Comparing Apples to Oranges

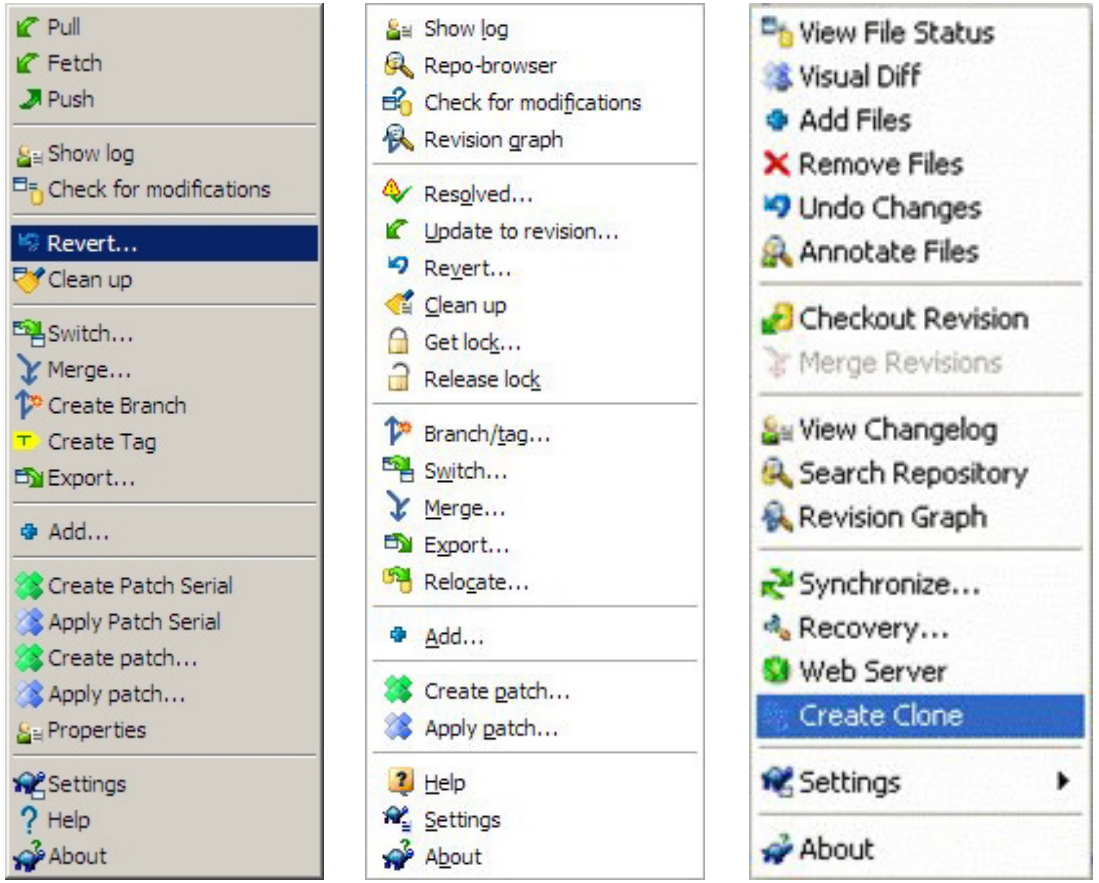
Subversion, Git, and Mercurial

SubConf 2009

Stefan Sperling <stsp@elego.de>

Stephen Butler <sbutler@elego.de>

Is there SCM tool convergence?



Real-life situations

elego Software Solutions GmbH

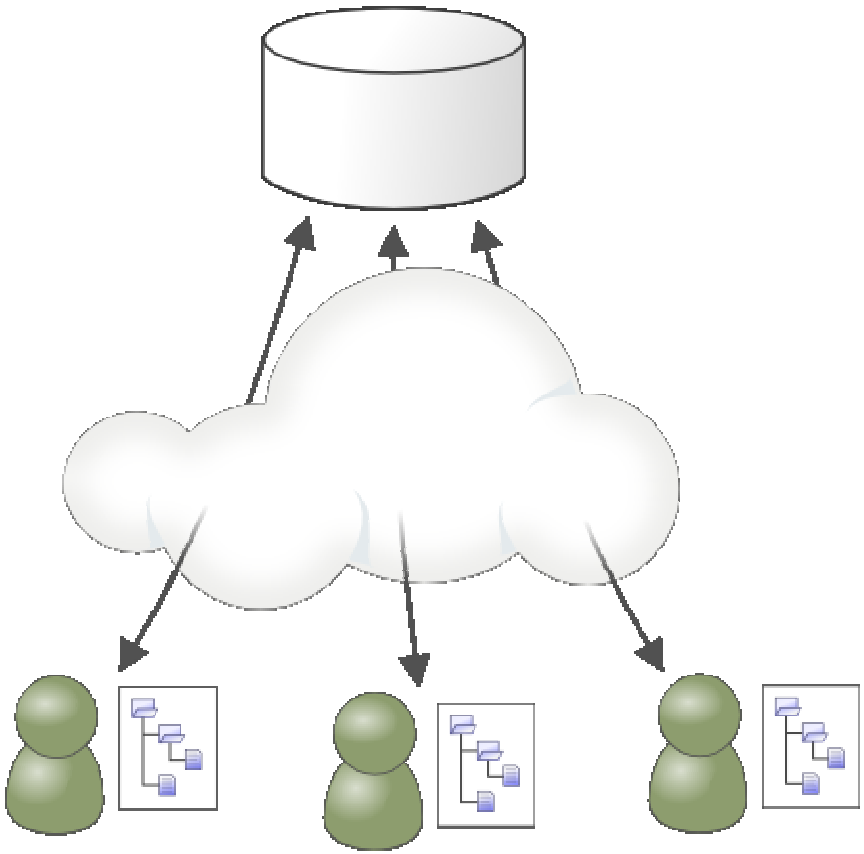
1. Reserving a file for editing
   - Centralized vs distributed
2. Merging a text edit to a renamed file
   - How is tree history modeled?
3. Merging a file move
   - Distinguishing file move from directory rename
4. Cherry-picking a revision
   - Fine-grained merge tracking

Real-life situations

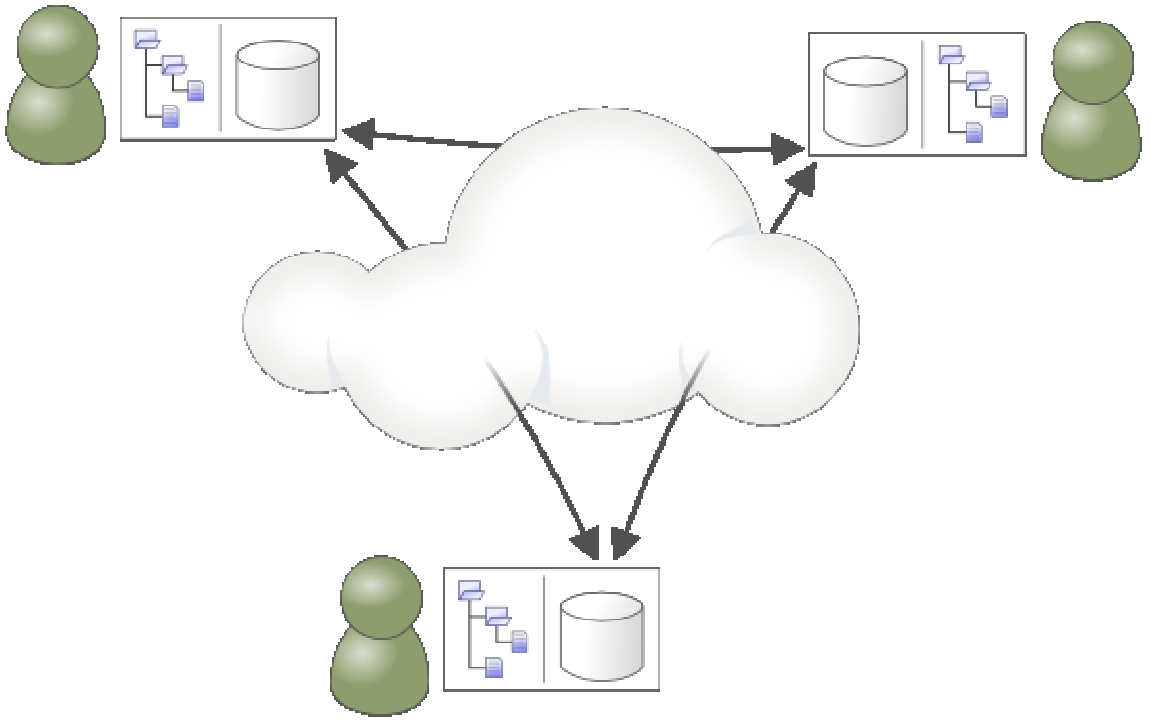
elego Software Solutions GmbH

1. Reserving a file for editing
   - Centralised vs distributed

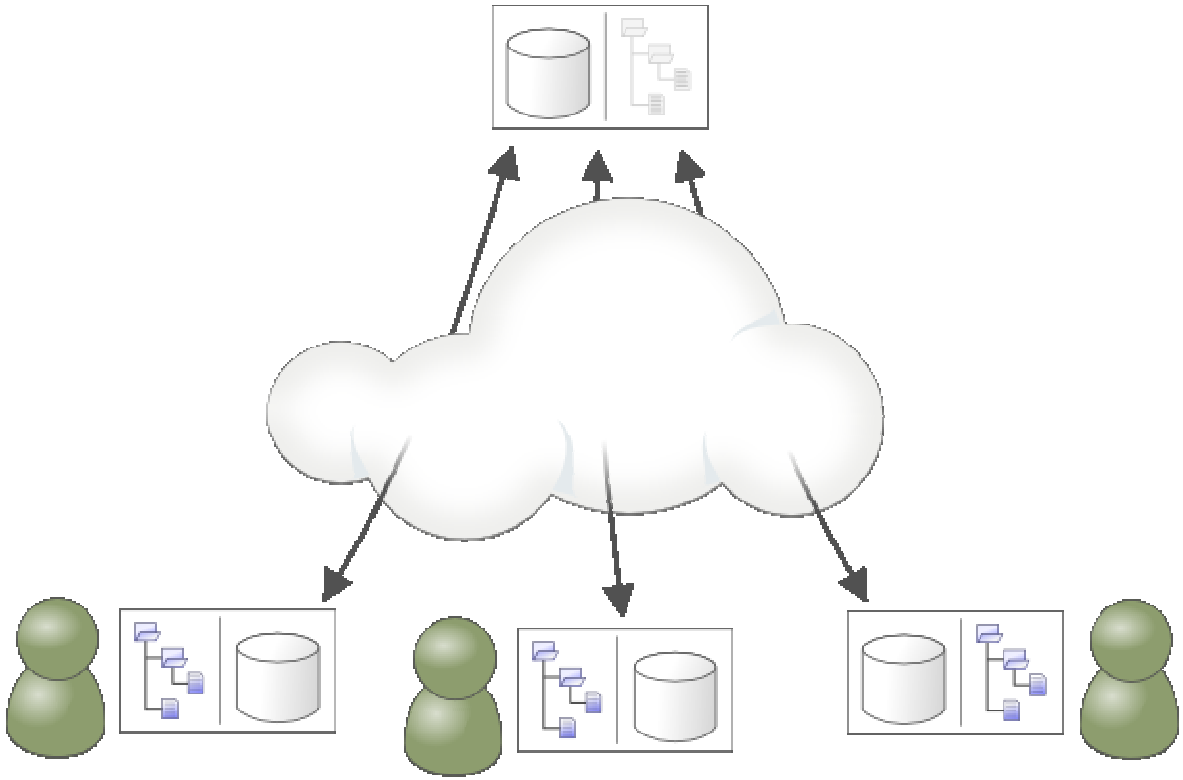
Centralised design



Distributed design



Distributed design with a central server



Real-life situations

elego Software Solutions GmbH

1. Reserving a file for editing
   - Centralised vs distributed
2. Merging a text edit to a renamed file
   - How is tree history modeled?

Merging a text edit to a renamed file

- Edit the **text** of file foo.c in branch A:

```
--- foo.c
+++ foo.c
@@ -1,4 +1,4 @@
 #include <stdio.h>
 void main() {
-    printf("Hello world!\n");
+    printf("Goodbye world!\n");
 }
```

- **Rename** file foo.c to bar.c in branch B
- Merge the text change from branch A to branch B

Merging a text edit to a renamed file

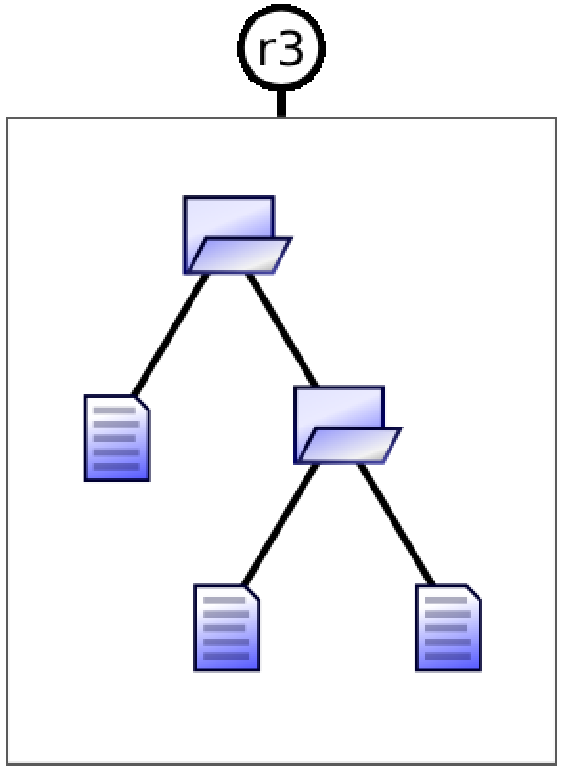
elego Software Solutions GmbH

- Subversion:

```
$ svn merge ^/A
--- Merging r3 through r5 into '.':
   C foo.c
Summary of conflicts:
  Tree conflicts: 1
$ svn status
 M       .
!     C foo.c
      >   local missing, incoming edit upon merge
$ svn merge ^/A/foo.c@2 ^/A/foo.c@5 bar.c
--- Merging r3 through r5 into 'bar.c':
U    bar.c
$ cat bar.c
#include <stdio.h>
int main() {
        printf("Goodbye world!\n");
}
$ svn resolved foo.c
```

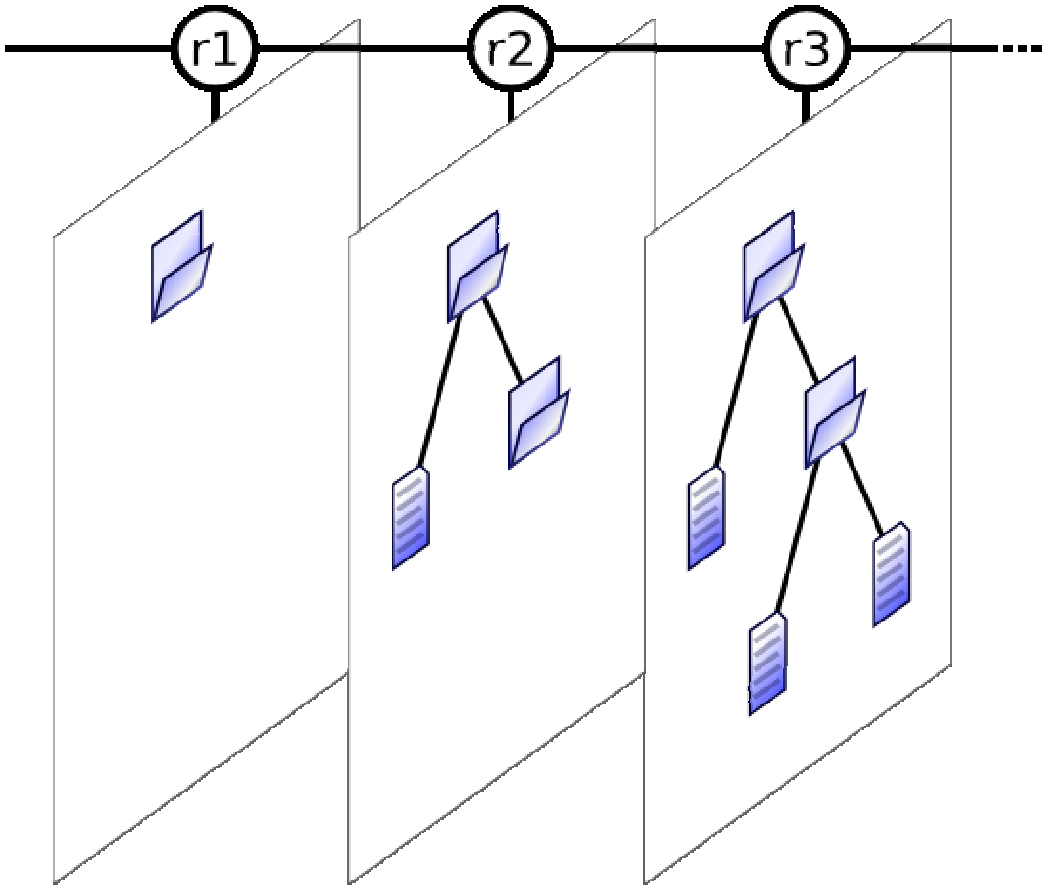
Subversion 3D tree

elego
Software Solutions GmbH

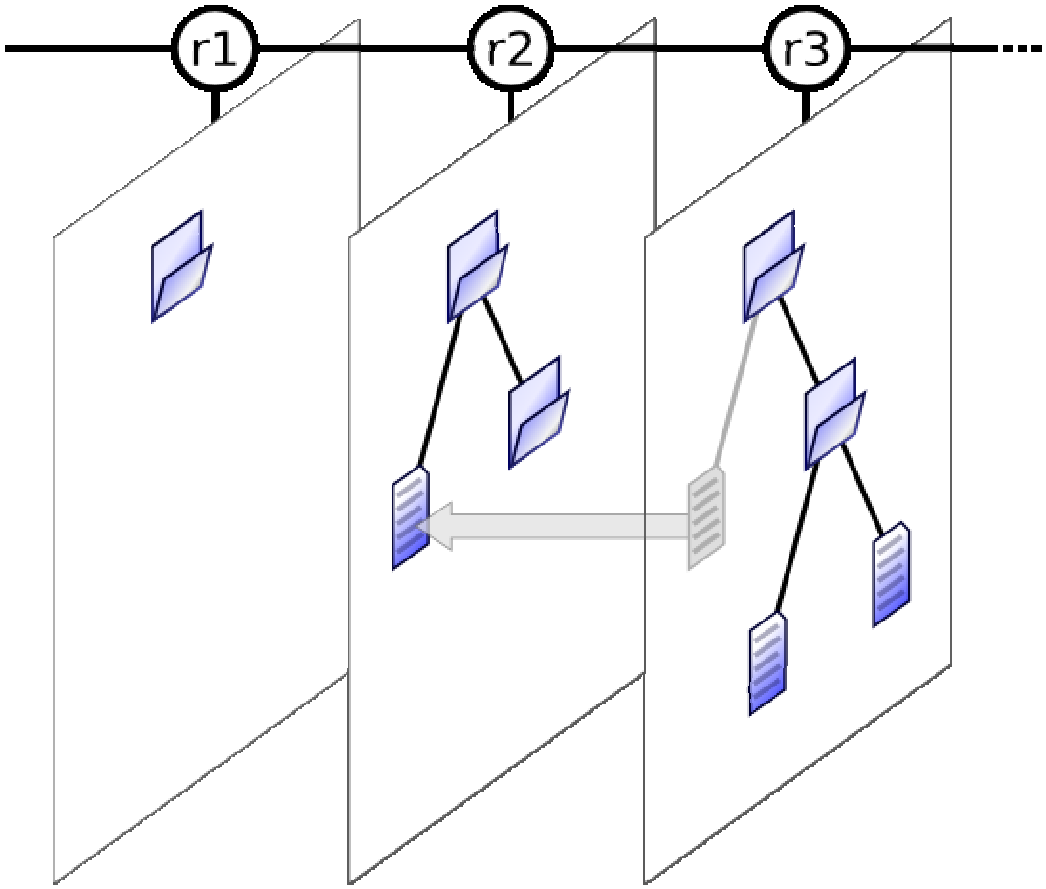


Subversion 3D tree

elego
Software Solutions GmbH



Subversion 3D tree



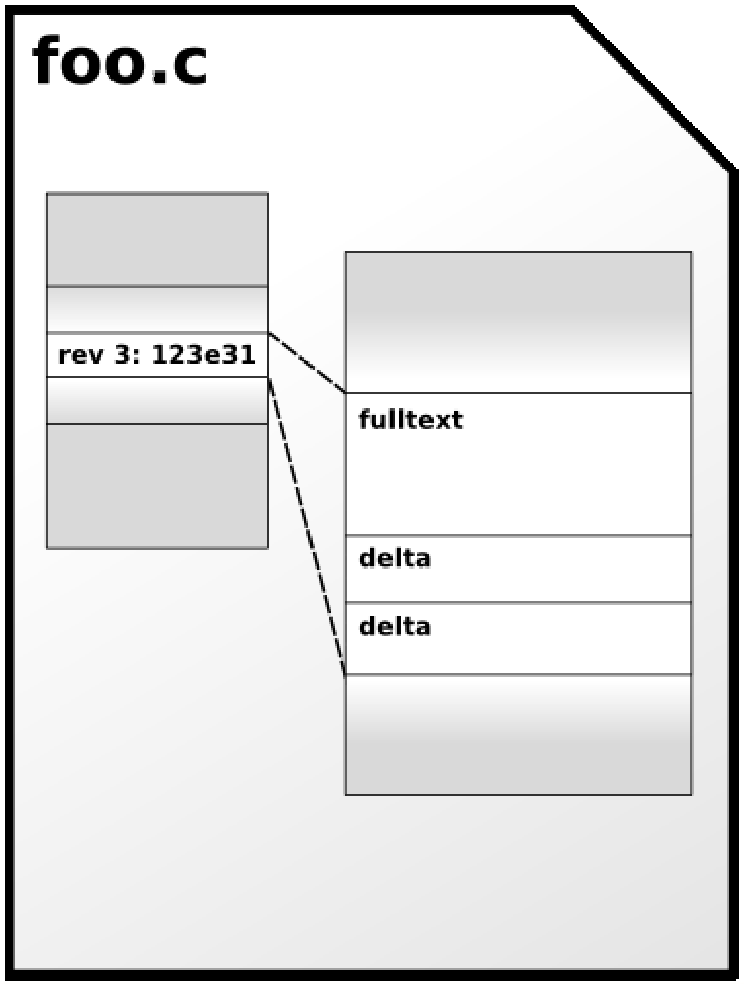
Merging a text edit to a renamed file

elego Software Solutions GmbH

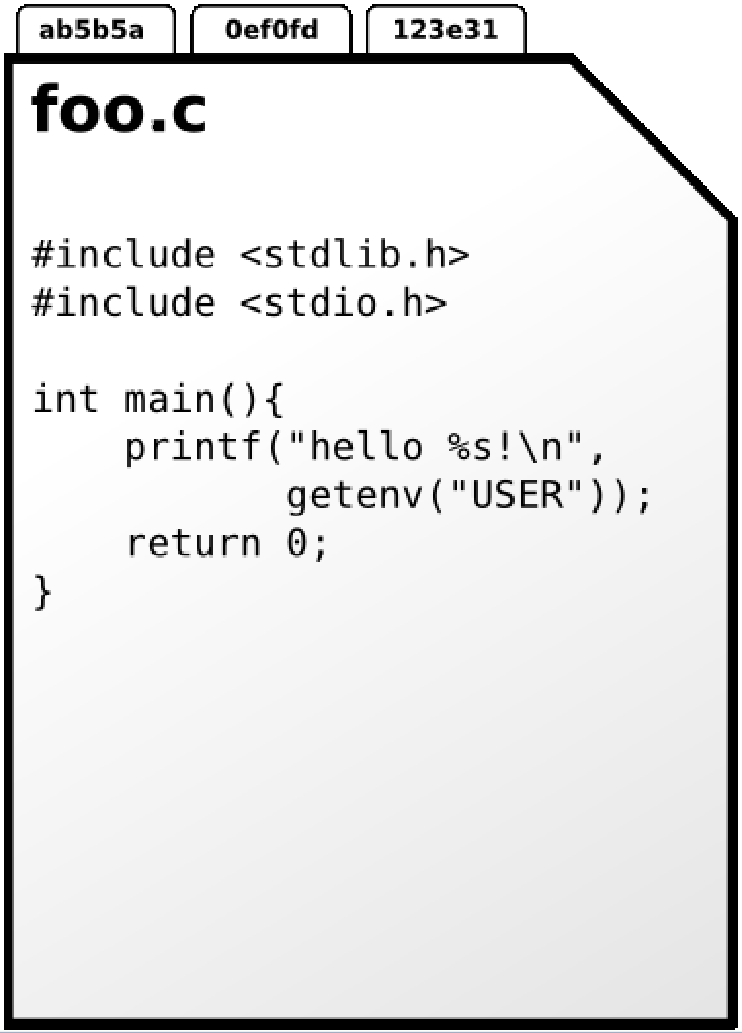
- Mercurial:

```
$ hg pull ../A
pulling from ../A
searching for changes
adding changesets
adding manifests
adding file changes
added 1 changesets with 1 changes to 1 files (+1 heads)
(run 'hg heads' to see heads, 'hg merge' to merge)
$ hg merge
merging bar.c and foo.c to bar.c
0 files updated, 1 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)
$ cat bar.c
#include <stdio.h>
void main() {
        printf("Goodbye world!\n");
}
$
```

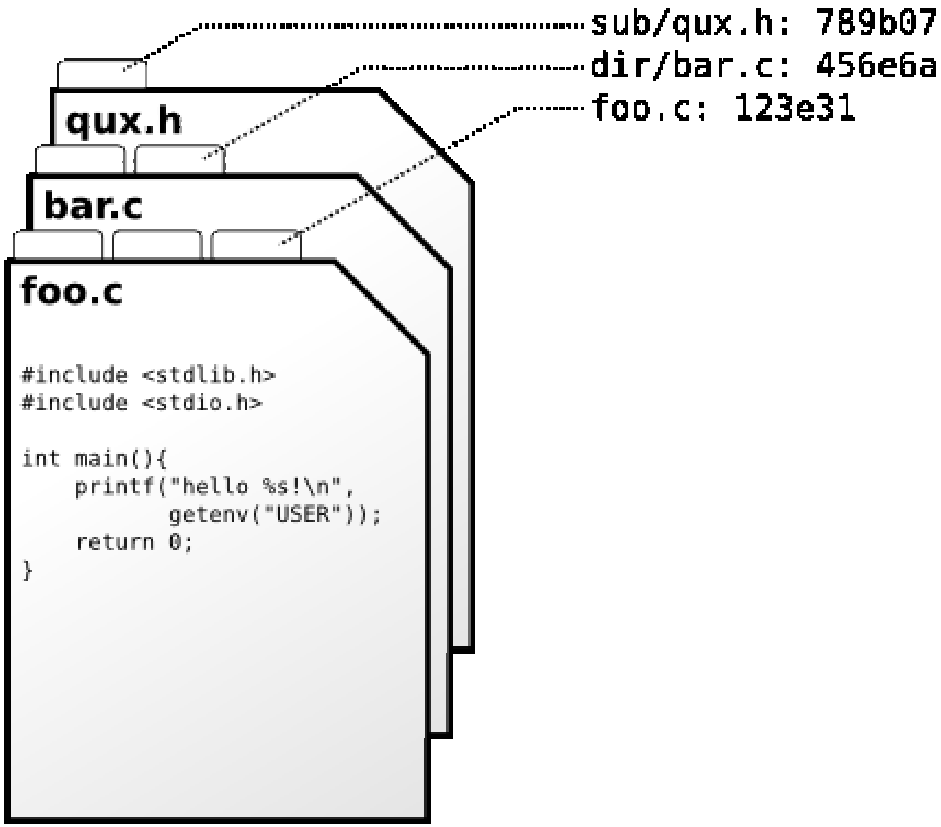
Mercurial: Revlog data structure



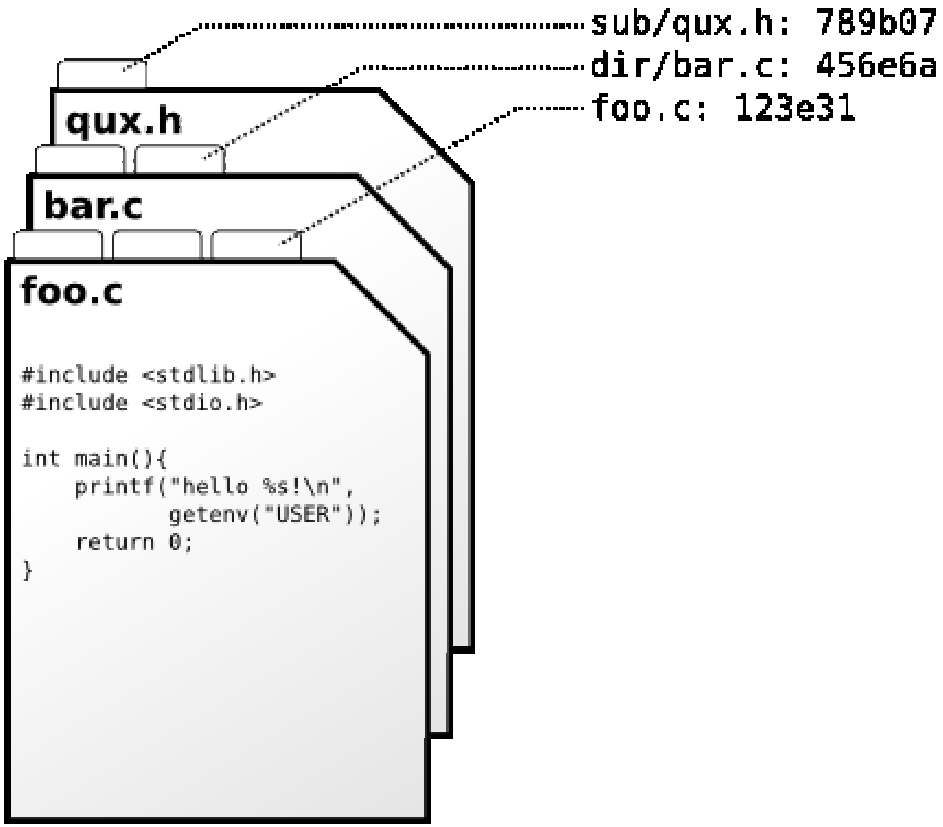
Mercurial: Filelog



Mercurial: Filelog

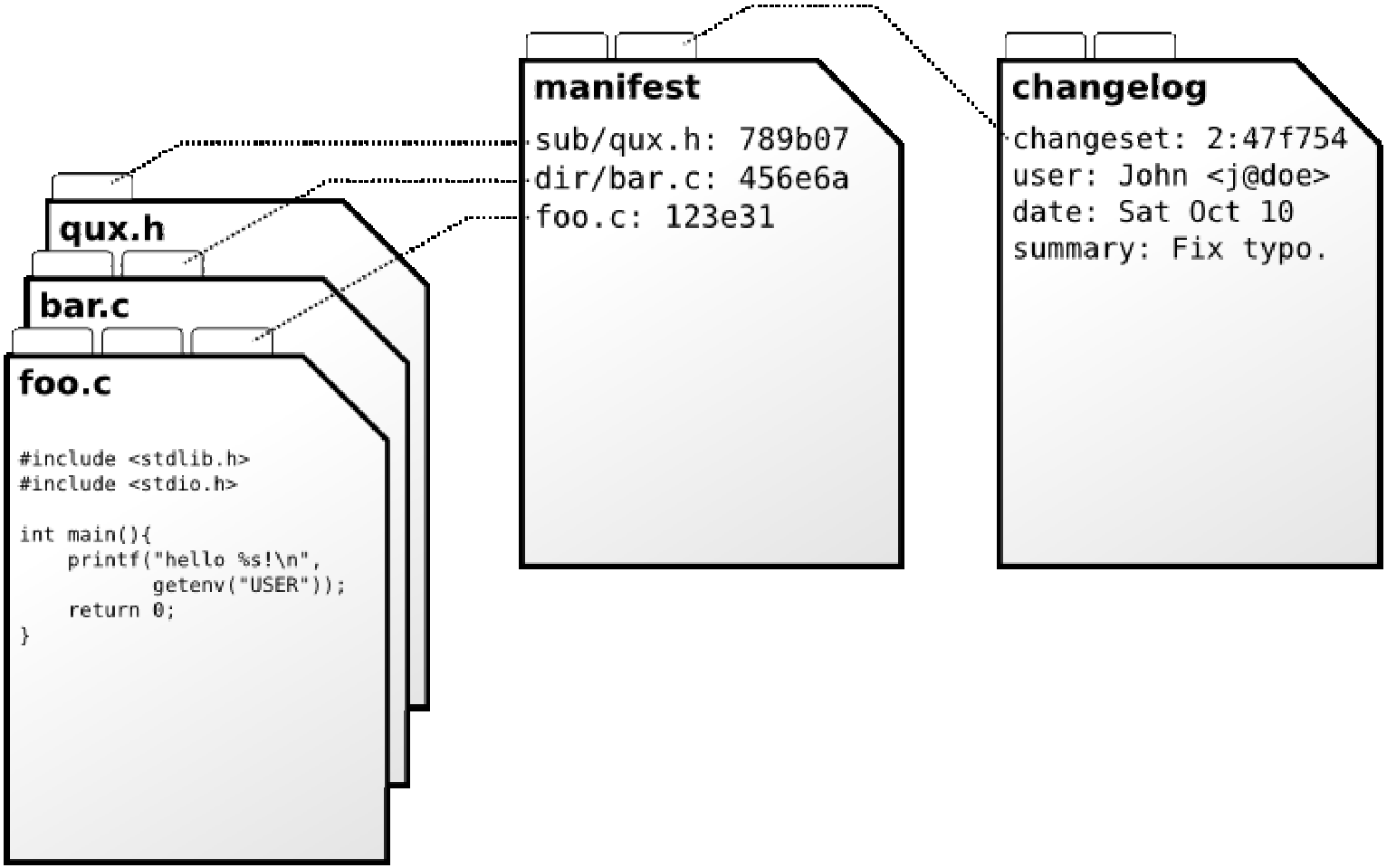


Mercurial: Changeset calculation




```
changeset: 2:47f754
user: John <j@doe>
date: Sat Oct 10
summary: Fix typo.
```

Mercurial: Manifest & changelog



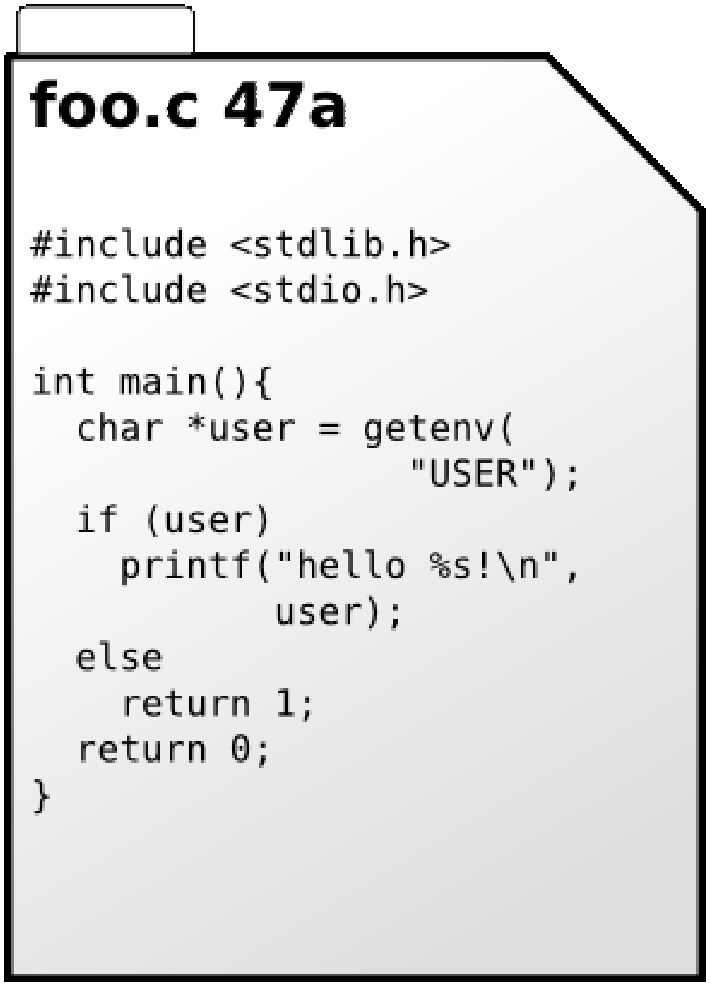
Merging a text edit to a renamed file

elego Software Solutions GmbH

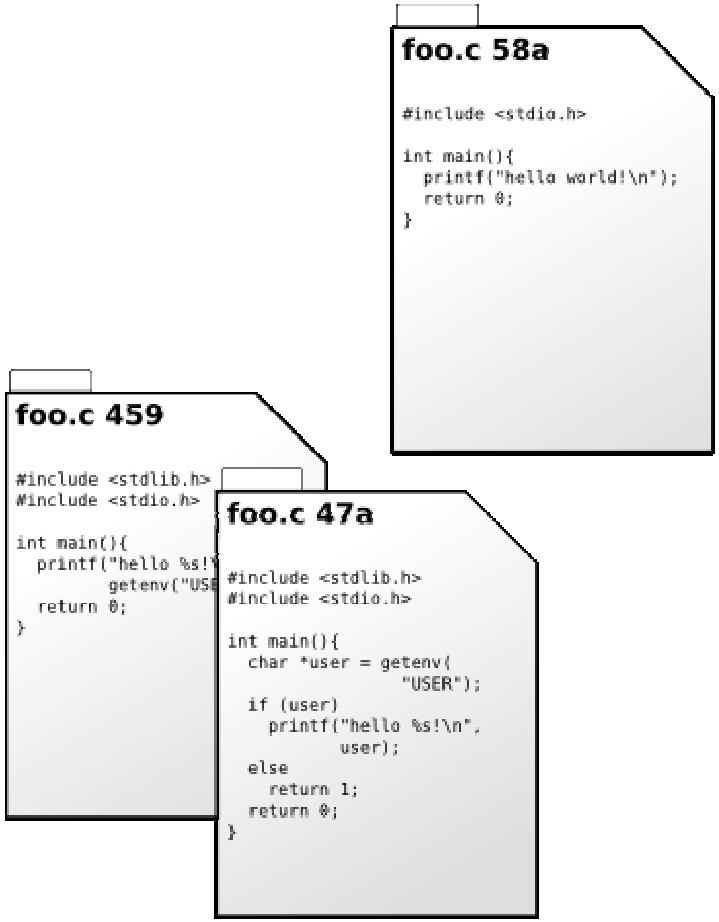
- Git:

```
$ git pull ../A master
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From ../A
 * branch            master     -> FETCH_HEAD
Merge made by recursive.
 bar.c |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)
$ cat bar.c
#include <stdio.h>
void main() {
        printf("Goodbye world!\n");
}
$
```

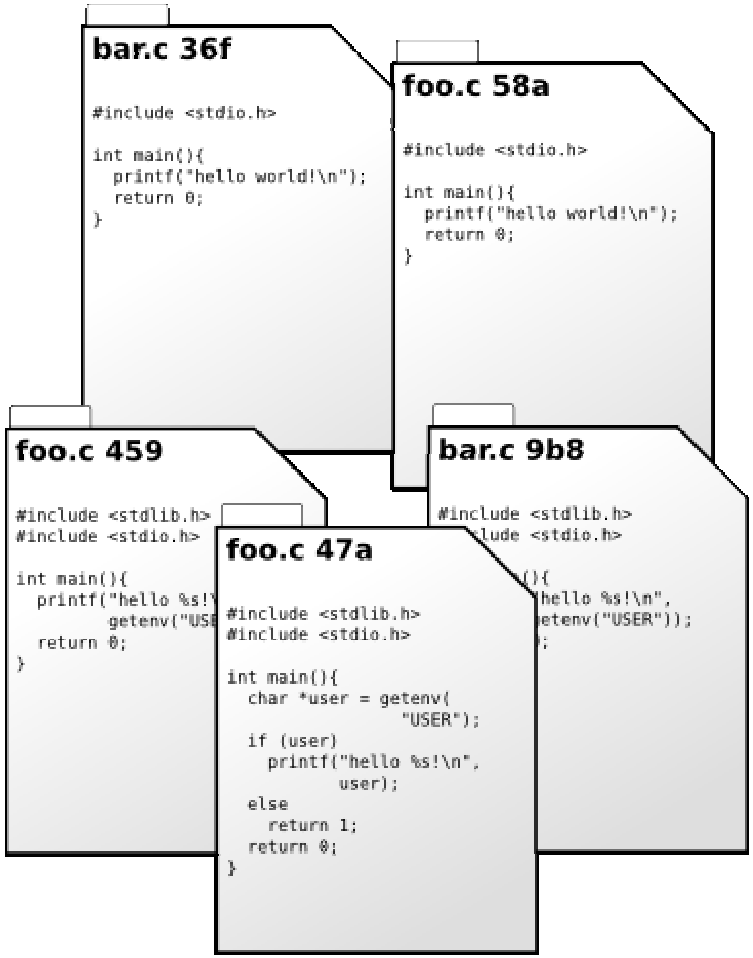
Git: Blob object stores file content



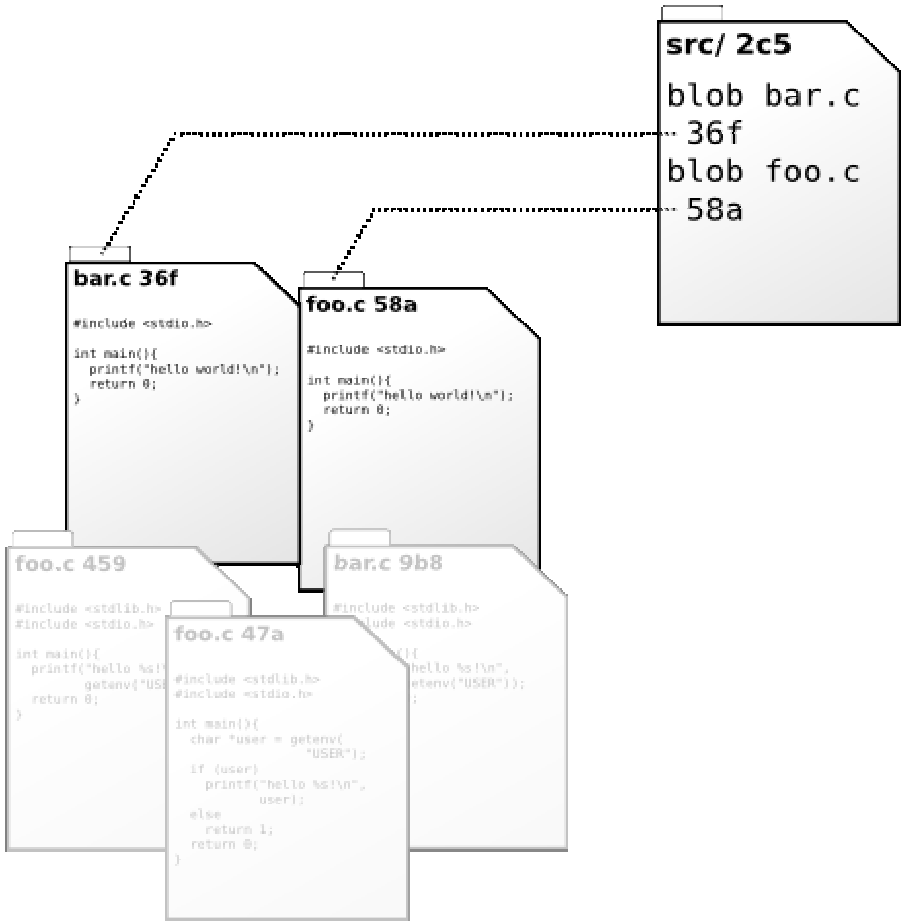
Git: Every revision of a file creates a new blob



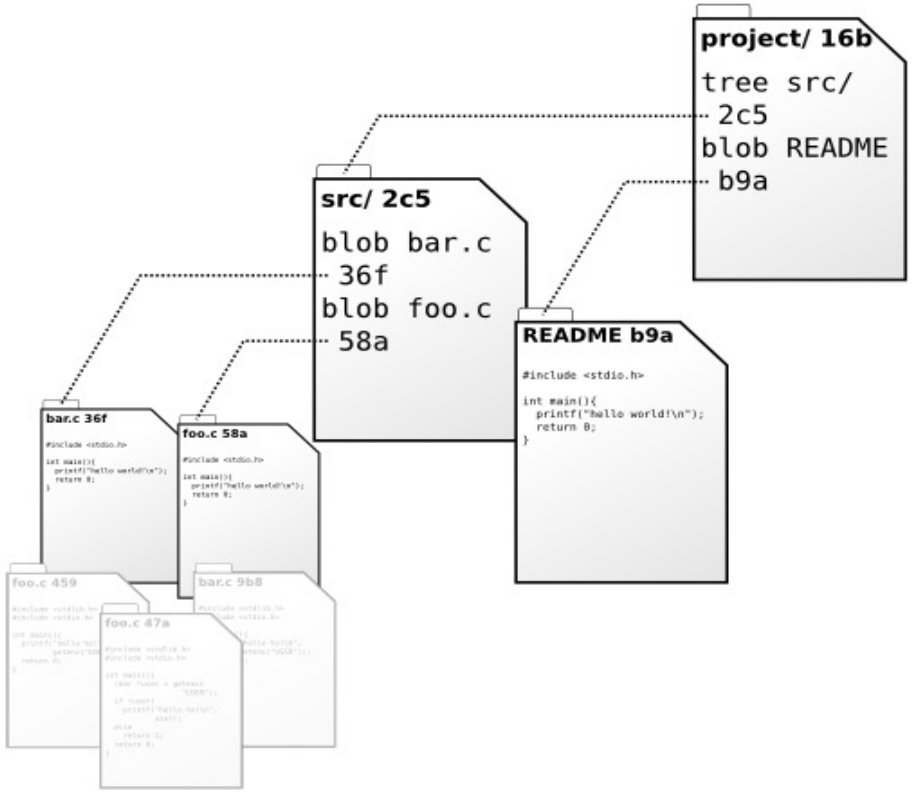
Git: All file blobs are stored together



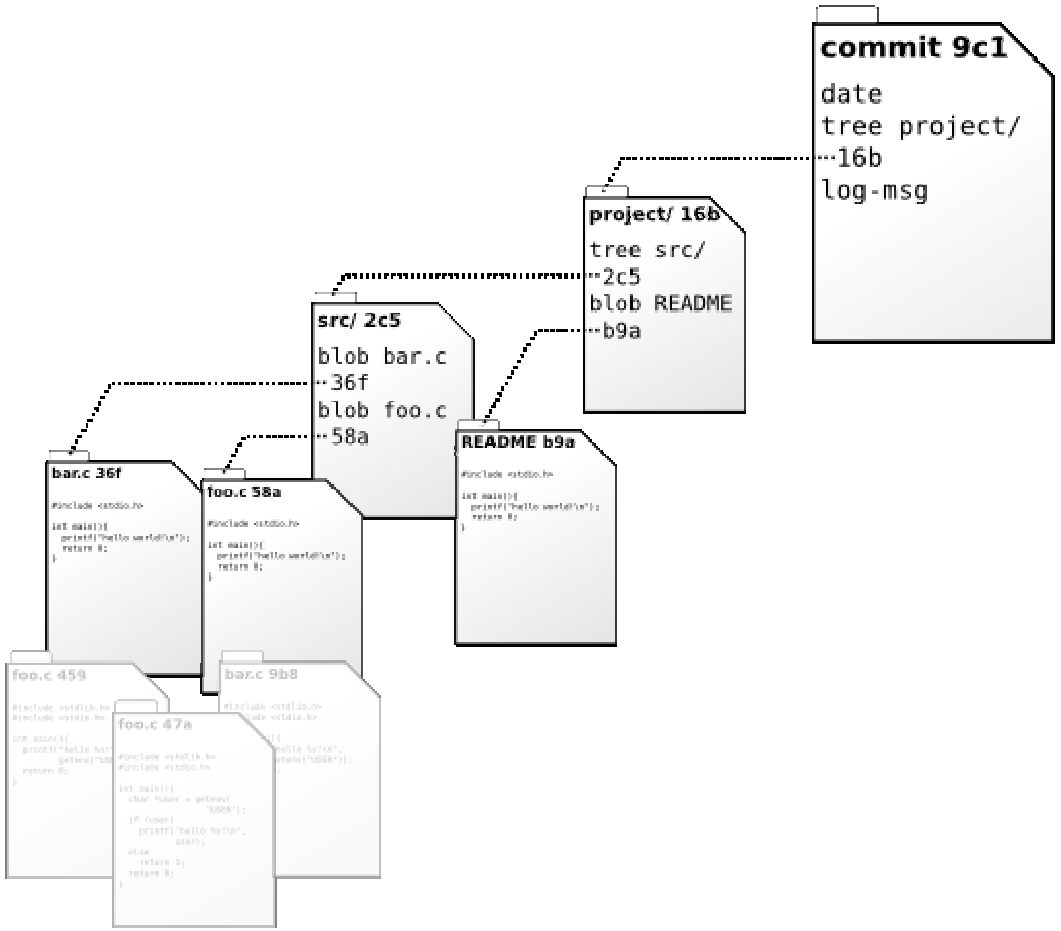
Git: Tree objects represent directories



Git: Tree objects form a hierarchy



Git: Commit object refers to a tree root



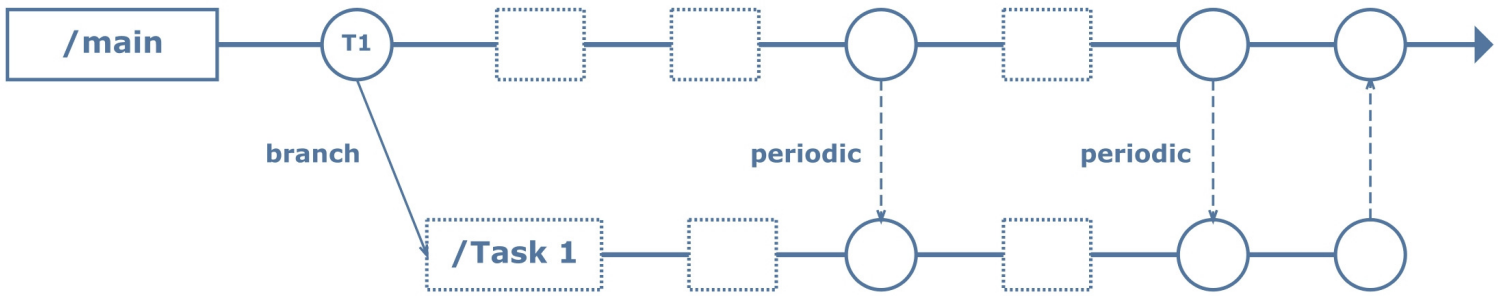
Real-life situations

elego Software Solutions GmbH

1. Reserving a file for editing
   - Centralised vs distributed
2. Merging a text edit to a renamed file
   - How is tree history modeled?
3. Merging a file move
   - Distinguishing a file move from a directory rename

Merging a file move

- Developers! Developers! Developers!
- They like to refactor their code
  - Renaming files
  - Renaming directories



nach Software Configuration Management Patterns Berczuk/Appelton

Merging a file move

- A project starts out as:

```
fruit/Color.java
fruit/Taste.java
fruit/citrus/Apple.java
```

- On branch A, **add** a file to the "citrus" package:

```
fruit/Color.java
fruit/Taste.java
fruit/citrus/Apple.java
fruit/citrus/Orange.java
```

- On branch B, **move** Apple.java to the "pome" package:

```
fruit/Color.java
fruit/Taste.java
fruit/citrus/
fruit/pome/Apple.java
```

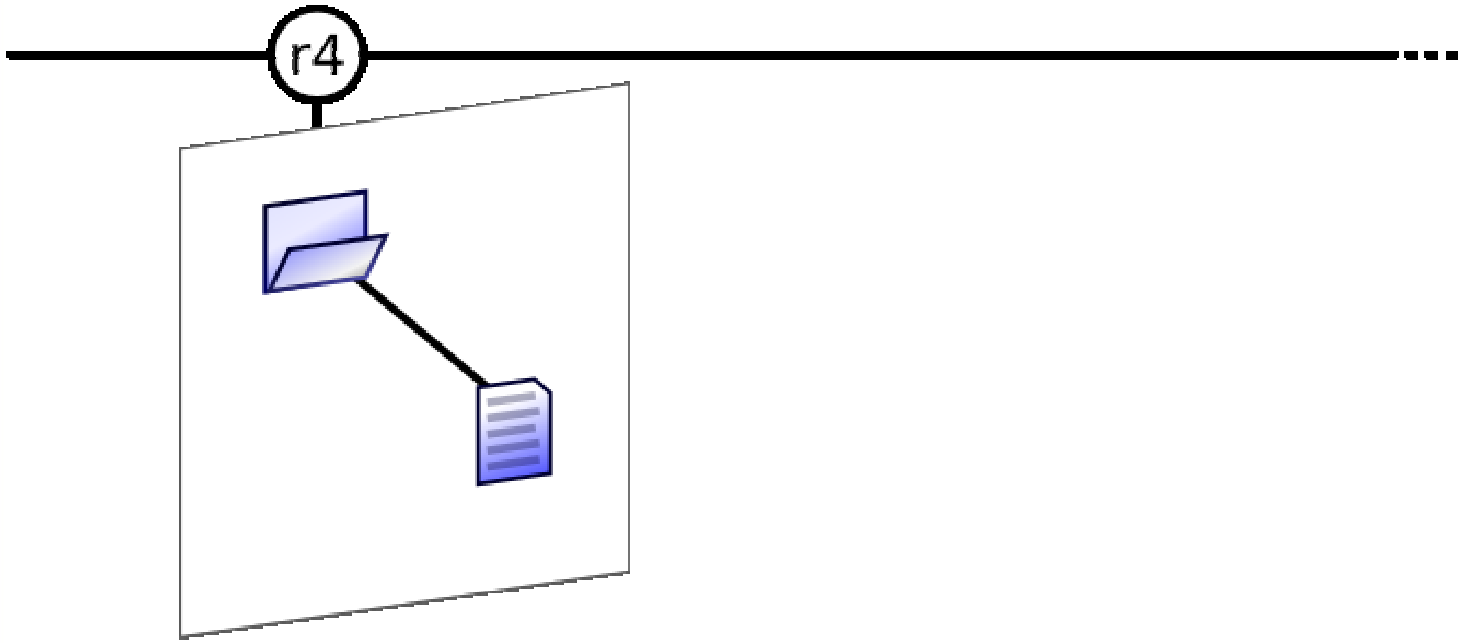
- Merge the change from branch A to branch B

Merging a file move

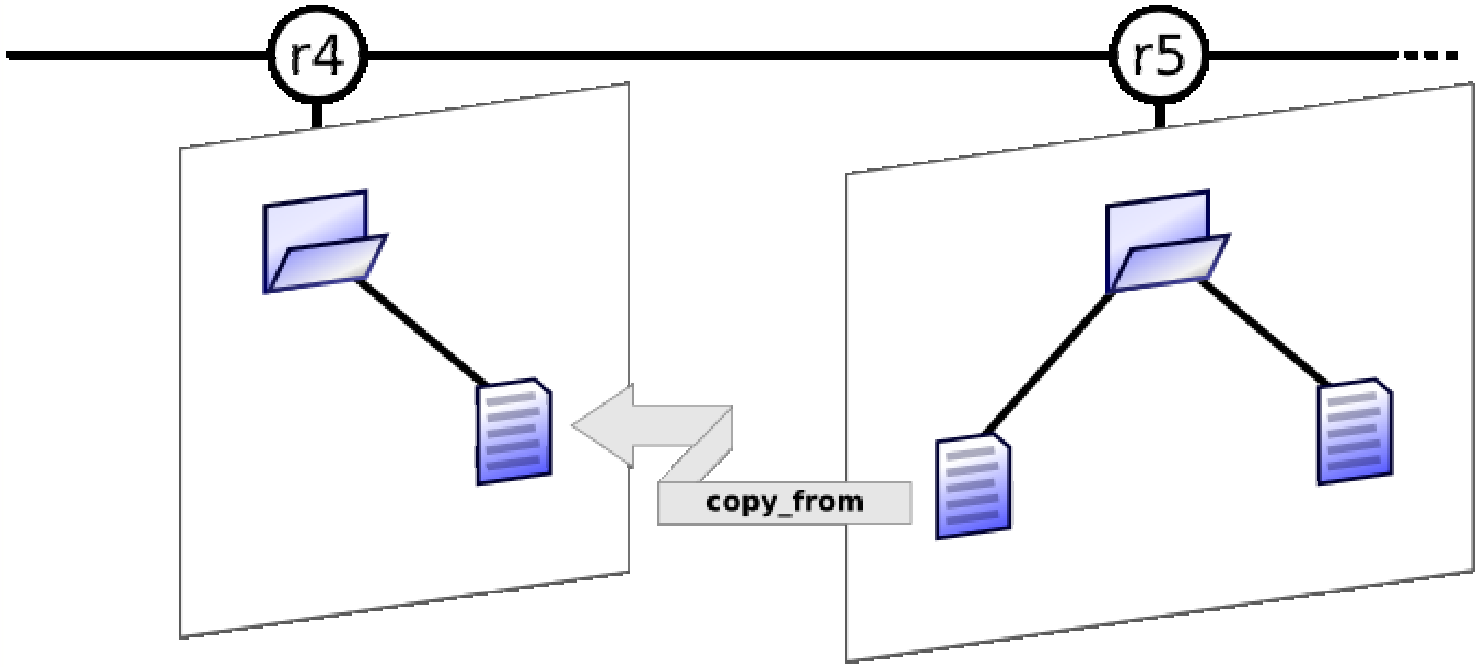
- Subversion:

```
$ svn merge ^/A
--- Merging r3 through r5 into '.':
A    fruit/citrus/Orange.java
$ svn status
 M       .
A  +    fruit/citrus/Orange.java
$ ls fruit/ fruit/citrus/ fruit/pome/
Color.java   Taste.java   citrus/      pome/

fruit/citrus:
Orange.java

fruit/pome:
Apple.java
```

Subversion records the source of a copy



Subversion records the source of a copy



Merging a file move

elego Software Solutions GmbH

- Mercurial:

```
$ hg pull ../A
pulling from ../A
searching for changes
adding changesets
adding manifests
adding file changes
added 1 changesets with 1 changes to 1 files (+1 heads)
(run 'hg heads' to see heads, 'hg merge' to merge)
$ hg merge
1 files updated, 0 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)
$ ls -R fruit/
Color.java   Taste.java   pome/

fruit/pome:
Apple.java   Orange.java
```

Merging a file move

- Git:

```
$ git pull ../A master
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (5/5), done.
From ../A
 * branch            master     -> FETCH_HEAD
Merge made by recursive.
 fruit/citrus/Orange.java |   20 ++++++++++++++++++++
 1 files changed, 20 insertions(+), 0 deletions(-)
 create mode 100644 fruit/citrus/Orange.java
$ ls -R fruit/
Color.java   Taste.java   citrus/      pome/

fruit/citrus:
Orange.java

fruit/pome:
Apple.java
$
```

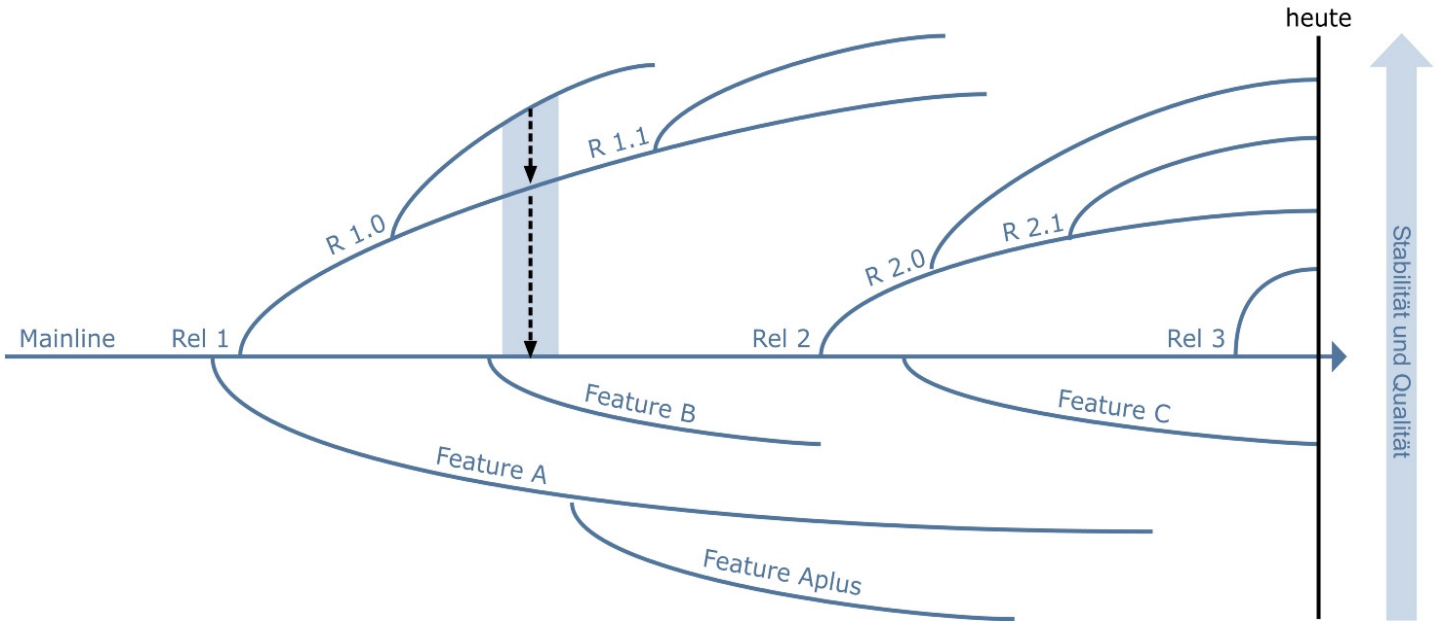
Real-life situations

elego Software Solutions GmbH

1. Reserving a file for editing
   - Centralised vs distributed
2. Merging a text edit to a renamed file
   - How is tree history modeled?
3. Merging a file move
   - Distinguishing file move from directory rename
4. Cherry-picking a revision
   - Fine-grained merge tracking

Cherry-picking a revision

- Pick a revision from a branch's history and merge it to the HEAD of another branch.



L. Wingard 2005 - Flow of Change

Cherry-picking a revision

elego Software Solutions GmbH

- Release branches need to receive bug fixes
- Two approaches:
  - Make fix in main line, then merge into release branch
  - Make fix in release branch, then merge into main line
- Ideally, the merge should be tracked
- Which approach works best for
  - Subversion?
  - Mercurial?
  - Git?

Why should I care about the design of my SCM tool?

elego Software Solutions GmbH

- Design defines constraints
- Your development process must adapt to constraints
- Fixing bugs is easy, fixing design is hard

Thank you!

Questions?