Data Management Tools
Hands-on Seminar with…

Carsten Schirnick & Claas Faber
GEOMAR data management team (2 out of 5)

GEOMAR

GEOMAR

"Problem":

data management or better: manage **your** data

- **useful** documentation of your research & data and your **daily** work
- **make** complex and creative **work**(flows) ready to become **FAIR**:
  - **F**indable
  - **A**ccessible
  - **I**nteroperable
  - **R**e-usable

GEOMAR

Our goal for today

- …learn **basic usage** of Git as **versioning** tool
  - hands-on **simple workflow** of versioning
  - understand the **basics** - be ready for more…
- …learn basics of Jupyter **Notebook** as a tool
  - **documenting** work steps to achieve data plot
  - use **versioning** for reproducible results
  - disseminate and re-use your research

GEOMAR

Why care about versioning

- **useful** documentation of your research & data and your daily work
  - **Create** things
  - **Save** things
  - **Edit** things
  - Save the thing **again**

GEOMAR

Why care about versioning - why Git

- **Git** is a fast and modern implementation
- Git is for **distributed** work
- Git provides a **history** of content changes
- Git facilitates **collaborative changes** to files
- Git is easy to use for

  any type of **knowledge worker**

GEOMAR

Git versioning - Get going

- installation - √ (command line and UI tools, all systems ;-)
- search for Git Bash
- configure your **username** and **email**
  - no credentials (yet) - give you credits for your work…:
- `$ git config –global user.name “Carsten Schirnick”`
- `$ git config –global user.email “cschirnick@geomar.de”`
- `$ git init myProject`
- `$ cd myProject`

Git versioning - Get going

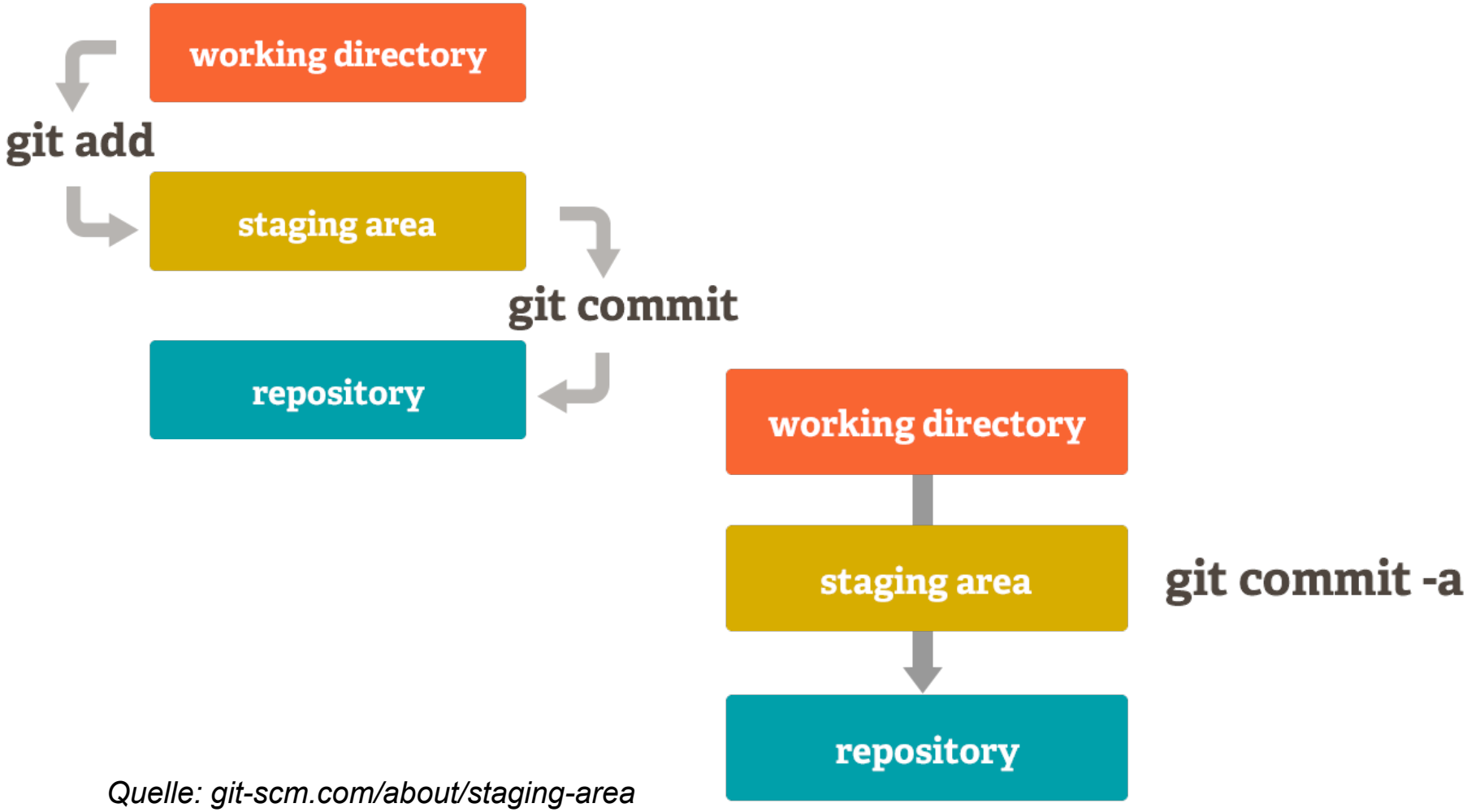
- `$ cd myProject`
- `$ ls -al`
  - .git = hidden folder with all Git needs to know
  - .gitignore = a file telling Git to ignore files, folders, … *(later more)*

create and write content to a file myFile.txt (echo “hello world” > myFile.txt)

- `$ git add myFile.txt`
- `$ git commit -m “Add my first file to versioning”`

- `$ git log`

Data Management Tools



Quelle: git-scm.com/about/staging-area

Git versioning - Get going

- `$ git add <files>, <folders>…`
  - …you have to explicitly add files to a commit !
  - …you may choose which files to add to a particular commit !
- `$ git commit -m '…'` **or**
- `$ git commit -am '…'`
- `$ git add <file>` ooops, wrong one
- `$ git reset HEAD <file>`

Git collaboration

- `$ git clone https://git.geomar.de/isos/hands-on.git`
  - …you can clone remote repositories (using SSH or HTTPS) !
  - …you can add / change / edit remote resources !
- `$ git remote -v`
  - …you receive updates from remotes (pull)!
  - …you send your commits to remotes (push) !
- `$ git pull origin <branch>`
- `$ git push origin <branch>`

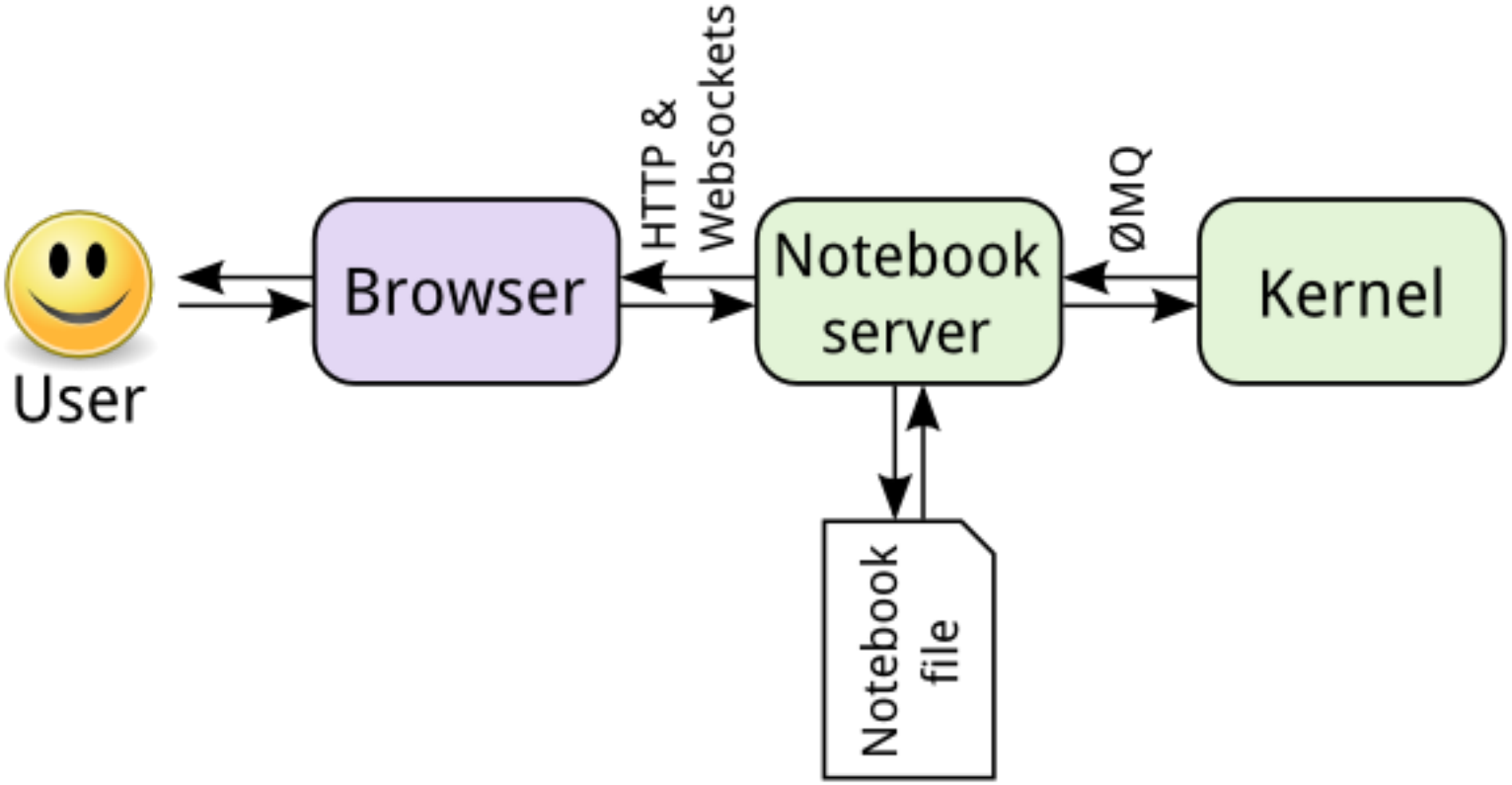
GEOMAR

The Jupyter Notebook

"The Jupyter Notebook is an open-source **web application** that allows you to create and **share documents** that contain **live code**, **equations**, **visualizations** and **explanatory text**. Uses include: **data cleaning** and **transformation**, **numerical simulation**, **statistical modeling**, **machine learning** and **much more**."

`jupyter.org Webseite`

Data Management Tools



Quelle: jupyter.org

GEOMAR

Jupyter Notebook

- interactive cells:
  - cells with code (default: Python)
  - cells with Markdown text

- add cell in browser window
- adjust type of cell [Code | Markdown]
- insert code or Markdown text
- execute cell with SHIFT-ENTER or menu button
- view the resulting output in browser window

GEOMAR

Jupyter Notebook Server

- user starts Jupyter Notebook Server as webserver
  - local (or central on Jupyter-Hub)
  - interaction in browser window per HTTP(S) & websockets
- Notebook Server loads and saves Notebooks
  - Notebooks editable even without a Kernel
  - but then not executable!
- Kernel “doesn’t know" anything about Notebook
  - receives "cells" of Notebooks with code and executes content

Jupyter Notebook - workflow in the wild

AUV Bathymetry

- problem: "brain drain"
- reconstruction of knowledge “…how to create bathymetric maps from data…"
- starting point: illustrated and well formatted MS WORD-document as manual
  - use pandoc to convert WORD to Markdown-text
  - extract images from .docx-file
  - convert TIFF to PNG images
  - copy & paste into Jupyter Notebook
  - separation of code and markdown
- foundation for new and reusable workflows with Jupyter Notebooks…:

Jupyter Notebooks - for complex processing…

AUV140_MK

April 7, 2017

0.1 AUV Abyss processing workflow

Version 3.0; to navadjust a single survey within itself

0.1.1 Dive 0140

Part of the **ODEMAR Irinovskoe** surveys
Purpose of this workflow:
Processing and navadjust of 5 dives at the 13°20′ OCC to

- compare results to prvious processing attempts.
- minimize the internal distortion ("shrinking") by either
- careful internal navadjust
- and/or by adjusting AUV navigation to overall shipboard survey (especially to the clearly visible termination of the OCC in shipboard and AUV data)

0.1.2 Define basic parameters

```
In [6]: # enter dive number
        dive='0140'
        print (dive)
        # Enter spatial resolution in meter
        sr=3
        print (sr)

        # enter rootpath of raw-data
        rootpath="/media/mklischies/MK_PhD2017/_ODEMAR_AUV_Irinovskoe/Abyss0140_26-
        !cd $rootpath

        # definition of more variables:
        datalistraw=('datalistraw_%(dive)s.mb-1' % locals())
        print (datalistraw)

        datalistp=('datalistp_%(dive)s.mb-1' % locals())
        print (datalistp)
        grd1=('Abyss_%(dive)s_raw_%(sr)sm' % locals())
        print (grd1)
```

…thanks