

dRuby

in the real-world embedded systems.

キヤノンメディカルシステムズ株式会社

医用コンポーネント技術開発センター 関将俊 園川龍也

Sep, 2021



recap last episode



Rinda

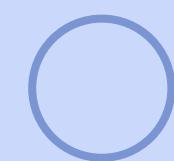
in the real-world embedded systems.

seki@ruby-lang.org



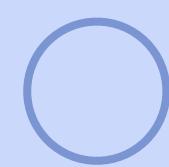
Robot Telescope using Rinda tuplespace



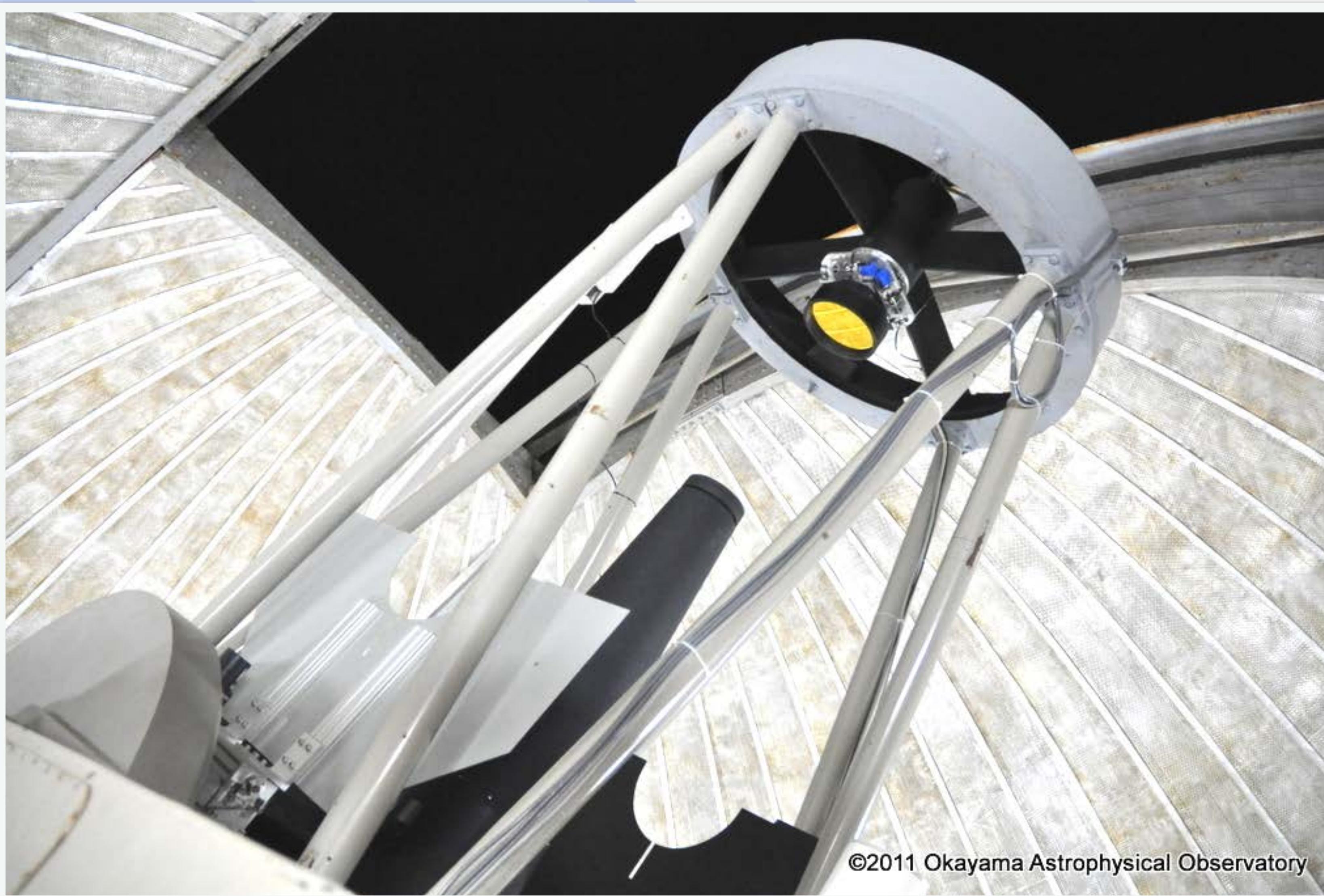


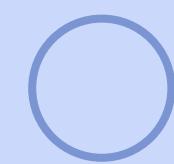
Okayama Astrophysical Observatory Wide-Field Camera

- *Publications of the Astronomical Society of Japan*, Volume 71, Issue 6, December 2019, 118, <https://doi.org/10.1093/pasj/psz117>
- 第5回 可視赤外線観測装置技術ワークショップ 2015年12月 「岡山天体物理観測所広視野赤外カメラの製作」 柳澤 順史（国立天文台） https://www.astr.tohoku.ac.jp/~akiyama/astroinst2015/20151207_yanagisawa.pdf



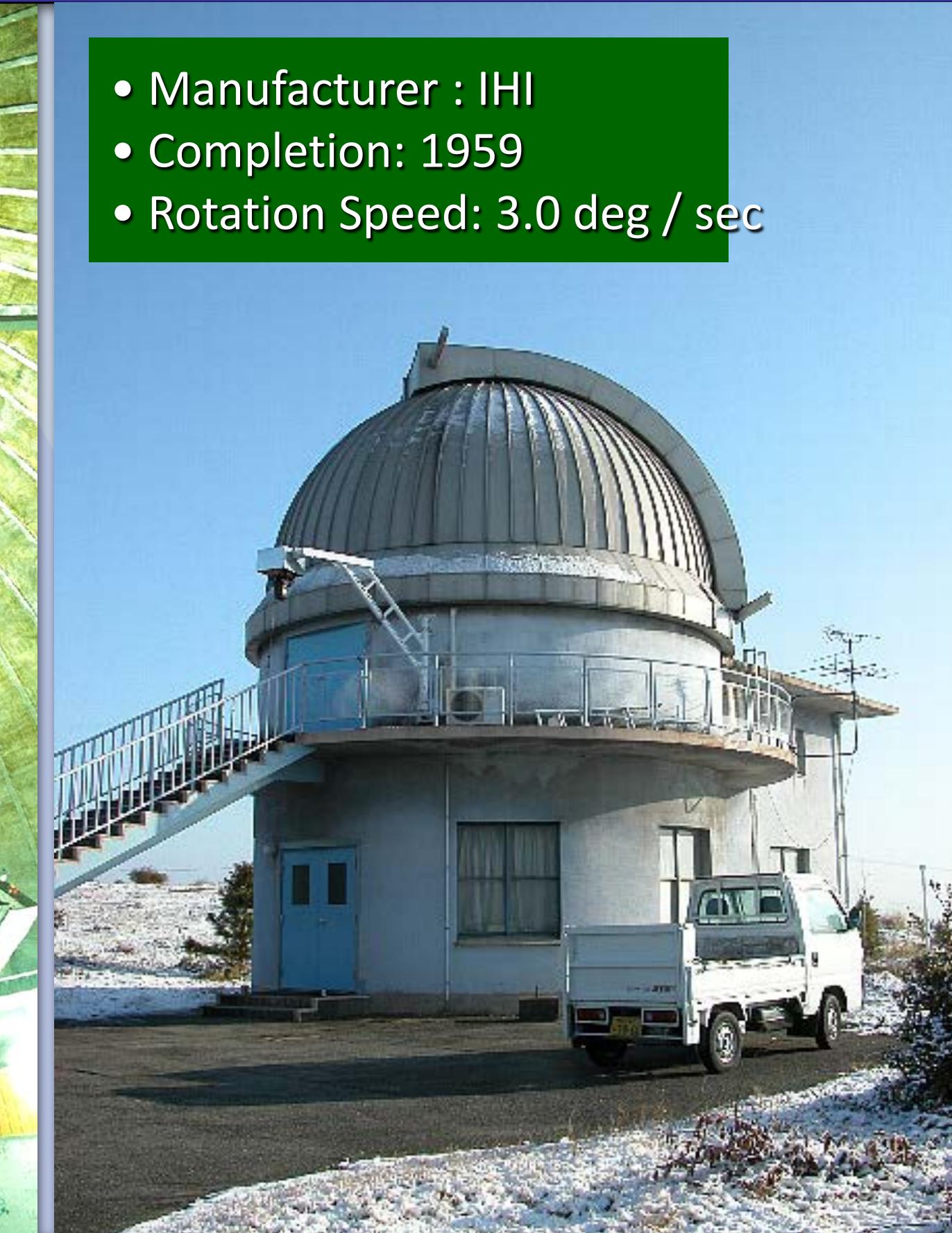
OAOWFC





renewal of the existing classical Cassegrain telescope

The OAO 0.9m Telescope before renewal



dRuby

in the real-world embedded systems.

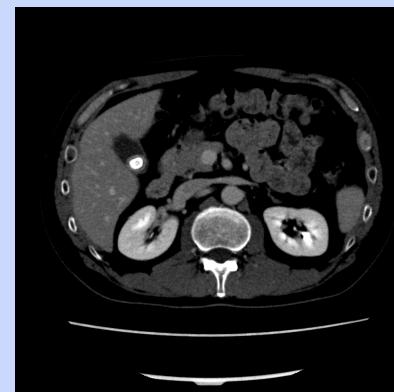
キヤノンメディカルシステムズ株式会社

医用コンポーネント技術開発センター 関将俊 園川龍也

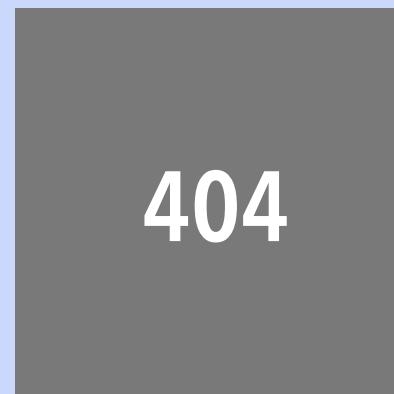
Sep, 2021



About us



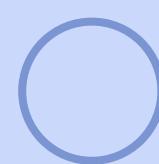
Masatoshi Seki, Ruby Core Committer (dRuby, Rinda, ERB), Programmer



Tatsuya Sonokawa, Programmer



Miwa Fukaya, Tochigi RubyKaigi Content Committee Chair, Software tester



Our products

Yukihiko Matsumoto @yukihiro_matz · 1日
金曜日、深夜に38.5°Cの熱が出て、この時期熱はヤバイなあと心配してたら、朝には平熱になってた。土曜日朝に病院に行ったらPCR検査されて、陰性。良かった

…

2 153 ↑

Yukihiko Matsumoto
@yukihiro_matz

その後、お腹が痛いのでCTとか撮ってもらったら、憩室炎らしいということで、2, 3日絶食のため入院だとか。こんなことで人生初入院になるとは。

19:38 · 2021/07/26 · TweetDeck

3件のリツイート 31件のいいね

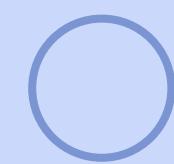
…

2 153 ↑

Point of Care Testing

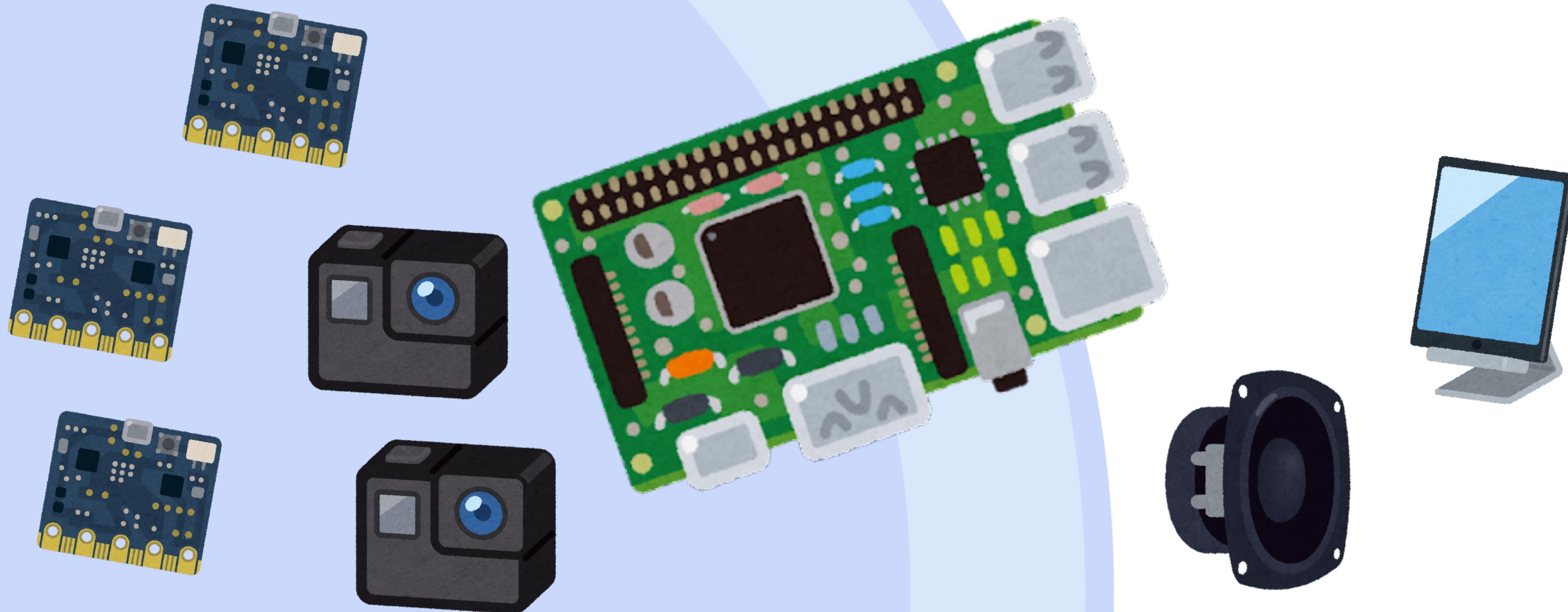
- Rapid tests for the Respiratory infection diseases such as SARS-CoV-2, Influenza, RS virus
- Next Generation
- +UI, Parallel processing





Hardware

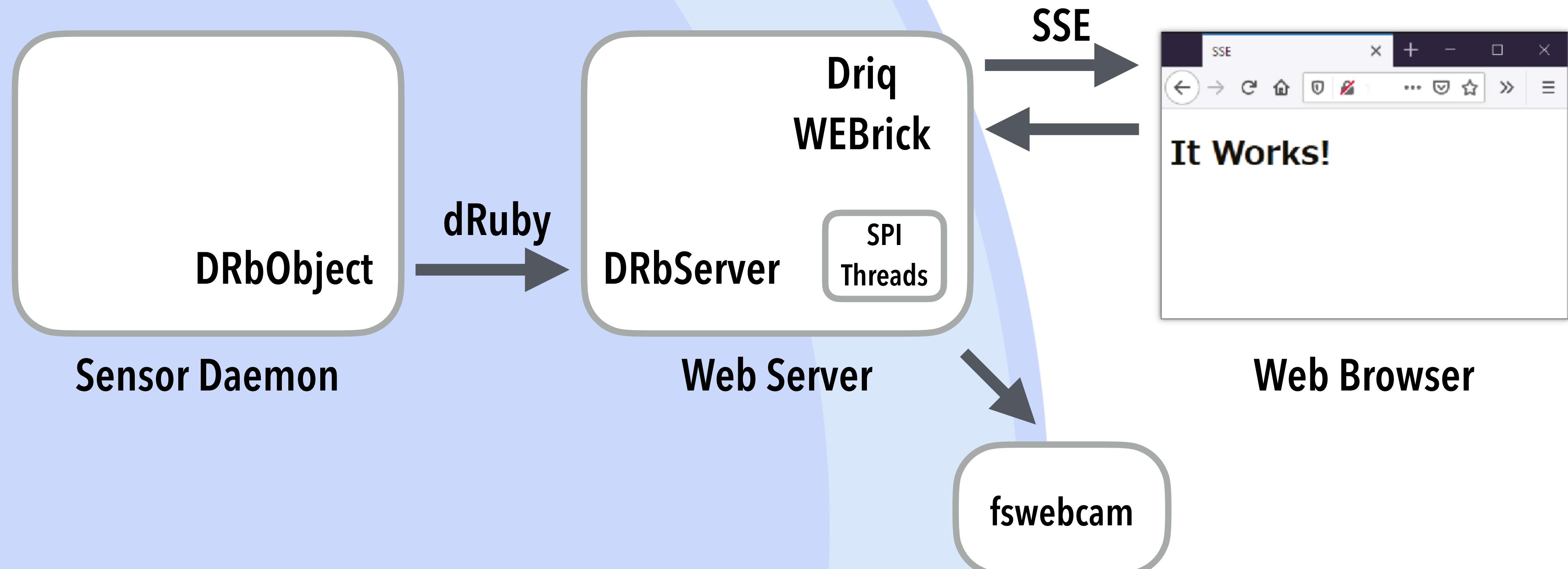
- Raspberry Pi + LCD monitor + cameras + sensors / devices





Software

- Chrome kiosk-mode + Single Page Application
- WEBrick + Server Sent Events + dRuby + SPI, I2C, GPIO





開発日記

- My first ruby script
- GUI
- WiringPi
- webcam response time



開発日記

- My first ruby script
- GUI
- WiringPi
- webcam response time



My first ruby script

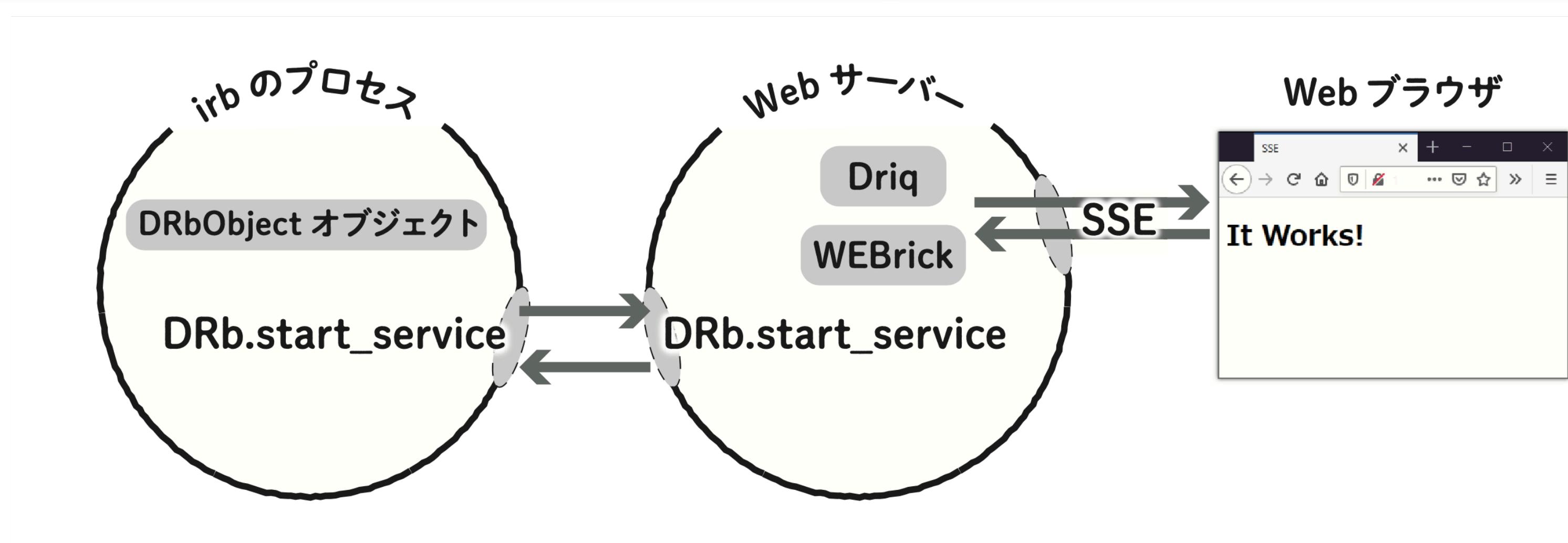
- Sono-san has no experience with Ruby, Linux, Ras-pi, and Web App
- Windows, C++
- pair work
- We need a textbook...

textbook

- I happened to be writing an article about Ruby, Ras-pi, and Web App

n-monthly Lambda Note Vol.2 No.1





AM DA OTE

パケットの設計から見るQUIC 西田佳史

dRubyで楽しむ分散オブジェクト 関将俊

Biscuitで学ぶ、GoならわかるOS実装 渡川よしき



Vol.2, No.1

Mar.2020



OSOS-16M
Ron's Joy



似てる...



開発日記

- My first ruby script
- GUI
- WiringPi
- webcam response time

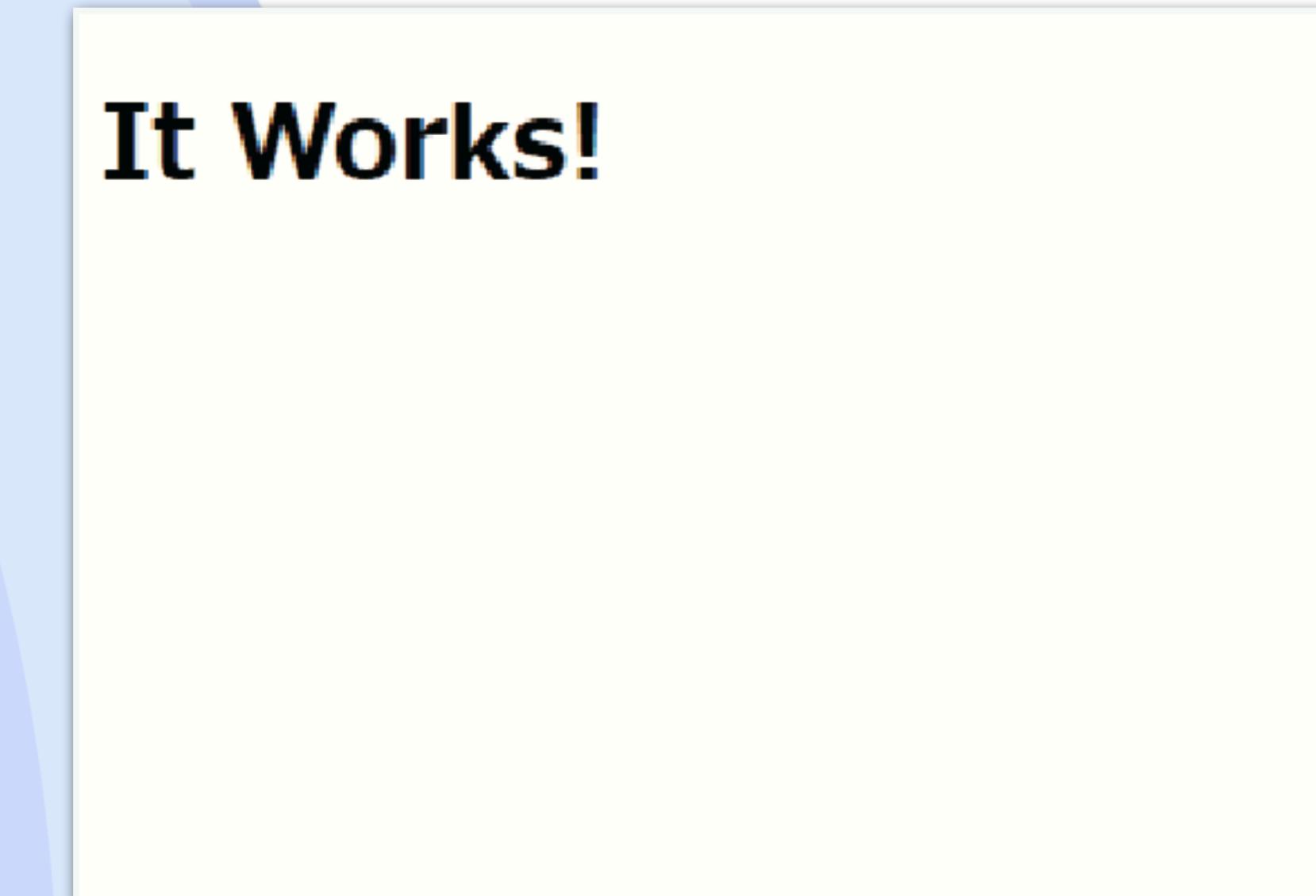
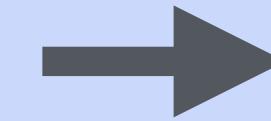
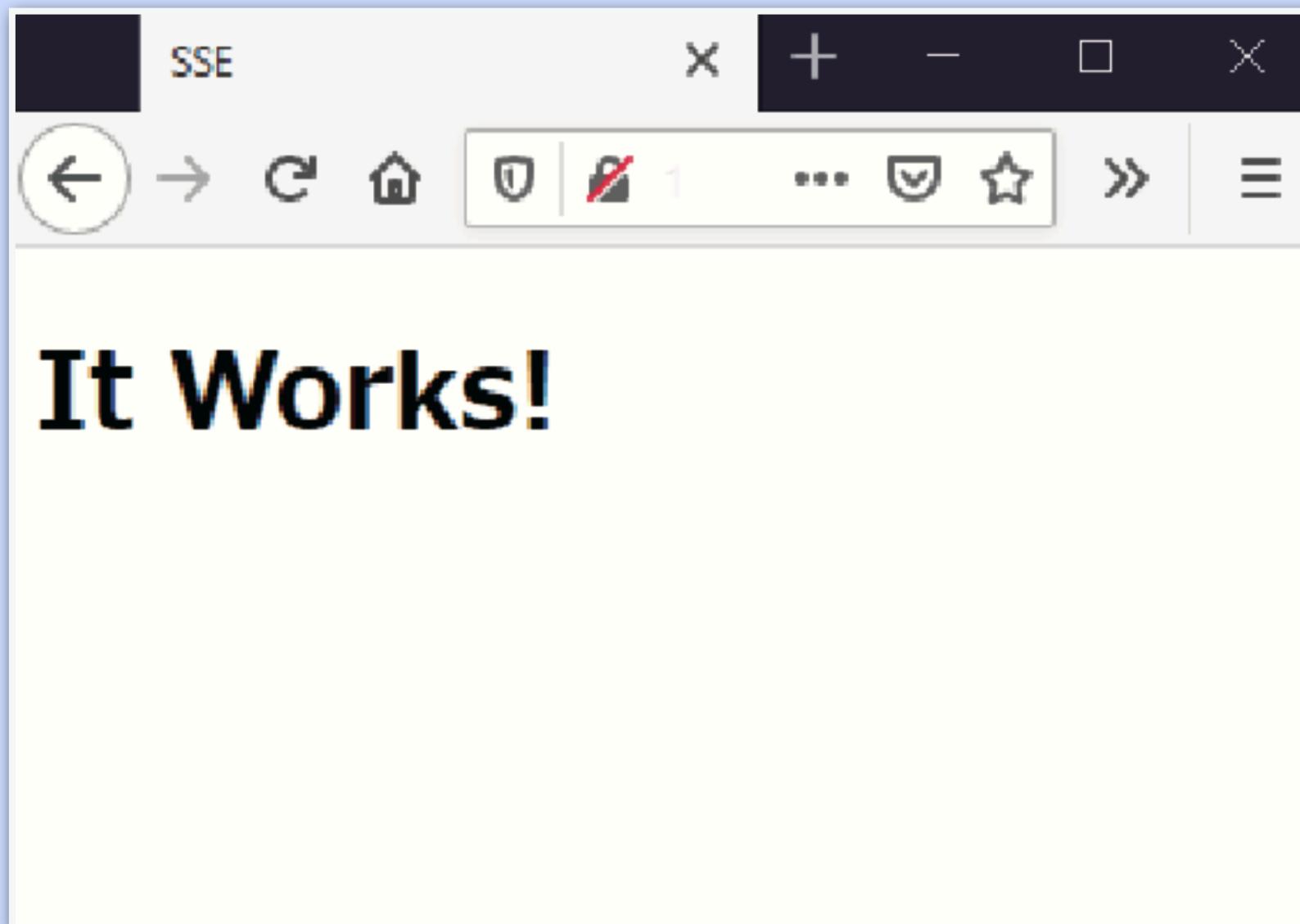


GUI

- Kiosk mode Chrome
- Single Page Application
- Driq, RDstream, Server Sent Events



Kiosk mode



kiosk mode

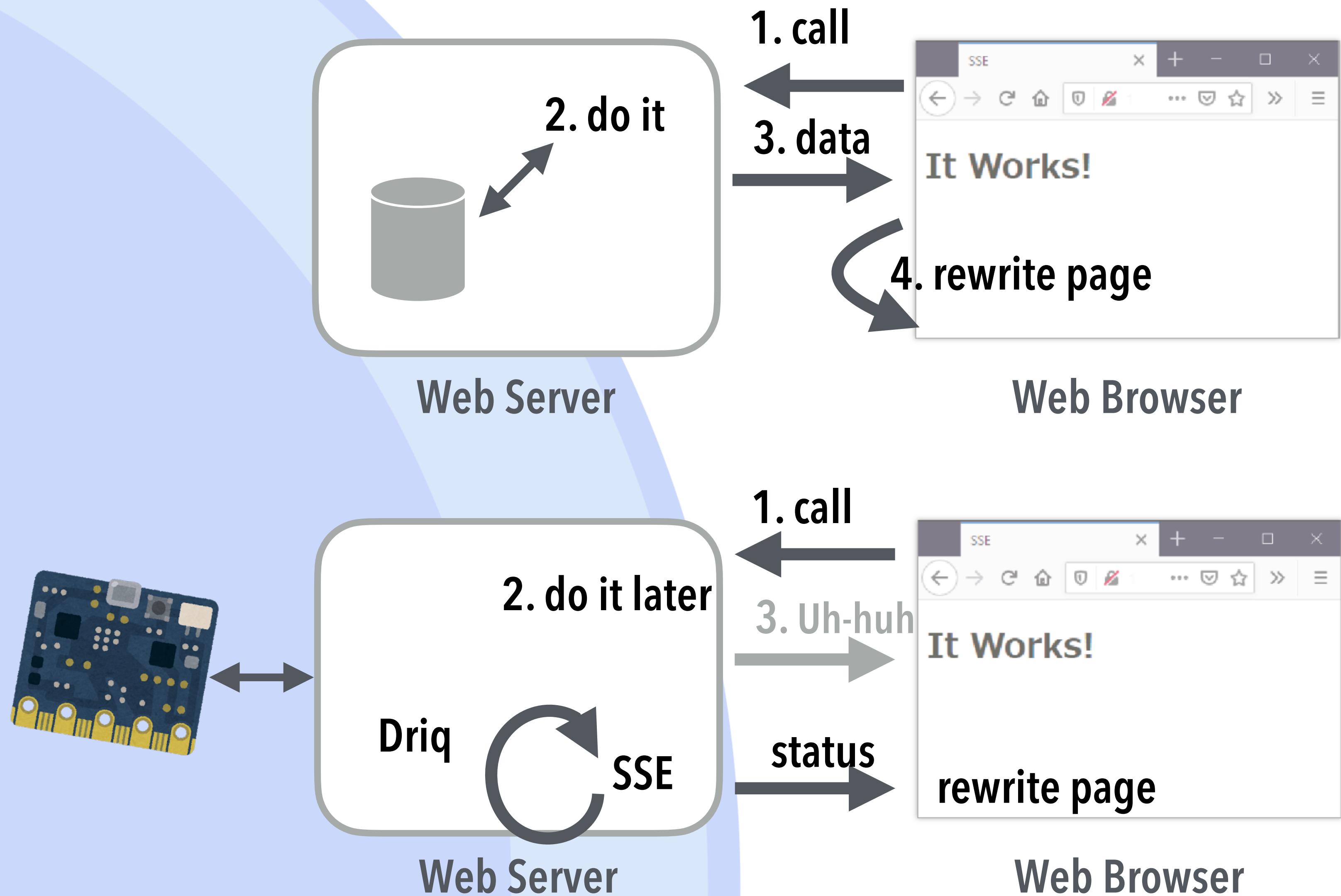


Single Page Application

- Dynamically rewrites the current page instead of loading the entire page.
- Vanilla JS
- Update mechanism

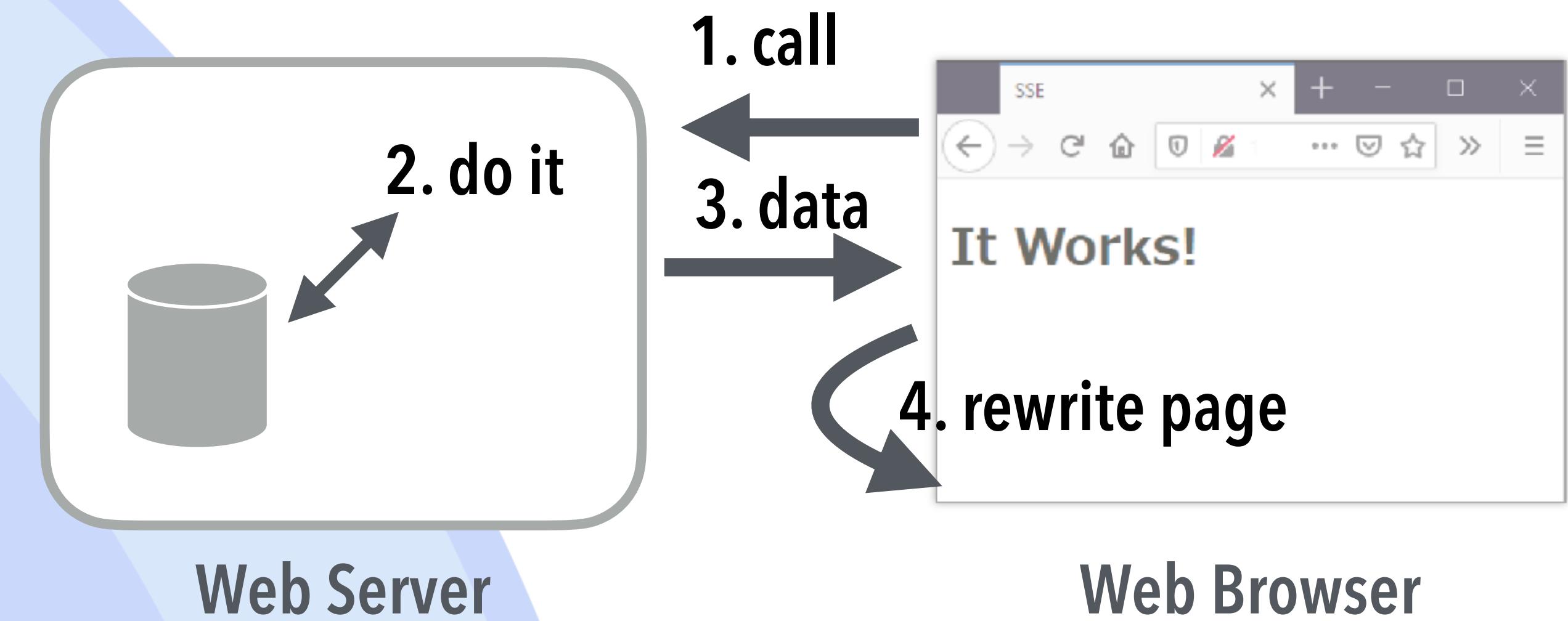


Update mechanism



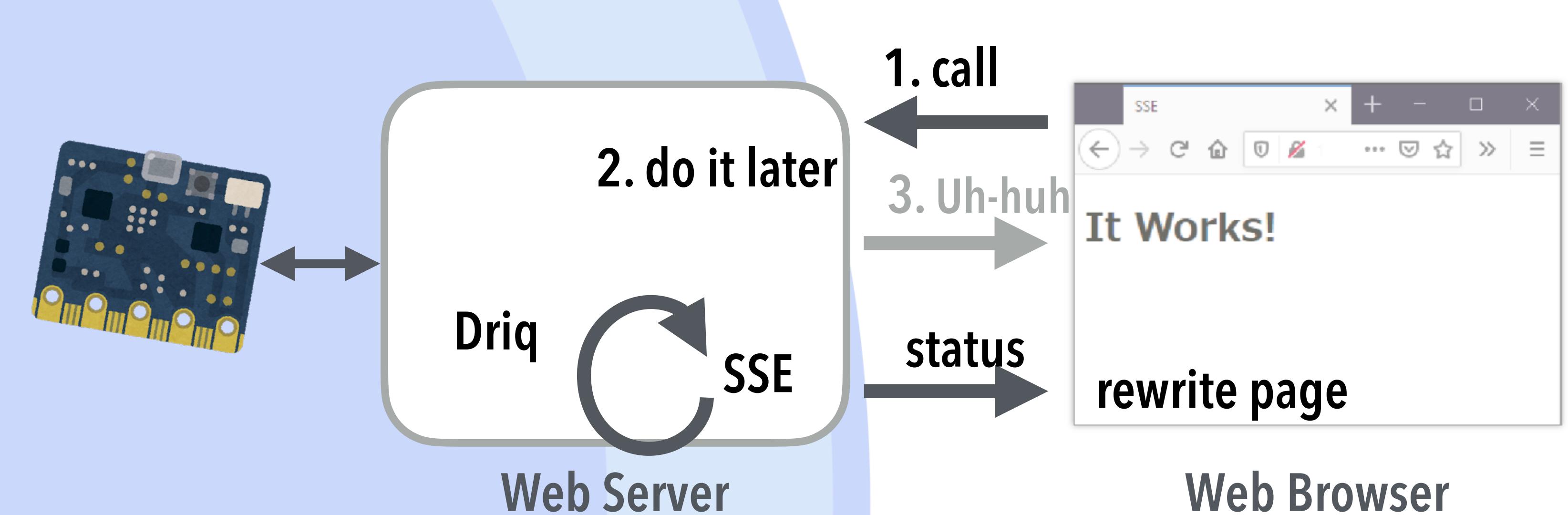


pull





push

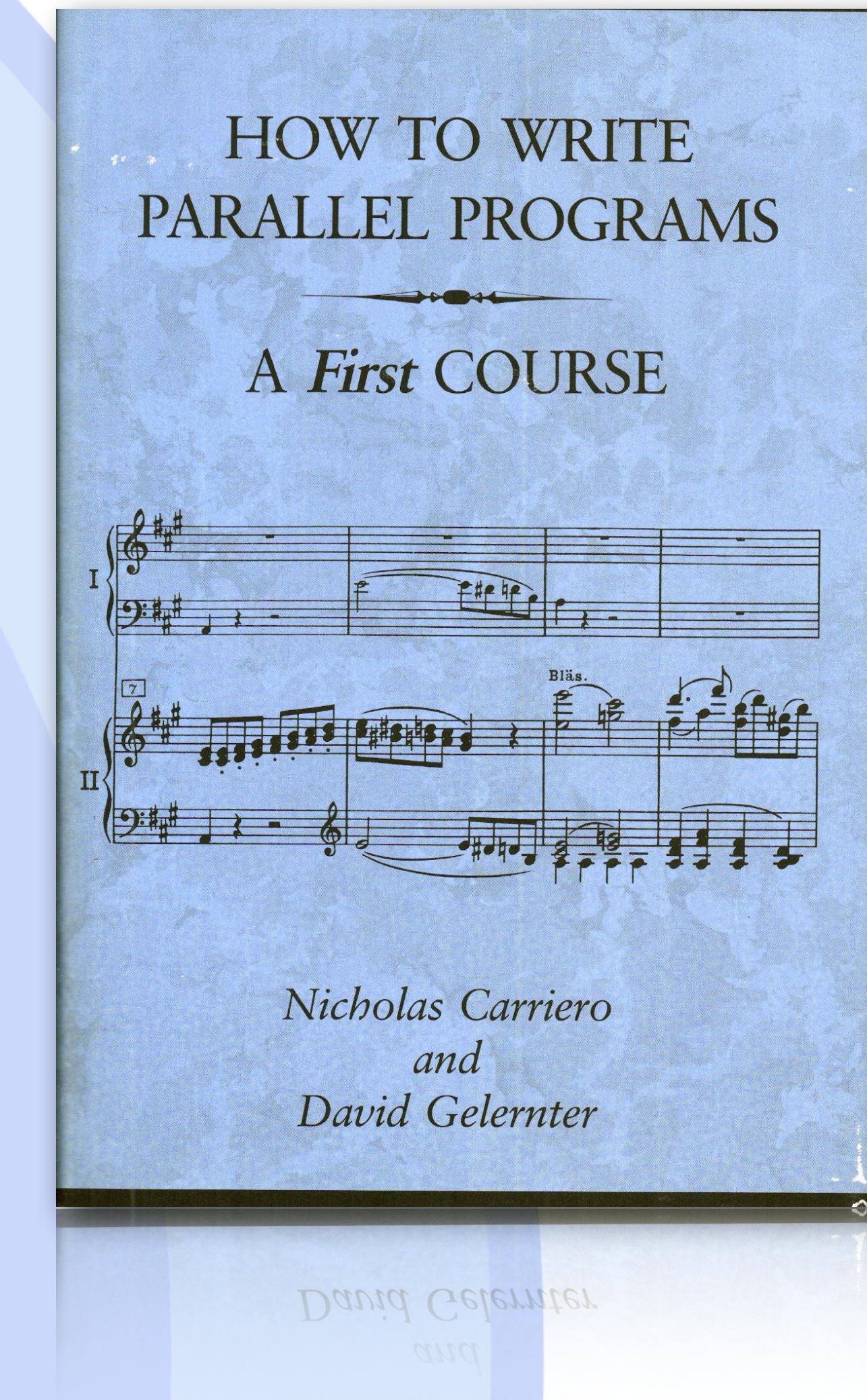
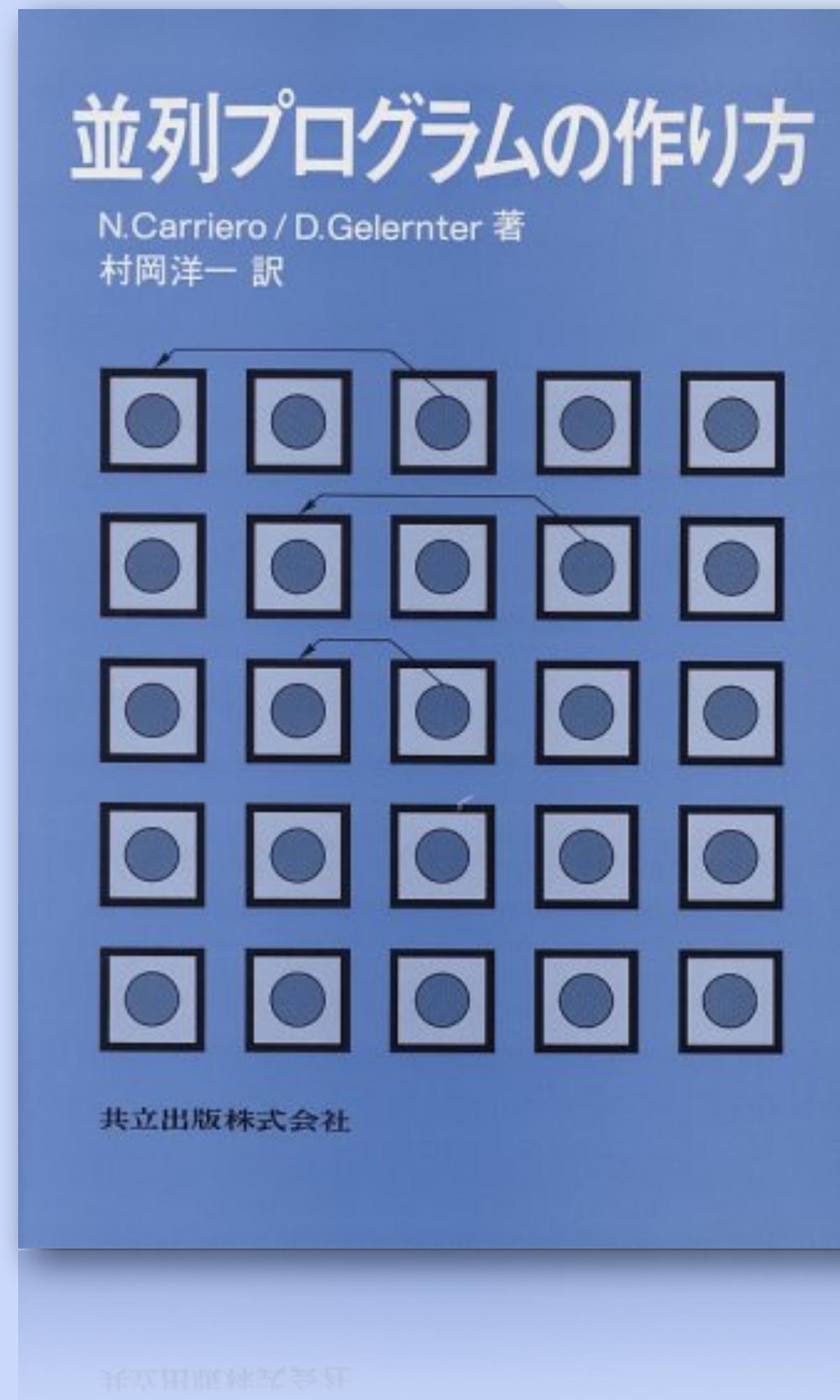




Driq

- in-memory RDstream
- Basic Distributed Data Structures
- SSE interface

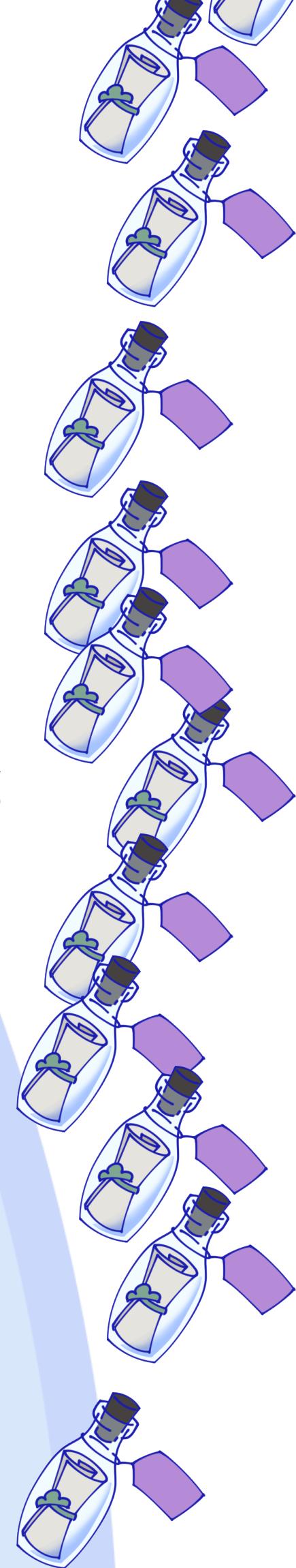
Basic Distributed Data Structures

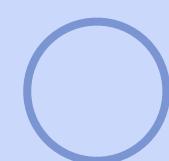




INstream, RDstream

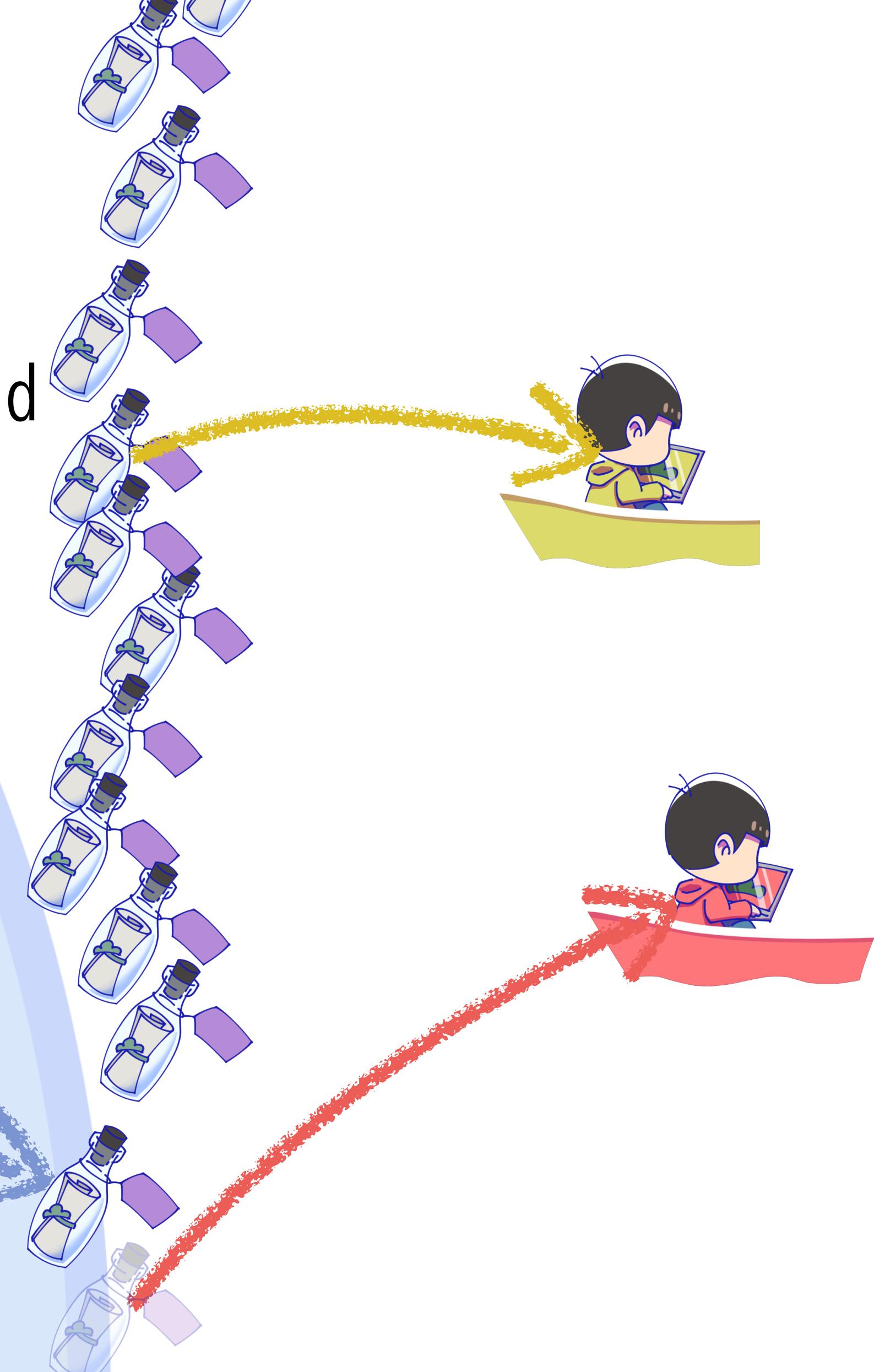
- IN stream consumes elements as they are read
- RD stream does not consume elements as they are read.





RDstream

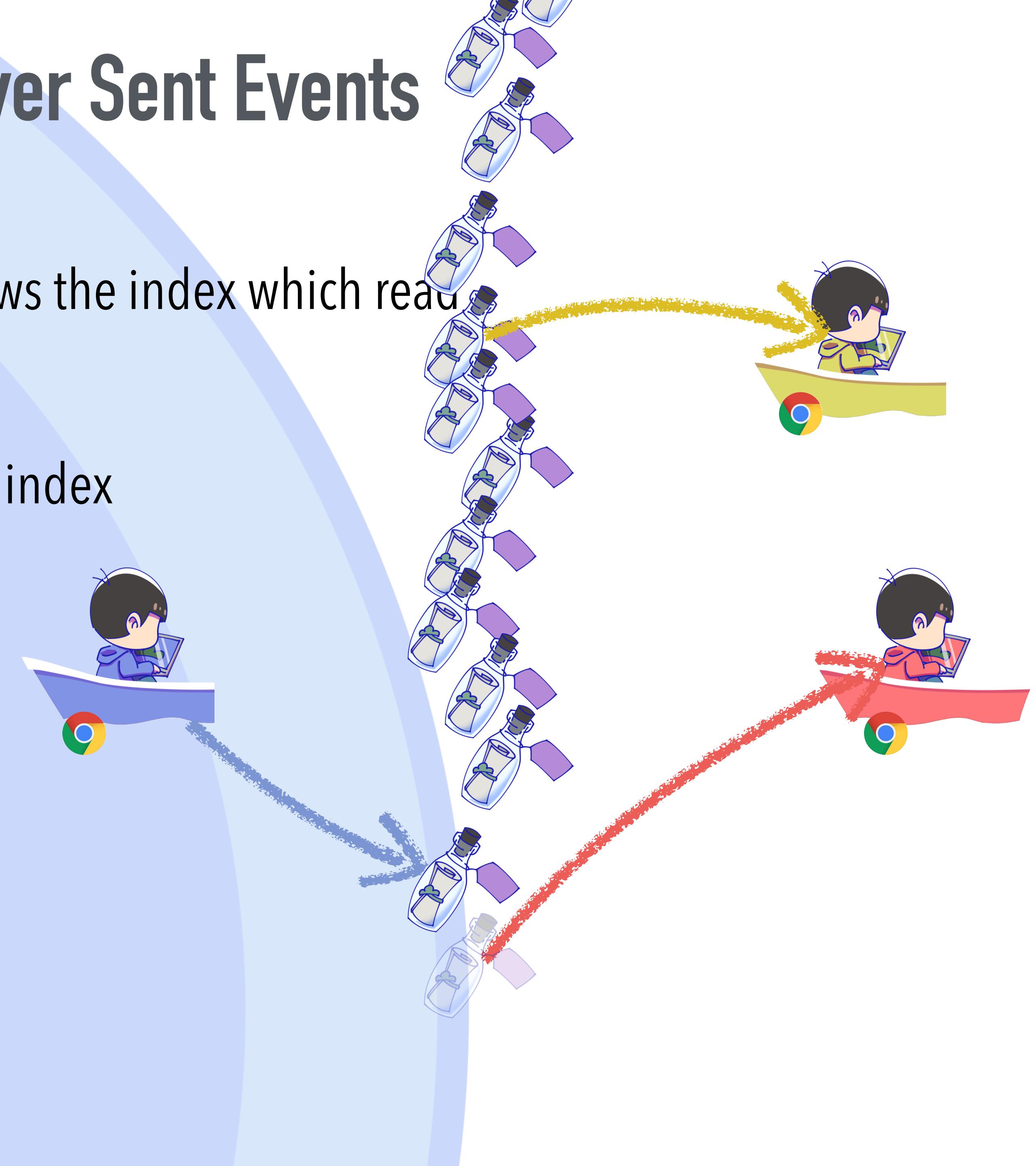
- each reader knows the index which read
- read at own pace
- resume reading at index





Driq + Server Sent Events

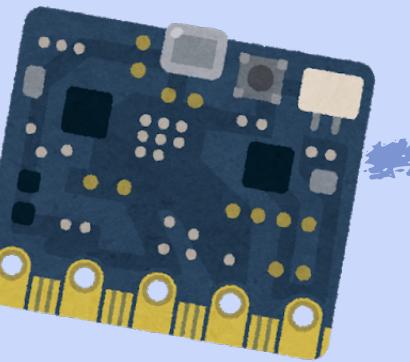
- each **browser** knows the index which read
- read at own pace
- resume reading at index



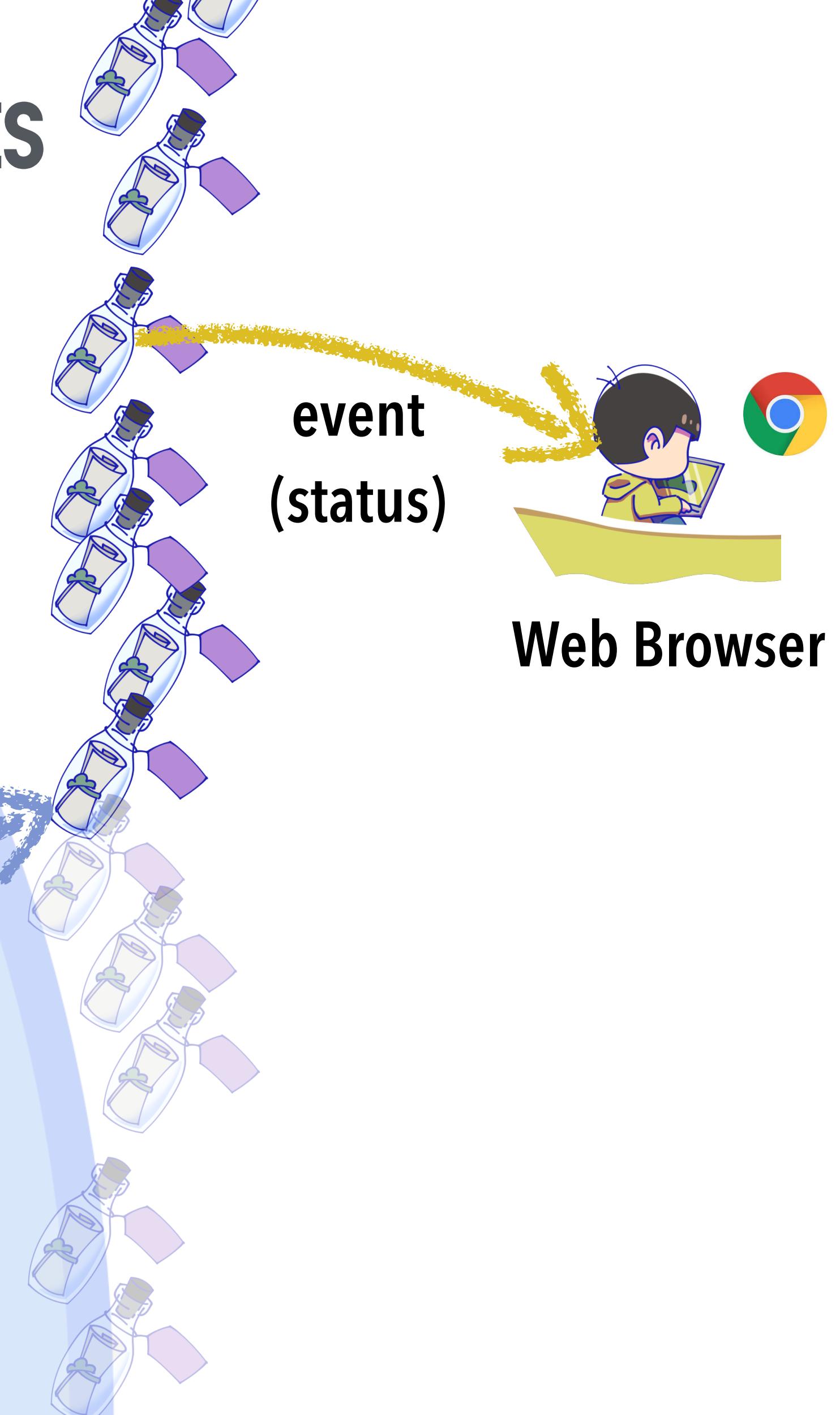


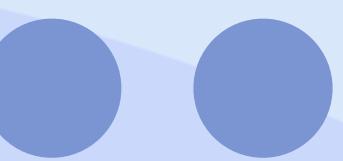
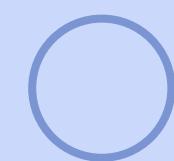
Driq + Server Sent Events

- Asynchronous write

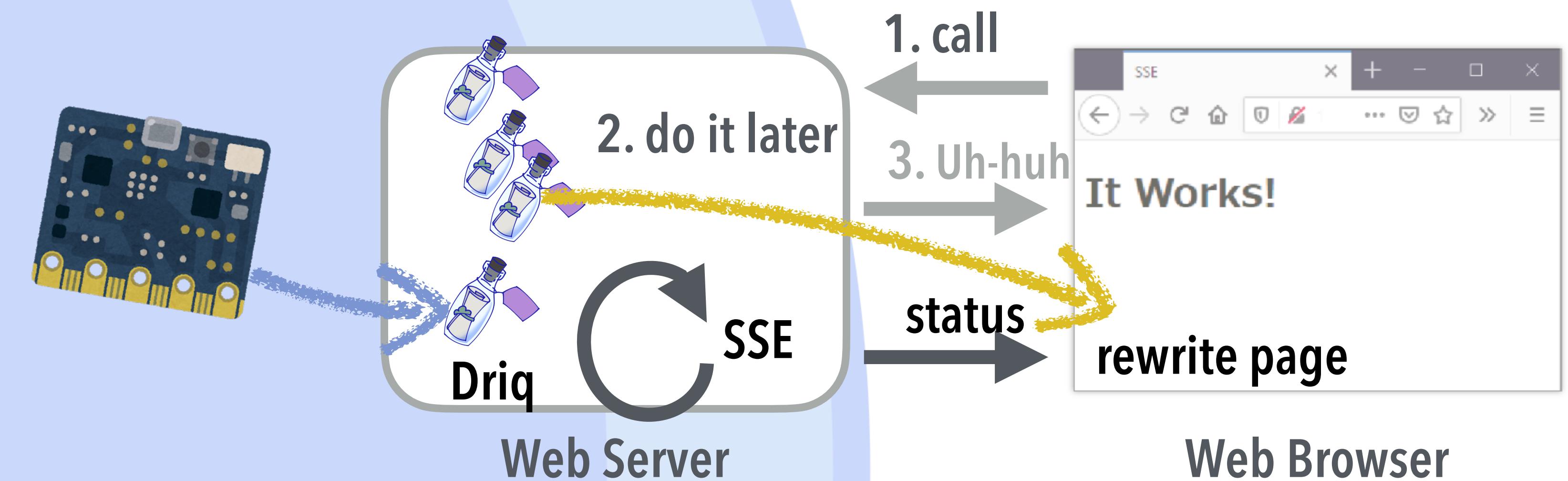


Sensor Daemon





Driq + Server Sent Events





開発日記

- My first ruby script
- GUI
- WiringPi
- webcam response time



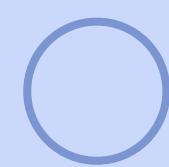
WiringPi



SPI



wiringpi gem



wiringpi gem problem

```
$ irb -r wiringpi
irb(main):001::0> Wiringpi.wiringPiSetupGpio
Unable to determine hardware version. I see: Hardware : BCM2835
- expecting BCM2708 or BCM2709.

..
```

- wiringpi gem contains code for the old WiringPi library. (static link)
- wiringPi - deprecated... | <http://wiringpi.com/wiringpi-deprecated/>



wiringpi gem problem

- fixed
- use shared library
- <https://github.com/seki/WiringPi-Ruby>

WiringPi（本体）のコードを読んでいて気づいたが、RubyからならC同様に全部の機能が呼べるので要らなかったかも



開発日記

- My first ruby script
- GUI
- WiringPi
- webcam response time

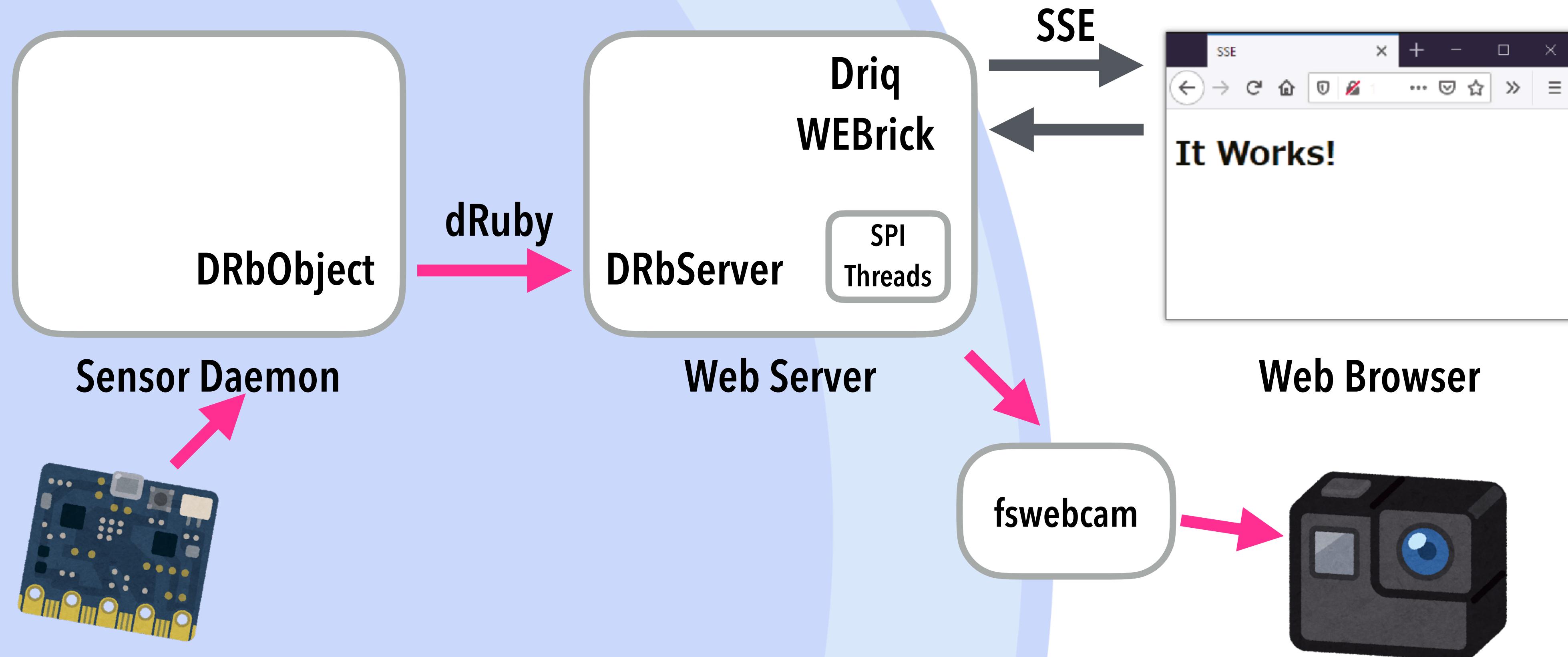


WebCam response time

- Take a picture with the webcam when the signal
- too slow
- quick hack

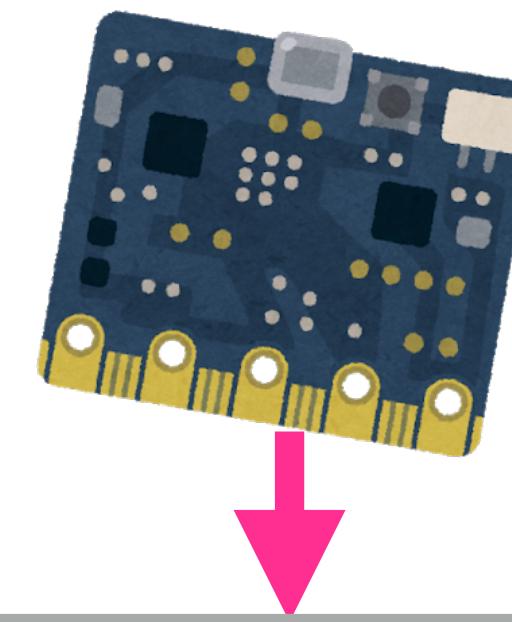
WebCam response time

- fork/exec fswebcam when the signal



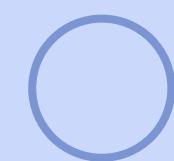
fswebcam

- when signal, then setup device, then capture, then write to file



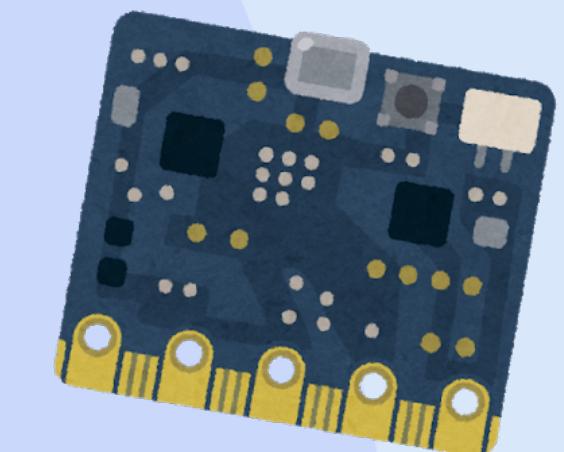
1. setup device
2. capture
3. write to file

fswebcam



fswebcam quick hack

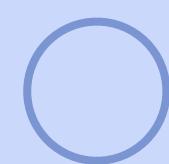
- setup device, then **wait for a cue**, then capture, then write to file



pre execution

- 0. setup device
- 1. **wait for a cue**
- 2. capture
- 3. write to file

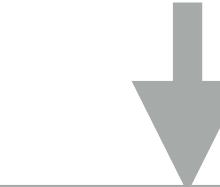
fswebcam



--delay option

-D, --delay <delay>

Inserts a delay after the source or device has been opened and initialised, and before the capture begins. Some devices need this delay to let the image settle after a setting has changed. The delay time is specified in seconds.



1. setup device
2. **delay**
3. capture
4. write to file

fswebcam



implementation

-D, --delay <delay>

Inserts a delay after the source or device has been opened and initialised, and before the capture begins. Some devices need this delay to let the image settle after a setting has changed. The delay time is specified in seconds.

```
774     /* Delay to let the image settle down. */
775     if(src->delay)
776     {
777         MSG("Delaying %i seconds.", src->delay);
778         usleep(src->delay * 1000 * 1000);
779     }
```



wait for a cue

```
774 +     /* Wait for a cue from stdin. */
775 +     if(src->delay == 0xffffffff)
776 +
777 +         MSG("Waiting for a cue from stdin.");
778 +
779 +         char buf;
780 +         read(0, &buf, 1);
781
774 781         /* Delay to let the image settle down. */
775 -     if(src->delay)
782 +     else if(src->delay)
776 783     {
777 784         MSG("Delaying %i seconds.", src->delay);
778 785         usleep(src->delay * 1000 * 1000);
779 786 }
```



```
→ cam = IO.popen("./fswebcam -D -1 foo.png", "w")
2 # wait signal
3 cam.close
```

```
774 + /* Wait for a cue from stdin. */
775 + if(src->delay == 0xffffffff)
776 +
777 + MSG("Waiting for a cue fr
778 + char buf;
779 + read(0, &buf, 1);
780 +
774 781 /* Delay to let the image settle
775 - if(src->delay)
782 + else if(src->delay)
776 783 {
777 784 MSG("Delaying %i seconds.
778 785 usleep(src->delay * 1000
779 786 }
```



cue

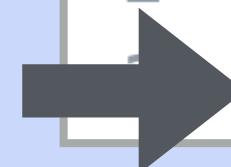
```
→ cam = IO.popen("./fswebcam -D -1 foo.png", "w")
2 # wait signal
3 cam.close
```

```
774 + /* Wait for a cue from stdin. */
775 + if(src->delay == 0xffffffff)
776 +
777 +
778 +
779 +
780 +
774 781 /* Delay to let the image settle */
775 - if(src->delay)
782 + else if(src->delay)
776 783 {
777 784 MSG("Delaying %i seconds.\n", src->delay)
778 785 usleep(src->delay * 1000);
779 786 }
```

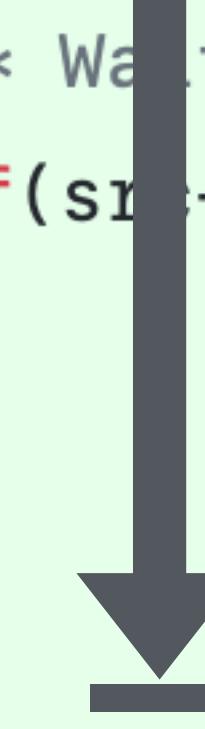


cue

```
1 cam = IO.popen("./fswebcam -D -1 foo.png", "w")
2 # wait signal
3 cam.close
```



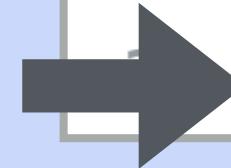
```
774 + /* Wait for a cue from stdin. */
775 + if(src->delay == 0xffffffff)
776 +
777 +
778 +
779 +
780 +
774 781 /* Delay to let the image settle */
775 - if(src->delay)
782 + else if(src->delay)
776 783 {
777 784 MSG("Delaying %i seconds.\n", src->delay)
778 785 usleep(src->delay * 1000);
779 786 }
```





cue

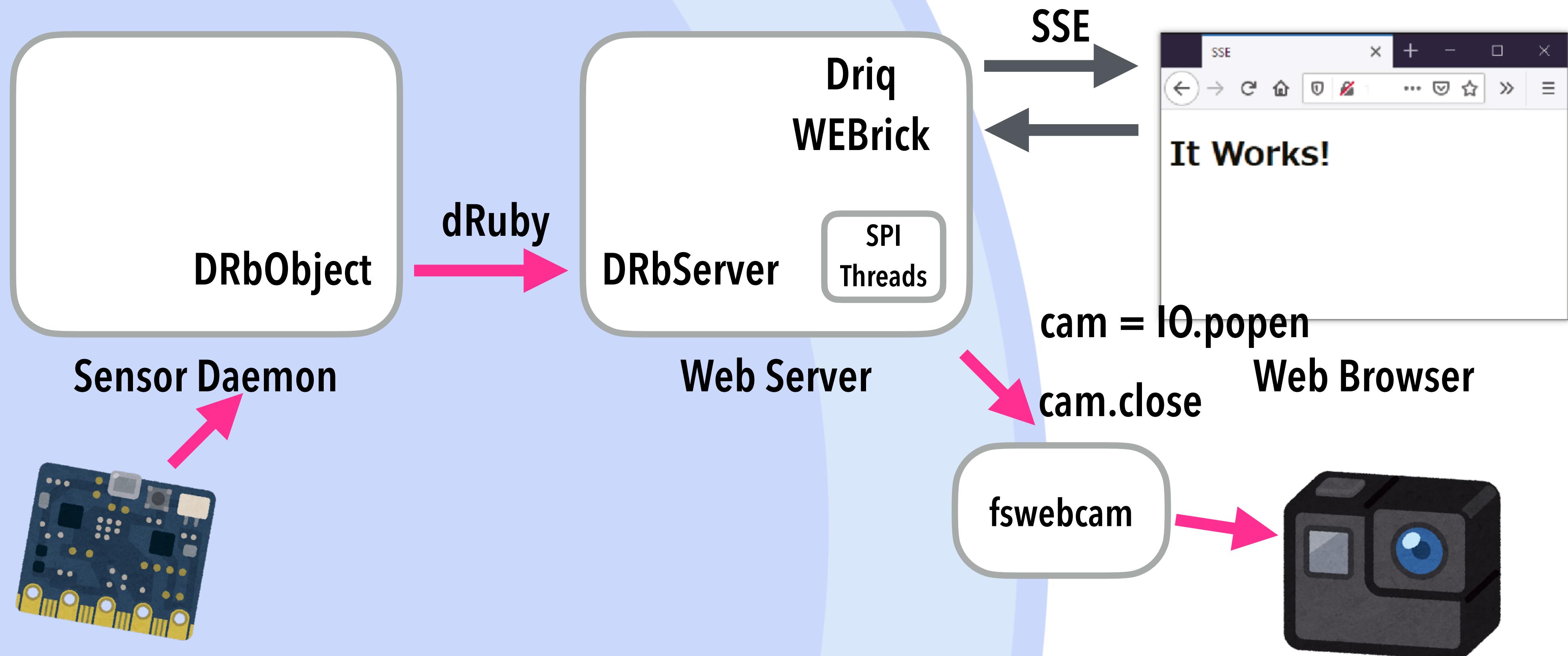
```
1 cam = IO.popen("./fswebcam -D -1 foo.png", "w")
2 # wait signal
3 cam.close
```



```
774 +     /* Wait for a cue from stdin. */
775 +     if(src->delay == 0xffffffff)
776 +     {
777 +         MSG("Waiting for a cue fr
778 +         char buf;
779 +         read(0, &buf, 1);
780 +     }
774 781     /* Delay to let the image settle
775 -     if(src->delay)
782 +     else if(src->delay)
776 783     {
777 784         MSG("Delaying %i seconds.
778 785         usleep(src->delay * 1000
779 786     }
```

fixed

- pipe.close when the signal





開発日記

- first contact with ruby
- WebUI implementation
- WiringPi gem does not work
- webcam response time

Made For life

キヤノンメディカルシステムズ株式会社