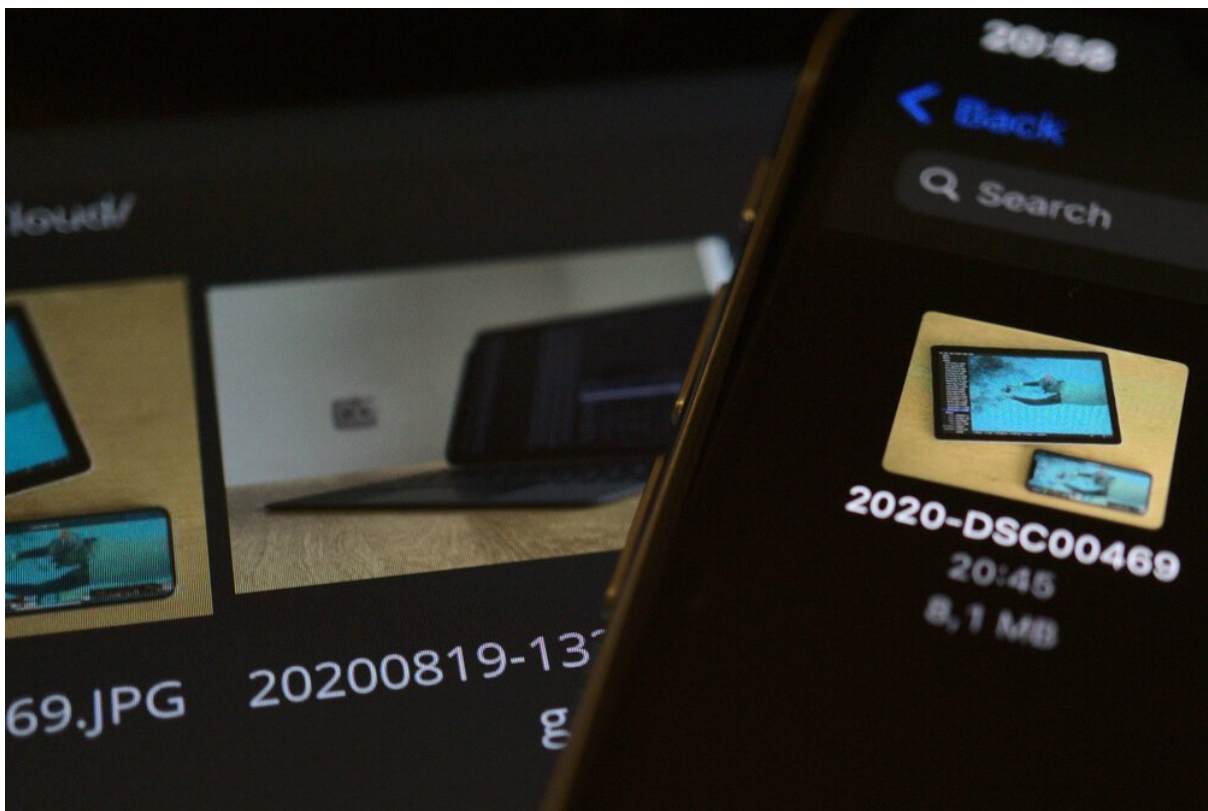


Syncing files seamlessly between smartphone and tablet

Using unison to automatically sync between GNU/Linux and the iPhone

Willem L. Middelkoop
Sep. 16, 2020



In order to optimise my workflow, I was looking for a way to seamlessly access the same files on both my computer and smartphone. This is useful to when you want to quickly send files from your computer using your smartphone through various messaging apps and vice versa. Read along to find out how I did it.

The goal: seamlessly shared files

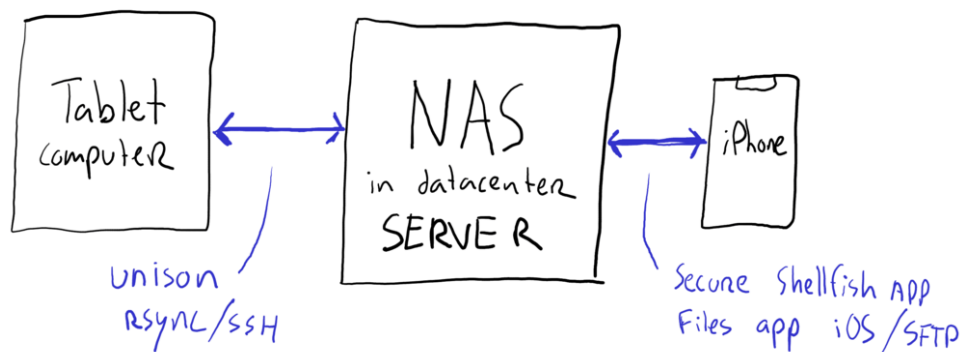
When I am working, I usually use both a tablet and a smartphone. I use a [tablet](#) as main computer. My iPhone is very useful as a second screen. I also use it to send files, screenshots and photos to people using the many different messaging apps that run on iOS.

For the better or the worse, there is a shift going on from email to apps such as Slack, WhatsApp, iMessage and Signal. My iPhone is the gateway to these channels of communication. As I do not want to use iCloud Drive or DropBox I needed another way to seamlessly have my files on both my tablet and smartphone.

Meet Unison: the multi platform file synchroniser

After a quick search I found [Unison](#) in the [Debian software repository](#). As my tablet runs on Debian GNU/Linux I looked for a way to synchronise files securely using open standards. Unison does this using rsync over SSH/SFTP. This means it will work with my [existing NAS server](#) that lives in a data center.

Although you could setup file sharing directly between your tablet and a smartphone, I use the central NAS server to allow communication between my devices, even if they're not on the same network. This happens when one (or both) devices are on cellular networks. To access files from my iPhone, I use the Secure ShellFish app that integrates nicely with the native iOS files app.



Synchronising files between my tablet and iPhone: the NAS server act as an intermediate station; overcoming differences in protocol and connection

Setting Unison up is easy. You should check out the English documentation on the [Unison website](#) or follow this [great Unison guide in German](#). You can run Unison from the command line, but I like it to run automatically using a preconfigured profile and a frequently running cron job.

```
willem@WillemSurface:~$ cat .unison/WillemBaseCloud.prf
# Unison preferences
root = /home/willem/Cloud
root = ssh://willem@base.willem.com/Cloud
sshargs = -C

batch=true
auto=true
confirmbigdel=false
willem@WillemSurface:~$
```

My Unison profile with parameters for the 'WillemCloud'

The Unison configuration profile is straightforward: you define two root folders and you can use some toggles to adjust the default behaviour. I have setup my profile to do things in batches and automatically decide which files to keep. I have a dedicated "Cloud" folder that I choose to share between my devices, it acts as a kind of "Desktop"-folder. Check out the Unison man page for all the possible parameters (try to type 'man unison' in your terminal).

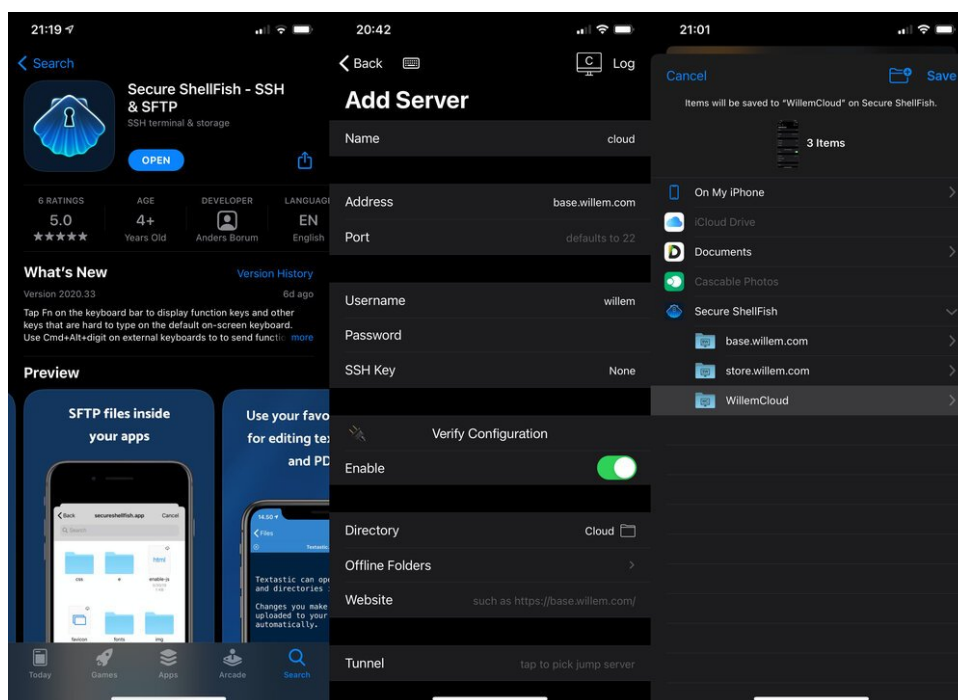
Authentication is done using your system's SSH configuration. You can (and should) setup [authentication using shared client keys](#). This will enable password-less SSH communication between your client and server. Once you have tested and configured your Unison profile, you can create a cron job to automatically synchronise your files. Because file transfers can take a longer time, the cron job should only run if the previous run was completed. You can do this using a PID file, acting as a check.

```
# Command to automatically sync files using unison, preventing double execution using a loc/pid file:
test -e /var/lock/unison-WillemBaseCloud && exit 0 || (touch /var/lock/unison-WillemBaseCloud;unison WillemBaseCloud;rm /var/lock/unison-WillemBaseCloud)
```

My cron command is wrapped in a check to prevent starting synchronisation when it is already running

Setup Secure ShellFish on iOS

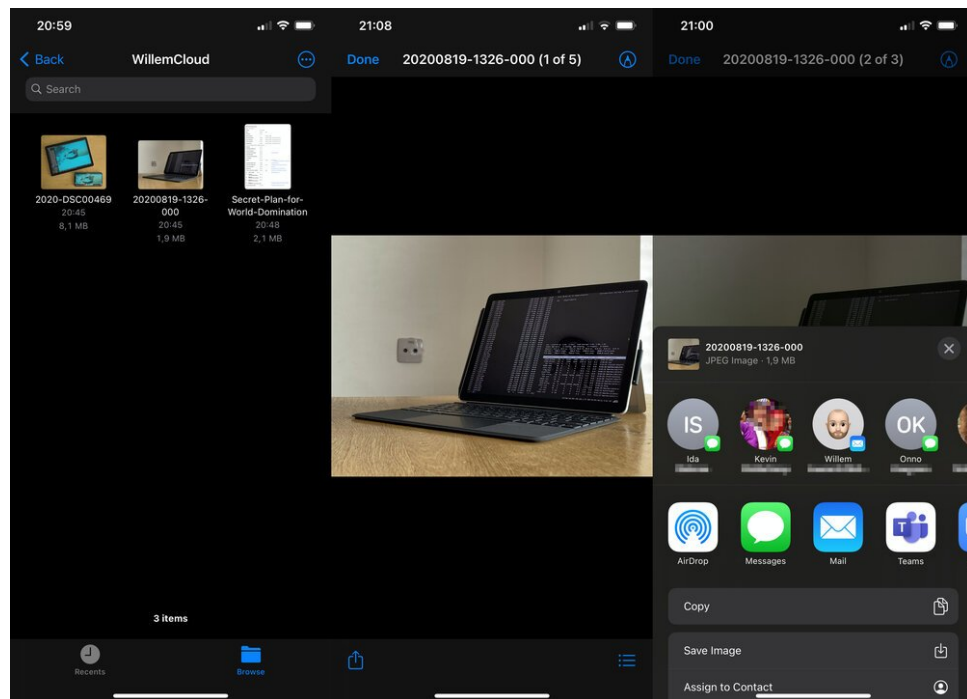
Once you have your computer synchronising files to your NAS, you should setup your smartphone. I use an iPhone and therefore looked for the best app that integrates well with the native iOS files app. After testing different apps I found [Secure ShellFish](#) to be the best:



Download and configure Secure ShellFish to connect to my "WillemCloud"

The beautiful thing of the Secure ShellFish app is that it integrates into the native iOS File app (and the underlying iOS storage provider API's). This means that all apps

that access local storage can connect with your NAS! This enables access to your files from practically any app.

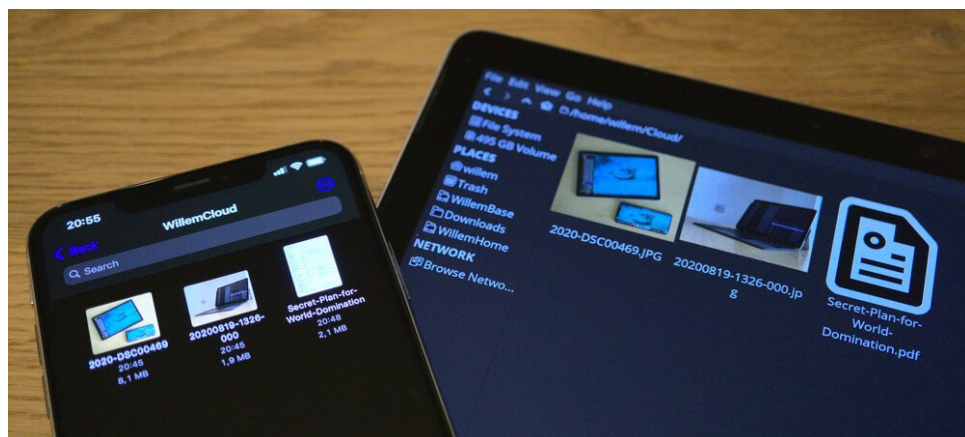


Accessing files using the native iOS Files app enables you to use many different apps and shortcuts

Conclusion

Optimising for your daily workflows can really provide you with huge benefits in the long run. By making it possible to seamlessly share files between my tablet and smartphone I did just that!

By automating the synchronisation, it moves into the background. I no longer have to think about it as an advanced process involving different protocols, servers and apps. It just works, and that's exactly how I like it!



The same files on both my smartphone and tablet - it just works!