SoftwareEntwicklung ▸ Beratung ▸ Schulung

Bringing Subversion to the next Level

Introduction, Concepts and usage of SVK

a distributed version control tool

Web Site:
www.soebes.com
Blog:
blog.soebes.com
Email:
info@soebes.com

Dipl.Ing.(FH) Karl Heinz Marbaise

Agenda

1. History
2. Installation
3. Features
4. Topology
5. The Working Cycle
6. Mirroring with SVK
7. Use Cases
8. Differences to SVN
9. Pro's and Con's
10. Tips and tricks

1. History

- Developped in 2003 by Chia Liang Kao during his sabbatical year.

		SVN	SVK
Feb.	2004	1.0.0	
May	2005		1.0.0
Sept.	2006	1.4.0	
Dec.	2006		2.0.0
June	2007	1.4.4	
July	2007		2.0.2

2. Installation

- Download the appropriate installation package
  - Windows
    - svk-v2.0.2-MSWin32-x86.exe
  - Unix(Linux)
    - SVK-v2.0.2.tar.gz
  - Mac
    - SVK-v2.0.2-5.dmg

3. Features Overview

- First of all:
  - All features which are part of Subversion are part of SVK as well.
    - Atomic commits, directory versioning, versioned metadata, efficient branching and tagging etc.

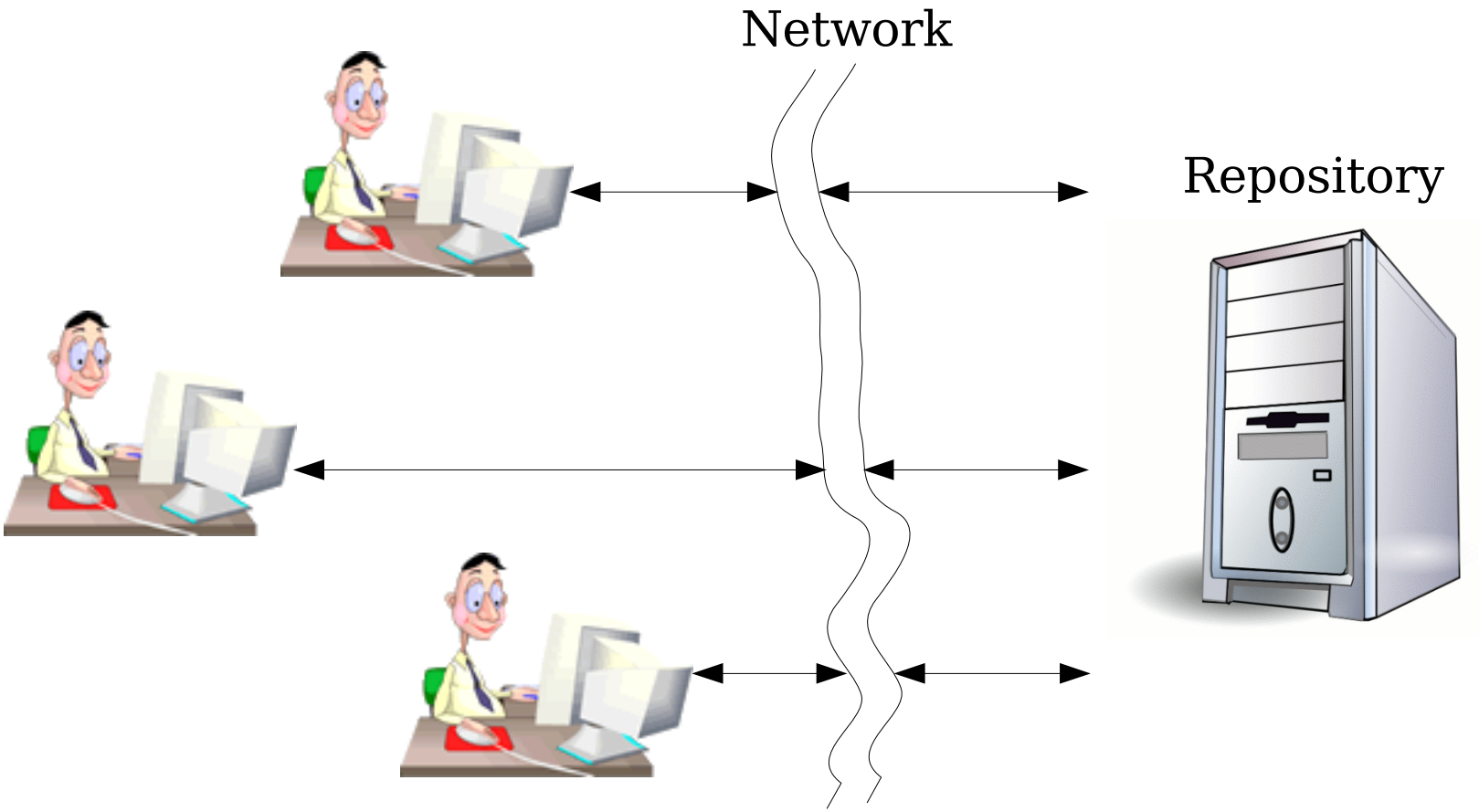
3. Features Unsupported

- But there are a few exceptions:
  - No support of „svn:externals“
    - use views instead (currently experimental).
  - No lock/unlock commands.
  - No support of complex tags (sometimes called shelving).

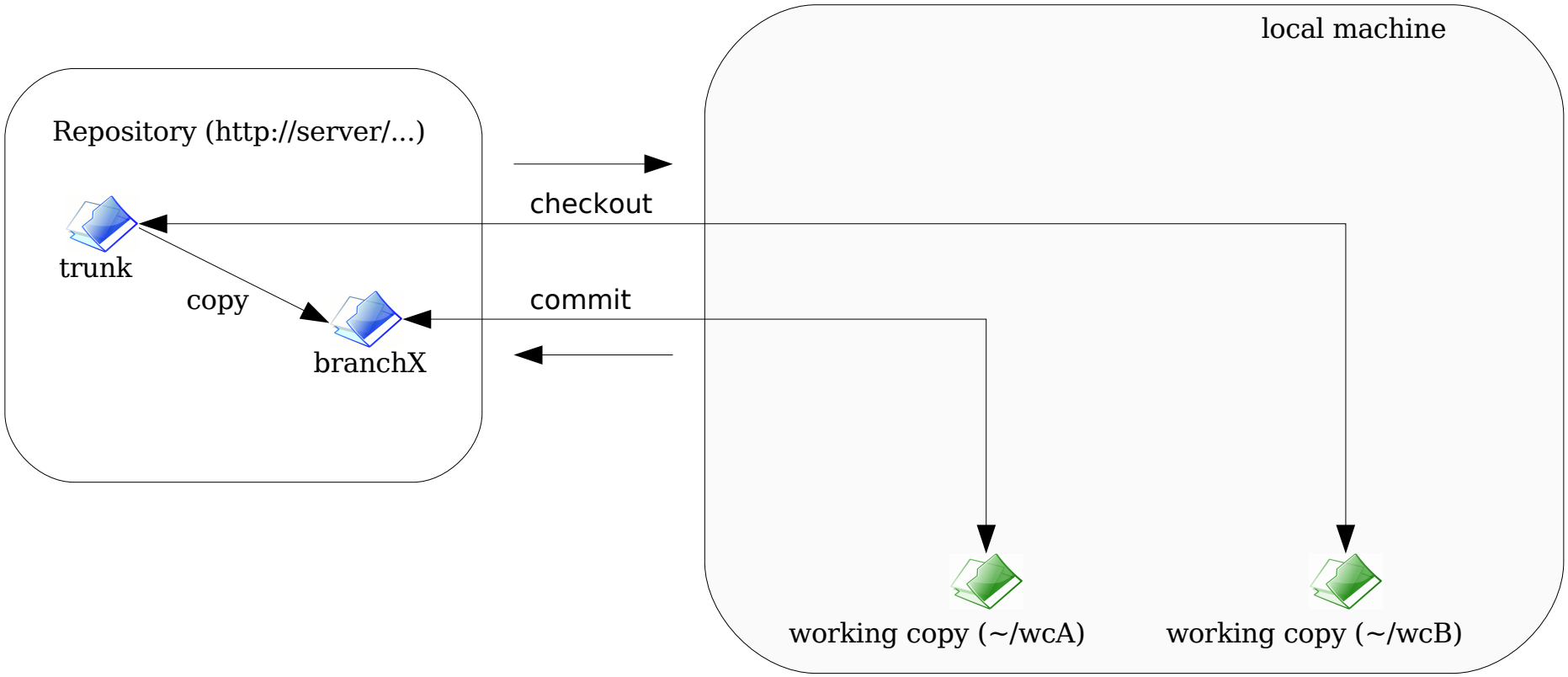
3. Features Supplemental

- But there are supplemental features:
  - Merge tracking
  - Disconnected operation
  - Decentralized operation

4. Topology Subversion



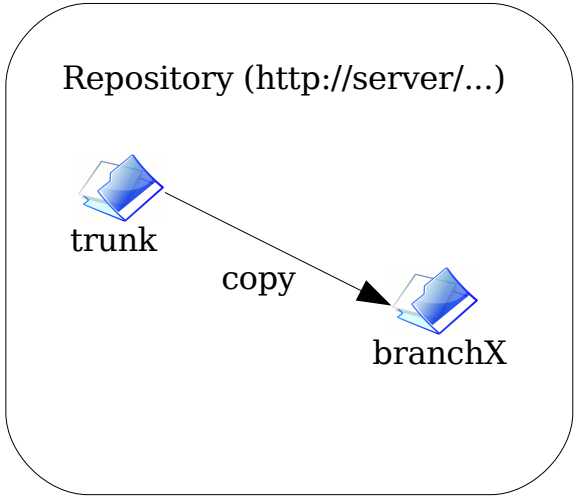
4. Topology Subversion

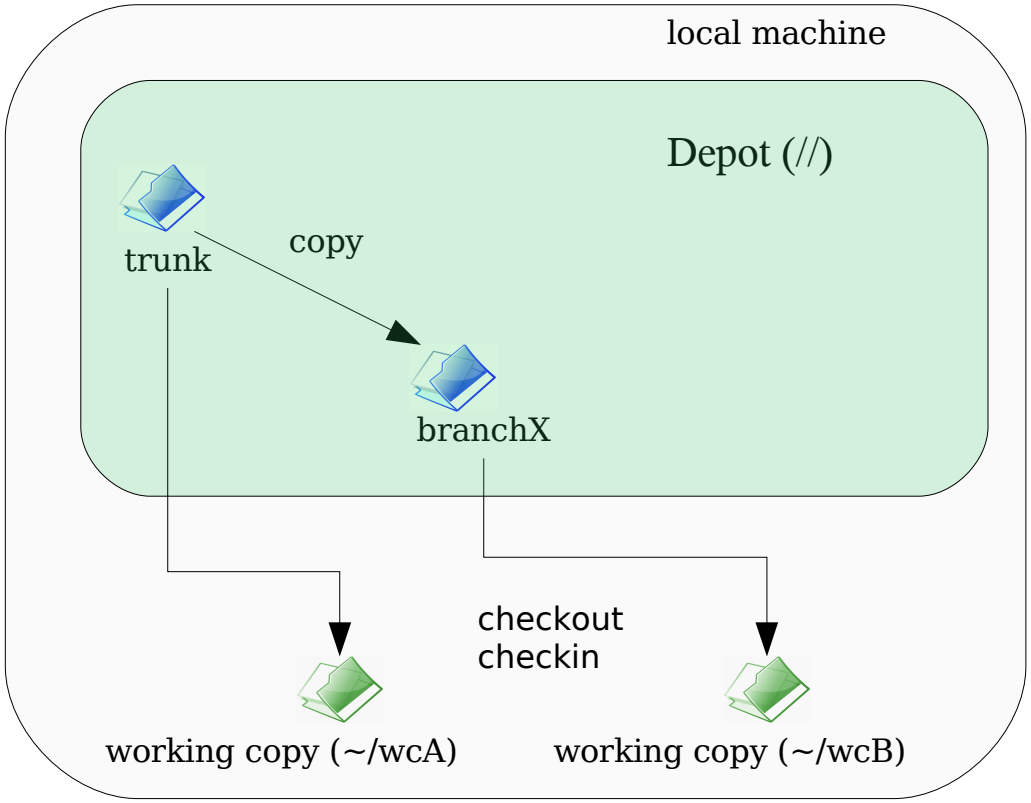


4. Topology SVK

- In SVK you have a thing which is called depot. It is indicated by „//“:
  - A depot is a Subversion repository which will be accessed via „file://“ protocol.
  - SVK is doing its work via Subversion Perl bindings.

4. Topology SVK





5. The working cycle Preparation

- Initialize the depot storage.

```
svk depotmap --init
Repository ~/.svk/local does
not exist, create? (y/n)
```

5. The working cycle Import a project

- Import a project into the depot:

```
1 svk import
2   /test/jagosi
3   //project/trunk
4   -m “Log Message”
```

```
Committed revision 1.
Import path //jagosi/trunk initialized.
Committed revision 2.
Directory C:\test\jagosi imported to depotpath
//jagosi/trunk as revision 2.
```

5. The working cycle
Start working

- In Subversion you have to checkout first from the repository, whereas in SVK from the depot to get a working copy.

```
1 svk checkout
2   //project/trunk
3   /testsvn/project-trunk
```

5. The working cycle The SVK Commands

- The commands of SVK are similar to the Subversion commands:
  - commit, diff, status, proplist, list, log, mkdir, move, resolved, revert, switch, info, copy, add, cat, annotate

5. The working cycle Branching

- A branch in SVK will be created as the same way as in Subversion via „copy“.

```
1
2  svk copy
3     //project/trunk
4     //project/branches/B_1
      -m “Message”
```

5. The working cycle
The simple merge

- The merge command:

```
1  svk merge
2     -r RevFrom:RevTo
3     //project/branches/B_1
```

5. The working cycle
The simple merge

- No merge tracking as in Subversion.
- This will change the working copy the same way as in Subversion. You have to commit your working copy to make the merge permanent.

5. The working cycle
The comfortable way

- The **s**tar-**merge** command:

```
1  svk smerge
2     //project/branches/B_1
3     //project/trunk
4     -m “Merging branch“
```

5. The working cycle The comfortable way

- This will merge the whole contents of the branch with a **merge ticket** in a single revision.
- This will happen within the depot and **NOT** within the working copy. If you like to see the changes made by the merge in your working copy you have to make an update.

5. The working cycle The comfortable way

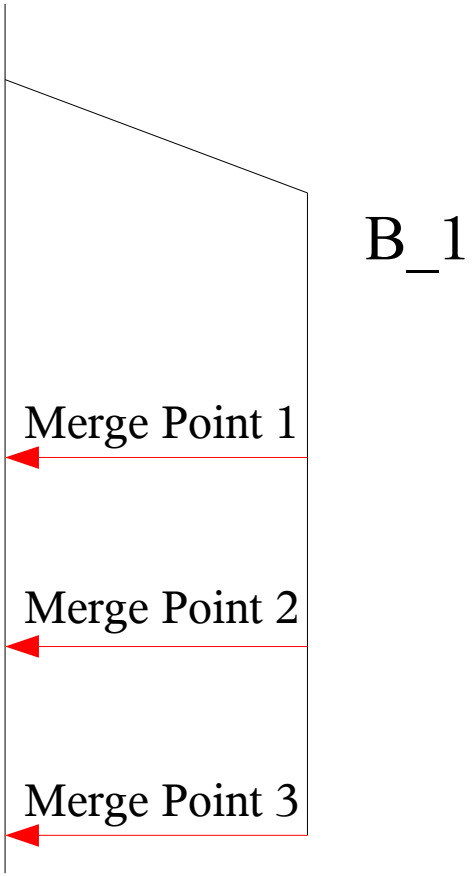
- If you like the merge to happen within the working copy you have to select a different target.

```
1  svk smerge
2    //project/branches/B_1
3    .
4    -m “Merging branch“
```

5. The working cycle
The comfortable way

- If you like to continue working on the branch just do it.
- If you like to merge the changes from the branch simply give the following command:

```
svk smerge
   //project/branches/B_1
   //project/trunk
   -m “Merging branch“
```



6. Mirroring with SVK
Problem of Subversion

Repository

6. Mirroring with SVK Problem of Subversion

- For many operations e.g. commit, check out, tagging, branching you need a network connection.
- Exception:
  - Local repository access via file:// protocol.

6. Mirroring with SVK Create a mirror

- You can mirror any existing Subversion development line via:

```
1  svk mirror
2    http://server/project/trunk
3    //project/remote
```

6. Mirroring with SVK Synchronizing

- Now you need to synchronize the mirror with remote repository.

```
1  svk sync
2     //project/remote
```

- This can take a while ;-)

6. Mirroring with SVK Synchronizing

- Output of the “svk sync” command:

```
kama@traveler:~/test> svk sync //test/remote
Syncing http://svn.traveler/test/trunk
Retrieving log information from 1 to 8
Committed revision 2 from revision 1.
Committed revision 3 from revision 2.
Committed revision 4 from revision 3.
Committed revision 5 from revision 4.
Committed revision 6 from revision 5.
Committed revision 7 from revision 6.
Committed revision 8 from revision 7.
Committed revision 9 from revision 8.
```

6. Mirroring with SVK On-line working

- Working on-line with the remote repository.

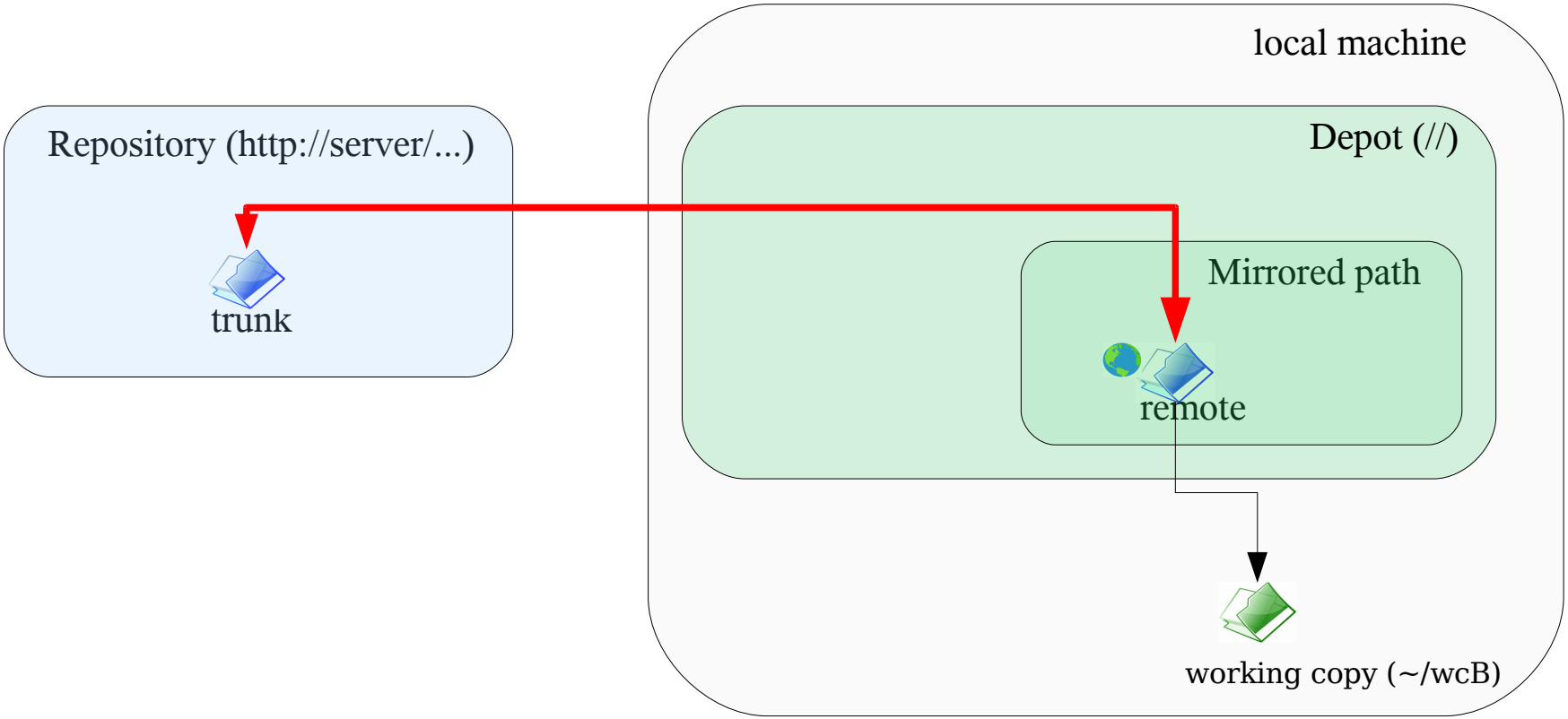
1. `svk checkout`
2. `//project/remote`
3. `/home/kama/remote`

6. Mirroring with SVK On-line working

- Every commit in this working copy is replicated to the remote repository immediately.

```
Commit into mirrored path: merging back directly.
Merging back to mirror source
http://server/test/trunk.
Merge back committed as revision 4.
Syncing http://server/test/trunk
Retrieving log information from 4 to 4
Committed revision 25 from revision 4.
```

6. Mirroring with SVK On-line working



6. Mirroring with SVK Off-line working

- If you want to work **off-line** you have to create a **local** branch.

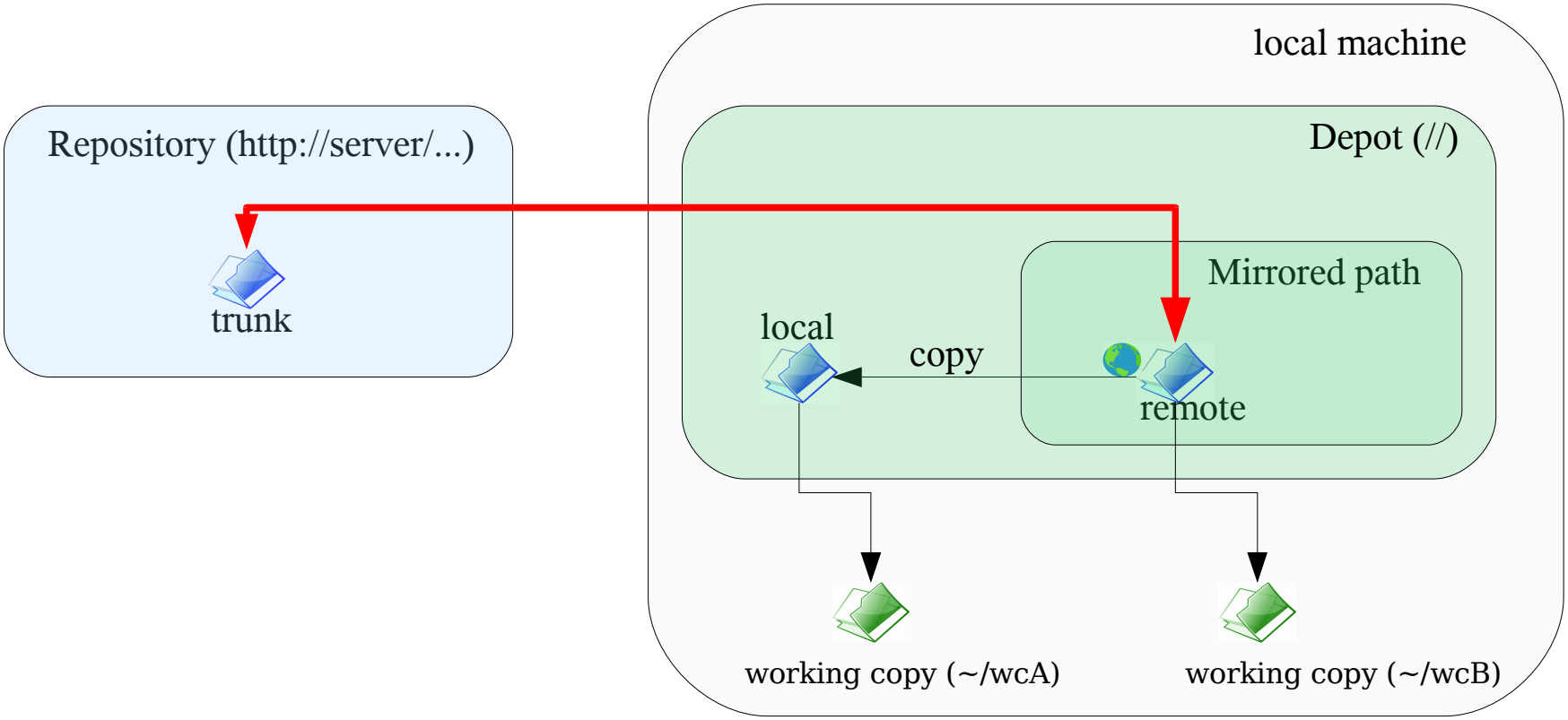
```
1  svk copy
2     //project/remote
3     //project/local
4     -m"- Create local branch"
```

6. Mirroring with SVK Off-line working

- Checkout a working copy which can be used to work off-line:

1. `svk checkout`
2. `//project/local`
3. `/home/kama/local`

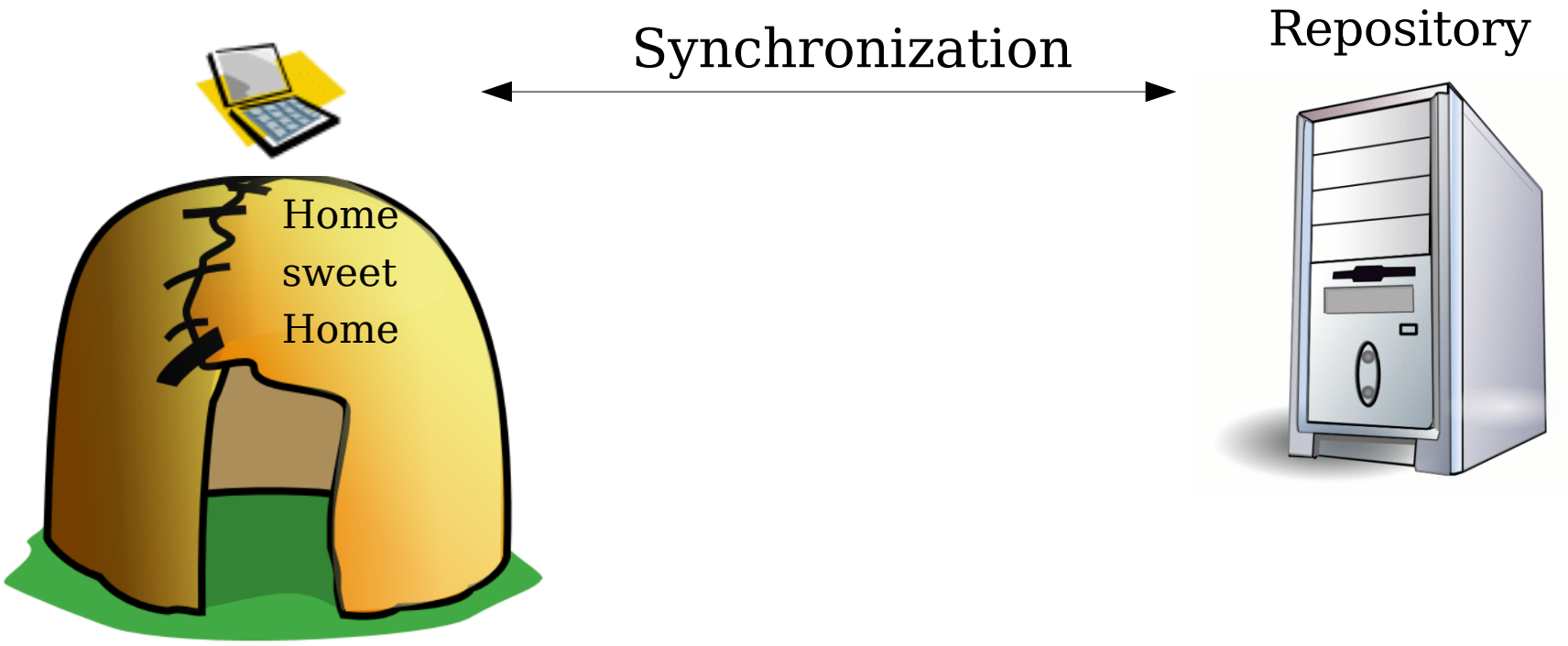
6. Mirroring with SVK Off-line working



6. Mirroring with SVK Off-line working

- Every commit in this working copy will only be done into the local copy of the mirror.
- This means you can continue your work on a project without losing any information about the history of your commits.

6. Mirroring with SVK Synchronization



6. Mirroring with SVK Synchronization

- You can synchronize all your local changes to the remote repository within a single transaction via:

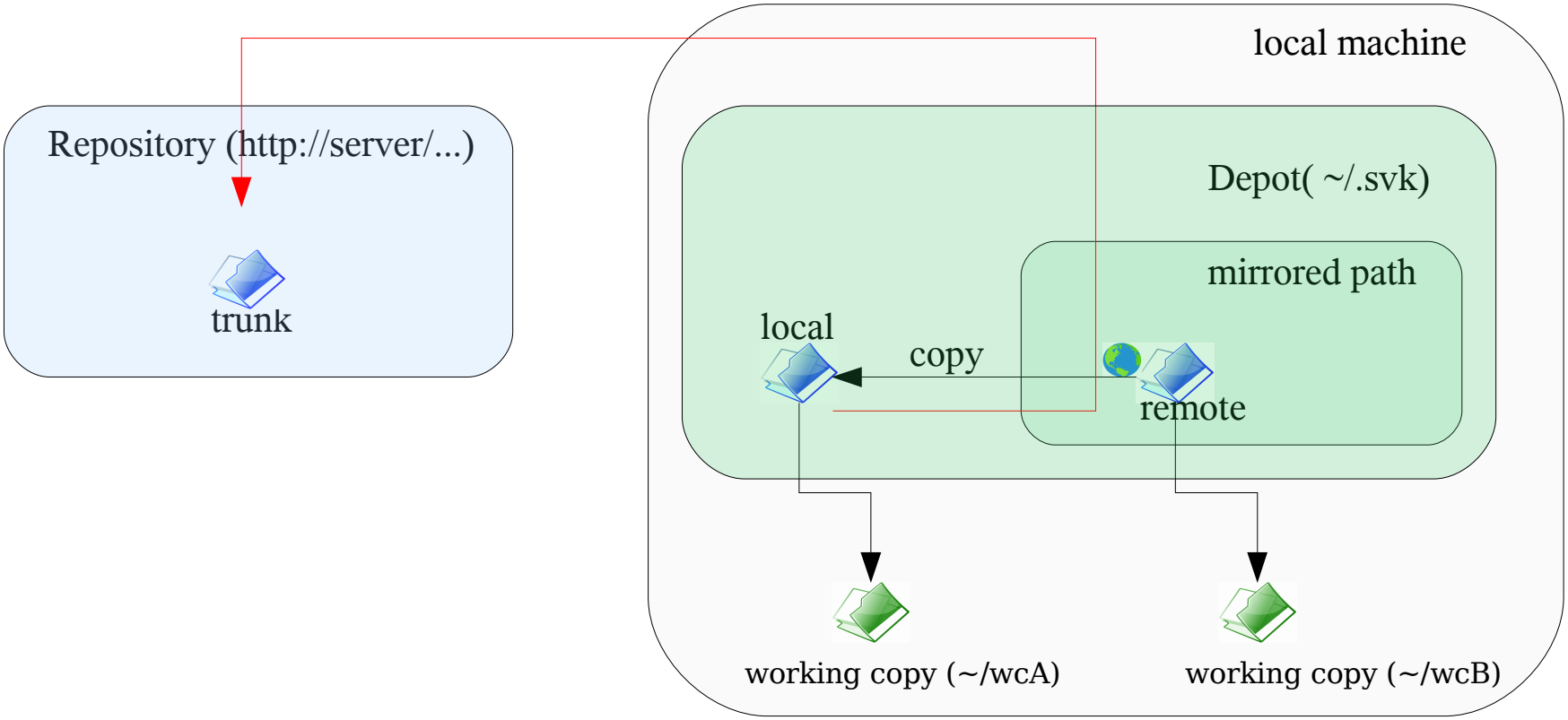
```
1  svk smerge
2    //project/local
3    //project/remote
4    -m"- Merge message"
```

6. Mirroring with SVK Synchronization

- You can synchronize all your local changes to the remote repository as if they were given to the remote repository via:

```
1  svk smerge
2    --incremental
3    --log
4    //project/local
5    //project/remote
```

6. Mirroring with SVK Synchronization



6. Mirroring with SVK Synchronization

- The log messages on remote repository:

```
------------------------------------------------------------------------
r8 | kama | 2007-09-27 09:51:23 +0200 (Do, 27 Sep 2007) | 3 lines

 r33@traveler:  kama | 2007-09-27 09:51:00 +0200
 Change 4

------------------------------------------------------------------------
r7 | kama | 2007-09-27 09:51:22 +0200 (Do, 27 Sep 2007) | 3 lines

 r32@traveler:  kama | 2007-09-27 09:50:51 +0200
 Change 3
```

6. Mirroring with SVK Synchronization

- If you don't like the indention of the log messages and the supplemental information you can use an particular option to the smerge command:

  --verbatim

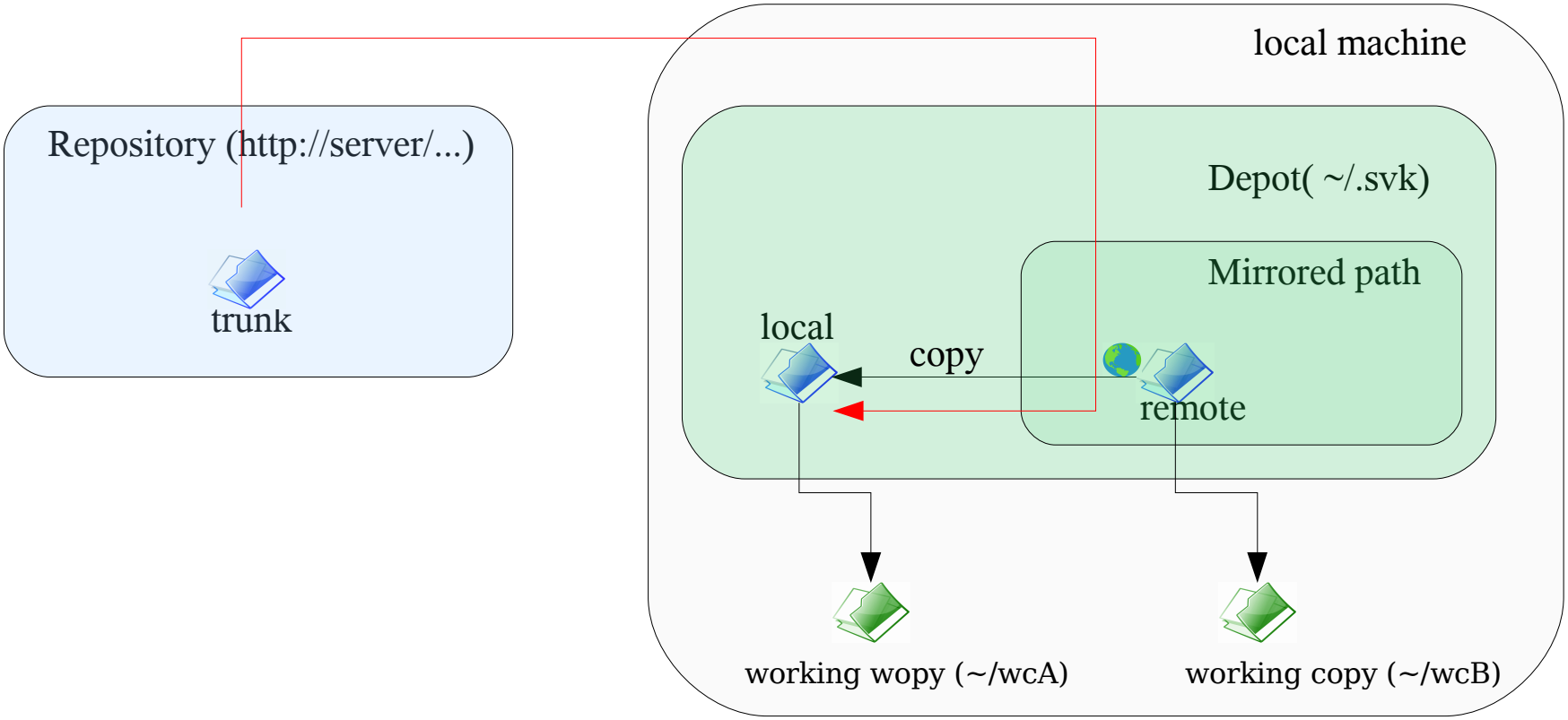
  This will result in log messages which you can't distinguish from local commits.

6. Mirroring with SVK Synchronization

- What happens if someone is changing the mirrored line?

```
1  svk smerge
2    --sync
3    --incremental
4    --log
5    //project/remote
6    //project/local
```

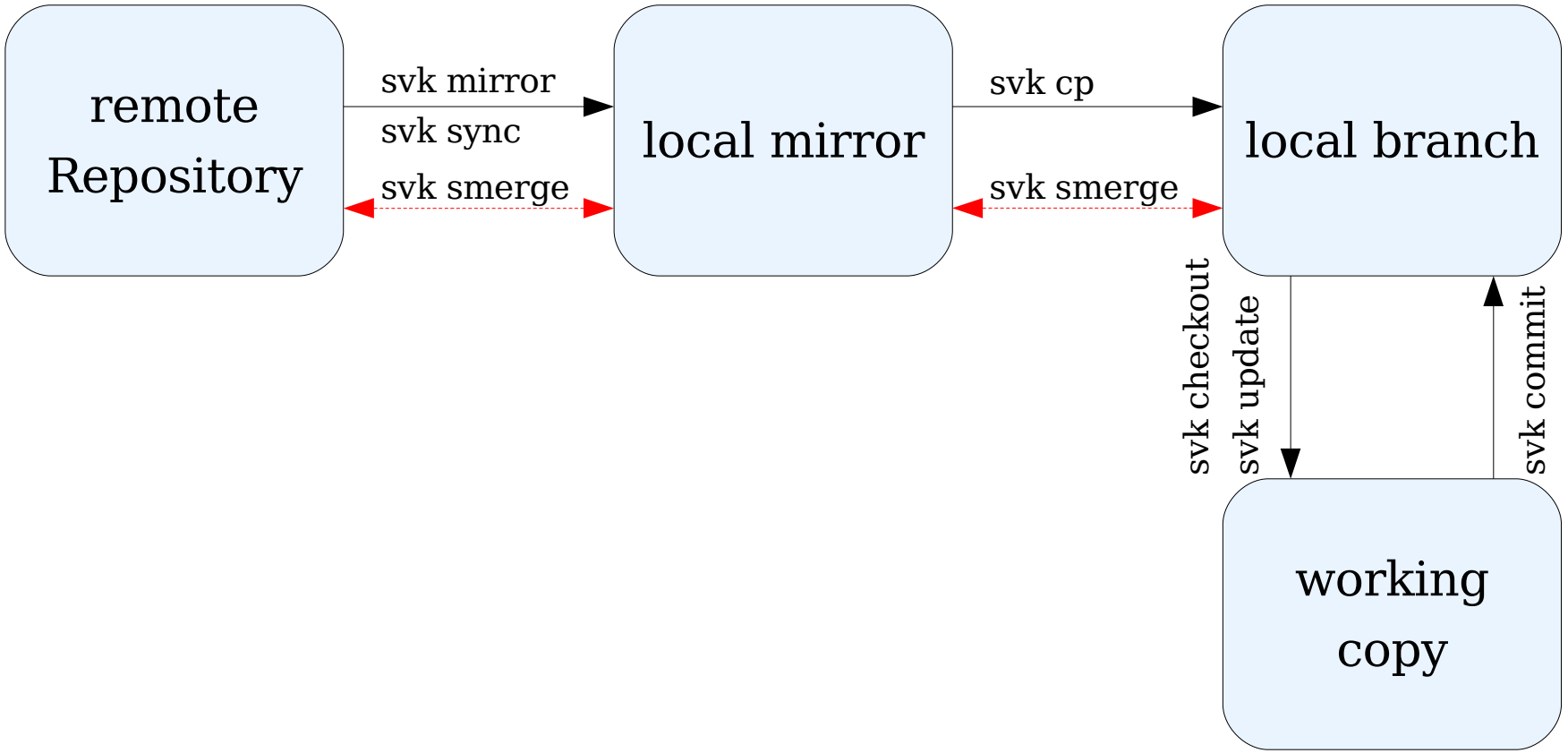
6. Mirroring with SVK Synchronization



6. Mirroring with SVK Synchronization

- If you want to update your working copy you have to give an extra **svk update** within your working copy.

6. Mirroring with SVK Work Flow



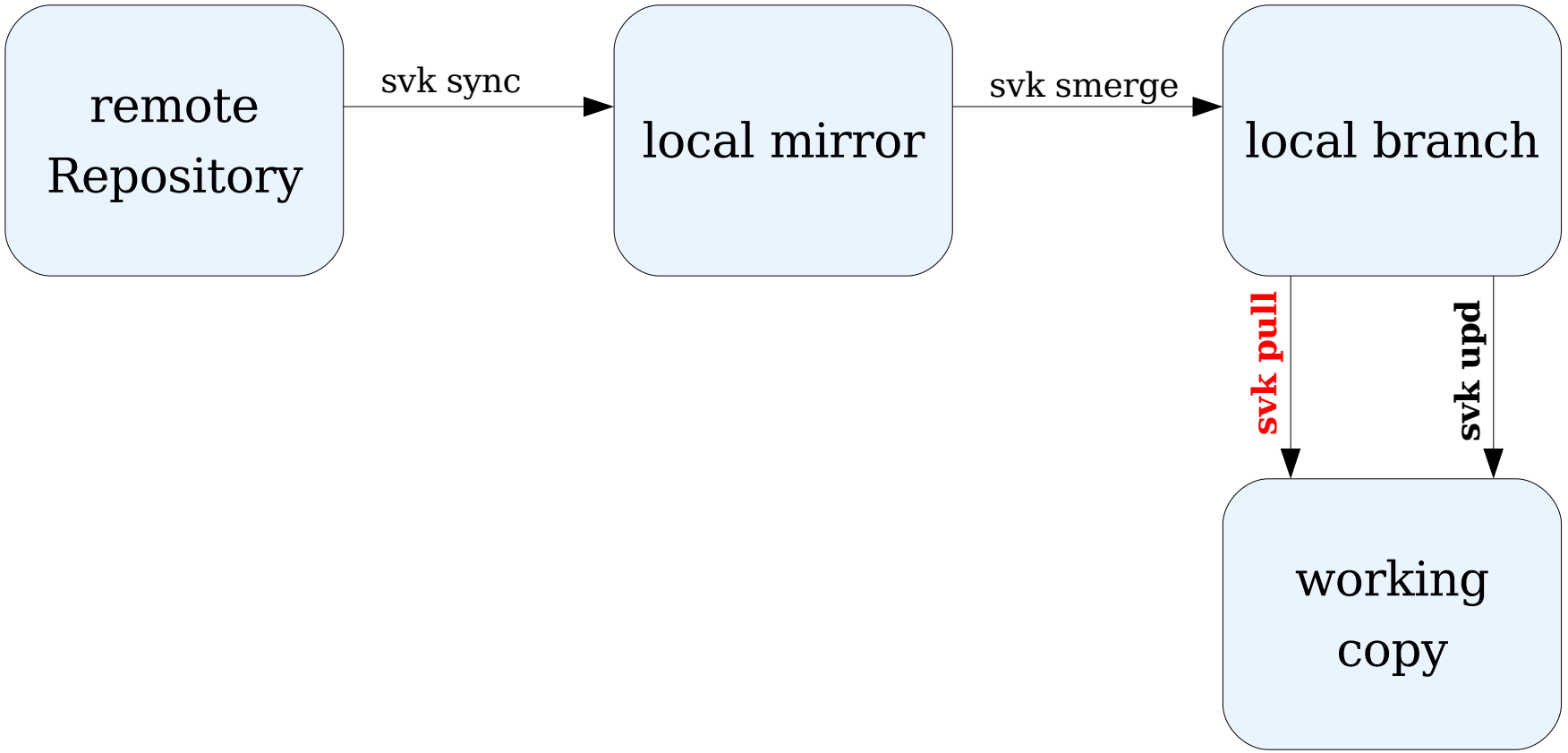
6. Mirroring with SVK Synchronization

- If you want to synchronize all in one single step just use:

```
1  svk pull
2     .
```

- This will synchronize only in one single transactions.

6. Mirroring with SVK Work Flow for svk pull



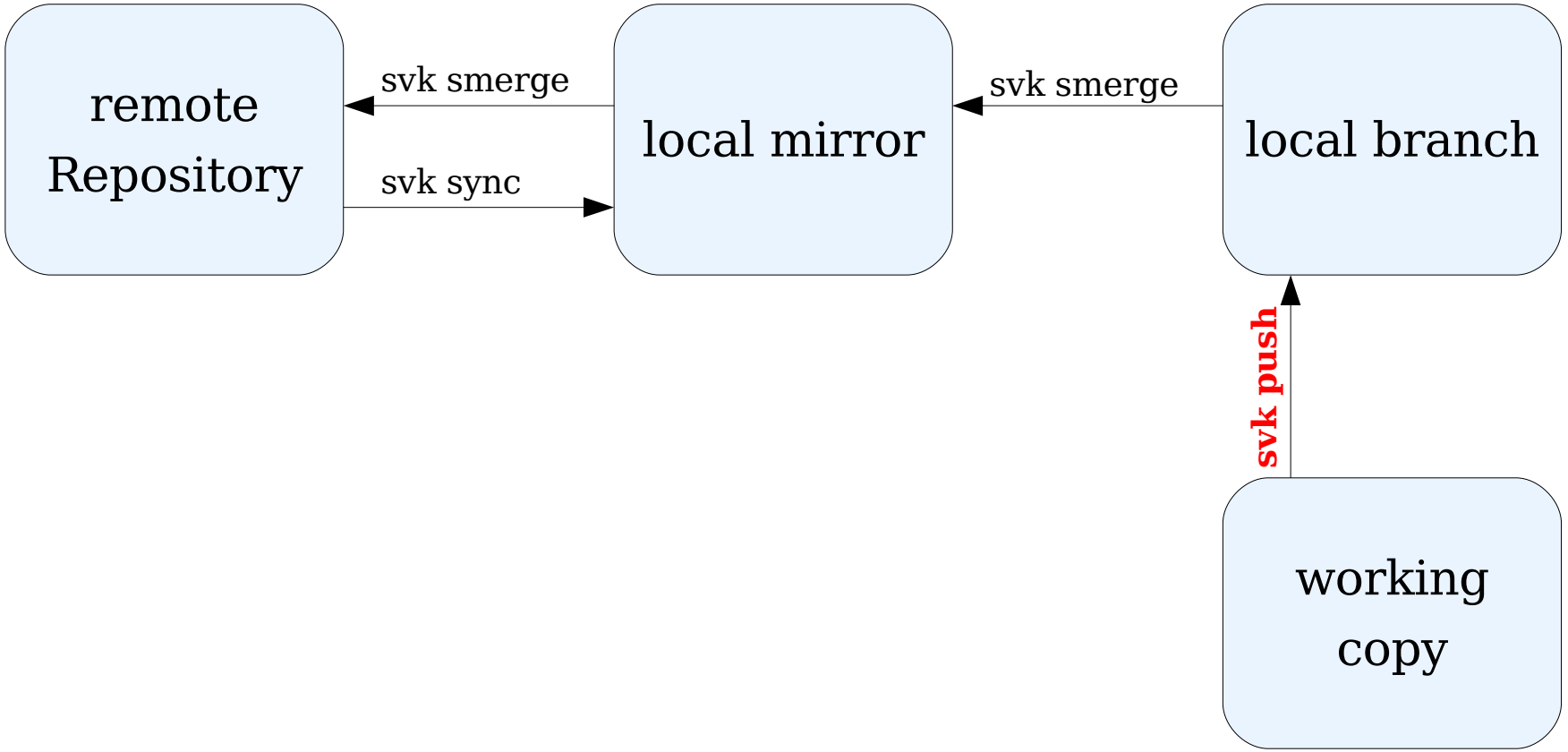
6. Mirroring with SVK Synchronization

- If you want to bring changes from your working copy to the remote repository you can simply use:

```
1  svk push
2     .
```

This will synchronize only into a number of transactions.

6. Mirroring with SVK Work Flow for svk push



6. Mirroring with SVK Working in a Team

- Every team member will be assigned to a single branch to work on. Those branches can be simply mirrored to their local machines.
- Merging back via SVK will not produce any conflicts. This will happen later.

7. Differences to SVN Checkout

- The SVK checkout command does usually not create „.svk“ subdirectories, but if you want to know where you have checked out working copies you can do this by the following command:

```
svk checkout --list
```

7. Differences to SVN The mkdir command

- You can use multiple URL's in Subversion as the parameters to the mkdir-Command:

```
svn mkdir
   http://server/project/branches
   http://server/project/tags
   http://server/project/trunk
   -m”- Create repos structure.”
```

7. Differences to SVN
The mkdir command

- But in SVK this does not work:

```
svk mkdir DEPOT DEPOT...
```

7. Differences to SVN The delete command

- Delete directories from the working copy:

  ```
  svk delete dir
  ```

  The directory will be deleted immediately from the working copy and not as in Subversion at the time of commit.

8. Pro's and Con's Advantages of SVK

- You can check-in/check-out locally and record the whole history of changes.
- You can continue development independently if you are on-line or not.

8. Pro's and Con's Advantages of SVK

- SVK working copies do not have supplemental „.svk“ directory (default).
- You can create a „floating“ working copy which has a „.svk“ directory.

8. Pro's and Con's Advantages of SVK

- You need noticable less space than Subversion needs for working copies.

#WC's	SVN	SVK
1	173	158 (83+75)
2	346	241
3	519	324

(sizes in MiB)

8. Pro's and Con's Disadvantages of SVK

- No tool support in form of PlugIns
  - e.g. for Eclipse, IntelliJ, Zend Studio etc.
  - Exception
    - approach for Ant.
- Non existing GUI clients
- Documentation is (not) complete.

9. Use Cases SourceForge Account

- You have a SourceForge project with SVN support.
  - Simply mirror your repository and synchronize it via SVK, because svnsync does not work on SourceForge (You need hook scripts on SF site).

9. Use Cases
On the road

- You are often abroad.
  - Simply mirror your repository or your working branch and synchronize it with SVK if you have been back from your travel.

10.Tips and tricks
Use Tools anyway

- You can checkout directly from the depot via file:// protocol to use Tools like TortoiseSVN, Eclipse, trac etc. and use SVK only to synchronize.

On-line sources I

- [1] Homepage SVK mit Wiki
  - http://svk.bestpractical.com
- [2] Book about SVK (work in progress)
  - http://svkbook.elixus.org
- [3] Homepage of Subversion
  - http://subversion.tigris.org
- [4] Book about Subversion
  - http://www.svnbook.org

On-line Sources II

- [5] Subversion Forum
  - http://www.svnforum.org
- [6] German Subversion forum
  - http://forum.subversionbuch.de
- [7] Forum for Software Configuration Management
  - http://www.xing.com/net/skm

On-line Sources III

- Different Clients / libraries for Subversion
  - http://subversion.tigris.org/links.html
- Comparsion of different SCM Tools
  - http://better-scm.berlios.de

Lectures

- Lecture on the Free OpenSource Conference (FrOSCon) 2006
  - Distributed Version Control with SVK
- Lecture on the Chemnitzer Linux Days 2006 „Introduction into Subversion“ (german)
  - Introduction into Subversion

Questions?

subconf2007@soebes.com

- Thank you for your attention.