Why Lua?

Performance

- Can be embedded in the same process as PHP
  - Call PHP → Lua: 2μs
  - Call Lua → PHP: 0.5μs
- Gives us flexibility in API construction
- Reduces the need for time/memory tradeoff (memoization)

Made for embedding

- Well-worn path to embedded operation
- No global variables or other technical "gotchas"
- Reference manual documents embedding in detail

Resource limits

- Allocation hook allows memory limiting
- VM line hook allows deadlock-free interruption after time limits expire
- Pages that exceed limits are added to [[Category:Pages with script errors]] instead of causing a Squid timeout

Good for non-Wikimedia users

- Standalone engine allows operation on a shared web host, as long as proc_open() is allowed
- Tiny 170KB binary
- Binaries available for many platforms
  - FreeBSD, AIX, Irix, Solaris
- Stock binary from Linux distributions is suitable

Why not V8?

- Minimal documentation
- Google's level of commitment to support for embedded applications is unclear
- No allocation hook
  - Uses ::operator new, malloc()
- Large (3.6MB) binary

Why not Rhino?

- Rhino is a Java implementation of JavaScript
- Can't be embedded in the same address space as PHP
- Very slow startup
- CPU limit expiry would cause Java process destruction and thus incur startup overhead