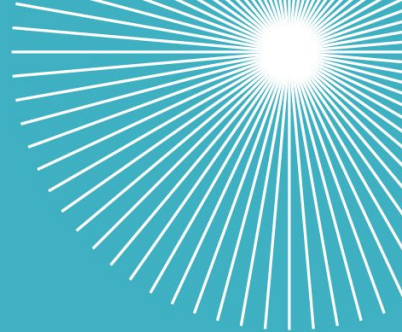


avito.tech

Ереван — 2023



ЛИНТЕРЫ

для сотен микросервисов



Павел Агалецкий

ведущий разработчик @ paas-dev



Павел Агалецкий

ведущий инженер

Юнит PaaS внутренняя платформа
для разработчиков

Занимаюсь созданием инструментов
для разработчиков и внутренней
шиной данных



о чём доклад

- ❏ контекст
- ❏ почему нельзя просто добавить Makefile
- ❏ что даёт платформа запуска линтеров
- ❏ как она работает под капотом
- ❏ pros & cons

КОНТЕКСТ



контекст об авито

- больше **1000** микросервисов...
- ... которые поддерживают **десятки команд разработки**
- ... написанные на четырёх языках:
Go, Python, PHP, NodeJS

РaaS в авито

- ★ решение типовых задач при разработке
- ★ распространение best practice
- ★ общие для всей компании решения

... В КОНТЕКСТЕ качества кода

- ★ хотим, чтобы код сервисов проходил проверки перед деплоем
 - ★ ... и локально
- ★ давать разработчикам управлять линтерами в своих проектах
- ★ иметь обязательные линтеры, общие для всех

КАК ОБЫЧНО ДОБАВЛЯЮТ ЛИНТЕР В ПРОЕКТ

**КТО-ТО
небезразличный...**

приходит
и всё настраивает!

добавляет Makefile

```
.PHONY: fmt
fmt: # Reformat code
    bin/gofumpt -w ./cmd ./internal ./tools
    bin/goimports -w -local code.pkg/srv/service ./cmd ./internal
.PHONY: lint
lint: # Lint code
    bin/golangci-lint run --skip-dirs='(^|/)pkg/pterm($|/).'
```

настраивает в CI

Administration / <Root project> / Libraries / Go libraries / gl/logger

☐ run tests

General Settings

Version Control Settings 1

Build Steps 5

Triggers 1

Failure Conditions

Build Features 2

Dependencies

Parameters 44

Agent Requirements

Suggestions

Based on Libraries / Go libraries / run tests (detach)

Build Step (4 of 5): Lint (Classic) | ▾

Runner type: ▾
Runs a command under the specified virtual environment

Step name:
Optional, specify to distinguish this build step from other steps.

Execute step: ? ▾ ?

Virtual Environment

пишет README.md

работает ли это?

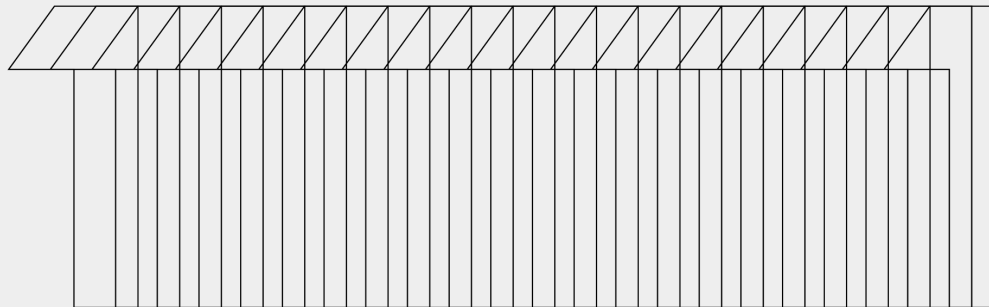
когда один, два,
10 сервисов — да

а если сервисов сотни?

1. разнородность подходов
2. требуется поддержка со стороны команды
3. иногда хочется каких-то общих линтеров

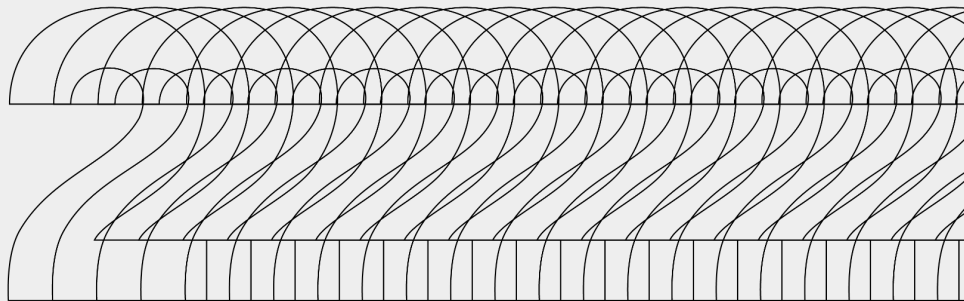
разнородность подходов

- у каждого свой Makefile, Taskfile, etc.
- переизобретение одного и того же



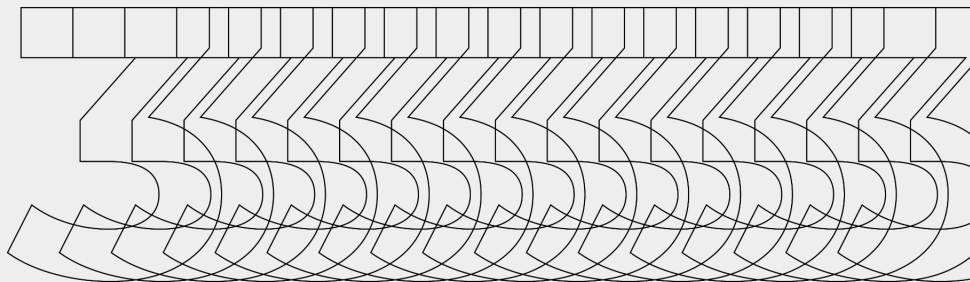
нужна поддержка в команде

чтобы не поросло мхом,
нужно периодически
возвращаться к проблеме



иногда хочется общих линтеров

- проверки безопасности
- валидация конфигов
- ...



какие инструменты использовать

Makefile — инструмент
на все времена

<https://www.yieldcode.blog/post/why-you-should-adpot-makefile-in-all-of-your-projects/>



какие инструменты использовать

Lefthook — автоматизация
работы с Git Hooks

<https://github.com/evilmartians/lefthook>



какие инструменты использовать

Taskfile — современная
альтернатива Make

<https://taskfile.dev/>



ПЛАТФОРМЕННОЕ РЕШЕНИЕ

платформа — Avito Actions

Avito Actions — это механика запуска пользовательских действий (actions) во время возникновения событий (events) над репозиторием

Avito Actions

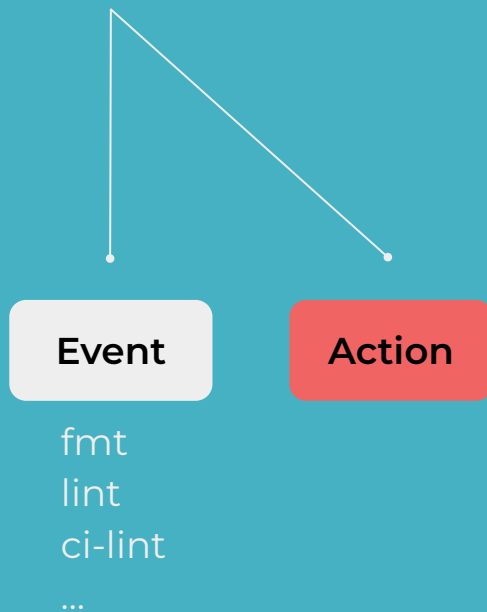


A vertical white line with a dot at the bottom connects the title 'Avito Actions' to a red rounded rectangle containing the word 'Event'.

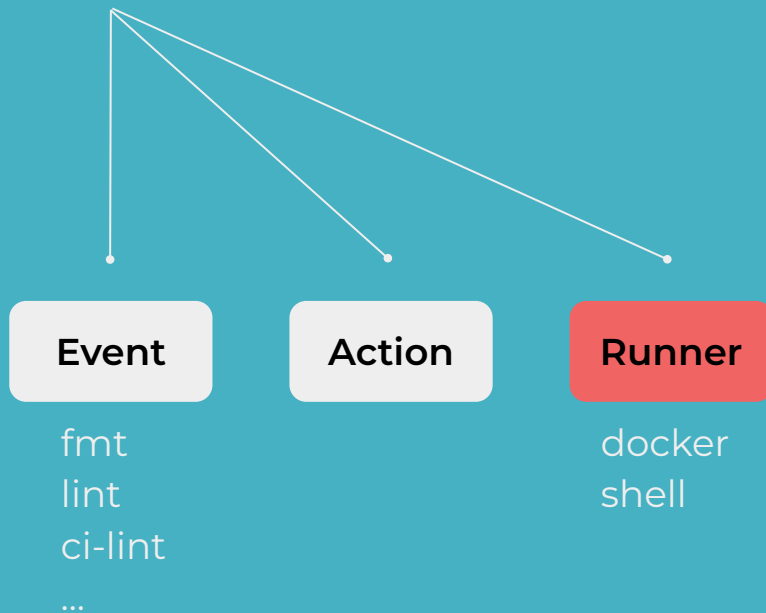
Event

fmt
lint
ci-lint
...

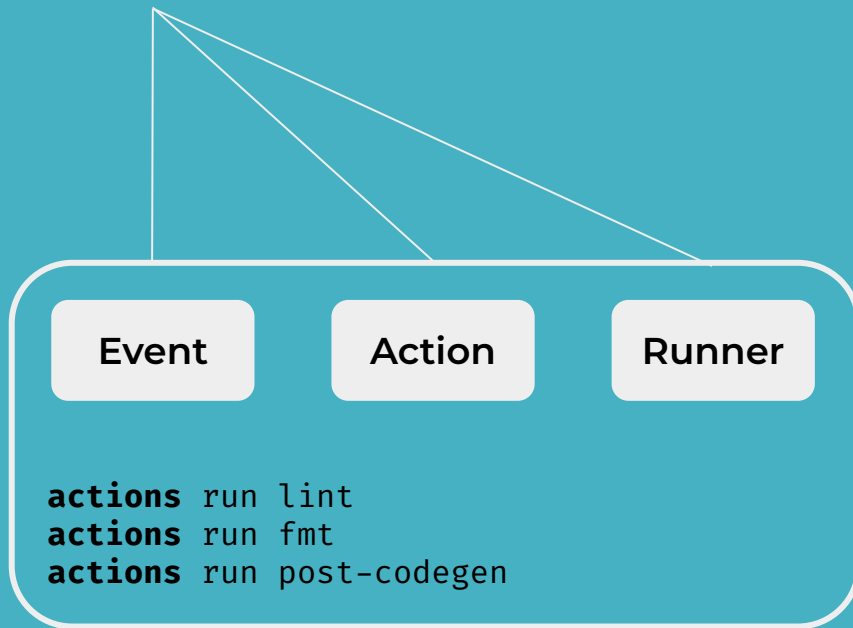
Avito Actions



Avito Actions



Avito Actions



что даёт общая платформа

- единый формат описания линтеров, форматтеров и прочего
- стандартный способ их запуска
- готовые языковые линтеры
- интеграция с другими компонентами платформы
- создание собственных линтеров
- воспроизводимость сборок

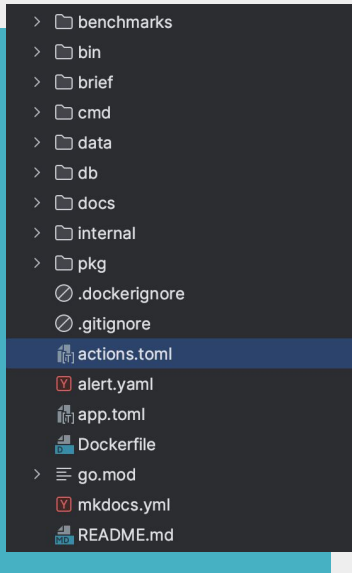
единый формат описания

*
вдохновлялись Github Actions

```
name: GitHub Actions Demo
run-name: ${{ github.actor }} is testing out GitHub
Actions 🚀
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🎉 The job was automatically
```

единый формат описания

actions.toml:



```
[[actions]]  
on = ["fmt"]  
runner="docker@avito/actions-golang:1.20"  
run = "fmt v1.53"
```

```
[[actions]]  
on = ["lint", "ci-lint"]  
runner="docker@avito/actions-golang:1.20"  
run = "lint v1.53"
```

стандартный способ запуска

```
$ avito fmt
```

```
$ avito lint
```

```
$ avito post-codegen
```

стандартный способ запуска

В CI:

Libraries / Go libraries / av/avito-cli / run unit tests

✓ #2872 at 14 Aug 13:47 ☆

Success

master Actions Details

Overview Changes 5 **Build Log** Dependencies Artifacts More

Expand All Collapse All All Messages Search in log...

Step 1/4: make test (Docker / Vagrant) 20s Step 2/4: Lint (Avito Actions) (Command Line) 20s

```
13:48:36 > docker | {"action":"","level":"INFO","msg":"" (docker@avito/actions-brief:1.0) lint
13:48:36 Actions summary (see details in the log above):
13:48:36 (docker@avito/actions-golang:1.20) lint v1.52 PASS
13:48:36 (docker@avito/actions-brief:1.0) lint PASS
```

ГОТОВЫЕ ЯЗЫКОВЫЕ ЛИНТЕРЫ

Golang: golangci-lint

Python: black, isort, ruff, mypy, ...

PHP: phpstan, php-cs-fixer, ...

NodeJS: eslint, csslint, ...

интеграции

отображение
результатов
на Code Review

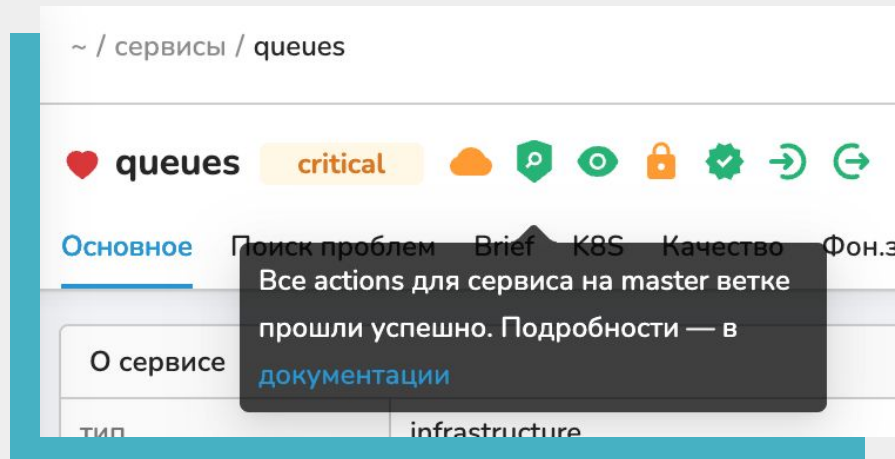
```
13 + const (  
14 +     cacheSize      = 10_000_000  
15 +     cacheTTL       = 5 * time.Minute  
16 +     cacheTtlStaging = 1 * time.Second  
  
17 + )  
18 +  
19 + type cacheDecorator struct {  
20 +     cache      lru.Cache  
21 +     story       protection.GetStatusStory  
22 +     actualCacheTtl time.Duration  
  
23 +  
24 +     m metrics.Client  
25 + }  
26 +
```

⚠ High · const `cacheTtlStaging` should be `cacheTTLStaging` (golint)

⚠ High · struct field `actualCacheTtl` should be `actualCacheTTL` (golint)

интеграции

статус в дашборде
платформы



создание своих линтеров

Подготовка docker-образов для actions

В качестве образа для actions может быть использован любой docker image, размещенный в `avito.ru`. Для упрощения работы и указания названий образов в секциях `runner` допускаются сокращения вида `avito/<image_name>`. Такое сокращение будет развернуто в полное имя `avito.ru/avito/<image_name>`.

Для единообразия большинство actions имеют имя вида

`.avito.ru/avito/actions-<name>`, например

`.avito.ru/avito/actions-golang`.

Содержание раздела

Задачи avito actions

Термины

Хранение actions

Запись в каталог

Чтение из каталога

Использование

Команды

catalog add

catalog list

воспроизводимость сборок

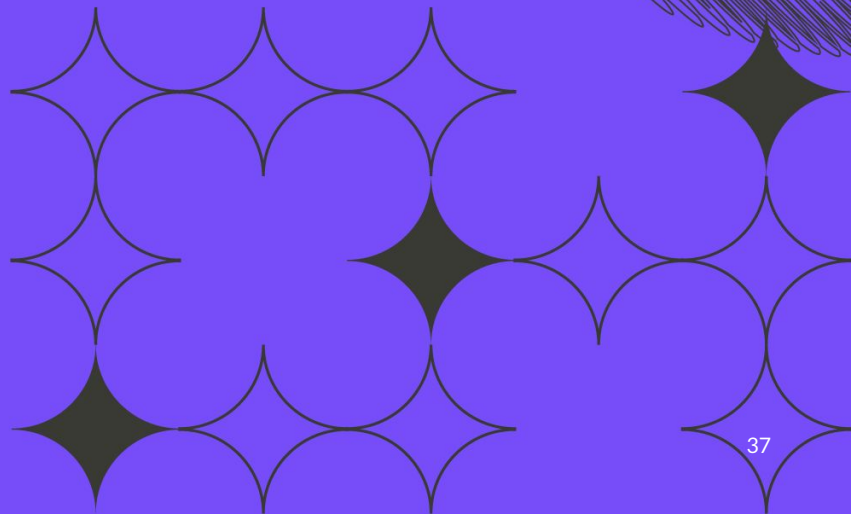
```
[[actions]]  
on = ["fmt"]  
runner="docker@avito/actions-golang:1.20"  
run = "fmt v1.53"
```

```
[[actions]]  
on = ["lint", "ci-lint"]  
runner="docker@avito/actions-golang:1.20"  
run = "lint v1.53"
```

ПОД КАПОТОМ



avito.tech



два типа runner (способа запуска)

- ❏ docker
- ❏ shell

docker

описываются
в конфигурационном
файле

```
[[actions]]  
on = ["fmt"]  
runner="docker@avito/actions-golang:1.20"  
run = "fmt v1.53"
```

docker

представляют собой
обычный docker container

```
FROM host.avito.ru/avito/actions:0.0.1 as actions
FROM host.avito.ru/avito/golang:1.20

# Order is important: from lowest to highest
ENV GOLANGCI_LINT_VERSIONS="v1.38.0 v1.39.0 v1.40.1"

RUN set -e; \
    mkdir -p /avito-actions/bin \
    go install github.com/avito-tech/golangci-lint@v1.40.1
```


docker

но следующий некоторому
набору соглашений

docker

например: наличие
информационной
метки

```
42 |  
43 LABEL ru.avito.actions.info="cat /avito-actions/info.toml"
```

docker

на основе метки мы
определяем, какие
команды доступны

```
description = "Набор actions для сервисов и библиотек, написанных на языке Go"
```

```
[maintainer]  
name = "PaaS Dev"  
contact = "~paas"
```

```
[[runs]]  
name = "lint"  
command = "/avito-actions/lint"  
description = ""  
Линтинг кода с использованием golangci-lint
```

В качестве параметра поддерживается версия golangci-lint

Утилита golangci-lint воспринимает конфигурацию
в соответствии с официальной документацией: <https://golangci-lint.run/>

СООТВЕТСТВЕННО,

- утилита **actions** делает pull образа
- монтирует в него локальную директорию
- выполняет нужную команду

shell

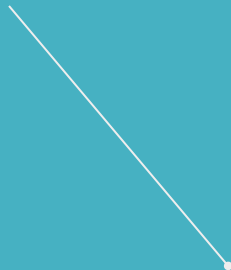
просто команда, которая
выполняется в оболочке bash

```
[[actions]]  
on = ["fmt"]  
runner="shell"  
run = "go mod tidy"
```

СООТВЕТСТВЕННО,

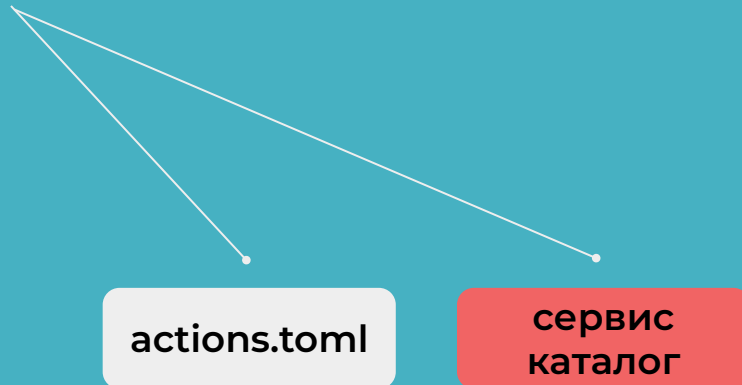
- утилита **actions** просто выполняет команду локально
- можно делать маленькие общие автоматизации

ИСТОЧНИКИ СПИСКА actions



actions.toml

ИСТОЧНИКИ СПИСКА actions



ИСТОЧНИКИ СПИСКА actions



ИСТОЧНИКИ СПИСКА actions



утилита actions

- ❖ написана на Go
- ❖ инкапсулирует логику запуска
- ❖ используется локально и в CI

что в сервисе-каталоге?

- обязательные для сервисов actions
- список доступных actions

что такое обязательные actions

- проверки платформенных конфигурационных файлов
 - манифест сервиса, конфигурация алертов и т.д.
- секьюрити-линтеры

что такое обязательные actions

языковые линтеры — только
в репозиториях сервисов
в actions.toml

КАКИЕ ЕСТЬ СЛОЖНОСТИ



avito.tech



скорость

скачивание, запуск и выполнение
в контейнере занимает время

скорость

- собираем образы под **x64** и **arm64**
- используем кэши для скачиваемых образов
- используем кэши для операций внутри

ПОНЯТНОСТЬ

не всегда пользователь понимает,
почему упал линтер

ПОНЯТНОСТЬ

- умеем собирать с линтеров результаты в формате checkstyle
- показываем саммари выполнения
- стараемся добавлять подсказки о причинах падения

КАКИЕ ЕСТЬ ПРЕИМУЩЕСТВА

стандартизация

единый способ запуска
и конфиг у всех команд

повторяемость

зафиксированы версии
запускаемых линтеров

декларативность

конфиг описан
в декларативном виде

распространение изменений

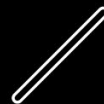
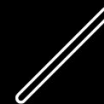
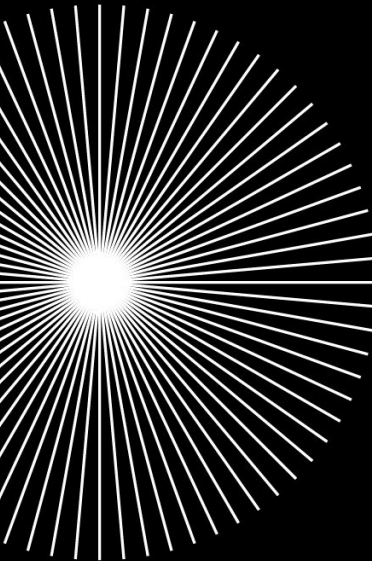
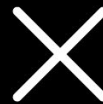
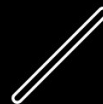
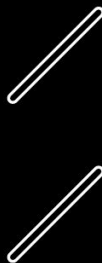
наличие общего каталога позволяет
нам менять списки обязательных
actions и обновлять их



ВЫВОДЫ



avito.tech

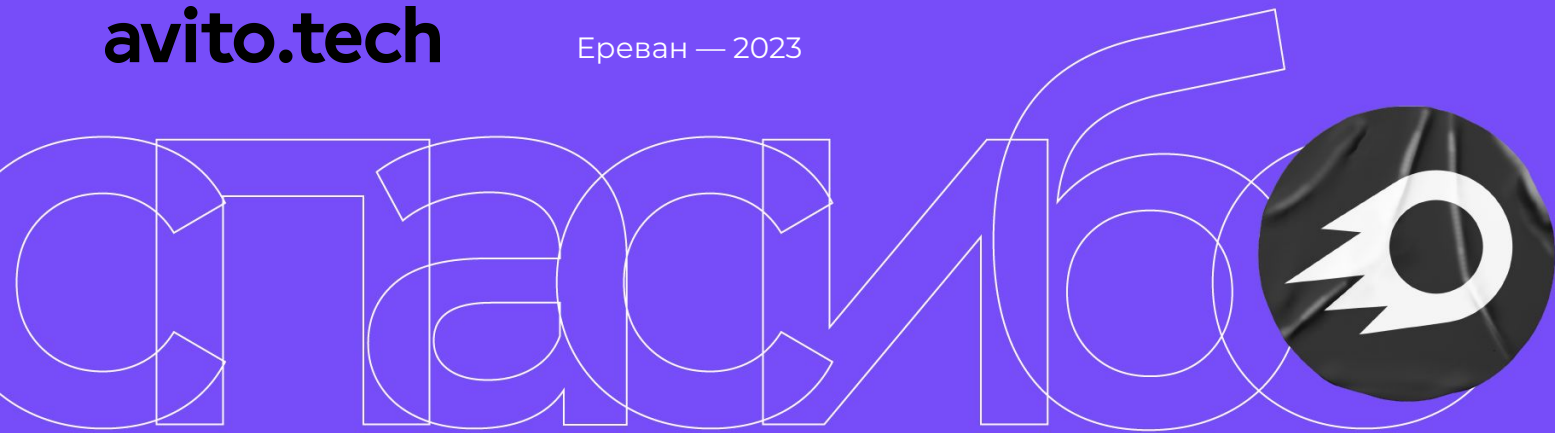


ВЫВОДЫ

- сделали возможным появление новых линтеров без боли
- продуктовые разработчики тоже могут контрибьютить свои решения
- можем централизованно управлять стандартными для всех линтерами
- не завязаны на какую-либо внешнюю технологию

avito.tech

Ереван — 2023



**Павел
Агалецкий**

ведущий инженер

psagaletskiy@avito.ru

