

PSI - Projekt

27.12.2023 r.

Zespół 32, skład:

Jakub Proboszcz

Paweł Kochański

Krzysztof Pałucki

Kamil Michalak

Założenia

Podstawowe funkcje:

- Węzeł nadzorca umożliwia pozyskiwanie listy węzłów pośredniczących.
- Nawiązywanie połączeń z serwerem za pośrednictwem węzła pośredniczącego.
- Ruch od klienta do węzła pośredniczącego jest szyfrowany.

Ruch od węzła pośredniczącego do serwera nie jest (dodatkowo) szyfrowany - dzięki temu serwer nie wymaga żadnego specjalnego oprogramowania. Jeżeli tunelowane połączenie klient-serwer jest szyfrowane, to ten ruch też będzie szyfrowany.

Anonimizacja ruchu:

- Węzeł pośredniczący wprowadza pseudolosowe opóźnienia w przesyłanym strumieniu.
- Węzeł pośredniczący zmienia rozmiar i segmentację przesyłanego ruchu.

To pozwala zlikwidować podobieństwa między ruchem od klienta do węzła pośredniczącego, a ruchem od węzła pośredniczącego do serwera, zapewniając anonimowość klienta przy wielu obsługiwanych naraz połączeniach.

Zakres realizacji

1. Implementacja węzła centralnego - nadzorcy:

- aplikacja w języku Python
- możliwość zapytania o listę adresów IP węzłów pośredniczących ze strony klienta
- zarządzanie węzłami pośrednimi - możliwość rejestrowania i wyrejestrowania węzłów pośrednich

2. Implementacja węzłów pośrednich:

- aplikacja w języku Python
- tunelowanie ruchu TCP (za pomocą IPSec)
- losowe opóźnianie segmentów
- losowa segmentacja danych

3. Implementacja klienta:

- biblioteka dla języka Python dla aplikacji klienckich
- wybór węzła pośredniczącego na podstawie pozyskanej listy
- komunikacja z serwerem za pośrednictwem węzła pośredniczącego

Przypadki użycia

Anonimowa łączność:

- Klient wysyła żądanie do nadzorcy.
- Nadzorca przesyła klientowi listę adresów IP węzłów pośredniczących.
- Klient łączy się z węzłem.
- Węzeł pośredniczący nawiązuje połączenie z docelową lokalizacją.

Architektura

Przedstawiliśmy architekturę naszego systemu w postaci diagramów C4.

Diagram kontekstu systemu:

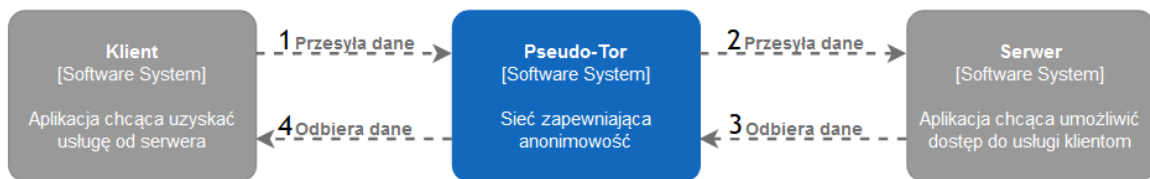
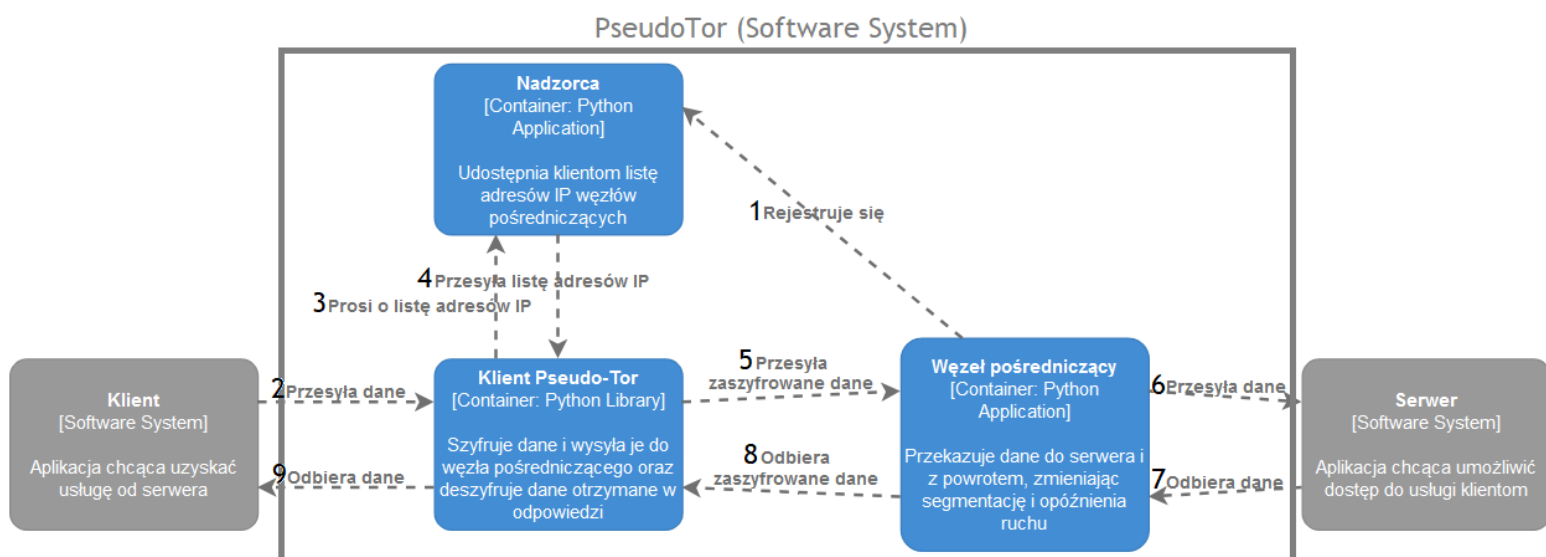


Diagram kontenerów:



Dodatkowo załączamy też diagram sekwencji pokazujący komunikację trzech kontenerów.

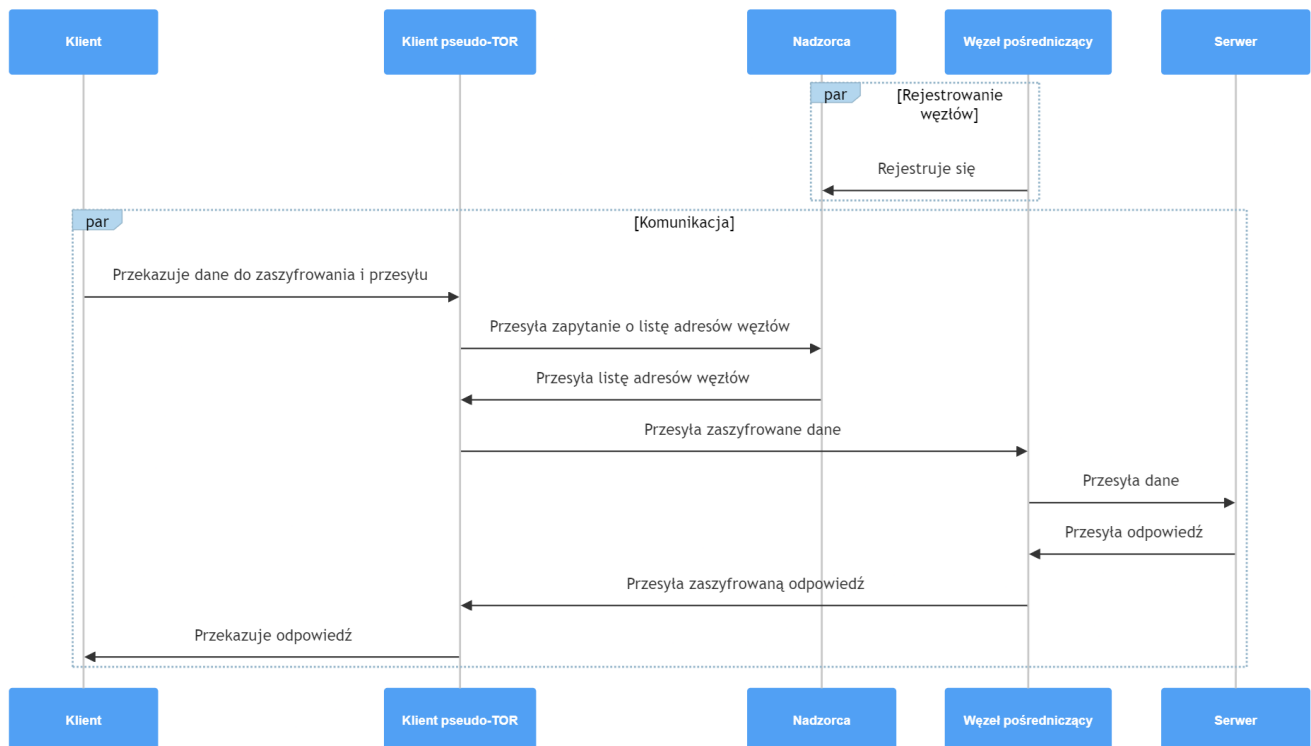


Diagram komponentów (dla nadzorcy):

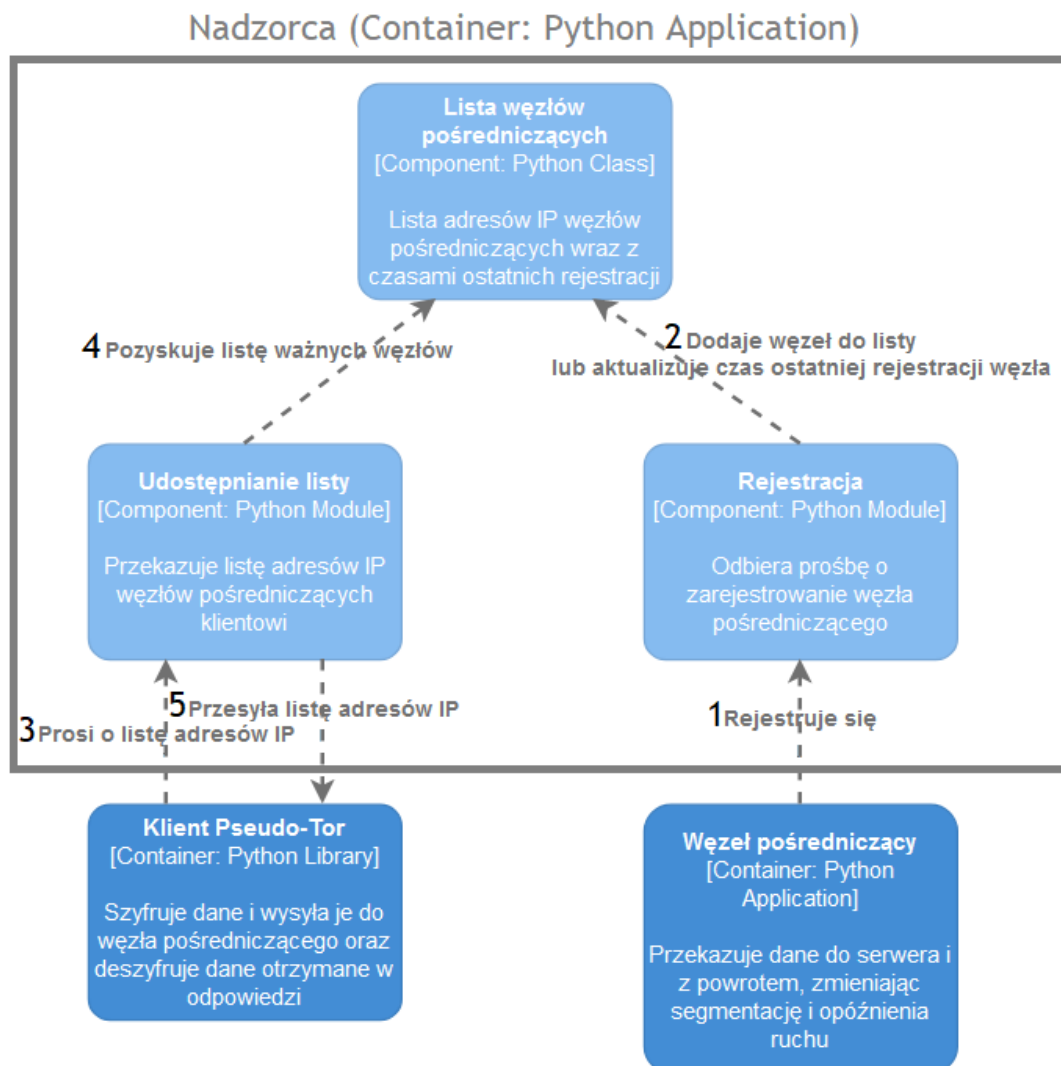


Diagram komponentów (dla klienta w naszej sieci - PseudoTor):

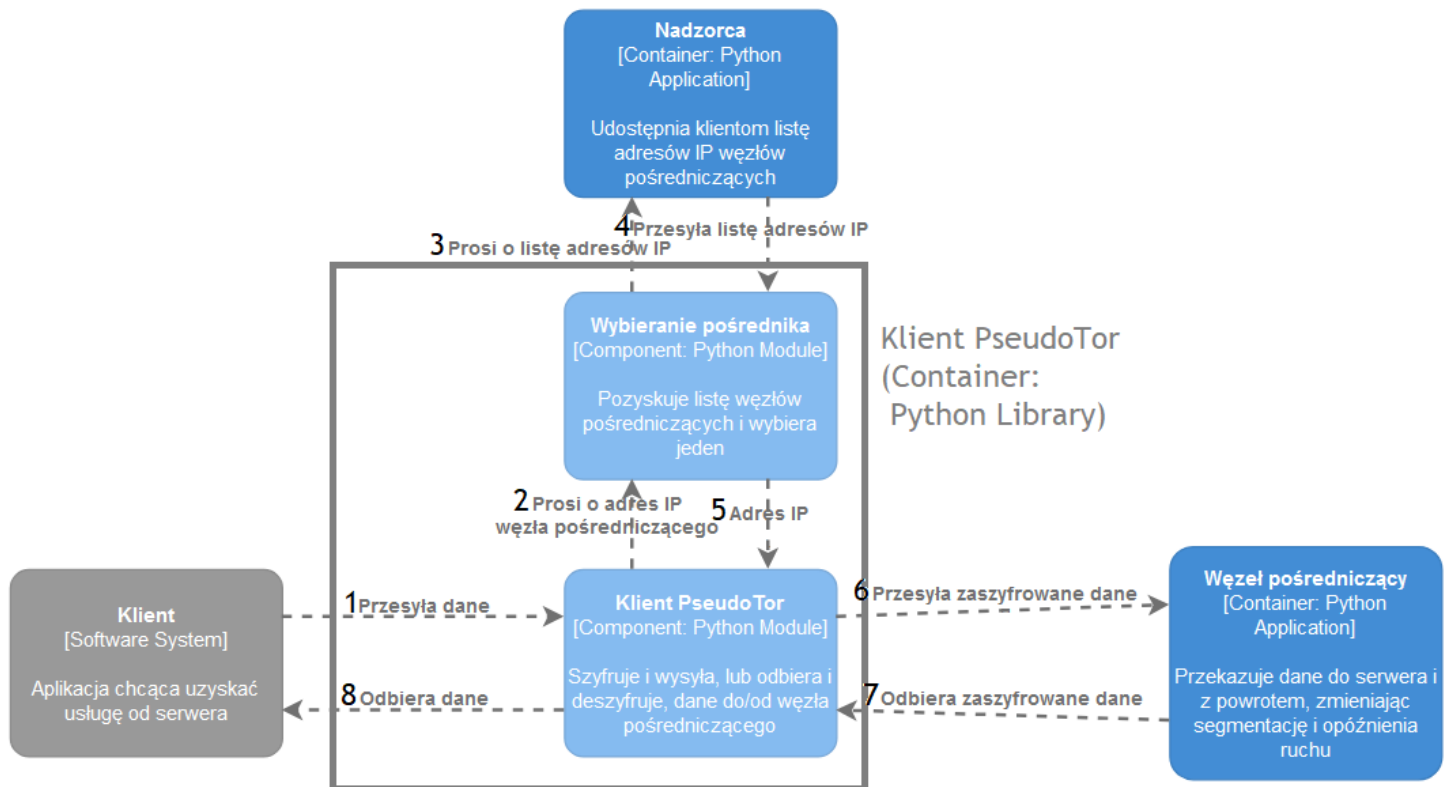
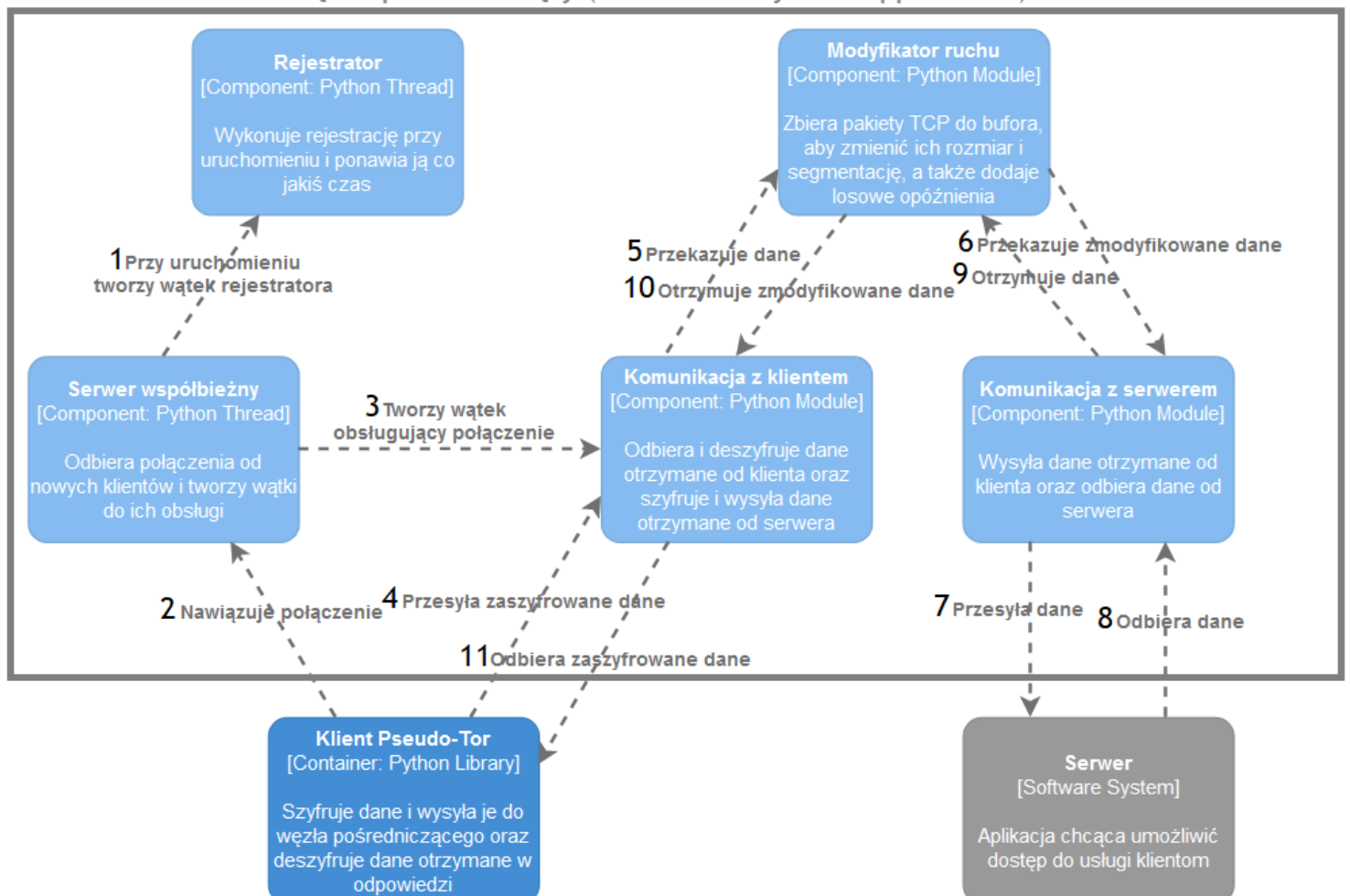


Diagram komponentów węzła pośredniczącego:

Węzeł pośredniczący (Container: Python Application)



Diagramów klas nie zamieszczamy, ponieważ na tym etapie projektu nie jesteśmy w stanie bardziej szczegółowo określić klas, które powstaną w naszym projekcie.

Najprawdopodobniej każdemu komponentowi będzie odpowiadać co najmniej jedna klasa.

Obsługa sytuacji błędnych

- Węzeł pośredniczący doznał awarii => Nie wyśle informacji o byciu dalej w stanie dostępnym do upływu timeout'u, więc nadzorca go wyrejestruje z listy dostępnych węzłów
- Węzeł pośredniczący doznał awarii po wyznaczeniu go jako pośrednika => Po timeout'cie pytamy nadzorcę o listę, po czym ponawiamy wysłanie danych, wybierając inny węzeł z listy (powtarzane do skutku)
- Nadzorca niedostępny (u klienta) => Sygnalizujemy wystąpienie błędu w aplikacji klienckiej
- Nadzorca niedostępny (u węzła) => Węzeł ponawia próbę rejestracji po timeout'cie
- Serwer niedostępny => Sygnalizujemy wystąpienie błędu w aplikacji klienckiej

Przypadki testowe

- Wysłanie wiadomości mieszczącej się w pojedynczym pakiecie do klienta i uzyskanie odpowiedzi
- Komunikacja wielopakietowa między klientem, a odbiorcą
- Komunikacja wielopakietowa między wieloma parami klient-odbiorca jednocześnie
- Komunikacja wielopakietowa między klientem, a odbiorcą z opóźnieniami
- Komunikacja wielopakietowa między klientem, a odbiorcą z gubieniem pakietów w sieci
- Komunikacja wielopakietowa między klientem, a odbiorcą z awarią węzła
- Komunikacja wielopakietowa między klientem, a odbiorcą z awarią nadzorcy

Podział prac w zespole

- Paweł Kochański - biblioteka kliencka
- Kamil Michałak - węzeł pośredniczący - zmiana segmentacji oraz dodanie pseudolosowych opóźnień do segmentów, komunikacja z prowadzącym
- Krzysztof Pałucki - węzeł pośredniczący - komunikacja z nadzorcą, klientem i serwerem
- Jakub Proboszcz - węzeł centralny (nadzorca)

Scenariusz demonstracji

1. **Prezentacja interfejsu użytkownika:**
 - Klient łączy się z węzłem pośredniczącym.
2. **Demonstracja tunelowania:**
 - Przesyłanie danych między klientem a węzłem pośredniczącym.
3. **Wprowadzanie opóźnień:**
 - Pokazanie wprowadzania pseudolosowych opóźnień
4. **Zmiana rozmiaru/segmentacji:**
 - Demonstracja zmiany rozmiaru i segmentacji przesyłanego ruchu
5. **Sytuacje błędne:**
 - Symulacja sytuacji błędnych i prezentacja obsługi awarii.

Parametryzacja węzła pośredniczącego

Węzeł pośredniczący będzie przyjmował następujące argumenty wywołania:

- maksymalny czas wprowadzanych pseudolosowych opóźnień w milisekundach (rozmiar wprowadzanego opóźnienia będzie losowany z rozkładem jednostajnym z przedziału od 0 do podanej liczby milisekund);
- opcjonalna flaga --no-segmentation-change (-ns) - jej obecność wyłącza funkcjonalność zmiany segmentacji tunelowanych pakietów;
- dwie liczby określające rozmiar pakietów po zmianie segmentacji; faktyczny ich rozmiar będzie losowany z rozkładu jednostajnego o podanych końcach. Jeżeli podany przedział będzie dopuszczał wartości większe niż maksymalny rozmiar segmentu TCP, dla którego nie zachodzi fragmentacja, to zostanie on odpowiednio zmniejszony. Jeżeli podana będzie flaga -ns, ten argument nie powinien być obecny.