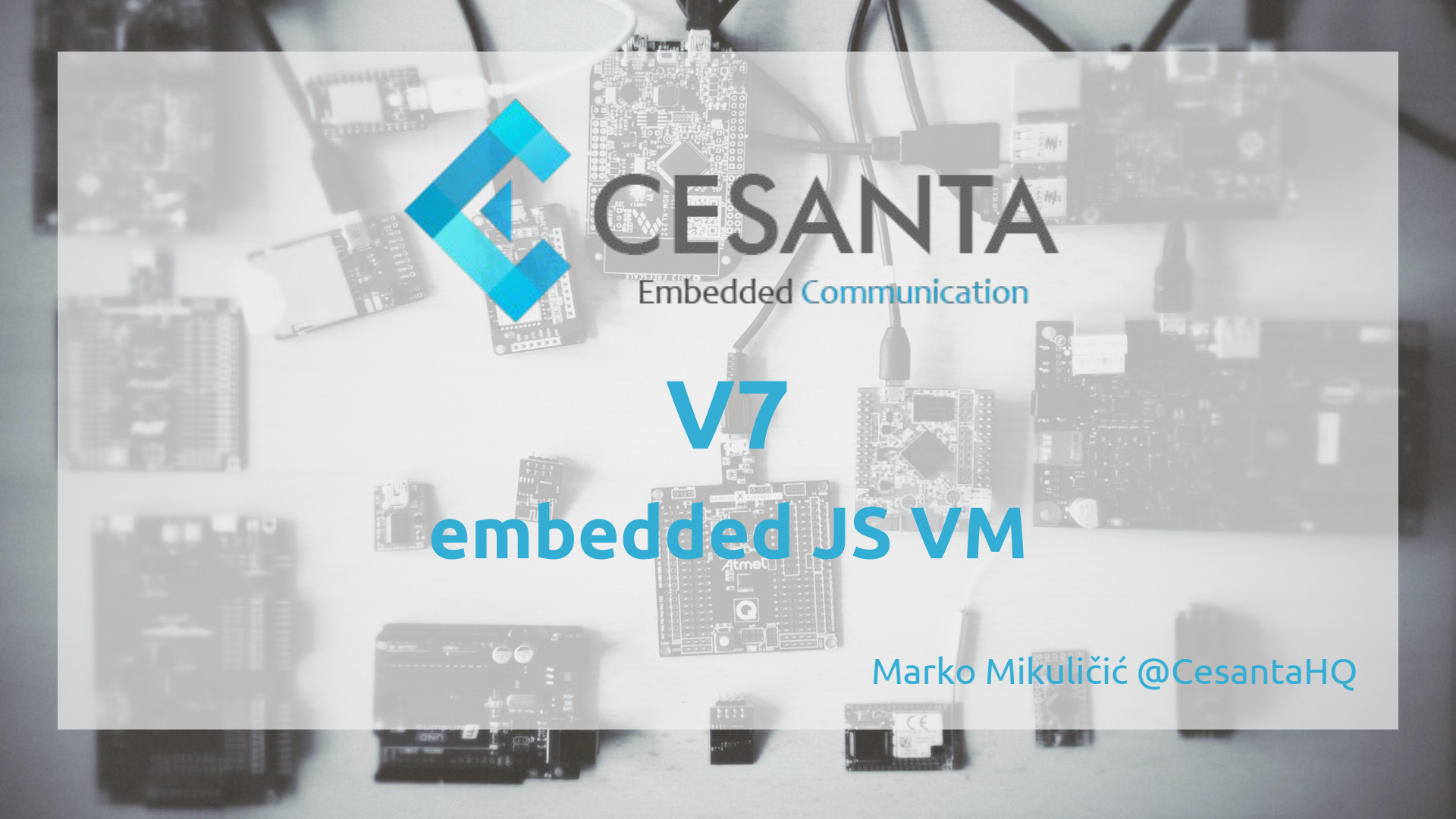

CESANTA
Embedded Communication
V7
embedded JS VM
Marko Mikuličić @CesantaHQ

Who is Cesanta?

- Irish Startup founded by ex-Googlers in 2013
- Makers of the Mongoose embedded web server, over 1M downloads since 2004
- Developers of Smart.js - full-stack, open source IoT platform
- Goal: make it easy for everyone to make connected devices

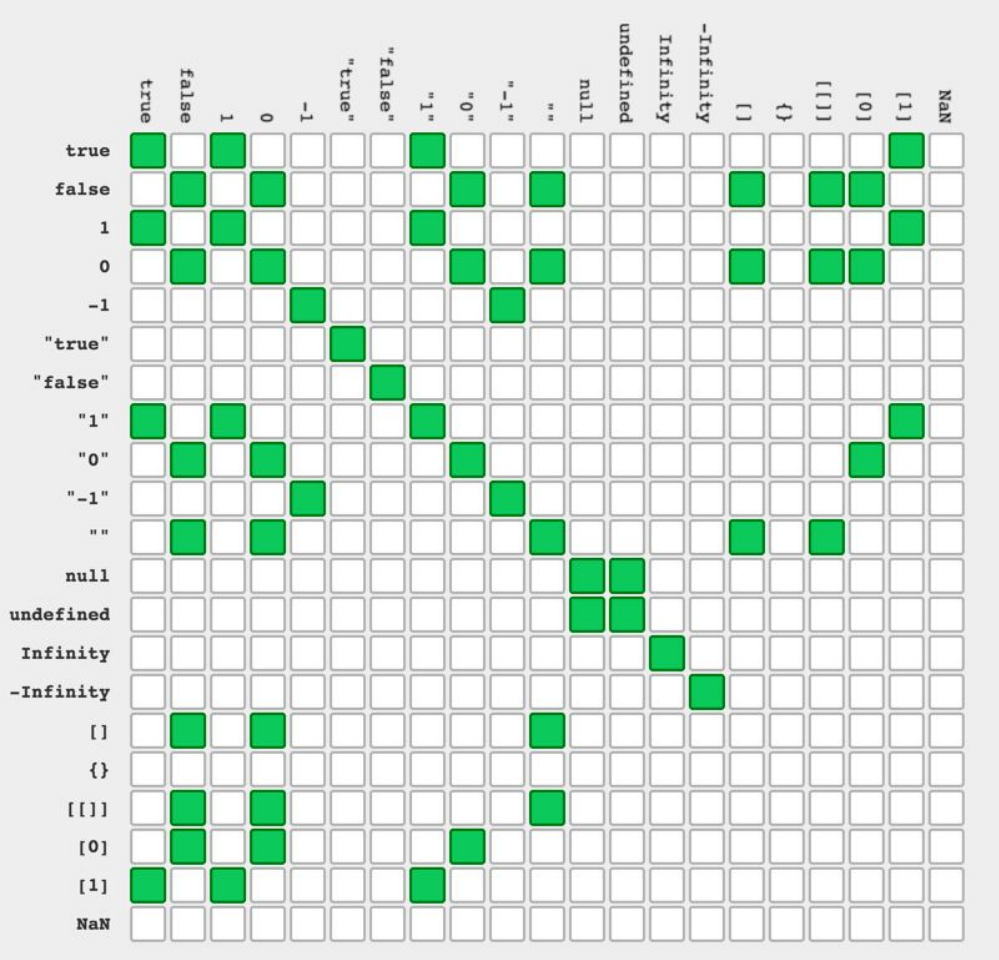
What is V7?

- Embedded JavaScript engine written in C
- Compliant to ISO C90 and ISO C++98
- Targets embedded systems
- Or C/C++ programs that need scripting engine
- Conforms to ECMA 5.1 standard
- Easy to embed: only two files, **v7.c** and **v7.h**
- Easy to use embedding API

Here’s how to try out V7

- Open Source, under GPLv2/commercial license
- GitHub repo: https://github.com/cesanta/v7/

Why JavaScript?



Source: https://dorey.github.io/JavaScript-Equality-Table/


EVERYTHING
IS TERRIBLE
memegenerator.net

Why JavaScript?

- Why not Lua, Python, LISP, brainfeck, ...
- Well, why scripting ?

Why JavaScript?

- Everybody (thinks) to know (a bit of) it
- Popular, Widely available
- Maturing but not stagnating

Goals: V7 vs other JS VMs

- Portable
- Small runtime overhead
- Small code footprint
- Reasonably fast
- Simple

Techniques

- Implicit data structures
- Compacting GC
- Static object graph snapshots
- mmapping compiled bcode
- Coroutine based parser (segmented stack)

Simple example

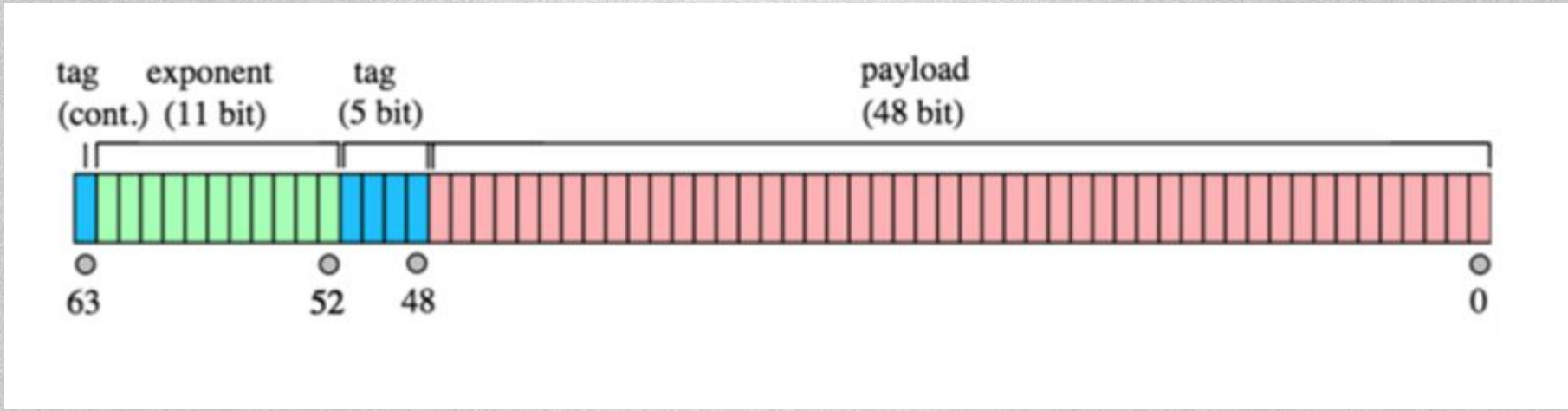
```c
#include <v7.h>

int main() {
  struct v7 *v7 = v7_create();
  v7_val_t res;

  v7_exec(v7, "function random() { return 42; }", NULL);
  v7_exec(v7, "function msg() { return 'foo'; }", NULL);


  v7_exec(v7, "random()", &res);
  printf("random number (really): %lg\n", v7_to_number(res));


  v7_exec(v7, "msg()", &res);
  printf("msg: %s\n", v7_to_cstring(res));


  v7_destroy(v7);
  return 0;
}
```

What is a value? naive approach

- tagged union?
- too much waste

```
struct Value {
        union {
                int boolean;
                double number;
                String *str;
                Object *object;
                /* more */
        } u;
        char type; /* type tag */
        /* padding */
};
```

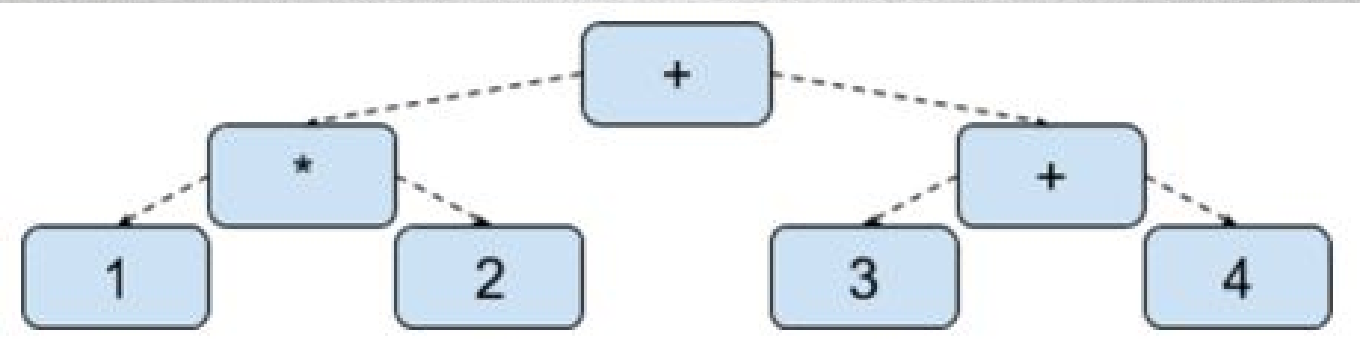
NaN packing



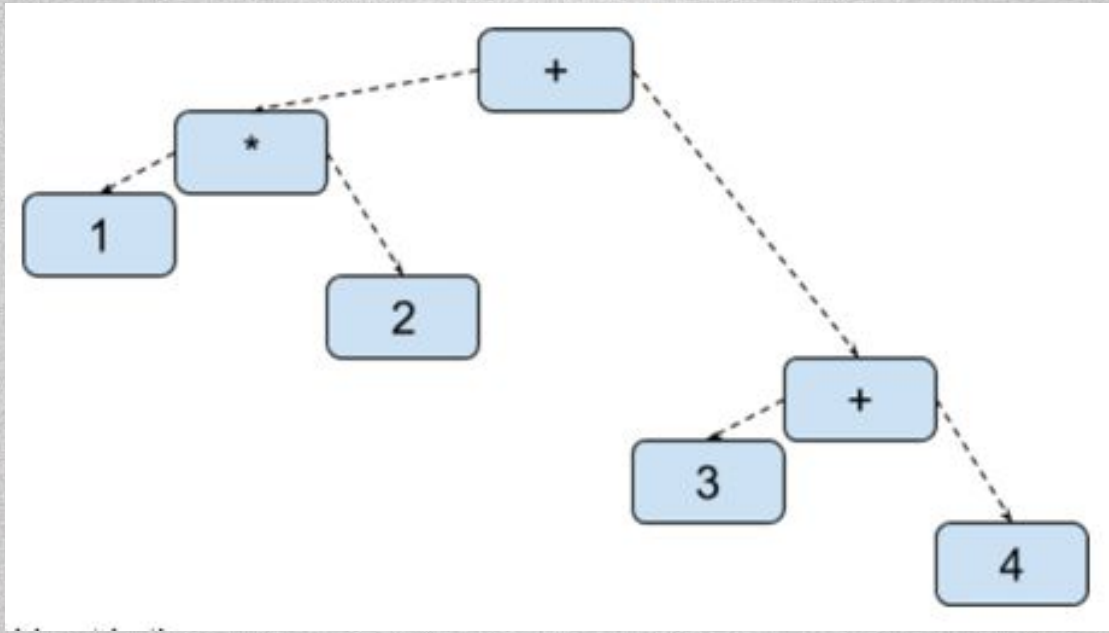
Implicit data structures

Implicit data structures are **data structures** with negligible space overhead; auxiliary information is mostly represented by permuting the elements cleverly

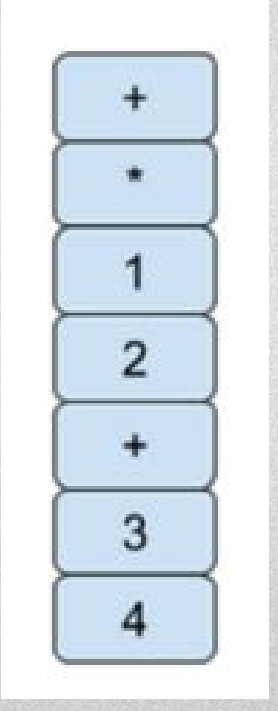
AST



AST



AST



AST - no pointers

(mostly)

Questions?