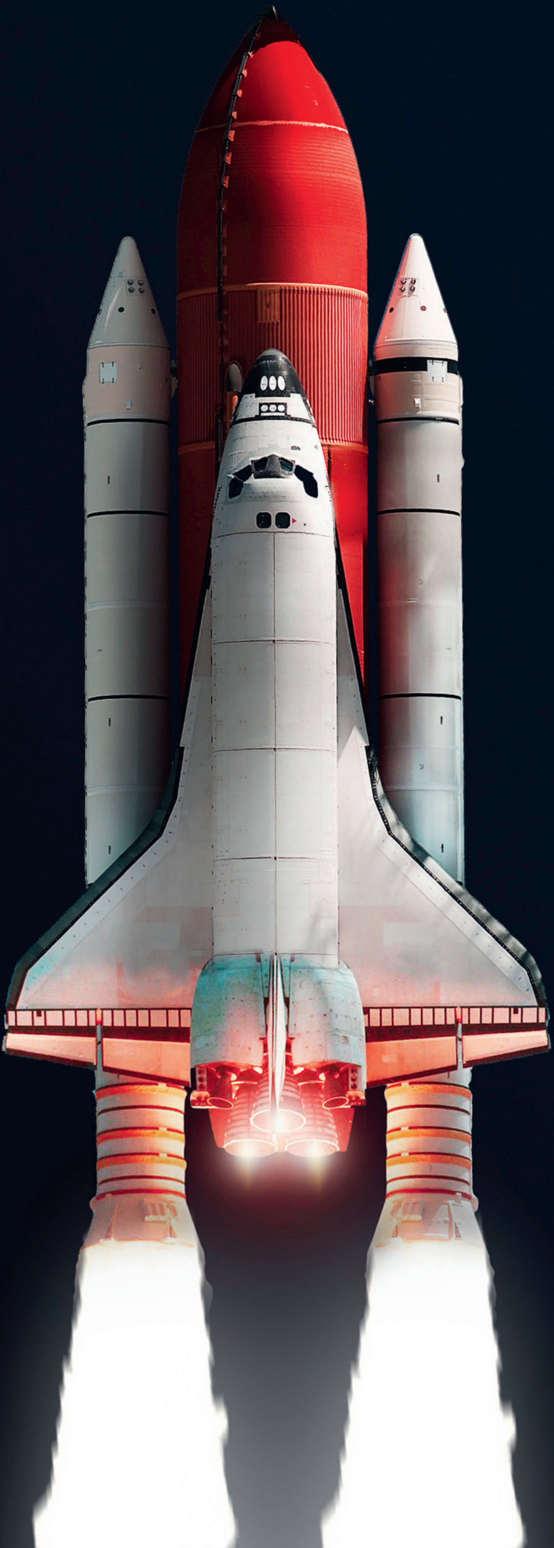
nexoid

System
Architecture



Version
2024.1.0.5

Nexoid Architecture Statement

Architecture, at its core, defines the very backbone of any robust system. Emphasising the importance of good architecture is paramount; systems that lack foresight in terms of upgrades or are anchored on vulnerable platforms can precipitate significant business disruptions. This truth underscores the philosophy behind the creation of Nexoid.

From its user-friendly interface to the intricate workings of its backend, Nexoid has been meticulously crafted from inception to be not only swift and adaptive but also to provide an impregnable secure solution to modern challenges. Fundamental to its operation is the utilisation of AWS Amazon's services – a powerhouse in the world of web hosting. As a testament to its scale and reach, AWS boasts an impressive infrastructure encompassing over 100 data centres across the globe. This spans 31 AWS Regions, 99 Availability Zones, and in excess of 400 edge locations/local zones. Moreover, its presence is felt in an astounding 245 countries and territories.

This strategic alignment with AWS ensures that, when appropriately configured, Nexoid always serves data from a location proximate to its users. Such an approach inherently reduces latency, ensuring an optimal user experience and bolstering performance. As we delve deeper into the architecture of Nexoid, this commitment to excellence and adaptability will become even more evident.

Nexoid Architecture Overview

At the heart of Nexoid lies a triad of sophisticated software solutions, each distinct yet harmoniously interwoven to provide an integrated and holistic user experience.

Nexoid API: The inaugural element of this trio is the Nexoid API, the linchpin that effectively acts as the gateway to the Nexoid Cloud. Geographically, the Nexoid API is region-locked, ensuring compliance with jurisdictional requirements. Structurally, it harnesses the immense scalability and flexibility of AWS, functioning as a serverless application. The flow commences with AWS CloudFront, which seamlessly routes through the AWS API Gateway, relaying commands to AWS Lambda. This, in turn, liaises with storage solutions like S3 and search engines such as OpenSearch. The prime directive of Nexoid API is clear: it is the singular conduit for transmitting data into and out of the Nexoid ecosystem. This crucial responsibility comes with an array of functionalities, from core tasks such as authentication, CRUD operations, and search, to auxiliary features like generating PDFs, dispatching text messages, and interfacing with platforms like OpenAI and Azure.

Nexoid Worker: Stepping into the arena next is the Nexoid Worker, the silent yet relentless automaton that diligently executes tasks behind the scenes. Its purpose revolves around automation; be it triggering an email upon a new record's inception or spawning a record when an email is detected. Notably, the Nexoid Worker operates in a restricted environment. It possesses no innate capability to directly interact with backend functionalities. Instead, it leans wholly on Nexoid API for all operations, from elementary tasks such as reading or updating records, to the more intricate ones like PDF creation or communication with platforms like Azure and OpenAI. Its architectural design is streamlined, leveraging the serverless nature of AWS, initiating from CloudFront, proceeding to the API Gateway, and culminating in AWS Lambda.

Nexoid APP: Concluding the trio is the Nexoid APP, the visual maestro that orchestrates the user interface. Crafted using React.js, the Nexoid APP stands as a contemporary HTML 5, JavaScript-driven application. It embodies the ethos of the modern digital era, being meticulously designed with a 'mobile-first' philosophy. This ensures that users, regardless of their device preference, are greeted with an interface that is both intuitive and fluid.

Nexoid API

The Nexoid API stands as the singular portal to your data, ensuring consistent access and streamlined interactions. This exclusivity in data access does not denote any limitations. In fact, it signifies a unified approach, as the same API entrance caters to both the Nexoid APP and Nexoid Worker solutions. This homogenised access strategy ensures consistent performance and minimises potential discrepancies that might arise from using multiple entry points.

Delving deeper into its architecture, the Nexoid API embodies the principles of serverless computing. The essence of serverless lies in its dynamic adaptability; it can seamlessly expand or contract based on the instantaneous demands of usage. Such an approach ensures optimal resource allocation, allowing for swift scalability in peak usage times and conserving resources during lulls. This elasticity ensures that the API is always responsive, delivering consistent performance without being overwhelmed or underutilised.

Adherence to the RestfulAPI standard underpins the Nexoid API's functionality. This globally recognised standard ensures that data transactions, whether it's an input or an extraction, adhere to well-defined conventions, making interactions intuitive and consistent. The adoption of this standard translates to a user experience that is both straightforward and reliable, reducing the learning curve for new users and ensuring smooth operations for seasoned ones.

However, what truly sets the Nexoid API apart is its integral role in security. While it serves as the gateway for data access, it simultaneously acts as the sentinel, safeguarding data from potential threats. Every interaction, every request, is meticulously scrutinised, ensuring that only legitimate and authorised transactions occur.

For those keen on understanding the intricacies of the Nexoid API's capabilities, a comprehensive documentation awaits. This resource, available at https://developer.nexoid.com/API, offers a deep dive into the API's functions, accompanied by practical examples. It serves as an invaluable guide for both novice users aiming to familiarise themselves with the platform and experts looking to harness the API's full potential.

Nexoid Worker

At its core, the Nexoid Worker operates using customisable Node.js scripts. These scripts are purpose-built to cater to a vast array of data operations, making the system remarkably versatile. One of the outstanding features of using Node.js is the ease with which these scripts can be read and modified. Moreover, there's no need to wade through the intricacies of compiling code or incurring additional expenses from software licences.

A notable strength of the Nexoid Worker lies in its event-driven architecture. Events within the system can be activated in various ways, providing flexibility in automation. For instance, webhooks are readily used to trigger specific actions—be it the creation, updating, or deletion of a record. Additionally, there's the provision to schedule tasks, either at designated times or at regular intervals. Furthermore, there's the option to initiate processes via inbound API calls, making it easier to respond in real-time.

One of the unique facets of the Nexoid Worker is its integration with the Nexoid Library (NL). This library is a repository of functions that maintain uniformity across the various interfaces within the Nexoid ecosystem, such as the Nexoid Worker and Nexoid APP. Ensuring consistency, the NL acts as the backbone, bridging the gap between different components and ensuring they operate harmoniously.

For those keen on staying updated with the latest enhancements and functions added to the Nexoid Library, the developers maintain a comprehensive documentation portal. To delve deeper and access up-to-date information, one can visit https://developer.nexoid.com/Workflow. This platform is continuously updated, reflecting the company's commitment to innovation and customer satisfaction.

Nexoid APP

The Nexoid APP, also referred to as the Nexoid application, represents the forefront of the brand's digital presence. It's the interactive web application tailored to engage users in an intuitive and efficient manner. Crafted meticulously using ReactJS, a renowned JavaScript library, the Nexoid APP boasts of seamless navigation and lightning-fast response times. The decision to harness the power of ReactJS not only ensures fluid user interactions but also offers a robust foundation for any potential future enhancements.

Flexibility is at the heart of the Nexoid APP. Users are presented with an expansive realm of customisation possibilities. Whether it's about swiftly creating new data objects or intricately designing modules, the platform is geared to accommodate. Additionally, there's the added convenience of an integrated store, offering a range of pre-developed modules ready for installation. For those with a flair for bespoke solutions, crafting personalised modules is also an option. To dive deeper into the myriad of functionalities and understand the full spectrum of components available, one can visit the comprehensive guide at https://developer.nexoid.com/Interface. The platform serves as a treasure trove of information, guiding users on how to unlock the application's full potential.

Modules

Modules play a pivotal role in shaping the user interface and determining its functionality. They define the data objects present and delineate the fields associated with these objects, along with the requisite validation criteria. Such design ensures that the data integrity and user experience are maintained at optimal levels.

All modules crafted by Nexoid are readily accessible to the public, ensuring transparency and ease of adoption. Users can seamlessly install these via the settings in the modules section. A standout feature is the capability to bundle these modules with workflows. This pairing amplifies the system's capabilities, offering users enhanced flexibility and functionality in their operations.

Overriding Modules

Modules within Nexoid adhere to a structured hierarchy, reminiscent of the polymorphism concept found in object-oriented programming languages. At the zenith of this structure lie the subscription modules; these are modules specifically crafted and diligently maintained by Nexoid themselves. These provide a foundation upon which other layers interact. Subsequent to this is the customisation level, a pivotal layer that operates at the organisation scale. Here, users have the liberty to either innovate by crafting novel objects and functions, or they might opt to modify or even discard existing structures. It's crucial to note that any adjustments made at this tier will resonate across the entirety of the user base, ensuring uniformity.

Diving deeper, the subsequent layer in this hierarchy is the group level. This level introduces a more granular approach to customisation, allowing interface modifications to be tailored for specific groups. This ensures that different teams or departments can have interfaces moulded to their unique needs. Lastly, at the most individualistic scale, is the per-user level. This tier is especially invaluable for developers and testers, granting them the capability to tinker with and evaluate modules in a sandboxed environment, ensuring that innovations are both robust and effective before wider deployment.

Serverless Overview

In the evolving landscape of digital architecture, the concept of serverless computing has emerged as a frontrunner, offering distinct advantages over traditional hosting options such as LAMP (Linux, Apache, MySQL, and PHP) stacks. One of the paramount benefits of a serverless approach lies in its ability to sidestep the common bottlenecks associated with traditional hosting. Whereas traditional systems often grapple with the constraints of fixed resources, serverless computing is inherently dynamic. It scales on-demand, responding in real-time to fluctuating traffic loads, thereby ensuring optimal performance regardless of the volume of user requests. This elasticity not only prevents system overloads but also translates to cost-efficiency, as resources are consumed strictly based on demand.

Another cornerstone of serverless computing is its innate capacity for streamlined deployments. The serverless development framework significantly simplifies the deployment pipeline. Rather than being encumbered by the multifaceted intricacies of cloud engineering, developers can deploy code and configurations directly to the cloud with ease. This paradigm shift effectively reduces the operational overhead, eliminating the pressing need for dedicated cloud engineers, and converging the deployment process into a singular, cohesive action.

Furthermore, this streamlined deployment process dovetails with the agile methodology of software development. With serverless architecture in place, deployments can be executed multiple times within a standard workday without causing service outages. This agility fosters a development environment conducive to micro-releases, allowing for iterative enhancements and fostering rapid application development. In essence, serverless computing paves the way for businesses to be more adaptive and responsive to evolving user needs, solidifying its position as a pivotal component in modern digital infrastructures.

AWS CloudFront

In the expansive cloud ecosystem provided by Amazon, AWS CloudFront emerges as a stalwart component, meticulously designed for both content acceleration and robust security. At its core, CloudFront operates as a content delivery network (CDN). Yet, to simply categorise it under this label would be to undersell its vast capabilities and integral role in fortifying and streamlining digital service delivery.

One of CloudFront's most vital offerings is its capacity to veil the details of backend servers. This strategic obscurity serves as a protective layer against potential cyber threats. By masking server IPs from the prying eyes of the external world, CloudFront ensures that malicious entities are left in the dark, significantly hampered in their efforts to target specific infrastructural elements.

This dynamic adaptability is another hallmark of CloudFront's design. The digital realm is ever-evolving, with backend components like AWS Lambdas and API endpoints frequently undergoing modifications. In this fluid environment, CloudFront stands as a beacon of consistency. It efficiently manages incoming traffic, ensuring that every request or command is appropriately routed and reaches its destined location within the backend system, irrespective of any internal alterations.

Beyond traffic management, CloudFront is also a sentinel against the looming threat of Distributed Denial of Service (DoS) attacks. Its vast network of strategically positioned edge locations around the world acts as a shield, designed to absorb and neutralise malicious traffic. By preventing backend systems from being swamped by these nefarious onslaughts, CloudFront offers an invaluable layer of protection against one of the most prevalent cyber threats.

Additionally, the role of CloudFront extends to acting as a digital bulwark. By interposing itself between users and backend systems, it not only enhances content delivery but also conceals potential system vulnerabilities. This means that even if there were exploitable gaps in a backend server, they remain shrouded, safe from external discovery and subsequent exploitation.

In essence, AWS CloudFront is a synthesis of speed and security. Its dual mandate ensures that modern digital platforms deliver content swiftly while maintaining an unwavering guard against the multifaceted threats of the online domain.

Nexoid Security

Nexoid's authentication framework is meticulously crafted to ensure security while offering a seamless user experience. Upon signing into Nexoid, users are provided with a distinct session_id. This unique identifier serves multiple purposes: it recognises the specific user, associates them with a particular machine_id, and demarcates their permissions. Such a multi-faceted approach guarantees the integrity of each user's session while maintaining a robust security barrier.

Session longevity is flexible in Nexoid. By default, sessions are transient and are programmed to expire. This expiry paradigm can be moulded in two ways. Firstly, there's the 'rolling expiry' methodology, where the session dissipates if a user remains inactive over a specified duration. Conversely, 'absolute expiry' sets a definite timeframe, post which the session concludes, irrespective of user activity.

Diving deeper into the login mechanism, Nexoid offers two primary pathways. The first is the traditional Nexoid account, where user credentials are under the platform's purview. To fortify security, user passwords undergo a transformation into a one-way hash using the sha512 algorithm. What sets Nexoid apart is its commitment to individualised security. Each hash is exclusively tailored to a particular user and client. This ensures that even if two users inadvertently choose identical passwords, their corresponding hashes remain distinct, safeguarding their unique identities.

Alternatively, for those inclined towards external account management, Nexoid extends compatibility with prominent platforms such as Microsoft Active Directory and Google accounts. To augment security further, while using these Single Sign-On (SSO) options, Nexoid incorporates dual-factor authentication, providing an added layer of protection against potential breaches.

Permission granularity is a standout feature of Nexoid. Users can be granted permissions at varied levels of specificity. For instance, a user might have a broad permission, like reading access to all patient records. Simultaneously, another might have a more targeted permission, such as access to records of patients affiliated with a specific medical practice.

Lastly, venturing into an arena where few dare, Nexoid pioneers the concept of record-level permissions with the capacity to recognise multiple owners. This not only ensures personalised access but also facilitates collaborative functionalities, making Nexoid a versatile and secure platform for diverse needs.

AWS S3

AWS S3 serves as the dedicated file storage solution within the AWS ecosystem. Nexoid facilitates seamless interactions with this platform, enabling users to upload, retrieve, and delete files with ease. However, it's worth noting that individual files in S3 have a size limit of 4GB. What truly distinguishes S3 is its unparalleled reliability. With a staggering 99.999999999% (often referred to as '11 9s') availability, it's widely regarded as one of the most secure and dependable online repositories for data storage.

AWS OpenSearch

Opensearch, previously known as Elasticsearch, is the engine behind our lightning-fast search and retrieval capabilities. It represents a paradigm shift from conventional relational databases. Instead of relying on the familiar 2D tables associated with traditional databases, Opensearch harnesses the power of document storage. It stores data in JSON format—a modern and flexible data structure reminiscent of XML. JSON files are multi-dimensional, allowing for diverse and intricate data configurations.

A common challenge with document storage systems in the past was the indexing or searching of these documents. Opensearch adeptly addresses this concern. It's capable of securely housing trillions of records within a single table, yet can index and sift through them with unparalleled speed and precision.

Nexoid's innovative implementation of Opensearch is a game-changer in terms of data flexibility. Unlike rigid structures that mandate specific schema setups, Nexoid ensures that Opensearch remains agnostic to the underlying data structure. This means users can harness the full potential of a system that allows for dynamic changes. Whether you wish to add new fields or remove existing ones, Nexoid's system accommodates these alterations on-the-fly, even in a live environment. Such flexibility empowers users, granting them the freedom to adapt and evolve their data models as their needs change, without the overhead of cumbersome migrations or system downtimes. This is not just about convenience—it's about fostering a truly responsive and adaptable data ecosystem.

Furthermore, Opensearch comes equipped with an array of features that many traditional databases lack. One such unique capability is its "fuzzy search" function. This means when conducting searches, Opensearch can recognise and return results even when there are slight discrepancies in the query, such as spelling errors or phonetically similar words. This ensures users get relevant results even when their search terms aren't exact, enhancing user experience and efficiency. In essence, Opensearch not only offers rapid and accurate data retrieval but also introduces an element of intelligence and adaptability to database searching.

Pricing

Given the vast array of customisation and functionalities that the Nexoid platform offers its users, determining a one-size-fits-all pricing model can be intricate. To address this, we've adopted a 'pay-as-you-go' approach. Clients are billed on a per-user-per-month basis, each accompanied by a specific allowance. These allowances are communal, pooled among users. If consumption exceeds the total allowance, additional charges are applied either per MB or per processing token. It's essential to note that pricing may vary by region. For a comprehensive and up-to-date overview of our pricing, please visit: https://www.nexoid.com/pricing

Recognising that this pricing structure might pose complexities in estimation, we've introduced a handy pricing calculator to help potential clients assess their projected costs. You can find this tool at: https://www.nexoid.com/pricing_calculator Our research indicates that our demand-responsive pricing typically undercuts similar competitors by about a third. Nevertheless, we understand that every organisation's financial structure is unique. With this in mind, we're flexible, offering alternative pricing arrangements, such as flat monthly or yearly rates, to some clients. Moreover, for those who require it, we're fully equipped to provide detailed invoicing.