From a Monolithic to Micro-services Architecture

Group Members
- EID, Alaa
- EL MASRY, Bachir
- MOKDAD, Jad
- SHAER, Ibrahim

Supervisors
- Prof. DHAINI, Ahmad
- Prof. JABER, Mohamad
- Ms. ABDALLAH, Rasha
- Mr. KASSAMANY, Yamen

Introduction

Micro-services: The micro-service architecture is based on the concept of Divide and Conquer that involves the decomposition of the monolithic architecture into many discrete independent mini-systems communicating in a well-defined protocol. The basic concept of micro-services is isolation where each service owns its data and resources. The concept of micro-services architecture is similar to SOA but of a more developed version. However, it learned from its mistakes and used modern infrastructures and reactive-principles.

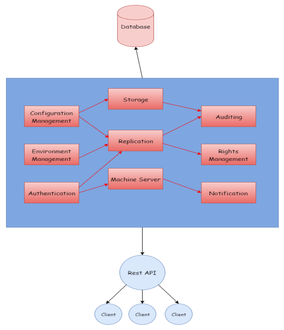
Monolithic: The monolithic architecture is a traditional architecture. All the services that are contacted by the client reside on a single machine. These different services share the same machine capabilities as the CPU transactions ability and memory abilities. This architecture has a lot of drawbacks that branch out from the size of the application. These problems include the difficulty of continuous development, difficulty in vertical scaling and reliability problems.

Objectives

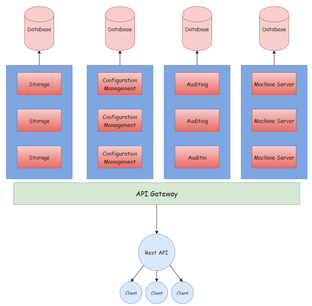
- Present a justification for the different technologies that will be used to develop the project. The justification must include a comparative study of competing technologies.
- Present a working module that mimics the environment management tool processing.
- Design a document to support the design of the project.
- The system that will be build must be resilient to all kinds of failures taking into account the different scenarios that mish occur on this environment management tool.
- The system must include a circuit breaker, load balancing and service discovery service which are integral in the micro-service architecture.

Architecture

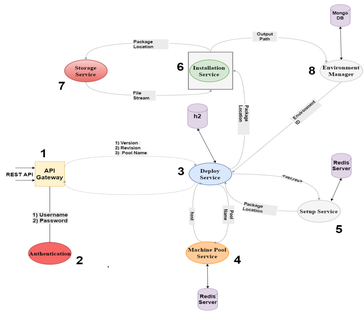
Monolithic Architecture



Micro-Services Architecture



Our Implementation



Technologies and Tools

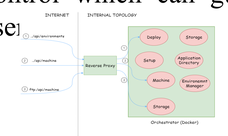
- **OAuth2**: is an industry-standard protocol/framework used to convey authorization decisions across a network. It will match the credentials offered with a local user repository. Also linking each user's authority in order to grant scope of their token.
- **Non-Relational Databases:** (Redis, MongoDB,H2) they have many forms like document-based, graph-based, object-based, key-value store, etc. Here there is no linking between tables, instead nonrelational databases use some other form of mechanism (some are mathematical basis) to storage and retrieve data.
- **Docker:** an open source project to pack, ship and run any application as a lightweight container". Basically, Docker could be considered very similar to a light weight virtual machine in the sense that It doesn't create the whole virtual operating system.
- **Hystrix Dashboard:** reduces the time required for the detection of a component failure [Jacobs, n.d.]. This is because this tool provides real-time insights about the different parameters that govern system behavior like health, traffic, volume…
- **Asynchronous message passing (RabbitMQ):** insures the delivery of the message even if the recipient cannot receive the message at the time of the calling; it resembles the store-and-forward technique. Therefore, in asynchronous message passing, the client can perform and request other tasks while the calling function is in transit.

- **Service Discovery (Netflix Eureka):**is a service registry tool that provides a REST API to manage service instances registration and query for available ones
- **Circuit Breaker:** It's similar to the electrical circuit breaker. The circuit opens whenever the number of failures exceeds a threshold.

FTP Server: File Transfer Protocol (FTP) is a standard Internet protocol for transmitting files between computers on the Internet over TCP/IP connections.

Synchronous message passing(REST): blocks the client until the requested function is completed. It resembles the reply-based communication that takes place in client-server system. REST is an architectural style that determines how the network resources and are defined, sent and received and it's best used in the development of Web Services

Gateway/Reverse Proxy (Zuul): is a type of proxy server that receives clients' requests and retrieves resources from servers located on the internal network on behalf of the requesting client. The requested resources may not be directly accessible from the Internet (i.e.: behind a firewall). A reverse proxy provides additional levels of abstraction and control which can guarantee smoother messages exchange between clients and se



Closing Remarks

- The job experience at Murex was very enriching on personal and technical level.
- Group members have learned a lot of technologies, the use and the theory behind them.
- On the personal level, it gave the group members an insight about how the things work in industry.
- The experience at Murex made the group members used to agile methodology in software development.
- SCRUM meeting, demo, retrospective and incremental development are part of the weekly routine of team members, which are part of the agile methodology.

References

- Apache ActiveMQ TM – Users. (n.d.). Retrieved February 05, 2017, from http://activemq.apache.org/users.html
- Fielding, R. (n.d.). Architectural Styles and the Design of Network-based Software Architectures. Retrieved May 07, 2017, from https://www.ics.uci.edu/ fielding/pubs/dissertation/top.htm.
- Richardshon, C. (n.d.). Pattern. Retrieved May 07, 2017, from http://microservices.io/patterns/reliability/circuit-breaker.html.
- Messaging Systems: An Introduction. (n.d.). Retrieved February 12, 2017, fromhttps://docs.oracle.com/cd/E26576_01/doc.312/e24949/messaging-systemsintroduction.htmGMTOV00115