



Servicios Web

Sesión 4: Procesos BPEL
síncronos y asíncronos



Puntos a tratar

- Procesos BPEL síncronos
- Procesos BPEL asíncronos
- Ciclo de vida de los procesos BPEL
- *Correlations*
- Conjuntos de correlación
- Pasos a seguir para usar correlación en BPEL
- Conjuntos de correlación con *Netbeans*



Invocación de servicios Web

- Invocación SECUENCIAL
- Invocación CONCURRENTE

```
<process...>
...
<sequence>
  <!-- espera una petición-->
  <receive.../>

  <!--invocación secuencial -->
  <invoke.../>
  <invoke.../>
  <invoke.../>

  ...
</sequence>
</process>
```

```
<process...>
...
<sequence>
  <!-- espera una petición-->
  <receive.../>

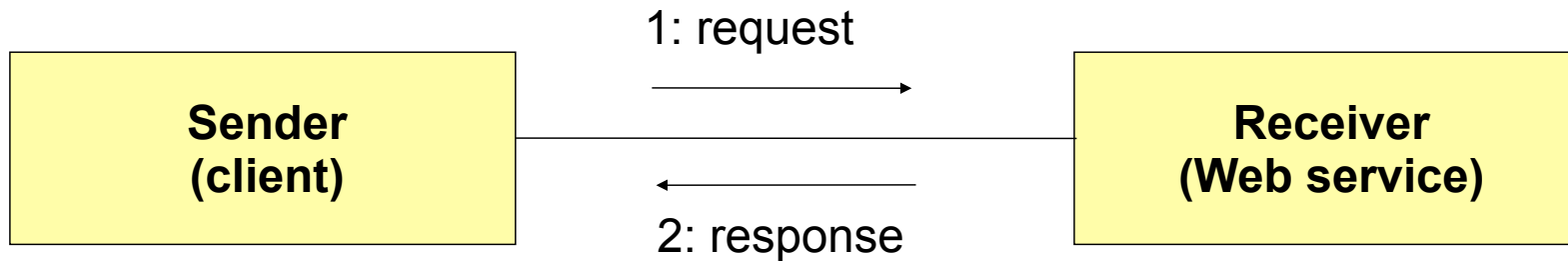
  <!--invocación concurrente -->
  <flow>
    <invoke.../>
    <invoke.../>
    <invoke.../>
  </flow>

  ...
</sequence>
</process>
```

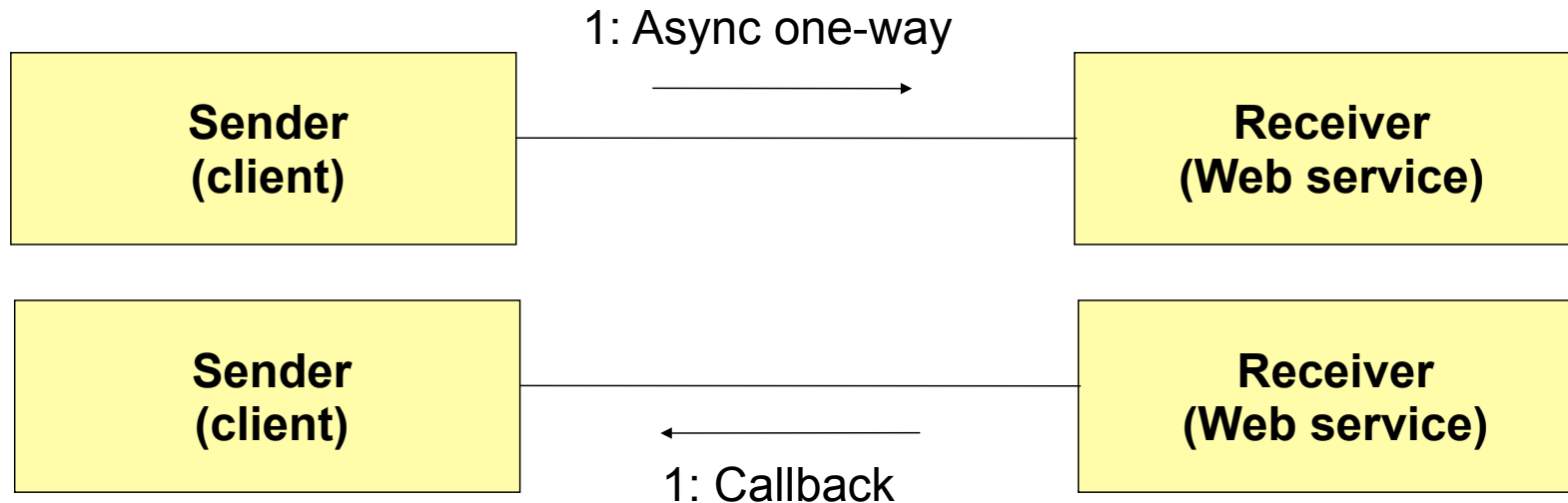


Invocación síncrona vs. asíncrona (I)

- Invocación síncrona (Operación **request-reply**)

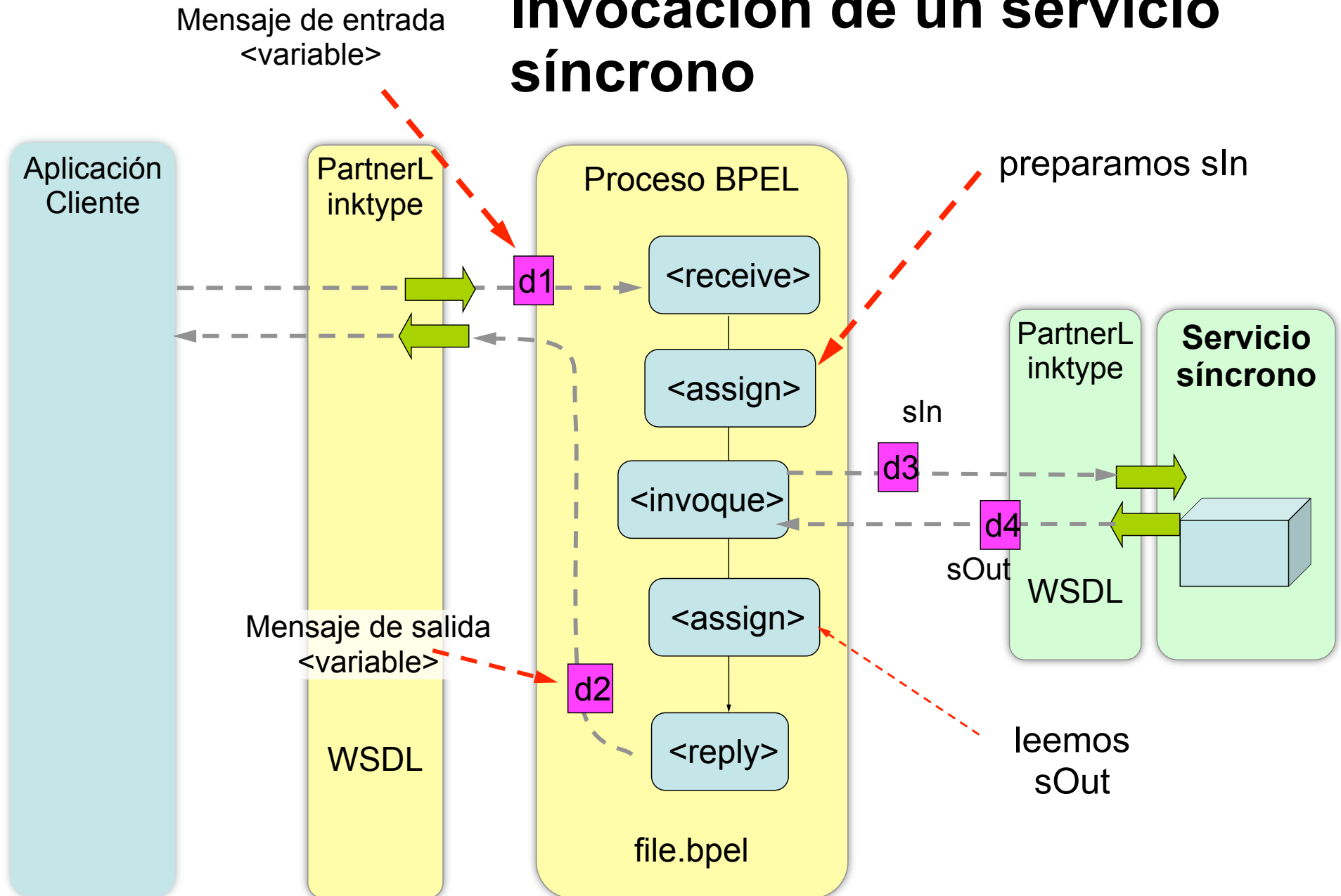


- Invocación asíncrona (Operación **one-way**)



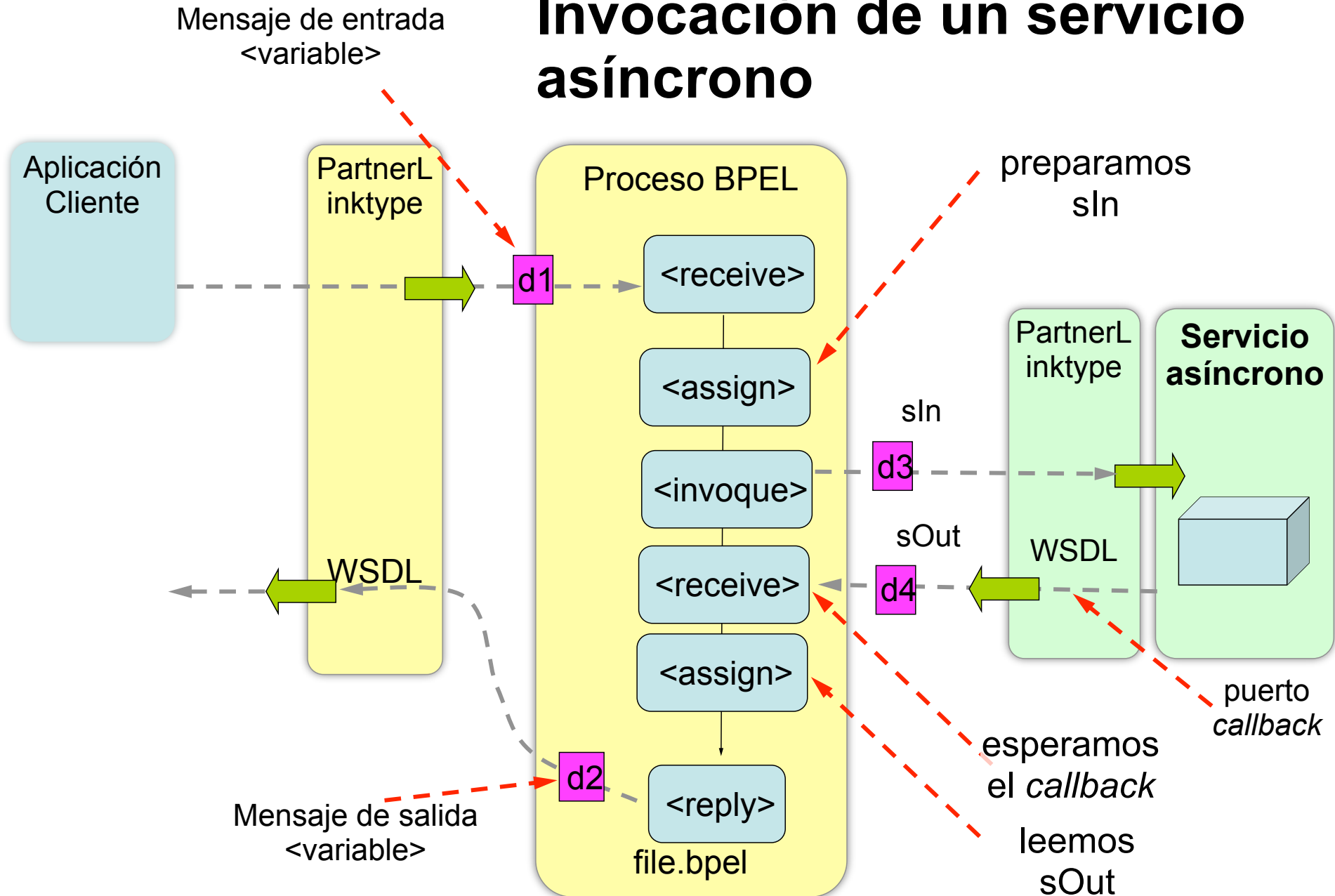


Invocación de un servicio síncrono





Invocación de un servicio asíncrono





Proceso BPEL síncrono (II)

```
<!-- Estructura de un proceso BPEL síncrono -->
<process>
  <receive partnerLink="pLinkA" portType="portA"
    operation="opA" variable="invarA">

    ..... realizar procesamiento .....

  <reply partnerLink="pLinkA" portType="portA"
    operation="opA" variable="outvarB">
</process>
```

- La respuesta se devuelve al cliente de forma inmediata
- Operación tipo **request-response**



Proceso BPEL asíncrono (II)

```
<!-- Estructura de un proceso BPEL asíncrono -->
<process>
  <receive partnerLink="pLinkA" portType="portA" operation="opA"
    variable="invarA">

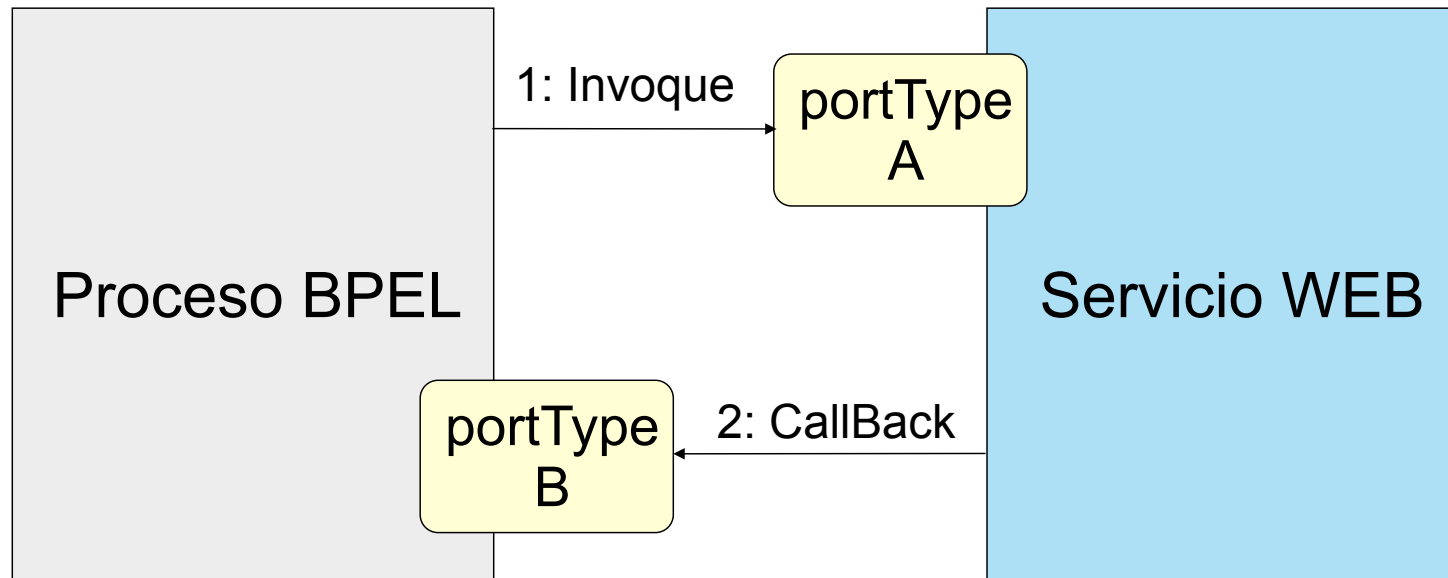
    ..... realizar procesamiento que consume mucho tiempo .....
    <!-- Realizar una invocación sobre el cliente para devolver
      los resultados -->
    <invoque partnerLink="pLinkA" portType="callbackportA"
      operation="opCallbackB" variable="outvarB">
  </process>
```

- El cliente no se queda bloqueado en la llamada
- El cliente implementa una interfaz *callback*
- El proceso BPEL invoca a la interfaz *callback* sobre el cliente. Operación tipo **one-way**



Partner Links en procesos asíncronos

- Los *partner links* definen las relaciones con los *partners*:
 - el *partner* invoca al proceso BPEL (2)
 - el *partner* es invocado por el proceso BPEL (1)
 - el *partner* invoca y es invocado por el proceso BPEL (2-1)





***Partner Link Types* en procesos asíncronos**

- Los *partner link types* permiten modelar las relaciones entre un servicio Web y un proceso BPEL mediante el uso de roles
 - Cada *partner link type* debe tener como mucho dos roles
 - Para cada rol se especifica qué *portType* se utilizará para la interacción
 - Un proceso asíncrono implementará dos roles, definidos en un *PartnerLinkType*



Ciclo de vida de los procesos BPEL

- Para procesos de negocio, es necesario utilizar procesos con estado
 - Los procesos BPEL siguen un modelo con estado y soportan interacciones que llevan mucho tiempo
- Para cada interacción, se crea una instancia del proceso
 - La creación se realiza de forma implícita cuando el proceso BPEL recibe el mensaje inicial (a través de actividades *receive* o *pick*, con el atributo *createInstance=yes*)
- Esta instancia, "vive" durante la ejecución del proceso de negocio



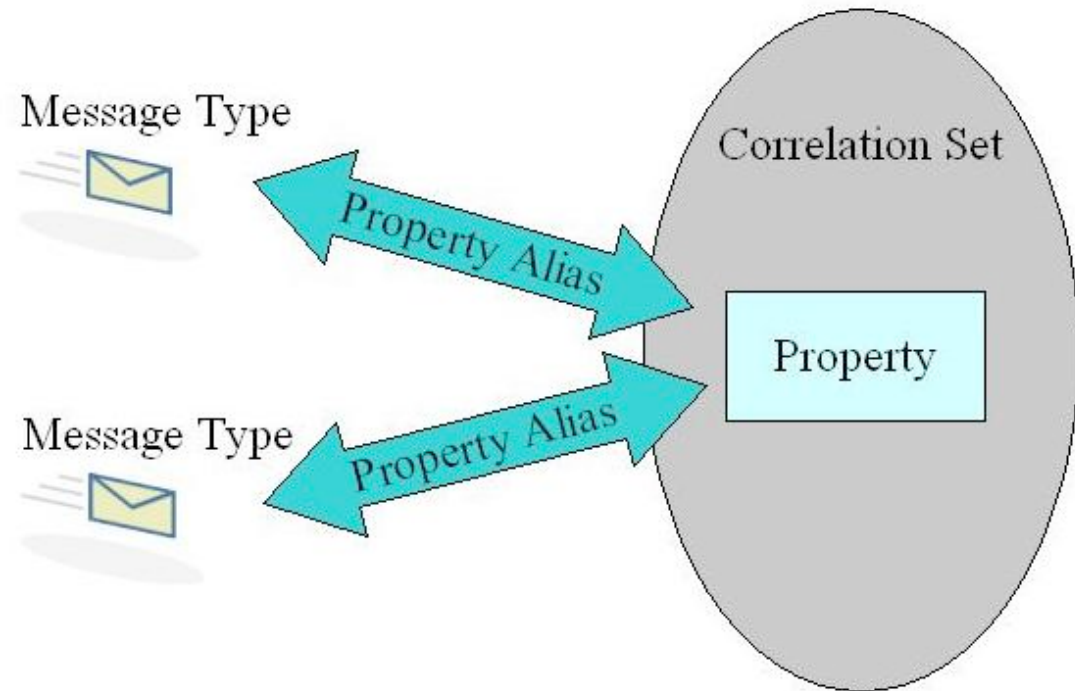
Correlación

- Es el proceso de hacer corresponder un mensaje de entrada a la máquina BPEL con un proceso específico
- Las llamadas síncronas NO necesitan correlación
- BPEL utiliza datos específicos del negocio para mantener referencias a instancias específicas del proceso de negocio
 - Conjuntos de correlación



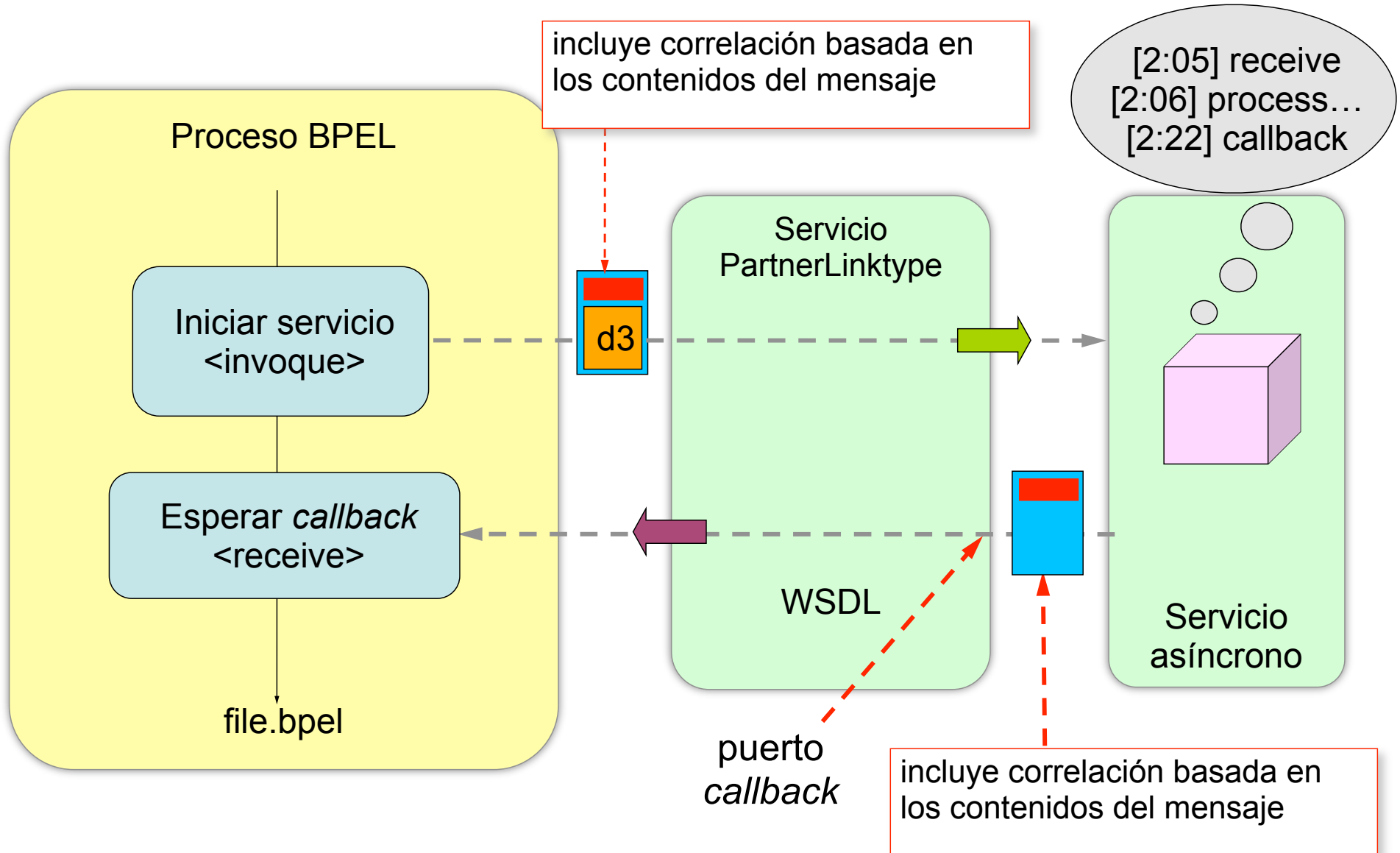
Conjuntos de correlación

- Un conjunto de correlación está formado por **propiedades** (concepto abstracto)
- Cada propiedad es mapeada con un elemento en uno o más tipos de mensajes mediante **property aliases** (aspecto concreto)





Uso de correlación





Pasos a seguir para usar correlación (I)

- Definimos una propiedad en nuestro WSDL

```
<bpws:property name="correlationData" type="xsd:int"/>
```

CONCEPTO ABSTRACTO

- Definimos una *propertyAlias* para cada elemento de los datos de la correlación

```
<bpws:propertyAlias messageType="CoreBankingP2P:CallbackType"  
    part="accountNumber"  
    propertyName="CoreBankingP2P:correlationData"/>  
<bpws:propertyAlias messageType="CoreBankingP2P:ServiceRequestType"  
    part="accountNumber"  
    propertyName="CoreBankingP2P:correlationData"/>
```

Relacionamos el concepto abstracto con “partes” CONCRETAS de los mensajes mediante ALIAS



Pasos a seguir para usar correlación (II)

- Definimos el conjunto de correlación en el fichero BPEL (antes de cualquier secuencia de actividades)

```
<correlationSets>  
  <correlationSet name="CS1"  
    properties="CoreBankingP2P:correlationData"/> </  
correlationSets>
```

- Referenciamos el conjunto de correlación dentro de la secuencia BPEL



Pasos a seguir para usar correlación (II)

```
<!-- - secuencia BPEL: referencia a conjuntos de correlación - ->
.....
<receive name="receiveRequest" partnerLink="Client"
  portType="CoreBankingP2P:CoreBankingP2PPortType"
  operation="applyForCC" variable="ServiceRequest"
  createInstance="yes">
  <correlations>
    <correlation initiate="yes" set="CS1"/>
  </correlations>
</receive>
.....
<receive name="P2Pcallback" partnerLink="CoreBankingP2PPLT"
  portType="CoreBankingP2P:CoreBankingCallbackP2PPortType"
  operation="callback" variable="CallbackRequest">
  <correlations> <correlation set="CS1"/> </correlations>
</receive>
```



Conjuntos de correlación con Netbeans

- Definir las propiedades y alias en los ficheros WSDL
 - Utilizamos la vista del Navegador: *Windows > Navigator*
- Definimos los conjuntos de correlación en el proceso BPEL
 - Utilizamos la vista de diseño del proceso BPEL
- Añadimos el conjunto de correlación a las actividades
 - Utilizamos la vista de diseño del proceso BPEL



¿Preguntas...?