

# **PRACTICE SCHOOL REPORT**

## **ON**

### **INSIDE SALES**

**RoboMQ Pvt. Ltd.**

**SUBMITTED BY**  
**DEEPAK SHARMA**  
**(2018BTECHEEE002)**

**FACULTY GUIDE**  
**Dr. PUSHPENDRA SINGH**



**Department of Electrical Engineering**  
**Institute of Engineering and Technology (IET)**  
**JK Lakshmipat University Jaipur**

# **INDUSTRIAL TRAINING REPORT**

## **ON**

# **INSIDE SALES**

RoboMQ Pvt. Ltd.

### **Practice School-2**

Submitted in partial fulfilment of the requirements

for the degree of

**Bachelor of Technology in Electrical and Electronics Engineering**

By:

**DEEPAK SHARMA**  
**(2018BTECHEEE002)**

Faculty Guide:

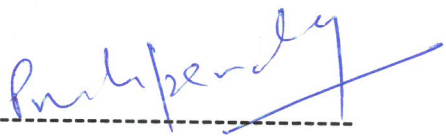
**Dr. Pushpendra Singh**



**Department of Electrical Engineering**  
**Institute of Engineering and Technology (IET)**  
**JK Lakshmipat University Jaipur**

## CERTIFICATE

This is to certify that the Practice School-2 project work entitled "**Inside Sales**" submitted by, **Deepak Sharma (2018btecheee002)** towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electrical and Electronics Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-2 examination.



Dr. Pushendra Singh  
Supervisor and Associate  
Professor

Department of Electrical and  
Electronics Engineering  
Institute of Engineering & Technology  
(IET)  
JK Lakshmipat University Jaipur



Dr. Devika Kataria  
Associate Professor & Coordinator,

Department of Electrical and  
Electronics Engineering  
Institute of Engineering & Technology  
(IET)  
JK Lakshmipat University Jaipur

Date of Submission \_\_\_\_\_

Prof. Dr. – Sanjay Goel (Director , IET)

## ACKNOWLEDGEMENTS

An encouraging and competitive atmosphere is very important for one's growth. Also, proper exposure to an industry of large scale is very helpful to a technical student to learn the practical applications of engineering. I acknowledge that I've been fortunate enough to get both mental and physical support in everything that I did at **RoboMQ Pvt. Ltd.** A very educative and safe atmosphere was provided by the staff. Besides this their help in getting me knowledge of different tools and engaging prospects and generating leads.

I would give special thanks to Mrs. Shreya Shukla (Assistant Manager), under whose guidance I completed my training at **RoboMQ Pvt. Ltd.** I would also like to thank Dr. Pushpendra Singh my faculty guide for giving me valuable input and guiding me from time to time.

I greatly acknowledge the help and the mental strength provided by my entire family for encouraging me and providing me with knowledge & guidance.

Yours sincerely,  
Deepak SHARMA  
(2018BTECHEEE002)



GST No. 06AAICR2941J1Z4  
Mob no. +91-9829013033  
Email: Mohanlat.goel@robomq.io  
mohanlatgoel@gmail.com

RoboMQ Pvt. Ltd.

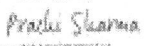
Headquarters  
2500 Greensboro Dr.  
Ste A-10, McLean, VA 22102  
USA

India Center  
307 Sankalp Tower, 8 Queens Rd,  
Jaipur, Rajasthan 302028  
India

To Whom It May Concern

This is to certify that Mr. Deepak Sharma of JK LakshmiPat university, Jaipur is currently pursuing his internship from December 27th 2021 to till date under the supervision of our Assistant Manager, Sales & Marketing on the process of inbound Lead Generation.

This is for information and record please.

Executed by:  
  
Prachi Sharma

Name: Prachi Sharma

Prachi Sharma  
HR Manager and Executive Assistant  
Prachi.sharma@robomq.io  
+91-8000844735

Date: 5/27/2022

## **Acknowledgement**

I would like to express my special thanks of gratitude to my Manager Mrs. Shreya Shukla who gave me the golden opportunity to do this wonderful project on the topic Inside Sales, which also helped me in doing a lot of Research and I came to know about so many new things.

Secondly, I would also like to thank my parents and colleagues who helped me a lot in finalizing this project within the limited time frame.

Deepak Sharma

## Content

| S.NO | Name                                               | Page Number |
|------|----------------------------------------------------|-------------|
| 1    | Chapter 1 (Introduction of the company)            | 7-12        |
| 2    | Chapter 2 (Introduction of the project undertaken) | 13-17       |
| 3    | Chapter 3 (How products works)                     | 18-38       |
| 4    | Conclusion                                         | 39          |
| 5    | Reference                                          | 40          |

## Chapter 1

### INTRODUCTION OF THE COMPANY

#### Company's Vision and Mission

At RoboMQ, we believe that the Internet of Things (IoT) will be created by millions of devices and sensors joining existing Internet of people, processes and systems. We believe in the future infrastructure built with diverse and the best of breed SaaS, PaaS and enterprise applications that will embrace Internet of Things (IoT). This diversity and specializations will create challenges of data silos and integration.

Our technology attempts to removes data silos and help customers realize true networking effect of data by integrating IoT, SaaS and enterprise applications together. In line with this vision, RoboMQ provides a Hybrid Integration Platform (HIP) that can connect any device, sensor, gateway or an application using any standard protocol across any cloud or data center to create context aware business workflows.

#### About the Company

RoboMQ is a SaaS product development company behind new generation of integration middleware and microservices application development platform for the Cloud, Hybrid Cloud, SaaS, Enterprise Applications and IoT.

#### 1.1 One-line about RoboMQ

RoboMQ is an innovative, leading edge integration platform that enables mid-to-large businesses to integrate data and APIs from SaaS, CRM, ERP and operations systems on-premise and on-cloud to create lasting competitive advantage.

## 1.2 Company Fast Facts

- RoboMQ was founded in 2012 and went in active operations in 2013.
- In 2013, started as “Message Queues as a Service” SaaS offering. Acquired WesTower as its first customer.
- In 2014, launched Hybrid Integration Platform (HIP) that is currently used by customers like Topgolf and Kattera.
- In 2015, Topgolf in its rapid growth phase became first Hybrid Integration Platform (HIP) customer.
- In 2019, launched Connect iPaaS, a no-code UX based API and data integration platform targeted at line of business users and citizen integrators.

## 1.3

### About RoboMQ Product

RoboMQ is a disruptive Internet of Things (IoT) and Hybrid Integration Platform (HIP) that connects any device, sensor, SaaS application, or enterprise system on any cloud, protocol or language to create context aware and distributed business workflows.

Its key differentiations are:

1. Any-to-Any Integration
2. Hybrid Messaging Cloud
3. Microservices & Event Driven Architecture

RoboMQ platform itself is containerized and can run on single board computers (SBC), IoT gateways, on cloud or on-premise - creating a federated and distributed guaranteed delivery messaging fabric. IoT devices and SaaS applications can be part of a location transparent, business workflows spanning networks and clouds.

Azure Cloud, customers have the option to use it "as a service" option (SaaS) or run it on their own account on Azure Container service.

## **What does RoboMQ do?**

- RoboMQ accelerates digital transformation and helps businesses create competitive advantage by being able to create workflows using best of the breed SaaS, cloud, on-premise applications and IoT devices.
- Accelerate SaaS and cloud adoption by building effective and strategic business processes.
- Leverage best of the breed application across clouds or on-premise to create compelling business advantages.
- Digital transformation on steroids by assembling functionality from SaaS, Cloud and on-premise applications.

## **1.7 Benefits of using RoboMQ?**

- Speed to market
- Reduction in cost
- Information availability at the right place and right time
- Competitive business advantage

## **Product Offering**

### **1.8**

#### ***1.8.1 Connect - iPaaS by RoboMQ***

Connect iPaaS provides drag and drop integration capabilities for SaaS applications and cloud services including, but not limited to, Microsoft Dynamics 365, ServiceNow, ADP, Office 365, Salesforce and other leading CRM, Finance, Sales and Enterprise Resource Planning (ERP) applications.

### **Hybrid Integration Platform**

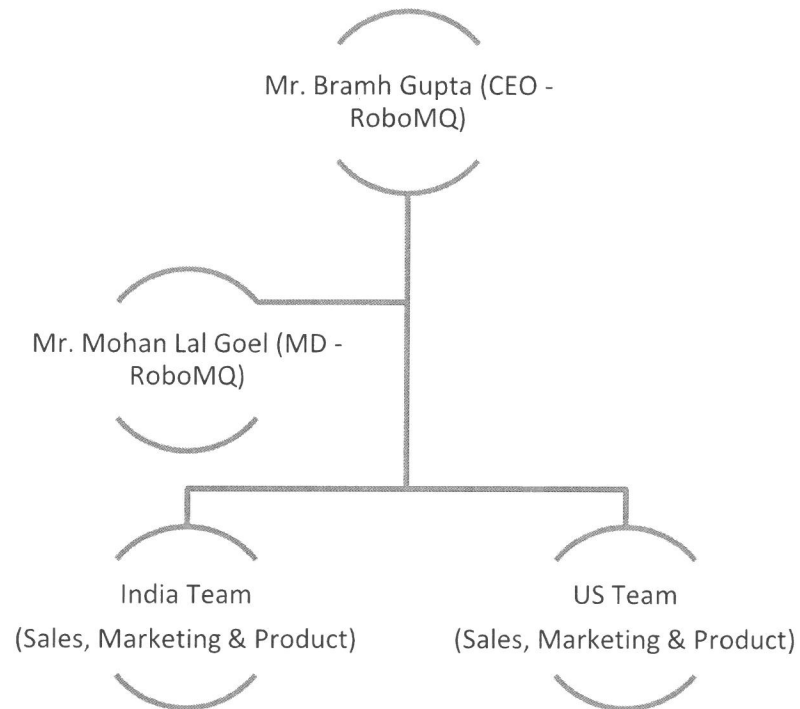
Hybrid Integration Platform (HIP) is a fully containerized microservices based service mesh that allows large scale integration of Cloud, SaaS and on-premise applications in hybrid and federated cloud setup. It allows complex event processing and enterprise application integration supporting complex, custom requirements that create unique competitive advantage for our customers.

Leveraged by large B2B enterprises, Hybrid Integration Platform provides pre-built connectors, business rules, mapping engine, transformation, and guaranteed delivery capabilities to facilitate development of business workflows, API management and API integration.

#### ***1.8.2 IoT Integration***

Create context aware applications by combining near real time device and sensor data with the analytics and application capabilities of Cloud and SaaS applications, allowing strategic decision making and real time tactical response to IoT events. IoT integration provides connectors to leading Internet of Things (IoT) devices, gateways and protocols to ingest and capture real time device data. It provides edge computing capabilities to react to data and events locally as well the capabilities to build complex event processing across enterprise with intelligence at the edge.

## RoboMQ Organization Structure



India Team operates from Jaipur Location and US team operates from Virginia – US.

## Chapter 2

### INTRODUCTION OF THE PROJECT UNDERTAKEN

**Project Name: Inside Sales**

#### **Objectives of the work undertaken**

The main objective of the project is to learn, how to generate leads for the company using the sales technique. The sales technique includes the sales pitch & sales funnel.

#### **Role and Profile**

My role in the company is Sales Development Representative Intern. My work includes researching the target audience and prospect them using the certain sales tools.

#### **Sales Prospecting Stages**

Sales prospecting and nurturing is done by the Sales Development Representative (SDR) at RoboMQ.

The SDR is expected to generate leads and take them to SAL (Sales Accepted Lead) stage where the lead agrees to and attends 30-minute discovery call. The discovery call establishes the mutual identification of the customer need and the fit of the RoboMQ product offerings. A lead will follow the following lifecycle stages:

**New lead** – Lead is generated from buyer intent data and contact databases that fit a specific persona that may be interested in buying a particular integration use cases offered by RoboMQ.

- a. The buyer intent data is generated as a excel file listing companies that are interested in products or offerings that are in the same space as RoboMQ. This intent data is generated by Bombora using the web search performed by the companies. This intent data is at the level of the company.
- b. Based on the buyer intent data, a set of leads or contacts are identified from contact database that fit the target persona of the buyer of RoboMQ products.

**Email Outreach** – New lead is nurtured via automated email sequence.

**Mail Read or Links clicked** – The lead interacts with nurturing emails and hits a threshold of number of mails read or links clicked.

**SAL** – The lead responds to and agrees to attend the discovery call to discuss his or her integration needs with the SDR and the account executive.

## **Life Cycle of SDR At RoboMQ**

The goal of the SDR is to follow this process to convert prospects being worked to Sales Qualified Leads (SQL's) by setting up discovery calls for the Account Executives at RoboMQ.

### **Company Prospect Classification**

Companies for the SDR to prospect are classified in three (3) tiers:

#### **Tier 1 – Companies that have visited RoboMQ**

These are considered as the hottest companies to go after as they are actively showing interest by visiting our website because of the following drivers:

- Search Engine Optimization (SEO)
- Search Engine Marketing and Google PPC ads
- Referral Traffic from YouTube, Vimeo or partner marketplace listings
- Social media
- Email marketing

#### **Tier 2 – Companies in our CRM that are surging in Bombora**

Tier 2 companies are considered to be warm as these companies already exist in our CRM (HubSpot) and they are actively surging in Bombora. We may have had some interaction in the past with some contacts within these companies.

#### **Tier 3 – Companies surging in Bombora**

Tier 3 companies are considered the coldest of the three tiers as these are companies that are surging in Bombora on topics but have not been on our website nor do they exist as company

in our CRM. In essence, these are companies in which we have had no interaction with previously that we want to go after.

## **Certificates from HubSpot Academy**

During my sales internship, I need to learn about HubSpot (CRM) so they provided me the platform to learn and received certificates for the same

### **Learning from this course:**

We often mixed sending lots of emails to clients/prospects to get a successful lead, but this does not work until you do personalization and know what they want.

I explored the A/B Testing, Width of emails, Text limit, and a lot more.

### **Learning from this course**

I explored various angles of inbound sales and how things go for successful inbound sales.

1. Right Target and mindset
2. Ways to achieve leads
3. How inbound works and what it means.

- **Learning from this course:**

- I explored:

Flywheel, Job's Theory, Business idea & Buyer's persona.

## Chapter 3

### HOW PRODUCT WORKS

#### API INTEGRATION

##### What is an API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

##### What is an API Integration?

An application programming interface (API) is a messenger that processes requests and ensures seamless functioning of enterprise systems. API enables interaction between data, applications, and devices. It delivers data and facilitates connectivity between devices and programs.

API can also be defined as an online programming interface of the organization. It allows applications to communicate with backend systems.

An application programming interface creates a channel for the company to sell its products and services online. API enables access of services by adding codes to applications. It further enhances connectivity and bolsters functionality.

API Integration ensure a seamless communication between various applications. But, it is so done by exposing limited amount of a program's internal functions. For example, an app like Zomato is enabled by API to show the restaurant locations on Google Maps. So, in other words, API integration eases businesses and benefits consumers.

##### API Integration Process

- **Research and Preparation**

When it comes to web API's it's important to understand that they are not all the same and they certainly haven't all been created equal. Your research should be focused on finding out three things.

- **What type of API are we dealing with?** There are three different types of web API's: SOAP (Simple Object Access Protocol), REST (Representational State Transfer), and RPC (Remote Procedural Call).

- **What data payload are we dealing with?** These API's can usually either send data in XML or JSON or both payloads, but that is not always the case.
- **Do we have the appropriate documentation?** If the applications are popular, the documentation should be robust and come complete with a WSDL (Web Services Definition Language) for SOAP applications or a WADL (Web Application Description Language) for REST applications. Documentation will let developers know how to format their requests and what data is available. Without proper documentation, an API integration is quite difficult.

- **Data Flow Planning**

After you have done the appropriate research and you determined an integration is possible. It is time to plan the data flows. Data flow planning is a critical part of the API integration process. Data flows are the objects or data you wish to transfer from one application to the other. Things to consider:

- **Authentication** – Some API's and most private web API's are protected by some form of security. You simply cannot create a URL and send it to across the web and expect the data to come back unless you have met the required security measures. Popular security protocols include OAuth and SAML. Each of these protocols has their methods of satisfying access that should be accounted for. You should also consider how long you need to keep the authentication instance running. Will these data flows need to open continually or intermittently?
- **Timing/Volume/Speed** – When, how often, and how much data needs to be transferred? An integration will only make sense if the data can get to the other system in a timely manner. The volume of data that needs to be transferred needs to be balanced with a rate of data transfer and the time that the data is needed to be in place.

- **Data Discrepancies** – When transferring data across applications, you'll need to make sure that this data is in a format that is acceptable to the other end-point. This usually involves things like date/time formatting, and field size limitation. Although seemingly trivial, neglecting to translate this data will result in a failed integration. .
- **CRUDS** – The intent of this data should be taken into account as well. CRUD stands for Create, Read, Update, Delete. When sending information to a new system, this system will need to know what you want to do with the information. It can either create a new record, read or select a record, Update a current record, or delete a record.

- **Data Map**

Once you have the flows planned out, it's best to create a master data map that puts all of this planning into a visual reference. This will make your life easier when it comes time for development. This guide is also the best way to make sure all the developers, consultants, and users are on the same page before the development begins and things become harder to change.

The data map will have the starting (Source) application on one side with all the objects you wish to transfer and all of the relevant information about those objects next to it. On the other (Target) side, you will list all the objects the starting objects will become once they are in the target system. Here is an example data map to give you a better idea of what we're referring too. This is a screen shot of an invoice we moved over to NetSuite from another system.

- **Development**

If you have done the first three steps, the development should flow easily. There should be proper planning for each and every step in this phase.

- **Testing**

Testing is a critical step in the API development process. Testing makes sure that your integration will work exactly as intended under the specified conditions of the end user.

### **Advantages of API Integration**

- Automating Tasks

Manual efforts, like clicking between different application, never seem to take up much time, but they add up. Integrations automate these manual tasks to create a smooth, effortless transition between linked applications.

- Remaining Relevant

With every industry evolving at breakneck speed, it's important to remain at the top of your customer's minds. That means integrating with the apps they've come to know and trust. APIs can provide you with essential cross functionality with current and up-and-coming apps that are vital to your customers. The association with popular and trusted apps can also add valuable credibility to your brand.

- Creating new Products

API integrations have the potential to yield entirely new products that can become a core offering. In the digital marketplace, creating new functionality between apps can create services that never existing before.

## **DATABASE**

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex, they are often developed using formal design and modelling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyse the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

Databases integrated in our work (RoboMQ):

- **PostgreSQL:**

While PostgreSQL has been one of the most frequently deployed relational database management systems, over the past few years it has seen a sharp rise in popularity with developers.

There are many factors that contribute to PostgreSQL's popularity, starting with its highly active open source community that, unlike a company-led open source DBMS like MongoDB or MySQL, is not controlled by any single sponsor or company. And, because PostgreSQL is so ubiquitous, there is reduced risk if there is a need to move the database from on premises to cloud or cloud to on premises, or even from one cloud platform to another.

It is also highly flexible, supporting relational and non-relational models, and can be deployed across a wide range of infrastructures from bare metal to VMs, containers, and DBaaS.

One of the biggest attractions of PostgreSQL for developers is that it fits well into continuous development, integration, and deployment processes and lends itself to DevOps and agile practices. Developers can ensure consistent infrastructure configurations across the entire application lifecycle by introducing infrastructure-as-code recipes using Chef, Ansible, Puppet, or any other infrastructure automation platform and PostgreSQL also fully supports the use of containers and microservices that are at the heart of today's modular, app architectures using Kubernetes. Plus, the fact that PostgreSQL is readily available as a cloud service aligns well with the cloud-based nature of DevOps methods.

- **Redis:**

Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster. Redis has three main peculiarities that sets it apart.

- **MySQL:**

MySQL is an open-source relational database management system (RDBMS). MySQL is free and open-source software under the terms of the GNU General Public License and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB. MySQL is written in C and C++. Its SQL parser is written in YACC, but it uses a home-brewed lexical analyser. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, or a proprietary license. Support can be obtained from the official manual.[19] Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of 3rd party organisations exist to provide support and services, including MariaDB and Percona.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".

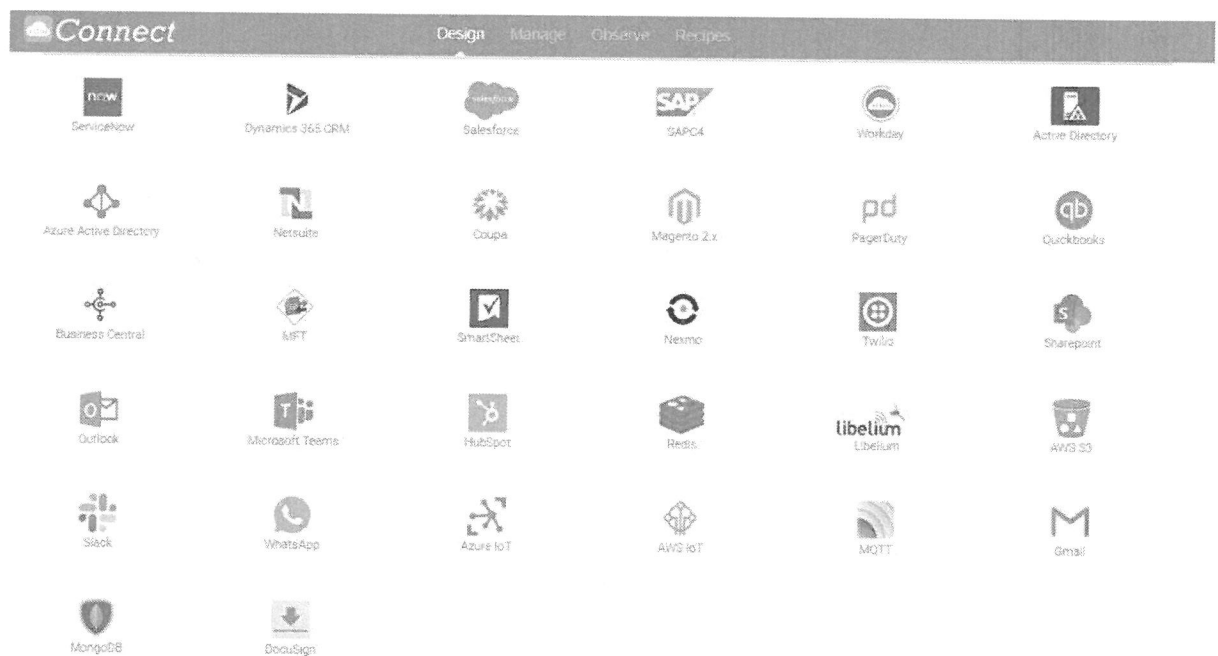
## ECOSYSTEM / APPLICATIONS

In connect app the ecosystems are various types of software and applications that are available in the market such as database software, CRM/ERP software, mailing websites, etc. The connect app connects these ecosystems and can transform data by using different inbuilt excel like functions. The ecosystems are connected in such a way that the data can be exchanged between them.

One can perform certain amount of operations that are present within the ecosystems such as sending an email from Gmail. These operations are divided as triggers and actions based on the type of operation -

1. If an operation is of trigger type that means you can trigger any event in the ecosystem which send data or payload to another ecosystem's action part, so to perform any operation whenever this even is triggered.
2. Actions are the operations which are performed in another ecosystem based on the payload or data coming from the triggering operation.

There are various ecosystems that are available on connect app. Some of them are enlisted below



**Redis** – Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes with radius queries and streams.

**MongoDB** - MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema. MongoDB is developed by MongoDB Inc. and licensed under Server Side Public License.

**ServiceNow** - ServiceNow is a Platform-as-a-service provider, providing technical management support, such as IT service management, to the IT operations of large corporations, including providing help desk functionality. The company's core business revolves around management of "incident, problem, and change" IT operational events.

**Coupa** - Its Cloud Spend Optimization SaaS product manages indirect purchases, invoices and expenses in real-time. Built on Coupa's cloud architecture, it provides executive dashboards, alerts, expense management and real-time benchmarking into a single user interface. This spend optimization software suite focuses on core spend areas including procurement, accounts payable, and expense management.

**Workday** - Workday makes money by selling subscriptions to its services rather than selling the software outright. Expenses are booked up front when it signs on customer. But the associated revenue is recognized over the life of multiyear agreements.

**QuickBooks** - QuickBooks is an accounting software package developed and marketed by Intuit. QuickBooks products are geared mainly toward small and medium-sized businesses and offer on-premises accounting applications as well as cloud-based versions that accept business payments, manage and pay bills, and payroll functions.

### **How Ecosystems are Connected by Connect App**

Selecting Gmail as trigger and Twilio as action –

1. **Gmail Trigger** – Gmail is a free email service developed by Google. Users can access Gmail on the web and using third-party programs that synchronize email content through POP or IMAP protocols. Gmail (pronounced Gee-mail) is a free Web-based e-mail service currently being

tested at Google that provides users with a gigabyte of storage for messages and provides the ability to search for specific messages. The Gmail program also automatically organizes successively related messages into a conversational thread

- Firstly, select Gmail as trigger from the list of ecosystems.
- Now after selecting Gmail as trigger you will be redirected to the list of triggers for Gmail where you can select a trigger by clicking on radio button.
- After selecting the trigger, click on the link account button that will redirect you to connection page where you can choose type of connection such as Basic Auth or OAuth and finally put in the you connection details such as your mail id and password in case of Gmail.
- After filling out your account details you need to click save button and you will be redirected to the Gmail page that will ask your permission to connect to Robomq. Give the permissions and finally you will be redirected to Event filter page.
- In even filter page you can choose from number of filters to transform the data you want to send as trigger. In case of Gmail the filters Subject, cc, From-name etc.
- Finally, after choosing the filters you can click on save button that will make the trigger part complete and now your trigger is activated on Gmail.

**2. Twilio Action** – Twilio is a developer platform for communications. Software teams use Twilio APIs to add capabilities like voice, video, and messaging to their applications. This enables businesses to provide the right communications experience for their customers. Behind Twilio APIs is a Super Network, a software layer that connects and optimizes communications networks around the world. This is what allows your users to reliably call and message anyone anywhere.

With Twilio, you can reach customers in the ways they prefer, and engage with them effectively using context related to that interaction. As customer

experience can increasingly make or break your brand, programmable communications become more crucial than ever to the success of businesses today.

- Now after trigger is activated the data is sent to the RabbitMQ queue.
- Action part will consume this data from the queue and will perform action selected by the user in the Twilio account.
- After completing trigger, you will be redirected to a new screen where you will find the list of ecosystems.
- Select Twilio as action and then you will be redirected to action screen where you will find the list of actions from which you can choose one.
- After this just click on link account and you will be redirected to connection page where you can fill out your Twilio account details and save this connection.
- After saving the connection you will be redirected to transformer page where you can post some data with filters with action request.
- Finally, you can click on save and deploy the flow and you Flow is completed.

This is how ecosystems are connected by connect app. Now, one can see the changes by triggering something in your Gmail account, for example by sending a mail and you can the action chosen by you while making the flow is performed successfully in your Twilio account.

## MICROSERVICES

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services. Microservice architecture enables the rapid, frequent and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack.

The central idea behind microservices is that some types of applications become easier to build and maintain when they are broken down into smaller, composable pieces which work together. Each component is continuously developed and separately maintained, and the application is then simply the sum of its constituent components. In contrast to a traditional, "monolithic" application which is all developed all in one piece.

Applications built as a set of modular components are easier to understand, easier to test, and most importantly easier to maintain over the life of the application. It enables organizations to achieve much higher agility and be able to vastly improve the time it takes to get working improvements to production. This approach has proven to be superior, especially for large enterprise applications which are developed by teams of geographically and culturally diverse developers.

### Advantages of microservices

- **Easier to Build and Maintain Apps:**

The key principle of microservices is simplicity. Applications become easier to build and maintain when they are split into a set of smaller, composable fragments. Managing the code also becomes less painful because each microservice is, in fact, a separate chunk of code. Services can be implemented using different programming languages, databases and software environments. This allows each service to be deployed, rebuilt, re-deployed and managed independently. For example, if a microservice allocates too much memory or puts a heavy load on the processor, it will only affect this service. Any problem with a microservice will not influence the entire system and the failure of individual microservices can be compensated relatively quickly.

- **Organized Around Business Capabilities:**

A microservice architectures invite teams to focus on building business functionality instead of writing glue code. In other words, development teams are organized around

business capabilities and not technologies. This means that services are adaptable for use in multiple contexts. The same service can be reused in more than one business process or over different business channels depending on the need. Each team member is responsible for a service which results in building a smart, cross-functional team.

- **Improved Productivity and Speed:**

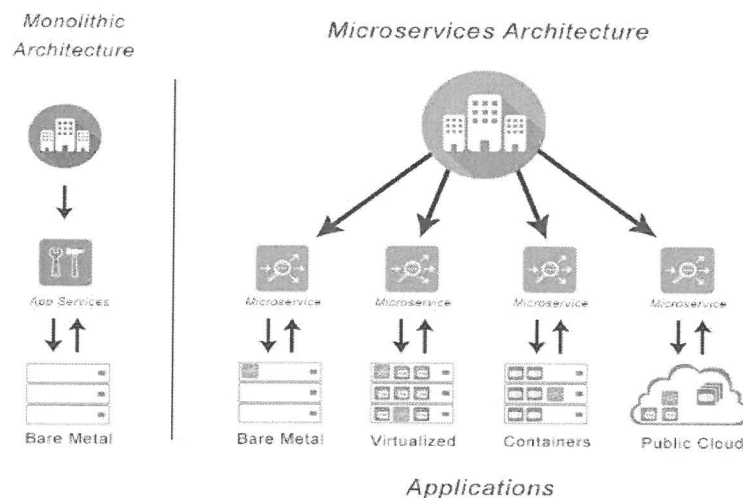
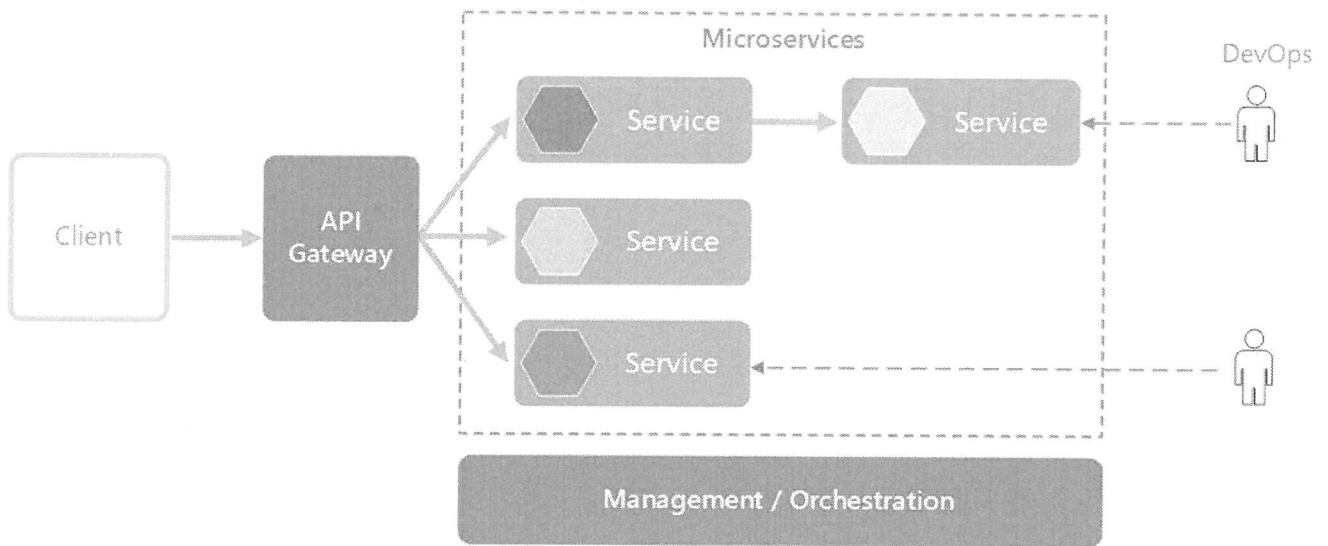
Microservices architecture tackles the problem of productivity and speed by decomposing applications into manageable services that are faster to develop. Different teams can be working on different components simultaneously without having to wait for one team to finish a chunk of work before starting theirs. Also, separate microservices are easier to locate and modify. It is also very handy for speeding up quality assurance since each microservice can be tested individually and you can test the components that have already been developed while the programmers are working on the other ones.

**Flexibility in Using Technologies and Scalability:**

The microservice architecture allows decoupled services written in different programming languages to peacefully coexist with other fragments. This is also good news if you are looking to scale your solution in the future. With microservices, you can add new components to the system painlessly or scale services separately from one another.

- **Autonomous, Cross-functional Teams:**

Microservices are a blessing for distributed teams. Carrying out the development of a massive monolith system can be complicated and messy if you are working with divisions around the globe or extended teams. Microservices grant the developers more independence to work autonomously and make technical decisions quickly in smaller groups. So, if the solution you are developing is expected to be large, be sure to consider the microservice architecture.



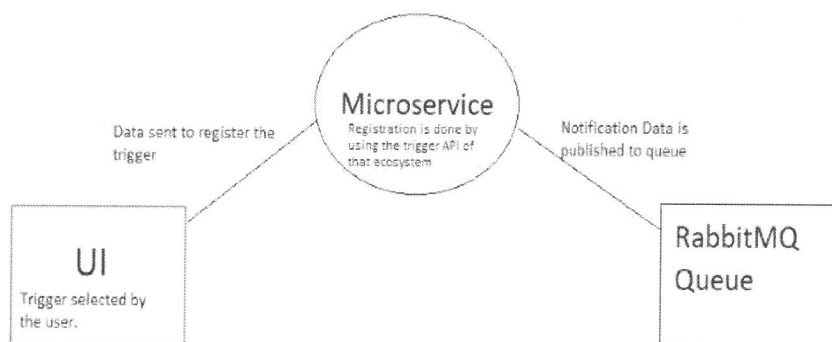
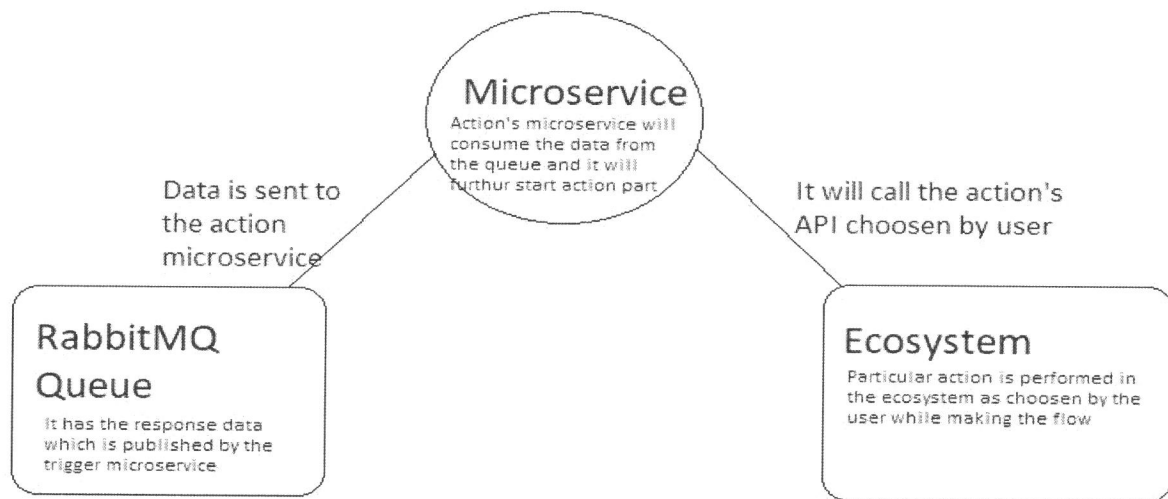
## Microservices in Connect App

In connect app microservices are built in Java. Both the triggers and the actions are handled by the microservices in the backend. Microservices means to divide the tasks into multiple smaller tasks and in connect the responses and requests at the backend is handled by the microservice. To divide the task of handling these requests and responses we divided it into two part one as trigger microservice and another is action microservice.

## Trigger Microservice

Trigger Microservice works in the following steps –

1. When user chooses a trigger from the front end or UI then a request is sent to register that trigger in the ecosystem chosen by the user.
2. The work of sending and handing this request to the ecosystem is done by trigger microservice.
3. Trigger microservice posts the notification or response from the ecosystem to the RabbitMQ queue. This process is called producing or publishing and trigger microservice acts as a producer.



## **Action Microservice**

Action microservice works as follows –

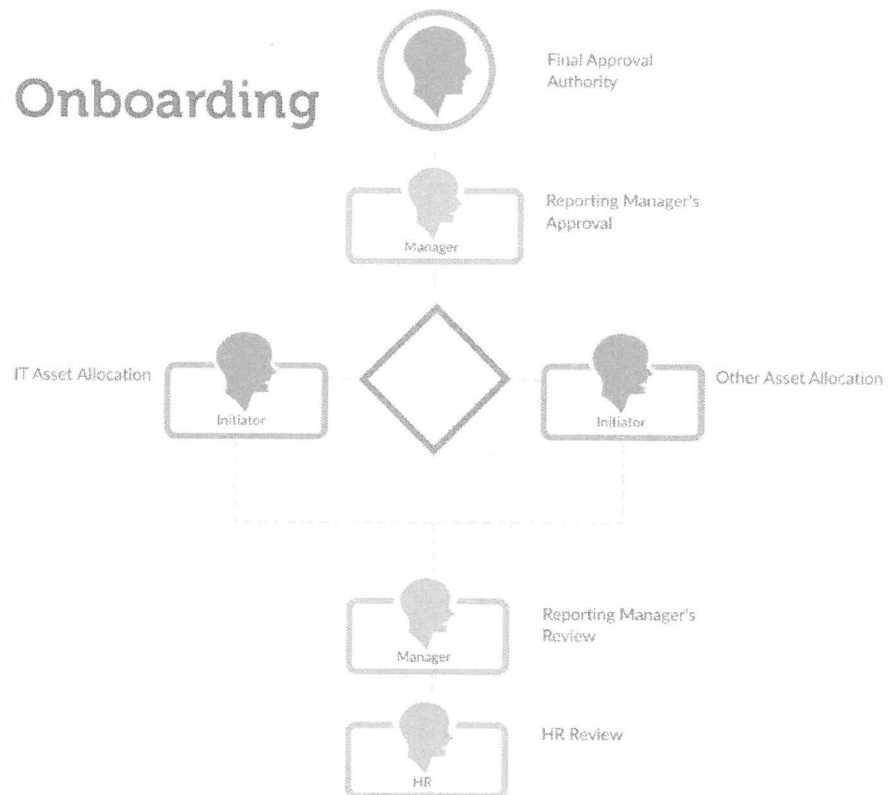
1. Action microservice starts the work by consuming the data from the same queue where data is sent when trigger is registered. Once the user triggers any event the response to this trigger is published to the queue and the action microservice starts consuming that data. That is why action microservice is known as consumer.
2. Now after that, user chooses one action that he wants to perform in another ecosystem. Once the action is selected and flow is deployed the action microservice starts its work.
3. Whenever the action microservice receives data from the queue then it just starts its action work and the action's API is called and that action is performed in the ecosystem chosen by user.

This is how the task of publishing and consuming the data is divided into trigger and action microservices. Each ecosystem present in the connect has their own trigger and action microservices. Hence the task of handling these requests and responses becomes much easier.

## WORKFLOW

A Workflow is a sequence of tasks that processes a set of data. Workflows occur across every kind of business and industry. Anytime data is passed between humans and/or systems, a workflow is created. Workflows are the paths that describe how something goes from being undone to done, or raw to processed.

### Employee Onboarding Workflow



#### Automated vs. Manual Workflows

In a manual workflow, a human is responsible for pushing each item from one task to another. For example, when an employee fills out a reimbursement claim, she must email it to her manager for approval. After approval, she must email it to the finance department.

The finance department must go into the software and schedule a payment and then email the employee to say it is complete.

In an automated workflow, when a human completes a task, she is not responsible for passing the data on to the next task. The workflow is programmed to handle this. The system manages the flow of tasks including notifications, deadlines, and reminders.

In the same reimbursement example, the employee might fill out a form and hit a submit button. It would automatically trigger a notification for the manager to review it and click Approve.

This would automatically take it to the finance team for processing, or if the amount is small enough, it would trigger a task to release the payments and send an automated email to the employee.

Tracking items are much easier in automated workflows. To track items in a manual workflow, you must either manually update a spreadsheet, or send lots of messages and emails to know the status. Automated workflows will show you instantly where the item is in the workflow.

### **Benefits of workflow automation**

- Eliminating redundant tasks
- Improving efficiency
- Simplifying delegation of tasks
- Reducing processing time
- Giving greater visibility
- Establishing accountability

### **Connect platform workflows**

- The connect platform of Robomq provide functionality to create workflows for business processes like HR, Sales, Marketing, finance, and operation teams.
- Robomq Connect provides a huge list of Connectors which leverages the power of API to create customized automated workflows according to the need of the customer.
- Robomq also provide premium recipes which are made to simplify and automate the business challenges like Service order fulfilment , employee access and asset provisioning , sales and marketing which can further help businesses to reduce huge cost of manual processes and increasing reliability and decreasing resource consumption.

RoboMQ Free Trial : <https://trial.robomq.io/connect/dashboard>

RoboMQ Website: <https://www.robomq.io/>

## **What is Software Testing?**

SOFTWARE TESTING is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

### **DIFFERENT TYPES OF SOFTWARE TESTING**

#### **Verification**

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

#### **Validation**

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

#### **Basics of software testing**

There are two basics of software testing: Black box testing and White box testing.

#### **Black box Testing**

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

#### **White box Testing**

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

## Types of testing

There are many types of testing like:

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Sanity Testing
- Smoke Testing
- Load Testing
- Non-Functional Testing
- Soak Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing
- **Unit Testing**

- Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.
- **Integration Testing**
- Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.
- **Functional Testing**
- Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.
- **System Testing**
- System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.
- **Stress Testing**
- Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.
- **Performance Testing**
- Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.
- **Usability Testing**
- Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use,

how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

- **Acceptance Testing**

- Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

- **Regression Testing**

- Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

## **Conclusion**

Inside sales operation is the most critical part of any organization as these people are the first one to approach to any customer.

Inside sales is always aligned with the marketing and the product team in the RoboMQ.

SDR before sending the emails or prospecting they should have full product knowledge.

The work carried out the RoboMQ is completely my original work.

## Reference

1. <http://robomq.io/>
2. <https://www.robomq.io/integration-use-cases/>
3. <https://tray.io/blog/what-is-an-api-integration-for-non-technical-people>
4. <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
5. <https://blog.axway.com/internet-of-things/iot-integration>
6. <https://www.gartner.com/en>
7. <https://www.forrester.com/bold>
8. <https://business.outlookindia.com/magazine/outlook-business>
9. <https://www.theindustryoutlook.com/>