(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 426 192**

(51) Int. CI.:

H04L 29/08 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **30.11.2006 E 06301199 (3)**

(97) Fecha y número de publicación de la concesión europea: **29.05.2013 EP 1928152**

(54) Título: **Procedimiento de comunicación entre un dispositivo que ejecuta Java ME y un servidor por vía aérea con mensajes SOAP bajo APDU desde/hacia un operador en un anfitrión, y sistema correspondiente**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.10.2013

(73) Titular/es:

CASSIS INTERNATIONAL PTE LTD. (100.0%)
51 BRAS BASAH ROAD NO. 08-07/08 PLAZA BY THE PARK
SINGAPORE 189554, SG

(72) Inventor/es:

NG, CHEE WEI

(74) Agente/Representante:

CURELL AGUILÁ, Mireia

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Procedimiento de comunicación entre un dispositivo que ejecuta Java ME y un servidor por vía aérea con mensajes SOAP bajo APDU desde/hacia un operador en un anfitrión, y sistema correspondiente.

La presente invención define un nuevo procedimiento de comunicaciones con un protocolo que permite una fragmentación adaptativa de paquetes de datos, un formato de mensaje optimizado y funcionalidades de autorecuperación para una aplicación Java ME® (JME) con el fin de comunicarse con un servidor de aplicaciones a través del aire por medio de HTTP o HTTPS. El servidor de aplicaciones, el cual puede entender el protocolo, realiza una traducción a otro protocolo de mensajes y lo encamina hacia el anfitrión de punto extremo. El mensaje de punto extremo puede ser un mensaje XML privativo o del Protocolo Simple de Acceso a Objetos (SOAP) de normativa abierta. El protocolo proporciona autorecuperación al volver a enviar el último mensaje una y otra vez cuando no se recibe ningún mensaje de respuesta dentro de un cierto periodo de tiempo. Encuentra aplicación especialmente en las industrias de comunicación. Se describe también un sistema.

Cada vez se están usando más dispositivos portátiles con tarjetas inteligentes. Dichos dispositivos están destinados a comunicarse con operadores en un anfitrión a través de una red. La comunicación entre el dispositivo de la red se efectúa a través de un servidor al que se denomina generalmente pasarela de canal. Algunos de dichos dispositivos ejecutan una aplicación Java ME® y esto se lleva a cabo habitualmente con recursos limitados en términos de memoria y poder de procesado. Por otra parte, la tecnología de comunicaciones usada en el dispositivo es lenta y posiblemente no fiable con una velocidad, siendo realista, de hasta 56 Kbps.

A medida que el servidor de aplicaciones cambia hacia una arquitectura orientada a servicios web, el protocolo de mensajería por defecto para comunicarse con el servidor es un mensaje SOAP a través de HTTP o HTTPS. No obstante, el análisis sintáctico y la serialización del mensaje SOAP requiere un procesado y una memoria mucho mayores que un mensaje binario. Por otra parte, el tamaño del mensaje es mucho mayor que el de un mensaje binario.

Tal como se ha mencionado anteriormente, un sistema de gestión de tarjetas inteligentes SOA encapsula mensajes APDU en SOAP y realiza transmisiones a través de una red de cable o inalámbrica. Para una aplicación Java ME que se ejecuta en un dispositivo móvil con conexión inalámbrica GPRS® o 3G® con el fin de comunicarse con el sistema, el dispositivo requiere una memoria mayor para el almacenamiento temporal de SOAP y un procesador más rápido para analizar sintácticamente y extraer los datos APDU fuera del mensaje SOAP. Los datos transmitidos por vía aérea se podrían limitar en cuanto a tamaño según la conexión por parte del operador móvil.

Por tanto, existe un límite sobre la magnitud de la transferencia de datos controlada en la pasarela del operador móvil. Es necesario que los datos de los paquetes intercambiados entre el dispositivo y el servidor se dividan en paquetes de menor tamaño el cual sea inferior a la magnitud del límite de manera que el paquete se pueda transmitir uniformemente a través de la red del operador. El paquete resultará determinado por la ubicación de la célula en la cual se registró el dispositivo.

Por ello, para superar esto, se define un ajuste adaptativo bajo demanda, del tamaño de los datos según la conexión.

El transporte de datos APDU en el tiempo es una conexión fiable no garantizada. Junto a ello, la JVM subyacente que se ejecuta en el dispositivo puede no realizar una recolección correcta de basura sobre los recursos dentro de un cierto periodo de tiempo, y por lo tanto puede derivar en el bloqueo de la subsiguiente operación de conexión abierta.

El documento WO-2006/087438 trata sobre un método y un dispositivo para acceder a una tarjeta SIM alojada en un terminal móvil por medio de una pasarela doméstica. La comunicación se efectúa a través de un enlace inalámbrico y especialmente del Bluetooth®.

La presente invención está destinada a superar dichos problemas y a proporcionar ventajas que se pondrán de manifiesto a partir de la descripción.

La invención trata sobre un procedimiento de comunicación por medio de HTTP o HTTPS entre un dispositivo que ejecuta Java ME® y un servidor por vía aérea, recibiendo y transmitiendo dicho servidor mensajes SOAP (Protocolo Simple de Acceso a Objetos) desde/hacia un operador en un anfitrión a través de una red y teniendo la capacidad de intercambiar mensajes SOAP bajo un formato de datos de Unidad de Datos de Protocolo de Aplicación (APDU) con el dispositivo.

Según la invención, los mensajes SOAP se traducen a partir de/en mensajes binarios según un protocolo en el servidor, intercambiándose dichos mensajes binarios con el dispositivo,

y en caso de que el mensaje no pueda ser contenido en un mensaje binario, el intercambio se efectúa en una transmisión por lotes de una pluralidad de mensajes binarios,

y para que se ejecute una comunicación según el protocolo, se implementan las siguientes etapas:

- el dispositivo en primer lugar envía al servidor un mensaje binario que incluye una solicitud de intercambio APDU e información de inicialización de servicio para identificar un operador,

- como respuesta, el servidor envía de vuelta al dispositivo un mensaje binario de un lote de órdenes APDU con detalles fragmentados en un Segmento de Información, a continuación:

- si dicho lote de órdenes APDU no está fragmentado, el dispositivo envía un mensaje binario que incluye un resultado de ejecución de dicho lote de órdenes APDU para completar la transacción correspondiente a dicho lote,

- si dicho lote de órdenes APDU está fragmentado, el dispositivo envía una solicitud de más fragmento con la siguiente secuencia de fragmentos y espera por un mensaje binario del servidor para el siguiente fragmento y, cuando se reciben todos los fragmentos, el dispositivo ejecuta dicho lote,

y, cuando la transacción se ha completado para dicho lote, el dispositivo espera por un mensaje binario del servidor en relación con un posible lote sucesivo.

De forma más precisa, la presente invención se refiere a un procedimiento según la reivindicación 1 y a un sistema según la reivindicación 10.

En la invención se consideran también los siguientes medios, posiblemente usados en todas las combinaciones técnicas posibles:

- los mensajes binarios son mensajes de solicitud binarios y mensaje de respuesta binario,

- el mensaje de solicitud binario tiene una parte de segmento de encabezamiento de siete bytes, una parte de segmento de cuerpo de n bytes y una parte de segmento de información de cinto bytes,

- el mensaje de respuesta binario tiene una parte de segmento de encabezamiento de siete bytes, una parte de segmento de cuerpo de n bytes y una parte de segmento de información de tres bytes,

- los mensajes binarios son de acuerdo con las tablas descritas en este documento,

- la parte de segmento de cuerpo de los mensajes binarios es fragmentable,

- se puede solicitar más fragmento con un mensaje de solicitud binario de más fragmento con una parte de segmento de encabezamiento de dos bytes y una parte de segmento de cuerpo de un byte, siendo la respuesta de la solicitud de más fragmento un mensaje de respuesta binario para más fragmento con una parte de segmento de encabezamiento de dos bytes y una parte de segmento de cuerpo de un byte,

- se implementan unos medios de recuperación y de autoreintento, memorizándose mensajes antes de ser enviados en puntos de memorización predefinidos del procedimiento,

- los puntos de memorización se producen, en el servidor, cada vez que el servidor envía al dispositivo un mensaje binario que incluye órdenes APDU, y, en el dispositivo, cada vez que el dispositivo envía al servidor un mensaje binario que incluye un resultado de ejecución de órdenes APDU,

- se inicia un contador de tiempo en el dispositivo cada vez que el dispositivo envía un mensaje de resultado de ejecución de órdenes APDU y, si el servidor no reacciona después de un recuento de tiempo predeterminado, el dispositivo ejecuta un reintento volviendo a enviar el mensaje desde el punto de memorización relacionado,

- se autoriza un número predeterminado de reintento para cada punto de memorización antes de abortar el procedimiento,

- en caso de que se aborte el procedimiento, el código de respuesta es un código de error y, en otro caso, el código de respuesta es un código de éxito.

También forma parte de la invención un sistema, el cual dispone de medios para la ejecución del procedimiento en una o más de sus acciones descritas. Por otra parte, el sistema está constituido por entidades físicas individuales (por ejemplo, un dispositivo que ejecuta JavaME®, un anfitrión...) y cada una de dichas entidades que dispone de medios para funcionar de acuerdo con el procedimiento forma también parte de la invención.

A partir de la invención, es posible una utilización completa de mensajes binarios como un protocolo para una comunicación efectiva entre el dispositivo limitado, tal como un teléfono móvil, y un sistema de gestión de tarjetas inteligentes SOA. La pasarela proporciona un canal optimizado y un traductor para el intercambio de mensajes APDU. La invención se apoya en la fragmentación adaptativa de datos para prestar servicio a múltiples operadores móviles sin volver a desarrollar la aplicación con el fin de gestionar la limitación de transferencia de datos controlada por el operador.

Adicionalmente, con la introducción de puntos de memorización tanto en la aplicación como en la pasarela, la aplicación reside en el dispositivo limitado y puede volver a enviar de forma eficaz el último mensaje de solicitud a la pasarela, y la pasarela podrá recuperar la respuesta correctamente.

A continuación se describirá la invención con la ayuda de detalles, sin limitarse a estos últimos, y en relación con:

la Figura 1, que ilustra el procedimiento de comunicaciones entre la aplicación Java ME® y la pasarela en el caso de la fragmentación de un lote de resultado de ejecución de APDU desde la aplicación Java ME a la pasarela,

la Figura 2, que ilustra el procedimiento de comunicaciones entre la aplicación Java ME® y la pasarela en el caso de la fragmentación de respuesta de órdenes APDU de vuelta desde la pasarela,

la Figura 3, que ilustra el uso de puntos de memorización en el procedimiento para gestionar la capacidad de autorecuperación.

A continuación se describe el procedimiento de la invención en forma de un nuevo protocolo de comunicaciones con un tamaño adaptativo de paquetes de datos, un formato de mensaje optimizado y funcionalidades de autorecuperación para una aplicación Java ME® con el fin de comunicarse con un servidor de aplicación por vía aérea por medio de HTTP o HTTPS.

La solicitud se inicia habitualmente desde una aplicación Java ME y, cuando el sistema de gestión de tarjetas inteligentes recibe la solicitud, el mismo preparará una secuencia de órdenes APDU que completan una transacción de procedimiento prerregistrada para el sistema (anfitrión) destinado a comunicarse con la tarjeta inteligente que está incorporada en el dispositivo habilitado para Java ME. El sistema responde enviando por lo menos una de las órdenes APDU a la aplicación Java ME remota. Cada intercambio de una colección de órdenes APDU entre la aplicación y el sistema se conoce como Lote. En otras palabras, un lote es un sobre que contiene múltiples órdenes APDU completas.

Para que se complete cada transacción resultan involucrados múltiples lotes de órdenes APDU que se intercambiarán entre el sistema (anfitrión) y el dispositivo de tarjeta inteligente. Cada lote consta de por lo menos una orden APDU enviada desde el sistema. Desde la aplicación Java ME se genera un lote correspondiente después de que se hayan ejecutado las órdenes APDU en la tarjeta. Este es el resultado de ejecución del lote APDU generado por la aplicación.

Para proporcionar un procesado computacional y un uso de memoria optimizados del dispositivo que está funcionando con una aplicación Java ME®, se define un protocolo de mensajes binarios para gestionar la comunicación. Existirá una aplicación que se ejecuta en el lado del servidor para llevar a cabo la traducción del mensaje de binario a SOAP y viceversa. Este servidor se conoce como Pasarela de Canal en la medida en la que gestiona la entrega del canal APDU por vía aérea. El mensaje binario contiene tres segmentos, describiendo lo siguiente estos segmentos:

- segmento de Encabezamiento: contiene los datos auto-descriptivos.
- segmento de Cuerpo: contiene las órdenes APDU o fragmento de las órdenes en caso de que se requiera fragmentación.
- segmento de Información: contiene la información útil y necesaria requerida por el receptor.

Por lo tanto, cada lote de órdenes APDU encapsuladas en el mensaje SOAP se traduce en un mensaje binario dentro de la pasarela antes de ser enviado a la aplicación que se ejecuta en el dispositivo. De modo similar, el resultado de ejecución de APDU que se encuentra en binario se traduce a SOAP antes de ser enviado al sistema (anfitrión) para un procesado posterior.

Tal como se ha mencionado, al estar limitado el tamaño de datos transmitido por vía aérea, por el operador de la red móvil, se requiere un planteamiento adaptativo para la distribución de datos en fragmentos más pequeños en cada lote de órdenes APDU. Por tanto, cada fragmento contiene parcialmente el lote de órdenes APDU. La aplicación Java ME® en el dispositivo puede proceder a ejecutar un lote una vez que ha recibido todos los fragmentos y realizar una concatenación para obtener de nuevo un único lote.

El mensaje de solicitud binario iniciado desde la aplicación Java ME®, que se transmite desde el lado del dispositivo al lado del servidor/pasarela de canal, tendrá el formato que se muestra en las siguientes tablas:

Segmento de Encabezamiento							Segmento de Cuerpo					Segmento de Información				
h(1)	h(2)	h(3)	(h4)	h(5)	h(6)	h(7)	b(1)	b(2)	...	b(n-1)	b(n)	i(1)	i(2)	i(3)	i(4)	i(5)

(mensaje de solicitud) (sol-1)

con el Segmento de Encabezamiento que es obligatorio:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de orden – describe la solicitud de orden que se debe entregar a la pasarela, por ejemplo, intercambio APDU.
2	h(2)	1	Longitud total del segmento de encabezamiento – describe el tamaño total del segmento de encabezamiento excluyendo el código de información.
3	h(3)	1	Longitud de ID de sesión – el valor de id de sesión es un campo opcional en la medida en la que es asignado por la pasarela. Por tanto, la longitud puede ser 0 para la primera solicitud a la pasarela.
4	h(4)	variable	Valor de ID de sesión – un valor de id de sesión asignado por la pasarela.
5	h(5)	1	Secuencia de lote – esto describe la secuencia actual del lote APDU de órdenes. El intervalo de la secuencia es de 1 a 255.
6	h(6)	1	Secuencia de paquete fragmentado – esto identifica la secuencia del paquete de datos en caso de que el mismo esté fragmentado. Si no hay fragmentación, el valor se fija a 0. El intervalo de la secuencia es 1 a 255.
7	h(7)	1	Índice de segmento de información – indica el índice de la información si el segmento existe para un mensaje particular. Si el segmento no existe, el valor se fija a 0. El intervalo del índice es de 1 a 255.

el segmento de Cuerpo, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	1	Longitud de datos APDU – describe la longitud del primer conjunto de órdenes APDU dentro de un lote.
2	b(2)	variable	Valor de datos APDU – el primer conjunto de datos de órdenes APDU dentro de un lote.
3	b(n-1)	1	Longitud del conjunto N de APDU.
4	b(n)	variable	Datos del conjunto N de APDU.

y el segmento de Información, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	i(1)	1	Código de información – describe la información a entregar a la pasarela, por ejemplo, info de Inicialización de Servicio.
2	i(2)	1	Longitud de ID de aplicación.
3	i(3)	variable	Valor de ID de aplicación – esta es una id de aplicación exclusiva asignada a cada aplicación.
4	i(4)	1	Longitud de datos de información opcional.
5	i(5)	variable	Valor de datos de información opcional.

De modo similar, el mensaje de respuesta binario enviado desde el servidor/pasarela de canal al dispositivo, tendrá el formato que se muestra en las siguientes tablas:

Segmento de Encabezamiento							Segmento de Cuerpo					Segmento de Información		
h(1)	h(2)	h(3)	(h4)	h(5)	h(6)	h(7)	b(1)	b(2)	...	b(n-1)	b(n)	i(1)	i(2)	i(3)

(Mensaje de respuesta) (res-1)

con el segmento de Encabezamiento, el cual es obligatorio:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de respuesta – identifica el estado de respuesta de la pasarela. El código 0 representa habitualmente éxito, si no, representa estado fallido.
2	h(2)	1	Longitud de ID de sesión.
3	h(3)	variable	Valor de ID de sesión – Este campo es un valor de id de sesión asignado por la pasarela.
4	h(4)	1	Código de error – esto describe el código de error de la transacción. Este campo es válido si el código de respuesta no es 0.
5	h(5)	1	Secuencia de lote – esto describe la secuencia actual del lote APDU de órdenes. El intervalo de la secuencia es de 1 a 255.
6	h(6)	1	Secuencia de paquete fragmentado – esto identifica la secuencia del paquete de datos en caso de que el mismo esté fragmentado. Si no hay fragmentación, el valor se fija a 0. El intervalo de la secuencia es de 1 a 255.
7	h(7)	1	Índice de segmento de información – indica el índice de la información si el segmento existe para un mensaje particular. Si el segmento no existe, el valor se fija a 0. El intervalo del índice es de 1 a 255.

el segmento de Cuerpo, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	1	Longitud de datos APDU – describe la longitud del primer conjunto de órdenes APDU dentro de un lote.
2	b(2)	variable	Valor de datos APDU – el primer conjunto de datos de órdenes APDU dentro de un lote.
3	b(n-1)	1	Longitud del conjunto N de APDU.
4	b(n)	variable	Datos del conjunto N de APDU.

y el segmento de Información, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	i(1)	1	Código de información – describe la información que se debe responder desde la pasarela, por ejemplo, tamaño de datos fragmentados tanto para el emisor como para el receptor.
2	i(2)	1	Longitud de datos de información opcional.
3	i(3)	variable	Valor de datos de información opcional.

Los datos se descomponen en fragmentos según se requiera para adaptarse al entorno del operador móvil. El único segmento que en el mensaje se puede descomponer en fragmentos es el segmento de Cuerpo. El resto se mantendrán como una entidad de mensaje completa en su totalidad. Cada vez que los datos se descompongan en fragmentos, la aplicación proporcionará un mensaje para solicitar más fragmentos o, si la fragmentación proviene de la aplicación, enviará la secuencia consecuente de fragmento a la pasarela.

Las siguientes tablas definen el formato de los mensajes binarios de solicitud y respuesta para la gestión de fragmentos:

Segmento de Encabezamiento		Segmento de Cuerpo
h(1)	h(2)	b(1)

(Solicitud de más fragmento) (sol-2)

con el segmento de encabezamiento, el cual es obligatorio:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de orden – existen diferentes órdenes para describir dos casos. 1. La parte solicitante solicita más fragmentos de la pasarela. 2. La parte solicitante envía más fragmentos a la pasarela.
2	h(2)	1	Secuencia de paquetes fragmentados – depende de la fragmentación de la parte solicitante o de la parte respondedora. Si es una fragmentación de la parte solicitante, la secuencia identifica la secuencia fragmentada actual que se envía, si no, la parte solicitante solicita la secuencia fragmentada esperada de la pasarela.

y el segmento de Cuerpo, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	variable	Los datos APDU fragmentados. Se hacen variar de acuerdo con el tamaño de paquete fragmentado definido, asignado por la pasarela. El segmento de cuerpo existe si y solo si la fragmentación se produce desde el lado solicitante.

De manera similar, el fragmento de respuesta presentará el formato que se muestra en las siguientes tablas:

Segmento de Encabezamiento		Segmento de Cuerpo
h(1)	h(2)	b(1)

(Fragmento de respuesta) (res-2)

con el segmento de Encabezamiento, el cual es obligatorio:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de respuesta – debería realizar una descripción para ambos casos mencionados en la Tabla 7. El código 0 habitualmente representa éxito, si no, representa estado fallido. 1. La parte solicitante solicita más fragmentos de la pasarela. 2. La parte solicitante envía más fragmentos a la pasarela.
2	h(2)	1	Secuencia empaquetada fragmentada – depende de la fragmentación de la parte solicitante o la parte respondedora. Si es una fragmentación de la parte respondedora, la secuencia identifica la secuencia fragmentada actual que se envía a la aplicación, si no, es la siguiente secuencia fragmentada esperada que se recibirá desde la pasarela.

y el segmento de Cuerpo, el cual es opcional:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	variable	Los datos APDU fragmentados. Se hacen variar de acuerdo con el tamaño de paquete fragmentado definido, asignado por la pasarela. El segmento de cuerpo existe si y solo si la fragmentación se produce desde el lado respondedor.

La aplicación Java ME® en el lado del dispositivo en primer lugar iniciará una solicitud hacia el operador registrado actual para recopilar detalles de identidad del operador. La información se enviará de vuelta a la pasarela cada vez que se produzca una necesidad de solicitud de servicio de intercambio APDU. El flujo de procesado de comunicaciones entre la aplicación Java ME® y la pasarela se ilustra en la Figura 1 y en la Figura 2 para casos en los que, respectivamente, se produjo una fragmentación de un lote de resultado de ejecución de APDU desde la aplicación Java ME a la pasarela y en los que se produjo una fragmentación de respuesta de órdenes APDU de vuelta desde la pasarela. En dichas Figuras, al tipo de solicitud y de respuesta se le hace referencia respectivamente con sol-1, sol-2, res-1 y res-2, como en las tablas anteriores. Debe indicarse que la primera solicitud no se fragmentará ya que no es portadora de ningún tamaño de datos fragmentado predefinido para la transferencia. La solicitud sucesiva se puede fragmentar en función de las necesidades de la misma, mientras que se puede elegir si cada respuesta será fragmentada.

El protocolo proporciona también una funcionalidad de autorecuperación en la cual la aplicación Java ME® es el iniciador del procedimiento de recuperación. El procedimiento de recuperación se lanzará si la aplicación no recibe una respuesta en un periodo de tiempo definido.

En relación con la Figura 3, las siguientes etapas describen cómo se usan los puntos de memorización en el procedimiento de recuperación y cuándo y dónde se activa el autoreintento desde la aplicación Java ME®.

1. En primer lugar, la aplicación Java ME envía una solicitud de intercambio APDU a la pasarela de canal.

2. La pasarela construye un mensaje SOAP de solicitud inicial y lo lanza al sistema de gestión de tarjetas inteligentes.

3. El primer lote de órdenes APDU se envía a la pasarela, y la pasarela traduce el mensaje SOAP en un mensaje binario según se define en la siguiente subsección.

4. Antes de que el mensaje binario se envíe de vuelta a la aplicación, la pasarela creará el primer punto de memorización y el mensaje se memoriza (en una memoria, archivo o Base de Datos).

5. Si este mensaje es recibido por la aplicación, la misma analizará sintácticamente el mensaje e intercambiará la APDU con el elemento seguro. A continuación, la aplicación prepara un lote del resultado de ejecución de órdenes APDU y realiza un envío de vuelta a la pasarela. Antes del envío de vuelta a la pasarela, se crea el primer punto de memorización en la aplicación.

6. La aplicación esperará la siguiente respuesta durante un cierto periodo de tiempo (por ejemplo, 2 s). El periodo de tiempo límite total estará dentro de los límites de la sesión de transacción completa. Si el cliente no ha recibido la siguiente respuesta dentro del periodo de tiempo límite, se activa un autoreintento. Se volverá a enviar nuevamente el mensaje desde el primer punto de memorización. El número máximo de reintentos se define de acuerdo con el acho de banda y la fiabilidad de la conexión.

7. En el lado de la pasarela, al producirse la recepción del lote de resultado de ejecución de órdenes APDU, la pasarela analiza sintácticamente el mensaje y construye un mensaje SOAP encapsulado con el resultado de ejecución y lo envía de vuelta al sistema de gestión de tarjetas inteligentes.

8. A continuación, la pasarela repetirá la etapa 4.

9. Mientras se crea el punto de memorización 2 en la pasarela, se recibe otro mensaje binario de solicitud desde la aplicación, la pasarela intentará volver a enviar el mismo mensaje desde lo memorizado en el punto de memorización 2 a la aplicación nuevamente.

10. Si la aplicación recibe el mensaje de respuesta, repetirá la etapa 5, si no, repetirá la etapa 6.

El procedimiento de intercambio APDU cesará si la transacción se completa y la aplicación recibe la transacción completada con un estado de éxito, si no, la transacción se aborta con un código de error especificado.

REIVINDICACIONES

1. Procedimiento de comunicación por medio de HTTP o HTTPS entre un dispositivo portátil que ejecuta Java ME con una tarjeta inteligente y un servidor por vía aérea, recibiendo y transmitiendo dicho servidor datos APDU encapsulados en mensajes SOAP desde/hacia un operador en un anfitrión a través de una red y transmitiendo y recibiendo datos APDU contenidos en mensajes binarios desde/hacia el dispositivo portátil, traduciéndose cada mensaje SOAP a partir de/en mensajes binarios de acuerdo con un protocolo en el servidor, intercambiándose dichos mensajes binarios con el dispositivo portátil,

y en caso de que los datos APDU no puedan estar contenidos en un mensaje binario, el intercambio se efectúa en una transmisión por lotes de una pluralidad de mensajes binarios,

caracterizado porque para ejecutar una comunicación según el protocolo con el dispositivo portátil que ejecuta Java ME y su tarjeta inteligente, y usando lotes de órdenes APDU completas, se implementan las etapas siguientes:

- el dispositivo portátil en primer lugar envía al servidor un mensaje binario que incluye una solicitud de intercambio APDU e información de inicialización de servicio para identificar un operador,

- como respuesta, el servidor envía de vuelta al dispositivo portátil un mensaje binario que contiene un lote de órdenes APDU o un fragmento del lote de órdenes APDU, conteniendo cada mensaje binario un segmento de Encabezamiento, un segmento de Cuerpo y un segmento de Información, presentando el segmento de Encabezamiento un campo de Secuencia de paquete fragmentado que proporciona información sobre la existencia o no de una fragmentación y, en caso de fragmentación, sobre el número de secuencia del fragmento, presentando el segmento de Encabezamiento un campo de índice de segmento de Información, y porque, dicho lote de órdenes APDU o su fragmento se almacena en el segmento de Cuerpo del mensaje binario, presentando el segmento de Información un campo de código de Información que describe la información enviada, a continuación:

- si dicho lote de órdenes APDU no se fragmenta, el dispositivo portátil envía un mensaje binario que incluye un resultado de ejecución de dicho lote de órdenes APDU para completar la transacción para dicho lote,

- si dicho lote de órdenes APDU se fragmenta, el dispositivo portátil envía un mensaje de solicitud binario de más fragmento con la siguiente secuencia de fragmento y espera un mensaje de respuesta binario para más fragmento, del servidor, para el siguiente fragmento y, cuando se reciben todos los fragmentos, el dispositivo portátil ejecuta dicho lote, a continuación el dispositivo portátil envía un mensaje binario que incluye un resultado de ejecución de dicho lote,

y cuando se completa la transacción para dicho lote, el dispositivo espera un mensaje binario del servidor para un posible lote sucesivo.

2. Procedimiento según la reivindicación 1, caracterizado porque los mensajes binarios son mensajes de solicitud binarios y mensaje de respuesta binario, presentando el mensaje de solicitud binario una parte de segmento de Encabezamiento de siete bytes, una parte de segmento de Cuerpo de n bytes, y una parte de segmento de Información de cinco bytes, presentando el mensaje de respuesta binario una parte de segmento de Encabezamiento de siete bytes, una parte de segmento de Cuerpo de n bytes y una parte de segmento de Información de tres bytes.

3. Procedimiento según la reivindicación 2, caracterizado porque, para el mensaje de solicitud binario de acuerdo con:

Segmento de Encabezamiento							Segmento de Cuerpo					Segmento de Información				
h(1)	h(2)	h(3)	(h4)	h(5)	h(6)	h(7)	b(1)	b(2)	...	b(n-1)	b(n)	i(1)	i(2)	i(3)	i(4)	i(5)

el Segmento de Encabezamiento que es obligatorio es:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de orden – describe la solicitud de orden que se debe entregar al servidor, por ejemplo, intercambio APDU.
2	h(2)	1	Longitud total del segmento de encabezamiento – describe el tamaño total del segmento de encabezamiento excluyendo el código de información.
3	h(3)	1	Longitud de ID de sesión – el valor de id de sesión es un campo opcional en la medida en la que es asignado por el servidor. Por tanto, la longitud puede ser 0

n.º	Campo	Longitud (Bytes)	Descripción
			para la primera solicitud al servidor.
4	h(4)	variable	Valor de ID de sesión – un valor de id de sesión asignado por el servidor.
5	h(5)	1	Secuencia de lote – esto describe la secuencia actual del lote APDU de órdenes. El intervalo de la secuencia es de 1 a 255.
6	h(6)	1	Secuencia de paquete fragmentado – esto identifica la secuencia del paquete de datos en caso de que el mismo esté fragmentado. Si no existe fragmentación, el valor se fija a 0. El intervalo de la secuencia es de 1 a 255.
7	h(7)	1	Índice de segmento de información – indica el índice de la información si el segmento existe para un mensaje particular. Si el segmento no existe, el valor se fija a 0. El intervalo del índice es de 1 a 255.

el segmento de Cuerpo, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	1	Longitud de datos APDU – describe la longitud del primer conjunto de órdenes APDU dentro de un lote.
2	b(2)	variable	Valor de datos APDU – el primer conjunto de datos de órdenes APDU dentro de un lote.
3	b(n-1)	1	Longitud del conjunto N de APDU.
4	b(n)	variable	Datos del conjunto N de APDU.

y el segmento de Información, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	i(1)	1	Código de información – describe la información que se debe entregar al servidor, por ejemplo, info de Inicialización de Servicio.
2	i(2)	1	Longitud de ID de aplicación.
3	i(3)	variable	Valor de ID de aplicación – esta es una id de aplicación única asignada a cada aplicación.
4	i(4)	1	Longitud de datos de información opcional.
5	i(5)	variable	Valor de datos de información opcional.

4. Procedimiento según la reivindicación 2, caracterizado porque, para el mensaje de respuesta binario de acuerdo con:

Segmento de Encabezamiento							Segmento de Cuerpo					Segmento de Información		
h(1)	h(2)	h(3)	h(4)	h(5)	h(6)	h(7)	b(1)	b(2)	...	b(n-1)	b(n)	i(1)	i(2)	i(3)

el segmento de Encabezamiento, el cual es obligatorio, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de respuesta – identifica el estado de respuesta del servidor. El código 0 representa habitualmente éxito, si no, representa estado fallido.
2	h(2)	1	Longitud de ID de sesión.
3	h(3)	variable	Valor de ID de sesión – Este campo es un valor de id de sesión asignado por el servidor.
4	h(4)	1	Código de error – esto describe el código de error de la transacción. Este campo es válido si el código de respuesta no es 0.
5	h(5)	1	Secuencia de lote – esto describe la secuencia actual del lote APDU de órdenes.
6	h(6)	1	Secuencia de paquete fragmentado – esto identifica la secuencia del paquete de datos en caso de que el mismo esté fragmentado.
7	h(7)	1	Índice de segmento de información – indica el índice de la información si el segmento existe para un mensaje particular.

el segmento de Cuerpo, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción

1	b(1)	1	Longitud de datos APDU – describe la longitud del primer conjunto de órdenes APDU dentro de un lote.
2	b(2)	variable	Valor de datos APDU – el primer conjunto de datos de órdenes APDU dentro de un lote.
3	b(n-1)	1	Longitud del conjunto N de APDU.
4	b(n)	variable	Datos del conjunto N de APDU.

y el segmento de Información, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	i(1)	1	Código de información – describe la información que se debe responder desde el servidor, por ejemplo, tamaño de datos fragmentados tanto para el emisor como para el receptor.
2	i(2)	1	Longitud de datos de información opcional.
3	i(3)	variable	Valor de datos de información opcional.

5. Procedimiento según la reivindicación 2, 3 o 4, caracterizado porque el dispositivo portátil puede solicitar más segmentos con un mensaje de solicitud binario de más fragmento que presenta una parte de segmento de Encabezamiento de dos bytes y una parte de segmento de Cuerpo de un byte, siendo la respuesta de la solicitud de más fragmento del servidor un mensaje de respuesta binario de más fragmento que presenta una parte de segmento de Encabezamiento de dos bytes y una parte de segmento de Cuerpo de un byte.

6. Procedimiento según la reivindicación 5, caracterizado porque, para la solicitud binaria de más fragmento de acuerdo con:

Segmento de Encabezamiento		Segmento de Cuerpo
h(1)	h(2)	b(1)

el segmento de encabezamiento, el cual es obligatorio, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de orden – existen diferentes órdenes para describir dos casos. 1. La parte solicitante solicita más fragmentos del servidor. 2. La parte solicitante envía más fragmentos al servidor.
2	h(2)	1	Secuencia de paquetes fragmentados – depende de la fragmentación de la parte solicitante o de la parte respondedora. Si es una fragmentación de la parte solicitante, la secuencia identifica la secuencia fragmentada actual que se envía, si no, la parte solicitante solicita la secuencia fragmentada esperada del servidor.

y el segmento de Cuerpo, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	variable	Los datos APDU fragmentados. Se hacen variar de acuerdo con el tamaño de paquete fragmentado definido, asignado por el servidor. El segmento de cuerpo existe si y solo si la fragmentación es desde el lado solicitante.

7. Procedimiento según la reivindicación 5, caracterizado porque, para el mensaje de respuesta binario de más fragmento de acuerdo con:

Segmento de Encabezamiento		Segmento de Cuerpo
h(1)	h(2)	b(1)

el segmento de Encabezamiento, el cual es obligatorio, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	h(1)	1	Código de respuesta. El código 0 habitualmente representa éxito, si no, representa estado fallido. 1. La parte solicitante solicita más fragmentos del servidor. 2. La parte solicitante envía más fragmentos al servidor.
2	h(2)	1	Secuencia empaquetada fragmentada – depende de la fragmentación de la parte

			solicitante o la parte respondedora. Si es una fragmentación de la parte respondedora, la secuencia identifica la secuencia fragmentada actual que se envía a la aplicación, si no, es la siguiente secuencia fragmentada esperada que se recibirá desde el servidor.

y el segmento de Cuerpo, el cual es opcional, es:

n.º	Campo	Longitud (Bytes)	Descripción
1	b(1)	variable	Los datos APDU fragmentados. Se hacen variar de acuerdo con el tamaño de paquete fragmentado definido, asignado por el servidor. El segmento de cuerpo existe si y solo si la fragmentación es desde el lado respondedor.

8. Procedimiento según cualquiera de las reivindicaciones anteriores, caracterizado porque se implementan unos medios de recuperación y de autoreintento, memorizándose mensajes antes de ser enviados en puntos de memorización predefinidos del procedimiento, estando los puntos de memorización, en el servidor, cada vez que el servidor envía al dispositivo un mensaje binario que incluye órdenes APDU, y, en el dispositivo portátil, cada vez que el dispositivo portátil envía al servidor un mensaje binario que incluye un resultado de ejecución de órdenes APDU.

9. Procedimiento según la reivindicación 8, caracterizado porque se inicia un contador de tiempo en el dispositivo portátil cada vez que el dispositivo portátil envía un mensaje de resultado de ejecución de órdenes APDU y porque si el servidor no reacciona después de un recuento de tiempo predeterminado, el dispositivo portátil ejecuta un reintento al volver a enviar el mensaje desde el punto de memorización relacionado, y porque se autoriza un número predeterminado de reintentos para cada punto de memorización antes de abortar el procedimiento.

10. Sistema, caracterizado porque presenta unos medios para la ejecución del procedimiento según cualquiera de las reivindicaciones anteriores.

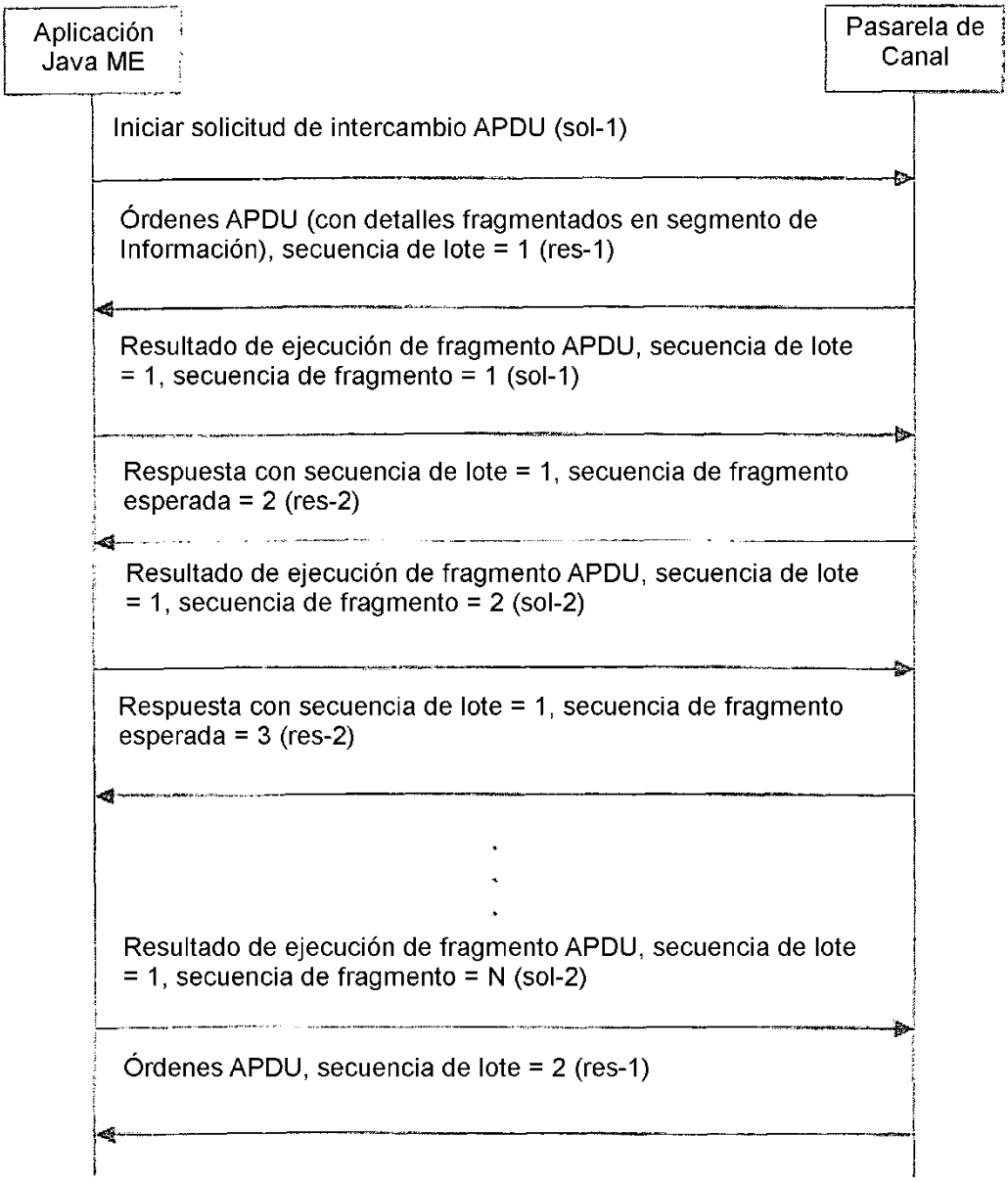
Aplicación Java ME
Pasarela de Canal
Iniciar solicitud de intercambio APDU (sol-1)
Órdenes APDU (con detalles fragmentados en segmento de Información), secuencia de lote = 1 (res-1)
Resultado de ejecución de fragmento APDU, secuencia de lote = 1, secuencia de fragmento = 1 (sol-1)
Respuesta con secuencia de lote = 1, secuencia de fragmento esperada = 2 (res-2)
Resultado de ejecución de fragmento APDU, secuencia de lote = 1, secuencia de fragmento = 2 (sol-2)
Respuesta con secuencia de lote = 1, secuencia de fragmento esperada = 3 (res-2)
Resultado de ejecución de fragmento APDU, secuencia de lote = 1, secuencia de fragmento = N (sol-2)
Órdenes APDU, secuencia de lote = 2 (res-1)

Fig. 1

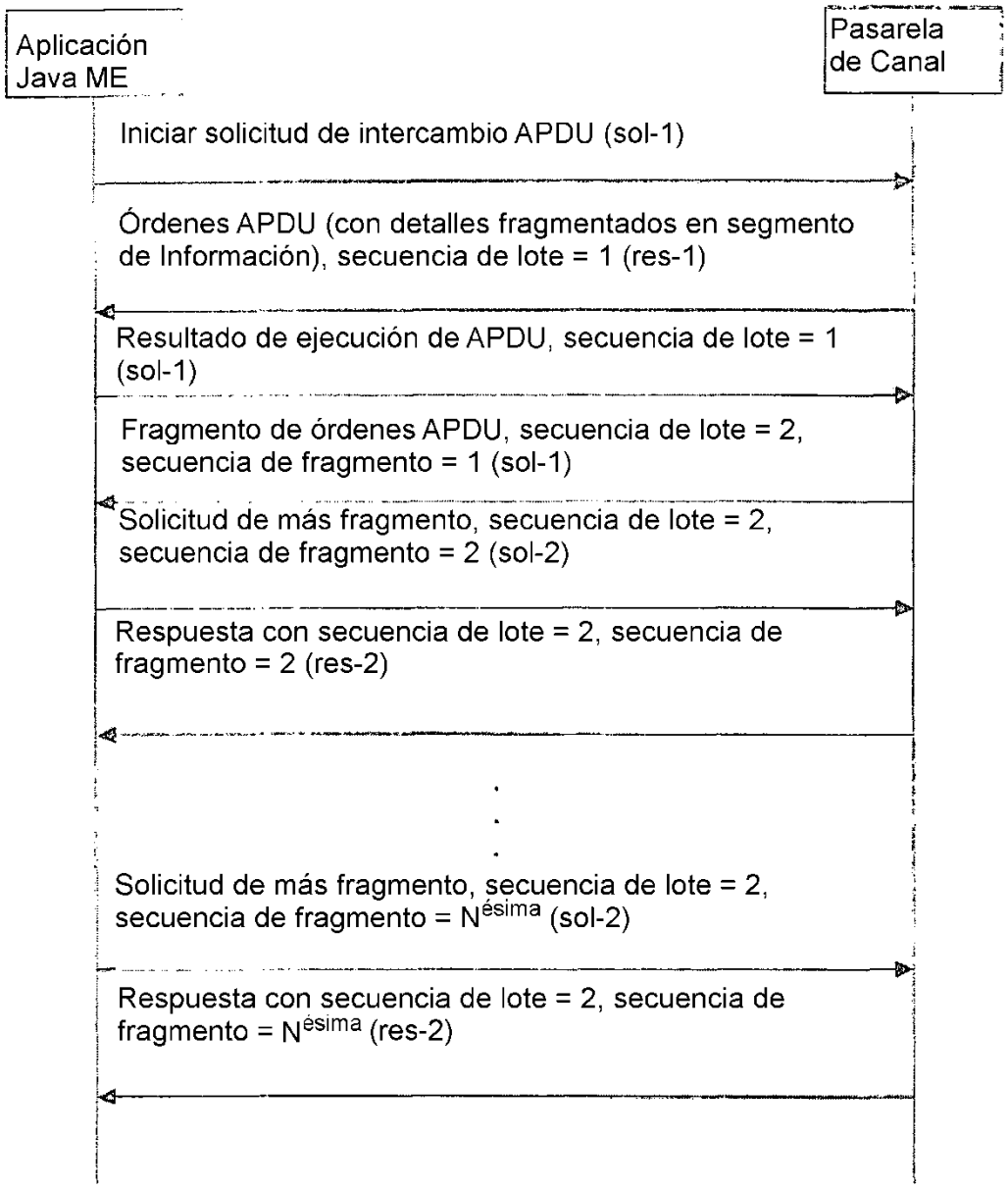
Aplicación Java ME
Pasarela de Canal
Iniciar solicitud de intercambio APDU (sol-1)
Órdenes APDU (con detalles fragmentados en segmento de Información), secuencia de lote = 1 (res-1)
Resultado de ejecución de APDU, secuencia de lote = 1 (sol-1)
Fragmento de órdenes APDU, secuencia de lote = 2, secuencia de fragmento = 1 (sol-1)
Solicitud de más fragmento, secuencia de lote = 2, secuencia de fragmento = 2 (sol-2)
Respuesta con secuencia de lote = 2, secuencia de fragmento = 2 (res-2)
Solicitud de más fragmento, secuencia de lote = 2, secuencia de fragmento = Nésima (sol-2)
Respuesta con secuencia de lote = 2, secuencia de fragmento = Nésima (res-2)

Fig. 2

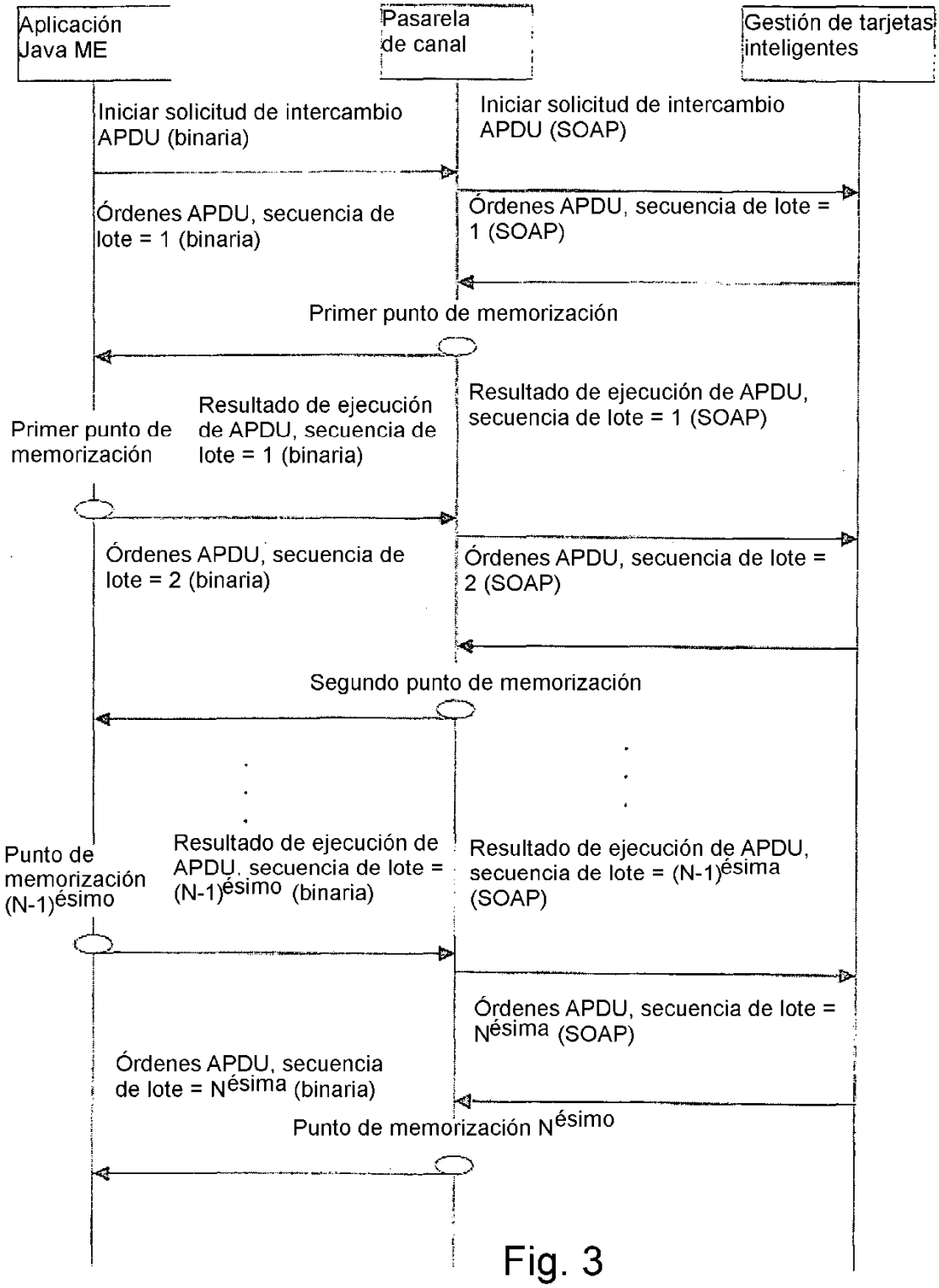
Aplicación Java ME
Pasarela de canal
Gestión de tarjetas inteligentes
Iniciar solicitud de intercambio APDU (binaria)
Iniciar solicitud de intercambio APDU (SOAP)
Órdenes APDU, secuencia de lote = 1 (binaria)
Órdenes APDU, secuencia de lote = 1 (SOAP)
Primer punto de memorización
Primer punto de memorización
Resultado de ejecución de APDU, secuencia de lote = 1 (binaria)
Resultado de ejecución de APDU, secuencia de lote = 1 (SOAP)
Órdenes APDU, secuencia de lote = 2 (binaria)
Órdenes APDU, secuencia de lote = 2 (SOAP)
Segundo punto de memorización
Punto de memorización (N-1)ésimo
Resultado de ejecución de APDU, secuencia de lote = (N-1)ésimo (binaria)
Resultado de ejecución de APDU, secuencia de lote = (N-1)ésima (SOAP)
Órdenes APDU, secuencia de lote = Nésima (SOAP)
Órdenes APDU, secuencia de lote = Nésima (binaria)
Punto de memorización Nésimo

Fig. 3