One Codebase, Two Platforms

Building Native Apps with Capacitor and Vue/React/Angular

By Ankit Agade

Analyst, Digital Application Development, Blue Altair

Introduction

Mobile apps remain essential for delivering smooth, native-like experiences. However, maintaining separate codebases for Android and iOS can significantly increase development costs and slow down delivery. With open-source platforms like Capacitor combined with modern frontend frameworks (Vue, React, or Angular), developers can bridge the web-native gap and accelerate timelines.

In this blog, we'll explore what Capacitor is, why it matters, best practices for using it with Vue/React/Angular, and how Blue Altair successfully leveraged these technologies to build a real-world cross-platform solution.

Why Choose Capacitor?

Capacitor isn't just another wrapper—it's a powerful runtime that bridges the gap between web and mobile development. It allows you to continue using your existing frontend framework (React, Vue, or Angular) while unlocking direct access to native APIs and components.

Need to call native code or build custom plugins? Capacitor makes it straightforward. Compared to Cordova, it provides better web compatibility and smoother performance. Plus, its CLI can generate native iOS and Android projects that open directly in Xcode or Android Studio.

In short: you write a web app, but you deploy a fully native experience across Android and iOS.

What is Capacitor?

Capacitor is an open-source native runtime developed by the Ionic team. It allows developers to run web apps (built with Vue, React, or Angular) as fully native mobile applications on both Android and iOS.

At its core, Capacitor encapsulates standard web code, enabling a Vue-, React-, or Angular-based single-page app (SPA) to be deployed as a fully self-contained iOS or Android binary—with direct access to native device APIs such as camera, filesystem, geolocation, and push notifications.

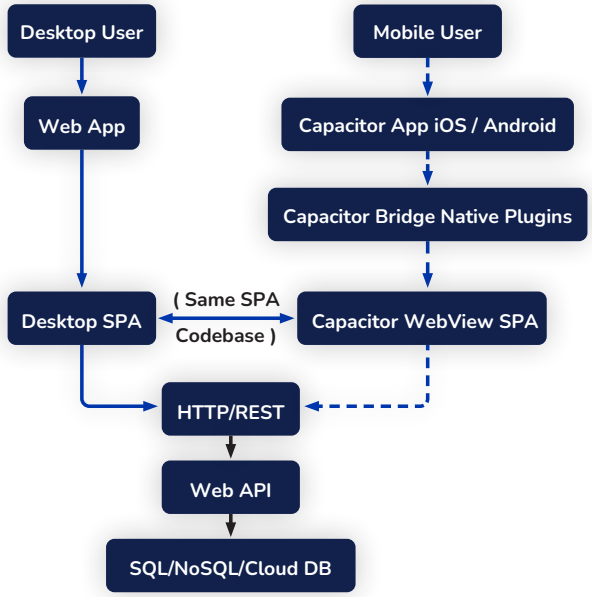
By pairing frontend frameworks for UI development with Capacitor (and optionally Ionic) for packaging and native integration, developers can achieve two platforms from one codebase, reducing both development time and ongoing maintenance.

High-Level Architecture

Think of Capacitor as the translator that lets your web app speak “native.” Here’s how the pieces connect:

Legend

- **End Users:** People using Desktop or Mobile Services
- **Desktop SPA:** Single Page Application running in desktop browser
- **Capacitor App:** Native shell for iOS/Android wrapping SPA
- **Capacitor Bridge:** Communication layer between Web and Native code
- **WebView SPA:** The SPA running inside a mobile WebView
- **HTTP/REST:** API communication between client and backend
- **Web API:** Backend serving requests
- **SQL/NoSQL/Cloud DB:** Databases used for persistence



Key Implementation Steps

Before building a custom plugin, always check the community ecosystem—chances are, someone has already built it.

Step 1: Install Libraries for Vue/React/Angular App

```
npm init vite@latest my-app --template vue ( if you are using Vue )
    cd my-app                                          # navigate to directory
    npm install                                        # install dependencies
    npm install @capacitor/core @capacitor/cli axios   # add new dependencies
    npx cap init myApp com.mycompany.myapp             # initialize capacitor plugin
```

Step 2: Hook Up the Capacitor Plugin

```
npx cap add android          # creates android folder
npx cap add ios              # creates iOS folder
npm run build                # generates web assets in dist/
npx cap copy                 # copies dist/ into native projects
npx cap open android         # opens in Android Studio
npx cap open ios             # opens in Xcode
```

From each IDE, you can now build, deploy to device or simulator, and debug, your Vue/React/Angular code live.

Step 3: Testing & CI/CD

Testing remains robust—we use Jest for unit testing and Playwright for end-to-end testing.

1. For CI/CD, our Azure pipeline: Runs tests on PRs
2. Builds Android and iOS artifacts
3. Deploys to TestFlight and the Play Store’s internal testing track

Step 4: Leverage Capacitor Plugins

Capacitor’s plugin system (official + community-developed) is one of its biggest advantages. Commonly used plugins include:

- Camera
- Gallery
- Geolocation
- SQLite (offline-first)
- Network
- Firesystem
- Browser
- Google Maps
- Push Notifications

Best Practices for Success

To maximize results with Capacitor and Vue/React/Angular:

- Keep your web code modular to increase reusability across platforms.
- Test on real devices early, not just simulators.
- Use Capacitor plugins first before writing custom integrations.
- Plan CI/CD pipelines to streamline app store deployments.

Case Study: Cross-Platform Inventory Management App

Problem

A client needed a modern inventory management app to support vehicle listings, auction workflows, and historical sales analytics. They wanted seamless deployment across Android, iOS, and web, but didn't want to triple their development investment.

The Fix

Blue Altair built the solution using a single codebase with Capacitor and Vue. Key challenges and solutions included:

Challenge	Solution
Cross-Platform Deployment	Adopted Capacitor to unify the codebase, enabling native Android and iOS builds from one source.
Offline Data Access	Implemented the SQLite plugin for offline-first capabilities with custom sync on reconnection.
Secure Authentication	Integrated JWT-based session management with token refresh via Axios interceptors.
Consistent Responsive UI	Used Vue components, CSS Grid, and media queries for seamless UX across devices.

The Payoff

- **Time-to-Market:** Reduced by 50% compared to traditional native builds
- **Code Reuse:** Achieved 90%+ reuse of UI and business logic across web, Android, and iOS
- **Maintenance Efficiency:** Centralized updates enabled faster fixes and smoother deployments

Conclusion

Building for multiple platforms no longer requires juggling multiple codebases. With Capacitor and frameworks like Vue, React, or Angular, businesses can achieve a native-quality experience from a single source of truth—accelerating delivery, cutting costs, and ensuring consistency across devices.

At Blue Altair, we've seen firsthand how this approach transforms digital projects. Our cross-platform inventory management app delivered faster time-to-market, higher code reuse, and lower maintenance overhead—all from one codebase.

Ready to fast-track your mobile app strategy?

Contact Blue Altair to explore how Capacitor can empower your next project.

About Blue Altair

Blue Altair is an innovative business and technology consulting firm that leverages transformative technologies to enable AI and drive digital success for its clients. We offer Assessment and Strategy, Technology Implementation, and Managed Services in API Management and Integration, Data Management, Digital Application Development, and Artificial Intelligence. Our Client Success capability ensures a higher-than-industry rate of successfully delivered projects, with a primary focus on program and project management, business analysis, and quality assurance. Blue Labs is our innovation hub, where we use cutting-edge technology to build offerings that deliver accelerators and solutions. Our culture is the heart of our existence, and our core values are the key drivers for our handpicked, top-tier performers.

About the Author

Ankit is an analyst on Blue Altair's Digital Application Development team, where he brings fresh energy and technical expertise to building modern applications. Skilled in JavaScript, Node.js, and Spring Boot, he has hands-on experience creating scalable backend systems and seamless web applications. Ankit is also familiar with message queues that improve communication between microservices, making systems run more efficiently. With a solid foundation in IT and HRMS domains, plus proficiency in tools like Apigee, Express.js, GitLab, and Postman, Ankit continues to deliver practical, reliable solutions that meet business needs.

bluealtair
Driving Digital Success