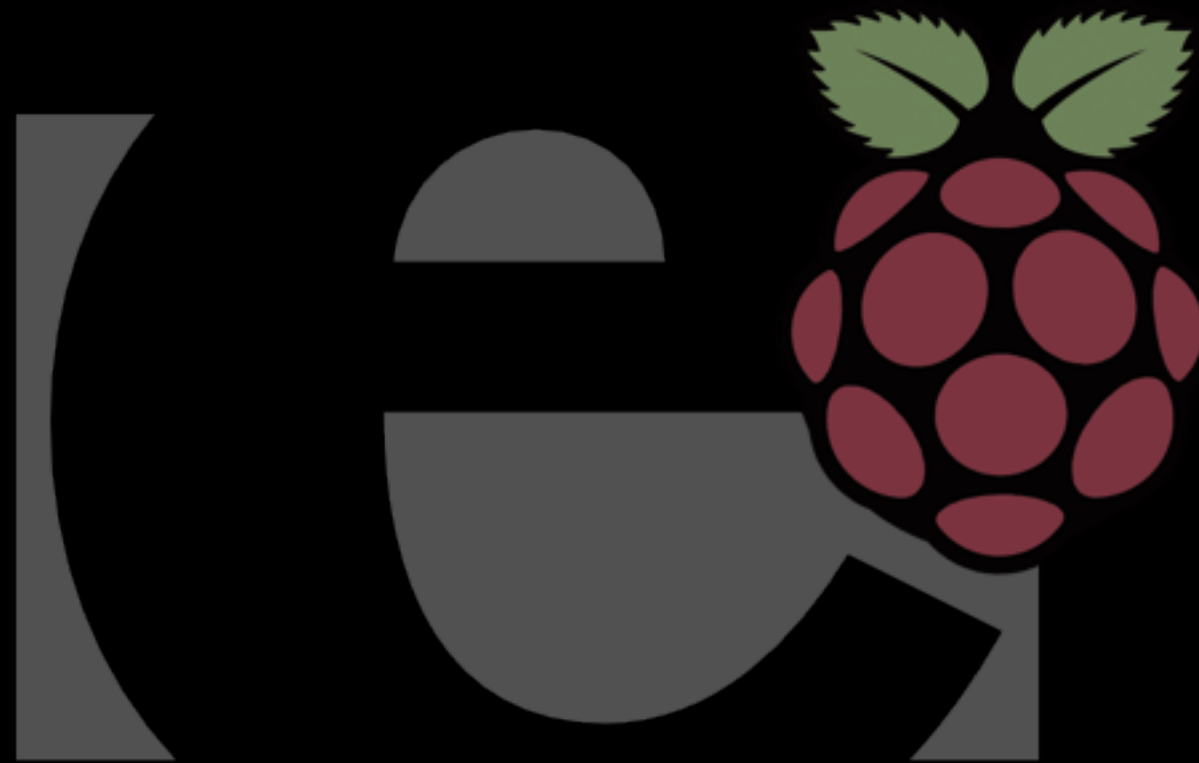


LING on Raspberry Pi

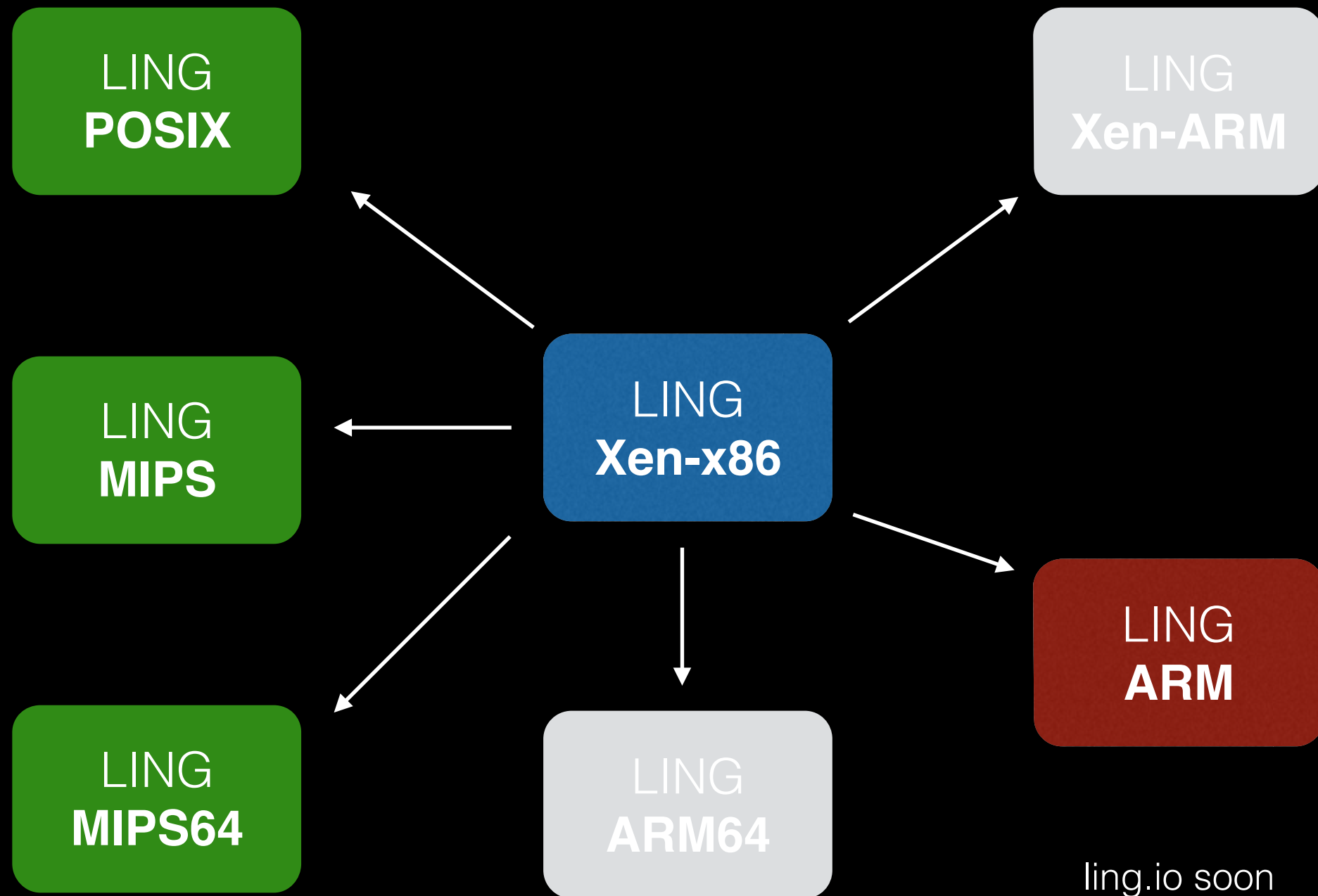
Erlang for embedded platforms



50 cent about LING

- “Yet another Erlang VM”
 - written from scratch
 - compact, fast, compatible (R17)
 - no SMP code, simple schedulers
 - low-latency GC
 - needs no disk access
- <http://erlangonxen.org>

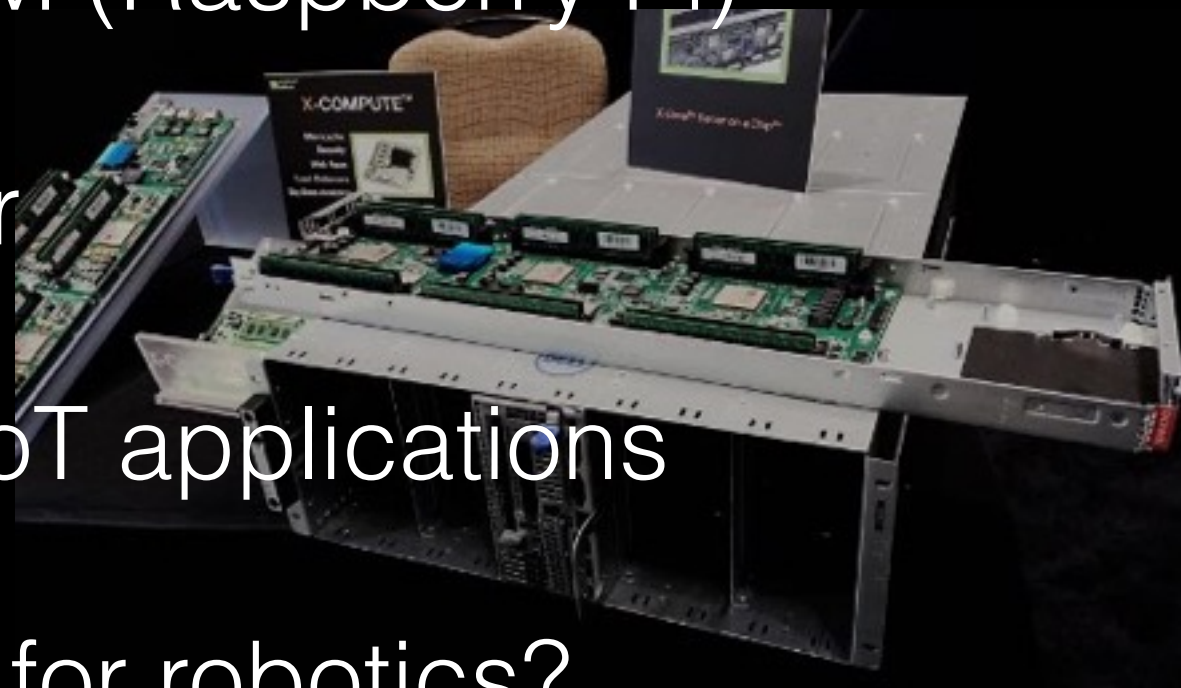
LING adds ports





LING on ARM

- No hypervisor, truly bare metal platform
- Start with 32-bit ARM (Raspberry Pi)
- ARM64 servers later
- Target mobile and IoT applications
- A software platform for robotics?



Demo

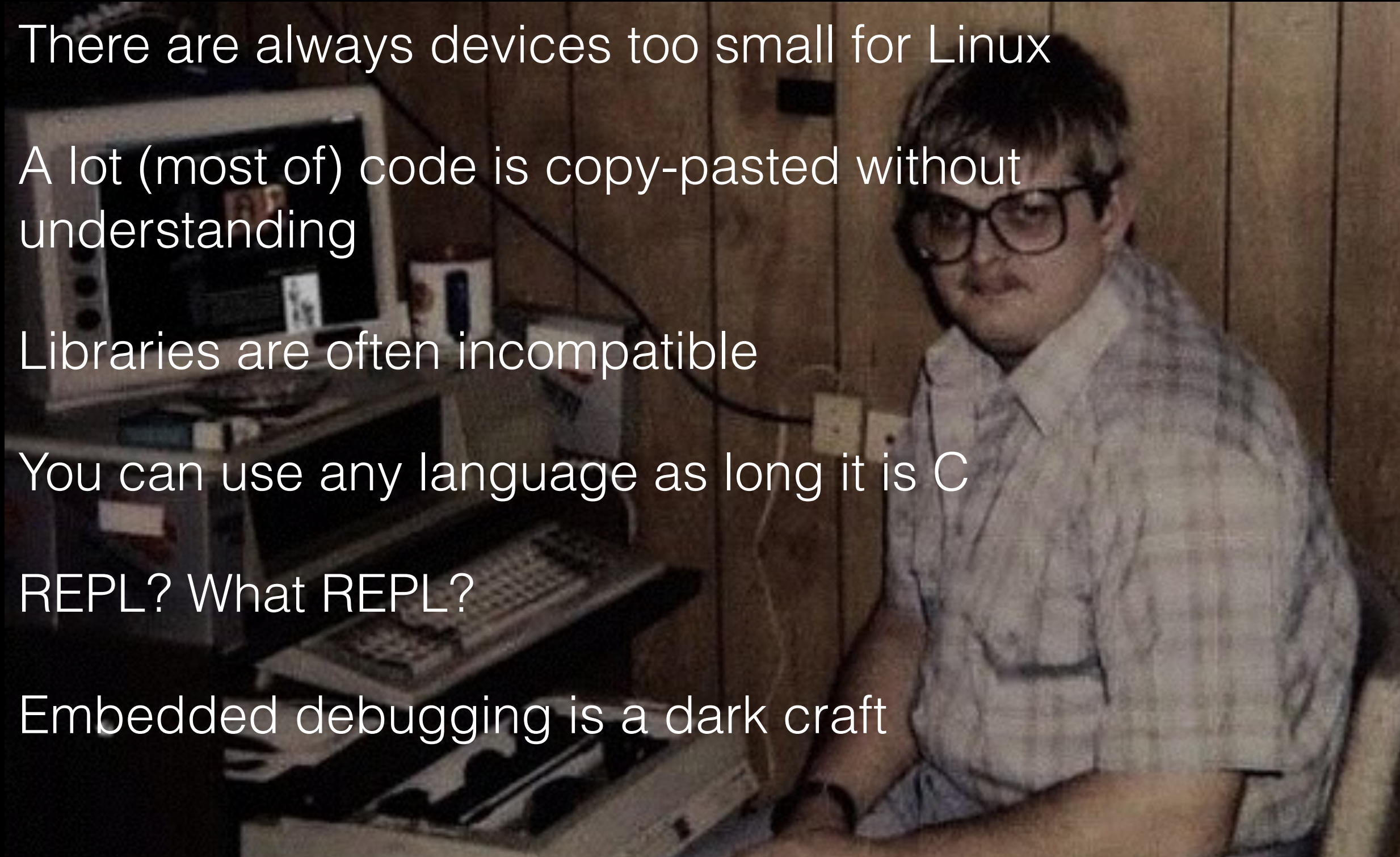


Challenges of the port

- “Hello, world” is the hardest part for a bare metal development
- Raspberry Pi serial interface requires a “level converter”
- Difficult to determine the Raspberry Pi flavour - currently only works on RPi v1b

Embedded dev today

- There are always devices too small for Linux
- A lot (most of) code is copy-pasted without understanding
- Libraries are often incompatible
- You can use any language as long it is C
- REPL? What REPL?
- Embedded debugging is a dark craft

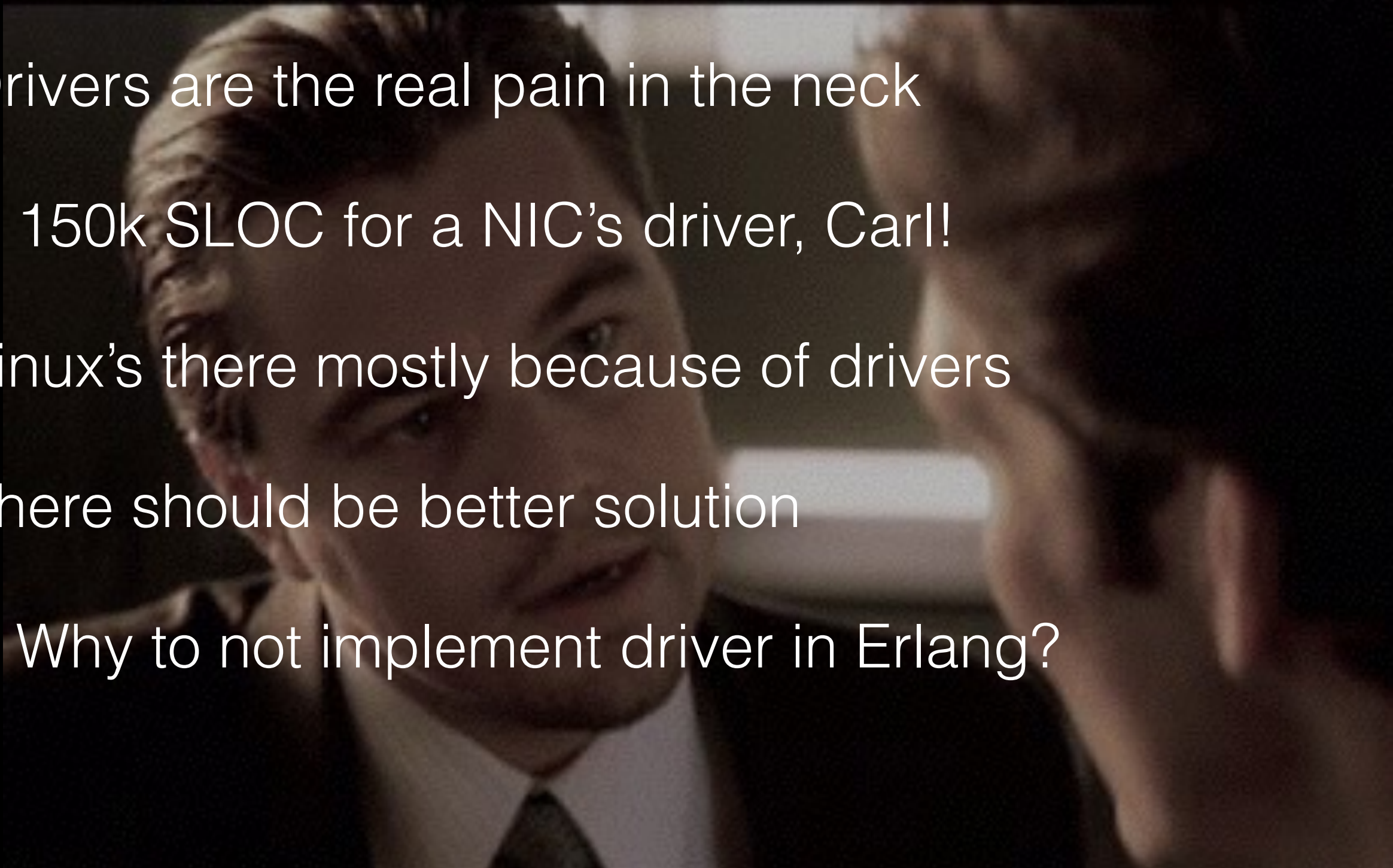


Brave new *embedded* world

- Use a high-level language (Erlang) for applications and drivers
- Dynamic languages are fast enough - justifies the use of bytecode VMs
- Erlang seems to be an ideal candidate because of:
 - Self-healing supervisors
 - Concurrency
 - Bit pattern matching
 - Hot code loading and more...

And we need to go deeper...™

- Drivers are the real pain in the neck
 - 150k SLOC for a NIC's driver, Carl!
- Linux's there mostly because of drivers
- There should be better solution
 - Why to not implement driver in Erlang?



New approach to drivers



NEW

- A lower part and a higher part a driver
- The lower part of a driver to be generated from Verilog/VHDL
- The higher part is implemented in Erlang as reaction to messages from the lower part
- No code yet
- The concept of forthcoming project codenamed L

Meet the L

- Erlang with strong types
- No type annotations - type inference only
- Compiles to binary code, no VM
- A minimal runtime (<8k)
- A compiler produces unikernels
- 2 months from Fibonacci to Black-Scholes



Join us!

