

# **Orbital Mission Control**

**Telegram Bots**

**Azeem Vasanwala | Simon Julian Lauw**



**Azeem  
Vasanthwala  
Y3 CS**



**Simon Julian Lauw  
Y2 CS**

# Simple Ground Rules

- Question? Use **the raise hand** 🙋 function on Zoom, or send your message in the chat
  - Don't flood the chat with random banter, it will be hard to pick out questions
- *Stuck?* Join the breakout room. Someone will be with you shortly
- Ask as many questions as you want, but keep it on topic!

What is a “bot”?


A software application that is  
programmed to do certain tasks

- Source: Cloudflare

What can I do with a bot?

# Common Use Cases

## How do people *usually* use bots?

- Performing simple tasks based on the input of the users such as:
  - “Get me the bus timings”
  - “Add these numbers up”
  - Solve the quadratic equation (**fun, I know**)
- to complex tasks such as:
  - “Find me the name of the object in this image”
  - “Be my friend and chat with me  ”

- Pre-req: You have a **telegram account** and a **gmail account**
- If you don't, use the zoom raise hand  function **NOW**



# Why Bots?

# Why Bots?

- ***“It’s Like a CLI on steroids”*** - [devops.com](https://devops.com)
- (Mostly) Navigationless Interface - No menu bars to dig around to find information and hence it can be a lot faster and efficient!
- Try finding the curriculum for an older batch on the SOC website without googling 🙄
- Try to find out the procedures of applying to a certain hall or residential college in NUS now without googling

etc...

# Telegram Bots





- Chat messaging platform
  - Similar to WhatsApp, WeChat, etc
  - Cloud based, not P2P
- Bots are **natively supported** on the platform
- Texting a bot is similar to texting another person

# Why Telegram Bots?

- Telegram native support for bots:
  - Vast array of SDKs and libraries to build a bot : <https://core.telegram.org/bots/samples>.
  - No complex integrations, device specific compatibility, etc. as most of them has been abstracted away by Telegram app.
- Familiar UI
  - We all know how to use messaging apps (Line, WhatsApp, QQ Chat, KakaoTalk, etc.)
  - Show what you have built to your friends and ask them to use it!!

# Some Famous Telegram Bots

- AirTrack
  - A bot that helps to keep track of the price of selected air routes
- Productivity Bots (e.g. Gmail Bots, Trello Bots, NV Bots, etc)
  - They are made to complement the web applications. For example, with gmail bots, you can check your new emails in telegram.
- Werewolf Moderator Bot
  - As the name suggests, it is a werewolf game moderator.

# Our Environment

- Google CoLaboratory
  - Python development environment on Google's servers
    - No autocompletion, very sad 😞
- Go to <https://colab.research.google.com/#create=true>
  - Link in zoom chat
- Create a notebook
  - Name it whatever you want, but this bot is a “request-response” example so maybe something along those lines

# Creating a Bot

- Launch Telegram
- Search for the user **@BotFather**
- Create a Bot
  - Name it whatever you want
  - Give it whatever handle you want, but make sure it ends in bot



Back to CoLab

# Getting started

- Pre-req: Your packages are installed, type `!pip install python-telegram-bot` in the Google Colab notebook
- Go to <https://tinyurl.com/2022MCBot01>
- Copy the contents into a CODE block on CoLab
- Update the token with your own token
- Run the bot once
- Text the bot and say hello to it, it should echo “hello” back to you

What's going on?

# So now... Solve the quadratic equation

- Expected input: 3 numbers representing a, b and c from  $ax^2+bx+c$ 
  - e.g: `/quadratic 1 2 1`
- Expected output: `"Your roots are {r1} and {r2}"`
- If you have no python experience, there will be a small hint to get you started
  - Googling the rest of the way will be enough
- Aim to finish in next 15 mins

Great! You can manipulate input



# Common Interactions

How do people *usually* use bots?

- Perform manipulation of input from user
  - “Get me the bus timings”
  - “Add these numbers up”
  - Solve a quadratic equation (**fun, I know**)
    - Let’s build a bot to do this
  - Pre-req: You have a telegram account and a gmail account
    - If you don’t, use the zoom raise hand function **NOW**

# Common Interactions

## How do people *usually* use bots?

- Retrieve information from somewhere else
  - Get information from a database or API (e.g Bus bots)
- Lets create another command that can get a picture of a cat
  - `https://cataas.com/cat`
  - API you need: `update.message.reply_photo(photo=url)`
- Do you keep getting the same picture back?
  - Telegram is caching the url
    - Fix by appending a random number to the end as a query parameter

# Common Interactions

How do people *usually* use bots?

- Alert user when something has happened
  - It started raining
  - Your semester results were released
- Lets make another bot that alerts the user when a webpage updates
  - How do we interact with a webpage programmatically?



# Side track: HTTP Verbs

What? Verbs? This isn't English class...

- HTTP: HyperText Transfer Protocol
  - Basically how the internet communicates
- 4 Main Verbs
  - GET [Retrieve]
  - POST [Create]
  - PUT [Update]
  - DELETE [Delete]

# Common Interactions

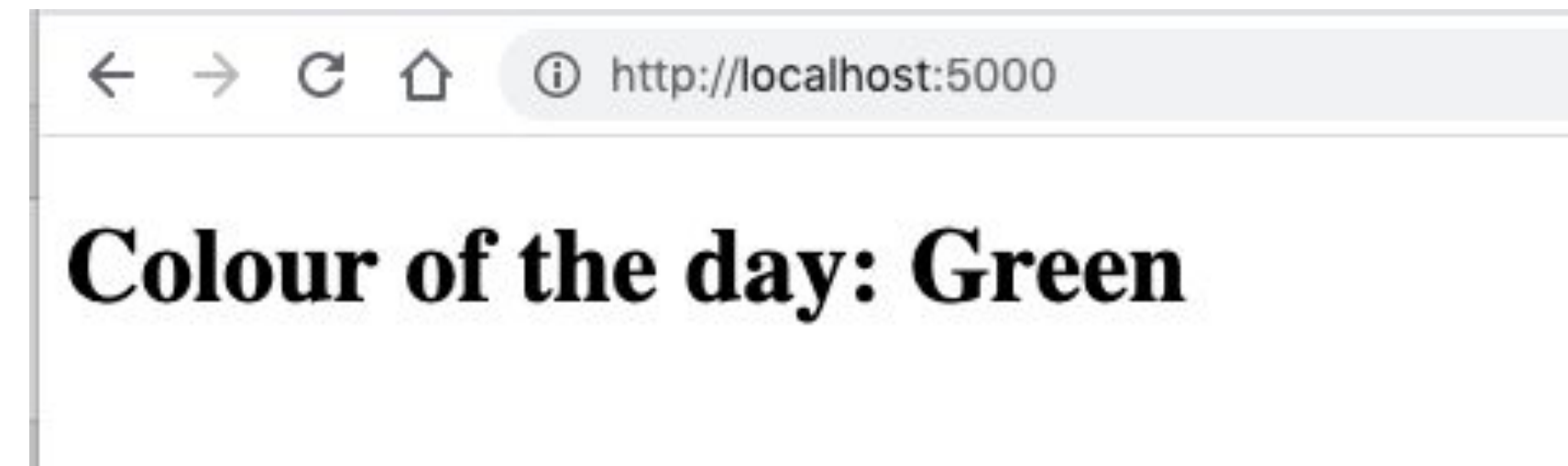
How do people *usually* use bots?

- Alert user when something has happened
  - It started raining
  - Your semester results were released
- Lets make another bot that alerts the user when a webpage updates
  - How do we interact with a webpage programmatically?
    - HTTP requests using the python `requests` library

# Your second bot

- Create a new notebook on CoLab
  - Import the following packages:
    - `python-telegram-bot`
    - `requests`
- Go to <https://tinyurl.com/2022MCBot02>
- Copy the contents into a CODE block on CoLab
- Update the token with your own token

# What your bot will do



- Query a webpage that looks like this =>
- When the colour is “Blue”, send a message to yourself
  - How do you message yourself?
    - You need your `chat_id`. Send `/start` to `@userinfobot`
- Where is the website?
  - On my computer, I will give you a URL soon.
  - Don't do anything funky, its just a simple webpage, if you DDOS it others can't query it and I will be very sad

Great! You can monitor other  
services 😄

# Let's Check the Code Out!

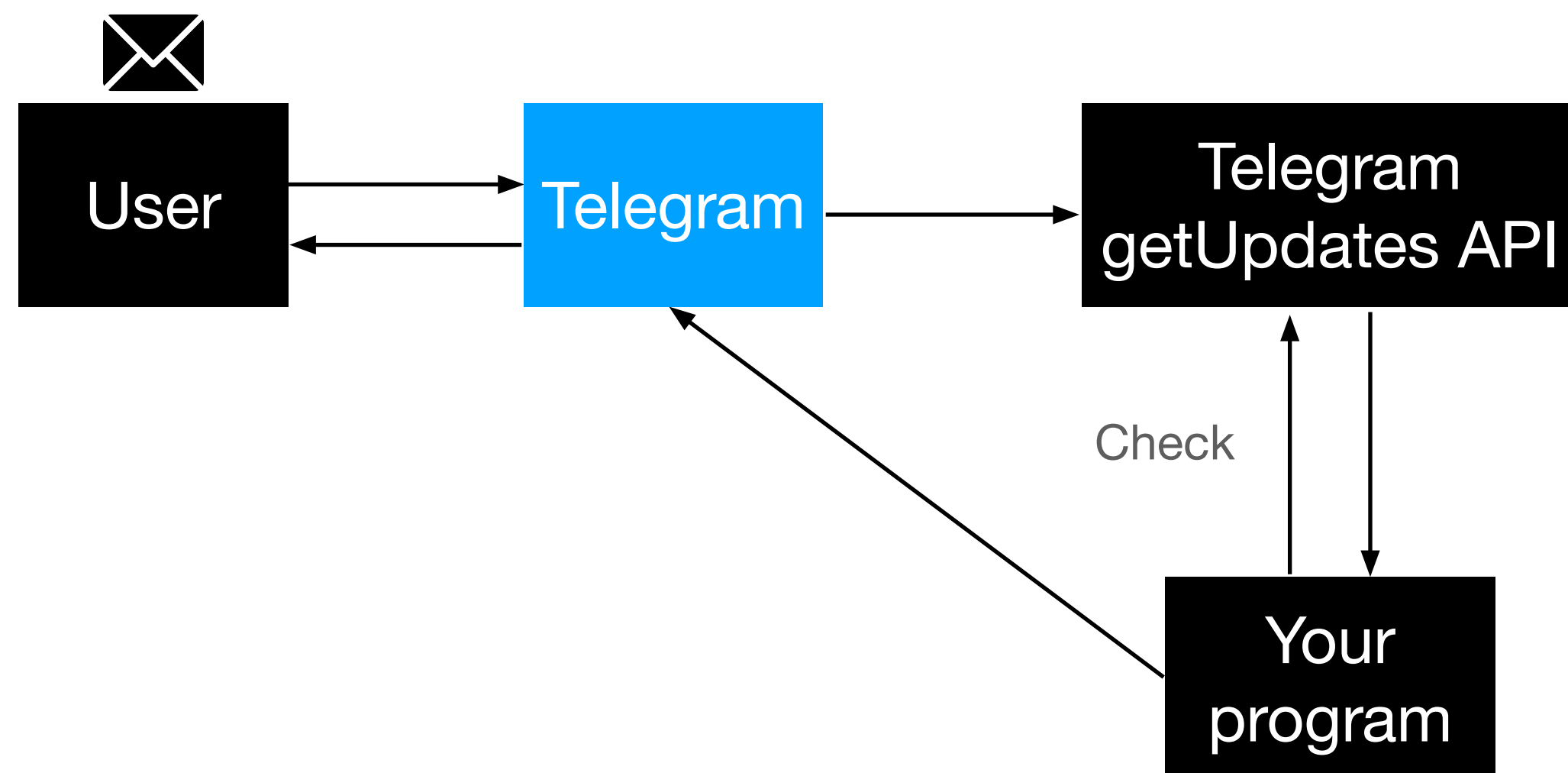
- Lets do something simple: An attendance bot
  - Follow along to understand the implementation first, then try it yourself later
    - <https://github.com/simonjulianl/orbital-2022-telegram-bots/blob/master/attendance-bot/bot.py>
    - [https://colab.research.google.com/drive/1TzvyALRF\\_z2buXLb7WOiWiM28nfw97rl](https://colab.research.google.com/drive/1TzvyALRF_z2buXLb7WOiWiM28nfw97rl)

Let's take a look

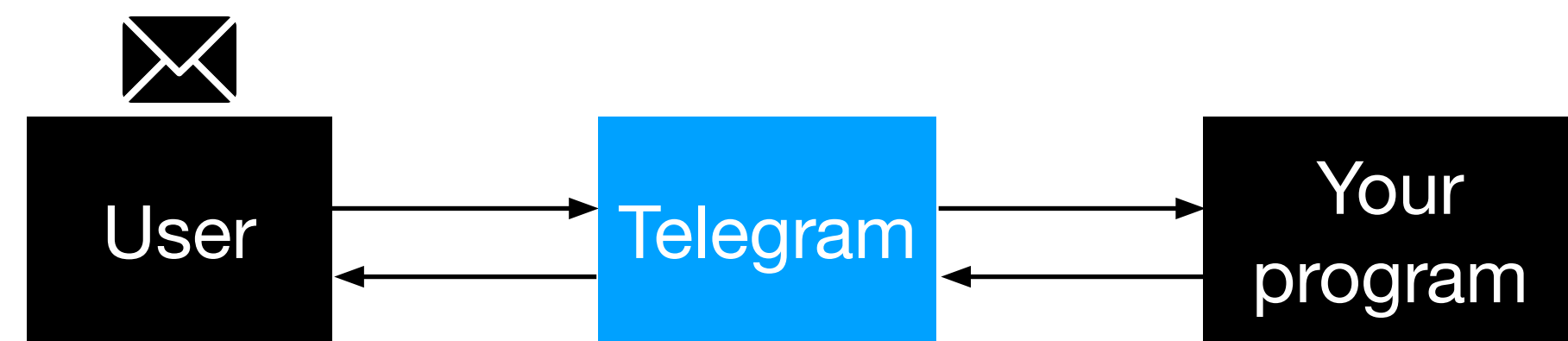
# Extra: Webhook vs Polling

- Remember `updater.start_polling()` ?
- What does that mean?

## Polling



## Webhook





# Extra: Hosting your Telegram Bot

- To this end, all of our telegram bots codes have been hosted by Google Colab, but a Google Colab Session is only 12 hours maximum in duration for a single session and Google Colab definitely is not a hosting platform
- There are a few options if you want to host your own Telegram Bot:
  - Heroku, Digital Ocean, etc: Cloud Platform-as-a-Service
  - AWS Beanstalk: Orchestration service to deploy applications, has support for AWS RDS (Database), Load Balancer, etc.
  - Starts your own Raspberry Pi server using any web server tools such as Nginx, or alternatively can use Ngrok

# Further Reading & Resources

## Telegram's Bot API

- Telegram
  - <https://core.telegram.org/bots/api>
- Other things you can do
  - Accept payments (<https://core.telegram.org/bots/payments>)
    - Talk to @ShopBot
  - Log In with Telegram (<https://core.telegram.org/api/passport>)
  - Create Games (<https://core.telegram.org/bots/games>)

# Further Reading & Resources

## SDKs

- <https://github.com/python-telegram-bot/python-telegram-bot>
  - Learning By Example
  - Documentation
- Want to use nodejs?
  - <https://github.com/yagop/node-telegram-bot-api>
- Other SDKs for other languages: <https://core.telegram.org/bots/samples>

# Good Practices

- Your token is precious
  - Don't commit it to Git
    - People can do nasty things with it
  - Inject it into your code via an environment variable
- Your username and user\_id is precious too
- Manage your packages properly
  - Use pip, npm, maven, whatever

# What we did today

- Request-Reponse:  
<https://colab.research.google.com/drive/15REolxRolyObRxBy3iEzNR6Bv7GjvieY>
- Server-Side Alerts:  
<https://colab.research.google.com/drive/1R655bmfnPaA-qRL6j3wHOjfZZkzrnpca>
- Group-based Interactions:  
[https://colab.research.google.com/drive/1TzvyALRF\\_z2buXLb7WOiWiM28nfw97rl?usp=sharing](https://colab.research.google.com/drive/1TzvyALRF_z2buXLb7WOiWiM28nfw97rl?usp=sharing)
- Git Repo with everything from today:  
<https://github.com/simonjulianl/orbital-2022-telegram-bots>