

To be Distributed or not to be Distributed

Pro's and Con's on

Distributed vs. Centralized Version
Control Systems

Web Site:

www.soebes.com

Blog:

blog.soebes.com

Email:

info@soebes.com

Dipl.Ing.(FH) Karl Heinz Marbaise

Agenda

1. Glossar
2. Centralized Systems
3. Distributed Systems
4. Workflows
5. Examples of different Systems
6. Pro's / Con's

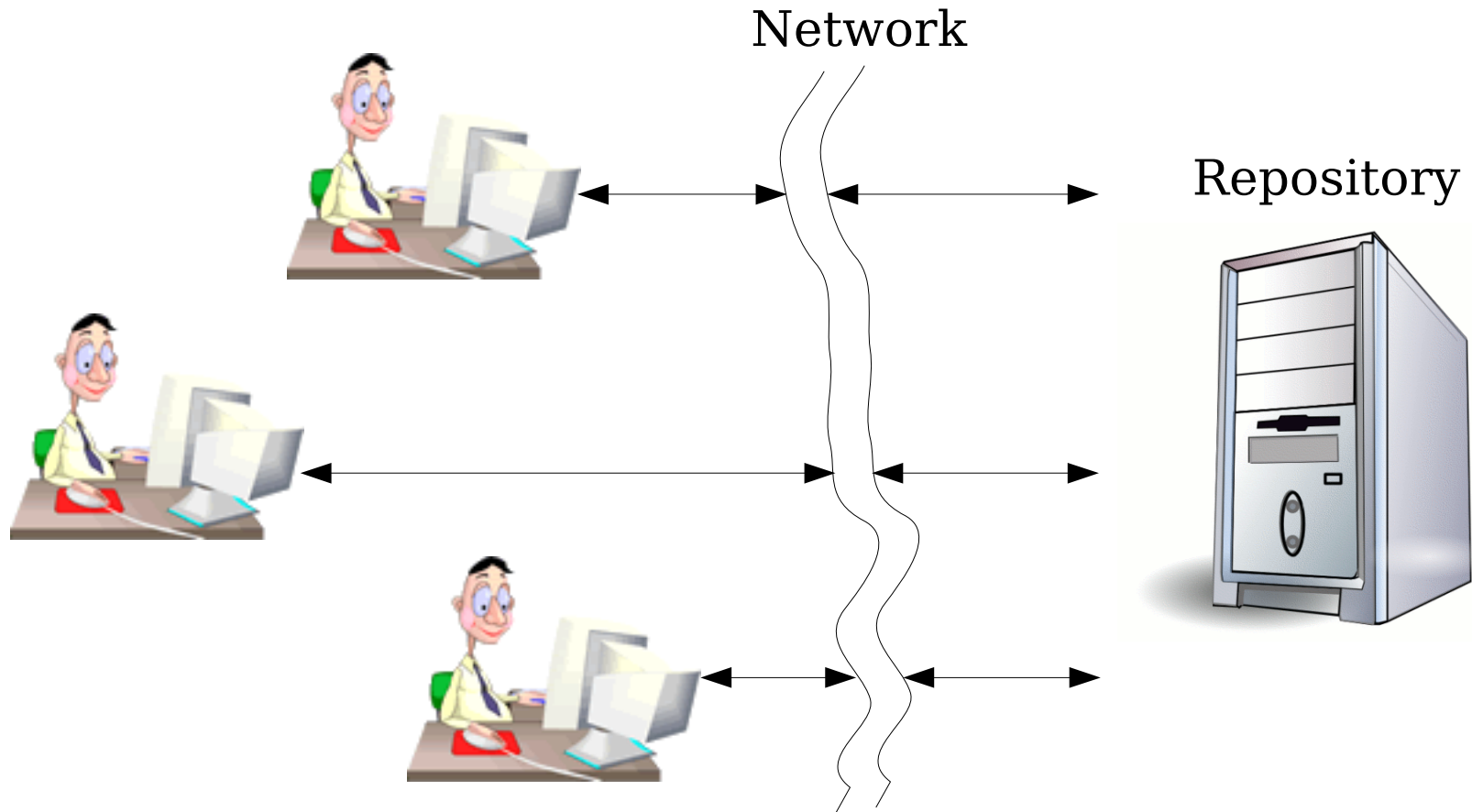
1. Glossar

- Decentralized Version Control System (DVCS)
 - Sometimes called **distributed Systems**.
- Centralized Version Control System (CVCS)
 - Sometimes called **centralized Systems**.

1. Glossar

- Continuous Integration (CI)
- Build Management (BM)
- Integration Manager (IM)

2. Centralized VCS Concept



2. Centralized VCS Concept

- Permission system is located on the central server.
- Backup is (?) made from the central repository
- Continuous Integration Systems can use the central repository to get everything they need.

2. Centralized VCS Concept

- Single Point of failure
 - Can be reduced if you use replicated repository etc. (usual industrial setup).

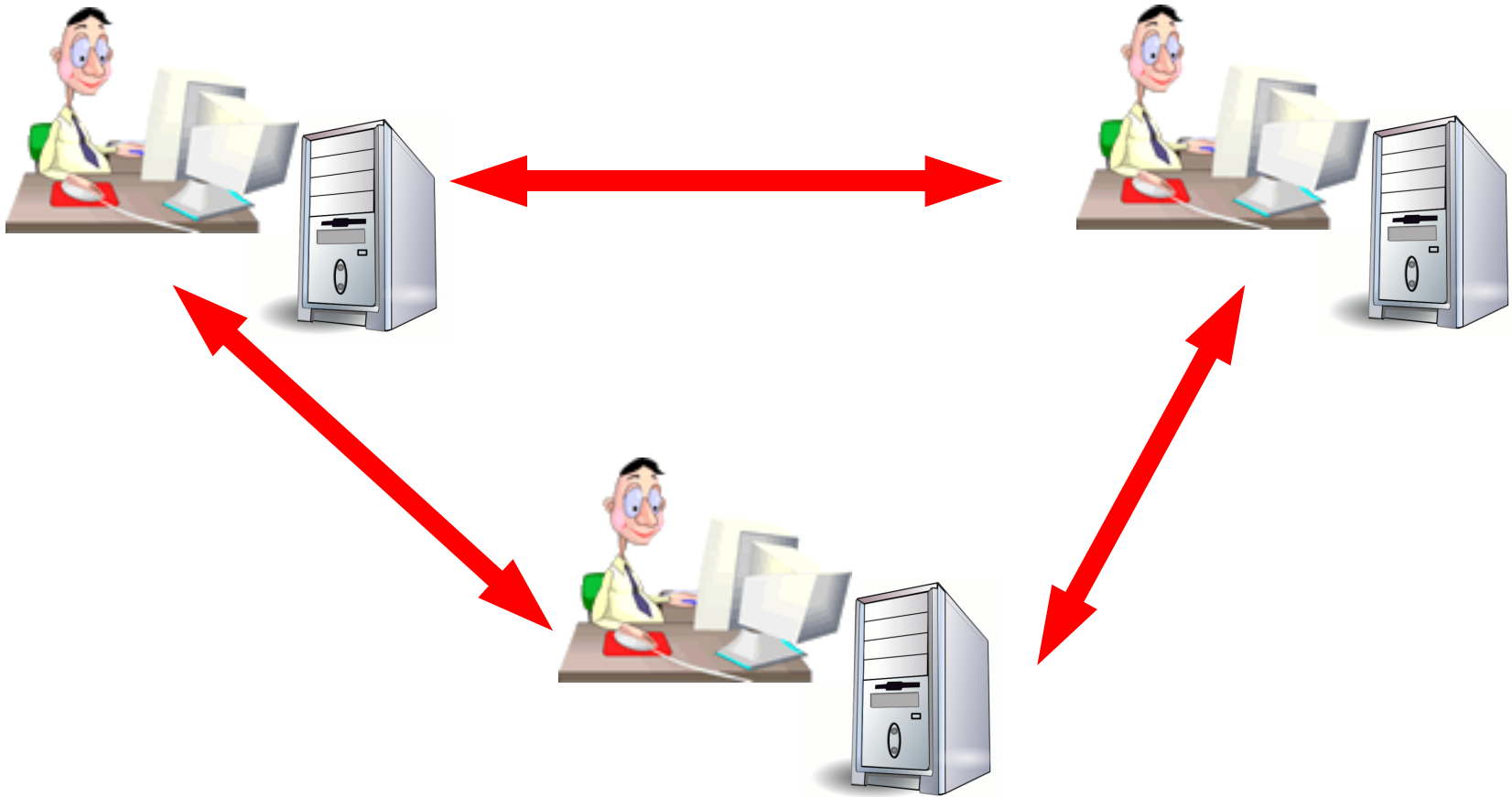
2. Centralized VCS Concept

- The network connection is needed for a particular set of operations.
 - checkout
 - checkin
 - viewing the history
 - making tags, branches etc.

2. Centralized VCS Concept

- You make changes by using the “Working Copy” which is a copy of a partial part of the repository (branch/trunk).

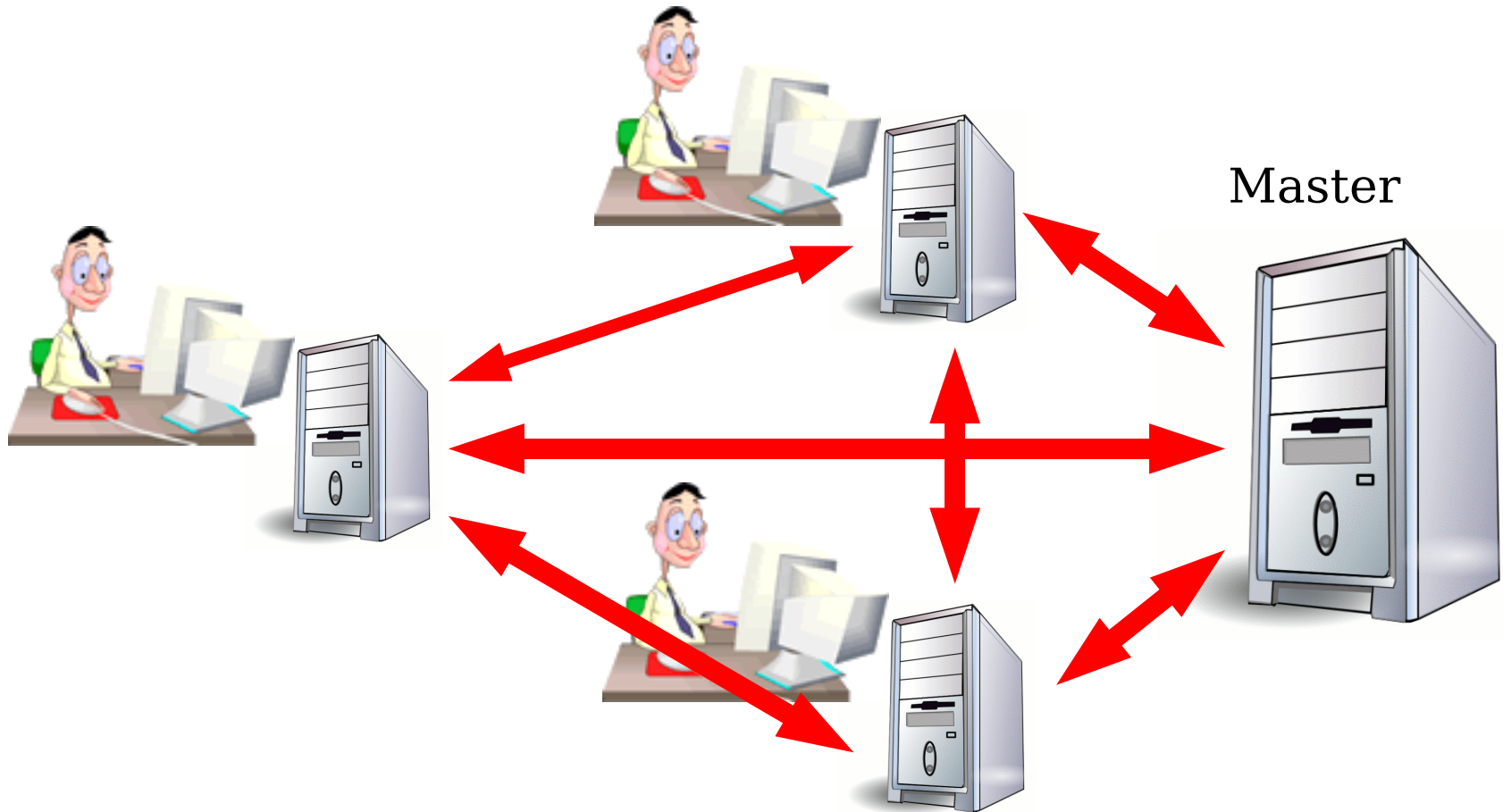
3. Distributed Systems Concept



3. Distributed Systems Concept

- Based on practical reasons we need a „centralized point“ in decentralized systems.

3. Distributed Systems Concept



3. Distributed Systems Concept

- No single point of failure, except for the master repository.
- Everyone has a full copy of the source incl. the history.
 - No network connection needed to view the history etc.

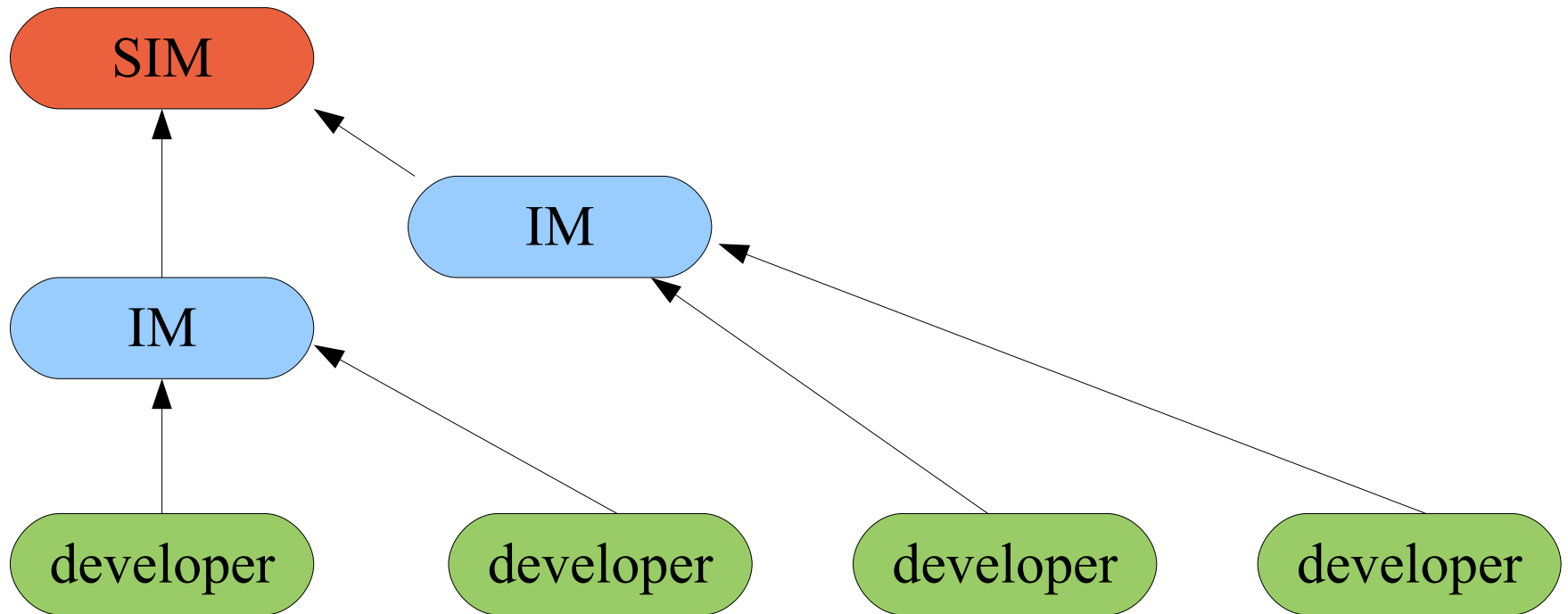
3. Distributed Systems Concept

- You can do off-line commits, cause it's yours.
- Everyone can share code (branches, patches) with each other (push/pull).
- Backup is „not“ needed, cause everyone has a backup.
- Usually no permission system implemented.

4. Workflows

- Using Branching strategy to define a workflow e.g.:
 - Different Integration lines
 - Different Test lines
 - Different Release lines
 - Different Quality Gates
 - etc.

4. Workflows



5. Examples of Distributed Systems

- Open Source
 - git, mercurial, monotone, bazaar, darcs, SVK etc.
- Commercial
 - BitKeeper

5. Examples

Centralized System

- Open Source
 - Subversion
 - CVS, ok not really.
 - ?
- Commercial
 - Perforce
 - ClearCase
 - etc.

6. Pro's / Con's Centralized

- Pro's
 - Single Point of Source
 - Permission can be defined at the central server
 - Centralized backup / Administration
 - CI has everything at one point.
- Con's
 - Single point of failure
 - No off-line commits
 - No backup of the working copies

6. Pro's / Con's Distributed

- Pro's
 - You can do off-line commits.
 - You have to define a master as a centralized part.
- Con's
 - No backup of the working copies, might be even worse than missing a single commit in a centralized environment.

6. Pro's / Con's Distributed

- Con's
 - Continuous integration can be a problem if things have not been pushed to the master repository.
 - What happens if you loose your notebook?

6. Pro's / Con's Distributed

- Con's
 - Companies don't like to give employees the full source code only parts of it etc.
- The culture of the company must cope with DCVS.

On-line Sources

- Subversion
 - <http://subversion.tigris.org>
- Git
 - <http://git-scm.com>
- Bazaar
 - <http://bazaar-vcs.org/en>
- Mercurial
 - <http://mercurial.selenic.com/wiki>

On-line Sources

- Darcs
 - <http://www.darcs.net>
- SVK
 - <http://svk.bestpractical.com/view/HomePage>
- Monotone
 - <http://www.monotone.ca>
- BitKeeper
 - <http://www.bitkeeper.com/>

On-line Sources

- ClearCase
 - <http://www-01.ibm.com/software/awdtools/clear>

On-line Sources

- Tree Conflicts
 - [BSc Thesis - Tree Conflict Handling](#)
- Comparisons between SVN etc.
 - <http://git.or.cz/gitwiki/GitSvnComparision>
 - <http://whygitisbetterthanx.com/>
 - <http://stackoverflow.com/questions/871/why-is-g>

Questions?

subconf2009@soebes.com

Thank you for your attention.