

Our vision for React Native is...

- **A healthy GitHub repository.** Issues and pull requests get handled within a reasonable period of time.
 - Increased test coverage.
 - Commits that sync out from the Facebook code repository should not break open source tests.
 - A higher scale of meaningful community contributions.
- **Stable APIs,** making it easier to interface with open source dependencies.
 - Facebook uses the same public API as open source
 - React Native releases that follow semantic versioning.
- **A vibrant eco-system.** High quality ViewManagers, native modules, and multiple platform support maintained by the community.
- **Excellent documentation.** Focus on helping users create high quality experiences, and up-to-date API reference docs.

We have identified the following focus areas to help us achieve this vision.

Lean Core

Our goal is to [reduce the surface area of React Native](#) by removing non-core and unused components. We'll transfer non-core components to the community to allow it to move faster. The reduced surface area will make it easier to manage contributions to React Native.

[WebView](#) is an example of a component that we transferred to the community. We are working on a workflow that will allow internal teams to continue using these components after we remove them from the repository. We have identified [dozens more components](#) that we'll give ownership of to the community.

Open Sourcing Internals and ~~✂~~Updated Tooling

The React Native development experience for product teams at Facebook can be quite different from open source. Tools that may be popular in the open source community are not used at Facebook. There may be an internal tool that achieves the same purpose. In some cases, Facebook teams have become used to tools that do not exist outside of Facebook. These disparities can pose challenges when we open source our upcoming architecture work.

We'll work on releasing some of these internal tools. We'll also improve support for tools popular with the open source community. Here's a non-exhaustive list of projects we'll tackle:

- Open source JSI and enable the community to bring their own JavaScript VMs, replacing the existing JavaScriptCore from RN's initial release. We'll be covering what JSI is in a future post, in the meantime you can learn more about JSI from [Parashuram's talk at React Conf](#).
- Support 64-bit libraries on Android.
- Enable debugging under the new architecture.
- Improve support for CocoaPods, Gradle, Maven, and the new Xcode build system.

Testing Infrastructure

When Facebook engineers publish code, it's considered safe to land if it passes all tests. These tests identify whether a change might break one of our own React Native surfaces. Yet, there are differences in how Facebook uses React Native. This has allowed us to unknowingly break React Native in open source.

We'll shore up our internal tests to ensure they run in an environment that is as close as possible to open source. This will help prevent code that breaks these tests from making it to open source. We will also work on infrastructure to enable better testing of the core repo on GitHub, enabling future pull requests to easily include tests.

Combined with the reduced surface area, this will allow contributors to merge pull requests quicker, with confidence.

Public API

Facebook will consume React Native via the public API, the same way open source does, to reduce unintentional breaking changes. We have started converting internal call sites to address this. Our goal is to converge on a stable, public API, paving the way towards the adoption of semantic versioning once React Native reaches version 1.0.

Communication

React Native is one of the [top open source projects on GitHub](#) by contributor count. That makes us really happy, and we'd like to keep it going. We'll continue working on initiatives that lead to involved contributors, such as increased transparency and open discussion. The documentation is one of the first things someone new to React Native will encounter, yet it has not been a priority. We'd like to fix that, starting with bringing back auto-generated API reference docs, creating additional content focused on creating [quality user experiences](#), and improving our [release notes](#).

The open source roadmap was originally posted to the React Native blog on [November 1, 2018](#).