



ET04-Redefining FDA Submissions: The Future of Interactive R Shiny Dashboards

André Veríssimo, Ph.D.

Vedha Viyash



André Veríssimo

Senior R/Shiny Engineer



/in/afsverissimo



averissimo



Vedha Viyash

Senior R/Shiny Engineer



/in/vedhav



vedhav

Meet the Team at R Consortium Submission WG (Pilot-4)



André Veríssimo

Vedha Viyash

Tymoteusz Makowski

Ismael Rodriguez

Pedro Silva

Damian Rodziewicz



Eric Nantz (Eli Lilly)

Sam Parmar (Pfizer)

Robert Devine (J&J)

Nate Mockler (Biogen)

Andrew Borgman (Bristol Myers Squibb)

Ben Straub (GSK)



Paul Schutte

Hye Soo Cho

...and so many more if we consider all the pilots!



Outline of Presentation

 **consortium**
**Submissions
Working Group**

**Pilot 4
Problem Statement**



podman



Container bundle



webR bundle

Explore alternative submission formats



Submissions Working Group

Cross industry pharma working group focusing on **improving practices of R-based clinical trial regulatory submissions**

This talk is about



Pilot 1

R package with
Tables, Listings
and Graphs
(TLGs)

Pilot 2

Teal-based Shiny
App bundled with
Golem

Pilot 3

Generation of
ADaM datasets
from STDM &
TLGs

Pilot 4

Self-contained
bundle with Teal-
based Shiny App

Pilot 5

Expand on Pilot
1&3 and use R to
generate ADaM
datasets

Installation instructions

The present



A Appendix 1: Pilot 2 Shiny Application Installation and Usage

To install and run the Shiny application, follow all of the procedures below. Ensure that you note the location of where you downloaded the Pilot 2 cTSD submission file. For demonstration purposes, the procedure below assumes the reader has been sent to the location `4\pilot2`.

In addition, create a new directory to hold the requested Pilot 2 Shiny application file for demonstration purposes; the procedure below assumes the new directory is the location `4\pilot2\Shiny`.

A.1 Installation of R and RStudio

Download and install R 4.1.2 for Windows from <https://www.r-project.org/doc/windows/InstallR412.pdf>. While optional, it is also recommended to view the Shiny application within the RStudio IDE. You can download RStudio for Windows by visiting <https://www.rstudio.com/products/rstudio/download/#download>.

A.2 Installation of Rtools

Due to certain R packages requiring the use of C++ compilers, it is also suggested that you install the Rtools Windows utility. For demonstration purposes, the procedure below assumes the Rtools build for R version 4.0.4.

A.3 Installation of R Packages

A minimum set of R packages are required to create the Pilot 2 Shiny application file; see our README regarding and the custom package environment used for the application is explained currently. The first packages to install are the common and platform packages:

```
install.packages(c("remotes", "httpfs"))
```

A.4 Extract Application Bundle

Use the `plg2` package to search the Shiny application bundle `plg2.r` within the Pilot 2 cTSD submission folder. This file is located in the following relative path within the cTSD bundle directory:

```
shinydemo\resources\tools\plg2\bin\platform\package\plg2.r
```



Enter the following command in the R console to extract the Shiny application file to the destination directory:

```
plg2(Shinydemo, "shinydemo", "shinydemo\resources\tools\plg2\bin\platform\package\plg2.r", "shinydemo\resources\tools\plg2\bin\platform\package\plg2.r")
```

The console will display messages of extracting and writing files to the destination directory. Note that the procedure creates a subdirectory called `plg2\resources` in the destination directory. Take note of that particular directory path on your system, as you will use this in the remaining procedure. In this example, the directory is located in the following path:

```
4\pilot2\Shinydemo\resources
```

A.5 Initiate R Package Environment for Shiny Application

The dependencies for the Shiny application are managed by the `renv` R package management system. To initialize the requested R package library used for the Shiny application, launch a new R session in the directory where you requested the application source file in the previous step. Choose one of the following options depending on your R computing environment and preference.

Option 1: RStudio

Create a new RStudio Project within the `plg2\resources` directory:

- Select `File` → `New Project`
- In the `Create Project` dialog box, choose `Existing Directory`
- In the `Create Project from Existing Directory` dialog box, click the `Browse` button and navigate to the `plg2\resources` directory. Once the location has been selected, click the `Create Project` button.

RStudio will extract the source and automatically install the `renv` package into the project directory. To complete the process of installing the pilot 2 packages, run the following command in the R console:

```
renv::install(quiet = TRUE)
```



Option 2: R Console

Launch a new R session in the `plg2\resources` directory of the requested application directory. By default, the R GUI interface on Windows will launch a new R console in your default Windows home directory (typically the `Documents` folder). Perform the following steps to ensure R is located in the proper directory:

1. Note

The procedure below assumes R 4.1.2 has been installed in a default location. If not, use the full path to the R GUI executable on your system; you can find the location on your system by performing the following steps:

- Open the Windows Start menu and expand to show all applications.
- Navigate to the R entry and select `New` → `Open File Location`.
- Right-click the `R GUI 4.1.2 entry` and select `Open File Location`.
- A new folder window will open with the location `R GUI 4.1.2` highlighted. Right-click this entry and select `Properties`.
- In the `Properties` window, note the path specified by the `Target` text field. The portion of the text in quotation gives the full path to the R GUI on location on your system.

2. Note

The procedure below assumes R 4.1.2 has been installed in a default location. If not, use the full path to the R GUI executable on your system; you can find the location on your system by performing the following steps:

- Open the Windows Firewall program for searching for Windows Firewall in the Windows Start menu.
- Change the external directory to the `plg2\resources` directory by running the following command (substitute the `plg2\Shiny` location for your application directory as needed):

```
Set location: path %1\pilot2\Shinydemo\resources
```

3. Launch the Windows Firewall in this session by running the following command:

```
Set location: path %1\pilot2\Shinydemo\resources
```

The R GUI will launch and automatically install the `renv` package into the project directory. To complete the process of installing the pilot 2 packages, run the following command in the R console:

```
renv::install(quiet = TRUE)
```

A.6 Update Shiny Application Configuration

The Shiny application needs an configuration system in order to support the API data sets contained in the cTSD bundle. The data files are located in the following relative path within the cTSD bundle directory:

```
shinydemo\resources\tools\plg2\bin\platform\package\plg2.r
```

Run the following command in the R console (substitute the `plg2` location for your application directory as needed):

```
plg2\resources\tools\plg2\bin\platform\package\plg2.r
```

A.7 Launch Shiny Application

To run the Shiny application, enter the following command in the R console:

```
plg2() --no-args
```

1. Note

After closing the application, you may see a warning appear in the R console regarding some C++ errors. This can be safely ignored as this has no impact on the application output.

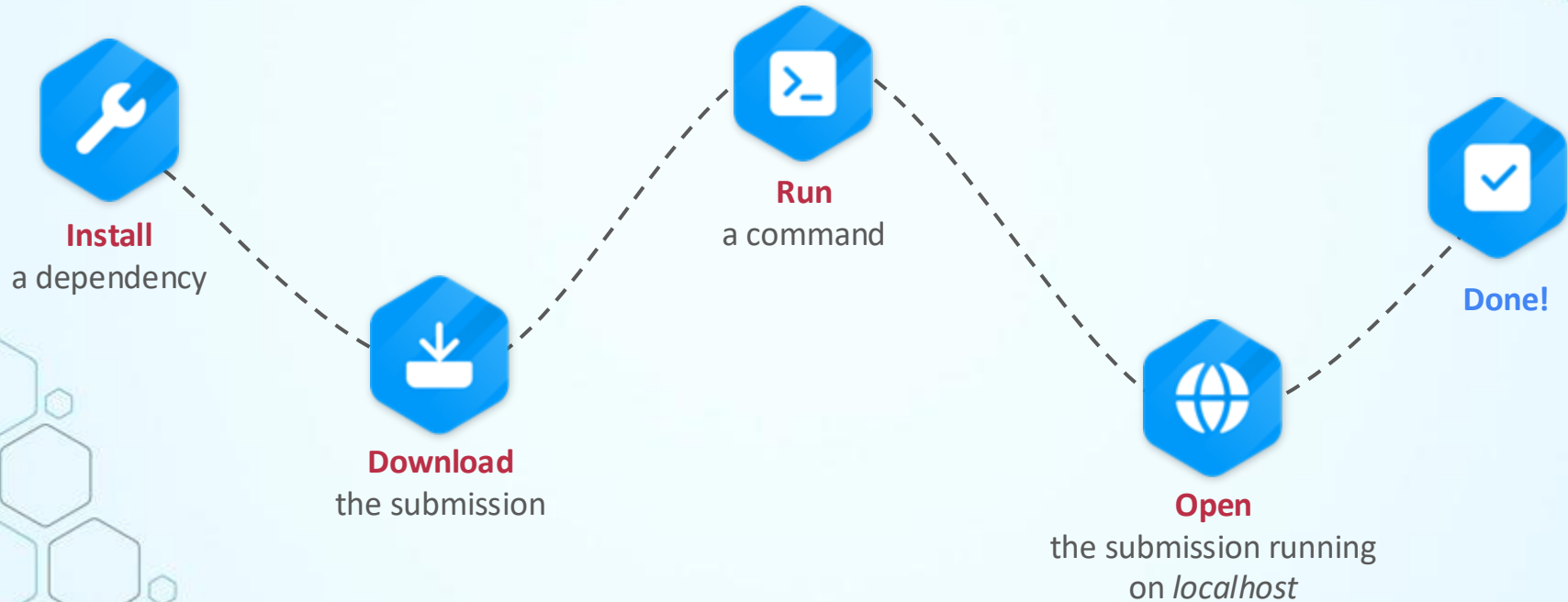


Explore alternative submission formats

Create **a new packaging format** for FDA submissions with **a simple installation** and **fewer dependencies**

Installation instructions

The future







An **open-source** alternative to Docker

- Runs R/Shiny **server** in a container
- Requires container installation
- Download of a big image
- Build once run anywhere

Interchangeable with Docker



webR *(powered by WebAssembly)*

An **in-browser** R runtime

- Runs R/Shiny **server** in a browser
- Requires a browser and an HTTP server
- Small footprint
- Build once run anywhere

WebAssembly is supported by 97.14% of browsers ([source](#)).



podman



docker

Container-based method to deploy pilot's Shiny App

Solution

- 🎯 **Configurable** Dockerfile
- 🎯 Documentation on creating the container
- 🎯 Automated build on `amd64` and `arm64` platforms

Configurable Dockerfile

```
1 ARG R_VERSION=4.2.0
2 ARG IMAGE_REGISTRY=docker.io
3 ARG IMAGE_ORG=rocker
4
5 FROM $IMAGE_REGISTRY/$IMAGE_ORG/r-ver:$R_VERSION
6
7 LABEL org.opencontainers.image.licenses="GPL-3.0-or-later" \
8       org.opencontainers.image.source="https://github.com/Appsilon/experimental-fda-submission-4-podman" \
9       org.opencontainers.image.vendor="Appsilon" \
10      org.opencontainers.image.authors="Andre Verissimo <andre.verissimo@appsilon.com>, Vedha Vijayash <vedha@appsilon.com>"
11
12 RUN apt-get update --quiet \
13     && apt-get install -y --quiet \
14     curl \
15     libssl-dev \
16     libcurl4-openssl-dev \
17     libxml2-dev \
18     libfontconfig1-dev \
19     libharfbuzz-dev libfribidi-dev \
20     libfreetype6-dev libpng-dev libtiff5-dev libjpeg-dev \
21     && apt-get autoremove -y --quiet \
22     && apt-get clean --quiet \
23     && rm -rf /var/lib/apt/lists/*
24
25 ENV RENV_PATHS_ROOT=/renv_cache
26
27 COPY ./renv_cache/ $RENV_PATHS_ROOT
28
29 ARG LOCAL_DIR=./submissions-pilot2
30 ARG APP_DIR=/usr/local/src/submissions-pilot2
31
32 COPY $LOCAL_DIR $APP_DIR
33
34 WORKDIR $APP_DIR
35
36 # Prevents RENV from mistakenly download from teal.* remotes (as the dependencies are
37 # already defined in renv.lock).
38 RUN Rscript \
39     --e "options(\"renv.config.install.remotes\" = FALSE)" \
40     --e "renv::restore()"
41
42 ARG R_SCRIPT=./entrypoint.R
43
44 COPY $R_SCRIPT $APP_DIR/entrypoint.R
45
46 CMD ["Rscript", "entrypoint.R"]
```

(build arguments for flexibility)

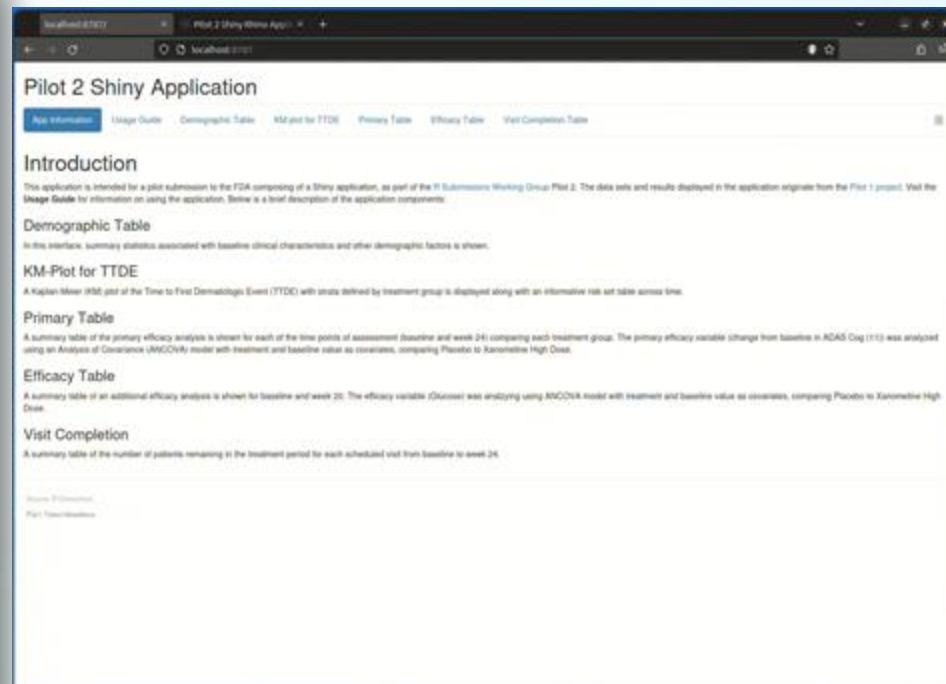
1. Install system requirements

(use host renv cache)

(configurable shiny directory)

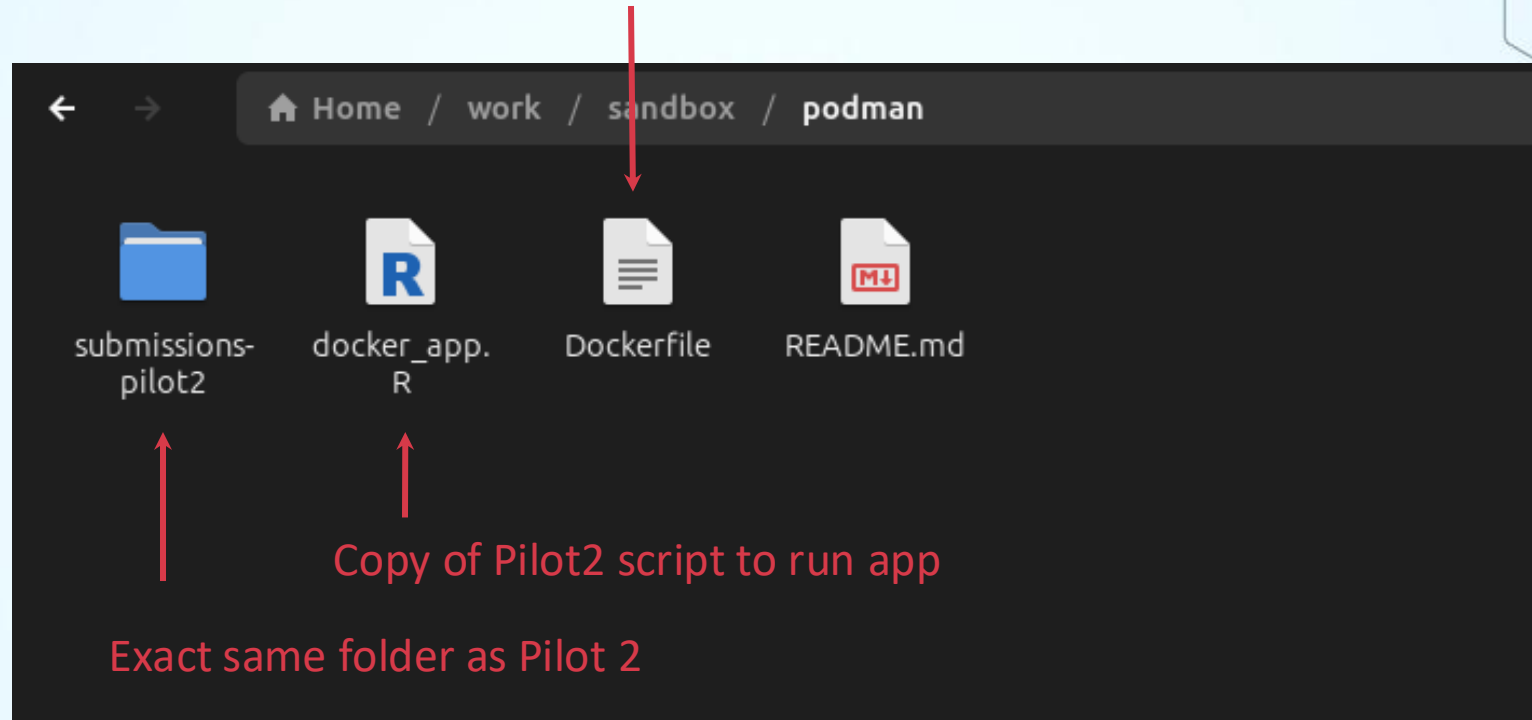
2. Install R packages

3. Init script



How does it look?

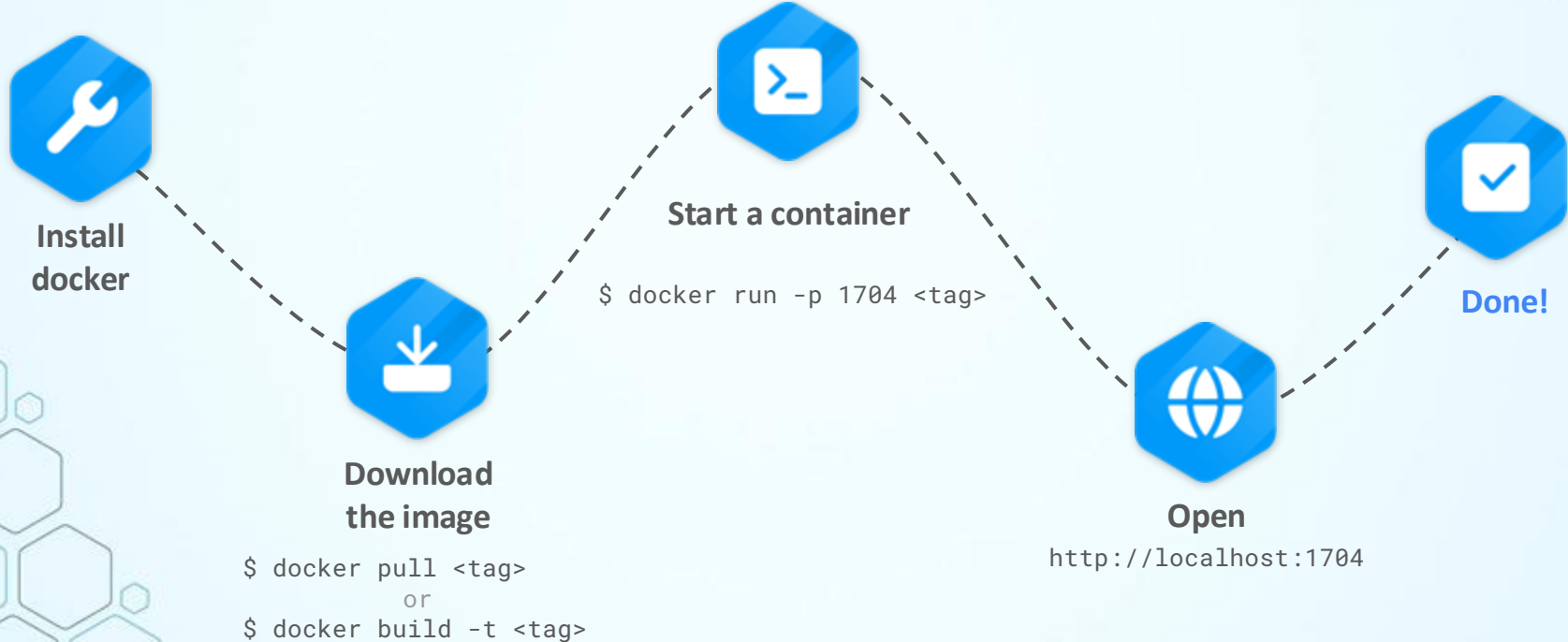
Dockerfile to build container image



<https://github.com/Appsilon/experimental-fda-submission-4-podman>

Installation instructions

The future with containers

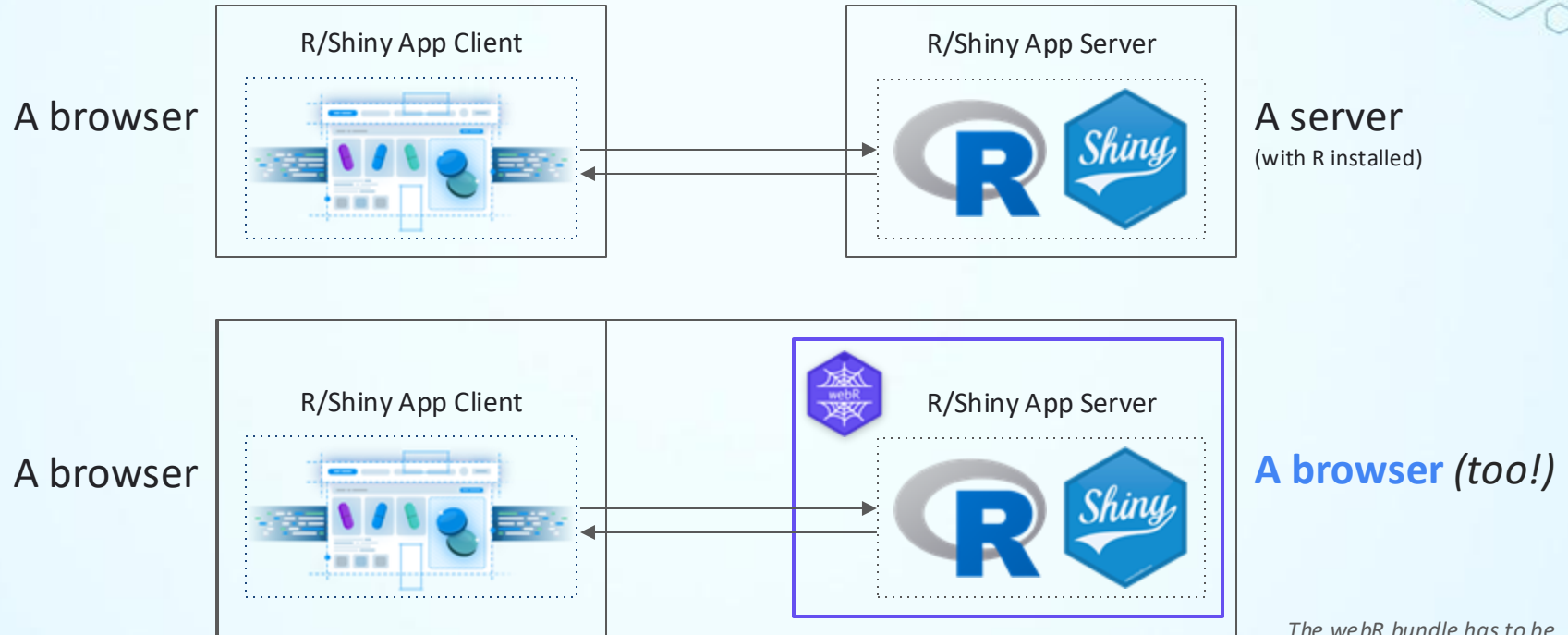




What is webR?

What is webR?

Explained using an R/Shiny setting



The webR bundle has to be served by an HTTP server



webR-based method to deploy pilot's Shiny App

Goals

- 🎯 Can the the **app run in webR**
- 🎯 Keep as many features from a previous iteration as possible, *ideally all*

The journey

Running an app locally

MENTAL GYMNASTICS

shiny

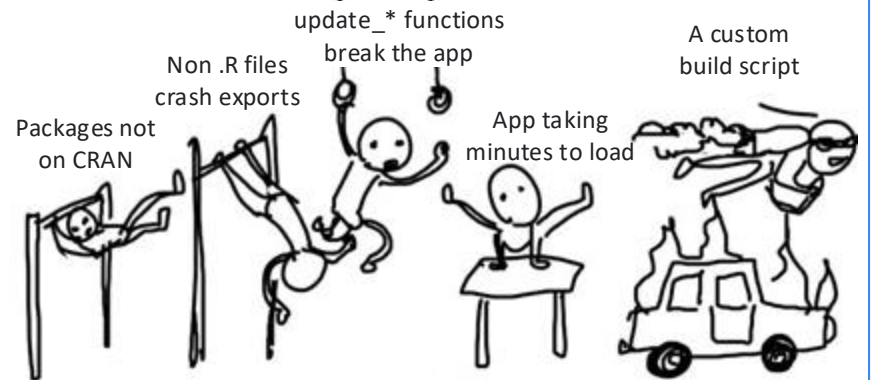
::

runApp()



Running an app in webR

MENTAL GYMNASTICS



The journey

```
library(shinylive)
```

utils.R

```
# build application
build_app <- function(...) {
  # ...
}

# run webassembly via http server
run_app_webassembly <- function(...) {
  # ...
}

# run traditional Shiny application
run_app_shiny <- function(...) {
  # ...
}

# bundle zip archive
create_ectd_bundle <- function(...) {
  # ...
}
```

- utils.R to build the bundle
- Shim {teal} and {teal.data} functionalities

```
TealShim <- R6::R6Class(
  "Shim",
  private = list(
    initialize = function(adsl, adas, adtte, adlb) {
      private$adsl <- adsl
      private$adas <- adas
      private$adtte <- adtte
      private$adlb <- adlb
    },
    get_data = function(name, ...) {
      switch(name,
        ADSL = private$adsl,
        ADAS = private$adas,
        ADTTE = private$adtte,
        ADLB = private$adlb
      )
    },
    get_varlabels = function(name, ...) {
      formatters$varlabels(self$get_data(name))[cols]
    },
    get_filters = function() {
      c()
    }
  ),
  public = list(
    get_data = NULL
  )
)
```

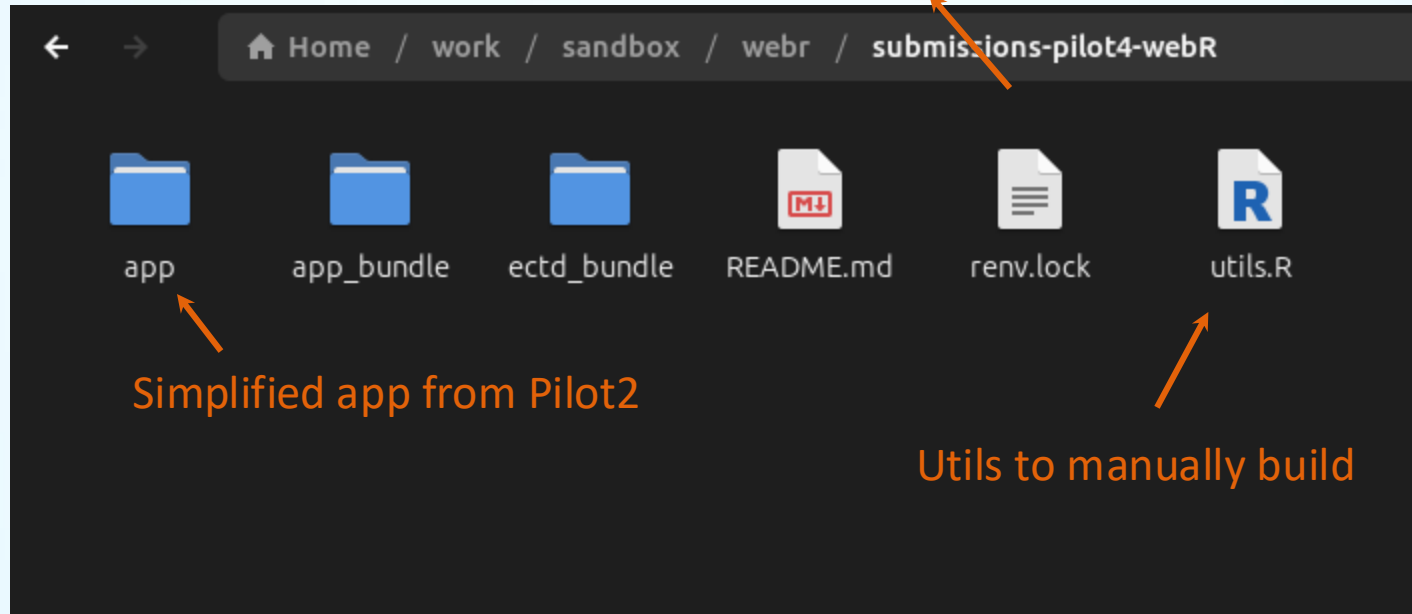
A shiny

teal: Interact
Shiny Web

CRAN 0.16.0 downloads 1

How does it look?

Package versions used to build current bundle (reproducibility)



Simplified app from Pilot2

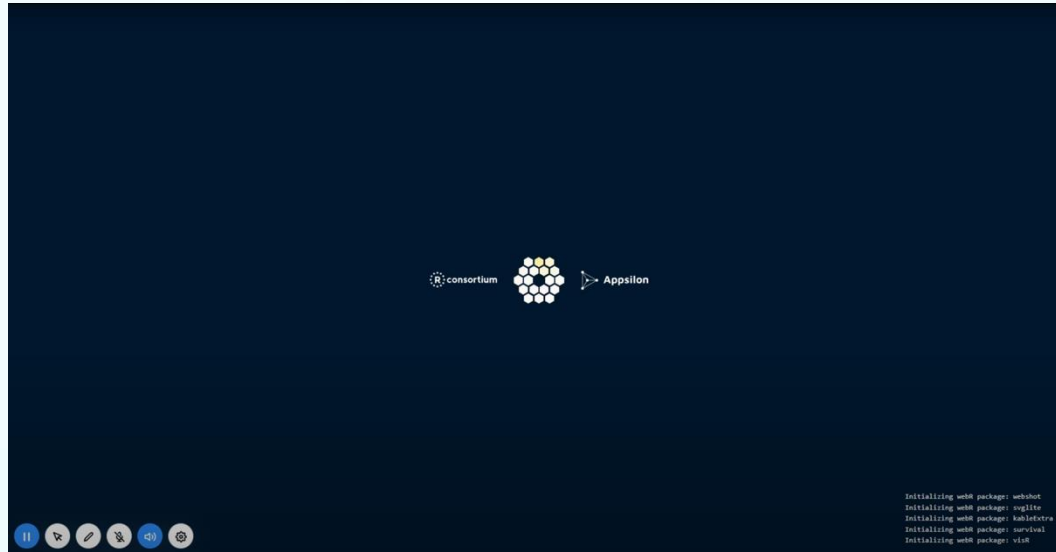
Utils to manually build

<https://github.com/RConsortium/submissions-pilot4-webR>

The result

Try it **yourself**

<https://submissions-pilot4-webr.netlify.app>



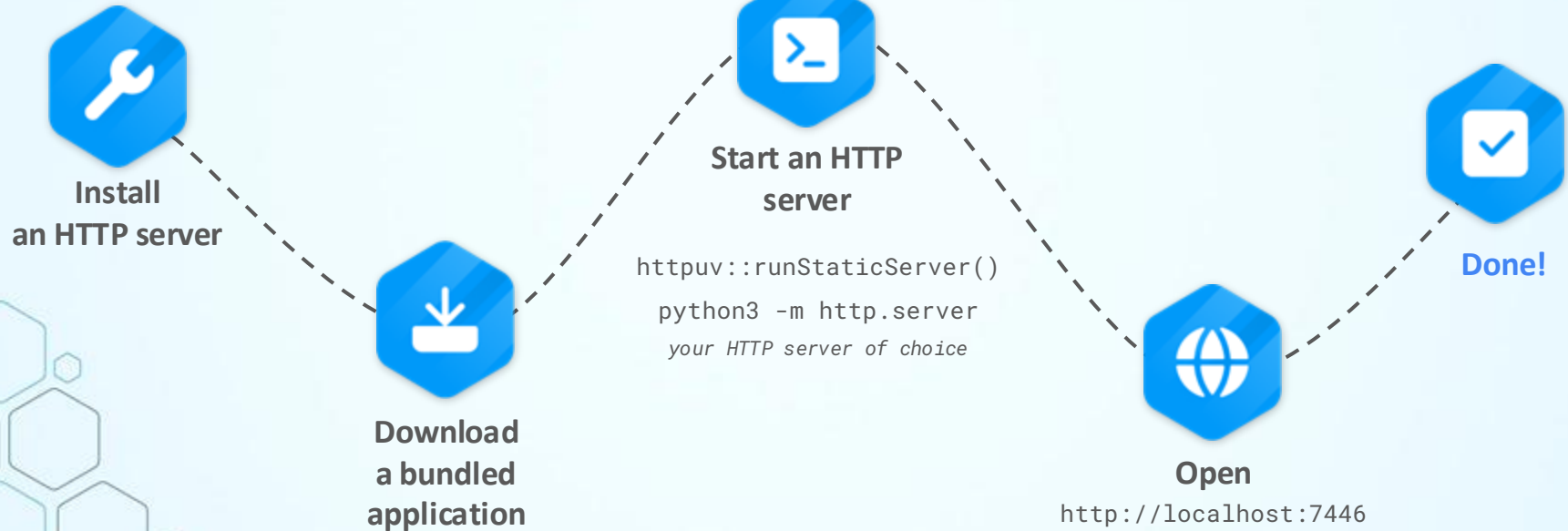
} Package loading notification

You can build it yourself too!

[Code is available on GitHub \(RConsortium/submissions-pilot4-webR\)](https://github.com/RConsortium/submissions-pilot4-webR)

Installation instructions

The future with





Which version will you use for your submission?

Want to contribute?



Join Submissions WG

Improving practices of R-based clinical trial
regulatory submissions.



Join Repositories WG

Building a public, validation-ready
resource for R packages

