Autorzy: Daniel Furgał, Wojciech Stępniak, Elżbieta Żukrowska

Ocena architektury grupy: Leszek Stasiak, Paweł Szmit

1. Przegląd rozwiązań architektonicznych

Na bazie dostarczonej dokumentacji oraz krótkiej rozmowy dostrzeżono następujące rozwiązania architektoniczne:

- Dwuetapowe uwierzytelnianie: login, hasło + SMS.
- Microsoft Identity zapewnia autentykację i autoryzację użytkowników do aplikacji
- Szyfrowanie całego ruchu sieciowego pomiędzy przeglądarką a serwerem IIS.
- Część komunikacji z serwerem zostanie zrealizowana asynchronicznie (REST)
- Wykorzystanie mechanizmu cachowania danych
- Architektura trójwarstwowa

2. Drzewo użyteczności

Charakterystyka jakościowa	**Podcharakterystyka jakościowa**	**Scenariusze z powiązanymi priorytetami (importance, difficulty)**
Wydajność	Opóźnienie	S1: Średni czas odpowiedzi 5s przy obciążeniu 200 użytkowników (H, H)
Modyfikowalność	Przenaszalność	S2: Czas poprawnie wykonanej migracji na nowy serwer < 2h. (M, M)
	Rozszerzalność	S3: Dodanie nowego modułu funkcjonalnego do systemu (M, H)
Bezpieczeństwo	Poufność	S4: Uniemożliwienie podsłuchania danych wrażliwych (H, M)
		S5: Kontrola dostępu użytkowników do poszczególnych zasobów systemu (H, M)
	Integralność danych	S6: Zapewnienie aktualności i spójności danych w systemie po każdej transakcji (H,M)
Dostępność	Awaria serwera	S7: Przywrócenie aplikacji po awarii do działania w czasie < 5min (M,M)
	Konserwacja	S8: Czas niedostępności wynikający z wdrożenia nowego modułu funkcjonalnego do systemu < 1 h. (M, M)
Testowalność	Testy integracyjne	S9: Możliwość wykonania automatycznych testów integracyjnych w czasie poniżej 3 godzin (M, M)

3. Analiza wybranych scenariuszy

Scenariusz: S1:	Średni czas odpowiedzi 5s przy obciążeniu 200 użytkowników			
Atrybuty:	Opóźnienie			
Środowisko:	Obciążenie przez 200 użytkowników			
Bodziec:	Użytkownik oczekuje płynnej interakcji z systemem			
Odpowiedź:	Czas realizacji zleconego działania i zwrócenie odpowiedzi przez aplikacje w średnim czasie krótszym niż 5s, dla próbki 100 wywołań przez użytkownika.			
Decyzje architektoniczne	Ryzyko	Wrażliwość	Kompromis	Brak ryzyka
Cache danych statycznych				NR1
Pojedyncza baza danych			T1	NR3
Szyfrowanie całej transmisji danych			T2	
Asynchroniczna komunikacja przy użyciu AJAX				NR4
Brak decyzji o lokalizacji serwerów	R1			
Analiza:	Na opóźnienie systemu dostrzegalne przez użytkownika będą miały wpływ: opóźnienie sieciowe związane z lokalizacją i infrastrukturą sieci, a także zastosowane protokoły, obciążenie całego serwera IIS, obciążenie systemu bazy danych, szyfrowanie transmisji danych.			
Diagram architektoniczny	Brak			

Scenariusz: S4:	Uniemożliwienie podsłuchania danych wrażliwych			
Atrybuty:	Poufność			
Środowisko:	Warunki normalnego użytkowania systemu.			
Bodziec:	Chęć korzystania z funkcjonalności systemu dostępnej po zalogowaniu użytkownika, w sposób nie informujący osób trzecich o działaniu użytkownika, ani systemu.			
Odpowiedź:	Trudność, granicząca z niemożliwością, w uzyskaniu wrażliwych danych systemu przez osoby trzecie.			
Decyzje architektoniczne	Ryzyko	Wrażliwość	Kompromis	Brak ryzyka
Szyfrowanie całej transmisji danych			T2	
Długość klucza	R2	S1		
Brak decyzji o szyfrowaniu danych między bazą danych a serwerem aplikacyjnym IIS	R3			
Wykorzystanie mechanizmu ASP.NET Identity			T4	
Analiza:	Do nieumożliwienia podsłuchania danych wrażliwych przyczyni się: szyfrowanie danych, mechanizmy autoryzacji i autentykacji, długość klucza, bezpieczeństwo komunikacji z bazą danych.			
Diagram architektoniczny	Diagram rozmieszczenia			

Scenariusz: S5:	Kontrola dostępu użytkowników do poszczególnych zasobów systemu			
Atrybuty:	Poufność			
Środowisko:	W czasie normalnego użytkowania			
Bodziec:	Potrzeba zabezpieczenia systemu, mająca uchronić przed nieuprawnionym dostępem			
Odpowiedź:	Dostęp użytkowników jedynie do przeznaczonych dla nich zasobów			
Decyzje architektoniczne	Ryzyko	Wrażliwość	Kompromis	Brak ryzyka
Dwuetapowe uwierzytelnianie			T3	
Wykorzystanie mechanizmu autentykacji ASP.NET Identity			T4	
Szyfrowanie procedury logowania				NR2
Analiza:	Dwuetapowe uwierzytelnianie za pomocą usługi ASP.NET Identity pozwala na uzyskanie wymaganego poziomu bezpieczeństwa			
Diagram architektoniczny	Brak			

4. Punkty wrażliwości i kompromisy

S1 – Zbyt krótki klucz umożliwi złamanie jego bezpieczeństwa, a co za tym idzie umożliwienie odszyfrowania danych w krótkim czasie (np. kilka dni/tygodni/miesięcy) na przeciętnej maszynie. Zbyt długi klucz wpływa z kolei negatywnie na wydajność, przede wszystkim deszyfrowania, ale też szyfrowania.

T1 – Pojedyncza baza danych negatywnie wpływa na wydajność oraz skalowalność rozwiązania. Z drugiej strony ułatwia modyfikowalność systemu. Korzystając z pojedynczej bazy danych łatwiej implementować nowe moduły. Przeniesienie aplikacji także jest łatwiejsze gdyż nie ma konieczności konfiguracji wielu serwerów baz danych.

T2 – Szyfrowanie całej transmisji pomiędzy przeglądarką, a klientem zapewni poufność informacji kluczowych. Szyfrowanie danych niewrażliwych z kolei przekłada się na spadek wydajności aplikacji związany z szyfrowaniem danych.

T3 – Dwuetapowe uwierzytelnianie zablokuje dostęp osobom, które nielegalnie weszły w posiadanie loginu i hasła. Spowoduje to jednak utrudnienie w przypadku potrzeby posiadania jednego konta dla paru osób, ponieważ dostęp będzie silnie powiązany z nr telefonu. Wysyłanie smsów generuje też dodatkowe koszty w utrzymaniu systemu. Także przypadek, gdy telefon użytkownika zostanie zgubiony lub rozładowany może okazać się problematyczny.

T4 - Ewentualne wykrycie dziur bezpieczeństwa w ASP.NET Identity przekłada się na bezpieczeństwo systemu, ale sam mechanizm dostarcza szereg gotowych, powszechnie używanych i sprawdzonych rozwiązań w zakresie autentykacji i autoryzacji użytkowników co przekłada się na utrzymywalność systemu.

5. Ryzyka i nie ryzyka

R1 – Lokalizacja serwera może mieć wpływ na opóźnienie komunikacji sieciowej. Mechanizm skalowania okna TCP zwiększa liczbę bajtów przesyłanych w jednym oknie. Wielokrotna sygnalizacja przez nadawcę zmiany długości okna może mieć wpływ na opóźnienie transmisji, co przełoży się na całościowe opóźnienie pracy z systemem. Nawet jeśli usługi Windows Azure umożliwiają wybór lokalizacji serwera, wybór ten jest mocno ograniczony i nie wzięto tego czynnika pod uwagę przy projektowaniu architektury.

R2 – Nie podano długości klucza z jakim będzie szyfrowana komunikacja pomiędzy klientem a serwerem.

R3 – Brak informacji o szyfrowaniu połączenia bazy danych MSSQL z serwerem IIS. Brak tej decyzji może umożliwić dostęp do danych wrażliwych przez osoby nieuprawnione

NR1 – Cache danych statycznych (np. szablonów html, plików css czy obrazków) jest korzystną decyzją podjętą ze względu na wydajność. Nawet w przypadku zmiany szaty graficznej nie przeszkodzi w użytkowaniu aplikacji.

NR2 – Szyfrowanie pomiędzy przeglądarką a serwerem uniemożliwi podsłuchanie loginu i hasła użytkownika systemu.

NR3 – Przy czasie średnim czasie wykonania zapytania 10 ms oraz obciążeniu 200 użytkowników czas odpowiedzi systemu mieści się w wyznaczonej normie.

NR4 – Zastosowanie AJAX do częściowej interakcji z systemem zmniejszy ilość przesyłanych danych poprzez sieć, a także odciąży serwer oraz klienta od ponownego generowania i renderowanie całej strony (widoku) od nowa. Wszystkie powszechnie używane przeglądarki obsługują w wystarczającym stopniu JavaScript i asynchroniczną komunikację AJAX.

6. Wnioski

Kluczowe problemy znalezione podczas analizy architektury:

- Brak podjętych decyzji dotyczących długości klucza dla komunikacji między klientem a serwerem może mieć wpływ na bezpieczeństwo całego systemu.
- Dla dostępnej dokumentacji przedstawionych rozwiązań na platformie Windows Azure, brakuje kluczowych informacji dotyczących bezpieczeństwa. Między innymi brakuje informacji o szyfrowaniu połączenia z bazą danych MSSQL. Pomimo faktu, że baza znajduję się na tym samym serwerze do aplikacji, to z diagramu rozmieszczenia nie wynika to bezpośrednio. Przy przeniesieniu aplikacji na inny hosting, fakt ten może zostać pominięty.
- Nie została podjęta decyzja dotycząca lokalizacji serwerów aplikacji co może przełożyć się na opóźnienie w komunikacji z serwerem.