Projekt: Aplikacja “JAM” do Zdalnego Zatwierdzania Decyzji

O nas:

Jesteśmy zespołem studentów 2. roku Politechniki Wrocławskiej, którzy łączą wiedzę z różnych dziedzin, aby tworzyć nowoczesne i funkcjonalne rozwiązania technologiczne. Każdy z nas studiuje na inny kierunku, dzięki temu tworzymy zgrany zespół. Połączenie cyberbezpieczeństwa, matematyki stosowanej i informatyki technicznej pozwala nam patrzeć na problemy z różnych perspektyw.

Motywacja:

Pomysł na nasz projekt wziął się prosto z życia — jako studenci bardzo dobrze zdajemy sobie sprawę, jak cenny jest czas. Obecnie wiele procesów wymaga osobistego udziału, co często stanowi utrudnienie. Zdajemy sobie również sprawę, jak ważne jest bezpieczeństwo w dzisiejszym cyfrowym świecie. Skupiamy się na cyberbezpieczeństwie, ponieważ wiemy, że w dobie rosnących zagrożeń, ochrona danych oraz zapewnienie bezpieczeństwa procesów jest kluczowe. Nasz projekt ma na celu nie tylko usprawnienie komunikacji, ale również zagwarantowanie najwyższego poziomu ochrony danych użytkowników, co jest teraz bardziej istotne niż kiedykolwiek.

Cel projektu:

Aplikacja umożliwiająca zdalne zatwierdzanie decyzji przez powiązane ze sobą osoby. Nasze rozwiązanie znajdzie zastosowanie m.in. w zarządzaniu współwłasnością, gdzie decyzje, takie jak sprzedaż ułamkowych części nieruchomości przez syndyka, mogą być skomplikowane i czasochłonne. Wprowadzenie cyfrowego systemu zatwierdzania usprawni procesy decyzyjne i zrewolucjonizuje sposób zarządzania wspólnymi aktywami. Inne zastosowania obejmują korporacje oraz inne miejsca, gdzie konieczne jest podejmowanie wspólnych decyzji, lub zatwierdzanie decyzji.

Główne funkcjonalności:

- Możliwość wysyłania zaszyfrowanych zapytań do grupy współwłaścicieli, interesariuszy.
- Decyzje mogą być podejmowane zdalnie, eliminując potrzebę osobistego uczestnictwa u notariusza czy wysyłania listów.
- System automatycznych przypomnień dla użytkowników, którzy nie odpowiedzieli na zapytanie.
- Możliwość śledzenia aktywności użytkowników i informowania o ich nieobecności.

Bezpieczeństwo:

W celu zapewnienia jak największego poziomu bezpieczeństwa planujemy wykorzystać:

- **SIM Swap - CAMARA** – zabezpieczenie przed przejęciem konta poprzez ataki związane z duplikowaniem kart SIM.

- **Quantum Random Number Generator (QRNG)** – charakteryzują się większą entropią od ich pseudolosowych odpowiedników, dzięki czemu zapewniają prawdziwą losowość, która jest istotna w wielu aspektach bezpieczeństwa cyfrowego, takich jak generowanie haseł, identyfikatorów, tokenów oraz kluczy uwierzytelniających. Tradycyjne algorytmy pseudolosowe, takie jak te oparte na deterministycznych funkcjach, mogą w pewnych warunkach być przewidywalne, co stanowi potencjalne zagrożenie dla bezpieczeństwa. **QRNG** wykorzystują zjawiska kwantowe, takie jak zasada nieoznaczoności Heisenberga, do generowania liczb, które są prawdziwie losowe i nieprzewidywalne, co czyni je znacznie bezpieczniejszym rozwiązaniem. Poniżej zamieszczam przykładowy kod służący do generowania losowych identyfikatorów za pomocą random number API stworzonemu przez *Australian National University Quantum Numbers*.

```python
import requests
QRN_URL = "https://api.quantumnumbers.anu.edu.au/"
QRN_KEY = "replace_with_your_api_key"
params = {"length": 1, "type": "hex16", "size": 10}
headers = {"x-api-key": QRN_KEY}
response = requests.get(QRN_URL, headers=headers, params=params)
if response.status_code == 200:
    js = response.json()
    if js["success"] == True:
        print(js["data"])
    else:
        print(js["message"])
else:
    print(f"Got an unexpected status-code: {response.status_code}")
    print(response.text)
```

Przykładowy output: `['bc167b7395cdd7f947425059f23c5ab2097d82ca']`

Naszą propozycją jest stworzenie własnego API do generowania liczb losowych z wykorzystaniem metody szumów kwantowych, która zapewnia wysoką jakość losowości i jest łatwa do implementacji.

- **One Time Password (OTP) SMS** – generowane za pomocą **QRNG** do potwierdzania decyzji. Poniżej przykładowy kod generujący 6-znakowy kod uwierzytelniający.

```python
import requests
QRN_URL = "https://api.quantumnumbers.anu.edu.au/"
QRN_KEY = "replace_with_your_api_key"
params = {"length": 1, "type": "hex8", "size": 3}
headers = {"x-api-key": QRN_KEY}
```

```
response = requests.get(QRN_URL, headers=headers, params=params)
if response.status_code == 200:
    js = response.json()
    if js["success"] == True:
        print(js["data"])
    else:
        print(js["message"])
else:
    print(f"Got an unexpected status-code: {response.status_code}")
    print(response.text)
```

Przykładowy output: `['db4a4d']`

- **Number Verification** – weryfikacja numeru telefonu użytkownika dla dodatkowej warstwy bezpieczeństwa.
- **Device Reachability Status - CAMARA Sandbox**:monitorowanie aktywności użytkownika oraz wysyłanie przypomnień w przypadku braku odpowiedzi. Jeśli użytkownik jest aktywny, ale nie odpowiada, system automatycznie wysyła ponaglenie, aby przypomnieć o konieczności udzielenia odpowiedzi. Jeśli użytkownik jest nieaktywny, system generuje wiadomość zwrotną informującą, że osoba jest obecnie niedostępna. Dzięki temu, osoba wysyłająca zapytanie może wówczas podjąć kontakt z użytkownikiem inną drogą, np. telefonicznie lub za pomocą innych dostępnych kanałów komunikacji. To rozwiązanie zapewnia ciągłość komunikacji i minimalizuje ryzyko opóźnień w procesach decyzyjnych.

Grupy użytkowników:

System oparty jest na tworzeniu grup użytkowników, którzy są powiązani z daną instytucją, własnością lub wydarzeniem. Przykładowo, w przypadku syndyka w jednej grupie znajdować się będą wszyscy współwłaściciele zakupionej części nieruchomości.

Weryfikacja tożsamości:

Aby zapewnić najwyższy poziom bezpieczeństwa, planujemy wykorzystać następujące metody weryfikacji użytkowników:

- **e-Dowód lub bankowość internetowa** – Weryfikacja na wzór systemów bankowych, polegająca na przesłaniu zdjęcia dokumentu potwierdzającego tożsamość oraz porównaniu twarzy użytkownika z dokumentem. Proces weryfikacji rozpoczyna się od zeskanowania dowodu osobistego, a następnie kamera skanuje twarz użytkownika, weryfikując, czy jest zgodna z danymi zawartymi w dokumencie. W tym procesie wykorzystamy **TensorFlow** i model **FaceNet**, który pozwoli na dokładne rozpoznanie twarzy. System będzie analizować cechy twarzy, porównując je z danymi z dokumentu. Wykorzystanie technologii rozpoznawania twarzy umożliwi również detekcję prób oszustw, takich jak użycie zdjęcia. W celu ochrony danych, proces przesyłania zdjęć dowodu osobistego oraz obrazu twarzy będzie realizowany za pomocą szyfrowania end-to-end z zastosowaniem protokołów **SSL/TLS**.

Zastosowanie technologii blockchain do zapisania wyników weryfikacji, co pozwala na zapewnienie niezmienności i audytowalności całego procesu.

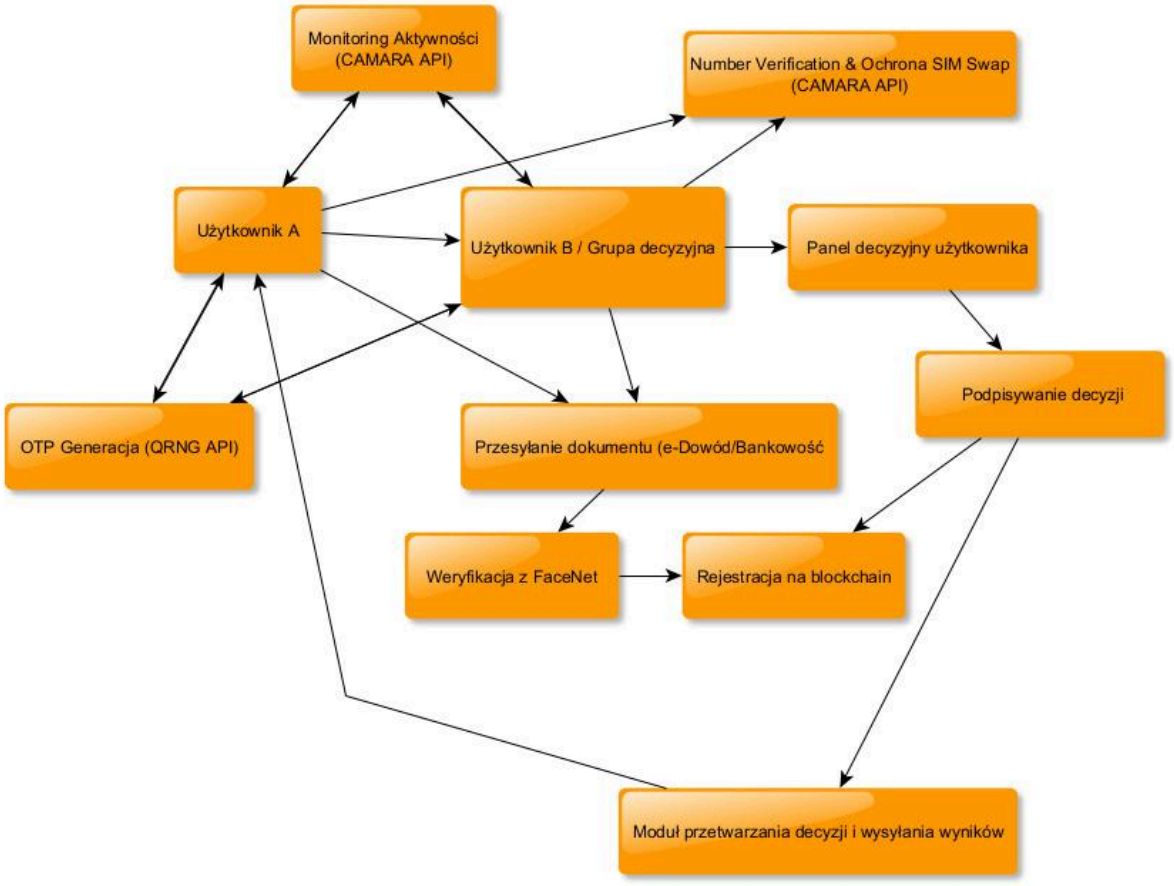
- **Aspekt notarialny** – podobna metoda weryfikacji stosowana w sytuacjach wymagających potwierdzenia tożsamości.

Szyfrowanie i transmisja danych:

- **E2EE + HTTPS** – Dzięki zastosowaniu **end-to-end encryption** operator systemu nie ma dostępu do treści przesłanych decyzji. Może jedynie **potwierdzić tożsamość nadawcy** poprzez **Number Verification** oraz mechanizm **SIM Swap**, co zabezpiecza przed atakami typu "man-in-the-middle" i przejęciem konta. Dodatkowo, transmisja odbywa się przez protokół **HTTPS**, co zapewnia dodatkową warstwę ochrony przed przechwyceniem danych.
- **Podpisywanie żądań** – Każde zapytanie jest podpisywane kryptograficznie, co uniemożliwia podszywanie się pod użytkowników i zabezpiecza przed fałszywymi żądaniami. Dzięki temu system może jednoznacznie zweryfikować autentyczność każdej operacji.

Prototyp:

W celu wizualizacji zastosowania aplikacji, stworzyliśmy prosty prototyp aplikacji na przykładzie syndyka. Aplikacje napisaliśmy w języku **Java**. W aplikacji nie ma implementacji zabezpieczeń, o których wyżej pisaliśmy. W celu lepszego pokazania, schematu działania zabezpieczeń i transmisji danych stworzyliśmy diagram zapytań:



W filmiku zamieszczonym linku, omawiamy jak działa aplikacja zarówno ze strony syndyka jak i użytkownika, kod do aplikacji dostępny jest w pliku **orange_hackathon-jam.zip**.

Podsumowanie:

Naszym celem jest zapewnienie maksymalnej wygody i bezpieczeństwa użytkowników, eliminując potrzebę osobistego uczestnictwa w wielu procesach decyzyjnych. Jako studenci, doskonale rozumiemy, jak ważne jest efektywne zarządzanie czasem oraz jak ograniczenia związane z osobistym udziałem mogą spowalniać ważne decyzje. Dlatego chcemy stworzyć rozwiązanie, które pozwoli na szybkie, bezpieczne i wygodne podejmowanie decyzji zdalnie, co będzie miało realny wpływ na usprawnienie codziennych procesów zarówno w zarządzaniu współwłasnością, jak i w innych obszarach, gdzie decyzje muszą być podejmowane wspólnie. Nasza praca ma na celu nie tylko upraszczanie tych procesów, ale także zwiększenie ich bezpieczeństwa i transparentności, co jest szczególnie ważne w dzisiejszych czasach, kiedy zaufanie do cyfrowych systemów jest kluczowe.

Orange Hackathon, Zespół “JAM”,

Janek, Adrian, Mateusz