



# On Software Release Engineering

Bram Adams



M  
C  
I  
S



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty times**  
a day.

in 00 different countries!

Sign in with:



[Choose Your FREE Avatar](#)

Over 2 Million people  
like IMVU on Facebook!

FREE

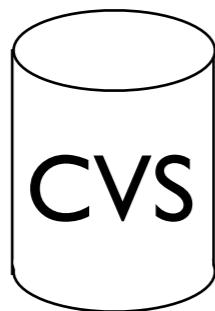
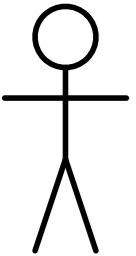
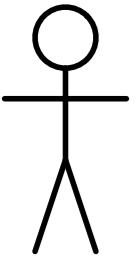
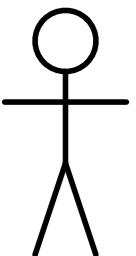
M  
C  
I  
S



# Continuous Delivery

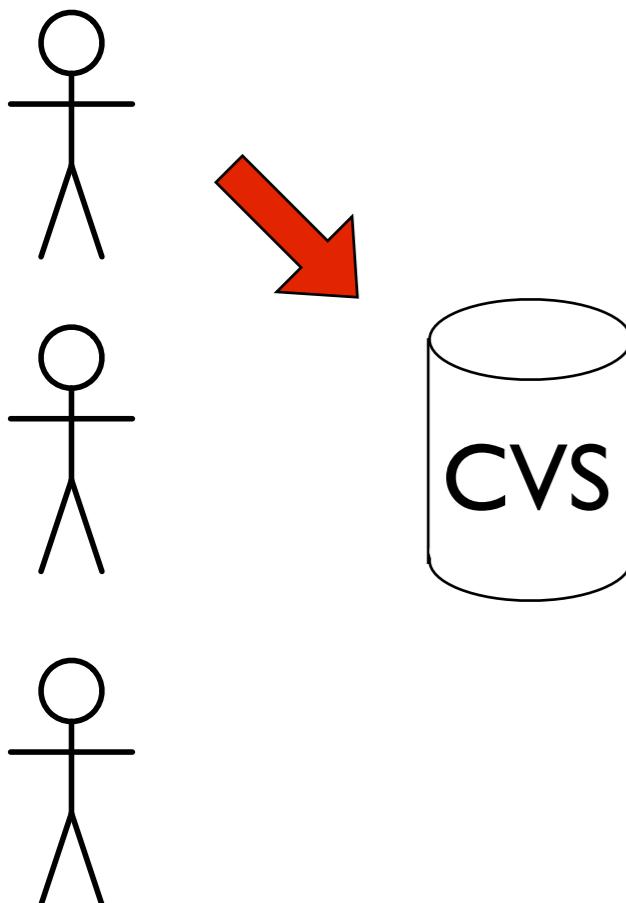


# Continuous Delivery





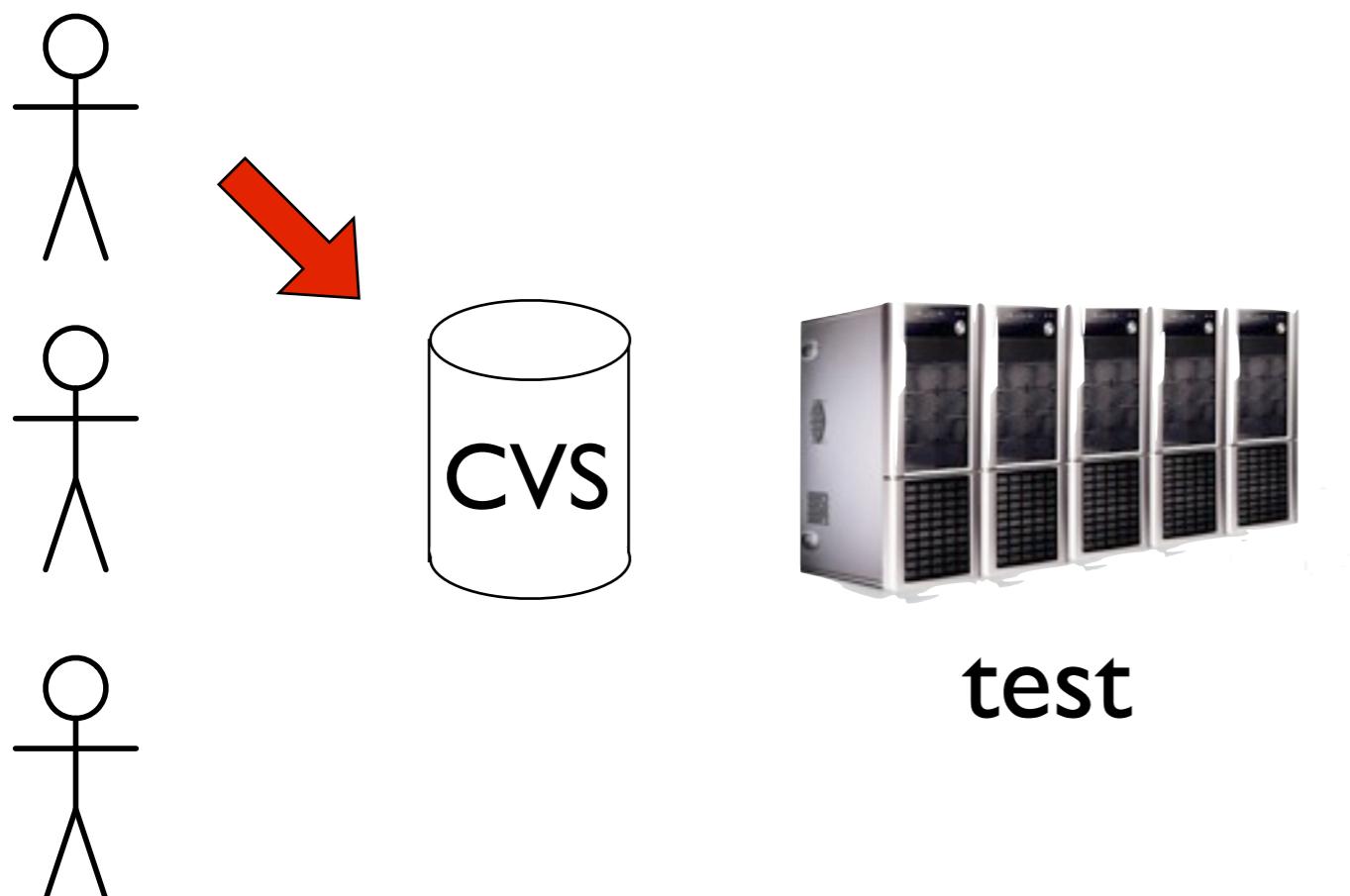
# Continuous Delivery



continuous  
integration



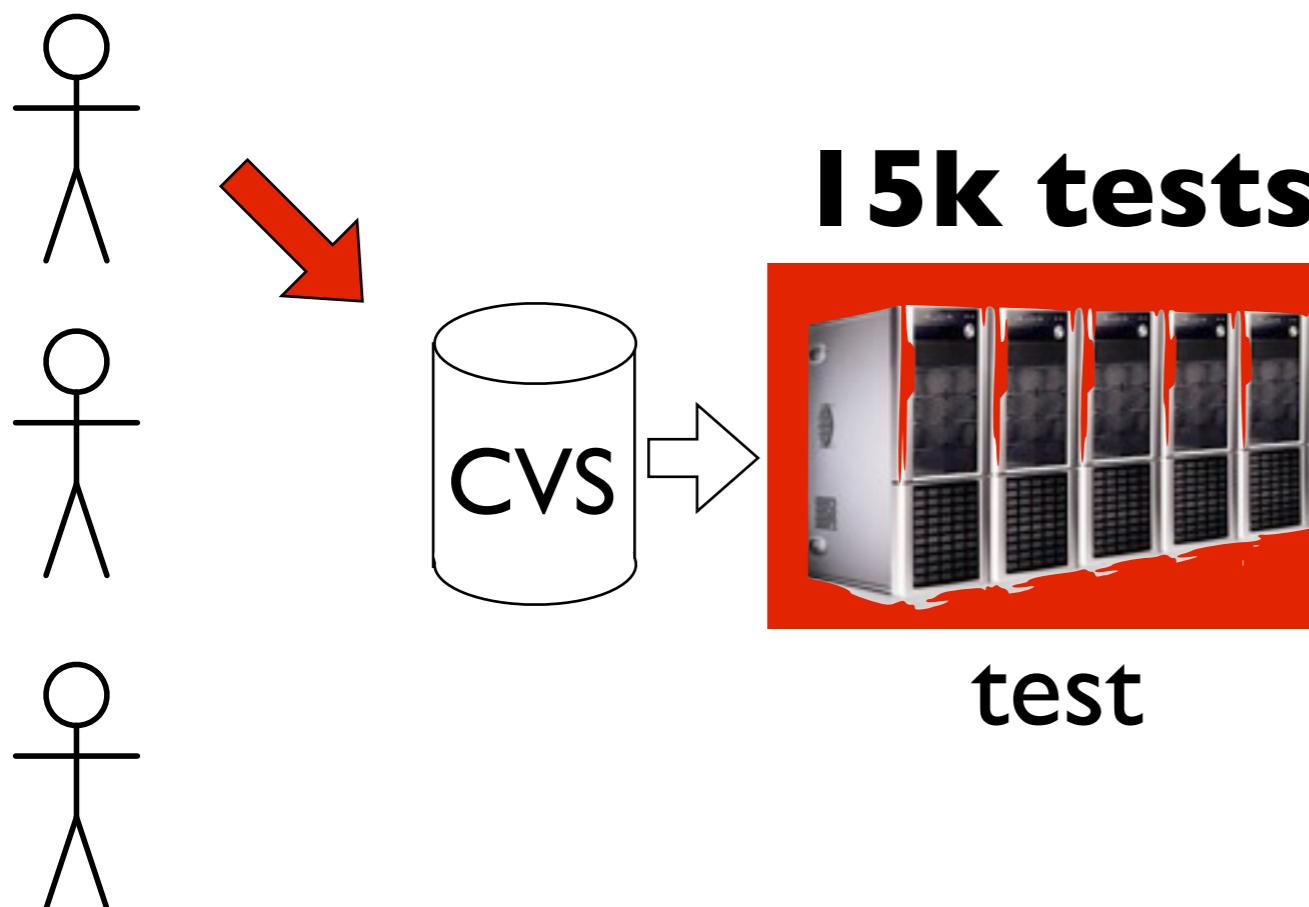
# Continuous Delivery



continuous  
integration



# Continuous Delivery

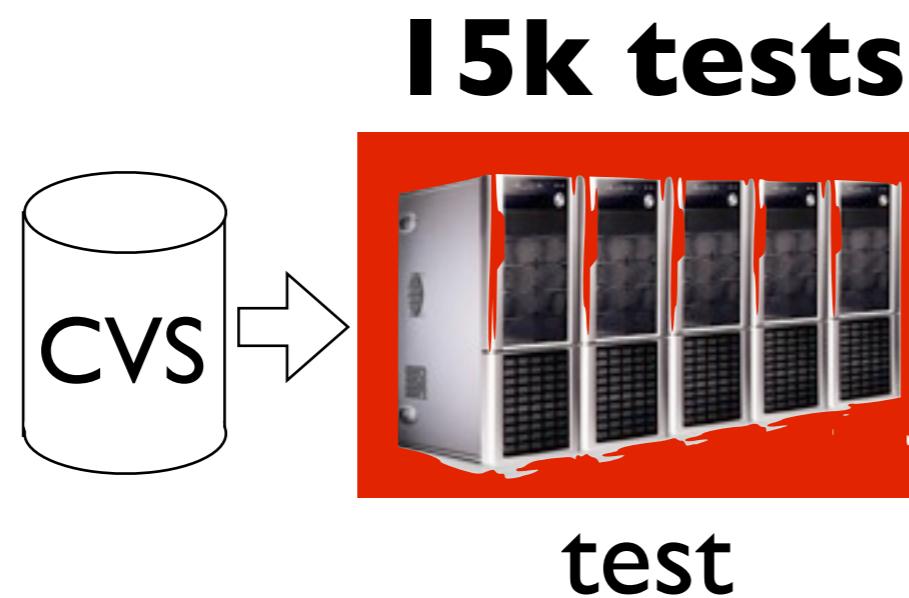
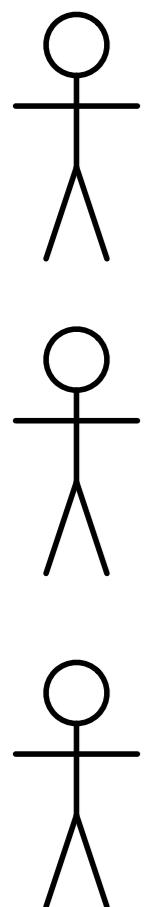


continuous  
integration

9 min.



# Continuous Delivery



continuous  
integration

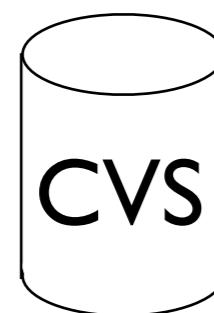
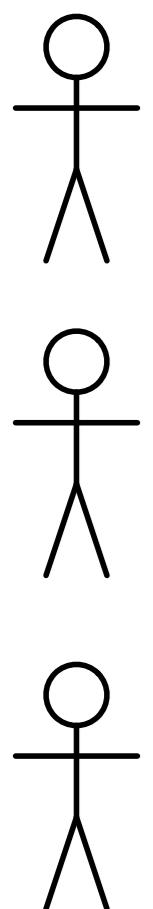
9 min.



staging/production



# Continuous Delivery



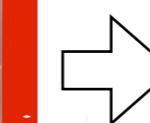
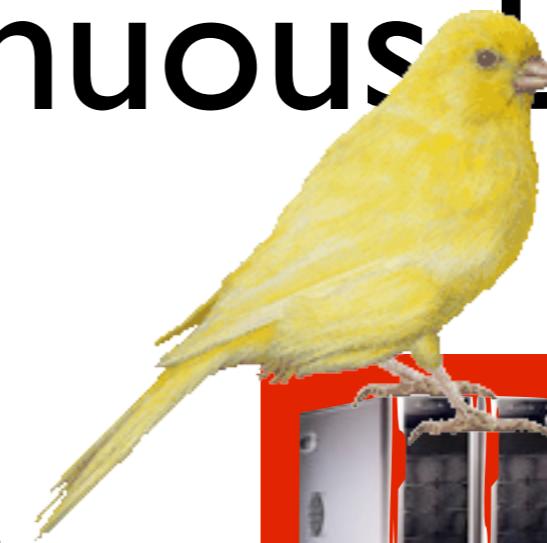
**15k tests**



**test**

**9 min.**

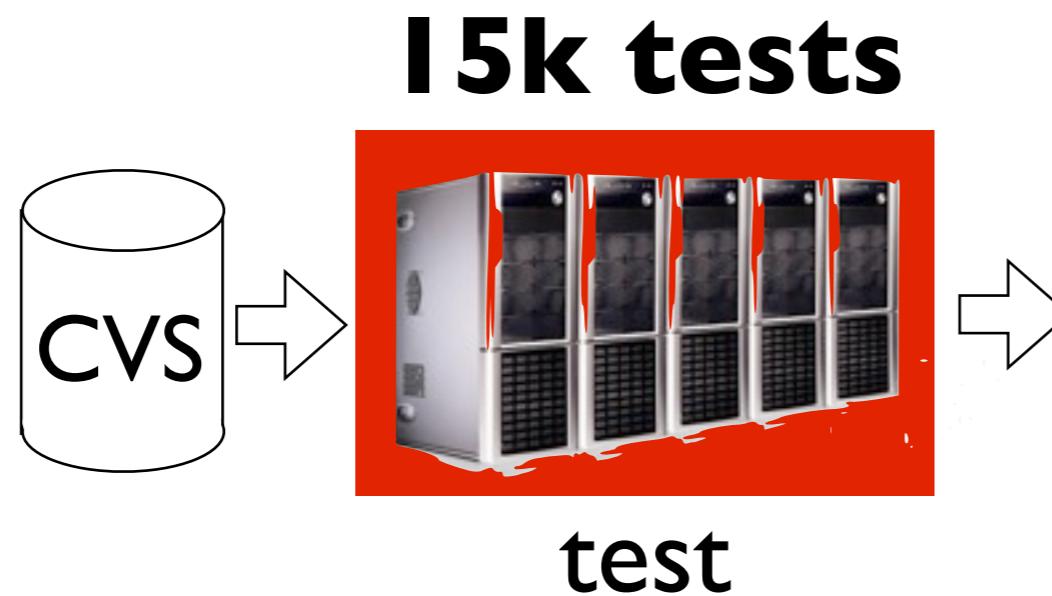
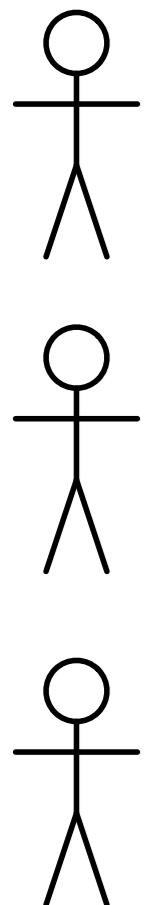
**continuous  
integration**



**staging/production**



# Continuous Delivery



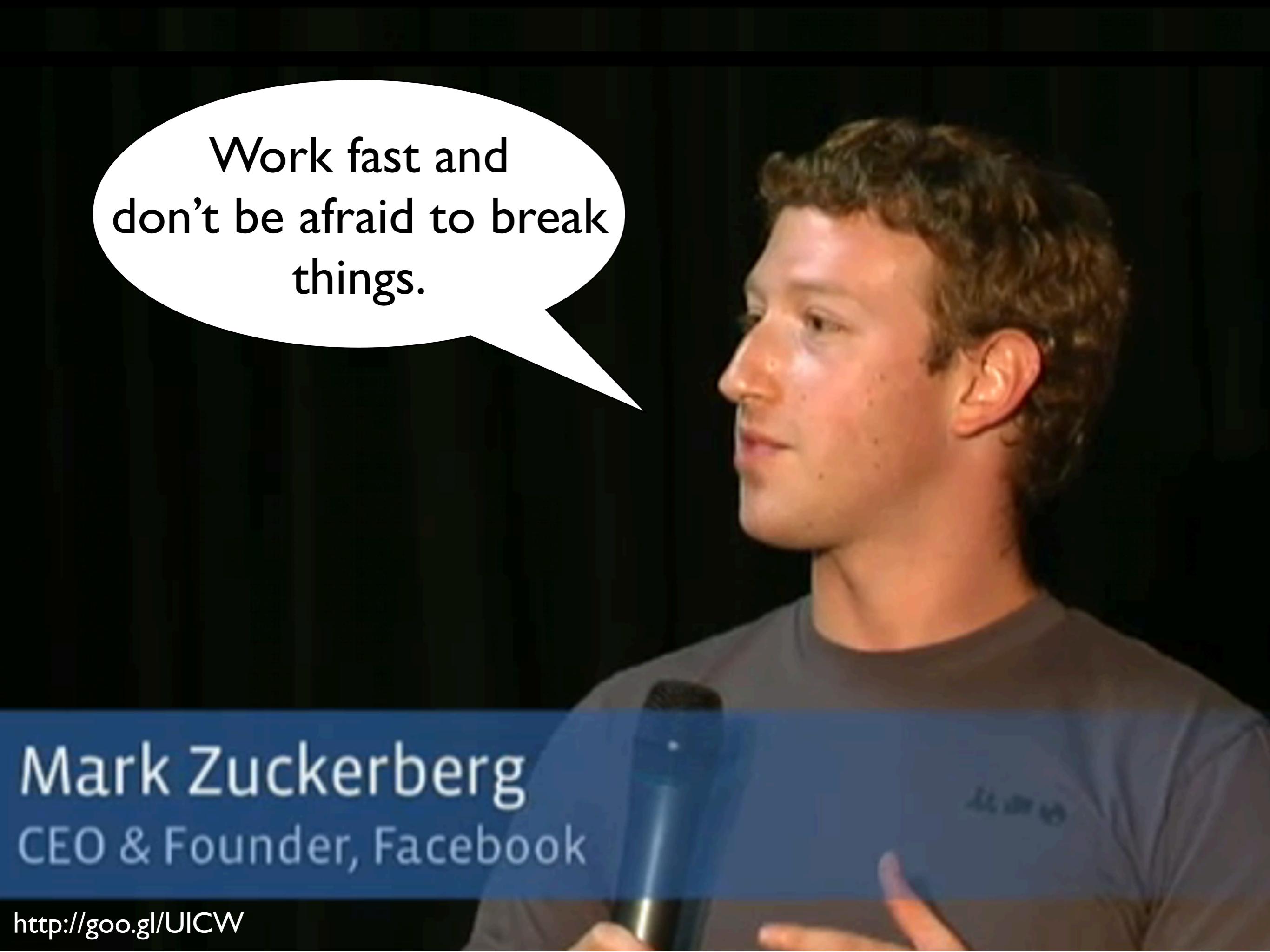
continuous  
integration

9 min.



staging/production

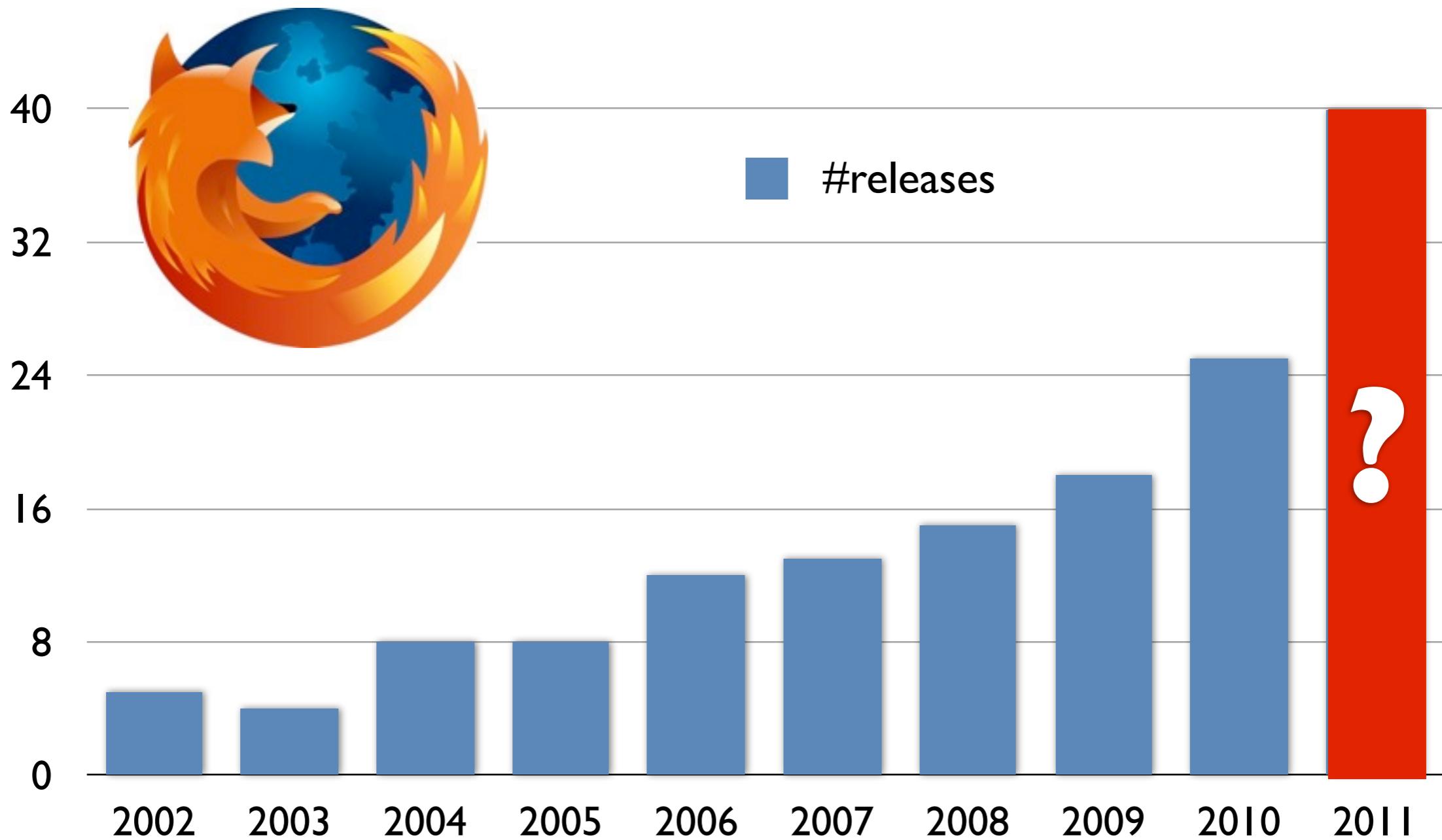
6 min.

A photograph of Mark Zuckerberg in profile, facing left. He is wearing a dark t-shirt and holding a microphone. A white speech bubble originates from his mouth, containing the text "Work fast and don't be afraid to break things.".

Work fast and  
don't be afraid to break  
things.

Mark Zuckerberg  
CEO & Founder, Facebook

# Even Desktop Apps Release More Frequently





... yet Software Systems  
keep on Growing!





# ... yet Software Systems keep on Growing!

>5k developers

build cache with  
>50TB memory

20 code  
changes/min.

<150k tests/  
commit

>2k projects

compilation  
in the cloud

>50k builds/day

>50M tests/day

# Release Engineering

# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



integration

VADO  
PERMANENTE



# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



integration



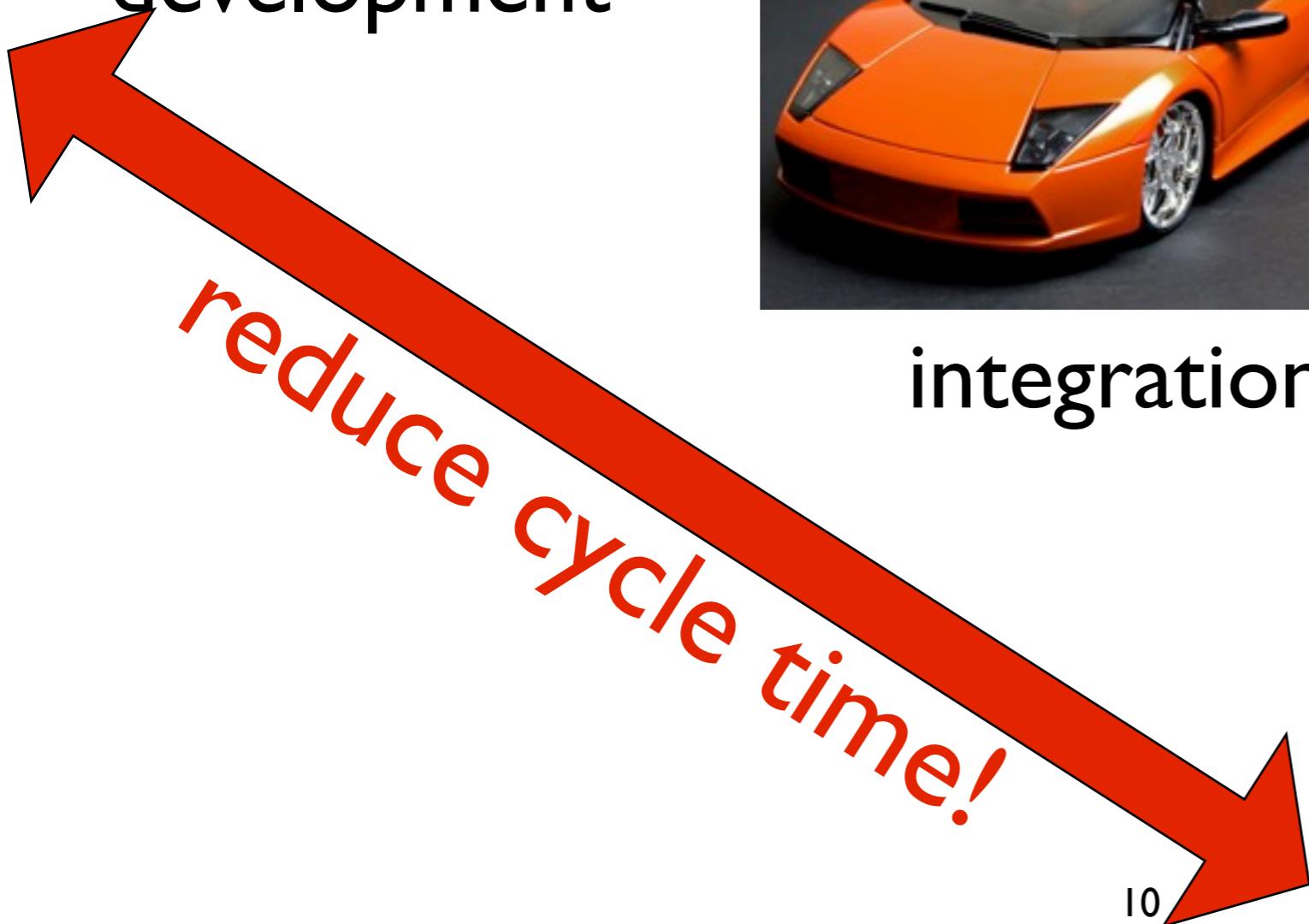
deployment

# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



integration



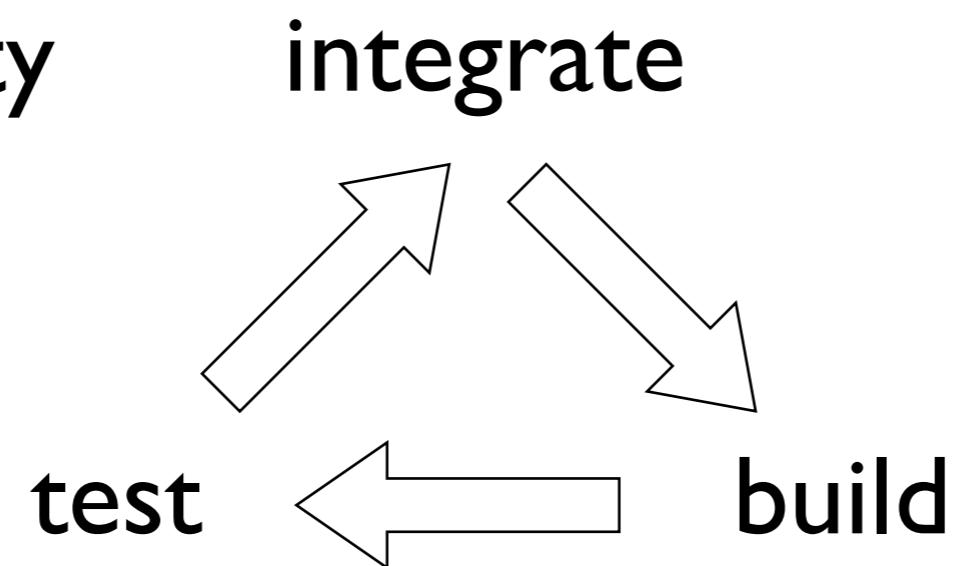
deployment

# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



10

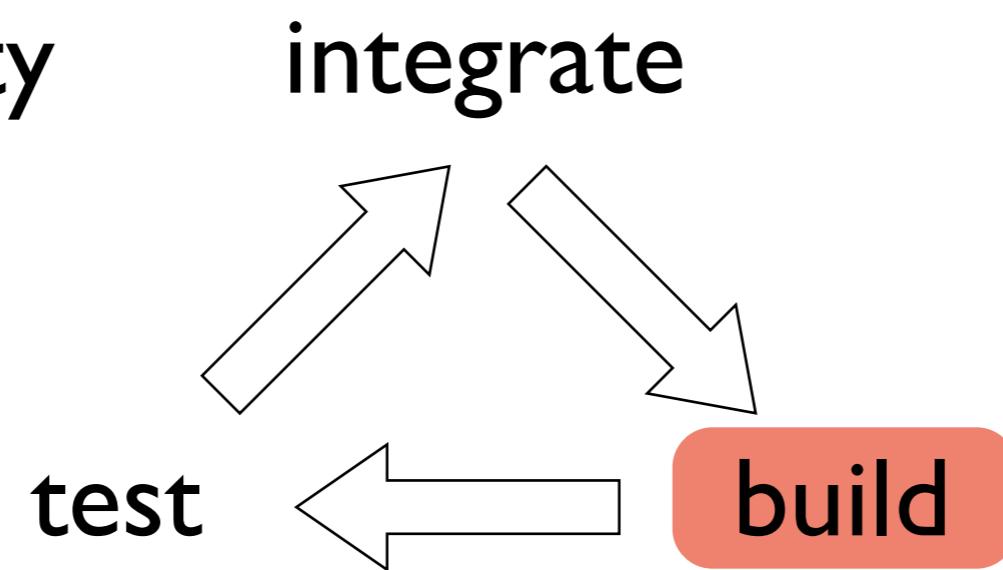
deployment



# Release Engineering



in-house/3rd party  
development



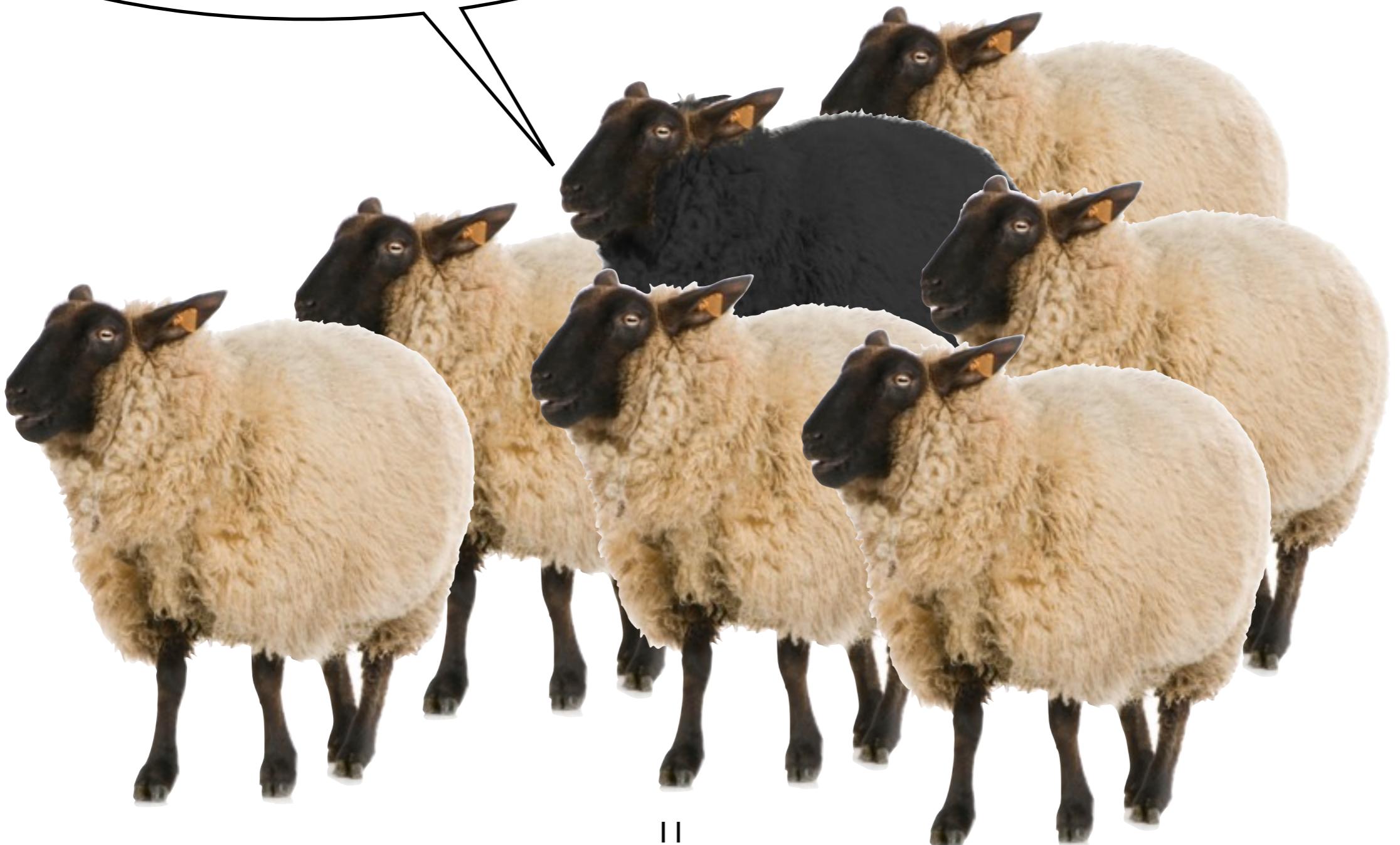
*reduce cycle time!*

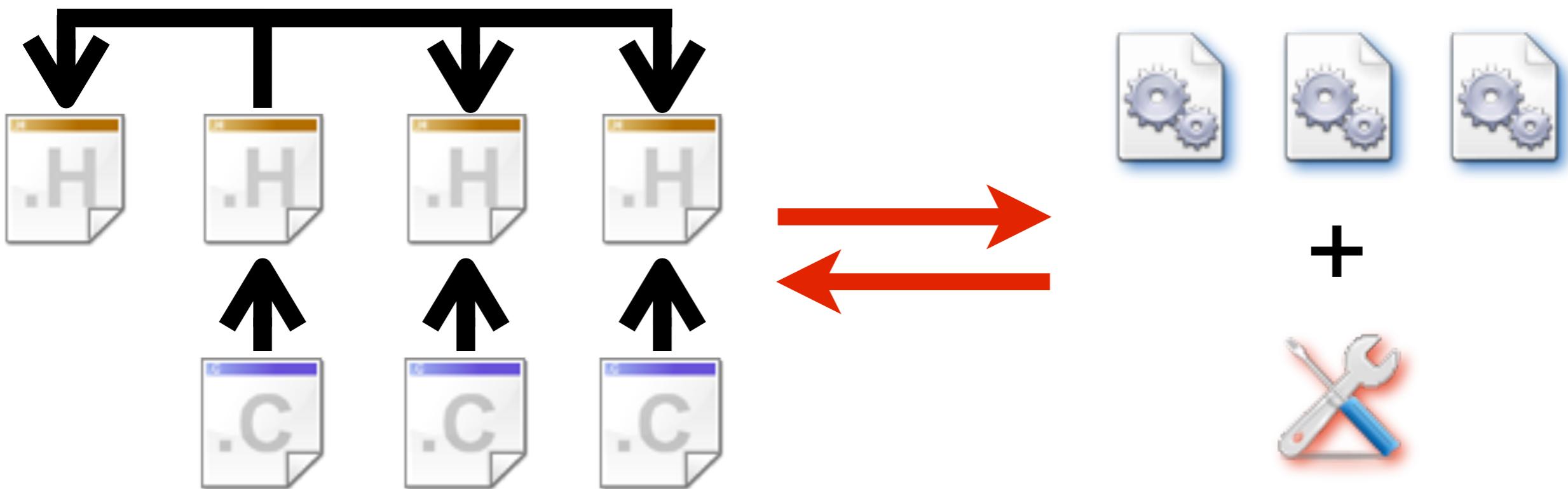
10

deployment



# The Build Process





# Source Code

# Build System



```
# KERNELRELEASE can change from a few different places, meaning version.h
# needs to be updated, so this check is forced on all builds

uts_len := 64
define filechk_utsrelease.h
    if [ `echo -n "$(KERNELRELEASE)" | wc -c ` -gt $(uts_len) ]; then \
        echo "'$(KERNELRELEASE)" exceeds $(uts_len) characters' >&2; \
        exit 1;
    fi;
    (echo \#define UTS_RELEASE \"$(KERNELRELEASE)\");
endif

define filechk_version.h
    (echo \#define LINUX_VERSION_CODE $(shell \
expr $(VERSION) \* 65536 + $(PATCHLEVEL) \* 256 + $(SUBLEVEL)); \
echo '#define KERNEL_VERSION(a,b,c) (((a) << 16) + ((b) << 8) + (c))');
endif

include/linux/version.h: $(srctree)/Makefile FORCE
    $(call filechk,version.h)

include/generated/utsrelease.h: include/config/kernel.release FORCE
    $(call filechk,utsrelease.h)

PHONY += headerdep
headerdep:
    $(Q)find include/ -name '*.h' | xargs --max-args 1 scripts/headerdep.pl
```



@deanberris

Dean Michael Berris

I hate GNU Make especially those Makefiles that are hand-crafted and undocumented. It all feels like a bad hack.

24 Feb via web Favorite Retweet Reply



KDE 4 is leaving the aging "autotool" build chain behind. Some developers, not only in KDE, like to **nickname the autotools as "auto-hell"** because of its difficulty to comprehend architecture.  
[<http://lwn.net/Articles/188693/>]

*Our record so far is a project we inherited with an Ant script weighing in at 10,000 lines of XML. Needless to say, this project required an entire team devoted to keeping the build working—a complete waste of resources.*  
[Jez Humble & David Farley]

Build Systems  
are Complex



[ICSM '07, ECEASST '08]

>5 MLOC & ~20 Years of History





# Quake 3

server

game logic

- o
- a
- h
- c
- dylib

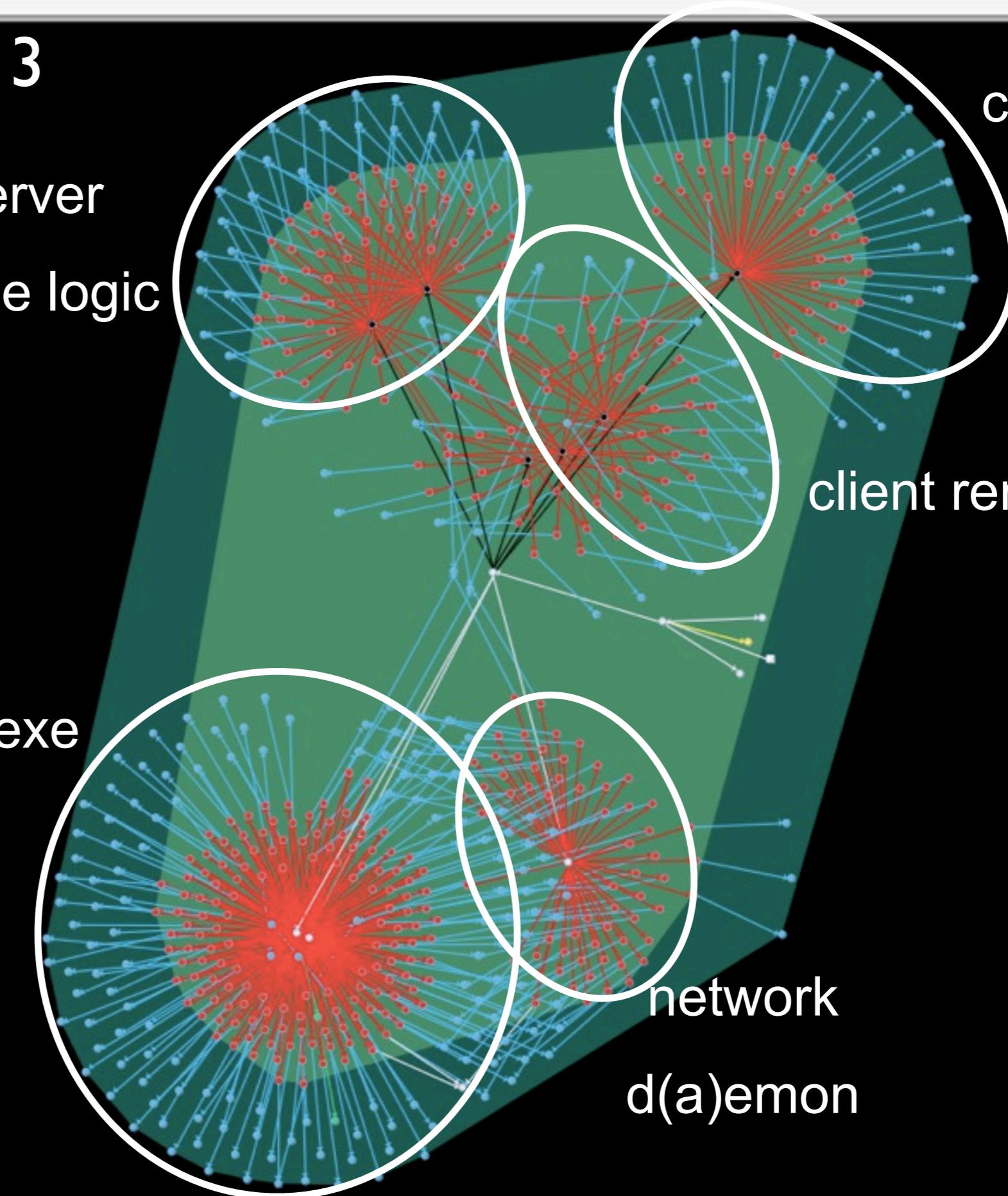
quake3.exe

network

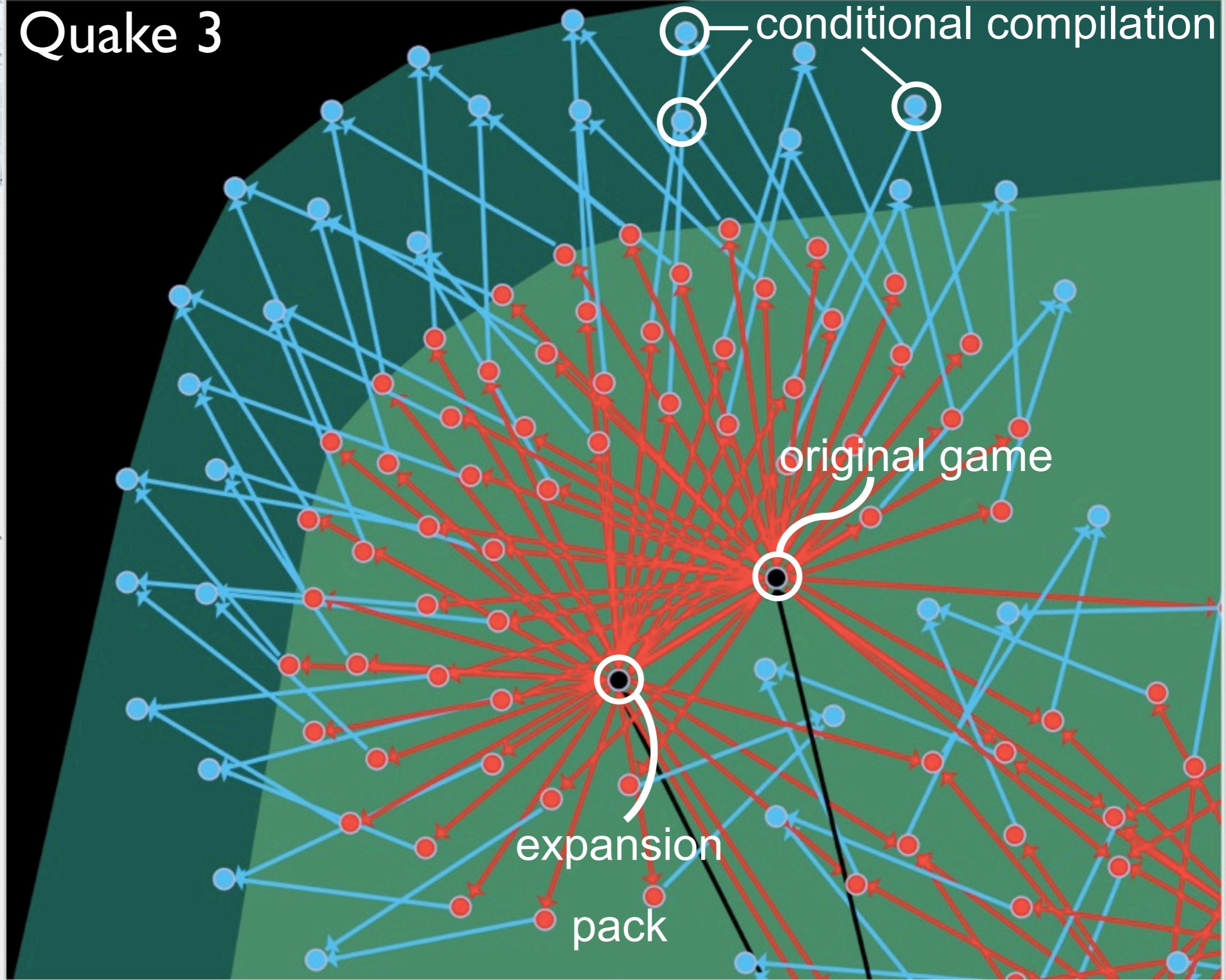
d(a)emon

client UI

client renderer

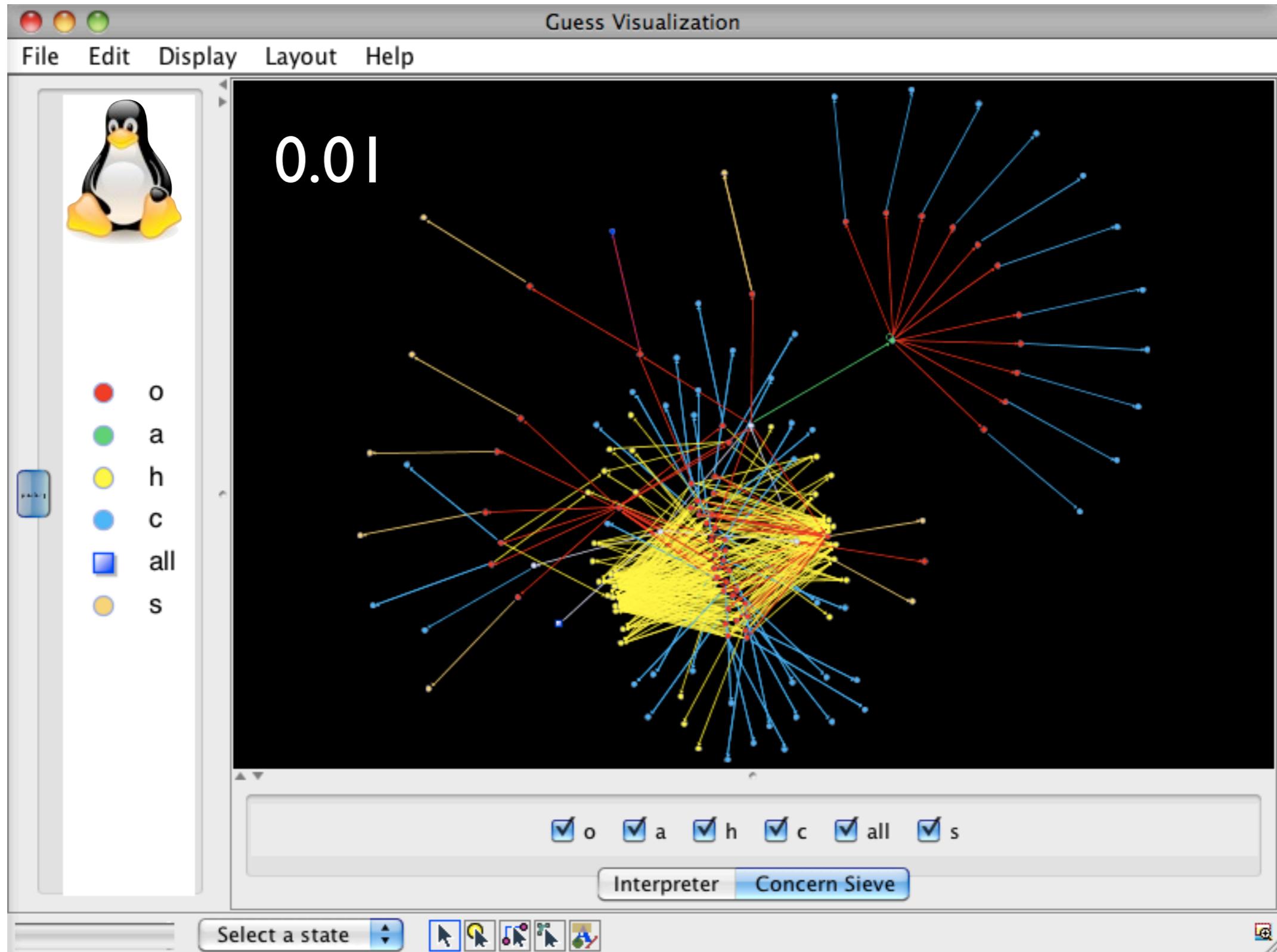


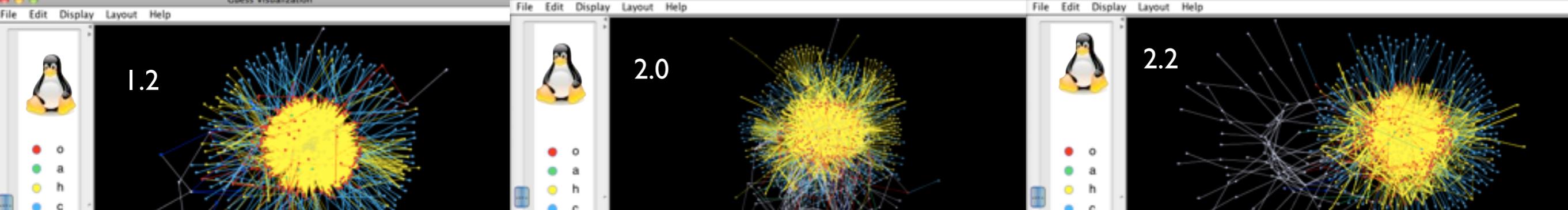
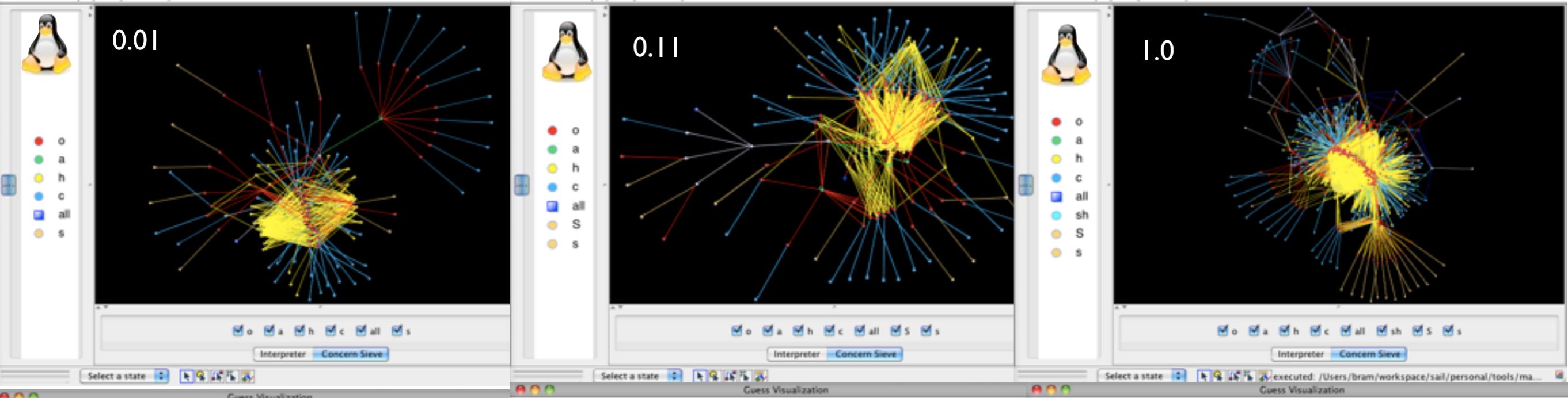
## Quake 3



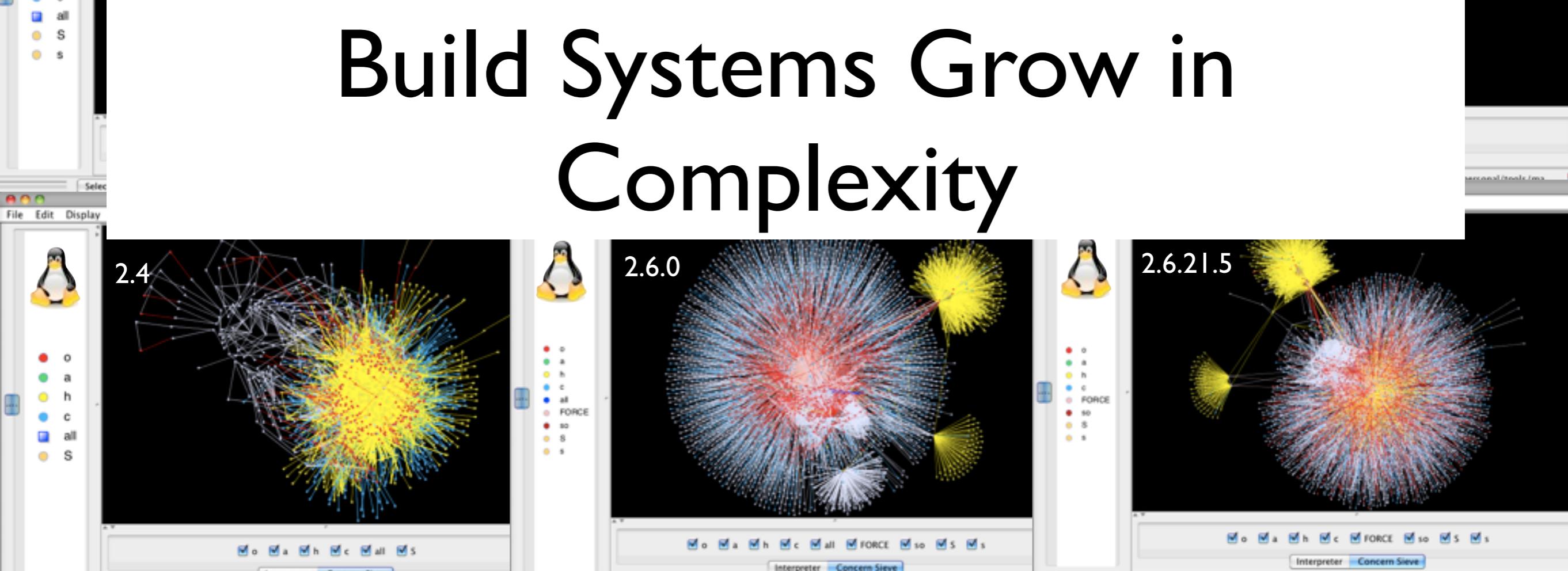


# [ICSM '07,ECEASST '08]





# Build Systems Grow in Complexity





**Build Systems Require 12%  
of a Developer's  
Time (on average)**

**Build maintenance  
slows development!**

Kumfert, G., and Epperly, T.  
*Software in the DOE: The  
Hidden Overhead of the “Build”*



[MSR '10, ICSE '11]

# >36 MLOC & ~120 Years of History

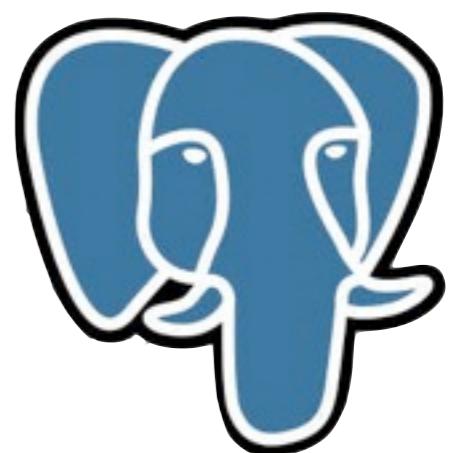


Apache  
Tomcat

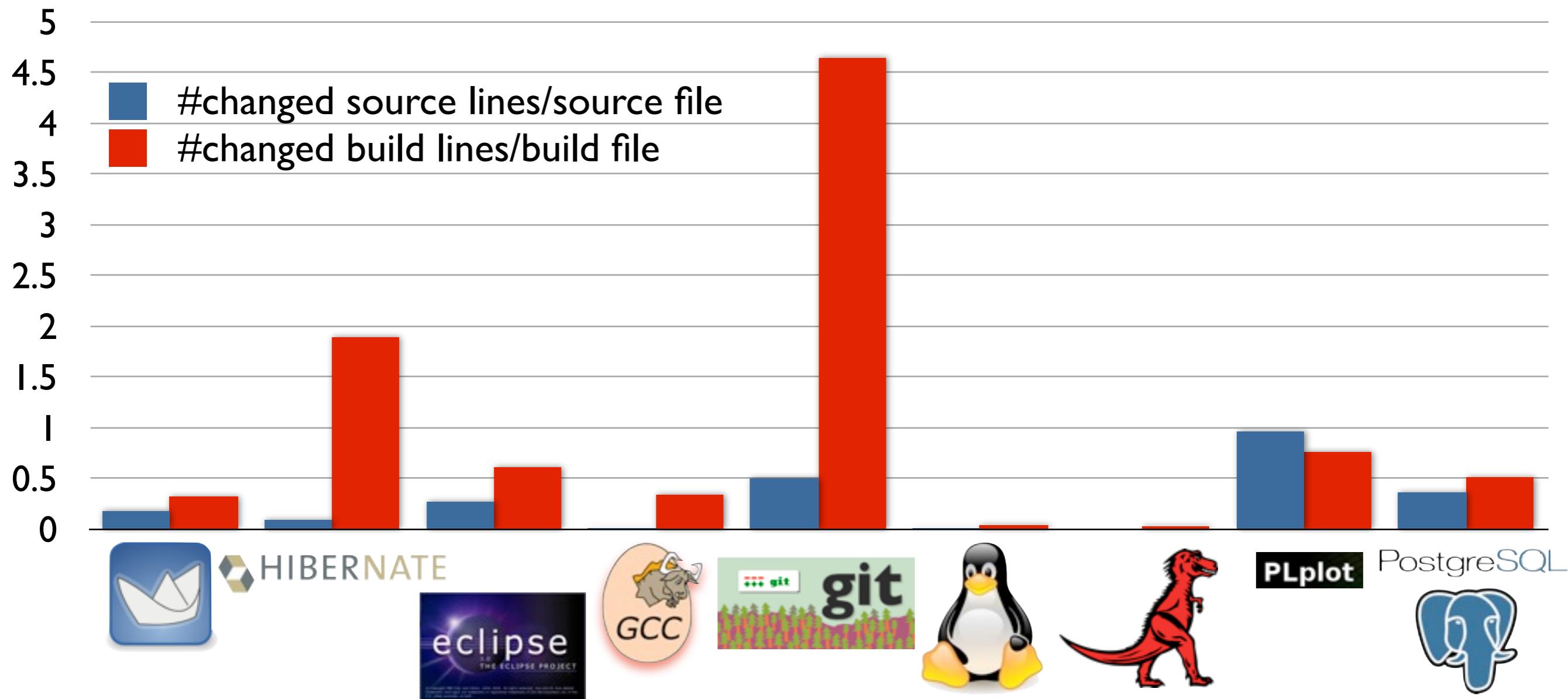
jazz  
PostgreSQL



HIBERNATE



# Build Files Change Relatively More than Source Code Files



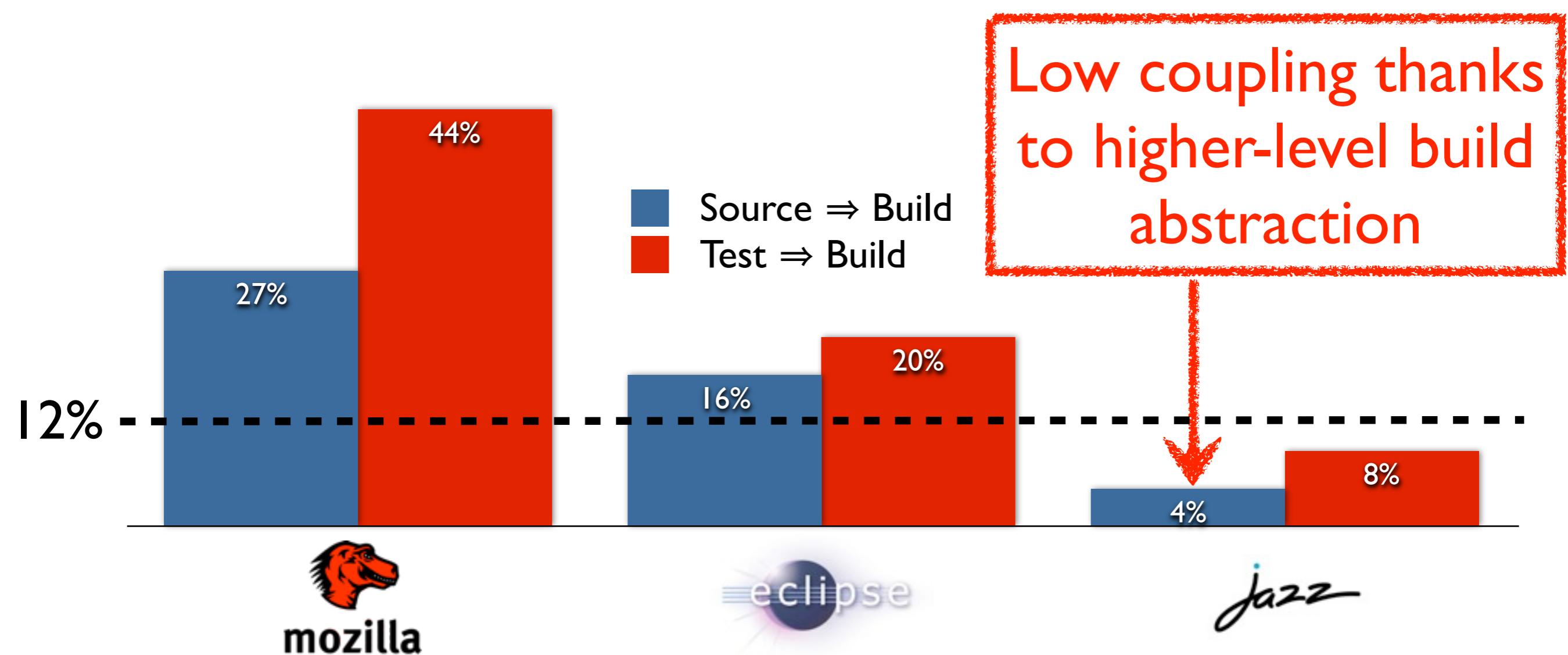
# The Build System Requires Significant Maintenance

- Source ⇒ Build
- Test ⇒ Build

12% -----



# The Build System Requires Significant Maintenance



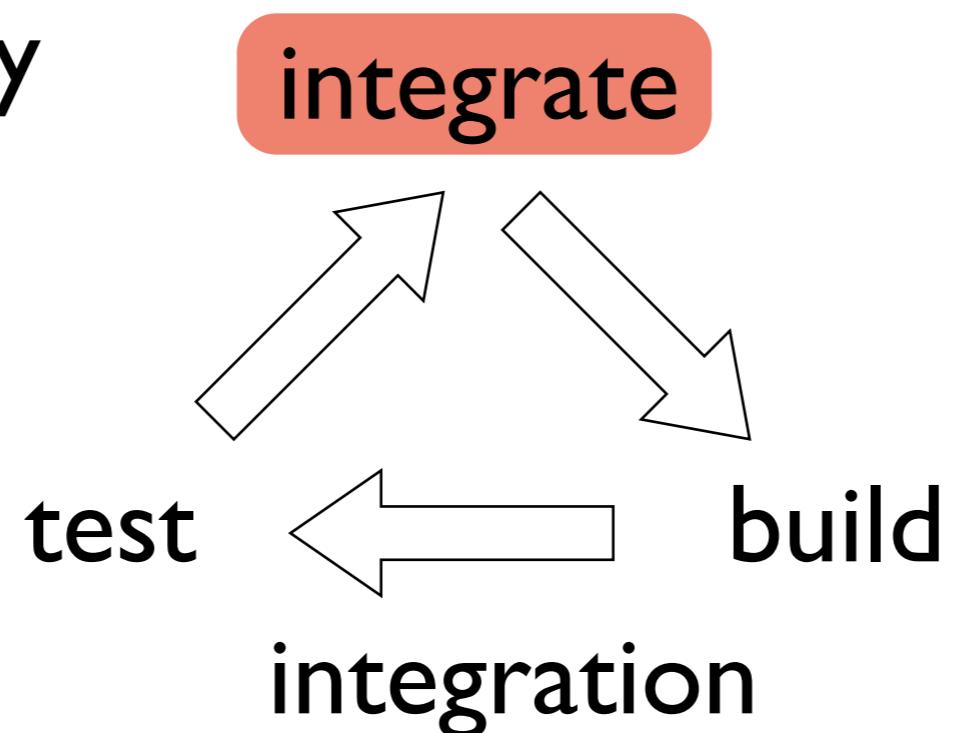


Our Build Systems need HELP

# Release Engineering



in-house/3rd party  
development



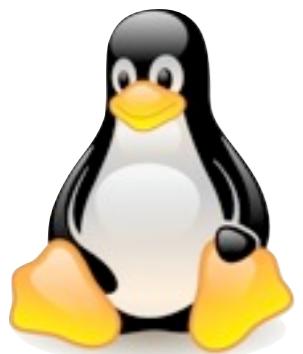
*reduce cycle time!*



deployment

# The Integration Process



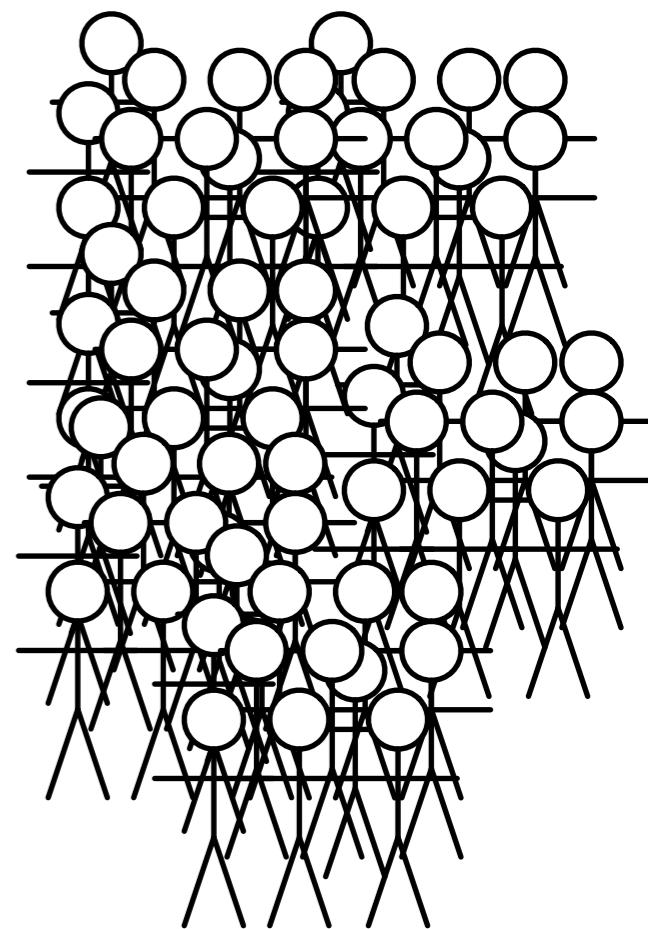


# Integrators are Gatekeepers

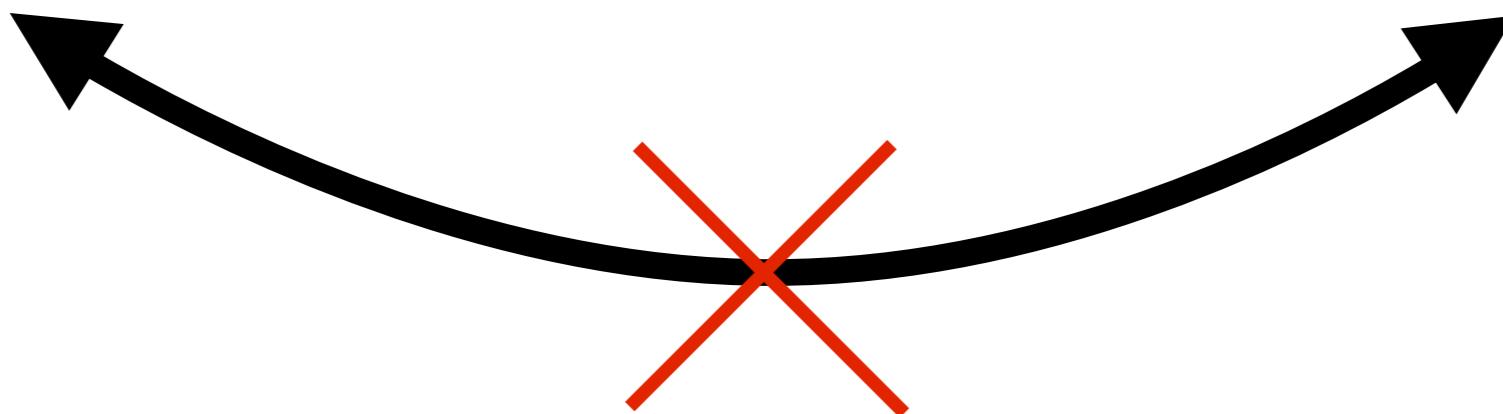
OpenOffice.org

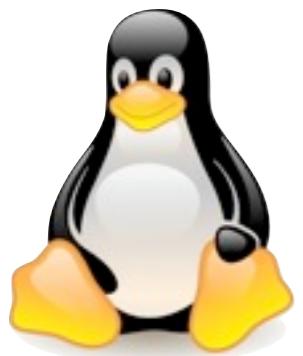


MySQL®



end users



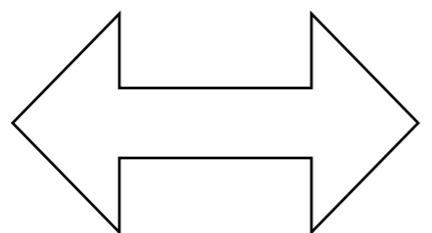


# Integrators are Gatekeepers

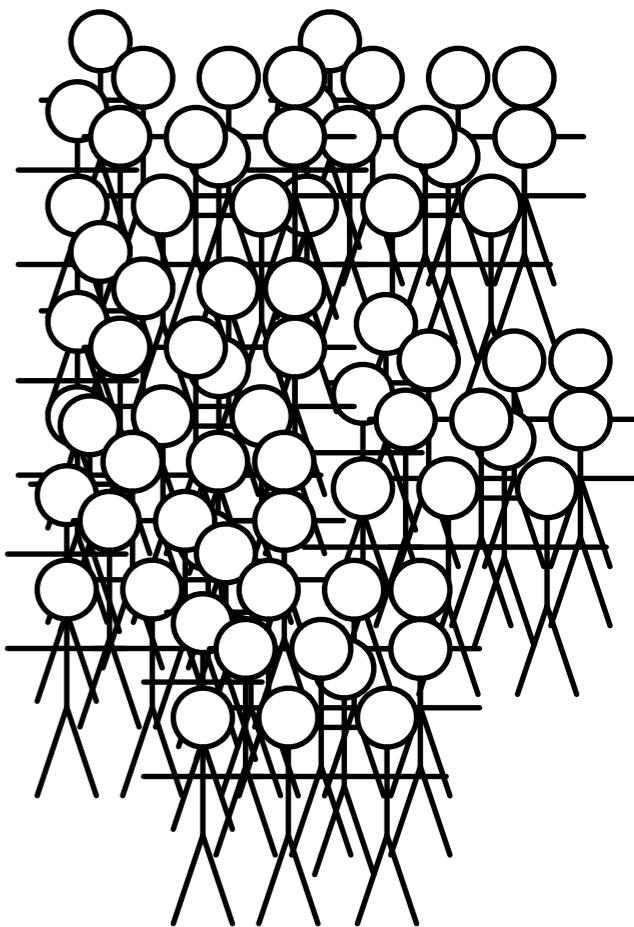
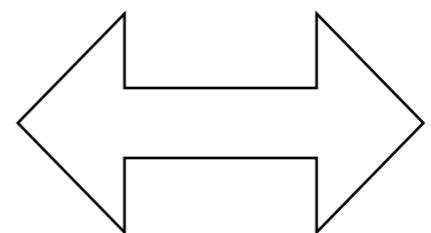
OpenOffice.org



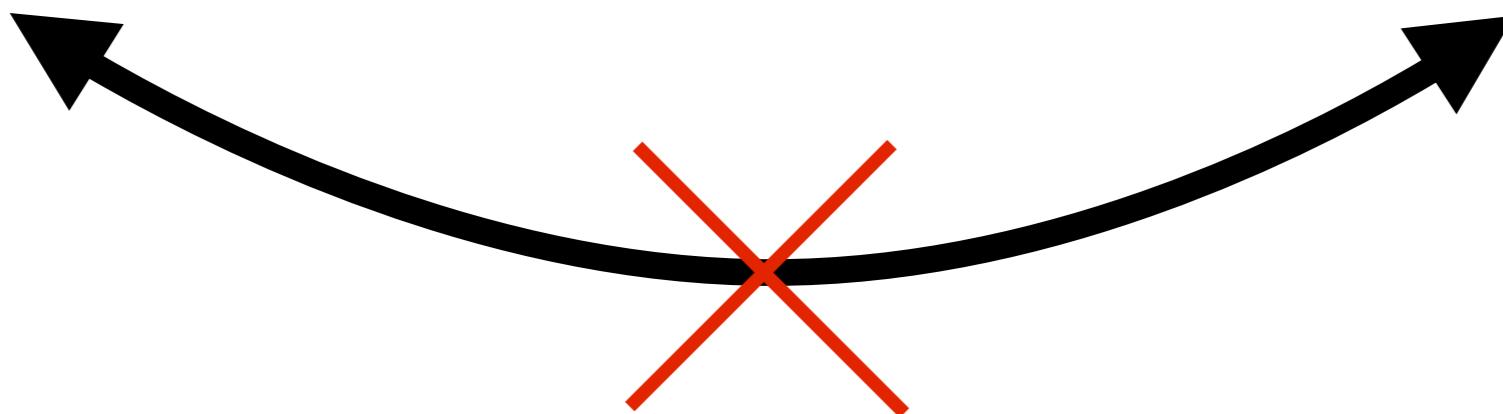
MySQL®



maintainer



end users



>180k Packages & ~28 Years of History



debian



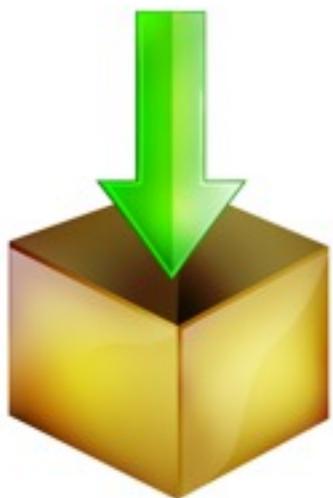
freeBSD®



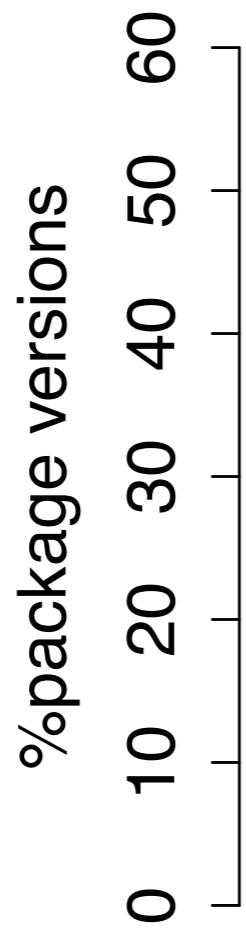
ubuntu



New Package



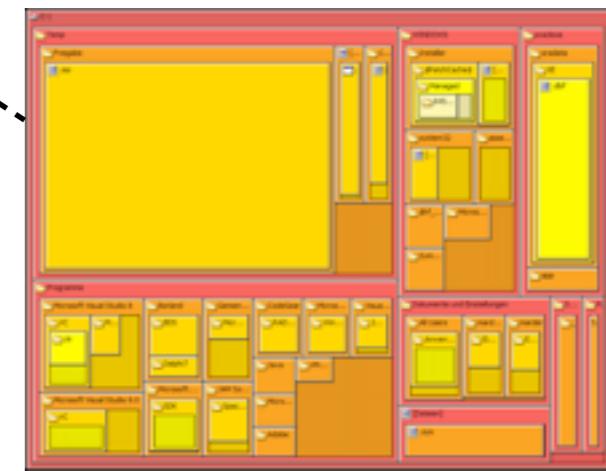
Upstream Sync



Dependency  
Management



Build Change

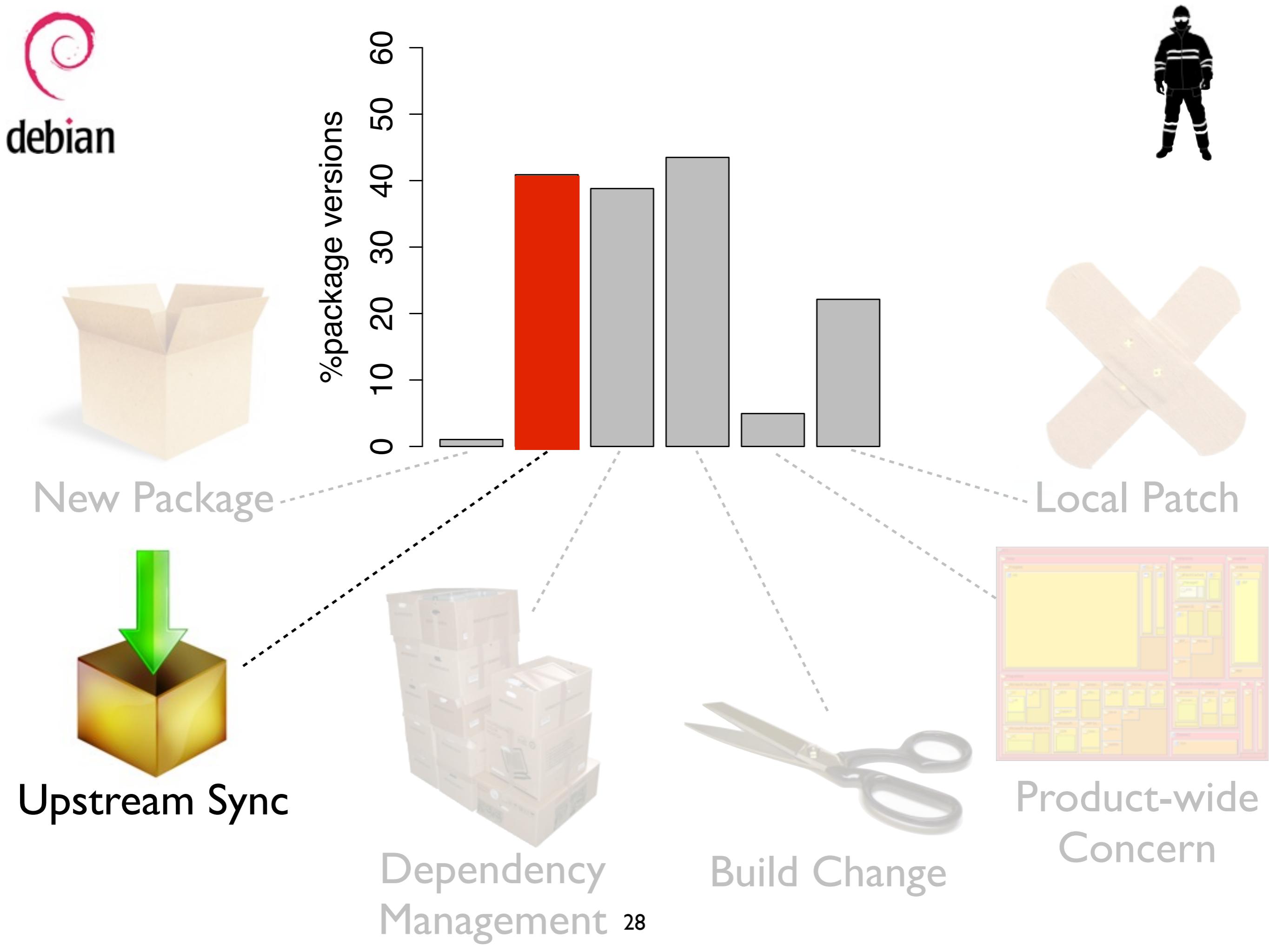


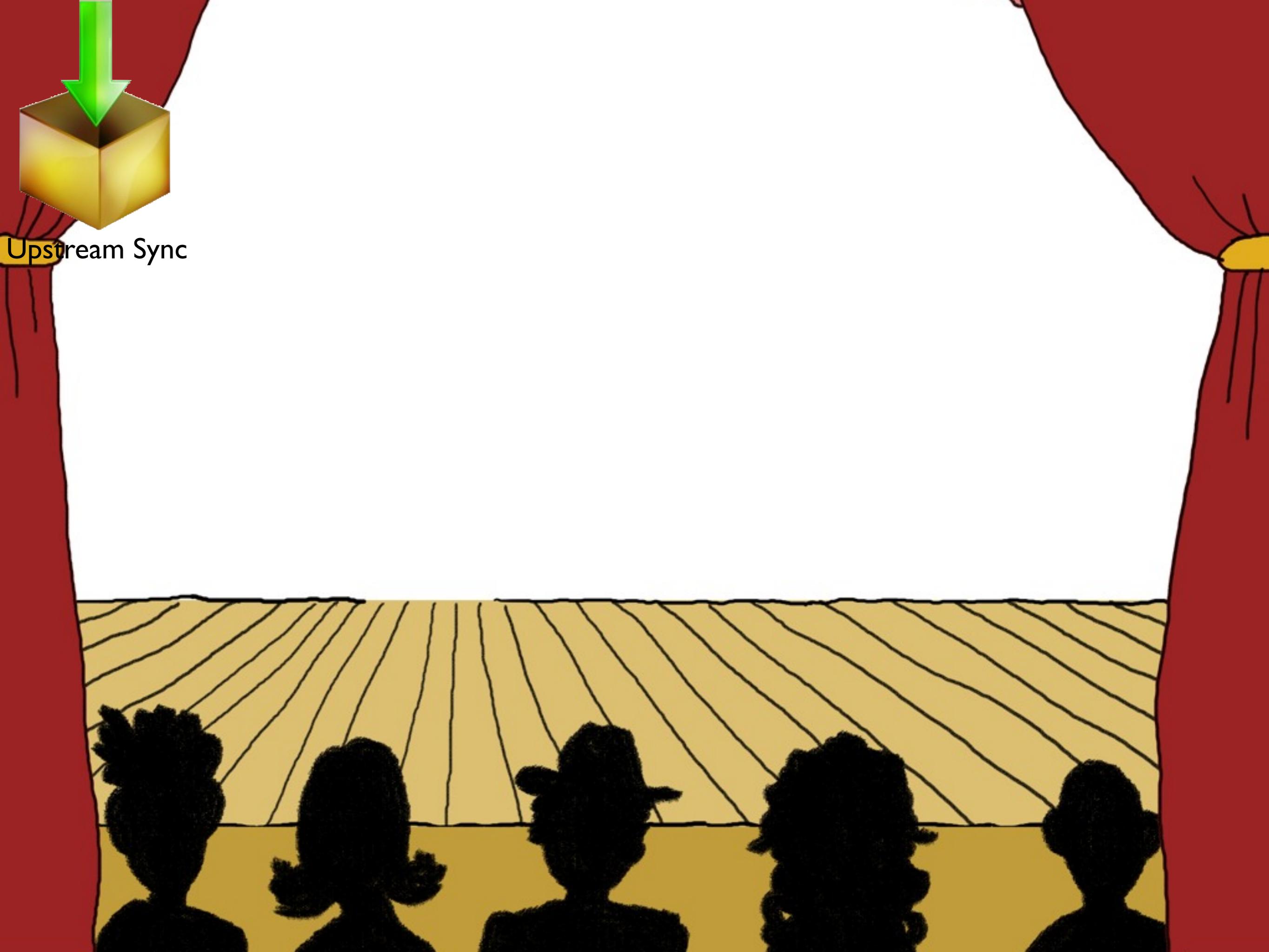
Product-wide  
Concern



Local Patch







Upstream Sync



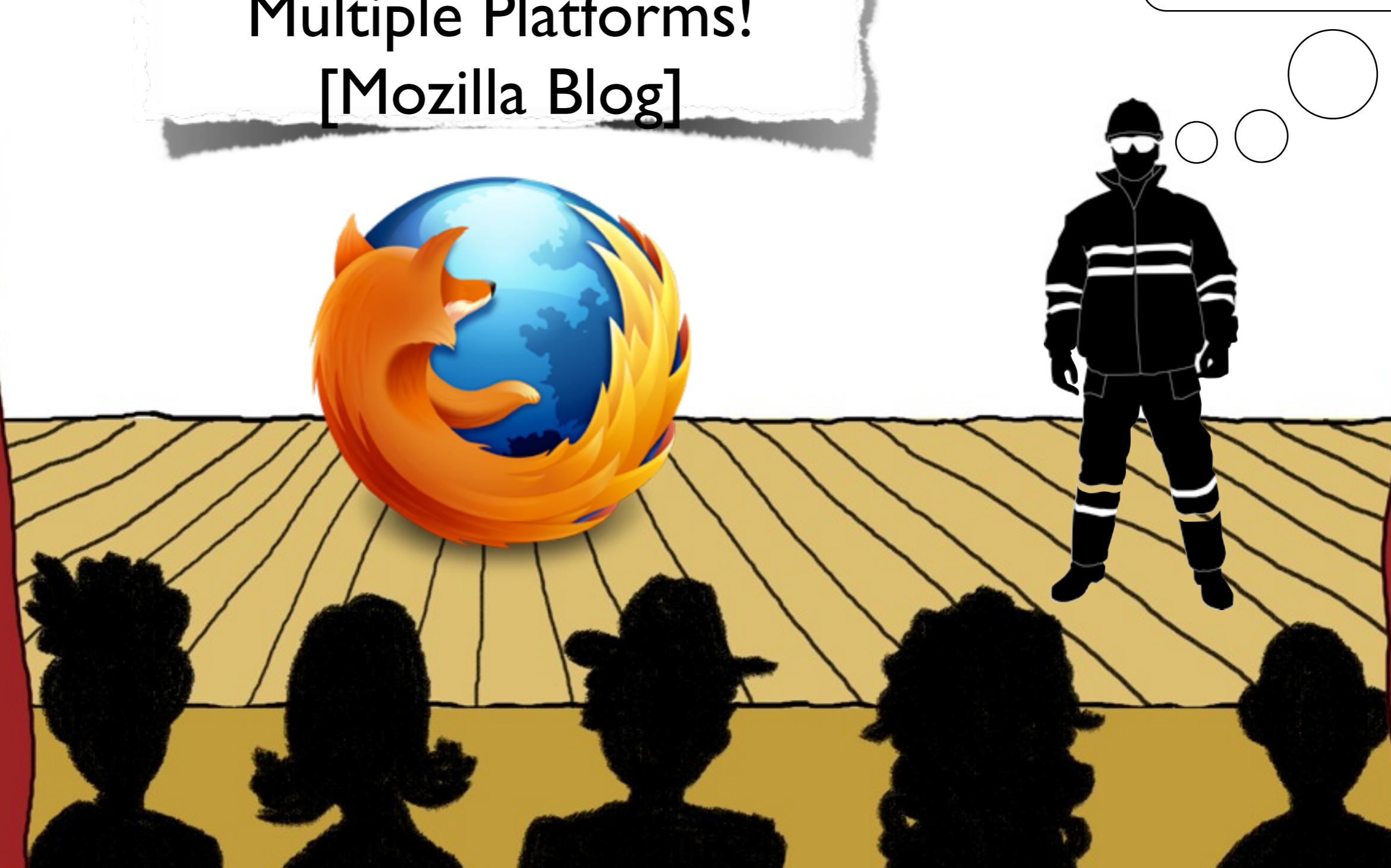
Mozilla Delivers **New Version** of Firefox – First Web Browser to Support Do Not Track on Multiple Platforms!

[Mozilla Blog]

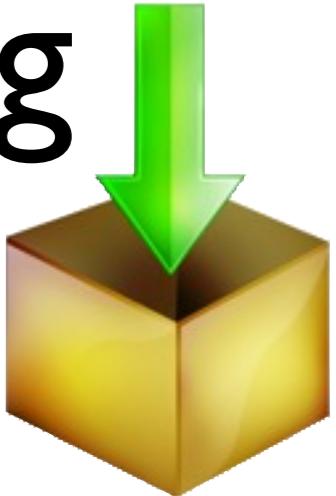
Upstream Sync



Necessary?  
Buggy?  
Secure?



# Risk Analysis & Cherry-picking



Upstream Sync



```
/* allocate 'order1' elements
 * 'order1' is used to store
 * 'order2' is used to store
 */
order1 = order>>4;
order2 = order & 15;

#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: Can't
    exit(2);
}
#endif
/* allocate 'order1' elements
 * 'order1' is used to store
 * 'order2' is used to store
 */
order1 = order>>4;
order2 = order & 15;

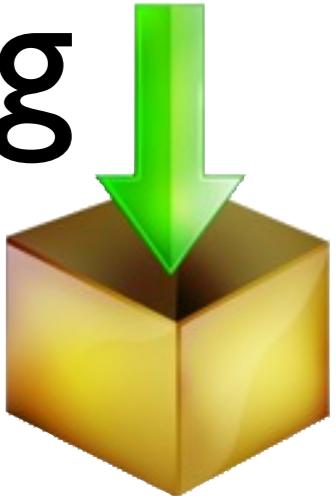
#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: Can't
    exit(2);
}
#endif
/* allocate 'order1' elements
 * 'order1' is used to store
 * 'order2' is used to store
 */
order1 = order>>4;
order2 = order & 15;
```



```
/* allocate 'order1' elements
 * 'order1' is used to store
 * 'order2' is used to store
 */
order1 = order>>4;
order2 = order & 15;

#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: Can't
    exit(2);
}
#endif
/* allocate 'order1' elements
 * 'order1' is used to store
 * 'order2' is used to store
 */
order1 = order>>4;
order2 = order & 15;
```

# Risk Analysis & Cherry-picking



Upstream Sync



```
/* allocate 'order+1' elements
 * 'order+1' is used to store the last element
 */
order1 = order>>4;
order2 = order & 15;

#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: can't open file\n");
    exit(2);
#endif
/* allocate 'order+1' elements
 * 'order+1' is used to store the last element
 */
order1 = order>>4;
order2 = order & 15;

#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: can't open file\n");
    exit(2);
#endif
/* allocate 'order+1' elements
 * 'order+1' is used to store the last element
 */
order1 = order>>4;
order2 = order & 15;
```

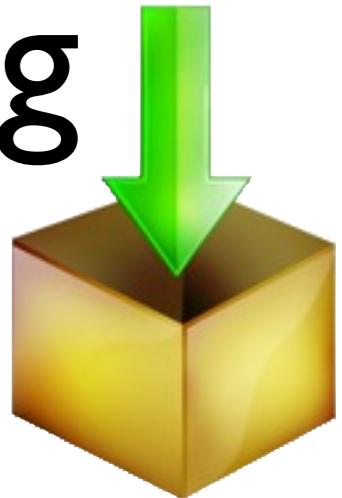
## UNIX diff



```
diff -u oldfile newfile
--- oldfile 2010-05-11 11:45:40.000000000 +0000
+++ newfile 2010-05-11 11:45:40.000000000 +0000
@@ -1,10 +1,10 @@
 /* allocate 'order+1' elements
  * 'order+1' is used to store the last element
  */
order1 = order>>4;
order2 = order & 15;

#ifndef TRACE
if (!fpm = fopen("ppmenc.doc", "wb"))
{
    fprintf(stderr, "An Error: can't open file\n");
    exit(2);
#endif
/* allocate 'order+1' elements
  * 'order+1' is used to store the last element
  */
order1 = order>>4;
order2 = order & 15;
```

# Risk Analysis & Cherry-picking



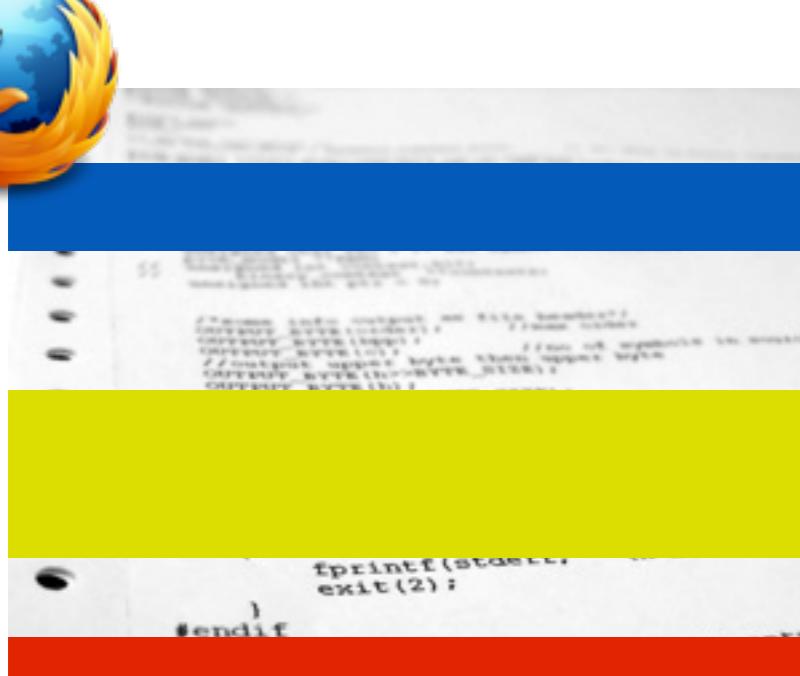
# Upstream Sync



# UNIX diff



# maintainer



# Risk Analysis & Cherry-picking



```
/* allocate 'order+1' elements
 * 'order+1' is used to store
 * the last element
 */
#endif

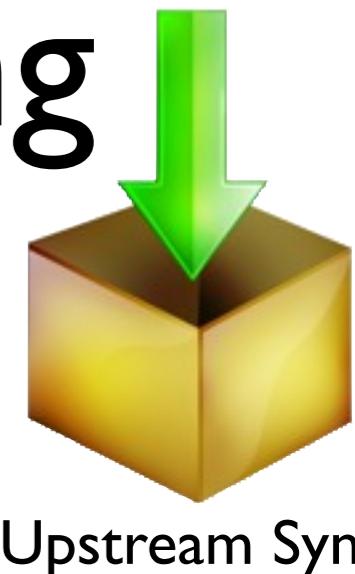
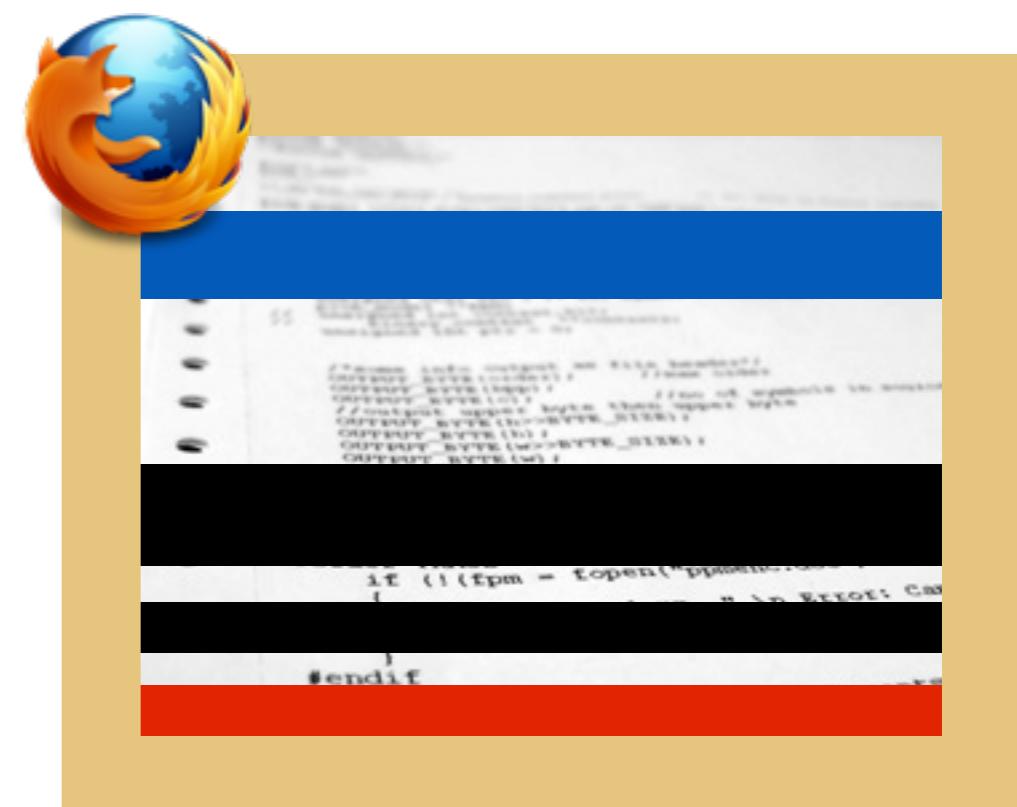
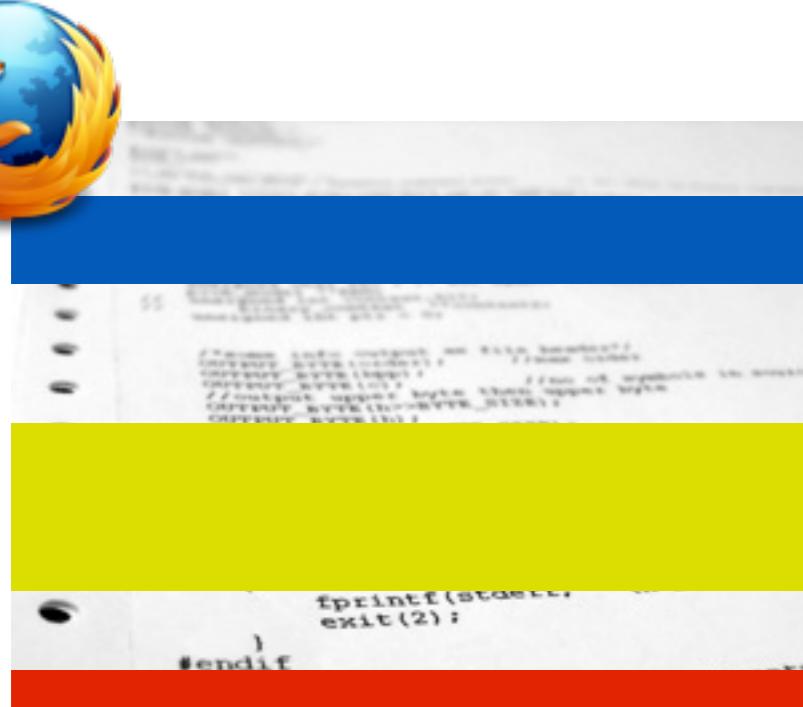
#endif /* !defined(TRACE) */

/* If we have reached here, then we have
 * successfully read all the data from
 * the file. Now we need to write it
 * back to the output file.
 */
if (fpm != NULL)
{
    if (fwrite(ppmemc, sizeof(ppmemc), 1, fpm) != 1)
        fprintf(stderr, "In Error: can't
write to file\n");
    else
        printf("Written %d bytes to file\n",
            sizeof(ppmemc));
}
```

# UNIX diff



# maintainer

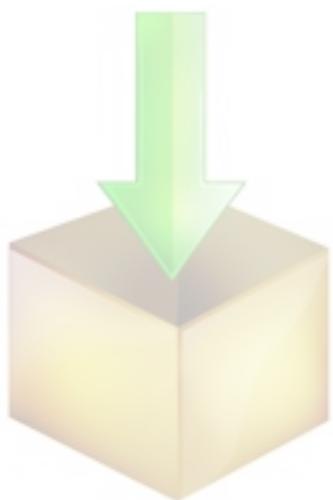




debian



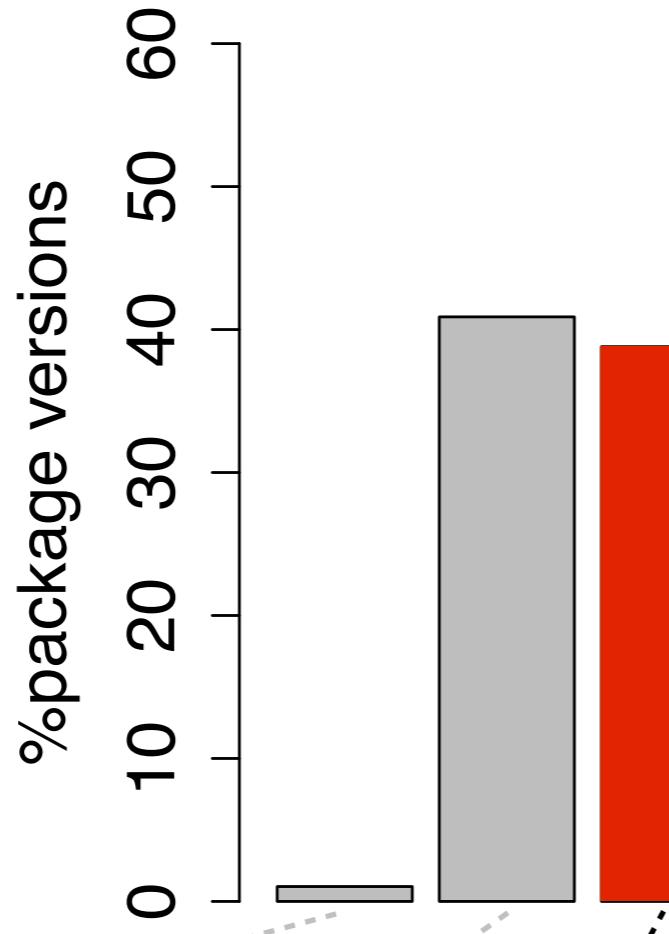
New Package



Upstream Sync

## Dependency Management

31



Build Change



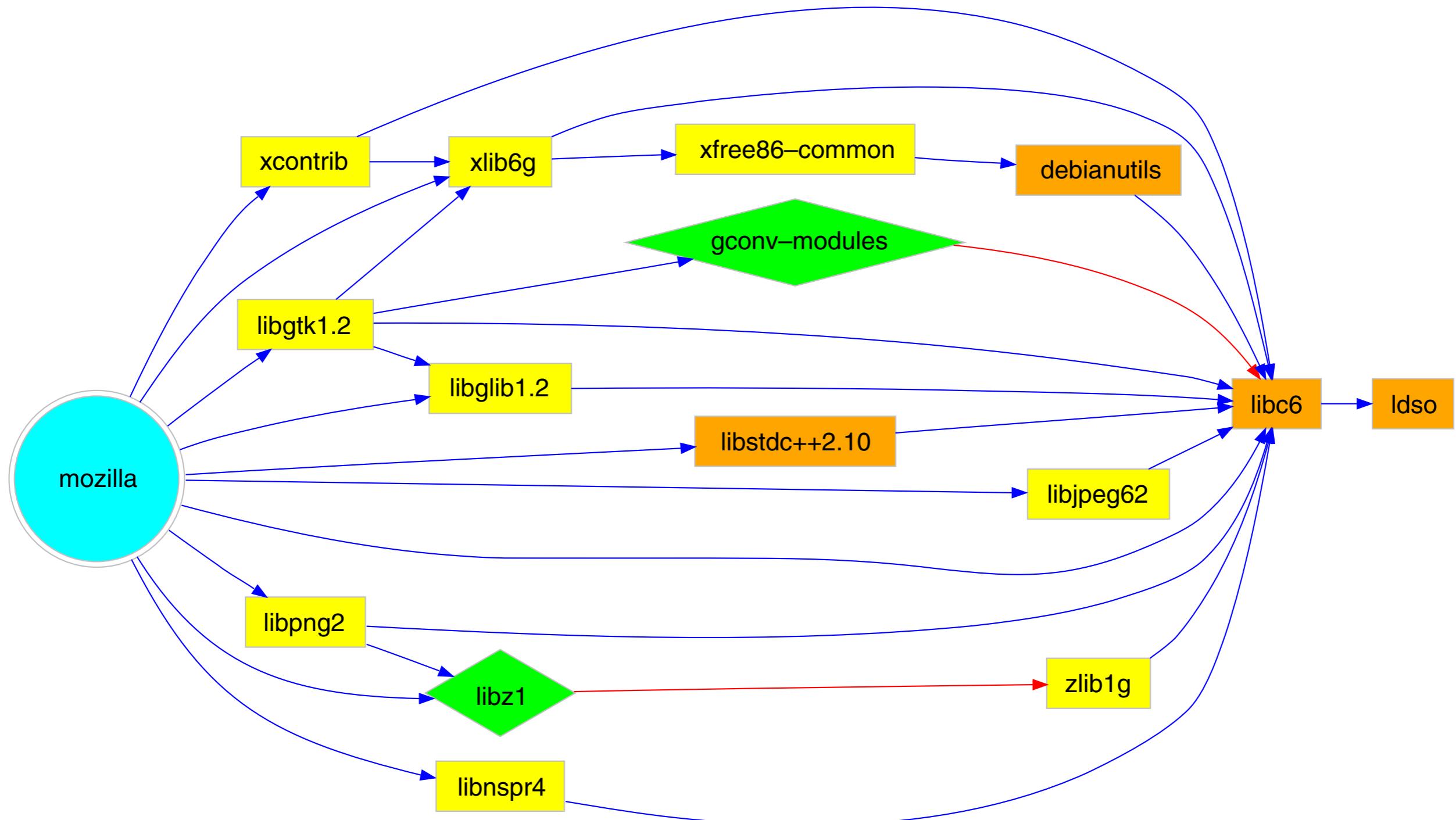
Product-wide  
Concern



# Library Transitions Ripple through the whole System



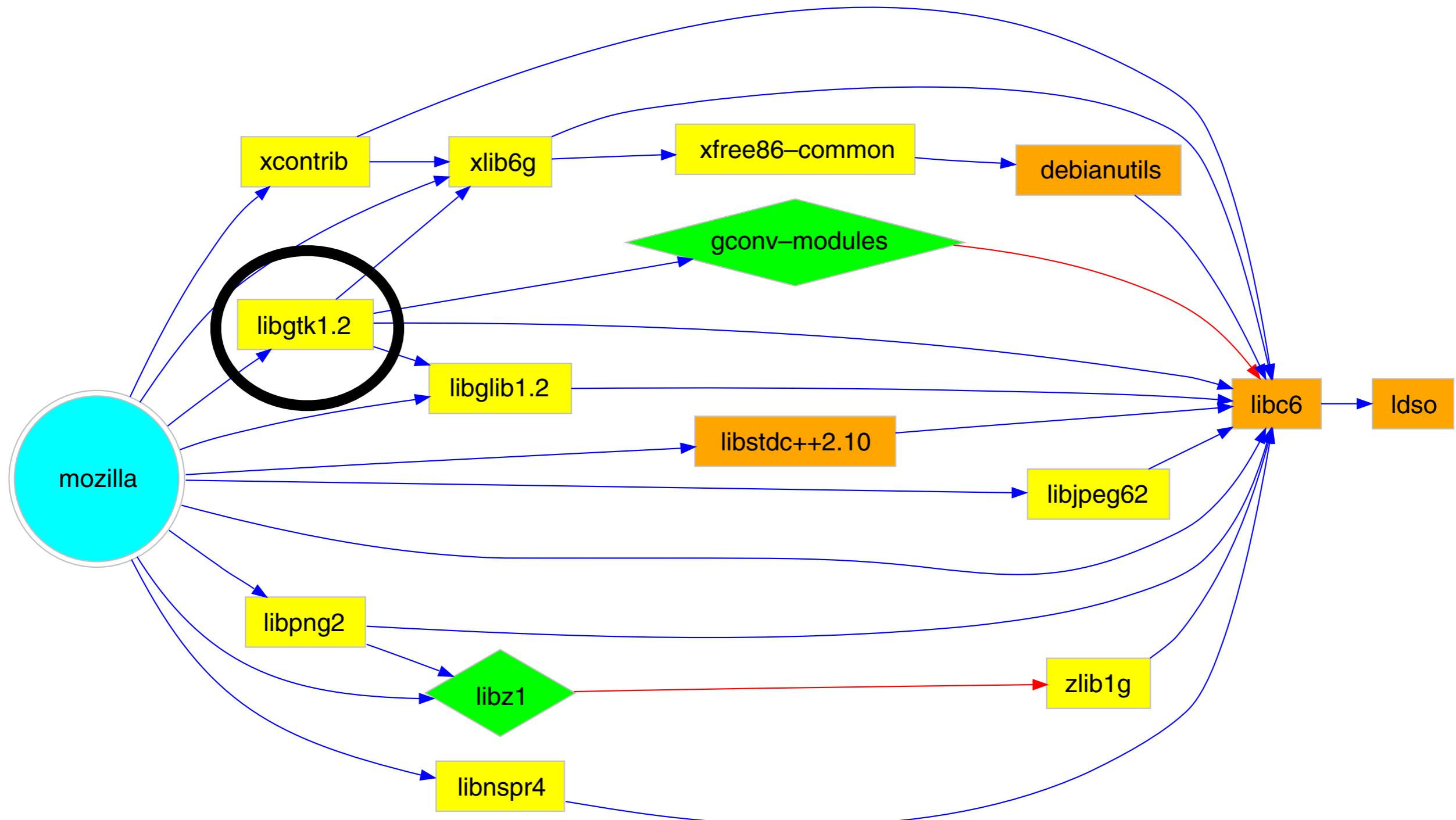
Dependency Management



# Library Transitions Ripple through the whole System



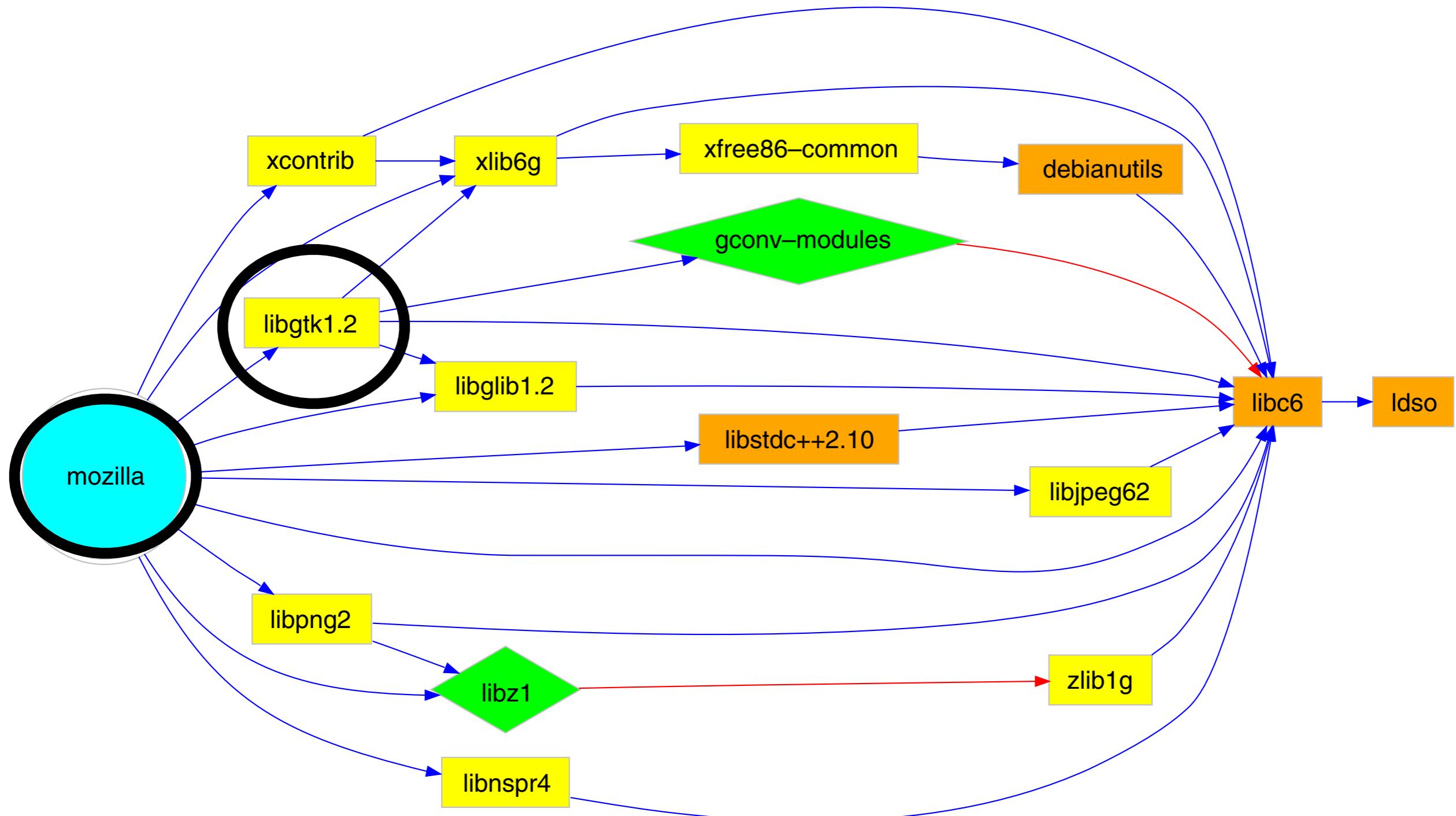
Dependency Management



# Library Transitions Ripple through the whole System



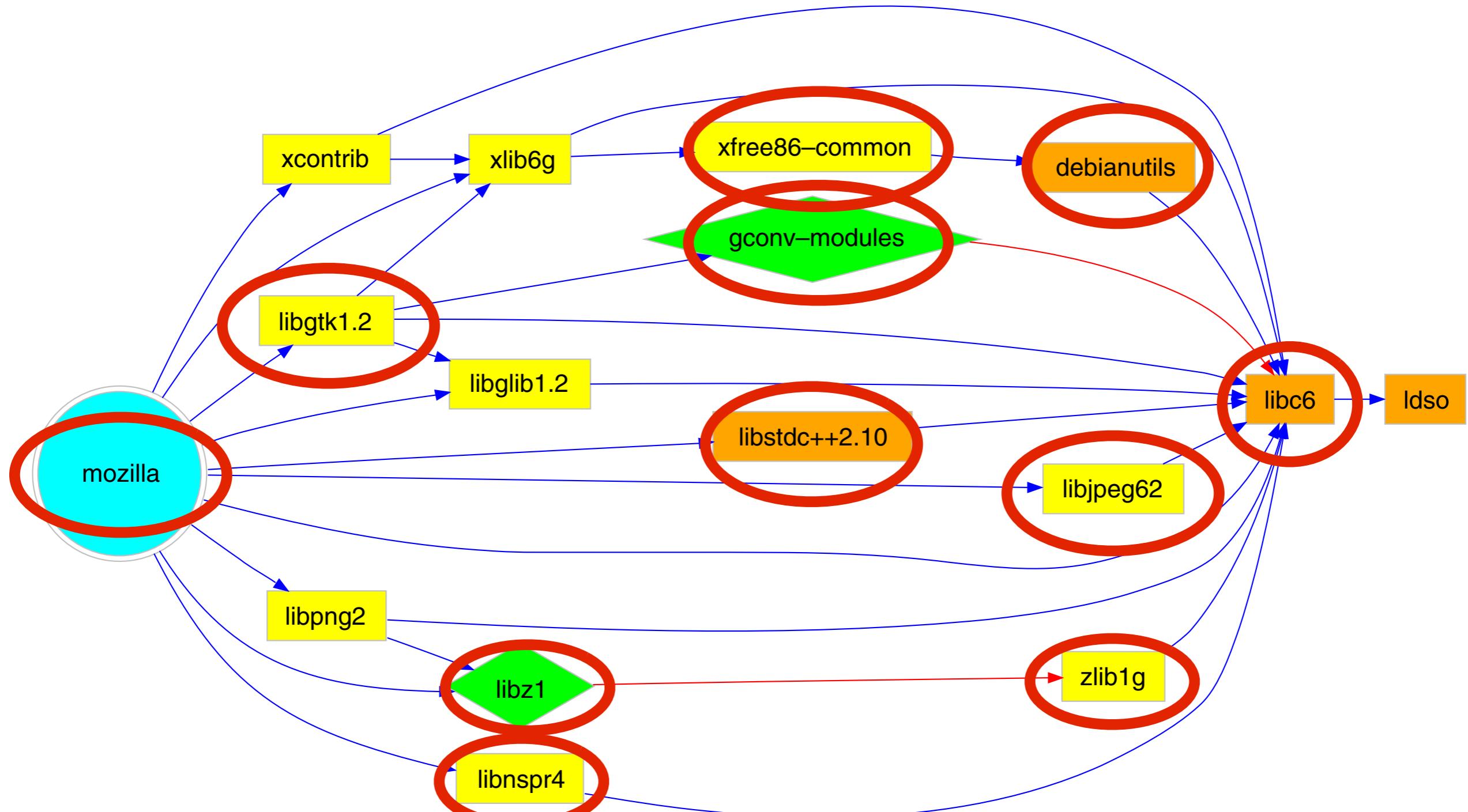
Dependency Management



# Library Transitions Ripple through the whole System



Dependency Management

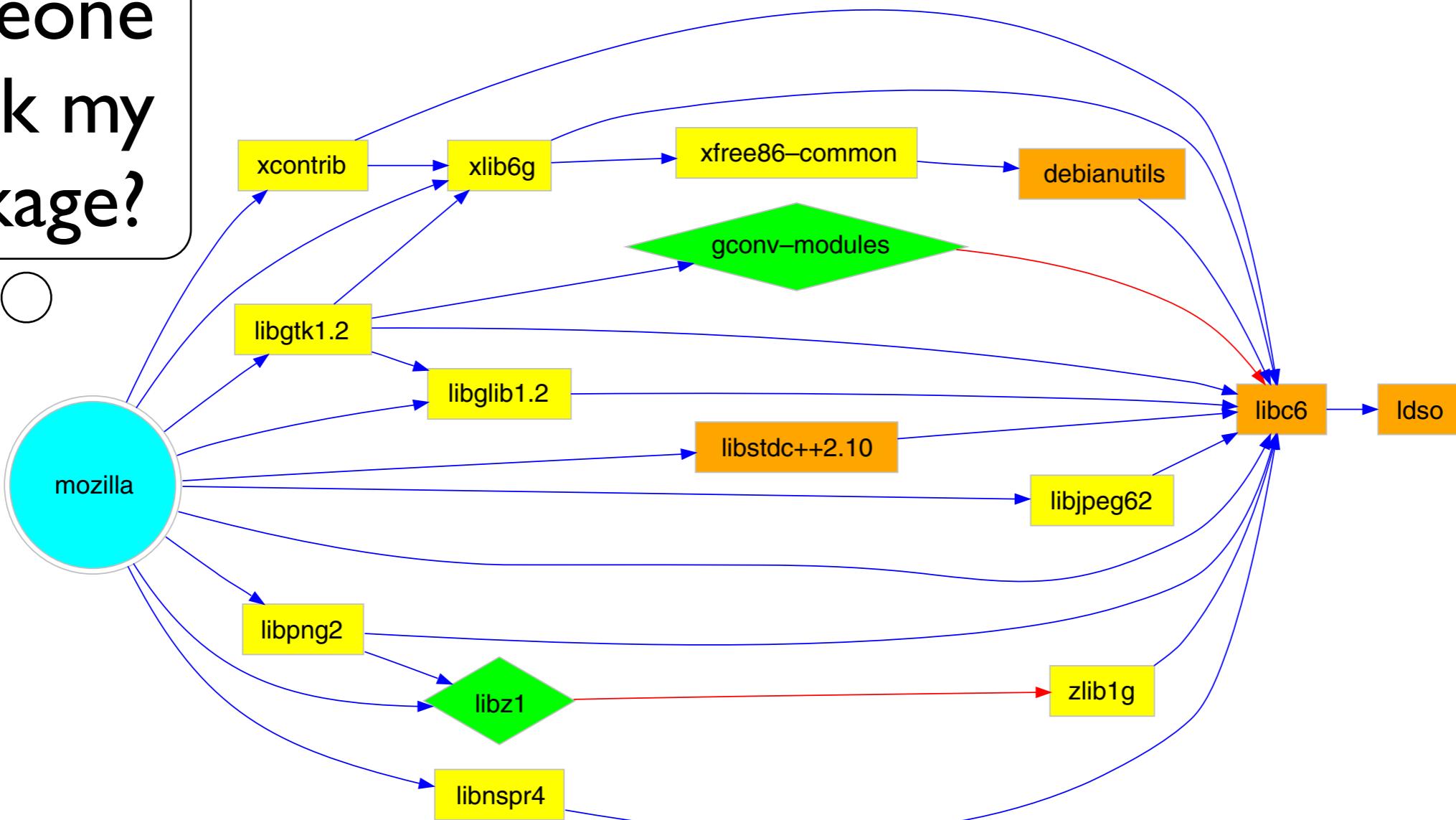
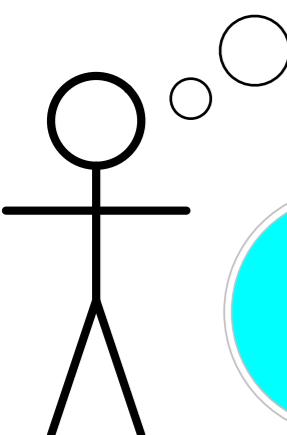


# 2-way Impact Analysis



Dependency Management

Did  
someone  
break my  
package?

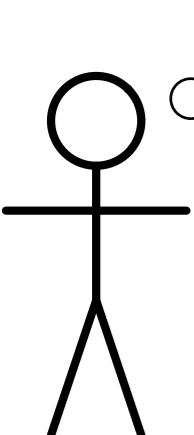


# 2-way Impact Analysis

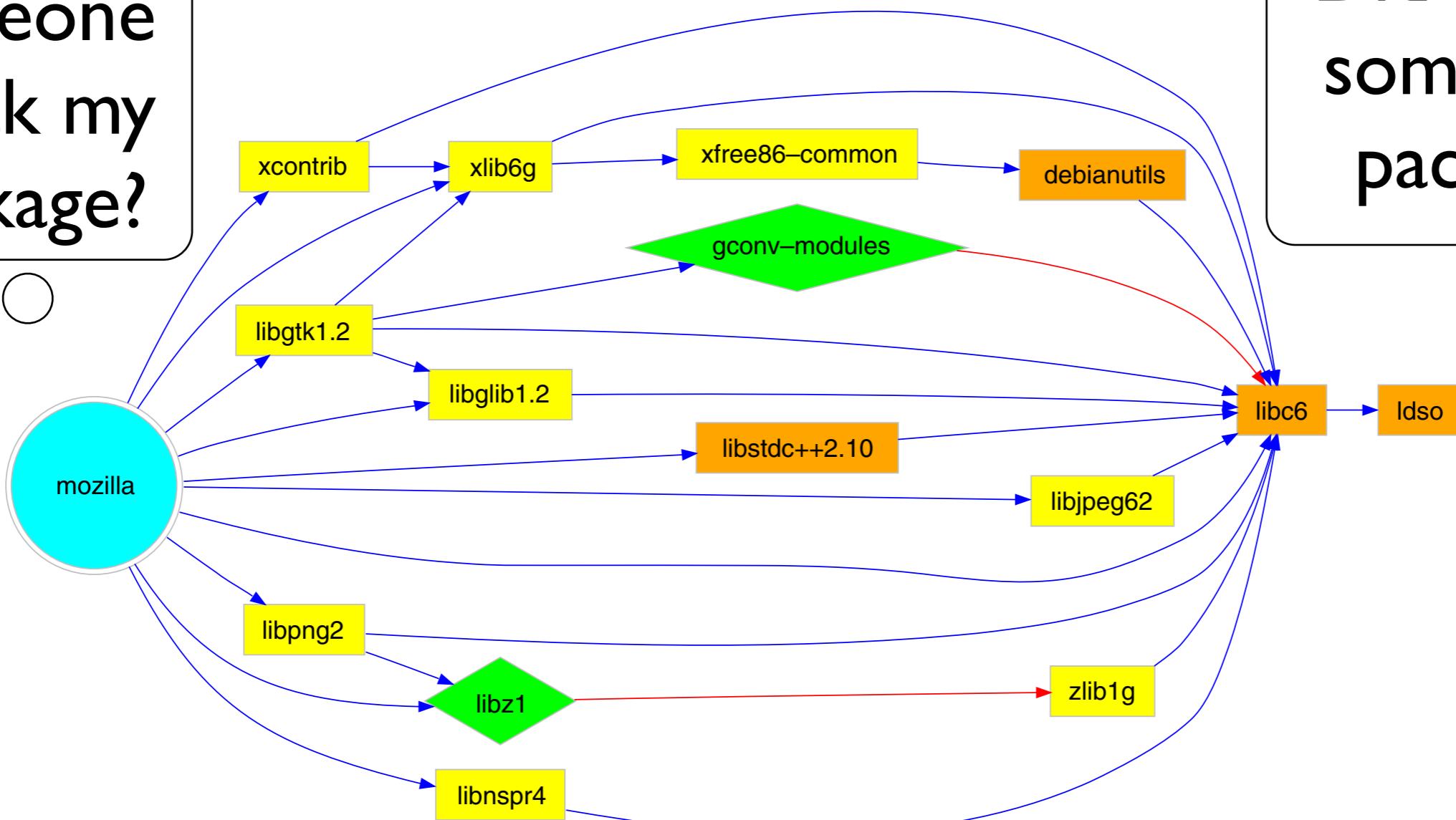
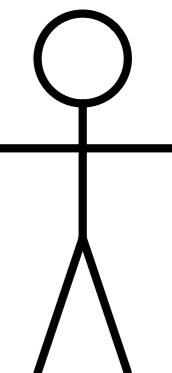


Dependency Management

Did someone break my package?



Did I break someone's package?

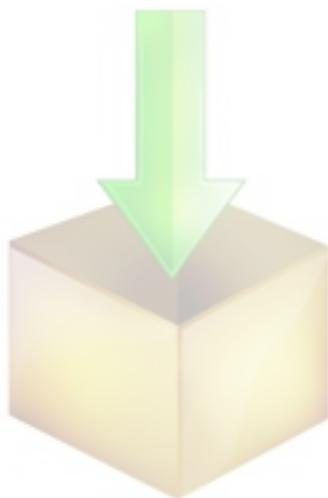




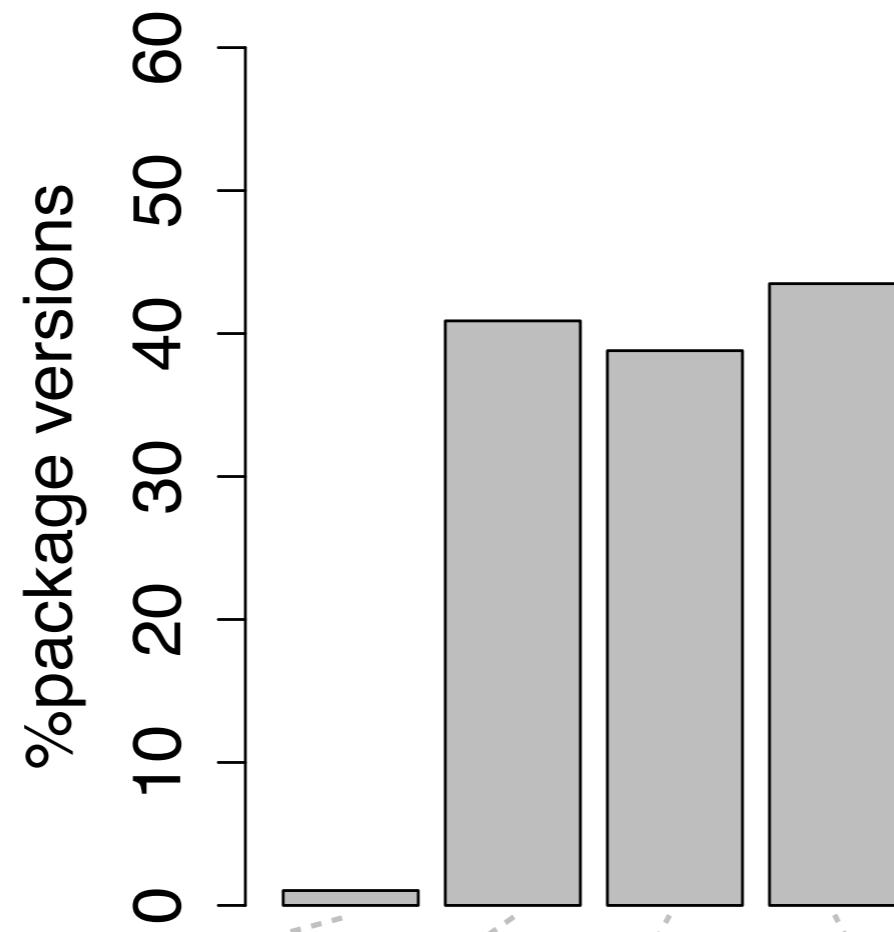
debian



New Package



Upstream Sync



Dependency  
Management

34



Build Change



Local Patch

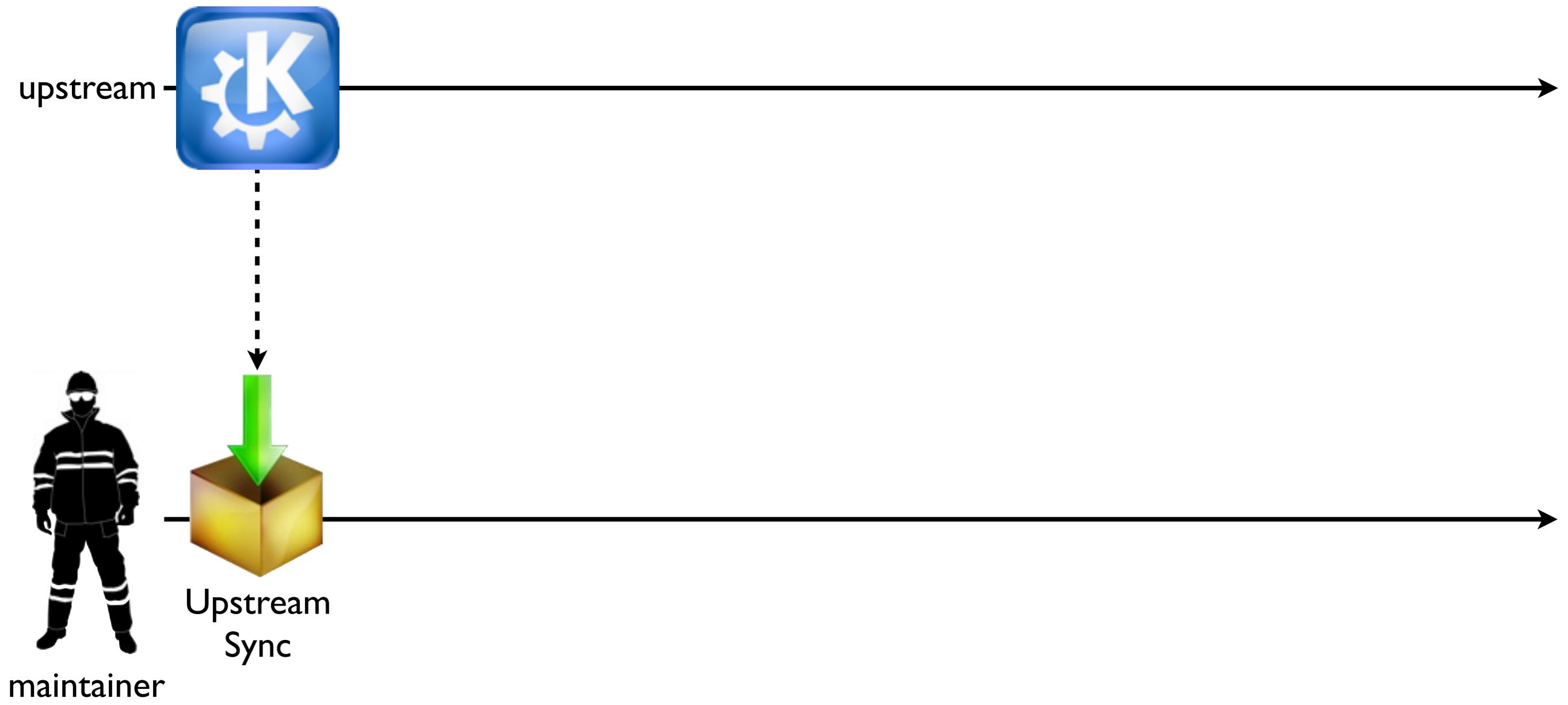


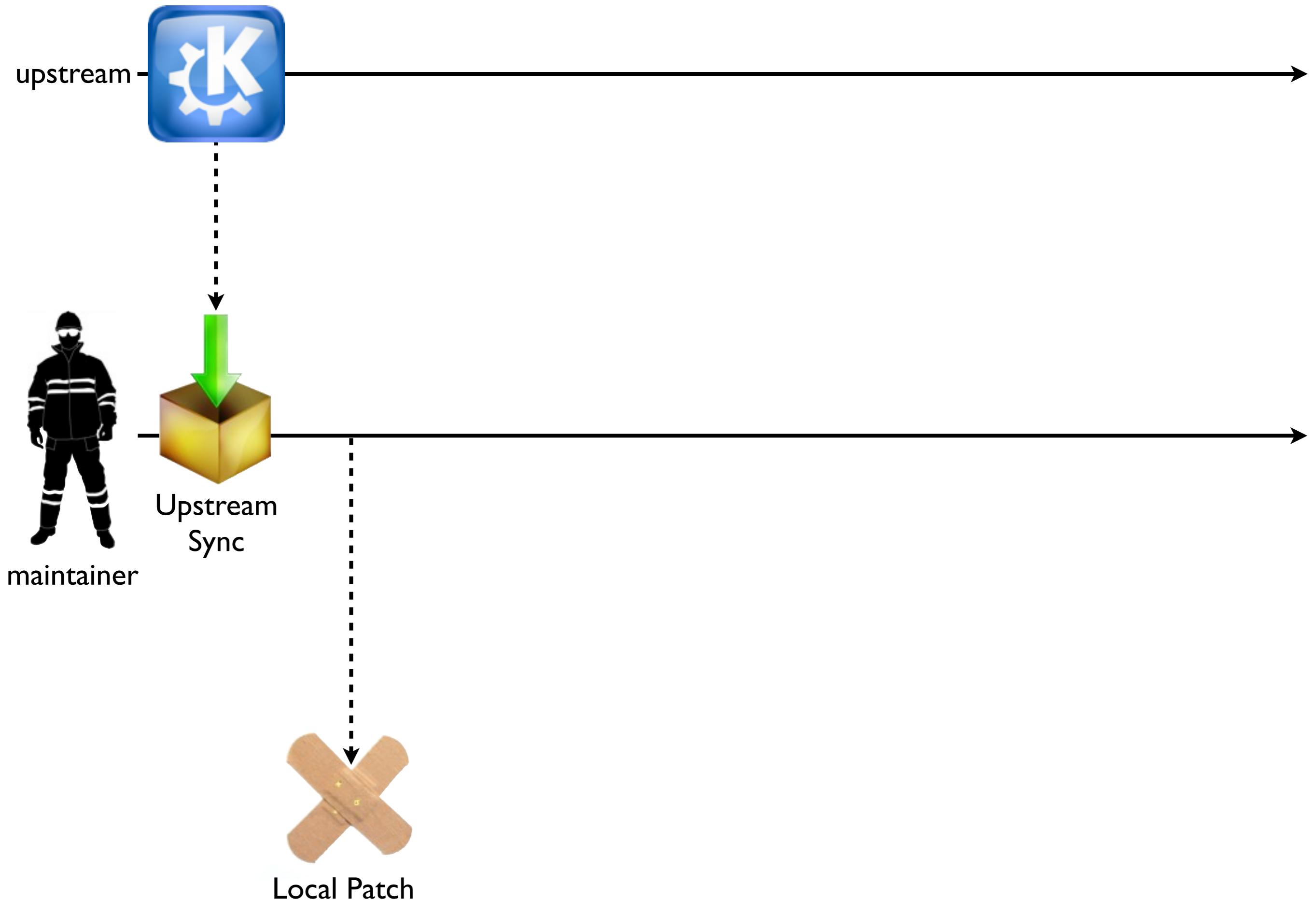
Product-wide  
Concern

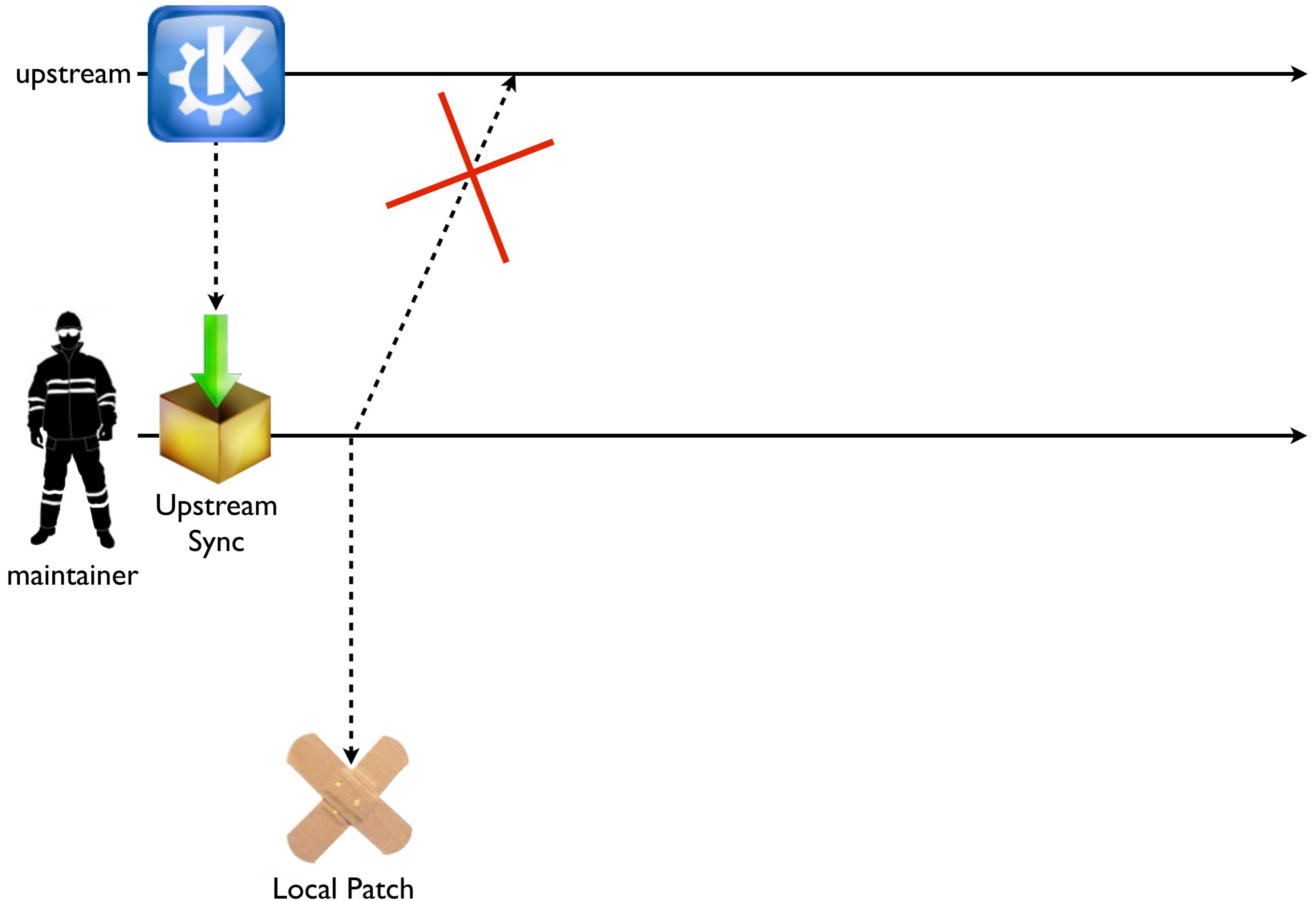
upstream —————→

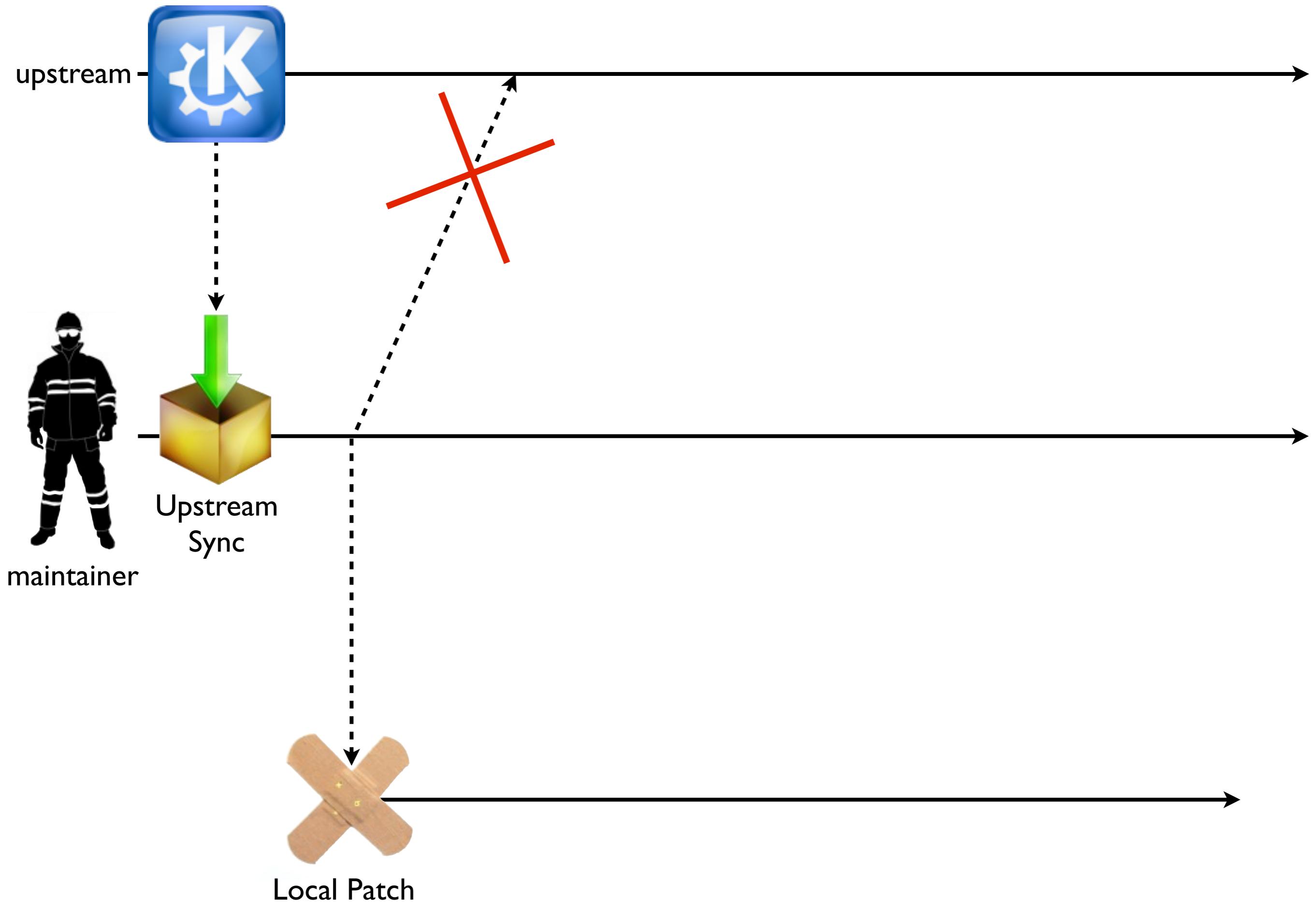


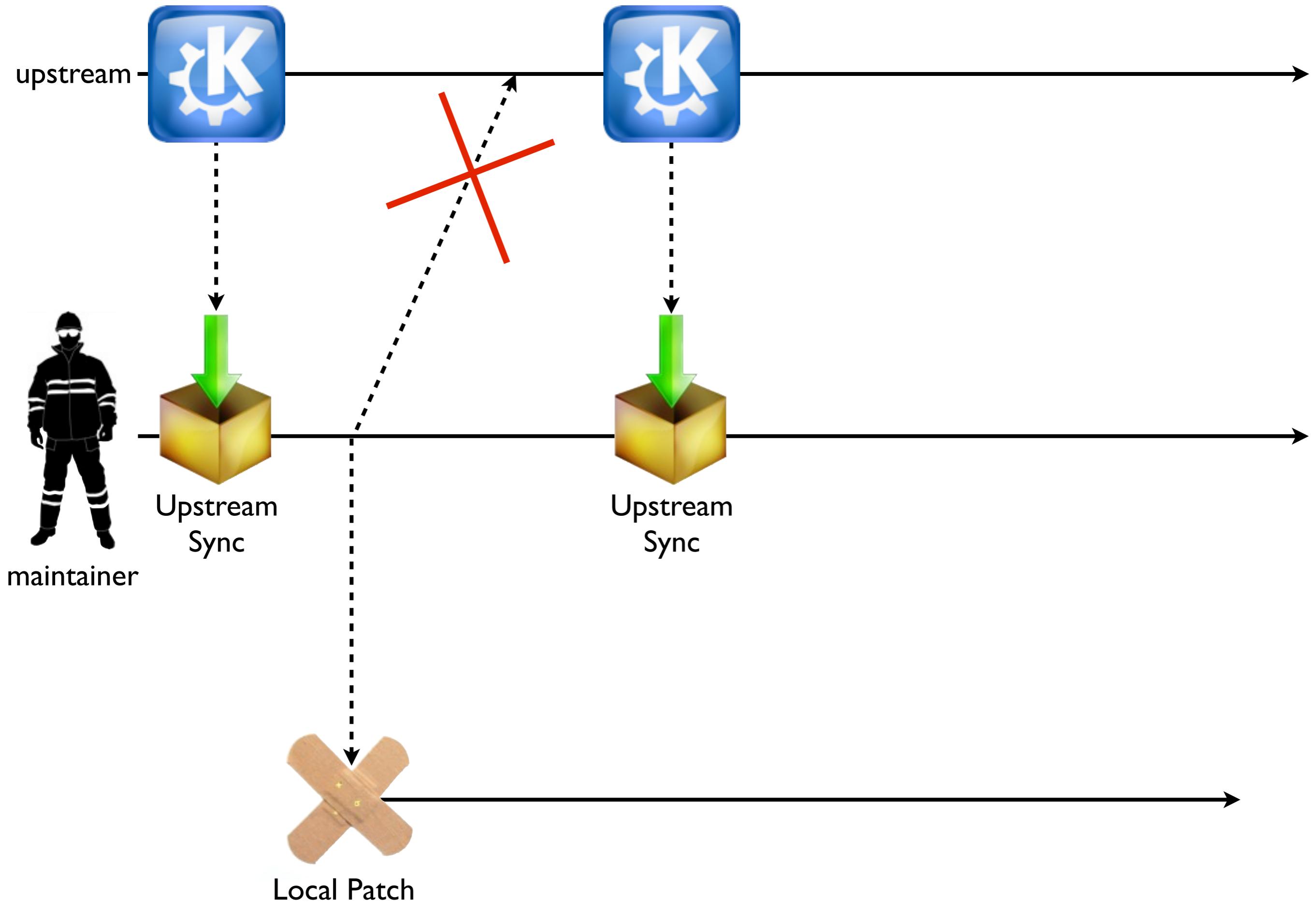
maintainer

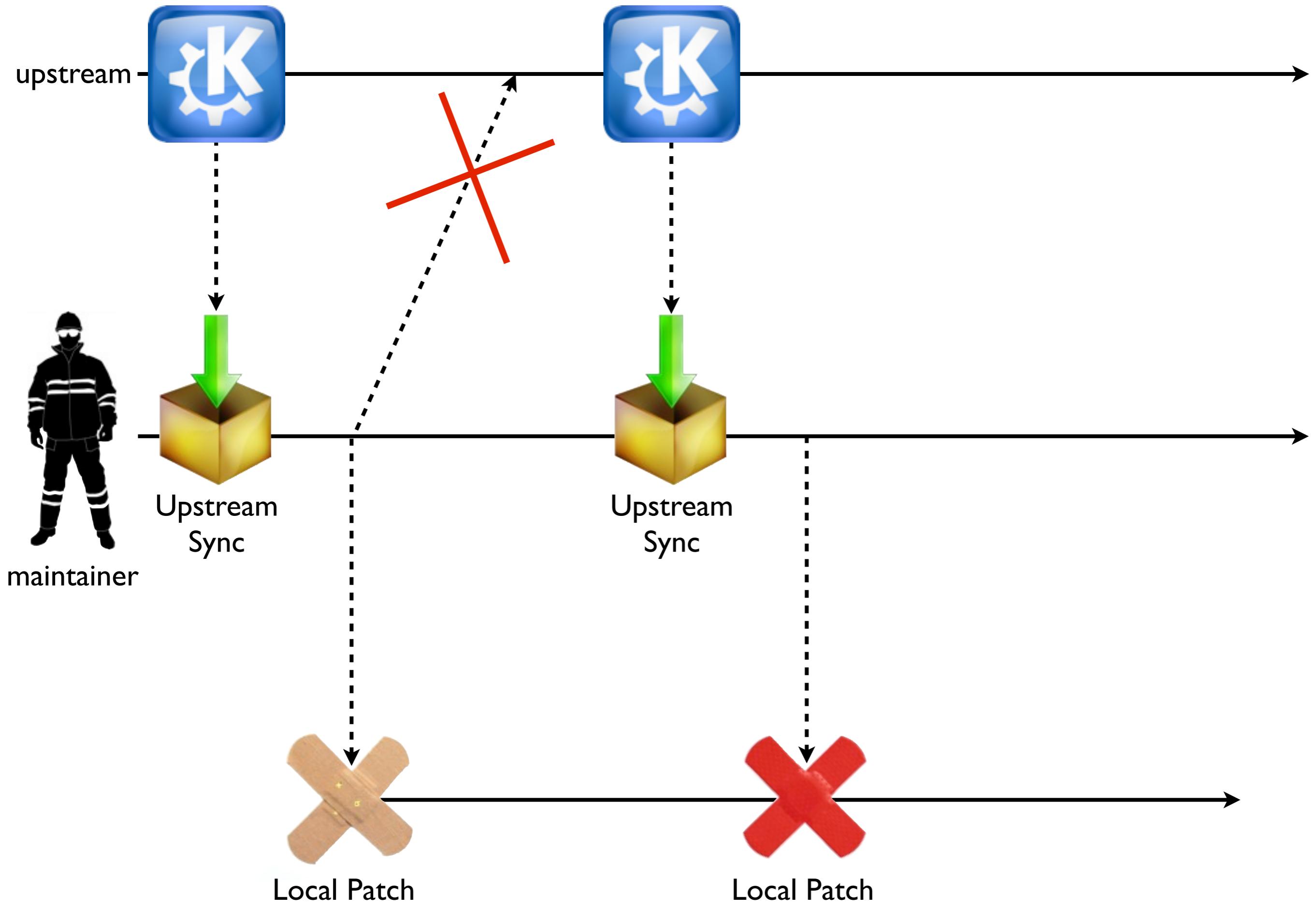


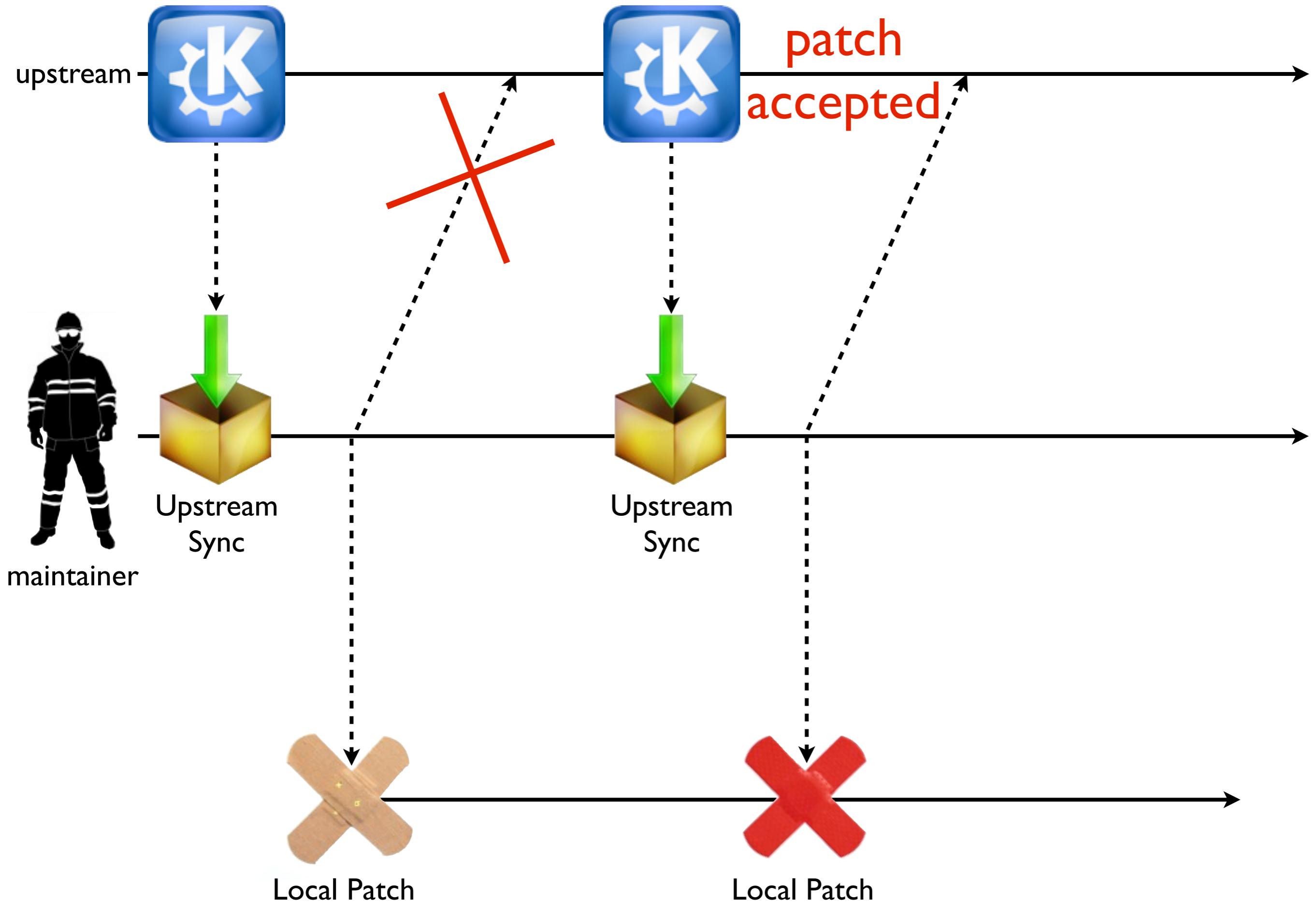


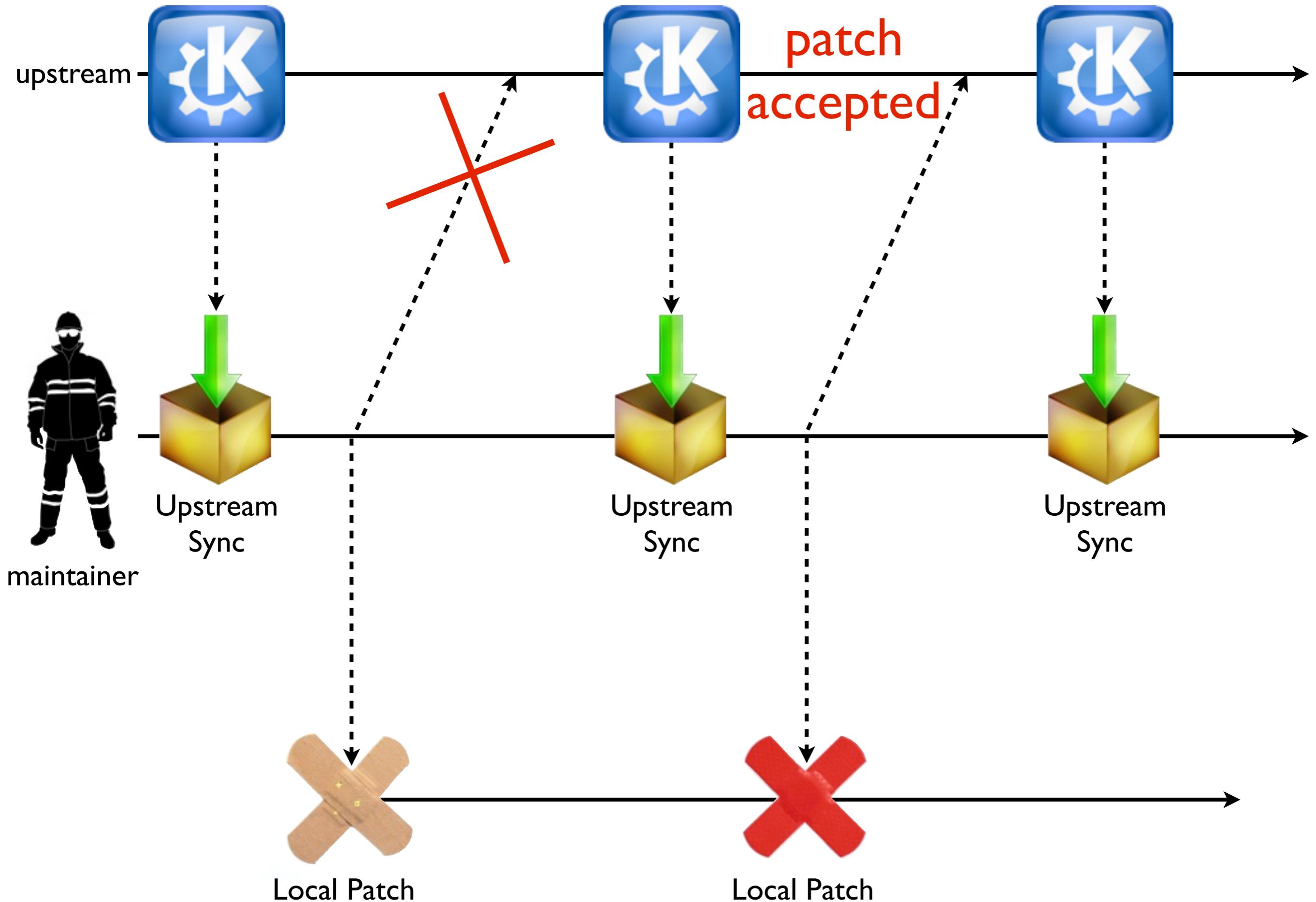


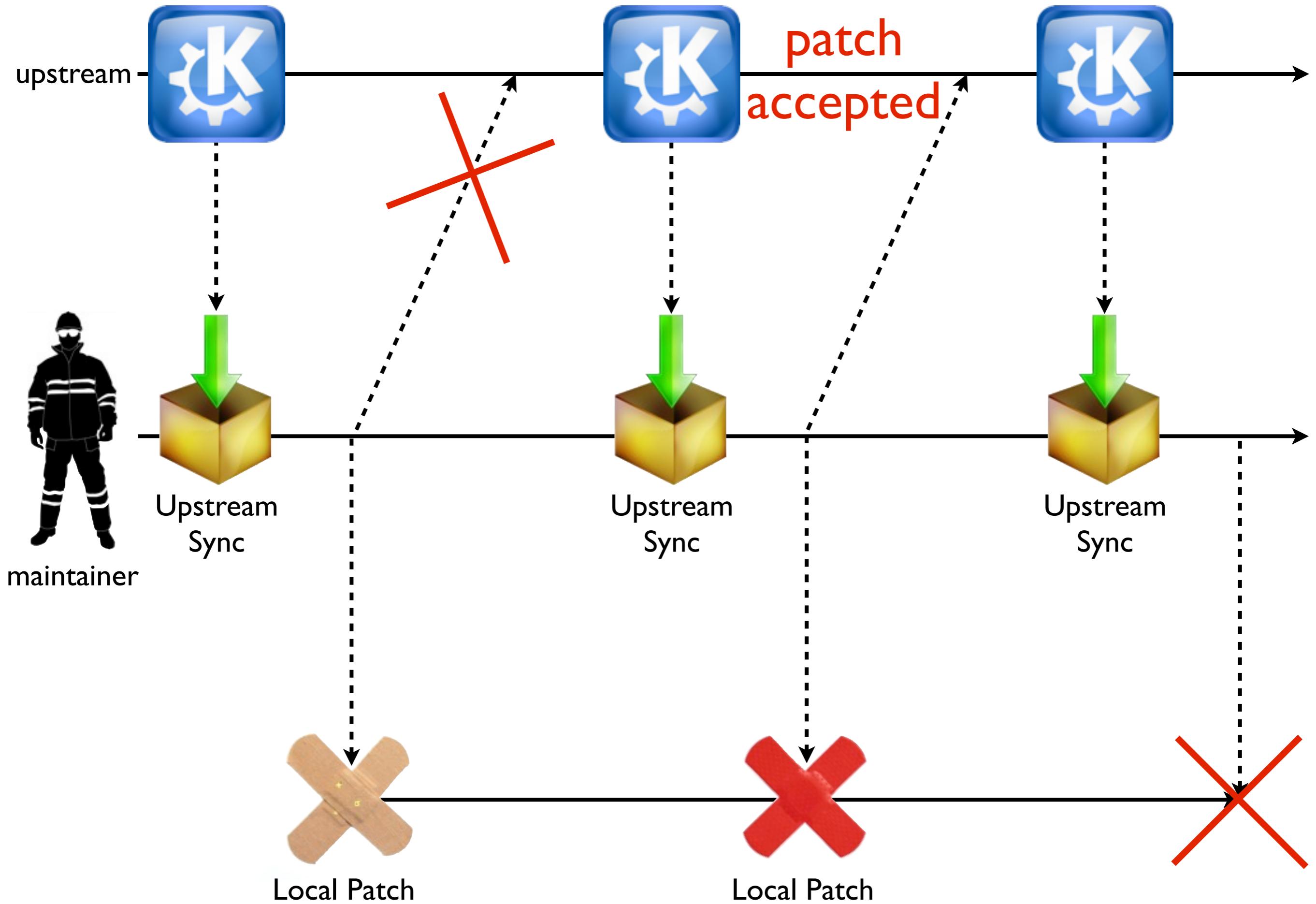


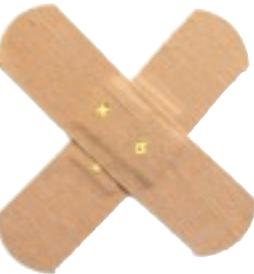












Package : openssl  
Vulnerability : predictable random number generator  
Problem type : remote  
Debian-specific: yes  
CVE Id(s) : CVE-2008-0166  
Date : 2008-05-13

Local Patch

Luciano Bello discovered that the random number generator in Debian's openssl package is predictable. This is **caused by an incorrect Debian-specific change** to the openssl package (CVE-2008-0166). As a result, cryptographic key material may be guessable.

It is strongly recommended that all cryptographic key material which has been generated by OpenSSL versions starting with 0.9.8c-1 on Debian systems is recreated from scratch. Furthermore, **all DSA keys ever used on affected Debian systems for signing or authentication purposes should be considered compromised**; the Digital Signature Algorithm relies on a secret random value used during signature generation.

The **first vulnerable version**, 0.9.8c-1, was uploaded to the unstable distribution on **2006-09-17**, and has since propagated to the testing and current stable (etch) distributions. The old stable distribution (sarge) is not affected.



Local Patch

Package : openssl  
Vulnerability : predictable random number generator  
Problem type : remote  
Debian-specific: yes  
CVE Id(s) : CVE-2008-0166  
Date : 2008-05-13

Luciano Bello discovered that the random number generator in Debian's openssl package is predictable. This is **caused by an incorrect Debian-specific change** to the openssl package (CVE-2008-0166). As a result, cryptographic key material is no longer guessable.

>=1.5 years 8-(

It is strongly recommended that all cryptographic key material which has been generated by OpenSSL versions starting with 0.9.8c-1 on Debian systems is recreated from scratch. Furthermore, **all DSA keys ever used on affected Debian systems for signing or authentication purposes should be considered compromised**; the Digital Signature Algorithm relies on a secret random value used during signature generation.

The **first vulnerable version**, 0.9.8c-1, was uploaded to the unstable distribution on 2006-09-17, and has since propagated to the testing and current stable (etch) distributions. The old stable distribution (sarge) is not affected.



Local Patch

Package : openssl  
Vulnerability : predictable random number generator  
Problem type : remote  
Debian-specific: yes  
CVE Id(s) : CVE-2008-0166  
Date : 2008-05-13

Luciano Bello discovered that the random number generator in Debian's openssl package is predictable. This is **caused by an incorrect Debian-specific change** to the openssl package (CVE-2008-0166). As a result, cryptographic key material is guessable.

>=1.5 years 8-(

It is strongly recommended that all cryptographic key material which has been generated by OpenSSL versions starting with 0.9.8c-1 on Debian systems is recreated from scratch. Furthermore, **all DSA keys ever used on affected Debian systems for signing or authentication purposes should be considered compromised**; the Digital Signature Algorithm relies on a secret random value used during signature generation.

>=44 distributions ;-)

The **first vulnerable version**, 0.9.8c-1, was uploaded to the unstable distribution on 2006-09-17, and has since propagated to the testing and current stable (etch) distributions. The old stable distribution (sarge) is not affected.

# Open Challenges



# Improving Software and Build Comprehension

# Where are the Tools?

refactor  
your makefiles!

test your  
build!

what did the  
other team break  
now ;-)

keep poking those  
upstream guys until they  
give in :-)







Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty times**  
a day.

in 00 different countries!

Sign in with:



[Choose Your FREE Avatar](#)

Over 2 Million people  
like IMVU on Facebook!

FREE

M  
C  
I  
S



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.

in 00 different countries!

Sign in with: [f](#) [g](#) [Y!](#)

[Choose Your FREE Avatar](#)

Over 2 Million people  
like IMVU on Facebook!

**FREE**  
M  
C:I:S

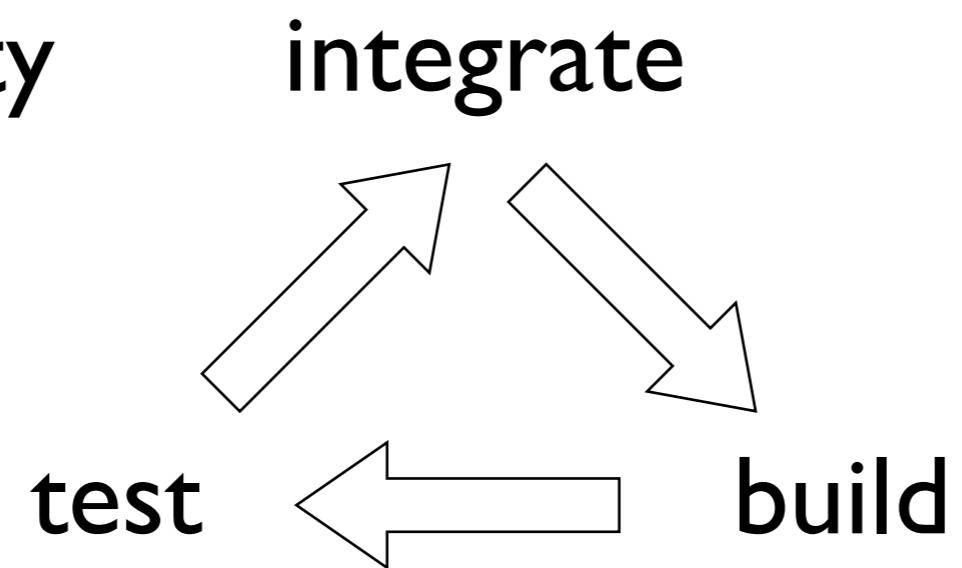
# Release Engineering



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development

*keeping cycle time down*



deployment

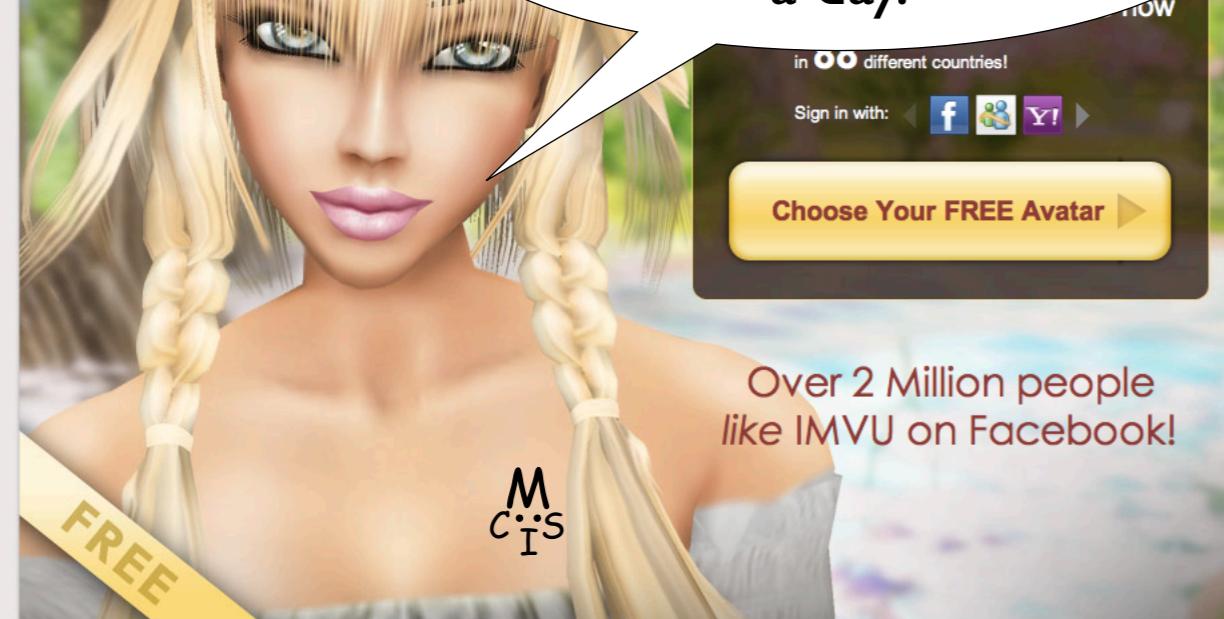




Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

# Release Engineering

in-house/3rd party  
development

integrate

test

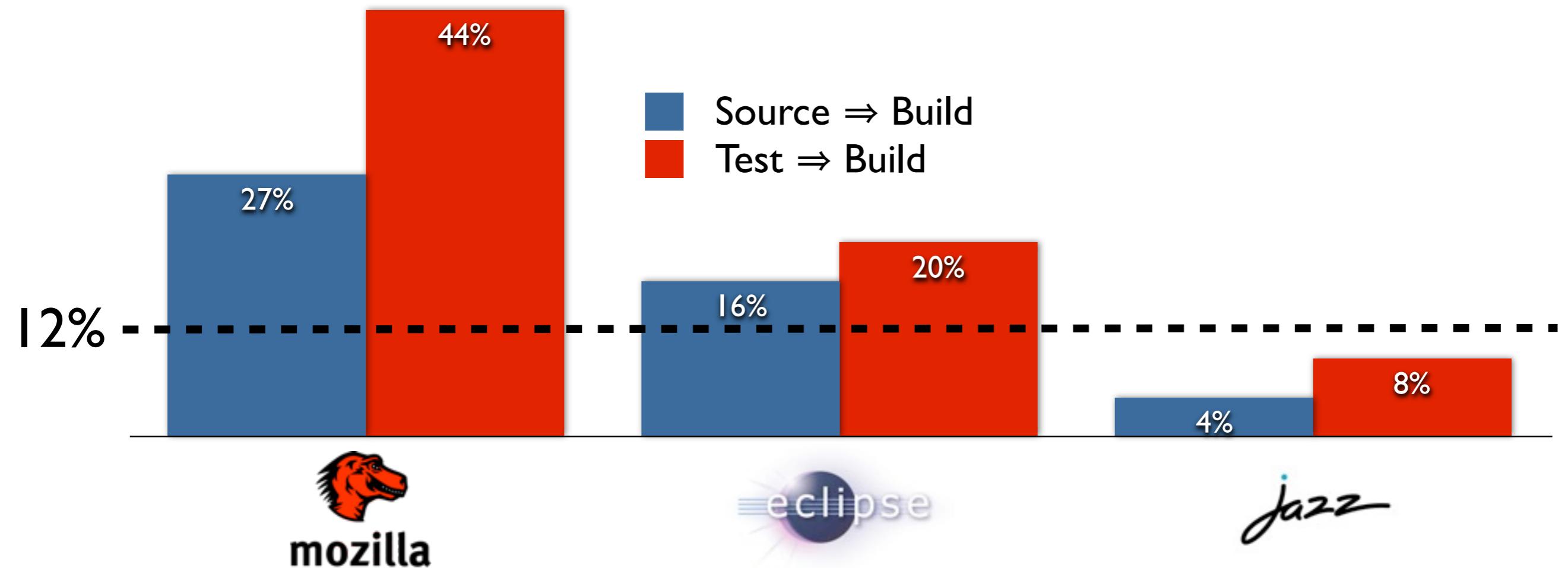
build

*keeping cycle time down*



deployment

# The Build System Requires Significant Maintenance

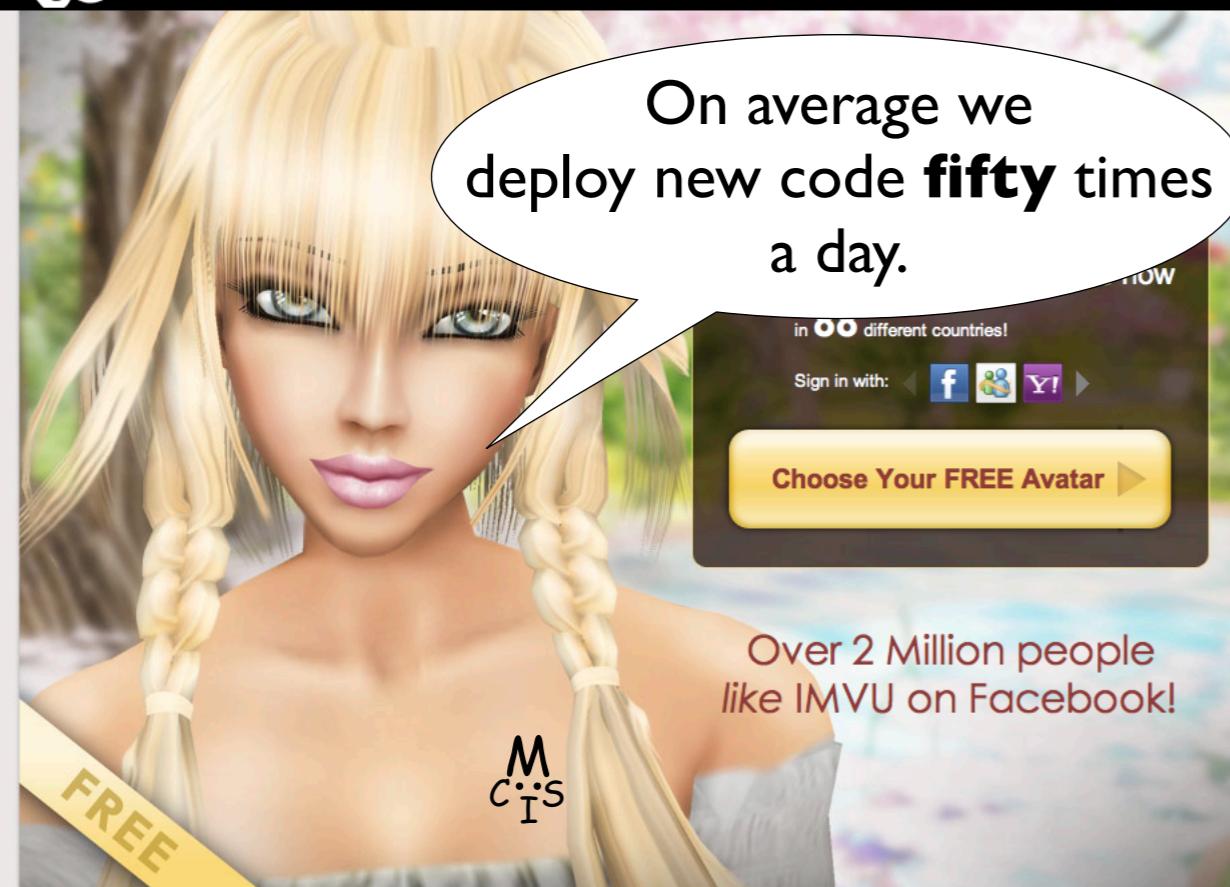




Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development

integrate

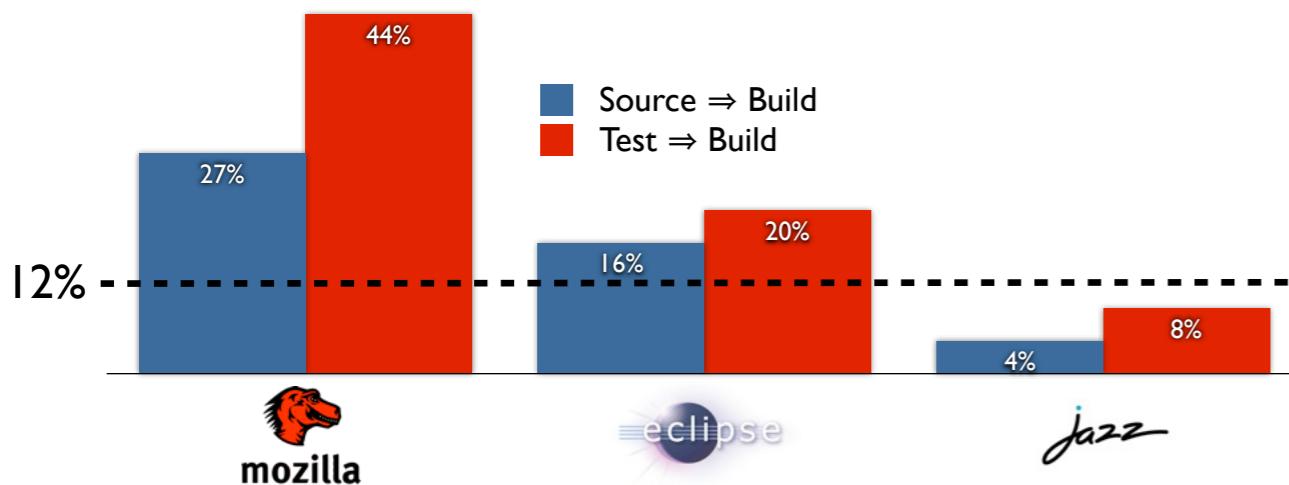
test ← build

*keeping cycle time down*



deployment

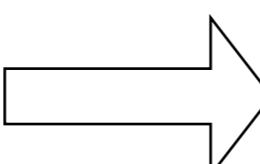
## The Build System Requires Significant Maintenance



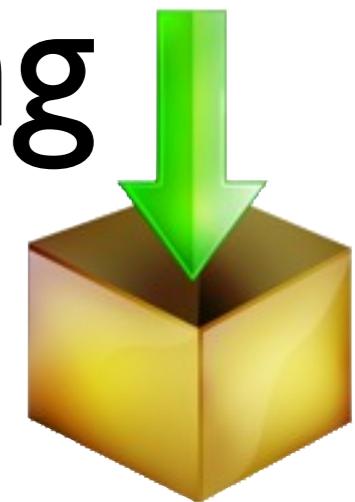
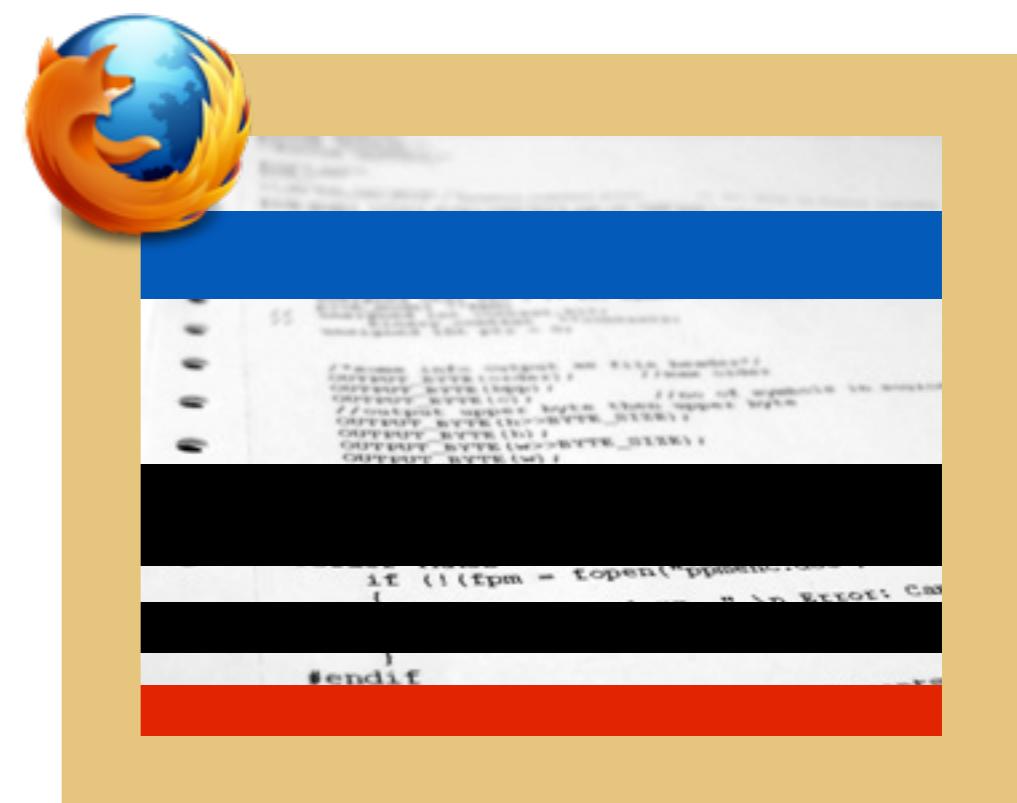
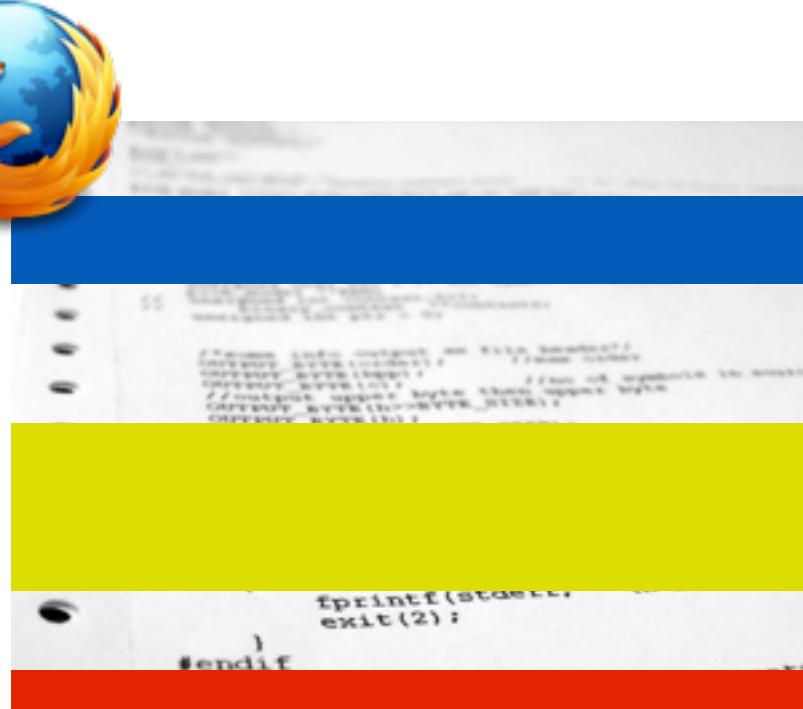
# Risk Analysis & Cherry-picking



diff



## **maintainer**



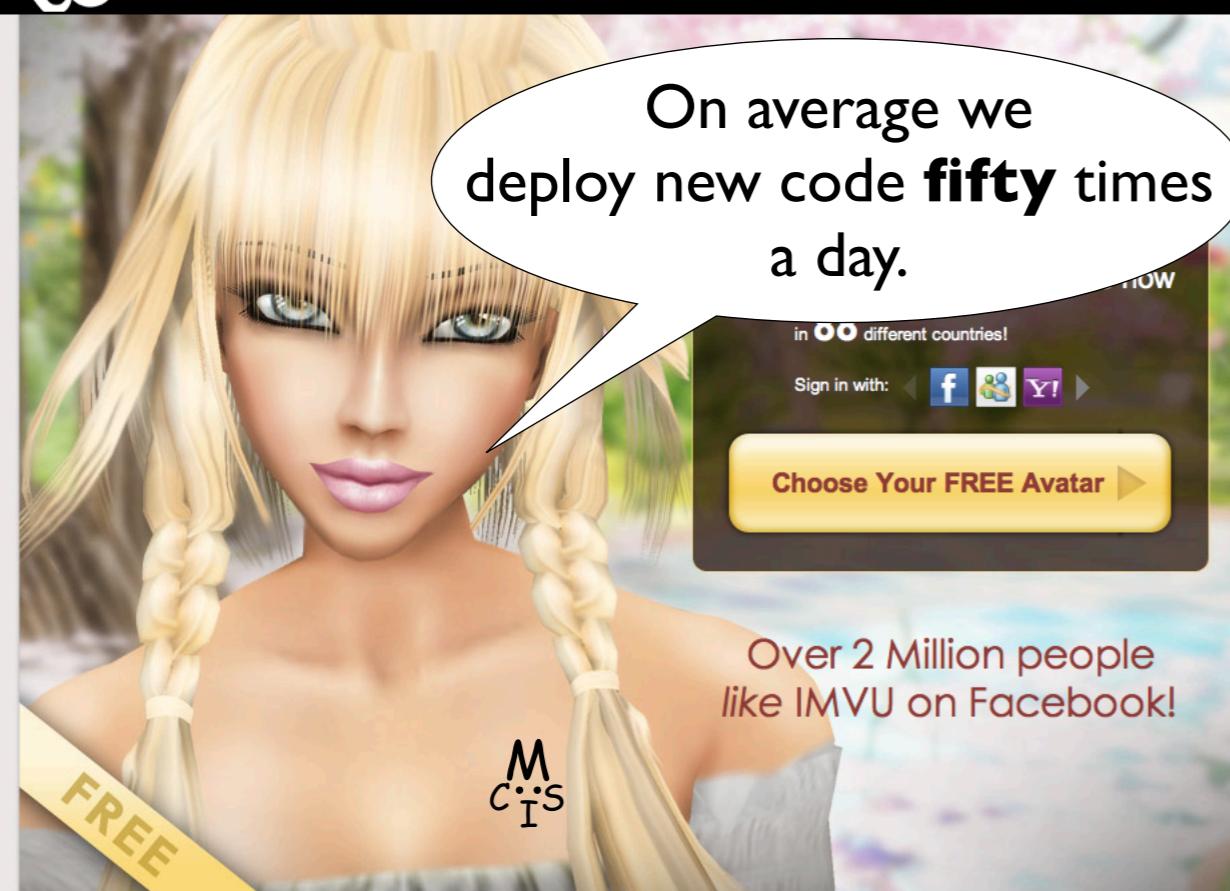
# Upstream Sync



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.



in-house/3rd party  
development

integrate

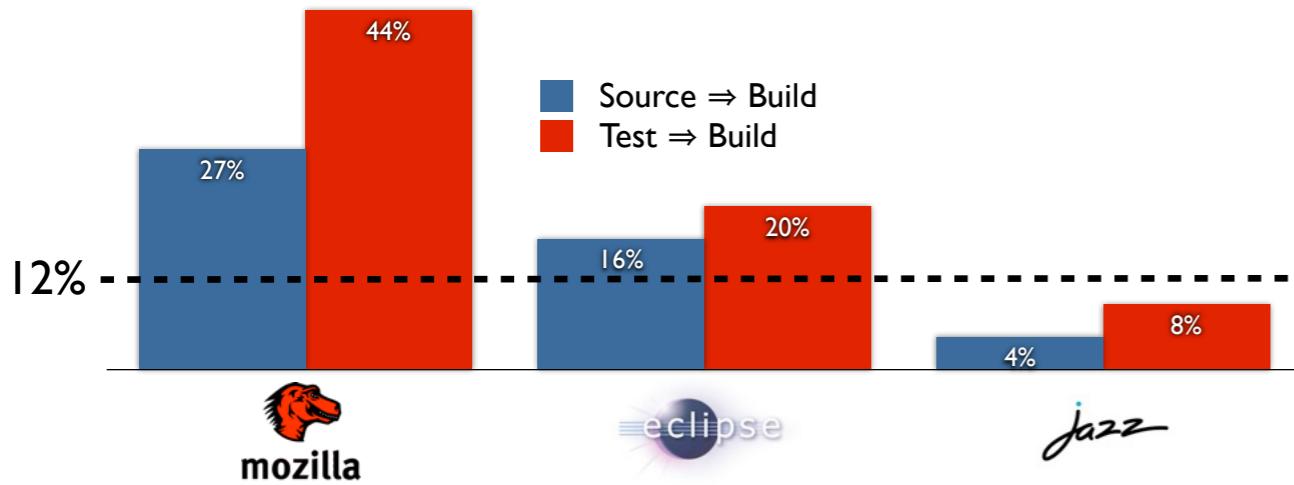
test ← build

*keeping cycle time down*



deployment

## The Build System Requires Significant Maintenance



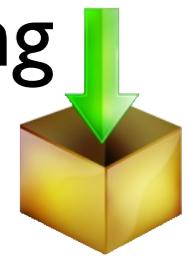
## Risk Analysis & Cherry-picking



diff



maintainer



Upstream Sync

