Science-IT Data Management Cheatsheet v1.4

Full info: https://wiki.aalto.fi/display/Triton/Data+management
General Aalto info: http://www.aalto.fi/en/research/research_data_management/

Basics

- There is great value in data, but if it is not handled value can be lost. Society now recognizes this and requires good data management all around.
- Funding agencies require good data management for future funding. Start now.
- New projects should request dedicated storage directories.
- **Data should be stored in project directories, not home directories.**
- Projects: Senior staff can request new projects, see CS-IT wiki. To get access to data, find project name, mail CS-IT and cc data owner for approval. See CS-IT wiki.
- Best to have one project per research theme, with a defined end date.
- We will accommodate almost any data as long as it is being managed properly.
- Ask us if you have any questions.

Organization strategies

- Separate data by type for proper management. Keeping different types of data separate is the most important step to take at the start.
    - For example: Code vs data, original data vs intermediate files, final results vs other, for-archival vs to-delete, can-be-opened vs confidential.
- Whatever you do, don't copy code. Have master repository in a VCS.
- Back up original data into /m/$dept/archive/.
- Traditional project - basic arrangement for one user on one project.
    - $proj/code/ - code, primary work. Backup to VCS.
    - $proj/original/ - original data. Backup to archive.
    - $proj/scratch/ - intermediate files. Replaceable with code+original.
    - $proj/doc/ - final results, final data, etc. Backup to archive when done.
- Multi-user project - users have dirs organized as above, original data can be shared.
    - $proj/$user1/…, $proj/$user2/… user's directories as above, code synced with VCS.
    - $proj/original/, $proj/scratch/ - shared files among all users.
- Master project - one project per research group with sub-projects.
    - $proj/$theme/$user/ - for organized themes.
    - $proj/$user/$theme/ - user's independent work.

Archival / deleting / opening

- You need to be able to end-of-life your data, otherwise it happens when disks die.
- Large amount of data can't be stored forever. Separate important from replaceable.
- The easiest way to ensure you always have data is to open it and put in a long-term public repository. Funders encourage this. Zenodo is recommended.

Types of data

Different types of data have different needs. Consider this and keep them separate, so some can be backed up, some deleted. Keep data organized from the start.

Requirements: L=large, F=fast, C=confidential, BU=backups, LT=long term archive/backup.
O=important

	L	F	C	BU	LT	
Code				O	O	Code and processing scripts
Documentation				O	O	Info on date, code, how to do your work.
Original data	o	o	O		O	Irreplaceable data
Intermediate files	O	O	O			code + original = intermediate, replaceable
Published results					O	Final results, papers, code. To archive and

Compare requirements to features on next table. Match data to location.

Data storage

Not all storage suits everything. Put the right data in the right place.

Qualities: L=large, F=fast, C=confidential, BU=backups, LT=long term backup, S=shareable
O=best, o=good, X=bad, **bold**=core services, use these if unsure.

	L	F	C	BU	LT	S	
$HOME			O				Home dir, 10GB
/m/$dept/project/$project/	**o**	**o**	**O**	**O**		**O**	Typical project files, 10-100s of GB.
/m/$dept/archive/$project/	**o**	**o**	**O**	**O**	**o**	**O**	Per-project archive., 10-100s of GB
/m/$dept/scratch/$project/	**O**	**O**	**o**	**X**	**X**	**O**	Triton, large, not backed up. 10-100sTB
/m/$dept/work/$username/	**O**	**O**	**o**	**X**	**X**		Like scratch but per-user.. 10-100sTB
/l/	o	OO					Local disk storage. Not backed up.
https://version.aalto.fi				**O**		**O**	Aalto git repository
Aalto laptops			X	X	X		High risk of data loss
External drives, USB, etc			X	X	X		High risk of data loss
$HOME/public_html/				O		O	Webspace (https://users.aalto.fi/~username)
CSC IDA service	O		o		O	O	Long-term archival by CSC
https://filesender.funet.fi						O	Cloud-based file sending. 50GB
Zenodo (EU project)			X		O	O	EU project for long-term open data
Public services (google, dropbox)			X		X		Be careful, there is no service or confidentiality guarantee.
Own computers			X	X	X		Unmanaged and risky.

Scratch and work require Triton accounts.

Version control

- Code and related data should be in a version control system. Learn one well.
- Git is most common: https://git-scm.com and best supported, but there are others.
- At Aalto, private repositories can be hosted an https://version.aalto.fi