

# LF Driver Backports & Distribution workgroup

Linux Foundation workgroup introduction

**Susanne Oberhauser**

workgroup chair  
froh@novell.com

**Novell.**<sup>®</sup>

## Susanne Oberhauser

- Linux user since **S.u.S.E. 1.0.9** (July 1994)
- **University** degree “Diplominformatiker” (10/1997)
- Enterprise resource planning **application development** (psi AG, Berlin, and partners until 12/1998)
- Insurance company **IT infrastructure development** (CosmosDirekt insurances, Saarbrücken, until 07/2000)
- SuSE Linux Enterprise / s390 / s390x / zSeries **operating system development** until 2003
- SuSE Technology **Alliances evaluation tool** until 2005
- Novell **Linux driver process project lead**

# Driver Backports and Distribution

New Linux Foundation workgroup:

- System integrators
- Component vendors
- Distributions

Vision: Linux runs 'anywhere'

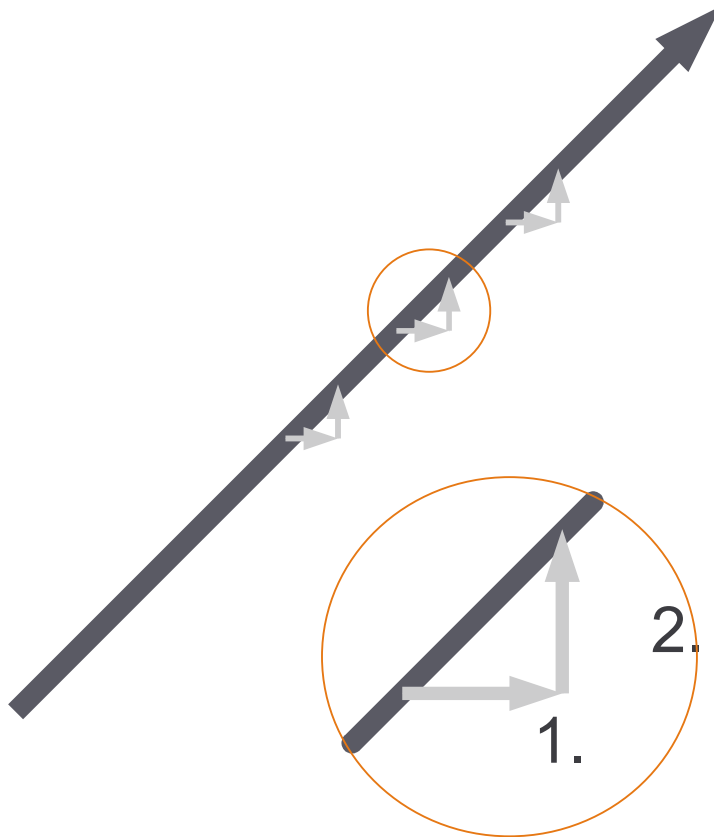
Mission: Simplify driver handling

# Outline

- History
  - The previous, old driver distribution problem
- Current situation
  - Many solutions, too many solutions, too many interfaces
- Proposed action
  - Converge to one solution, one interface
- Next steps
  - Driver backports and distribution workgroup
  - Analysis, design, prototype, standard
  - Open to influence

History: The previous “driver problem”

# Background: kernel community development



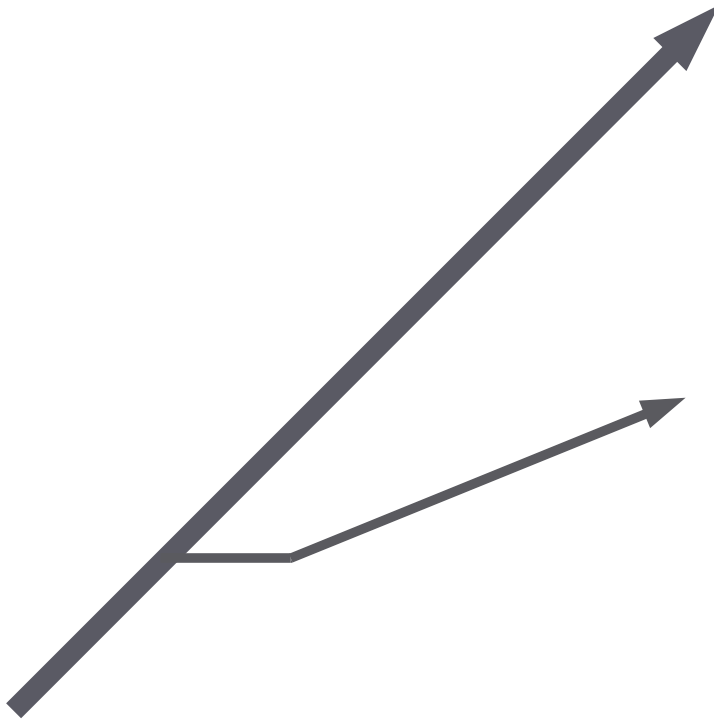
- Continuous integration of enhancements and bug fixes
- One kernel for all
- **Same drivers for all**
- **Driver update:**
  1. Twig locally
  2. Merge upstream

# Background: kernel community testing



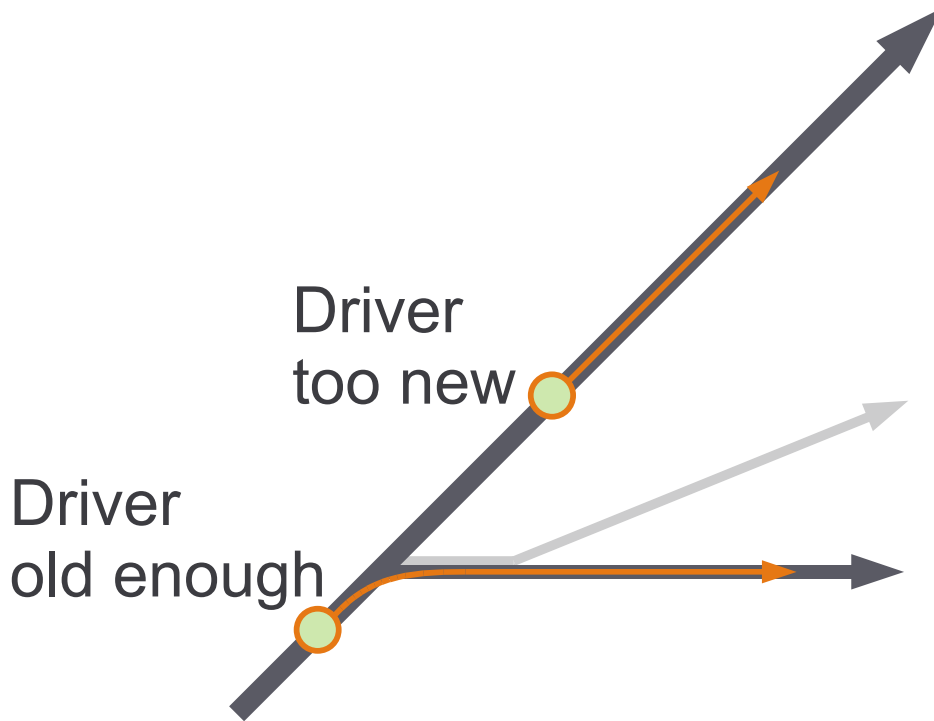
- Community QA:
  - Release early
  - Release often
  - 'Bash' the code
- Bug fixes
  - Quick turn-around time
  - Fix is released to all
- To avoid regressions
  - Use stable release
  - Use distribution release

# Background: kernel community stable release



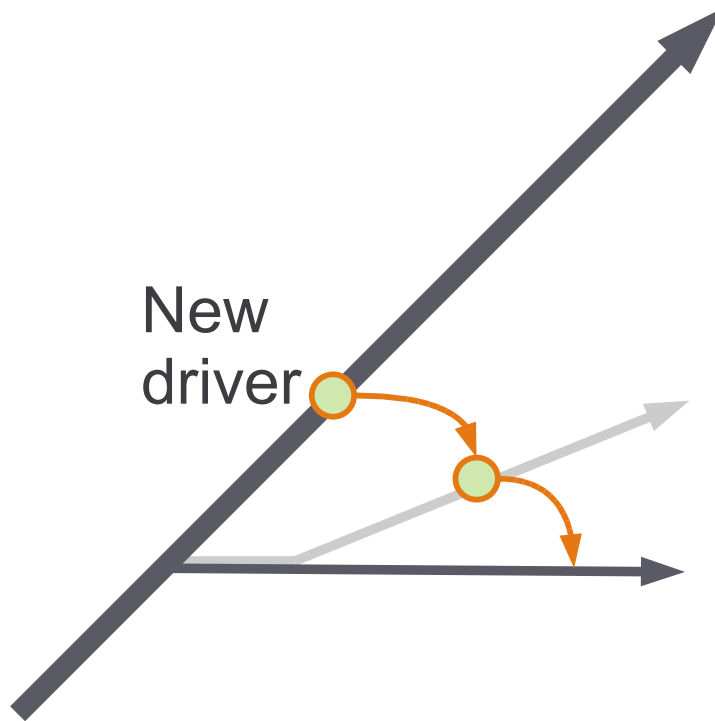
- At some points stable releases branch off the mainline kernel
  - Security bug fixes
  - Selected driver updates
  - If an update to the stable release should break some system, it will be fixed quick as usual
- To furthermore avoid regressions
  - Use distribution release

# Background: Linux distributions



- Linux distributions do even less changes
  - Keep management interface stable (sysfs, procfs, ioctl, ...)
  - Keep API stable
  - Try to keep ABI stable
  - Thorough, systematic QA before release
- Give to get:
  - More reliability, stability
  - Less flexibility

# Background: Linux distributions & driver updates



- Current drivers need to be **backported**
  - Backport becomes difficult as API evolves
- The backported driver needs to be **distributed**
  - Providing in stable is simple
  - Providing in distribution is hard

# History: Linux driver gap I

- **Problem:** no accepted method to support most recent systems with current distributions
  - Distributions don't code level (L3) support arbitrary self-installed code (engineering risk too high)
  - System vendors need distributions' L3 commitment to help themselves (to get the base kernel fixed if needed)
  - Precompiled, well-defined drivers may break with kernel updates (kABI changes)
  - Distributions don't update their kernel for drivers (retesting effort too high)
  - Special one-off kernels don't receive security updates (double effort)

# History: Linux driver gap II

- **Consequences**

- Recent hardware did not work with the distribution (distro)
- System integrator (SI) Linux business happens mostly with most recent distribution
- Distribution business happens mostly directly after system release
- Very high pressure to update the kernel and retest

***Very unfortunate situation for the Linux business***

# History: Linux driver gap III

- **Individual attempts** to ship drivers
  - dkms
  - Support Packs, UpdateExpress, System Packs, Primesetup...
- **Shortcomings** of existing solutions
  - kABI robustness
    - > Make system robust wrt kernel updates?
  - Integration with distribution system management
    - > Install kernel updates, provide driver updates, merge back to distro kernel?
  - Support commitment
    - > Who fixes problems if the customer needs code fixes?

***Need for distribution support***

# History Breakthrough: Distros embrace 3<sup>rd</sup> party drivers

- 2006: **driver process:**  
distributions accept and support 3<sup>rd</sup> party drivers
  - Driver packages (KMPs, kmod, ...)
  - Driver kits (select drivers per target system)
  - kABI awareness (kABI commitments, kABI watches, whitelists)
  - Driver package plug-ins (pci.ids, firmware, user space, xorg, ...)
  - 3<sup>rd</sup> party driver certification (joint support commitment)
  - Driver management (Jockey, YaST support, ...)

# Isn't the driver problem solved now?

- Lots of solutions:
  - KMP, kmod, restricted manager, System Pack, UpdateExpress, Support Pack, PrimeSetup, dkms, ...
- New refinements under way
  - Jockey, build server, ...

***So what?***

The *current* “driver problem”

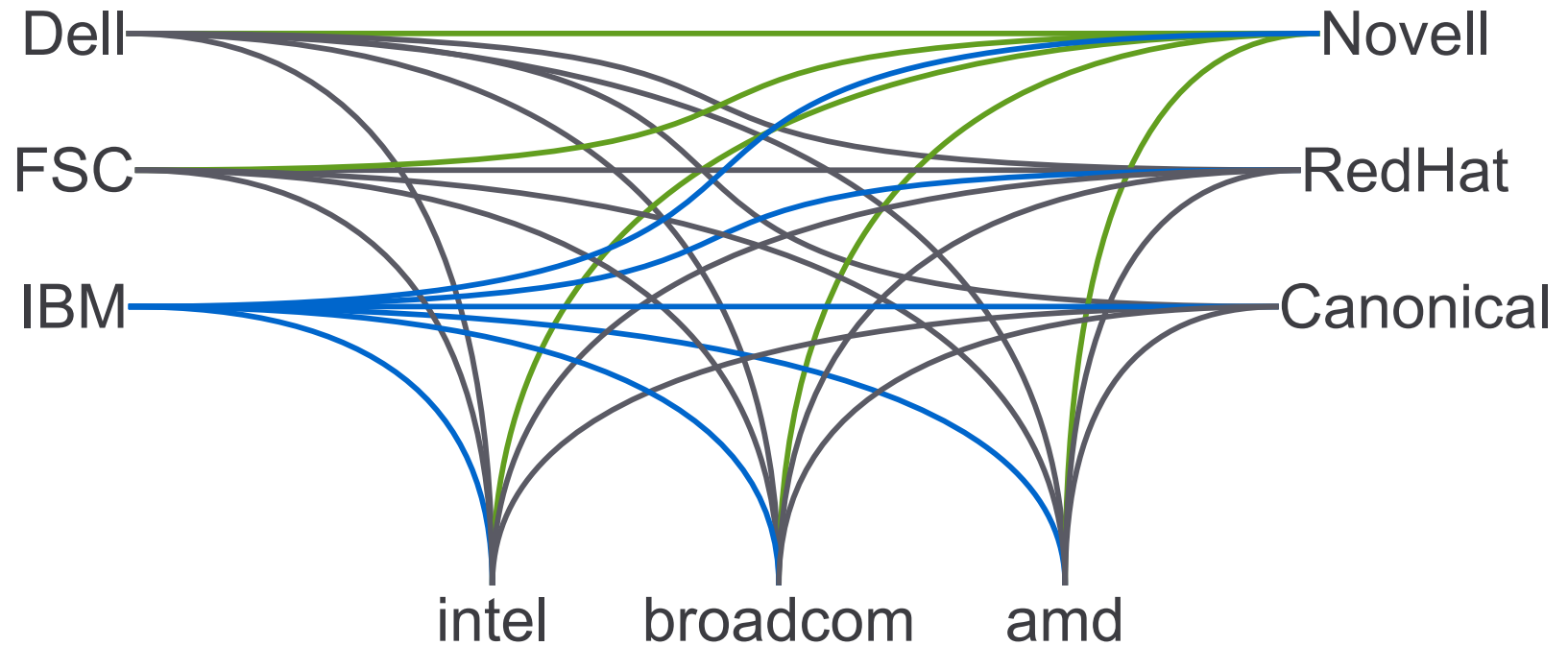
## Current Situation I: **Many worksforme solutions**

- Several component vendor / driver author solutions
  - Upstream and 'out of tree'
  - tgz, diff, zip, shell-archives, ...
- Several system integrator solutions
  - IBM: UpdateExpress, FSC: primesetup, Dell: dkms, HP: SupportPack
- Several distribution solutions
  - SUSE: KMP, driver kits
  - RHEL: kmod, driver database
  - ubuntu: restricted manager

## Current Situation II: **Many interfaces**

- Component vendor interfaces
  - Each component vendor deals with each distro and each system driver shipment method
  - Each component vendor deals with each distro kernel
- System integrator interfaces
  - Each SI deals with each component vendor's driver shipment method and with each distro's driver model
  - Each SI's techsupport deals with each distro's driver toolset
- Distribution interfaces
  - Each distribution deals with each component and system vendor philosophy

## Current Situation III: Many to many relationship



***Many too many interfaces  
Which way to go?***

The background of the slide is a solid blue color with a pattern of diagonal lines in various shades of blue, creating a sense of motion and depth. The lines are more prominent on the right side and fade towards the left.

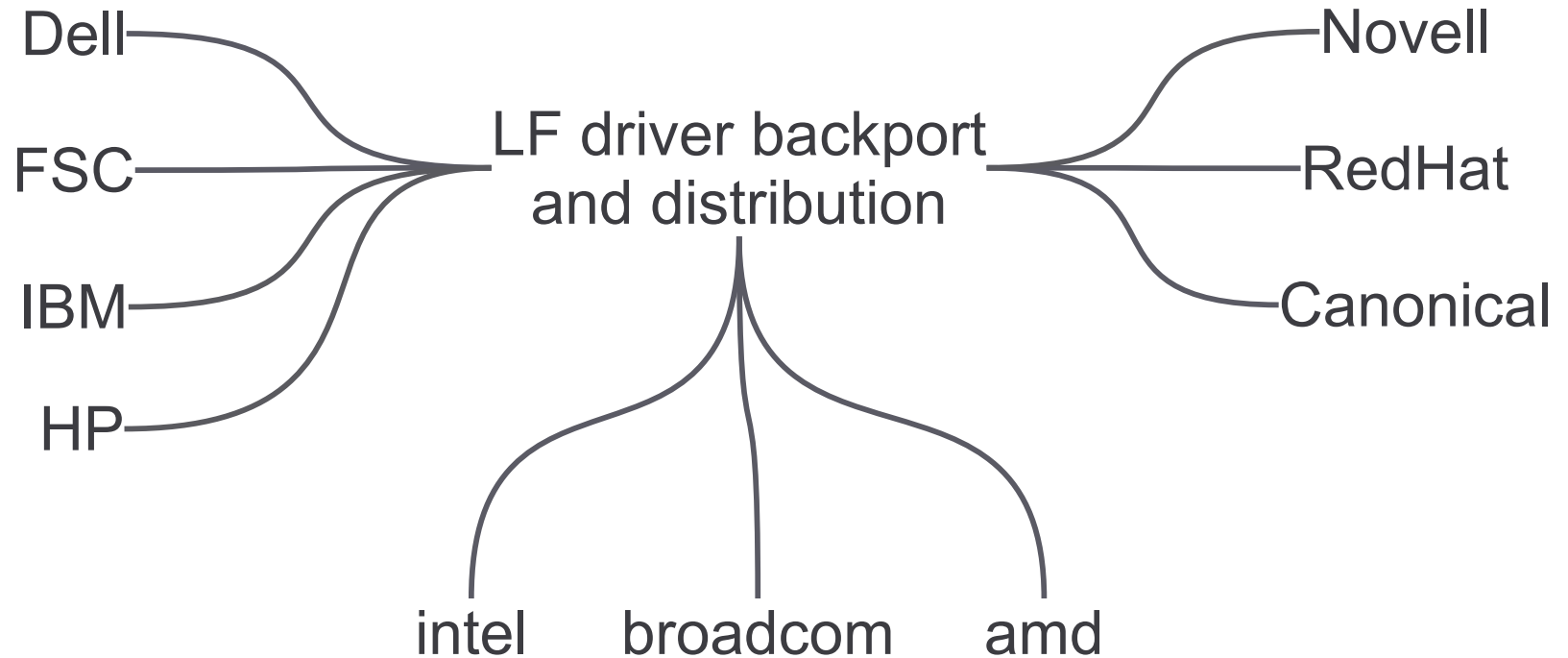
Finding a common solution

# Vision: Many to one

- Component vendor
  - Provides backported drivers in one way to any system integrator or distribution or end user
- System integrator
  - Picks drivers from component vendors in one way
  - Provides drivers to end users or testers in one way
  - Techsupport uses one toolset for driver maintenance
- Distribution
  - Receives drivers in one way for integration in OEM products

***One way of doing Linux drivers***

# Vision: Many to one



***If done right, this will make  
Linux drivers much easier***

# Linux Foundation workgroup



# LF driver backports workgroup

## Agenda

1. Commit to solve the problem
2. Understand the problem
3. Design the solution
4. Prototype and develop a solution
5. Agree on tools and infrastructure in the LSB

# Commit to solve the problem

- Some active workgroup members
  - Susanne Oberhauser, Novell; Jon Masters, Red Hat; Martin Pitt, Canonical
  - Ram Pai, IBM; Matt Domsch, Dell; Martin Wilck, Fujitsu Siemens Computers
- Bi-weekly call, active mailinglist
- Austin, TX at the LF collaboration summit:
  - first face2face: alignment, design, share prototypes

***Agreed workgroup objective:  
Reduce the number of interfaces***

# Understand the problem I

- Driver lifecycle

- Creation

- > In-house driver during hardware alpha, beta
    - > Upstream submission at hardware release
    - > Distribution kernel backport for certifications
    - > Distribution driver package build (KMP / kmod)
    - > “driver kit” creation (driver package bundling per target system)
    - > Final test, release, install

- Support

- > Update to current driver, test, fix driver, ship it

- Maintenance

- > Provide current driver to all systems that need it

# Driver lifecycle: Participants

**CV**

Component  
Vendor

**SI**

System  
Integrator

**User**

Sarah  
Sysadmin

**Kernel**

Open Source  
Community

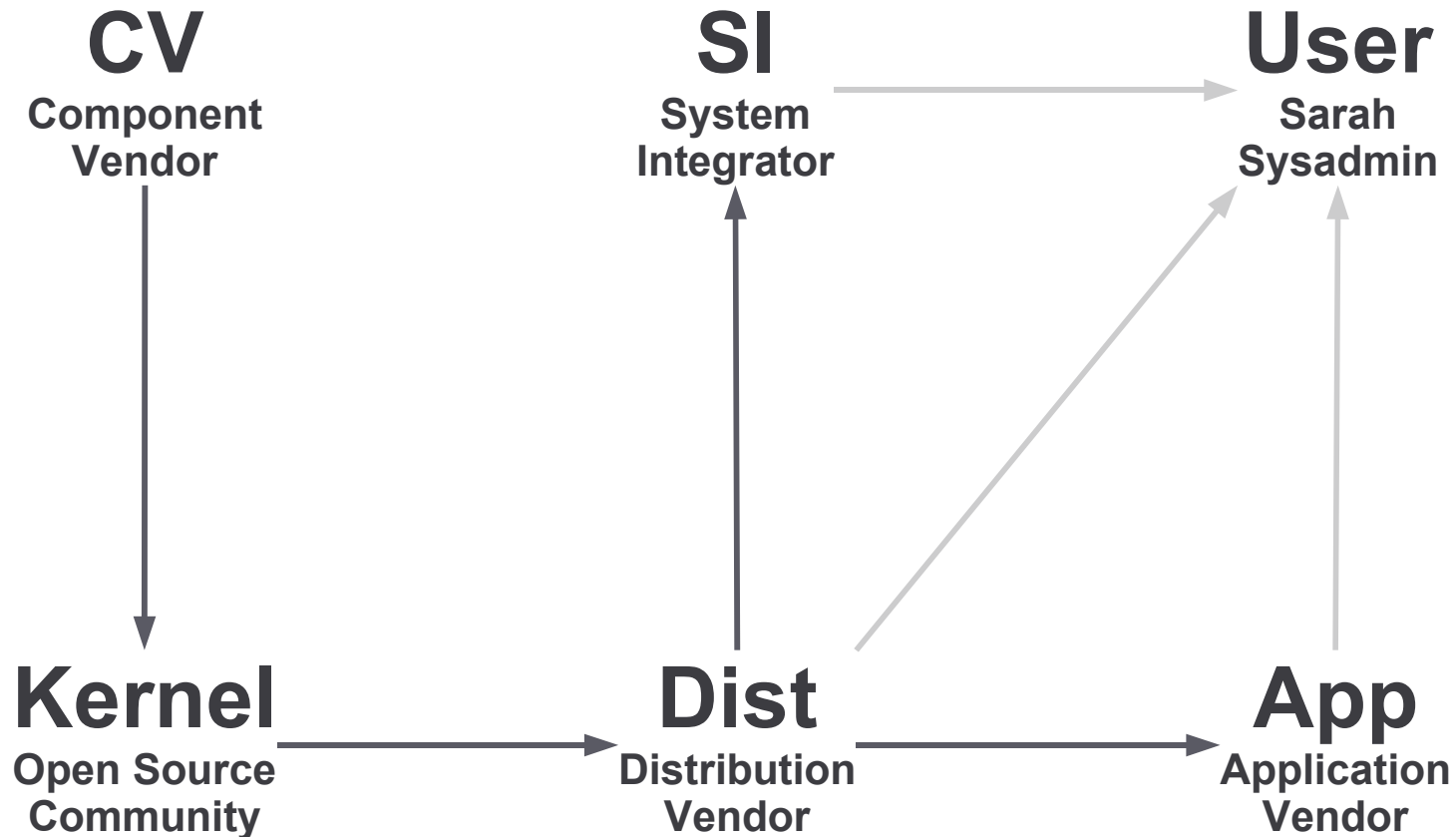
**Dist**

Distribution  
Vendor

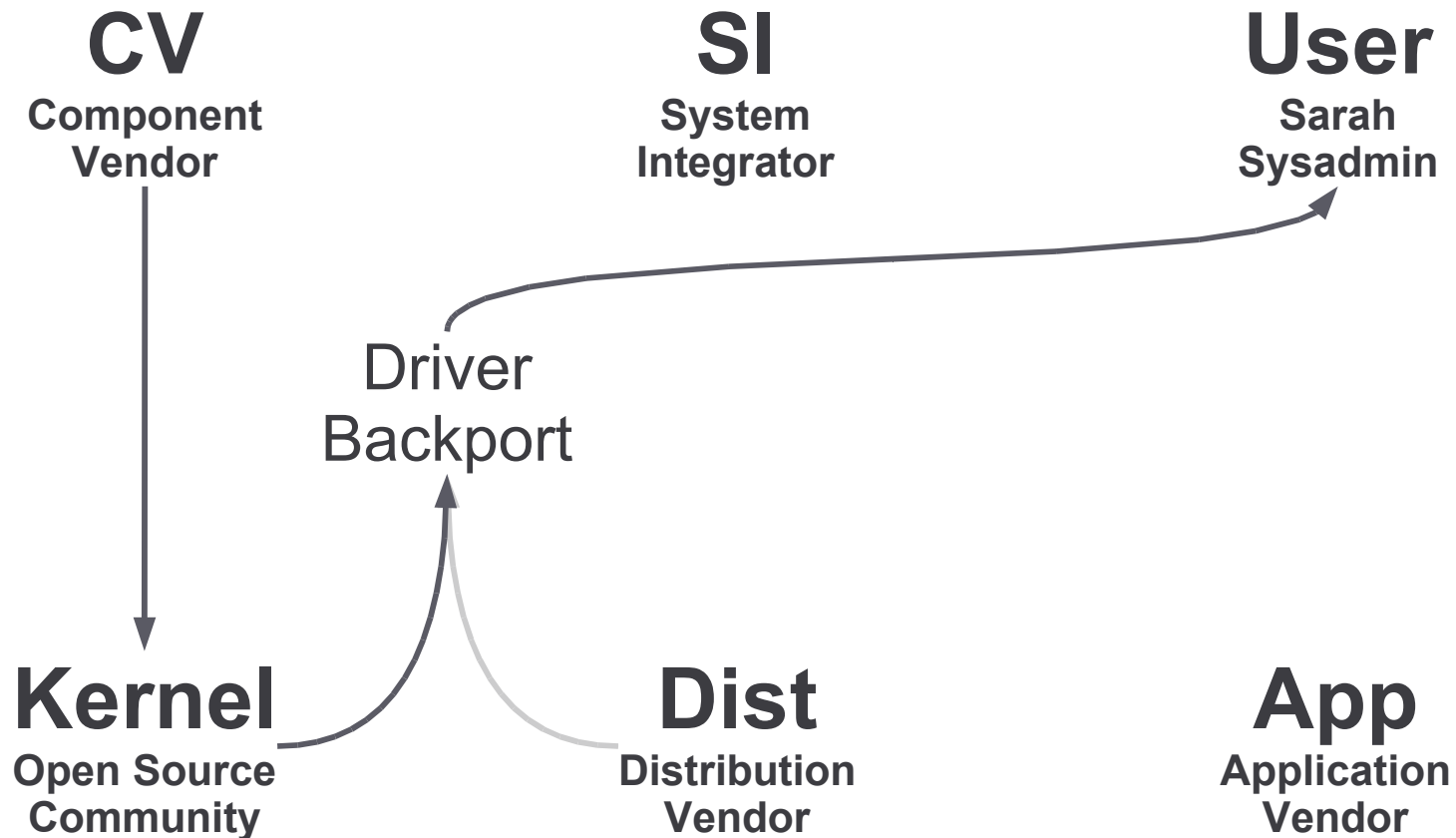
**App**

Application  
Vendor

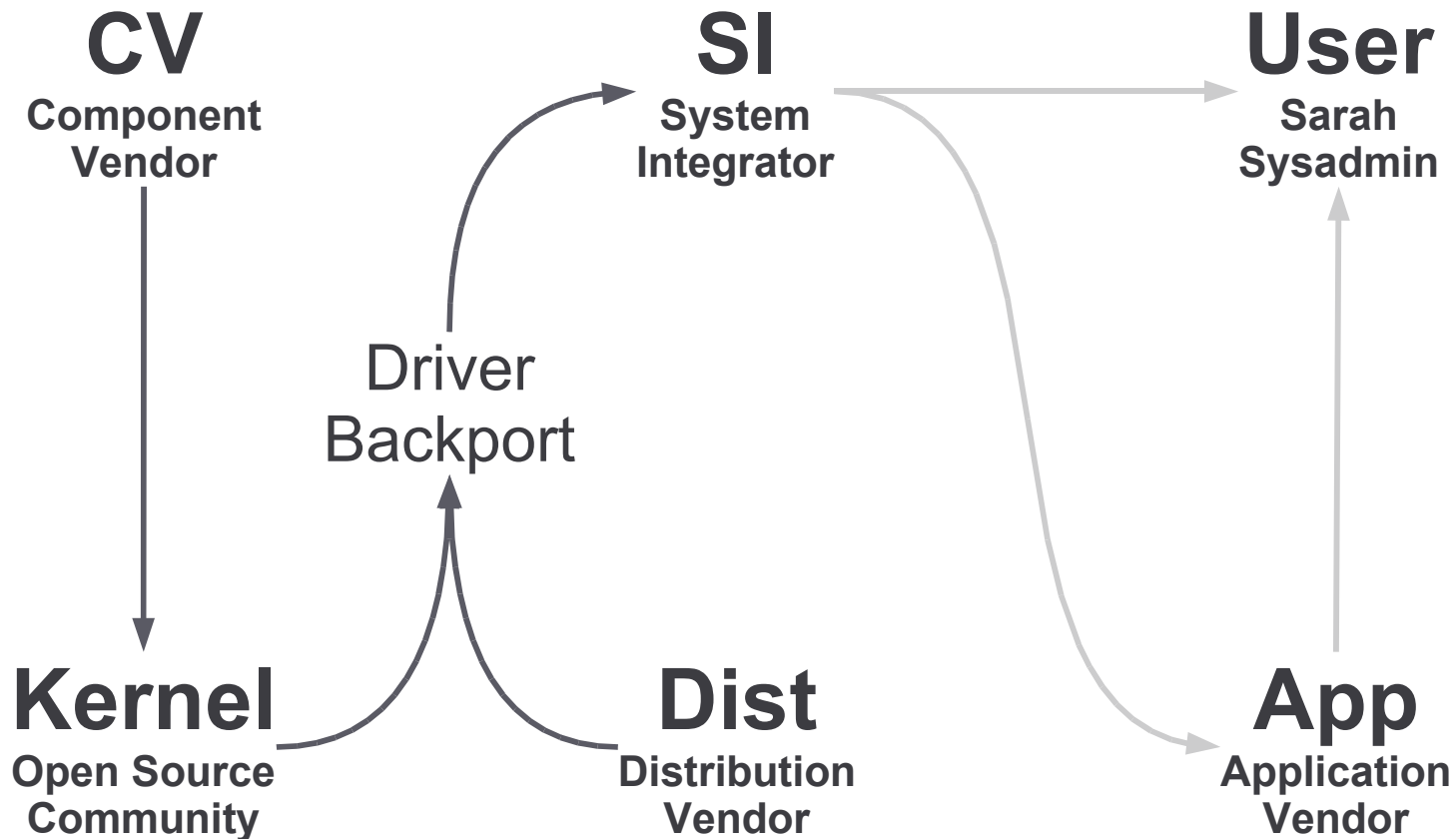
# Driver lifecycle: Preexisting hardware



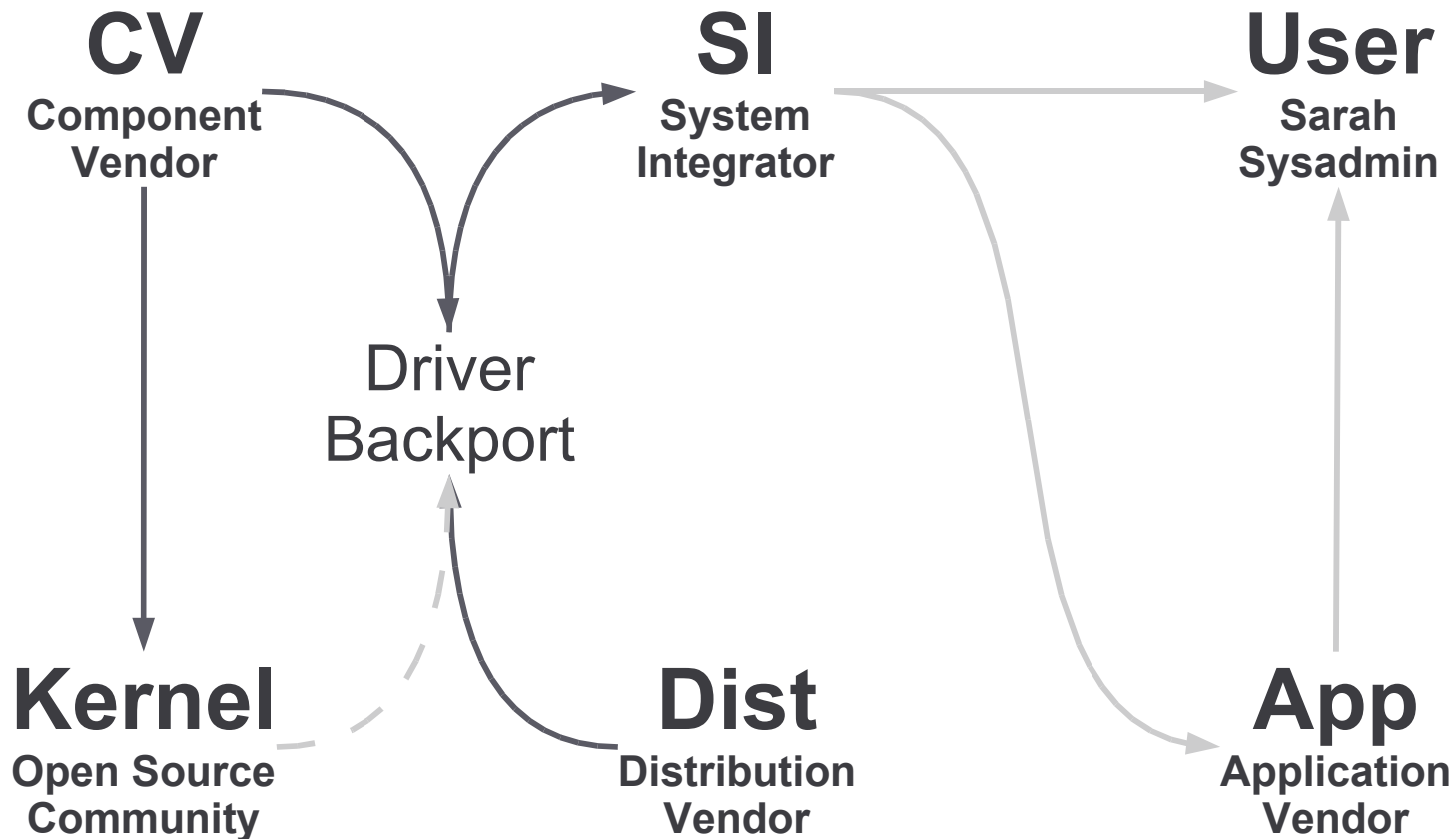
# Driver lifecycle: New component



# Driver lifecycle: New system



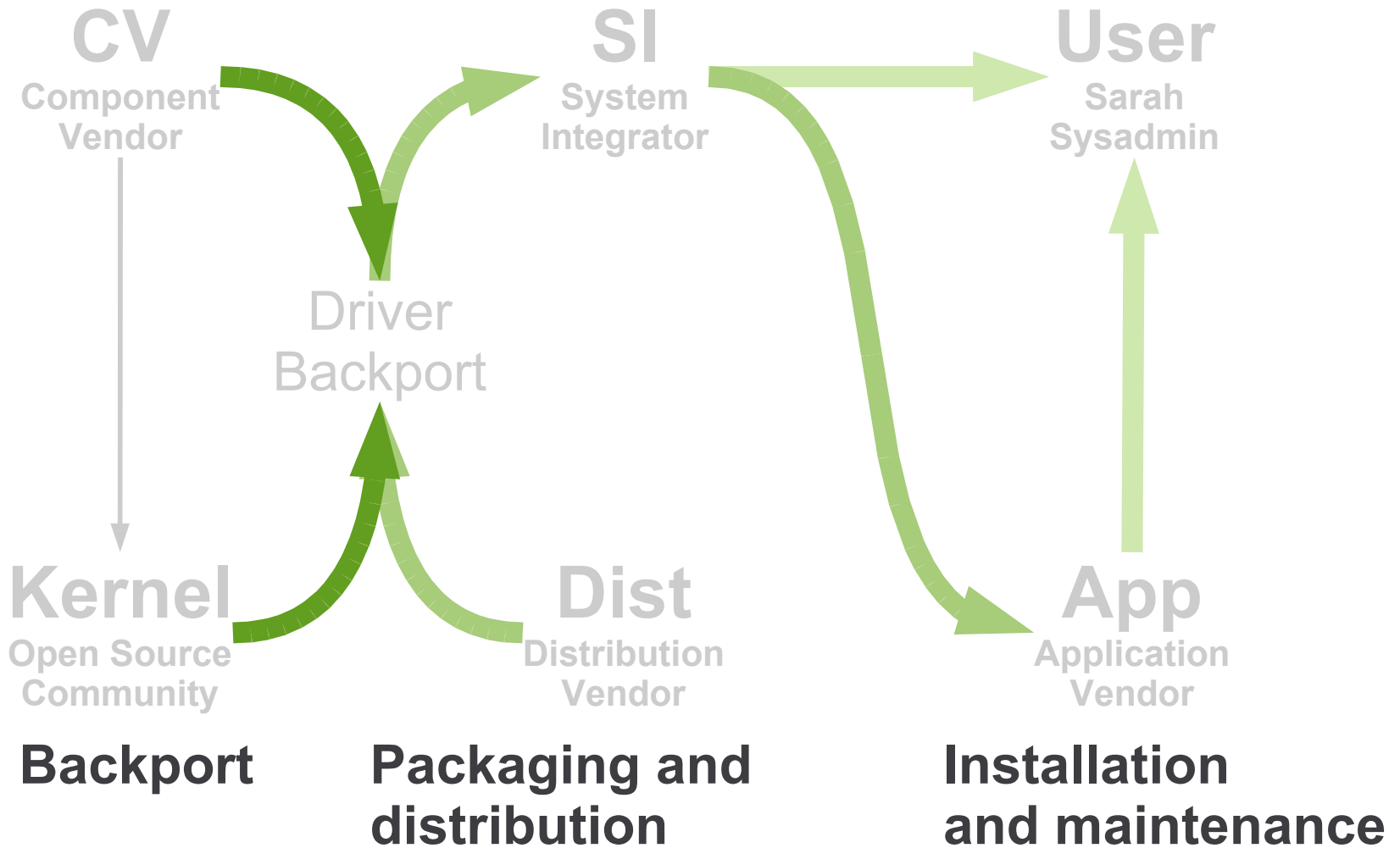
# Driver lifecycle: Driver bug fix



# Understand the problem II

- Problem is cleanly partitioned
  - Driver **backport**
    - > Source and backport patch management
    - > Driver backports and dynamic upstream API
  - Driver **packaging and distribution**
    - > Build drivers for distribution kernels
    - > Aggregate built drivers for specific usage contexts
    - > Provide driver aggregations for installation and use
  - Driver **installation and maintenance**
    - > Select “The Right Driver” (TM) for this system
    - > “Do Things Right” (TM) for kernel, system, driver updates

# Driver lifecycle: partitioning



# Design the solution

- Interfaces to other efforts
  - This is about backporting the fruits of <http://linuxdriverproject.org/>
  - The result needs to get into LSB
  - How about user space drivers, like printer drivers?
  - Upstream support for API backports?
- Align on analysis and architecture
  - Analysis/design document: OLS'08 whitepaper

# Prototype and develop a solution

- Currently ongoing efforts:
  - Source management: dkms
  - Package format: kmod and KMP sync
  - Build infrastructure: [driverbuild.suse.de](http://driverbuild.suse.de)
  - Driver database: [drivertool.org](http://drivertool.org), [kerneldrivers.org](http://kerneldrivers.org)
  - Driver installation and management tool: Jockey

# Tools and infrastructure into LSB

- Once the solution works ...
- ... any LSB compliant distribution shall support the official Linux driver distribution model

Next steps: This is an open workgroup

# Next Steps

- Next Steps
  - ETA April, Austin: align on analysis
  - ETA June, Ottawa: jointly present process architecture, gaps and tools prototypes
- The workgroup would like to ensure all industry needs are covered
  - We very much welcome any SI, distro and component vendor
- Contact:
  - [http://www.linux-foundation.org/en/Driver\\_Backport](http://www.linux-foundation.org/en/Driver_Backport)
  - [http://lists.linux-foundation.org/pipermail/lf\\_driver\\_backport](http://lists.linux-foundation.org/pipermail/lf_driver_backport)
  - freenode: #driverbackports

The background of the slide is a solid blue color with a pattern of diagonal lines in various shades of blue, creating a sense of motion and depth. The lines are most prominent on the right side and fade towards the left.

Thank you.

## **Unpublished Work of Novell, Inc. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell 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 Novell, Inc. 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. Novell, Inc. 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 Novell products remains at the sole discretion of Novell. Further, Novell, Inc. 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 Novell 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.

