

# 10

## Linux Standards

10 ianuarie 2017

1. What does LSM mean?
2. What is the activity of yocto-security mailing list?
3. PREEMPT\_RT is described as:
  - a) Patches applied onto Linux kernel repo
  - b) A Linux kernel repo maintained by companies
  - c) A Linux kernel repo that is now integrated into mainline
4. OpenStack is provided by the following layer:
  - a) meta-virtualization
  - b) meta-cloud-services
  - c) meta-security

1. Linux Security Modules
2. Activity includes identifying the latest and most dangerous security threats(CVEs) and fixing them
3. A
4. B

- Linux standards
- Linux Standard Base
- Carrier Grade Linux
- Carrier Grade Virtualization
- Automotive Grade Linux
- Yocto Project standards support

- Linux virtualization: concentrated on network resources virtualization
- SDN and NFV: mutually beneficial but not dependent on one another
- OPNFV: Open source reference platform for NFV
- ODL: Open source SDN project hosted by the Linux Foundation
- Yocto Project virtualization support: is defined by meta-virtualization and meta-cloud-services layers which are the baselines for network resources virtualization support

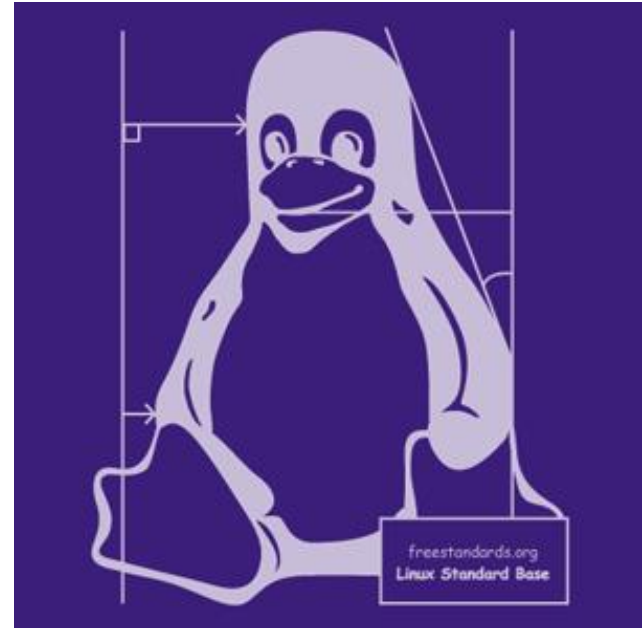
- Linux standards
- Linux Standard Base
- Carrier Grade Linux
- Carrier Grade Virtualization
- Automotive Grade Linux
- Yocto Project standards support

- Linux ecosystem evolved over time
- Most of the development done by companies that use Linux for everyday activities
- Necessity for a common languages for Linux distributions
- LSB – Yocto Project support
- CGL – Yocto Project support
- AGL – Yocto Project support

- Linux standards
- **Linux Standard Base**
- Carrier Grade Linux
- Carrier Grade Virtualization
- Automotive Grade Linux
- Yocto Project standards support



- Linux Standard Base
- Lowers costs on Linux support
  - Reduces the differences between various available Linux distributions
  - Cuts the application porting costs
- A Linux Foundation workgroup
- Standardizes the software system structure
- LSB compliance is certified by a certification procedure
- Binary compatible and produces a stable ABI



The goal of the LSB is to develop and promote a set of open standards that will increase compatibility among Linux distributions and enable software applications to run on any compliant system even in binary form. In addition, the LSB will help coordinate efforts to recruit software vendors to port and write products for Linux Operating Systems.

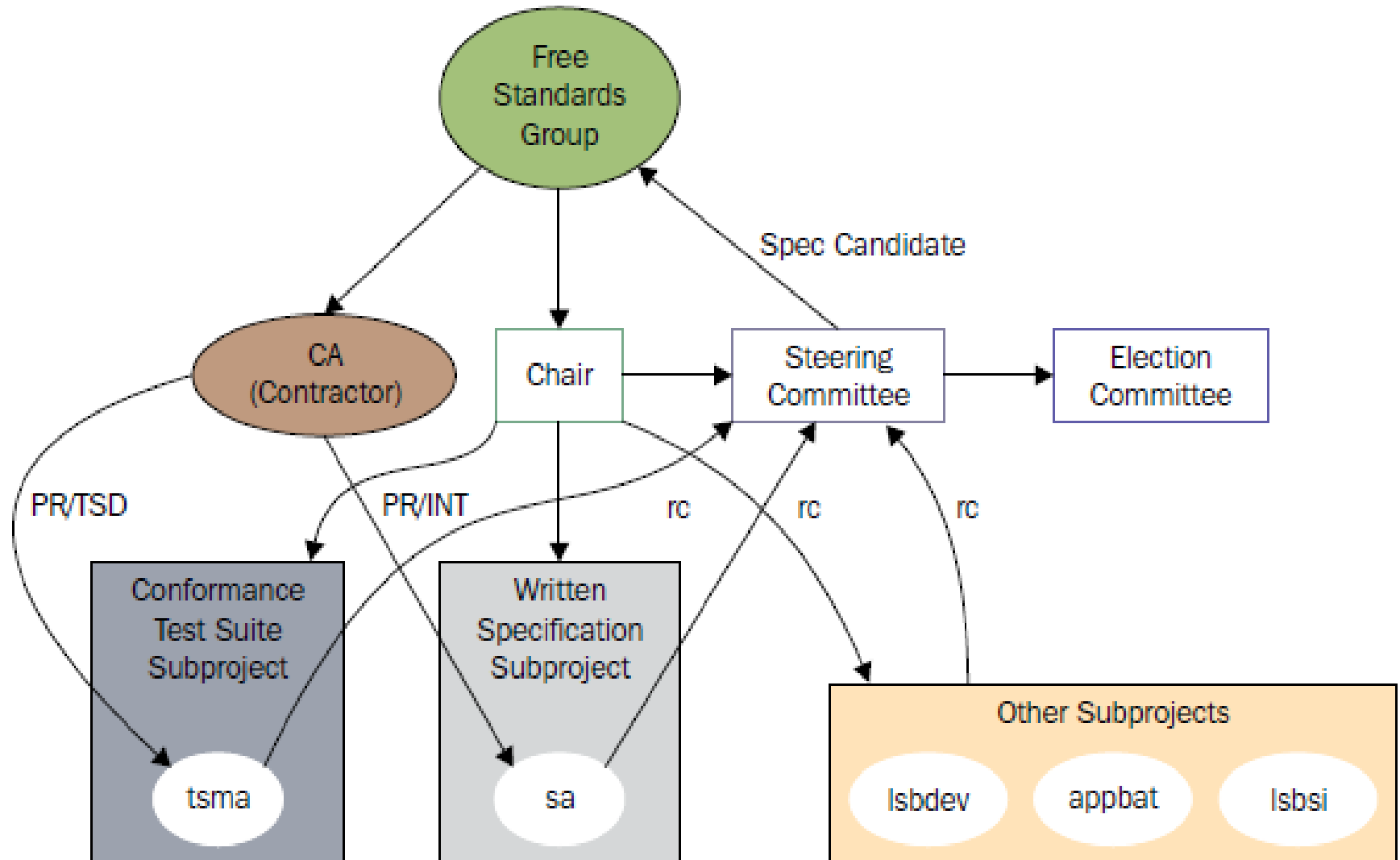
- LSB specifies:
  - Standard libraries
  - Filesystem hierarchy layout
  - Command and utilities that extend POSIX standard
  - Run levels
  - Printing system
  - Several other extensions
- LSB has a modular design
- Has a set of tools and tests to measure support level

- Provides standardization for various Linux areas
- All modules have the same roadmap
- All modules deliver their corresponding set of specifications, software components including conformance test, development tools and other samples
- All modules have a common format
- There are mandatory and optional modules available
  - Mandatory – are the ones that meet the LSB acceptance criteria
  - Optional – work in progress ones, will be included in future LSB standard versions

- There are activities that do not produce LSB modules
  - Integration of patches – for compliance alignment
  - Documentation
- LSB specification release is not done after a schedule
- A testing kit is also released with the new specification
- The result of the testing kit is the LSB certification
- LSB certification comes in two forms:
  - Distribution certification
  - Application certification

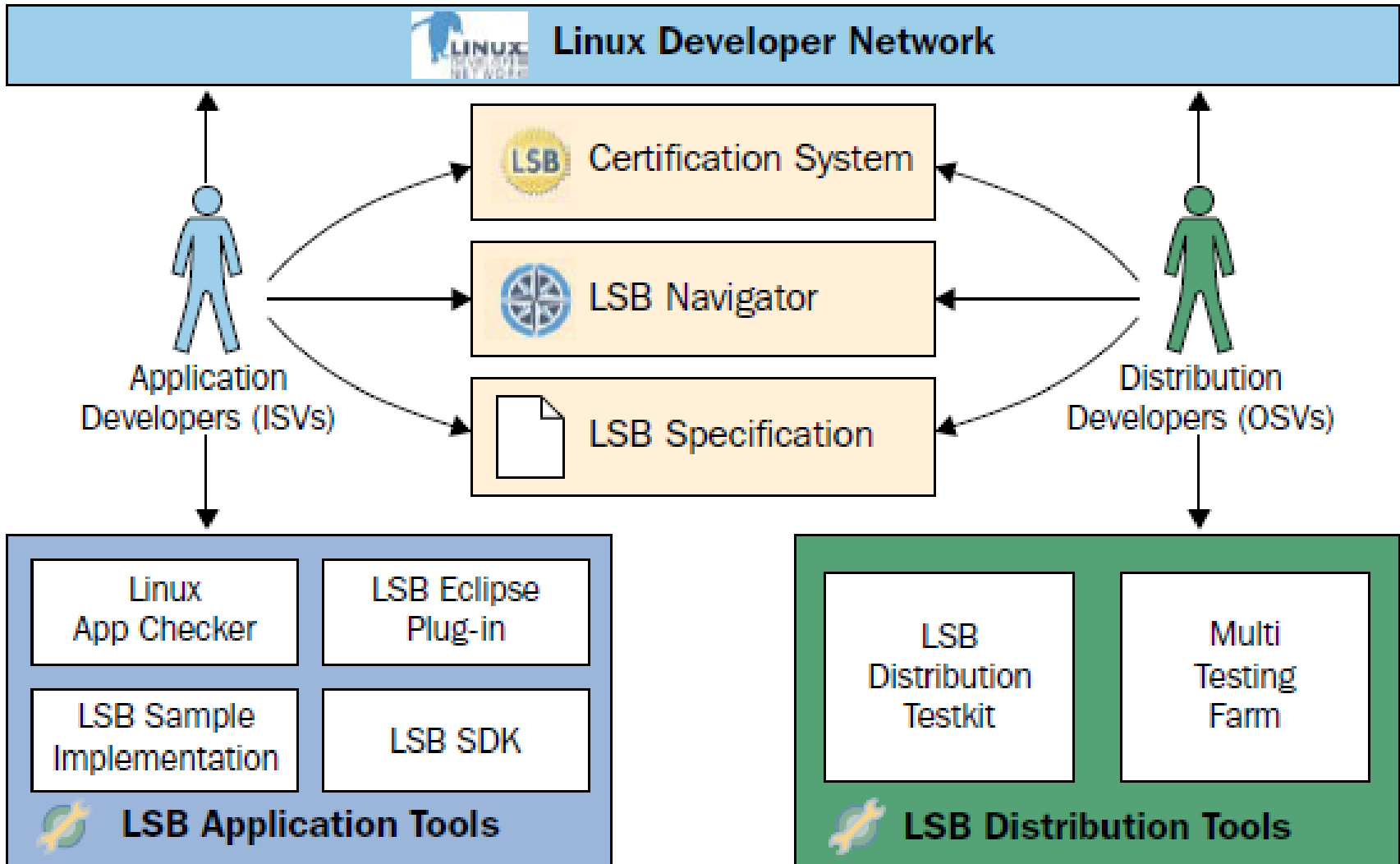
- Steering committee - government
- Chairperson – leader, elected
- Both represent the interest of the group
- Solutions are determined by the Chairperson, if it is not considered valid and does not meet the required criteria for rough consensus, the Steering committee is involved
- Business is carried inside an open forum:
  - Mailing list
  - Conference
  - Wiki page
  - Face-to face meetings

- Contributor: actively involved individuals
- Chairperson: representative project leader, holds position for 2 years, no limits to the number of election times
- Election committee: committee of contributors, established by the steering committee for the Chairperson election
- Steering committee: representative workgroup stakeholders (distribution vendors, OEMs, ISVs, developers and the Chairperson)





- Architecture dependent LSB modules:
  - Core
  - C++
  - Desktop
  
- Architecture independent LSB modules:
  - Core
  - C++
  - Desktop
  - Printing
  - Languages
  
- Some of them are mandatory while others are not



- Through feedback
- Code contribution on the components
- Testing framework and testing infrastructure support
- Bug fixing
- All the above should be done according to LSB specifications

- Linux standards
- Linux Standard Base
- **Carrier Grade Linux**
- Carrier Grade Virtualization
- Automotive Grade Linux
- Yocto Project standards support

- Carrier grade = system, software or even hardware that is extremely reliable
- Reliability standards:
  - Five-nines – available 99.999% of the time, downtime around 5 minutes
  - Six-nines – available 99.9999% of the time, downtime of 30 seconds
- The available support includes
  - CGL
  - CGV
  - AGL

- Carrier Grade Linux
- A Linux Foundation workgroup
- First and oldest option available for carrier grade support
- Necessity for the telecommunication industry
- Open architecture, alternative to the available proprietary closed sourced solutions.
- Represents a set of specifications which when implemented and/or configured would result in a Linux-based OS carrier grade capable

- Initiated by the Open Source Development Lab (OSDL)
- OSDL merged with Free Standard Group and became Linux Foundation
- Currently released version 5.0

- Signaling server applications: includes those products that control support for calls and services. Usually handles around 10000 to 100000 simultaneous connections and requires results under a millisecond
- Gateway applications: provides bridging of technology and administrative domains. Besides the mentioned characteristics also requires to not lose frames of packages in the communication
- Management applications: provide billing operations, network management and other tradition services. Concentrates on fast operations

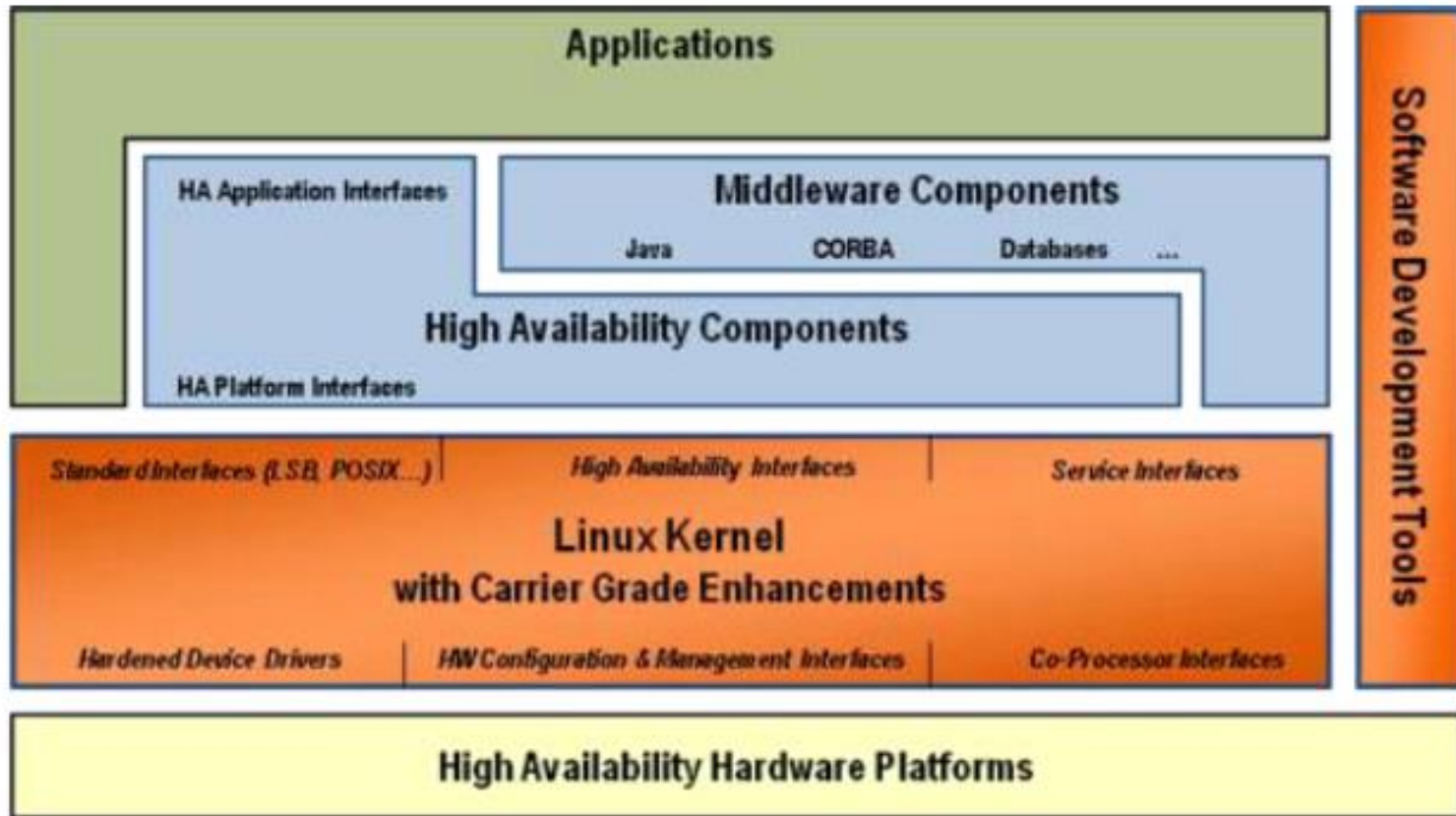


## ➤ Focus:

- Involves communication with the preceding categories, identifying their requirements and writing specifications that need to be implemented
- Gathering and helping projects that meet the defined requirements

## ➤ Workflow:

- CGL certification is offered as a template which defines package version, name and extra information
- The implementation process is not disclosed, the packages can also be proprietary
- CGL tries to represent not only telecommunication industry but also end users and service providers



- Scope of the Carrier Grade Linux Working Group*
- Scope of the Service Availability Forum*

- Availability
- Clustering
- Serviceability
- Performance
- Standards
- Hardware
- Security

- Linux standards
- Linux Standard Base
- Carrier Grade Linux
- **Carrier Grade Virtualization**
- Automotive Grade Linux
- Yocto Project standards support

- Carrier Grade Virtualization
- Derived from CGL
- Vital part of carrier grade platforms available
- Preserves the attributes and performance of the system
- Extends appliance target and derive the benefits of CGL
- Extra information:  
[http://www.linuxpundit.com/documents/CGV\\_WP\\_Final\\_FN.pdf](http://www.linuxpundit.com/documents/CGV_WP_Final_FN.pdf)

- Extends the carrier grade term from telecom to also equipment management, billing security, traffic performance, packaging and powering for development, product support and operations.
- NFV transformed the way networks were built, provisioned and managed
- NVFI products meet the carrier grade key capabilities
- Such products are already available from companies such as Wind River, Telco Systems etc.

- Linux standards
- Linux Standard Base
- Carrier Grade Linux
- Carrier Grade Virtualization
- **Automotive Grade Linux**
- Yocto Project standards support

- Automotive Grade Linux
- A Linux Foundation workgroup
- Started in the end of 2015 managed to stir a lot of interest from the OEM and Tier 1s
- Focuses on defining a reference distribution for automotive
- Tries to enable fast development
- Concerned on things such as IVI, telematics systems, instrument cluster but also on security, remote updates, hypervisors etc.



- A transparent, collaborative and open environment for involved elements
- A Linux operating system stack that is focused on automotive and uses the open source community represented by exponents, such as developers, academic components and companies as back support
- A collective voice for interaction in the open source community released this time in the reverse form, from the AGL to the community
- An embedded Linux distribution used for fast prototyping

- Linux standards
- Linux Standard Base
- Carrier Grade Linux
- Carrier Grade Virtualization
- Automotive Grade Linux
- Yocto Project standards support

- Standards and Yocto Project are an instant match
- Inside Yocto Project is available support for:
  - LSB – available from early existence of Yocto Project
  - CGL – available as a starting point, is mostly developed privately by companies inside their development groups
  - AGL – available as a work in progress, initially started on Tizen, but now ported all on Yocto Project
- Any extra support is encouraged

- Done through the poky-lsb.conf distribution policy configuration
- `git clone git://git.yoctoproject.org/poky.git`
- `git clone git://git.yoctoproject.org/meta-qt3`
- `source oe-init-build-env -b ../build_lsb`
- Define proper MACHINE variable value
- `bitbake core-image-lsb`
- Boot target and after login execute: `/usr/bin/LSB_Test.sh`

- Individual test can also be executed:
  - `cd /opt/lsb/test/manager/utils`
  - `./dist-checker.pl -update`
  - `./dist-checker.pl -D -s 'LSB 4.1' <test_suite>`
  
- Besides `core-image-lsb.bb` recipes the following alternatives are available:
  - `core-image-lsb-sdk.bb` – includes meta-toolchain and necessary libraries and headers needed to generate an SDK for application development
  - `core-image-lsb-dev.bb` – suitable for development work on targets since it the necessary headers and libraries for image-specific packages

- Central piece is the meta-cgl layer
- Includes:
  - meta-cgl-common – offers support for generic machines available inside poky
  - meta-cgl-fsl-ppc – offers support for Freescale BSP specific support. Any other extra BSP requires to have a layer added to this structure
- The supported packages are grouped inside the packagegroup-cgl.bb packagegroup
- The resulting image is called core-image-cgl

- `git clone git://git.yoctoproject.org/poky.git; cd ./poky`
- `git clone git://git.yoctoproject.org/meta-openembedded.git`
- `git clone git://git.enea.com/linux/meta-cgl.git`
- `git clone git://git.yoctoproject.org/meta-qt3`
- `git clone git://git.yoctoproject.org/meta-virtualization`
- `git clone git://git.yoctoproject.org/meta-selinux`
- `git clone git://git.yoctoproject.org/meta-cloud-services`
- `git clone git://git.yoctoproject.org/meta-security`
- `git clone https://github.com/joaohf/meta-openclovis.git`

- `source oe-init-build-env -b ../build_cgl`
  
- Add inside `conf/bblayers.conf` file:
  - `meta-cgl/meta-cgl-common`
  - `meta-qt3`
  - `meta-openembedded/meta-networking`
  - `meta-openembedded/meta-fileystems`
  - `meta-openembedded/meta-oe`
  - `meta-openembedded/meta-perl`
  - `meta-virtualization`
  - `meta-openclovis`
  - `meta-selinux`
  - `meta-security`
  - `meta-cloud-services/meta-openstack`
  
- Add inside `conf/local.conf` file:
  - `BBMASK = "meta-openembedded/meta-oe/recipes-support/multipathtools"`
  - `DISTRO = " poky-cgl"`
  - `MACHINE = "qemuppc"`
  
- `bitbake core-image-cgl`



- `repo init -u`  
<https://gerrit.automotivelinux.org/gerrit/AGL/AGL-repo>
- `repo sync`
- `source meta-agl/scripts/aglsetup.sh -h`
- `source meta-agl/scripts/aglsetup.sh -m qemu86-64 agl-demo agl-netboot agl-appfw-smack`
- `bitbake agl-demo-platform`

- In meta-agl:
  - agl-devel: activate development options
  - agl-netboot: enable network boot support through TFTP and NBD (see meta-netboot layer)
  
- In meta-agl-extra:
  - agl-appfw-smack: enables IoT.bzh Application Framework + SMACK + Cynara
  - agl-demo: enable layer meta-agl-demo and meta-qt5 - required to build agl-demo-platform
  - agl-localdev: add a local layer named “meta-localdev” in meta directory and a local.dev.inc conf file if present
  - agl-sota: enable SOTA components and dependencies (meta-sota, meta-fileSystems, meta-ruby, meta-rust are added)

- The usecase/usecases are tested in 5 minutes
- Documentation is available and has an easy to read format – 1p
- The list of defined features is implemented as documented – 2p
- Inspection of the source code and work in general – 0.5p
- Versioning system is used – 0.5p
- Working usecase/usecases – 1p
- Yocto Project integration support state – 3p
  - Do we have a resulting distribution – 1p
  - Are the usecases executed as part of that distro – 0,5p
  - What is the state of the code – 0.5p
  - Does the usecase/usecases work as documented – 1p

?

