Asynchronous programming in R

(sort of, but not really)

Winston Chang

RStudio

Asynchronous programming

Parallelism

Concurrency

Event-driven programming

Asynchronous programming

When you call a function, it doesn't block

Parallelism

Doing multiple things at the same time

Concurrency

Seem like doing multiple things at the same time

Event-driven programming

Events cause code to run

The later package

- later provides an ***event loop***
- Event loop = a queue of functions that will run in the future
- Similar to `setTimeout()` in JavaScript

```r
library(later)
done <- FALSE
later(
  function() {
    message("Hello world!")
    done <<- TRUE
  },
  delay = 5
)

while (!done) {
  run_now()
}
```

Plot watcher

```
data <- NULL
last_value <- NULL

plot_watch <- function() {
  if (!is.null(data) && !identical(data, last_value)) {
    plot(data$x, data$y)
    last_value <<- data
  }
  later(plot_watch, 0.25)
}

plot_watch()
```

later's C API

- From C code, you can call **`later()`** to schedule another C function to execute.
- That C function can call an R function.
- The C **`later()`** function is thread-safe.

Real-world uses of *later*

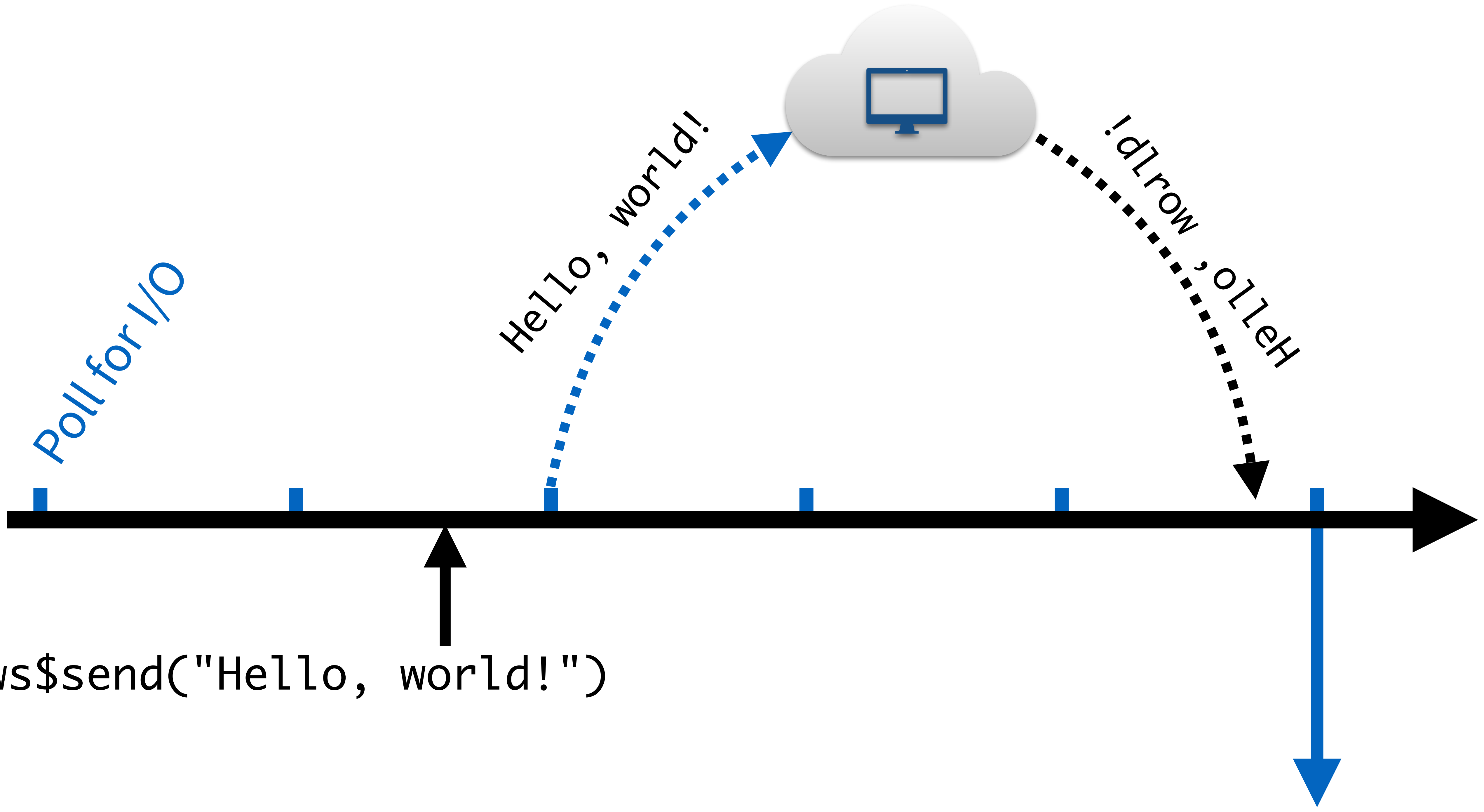
WebSocket client

```r
library(websocket)
ws <- WebSocket$new("ws://127.0.0.1:4000/")

ws$onMessage(function(event) {
  message('Received: "', event$data, '"\n', sep="")
})

ws$send("Hello, world!")
```

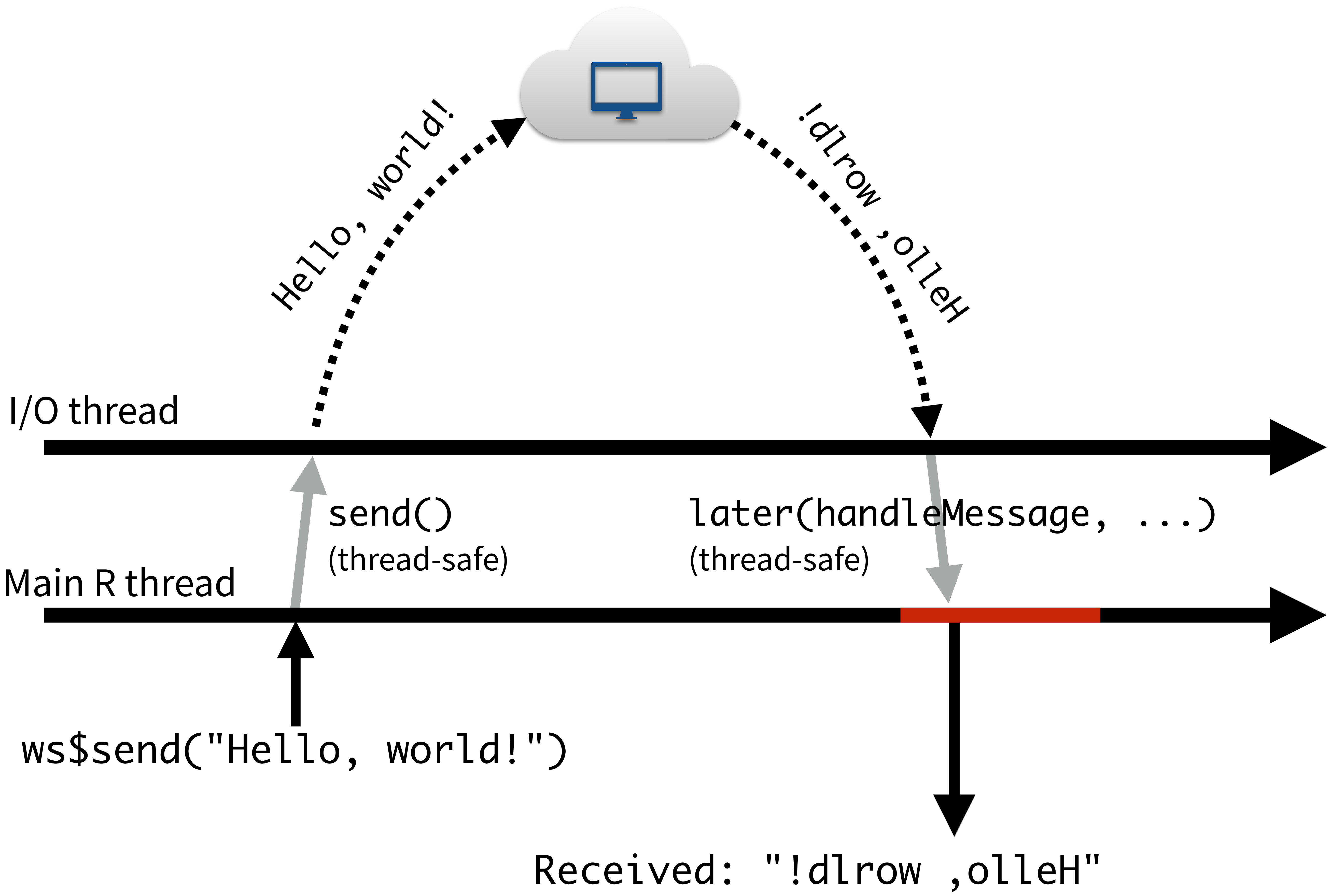
WebSocket - polling



```
ws$send("Hello, world!")
```

```
Received: "!dlrow ,olleH"
```

WebSocket - threaded



httpuv web server

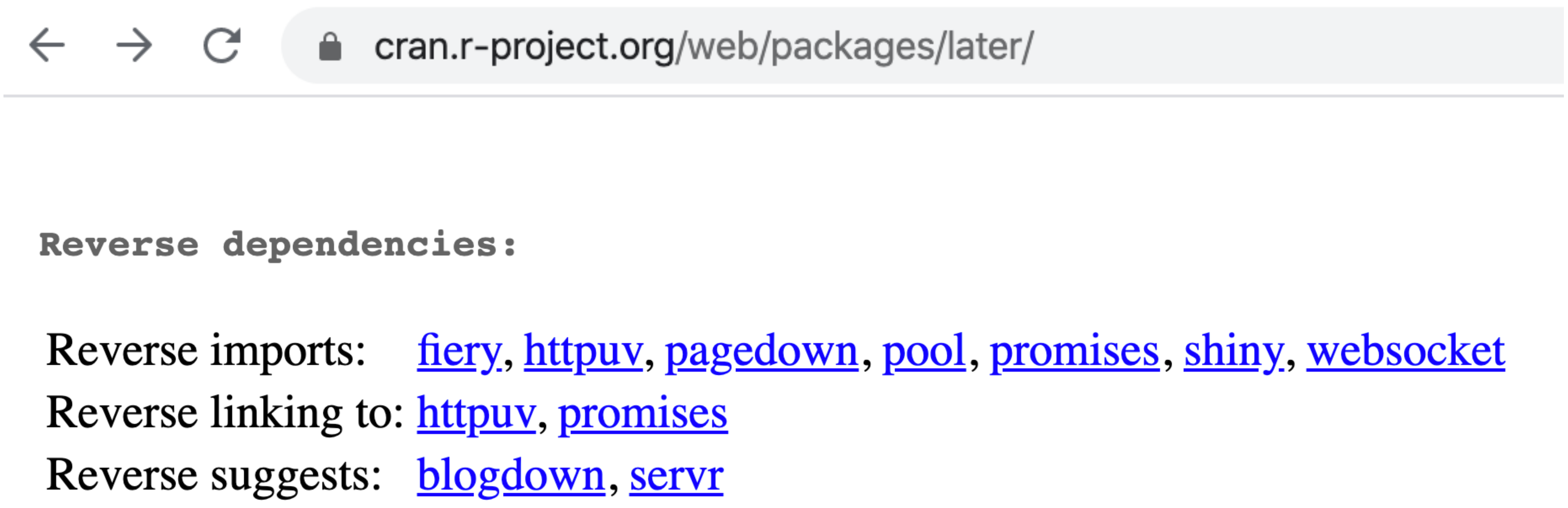
```r
library(httpuv)
s <- startServer("127.0.0.1", 5000,
  list(
    call = function(req) {
      body <- paste0("<h2>Time: ", Sys.time(),
       "<br>Path requested: ", req$PATH_INFO,
       "</h2>")

      list(
        status = 200L,
        headers = list('Content-Type' = 'text/html'),
        body = body
      )
    }
  )
)
```

- WebSocket server: String reverser
- Web server: Time
- Web server: Remote R "console"
- Shiny application
- Plot data watcher
- Headless Chrome client

… all in one R process

Packages that use later



cran.r-project.org/web/packages/later/

Reverse dependencies:

Reverse imports:	fiery, httpuv, pagedown, pool, promises, shiny, websocket
Reverse linking to:	httpuv, promises
Reverse suggests:	blogdown, servr

Asynchronous programming

Parallelism

Concurrency

Event-driven programming

https://github.com/wch/2020-01-later