



ENS 492 Graduation Project (Implementation)



Supervisors

Duygu Karaoğlu Altop, Reyhan Yeniterzi

Students

Cavit Çakır, Kaya Kapağan

Problem

Scheduling a meeting is a **boring** task

Usually requires **many emails**

Someone always **disagree** with current schedule time

Requires re-schedule from the scratch



Suggested Solution

An integrated **chatbot** to calendars and workspaces!

Automated tool!

Schedule a group meeting under **a minute!**



■ Market Validation

calendly.com

5 M users

doodle.com

30 M users





Product



Hi Stephanie, How can I help you ?

I need a meeting



When ? With whom ?

Tomorrow, with Joe.



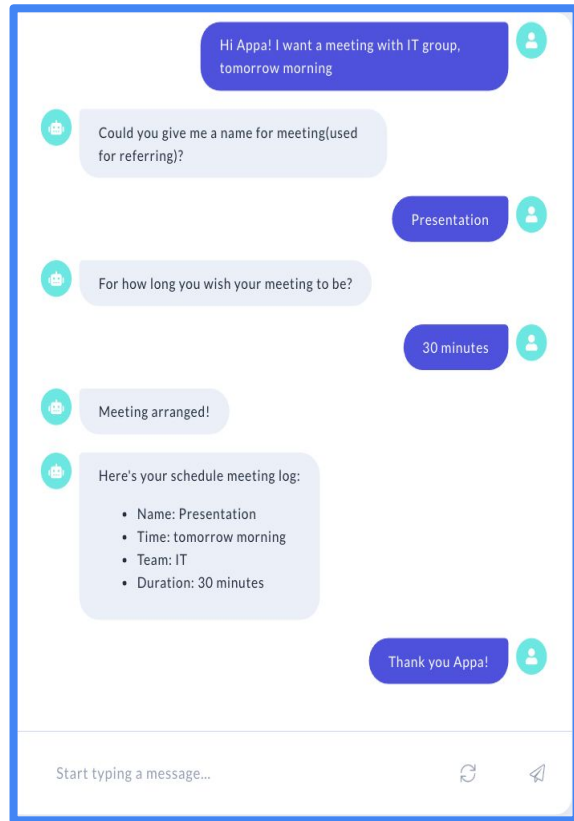
Sync your **Google calendar** to our bot.




Sync the bot to your **workspace**




Schedule a meeting



Chatbot  TARS

Welcome to the TARS scheduler

To book a demo call choose a time that works for you below

24-Jul-2019 10:00 AM 

Calendar Time

9:00 AM	9:30 AM	10:00 AM
10:30 AM	11:00 AM	12:30 PM
1:00 PM	1:30 PM	2:00 PM
2:30 PM	3:00 PM	3:30 PM

Could you give me a name for meeting(used for referring)?

Presentation

For how long you wish your meeting to be?

30 minutes

Meeting arranged!

Here's your schedule meeting log:

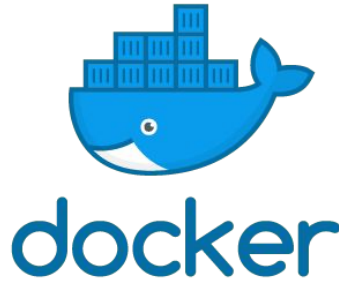
- Name: Presentation
- Time: tomorrow morning
- Team: IT
- Duration: 30 minutes

Thank you Appa!

Start typing a message...

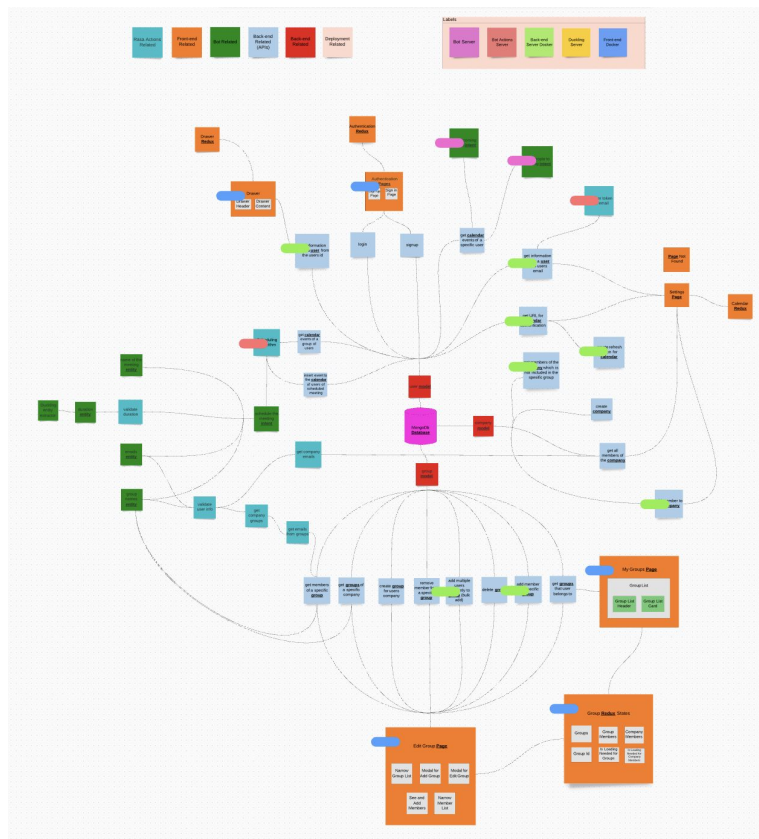
Technical Details

- Chatbot
- Back-end Server
- Website (Front-end)
- Deployment
- Security Analysis



General Structure

- Diagram



Database

```
var companySchema = new mongoose.Schema({  
  name: {  
    type: String,  
    required: true,  
  },  
  createdBy: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: "User",  
    required: true,  
  },  
});
```

```
var groupSchema = new mongoose.Schema({  
  createdBy: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: "User",  
    required: true,  
  },  
  belongCompany: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: "Company",  
    required: false,  
  },  
  name: {  
    type: String,  
    required: true,  
  },  
  members: {  
    type: [  
      {  
        userId: {  
          type: mongoose.Schema.Types.ObjectId,  
          ref: "User",  
        },  
      },  
    ],  
    required: false,  
  },  
});
```

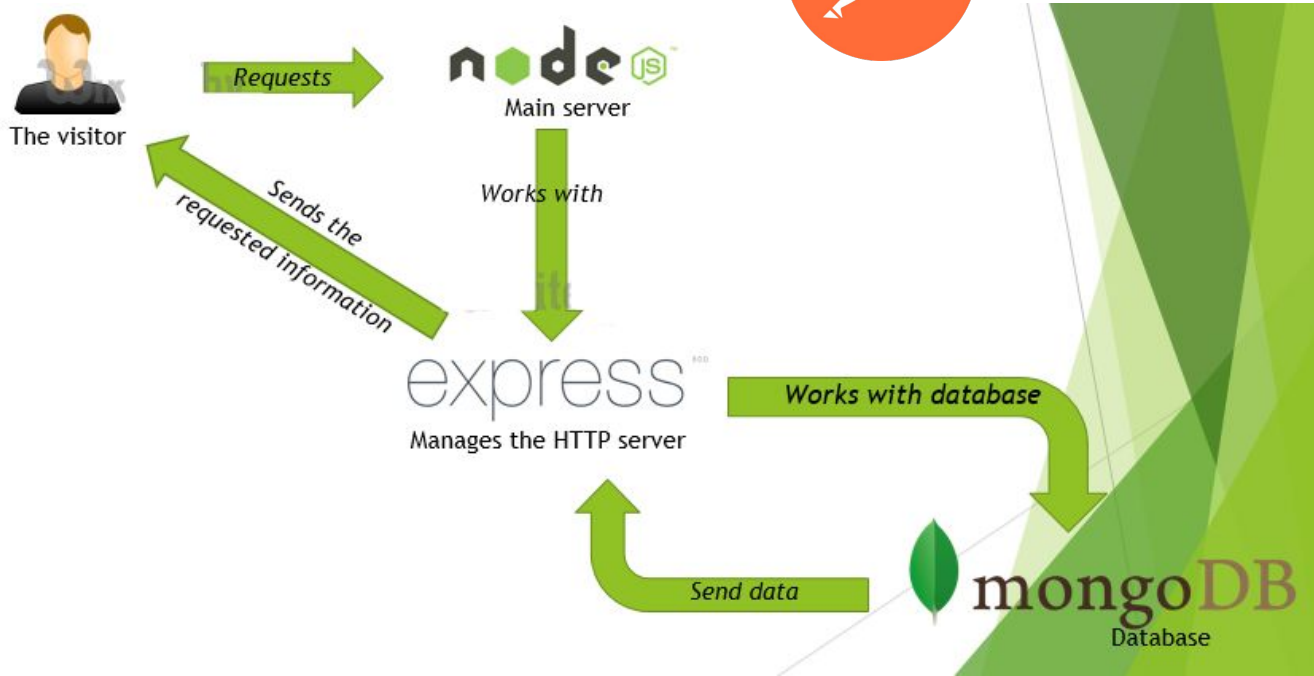
```
const userSchema = new Schema({
  email: {
    type: String,
    index: true,
    unique: true,
    dropDups: true,
    required: true,
  },
  password: {
    //salted and hashed using bcrypt
    type: String,
    required: true,
  },
  userType: {
    type: String,
    required: true,
  },
  belongCompany: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "Company",
    required: true,
    default: "6086aaac5fe0b800647463d1",
  },
  belongGroups: {
    type: [
      {
        groupId: {
          type: mongoose.Schema.Types.ObjectId,
          ref: "Group",
        },
      },
    ],
    required: false,
  },
  googleCalendarRefreshToken: {
    type: String,
    required: false,
  },
});
```

Backend Server

Detailed Documentation of APIs: [Postman](#)

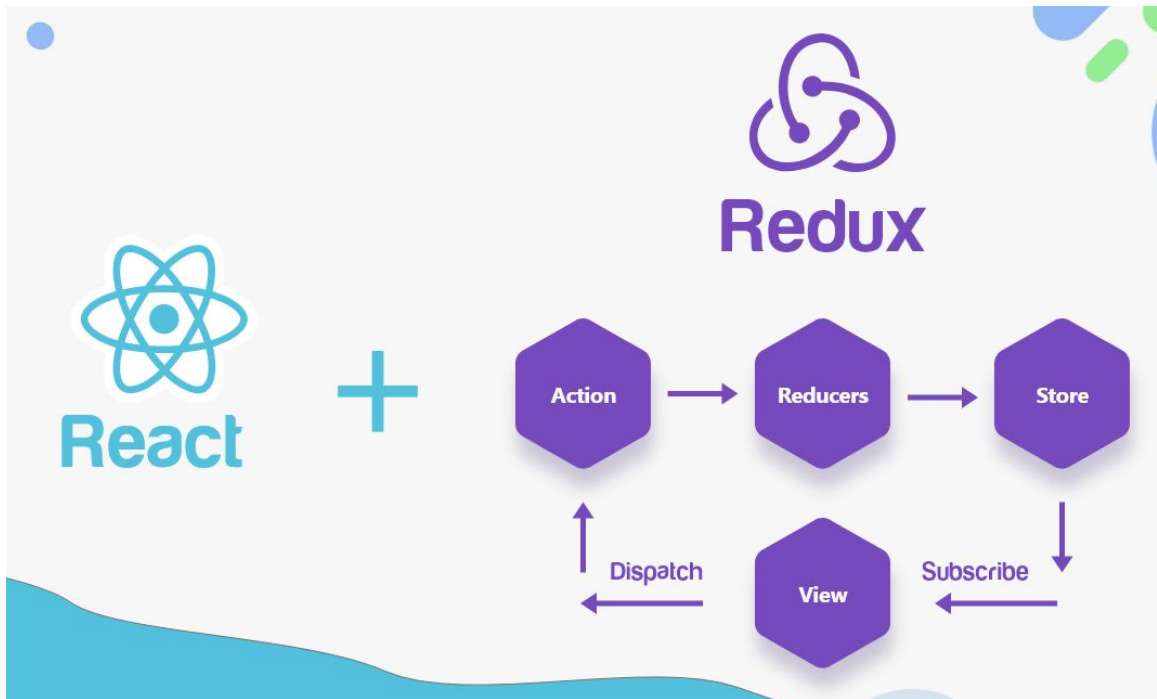


POSTMAN



Website (Front-end)

- My Groups
- Group Settings
- My Settings



Chatbot

Intents:

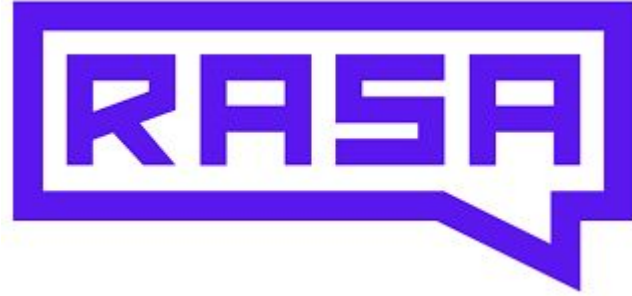
- inform
- schedule_meeting
- greet
- goodbye
- affirm
- deny
- mood_great
- mood_unhappy
- bot_challenge
- thankyou
- out_scope
- show_upcoming_events
- out_of_scope

Entities:

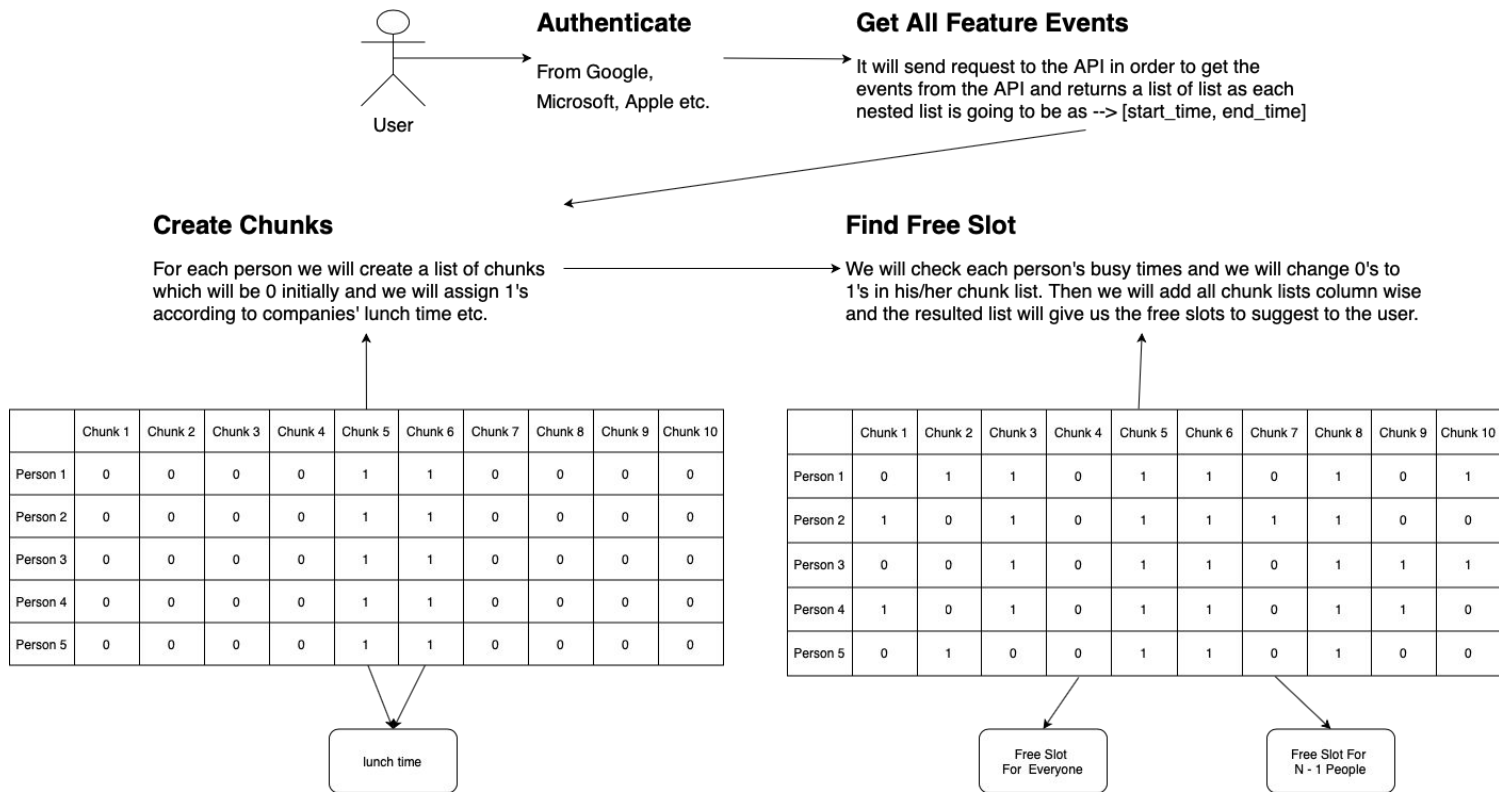
- duration
- group_name
- meeting_name
- user_info
- user_mail

Actions:

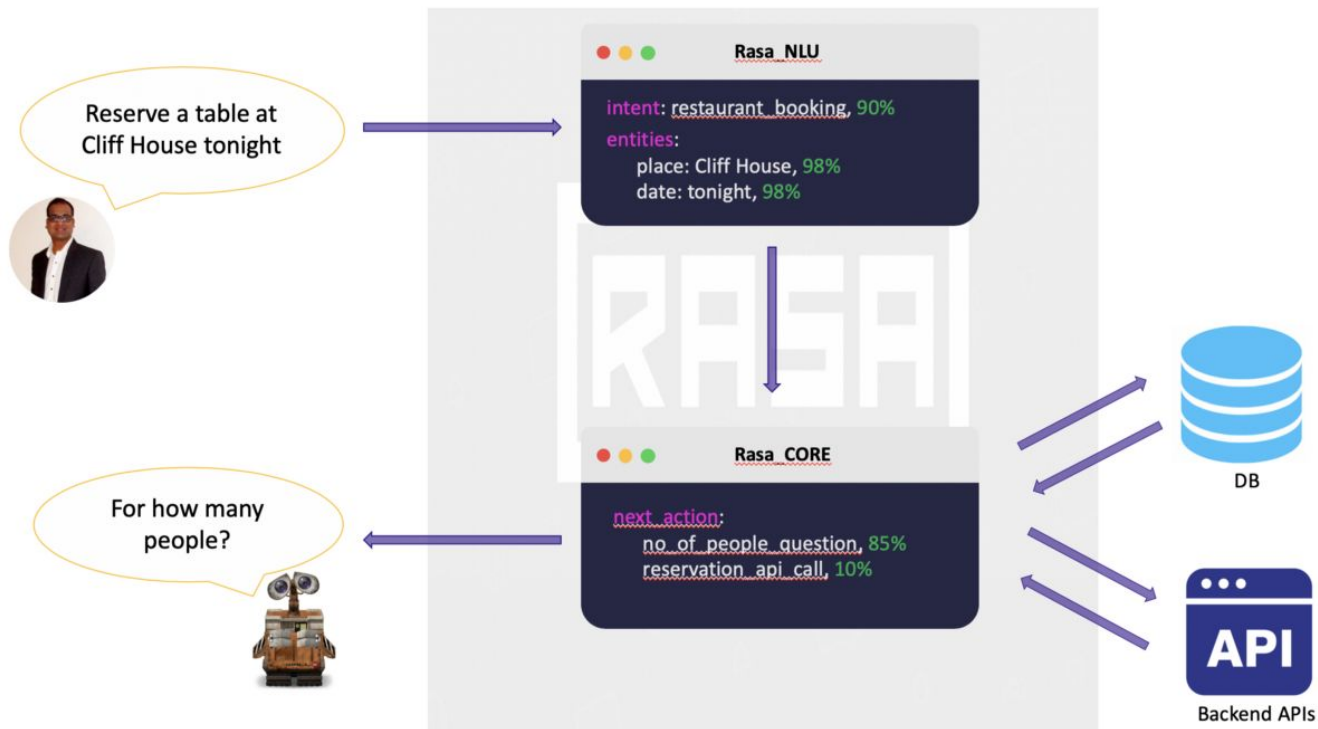
- action_add_new_people_or_groups_to_meeting
- action_remove_people_or_groups_from_meeting
- action_show_upcoming_events
- action_schedule_event
- action_inform_doing_scheduling
- utter_slot_values
- utter_submit
- validate_schedule_meeting_form



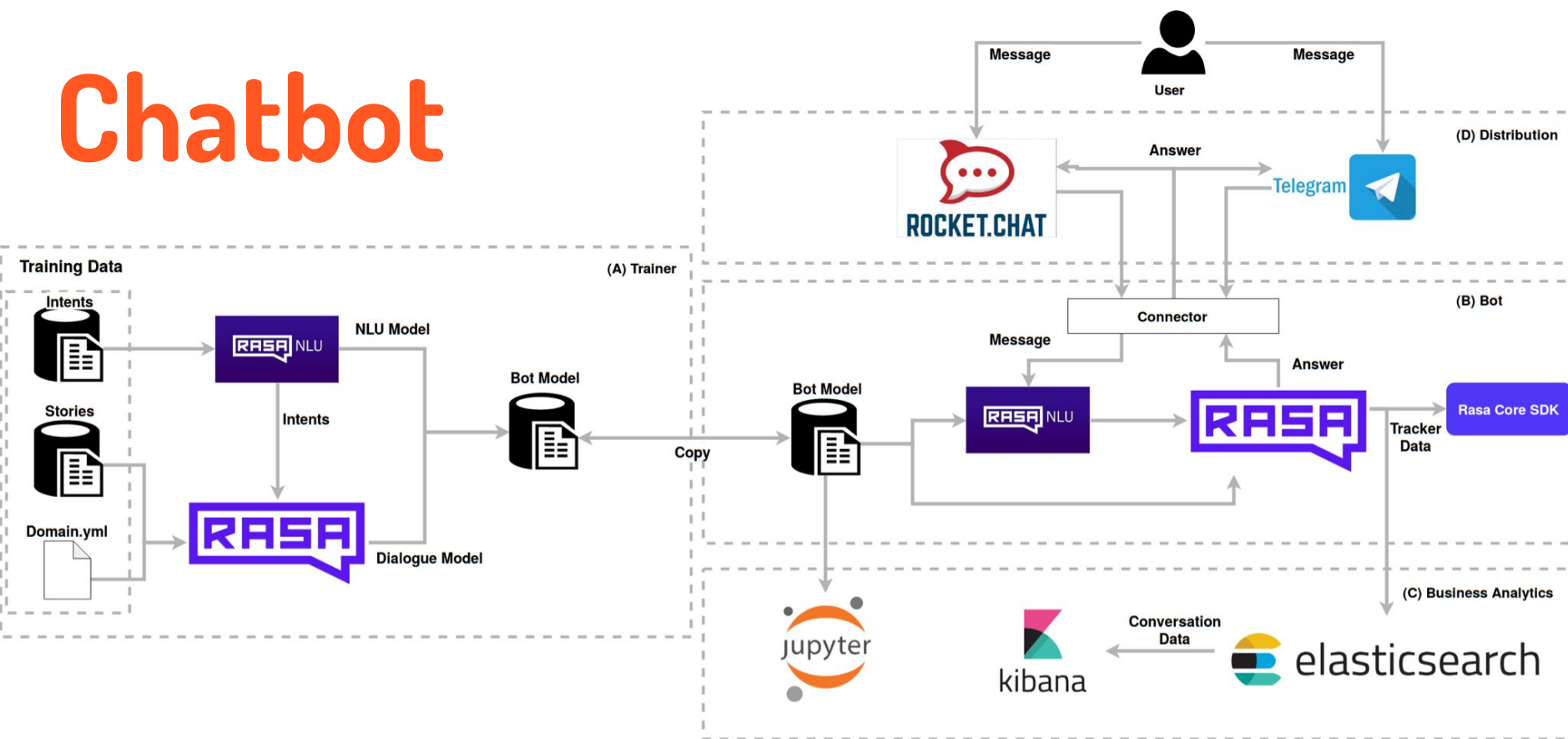
Scheduling Algorithm



Chatbot



Chatbot



Chatbot Pipeline

- SpacyNLP
- RegexFeaturizer
- LexicalSyntacticFeaturizer
- CountVectorsFeaturizer
- DucklingHTTPExtractor
- DIETClassifier
- EntitySynonymMapper
- ResponseSelector

spaCy



<(*)

(>)

Deployment

We open the necessary ports in order to users, hangouts chat and the components to communicate with each other.

- Database → MongoDB Atlassian
- Front-end → DigitalOcean with Docker
- Back-end → DigitalOcean with Docker
- Duckling → DigitalOcean with DockerHub
- Rasa → DigitalOcean (nohup for running it as back process)

Security Analysis & Testing

- Authentication with JWT and Google Calendar APIs
- Helmet.js

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMNTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.XbPfbHMI6arZ3Y922BhjWgQzWXcXNrZ0ogtVhfEd2o 3

1 Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

2 Payload

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

3 Signature

```

    HMACSHA256(
      BASE64URL(header)
      .
      BASE64URL(payload) ,
      secret)

```



HELMET

Express.js security with HTTP headers.

Source Code Analyzer

BASIC SCAN INFORMATION

Information

File

WEBSITE.ZIP

SHA256 Hash

5EE505721BD087FCBE3C6BF9157BF6EAB8041FB2DD60ECE96479E8A86C6AC9E0

Total Files Scanned

67

Types of Issues 17

Total No of Issues 37

DISTRIBUTION OF SEVERITY BY ISSUE TYPES

Severity



NO OF FILES SCANNED VS ISSUES IDENTIFIED

Detections



BASIC SCAN INFORMATION

Information

File

WEBSITE.ZIP

SHA256 Hash

0D3D43B66AC79CD4668473CAD1BFF9F2FB0E0BFF5BF6704ACFE1CA7A8AB7625D

Total Files Scanned

791

Types of Issues 0

Total No of Issues 0

DISTRIBUTION OF SEVERITY BY ISSUE TYPES

Severity



NO OF FILES SCANNED VS ISSUES IDENTIFIED

Detections



Thank you for listening.