



Project Builder:

A GPL continuous packaging solution

Bruno Cornec

HP EMEA EG Presales Strategist

WW Linux Community Lead, HP Open Source Profession

V2.3 June 2015

Introducing Myself



- **Software engineering and Unices since 1988**

- Mostly Configuration Management Systems (CMS), Build systems, quality tools, on multiple commercial Unix systems
- Discovered Open Source & Linux (OSL) & made first contributions in 1993
- Full time on OSL since 1995, first as HP reseller then @HP



- **Currently:**

- OSL Technology **Strategist**, EMEA EG Innovation Solution Center aka HP/Intel Solution Center, Grenoble
- HP OSL **Advocate** and Converged Infrastructure **Ambassador**
- WW Linux Community Lead for the HP **Open Source Profession**
- POSS conference, OpenStack.fr and AFUL board member. Conferences at WW level at LinuxCon, Linux.conf.au
- MondoRescue, Project-Builder.org, UUWL and PUSK **Project Lead**
- LinuxCOE, mrepo, tellico, rinse, fossology, collectl, Ironic **contributor**
- FOSSBazaar/SPDX and OSL **Governance** enthusiast
- Mandriva, Mageia, Fedora **packager**



- **And also:**

- Amateur singer (Alto / Tenor) and recorder player since 1976 and Choir **director** since 1987
- CD collector since 1981 (5000+ and counting) – Concert attendance since 1976
- Amateur photograph since 1976



What is project-builder.org's goal ?



**Make upstream projects life
easier with regards to
packaging their software**

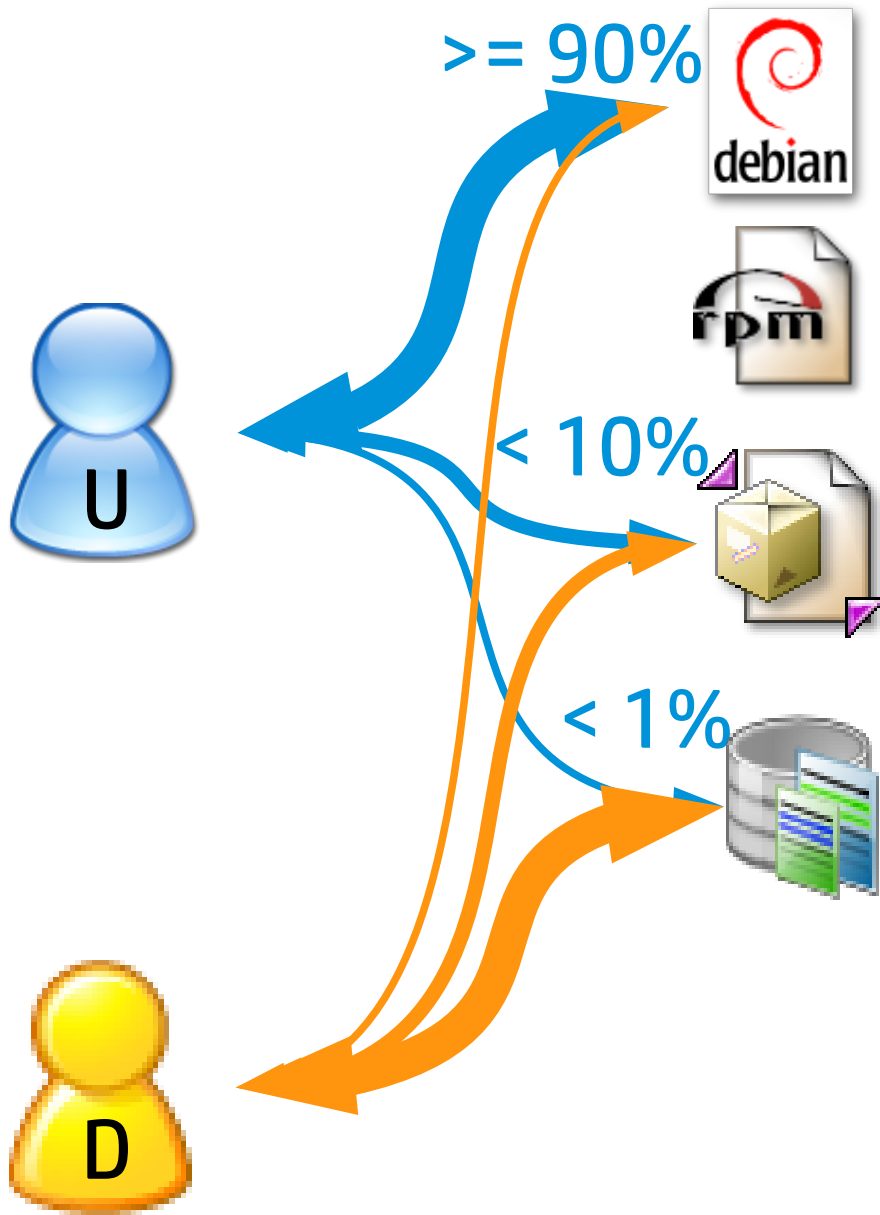
What is project-builder.org's itch ?



**Make my life
easier with regards to
packaging my software**



Users / Sysadmins want packaged software



- Ease of use: GUI, CLI
- Distribution compliance
- Smooth Integration with deployment tools **Ambassador**
- Lag between SW and Pkg availability
- Test of alpha/beta/VCS SW
- Too many:
 - distributions,
 - versions,
 - package formats,
 - tools to generate packages
 - tools to manage repositories

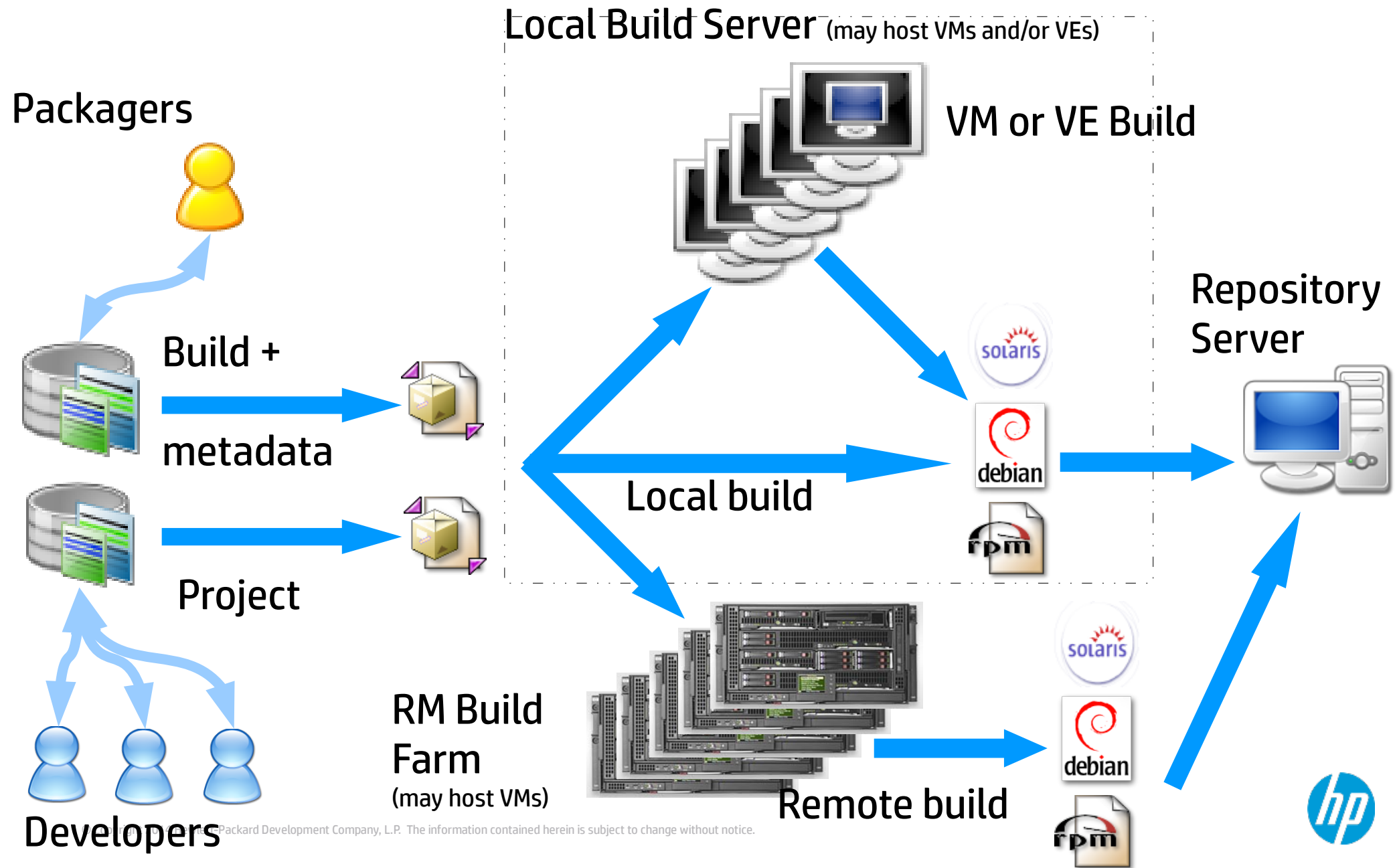
Upstream benefits from Continuous packaging



- **Packaging should be a project concern as well as coding, testing, installing, especially for smaller projects**
- **Packaging as your only way of delivery (not a dream)**
Minimal overhead, maximum benefit:
 - Consistency and reproducibility for devs and users
 - Distribution & deployment server integration,
 - Improved deployment without risk of screwing up the system
- **Packaging as a marketing activity for the upstream project. Easy way to extend your user base, and improve your community relationship and is a “competitive advantage”.**
- **New mantra: “Package early, package always”**
- **THE SOLUTION IS INDEED CONTINUOUS PACKAGING (whatever the tool)**



Continuous Packaging Architecture



Goals



- **Project-builder main goal: help package continuously being agnostic**
- **VCS agnostic:** no VCS but guys it's 21st century now, SVN, CVS, Mercurial, GIT and GIT/SVN, SVK....
- **OS agnostic:** Linux: RPM, deb, ebuild, slack based, ... 150+ distro tuples made and counting – repositories for yum, urpmi, apt. Solaris pkg. HP-UX sd in roadmap
- **Build environment agnostic:** local, VM (QEMU, KVM...), VE (chroot, Docker, rpmbootstrap, rinse, mock, debootstrap...), RM (build farm)
- **No project impact:** preserves the md5sum of the delivered upstream sources. Can be completely external to the upstream project.
- Avoids duplication of code and metadata
- THE SOLUTION IS INDEED **CONTINUOUS PACKAGING** (with project-builder.org !)



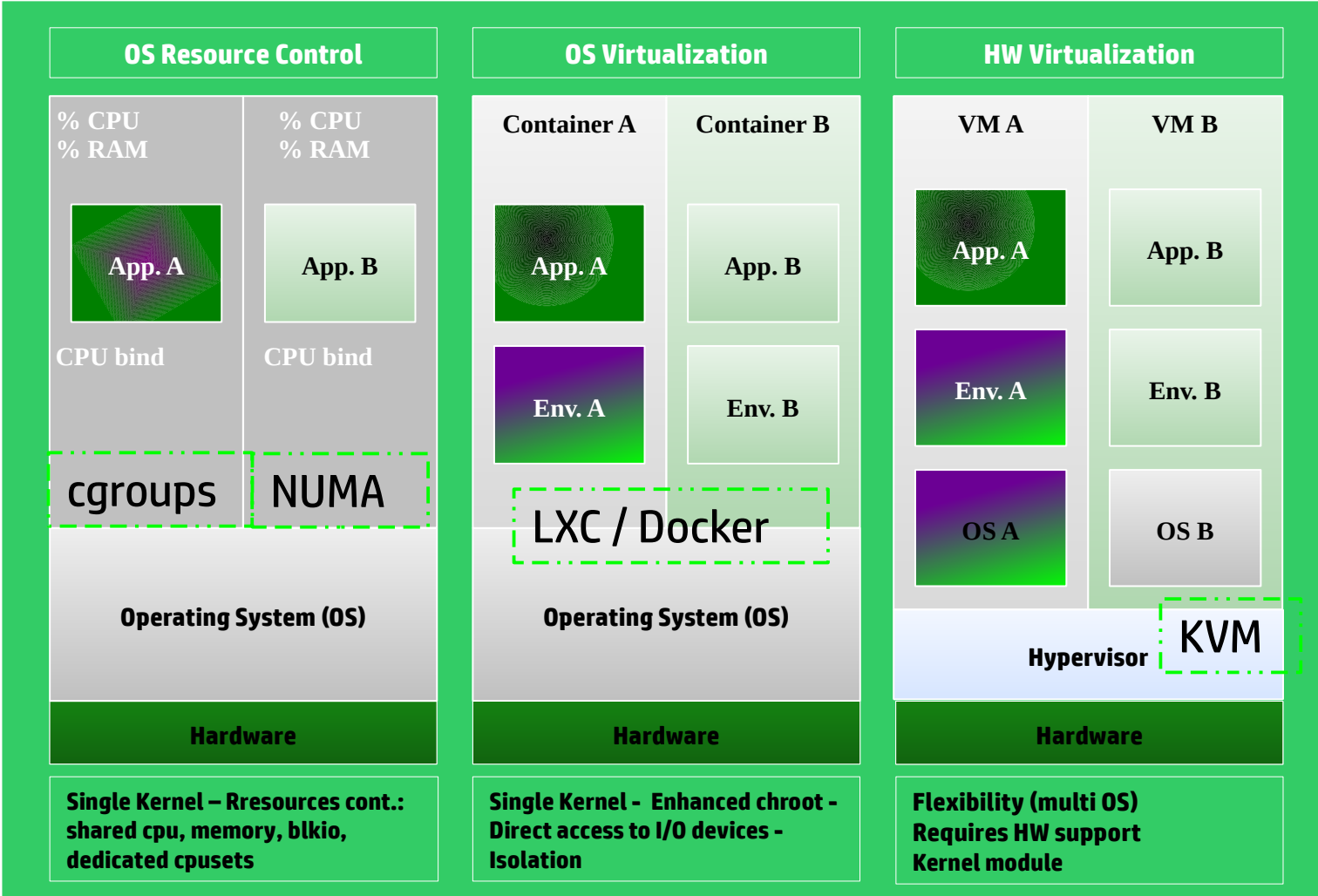
Various Virtualization architectures

Share

Dedicate

OS Virt

HW Virt



performance	good	best	good
isolation security	normal	good	best
runtime flexibility	best	good	normal
maintenance	best	normal	normal

Linux Control Groups

- Help to manage groups of processes
- Dedicated **subsystems** to manage specific resources:
 - **Resource management:** CPU shares, memory, blkio
 - **Resource pools:** CPU sets, ns, freezer, checkpoint/restart
- Implemented as a special cgroup **file system**
- Cgroup commands: cgcreate, cgexec, cgclear, cgclassify, ...
- Cgroup services :
 - cgconfig (*/etc/cgconfig.conf*), defines cgroups and mount points
 - cgroupd (*/etc/cgrules.conf*), defines rules between tasks and cgroups
- These are provided by libcgroup on top of the kernel cgroup virtual file system to allow persistence across reboot and ease of use.
- Docker associates a cgroup to each container

Share

Dedicate

Linux technologies for HW Virtualization



HW virtualization ('KVM' virtual machines)

HW Virt

- Proposed in RHEL 5.4+/RHEL 6/RHEV and TP in SLES 11
- Requires a kernel module and HW support
- Applications in a VM are isolated from the others
 - Security (Svirt)
 - Fully separated environment (own OS, kernel, init, admin, user, network, PID)
 - Paravirtualized drivers, PCI-Attach, SR-IOV.
 - Excellent performances / Full transparency.
- Can use all nice features (NUMA, cgroups, SELinux, iptables...) in the guest, if Linux, and on the host. 1 VM == 1 Linux process
- Being part of the standard kernel : no specific certification required.
- Supports **X2APIC** (reduce guest overhead), transparent **huge pages**, Async I/O



Linux Containers



OS Virt

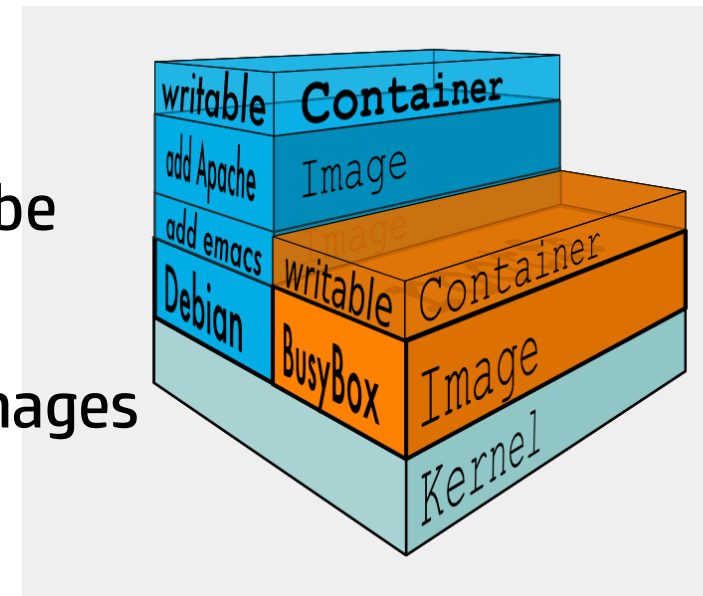
OS Virtualization example: 'LXC / Docker'

- Applications in a zone are isolated from the others
 - Security
 - New namespaces (own admin, user, group, network, PID, init)
 - Granularity (sub-cpu granularity)
 - Virtualization (underlying OS manages HW accesses)
 - Performance / transparency: neither emulation nor full virtualization
- Built upon cgroup resource management and namespaces
- Other technos available
 - V-Server
 - Virtuozzo
 - OpenVZ

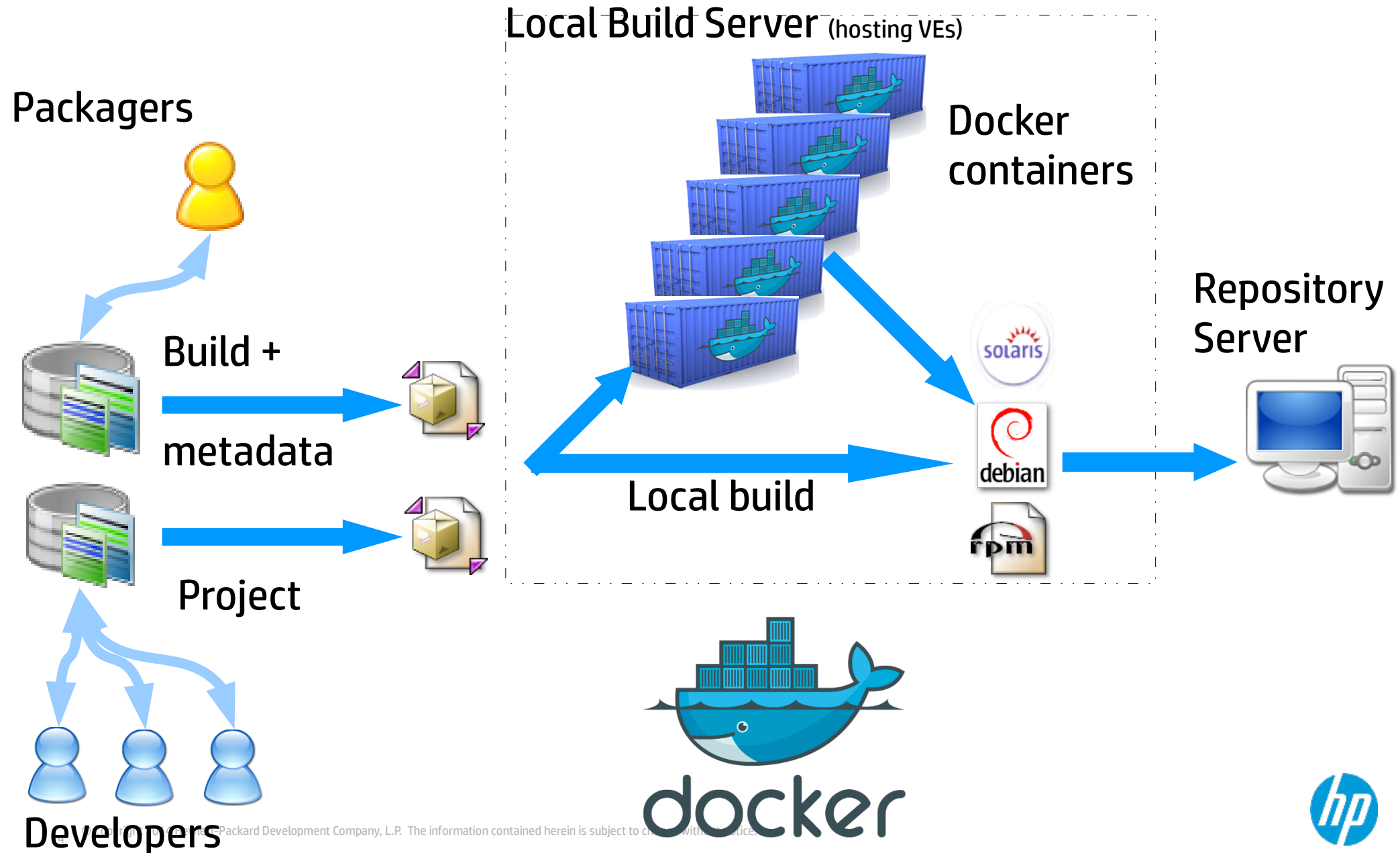


What is Docker ?

- **Docker main goal: run applications in containers in a neutral, lightweight and portable way**
- **Bundle:** Everything packed together
- **Layers:** Images (ro) and containers (rw) can be build upon other images (using a union FS)
- **Registry:** public and/or private registry of images availability
- **Dockerfile:** descriptive build of an image
- **Volumes:** loopback mounting host FS into container
- **Ports:** expose container services port to the host
- 1 application + its deps == 1 container



Continuous Packaging Architecture with docker



- **Project-builder provides additional goodies**
 - Easy VMs/VEs/RMs installation/setup. Update not mandatory.
 - Macro system with perl variables to avoid duplication of metadata
 - Skeleton generation to help starting
 - Manages package delivery up to your repository (ssh based) with repository management (yum, apt, urpmi)
 - Integration of tests in the process
 - Manages patches/additional sources when not upstream
 - Checks validity of packages built (lintian, rpmlint)
 - Easy creation of new versions for upstream management
 - Manages website delivery, announces on mailing-lists

VM, VE & RM Management



● Phase 0

- VE Creation
 - debootstrap for .deb based ones.
 - rpmbbootstrap (modeled after debootstrap): create chroot for multiple RPM based distributions (Mageia, Fedora, Mandriva, CentOS, OpenSuSE)
 - Reuse of existing docker containers (current devs)
- VM|RM Creation
 - KVM or QEMU iso install possibilities.
 - LinuxCOE/OpenStack as provisioning tool for your VM|RM



VM, VE & RM Management (cont.)



- **Phase 1**

- VM|VE|RM setup: adds pb tools and build account

- **Phase 2**

- VM|VE|RM preparation: adds project build dependencies

- **Phase 3**

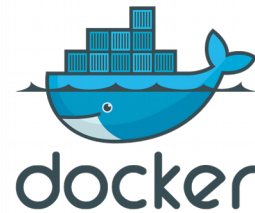
- Build project in VM|VE|RM (== local build for the VM|VE|RM)

- **VM|VE Snapshot feature**

- **ssh communication for VM/RM**



Docker containers Management



- **Phase 0 (tag: d-v-a) - newve**

- Local Image Creation (debootstrap/rpmbotstrap)
- Reuse Image from Docker Registry



Minimal OS Image

- **Phase 1 (tag: d-v-a-pb) - (sbx2|)setupve**

- adds pb tools and build account



Image + pb tools

- **Phase 2 (tag: d-v-a-pb-proj) - prepve**

- adds project build dependencies



Image Proj1 deps



Image Proj2 deps

- **Phase 3 (container) - build2ve**

- Build project in the Docker container



Proj1 container



Proj2 container



Configuration file usage

- **Macro example:**

./rpmbootstrap/pbfilter/all.pbf:filter PBSUMMARY = rpmbootstrap is a tool similar to debootstrap for RPM based distributions

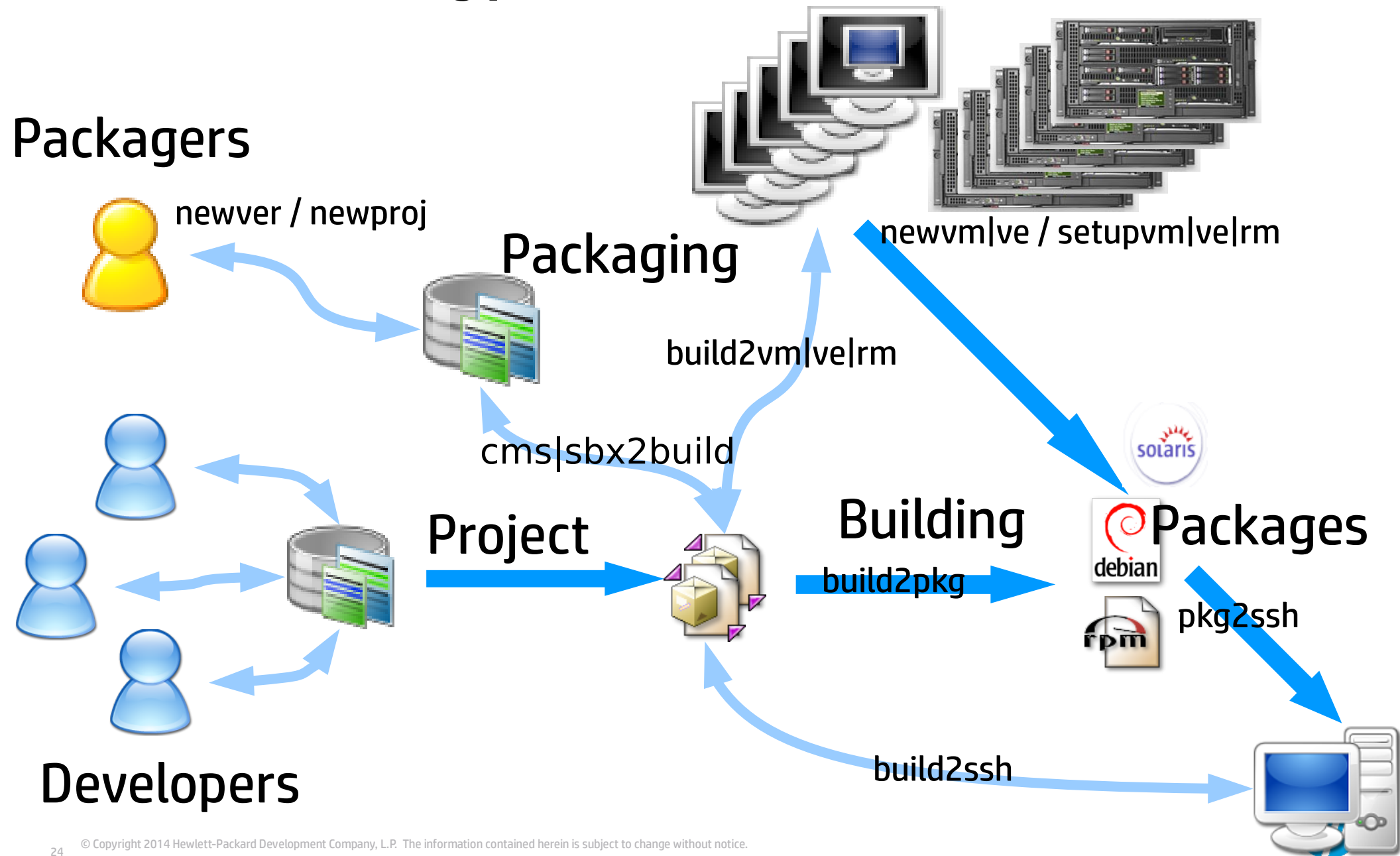
./rpmbootstrap/rpm/rpmbootstrap.spec:Summary: PBSUMMARY

./rpmbootstrap/deb/control:Description: PBSUMMARY

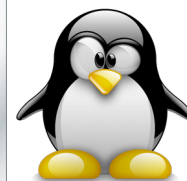
./rpmbootstrap/pkg/pkginfo:NAME="PBSUMMARY"

- **Instantiation possible from distro-ver-arch, distro-ver, distro, distro-family, distro-type, os**
- **Similar approach for all other parameters (Cf: man pb.conf)**
- **Metadata isolated - No duplication**
- **Packaging process independent of the project and external**

The Big picture



Multi platform life cycle



User Requirements

Functional
Specification

Design
Specification

Coding

Unit tests

Integration tests

Validation

solaris

solaris

solaris

solaris

solaris

solaris

solaris

solaris



HP+Intel Solution Brief



One Source for Multiple Platforms: Best Practices

Improving Quality and Reducing TCO on the Road to Intel® Xeon® processors Migration

Common Code Development for Multiple Platforms

For network equipment and software solution providers, developing and maintaining the same applications on multiple platforms are often the result of three use-cases: Migrating to industry standard platforms themselves (Solaris or Linux), dealing with customers moving from SPARC to industry standard platforms, or dealing with customers' heterogeneous systems comprised of all of the above.

If coding in C or C++, at least 90% of the application is identical between a Solaris and a Linux platform.

During a software and solution project development lifecycle, continuous packaging enables the production of identical deliverables at each step of the development process including testing, integration, validation, or final delivery, while addressing and taking into account dependencies and potential conflicts. This very important task generates the need for packaging early and often.

Applications developed and packaged for telecom operators are frequently coded in C or C++, for which at least 90% of the code is identical between a Solaris and a Linux platform¹. HP and Intel's experts took advantage of this favorable condition to reduce costs and enhance overall efficiency through the creation of a tool aimed at easier applications development, packaging, and maintenance on multiple platforms.

¹ "HP's SLPK and SHPK Tools: Enabling rapid migrations from Solaris to Linux and HP-UX", p.17

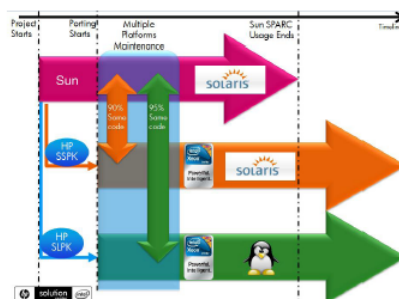


Fig.1: Timeline shows porting and maintenance on multiple platforms

Migration activities with support from HP and Intel

The centralized HP Lazer Team gathers Oracle experts who work closely with infrastructure project leaders on infrastructure management, provisioning, deployment, imaging and other related activities.

In addition the HP System Technology and Software Division (STSD) and team apply its expertise to the development of the Solaris to Linux Porting Kit (SLPK) while the HP Enterprise Services (HPES) is in charge of porting activities, consulting and outsourcing of ported systems and infrastructure.

Intel team of experts contributes at critical junctions when tackling Oracle, Java, compilers, and ported infrastructure tuning topics.

Description of the Development Process

Supporting several target platforms in the development process involves providing tools at two important steps:

Step 1: Code development

The customer may already have an existing process and build chain to produce the Solaris binaries from the source repository. Using the SLPK helps assess the code differences in regards to Linux and provides the low level modules to emulate the Solaris environment and produce the binaries on a Linux environment. The HP Intel CME Solution Center helps with the overall process and with the collaboration of both HP SLPK team and the customer's team.

Step 2: Software Packaging

Since 2007, HP Intel CME Solution Center contributors have developed the open source project-builder.org project in order to support continuous software packaging activities. This tool has been developed to avoid code duplication and provide separation from the project and management in a distributed source repository. In addition, it makes the code agnostic to the build environment and targeted operating system. Eventually, it provides macro and virtual machines/environments features.

Along with the code development, intermediary deliveries are packaged just like final deliveries of the applications. This tool allows customers to both minimize the risk and leverage maximum code re-use when moving from one to multiple platforms, while ensuring the quality of the software.

Steps 1 and 2 will soon be joined in one generic tool, supporting and performing code development and software packaging.

HP & Intel Key hardware components

HP Blade System c7000 Enclosure

The BladeSystem c7000 enclosure provides all the power, cooling, and I/O infrastructure needed to support modular server, interconnect, and storage components today and throughout the next several years. The enclosure is 10U high and holds up to 32 servers or 16 storage blades plus optional redundant network and storage interconnect modules.



The HP Intel Solution Centers provide complete telecom infrastructures for demonstrating the Communications Media and Entertainment Solution Portfolio to HP customers and partners. The centers are located in the three regions: Grenoble, France for EMEA; Plano, Texas, USA for Americas, and Shanghai, China for APJ. These unrivalled technical facilities offer our customers and partners, the unique opportunity to evaluate new services in real-world environments, test new technologies and select the solutions most likely to succeed.

Technology for better business outcomes

© 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein. Copyright © 2008 Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. * Other names and brands may be claimed as the property of others.

For more information, visit <http://www.hpintelco.net>

One Source for Multiple Platforms, January 2010.



Fig.2: Development Process Principles

HP Intel Communications, Media and Entertainment (CME) Solution Center's team and facilities

In association with HP and Intel's teams of experts, the HP Intel CME Solution Center has developed a program on applications migration from SPARC to Intel Xeon processors platforms. Supporting telecom operators and service providers, a migration roadmap has been created, including services and tools such as:

- On-site Innovation Workshops
- Executive and Technical HP Intel Migration Workshops
- Solaris on Intel Xeon processors Proofs-of-Concept and Benchmarks
- Migration Project Management
- Single-source to multiple targets build environment
- Solaris to Linux, Solaris to Solaris Porting Kits

Demonstrations of the software migration tools are available in our facility, as part of workshops conducted at customers' sites or undertaking in-house PoCs using [HP ProLiant G6 servers](http://hp-proliant-g6-servers) based on [Intel Xeon processors 5500 series](http://intel-xeon-processors-5500-series).

Intel® Xeon® processors

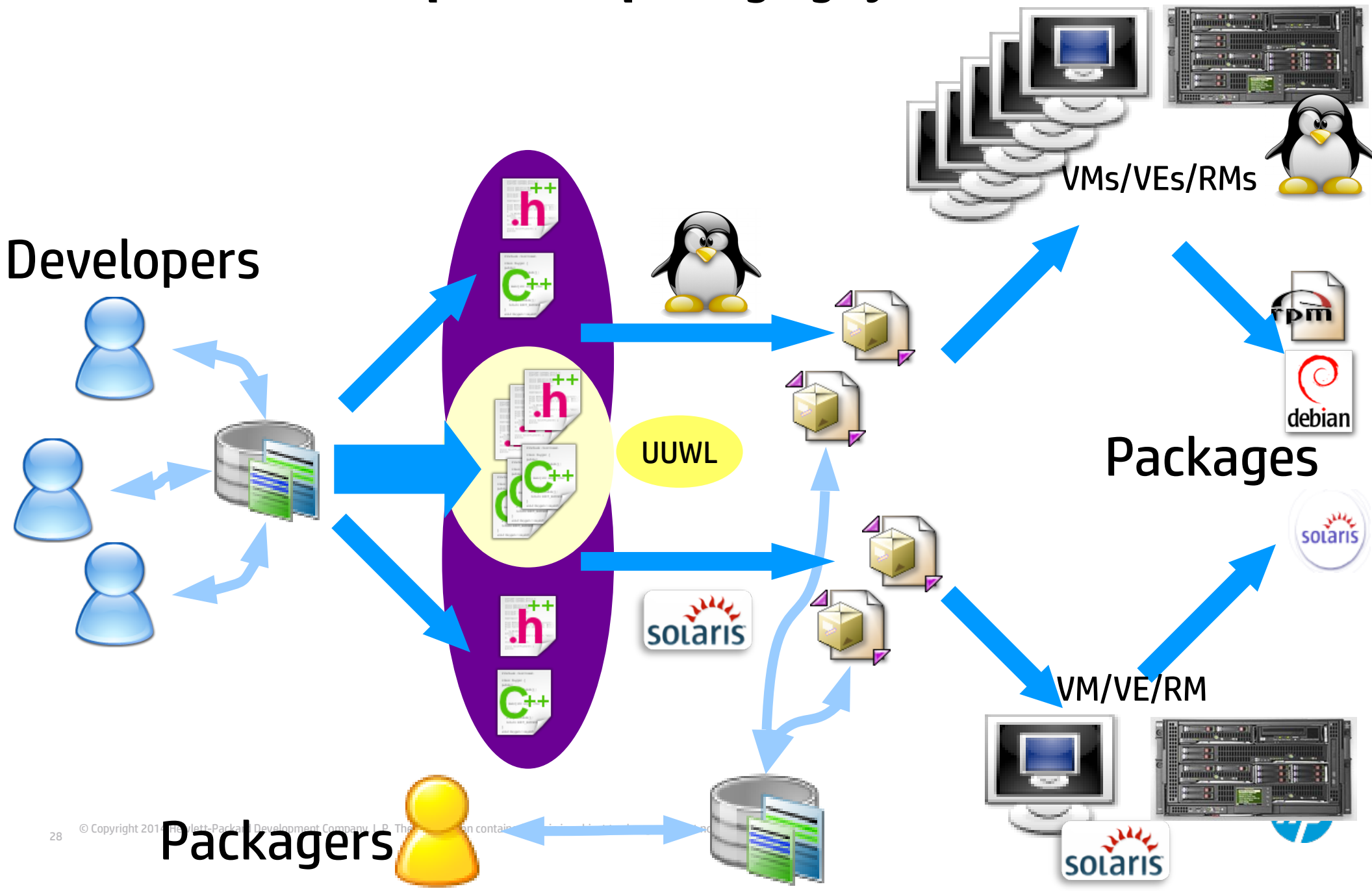
The Intel Xeon processor 5500 series are built with 45nm new Nehalem micro architecture with up to eight cores in a two-processor configuration. This new micro architecture delivers more performance in the same platforms and at the same power consumption, giving customers the flexibility to match performance, power and cost requirements with your unique requirements and delivering advantages beyond just pure performance.



without notice.



Multi platform packaging cycle



Q & A up to now ?

DEMONSTRATION

To be done

- **Support for libvirt, virsh, openstack API, LXC (?)**
- **Other CMS (Bazaar, ...) only when/if needed**
- **Other VMs (VMWare, Xen, ...) only when/if needed**
- **Multiple delivery means – usage of ansible ?**
- **Look at interactions with Buildbot**
- **Add non-interactive signature support for .deb**
- **Config-Model for configuration file management**
- **REST API and Web Interface would be nice. Any contributor ?**

Learn Project-Builder.org



HP EMEA

HP / Intel Solution-Center

HP Tech-Forum 2010

- **Start with the Lab (63 pages)**
- **Use man (pb, pb.conf + 8 ProjectBuilder::* man pages)**
- **Use the mailing-list pb-announce and pb-devel at <http://www.mondorescue.org/sympa>**
- **Use examples from <http://trac.project-builder.org/browser/projects/>**

Lab – Project-Builder.org

Lab Contents

This lab purpose is to install and use Project-Builder.org to produce packages for native and non-native Operating System, as well as experiencing the Continuous Packaging concept.

Lab Writer and Trainer

Bruno.Cornec@hp.com

Table of content

Objectives.	2
Reference documents.....	3
Environment setup.....	3
Project-builder.org installation.....	3
Local subversion setup.....	4
Using pb with an existing project (tar).....	5
Pb setup.....	5
pb repository setup.....	7
Tar file creation.....	9
Dealing with filters.....	12
Building packages.....	19
Building for another distribution.....	29
By using an existing rpm-based distribution VM.....	29
By using an existing deb-based distribution VM.....	43
By creating a new VM.....	57
By creating a new VE (optional).....	58
Using pb with an existing project (SVN).....	58
pb setup.....	58
pb repository setup.....	59
Tar file and package creation.....	60

Web Resources

Project-Builder.org Web sites:

- <http://www.project-builder.org>
- <http://trac.project-builder.org>

Projects using project-builder.org:

- <http://www.project-builder.org> (of course :-)
- <http://www.mondorescue.org>
- <http://www.linuxcoe.org>
- <http://www.fossology.org>

Related tools



[Project-Builder.org](http://www.project-builder.org) is mostly suited for upstream projects wanting to package their applications

Distributions provide each their build tools

- SuSE: Open Build Service (Multi distro, Web based, BaaS)
- Fedora (Koji)
- Mandriva/Mageia (Youri, mdvsys, mgarepo...)

Other complementary tools:

- Buildbot
- Parallel::ForkManager, DBI, DB::SQLite, [File::MimeInfo](#), Mail::Sendmail





Contact



Bruno_Cornec@hp.com

**(Open Source & Linux Strategist in the HP
EMEA Innovation Solution Center)**

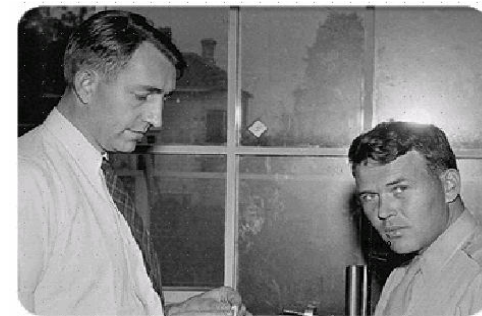
<http://www.hp.com/linux>

"Changes are never easy to make.
There is comfort and safety in
tradition, but change must come,
no matter how painful or expensive
it may be."

Bill Hewlett

Thanks

Open Source at HP
<http://opensource.hp.com>



Linus Torvalds, Richard Stallman, Eric Raymond, Nat Makarevitch, René Cougnenc, Eric Dumas, Rémy Card, Bdale Garbee, Bryan Gartner, Gallig Renaud, Solomon Hykes, Jérôme Petazzoni, Joachim Langenbach among others, for their work and devotion to the Open Source Software cause... and my family for his patience :-)

