

◀ Kotlin 1.4 Online Event

Opening Keynote

@kotlin
@jetbrains

October 12, 2020



Speakers



Egor Tolstoy
Product Management



Stanislav Erokhin
Development



Svetlana Isakova
Developer Advocacy



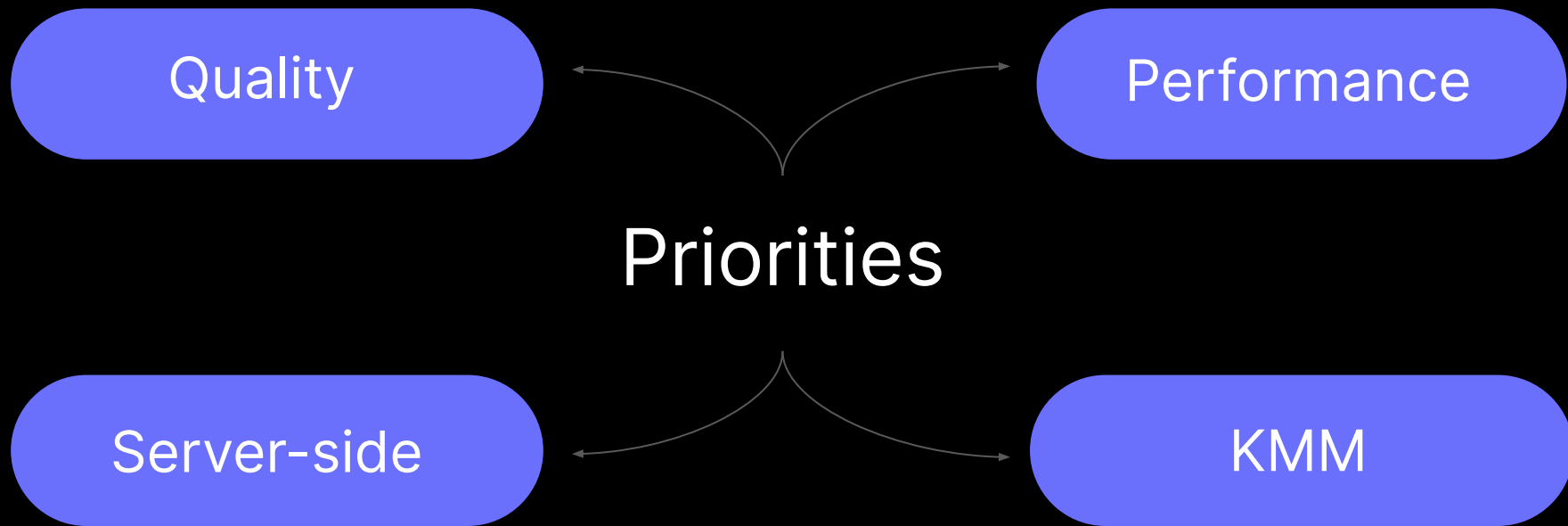
Roman Elizarov
Language Research



Anastasia Kapanina
Product Manager for
Kotlin Multiplatform Mobile

◀ Kotlin 1.4.10

Kotlin 1.5 spring 2021



Kotlin ecosystem = people + ideas + software



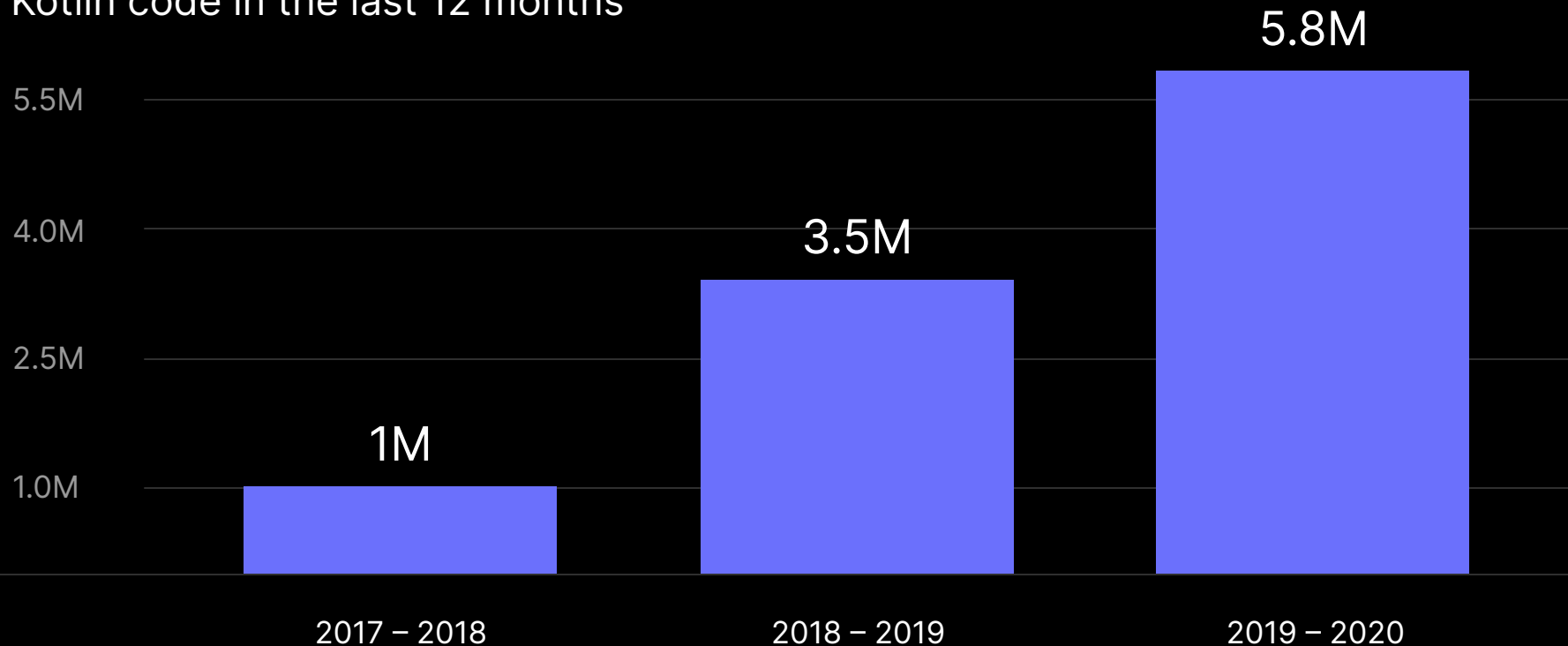
Community Growth

Egor
Tolstoy

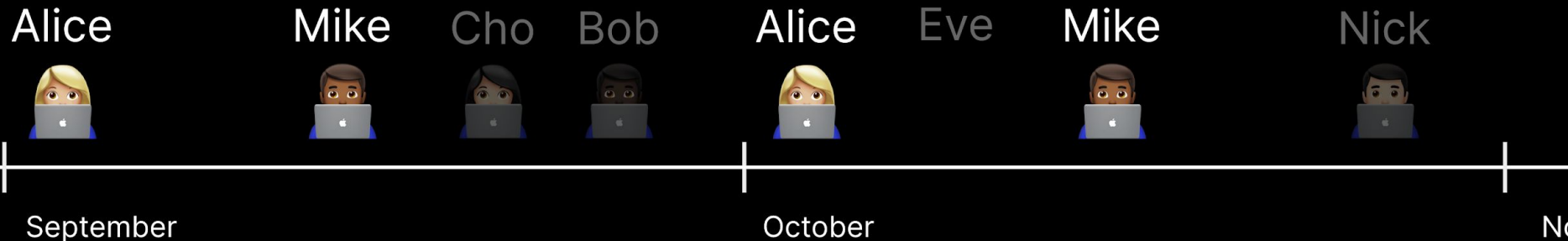
@igrekde

Total yearly users

Number of people who have edited
Kotlin code in the last 12 months



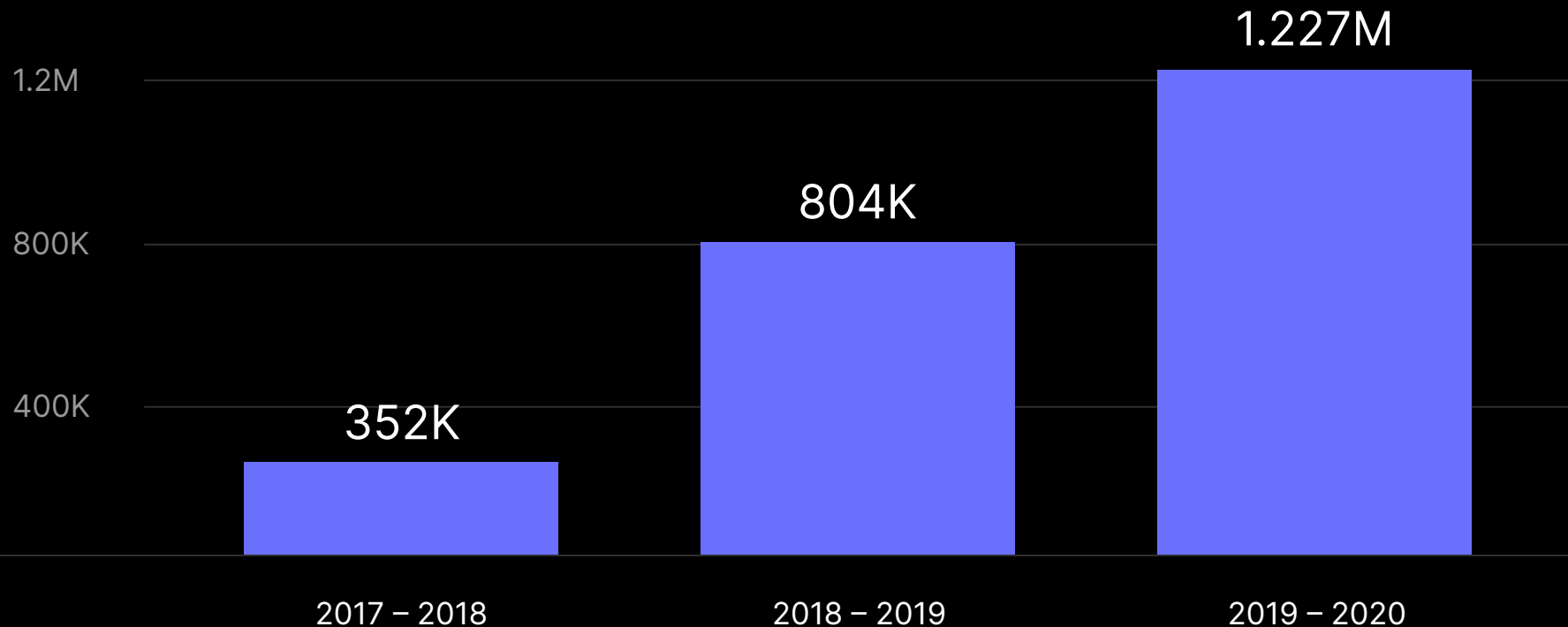
Active Kotlin users



An active user is a developer who uses Kotlin regularly.
E.g. Alice and Mike are an active user, Cho, Bob, Eve and Nick are not.

Active yearly users

Number of people who regularly edit Kotlin code in the last 12 months



Community


Toshiaki Kameyama

800 PR  on GitHub


Igor Wojda

145 tickets  in YouTrack


Cedric Beust

maintainer of the Klaxon library with 1.5k 


Márton Braun

700 answers  on Stack Overflow

Florina Muntenescu

blogs about Kotlin for 15.7k followers 

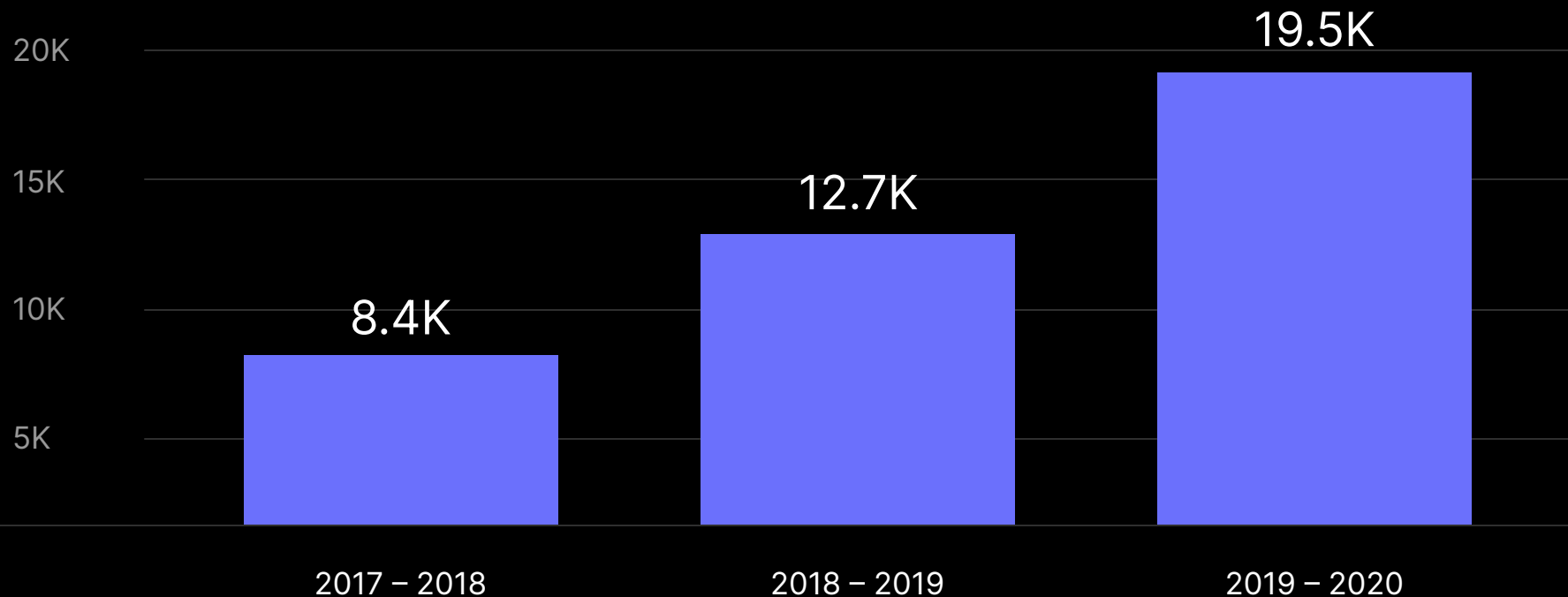
Google Compiler Team

more than 900 commits  in Kotlin compiler

And many more!

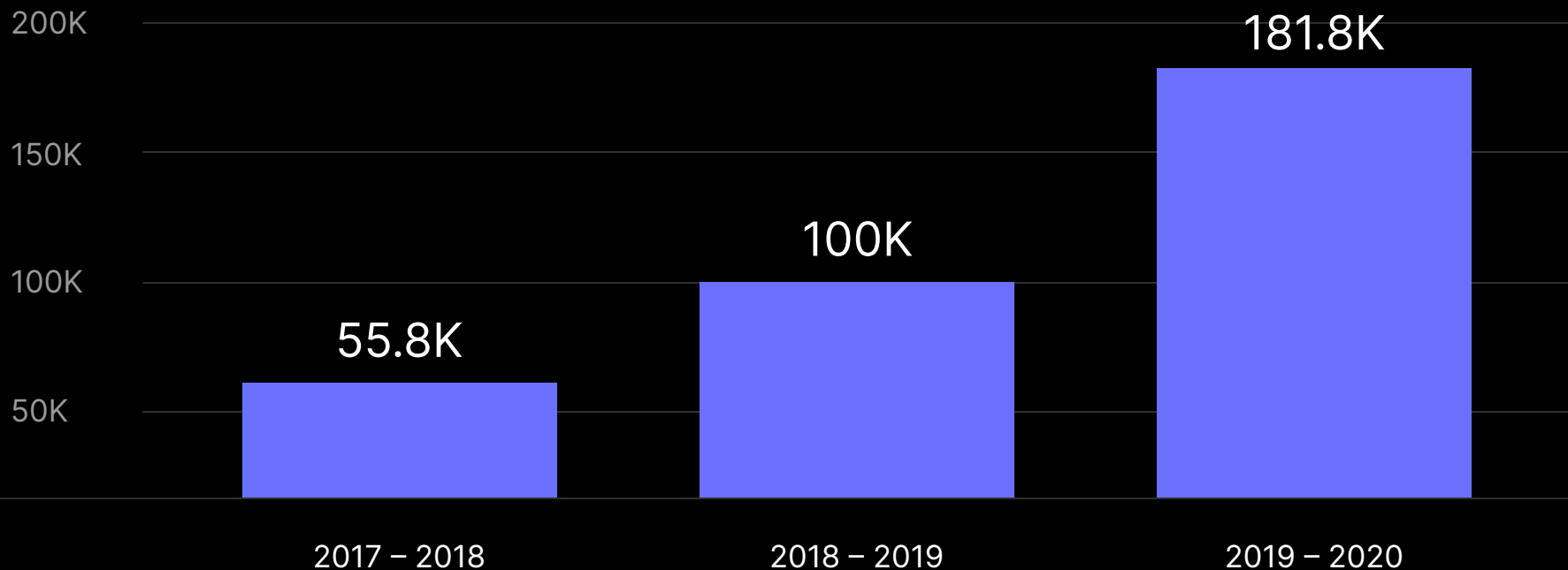
Stack Overflow growth

Answers



GitHub growth

New Kotlin projects



Community numbers

 32K Slack

 37K Reddit

 94K Twitter

 200 User groups

Contributing to Kotlin (Staying in Touch)



Egor Tolstoy
Head of Product
Management

Performance & Quality

The background features a complex geometric composition. On the left, several translucent planes in shades of blue and purple intersect, creating a sense of depth and movement. On the right, a large, solid grey triangle points towards the center, partially overlapping the translucent shapes.

State of deep focus



State of dev focus



Exceptions

Boring tasks

UI freezes

Cryptic error
messages

Complicated
documentation

Boilerplate
code

Performance
problems

Lack of
challenges

Ceremony

Exceptions

Boring tasks

UI freezes

St

Highlighting exception

c error
ages

Complicated
documentation

Boilerplate
code

Perfor.
problem

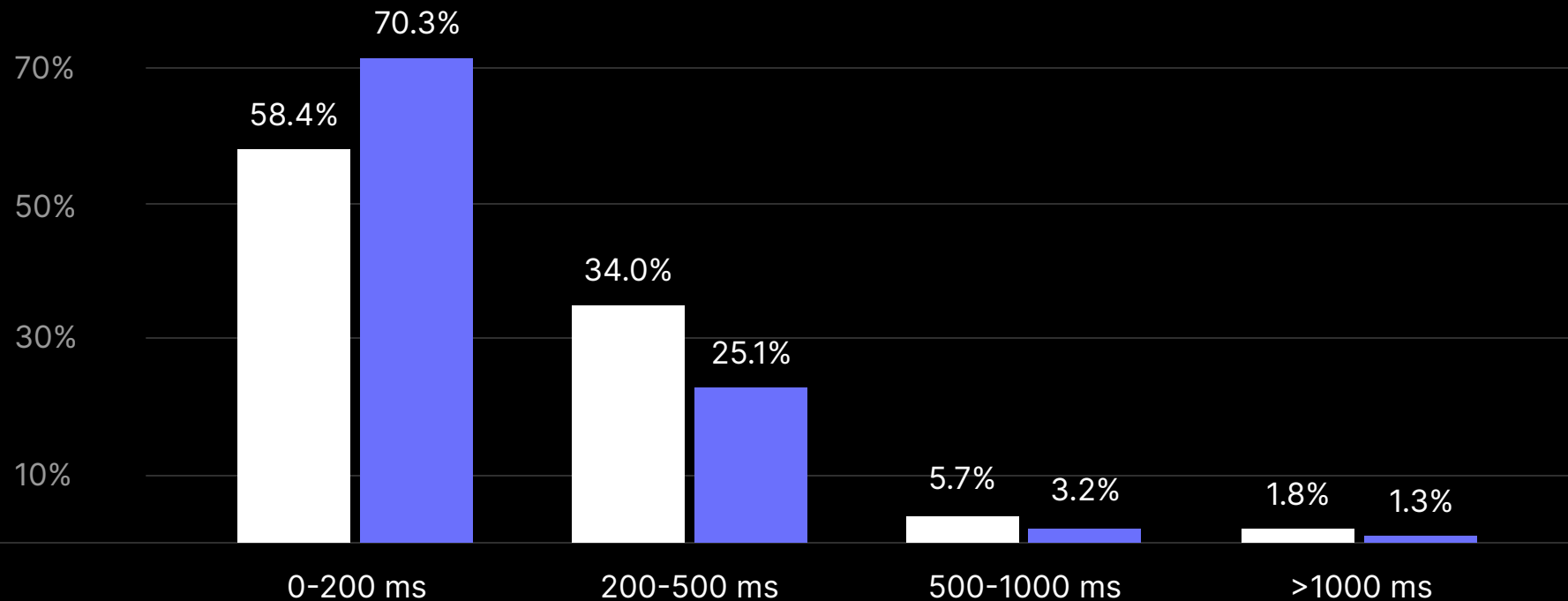
Lack of
challenges

Ceremony

Code completion speed

● Current
● Previous

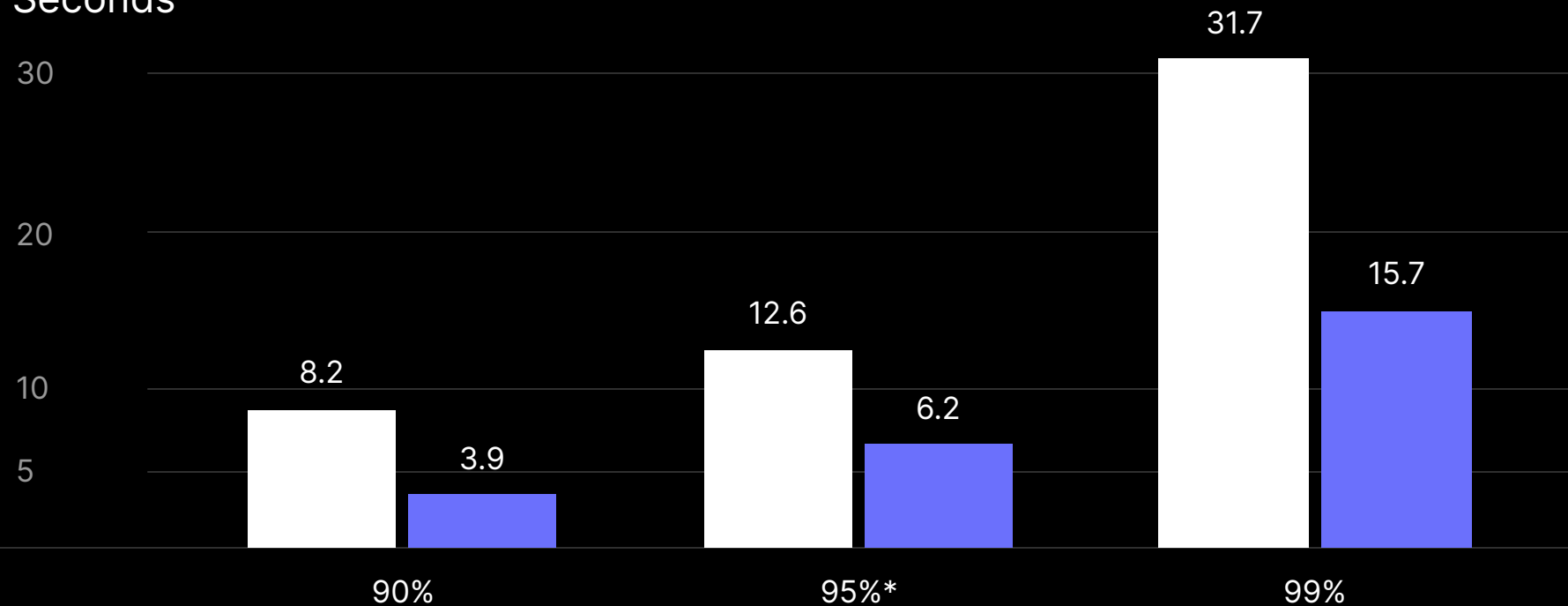
Based on anonymous usage statistics



Highlighting times in large Kotlin files

Seconds

● Current
● Previous



* – 95th percentile means that in 95% of cases highlighting time is less than 6.2s

Kotlin 1.4 in IntelliJ-based IDEs: Boosting Quality and Performance



Anton Yalyshev
Product Manager
in IntelliJ Kotlin plugin

Project import process
performance

Scripting performance
and stability

Faster Kotlin/Native
compilation

Navigation and Find
Usages speed

Quality of cross-
language refactorings

Typing / completion /
highlighting speed

UltraLight classes to accelerate
Java / Kotlin interop

Kotlin/Native disk
footprint improvements

Gradle incremental
compilation performance

Project import process
performance

Scripting performance
and stability

Faster Kotlin/Native
compilation

New compiler frontend

Navigation and Find
Usages speed

Quality of cross-
language refactorings

Typing / completion /
highlighting speed

UltraLight classes to accelerate
Java / Kotlin interop

Kotlin/Native disk
footprint improvements

Gradle incremental
compilation performance



New Compiler

Stanislav
Erokhin

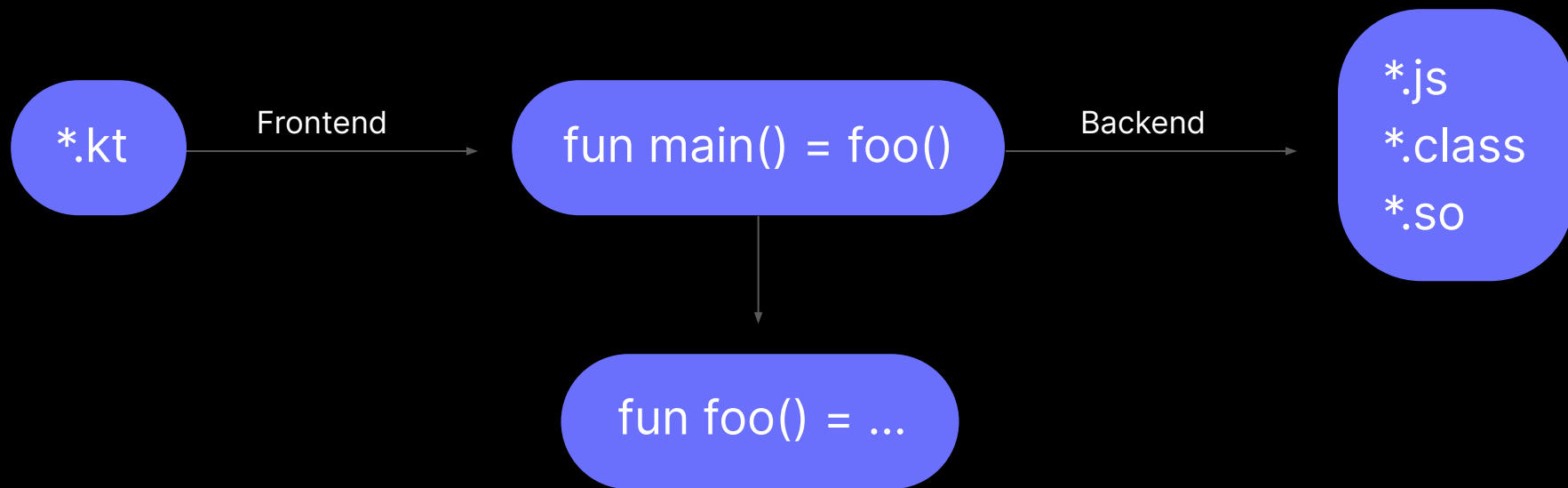
New compiler

⇒ Fast

○ Uniform

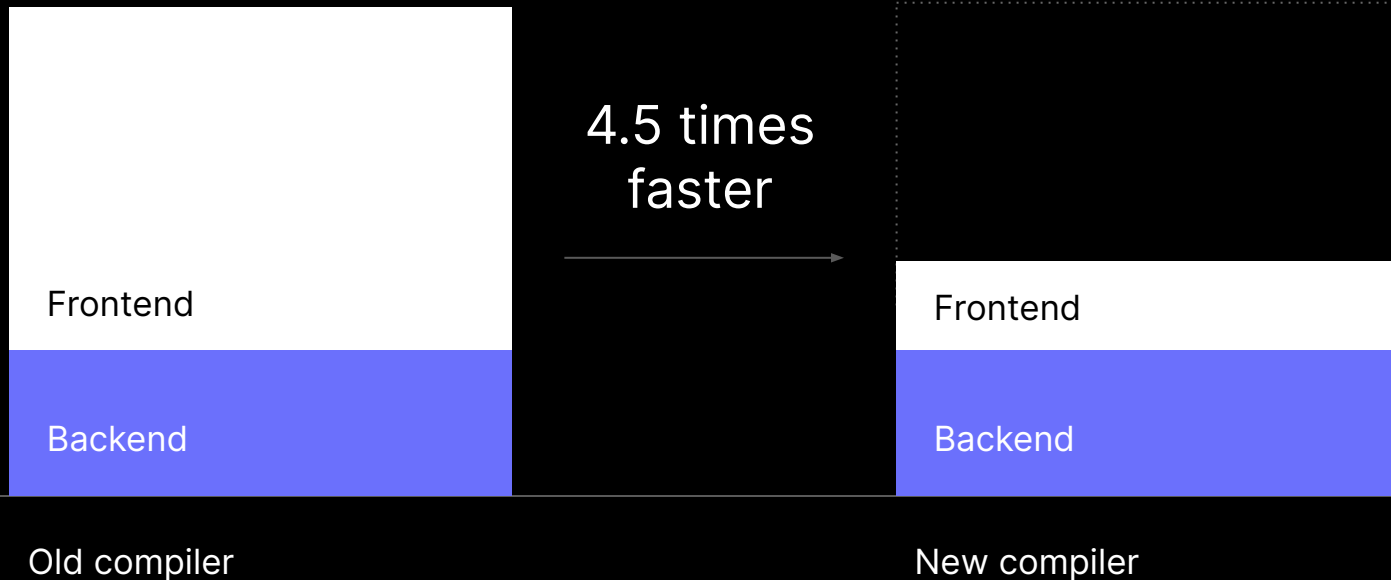
◇ Pluggable

Kotlin compiler



Expected speed improvements with new frontend

Not in 1.5



Reworked type inference

Stable in Kotlin 1.4

- Fixes 300+ existing issues
- Enables new language features
- Supports more complicated scenarios

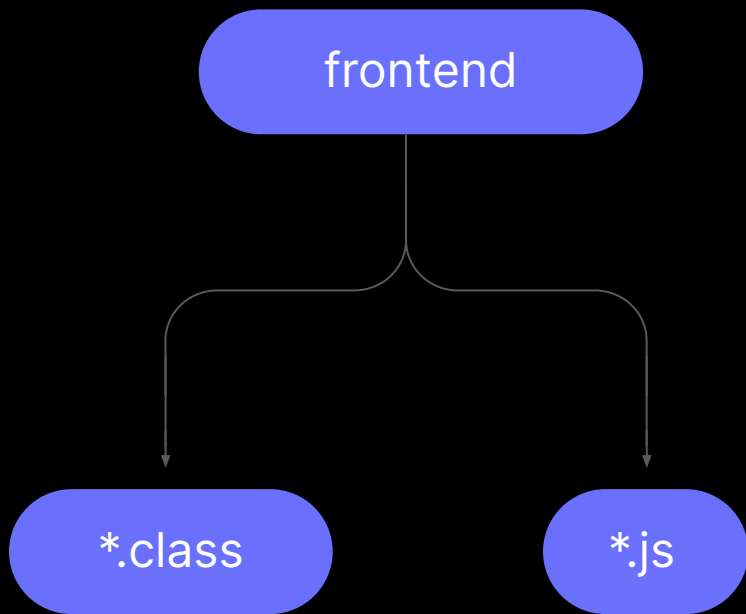
New Language Features



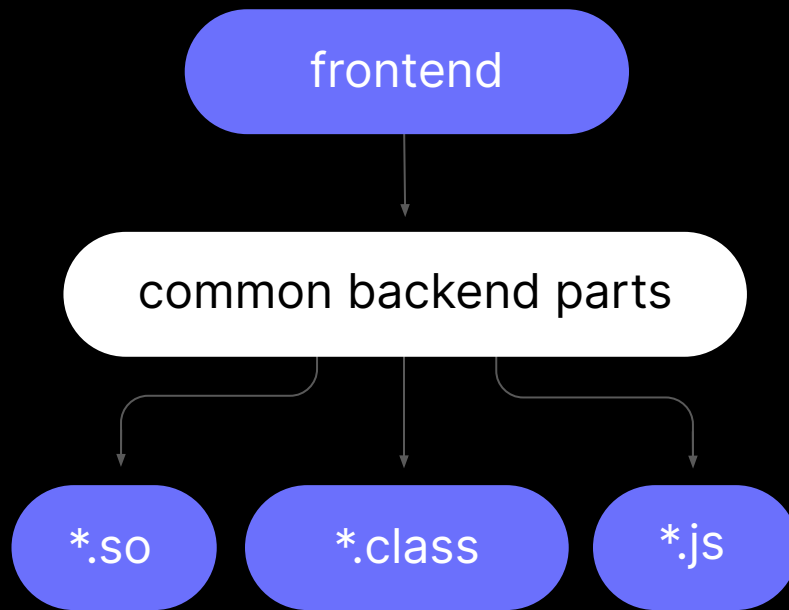
Svetlana Isakova
Developer Advocate
in Kotlin

Unified backends

old infrastructure



new infrastructure



Kotlin 1.4 backends

Kotlin/Native is already there

Kotlin/JS backend is in Alpha

Resulting JS size: up to 2 times smaller

Kotlin/JVM backend is in Alpha

used by JetPack Compose



Server-side Programming Roman Elizarov

@relizarov

◀ Kotlin 1.0:

Pragmatic Language
for JVM and Android



```
@Entity
class User(
    var login: String,
    var firstname: String,
    var lastname: String,
    @Id @GeneratedValue var id: Long? = null
)
```



Java 14

Java 15

JEP 359: Records (preview)

JEP 384: Records (2nd preview)

JEP 360: Sealed classes (preview)

Java SAM conversion

```
public interface Action {  
    void perform();  
}
```

```
acceptAction {  
    // do something  
}
```

Kotlin 1.4 functional interfaces

```
fun interface Action {  
    fun perform()  
}
```

```
acceptAction {  
    // do something  
}
```

Productive

server-side

development

with



Kotlin

Server-side Kotlin coroutines

```
@GetMapping("/{id}")  
suspend fun findOne(@PathVariable id: String): User? =  
    userRepository.findOne(id) ?:  
        throw CustomException("User not found")
```



→ kotl.in/server-side



```
fun main() {  
    val server = embeddedServer(Netty, port = 8080) {  
        routing {  
            get("/") {  
                call.respondText("Hello Ktor!")  
            }  
        }  
    }  
    server.start(wait = true)  
}
```

Kotless

Experimental

```
class Server: Kotless() {  
    override fun prepare(app: Application) {  
        app.routing {  
            get("/") {  
                call.respondText("Hello Kotless!")  
            }  
        }  
    }  
}
```

Frameworks



Runtimes



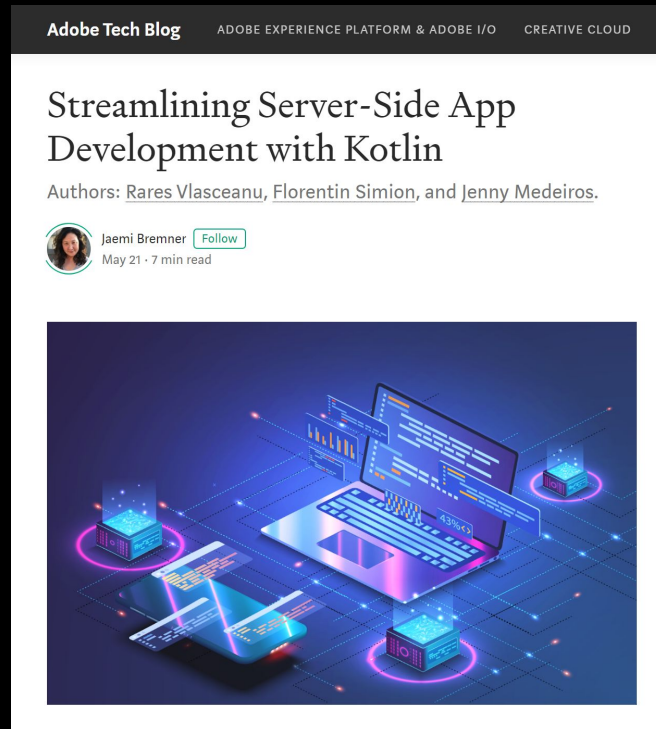
GraalVM™



Clouds



Kotlin server-side adoption



Server-Side Development with Kotlin



Anton Arhipov
Developer Advocate
in Kotlin

The State of Kotlin Support in Spring



Sébastien Deleuze
Spring Framework committer
and Kotlin at VMware

Ktor: Past, Present, and Future!



Hadi Hariri
Developer Advocate in Kotlin,
Team Lead in Ktor

Serverless Development with Kotlin



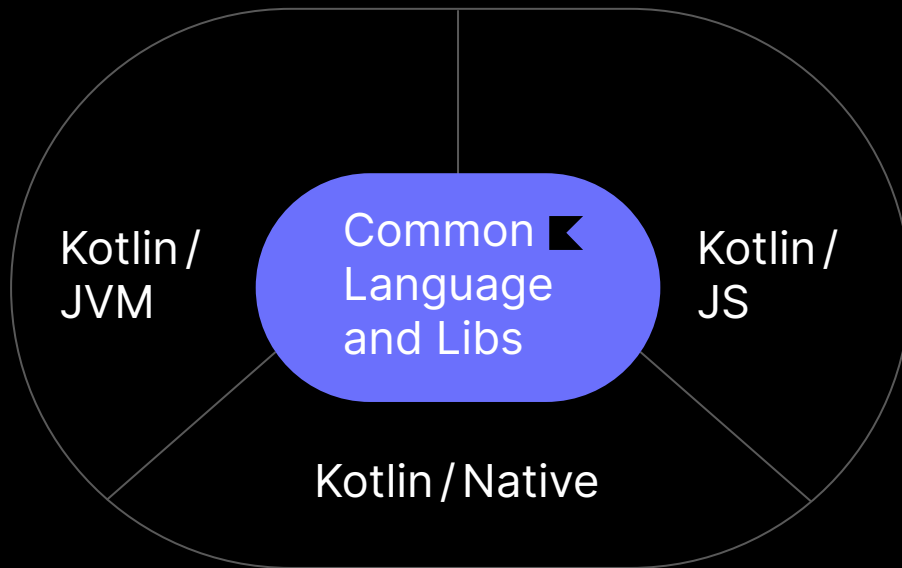
Vladislav Tankov
Team Lead
in Kotless



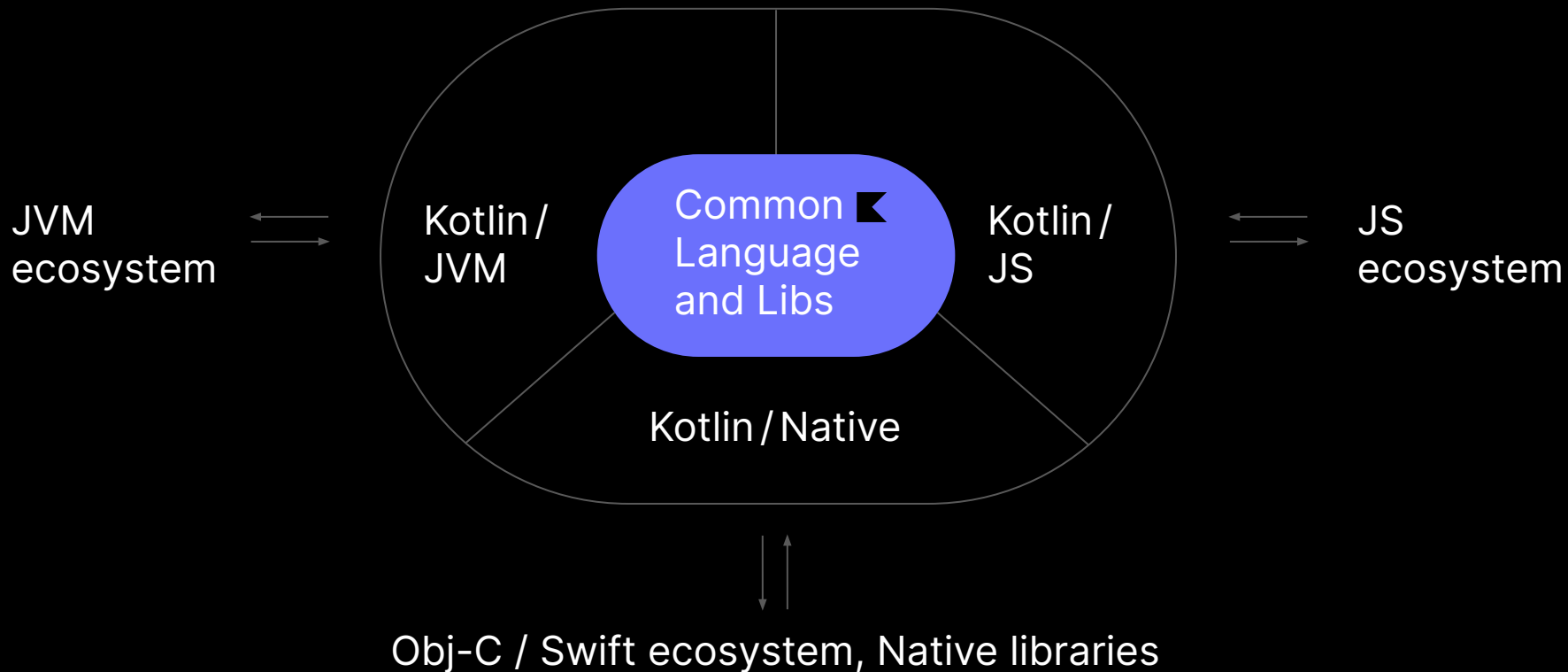
The MPP Vision Svetlana Isakova

@sveta_isakova

Kotlin Multiplatform



Kotlin Multiplatform



Multiplatform core libraries

Stdlib

Coroutines

Serialization

DateTime

Ktor



Svetlana Isakova

News From the Kotlin
Standard Library



Seva Tolstopyatov

Coroutines Update



Leonid Startsev

kotlinx.serialization 1.0



Ilya Gorbunov

Introducing DateTime



Hadi Hariri

Ktor: Past, Present,
and Future!

10%
MPP

77%
JVM

13%
Android, JS,
Native,
Unknown



Kotlin usage in IntelliJ IDEA


Here we don't count
Android Studio

Kotlin/JS

The background features a series of overlapping, semi-transparent geometric planes in shades of blue and purple. These planes are arranged in a way that creates a sense of depth and perspective. A large, solid grey area occupies the right side of the image, contrasting with the vibrant colors of the planes.



Space^{Beta}

The integrated team environment
by  JetBrains

Kotlin DSL for styled-components

Ktor

Exposed

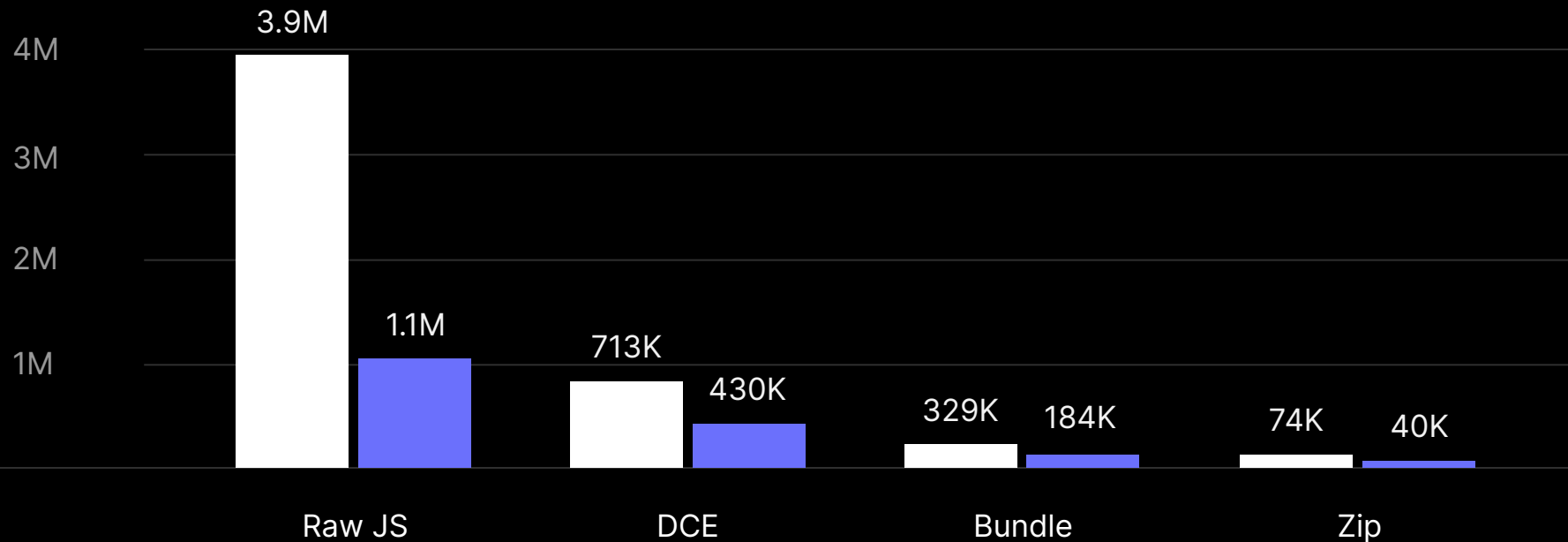
Kotlin-React

New Kotlin/JS Compiler Backend

● New JS Backend
● Old JS Backend

Improved interoperability with TypeScript / JavaScript

Decreased size of generated binaries



Community frameworks & libraries



WebAssembly + Kotlin

Prototype in progress



Kotlin/JS in 1.4 and beyond



Sebastian Aigner
Developer Advocate
in Kotlin

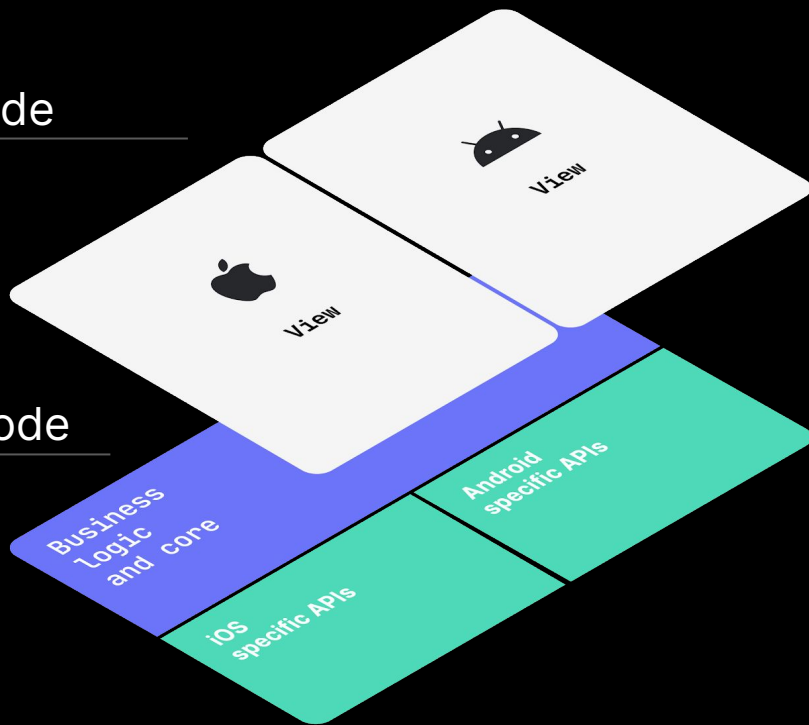


Kotlin Multiplatform Mobile

Anastasia
Kapanina

@a_kapanina

Native code



Shared code



Kotlin

Multiplatform

Mobile:

an

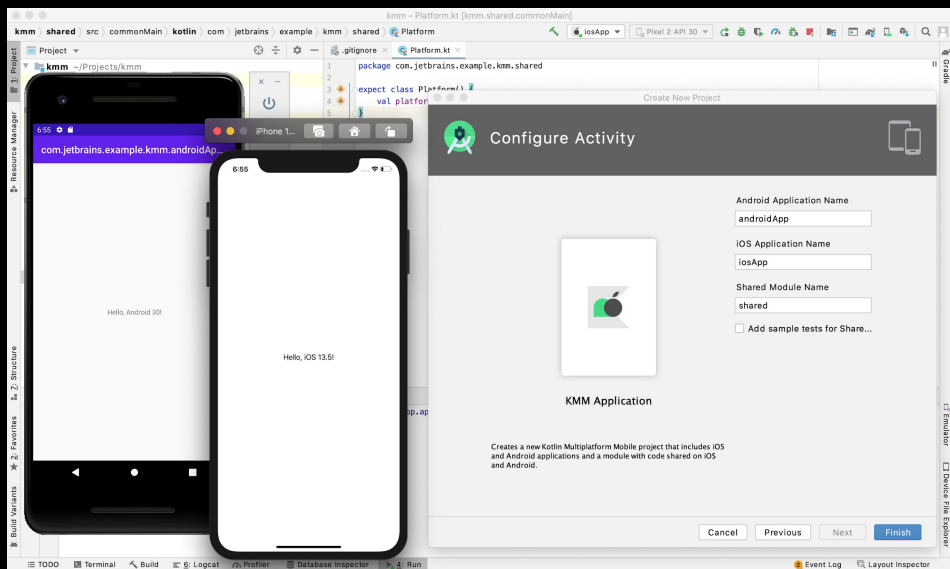
SDK

for

cross-platform

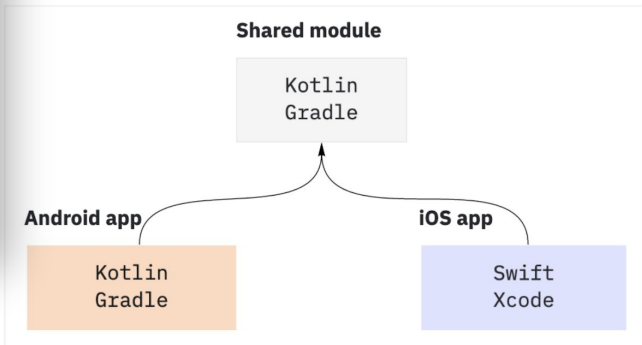
mobile

development



- Event Log
- Layout Inspector
- Organize a process around KMM
- Connect to platform-specific APIs
- Configure SQLDelight for data storage
- Use Ktor for networking
- Add dependencies to KMM modules
- Publish KMM apps
- Work with concurrency
 - Concurrency overview
 - Concurrent mutability
 - Concurrency and coroutines

Root project

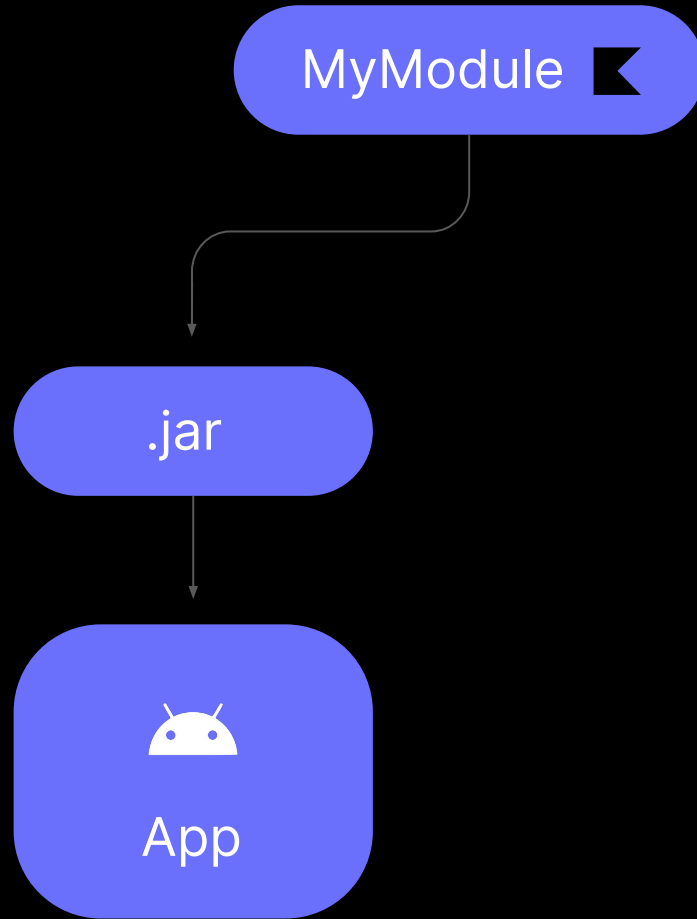


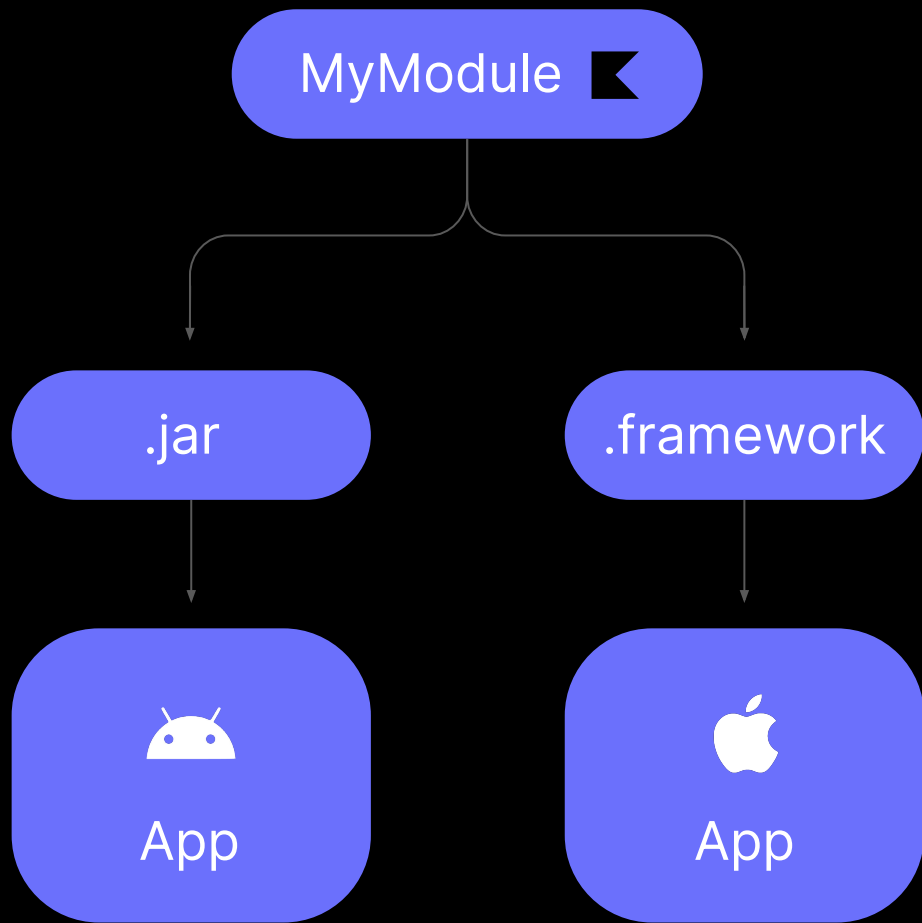
This is the structure of a KMM project that you create with a Project Wizard in IntelliJ IDEA or Android Studio. Real-life projects can have more complex structure; we consider these three components essential for a KMM project.

Let's take a closer look at the basic project and its components.



App





 **AUTODESK.**

Quizlet

TOUCHLAB

MIREGO

vmware®

 **chalk**

 **Hue Essentials**

ICEROCK

Yandex

KMM Alpha → Beta

Faster Kotlin/
Native compilation

Gradle
improvements

Onboarding
experience

Better
interop UX

KMM Alpha → Beta

Better error
messaging

New Kotlin/Native
memory model

Faster completion
and highlighting

Common tests
on local JVM

iOS interop
improvements

Kotlin 1.4 in Android



Florina Muntenescu
Android Developer
Advocate in Google

Diving into Kotlin Multiplatform



Dmitry Savvinov
Team Lead in Kotlin
Multiplatform Mobile

It's time to Kotlin Multiplatform Mobile



Ekaterina Petrova
Developer Advocate
in Kotlin



Wrap-up Andrey Breslav

@abreslav

Kotlin developers

Back-end / Server

Front-end / Mobile / Web

Cloud

Data Science

And more

Summary

Performance and quality

New compiler

Server-side

KMM

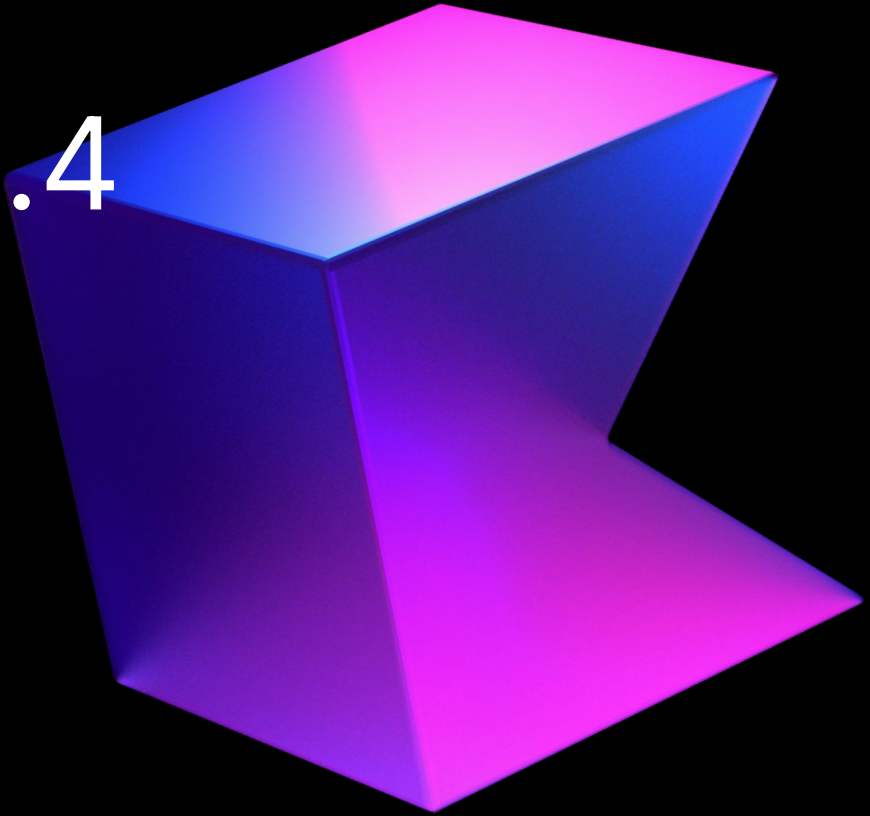
→ kotl.in/roadmap

A Look Into the Future



Roman Elizarov
Team Lead in Kotlin
Language Research

Welcome to the Kotlin 1.4 Online Event!



@kotlin
@jetbrains