Delta Lake *vs* Iceberg

Porównanie:
Delta Lake i Iceberg to dwa formaty do przechowywania danych, które wspierają transakcyjność, time travel i ewolucję schematów.

Delta działa najlepiej w Databricks – ma super integrację, jest prosta w obsłudze i dobra na start.

Iceberg z kolei lepiej radzi sobie w dużych środowiskach – ma mocniejsze zarządzanie metadanymi i łatwiej go połączyć z różnymi silnikami. Technicznie Iceberg jest bardziej skalowalny, ale Delta ma przewagę w ekosystemie Databricks.

Kiedy co wybrać:
Jeśli trzeba szybko ruszyć, mamy Databricks i nie chcemy się rozdrabniać – wybór to Delta Lake. A jeśli budujemy coś większego, zależy nam na multi-platformowym dostępie i mamy dużo danych,wtedy Iceberg może być lepszym wyborem.

Krytyka architektury medalionowej

1. **Nadmierna złożoność warstw**

Trzy warstwy (bronze/silver/gold) mogą wprowadzać zbędne komplikacje przy prostych pipeline'ach.

2. **Duplikacja danych**

Dane są kopiowane między warstwami, co zwiększa koszty storage i utrzymania.

3. **Wydłużenie czasu przetwarzania**

Każda warstwa to osobny krok – może to spowolnić cały proces ETL.

4. **Trudność w zarządzaniu wersjami danych**

Synchronizacja i wersjonowanie między warstwami może prowadzić do niespójności.

5. **Zwiększony koszt operacyjny**

Więcej warstw to więcej pipeline'ów, monitoringów, testów i błędów do obsługi.

6. **Nieelastyczność dla ad-hoc analiz**

Czasem trzeba poczekać, aż dane przejdą do warstwy gold, zanim będzie można je analizować.

7. **Zależność od konwencji nazewnictwa i katalogów**

Brak standaryzacji struktury danych może prowadzić do bałaganu w dużych zespołach.

8. **Przerost formy nad treścią dla małych zespołów**

Dla małych firm to często overengineering.

9. **Podatność na opóźnienia i zależności**

Awaria w bronze wstrzyma cały proces aż do gold.

10. **Mało elastyczne dla danych nieustrukturyzowanych**

Tradycyjna forma medalionu sprawdza się lepiej przy tabelarycznych danych.

11. **Potrzeba dokładnego planowania schematów**

Ewolucja danych między warstwami musi być przemyślana, inaczej wprowadza bałagan.

12. **Trudniejsze debugowanie**

Błąd w gold wymaga cofnięcia się do silver/bronze i prześledzenia całego procesu.

13. **Niepotrzebna standaryzacja w niektórych projektach**

Czasem dane mogą być gotowe do użycia bez konieczności „oczyszczania" w kilku krokach.

14. **Wydłużony czas dostarczenia wartości**

Business value może być odłożona w czasie, bo dane muszą przejść cały cykl.

15. **Ograniczona przydatność w czasie rzeczywistym**

Architektura zakłada przetwarzanie batchowe, co może być za wolne dla streamingu.

16. **Zbyt ogólna koncepcja**

„Bronze/Silver/Gold" nie definiuje konkretnych reguł – różne zespoły rozumieją to różnie.

17. **Brak uniwersalności między narzędziami**

Architektura jest silnie związana z Databricks i Delta Lake – trudna do przeniesienia np. na Snowflake lub BigQuery.

18. **Trudniejsze testowanie danych**

Każda warstwa wymaga osobnych testów jakości i reguł walidacyjnych.

19. **Powielanie logiki biznesowej**

Często ta sama logika jest aplikowana w różnych warstwach, tylko inaczej sformatowana.

20. **Zamknięcie w schemacie „pipeline-centric"**

Trudno wdrożyć podejście bardziej event-driven lub API-first przy tej strukturze.