ba·bash·ka

ClojureStream Workshop

A comprehensive and practical guide to **bb** and its ecosystem

Presented by

LISPYCLOUDS

Date

17/03/2023

</>

Hello and a disclaimer

@LISPYCLOUDS CHANNELING THE INNER @BORKDUDE

Find me on:

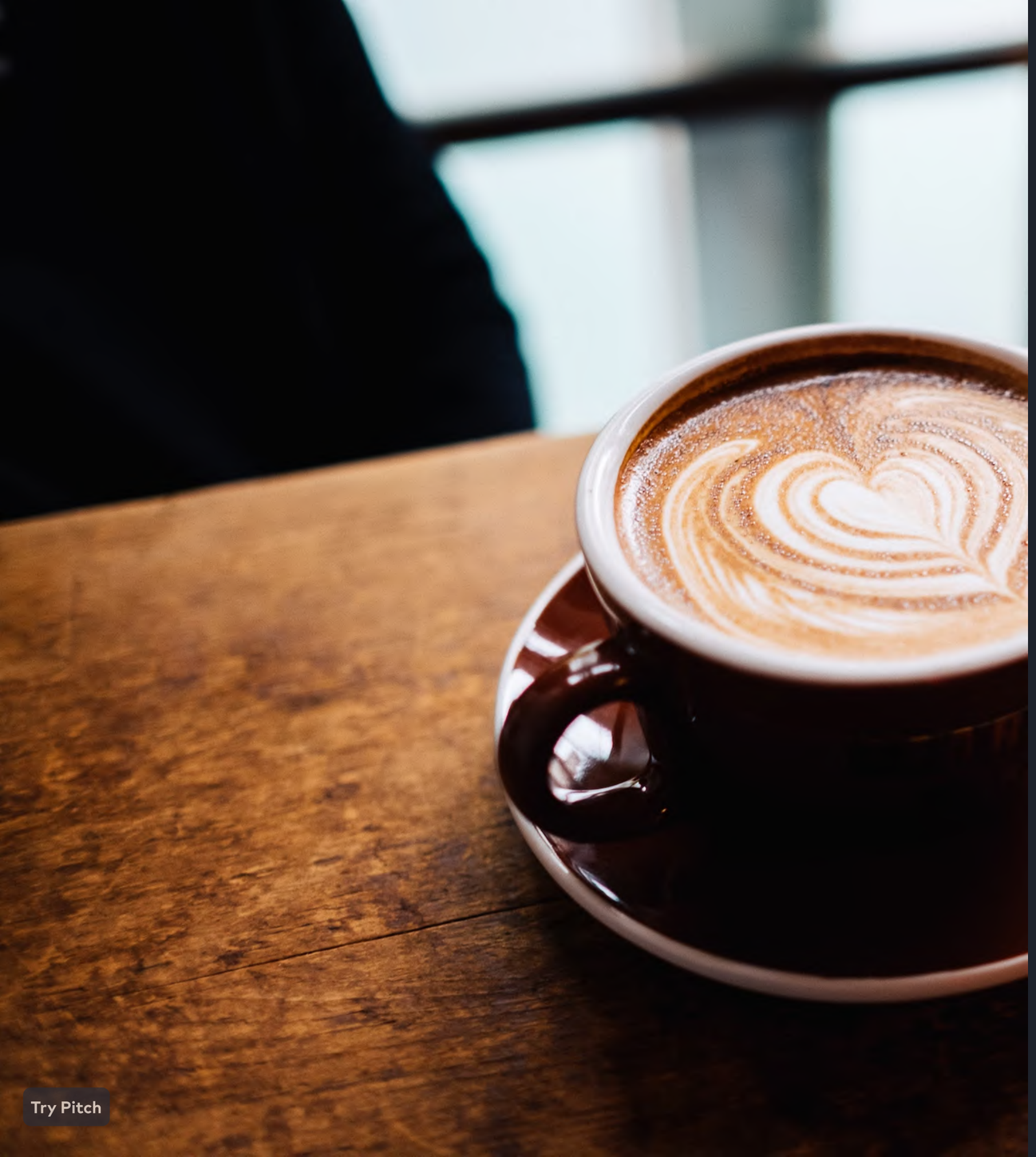
- https://github.com/lispyclouds
- https://twitter.com/lispyclouds
- https://mastodon.online/@lispyclouds
- **lispyclouds** on the Clojurians Slack

Agenda (~4 hours)

- What and Why of **bb**
- Editor and REPL setup
- Setting up the project
- Exploring the scripting issues
- Moving over to **bb**, one script/make at a time
- Building a CLI for the project
- Can this be run on **bb** itself?
- [If time and/or interest permits] Hacking and contributing to **bb**
- Conclusion and References

What and **Why** of **bb**

IT DOES HAVE A COOL NAME RIGHT?



Clojure on the JVM rocks, but...

- We've been the butt of many jokes about the **slow start-up time**. The slow start-up time is for good reasons.
- **Too much ceremony** just to write a simple script.
- Damn the python/bash/ruby **scripts run quick**!
- The clojure+java ecosystem is great but **one day I can parse JSON without a dependency**!
- lein, deps, tools.build etc etc... all **I want is a simple thing to automate** my boring tasks.
- Yeah, might as well bite the bullet and write bash/python/make. **Good that I have ChatGPT too along with StackOverflow**.

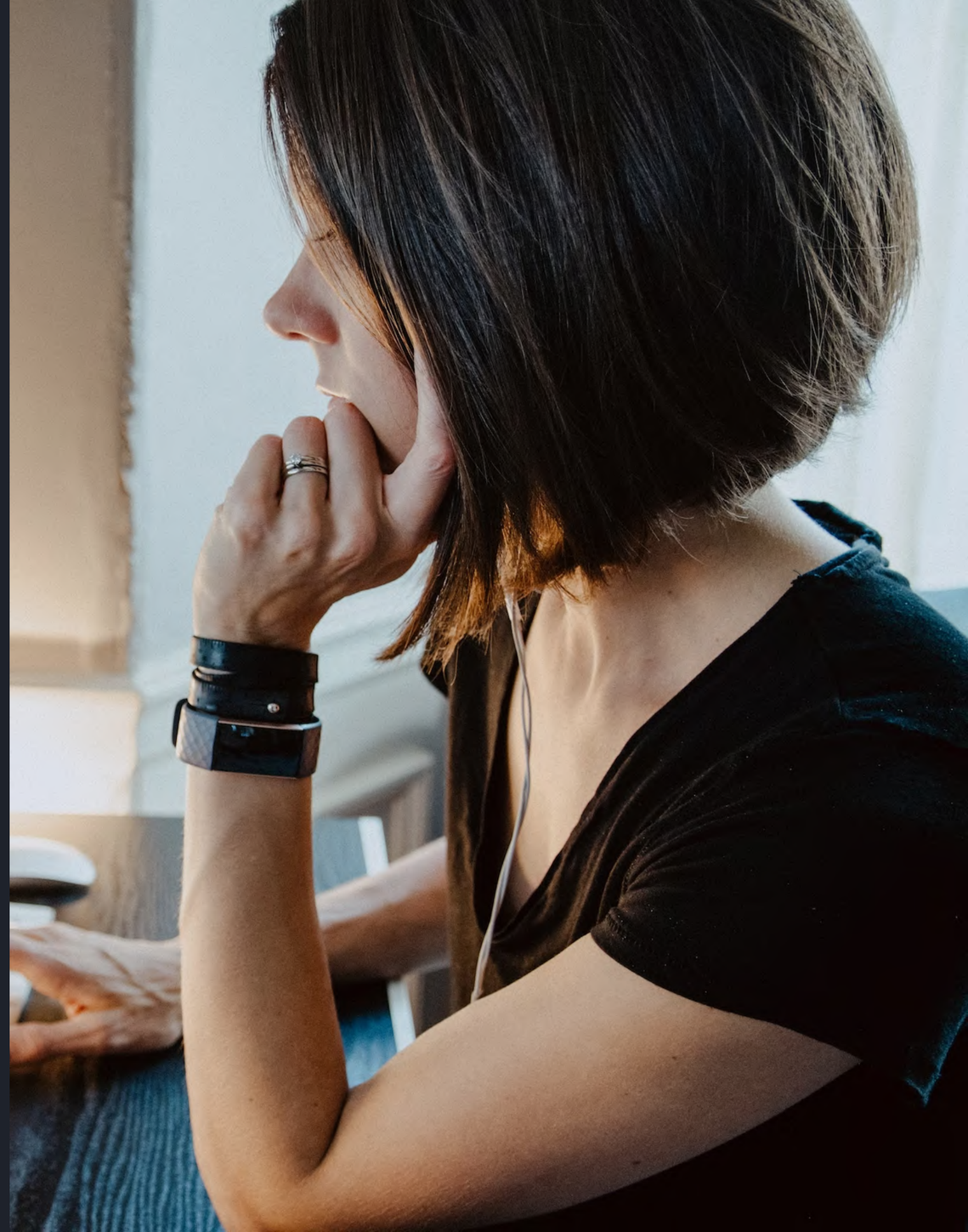
What if you could keep writing Clojure?

A WORLD WITHOUT STACKOVERFLOW/CHATGPT DRIVEN DEVELOPMENT

Who is bb for?

WHERE ALL CAN IT BE USED?

- The experienced Clojure dev **can keep writing their scripts in Clojure**.
- Folks needing more **scripting/glue code** to automate parts of their code and deployments.
- Folks working closer to the infra and **not on the JVM can still write Clojure**. Can this finally kill the stupid YAML?
- Running Clojure on **resource constrained environments as cross-platform and self contained scripts**.
- **Need for (start-up) speed**: Write quick and lean Clojure scripts for functions as a service like AWS Lambda.
- **Sneak in Clojure as more maintainable scripts** in non Clojure codebases or teams.



Editor and REPL setup

BALANCE THE ((( )))

```
bb --version
```

```
clojure --version
```

Let the editor wars begin!

The indispensable Clojure LSP: https://clojure-lsp.io/installation/

Neovim, the best one!

https://github.com/Olical/conjure

Emacs

https://docs.cider.mx/cider/index.html

VSCode/Codium

https://calva.io/

Vim

https://liquidz.github.io/vim-iced/

Sublime Text 4

https://github.com/tonsky/Clojure-Sublimed

IntelliJ IDEA

https://cursive-ide.com/

script.clj

```clojure
(defn fact
  [n]
  (reduce * 1 (range 1 (inc n))))

(when (= *file* (System/getProperty "babashka.file"))
  (println (fact 5)))

(comment
  (fact 5))
```

Setting up the project

FINALLY SOME CODE!

The source code

git clone https://github.com/clojurestream/babashka-workshop

Trying it out

- Running tests: **make test** or **clojure -X:test**
- Start nREPL: **make nrepl** or **clojure -M:nrepl**
- Try connecting your editor and try out eval-ing.
- Start it locally: **make start** or **clojure -M -m todos.main**
- Interacting with the API with babashka.http-client:
  - `(require '[babashka.http-client :as http] '[clojure.java.io :as io])`
  - `(http/get "http://localhost:8080/todos")`
  - `(http/post "http://localhost:8080/todos" {:body (io/input-stream "test/payload.json")})`
  - `(http/delete "http://localhost:8080/todos/<id>")`
- Build the container image: **make image** or **clojure -T:build uber && ./script/image.sh**

Do we see any issues?

Exploring the scripting issues

HERE BE DRAGONS

Issues

NOT A BED OF ROSES

- FACT: **No one can write shell scripts** correctly.
- FACT: **No one can write Makefiles** correctly. Also, *WHAT* is up with the **.PHONY tabs**?
- The shell scripts are **not cross platform**. Windows shouldn't be ignored.
- **Complexity creeps up** in the automation code fast.
- **Big mental shift** when writing scripts.
- Pack everything in docker? **RIP hard disk space**.
- Would be nice to not only **keep writing Clojure** here but also **have the same reuse and maintainability** as the app code.

</>

Moving over to bb

ONE SCRIPT/MAKE AT A TIME

bb.edn and tasks

RIP MAKE

- https://book.babashka.org/#project-setup
- https://book.babashka.org/#tasks

Blazingly Fast™ builds

SERIOUSLY, RIP(IECES) MAKE AND SHELL SCRIPTS TOO

Building a CLI for the project

THE BB SWEET SPOT

TODO: CLI

WHEN YOU HAVE NO IDEA HOW TO DO FRONTEND

- `$ bb cli todo --title "learn frontend" --due "2023-03-11 13:00:00.000"`
- Should do an interactive prompt if any one is missing.
- `$ bb cli done --id "<id of the task>"`
- This should list all the todos if the id isn't supplied and delete the one chosen.
- `$ bb cli ls`
- Lists all of the tasks in a nice table.

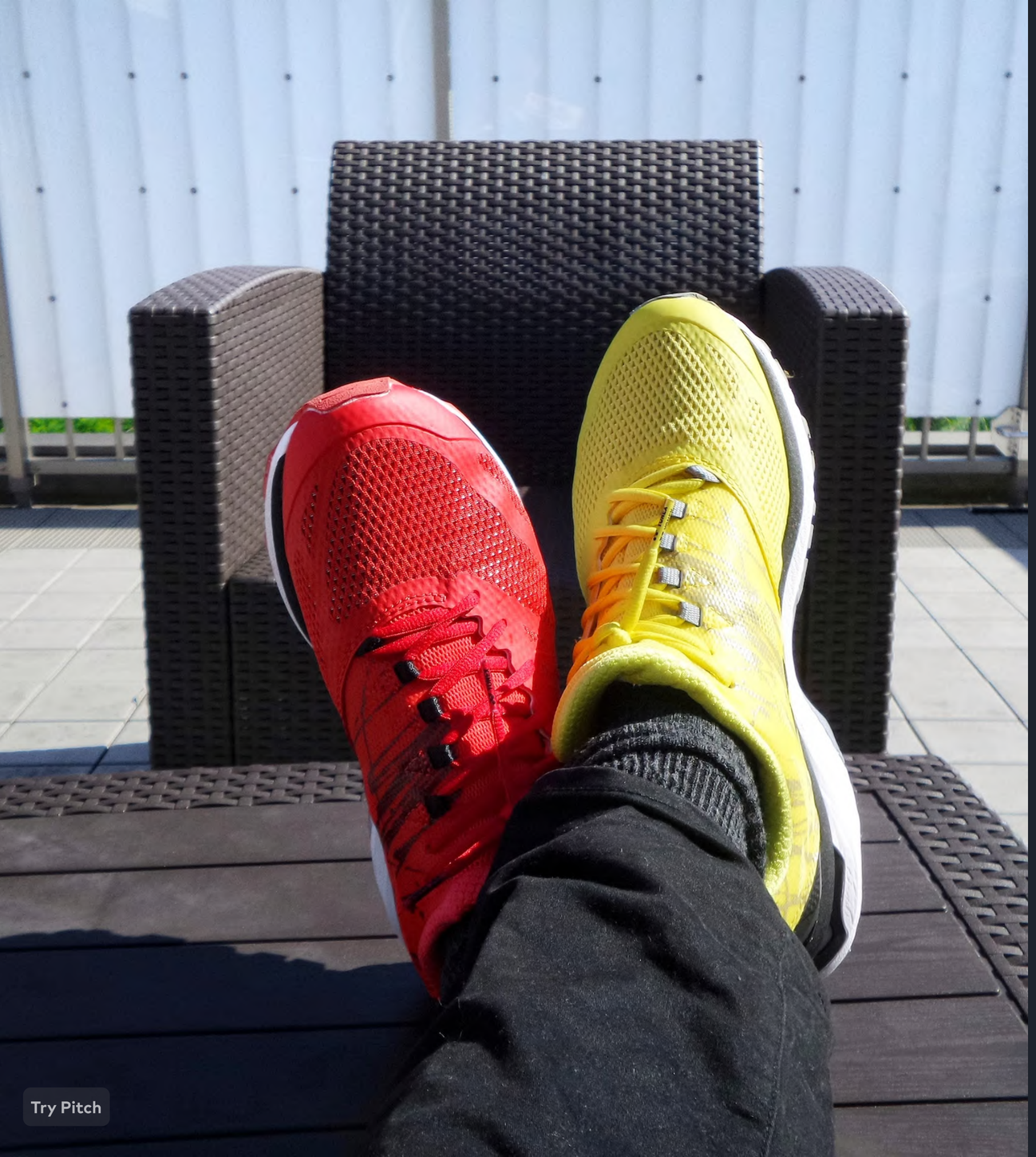
bbin

MAKE INSTALLING SCRIPTS **EASY** AGAIN

https://github.com/babashka/bbin

Can this be run on bb itself?

I NEED TO RUN THIS ON AN AWS LAMBDA!



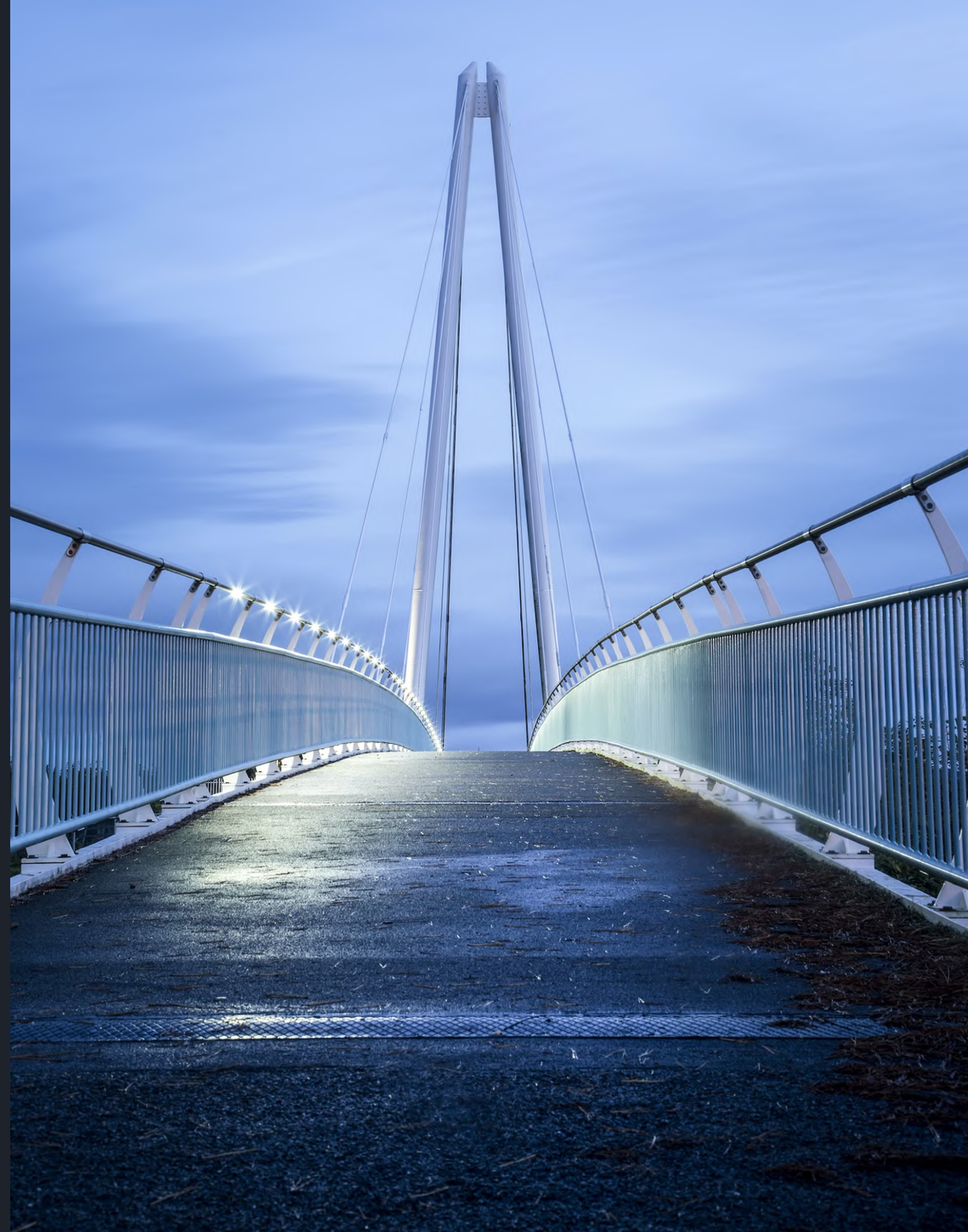
Differences from Clojure/JVM

- bb **is an Ahead Of Time compiled** binary which **interprets Clojure code** unlike the Clojure compiler.
- Only **a pre-selected set of Java classes are supported** as there is no bytecode compiler of the JVM.
- **Native code access from bb is not possible** for similar reasons as of now. (Might change, stay tuned!)
- More info and differences: https://book.babashka.org/#differences-with-clojure

Babashka Pods

STEALING ALL THE COOL THINGS FROM OTHERS

- Pod is "Bridge" in Romanian. Yeah, **we really love naming things.**
- The SQLite JDBC driver we used:
  - **Has custom Java classes**, bb needs to be compiled with them.
  - **Performs calls to the native SQLite library**.
- Pods **provide a mechanism for interacting over stdout/in or sockets with things outside of bb**. Think Remote Procedure Calls (RPC) but simpler.
- Pods **can be any language too**!
- We can use the SQLite3 pod **written in Go** to make this code bb compatible.
- More info: https://book.babashka.org/#pods



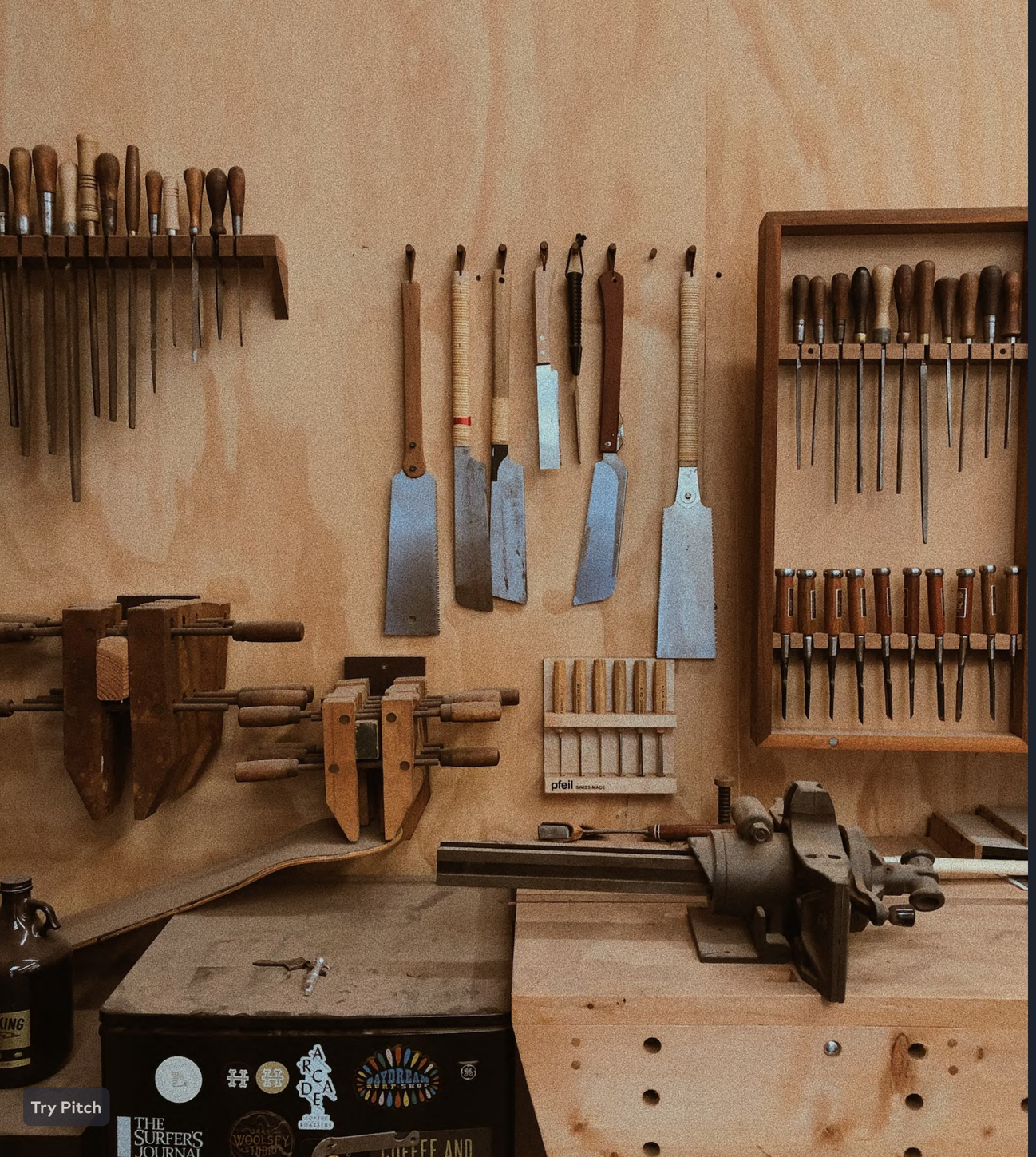
Podding away!

IT'S ALL IN THE BB UNIVERSE

1. Switch from **next-jdbc+xerial/sqlite-jdbc** to **babashka/go-sqlite3**
2. Change the start task to **bb -m todos.main**
3. Make an uberjar with bb
4. Change the Dockerfile to use bb as the driver
5. Test it all out

Hacking and contributing to bb

YES, WE DO PULL REQUESTS 😉



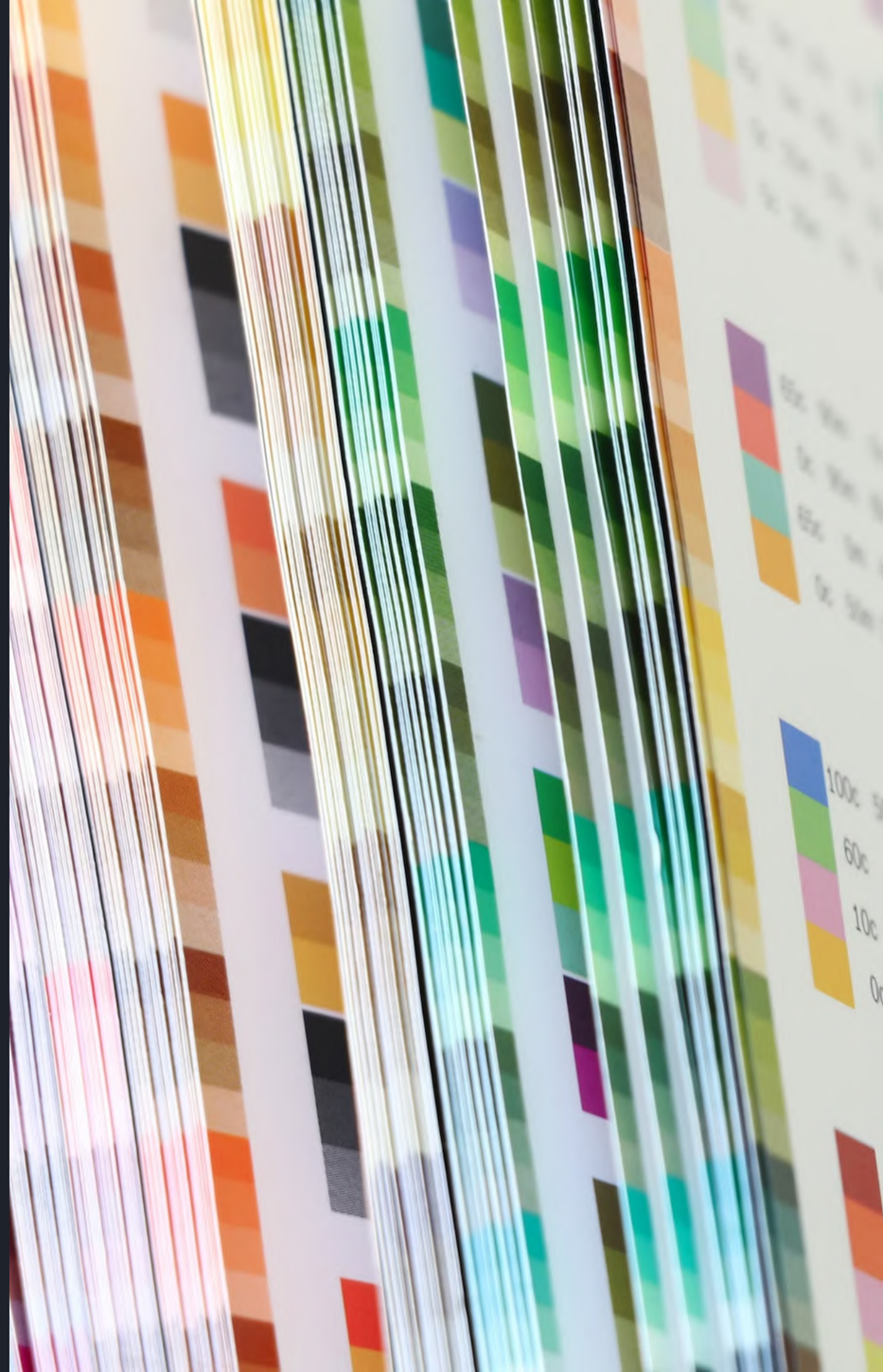
Contributing

- We are **always on the lookout for more folks** to get their hands dirty with the source! There's **a whole universe of things to contribute** to around bb as well!
- Code lives on: https://github.com/babashka/babashka
- Come **chat with us** on the #babashka channel on the Clojurians Slack.
- Checkout the developer doc.
- Recommended steps:
  - **Look for an issue** that may already address your needs **or make one** before contibuting. A conversation can also be started on the slack channel.
  - **Raise a Pull Request(PR)** from a fork containing:
    - The code **changes**
    - **Tests** for the changes
    - **Changelog**
    - **Docs** if needed
- **PRs are given high priority** and would be addressed ASAP!

References

ONE-STOP SHOP

- Site: https://babashka.org
- The Book: https://book.babashka.org
- The Babooka: https://www.braveclojure.com/quests/babooka
- The wiki: https://github.com/babashka/babashka/wiki
- GitHub: https://github.com/babashka/babashka
- The first talk: https://www.youtube.com/watch?v=Nw8aN-nrdEk
- Clojurians Slack: #babashka
- The bb ecosystem: https://github.com/babashka
- Collection of more external resources: https://github.com/babashka/babashka#articles-podcasts-and-videos
- Follow the **#babashka** hashtag on twitter and mastodon and the Babashka News.



The ~~bork~~superdude

Consider sponsoring **borkdude** on:

- https://github.com/sponsors/borkdude
- https://opencollective.com/babashka
- https://ko-fi.com/borkdude

Thank You!

MAY THE START-UP TIME BE WITH YOU!