

# Analiza projektu

Krzysztof Smogór

Piotr Widomski

## 1 Streszczenie

Celem projektu jest stworzenie systemu do zdalnej pracy opartego na architekturze rozproszonej. System będzie umożliwiać tworzenie, konfigurację i zarządzanie maszynami wirtualnymi. Użytkownik będzie mógł uzyskać działającą maszynę wirtualną i pracować na niej przy pomocy protokołu zdalnego pulpitu (RDP). Maszyny wirtualne mogą używać samego procesora lub procesora z bezpośrednim dostępem do GPU.

## 2 Słownik pojęć

- Aplikacja kliencka - aplikacja uruchamiana na komputerze użytkownika, która umożliwi komunikację z systemem oraz uruchomienie zewnętrznego programu implementującego protokół RDP.
- Aplikacja nadzorczą (Nadzorca) - aplikacja, która przetwarza zapytania od aplikacji klienckiej oraz komunikuje się ze wszystkimi serwerami wirtualizacji. Na podstawie tych informacji buduje model zajętości każdego z serwerów wirtualizacji oraz decyduje kiedy trzeba, i na którym serwerze, uruchomić nowe maszyny wirtualne i do której wirtualnej maszyny ma podłączyć się użytkownik proszący o utworzenie sesji.
- Serwer wirtualizacji - komputer, który udostępnia swoje zasoby (CPU, GPU, pamięć, przestrzeń dyskową) w postaci uruchamianych na nim maszyn wirtualnych. Dodatkowo na tym komputerze będzie uruchomiona aplikacja, która będzie odpowiadać na pytania aplikacji nadzorczej oraz wykonywać operacje na maszynach wirtualnych (uruchamianie i wyłączanie), która dalej będzie utożsamiana z serwerem wirtualizacji.
- Maszyna wirtualna CPU - jest to maszyna wirtualna, która udostępnia użytkownikowi podstawowe zasoby (procesor, pamięć i przestrzeń dyskową) przeznaczona raczej do pracy biurowej. Uruchamiana jest na pewnym serwerze wirtualizacji z liczbą zasobów zdefiniowaną wcześniej w konfiguracji.

- Maszyna wirtualna GPU - maszyna analogiczna do maszyny wirtualnej CPU tyle, że ma do dyspozycji przekazaną przez mechanizm GPU Passthrough kartę graficzną podłączoną do serwera wirtualizacji.
- RDP - protokół zdalnego dostępu do pulpitu od firmy Microsoft.
- Sesja - jest to określenie jednorazowego dostępu do systemu przez użytkownika. Utworzenie sesji wiąże się z przypisaniem do użytkownika konkretnej maszyny wirtualnej, na której będzie pracować. Sesja kończy się w przypadku, gdy użytkownik poinformuje system o zakończeniu pracy lub gdy minie czas oczekiwania na odzyskanie połączenia jego utracie.
- Vagrant-box<sup>1</sup> - jest to przygotowany wcześniej obraz maszyny wirtualnej, któremu można zmieniać dostępne zasoby. Uruchamiają się bardzo powtarzalnie w środowisku programu Vagrant
- Ansible playbook<sup>2</sup> - jest to pewien rodzaj skryptu konfiguracyjnego dla systemu operacyjnego, który można parametryzować i wykonywać przy starcie Vagrant-boxa.
- Panel administratora - jest to aplikacja internetowa, na której administrator może sprawdzić jakie serwery wirtualizacji znajdują się w systemie oraz stan ich zasobów (wolne, zajęte oraz całkowite).
- Konto użytkownika - jest to profil użytkownika w systemie, do którego ma dostęp na każdej maszynie wirtualnej. Używając przygotowanych wcześniej danych logowania może za ich pomocą logować się do maszyn wirtualnych. Będą one przechowywane w zewnętrznym (poza opisanym systemem) systemie katalogowym.
- Katalog użytkownika - jest to prywatny folder dostępny dla użytkownika na każdej maszynie wirtualnej. Przechowywany będzie na zewnętrznym (poza opisanym systemem) dysku sieciowym.
- Konfiguracja stała - jest to konfiguracja maszyny wirtualnej, która nie zmienia się w zależności od miejsca uruchomienia. Docelowo ta konfiguracja ma być zapisana w Vagrant-boxie. W razie potrzeby można ją także zdefiniować w odpowiednim Ansible playbooku.
- Konfiguracja zmienna - jest to konfiguracja wirtualnej maszyny, która zmienia się w zależności od miejsca uruchomienia. Jest definiowana w odpowiednim Ansible playbooku uruchamianym przy każdym włączeniu maszyny.

---

<sup>1</sup>Dokumentacja i opis na stronie Vagranta

<sup>2</sup>Dokumentacja i opis na stronie Ansible'a

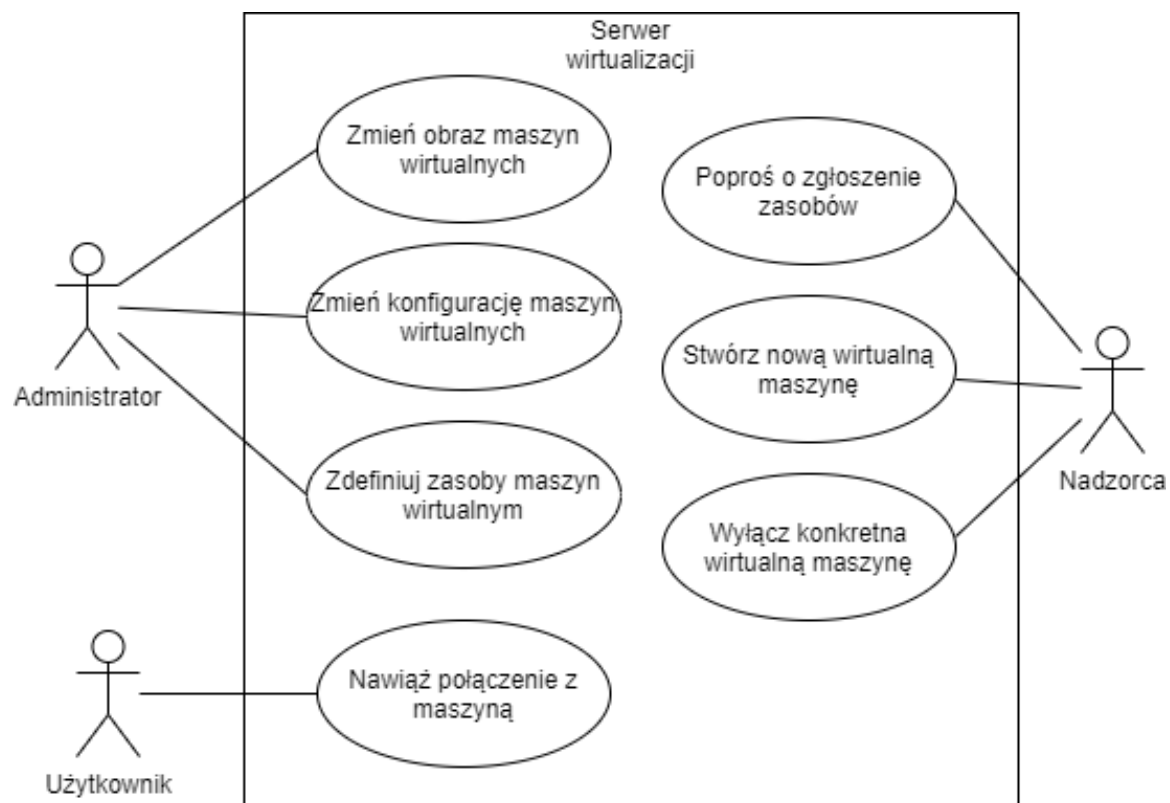
### 3 Wymaganie funkcjonalne

#### 3.1 Nadzorca



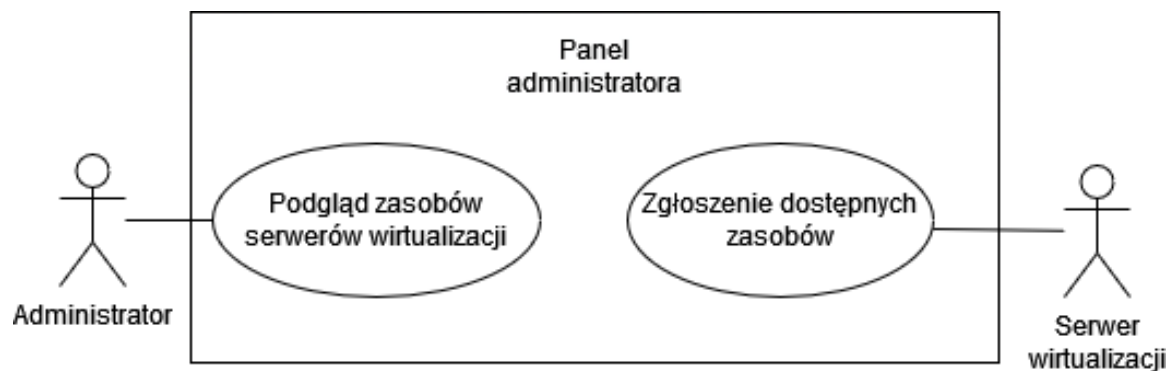
Aktor	Nazwa	Opis	Odpowiedź systemu
Użytkownik	Uzyskanie sesji do pracy	Uzyskanie sesji do pracy na maszynie wirtualnej CPU lub GPU	Do użytkownika zostaje przydzielona maszyna wirtualna oraz zestawione połączenie RDP. W przypadku, gdy utracił on połączenie, to przydzielana jest do niego poprzednio używana maszyna, jeżeli jego sesja nie została jeszcze umorzona.
	Poznanie ilości dostępnych maszyn	Wyświetlanie szacowanej ilości dostępnych maszyn każdego typu	Użytkownikowi zostaje wyświetlona szacowana liczba dostępnych maszyn obliczona na podstawie informacji o dostępnych zasobach każdego z serwerów wirtualizacji
Serwer wirtualizacji	Zgłoszenie dostępnych zasobów	Serwer zgłasza nadzorcy dostępne zasoby	Nadzorca wykorzystuje zgłoszone zasoby do wyliczania szacowanej liczby dostępnych maszyn oraz do balansowania obciążenia serwerów wirtualizacji

## 3.2 Serwer wirtualizacji



Aktor	Nazwa	Opis	Odpowiedź systemu
Użytkownik	Nawiązanie połączenia z maszyną	Użytkownik nawiązuje połączenie z maszyną wirtualną	Maszyna wirtualna zostaje zajęta przez użytkownika; serwer wirtualizacji rozpoczyna monitorowanie, czy sesja wciąż trwa
Nadzorca	Poproś o zgłoszenie zasobów	Nadzorca wysyła do wszystkich serwerów wirtualizacji prośbę o zgłoszenie swoich używanych i wolnych zasobów	Serwer wirtualizacji informuje nadzorcę o stanie swoich zasobów
	Stwórz nową wirtualną maszynę	Nadzorca prosi serwer wirtualizacji o stworzenie nowej wirtualnej maszyny dla danego użytkownika na wybranym typie maszyny	Serwer wirtualizacji tworzy wirtualną maszynę i udostępnia możliwość połączenia się z nią
	Wyłącz konkretną wirtualną maszynę	Nadzorca prosi serwer wirtualizacji aby wyłączył konkretną wirtualną maszynę.	Serwer wirtualizacji wyłącza konkretną wirtualną maszynę oraz pilnuje aby na pewno się wyłączyła.
Administrator	Zmień obraz maszyn wirtualnych	Zmiana obrazu źródłowego maszyn wirtualnych	Zdefiniowany przez administratora vagrant-box jest używany przez serwery wirtualizacji
	Zmień konfigurację maszyn wirtualnych	Zmiana zmiennej konfiguracji maszyn wirtualnych	Zmodyfikowany ansible playbook jest używany przez serwery wirtualizacji
	Zdefiniuj zasoby maszyn wirtualnych	Zmiana ilości zasobów przydzielanych na każdy z typów maszyn wirtualnych oraz łączną ilość zasobów przeznaczonych na maszyny	Zmodyfikowana konfiguracja zasobów będzie wykorzystywana przez serwer wirtualizacji przy kolejnym uruchomieniu

### 3.3 Panel administratora



Aktor	Nazwa	Opis	Odpowiedź systemu
Administrator	Podgląd zasobów serwerów wirtualizacji	Wyświetlanie wolnych oraz zajętych zasobów serwerów wirtualizacji	Wyświetlenie zasobów poszczególnych serwerów wirtualizacji, liczby zajętych maszyn oraz szacowanej liczby wolnych maszyn
Serwer wirtualizacji	Zgłoszenie dostępnych zasobów	Serwer zgłasza panelowi administratora dostępne zasoby	Panel administratora wykorzystuje zgłoszone zasoby do wyliczania szacowanej liczby dostępnych maszyn oraz wyświetlania zasobów poszczególnych serwerów wirtualizacji

## 4 Wymaganie нефункционалне

Grupa wymagań	Nr wymagania	Opis
Użytkowanie (Usability)	1	Aplikacja kliencka ma działać na systemach operacyjnych GNU/Linux oraz MS Windows
	2	Aplikacja kliencka musi udostępniać możliwość użycia własnego klienta RDP do nawiązania połączenia z maszyną wirtualną
	3	Maszyny wirtualne muszą mieć dostęp do systemu przechowującego konta użytkowników wraz z ich katalogami domowymi
Niezawodność (Reliability)	4	System musi być odporny na awarie poszczególnych serwerów wirtualizacji i kontynuować działanie w sposób niezauważalny dla użytkowników nie używających danego serwera.
	5	Awaria nadzorcy może spowodować uniemożliwienie rozpoczęcia nowych sesji, ale nie może przerwać istniejących sesji
Wydażność (Performance)	6	Łącznie zużywane zasoby przez maszyny wirtualne na poszczególnym serwerze wirtualizacji nie mogą przekroczyć wcześniej zdefiniowanych limitów
	7	Nadzorca musi balansować obciążenie serwerów wirtualizacji
	8	W systemie zawsze musi istnieć jedna działająca maszyna wirtualna nie połączona z żadną sesją, aby można było ją szybko przydzielić użytkownikowi
	9	Zwolnione maszyny wirtualne, które nie są wykorzystywane jako zapas, muszą być wyłączane
Utrzymanie (Supportability)	10	Możliwe jest działanie więcej niż jednego nadzorcy w systemie, w celu zwiększenia dostępności lub przeprowadzenia prac utrzymaniowych

## 5 Analiza ryzyka

<b>Mocne strony</b> <ol style="list-style-type: none"><li>1. Łatwa skalowalność pod względem liczby sesji w systemie</li><li>2. Wiele rozwiązań Open Source</li><li>3. Elastyczność pod względem konfiguracji</li><li>4. Tańsze rozwiązanie niż kupno stacji roboczych</li></ol>	<b>Słabości</b> <ol style="list-style-type: none"><li>1. System trudny w konfiguracji</li><li>2. Potrzeba wymiany sprzętu komputerowego</li><li>3. Krótki czas rozwoju systemu</li><li>4. Ograniczenie doświadczenie twórców systemu</li><li>5. Małe prawdopodobieństwo wsparcia projektu po zakończeniu prac</li></ol>
<b>Okazje</b> <ol style="list-style-type: none"><li>1. Grupa docelowa to firmy z dużą ilością stacji roboczych</li><li>2. Zwiększenie zapotrzebowania na prace zdalną na rynku pracy</li></ol>	<b>Zagrożenia</b> <ol style="list-style-type: none"><li>1. Istnienie konkurencji ugruntowanej na rynku</li><li>2. System w dużej mierze oparty o oprogramowanie rozwijane przez inne organizacje</li></ol>

### 5.1 Omówienie zagrożeń

1. System trudny w konfiguracji - wysoko prawdopodobne  
Można temu zaradzić poprzez udostępnienie dokładnej dokumentacji lub ścisłą współpracę z klientem przy wdrażaniu systemu.  
Wartość: duża
2. Potrzeba wymiany sprzętu komputerowego - średnio prawdopodobne  
Klient może potrzebować wymienić aktualne stacje robocze na terminale oraz zainwestować w sprzęt serwerowy. Jednak gdy klientami będą firmy, które mają dużo pracowników pracujących spoza firmy lub dopiero tych pracowników pozyskują, to kupno terminali i serwerów powinno być bardziej zachęcające niż kupno stacji roboczych.  
Wartość: średnia.
3. Krótki czas rozwoju systemu - wysoko prawdopodobne  
Czas rozwoju systemu jest bardzo ograniczony. Aby pomimo tego ograniczenia działał on w sposób akceptowalny powinniśmy skupić się na dobrym przedyskutowaniu i opisanu kluczowych modułów systemu. W czasie projektu należy pilnować aby nie dodawać nadmiarowych funkcjonalności do systemu. W czasie implementacji krytyczne będzie dokładne zaplanowanie aplikacji pod kątem testowania automatycznego. Ułatwi to wyłapywanie prostych błędów jeszcze we wczesnej fazie projektu.  
Wartość: wysoka
4. Ograniczone doświadczenie twórców systemu - pewne



Tutaj jedynym sposobem na ograniczenie ryzyka jest rozważna implementacja.

Wartość: średnia

5. Małe prawdopodobieństwo wsparcia projektu po zakończeniu prac - wysoko prawdopodobne

Trudno teraz przewidzieć co się stanie z projektem po zakończeniu prac. Jednak prawdopodobnie twórcy systemu zajmą się innymi projektami. Można jedynie dokładnie komentować kod i pokrywać jak najwięcej jego części testami. Wtedy inne osoby będą w stanie szukać błędów albo próbować w taki sposób uzupełnić brakującą wiedzę o systemie.

Wartość: niska

6. Istnienie konkurencji ugruntowanej na rynku - bardzo prawdopodobne

Konkurencyjne systemy oferujące podobne rozwiązania są już dobrze ugruntowane na rynku i przetestowane. Nasz system może spróbować konkurować jedynie z nimi ceną implementacji oraz elastycznością.

Wartość: średnia

7. System w dużej mierze oparty o oprogramowanie rozwijane przez inne organizacje - nisko prawdopodobne

W czasie życia systemu mogą pojawić się błędy w oprogramowaniu nie rozwijanym w ramach naszego systemu. naprawa takich błędów może trwać bardzo długo. Pewnym sposobem wsparcia takiego systemu jest własnoręczne poprawianie błędów w zewnętrznym oprogramowaniu i zgłaszanie ich do odpowiedniej organizacji. Do czasu zastosowania poprawki jest możliwość korzystania z wersji, na którą nanieśliśmy własną poprawkę.

Wartość: wysoka

## 6 Harmonogram projektu

