Politechnika Warszawska

Zaawansowane programowanie obiektowe i funkcyjne (Java)

Bramka REST API na TCP

Wojciech Gajda 304494

Krzysztof Leszek 313310

Marcin Latoszek 313307

Prowadzący: dr inż. Janusz Rafałko

Data oddania: **13 grudnia 2022**

Motywacja

Rynek urządzeń IoT rozwija się niezwykle prężnie. Szczególnie widoczne jest to w przypadku inteligentnych urządzeń typu smart-home. Na rynku dostępne są coraz to nowsze urządzenia mające na celu ułatwić codzienne życie domownikom. Dzięki zwiększającej się różnorodności ceny urządzeń spadają, tym samym zwiększając swoją dostępność.

Jedną z gałęzi urządzeń smart-home stanowi inteligentne oświetlenie. Do niedawna liderem tej kartegorii była seria lamp i żarówek Hue, produkcji Philips. W ostatnim czasie na rynek wkroczyła firma Yeelight będąca spółką córką marki Xiaomi. Produkty z linii Yeelight okazują się być o rząd wielkości tańsze od swoich odpowiedników z serii Hue. Pozwala to myśleć o nich jako o doskonałej alternatywie. Jest jednak mały haczyk...

Rysunek 1: Żarówka Yeelight RGB wykorzystana w projekcie

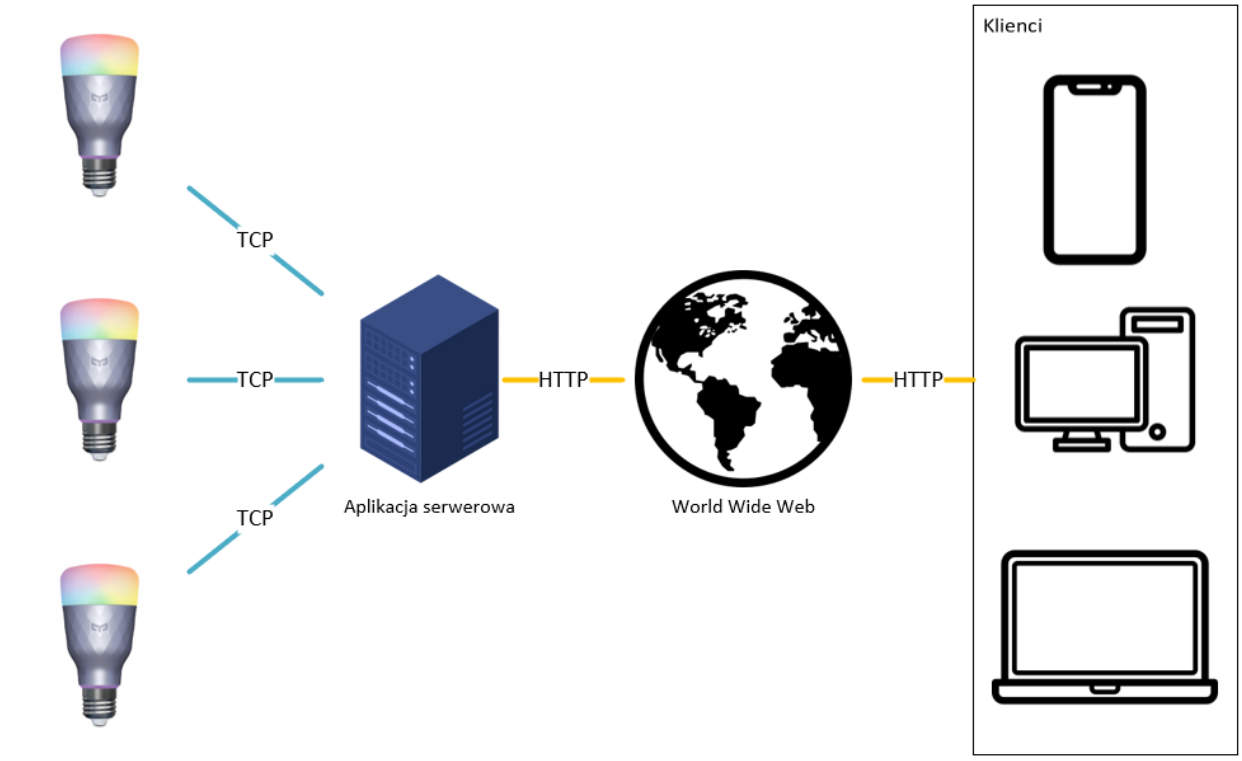
O ile w przypadku serii Hue komunikacją z urządzeniami odbywa się w sieci lokalnej, z ewentualnym wystawieniem dostępu na świat zewnętrzny, tak cała komunikacja z żarówkami Yeelight odbywa się przez serwer producenta zlokalizowany gdzieś na wschodniej części globu. Oprócz widocznych opóźnień, takie rozwiązanie posiada wiele wad, zaczynając od wytwarzania zbędnego ruchu sieciowego i kończąc na kwestiach związanych z bezpieczeństwem sieci. Obserwując tendencje na rynku wsparcie producenta może nie trwać długo, a ewentualne wyłączenie serwera głównego spowodowało by załamanie systemu. Aby zwiększyć porządną niezawodność systemu niezbędne jest wykluczenie zewnętrznych zależności. Na szczęście, producent przewidział dla swoich urządzeń alternatywny sposób komunikacji.

W dokumencie **Yeelight WiFi Light InterOperation Specification**[1] wyspecyfikowany został interfejs API pozwalający na wyszukiwanie żarówek za pośrednictwem protokołu UDP i komunikację z nimi przez protokół TCP.

W przypadku urządzeń IoT stanowość połączenia nie jest zjawiskiem porządanym. Komunikacja TCP nie jest powszechnie wspierana, gdyż wymaga ona przechowywania otwartych połączeń. Znacznie lepiej w tej roli sprawdza się komunikacja poprzez REST HTTP.

Cel projektu

Celem projektu jest stworzenie aplikacji serwerowej w postaci bramki wystawiającej API REST, która zarządza urządzeniami znajdującymi się w sieci lokalnej. Aplikacja znajduje wszystkie inteligentne urządzenia w sieci lokalnej, po czym nawiązuje z nimi połączenie. Urządzenie z którymi połączył się serwer stają się dostępne dla klientów łączących się protokołem HTTP. Serwer pełni jednocześnie role integratora. System okresowo odpytuje urządzenia w celu aktualizowania stania połączeń i ewentualnego usuwaniecia nieaktywnych urządzenia z listy dostępnych.



Rysunek 2: Struktura komiunikacji

[1] https://www.yeelight.com/download/Yeelight_Inter-Operation_Spec.pdf

Funkcjonalności projektu - use cases

Jako użytkownik mogę:

- pobrać listę dostępnych żarówek
- wyłączać i włączać żarówkę
- zmieniać kolor żarówki za pomocą modelu RGB żeby móc kontrolować kolor świecenia żarówki
- zmieniać jasność żarówki żeby kontrolować ilość dostarczanego przez żarówkę światła

Jako system mogę ...

- co określony czas przeszukiwać okolicę w celu odnalezienia żarówek oraz łączyć się z nimi żeby pozwolić użytkownikowi na ich kontrolę
- co określony czas mogę sprawdzić czy urządzenia w puli otwartych połączeń są nadal dostępne, aby usunąć urządzenia nieaktywne

Wybrane technologie

Aplikacja serwera zaprogramowana zostanie z wykorzystaniem języka Java 17. W aplikacji wykorzystany zostanie framework Spring Boot znacząco ułatwiający tworzenie serwera HTTP. Serwer zostanie stworzony z wykorzystaniem zasad Dependency Injection oraz Invertion of Control. Do komunikacji z urządzeniami wykorzystane zostaną standardowe biblioteki do komunikacji TCP/UDP. Do testów wykorzystany zostanie pakiet Mockito. Rozwiązane zostanie ponadto skonteneryzowane (Docker), a za platformę testową posłuży domowy serwer.