KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

POLITECHNIKA BIAŁOSTOCKA

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Rozproszone systemy internetowe 2

WS-Reliable Messaging

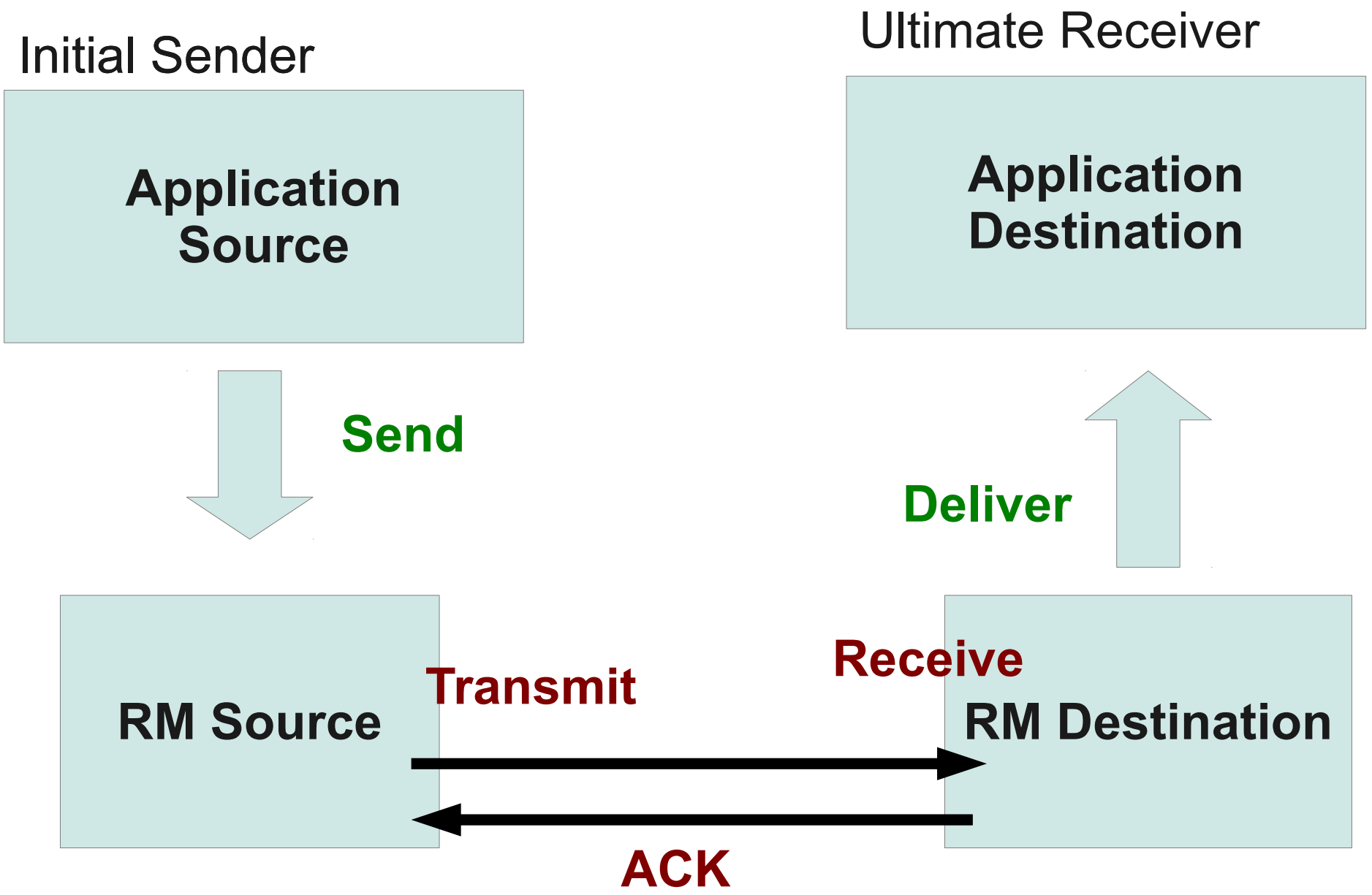
„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”

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

Wstęp

- Standard OASIS [**Organization for the Advancement of Structured Information Standards**]
- Elementy składowe systemu „Reliable Messaging”:
  - Sekwencjonowanie komunikatów
  - Semantyka dostarczania wiadomości
  - Potwierdzenie dotarcia komunikatu
- Czy rzeczywiście potrzebujemy takiego mechanizmu ? - dyskusja
  - Implementacje na poziomie logiki biznesowej
  - Integracja systemów

Model


Initial Sender
Ultimate Receiver
Application Source
Application Destination
Send
Deliver
RM Source
Transmit
Receive
RM Destination
ACK

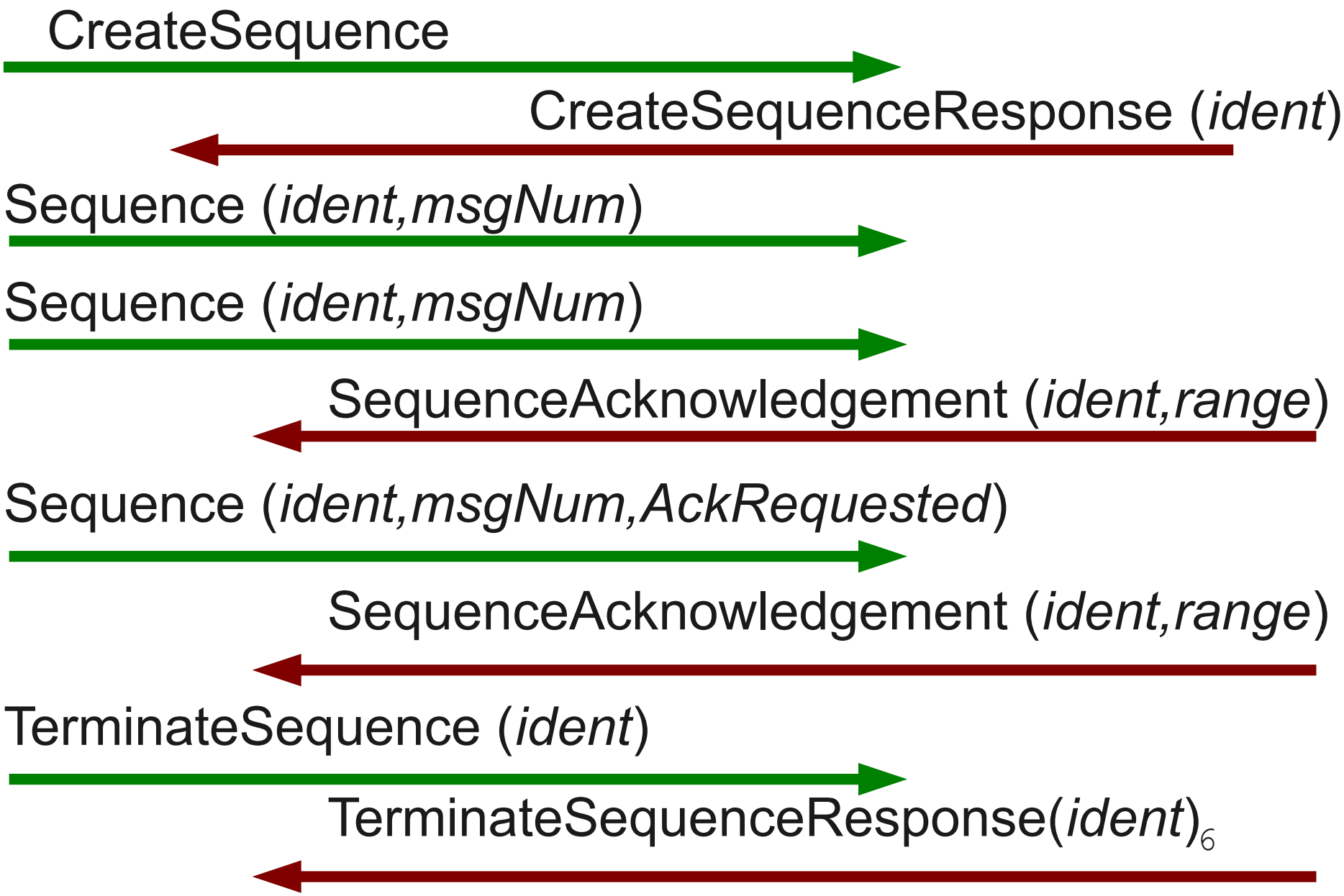
Oferta WS-RM

- **AtLeastOnce** – RMS wysyła komunikaty dopóki nie otrzyma potwierdzenia, RMD przekazuje te komunikaty do aplikacji
- **AtMostOnce** – RMS może wysyłać duplikaty, RMD filtruje je
- **ExactlyOnce** – RMS musi wysyłac duplikaty, RMD musi przekazywać je do aplikacji dopóki nie stwierdzono poprawnego dostarczenia.
- **InOrder** – komunikaty dostarczane są dokładnie w takiej kolejności w jakiej są wysyłane przez aplikację.

WS-RM a inne standardy

- Komunikat generowany przez usługę musi określać element wsa:Action jako: *http://docs.oasis-open.org/ws-rx/wsrm/200702/<nazwa dziecka elementu Body>*
- Komunikaty, które nie zawierają treści (CreateSequence, SequenceAcknowledgement, AckRequested) - zamiast elementu „dziecka” wstawiamy nazwę operacji
- Inne: mozliwość przenoszenia nagłówków WS-RM w innych komunikatach skierowanych do właściwych punktów dostępu (piggy-backing)

Schemat komunikacji



Oskar Świda

Tworzenie sekwencji 1

```
<wsrm:CreateSequence>
 <wsrm:AcksTo>                                  </wsrm:AcksTo>
 <wsrm:Expires>                                 </wsrm:Expires>
 <wsrm:Offer>
  <wsrm:Identifier>                             </wsrm:Identifier>
  <wsrm:Endpoint>                               wsrm:Endpoint>
  <wsrm:Expires> </wsrm:Expires>
  <wsrm:IncompleteSequenceBehavior >
  </wsrm:IncompleteSequenceBehavior >
 </wsrm:Offer>
</wsrm:CreateSequence>
```

EPR dla potwierdzeń

Czas aktywności sekwencji (PT0S)

Proponowana sekwencja zwrotna

Identyfikator

EPR sekwencji zwrotnej

DiscardEntireSequence, DiscardFollowingFirstGap, NoDiscard

Tworzenie sekwencji 2

```
<wsrm:CreateSequenceResponse>
   <wsrm:Identifier>
   </wsrm:Identifier>

   <wsrm:Expires> </wsrm:Expires>
   <wsrm:IncompleteSequenceBehavior >
   </wsrm:IncompleteSequenceBehavior >
    <wsrm:Accept>
      <wsrm:AcksTo />
   </wsrm:Accept>
</wsrm:CreateSequenceResponse>
```

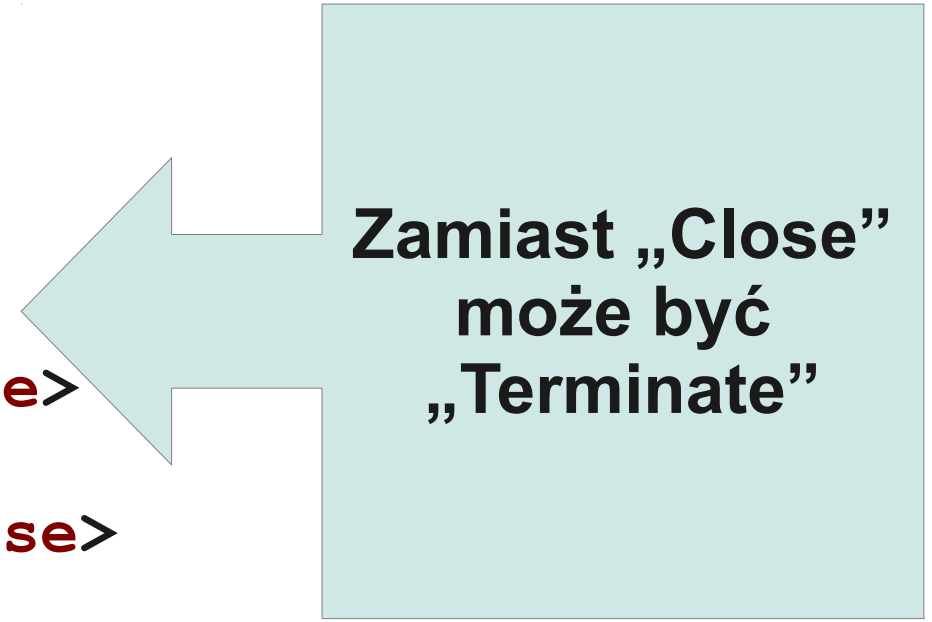
Identyfikator
definiowany przez Destination

Akceptacja sekwencji zwrotnej

Zamknięcie i przerwanie

```
<wsrm:CloseSequence>
  <wsrm:Identifier />
  <wsrm:LastMsgNumber />
</wsrm:CloseSequence>

<wsrm:CloseSequenceResponse>
  <wsrm:Identifier />
</wsrm:CloseSequenceResponse>
```



- **Zamknięcie (Close)** – wymaga uzgodnienia stanu po obuu stronach
- **LastMsgNumber** - pomaga wykonać odpowiednie procedury dla niekompletnej sekwencji

Przekazywanie komunikatów

```
<wsrm:Sequence>
  <wsrm:Identifier />
  <wsrm:MessageNumber />
</wsrm:Sequence>
```

Komunikaty numerowane są od 1

```
<wsrm:AckRequested>
  <wsrm:Identifier />
</wsrm:AckRequested>
```

Źródło zawsze może zażądać potwierdzenia

```
<wsrm:SequenceAcknowledgement>
  <wsrm:Identifier />
  <wsrm:AcknowledgementRange Upper="" Lower="" />
  <wsrm:None /> <!-- Nie otrzymano żadnych -->
  <wsrm:Final /> <!-- Nie odbieramy więcej -->
  <wsrm:Nack /> <!-- Brakuje nam -->
</wsrm:SequenceAcknowledgement>
```

Bezpieczeństwo

- WSRM nie definiuje wymagań dot. bezpieczeństwa ale dodanie tego mechanizmu nie powinno kreować dodatkowych możliwości ataku
- Retransmisje komunikatów vs ataki typu „replay”
- Integralność komunikatu może być warunkiem utrzymania kolejności
- Zagrożenia typu DoS – kontekst sekwencji zużywa zasoby – warto weryfikować tożsamości rozmówców
- Weryfikacja tożsamości zabezpiecza również przed przejęciem lub zerwaniem komunikacji

Przykład: implementacja Apache CXF

```
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:cxf="http://cxf.apache.org/core"
      xmlns:wsa="http://cxf.apache.org/ws/addressing"
      xmlns:http="http://cxf.apache.org/transports/http/configuration"
      xmlns:wsrm-policy="http://schemas.xmlsoap.org/ws/2005/02/rm/policy"
      xmlns:wsrm-mgr="http://cxf.apache.org/ws/rm/manager">

    <cxf:bus>
        <cxf:features>
            <cxf:logging/>
            <wsa:addressing/>
            <wsrm-mgr:reliableMessaging>
                <wsrm-policy:RMAssertion>
                    <wsrm-policy:BaseRetransmissionInterval Milliseconds="4000"/>
                    <wsrm-policy:AcknowledgementInterval Milliseconds="2000"/>
                </wsrm-policy:RMAssertion>
                <wsrm-mgr:destinationPolicy>
                    <wsrm-mgr:acksPolicy intraMessageThreshold="0" />
                </wsrm-mgr:destinationPolicy>
            </wsrm-mgr:reliableMessaging>
        </cxf:features>
    </cxf:bus>

</beans>
```