Git
Users and Repository Configuration

Version Control Systems

Juraj Oravec

STU

Slovak University of Technology in Bratislava

Git – recall: main objectives

- recordes changes over time (*changesets*)
- recalls a specific version later (*checkout*)
- enables collaboration (*distributed*)
- allows nonlinear development (*branches*)

Git – recall: DOs and DONTs

DONTs:

– do not use archives as a substitute for version control

– never use a centralized VCS (CVS, SVN)

– do not store automatically generated files (PDFs, binaries, etc.)

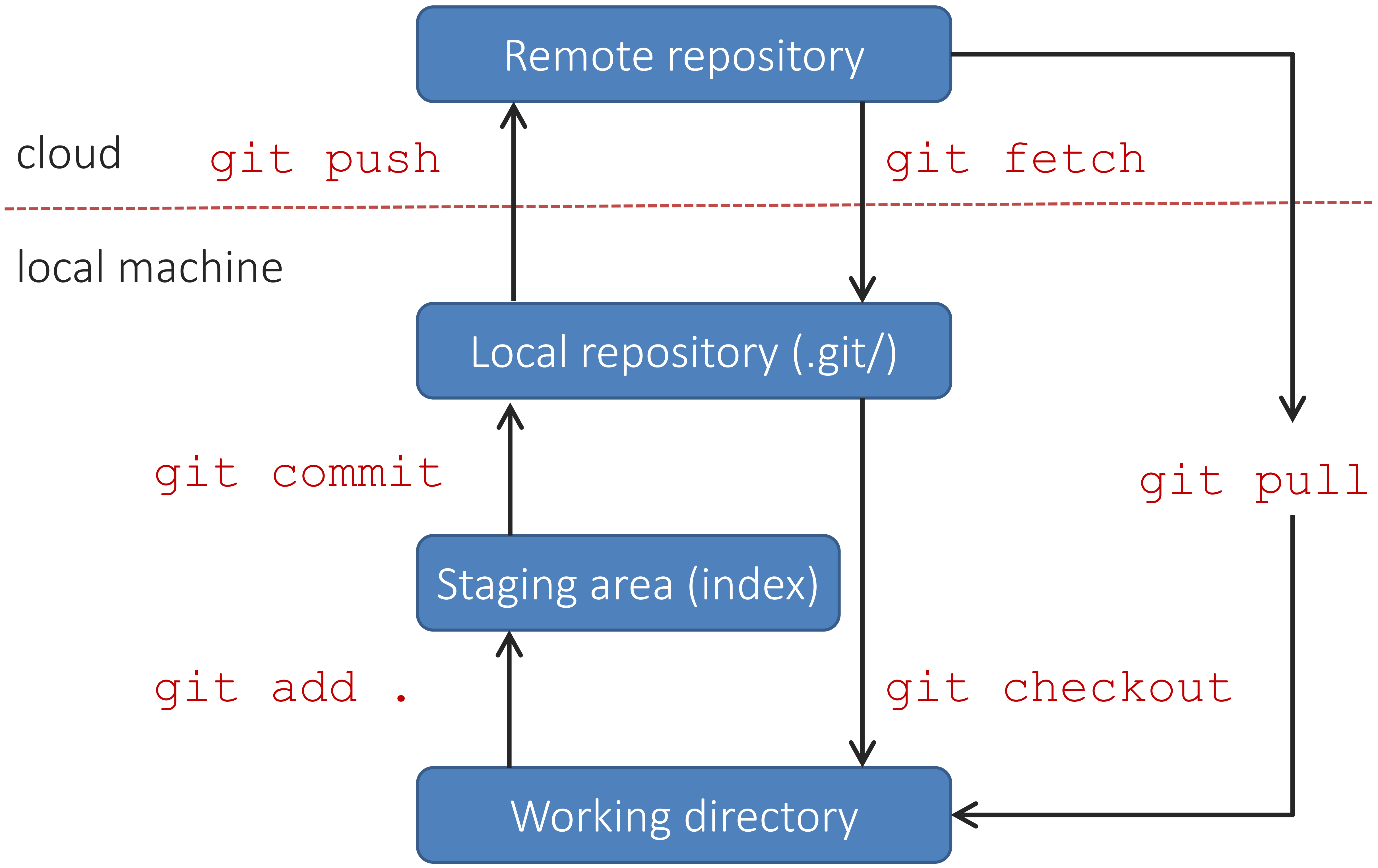
– do not store sensitive data (passwords, logins, SSH keys, etc.)

DOs:

– use a distributed VCS (Git, Mercurial)

– use proper branching model

Git – recall: basic workflow



Git – recall: basic objects

`hash`

object identifier – each object has its (hexadecimal) *hash*

`blob`

contains *pure* file content (without file names)

`tree`

directory listings – path entries paired with other objects

`commit`

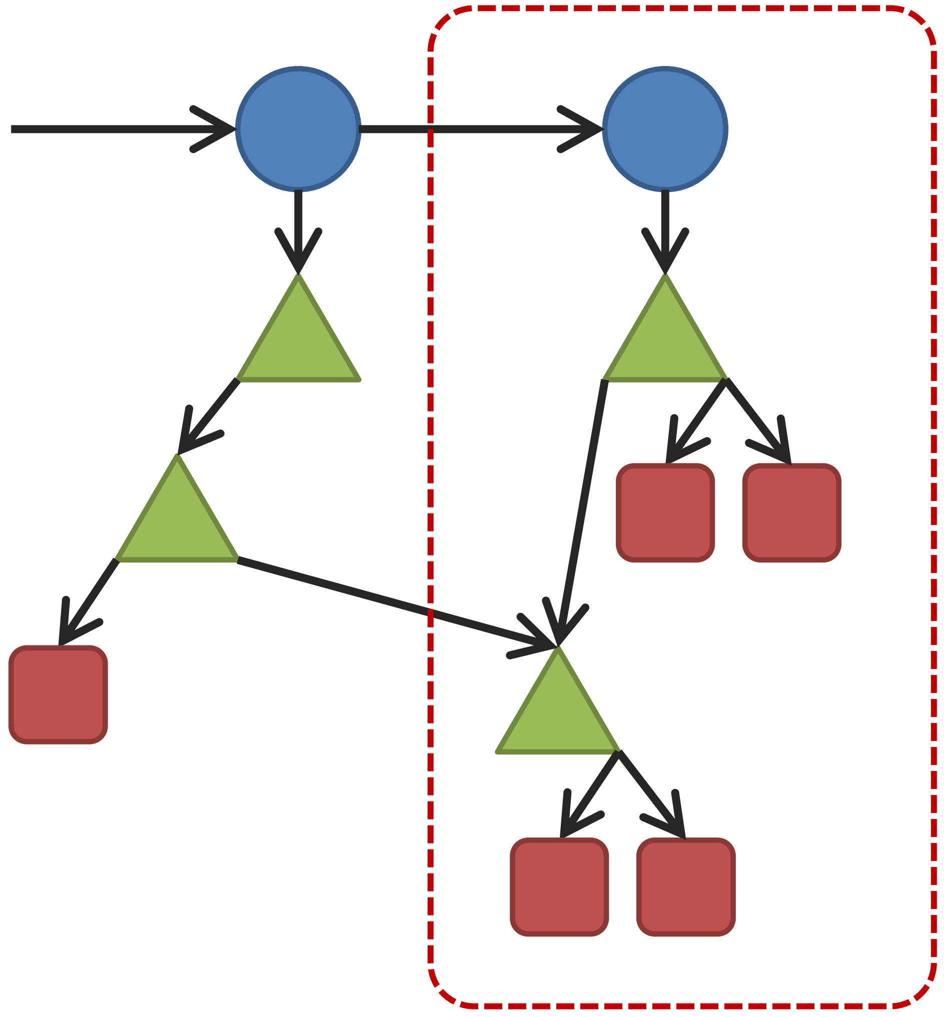
snapshots (*changeset*) of repository in time; each commit has its root tree

Git – recall: basic objects

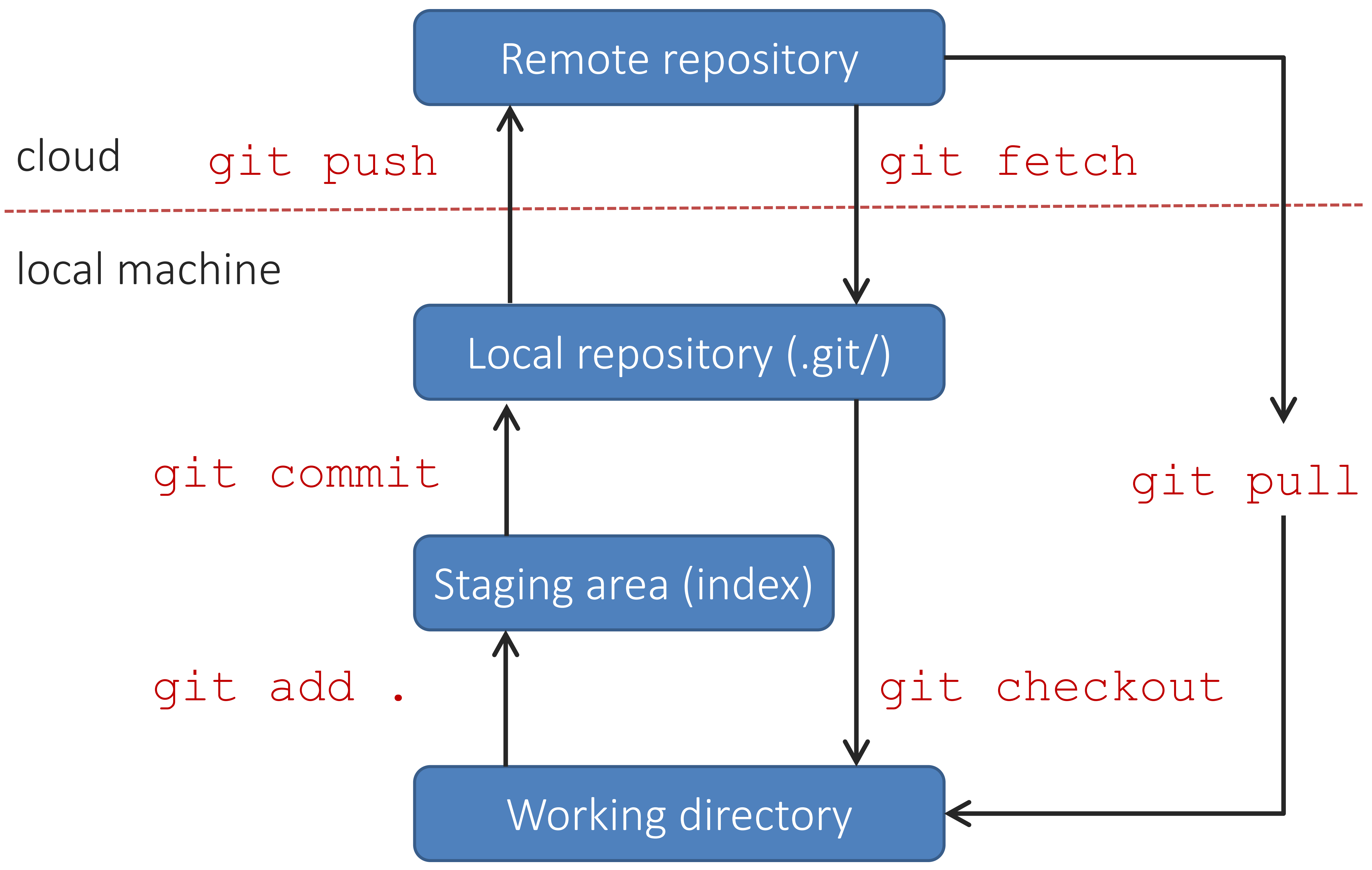
`commit` (circle)

`tree` (triangle)

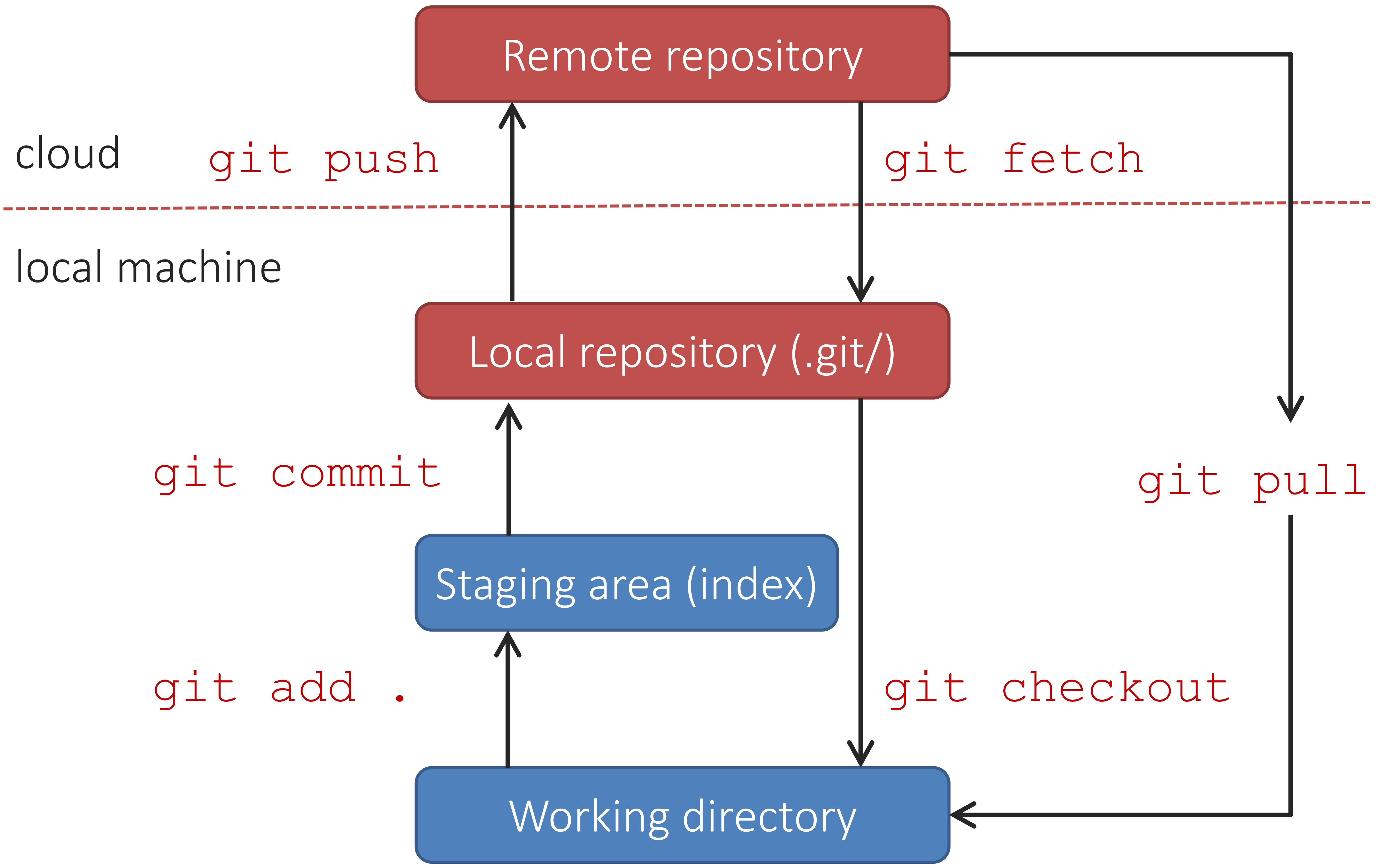
`blob` (box)



Git – basic workflow



Git – basic workflow



Git – version

```
git version
```

returns information about current installation of Git

Git – support

We may recall help anytime we need it

```
git help
```

opens web-browser to display help information related to Git

```
git help help
```

opens help of command `help`

```
git help git
```

opens help of basic usage of Git

Git – create repository

```
git init
```

creates new empty repository in current folder

Git – configure repository

```
git config -e
git config --edit
```

edits configuration file to configure git

Git – configure user

```
git config --global user.name "Juraj Oravec"
git config --global user.email juraj.oravec@stuba.sk
```

configures user and e-mail address

Git – configure merging tool

```
git config --global merge.tool vimdiff
```

configures default merging tool vimdiff

```
git config --global merge.tool kdiff3
```

configures default merging tool kdiff3

```
git config --global mergetool.kdiff3.cmd
'"C:\\Program Files\\TortoiseHg\\lib\\kdiff3" $BASE
$LOCAL $REMOTE -o $MERGED’
```

configures default merging tool kdiff3

Git – configure text editor

```
git config --global core.editor vim
```

configures VIM as *default* (core) text editor

```
git config --global core.editor "'C:\Program
Files\Sublime Text 3\sublime_text.exe'"
```

configures Sublime Text as *default* (core) text editor

Git – manage remote repositories

`git remote`

manages set of repositories whose branches you track

`git remote --verbose`

shows remote URLs in verbose mode

`git remote add` ***`name path`***

adds a remote named *name* (origin) for the repository at *path* (URL)

`git remote rm` ***`name`***

removes a remote named *name*

Git – quick start (to be understood later)

```
touch filename
```

creates empty file *filename*

```
git add .
```

adds all files into staging area

```
git commit -m "message"
```

commits changeset labeled by *message*

```
git push
```

pushes changeset to remote repository – but this command fails now!

Git – set upstream

It is possible to directly set upstream for pushing

```
git push –u origin main
git push --set-upstream origin main
```

sets upstream for communication with remote repository

Git – set upstream

Keywords:

`origin` – is alias for remote repository (URL)

`origin/main` – is remote branch on remote repository

`main` – is branch in local repository

Note:

original default term `master` has been renamed to `main` in 2021

Git – initialize pull

It is necessary to initialize upstream for pull

```
git pull origin main
```

sets upstream for communication with remote repository (always)

```
git pull --set-upstream origin main
```

sets upstream for communication with remote repository (just once)

```
git branch --set-upstream-to=origin/main main
```

if there already exist a branch `main` , then this command sets upstream

for communication with remote repository (just once)

Git – push/pull changesets and check status

`git push`

pushes changeset to declared remote repository

`git pull`

pulls changesets from remote repository to local directory

`git status`

shows the working tree status

Git – brief summary of initialization

`git init`

initializes repository directly into current folder

`git remote add` *`origin URL`*

assigns a remote repository

`git push –u` *`origin main`*

assigns upstream

Git – cloning repository

`git clone` *`path`*

clones existing repository into current folder from preset *path* (URL)

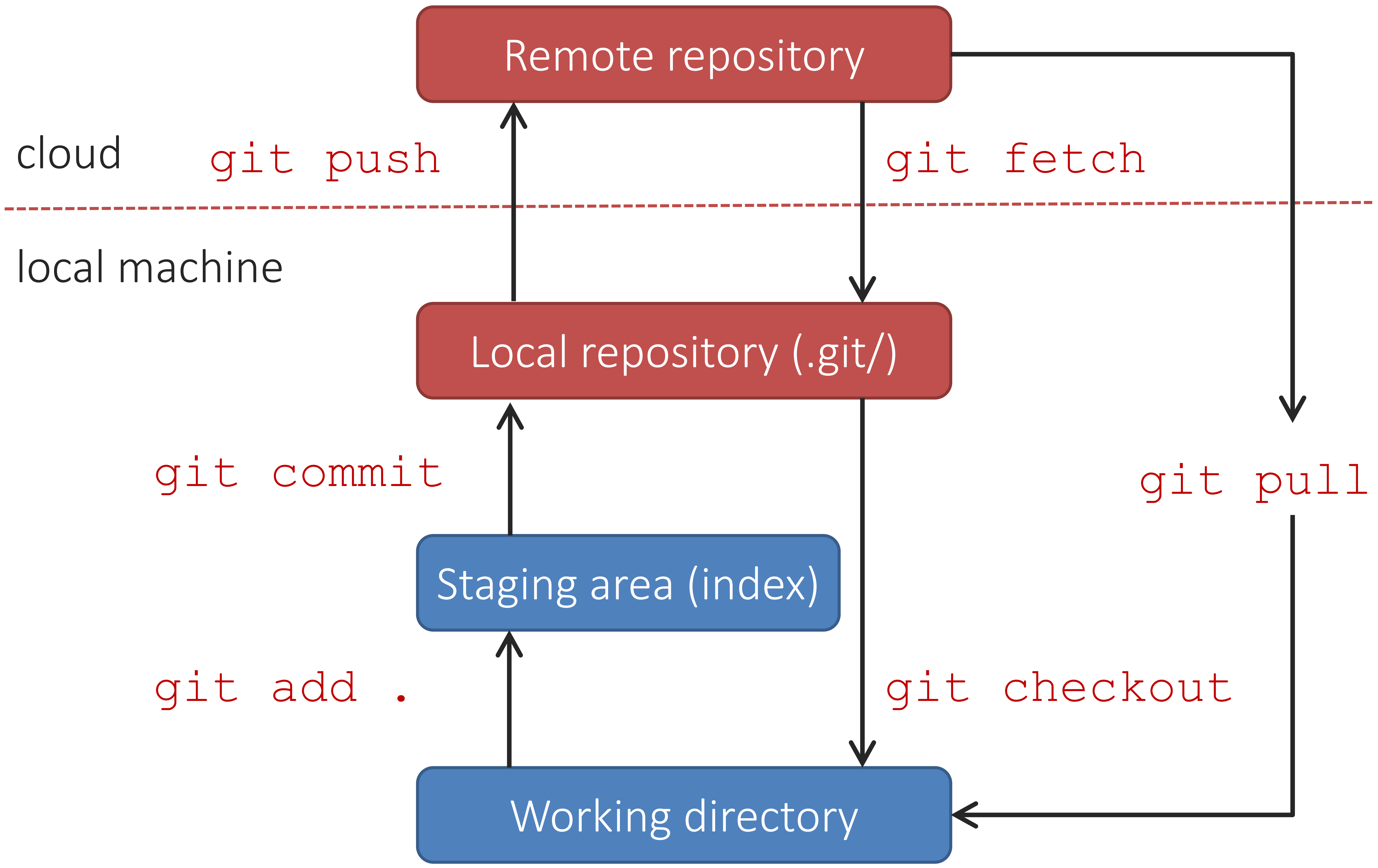
Note: cloning automatically assigns alias *origin* of remote repository

and sets upstream/downstream

Git – summary

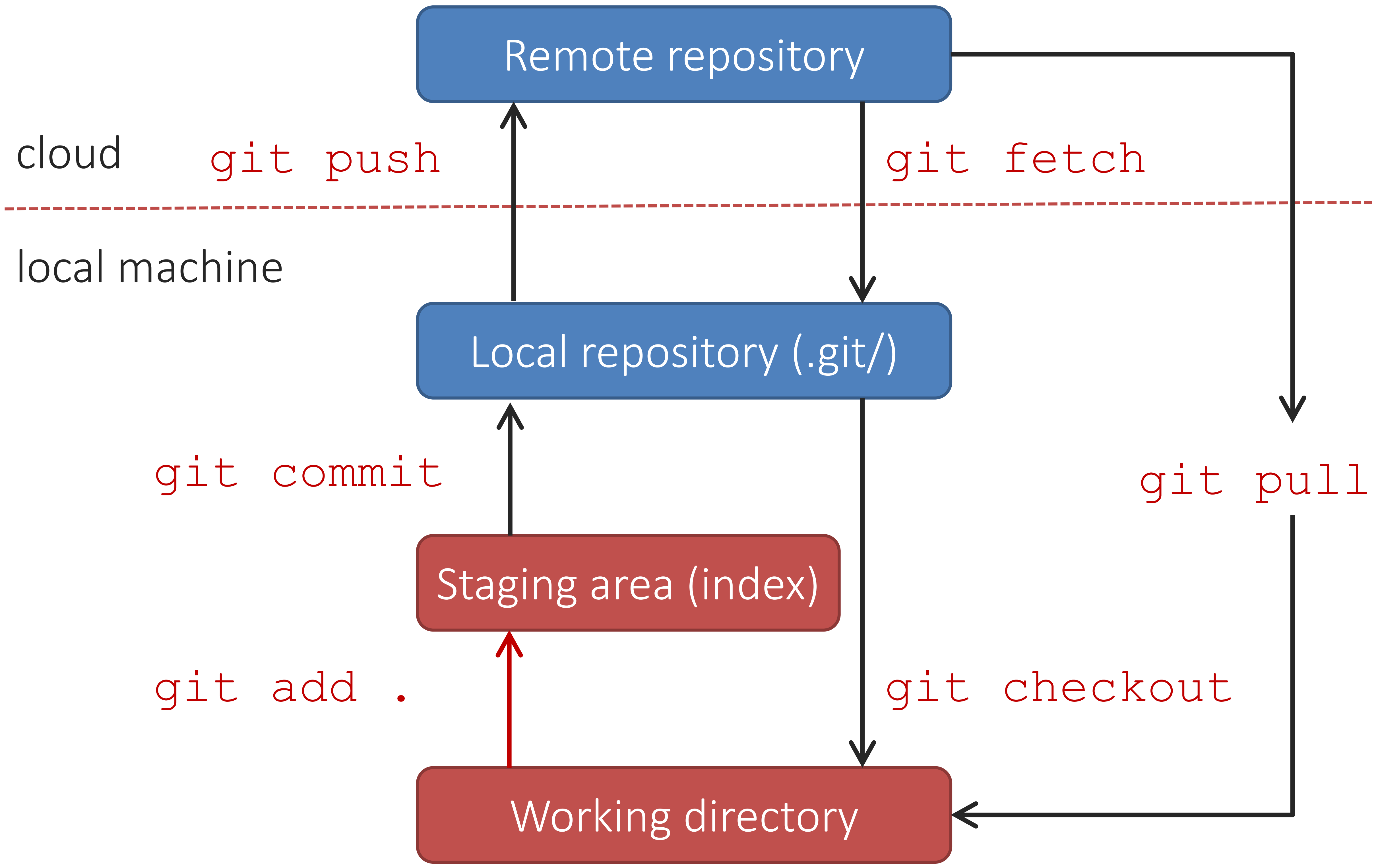
```
git version

git help

git init

git config

git remote add name path

git remote rm name

git clone path
```

Git – summary



Remote repository
cloud
git push
git fetch
local machine
Local repository (.git/)
git commit
git pull
Staging area (index)
git add .
git checkout
Working directory

Git – outlook



cloud
local machine
Remote repository
git push
git fetch
Local repository (.git/)
git commit
git pull
Staging area (index)
git add .
git checkout
Working directory