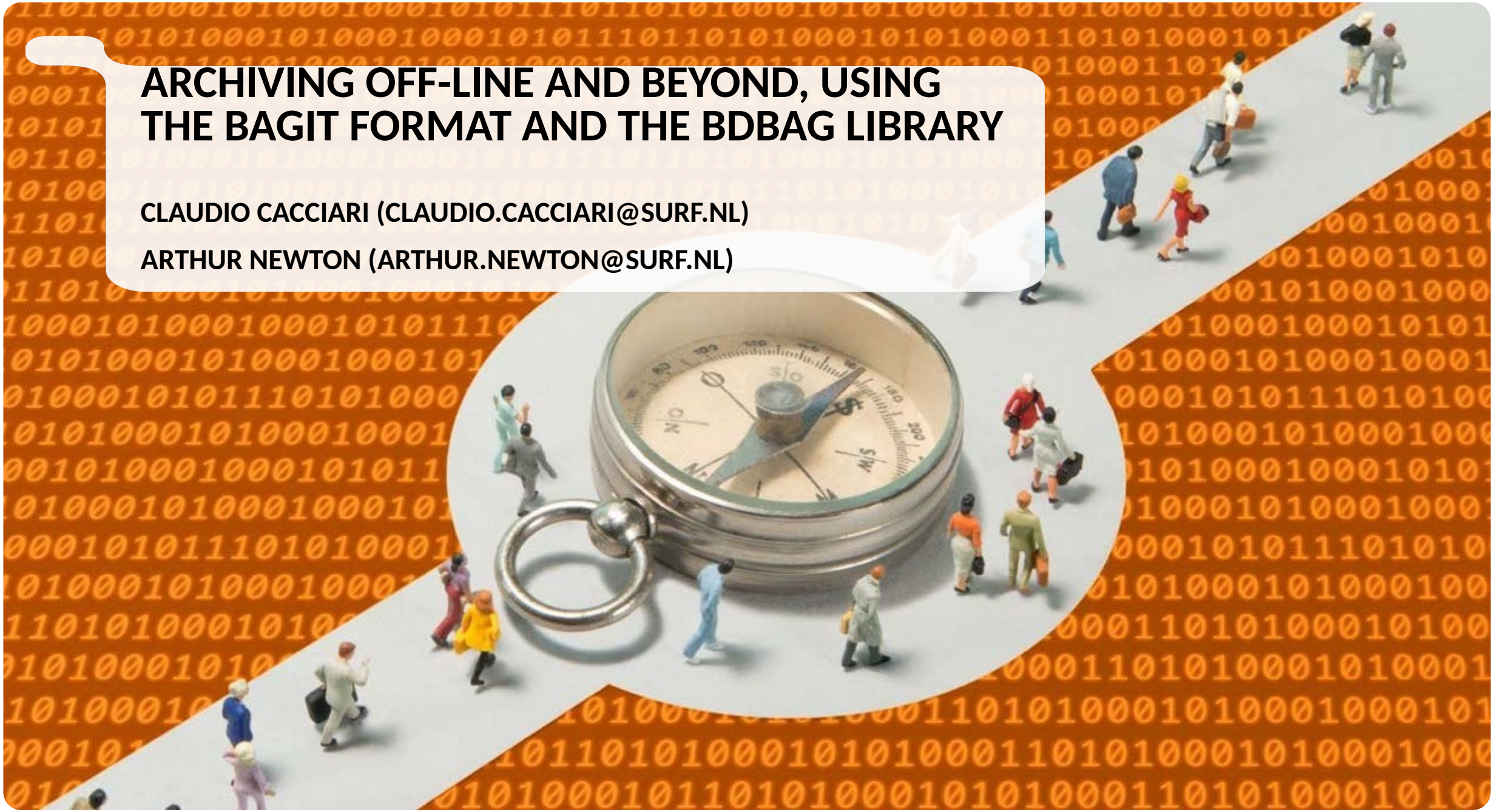
ARCHIVING OFF-LINE AND BEYOND, USING THE BAGIT FORMAT AND THE BDBAG LIBRARY

CLAUDIO CACCIARI (CLAUDIO.CACCIARI@SURF.NL)

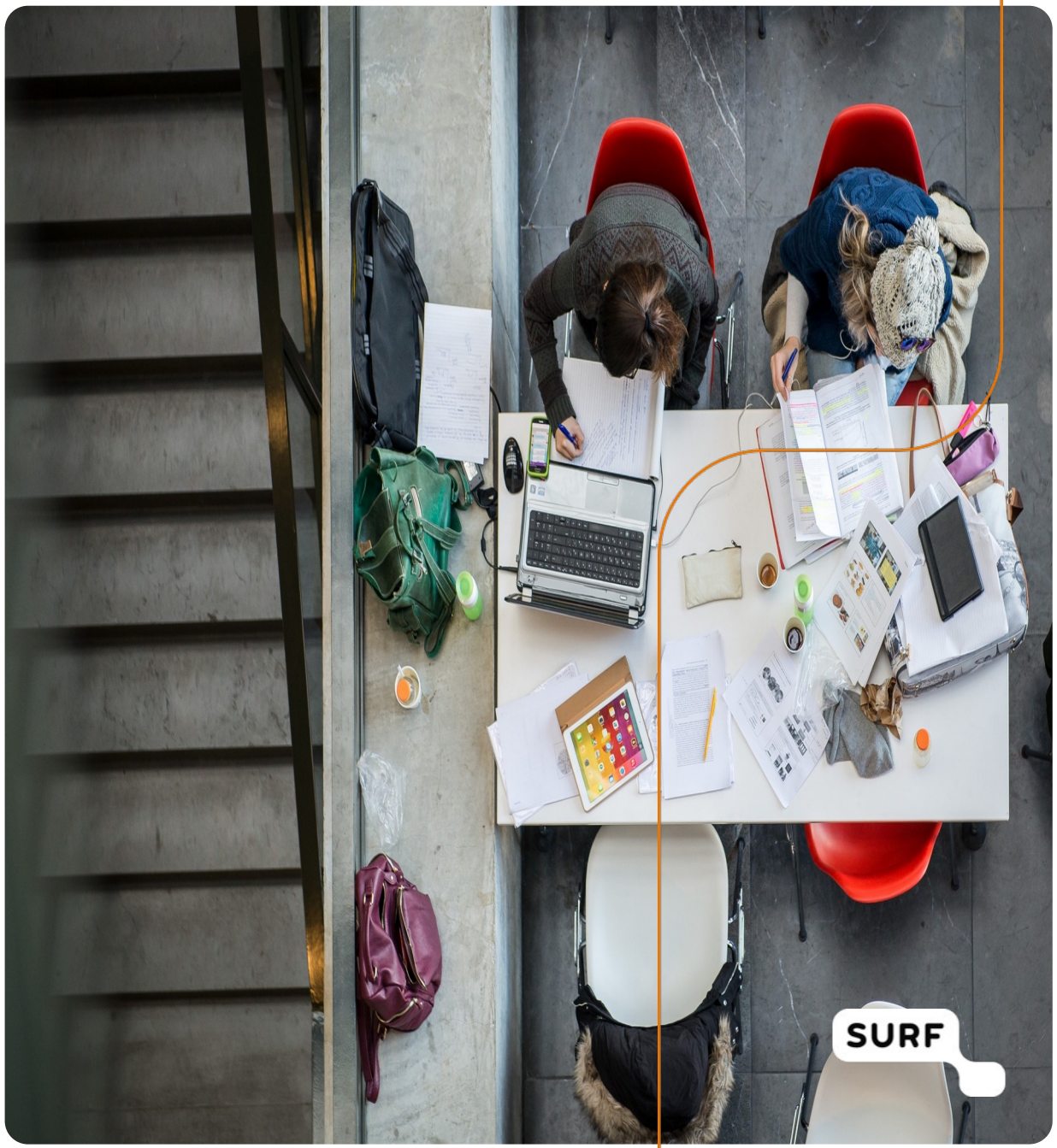
ARTHUR NEWTON (ARTHUR.NEWTON@SURF.NL)



- SURF
- Tape library and archiving data flow
- The issue with small files
- “Classic” approach
- Alternative: BagIt and bdbag
- How it works
- Benefits
- Potential

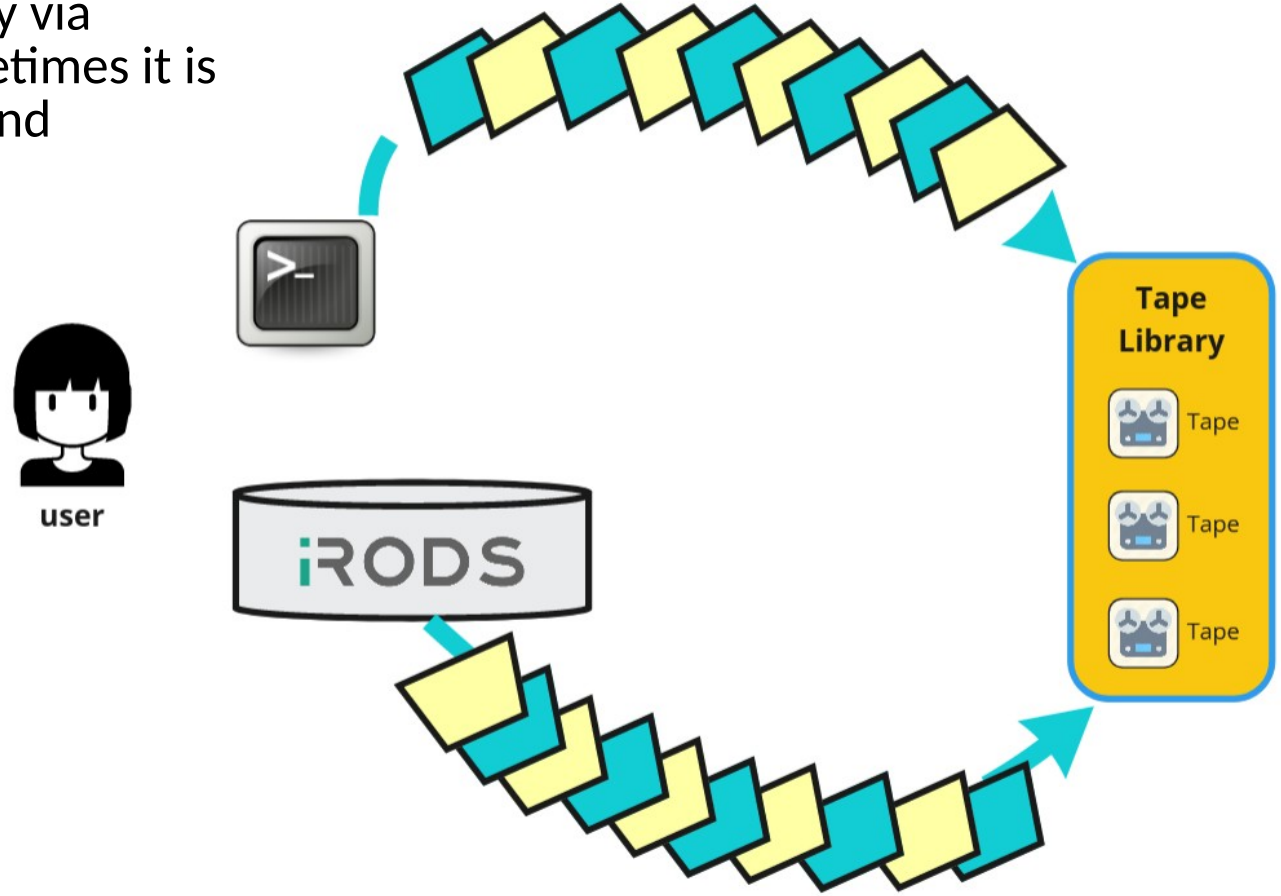
SURF is an ICT cooperative for education and research

At the SURF cooperative, education and research work together to make full use of the opportunities offered by digitalisation,
with the aim: making education and research better and more flexible.



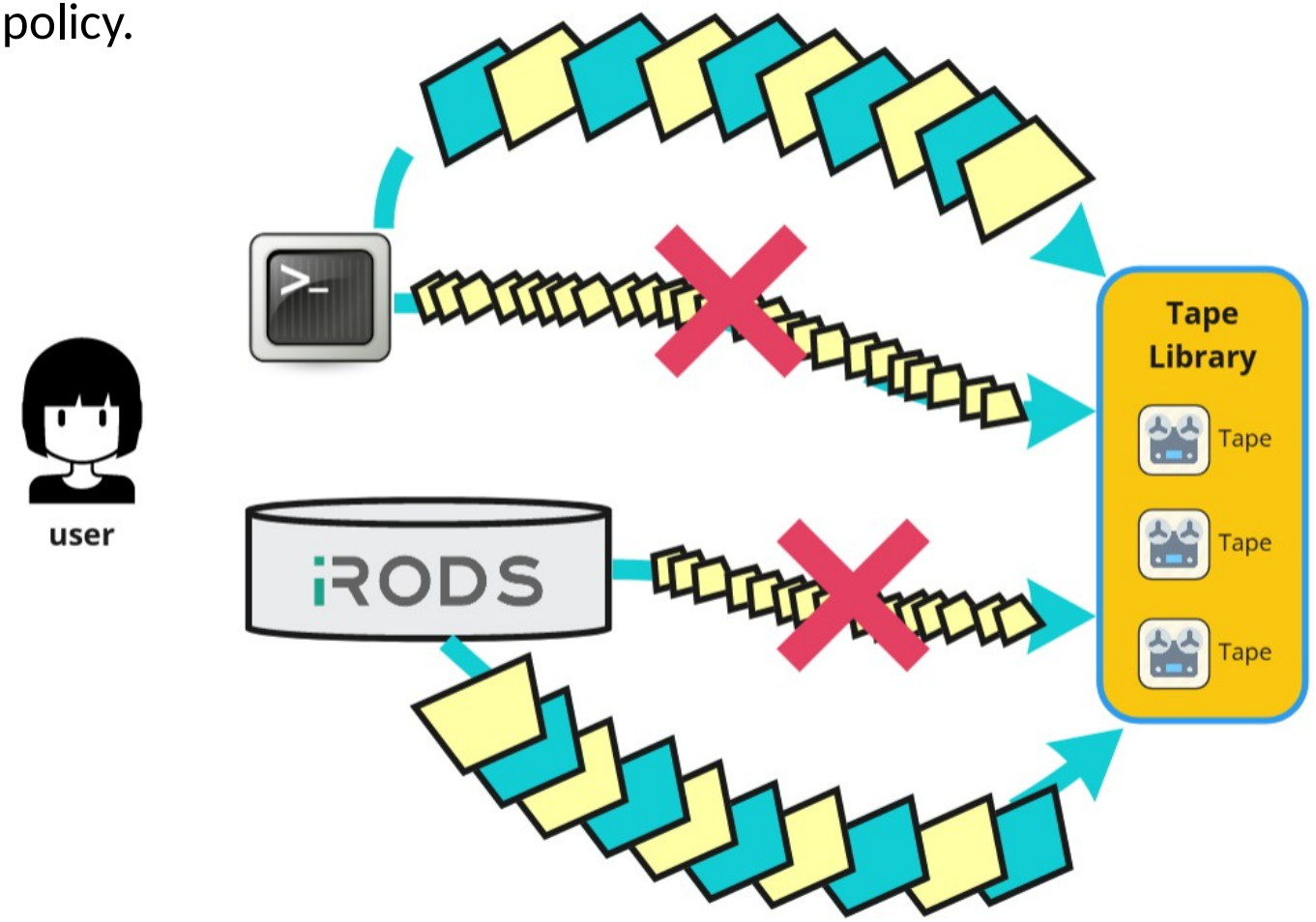
TAPE LIBRARY AND ARCHIVING DATA FLOW

- The SURF Data Archive service offers storage space on a Tape Library
- The Service is accessible directly via command line clients, but sometimes it is configured as iRODS resource and accessed only through iRODS.



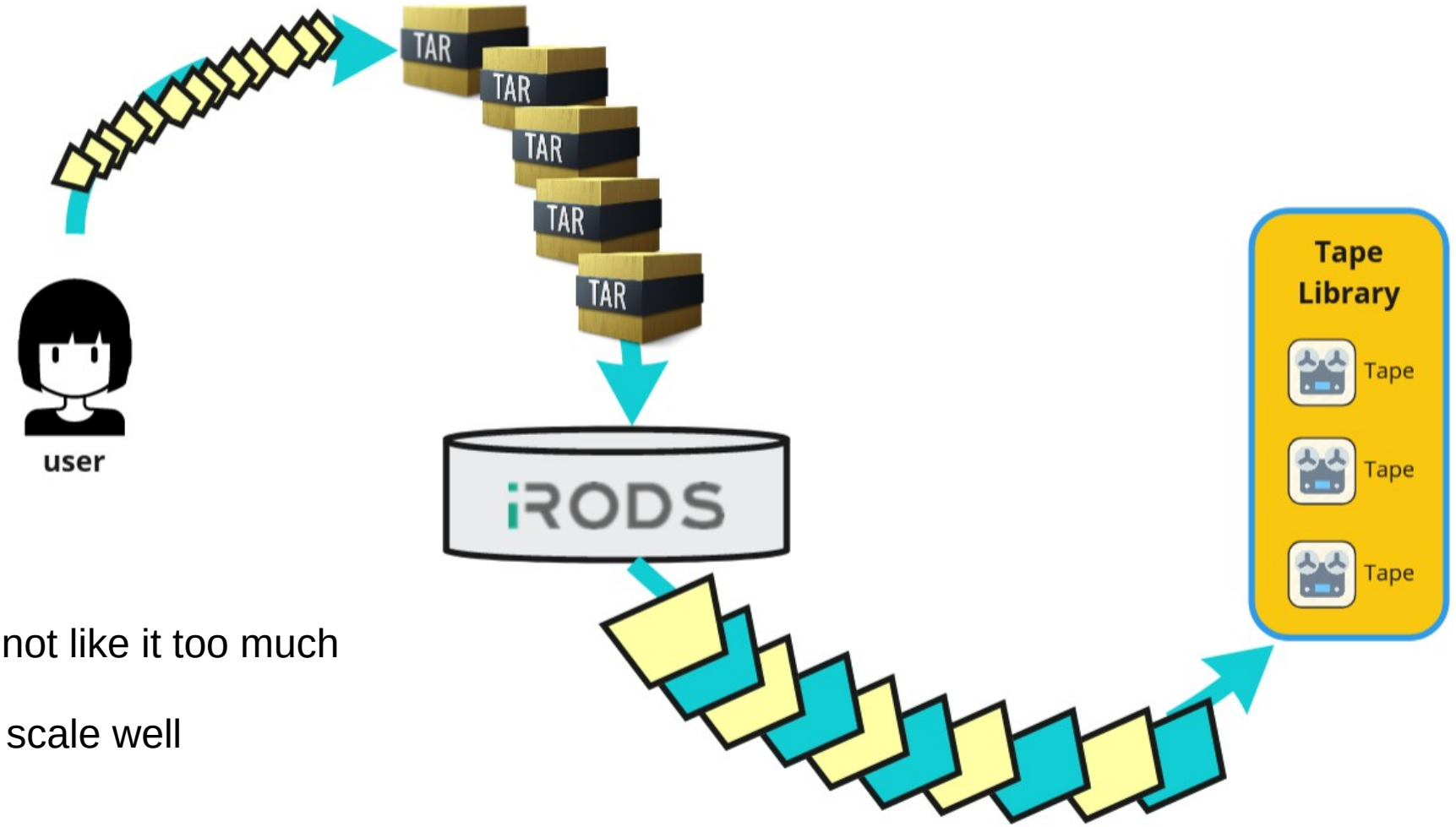
THE ISSUE WITH SMALL FILES

- Tape library performance are optimized for files of the order of magnitude of 1 GB or more and that constraint is enforced through a policy.
- The users must comply with that policy.



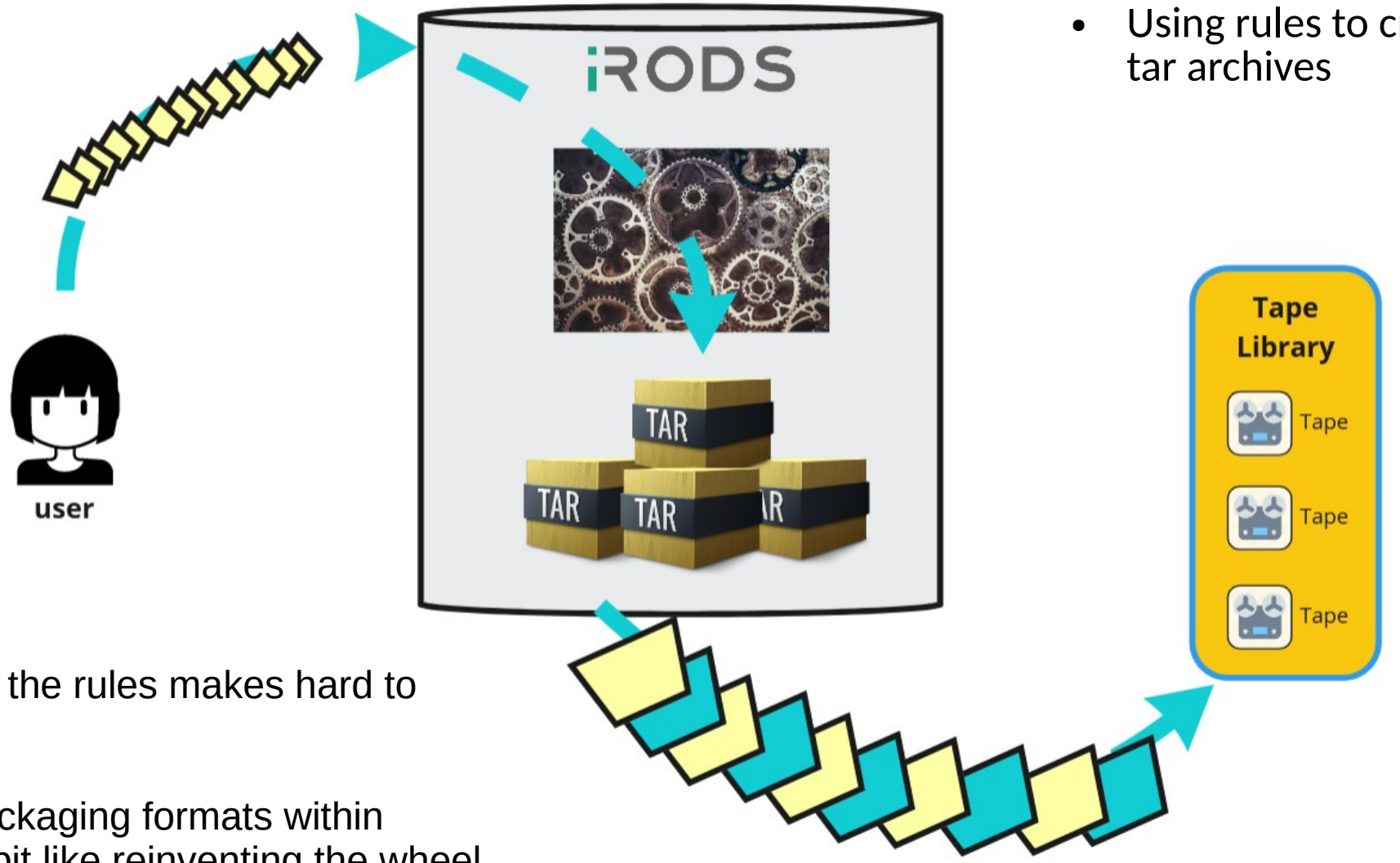
CLASSIC APPROACH 1: DELEGATING TO THE USER

- Using of iRODS command ibun



- User does not like it too much
- It does not scale well

CLASSIC APPROACH 2: AUTOMATING THROUGH IRODS RULES

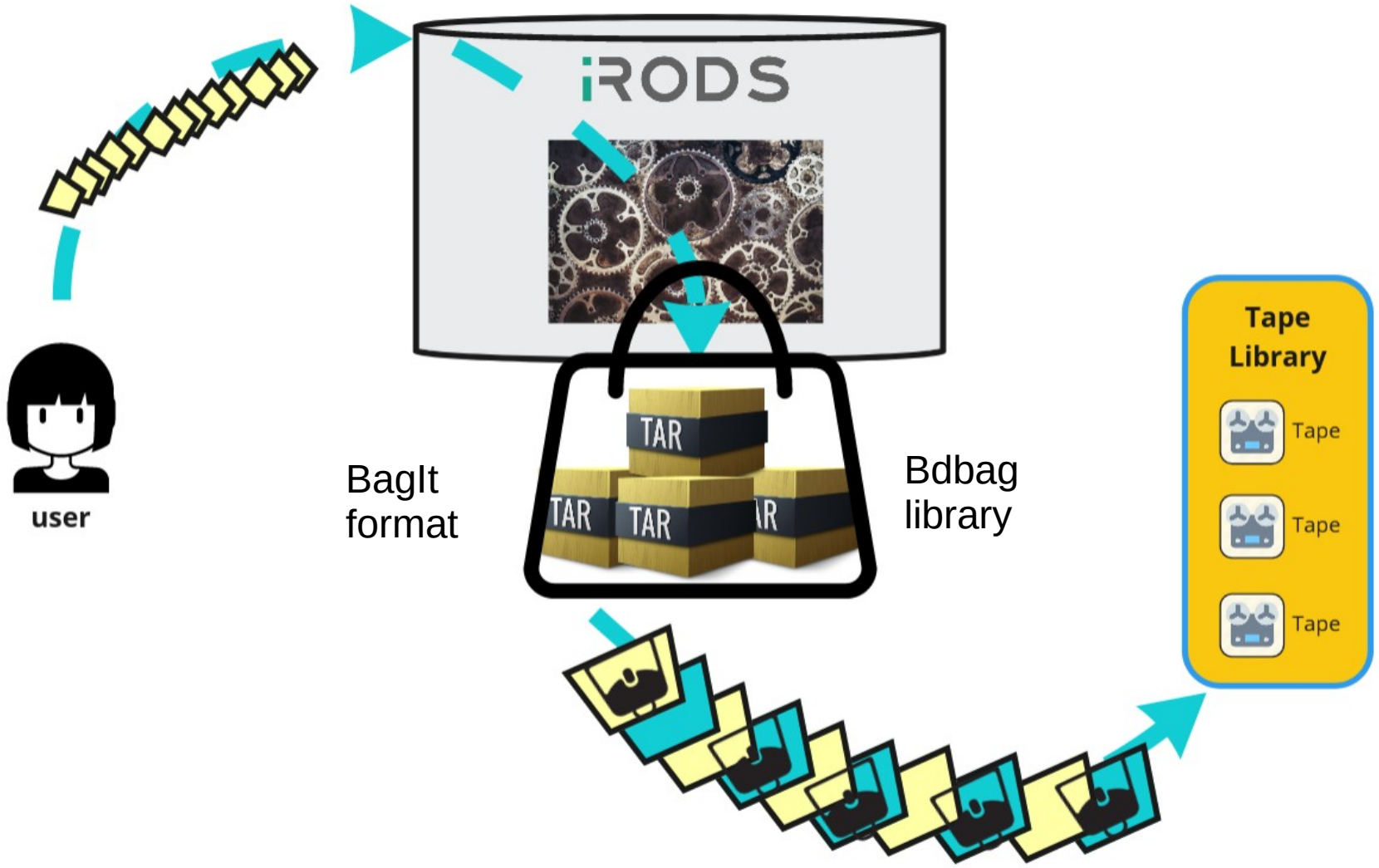


- Using rules to create tar archives

- The complexity of the rules makes hard to maintain them
- Coding certain packaging formats within iRODS rules is a bit like reinventing the wheel

ALTERNATIVE: PARTIAL "OUTSOURCING"

- BagIt format
- Bdbag library



THE BAGIT FORMAT AND THE BDBAG LIBRARY

- The **bdbag** utilities are a collection of software programs for working with BagIt packages that conform to the BDBag and Bagit/RO profiles.
  - https://github.com/fair-research/bdbag/

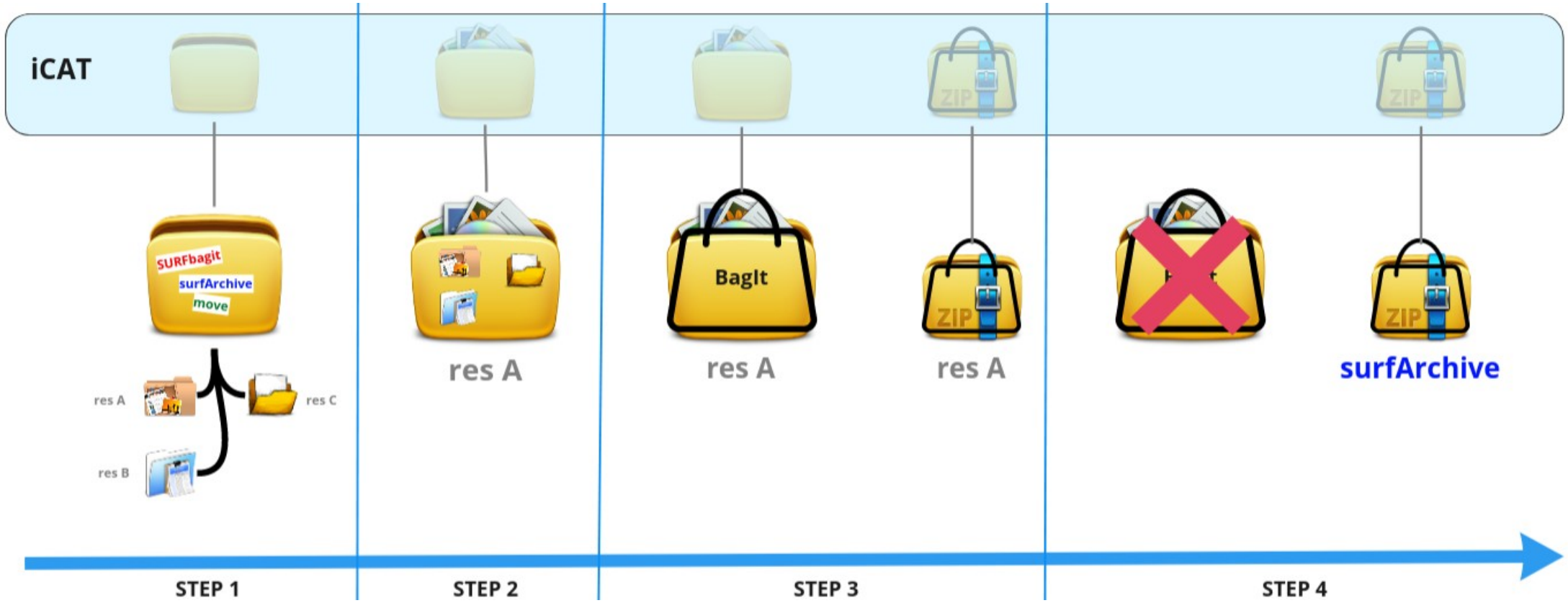
- **BagIt** is a set of hierarchical file layout conventions for storage and transfer of arbitrary digital content. A "bag" has just enough structure to enclose descriptive metadata "tags" and a file "payload" but does not require knowledge of the payload's internal semantics.
  - https://datatracker.ietf.org/doc/rfc8493/

SURF

HOW IT WORKS: IRODS RULES

- We have added to iRODS a rule set to “prepare” the data to be packed or unpacked. A default resource is assumed to be defined as input.
  - **SURFgroupDataByResource**: the data are splitted in two groups, those stored on the designated default resource and all the others.
  - **SURFphymoveDataPerCollection**: loops over all the objects in other resources, if the object is already stored in the default resource, trim the other copy, if not, move it to the default resource
  - **SURFbagitAlignDataResources**: applies SURFgroupDataByResource and SURFphymoveDataPerCollection recursively
  - **SURFbagit**: wrapper around the bdbag library, used to create the BagIt package
  - **SURFbagitBatch**: it applies the SURFbagit rule only to collections tagged with specific metadata
  - **SURFunbagit**: wrapper around the bdbag library, used to “materialize” the bagit package into iRODS as a normal collection
  - **SURFunbagitBatch**: it applies the SURFunbagit rule only to collections tagged with specific metadata

HOW IT WORKS: IRODS RULES



Tag the collection

Move all the data on the same resource

BagIt! and register the BagIt package

Move the BagIt package (and remove the original collection)

HOW IT WORKS: KEYWORDS

- To avoid that a user can trigger directly the creation of packages and their transfer to the tape library, we rely on metadata: the user can add certain metadata to the collections and a rule (**SURFbagitBatch**) executed periodically will find those collections and trigger the actual archiving process.
- For example:
  - $ imeta set -C mycollection **SURFbagit surfArchive move**

    will make the batch process to move the collection “mycollection” to the resource surfArchive and packing it.
  - $ imeta set -d mylibrary.zip **SURFunbagit demoResc copy**

    will make the batch process to copy the BagIt package “mylibrary.zip” to the target resource “demoResc” and materrialize it.

HOW IT WORKS: IRODS METADATA

- **The metadata are exported recursively for all the sub-collections and objects present under the main collection**.
  - By default the avus are exported, while acls must be selected explicitly. However acls are not imported.
  - The rule **SURFunbagitBatch** checks automatically if the exported metadata are available within the bagit package and, if it is the case, then they are imported back to iRODS.

How it works: BDBag library

- The library takes care of:
  - Creating the BagIt package
  - Computing the checksums
  - (Un)Compressing the package
  - Add the exported iRODS metadata as a json file to the BagIt payload
  - Materializing the BagIt package

- Caveats:
  - The library works on a plain file system, so that must always be the starting resource, in case of packing or the target resource in case of unpacking operations

BENEFITS

- Distributing the operations between iRODS and an external library allows to focus the logic of the rules on the management of the iRODS resources, while the library deal with the specific details of packaging formats and related options, like compression algorithms and metadata.
  - Less maintenance work within iRODS
  - Re-use of an open source library for a well defined format without the need to reimplement the same functions
  - Easier to extend the concept with additional libraries and formats re-using the same rules
  - Support for importing/exporting iRODS AVUs and ACLs

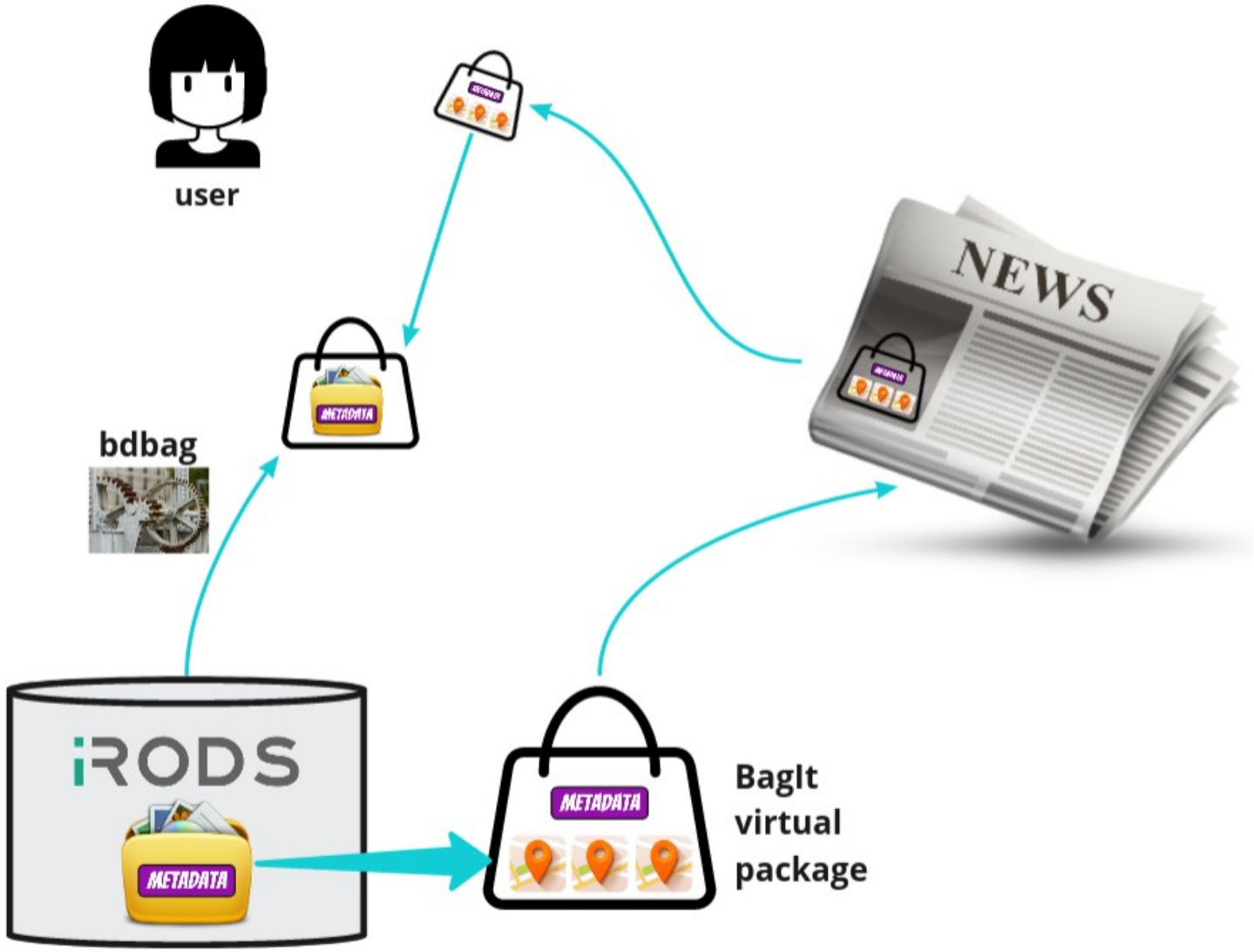
POTENTIAL

- The bdbag library supports the creation of “virtual” BagIt packages: the payload of the package is just a list of pointers to the real data.
- It support “transport handlers” to fetch remote data. Transport handlers for http(s), ftp, s3, gs and globus are provided, along with an extensibility mechanism for adding externally developed transports.

  These features could be used in two interesting ways.

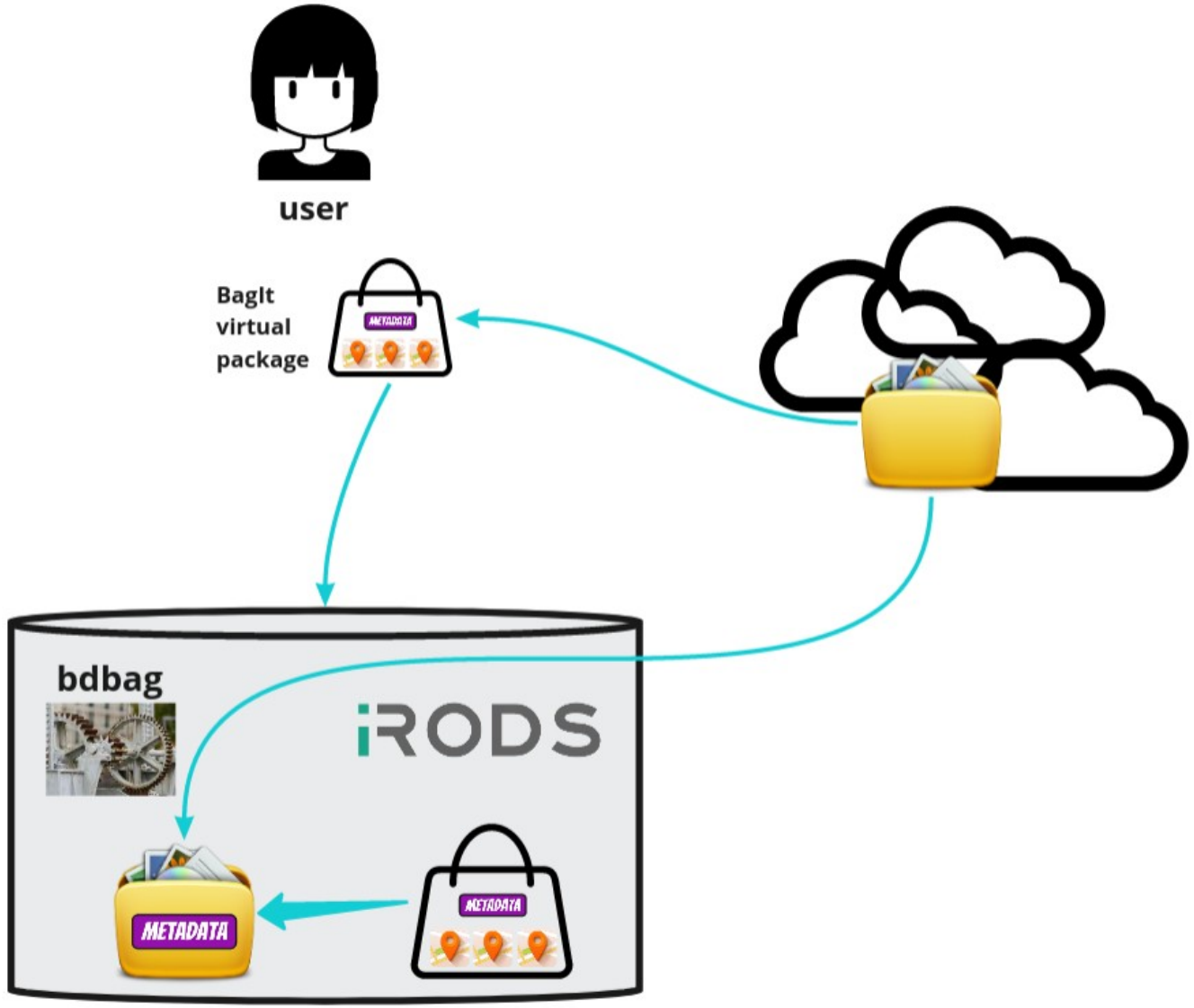
POTENTIAL: EXPORT

1) In case of big data sets, it could be possible to publish the virtual package without moving the data. It would be much faster. When the user download the virtual package, then she can download the real data directly from iRODS.



POTENTIAL: IMPORT

2) In case of need to transfer data from another data repository or cloud storage space to iRODS, it would be possible to move just the virtual package, delegating to iRODS the real data transfer. In this way we would save one jump in the data flow and allow a server-to-server transfer.



Thanks for your attention

Questions?