



# Diktat: Lightweight Static Analysis for Kotlin

Andrey Kuleshov<sup>1</sup>, Petr Trifanov<sup>1</sup>, Vladislav Frolov<sup>1</sup>, Guangtai Liang<sup>2</sup>

<sup>1</sup> Russian Research Institute, Huawei Technologies Co., Ltd, Moscow, Russia

<sup>2</sup> Software Analysis Lab, Huawei Technologies Co., Ltd, Beijing, China

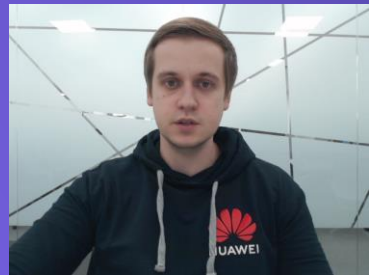
Email: {andrey.kuleshov, trifanov.petr, vladislav.frolov, liangguangtai}@huawei.com



# Intro

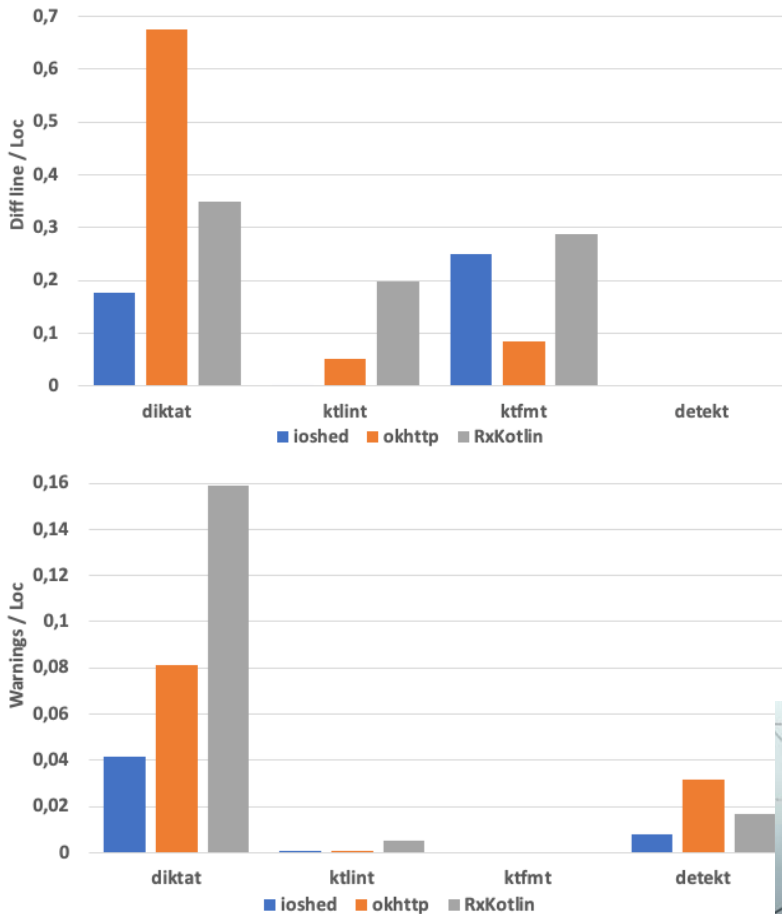
- In the paper authors reviewed their popular open-source static analyzer and automatic code fixer for Kotlin called **Diktat**\*
- Authors tried to show that a lightweight AST analysis can be very powerful in specific scenarios
- Authors observed different types of Kotlin Internal Representation that could be used for static analysis: AST, PSI, new IR
- Paper contains the comparison of existing popular open-source static analyzers for Kotlin

\*<https://github.com/cqfn/diKTat>



# Results

- In this paper authors of Diktat reviewed their way of solving industrial problems related to the code analysis in Kotlin language
- On the real-life scenarios authors described problems when a lightweight AST analysis can be successfully used for detection and automated elimination of code issues
- This paper contains experiments with popular open-source static analyzers applied to mature and popular open-source libraries
- Results have proved that Diktat inspections detect and fix the number of issues higher than average



# THANK YOU

Follow the project updates:  
<https://github.com/cqfn/diKTat>

To check the demo visit:  
<https://ktlint-demo.herokuapp.com>

