

# Praktyczne zastosowanie przestrzeni krotek

Krzysztof Dębski

# Krotki

- Krotka: („point”, 12, 34)
- Antykrotka: („point”, ?x, ?y)

# Linda

- out
- in
- rd
- inp, rdp

JavaSpaces

# JavaSpaces

```
public class Message implements Entry {  
    public String content = null;  
  
    public Message() {  
    }  
}
```

# JavaSpaces

- `Message template = new Message();`  
`Message result = (Message)`  
`space.read( template, null, Long.MAX_VALUE);`
- `/* pobiera krotkę i usuwa ją z przestrzeni */`  
`space.take( ... );`
- `Message msg = new Message();`  
`msg.content = "Hello World";`  
`space.write(msg, null, Lease.FOREVER);`

# Implementacje

- Outrigger Suna
- Blitz
- GigaSpaces

Przykład....



# Ciekawostka...

## JavaSpaces (Magdalena Dukielska)

This project will provide plugins that make working with JavaSpaces in NetBeans easy -- even for beginners. These plugins will let developers concentrate on logic of the program instead of tinkering with the JavaSpaces technology itself.

(<http://www.netbeans.org/grant/>)

# Zalety

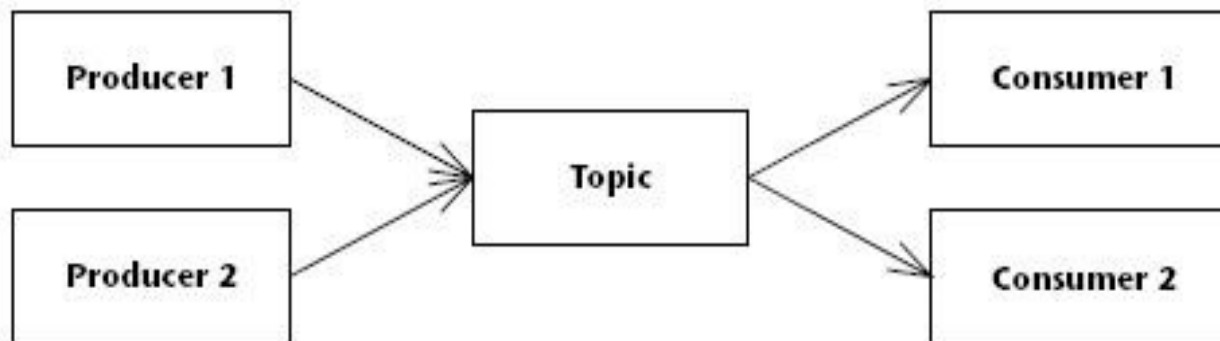
- Skalowalność
- Adaptacyjność
  - Zmienna liczba agentów
  - Czas i miejsce nie jest ważne
  - Agenci się nie znają
  - Autousuwanie wiadomości
- Odporność na błędy
  - Błędy po stronie agentów
  - Problemy z siecią
  - Transakcje

# Wady

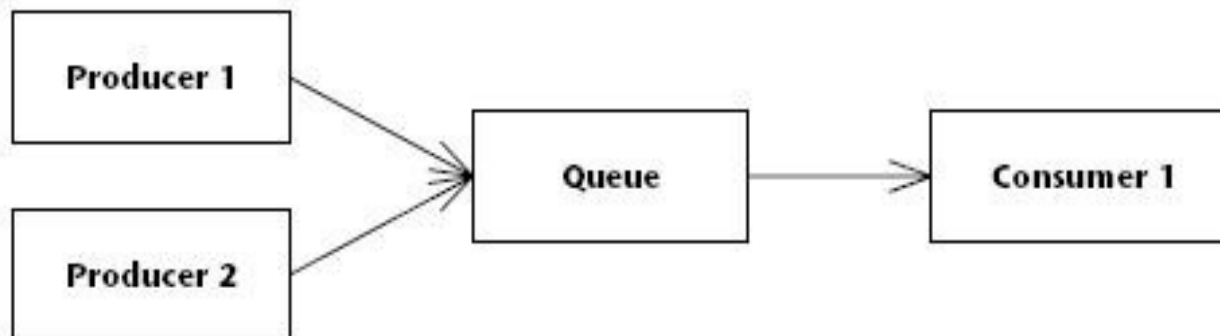
- Ewentualna awaria przestrzeni
- Problem z zasobami

**JMS**

Publish/Subscribe:



Point-to-Point:



Kolejki i tematy w JMS.

Źródło: [http://www.mimuw.edu.pl/~sroka/dydaktyka/jee/wyklady/java\\_ee\\_16.pdf](http://www.mimuw.edu.pl/~sroka/dydaktyka/jee/wyklady/java_ee_16.pdf)

```
public class Producent {
public static void main(String[] args) throws Exception {
    InitialContext ctx = (...);
    TopicConnectionFactory factory = (TopicConnectionFactory)
        ctx.lookup("fabryka");
    TopicConnection connection =
        factory.createTopicConnection();
    TopicSession session =
        connection.createTopicSession(false,
            Session.AUTO_ACKNOWLEDGE);
    Topic topic = (Topic)ctx.lookup("temat");
    TopicPublisher publisher =
        session.createPublisher(topic);
    TextMessage msg = session.createTextMessage();
    msg.setText("Komunikat");
    publisher.send(msg);
    publisher.close();
    connection.close();
}
}
```

```
TopicSubscriber subscriber =  
    session.createSubscriber(topic);  
subscriber.setMessageListener(new Consumer());  
(...)
```

```
class Consumer implements MessageListener {  
public void onMessage(Message message) {  
    try {  
        TextMessage textMessage = (TextMessage) message;  
        String text = textMessage.getText();  
        System.out.println(text);  
    } catch (JMSEException jmse) {  
        jmse.printStackTrace();  
    }  
}  
}  
}
```

- ➔ ByteMessage
- ➔ ObjectMessage
- ➔ TextMessage
- ➔ StreamMessage
- ➔ MapMessage



# MDB

```
@MessageDriven(activationConfig =
{ @ActivationConfigProperty
(
    propertyName = "destinationType",
    propertyValue = "javax.jms.Topic")
} )
public class MyMDB implements MessageListener {

public void onMessage(Message msg) {
    if (msg instanceof TextMessage) {
        TextMessage tm = (TextMessage) msg;
        try {
            String text = tm.getText();
            System.out.println(text);
        } catch (JMSEException e) {}
    }
}
```

# Selektory

- Identyfikatory
- AND, OR, LIKE, BETWEEN, =, <>, <, >, <=, >=, IS NULL, IS NOT NULL
- Stałe
  - TRUE, FALSE
  - Dokładne: +25, -399, 40
  - Aproksymacje: -21.4E4, 5E2, +34.4928

Np. NewsType = 'Opinion'

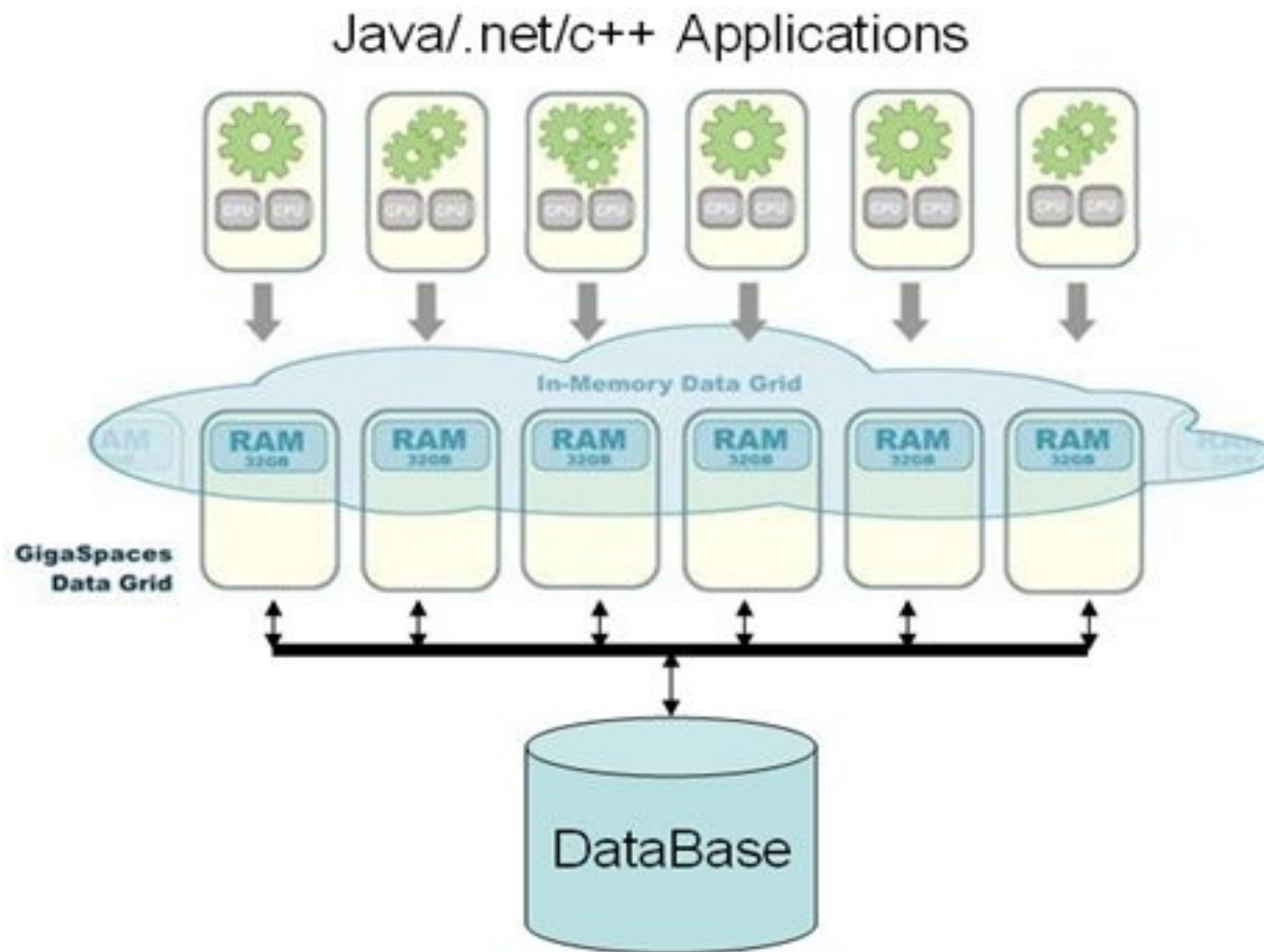
OR NewsType = 'Sports'

# Giga Spaces

# Wersje

- GigaSpaces XAP  
JavaSpaces, JMS, Spring, Hibernate, .Net, C++, Groovy, JRuby
- GigaSpaces EDG
- GigaSpaces Community Edition

# GigaSpaces EDG



Architektura EDG.

Źródło: <http://www.gigaspaces.com/edg>

```
using System;
using System.Collections.Generic;
using System.Text;
using GigaSpaces.Core;

namespace myHelloSpace
{
    class Program
    {
        static void Main(string[] args)
        {
            ISpaceProxy space =
SpaceProxyProviderFactory.Instance.FindSpace("rmi://localhost/./mySpace");
            MyData data = new MyData();
            data.firstName = "Joe";
            data.lastName = "lastname";
            data.dob = new DateTime(1980, 1, 20, 10, 20, 0, DateTimeKind.Utc);
            space.Write<MyData>(data, null, 1000 * 60 * 24);
            MyData template = new MyData();
            template.firstName = "Joe";
            template.dob = new DateTime(1900, 1, 1, 12, 0, 0, DateTimeKind.Utc);
            MyData dataFromSpace = space.Read<MyData>(template, null, 1000);
        }
    }
}
```

# Bibliografia

- [http://en.wikipedia.org/wiki/Linda\\_\(coordination\\_language\)](http://en.wikipedia.org/wiki/Linda_(coordination_language))
- [http://en.wikipedia.org/wiki/Tuple\\_spaces](http://en.wikipedia.org/wiki/Tuple_spaces)
- <http://www.jini.org/>
- <http://www.dancre.org/blitz/>
- <http://www.gigaspace.com/>
- <http://www.netbeans.org/grant/>
- <http://www.mimuw.edu.pl/~sroka/dydaktyka/dydaktyka.html>