

# SERGEI MAKHINOV

Android Developer

📧 [t.me/sermah](mailto:sermah@bk.ru)   [github.com/sermah](https://github.com/sermah)   ✉ [sermah@bk.ru](mailto:sermah@bk.ru)

## WORK EXPERIENCE

---

### Yandex

April 2024 - Present

*Android Developer*

*Moscow, Russia*

- **Stack:** Kotlin, Coroutines, Exoplayer, Mockito+Robolectric;
- **Features:** Video player development based on Exoplayer, multi-module architecture;
- Developing and maintaining **short video feed SDK**, as well as the basic video player library;
- Implemented a multi-platform network bandwidth calculation solution, which reduced video loading interruptions by 5-30%, and **improved accuracy and stability** of measurements;
- Added the ability to disable audio preloading for silent videos, thereby **reducing traffic consumption** by 5-10%.
- Added asynchronous initialization capability for short video feed SDK, and laid the foundation for freeing the main application thread from the player, which **will improve performance in the future**;
- Modified the player SDK architecture to **speed up video startup** by reusing the same player;
- Maintain and improve the **demo application** for more efficient testing of short video feeds, as well as for use by other teams;

### SberDevices

March 2023 - April 2024

*Android Developer*

*Moscow, Russia*

- **Stack:** Kotlin, Dagger, Coroutines+Flow, Mockk+Robolectric;
- **Features:** Multi-module architecture, working with system applications;
- Developed **system services and applications** for *SaljutOS* (Android-based);
- **Wrote services and APIs/SDKs** for various interactions - remote control input, open application tracker, detection of specific applications in the system;
- Used **inter-process communications (AIDL)** when working with services, studied the nuances of their operation and features of backward compatibility implementation;
- Optimized parsing of **JSON** config files for system applications, accelerated processing from 4 seconds to 200ms;
- **Researched Android source code** to understand the operation of system network settings;
- Wrote **Unit tests** for my own and others' code using Mockk, increased coverage of one module of 2000+ lines from 40% to 80%+;
- Automated **coverage measurement** of projects with **Unit tests** using Kover, which allowed any developer to quickly assess code coverage;
- Created rules for the **static code analysis** tool Detekt, including one that prevents outputting verbose messages to logs.

## PROJECTS

---

**VK App** | Kotlin, Jetpack Compose

Jan. 2024 - Feb. 2024

🔗 *Github* | *Minimal VK client in Material3 style*

- Stack - Kotlin, Jetpack Compose, Flow, Glide;
- **Implemented infinite scroll** for user wall/feed with loading;
- **Customized scroll mechanism** to compress profile header into an AppBar-like element;

## TECHNICAL SKILLS

---

**Programming Languages:** Kotlin, Java, Go, JavaScript;

**Technologies:** Dagger, Retrofit, OkHttp, Room, Mockk, Mockito;

**Frameworks:** Jetpack Compose;

**Tools:** Gradle, Git, Jira, Bitbucket, Gitlab;

## EDUCATION

---

**MIREA - Russian Technological University**

2021 - 2025

*Bachelor's degree in "Information Systems and Technologies".*

*Moscow, Russia*