Software Deployment

Packages

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Table of contents

1. Package

2. Package Manager

3. Package Development Process

4. Related concepts

5. Overview

Package

Package Manager | Definition

Package Manager $\rightarrow$ **Software tools** automating **software** (packages) **installation**, **upgrade**, **configuration**, and **removal**

Archive file

Package $\rightarrow$ Archives

Archive $\rightarrow$ Single File collecting (multiple) **computer files** and **metadata**.

- Easier **portability**
- Easier **storage**
- Compress files $\rightarrow$ reduce storage.

Package

Package $\rightarrow$ **Archive file** containing:

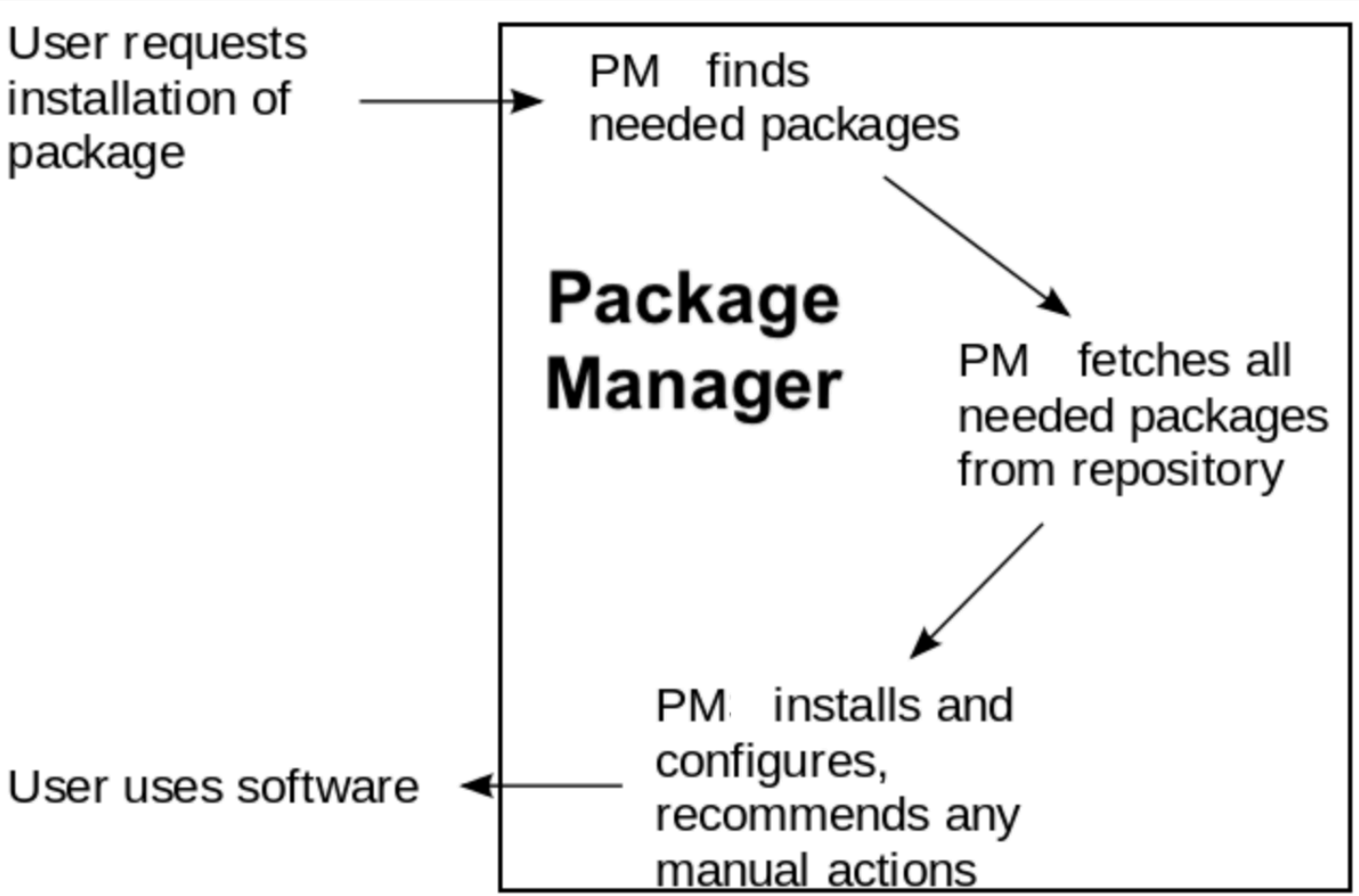
- **Computer programs** (source code/binaries)
- **Additional metadata** used by a **package manager** to **deploy** the **software**

Package Metadata

- Program **name**
- Program **Description**
- Program **Goal**
- Program **Version**
- **Author**
- **Checksum** (integrity check)
- **Dependencies**: required packages

Package Manager

Package manager



User command $\rightarrow$ Find, install, maintain or uninstall software packages.

Package manager functions

- **Eliminate need** for **manual installs/updates**.
- **Download**, **install** and **update** software from **repositories**, **binary repository managers**, and **app stores**.
- Check **authenticity** (digital certificates)
- Check **integrity** (checksums)
- Manage **dependencies** (missing prerequisites)
- Maintain **versions** and **dependencies database** (store metadata in a local package database)
- **Prevents** software **mismatches**

Proprietary vs. open OS package management

Package Management System

- **Open source** (e.g., Linux)
  Use the **same package manager** to **install/upgrade third-party packages**

- **Proprietary OS** (e.g., Mac OS X, Windows)
  Only Upgrade software provided by the **OS owner**

Package Management Systems

- **Debian**:
  Advanced Packaging Tool (APT)
- **Red Hat**:
  yum
- **Arch Linux**:
  pacman
- ...

Commands

- **Commands** are **specific** to each **package manager**
- **Package managers** have **similar functions**.

Repositories

- **Storage location** for **software packages** to be **downloaded** and **installed**.
- For particular **applications** or **OS**

Package Development Process

Package Development Process

Process to **develop** a **package of software**

- **Code**
- **Documentation**
- Some level of **unit tests** / **examples**

Application-level package manager

- Packages can be made for **OS** or **applications** (e.g., **programming languages**)
- **Application-level**: **Smaller** part of the system
- Often **not maintained** by the **OS package manager**

	Package Dev. Proc.	Repository	Installation tool
Python	Setuptools	PyPI	pip, EasyInstall, PyPM, conda
R	R CMD check process	CRAN	install.packages

Related concepts

Installer vs. Package Manager

	Package manager	Installer
Distributed with:	OS	Application
Installation information location	Packages central database	Variable
Formats	Few well-known formats	Not specific
Maintenace	All system packages	Only the application

Package Manager vs build automation

Automatic process to create **software** from **source code**

- **Compile** source code → binary code
- **Package binary code**
- Run **automated tests**

pre-built binary packages can then be **downloaded** and installed by a **package manager**.

Overview

Overview (Carrot2)