



**Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego**

# Rozproszone systemy internetowe 2

## Common Object Request Broker Architecture

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

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

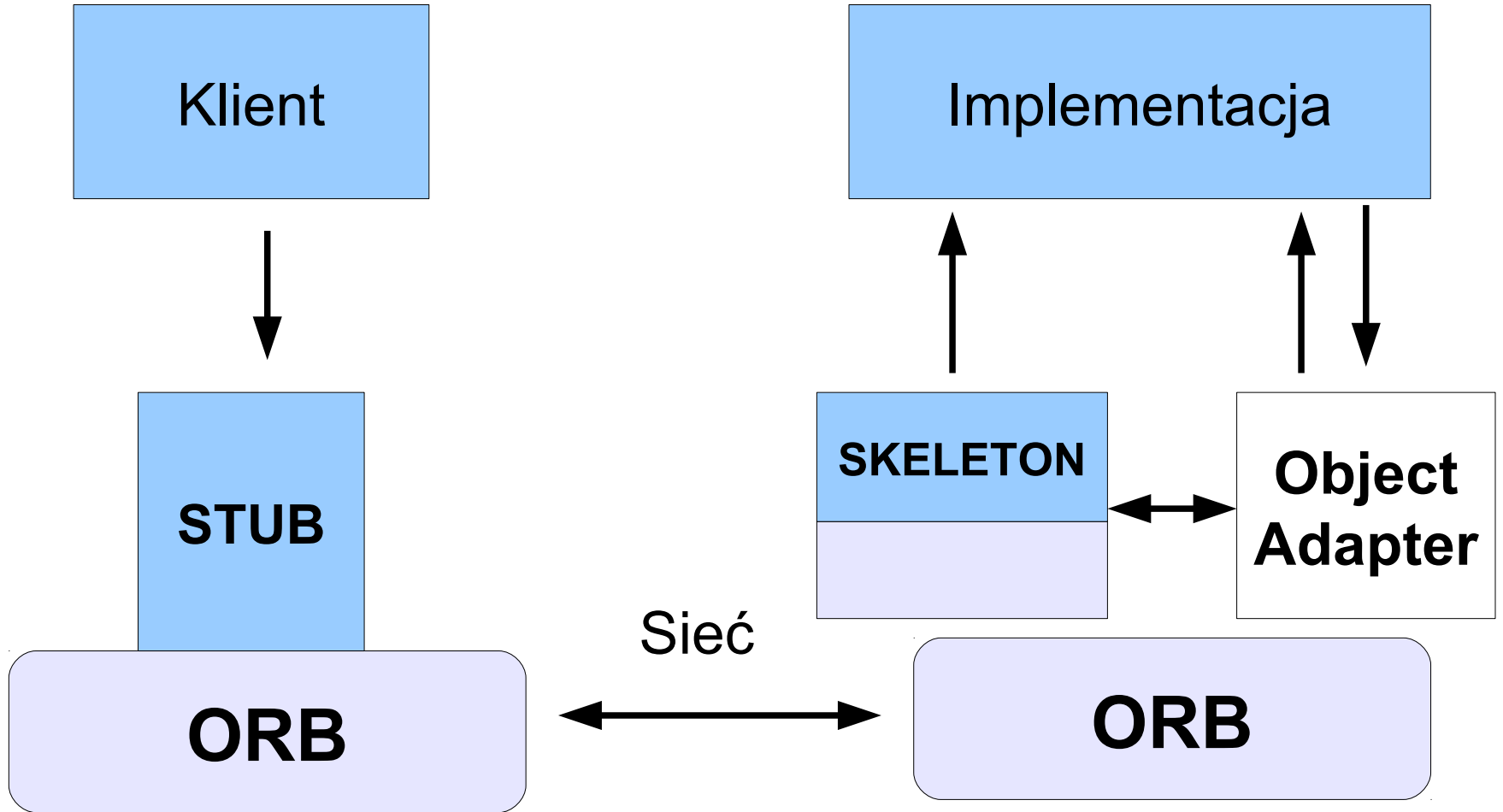
# CORBA

- Object Management Group: IIOP, CORBA ale także UML
- General Inter-ORB Protocol
- Middleware
- Standard a nie implementacja
- Model obiektów rozproszonych
- Wspomaga współpracę starych implementacji z nowymi (niezależne od języka programowania, platformy itp.)

# Model obiektowy

- Wszystkie dobrodziejstwa OOP:  
hermetyzacja, dziedziczenie, polimorfizm
- Hermetyzacja umożliwia:
  - przezroczystość dystrybucji obiektów  
(interfejsy + implementacje oraz pieńki klienta + szkielety serwera)
  - Współpraca pomiędzy różnymi językami programowania (interoperability)
    - Definiujemy interfejs w standardowym języku a potem generujemy pieńki i szkielet

# Struktura wywołania v. 1.0



# Interface Definition Language

- Niezależny od: języka programowania, systemu operacyjnego, sprzętu, realizacji sieci (?)
- Konwertowany do postaci binarnej i przechowywany w Interface Repository (unikalne ID)
- Obiekty CORBy też są opisywane interfejsami IDL
- Składnia podobna do C (kierunek przekazywania parametru)
- Mapowania dla języków programowania
- Kompilatory IDL (stub, skeleton)

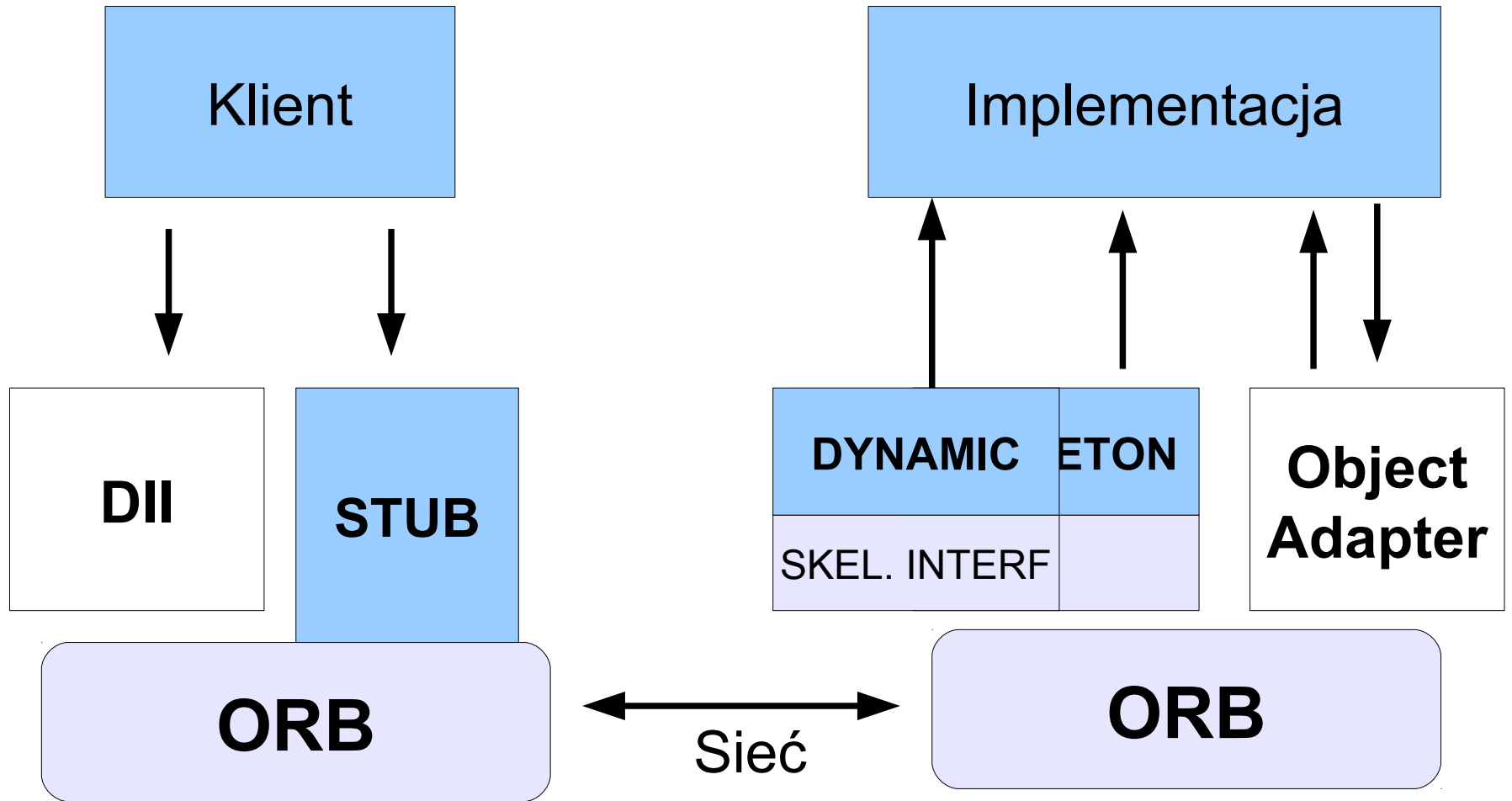
# Dynamic Invocation Interface

- Załóżmy, że interfejs się zmienia:
  - Wyszukaj i załaduj IDL z IR
  - Wygeneruj żądanie
  - Określ cel, operację i parametry
  - Wywołaj operację

```
invoke(remoteObj, "oblicz", params)
```

```
remoteObj.oblicz(params)
```

# Struktura wywołania v. 1.1



# Referencje obiektów

- Lokalna reprezentacja (zależna od języka np. wskaźnik na pieniądze)
- IOR - Inter-operable Object Reference

<b>Nazwa typu ID w repozytorium</b>	<b>Protokół, Host, port</b>	<b>Identyfikator obiektu (Adapter + nazwa)</b>
---	---------------------------------	--

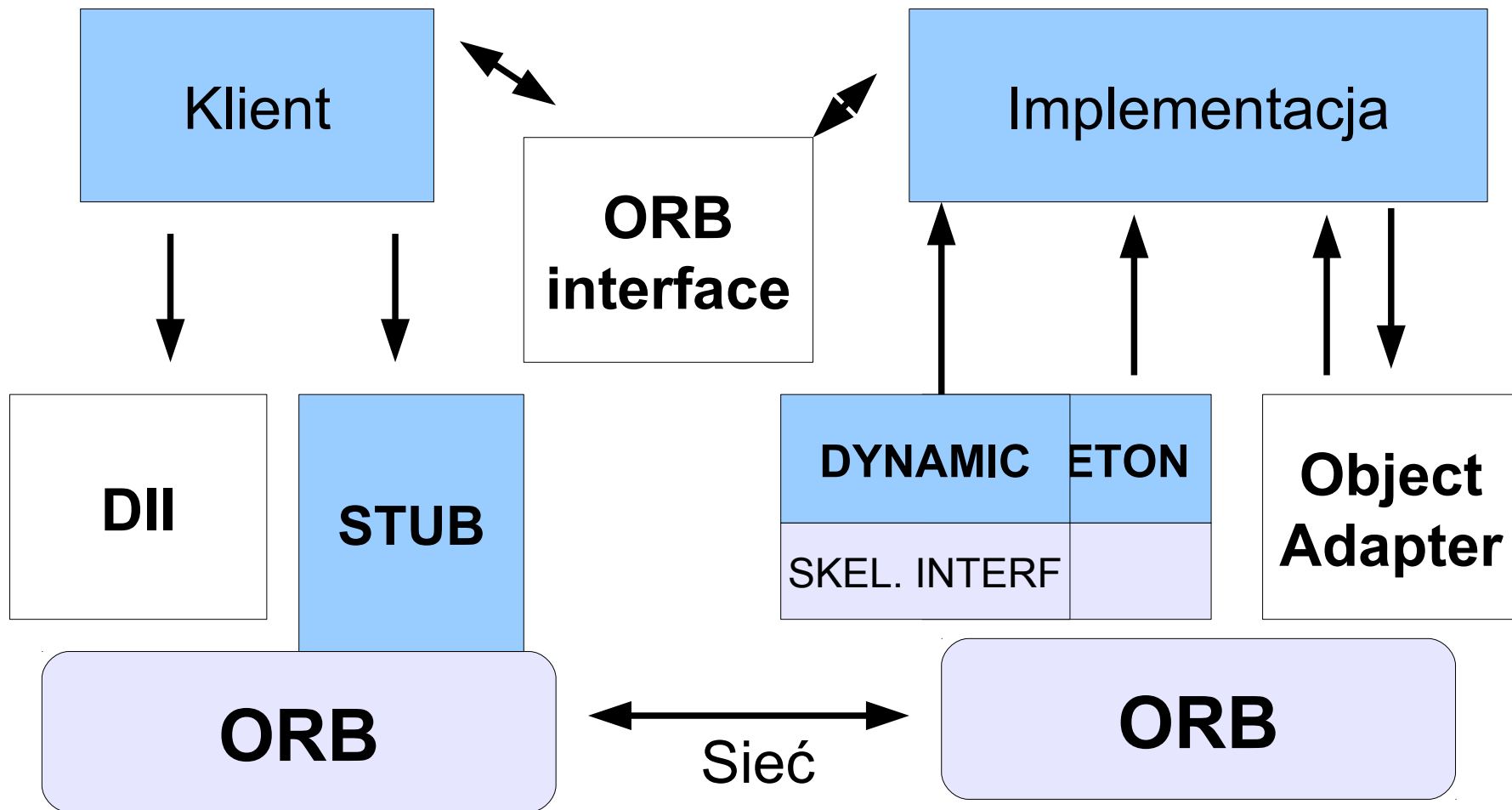
- Reprezentacja tekstowa



# Interfejs ORB

- Umożliwia komunikację z podsystemem CORBA (implementowany w każdym brokerze)
- Inicjalizacja środowiska – `resolve_initial_references()` daje referencje do systemu nazw, IR itp..
- `string_to_object()` - zamiana nazw na referencje
- Rozproszone odśmiecanie (zliczanie ref.)

# Struktura wywołania v. 1.2



# Portable Object Adapter

- W przypadku wielu wywołań skierowanych do tego samego interfejsu należy określić politykę wykonania operacji (kolejkowanie żądań, wielowąkowość)
- POA posiada referencje do szkieletów
- Każdy obiekt ma identyfikator
- Active Object Map przechowuje powiązania pomiędzy obiektami a szkieletami
- ORB multipleksuje wywołania pomiędzy odpowiednimi POA

# Obiekty trwałe (persistent)

- IOR obiektu trwałego zawsze wskazuje na ten sam obiekt
- Automatycznie tworzony po restarcie ORB
- Implementation Repository (IMR)
  - Object Adapter + Startup Command + Server
- Replikacja – kilka lokalizacji dla jednej referencji

# Znajdowanie IOR

Dwa schematy:

**corbaloc** – lokalizacja obiektu na wskazanym ORB

**corbaname** – znajdowanie obiektów w systemie nazewniczym

**corbaloc:iiop:1.2@www.server.pl:1050/NameService**

**corbaloc:rir:/NameService**

**corbaname::localhost:1050/NameService/Hello**