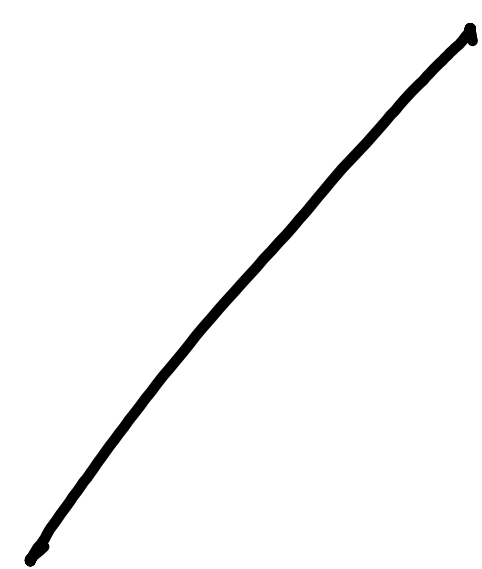


Od statycznej strony
do natywnego klienta



Gdzie jest stem aplikacji
webowych?

MOŻLIWOŚCI

ŁATWOŚĆ

DO STOSOWUJEMY

- STRONY SERWER
- STRONY STATYCZNE
- WYMIANA PRODUKOWANIE HTML'A
- DZIŁANIE HTML'A
- CAŁY KLIENT
- CAŁY KLIENT (SCRIPTS)

STRONY STATYCZNE

PO CO HTML i HTTP

- powstanie internetu: 1969
- powstanie HTTP i HTML: 1991

HTML (+CSS + JS)

- specyfikacja klienta

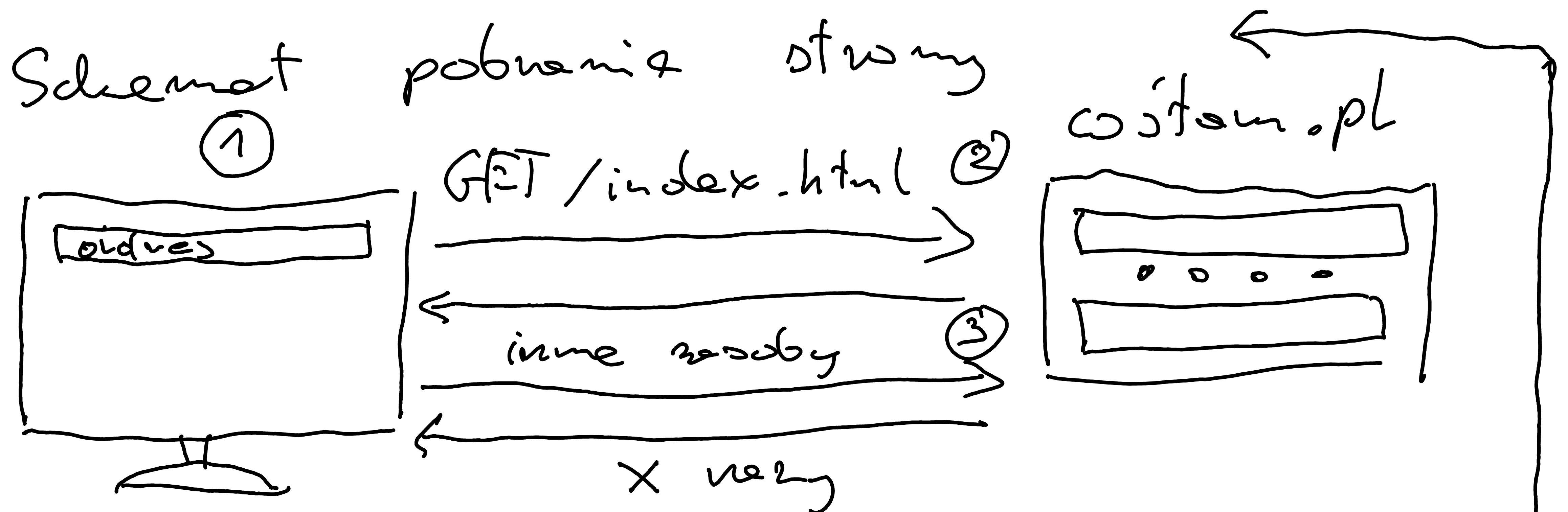
HTTP

- sposób wymiany HTML'a (i innych zasobów)

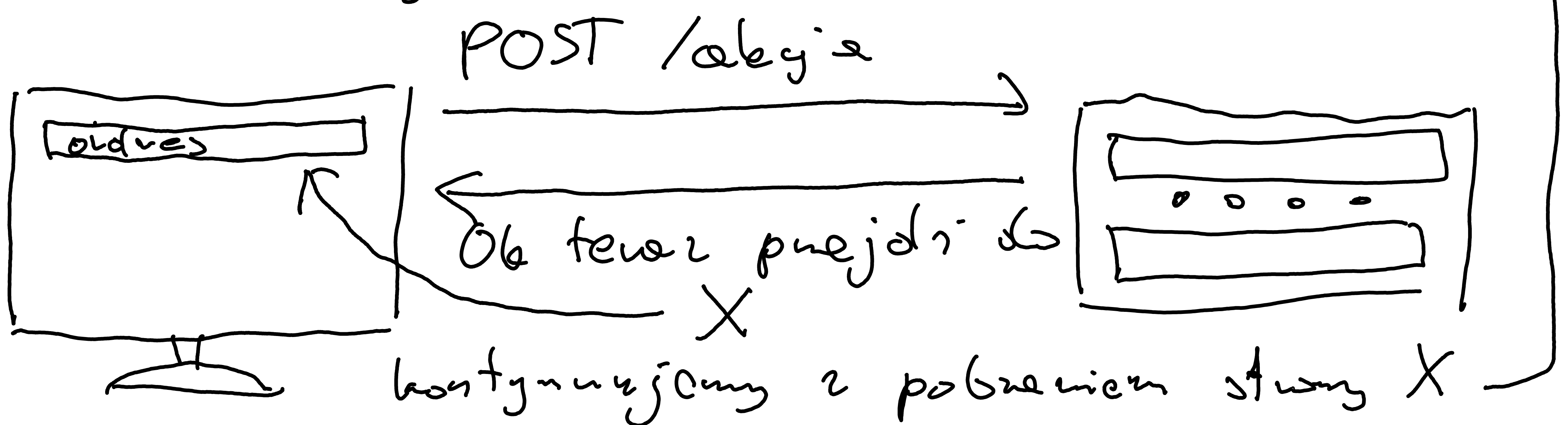
SPOSÓB KOMUNIKACJI : HTTP

Dostępne metody komunikacji:

- GET (Pobranie statycznego zasobu)
- POST (Wykonanie akcji)



Schemat wykonanie akcji

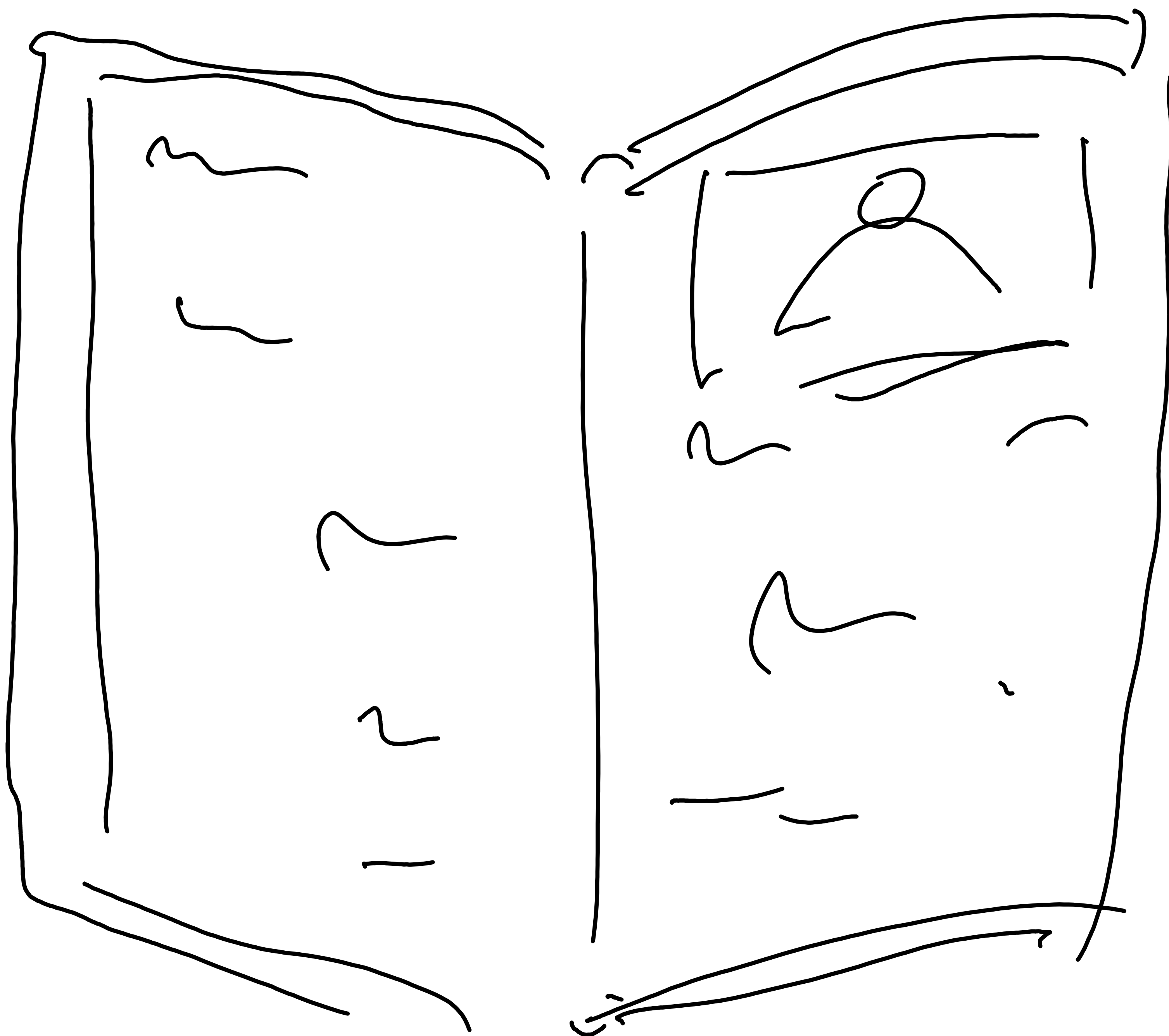


GD21E JEST STAN?

https:// www.Example.com/news/1

?x=1&y=2#title

- server address
- query string
- doctype name (+ the POST) } formdata
- cookies (or data?)



KROK W TYŁ - MARKDOWN

KROK W PRZÓD -

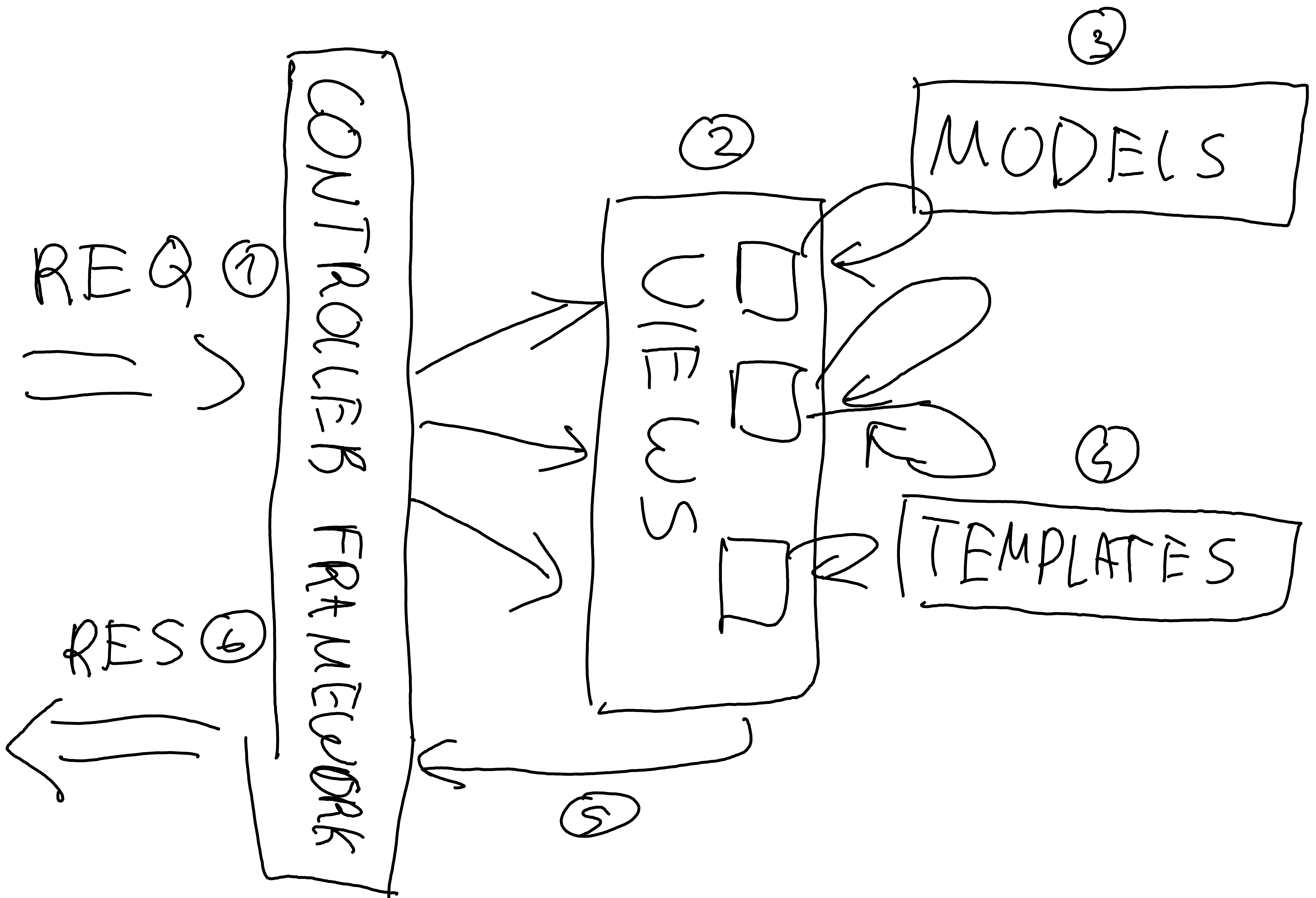
- DOSTOSOWANIE SERWERA

- klasycznie serwujemy pliki
w katalogach

- dostosowanie server robi co
inne

MTV+C

Model Template View
+ Controller



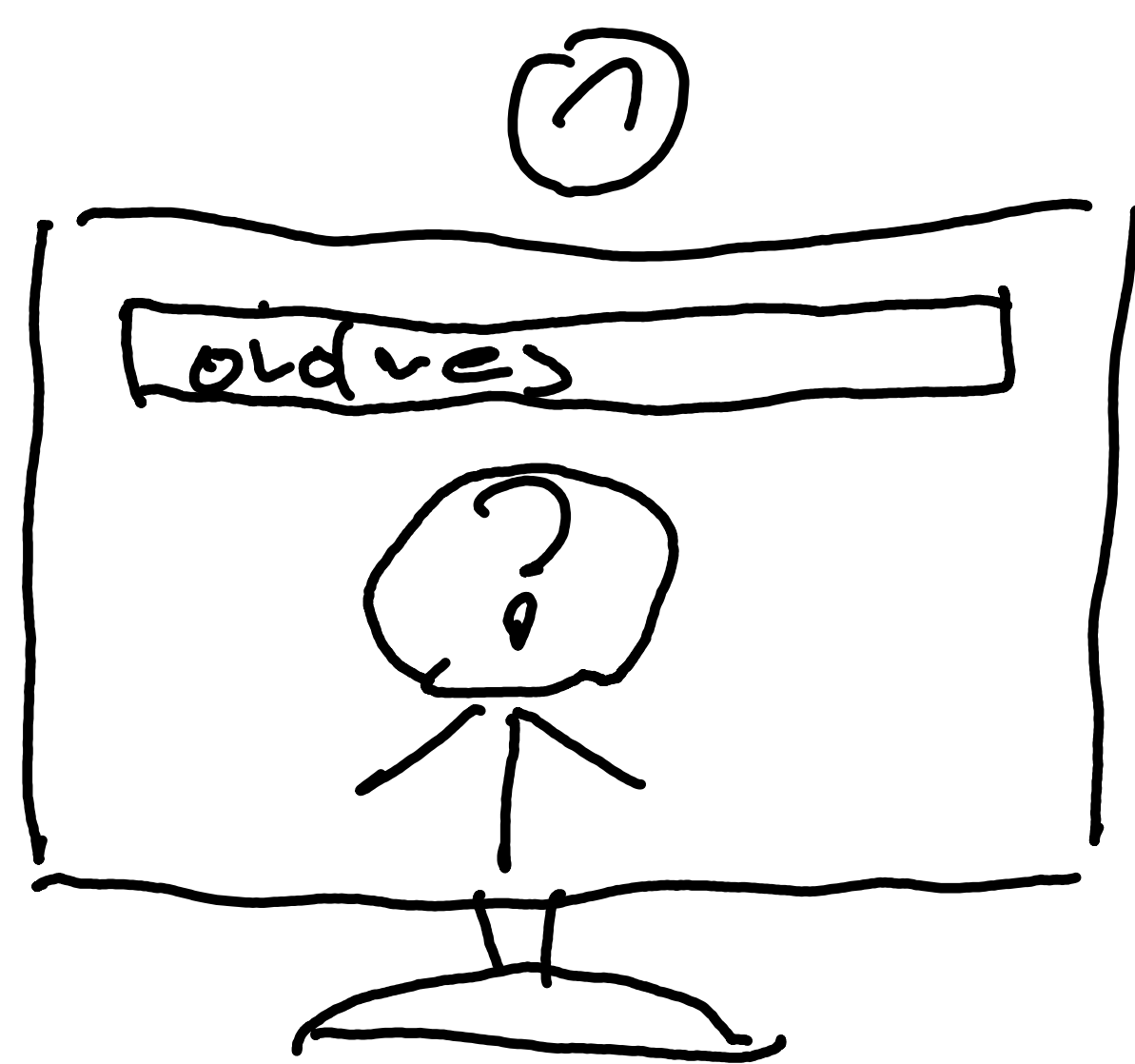
CO ZROBIĆ Z DUŻYM STANEM?

- trzymamy go na serwerze
- identyfikujemy klienta z użyciem
COOKIES

COOKIES

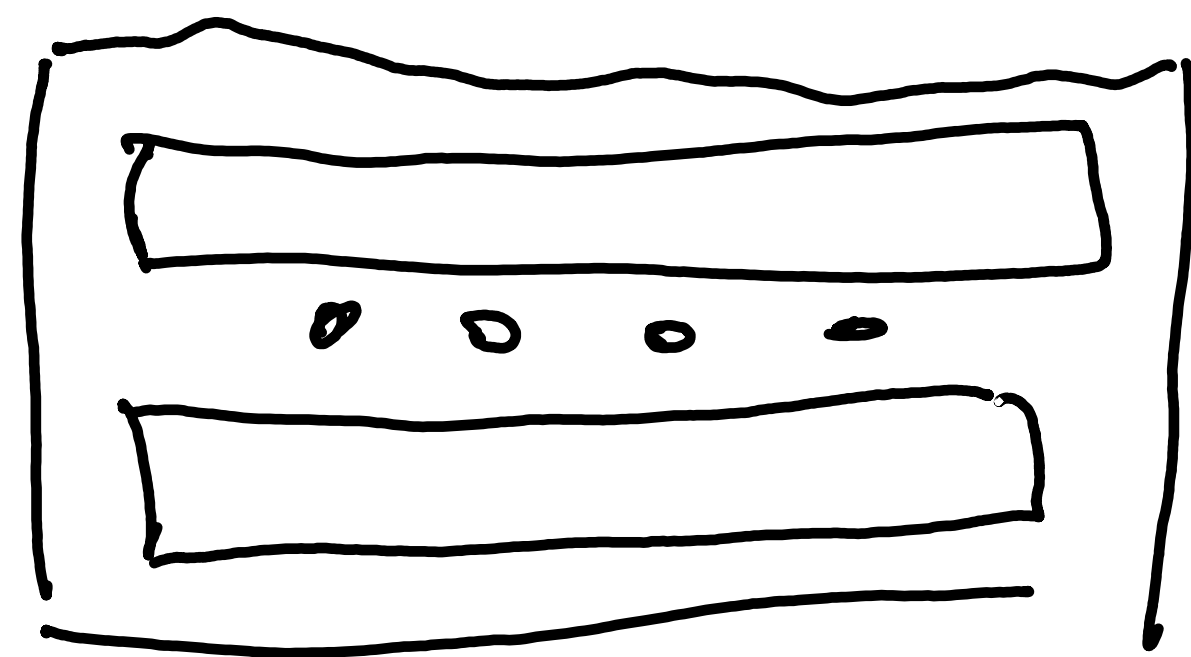
- małe dane ustawione przez
serwer dla domeny
- odsyłane przy następnych odwiedzeniach

SCHEMAT LOGOWANIA



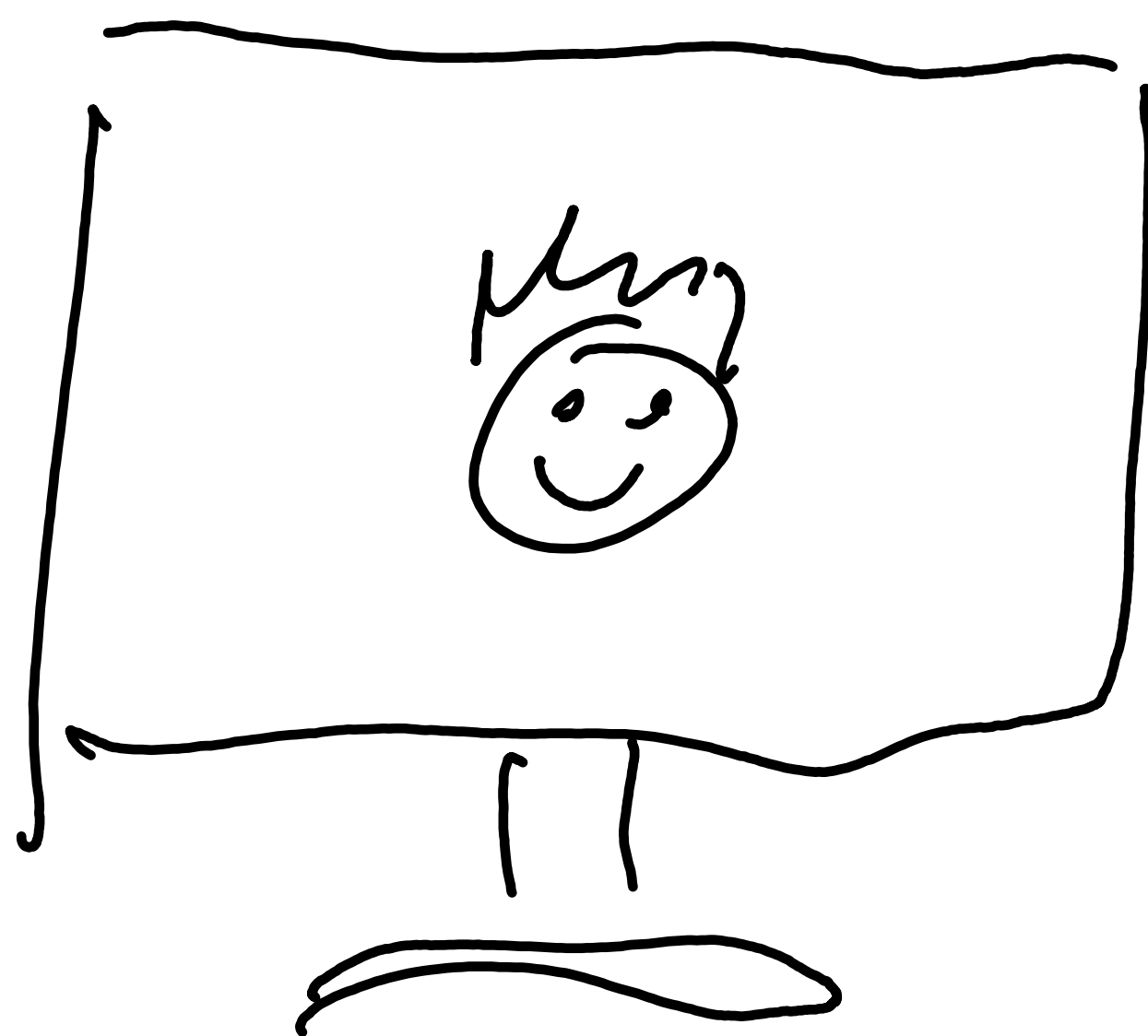
POST / login (Doe Gyanez) ②

Jestem Marek.
Możesz być ptym
pewow, to moje
konto to Marek?



Przebieg
Marek ③

Jdn do strony domowej
Header set cookie ③



session_id : - - - -

Ob dostajesz
funkcyjny identyfikator

GET / home + cookie ④

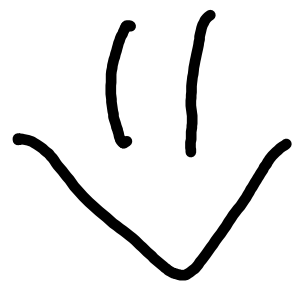
Jestem Marek z identyfikatorem
Poj mi stronę domową

ZALETY

- dość prosty, ale prosty setup

WADY

- wszystko przechodzi przez server



mało dobre user experience

ROZWIĄZANIE:

- Skrypty JS

- AJAX (Asynchroniczny JS i XML)

- dostosowanie klienta

- swobodne wymiana danych z serverem

(użyto JSON, REST API)

WYZWAŁIA SKRYPTÓW

- duplikacja stanu (na stronie i w lokalnie)

ROZWIĄZANIE WIELOKROTNEGO STANU

- podejście standardowe:

- stan tylko po stronie skryptu
- obiektywne opis interfejsu jako reguły produkcji ze stanem

• reaktywność

• nowoczesne frameworki webowe:

React, Vue JS, ...

- podejście nietandardowe

- stan tylko w HTML'u
- ulepszenie HTML'a [HTMX]

CO DALEJ? (KIERUNKI ROZWOJU)

- Stan u klienta (Indexed P/B i inne)
- Dostosowanie komunikacji HTTP
 - web workers
 - stany offline
 - PWA (przecie natywne aplikacje)
- Dodatkowe api webowe min. WebRTC
- Natywne klienty oparci o jedn. stopnia o technologie webowe (Electron, React Native)

KONKLUZJE:

DOBIE RZĄDZENIA

DO ZADANIA

