UNIVERSITÉ Grenoble Alpes

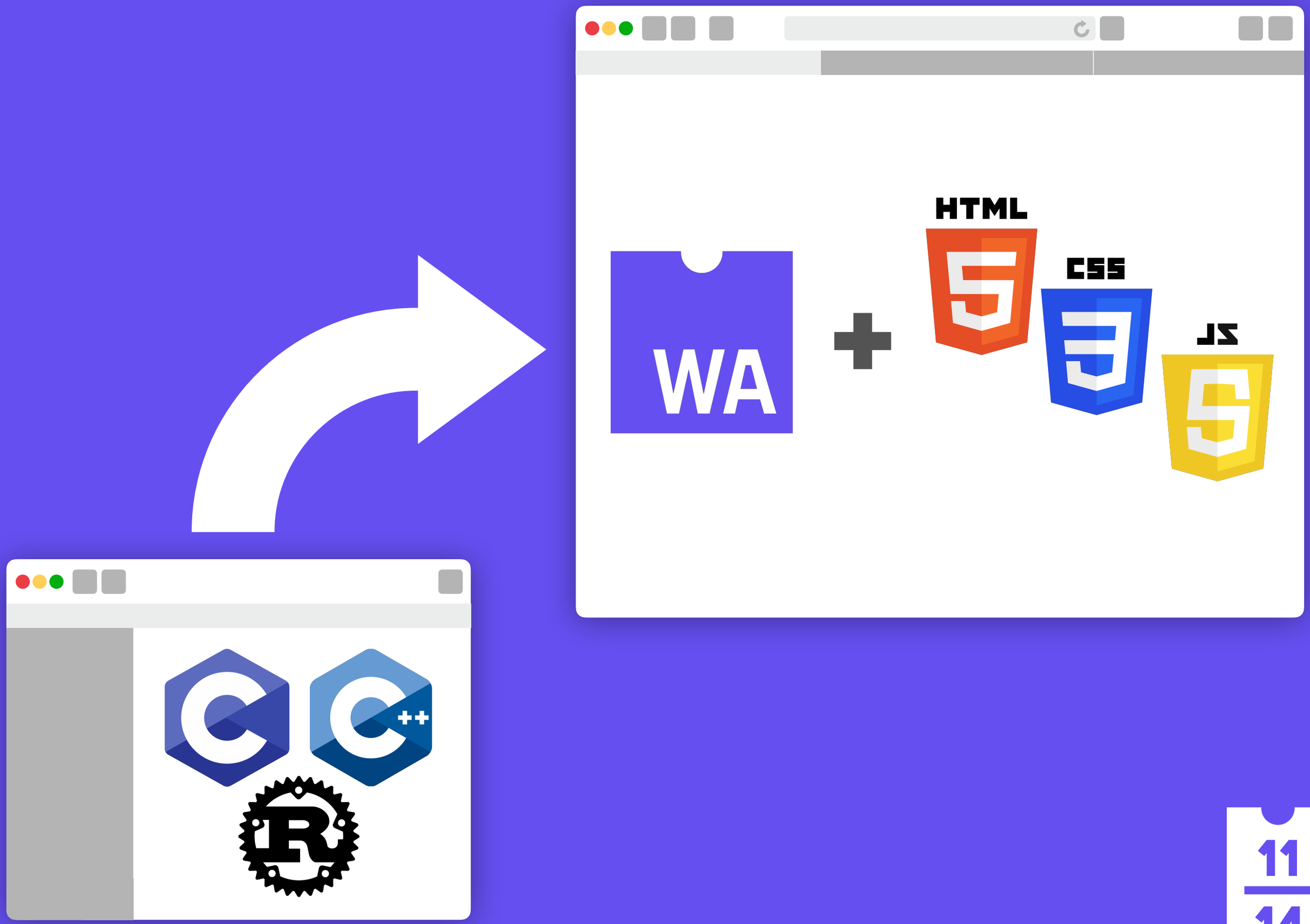
POLYTECH GRENOBLE

WA

WEBASSEMBLY

PAR QUENTIN FOMBARON

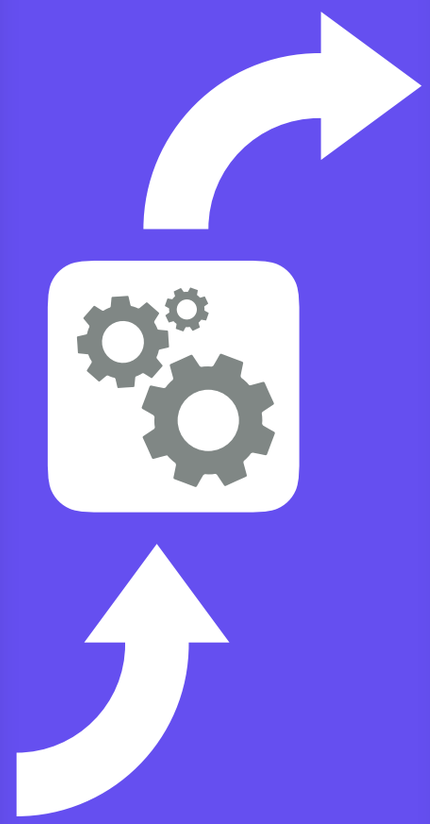
HOW?


WA
HTML
CSS
JS
C
C++
R

ASSEMBLEUR

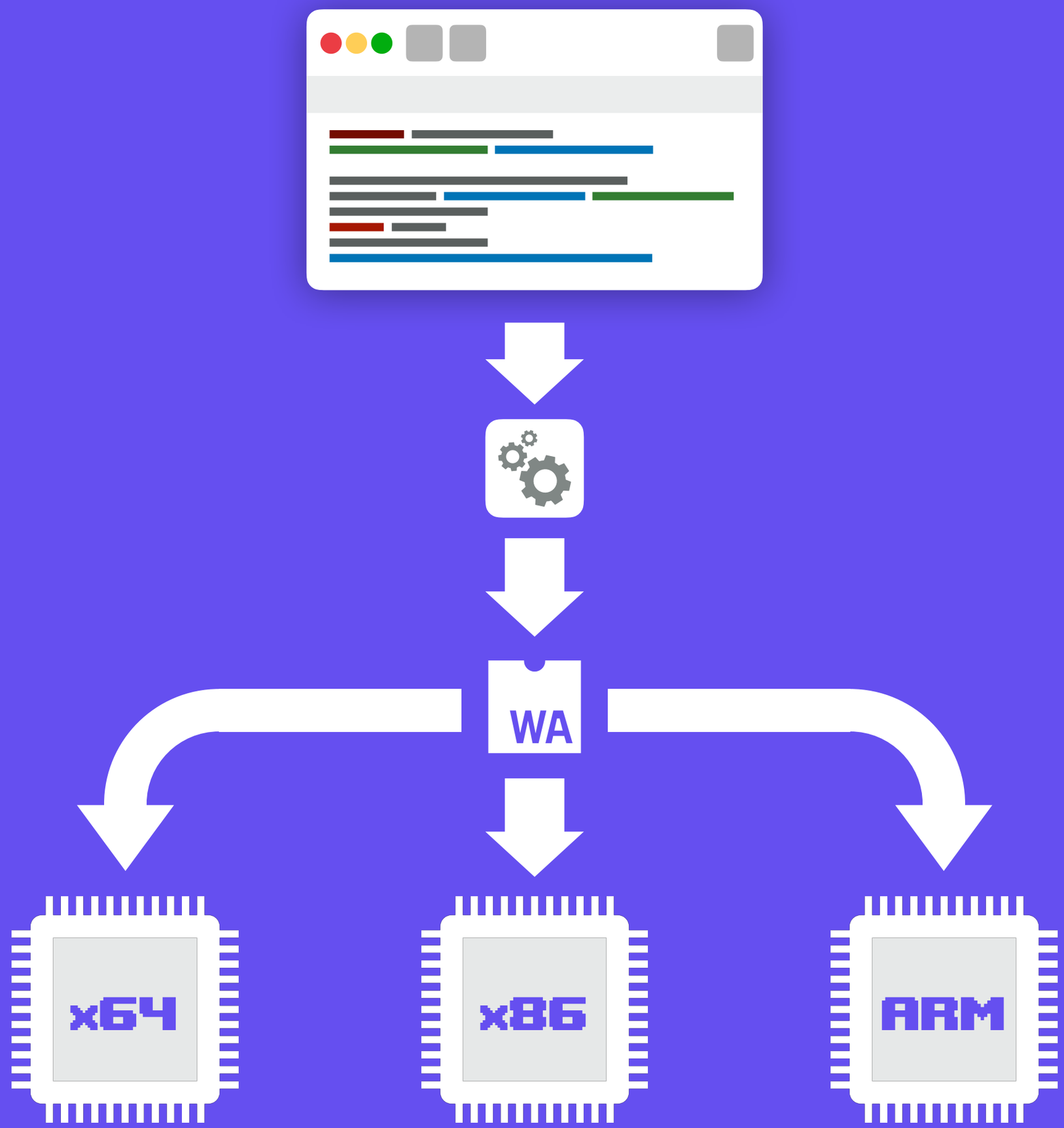
```
MOV R0, #10
MOV R1, #3
ADD R0, R0, R1

MOV R0, #0x18
LDR R1, =0x26
SVC #0x008
END
```



CODE MACHINE

```
0001 1100 1001
1000 1110 0101
1100 0011 0001
1010 1111 0111
0011 1011 1110
0010 1101 0001
1101 1010 1001
1110 0100 0011
```


WA
x64
x86
ARM

FORMAT TEXTUEL

.wat

```
(module
  (func $addTwo (param i32 i32)
    (result i32)
    (i32.add
      (get_local 0)
      (get_local 1))
  (export "addTwo" $addTwo))
```

=

FORMAT BINAIRE

.wasm

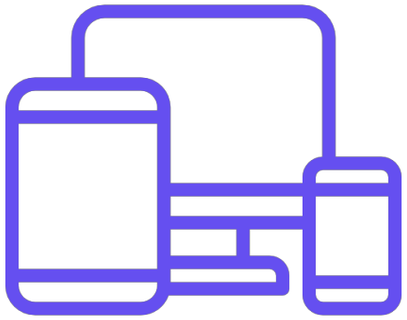
```
48 83 EC 08
8B CF
8B C1
03 C6
66 90
48 83 C4 08
C3
```

WA
HTML
CSS

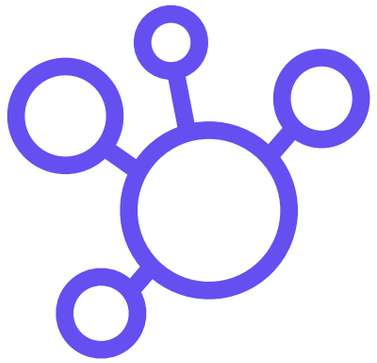
WHY?

WHAT?

RAPIDITÉ



PORTABILITÉ

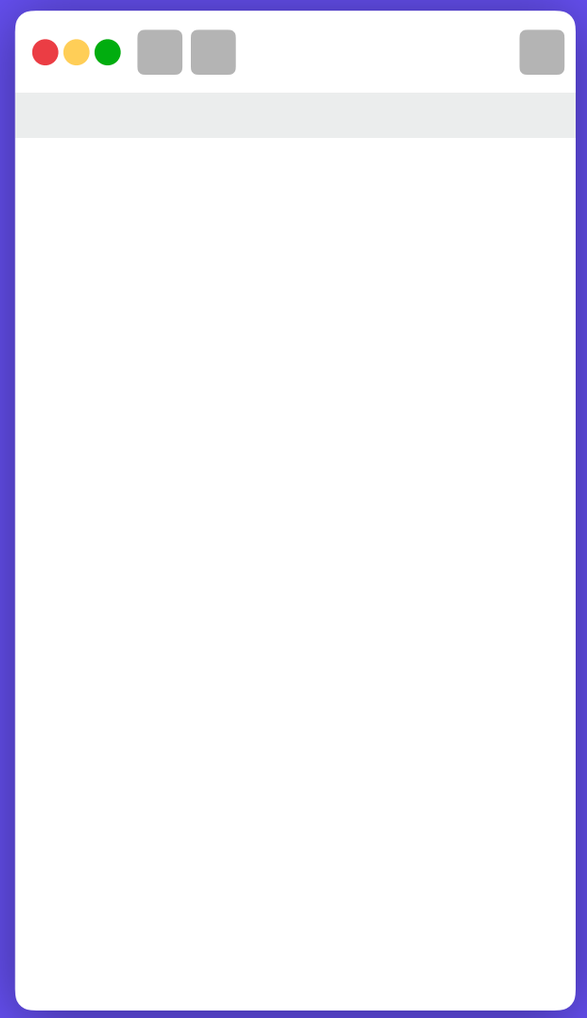


FLEXIBILITÉ

RAPIDITÉ

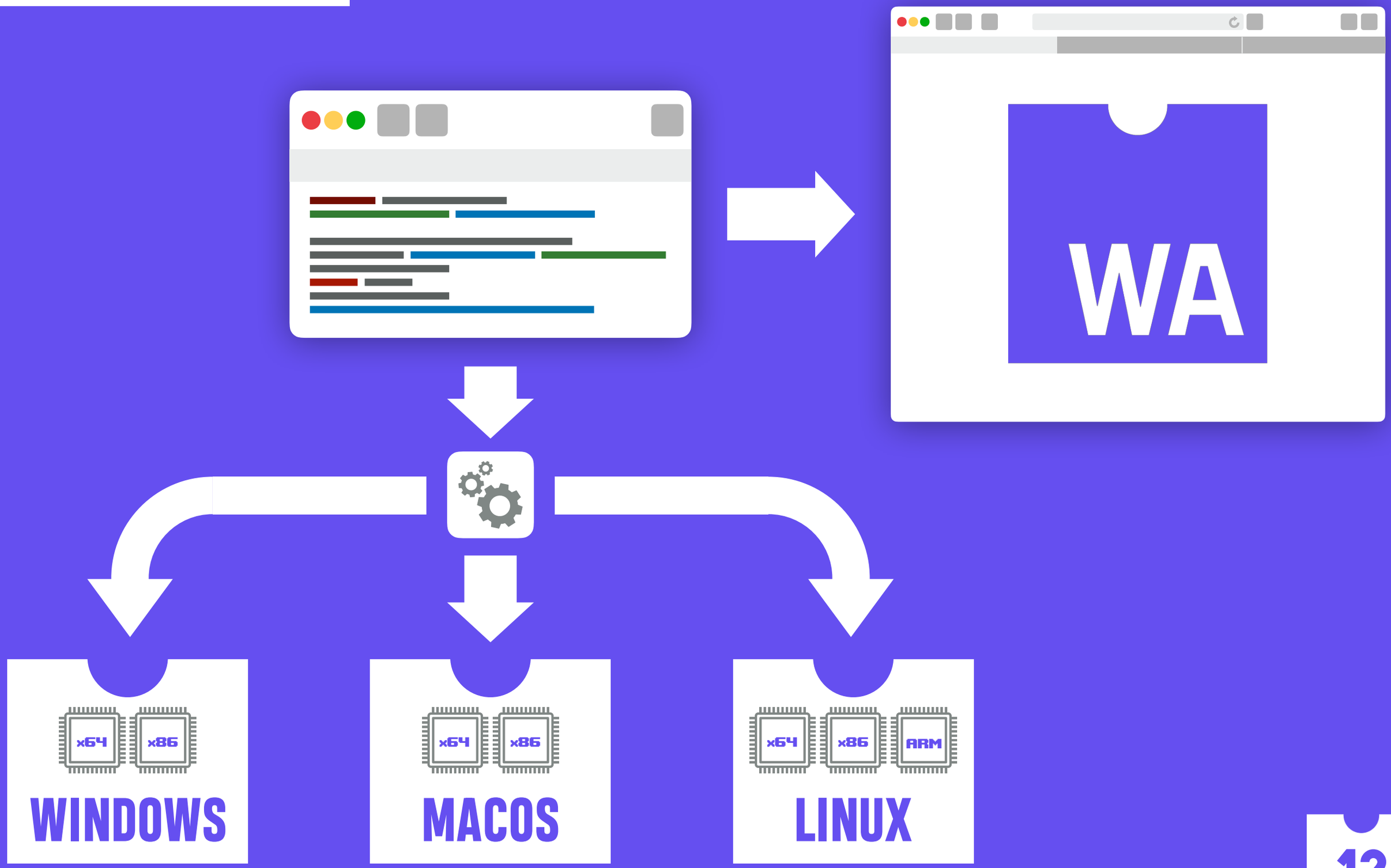
PARSER, COMPILER ET OPTIMISER

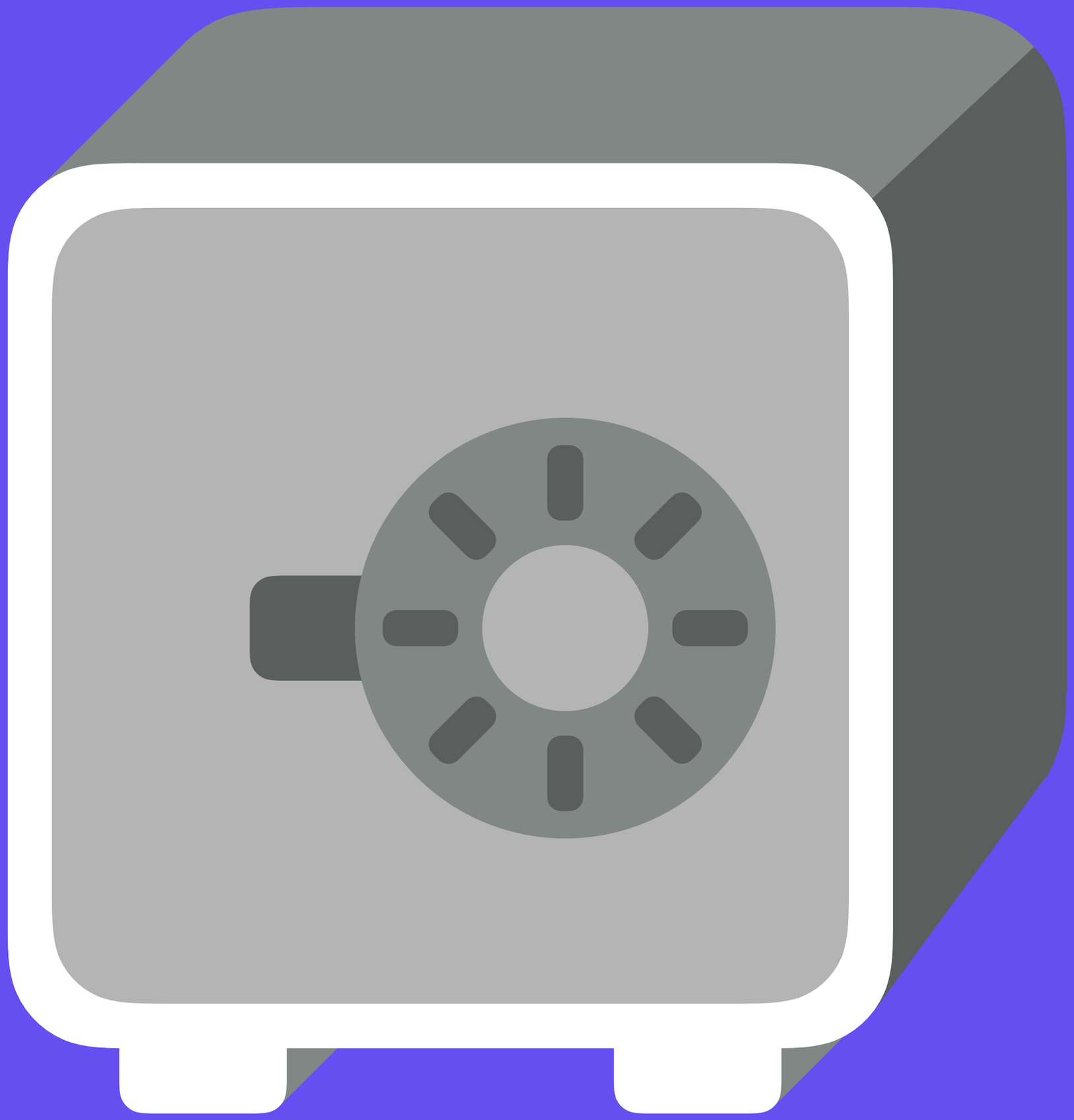
DÉCODER ET COMPILER

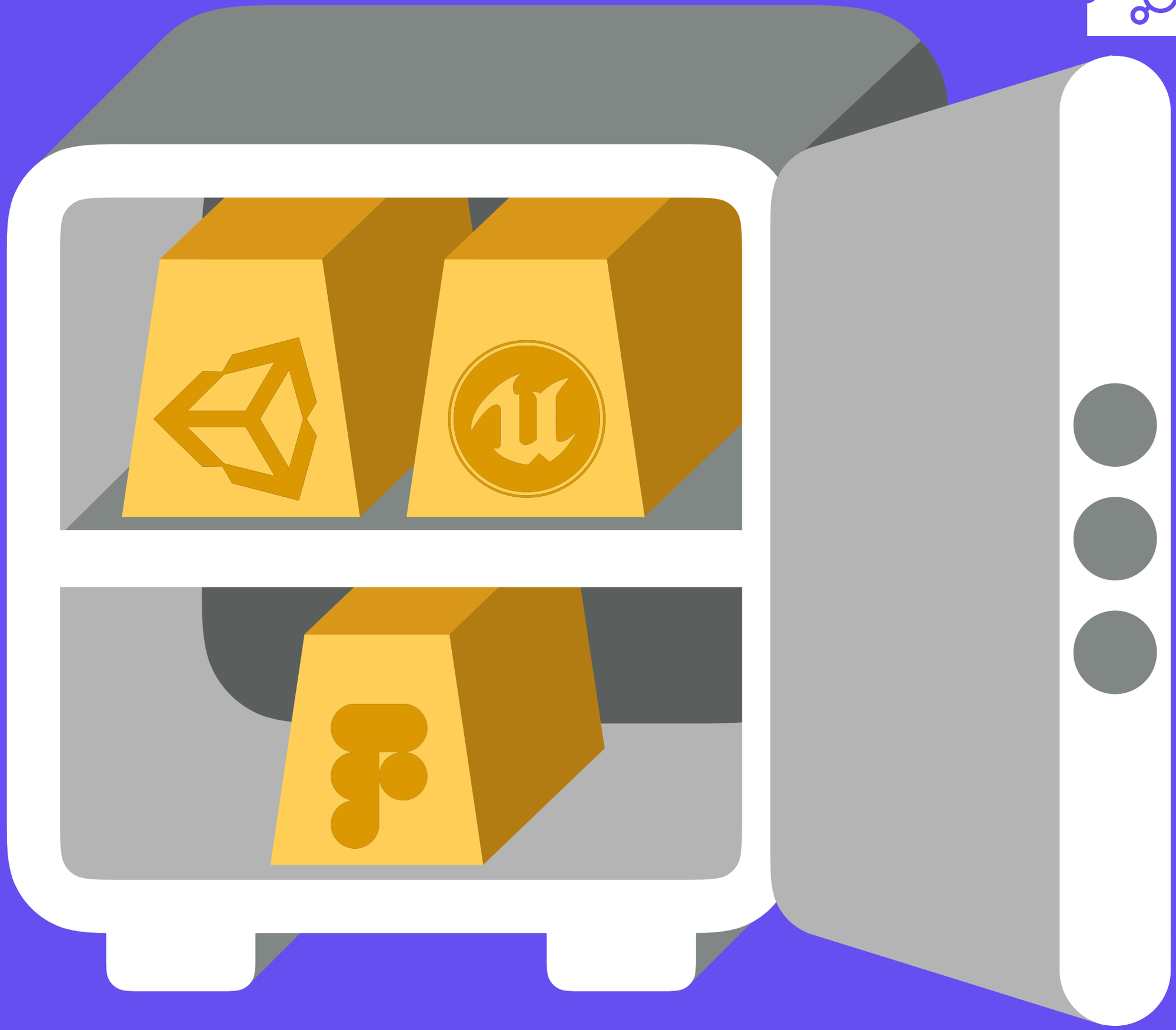


WA

PORTABILITÉ







CONCLUSION

DÉMONSTRATION

```
$> emcc hello.c -s WASM=1-o hello.html
```

hello.wasm hello.html

+

hello.js

SOURCES

https://blog.logrocket.com/webassembly-how-and-why-559b7f96cd71

https://webassembly.org

https://www.youtube.com/watch?v=5XtA3k07jEQ

https://mbebenita.github.io/WasmExplorer/

https://webassembly.studio

https://takahirox.github.io/WebAssembly-benchmark/

MERCI
DE VOTRE ÉCOUTE

PLACE AUX QUESTIONS ET REMARQUES

2020