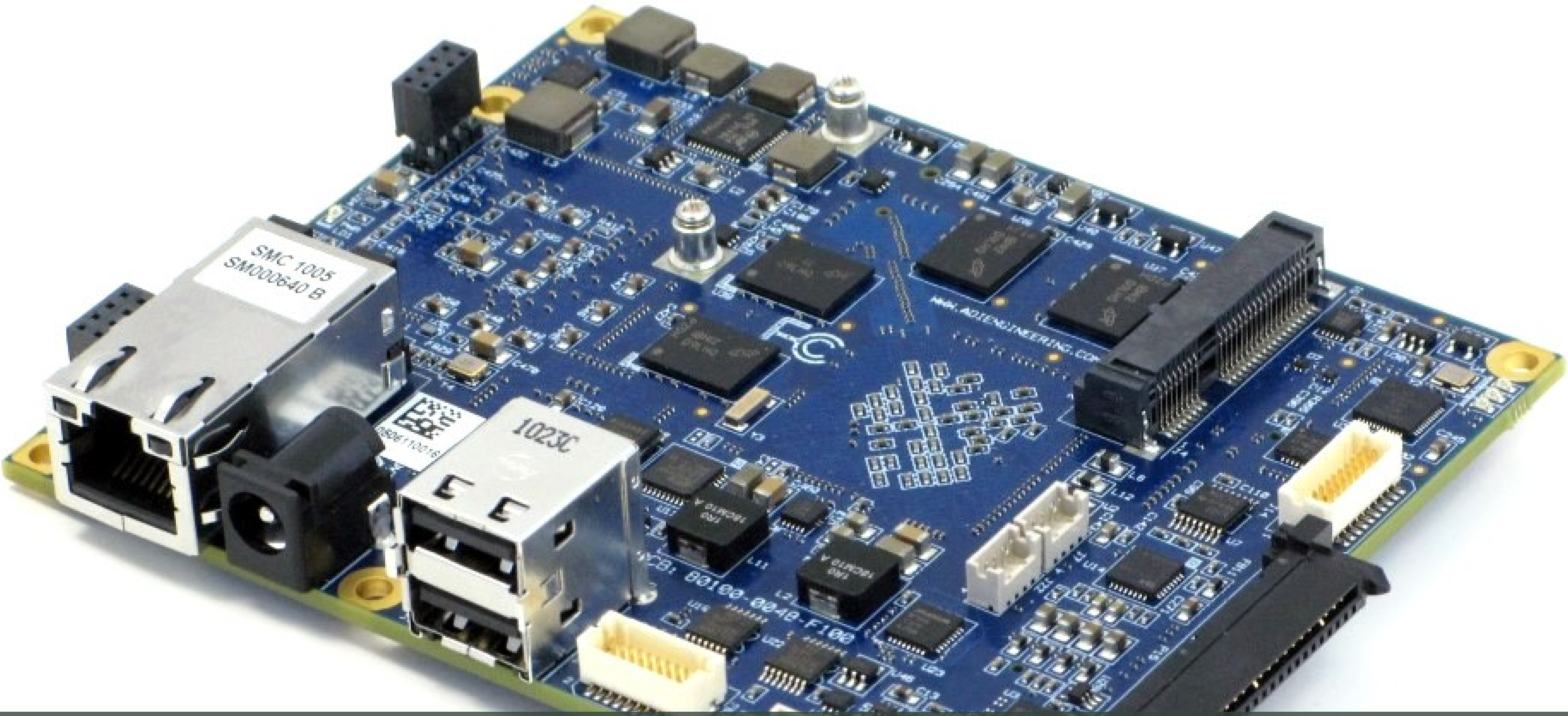


Reducing Time-to-Market with COTS, SBCs and Linux

FEABHAS

Why are you here?

You can't afford to ignore it

Why COTS?

Good enough

is

good enough.

Why SBCs?

An SBC is quicker and cheaper than building your own hardware.

Why Linux?

Linux has a stronger ecosystem than any other platform

Why Linux on Atom?

Atom is x86

The Bare Minimum

What do you need?

The Three Approaches

Bottom up

Top down

Middle out

Top Down Approach

A full desktop distribution is NOT a good basis for your product

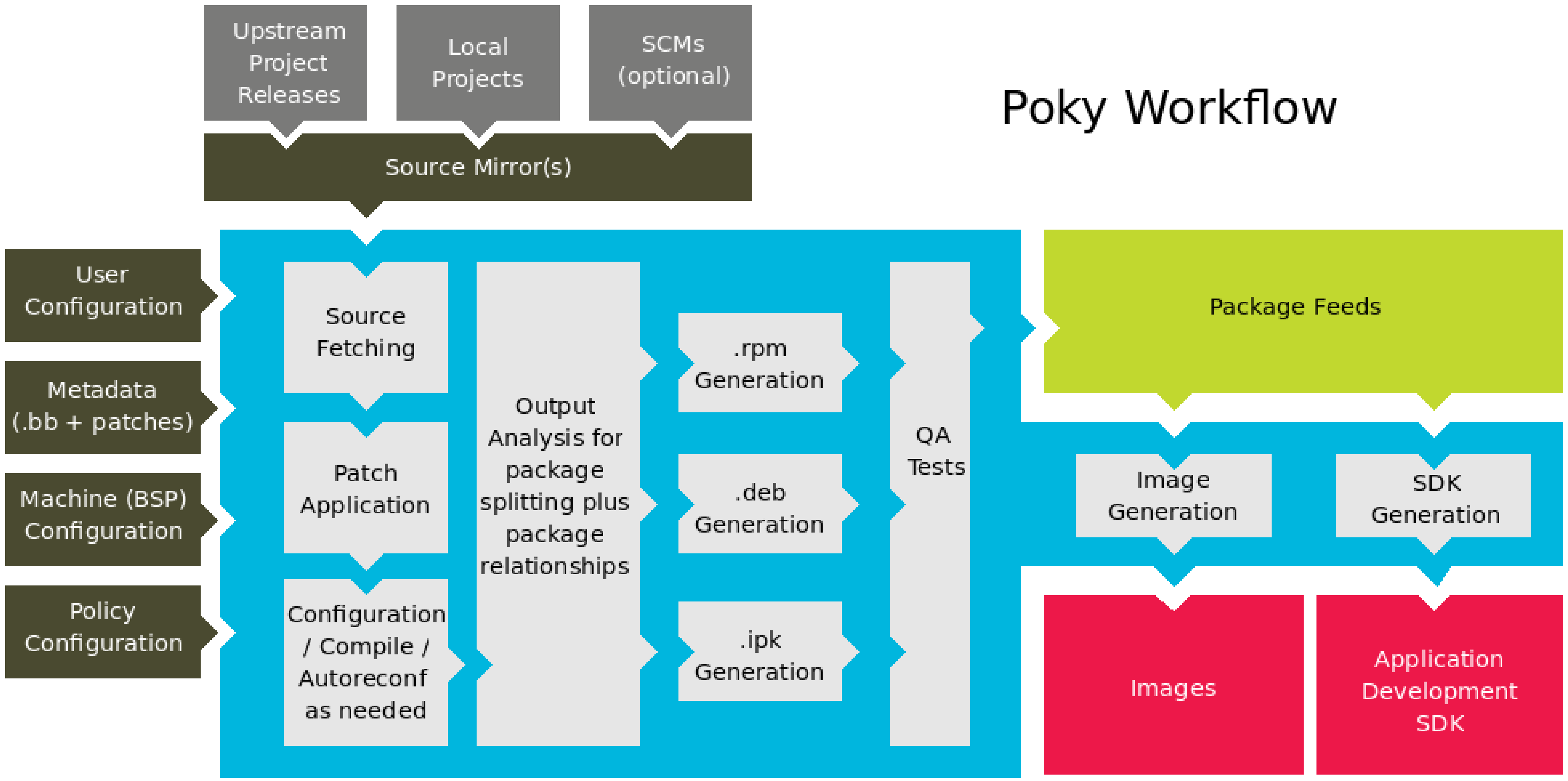
Bottom Up Approach

Bottom up is painstaking work

Yocto isn't a Linux distribution - it builds one for you.

https://wiki.yoctoproject.org/wiki/FAQ

Poky | The Yocto Build System



The How

So how do we use this?

Yocto | Setup

You will need to install the following packages:

```
sudo apt-get install sed wget subversion git-core coreutils \
unzip texi2html texinfo libsdl1.2-dev docbook-utils fop gawk \
python-pysqlite2 diffstat make gcc build-essential xsltproc \
g++ desktop-file-utils chrpath libgl1-mesa-dev \
libglu1-mesa-dev autoconf automake groff libtool xterm \
libxml-parser-perl
```

http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html

Yocto | Download & Install

Get the release tarball

```
wget http://downloads.yoctoproject.org/releases/yocto/ ↲
                yocto-1.2/poky-denzil-7.0.tar.bz2
```

Untar it

```
tar xjf poky-denzil-7.0.tar.bz2
```

Source and setup the environment

```
source poky-denzil-7.0/oe-init-build-env <my-project>
```

Yocto | Configure the Build

Set the machine type so that bitbake knows what to build by editing `conf/local.conf`

```
...
MACHINE=? “atom-pc”
...
```

And set the build going:

```
[nick@slimtop myatomsbc]$ bitbake -k core-image-minimal
```

Some Time Later…

Yocto | Output

Kernel Image

```
bzImage-atom-pc.bin
```

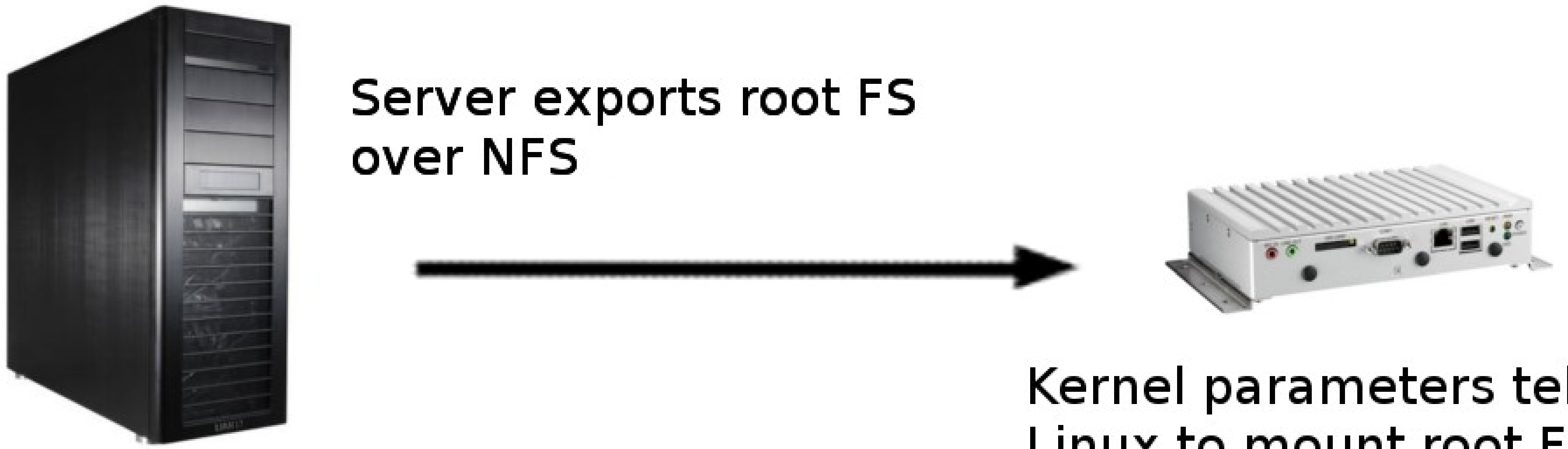
Root FS

```
core-image-minimal-atom-pc.ext3 (and .iso)
```

NFS?

NFS is the simplest way to deploy your root filesystem

NFS? Simple?



Server exports root FS over NFS

Kernel parameters tell Linux to mount root FS over NFS

Deployment | NFS

You will need:

```
sudo apt-get install nfs-kernel-server libgssglue1 \
libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server \
rpcbind
```

And do the following:

```
mkdir /nfsroot
```

Add it to your `/etc/exports`

```
/nfsroot *(rw,sync,no_subtree_check,no_root_squash)
```

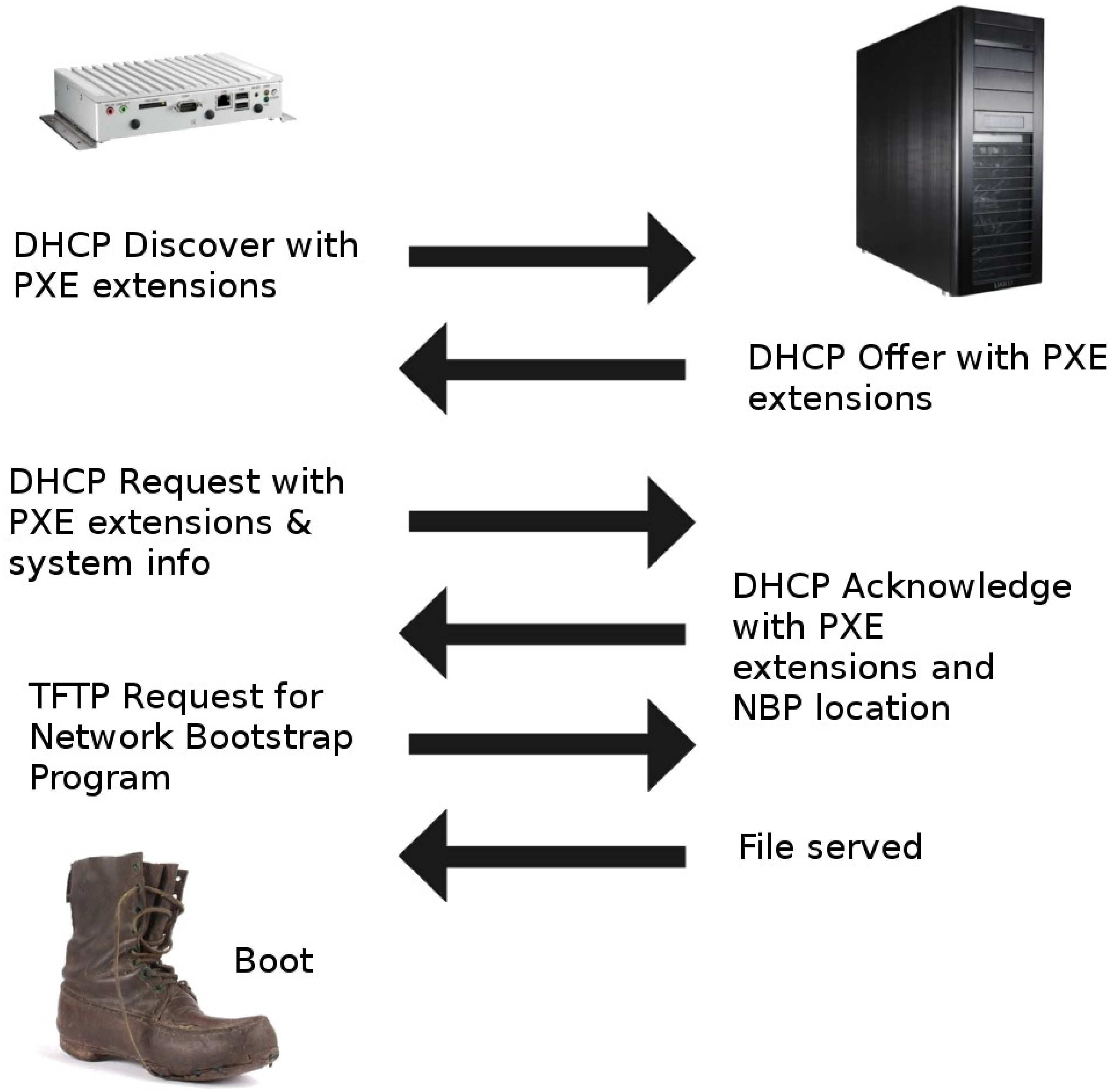
Mount the ext3 FS into your exported root

```
mount -t ext3 -o loop core-image-minimal-atom-pc.ext3 ↲
     /nfsroot
```

Deployment | PXEBoot

PXEBoot allows you to boot your image over the network.

Deployment | PXEBoot


DHCP Discover with PXE extensions
DHCP Offer with PXE extensions
DHCP Request with PXE extensions & system info
DHCP Acknowledge with PXE extensions and NBP location
TFTP Request for Network Bootstrap Program
File served
Boot

Deployment | PXEBoot

You will need:

`sudo apt-get install dhcp3-server tftpd-hpa libgssglue1` **`syslinux initramfs-tools`** `isc-dhcp-server libtirpc1`

And set `/etc/dhcp/dhcpd.conf` to the following:

```
allow booting;
allow bootp;

subnet 192.168.2.0 netmask 255.255.255.0 {
  range 192.168.2.xxx 192.168.2.xxx;
  option broadcast-address 192.168.2.255;
  option routers 192.168.2.xxx;
  option domain-name-servers 192.168.2.xxx;

  filename "/pxelinux.0";
}
```

Deployment | PXEBoot

Setup the Initial RAM Disk

`/etc/initramfs-tools/initramfs.conf`

```
...
# BOOT : [ local | nfs ]
BOOT=nfs
...
# MODULES: [ most | netboot | dep | list ]
MODULES=netboot
...
```

Create the RAM disk

```
mkinitramfs -o ~/initrd_nfs
```

Deployment | PXEBoot

Getting Your Files Straight

```
mkdir -p /tftpboot/pxelinux.cfg

cp /usr/lib/syslinx/pxelinux.0 /tftpboot

cp ~/initrd_nfs /tftpboot

cp /path/to/vmlinux-3.0.24-yocto-standard ↲
     /tftpboot
```

Deployment (Nearly there) | PXEBoot

Create the configuration file

```
/tftpboot/pxelinux.cfg/default
```

Edit as follows

```
DEFAULT linux

LABEL linux
KERNEL vmlinux-3.0.24-yocto-standard
APPEND root=/dev/nfs initrd=initrd_nfs ↲
      nfsroot=192.168.2.1:/nfsroot,rw ip=dhcp rw
```

Ready? Set. Go!

Reboot and Go!

Thanks for Listening

FEABHAS