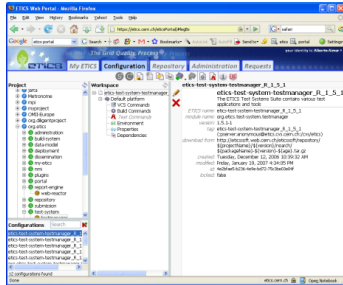# ETICS in Plain English

Lorenzo Dini
CERN

CERN, 24 February 2011

Architecture

Web
Portal

Repository
Web Service

Build/Test
Artefacts

Report/Metrics
DB

Configuration
Web Service

Configuration
DB

Virtual
OS Images

Physical
Worker
Nodes

Execution Engines
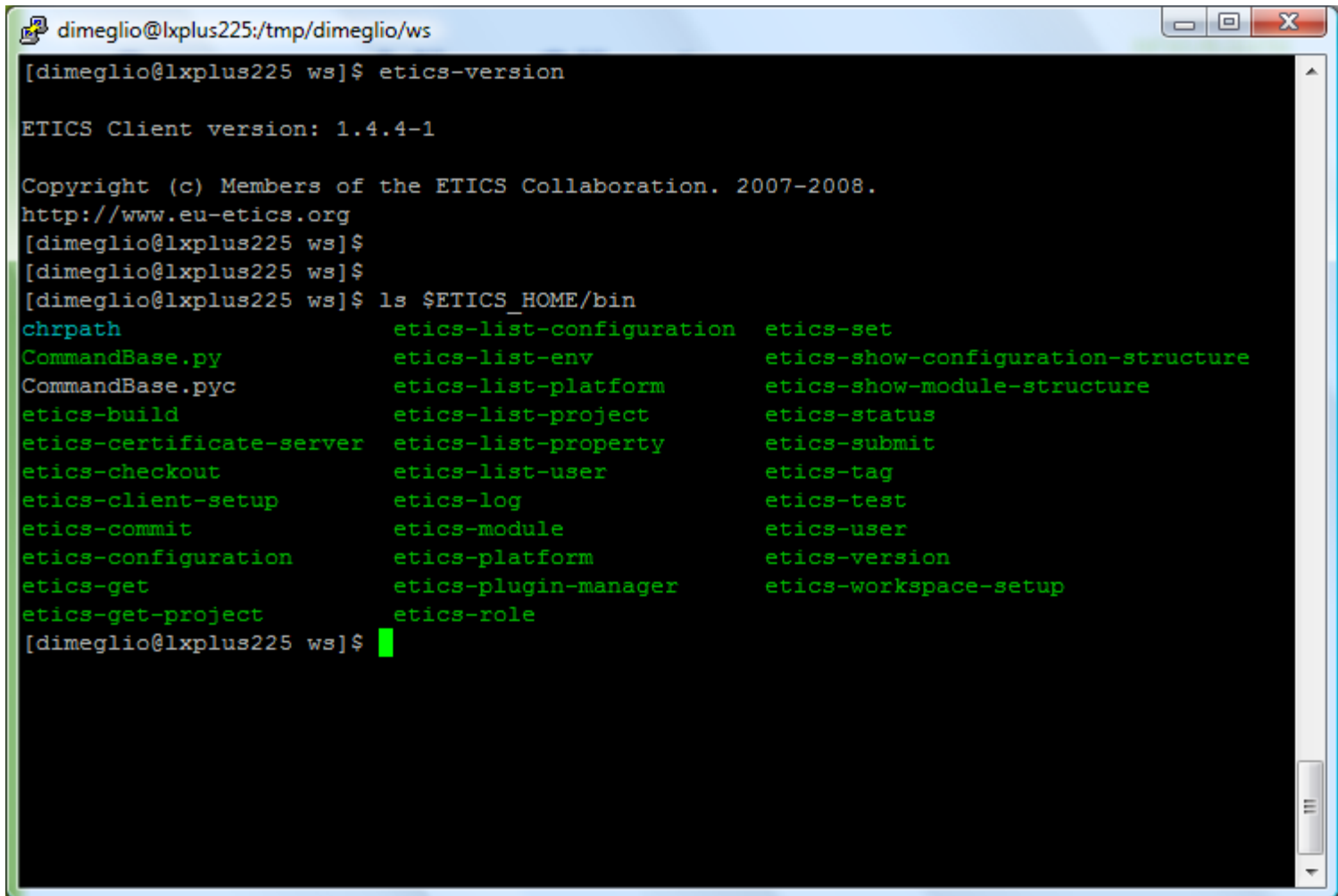(Metronome,
gLite, UNICORE, etc)

Command Line
User Interface

ETICS Infrastructure
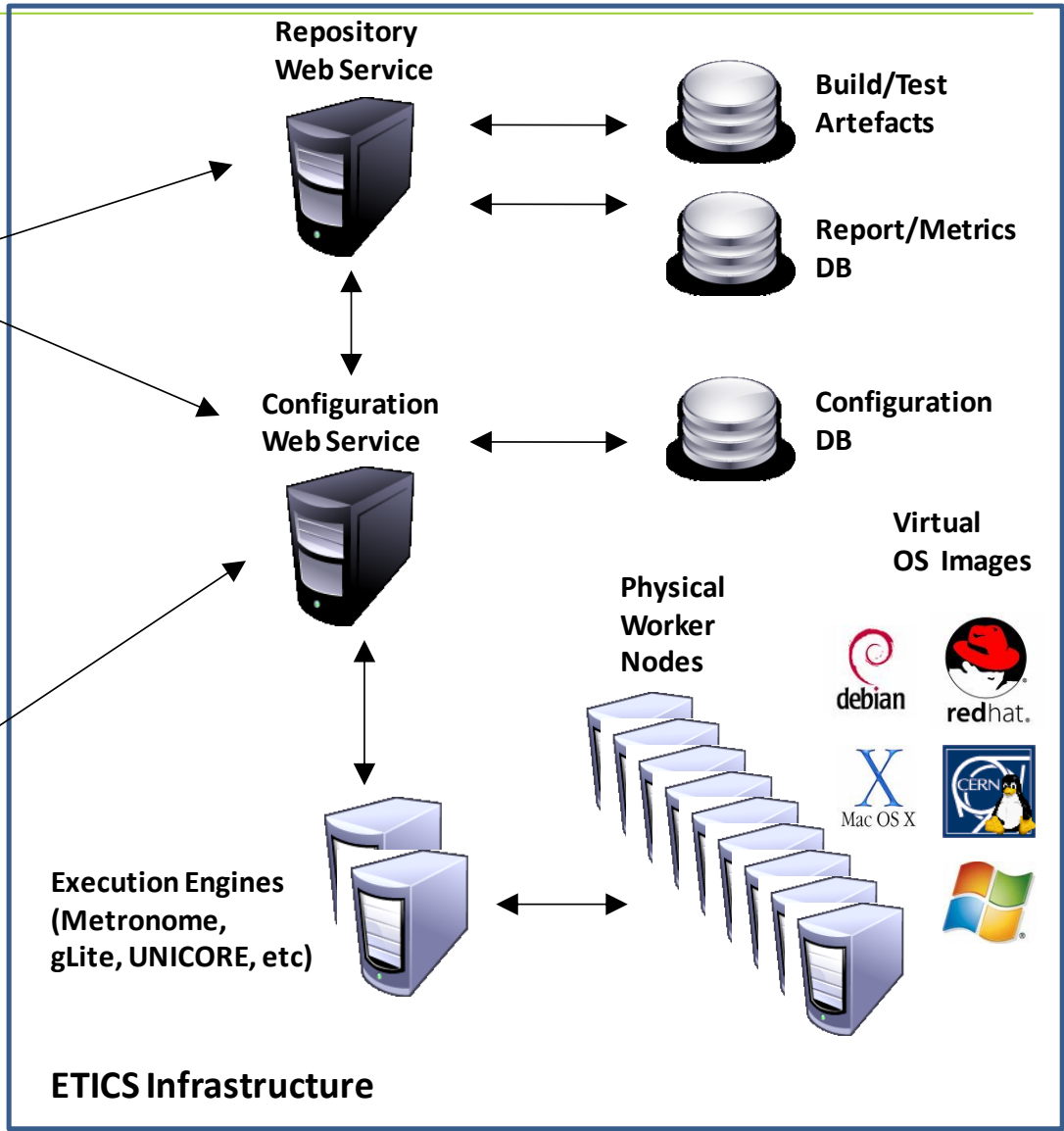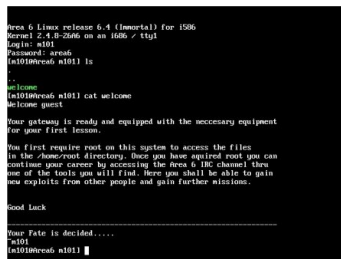
Command-line Client

```
dimeglio@lxplus225:/tmp/dimeglio/ws

[dimeglio@lxplus225 ws]$ etics-version

ETICS Client version: 1.4.4-1

Copyright (c) Members of the ETICS Collaboration. 2007-2008.
http://www.eu-etics.org
[dimeglio@lxplus225 ws]$
[dimeglio@lxplus225 ws]$
[dimeglio@lxplus225 ws]$ ls $ETICS_HOME/bin
chrpath                   etics-list-configuration  etics-set
CommandBase.py            etics-list-env            etics-show-configuration-structure
CommandBase.pyc           etics-list-platform       etics-show-module-structure
etics-build               etics-list-project        etics-status
etics-certificate-server  etics-list-property       etics-submit
etics-checkout            etics-list-user           etics-tag
etics-client-setup        etics-log                 etics-test
etics-commit              etics-module              etics-user
etics-configuration       etics-platform            etics-version
etics-get                 etics-plugin-manager      etics-workspace-setup
etics-get-project         etics-role
[dimeglio@lxplus225 ws]$
```

The ETICS Portal

The Configuration Web Client

The Repository Web Client

Data Model
Component versions and dependencies
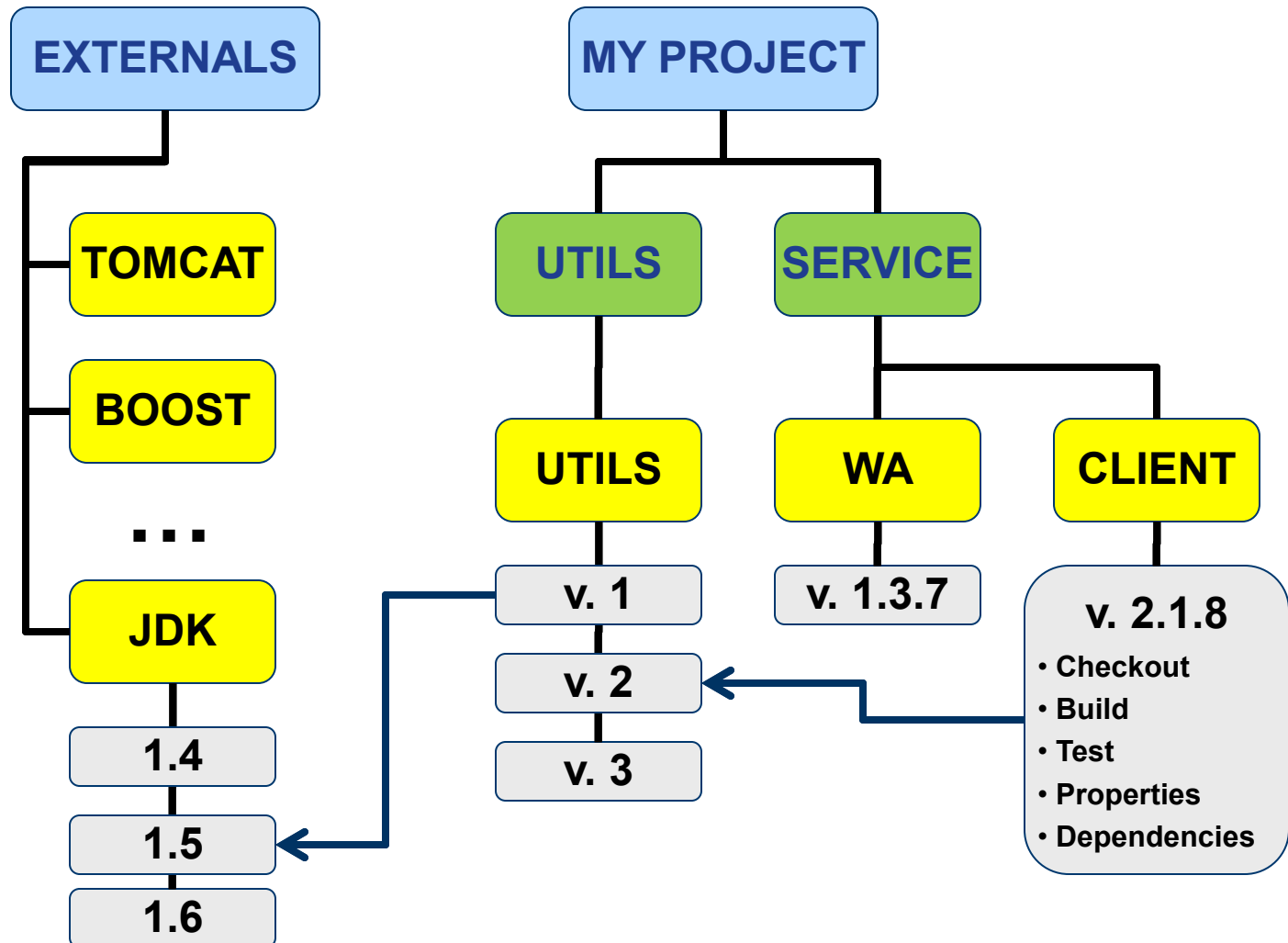
Data Model
Defining releases
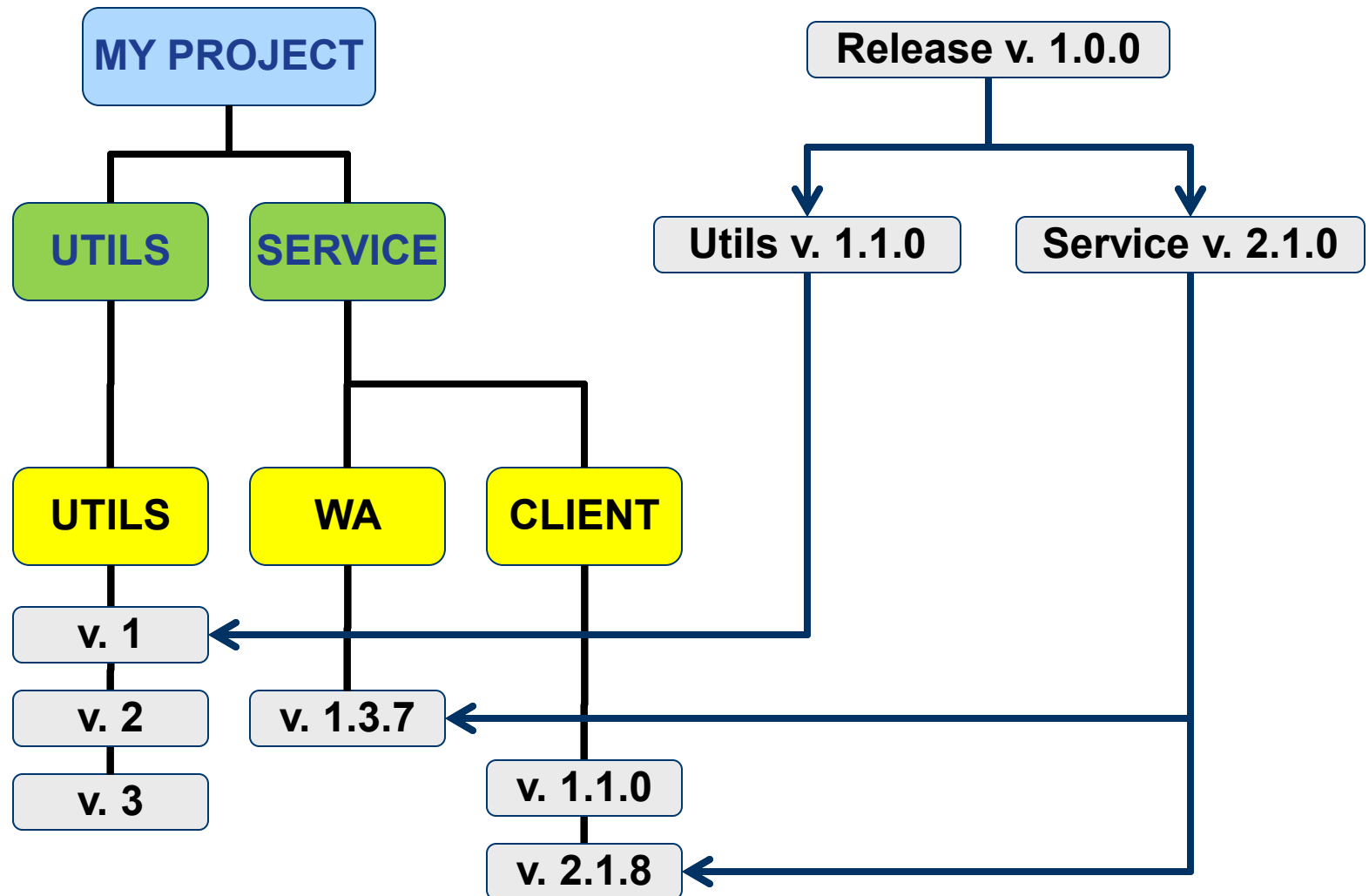
Dependencies

- Automatic build-order resolution

- WHEN?
  - **Build-time:** pointer to binaries made available
  - **Run-time:** REQUIRES added in package

- WHAT VERSION?
  - **Static:** hardcoded by the component
  - **Dynamic:** decided at project level
  - **Version based:** Dynamic with >= X.Y.Z

Properties

- Key-Value Pairs for customization

- **CUSTOM**
    - **Inherited, overwritable**
    - enable.optimization=yes

- **DEFAULT**
    - **Used to manipulate dependencies**
    - MySQL.DEFAULT=mysql v. 5.0.28

- **SYSTEM**
    - **Used to tune the system**
    - package.buildarch=noarch

Typical ETICS Build Execution Sequence

etics-get-project	→	etics-checkout	→	etics-build

Extract metadata information about a project from the ETICS DB

Extract configuration information from the ETICS DB and source/binary packages from different repositories

Execute the build operations

- cvs
- svn
- http

- make
- ant
- Unit tests, coverage, compliance, packaging, reporting
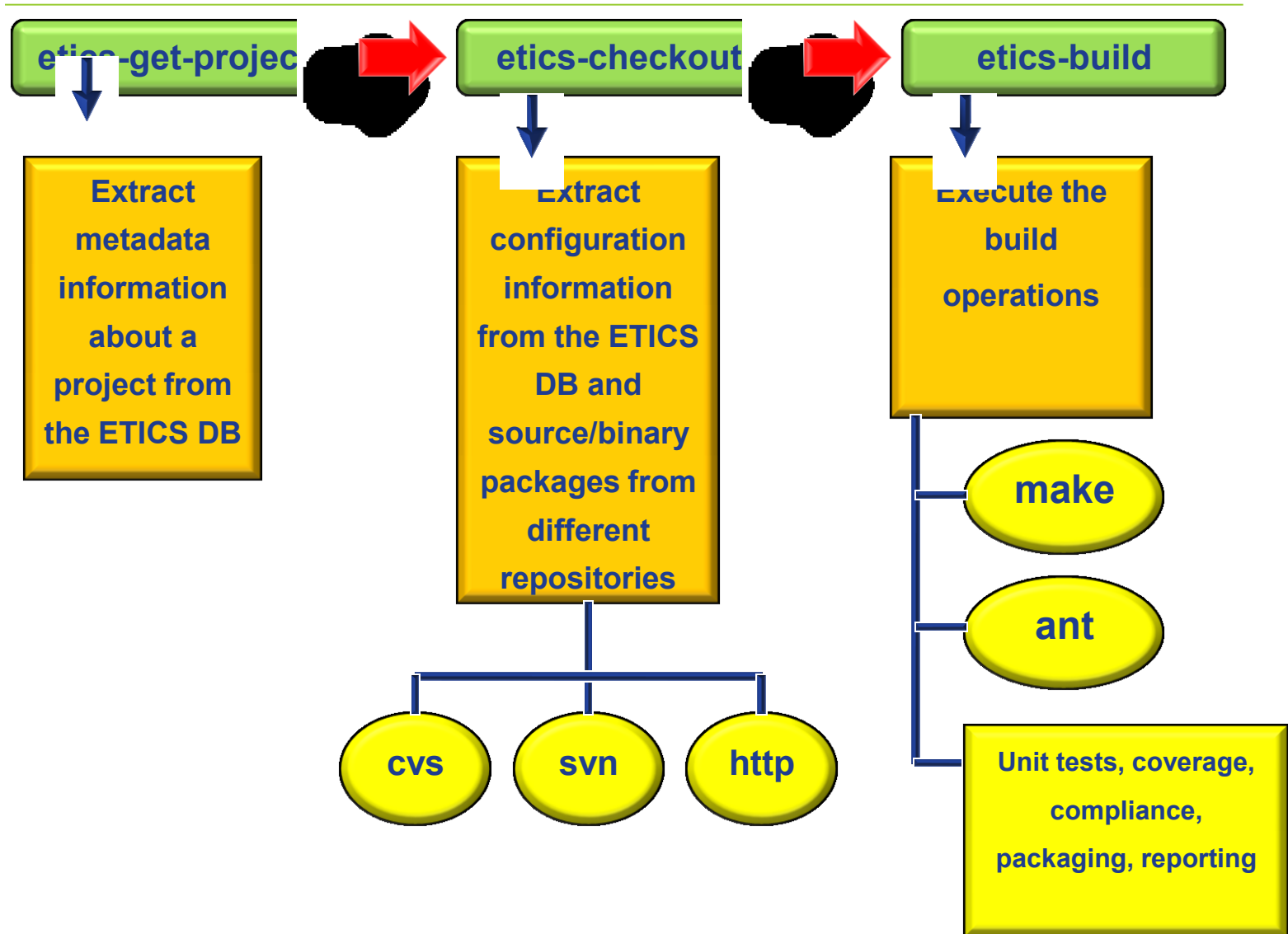
The ETICS Packager

- ETICS contains an automated packaging system, the **ETICS Packager**

- It uses **configuration metadata** to create package or packages for a given platform (RPMS, deb, MSIs, tarballs, zip files etc)

- It covers about 80% of the packaging needs of a typical software component

- It **can be overridden** by the users in various ways to get more specific behaviours

- Developers **focus on the code** not the packaging for N different packaging systems, it allows to **enforce** consistent **packaging guidelines**

The Repository

- ETICS provides an **online repository** where all software packages and build, test and QA reports are stored

- The Repository generates **dynamic** YUM repository at every build to automate the execution of deployment, integration and functional tests

- It can be **browsed and queried** using a web application with a REST interface (XPATH and SQL syntax), a command-line client or a mountable file system (AFS-based)

Plugins and Metrics Collectors

- The ETICS system is **plugin-based**

- Plugins are **wrappers** around existing or custom tools providing specific functionality

- Can be activated by **user choice (profiles)** or letting ETICS decide which tools are **most appropriate (autoprofile)**

- Plugins can publish information in the ETICS system as **metrics** used to do trend analysis and quality monitoring using the ETICS reporting tools

Examples of metrics collectors

Metrics	Type	Programming languages/ technologies	Tool	ETICS Plugin
Complexity	static	Java Python	Javancss	JCcnPlugin PyComplexityPlugin.py
Design quality	static	Java	Jdepend	JDependPlugin
N of potential bugs	static	C/C++ Python Perl PHP Java	Flawfinder, RATS PMD Findbugs	CFlawfinderPlugin CPyPhpRatsPlugin JPmdPlugin JFindbugsPlugin
N of potential bugs	dynamic	C/C++	Valgrind	CValgrindPlugin
Lines of code	static	All	SLOCCount	SLOCCountPlugin
Coverage	dynamic	Java	Emma, Cobertura PyCoverage, LCOV	JUnitemmaPlugin JCoberturaPlugin
Unit tests success rate	dynamic	Java Python	JUnit PyUnit	JUnitPlugin JUnitreportsPlugin.py PyUnitPlugin.py
Compliance with standards	static	IPv6 WSI		IPv6Plugin WSInteroperabilityPlugin
Profiling	dynamic	C/C++ Java	Jrat Valgrind	JRatPlugin CValgrindPlugin

Thanks!

http://www.eticsproject.eu