slack

Slack, the communication tool used by 8 million daily users around the world, is beloved for its reliability, ease of use, and thoughtful feature set.

Additionally, many developer teams have integrated Slack into their code approval and deployment chains using integrations and webhooks. Slack's value to businesses depends on its being both useful and configurable for external developers. That means that their internal and external development experience must be fast, reliable, and stable. It's not a surprise that they work with npm.

Both of Slack's front-end teams—desktop and browser—use npm to manage their dependencies and projects. Charlie Hess, who leads the team developing the desktop app, explained how npm has benefitted their development process: "npm provides a unified workflow for both the JavaScript utilities we take off the shelf and native dependencies that boil down to a binary or an OS binding."

npm is critical to the desktop app team's development cycle, giving them an easy way to utilize and manage a powerful and rapidly growing JavaScript ecosystem. npm is how Slack has been able to develop faster and ship safter code to their 8 million plus users.

"npm provides a unified workflow for both the JavaScript utilities we take off the shelf and native dependencies that boil down to a binary or an OS binding."

—Charlie Hess

Engineer, Desktop Team

npm

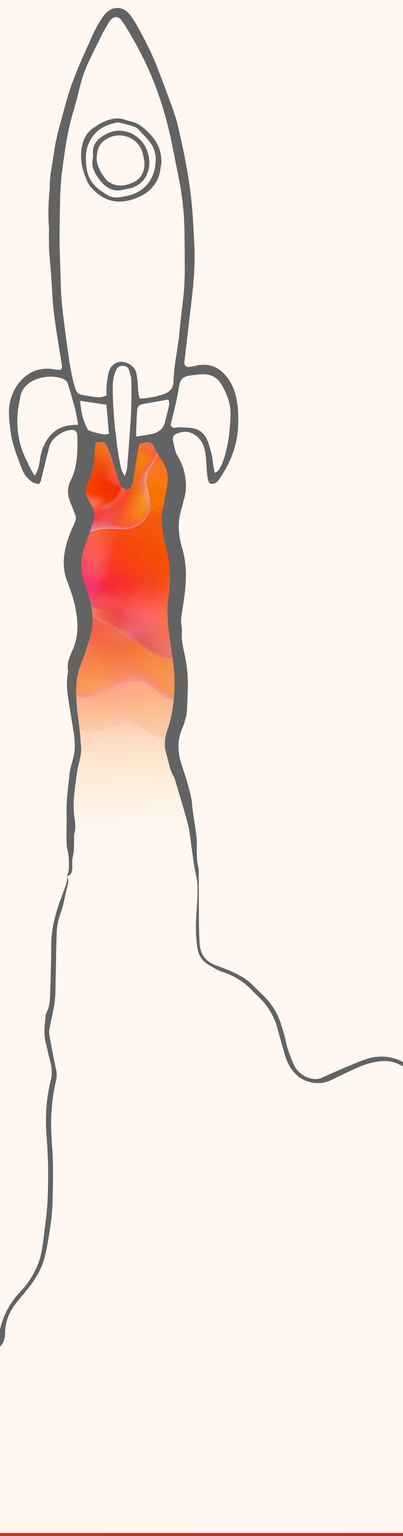
To learn more, visit www.npmjs.com/enterprise

Explosive growth

Slack has grown massively in recent years, adding 1 million paid users between September 2017 and May 2018. 65% of companies in the Fortune 100 pay to use Slack.

Central to Slack's popularity as a platform is the reliability of the Slack desktop app, which has become one of the primary ways teams and companies across every industry communicate.

As Slack began to grow into a multifunctional communications tool, the desktop team faced the need to add features not typically available to web apps: "Features like installers, spell-check, video calls, screen-sharing, and even some parts of notifications would not be possible without Node modules, many of which were written by the Electron community and shared on npm."



The wealth of code available on the npm Registry and the simplicity of code sharing and reuse allowed Slack to transform from a basic communication tool to an **industry standard.**

npm

To learn more, visit www.npmjs.com/enterprise

Before npm

Before adopting npm, front-end teams at Slack had to work with static packages and rigid dependencies. The Slack team was forced to add pieces of other projects to their source code, a practice called 'inlining dependencies'. This resulted in productivity slowdowns, as this code had to be updated manually. They also resorted to monkey patching, applying fixes to their code without going through the standardized process of security checks and testing. The lack of a central dependency management resource resulted in both security and efficiency issues.

This was not scalable, and it prevented developers from doing their best work. Slack needed a way to use up-to-date dependencies and keep them uniform between the application and browser-based versions. They also needed to make changes more quickly in response to user demands.

"Dependency management before npm was kind of like an ice age — packages were quite literally frozen in time."

—Charlie Hess

Engineer, Desktop Team

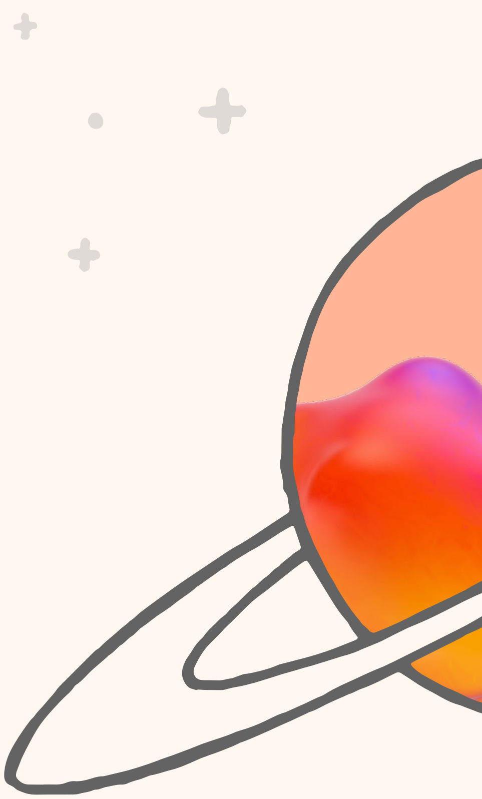
How npm helped

npm has helped Slack's front-end teams in multiple ways: solving dependency management, supplying dependencies directly to both the Slack Electron desktop app and Slack web client, and using scripts to configure and standardize internal tools. These solutions have freed Slack up to rapidly create new features that meet user demand while solving the difficult problem of dependency management. Instead of a development process defined by the limitations of static dependency trees, the team was able to experiment and find novel solutions to issues.

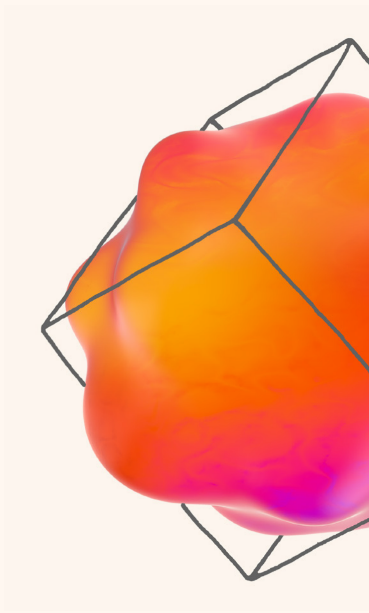
Using npm to manage external dependencies gives the Slack desktop team control over their packages from a single point and vastly reduces tooling complexity and overhead for developers.

It also allows developers to take hold of the ever-expanding number of modules in the npm Registry, using the work of millions of open source developers to build more powerful features.

Hess, who has a background in C# and native Windows apps, was pleasantly surprised by the rapid evolution that is characteristic of the npm ecosystem."I don't think I've ever seen a developer ecosystem that changes so quickly," Hess said. "While it's tempting to call every paradigm shift a flavor of the month, I do feel that we're seeing a bit more convergence or consensus, raising the bar for the average developer. It's been fascinating to watch." The constant development and improvement of libraries, build tools, and packages was a welcome change from languages that do not promote code sharing and open source collaboration.



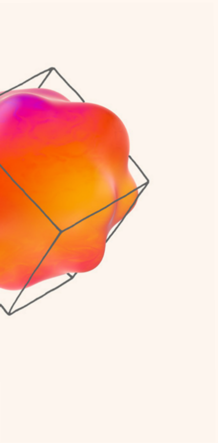
Private npm packages integrate open-source and proprietary code



Like many teams, Slack's Desktop team has proprietary code they can't host on publicly-accessible servers. Using npm's private packages and dependency management makes it possible to work with their code using the exact same workflows as they use for public packages.

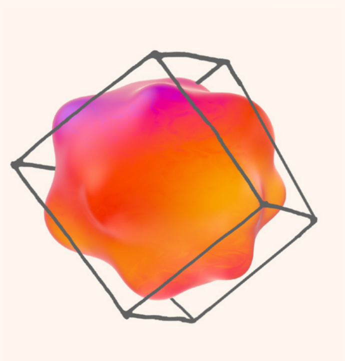
In 2017, Paul Betts, then-lead developer for Slack Desktop, described npm's private code sharing as "some kind of wizardry" compared to older methods.

The desktop team needed to solve for shipping a precompiled version of a communications library, WebRTC, and interface it with Electron. The solution? Distributing WebRTC itself within Slack using npm Orgs.



We found that the easiest way to get that Node module to have the versions of WebRTC it needed was to make WebRTC itself a npm package, so that npm install just pulls everything in correctly and builds the Calls native module. Who knew that npm is a great C++ package system?

Using private npm packages with npm Orgs or in an npm Enterprise private registry enables companies to use npm to manage dependencies, even while keeping their code base private and under control.



"I don't think I've ever seen a developer ecosystem that evolves so quickly, and while it's tempting to call every paradigm shift a flavor of the month I do feel that we're seeing a bit more convergence or consensus that is raising the bar for the average developer. It's been fascinating to watch."

—Charlie Hess

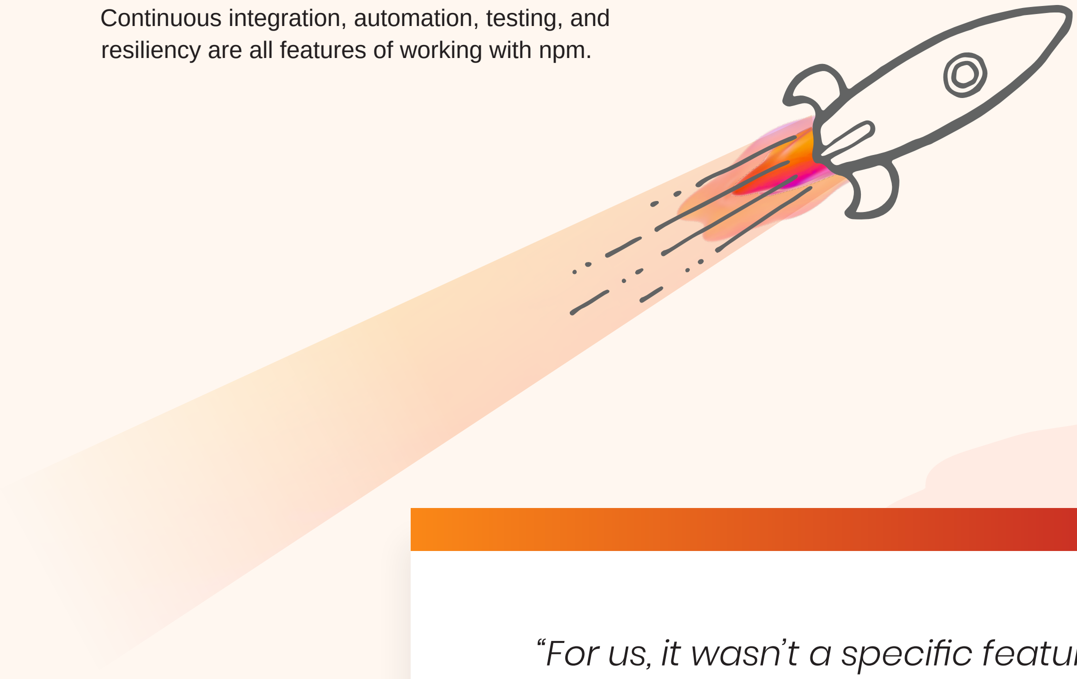
Engineer, Desktop Team

To learn more, visit www.npmjs.com/enterprise

Where Slack plans to go with npm

npm is critical to Slack's desktop app—they don't see themselves moving away any time soon.

As npm has added additional features for executing code and integration with Continuous Integration/ Continuous Deployment (CI/CD) systems, npm has grown only more integral to their workflow tooling. Continuous integration, automation, testing, and resiliency are all features of working with npm.



"For us, it wasn't a specific feature of npm so much as the gravitational pull of the ecosystem. If you're writing JavaScript today and not using external libraries you're certainly reinventing a wheel. And there are just so many wheels to choose from on npm. There are doors, engines, anti-lock brake systems, hell: your whole car might be published."

—Charlie Hess

Engineer, Desktop Team

npm

To learn more, visit www.npmjs.com/enterprise