JD+ and R

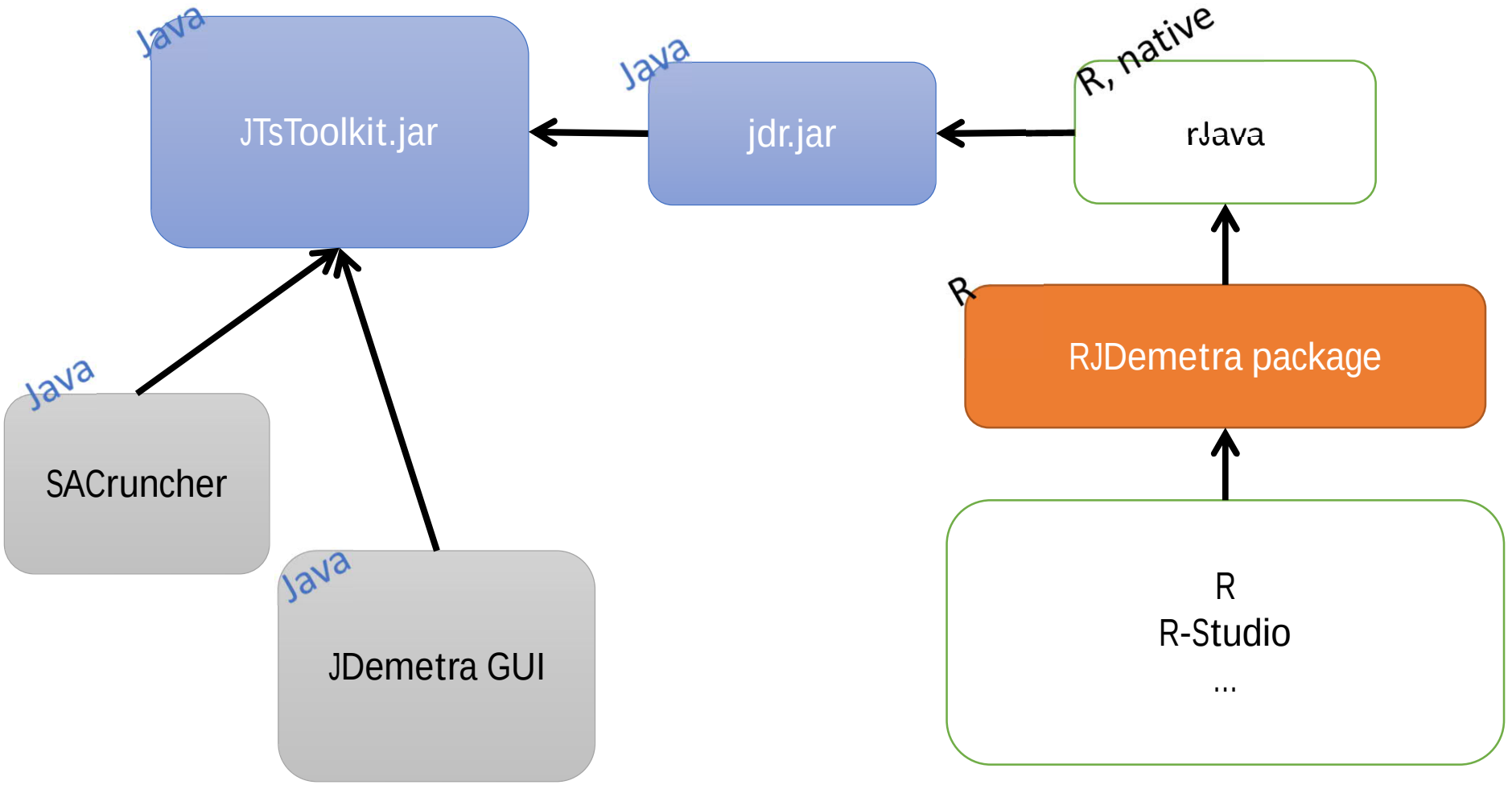
ESTP Training

Main requirements

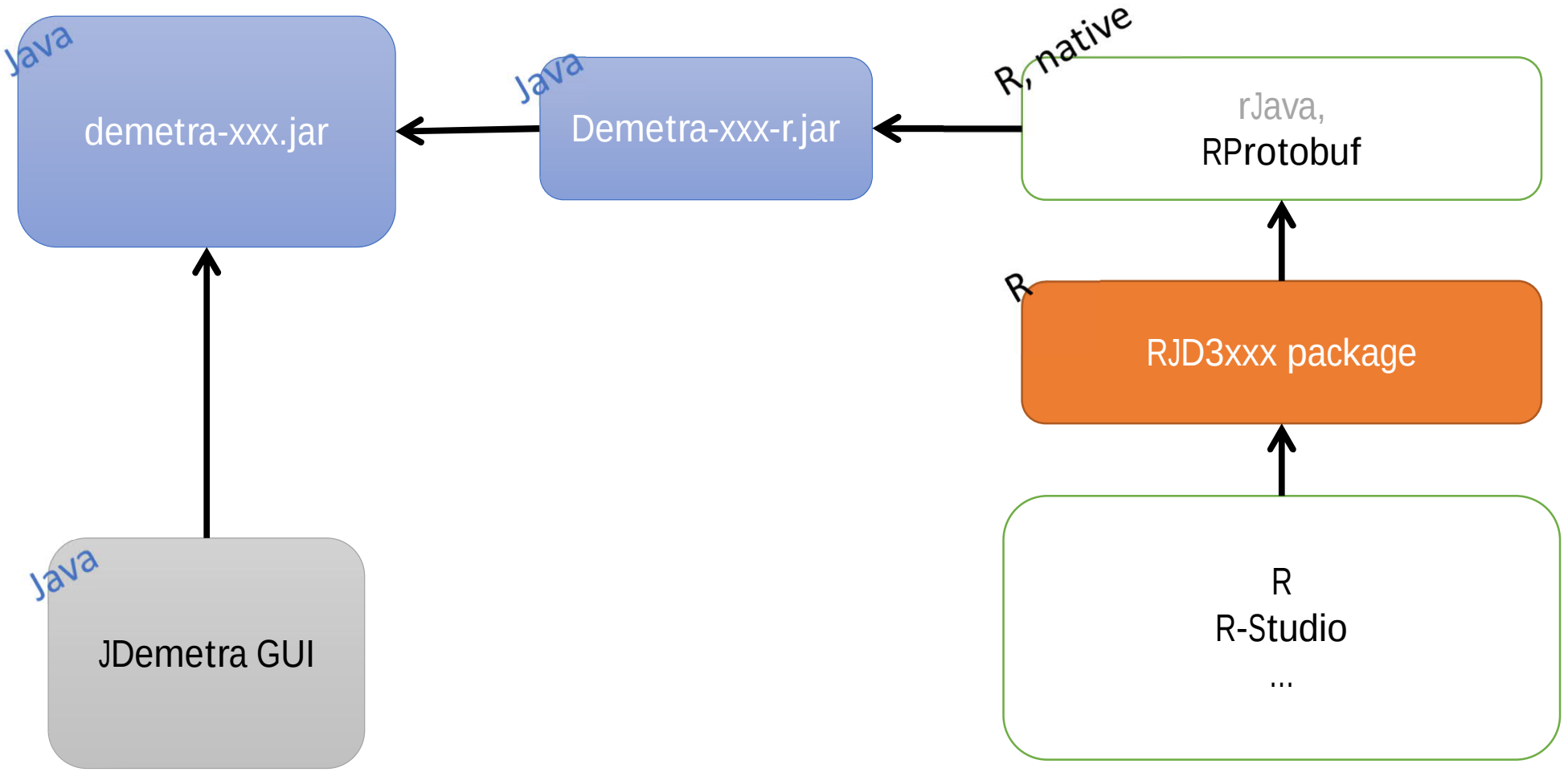
- RJDemetra
  - Java runtime (>=11)
  - R (>= 3.1.1)
  - rJava (>= 0.9-8)

- RJDemetra3
  - Java runtime (>=17.0)
  - R (>= 3.6.0)
  - rJava (>= 1.0-6),
  - RProtoBuf (>= 0.4.17)

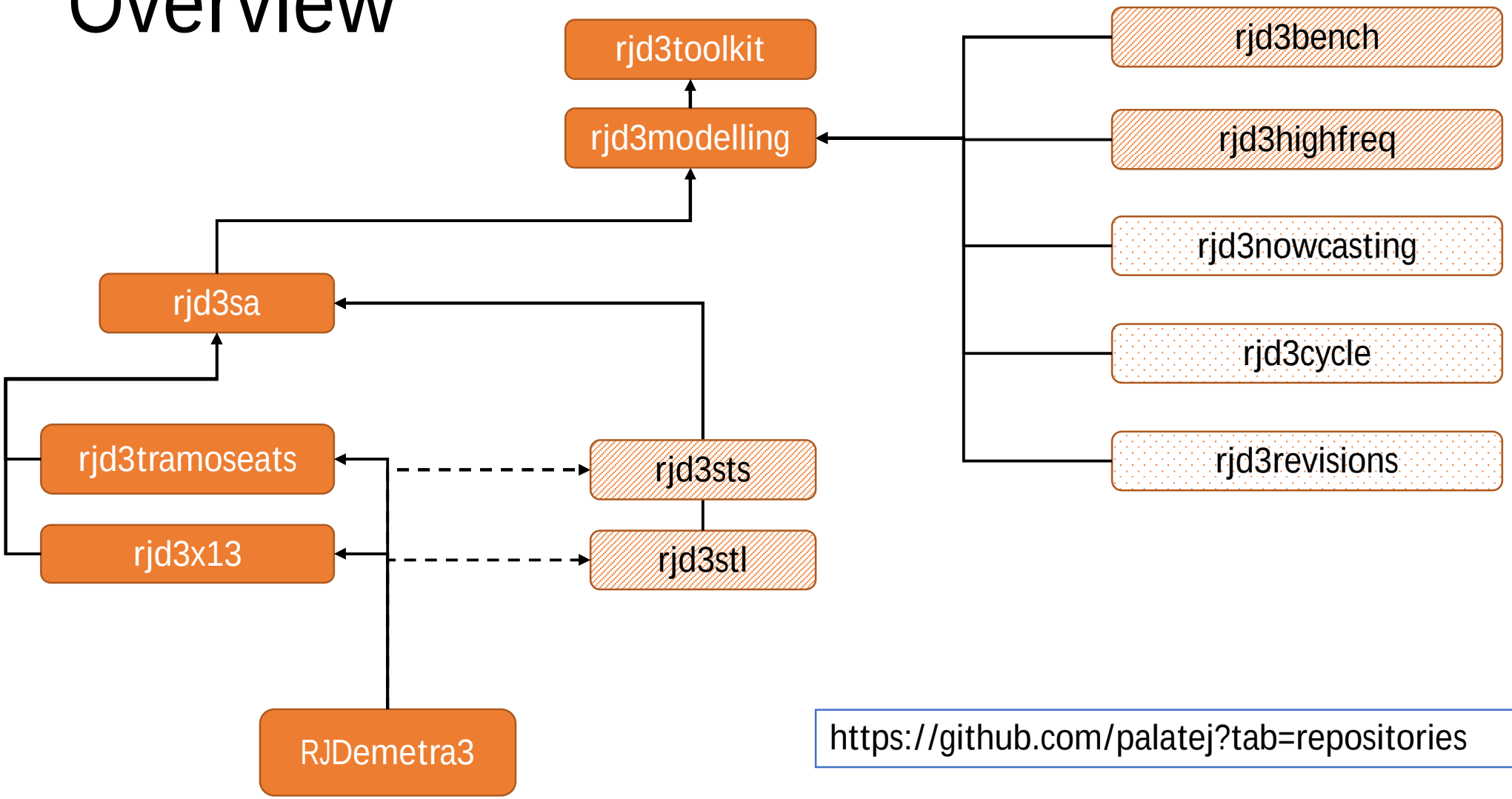
Technical design (Rjdemetra)



Technical design (Rjdemetra3)

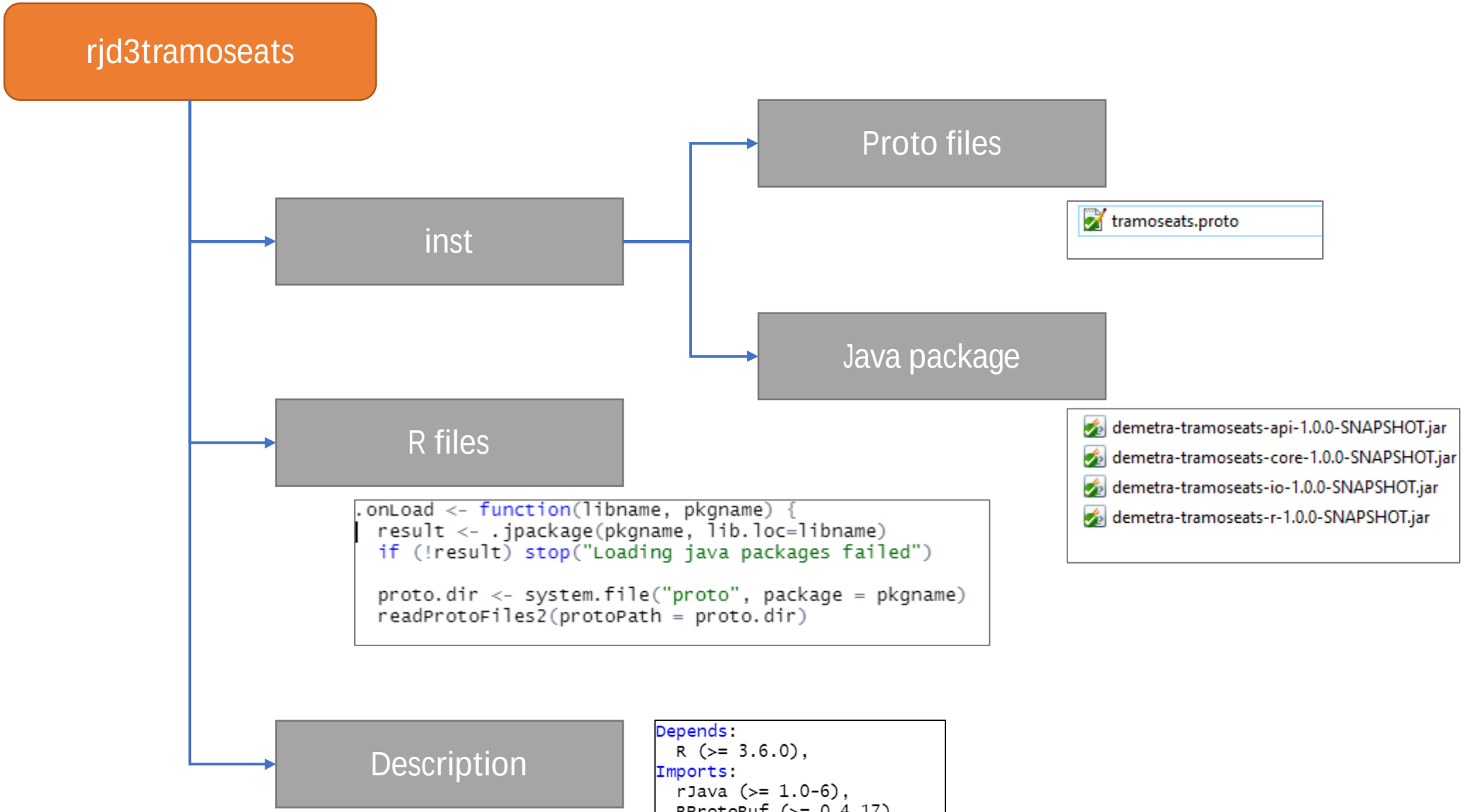


Overview



https://github.com/palatej?tab=repositories

Design similar to the Java libraries !



Objectives of R packages

- High-level functions with most common results
- Low-level functions
  - Advanced users
  - Research
  - Training

TramoSeats / X13 packages

- Specifications
  - Default specifications
  - Modifications of the different options (with auxiliary functions)

```
suppressPackageStartupMessages(library(RJDemetra3))
library(rjd3modelling)

tr4<-spec_tramo_default("tr4")
tr4$estimate$tol<-1e-12

out1<-rjd3modelling::createOutlier("AO", "2000-12-31", name="out1")
out2<-rjd3modelling::createOutlier("LS", "2008-08-01", name="out2")

tr4$regression$outliers<-list(out1, out2)

q<-fast.tramo(retail$RetailSalesTotal, tr4)
```

TramoSeats / X13 packages (cont)

- Processing
  - Fast processing (fast.xxx method)
  - Full processing (xxx.method)
    - Results
    - Estimation spec
    - Result spec
  - Refreshing of spec

```
s<-window(retail$RetailSalesTotal, end=2009)

qfast<-fast.tramo(s, tr4)
qfull<-tramo(s, tr4)

# define a "Frozen" domain
tr4new<-tramo.refresh(qfull$result_spec, policy="Outliers", end="2009-12-31")
qfastnew<-fast.tramo(retail$RetailSalesTotal, tr4new)

# Re-estimate all outliers
tr4new2<-tramo.refresh(qfull$result_spec, policy="Outliers")
qfastnew2<-fast.tramo(retail$RetailSalesTotal, tr4new)
```

TramoSeats / X13 packages (cont)

- Generic functions

```
sa<-fast.tramoseats(retail$BookStores)
decomp<-sa.decomposition(sa)

ts.plot(ts.union(decomp$series, decomp$sa, decomp$t), col=c("gray", "blue", "red"))

sax<-fast.x13(retail$BookStores)
decompx<-sa.decomposition(sax)

ts.plot(ts.union(decompx$series, decompx$sa, decompx$t), col=c("gray", "blue", "red"))
```

- TODO
  - Preprocessing
  - Diagnostics
  - ...

Details. Seasonality tests

- Kruskal-Wallis (on stationary series)
- Friedman (...)
- Qs (...)
- Ftest (seasonal dummies)

```
s<-Imports$Latvia

plot(s)
st<-do.stationary(log(s), 12)

qs<-seasonality.qs(st$ddata, 12)
print(qs)

f<-seasonality.f(log(s), 12, model = "AR")
print(f)

kw<-seasonality.kruskalwallis(st$ddata, 12)
print(kw)

fr<-seasonality.friedman(st$ddata, 12)
print(fr)
```

```
w<-lapply(Imports, function(z){st<-do.stationary(z, 12);
return (seasonality.kruskalwallis(st$ddata, period=12)$pvalue)})
print(w[w>0.05])
```

```
$Greece
[1] 0.07726277

$Estonia
[1] 0.07494337

$Latvia
[1] 0.379018

$Lithuania
[1] 0.6527985
```

- Combined tests (on S+I)

```
z<-tramoseats(log(s))
cs<-seasonality.combined(z$result$decomposition$stochastics$i$data, period=12, mul = F)
```