TELEGRAM BOT

(UNA INTRODUZIONE)

Giuseppe Augiero

11 gennaio 2016 - Area della Ricerca di Pisa

TELEGRAM

- Alternativa a Whatapp.
- Nato nell'**agosto** del **2013**.
- Principali differenze rispetto ai suoi concorrenti:
  - Maggiore sicurezza.
  - Interfaccia user-friendly.
  - Protocollo open source.
  - Sono open anche il client ufficiale e le API.
  - Maggiore e più semplice sviluppo di applicazioni di terze parti.

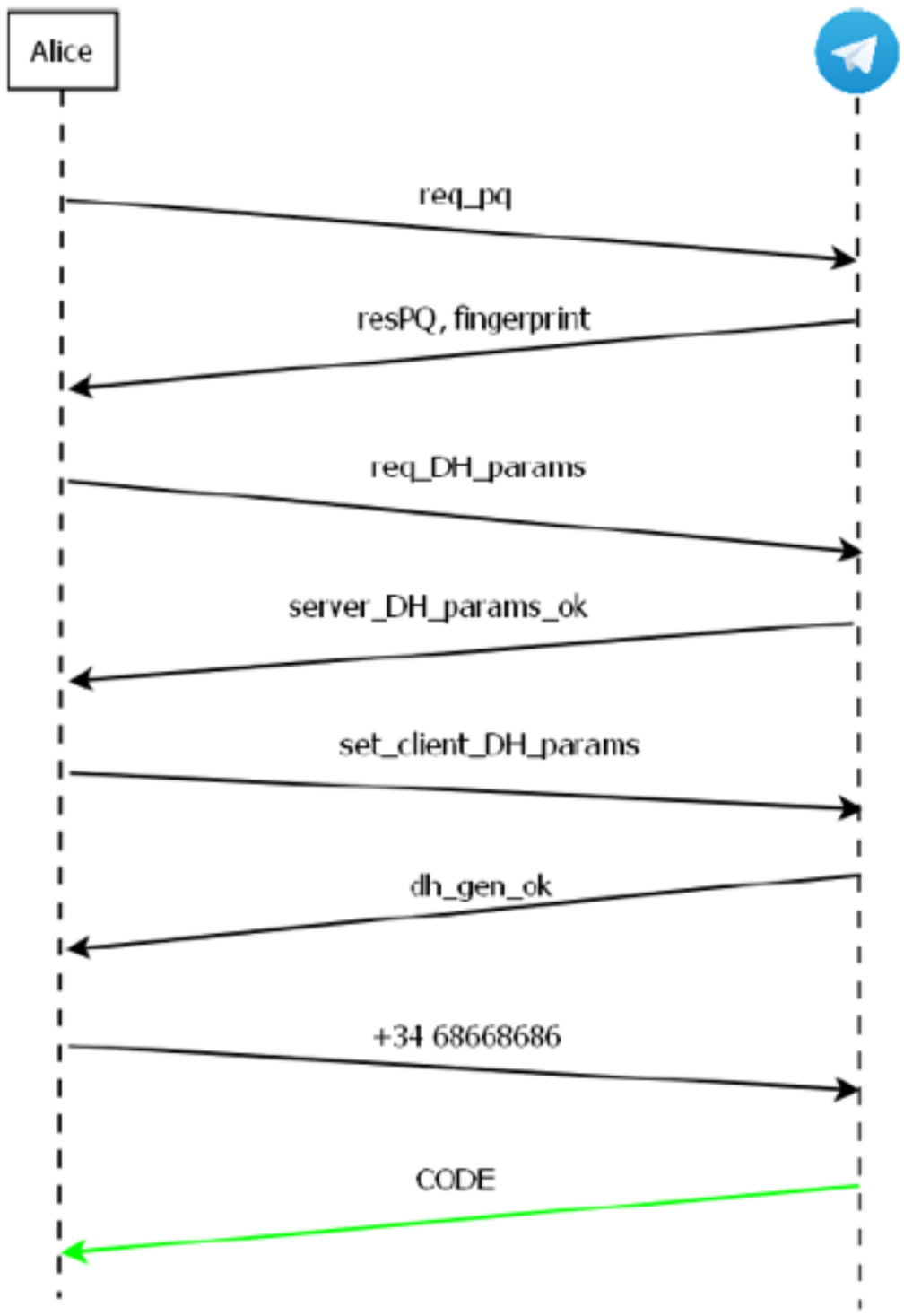
SECURITY

- Non viene rispettato il **principio di Kerckhoffs**.
  - *"… un sistema crittografico dovrebbe essere sicuro anche se tutto ciò che riguarda quel sistema, tranne la chiave, è di dominio pubblico …"*
- Il sistema di autenticazione di Telegram presenta alcune debolezze che possono portare a gravi vulnerabilità.
- E' possibile prendere il controllo della vittima attraverso la modifica del client usato per accedere a Telegram.

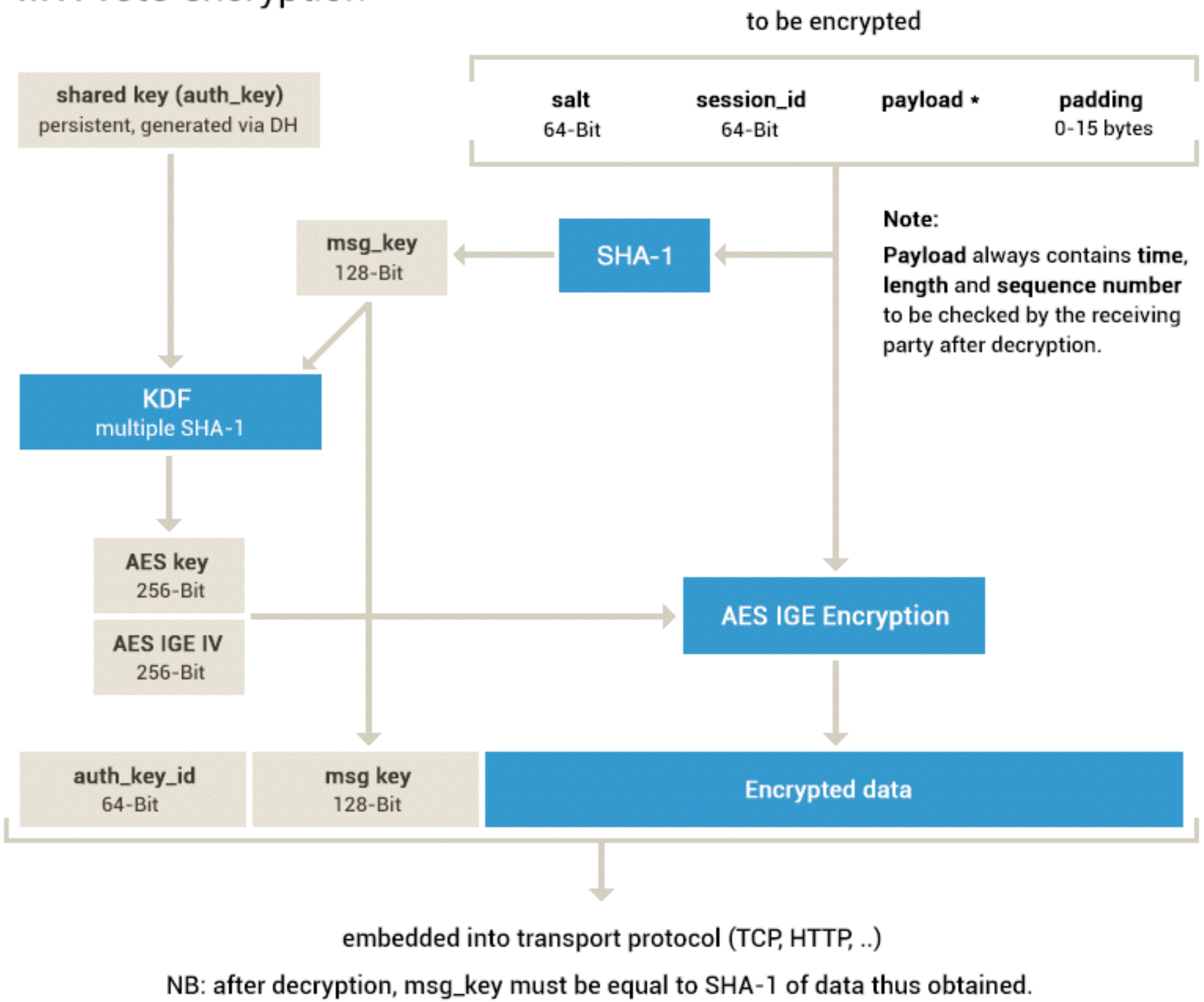
AUTENTICAZIONE

- Quando un nuovo utente installa un client di telegram in un dispositivo compatibile, il client comunica con i server di telegram al fine di creare una chiave condivisa.
- La chiave, chiamata "***authorizzation key***" verrà usata per cifrare tutte le comunicazioni tra client e server.
- Per scambiare le chiavi Telegram utilizza il protocollo Diffide-Hellman.
- La chiave non viene mai trasmessa in chiaro.
- Il client può verificare se sta dialogando con un server Telegram legittimo.

SETUP – AUTENTICAZIONE



MTProto encryption




The Botfather

I BOT DI TELEGRAM

- Dalla versione 3.0 di Telegram, tra le varie novità, è stata resa pubblica la possibilità di creare Bot.
- E' possibile "pilotare" il Bot attraverso GET e POST e ricevere i risultati attraverso Json.
- Non occorre associare il Bot a uno specifico numero di telefono.
- Lo username di ogni bot deve terminare per bot (p.es. @myuserbot).

DIFFERENZE TRA BOT E USER

- I Bot non hanno lo stato online o offline ma bensì “bot”.
- E’ possibile personalizzare la tastiera.
- Il bot può interagire con l’utente solo dopo che quest’ultimo abbia dato uno start.
- Lo spazio di storage è limitato.
- Se il bot viene aggiunto a un gruppo non riceverà tutti i messaggi.

CREAZIONE DI UN BOT

- Per creare un bot occorre scrivere al bot **@BotFather**.
- E' possibile consultare la documentazione attraverso il link **http://core.telegram.com/bots**
- Il comando da usare per la creazione di un bot è **/newbot**
- Il comando è interattivo.
- Ci verrà richiesto prima il nome e poi lo username (da 5 a 32 caratteri) relativo al bot.
- Riceveremo da BotFather un TOKEN che ci occorrerà per gestire il nostro bot.
- E' possibile utilizzare il comando **/help** per l'help in linea.

COMANDI DISPONIBILI

- /newbot - create a new bot
- /token - generate authorization token
- /revoke - revoke bot access token
- /setname - change a bot's name
- /setdescription - change bot description
- /setabouttext - change bot about info
- /setuserpic - change bot profile photo
- /setinline - change inline settings
- /setcommands - change bot commands list
- /setjoingroups - can your bot be added to groups?
- /setprivacy - what messages does your bot see in groups?
- /deletebot - delete a bot
- /cancel - cancel the current operation

ALCUNI ESEMPI DI BOT

- @ImageBot – send this bot a keyword and it'll provide you with a relevant picture.
- @TriviaBot – test your trivia knowledge or add to groups to compete with friends.
- @PollBot – add this one to group chats to create polls.
- @RateStickerBot – discover and rate new stickers.
- @AlertBot – set a time and this bot will send you a reminder for anything you like.
- @HotOrBot – find friends with this Tinder-like dating bot.
- @GithubBot – track GitHub updates.
- @StoreBot – find new bots and rate them.

ABBIAMO CREATO IL NOSTRO BOT

- Link per richiamare il nostro bot:
- https://telegram.me/your_bot

- E' possibile passare comandi attraverso parametri get.
- https://telegram.me/your_bot?start=value

RICHIESTE GET

- Ogni richiesta al bot avverrà via get in https.
- https://api.telegram.org/bot<token>/<metodo>
  - Dove al posto di <token> andrà inserito il token fornito da @BotFather, e al posto di <metodo> la specifica richiesta da voler dare.
- Per effettuare un test di buon funzionamento è possibile usare la funzione getMe:
- https://api.telegram.org/bot<token>/getMe
  - Il metodo ritorna le informazioni sul Bot, quali id, first_name, last_name e username.
- La prima cosa da fare, una volta creato il bot e verificato il suo funzionamento, è richiedere i messaggi ricevuti, per poi elaborarli e fornire una risposta.

COMUNICAZIONE BOT-SERVER

- Esistono due modalità di comunicazione:
  - Via GETUPDATES (facendo polling)
  - Via WEBHOOK

GETUPDATES

- E’ il metodo più semplice per “comandare” il bot ed è rappresentato da una richiesta GET cosi composta:
- https://api.telegram.org/bot<token>/getUpdates
- Eventuali parametri da passare a questa richiesta sono offset, limit, e timeout.
- Per esempio è possibile richiedere tutti gli update più recenti di quello numero 254544925, al massimo 10 update, e, se non sono presenti, attendere 30 secondi:
- https://api.telegram.org/bot<token>/getUpdates?offset=254544926&limit=10&timeout=30

- 254544925 è il valore del parametro update_id del metodo getUpdates
- GetUpdates fornisce, al massimo, gli ultimi 100 messaggi.
- Long polling problem.

SETWEBHOOK

- Un secondo metodopeper ricevere messaggi è setWebhook.
- Non è per neofiti.
- Webhook, attraverso l'impostazione del parametro url, indicherà al Bot che non appena quest'ultimo riceverà un messaggio, lo dovrà inviare, attraverso a una richiesta POST in HTTPS, all'indirizzo specificato con i dati formattati in JSON.
- Questo metodo, se usato, impedisce di ricevere i messaggi con getUpdates.
- Non supporta connessioni http e non supporta certificati auto-firmati.

RISPOSTE

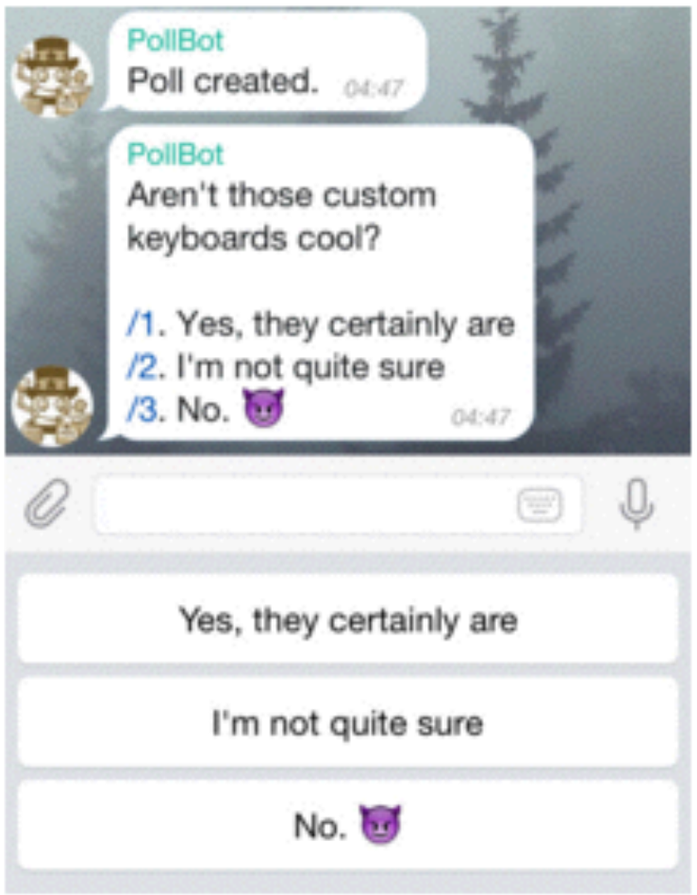
- Per rispondere ai messaggi ricevuto occorre usare il metodo SENDMESSAGE.

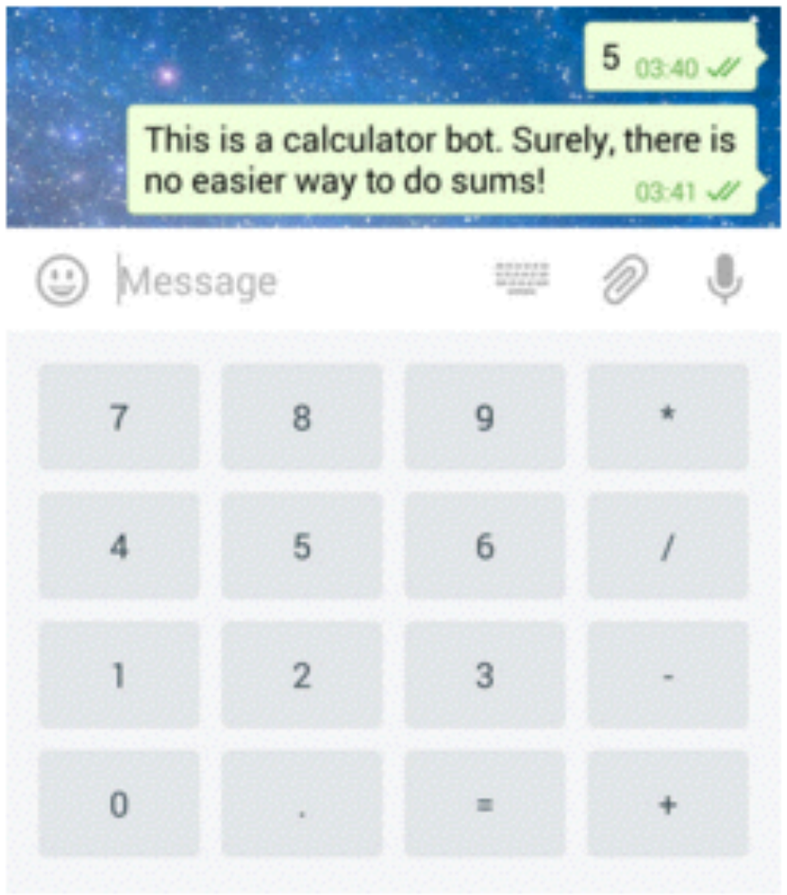
- I parametri obbligatori del metodo sendmessage sono l'id della chat (utente o gruppo che sia) e il testo del messaggio, rispettivamente chat_id e text.

- Il parametro chat_id è presente in qualsiasi messaggi ricevuto.

BOT PERKS

- E’ possibile personalizzare la tastiera virtuale usata per chattare con il bot.





LINGUAGGI E LIBRERIE.

- E’ possibile scrivere il proprio bot in qualsiasi linguaggio che permetta di effettuare una get e un post. Per esempio:
  - Php
  - Python
  - Ruby
  - Java
- Esistono librerie ad hoc per gestire il bot a un livello più alto.