

Opisy systemów opartych na infrastrukturze chmurowej (vcloud)

1. System Monitorowania kształcenia Pracowników Medycznych (SMK II)

Opis systemu

System jest zintegrowanym systemem wspomagającym przebieg procesu kształcenia podyplomowego kadr medycznych oraz gromadzącym i przetwarzającym dane na temat kształcenia. Gromadzone w SMK informacje mają umożliwić nie tylko prowadzenie analiz dotyczących zapotrzebowania na specjalistów w systemie ochrony zdrowia, lecz także powinny zapewnić efektywną organizację, planowanie i monitorowanie procesu specjalizacji oraz ocenę jakości i skuteczności tego kształcenia.

Zarys architektury

System zbudowany jest w architekturze mikroserwisowej, z wykorzystaniem orkiestratora konteneryzacji Kubernetes. Część serwerowa /backend/ stworzona jest z wykorzystaniem języka Java, uruchamiany jako procesy spring boot. Część interfejsu użytkownika /frontend/ stworzona jest we frameworku Angular, działająca w przeglądarce jako SPA. System SMK uruchamiany będzie etapowo, zatem w okresie przejściowym ustanowione zostało połączenie z systemem SMK1 pozwalające jednocześnie pracę zalogowanego użytkownika w obu systemach.

Spójną sesję zapewnia usługa SSO dostarczana przez system ePŁOZ, dodatkowo zbudowane zostaną usługi integracyjne pozwalające na wymianę danych pomiędzy systemami SMK1 i SMK.

Kontenery systemu poza relacyjną bazą danych uruchamiane są pod kontrolą orkiestratora Kubernetes.

Relacyjne bazy danych uruchamiane są na dedykowanych maszynach wirtualnych. W przypadku środowiska produkcyjnego są to dwie niezależne maszyny wirtualne.

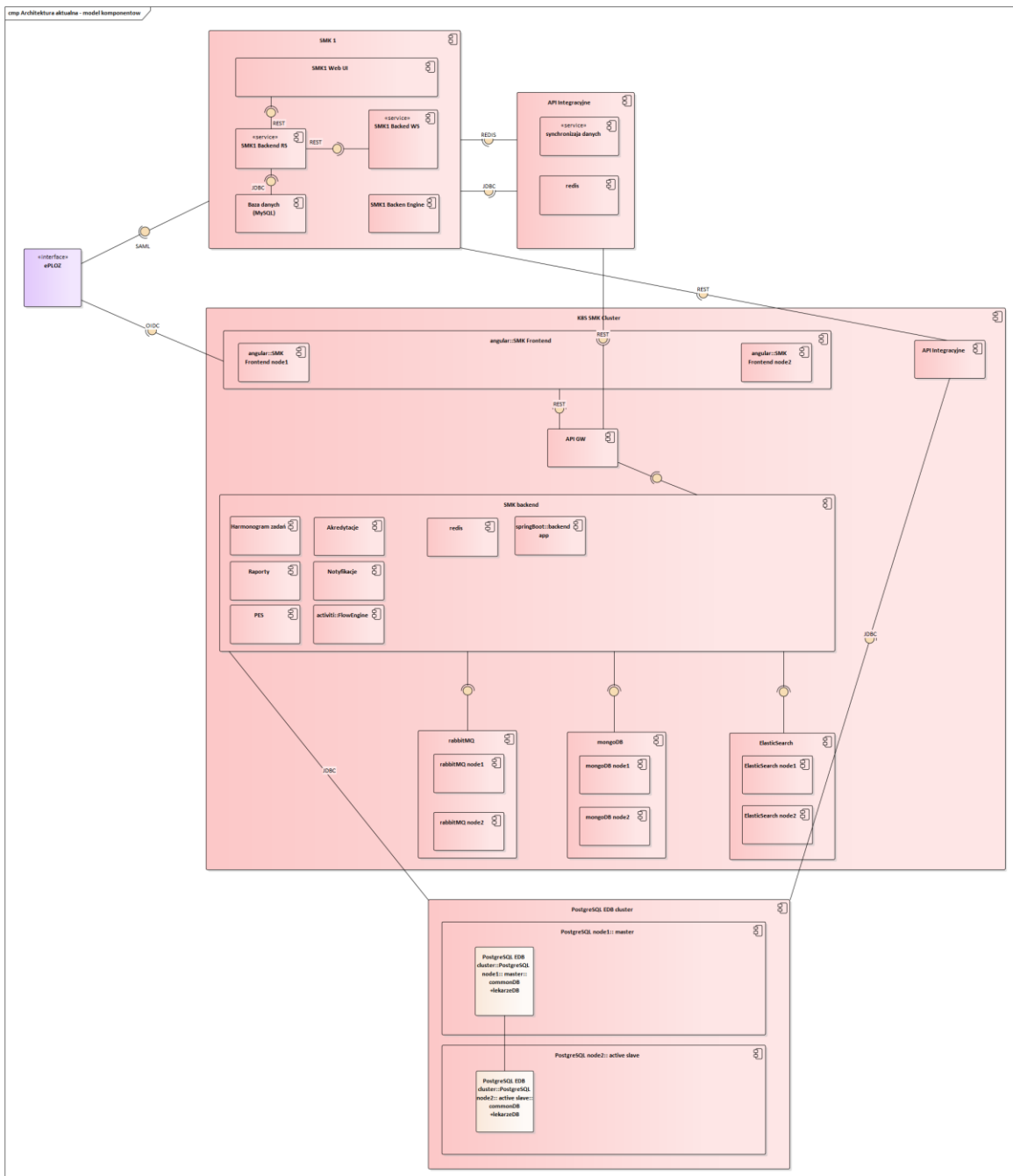
Architektura systemu SMK składa się z komponentów wymienionych i opisanych poniższej tabeli.

Komponenty architektury systemu

Nazwa elementu	Opis
Activiti	Silnik klasy workflow, licencjonowany na zasadach Open Source
API integracje 1	Komponent obsługujący komunikację pomiędzy systemami SMK1 oraz SMK
API Integracje 2	Komponent obsługujący komunikację pomiędzy systemami SMK oraz SMK1
Backend	Kontener procesu serwerowego systemu SMK, udostępniającego usługi dla aplikacji zewnętrznych i wewnętrznych.
EDB	Komercyjny Silnik rdbms zgodny z PostgreSQL. Baza danych systemu SMK, rozumiana jako baza produkcyjna, zawiera wszystkie obiekty bazodanowe wraz z relacjami - wynikające z modelu dziedziny.

	Baza danych będzie instalowana bezpośrednio na maszynach wirtualnych Vmware.
ELK	Silnik nierelacyjnej tekstowej bazy danych Elastic Search, system przetwarzania logów Log Stash, system klasy business intelligence Kibana. Rozwiązanie wykorzystywane w projekcie do przechowywania i przetwarzania dzienników zdarzeń, śladów audytowych, logów aktywności użytkownika.
MongoDB	Silnik nierelacyjnej obiektowej bazy danych, wykorzystywany w projekcie do przechowywania binarnych załączników.
Rabbit MQ	System kolejkowy
Redis	Silnik bazy kucz-wartość, działającej w trybie in-memory, wykorzystywany w projekcie do buforowania zbiorów danych.
SMK	System Monitorowania Kształcenia.
SMK1	Poprzednia wersja Systemu Monitorowania Kształcenia.

Poniższy diagram przedstawia ogólną architekturę rozwiązania SMK (model C2)

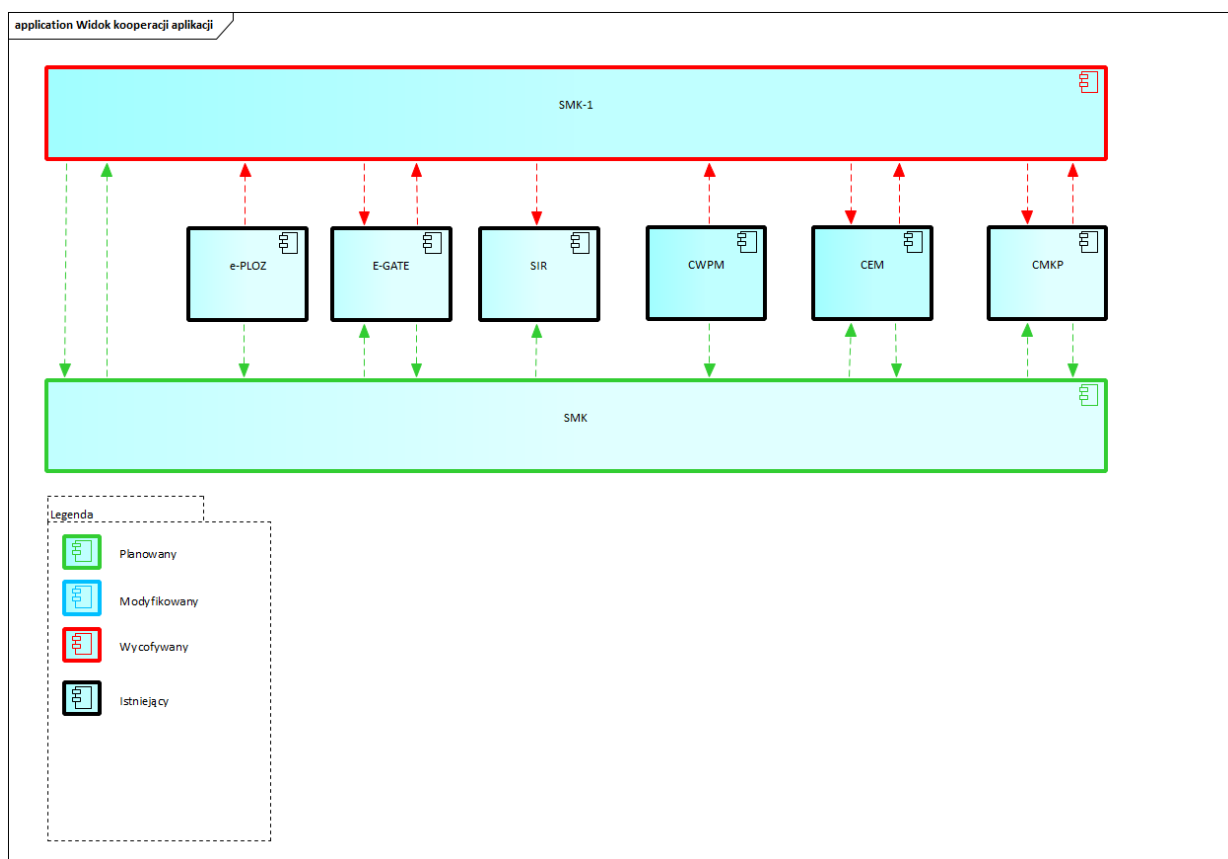


Integracje z systemami zewnętrznymi

System SMK zintegrowany będzie z następującymi systemami i zewnętrznymi usługami:

Lp.	Nazwa systemu / usługi	Właściciel systemu / usługi	Opis
1	Elektroniczna Platforma Logowania Ochrony Zdrowia – e-PLOZ	Centrum e-Zdrowia	System świadczący usługi uwierzytelniania i autoryzacji użytkowników.
2	SIR System Informatyczny Rezydentur	Ministerstwo Zdrowia	System Informatyczny Rezydentur obsługujący umowy rezydentur MZ.

3	CKPPIP	Centrum Kształcenia Podyplomowego Pielęgniarek i Położnych.	WS wymiany danych na potrzeby egzaminów.
4	CMKP	Centrum Medyczne Kształcenia Podyplomowego	WS wymiany danych na potrzeby organizacji kursów w ramach specjalizacji.
5	CEM	Centrum Egzaminów Medycznych	WS wymiany danych na potrzeby organizacji egzaminów specjalizacji.
6	CWPM	Centralny Wykaz Pracowników Medycznych	
7	E-Gate	Podpis elektroniczny	Wymiana danych w zakresie podpisów elektronicznych
8	SMK1	Centrum e-Zdrowia	Dotychczasowy system SMK



Wykaz i skalowanie środowisk

Skalowanie środowisk

	#nodes	test	#nodes	eva	#nodes	prod	suma per node
Frontend host core (vCpu)	2	2	1	2	2	4	14
Frontend host ram (GB)	2	8	1	12	2	16	60
Frontend host storage (GB)	2	64	1	64	2	64	320
Backend APP core (vCpu)	2	3	1	4	4	4	26
Backend APP ram (GB)	2	8	1	24	4	32	71
Backend APP storage (GB)	2	64	1	64	4	64	199
PostgreSQL EDB (vCpu)	2	2	1	2	2	8	17
PostgreSQL EDB ram (GB)	2	8	1	8	2	64	85
PostgreSQL storage (GB)	2	256	1	256	2	512	1029
MongoDB core (vCpu)	2	2	1	2	2	4	13
MongoDB ram (GB)	2	12	1	32	2	64	113
MongoDB storage (GB)	2	32	1	64	2	72	173
RabbitMQ core (vCpu)	2	2	1	2	2	2	11
RabbitMQ ram (GB)	2	8	1	12	2	16	41
RabbitMQ storage (GB)	2	24	1	24	2	64	117
Elastic Search core (vCpu)	2	2	1	2	2	4	13
Elastic Search ram (GB)	2	8	1	16	2	24	53
Elastic Search storage (GB)	2	32	1	64	2	128	229
API Integracyjne SMK2 (vCpu)	3	2					
API Integracyjne SMK2 (ram)	3	16					
API Integracyjne SMK2 (GB)	3	64					
VMs							
suma CPU		26		14		60	100
suma RAM		104		104		496	704
suma storage		944		536		1936	3416

Informacje dotyczące przestrzeni dyskowej środowisk

	Baza danych	Aplikacja
Środowisko produkcyjne	1,8 TB	768 GB
Środowisko przedprodukcyjne (szkoleniowe)	512 GB	248 GB
Środowisko testowe	512 GB	248 GB



2. Rejestr Systemu Kodowania (RSK)

Opis systemu

Celem działania systemu jest poprawa jakości usług medycznych dzięki dostosowaniu do warunków polskich klasyfikacji ICD-11, opracowanie ostatecznej polskiej wersji ICD-10, stworzenie polskich wersji informatycznych narzędzi, wypracowanych przez WHO, które wspomogą użytkowników ICD-11. W celu optymalnego korzystania z klasyfikacji ICD-11 w ramach Projektu Klasyfikacja ICD-11 zostanie dostosowana do formatów używanych w systemie informacji w ochronie zdrowia oraz umieszczona na platformie - Rejestr Systemów Kodowania. Rejestr Systemów Kodowania stanie się referencyjną bazą klasyfikacji i słowników medycznych wykorzystywanych w systemie informacji ochrony zdrowia, w szczególności przez systemy i rejestry medyczne tworzone przez Centrum e-Zdrowia. Międzynarodowa klasyfikacja chorób i związanych z nimi problemów zdrowotnych (ICD) jest międzynarodowym standardem systematycznego rejestrowania, raportowania, analizy, interpretacji i porównywania danych dotyczących śmiertelności i zachorowalności. Wdrożenie w krajowych systemach teleinformatycznych najnowszej wersji klasyfikacji ICD-11 pozwoli na identyfikację problemów zdrowotnych przy użyciu aktualnego i klinicznie odpowiedniego systemu klasyfikacji. Stanom zdrowotnym lub wypadkom przypisuje się kody ICD-11, co skutkuje danymi, które mogą być wykorzystywane do opracowywania skutecznych polityk zdrowia publicznego oraz do pomiaru ich wpływu, przydzielania zasobów, poprawy leczenia, poprawy profilaktyki lub wykorzystania do rejestracji klinicznej.

Obecnie w Polsce stosowana jest klasyfikacja ICD-10 w wersji z roku 2008. Aktualizacja klasyfikacji umożliwi właściwe i zgodne z aktualną wiedzą, identyfikowanie zdarzeń medycznych.

Przykład realizacji usług

Międzynarodowa klasyfikacja chorób i związanych z nimi problemów zdrowotnych (ICD) jest międzynarodowym standardem systematycznego rejestrowania, raportowania, analizy, interpretacji i porównywania danych dotyczących śmiertelności i zachorowalności. Wdrożenie w krajowych systemach teleinformatycznych najnowszej wersji klasyfikacji ICD-11 pozwoli na identyfikację problemów zdrowotnych przy użyciu aktualnego i klinicznie odpowiedniego systemu klasyfikacji. Stanom zdrowotnym lub wypadkom przypisuje się kody ICD-11, co skutkuje danymi, które mogą być wykorzystywane do opracowywania skutecznych polityk zdrowia publicznego oraz do pomiaru ich wpływu, przydzielania zasobów, poprawy leczenia, poprawy profilaktyki lub wykorzystania do rejestracji klinicznej.

Obecnie w Polsce stosowana jest klasyfikacja ICD-10 w wersji z roku 2008. Aktualizacja klasyfikacji umożliwi właściwe i zgodne z aktualną wiedzą, identyfikowanie zdarzeń medycznych.

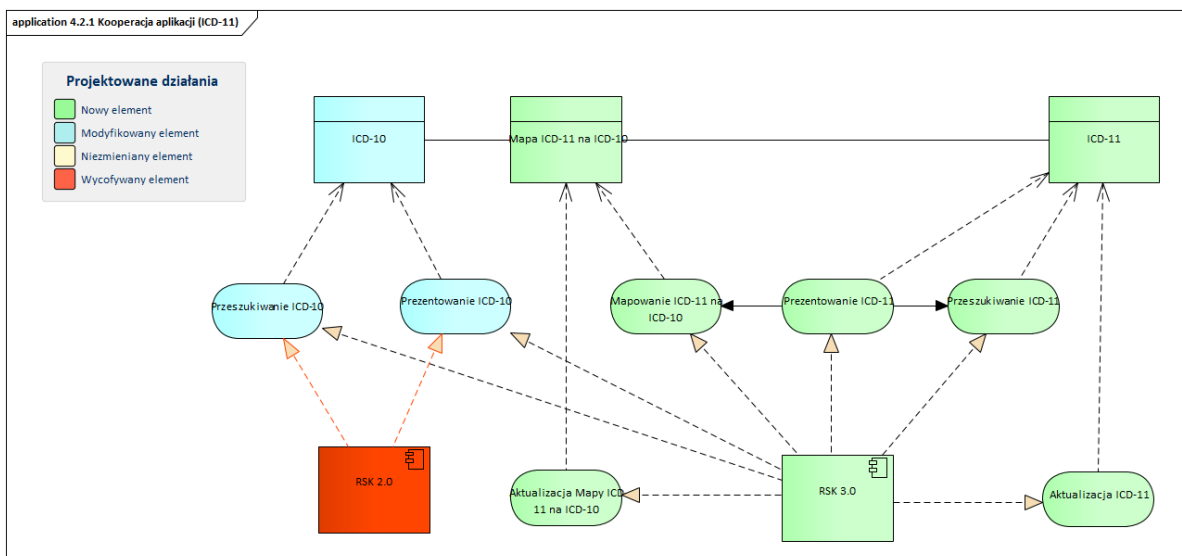
Zarys architektury

Stos technologiczny składa się z warstw:

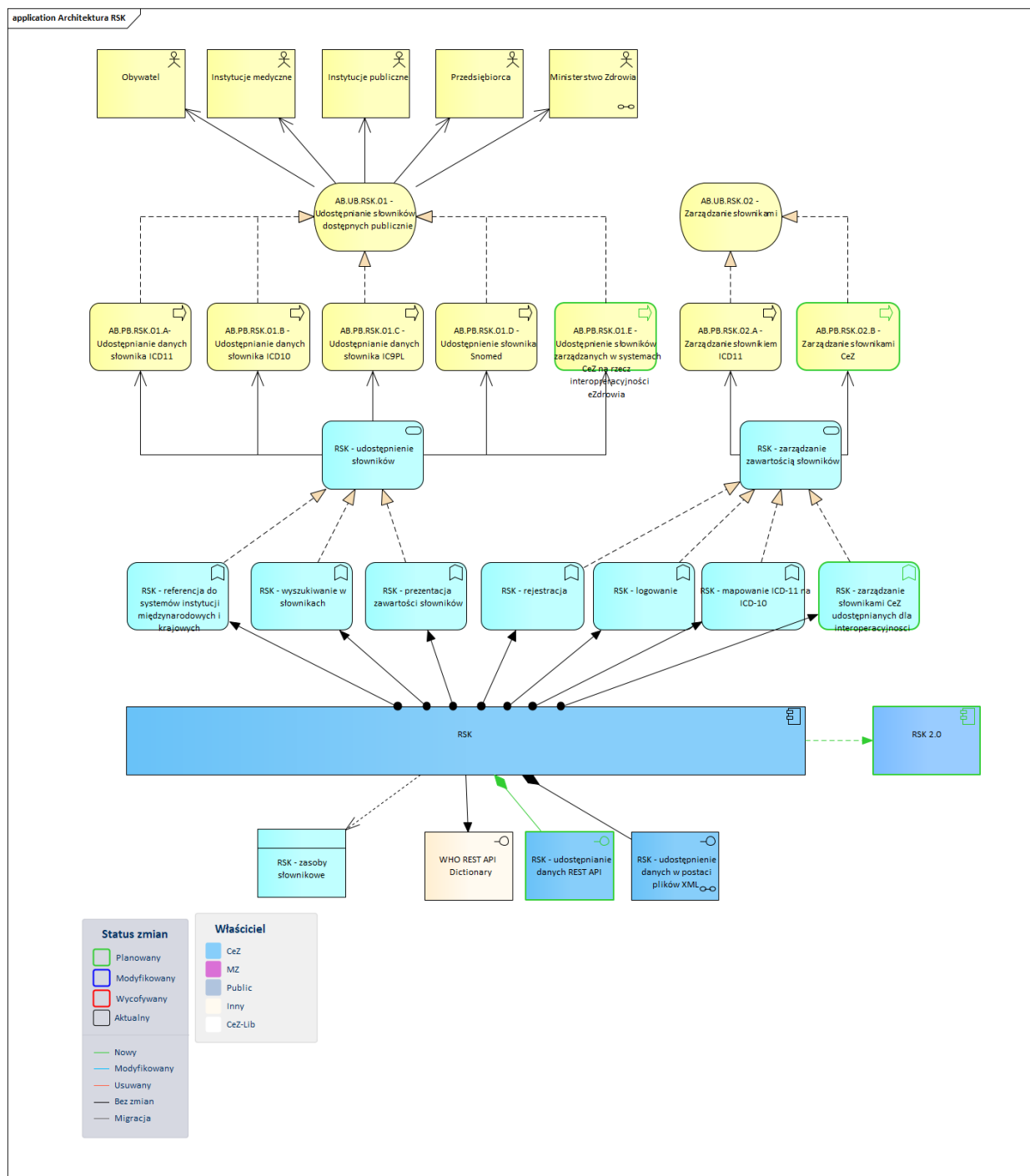
1. Warstwa danych (PostgreSQL, MongoDB)
2. Warstwa logiki biznesowej (backend: Java)
3. Warstwa prezentacji (frontend: JS + Angular)
4. Warstwa komunikacji (SFTP, REST, HL7 FHIR API, ICD-11 API)

Kooperacja aplikacji

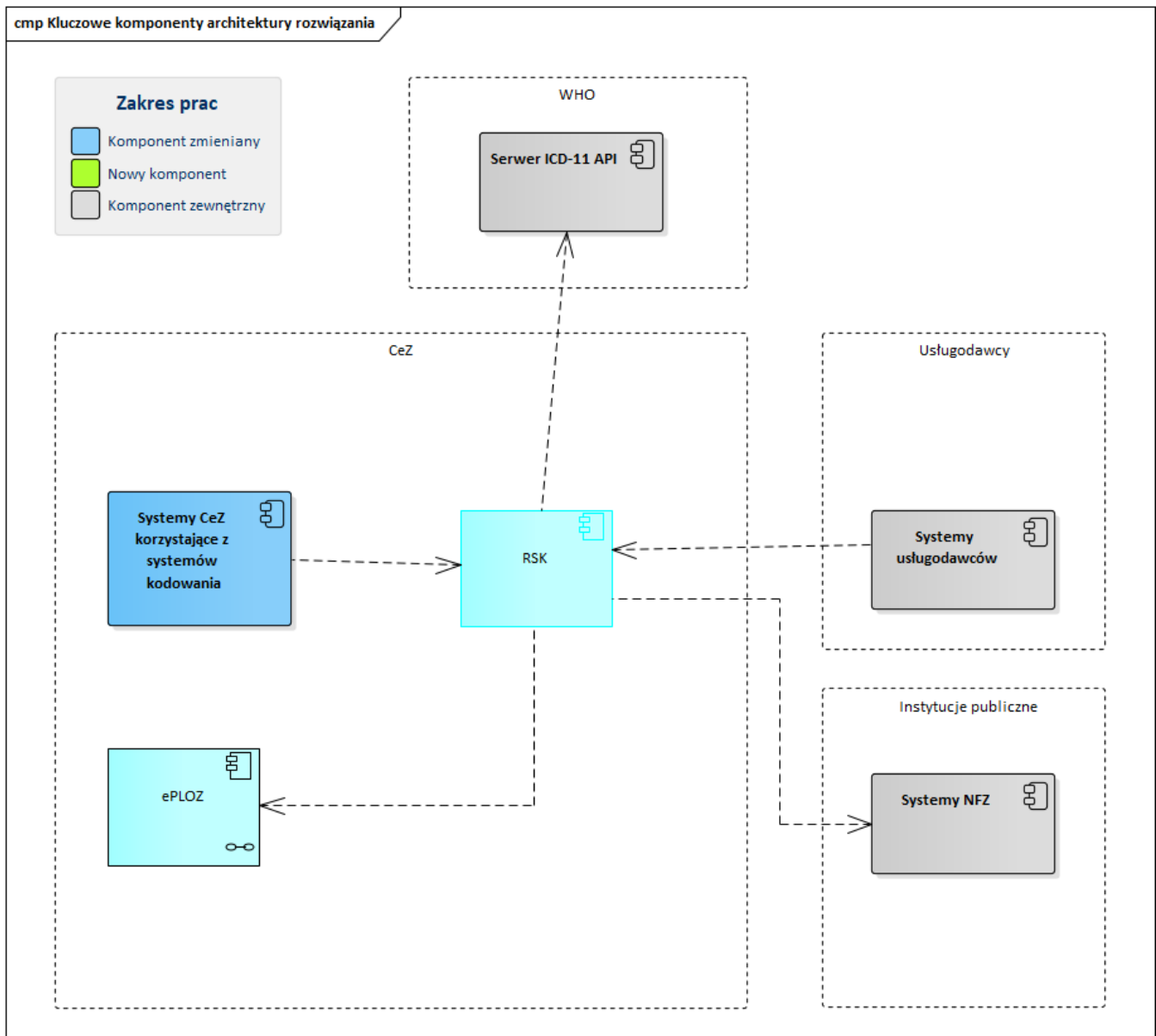
application 4.2.1 Kooperacja aplikacji (ICD-11)



Architektura RSK



Kluczowe komponenty architektury rozwiązania



Wykaz Środowisk

Infrastruktura

Serwery (lista serwerów i zainstalowanego oprogramowania wraz z numerem wersji)			
Środowisko produkcyjne	172.34.0.111 172.34.0.112 172.34.0.113	Serwer	TKG 1.13 (TANZU)
	(LB 172.32.14.6)	Serwer aplikacyjny PROD	https://rsk3.ezdrowie.gov.pl/
	10.2.11.228	Baza danych	PostgreSQL 13.5 (EnterpriseDB Advanced Server 13.5.9)
	10.2.11.41	Baza danych	MongoDB version: 5.0.6
Środowisko ewaluacyjne (przedprodukcyjne, szkoleniowe)	172.34.0.119 172.34.0.120 172.34.0.121	Serwer	TKG 1.13 (TANZU)
	(LB 172.32.14.7)	Serwer aplikacyjny EVA	https://rsk3-eva.ezdrowie.gov.pl/
	10.2.11.78	Baza danych	PostgreSQL 13.5 (EnterpriseDB Advanced Server 13.5.9)
	10.2.11.226	Baza danych	MongoDB version: 5.0.6
Środowisko testowe	172.34.0.105 172.34.0.106 172.34.0.107	Serwer	TKG 1.13 (TANZU)
	(LB 172.32.14.17)	Serwer aplikacyjny TEST	https://rsk3-test.ezdrowie.gov.pl/
	10.2.11.171	Baza danych	PostgreSQL 13.5 (EnterpriseDB Advanced Server 13.5.9)
	10.2.11.101	Baza danych	MongoDB version: 5.0.6
Środowisko deweloperskie	172.34.0.226 172.34.0.227 172.34.0.228	Serwer	TKG 1.13 (TANZU)
	(LB 172.32.14.25)	Serwer aplikacyjny DEV	https://rsk3-dev.ezdrowie.gov.pl/
	10.2.11.171	Baza danych	PostgreSQL 13.5 (EnterpriseDB Advanced Server 13.5.9)
	10.2.11.196	Baza danych	MongoDB version: 5.0.6

Zasoby dyskowe

Środowisko produkcyjne	Aplikacja: Baza danych MongoDB: 600MB / 50GB Baza danych PostgreSQL: 1,5GB / 50GB
Środowisko ewaluacyjne (przedprodukcyjne, szkoleniowe)	Aplikacja: Baza danych MongoDB: 600MB / 50GB Baza danych PostgreSQL: 1,5GB / 50GB

Środowisko testowe	Aplikacja: Baza danych MongoDB: 600MB / 50GB Baza danych PostgreSQL: 1,5GB / 50GB
Środowisko deweloperskie	Aplikacja: Baza danych MongoDB: 600MB / 50GB Baza danych PostgreSQL: 1,5GB / 50GB

3. eHemofilia oraz System Chorób Rzadkich (SChr)

Opis systemu

eHemofilia to zintegrowany system wspomagający przebieg procesu opieki nad pacjentami chorymi na hemofilię lub inne skazy krwotoczne oraz przetwarzającym dane osobowe i medyczne pacjentów. Gromadzone w eHemofilii informacje mają umożliwić nie tylko ułatwić dostęp do najważniejszych informacji o przebiegu leczenia ale także usprawnić proces zamawiania oraz dystrybucji leków dla pacjentów. Usprawnienie polega między innymi na wsparciu procesu od zapotrzebowania na lek, monitorowaniu stanu leków do samego wydania leku.

Ważną częścią systemu eHemofilia jest prowadzenie dzienniczka pacjenta – komponent umożliwia raportowanie przez pacjenta zarówno podań leków jak i ewentualnych krwawień.

Częścią systemu jest komponent odpowiedzialny za zarządzanie dostawami leków od producentów.

System Chorób Rzadkich to rejestr epidemiologiczny przetwarzający dane o pacjentach chorych na choroby rzadkie. Przetwarza on informacje o pacjentach oraz zdiagnozowanych u nich chorobach rzadkich. Istotnym komponentem systemu jest także dzienniczek pacjenta czyli dokument zawierające najważniejsze informacje o chorobie, wskazaniach oraz przeciwwskazaniach które mogą być istotne przy podejmowaniu wszelkich procedur medycznych.

Zarys architektury

Systemy posiadają wspólną architekturę. Zbudowane są głównie w architekturze mikroserwisowej, z wykorzystaniem orkiestratora konteneryzacji Kubernetes. Część serwerowa /backend/ stworzona jest z wykorzystaniem języka Java, uruchamiany jako procesy spring boot. Część interfejsu użytkownika /frontend/ stworzona jest we frameworku Angular, działająca w przeglądarce jako SPA. Spójną sesję zapewnia usługa SSO dostarczana przez system UEQZ. Kontenery systemu poza relacyjną bazą danych uruchamiane są pod kontrolą orkiestratora Kubernetes. Relacyjne bazy danych uruchamiane są na dedykowanych maszynach wirtualnych. W przypadku środowiska produkcyjnego będą to dwie niezależne maszyny wirtualne.

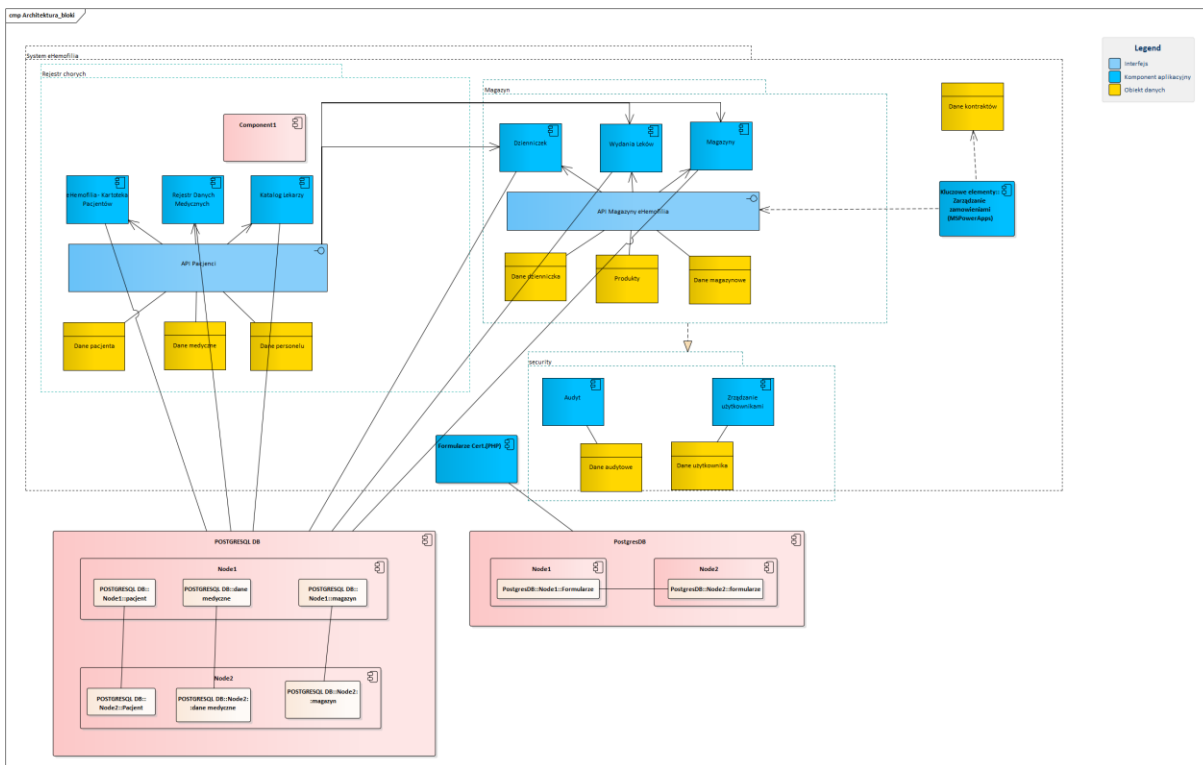
Architektura systemów eHemofilia oraz Systemu Chorób Rzadkich składa się z komponentów wymienionych i opisanych poniższej tabeli.

Komponenty architektury systemu

Nazwa elementu	Opis
API GateWay	Komponent obsługujący komunikację pomiędzy backendem systemu a frontendem oraz systemami zewnętrznymi
Formularze	Komponent obsługujący składanie formularzy o certyfikację podmiotów. Komponent realizujący wymagania biznesowe dla SChRz. Komponent w technologii PHP z własną bazą danych.

UmowyLeki	Komponent realizujący zarządzanie realizacjami umów z dostawcami leków. Komponent realizuje wymagania na potrzeby eHemofilii. Realizacja z wykorzystaniem MS PowerApps na infrastrukturze Microsoft.
Backend	Kontener procesu serwerowego systemu, udostępniającego usługi dla aplikacji zewnętrznych i wewnętrznych
EDB	Komercyjny Silnik rdbms zgodny z PostgreSQL. Baza danych systemu SMK, rozumiana jako baza produkcyjna, zawiera wszystkie obiekty bazodanowe wraz z relacjami - wynikające z modelu dziedziny. Baza danych będzie instalowana bezpośrednio na maszynach wirtualnych Vmware.

Poniższy diagram przedstawia ogólną architekturę rozwiązania SMK (model C2)

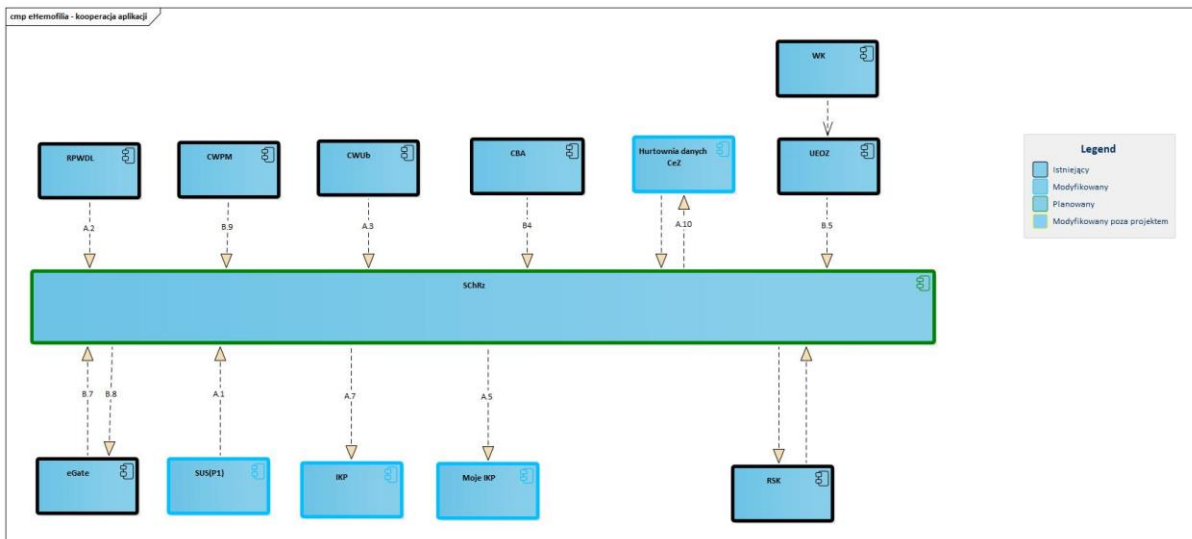


Integracje z systemami zewnętrznymi

Opisywane systemy zintegrowane będą z następującymi systemami i zewnętrznymi usługami:

Lp.	Nazwa systemu	Gestor systemu	Opis systemu	Status	Krótki opis ewentualnej zmiany
1	Rejestr Podmiotów Wykonujących Działalność Leczniczą - RPWDL	CeZ	System służący do gromadzenia danych o Podmiotach Wykonujących Działalność Leczniczą	Istniejący	
2	Centralny Wykaz Personelu Medycznego - CWPM (RPM)	CeZ	System służący do gromadzenia danych o Personelu Medycznym	Istniejący	
3	Centralny Wykaz Usługobiorców - CWUub	CeZ	System będący repliką rejestrów centralnych (w szczególności rejestru PESEL)	Istniejący	
4	Centralna Baza Adresowa (CBA)	CeZ	System obsługujący wyszukiwanie adresów i lokalizacji	Istniejący	
5	Hurtownia danych CeZ	CeZ	Platforma aplikacyjna obsługująca zadania raportowania i analizy i statystyki	Modyfikowany	Planuje się stworzyć hurtownię tematyczną chorób rzadkich w której powstawać będą wszelkie zlecane raporty





Wykaz i skalowanie środowisk

	Test1		eva		prod	
	#Worker nodes		#Worker nodes		#Worker nodes	
FE/BE host core (vCpu)	2	4	6	12	6	12
FE/BE host storage (GB)		64		512		512
FE/BE APP ram (GB)		8		24		24
PostgreSQL EDB (vCpu)	1	2	2	8	2	8
PostgreSQL EDB ram (GB)		8		128		128
PostgreSQL storage (GB)		128		512		512
		Test1		eva		prod
suma CPU		6		20		20
suma RAM		16		152		152
suma storage		192		1024		1024

4. e-Krew

Opis systemu

System e-Krew jest realizowany w ramach projektu e-Krew, którego celem bezpośrednim jest „Wsparcie publicznej służby krwi oraz nadzoru nad krwiolecznictwem w optymalnym wykorzystaniu zasobów krwi i jej składników poprzez zastosowanie nowoczesnych narzędzi teleinformatycznych”.

System „e-Krew” będzie udostępniać mechanizmy zarządzania zapasami krwi i jej produktów pochodnych, celem przewidywania niedoborów oraz wsparcie logistyki, celem optymalnej dystrybucji krwi pomiędzy Centrami oraz PWDL zgodnie z zapotrzebowaniem zgłaszanym i prognozowanym.

Uwaga: system w trakcie rozwoju. Diagram architektury technicznej, listy komponentów obrazują stan na kwiecień 2023. Dla opisu integracji systemowej zaznaczono stan poszczególnych integracji.

Zarys architektury

Usługi dedykowane dla Centrów Krwiodawstwa i Krwiolecznictwa (CKiK)

- Usługi udostępniane dla pracowników centrów za pomocą połączenia VPN
- Usługi udostępniane za pomocą
 - Aplikacja Angular dla użytkowników (pracowników) CKiK
 - Współdzielona kolejka RabbitMQ na potrzeby integracji urządzeń w centrach (np. diagnostyka, preparatyka krwi)
 - REST API na potrzeby innych wybranych urządzeń w centrach

Usługi dla Podmiotów Wykonujących Działalność Leczniczą (Szpitali)

- Usługi udostępniane w formie REST API za pośrednictwem szyny danych P1
- Szpital korzysta z usług za pośrednictwem lokalnego systemu HIS lub aplikacji gabinet.gov
- Usługi w trakcie rozwoju

Usługi dla obywatela

- Usługi udostępniane w formie REST API za pośrednictwem szyny danych P1
- Obywatel korzysta z usług za pośrednictwem aplikacji IKP
- Usługi w trakcie rozwoju

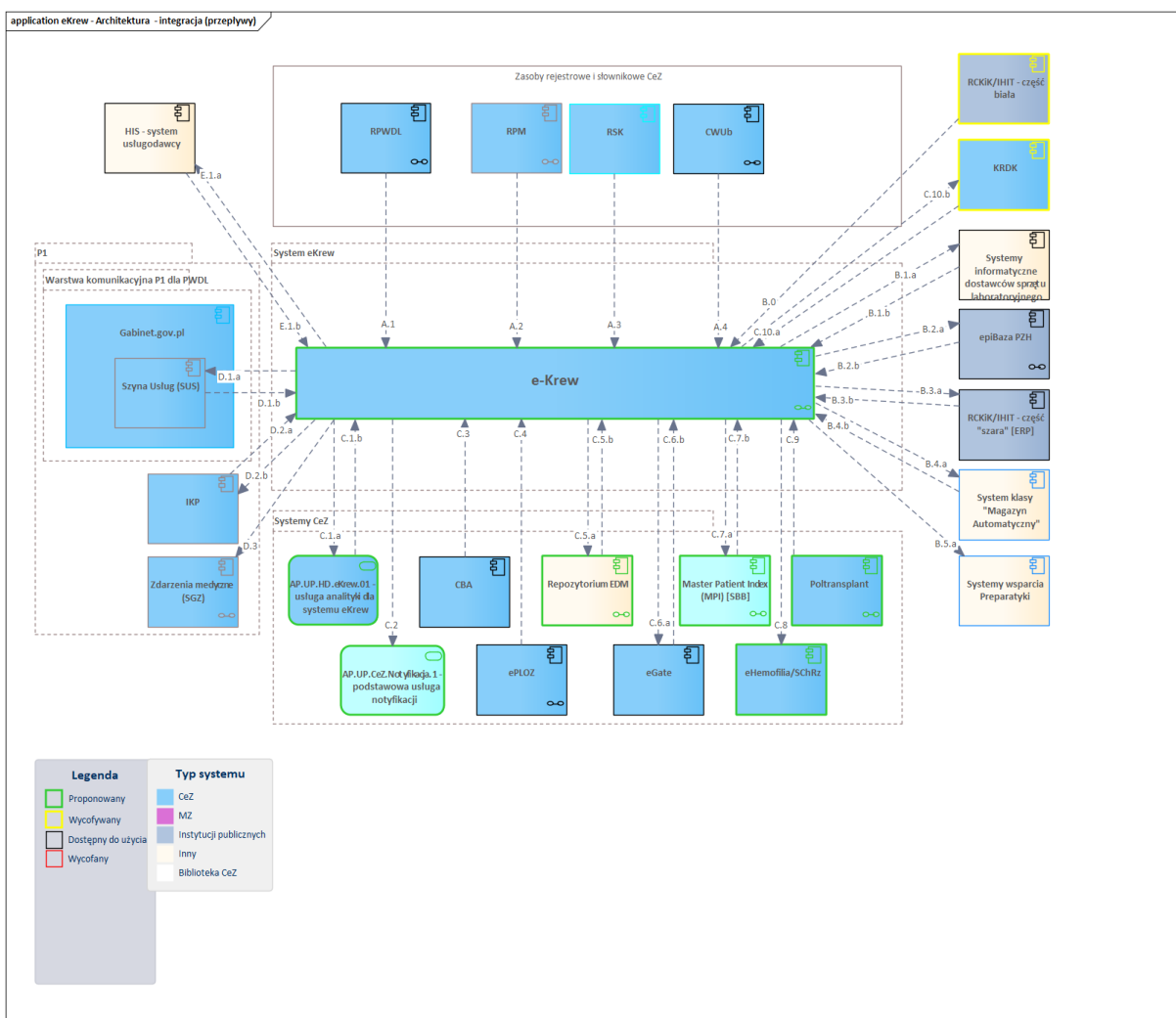
Architektura aplikacji

Aplikacja jest oparta o komponenty zlokalizowane w środowisku VMWare Tanzu. Poszczególne komponenty są zlokalizowane na klastrze Kubernetes lub na dedykowanych maszynach wirtualnych.



ekrew-mongodb	Dokumentowa baza danych MongoDB na potrzeby przechowywania komunikatów od urzędów medycznych
ekrew-edb-migracja	Relacyjna baza danych na potrzeby obsługi migracji danych CKiK'ów w ramach wdrożenia do systemu e-Krew. Oparta o Postgres EDB
kibana	Obsługa logów aplikacyjnych
filebeat	Obsługa logów aplikacyjnych
elastic-search	Obsługa logów aplikacyjnych
grafana	Obsługa monitoringu aplikacji
prometheus	Obsługa monitoringu aplikacji

Integracje z systemami zewnętrznymi



Lp.	System źródłowy	System docelowy	Zakres wymienianych danych	Sposób wymiany danych	Typ interfejsu	Status
A.1	RPWDL	e-Krew	· Dane podmiotów medycznych	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi	Planowany

					sieciowe (Web service) oparte o REST.	
A.2	CWPM (RPM)	e-Krew	<ul style="list-style-type: none"> · Dane personelu medycznego · Specjalizacje personelu 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST.	Planowany
A.3	Rejestr Systemów Kodowania	e-Krew	<ul style="list-style-type: none"> · ICD10 · ICD9 · TERYT · Kody Pocztowe 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST albo o zasilanie z plików płaskich	Planowany
A.4	CWUb	e-Krew	<ul style="list-style-type: none"> · Dane osobowe · Dane adresowe · Status osoby (żyje, nie żyje) 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST.	Istniejący
B.1.a	e-Krew	Systemy informatyczne dostawców sprzętu laboratoryjnego	<ul style="list-style-type: none"> · Dane osobowe · Rodzaj badania · Dane zlecającego (personel, PWDL) 	Wywołanie metody w interfejsu API urządzenia (o ile dostępne)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST. Interfejsy programistyczne API służące do wymiany informacji z	Istniejący

					innymi systemami zostaną zaimplementowane z wykorzystaniem następujących standardów: HTTP/S, SOAP 1.1, XML 1.0, WSDL 1.1. Alternatywnie inna metoda komunikacji zgodna z specyfikacją urzędnika objętego zakresem wdrożenia systemu	
B.1.b	Systemy informatyczne dostawców sprzętu laboratoryjnego	e-Krew	<ul style="list-style-type: none"> · Dane osobowe, dane wrażliwe o stanie zdrowia, stanowiące dokumentację medyczną · Dane laboranta (personel) 	Wg KRI: tryb odwołań bezpośrednich (§13 ust. 2)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST.	Istniejący
B.2.a	e-Krew	PZH	<ul style="list-style-type: none"> · dane osobowe, dane adresowe, dane kontaktowe · dane wrażliwe o stanie zdrowia, stanowiące dokumentację medyczną · personel · szczegóły zdarzenia epidemiologicznego 	Wywołanie usługi systemu epiBaza	REST API	Planowany
B.2.b	PZH	e-Krew	<ul style="list-style-type: none"> · dane identyfikacyjne · dane wrażliwe o stanie zdrowia 	Wywołanie usługi systemu e-Krew	REST API	



B.3.a	e-Krew	Systemy wspomagające CKiK i IHiT (część „szara”)	<ul style="list-style-type: none"> · dane osobowe, · dane wrażliwe o stanie zdrowia, stanowiące dokumentację medyczną · stany magazynowe · zlecenia 	Wg KRI: tryb odwołań bezpośrednich (§13 ust. 2)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP.	Planowany
B.3.b	Systemy wspomagające CKiK i IHiT (część „szara”)	e-Krew	<ul style="list-style-type: none"> · Dane osobowe, · dane wrażliwe o stanie zdrowia, stanowiące dokumentację medyczną · zlecenia · stany magazynowe 	Wg KRI: tryb odwołań bezpośrednich (§13 ust. 2)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP.	Planowany
B.4.a	e-Krew	System klasy "Magazyn Automatyczny" (różni dostawcy)	zlecenie wydania z magazynu cechy składnika krwi	Wywołanie usługi systemu e-Krew	REST API	Planowany
B.4.b	System klasy "Magazyn Automatyczny" (różni dostawcy)	e-Krew	potwierdzenie przyjęcia na magazyn potwierdzenie wydania z magazynu potwierdzenie aktualizacji pozycji w magazynie	Wywołanie usługi systemu e-Krew	REST API	Planowany
B.5.a	e-Krew	Systemy wsparcia Preparatyki		Wywołanie usługi systemu e-Krew	REST API	Planowany
C.1.a	System eKrew	Hurtownia CeZ	<ul style="list-style-type: none"> a. dane osobowe, b. dane wrażliwe o stanie zdrowia, stanowiące dokumentację medyczną c. struktura działalności d. donacje i pobrania e. stany magazynowe, dokumenty WZ, PZ 	<ul style="list-style-type: none"> 1. Bezpośrednia wymiana danych (zasilanie ETL) 2. Zasilanie zdarzenie 	Bazodany (ETL) Zdarzenowy – kolejkowy (Apache Kafka, Rabbit MQ)	Planowany



			f. kartoteki (urządzeń, stacji, personelu) aktywności			
C.1.b	Hurtownia CeZ	e-Krew	<ul style="list-style-type: none"> · dane zagregowane · Komplet danych w wskazanym w p.1 w postaci raportów i analiz 	Publikacja agregatów danych zgodnie z harmonogramem Dostęp do raportów i analiz	Kolejkowy asynchroniczny (Apache Kafka, Rabbit MQ). Wymiana plikowa. Dostęp bezpośredni do raportów i analiz (integracja frontowa rozwiązań)	Planowany
C.2	e-Krew	Usługi infrastrukturalne CeZ	<ul style="list-style-type: none"> · Dane do notyfikacji (kontakt, informacje) 	Wywołanie usługi	Kolejkowy asynchroniczny (Apache Kafka, Rabbit MQ), API REST	Planowany
C.3	CBA	e-Krew	Dane adresowe	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST	Planowany
C.4	e-PLOZ	e-Krew	Dane uwierzytelnionego konta	Bezpośrednia integracja (API)	Interfejs WebService	Istniejący
C.5.a	e-Krew	Repozytorium Elektronicznej Dokumentacji Medycznej EDM	<ul style="list-style-type: none"> · Dane identyfikacyjne (MPI_ID) · Dane wrażliwe (dokument medyczny – wynik badania) 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi oparte o REST lub zgodnie z standardem IHE	Planowany
C.5.b	Repozytorium Elektroniczne	e-Krew	<ul style="list-style-type: none"> · Dane identyfikacyjne (MPI_ID) 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi	Planowany



	Dokumentacja i Medycznej		<ul style="list-style-type: none"> Dane wrażliwe (dokument medyczny – wynik badania) 		oparte o REST lub zgodnie z standardem IHE	
C.6.a	e-Krew	eGate	<ul style="list-style-type: none"> Dokument do podpisu 	Integracja biblioteczna (użycie komponentu)	Standardy rynkowe (podpis dokumentów w różnym typem podpisu) Standardy rynkowe	Planowany
C.6.b	eGate	e-Krew	<ul style="list-style-type: none"> Podpisany dokument 			
C.7.a	e-Krew	Master Patient Index (MPI)	<ul style="list-style-type: none"> Dane osobowe Dane adresowe Dane opiekuna Zgody 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST	Planowany
C.7.b	Master Patient Index (MPI)	e-Krew	<ul style="list-style-type: none"> Dane osobowe Dane adresowe Dane opiekuna Zgody 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST	Planowany
C.8	e-Krew	Hemofilia	<ul style="list-style-type: none"> dane o stanach magazynowych 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST	Planowany
C.9	Poltransplant	e-Krew	<ul style="list-style-type: none"> dyskwalifikacje, słowniki 	Bezpośrednia integracja (API)	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST	Planowany



D.1.a (E.1.a)	e-Krew	Gabinet.gov.pl (system PWDL z wykorzystaniem Szyny Integracyjnej P1)	<ul style="list-style-type: none"> · Dane identyfikacyjne osób · Wydania · Statusy spraw · Blokady · Wyniki konsultacji · Dane wrażliwe (dokument medyczny) · Dane osobowe 	Szyna P1	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST zgodnie z obowiązującym standardem P1.	Planowany
E.1.b (D.1.b)	System PWDL (z wykorzystaniem Szyny Integracyjnej P1 w ramach Gabinet.gov.pl)	e-Krew	<ul style="list-style-type: none"> · Dane osobowe · Dane adresowe · Dane kontaktowe · Dane opiekuna · Dane wrażliwe · Pobranie · Zamówienie · Reklamacja · Niepożądany odczyn pobraniowy · Zwrot · Zamówienie konsultacji 	Szyna P1	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST zgodnie z obowiązującym standardem P1.	Planowany
D.2.a	e-Krew	IKP	<ul style="list-style-type: none"> · Dane identyfikacyjne osób · Donacje · Statusy wniosków · Decyzje · Zaświadczenia · Notyfikacje 	Szyna P1 / bezpośrednia integracja API	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST zgodnie z obowiązującym standardem tworzenia rozwiązań na rzecz IKP.	Planowany
D.2.b	IKP	e-Krew	<ul style="list-style-type: none"> · Dane identyfikacyjne osób · Wnioski · Rejestracja 	Szyna P1 / bezpośrednia	Interfejsy będą zrealizowane jako usługi	Planowany

				integracja API	sieciowe (Web service) oparte o SOAP lub REST zgodnie z obowiązującym standardem tworzenia rozwiązań na rzecz IKP.	
D.3	e-Krew	P1 – Zdarzenia medyczne	<ul style="list-style-type: none"> · Dane osobowe · Dane wrażliwe 	Bezpośrednia integracja zgodnie z standardem P1	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o SOAP lub REST – zgodnie z aktualnie obowiązującą specyfikacją.	Planowany

Wykaz i skalowanie środowisk

Serwery bazodanowe

nazwa hosta	ilość cpu	ilość ram	wielkość hdd
ekrew-edb-dev	4	8GB	500GB
ekrew-edb-bufor	2	2GB	50GB
ekrew-edb-testaut	2	2GB	50GB
ekrew-edb-test2	8	16GB	300GB
ekrew-edb-test3	8	16GB	550GB
ekrew-edb-uat	2	2GB	300GB
ekrew-edb-pre	8	32GB	50GB
ekrew-edb-kwalifikacja	2	2GB	50GB
ekrew-edb-prod	8	32GB	200GB
ekrew-edb-migracja	4	16GB	6.0TB
ekrew-edb-krdk-migr	2	2GB	20GB

Serwery narzędziowe

nazwa hosta	ilość cpu	ilość ram	wielkość hdd
ekrew-tools-dev	8	8GB	100GB
ekrew-tools-test2	4	8GB	50GB
ekrew-tools-uat	4	8GB	50GB
ekrew-tools-pre	4	8GB	50GB
ekrew-tools-kwalifikacja	2	2GB	50GB
ekrew-tools-prod	4	8GB	50GB
ekrew-swarm-a	2	4GB	50GB
ekrew-swarm-b	2	4GB	50GB
ekrew-rabbitmq-prod-1	4	4GB	50GB
ekrew-rabbitmq-prod-2	2	2GB	50GB
ekrew-rabbitmq-prod-3	2	2GB	50GB
ekrew-mongodb-prod	4	8GB	50GB
ekrew-mongodb-dev	4	8GB	50GB

Serwery CI/CD

nazwa hosta	ilość cpu	ilość ram	wielkość hdd
ekrew-jenkins-master	8	16GB	50GB
ekrew-jenkins-slave1	24	64GB	2GB
ekrew-jenkins-slave2	24	64GB	2GB
ekrew-jenkins-slave3	24	64GB	2GB
ekrew-jenkins-slave4	24	64GB	2GB
ekrew-jenkins-slave7	24	64GB	2GB
ekrew-jenkins-slave8	4	8GB	2GB
ekrew-jenkins-slave11	8	32GB	50GB
ekrew-jenkins-slave12	8	32GB	50GB
ekrew-jenkins-slave13	8	32GB	50GB
ekrew-jenkins-slave14	8	32GB	50GB
ekrew-jenkins-slave15	8	32GB	50GB
ekrew-jenkins-slave16	8	32GB	50GB
ekrew-jenkins-slave17	8	32GB	50GB



ekrew-jenkins-slave18	8	32GB	50GB
ekrew-jenkins-slave19	8	32GB	50GB
ekrew-jenkins-slave20	8	32GB	50GB

Klastry kubernetes

Nazwa klastra	Zasoby
DEV klaster	6 workery (8 CPU, 16GB RAM)
TEST2 klaster	4 workery (8 CPU, 8GB RAM)
UAT klaster	2 workery (8 CPU, 8GB RAM)
PRE klaster	2 workery (8 CPU, 8GB RAM)
PROD klaster	8 workery (8 CPU, 8GB RAM)

5. Poltransplant

Opis systemu

System Poltransplant ma na celu wsparcie procesów koordynowania i nadzoru nad działaniami związanymi z pobieraniem, przechowywaniem i przeszczepianiem narządów, tkanek i komórek.

System ma umożliwiać:

- informatyzację procesów biznesowych koordynowania przeprowadzanych czynności transplantacyjnych,
- prowadzenie Rejestrów Transplantacyjnych,
- korzystanie z narzędzi i algorytmów służących do alokacji narządów i doboru dawców,
- zapewnienie bezpieczeństwa dostępu przechowywanych danych.

Zarys architektury

W ramach projektu zostały użyte technologie i frameworki

- JDK 11
- Spring Boot
- Keycloak
- PostgreSQL
- Liquidbase
- Swagger
- Angular
- Angular Material
- Typescript
- RabbitMQ

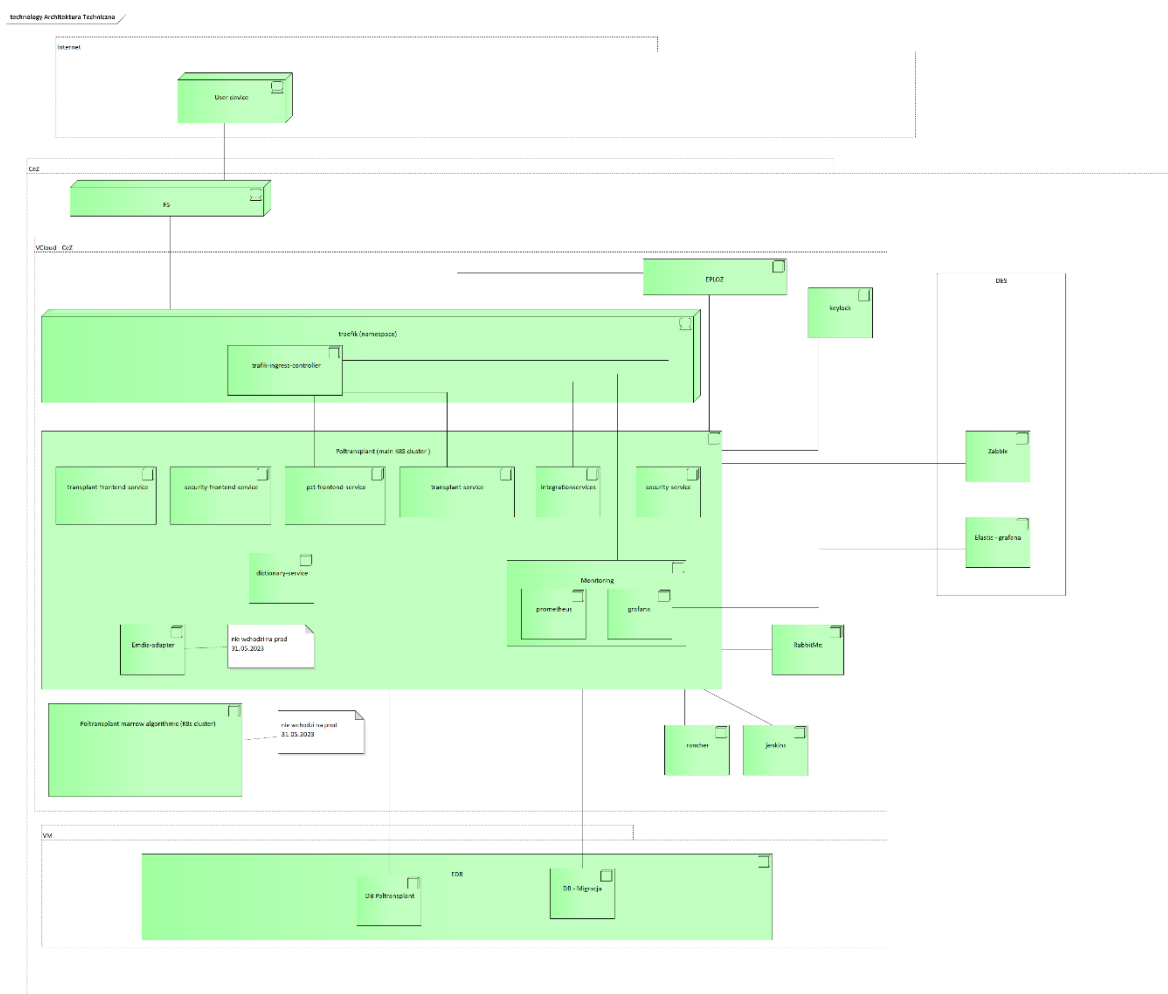
Część serwerowa /backend/ stworzona jest z wykorzystaniem języka Java, uruchamiany jako procesy spring boot. Część interfejsu użytkownika /frontend/ stworzona jest we frameworku Angular, działająca w przeglądarce jako SPA. Relacyjne bazy danych uruchamiane są na dedykowanych maszynach wirtualnych. W przypadku środowiska produkcyjnego będą to dwie niezależne maszyny wirtualne.

Architektura systemu Poltransplant składa się z komponentów wymienionych i opisanych poniższej tabeli.

Komponenty architektury systemu

Nazwa elementu	Opis
Transplant-service	Główny komponent (backend) odpowiedzialny za obsługę procesów, karty pacjenta, zdarzeń, komunikacje oraz kwestię załączników.
Dictionary-service	Usługa do wystawiania danych słownikowych.
Security-service	Zarządzanie uprawnieniami i użytkownikami w systemie. Propagacja informacji do IP (Keycloak) Serwis odpowiedzialny za tworzenie użytkowników w etransplant Keycloak oraz w Eploz Keycloak oraz odpowiednie mapowanie ich między nimi. Zarządzą uprawnieniami, rola, funkcjami użytkowników przypisanych w poszczególnych ośrodkach oraz słownikami.
Authorization-server (Keycloak)	Napisany dedykowany moduł pozwalający na odzwierciedlenie struktury drzewiastej ról użytkowników, obsługujący autoryzację dla keycloak
Transplant-frontend-service	Frontend dla procesów translologicznych w zakresie narzędzi oraz szpiku
Security-frontend-service	Front dedykowana pod administrację oraz zarządzanie (słownikami, danymi w zakresie ośrodków, uprawnień)
Pzt-frontend-service	Front dedykowana pod portal Pracownie Zgodności
EMDIS-Adpater	Serwis do pobierania danych z EMDIS-a. Obecnie realizowany procesem pobierania maili, mapowania na model dany zgodnych z naszą strukturą oraz wrzucania na kolejkę rabbitmq w celu zasilenia algorytmów szpikowych.
IntegrationServices	Usługa dla poltransplant do integracji zewnętrznych: - pobieranie zgonów, - integracja z P1 (PESEL) - eGate - słowniki zewnętrzne (RSK)
Haplomat	Maszyna w infrastrukturze (system WINDOWS SERVER 2019) z bardzo dużym zasobem RAM. Dedykowana do uruchamiania na żądanie skryptów
Rabbitmq	System kolejkowy dla rozwiązania.
Postgresql	Baza relacyjna rozwiązania.
MarrowAlgorytm	Grupa rozwiązania do uruchamiania algorytmów szpikowych
Kibana	Obsługa logów aplikacyjnych
Filebeat	Obsługa logów aplikacyjnych
Elastic-search	Obsługa logów aplikacyjnych
Grafana	Obsługa monitoringu aplikacji
Prometheus	Obsługa monitoringu aplikacji

Poniższy diagram przedstawia ogólną architekturę rozwiązania Poltransplant

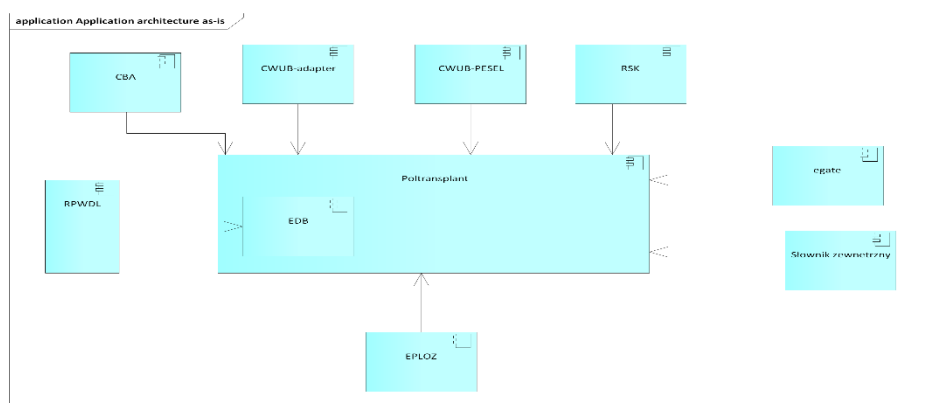


Integracje z systemami zewnętrznymi

System SMK zintegrowany będzie z następującymi systemami i zewnętrznymi usługami:

Lp.	Nazwa systemu / usługi	Właściciel systemu / usługi	Opis
1	Elektroniczna Platforma Logowania Ochrony Zdrowia – e-PLOZ	Centrum e-Zdrowia	System świadczący usługi uwierzytelniania i autoryzacji użytkowników.
2	RSK	Centrum e-Zdrowia	Usługa słowników. (pobieranie słowników w zakresie ICD9, ICD10). Usługa RESTAPI.
3	CWUB-adapter	Centrum e-Zdrowia	Pobieranie aktualizacji danych pacjenta (zgodny, aktualizacja danych personalnych). Usługa po RESTAPI.
4	CWUB (p1 pesel)	Centrum e-Zdrowia	Pobieranie danych po nr Pesel. Usługa pobierania po SOAP.
5	Npdm.org	Usługa zewnętrzna	Usługa słowników medycznych potrzebnych w zakresie transplantologii oraz przeszczepu krwi. Pobieranie po RESTAPI
6	CBA	Centrum e-Zdrowia	Centralna baza adresowa – adresy. Usługa po RestAPI.

7	E-Gate	Centrum e-Zdrowia	Wymiana danych w zakresie podpisów elektronicznych. Usługa SOAP.
8	RPWDL (P1)	Centrum e-Zdrowia	Replikacja danych z bazy P1 – sor. Replikacja DB → DB.



Wykaz i skalowanie środowisk

Skalowanie środowisk

(obecnie środowisko produkcyjne jest w trakcie tworzenia).

Serwery baz danych			
Nazwa	Ilość CPU	Ilość RAM	Wielkość dysku w GB
EDB Dev	4	12	95
EDB Test	4	12	95
EDB Pre	4	12	570
EDB Pre-Mig	4	12	645
Serwery RabbitMQ			
Nazwa	Ilość CPU	Ilość RAM	Wielkość dysku w GB
Rabbit Dev	4	12	45
Rabbit Test	4	12	45
Rabbit Pre	4	12	45
Serwery narzędziowe			
Nazwa	Ilość CPU	Ilość RAM	Wielkość dysku w GB
Sonar & Nexus	2	6	20
Sonar & Nexus	4	12	20
Sonar & Nexus	2	6	20
Bastion	4	12	45
Serwery CI/CD			

Nazwa	Ilość CPU	Ilość RAM	Wielkość dysku w GB
Jenkins	4	12	545
Jenkins worker	8	32	45
Jenkins worker	8	32	45
Jenkins worker	8	32	45
Jenkins worker	8	32	45
Klasy kubernetes			
Nazwa	Ilość workerów		
poltransplant-dev	8x (8core/8ram)		
poltransplant-test	12x 8core/8ram)		
poltransplant-pre	19x (8core/8ram)		
poltransplant-marrow-test	12x (8core/80ram)		
poltransplant-eva	6x (8core/12ram)		
Środowiska Marrow			
Nazwa	Ilość CPU	Ilość RAM	Wielkość dysku w GB
Marrow ab&cbu dev	4	50	45
Marrow donors dev	8	80	45
Marrow ab&cbu test	4	50	115
Marrow donors test	8	80	45
Marrow ab&cbu test-at	4	50	45
Marrow donors test-at	8	80	145
Marrow ab&cbu pre	4	50	45
Marrow donors pre	8	80	145
Marrow ab&cbu pre-fix	4	50	45
Marrow donors pre-fix	8	80	45
Marrow ab&cbu pre-mig	4	50	45
Marrow donors pre-mig	8	80	65
Marrow ab&cbu pre-biz	4	50	45
Marrow donors pre-biz	8	80	95

Informacje dotyczące zakładanego środowiska PROD:

Fronted:	#nodes	Prod
frontend host core (vCpu)	8	4
frontend host ram (GB)	8	16
frontend host storage (GB)	8	64
frontend host core (vCpu)	8	4
frontend host ram (GB)	8	16
frontend host storage (GB)	8	64
frontend host core (vCpu)	8	4
frontend host ram (GB)	8	16

frontend host storage (GB)	8	64
backend APP core (vCpu)	8	4
backend APP ram (GB)	8	32
backend APP storage (GB)	8	64
backend APP core (vCpu)	8	4
backend APP ram (GB)	8	32
backend APP storage (GB)	8	64
backend APP core (vCpu)	8	4
backend APP ram (GB)	8	32
backend APP storage (GB)	8	64
PostgreSQL EDB (vCpu)	2	8
PostgreSQL EDB ram (GB)	2	64
PostgreSQL storage (GB)	2	512
Szpik core (vCpu)	1	8
Szpik ram (GB)	1	128
Szpik storage (GB)	1	72
RabbitMQ core (vCpu)	2	3
RabbitMQ ram (GB)	2	16
RabbitMQ storage (GB)	2	250
Adapter EMDIS core (vCPU)	3	4
Adapter EMDIS ram (GB)	3	8
Adapter EMDIS storage (GB)	3	32
Inne vCPU	1	32
inne RAM GB	1	64
Inne storage GB	1	1000
Haplomat (vCPU)	1	8
Haplomat ram(GB)	1	32
Haplomat storage (GB)	1	256
		prod
suma CPU		87
suma RAM		456
suma storage		2506

6. Rejestr Podmiotów Wykonujących Działalność Leczniczą (RPWDL 2.0)

Opis systemu

System odpowiedzialny za Rejestr Podmiotów Wykonujących Działalność Leczniczą w wszystkich aspektach funkcjonalnych począwszy od obsługi wniosków rejestrowych/wniosków o zmianę poprzez przetworzenie a następnie udostępnienie danych zarówno dla systemów CeZ jak i powszechnie dostępnych. Forma udostępnienia danych aktualnie wykorzystuje pliki płaskie/xml jak i interfejsy wewnętrzne w CeZ. Przewiduje się (w najbliższym okresie) udostępnienie API REST dla podmiotów/osób zarejestrowanych. Zakłada się, z powodu urealnienia działania RPWDL w zakresie usług interoperacyjnych Kraju, bardzo wysoką dostępność dla usługi weryfikacji danych podmiotu w rejestrze oraz wysoką dla funkcjonalności wnioskowania. RPWDL jest zintegrowany z systemami CeZ (pobranie



danych, udostępnienie danych, uwierzytelnienie, podpis) jak i zewnętrznymi (planowane wdrożenie systemu dla ePłatności).

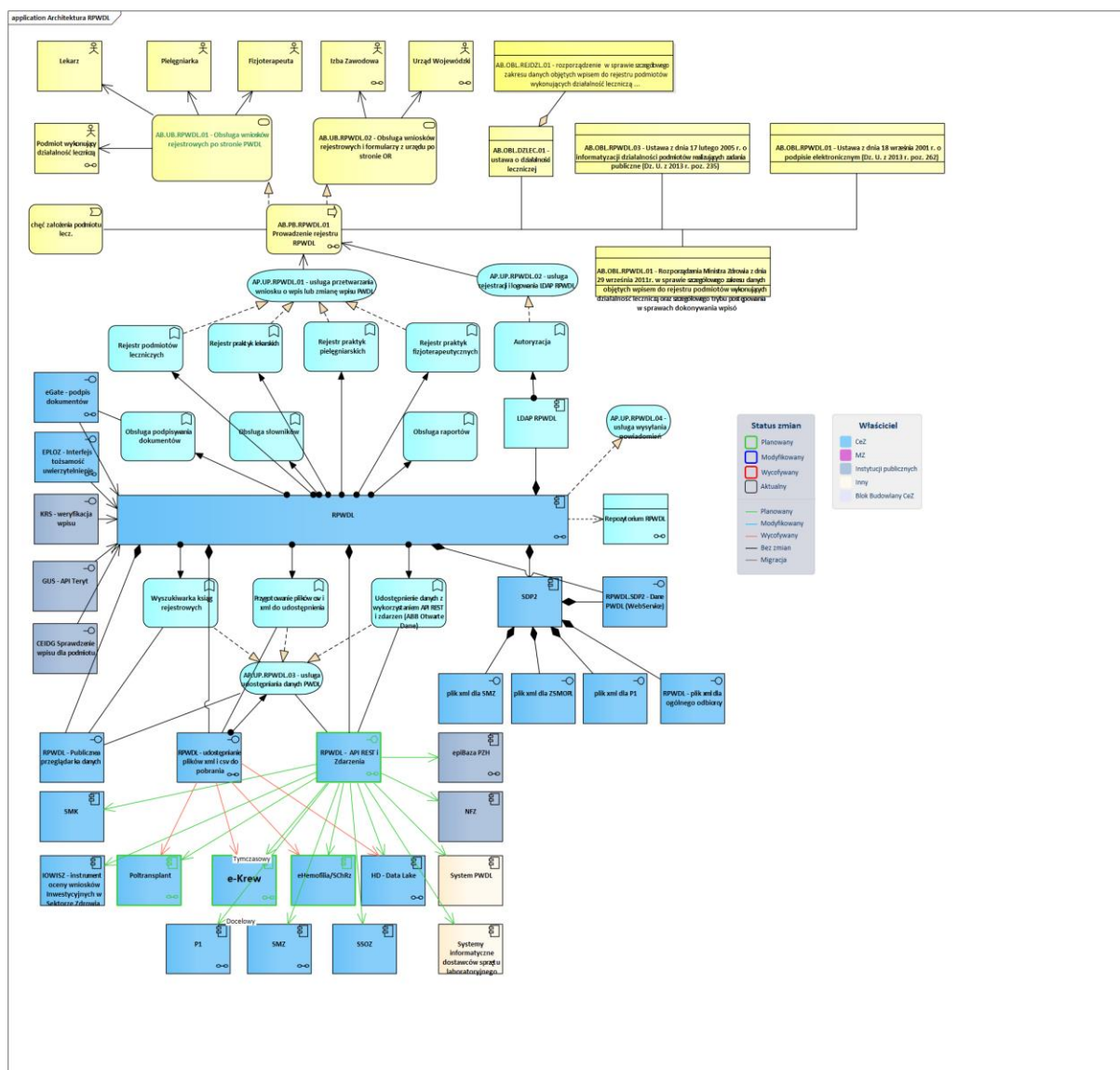
RPWDL 2.0 jest nową wersją bazowego RPWDL 1.0 z przebudową architektury rozwiązania oraz uwzględnieniem potrzeb biznesowych.

Zarys architektury

System jest zbudowany w architekturze mikroserwisowej w oparciu o konteneryzację Kubernetes.

Kontenery systemu (z wyłączeniem baz danych) uruchamiane są pod kontrolą orkiestratora Kubernetes. Bazy danych uruchamiane są na dedykowanych maszynach wirtualnych.

Diagram ogólny architektury

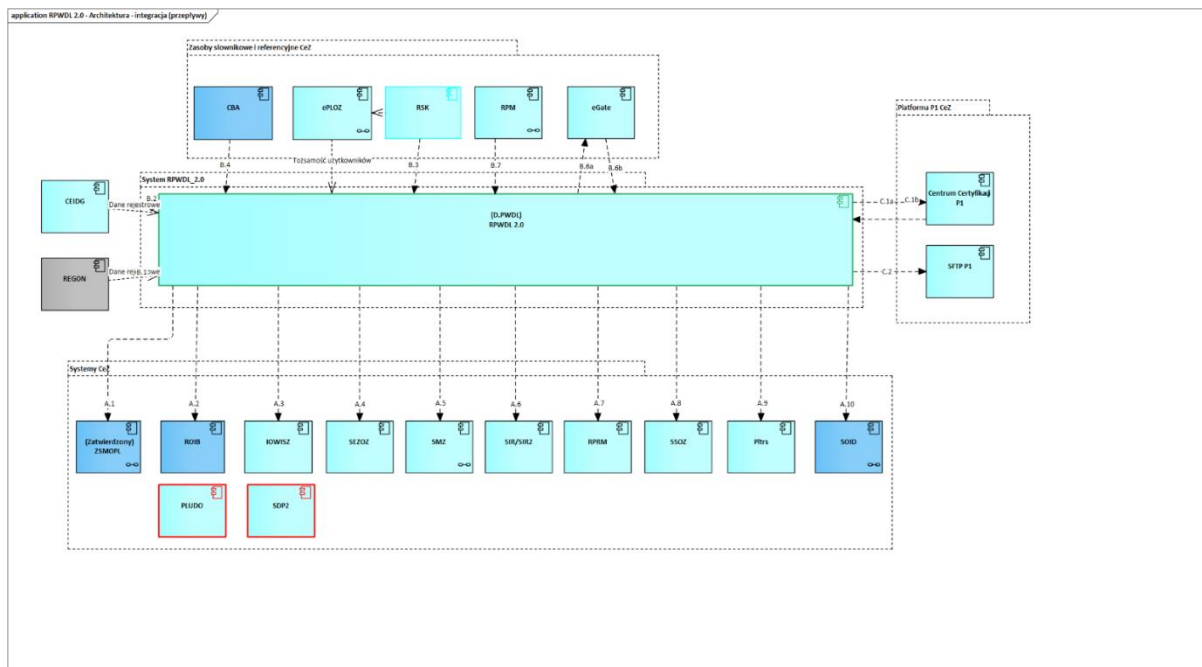


Integracje z systemami zewnętrznymi

Lp.	Nazwa systemu	Gestor systemu	Opis systemu	Status	Krótki opis ewentualnej zmiany
1	Rejestr podmiotów wykonujących działalność leczniczą – RPWDL	CeZ	System służący do gromadzenia danych o podmiotach wykonujących działalność leczniczą.	Istniejący/W trakcie realizacji (przepisanie)	System będzie rozbudowany z uwzględnieniem wprowadzenia wzorca architektonicznego dla Rejestrów CeZ w modelu interoperacyjnym zarówno dla rozwiązań wewnętrznych CeZ jak i zewnętrznych (model otwarty)
2	Centralny Wykaz Personelu Medycznego - CWPM (RPM)	CeZ	System służący do gromadzenia danych o personelu medycznym.	Istniejący / Rozbudowany	System będzie rozbudowany z uwzględnieniem wprowadzenia wzorca architektonicznego dla Rejestrów CeZ w modelu interoperacyjnym zarówno dla rozwiązań wewnętrznych CeZ jak i zewnętrznych (np.: system NFZ, system PWDL)
3	Centralny Wykaz Usługobiorców – CWUob	CeZ	System będący repliką rejestrów centralnych (w szczególności rejestru PESEL)	Istniejący / Rozbudowany	System będzie rozbudowany z uwzględnieniem wprowadzenia wzorca architektonicznego dla Rejestrów CeZ w modelu interoperacyjnym (dla rozwiązań wewnętrznych CeZ)
4	Rejestr Systemów Kodowania	CeZ	System zawierający współdzielone słowniki	Istniejący / Rozbudowany	System będzie rozbudowany z uwzględnieniem wprowadzenia wzorca architektonicznego dla Rejestrów CeZ w modelu interoperacyjnym (dla rozwiązań wewnętrznych i zewnętrznych CeZ)
5	Centralna Baza Adresowa (CBA)	CeZ	System obsługujący wyszukiwanie adresów i lokalizacji	Istniejący / Rozbudowany	System zostanie zmodernizowany pod kątem wymagań projektów CeZ (w tym RPWDL)
6	Hurtownia danych CeZ	CeZ	Platforma aplikacyjna obsługująca zadania raportowania, analizy i statystyki	Istniejący / Rozbudowany	Planuje się stworzyć hurtownię tematyczną RPWDL w której powstawać będą wszelkie raporty zlecane przez Właściciela Biznesowego Systemu RPWDL
7	Elektroniczna Platforma Logowania Ochrony Zdrowia (ePLOZ)	CeZ	Platforma do logowania Usług Elektronicznych Ochrony Zdrowia	Istniejący	Integracja z modułem do uwierzytelniania oparty domyślnie o funkcjonalność uwierzytelnienia Węzłem Krajowym

8	Komponent podpisu dokumentów (eGate)	CeZ	Biblioteka umożliwiająca podpisanie dokumentów	Istniejący	Wykorzystanie funkcjonalności rozwiązania
---	--------------------------------------	-----	--	------------	---

Lista przepływów



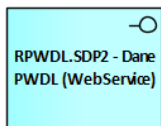
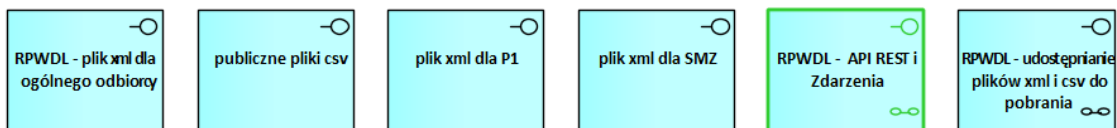
Lp.	System źródłowy	System docelowy	Zakres wymienianych danych	Sposób wymiany danych	Typ modyfikacji	Typ interfejsu
A.1	RPWDL	ZSMOPL	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.2	RPWDL	ROiB	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich

A.3	RPWDL	IOWISZ	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.4	RPWDL	SEZOZ	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.5	RPWDL	SMZ	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.6	RPWDL	SIR	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.7	RPWDL	RPRM	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.8	RPWDL	SSOZ	<ul style="list-style-type: none"> Dane podmiotów medycznych 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
A.9	RPWDL	Poltransplant	<ul style="list-style-type: none"> Dane podmiotów medycznych 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST.

A.10	RPWDL	SOID	<ul style="list-style-type: none"> Dane podmiotów medycznych 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
B.1	REGON	RPWDL	<ul style="list-style-type: none"> Dane podmiotów gospodarczych 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST.
B.2	CEIDG	RPWDL	<ul style="list-style-type: none"> Dane podmiotów gospodarczych 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST.
B.3	Rejestr Systemów Kodowania (ICD11)	RPWDL	<ul style="list-style-type: none"> TERYT Kody Pocztowe Kody resortowe Zakresy świadczeń 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
B.4	CBA	RPWDL	<ul style="list-style-type: none"> Dane adresowe 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST
B.5	e-PLOZ	RPWDL	<ul style="list-style-type: none"> Dane uwierzytelnionego konta 	Bezpośrednia integracja (API)	Krytyczne dla sukcesu projektu	Interfejs WebService
B.6a	RPWDL	eGate	<ul style="list-style-type: none"> Dokument do podpisu 	Integracja biblioteczna (użycie komponentu SDK Szafir)	Krytyczne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST
B.6b	eGate	RPWDL	<ul style="list-style-type: none"> Podpisany dokument 		Krytyczne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST
B.7	CWPM (RPM)	RPWDL	<ul style="list-style-type: none"> Dane personelu medycznego 	API/pliki	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe

			<ul style="list-style-type: none"> Specjalizacje personelu 			(Web service) oparte o REST oraz o zasilanie z plików płaskich
C.1a	RPWDL	Centrum Certyfikacji P1	<ul style="list-style-type: none"> Dane PWDL Certyfikaty P1_WSS, P1_TLS 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST. (ewentualnie pozostaniemy przy wniosku)
C.1b	Centrum Certyfikacji P1	RPWDL	<ul style="list-style-type: none"> Węzeł OID Linki do certyfikatów dostępowych 	Bezpośrednia integracja (API)	istotne dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich
C.2	RPWDL	SFTP P1	<ul style="list-style-type: none"> Dane PWDL na potrzeby instancji RPWDL w P1 	<ol style="list-style-type: none"> API wypracowane w ramach analizy Zasilanie plikowe 	krytyczny dla sukcesu projektu	Interfejsy będą zrealizowane jako usługi sieciowe (Web service) oparte o REST oraz o zasilanie z plików płaskich. Z SFTP P1 nastąpi zasilanie komponentów po stronie P1: CWUD oraz HD CeZ

Udostępniane interfejsy



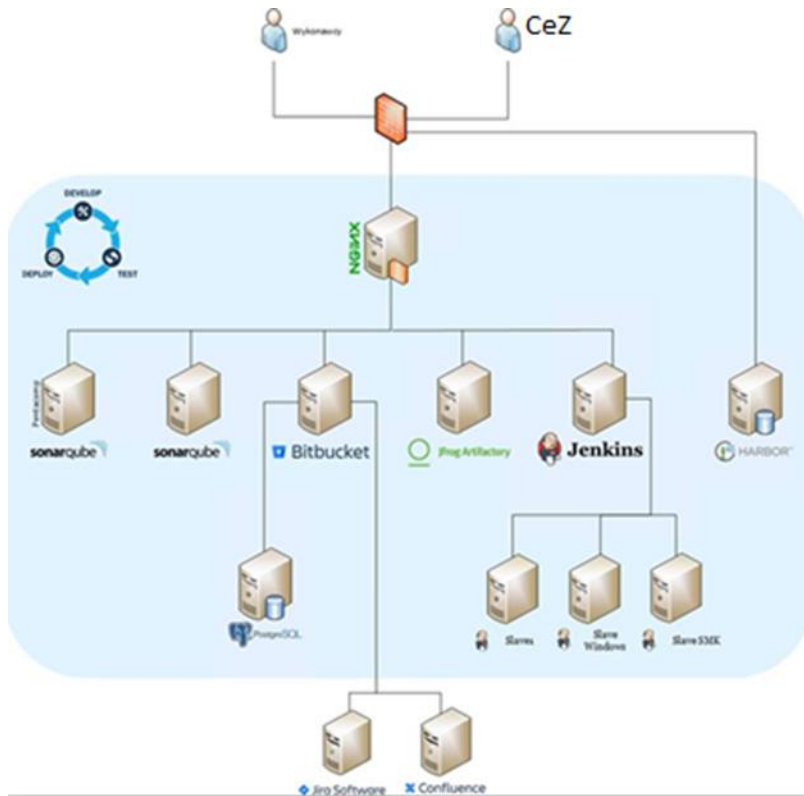
Status zmian

- Planowany
- Modyfikowany
- Wycofywany
- Aktualny

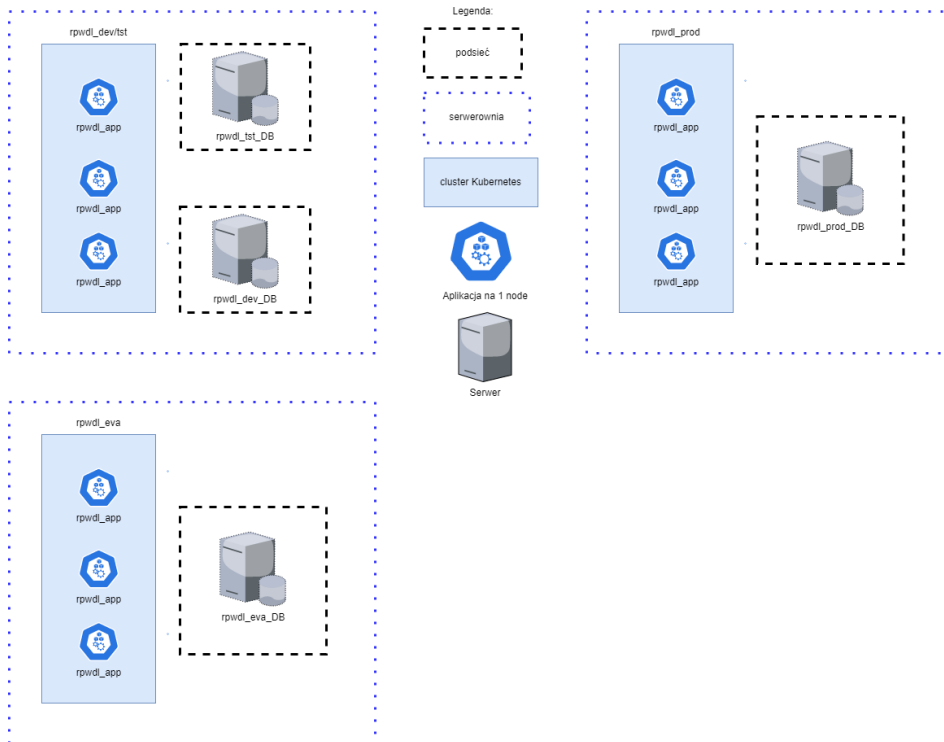
- Planowany
- Modyfikowany
- Wycofywany
- Bez zmian
- Migracja

Technologia

Narzędzia



Orkiestracja Kubernetes



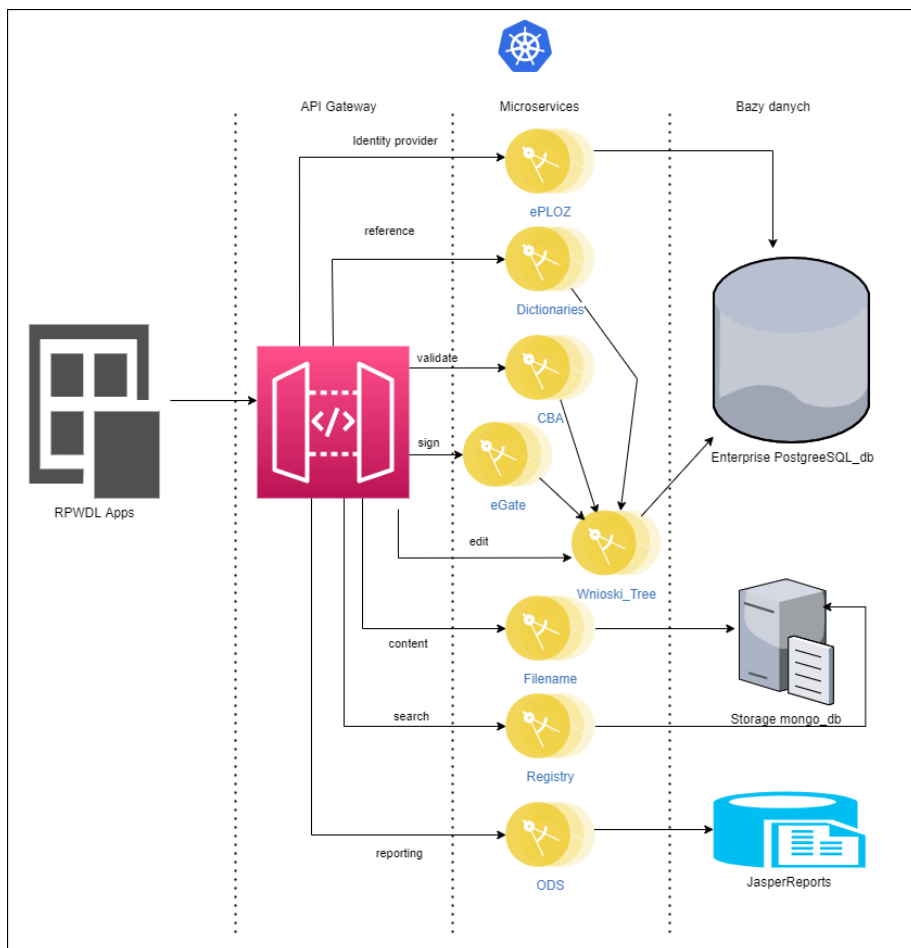
Zestawienie maszyn:

Nazwa	CPU	RAM	HDD	Address Pool	Port	Description
DEV rpwdl2-test-612	3	12	90 GB	10.2.11.211	5444	db dev
TEST rpwdl2-test-609	3	12	90 GB	10.2.11.210	5444	db test
EVA rpwdl2-eva-603	6	12	90 GB	10.2.11.208	5444	db eva
PROD rpwdl2-prod-606	6	12	90 GB	10.2.11.209	5444	db prod
TEST rpwdl2-test-568	4	12	1,5 T	10.2.13.42	5444	db test big data
EVA rpwdl2-eva-571	4	12	1,5 T	10.2.13.43	5444	db eva bug data
EVA2 rpwdl2-eva-579	4	12	90	10.2.13.45	5444	db eva2
TEST rpwdl2-test-610	4	12	90	10.2.13.47	27017	mongo test
DEV rpwdl2-test-606	4	12	90	10.2.13.46	27017	mongo dev
EVA rpwdl2-eva-618	4	12	90	10.2.13.49	27017	mongo eva
EVA2 rpwdl2-eva-622	4	12	90	10.2.13.50	27017	mongo eva2
PROD rpwdl2-prod-627	4	12	90	10.2.13.51	27017	mongo prod

Klastry kubernetes

Nazwa klastra	Zasoby
DEV klaster	3 workery (8 CPU, 16GB RAM)
TEST klaster	7 workery (8 CPU, 8GB RAM)
EVA klaster	3 workery (8 CPU, 8GB RAM)
EVA2 klaster	3 workery (8 CPU, 8GB RAM)
PROD klaster	3 workery (8 CPU, 8GB RAM)

Przeptywy sieciowe



Główne składowe stosu:

- Front: Angular
- Backend: Java
- Baza danych: EnterpriseDB, MongoDB
- Orkiestracja: Kubernetes

Uwierzytelnienie wywołań: SSL, token JWT

7. System Obsługi Egzaminów Medycznych (CEM)

Opis systemu

System wspiera działania statusowe Centrum Egzaminów Medycznych w Łodzi a realizuje je z delegacji ustawowej Ministerstwa Zdrowia. Instytucja ta przeprowadza egzaminy i recertyfikacje personelu w ramach większości grup zawodowych ezdrowia w Polsce. System współpracuje z SMK (w tym wyróżniony jest współdzielony SBB) oraz z innymi systemami CeZ (ePLOZ, eGate, CBA, RPM, płatności online). System dostarcza również komponent pieczęci organizacji.

Zarys architektury

System (w części centralnej – CEM) będzie zbudowana w architekturze mikroserwisowej:

- Wariant 1 - **wybrany** - z wykorzystaniem orkiestratora konteneryzacji Kubernetes w wariacie 1 (dodanie komponentów do infrastruktury Kubernetes SMK 2.0).

- Wariant 2 – w standardowym stosie klasycznych aplikacji dystrybuowanych na serwery aplikacyjne (współdzielenie infrastruktury RPM 2.0 – projekt RIZM)

Wybór wariantu jest otwarty i zależy od strategii i kosztów działania rozwiązań w danym typie kolokacji infrastrukturalnej.

Część serwerowa /backend/ będzie stworzona z wykorzystaniem języka Java.

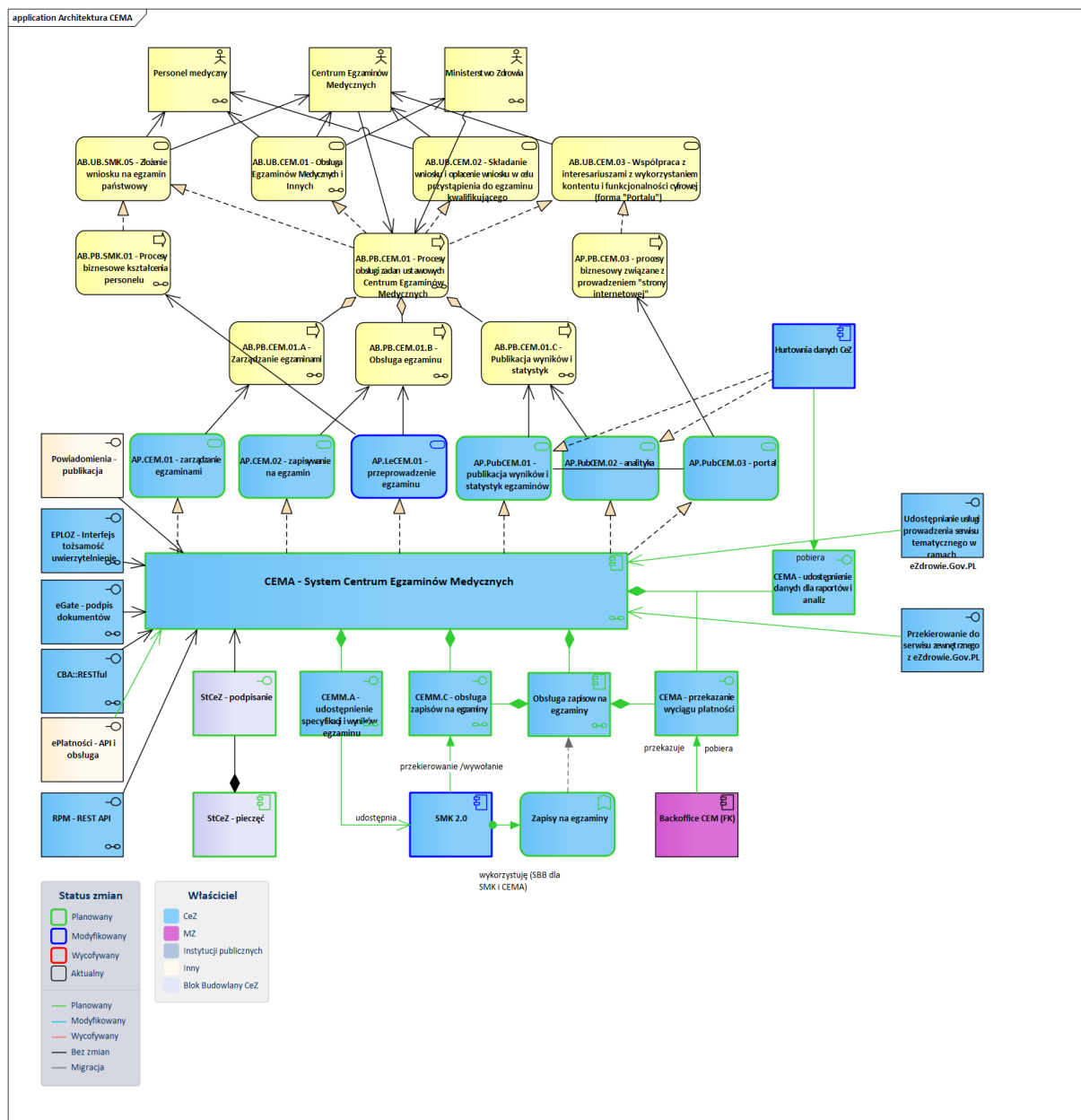
Część interfejsu użytkownika /frontend/ stworzona jest we frameworku Angular, działająca w przeglądarce jako SPA.

System CEM uruchamiany będzie etapowo gdzie istnieje możliwość stopniowego włączania wsparcia funkcjonalnego dla użytkownika (zgodnie z regułą: Must->Should->Could). Spójną sesję zapewnia usługa SSO dostarczana przez system ePŁOZ. Podpis imienny dokumentów zapewnia eGate. Podpis (stempel) organizacji zapewnia eCert (planowany do wytworzenia na rzecz rozwiązań CeZ). Płatność online zapewnia usługa KIR PayByNet. Integracja z systemami będzie interoperacyjna (dotyczy integracji z: SMK, RPM, LeCEM (aplikacja do obsługi egzaminów) lub inna (dotyczy zostaną usługi integracyjne pozwalające na wymianę danych pomiędzy systemami:

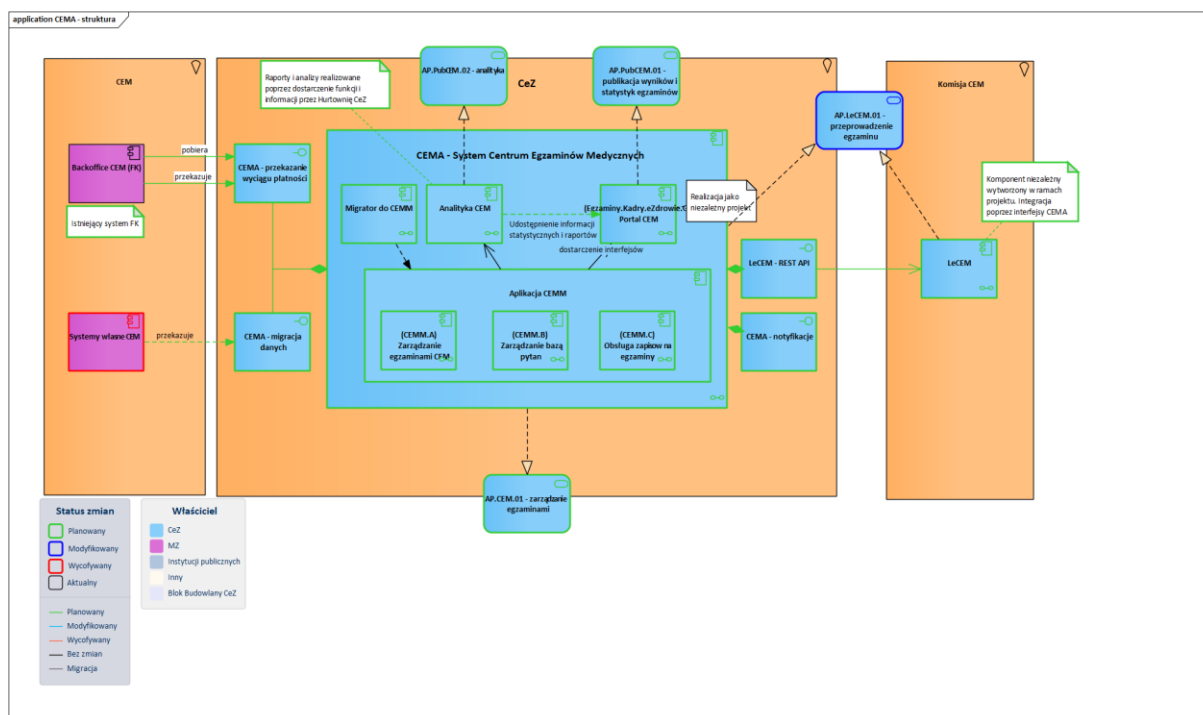
W wariantcie 1 kontenery systemu (z wyłączeniem relacyjnej bazy danych) uruchamiane są pod kontrolą orkiestratora Kubernetes.

Relacyjne bazy danych uruchamiane są na dedykowanych maszynach wirtualnych. W przypadku środowiska produkcyjnego są to dwie niezależne maszyny wirtualne.

Diagram ogólny architektury CEM



Architektura aplikacyjna

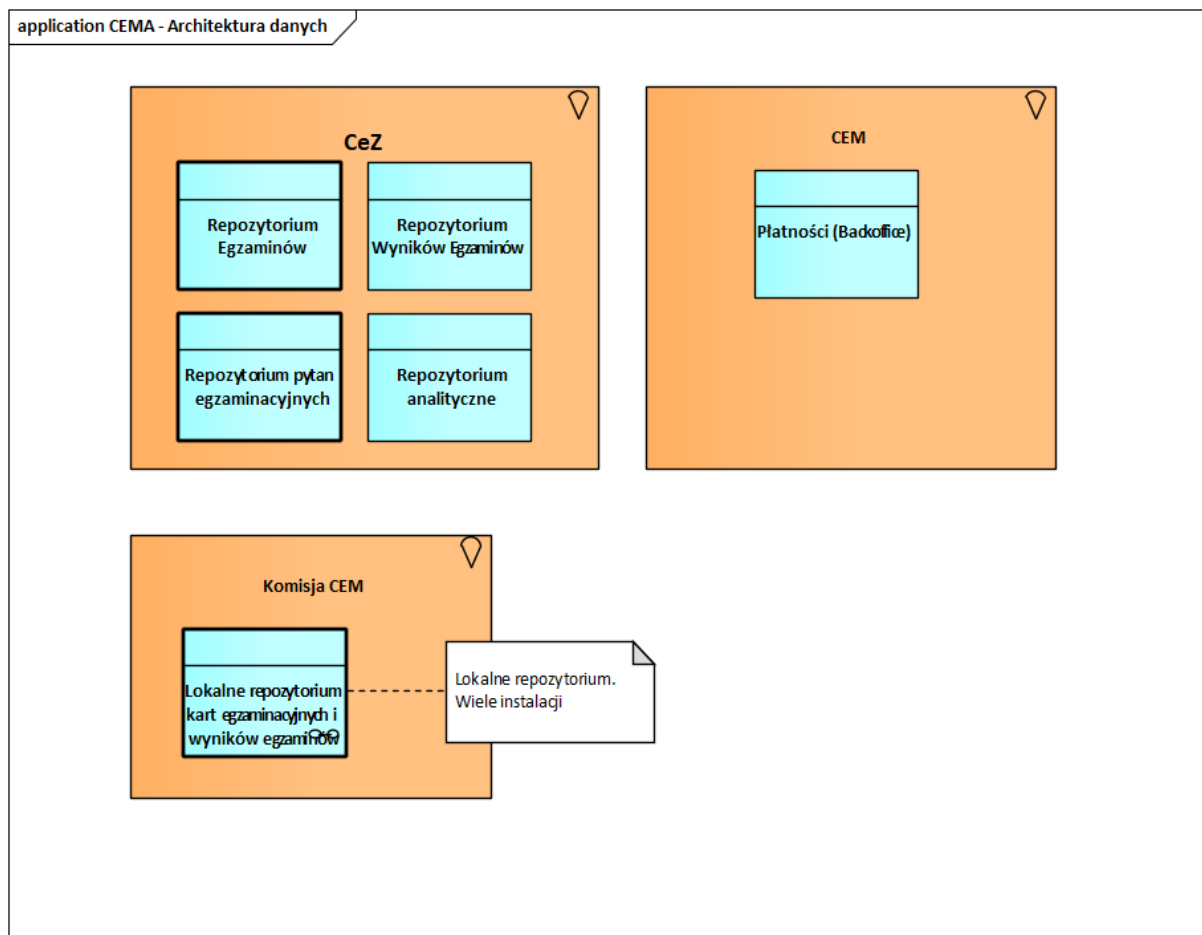


Zestawienie komponentów aplikacyjnych

Nazwa komponentu	Opis i funkcje komponentu	Opis obiektu danych przetwarzanego przez komponent
CEM	Zarządzanie i obsługa definicjami oraz wynikami egzaminów. Obsługa wnioskowania o przystąpienie	Egzamin Przystępujący Wynik
CEM.A - Zarządzanie egzaminami CEM	Funkcjonalność zarządzania egzaminami, komisjami, zatwierdzanie wyników egzaminów, publikacja informacji, wydawanie certyfikatów	Egzamin Wynik Przystępujący pytanie Certyfikat Członek komisji Rozliczenie
CEM.B - Zarządzanie bazą pytań	Tworzenie bazy pytań, udostępnianie wyciągu pytań, obsługa ekspertów dziedzinowych	Pytanie log pracy
CEM.C - Obsługa zapisów na egzaminy	Współdzielona z SMK funkcjonalność (SBB) do obsługi zapisu na konkretny egzamin z obsługą płatności online i integracją z systemem Backoffice CEM	zapis na egzamin płatność przystępujący
Portal CEM	Opis funkcjonalności w ramach KBT - Portal CEM a w szczególności: Publikacja wyników egzaminu Publikacja statystyk Publikacja wyciągu z listy pytań Publikacja listy egzaminów	Egzamin Wynik Pytania Statystyki Artykuły

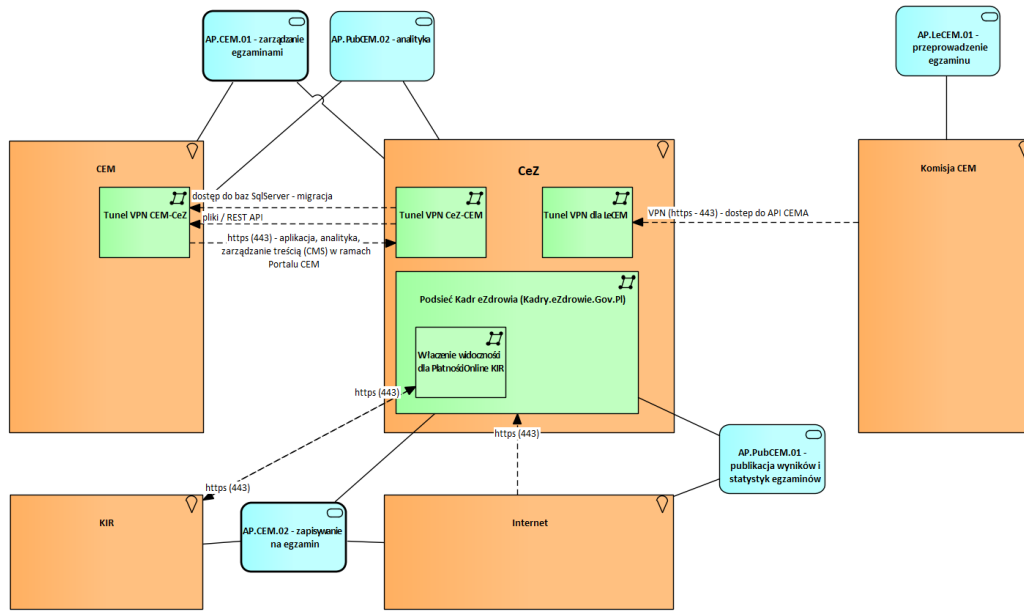
Analityka CEM	Tworzenie raportów i analiz	Komplet danych przetwarzanych w obszarze
LeCEM	Obsługa lokalna egzaminu	Egzamin Wynik
Migrator CEM	Jednorazowa funkcjonalność migracji danych	Egzamin Wynik

Architektura danych

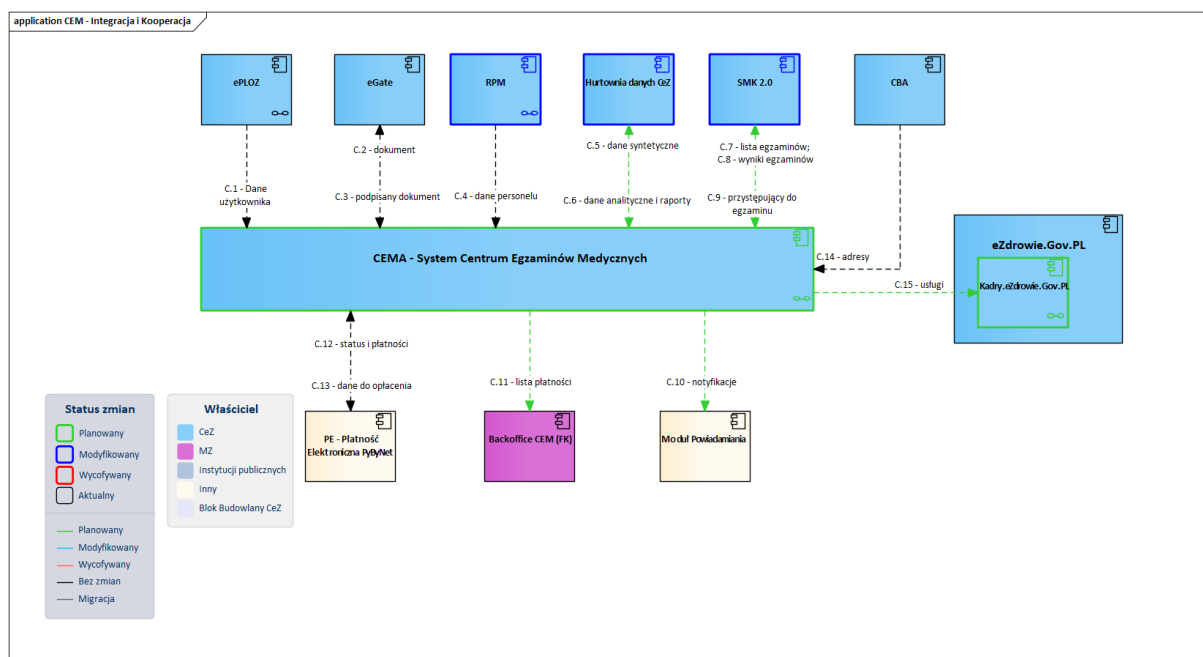


Infrastruktura sieciowa

Diagram: Granice lokalizacyjne i przepływy na poziomie międzylokalizacjami



Integracje z systemami zewnętrznymi



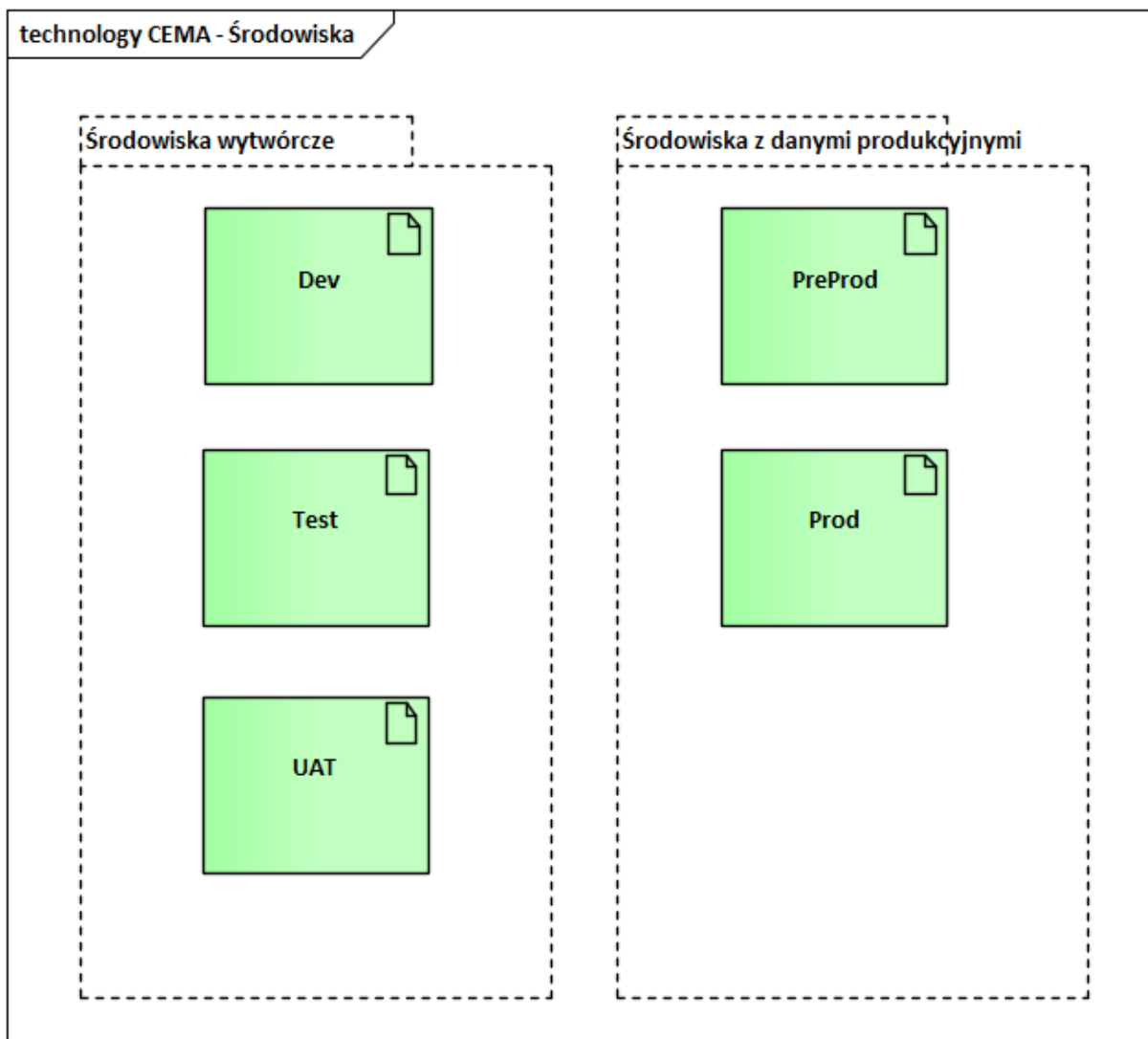
ID	System źródłowy -> docelowy	Obiekt danych	Typ interfejsu	Czy wymaga utworzenia lub zmiany?
C.1	ePŁOZ → CEM	Uwierzytelnienie	sesja	NIE
C.2	CEM → eGATE;	Dokument	front, usługa	NIE
C.3	eGATE → CEM			
C.4	RPM → CEM	Dane personelu	usługa (REST API)	NIE
C.5	CEM → Hurtownia CeZ	Dane syntetyczne gromadzone w CEM	bezpośredni dostęp do danych, usługa (REST API)	TAK
C.6	Hurtownia CeZ → CEM	Analizy i raporty	zgodnie z standardem narzędziowym	NIE
C.7	CEM → SMK	Definicja egzaminu	usługa (REST API) / kolejka (Rabbit MQ)	TAK
C.8	CEM → SMK	Wynik egzaminu	usługa (REST API) / kolejka (Rabbit MQ)	TAK
C.9	SMK → CEM	Przystępujący do egzaminu	usługa (REST API), przekierowanie URL	TAK
C.10	CEM → Notyfikacje	Notyfikacja	usługa (REST API)	TAK (SBB w trakcie wytwarzania)
C.11	CEM → Backoffice CEM	Płatność	usługa (REST API) / kolejka zdarzeniowa / plik płaski	TAK
C.12	ePłatności → CEM	Potwierdzenie realizacji	usługa	NIE
C.13	CEM → ePłatności	Realizacja płatności	usługa, przekierowanie URL	NIE
C.14	CBA → CEM	Adresy	usługa	NIE
C.15	CEM → Portal CEM	Kompleksowe udostępnienie danych i funkcjonalności	usługi (REST), funkcjonalności, współdzielenie konta zalogowanego użytkownika	TAK

L.1	CEM → LeCEM	definicja egzaminu, pytania, lista odpersonalizowana zdających	usługa (REST)	TAK
L.2	LeCEM → CEM	wyniki	usługa (REST)	TAK

Wykaz i skalowanie środowisk

Nie zostały na etapie koncepcji określone precyzyjne wymagania w zakresie ITS i środowisk.

Środowiska (maksymalny zakres – do oszacowania realne potrzeby):



UWAGA: środowiska mają współdzielone zasoby z SMK (są dodane do infrastruktury SMK).

Środowisko	vCPU	RAM	Dysk
DEV i TEST	2	64	120
PRE-PROD	8	120	240
PROD	16	128	1000

Bazy danych są wydzielone z konteneryzacji Kubernetes.

Główne składowe stosu:

- Java,
- PostgreSQL (baza),
- Angular.