

# ba·bash·ka

**A native Clojure interpreter for scripting**

Michiel Borkent  
@borkdude  
2021-04-14

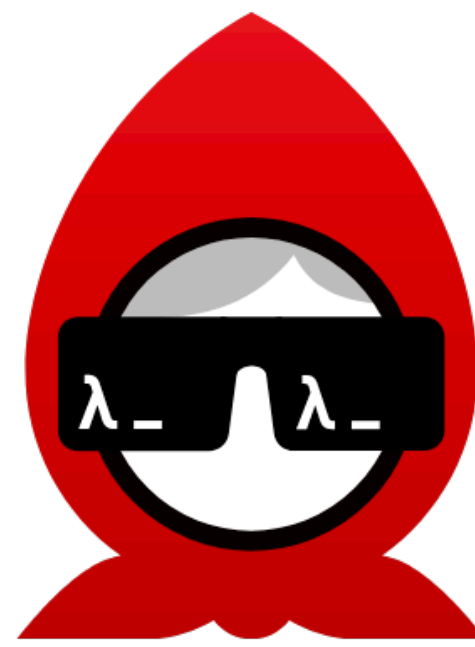


# GraalVM™

+



- CLI tools with instant startup! (< 10ms)
- clj-kondo: a linter for Clojure that sparks joy
- jet: convert between JSON, EDN and Transit
- No eval: dynamic classloader not supported!



# ba·bash·ka

```
$ time bb -e '(+ 1 2 3)'  
6  
0.00s user 0.00s system 67% cpu 0.013 total
```

- Native Clojure scripting tool, single binary, no JVM
- Can be used to replace “the grey areas” of bash
- Installable via script, brew (macOS, linux), aur (linux), scoop (Windows)

# CLJ scripting

	Runtime	Impl	Startup*	Interop	Windows	Execution	Threads
<b>clojure</b>	JVM	Java	~1048ms	+	+	Compiled	+
<b>planck</b>	JSCore	CLJS / JS	~ 728ms	+	-	Compiled	-
<b>joker</b>	Native	Go	~7ms	-	+	Interpreted	-
<b>babashka</b>	Native	GraalVM	~10ms	+	+	Interpreted	+

\*) measured with `multitime -n10 <cmd> -e '(+ 1 2 3)'` on Ubuntu Bionic with Intel i7-3770K CPU @ 3.50GHz

# Scripts

```
#!/usr/bin/env bb
(def now (java.time.ZonedDateTime/now))
(def LA-timezone (java.time.ZoneId/of "America/Los_Angeles"))
(def LA-time (.withZoneSameInstant now LA-timezone))
(def pattern (java.time.format.DateTimeFormatter/ofPattern
"HH:mm"))
(println (.format LA-time pattern))
```

```
$ pst.clj
04:58
```

# Cross platform! 🧐

Command Prompt

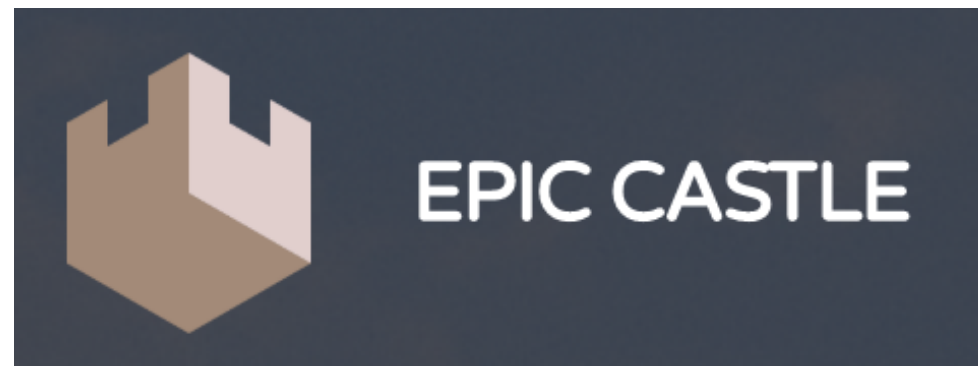
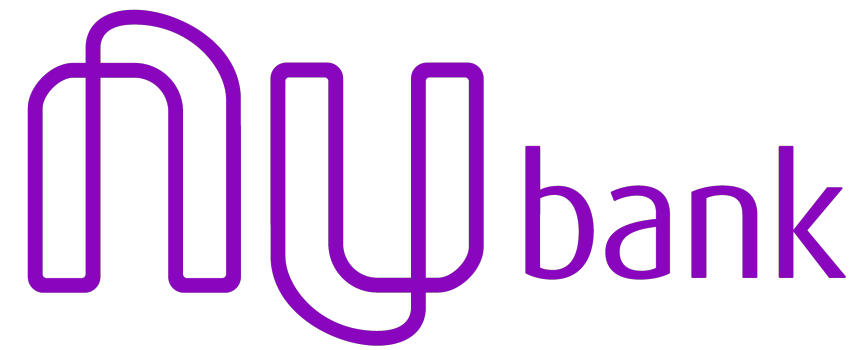
```
C:\Users\borkdude\Downloads>type pst.clj
(def now (java.time.ZonedDateTime/now))
(def LA-timezone (java.time.ZoneId/of "America/Los_Angeles"))
(def LA-time (.withZoneSameInstant now LA-timezone))
(def pattern (java.time.format.DateTimeFormatter/ofPattern "HH:mm"))
(println (.format LA-time pattern))
```

```
C:\Users\borkdude\Downloads>bb pst.clj
04:50
```

```
C:\Users\borkdude\Downloads>_
```

- Built-in libraries
- Examples:
  - python -m http.server replacement
- Pods
- External libraries / projects
- Docs: book.babashka.org

# Companies using babashka / sci



Add your company to the list:

**Github #254**







# Los Angeles Clojure Users Group

Wednesday April 8th, 2020

Writing Clojure on the  
Command Line with Babashka

0:00 / 1:01:48



Meetup: Writing Clojure on the Command Line with Babashka

451 views • Nov 17, 2020

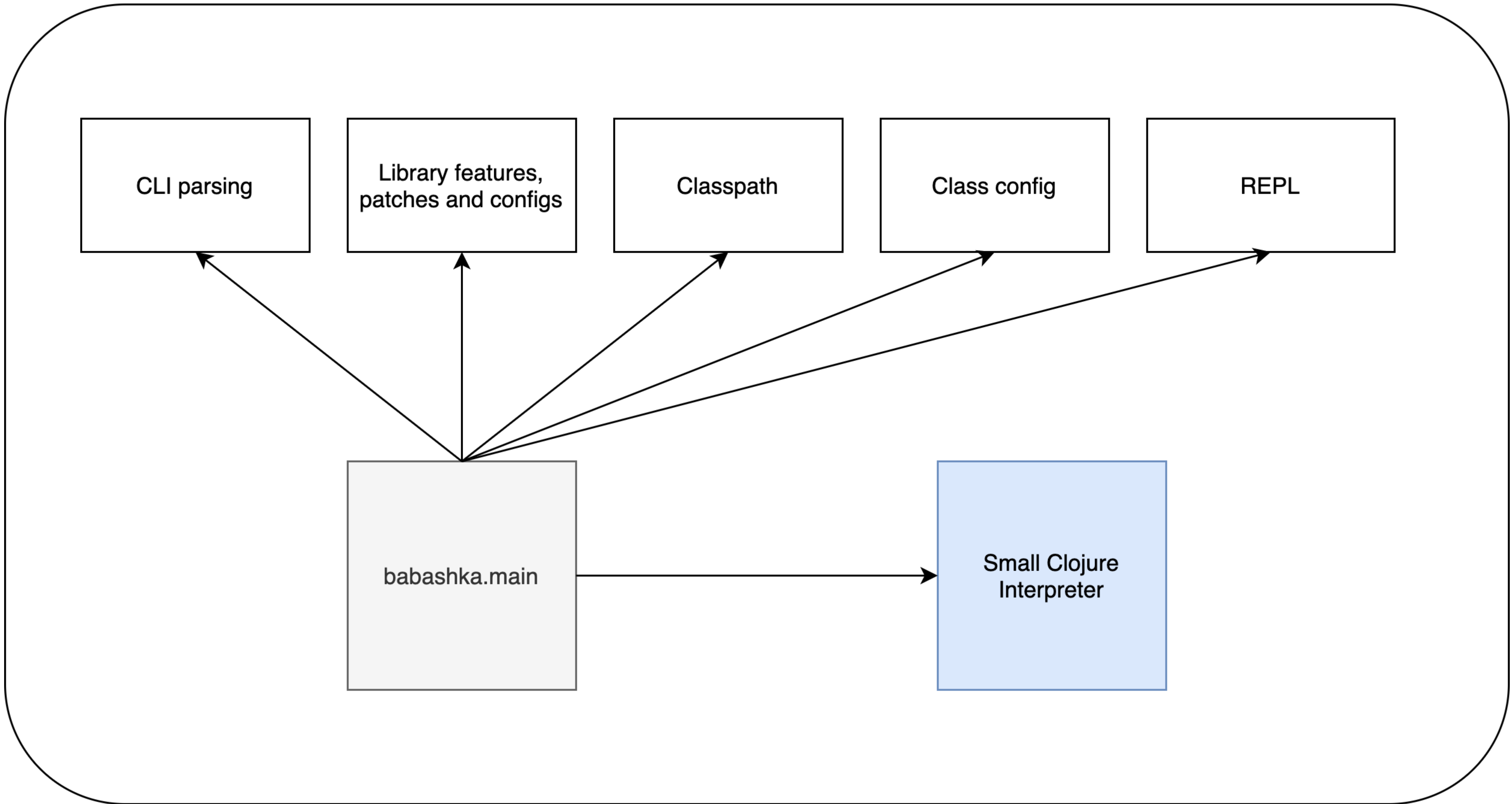
19 0 SHARE SAVE ...



Los Angeles Clojure Users Group  
39 subscribers

SUBSCRIBED





babashka

# Small Clojure Interpreter

- Clojure interpreter
- Works on JVM / GraalVM / JS
- Sandboxing
- Works in CLJS advanced compiled apps

```
(def f (sci/eval-string "#(+ 1 2 %)"))  
(f 1)  
;;=> 4
```

# SCI eval

- Why does sci work with GraalVM and `clojure.core/eval` doesn't?
- `core eval` = `compile` = generating classes
- A native image cannot load new classes at runtime
- Sci eval does not generate any new classes.  
Compilation -> interpretation.

# <https://github.com/borkdude/sci>

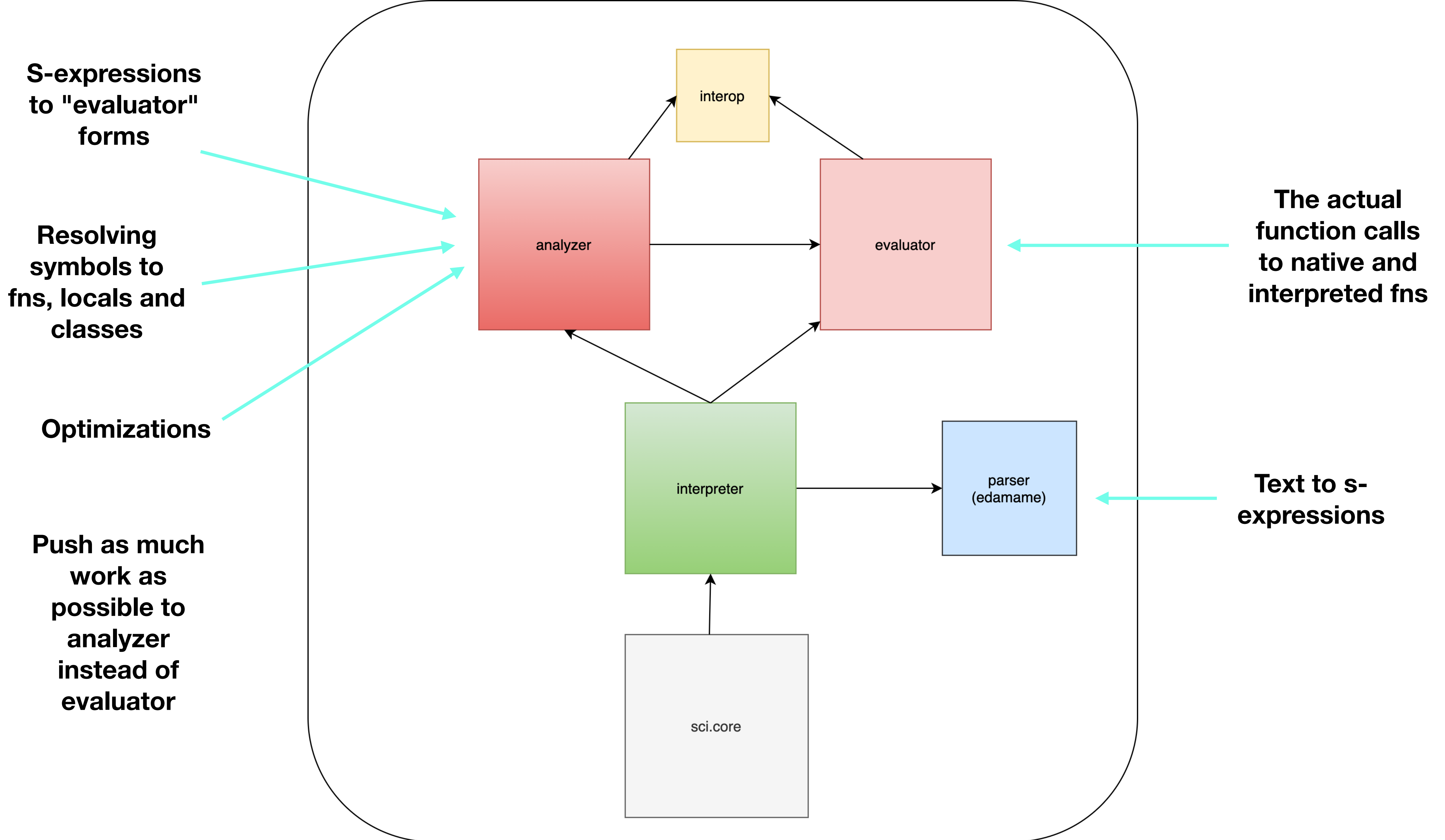
## Projects using sci

---

Sci is used in:

- [Babashka](#). A Clojure scripting tool that plays well with Bash.
- [Bootleg](#). An HTML templating CLI.
- [Bytefield-svg](#). NodeJS library to generate byte field diagrams.
- [Chlorine](#). Socket-REPL and nREPL package for Atom editor.
- [Clj-kondo](#). A Clojure linter that sparks joy.
- [ClosH](#). Bash-like shell based on Clojure. GraalVM port is work in progress.
- [Dad](#). A configuration management tool.
- [Datalevin](#). Durable Datalog database.
- [Firn](#). Org-mode static site generator.
- [Jet](#). CLI to convert between JSON, EDN and Transit.
- [Logseq](#). A local-only outliner notebook which supports both Markdown and Org mode.
- [Malli](#). Plain data Schemas for Clojure/Script.
- [PCP](#). Clojure Processor (PHP replacement).
- [PGMig](#). Fast Standalone PostgreSQL Migration Runner.
- [Prose](#). Alternate syntax for Clojure, similar to what Pollen brings to Racket.
- [SICMUtils](#). Computer Algebra System in Clojure, tailored for math and physics investigations.
- [Spire](#). Pragmatic provisioning using Clojure.
- [Zprint](#). Tool to beautifully format Clojure(script) code and data.





**S-expressions to "evaluator" forms**

**Resolving symbols to fns, locals and classes**

**Optimizations**

**Push as much work as possible to analyzer instead of evaluator**

**The actual function calls to native and interpreted fns**

**Text to s-expressions**

Small Clojure Interpreter

# Interop support

```
babashka/classes.clj at master x +
github.com/borkdude/babashka/blob/master/src/babashka/impl/classes.clj

15  (def classes
16    '[:all [java.io.BufferedReader
17           java.io.BufferedWriter
18           java.io.ByteArrayInputStream
19           java.io.ByteArrayOutputStream
20           java.io.File
21           java.io.InputStream
22           java.io.IOException
23           java.io.OutputStream
24           java.io.Reader
25           java.io.StringReader
26           java.io.StringWriter
```

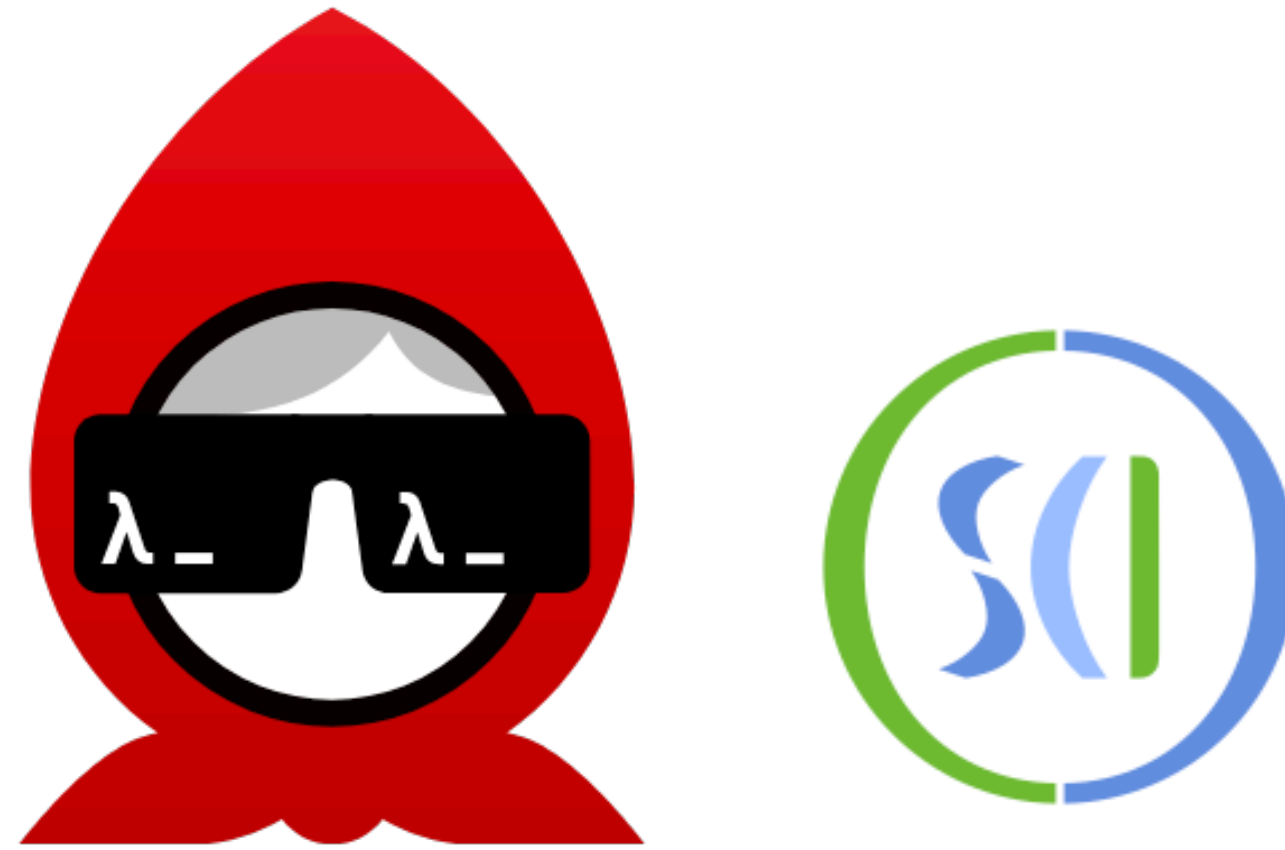
(Reflector/invokeInstanceMethod obj method (object-array args))

# REPL

<https://gist.github.com/borkdude/9390e0d450ca61d42a037c969d646d9b>



# Thank you!



## ba·bash·ka

**A native Clojure interpreter for scripting**

<https://github.com/borkdude/babashka>

<https://github.com/borkdude/sci>