White Paper

Turning Up the Heat on API Development with ColdFusion 2016

Sponsored by: Adobe

Al Hilwa
July 2016

IDC OPINION

For over two decades, ColdFusion has delivered an end-to-end web solution for building highly functional websites and web applications. The secret to ColdFusion's endurance is constant evolution and adaptation to changing application patterns and architectures. ColdFusion 2016 exemplifies this evolution with a focus on server performance and the addition of new enhancements including a rich application programming interface (API) development and management capability aimed at the shift to service-oriented architectures. ColdFusion offers the following key benefits:

- An easy-to-learn and easy-to-master programming metaphor builds on the general web architecture and provides flexibility in the way it can be programmed. The platform supports both tag-based and script-based programming styles, allowing flexibility in expressiveness.
- ColdFusion 2016 adds a rich set of API development and management capabilities to support modern application architectures. The new features broaden the previous release's focus on mobile application development by adding back-end development capabilities now considered essential for backing B2B systems and B2C or B2E mobile and web application front ends.
- ColdFusion 2016 brings a slew of new performance improvements across many areas of the product, including the language processing and execution. New language mechanisms and optimizations, such as implicit scope lookup and array passing by reference, significantly boost the performance of the product in multiple areas.
- A new Security Code Analyzer is designed to detect a variety of vulnerabilities and attack types. In addition, ColdFusion 2016 has made significant improvements in lockdown and secure protocol support.
- ColdFusion 2016 offers mobile application development workflow support with a client-side version of the CFML familiar to ColdFusion developers and integration with Adobe's popular PhoneGap tools to create mobile hybrid apps.
- New capabilities include document sanitization, redaction, and metadata management, which add to the rich PDF manipulation and management capabilities that have differentiated ColdFusion 2016.
- ColdFusion 2016 adds many new interoperability and programming language enhancement features. Connectors have been refined and improved in many areas, and new language features such as ordered collections and passing arrays by reference have been added.
- ColdFusion 2016 has hit a sweet spot in various document-centric verticals such as legal services, the public sector, and midsize companies and ISVs seeking ease of development. IDC believes that the unique blend of developer productivity, document integration, and the new API capabilities in ColdFusion 2016 puts the technology on a durable path forward.

TABLE OF CONTENTS

IN THIS WHITE PAPER

In this white paper, we examine the value proposition of the Adobe ColdFusion application development platform and assess its position in the market. We take a look at the unique set of capabilities that ColdFusion has delivered over the years as it has cultivated certain use cases for which it is exceptionally well suited. We also explore the outlined road map for the technology and identify some of the strategic challenges and opportunities that it faces.

SITUATION OVERVIEW

ColdFusion plays in a widely covered area of the market, namely, that of server scripting technologies such as PHP, JSP, and ASP.NET. However, ColdFusion differentiates itself from other products in this broadly popular competitive landscape by offering a variety of unique capabilities and a blend of characteristics that have kept it popular among its loyal following of developers and allowed it to continue to garner new deployments.

APIs and the Changing Nature of Software Development

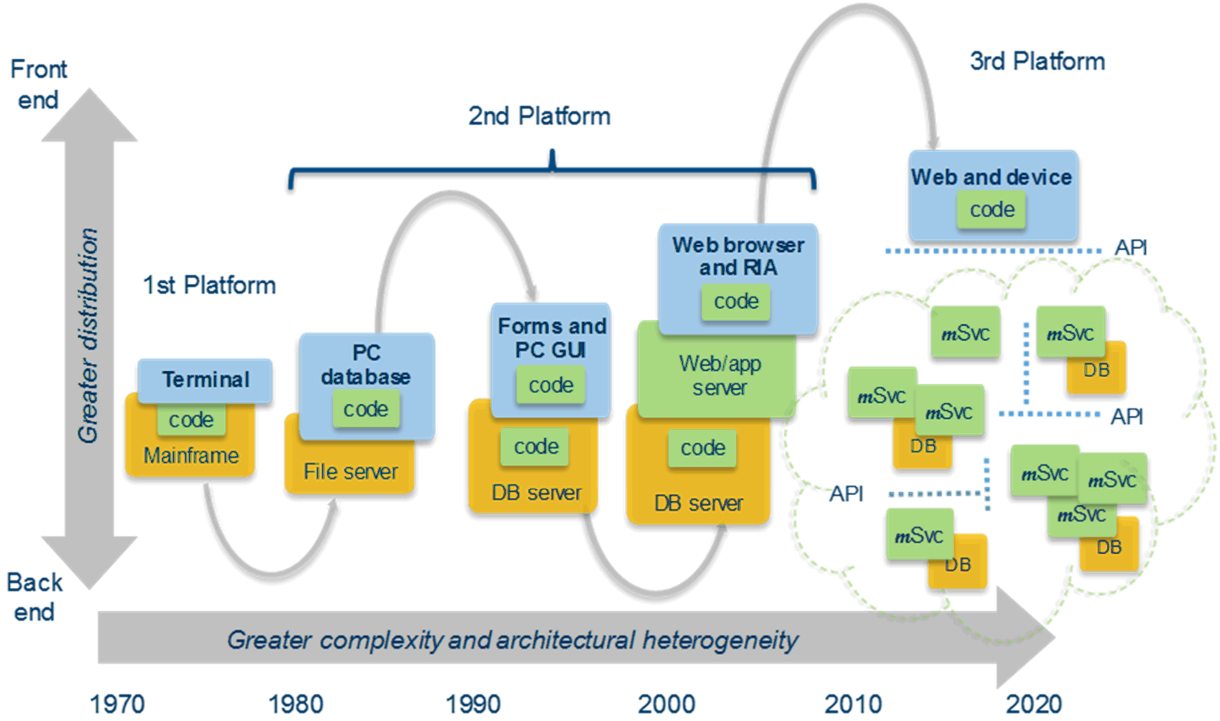
Software development is constantly changing, and of late, it has embraced change itself. The recognition that software is always in motion and that the focus of the software engineering process has to shift from that of creating static products to creating continuously evolving systems was highlighted by the agile development movement almost two decades ago as a distinct departure from the big planning approach of the waterfall approach to software development. Today, in a software delivery climate increasingly gravitating toward continuous flow of new innovation, approaches like microservices are rising to the top of the agenda as ways to organize software and even software development teams.

APIs as Heart of Mobile, Integration, and Modern Systems

To make way for a much faster pace of software engineering and continuously evolving software, the key is to create systems with independently evolvable components interacting over well-defined boundaries known as APIs. APIs are at the heart of the client/server divide and the separation we have today between mobile front ends and server or cloud back ends. APIs are also at the heart of how businesses integrate with one another and how different applications enable interoperability. Perhaps most importantly, APIs are increasingly at the heart of modern customer engagement, allowing customers and partners to interact with the business through public APIs. In the long run, however, APIs are recognized as the heart of a new approach to building large-scale software systems through a microservices architecture. APIs are a comprehensive approach to building systems of all types and are thus at the heart of the digital economy. Figure 1 details the evolution of software architecture through what IDC calls the 1st, 2nd, and 3rd Platforms and highlights the role of APIs in modern software.

FIGURE 1

Evolution in Software Architectures and the Emergence of Service APIs



Source: IDC, 2016

APIs Allow Old to Work with New

One key advantage of the API-based microservices approach to software architecture is the enablement of old systems to run side by side with, and indeed form parts of, larger systems. The key requirement in enabling this is the encapsulation of technology decisions behind API boundaries. Well-defined, communicated, and documented access points to older applications through modern interface styles known as representational state transfer (REST) make data and even business processes developed in any technology easily available for interaction and reuse by new software. This approach allows new software development to proceed with different tools and languages that are more appropriate for the specific problem domain of the modules being developed or for which there is a wider skill base available.

The level of freedom allowed for the service design and implementation team using an API approach is responsible for reduced synchronization across teams and thus responsible for reduced complexity in the overall design and enablement of faster software delivery. In a microservices-oriented architecture, the team working on a service is typically free to version the service and evolve it independently and may potentially choose to build it with a different programming language or tool chain if that is more likely to meet the design criteria. The team has to work on a well-defined and -communicated API for the service, which can evolve according to a more structured process. The idea is to leave the rest of the service internals decoupled and independent to the greatest degree possible. The importance of the independence of service evolution cannot be overstated because it is the key to scaling complex, modern, and large web-scale systems. The minimization of design synchronization is an essential value in the API-based microservices architecture.

To support a high level of service independence, successful microservices architectures require strong API design and evolution capabilities. APIs are the composition language of microservices, and API design and management skills are an essential core competency in modern software development. Tools for their creation and evolution are essential for modern system development, systems integration, and the support of new mobile and IoT deployments. With this backdrop, it is natural to expect application platform vendors to invest in mobile application development workflows, and indeed, ColdFusion has taken an aggressive jump into the API world with ColdFusion 2016. It is interesting to note that the API capabilities in ColdFusion 2016 are particularly rich and performant even though this is the first release of these capabilities in the product. The team appears to have focused on the usability of the technology and its performance with a comprehensive set of tests and internal assessments.

API Design, Creation, and Evolution

New in ColdFusion 2016 is a set of capabilities for the creation and management of APIs. The newly added ColdFusion API Manager helps create APIs that expose the various functions of a ColdFusion system and manage these APIs in operation and through their life cycle. API versioning is an essential capability for modern REST API, which allows developers to make interface changes without affecting the existing user base of the published API.

ColdFusion 2016 provides the following functions around API creation and versioning:

- **Developer portal.** The platform allows API creation and publishing through a capable and easy-to-use developer portal, which handles developer onboarding and allows APIs to be registered and tried out directly from the user-friendly UI without coding. The portal also allows developers to view API documentation and detailed API analytics and usage reports to track operations and system health. From the portal, developers can associate different usage plans with APIs.
- **API import.** The platform allows importing of APIs from Swagger, including existing ColdFusion APIs, and the importing of SOAP APIs through a proxy service and mapping SOAP endpoints to proxy endpoints. It should be noted that the ColdFusion API Manager works with APIs created in any language or environment (e.g., .NET). Many organizations are likely to have APIs written in a variety of environments, which can be holistically secured and managed through Cold Fusion.
- **Wrapping and SOAP conversion.** The platform allows API wrapping and conversion of SOAP services, which are heavily used by ColdFusion customers, into REST services more suited for modern mobile or public API usage. A wizard is provided to help in mapping of SOAP input parameters to REST parameters.
- **Versioning.** The platform offers support for multiple versions of the same API, with the ability to manage, deprecate, and retire versions independently.

- **Testing.** The platform offers testing support through an API sandbox URL, which is accessed with a testing key. The platform also supports testing through an API tryout tool for subscribers.
- **Documentation.** The platform offers features to support the documentation of APIs, including Swagger generation.

API Management and Analytics

The Cold Fusion 2016 API Manager supports and enforces user-based roles and is able to control API access to defined, trusted API subscribers. The manager also provides portals for API publishers, administrators, and subscribers, with custom functionality and workflows appropriate for the specific user role. Analytics on API usage are also provided. Key capabilities available in ColdFusion 2016 for managing APIs include the following:

- **Security and access control** allows developers to restrict access to APIs around specific thresholds based on the number of requests per unit of time. The ability to impose soft limits (notification only) or hard limits (denying further access to the API) is provided.
- **SLA management** provides support for API SLA tiers through defining throughput parameters and enforcing approvals. Developers are free to choose from predefined SLAs or create new SLA plans.
- **Clustering** provides support for clustering using the IP address of distributed data store services machines.
- **Metering** provides support for scalable metering that works across all cluster nodes, employing the use of asynchronous throttle calls based on the metrics data. Metering data is stored in a distributed in-memory database for real-time control.
- **Caching** provides support for API response caching to improve performance. The caching can be enabled at the subresource level or the API level.
- **Notification** provides support for notification and alerts such as notifying API subscribers of API status (e.g., deprecated APIs and API downtime) or notifying API publishers of requests for API subscription or the impending end of a subscribed SLA.
- **Monitoring and analytics** includes a monitoring server that can be set up to collect and provide reports on a variety of information such as publishers and subscribers, API resources, request and response payload, response status from endpoints, response times, response formats, and SLA plans. Dashboards are provided to view API usage statistics at three levels: administrator, publisher, and subscriber. Information appropriate for each level is securely displayed.

ColdFusion 2016 provides a strong set of capabilities to complement its multiparadigm approach to application development, allowing existing systems to be wrapped with modern service APIs and participate in modern architectures as independent services.

The Mobile Revolution

Organizations are increasingly gravitating toward mobile-first application development approaches as they continue to assess the degree to which their application portfolios need to be revamped to support touch computing and mobility. Consumerization has brought higher expectations for application usability along with the mobile devices themselves. With this backdrop, it is natural to expect application platform vendors to invest in mobile application development workflows. ColdFusion builds on Adobe's experience with HTML5 and the PhoneGap technology that Adobe acquired in 2011 by providing full support for mobile application development.

Approaches to Developing Mobile Apps

ColdFusion's approaches to developing mobile applications follow the web and hybrid app approaches. Developers are provided with a choice to deliver either of the two types of approaches depending on the degree to which the app uses native device features. Applications that do not use device features beyond the capability of the browser can be delivered as mobile web apps, leveraging the native device browser to run the HTML/JavaScript applications. Most enterprise applications are more likely to be delivered in hybrid form where they can be packaged with other libraries, offline databases, or device APIs to do more interesting things from within the application.

Mobile Application Creation

ColdFusion brings the power of the CFML server programming language to client devices. Client-side CFML is very similar to server-side CFML and features some of the most powerful CFML tags such as cfoutput and cfquery, but it is processed differently. Unlike server-side CFML, which is translated into Java bytecode and runs in a Java Virtual Machine (JVM), client-side CFML is translated into JavaScript. The translation of client-side CFML is performed on the ColdFusion server, but the resulting JavaScript code runs on the client. Translated code can be executed either in the device's browser or in the WebView component available on the device platform for embedding into hybrid apps.

In ColdFusion, code intended for device execution is demarcated with the use of the cfclient tag. The advantage of this approach is that developers can insert chunks of ColdFusion code in their application to produce the client side of the mobile application using many of the powerful, mature, and familiar features of the CFML. The ColdFusion approach for the creation of mobile apps allows developers to use a mix of ColdFusion skills and standard web ecosystem skills (e.g., HTML5/CSS/JavaScript) to construct the app. This allows developers with different skill sets to collaborate on projects.

Mobile Application Debugging

ColdFusion 2016 uses the popular Adobe PhoneGap process for developing web and hybrid mobile applications. The PhoneGap Developer app is designed to lower the barrier of entry to creating PhoneGap applications by allowing apps to be previewed on a device without installing platform SDKs, registering devices, or even compiling code. Unlike browsers, the PhoneGap app, which is downloadable from the standard platform owner app stores, allows full use of the standard PhoneGap APIs, including all the core plug-ins, and allows developers to work on Windows machines while developing iOS apps (which in the past have required using the Apple Xcode IDE). It is important to note that while core PhoneGap plug-ins are all available in the PhoneGap Developer app, which saves developing and debugging time and does away with the complexity of separate plug-in integration, any additional plug-in the mobile app uses in production should be packaged in the production app.

To support more effective debugging and security code analysis, the ColdFusion team developed a mobile debugger and a code analyzer that integrates with ColdFusion Builder 2016. The code analyzer is designed to identify vulnerabilities and potential security breaches and flag them with the appropriate severity levels and suggestions for remedy. The debugger provides customary debugger functions such as break points, step-by-step code advancement, and variable inspection and setting. The on-device debugging architecture uses a device agent that acts as a broker between the IDE and the application on the device to effect two-way wireless communication. The architecture avoids the use of an emulator and supports the different types of apps that ColdFusion supports, such as standalone mobile web apps and PhoneGap packaged apps.

Mobile Application Delivery

ColdFusion mobile applications are constructed to run either in the mobile web form, using the standard stock browser of the mobile platform, or in the hybrid form, where they are packaged with Adobe PhoneGap into native apps. Packaging the application for specific platforms (e.g., iOS or Android) is required only when using hardware- or device-specific functionality. Once a mobile app is created in ColdFusion Builder, a platform-specific package can be generated to be installed on the mobile device. After ColdFusion Builder sends the ColdFusion (.cfm) files to the ColdFusion server to convert them to .html and .js files, these files are bundled with a generated PhoneGap configuration file (config.xml) and submitted to the PhoneGap Build service. Complete builds can then be downloaded to the local file system once the PhoneGap Build service completes.

ColdFusion 2016 supports a complete workflow for mobile development that is highly suited to ColdFusion developers and their skill sets, allowing the developers to leverage their investment in learning and master ColdFusion technology for mobile application development.

Enduring Attributes of ColdFusion

ColdFusion 2016 focuses on evolving the platform to support modern application development while enhancing the performance of many areas of the product. It is important to review the essential attributes of ColdFusion to best understand why the technology has been compelling for certain types of requirements, especially those that integrate a variety of application domains. IDC has found the following to be the most often cited broad qualities that have made ColdFusion an enduring choice for its adopters:

- **Productivity and ease of use.** Almost all ColdFusion adopters cite its approachability as one of the key features that drew them in initially. Allowing developers to accomplish complex tasks with seemingly simple scripting is compelling, but ColdFusion also allows advanced developers to dive in and code in Java. ColdFusion 2016, and ColdFusion 11 before it, added many new language productivity improvements as well as mobile and API development capabilities. ColdFusion Builder 2016 is a high-performance IDE that allows CFML and CFScript to be edited along with web languages like HTML5, CSS, and JavaScript in a single developer console.
- **Bridging technology domains.** The ability to integrate Java systems with Microsoft systems or data from relational databases with content such as documents and presentations, to interface with email and text messaging or chat systems, or to integrate with client-side technologies such as Flash- and now HTML5-capable browsers has proved to be one of the most attractive and unique aspects of ColdFusion that has kept users loyal as new technology domains are

added. In ColdFusion 2016, the product enters the API design and management domain, highlighting its adaptability to evolving architectures.

- **Document management and manipulation.** ColdFusion includes powerful PDF manipulation and generation features, giving it unique abilities in its class as an app server. These capabilities had their roots in early versions of the product and have continued to receive investment in almost every release since. ColdFusion 11 fully revamped the PDF engine, leveraging technology from Adobe's LiveCycle and Acrobat products and the WebKit project. To the higher performance and PDF quality, ColdFusion 2016 adds document sanitization and redaction capabilities and new metadata management features, which continue to add to this important and differentiating area of the product.
- **Integration with Microsoft technologies.** First introduced in version 8, .NET integration was a key enabler because it allowed code to leverage .NET assemblies, making ColdFusion a true crossover technology connecting both the Java and the Microsoft platform technology ecosystems. In addition, ColdFusion is able to integrate with Microsoft Exchange and the Active Directory database and extract and manipulate documents from Microsoft Office and SharePoint.

A Brief History of ColdFusion

ColdFusion has seen persistent R&D investment as it has evolved over the years (refer to Table 1 in the Appendix). Table 1 highlights some of the new features introduced with major versions over the years and demonstrates evolution with market conditions and customer needs. The relatively aggressive new pace of evolution has brought ColdFusion to modern application development practices. The history shows a continued commitment by Adobe to support the product's user base as the company navigates new market conditions in the fast-changing application development space. To do so while ensuring compatibility with existing applications and maintaining older versions is part of what defines a successful enterprise platform technology.

It is important to note that releases of ColdFusion over the past five years have provided a comprehensive update for the technology, bringing it thoroughly up to date with the requirements of modern application development.

FUTURE OUTLOOK

ColdFusion has been continuously evolving. In recent years, the product has quickened its pace of evolution, adding new capabilities more rapidly and beginning to keep with the ever-expanding and fast-developing application development landscape. ColdFusion has offered substantial value as a web technology that integrates heterogeneous application areas and paradigms from diverse vendors and ecosystems. ColdFusion's role as an integration hub of multiple paradigms of technologies and multiple developer ecosystems has only expanded with ColdFusion 2016 as it has added the new realm of API design and management to its capabilities. IDC expects ColdFusion to continue to play this integrative role in the industry and tackle other capabilities that are becoming more important to enterprises, such as deeper integration with enterprise social engines and the ability to handle big data manipulation.

Server Road Map

Building on the feature-rich releases of ColdFusion 2016 and ColdFusion 11, Adobe is likely to continue to invest in a few important areas. The product is likely to continue to receive refinements in performance and enhancements in security as these continue to be overriding areas of interest in the ColdFusion customer base. In the security space, the team is looking at adding capabilities such as two-way SSL authentication, support for the JSON Web Token (JWT) standard, and support for the OpenID Connect authentication layer. The investment in API design and management capabilities is also likely to continue with enhancements in API design, debugging, and analytics capabilities and new features such as API revision history and support for the popular RESTful API Modeling Language (RAML). From a longer-term perspective, it is expected that ColdFusion will keep amplifying its API capabilities by adding API discovery and support for API marketplaces and build and enhance its integration capabilities with new connectors and orchestration capabilities. Stronger support for hybrid deployment and enablement for SaaS provisioning are also on the long-term investment slate.

CHALLENGES/OPPORTUNITIES

In recent years, ColdFusion has been evolving on a relatively aggressive schedule as Adobe has picked up the pace considerably in keeping the technology current. Nevertheless, application development technologies like ColdFusion have to stay on a technology treadmill to keep up with all the transformations taking place in the broader tech industry, many of which require support in the software development domain. We highlight challenges that also present evolution opportunities for ColdFusion in the sections that follow.

Progress on IDC's 3rd Platform

IDC has identified four key areas that encapsulate the nature of the changing tech industry over the next few years. The four anchors of the 3rd Platform are mobile, social, cloud, and big data and analytics. Each requires significant deep technical investment for any platform to support. The ColdFusion team has mobilized to seize this opportunity. ColdFusion 2016's embrace of API design and development is a key illustration of this. Similarly, ColdFusion 11's investment in mobile development provided important support for IDC's 3rd Platform, most visibly along the mobile dimension, and also featured capabilities to support social integration. Along the cloud dimension, Adobe has worked with Amazon's AWS team to support ColdFusion on the AWS Marketplace, and a number of customers are already running applications in this environment. Cloud adoption by small to midsize enterprises and ISVs, which is a key area of strength for ColdFusion, is accelerating. Adobe should continue to leverage this opportunity by investing in ties with other cloud providers as well as by considering a multitenant PaaS tier offering that is aimed at reducing installation and management effort. If ColdFusion is to capitalize on emerging growth opportunities in the application integration market, the technology also has to tackle data manipulation and transformation requirements that are beginning to emerge as smaller enterprises adopt social and big data initiatives.

Language and Model Modernization

ColdFusion was born in the early days of the web and has the inherent charter to be an integration technology for a variety of technologies. ColdFusion has been evolving continuously and is in many ways a collage of features and programming paradigms that were popular in their day. This is a problem for all software as they age over time, and while regular updates provide alignments with emerging trends and support new ideas, over time, any technology is at risk of losing its design cohesiveness and initial elegance. In particular, software products are at risk of incrementally adding features over time while retaining older features for existing users, thereby contributing to increased complexity in installation, deployment, and development. This complexity, while in many ways a necessary consequence of the need to create new applications and provide flexibility to customers, makes the technology harder to learn than originally conceived and exposes the product to competitors with newer, more elegant alternative application platforms. The types of applications ColdFusion is used to build, namely web apps, are experiencing a significantly faster pace of evolution than at any point in the past decade. Security issues and constant evolution in the now much faster-moving HTML5 browser world, especially on mobile platforms, contribute to this environment of change that keeps web application developers on their feet, constantly adapting their apps. This provides an opportunity for more proactive product management for application platforms such as ColdFusion, allowing automated updating, faster pace of deprecation of older technology, and episodic reworks of the underlying libraries, APIs, and even tags and language constructs to keep the environment modern and fresh for new developers. ColdFusion 2016's new security capabilities such as the Security Code Analyzer build on the product's accelerating investment in this space. Adobe has to continually evolve ColdFusion to keep it attractive to modern application developers.

Quality and Release Cycle Acceleration

ColdFusion 2016 and ColdFusion 11 have demonstrated considerably faster release cycles than previous releases. Today, release cycles in the industry are shortening, and many modern application development technologies are increasingly served in the cloud where they are evolved almost continuously or with much greater frequency. Changing the release cycle time frames without changing internal engineering processes may result in more buggy software and dissatisfied customers, a situation that befell a few recent releases of ColdFusion. This situation appears to have largely improved as Adobe has diligently patched bugs and improved delivery processes. Nevertheless, over time, large software products such as ColdFusion have to be delivered in a more incremental fashion. Adobe itself has been at the forefront of transformation to a more continuous delivery of product features with its successful Creative Cloud offering. It is expected that the ColdFusion team will adopt some of the delivery practices learned in the Creative Cloud transformation for ColdFusion.

Strong Model-Driven Application Platform Competition

ColdFusion plays in a large and growing ecosystem of web languages and frameworks. This ecosystem is rich with competent technologies such as Ruby on Rails and PHP. New innovation continues in the field, and more recently, the server-side JavaScript programming language called Node has captured the imagination of web developers because of the suitability of its asynchronous architecture for back-end API development. Midsize enterprises and organizations in the government and education sectors, where ColdFusion is most widely used, continue to favor higher-level approaches with data- and domain-based languages. Model-driven application platforms, which use visual models to construct application visual elements, data relationships, and process workflows, are increasingly popular with this audience. In this context, it is legitimate to ask whether ColdFusion can grow its base of developers and garner more adoption. IDC believes that ColdFusion has a loyal and mature base of developers that are unlikely to leave the product as long as it is aggressively evolved. IDC also believes that the unique blend of developer productivity, document integration, and Java fallback that ColdFusion delivers ensures lasting and potentially growing adoption as long as Adobe's ColdFusion investment endures. Nevertheless, investment in more visual tools at the front end may be important in simplifying application development for business applications where most of the usage for ColdFusion is found.

CONCLUSION

ColdFusion has hit a sweet spot in various customer segments, especially midsize companies and smaller ISVs, because of its ease of development and flexibility. ColdFusion 2016 brings the technology to the modern world of APIs and microservices, building on the mobile and social capabilities introduced in ColdFusion 11. The new set of API capabilities complement ColdFusion's multiparadigm approach to application development, allowing ColdFusion servers to be used for public API services or to be integrated into new systems through REST API services. A scan of ColdFusion's evolution exposes a technology on a mission to integrate evolving approaches to software development with varied new capabilities from external platforms. If it has to be boiled down to its essence, the key innovation of ColdFusion is to act as the Switzerland of varied technologies and platforms, often bridging diametrically opposed ecosystems such as Java, .NET, and web. The amazing applications that result often bring these varied capabilities into rich, integrated systems, which are impossible to fashion with so few resources, so few lines of code, or so little time without ColdFusion.

LEARN MORE

Related Research

- *Worldwide Development Languages, Environments, and Tools Forecast, 2016-2020: Leading with Mobile* (IDC #US41519915, June 2016)
- *Worldwide Development Languages, Environments, and Tools Market Shares, 2015: The Effects of Open Source* (IDC #US41519916, June 2016)
- *The Evolving State of PaaS, Part 1: Model-Driven PaaS for the Elusive Citizen Developer* (IDC #US41519916, June 2016)
- *The New Developer Landscape — Understanding the Modern Software Developer* (IDC #DR2016_T5_AH, March 2016)
- *The Evolving State of Mobile Software Development* (IDC #US40733015, December 2015)
- *Worldwide Enterprise Mobile Application Development Platform Forecast, 2015-2019* (IDC #US40705615, December 2015)
- *Market Analysis Perspective: Worldwide Application Development Software, 2015* (IDC #259375, September 2015)

APPENDIX

TABLE 1

ColdFusion Major Release History

Release	Date	New Feature Highlights
CF 2016	February 2016	New API design, development, and management capabilities; investment in performance across the product; security enhancements; support for PDF sanitization and redaction
CF 11 CF Builder 3	April 2014	New CF Builder, mobile development with PhoneGap and integrated debugger, client-side CFML, revamped PDF engine, enhanced security, multisite RESTful web services, social network integration, member functions, full CFScript
CF 10	May 2012	JRun replaced by Tomcat, HTML5 WebSocket and interactive charting, improved web services and REST support, Hotfix installer, improved scheduler, support for Windows 8 and Windows Server 2012
CF Builder 2	May 2011	Improved code navigation and searching, granular code formatting, automatic method stub creation, code assists for argument context and hover support
CF Builder 1	March 2010	Eclipse-based IDE with CFML, HTML, JavaScript, and CSS syntax highlighting; code folding and refactoring; outline viewing; line-level debugging
CF 9	October 2009	Java ORM with Hibernate, Apache Solr, Microsoft Office and SharePoint integration, CFScript ColdFusion components (CFCs), language enhancements
CF 8	July 2007	Many new features such as JSON serialization, PDF and Acrobat Connect integration, Microsoft Exchange and .NET integration, language additions and new tags, AJAX widgets, server monitoring and reporting
CF MX 7	February 2005	Flash and XForms-based web forms, PDF ReportBuilder, integration with non-HTTP request services (e.g., SMS and IM services)
CF MX 6.1	July 2003	Ability to rewrite in Java to run on Java EE (JRun), object orientation with CFCs, Flash integration features such as Flash Remoting, compilation to Java bytecode, other JEE app servers, CFC model improvements
CF 5	June 2001	First release after Macromedia acquisition, enhanced query support, new reporting and charting features, user-defined functions, improved management tools
CF 4.5	November 1999	Ability to access external system resources (e.g., Java objects), enhanced editor, scriptable deployment, advanced project management, true debugger
CF 4	November 1998	Multithreading and enhanced security and multiserver support
CF 3.1	January 1998	ColdFusion Studio with HTML syntax checking and live page view
CF 3	July 1997	Custom tags, a search and indexing system for text, ColdFusion Studio IDE
CF 2	November 1996	Custom tags and new language features, 150+ new functions
CF 1	July 1995	Tag-based language for data-oriented websites written in C++ by J. J. Allaire, founder of Allaire Corp.

Source: IDC, 2016

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-insights-community.com
www.idc.com