SUSE® Linux Enterprise

CAVMEN January 2012

Don Vosburg
Sales Engineer
dvosburg@suse.com

4 COMPANIES THAT GET **IT**
1 **PURPOSE**

The Attachmate Group, Inc.

© 2011 The Attachmate Group, Inc. All rights reserved.

Our Commitments

Quality Products

Exceptional Service

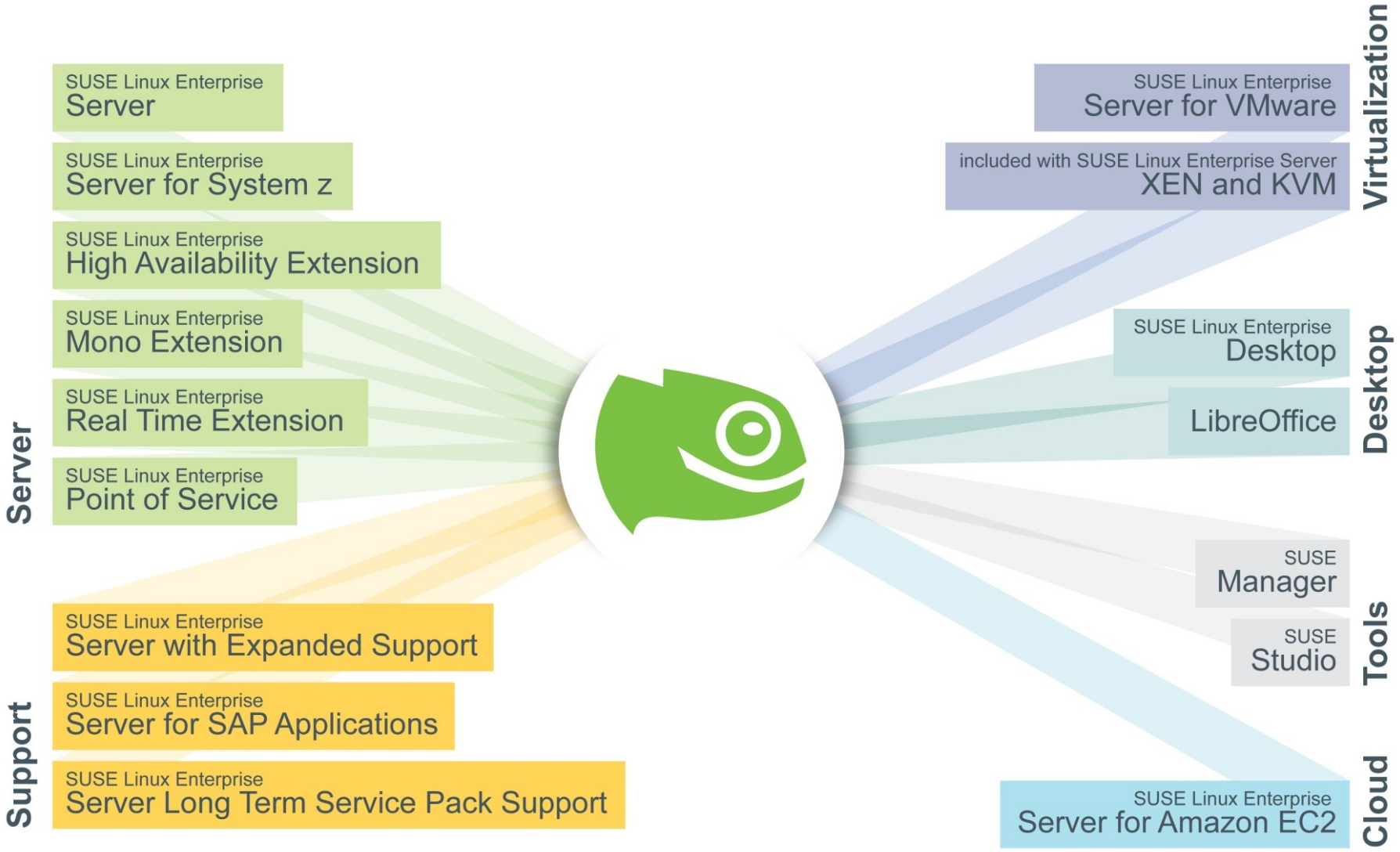
Easy to Do Business With

The Attachmate Group, Inc.

© 2011 The Attachmate Group, Inc. All rights reserved.

suse®

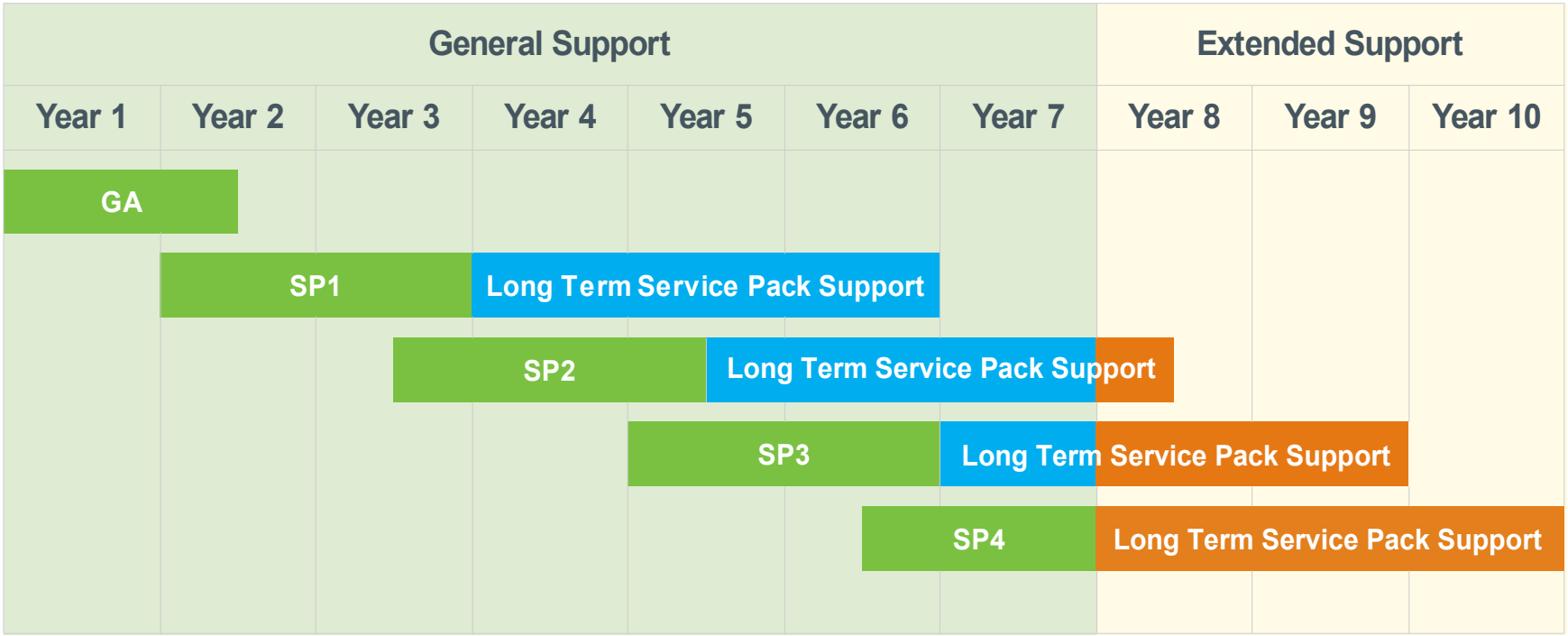
Comprehensive Portfolio



Roadmap

SUSE® Linux Enterprise

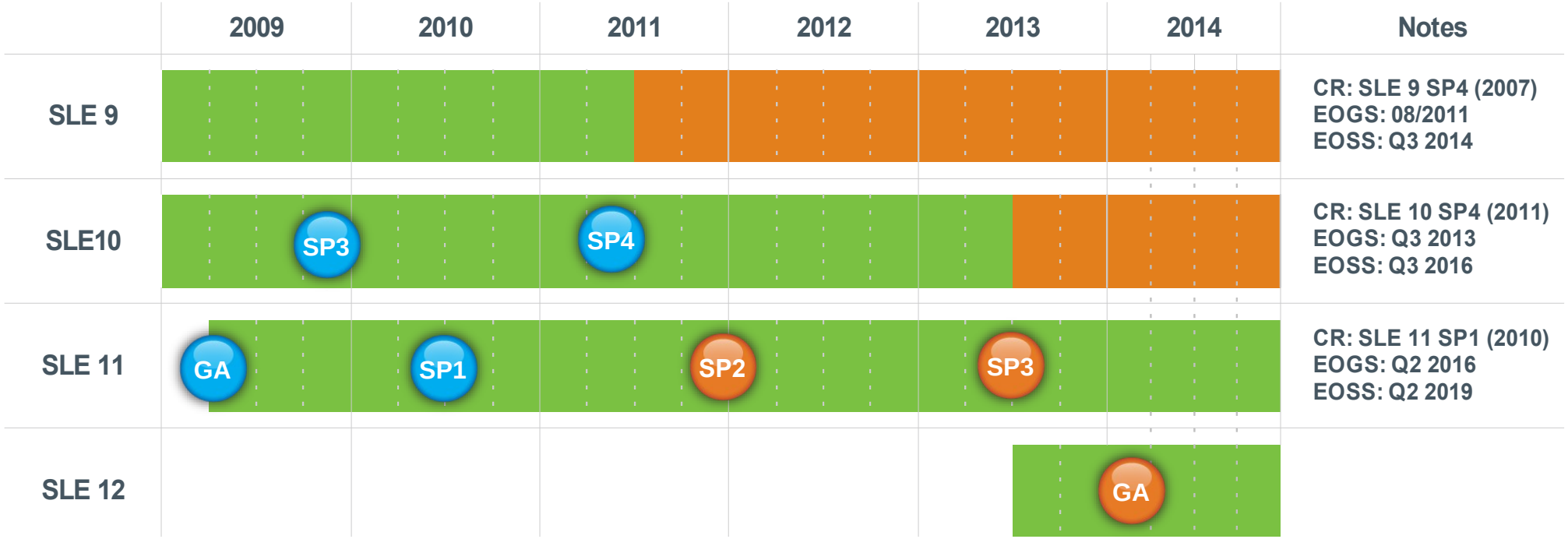
Standard Platform Lifecycle



- 10-year lifecycle (7 years general support, 3 years extended support)
- Major releases every ~4 years, service packs every ~18 months
- Six month upgrade window
- Long term service pack support – extend upgrade window or extend major release lifecycle

SUSE® Linux Enterprise

Current Platform Lifecycle



- SUSE announces service pack releases and development and product schedules to customers and partners
- Dependable release timing
- Predictability for planning rollouts and migrations

SUSE® Linux Enterprise 11 SP2

Key improvements

- New support model designed to ease updates
- Introduce full btrfs support, improving manageability
- Kernel moving to 3.X

SUSE® Linux Enterprise 11 SP2

- Hardware enablement and RAS
- Solaris compete
  - btrfs: file system with Copy on Write", checksums, snapshotting
  - LXC: container support based on control groups
  - LTTng (Linux Trace Toolkit) capabilities
- Snapshot / rollback for package and configuration updates
  - YaST2 + ZYPP + btrfs
- New maintenance model
  - Increased flexibility while retaining full control
- SUSE Linux Enterprise High Availability Extension: Geo-cluster, automated and pre-configuration

SUSE® Linux Enterprise 11 SP2

Reliability

- Improve support for hardware based RAS capabilities on all architectures, specifically
  - Intel Sandybridge
  - IBM System z
- Support for the btrfs filesystem
  - Reduce cost of storage management by providing an integration of logical volume management and filesystem
  - Checksums on data and metadata ensure data integrity
- LTTng (Linux Trace Toolkit Next Generation)
  - SUSE Labs contributing to LTTng technology
  - Availability of base functionality for SLE 11 SP2 (as part of an Extension)

Scalability

- Increasing number of cores per system → power management and scheduling key to control costs
- Local storage
  - Maintain existing capabilities (e.g. XFS)
  - Introduce full btrfs support, improving manageability
  - Maximum flexibility for customers
- Expand network filesystem capabilities (NFSv4.x/pNFS)
  - Improve performance, reliability and security
  - pNFS client support planned for 2012,
    server support for later version of SUSE Linux Enterprise High Availability Extension 11

SUSE® Linux Enterprise 11 SP2

Interoperability

- Remote Desktop Protocol (RDP, Windows environments)
  - Support xrdp server on AMD64/Intel64
  - Introduce FreeRDP client: better Performance and Interoperability
- Samba 3.6
  - SMB2 protocol support
- Improve CIFS kernel module
- IPv6
  - Improve DHCP server and client (ISC dhcp)
  - IPv6 support in NFS
  - IPv6 capable Squid proxy (most other network daemons are IPv6 ready since SLE 10 times)
  - Ensure IPv6 capabilities with UEFI network boot

Today

- Full Dynamic Resource Handling
  - Two level of virtualizations to choose (LPAR and z/VM)
    - Choose the level of isolation mandated by compliance
    - Flexible resource allocation and reallocation
  - CPU, memory, I/O hotplug
    - Provide the resource where they are needed in LPAR and zVM guest
- Abundant memory and IO bandwidth and transaction capability
  - Hipersocket support connects Linux and z/OS applications and data
- RAS
  - SUSE Linux Enterprise High Availability Extension included
  - IO performance statistics
  - Dump generation and inspection facilities
  - System z specific kernel messages with documentation

Tomorrow

- z196 + zBX = zEnterprise exploitation
  - CPU topology and instruction set exploitation of z196
    - Improve single workload performance -> improve workload consolidation
  - New CHPID support connecting both environments
- Choose the right workload environment
  - SLES available for both hardware architectures
  - ISVs application support might mandate the platform
- Improved tools and infrastructure support
  - hypertop – check other LPAR or z/VM guests resource needs
  - Dynamic PAV toleration and improved FICON support

btrfs

Why another filesystem?

- Solve Storage Challenges
  - Scalability
  - Data Integrity
  - Dynamic Resources (expand and shrink)
  - Storage Management
  - Server, Cloud – Desktop, Mobile
- Compete and exceed other Operating Systems

btrfs (better fs) – Features

- Integrated Volume Management
- Support for Copy on Write
- Powerful Snapshot capabilities
- Scalability (16 EiB) including effective shrink
- Supports offline in-place migration from ext2+
- Other Capabilities:
  - Compression
  - Data integrity (checksums)
  - SSD optimization

Why btrfs?

btrfs (better fs) – Background

- Summarizes ideas from
  - ext2,3,4
  - reiserfs
  - xfs (SGI)
  - aufs (HP)
- Implements the same concepts as ZFS does
- Built for scalability and extensibility
- Integration with Volume Management
- Cloud readiness
  - builds the foundation for the Ceph distributed filesystem and its RADOS object store layer
- Active Community

Why btrfs?

btrfs (better fs) – Status

- SUSE® Linux Enterprise 11 SP1: Technology Preview
- SUSE® Linux Enterprise 11 SP2: Supported
- SUSE® Linux Enterprise 12 (plan): Default Filesystem

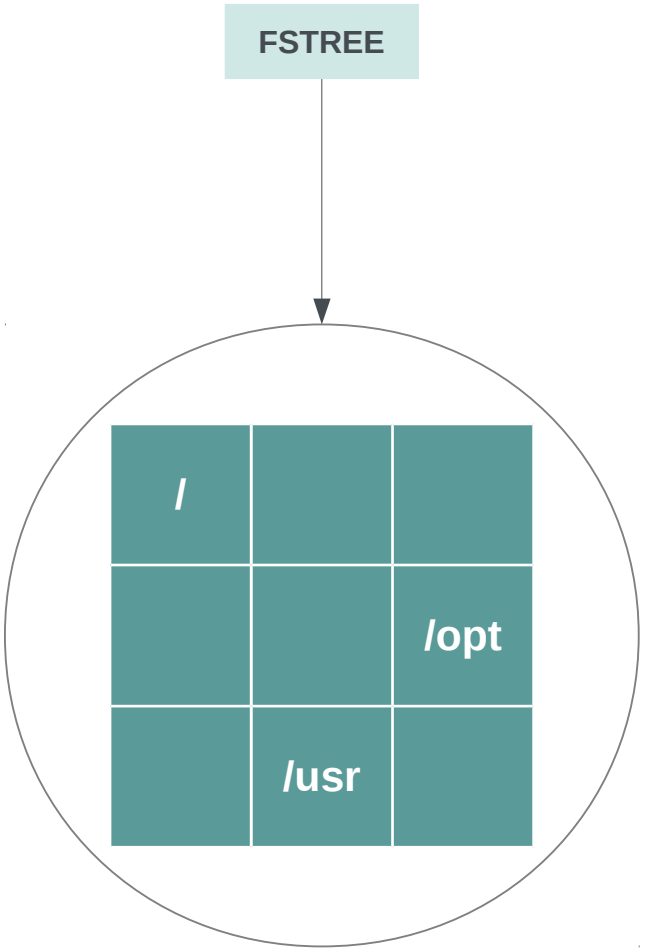
Subvolume (1)

- A complete filesystem tree
- Usually appears as a sub-directory in the “parent” fs
- Can be mounted separately
- not “just a subdirectory”
- Simliar to
  - two “foreign” filesystems, which are
  - using the same pool of data blocks (and other infrastructure)
- Benefits
  - different parts (subvolumes) of a filesystem can have different attributes, such as quotas or snapshotting rules
  - Copy on Write is possible across volumes
- Basic commandline management
  - “btrfs subvolume …”

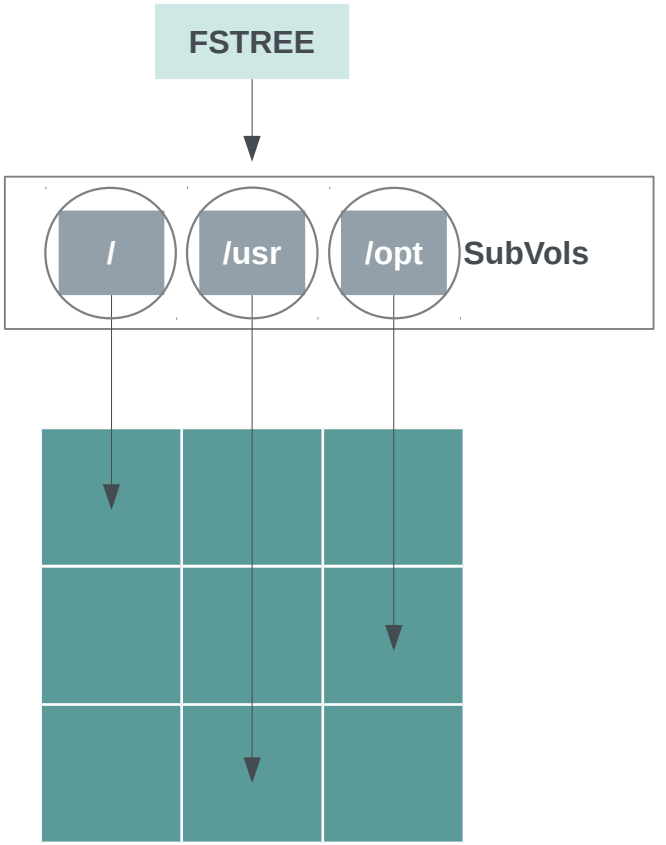
Technology Overview

Subvolume (2)

Normal Filesystem



With Subvolumes



Technology Overview

Snapshots

- Copy on Write on a
  - full subvolume tree
  - instead of a single file only
- Every snapshot is again a subvolume of its own
- Snapshots (as subvolumes) can be mounted and accessed as every other subvolume
- Snapshots can be created read-only
- Basic commandline management
  - “btrfs subvolume snapshot …”

Rollback – per Subvolume

How it works

- Instead of the original subvolume, the snapshot is mounted with the options “subvol=<name>”
  - Remember: snapshots are subvolumes
- Talking about the “/” filesystem, the “subvol” can also be hardcoded using “btrfs subvolume set-default ...”

Benefits

- “atomic” operation
- Very fast

Disadvantages

- Additional complexity
  - May require explicit mounting of subvolumes
- No “rollback” per single file

Rollback – File based → “undochange”

How it works

- The system uses the same instance of a subvolume: “working instance”
- single files are copied from the snapshot to the “working instance” – using CoW

Benefits

- Subvolumes are treated as read-only
- Subvolumes can be used for Backup
- Supports Pick and Choose

Disadvantages

- rollback not “atomic”
  - may need some more computing time

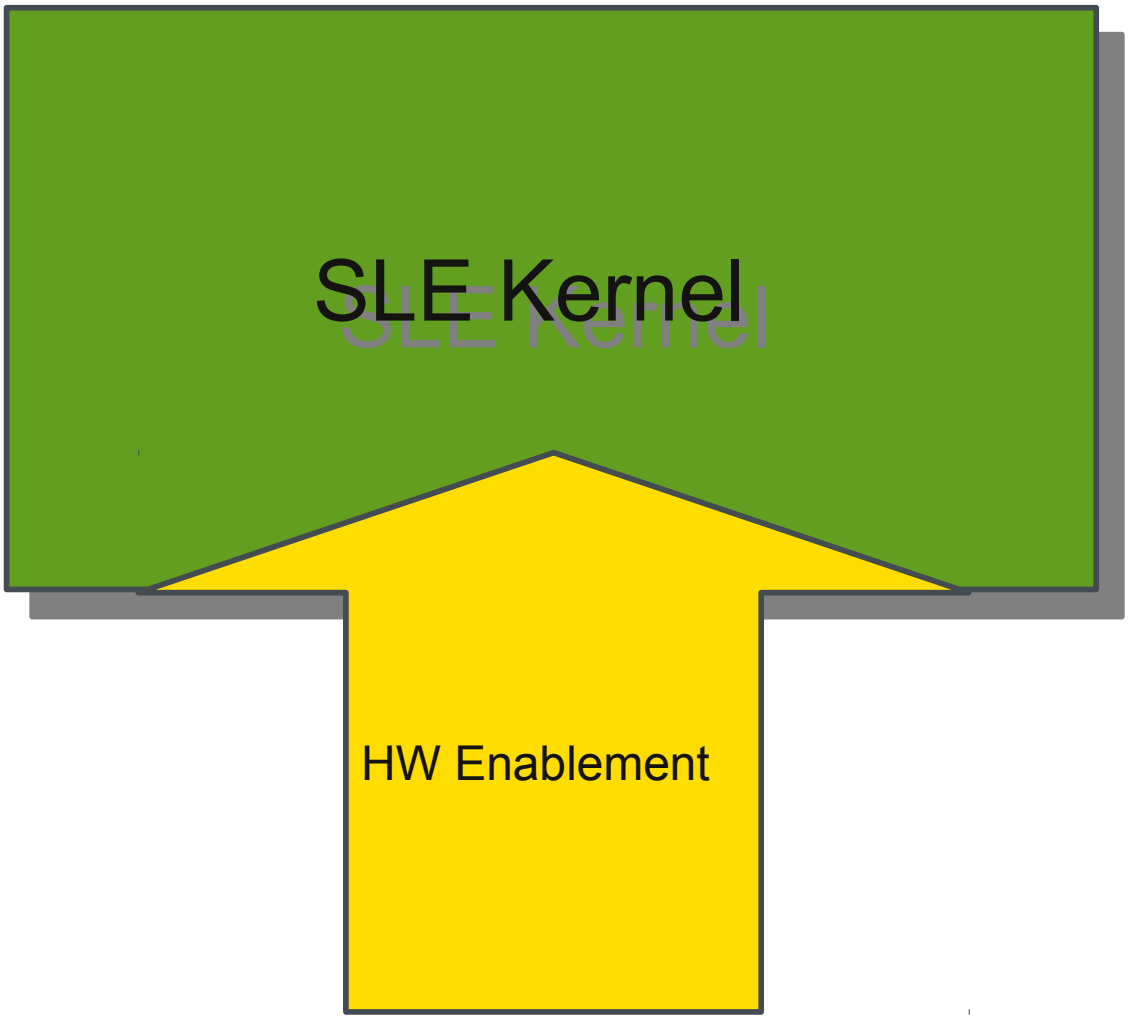
Requirements

- Single file rollback (“undochange”)
- Keep option to implement subvolume based rollback in the future
- User interface: cmdline and YaST2 integration
- Snapshots for YaST2 and zypper activities
- Automated snapshots (time based)
- Automated snapshot cleanup (time/number based)
- Allow to work with several kernels
  - implemented independently of btrfs (in ZYpp)

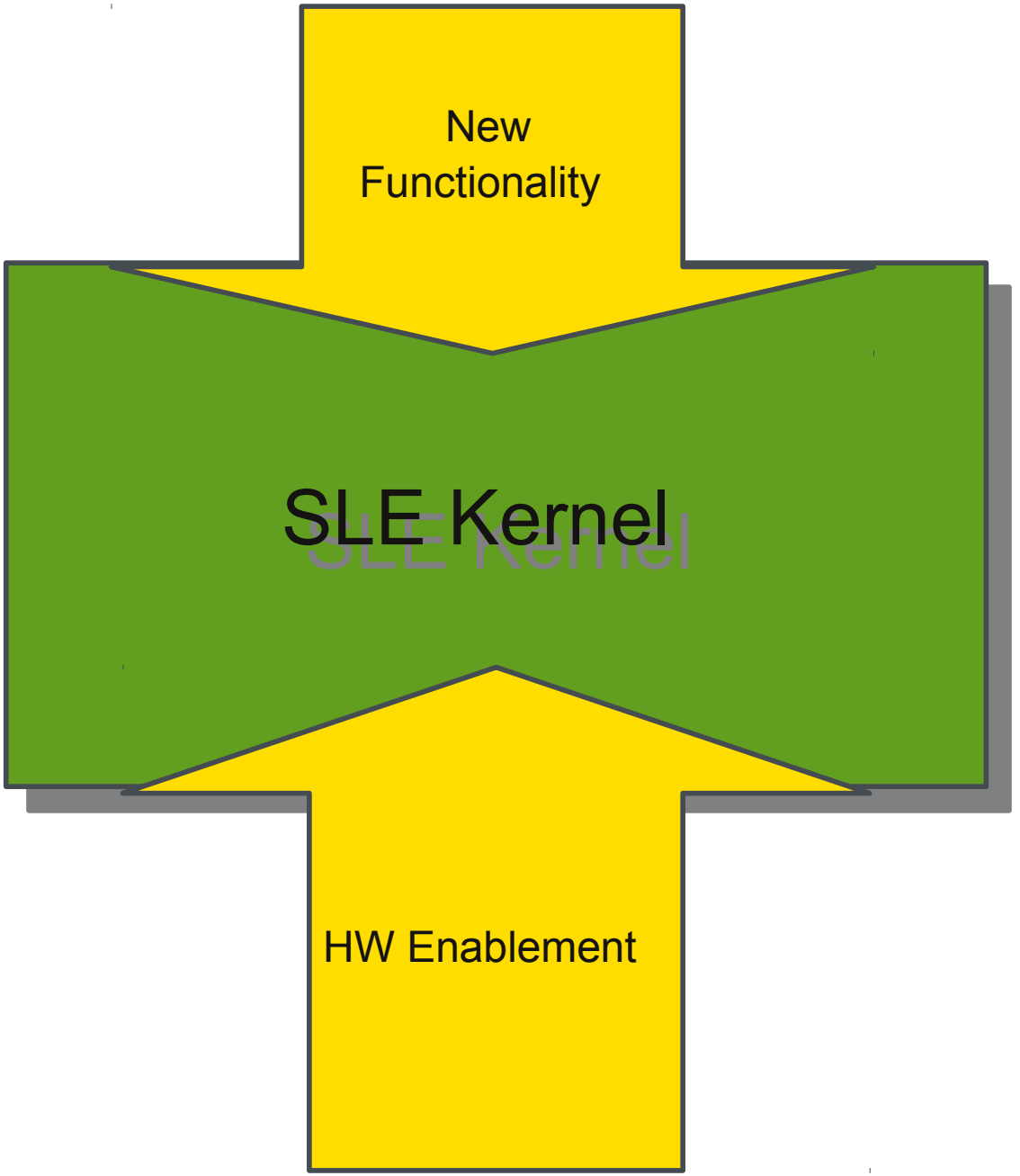
Kernel changes – the why and what

Four Forces of a Service Pack Kernel

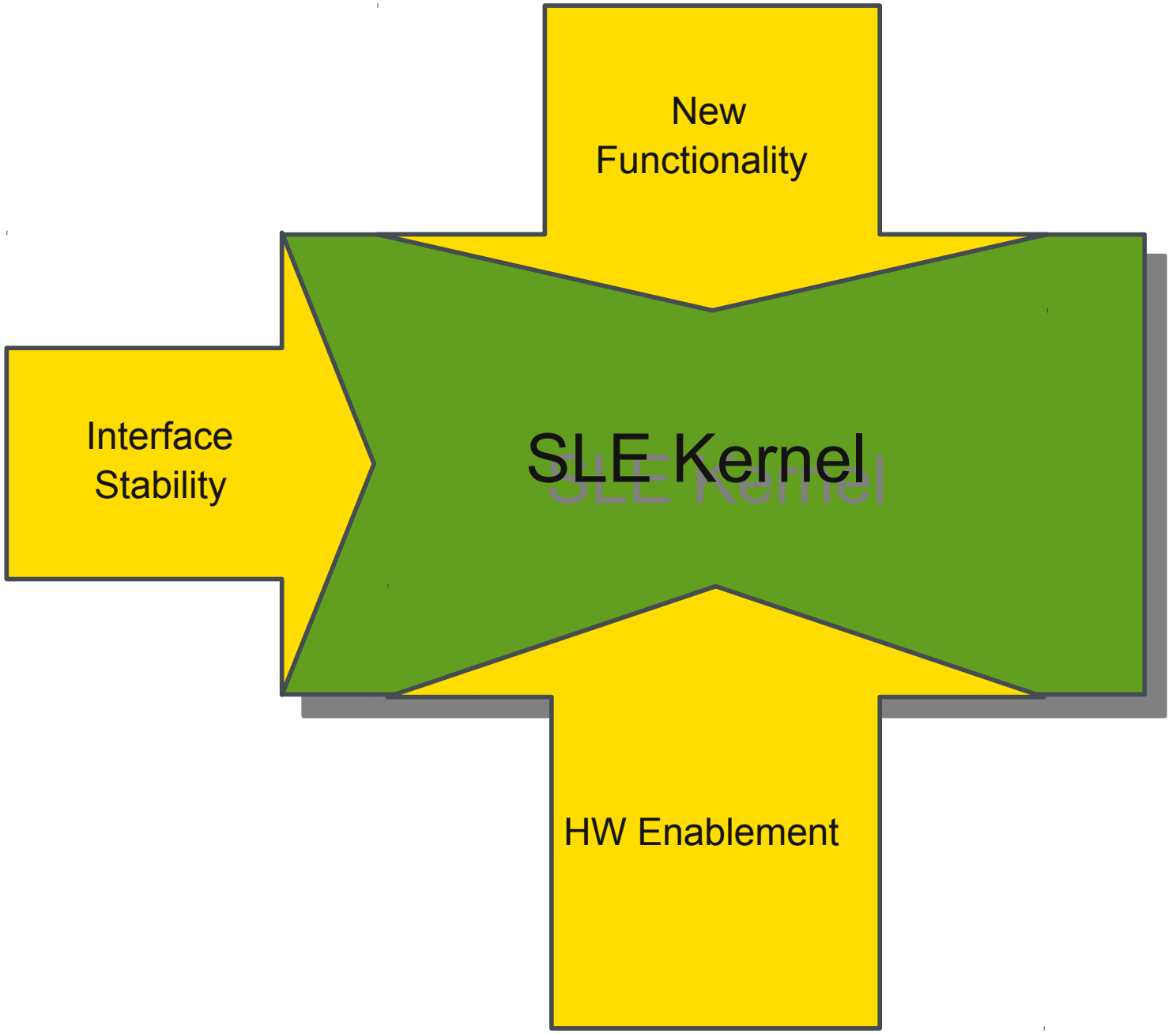
Four Forces of a Service Pack Kernel



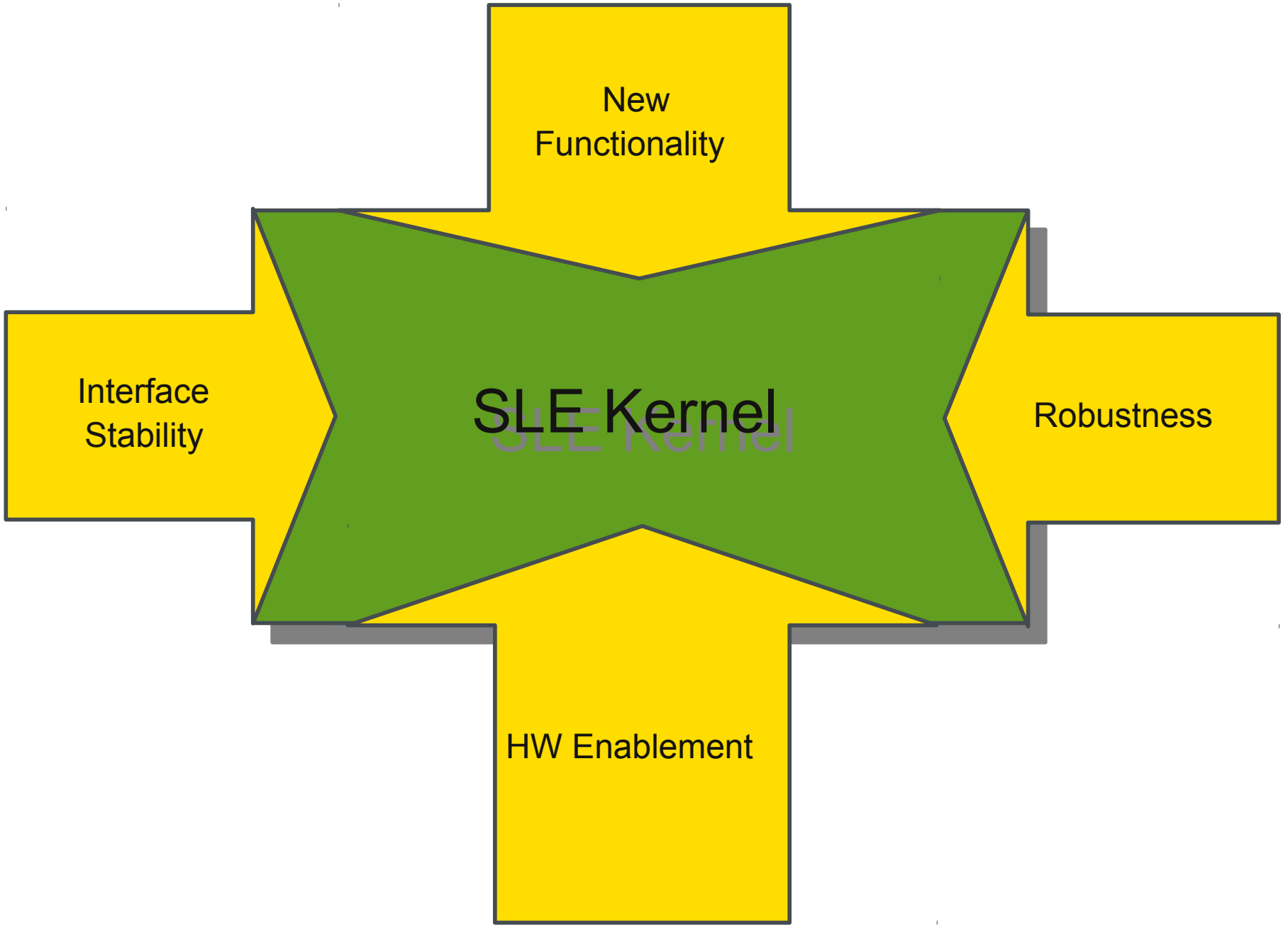
Four Forces of a Service Pack Kernel



Four Forces of a Service Pack Kernel



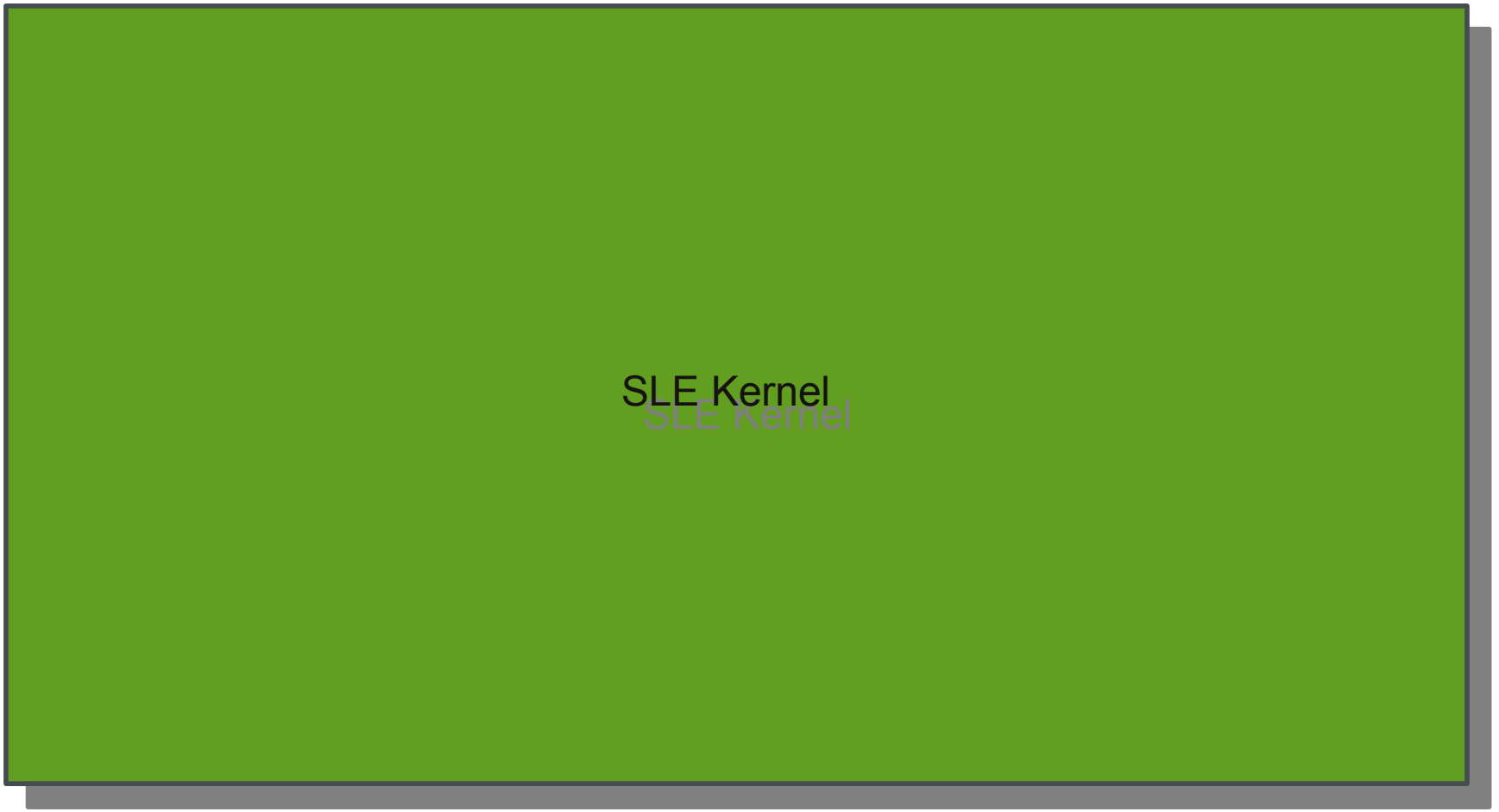
Four Forces of a Service Pack Kernel



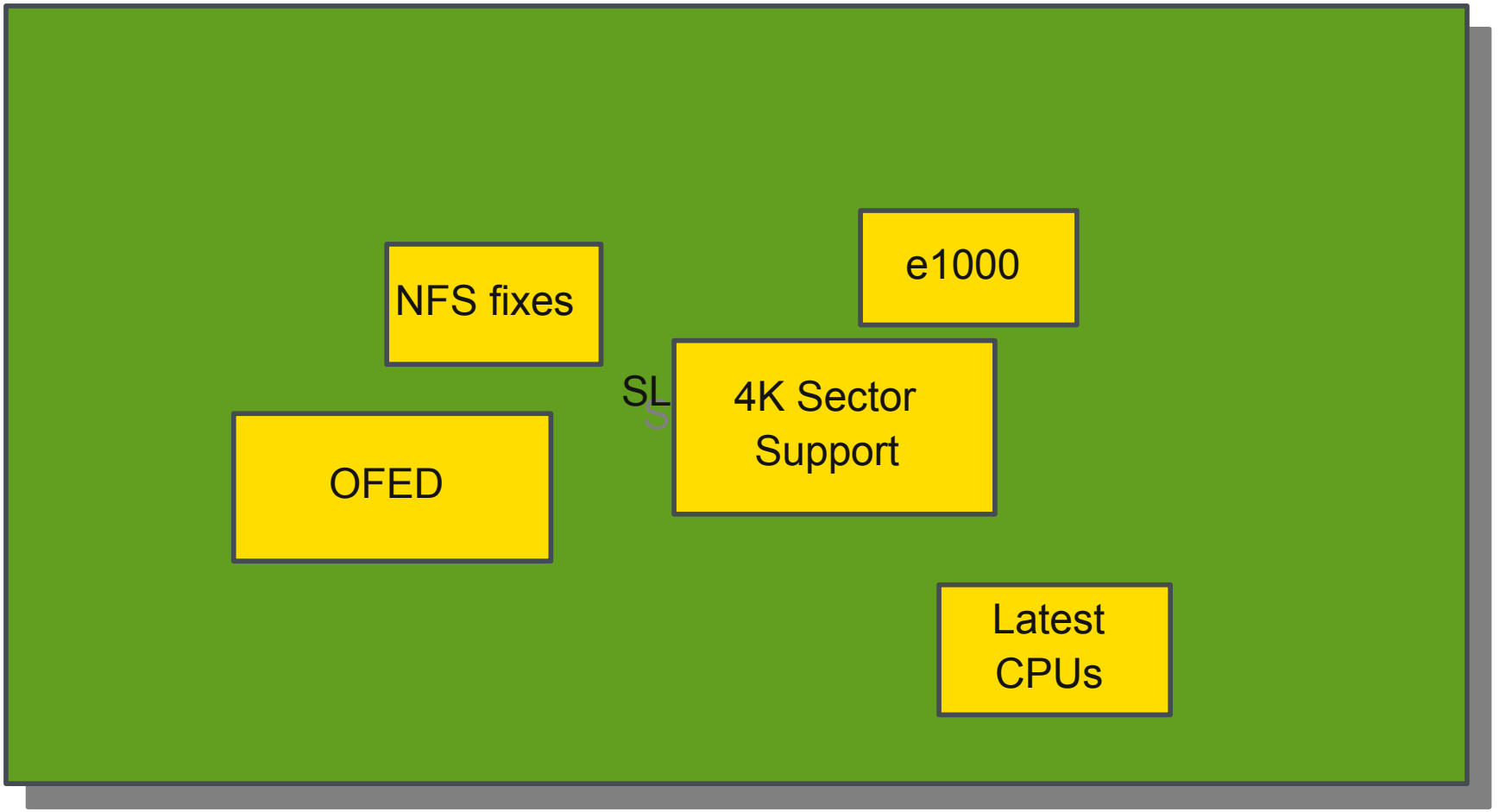
What Drives Kernel Changes in a Service Pack

- We make changes to the kernel based on
  - customer requests
  - requirements of upcoming hardware
    - new CPUs, chipsets, peripherals, and entire subsystems like USB 3.0
  - innovative changes in upstream kernel
    - btrfs, fcoe, ...
  - anticipated market needs
- And the changes are both massive and often intrusive

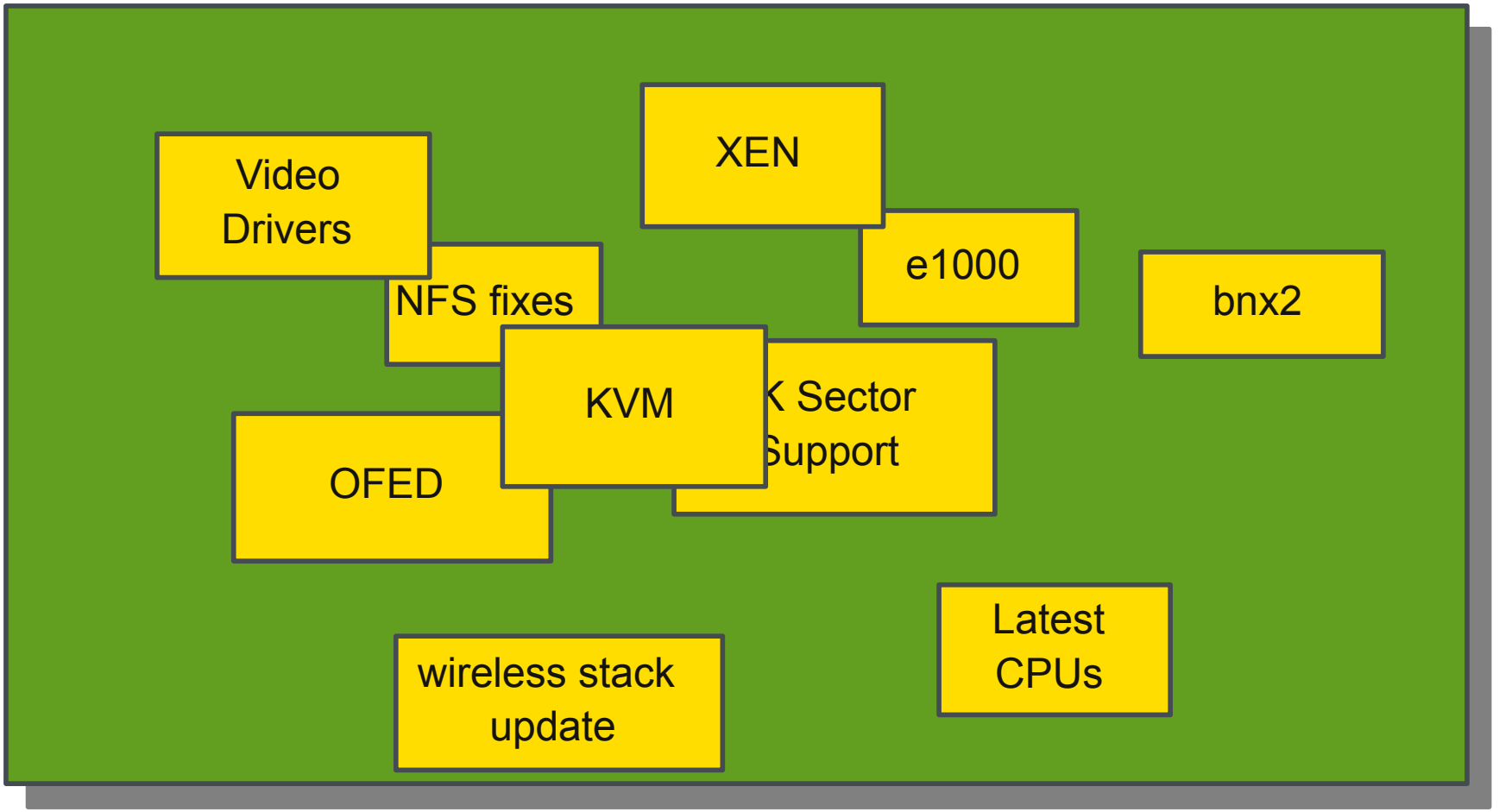
Kernel Components Changed in a Service Pack



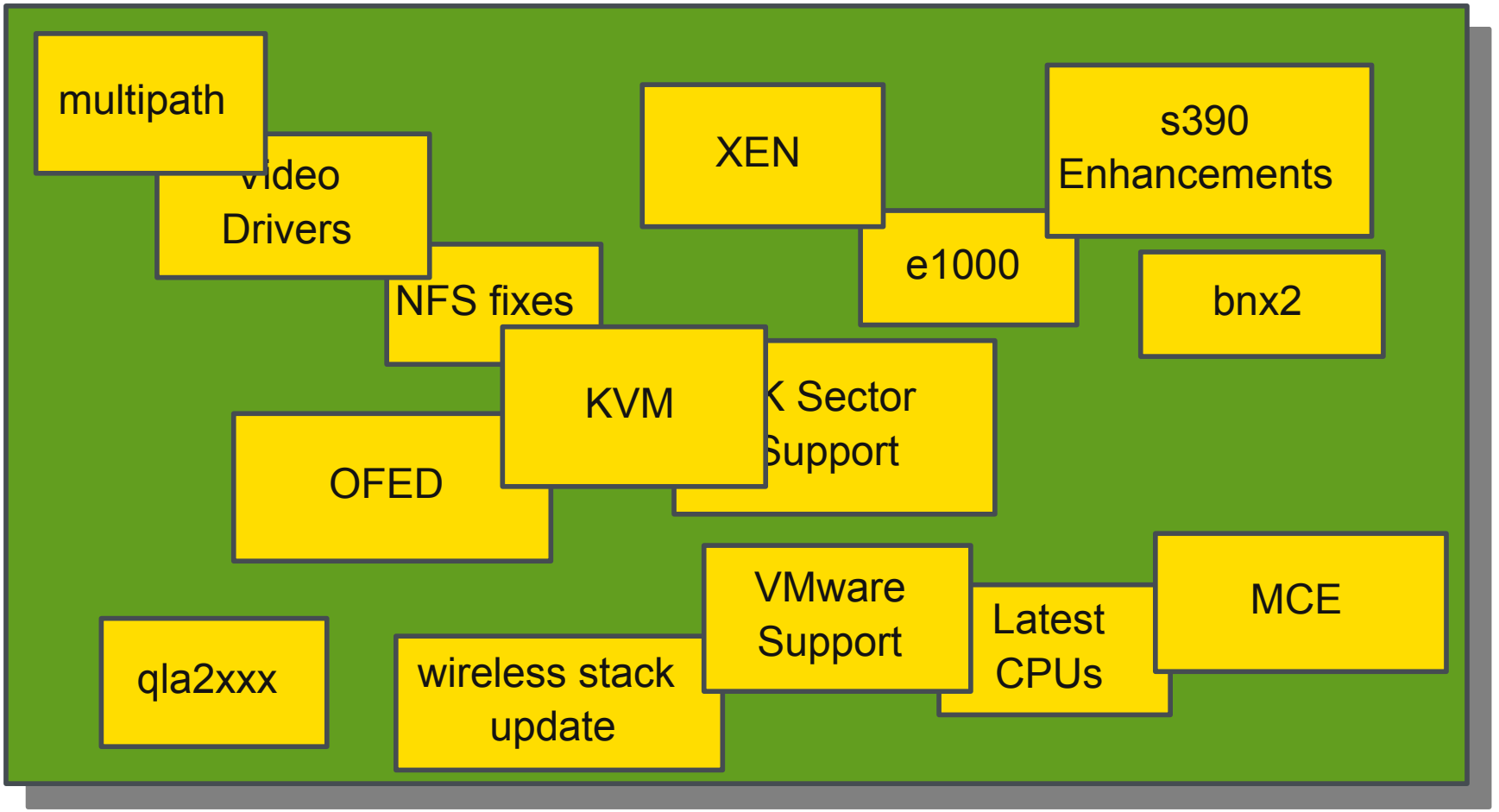
Kernel Components Changed in a Service Pack



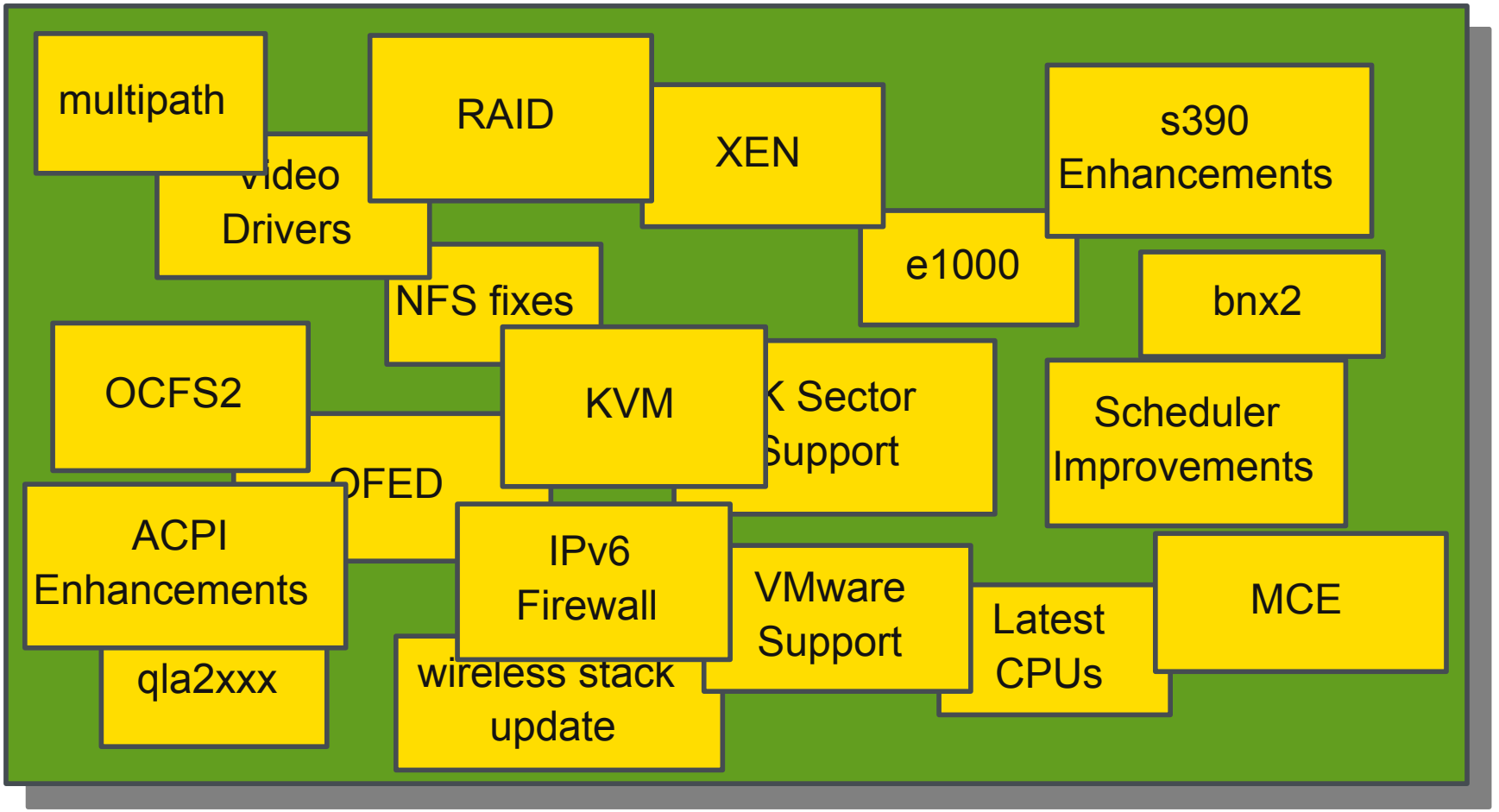
Kernel Components Changed in a Service Pack



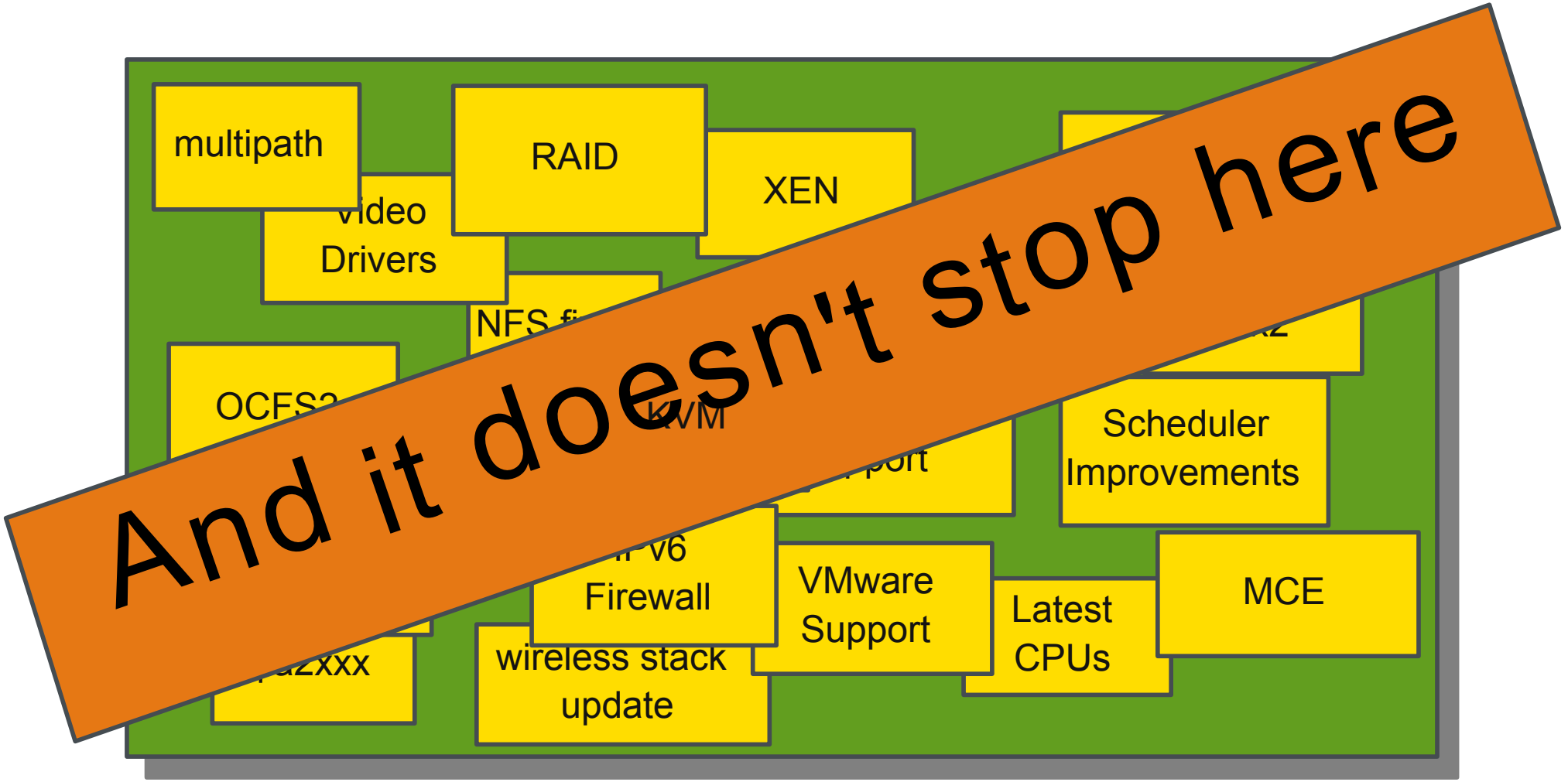
Kernel Components Changed in a Service Pack



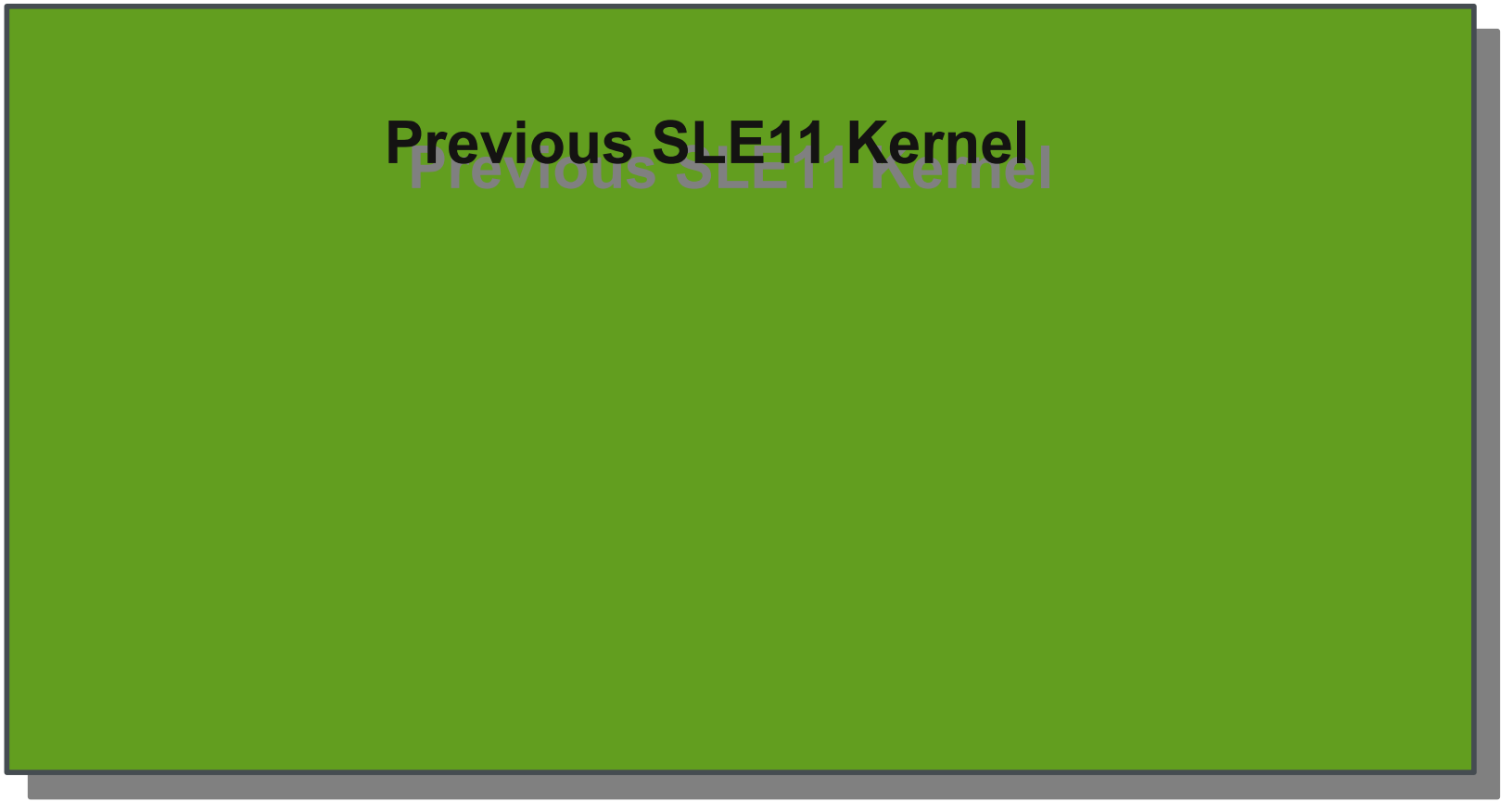
Kernel Components Changed in a Service Pack



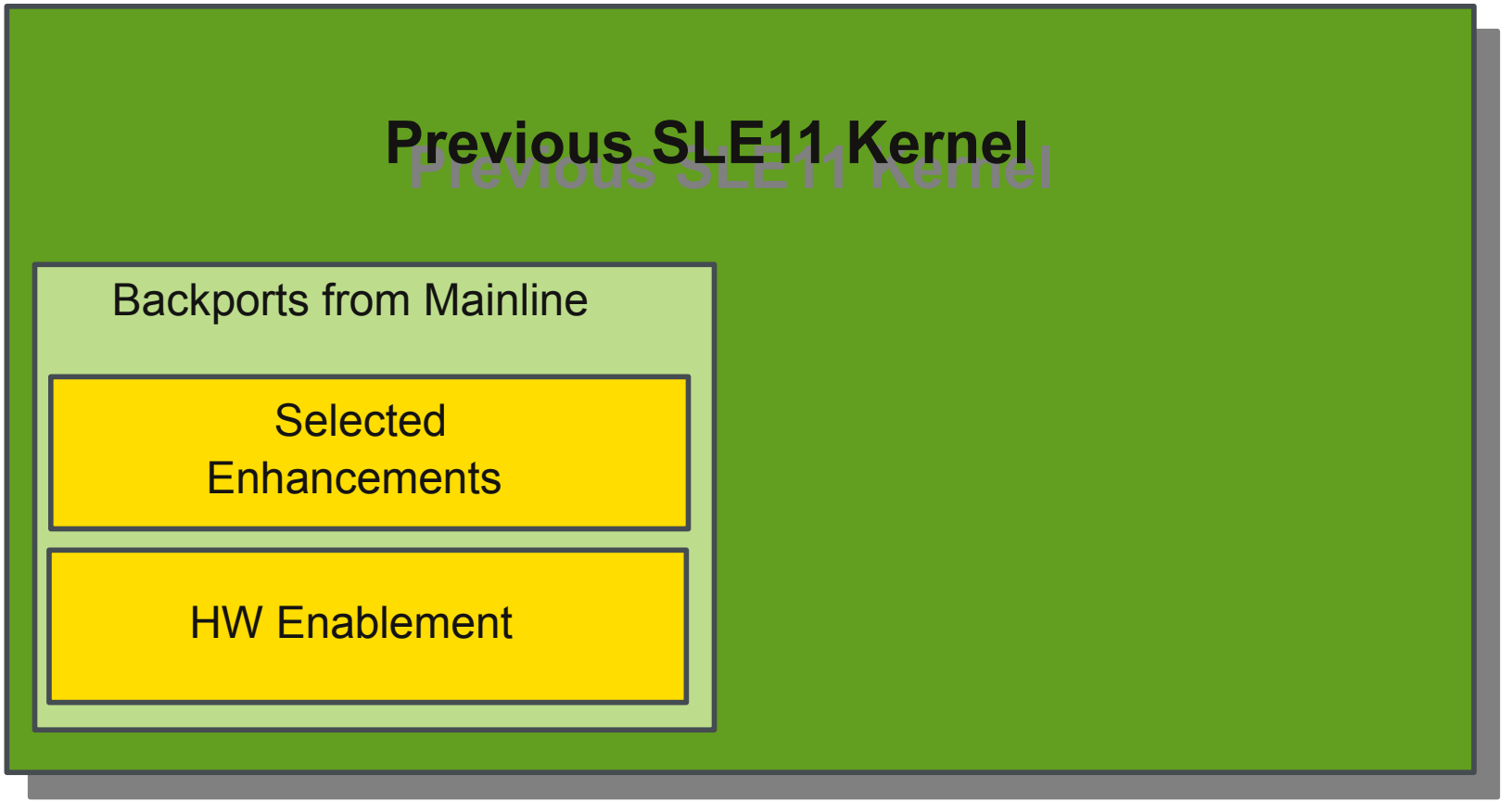
Kernel Components Changed in a Service Pack



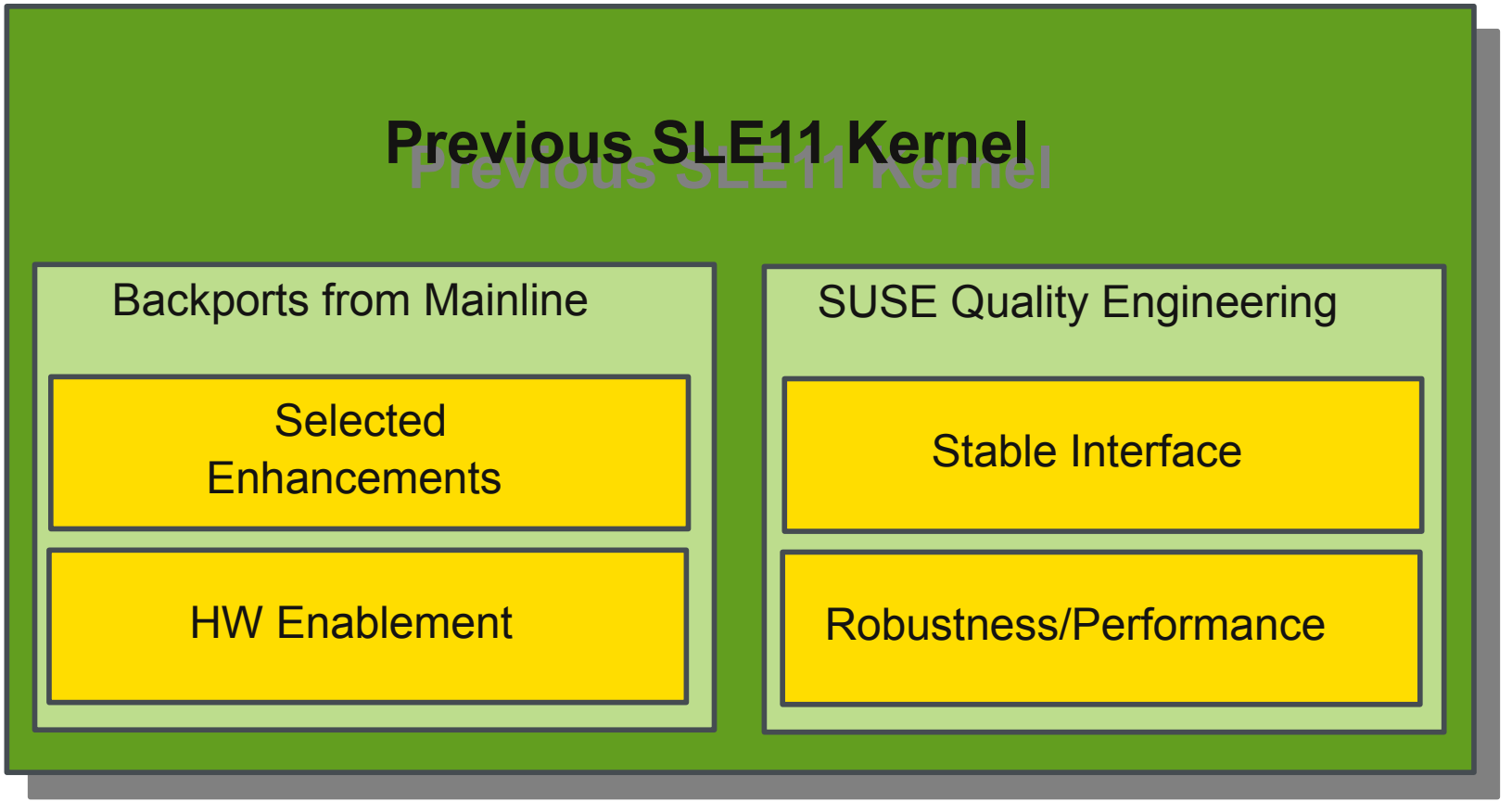
How do we make this all work?



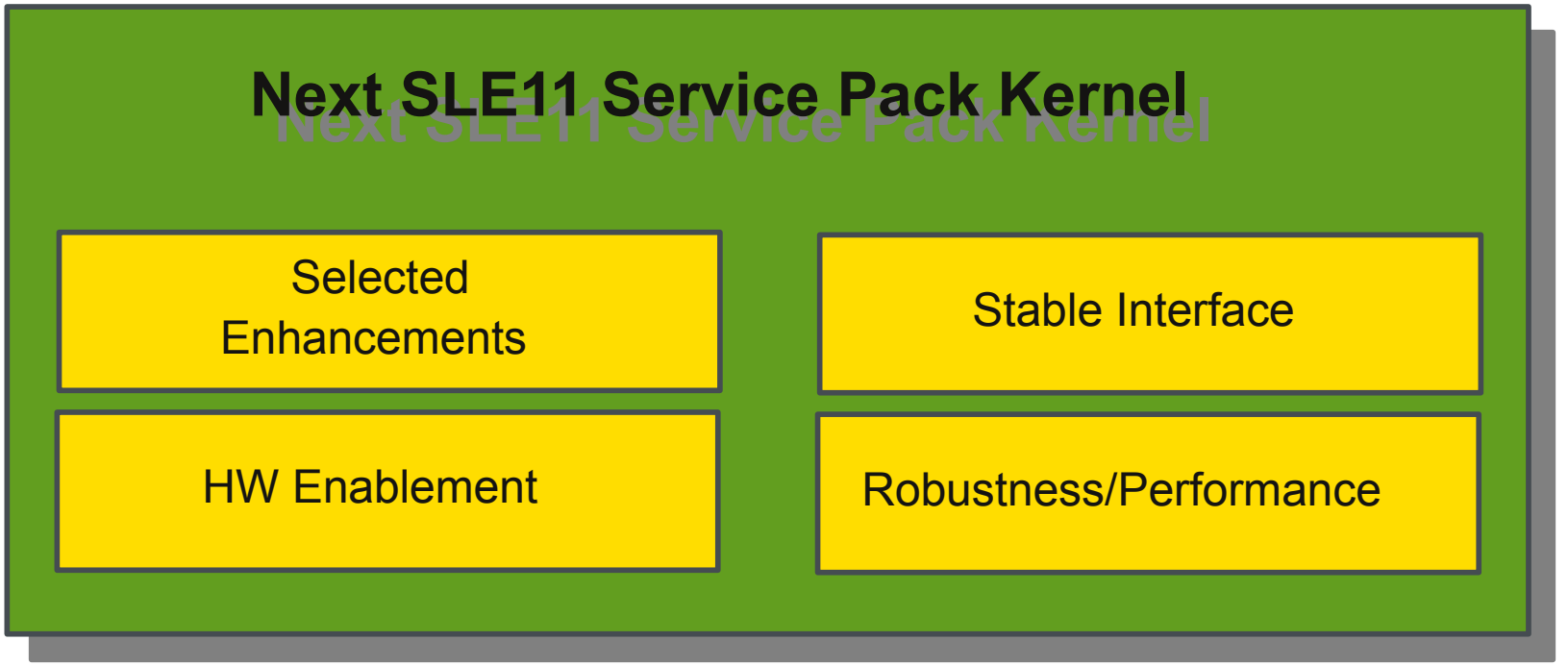
How do we make this all work?



How do we make this all work?



How do we make this all work?



SUSE Linux Enterprise 11 SP2

- The pace of innovation has picked up further
  - Using 2.6.32, the SP2 kernel carried 12,500 patches, and >100,000 lines of code were changed
  - This is more than twice the number of code lines changed than usual
- The changes were almost all across the map
  - We replaced scheduler, block I/O layer, virtual memory management, the entire USB stack, plus a few more
  - We introduce btrfs as supported file system

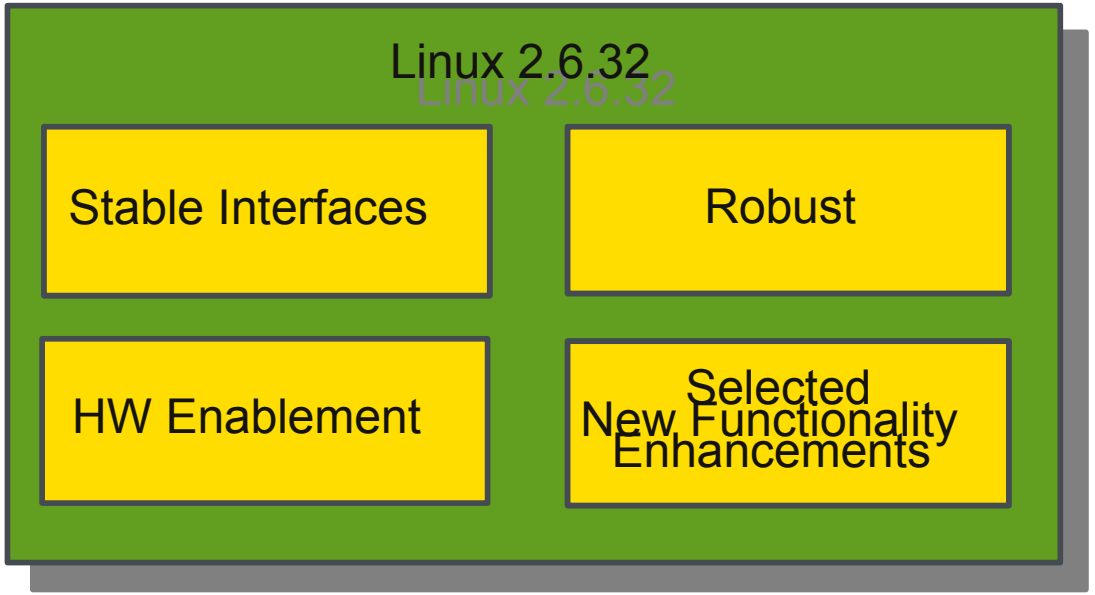
But is this the Best we can give you?

- We did not incorporate some major features
  - For instance pNFS would have meant another 700 patches
  - A number of virtualization improvements didn't make it
- We would have to be even more selective with future service packs
- We certainly can do better!

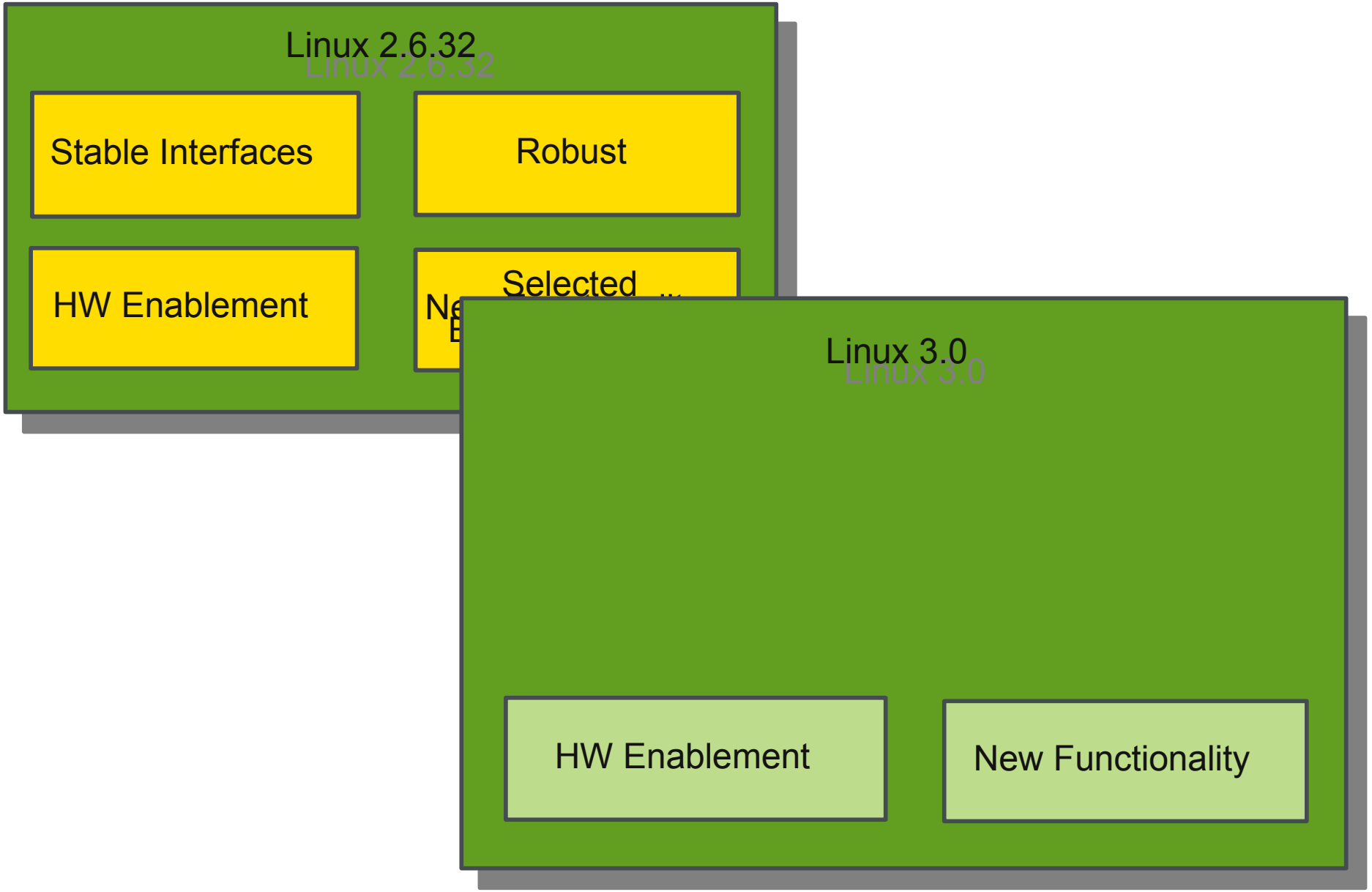
For SLE 11 SP2, we're proud to present to you

Linux 3.0

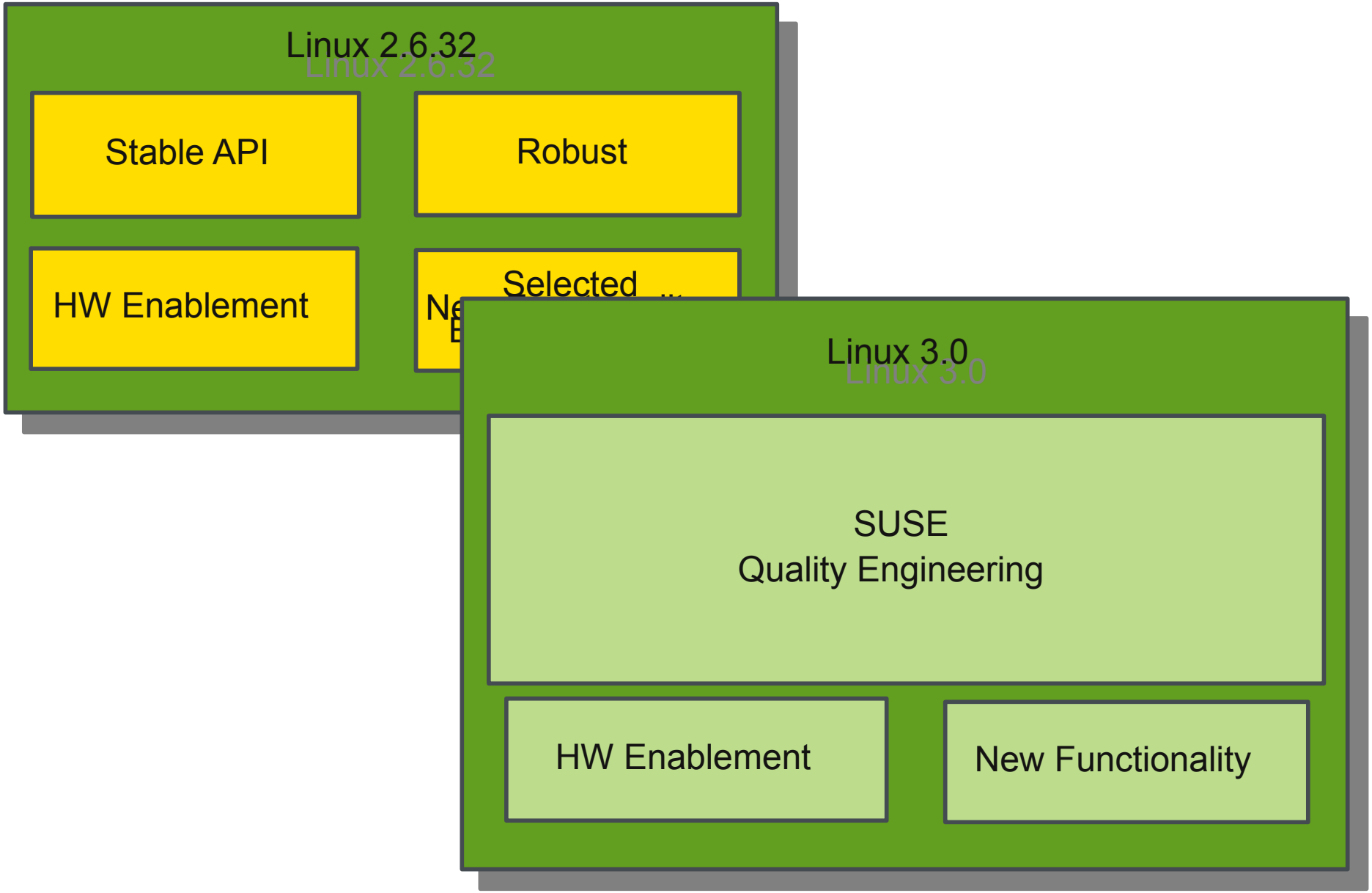
What does this change mean?



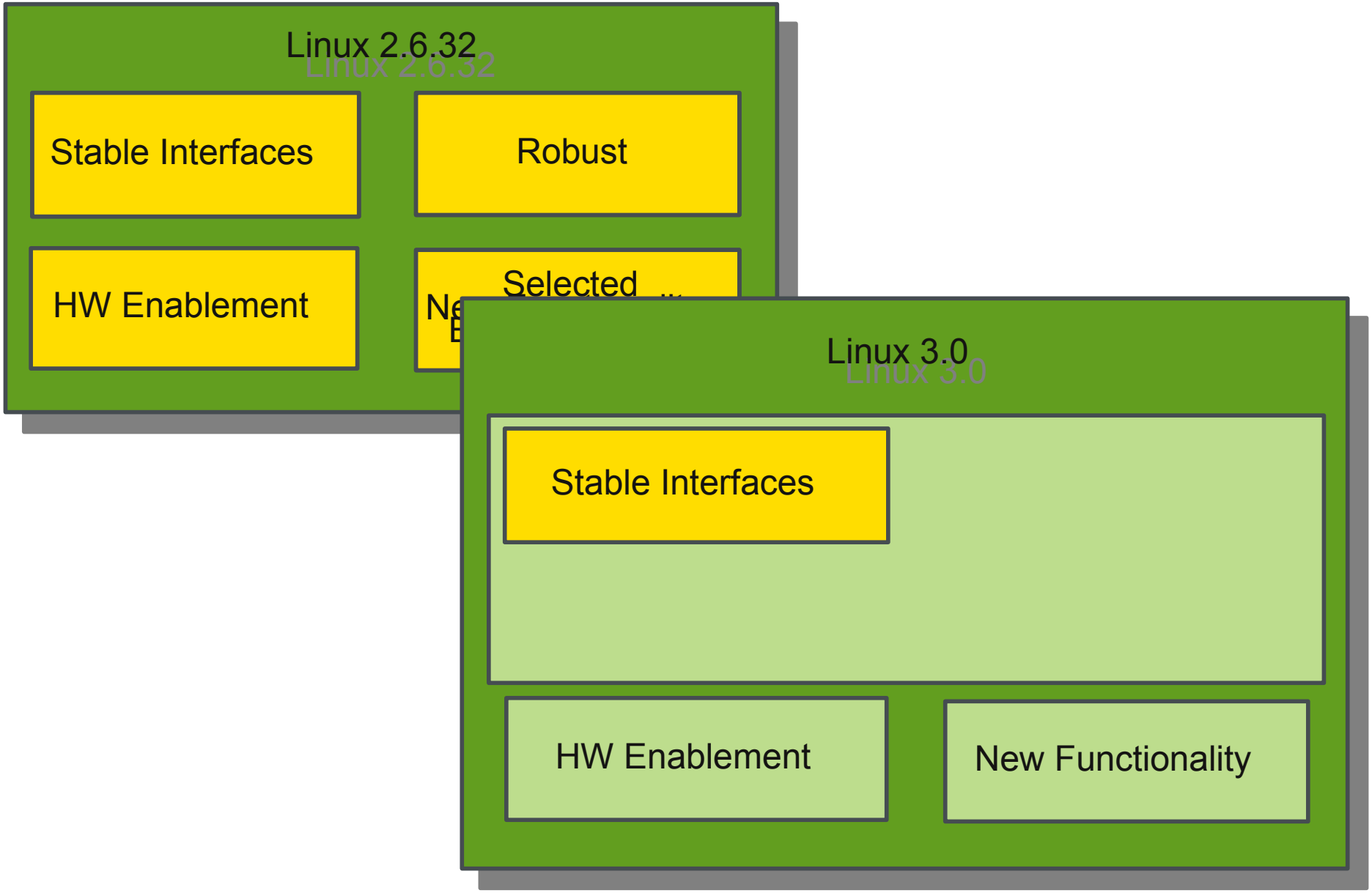
What does this change mean?



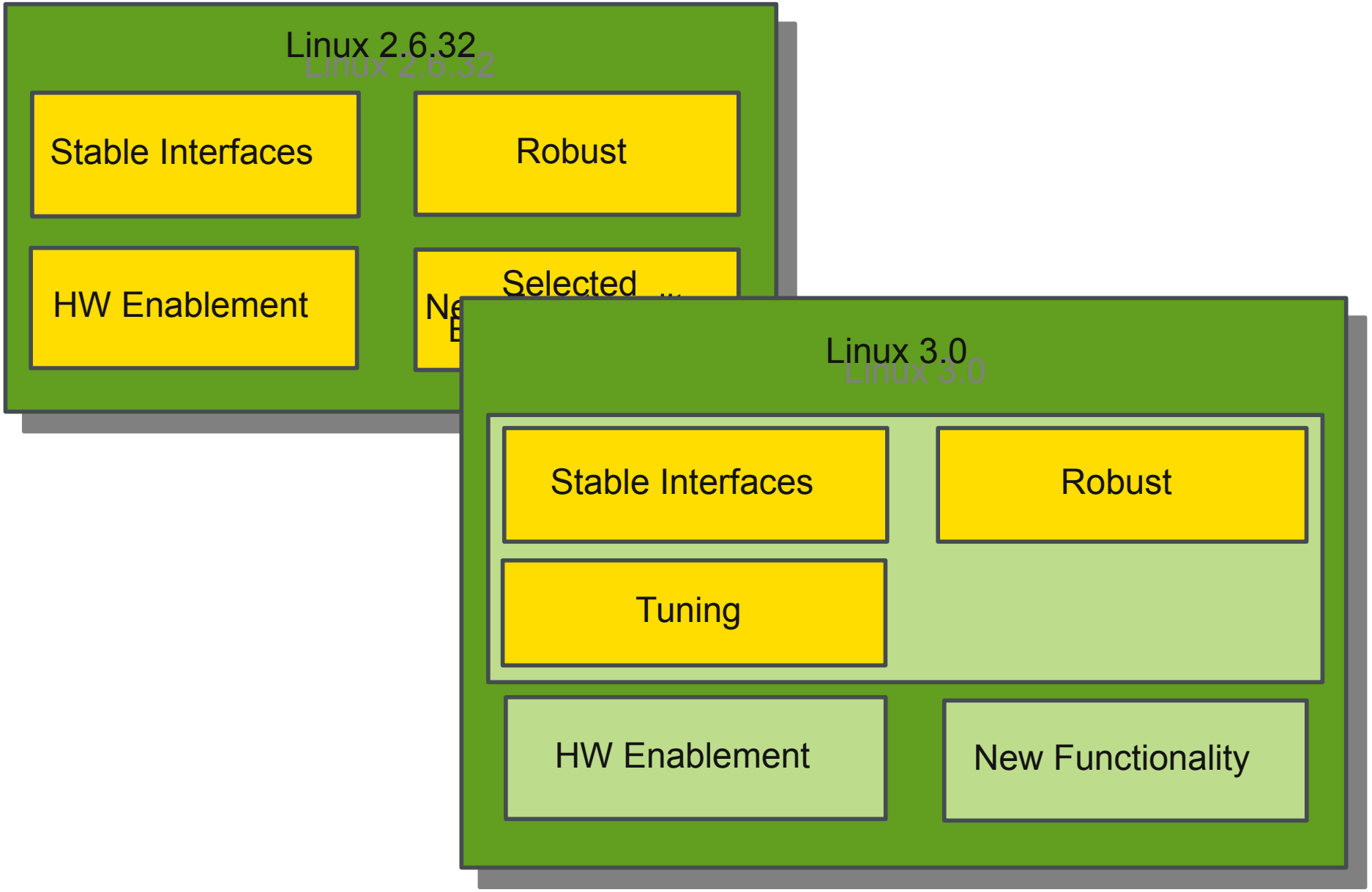
What does this change mean?



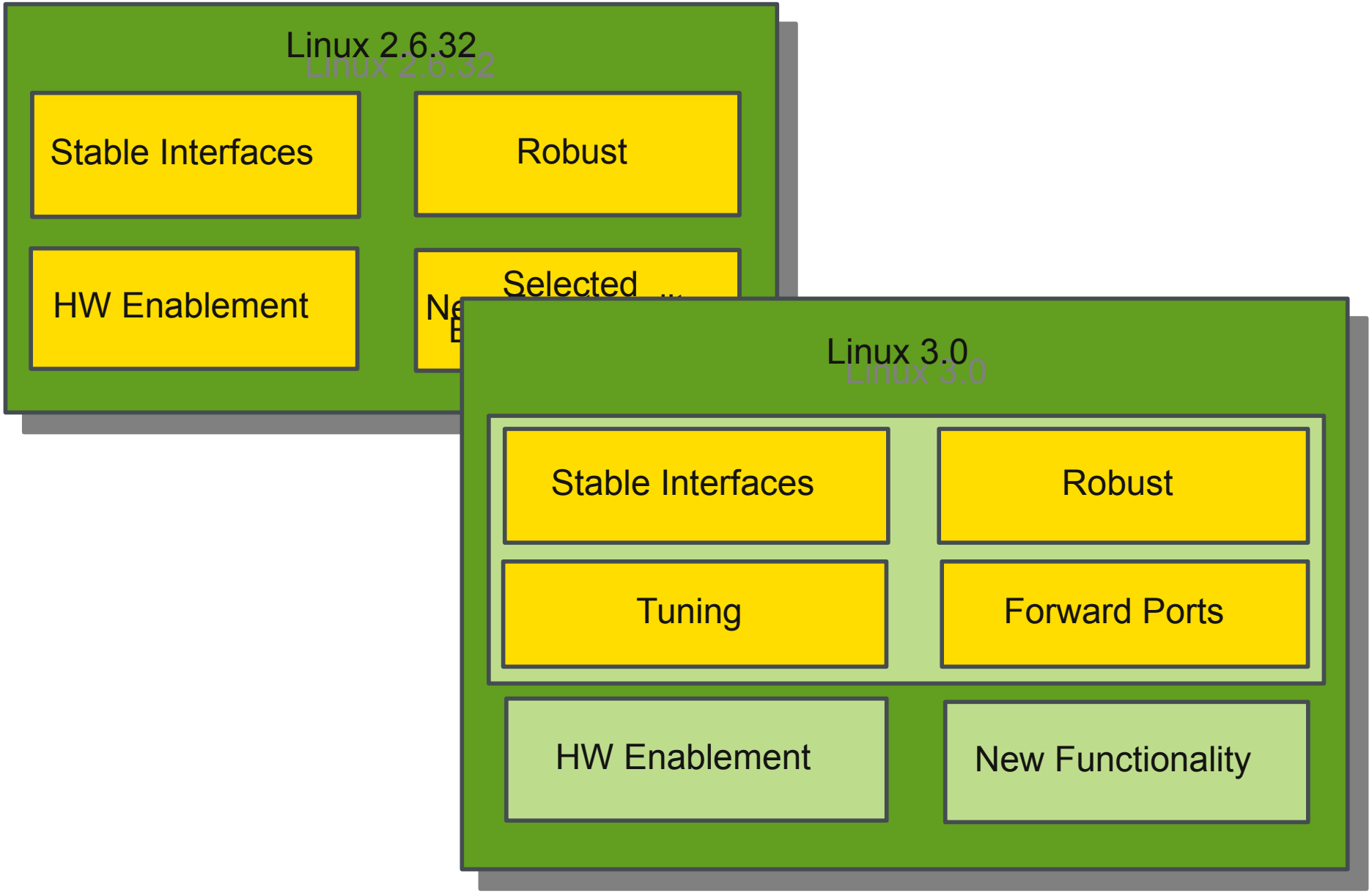
What does this change mean?



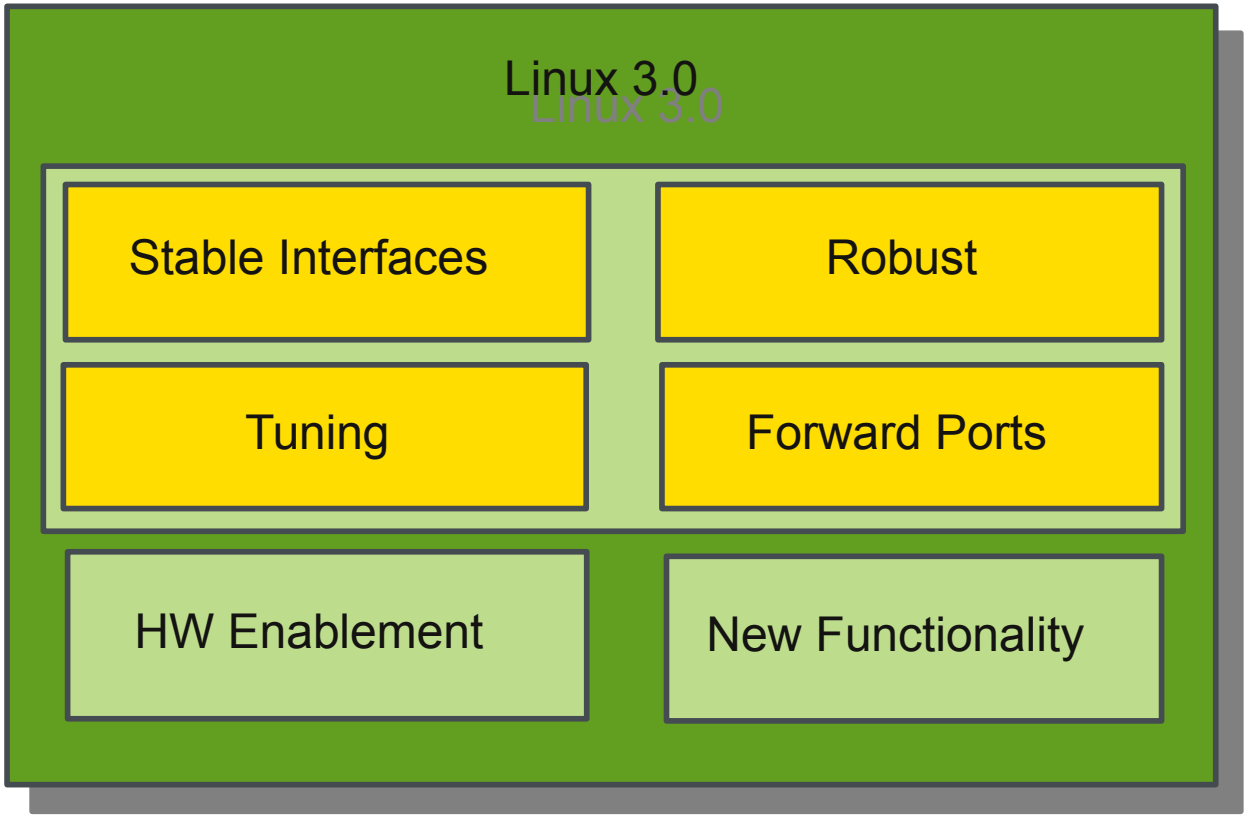
What does this change mean?



What does this change mean?



SUSE Linux Enterprise 11 SP2



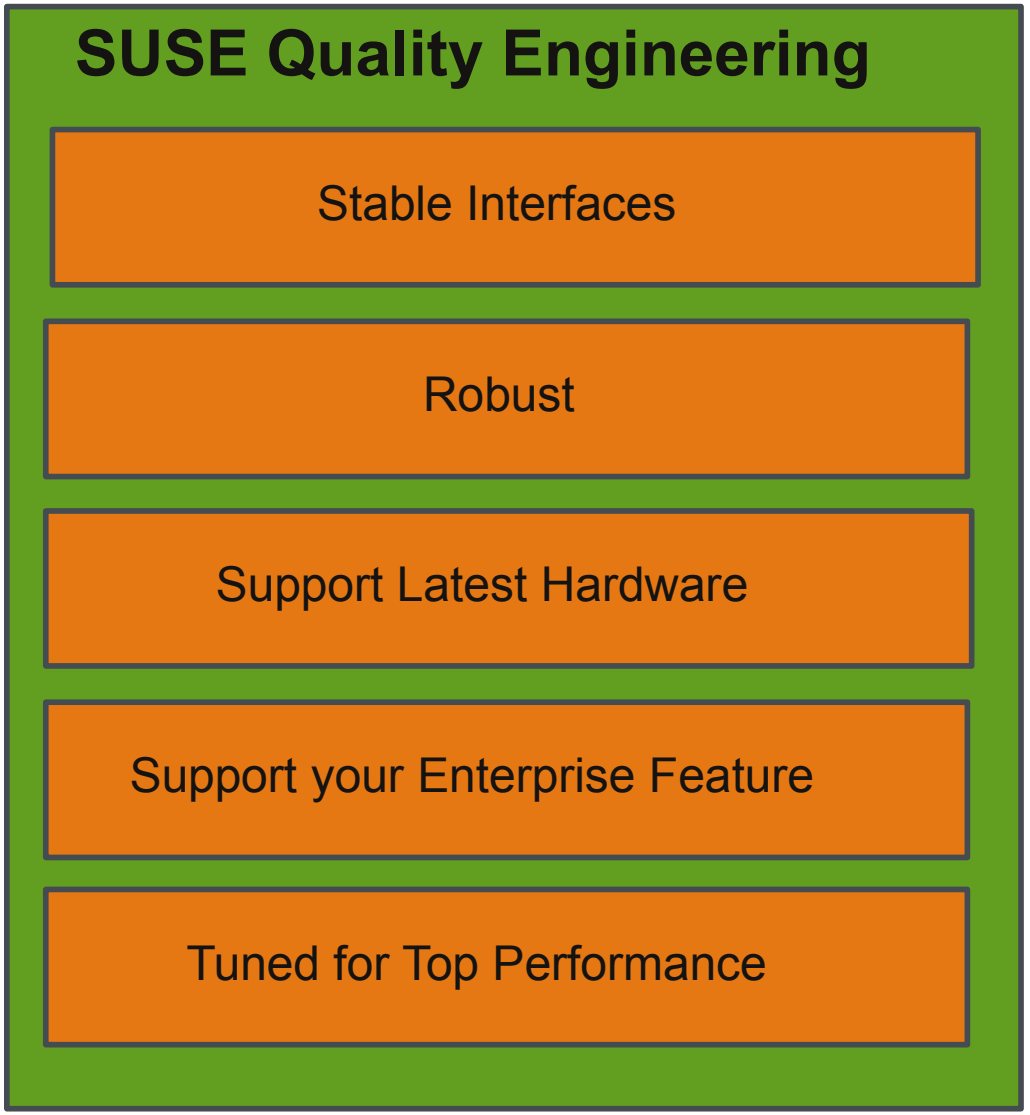
What this means for You!

We deliver the quality you are used to

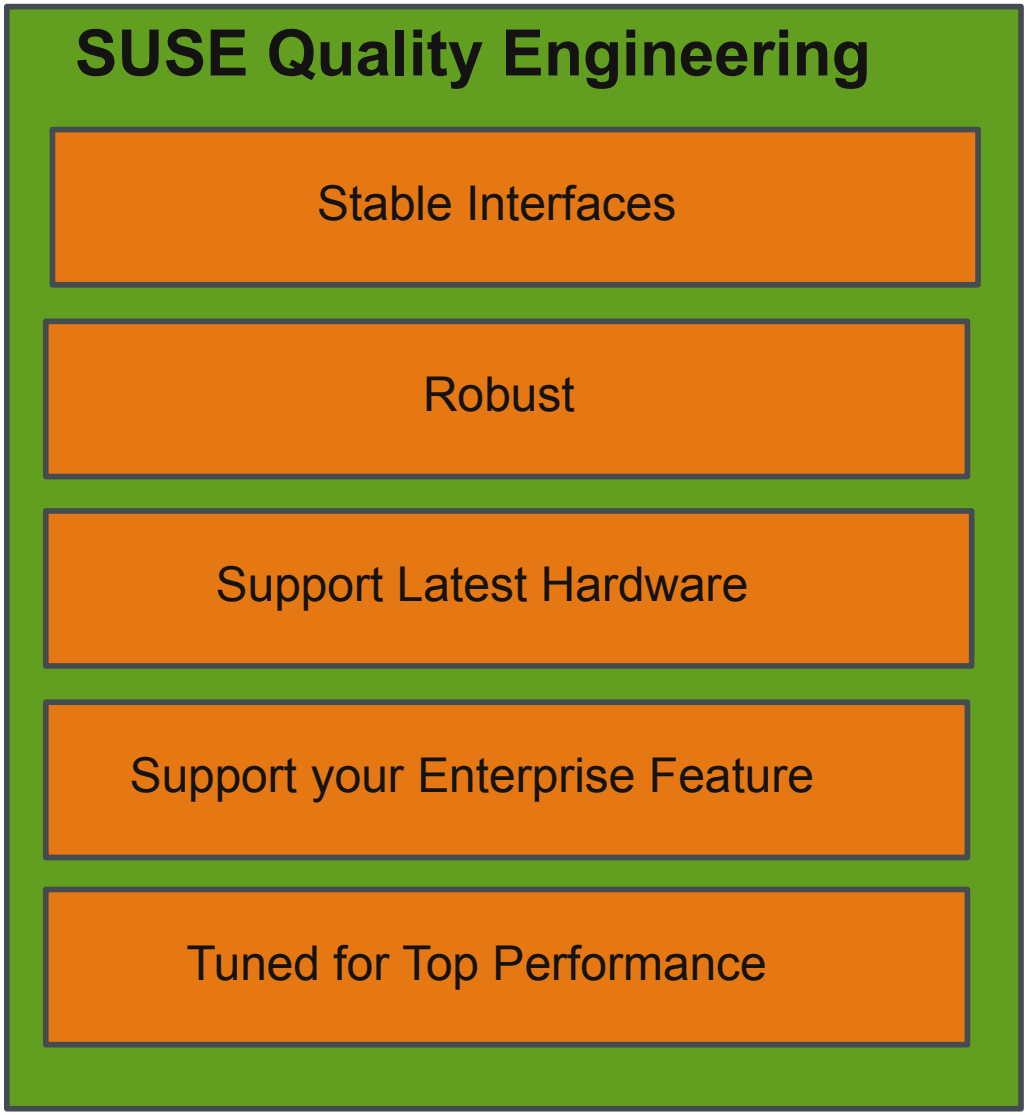
and

we give you more innovation

Yesterday, Today, Tomorrow



Yesterday, Today, Tomorrow



...

suse®

Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE.
Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.
Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.