

Frontend Kotlin · from the trenches

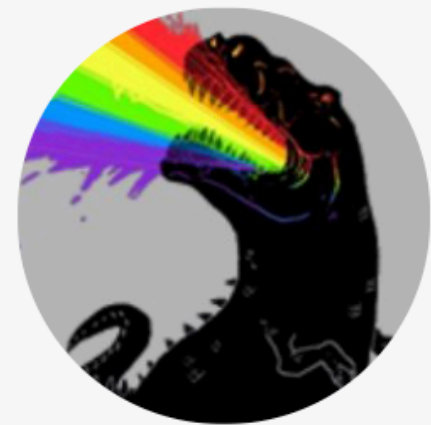


KotlinConf

context



Gaëtan Zoritchak
[@gz_k](#)



Pierre Mariac
[@imtam5](#)



Nicolas Baldran
[@azertypow](#)



Data2viz
[@data2viz_io](#)

The need of an efficient dev workflow

- ☐ build
- ☐ debug
- ☐ test
- ☐ optimisations

dead code elimination

JVM

minification

JS

test

JVM

JS

distribution

JVM

JS

kotlin2JS

JVM

bundle

JS

debug

JVM

JS

CSS processing

JS

dependency management

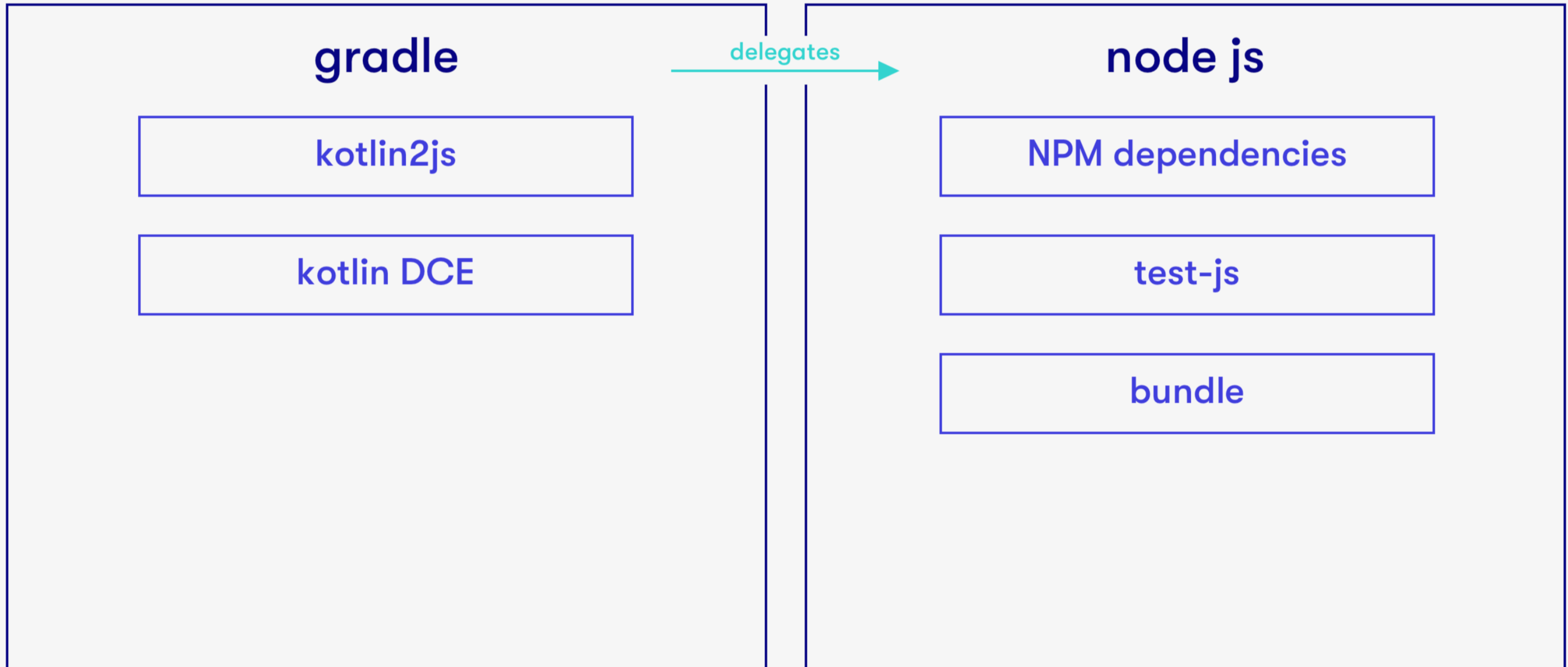
JVM

JS

Build with Gradle...

- +++ Best pace for new features releases (DCE,... , IC)
- ++ performances
- ++ easier to adapt to a complex build
- lots of releases





Let's see it in action...



build

- gradle
moduleKind = "umd"



debug

- kotlin sources
sourceMap = true
sourceMapEmbedSources = "always"



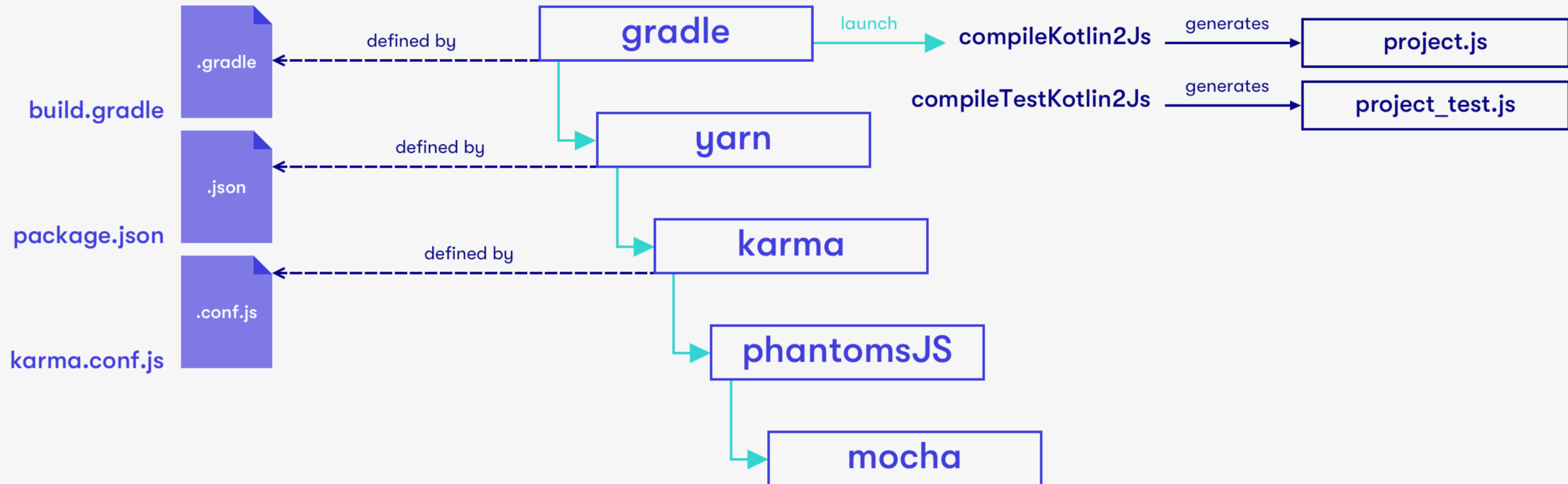
test



optimisations

Tests

#1 Js tests



✓	build	— gradle moduleKind = "umd"
✓	debug	— kotlin sources sourceMap = true sourceMapEmbedSources = "always"
✓	test	— karma + mocha — multiplatform
○	optimisations	

Release

- Dead Code Elimination
- Webpack

✓	build	— gradle moduleKind = "umd"
✓	debug	— kotlin sources sourceMap = true sourceMapEmbedSources = "always"
✓	test	— karma + mocha — multiplatform
✓	optimisations	— DCE — webpack + uglifyJS + compression

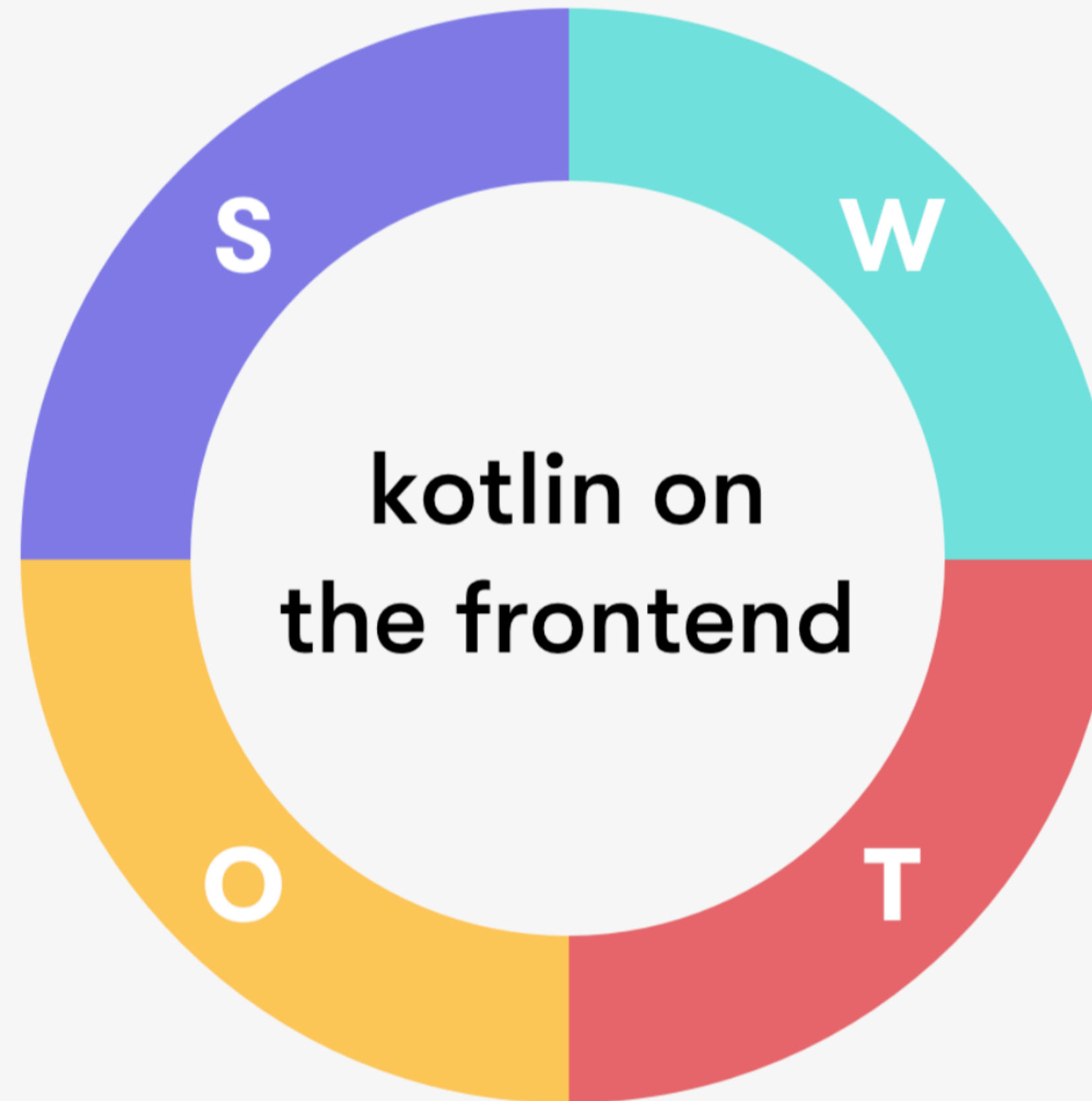
Feedback on kotlin.js performances.

strengths

- multiplatform
- tooling
- DSL

opportunities

- kotlin community
- webassembly



weaknesses

- heavier process
- JS interoperability

threats

- JetBrains JS focus
- no kotlinJS ecosystem

Thank you

follow us on <https://github.com/data2viz/data2viz>