
WORK
&CO

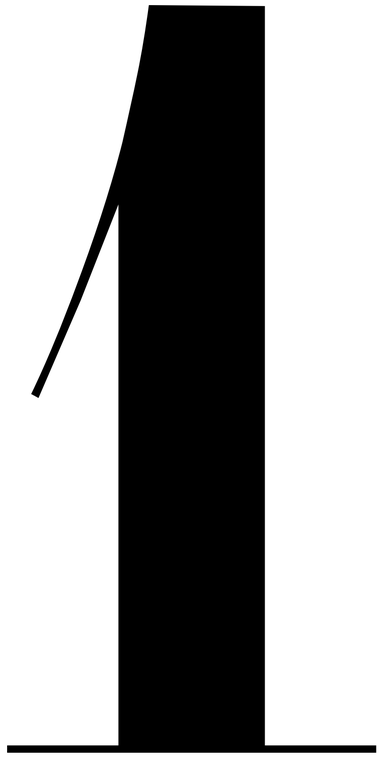
GO FAST WITH

WebWorkers

WORK & CO

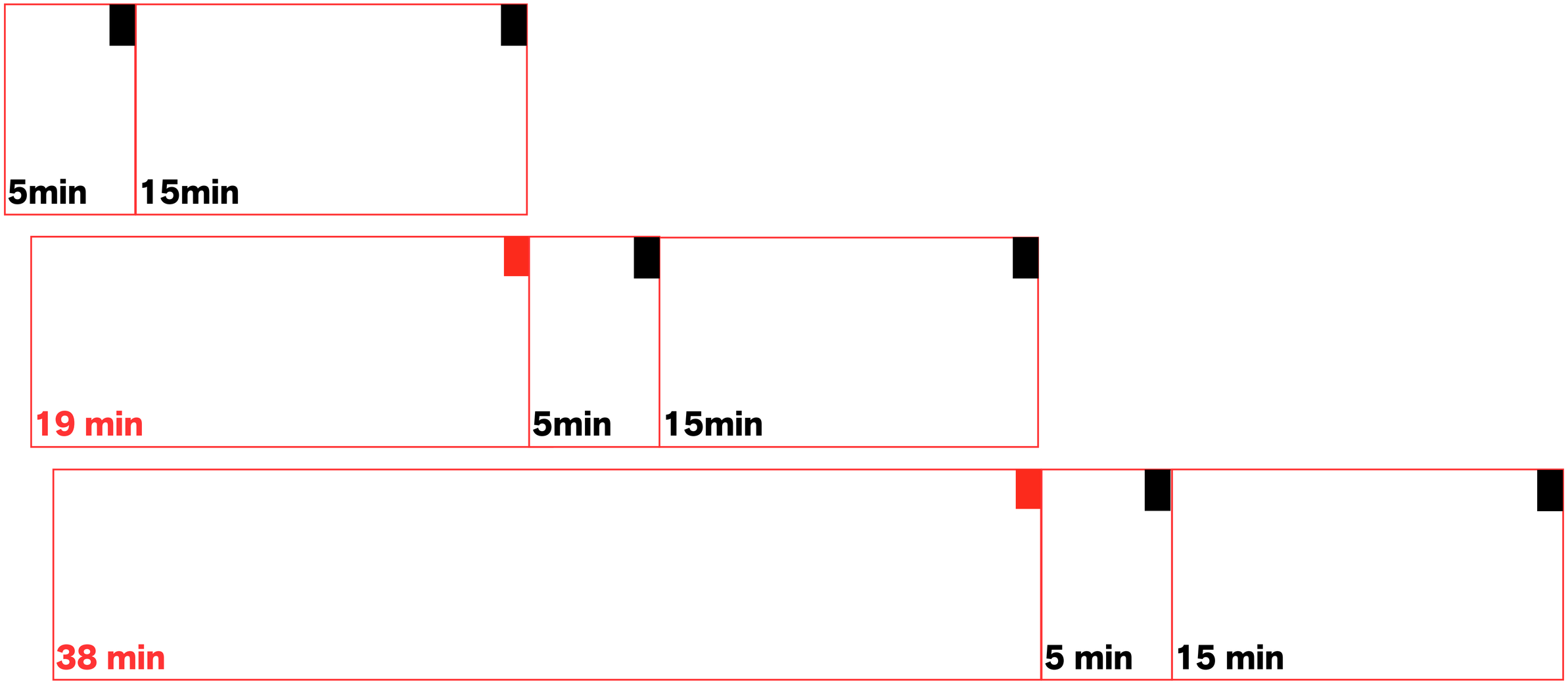
Ivan Nikitovic

Senior Software Developer

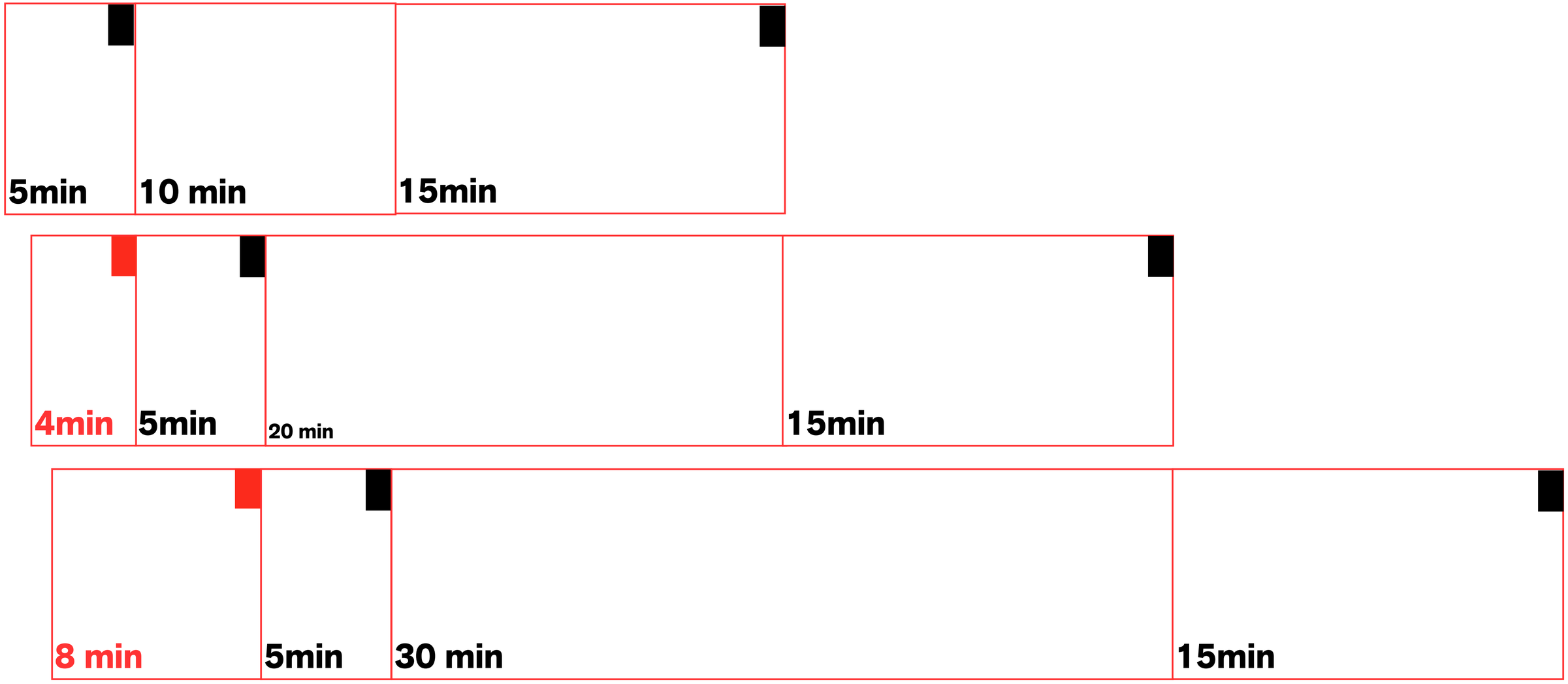


REAL WORLD EXAMPLE

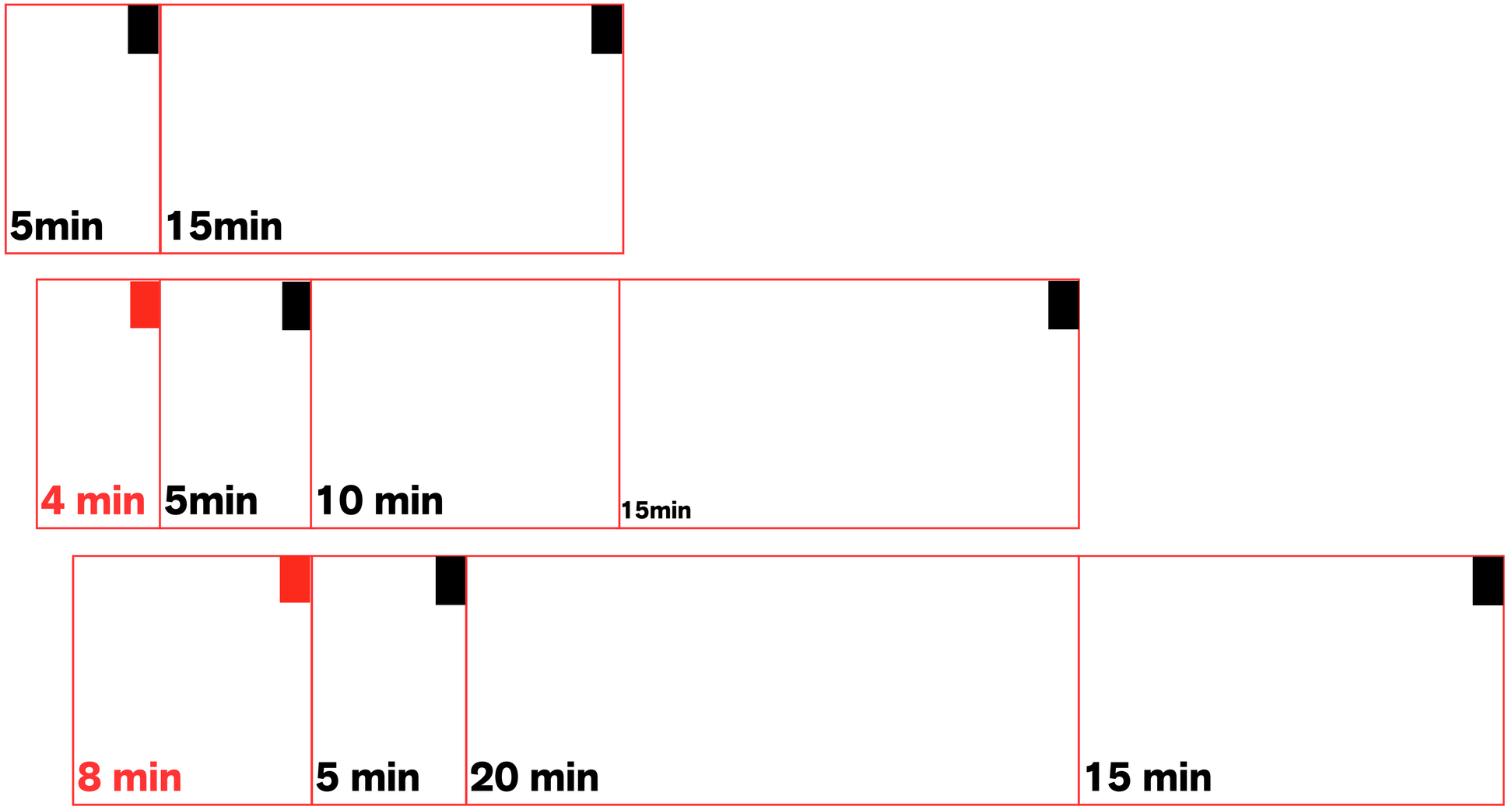
1 WAITER ALONE
SERIAL EXECUTION SCENARIO


5min
15min
19 min
5min
15min
38 min
5 min
15 min
PROCESSING
NO RESPONSE

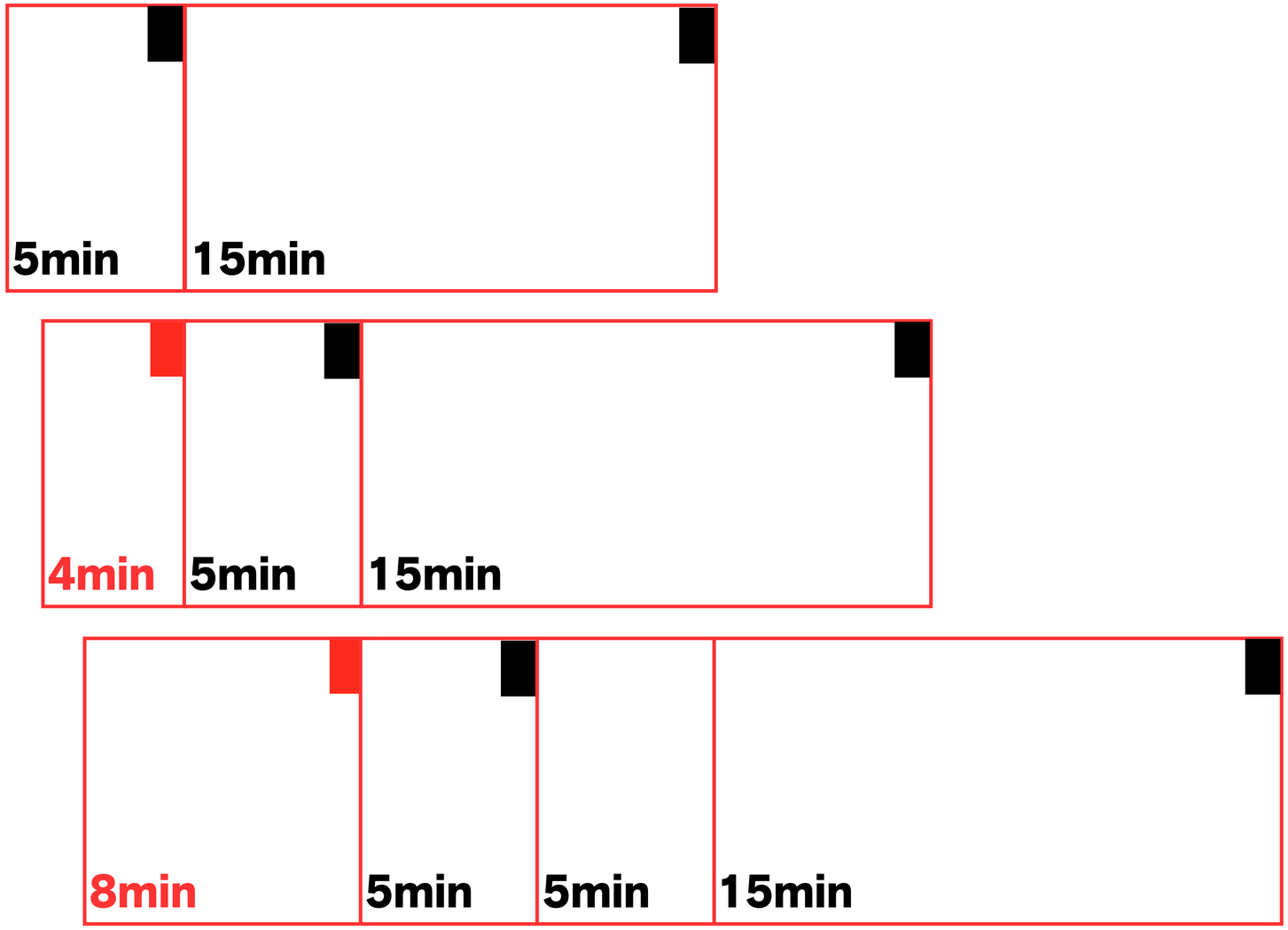
1 WAITER ALONE
ASYNCHRONOUS EXECUTION
SCENARIO


5min
10 min
15min
4min
5min
20 min
15min
8 min
5min
30 min
15min
PROCESSING
NO RESPONSE

1 WAITER 1 COOK
PARALLEL EXECUTION SCENARIO


5min
15min
4 min
5min
10 min
15min
8 min
5 min
20 min
15 min
PROCESSING
NO RESPONSE

1 WAITER 2 COOKS
HIGHLY PARALLEL EXECUTION
SCENARIO



PROCESSING
NO RESPONSE

ASYNCHRONOUS EXECUTION SCENARIO

Javascript

PARALLEL EXECUTION SCENARIO
Web Worker

CUSTOMERS' WAIT TIME WITHOUT RESPONSE

19 MINUTES

SERIAL EXECUTION SCENARIO

4 MINUTES

ASYNCHRONOUS EXECUTION SCENARIO

4 MINUTES

PARALLEL EXECUTION SCENARIO

4 MINUTES

HIGHLY PARALLEL EXECUTION SCENARIO

CUSTOMERS' WAIT TIME PROCESSING

39 MINUTES

SERIAL EXECUTION SCENARIO

44 MINUTES

ASYNCHRONOUS EXECUTION SCENARIO

34 MINUTES

PARALLEL EXECUTION SCENARIO

26 MINUTES

HIGHLY PARALLEL EXECUTION SCENARIO

WEB WORKER DEFINITION

"A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page."

— W3SCHOOLS

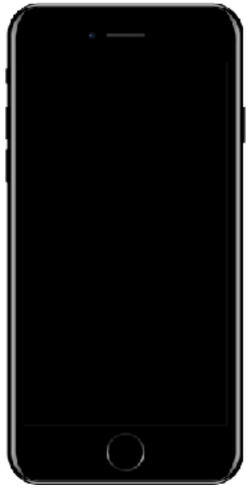
"Web Workers provide a simple means for web content to run scripts in background threads."

— MDN

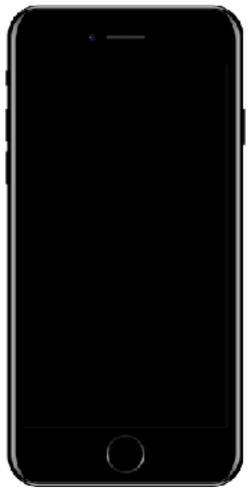
3

USAGE OF CLIENT RESOURCES

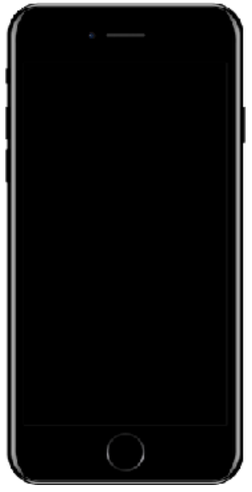
CURRENT STATE OF TECHNOLOGY

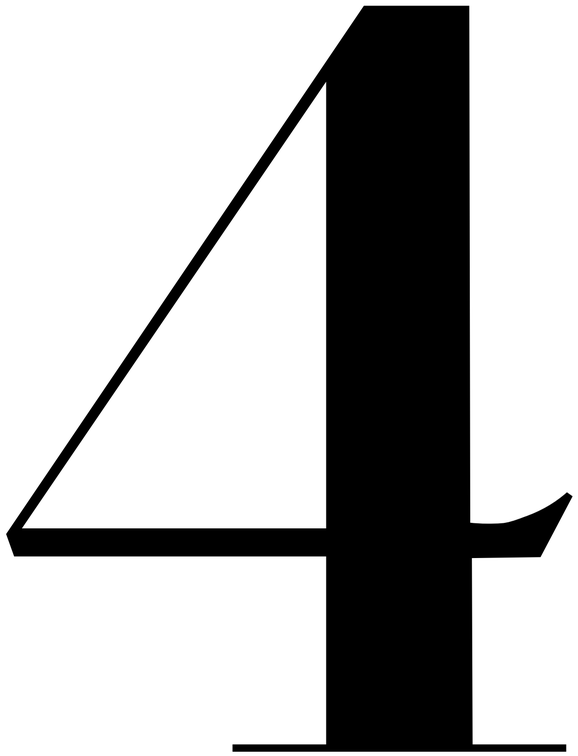


RESOURCE UTILIZATION



UNUSED RESOURCES





WEB WORKER TYPES

DEDICATED WORKER

SHARED WORKER

SERVICE WORKER

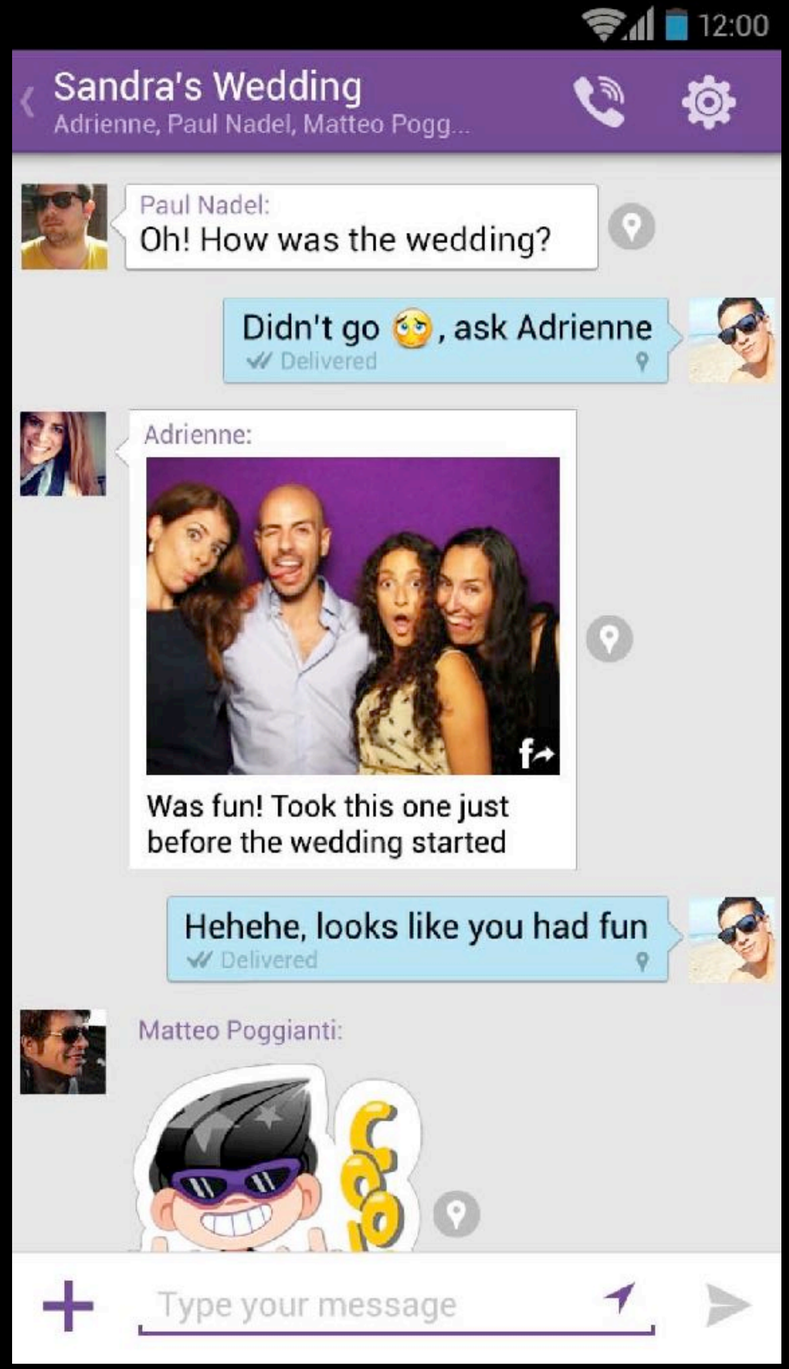
5

HOW TO COMMUNICATE TO WEB WORKER

MESSAGE PASSING

THE STRUCTURED CLONE ALGORITHM

The structured clone algorithm is a new algorithm defined by the HTML5 specification for serializing complex JavaScript objects. It's more capable than JSON in that it supports the serialization of objects that contain cyclic graphs — objects can refer to objects that refer to other objects in the same graph. In addition, in some cases, the structured clone algorithm may be more efficient than JSON.



TRANSFERABLE INTERFACE

The Transferable interface represents an object that can be transferred between different execution contexts, like the main thread and Web workers.

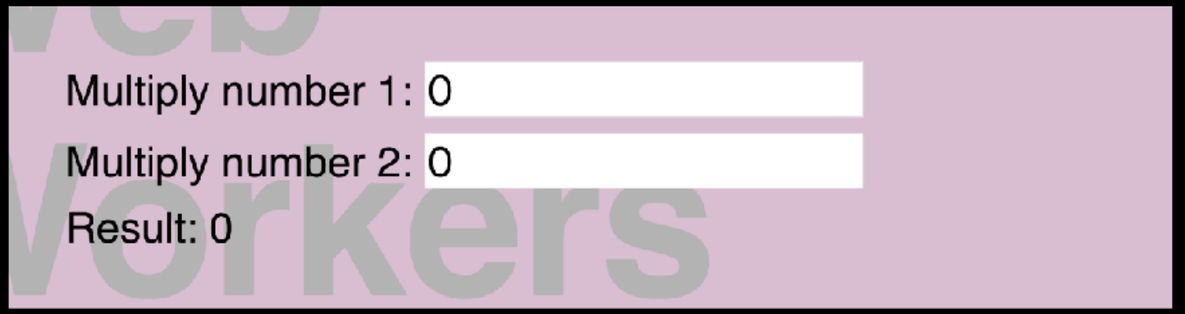
6

SIMPLE WEB WORKER EXAMPLE

```
var myWorker = new Worker('worker.js');
```

```
first.onchange = function() {
  myWorker.postMessage([first.value,second.value]);
  console.log('Message posted to worker');
}

second.onchange = function() {
  myWorker.postMessage([first.value,second.value]);
  console.log('Message posted to worker');
}
```




```
onmessage = function(e) {
  console.log('Message received from main script');
  var workerResult = 'Result: ' + (e.data[0] * e.data[1]);
  console.log('Posting message back to main script');
  postMessage(workerResult);
}
```

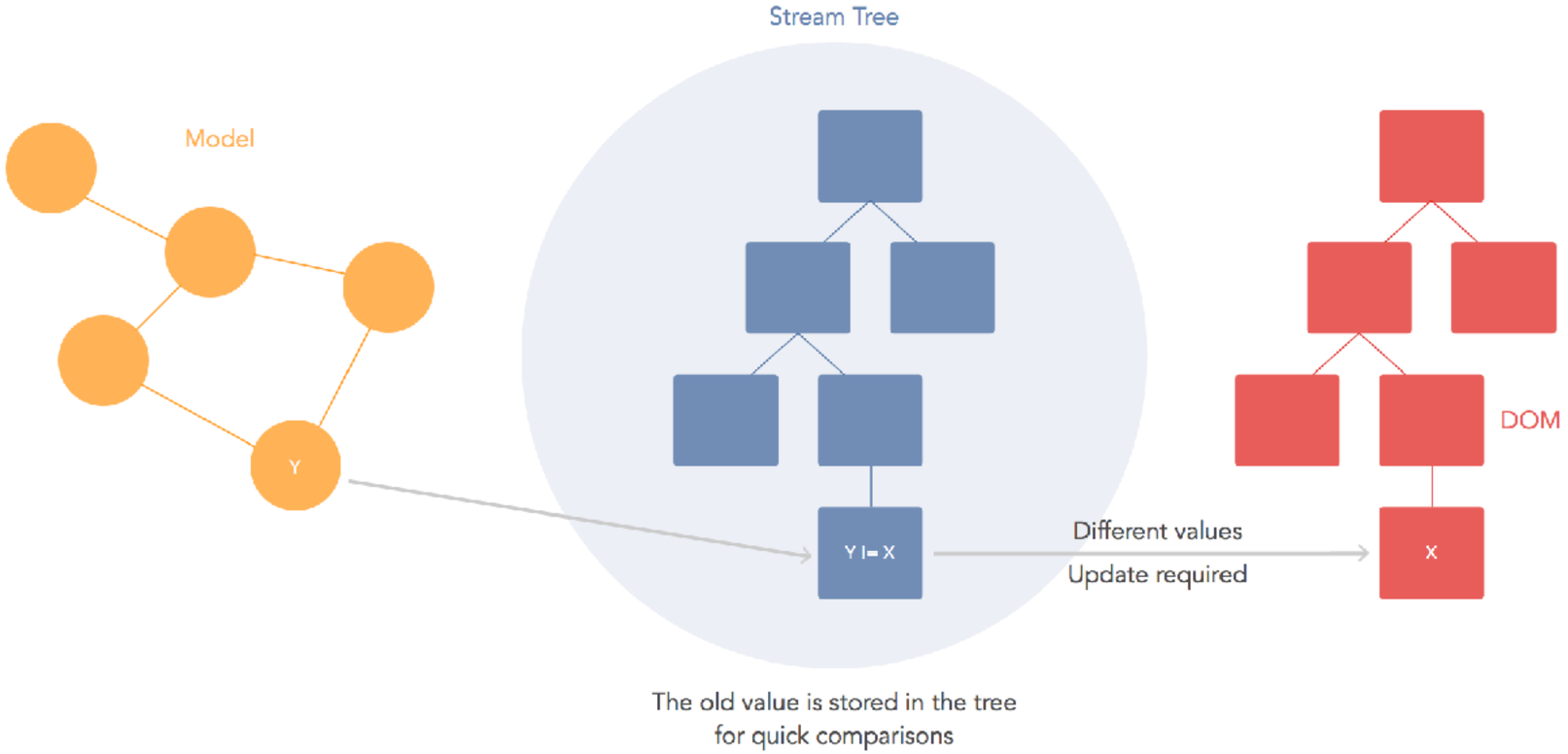
```
myWorker.onmessage = function(e) {
  result.textContent = e.data;
  console.log('Message received from worker');
}
```

```
myWorker.terminate();
```

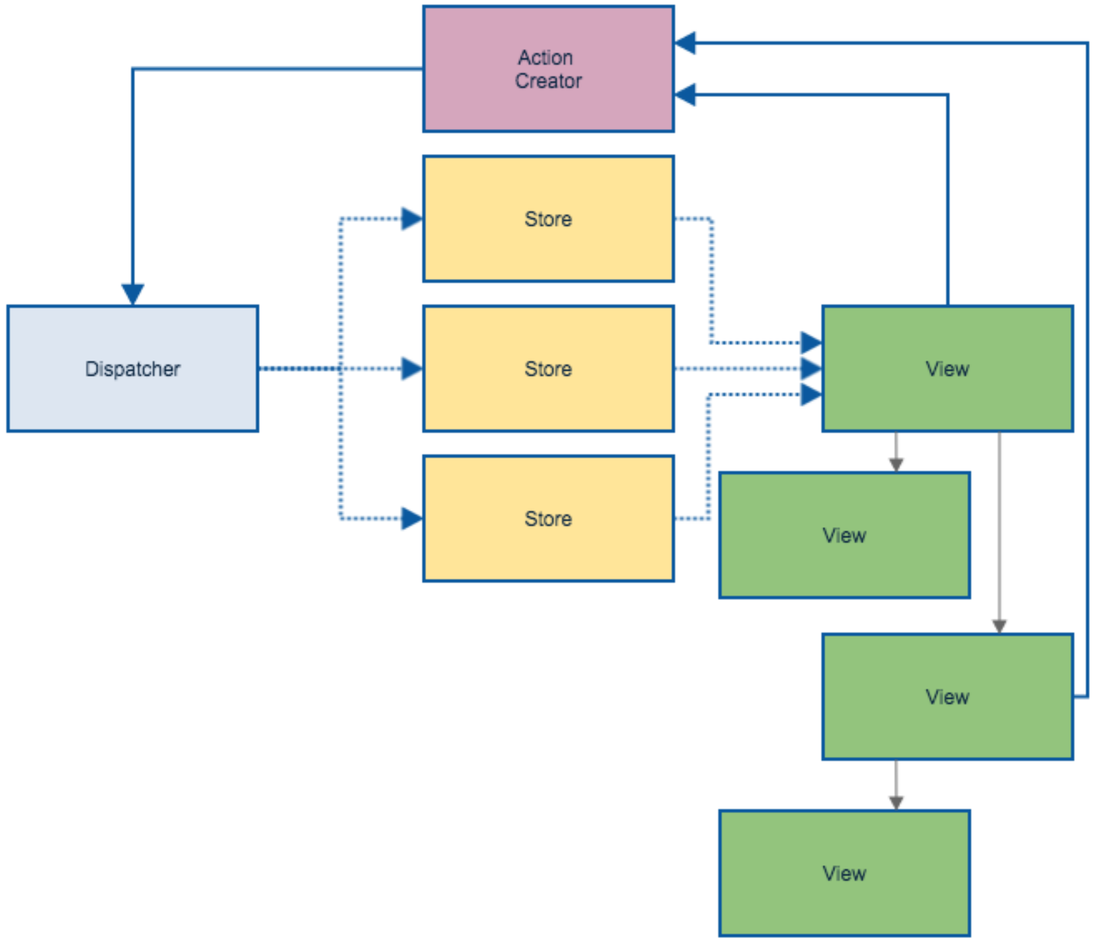
```
close();
```

7

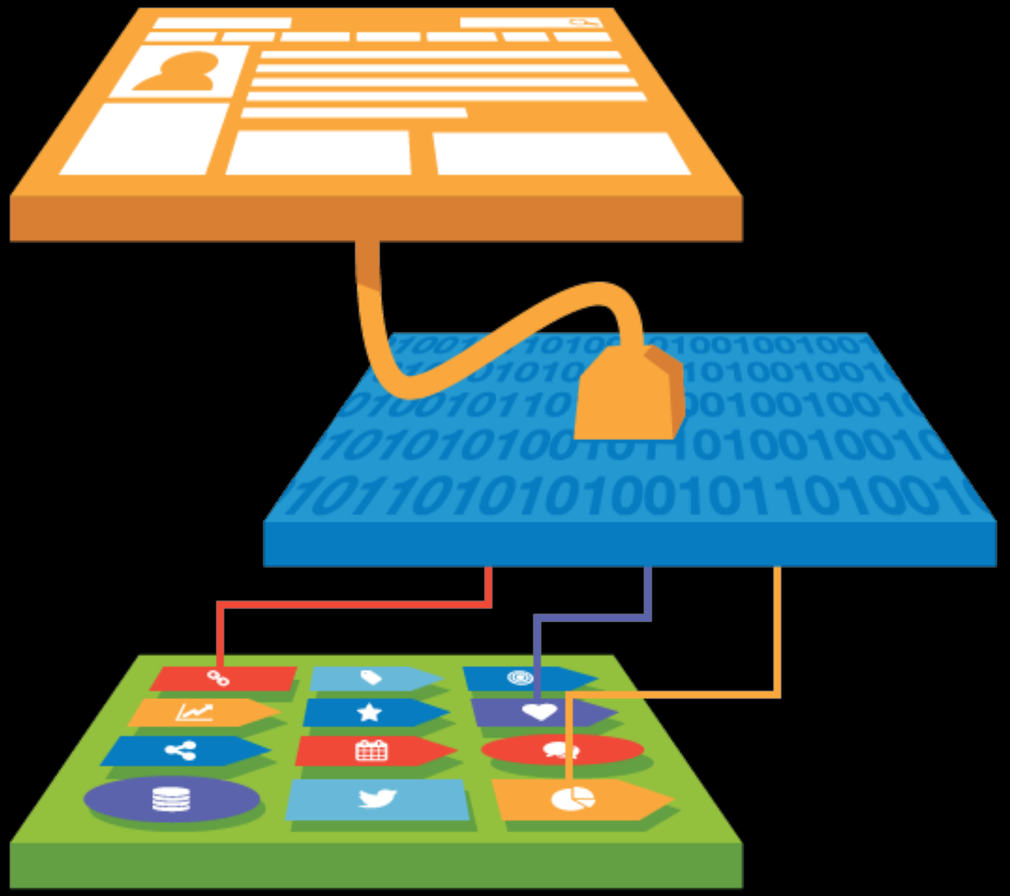
UNREAL WORLD EXAMPLES



VIRTUAL DOM INSIDE WEB WORKER



FLUX INSIDE
WEB WORKER

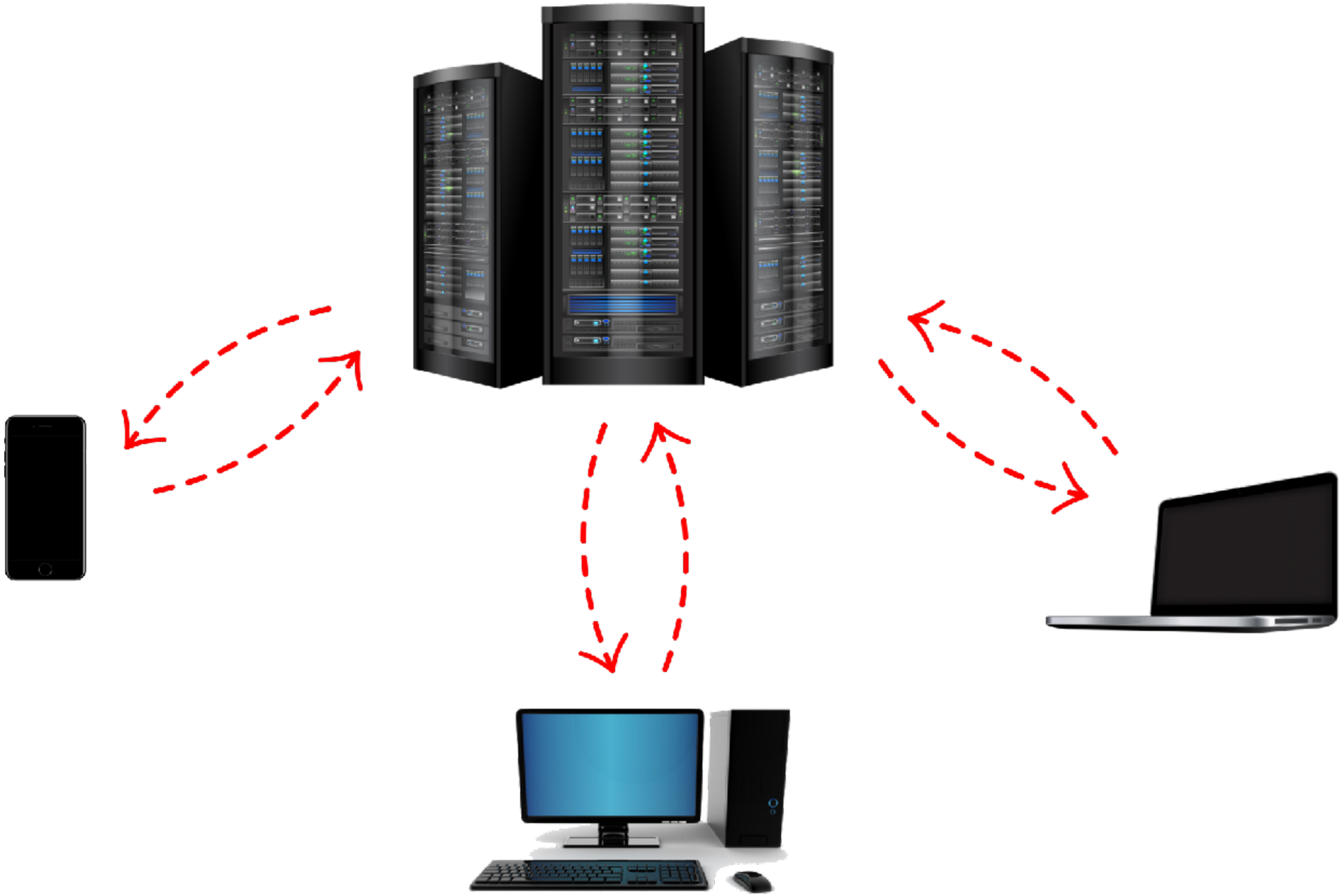


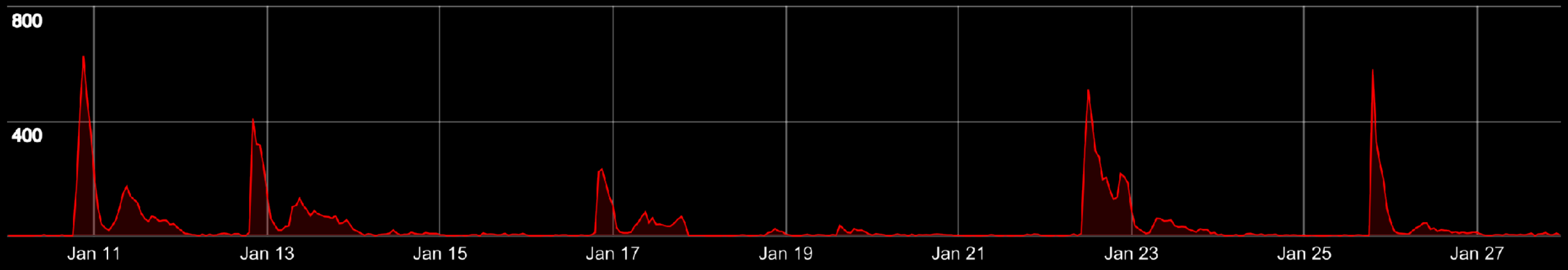
DATALAYER INISIDE WEB WORKER

8

APPLICATION SCALING

HOW USING MORE CLIENT RESOURCES IMPACTS OVERALL INFRASTRUCTURE





USERS PER HOUR CHART

COST EFFECTIVE ARCHITECTURE

LOAD RESILIENT ARCHITECTURE

Q&A

www.work.co