

March 5th, 2025

# Lynx: Unlock Native for More



**Xuan Huang**  
Architect @ Lynx



**The Lynx Team**  
[lynxjs.org](https://lynxjs.org)

Today, we're excited to introduce **Lynx**, a *family* of technologies empowering developers to use their existing web skills to create truly native UIs for both mobile and web from a single codebase. Designed for diverse use cases and rich interactivity, Lynx delivers vibrant and engaging UIs for large-scale apps like TikTok, featuring a speedy, versatile rendering engine, performance-driven dual-threaded UI programming, modern Rust-based tooling, and more!

We deeply appreciate all the prior arts that have shaped the field of cross-platform development, and aspire to contribute to this movement by **open sourcing** Lynx today, inviting more developers to build applications with more freedom and productivity.

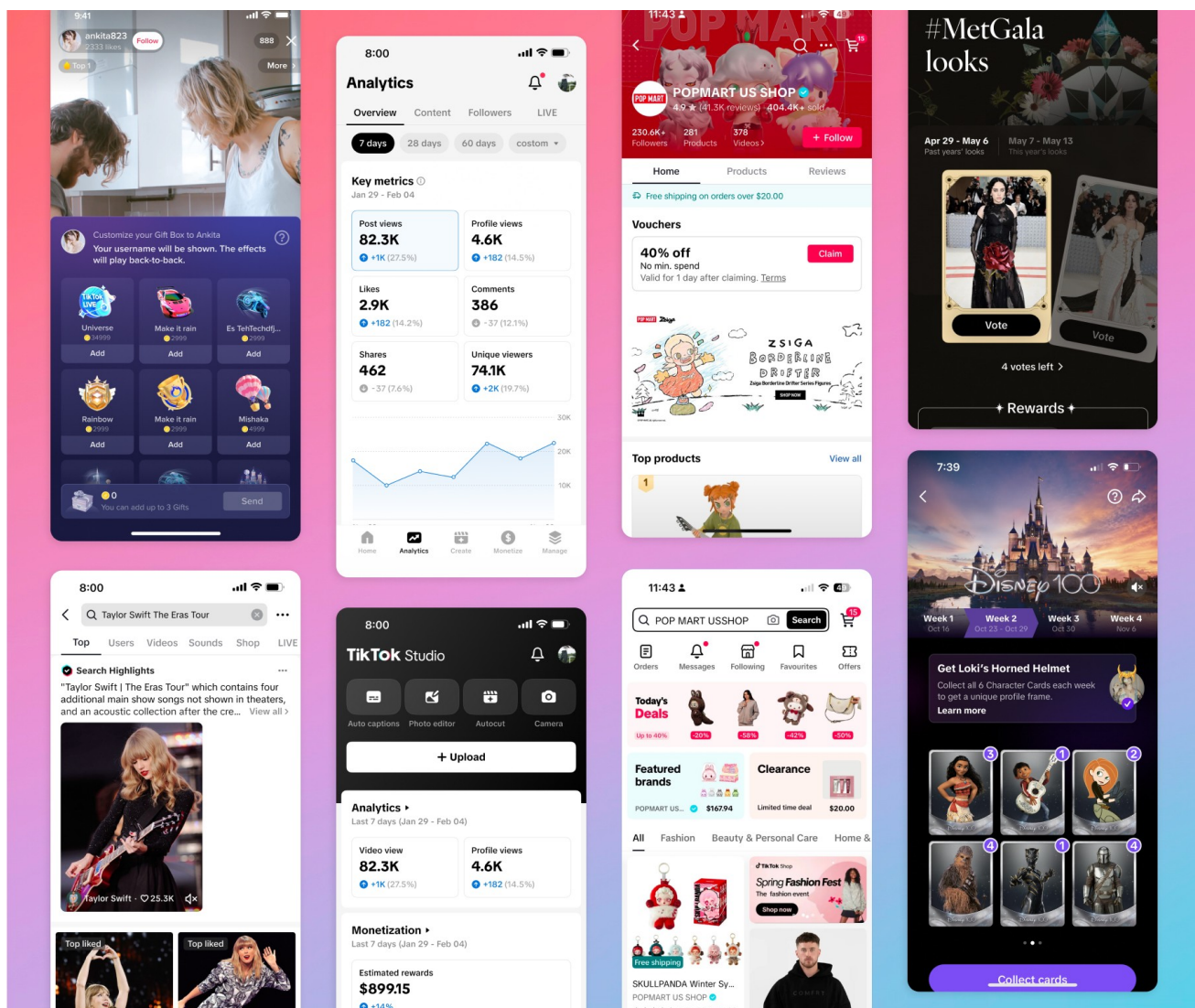
## Ship Native at Scale and Velocity

For the generation of digital natives, mobile phones—and the apps on them—are their first digital experiences. For these app-centric users, a non-native experience isn't just inconvenient; it's a red flag. A blank screen, a 0.1s lag in a "like" animation, or an unfamiliar UI pattern can make an interface feel "cheap" or untrustworthy. We believe that native primitives and responsiveness aren't just nice-to-haves—***native is a necessity***.

Despite the rapid growth of the app economy, delivering such experiences at *scale* and *velocity* remains to be a challenge. The growing diversity of form factors and platforms forces developers to rebuild the same experiences multiple times, leading to wasted effort,

siload teams, and delayed time-to-market. We believe that by enabling developers to build once and reach more platforms, *we can deliver joys for more users, faster.*

TikTok, known and loved as a mobile-first platform, continuously brings innovative and engaging experiences to diverse and dynamic users around the world. To meet these demands, it gradually adopted Lynx and has increasingly bet on it. Today, Lynx powers an extremely wide spectrum of surfaces—from the lightweight, high-frequency **Search** panel to full-fledged **TikTok Studio** ↗ app; from complex e-commerce storefronts like **Shop** that demand reliability and trust, to highly engaging experiences like **LIVE**, as well as powering high-profile events and cultural moments such as **Disney100 on TikTok** ↗ and **The Met Gala on TikTok** ↗.



TikTok's extensive use of Lynx

We believe *scale* and *velocity* mean more than just applying a feature, bug fix, or optimization across all platforms to improve cost efficiency. It also means the *versatility* to adapt the same shared technological foundation to a broad spectrum of scenarios—without

the need to build new teams or reinvent the wheel for every unique use case.

---

## Inspire and Enrich the Web community

The web platform was historically created for [documents ↗](#) and then [gradually ↗ evolved ↗](#) into a development platform. Yet, it continues to [face constraints ↗](#) that slow down its ability to adapt and innovate quickly. Therefore, shaping the web technologies for cross-platform development has become a defining norm in the industry. Since 2008, PhoneGap (later [Cordova ↗](#)) [strived to keep the Web as cross-platform solution ↗](#) by augmenting Webview with native capabilities. In 2015, [React Native pioneered bridging native UI with web technologies ↗](#), enabling declarative UI with React.js on mobile. That same year, [Flutter ↗](#), with its web heritage, adopted a similar declarative UI model and introduced a new approach using a custom rendering engine.

Lynx followed a similar spirit—you can think of it as an "***alternative Web tailored for app development***". It aims to honor the assets of web technologies while taking an opinionated approach, supporting web-like APIs and adding constraints and extensions with clear intent.

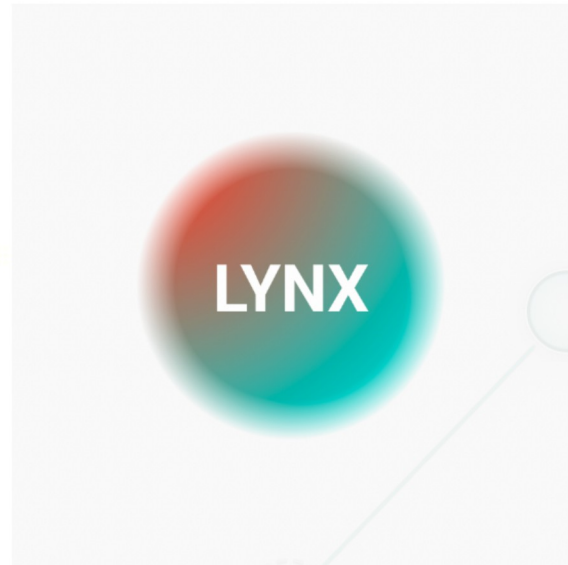
To illustrate this balance, let me share two examples: one where we choose to follow the web and another where we intentionally make a difference.

## Craft Designs with Markups and CSS as Usual

At its core, UI technologies exist to deliver exceptional product design. Lynx embraces the familiarity of web development, allowing developers to write markup and CSS just as they would for the web. Lynx natively supports [CSS animations and transitions](#), [CSS selectors and variables for theming](#), and modern CSS visual effects like [gradients](#), [clipping](#), and [masking](#)—all designed to unlock the creativity of the web community to achieve trending *design engineering*.



```
1 import { root } from "@lynx-js/react";
2
3 function App() {
4   return (
5     <view
6       style={{
7         marginTop: "50%",
8         transform: "translate(-50%, -50%)",
9         marginLeft: "50%",
10        width: "150px",
11        height: "150px",
12      }}
13     >
14     <view
15       style={{
16         flexDirection: "column",
17         background: "radial-gradient(circle, #f00 0%, #00f 100%)",
18         width: "100%",
19         height: "100%",
20         maskImage: "radial-gradient(circle, transparent 1px, black 1px)",
21         position: "absolute",
22       }}
23     >
```

[Preview](#)[QRCode](#)[css](#) [src/mask\\_image\\_circle\\_gradient/index.tsx](#)[Preview](#)

Using `mask-image` to create a circle area with fading edge

## Use the Main Thread Responsibly for Interactivity

One of Lynx's most notable architectural decisions is its **statically-enforced division** of user scripting into two distinct runtimes: a *main-thread runtime*, powered by 🧑‍🔬 [PrimJS](#) <sup>↗</sup>, a custom JavaScript engine specifically optimized for Lynx, dedicated to privileged, synchronous UI tasks like initial launch and high-priority event handling, and a *background runtime* as the default for user code, ensuring the main thread remains low workload and *non-blocking*. This enables two killer features of Lynx:

1. *Instant First-Frame Rendering (IFR)*: Backed by [usability research](#) <sup>↗</sup>, if rendering is fast enough—and Lynx is—no special intermediate feedback is needed. By briefly blocking the main thread until the first frame is fully rendered, Lynx eliminates blank screens,

creating a perceived instant experience.

2. **Main-Thread Scripting (MTS)**: A small, statically scheduled piece of code, privileged to run on the main thread, handles high-priority events and gestures—making it ideal for implementing silky-smooth, highly responsive interfaces that feels native.

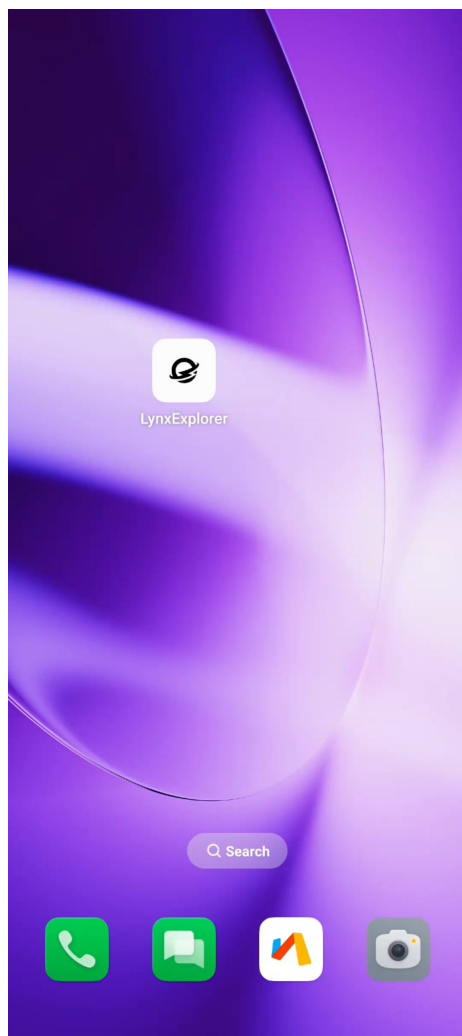


IFR



MTS

Do you notice how instant the *IFR* feels and how snappy the *MTS* is? Let's **slow them down by 4x** so we can take a closer look!



IFR (0.25x)



MTS (0.25x)

Internally, we've seen surfaces migrating from Web to Lynx often achieve a **2–4× reduction in launch times** across the board. Our in-house benchmarks also show that Lynx consistently launches faster on Android while remaining competitive with similar technologies on iOS.

---

## Open Sourcing Lynx

Lynx was originally developed by an engineering team of ByteDance, which continue to drive its development. As a significant user of Lynx, TikTok recognizes the innovation and potential of Lynx and has been making extensive use of it in a series of TikTok applications. TikTok will facilitate the open-source release of Lynx, offering support in several areas, including funding, technical enhancement, community promotion, and ecosystem growth.

## Democratize Cross-Platform Technologies

For years, cross-platform development for web developers has been defined by the patterns and solutions established by a few dominant players. While we celebrate their remarkable contribution, there is a growing demand for greater diversity in approaches. Lynx seeks to change that by not open-sourcing a single fixed solution, but a ***meta-infrastructure*** that enables teams and businesses facing similar scale and velocity challenges to build their own cross-platform solutions.

We are open-sourcing **ReactLynx** ("React on Lynx") as Lynx's initial frontend framework flavor, enabling componentized, declarative UI on Lynx. *However, Lynx isn't limited to React.* In fact, other frameworks already represent roughly half of Lynx's overall usage, demonstrating its neutrality in hosting different flavors. As modern apps grow increasingly complex, with dozens or even hundreds of developers collaborating on a single app, Lynx ships with **Rspeedy**, a toolchain based on the popular Rust-based bundler 🦀 **Rspack** ↗, to enable fast builds and pave the way for a multi-framework **micro-frontend** ↗ future via **Module Federation** ↗. We are eager to work with the open-source JavaScript framework community to bring even greater diversity to cross-platform development.

Not only is 🦄 **the core engine** ↗ of Lynx framework-agnostic, but it's also agnostic to host platforms and rendering backends. Drawing inspiration from a spectrum of projects like **Chromium** ↗, 🍌 **Flutter** ↗, and 🦄 **React Native** ↗, it's designed to adapt to new platform primitives, and it's flexible enough to switch to a *custom renderer*, enabling pixel-perfect, consistent rendering across any platforms with a graphics interface. With 🦄 **Lynx for Web** ↗, Lynx can even run natively within web browsers. Together, they give Lynx ultimate flexibilities in how it can expand to even more platforms, such as Desktop, TV, or IoT devices.

## This Is Not the End, But a New Beginning

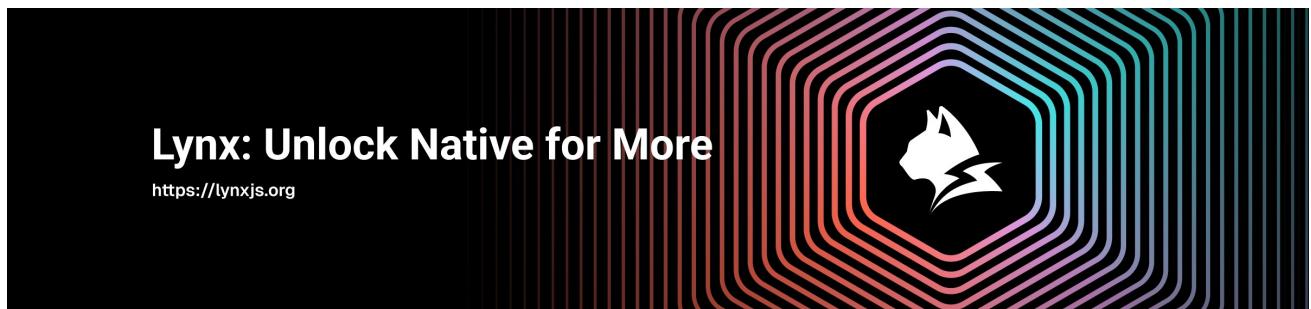
Lynx is *production-ready* and already powers an incredible number of businesses. What we are open-sourcing today is the exact version we use in production, which is why it starts at version 3.x. It even includes legacy code and APIs we intend to deprecate, but we believe in opening source what we actually rely on. In fact, **we are moving all development to GitHub**, making it open and transparent to the community.

This release marks only the beginning. What we are open sourcing today is *not* everything. Many peripherals that we're proud of—including additional UI components, advanced built-

in graphics capabilities, the custom renderer, and other frameworks—are yet to come.

More importantly, this is the beginning of the journey because, as a relatively young team in the open-source space, we know we have much to learn about working, collaborating, and growing with the community. But we are glad we took this step, because we believe open source is the right path forward—to **foster collaboration, push the boundaries of what's possible in cross-platform development, and give back to the community that has given us so much**. We invite you to join us on this exciting journey and welcome your feedback and contributions.

*What will you build with Lynx?*



  [Edit this page on GitHub ↗](#)

Previous Page  
[Lynx Blog](#)

---

Except as otherwise noted, this work is licensed under a Creative Commons Attribution 4.0 International License, and code samples are licensed under the Apache License 2.0.