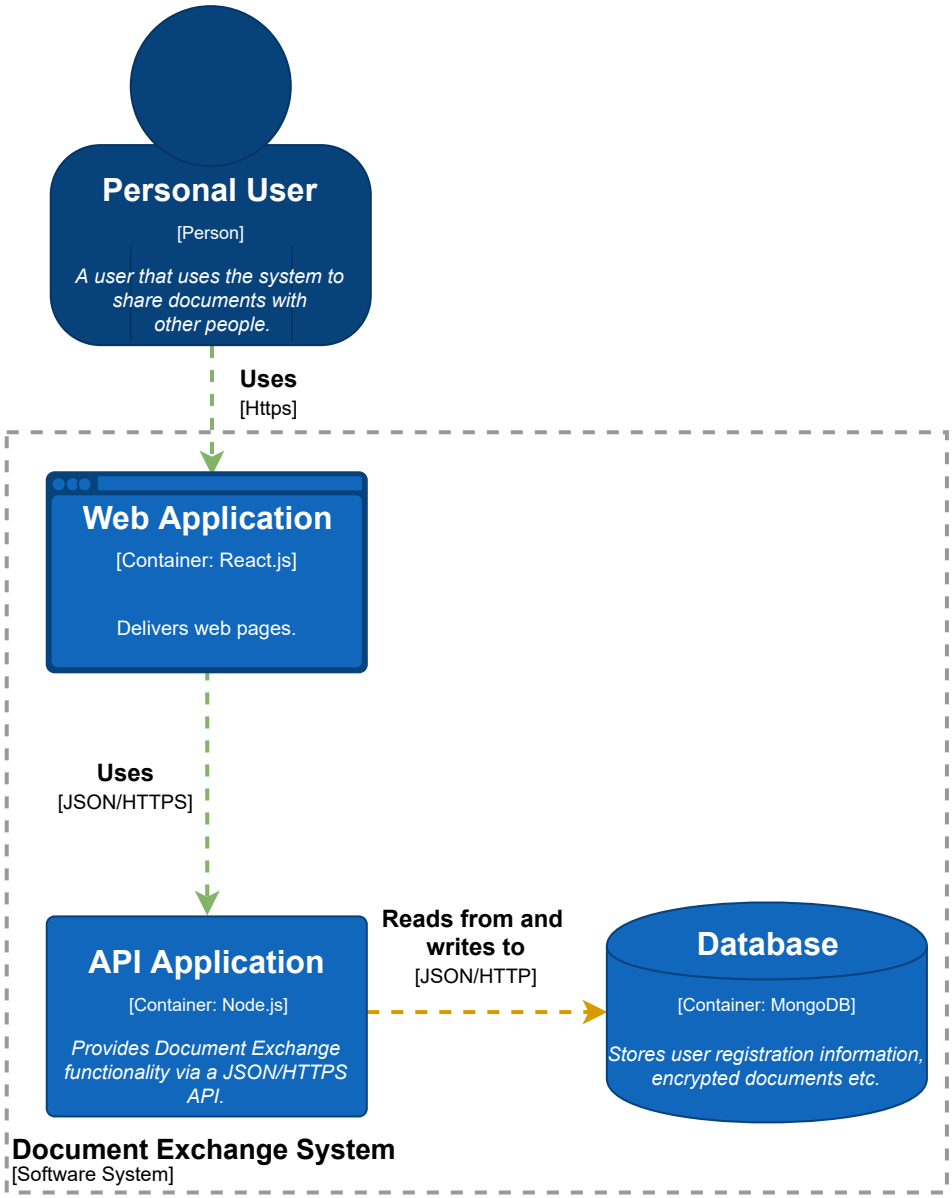
Projekt wysokiego poziomu

1. Ogólne właściwości systemu

1. Architektura aplikacji

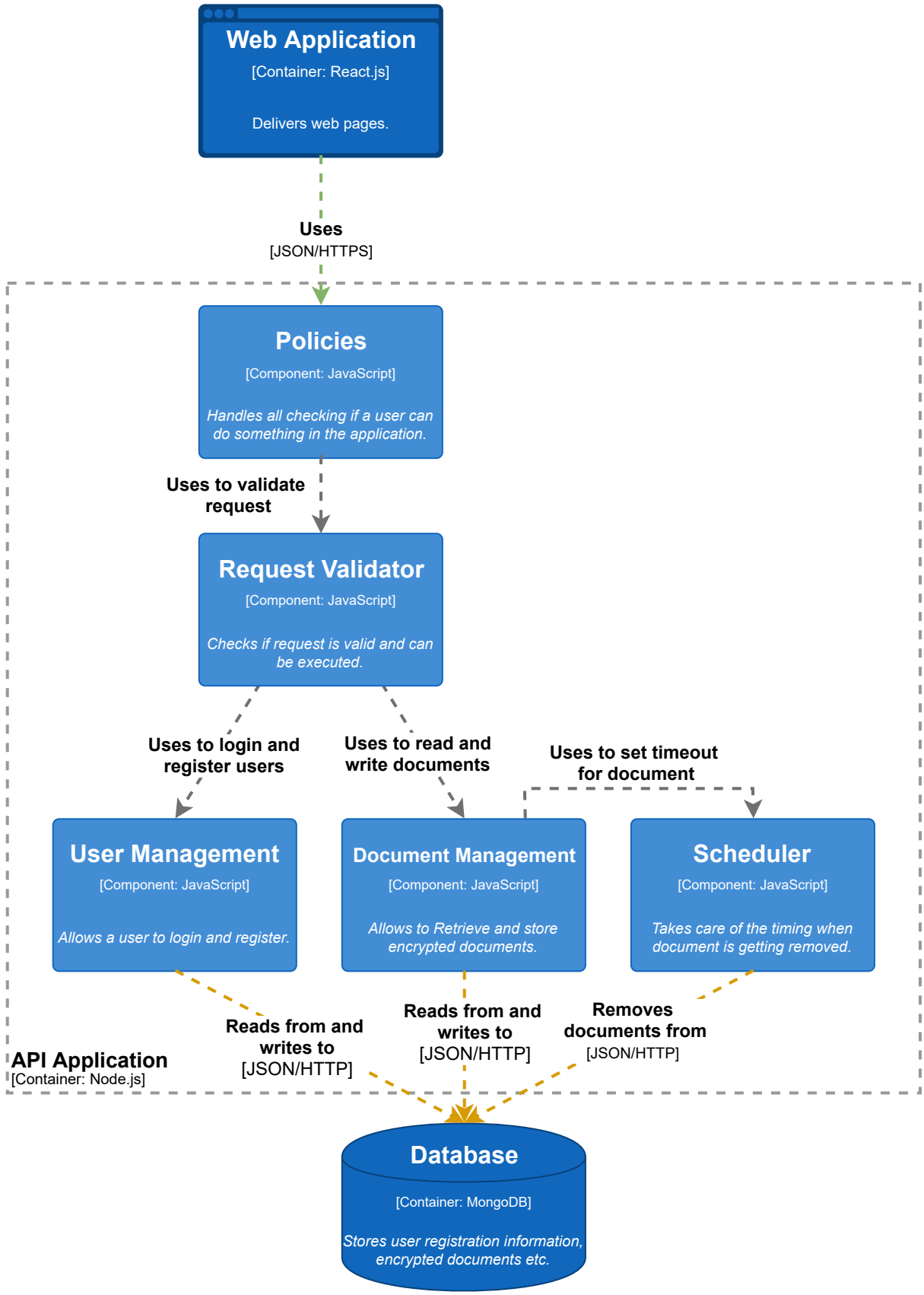


Container diagram for the Document Exchange System

Architektura aplikacji składa się z trzech głównych komponentów:

- Web Application - front-end [React.js]
- API Application - back-end [Node.js]
- Database - serwer bazodanowy przechowujący dane (dane użytkowników, zaszyfrowane dokumenty) [MongoDB]

1.1 API Application Component



Component diagram for the API Application Container

Web Application

Jest to komponent odpowiedzialny za komunikację użytkownika z systemem. Jest odpowiedzialny za pobieranie danych od użytkownika oraz przekazanie ich do back-endu. Następnie back-end aplikacji na podstawie tych danych wykonuje określone zadanie i przekazuje wynik końcowy z powrotem do Web Application.

Policies

Do zadań tego komponentu należy walidacja uprawnień użytkownika. Dla każdorazowej próby wykonania akcji przez użytkownika komponent ten sprawdza poziom uprawnień użytkownika. Do głównych zadań tego komponentu należą:

- Czy użytkownik jest zalogowany do systemu;
- Czy użytkownik posiada uprawnienia do pobrania docelowego pliku;
- Czy użytkownik posiada uprawnienia do udostępnienia danego pliku innym użytkownikom system;
- Czy użytkownik posiada wystarczające uprawnienia grupy do określonego zasobu systemu;
- Czy ważność sesji połączeniowej dla użytkownika nie wygasła.

Request Validator

Komponent powiązany z Policies - sprawdza poprawność wprowadzonych danych, zapytań, które będą wykonane po stronie aplikacji serwerowej.

User Management

Komponent zarządzający sesją zalogowanego użytkownika w systemie. Zapewnia mechanizmy logowania oraz rejestracji użytkownika do systemu.

Document Management

Odpowiada za zarządzanie dokumentami, które zostały udostępnione przez użytkowników. Do głównych zadań tego komponentu należą:

- Operacja zapisu zaszyfrowanego dokumentu do serwera bazodanowego wraz z datą wygaśnięcia pliku (timeout przekazywany jest do komponentu Scheduler);
- Operacja usuwania pliku przez jego właściciela bądź osoby uprawnionej (należącej do grupy Administrator);
- Przekazywanie uprawnień innym użytkownikom systemu przez właściciela pliku;
- Pobranie pliku z bazy danych i przekazanie go uprawnionym użytkownikom w przypadku wywołania operacji *Pobierz plik* przez docelowego użytkownika;
- Uprawnienia do wyświetlenia i pobrania zaszyfrowanych dokumentów są dostępnie jedynie w obrębie właściciela oraz dodanych przez niego osób uprawnionych do pobrania zasobu.

Komponent zapewnia również bezpieczne połączenie pomiędzy użytkownikami (nadawcy i odbiorcy) w celu przekazania klucza dostępu umożliwiającego odszyfrowanie udostępnionego pliku. Klucz dostępu jest szyfrowany kluczem symetrycznym i przekazany według protokołu Diffiego-Hellmana. System **nie przechowuje** w jakikolwiek sposób klucza dostępu do pliku i jest znany jedynie przez właściciela udostępnianych danych oraz osobom, którym został on przekazany.

Scheduler

Komponent przechowujący znaczniki czasu (timestamp) wygasnięcia dokumentów. W przypadku upłynięcia ważności pliku komponent ten usuwa z serwera bazodanowego dokument.

Database

Przechowuje rekordy o użytkownikach znajdujących się w systemie wraz z poziomem ich uprawnień, przypisania do grupy użytkowników oraz przypisanym im zasobów (pliki, dokumenty, czy są właścicielem zasobu). Przechowywane hasła dostępu do użytkowników są zabezpieczone hashem: bcrypt. Przechowywane są zaszyfrowane dokumenty ze znacznikiem czasu wygaśnięcia oraz z listą uprawnionych do odczytu użytkowników. Klucze deszyfrujące dokumenty nie są dostępne w systemie bazodanowym.

2. Stany aplikacji

Główna część aplikacji - aplikacja serwerowa z dostępem przez przeglądarkę internetową, napisana w języku JavaScript (Front-end React.js; Back-end Node.js) jest obsługiwana przez platformę Docker.

3. Zasoby aplikacji

Aplikacja działa na oprogramowaniu Docker - jest to otwarte oprogramowanie do realizacji wirtualizacji na poziomie systemu operacyjnego, działające jako "platforma dla programistów i administratorów do tworzenia, wdrażania i uruchamiania aplikacji rozproszonych".

4. Sposób obsługi błędów

W przypadku wystąpienia błędów aplikacji po stronie użytkownika lub po stronie klienta, procedura obsługi błędów wygląda następująco:

- Wystąpienie błędu lub nieobsługiwanego wyjątku;
- Zapisanie informacji o kodzie błędu z danymi, które te błędy wywołały do pliku logu znajdującego się w katalogu Application/ErrorLogs;
- Przekazanie ogólnej informacji o błędzie z kodem błędu użytkownikowi - wyświetlenie komunikatu.

5. Przechowywane dane

W ramach działania aplikacji dane przechowywane są w dwóch magazynch:

- Pamięć dyskowa jako instancja utworzona z wirtualnego kontenera dockera - zasoby zarządzające aplikacją po stronie serwera, logi systemowe, zabezpieczona kopia bazy danych.
- Baza danych - przechowywanie wszelkich informacji dotyczących systemu takich jak:
    - Dane użytkownika systemu - login, hasło - hasło zabezpieczone hashem: bcrypt, adres e-mail, przypisana grupa użytkownika;
    - Grupy użytkowników - użytkownik, administrator;
    - Zaszyfrowane kluczem symetrycznym dokumenty z odnośnikiem do właściciela pliku, z listą uprawnionych do pobrania zasobu użytkowników oraz znacznikiem czasowym (timestamp) ustalającym czas wygaśnięcia pliku, klucze deszyfrujące nie są zapisywane w obrębie systemu bazodanowego.

6. Dostęp do bazy danych

W aplikacji występuje zależność do systemu bazodanowego przechowującego dane o użytkownikach, ich uprawnieniach oraz zaszyfrowanych plikach. Komunikacja aplikacji ze strony serwerower z bazą danych odbywa się w lokalnym środowisku - dostęp do systemu bazodanowego jedynie wewnątrz infrastruktury serwera.