Workshop @Uni Bonn, D. Barbi (dbarbi@uni-bonn.de)

Version Control Systems – General Concepts

Version Control Systems

Why Version Control?

Organize versions

Different versions of the same files on different „hierarchies“:

- ***Major revisions (e.g. model releases)***
- ***Projects (Ice coupling, hosing, solid earth coupling)***
- ***Minor changes (bugfixes, parameters)***
- ***Work in progress (Hey, it used to work yesterday…)***

Don't loose progress

Using a server with backup means:

- ***A backup for all (!) your work***
- ***Accessible from all machines (no matter if ollie crashed…)***
- ***Synchronize work on different platforms / for different co-workers***

Organize team work

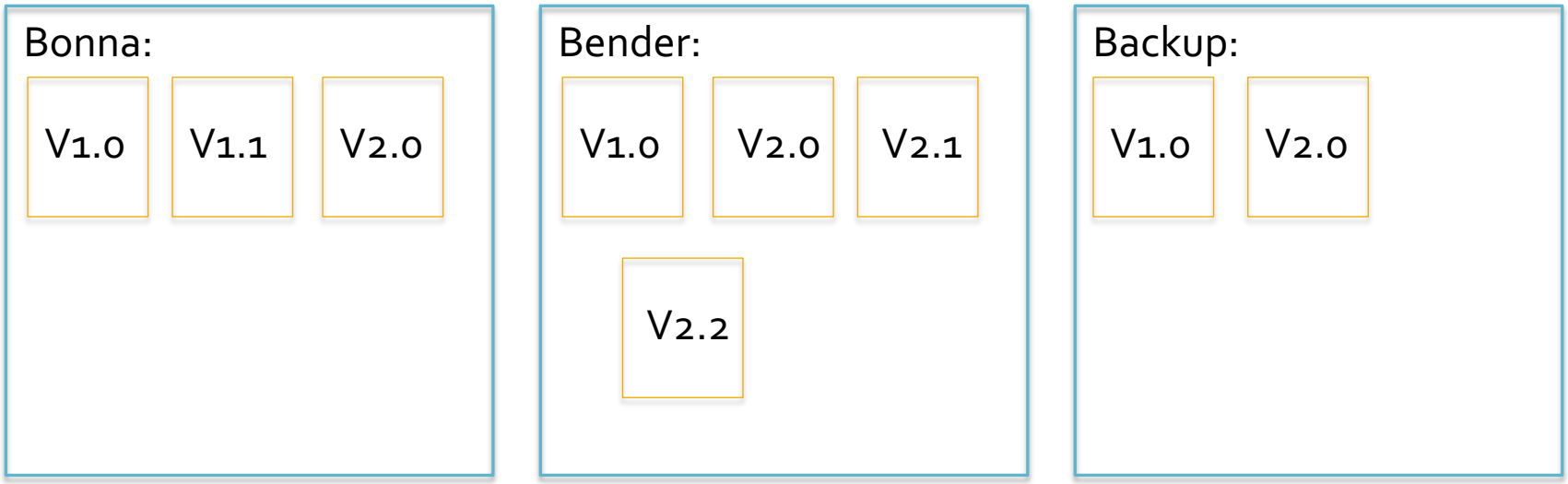
Using a database to keep work sorted means:

- ***Merge your work with that of others, share problems and solutions***
- ***Distribute your work, make it easy to access***
- ***Apply your work to new major revisions***

Version Control Systems

Basic Concept

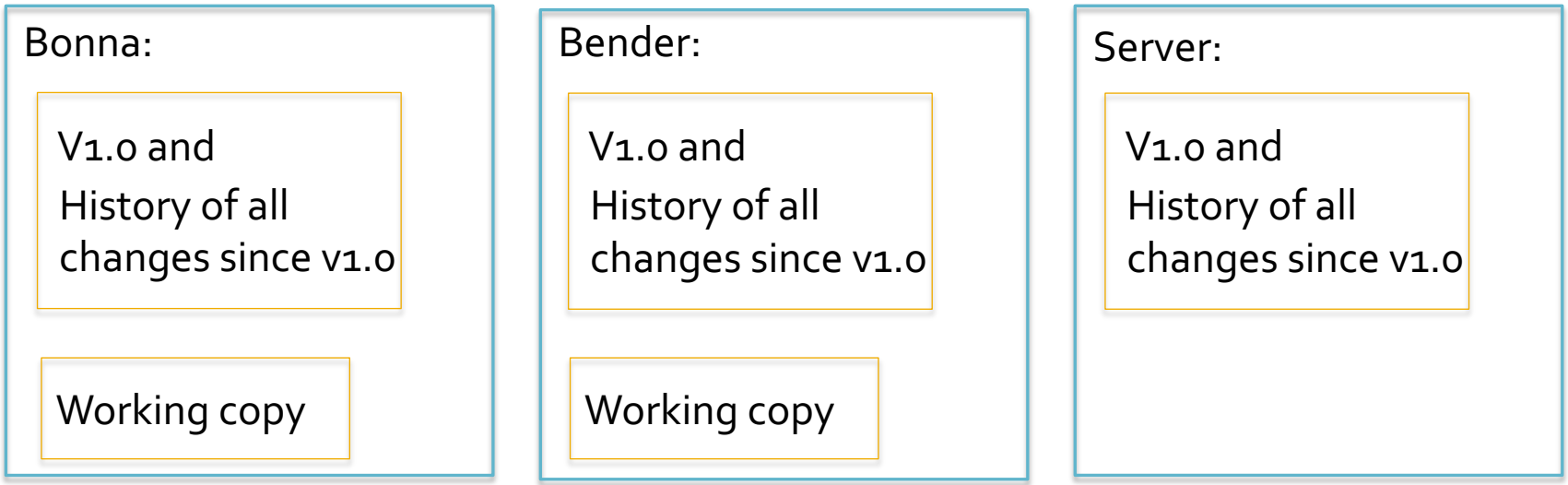
Without Version Control



Problems:

- Bender crashes, but you want to work on v2.2
- Are v2.0 in all three cases identical? How can you be sure it stays that way?
- A bug is found in all v2.x
- Your coworker has some good changes for v2.1 – but so do you
- Let's say each folder is 50 GB, but only small differences between them…

With Version Control (-- git)



Features:

- Synchronization between Machines and Server
- Remember changes instead of versions, save in a database
- Choose ANY version you ever had of your project – not only v1.1, v2.0,…
- Choose that on ANY machine, even if you log in for the first time
- Save a lot of place (only one instance)
- Save a lot of work…

A lot of choice…

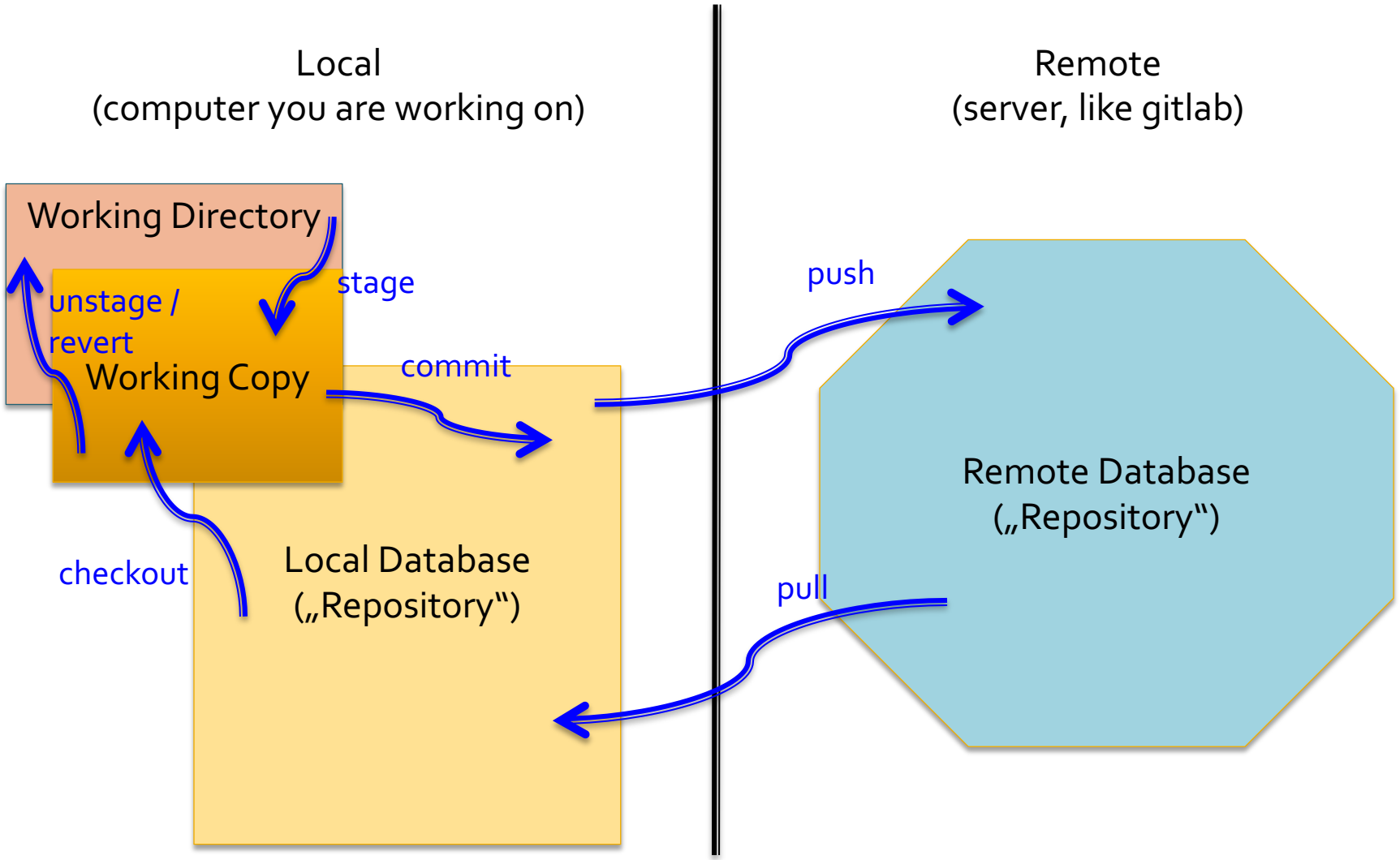
Version Control Systems:

- **git**
- svn
- cvs
- monotone
- …

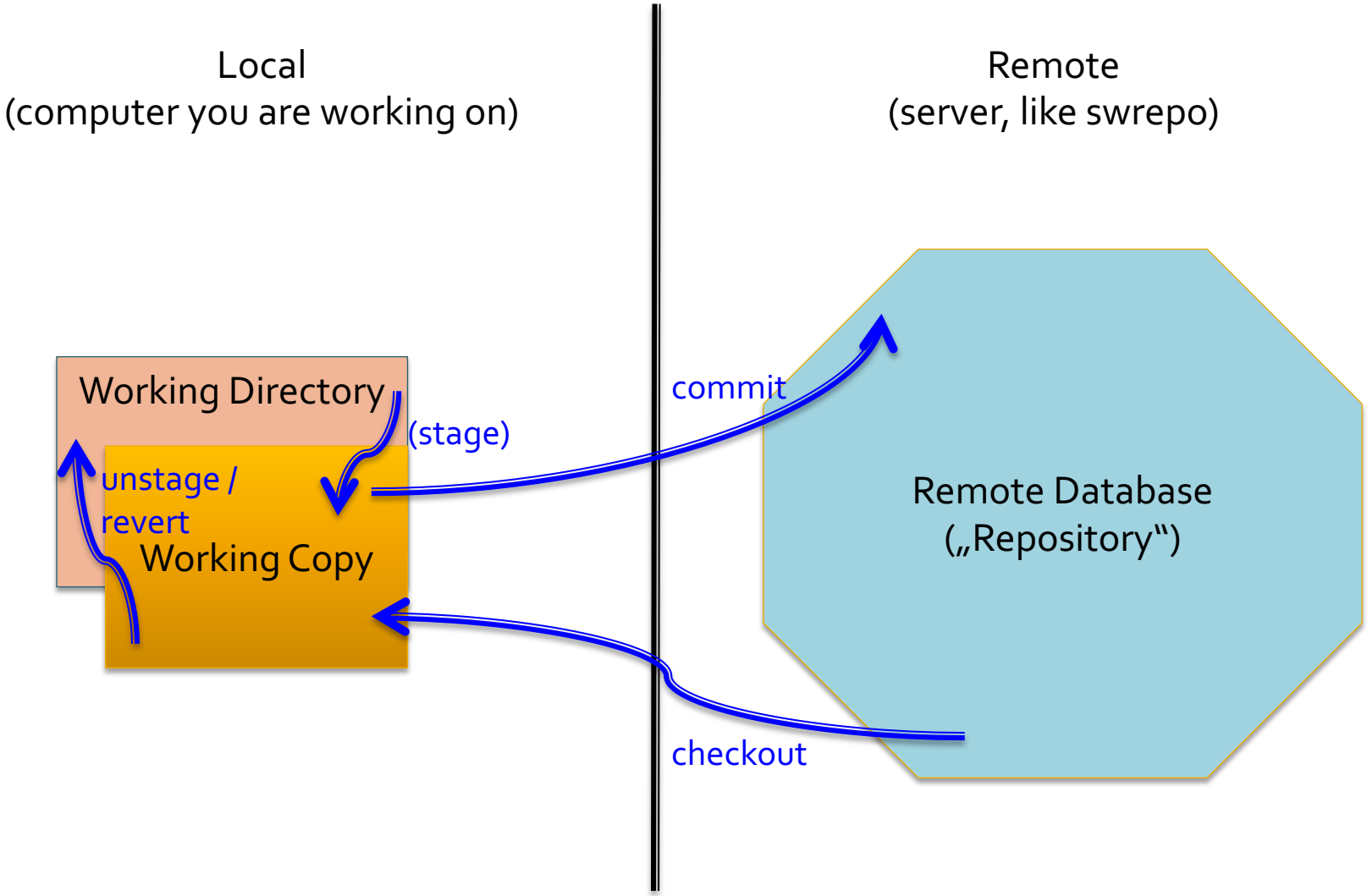
Repository Servers:

- bitbucket @ Uni Bonn
- Gitlab
- github
- …

Basic concept - git



Basic concept - svn



(Local) File Status Lifecycle -- git