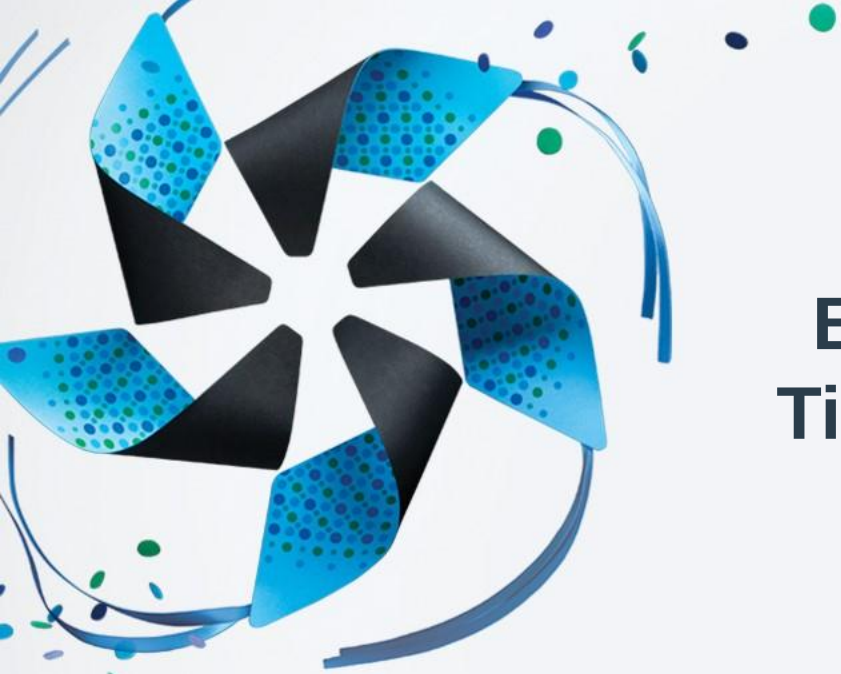




Build your smart device with Tizen-micro by using yocto in only one day

Biao Lu
Austin Zhang



Agenda

- **Current states & existing Issues**
- **Tizen-micro Features**
- **Build your device quickly & flexibly**
- **Case study**
- **Summary**
- **Q & A**

Current states



Images source:

<http://www.rudebague.com/2014/02/13/mobile-smart-devices-set-drive-turnaround-frances-tech-device-market/>

<http://gadgets.in.com/smarthings-make-your-home-smarter.htm>

<http://www.cnn.com/2014/03/28/tech/innovation/smart-device-communication/>

Existing Issues

- Setup development environment is complicated.
- Whole stack knowledge is required.
- Various system components stability is hard to verify.
- Not easy to meet Footprint requirement.



How Can Tizen-micro help
resolve these issues?

What is it?



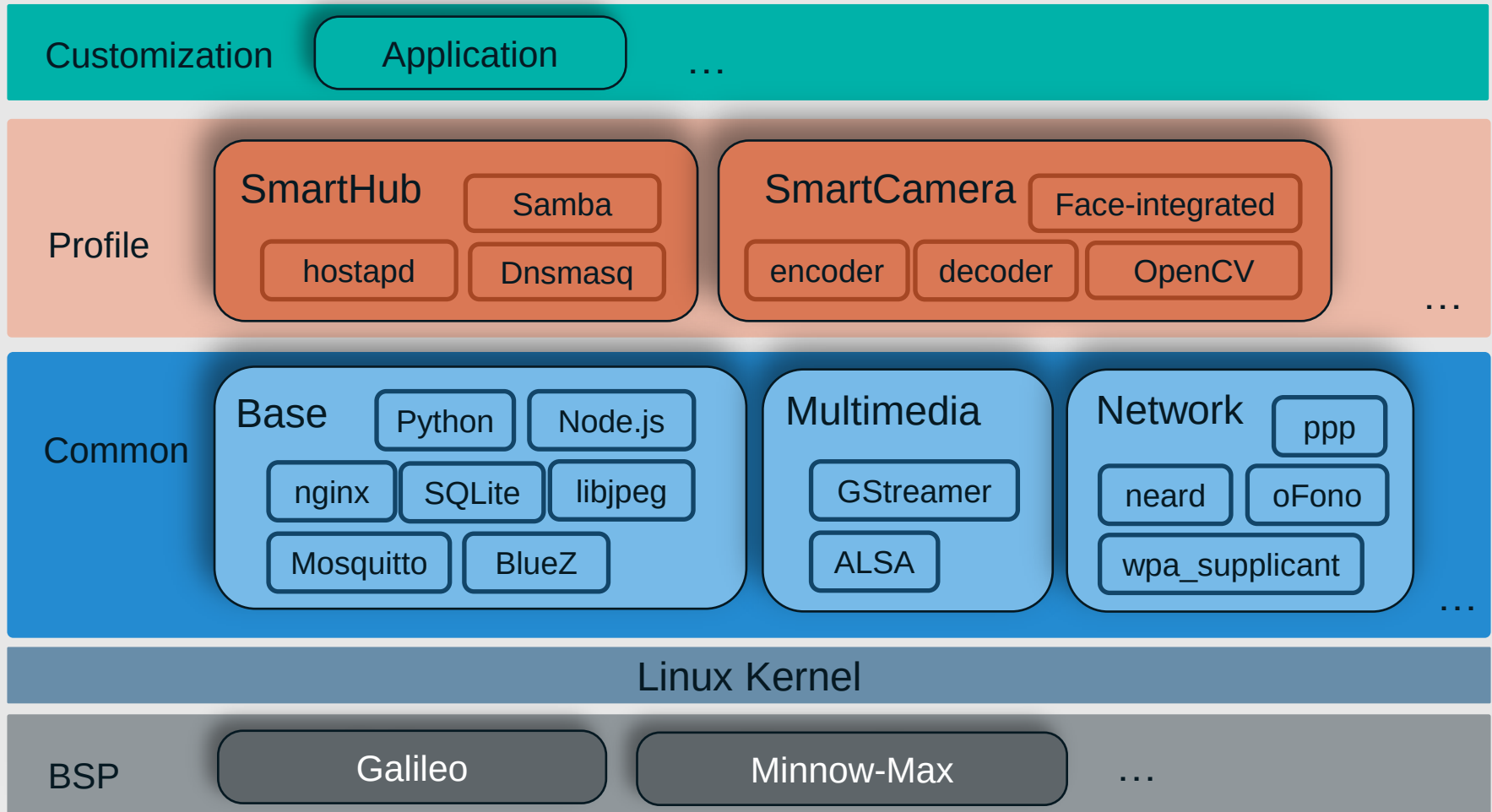
Images source:

<http://www.xconomy.com/wordpress/wp-content/images/2011/11/question-mark-stockimage.jpg>

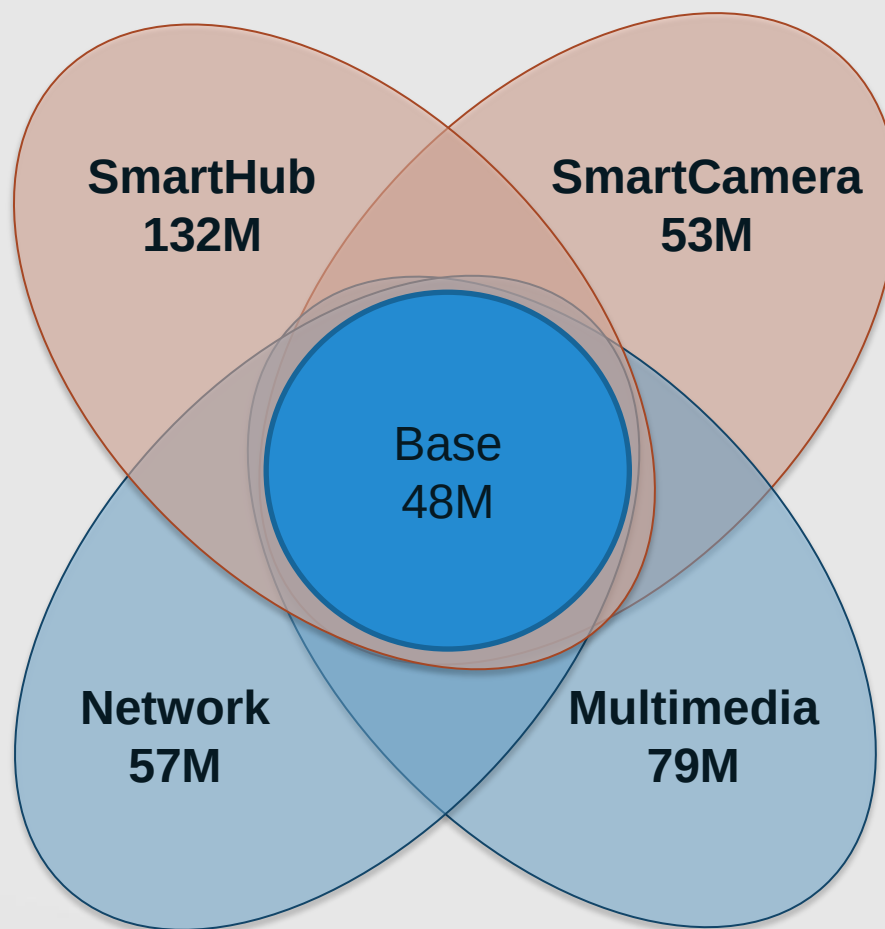
Tizen-micro

- **One Tizen system with below features:**
 - Lightweight (headless) & Small footprint & Multi-tasking
 - Highly configurable & Flexibility
 - Stability & Verification
 - Clear & Quickly build

Tizen-micro Architecture



Tizen-micro Footprint





How to use Tizen-micro to build your device

Build your device quickly & flexibly

- **Get Tizen-micro**

- Get all those codes/layers/pkg group definition/scripts ... to build one Tizen-micro based system

- **Tree structure glance**

```
.├── tizen-micro/│   ├── meta-micro-common/│   ├── meta-smarthub/│   ├── meta-intel/│   │   ├── meta-galileo/│   │   └── meta-minnowmax/│   └── .....├── setup/└── setup.sh
```

Build your device quickly & flexibly (cont.)

- It used Yocto as build tools and some components repo
- The easiest way to create a image

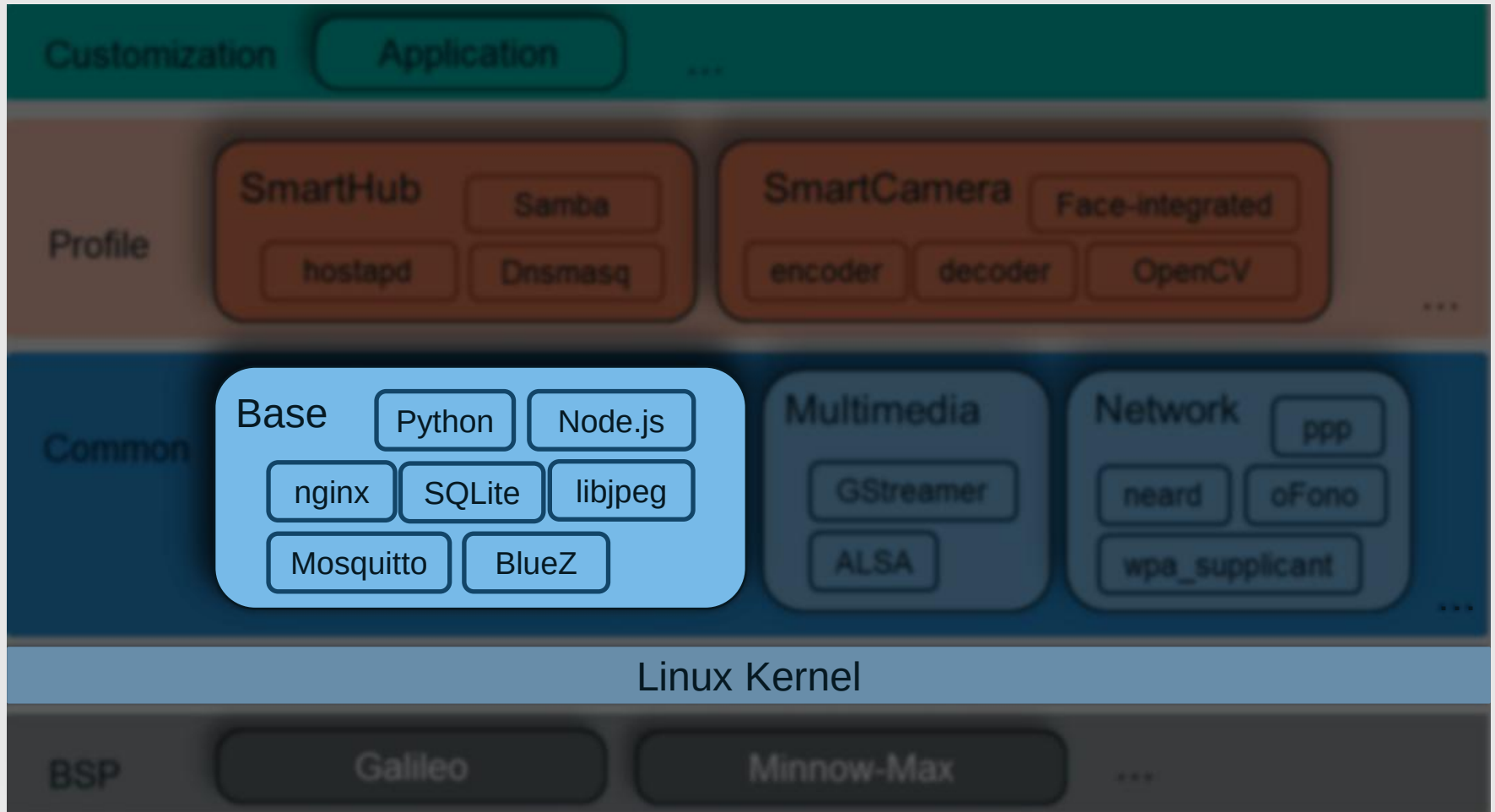
```
$ ./setup.sh  
$ bitbake micro-image
```

- Run in VM

```
# ./setup/vm-run.sh
```

Yocto project: <https://www.yoctoproject.org/>

Base system only



Build your device quickly & flexibly (cont.)

- Add the components as needed in previous image
 - Pre-defined middleware group

```
$ vi meta-micro-common/recipes-core/images/micro-image.bb  
  
IMAGE_INSTALL += "packagegroup-multimedia"
```

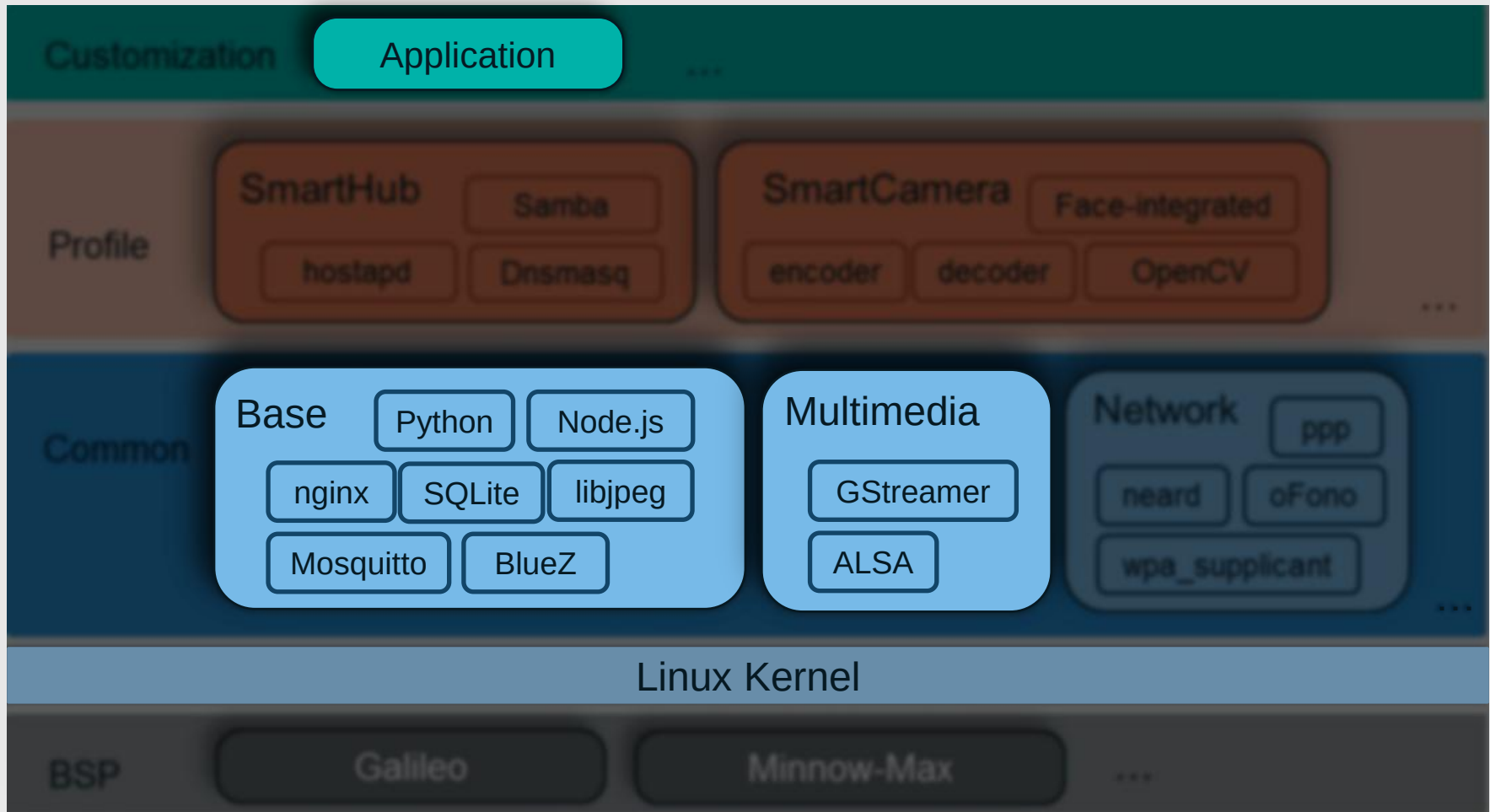
- Your needed middleware components/groups

```
IMAGE_INSTALL += "your-middleware"
```

- Your application

```
$ vi your-app_0.1.bb
```

Add additional middleware and application



Build your device quickly & flexibly (cont.)

- Select BSP & Profile for previous image
 - BSP

```
$ ./setup.sh -b boardname  
$ bitbake micro-image
```

- How to integrate your BSP codes into?

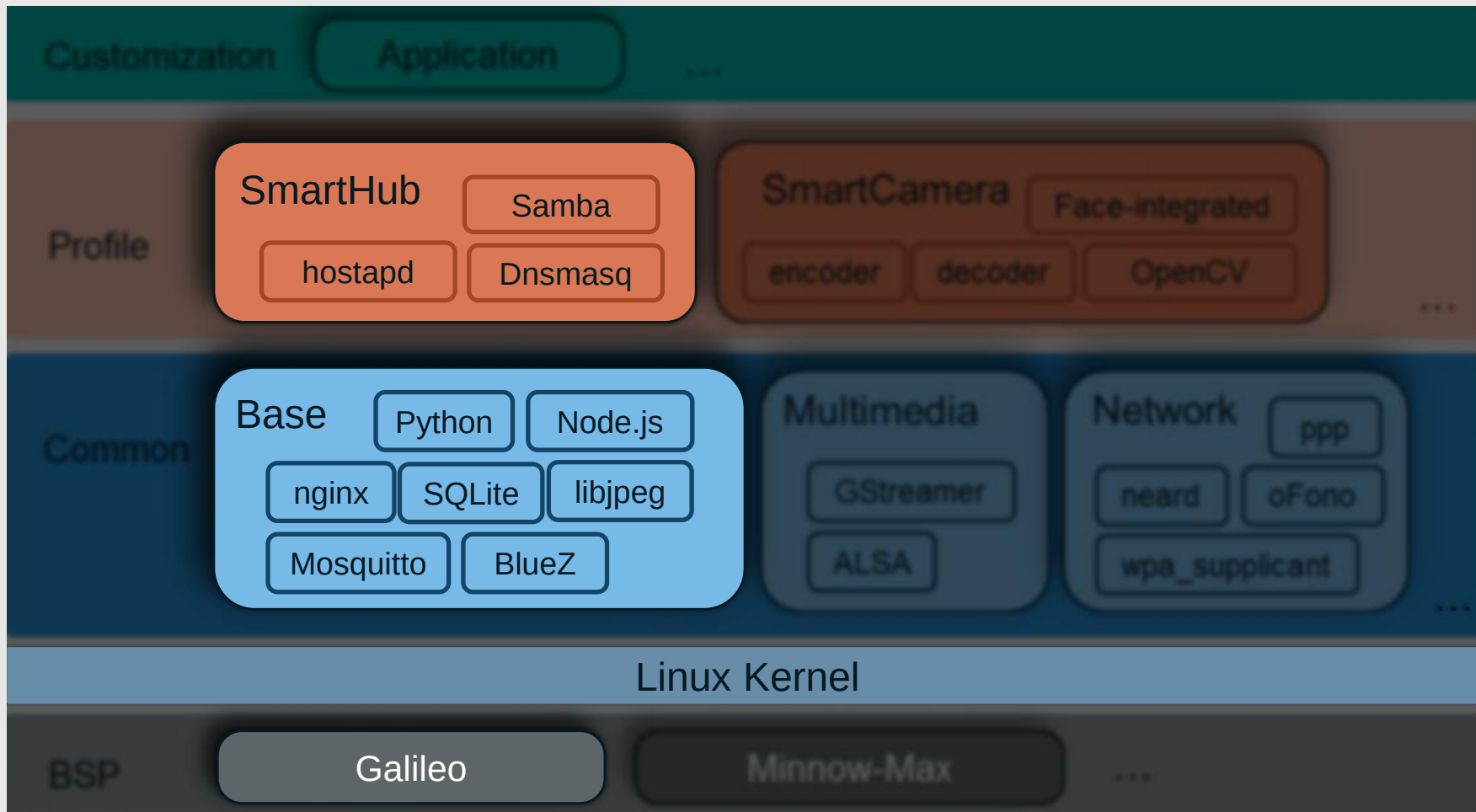
Put BSP under directory "meta-intel", e.g. meta-intel/meta-your-bsp

```
$ ./setup.sh -b your-bsp
```

- Profile

```
$ ./setup.sh -b boardname -p smarthub  
$ bitbake smarthub-image
```

Add BSP / Profile support





Build one smart device from scratch:

we can build our smart device in one very
quick and workable way now!

Case study – Minnow-Max smart hub

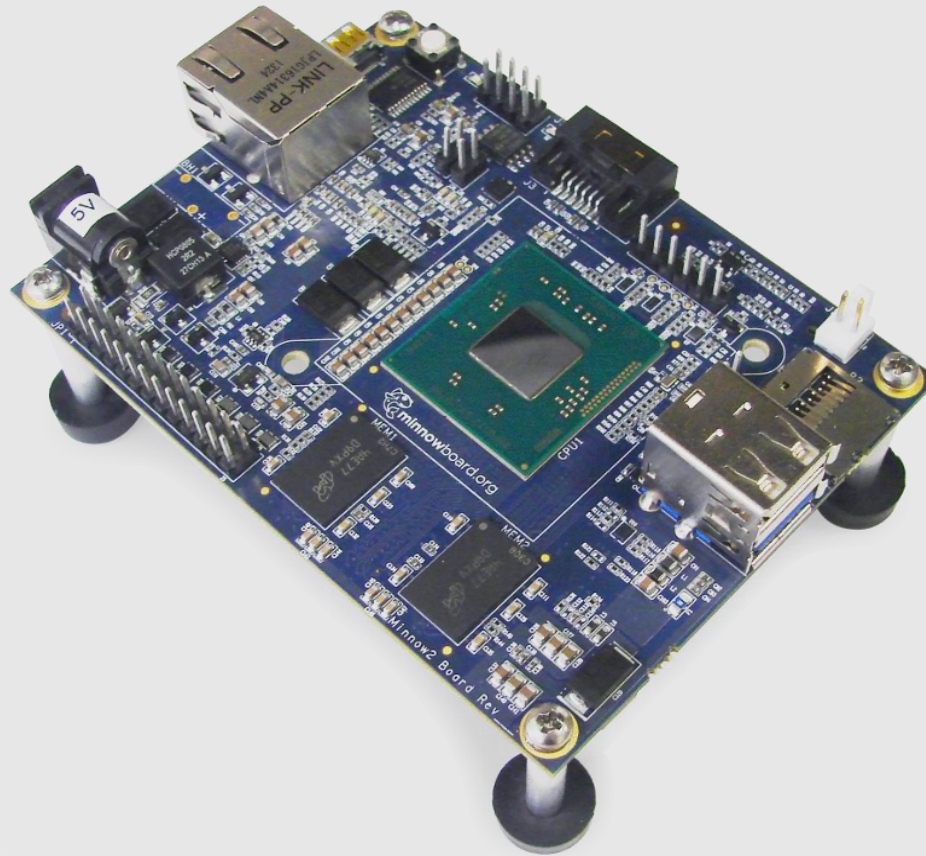


Image from: <http://www.minnowboard.org/meet-minnowboard-max/>

Case study – Minnow-Max smart hub

- One stable and verified base system: Tizen-micro common base
- Additional product profile: smarthub
- Customized UI/UX
- One great HW platform: Minnow-Max

Case study – Minnow-Max smart hub (cont.)

- Create a BSP Layer

```
└─ tizen-micro/  
    ├── ...  
    ├── meta-intel/  
    │   ├── meta-galileo/  
    │   └── meta-minnowmax/  
    └── ...
```

- New Layer config file (layer.conf)

```
BBPATH .= ":${LAYERDIR}"  
  
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \  
            ${LAYERDIR}/recipes-*/*/*.bbappend"  
  
BBFILE_COLLECTIONS += "minnowmax"  
BBFILE_PATTERN_minnowmax := "^${LAYERDIR}/"  
BBFILE_PRIORITY_minnowmax = "5"
```

Case study – Minnow-Max smart hub (cont.)

- New Machine config file (machine/minnowmax.conf)

```
DEFAULTTUNE ?= "corei7-64"  
require conf/machine/include/tune-corei7.inc  
require conf/machine/include/x86-base.inc  
  
PREFERRED_PROVIDER_virtual/kernel = "linux-minnowmax"  
PREFERRED_VERSION_linux-minnowmax = "3.14%"  
  
SERIAL_CONSOLES = "115200;ttyS0"
```

Case study – Minnow-Max smart hub (cont.)

- Kernel config file (linux-minnowmax_3.14.bb)

```
require recipes-kernel/linux/linux-yocto.inc

SRC_URI = "git://git://git.kernel.org...git;branch=linux-3.14.y;"
SRC_URI += "file://minnowmax.patch"
SRC_URI += "file://minnowmax.cfg"

LINUX_VERSION ?= "3.14.4"
LINUX_VERSION_EXTENSION ?= "-minnowmax"

SRCREV_machine = "62f236c734996f240d91daee2cb6a05669c7326c"

PR = "r0"
PV = "${LINUX_VERSION}"

COMPATIBLE_MACHINE = "minnowmax"
```

Case study – Minnow-Max smart hub (cont.)

- Build image

```
$ ./setup.sh -b minnowmax -p smarthub  
$ bitbake smarthub-image
```



Last but not least😊 Summary

Summary

- **With the continuous innovation of smart devices, developers need easy way to enable base system for speeding up to put their ideas into reality.**
- **Tizen-micro offers many features for build your smart device in quickly way.**
 - Full build environment
 - Stable and verified middleware
 - Small footprint
- **Developer can quickly and flexibly build system through simple steps.**
- **Take Minnow-Max as an example to show how to build system using Tizen-micro .**



Q & A



TIZENTM

DEVELOPER SUMMIT 2014



SHANGHAI



TIZEN开发者峰会（上海）

TIZENTM DEVELOPER SUMMIT
2014 SHANGHAI
TIZEN开发者峰会（上海）