

# Systemy Rozproszone

## Technologia ICE

Zespół Systemów Rozproszonych (DSRG)

Katedra Informatyki AGH – Kraków

Kornel Skalkowski

[skalkow@agh.edu.pl](mailto:skalkow@agh.edu.pl)

Opracowano na podstawie książki: Henning M, Spruiell M. – Distributed Programming with Ice, ZeroC

# Agenda

- Wprowadzenie
- Architektura
- Możliwości
- Tworzenie własnych aplikacji
- Ćwiczenie
- Projekt
- Literatura

# Wprowadzenie

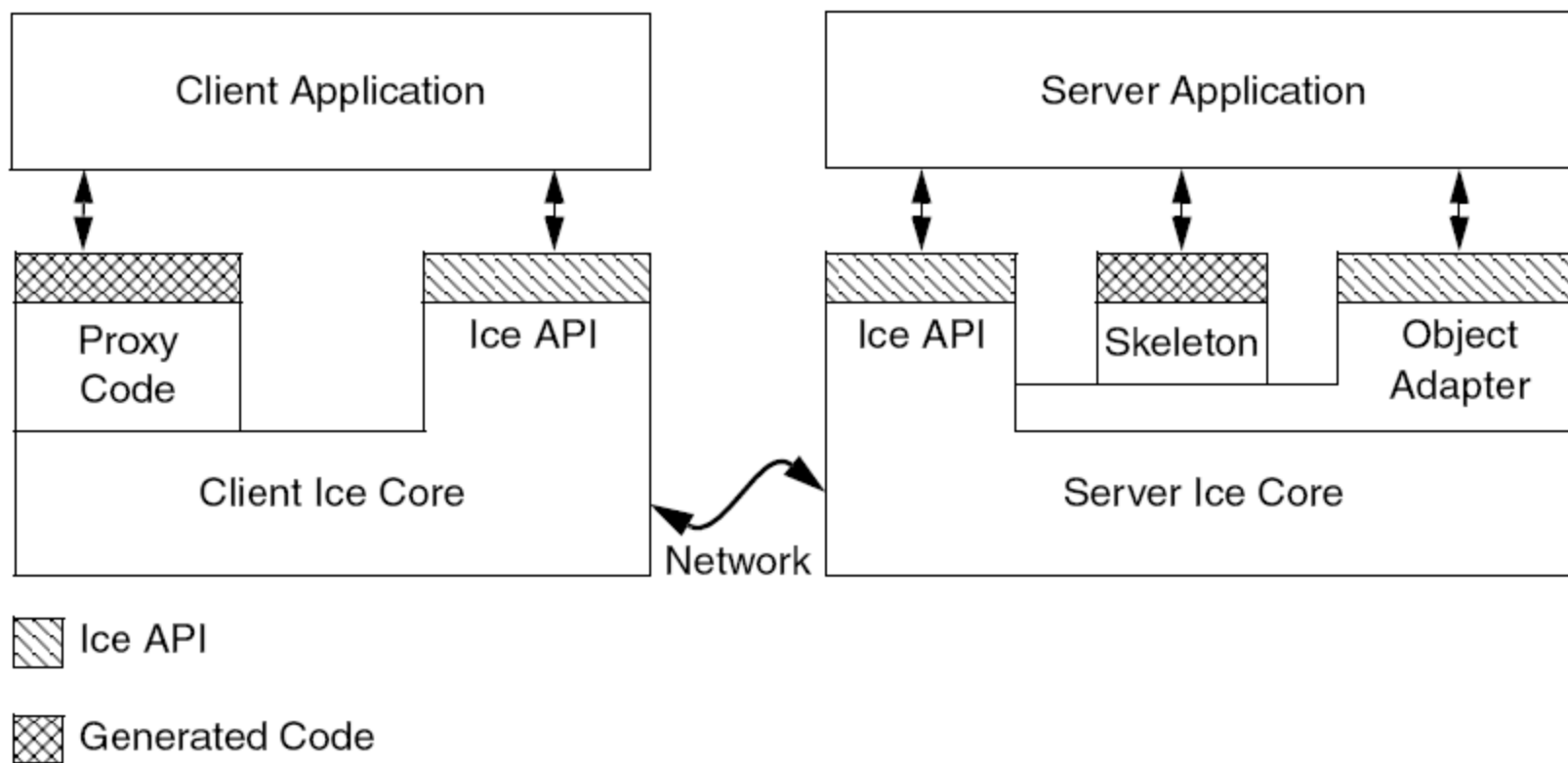
- ICE - Internet Communication Engine
- Stworzona przez ZeroC corporation
- Middleware do budowy obiektowych systemów rozproszonych
- Wsparcie dla wielu platform i języków (C++, Java, C#, Ruby, PHP, ...)
- Nacisk na **obiektość, prostotę i wydajność**
- Aktualna wersja stabilna to 3.4.2



# Dlaczego jeszcze jedna technologia?

- CORBA – wysoki poziom złożoności, niekompatybilność middleware'ów dostarczanych przez różnych providerów
- .NET/WCF – ograniczona ilość języków i platform
- RMI – tylko JAVA
- WS – duże narzuty, brak komunikacji asynchronicznej

# Architektura



# Modele wywołań

- Domyślna semantyka operacji to **at-most-once**
- Możliwość wyspecyfikowania operacji **idempotentnych**
- Domyślnie wywołania są synchroniczne, dostępnych jest także szereg modeli asynchronicznych (m.in. oneway czy datagram)

# Dodatkowe możliwości

- Bardzo dobra i ustandaryzowana obsługa wyjątków (w przeciwieństwie do CORBY)
- Mnogość modeli wywołań
- Facets – widoki obiektu (przydatne np. do wersjonowania)
- Dodatkowe usługi (Freeze, Glacier, IceGrid, IceStorm, IceBox)
- Konfigurowalność za pomocą plików .properties
- Obsługa SSL
- Automatyczne gromadzenie statystyk konsumpcji pasma (Ice:Stats)
- Wsparcie dla urządzeń mobilnych (np. Android)

# SLICE

- Język specyfikacji interfejsu
- Gotowe generatory kodu dla różnych języków (slice2java, slice2cpp)
- Zwyczajowo pliki mają rozszerzenie .ice
- Wiele usług ICE ma interfejsy SLICE



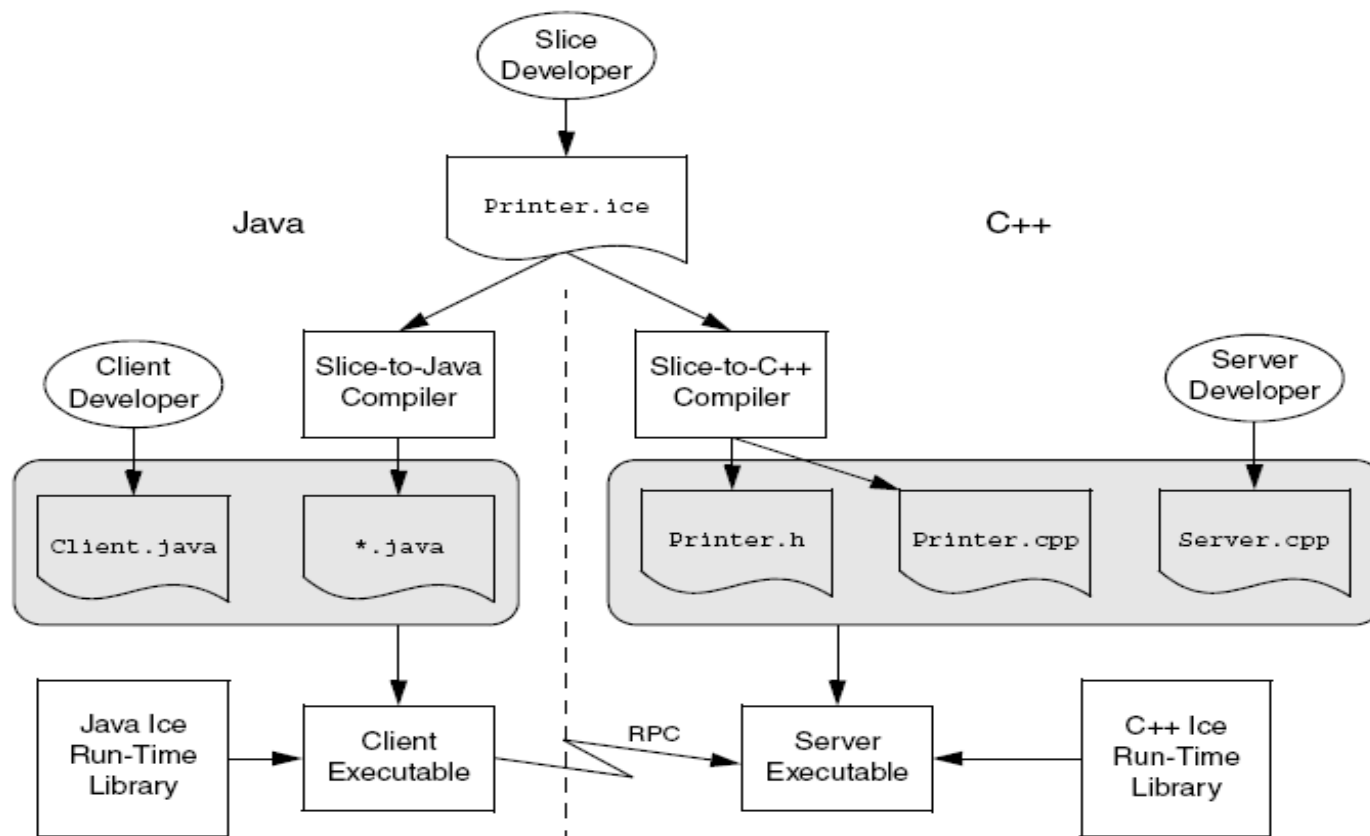
# SLICE

- Typy danych:
  - Proste: int, float, bool, double, string, ...
  - Złożone: sekwencje, wyliczenia, struktury, ...
- Interfejsy, klasy
- Wyjątki
- Operacje idempotentne
- Możliwość przekazywania obiektów przez wartość (domyślne) i referencję (np. Printer\*)
- Moduły (por. przestrzenie nazw w C++/C#)

# Tworzenie aplikacji

- Definicja interfejsów w SLICE
- Generacja kodu
- Implementacja serwantów
- Implementacja serwera
- Implementacja klienta
- Kompilacja, uruchomienie i testowanie

# Tworzenie aplikacji



Źródło: M. Henning, M. Spruiell „Distributed Programming with ICE”

# Przykład

- Proszę pobrać i rozpakować plik  
<http://galaxy.agh.edu.pl/~skalkow/dydaktyka/is-sr/SampleIceAPP.zip>
- Strona ICE: <http://www.zeroc.com/download.html>
- Proszę spróbować skompilować i uruchomić aplikacje Server i Client (dla ułatwienia projekt można zaimportować do Eclipse)
- Proszę przeanalizować zawartość plików  
FileSystem.ice/DirectoryI.java/Server.java/Client.java

# Pliki C++

- <interface>.cpp i <interface>.h
- Klient
  - <interface>Prx – proxy
  - <interface>Ptr – pointer
- Serwer
  - Implementacja klasy <interface> która jest skeletonem z metodami virtual

# Pliki Java

- `<interface>.java` – interfejs
- `<interface>Holder.java` – wrapper interfejsu holder (do przekazywania jako parametr out)
- **`<interface>Prx.java` – interfejs proxy**
- **`<interface>PrxHelper.java` – rzutowanie proxy**
- `*Operations*` – metody interfejsu `<interface>`
- **`_<interface>Disp.java` – skeleton**
- pozostałe – nieistotne

# Ćwiczenie

- Proszę dodać do pliku Slice interfejs DirectoryV2 zawierający metodę listFiles listującą **tylko pliki** z katalogu podanego jako argument
- Skompiluj plik za pomocą programu slice2java, podmień zawartość katalogu /src w przykładowej aplikacji (zachowując klasy DirectoryI, Server i Client)
- Dodaj klasę serwanta implementującego interfejs DirectoryV2

# Ćwiczenie

- W implementacji serwera dodaj do adaptera serwanta `DirectoryV2` jako facet o nazwie „`DirectoryV2`” (wskazówka: potrzebne będą metody `ObjectPrx.ice_getIdentity()` – do pobrania identity pierwszego serwanta oraz `Adapter.addFacet()` – do dodania serwanta)
- W implementacji klienta za pomocą metody statycznej `DirectoryV2PrxHelper.checkedCast(dirPrx, "DirectoryV2")` **zrzutuj proxy obiektu `Directory`** do interfejsu `DirectoryV2` i wywołaj metodę `listFiles()`



# Literatura

- Henning M, Spruiell M. – Distributed Programming with Ice, ZeroC 2003-2011
- Strona internetowa ZeroC [www.zeroc.com](http://www.zeroc.com)
- Blogi, fora dyskusyjne

# Projekt

- Projekt -  
<http://galaxy.agh.edu.pl/~skalkow/dydaktyka/is-sr/projekt.html>
- Pytania???



Dziękuję za uwagę!