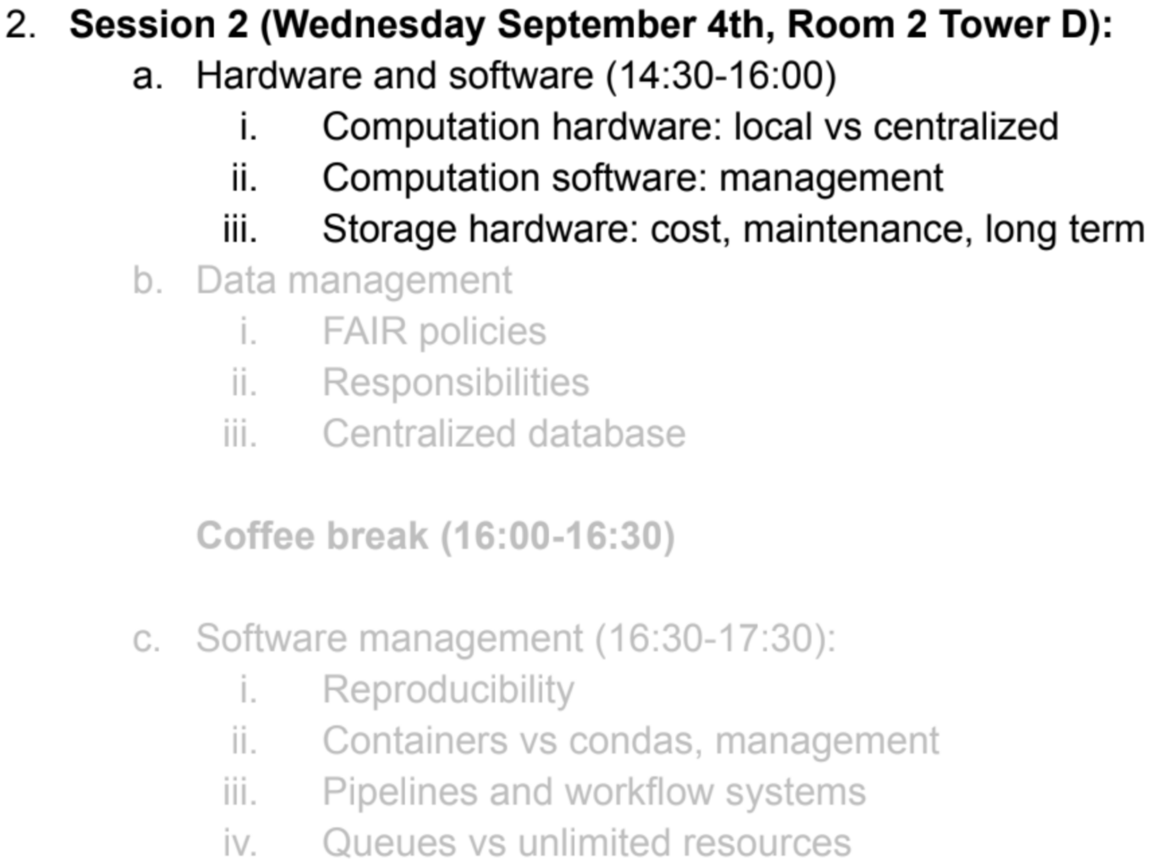
Session 2a – Hardware and Software



2. **Session 2 (Wednesday September 4th, Room 2 Tower D):**
    a. Hardware and software (14:30-16:00)
        i. Computation hardware: local vs centralized
        ii. Computation software: management
        iii. Storage hardware: cost, maintenance, long term
    b. Data management
        i. FAIR policies
        ii. Responsibilities
        iii. Centralized database

    Coffee break (16:00-16:30)

    c. Software management (16:30-17:30):
        i. Reproducibility
        ii. Containers vs condas, management
        iii. Pipelines and workflow systems
        iv. Queues vs unlimited resources

Local (facility) hardware

Local (biostats) hardware

(1) Local hardware

- Officer workstations / laptops (Mac by tradition, anything with Linux and a lot of RAM is fine)
- ~999 cables and adapters
- GOOD chair, screen, mouse and keyboard

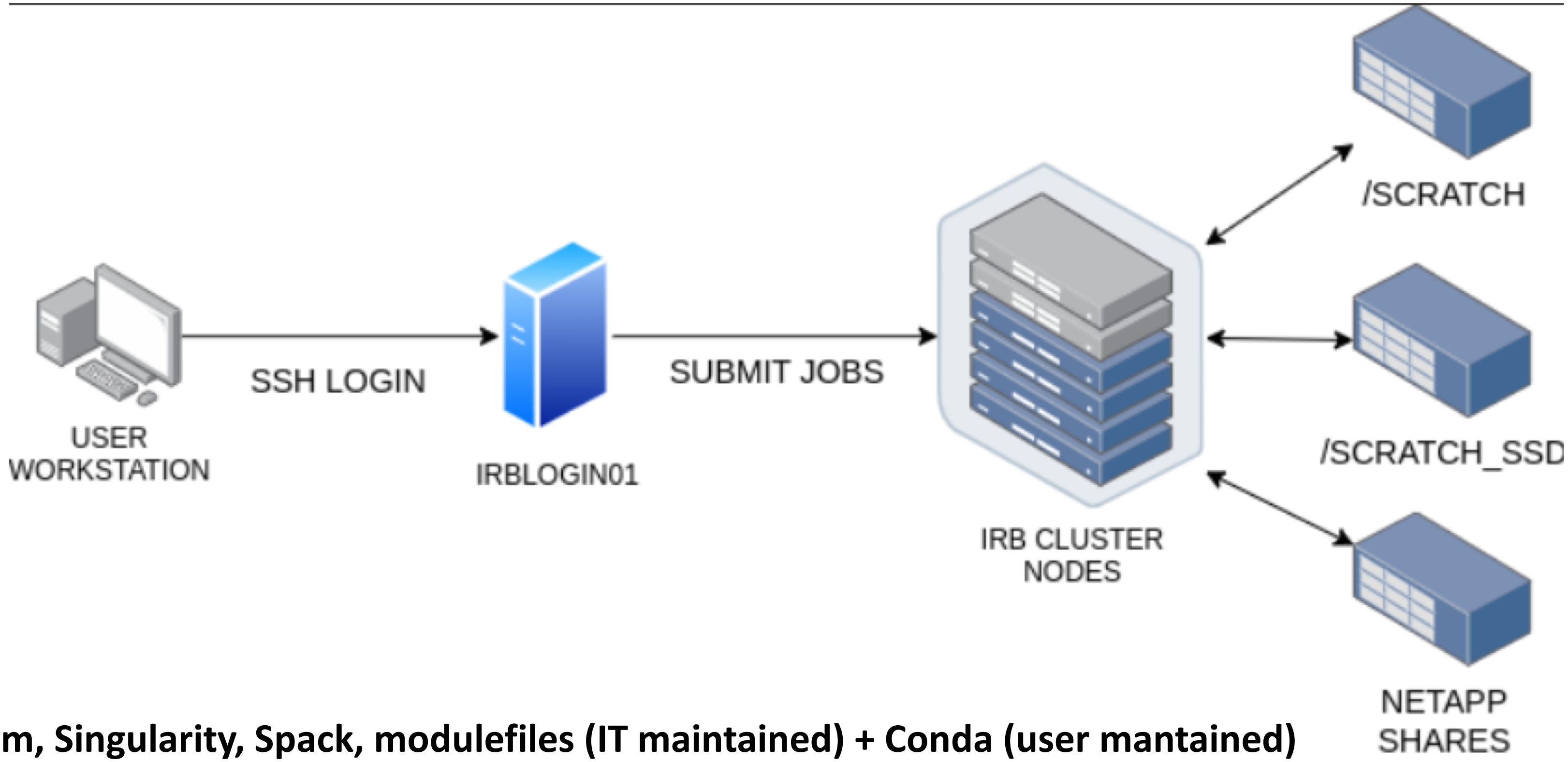
(2) Computing (FAT servers)

- Both Rocky Linux 64 bit
- Aurora (end-of-life): AMD Opteron 64 bits 4x12 Cores, RAM ~0.75 TB
- Moana: AMD EPYC 2x28 Cores @3.5GHz, RAM ~1.8 TB DIMM DDR4
- Both: local HDD/SSD (system, homes, etc) + fast scratch disk

(3) Virtual (utility) servers

- Wiki, website, light shiny-like apps
- Users reports (apache, internal and external)

Institutional (IRB) hardware environment



Slurm, Singularity, Spack, modulefiles (IT maintained) + Conda (user mantained)

For all Institute users (with special queues for facility)

ssh + salloc OR On-demand (Rstudio, Jupyter)

HPC Computing: IRBCluster

Centralized (Institutional) Hardware: IRBCluster

The IRB cluster is based on Intel Xeon Platinum processors from the Ice Lake generation, with NVIDIA's Mellanox high-performance network interconnect, and running Rocky Linux 8.7 as the operating system.
This general-purpose cluster consists of 53 CPU nodes and 5 GPU nodes, with a total of 6.21k CPU cores and 34 TB of RAM.
All compute nodes consist of 2 numa nodes in total per node.

CPU Nodes (Only 8 Icelake nodes shown):

- 8 nodes **(irbccn[01-08]**)
  - Processor: Intel Xeon Platinum 8358 CPU @ 2.60GHz
  - Architecture: Ice Lake
  - Cores: 64 per node (2 sockets, 32 cores each)
  - RAM: 1024 GB DDR4 3200 MT/s (irbccn[01-03]), 256 GB DDR4 3200 MT/s (irbccn[04-08])
  - Cache Size: L1d: 48 KB, L1i: 32 KB, L2: 1280 KB, L3: 48 MB
  - Network: NVIDIA Mellanox 25 Gbit/s PCI-E adapter
  - Local Storage: 350 GB SSD (temporary storage)

Software management & maintenance

- **System-level software (still useful):**
  - Emacs + ESS + Screen / Tmux (steep learning curve but very flexible)
  - All Linux command tools
  - R + CRAN / BioC / Other libraries (compiled, several versions, not always ideal)
  - Nextflow + Docker/Singularity
  - Other software: trimming, fastqc, aligners, really anything
  - Limited flexibility (devel versions), requires maintenance and discipline
  - Dependencies !

- **User-level software (conda):**
  - Custom and Shared conda recipes / environments
  - Solves dependency nightmares & system-level screw-ups
  - Almost unlimited flexibility, great to test devel libs
  - Beware of storage, caches, duplicated software packs

- **User-level software (Nextflow / nf-core):**
  - 'Complete' reproducibility
  - Beware of storage

- **Software development (Rlibs)**
  - R / Bioconductor
  - Github + Zenodo

Storage: costs & considerations

Storage

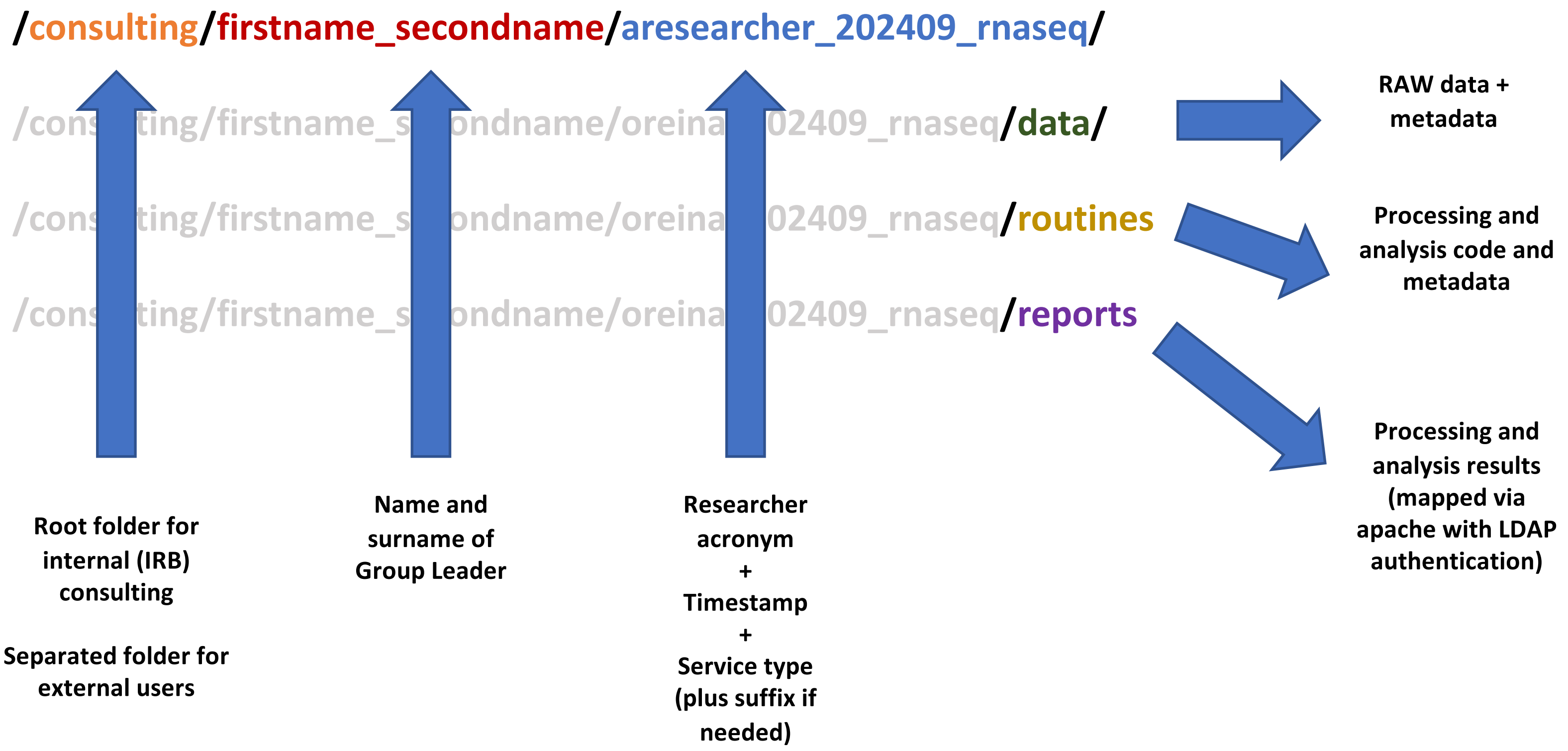
- **Overview**
  - 2 x Netapp NFS/SMB Volumes
  - biostats: 60 TB, work / critical data volume
  - ga: XX TB, scratch / utility volume (reused from Illumina Genome Analyzer)
  - Cloud/Tape mid/long term storage

- **Cost**
  - Per-year, paid to ITS
  - Includes maintenance & backup (snapshot) with customizable periodicity
  - Keep it always in mind (upgrade with plenty of advance)
  - Consider charging users for mid/long term storage/management

- **Long-term storage approaches (data)**
  - It is always a delicate balance
  - Radical approach: on publishing RAW data goes to FAIR repository, all else: rm -rf *
  - Hyper-friendly approach: Keep (almost) everything
  - Sensible approach: ensure reproducibility & make your life easier

Our consulting folder structure

Biostats filesystem structure (research Project data, code and results)



/consulting/firstname_secondname/aresearcher_202409_rnaseq/

/cons ting/firstname_s ondname/oreina 02409_rnaseq/data/

/cons ting/firstname_s ondname/oreina 02409_rnaseq/routines

/cons ting/firstname_s ondname/oreina 02409_rnaseq/reports

RAW data + metadata

Processing and analysis code and metadata

Processing and analysis results (mapped via apache with LDAP authentication)

Root folder for internal (IRB) consulting

Separated folder for external users

Name and surname of Group Leader

Researcher acronym
+
Timestamp
+
Service type (plus suffix if needed)

Session 2b – Research Data Management

2. **Session 2 (Wednesday September 4th, Room 2 Tower D):**
   a. Hardware and software (14:30-16:00)
      i. Computation hardware: local vs centralized
      ii. Computation software: management
      iii. Storage hardware: cost, maintenance, long term
   b. Data management
      i. FAIR policies
      ii. Responsibilities
      iii. Centralized database

   Coffee break (16:00-16:30)

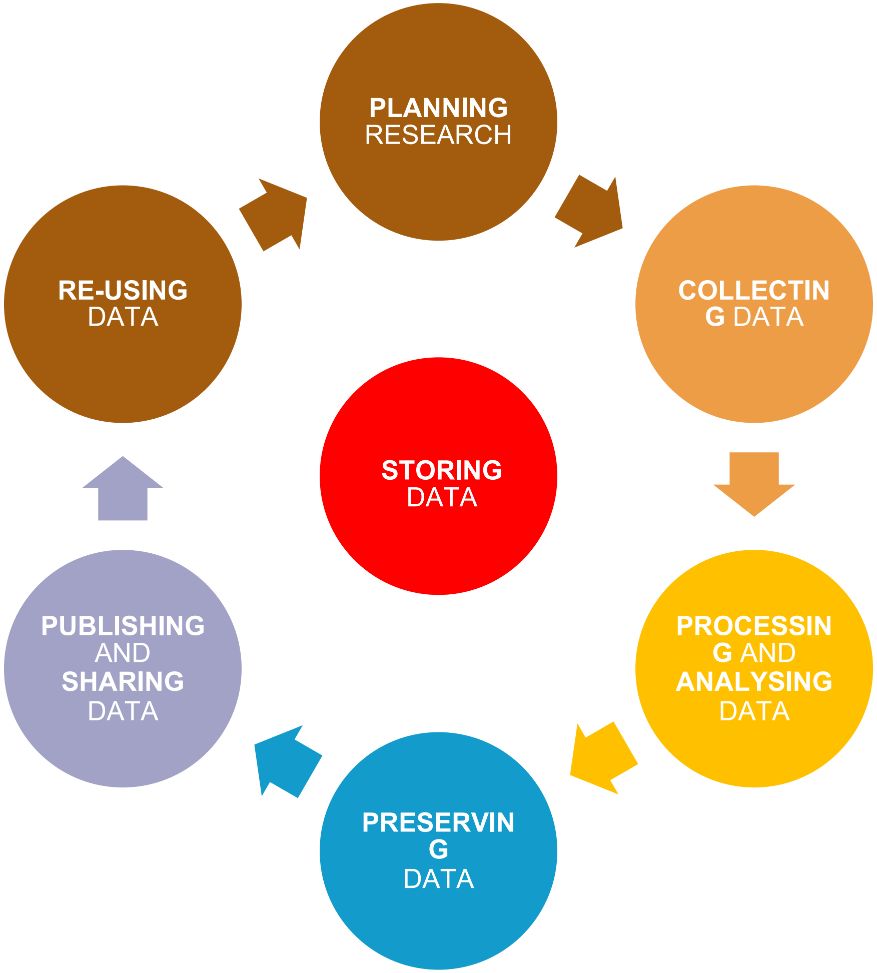
   c. Software management (16:30-17:30):
      i. Reproducibility
      ii. Containers vs condas, management
      iii. Pipelines and workflow systems
      iv. Queues vs unlimited resources

The Research Data Life Cycle + FAIR

Each phase of the data life cycle for every research Project presents its own **practical challenges** regarding

Research Data Management

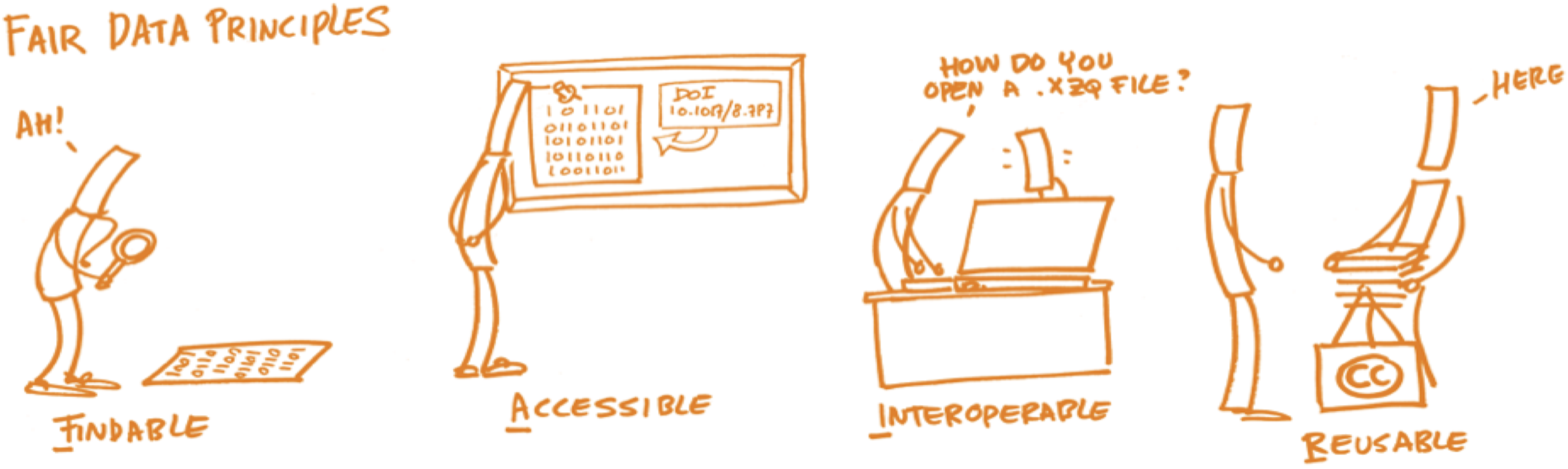
In 2016, the '[FAIR Guiding Principles for scientific data management and stewardship'](#) were published in *Scientific Data*. The authors intended to provide guidelines to improve the **F**indability, **A**ccessibility, **I**nteroperability, and **R**euse of digital assets.



Adapted from: https://www.youtube.com/watch?v=OL_Vd9dd-AQ

For more: https://rdmkit.elixir-europe.org

How to FAIRify my data ?

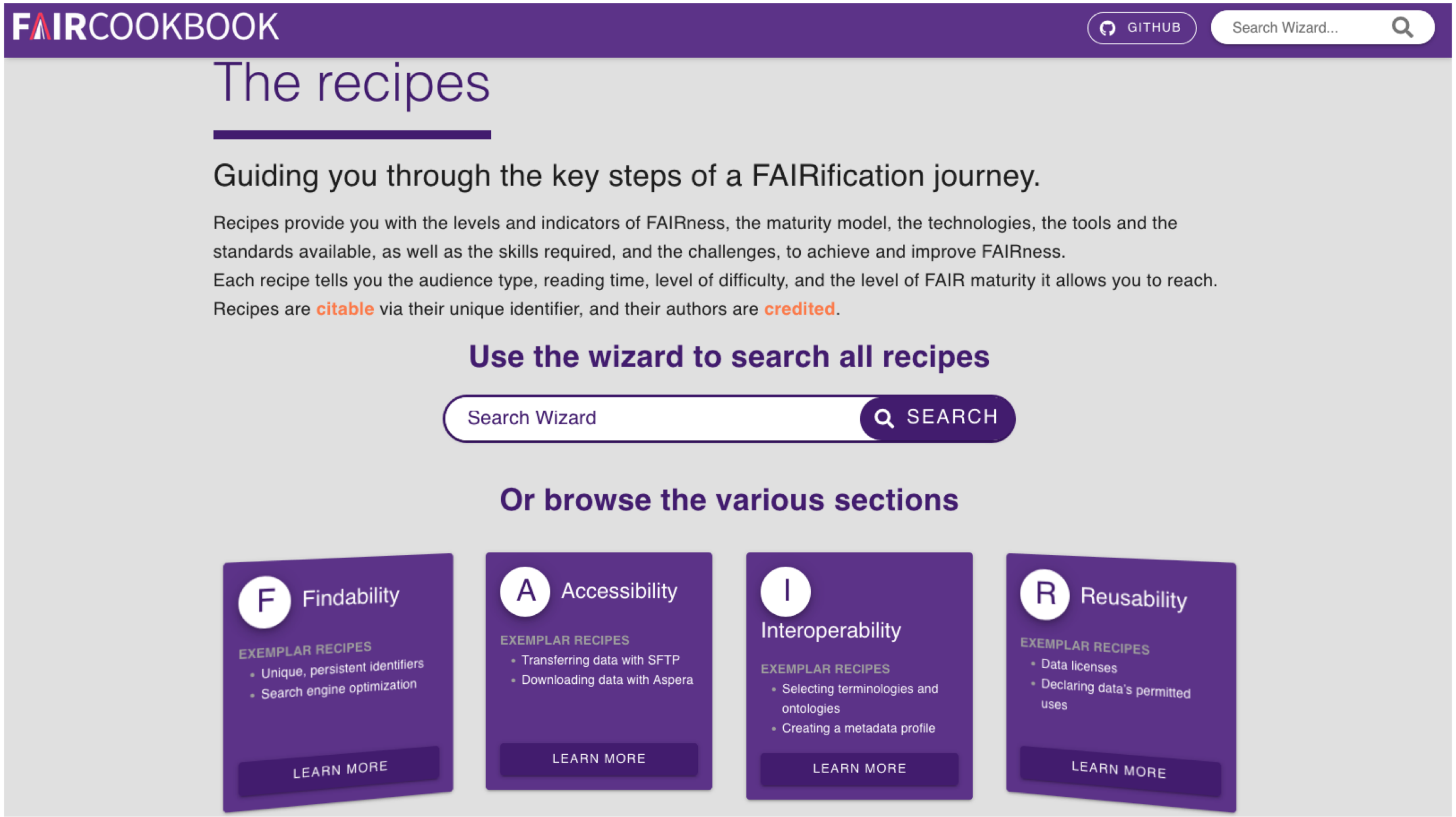


https://www.go-fair.org/fair-principles/

It all starts with (good) metadata

Useful tools for FAIRification

https://faircookbook.elixir-europe.org/



The recipes

Guiding you through the key steps of a FAIRification journey.

Recipes provide you with the levels and indicators of FAIRness, the maturity model, the technologies, the tools and the standards available, as well as the skills required, and the challenges, to achieve and improve FAIRness.
Each recipe tells you the audience type, reading time, level of difficulty, and the level of FAIR maturity it allows you to reach.
Recipes are citable via their unique identifier, and their authors are credited.

Use the wizard to search all recipes

Or browse the various sections

F Findability

EXEMPLAR RECIPES
- Unique, persistent identifiers
- Search engine optimization

A Accessibility

EXEMPLAR RECIPES
- Transferring data with SFTP
- Downloading data with Aspera

I Interoperability

EXEMPLAR RECIPES
- Selecting terminologies and ontologies
- Creating a metadata profile

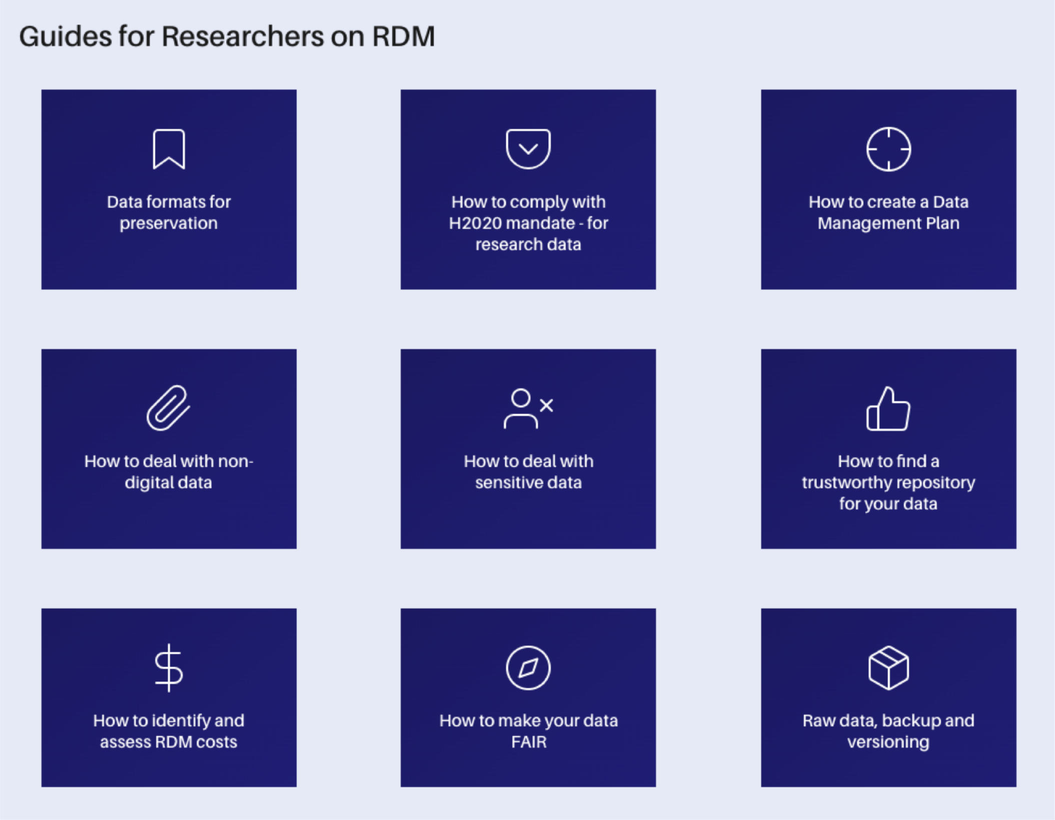
R Reusability

EXEMPLAR RECIPES
- Data licenses
- Declaring data's permitted uses

https://faircookbook.elixir-europe.org/content/search-wizard.html

Useful tools for FAIRification

https://www.openaire.eu



Guides for Researchers on RDM

Data formats for preservation

How to comply with H2020 mandate - for research data

How to create a Data Management Plan

How to deal with non-digital data

How to deal with sensitive data

How to find a trustworthy repository for your data

How to identify and assess RDM costs

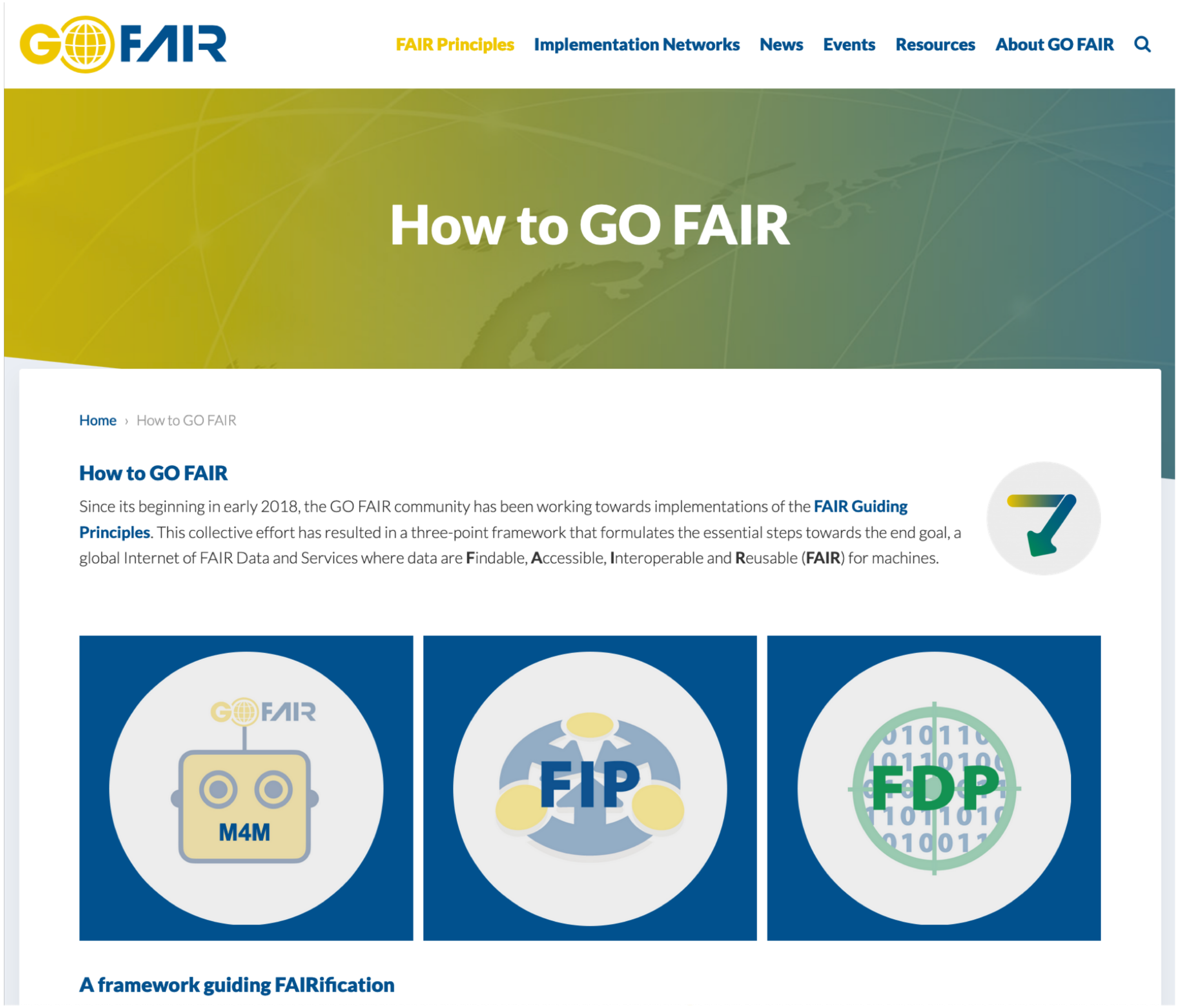
How to make your data FAIR

Raw data, backup and versioning

https://www.openaire.eu/guides

Useful tools for FAIRification

https://www.go-fair.org/how-to-go-fair/



How to GO FAIR

Home › How to GO FAIR

How to GO FAIR

Since its beginning in early 2018, the GO FAIR community has been working towards implementations of the **FAIR Guiding Principles**. This collective effort has resulted in a three-point framework that formulates the essential steps towards the end goal, a global Internet of FAIR Data and Services where data are **F**indable, **A**ccessible, **I**nteroperable and **R**eusable (**FAIR**) for machines.

A framework guiding FAIRification

Useful tools for FAIRification

https://the-turing-way.netlify.app/index.html

Fig. 1 The Turing Way project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

Good practices for good metadata

- Use **folders** and **subfolders** to reflect data organization
- Name folders and files appropriately with **explicit names**
- Add **version control, README files, etc**
- Use **YYYYMMDD** for dates
- **Avoid** spaces and unnecessary special characters
- Be **consistent**
- Separate **ongoing** and **completed** work

Practical guidelines on file naming conventions:

Harvard: https://datamanagement.hms.harvard.edu/collect/file-naming-conventions

Princeton: https://researchdata.princeton.edu/research-lifecycle-guide/file-organization

Stanford: https://www.sup.org/digital/authors/current/docs/FileNamesFormats.pdf

Data and Metadata standards

Use good, complete metadata and consider **metadata cores and standards**

- Dublin Core generic metadata
  https://www.dublincore.org/specifications/dublin-core/dcmi-terms
- DCC Discipline-related metadata
  http://www.dcc.ac.uk/resources/metadata-standards

Get familiar with standard vocabularies / ontologies (syntax) for your research data and field

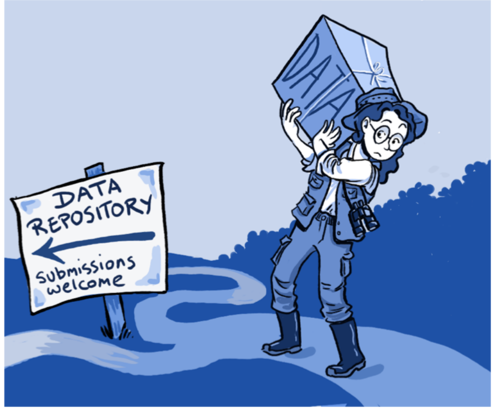
- Ensure consist **spelling** and meaning of (for instance) **keywords**
- Check the corresponding **ontology** https://www.ebi.ac.uk/ols/index

FAIR data repositories

What to share and what to preserve, from the perspective of **reproducibility** and **scientific integrity**

In FAIR repository (when possible):

- **Data**: Raw, (processed), analyzed
- **Metadata** for Data
- Metadata for Data **Processing / Analyses**
- Link / DOI for **publication**(s) (if applicable)
- **Data Management Plan (!)**



When in doubt, check
https://www.re3data.org
https://fairsharing.org

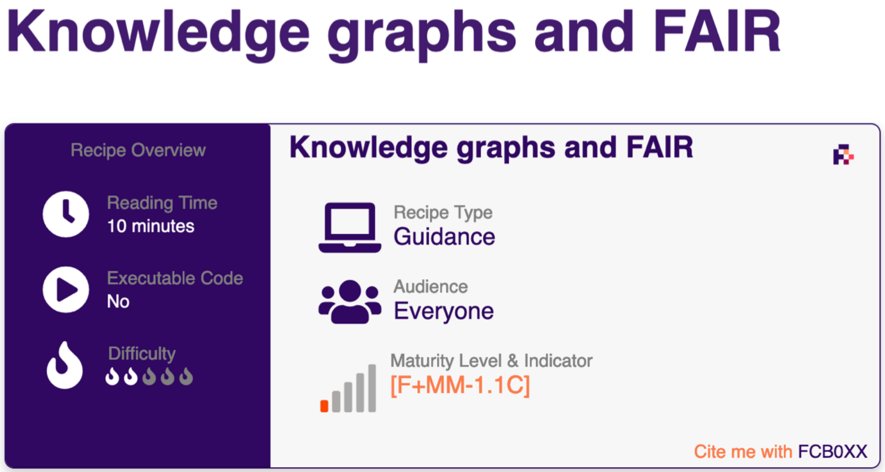
What, where and when to deposit (and who)

What	Where	When	Who
RAW data + Metadata	Specific repositories whenever possible www.re3data.org/, fairsharing.org, ELIXIR deposition databases)	Anytime, and at least before paper submission (required)	Check with the facility generating / analyzing the data
Publication-related final figures, data and results tables, supplementary materials of any kind	Journal-provided data entry or check with journal for links to accepted generic repositories (CSUC > Others)	At paper submission time (at least)	You (or your PI / other author)
Programming code and related documentation, tutorials, etc	Recommended code repository (GitHub, GitLab, etc) + DOI or/and journal-provided materials or platform (e.g. CodeOcean)	At anytime / at paper submission time (can be required !). Consider versioning / code freeze.	The developer of the code or the expert who reused existing workflows (e.g. on Galaxy)
Other generic data	Generic repository (CSUC > Others)	At anytime (check periodically)	You (or your PI / other author)

Institutional (or facility) data catalog

Slowly going there…

- FAIR data Knowledge graphs
- Data meta-indexers (Dataverse, Mendeley data…)
- Research Data Management systems (Renku / Gatekeeper)
- FAIR Assessment tools



Knowledge graphs and FAIR

Recipe Overview

Reading Time
10 minutes

Executable Code
No

Difficulty
●●○○○

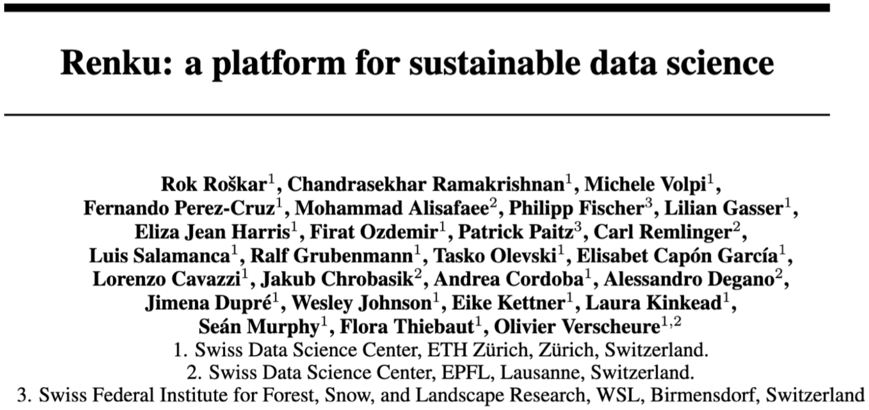
Knowledge graphs and FAIR

Recipe Type
Guidance

Audience
Everyone

Maturity Level & Indicator
[F+MM-1.1C]

Cite me with FCB0XX



Renku: a platform for sustainable data science

Rok Roškar[1], Chandrasekhar Ramakrishnan[1], Michele Volpi[1],
Fernando Perez-Cruz[1], Mohammad Alisafaee[2], Philipp Fischer[3], Lilian Gasser[1],
Eliza Jean Harris[1], Firat Ozdemir[1], Patrick Paitz[3], Carl Remlinger[2],
Luis Salamanca[1], Ralf Grubenmann[1], Tasko Olevski[1], Elisabet Capón García[1],
Lorenzo Cavazzi[1], Jakub Chrobasik[2], Andrea Cordoba[1], Alessandro Degano[2],
Jimena Dupré[1], Wesley Johnson[1], Eike Kettner[1], Laura Kinkead[1],
Seán Murphy[1], Flora Thiebaut[1], Olivier Verscheure[1,2]
1. Swiss Data Science Center, ETH Zürich, Zürich, Switzerland.
2. Swiss Data Science Center, EPFL, Lausanne, Switzerland.
3. Swiss Federal Institute for Forest, Snow, and Landscape Research, WSL, Birmensdorf, Switzerland

Coffee !

2. **Session 2 (Wednesday September 4th, Room 2 Tower D):**

a. Hardware and software (14:30-16:00)

i. Computation hardware: local vs centralized

ii. Computation software: management

iii. Storage hardware: cost, maintenance, long term

b. Data management

i. FAIR policies

ii. Responsibilities

iii. Centralized database

Coffee break (16:00-16:30)

c. Software management (16:30-17:30):

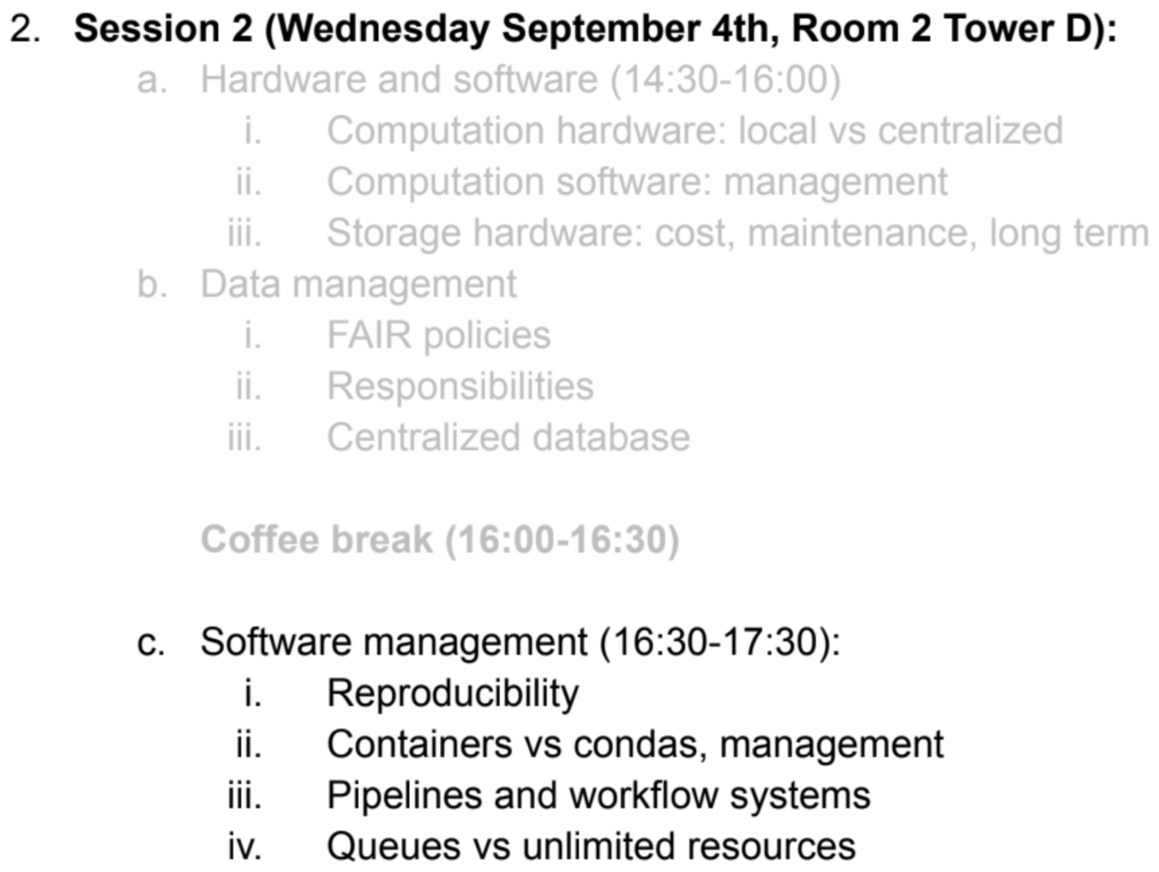
i. Reproducibility

ii. Containers vs condas, management

iii. Pipelines and workflow systems

iv. Queues vs unlimited resources

Section 2c – Software management



2. **Session 2 (Wednesday September 4th, Room 2 Tower D):**
   a. Hardware and software (14:30-16:00)
      i. Computation hardware: local vs centralized
      ii. Computation software: management
      iii. Storage hardware: cost, maintenance, long term
   b. Data management
      i. FAIR policies
      ii. Responsibilities
      iii. Centralized database

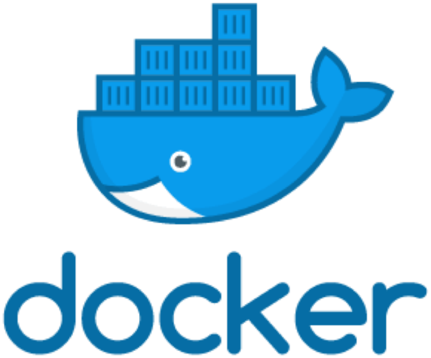
   Coffee break (16:00-16:30)

   c. Software management (16:30-17:30):
      i. Reproducibility
      ii. Containers vs condas, management
      iii. Pipelines and workflow systems
      iv. Queues vs unlimited resources

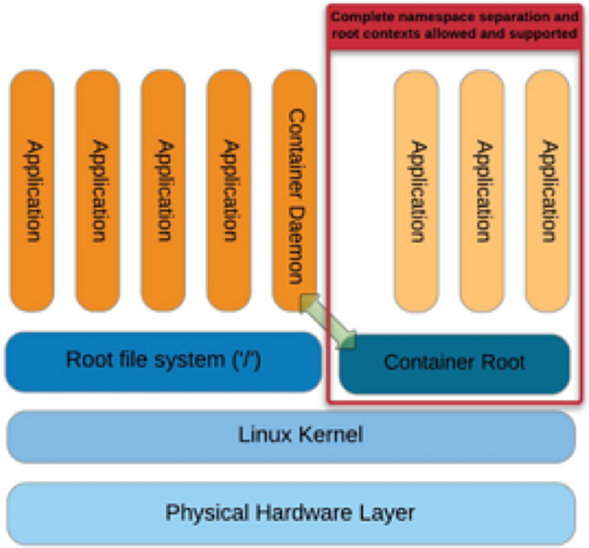
Reproducibility and “Replicability”

- R is for **Reproducible**
- **Data** and **metadata** (for both **data** and processing / **analysis**)
- **Software versions**
- **Random seeds**
- **Version control** for your code
- **Workflow** management systems
- Reproducibility of **biological** results and **conclusions**
- Do not **obsess** with perfect numeric reproducibility (99th decimal)
- Good data management habits can help a lot

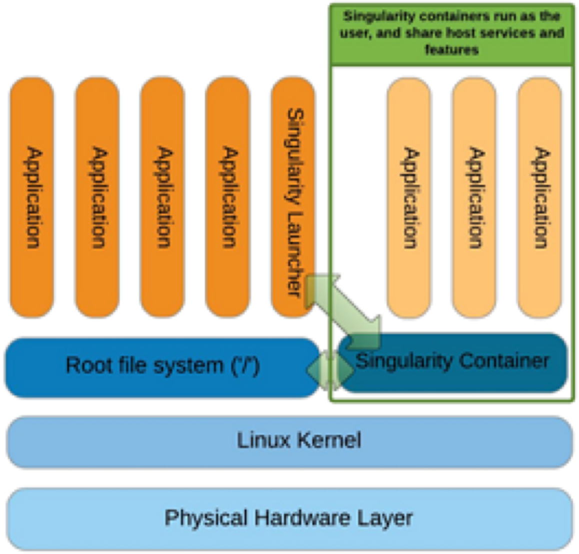
Tools for reproducibility: Containers



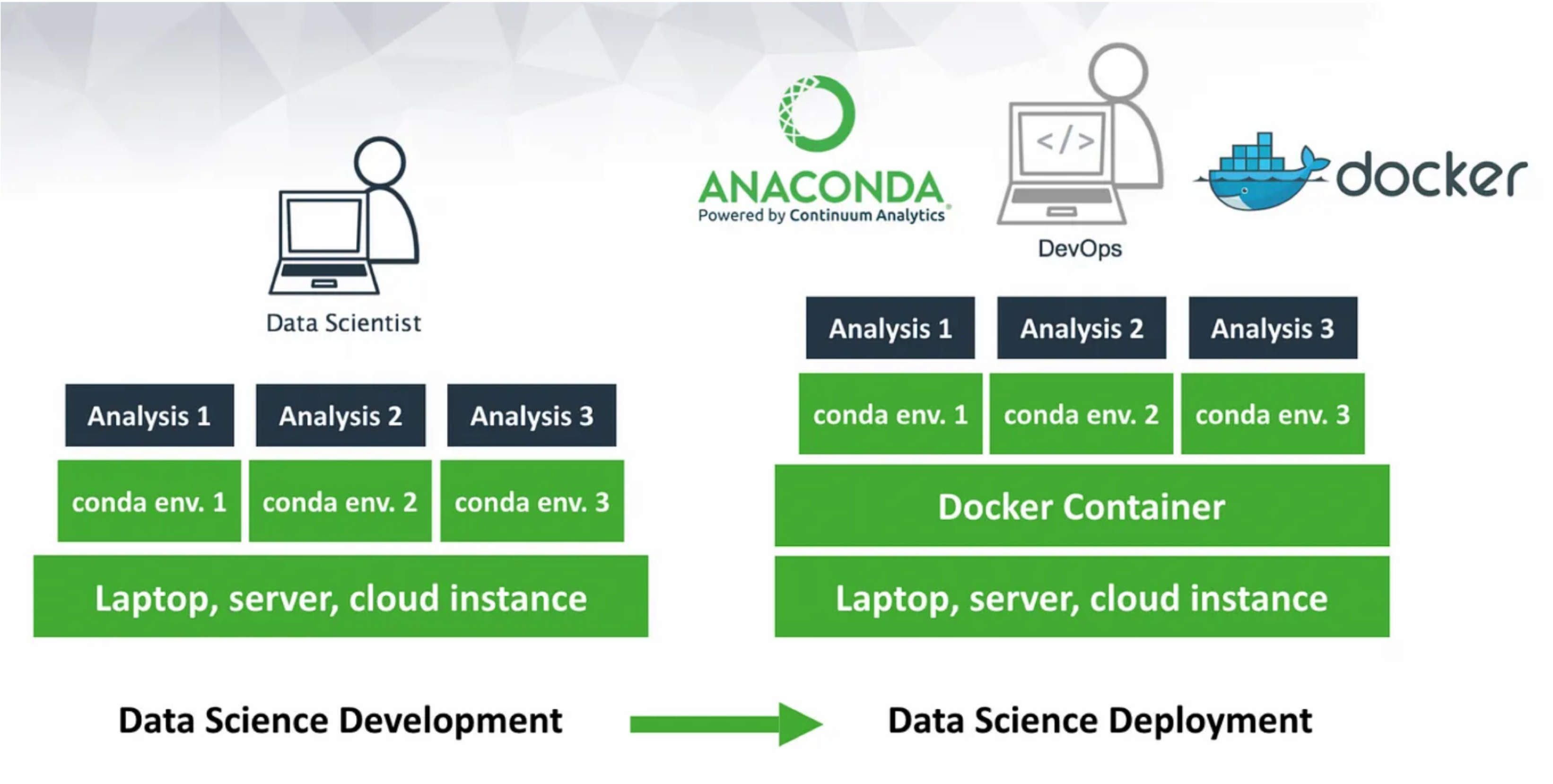
Docker



Singularity



Tools for reproducibility: Containers & Conda



Analysis 1 | Analysis 2 | Analysis 3

conda env. 1 | conda env. 2 | conda env. 3

Laptop, server, cloud instance

Data Science Development → **Data Science Deployment**

Analysis 1 | Analysis 2 | Analysis 3

conda env. 1 | conda env. 2 | conda env. 3

Docker Container

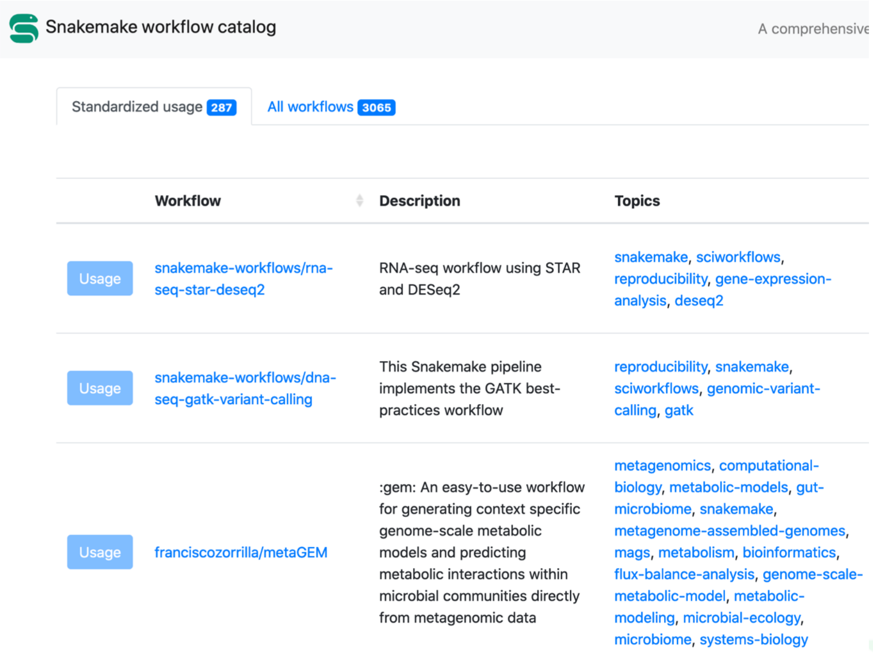
Laptop, server, cloud instance

https://medium.com/@patrickmichelberger/getting-started-with-anaconda-docker-b50a2c482139

Pipelines & workflow management systems

snakemake
A framework for reproducible data analysis

nextflow

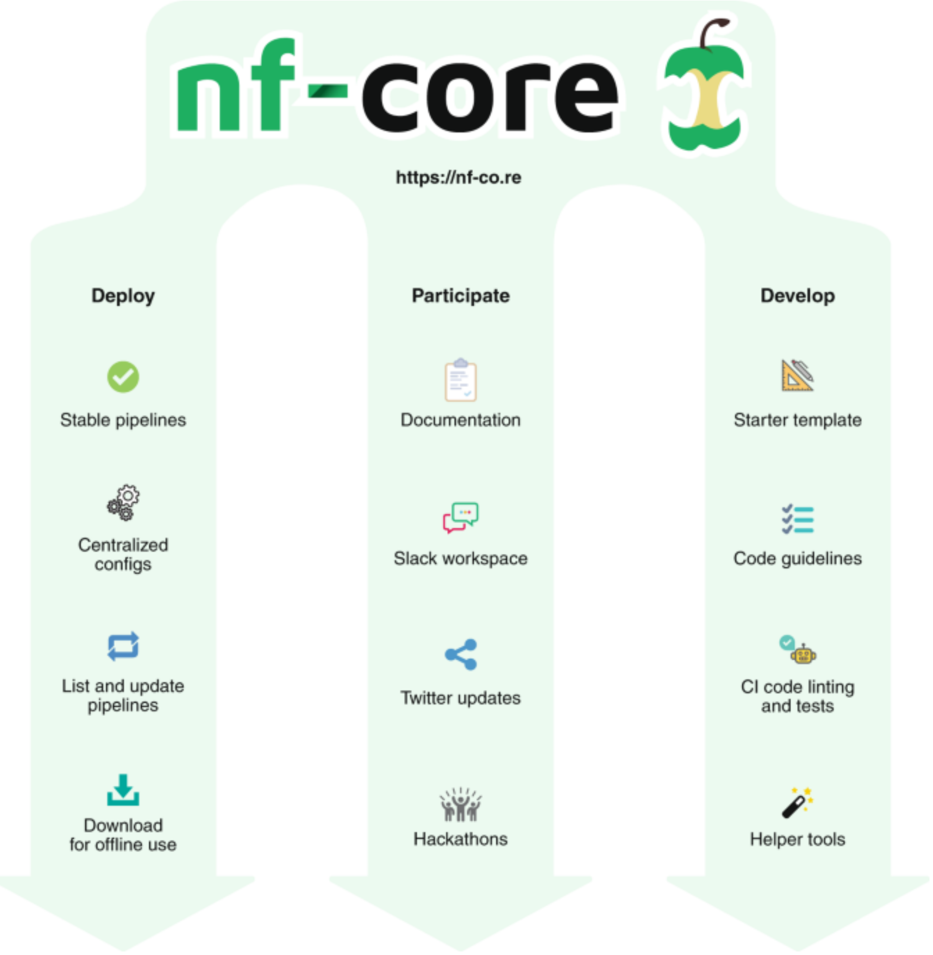


Snakemake workflow catalog

A comprehensive

Standardized usage 287 | All workflows 3065

	Workflow	Description	Topics
Usage	snakemake-workflows/rna-seq-star-deseq2	RNA-seq workflow using STAR and DESeq2	snakemake, sciworkflows, reproducibility, gene-expression-analysis, deseq2
Usage	snakemake-workflows/dna-seq-gatk-variant-calling	This Snakemake pipeline implements the GATK best-practices workflow	reproducibility, snakemake, sciworkflows, genomic-variant-calling, gatk
Usage	franciscozorrilla/metaGEM	:gem: An easy-to-use workflow for generating context specific genome-scale metabolic models and predicting metabolic interactions within microbial communities directly from metagenomic data	metagenomics, computational-biology, metabolic-models, gut-microbiome, snakemake, metagenome-assembled-genomes, mags, metabolism, bioinformatics, flux-balance-analysis, genome-scale-metabolic-model, metabolic-modeling, microbial-ecology, microbiome, systems-biology



nf-core

https://nf-co.re

Deploy
- Stable pipelines
- Centralized configs
- List and update pipelines
- Download for offline use

Participate
- Documentation
- Slack workspace
- Twitter updates
- Hackathons

Develop
- Starter template
- Code guidelines
- CI code linting and tests
- Helper tools

Queues vs “Unlimited” resources

- Our experience is mostly with facility managed and dedicated FAT servers
- Has many pros and several cons too
- Still a very valid approach if for intstance cost is shared by several facilities

- Currently starting to use IRBCluster
- Dedicated queue and special resource allocation
- Frees you from (most of) software management
- High level of software optimization, software stack is almost independent of OS

Questions, Comments, More coffee ?