

# Contributing to the openSUSE distribution

Hendrik Vogelsang / Adrian Schroeter  
openSUSE Team  
Novell

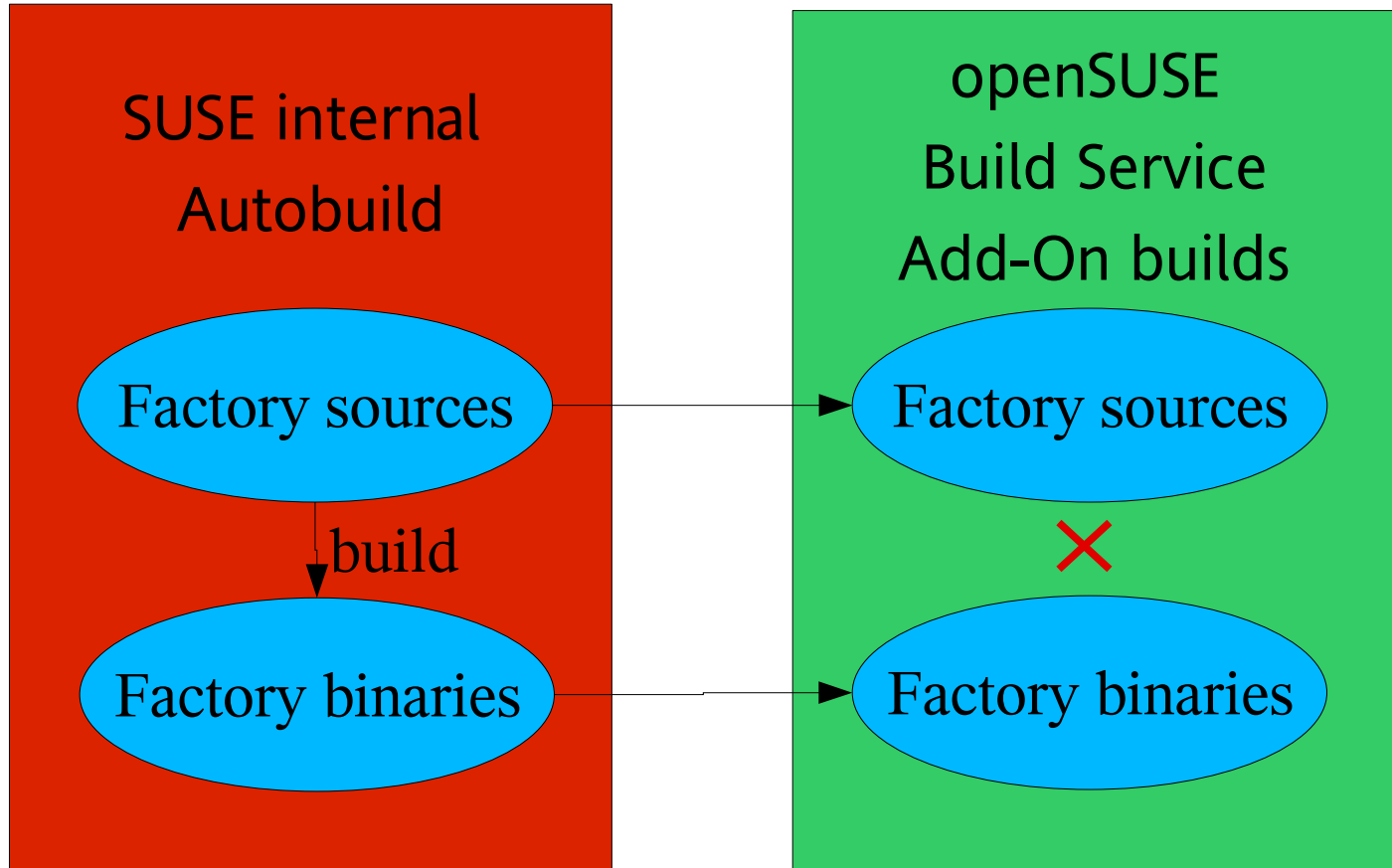


**Novell.**

Status Quo  
How it is



# Status Quo – How it is





# Status Quo – How it is

---

- Everybody wants to submit changes as easy as possible
  - Everybody has a different trust relationship
  - Packagers have different working methods
  - Packages have different requirements
  - Everybody wants a stable base for testing his changes
  - The software pool within openSUSE build service is used for multiple distributions, with different requirements
- > multiple workspaces, defined processes



# Status Quo – How it is

## Example Quality Requirements

---

SLE / EAL class requirements:

- Full source review before check-in
- 7 years of full maintenance

openSUSE distribution requirements:

- Source peer review ( is this really enough ? )
- 2 years of security maintenance

Leaf package requirements:

- Fulfillment of guidance principles
- The package should benefit the whole distribution

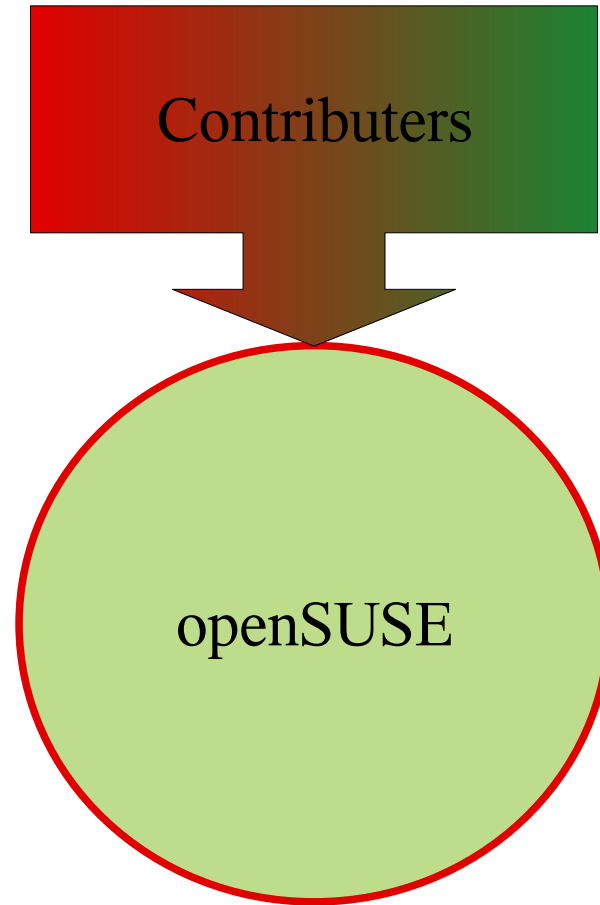
# Contribution Processes

How it could be



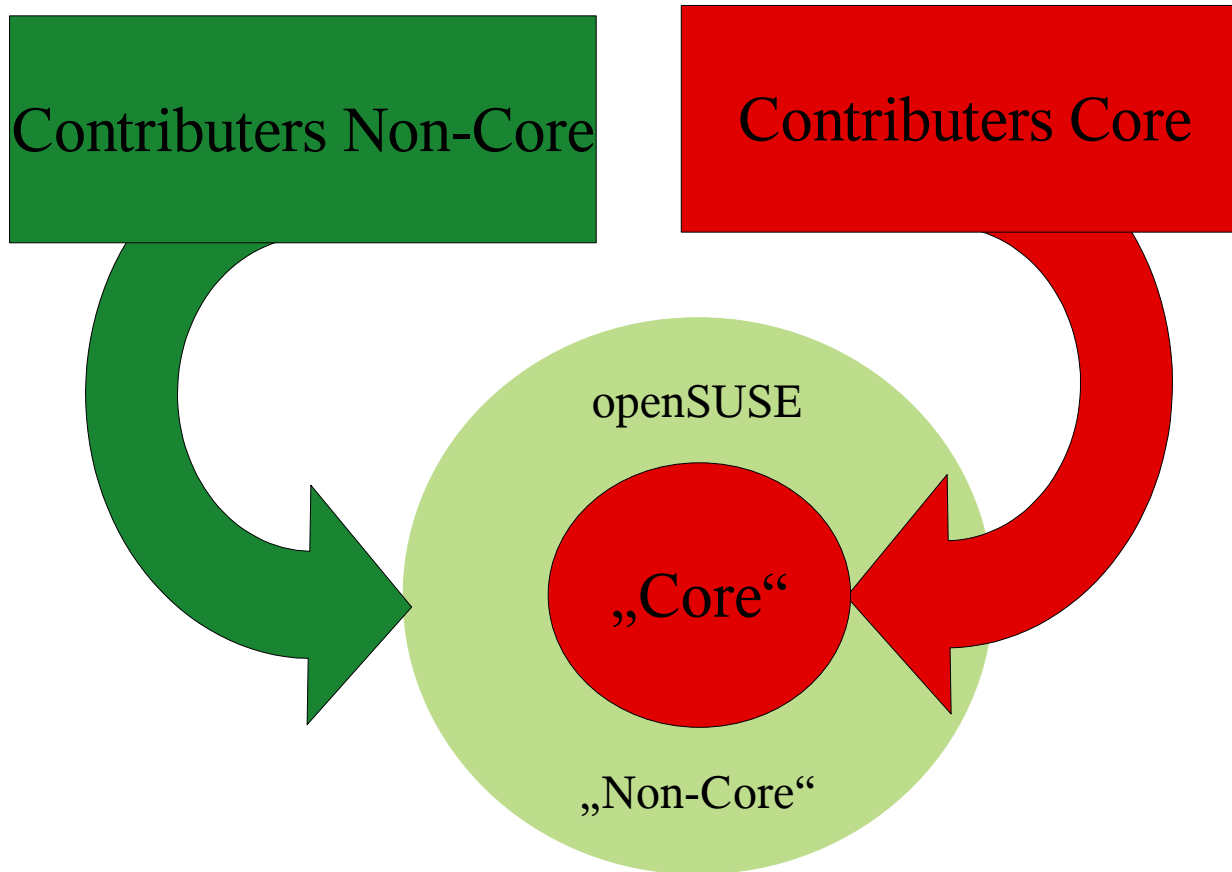
# How it could be

---





# How it could be





# Contribution to the Distribution

Which packages should be part of the distribution?



# Package based Contribution Process Requirements

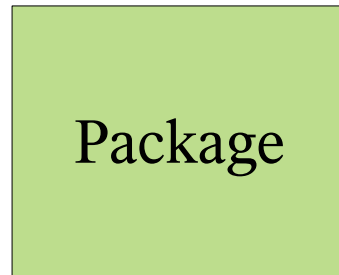
---

- Make contributions as easy as possible
- Involve all types of contributors. From Beginners to the most powerful ones
- Only the best packages become part of openSUSE
- openSUSE must be innovative and at the same time stable
- openSUSE must be secure and maintained



# Package based Contribution Process

---





# Package based Contribution Process

Package	Attributes
	<pre>is_stable_version_number=true number_of_installs=10000 number_of_open_bugs=1 package_exists_since_days=365 upstream_active_development=true listens_on_port=false is_setuid=false runs_as_system_daemon=false runs_as_root=false code_public_in_vcs=true number_developers=20 widely_used=true reviewed_code=false reviewed_code_by_secteam=false package_maintainer_trust_level=100 reviewed_spec=true unique_functionality=false provides_core_functionality=false desired_functionality=true number_of_dependent_packages=6 upstream_has_stable_branch=false upstream_exists=true maintenance_provider_exists=true osi_approved_license=true links_incompatible_license=false was_before_part_of_distro=true</pre>



# Package based Contribution Process

Package	Attributes
	<ul style="list-style-type: none"><li data-bbox="956 692 1168 749">● Quality</li><li data-bbox="956 756 1188 813">● Security</li><li data-bbox="956 821 1120 878">● Trust</li><li data-bbox="956 885 1246 942">● Innovation</li><li data-bbox="956 949 1294 1006">● Maintenance</li><li data-bbox="956 1013 1130 1071">● Legal</li></ul>



# Package based Contribution Process

---

Info Provider:

- The Package owner
- The distribution owner
- A user rating/trust system
- A package popularity system
- Bugzilla
- BuildService
- Different openSUSE Teams (Security, Usability etc.)



# Package based Contribution Process

---

```
security_relevant = listens_on_port || is_setuid || runs_as_system_daemon || runs_as_root;
```

```
quality_value = is_stable_version_number * 100  
+ upstream_active_development * 50  
+ upstream_has_stable_branch * 100  
- number_of_new_red_bugs_last_6_months * ( 10 + security_relevant * 20 )  
- number_of_open_bugs * ( 5 + security_relevant * 10 );
```

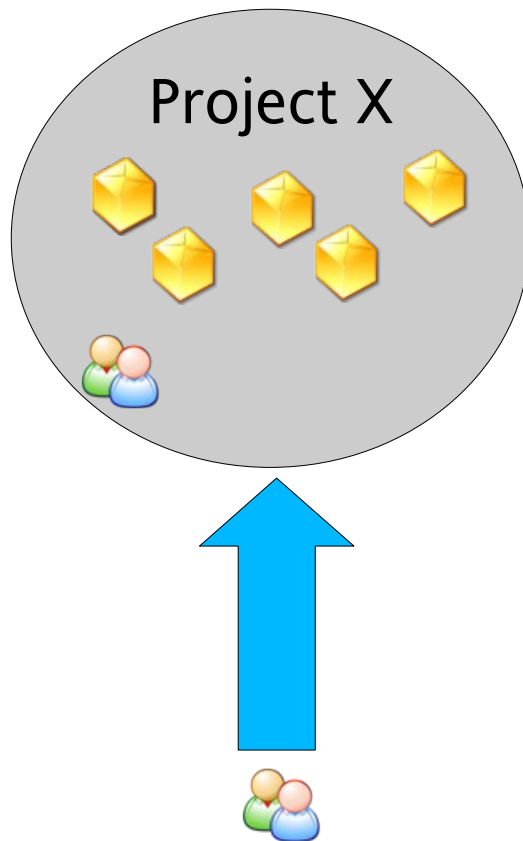
```
wanted_package = provides_core_functionality == true  
|| widely_used == true  
|| ( unique_functionality == true && desired_functionality == true )  
|| ( was_before_part_of_distro == true && desired_functionality == true );
```

# Contribution to Packages

Working on Sources/Packages



# Future Source Handling Process I

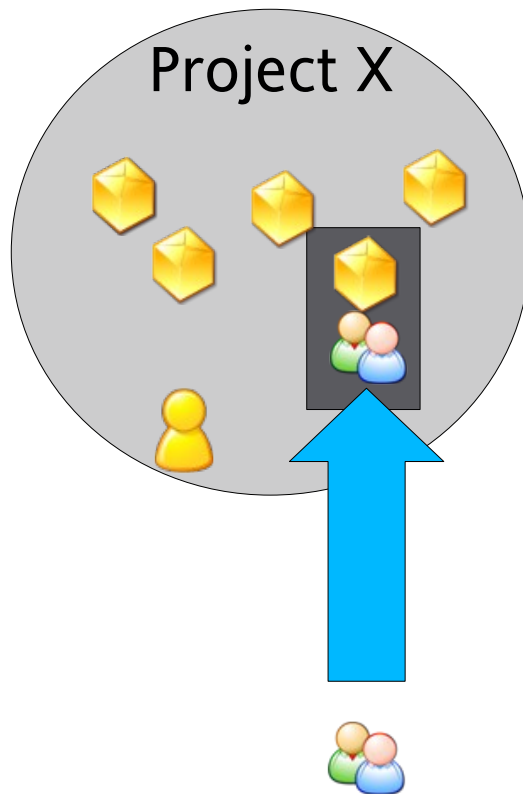


Open Project:

- + Immediate effect in project
- + fastest way to work on project
- Peer review only
- Can block finish package building
- Everybody can add/remove packages

\* Defined users  can do everything

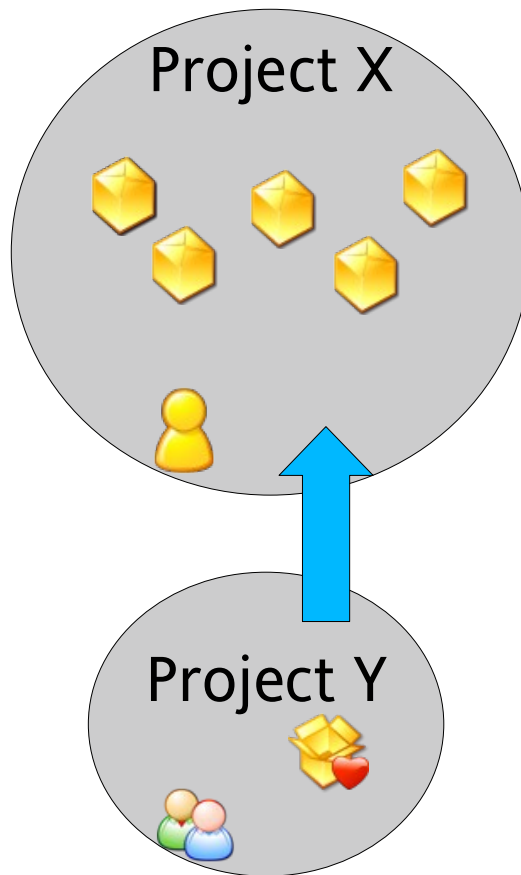
# Future Source Handling Process II



## Open Packages:

- + Immediate effect in packages
  - + fastest way to work on package
  - Peer review only
  - Can block finish package building
- \* Admin (👤) defines used packages

# Future Source Handling Process III



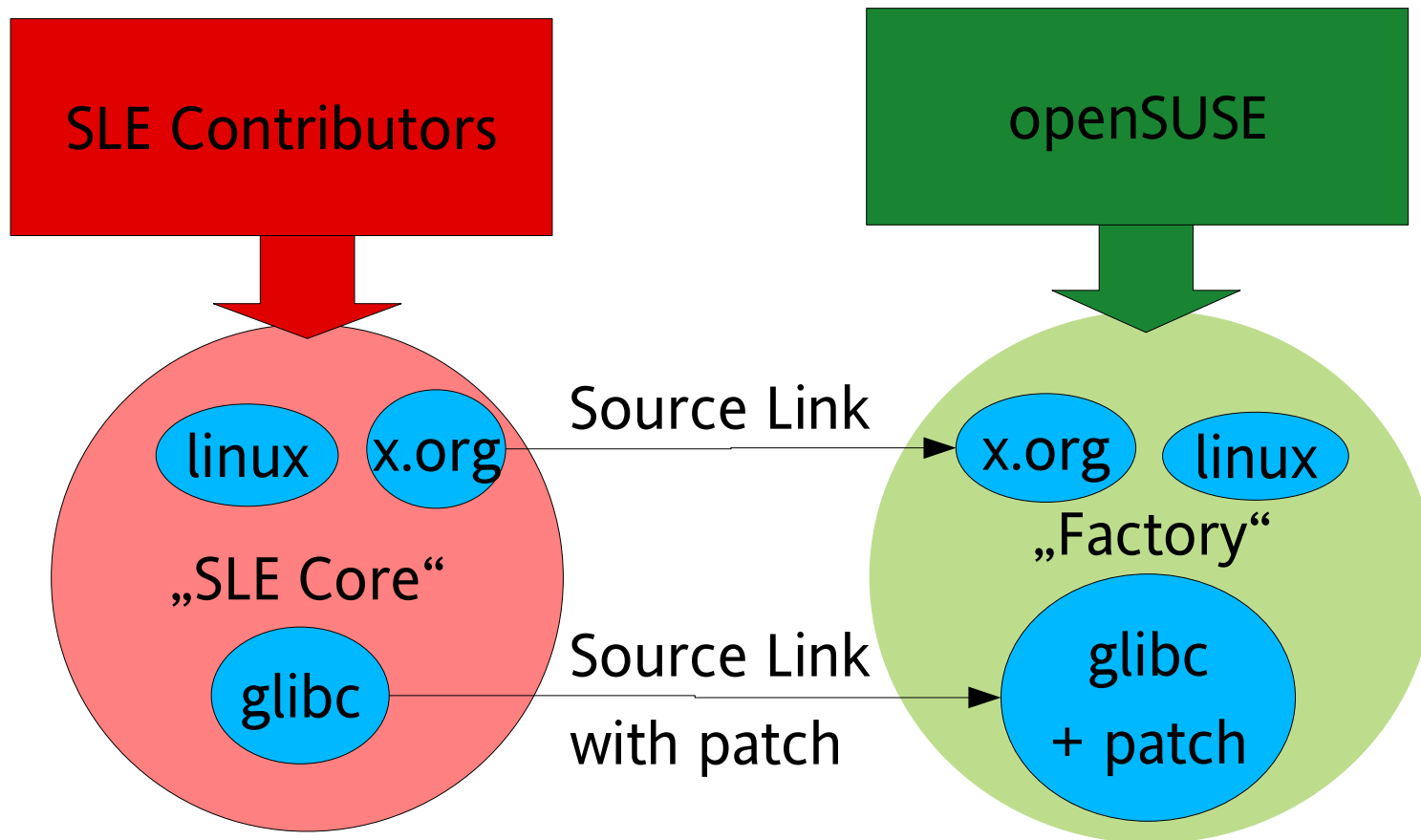
Closed Project:

- + Review before check-in
- + QA happens before check-in
- + Time coordinate check-ins
- Slows down development in X

\* Admin (👤) merges requested changes or refused with given reason

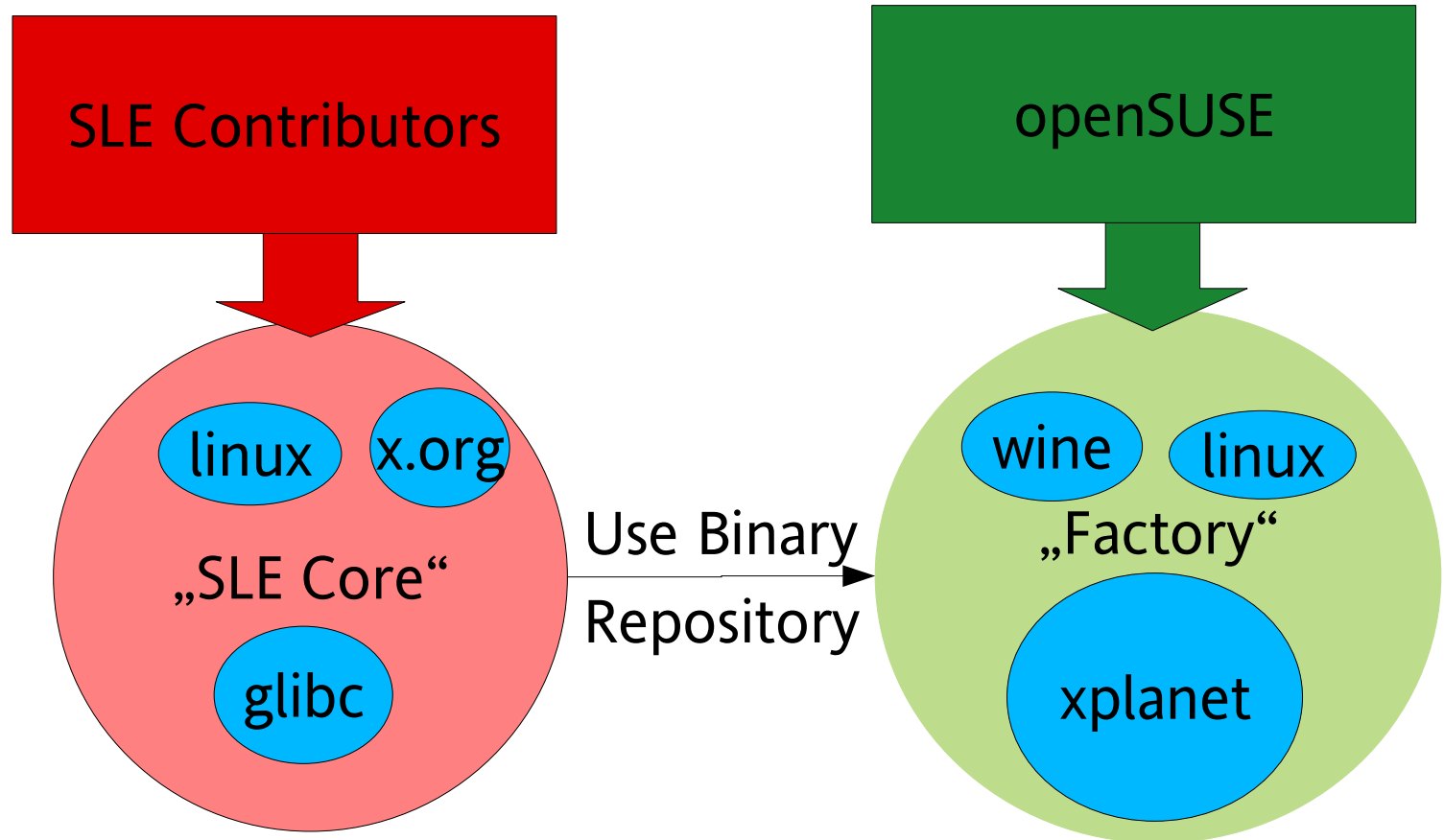


# A Possible Example Setup





# Another Possible Example Setup



Missing Parts



# Missing Components

---

- Efficient patch handling
- Merge handling
- Defined package requirements for Factory  
-> [en.opensuse.org/Contribute](http://en.opensuse.org/Contribute)
- User Trust Relationship System
- Relation of bug statistics to packages
- Overall UI concept

## General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. 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. 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.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/>.

For other licenses contact author.



**Novell.**