Raport z warsztatów nt. planowanego konkursu na pilotażowe wdrożenia OSS w JST

Niniejszy raport przedstawia wnioski i postulaty podejmowane w ramach warsztatów konsultacyjnych na temat planowanego konkursu na pilotażowe wdrożenia OSS w JST. Warsztaty odbyły się 16 września 2026 r. w siedzibie Ministerstwa Cyfryzacji. Kluczowe dyskusje podejmowane były w ramach czterech podstolików tematycznych: Zakres wdrożeń, Replikowalność i interoperacyjność, Podmioty wdrażające i wsparcie dla nich oraz Bezpieczeństwo.

Dyskusja odbyła się w czterech grupach, każdy z uczestników miał okazję wypowiedzieć się w każdym temacie. Wiele uwag powtarzało się między grupami, dlatego raport przedstawia wnioski zbiorczo. Tam, gdzie uczestnicy mieli różne zdania, raport przedstawia oba stanowiska. Pojedyncze głosy zostały oznaczone jako takie.

Zakres wdrożeń

Stolik zajmował się zakresem wdrożeń, które będą mogły otrzymać finansowanie w planowanym konkursie na pilotażowe wdrożenia oprogramowania open source w jednostkach samorządu terytorialnego (JST). Punktem wyjścia były obszary funkcjonalne zapowiedziane przez Ministerstwo Cyfryzacji: chmura, edytory tekstu, arkusze kalkulacyjne, programy do prezentacji, programy pocztowe, programy do wideokonferencji, programy do zarządzania kalendarzem oraz komunikatory.

Najważniejsze rekomendacje

1. Katalog obszarów funkcjonalnych powinien zostać uzupełniony o zarządzanie tożsamością i dostępem (SSO). Należy też rozważyć włączenie innych obszarów wskazanych przez uczestników, w szczególności zarządzania projektami, danych przestrzennych oraz komunikacji z mieszkańcami.
2. Katalog kosztów kwalifikowanych powinien obejmować kopie zapasowe, monitoring i bezpieczeństwo, warstwę wirtualizacji, migrację danych oraz integracje z systemami dziedzinowymi, w tym z EZD.
3. Konkurs nie powinien finansować budowy serwerowni od podstaw. Powinien natomiast dopuszczać wydatki na rozbudowę istniejących serwerowni w zakresie potrzebnym do wdrożenia.
4. Konkurs nie powinien narzucać JST modelu infrastruktury. Wnioskodawca powinien uzasadnić wybrany model, także pod względem kosztów.
5. Decyzja o wymaganiu lub premiowaniu pakietów rozwiązań powinna wynikać z jasno określonego celu pilotażu i zostać czytelnie zakomunikowana JST.
6. Katalog rozwiązań powinien być otwarty i oparty na obszarach funkcjonalnych oraz wymaganiach, a nie na nazwach produktów. Lista przykładowych rozwiązań może pomóc JST w przygotowaniu wniosku.

Obszary funkcjonalne i katalog kosztów

Uczestnicy uznali, że zapowiedziane obszary obejmują podstawowe narzędzia pracy biurowej i komunikacji, ale nie odpowiadają na wszystkie potrzeby JST. Wskazali następujące obszary, w których wdrożenie open source może być dla JST szczególnie korzystne:

- Zarządzanie tożsamością i dostępem.
- Zarządzanie projektami i zadaniami.
- Dane przestrzenne i zarządzanie danymi.
- Ochrona ludności i zarządzanie kryzysowe.
- Komunikacja z mieszkańcami.
- Współdzielenie zasobów i repozytoria kodu.
- Systemy dziedzinowe.
- Narzędzia administratora.

Pojedyncze głosy dotyczyły narzędzi wykorzystujących sztuczną inteligencję, w tym transkrypcji i systemów wyszukiwania w dokumentach, przy czym przetwarzanie danych powinno odbywać się lokalnie. Jeden z uczestników uznał, że wideokonferencje powinny być zapewnione centralnie przez państwo, a nie wdrażane przez poszczególne JST.

Uczestnicy wskazali elementy, które nie są osobnymi obszarami funkcjonalnymi, ale bez których wdrożenie nie będzie działać stabilnie i bezpiecznie. Ich zdaniem powinny one znaleźć się w katalogu kosztów kwalifikowanych:

- kopie zapasowe wraz z przetestowaną procedurą odtwarzania, które przy infrastrukturze własnej są podstawowym zabezpieczeniem ciągłości działania;
- monitoring i bezpieczeństwo operacyjne, czyli monitoring dostępności, zbieranie logów, wykrywanie incydentów i skanowanie podatności;
- warstwa wirtualizacji i hostingu dla wdrożeń we własnej infrastrukturze;
- migracja danych i dokumentów, w tym szablonów pism, uchwał i zarządzeń, arkuszy z formułami i makrami, archiwów poczty oraz kalendarzy;
- integracje z systemami dziedzinowymi, w szczególności z EZD RP, systemami finansowo-księgowymi i podpisem kwalifikowanym.

Uczestnicy wskazali także koszty, które często są pomijane przy planowaniu: szkolenia przypominające dla nowych pracowników, wsparcie użytkowników i dostęp do inżyniera po wdrożeniu, nadzór merytoryczny nad wdrożeniem oraz dokumentację wykonawczą przygotowaną według przyjętych standardów. Zwrócono też uwagę na koszty dostosowania oprogramowania do nowych regulacji, na przykład wymagań kryptograficznych, oraz na pytanie, kto będzie za to odpowiadać: JST czy wykonawca.

Uczestnicy zarekomendowali podjęcie następujących działań:

- Uzupełnić katalog obszarów funkcjonalnych o zarządzanie tożsamością i dostępem.
- Rozważyć włączenie zarządzania projektami, danych przestrzennych, komunikacji z mieszkańcami i ochrony ludności jako obszarów dodatkowych.
- Wpisać do katalogu kosztów kwalifikowanych kopie zapasowe, monitoring i bezpieczeństwo, wirtualizację.
- Dopuścić koszty szkoleń, wsparcia użytkowników i utrzymania w okresie trwałości projektu oraz jasno określić podział odpowiedzialności za aktualizacje między JST a wykonawcą.

Sprzęt i infrastruktura

Uczestnicy zgodzili się, że JST, które nie mają serwerowni, nie zbudują jej wyłącznie ze środków z konkursu, ponieważ planowana wysokość grantów jest na to zbyt niska. JST, które mają już serwerownię, mogą natomiast zdecydować się na jej rozbudowę na potrzeby wdrożenia.

Jeden z uczestników zaproponował finansowanie wymiany stacji roboczych na sprzęt z systemem Linux. Pojawiła się też propozycja przygotowania listy komponentów sprzętowych dobrze wspieranych przez popularne dystrybucje systemu Linux.

On-premise, chmura i model hybrydowy
W tej sprawie uczestnicy mieli różne zdania.

Część uczestników przekonywała, że konkurs nie powinien narzucać JST rozwiązań chmurowych. Argumentowali, że opłacalność chmury i własnych serwerów zależy od wielkości JST, a infrastruktura on-premise może być skalowalna. Zwracali też uwagę na suwerenność danych i na to, kto jest dostawcą usług chmurowych. Inni uczestnicy uważali, że finansowanie infrastruktury on-premise pochłonie dużą część budżetu i nie powinno być dopuszczone.

Wiele głosów wskazywało na model hybrydowy, który łączy własną serwerownię z chmurą i pozwala w przyszłości przenieść usługi z infrastruktury własnej do chmury. Proponowano, aby wybór infrastruktury on-premise był oceniany pod kątem efektywności ekonomicznej. Pojawiły się też postulaty dotyczące rozwiązań wspólnych: chmury krajowej z wydzielonymi zasobami dla gmin, narzędzi udostępnianych centralnie przez państwo oraz współdzielenia infrastruktury przez kilka gmin.

Uczestnicy zarekomendowali:

- Nie finansować budowy serwerowni od podstaw.
- Dopuścić wydatki na rozbudowę istniejących serwerowni w zakresie niezbędnym do wdrożenia, i wskazać w katalogu kosztów kategorie takich wydatków (serwery, pamięć masowa, wirtualizacja, centrum zapasowe, klimatyzacja, energia).
- Pozostawić JST wybór modelu infrastruktury i wymagać we wniosku uzasadnienia tego wyboru, w tym porównania kosztów.

Pakietowość wdrożeń
Najwięcej dyskusji wywołało pytanie, czy konkurs powinien wymagać lub premiować wdrożenie kilku obszarów funkcjonalnych jednocześnie. Uczestnicy przedstawili argumenty za oboma podejściami.

Zwolennicy pakietowości argumentowali, że zmiana tylko jednego obszaru niesie wysokie ryzyko problemów z interoperacyjnością i zmniejsza szansę na sukces wdrożenia. Wskazywali też, że oszczędności pojawiają się dopiero wtedy, gdy JST zastąpi cały dotychczasowy pakiet biurowy. Ich zdaniem potrzeby JST są w dużej mierze takie same, więc w dłuższej perspektywie powinien powstać jeden pakiet, który je pokryje.

Przeciwnicy wymogu pakietowości argumentowali, że wymaganie wielu obszarów zniechęci JST do udziału w konkursie i że wiele JST nie podejmie się wymiany wszystkich narzędzi naraz. Proponowali, aby JST mogła wybrać tylko te obszary, których potrzebuje, nawet jeden. Pakietowość mogłaby być premiowana, ale nie wymagana. Proponowano także premiowanie wdrożeń etapowych, w których kolejne narzędzia są wprowadzane stopniowo.

Osobną kwestią jest sytuacja JST, które już wdrożyły część rozwiązań open source. Jeżeli konkurs będzie premiował pakiety, takim JST będzie trudniej uzyskać finansowanie. Z jednej strony pozwala to skierować środki tam, gdzie open source jeszcze nie ma. Z drugiej strony JST z doświadczeniem mogą być najbardziej zaangażowane i najskuteczniej wdrożyć kolejne rozwiązania. Uczestnicy zwrócili uwagę, że premiowanie JST zaawansowanych może sprawić, że pilotaż nie będzie reprezentatywny. Odpowiedź na to pytanie zależy więc od tego, jaki cel ma osiągnąć pilotaż.

Część uczestników patrzyła na pakietowość w perspektywie długoterminowej. Pilotaż traktowali jako początek drogi do modelu, w którym administracja sama rozwija rozwiązania na potrzeby administracji, tak jak we Francji. Model ten różni się od niemieckiego, w którym administracja przede wszystkim łączy w pakiety rozwiązania dostępne na rynku.

Uczestnicy zarekomendowali:

- Przed rozstrzygnięciem kwestii pakietowości określić, czy celem pilotażu jest przetestowanie możliwie pełnej zmiany środowiska pracy, czy sprawdzenie wdrożeń w możliwie różnych JST.
- Jeżeli pakietowość będzie premiowana, jasno zakomunikować w dokumentacji konkursowej, że nie jest warunkiem udziału.
- Określić, czy obszary wdrożone przez JST przed konkursem są brane pod uwagę przy ocenie pakietowości.
- Rozważyć dopuszczenie wdrożeń etapowych.

Pułapki w definiowaniu obszarów funkcjonalnych

Uczestnicy wskazali kilka problemów, które mogą pojawić się przy opisywaniu obszarów funkcjonalnych w dokumentacji konkursowej.

- **Klient a serwer.** Określenie „program pocztowy” może oznaczać zarówno klienta poczty na stacji roboczej, jak i serwer pocztowy. To samo dotyczy kalendarza. Definicje powinny te przypadki rozróżniać.
- **Pakiety zintegrowane.** Rozwiązania takie jak Nextcloud Hub, openDesk czy Grommunio obejmują jednocześnie kilka obszarów. Definicje nie mogą wymuszać sztucznego rozbijania takich pakietów ani liczenia ich kilka razy.
- **Niekompletne ekosystemy.** Ekosystem Nextcloud nie zawiera serwera poczty, więc pocztą trzeba zarządzać osobno i nie ma jednego panelu administracyjnego. Uczestnicy podkreślali, że poczta jest dla JST bardzo ważną funkcją.
- **Open core.** Część rozwiązań łączy otwarty rdzeń z płatnymi lub zamkniętymi modułami. Należy sprawdzać, czy funkcje potrzebne do wdrożenia są dostępne w części otwartej.
- **Zbyt wąskie definicje.** Zbyt wąska definicja obszaru może ograniczać wybór rozwiązań. Uczestnicy zwracali uwagę na ostrożne używanie pojęć takich jak „pakiet” czy „procesor tekstu” oraz na określenie, jakie formaty plików rozwiązanie ma obsługiwać.
- **Integracja z istniejącymi systemami.** Nowe rozwiązania muszą współpracować z EZD RP oraz z lokalnymi systemami JST. Uczestnicy podkreślali, że nie da się stworzyć jednego systemu dla wszystkich, ale różne rozwiązania muszą być ze sobą kompatybilne. Wskazywali też na trwałość standardów oraz bezpieczeństwo dokumentów, w tym podpisu elektronicznego.

Uczestnicy zarekomendowali:

- W definicji każdego obszaru określić, czy obejmuje oprogramowanie klienckie, serwerowe czy oba rodzaje.
- Dopuścić rozwiązania zintegrowane, które obejmują kilka obszarów, i jasno opisać, jak będą one liczone.
- Wymagać, aby funkcje niezbędne do realizacji wdrożenia były dostępne na licencji open source.
- Opisywać obszary przez funkcje i obsługiwane formaty plików, a nie przez typy produktów.

Przykładowe rozwiązania i otwarty katalog

Uczestnicy poparli koncepcję otwartego katalogu, który określa obszary funkcjonalne, a nie konkretne produkty. Podkreślali, że konkurs nie powinien koncentrować się na jednym oprogramowaniu. Wdrożenia będą realizować wykonawcy, którzy powinni móc wybierać spośród różnych rozwiązań i muszą je dobrze znać. Jeden z uczestników zaproponował, aby Ministerstwo przygotowało listę dopuszczonych rozwiązań.

Poniższa tabela zawiera przykłady rozwiązań wskazanych podczas warsztatów i po nich.

Obszar	Przykładowe rozwiązania
Pliki i chmura	Nextcloud, OpenCloud, Seafile
Edytory, arkusze, prezentacje	LibreOffice (stacja robocza), Collabora Online (edycja w przeglądarce)
Poczta – serwer	Postfix z Dovecot i Roundcube lub SOGo, Grommunio, Stalwart, Open-Xchange (po weryfikacji ryzyka open core)
Poczta – klient	Thunderbird
Kalendarz i kontakty	SOGo, Grommunio, Nextcloud (CalDAV i CardDAV)
Wideokonferencje	Jitsi Meet, BigBlueButton, Nextcloud Talk
Komunikator	Matrix (Element, Synapse), Nextcloud Talk
Tożsamość i SSO	FreeIPA, Keycloak, Samba AD (środowiska mieszane)
Kopie zapasowe	Proxmox Backup Server, Bareos
Monitoring i bezpieczeństwo	Wazuh, Zabbix, Suricata, Graylog, Greenbone (OpenVAS)
Dane przestrzenne i bazy danych	QGIS, PostgreSQL
Repozytoria kodu	GitLab

Replikowalność i interoperacyjność

Stolik zajmował się poszukiwaniem rozwiązań służących maksymalizacji stopnia replikowalności wdrożeń – zarówno w innych JST, jak i na poziomie centralnym. Podejmowano również wątki związane z interoperacyjnością wdrażanych rozwiązań.

Replikowalność oznacza możliwość odtworzenia rozwiązania, a nie kopiowanie wdrożenia 1:1
Uczestnicy wyraźnie rozdzielali możliwość ponownego wykorzystania aplikacji od możliwości odtworzenia infrastruktury, w której rozwiązanie działa. W przypadku aplikacji kluczowa jest możliwość jej uruchomienia w innym środowisku na podstawie odpowiedniej dokumentacji. W przypadku infrastruktury trzeba uwzględnić lokalne uwarunkowania jednostki, m.in. konfigurację środowiska, domenę, dostęp sieciowy, zabezpieczenia, zasoby sprzętowe i inne zależności techniczne.
Zagrożenie: udostępnienie samego kodu, kontenera lub konfiguracji aplikacji może stworzyć pozorną replikowalność. Rozwiązanie przeniesione do innej JST może nie działać, jeżeli dokumentacja nie opisuje elementów środowiska niezbędnych do jego uruchomienia.
Potrzeba: po każdym wdrożeniu powinien pozostać zestaw informacji umożliwiający jego odtworzenie w innej jednostce z uwzględnieniem jej własnego środowiska.

Po wdrożeniu powinien pozostać kompletny pakiet artefaktów
Samo udostępnienie kodu źródłowego uznano za niewystarczające. Materiał po wdrożeniu powinien pokazywać nie tylko efekt techniczny, ale również sposób dojścia do niego i doświadczenia zdobyte w trakcie realizacji.

- kod, skrypty i konfiguracje, w tym – tam, gdzie jest to zasadne – Infrastructure as Code;
- opis wymagań wstępnych i infrastruktury niezbędnej do uruchomienia rozwiązania;
- opis instalacji, deploymentu i konfiguracji, wykorzystywanych dodatków i plug-inów oraz standardów działania;
- informacje o integracjach, formatach danych i zastosowanych standardach;
- materiały dotyczące migracji, eksportu, importu i przenoszenia danych;
- opis problemów napotkanych podczas wdrożenia, sposobów ich rozwiązania oraz kwestii, których nie udało się rozwiązać;
- wyniki analiz lub audytów przedwdrożeniowych i powdrożeniowych;
- case study, dobre praktyki i lessons learned przydatne przy kolejnych wdrożeniach.

Potrzeba: dokumentacja powinna być na tyle kompletna, aby kolejna JST nie musiała od początku odtwarzać wiedzy o wdrożeniu ani szukać kluczowych informacji poza materiałami pozostawionymi po pilotażu.

Audyt przed wdrożeniem i analiza po wdrożeniu zwiększają wartość pilotażu
Uczestnicy wskazywali na wartość analizy stanu wyjściowego: jakie rozwiązania i zasoby jednostka posiadała, co mogło zostać zastąpione przez OSS, czego nie udało się zastąpić i z jakich przyczyn. Po zakończeniu wdrożenia warto zestawić pierwotne założenia z rzeczywistym przebiegiem i efektami, w tym kosztami, korzyściami, dodatkowymi nakładami i problemami. Pojawiła się również propozycja porównywania podobnych wdrożeń realizowanych w różnych JST. Taki benchmark mógłby pokazać, jak różne podejścia sprawdzają się w praktyce i dostarczyć danych przydatnych przy ewentualnym skalowaniu programu.
Zagrożenie: bez udokumentowania stanu wyjściowego i rzeczywistych efektów pilotaż może pozostawić jedynie informację, że rozwiązanie zostało wdrożone, bez wiedzy potrzebnej do oceny jego przydatności w innych JST.

Materiały powinny być użyteczne również dla użytkowników, nie tylko dla informatyków
Oprócz dokumentacji technicznej wskazywano instrukcje dla użytkowników, krótkie materiały wideo, filmy instruktażowe, e-learning oraz transkrypcje materiałów audiowizualnych. Transkrypcje mogą ułatwiać wyszukiwanie treści i ich dalsze przetwarzanie, w tym z wykorzystaniem narzędzi AI.

W dyskusji powtarzał się problem oporu użytkowników przed zmianą. Barierą może być nie tylko technologia, ale również przyzwyczajenie do dotychczasowych narzędzi oraz brak szybkiej, praktycznej pomocy.
Potrzeba: obok materiałów technicznych warto pozostawić materiały wspierające użytkownika końcowego i osoby odpowiedzialne lokalnie za wdrożenie.

Model infrastruktury wpływa na możliwość późniejszego powielenia wdrożenia
Dużo miejsca poświęcono dyskusji nad modelami on-premise, chmurowym i hybrydowym. Uczestnicy wskazywali, że jednostka powinna mieć możliwość doboru modelu adekwatnego do własnych zasobów i potrzeb. Część uczestników oceniała, że środowiska chmurowe mogą ułatwiać standaryzację i powielanie wdrożeń, natomiast on-premise wymaga uwzględnienia większej liczby lokalnych zmiennych. Nie wypracowano jednak postulatu narzucenia jednego modelu.
Pojawił się również model chmury prywatnej oraz rozwiązań hybrydowych. Wskazywano na możliwość wykorzystania gotowych i zautomatyzowanych procesów deploymentu oraz rozwiązań pakietowych, które ograniczają liczbę decyzji infrastrukturalnych podejmowanych indywidualnie przez JST.
Potrzeba: dokumentacja wdrożenia powinna jednoznacznie wskazywać model środowiska oraz wymagania potrzebne do odtworzenia rozwiązania w innym wariancie infrastrukturalnym, jeżeli jest to możliwe.

Rezultaty powinny być wymieniane w bezpiecznym środowisku współpracy
Replikowalność była łączona z budowaniem sieci osób odpowiedzialnych za wdrożenia w JST. W szczególności przy prostszych wdrożeniach lokalni administratorzy lub liderzy mogliby wymieniać doświadczenia, zgłaszać problemy i korzystać z wiedzy innych jednostek. Pojawił się również wątek współdzielenia zasobów lub infrastruktury przez mniejsze JST.
Jednocześnie zwracano uwagę, że nie wszystkie szczegóły techniczne powinny być publikowane całkowicie otwarcie. Zakres udostępniania materiałów powinien uwzględniać bezpieczeństwo i charakter informacji. Jako jeden z możliwych kanałów wymiany wiedzy wskazano istniejące środowisko ISAC, dostępne dla zidentyfikowanych użytkowników.

Rozwój brakujących funkcjonalności OSS wymaga odrębnych zasad
W warsztacie wspomniano przypadek, w którym istniejące rozwiązanie OSS odpowiada potrzebom JST, ale wymaga dopisania konkretnej funkcjonalności. Dyskutowano możliwość finansowania takich prac programistycznych w ramach wdrożenia oraz zapewnienia, aby rezultat mógł być później wykorzystany również przez inne jednostki.
Do dalszego rozstrzygnięcia: kwalifikowalność prac programistycznych oraz zasady licencjonowania i zapisy umowne dotyczące kodu wytworzonego w ramach finansowanego wdrożenia.

Punktem wyjścia powinny być procesy i przepływy danych
Uczestnicy wskazywali, że przed określeniem wymagań technicznych trzeba ustalić, jakie informacje są wykorzystywane w danym procesie, skąd pochodzą, pomiędzy jakimi systemami są przekazywane i w jakim celu. Procesy realizowane przez JST mogą wynikać z tych samych przepisów, ale ich organizacja i obsługa techniczna różnią się pomiędzy jednostkami.
Zagrożenie: formułowanie wymagań interoperacyjności wyłącznie na poziomie narzędzi może prowadzić do wdrożenia systemów, które technicznie działają, ale nie rozwiązują rzeczywistych problemów wymiany informacji.

Standard wymiany informacji jest ważniejszy niż wskazanie konkretnego narzędzia
Jednym z najmocniejszych wniosków było oparcie interoperacyjności na standardach, protokołach i sposobie opisu danych, a nie na narzucaniu konkretnego produktu. Różne

rozwiązania powinny móc wymieniać informacje niezależnie od tego, czy są oparte na OSS, czy na oprogramowaniu własnościowym.
Wskazywano, że część braków standaryzacyjnych ma charakter systemowy i nie może zostać rozwiązana samym konkursem grantowym. Wymagania projektu powinny więc koncentrować się na tym, co JST i wykonawca mogą zapewnić w ramach konkretnego wdrożenia, bez przenoszenia na grantobiorców obowiązku tworzenia standardów dla całej administracji.
Potrzeba: w dokumentacji konkursowej należy oddzielić wymagania możliwe do postawienia pojedynczemu wdrożeniu od problemów wymagających rozwiązań na poziomie centralnym.

KRI, IIP i INSPIRE mogą stanowić punkt odniesienia
W dyskusji odwoływano się do Krajowych Ram Interoperacyjności, Infrastruktury Informacji Przestrzennej (IIP) oraz INSPIRE. Geodezję wskazywano jako przykład obszaru, w którym charakter danych wymusił praktyczne zasady ich współdzielenia pomiędzy niezależnymi systemami. Uczestnicy proponowali wykorzystanie tych doświadczeń jako punktu odniesienia przy formułowaniu zasad dla innych obszarów.
Pojawiła się również propozycja opracowania na poziomie centralnym bardziej przekrojowych wytycznych dotyczących interoperacyjności. Jest to postulat wykraczający poza bezpośredni zakres pilotażu, ale warto go zachować jako wniosek systemowy z konsultacji.

Otwarte formaty muszą być rzeczywiście użyteczne w wymianie danych
Praktycznym problemem jest wymiana dokumentów i danych pomiędzy rozwiązaniami otwartymi i własnościowymi. Uczestnicy przywoływali sytuacje, w których dokument po otwarciu w innym rozwiązaniu zmienia wygląd lub strukturę, mimo deklarowanej zgodności formatu.
Podkreślano znaczenie importu, eksportu i konwersji danych oraz zachowania dostępu do zasobów wytworzonych przed wdrożeniem OSS. Zwracano również uwagę, że sama deklaracja, iż format jest otwarty, może być niewystarczająca. Jako pomocniczą wskazówkę wskazywano publiczną i kompletną specyfikację oraz – tam, gdzie ma to zastosowanie – oparcie na uznanym standardzie lub normie.
Zagrożenie: wdrożenie rozwiązania OSS bez sprawdzenia rzeczywistej wymienialności danych może doprowadzić do problemów z dotychczasowymi plikami, sprawozdawczością lub współpracą z innymi instytucjami.

Otwarte i udokumentowane API są jednym z kluczowych mechanizmów integracji
W odniesieniu do rozwiązań współpracujących z innymi systemami wielokrotnie wskazywano na znaczenie otwartych i udokumentowanych API. Argumentowano, że nie da się przewidzieć wszystkich systemów, z którymi rozwiązanie będzie musiało współpracować za kilka lat, dlatego możliwość przyszłej integracji powinna być zachowana.
Jednocześnie API nie powinno być traktowane jako jedyny sposób zapewnienia interoperacyjności. Zakres wymagań powinien zależeć od rodzaju rozwiązania i rzeczywistych przepływów danych.

Interoperacyjność dotyczy również znaczenia, pochodzenia i wiarygodności danych
W dyskusji mocno wybrzmiał problem semantyki danych. Sama możliwość przesłania wartości pomiędzy systemami nie wystarcza, jeżeli odbiorca nie wie, co wartość oznacza, w jakiej jednostce została wyrażona, z jakiego źródła pochodzi i czy źródło ma charakter referencyjny. Zwracano uwagę na integralność i autentyczność informacji oraz możliwość ich weryfikacji po wielu latach. System lub platforma może zostać zastąpiona, podczas gdy dokument administracyjny lub dane mogą nadal wymagać odczytania i potwierdzenia ich pochodzenia.
Potrzeba: przy określaniu interoperacyjności należy uwzględnić nie tylko kanał techniczny wymiany danych, ale również ich strukturę, metadane, źródło, znaczenie i możliwość późniejszej weryfikacji.

Trzeba uwzględnić istniejące i starsze zasoby
Nowe rozwiązania będą korzystały z danych wytworzonych w istniejących systemach. Uczestnicy wskazywali potrzebę konwersji i mapowania danych ze źródeł używanych obecnie przez JST oraz zachowania dostępu do starszych formatów po zmianie oprogramowania.
W tym kontekście pojawiał się również temat kompatybilności wstecznej. Z dyskusji wynikało jednak, że wymagania nie powinny być narzucane identycznie wszystkim JST, ponieważ jednostki dysponują różnymi systemami i zasobami.

OSS musi współistnieć z rozwiązaniami już funkcjonującymi w administracji
Uczestnicy wskazywali, że pełne zastąpienie rozwiązań własnościowych nie zawsze będzie możliwe lub uzasadnione. W części procesów JST może nadal potrzebować określonych funkcji lub formatów wykorzystywanych przez inne instytucje. Pilotaż powinien więc pozwolić również na modele mieszane i stopniowe wprowadzanie OSS bez zakłócania ciągłości pracy.
Dotyczy to również współpracy z systemami administracji centralnej. W warsztacie wskazano m.in. na znaczenie interoperacyjności z systemami elektronicznego zarządzania dokumentacją, w szczególności EZD RP, jeżeli wdrażane rozwiązanie uczestniczy w procesach obsługiwanych elektronicznie.

Należy uważać na rozwiązania typu open core
W dyskusji przypomniano ryzyko związane z wdrażaniem modeli, w których podstawowa część rozwiązania jest otwarta, natomiast funkcje potrzebne w rzeczywistym środowisku organizacji – np. część integracji, API, SSO lub dodatkowe moduły – pozostają zamknięte albo odpłatne.
Zagrożenie: formalnie otwarte rozwiązanie może w praktyce odtworzyć zależność od jednego dostawcy, jeżeli kluczowe funkcje integracyjne lub administracyjne nie są dostępne w części OSS.
Potrzeba: przy ocenie rozwiązania należy badać nie tylko licencję jego podstawowego komponentu, ale również otwartość elementów niezbędnych do realizacji zakresu finansowanego wdrożenia.

Podmioty wdrażające i wsparcie dla nich

Należy określić jasne zasady, założenia konkursu i ramy realizacji projektów.

Jak wskazywali uczestnicy warsztatów, dokumentacja powinna zawierać w szczególności:

- katalog kosztów kwalifikowanych,
- minimalne wymagania techniczne,
- wymagania dotyczące bezpieczeństwa,
- zasady finansowania,
- zasady współpracy z jednostkami podległymi,
- zasady zamówień,
- wzory OPZ,
- przykładowe zapisy do dokumentacji zakupowej,
- wzory dokumentów,
- wymagania dotyczące dokumentacji końcowej,
- zasady utrzymania rezultatów,

- katalog potencjalnych dostawców/wykonawców,

Zagrożenia: Uczestnicy warsztatów wyrażali obawę, że zbyt luźne, mało precyzyjne zasady spowodują, że trudno będzie opracować OPZ i tym samym wybrać usługodawcę. JST obawiają się, że KIO zakwestionuje OPZ i urzędy będą ponosiły konsekwencje finansowe. Zdaniem części uczestników, nawet jeśli uda się sprecyzować potrzeby, opracować szczegółowy plan działań w ramach pilotażu, to problematyczne będą także same zamówienia. Ryzyko może wiązać się z przedłużeniem procedur wyboru usługodawców, brakiem możliwości realizacji projektu i zwrot środków. Jak stwierdzono, taki scenariusz może uprawdopodobniać roczny termin realizacji pilotaży.

Potrzeba: określenie zapisów konkursu w taki sposób, aby JST mogły umieścić zapisy (nawet przekopiować niektóre wątki) w OPZ, albo opracować dodatkowo poradnik, wskazówki jak właściwie opisać przedmiot zamówienia. Te rekomendacje były powszechne i wyszły od przedstawicieli JST (kierownika IT, kierownika jednostki oraz skarbnika) i NGO.

Celem powinno być zainstalowanie OSS w każdej jednostce i stworzenie spójnego środowiska cyfrowego całej JST, a nie ograniczanie się do pojedynczego podmiotu.

Takie podejście jest istotne również dlatego, że rozwiązania cyfrowe działają na styku wielu jednostek np. poczta, kalendarze, komunikacja, obieg dokumentów, współdzielenie plików czy wideokonferencje mogą wymagać wspólnej architektury. Możliwe jest testowanie wdrożenia OSS na pojedynczych jednostkach lub tylko centralnie, ale to jest zależne od tego, jakie oprogramowanie wdrażamy i komu ono ma służyć. Testowanie w wybranych jednostkach traktuje temat wybiórczo i wymaga w przyszłości rozszerzenia i włączenia pozostałych jednostek i wypracowania rozwiązań związanych ze współpracą zewnętrzną.

Zagrożenia: Zdaniem części uczestników, wdrożenie i testowanie OSS w pojedynczych jednostkach JST nie pozwoli na zweryfikowanie rzeczywistych problemów związanych z integracją i codzienną współpracą pomiędzy różnymi jednostkami JST. W konsekwencji wnioski z takiego wdrożenia mogą mieć ograniczoną przydatność przy późniejszym rozszerzaniu rozwiązania na pozostałe podmioty.

Potrzeba: stworzenie spójnego, jednolitego i bezpiecznego środowiska cyfrowego obejmującego wszystkie jednostki organizacyjne. Dotyczy to w szczególności poczty elektronicznej, kalendarzy, komunikacji wewnętrznej, elektronicznego obiegu dokumentów. Spójne środowisko pozwoli ograniczyć rozproszenie kompetencji i zasobów oraz ułatwi zapewnienie jednolitego poziomu bezpieczeństwa we wszystkich jednostkach. Ważną potrzebą jest uzyskanie rzeczywistych doświadczeń z funkcjonowania OSS w warunkach obejmujących różne jednostki, zakresy zadań, użytkowników i poziomy kompetencji cyfrowych. Z tego względu wdrożenie powinno być planowane jako rozwiązanie obejmujące całą JST, a nie jako izolowany pilotaż w jednej, wybranej jednostce.

Lepsze efekty przyniesie JST realizacja projektu samodzielnie, bez partnerstw.

Zdaniem niektórych przedstawicieli JST realizacja projektu polegająca na wdrażaniu OSS w partnerstwie z innymi podmiotami np. gminami jest obarczone dużym ryzykiem. Obawy dotyczą braku porozumienia, wycofania się z partnerstwa, różnicy zdań w podejmowaniu kluczowych decyzji.

Zagrożenia: JST mają różne potrzeby, oczekiwania i priorytety. Chcą mieć wybór rozwiązań, być niezależne w decyzjach i nie brać odpowiedzialności za inne podmioty. Partnerstwo wydłuża czas realizacji i podejmowania decyzji co wpływa na harmonogram, zakres realizacji oraz możliwość osiągnięcia zakładanych rezultatów. Różnice w poziomie dojrzałości cyfrowej, potrzebach, posiadanej infrastrukturze i sposobie organizacji pracy mogą utrudniać wypracowanie wspólnego modelu wdrożenia.

Potrzeba: JST chcą samodzielne wdrażać OSS, być niezależnym w podejmowaniu decyzji, realizować projekt wyłącznie w odniesieniu do własnych potrzeb i zasobów.

Gradacja JST z uwzględnieniem standaryzacji.

Jak podkreślali uczestnicy warsztatów, pilotaż i mechanizm wsparcia powinny uwzględniać różne poziomy administracji: gminy, miasta na prawach powiatu, powiaty, województwa. Potrzeby i możliwości wdrożeniowe każdej z tych grup są bowiem inne. Proponowano rozważanie ścieżek lub modułów konkursowych dostosowanych do skali JST, zamiast jednego modelu dla wszystkich.

Zagrożenia: Zdaniem niektórych uczestników, małe gminy nie widzą potrzeby wdrażania rozwiązań OSS, bo mają małe zasoby, którymi doskonale zarządzają wykorzystując obecny system. Małe gminy nie posiadają zasobów kadrowych i możliwości wzajemnego wsparcia pracowników we wdrażaniu OSS. Nowy system może spowodować opór i wracanie do poprzednich rozwiązań. Z rozmów wynikało, że pracownicy zawsze z niechęcią przyjmują nowe rozwiązania. Potrzebują szkoleń, doradztwa, wzajemnego wsparcia i fachowej pomocy szczególnie na początku. Najbardziej obawiają się braku pomocy i czasu oczekiwania na reakcję ze strony „wsparcia zewnętrznego". Niepewność, długi czas oczekiwania i zaopiekowania ich problemem powoduje, że szybko zniechęcają się i deklarują powrót do starych, sprawdzonych systemów.

Duże gminy, powiaty i województwa ponoszą duże koszty utrzymania i zmiana będzie zasadna wyłącznie jak będzie się to opłacało finansowo. Zdaniem niektórych uczestników, JST nie zawsze pewności, czy wdrożenie OSS się przyjmie i na ile będzie pewnym systemem, więc mogą obawiać się inwestowania w nie sił i środków. Tym samym postulują zapewnienie bardziej długofalowego wsparcia.

Potrzeba: Centralne repozytorium wiedzy i dokumentacji, z którego w prosty sposób będą mogli skorzystać, przenieść 1/1 rozwiązania najlepiej dedykowane dla różnych wielkości JST gmina, powiat, województwo. Repozytorium powinno zawierać m.in.: instrukcje instalacji, instrukcje administracyjne, instrukcje dla użytkowników, dokumentację techniczną, kody źródłowe, procedury bezpieczeństwa, przykładowe konfiguracje.

Zasada powinna być prosta: JST otrzymuje komplet informacji niezbędnych do samodzielnego utrzymania rozwiązania, bez konieczności poszukiwania kluczowej wiedzy poza projektem. Celem powinno być zapewnienie, że różne rozwiązania mogą ze sobą współpracować.

Konieczny jest audyt potrzeb przed wdrożeniem

Podczas warsztatów postulowano, by przed wyborem rozwiązania JST miały możliwość przeprowadzenia audytu procesów, systemów, danych i potrzeb.
Pozwoli to ustalić:

- co rzeczywiście należy zmienić,
- jakie są możliwości ich integracji systemów,

- gdzie występują największe koszty,
- jakie procesy można usprawnić,
- jakie kompetencje należy rozwinąć,
- gdzie OSS rzeczywiście przyniesie wartość.

Zagrożenia: brak możliwości realnego zaplanowania działań przez JST związanych z wdrożeniem OSS nawet przy założeniu, że doskonale znają swoje zasoby i potrzeby. Brak wiedzy specjalistycznej. Konieczne jest współdziałanie JST z usługodawcą/audytorem i dostawania do potrzeb, budżetu mając na uwadze konsekwencje osobowe i finansowe w przyszłości.

Potrzeba: JST zgłaszały, że samodzielnie nie są w stanie przeprowadzić rozeznania i podjąć wiążących decyzji w zakresie wdrażania OSS w swojej jednostce. Potrzebują wsparcia profesjonalnych osób/podmiotów z zewnątrz. Decyzję o przystąpieniu podejmą po przeliczeniu kosztów. To będzie czynnik decydujący szczególnie dla dużych i średnich JST, bo ponoszą duże wydatki na utrzymanie rozwiązań własnościowych. Z tej perspektywy im krótszy czas trwałości projektu tym lepiej.

Kompetencje w zakresie zarządzania zmianą i zwiększania świadomości pracowników we wdrażaniu OSS

We wszystkich grupach mocno powtarzał się wątek czynnika ludzkiego, kluczowego we wprowadzaniu zmian. Jak podkreślano, samo wdrożenie technologii nie gwarantuje jej wykorzystania. Potrzebne są: szkolenia, doradztwo, materiały wdrożeniowe, wsparcie użytkowników, lokalny koordynator/„ambasador zmiany", wsparcie w zarządzaniu zmianą, komunikacja wewnętrzna.

Zagrożenia: pracownik bez wiedzy, świadomości i kompetencji szybko zniechęci się, wycofa ze zmiany. Kierownik bez umiejętności zrządzania zamianą nie wpłynie na pracowników skutecznie, nie zmotywuje, nawet kosztem zmiany, uraty stanowiska. Kierownicy mają tego świadomość i nie chcą podejmować się wprowadzenia tak dużych zmian bez przygotowania.

Potrzeba: Potrzebny jest program, cykl budowania kompetencji, który obejmuje świadomość, kompetencje, samodzielność, konsekwencje w użytkowaniu. Pracownik musi zrozumieć po co wprowadzana jest zmiana, jak korzystać z nowego rozwiązania, musi też wiedzieć gdzie szukać dalszej pomocy w razie problemów. Szkolenie powinno być oparte na rzeczywistych procesach JST, a nie wyłącznie na funkcjach technicznych systemu. Projekt powinien obejmować zarówno kadrę zarządzającą JST, jak i pracowników odpowiedzialnych za IT, administrację oraz użytkowników końcowych. Wsparcie powinno rozpoczynać się na etapie przygotowania projektu i być kontynuowane przez cały okres wdrożenia oraz po jego zakończeniu.

Zapewnić lokalnego „ambasadora zmiany"

Każdy projekt powinien wskazywać osobę odpowiedzialną za koordynację wdrożenia po stronie JST. Jej zadaniem byłoby m.in.: koordynowanie prac, komunikacja z pracownikami, współpraca z wykonawcą, monitorowanie adopcji, zgłaszanie problemów, zapewnienie trwałości rozwiązania. To szczególnie ważne w małych JST, gdzie jedna osoba często pełni kilka funkcji i nie ma możliwości stworzenia rozbudowanego zespołu IT.

Zagrożenie: długotrwałe rozwiązywanie problemów, czas oczekiwania blokuje pracę jednostki. Ma to swoje konsekwencje w działaniach wielu osób i działów, bo to system powiązany. JST nie

mogą sobie pozwolić na dłuższą bezczynność i oczekiwanie w kolejce na pomoc. Przy braku zaopiekowania pracownicy będą powracać do poprzednich metod, narzędzi pracy, albo będą wyrażać sprzeciw wobec wprowadzaniu nowych rozwiązań. Problemy muszą być rozwiązywane szybko.

Potrzeba: JST podkreślały, że potrzebują osoby zatrudnionej u siebie do obsługi, pomocy, wsparcia innych pracowników stosujących OSS. Nie mogą liczyć wyłącznie na wsparcie zewnętrzne i zgłaszanie błędów, problemów na helpdesk. Powinna być zatrudniona osoba zaufana, dostępna w każdym momencie, otwarta na każdy problem – mały i duży.

Zbudować sieć współpracy JST

Postulowano stworzenie mechanizmu, w którym JST uczestniczące w programie mogłyby:

- wymieniać doświadczenia,
- dzielić się rozwiązaniami,
- zgłaszać problemy,
- wspólnie wypracowywać standardy,
- korzystać z doświadczeń innych jednostek.

Może to przyjąć formę otwartej platformy wsparcia połączonej z konsultacjami.

Bezpieczeństwo

Ryzyko porzucenia rozwiązania
Uczestnicy warsztatów zwrócili uwagę na ryzyko, jakie wiąże się z ewentualnością zaprzestania rozwoju danego rozwiązania typu OS przez stojącą za nim społeczność. Ryzyko takie jest szczególnie wysokie w przypadku rozwiązań, za którymi stoi niewielka liczba maintainerów czy wręcz jedna osoba. W przypadku porzucenia rozwiązania, które zostałoby wdrożone w konkretnej JST, jednostka ta zostałaby ostatecznie zmuszona do kolejnej migracji.

Choć uczestnicy zgadzali się że całkowita eliminacja takiego zagrożenia nie jest możliwa, to jednocześnie wskazywano szereg rozwiązań pozwalających na zmniejszenie poziomu ryzyka. Były wśród nich:

- Wykorzystanie publicznie znanych metodyk oceny projektów OSS do weryfikacji żywotności poszczególnych rozwiązań: wskazywano na CHAOSS i OpenSSF.
- Stworzenie katalogu rozwiązań, których wdrożenie będzie wspierane – na podstawie ocen w wyżej wspomnianych metodykach.
- Premiowanie projektów, za którymi stoją organizacje stewardzkie.

Pojawiły się także głosy wzywające do oparcia się wyłącznie na rozwiązaniach o przynajmniej kilkuletniej historii rozwoju. Nie było jednak pełnej zgody w tym zakresie – część uczestników podkreślała, że może to spowodować wykluczenie z wdrożenia użytecznych rozwiązań. Jeden z uczestników zaproponował także, by wdrażać wyłącznie rozwiązania funkcjonujące już w instytucjach publicznych na terenie UE/EOG – padły jednak kontrargumenty związane z trudnością weryfikacji takich wdrożeń.

W odpowiedzi na pytanie dotyczące tego, czy dostawcy rozwiązań dla JST powinni być zobowiązani do zapewnienia alternatywy funkcjonalnej w razie zaprzestania rozwoju danego

rozwiązania, większość uczestników opowiedziała się przeciwko takiemu obowiązkowi. Jak argumentowano, znacząco podwyższyłoby to koszty wdrożeń w sytuacji, gdy podobny efekt można osiągnąć innymi metodami – dzięki wybraniu rozwiązań o minimalnym ryzyku, trzymaniu danych jedynie w otwartych formatach, obowiązkowi stworzenia planu wyjścia i przeprowadzenia testów oraz zapewnieniu możliwości migracji w razie realizacji najgorszego scenariusza. Migracji takiej nie musiałby przeprowadzać podmiot, który przeprowadzał wdrożenie.

Bezpieczeństwo techniczne

Uczestnicy warsztatów co do zasady zgodnie podkreślali, że rozwiązania typu OSS nie są inherentnie bardziej lub mniej bezpieczne od rozwiązań własnościowych – jednak mają inne wymagania związane z modelem wdrożeń, w którym nie ma komercyjnego podmiotu stale odpowiedzialnego za rozwój i łatanie danego programu. W związku z tym konieczne jest ich zdaniem zapewnienie szeregu zabezpieczeń – zarówno w ramach realizacji wymagań wynikających z prawa (w tym CRA czy uKSC) jak i w ich uzupełnieniu. Zdaniem uczestników, szczegółowe wymagania określone w dokumentacji muszą prowadzić do zapewnienia odpowiedniego poziomu bezpieczeństwa, integralności i poufności danych.

Wśród koniecznych zabezpieczeń wymieniano:

1. Obowiązek przeprowadzenia audytów przedwdrożeniowych;
2. Prowadzenie testów penetracyjnych i monitorowania podatności oraz ich łatania;
3. Szyfrowanie danych w locie i spoczynku;
4. Zapewnienie możliwości odzyskania danych dzięki kopiom zapasowym – w tym offsite'owym. Uczestnicy opowiedzieli się też powszechnie za trzymaniem danych jedynie na terenie UE/EOG;
5. Weryfikację zależności i zagrożeń w ramach łańcucha dostaw, m.in. dzięki SBOM.

W ramach dyskusji na temat bezpieczeństwa łańcucha dostaw pojawił się też postulat weryfikacji podmiotów, które będą podejmowały się realizacji wdrożeń na rzecz JST. Spotkał się on jednak z kontrargumentami związanymi z tym, że w Polsce niewiele podmiotów będzie mogło zaprezentować referencje z dotychczasowych wdrożeń ze względu na małą powszechność rozwiązań typu OSS w kraju.

Wsparcie dla JST

Jak wielokrotnie wspominano, JST często dysponują niewielkimi budżetami na wdrożenia i nie posiadają zespołów IT w skali pozwalającej na zapewnienie użytkownikom kompleksowego wsparcia – zwłaszcza przy wdrażaniu nowego rozwiązania. Dlatego podkreślano konieczność zapewnienia wsparcia ze strony podmiotów zakontraktowanych przez JST, w tym w ramach monitorowania i łatania podatności.

Bardzo często pojawiał się postulat realizacji kompleksowych szkoleń z cyberbezpieczeństwa zarówno dla administratorów IT, jak „zwykłych" użytkowników.

Jeden z uczestników proponował wdrożenie programu typu bug bounty dla rozwiązań wybranych do wdrożeń w ramach pilotaży.

Często pojawiały się też postulaty dotyczące poziomu krajowego, a nie planowanych pilotaży – by docelowo na poziomie centralnym istniał mechanizm centralnej weryfikacji rozwiązań typu OSS, których wdrożenia administracja wspiera.