

0.1. Końcowy stan systemu

System, w swojej ostatecznej postaci, spełnia wszystkie postawione mu wymagania. Mimo trudności w implementacji niektórych funkcjonalności, w szczególności zarządzania maszynami wirtualnymi z poziomu kodu `C#`, ostateczny projekt działa zgodnie z oczekiwaniami. Projekt tworzony w ramach tej pracy uznajemy zatem za udany.

Elementem, który mógł zostać lepiej wykonany jest zarządzanie maszynami wirtualnymi przy pomocy Vagranta. Z powodu braku interfejsu programistycznego udostępnianego przez ten program, wykorzystywany jest on w naszym systemie za pośrednictwem wywołań poleceń powłoki. Nie jest to najlepsze rozwiązanie, o czym świadczy liczba błędów, które wynikły podczas tworzenia systemu. Jest to w naszym przekonaniu najbardziej podatny na błędy element systemu. Taki sposób rozwiązania jest spowodowany chęcią skorzystania z Vagranta podczas projektowania systemu. Na tamtym etapie nie sprawdziliśmy, czy dostępne są narzędzia umożliwiające korzystanie z niego z poziomu kodu. W momencie odkrycia braku takich narzędzi, było już za późno na zastąpienie Vagranta innym rozwiązaniem.

Projektując komunikację wewnętrzną za pośrednictwem RabbitMQ postawiliśmy na wykorzystanie po jednej kolejce na odbieranie wiadomości bezpośrednich oraz zbiorczych. W połączeniu z faktem, że przesyłane wiadomości posiadają różną formę, doprowadziło to do potrzeby deserializacji wiadomości na różne typy obiektów, w zależności od typu otrzymanej wiadomości. Naszym zdaniem problem ten nie został przez nas rozwiązany zadowalająco i mógłby zostać wykonany lepiej.

Z pominięciem wyżej wymienionych aspektów, uważamy projekt komunikacji i procesów biznesowych za dobrze wykonany, w sposób umożliwiający łatwą implementację. Dzięki głęboko przeemyślanym rozwiązaniom w tych płaszczyznach, ich implementacja była mało problematyczna. Podczas testów nie wynikły również żadne problemy, które wymagałyby modyfikacji ustalonych procesów i sekwencji.

0.2. Możliwe ścieżki rozwoju

0.2.1. Panel administratora

Funkcjonalność panelu administratora jest obecnie ograniczona. Pozwala on jedynie na podejrzanie stanu zasobów w systemie. Informacja ta pozwala ocenić, czy potrzebne jest uruchomienie nowych instancji serwera wirtualizacji. Jest to jednak jedyna dostępna informacja.

Możliwym rozwinięciem jest udostępnienie wglądu w cały model systemu przechowywany przez nadzorców. Umożliwi to ocenę stopnia zajętości maszyn każdego typu oraz ilości użytkowników systemu. Dodatkowo wgląd do stanu konkretnych maszyn i sesji może ułatwić rozwiązywanie problemów.

0.2.2. Użycie konkretnych kart graficznych

W konfiguracji typu maszyny wirtualnej możliwe jest jedynie wskazanie, czy ma ona posiadać kartę graficzną na wyłączność. Oznacza to, że otrzymana karta graficzna nie jest sprecyzowana i może być dowolną ze skonfigurowanych w serwerze wirtualizacji.

W celu uniknięcia niespodzianek, co do modelu otrzymanej karty graficznej, preferowane byłaby możliwość sprecyzowania konkretnego modelu karty graficznej przekazywanej do maszyny danego typu. Możliwym sposobem osiągnięcia tej funkcjonalności jest rozszerzenie konfiguracji serwera wirtualizacji o możliwość nadania identyfikatora przekazywanym kartom graficznym.

0.3. Otwarte problemy