Pharo Launcher from command line

PharoDays 2022

David Bajger

Agenda

Intro

Purpose, Brief history

Launcher command structure (organization)

Using CLI (Demo)

How it is implemented (Demo)

How it is tested (Demo)

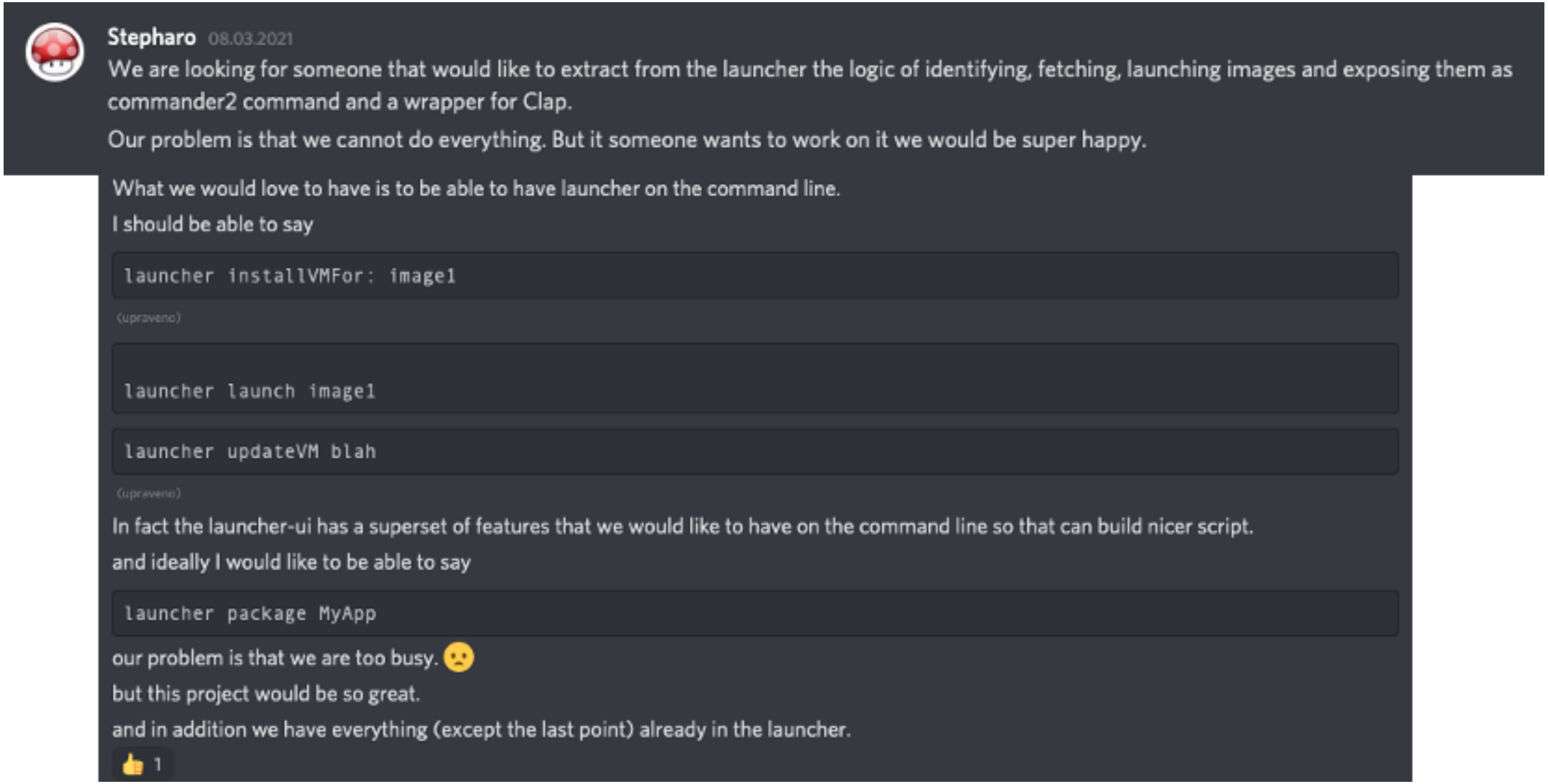
What’s next

Purpose of PharoLauncher CLI

Motivation for CLI
- Automate Pharo image manipulation in dev or production environment without need to use UI
-Faster than interactive version, for those who prefer CLI

Pharo users can:
Control Pharo images and its VMs via command line interface (programatically)
create, list, update, delete, execute, launch, package Pharo images
Update, list, delete VMs
List templates and categories

How it started



Stepharo 08.03.2021

We are looking for someone that would like to extract from the launcher the logic of identifying, fetching, launching images and exposing them as commander2 command and a wrapper for Clap.

Our problem is that we cannot do everything. But it someone wants to work on it we would be super happy.

What we would love to have is to be able to have launcher on the command line.

I should be able to say

```
launcher installVMFor: image1
```

(upraveno)

```

launcher launch image1
```

```
launcher updateVM blah
```

(upraveno)

In fact the launcher-ui has a superset of features that we would like to have on the command line so that can build nicer script.

and ideally I would like to be able to say

```
launcher package MyApp
```

our problem is that we are too busy. 🙁

but this project would be so great.

and in addition we have everything (except the last point) already in the launcher.

👍 1

The momentum

David – initial specification and class structure
Axel Marlard – core command logic and command execution
Christophe Demarey – CLAP issues, framework, refactoring, CI setup

Now

- Proper exception handling, shell unit tests, OSSubprocess use to catch errors, additional commands, bug fixing, pretty-printing, PR reviews

Github project: https://github.com/PharoLauncherCmdLine/CommandLineLauncher

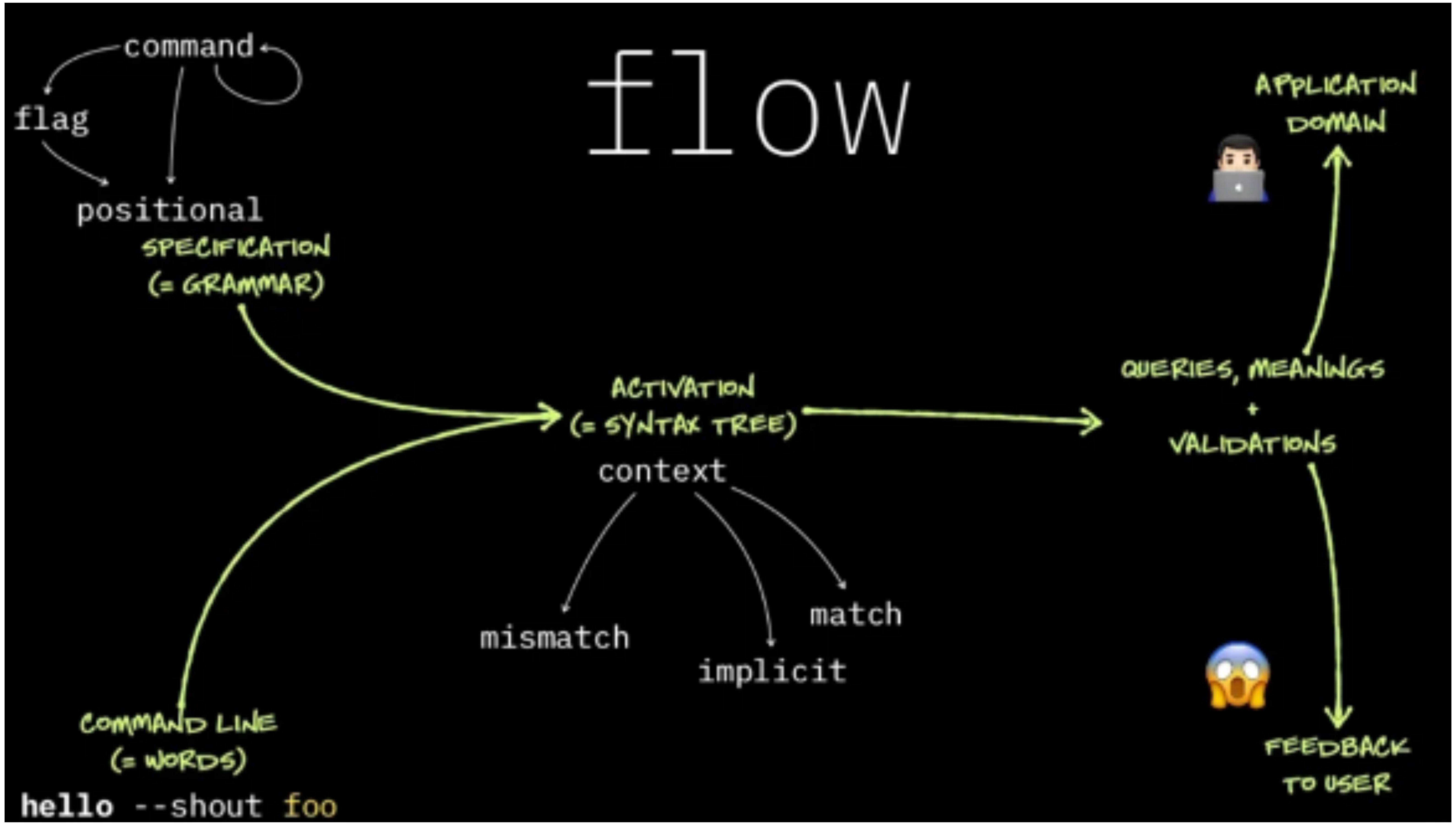
CLI structure

Thoughts on **Verb-noun oriented**
Using imperative, Like Docker or Git
- Some verbs are just for only one subject
- examples: *create image “myImage”, list vm, launch image “myImage”*
- Structure is more flat (launcher help will show many commands)

Noun-verb oritented (winner)
- More smalltalkish (object receiving message)
- Less complex on 1st level (launcher help prints less commands)
- Examples: *vm list, image create fromTemplate “myImage”, image processList, template categories*

Launcher CLI description (WIP)

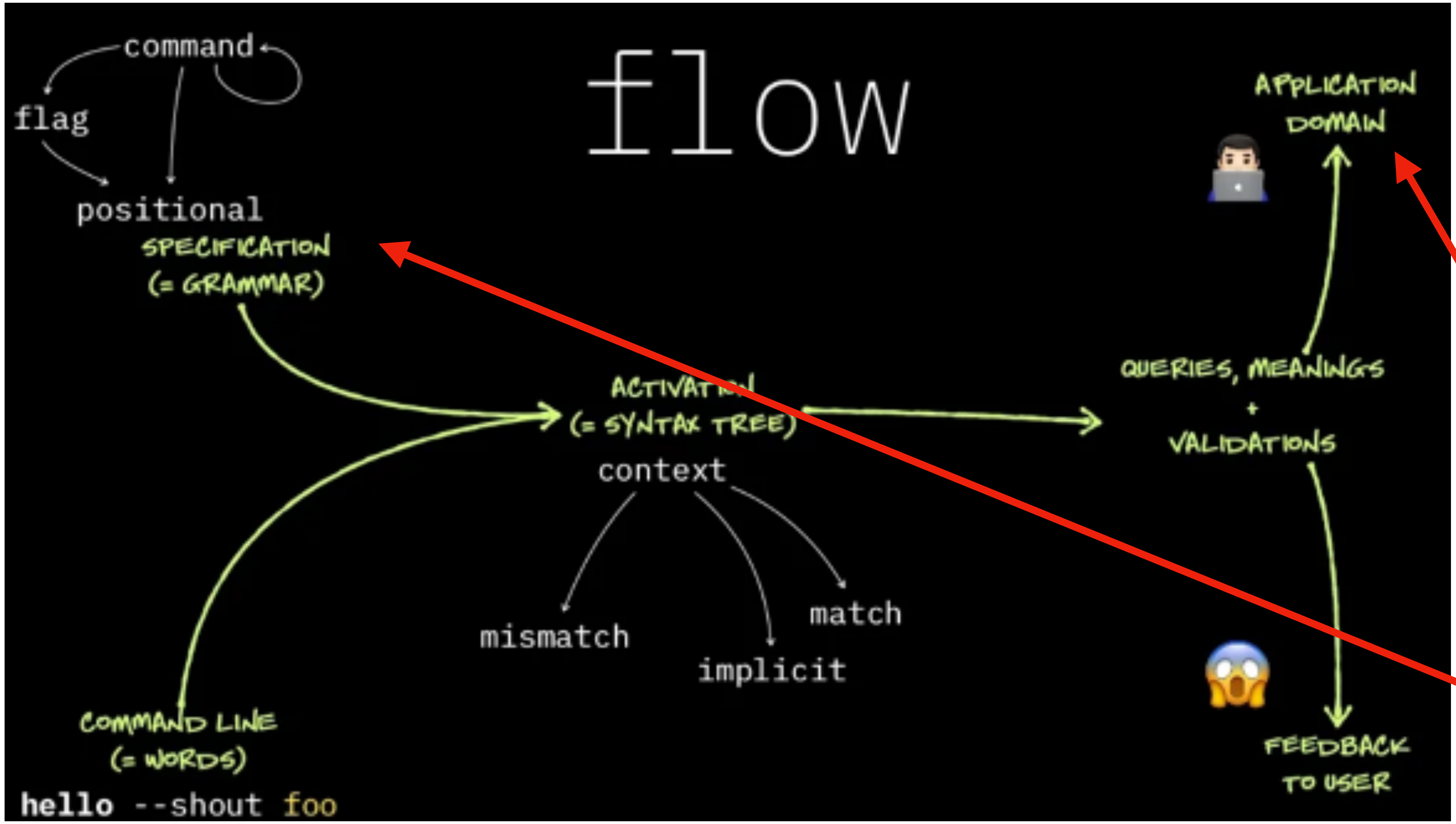
How it is implemented



Command
Line
Argument
Parser

https://github.com/pharo-contributions/clap-st

How it is implemented



C*ommand*
L*ine*
A*rgument*
P*arser*

2. Define ”execute” evaluated on instance of ClapApplication

1.Command structure (args, flags) —> class side of ClapApplication hiearchy

https://github.com/pharo-contributions/clap-st

How it is tested (demo)

What’s next?

1. Adopt project to work on P10
- UI (switch to Spec2)
- Baseline adoption (minimalistic version without UI)
- issue fixing, robustness

2. Merge project repo with main PharoLauncher repository
3. Continue with command improvements (port to UI some new: processes, packaging)
4. Proper exception handling
5. Additional tests

Thank you for listening!
(Q&A)