

前端技术及开源发展 研究报告

云计算开源产业联盟
2023年5月

编制说明

本报告由云计算开源产业联盟撰写，限于撰写组时间、知识局限等因素，内容恐有疏漏，烦请各位读者不吝指正。

在编写过程中得到多家单位的大力支持，在此特别致谢中国信息通信研究院、华为技术有限公司、北京京东世纪贸易有限公司、深圳市腾讯计算机系统有限公司、上海浦东发展银行股份有限公司、建信金融科技有限责任公司、中原银行股份有限公司、苏州棱镜七彩信息科技有限公司、浪潮云信息技术股份公司、中兴通讯股份有限公司、中国农业银行软件研发中心、东软集团股份有限公司、深圳开源互联网安全技术有限公司、中国工商银行股份有限公司、北京小桔科技有限公司、中国联合网络通信有限公司软件研究院等单位的各位专家，感谢各位秉承开源精神，协同共创，从多个视角提出了对于前端技术的理解，最终完成本报告的撰写工作。

本报告核心编写者：张燕、俊哲、张毅、李一昂、王泽增、贾武林、林琳、焦颖。

本报告贡献者：吴海鑫、马剑、苏昊欣、胡雄伟、黄立、黄小然、谢钊、吴鑫、孙超、孙亚楠，丁涛、陈旭、丁涛、马琳琳、高杨、徐志亮、赵群、李鑫、梁锐、孟繁贵、李迎辉、牟宁波、李佩芳、杨鸣，赵赫、陈莲、王颀、贺文轩、易焕腾、谢家兴、王媛媛。

前 言

前端即网站或应用程序的用户界面部分，作为视觉和功能连接的核心，其通过程序语言整合文字、图片、视频等内容信息，呈现给用户，从而与用户建立沟通渠道。

前端技术也即用于构建网站和应用程序用户界面的各种技术。在前端技术的发展轨迹中，开源贯穿始终，逐渐成为推动前端技术全面升级的中坚力量。随着数字化转型的不断深入，市场需求日益丰富，终端种类和数量高速增长，前端正在通过持续的技术革新和融合不断突破自身边界，进而重新定义自身价值。无论是前端低代码平台建设，还是前端智能化落地，抑或是微前端体系构建，都已成为前端技术发展的重点方向。

在这种大变革的时代背景下，云计算开源产业联盟联合产业界共同发布前端技术及开源发展研究报告，聚焦前端技术领域不断涌现的新技术、新模式，分析总结全球和我国发展的总体态势，并重点针对我国前端开源技术的创新趋势和应用状况进行探讨，最后对我国前端技术的未来发展方向进行展望与研判。希望本报告为社会各界深入了解前端开源技术提供有价值的参考。

目 录

一、前端发展总体态势.....	1
（一）开源持续推动前端技术快速迭代，赋能产业全面升级.....	1
（二）软件供应链趋于复杂化和多样化，前端开源风险不断加剧.....	2
二、前端技术创新趋势.....	4
（一）开源语言和工具基础设施为前端技术发展奠定坚实基础.....	4
1.开源语言 TypeScript 成为新一代前端标准.....	4
2.Rust 工具融入前端生态，成为 JS 基础设施的未来.....	5
3.开源工具基础设施推动前端向前发展.....	7
（二）前端技术应用领域不断拓展.....	8
1.前端开源框架深耕全栈领域.....	8
2.WebRTC 在音视频技术中独放异彩.....	9
（三）新技术驱动业界开始研发模式变革.....	10
1.低代码支撑大前端工程体系构建.....	10
2.前端智能化方兴未艾.....	12
3.微前端极大提升前端生产力.....	13
4.前端“边缘渲染”成为大势所趋.....	14
三、我国前端技术发展现状调研.....	15
（一）三大主流开源框架稳居前列，黑马开源框架 Svelte 异军突起.....	16
（二）开源前端库 Axios 成为“事实标准”，AntDesign 取得广泛应用.....	17
（三）TS 采用率稳健增长.....	19
（四）微前端成为不可轻视的一环.....	20
（五）前端技术未来发展趋势调研.....	20
四、我国前端技术发展机遇.....	22
五、附录.....	23
（一）前端开源技术评价方法.....	23

一、前端发展总体态势

（一）开源持续推动前端技术快速迭代，赋能产业全面升级

开源推动前端技术持续迭代。互联网发展早期，万维网（WWW）、浏览器、开源语言 JavaScript（以下简称“JS”）相继诞生，前端界面以纯静态的文本和图片为主，前端开发工作低级单一。Ajax 的诞生助力前端技术成功迈向 2.0 时代，动态网页逐渐成为主流。与此同时，jQuery 的诞生，推动了开发者注重前后端分离。jQuery 的流行降低了前端门槛，使得前端工程师的队伍越来越壮大。而 jQuery 的广泛应用所带来的白屏和全局污染问题又进一步加速了前端模块化阶段的到来。2009 年开源技术 Node.js 诞生，并催生了前端工程化。Svelte、React、Vue、Angular 等开源框架的流行驱动了前端产业高速发展。

前端开源赋能产业全面升级。在开源技术的持续加持下，前端早已跳脱出页面展示的技术局限和功能分工，在跨平台融合技术、音视频技术以及前端整体工程效率提升技术等领域都占有一席之地。在跨平台融合技术上，React Native、WEEX、Flutter、Kraken 等一系列优秀的开源技术解决方案的出现，成功实现了逻辑跨端和渲染跨端，保障了异构终端研发的一致性，从而使得 Web 端、移动端、及各类 IoT 设备的统一前端展现成为可能。在音视频技术上，随着直播带货、在线会议、在线教育等远程实时场景的兴起，音视频技术也得到广泛关注和使用，其中 WebRTC、WebAssembly 等开源技术可帮助用户在使用音视频时获得更加理想的用户体验。随着 WebRTC 在云和端上技

术体系的成熟，视频工程开发领域的门槛也将大幅度降低。在持续的技术革新和技术融合中，前端不断自我突破、自我演进，推进产业高速发展和全面升级。在工程效率上，前端 DevOps 技术生态正持续完善，新的工具链及方法论的引入显著提升了工程研发及运维效率。低代码和无代码领域的发展仍在升温，作为一种全新的开发范式，低代码方案基于可视化和模型驱动的理念，可大幅降低业务交付成本，各家企业正在积极布局该领域，促进技术和业务深度合作。目前，国内各大互联网厂商已经开始布局低代码开源社区，积极引导低代码生态建设。

（二）软件供应链趋于复杂化和多样化，前端开源风险不断加剧

软件供应关系日趋复杂多元，供应网络安全风险不断升级。伴随着产业发展，软件开发的规模化协作模式已逐渐取代单体模式，软件各组件间连接日益紧密，从而共同形成一个包含众多节点的供应关系网络，即软件供应链。规模化协作模式盛行之下，开源模式的优势日益彰显。而引入开源软件，便为软件供应链再度引入开源软件的上游社区、基金会、开发者、维护者以及源码包、二进制包、包管理器和存储仓库等诸多节点，因此软件供应链复杂度全面升级、多样性显著增强。另一方面，开源供应链产品更新迭代速度快、软件模块数量多、生产线上化、供应全球化、仓储集中化、用户多样化等特点¹，也加剧了软件供应链的安全风险。

¹ https://www.cas.cn/cm/202011/t20201119_4767459.shtml

前端开源风险加剧，开源使用问题凸显。由于前端开源技术往往迭代更新迅速，同时缺乏相应安全机制，因此前端开源风险也随着开源在前端领域的持续渗透而有不断加剧之势。常见的开源前端软件风险主要包括三大方面。**一是安全风险**，包括开源软件本身安全漏洞所导致的风险，以及因依赖开源软件而产生的安全漏洞风险。前端开源软件具有漏洞数量多、漏洞影响范围大的特点。根据新思科技发布的《2023 开源安全和风险分析报告》²，审计代码库发现的排名前十的大漏洞中与前端相关的漏洞高达 7 个。其中排名前三的漏洞组件分别是 jQuery（47%的被审计代码库中包含）、Lodash（31%的被审计代码库中包含）、Bootstrap（23%的被审计代码库中包含），前端项目依赖的复杂性也增加了引入漏洞的风险。根据 Snyk 与 Linux 基金会联合发布的《2022 年开源安全报告》³，JavaScript 项目的平均依赖数为 174 个，是所有技术生态中最高的（Java 项目平均 40 个依赖）。**二是供应链风险**，前端开源供应链风险大有加剧之势。前端 npm 由于管控薄弱、攻击成本低而成为软件供应链攻击的重灾区，近年来已出现多起严重的前端开源供应链投毒事件。2022 年 3 月，开源组件 node-ipc 作者出于个人政治立场在代码中进行 npm 供应链投毒，作者通过添加恶意 JS 文件在用户桌面创建标语，甚至还曾计划删除特定区域用户的本地磁盘文件，vue-cli、unity-hub 等工具都因此受到了不同程度影响。**三是合规风险**，包括开源许可协议、版权和出口管制等方面的合

² Synopsys 2022 开源安全与风险分析报告 <https://www.synopsys.com/zh-cn/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html>

³ State of Open Source Security 2022 <https://snyk.io/reports/open-source-security/>

规风险。虽然大部分前端开源项目会采用比较宽松的 MIT、Apache License 等许可协议，但由于前端依赖的复杂性，需要特别注意依赖软件引入的许可协议和版权合规风险。例如 React-Native 本身采用 MIT 协议，但其中部分代码却采用了 Creative Commons Attribution ShareAlike 3.0 进行授权许可，从而导致了潜在的许可证冲突。此外，前端开源软件的许可证种类也可能发生变更。例如 React 曾经在开源许可协议 BSD 的基础上添加附加专利条款，要求使用 React 者不可做任何与 Facebook 构成竞争的事情。然而，由于社区的强烈反对以及 Apache 基金会强制要求禁用，Facebook 又将许可证更新为了 MIT 协议。

二、前端技术创新趋势

（一）开源语言和工具基础设施为前端技术发展奠定坚实基础

1. 开源语言 TypeScript 成为新一代前端标准

TypeScript 直击 JS 痛点。JS 作为前端技术的核心语言奠定了前端技术的发展基础，但当业务应用的代码量过大或逻辑过于复杂时，动态化极高的 JS 代码的维护成本和难度便会迅速爬升，甚至到达一个不可能维护的程度。而 TypeScript（下文简称“TS”）凭借其强大的类型系统以及完善的 OOP（面向对象编程）实现，在大型、复杂业务应用的开发中更具优越性。

TS 成新一代前端行业标准。由于当前几乎所有浏览器内核都将 JS 作为其唯一的默认内置脚本语言，而 TS 是基于 JS 的一个超集，

故在前端开发中使用 TS 无需转化编译，此外，对于 JS 程序员而言，TS 的使用门槛也极低，无需进行系统学习便可应用于实际开发，故 TS 颇受前端开发人员的青睐。据软件开发与设计公司 The Software House 2022 年就前端市场状态所进行的市场调查，TS 使用人数的年增长率超 7%，目前已达 84%，并且约 43% 的受访者认为 TS 将超越 JS 成为前端开发的主要解决方案。⁴另外，GitHub 2022 年调查报告显示，自 2014 年以来，TS 在主流编程语言的排名中稳步提升，当前已跃居第四。⁵

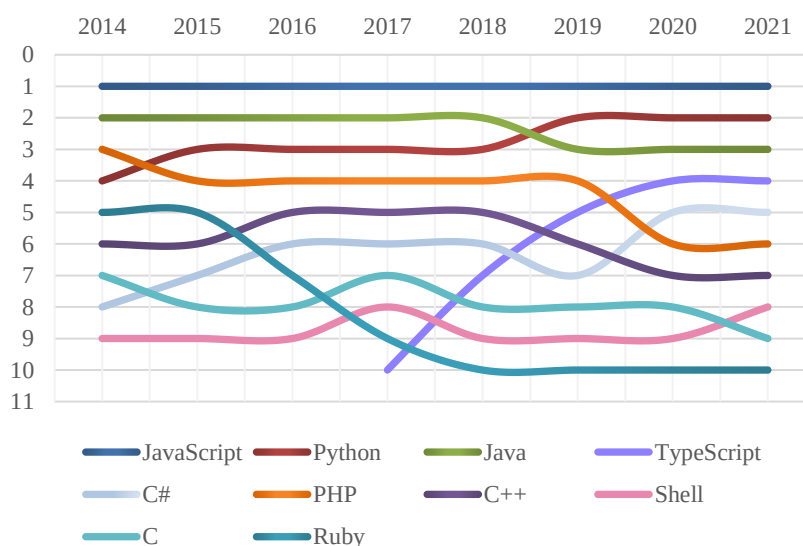


图 1 GitHub 近七年开发语言排名变化

来源：GitHub，2022 年 3 月

2. Rust 工具融入前端生态，成为 JS 基础设施的未来

Rust 逐步融入前端生态，或成 JS 工具化未来。在对基础设施性能要求极高的前端领域，核心语言 JS 却在该点上一直饱受诟病。而开源项目 Rust 自诞生以来，便以优于其他脚本语言的高可靠性和高性能吸引了大批前端开发者。自 2009 年 Rust 开源以来，社区历经十

⁴ <https://tsh.io/state-of-frontend/#report>

⁵ <https://segmentfault.com/a/1190000040980449>

余年的发展，目前 Rust 开源社区已有超过 6000 名贡献者⁶，和 1300 万开发者⁷。另根据 Stack Overflow 2021 年开发者调查，Rust 已连续六年成为最受开发者喜爱的编程语言⁸。此外，在前端领域也出现了众多由 Rust 编写的项目，如 2021 年的 swc 就是通过 Rust 编写的构建工具，具有编译、压缩、打包等功能，目前该工具已被一些知名项目采用，并显示出了强大的功效，如 Next.js 12 在以 swc 替代 babel 后，热更新速度提升到原来的 3 倍，构建速度提升到原来的 5 倍⁹。dprint，Parcel 在使用 Rust 改写后，性能也出现显著提升。前端全家桶目前已经有了一整套 Rust 实现，可见 Rust 对于 JS 的基建支持已相当完备。随着工具层逻辑覆盖率问题的逐步解决，Rust 将成为 JS 基础设施的未来。

Rust 语言学习或成前端行业重要趋势。前端领域专家 Dan Abramov 认为 Rust 是未来三年最值得学习的语言。¹⁰Rust 的学习路线虽陡峭，但其与某些经典的前端开发工具也存在相似的地方。如在工具链上，Rust 的 rustup 的使用和实现原理就类似 nvm，cargo 类似 node 的 npm。另外，在语法上，Rust 保留了函数式编程、结构化语言的特点，如 JavaScript 的箭头函数以及 map、reduce、filter 等方法皆在 Rust 语言中有所体现。随着 Rust 的广泛运用，未来 Rust 语言学习或将成为大势所趋，而得益于 Rust 更现代化的依赖管理、语法和工具链，

⁶ thanks - <https://thanks.rust-lang.org/>

⁷ State of the Developer Nation 20th Edition - Slash Data <https://www.slashdata.co/free-resources/developer-economics-state-of-the-developer-nation-20th-edition?>

⁸ <https://link.segmentfault.com/?url=https%3A%2F%2Finsights.stackoverflow.com%2Fsurvey%2F2021>

⁹ <https://www.oschina.net/news/166175/nextjs-12-released>

¹⁰ https://blog.51cto.com/u_15478221/5262617

Rust 的入门相对 C++ 等语言而言也更为容易。

3. 开源工具基础设施推动前端向前发展

前端工具基础设施持续优化。语言的流行离不开丰富完善的基础库及工具集的赋能，前端的发展亦然。在前端程序体量和复杂度不断提升的过程中，基础库或者工具对于前端的助推作用不可忽视。早期前端偏向于 MVC 开发模式，基础库旨在提供便捷好用的工具类函数，如早期的开源项目 jQuery 和 YUI，皆聚焦于提升开发者效率。ES6 版本的发布开辟了前端发展新局面，该版本不仅对原有对象进行大幅扩展，同时还新增了很多语法特性。在 Babel 诞生之前，没有任何一款浏览器引擎可实现对 ES6 全部标准的支持，但 2014 年 6to5 的库实现了 ES6 至 ES5 的转化，并在 2015 年 2 月正式更名为 Babel¹¹。Babel 完美实现了 ES6，尤其是模块的引入，使得很多优秀的设计思想（如 MVVM）和理念得以实现，后续涌现了 webpack、Rollup 和 Parcel、Esbuild 等开源工具，极大地改善了前端开发者的开发体验。

开源模式助力基础库生态体系完善发展。随着前端工程的日渐庞大和复杂，需要处理的代码量也呈指数级增长，包含数千个模块的大型项目如今已相当普遍。基于 Babel、Webpack 等开发的构建打包工具便开始遇到性能瓶颈。ECMAScript 已经发布到了 ECMAScript 2022 版本，各大浏览器引擎几乎已经完全实现了 ECMAScript 2015，因此 Babel 的转换功能不再被需要，甚至成为负累。为解决该问题，2021 年构建工具 Vite 发布，使用该工具的浏览器可运行原生 ES 模块。Vite

¹¹ <https://babeljs.io/blog/2015/02/15/not-born-to-die>

开源短短一年多来已经获得 46k star，并在 2021 年 state of js 的调查中获得最受欢迎和最满意的技术奖项¹²。开源社区也开始出现从 Webpack 迁移至 Vite 的相关工具，如 webpack-to-vite¹³。依托于开源社区，越来越多的开发者开始参与基础工具的建设，同时更多智慧和想法的碰撞和汇聚，使得前端基础库功能更加丰富和强大，如 plugin-vue 和 plugin-react 插件可使得 Vite 同时兼容 Vue 和 React，originjs 的 commonjs 和 require-context 插件可使 Vite 同时支持 ESM 和 CommonJS 两种规范，从而降低历史项目的迁移成本。

（二）前端技术应用领域不断拓展

1. 前端开源框架深耕全栈领域

前端边界面临不断突破，Serverless 赋予全栈化无限可能。前后端分离依然是当前主流的开发模式，但对于绝大多数并不十分复杂的业务场景而言，前后端联调工作所需耗费的开发成本和运维成本过于高昂。因此，无需搭建开发环境，在浏览器中完成项目部署、代码开发、配置测试以及运行管理等整套项目流程便成为现实需求。Serverless 以“无服务器”为终极理念，将提供服务资源的基础设施抽象成各种服务，供用户通过接口按需调用，开启了前端全栈化的无限可能。基于 Serverless 的页面交互、业务逻辑、数据处理的前端实现，再度突破了前端发展的技术边界。

¹² <https://2021.stateofjs.com/en-US/awards>

¹³ <https://github.com/originjs/webpack-to-vite>

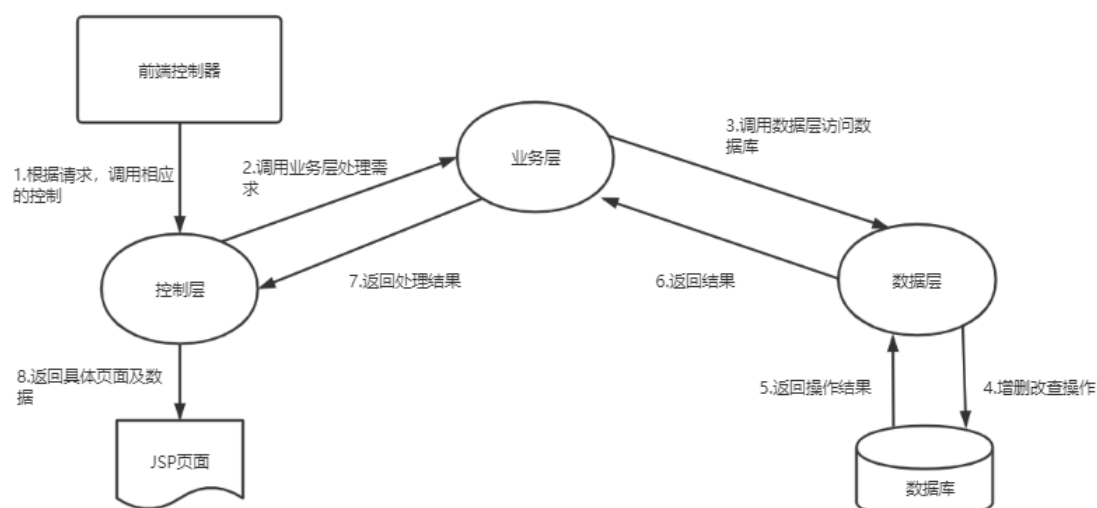


图 2 全栈项目基本框架图

来源：CSDN 网站，云计算开源产业联盟

开源赋能 Serverless 高速迭代，成行业共同选择。IBM 早在 2016 年便将 OpenWhisk 项目捐献开源，随着开源模式的持续赋能，OpenWhisk 已日趋完善，并逐渐成长为一个功能完备的 Serverless FaaS 平台，颇受开发人员的关注和青睐。2018 年，谷歌开源 KNative 架构方案，以期实现 Serverless 的标准化，伴随着 KNative 项目的迅速迭代，大型应用项目的顺利落地也将指日可待。除此之外，Openfass、Kubeless、Fn 等系列开源项目的蓬勃发展也正印证了开源具有赋能 Serverless 发展的不竭动力和无限可能。

2.WebRTC 在音视频技术中独放异彩

开源项目 WebRTC 已成为浏览器音视频应用热门解决方案。为改善在线服务质量，2010 年 Google 公司收购 Global IP Solutions 公司以获得 WebRTC 项目，并于 2011 年 6 月 1 日将其开源，后该项目又在 Google、Mozilla、Opera 的支持下被纳入万维网联盟的 W3C 推荐标准。2021 年 1 月 26 日，W3C 和 IETF 同时宣布 WebRTC 发布为正

式标准。¹⁴WebRTC 项目汇集了音视频编解码器（Opus 和 VP8/9）、强制加密协议（SRTP 和 DTLS）和网络地址转换器（ICE & STUN）等先进的实时通信技术，使得无需下载安装任何浏览器，便可实现多互联网浏览器间健壮的实时通信。而更为重要的是，开源项目 WebRTC 开放且可自由实施，在全球已安装的浏览器中，超 85% 客户端可实现音视频实时通信。¹⁵

实时音视频技术迎来风口，WebRTC 热度不断上升。伴随着移动互联网的高速普及以及音视频技术的飞速发展，越来越多的行业开始通过音视频技术以实现更高质量的用户体验，音视频会议、直播带货、在线教育、远程医疗、娱乐游戏等服务产品纷纷面世。2019 年突发新冠疫情更是从根本上改变了人类社会的生活方式，网上购物、线上办公、线上医疗、在线课堂一度成为特殊时期全人类的共同选择，实时音视频通讯的需求迅速爆发，从而使得 WebRTC 技术不断升温。WebRTC 让数十亿人跨越地域和设备，在新冠疫情期间也能保持联络。据统计，自 2022 年 3 月开始，Chrome 浏览器中通过 WebRTC 接收的视频流增加了 13 倍。¹⁶而在元宇宙、虚拟化时代到来之际，WebRTC 也有望凭借开源所实现的持续技术创新，在未来的音视频领域大放异彩。

（三）新技术驱动业界进行研发模式变革

1. 低代码支撑大前端工程体系构建

¹⁴ https://www.sohu.com/a/447299542_827544

¹⁵ <https://zhuanlan.zhihu.com/p/359202921>

¹⁶ <https://zhuanlan.zhihu.com/p/359202921>

数字化转型火热趋势下，低代码技术彰显优越性。随着数字经济时代全面开启，数字化转型已成为各行各业的选择。而如何联通数字化转型过程中所产生的大量数据以及信息化转型下遗留的数据孤岛，成为了寻求数字化转型企业不得不面对的共通难题。低代码平台则“授之以渔”，提供了有效解决该问题的开发能力。一方面，低代码平台使得不同企业可基于自身特定的数据编排需求，通过服务层的汇聚和分析，进而产生可直接服务于特定业务的新数据。另一方面，基于低代码平台拖拽而成的界面设计使得这些数据可被轻松使用，并显示在 PC、手机等不同终端。



图 3 大前端架构图

来源：CSDN 网站，云计算开源产业联盟

低代码支撑大前端工程体系构建，国内低代码开源建设成效初显。

大前端面向一切负责展示的端，包括 web 端、移动端、微信公众号、小程序等。大前端开发承担的任务更多，所需实现的功能更强大更复杂，技术形态也更加多元化，因此，大前端工程构建的工程化、自动化便势在必行。而低代码可视化配置、组件化以及多端适配等突出优势正与大前端工程标准化、数据化、配置化构建需求完美适配，因而

成为大前端工程体系构建的可靠解决方案。与此同时，各大国内互联网厂商已经开始布局低代码开源社区，引导低代码物料生态建设。业务场景的多样性决定了低代码平台对物料丰富度的强诉求，使得开源模式与低代码模式天然契合，也正因如此，低代码平台在开源社区建设中更加注重标准与协议，并乐于对外开源设计器等核心项目。在国内外互联网厂商的引领下，低代码领域已陆续涌现不少优秀开源项目，例如页面设计器 Lowcode Engine、表单设计器 DripForm、表格设计器 DripTable 等。借助低代码开源项目，开发者可以更快构建大前端工程，同时也会按照开源标准开发出丰富的低代码物料，从而实现低代码厂商与开发者之间的良性互助。

2. 前端智能化方兴未艾

前端智能化助力前端提效升级。2017 年论文 pix2Code 的提出，引发了前端开发工作者对于图像生成代码的思考，2018 年，微软 Sketch2Code 项目的开源进一步验证了前端智能化方向的可行性。纵观近年来国内外公司在前端智能化领域的持续深耕，以及在衔接设计和领域语言方面的不竭努力，已有不少成熟产品相继上线，如 Anima、imgcook、Deco、Codefun 以及开源项目 Picasso 等。其中 imgcook 作为国内知名的前端智能化平台，更是在 2021 年双 11 会场中承担了 90.4% 的新模块开发，大大降低了前端开发的产研效率。



图 4 D2C 整体流程图

来源：京东，云计算开源产业联盟

D2C 优秀开源方案落地未来可期。D2C 实现的核心在于算法，基础算法一般包括数据提取、结构布局、样式计算、命名和 DSL 等。目前业界大部分产品依然处于策略算法阶段，即如何让生成的代码更加符合开发者的预期，该过程需要进行手工策略调整，比如设计人员须参考某些规范，以提高代码生成的准确性以及对于设计稿某个区域的算法策略进行控制等。在开源社区中，前端智能化尚未出现具备足够影响力的开源项目。但随着越来越多公司的参与，以及开源事业的蓬勃发展，未来智能前端化领域百花齐放，出现行业通用解决方案以及优秀的开源方案指日可待。

3.微前端极大提升前端生产力

微前端适配开发新特点，显著提升前端生产力。随着移动互联网的高速发展，前端需求的爆发增长及日趋复杂，不仅使得前后端分离成为应用开发的大趋势，也催生了微前端架构理念。微前端实质上是微服务理念在浏览器端的巧妙应用，即通过将单一的 Web 应用拆分为多个微型前端应用，各应用独立开发部署、独立运行。微前端架构的广泛应用显著提升了前端生产力。前端模块化之下，各业务线可并行开发，各团队独立自治、互不干扰，研发、运维效率因此大为提升，应用级功能复用还可有效降低交付成本；多技术栈并存使得技术选择

增多、复用性增强以及重构风险降低；成熟的微前端应用不仅有助于形成细粒度、可组合的前端产品形态，更是微前端应用生态建设的重要基石。

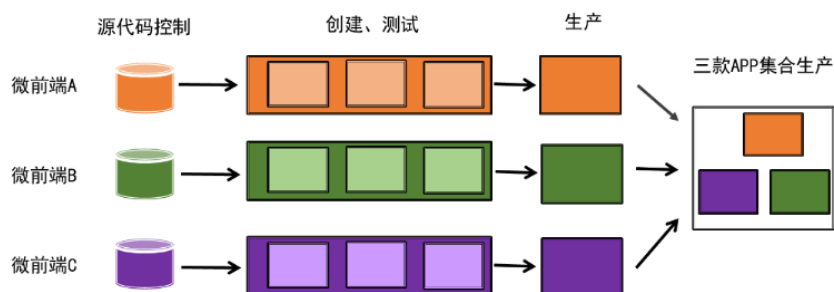


图 5 微前端架构图

来源：稀土掘金，云计算开源产业联盟

微前端应用场景全面丰富，开源发展欣欣向荣。微前端架构体系自提出以来，直击前端开发痛点问题，在遗留项目的兼容、大型应用项目的聚合以及单体应用的拆解等场景中都发挥出了不可替代的作用。微前端架构体系的高速发展，离不开开源模式的持续赋能。2019年，阿里基于 single-spa 开发出了 qiankun 开源微前端框架，当前在 Github 平台上，qiankun 的 Star 数已超 13.2k¹⁷。京东的微前端框架 MicroApp 于 2021 年开源，该框架基于组件化思想构建，以降低使用门槛、提高开发效率为宗旨，目前是市面上接入成本最低的微前端框架。¹⁸

4.前端“边缘渲染”成为大势所趋

边缘渲染成前端性能优化主流方案。根据 The Software House 于 2022 年发布的前端开发市场状态调查报告，边缘渲染成为前端性能优化的重要趋势。边缘渲染即借助边缘计算能力，将返回的内容进行

¹⁷ <https://github.com/umijs/qiankun>

¹⁸ <https://juejin.cn/post/6989435430559023117>

动、静态拆分，静态内容缓存在 CDN 中优先返回给用户，同时在 CDN 节点上继续发起动态数据请求，拼接在静态部分之后，继续流式返回。边缘渲染通过优化首屏加载，保障了边缘地区和弱网地区的用户体验。

开源框架赋能边缘渲染全面升级。开源框架 Nuxt 3、Remix 等可直接在 CDN 级别支持按需渲染。¹⁹如 Nuxt 3 采用了全新的服务端渲染引擎 Nitro，适配浏览器、Node.js、Deno、Workers 等多种环境，因此支持在 CDN worker 中渲染 Nuxt 应用，让用户在访问网站时感受到的网络延迟更低，²⁰从而进一步提升用户体验。

三、我国前端技术发展现状调研

本章内容梳理总结了来自我国互联网、金融、通信、制造、汽车等 8 大领域，超 50 个企业，200 名前端开发专业人士所填写的调研报告，同时也融合了前端技术开发领域的多位专家观点，从多个不同视角分析了我国前端发展态势以及未来可能出现的重要趋势与机遇。总体来说，前端技术发展到了一个较为稳定的阶段，当前最突出的发展态势可以概括为以下几个关键词，即：可视化、工程化、音视频、跨端、原生等。发展特点可以划分为两类，一是聚焦化，即前端技术方向划分较为细致，同时每个方向迭代到极致；二是统筹化，即打通整个前端技术上下游链路，扩展了前端的边界。

¹⁹ <https://segmentfault.com/a/1190000041943445>

²⁰ <https://juejin.cn/post/7101480249153552414>



图 6 前端技术关键词调研

来源：云计算开源产业联盟，2022 年 8 月

（一）三大主流开源框架稳居前列，黑马开源框架

Svelte 异军突起

前端界的三大主流框架引领技术创新。由于 React、Vue 与 Angular 三大开源框架诞生日期较早，取得先发优势，所以我国大部分企业内部已经具备完善的体系配套方案，市场占有率高。但相比于 React 和 Vue 在技术期望值和满意度方面的高评分，Angular 的评分较低。用户观点可以概括为以下几个关键词，即：“过于复杂”、“上手难度高”、

“框架沉重”等。Svelte 成为前端开源框架中的黑马。在众多前端框架中，Svelte 在技术期望值和满意度方面脱颖而出。近一年来，Svelte 不断推陈出新，不仅支持了 TS，同时依托社区开发者不断丰富单元测试、Web Components、SSR 等方面的实践。通过调查，用户喜爱 Svelte 的原因包括：“更高的性能和开发效率”、“更小的体积”、“更优的 Component 分发策略”等。

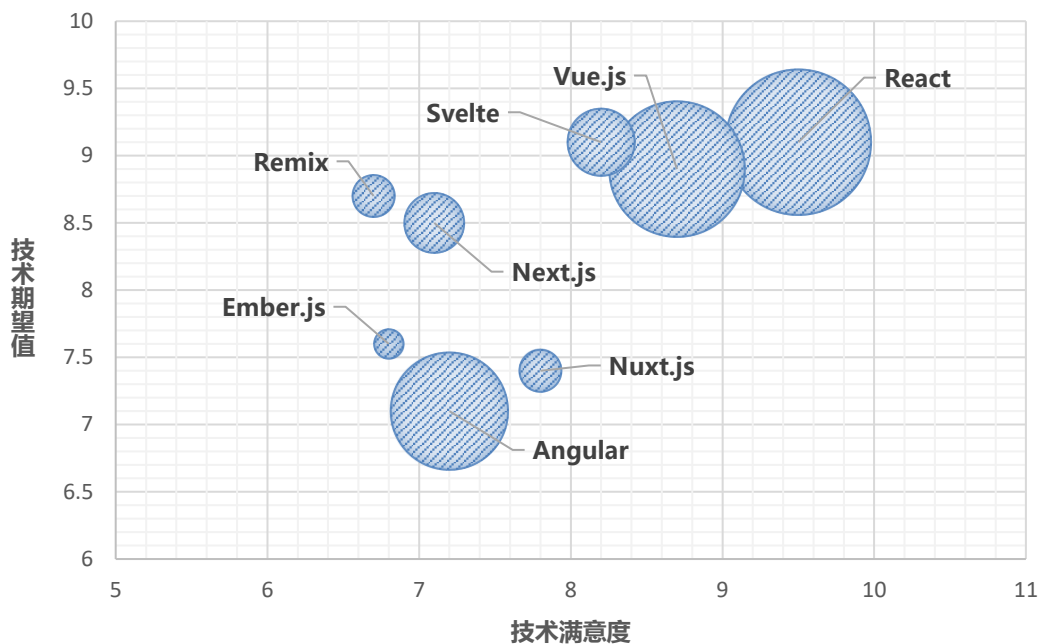


图 7 我国前端开源框架应用发展态势图

来源：云计算开源产业联盟，2022 年 8 月

（二）开源前端库 Axios 成为“事实标准”，设计系统 AntDesign 取得广泛应用

网络请求库 Axios 成为“事实标准”。Axios 是一个基于 Promise 的 HTTP 客户端，被广泛地应用在大量的 Web 项目中。目前，Axios 项目在 GitHub 上的 Star 数达到 95.4k，Fork 数达到 9.7k。与此同时，Axios 的市场份额、技术满意度与技术期望值稳居第一，超过 70% 的调研用户认为 Axios 满足企业内部的技术需求。Redux 被软件公司广泛使用。调研用户反馈 Redux 适合需要复杂状态管理的大型 Web 项目，然而 Redux 的学习门槛相当高，导致 Redux 的技术满意度与期望值较低。Moment 的用户正在转向使用其他时间库。Moment 团队于 2020 年宣布 Moment.js 已进入维护状态，后续不会再为其增加新功能，并建议新项目不要使用 Moment.js，推荐使用 Date-FNS 等更现

代的时间库，所以 Moment 项目的技术期望值与满意度是所有库中最低。

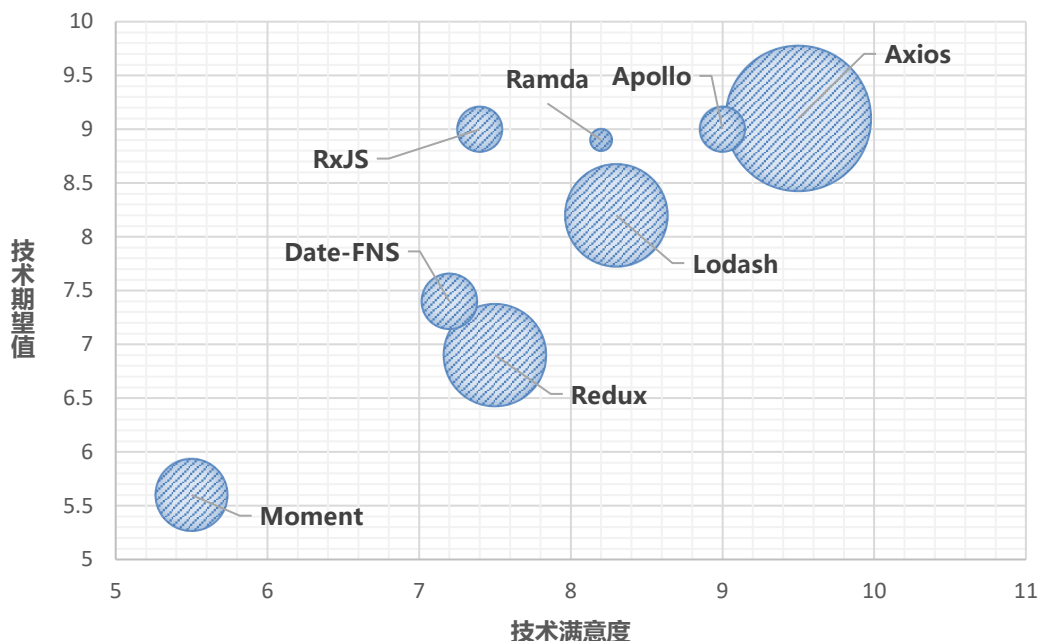


图 8 我国前端开源库应用发展态势图

来源：云计算开源产业联盟，2022 年 8 月

设计系统 AntDesign 广泛用于我国企业级中后台产品。根据调研结果，我国 42%的企业优先选用 AntDesign 作为企业级产品的设计系统。AntDesign 经过大量项目实践和总结，逐步打磨出一个服务于企业级产品的设计体系，其优点在于可以通过组件化设计规范，降低冗余的生产成本，让设计者专注于更好的用户体验。

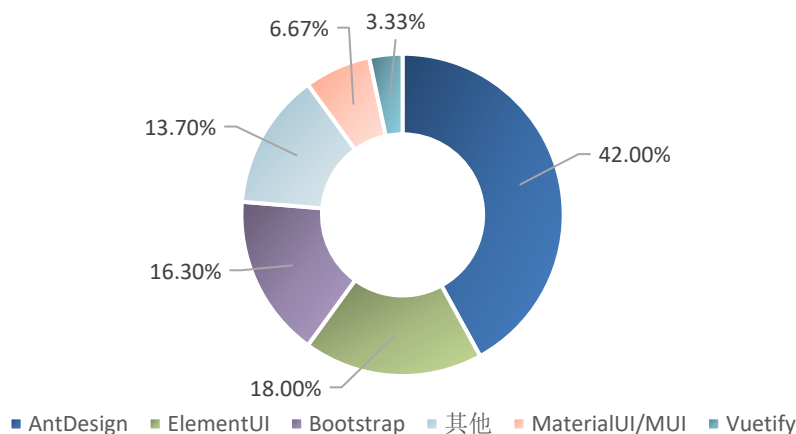


图 9 我国前端开源设计系统使用情况

来源：云计算开源产业联盟，2022 年 8 月

（三）TS 采用率稳健增长

根据调研结果，我国 75.2%的企业已经使用 TS。从 2019 年开始，社区内掀起了学习使用 TS 的浪潮，从 npm 的下载量来看，近几年 TS 采用率呈现爆发式增长。在如此高涨的增长浪潮中，TS 必将给前端生态带来新气象，同时也会促使前端开发者更多地关注代码设计、易用可维护、编码原则及设计理念等方面的知识。

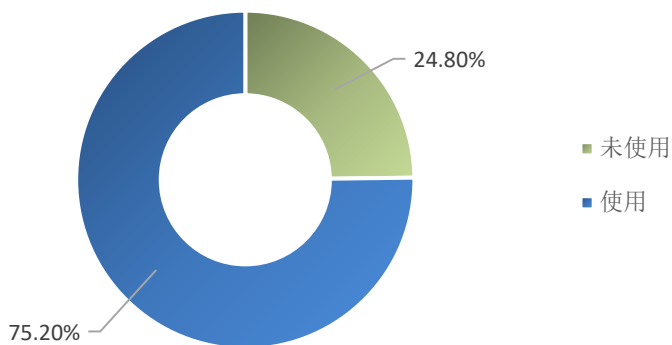


图 10 我国 TypeScript 使用情况

来源：云计算开源产业联盟，2022 年 8 月

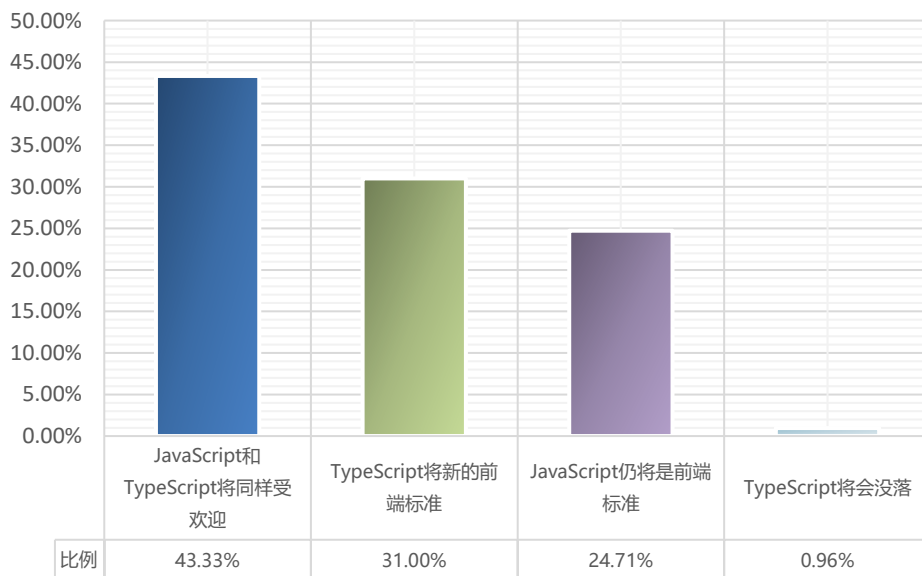


图 11 TypeScript 发展预测情况

来源：云计算开源产业联盟，2022 年 8 月

（四）微前端成为不可轻视的一环

如今，微型前端被各种公司所接受。根据调研结果，我国 65.2% 的企业已经使用微前端。除此之外，国外头部企业 Netflix、PayPal 和 Amex 等已经在内部系统中实施了这种架构方法。与此同时，成熟的开源微前端框架不断涌现，如 Single SPA、qiankun 等。

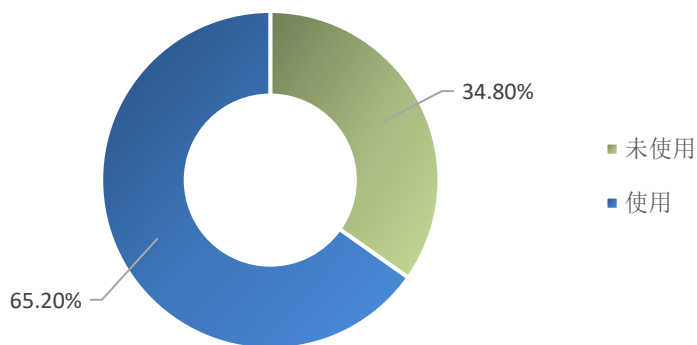


图 12 我国微前端使用情况

来源：云计算开源产业联盟，2022 年 8 月

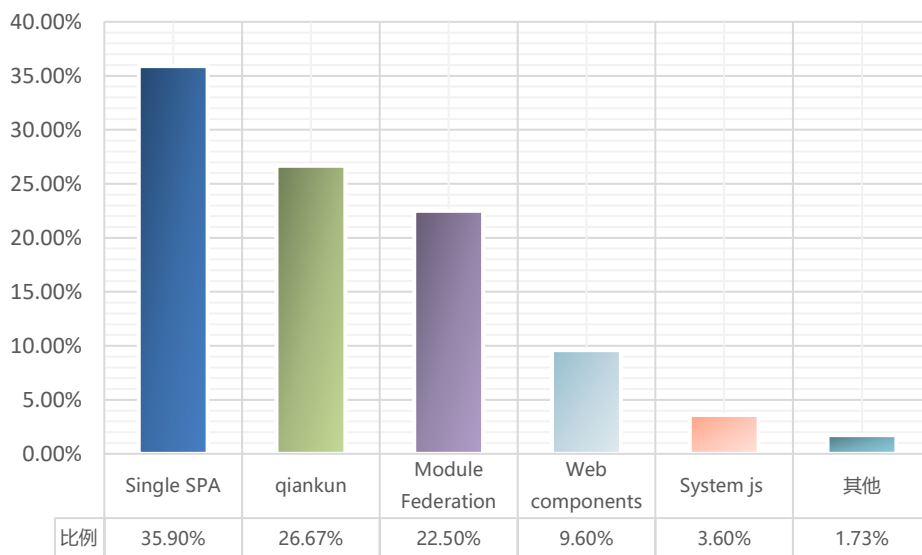


图 13 我国微前端使用情况

来源：云计算开源产业联盟，2022 年 8 月

（五）前端技术未来发展趋势调研

服务端渲染大有可为。根据调研结果，我国 76.7% 的企业认为服

务端渲染将获得人气。服务端渲染为降低渲染时间、提高用户体验开启了一扇全新的大门。**微前端值得期待**。根据调研结果，我国 86.5% 的企业认为微前端将获得人气。随着业务发展，管理系统不断增长以及大统一的趋势下，巨石应用数量将不可避免的增多，微前端则是解决这一问题的绝佳方法。**跨端开发逐渐升温**。根据调研结果，我国 80.8% 的企业认为跨端开发将获得人气。近年来，依托 Electron、React Native、Flutter 等应用框架，前端跨端开发桌面应用、移动端应用的概念持续升温，另外随着“云游戏”的火热，“云应用”也展示出了非凡的潜力。

表 1 前端技术未来发展趋势调研结果

技术	获得人气	没有变化	消亡	没有意见
无障碍性 (Accessibility)	53.3%	26.5%	2.1%	18.1%
原子设计 (atomic design)	50.2%	20.1%	0%	25.7%
组件驱动开发 (CDD)	73.4%	16.3%	0%	10.3%
跨端应用	80.8%	12.6%	0%	6.6%
GraphQL	33.3%	26.7%	7.6%	32.4%
Headless CMS	26.6%	6.7%	6.5%	60.2%
JAMstack	23.5%	33.3%	0%	43.2%
微前端	86.5%	6.6%	0%	6.9%
Progressive Web Apps	47.1%	13.4%	1.2%	38.3%
服务端渲染	76.7%	20.2%	0%	3.1%
WebAssembly	46.7%	20.5%	0%	32.8%

来源：云计算开源产业联盟，2022 年 8 月

四、我国前端技术发展机遇

通过开源协作，持续提升我国前端技术产业活力。当前开源软件已成为软件供应链的关键组成部分，开源生态建设俨然成为全球软件产业发展的共同选择。2021 年，开源首次被列入我国“十四五”规划。在国家政策的持续鼓励下，我国前端开源技术也得到了长足的发展。如尤雨溪主导的 Vue 开源框架已成为新一代前端开发的必备工具，以及我国主导的优秀前端开源项目 AntDesign、ElementUI 等，皆在国际上展现出不凡的影响力。开源模式的突出优势正在前端建设甚至整个软件产业中彰显，我国未来坚持多样性、包容性、可信的开源生态建设，依然是加速前端技术迭代创新、推动技术和产业协同稳定发展的必然选择。

参考大前端工程化理念，全面探索全栈式全自动开发方案。从软件开发的整体流程来看，借鉴大前端工程化理念，推进软件全栈式开发、测试、运维、管理的自动化和一体化将能从根本上突破软件开发传统模式的局限性，助力软件产业迅速迭代、繁荣发展。在当前大前端跨端技术融合理念的持续驱动下，低代码作为一种全新的开发范式，已成为数字化转型的重要利器。据 IDC 预测，2024 年低代码开发将成为 65% 的应用软件的首选，2025 年通过低代码/无代码技术完成开发的应用软件将达到 70%。²¹未来我国也应以低代码技术为基点，持续全面探索全栈式全自动开发方案。

²¹ <https://mp.weixin.qq.com/s/YEGDx8LteUIMLYXjoITgsQ>

大力推进前端技术规范化、工业化、智能化发展。大前端跨端融合技术发展是多端开发的必然趋势，而为进一步贯彻降本提效的大前端理念，技术规范化、工业化和智能化发展势在必行。代码风格及文档的规范化是开发协作和运行维护效率的重要保障，因此大前端跨端融合应当注重源代码的规范化，在技术产生之初，规范开发行为和技术标准，有方向的引领技术积累和发展。工业化、智能化是提高前端开发效能、释放前端产业活力的必由之路，因此大前端跨端融合应当以低代码平台、可视化工具为基，借助开源社区不断孵化出更为智能的开发工具，同时制定和推广大前端技术工业化、智能化标准，全面促进我国前端技术高质量发展。

抓住我国 5G 网络普及新机遇，前端音视频发展迎来新契机。据工信部统计数据，截至 2021 年 11 月，中国已完成 139.6 万座 5G 基站建设，占全球 5G 基站总数量的 70%，而全国所有的地级市城区中，已有超过 97% 的县城城区和 40% 的乡镇镇区实现了 5G 网络覆盖。随着 5G 技术的成熟和普及，超高可靠低时延通信将成为可能，大大提升了各行各业的工作效率。5G 普及和手机硬件提升为前端音视频发展提供了基础，而前端应用领域的持续拓展则是驱动音视频技术发展的不竭动力。因此，在元宇宙、虚拟化时代到来之际，我国应深挖前端音视频技术应用需求，通过开源促进前端音视频技术的创新及沉淀，以推动整个产业的繁荣发展。

五、附录

（一）前端开源技术评价方法

报告针对我国互联网、金融、通信、制造、汽车等 8 大领域开展前端开源技术应用发展态势观察，邀请各行业用户、专家依据前端开源技术评价方法(表 4)，对前端开源技术在各行业的使用情况进行客观打分，综合计算后形成我国前端开源技术应用发展态势图。“圆圈大小”代表该技术的市场占比大小；圆圈越大，代表该技术市场占比越大。“横坐标”代表专家对该技术的满意度，分值为 1-10 分；分值越高，代表该技术的满意度越高。“纵坐标”代表专家对该技术未来演进的期望值，分值为 1-10 分；分值越高，代表该技术的期望值越高。

表 2 前端开源技术评价方法

一级指标	二级指标	三级指标
技术满意度	喜爱度	用户是否使用和喜爱该技术
	服务满足度	外部服务支持需求
		系统版本长期维护，补丁修复及时
	生态圈	技术生态圈项目数
技术期望值	技术演进趋势	用户认为该技术是否会越来越受欢迎
	学习期望	用户是否有意愿学习该技术

云计算开源产业联盟

地址：北京市海淀区花园北路 52 号

邮箱：junzhe@caict.ac.cn

电话：18900125677

网址：trustedopensource.org.cn

