Real-Time Location Insights: Leveraging Bright Diagnostics for Superior User Engagement

Author

Tharun K Nallamothu

Senior iOS Software Engineer, Microsoft, USA

Abstract : Contemporary mobile applications need reliable and ongoing telemetry gathering for personalized experiences and data-driven business decisions. Custom infrastructure is a traditional approach that can lead to resource inefficiencies, performance issues, and maintenance complexity across several platforms. This kind of approach provides fine-grained control but tends to have scalability restrictions, uneven data quality, and higher development overhead. The BrightDiagnostics SDK is a cross-platform, modular, and lightweight solution aimed at simplifying high-fidelity telemetry gathering on Android and iOS, as well as tvOS devices. Real-time sensor data acquisition integrates with smart middleware to maximize granularity and efficiency. A secure data transmission layer guarantees safe delivery of the data, and a solid backend ingestion engine provides real-time location-based promotion triggers and attribution. Influenced by industry best practices such as OpenTelemetry and mobile observability frameworks, BrightDiagnostics merges established paradigms with cutting-edge telemetry management methods to enable developers to keep data collection policy in their control, ensure compliance with privacy, and concentrate on monetization and user retention. The SDK also offers actionable information via the use of sophisticated time series analysis and visualization of GPS accuracy variations.

Keywords- Bright Diagnostics SDK, OpenTelemetry, iOS, tvOS, Optimize Granularity, Monetization, User Retention

DOI: 10.21590/ijtmh.2024100103

Introduction

An SDK is a set of tools, libraries, APIs, and documentation that allows developers to make applications for a particular platform. Apple's iOS SDK is meant to assist developers in building apps for iOS and iPadOS, ranging from compilers and simulators to interface builders. It encompasses frameworks such as UIKit, SwiftUI, Core Data, and ARKit, and can be accessed through Xcode.

The tvOS SDK is a special build of the iOS SDK for Apple TV applications, with remote control input replacing touch and giant-screen display. It contains frameworks like UIKit/SwiftUI, adapted for tvOS user interfaces, and AVKit for video playback. Google's Android SDK is Google's development kit for developing native Android applications, with emulators, debuggers, development tools, Android Studio integration, and Java/Kotlin APIs. It covers subjects like user interface, background services, sensors, and connectivity.

Cross-platform and shared code structures are creating shared codebases (JavaScript, Dart, and C#) for iOS, Android, and sometimes tvOS with the help of React Native, Flutter, Xamarin, and Cordova. Native SDK methodologies offer optimal performance and full access to platform features and are ideal for usage such as mobile applications with BLE support. The importance of the use of official SDKs is that they assure optimal performance, conformance, and access to hardware features for high-performance platform-specific tasks. Cross-platform frameworks may speed development towards a broader deployment with common logic [1].

Location data is essential for apps to enhance user experience through useful features and information. Apps can vary recommendations based on movement tendencies by examining user behaviors. Nearby recommendations are also available, enabling users to locate nearby venues or points of interest. Real-time information is available, taking into account traffic and other factors. Route optimization is enhanced through the estimation of trip duration and path optimization based on location information. Location services such as ride-sharing and delivery monitor drivers and deliveries, offering users notifications and estimated delivery times. Users can use location information to find matches in dating applications and social media, find friends, and discover activities around them. Convenience functions involve geofencing for providing location-based notifications and whereabout sharing for real-time interaction. User experience analytics and data collection are also necessary, with apps interpreting location information to discern user behavior and areas for enhancement [2]. Tools such as Fulcrum and FastField are employed for data collection and location analysis.

Excel spreadsheets, form builders, and survey software do not have the functionality to collect quality field data. Researchers, data collectors, inspectors, and stakeholders might have to capture time-stamped pictures and videos, GPS coordinates, and complete questionnaires prior to lawfully approving and submitting the form. They might also have to collect data in the field where there is no internet. For automating data collection process, use top five data collection software that can facilitate your team in collecting data faster, reduce errors, and increase productivity [3].

By connecting your data collection program to Zapier, you can send the data to any database program you prefer. The top five tools used for data collection are Geolocation fulcrum with customized maps, FastField for overall usability, Jotform for form creation, KoboToolbox for free data collection, and Fluix for building complex processes. These tools will make it easier for your team to collect data faster, reduce errors, and make more out of your time [3].

With the current data-oriented world, applications that capture true location and diagnostic data are essential. The Bright Diagnostics SDK to enable seamless run of such applications. It provides high-resolution GPS, Wi-Fi, cellular signal, device health, battery status, error logs, and others, serving as a single data source tool for source and partner applications. By having control over data gathering and when, Bright Diagnostics liberates app developers from data collection hassles so they can concentrate on user experience and monetization [4].

Bright Diagnostics SDK is a system that is meant to ease the process of acquiring location and diagnostic data for apps by the source and its partners. It offers one platform for getting data for these apps, with massive amounts of precise location data. This enables developers to concentrate on monetizing applications through data handling, thus saving time and money. Bright Diagnostics provides a competitive edge in enabling developers to build and handle complex diagnostic tools independently, enabling them to derive the benefits of context-aware interactions and enhance the user experience. The SDK's easy-to-use and dependable solution enables developers to concentrate on monetizing their apps, not managing data acquisition infrastructure and performance tuning [5].

Precise and real-time positioning information is changing how businesses interact with their customers in the online world. Such apps create highly personalized and context-sensitive experiences, engaging people and enhancing business results. Nonetheless, precise location and diagnostic data acquisition and management are complicated by intricate infrastructure, continuous performance tuning, and judicious attention to issues of privacy and data security. The Bright Diagnostics SDK, designed for AT&T and its ecosystem partners, offers a scalable, reliable, and effective means of acquiring thorough location and diagnostic information from applications.

Bright Diagnostics acts as a platform for consolidated data sourcing, delivering high-volume, precise location data that boosts user satisfaction and engagement. It also enables developers to concentrate on monetizing their applications and enhancing core features by handling data gathering, performance, and diagnostics, enabling faster time-to-market [4]. The SDK provides stable performance and reliability, ensuring applications are effective and responsive without sacrificing data quality.

Bright Diagnostics assists developers in fulfilling legal requirements and building user trust by incorporating best practices for data security and privacy. It is the property of more than just an instrument of AT&T and its partner network; it is a strategic asset that enables data-driven innovation. By obtaining the complete location and diagnostic information, AT&T is able to develop smarter services, improve network performance, and deliver customized experiences that differentiate them in a competitive marketplace [5].

Source and partner applications can centrally gather high-frequency data using a single platform, providing developers with pipeline management control and comprehensive data collection. This makes developers able to concentrate on monetization, for instance, location-based promotions. Developers are able to retain privacy and control by enabling applications to determine what information is collected and when [7]. Rich Telemetry Pipeline gathers data on Wi-Fi, GPS coordinates, device, cell signal, error logs, resource utilization, battery percentage, and application version. Diagnostic and customization features are easily available because of its support for Android, tvOS, and iOS. Powers solutions applications are customer service, customization, and ad attribution.

Benefits include quicker time-to-market, because developers can directly add diagnostics and telemetry without developing a custom data-collection pipeline. This enables location-based offers, more targeted marketing, and higher revenue and user activity. In-depth diagnostic data enables analytics and proactive customer support. A shared SDK for iOS, tvOS, and Android enables seamless maintenance and integration across all platforms [6].

Related Work

Android's ConnectivityDiagnostics API supports thorough network-level carrier and VPN app diagnostics, such as connection reports and data stall detection. BrightDiagnostics correlates multi-platform telemetry from multiple sources, such as GPS, Wi-Fi, cellular, battery, and app logs [9]. Location SDKs such as Roam.ai enable scalable infrastructure and faster time to market, enhancing Juniper Mist SDK for Enterprise BLE Positioning, which relies on location measurement using BLE directional antennas and machine learning to estimate sub-second indoor location based on time probability surfaces. Developer-facing patterns of LaunchDarkly Mobile SDK highlight asynchronous initialization, low startup overhead, and supporting environment metadata for targeting [9].

DJI and KPIT Diagnostics are vertically specialized mobile SDKs for device and flight diagnostic data. BrightDiagnostics' BLE-based pass-through and telemetry to backend systems are analogous to these models. Research foundations are FedLoc, CARP, and DeepLoc, which offer reactive, cross-platform sensor data collection for digital phenotyping research. DeepLoc achieves GPS-quality accuracy for external localisation using negligible battery life, whereas FedLoc explores federated and distributed localisation that balances privacy and precision. These architectures give an indication of how BrightDiagnostics can utilize machine learning in terms of federated privacy and energy-efficient localisation.

The Android Connectivity Diagnostics API offers live network diagnostic data to carrier/VPN applications, under the ACCESS_FINE_LOCATION permission. BrightDiagnostics unifies iOS, tvOS, and Android network information with GPS, Wi-Fi, cellular RSSI, battery, and app telemetry, among others. OpenTelemetry SDK provides batching, throttling, exporters, and collectors for structured telemetry gathering, while New Relic Telemetry SDKs provide flexible client libraries in a variety of languages [6].

BrightDiagnostics applies batching, offline caching, export pipelines, and filtering of data in the same way but for diagnostics and mobile sensor telemetry as opposed to server-side observability. Nordic/TI BLE, WiFiML, Juniper Mist, and other Wi-Fi and BLE positioning SDKs process BLE/RF data for indoor positioning and deliver location updates within a second. WiFiML enhances indoor location by employing deep learning with Wi-Fi RSS fingerprinting. BrightDiagnostics collects Wi-Fi and BLE information in a multipurpose telemetry pipeline, enabling diagnostics, attribution, and monetization.

Mobile pass-through to backend servers is utilized by drone and car telemetry SDKs (DJI, KPIT) to collect vehicle or UAV telemetry. BrightDiagnostics has support for location triggers and device/app telemetry and generalizes similar designs with BLE pass-through from DataLogger devices [10].

Architecture

The architecture of BrightDiagnostics SDK describes data flow from sensors to backend operations, the importance of each module is depicted below Figure 1 [10]:

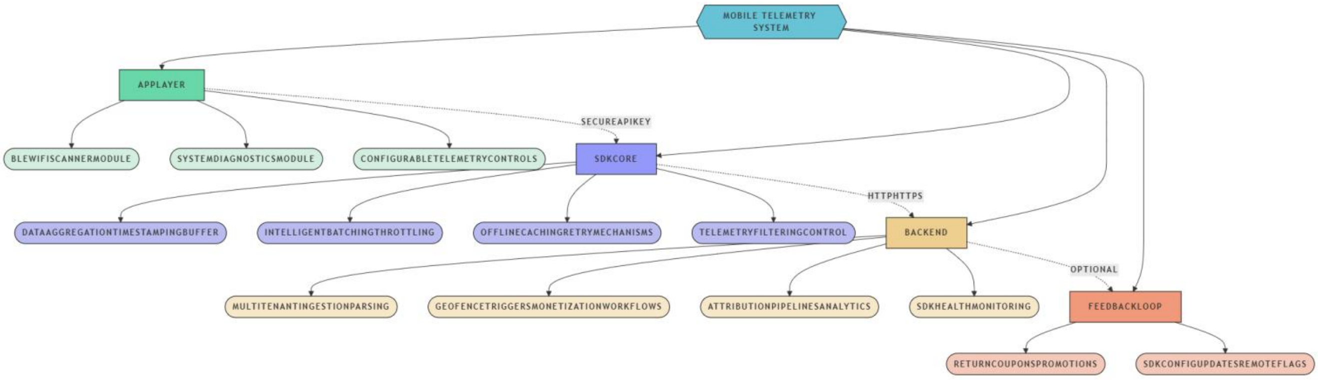

Figure 1: Process of BrightDiagnostics SDK's

1. App Integration & Real-Time Data Capture:
   Upon integration into an iOS, Android, or tvOS mobile app, the SDK continuously monitors radiolocation signals and device sensors, including GPS, Wi-Fi, BLE, battery, cellular RSSI, logs, and application performance metrics. The real-Time diagrams in SDK describes
   - GPS tracks latitude, longitude, and accuracy every five seconds.
   - Wi-Fi scans for present SSIDs and signal strength every minute.
   - Real-time beacon discovery using BLE.
   - Reports program crashes immediately and monitors CPU and battery consumption.
   - Offers complete user context such as location, connectivity, device health, and application behavior.
2. **SDK Core: Aggregation & Buffering**
   Sensor data received comes with data such as time, platform, SDK version, and user/app ID and enters an internal buffer where the SDK comes into play:
   - Batching - Splitting GPS data into 1-minute intervals.
   - Throttling - Slowing down battery telemetry to every 15 minutes.
   - Filtering - WiFi only when SSID changes.

   This follows Web-based paradigms that focus on efficient buffering, throttling, and batching, e.g., OpenTelemetry.

3. **Secure Transmission:**

   Batches are transmitted to the backend over HTTPS with:

   - theader for the authorisation API key,
   - Encrypting data in transit or at rest
   - Offline caching using local storage (e.g., SQLite or Room on Android) and exponential retry methods when connectivity fails.

   This echoes Android MVVM offline caching patterns and ensures robustness against intermittent network failure.

4. **Backend Ingestion & Real-Time Triggers:**

   Once ingested and processed, the structured telemetry, the server:

   - normalises and parses data from multiple sources,
   - triggers geofence checks, e.g., the user within 100 meters of the entrance of a retailer,
   - triggers trigger mechanism for attribution activities or promotions,
   - passes diagnostic data into monitoring systems.

   This pattern is similar to real-time dashboard and media streaming observability architecture.

5. **Return Path & Dynamic Configuration:**

   The server can react after the trigger:

   - In-app action: Pushes the "10% off nearby" offer to be displayed.
   - Updates to the SDK: Inserts geofences or alters the sample rate
   - Diagnostics/report: Records duplicate or erroneous entries and verifies successful telemetry consumption.

   This dynamic loop varies telemetry rates over time to find a balance between accuracy and battery life.

The user initiates a gasoline rewards app with BrightDiagnostics, capturing GPS, Wi-Fi scanning, and picking up BLE beacons, and transmitting a batch upon buffering data for 30 seconds shown in Figure 2:


```
{
 "timestamp"      : 1696483200,
 "gps"            : [...],
 "wifi"           : [...],
 "ble"            : [...],
 "battery"        : 78,
 "appMemoryMb"    : 120
}
```


Figure 2: SDK Telemetry Payload Structure

The server interprets GPS information to check if there is a petrol station around, and upon detection, the backend informs the user of reduced gasoline. Remote configuration changes are pushed to the SDK, and the app UI shows a real-time pop-up window with discounts.

Methodology & Implementation

Coordination and leadership in a team are important for effective project management. This includes having daily meetings to address hurdles, outlining roles, and evaluating progress. Clarity in roles allows everyone to know their responsibilities in documentation, testing, development, and managing stakeholders. Quick decision-making and eliminating barriers are the keys to momentum [11].

Project planning and execution entail developing a comprehensive project plan with specific deadlines, deliverables, and milestones. Stakeholder coordination is critical to aligning expectations and gathering needs. Project management tools such as Jira and Trello are utilized to track progress. Iterative development via an Agile process is utilized for inputs and incremental delivery.

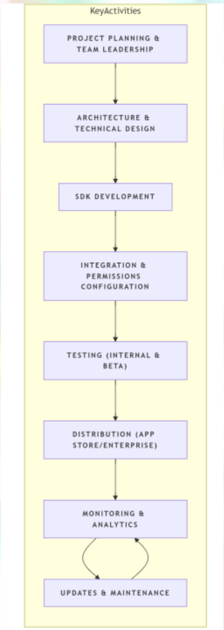

Figure 3: Methodology for Implementing BrightDiagnostics SDK

Technical direction includes architecture review, code health, debugging, and performance optimization. Regular updates and feedback loops are employed in keeping stakeholders up to date and mapping project goals onto user needs and business objectives. Resource usage is assured by ensuring the proper delegation of documentation experts, QA engineers, and developers. Budget control is tracked for the sake of optimal resource utilization. Tool provisioning guarantees the availability of necessary development environments, software licenses, and hardware. Risk management entails the active identification of operational, technical, and scheduling problems, preparation for mitigation, and establishing rapid reaction systems for new problems [12].

The software development kit (SDK) must include context, state the purpose of the project, and have an interesting hook. Technical issues like security, optimization of speed, and cross-platform support, as well as business issues like timelines and stakeholder expectations, need to be identified. The SDK must address user needs.

The project plan must document detailed project planning, set deadlines, and managed risks. The architecture design must have a secure, scalable, and modular architecture. The technology stack must specify the primary frameworks, tools, and libraries used.

Technical implementation and evolution should include team management, code review, security features, and innovations. Testing and integration should include establishing rights and integrating the SDK into software. Testing strategy should include quality assurance, in-house testing, and beta testing through TestFlight. Problem-solving should include performance optimization and problem fixes [13].

Distribution and release should be handled for the App Store and enterprise distribution. Documentation is essential, and performance measurement should be carried out using analytics. User input should be utilized to collect and make use of it, and continuous improvement should be emphasized.

Project outcomes ought to elaborate on notable achievements, advantages, and lessons obtained. The conclusion ought to summarize the significance of the SDK and the project lead's role. A call for action ought to encourage collaboration, adoption, or debate. Additional tips should be in the form of clear headings, images, real-world illustrations, balance between technical jargon and writing, and highlight the leader's effect [8].

Preparing an SDK for release is a multi-step process. Ensure that the SDK supports all target platforms, such as iOS and tvOS, and test for cross-device and cross-OS version compatibility. Provide detailed troubleshooting guides, API documents, and integration guidelines and share best practices, along with sample code snippets, with developers.

Then add the SDK to the project by importing files, establishing dependencies, and compiling parameters. Comply with platform standards and privacy policies. Conduct internal testing on registered iOS and tvOS test devices, checking the app's performance, SDK stability, and data accuracy [9].

Beta testing includes releasing the software to a controlled group of beta testers via TestFlight, collecting usability feedback and bug reports for refinement. Submit the app for review in the Apple App Store, adhering to all Apple requirements. Use the Apple Developer Enterprise Program for partners or for internal purposes, and manage safe distribution channels outside of the App Store. Update and maintain the SDK on a routine basis, monitoring tvOS and iOS updates, fixing bugs, and implementing features. Use analytics tools to monitor performance and improve the user experience [10].

The incorporation of real-time location data into applications is transforming consumer engagement in the interconnected world of today. This information can be used to improve user experience and operational effectiveness, allowing enhanced diagnostics and customized promotions. AT&T developed the BrightDiagnostics SDK to enable this to happen on iOS and TVOS platforms. The project lead managed a cross-functional team, developed a scalable solution, and guaranteed the SDK satisfied rigorous quality, security, and usability standards. This article outlines the development process, noting challenges, breakthroughs, and influences that impacted its creation [14].

Pre-made SDK provides developers with efficiency by avoiding the necessity to develop location and diagnostic data solutions from scratch. It enables developers to have control over data collection and use, allowing compliance with privacy legislation. The SDK makes available accurate location-based information, allowing developers to develop targeted user experiences that drive revenue, engagement, and retention. GPS coordinates, Wi-Fi connection, device and user data, cellular radiofrequency readings, and battery level are some of the key information gathered. Errors reported, usage of network, usage of resources, and version form part of the application metrics. It can be utilized for customer support, improving support, and tailoring marketing through location-based offers. It also enables accurate attribution for marketing campaigns and provides developers with control of data collection and compliance with privacy regulations [11].

Location accuracy is the exact location of a device, typically in meters. There are two typical ranges: High Accuracy (GPS) with 90% accuracy, which is a few meters, and Low Power (Cell Only) with city-level accuracy with minimal battery usage. The standard deviation percentile is critical for mission-critical or business application scenarios. Battery efficiency trade-offs include opting for the lowest frequency and accuracy level that will work with your application's needs [12]. Latency is a time delay between a location change and update of the app. Typical values are batch updates for passive or background application scenarios, while real-time applications such as mapping may require updating every 5 seconds with high accuracy. Configurability SDK settings enable developers to define the degree of accuracy and update frequency needed. Practical example mapping apps can provide location accuracy of a few feet or meters by using PRIORITY_HIGH_ACCURACY and 5-second update frequency. 100-meter accuracy would generally suffice and be more battery-efficient for applications such as proximity marketing. The metric data is presented in Table 1 [13]

Metric	High Accuracy (GPS)	Balanced (Wi-Fi/Cell)	Low Power (Cell Only)
Accuracy (meters)	< 5	~100	~10,000
Battery Impact	High	Moderate	Low
Use Case	Real-time mapping, mission-critical	Marketing, attribution	Passive tracking
Update Frequency	5-10 sec possible	1-5 min typical	10+ min/batched

Table 1: Metrics for Location Accuracy

To display Location Accuracy Over Time in a line graph, structured data points with timestamps and matching accuracy values (in meters) are needed, efficiently prepared and displayed as shown in Figure 4

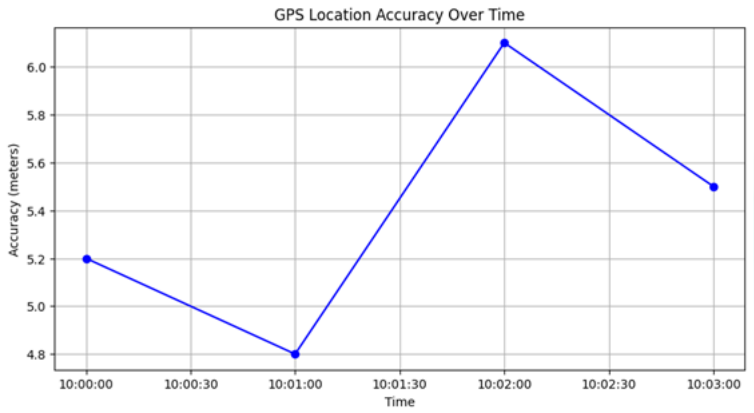

Figure 4: GPS Location Accuracy Over Time

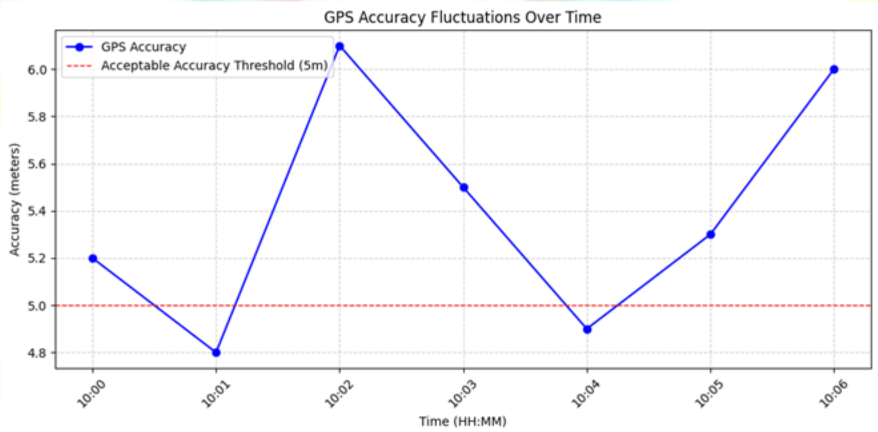

Figure 5: GPS Accuracy Fluctuations over Time

Time series GPS accuracy is perturbed by noise and seasonal conditions such as weather and tidal forces. Spatial-temporal methods can be used to correct these changes, which enhance the reliability of data. Smartphone GPS accuracy under open skies is normally 4.9 meters, although signal conditions can be different [13]. For better accuracy estimates, time series GPS station data can be analysed to determine offsets, seasonal signals, and outliers are demonstrated in Figure 5.

Conclusion & Future Scope

The BrightDiagnostics SDK is a solution that is positioned to address the increasing need for precise, real-time telemetry information in mobile applications. It makes data collection easy, maintains developer control, and provides actionable intelligence through analysis and visualization. The plans for the future of the SDK include better location intelligence, expanded telemetry coverage, real-time action and analytics triggers, edge computing features, and open ecosystem integration.

Cross-cutting positioning techniques like sensor fusion and machine learning-based location prediction will enhance GPS accuracy and reliability, particularly in hard-to-reach locations. The SDK will also enable more sensors and sources of data so that user experience can be understood more broadly. Real-time action triggers and analytics will become feasible via enhanced backend ingestion engines, edge computing features, and open ecosystem integration. BrightDiagnostics is committed to being at the vanguard of location intelligence and mobile telemetry to facilitate creativity, customization, and optimization in the ever-evolving mobile market.

References

[1] "Project Management for Software Development: Key Considerations and Best Practices", March 3, 2023 https://solutionshub.epam.com/blog/post/project-management-for-software-development.
[2] "Lessons from a Tech Lead: Roles, responsibilities, and words of advice", Tyler Hawkins, Jan 24, 2022 https://dev.to/thawkin3/lessons-from-a-tech-lead-roles-responsibilities-and-words-of-advice-ldj.
[3] The 5 best data collection tools", Harry Guinness, zapier, August 3, 2023 https://zapier.com/blog/best-data-collection-apps/.
[4] "9 Steps to creating a successful technology implementation plan", Nash Tech, 2023, https://our-thinking.nashtechglobal.com/insights/creating-a-successful-technology-implementation-plan.
[5] "BrightInsight launches new digital disease management solution for biopharma and medtech", Josh Baxt on June 13, 2022.
[6] "A digital platform to transform connected diagnostics", 2022, https://brightinsight.com/connected-diagnostics-platform.
[7] "New features from Bright to secure your apps!", Admir Dizdar, Bright, https://www.brightsec.com/blog/product-update-may-2022/.
[8] "Connectivity Diagnostics API", September 2020 https://source.android.com/docs/core/connect/connectivity-diagnostics-api?utm_source=chatgpt.com.
[9] "What is a location SDK? A Complete Guide for 2024", Marc Kranendonk, March 24, 2022.
[10] "Vehicle diagnostics architecture that enables vehicle services to move out of workshops", Auto Tech Review, 13 Dec 2018.
[11] "Design Spotify – System Design Interview", AlgoMaster, October 2, 2021 https://blog.algomaster.io/p/design-spotify-system-design-interview.
[12] "How accurate is accurate enough when it comes to location data?", Fabio Belloni, Quuppa, 12 Apr 2019 https://www.techtarget.com/iotagenda/blog/IoT-Agenda/How-accurate-is-accurate-enough-when-it-comes-to-location-data.
[13] "How Health App Data Improves Location Accuracy and Activity Identification for Investigations", Barak Goldberg, October 24, 2019.
[14] "6 Data visualization techniques to display your key metrics", Geckoboard, 19 August 2020, https://www.geckoboard.com/blog/6-data-visualization-techniques-to-display-your-key-metrics/.