

Application programming interface: cornerstone for digital business

APIs are proliferating, as they are the cornerstone for digital business. The major challenge for enterprises is to cope with the increasing demand around API ecosystem. This typically includes, creating APIs rapidly, controlling them by managing a catalogue and enforcing a powerful security framework and consuming them effectively with detail insight on the usage and consumption behaviour.

API Lifecycle Management is the process of designing, publishing, documenting and analysing APIs in a secure environment. It encompasses following major aspects –

- API design & governance - design, publish and deploy
- API store - a store or catalogue for internal and/or external consumers.
- API exposure - gatekeeper for all APIs by enforcing relevant API security policies
- API analytics - monitor API usage, load, transactions, historical data and other metrics
- API monetization – monetize APIs based on defined plan and usage
- API marketplace – collaborate with partners to build solutions and monetize

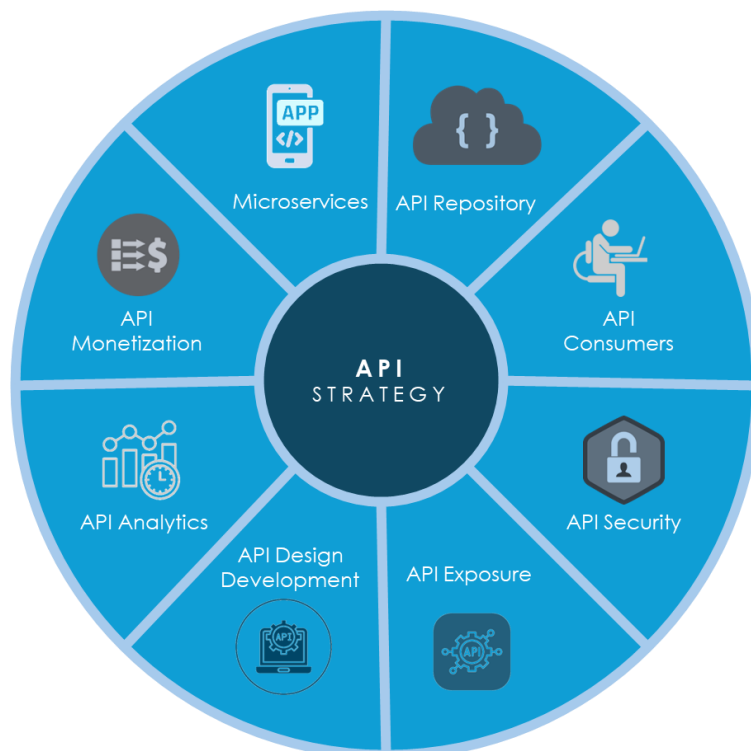


Fig:1 – Considerations for API Strategy

Market Analysis

The Application Programming Interface (API) management market size is expected to grow from USD 7.6 billion in 2024 to USD 16.9 billion by 2029, at a Compound Annual Growth Rate (CAGR) of 17.1% during the forecast period. The growing demand for modern APIs is to seamlessly connect various applications and devices as well as advanced analytics to get real-time insights on API performance and usage are some of the major factors driving the growth of the API management market.

North America is leading the API Management market, and the trend is to continue until 2029 with a strong demand for complex API management and security requirements. Also, Integration of Artificial Intelligence-Powered Technology with API Management System to Drive Market Growth.

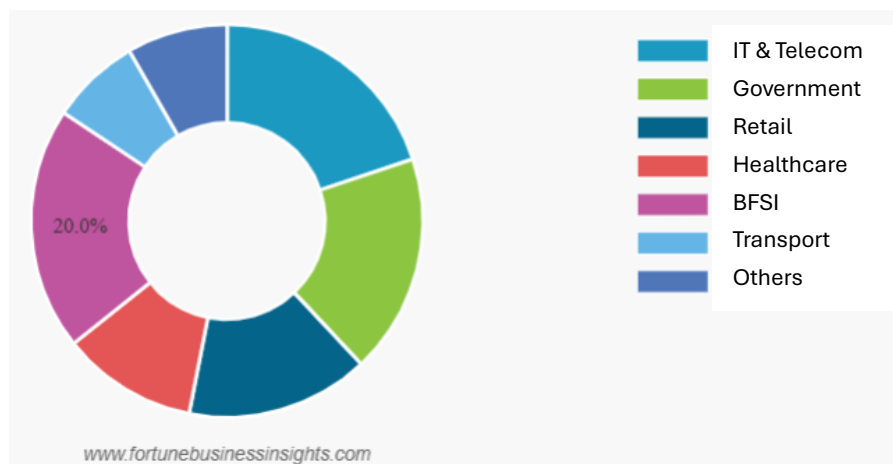


Fig: 2 – Global API Management market share across industry verticals

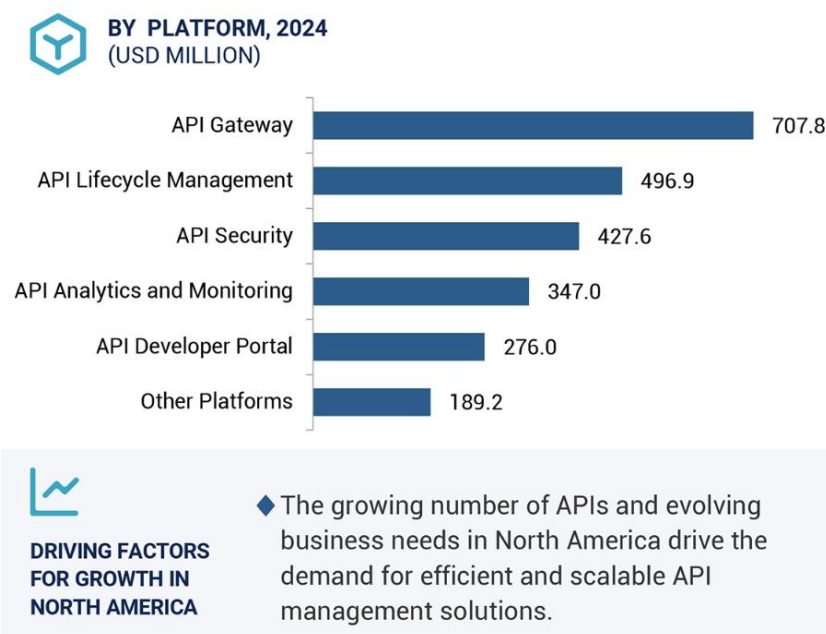


Fig: 3 – Growth areas of API Lifecycle in North America market

API Enablement & Lifecycle Management

API lifecycle management is all about overseeing of all aspects of managing an API or API program. It manages starting from development to deployment and beyond. API lifecycle management allows enterprises to attract the right app developers, troubleshoot problems, enhance their APIs and make better business decisions related to their API programs. API management foster collaboration by sharing mocked services and code snippets to validate functionality. Manage APIs monitor and analyse usage, control access, and protect sensitive data with security policies.

API management is the process of designing, publishing, documenting and analysing APIs in a secure environment.

It encompasses following major components –

- API design & governance - design, publish and deploy
- API gateway - gatekeeper for all APIs by enforcing relevant API security policies
- API store - a store or catalogue for internal and/or external consumers.
- API analytics - monitor API usage, load, transactions, historical data and other metrics

API management services have a wide scope of usage, ranging from feasibility analysis, strategy and implementation, and valuations and development of modern APIs to cater business-specific needs of organizations. API management services help businesses bridge legacy systems to modern applications and deliver connected experiences to clients and partners while also maintaining control and reliability of backend systems. This has created opportunities for vendors to provide services to enterprises across different industry verticals and help them deal with complexities while configuring the API management solution.

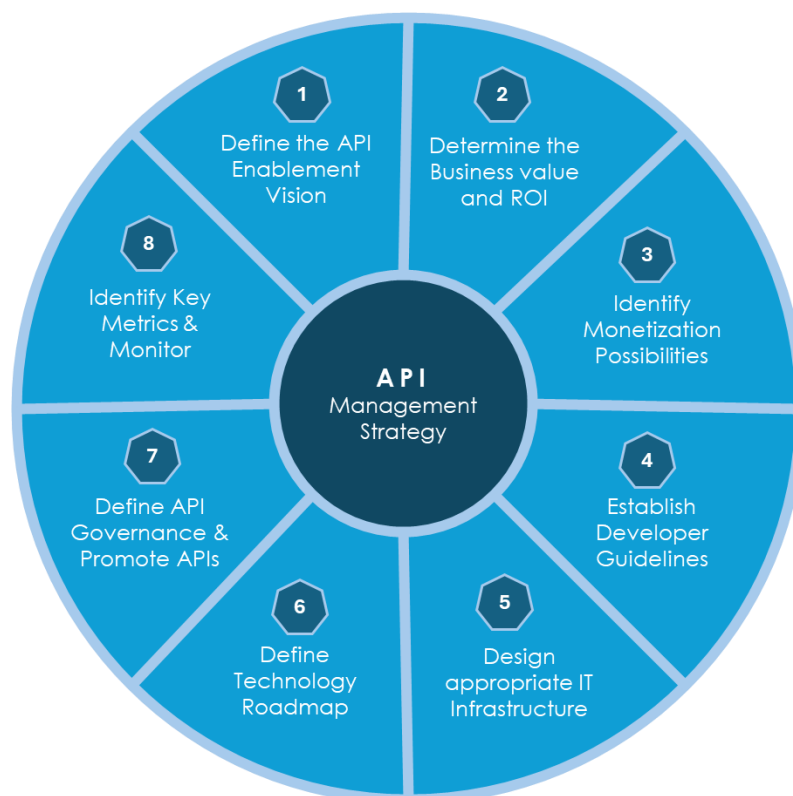


Fig: 4 – Enterprise API Management Strategy

Core principles of API management -

- Maintaining security at every stage of the API lifecycle
- Onboarding and managing partners
- Monitoring analytics and delivering data-driven insight
- Browsing, creating, updating, and managing APIs
- Establish API design governance

API Lifecycle stages

There are three major components of an API lifecycle (create, control, consume) with analytics at the centre of it. Each component of the lifecycle can be broken down into several stages.

- **Building:** Once the strategy and plan are ready, it's time to build the API. In this phase the API is designed in Open API standard, so that it will be seamless for API consumers.
- **Testing:** DevOps infrastructure and instrumentations can be used to test the API. In this phase API performance and test API throttling are checked.
- **Publishing:** Once tested the API can be published.
- **Securing:** API security is a major concern today. It is important to provide data related to the context and privileges of the consumer
- **Managing:** Once the API is published it is to be maintained, and integrity should be controlled by the publisher.
- **Onboarding:** When APIs are offered for public or private consumption, onboarding is an important component so that developers understand the key capabilities.
- **Analysing:** API Analytics help understand the digital business and catch abnormalities before they impact the business.
- **Promoting:** There are several ways to market our API including listing it in an API marketplace.
- **Monetizing:** One can choose to offer APIs for free or when there is an opportunity for it, same can be monetized and drive additional revenue for the business.
- **Retirement:** Retiring APIs is the last stage of an API lifecycle and happens for a variety of reasons including technology changes and security concerns.

Building API

Design and build of API solution should ideally be based on implementation roadmap (Maturity assessment) following Microservice Architecture principles and following Open API specifications.

Key components of API solution should include:

- **API Gateway:** Central point for all abstracted APIs
- **Developer portal:** Developers can register, access details & monitor performance

- Management portal: For policy & access control
- Implementation of key API management functionalities:
- API subscription management
- API Repository and API Orchestration
- Building access control mechanism in terms of usage limits, traffic
- Set up of API LCM for support and enhancement of APIs

Testing API

A microservices architecture involves many moving parts with different guarantees and failure modes. Testing and verification of these systems are significantly more nuanced and complex than testing a traditional monolithic application. An effective test strategy needs to account for both testing individual services in isolation and the verification of overall system behaviour. Testing can be broadly broken down into two categories: pre-production testing and monitoring and testing in production.

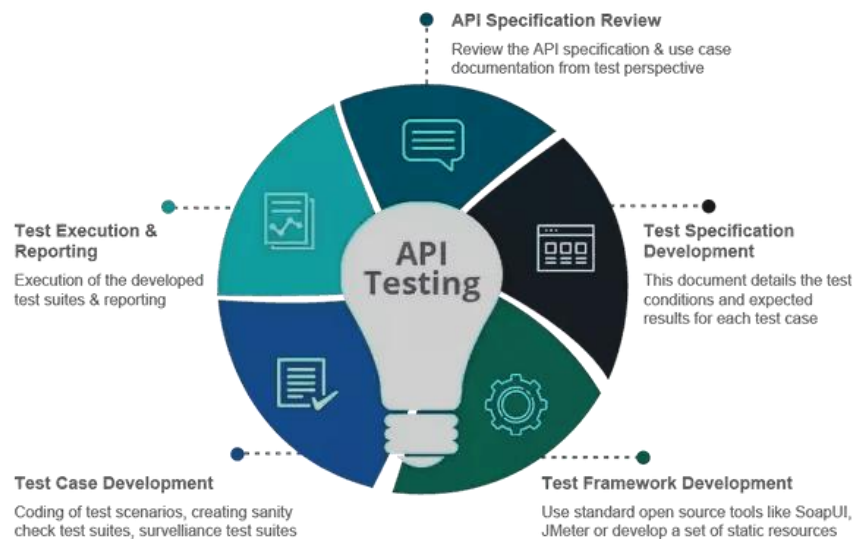


Fig: 5 – Enterprise API Testing Strategy

Publishing and Deploying API

Once the microservice architecture pattern has been applied and the system architected as a set of services. Each service is deployed as a set of service instances for throughput and availability. As APIs are packaged microservices and have many moving parts the following guidelines must be followed.

Deployment strategies

There are several different types of deployment strategies which can be adopted depending on the objective.

Rolling Deployment: The rolling deployment is the standard default deployment to Kubernetes. It works by slowly, one by one, replacing pods of the previous version of the application with pods of the new version without any cluster downtime.

Recreate: In this type of very simple deployment, all the old pods are killed all at once and get replaced all at once with the new ones.

Blue/Green (or Red/Black) deployments: In a blue/green deployment strategy (sometimes referred to as red/black) the old version of the application (green) and the new version (blue) get deployed at the same time. When both are deployed, users only have access to the green, whereas the blue is available to the QA team for test automation on a separate service or via direct port-forwarding.

Canary: Canary deployments are a bit like blue/green deployments but are more controlled and use a more "progressive delivery" phased-in approach. There are several strategies that fall under the umbrella of canary including dark launches, or A/B testing.

API Security

Following are the key considerations to ensure end to end API security.

- Design for secure exposure of private and public APIs.
- Out of the policies in edge to improve API security.
- Threat protection policies and token management.
- RBAC for deployment environments & teams.
- Secure logging and forwarding events.
- Breach detection & Mitigation.

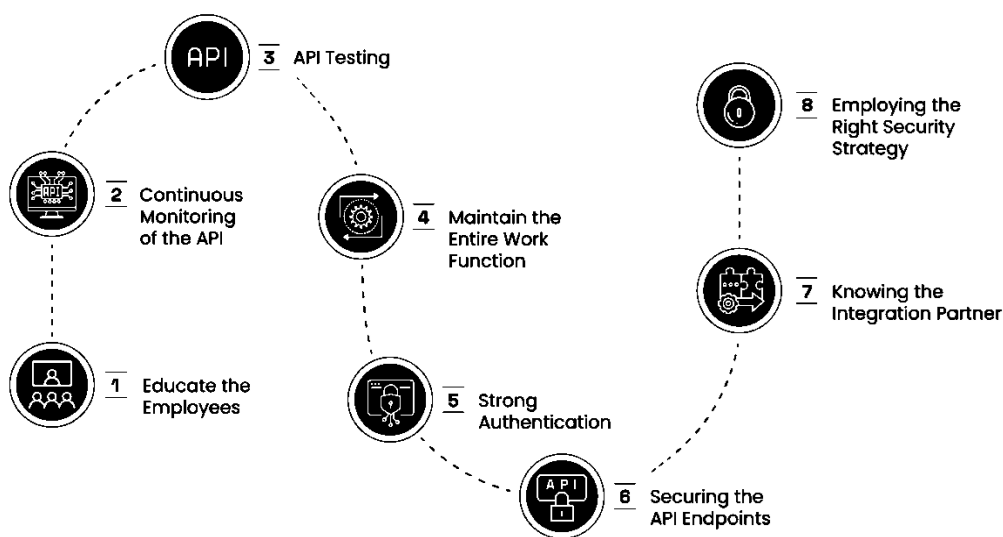


Fig: 6 – Enterprise API Security Strategy

Our Approach for API Enablement & Lifecycle Management

API enablement is a very key aspect of overall IT modernization. Applications those will be rearchitected or rebuild following Microservice approach are expected to expose their required business functionalities as a service. However, a larger no of applications that may be just lifted and shifted to cloud platform needs to be API enabled as well, so that the required functionalities can be exposed across the ecosystem and to external service consumers.

Same is applicable for the larger or extremely complex legacy applications which might needs to be retained as it is. For all these cases an API exposure layer might be added on top of those applications to make them interoperable with the rest of the modernized IT stack.

All these considerations make API Lifecycle Management an extremely important component and an integral part of Microservice based cloud native landscape.

NuboNS API Lifecycle Management Approach

Strategy Evolution

- Assess the Enterprise Landscape
- Define comprehensive API Lifecycle Management Roadmap
- Identify monetization approach and exposure strategy
- Identify rationalization possibilities with appropriate service granularity
- Adopt all round security principles and guidelines
- Define API and service governance plan

Design & Build

- Architect API ecosystem following Microservices architecture
- Build APIs following industry standard frameworks and protocols
- Test APIs using automation framework
- Deploy using established DevSecOps pipeline
- Secure APIs using embedded Security frameworks

Manage

- On-board partners securely into the platform
- Monitor platform and API performance against defined KPIs
- Manage the platform through data driven operations
- Enhance platform as part of continuous improvement

Nubo Native Platform, NNP, provides a state-of-the-art Cloud Native Platform to support Cloud Repatriation, rapid Software Development, Application Modernization and Cloud Migration.

NNP Provides -

- Public Cloud like Cloud Services on any Cloud setup with much needed flexibility and customization possibilities, ensuring reduced Total Cost of Ownership
- Software Development Platform to support rapid Cloud Native Development with complete Development Lifecycle Automation, brings in much needed standardization across the architecture, design and development process with common set of tools, processes and frameworks supported by Enterprise Low Code.
- Enabling inclusive API Lifecycle Management with an advanced Solution Marketplace
- Comprehensive platform to capture, store, process and expose large and diverse Enterprise Data and enable AI/ML use cases

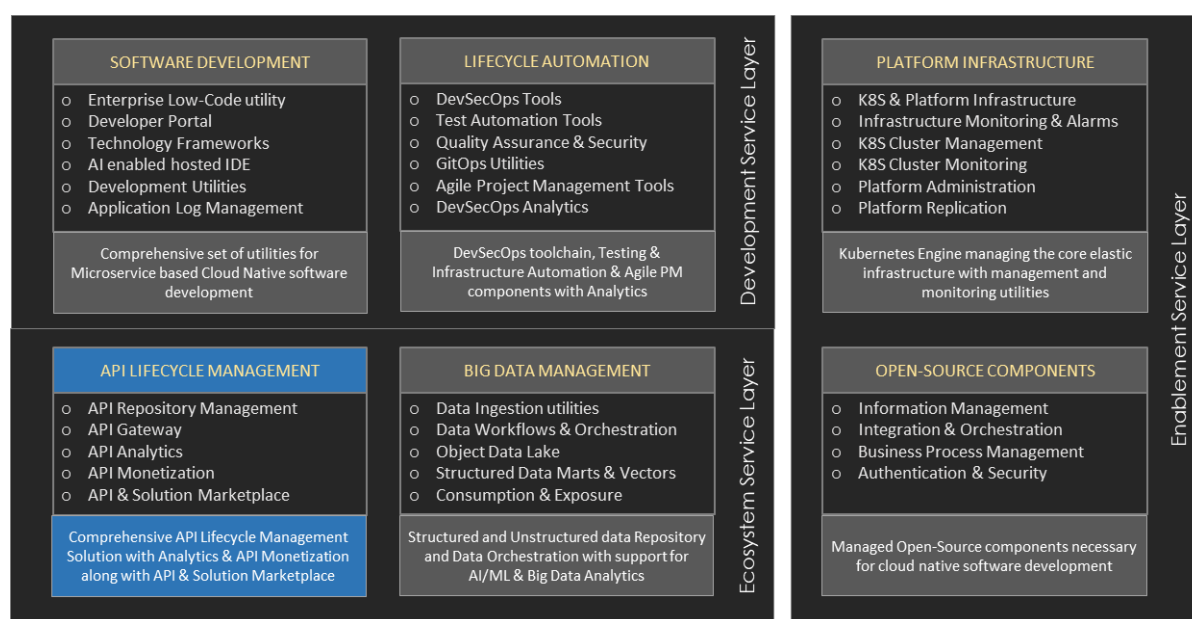


Fig: 7 – NNP with its API LCM Components

NuboNS has a comprehensive solution for API Enablement and to support its Lifecycle as part of its Nubo Native Platform (NNP). It is an all-inclusive implementation of standard application programming interfaces following industry best practices and specifications. It supports comprehensive lifecycle of API Management encompassing API Design, Development, Management & integrated API Analytics along with Monetization capabilities.

This highly secure API ecosystem provides all necessary tools to modernize & monetize your API ecosystem.

Following are the key constituents of this solution

API Design Studio – Low Code development utility to quickly design & develop microservice assembly and expose them as API in the integrated API Gateway.

API Orchestration – Low Code development utility to visually integrate and orchestrate developed APIs through an intuitive drag and drop interface.

API Development Lifecycle Automation – managed through an integrated DevSecOps pipeline, automating complete CI/CD/CT process with 5 step security considerations from pre commit to web vulnerability check.

API Registry – Comprehensive API registry with multi-level API provider and consumer definition with invoker specific API Key & Secret configuration.

API Gateway – Low latency, lightweight and ultra-fast API gateway to securely expose APIs with standard functionalities like authentication, authorization, chasing, throttling and more.

Protocol Support – Gateway supports all standard communication protocols like REST, SOAP, Protocol Buffers, XML-RPC and can be customized to support additional ones.

Service Mesh - Managing rule based secure communication among APIs, with detailed application performance management insight.

API Analytics – API Analytics with real-time Dashboard encompassing business Transaction & Infrastructure utilization perspectives

API Monetization – Monetization capabilities supported by defined API Plans with account level discounts and promotions. Integrated rating and billing modules to create invoices based on subscription and usage.

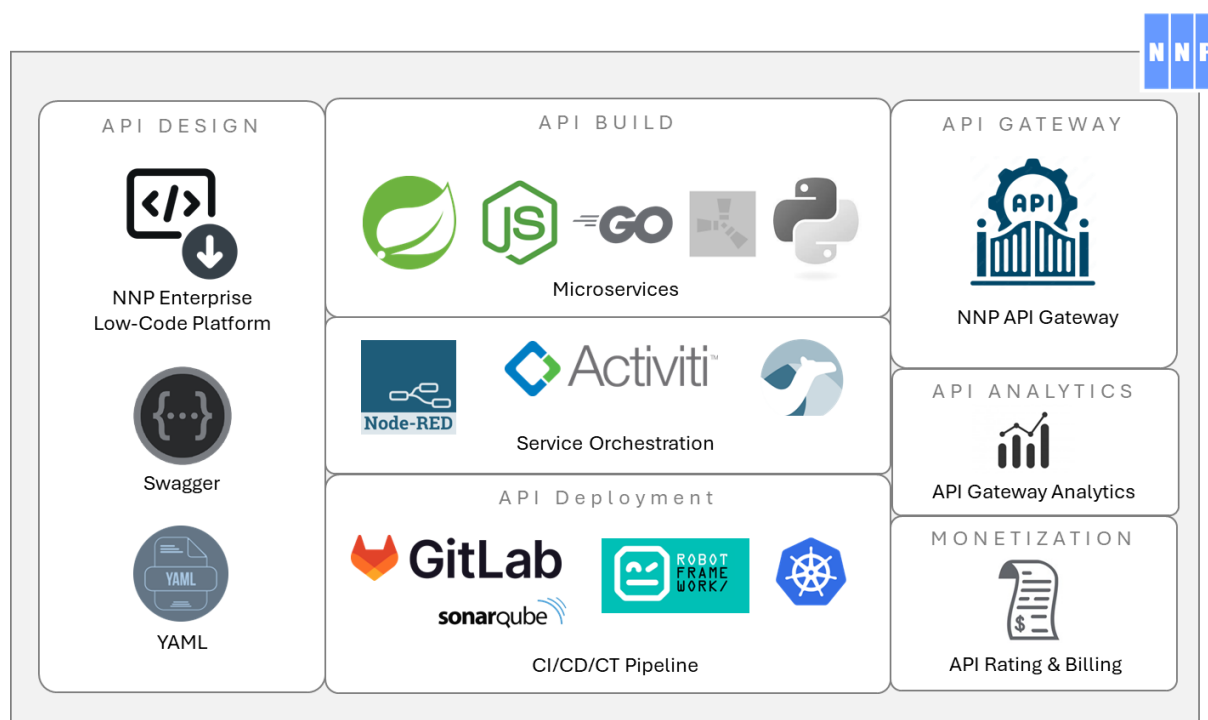


Fig: 8 – NNP API LCM Solution Components

Key features provided by this solution framework:

- Supports comprehensive & rapid API development - NuboNS Low Code utility allows extension and necessary customizations on standard API specifications
- NNP platform supports rapid development of APIs as RESTful Microservices and deploy them as containers
- NNP provides options to deploy generated APIs as Serverless Components to optimize valuable infrastructure usage
- Integrated API Gateway provides all necessary security features including Access Control, Key Authentication etc
- API Gateway provides all necessary mechanism for Transaction Management including Throttling, Rate Limiting and much more
- Integrated API Analytics provides deep insight on Business Transactions along with AI based API Anomaly detection mechanism

Conclusion

API Lifecycle Management is a much-discussed topic and an integral part of Cloud Native Application development and modernization. According to market research Global telecom API market is expected to reach USD 218.84 billion by 2022, growing at a CAGR of 22% between 2017 and 2022 NuboNS's API LCM Framework, build on top of Nubo Native Platform, covers every possible aspect of an enterprise scale API ecosystem and provides necessary insights and tools to monetize the same.

NNP Enterprise Low Code Development Platform along with API LCM unleashes the full potential of the solution. Nubo Native Solution professional service works with the enterprises to define the Cloud Native and API strategy in the best possible way and enables complex API transformation initiatives as a valued partner.

About Nubo Native Solution

Nubo Native Solution is working on a mission to democratize cloud by providing a sovereign, adaptable and comprehensive Cloud Platform referred as Nubo Native Platform (NNP) for state-of-the-art Cloud Native Development and Hosting.

Nubo Native Solution with its path-breaking Cloud Platform and associated Consulting and Professional Services enables large-scale Cloud Repatriation, complex Application Modernization, API Lifecycle Management, AI Enablement, Edge Computing and accelerated Software Development ensuring lower TCO and improved TTM, for the Enterprises worldwide.

Compiled by Nubo Native Platform team

November 2025

Website: nubons.com

Email: contact@nubons.com