



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Rozproszone systemy internetowe

*Representational
State Transfer*

„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”

Nr projektu: UDA-POKL.04.01.01-00-143/09-00

REST

Fielding, Roy T.; Taylor, Richard N. (2002-05),
*"Principled Design of the Modern Web
Architecture"*

- Rozproszone, interaktywne hipermedia w ogromnej sieci - czy można to opisać ?
- ***Głównym zadaniem sieci WWW jest stworzenie współdzielonej przestrzeni informacyjnej, którą ludzie i maszyny wykorzystają do komunikacji***

(T.Berners-Lee).

Wymagania stawiane WWW

- Proste a jednocześnie elastyczne formaty danych
 - hipermedia dają możliwość konstrukcji złożonych zależności
 - informacja może być niekompletna lub dostępna tylko częściowo
 - tekstowe protokoły łatwo rozwijać i testować
- Rozszerzalność
- Minimalizacja interakcji sieciowych (rozproszone dane)

Wymagania stawiane WWW

- "Anarchiczna skalowalność" (anarchic scalability)
 - zdolność do pracy w warunkach nieprzewidzianego obciążenia lub po otrzymaniu błędnych/zniekształconych danych
 - wiele jednostek zarządzających danymi
 - klient nie musi posiadać pełnej informacji o serwerach, serwer nie musi pamiętać stanu klienta pomiędzy wywołaniami
 - podstawowy zestaw operacji nie może wymagać danych zaufanych
- Niezależna dystrybucja oprogramowania

Elementy architektury: dane

- **zasób (resource)** - cel hipertekstowej referencji jako idea a nie konkret
- **identyfikator zasobu** - URI
- **reprezentacja** - np. dokument HTML
- **metadane reprezentacji** - np. typ mediów
- **metadane zasobu** - np. nagłówek Vary
- **dane kontrolne** - np. nagłówki If-Modified-Since, Cache-Control

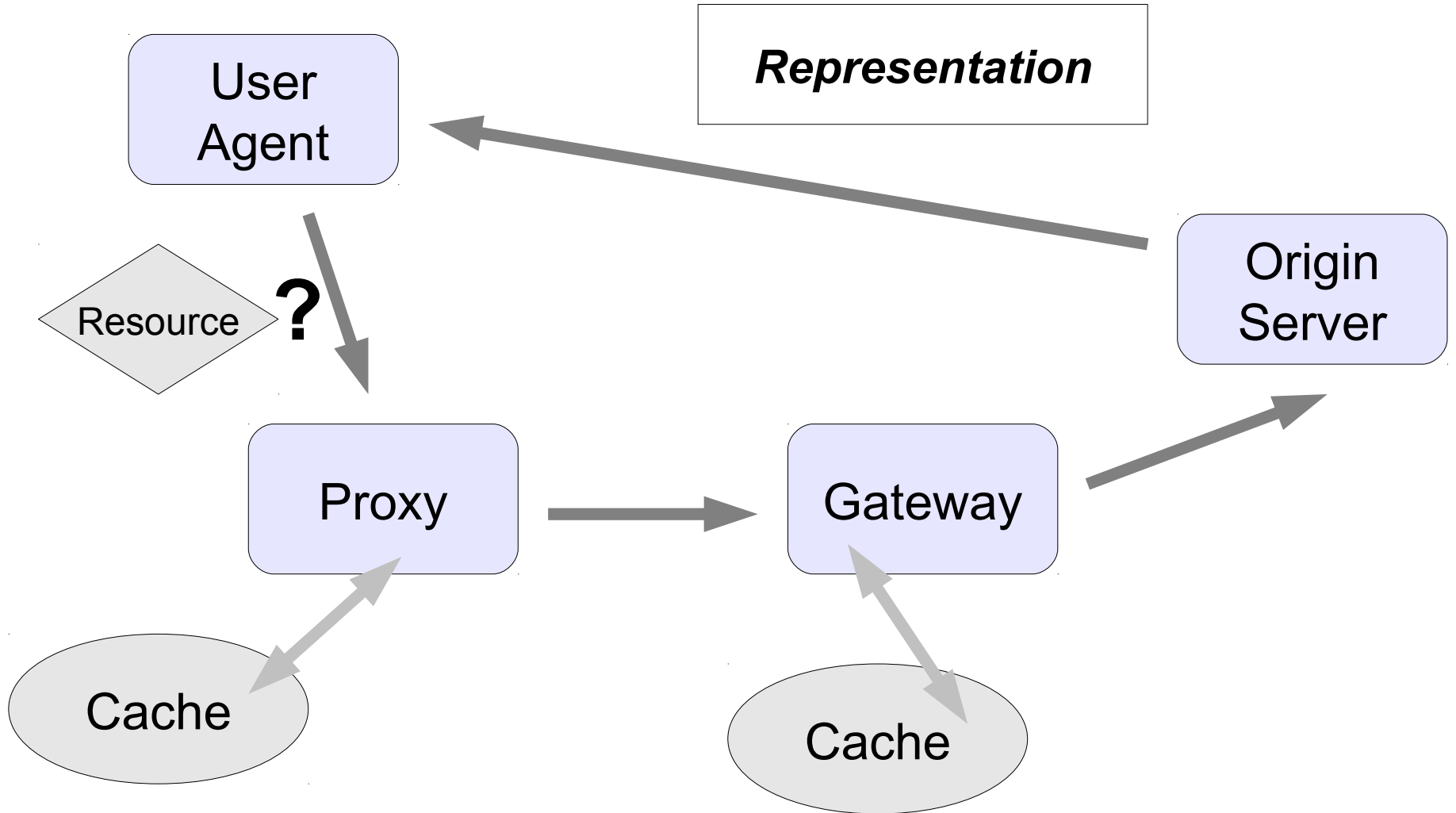
Elementy architektury: złącza

- **Złącze (connector)** - element odpowiedzialny za dostęp sieciowy do zasobu oraz transfer reprezentacji
 - dostęp bezstanowy - każdy komunikat zawiera komplet informacji potrzebnych do jego przetworzenia (mniejsze zużycie zasobów, przetwarzanie równoległe oraz proxy, wykorzystanie pamięci podręcznej)
- Typy: klient, serwer, *cache*, *resolver*, *tunnel* (*SOCKS*, *SSL*)

Elementy architektury: komponenty

- Komponenty to elementy przetwarzające komunikaty i dane
 - **Origin server**
 - **Gateway** - element pośredniczący wykorzystywany przez serwer
 - **Proxy** - element pośredniczący wykorzystywany przez klienta
 - **User agent**

Przetwarzanie



Aplikacja

- *Stan aplikacji definiowany jest przez: oczekujące żądania, topologię połączonych komponentów (filtrowanie), aktywne żądania, przepływ reprezentacji będących odpowiedziami oraz przetwarzanie reprezentacji po stronie klienta.*
- *Aplikacja osiąga stan stabilny jeżeli nie ma już żadnych niezrealizowanych żądań a wszystkie reprezentacje zostały przetworzone (lub można je tak traktować)*
- *Użytkownik postrzega aplikację jako sekwencję przejść pomiędzy stanami stabilnymi*

R.T.Fielding, R.N.Taylor

Restful Web Services

- Zestaw URI do odwołań (bazowe + dodatki)
- Typ mediów, który służy do budowania reprezentacji
- Zestaw operacji (np. metody HTTP) do manipulowania zasobami
 - CREATE = **PUT**
 - READ = **GET**
 - UPDATE = **POST**
 - DELETE = **DELETE**

JAX-RS: Java API for RESTful Web Services

```
@Path("/wyklad/test/")
@Produces("application/xml")
public class RWS {

    @GET
    public List getList() { }

    @GET
    @Path("/customers/{id}")
    @Produces("application/json")
    public Object getData(
        @PathParam("id") String id) { }
}
```

The diagram illustrates the mapping of JAX-RS annotations to their functional roles:

- URI**: Points to the `@Path` annotations in the class and the `getData` method.
- MIME (metadane reprezentacji)**: Points to the `@Produces` annotations in the class and the `getData` method.
- Metoda**: Points to the `@GET` annotations on the `getList` and `getData` methods.

Zagadnienia do samodzielnego zbadania/przemyślenia

- Przeczytaj artykuł podany na początku wykładu i zastanów się nad następującymi pytaniami:
 - jakie elementy specyfikacji HTTP (oraz specyfikacji dodatkowych) są niezgodne z koncepcją REST ?
 - czy model REST może być realizowany za pomocą protokołów innych niż HTTP ?
- Zapoznaj się ze specyfikacją standardu ATOM (dokumenty RFC 4287 i 5023)