Node Addons avec Rust

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

V8 is Google’s open source high-performance JavaScript and WebAssembly engine, written in C++.

Link between Node & V8

Each version of node is linked to a V8 version

Since node version 8, node avoid the `v` (like in node v7) to dismiss confusion between node « version » and « v8 » engine

Node 8.x ==> V8 6.2.x

Node 9.x ==> V8 6.2.x

Node 10.x ==> V8 6.7.x

https://v8docs.nodesource.com/

Node.js Addons

- Are dynamically-linked shared objects, written in C++, using V8 api, that can be loaded using the `require()` function (๑>ᴗ<๑)
- Are compiled modules dependent to V8 API
  => build is dependent to system architecture, os and node version (╬ Ò ‸ Ó)
- Since Node 8, can be written with n-api, an abstraction that is independent from the underlying JavaScript runtime (V8)
  ᕦ(ò_óˇ)ᕤ … still not the « norm » (╥_╥)

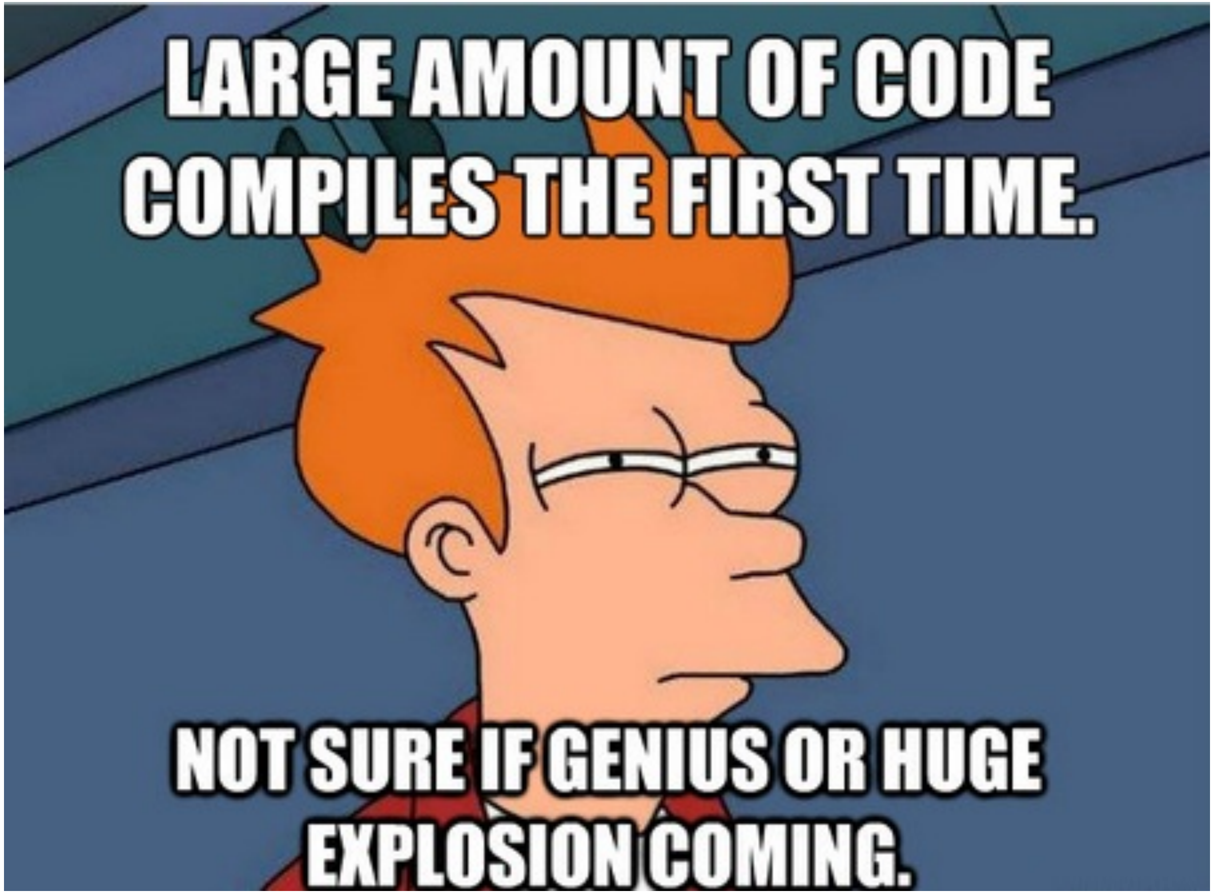
Why & When I need to write a native add-on

- Re-use existing C++ code
- Use binary libraries : static libraries (.a), shared libraries (.so / .dylib / .dll)
- Target native performance
- Get access to system ressources (I/O, serial ports, GPU, …)
- Use worker threads (multi-threading exist in Node) … at least before Node 11
- Statically type-check a library

Why & When I shouldn’t write a native add-on

- Rewrite the event-loop
- Perform extensives I/O
- Statically type-check a library … there is better alternative if it is the only motivation (ReasonML/Buckelscript, OCaml/JSoo, Kotlinjs, Typescript, …)

So I can use Javascript and C++


LARGE AMOUNT OF CODE COMPILES THE FIRST TIME.
NOT SURE IF GENIUS OR HUGE EXPLOSION COMING.

Rust & crates to the rescue

Cutii

I'll be designated driver tonight!
we know
Don't talk while eating you could choke and die haha

https://neon-bindings.com/docs/getting-started