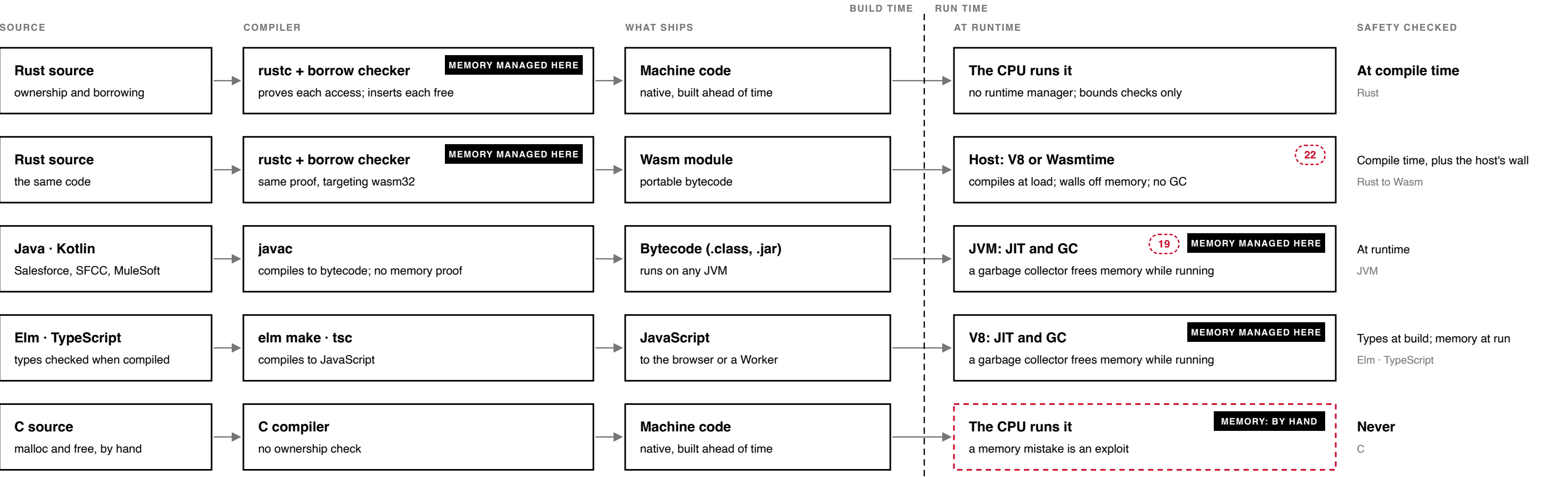


WHERE MEMORY GETS MANAGED — RUST AT COMPILE TIME, THE JVM AT RUNTIME

Rust has no bytecode and no memory manager at runtime: the compiler proves memory safety, then gets out of the way

Red dashed = a gap. Black tag = where memory is managed.

FIVE PIPELINES FROM SOURCE TO RUNNING CODE



WHY IT MATTERS FOR SECURITY

No runtime to patch, but you rebuild
A JVM fix can ship as a runtime update.
A Rust fix means recompiling and redeploying the binary.

Secrets can be wiped
Rust frees memory at a known point, so zeroize can erase a key. JVM Strings can't be changed, and the GC copies them.
19

Nothing to escape, except unsafe
The compile-time proof holds everywhere but unsafe blocks: the foot gun to search for in any Rust code review.

The JVM manages memory at runtime. Rust proves it at compile time.