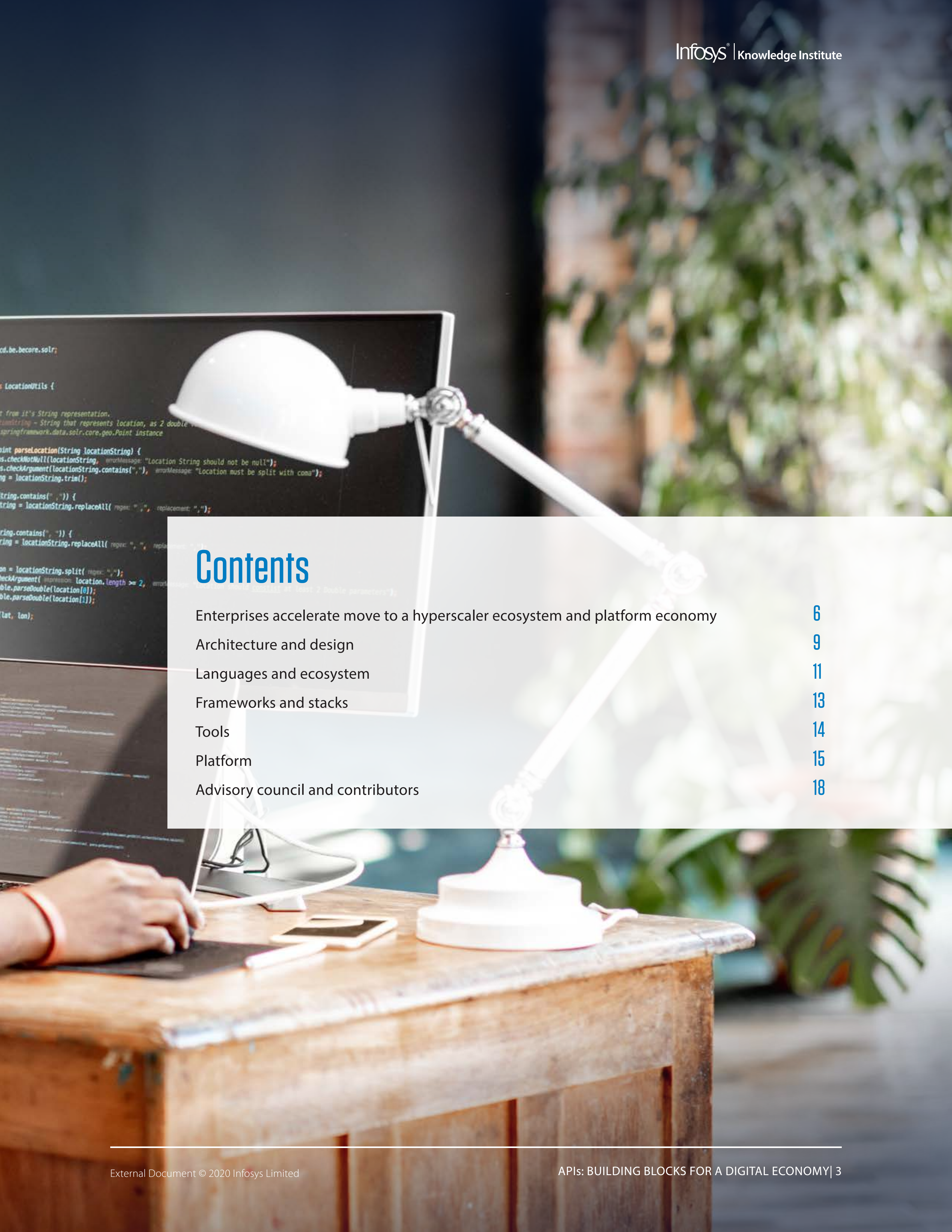
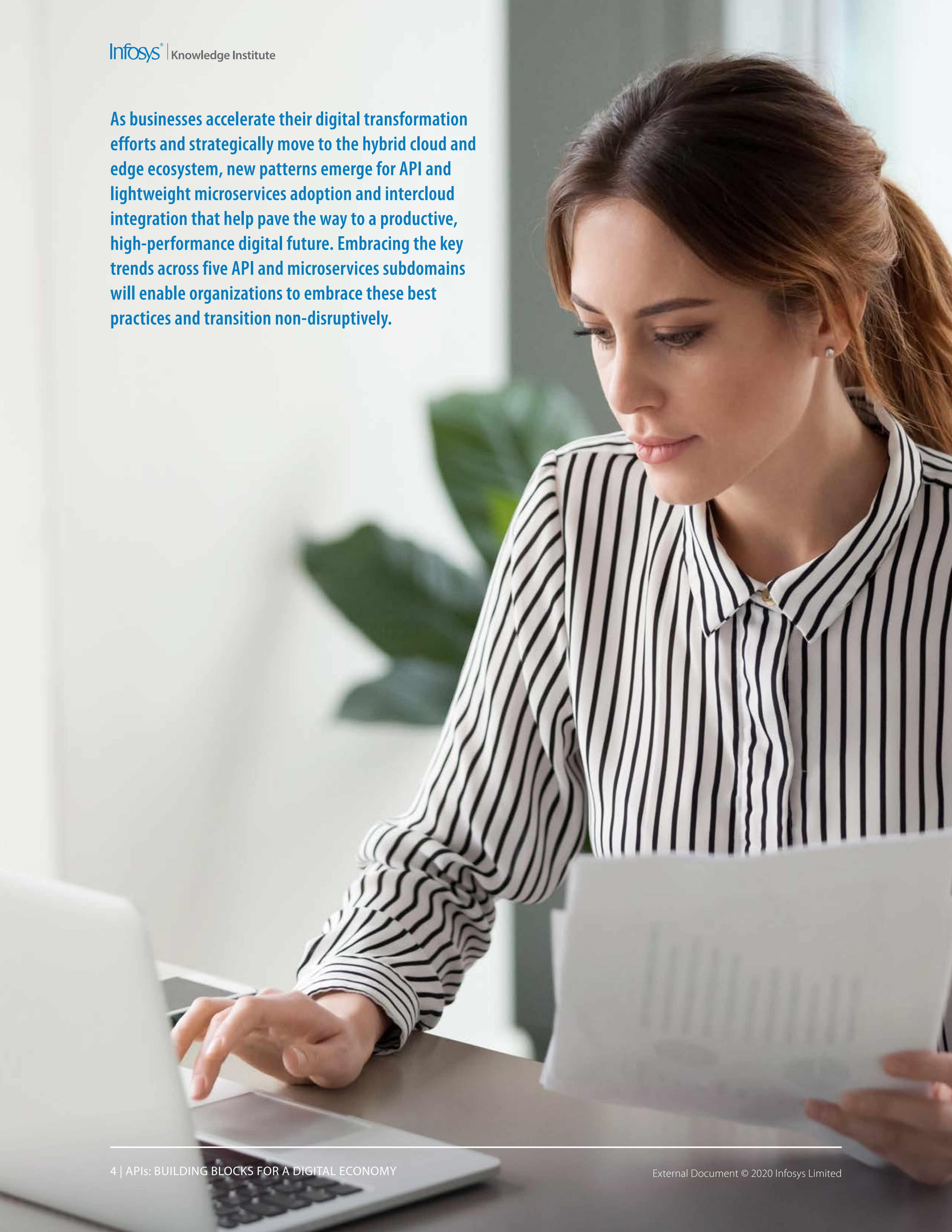
Infosys®
Navigate your next

APIs: BUILDING BLOCKS FOR A DIGITAL ECONOMY

Infosys® | Knowledge Institute



Contents



As businesses accelerate their digital transformation efforts and strategically move to the hybrid cloud and edge ecosystem, new patterns emerge for API and lightweight microservices adoption and intercloud integration that help pave the way to a productive, high-performance digital future. Embracing the key trends across five API and microservices subdomains will enable organizations to embrace these best practices and transition non-disruptively.

Enterprises rely heavily on APIs and microservices to build and connect applications, a fundamental requirement for their digital transformation. In today's application development ecosystem, API is the visible layer while distributed application runtime, containers, interfaces, integration, serverless computing, DevSecOps tools and platforms form a strong backbone. APIs are layered over on-premise systems to promote partner connectivity and developer productivity.

In the new normal, enterprises must re-examine the collaboration and connectivity between applications, data and processes and identify new ways to tap into their data founts to make information technology (IT) operations more resilient and agile. Organizations must also decompose and deploy applications as independent units to accommodate remote work. 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 can run on any cloud platform. Other outcomes include AI-assisted development to increase productivity, DevSecOps to enable extreme automation and to seek new ways of connecting applications.

For businesses to be resilient, they must transition to a hyperscaler ecosystem and platform economy now.

Enterprises accelerate move to a hyperscaler ecosystem and platform economy

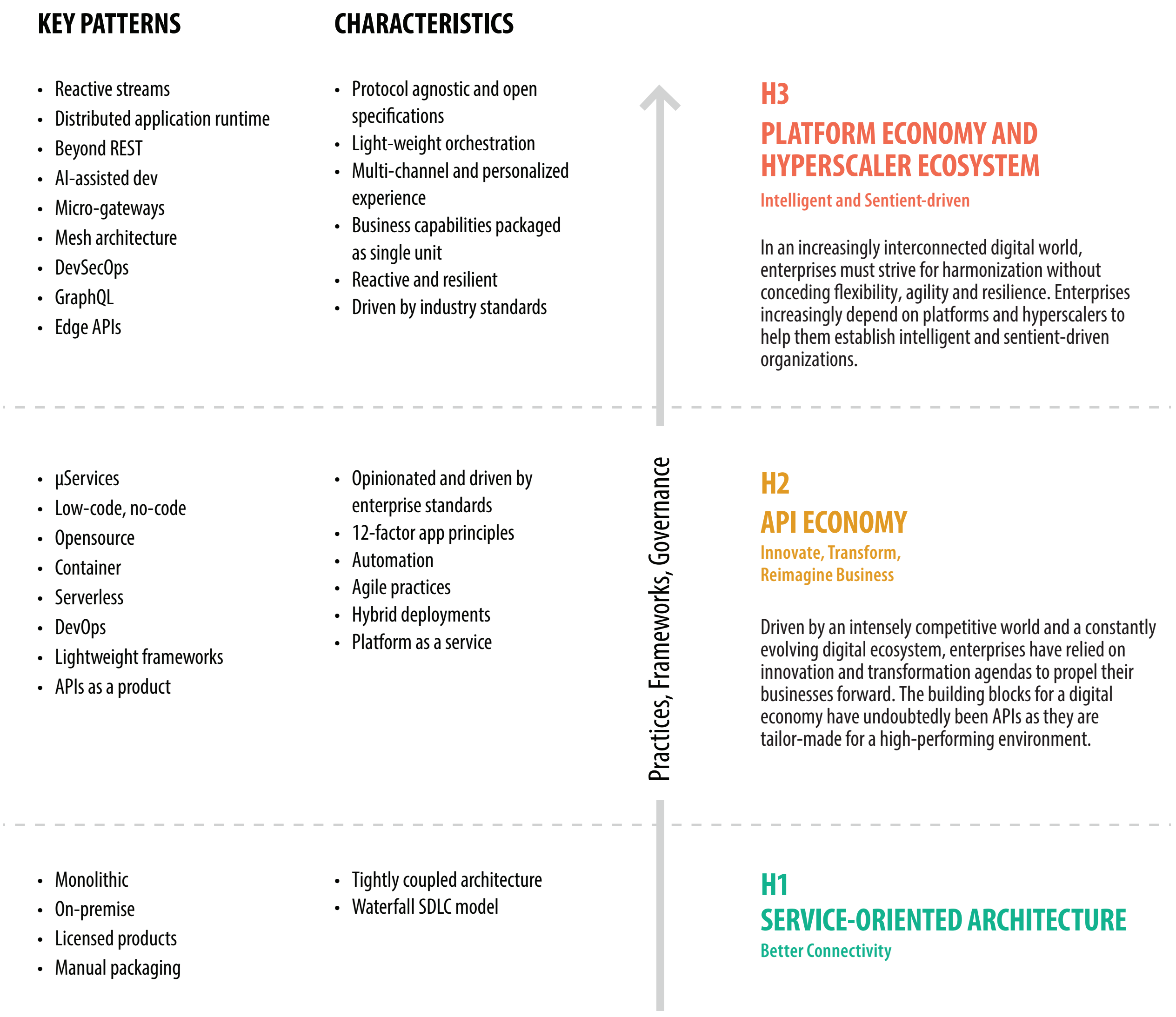
Digital platforms are now the de facto way to conduct business, find entertainment and interact with others. In our digital world, people have higher expectations like personalized interactions, real-time responses, and an omnichannel experience. Only companies that evolve their business models and digitally transform their business can succeed in this new paradigm.

IT teams can break loose from the tightly coupled and monolithic chains of a service-oriented architecture (SOA) culture if they employ key characteristics of the API economy – automation, platform-as-a-service (PaaS), agile practices and 12-factor app principles. For instance, platforms eliminate manual intervention and provide access to prebuilt components, which boost productivity and increase efficiencies.

The SOA wave (horizon 1, H1) began in the 1990s to establish better connectivity through standardized interfaces between systems. However, the SOA could not eliminate monolithic and on-premise systems that dragged down performance and inhibited fast responses. As enterprises embarked on digitization, they switched to the API economy (H2).

Indicators for the next horizon point to a platform economy and hyperscaler ecosystem (H3). In H3, the focus will be to maximize flexibility, agility, resilience and, at the same time, deliver business platforms that are intelligent and provide mobile- and AI-first experiences through hybrid APIs, microservices and edge APIs. To manage this, enterprises will have to make the most of these new trends.

Figure 1. Adapting to market dynamics: the three horizons


Practices, Frameworks, Governance

KEY PATTERNS	CHARACTERISTICS	
• 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

To transition to H3, enterprises will have to consider these API subdomains as part of their journey:

1. **Architecture and design**
2. **Languages and ecosystem**
3. **Frameworks and stacks**
4. **Tools**
5. **Platform**

Figure 2. Key trends across API subdomains

Architecture and design

Going beyond monoliths to microservices

Trend 1: Enterprises adopt CNCF cloud-native development standardization

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

Languages and ecosystem

Programming paradigm expansion

Trend 3: Go and Kotlin emerge as primary choices for lightweight microservices with a low memory footprint

Trend 4: .NET 5 expands the developer experience and unifies runtime behavior on all platforms

Frameworks and stacks

The transition from web-native to cloud-native

Trend 5: Focus shifts from languages and frameworks to stacks

Trend 6: Polyglot frameworks take center stage

Tools

The shift from on-premise to the cloud

Trend 7: Browser-based IDE and cloud workspaces proliferate

Trend 8: Cloud Engineering using DevSecOps is the norm

Platforms

Progression from monolithic services to flexible infrastructure

Trend 9: 100% API-driven infrastructure supports multiple cloud platforms

Trend 10: Hybrid cloud models employ hybrid API management

Source: Infosys

ARCHITECTURE AND DESIGN

Going beyond monoliths to microservices

Architecture paradigms continue to see a significant shift. In H1, the trends revolved around monolithic services and web applications with centralized integration achieved with enterprise service bus (ESB). H2 heralded an API-first approach with distributed microservices architecture 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 the help of newer technology levers such as container orchestration platforms, serverless computing, cloud-native tools and frameworks. We expect increased standardization of the microservices ecosystem driven by consortiums like the Cloud Native Computing Foundation (CNCF).

Trend 1: Enterprises adopt CNCF cloud-native development standardization

The microservices architecture helped achieve on-demand elasticity and scalability of the APIs for both on-premise and public hyperscaler infrastructure. APIs and microservices had to deploy on hybrid infrastructure in addition to serverless infrastructure to support the enterprise requirements. This brought the focus to the standardization of orchestration, container management, cluster management, circuit breaker and monitoring on 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 (trust) to quickly (speed) build open-source-based (freedom), cloud-native 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, which are used in many deployments today.

Enterprises must work toward standardized cloud-native development. The CNCF cloud-native landscape acts as a good reference point to identify and use appropriate building blocks. Technology leaders like Google, Microsoft, Amazon, and Cisco are members of the CNCF. In fact, 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.

Infosys partnered with one of the world's leading car manufacturers to architect and develop a next-generation connected car platform for safety and security services. The platform was based on cloud-native and cloud vendor-agnostic technology stacks, which allowed the customer to migrate smoothly across cloud vendors.

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

Industry adoption of the hyperscaler brought the critical focus on security, performance, lightweight containers and availability. The APIs and microservices need to support the hybrid UI/UX ecosystem in addition to the serverless solutions. This 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-based 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.

With modern cloud-native systems, there will be a 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 flow (REST/HTTP) and response flow (GraphQL). Today, a mix of application protocols (REST/gRPC/GraphQL) work together to improve efficiencies.

Additionally, with the Internet Engineering Task Force's work on the draft HTTP/3 specifications, it will not be long before application protocols based on HTTP/3 also enter the mix.

Infosys partnered with a manufacturing giant to architect and develop a multi-cloud 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 ECOSYSTEM

Programming paradigm expansion

Programming languages form the core of the technology landscape and include procedural, object-oriented, functional, imperative and declarative languages. Many programming languages are evolving into multi-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 programming languages which are cross-platform. Powerful languages such as Scala, Kotlin and Clojure are popular because of the Java ecosystem. Also, these languages provide flexibility to use existing libraries and frameworks. .C#, C++ and F# are examples of popular languages which run on the .NET CLR ecosystem and have evolved over time.

With the ECMAScript specification evolving every year, the JavaScript adopts these specifications to become better and more powerful. The Node ecosystem and programming paradigm (functional, event-driven, prototypical inheritance and ability to run both at the client and server side) have boosted the pace of innovation.

RUST and Go are primarily seen as system development languages that provide the best performance.

Trend 3: Go and Kotlin emerge as primary choices for lightweight microservices with a low memory footprint

With strong memory safety, garbage collection and structural typing, Go offers high runtime efficiency. It is already a top choice for system design but also is widely used for microservices. At Infosys, we use Go for many projects where a memory footprint is critical. Kotlin, which evolved primarily from developers using Android, is seeing a shift to the microservices domain due to the conciseness, interoperability and safe nature of its programming. 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 virtual machines (VMs) like GraalVM that support multiple languages and libraries, application teams now have a choice of languages and access to advanced tools to debug, monitor, profile and optimize resource consumption.

Infosys DevOps platform (IDP) is rebuilt on Go, which helps reduce the memory footprint threefold. With its efficient memory management capabilities, Go is now becoming a preferred language for many engineering platforms.

Trend 4: .NET 5 expands the developer experience and unifies runtime behavior on all platforms

The .NET community eagerly awaits the release of .NET 5, which will unify 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 types of applications such as 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 at the same time.

.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

The transition from web-native to cloud-native

With reactive and serverless architectures a priority today, cloud-specific frameworks such as Sparta and Flogo Core, and modern Java frameworks such as 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, 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 regular JavaScript object notation.

Trend 5: Focus shifts from languages and frameworks to stacks

Expect frameworks, messaging systems, transport layers, data serialization formats, APIs and more to emerge with the new microservices landscape. Since these components now constitute a full-stack, developers are less likely to choose individual languages and frameworks for their application development, as was the case in the past.

Previously, 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 than ever before.

Infosys partnered with a large financial institution to modernize their payment services technology with GRAND stack, which, in turn, used native E2E synergies in place of individual programming languages.

Trend 6: Polyglot frameworks take center stage

Modern Java frameworks that offer fast throughput and nominal startup time (e.g., Quarkus, Micronaut and Helidon), will be 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.

In the past the absence of appropriate dependency injection standards (JSR), JVM related limitation on the modules and cloud native features drove the community/enterprise to use Spring as defacto 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 and is exploring the use of Quarkus, which focuses on non-blocking, fast throughput and minimal startup time to handle massive concurrent sessions.

TOOLS

The shift from on-premise to the cloud

Many tools have migrated to the cloud, and others are in process. 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 as well as 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, and efforts are underway to eliminate any system constraints. The 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, Codenvy based on Eclipse Che 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 is the 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. Tools such as NeuVector, however, 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.

Infosys partnered with a global company to implement greenfield DevSecOps and onboard 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.

PLATFORM

Progression from monolithic services to flexible infrastructure

Over two-thirds of enterprises have adopted more than two public hyperscaler providers in addition to the on-premise 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, which enables us to document, share and discuss every intention as code.

Trend 9: 100% API-driven infrastructure supports multiple cloud platforms

In today's dynamic environment, API-driven infrastructure provides the ability to set up infrastructure in the cloud or on-premise with the use of programming languages and libraries. Adoption of Terraform and Ansible in the industry has fuelled 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 the APIs interface 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 their multi-cloud management platforms, such as AWS Outpost, Azure Arc and Google Anoths. These platforms provide the APIs interface to manage, provision and audit their hyperscaler infrastructure or on-premise 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 gain momentum as tools and innovation are utilized appropriately to define entire data centers as code.

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

Trend 10: Hybrid cloud models employ hybrid API management

The API management tool provides the ability to discover the service, integrate to create security tokens and enable monitoring. Many enterprises now have applications distributed across hybrid infrastructures, public hyperscalers and on-premise data centers. The applications integrate through multiple API gateways, such as one for the public hyperscaler and one for the on-premise data center to support business capabilities. This brings additional latency in the roundtrip response time of the APIs. API management tools provide the critical functions of service registry, enable client-side service discovery and manage service inventory with their instance and location.

The Apigee and MuleSoft tools provide the capability to support hybrid API management. These platforms support all dimensions of API management, developer portal allowing developers to register with the system, create APIs, provision credentials, subscribe to APIs, and view documentation. Similarly, the Administration Management Module provides the capability to manage the service registry, user management, security - TLS/SSL & SAML, and API access.

Infosys implemented Hybrid API management for one of the largest equipment manufacturers to support the applications deployed on AWS and in a private cloud using Apigee.

Advisory council

Mohammed Rafee Tarafdar
SVP and Unit Technology Officer

Dinesh Rao
EVP - Head Global Services -
Enterprise Package Application Services

Srinivas Kamadi
VP - Service Offering Head –
Enterprise Integration & Services

Manas Kumar Sarkar
AVP - Delivery Head – API Economy

Naresh Choudhary
VP - Reuse and Tools - Head, QLTY

Contributors

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

Vittal Setty

Producer

Ramesh N
Infosys Knowledge Institute
ramesh_n03@infosys.com

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.

Infosys®
Navigate your next

For more information, contact askus@infosys.com

Infosys.com | NYSE : INFY

Stay Connected