Poznańskie Centrum Superkomputerowo-Sieciowe
ICHB Polska Akademia Nauk

Dokumentacja architektury Platformy Technologicznej **epodreczniki.pl**

Platformy Technologicznej
epodreczniki.pl
w wersji 3.0

Poznań, 30 listopada 2015 r.

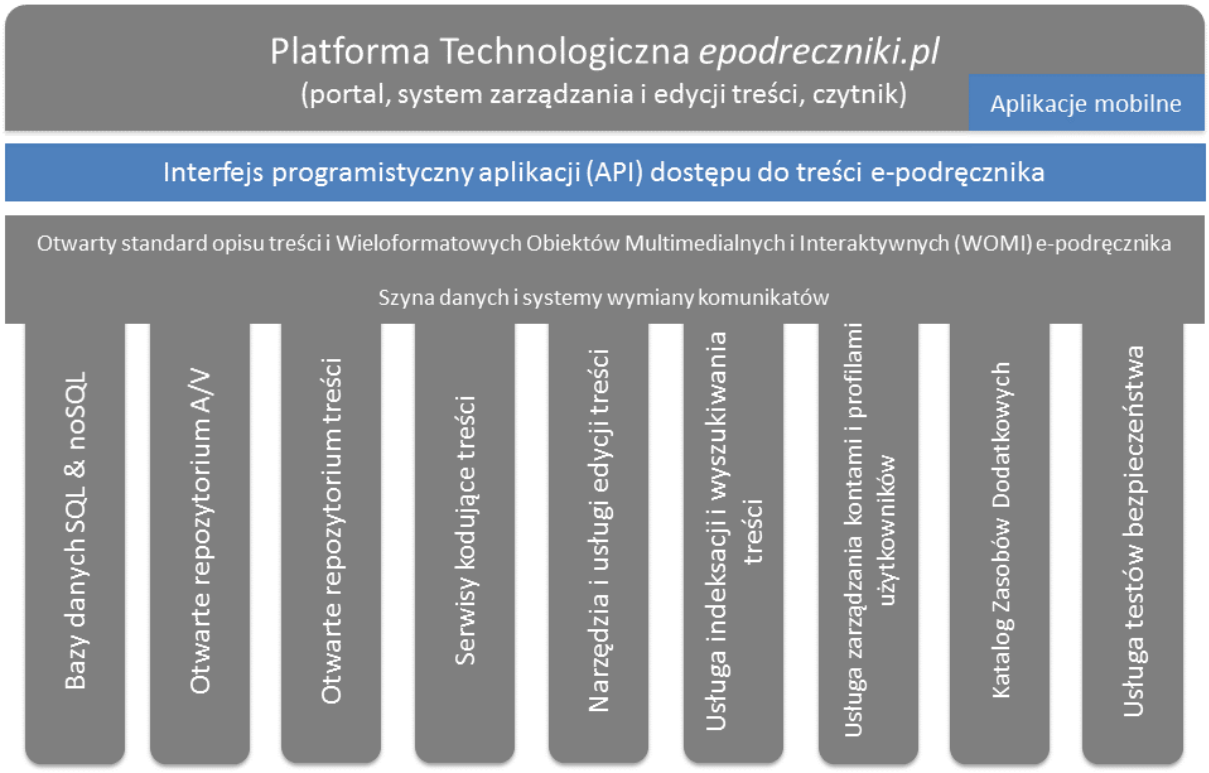
Spis treści

Wstęp 3
Podstawowe informacje 4
Komponenty 6
Otwarte Repozytorium A/V 6
Otwarte Repozytorium Treści 12
Serwisy kodujące treści (SK) 15
System zarządzania Katalogiem Zasobów Dodatkowych 16
Narzędzia i usługi edycji treści 18
Dodatek MS Word 18
Konwerter dokumentów OOXML 19
Serwis wymiany komunikatów pomiędzy komponentami 22
Usługa wyszukiwania treści 23
Usługa kont użytkowników 25
Usługa rejestracji użytkowników 25
Usługa uwierzytelniania użytkowników 26
Kontekst użytkownika (postęp zadań, notatki) 27
Portal (CMS + Biblioteka + Czytnik + KZD) 28
Aplikacje mobilne 34
Aplikacja off-line dla treści do nauczania wczesnoszkolnego 36
Usługa utrzymania podwyższonego poziomu bezpieczeństwa 37

Wstęp

Platforma technologiczna www.epodreczniki.pl jest złożonym systemem informatycznym składającym się z zestawu niezbędnych usług i narzędzi udostępniających kluczowe funkcje wspierające proces gromadzenia, zarządzania, edycji i masowej dystrybucji otwartych treści edukacyjnych zgodnych z aktualną podstawą programową. Platforma definiuje zestaw otwartych standardów opisu cyfrowych treści e-podręcznika, a jednocześnie umożliwia współbieżne zapisy, odczyty, modyfikacje, edycje i wymianę treści poprzez interfejsy programistyczne aplikacji (API). Warstwa prezentacji otwartych treści edukacyjnych e-podręcznika, w tym tekstu, obrazu, audio, wideo, zadań interaktywnych itp. odseparowana jest od źródeł treści gromadzonych w bazach danych i repozytoriach. Platforma zapewnia stały dostęp użytkownikom w trybie pracy stałego połączenia internetowego, a za pomocą aplikacji mobilnych umożliwia im pobranie i dostęp do wybranych cyfrowych treści edukacyjnych bez konieczności utrzymania połączenia internetowego. Ponadto, platforma udostępnia intuicyjny i łatwy w obsłudze portalowy interfejs użytkownika, zapewnia automatyczne dostosowanie interfejsu użytkownika oraz przeskalowanie treści cyfrowych odpowiednio do typu, rozdzielczości ekranu, formatów multimedialnych oraz standardów obsługiwanych przez różne urządzenia końcowe uczniów i nauczycieli. Platforma technologiczna spełnia również kluczowe wymagania odnośnie poziomu bezpieczeństwa oprogramowania oraz zabezpieczeń infrastruktury obliczeniowo-sieciowej. Wykorzystując najnowsze technologie chmurowe, zapewnia w dynamiczny sposób skalowalność liczby użytkowników końcowych oraz wspiera ciągłą integrację z zewnętrznymi systemami IT wdrażanymi w szkołach, np. e-dziennikami i systemami e-Learning.

Ogólna architektura oprogramowania i podstawowe komponenty systemu:



Testowa instalacja i konfiguracja platformy obejmuje pełen nadzór oraz monitoring wyżej wymienionych komponentów w celu zagwarantowania niezwłocznej reakcji Administratorów platformy na ewentualne awarie i/lub sytuacje niepożądane takie jak cyberataki, przeciążenia itp. mogące skutkować ograniczeniem dostępności usługi.

Testowa instalacja oprogramowania platformy umożliwia także dynamiczne gromadzenie statystyk dotyczących wykorzystania platformy, obciążenia infrastruktury sieciowo-obliczeniowej i zajętości przestrzeni dyskowej dedykowanej dla systemu, w tym statystyki takie jak:

- liczba odsłon treści na portalu, liczby unikalnych użytkowników zalogowanych i nie zalogowanych, liczba jednoczesnych strumieni A/V wydawanych przez platformę, ruch sieciowy przychodzący i wychodzący;
- stopnia obciążenia procesorów poszczególnych maszyn fizycznych obsługujących platformę, statystyk sieciowych dla interfejsów maszyn fizycznych i wirtualnych;
- liczba i czas wystąpienia przeciążeń (przekroczeń wydajności) platformy.

Zapewnione jest bezpieczeństwo danych udostępnianych na platformie, w szczególności danych wytwarzanych przez użytkowników platformy. Dodatkowe procedury zapewniają automatyczne wykonywanie kopii bezpieczeństwa ww. danych w trybie dobowym w taki sposób, aby wykonywanie kopii nie powodowało utrudnienia w dostępie do usługi. Wykonywanie kopii bezpieczeństwa wykonywane jest pod nadzorem Administratora platformy dla weryfikacji prawidłowości wykonania kopii. Procedury bezpieczeństwa w szczególności wymagają, aby nośniki, urządzenia i/lub systemy, przy pomocy których wykonywane są kopie bezpieczeństwa umieszczone były w innej fizycznej/geograficznej lokalizacji niż lokalizacja serwerów platformy. Testowa instalacja oprogramowania platformy obejmuje również szczegółowy plan ciągłości działania platformy z uwzględnieniem planu wznawiania. Dla weryfikacji i aktualizacji poprawności planu wznawiania przeprowadzone zostały, na bazie tworzonych kopii bezpieczeństwa, testy symulowanego odtwarzania awaryjnego na zasobach sieciowo-obliczeniowych innych niż zasoby wchodzące w skład platformy.

Podstawowe informacje

Niniejsza sekcja przedstawia wyjaśnienie podstawowych pojęć występujących w opisach komponentów platformy, w dalszej części dokumentacji.

W projekcie utrzymana jest separacja warstwy prezentacji od warstwy treści. Z tego względu każdy e-podręcznik przechowywany jest w formie źródłowej, w której wszystkie elementy opisane są semantycznie, a na podstawie formy źródłowej odpowiednie komponenty platformy generują różne formaty emisyjne, czyli formy prezentacyjne e-podręcznika i jego składowych.

Budowa i format źródłowy e-podręcznika

W skład pojedynczego e-podręcznika, obok jego własnych metadanych, wchodzą **moduły** treści oraz **WOMI** (Wieloformatowe Obiekty Multimedialne i Interaktywne). E-podręcznik jest dalej

nazywany także **kolekcją** (kolekcją modułów) lub po prostu podręcznikiem. Każdy z tych trzech typów obiektów na platformie - kolekcji, modułów i WOMI - ma swój specyficzny format źródłowy.

Sam obiekt **kolekcji** (e-podręcznika) jest przechowywany w formie pliku XML, w formacie zwanym *epCollXML*. epCollXML zawiera m.in. metadane kolekcji (np. tytuł, autorzy, przedmiot, klasa), strukturę i odwołania do obiektów modułów tworzących rozdziały i podrozdziały podręcznika, odwołania do WOMI stanowiących pewne metadane podręcznika (np. WOMI będące okładką podręcznika).

Moduł treści reprezentuje fragment e-podręcznika z właściwą treścią - najczęściej zamknięty rozdział lub podrozdział. Moduł przechowywany jest w formie pliku XML, w formacie zwanym *epXML*, zawierającym przede wszystkim właściwą treść, której poszczególne fragmenty są oznaczone sementycznie (nie zawierają natomiast informacji o sposobie prezentacji danego fragmentu). W treści modułu osadza się odwołania do WOMI.

WOMI (Wieloformatowe Obiekty Multimedialne i Interaktywne) są dopełnieniem treści e-podręcznika i stanowią materiały graficzne, materiały audio-wideo czy też bardziej rozbudowane aplikacje interaktywne, które pojawiają się w treści e-podręczników. Częścią wspólną reprezentacji wszystkich rodzajów WOMI są opisy metadanych, takich jak tytuł, autor czy licencja, na której udostępniany jest dany zasobów. Poza metadanymi struktura źródłowa WOMI zależy w dużej mierze od rodzaju WOMI i może być jednym plikiem - w przypadku WOMI typu graficznego - a także rozbudowaną strukturą plików i katalogów - w przypadku WOMI będącego zaawansowaną grą interaktywną.

Formaty emisyjne

Formy źródłowe kolekcji, modułów i WOMI są przez odpowiednie komponenty platformy konwertowane do postaci emisyjnych, czyli przeznaczonych do warstwy prezentacji.

Całe e-podręczniki prezentowane są w formie, która zależy od sposobu, w jaki użytkownik uzyskuje dostęp do tych materiałów. E-podręcznik może być np. przeglądany w wersji on-line, czyli przeznaczonej do czytnika w Portalu, może być wydrukowany z użyciem formatu PDF, czy przeglądany w wersji off-line w aplikacjach mobilnych. Opis semantyczny źródłowej postaci e-podręcznika pozwala komponentom automatycznie generować różnie wyglądające i różnie działające formy emisyjne.

Również same WOMI są przetwarzane z postaci źródłowej do różnych postaci emisyjnych, zależnych przede wszystkim od urządzenia, na którym będzie je przeglądał użytkownik końcowy. Np. na odpowiednio dobrym komputerze PC materiały graficzne są prezentowane w dużo wyższej rozdzielczości niż na smartfonie, a ćwiczenie interaktywne dostępne w przeglądarce internetowej zostanie odpowiednio zastąpione uproszczoną wersją w formacie e-podręcznika przeznaczonym do wydruku.

Komponenty

Otwarte Repozytorium A/V

Krótki opis komponentu

Repozytorium AV jest systemem służącym do przechowywania, przetwarzania oraz udostępniania treści audiowizualnych. W repozytorium przechowywane są materiały AV oraz ich metadane, a dokładniej metadane opisowe oraz zbiór formatów, na który składa się format źródłowy umieszczony w repozytorium przez dostawcę treści oraz formaty pochodne powstałe w wyniku rekodowania formatu źródłowego. Poprzez usługi dostępowe repozytorium możliwe jest z jednej strony dodanie, aktualizacja, usunięcie oraz sprawdzenie stanu materiału, z drugiej zaś pobranie wskazanego formatu danego materiału. Jednym z głównych zadań repozytorium jest zapewnienie wysokiej dostępności przechowywanych materiałów.

Realizowane funkcjonalności

Struktura repozytorium

Repozytorium jest systemem rozproszonym, złożonym z wielu węzłów funkcjonujących niezależnie. Dostępność materiałów oraz realizowanych usług zapewniana jest przez replikację. Każdy format materiału przechowywany jest na kilku różnych węzłach. Podobnie poszczególne funkcje repozytorium są realizowane przez kilka różnych węzłów.

Na każdym z węzłów repozytorium działają podsystemy realizujące poszczególne jego funkcje. Do podsystemów należą:

- *manager* - nadzoruje pracę repozytorium, dokonuje wyboru węzłów do przechowywania formatów, monitoruje i koryguje liczbę kopii poszczególnych formatów materiałów, nadzoruje proces przetwarzania formatów oraz dba o spójność repozytorium;
- *snode* – przechowuje i udostępnia formaty materiałów; podsystem dysponuje lokalnym repozytorium dyskowym konfigurowanym i zarządzane autonomicznie w ramach danego węzła;
- *recoder* - realizuje operacje przetwarzania, inaczej rekodowania, formatów materiałów; recodery mogą różnić się zbiorem operacji przetwarzania, które są w stanie wykonywać, co jest wynikiem m.in. zróżnicowania wymagań sprzętowych związanych z charakterem różnych operacji; poszczególne rodzaje operacji rekodowania są z reguły wykonywane przez recodery działające na kilku różnych węzłach w celu zwiększenia skalowalności systemu oraz dostępności usługi rekodowania.

Węzły repozytorium mogą pełnić różne role wyznaczające zbiór podsystemów oraz tryb ich działania w danym węźle. Role te obejmują: węzeł przechowujący (storage node – SNode), węzeł rekodujący (Recoder) oraz węzeł pełniący obie te role (SNodeRecoder). Podsystem manager może działać na każdym z węzłów i nie wpływa na pełnioną przez niego rolę.

Podsystemy realizują zadania wewnętrzne wynikające z ich funkcjonalności oraz zadania zlecane im z zewnątrz. Dla podsystemów snode i recoder zadania zewnętrzne zlecane są przez managera, natomiast dla podsystemu managera zadania zewnętrzne zlecane są przez użytkownika za

pośrednictwem interfejsu RepAPI. Zlecanie zadań odbywa się za pośrednictwem bazy danych, z której węzły pobierają zadania do wykonania w trybie „pull" i do której zapisują wyniki realizacji zadania.

Materiały i formaty

Materiał składa się z metadanych oraz zbioru formatów, czyli postaci pod jakim dany materiał występuje w repozytorium. Formaty różnią się między sobą np. sposobem kodowania czy parametrami kodowania, ale również ścieżkami dźwiękowymi i napisami. Formaty materiału podzielone są na grupy następującego typu:

- grupa źródłowa - zawiera format źródłowy wideo lub audio oraz opcjonalnie pliki z napisami typu *subtitles* i *captions;*
- grupa pochodna - tworzona dla każdej kombinacji ścieżki dźwiękowej formatu źródłowego oraz pliku z napisami lub jego braku (dla grupy wskazany jest format źródłowy audio lub wideo, ścieżka audio oraz plik z napisami); grupa tego typu zawiera zwykle jeden format dla każdego zdefiniowanego w systemie profilu rekodowania.

Grupa źródłowa tworzona jest w momencie dodawania materiału do repozytorium, natomiast grupy pochodne powstają na etapie rekodowania materiału. Przykładowo, dla materiału, którego format źródłowy zawiera dwie ścieżki dźwiękowe i dla którego do repozytorium dodane zostały dwa pliki z napisami, powstanie siedem grup formatów:

- źródłowa zawierająca format źródłowy oraz napisy typu *captions* i *subtitle*,
- pochodna: audio 1,
- pochodna: audio 1, *subtitle*
- pochodna: audio 1, *captions*
- pochodna: audio 2,
- pochodna: audio 2, *subtitle*
- pochodna: audio 2, *captions*.

Formaty w danej grupie pochodnej zwierają tylko jedną wskazaną ścieżkę dźwiękową. Grupy, dla których wskazany został plik z napisami, zawierają formaty wideo z osadzonymi napisami z odpowiedniego pliku.

Materiał dodawany do repozytorium przechodzi dwie fazy przetwarzania: dodawanie i rekodowanie. W związku z tym materiał może znajdować w jednym ze stanów przedstawionych na rysunku. Na czerwono zaznaczone zostały stany wskazujące na wystąpienie błędu. Na etapie dodawania materiał może przejść w stan: *NotFound* w przypadku niemożności pobrania formatu źródłowego lub pliku z napisami z adresu URL podanego w zleceniu dodania materiału (błąd 404), *InvalidFile* w przypadku gdy któryś z pobranych plików ma format niepoprawny lub niezgodny z zadeklarowanym typem MIME,*AddError* w przypadku niemożności umieszczenia minimalnej wymaganej liczby kopii któregoś z formatów źródłowych w repozytorium. Stan *RecError*nadawany jest materiałowi w przypadku wystąpienia błędu rekodowania formatu źródłowego lub problemu z umieszczeniem formatów wynikowych w repozytorium. W przypadku stanów *AddError* i *RecError* podejmowana jest próba

naprawienia błędu, stąd możliwość przejścia materiału z tych stanów do stanu odpowiednio *Adding* lub *Recoding*. Stany *Adding* i *Recoding* są stanami przejściowymi i świadczą o tym, że na materiale wykonywana jest jakaś operacja. Materiał pozostaje w stanie *Added*, jeśli nie ma dla niego zdefiniowanej żadnej operacji rekodowania, zaś w stanie *Recoded* po zakończeniu wszystkich operacji rekodowania.

Rys. Stany materiału w repozytorium

Liczba formatów umieszczanych na etapie rekodowania w każdej z grup formatów pochodnych zależy od liczby zdefiniowanych profilów rekodowania. Zdefiniowanych zostało 6 operacji rekodowania dla materiału typu wideo oraz trzy operacje rekodowania dla materiałów typu audio. Dwie pierwsze tabele prezentują listę i parametry profilów. Dla niektórych profilów rekodowania materiałów wideo zdefiniowane są warunki, jakie musi spełniać format źródłowy wideo, aby dany profil mógł zostać zastosowany do tego materiału. Warunki te dotyczą rozdzielczości formatu źródłowego (trzecia tabela). W konsekwencji jedna grupa formatów pochodnych może zawierać od dwóch do sześciu formatów.

Tabela. Profile kodowania wideo

Nazwa profilu	**Bitrate**	**Framerate**	**Coding**	**Profile**	**Level**	**Frame height**	**Frame width**
mp4_vlow_bl	110000	8	H.264/MPEG-4 AVC	Baseline	1,3	226	400
mp4_low_bl	500000	25	H.264/MPEG-4 AVC	Baseline	3	270	480
mp4_med_ml	1350000	25	H.264/MPEG-4 AVC	Main	3	360	640
mp4_hi_hl	4000000	25	H.264/MPEG-4 AVC	High	4,1	720	1280
webm_med	1350000	25	On2 VP8			360	640

webm_hi	4000000	25	On2 VP8			360	640

Tabela. Profile kodowania audio

Nazwa profilu	**Birate**	**Coding**	**Sample**	**SampleRate**
audio_low_acc	48000	Advanced Audio Coding	16	32000
audio_med_acc	96000	Advanced Audio Coding	16	44100
audio_med_ogg	96000	Vorbis	16	44100

Tabela. Warunki zastosowania profilów rekodowania wideo

Nazwa profilu	**MinHeight**	**MinWidth**
mp4_low_bl	NULL	432
mp4_med_ml	NULL	576
mp4_hi_hl	648	NULL
webm_hi	648	NULL

W celu zapewnienia dostępności każdy format przechowywany jest w kilku kopiach, z których każda umieszczona jest na innym węźle. Liczba kopii formatu powinna zawierać się w przedziale zdefiniowanym przez dwa parametry: *<minReplicaCount, targetReplicaCount>*. Pierwszy z parametrów określa minimalną liczbę kopii i powinien mieć wartość nie mniejszą niż 2. Drugi parametr określa docelową, czy preferowaną, liczbę kopii formatu, przy czym spełniony powinien być warunek $targetReplicaCount \geq minReplicaCount$.

Bazy danych

System repozytorium A/V składa się z oprogramowania serwerowego pełniącego różne role, a wszystkie metadane przechowywanych materiałów i pozostałe informacje wspólne składowane są w bazach danych. Wszystkie wykorzystywane bazy są relacyjne, zbudowane w technologii T-SQL i hostowane przez SQL Server w wersji 2012 lub nowszej. Bazy oraz elementy oprogramowania znajdują się na logicznie wydzielonych węzłach, realizujących określone funkcje na platformie. Z danych korzystają elementy oprogramowania realizującego dystrybucję i replikację materiałów oraz komponenty umożliwiające dodanie lub pozyskanie treści przez zewnętrznych klientów, czyli m.in.

aplikacje użytkowników platformy. Z tego powodu bazy danych są w systemie kluczowym elementem i bez nich nie jest możliwa realizacja żadnych zadań. Wydanie materiału zewnętrznemu klientowi możliwe jest przy działającym co najmniej jednym węźle z lokalną bazą danych – pozostałe komponenty nie muszą być w tym momencie dostępne.

Centralną bazą danych jest RepDB, zawierająca metadane wszystkich materiałów i formatów dostępnych w systemie oraz informacje o aktualnych lokalizacjach, czyli węzłach posiadających treści w swoich lokalnych repozytoriach. Ponadto znajdują się w niej informacje wymieniane pomiędzy oprogramowaniem różnych węzłów. W ten sposób w systemie w zasadzie wyeliminowano bezpośrednią komunikację pomiędzy różnymi komponentami oprogramowania. Niezależnie od położenia węzłów, oprogramowanie na nich działające komunikuje się ze sobą poprzez odpowiednie tabele bazy danych RepDB.

Z powodu znaczenia bazy RepDB dla całości systemu zalecane jest skonfigurowanie dla niej „mirroringu" ze wspólnym adresem klastrowym lub alternatywnej metody zapewnienia wysokiej dostępności. Dostęp do bazy musi być zapewniony dla oprogramowania Zarządcy systemu, usług realizujących dostawę nowych materiałów oraz oprogramowania każdego z węzłów w roli SNode oraz Recoder.

Jako że baza RepDB jest elementem wspólnym dla wielu komponentów systemu, to w niej przechowywane są dane konfiguracyjne.

Poza główną bazą każdy węzeł realizujący rekodowanie (Recoder), dystrybucję i replikację (SNode) formatów posiada lokalną bazę MaterialFormatDB. W niej składowane są metadane techniczne przechowywanych formatów oraz informacje o ich fizycznej lokalizacji w systemie plików. Dostępność tej bazy dla oprogramowania węzłów typu Recoder i SNode jest krytyczna – bez niej węzły nie zgłoszą gotowości do pracy. Istotne jest jednak, że nie ma w systemie odwołań do baz tego typu z innych komponentów oprogramowania poza lokalnym węzłem (brak odwołań sieciowych).

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Repozytorium A/V jest połączone dwukierunkowym interfejsem REST z Repozytorium treści. Celem tej integracji jest umożliwienie i automatyzacja przekazywania materiałów AV dodawanych do Repozytorium Treści do Repozytorium AV. Odebrane materiały Repozytorium A/V automatycznie rekoduje do zadanych profili i udostępnia do pobrania przez systemy dystrybucyjne do użytkowników końcowych. O zmianach statusu rekodowania materiału Repozytorium A/V informuje Repozytorium treści.

Wymagania sprzętowe i wobec systemu operacyjnego

Repozytorium A/V zostało wykonane przy użyciu technologii .NET. Część modułów pełniących role aktywne działa jako usługi systemu Windows (Windows Service). Moduły pasywne zostały osadzone jako aplikacje i moduły serwera IIS (Internet Information Server).

Sprzęt

Węzeł repozytorium może być uruchomiony zarówno na maszynie wirtualnej, jak również na fizycznym serwerze. Ze względu na niezawodność zalecane jest, aby fizyczny serwer zapewniał redundancje w zakresie dysków systemowych, zasilania oraz połączenia sieciowego.

Wymagania w stosunku do CPU zależą od roli i konfiguracji zadań przetwarzania. Węzeł pełniący rolę Zarządcy i SNode wymaga dostępnych dwóch rdzeni procesora o wydajności nie gorszej niż Intel I3

(powyżej 2500 PassMark CPU Mark). W przypadku węzła typu Rekoder wymagania te zwiększają się. Właściwa moc obliczeniowa zależy od ilości dodawanych materiałów w jednostce czasu oraz wymagania odnośnie maksymalnego czasu przetwarzania materiału.

Minimalna wielkość pamięci węzłów typu SNode i Zarządca to 8GB. Minimalna wielkość pamięci węzłów typu Rekoder to 16GB, przy czym może zaistnieć konieczność rozbudowy pamięci w przypadku dużej liczby jednoczesnych zadań kodowania. Zależy to od obciążenia repozytorium, czyli ilości dodawanych materiałów w jednostce czasu.

Ilość dostępnej przestrzeni dyskowej na składowane treści zależy od planowanej pojemności repozytorium oraz stopnia redundancji treści pomiędzy węzłami.

Zalecana minimalna prędkość połączenia sieciowego pomiędzy węzłami repozytorium to 2x1Gbit/s

System operacyjny

Węzły repozytorium zostały stworzone i przetestowane do pracy pod kontrolą systemu Windows 2012R2.

Węzeł repozytorium wymaga instalacji następujących komponentów:

- .NET Framework 4.5.1
- ASP.NET 4.5
- Serwer Web (IIS)
- MS SQL Express 2012 lub nowszy

Dodatkowo w ramach repozytorium użytkowana jest wspólna baza danych. Minimalna instalacja wymaga MS SQL Express 2012, jednak w celu uzyskania wyższych parametrów niezawodności należy użyć pełnej wersji MS SQL.

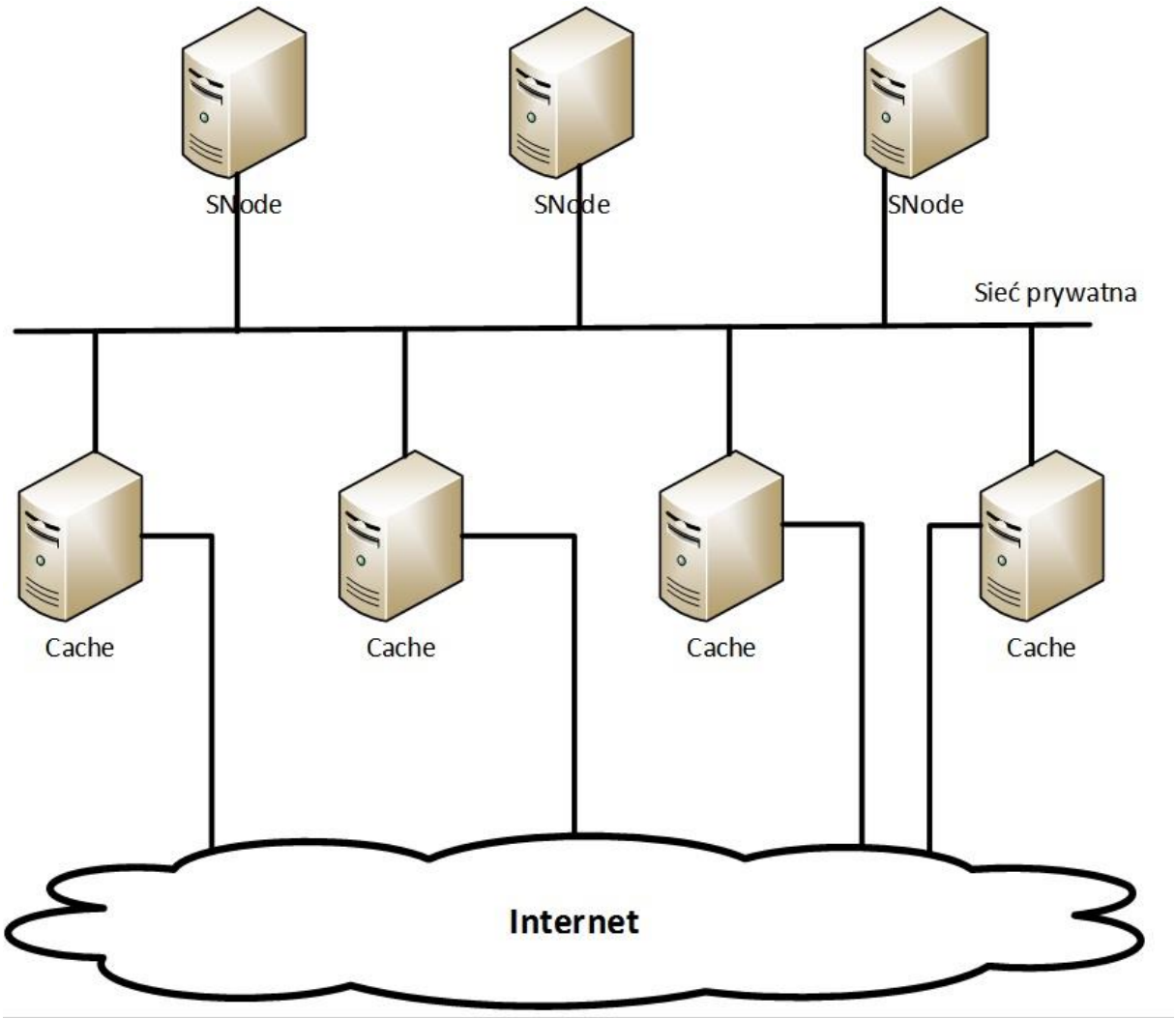
Zwiększenie dostępności całego repozytorium wymaga wdrożenia wielu węzłów poszczególnych ról. Spójny interfejs zewnętrzny jest realizowany poprzez Network Load Balancing – kolejny komponent systemu operacyjnego.

Wymagania sprzętowe i programowe warstwy dostępowej

Udostępnianie plików repozytorium użytkownikom końcowym musi się odbywać poprzez warstwę pośredniczącą zbudowaną z Reverse HTTP Proxy, która ma pełnić rolę cache dla udostępnianych treści. Repozytorium nie ma szczególnych wymagań odnoście warstwy cache. Polecane jest by Reverse Proxy blokowała nieprawidłowe żądania do repozytorium, w tym żądania ze złą sumą kontrolną zawartą w adresie URL.

Przykładowo warstwę Reverse Proxy można zrealizować na węzłach opartych o Microsoft Server 2012R2 z IIS wraz z usługami URL Rewrite v. 2.0 lub nowszej oraz ARR (Application Request Routing).

Schemat przykładowego rozwiązania warstwy dostępowej przedstawia poniższy rysunek:



Instalacja i konfiguracja

Instalacja każdego węzła repozytorium odbywa się poprzez wykonanie skryptu instalacyjnego PowerShell z prawami administratora. Przed uruchomieniem tego skryptu należy przygotować nośnik z odpowiednimi plikami instalacyjnymi.

Do skryptu instalacyjnego podaje się szereg parametrów, które m.in. pozwalają na identyfikację danego węzła repozytorium, określają jego typ i parametry czy wskazują sposób połączenia z centralną bazą danych i pozostałymi węzłami repozytorium.

Otwarte Repozytorium Treści

Krótki opis komponentu

Otwarte Repozytorium Treści to system, w którym przechowywane są treści i metadane wszystkich podręczników, modułów i WOMI. Składa się z kilku części:

- Aplikacja Serwerowa: główny moduł repozytorium.

- Serwer Indeksujący: odpowiada za budowanie i przechowywanie indeksu metadanych oraz treści, umożliwia jego przeszukiwanie.
- Aplikacja Webowa: udostępnia interfejs użytkownika z uproszczonym podglądem WOMI oraz interfejsy dostępu do treści i metadanych dla pozostałych komponentów e-podreczniki.pl w trybie do odczytu.
- Aplikacja Redaktora: aplikacja JNLP, która pozwala użytkownikom wprowadzać treści obiektów i operować na nich.
- Zestaw narzędzi eprt-tools: skrypty pozwalające wprowadzać i modyfikować treści w trybie wsadowym, wykorzystywane przez zaawansowanych użytkowników oraz przez pozostałe komponenty platformy (w tym Dodatek MS Word).

Realizowane funkcjonalności

- Przechowywanie i udostępnianie treści i metadanych obiektów składowych e-podręczników
- Przyjmowanie nowych i zmodyfikowanych obiektów od użytkowników, weryfikacja formalnej poprawności
- Informowanie pozostałych komponentów o zmianach w treści i w metadanych obiektów
- Konwersja źródłowych obrazów WOMI do formatów emisyjnych
- Wstępne przetwarzanie wybranych WOMI interaktywnych do formatów emisyjnych
- Umożliwienie wyszukiwania obiektów na podstawie metadanych

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Repozytorium Treści komunikuje się z kilkoma komponentami platformy:

1. Otwarte Repozytorium A/V: po każdej zmianie zawartości plików w WOMI typu dźwięk lub wideo następuje wysłanie do Repozytorium A/V żądania przetworzenia takiego materiału. Oczekiwane jest żądanie zwrotne, zawierające informację o statusie przetwarzania. Komunikacja odbywa się za pomocą interfejsów REST z użyciem protokołu HTTP.
2. Serwis wymiany komunikatów pomiędzy systemami: po każdej zmianie w metadanych lub treściach obiektów następuje wysłanie komunikatu o takiej zmianie, żeby pozostałe komponenty platformy mogły odświeżyć stan formatów emisyjnych lub pamięci podręcznych. Komunikacja z tym serwisem odbywa się w protokole AMQP.
3. Inne komponenty platformy mogą wprowadzać zmiany do treści w Repozytorium za pomocą narzędzi eprt-tools.

Wymagania sprzętowe i wobec systemu operacyjnego

1. Aplikacja Serwerowa i Aplikacja Webowa mogą być uruchamiane na jednej maszynie. Zalecany sprzęt to co najmniej 8-rdzeniowy procesor, 32 GB RAM, 1 TB przestrzeni dyskowej na dane. Wymagania odnośnie oprogramowania:
    - System operacyjny CentOS 6.5
    - Musi być dostępna wirtualna maszyna Java w wersji 6 lub 7

- Aplikacja Serwerowa
    - baza danych PostgreSQL
    - ImageMagick 6.8.6-9 2014-08-14 Q8
    - Inkscape 0.47
    - szybki wirtualny dysk w pamięci RAM na składowanie cache (8 GB)
- Aplikacja Webowa działa w kontenerze Apache Tomcat 7

2. Serwer Indeksujący również może być uruchamiany na tej samej maszynie, co Aplikacja Serwerowa, ale zalecane jest jego wydzielenie ze względu na duże obciążenie maszyny w czasie ładowania wielu nowych treści. Wymaga tylko dobrego procesora (zalecane 8 rdzeni), wystarczy 4 GB RAM i 100 GB przestrzeni dyskowej.
    - System operacyjny CentOS 6.5
    - Musi być dostępna wirtualna maszyna Java w wersji 6 lub 7.
3. Aplikacja Redaktora oraz Zestaw narzędzi eprt-tools są uruchamiane na komputerach użytkowników. Wymagają zainstalowanej maszyny wirtualnej Java w wersji co najmniej 7.

Instalacja i konfiguracja

1. Aplikacja Serwerowa i Aplikacja Webowa:
    a. Instalacja wymaganego oprogramowania i utworzenie bazy danych
    b. Zaimportowanie zrzutu bazy danych i skopiowanie katalogu z plikami treści
    c. Rozpakowanie dostarczonych paczek ze skompilowanym i wstępnie skonfigurowanym oprogramowaniem
    d. Zmiana plików konfiguracyjnych i tabel w bazie danych, zawierających adresy IP, konfigurację połączenia z bazą danych, ścieżki do katalogów itp.
    e. Konfiguracja Apache Tomcat wraz z zaufanym certyfikatem do połączenia HTTPS
    f. Ewentualna konfiguracja automatycznego uruchamiania z systemem
2. Serwer Indeksujący:
    a. Instalacja wymaganego oprogramowania
    b. Skopiowanie katalogu z indeksem
    c. Zmiana plików konfiguracyjnych zawierających adresy IP, ścieżkę do katalogu indeksu itp.
    d. Ewentualna konfiguracja automatycznego uruchamiania z systemem
3. Aplikacja Redaktora jest instalowana i konfigurowana w ramach Aplikacji Webowej. Użytkownicy pobierają ją przez www.

4. Zestaw narzędzi eprt-tools to zestaw skryptów (.bat/.sh) oraz plików wykonywalnych Java, zawiera plik konfiguracyjny, w którym przed dalszą dystrybucją należy ustawić parametry połączenia z głównym serwerem.

Kody źródłowe wszystkich części Repozytorium są przygotowane do kompilacji przy pomocy narzędzia Maven 1.1. Do tego celu wymagany jest JDK w wersji 1.6 lub 1.7, zgodny z wersją Java środowiska uruchomieniowego.

Przygotowanie Aplikacji Redaktora wymaga podpisania skompilowanych plików JAR zaufanym certyfikatem.

Serwisy kodujące treści (SK)

Krótki opis komponentu

Serwis kodujący treści lub Serwer Kodujący (SK) jest odpowiedzialny za reagowanie na komunikaty z repozytoriów treści o utworzonych i zmodyfikowanych kolekcjach poprzez wygenerowanie ich formatów emisyjnych, a także na reagowanie na zmodyfikowane WOMI poprzez wygenerowanie zaktualizowanych formatów emisyjnych kolekcji, które z tych WOMI korzystają.

Realizowane funkcjonalności

Realizowaną przez SK funkcjonalnością jest generowanie formatów emisyjnych kolekcji - HTML, PDF, format dla aplikacji natywnych - w reakcji na zdarzenia modyfikacji kolekcji lub jej obiektów zależnych (WOMI, treści modułów). Wygenerowane formaty emisyjne są udostępniane za pomocą HTTP dla innych komponentów oraz dla administratorów.

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

SK reaguje na zdarzenia utworzenia, modyfikacji i usunięcia obiektów (tj. kolekcji, modułów i WOMI), które otrzymuje z serwisu wymiany komunikatów.

Dane niezbędne do wygenerowania formatów emisyjnych podręczników, tj. źródłowe pliki XML kolekcji i modułów oraz formaty emisyjne i metadane WOMI Serwer Kodujący pobiera za pomocą interfejsów HTTP z Repozytorium Treści.

Wygenerowane formaty emisyjne SK udostępnia dla innych komponentów za pomocą API HTTP.

Wymagania sprzętowe i wobec systemu operacyjnego

SK wymaga systemu operacyjnego CentOS. Ilość niezbędnej pamięci RAM i liczba rdzeni procesora zależą od obciążenia systemu przychodzącymi zleceniami zmian w treściach i pożądanym czasem oczekiwania na wygenerowanie formatów emisyjnych. Minimalne wymagania to 12 GB pamięci RAM oraz 12 rdzeni procesora. Dysk twardy musi mieć pojemność pozwalającą na przechowywanie formatów emisyjnych oraz ich plików pośrednich dla wszystkich kolekcji (i kolejnych wersji kolekcji), które są w określonym czasie przetwarzane w podglądzie treści lub w systemie publikacji. Na dzień pisania niniejszej dokumentacji jest to min. 2 TB przestrzeni dyskowej.

Instalacja i konfiguracja

Do działania SK niezbędne są:

- środowisko Python 2.7.x z zainstalowaną globalnie w systemie określoną listą zależności

- środowisko Java Development Kit 1.7
- biblioteka PrinceXML 8.1 rev. 5
- biblioteka Euler MathSpeech
- zestaw fontów msttcorefonts oraz fonty latinmodern-math

Do skompilowania części aplikacji napisanej w Javie potrzebny jest Apache Maven 3.x. Przed kompilacją konfiguruje się ustawienia w zakresie komunikacji z serwisem wymiany komunikatów, schemat generowania adresów URL dla pobierania treści z Repozytorium Treści oraz ścieżki do katalogów roboczych.

System zarządzania Katalogiem Zasobów Dodatkowych

Krótki opis komponentu

System zarządzania Katalogu Zasobu Dodatkowych (KZD) umożliwia uprawnionym użytkownikom dodawanie i usuwanie zasobów z Katalogu oraz edycję wybranych atrybutów zasobów.

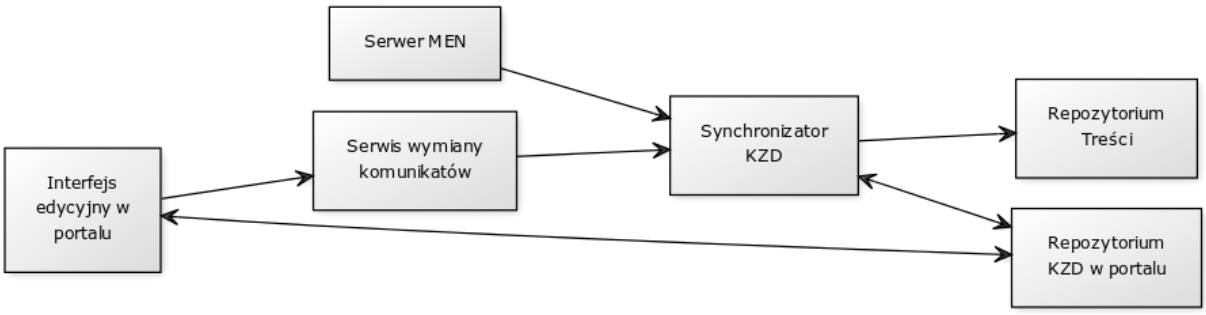
Realizowane funkcjonalności

- dodawanie nowych zasobów do KZD na podstawie plików umieszczonych na serwerze Ministerstwa Edukacji Narodowej (MEN)
- edycja wybranych metadanych zasobów za pomocą interfejstu użytkownika w portalu
- usuwanie zasobów za pomocą interfejsu użytkownika w portalu

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Na system edycji KZD składają się:

- interfejs w portalu umożliwiający edycję i usuwanie zasobów
- repozytorium Edytora KZD w portalu - system plików na serwerze aplikacyjnym portalu odpowiedzialny za przetrzymywanie źródłowych plików XML z opisem zasobów
- synchronizator KZD - aplikacja odpowiedzialna za nanoszenie zmian wprowadzonych w zasobach do Repozytorium Treści (RT)



(strzałki pokazują główny kierunek przepływu informacji)

Scenariusz dodania zasobu

Synchronizator komunikuje się okresowo z serwerem MEN za pomocą FTP, pobiera pliki XML z opisem zasobów, sprawdza za pomocą API HTTP, które z zasobów nie istnieją jeszcze w RT i dla nowych zasobów generuje grupę WOMI reprezentujących zasób, które wysyła do RT za pomocą narzędzia eprt-tools. Synchronizator przesyła źródłowy plik XML zasobu do portalu za pomocą API HTTP. Prawidłowo wczytane pliki XML z opisem zasobów są usuwane z serwera MEN.

Scenariusz edycji metadanych

Przesłanie zmian w formularzu edycyjnym w portalu powoduje zweryfikowanie poprawności wprowadzonych informacji (np. istnienie wskazanych ścieżek na serwerze MEN poprzez nawiązanie połączenia FTP z tym serwerem), zapisanie tych zmian do źródłowego pliku XML w repozytorium KZD w portalu oraz wysłanie komunikatu o modyfikacji do serwisu wymiany komunikatów. Z serwisu wiadomość odbiera Synchronizator, który dla zmodyfikowanych zasobów pobiera pliki XML za pomocą API HTTP z Repozytorium KZD w portalu, łączy się z serwerem MEN za pomocą FTP i na podstawie tych dwóch źródeł generuje grupę WOMI reprezentujących zasób. WOMI, które uległy zmianie, są następnie wysyłane przez Synchronizator do RT za pomocą narzędzia eprt-tools.

Scenariusz usunięcia zasobu

Zatwierdzenie usunięcia zasobu w interfejsie edycyjnym powoduje usunięcie pliku XML zasobu z repozytorium KZD w portalu oraz wysłanie komunikatu o usunięciu do serwisu wymiany komunikatów. Z serwisu wiadomość odbiera Synchronizator, który dla każdej informacji o usuniętym zasobie pobiera informacje o tym zasobie z RT za pomocą API HTTP, a nastepnie zleca usunięcie z RT grupy WOMI reprezentujących dany zasób za pomocą narzędzi eprt-tools.

Wymagania sprzętowe i wobec systemu operacyjnego

Interfejs edycyjny jest częścią portalu, więc nie ma własnych wymagań sprzętowych. Repozytorium KZD w portalu jest zorganizowane w postaci wydzielonej struktury katalogowej na dysku serwera aplikacyjnego portalu.

Synchronizator KZD jest aplikacją działającą w systemie Linux. Aplikacja Synchronizatora wymaga ok. 1 GB pamięci RAM oraz przestrzeń dyskową dla pomieszczenia kopii plików XML opisujących zasoby, które są przetwarzane przez Synchronizatora oraz tymczasowego przechowywania plików zasobów pobieranych z serwera MEN.

Instalacja i konfiguracja

W aplikacji portalu należy skonfigurować ustawienia połączenia FTP do serwera MEN oraz ustawienia niezbędne do komunikacji z serwisem wymiany komunikatów. W konfiguracji należy wskazać lokalizację repozytorium KZD w portalu. Instalacja tej części systemu edycji polega na instalacji portalu.

Sychronizator jest aplikacją napisaną dla środowiska Java 7. W Sychronizatorze należy skonfigurować ustawienia niezbędne do komunikacji z serwerem MEN, portalem (repozytorium KZD w portalu) oraz serwisem wymiany komunikatów. Kompilacja Synchronizatora przebiega z użyciem narzędzia Apache Maven 3.x. Skompilowana aplikacja Synchronizatora musi mieć dostęp do narzędzi eprt-tools, które są skonfigurowane do komunikacji z określoną instancją RT.

Narzędzia i usługi edycji treści

Wytwarzanie treści dla użytkowników opiera się na komponentach - Dodatku MS Word z Konwerterem dokumentów OOXML oraz na Edycji Online.

Administrator platformy ma także możliwość edycji treści poprzez wprowadzanie bezpośrednich zmian do plików źródłowych podręczników, modułów i WOMI, przesłanie poprawionych plików do Repozytorium treści oraz, w razie potrzeby, publikację nowej wersji podręcznika w Bibliotece Portalu.

Dodatek MS Word

Krótki opis komponentu

Dodatek MS Word to uzupełnienie programu Word o zestaw narzędzi, które wspomagają pracę nad tworzeniem e-podręcznika. Każde z narzędzi to szablon, który zawiera zestaw pól składający się na pojedynczy element e-podręcznika, np. szablon dla tworzenia ćwiczenia. Ponadto dodatek umożliwia wyszukanie WOMI z Repozytorium Treści i osadzenie go w treści.

Dodatek pozwala również na konwersję dokumentów MS Word do postaci źródłowej XML, przesłanie modułu lub całego e-podręcznika do Otwartego Repozytorium Treści oraz podgląd formatu emisyjnego podręcznika w przeglądarce internetowej.

Realizowane funkcjonalności

- edycja podręcznika
    - wyszukiwanie WOMI
    - dodawanie/usuwanie WOMI
    - dodawanie/usuwanie narzędzi
    - edycja treści
- konwersja całego podręcznika do postaci źródłowych XML i opcjonalnie przesłanie do Repozytorium Treści
- konwersja pojedynczego pliku
- przesłanie pojedynczego pliku do Repozytorium Treści
- wyświetlenie historii dokumentów przesłanych do Repozytorium Treści
- korzystanie z predefiniowanego zestawu stylów semantycznych e-Podręczniki

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

- Konwerter dokumentów OOXML - konwerter jest uruchamiany jako podprogram i operuje na bieżącym dokumencie lub na zbiorze dokumentów z wybranej struktury katalogów na dysku lokalnym; wynikiem konwersji są źródłowe postaci XML modułów treści oraz raport błędów i ostrzeżeń, które pojawiły się podczas konwersji.

- Repozytorium Treści - za pośrednictwem zestawu narzędzi eprt-tools możliwe jest przesłanie utworzonych modułów do Repozytorium Treści; za pośrednictwem interfejsu HTTP pobierane są z Repozytorium szczegółowe informacje o wybranych WOMI.

Wymagania sprzętowe i wobec systemu operacyjnego

Dodatek na komputerze użytkownika wymaga programu MS Word 2010 (lub nowszego) i systemu operacyjnego MS Windows XP (lub nowszego). Minimalne wymagania sprzętowe to procesor Pentium IV 2 GHz, 1 GB pamięci operacyjnej, 300 MB miejsca na dysku.

Do kompilacji dodatku i opublikowania jego instalatora potrzebny jest program Microsoft Visual Studio 2013 lub nowszy (w tym Microsoft Visual C# 2013), Office Developer Tools for Visual Studio 2013 lub nowszy oraz Microsoft .NET Framework w wersji 4.5. Minimalne wymagania sprzętowe to procesor Intel Core i7-2760QM 2,4 GHz, 8 GB pamięci operacyjnej, 10 GB miejsca na dysku.

Instalacja i konfiguracja

Instalacja na komputerze użytkownika polega na uruchomieniu programu w formacie vsto z lokalizacji sieciowej ze składnikami dodatku.

Aktualizacja dodatku wykonywana jest automatycznie przy uruchomieniu programu Word (jeśli w w/w lokalizacji sieciowej pojawi się nowa wersja dodatku).

Kompilacja i publikowanie instalatora

W plikach projektu należy osadzić zestaw narzędzi eprt-tools, a w konfiguracji wskazać adres URL, pod którym udostępniona jest aplikacja konwertera. W ustawieniach projektu w Visual Studio należy wskazać certyfikat służący do podpisania skompilowanej wersji aplikacji.

Wreszcie należy skompilować aplikację i opublikować w odpowiedniej lokalizacji sieciowej.

Konwerter dokumentów OOXML

Krótki opis komponentu

Konwerter dokumentów OOXML (Office Open XML) pozwala na przetworzenie specjalnie sformatowanych plików *.docx oraz *.docm wytworzonych z wykorzystaniem Dodatku MS Word do formatu źródłowego platformy epodreczniki.pl, którym jest epXML. Konwerter przetwarza cały wskazany katalog wraz z jego podkatalogami i wszystkimi zawartymi w nich plikami, wytwarzając przy tym strukturę katalogową, która odzwierciedla budowę e-podręcznika i która może zostać przesłana do Otwartego Repozytorium Treści. W trakcie przetwarzania ma miejsce kontrola spójności całego e-podręcznika.

Realizowane funkcjonalności

- konwersja specjalnie sformatowanych dokumentów OOXML do formatu epXML
- generowanie podglądu technicznego w postaci plików PDF zawierających informacje na temat wytworzonych elementów semantycznych oraz problemów napotkanych w trakcie konwersji
- wygenerowanie struktury katalogowej odzwierciedlającej budowę e-podręcznika

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Konwerter dokumentów OOXML jest plikiem wykonywalnym, jest uruchamiany przez Dodatek MS Word i komunikuje się z nim przez standardowe wejście i wyjście.

Konwerter komunikuje się z opisanym poniżej systemem aktualizacji z wykorzystaniem protokołu HTTP.

Wymagania sprzętowe i wobec systemu operacyjnego

Do uruchomienia konwertera wymagane są program MS Word 2010 lub nowszy, system operacyjny MS Windows XP lub nowszy oraz środowisko uruchomieniowe Java w wersji 7 lub nowszej.

Uruchomienie konwertera jest możliwe nawet na słabszych obliczeniowo komputerach, niemniej jednak będzie miało to wpływ na szybkość przetwarzania dokumentów, dlatego zaleca się skorzystać z jednostki opartej co najmniej o procesor Pentium IV 2 GHz z 1 GB pamięci operacyjnej oraz 300 MB wolnej przestrzeni na dysku.

Do skompilowania konwertera wymagana jest opisana powyżej konfiguracja uzupełniona o narzędzie PerlApp (np. w wersji 5.3.0 build 530) dystrybuowane przez ActiveState oraz ActivePerl (np. w wersji 5.8.9).

Instalacja i konfiguracja

Na komputerze użytkownika główny program konwertera jest pobierany i instalowany przez Dodatek MS Word.

Administrator udostępnia plik instalacyjny z wykorzystaniem protokołu HTTP. W trakcie pierwszego uruchomienia konwertera następuje pobranie oraz instalacja wszystkich modułów niezbędnych do przetwarzania dokumentów OOXML (zgodnie z deskryptorem repozytorium modułów konwertera), sprawdzanie dostępności aktualizacji następuje również przy każdym następnym uruchomieniu.

Proces kompilacji konwertera oraz przygotowania repozytorium został opisany w pliku README.txt. Plik ten jest dystrybuowany wraz ze źródłami konwertera.

Kompilacja

Plik wykonywalny konwertera został napisany w języku Perl i skompilowany do samodzielnej aplikacji z wykorzystaniem narzędzia PerlApp w wersji 5.3.0 build 530 dystrybuowanego przez firmę ActiveState. W trakcie procesu budowania dystrybucji należy dołączyć do niej narzędzie do dekompresji archiwów zip (7-ZIP).

Dodatkowo przed kompilacją pliku głównego konwerter_OOXML.pl należy zdefiniować w jego źródle:

- adresy URL wskazujące repozytoria modułów konwertera
- hasło zdalne - wykorzystywane do deszyfrowania pakietów pobieranych przez konwerter z repozytorium,
- hasło lokalne - wykorzystywana do zabezpieczania lokalnych plików konfiguracyjnych.

Repozytorium modułów

Do prawidłowego działania konwertera wymagane jest stworzenie repozytoriów modułów, dzięki którym możliwe jest dokończenie procesu instalacji oraz przeprowadzanie aktualizacji w

przyszłości. Konwerter przystosowany jest do współpracy z wieloma repozytoriami, jednak aby zapewnić ciągłość jego pracy, zaleca się stworzenie minimum dwóch niezależnych od siebie pod względem sieciowym oraz sprzętowym repozytoriów.

Repozytorium modułów składa się z deskryptora zawierającego informację dotyczącą modułów kompatybilnych z konkretną wersją konwertera oraz opcjonalnie modułów konwertera (moduły mogą być udostępniane z innych, niezależnych serwerów).

Deskryptor jest zaszyfrowany z wykorzystaniem hasła zdalnego (algorytm Blowfish) plikiem zawierającym tablicę asocjacyjną o nazwie REPOSITORY zdefiniowaną w języku Perl (ewaluowaną w trakcie wykonania). Kluczami tablicy asocjacyjnej deskryptora są skróty z plików wykonywalnych konwerter_OOXML.exe, zwane dalej wersją konwertera, natomiast wartościami odwołania do anonimowych tablic asocjacyjnych zawierających spis modułów kompatybilnych z daną wersją konwertera.

Tablica asocjacyjna będąca spisem modułów jako klucz wykorzystuje nazwę modułu, natomiast wartością jest anonimowa tablica asocjacyjna zawierająca deskryptor danego modułu.

W skład deskryptora każdego modułu wchodzą:

- URL - adres URL, spod którego serwowany jest plik modułu,
- ORDER - kolejność przetwarzania modułu w trakcie instalacji,
- DIGEST - skrót (MD5 zapisane w hex) z pliku modułu,
- VERSION - wersja modułu (do celów informacyjnych).

Ze względów bezpieczeństwa oraz w celu zapewnienia integralności moduły nie będące zewnętrznymi pakietami muszą zostać zaszyfrowane (algorytm Blowfish dla modułów jednoplikowych lub zaszyfrowane archiwum zip dla modułów składających się z większej liczby plików; w ramach konfiguracji należy ustawić wartości związane z tymi metodami szyfrowania modułów).

Lista modułów:

- FOP (moduł zewnętrzny) - Apache FOP w wersji 1.1 bin,
- OOXML2DOCBOOK (moduł jednoplikowy) - główny moduł XSLT,
- FOP_UPDATE (moduł archiwum) - dostosowanie Apache FOP,
- JEUCLID (moduł zewnętrzny) - JEuclid w wersji 3.1.9,
- MAIN_MODULE (moduł jednoplikowy) - główny moduł konwertera,
- SAXON (moduł zewnętrzny) - Saxon w wersji 9.1.0.8j,
- XSLT_PACKAGE (moduł archiwum) - dodatkowe pliki XSLT,
- CORE_CURRICULUM_MAP (moduł jednoplikowy) - mapa podstawy programowej,
- DOCBOOK_XSL (moduł zewnętrzny) - Docbook XSLT w wersji 1.77.1.

Edycja Online

Krótki opis komponentu
Edycja Online (EO) jest środowiskiem edycyjnym udostępnionym redaktorom e-podręczników w ramach komponentu portalu. W ramach EO możliwe są następujące działania:

- edycja kolekcji za pomocą dedykowanego edytora kolekcji lub za pomocą tekstowego edytora XML,
- edycja modułu treści za pomocą edytora treści online typu WYSIWYG lub za pomocą tekstowego edytora XML,
- podgląd formatu emisyjnego HTML edytowanych kolekcji i modułów,
- pieczętowanie, czyli zapisywanie wyedytowanych wersji kolekcji i modułów do Repozytorium Treści (RT).

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami
Komponent Edycji Online komunikuje się serwisem wymiany komunikatów za pomocą protokołu AMQP. EO wysyła komunikaty o zmianach w treściach kolekcji i modułów znajdujących się w przestrzeniach EO, na które reagują inne komponenty platformy. EO udostępnia API HTTP, za pomocą którego inne komponenty mogą uzyskać formaty źródłowe kolekcji i modułów znajdujących się w przestrzeniach EO. EO pieczętuje obiekty do RT za pomocą pakietu eprt-tools.

Instalacja i konfiguracja
Patrz „Instalacja i konfiguracja” portalu.

Serwis wymiany komunikatów pomiędzy komponentami

Krótki opis komponentu
Usługa służy do wymiany komunikatów pomiędzy poszczególnymi komponentami o zdarzeniach zachodzących na platformie. Przekazywane wiadomości dotyczą przede wszystkim zdarzeń wprowadzenia zmian w plikach źródłowych lub w formatach emisyjnych podręczników, WOMI czy zasobów KZD.

Funkcjonalność jest zrealizowana za pomocą aplikacji RabbitMQ Server.

Realizowane funkcjonalności
Usługa wymiany komunikatów zapewnia przede wszystkim, że:

- komponent nadający komunikat nie musi znać listy komponentów, które odbierają komunikat,
- komunikat zostanie dostarczony do każdego komponentu-odbiorcy, który musi go otrzymać, nawet gdy w momencie wysłania komunikatu komponent-odbiorca jest wyłączony lub nie działa prawidłowo.

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami
Inne komponenty komunikują się z usługą wymiany komunikatów za pomocą protokołu AMQP.

Wymagania sprzętowe i wobec systemu operacyjnego

Zgodnie z wymaganiami RabbitMQ Server.

Instalacja i konfiguracja

Instalacja i podstawowa konfiguracja przebiega zgodnie z dokumentacją RabbitMQ Server. Ponadto w konfiguracji usługi należy zdefiniować:

- jeden wspólny *exchange* dla wszystkich komponentów,
- odrębne kolejki (*queues*) dla każdego komponentu, który odczytuje komunikaty z tej usługi,
- odrębne dane uwierzytelniające, pozwalające na odczyt komunikatów z danej kolejki tylko uprawnionym komponentom,
- zasady routingu komunikatów (wiadomości) pomiędzy kolejkami.

Usługa wyszukiwania treści

Krótki opis komponentu

Wyszukiwanie jest wewnętrznym modułem systemu e-podręczniki wykorzystującym do realizacji swojej funkcjonalności repozytoria indeksów pełnotekstowych uruchomione jako oddzielne usługi. Moduł wyszukiwania treści obejmuje swoją funkcjonalnością usługi przeznaczone zarówno do odnajdywania elementów w kontekście Katalogu Zasobów Dodatkowych, jak i Biblioteki e-podręczników. Realizacja wyszukiwania odbywa się w dwóch niezależnych obszarach: wyszukiwania metaopisów elementów Katalogu Zasobów Dodatkowych oraz wyszukiwania pełnotekstowego w treści e-podręczników oraz elementów w nich wykorzystywanych. Zadaniem komponentu jest więc z jednej strony przyjmowanie nowych bądź zaktualizowanych treści, a z drugiej strony realizacja wyszukiwania w zaindeksowanym zbiorze.

Realizowane funkcjonalności

Moduł odpowiedzialny za wyszukiwanie realizuje dwie podstawowe funkcjonalności. Pierwsza, związana ze zbieraniem danych, oznacza proces indeksowania treści i metadanych. Zbudowane w ten sposób repozytorium wykorzystywane jest na etapie udostępniania wyników wykonywanych zapytań jemu zlecanych.

Z uwagi na specyfikę dokumentów przeznaczonych do przechowywania w indeksie pełnotekstowym utworzono dwa niezależne repozytoria – oddzielnie dla Katalogu Zasobów Dodatkowych (przechowujących głównie metadane opisane w formacie JSON o obiektach) oraz odrębne dla treści podręczników wzbogaconych o sformalizowane metadane dla poszczególnych kolekcji. Do przeprowadzenia wypełnienia treścią każdego z repozytoriów przygotowane zostały odpowiednie procesy uruchamiane automatycznie w odpowiedzi na zdarzenia pojawiające się w systemie (jak np. opublikowanie podręcznika) bądź ręcznie przez administratora (np. w przypadku konieczności odbudowy repozytorium indeksu). Indeksowaniu podlegają deskryptory obiektów dla Katalogu Zasobów Dodatkowych oraz pełne pliki źródłowe kolekcji i modułów dla wyszukiwania pełnotekstowego.

Repozytoria indeksów są udostępniane poprzez funkcjonalność wykonywania zapytań. Przyjęto przy tym formalny opis zapytania przekazywanego do wyszukiwania w postaci deskryptora zapytania przekazywanego w postaci JSON dla wyszukiwania opartego na atrybutach, natomiast wyszukiwanie

pełnotekstowe realizowane jest poprzez przekazanie wprost frazy do odnalezienia. Wyniki zwracane są w sposób sformalizowanej, przetworzonej listy bądź interpretowane bezpośrednio.

Wyszukiwanie pełnotekstowe odbywa się w kontekście całej platformy, konkretnego podręcznika (kolekcji) bądź w obrębie podręczników z wybranego poziomu edukacyjnego i klasy. Wyszukiwanie pełnotekstowe w obrębie platformy realizowane jest zarówno z poziomu strony głównej Portalu (przycisk „Wyszukaj w epodręcznikach") jak i poprzez odpowiedni przycisk na pasku narzędzi w górnej części okna przeglądarki. W przypadku podglądu szczegółów konkretnego podręcznika jak i podczas przeglądania (czytania) jego zawartości w Czytniku funkcja wyszukiwania realizowana jest w obrębie bieżącego podręcznika.

Wyszukiwanie kontekstowe natomiast wywoływane jest poprzez Katalog Zasobów Dodatkowych bądź odpowiedni przycisk.

W każdym z tych przypadków wyszukiwanie realizowane jest poprzez odnajdywanie podanej przez użytkownika znormalizowanej frazy (pozbawionej znaków cudzysłowu i ewentualnych elementów sterujących procesem wyszukiwania).

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Wyszukiwanie pełnotekstowe w platformie

Wyszukiwanie pełnotekstowe wykorzystuje instancję silnika indeksu pełnotekstowego SOLR, który przechowuje w zdefiniowanym schemacie dane na potrzeby realizacji kwerend wyszukiwawczych. Dodatkowo do dodawania nowych treści wykorzystywana jest zewnętrzna aplikacja webowa indeksera, który parsując odpowiednie pliki źródłowe podręczników, ich modułów, czy WOMI, przygotowuje dane do umieszczenia w indeksie pełnotekstowym. Aplikacja ta jest aplikacją webową, która może być uruchomiona na dowolnym serwerze aplikacyjnym (w szczególności tożsamym z serwerem aplikacyjnym służącym do uruchomienia samego silnika SOLR).

Zdefiniowane w obrębie platformy zadania, jak i komendy z nich korzystające, pozwalają na przebudowanie indeksu zarówno ręcznie przez administratora, a także w ramach automatycznego uspójniania stanu repozytorium.

Wyszukiwanie obiektów w Katalogu Zasobów Dodatkowych

Wyszukiwanie obiektów na potrzeby Katalogu Zasobów Dodatkowych wykorzystuje silnik indeksu pełnotekstowego Elastic (Elasticsearch). Instancja tej usługi musi być uruchomiona na oddzielnym zestawie maszyn, które tworzą pojedynczy klaster indeksu, co zapewnia odpowiedni poziom dostępności usługi i reduncancję danych w przypadku awarii.

Interfejs programistyczny (API) realizuje funkcjonalność założenia/usuwania indeksu pełnotekstowego na potrzeby wyszukiwania, indeksowania danych oraz realizacji wyszukiwania. Interfejs jest zrealizowany w metodologii REST z użyciem protokołu HTTP.

Wymagania sprzętowe i wobec systemu operacyjnego

W ramach platformy wykorzystywany jest SOLR w wersji 4.4.

Wymagania wobec infrastruktury

Na uruchomienie każdego z komponentów Elastic i SOLR należy przeznaczyć co najmniej jedną maszynę. Minimalne wymagania wobec pojedynczej maszyny to:

- Liczba CPU: 4
- Ilość pamięci RAM: 8 GB
- Rozmiar dysku: 160 GB

Wymagania wobec systemu operacyjnego

Instalacja Elastic

Oracle Java JDK 6 lub nowsza (najlepiej jednak JDK 7 u55 bądź JDK8)

Wyłączony systemowy plik wymiany.

Instalacja SOLR

Oracle Java JDK 6 lub nowsza (najlepiej jednak JDK 7 u1 bądź JDK8)

Serwer aplikacyjny Java np. Apache Tomcat 6.

Instalacja i konfiguracja

Do prawidłowego działania systemu niezbędne jest uruchomienie instancji repozytorium indeksu opartego o usługi serwera Apache Solr (dla wyszukiwania pełnotekstowego) oraz Elasticsearch (dla wyszukiwania w obrębie Katalogu Zasobów Dodatkowych), a także aplikacji indeksera treści zajmującego się parsowaniem plików wejściowych na potrzeby wyszukiwania i przygotowującego dokumenty przechowywane w indeksie SOLR.

Instalacja komponentów SOLR i Elastic przebiega zgodnie z dokumentacją tych aplikacji.

Kompilacja aplikacji indeksera przebiega przy użyciu narzędzia Maven. Skompilowany plik war instaluje się na serwerze aplikacyjnym (Apache Tomcat), najlepiej tożsamym z instancją przeznaczoną dla usługi SOLR. W plikach konfiguracyjnych aplikacji należy odpowiednio ustawić adres instacji SOLR.

Usługa kont użytkowników

Usługa kont użytkowników pozwala zarządzanie tożsamościami użytkowników w systemie epodreczniki.pl. W ramach usługi można wyróżnić dwa zasadnicze komponenty

- komponent odpowiedzialny za tworzenie tożsamości użytkownika
- komponent odpowiedzialny za uwierzytelnianie użytkownika

Szczegółowy opis każdego z komponentów znajduje się poniżej.

Usługa rejestracji użytkowników

Krótki opis komponentu

Usługa rejestracji użytkowników pozwala na utworzenie nowej tożsamości użytkownika na platformie. Jest to usługa komplementarna do usługi uwierzytelniania użytkowników.

Realizowane funkcjonalności

Zakres funkcjonalności świadczonej na rzecz innych komponentów systemu epodreczniki.pl:

- utworzenie nowej tożsamości użytkownika
- aktywacja konta użytkownika (automatyczna, możliwa aktywacja przez administratora)
- test czy użytkownik o zadanym identyfikatorze już istnieje w systemie
- zmiana danych wchodzących w skład tożsamości użytkownika
- zmiana hasła użytkownika (reset hasła)

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Usługa rejestracji użytkowników do realizacji swojej funkcjonalności wykorzystuje następujące zewnętrzne usługi:

Instancja Portalu

- Usługa bazuje na wewnętrznych mechanizmach Portalu i jest z nim ściśle związana.

Baza danych

- Baza danych pozwala rejestrować żądania utworzenia tożsamości użytkowników.
- Dostęp do bazy danych realizowany jest poprzez standardowe interfejsy i sterowniki portalu.
- Wykorzystywana baza danych: Oracle 11g Enterprise Edition Release 11.2

Usługa poczty elektronicznej

- Usługa poczty elektronicznej pozwala na wysyłanie do użytkownika korespondencji zawierającej informacje niezbędne do poprawnej aktywacji konta
- Dostęp do usługi poczty elektronicznej realizowany jest za pomocą protokołu SMTP
- Wykorzystywana implementacja usługi poczty elektronicznej: dowolna

Wymagania sprzętowe i wobec systemu operacyjnego

Minimalne wymagania wobec infrastruktury:

- Usługa bazuje na instancji Portalu dlatego dziedziczy z niej wymagania.

Instalacja i konfiguracja

W ramach konfiguracji aplikacji (konfiguracja zmiennych konfiguracyjnych portalu) ustawia się parametry związane z bezpieczeństwem, do bazy danych, do serwera SMTP oraz do systemu wymiany komunikatów pomiędzy komponentami.

Usługa uwierzytelniania użytkowników

Krótki opis komponentu

Usługa uwierzytelniania użytkowników jest odpowiedzialna za potwierdzenie tożsamości użytkownika przedstawiającego stosowne poświadczenie w postaci nazwy użytkownika oraz hasła.

Usługa realizuje koncepcję pojedynczej tożsamości użytkownika, współdzielonej między wieloma komponentami systemu (np. Portal, CMS, Edycja Online).

Realizowane funkcjonalności

Funkcjonalności realizowane przez usługę uwierzytelnienia użytkowników są określone poniżej:

- uwierzytelnienie użytkownika
- zwrócenie ciasteczka sesji dla aplikacji potrzebujących poświadczeń
- udostępnienie informacji z profilu użytkownika
- zakończenie sesji - wylogowanie użytkownika poprzez unieważnienie wszystkich ciasteczek

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Usługa uwierzytelniania użytkowników jest częścią Portalu. Do działania potrzebuje instancji Portalu.

Wymagania sprzętowe i wobec systemu operacyjnego

Minimalne wymagania wobec infrastruktury:

- Jako część Portalu dziedziczy jego wymagania.

Instalacja i konfiguracja

Instalacja odbywa się podczas instalacji portalu, a także potrzebuje do działania ustawienia jego zmiennych uruchomieniowych.

Kontekst użytkownika (postęp zadań, notatki)

Krótki opis komponentu

Kontekst użytkownika jest komponentem realizującym zagadnienie personalizacji treści oraz środowiska pracy na platformie. Dotyczy on zarówno samych treści, jak i sposobu nawigacji po portalu.

Realizowane funkcjonalności

- Notatki oraz zakładki: oznaczanie treści oraz anotowanie ich komentarzami użytkownika. Komentarze mogą być dodatkowo spersonalizowane graficznie oraz mogą zawierać linki do zasobów zewnętrznych.
- Profil użytkownika: osobista półka na książki - ostatnio przeglądane podręczniki.
- Profil użytkownika: zagregowany widok notatek - przegląd wszystkich notatek użytkownika.
- Profil użytkownika: edycja danych użytkownika - umożliwia zmianę danych użytkownika takich jak dane osobowe, awatar.
- Profil użytkownika: lista adresów e-mail nauczycieli skojarzonych z danym przedmiotem - tak, aby użytkownik mógł automatycznie dzielić się wynikami ćwiczeń.

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Kontekst użytkownika zależy bezpośrednio od treści epodręczników. Wspiera też współdzielenie wyników ćwiczeń.

Wymagania sprzętowe i wobec systemu operacyjnego

Komponent jest częścią portalu i współdzieli z nim wymagania sprzętowe.

Instalacja i konfiguracja

Baza danych

Profil użytkownika korzysta z bazy danych nosql Apache Cassandra. Zaleca się zainstalowanie oraz skonfigurowanie klastra z bazą z liczbą węzłów pozwalającą na bezpieczny poziom replikacji minimum 3 oraz odpowiednią odpornością na ewentualne awarie poszczególnych węzłów. Oprócz replikacji przewiduje się też automatyczny backup węzłów wraz z danymi.

Instalacja i konfiguracja przebiega zgodnie z dokumentacją Apache Cassandra.

Po skonfigurowaniu klastra należy stworzyć schemat bazy danych wykorzystywany przez profil użytkownika oraz skonfigurować użytkownika dostępowego, a także *keyspace* w ustawieniach portalu.

Wersja apache cassandra:

- 2.0.x lub 2.1.x w wersji stabilnej (tj. x powyżej 6)

Minimalne wymagania sprzętowe:

Dla konfiguracji jednoserwerowej zalecana minimalna konfiguracja:

- 4 GB RAM
- 2-rdzeniowy CPU
- 100 GB HDD (RAID-1)

Portal (CMS + Biblioteka + Czytnik + KZD)

Krótki opis komponentu

Portal - warstwa prezentacji treści jest głównym składnikiem platformy epodręczniki. Odpowiada za graficzną prezentację modeli danych, treści w formatach źródłowych, a także zapewnia warstwę nawigacji na poziomie strony web. Na prezentację treści składa się kilka obszarów, do których użytkownik ma dostęp za pomocą odpowiednich widoków. Głównymi elementami tej warstwy są:

- *CMS* - system zarządzania treścią, który pozwala na przygotowanie strony startowej platformy epodręczniki oraz na dodawanie materiałów informacyjno-promocyjnych.
- *Biblioteka* - odpowiada za prezentację listy podręczników, biorąc pod uwagę podział podręczników ze względu na poziomy nauczania i klasy. Zawiera ona filtry, po których można wyszukiwać podręczniki oraz udostępnia widok szczegółów podręcznika.
- *Czytnik* - odpowiada za prezentację treści podręczników. Oferuje responsywny interfejs, który dostosowuje treści do urządzenia użytkownika (treści podlegają przekształceniom wizualnym i funkcjonalnym w zależności od możliwości urządzenia), a także dostarcza funkcjonalność związaną z ułatwieniami dostępności treści (WCAG).

- *Katalog Zasobów Dodatkowych* - interfejs pozwalający na przeglądanie i wyszukiwanie zasobów, które nie są bezpośrednio związane z podręcznikami, ale mają istotne znaczenie dydaktyczne.
- *Publikacja treści* - interfejs pozwalający na trwałe opublikowanie formatów źródłowych oraz emisyjnych w portalu celem późniejszej prezentacji.

Realizowane funkcjonalności

CMS

- Strona główna systemu epodręczniki.
- Funkcje zarządzania dostępne dla administratorów systemu.
- Dodawanie aktualności związanych z projektem/systemem.
- Dodawanie informacji o wydarzeniach (kalendarz).
- Dodawanie zasobów takich jak zdjęcia, wygaszacze ekranu, kalendarze, dokumenty (materiały informacyjno-promocyjne).
- Łączenie z innymi zasobami i komponentami systemu epodręczniki (Biblioteka, KZD itp.).

Biblioteka

- Prezentacja i filtrowanie listy dostępnych (opublikowanych) podręczników.
- Uwzględnienie poziomów nauczania i klas.
- Prezentacja szczegółowych danych podręcznika, takich jak: spis treści, formaty emisyjne, okładka, autorzy.

Czytnik

- Prezentowanie treści podręcznika w sposób responsywny i spójny dla różnych urządzeń końcowych.
- Wsparcie WCAG 2.0 na poziomie treści oraz nawigacji.
- Nawigacja po podręczniku.
- Prezentacja treści multimedialnych zawartych w podręczniku.
- Elementy wizualne poprawiające czytelność treści.
- Interfejs użytkownika stworzony w sposób minimalistyczny, dostarczający maksimum funkcjonalności.
- Mechanizmy dodawania notatek do podręcznika oraz zapisu stanu poszczególnych zadań.
- Możliwość wykorzystania czytnika zarówno do wyświetlania treści opublikowanych, jak i nieopublikowanych (treści w podglądzie podlegających redakcji i korekcie).

Katalog Zasobów Dodatkowych (KZD)

- Możliwość filtrowania zasobów wg kategorii i poziomu nauczania.

- Możliwość pełnotekstowego przeszukiwania zasobów.
- Prezentacja atrybutów poszczególnych zasobów.
- Odtwarzanie zasobów audio, wideo i interaktywnych bezpośrednio w portalu (w przeglądarce internetowej) oraz możliwość pobrania wybranych zasobów na dysk lokalny.

Proces publikacji treści

- Możliwość zlecenia i obserwacji zadania publikacji treści.
- Możliwość dodawania do biblioteki i usuwania z niej opublikowanych podręczników.
- Kopiowanie formatów źródłowych i emisyjnych treści z Repozytorium Treści i Serwera Kodującego na serwery hostujące pliki opublikowanych podręczników.

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Elementy prezentacji treści służą do przetwarzania treści, która znajduje się w innych elementach systemu. Ten komponent korzysta ze wszystkich istotnych komponentów platformy.

Baza danych Portalu

Istotnym elementem składowym tego komponentu jest baza danych, w której przechowywane są dane potrzebne do prezentacji treści.

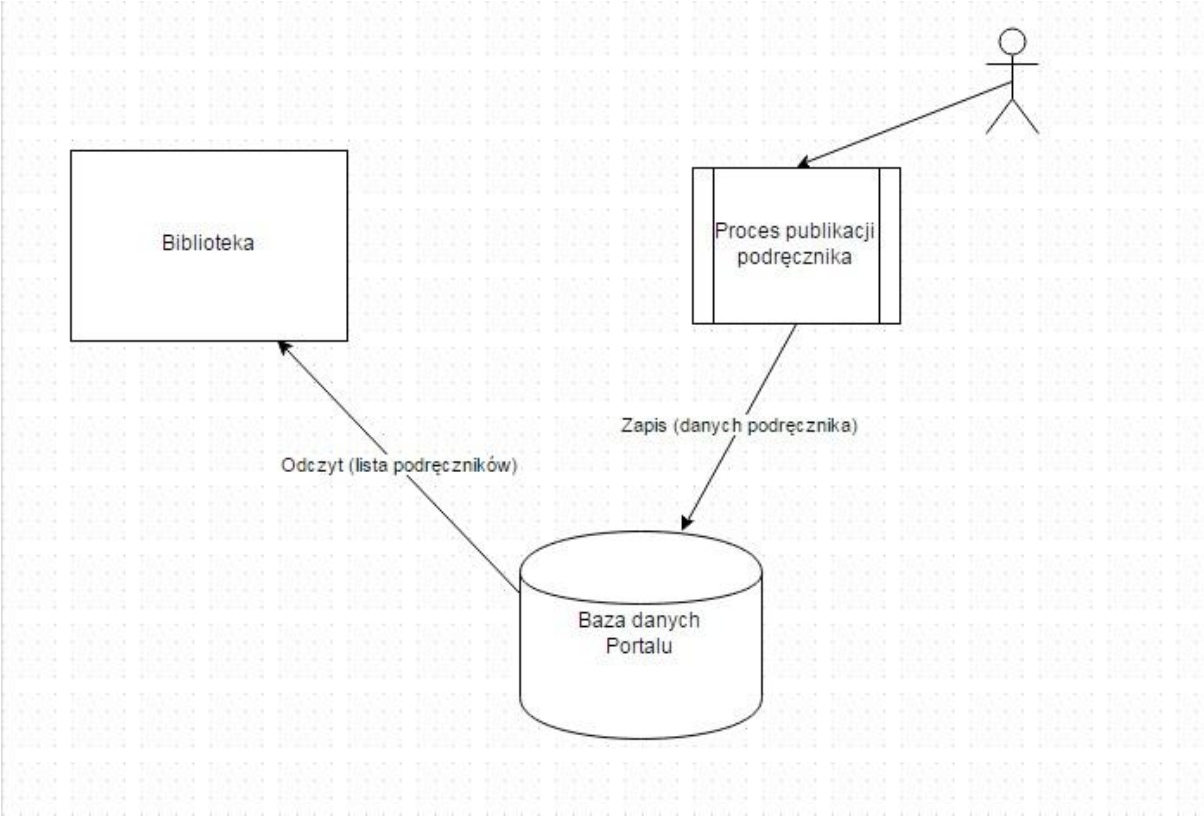
Baza danych jest systemem relacyjnym SQL.

CMS

Ten element systemu nie komunikuje się z innymi. Do poprawnego działania potrzebuje bazy danych, wykorzystywanej także w innych elementach portalu.

Biblioteka

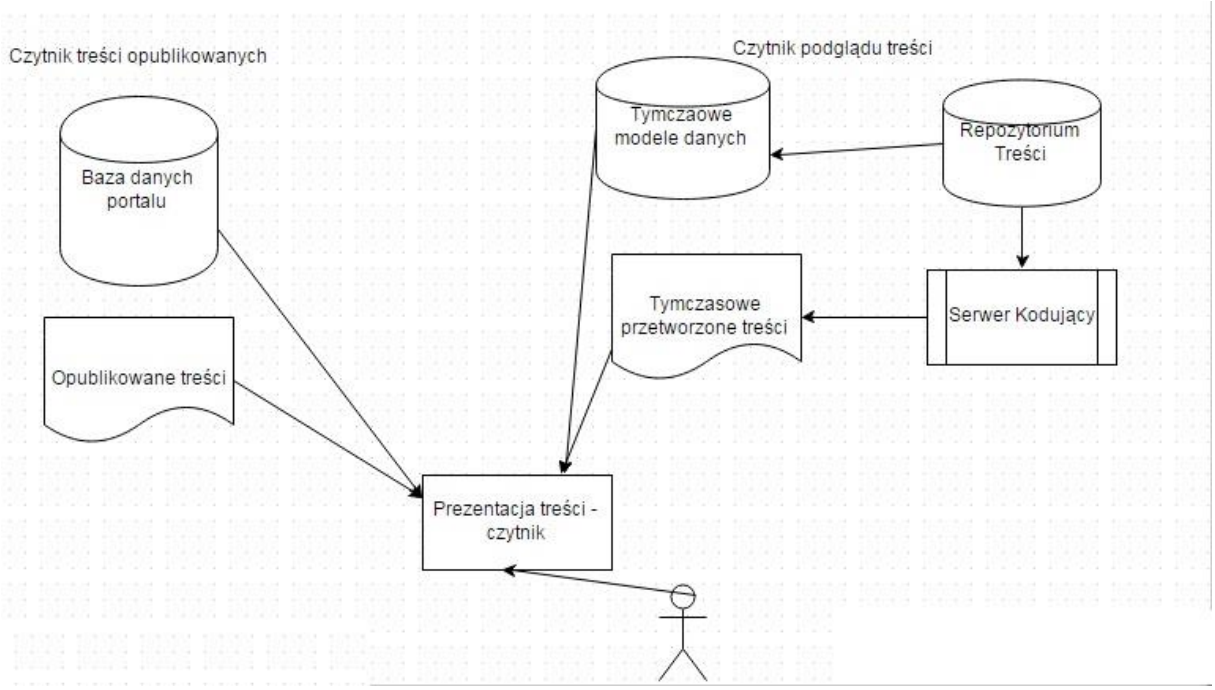
Biblioteka korzysta z bazy danych Portalu. Procesy związane z Biblioteką tylko odczytują dane z bazy. Zapis odbywa się tylko i wyłącznie podczas procesu publikacji (obiekty są niezmienne po tym procesie).



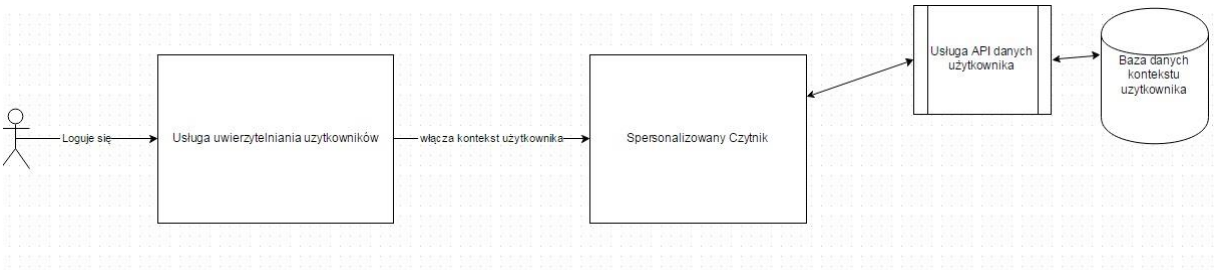
Czytnik

Czytnik to moduł prezentujący treść w przeglądarce internetowej. Treść na potrzeby Czytnika może być pobierana z dwóch różnych źródeł:

1. W podstawowym założeniu Czytnik prezentuje treści opublikowane, które przeszły procesy opisane w sekcji związanej z Biblioteką. W tym aspekcie korzysta jedynie z Bazy danych Portalu.
2. Czytnik może działać w trybie Podglądu treści, czyli wyświetlać nieopublikowane treści znajdujące się w Repozytorium Treści oraz na Serwerze Kodującym. Ten proces jest znacznie bardziej złożony i wymaga do współpracy komponentów: Repozytorium Treści, Serwer Kodujący, Dokumentowa Baza Danych (REDIS).



Czytnik służy również do personalizacji podręczników przez użytkowników. Komponenty odpowiedzialne za uwierzytelnianie i przechowywanie danych użytkowników pozwalają na tworzenie osobistych notatek do treści oraz zapis stanu zadań przez zalogowanych użytkowników.



Czytnik pozwala na odtwarzanie treści multimedialnych specjalnie dostosowanych do potrzeb autorów treści i komunikujących się z systemem poprzez interfejs programistyczny (API).

Katalog Zasobów Dodatkowych

Katalog Zasobów Dodatkowych korzysta z zasobów znajdujących się w Repozytorium Treści. Zasoby te są specjalnie oznaczonymi strukturami WOMI, dzięki czemu mogą być katalogowane jako zasoby dodatkowe.

Do katalogowania tych zasobów wymagany jest komponent Wyszukiwania treści, który pozwala zebrać metadane ze wszystkich dostępnych zasobów w Repozytorium Treści i na ich podstawie wyświetlać listę zasobów zawężoną parametrami wyszukiwania.

Proces publikacji treści

Po zleceniu publikacji przez użytkownika system publikacji rozpoczyna kopiowanie formatów źródłowych treści (z Repozytorium Treści) oraz formatów emisyjnych (z Serwera Kodującego). Po ich

skopiowaniu część zawartych w nich informacji zostaje zaimportowana do relacyjnej bazy danych, skąd później są one wykorzystywane przez bibliotekę oraz czytnik.

Wymagania sprzętowe i wobec systemu operacyjnego

Wymagania systemowe:

- System operacyjny Linux
- Python w wersji 2.7.x
- Wsparcie dla virtualenv
- Możliwość kompilacji bibliotek pythona (GCC)
- Przy powyższych założeniach wszelkie wymagane biblioteki powinny zostać pobrane przez PIP z konfiguracji projektu
- Serwer HTTP obsługujący WSGI (zalecany uWSGI)
- Serwer REDIS
- Baza danych SQL (zalecana Oracle)

Minimalne wymagania sprzętowe:

- Procesor klasy Intel Core Duo lub wyższy
- 2 GB RAM
- ok. 5 GB przestrzeni dyskowej + przestrzeń niezbędna do przechowania opublikowanych podręczników

Instalacja i konfiguracja

Portal napisany jest w języku Python 2.7.x z wykorzystaniem Django. Portal wymaga zainstalowania w systemie następujących aplikacji lub bibliotek:

- Python 2.7.x
- virtualenv
- kompilator gcc
- kompilator less
- Git w wersji 1.8.4 (lub nowszej)
- python-dev
- python-pip
- libxml-devel
- libxslt-devel
- libffi-devel
- openssl-devel (czasem pod nazwą libssl-dev)

- libjpeg-devel

W ramach utworzonego wirtualnego środowiska (z użyciem narzędzia virtualenv), przeznaczonego do uruchomienia aplikacji Portalu, należy na podstawie plików wejściowych dla narzędzia pip zainstalować wszystkie pozostałe zależności.

Po spełnieniu w/w wymagań należy skonfigurować szereg ustawień aplikacji Portalu (m.in. sposób połączenia do pozostałych komponentów platformy i do bazy danych, mapowania hostów /etc/hosts, lokalizacji katalogów roboczych), wygenerować schemat bazy danych i uruchomić samą aplikację.

Aplikacje mobilne

Krótki opis komponentu

E-podręczniki LITE jest mobilnym narzędziem wspomagającym pracę z platformą epodreczniki.pl. Aplikacja umożliwia pracę z zasobami e-podręczników na urządzeniach mobilnych z systemami Android, iOS, Windows 8 oraz Windows Phone 8 w sposób niewymagający stałego dostępu do Internetu.

Realizowane funkcjonalności

- wyświetlanie listy e-podręczników
- wyświetlanie szczegółowych informacji na temat e-podręczników
- pobieranie i zapisywanie w pamięci urządzenia e-podręczników do pracy w trybie offline
- zarządzanie e-podręcznikami (usuwanie, uaktualnianie)
- praca z e-podręcznikiem (wyświetlanie treści, praca z zadaniami interaktywnymi, dodawanie notatek i zakładek)
- w przypadku współdzielenia urządzenia (np. tablet w szkole) aplikacja może pracować w trybie wielu użytkowników (każdy użytkownik ma własne notatki, zakładki oraz ustawienia)

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Aplikacje mobilne komunikują się z interfejsem programistycznym (API) Portalu, z którego uzyskują informacje o istniejących podręcznikach oraz pobierają treści wybranych podręczników. Interfejs jest zrealizowany w metodologii REST z użyciem protokołu HTTP.

Wymagania sprzętowe i wobec systemu operacyjnego

- Android - aplikacja wymaga systemu operacyjnego Android w wersji 4.0 lub nowszej
- iOS - aplikacja wymaga systemu operacyjnego iOS w wersji 7.0 lub nowszej
- Windows - aplikacja wymaga systemu operacyjnego Windows w wersji 8.1 lub nowszej
- Windows Phone - aplikacja wymaga systemu operacyjnego Windows Phone w wersji 8.0 lub nowszej

Wybrane funkcje, np. pobieranie i zapisywanie w pamięci urządzenia treści e-podręczników, wymagają połączenia z Internetem na urządzeniu.

Instalacja i konfiguracja

Instalacja na urządzeniu końcowym użytkownika przebiega z wykorzystaniem repozytoriów aplikacji dla danego systemu operacyjnego:

- Android - Google Play https://play.google.com/store/apps/details?id=pl.epodreczniki
- iOS - iTunes App Store https://itunes.apple.com/app/e-podreczniki-lite/id878974304?mt=8
- Windows - Windows Store http://apps.microsoft.com/windows/app/597c8720-b51d-464a-be5c-6f510f67e98c
- Windows Phone - Windows Store http://windowsphone.com/s?appid=efcb0b39-40a6-4764-bfe1-b0ef7b86bddf

W przypadku konieczności opublikowania nowej wersji aplikacji należy:

- Android (korzystając z Eclipse Android Developer Tools)
  - wyeksportować niepodpisany plik apk
  - korzystając z programu jarsigner (będącego częścią pakietu Oracle Java SE Development Kit) podpisać plik apk kluczem ORE (należy zastosować algorytm podpisu elektronicznego SHA1withRSA oraz SHA1 jako funkcję skrótu)
  - zoptymalizować plik apk korzystając z programu zipalign (będącego cześcią Android SDK)
  - tak przygotowany plik można opublikować w sklepie Google Play
- iOS (korzystając z Xcode 6.4 lub nowszego)
  - wybierając z menu Xcode > Preferences > Accounts, dodać konto deweloperskie Apple Developer Program iOS
  - upewnić się, że ustawione są odpowiednie profile Provisioning Profiles, klikając w Project Navigator > EpodrecznikiApp > Build Settings > Code Signing
  - przygotować aplikację do sklepu poprzez wybranie polecenia Archive z menu Product
  - w organizerze archiwów wybrać Submit to App Store
  - postępować zgodnie z wytycznymi kreatora
- Windows 8 (Visual Studio w wersji 2013 lub nowszej)
  - wybrać z menu: Project > Store > Create App Packages
  - nie wybierać opcji przesłania pakietów bezpośrednio do *Windows Store*
  - utworzyć pakiet we wskazanej lokalizacji na dysku lokalnym
  - zalogować się na konto ORE w *Windows Dev Center*
  - odnaleźć aplikację "E-podręczniki LITE"
  - na liście "Submissions" wybrać opcję "Update"

    - na podstronie "Submission" wybrać opcję "Packages" i przesłać zapisany lokalnie pakiet aplikacji
    - kliknąć przycisk "Submit to the Store"
- Windows Phone 8 (Visual Studio 2013)
    - skompilować projekt w trybie Release
    - zalogować się na konto ORE w *Windows Dev Center*
    - odnaleźć aplikację "E-podręczniki LITE"
    - na liście "Submissions" wybrać opcję "Update"
    - przesłać skompilowany plik XAP aplikacji i potwierdzić aktualizację aplikacji

Aplikacja off-line dla treści do nauczania wczesnoszkolnego

Krótki opis komponentu

E-podręczniki LITE do nauczania wczesnoszkolnego jest aplikacją umożliwiającą dostęp do treści e-podręczników z zakresu nauczania wczesnoszkolnego w sposób niewymagający dostępu do Internetu. Aplikacja przeznaczona jest na urządzenia z systemem operacyjnym Windows 7, 8, 8.1 oraz 10.

Realizowane funkcjonalności

- wyświetlanie listy e-podręczników dla treści do nauczania wczesnoszkolnego
- wyświetlanie szczegółowych informacji na temat e-podręczników
- instalowanie w lokalnym systemie dodatkowych e-podręczników dla treści do nauczania wczesnoszkolnego, do pracy w trybie offline, pochodzących z dedykowanych paczek w formacie zip
- zarządzanie e-podręcznikami (usuwanie)
- praca z e-podręcznikiem (wyświetlanie treści)

Interfejsy do innych komponentów i powiązania z zewnętrznymi usługami

Aplikacja wraz z e-podręcznikami instaluje się w sposób kompletny w systemie użytkownika i nie wymaga komunikacji z zewnętrznymi usługami.

Kompilacja nowej odsłony aplikacji przez administratora, w szczególności z powodu aktualizacji treści e-podręczników, wymaga pozyskania treści podręczników za pośrednictwem innych komponentów platformy.

Wymagania sprzętowe i wobec systemu operacyjnego

Skompilowana aplikacja dla użytkownika wymaga komputera PC z zainstalowanym systemem operacyjnym Windows 7, 8, 8.1 lub 10.

Dla wygenerowania treści e-podręczników na potrzeby wydania nowej odsłony aplikacji niezbędna jest działająca instancja Portalu, odrębną od instancji produkcyjnej, prawidłowo komunikującą się z Serwerem Kodującym i Repozytorium Treści. Do kompilacji aplikacji i przygotowania pliku instalacyjnego należy mieć system Windows ze środowiskiem .NET.

Instalacja i konfiguracja

Instalacja na komputerze użytkownika

Instalację na komputerze użytkownika przeprowadza się poprzez uruchomienie instalacyjnego pliku exe, przechodząc przez standardowy kreator instalacji.

Konfiguracja i przygotowanie nowej odsłony aplikacji przez administratora

Przygotowanie nowej odsłony aplikacji, w szczególności ze zaktualizowaną treścią e-podręczników, wymaga pozyskania nowych treści za pomocą innych komponentów systemu oraz kompilacji nowej wersji aplikacji.

Pozyskanie treści

Do pozyskania treści e-podręczników dla aplikacji należy:

- w instancji Portalu wywołać odpowiednią komendę administracyjną, która dla zadanych parametrów identyfikujących podręcznik wygeneruje zestaw statycznych plików Czytnika, do osadzenia w aplikacji;
- pobrać strukturę katalogową WOMI, które są używane w danym podręczniku, oraz pliki HTML modułów tego podręcznika; pliki można uzyskać, kopiując je z serwera z plikami statycznymi tego podręcznika opublikowanego już w Bibliotece Portalu lub też wygenerować listę tych plików na bazie Podglądu, korzystając z narzędzi administracyjnych Portalu, i wtedy pobrać te pliki z Repozytorium Treści

Kompilacja aplikacji

Aplikacja jest przygotowana w Windows Presentation Foundation, a do jej skompilowania potrzebne jest środowisko Microsoft Visual Studio. Pozyskane treści e-podręczników, które mają być dostępne w aplikacji, osadza się w odpowiednim miejscu struktury projektu, przeprowadza się kompilację z wykorzystaniem Visual Studio i wreszcie generuje się instalator aplikacji, który udostępnia się użytkownikom końcowym.

Usługa utrzymania podwyższonego poziomu bezpieczeństwa

Konieczne jest ciągłe i wszechstronne monitorowanie poziomu zabezpieczeń infrastruktury wytworzonej w ramach projektu, w tym nie tylko zapewnienie maksymalnie wysokiego poziomu odporności na ataki, ale i właściwej jakości reakcji na wykryte albo zgłoszone incydenty. Na usługę składają się następujące elementy:

1. Zabezpieczenia fizyczne infrastruktury - jako część bieżącej pracy operacyjnej
    a. Ciągły monitoring przestrzeni przy pomocy systemu telewizji przemysłowej (CCTV)
    b. Wdrożony system kontroli dostępu do pomieszczeń

c. Wdrożony System Sygnalizowania Włamania i Napadu (SSWiN)

d. Całodobowy nadzór pracowników ochrony nad obiektem

e. Wdrożony system ochrony przeciwpożarowej (detekcji oraz gaszenia pożaru)

2. Bieżąca weryfikacja poziomu zabezpieczeń

   a. Szkolenia dla osób odpowiedzialnych za bezpieczeństwo (a priori, a posteriori)

      i. Szkolenia a priori - dla administratorów wdrażających nową technologię, dla programistów przygotowujących oprogramowanie. Przed szkoleniami przeprowadzono wewnętrzną ankietę wśród uczestników w zakresie proponowanego zakresu (konsultacje na temat zakresu realizowano także w trakcie pierwszych 2 szkoleń). Przygotowano i przeprowadzono ogółem 3 całodzienne szkolenia:

         - I szkolenie - maj 2013. Tematyka: znaczenie bezpiecznego programowania dla projektu informatycznego, podstawy modelowania zagrożeń, filtrowanie danych najważniejszą techniką ochrony aplikacji, wprowadzenie do specyficznych aspektów bezpieczeństwa technologii Web, zagrożenia związane ze zbieraniem informacji o systemie, ochrona aplikacji Web przed najważniejszymi klasami błędów, mechanizmy wymiany plików specyficznym problemem bezpieczeństwa w aplikacjach Web

         - II szkolenie - grudzień 2013. Tematyka: bezpieczne programowanie z użyciem Python i Django, .NET - narzędzia do analizy bezpieczeństwa kodu i konfiguracji środowiska, .NET a OWASP Top 10, warsztat z ochrony aplikacji przed złośliwymi danymi

         - III szkolenie - lipiec 2014. Tematyka: metodyki ręcznej analizy kodu źródłowego, bezpieczne programowanie na urządzenia mobilne, problemy kodowania danych a bezpieczeństwo oprogramowania, implementacja bezpiecznych mechanizmów uwierzytelniania

      ii. Szkolenia a posteriori - z rezultatów audytów okresowych (dla administratorów infrastruktury, dla programistów) - także uwzględniane w trakcie szkoleń zrealizowanych po pierwszych audytach bezpieczeństwa

   b. Ciągły monitoring poziomu zabezpieczeń
   Monitorowanie listy wykorzystywanych technologii oraz bieżąca weryfikacja, czy nie pojawiły się - pomiędzy testami okresowymi - znaczące błędy bezpieczeństwa (przykład: OpenSSL Heartbleed, SSL Poodle) albo istotne zmiany w zaleceniach bezpieczeństwa dla tych technologii. Realizacja testów w zakresie wynikającym z ww. kwestii oraz wydawanie bieżących zaleceń dla administratorów. Metodyka testów była analogiczna do stosowanej w zakresie testów penetracyjnych albo przeglądów konfiguracji - zależnie od usługi podlegającej analizie

   c. Okresowe analizy poziomu bezpieczeństwa (docelowo co 6 miesięcy lub częściej w odniesieniu do całości infrastruktury; w trakcie projektu testy realizowano w miarę

pojawiania się poszczególnych komponentów w środowisku testowym lub produkcyjnym - najpełniejsze testy zrealizowano pod koniec projektu). Okresowe przeglądy koncentrują się na zmodyfikowanych albo dodanych elementach infrastruktury oraz na tych elementach usług, względem których stan wiedzy w zakresie zabezpieczeń IT zmienił się od czasu poprzedniego audytu.

i. Testy penetracyjne z sieci publicznej
Test penetracyjny odzwierciedla faktyczne scenariusze realizowane przez napastników i w relatywnie krótkim czasie umożliwia wykrycie najbardziej rzeczywistych scenariuszy ataku. Test uwzględnia zarówno działanie narzędzi automatycznych, jak i manualne testy, realizowane przez doświadczonych ekspertów bezpieczeństwa, co pozwala na pożądane dopasowanie metodologii do szczegółów infrastruktury. Stosowane metodyki: autorskie, bazujące na OWASP / OSSTM. Przykładowo, dla usług Web (najbardziej eksponowanych w projekcie), wykorzystano metodykę opartą na OWASP Testing Guide (por.https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents), koncentrując się na liście podatności OWASP Top 10 - ale nie ograniczając się do niej. Wspomniana lista obejmuje:

- A1 - Injections
- A2 - Broken Authentication and Session Management
- A3 - Cross-Site-Scripting (XSS)
- A4 - Insecure Direct Object References
- A5 - Security Misconfiguration
- A6 - Sensitive Data Exposure
- A7 - Missing Function Level Data Control
- A8 - Cross-Site Request Forgery (XSRF)
- A9 - Using Components with Known Vulnerabilities
- A10 - Unvalidated Redirects and Forwards

Zadbano również o to, aby nie naruszyć dostępności usług widocznych produkcyjnie lub niezbędnych dla procesu tworzenia oprogramowania i przygotowania infrastruktury w projekcie. Testy inwazyjne uzgadniano w specjalnie dobranych terminach.

ii. Testy dostępności (symulowane ataki DoS/DDoS)
Symulowane ataki wolumetryczne i aplikacyjne pozwalają na dokładne zmierzenie wolumenu ataku, przy którym infrastruktura przestaje odpowiadać na żądania prawowitych użytkowników. Umożliwiają również identyfikację tzw. wąskich gardeł. Analiza uwzględnia symulację ataku typu DDoS na warstwę sieciową i aplikacyjną. W szczególności przeprowadzane są symulacje ataków typu: SYN flood, ICMP flood, HTTP GET flood, HTTPS GET

flood. Ataki testowe przeprowadzano kilkukrotnie m.in. w styczniu i wrześniu 2015 r.

iii. Przeglądy konfiguracji kluczowych systemów i usług
Przeglądy konfiguracji pozwalają na identyfikację zagrożeń, które są trudno albo wcale zauważalne w fazie testów penetracyjnych. W trybie z pełną wiedzą o systemach możliwe jest także wskazanie problemów bezpieczeństwa, które pojawią się dopiero w określonych warunkach w przyszłości. Ustawienia usług konfrontowane są z zaleceniami producentów i dostawców usług, niezależnych ekspertów bezpieczeństwa (np.: w zakresie konfiguracji SSL/TLS - bettercrypto.org), a także doświadczeniami audytorów.

iv. Audyty kodu
Audyty kodu są optymalnym uzupełnieniem testów penetracyjnych. Podobnie jak w przypadku testów penetracyjnych, posługiwano się narzędziami do automatycznej statycznej analizy kodu, ale nie przyjmowano wyników ich działania w sposób bezrefleksyjny. Zastosowano metodykę opartą na technikach przeglądu kodu (wg. *The art of Software Security Assessment*) *Black-box Generated Candidate Points* oraz *Code Comprehension – Analyze a Module/Analyze a Class/Analyze an Algorithm* dla wybranych partii kodu źródłowego, uznanych za krytyczne (pełna analiza kodu źródłowego jest zbyt czasochłonna, ponadto duża część kodu pochodzi z powszechnie stosowanych bibliotek i sens dokładnej jego analizy jest niewielki). Metodykę, pozwalającą w zależności od liczby wybranych krytycznych partii kodu, na osiągnięcie optymalnego kompromisu pomiędzy poświęconym na analizę czasem a poziomem dogłębności, można podzielić na następujące kroki:

1. Automatyczny przegląd kodu aplikacji z weryfikacją wyników
2. Manualny przegląd krytycznych partii kodu
    a. Budowanie listy *Candidate Points*
    b. Określenie krytycznych partii kodu
    c. Właściwa analiza manualna wybranych partii kodu

v. Podsumowanie analiz bezpieczeństwa
Z uwagi na możliwość zakwalifikowania części wykonanych prac do kilku kategorii, poniżej przedstawiono listę wykonanych większych analiz bezpieczeństwa w ujęciu chronologicznym:
Realizacja testowych ataków DDoS oraz wsparcie testów wydajności w projekcie (lipiec-sierpień 2013)
Przegląd bezpieczeństwa systemów operacyjnych i usług Repozytorium Audio-Wideo (przegląd konfiguracji, lipiec 2013)
Audyt bezpieczeństwa bloga technologicznego (blog.epodreczniki.pcss.pl) – lipiec/sierpień 2013, testy penetracyjne oraz przegląd konfiguracji usługi
Test penetracyjny usługi dev.beta.epodreczniki.pl (sierpień 2013)

Test penetracyjny usługi www.beta.epodreczniki.pl (sierpień-wrzesień 2013)
Ogólne testy bezpieczeństwa infrastruktury przed I odsłoną projektu (sierpień/wrzesień 2013)
Test penetracyjny usługi Repozytorium Treści (wrzesień 2013)
Bieżące przeglądy usług udostępnianych na zewnątrz (październik-grudzień 2013)
styczeń 2014: realizacja testowych ataków DDoS w odniesieniu do zmodyfikowanej infrastruktury projektu oraz wsparcie testów wydajności w projekcie.
Sukcesywne przeglądy konfiguracji SSL/TLS w udostępnianych usługach w związku z częstym uaktualnianiem zestawu zaleceń dotyczącego konfiguracji transmisji szyfrowanej SSL/TLS (np. grudzień 2013, styczeń, kwiecień, grudzień 2014, styczeń 2015).
Bieżące ogólne testy bezpieczeństwa infrastruktury projektu oraz udostępnianych na zewnątrz usług (np. w kwietniu 2014 przetestowano występowanie błędu OpenSSL Heartbleed, w maju 2014 poinformowano administratorów infrastruktury projektowej o umieszczeniu kilku systemów na liście N6, gromadzącej informacje o podatnych systemach dostępnych w sieci
lipiec/sierpień 2014: powtórny audyt bezpieczeństwa bloga technologicznego (blog.epodreczniki.pcss.pl)
Przegląd bezpieczeństwa serwisu uspp.pl.
Pełne testy infrastruktury (lipiec-wrzesień 2015)
Dedykowany, pogłębiony audyt bezpieczeństwa mechanizmu uwierzytelniania (wrzesień 2015)
Audyty kodu wybranych komponentów programowych (sierpień-wrzesień 2015)

d. Bezpieczeństwo informacji - kwestie związane z zarządzaniem bezpieczeństwem informacji, ochroną danych osobowych, identyfikacją oraz spełnieniem ewentualnych nowych wymagań prawnych w zakresie ochrony informacji itd. W ramach zapewnienia właściwego poziomu bezpieczeństwa informacji, oprócz opracowywania ekspertyz i zaleceń, przedstawiciele projektu brali również udział w konferencjach i szkoleniach związanych z ochroną danych osobowych, m.in.:

    i. Udział w konferencji „Umowy o świadczenie usług przetwarzania danych osobowych <<w chmurze>> - regulacja cloud computingu przez Komisję Europejską” (marzec 2014, Warszawa)

    ii. Udział w webinarze „Przygotowanie organizacji do kontroli GIODO” (luty 2014, dostęp online).

e. Bieżące wsparcie konsultacyjne w zakresie analiz bezpieczeństwa - m.in (wymieniono najważniejsze analizy).:

    i. Opracowanie i przedstawienie na spotkaniu projektu w marcu 2013 r. ramowego planu prac w zakresie cyberbezpieczeństwa

ii. Specyfikacja wymagań zewnętrznej analizy bezpieczeństwa infrastruktury projektu (maj 2013)

iii. Propozycje rozwiązań sprzętowych i programowych mogących wesprzeć ochronę infrastruktury projektu przed zagrożeniami IT (lipiec 2013)

iv. Informacje dotyczące kwestii wydajnościowych w ujęciu parametrów peeringu PCSS z zewnętrznymi operatorami telekomunikacyjnymi (lipiec 2013)

v. Informacje na temat komercyjnych certyfikatów SSL/TLS (lipiec 2013)

vi. Opracowanie procedur działania w razie ataku DDoS na infrastrukturę projektu (sierpień 2013)

vii. W lutym 2014 r. - wsparcie procesu pozyskiwania zaufanego certyfikatu Symantec Inc. dla celów podpisywania modułów kodu wytworzonych w projekcie (w tym rozwiązywanie problemów z pobraniem certyfikatu oraz wykorzystaniem rekomendowanego algorytmu generowania skrótu certyfikatu SHA-2, którego jednak nie wspierały wykorzystywane środowiska programistyczne; ostatecznie wymiana certyfikatu)

viii. W lutym i marcu 2014 - analiza raportu „Standardy bezpieczeństwa dla e-Podręczników" autorstwa Marcina Sołodki, udostępnianego przez Ośrodek Rozwoju i Edukacji. Raport przygotowano na zlecenie ORE w celu dostarczenia wytycznych i rekomendacji autorom treści e-Podręczników oraz twórcom platformy do rozpowszechniania e-Podręczników (głównie pod kątem możliwego wpływu na działania zespołów technicznych PCSS przygotowujących platformę techniczną dla e-Podręczników)

ix. W marcu 2014 r. - ekspertyza na temat zawartości dokumentu „Propozycja ogólnej specyfikacji infrastruktury sieciowo-usługowej oraz wykaz środków dydaktycznych w procesie cyfryzacji szkoły w latach 2014-2020 – Założenie Merytoryczne"

x. W czerwcu 2014 - ekspertyza dot. właściwej ochrony danych osobowych w aplikacjach mobilnych

xi. W lipcu 2014 - weryfikacja poziomu bezpieczeństwa systemu wykorzystywanego w ramach projektu przez jednego z partnerów merytorycznych.

xii. We wrześniu 2014 - konsultacje w zakresie informacji wskazującej, że w wersji demonstracyjnej systemu nie można oczekiwać docelowej wersji mechanizmów zabezpieczeń.

xiii. We wrześniu i październiku 2014 - analiza poziomu zabezpieczeń implementacji OAuth 2.0/OpenID Connect do sfederowanego uwierzytelnienia, przewidzianej do wykorzystania w projekcie

xiv. Analiza dokumentacji okołoprojektowej w kontekście bezpieczeństwa IT, m.in. analiza raportu "Standardy bezpieczeństwa e-podręczników" autorstwa M. Sołodki (luty-marzec 2014), współudział w tworzeniu opracowania "Bezpieczna szkoła cyfrowa - Zalecenia i rekomendacje dla samorządów - realizatorów projektów unijnej perspektywy budżetowej 2014-2020" (maj-czerwiec 2015)

xv. Reagowanie na zapytania bieżące zespołów projektowych, wyrażane na wewnętrznej liście mailingowej, na cotygodniowych spotkaniach zespołu projektowego ze stałym udziałem przedstawiciela Działu Bezpieczeństwa oraz w kontaktach bezpośrednich

3. Wykrywanie i reakcja na incydenty bezpieczeństwa - jako część bieżącej pracy operacyjnej
    a. Działanie zespołu do obsługi incydentów sieciowych (CERT/CSIRT)
    b. Centrum zarządzania, monitorujące działanie infrastruktury sieciowej w trybie 24/7/365
    c. Bieżąca obsługa i dostosowywanie konfiguracji zaawansowanych systemów bezpieczeństwa
    d. Przygotowanie do zabezpieczenie przed nieznanymi zagrożeniami przy pomocy systemu detekcji anomalii

Usługa uwzględnia nieustanne zmiany w obszarze zabezpieczeń IT, co można osiągnąć poprzez okresową (docelowo - w fazie utrzymania projektu - co najmniej 2 razy w roku) realizację elementu 2a, 2c oraz ciągłą realizację elementu 2b.

Z realizacji mniejszych komponentów usługi sporządzane były i wysyłane administratorom, z bardziej kompleksowych testów przygotowywano raporty, obejmujące:

1. Streszczenie dla Kierownictwa – zawierające podsumowanie oceny analizowanych obszarów infrastruktury IT oraz usług z punktu widzenia bezpieczeństwa, a także ewentualne inne wnioski natury ogólnej będące wynikiem przeprowadzonych testów.
2. Opis wykonanych testów wraz ze szczegółami zastosowanej metodologii.
3. Szczegółowe rezultaty zrealizowanych testów – w szczególności opis zidentyfikowanych luk bezpieczeństwa, na który składa się dla każdej luki:
    a. Poziom krytyczności luki zgodny z przyjętą w testach i opisaną metodyką
    b. Opis luki, w tym informacje pozwalające powtórzyć test, jaki doprowadził do jej wykrycia
    c. Sposób wykrycia luki
    d. Zagrożenia wynikające z istnienia luki
    e. Propozycje podniesienia poziomu bezpieczeństwa poprzez usunięcie luki lub załagodzenie jej skutków

4. Załączniki z poszerzonymi informacjami – odpowiednio do zawartości części raportu wymienionej w punkcie 3. bezpośrednio powyżej.

Dzięki obejmowaniu różnorodnych, nie tylko ściśle technologicznych kwestii (procedury, ochrona informacji, bezpieczeństwo sieci, systemu, usługi, danych) usługa realizuje strategię ochrony dogłębnej (ang. Defence-in-Depth), maksymalizującą poziom trudności ewentualnego udanego ataku, a zatem minimalizującą użyteczność zabezpieczanej infrastruktury dla agresora.

Usługa uwzględnia sprzężenie zwrotne - np. rezultaty szkolenia podsumowującego wyniki audytu mogą być wykorzystane przez Administratorów w dalszej codziennej pracy, a sami audytorzy wykorzystują te wyniki w celu ustalenia priorytetów dla kolejnych testów okresowych (np. koncentracja na obszarach, na których częściej były popełniane błędy.