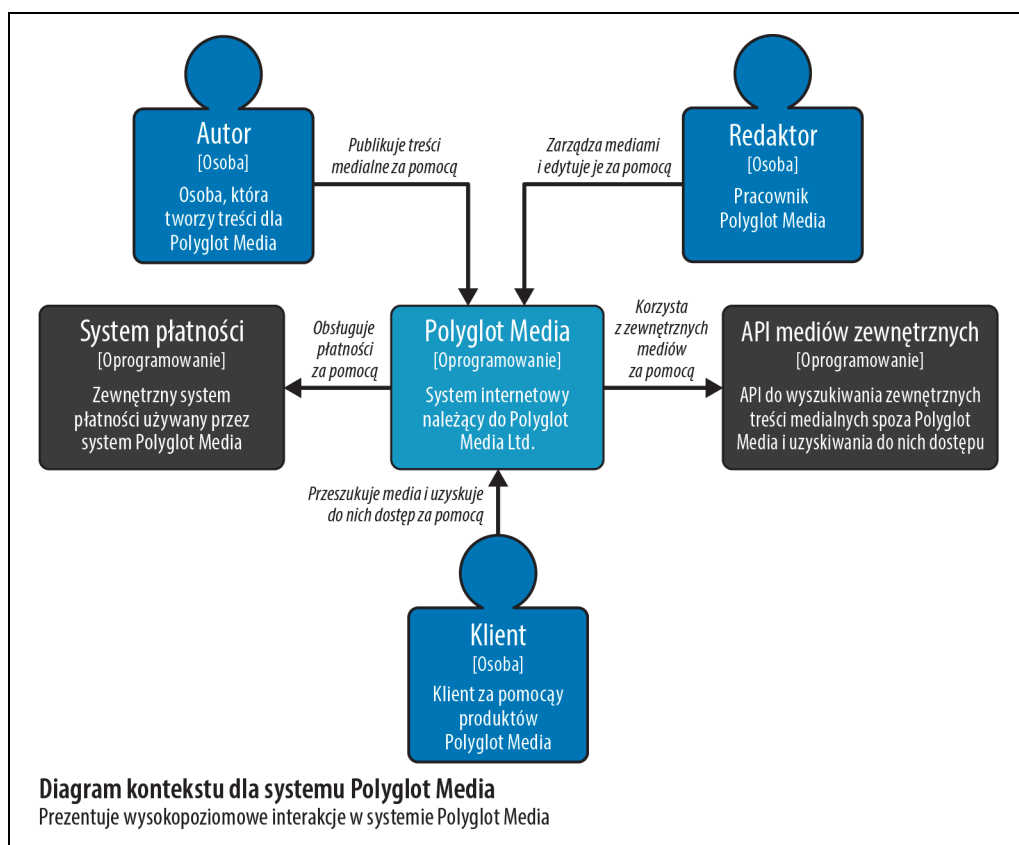


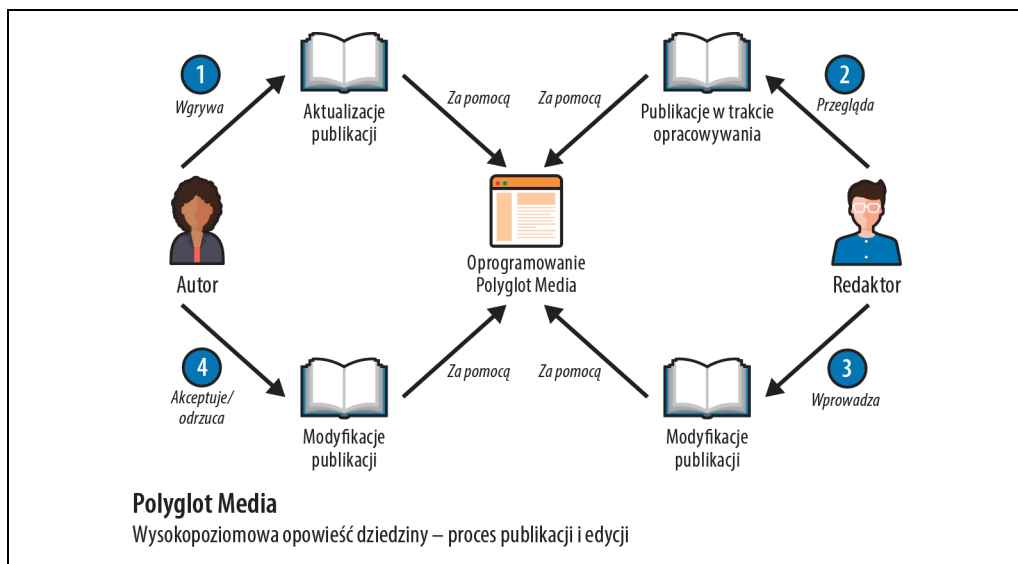
Kolorowe wersje rysunków do książki „Wzorce komunikacji. Przewodnik dla programistów i architektów”

ROZDZIAŁ 1.

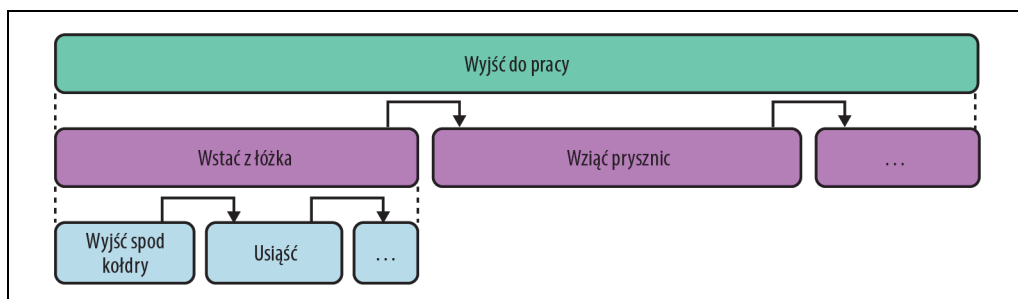
Podstawy komunikacji



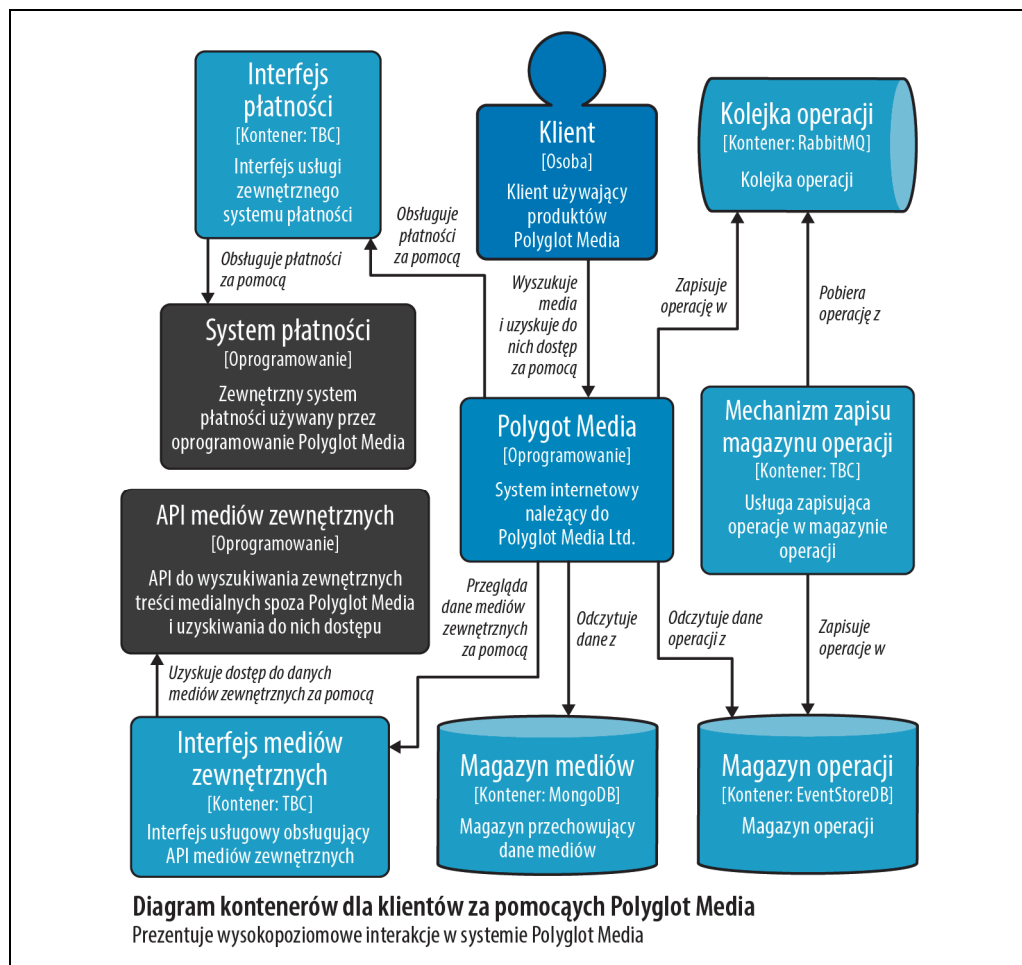
Rysunek 1.2. Diagram kontekstowy C4 przydatny dla większości odbiorców



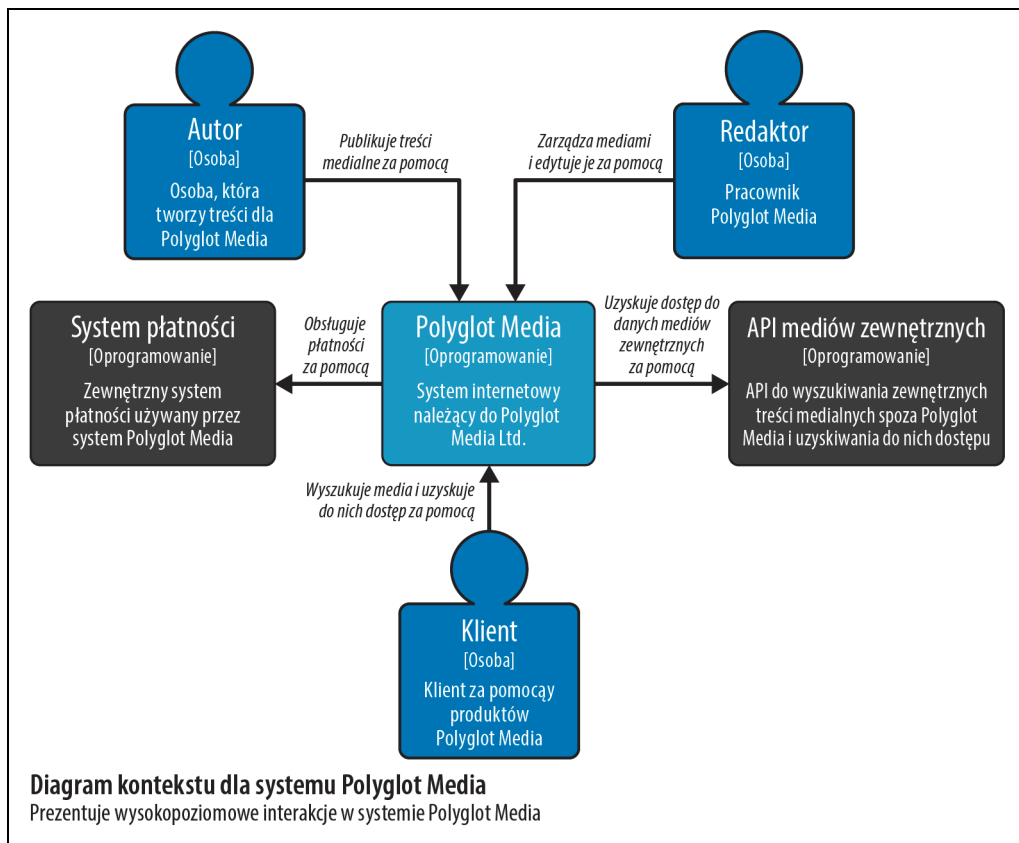
Rysunek 1.3. Diagram opowieści dziedziny przeznaczony dla ról biznesowych i osób przekształcających potrzeby biznesowe w wymagania techniczne, takich jak architekci i analitycy biznesowi



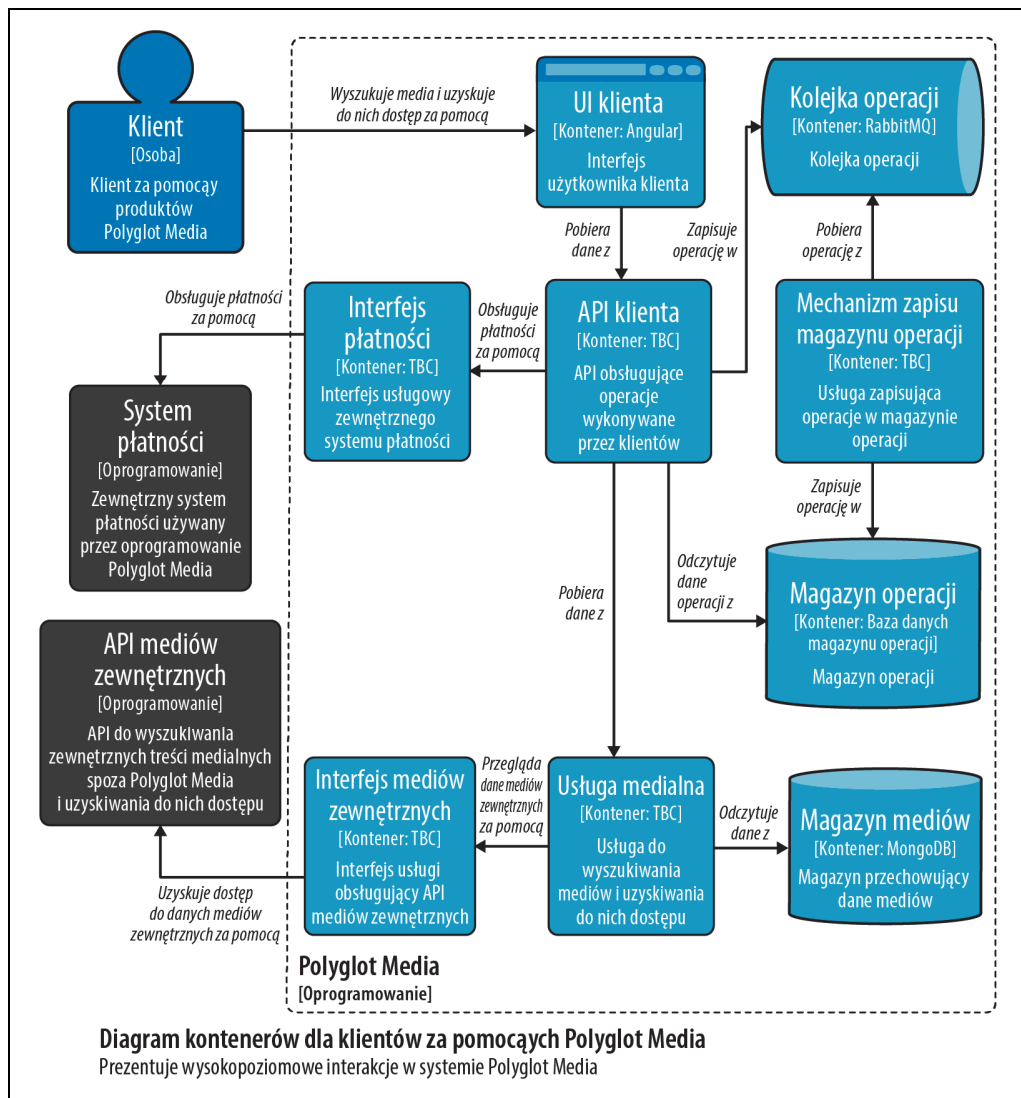
Rysunek 1.4. Poziomy abstrakcji w codziennym życiu



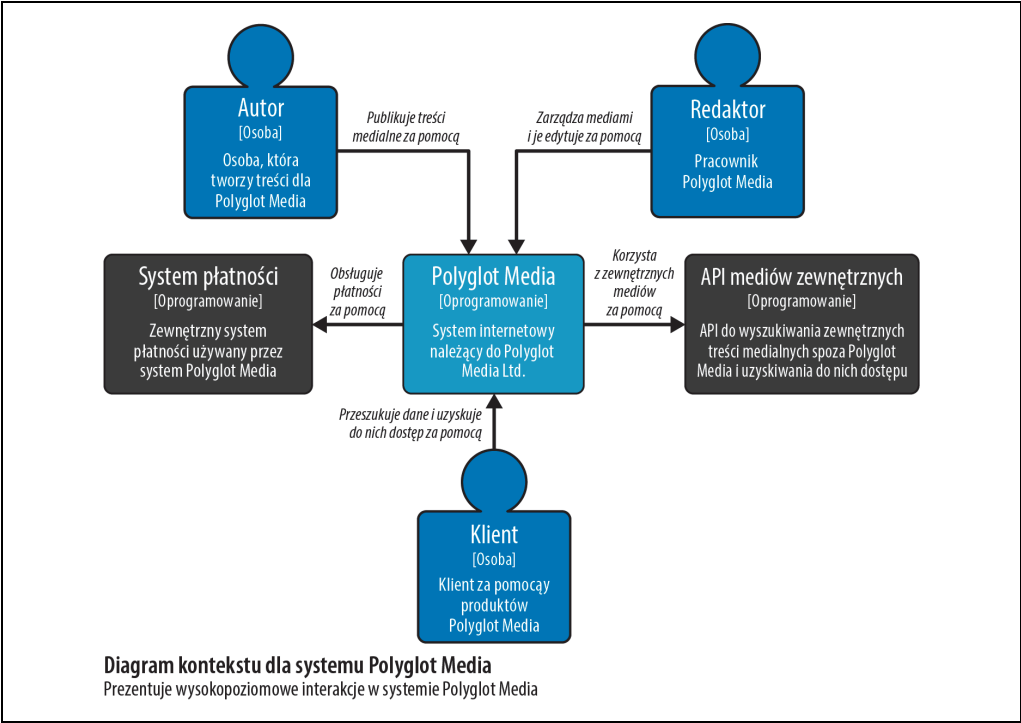
Rysunek 1.5. Diagram C4 prezentujący zarówno poziom abstrakcji kontekstu, jak i kontenerów (antywzorzec)



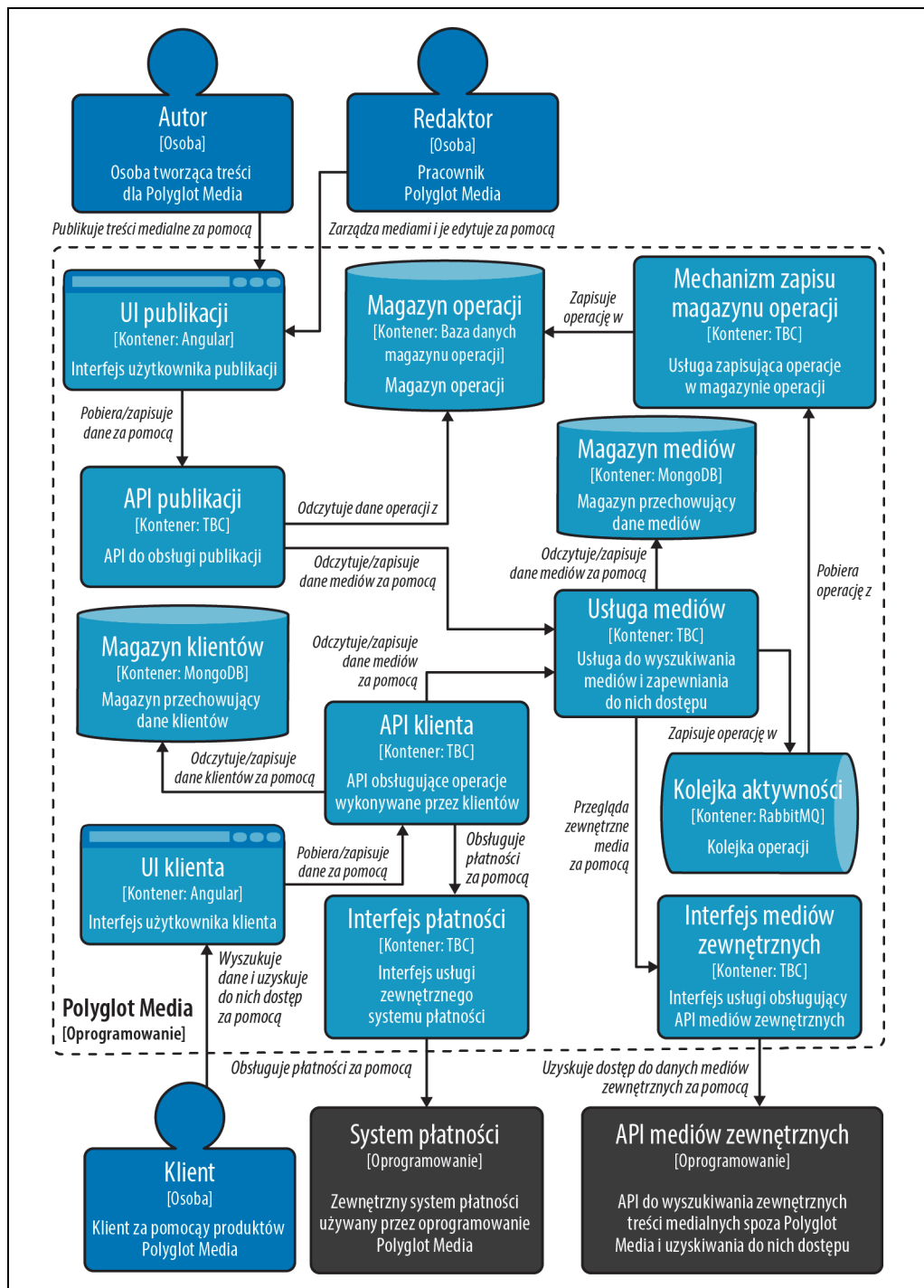
Rysunek 1.6. Diagram kontekstu C4



Rysunek 1.7. Diagram kontenerów C4

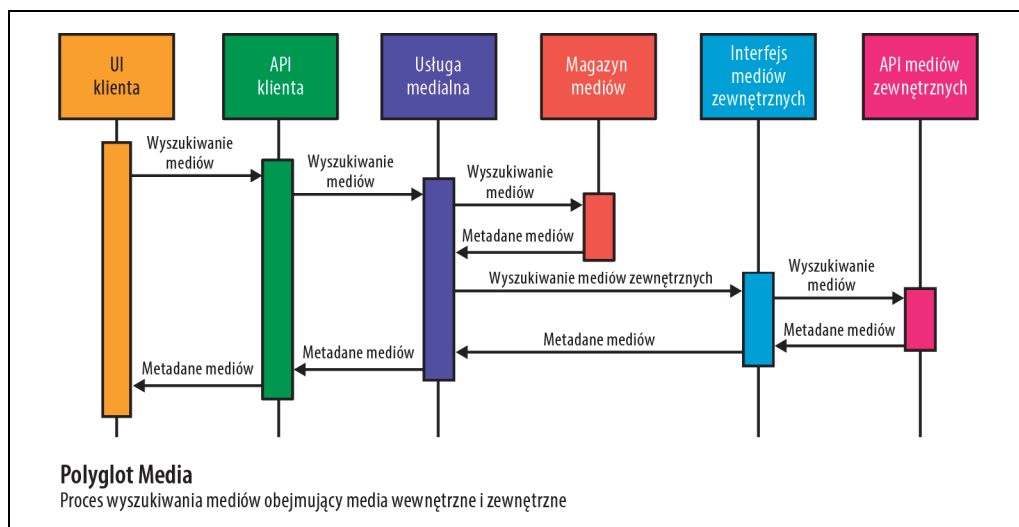


Rysunek 1.8. Diagram kontekstu C4

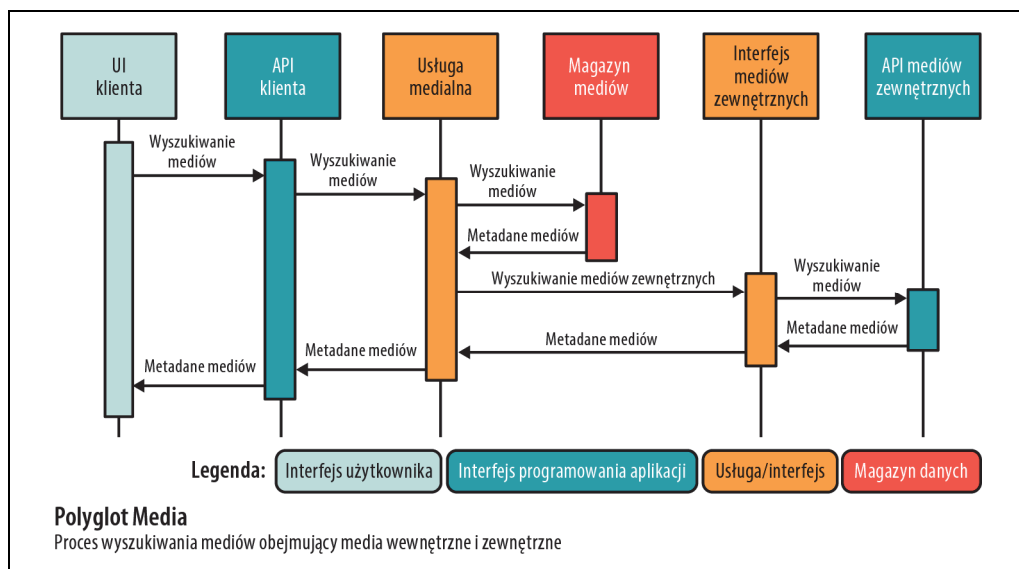


Rysunek 1.9. Diagram kontenerów C4 dla Polyglot Media (przedstawiający wysokopoziomowe interakcje w oprogramowaniu Polyglot Media)

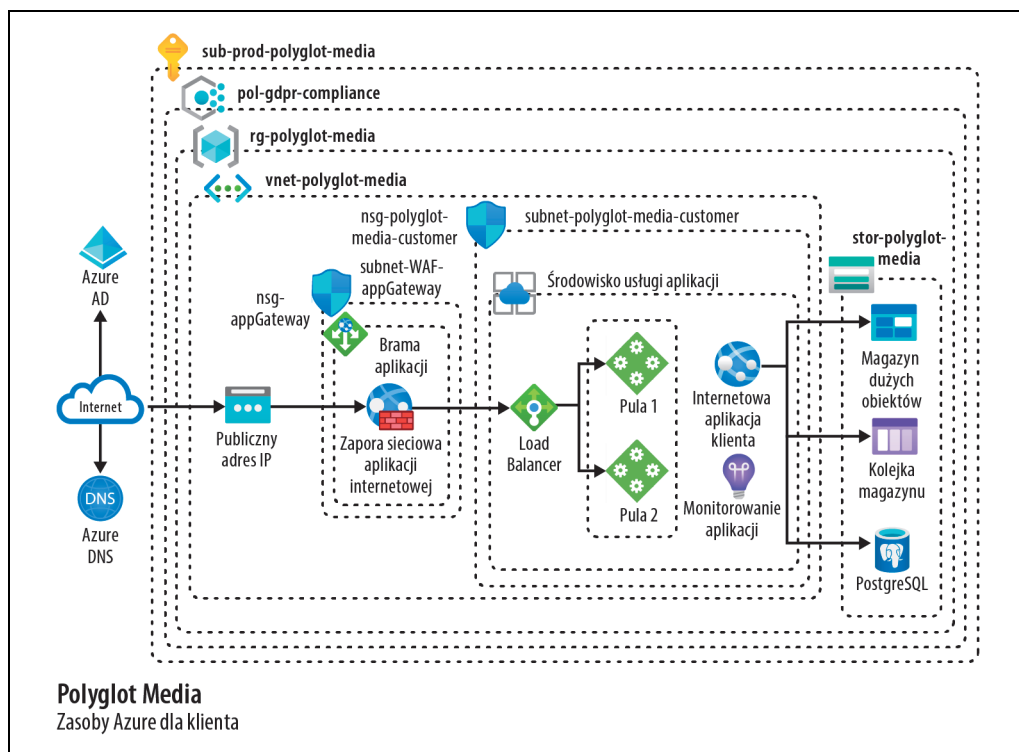
Uporządkuj bałagan



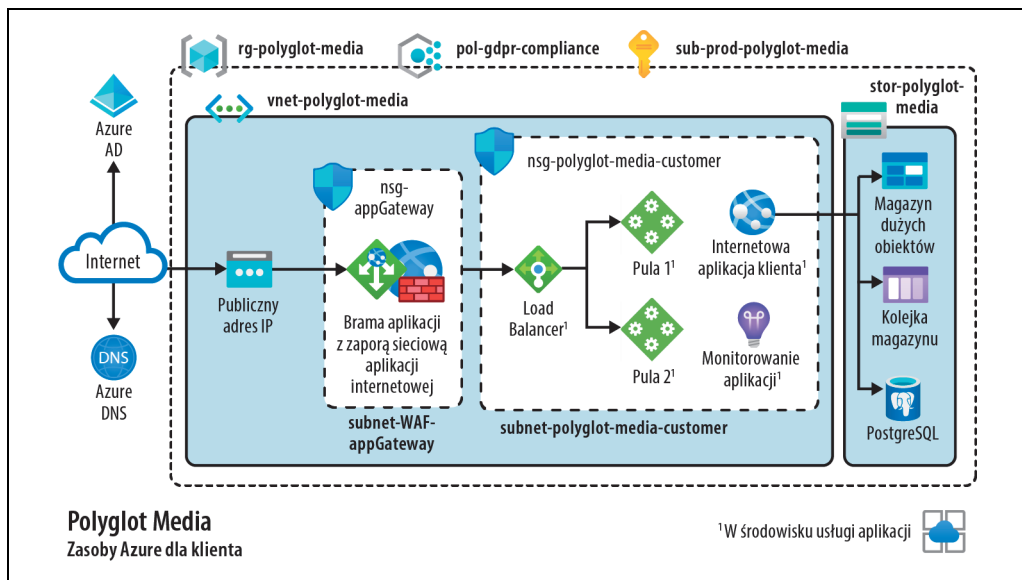
Rysunek 2.1. Tęczowy diagram sekwencji (kolorowa wersja diagramu dostępna w materiałach)



Rysunek 2.2. Kolory zastosowane w celu pogrupowania według typu (kolorowa wersja diagramu dostępna w materiałach)

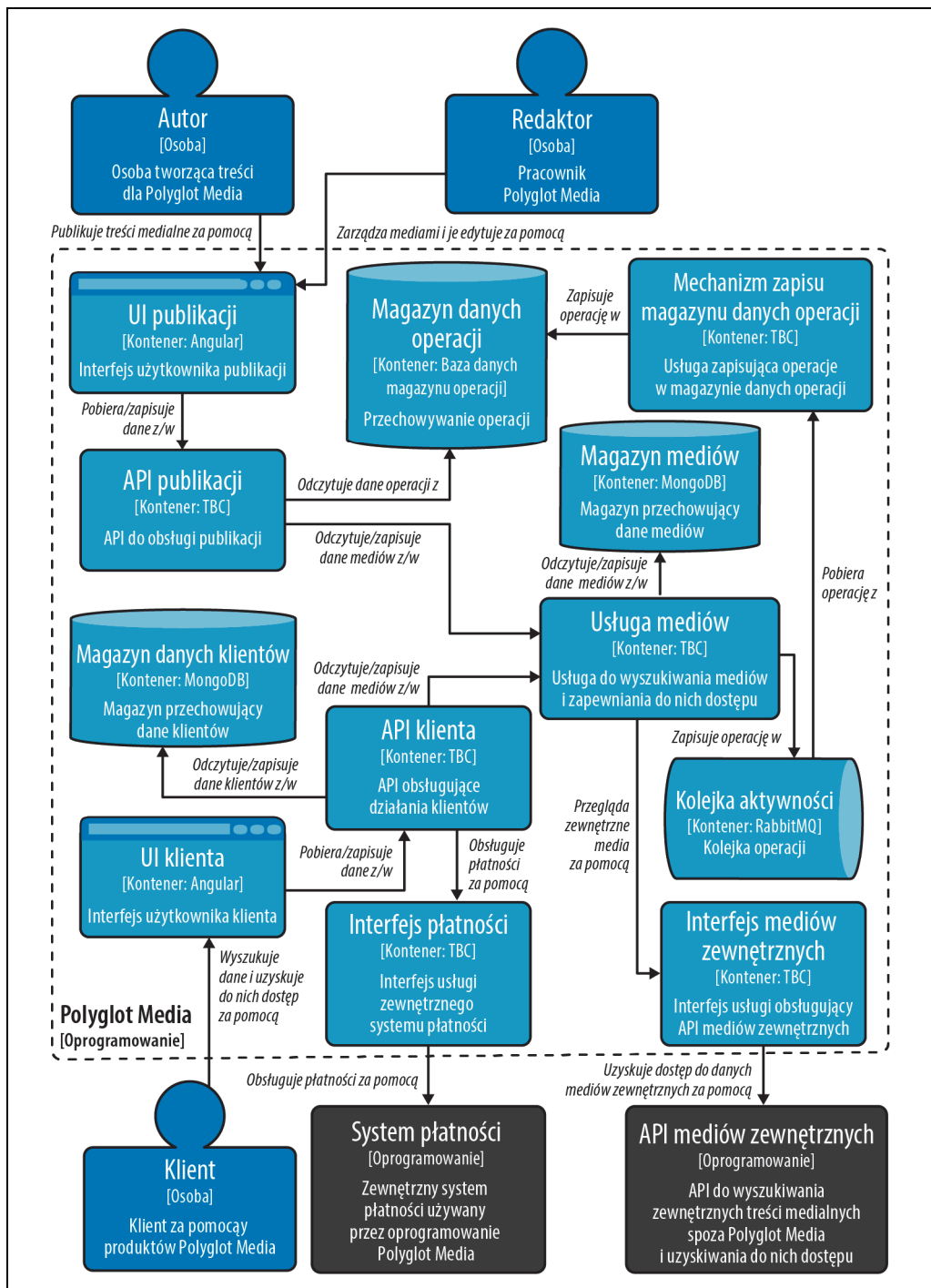


Rysunek 2.3. Ramki w ramach w ramach na diagramie zasobów chmurowych (antywzorzec)

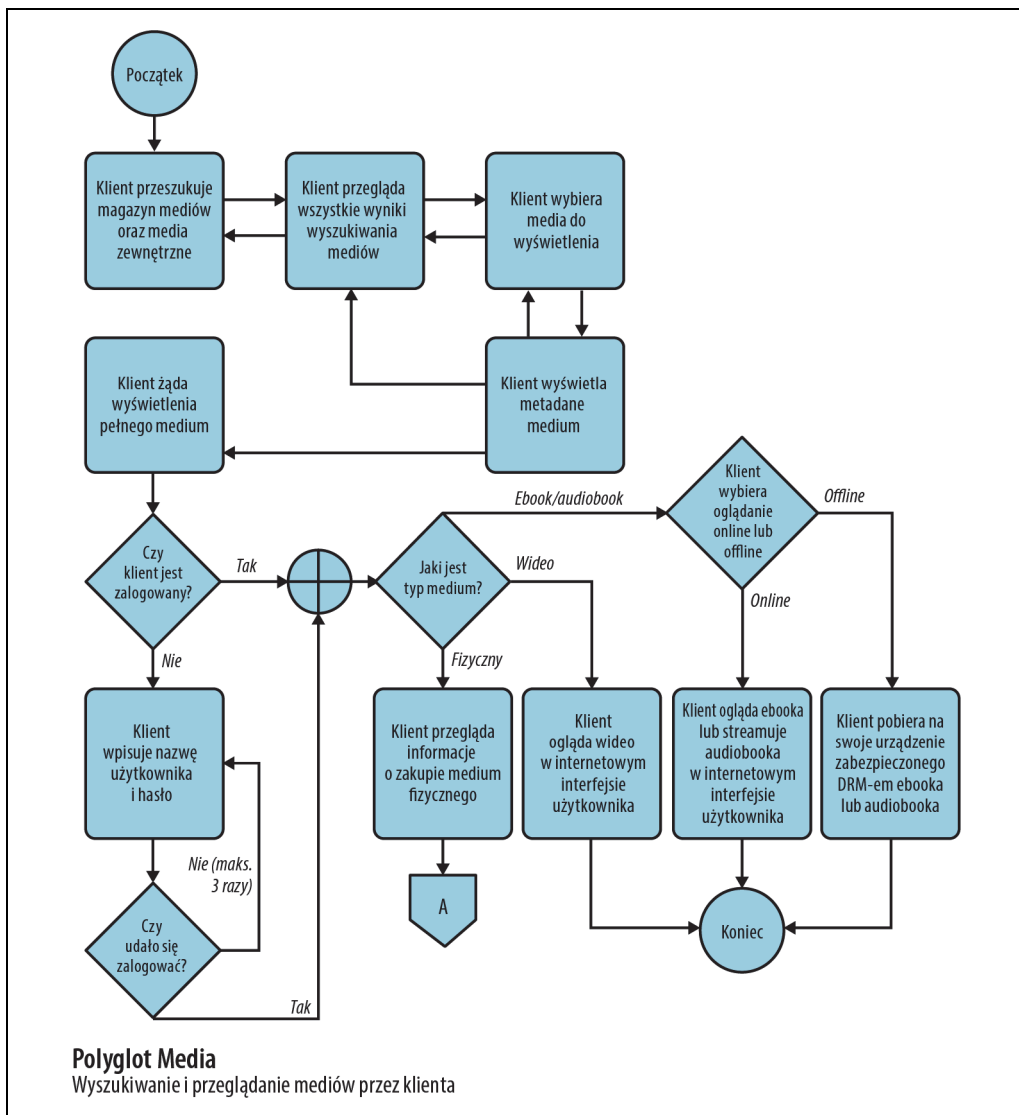


Rysunek 2.4. Uporządkowany diagram zasobów chmurowych

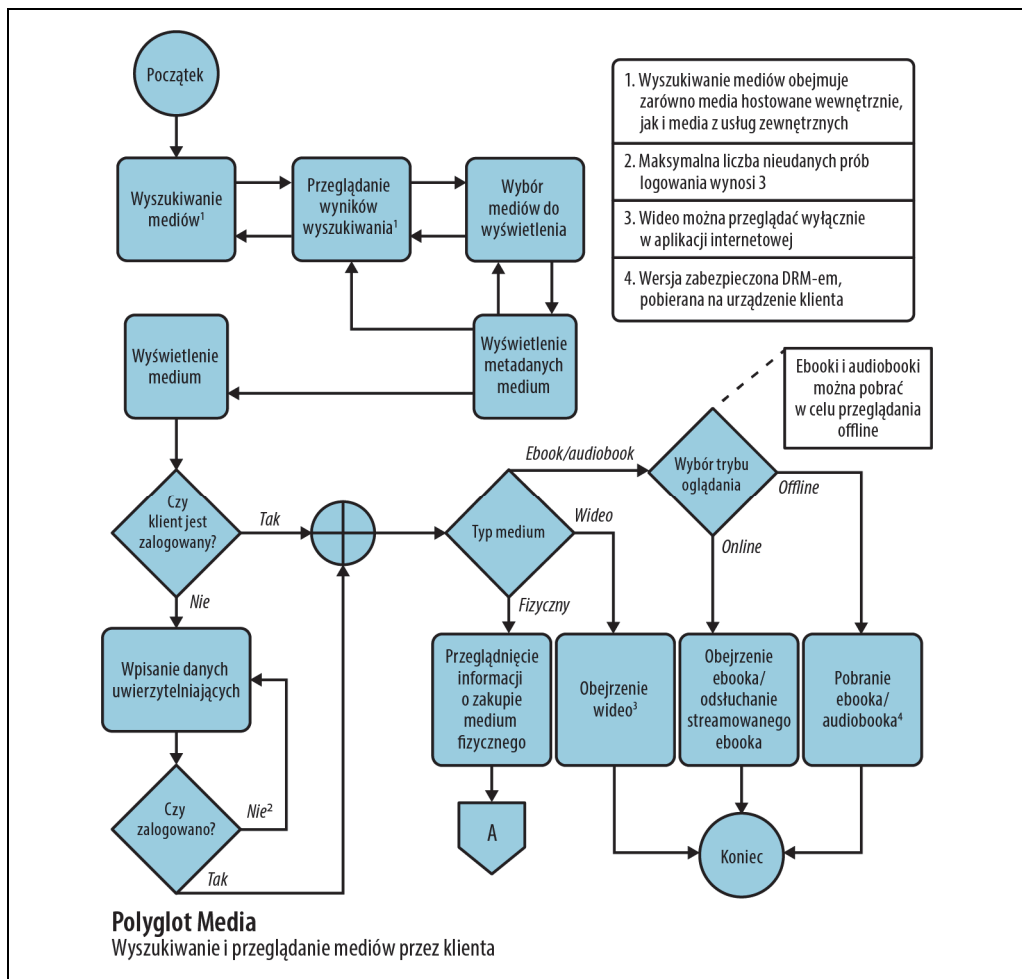
Rysunek 2.5. Pajęczyna na diagramie kontenerów C4 oprogramowania Polyglot Media (antywzorzec)



Rysunek 2.6. Uprządkowany diagram C4

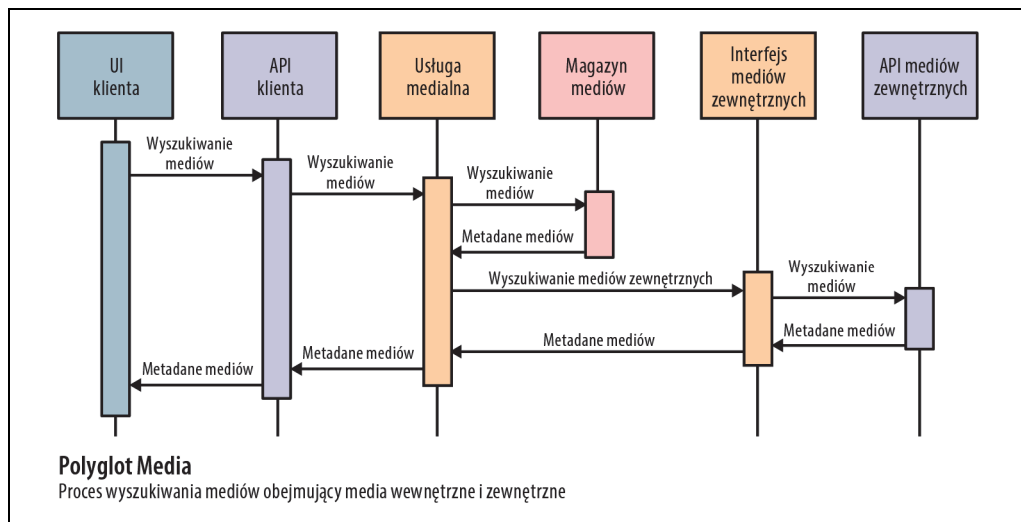


Rysunek 2.7. Diagram przepływu ze zbyt dużą ilością tekstu (anty wzorzec)

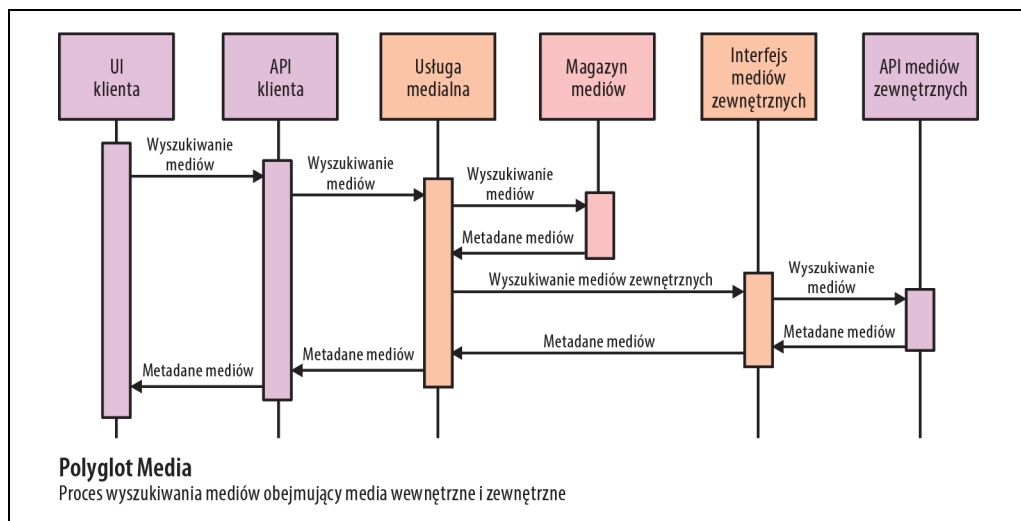


Rysunek 2.8. Diagram przepływu z nadmiarowym testem przeniesionym do adnotacji i odwołań

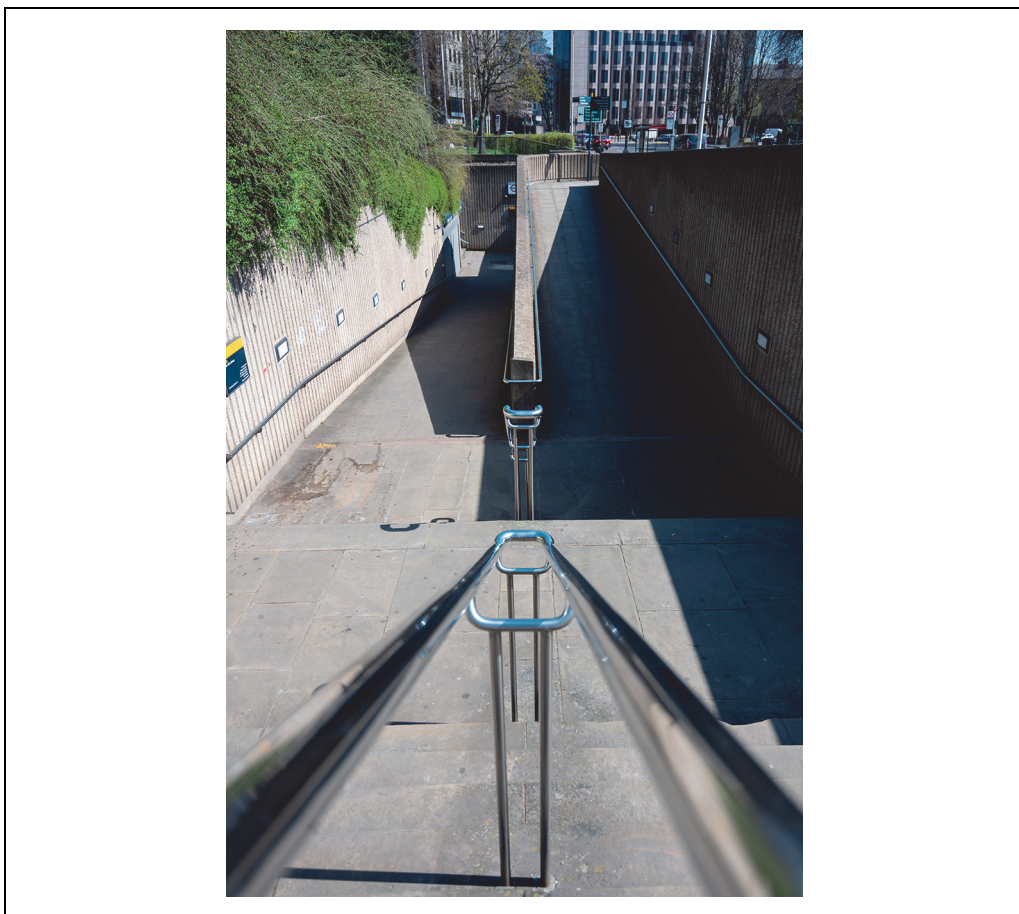
Dostępność



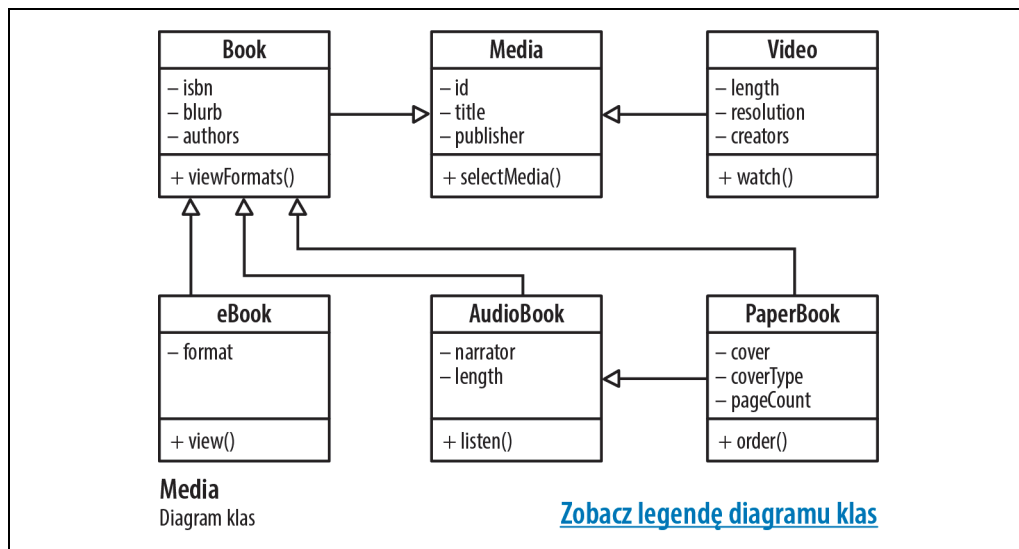
Rysunek 3.2. Diagram przepływu prezentujący niektóre standardowe kolory dostępne w aplikacji draw.io (kolorowa wersja diagramu dostępna w materiałach)



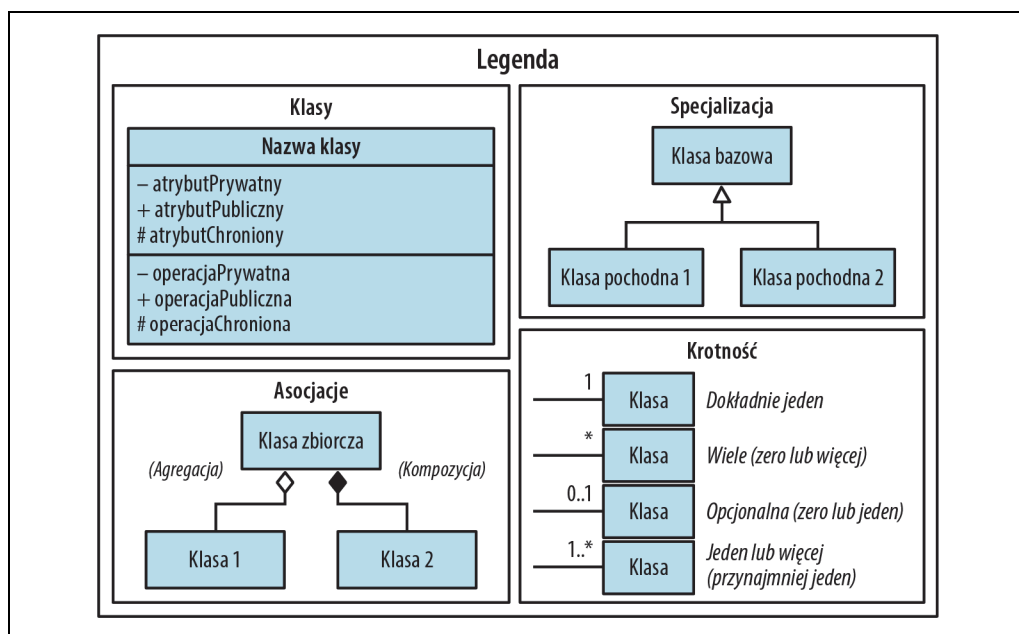
Rysunek 3.3. Symulacja deuteranopii (nierozpoznawanie barwy zielonej) na rysunku 3.2 (kolorowa wersja diagramu dostępna w materiałach)



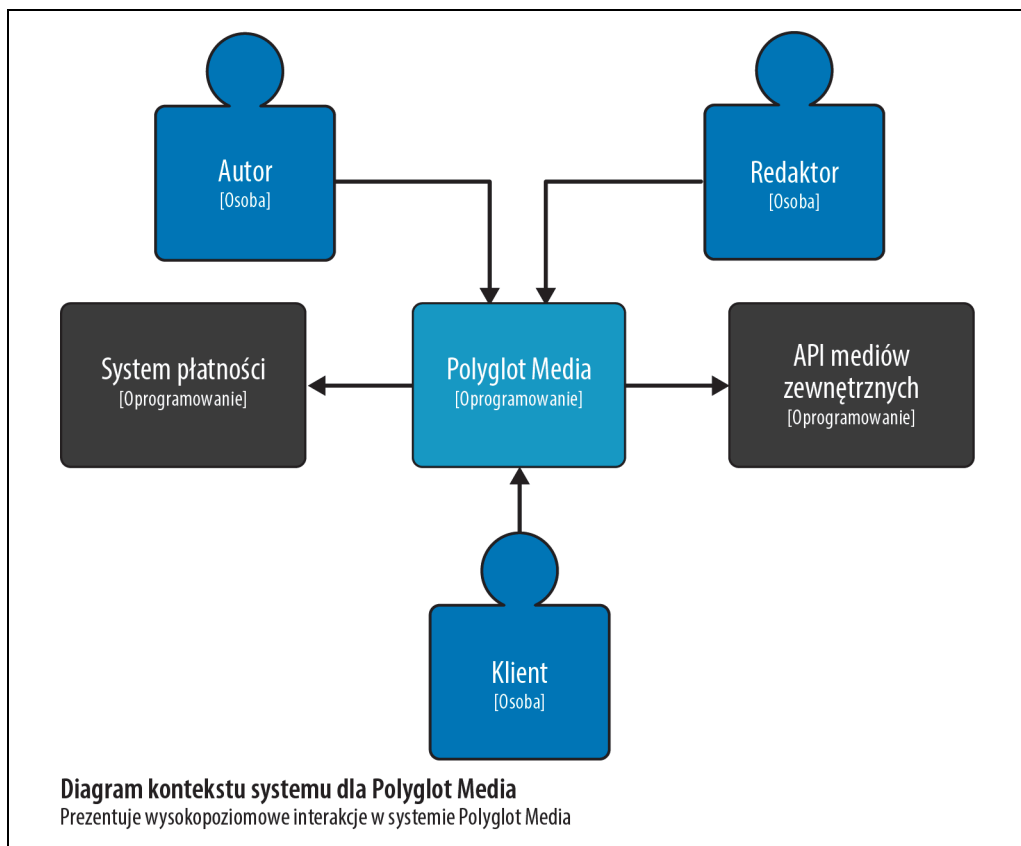
Rysunek 3.4. Rampy dla tych, którzy ich potrzebują lub chcą z nich korzystać, oraz stopnie dla każdego, kto może i chce używać schodów



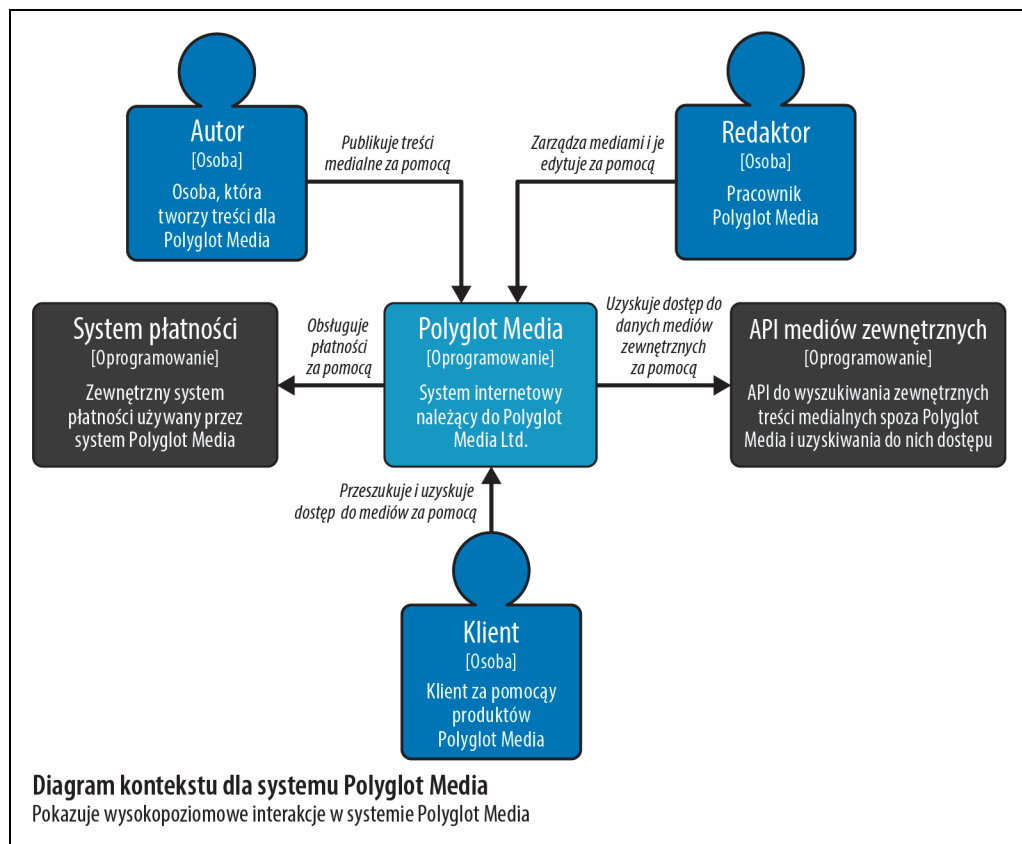
Rysunek 3.5. Dołączanie odnośnika do legendy na stronie WWW lub w dokumentacji



Rysunek 3.6. Legenda UML pokazująca dziwności języka UML

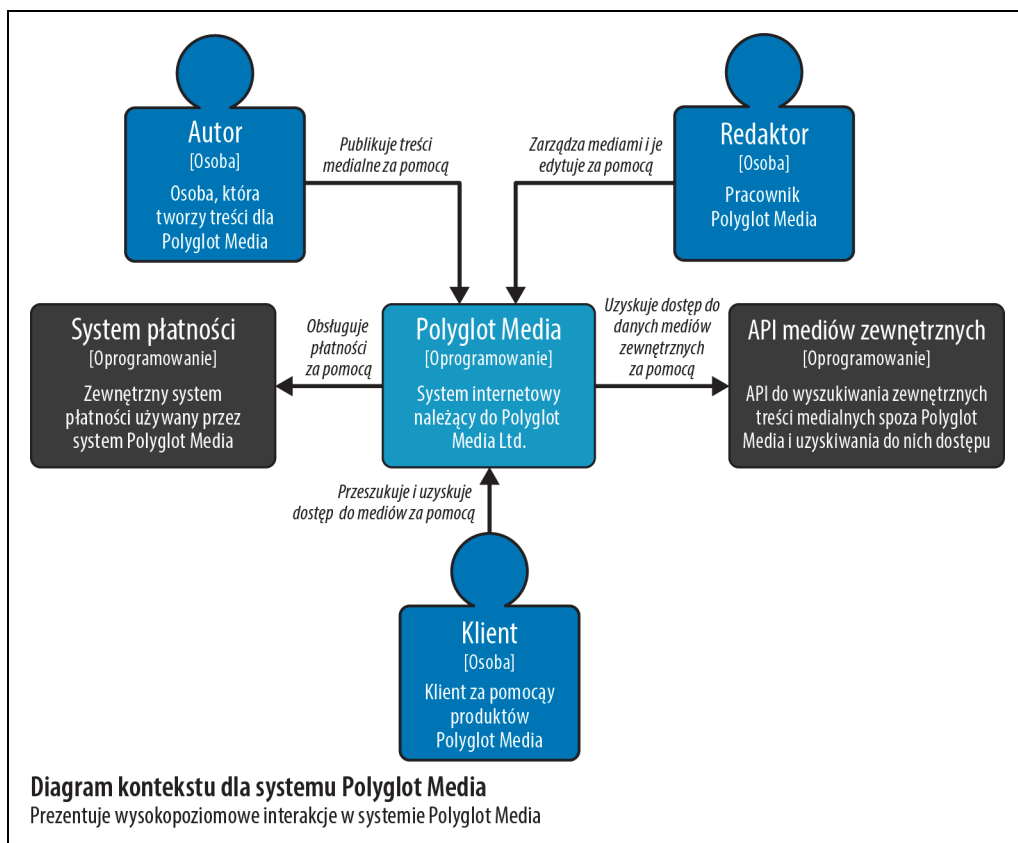


Rysunek 3.7. Diagram kontekstu C4 z nielicznymi etykietami i tekstem dostarczający niewielki kontekst (antywzorzec)

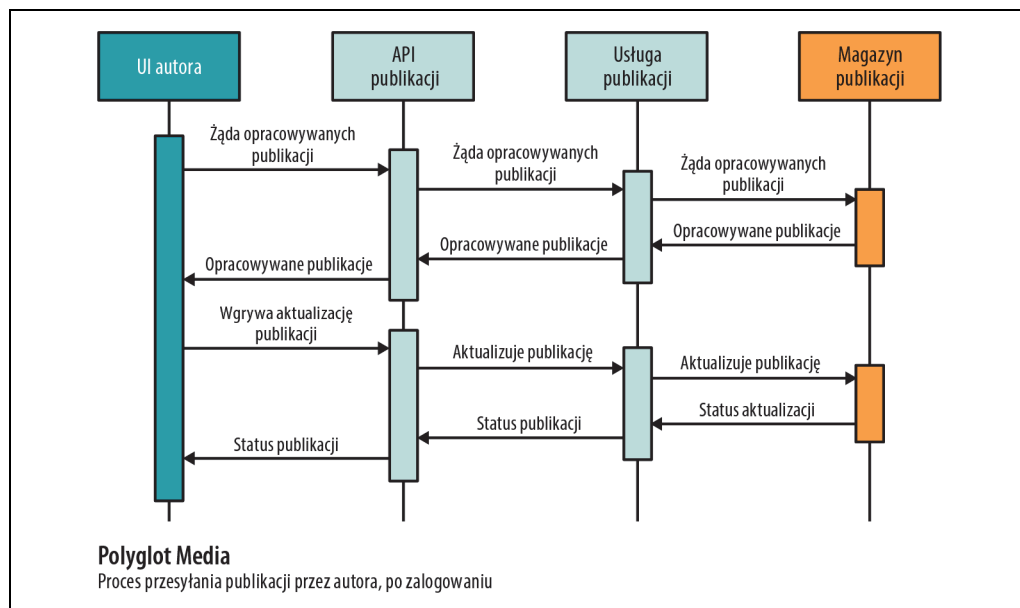


Rysunek 3.8. Diagram kontekstu C4 prezentujący odpowiednio wysoki poziom informacji zawartych w etykietach i opisach

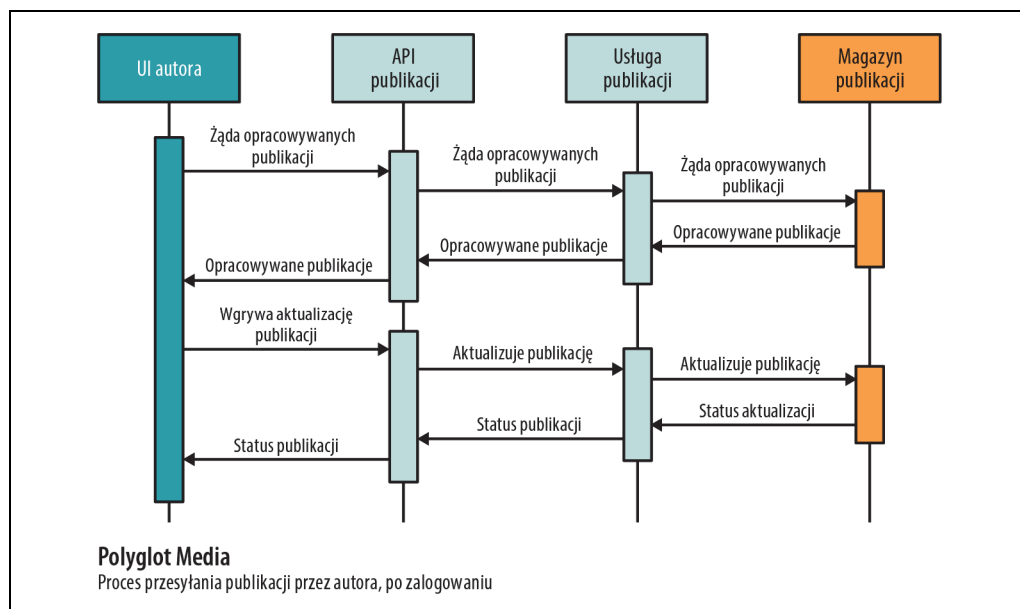
Narracja



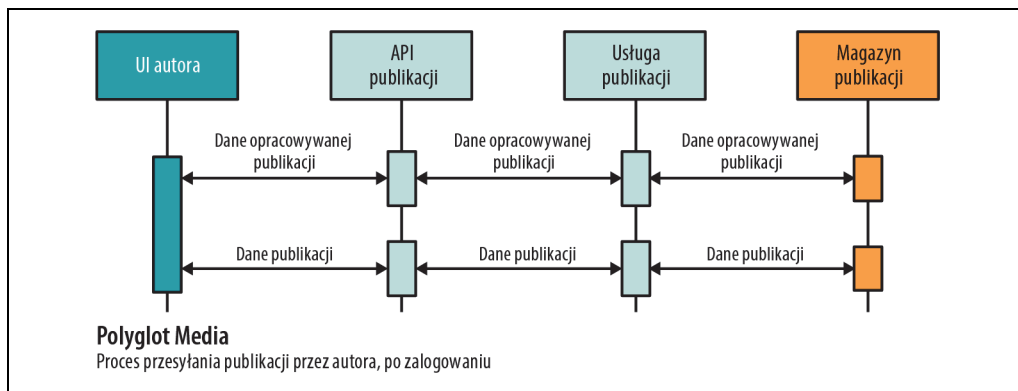
Rysunek 4.3. Diagram kontekstu C4 prezentujący odpowiednio wysoki poziom informacji zawartych w etykietach i opisach



Rysunek 4.6. Diagram sekwencji pokazujący żądania przepływające od lewej do prawej i odpowiedzi przepływające od prawej do lewej

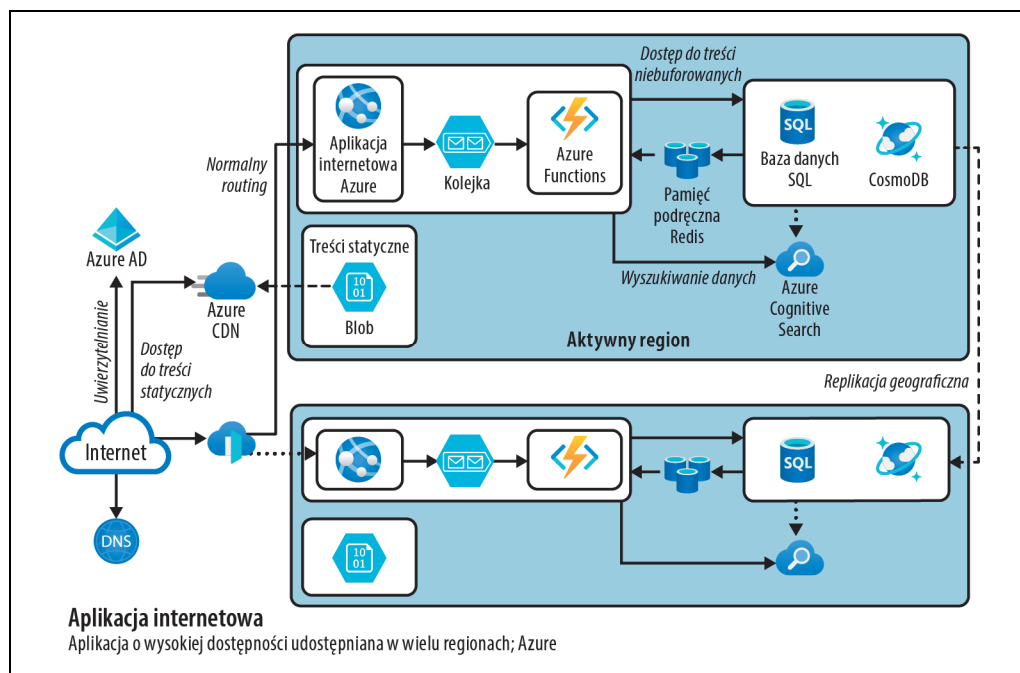


Rysunek 4.7. Diagram sekwencji z relacjami jednokierunkowymi

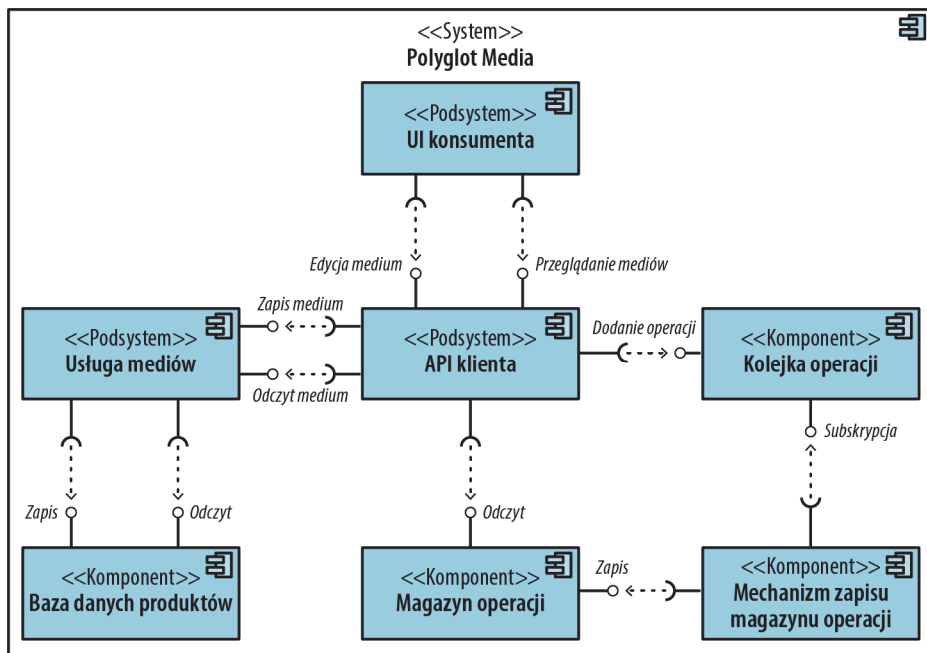


Rysunek 4.8. Diagram sekwencji z relacjami dwukierunkowymi

Notacja

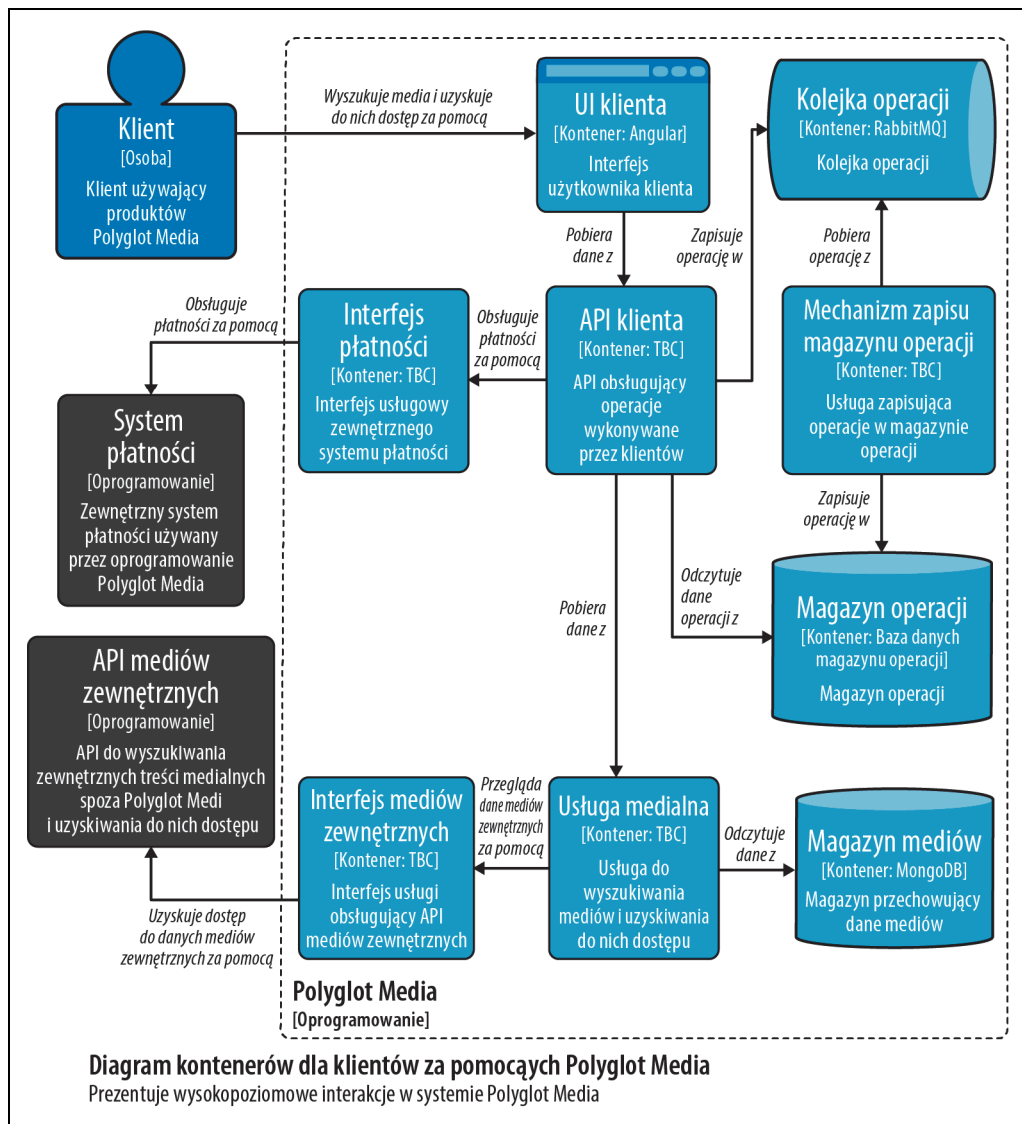


Rysunek 5.1. Diagram dostawy usług chmurowych z etykietami (u góry) i bez nich (u dołu)

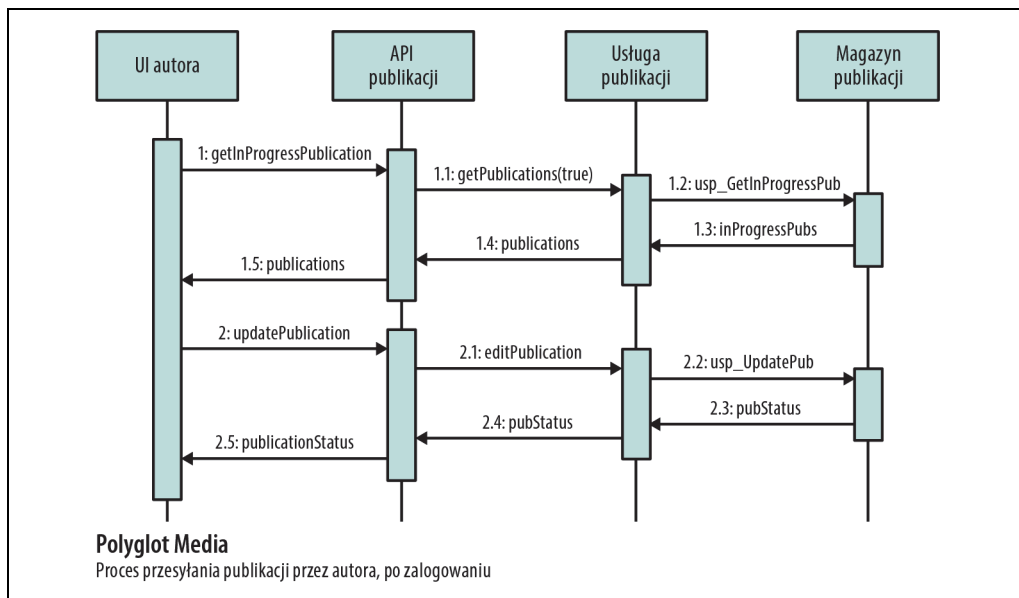


Polyglot Media – publikacja
Diagram komponentów UML

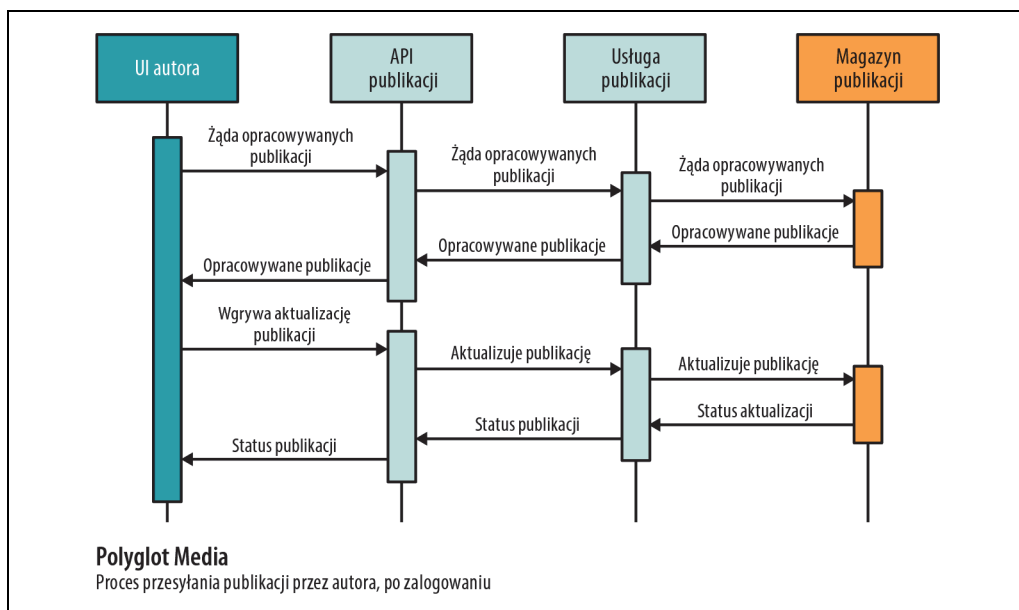
Rysunek 5.2. Diagram komponentów UML



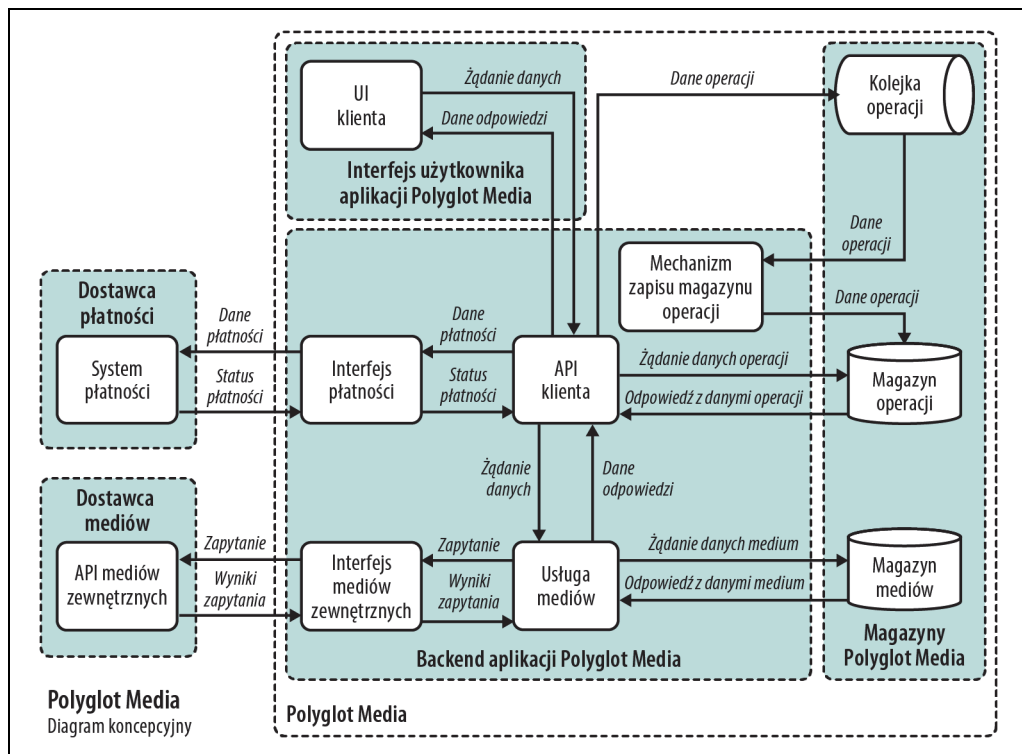
Rysunek 5.3. Diagram kontenerów C4



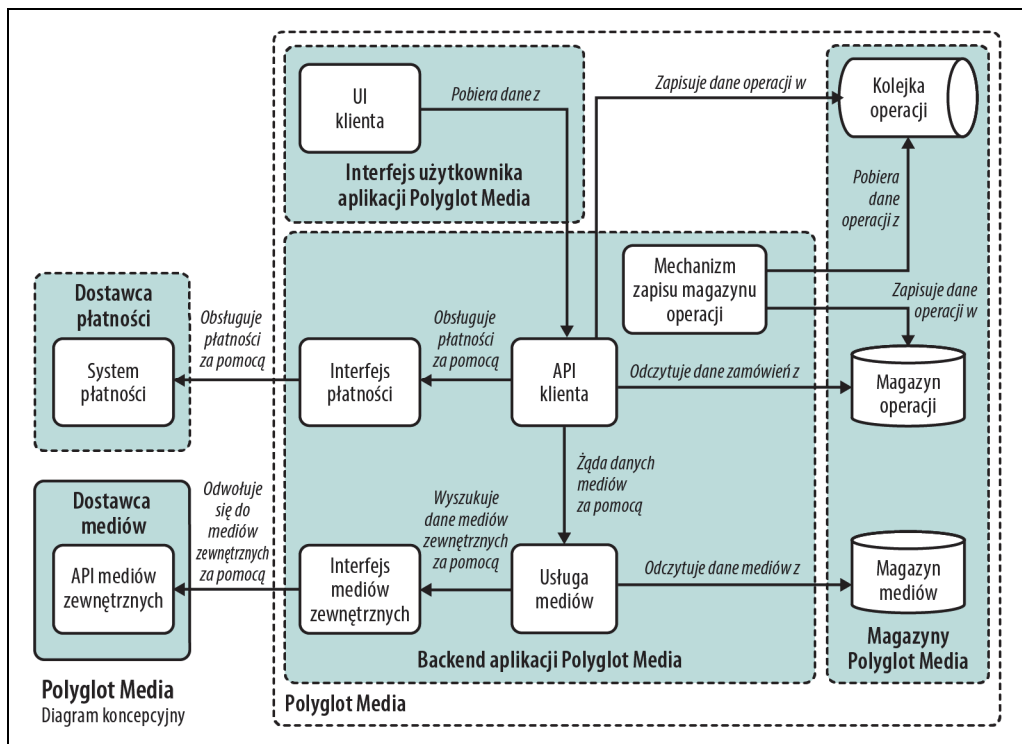
Rysunek 5.4. Diagram sekwencji UML



Rysunek 5.5. Uproszczony diagram sekwencji

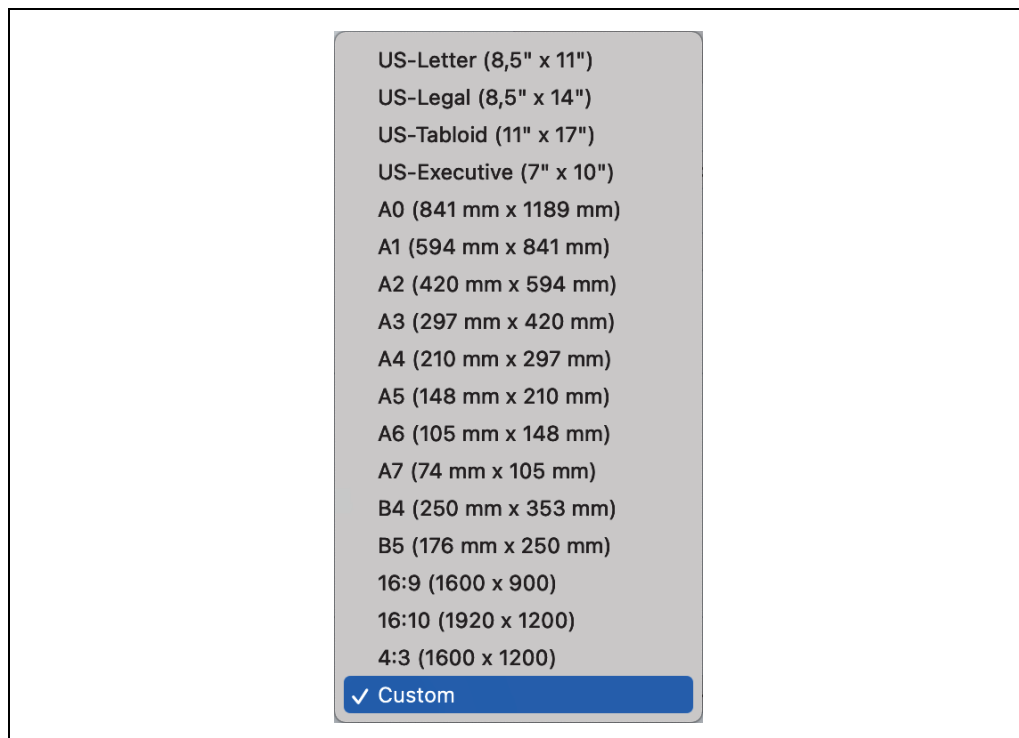


Rysunek 5.6. Nie mieszaj struktury i zachowania, jak zrobiono na tym diagramie (antywzorzec)

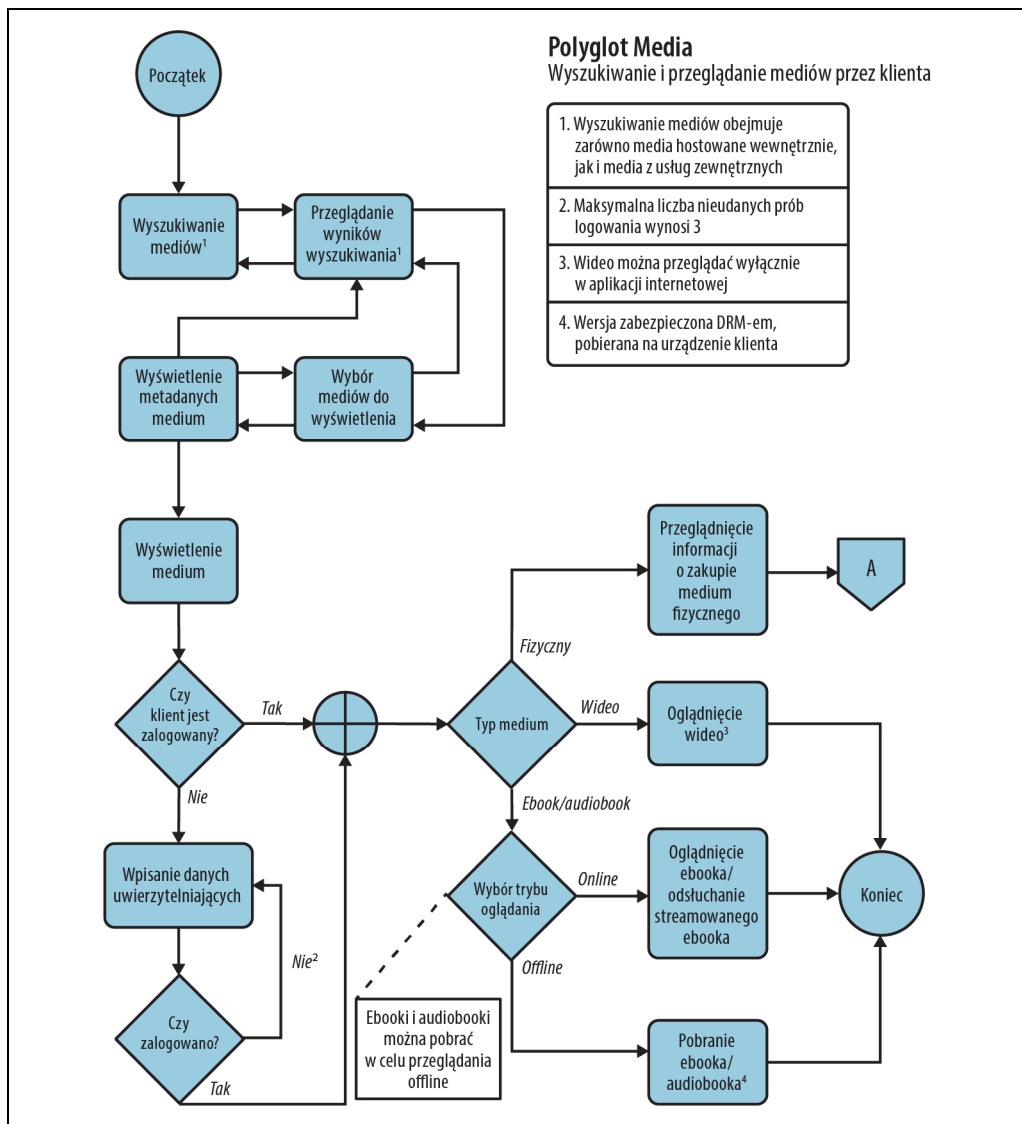


Rysunek 5.7. Diagram strukturalny na koncepcyjnym poziomie abstrakcji

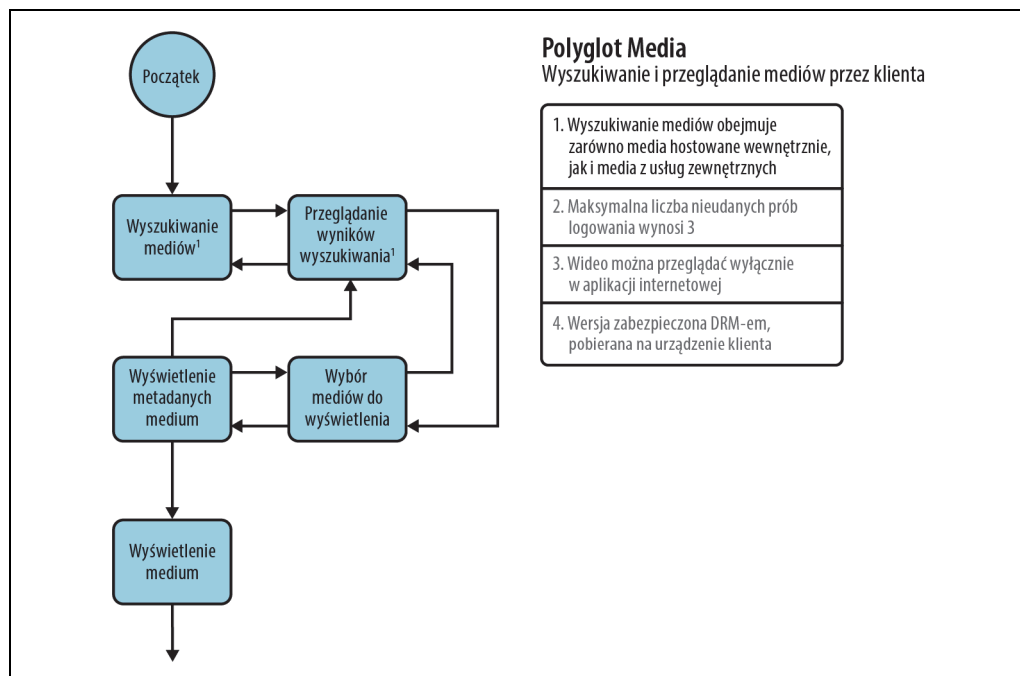
Kompozycja



Rysunek 6.1. Przykładowe opcje obszaru roboczego/ strony w aplikacji drawio.io



Rysunek 6.4. Wersja diagramu przepływu w układzie pionowym, która nie będzie dobrze wyglądać wyświetlona w układzie poziomym

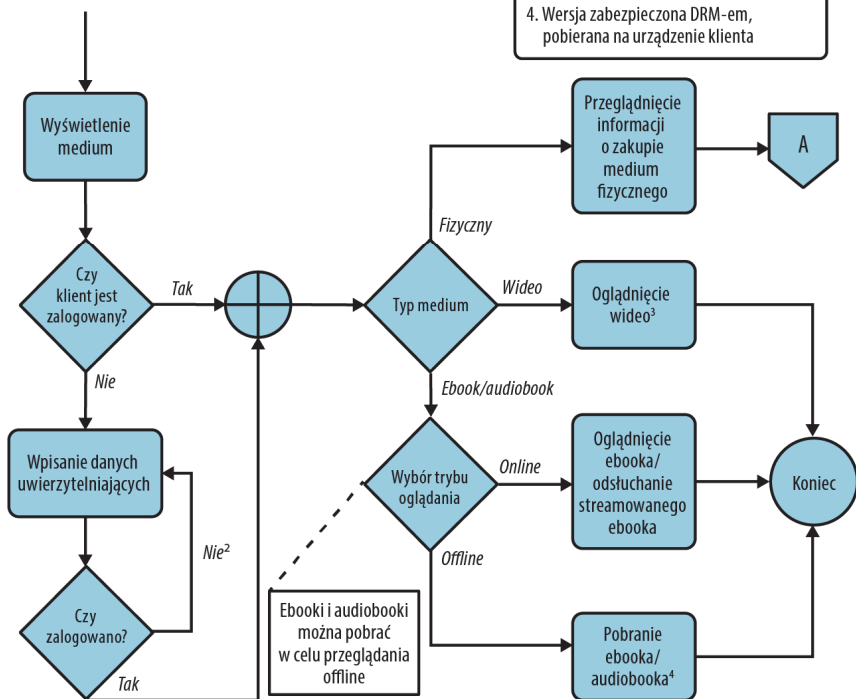


Rysunek 6.5. Przykładowy fragment diagramu z rysunku 6.4.

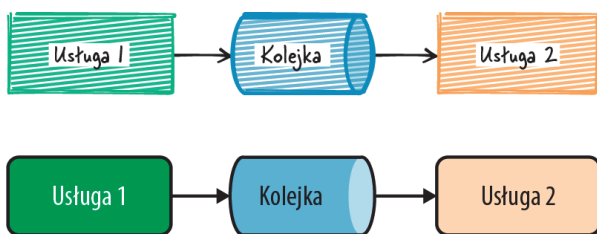
Polyglot Media

Wyszukiwanie i przeglądanie mediów przez klienta

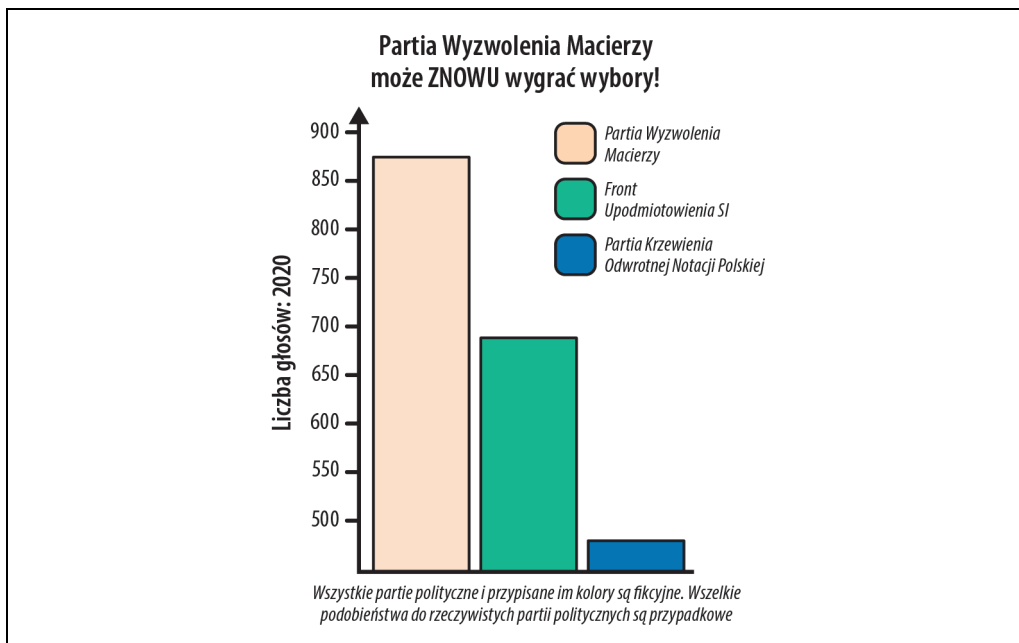
1. Wyszukiwanie mediów obejmuje zarówno media hostowane wewnętrznie, jak i media z usług zewnętrznych
2. Maksymalna liczba nieudanych prób logowania wynosi 3
3. Video można przeglądać wyłącznie w aplikacji internetowej
4. Wersja zabezpieczona DRM-em, pobierana na urządzenie klienta



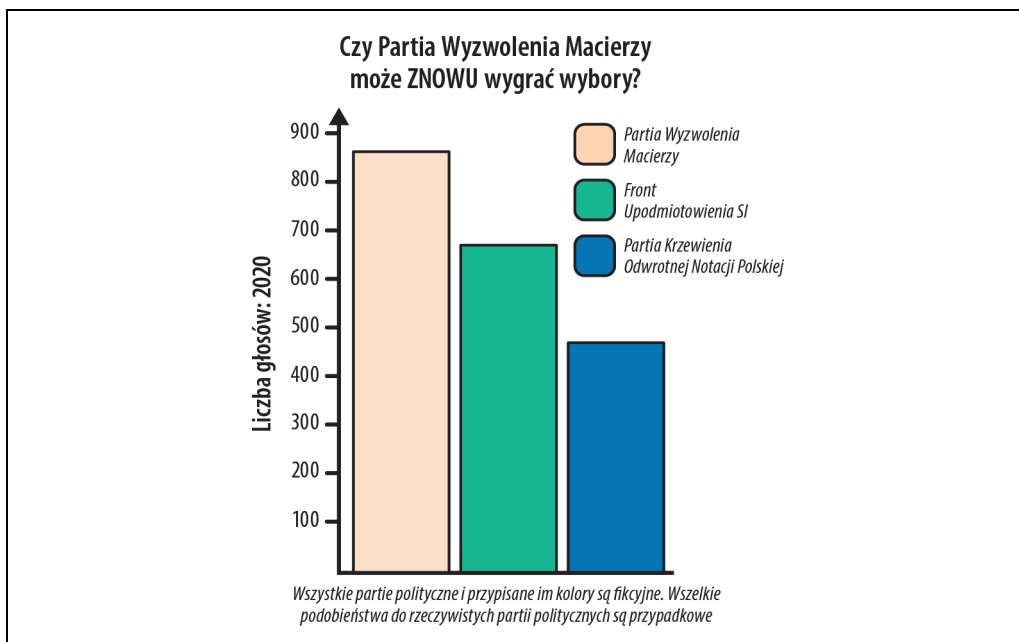
Rysunek 6.6. Drugi przykładowy fragment rysunku 6.4 z pozostawionym tytułem i notatkami



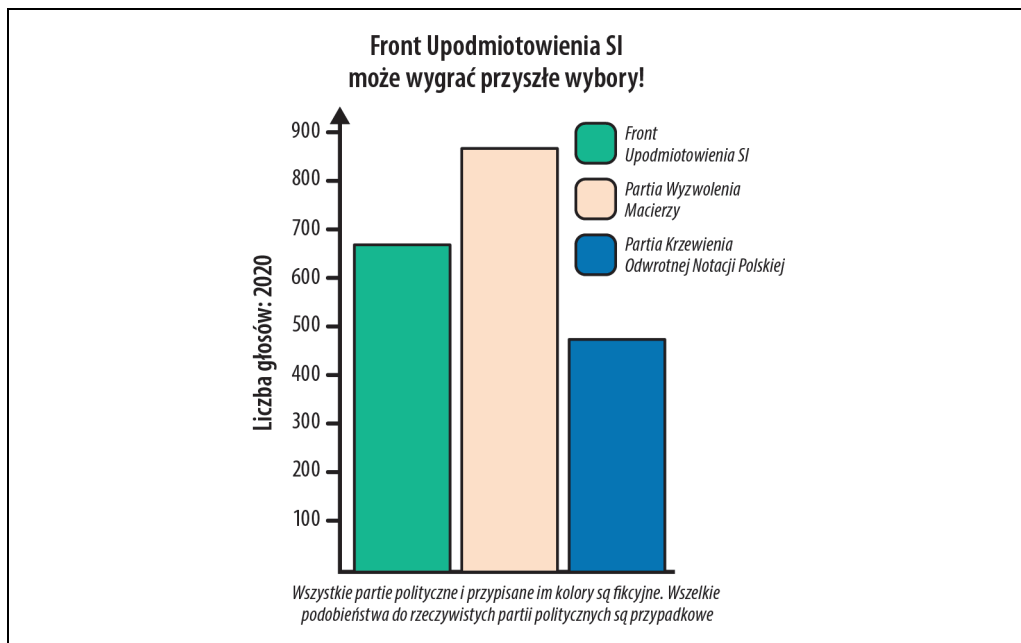
Rysunek 6.7. Porównanie stylów: szkicu i linii ciągłej



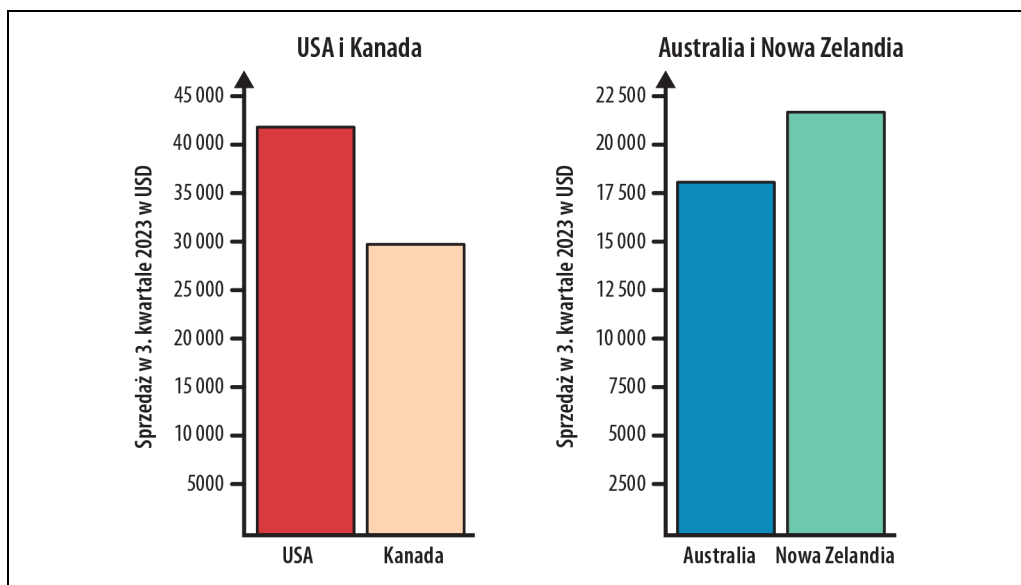
Rysunek 6.8. Wykres z linią bazową ustawioną na poziomie wartości 450 (anty wzorec)



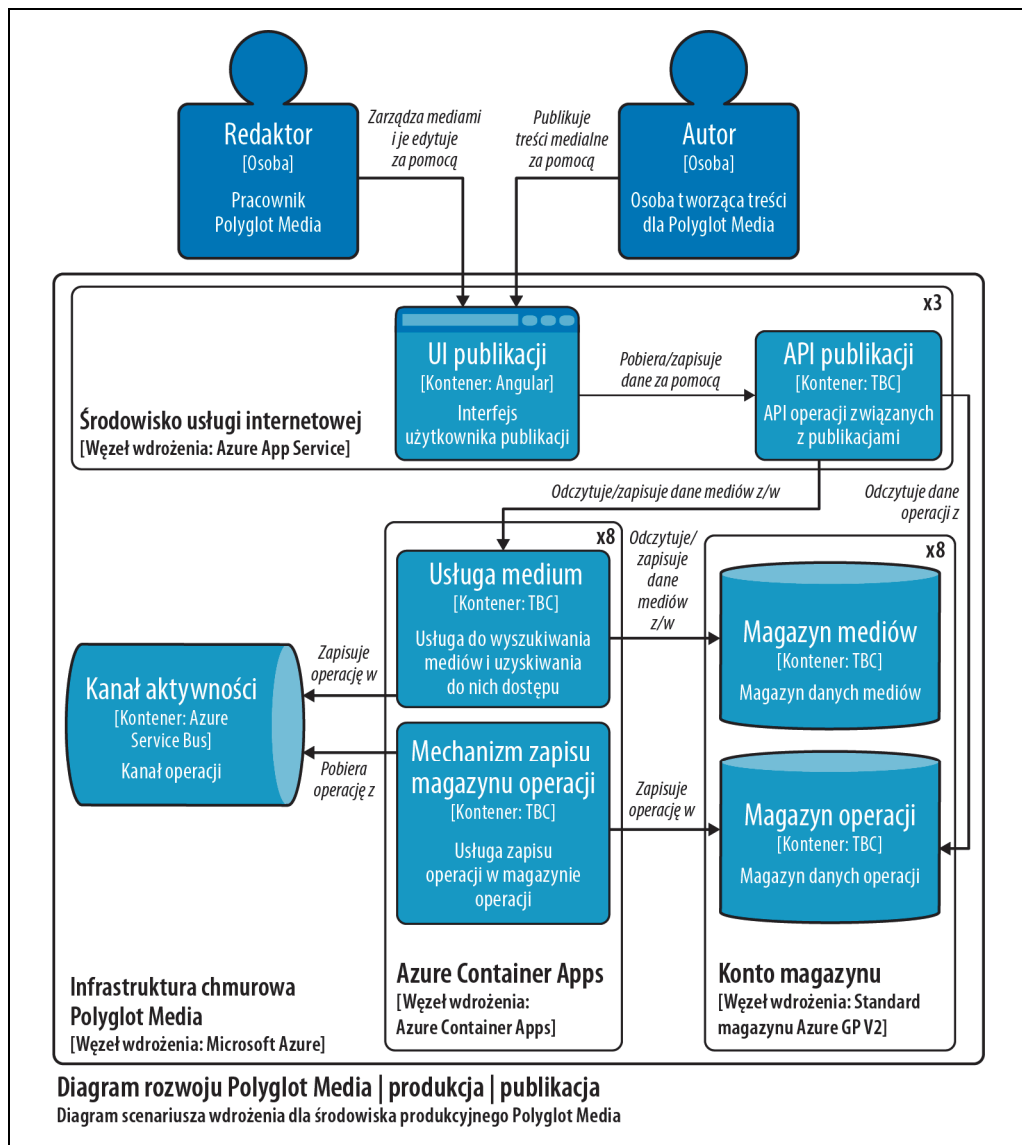
Rysunek 6.9. Wykres z linią bazową ustawioną na poziomie wartości 0



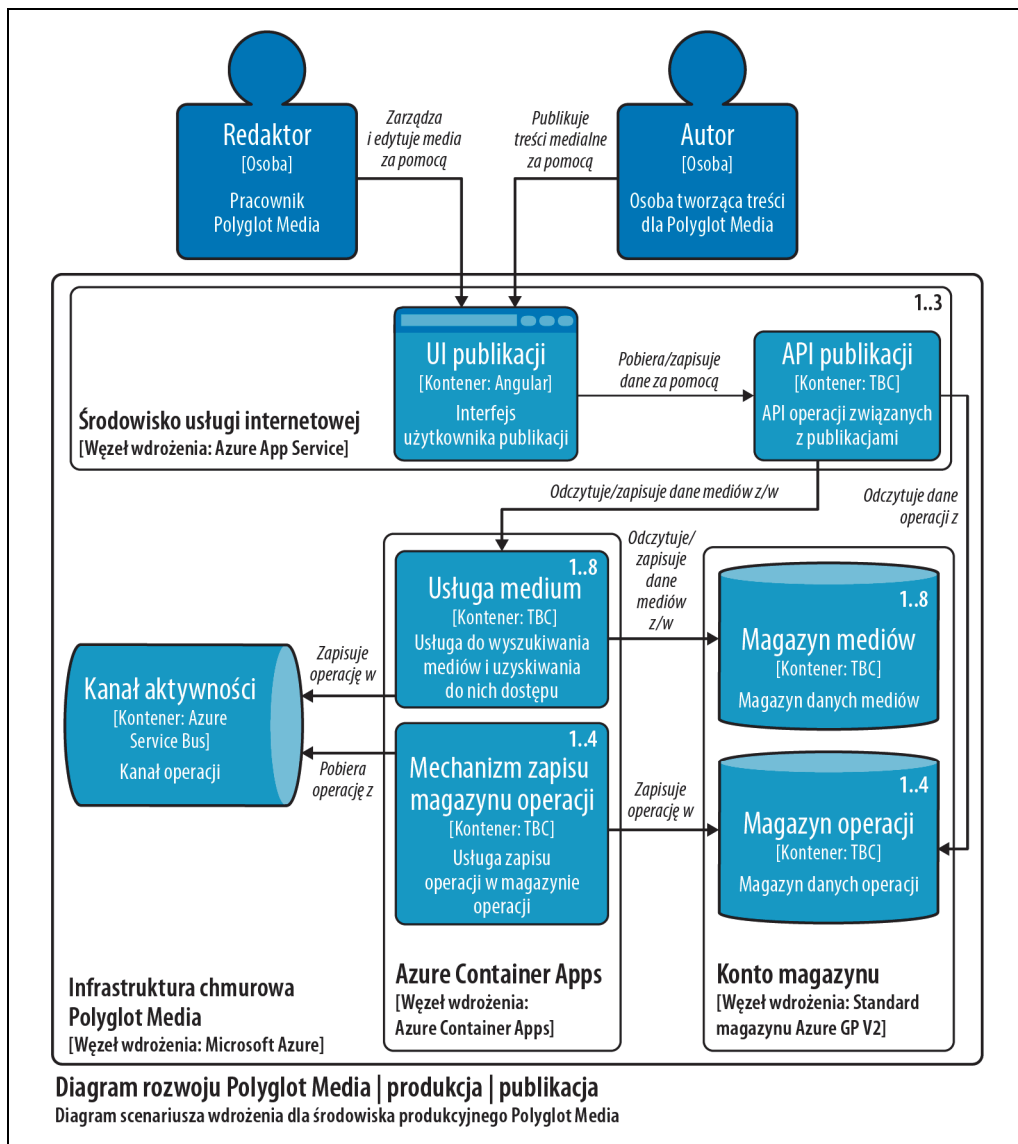
Rysunek 6.10. Wykres z tak samo ustawioną linią bazową i z tymi samymi wartościami co na rysunku 6.9, lecz zmodyfikowany tak, by wspierać przekaz innej partii



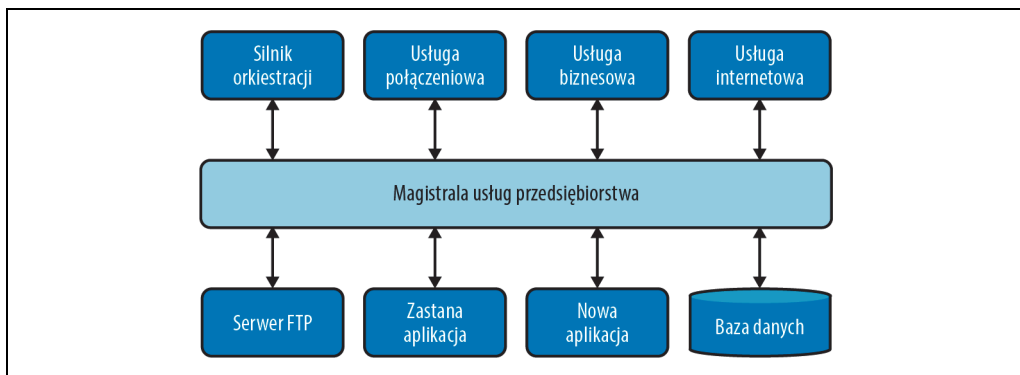
Rysunek 6.11. Wykresy pokazujące, jak porównanie może zmylić odbiorców (antywzorzec)



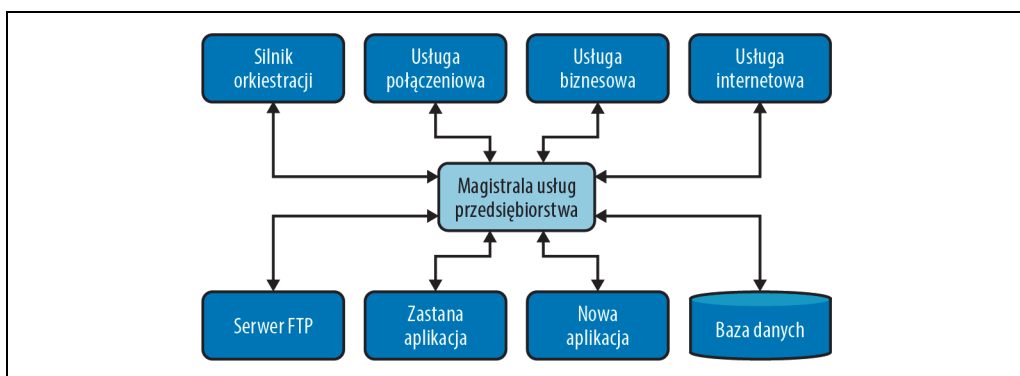
Rysunek 6.12. Mylący diagram wdrożenia



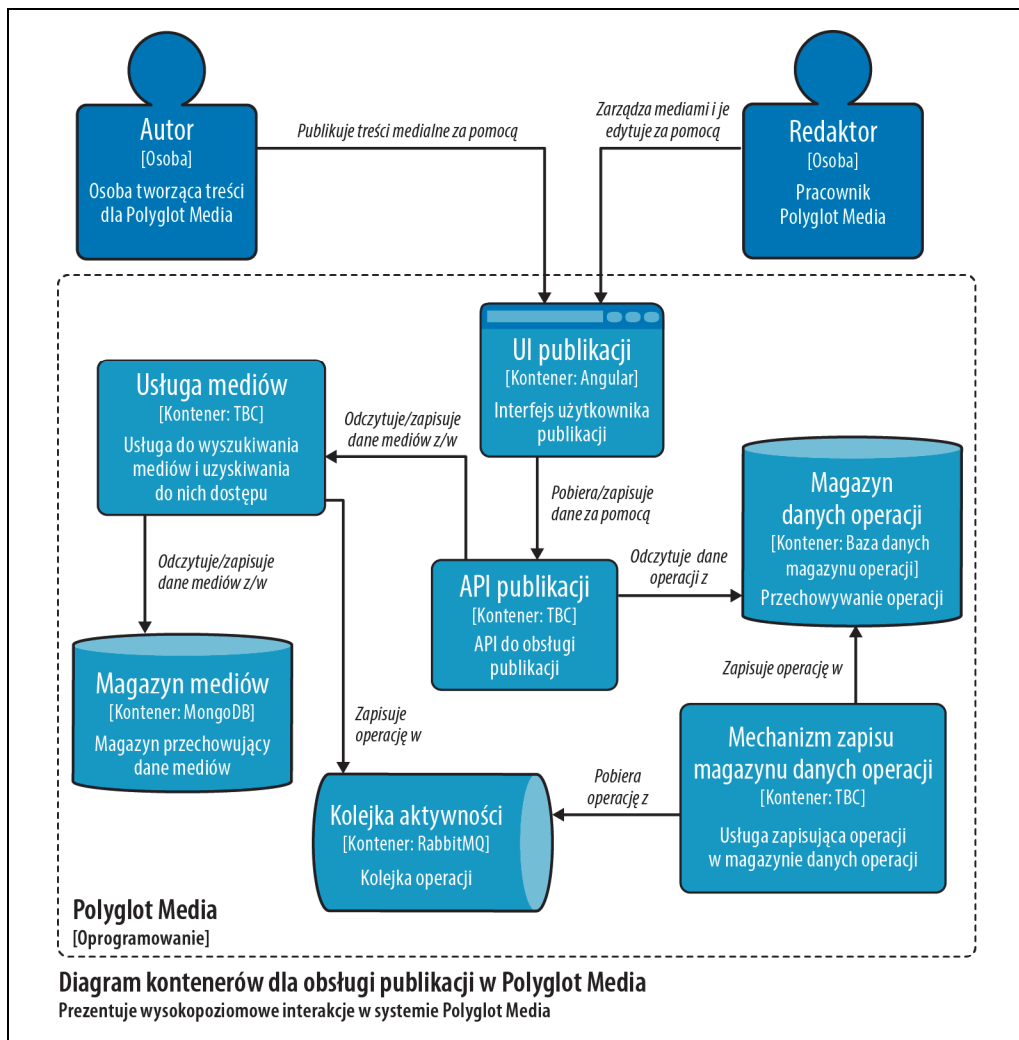
Rysunek 6.13. Prawdziwy diagram wdrożenia



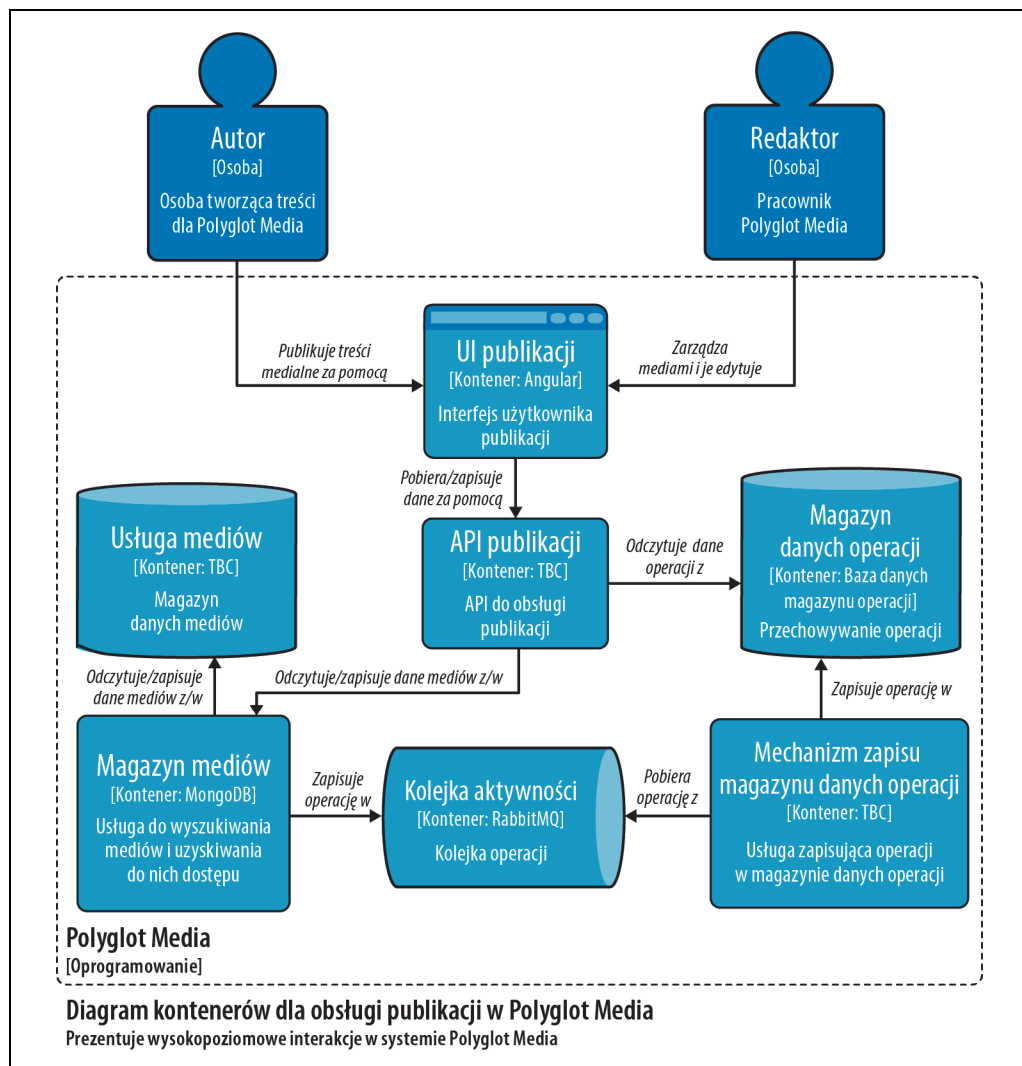
Rysunek 6.14. Diagram magistrali usług, który można z łatwością błędnie zinterpretować



Rysunek 6.15. Diagram magistrali usług bliższy stanowi faktycznemu

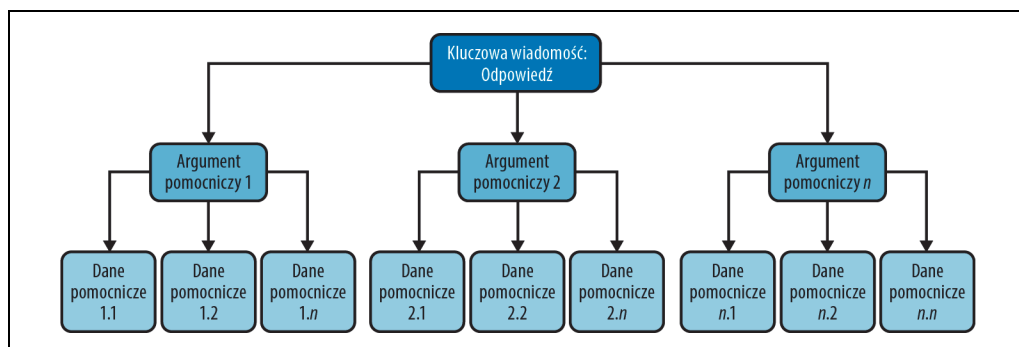


Rysunek 6.16. Niezrównoważony diagram kontenerów C4 (antywzorzec)



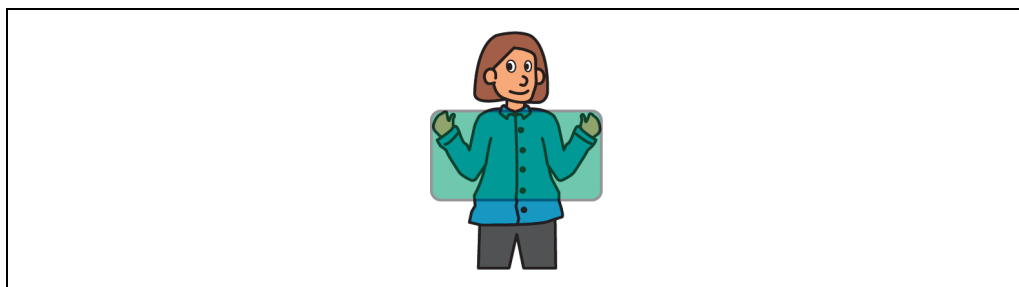
Rysunek 6.17. Zrównoważony diagram kontenerów C4

Komunikacja pisemna



Rysunek 7.1. Przykładowa struktura zasady piramidy Minto

Komunikacja werbalna i niewerbalna



Rysunek 8.1. Gesty powinny się mieścić w prostokątnej przestrzeni od górnej części klatki piersiowej do bioder w pionie i w odległości mniej więcej połowy szerokości ciała po obu stronach w poziomie



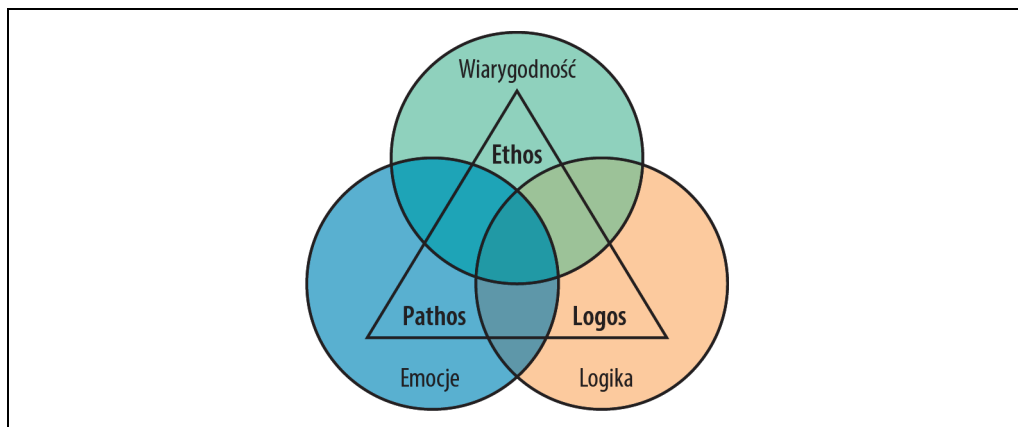
Intensywność, sukces, agresja

Mały, niewielki

Wyliczanie elementów listy

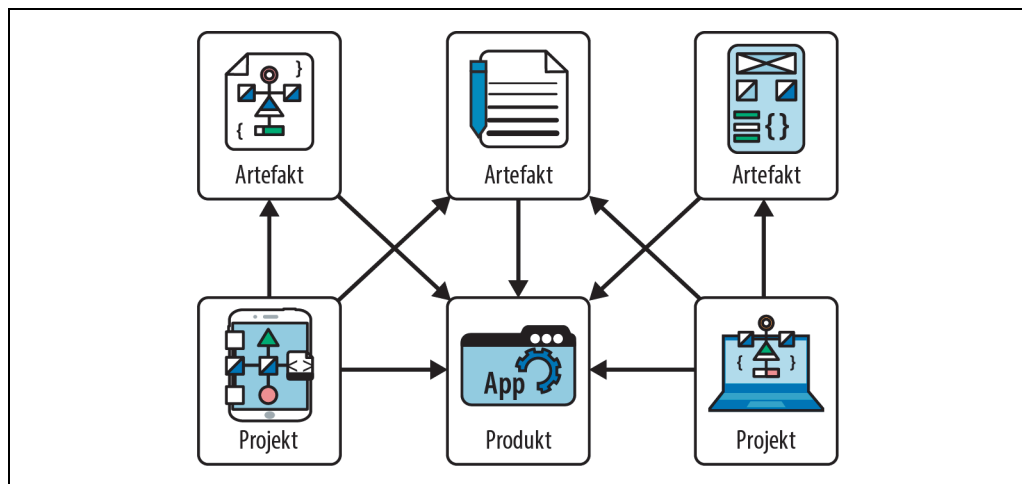
Rysunek 8.2. Przykładowe gesty dłonią

Trójkąt retoryczny



Rysunek 9.1. Trójkąt retoryczny Arystotelesa

Zasady zarządzania wiedzą



Rysunek 10.1. Artefakty wiedzy zorganizowane według produktów wraz z projektami, które się do nich odwołują

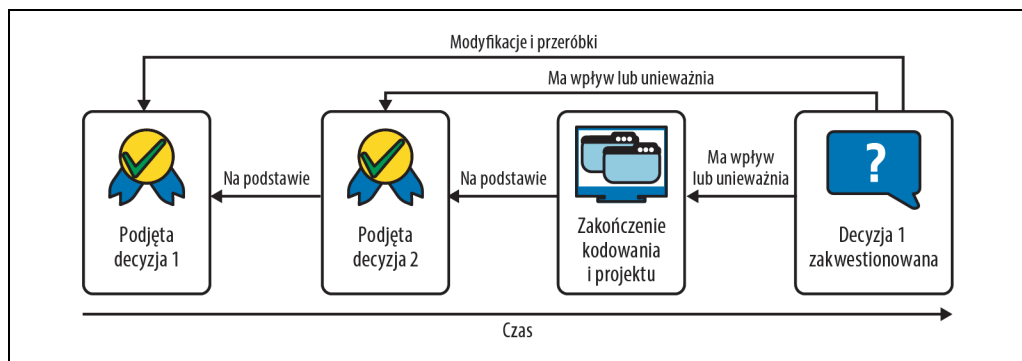


Rysunek 10.2. Ocena wyrażona w gwizdkach

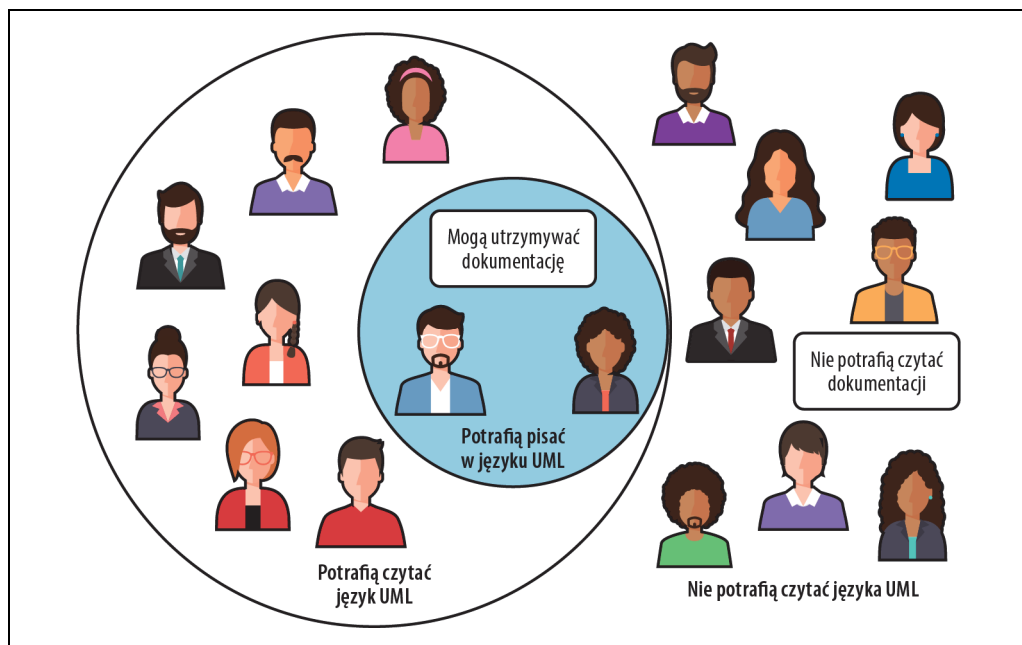


Rysunek 10.4. Chmura słów wygenerowana w serwisie <https://simplewordcloud.com>

Wiedza i ludzie

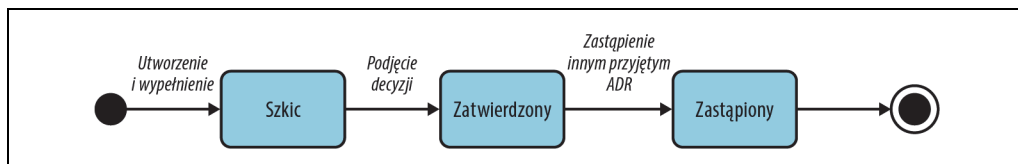


Rysunek 11.1. Brak gromadzenia informacji w momencie podejmowania decyzji 1. prowadzi do późniejszych kosztownych zmian

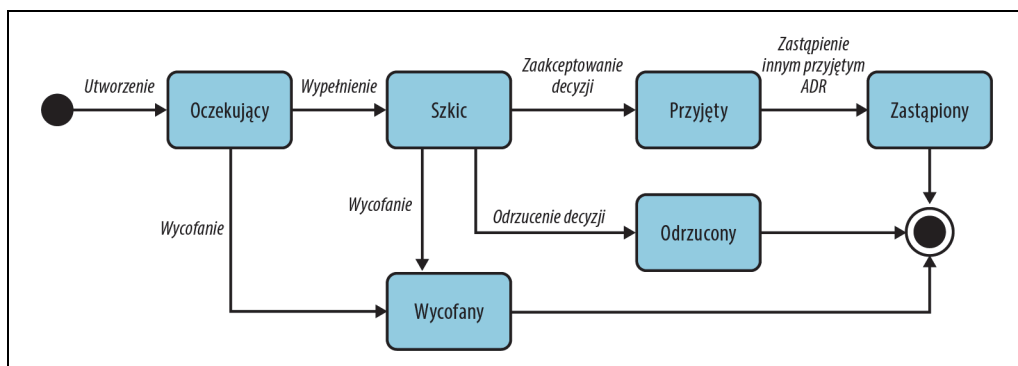


Rysunek 11.2. Wybór notacji lub formatu ma wpływ na to, kto może zrozumieć lub prowadzić dokumentację

Efektywne praktyki



Rysunek 12.1. Prosty (zalecany) zestaw statusów dokumentów ADR



Rysunek 12.2. Złożony (niezalecany) zestaw statusów ADR

ADR-044 Korzystanie z architektury rozproszonej sterowanej zdarzeniami

Status

Zatwierdzony, 04.10.2023

Zastępuje [ADR-031 Używanie funkcji bezserwerowych]

Kontekst

System Polyglot Media jest obecnie systemem rozproszonym, składającym się głównie z funkcji serverless. Przejście z architektury monolitycznej na rozproszoną architekturę serverless miało na celu rozwiązanie problemów z responsywnością systemu w trakcie działania i długim czasem oczekiwania na przygotowanie i wdrożenie w środowisku produkcyjnym zarówno nowych funkcjonalności, jak i poprawek.

Architektura serverless nie rozwiązała problemów z responsywnością w wymaganym stopniu ani nie zapewniła oczekiwanej łatwości utrzymania z powodu bardzo wysokiego poziomu powiązań pomiędzy poszczególnymi funkcjami.

Należy znaleźć rozwiązanie problemów z szybkością reakcji i czasem wprowadzania produktu na rynek.

Kryteria oceny

Zobacz [ADR-002 Wybór cech architektury].

- **Responsywność:** Klienci skarżyli się na problemy z responsywnością systemu i należy się tym zająć. Ta cecha jest uważana za najważniejsze kryterium.
- **Łatwość utrzymania:** Funkcje serverless skróciły czas wprowadzania na rynek poprawek błędów i nowych funkcjonalności, ale nie w wymaganym stopniu.
- **Łatwość wdrażania:** Oprócz łatwości konserwacji łatwość wdrażania jest ważna dla skrócenia czasu wprowadzania na rynek poprawek błędów i nowych funkcji.
- **Skalowalność:** Większość skarg klientów z zakresu czasu reakcji dotyczyła szczytowych okresów największego obciążenia, więc system musi obsługiwać maksymalną przewidywaną liczbę użytkowników bez wpływu na wydajność działania.

Opcje

1. Mikrousługi

Kryteria	Wynik	Uzasadnienie
Responsywność	★★★☆☆ 3/5	[Nie jest wydajny z natury], ale można wprowadzić optymalizacje w wąskich gardłach, np. poprzez skalowanie.
Łatwość utrzymania	★★★☆☆ 3/5	Zależności mogą stanowić problem, konieczność utrzymania wielu magazynów danych.
Łatwość wdrażania	★★★★★ 5/5	Wdrażane jest tylko to, co się zmieniło.
Skalowalność	★★★★★ 5/5	Można skalować wybrane usługi.
	Razem: 16/20	Inne kompromisy: <ul style="list-style-type: none">▪ Konieczność dzielenia danych na [jeden magazyn danych na usługę].▪ Koszty budowy są zazwyczaj wysokie.▪ Złożoność i trudność w tworzeniu przepływów pracy.

2. Rozwiązanie oparte na usługach

Kryteria	Wynik	Uzasadnienie
Responsywność	★★★☆☆ 3/5	[Nie jest wydajny z natury], ale można wprowadzić optymalizacje w wąskich gardłach, np. poprzez skalowanie.
Łatwość utrzymania	★★★☆☆ 4/5	Mniej magazynów danych do utrzymania niż w przypadku wykorzystania mikrousług.

Rysunek 12.3. ADR-044 Korzystanie z architektury rozproszonej sterowanej zdarzeniami, część 1

ADR-044 Zmiana na architekturę sterowaną zdarzeniami

Status

Zatwierdzony, 04.10.2023

Zastępuje [ADR-031 Używanie funkcji serverless]

Kontekst

System Polyglot Media jest obecnie systemem rozproszonym, składającym się głównie z funkcji serverless. Przejście z architektury monolitycznej na rozproszoną architekturę serverless miało na celu rozwiązanie problemów z responsywnością systemu w trakcie działania i długim czasem oczekiwania na przygotowanie i wdrożenie w środowisku produkcyjnym zarówno nowych funkcjonalności, jak i poprawek.

Architektura serverless nie rozwiązała problemów z responsywnością ani utrzymaniem w wymaganym stopniu z powodu bardzo wysokiego poziomu powiązań pomiędzy poszczególnymi funkcjami.

Należy znaleźć rozwiązanie problemów z szybkością reakcji i czasem wprowadzania produktu na rynek.

Decyzja

Wykorzystamy architekturę sterowaną zdarzeniami, bazując na jej wysokiej skalowalności i responsywności. Mikrousługi i rozwiązanie oparte na usługach nie spełniają najważniejszego kryterium — responsywności.

Skutki

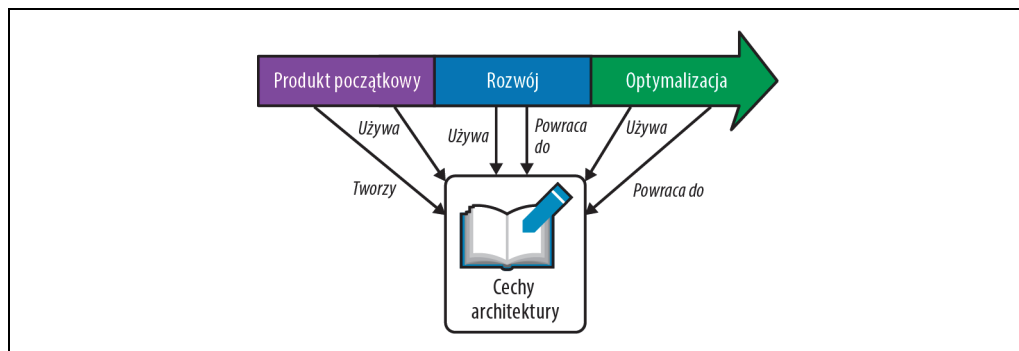
Pozytywy

- Zyskujemy ogólnie wyższą responsywność, łatwość konserwacji i skalowalność niż te, które zapewniają obecnie stosowane funkcje serverless.
- Przetwarzanie zdarzeń jest skalowalne poprzez uruchamianie dodatkowych instancji usług do przetwarzania zdarzeń w kolejce.
- Przetwarzanie w ramach usług jest skalowalne poprzez uruchamianie dodatkowych instancji danej usługi w celu zaspokojenia zapotrzebowania na wykonywane przez nią działania.

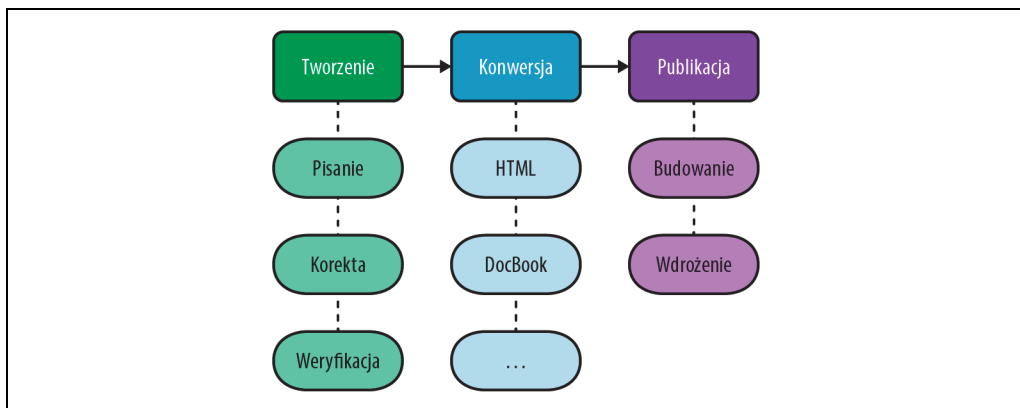
Negatywy

- Może wystąpić konieczność zdobycia przez zespoły lub osoby nowych umiejętności albo opanowania nowych technologii (takich jak zarządzanie kolejkami zdarzeń).
- Zarządzanie zdarzeniami (takimi jak kolejki) dodatkowo komplikuje proces programowania i wdrażania.
- Testy integracyjne i DevOps będą musiały zostać ponownie zweryfikowane.

Rysunek 12.5. ADR-044 Zmiana na architekturę sterowaną zdarzeniami

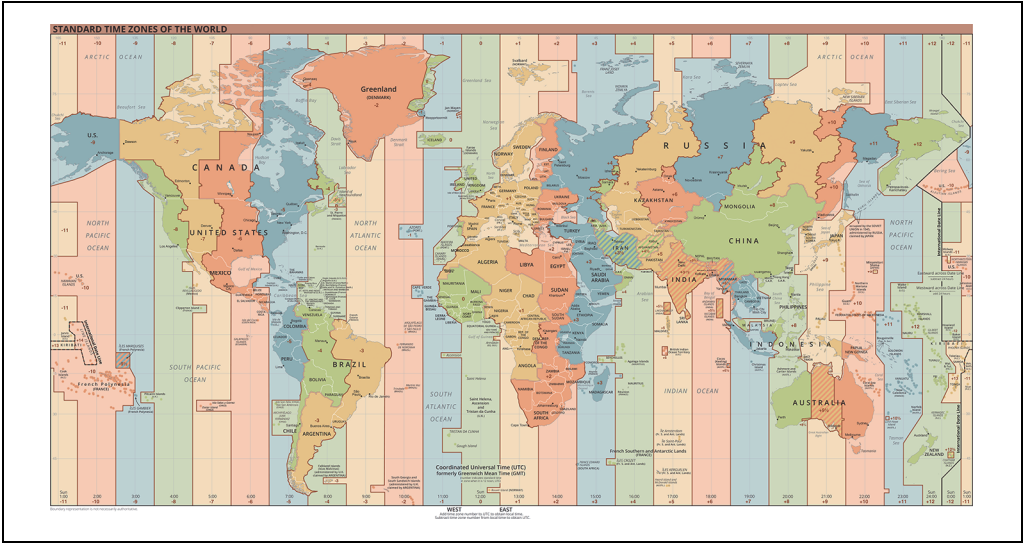


Rysunek 12.6. Korzystanie z cech architektury w miarę postępu cyklu życia produktu



Rysunek 12.7. Schemat koncepcji dokumentacji w formie kodu

Czas zdalny

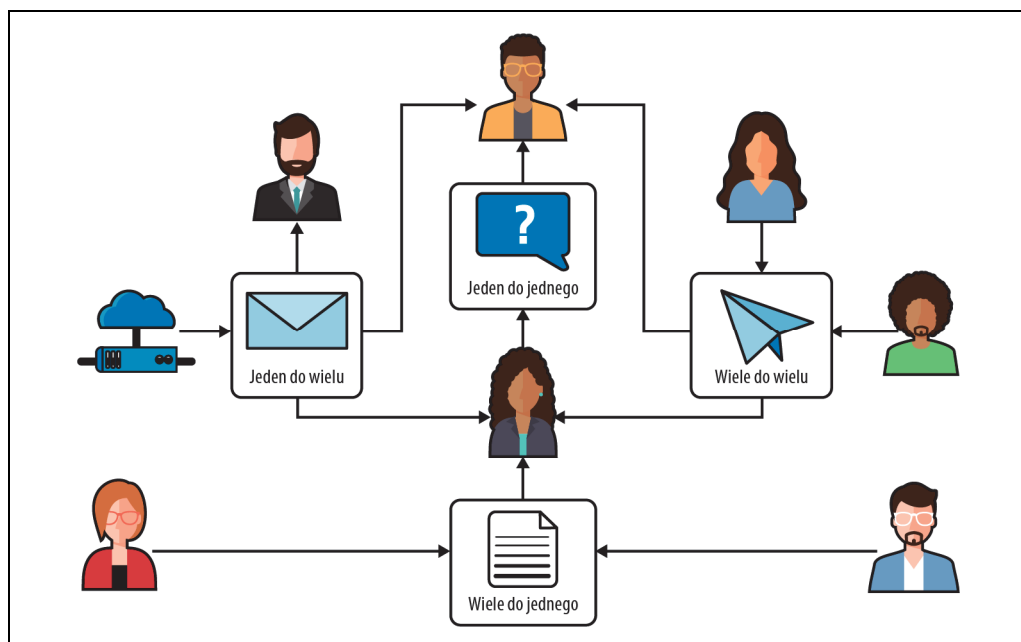


Rysunek 13.1. Mapa aktualnych stref czasowych. Źródło: Wikimedia
(https://commons.wikimedia.org/wiki/File:World_Time_Zones_Map.png)



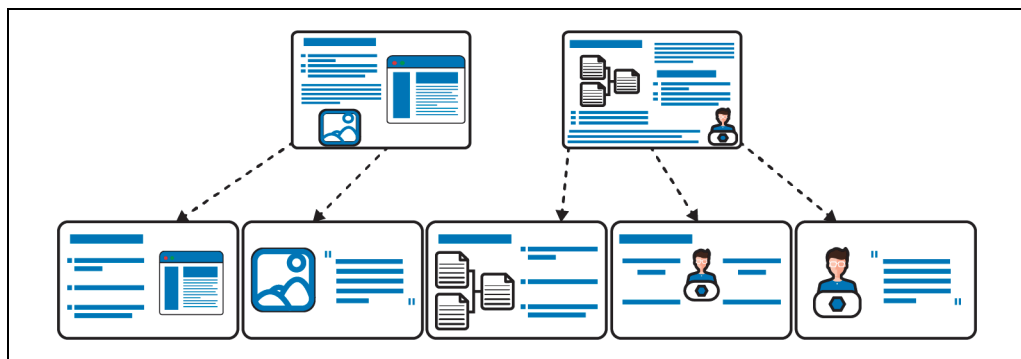
Rysunek 13.2. Względne różnice czasu między Londynem, Phoenix i Salt Lake City, kiedy w Londynie i Salt Lake City jest czas letni (World Time Buddy)

Zasady komunikacji zdalnej

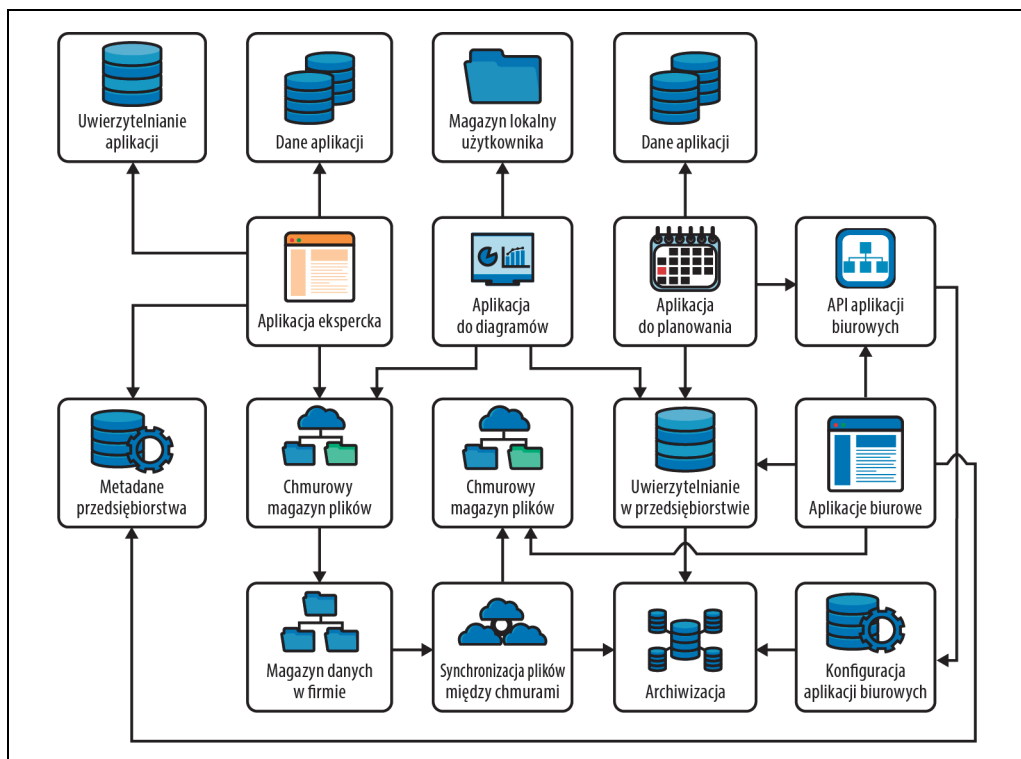


Rysunek 14.1. Kierunki komunikacji

Kanały komunikacji

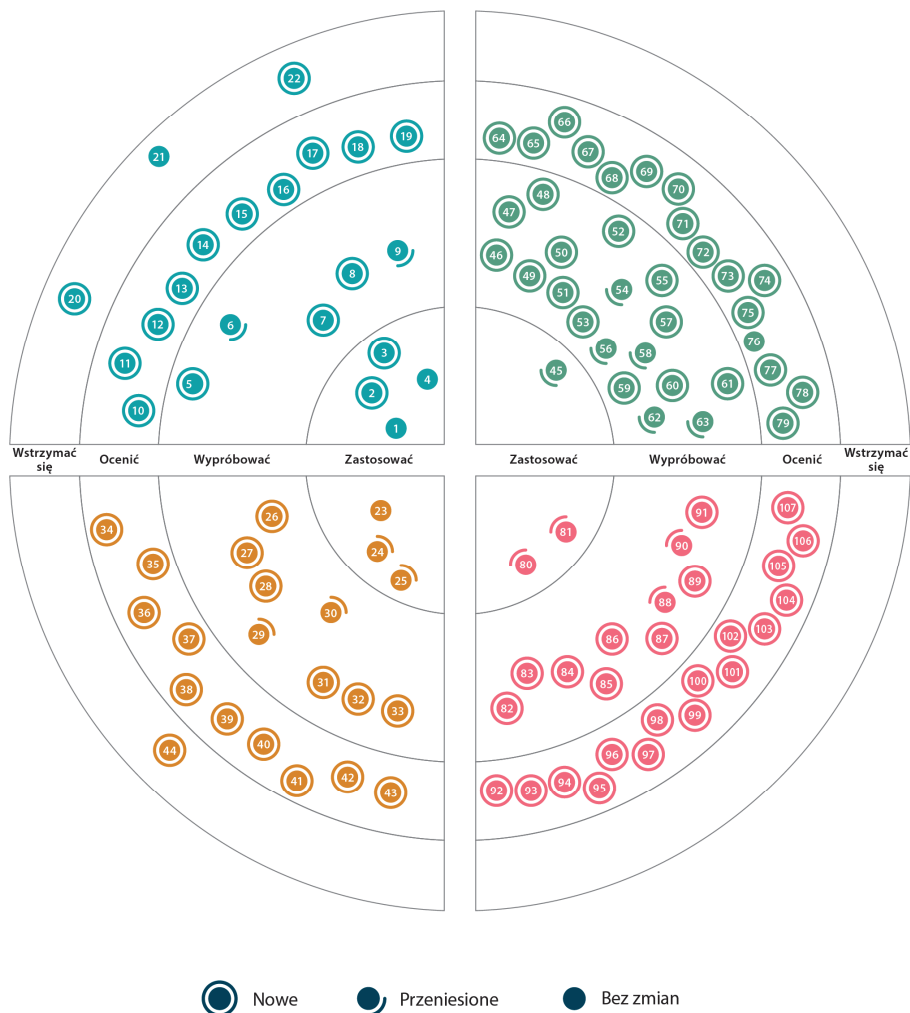


Rysunek 15.1. Ulepszanie prezentacji poprzez dzielenie slajdów



Rysunek 15.2. Rozprzestrzenianie i udostępnianie danych

Radar



Rysunek 15.3. Radar technologiczny firmy Thoughtworks (https://www.thoughtworks.com/content/dam/thoughtworks/documents/radar/2023/04/tr_technology_radar_vol_28_en.pdf)¹

¹ „Technology Radar”, Thoughtworks, tom 28, 2023 r.

Techniki

Zastosować

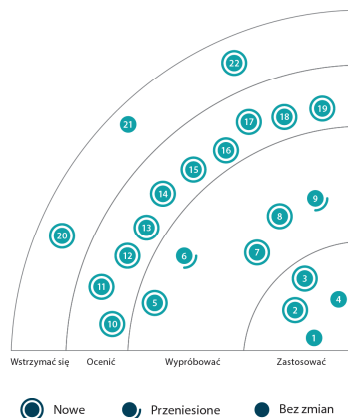
1. Zastosowanie zarządzania produktem w platformach wewnętrznych
2. Infrastruktura CI/CD jako usługa
3. Ograniczanie zależności
4. Koszt uruchomienia jako funkcja dopasowania architektury

Wypróbować

5. Adnotacje dotyczące dostępności w projektach
6. Ograniczone platformy o niewielkiej ilości kodu
7. Frontendy demonstracyjne dla produktów opartych wyłącznie na API
8. Architektura Lakehouse
9. Weryfikowalne dane uwiarygodniające

Oceń

10. Projektowanie testów komponentów z uwzględnieniem dostępności
11. Programowanie oparte na testach wspomagane przez sztuczną inteligencję
12. LLM specyficzne dla dziedziny
13. Inteligentne testy dostępności
14. Logseq jako baza wiedzy zespołu
15. Inżynieria poleceń
16. Analiza osiągalności podczas testowania infrastruktury



Rysunek 15.4. Ćwiartka technik z radaru technologicznego Thoughtworks (https://www.thoughtworks.com/content/dam/thoughtworks/documents/radar/2023/04/tr_technology_radar_vol_28_en.pdf)²

² Radar technologiczny formy Thoughtworks.