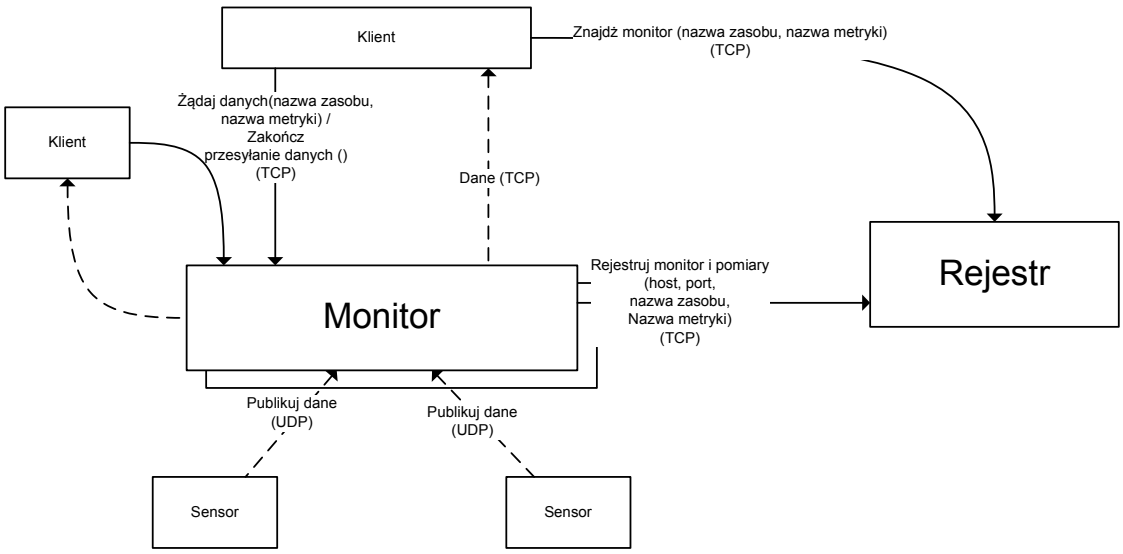
Bartosz Baliś

## Systemy rozproszone, laboratorium – zadanie z socketów

Komponenty systemu:

1. **Sensor**: dokonuje pomiarów **metryk** dla określonych **zasobów** i wysyła je cyklicznie do Monitora. Przykład pary zasób / metryka: *Host* / *CPU Usage* (aktualnie zużycie CPU dla hosta). Wiadomość przesyłana przez sensor do monitora powinna zawierać: nazwę zasobu, nazwę metryki, dane pomiarowe (zależne od metryki).
2. **Monitor**: zbiera dane od sensorów, również odbiera żądania danych od klientów i przesyła do nich dane zgodne z tymi żądaniami. Żądanie przesylane od klienta do Monitora zawiera nazwę zasobu i nazwę metryki.
3. **Rejestr**: usługa do rejestrowania i wykrywania monitorów, które dostarczają żądanych pomiarów. Monitor rejestruje się w rejestrze (swój host i numer portu, na które klient może wysyłać żądania), a także listę udostępnianych zasobów i metryk. (Jeśli nowy sensor dołączy się do Monitora, powinien on uaktualnić swój wpis w rejestrze). Klient może w rejestrze znaleźć monitor, który dostarcza żądanej metryki dla podanego zasobu.
4. **Klient**: końcowy konsument danych. Interakcje z pozostałymi komponentami opisane wyżej. Uwaga: klient kończy subskrypcję na dane wysyłając do Monitora żądanie zakończenia przesyłania danych (w żądaniu tym może przesłać id subskrypcji, które wcześniej otrzymał od Monitora).

Połączenia między komponentami i protokoły komunikacji przedstawia poniższy rysunek.




Wymagania:

1. Do Monitora może podłączyć się wiele sensorów i wielu klientów jednocześnie.
2. W systemie może działać wiele niezależnych monitorów. (Ale konkretny sensor zawsze jest podłączony tylko do jednego).

3. W wersji uproszczonej (ocena maksymalna 4.0) może nie być Rejestru.
4. Implementacja:
    a. Monitor: Java lub C#. Do obsługi wielu kanałów komunikacji należy wykorzystać selector (W Javie: **`java.nio.channels.Selector)`.**
    b. Sensor: inny język niż powyższe (C, Python, etc.).
    c. Klient i Rejestr: dowolny język według uznania.