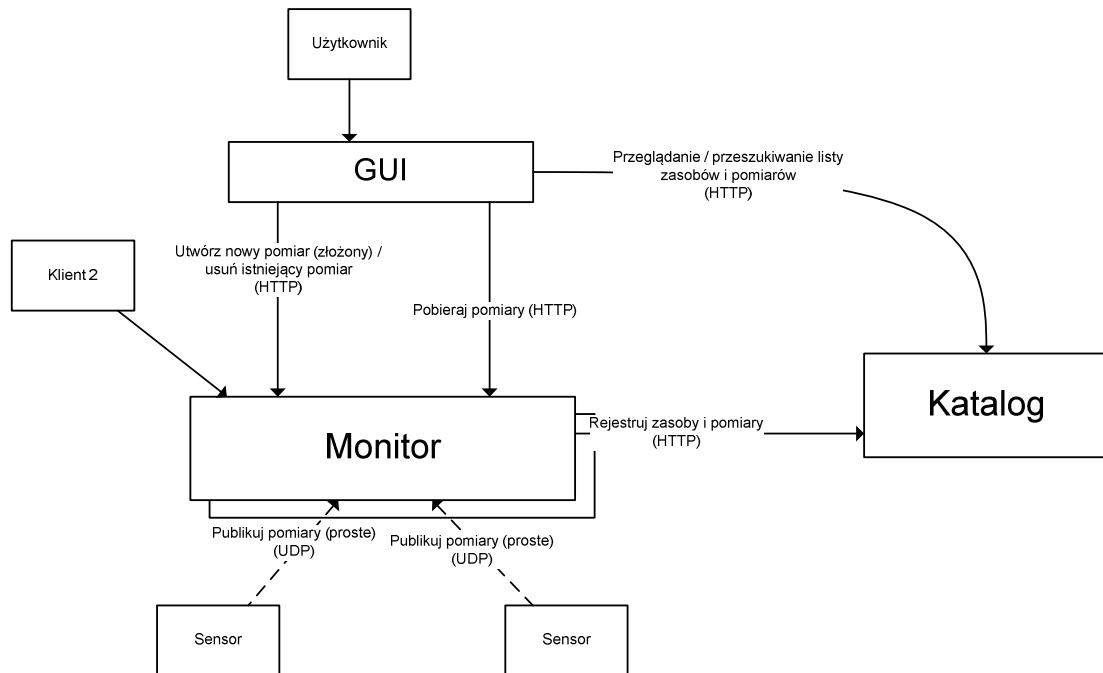


Temat projektu

System monitorowania rozproszonych zasobów komputerowych, np. obciążenia CPU, zużycia pamięci, obciążenia sieci.

Architektura systemu



Rysunek 1: Architektura systemu do monitorowania zasobów.

Architekturę przedstawia Rys. 1. Główne komponenty systemu to:

Sensor

- Dokonuje **pomiarów** wybranych **metryk** dla określonych **zasobów** i wysyła pomiary cyklicznie do Monitora.
- Przykład zasobu i metryki: *Host / CPU Utilization* (aktualne zużycie CPU dla hosta).
- Wiadomość przesyłana przez sensor do monitora powinna zawierać: nazwę (identyfikator) zasobu, nazwę metryki, dane pomiarowe (zależne od metryki).
- Wiadomości powinny mieć formę tekstową (a nie np. serializowanych obiektów). Sugerowany format bazowy: JSON.

Monitor

- Zbiera dane od sensorów i **udostępnia pomiary dla klientów**.
- Również umożliwia tworzenie nowych **złożonych pomiarów**, które np. agregują pomiary proste.
- Przykład złożonego pomiaru: *średnie obciążenie CPU z ostatnich 5 minut obliczane co minutę*.

Katalog

- Prosty komponent, którego jedynym zadaniem jest umożliwienie przeglądania i przeszukiwania dostępnych monitorowanych zasobów i pomiarów.
- Każdy Monitor odpowiedzialny jest za rejestrowanie i uaktualnianie w Katalogu swojej listy zasobów i pomiarów.

Klient

- Klient 1: przeglądarka sterowana przez użytkownika, umożliwiającą tworzenie nowych strumieni i wyświetlającą wykresy wybranych pomiarów.

- Klient 2: program automatycznie przeszukujący monitorowane zasoby i pomiary w jakimś celu (np. podobnie do polecenia 'top' wyświetlający najbardziej obciążone komputery).

Zastosowanie usług REST

Komponenty serwerowe (Monitor i Katalog) mają udostępniać swoje usługi przez interfejs REST (usługa sieciowa stosująca protokół HTTP). Metodologia tworzenia usług w paradygmacie REST oznacza m.in., że:

1. System rozproszony jest zespołem „zasobów”, które mają swoje identyfikatory – URI.
2. Na każdym z zasobów można wykonać tylko cztery proste operacje, tzw. CRUD:
 - **C (Create)**: stworzenie nowego zasobu
 - **R (Read)**: odczytanie stanu zasobu
 - **U (Update)**: zmiana stanu zasobu
 - **D (Delete)**: usunięcie zasobu
3. Powyższe operacje odwzorowują się na metody protokołu HTTP następująco ('<...>' oznacza treść wiadomości ('body') przesyłanej do serwera):
 - GET {URI} → pobranie stanu zasobu
 - POST {URI_kolekcji} <reprezentacja_zasobu> → utworzenie nowego zasobu
 - POST {URI_zasobu} <zmiana_stanu> → (częściowa) zmiana stanu zasobu
 - PUT {URI} <nowy_stan> → całkowita podmiana stanu zasobu na nowy
 - DELETE {URI} → usunięcie zasobu
4. Wszystkie bardziej złożone operacje są efektem ubocznym zmiany stanu zasobu.
 - Przykładowo operacja POST <http://www.example.com/orders/123> {"status"="paid"} spowoduje zmianę statusu zamówienia na „zapłacono”, ale również przekazanie zamówienia do działu realizacji, nadanie paczki, wysłanie e-maila z powiadomieniem, itp.
5. Serwer nigdy nie powinien przechowywać tzw. stanu sesji (czyli stanu aplikacji). Innymi słowy wszystkie operacje na serwerze są bezstanowe.
 - Przykład: bezstanowy koszyk na zakupy (<http://alandean.blogspot.com/2008/11/on-restful-basket-state.html>)

Dodatkowe wymagania

1. Klient ma mieć możliwość pobrania z Monitora listy wszystkich monitorowanych zasobów i dostarczanych dla nich pomiarów.
2. Klient może przeszukiwać Katalog przy pomocy zapytań, np. znaleźć zasób, który jest hostem i ma w nazwie 'zeus'.
3. Pomiary proste (dostarczane bezpośrednio przez sensory) powinny być udostępniane przez Monitor jako lista wartości. Dla każdego pomiaru określone jest ile ostatnich wartości ma być zapamiętywanych.
4. W przeglądarce można:
 - a. Przeglądać listy dostępnych zasobów i pomiarów (również aktualne ich wartości).
 - b. Wyszukiwać zasoby i pomiary (np. po nazwie).
 - c. Wyświetlać wybrane pomiary na wykresie, uaktualnianym co pewien czas (np. 5 sekund).
5. Klient ma mieć możliwość stworzenia nowego pomiaru (złożonego) w Monitorze. Sposób tworzenia nowego pomiaru może być bardzo skomplikowany, więc ograniczymy się do jednego typu pomiaru złożonego: **agregacji przez obliczanie średniej ruchomej** (np. średnia z ostatnich 5 minut obliczana co minutę).
6. Pomiar złożony powinien być dostarczany przez Monitor w ten sam sposób jak prosty.
7. Pomiar złożony można usuwać.
8. Usunąć pomiar złożony może tylko ten użytkownik, który go stworzył. Oznacza to, że w systemie musi istnieć mechanizm uwierzytelniania i autoryzacji.
9. W przeglądarce dodatkowo można:
 - a. Definiować nowy pomiar złożony jako średnią ruchomą istniejącego pomiaru.
 - b. Usuwać wcześniej zdefiniowany pomiar złożony.

10. Ma być zaimplementowany klient-program, który automatycznie realizuje jakiś cel, np. cyklicznie wypisuje top 10 najbardziej obciążonych maszyn. Program musi uwzględniać zmiany, np. dodanie nowych lub usunięcie istniejących maszyn z listy monitorowanych zasobów.

Kamienie milowe i produkty

KM1: 1-2 kwietnia 2014

Projekt systemu. Jego głównymi elementami ma być projekt interfejsu REST, tj. listy zasobów, ich reprezentacji, a także projekt działania podstawowych scenariuszy użycia systemu z wykorzystaniem interfejsów REST.

Produkty:

- Dokument opisujący projekt systemu.

KM2: 6-7 maja 2014

Spełnione są wymagania 1-4. Działa kilka sensorów rozmieszczonych na różnych komputerach, które mierzą co najmniej dwie metryki (np. obciążenie CPU i zużycie pamięci).

Produkty:

- Prototyp systemu + demonstracja działania.
- Raport : opis implementacji systemu.

KM3: 10-11 czerwca 2014

Spełnione są pozostałe wymagania (5-10).

Produkty:

- Prototyp systemu + demonstracja działania.
- Raport : opis końcowej implementacji systemu.