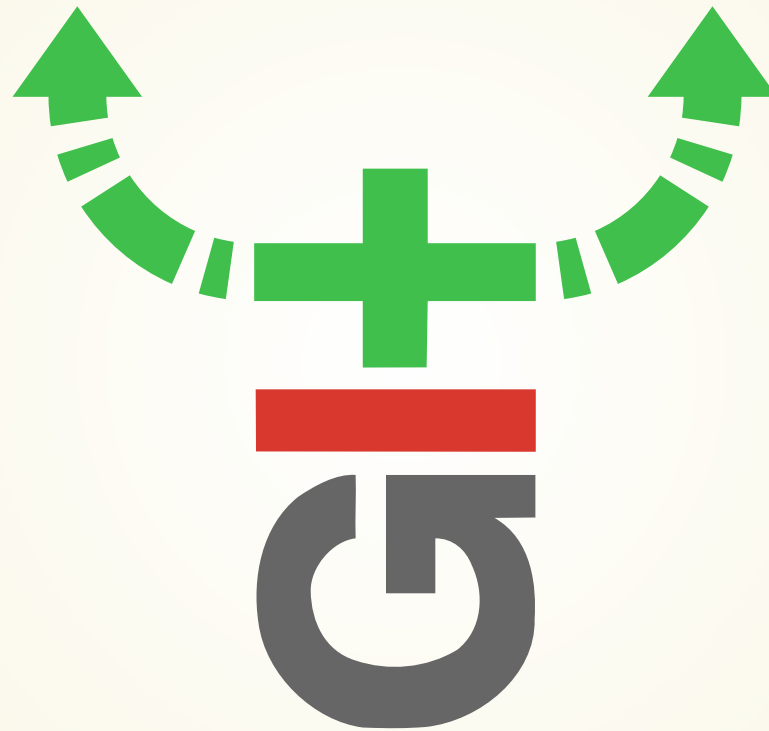




# DISTRIBUTED FILEMANAGEMENT USING GIT- ANNEX



© 2014 Christoph Piechula  
University of Applied Sciences Augsburg

# INTRODUCTION

This presentation will give you an overview about the concepts and features provided by the file synchronisation tool git-annex.

**Well, you will learn about:**

- File synchronisation systems and their issues
- Git annex **repositories** and **remotes**
- Git annex **distributed** synchronization concept
- Git annex **security concepts**
- Git annex assistant **webapp** tool

# CLOUD STORAGE SERVICES

Cloud storage is a comfortable solution to share data across several systems and networks

- Today data should be always accessible, on different system
- Using a cloud service like
  - Box.com
  - Dropbox
  - Megaupload
  - ...

makes sense...but

# THERE ARE ISSUES

- Is your data „safe" inside the cloud?
  - Dropbox issue made password optional for several hours
  - Data loss caused by a bug in Dropbox Client
  - Megaupload taken down by FBI
  - ...
- Local storage
  - Hard drives often silently fail
  - Data gets corrupted by malicious software
  - ...

# A POSSIBLE SOLUTION

To keep your data safe and secured not just a backup but a backup strategy is needed.

- Backup important data to different location
- Encrypt sensible data if it is stored in the „cloud“
- Track your data, verify your data because hardware is error-prone

Too much effort, right?

# GIT - A DEVELOPERS VIEW!

Git is a version control system primary used by developers to track their code changes.

- Linus Torvalds calls it: A „stupid" content tracker.



- Problem:
  - Not suitable for big binary data

# GIT CONCEPT - REPOSITORIES AND REMOTES

- A **repository** is the **place where your files are stored**. Usually this are so-called git repositories. Git repositories have a `.git` folder with a specific structure.
- A **remote** is a **external place where a repository may be stored**. This might be for example a external usbdrive oder a cloud service



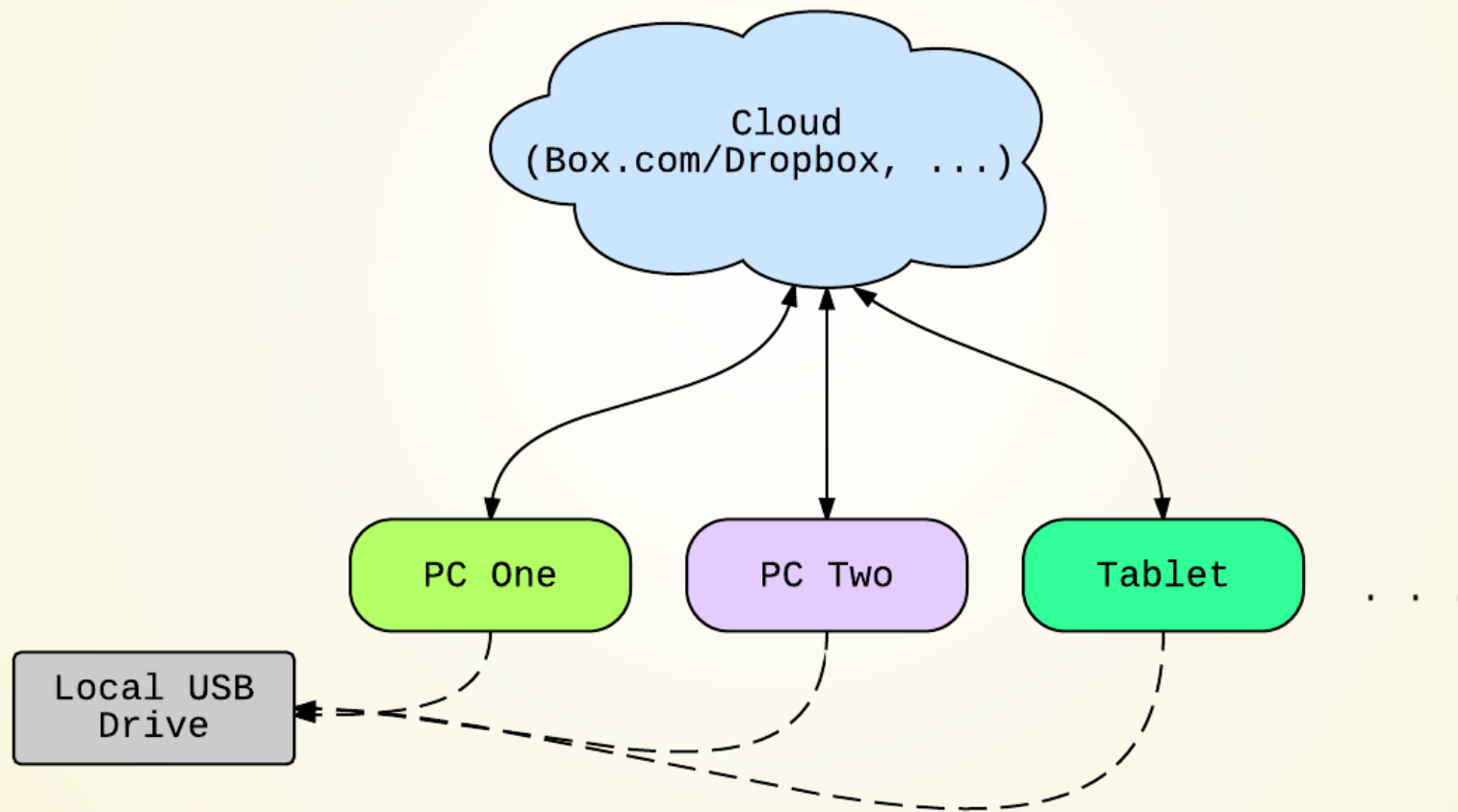
# GIT-ANNEX OVERVIEW

Git-annex is a tool that extends the git concept by only tracking the content's metadata. In this way it allows git to track large binary files without checking them into git.

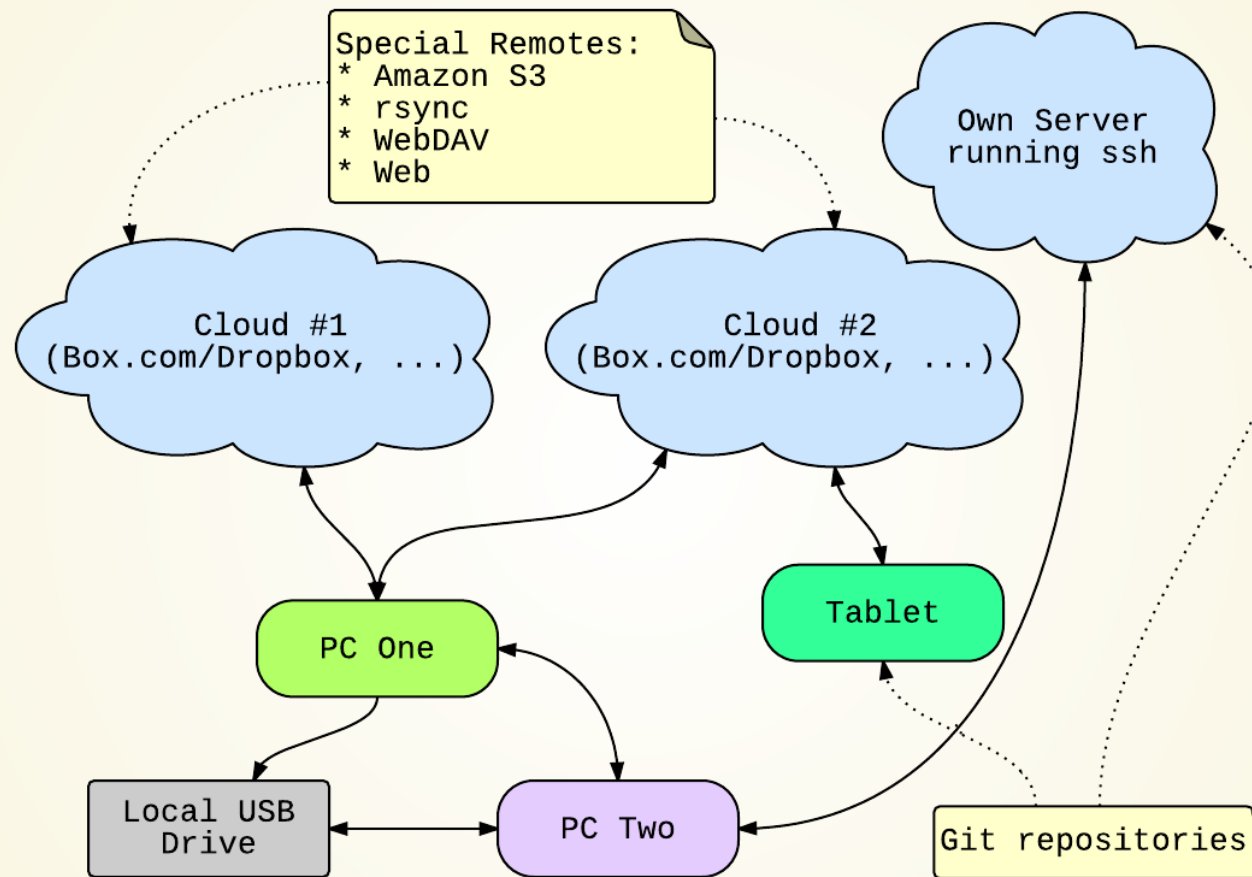


- Crowdfunding project, reached over \$20000
- Developed in Haskell by Joey Hess
- Free Software
- Primary a commandline tool
- Now, there is a fancy GUI

# FILE SYNCHRONISATION ACROSS DIFFERENT DEVICES AND NETWORKS

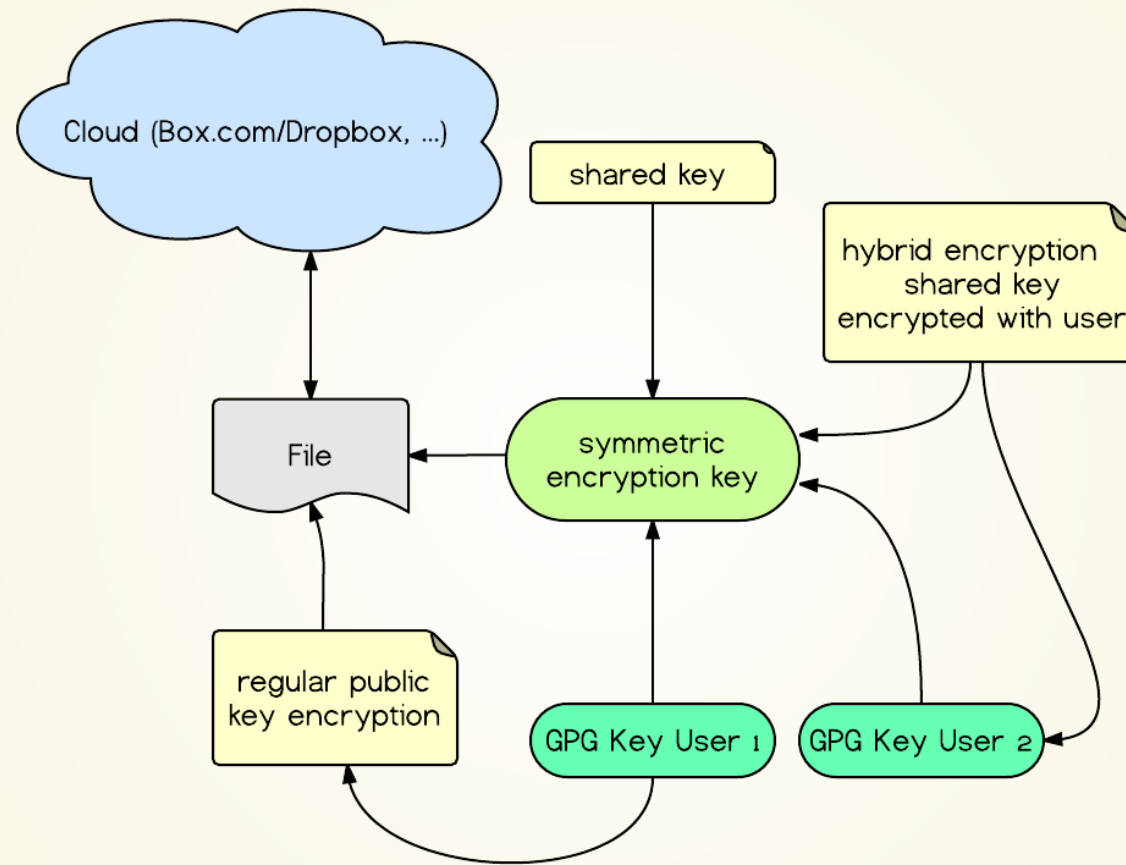


# GIT-ANNEX WAY TO SYNCHRONIZE YOUR DATA



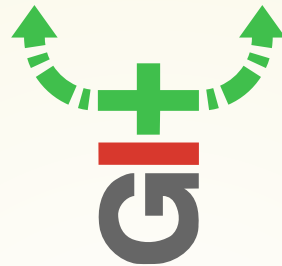
- Git-annex extends git concept by special remotes

# SHARED AND HYBRID ENCRYPTION CONCEPT



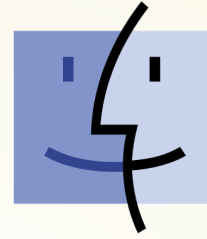
- Supports different encryption use cases

# OTHER GIT ANNEX FEATURES

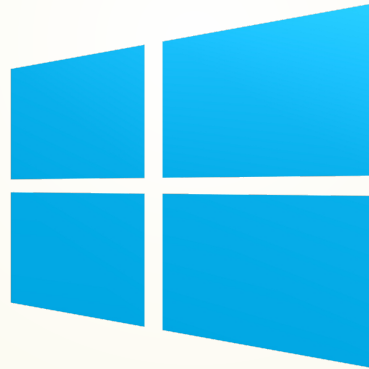


- File integrity check (cryptographic hashsum)
- Location tracking
- Minimum number of copies
- Different repository groups
- Repository trust levels
- Chunked file upload support

# OPERATING SYSTEM SUPPORT



Mac<sup>TM</sup> OS android



# THE GIT-ANNEX COMMANDLINE USAGE

```
$ git init myrepo
Initialized empty Git repository in /home/qitta/myrepo/.git/
$ cd myrepo
$ git annex init 'myrepo'
init myrepo ok
(Recording state in git...)
```

- `git-annex` commands to work with:
  - `add`
  - `drop`
  - `whereis`
  - `info`
- Just type `git-annex` to get the whole list or check the manpage. Usual `git` commands are used to add/remove remotes, check status or commit changes.

# GIT-ANNEX WEBAPP GUI-INTERFACE

- „Like DropBox, but with your own cloud"

The screenshot displays the Git-Annex Webapp GUI interface in a web browser. The browser's address bar shows the URL `127.0.0.1:38244/?auth=6ee6d6bcc5abe4f3b9d957c7663211455b389e4135588674531220e76f83a6764dc92b6bd27a60c8554`. The interface has a dark header bar with the text "git-annex" and navigation links for "Dashboard", "Configuration", and "About". On the right of the header, it shows "Files" and "Current Repository: ~/lib/family".

On the left side, there is a sidebar with several status boxes:

- A blue box with a circular arrow icon and the text "Syncing with family".
- A green box with a checkmark icon and the text "Synced with kite".
- A green box with a checkmark icon and the text "Synced with family".
- A green box with a checkmark icon and the text "Downloaded", followed by a list of files: DSCF6100.JPG, DSCF6096.JPG, DSCF6094.JPG, DSCF6092.JPG, DSCF6091.JPG, DSCF6089.JPG, DSCF6088.JPG, DSCF6087.JPG, DSCF6086.JPG, and DSCF6085.JPG.
- A green box with a checkmark icon and the text "Committed changes to git".
- A green box with a checkmark icon and the text "Performed startup scan".

The main content area is divided into two sections:

### Repositories

Repository	Status	Settings
here (Joey's laptop)	syncing enabled	settings
kite	syncing enabled	settings
family (Adrianne and Maggie)	syncing enabled	settings
rsync.net	syncing enabled	settings

Below the table is a button "Add another repository" and a description: "Sync your files with another device, or share with a friend."

### Transfers

Transfer	Progress	Buttons
trip/DSCF6101.JPG ← rsync.net	78% of 5 MiB	Pause, Cancel
trip/DSCF6102.JPG ← rsync.net	queued (5 MiB)	Play, Cancel
trip/DSCF6104.JPG ← rsync.net	queued (4 MiB)	Play, Cancel
trip/DSCF6108.JPG ← rsync.net	queued (4 MiB)	Play, Cancel
trip/DSCF6110.JPG ← rsync.net	queued (4 MiB)	Play, Cancel

The bottom of the browser window shows a taskbar with system icons, the text "~/lib/family - Chromium", and the system clock: "51 °F 2:13 pm Fri Apr 05".



# EVERYTHING PERFECT? - LET'S SUM UP

## Pros:

- Integrates itself into existing cloud environment
- Encryption to secure your data
- File integrity and location tracking
- Chunked file upload support
- Free and Open Source Software
- Operating system support

## Cons:

- Git annex is still under heavy development
- Commandline Interface is a high entry barrier

# REFERENCES

Main ressource:

<https://git-annex.branchable.com/>

The basics, git annex walkthrough:

<https://git-annex.branchable.com/walkthrough/>

Some git-annex video tutorials

<https://git-annex.branchable.com/videos/>

**Thank you for your attention!**

**Questions?**