

INDUSTRIAL ASSIGNMENT - 9

EXPLORE AND COMPARE THE FEATURES OF VARIOUS NO SQL DATABASES

Mongo-DB	Couch-DB
▪ Data stores in BSON format	▪ Data stores in JSON format
▪ The database contains collection	▪ The database contains documents
▪ It favors consistency	▪ It favors availability
▪ It written in C++	▪ It written in Eralang
▪ It is strongly consistent	▪ It is eventually consistent
▪ Mongo-DB provides faster read	▪ Mongo-DB id faster then Couch-DB
▪ It uses a TCO IP based interface	▪ It uses a HTTP/REST based interface

Neo4j	Redis
▪ It was developed by Nwo4j Inc	▪ It was developed by Redis labs
▪ It is Written using java language	▪ It is written in C language
▪ Operating systems are Linux, OS X etc	▪ Operating system are BDS, Linux etc
▪ It has foreign key	▪ It does not foreing key
▪ Its primary database model is graph DBMS	▪ Its primary database model is key value store
▪ It does not supports partitioning metods	▪ It does not supports sharing partitioning metods
▪ Nwo4j doesn't support map reduce	▪ Its support map reduce through redis gears

H-base	Cassandra
It is based on masters slave architecture model	It is based on Active-Active node architecture model
It is based on Google big-table	It is based on Amazon Dynamo DB
It does not support ordered partitioning	It allows for ordered partitioning
It provides more consistency	It does not provides as much consistency as H-base provides
It has ability to use a coprocessor	It is not capable to support use coprocessor
H-base cluster setup is not easy	Cluster setup of Cassandra is easier then H-base
It support automatic rebalancing within cluster	It support the features of rebalancing but not of the entire cluster

WHAT IS API :-

An application programming interface (API) gateway is software that takes an application user's request, routes it to one or more backend services, gathers the appropriate data and delivers it to the user in a single, combined package. It also provides analytics, layers of threat protection and other security for the application.

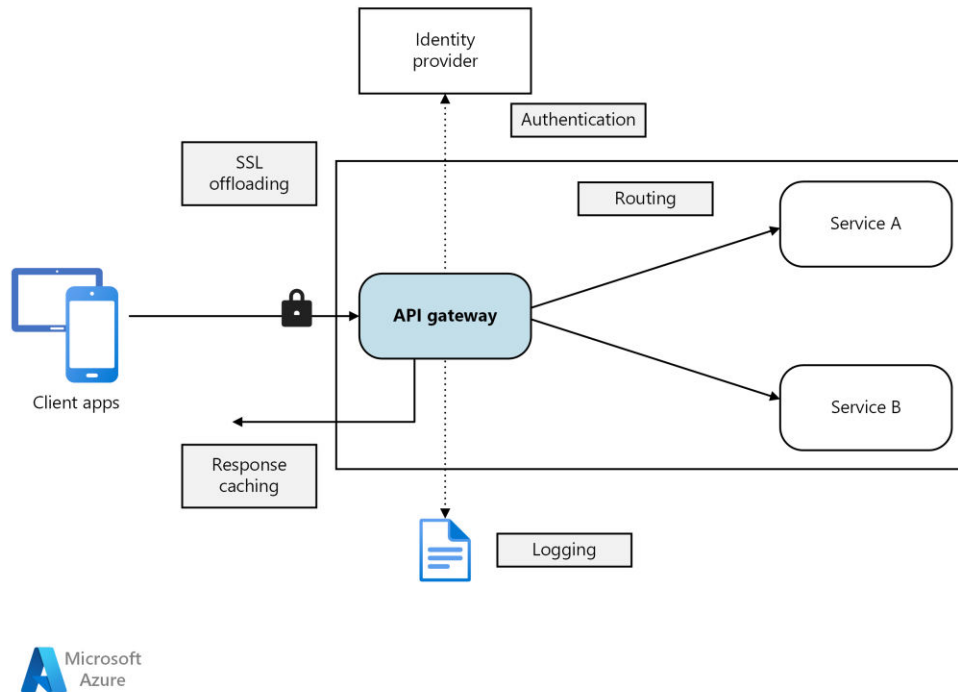
➤ How do API gateways work?

API gateways sit between a user and a collection of microservices, providing three key services:

1.Request routing: An API gateway receives a new API request, turns it into multiple requests, consults a routing map that shows where each request should be sent and sends the requests to the appropriate internal microservice or microservices,

2.API composition: The API gateway provides workflow orchestration as it aggregates the requested information from multiple microservices, bundles the data and returns it to the requestor in composite form.

3.Protocol translation: API gateways know that API requests come in via devices that use different API protocols, and they help client requests and microservices communicate with each other by translating those protocols. The gateway translates API protocols from what the end-user device uses whether a web browser, mobile or another endpoint to the microservice's protocols. A wide area network (WAN) and local area network (LAN), for instance, perform differently and have different API needs. When information comes back, the gateway transforms and sends it back to requestors in the way they can view it. If, for instance, a microservice provides a response in XML, but the request came in using JSON, the gateway automatically does that translation. A REST API uses the HTTP protocol to request API services.



► Types Of API's gateways

Open-source API gateways

An open-source API gateway lets DevOps teams create new API sources without writing code. Some of the benefits of an open-source API gateway include letting a company start small and scale up fast, allowing the flexibility to innovate and change quickly and providing transparency for users.

RESTful APIs

Build RESTful APIs optimized for serverless workloads and HTTP backends using HTTP APIs, HTTP APIs are the best choice for building APIs that only require API proxy functionality. If your APIs require API proxy functionality and API management features in a single solution, API Gateway also offers REST APIS

WEBSOCKET APIS

Build real-time two-way communication applications, such as chat apps and streaming dashboards, with WebSocket APIS API Gateway maintains a persistent connection to handle message transfer between your backend service and your clients

Benefits of API gateways

Adding one or more API gateways to your microservice applications provides many benefits:

1. **Microservices security:** An API gateway puts a barrier in front of an application's backend,

making it more secure. It means an application's endpoints are not exposed; therefore, there's less threat of attack. A company can also use HTTPS for additional security or HTTP encrypted with SSL, which improves performance.

2. **API authentication:** An API gateway provides another security layer that protects against mistakes, hacks and data breaches by authenticating API calls. Authentication and authorization can include antivirus scanning, decryption and encryption, token translation, validation and other security functions.
3. **Input validation:** Input validation ensures an API request has all the necessary information in the correct format before the gateway passes it along to a microservice. If something is missing or wrong, the gateway rejects the request. When it's validated as being correct, the gateway sends the request.
4. **Faster response times:** Because an API gateway sends requests directly to the right services, there are fewer roundtrips and less traffic, reduced latency and better performance overall, which means an application provides an improved user experience.
5. **Microservices load balancing:** An API gateway keeps track of requests sent to different microservices, balances the load between nodes for efficiency and ensures the application remains available. This load balancing is critical when high traffic levels are expected such as during a Black Friday sale or new product launch to prevent spikes or denial of service events.
6. **Rate limiting:** Rate limiting means an API gateway monitors traffic coming in from all sources and limits how many API requests a client (or malicious bot) can make in a specific time period per second or per day, like mobile to protect the system from being flooded with requests and possibly crashing.
7. **Billing for microservices:** Some businesses monetize some of their APIs by offering a service to consumers through APIs. The API gateway handles traffic, monitors usage for specific products or services and sends pricing information to a connected billing system. There are different types of direct monetization, including users paying as they access a service or resource, for a certain number of services or via tiers (where different services are provided at different levels). Other APIs share revenue with consumers through ad revenue share, affiliate marketing or credits to a

consumer's bill.

► Challenges of API gateways

While there are many benefits to adding an API gateway, there can also be challenges:

1. **Response time:** While latency and response time are often decreased due to requests traveling more efficiently, the additional step of a request passing through an API gateway can potentially add to response time.
2. **Dependencies:** Anytime a business adds, changes or removes a microservice, it must update its API gateway. That can be challenging with an application that has evolved from having just a few microservices to encompassing many. However, creating API design rules can help with this.
3. **Complexity:** Routing logic can make communication with microservices more complex. The API gateway is another system that must be developed, deployed and maintained
4. **Security:** Because an API gateway touches many areas of an enterprise's systems, its compromise can seriously impact an application's safety.
5. **Reliability:** If there's only one API gateway and it goes down, the whole application becomes unavailable. Creating multiple API gateways and using load balancers can help avoid this situation