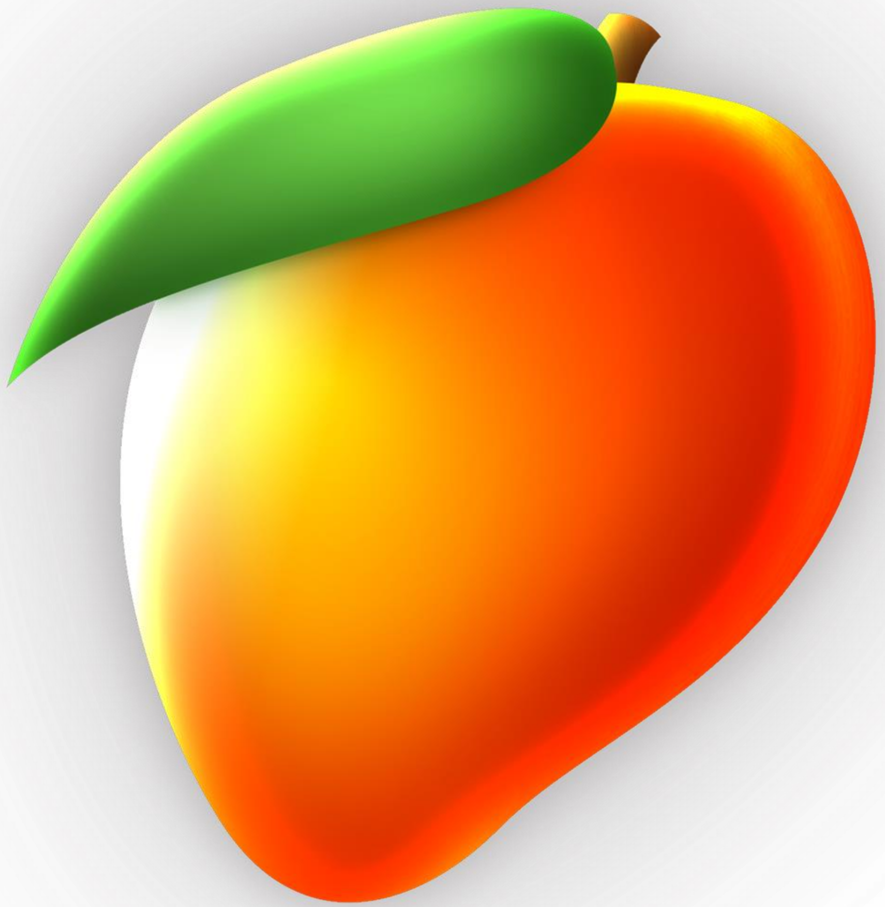
MANGO MESSENGER PROJECT

Authors:
Petro Kolosov (107417),
Serhii Holishevskyi (107619),
Illia Zubachov (107434),
Arslanbek Temirbekov (107419)

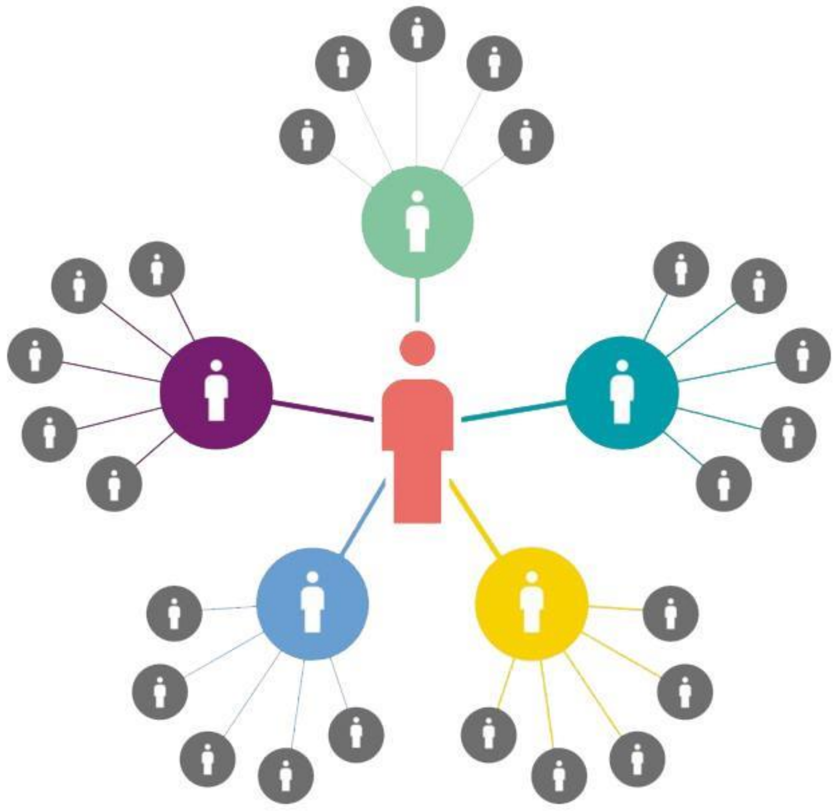


MANGO MESSENGER

Table of contents

MANGO MESSENGER

Introduction



Unsurprisingly, messengers have become widely popular communication tools across the globe. Quick and convenient communication channel, some of which have originated via social networking sites, have developed into independent social platforms.

“Messaging is one of the few things that people do more than social networking.” — Mark Zuckerberg, 2014

MANGO MESSENGER

Our mission, vision and motto

Mission

"To create a safe and comfortable platform for communication and ability to share any feelings for people"

Vision

"To provide free and simple way of communication between people around the world."

Motto/Slogan

"Privacy and personal security come first."

MANGO
MESSENGER

To be Mango

Mango Messenger is a free instant messaging service. It provides end-to-end encryption and focus on the speed and security of communication.

MANGO
MESSENGER

MVP’s on the market

WhatsApp > 2.0 bil

Viber > 800 mil

WeChat > 1.2 bil

Tencent QQ > 591 mil

Messenger > 1.3 bil

Snapchat > 538 mil

MANGO MESSENGER

Our project proposes

Business Usage

Personal Usage

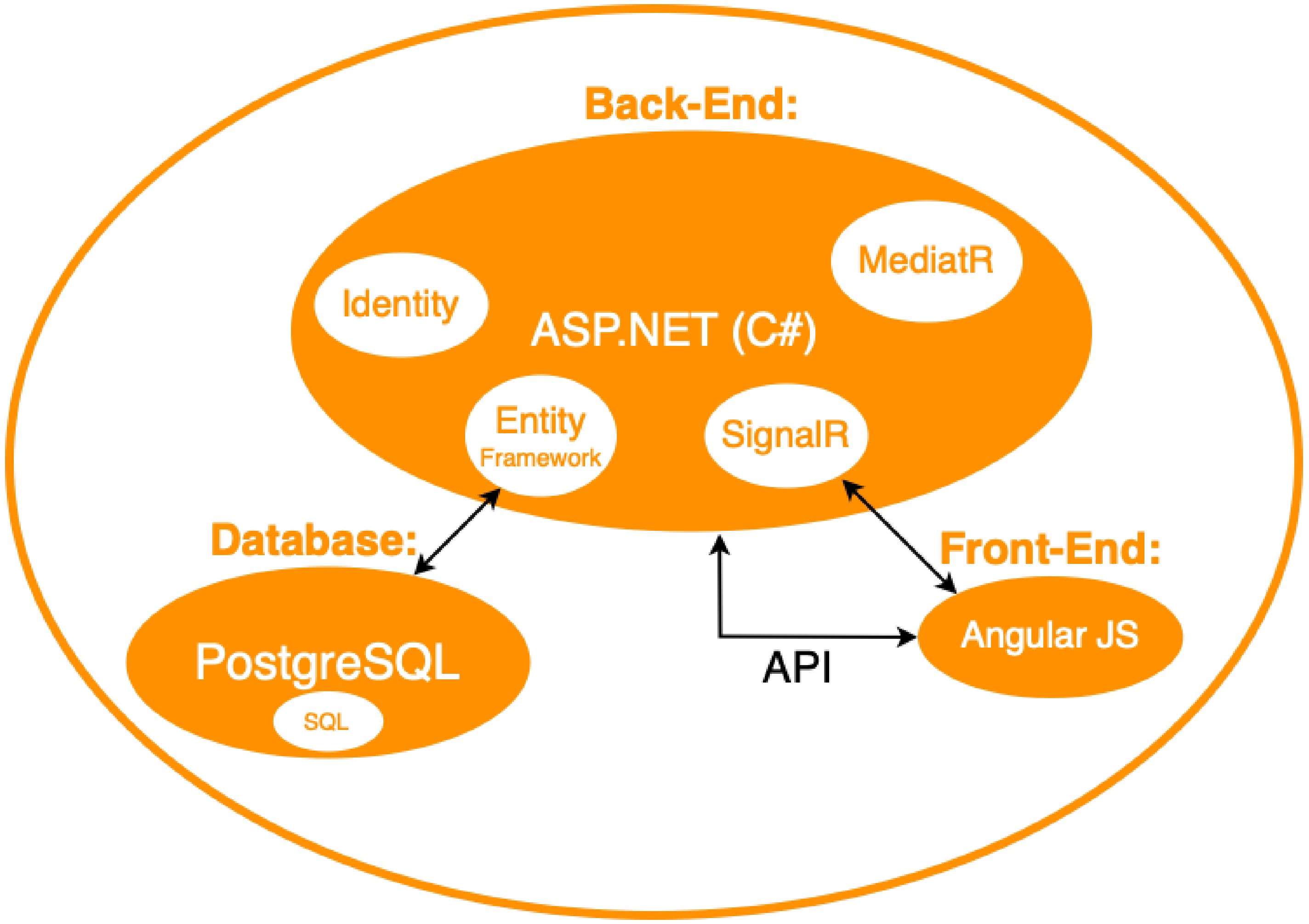
MANGO MESSENGER

Objectives

- To analyze the security and user privacy vulnerabilities of the IMS.
- To provide the system requirements.
- To propose web service (API's) architecture and authorization mechanism.
- To implement E2E encryption to the system.
- To implement Web service (API).
- To implement web, mobile and desktop clients.

MANGO MESSENGER

Project structure



Supported by:

Docker

Git

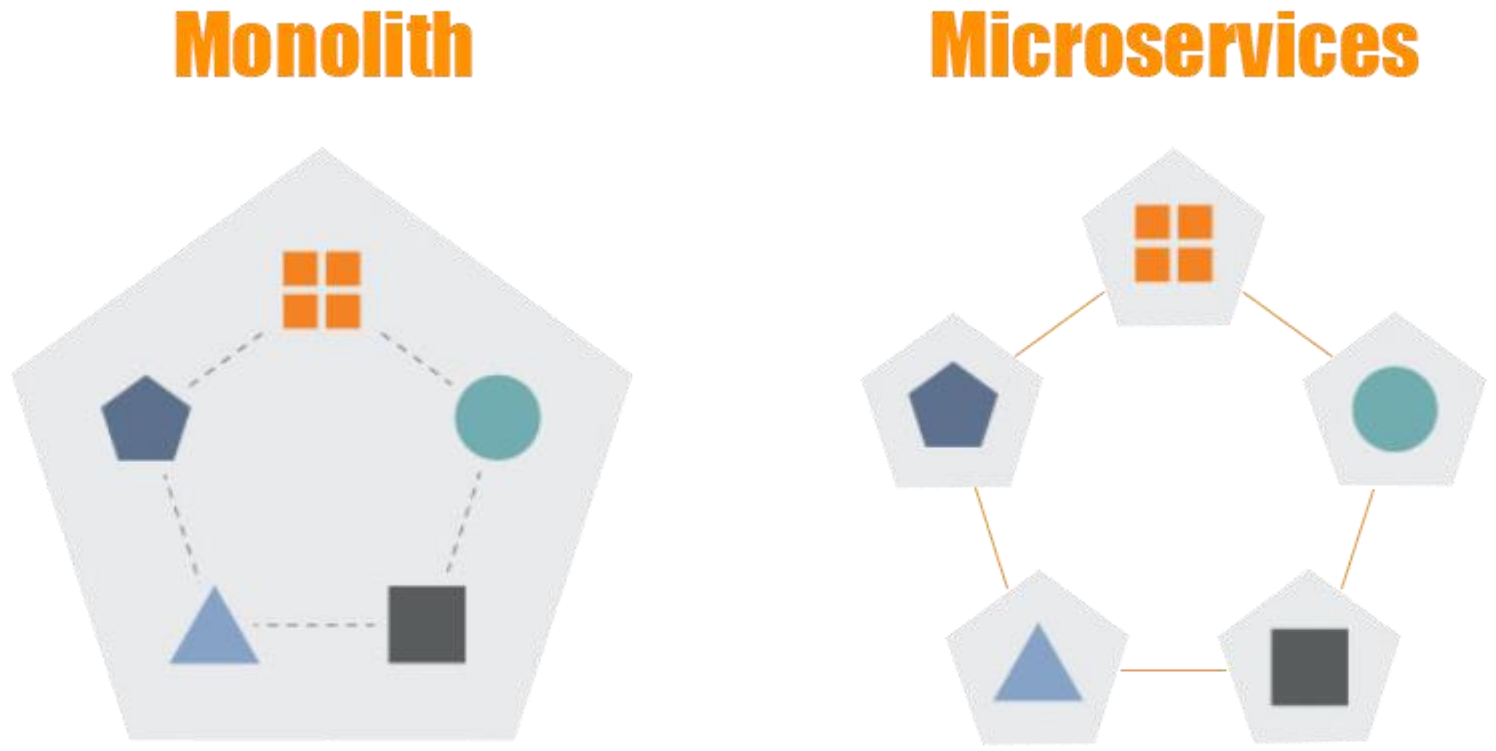
Heroku

Web-service architecture

Implementing the instant messenger system, we consider applying a well- known N-tier Monolithic Architecture [Bucchiarone et al., 2018], which provides a time-proven model that allows software developers to create flexible and reusable applications.

MANGO MESSENGER

Which architecture to choose?



MANGO MESSENGER

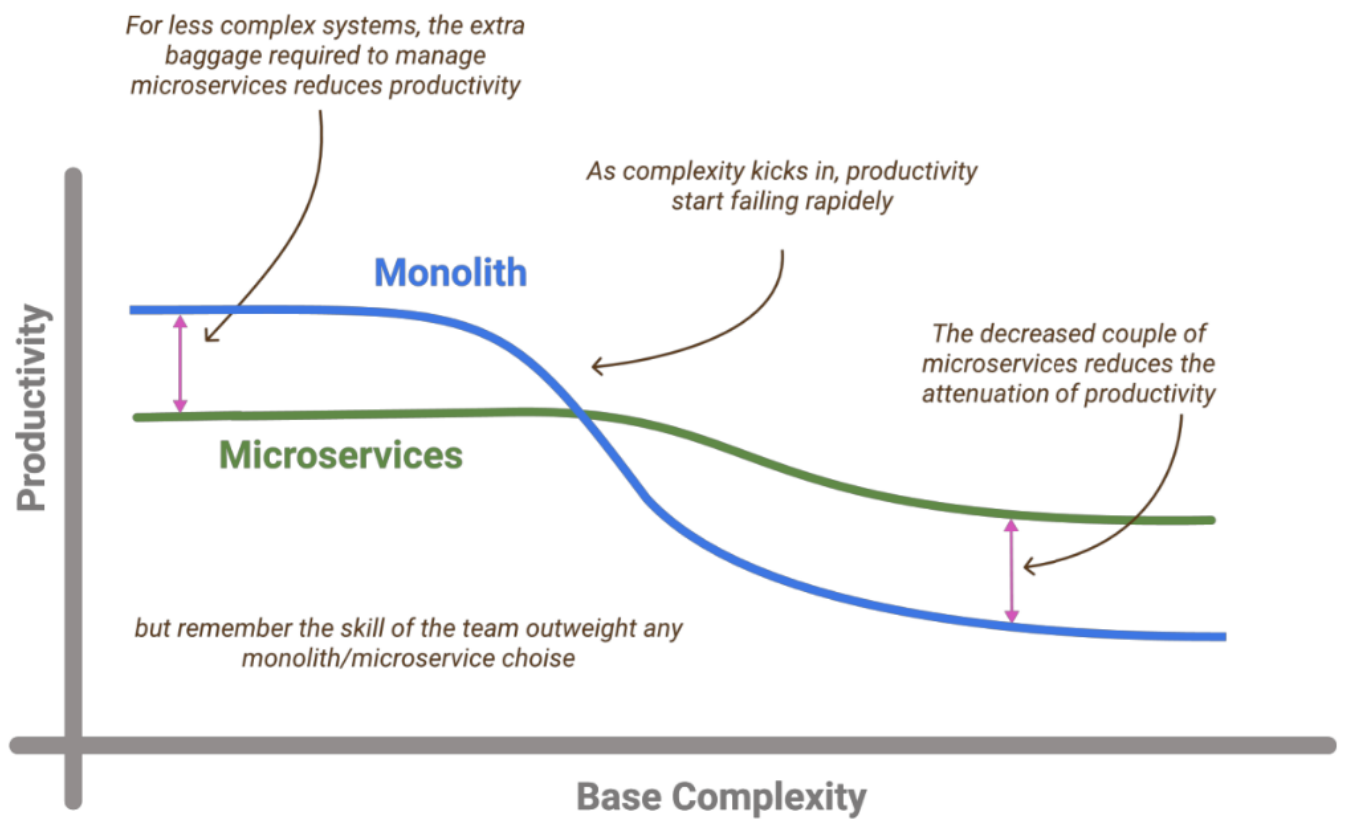
Which architecture to choose?

"You shouldn't start a new project with microservices, even if you're sure your application will be big enough to make it worthwhile."

Martin Fowler [Fowler, 2015b]

MANGO MESSENGER

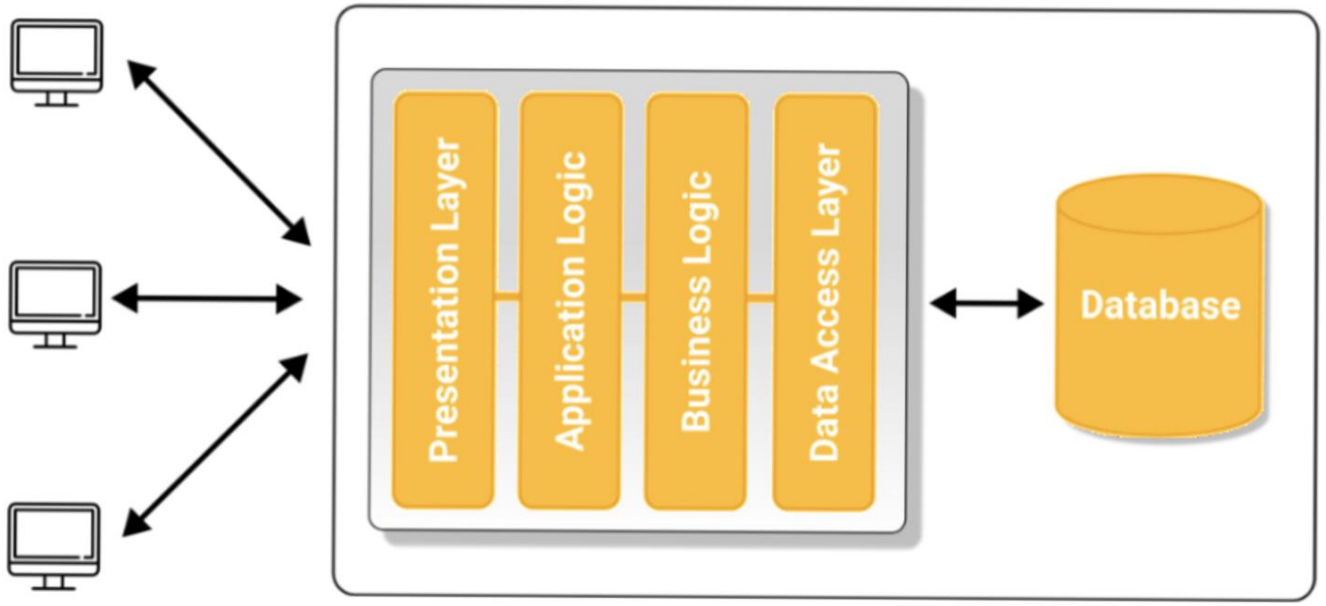
Monolith vs Microservices



Relation between system complexity and architectures. Source: [Fowler, 2015a].

MANGO MESSENGER

Monolith concept diagram



Monolith concept diagram. Source: [Mango Messenger Figma, 2021].

MANGO MESSENGER

Monolithic architecture: Pros and Cons

PROS:

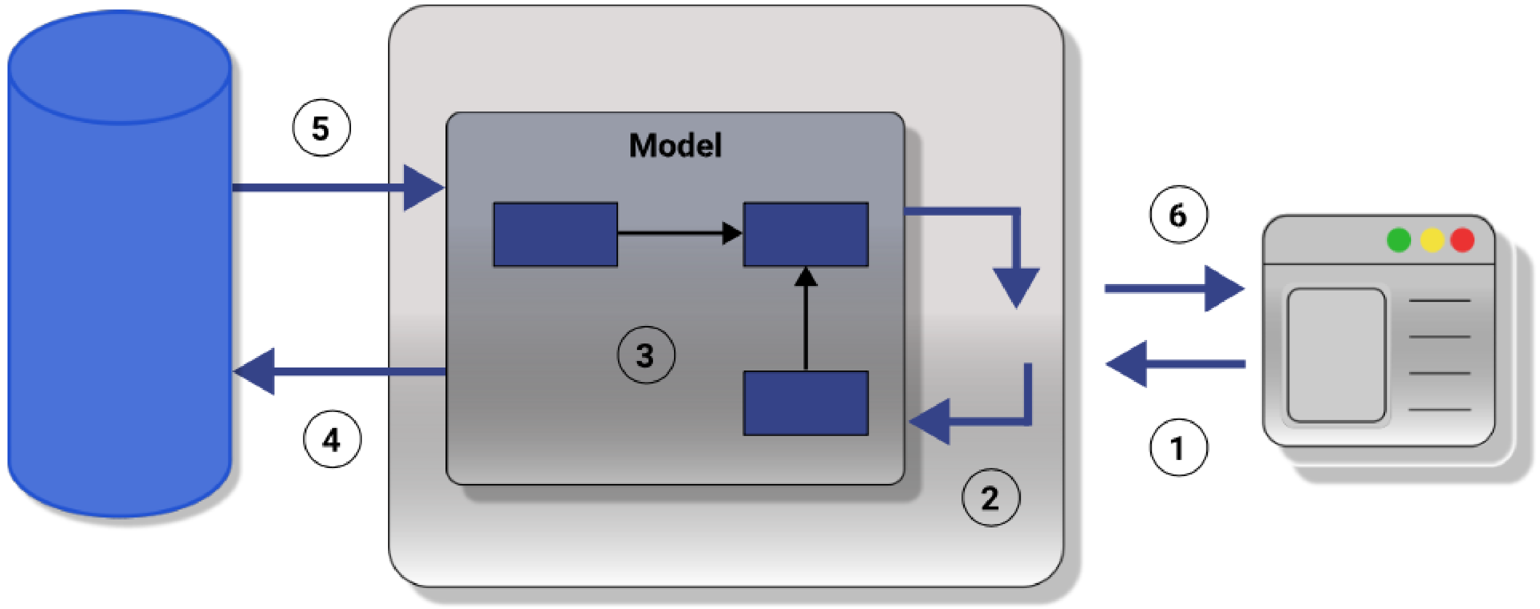
- *Simplicity*
- *Performance*
- *Easier development*
- *Easier debugging and testing*

CONS:

- *Tight Coupling*
- *Understanding*
- *Reliability to errors*
- *Updates*
- *Technology stack*

MANGO MESSENGER

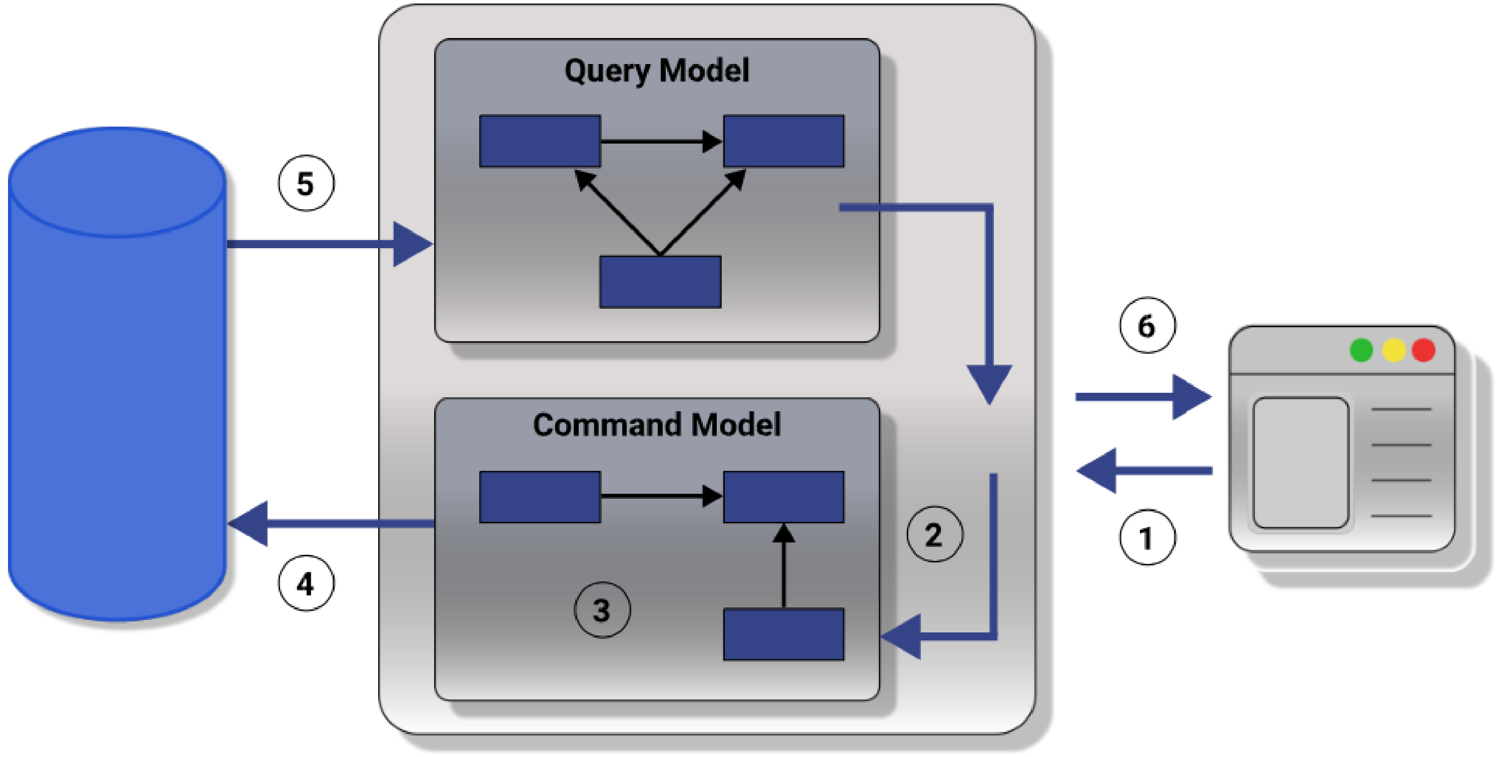
Service entity concept diagram



Service entity concept diagram. Source: [Fowler, 2011].

MANGO MESSENGER

CQRS Concept Diagram



CQRS concept diagram. Source: [Fowler, 2011].

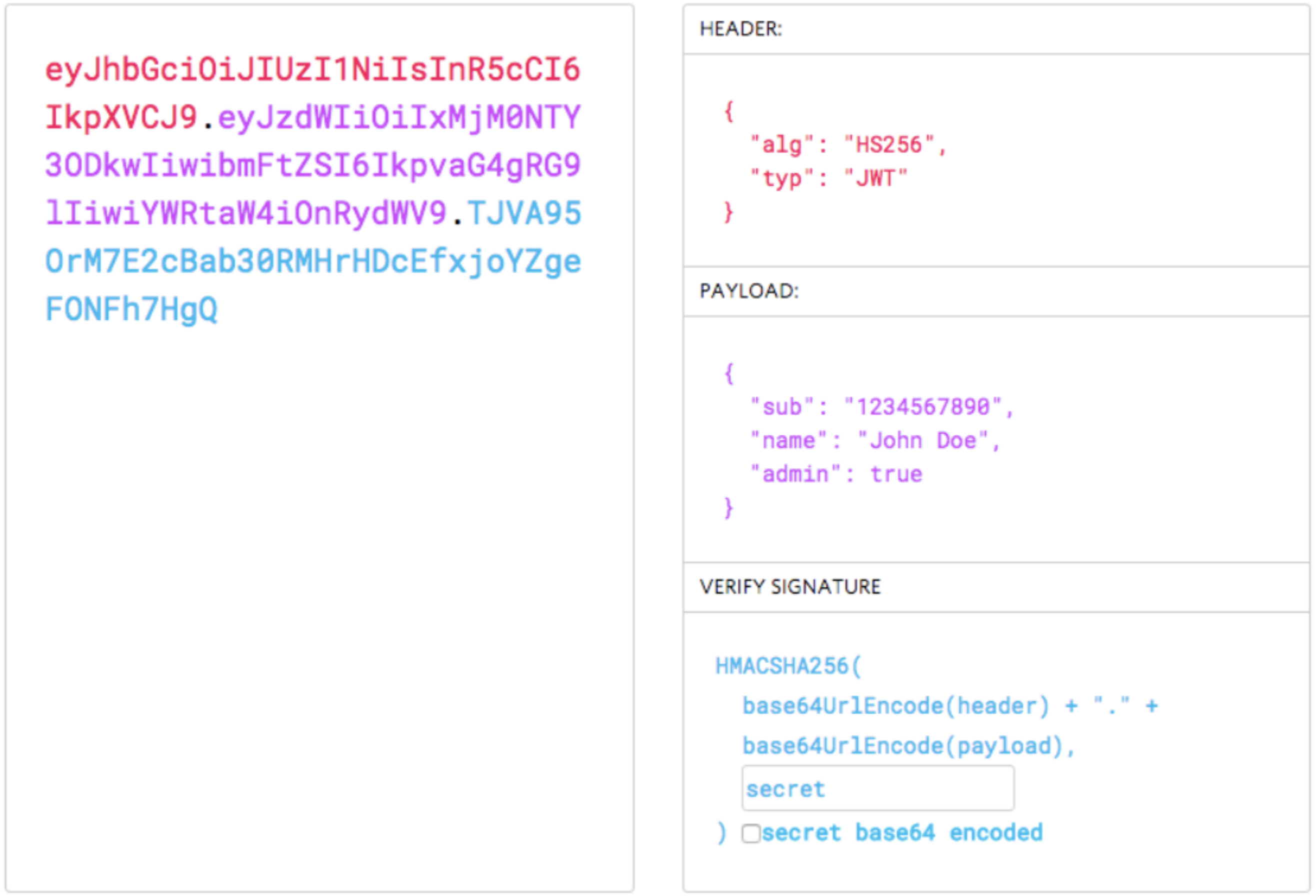
MANGO MESSENGER

Authorization

A JSON Web Token, or **JWT** is a standardized, in some cases signed and encrypted, data packaging format that is used to securely transfer information between two parties.

MANGO
MESSENGER

JWT Token



```
eyJhbGciOiJIUzI1NiIsInR5cCI6
IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
3ODkwIiwibmFtZSI6IkpvaG4gRG9
lIiwiYWRtaW4iOnRydWV9.TJVA95
OrM7E2cBab30RMHrHDcEfxjoYZge
FONFh7HgQ
```

HEADER:

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

PAYLOAD:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) ☐secret base64 encoded
```

MANGO MESSENGER

Recommendation for JWT usage

- JWT should have a short lifetime, since it cannot be revoked.

  However, extremely short lifetimes of the tokens would affect the overall performance of the system.

  Therefore, we consider access token lifetime to be 5 minutes and refresh token to be 7 days.

- JWT should be used in a single time, e.g. JWT per request.

  For each request client preliminarily checks access token lifetime.

  If access token it expired, client sends request for updating a pair of access-refresh tokens.

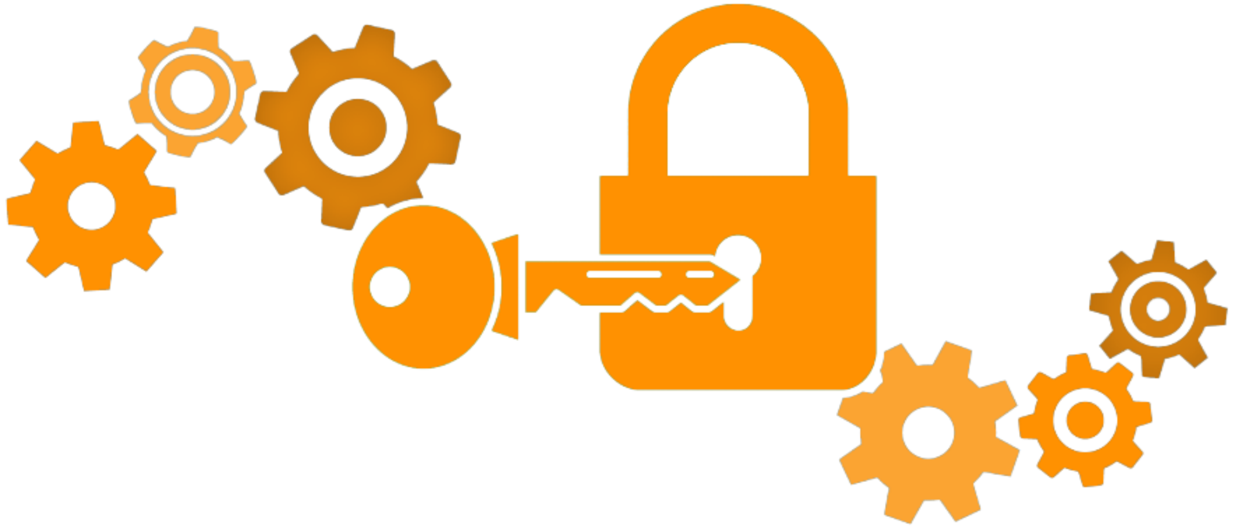
  For more confidence, we can update tokens a few seconds earlier.

MANGO MESSENGER

Encryption

Encryption – is a conversion of the original representation of the information, known as plaintext, into an alternative form known as ciphertext. The inverse operation is called Decryption.

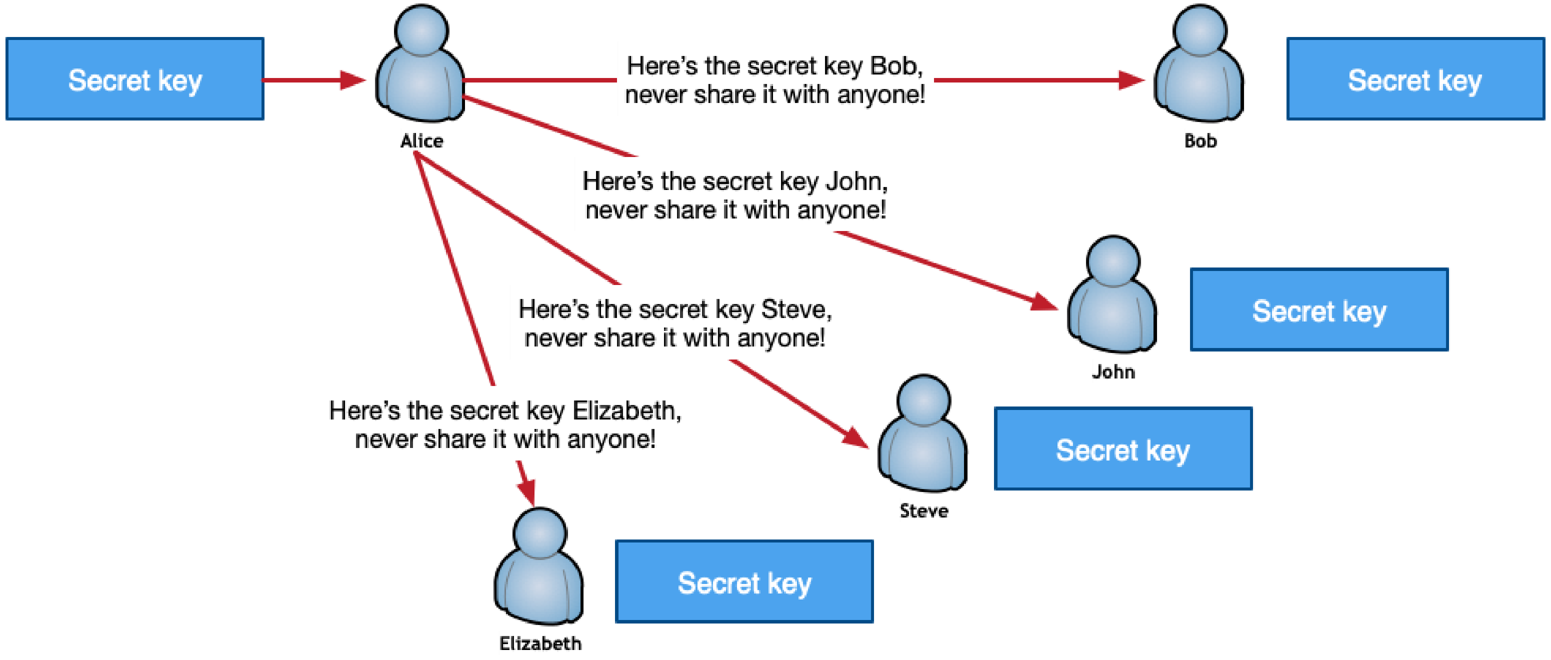
Encryption may be both symmetric or asymmetric.



MANGO MESSENGER

Symmetric cryptography

Symmetric encryption – is a type of encryption where only one secret key is used to both encrypt and decrypt electronic information.


Secret key
Alice
Here's the secret key Bob, never share it with anyone!
Bob
Secret key
Here's the secret key John, never share it with anyone!
John
Secret key
Here's the secret key Steve, never share it with anyone!
Steve
Secret key
Here's the secret key Elizabeth, never share it with anyone!
Elizabeth
Secret key

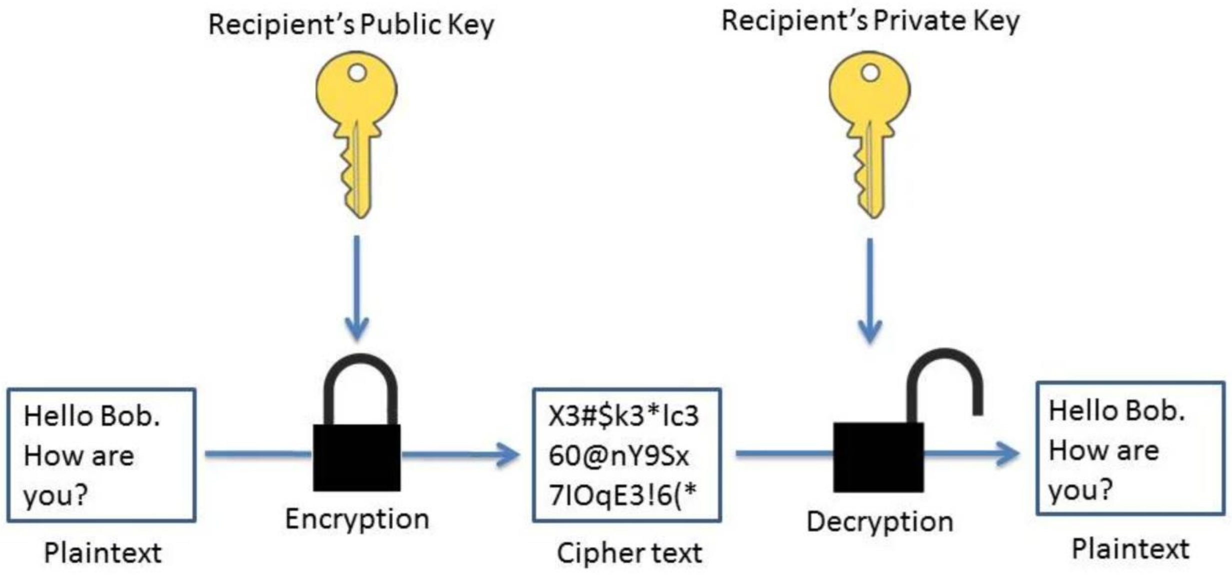
MANGO MESSENGER

Asymmetric cryptography

Asymmetric encryption – is an encryption such that a message is encrypted using public key, and decrypted using private key.

Asymmetric encryption uses computationally hard problems with a secret (private), and shared (public) key.

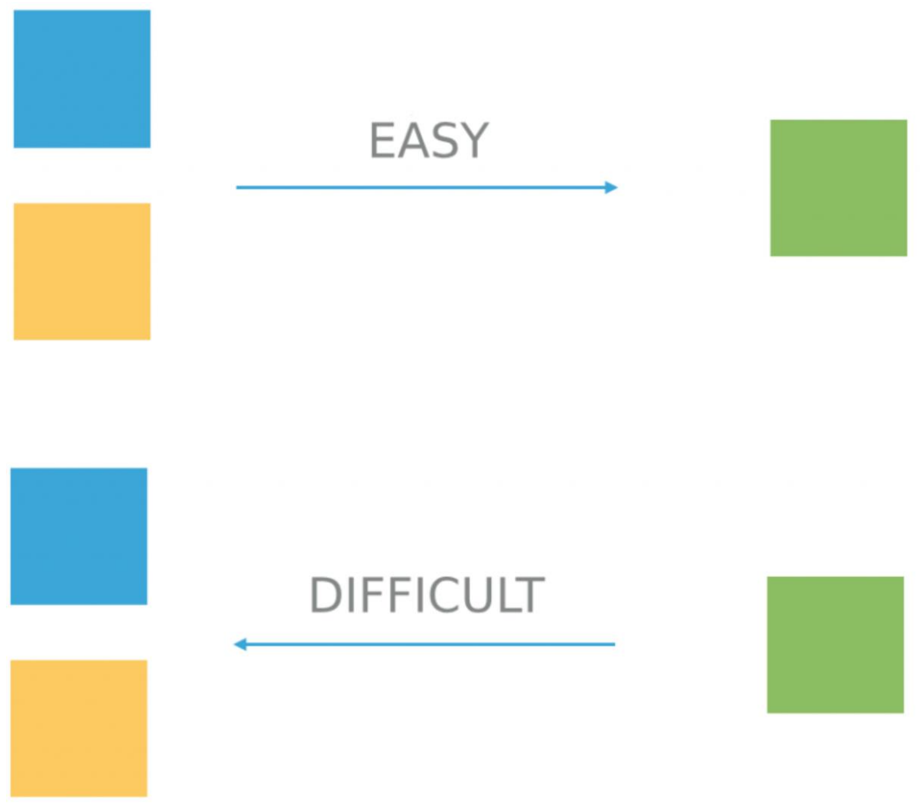
Asymmetric encryption is based on the concept of the one-way functions.



MANGO MESSENGER

One-way functions

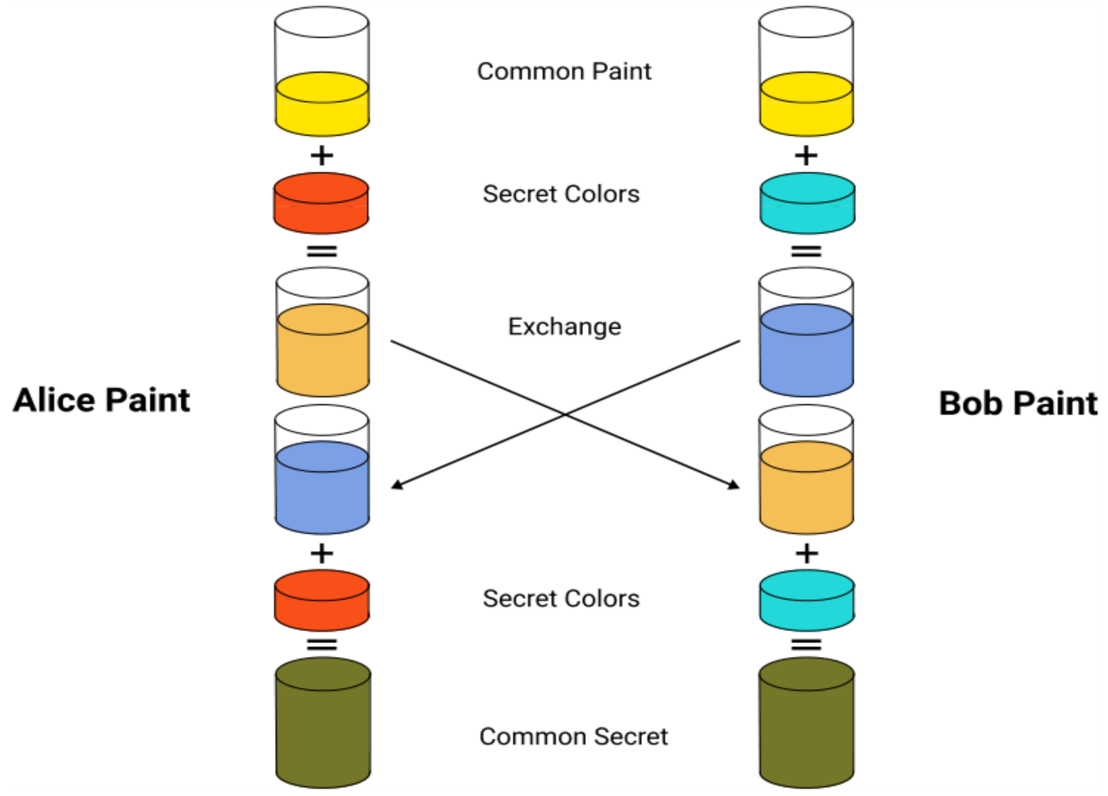
One-way function – is a function that is easy to compute on every input, but hard to invert given the image of a random input.



MANGO MESSENGER

DH Key Exchange

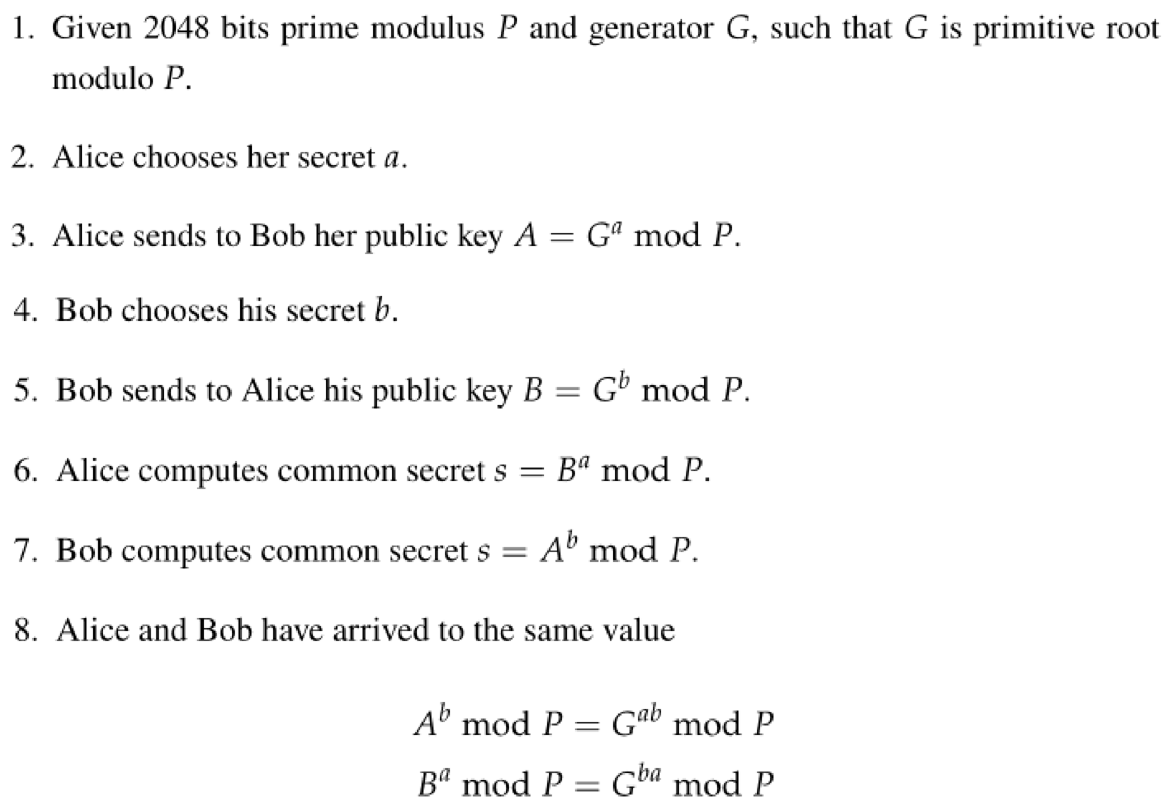
Diffie-Hellman key exchange – is a method of securely exchanging cryptographic keys over a public channel. Diffie-Hellman is based on the on the one-way function A = G^a mod P.



MANGO MESSENGER

Math behind the DH

Diffie-Hellman key exchange – is a method of securely exchanging cryptographic keys over a public channel. Diffie-Hellman is based on the on the one-way function A = G^a mod P.



1. Given 2048 bits prime modulus P and generator G, such that G is primitive root modulo P.
2. Alice chooses her secret a.
3. Alice sends to Bob her public key $A = G^a \bmod P$.
4. Bob chooses his secret b.
5. Bob sends to Alice his public key $B = G^b \bmod P$.
6. Alice computes common secret $s = B^a \bmod P$.
7. Bob computes common secret $s = A^b \bmod P$.
8. Alice and Bob have arrived to the same value

$$A^b \bmod P = G^{ab} \bmod P$$
$$B^a \bmod P = G^{ba} \bmod P$$

MANGO MESSENGER

End-to-end encryption

End-to-end encryption – is an asymmetric encryption such that the only communicating parties are able to decrypt the data.

End-to-end encryption may be implemented using

- **Asymmetric encryption algorithm (ex. RSA)**
- **Asymmetric common secret exchange + symmetric encryption (ex. DH + AES)**

In Mango Messenger E2E encryption is implemented via ECDH + AES256 symmetric encryption.

MANGO MESSENGER

Choose public constants properly

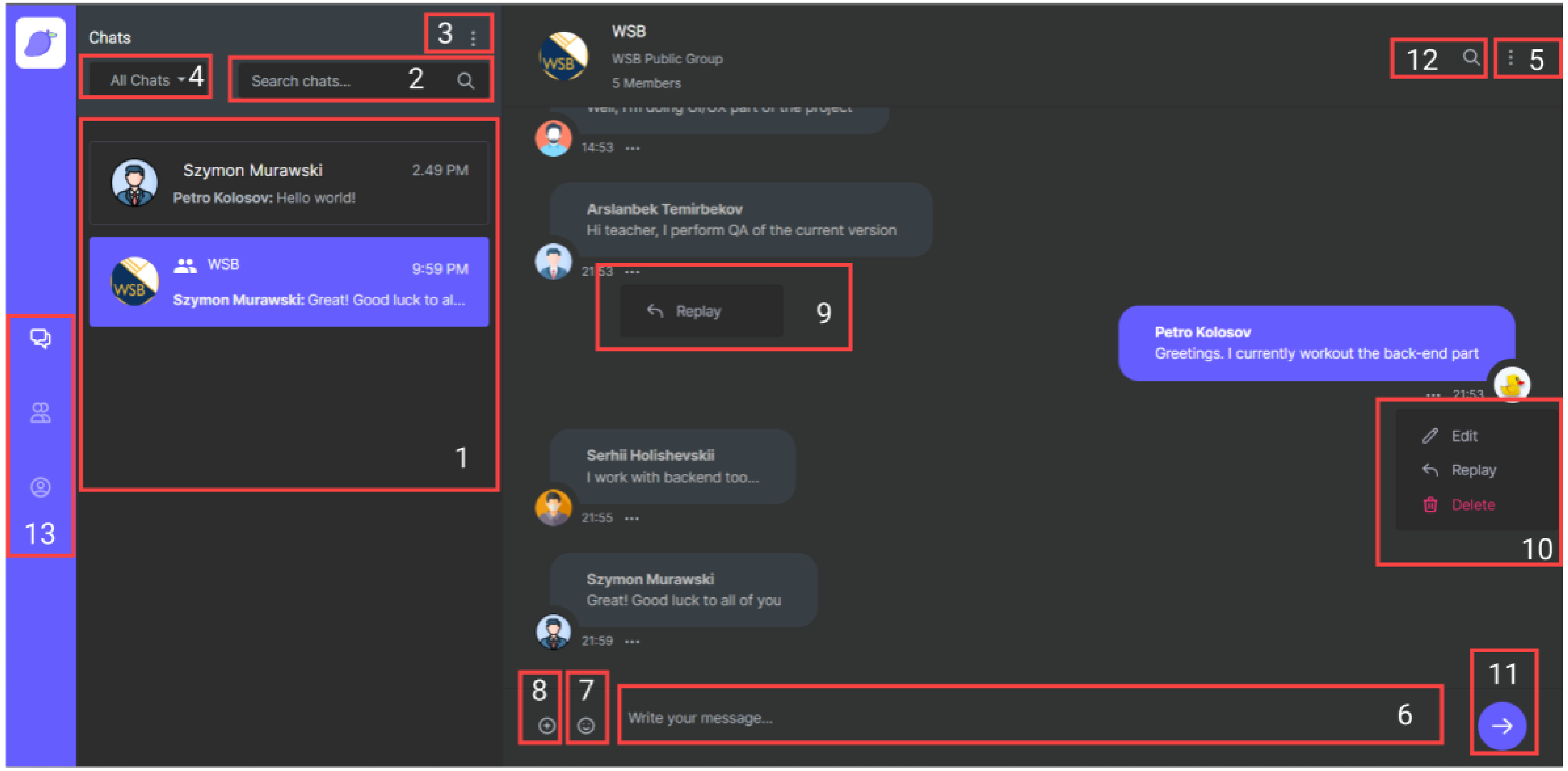
In DH key exchange, it is important to properly choose public constants.

It is recommended

- Modulus **P** to be **P = 2Q + 1** such that **2048** bits safe-prime because **Q** is also prime
- Generator **G** such that **G** is primitive root modulo **P**
- For elliptic curve cryptography it is enough to use **512** bit prime as **P**

MANGO MESSENGER

Project outcomes



Szymon Murawski 2.49 PM
Petro Kolosov: Hello world!

WSB 9:59 PM
Szymon Murawski: Great! Good luck to al...

Well, I'm doing UI/UX part of the project
14:53

Arslanbek Temirbekov
Hi teacher, I perform QA of the current version
21:53

Petro Kolosov
Greetings. I currently workout the back-end part
21:53

Serhii Holishevskii
I work with backend too...
21:55

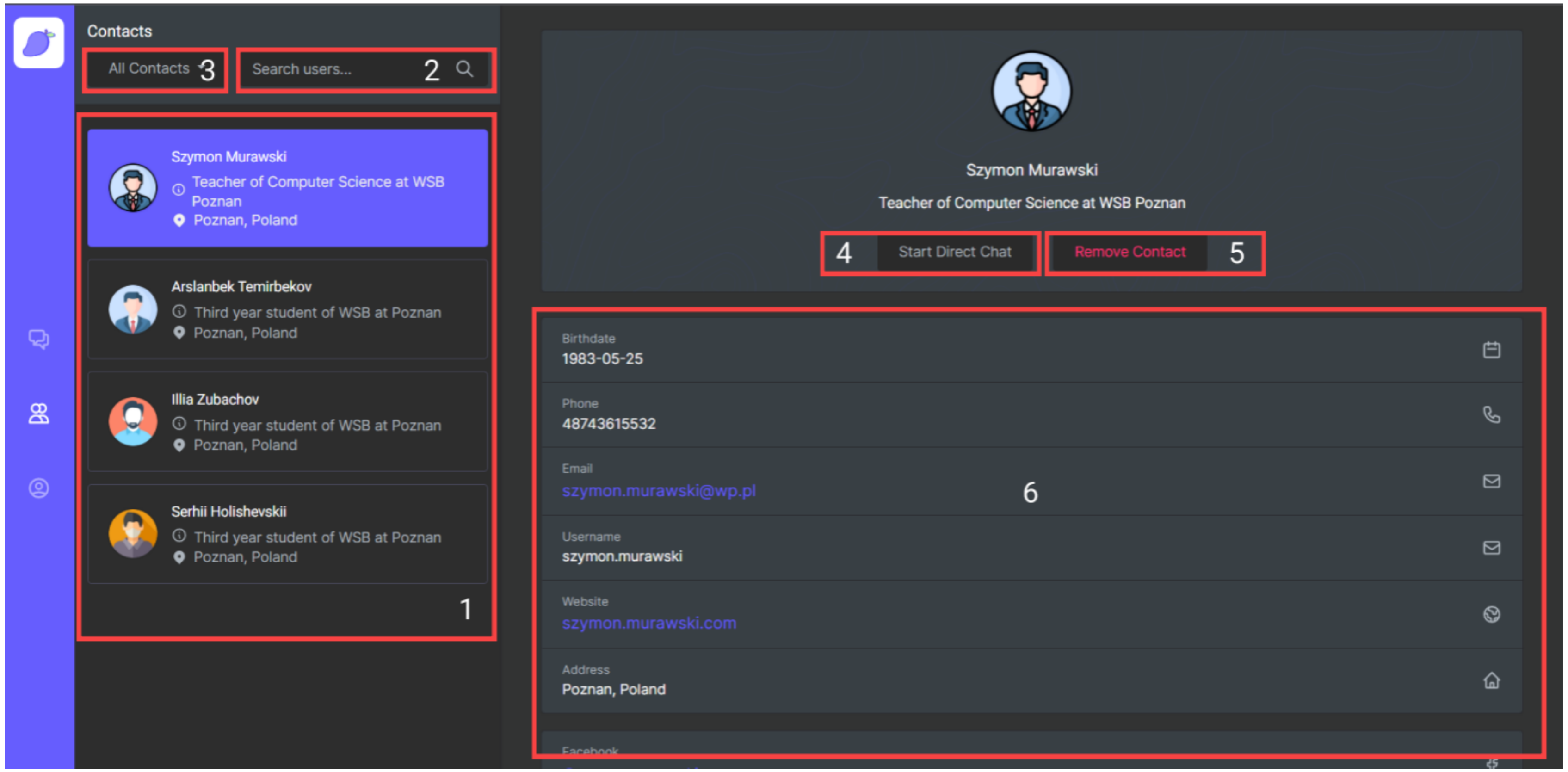
Szymon Murawski
Great! Good luck to all of you
21:59

1. Chatlist
2. Chat Searching
3. Creating Groups
4. Filtering Chats
5. Leaving or Sharing the Chat
6. Entering the Text
7. Adding Emoji
8. Adding Attachments
9. Replying to the Messages
10. Edit or Delete Messages
11. Send Message
12. Message Search
13. Navigation to the Main Page

https://www.front.mangomessenger.company

MANGO MESSENGER

Project outcomes



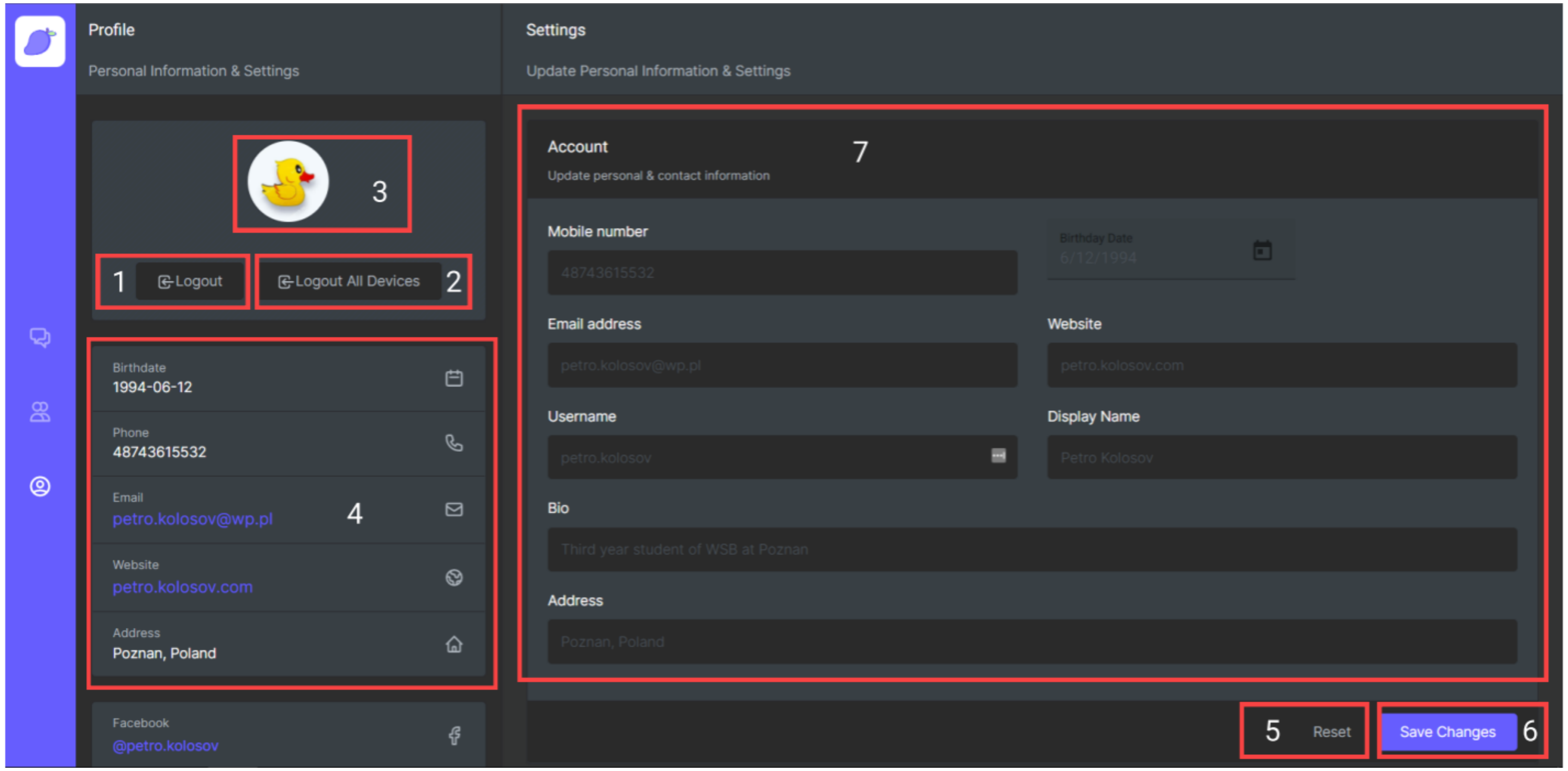
Birthdate	1983-05-25
Phone	48743615532
Email	szymon.murawski@wp.pl
Username	szymon.murawski
Website	szymon.murawski.com
Address	Poznan, Poland

1. User's Contact List
2. Searching Users
3. Filtering Contacts
4. Start a Chat
5. Delete User from Contacts
6. Specified User Info

https://www.front.mangomessenger.company

MANGO MESSENGER

Project outcomes



Profile

Personal Information & Settings

Logout

Logout All Devices

Birthdate
1994-06-12

Phone
48743615532

Email
petro.kolosov@wp.pl

Website
petro.kolosov.com

Address
Poznan, Poland

Facebook
@petro.kolosov

Settings

Update Personal Information & Settings

Account

Update personal & contact information

Mobile number: 48743615532	Birthday Date: 6/12/1994
Email address: petro.kolosov@wp.pl	Website: petro.kolosov.com
Username: petro.kolosov	Display Name: Petro Kolosov

Bio: Third year student of WSB at Poznan

Address: Poznan, Poland

Reset

Save Changes

https://www.front.mangomessenger.company

1. Log Out from current Device
2. Log Out from All Devices
3. User's Profile image
4. User's Information
5. Reset Updated Data
6. Save Updated Data
7. Panel for Changing Personal Data

MANGO MESSENGER

Bibliography

Alwin, Duane F and Brett A Beattie (2016). “The KISS principle in survey design: question length and data quality”. In: *Sociological methodology* 46.1, pp. 121–152.

Bellare, Mihir et al. (1997). “A concrete security treatment of symmetric encryption”. In: *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, pp. 394–403.

Brataas, Gunnar and Peter Hughes (2004). “Exploring architectural scalability”. In: *Proceedings of the 4th international workshop on Software and performance*, pp. 125– 129.

Bucchiarone, Antonio et al. (2018). “From monolithic to microservices: An experience report from the banking domain”. In: *Ieee Software* 35.3, pp. 50–55.

Burrows, Michael, Martin Abadi, and Roger Michael Needham (1989). “A logic of authentication”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 426.1871, pp. 233–271.

Chung, Lawrence et al. (2012). *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media.

Cohn, Mike (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Da Silva, Tiago Silva et al. (2018). “The evolution of agile UXD”. In: *Information and Software Technology* 102, pp. 1–5.

Degges, Randall (2019). *JWTs Suck*. https://speakerdeck.com/rdegges/jwts- suck. [Online; accessed 15-August-2021].

Dhamija, Rachna, J Doug Tygar, and Marti Hearst (2006). “Why phishing works”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 581–590.

Dilworth, John and AK Kochhar (2007). “Creation of an e-business requirements specification model”. In: *Journal of Manufacturing Technology Management*.

El-Hajj, Wassim (2012). “The most recent SSL security attacks: origins, implementation, evaluation, and suggested countermeasures”. In: *Security and Communication Networks* 5.1, pp. 113–124.

Alwin, Duane F and Brett A Beattie (2016). “The KISS principle in survey design: question length and data quality”. In: *Sociological methodology* 46.1, pp. 121–152.

Bellare, Mihir et al. (1997). “A concrete security treatment of symmetric encryption”. In: *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, pp. 394–403.

Brataas, Gunnar and Peter Hughes (2004). “Exploring architectural scalability”. In: *Proceedings of the 4th international workshop on Software and performance*, pp. 125– 129.

Bucchiarone, Antonio et al. (2018). “From monolithic to microservices: An experience report from the banking domain”. In: *Ieee Software* 35.3, pp. 50–55.

Burrows, Michael, Martin Abadi, and Roger Michael Needham (1989). “A logic of authentication”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 426.1871, pp. 233–271.

Chung, Lawrence et al. (2012). *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media.

Cohn, Mike (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Da Silva, Tiago Silva et al. (2018). “The evolution of agile UXD”. In: *Information and Software Technology* 102, pp. 1–5.

Degges, Randall (2019). *JWTs Suck*. https://speakerdeck.com/rdegges/jwts- suck. [Online; accessed 15-August-2021].

Dhamija, Rachna, J Doug Tygar, and Marti Hearst (2006). “Why phishing works”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 581–590.

Dilworth, John and AK Kochhar (2007). “Creation of an e-business requirements specification model”. In: *Journal of Manufacturing Technology Management*.

https://www.vectorstock.com