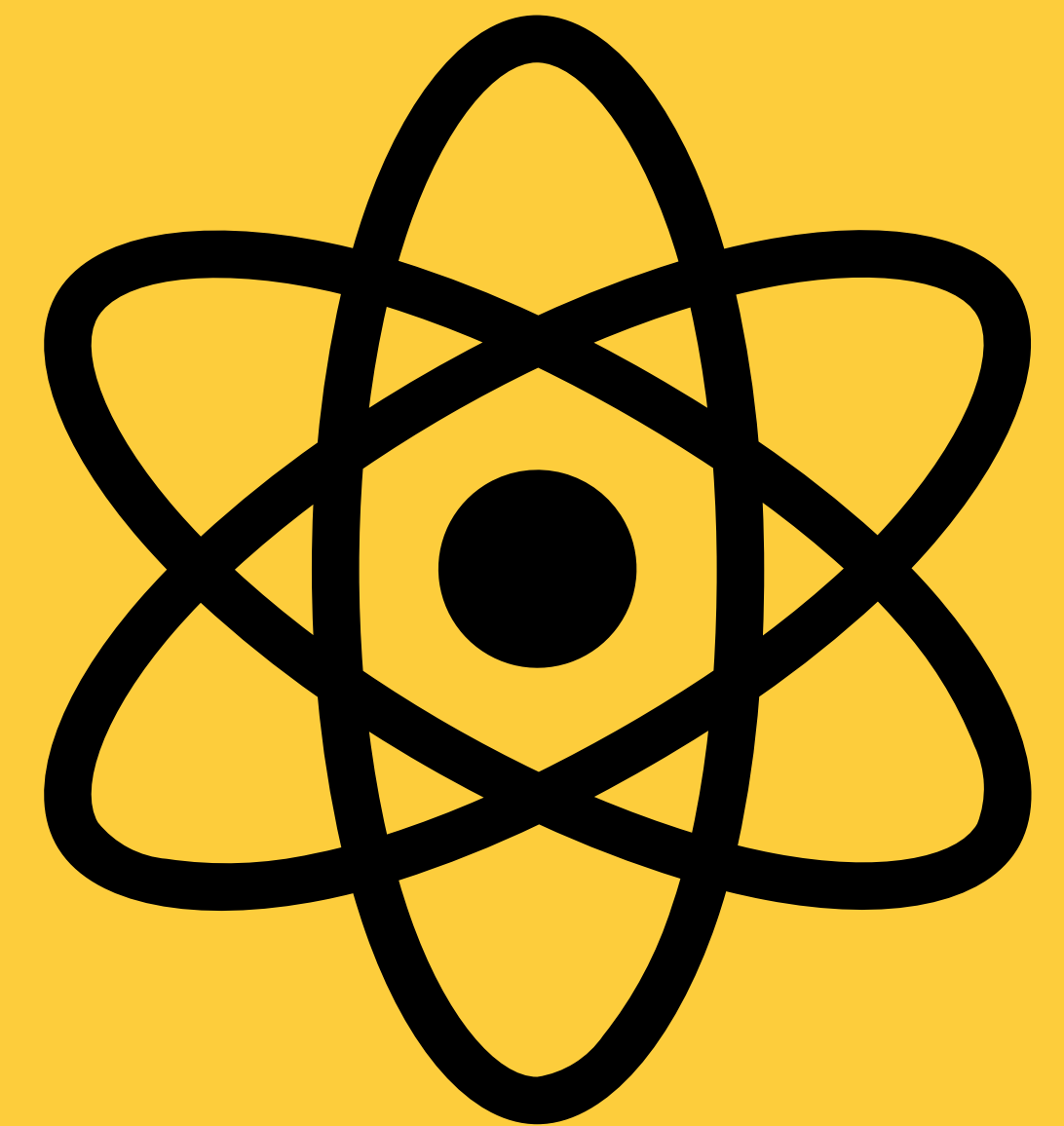


Ogarnij dane na dużą skalę za pomocą Reacta 18

Dominik Guzy





Ja, Dominik Guzy

Fullstack Software Developer
Chief Innovation Officer

<https://gorrion.io>

<https://guzy.dev>





Ja, Dominik Guzy

Fullstack Software Developer
Chief Innovation Officer

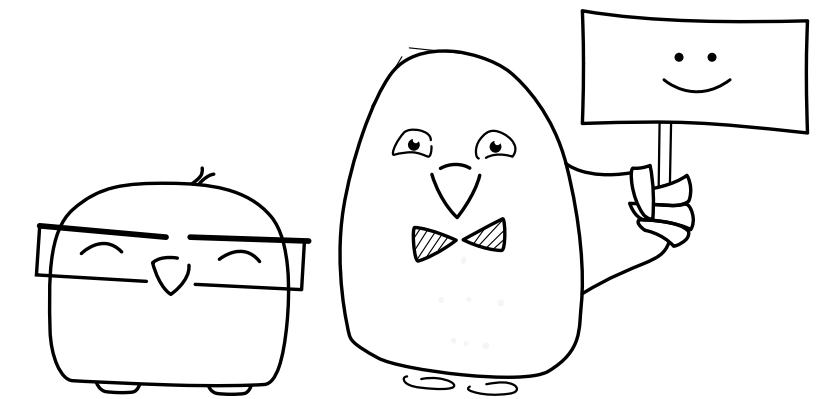
<https://gorrion.io>

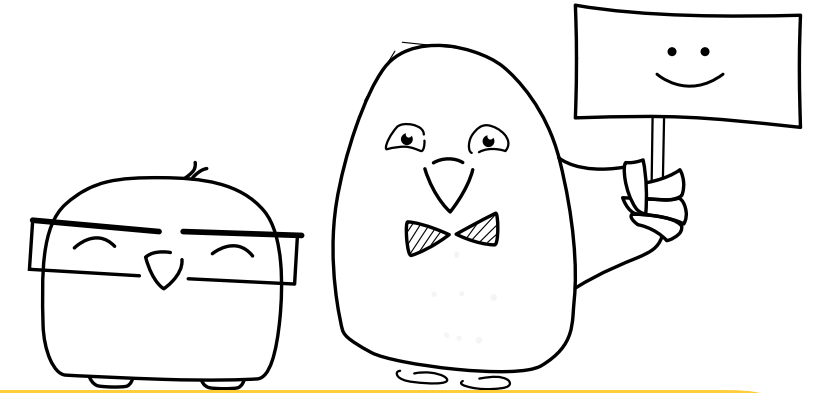
<https://guzy.dev>



Agenda

- Diagram aplikacji skalowalnych
- Nowości w React 18
- Koncepty pobierania danych w React
- Do czego (nie) służy useEffect
- Co warto wynieść z tej prezentacji (tl;dr)





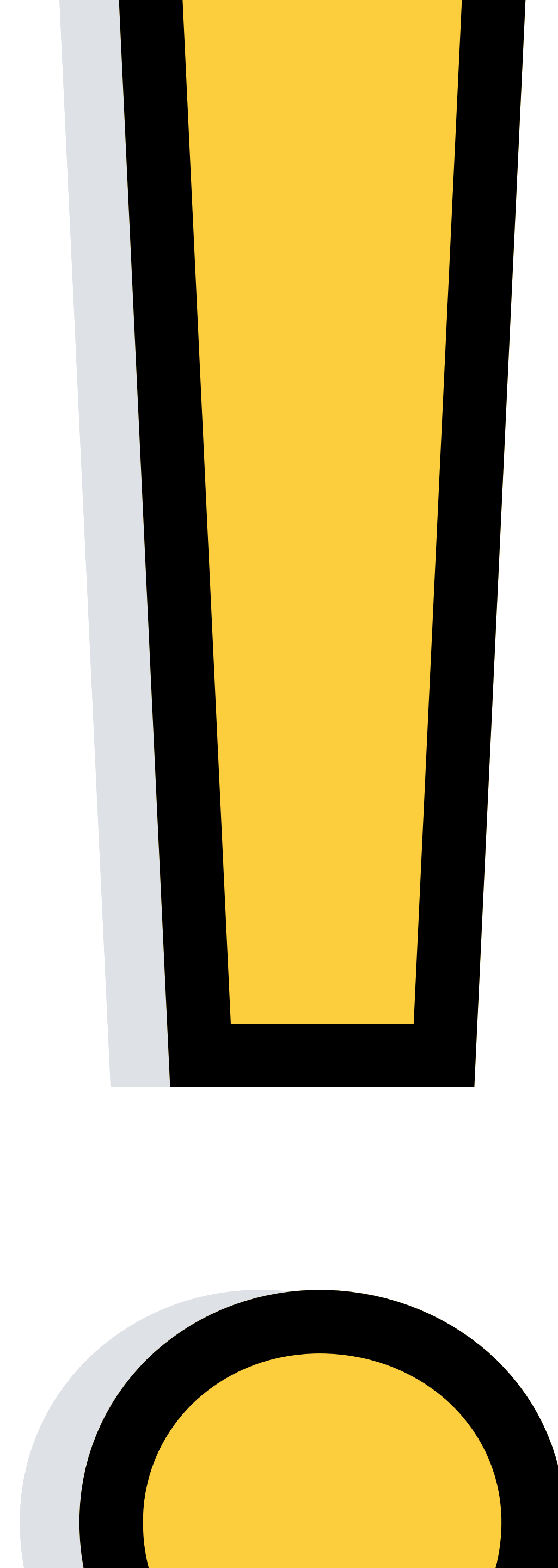
Jak wyglądają aplikacje skalowalne

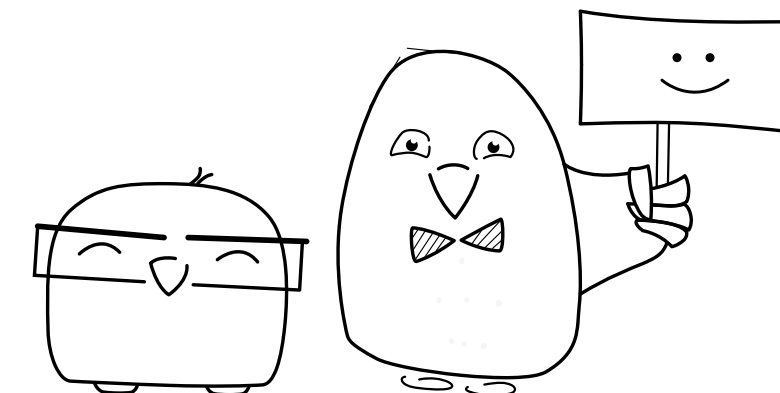


Excalidraw

Narzędzie do rysowania diagramów,
którego użyjemy, zapraszam.

<https://excalidraw.com>





Nowości w React 18

React 18 powstawał prawie 2 lata

Automatic batching

- Grupowanie kilku modyfikacji stanu komponentu w jednym re-renderze
- Wcześniej dostępne tylko w event handlerach Reacta
- Wcześniej brak wsparcia dla promise'ów, setTimeoutów, natywnych eventów

Nowe funkcjonalności

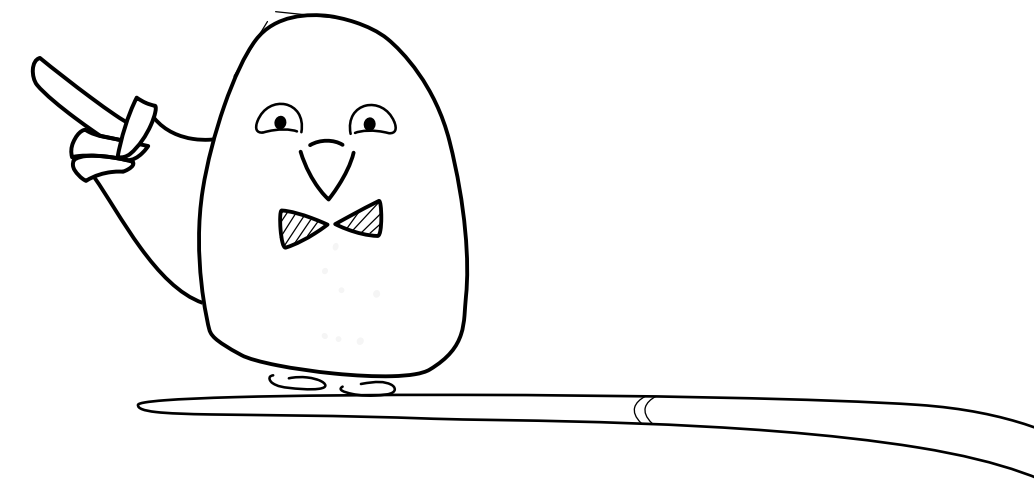
- **Transitions**
- **Nowe zachowanie strict mode**
- `useId`
- `useDeferredValue`
- **`useSyncExternalStore`**

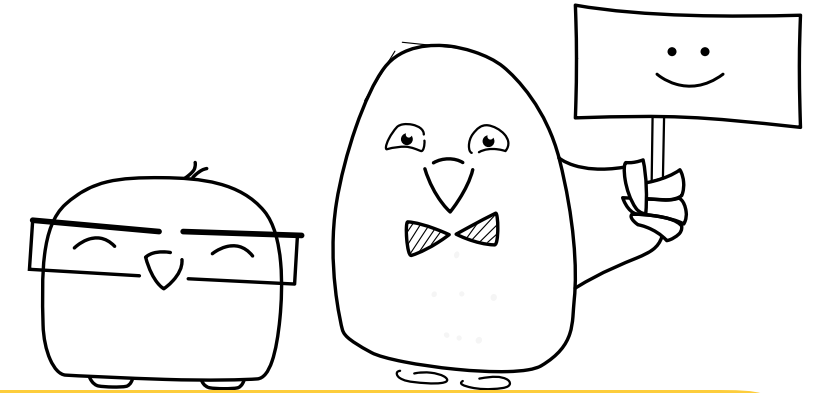
Suspense

- Wsparcie dla server-side
- Wsparcia dla frameworków
- Szerokie **wsparcie dla operacji asynchronicznych**

Concurrent renderer

- Współbieżność to mechanizm, którym w teorii nie trzeba się przejmować
- Polega na przygotowaniu przez bibliotekę kilku wersji UI w tym samym czasie
- Najważniejsza zaleta - możliwość przerwania renderu
- React może przygotowywać UI w tle, bez blokowania głównego wątku





Koncepty pobierania danych w React



Pobieranie danych

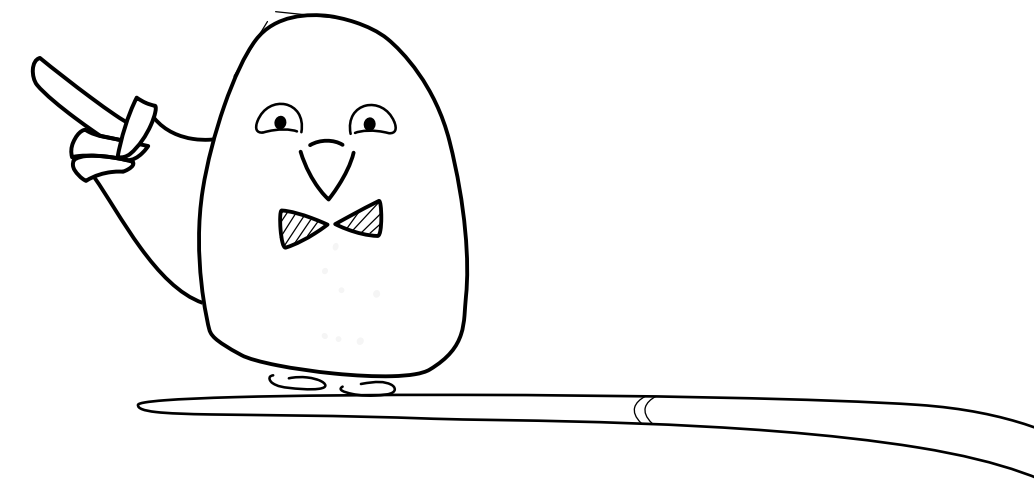
Fetch-on-Render

Fetch-then-Render

Render-as-you-Fetch

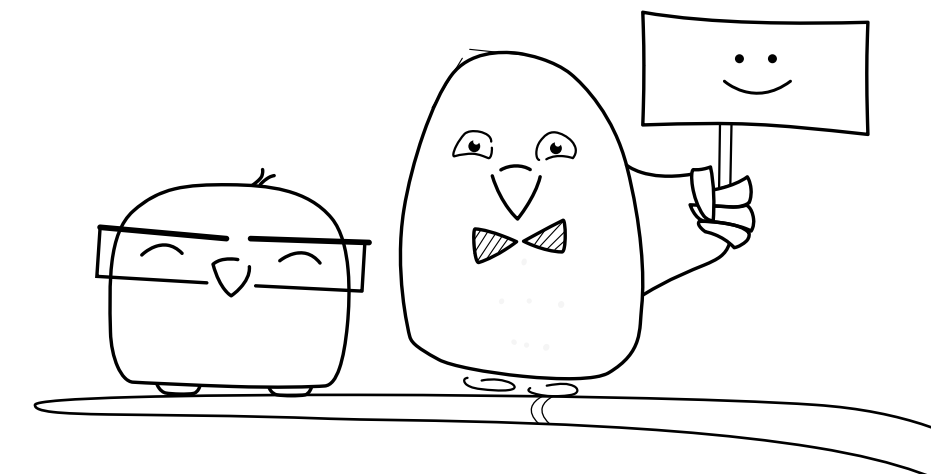
Fetch-on-Render

- Najbardziej popularny sposób
- Może powodować waterfall
- Dobry, do prototypowania
- Słabo sprawdza się na większą skalę



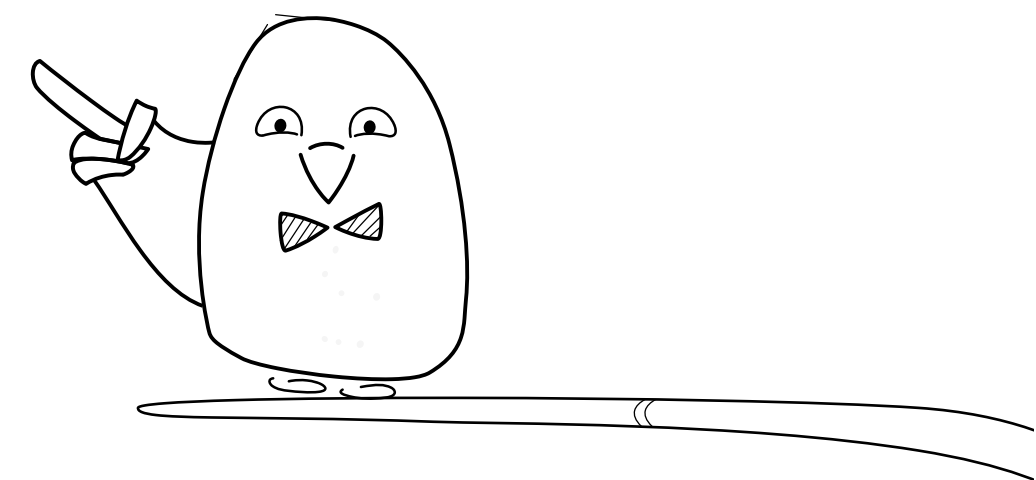
Fetch-then-Render

- Eliminuje problem Waterfall
- Może wprowadzać problem blokowania UI



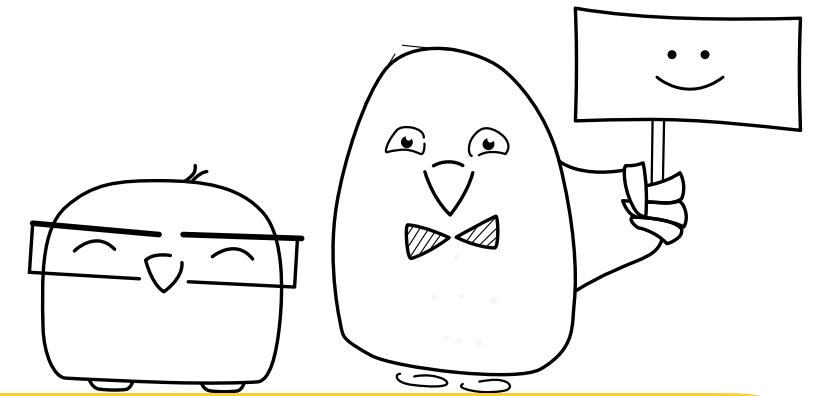
Render-as-you-Fetch

- Renderowanie z użyciem Suspense
- Nie czekamy na dane, tylko renderujemy zanim dotrą one do komponentu



**Zobaczmy każdy z tych konceptów
w akcji**

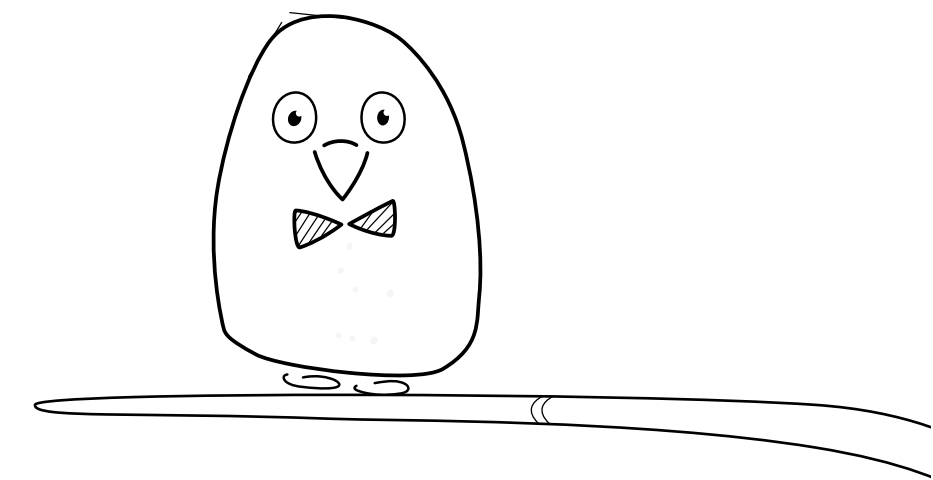




useEffect w React 18

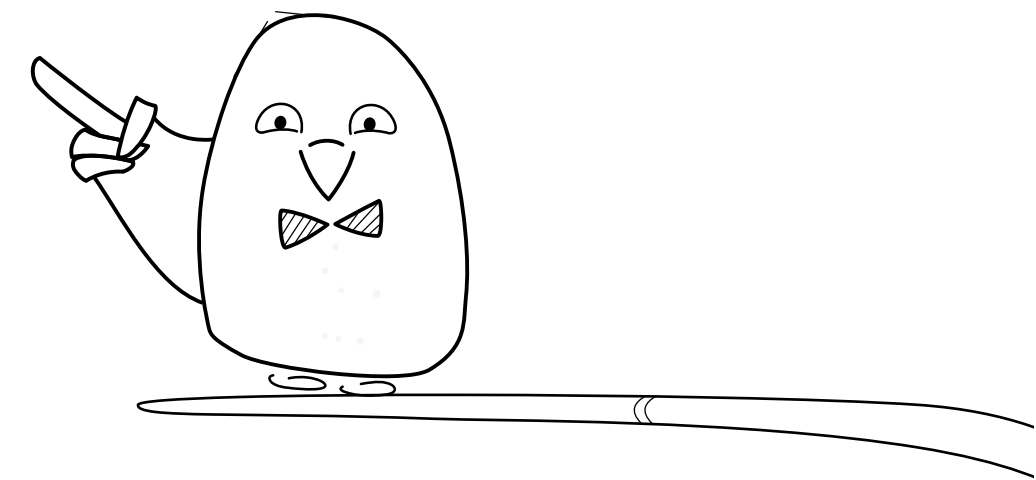
useTransition

- Zwraca stan zmiany oraz funkcję do jej startu
- Pozwala odkładać w czasie mniej “pilne” zmiany



useSyncExternalStorage

- Hook stworzony z myślą o bibliotekach
- Służy do odczytu i subskrypcji zewnętrznych źródeł danych
- Ma wsparcie dla narzędzi współbieżnych



Zobaczmy oba te Hooki w akcji



Bo useEffect nie służy do pobierania danych...





Pobieranie danych

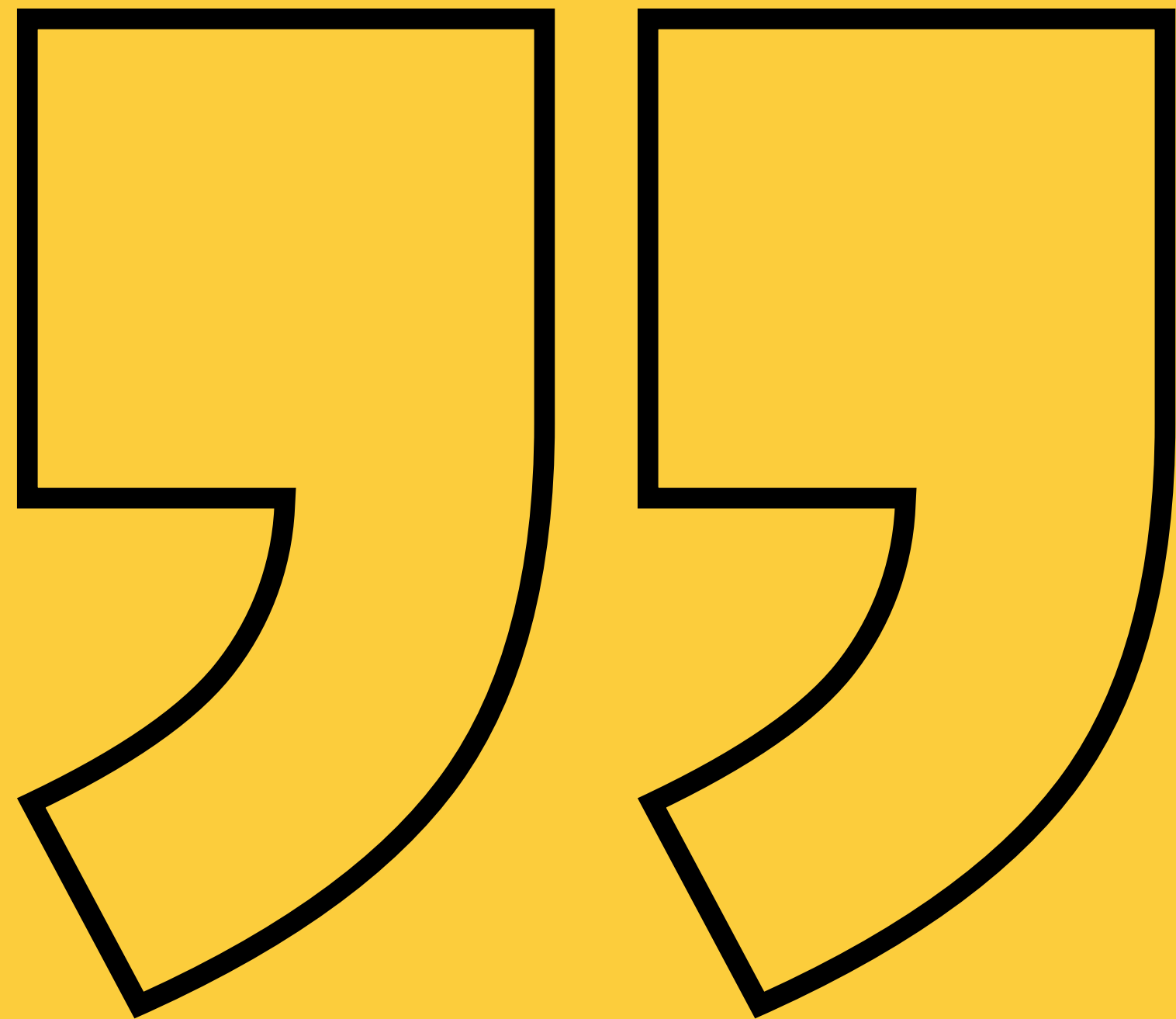
react-query

@apollo/client

swr

i inne

tl;dr



- React 18 przychodzi z współbieżnością
- Współbieżność pozwala na przerywanie renderowania i tworzenia na raz kilku wersji UI
- Żeby poprawnie korzystać z możliwości współbieżności, trzeba użyć bibliotek, np. react-query lub frameworków, np. Next.js
- useEffect w React nie służy do pobierania danych

Dziękuję serdecznie!

Dominik Guzy

