European
Commission

JD+ and R: **rjdverse**

ESTP Training

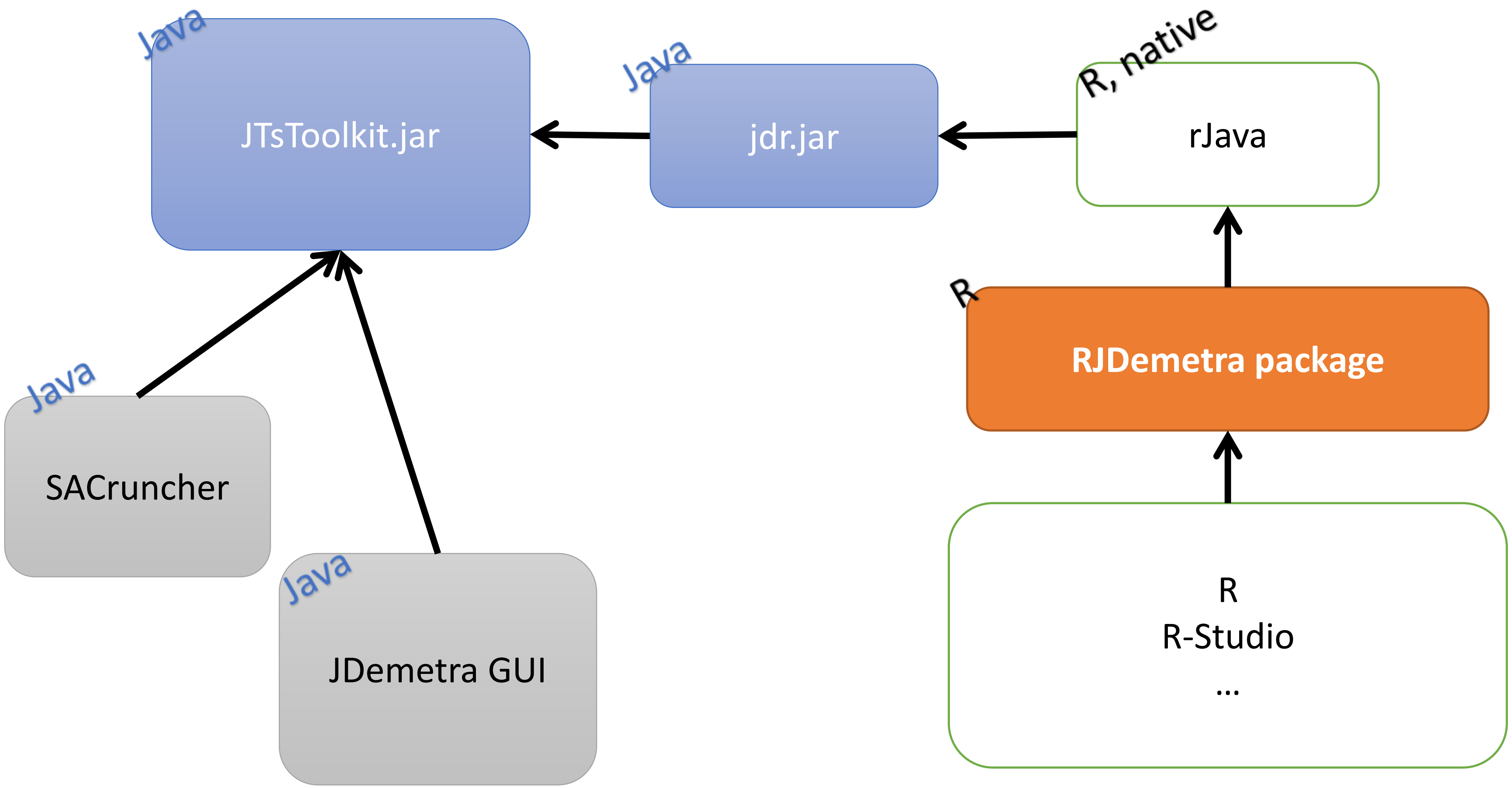
15-17/10/2024

1. Main requirements

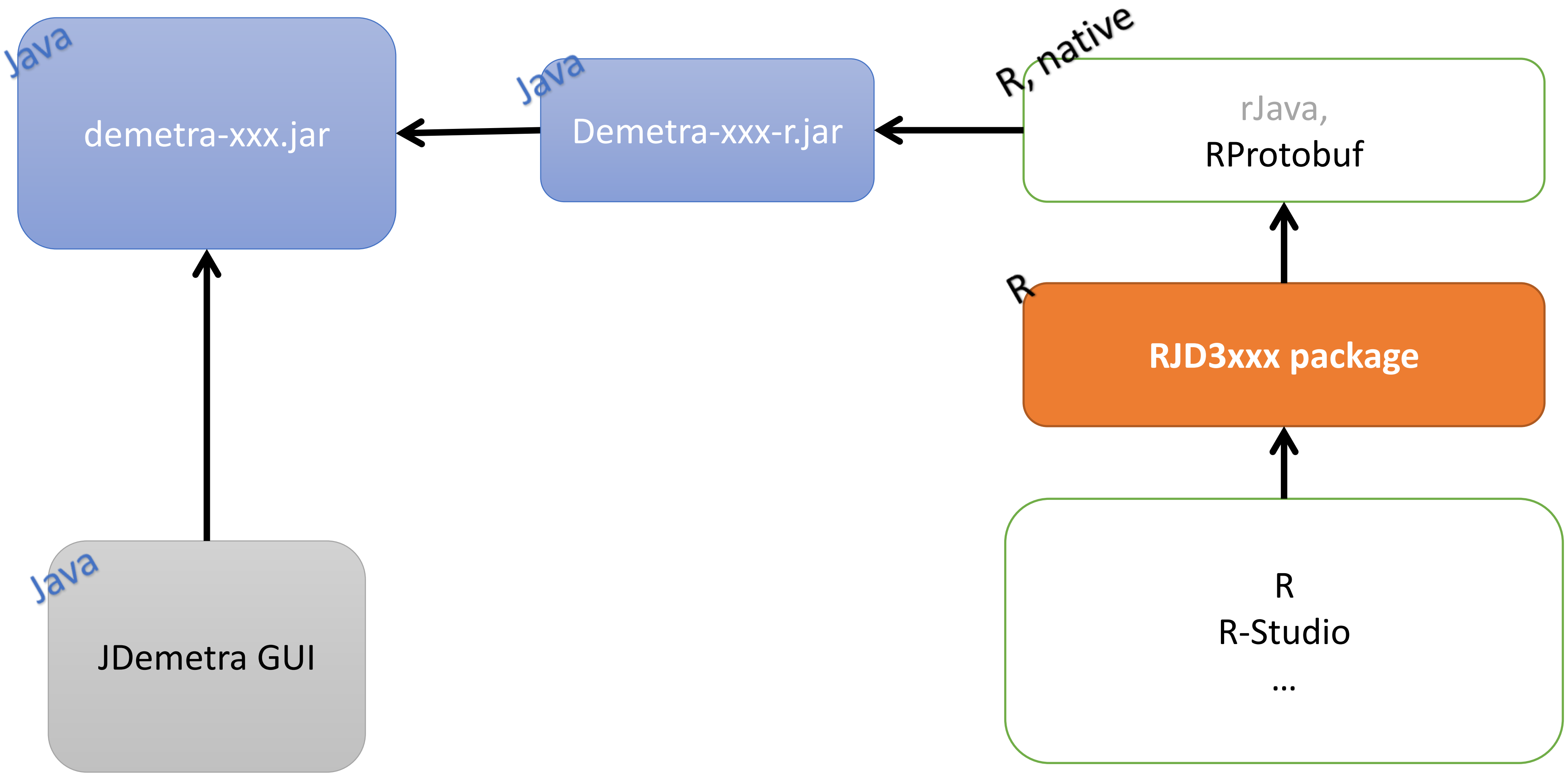
- RJDemetra
  - **Java runtime (>=11)**
  - R (>= 3.1.1)
  - rJava (>= 0.9-8)

- rjdverse
  - **Java runtime (>=17.0)**
  - R (>= 3.6.0)
  - rJava (>= 1.0-6),
  - RProtoBuf (>= 0.4.17)

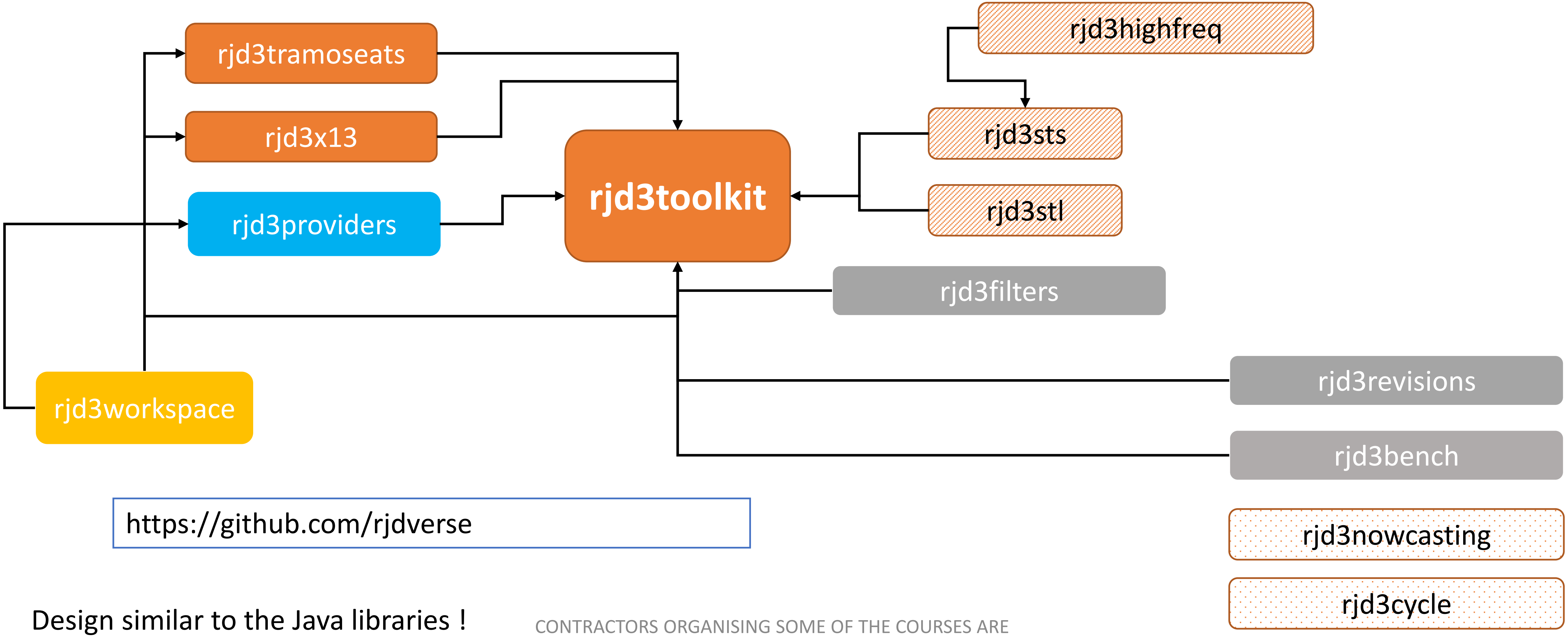
2. Technical design of Rjdemetra (v2)



3. Technical design of rjdverse (v3)

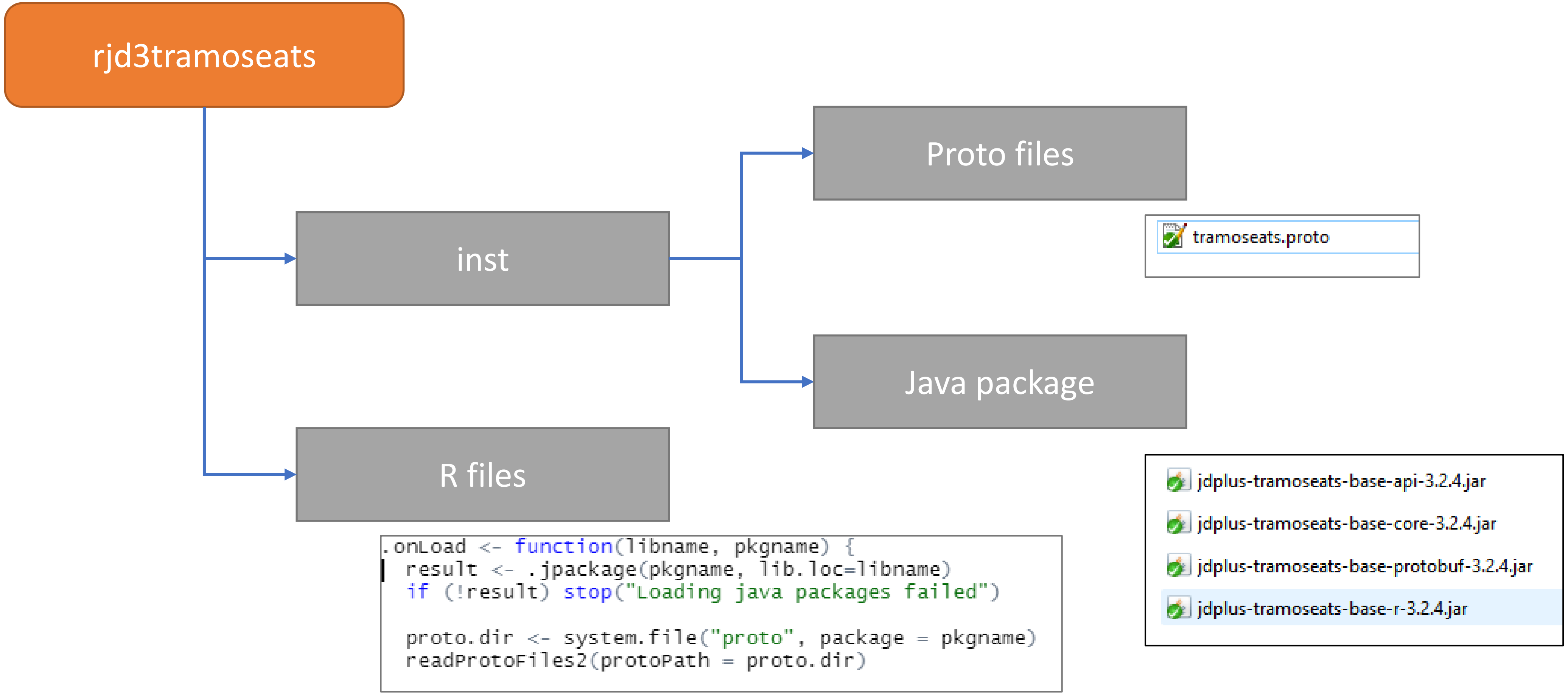


4. rjdverse: Overview



https://github.com/rjdverse

Design similar to the Java libraries !



tramoseats.proto

```
.onLoad <- function(libname, pkgname) {
  result <- .jpackage(pkgname, lib.loc=libname)
  if (!result) stop("Loading java packages failed")

  proto.dir <- system.file("proto", package = pkgname)
  readProtoFiles2(protoPath = proto.dir)
```

jdplus-tramoseats-base-api-3.2.4.jar

jdplus-tramoseats-base-core-3.2.4.jar

jdplus-tramoseats-base-protobuf-3.2.4.jar

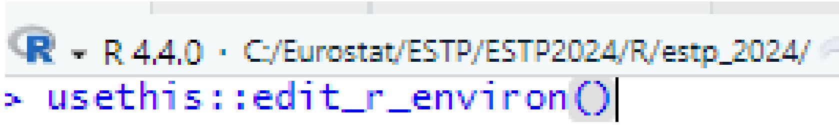
jdplus-tramoseats-base-r-3.2.4.jar

5. Objectives of R packages

- High-level functions with most common results
- Many low-level functions
  - Advanced users
  - Research
  - Training
  - Additional tools

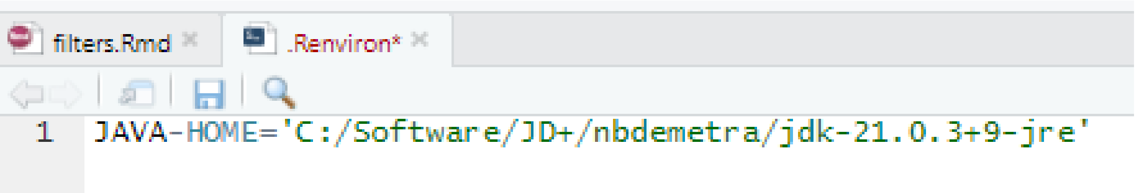
6. Installing the packages

- If need be, referencing the correct java runtime (>= 17.0)

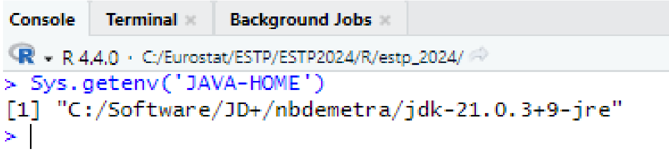


```
> usethis::edit_r_environ()
```

  - Set your JAVA_HOME variable (tip: use the jre provided with JD+)



```
JAVA-HOME='C:/Software/JD+/nbdemetra/jdk-21.0.3+9-jre'
```



```
> Sys.getenv('JAVA-HOME')
[1] "C:/Software/JD+/nbdemetra/jdk-21.0.3+9-jre"
> 
```

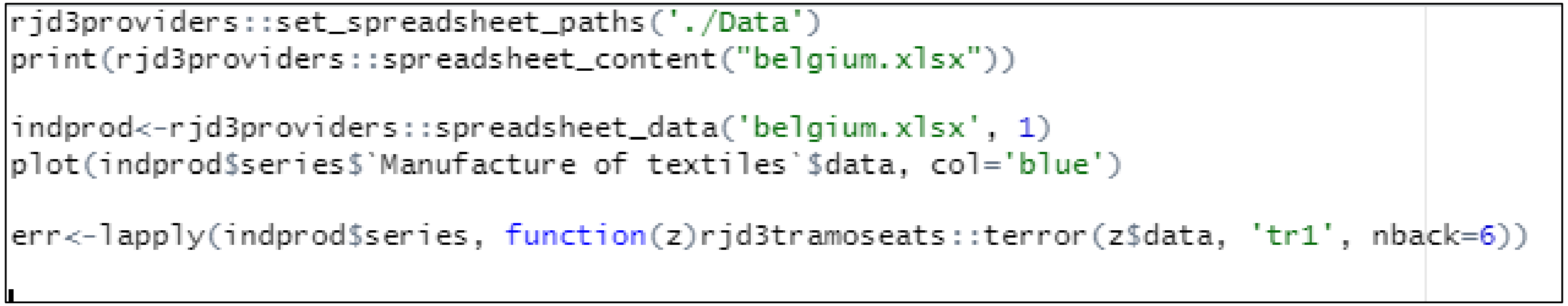
- Install the various packages (internet access needed)
  - # install.packages("remotes")
  - remotes::install_github("rjdverse/rjd3workspace@*release")

6. Installing the packages (cont)

- `remotes::install_github("rjdverse/rjd3filters@*release")`
- `remotes::install_github("rjdverse/rjd3sts@*release")`
- `remotes::install_github("rjdverse/rjd3highfreq@*release")`
- `remotes::install_github("rjdverse/rjd3x11plus@*release")`

7. Examples

- Reading Excel files (JD+-like) and detecting errors

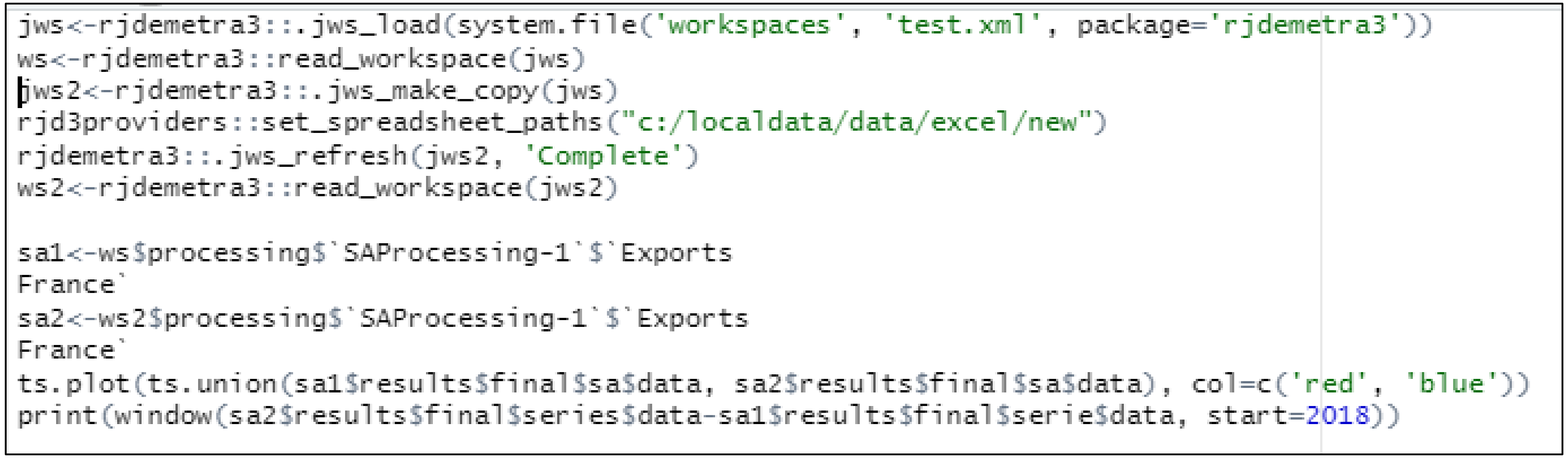


```
rjd3providers::set_spreadsheet_paths('./Data')
print(rjd3providers::spreadsheet_content("belgium.xlsx"))

indprod<-rjd3providers::spreadsheet_data('belgium.xlsx', 1)
plot(indprod$series$`Manufacture of textiles`$data, col='blue')

err<-lapply(indprod$series, function(z)rjd3tramoseats::terror(z$data, 'tr1', nback=6))
```

- Refreshing a workspace



```
jws<-rjdemetra3::.jws_load(system.file('workspaces', 'test.xml', package='rjdemetra3'))
ws<-rjdemetra3::read_workspace(jws)
jws2<-rjdemetra3::.jws_make_copy(jws)
rjd3providers::set_spreadsheet_paths("c:/localdata/data/excel/new")
rjdemetra3::.jws_refresh(jws2, 'Complete')
ws2<-rjdemetra3::read_workspace(jws2)

sa1<-ws$processing$`SAProcessing-1`$`Exports
France`
sa2<-ws2$processing$`SAProcessing-1`$`Exports
France`
ts.plot(ts.union(sa1$results$final$sa$data, sa2$results$final$sa$data), col=c('red', 'blue'))
print(window(sa2$results$final$series$data-sa1$results$final$serie$data, start=2018))
```

8. Final remarks

- Most features provided in the Java libraries can be called from R

- Most tasks can be automated
- Many additional tools could be developed in R