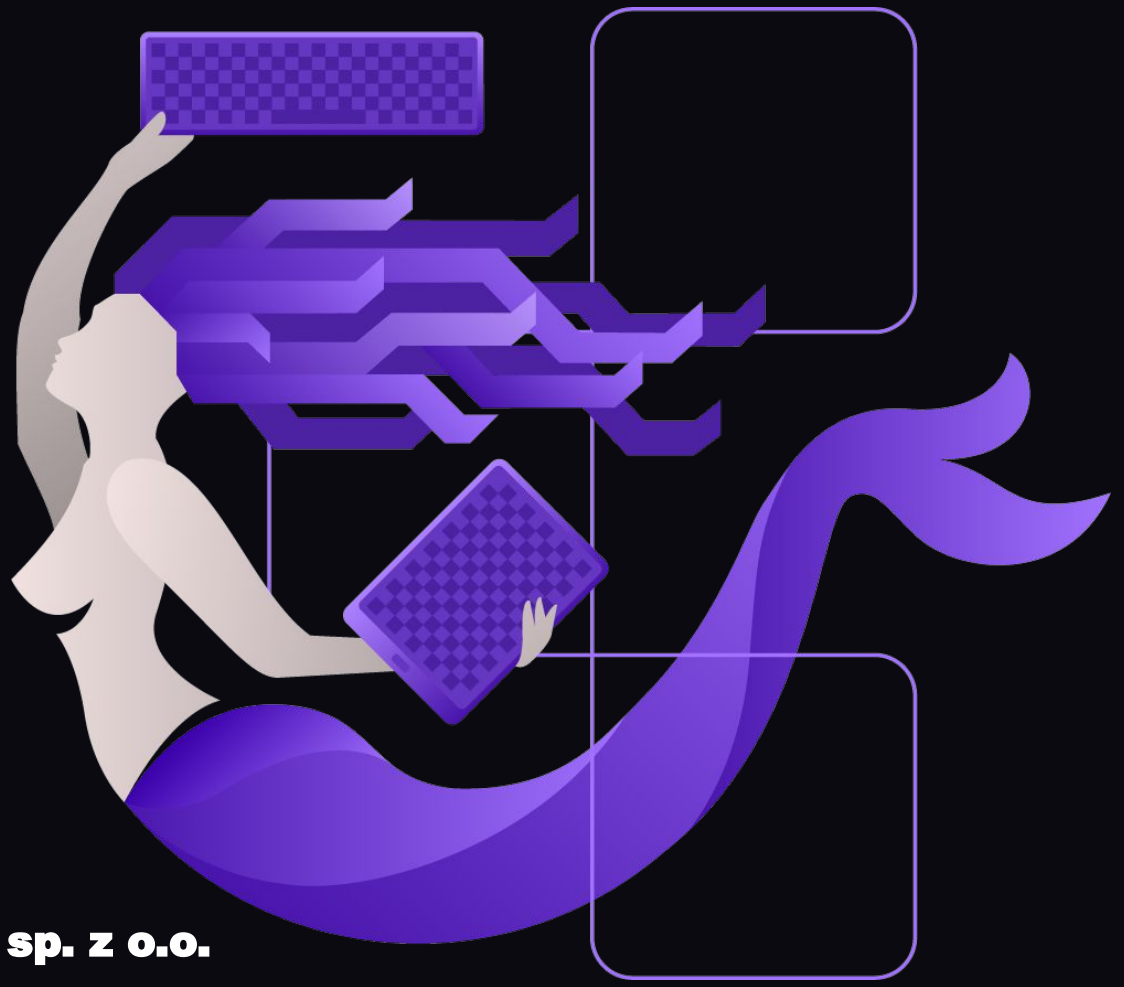
WDI WARSZAWSKIE DNI INFORMATYKI



Po kiego kopiujesz te dane? Dlaczego nie zawsze potrzebujesz Data Lake/Warehouse. Rzecz o Trino i Data Mesh.

Olaf Górski aka @grski
najbardziej przepłacany junior w Wwa @OnionMindset sp. z o.o.

ORGANIZATOR GŁÓWNY: Academic Partners FUNDACJA

KOMITET ORGANIZACYJNY: kilkadziesiąt organizacji z sektora IT / data science (pełna lista na stronie wydarzenia)

Dwa słowa o mnie

Olaf Górski aka @grski
najbardziej przepłacany junior w Wwa @OnionMindset sp. z o.o. (tak, to serio nazwa mojej spółki)

autor https://juniorpythondeveloper.pl/ i mityczny #programista40k

do tego 23-letni młodzieniaszek bez matury z ponad 5. letnim expem na koncie, przygodami z własna firma, mentor i takie tam

https://grski.pl/
https://github.com/grski
https://www.linkedin.com/in/olafgorski/
olafgorski@pm.me

Nowa waluta

Żyjemy obecnie w czasach gdy stwierdzenie "Czas to pieniądz" nie traci swojego sensu, bezapelacyjnie.

Mało tego, moim zdaniem należy je nieco rozszerzyć o nową walutę. Obecnie zmierzamy, lub już jesteśmy w czasach, kiedy pieniądz fiducjarny traci na wartości, zastępuje go coś innego, coś innego staje się bardzo ważne i krytyczne. Wszelkiego rodzaju dane.

Trzeba to zrozumieć. W danych, zwłaszcza w dużej skali, drzemie dość spora moc, jeśli odpowiednio ich użyć.

Manipulacje przy wyborach, kształtowanie opinii publicznej, uczenie modeli pokroju GPT.

Opóźniona reakcja większości

Do niedawna firmy nie zdawały sobie sprawy z tego faktu. Brakowało narzędzi, sprzętu i innych rzeczy, by efektywnie analizować i wyciągać wnioski z ogromnych ilości danych w taki sposób, by miało to sens i było opłacalne. Czasy się jednak zmieniają.

W tym wypadku zmiana nastąpiła już lata temu, co w IT jest tak naprawdę całą wiecznością, gdyż tutaj wszelakie zmiany propagowane są bardzo szybko. Nowych dziedzin, technologii i wszystkiego innego jest na tyle dużo, że ciężko za tym wszystkim nadążyć. Dlatego też do wielu firm i ludzi pewne rzeczy jeszcze nie dotarły, powodując, że tkwią oni w zabytkowych wzorcach.

Jedną z takich rzeczy, jak dla mnie, jest Data Lake i cały proces związany z centralnym magazynowaniem danych. O tym może jednak później, dla osób mniej wtajemniczonych, zrobię małe intro.

Dane w każdej firmie

Każda firma, każdy produkt, posiada jakieś dane. Swoje, o swoich usługach, dane klientów. Whateva. Tradycyjnie i w uproszczeniu trzymamy je w bazach danych.

Suprprise, surprise. Sprawa jest dość prosta, kiedy mamy jedną malutką bazę, jeden produkt, jeden zespół, małą firmę i tak dalej. Tutaj wszystko ładnie nam się skleja, mamy pod ręką, **jest elegancko**, **fajnie fajniusio.**

No właśnie nie jest.
Życie to nie bajka, a życiowy parkiet bywa śliski, zatem uważaj jak tańczysz młody developerze.

Nie ma tak dobrze

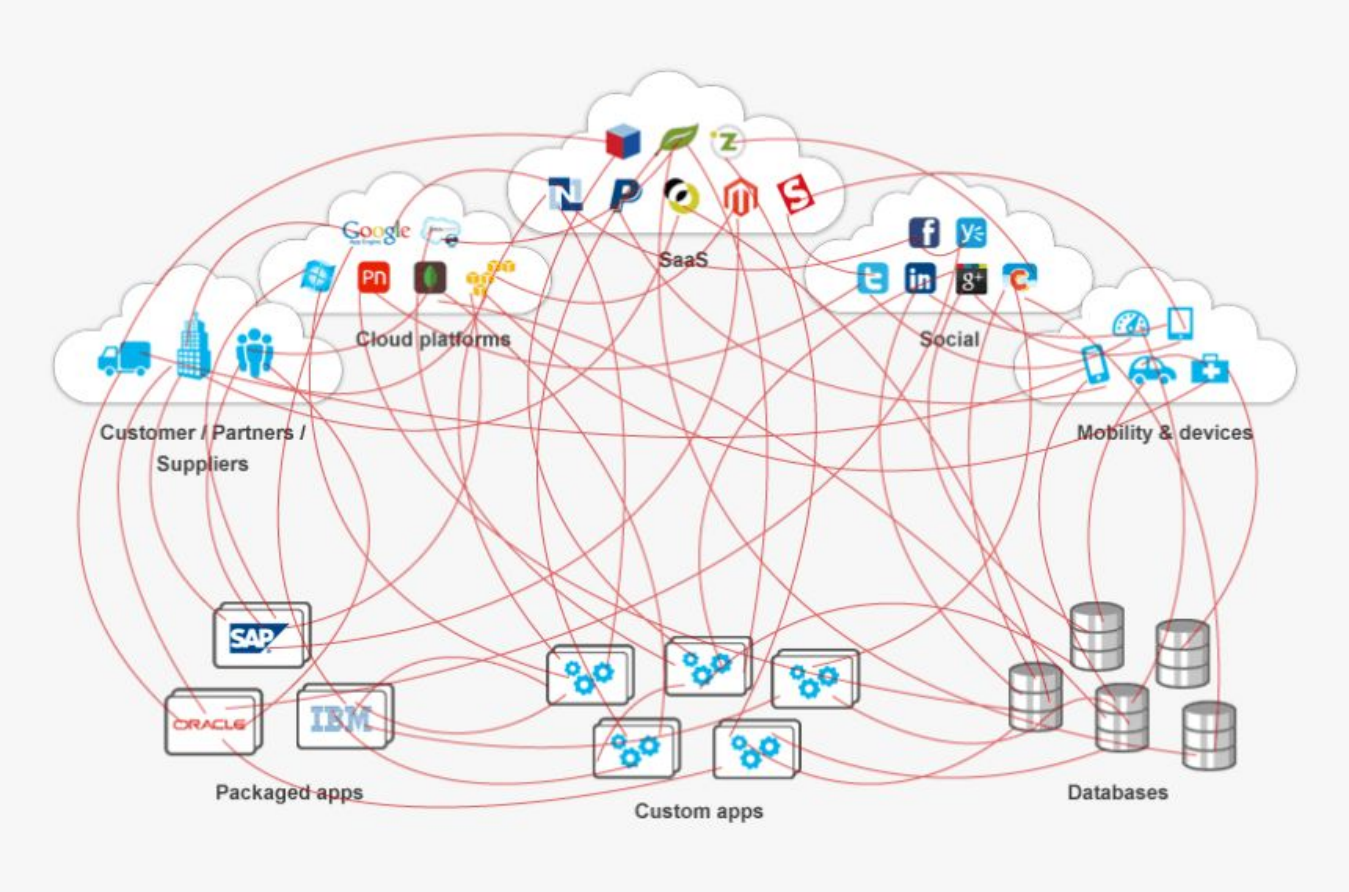
Natomiast takie scenariusze nie są zbyt częste. Podczas rozwoju firmy, aplikacji, produktu, prędzej czy później rozrasta nam się warstwa danych. Zaczynają dochodzić nowe aplikacje, część z nich to nasze dzieło, za część odpowiadają zewnętrzni dostawcy, część to SaaSy, pojawiają się nowe bazy danych, nowe języki, nowe technologie. W końcu żyjemy w czasach gdzie mikroserwisy są takie modne, więc niczym dziwnym jest posiadanie 5 skrajnie różnych technologii w projekcie, które jakoś ze sobą muszą gadać.

Tutaj rzeczy zaczynają się komplikować. Bo jak przeanalizować takie dane, których kawałek mam na przykład w mongo, z którego korzysta backend aplikacji mobilnej, część w postgresie, gdzie jakiś mikroserwis wrzuca dane, a inną cześć klient dosyła w csvkach co jakiś czas.

Dodatkowo każde z tych źródeł danych ma nieco inne standardy, nomenklatury. Część jest dodatkowo mega wolna bo stoi gdzieś na serwerze w piwnicy u klienta. Wszystko spięte trytytkami i taśmą klejącą, ale jakoś działa. Natomiast później dochodzą jeszcze dane z API do nowego vendora potrzebne do wzbogacenia naszych danych. Tam jeszcze jakiś zabytkowy mysql się ostał a są w nim dane statystyczne, których nam potrzeba. Wtedy jest źle.

Nie ma tak dobrze

Brzmi jak spaghetti? I słusznie, niestety taka rzeczywistość i to wcale nierzadka, z mojego doświadczenia wynika, że życie to nie bajka i trzeba uważać jak się tańczy bo software bywa śliski. Czy nam się to podoba, czy nie. Jakiekolwiek przetwarzanie danych w tak powstałym systemie bywa trudne. Wszystko oddzielnie, rozsiane w różnych formatach, nie skalowalne i w ogóle jakieś takie be.



Zamknięte silosy

Dodatkowo to powoduje inne problemy. Każdy zespół odpowiada za swój kawałek danych. Zespół A często może nie wiedzieć bo jest w danych zespołu B. To rodzi problemy i nieefektywne wykorzystanie dostępnych zasobów. Tak zwane Silosy. Co jeśli oddzielnie dane zespołu A i B mają wartość 2 i 2, ale jedne w połączeniu z drugimi umożliwiają dokonanie czegoś zupełnie nowego i zamiast sumy mamy tutaj iloczyn lub wykładniczość nawet? Wartość ich sumy czasami taka bywa.

W przypadku braku zastosowania jakiejś sensownej starategi często jest tak, że powstają silosy, który ze sobą nie rozmawiają z niejasnym podziałem obowiązków/obszarów. Nie jest oczywistym kto odpowiada za co, albo gdzie czego szukać.

A zrobić to w sposób wydajny i bezproblemowy, z pomocą tradycyjnych rozwiązań, to już w ogóle sztuka. Do niedawna świat Big Data oferował nam nowe rozwiązanie na wszystko. Lek na raka.

Zamknięte silosy

Dodatkowo to powoduje inne problemy. Każdy zespół odpowiada za swój kawałek danych. Zespół A często może nie wiedzieć bo jest w danych zespołu B. To rodzi problemy i nieefektywne wykorzystanie dostępnych zasobów. Tak zwane Silosy. Co jeśli oddzielnie dane zespołu A i B mają wartość 2 i 2, ale jedne w połączeniu z drugimi umożliwiają dokonanie czegoś zupełnie nowego i zamiast sumy mamy tutaj iloczyn lub wykładniczość nawet? Wartość ich sumy czasami taka bywa.

W przypadku braku zastosowania jakiejś sensownej starategi często jest tak, że powstają silosy, który ze sobą nie rozmawiają z niejasnym podziałem obowiązków/obszarów. Nie jest oczywistym kto odpowiada za co, albo gdzie czego szukać.

A zrobić to w sposób wydajny i bezproblemowy, z pomocą tradycyjnych rozwiązań, to już w ogóle sztuka. Do niedawna świat Big Data oferował nam nowe rozwiązanie na wszystko. Lek na raka.

O tym całym jeziorze

Data Lake

Data Lake. O co tutaj chodzi? To taki koncept centralnego miejsca gdzie przechowujemy dane we wszelakich formatach i stanach o różnych źródłach. Jest to coś, co pozwala nam pokonać jeden z problemów jaki mieliśmy wcześniej w tej naszej Zupie z Danych.

Część z nich była tu, cześć gdzie indziej, bałagan. To powoduje komplikacje gdy chcemy te dane jakoś razem przetwarzać w efektywny sposób. Data Lake z tym pomaga.

Teraz, zamiast trzymać nasze dane a tu w tym mongo, a tu postgresie, a to na tym leciwym serwerku z csvkami klienta, kopiujemy je wszystkie w cholerę do naszego Data Lake. Dzięki temu są chociaż 'blisko siebie', nawet jeśli w różnych formatach czy stanach. To nic, bo znaczny krok już poczyniony.

Dodatkowo jak już je mamy u 'siebie' i na własnych warunkach, to można trochę popoprawiać przy okazji kopiowania. A to zaś z kolei zahacza o coś zwanego ETL -> Extract Transform Load, ale o tym w szczególe mówić nie będziemy, natomiast warto znać ten akronim. W każdym razie.

Jezioro czy bajoro

No własnie. Pytanie się pojawia. Jesteśmy krok dalej, bo mamy rzeczy chociaż plus minus w jednym miejscu, może nawet w miare skalowalnie to wszystko zrobione a i silosy danych wyeliminowane! Całkiem zacnie, prawda? Niby tak, ale nie do końca.

Problem pojawia się, kiedy tych danych mamy dużo, kiedy dużo jest źródeł danych. Ogarnięcie całego tego overheadu zaczyna być skomplikowane.

Jakby zarządzanie wieloma źródłami, formatami i typami danych nie było samo w sobie problemem. Dodatkowo jak możesz się domyślić, posiadanie wszystkiego w jednym miejscu niesie ze sobą pewne problemy jak i koszta. Z czasem Data Lake staje się ogromem nie do ogarnięcia. Staje się bajorem.

Jezioro czy bajoro

Znam projekty, gdzie to podejście doprowadziło do ślepego zaułka.

Wiadomo, w dużej mierze była to wina osób implementujących aniżeli samej strategii, ale wciąż. Bajoro oprócz centralizacji ma też jeszcze jedną wadę, o której często nie myśli się w przypadku aplikacji o małej skali. Koszt i czas przesyłu danych. Mianowicie wyobraźcie sobie, że do naszego Data Lake trzeba kilka TB danych wrzucić. W obecnych czasach to nie jest jakoś dużo szczerze mówiąc. Nagle robi się problem. Nagle okazuje się, że wysyłamy ciężarówki z fizycznymi dyskami, które przekopiuja dane z serwerów klienta i przywiozą je spowrotem bo będzie taniej i szybciej. Nie, nie robię sobie żartów. AWS oferuje nawet takie specjalne ciężarówki do tego - Snowmobile xD

Obecnie ta usługa dostępna jest chyba jedynie dla klientów o bardzo dużej skali, nie zaś o TB skali, natomiast kiedyś było kapkę inaczej. Point being - kopiowanie danych dostarcza kłopotów i kosztów.

Kiedy Tech Lead kazał stażyście przynieść wiadro danych z internetem a ten się zagalopował i zbudował z tego produkt.

Jezioro czy bajoro

I dostarcza kolejnego problemu - po skopiowaniu danych trzeba przecież dbać o ich aktualizację.

Jeśli to dane archiwalne, to pal sześć, ale co jeśli to dane w miarę aktualne, które są uaktualniane? Update w Data Lake to czasami kosztowna rzecz. Z racji centralizacji Bajoro potrafi być też nieco wolne, czasami. Wolne to w sumie złe słowo, bo Bajora są szybkie.

Relatywnie wolne, albo wolniejsze od innych rozwiązań. Jakich konkretnie? Bo tak paplam i paplam, narzekam, a jaka jest alternatywa?

Trino i Data Mesh

Data Mesh to coś innego niż nasz Data Lake. Bajorko, to, w ramach przypomnienia, centralne miejsce zbioru danych. Data Mesh zaś to bardziej strategia, mindset i nastawienie, swego rodzaju strategia i dizajn systemu.

Jedną z rzeczy, które czasami będą wynikać z Data Mesh jest to, że prawdopodobnie nie będziemy mieć jednego dużego Data Lake centralnego dla całej firmy i będącego całym wszechświatem.

Zamiast tego będziemy mieli kolaborujące ze sobą mini jeziorka, stawiki, które mają między sobą gęstą sieć połączeń, dzięki której przepływ danych/wody między nimi jest efektywny i oszczędny.

Wyobraźmy sobie dla porównania Data Lake jako ogromne miasto z upakowanymi ciasno setkami tysięcy mięszkańców a Data Mesh bardziej jak dobrze skomunikowana ze sobą sieć nieco mniejszych miasteczek, rozlana aglomeracja.

Data Mesh

Data Mesh nie jest technologią samo w sobie. Nie wyklucza się też z Data Lake, powiedziałbym nawet, że są to komplementarne koncepty.

Natomiast na poziomie organizacji to Data Mesh jako strategia FTW. Nie hejtuję tutaj ani jednego ani drugiego rozwiązania. Każde z nich ma swoje oddzielne zastosowania, wady i zalety. Niech będzie to jasne.

A wtem wchodzi Trino całe na biało

Rzecz o Trino

Mamy porównanie jeden centralny samorząd w ogromnym mieście i kilka dobrze skomunikowanych ze sobą miasteczek z mniejszymi organami.

Centralizacja czasami działa, zawsze jednak do pewnej skali. Tak samo jak w urzędzie, jeśli wszystko mamy scentralizowane, to dojdziemy do punktu, gdzie co prawda może i wszystko załatwia się w jednym okienku, ale uzyskanie najprostszego pozwolenia zajmuje wieki, proces jest długi, drabina odpowiedzialności ogromna. Podobnie z Data Lake.

Data Mesh nieco zapobiega temu konceptowi i powoduje, że odpowiedzialność za rzeczy jest bardziej wyraźna, jasno wytyczona, struktura mniejsza. Tylko jak to konkretnie zrobić? Tu z pomocą przychodzi właśnie Trino. Trino to rozproszony system kwerend/zapytań, który idealnie wpasowuje się w ideę Data Mesh.

Rzecz o Trino

W Trino zamiast kopiować dane do centralnego magazynu, przetwarzamy je z pomocą 'technologii' w której są przechowywane.

Silnik Trino potrafi rozmawiać z wieloma bazami danych, potrafi je odpytywać, wyciągać z nich tylko to, co jest potrzebne i dopiero na tej podstawie dokonywać analizy.

Czyli zamiast kopiować dane do centrali i mielić przez nie wszystkie w jednym miejscu, jesteśmy w stanie wysłać prośbę o 5 różnych rzeczy do 5 różnych baz, każda z nich mieli to jednocześnie a dopiero potem przetwarzamy wyniki w silniku.

Dodatkowo Trino zapewnia unifikację sposobu interakcji i wiele innych rzeczy. W takim ogromnym skrócie możemy powiedzieć, że Trino pozwala nam na to, by odpytywać prawie dowolne dane tak, jakby to była jakaś SQLowa baza danych.

Dodatkowo mimo tego, że te źródła danych potrafią być kompletnie różne np. Mongo, mysql, postgres, rest api, csv, parquet, orc na s3, Trino potrafi zrobić tak, byśmy my z poziomu użyszkodnika mogli traktować to wszystko jako jedną bazę z różnymi schematami!

Trino i amazing ficzery

W Trino (czy jego komercyjnych dystrybucjach jak np. Starburst) praktycznie za darmo dostajemy pewne rzeczy, które “fizjonomom” normalnie się nie śniły.

Skalowalność rzędu Tera Czy Peta bajtów lub tysięcy maszyn.

Integracja z wszelakimi menadżerami dostępów, permissionów. Row-level/Column-level permissions.

Chcesz by power user z europy mógł przetwarzać tylko dane europejskich klientów bo GDPR? Nie ma problemu.

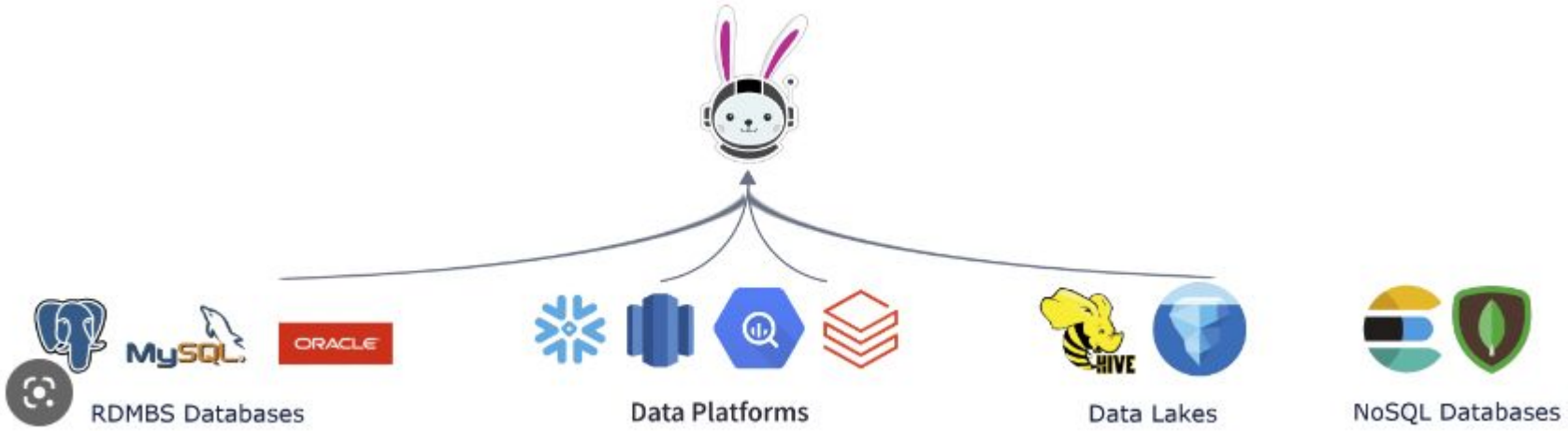
Co z przetwarzaniem danych kiedy np. masz firmę w US, ale dane twoich klientów nie mogą wyjść poza określony kraj? Starburst StarGate

Chcesz się podpiąć pod inne toole? Napisz własny connector albo skorzystaj z tych dostarczonych przez społeczność i inne firmy!

zytamy z CSVki gdzieś na ftpie jakby to był SQL? ALEŻ PROSZĘ. Z REST API? TO SAMO. MOŻE XML? NIE MA PROBLEMU.

Trino*

Przykładowe źródła danych, które Trino ogarnia



I wiele więcej.

I to wszystko darmo, bo Trino jest free & open source. Za takie ficzery i rzeczy Darmo to więcej niż uczciwa cena.

Jak chcemy mieć premium support i być klientami VIP, to są komercyjne dystrybucje z takimi bajerami w pakiecie. Dobry przykład tutaj to na przykład Starburst. Polecam.

Starczy tego gadania, bo się człowiek zmęczył.

Zapamiętajmy słowa klucze: Trino jest w pyte, Data Mesh to ciekawy koncept, Data Lake nie jest lekiem na raka, w sumie Data Mesh też nie, ale ciekawa sprawa.

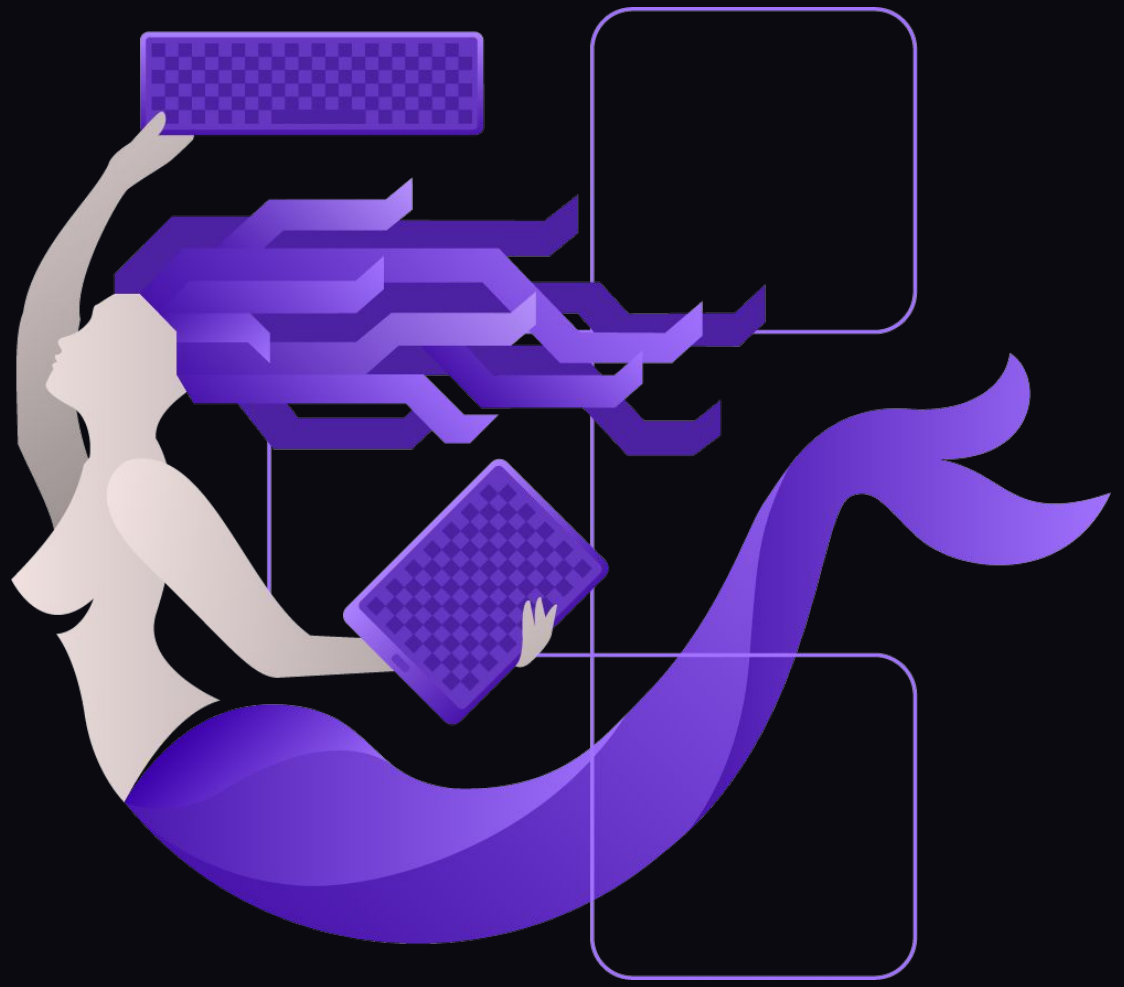
Tyle ode mnie!

Jak jesteś na b2b i lubisz CLI to oto tool, jaki stworzylem do wystawiania faktur z poziomu CLI:
https://github.com/grski/brena

A tu mój system do generowania bloga:
https://github.com/grski/braindead

Na koniec link do ksiazki - Programowanie z Górskim: Junior Python Developer
https://juniorpythondeveloper.pl/

WARSZAWSKIE DNI INFORMATYKI



Dziękujemy za oglądanie!

Zapraszamy do zadawania pytań
oraz oceny prelekcji pod nagraniem.

www.WarszawskieDniInformatyki.pl

31 marca - 1 kwietnia 2023

Politechnika Warszawska + online

ORGANIZATOR GŁÓWNY: Academic Partners FUNDACJA

KOMITET ORGANIZACYJNY: kilkadziesiąt organizacji z sektora IT / data science (pełna lista na stronie wydarzenia)