

Pacchetti

Applicativi distribuiti in forma sorgente

- le licenze di applicativi software libero garantiscono l'accesso ai sorgenti
 - conseguenza pratica sulla distribuzione: la totalità di tali applicativi sono distribuiti dagli autori in forma sorgente
- procedura tipica di installazione applicativi:
 1. download dell'archivio dei sorgenti [client http/ftp/...]
 2. estrazione dei sorgenti [tar]
 3. configurazione dei sorgenti [configure]
 - verifica automatica delle dipendenze [configure]
 - risoluzione manuale dei problemi derivanti [sysadm]
 4. compilazione da sorgenti a binari [make]
 5. installazione dei binari [make]

Applicativi distribuiti in forma sorgente

- vantaggi
 - uniformità della procedura di installazione (a volte)
 - incoraggia portabilità e studio dei sorgenti
 - flessibilità nella scelta delle opzioni di configurazione
 - paranoia / control freak
- svantaggi
 - lunghi tempi di compilazione
 - necessità di applicativi e librerie a compile-time, non necessari a run-time
 - inefficienza nell'uso di risorse di calcolo comunitarie (sarebbe sufficiente una compilazione per ogni architettura)
 - risoluzione manuale delle dipendenze
 - gestione degli upgrade
 - (configurazione manuale dei sorgenti)

Applicativi distribuiti in forma binaria

- vantaggi
 - duali al caso degli applicativi distribuiti in forma sorgente
- svantaggi aggiuntivi
 - dipendenza dall'architettura
 - dipendenza da librerie dinamiche
 - fiducia in chi ci fornisce il binario

Pacchetti applicativi

- per superare gli svantaggi della distribuzione degli applicativi, ogni distribuzione GNU/Linux offre l'astrazione del “pacchetto”
 - ogni pacchetto rappresenta una componente del sistema
 - sono componenti: applicativi binari, librerie, documentazione, dati, sorgenti, ...
 - ogni pacchetto è associato ad una versione
 - esiste un ordinamento totale tra versioni
 - i pacchetti definiscono la granularità alla quale è possibile aggiungere e rimuovere componenti del sistema
 - esistono relazioni binarie che legano tra loro i pacchetti
 - e.g.: “dipende da”, “è incompatibile con”
- ogni distribuzione offre il proprio sistema di gestione dei pacchetti

Vita di un pacchetto

- esistono due tipi di pacchetti: sorgenti e binari
 - i pacchetti binari sono composti da ciò che è necessario per l'utilizzo a runtime di una componente di sistema
 - e.g. eseguibili binari, librerie compilate
 - i pacchetti sorgenti contengono tutte le informazioni necessarie per generare i pacchetti binari corrispondenti
 - e.g. forma sorgente di un applicativo o di una libreria
- la vita di un pacchetto è articolata nelle seguenti fasi:
 1. rilascio della versione X di una componente
 2. creazione della versione X del pacchetto sorgente
 3. compilazione delle versioni X dei pacchetti binari (un pacchetto binario per ogni architettura supportata)

Utilizzo di un pacchetto

- l'amministratore di sistema non deve necessariamente interagire con la vita di un pacchetto
- ogni sistema di gestione dei pacchetti
 - presenta all'amministratore:
 - l'insieme dei pacchetti (binari) installati
 - un insieme di pacchetti (binari) disponibili
 - permette di eseguire azioni sui pacchetti:
 - installare un pacchetto (binario) non installato
 - disinstallare un pacchetto (binario) installato
 - aggiornare un pacchetto (binario) installato all'ultima versione
 - assicura la consistenza delle relazioni tra pacchetti (binari)
 - e.g. impedisce di installare un pacchetto (binario) senza installare i pacchetti da cui esso dipende

Pacchetti applicativi

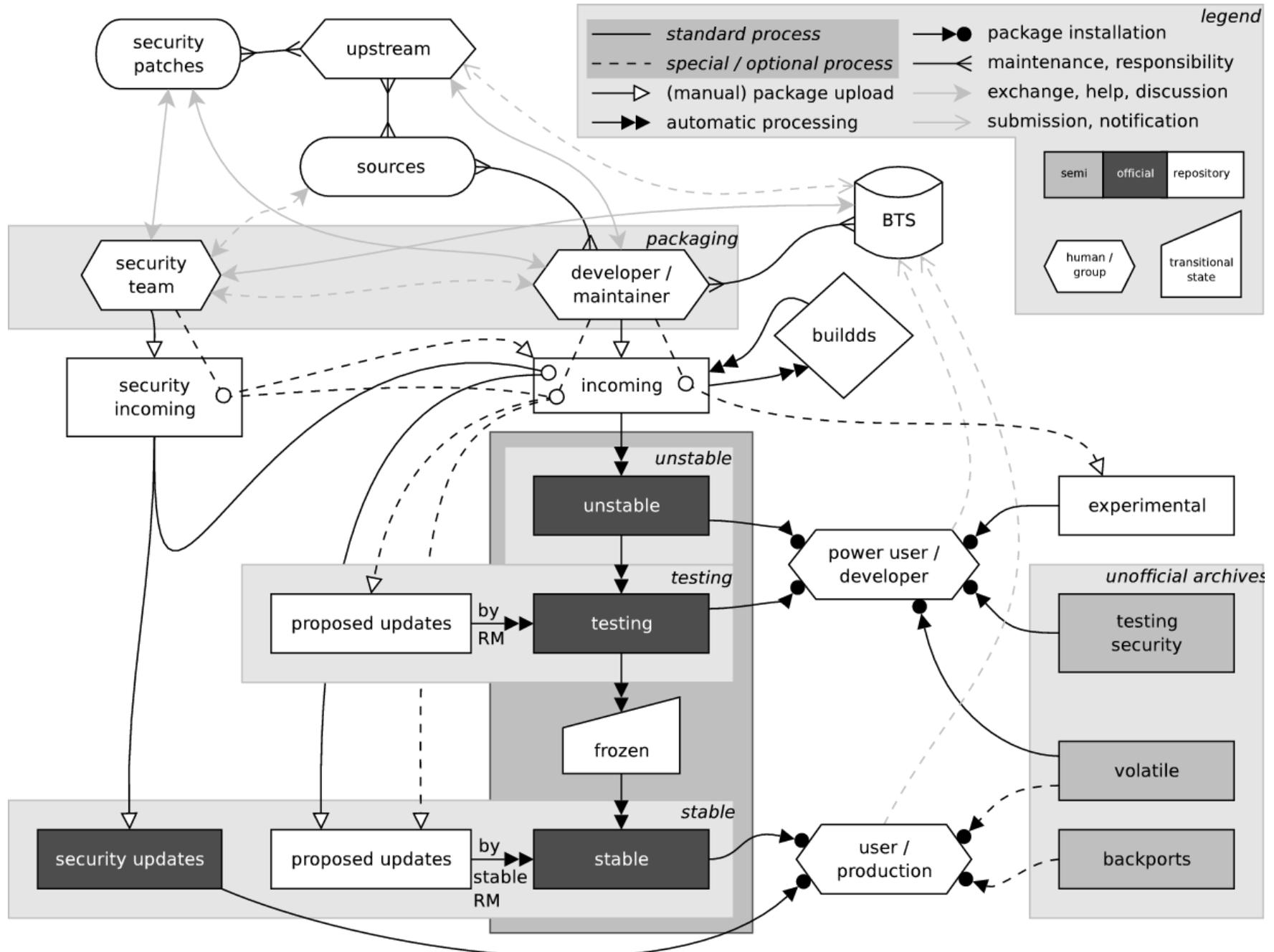
- vantaggi
 - semplicità e velocità di gestione delle componenti di sistema
 - non richiede l'installazione di applicativi e librerie necessari solo compile-time
 - ogni pacchetto sorgente viene compilato una sola volta (per architettura)
- svantaggi
 - fiducia nei manutentori della distribuzione
 - rigidità nella scelta delle opzioni di configurazione
- riferimenti: appunti, capp.38-39

Case study: Debian

- sorgenti vs binari
- dipendenze
 - depends, recommends, suggests, conflicts, replaces, provides
- stato dei pacchetti
 - installed, half-installed, not-installed, unpacked, half-configured, config-files
- azioni sui pacchetti
 - install, remove, purge
- gestione dei pacchetti, la gerarchia Debian:
 - dpkg-deb / dpkg / APT / {dselect,aptitude,synaptic}

Case study: Debian

© "madduck" martin f. krafft, license CC by-sa



Case study: Debian

- comandi: dpkg, apt-get, apt-cache
- riferimenti
 - appunti, cap. 42–46
 - Debian Tutorial
<http://www.debian.org/doc/manuals/debian-tutorial/>
 - APT HOWTO
<http://www.debian.org/doc/manuals/apt-howto/>
 - “The Debian System, Concepts and Techniques”, Martin F. Krafft, Open Source Press,
<http://debiansystem.info/about/book>