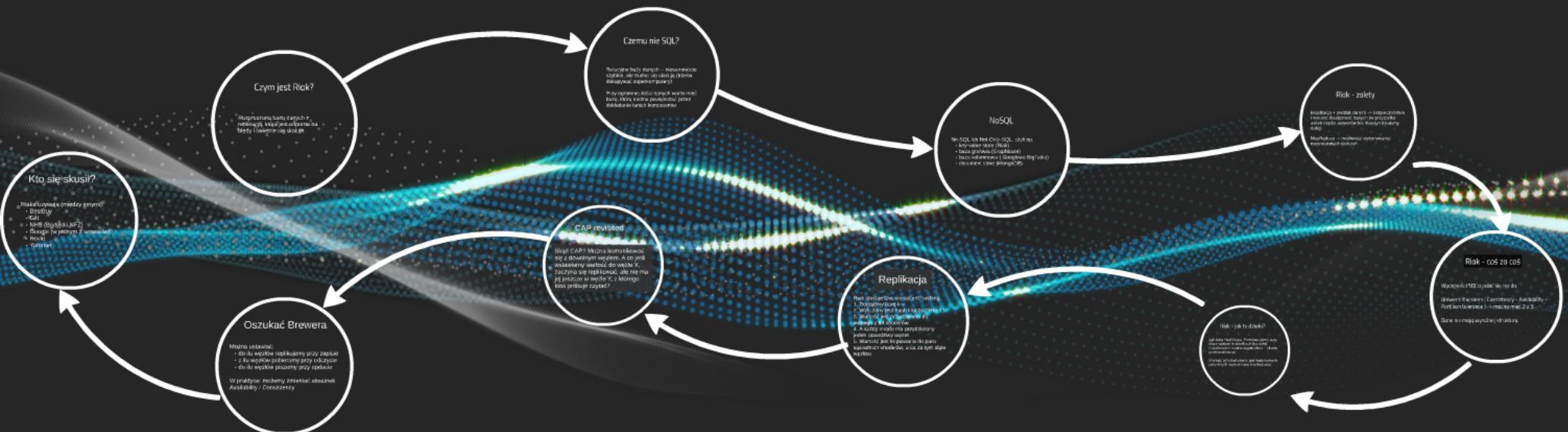
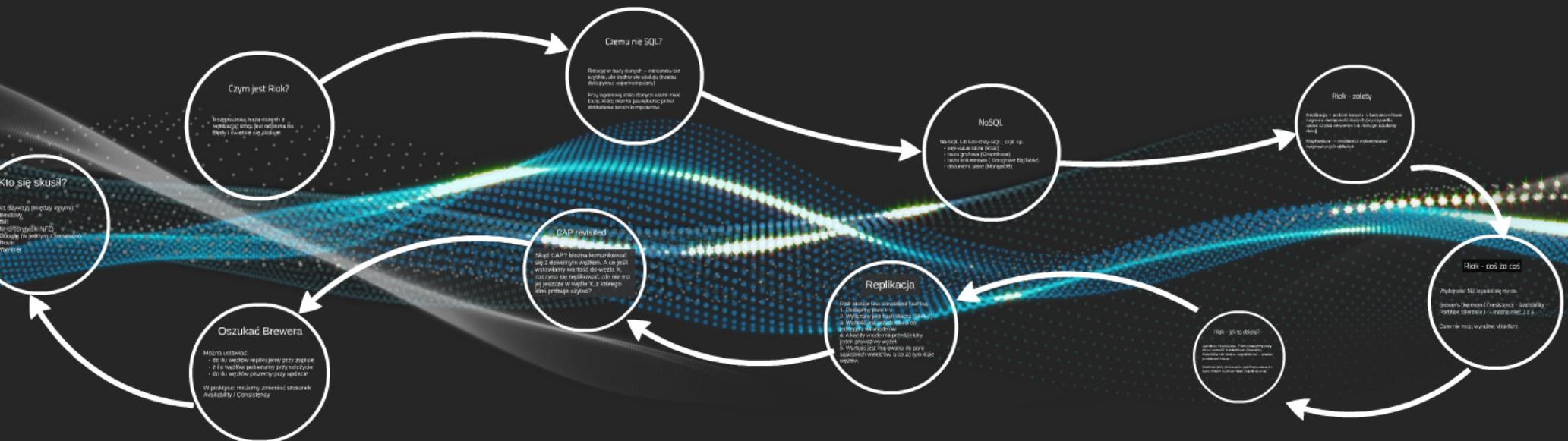


Riak



Riak



Czym jest Riak?

Rozproszoną bazą danych z replikacją, która jest odporna na błędy i świetnie się skaluje.

Czemu nie SQL?

Relacyjne bazy danych -- niesamowicie szybkie, ale trudno się skalują (trzeba dokupywać superkomputery)

Przy ogromnej ilości danych warto mieć bazę, którą można powiększać przez dokładanie tanich komputerów.

NoSQL

No-SQL lub Not-Only-SQL, czyli np.

- key-value store (Riak)
- baza grafowa (Graphbase)
- baza kolumnowa (Google BigTable)
- document store (MongoDB)

Riak - zalety

Replikacja + podział danych -> bezpieczeństwo i wysoka dostępność danych (w przypadku awarii części serwerów lub maszyn działamy dalej)

MapReduce -> możliwość wykonywania rozproszonych obliczeń



Riak - coś za coś

Wydajności SQL'a pobić się nie da.

Brewer's theorem (Consistency - Availability - Partition tolerance) -> można mieć 2 z 3

Dane nie mają wyraźnej struktury.

Riak - jak to działa?

Jak duża HashMapa. Przechowujemy pary klucz-wartość w kubełkach (buckets). Kubełków nie można zagnieździć -- płaska przestrzeń nazw.

Wartość przy dodawaniu jest kopiowana do paru innych węzłów riaka (replikowana).

Replikacja

Riak stosuje tzw. consistent hashing.

1. Dodajemy parę k-v
2. Wyliczany jest hash klucza (SHA-1)
3. Wartość jest przydzielana do jednego z 64 vnode'ów.
4. A każdy vnode ma przydzielony jeden prawdziwy węzeł.
5. Wartość jest kopowana do paru sąsiednich vnode'ów, a co za tym idzie węzłów.

CAP revisited

Skąd CAP? Można komunikować się z dowolnym węzłem. A co jeśli wstawiamy wartość do węzła X, zaczyna się replikować, ale nie ma jej jeszcze w węźle Y, z którego ktoś próbuje czytać?

Oszukać Brewera

Można ustawiać:

- do ilu węzłów replikujemy przy zapisie
- z ilu węzłów pobieramy przy odczycie
- do ilu węzłów piszemy przy updacie

W praktyce: możemy zmieniać stosunek
Availability / Consistency

Kto się skusił?

Riaka używają (między innymi):

- BestBuy
- Gilt
- NHS (Brytyjski NFZ)
- Google (w jednym z serwisów)
- Rovio
- Yammer