Louis CAD

Kotlin Multiplatform and Android expert, focused on reliability

Email: louis.cognault@gmail.com
Phone: Ask via email
Currently located in Tours, France
See my work on GitHub: github.com/LouisCAD
See my Twitter @Louis_CAD

♥ What I love - values

At work, on projects, and generally speaking

Innovating, ideating, and contributing to what I believe in.
Simplifying complex things
Questioning the status quo
Expanding my knowledge, and sharing it back
Spreading improvement (and scaling it with software where applicable and appropriate)
Keeping relationships and people healthy
Technical things, so long they are useful
Ensuring attention is paid to the important details
Efficiency (energy, time)
Helping others, in my social circle and beyond
Having fun while doing all of the above

{} In software development

API design
OSS projects contributions and authoring
Pair-programming and collaborating
Reliability, efficiency, and maintainability
Inspiring people (developers or not)
Well designed programming languages like Kotlin, Swift, and Rust

Activities/hobbies

Science (learning and using)
Understanding people in depth (sometimes)
Recumbent cycling (that's also how I commute)
Lego technic (custom builds)
DIY (including electronics)

💼 Experience

Open source authoring

Splitties (set of mini libraries for Android & Kotlin Multiplatform development), 2.4K stars on GitHub
refreshVersions, Gradle plugin, almost 1.6K stars
CompleteKotlin, 265 stars since release in June 2021
beepiz-external-control (IPC on Android)
BleGattCoroutines (Bluetooth Low Energy GATT client for Android), 448 stars
Draft projects: CameraCoroutines, BleScanCoroutines (draft)

2014-2022 Android app developer at Beepiz (fulltime)

Technologies used

Java 7 (mostly until 2016)
Kotlin (starting v1.0.4)
Kotlin coroutines (since 2017)
Gradle (including Gradle plugin development)

Tasks

Made 7 production apps (alone)
Ensuring the reliability of 2 life critical apps
Leveraging accelerometer data across many different devices
Managing/replacing self-induced legacy code
Providing a user permission guarded IPC API for customer Android apps to control ours
Keeping Beepiz energy-efficient

🎤 Public speaking

Gave 27 talks (17 in 🇬🇧 and 10 in 🇫🇷) on Kotlin, coroutines, Android, and WearOS between 2017 and 2023 at conferences like Droidcon, Android Makers, DevFest, Talking.kt, Mobilization (PL), and FOSDEM, and at various meetups
Participated in 3 podcast episodes: Android Leaks twice, and The Developers Bakery once

★ Achievements

Issue reporting

Reported 202 Kotlin issues on YouTrack, with 126 resolved to date
Reported 295 issues on Google's/Android's issue tracker, with 81 fixed to date

Recognition

Google Developer Expert for Kotlin since 2020

Skills (general)

Communication

🚩 Languages

Fluent in English, French native
Good at spotting misspellings and typos (incl. own)
Comfortable speaking publicly

Presentation tools

Mind mapping tools
Keynote with animations
Google Slides

Vector graphics

Familiar with Sketch (macOS app)
Affinity Designer (light user so far)
Can write SVG path data by hand

3D modeling

Familiar with Shapr3D (iPad & macOS app), with it, I can model the Kotlin logo in under 30s
Getting started with Fusion 360

📐 Skills (software dev)

Concepts

Binary, hexadecimal representation
Backwards and forward compatibility
Code reuse
Code readability - mental load
Code and systems malleability
Maintainability
Migration
Defensive programming

Kotlin

Coroutines (in depth)
Multiplatform library development, including Kotlin/Native
Scripting, partial and full automation
Gradle plugin development (build automation)

Android

Specific skills

Making entire mobile apps
Making WearOS apps and WatchFaces
IPC (inter-process-communication)
Extensive knowledge about Android dev history

APIs I'm familiar with

Jetpack Compose
android.view (programmatically)
BLE (BluetoothLeScanner & BluetoothGatt)
SensorManager
CameraX & Camera2
Most of AndroidX
Several/many other Android platform APIs

Experimenting with

Swift, and Kotlin/Native interop
iOS dev, including programmatic UIKit
Compose HTML, and Compose for Desktop

Made in Kotlin with Compose for Web
See the code and the Android version on
github.com/LouisCAD/ResumeComposition

Generation date: 2023-08-30