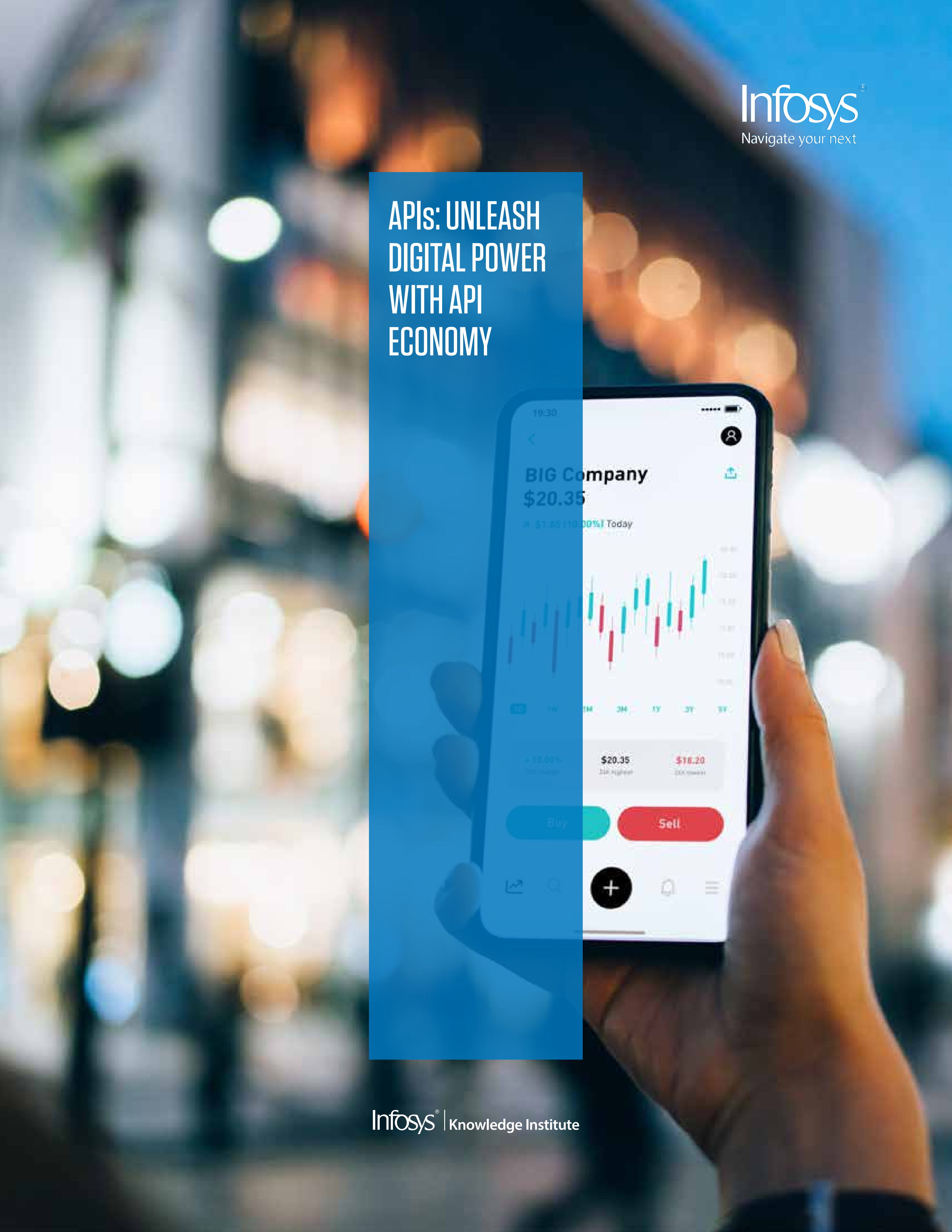
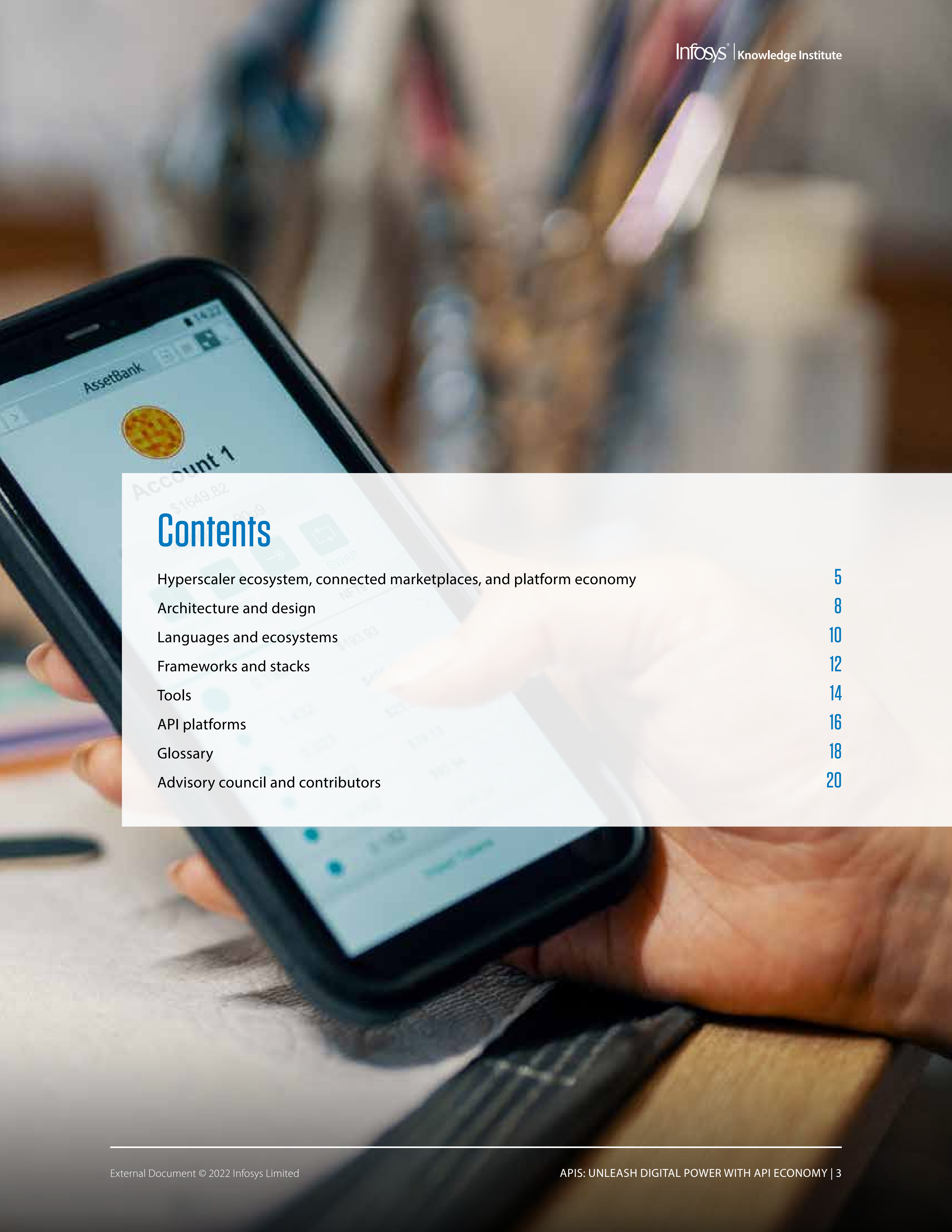


Infosys
Navigate your next

APIs: UNLEASH DIGITAL POWER WITH API ECONOMY

Infosys | Knowledge Institute



Contents

Enterprises progressively shift to cloud and hybrid ecosystems to accelerate their digital journey. API-led transformation helps expand businesses, modernize core functions, and enhance customer experience.

Enterprises rely heavily on application programming interfaces (APIs) and microservices to build and connect applications. In the application development ecosystem, API is the visible layer and distributed application runtime, containers, interfaces, integration, serverless computing, DevSecOps tools and platforms work as catalysts. APIs are layered over on-premises systems to promote partner connectivity and developer productivity.

Enterprises must closely monitor the collaboration and connectivity between applications, data, and processes and explore ways to utilize data sources to become more resilient and agile. They should also decompose and deploy applications as independent units. Together, APIs and microservices developed over the cloud will support these efforts and help businesses fast-track to a platform economy and hyperscaler ecosystem. The platform economy entails developing repeatable, composable, and reactive applications that are scalable, distributed, and run on any cloud platform. Other outcomes include AI-assisted development to increase productivity, DevSecOps to enable extreme automation, and new ways of connecting applications.

Hyperscaler ecosystem, connected marketplaces, and platform economy

Digital platforms have become the de facto way to conduct businesses as they enable real-time responses, personalized connections, and an omnichannel experience. Only companies that unlock the potential of the API ecosystem and modernize their core business integration will succeed in this new paradigm.

The SOA wave (horizon 1, H1) established better connectivity through standardized interfaces between systems. However, the SOA could not eliminate monolithic and on-premises systems that slowed performance and inhibited fast responses. Thereafter, the boom in digital economy over last decade pushed many businesses towards embracing APIs on their digitization journey (H2). During this phase, service automation, backed by Agile practices, became dominant.

> IT teams can abridge the multifaceted service-oriented architecture (SOA) culture by employing key characteristics of the API economy — automation, platform-as-a-service (PaaS), agile practices, and 12-factor app principles. For instance, platforms boost productivity and increase efficiencies by eliminating manual intervention and providing access to prebuilt components.

Indicators for the next horizon (H3) point to a platform economy and hyperscaler ecosystem. In H3, the focus is to maximize flexibility, agility, and resilience, and deliver intelligent business platforms that provide mobile- and AI-first experiences through microgateways, hybrid APIs, microservices, and edge APIs.

> Infosys' One Cloud Marketplace and One Cloud Sandbox address two fundamental needs of the API journey. They provide governance solutions for all users, including businesses, IT, and developers.

Figure 1. Adapting to market dynamics across the three horizons

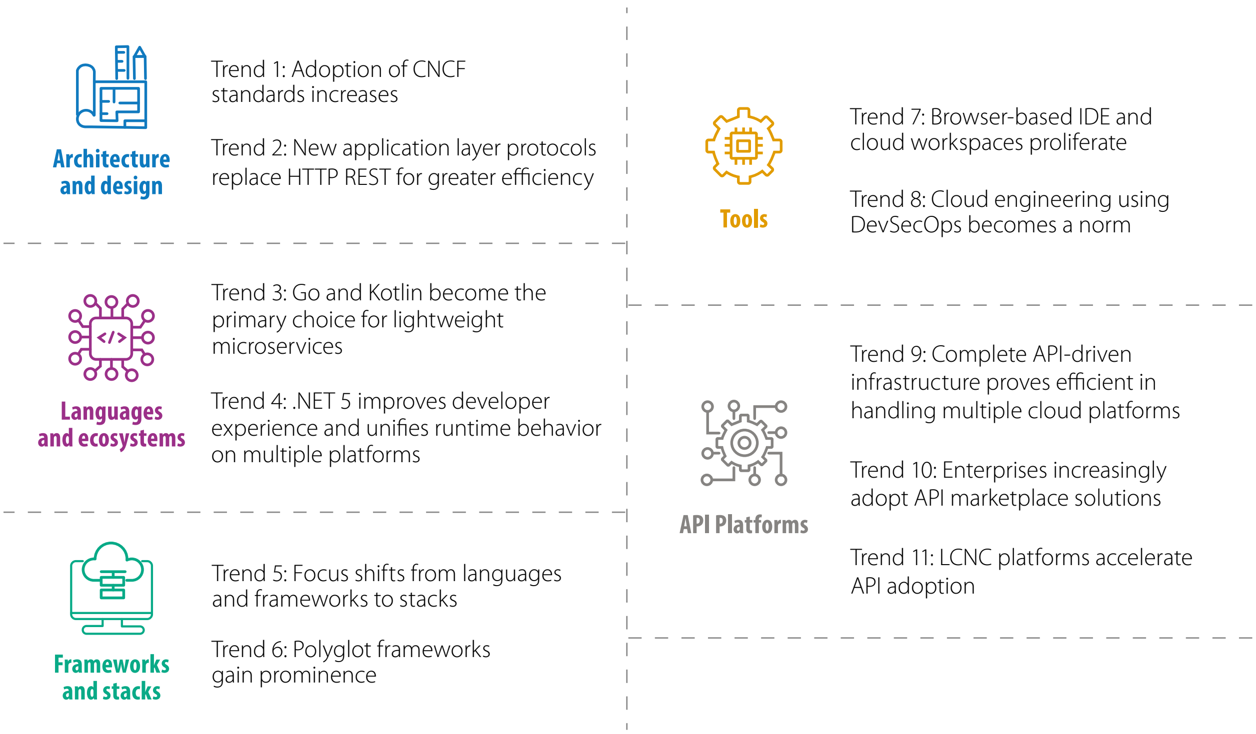
KEY PATTERNS	CHARACTERISTICS	Practices, Frameworks, Governance
• Reactive streams • Distributed application runtime • Beyond REST • AI-assisted dev • Micro-gateways • Mesh architecture • DevSecOps • GraphQL • Edge APIs	• Protocol agnostic and open specifications • Light-weight orchestration • Multi-channel and personalized experience • Business capabilities packaged as single unit • Reactive and resilient • Driven by industry standards	**H3** **PLATFORM ECONOMY AND HYPERSCALER ECOSYSTEM** Intelligent and Sentient-driven In an increasingly interconnected digital world, enterprises must strive for harmonization without conceding flexibility, agility and resilience. Enterprises increasingly depend on platforms and hyperscalers to help them establish intelligent and sentient-driven organizations.
• µServices • Low-code, no-code • Opensource • Container • Serverless • DevOps • Lightweight frameworks • APIs as a product	• Opinionated and driven by enterprise standards • 12-factor app principles • Automation • Agile practices • Hybrid deployments • Platform as a service	**H2** **API ECONOMY** Innovate, Transform, Reimagine Business Driven by an intensely competitive world and a constantly evolving digital ecosystem, enterprises have relied on innovation and transformation agendas to propel their businesses forward. The building blocks for a digital economy have undoubtedly been APIs as they are tailor-made for a high-performing environment.
• Monolithic • On-premise • Licensed products • Manual packaging	• Tightly coupled architecture • Waterfall SDLC model	**H1** **SERVICE-ORIENTED ARCHITECTURE** Better Connectivity

Source: Infosys

Enterprises need to consider the following five API subdomains to transition to H3:

1. **Architecture and design**
2. **Languages and ecosystems**
3. **Frameworks and stacks**
4. **Tools**
5. **API Platforms**

Figure 2. Key trends across API subdomains



Source: Infosys

ARCHITECTURE AND DESIGN

Architecture paradigms have significantly evolved. Initially (H1), the trends revolved around monolithic services and web applications, with centralized integration achieved through the enterprise service bus. H2 heralded an API-first approach with distributed microservices architecture, modular monoliths, and API-driven integration as well as agile scale and automation. In H3, microservices architecture (with requirements from the Reactive Manifesto) will continue to proliferate with advanced technology levers such as container orchestration platforms, serverless computing, data mesh, security mesh, event mesh, low-code, no-code (LCNC), cloud-native tools, and frameworks. The microservices ecosystem will further evolve with consortiums such as the Cloud Native Computing Foundation (CNCF).

Trend 1: Adoption of CNCF standards increases

The microservices architecture helped achieve on-demand elasticity and scalability of APIs for both on-premises and public hyperscaler infrastructure. APIs and microservices had to deploy hybrid infrastructure in addition to serverless infrastructure to support evolving requirements. This highlighted the necessity to standardize orchestration, container management, cluster management, and circuit breaker and monitor hybrid and serverless infrastructure. The CNCF curates and promotes open-source projects that enable modern, cloud-native applications. The industry now moves toward the adoption of projects in the CNCF landscape to quickly build open-source, cloud-native, LCNC tools, and agnostic applications.

Modern practices (microservices, monitoring, logging), packaging (containerization, orchestration), and automation (DevOps-based pipelines) are crucial to cloud-native solution delivery. The CNCF acts as a repository for trusted open-source projects such as Kubernetes, HELM, Jaeger, and Istio that are used in several deployments today.

Enterprises must work toward standardizing cloud-native development. The CNCF cloud-native landscape is a good reference to identify and use appropriate building blocks. Technology leaders like Google, Microsoft, Amazon, and Cisco are members of the CNCF. Microsoft's recent distributed application runtime (DAPR) framework (a portable, event-driven runtime building block for microservices) already incorporates the CNCF interactive landscape to help build cloud-native solutions.

> An American consumer goods corporation partnered with Infosys to modernize and develop a next-generation handheld platform for in-store order capturing and customer experience services. The platform followed an API-driven and domain-led design approach. It was developed using cloud-native, containerized microservices, and event mesh technology stacks. The solution was designed, built, and rolled out to sales representatives in just 16 months, reducing overhead costs in running handheld devices and enabling real time integration of data and insights. The platform was expanded to run in an Active-Active mode to achieve higher efficiency and meet growing business demands. It also enhanced resiliency for disaster management and data recovery.

Trend 2: New application layer protocols replace HTTP REST for greater efficiency

Hyperscaler adoption has enhanced focus on security, performance, lightweight containers, and availability. The APIs and microservices should support the hybrid user interface/user experience ecosystem in addition to serverless solutions. This has brought a new requirement to look beyond TCP/IP, HTTP protocols. Previously, interservice communications in the microservices world were primarily REST, despite their complexity and inefficiencies in certain use cases. Microservices solutions increasingly use new application layer protocols like Google's Remote Procedure Call (gRPC) and RSocket for improved security and lightweight deployment images to support serverless needs.

Modern cloud-native systems need to support multiple application protocols in the context of use-case needs. A good example of a mixed implementation is the use of potentially different application protocols in query (REST/HTTP) and response (GraphQL) flows. A mix of application protocols (REST/gRPC/GraphQL) help improve efficiencies. Additionally, soon application protocols based on HTTP/3 will also be a part of the broader pool protocols.

> Infosys partnered with a manufacturing giant to architect and develop a multicloud microservices platform. The solution involved various modern application layer protocols other than HTTP REST (gRPC, event messaging) to integrate with services across the landscape.

LANGUAGES AND ECOSYSTEMS

Programming languages are the core of the technology landscape and include procedural, object-oriented, functional, imperative, and declarative languages. Several programming languages have evolved to work across different paradigms. Over the past decade, prominent languages such as Java, C#, Python, JavaScript, and C/C++ underwent significant changes to remain relevant in today's era of microservices and cloud-native, highly resilient applications.

Java virtual machine (JVM) has been a platform of choice for building cross-platform programming languages. Languages such as Scala, Kotlin, and Clojure are popular because of the Java ecosystem. These languages provide flexibility to use existing libraries and frameworks. Some other popular languages are .C#, C++, and F# that run on the .NET CLR ecosystem.

With the ECMAScript specification evolving every year, JavaScript adopts these specifications to become more impactful. The Node ecosystem and programming paradigm (functional, event-driven, prototypical inheritance, and the ability to run at the client and server sides) have boosted the pace of innovation. RUST and Go are some of the preferred system development languages.

Trend 3: Go and Kotlin become the primary choice for lightweight microservices

Go offers high runtime efficiency with strong memory safety, garbage collection, and structural typing. It is already a top choice for system design and is widely used for microservices. Infosys uses Go for projects with critical memory footprint.

Primarily evolved from Android developers, Kotlin, is gaining attention in the microservices domain due to its conciseness, interoperability, and safe programming features. Banking, telecom, and other sectors adopt these languages as part of their modernization programs to develop large, cloud-native, and scalable microservices.

Typically, Java, .NET, JavaScript, or Python, in combination with frameworks such as SpringBoot, Django, and Nameko, were used to develop microservices. The outcome was bulky applications that consumed significant memory and lacked resilience. With the advent of languages (such as Go) and polyglot VMs (such as GraalVM), application teams now have a choice of languages and access to advanced tools to debug, monitor, profile, and optimize resources consumption. Notably, polyglot VMs support multiple languages and libraries.

For many engineering platforms, Go has become a preferred language with its efficient memory management capabilities. Infosys DevOps platform, rebuilt on Go, can reduce the memory footprint threefold.

Trend 4: .NET 5 improves developer experience and unifies runtime behavior on multiple platforms

The release of .NET unified ASP.NET, .NET Core, Entity Framework Core, WinForms, Windows Presentation Foundation, Xamarin, and ML.NET and provide a single platform to build cross-platform applications.

The .NET framework was initially designed to build a robust framework for Windows-based desktop, web, and enterprise applications. The addition of .NET Core provided support for non-Windows environments, although it required different libraries to develop other applications for mobile, desktop, and Windows Communication Foundation. The new .NET 5 unified platform aims to provide a rich developer experience with high performance and scalable, consistent runtime behavior on multiple target platforms simultaneously.

.NET 5 takes the best of .NET Core, .NET Framework, Xamarin, and Mono to produce a single .NET runtime and framework that can be used everywhere.

FRAMEWORKS AND STACKS

As reactive and serverless architectures become a priority, cloud-specific frameworks (Sparta and Flogo Core), and modern Java frameworks (Quarkus and Micronaut) are fast emerging. CloudEvents and NATS are two prominent CNCF projects in the integration space. While CloudEvents standardizes the event data format and makes it vendor neutral, NATS provides a high-performance messaging system. As Kubernetes gains momentum, its native frameworks such as Camel-K, TriggerMesh, Kogito, and Zeebe provide integration and a workflow engine. Data serialization has seen advances with Google's protocol buffers (Protobuf) and Apache Avro, as these formats are much smaller and faster than the regular JavaScript object notation.

Trend 5: Focus shifts from languages and frameworks to stacks

With the evolving microservices landscape, more frameworks, messaging systems, transport layers, data serialization formats, APIs, etc., will emerge. Since these components constitute a whole stack of offerings, developers are less likely to choose individual languages and frameworks for their application development.

Earlier, the microservices component landscape offered few choices. Developers were forced to select individual languages and frameworks; web application stacks such as LAMP, WISA, MEAN, and PERN; and Netflix OSS microservices stacks. Some of these stacks are now obsolete, while others try to catch up with required augmentation with newer components to work properly. However, with today's ultralow latency, highly efficient data serialization, and API querying options, these stacks will become more powerful.

A large financial institution, in partnership with Infosys, developed the GRAND stack framework to modernize its payment services technology. This stack uses native end-to-end synergies in place of individual programming languages.

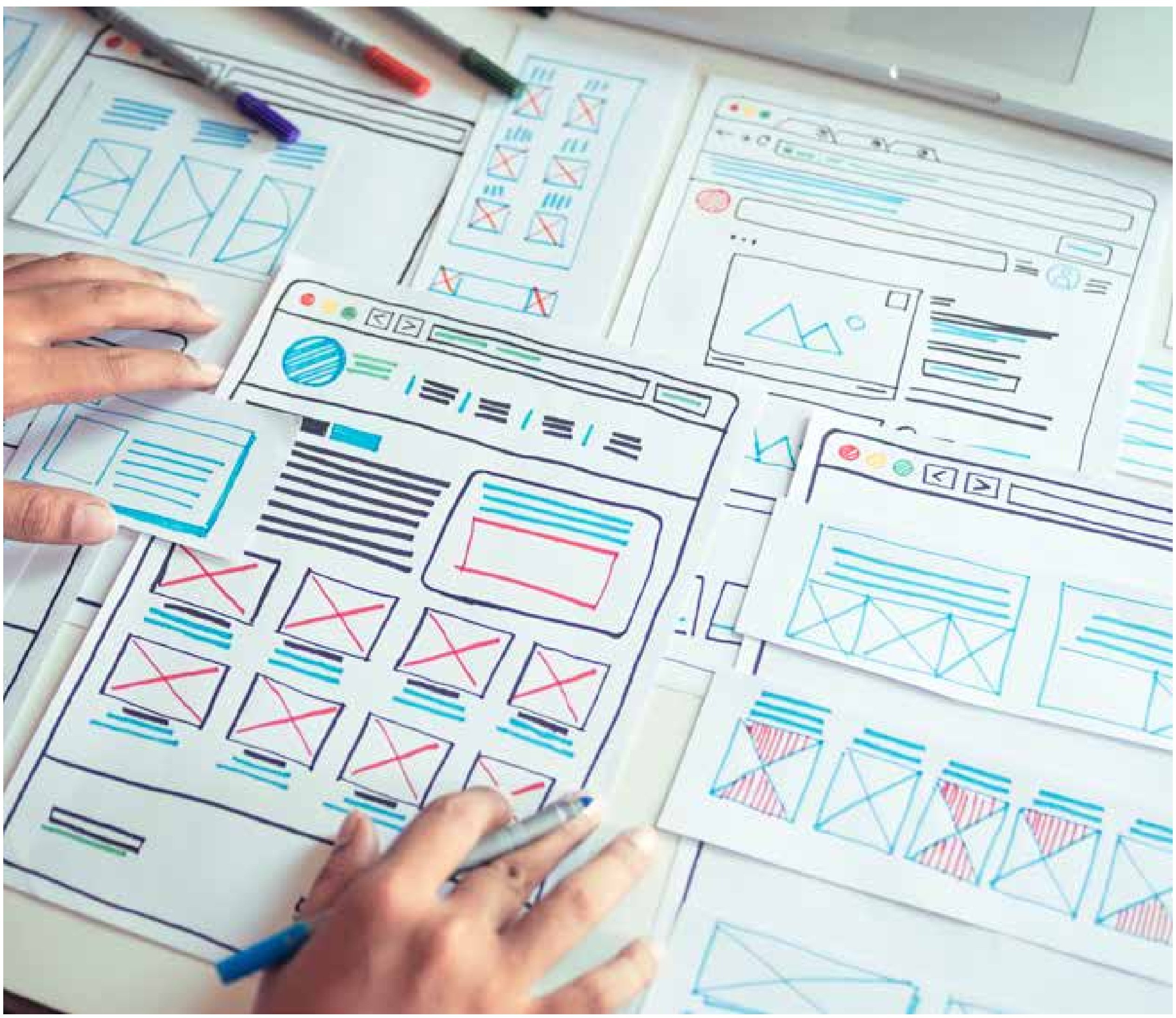
Trend 6: Polyglot frameworks gain prominence

Modern Java frameworks that offer fast throughput and nominal startup time (e.g., Quarkus, Micronaut, and Helidon) have become instrumental in robust microservice and serverless application builds. These frameworks support Amazon Web Services (AWS) Lambda, and Azure Functions, as well as non-blocking reactive styles of programming and declarative types.

Earlier, the absence of appropriate dependency injection standards, JVM-related limitation on the modules and cloud-native features drove the community/enterprise to use Spring as a de facto standard of development despite Spring lacking the memory efficiency.

New frameworks are polyglot in nature, with serverless extensions and Kubernetes support. While polyglot VM offers ahead-of-time compilation, the frameworks support compile-time dependency injection and greatly enhance the developer experience and runtime performance.

> Infosys has used Reactive Frameworks in several engagements. It is now exploring the use of Quarkus, which focuses on non-blocking, fast throughput, and minimal startup time to handle massive concurrent sessions.



TOOLS

Many tools have already migrated or in the process of migrating to the cloud. They started with the integrated development environment (IDE) and evolved to DevSecOps, where security has advanced to the build phase that includes image scanning for Kubernetes containers. IDE transformation will consist of offline IDEs like Microsoft Visual Studio and Eclipse and their browser versions, Visual Studio Codespaces, and Codenvy. The cloud-hosted and browser-based IDEs make it easier for teams to collaborate and integrate with DevOps tools that are also hosted in the cloud. Similar is the case with security from the extensive use of open-source software and Docker images in Kubernetes. Security scanning has shifted ahead to the build and deployment phases, where third-party, open-source, software vulnerability scans and Kubernetes containers are carried out.

Trend 7: Browser-based IDE and cloud workspaces proliferate

In-browser IDEs help with mobility, portability, and better team-level collaboration. Efforts are underway to eliminate any system constraints. IDEs also possess AI-assisted intelliSense features that utilize the developer's current context and patterns based on thousands of highly rated open-source projects on GitHub. Thanks to increased cloud adoption, the browser-based IDEs such as Codespaces and Codenvy have gained traction.

> Infosys is currently piloting cloud workspaces and in-browser IDEs with advanced in-built AI capabilities that enable rapid application development and enhance team productivity and collaboration.

Trend 8: Cloud engineering using DevSecOps becomes a norm

Agile methodology and microservices have triggered frequent builds and deployment. As a result, open-source components in the software and image containers deployed in Kubernetes clusters remain vulnerable. However, tools such as NeuVector make the "shift left" security to the build phase possible, and the scans occur at a faster pace.

The waterfall model and monoliths in the past did not require intensified security. But, as more companies switch to the cloud, security will move to the front of the line in the build phase. Moreover, companies will need to research the best tools available to analyze open-source software vulnerabilities and scan images deployed to Kubernetes.

A global company, in partnership with Infosys, implemented greenfield DevSecOps and onboarded over 120 applications across Java, .NET, and JS applications. As part of the project, they automated the build and deployment to production setup using pipeline-as-code with automated web-security and vulnerability testing.

API PLATFORMS

Over two-thirds of enterprises have adopted more than two public hyperscaler providers in addition to the on-premises data center that includes private cloud as its infrastructure strategy. Automation at all levels is the prevalent trend in platforms. It can be summarized as "<everything> as code," as every layer in the architecture is now scripted and stored in source control. Infrastructure has evolved to become more API-driven and scriptable, enabling to document, share, and discuss every intention as code.

Trend 9: Complete API-driven infrastructure proves efficient in handling multiple cloud platforms

In today's dynamic environment, API-driven infrastructure provides the ability to set up infrastructure in the cloud or on-premises with the use of programming languages and libraries. The adoption of Terraform and Ansible in the industry has fueled more APIs in the infrastructure as code. In the past, inefficient methods such as shell scripting involved significant manual efforts to set up a platform. All hyperscaler management console platforms now provide APIs to manage the infrastructure. For example, container orchestration, gateways, caching, and more will be stored, versioned, upgraded, and maintained as code.

Many hyperscalers have come up with multicloud management platforms, such as AWS Outpost, Azure Arc, and Google Anthos. These platforms provide APIs to manage, provision, and audit their hyperscaler infrastructure or on-premises infrastructure using Ansible, Terraform, Ballerina, Pulumi, and CloudFormation.

Most companies that work with the public cloud, Kubernetes, or photonic crystal fiber-based tools, use at least one infrastructure scripting language. This trend will further gain momentum, as tools and innovation are utilized appropriately to define entire data centers as code.

Infosys has built a multicloud environment management tool using open-source technologies and frameworks. The polycloud platform is powered using standard-based abstraction through API. This unified cloud management (single pane of glass) offers seamless control, provisions, and management of multiple clouds. It also minimizes vendor lock-in span. It enables easier workload migration across multiple clouds.

Trend 10: Enterprises increasingly adopt API marketplace solutions

The API marketplace platform solutions enable enterprises to integrate a single governance framework for multi-API gateways, and accelerate development with prebuild standard and domain APIs to expedite digital transformation.

API marketplaces are allowing businesses to aggregate APIs and create a space for developers to upload, distribute, and monetize APIs quickly. These marketplaces allow consumers to easily discover and implement APIs. The marketplace lists APIs based on categories and classifications which makes the discovery even easier. It also makes it easier to compare different APIs on several parameters such as costs, functionalities, endpoint etc.

Infosys has built One Cloud Marketplace and One Cloud Sandbox solutions to build a strong partner ecosystem. These solutions enable enterprises to experiment and innovate with multicloud deployment models and prebuild accelerators and industry standard APIs. They also offer benefits like API governance and automations, less lead time, better developer experience, and higher productivity. This helps enterprises achieve an API-first strategy with a product-centric approach.

Trend 11: LCNC platforms accelerate API adoption

LCNC platforms help integrate and automate workflows with little or no programming knowledge. These platforms support a wide range of out-of-box integration capabilities and connectors that can be built and deployed through drag-drop code features and configurations. These platforms enable customers to accelerate their API and microservices journey, either on-cloud or on-premises.

Infosys accelerators provide frameworks and solutions for integrations, microservice development, and API marketplace. Infosys has partnered with several vendors to use LCNC solutions for API delivery. These solutions implement microservice acceleration platforms and enable value proposition for the customer modernization journey.

Glossary

Abbreviation/acronym	Full-form
AI	Artificial intelligence
API	Application programming interface
CNCF	Cloud native computing foundation
DAPR	Distributed application runtime
DevOps	Development and operations
DevSecOps	Development, security, and operations
gRPC	Google's remote procedure call
HTTP	Hypertext transfer protocol
IDE	Integrated development environment
JVM	Java virtual machine
LCNC	low-code, no-code
ML	Machine learning
PaaS	Platform-as-a-service
REST	Representational state transfer
SDLC	Software development life cycle
SOA	Service-oriented architecture
TCP/IP	Transmission control protocol/Internet protocol
VM	Virtual machine

Advisory Council

Mohammed Rafee Tarafdar

SVP and Chief Technology Officer

Dinesh Rao

EVP - Head Global Services - Enterprise Package Application Services

Srinivas Kamadi

SVP - Service Offering Head – Enterprise Integration & Services

Manas Kumar Sarkar

VP - Delivery Head – API Economy

Naresh Choudhary

VP - Reuse and Tools - Head, QLTY

Contributors

Anirban Ghosal

Ajay N

Allahbaksh Mohammedali Asadullah

Charudatta Joshi

Dinesh Nagabushanam

Gaurav Sharma

Kannan Narayanan

Krishna Kanth B. N.

Krishna Markande

Krishnakumar V

Mohammed Rafee Tarafdar

Naresh Duddu

Priyapravas

Saurav Kanti Chandra

Shreyash Subhash Mantri

Suraj Nair

Venkata Lakshminarayana Indraganti

Vijay Kannan

Vijayaraghavan Varadharajan

Vineesh Thomas Devasia

Vinod Sivashankaran

Vishwanath Taware

Producers

Ramesh N

Infosys Knowledge Institute
ramesh_n03@infosys.com

Abhinav Shrivastava

Infosys Knowledge Institute
abhinav.s08@infosys.com


Current account
1.297
Deposit account
13.500

Notes :

About Infosys Knowledge Institute

The Infosys Knowledge Institute helps industry leaders develop a deeper understanding of business and technology trends through compelling thought leadership. Our researchers and subject matter experts provide a fact base that aids decision-making on critical business and technology issues.

To view our research, visit Infosys Knowledge Institute at infosys.com/IKI or email us at iki@infosys.com.

For more information, contact askus@infosys.com