

SISTEMA DE AUTENTICACIÓN Y PROCESAMIENTO DE TRANSACCIÓN DISTRIBUIDA

Campo

La presente descripción se refiere a sistemas y métodos para realizar transacciones de todo tipo en una implementación única a escala, de forma segura y casi en tiempo real.

Estado del Arte

El procesamiento de transacciones involucra una amplia gama de sistemas informáticos distribuidos y varios transactores que realizan transacciones, particularmente con respecto a los pagos, pero también con el comercio de otros activos e instrumentos financieros, control de acceso físico, acceso lógico a datos, administración y dispositivos de monitoreo que comprenden la Internet de las Cosas (conocido en Inglés como IoT), etc.

Al crear sistemas de procesamiento de transacciones hoy en día, los ingenieros tienen que hacer concesiones difíciles. Estas incluyen la elección entre velocidad y resistencia, rendimiento y consistencia, seguridad y rendimiento, consistencia y escalabilidad, etc. Dichas concesiones invariablemente conducen a compromisos que afectan al sistema en general. Los sistemas de procesamiento de pagos exhiben el efecto de estas compensaciones. Es posible que necesiten manejar entre 600 y unas pocas decenas de miles de transacciones por segundo, pero solo pueden hacerlo procesándolas en parte y almacenando los detalles para su procesamiento posterior durante una pausa en la carga de trabajo del sistema. A menudo, esto conduce a problemas para conciliar los registros faltantes, duplicar transacciones, exponerse a problemas de crédito en los que las cuentas se han sobregirado entre el momento de una transacción y el momento de procesar esa transacción, y así sucesivamente. Sin embargo, los problemas no se limitan a los pagos.

ACID (atomicidad, consistencia, aislamiento y durabilidad) es un modelo de consistencia para las bases de datos que establece que cada transacción de la base de datos debe realizarse si la transacción completa se revierte (atomicidad), no puede dejar la base de datos en un estado

inconsistente (coherencia), no puede interferir entre sí (aislamiento); y debe persistir, incluso cuando los servidores se reinician (durabilidad). En general, se considera que este modelo es incompatible con los requisitos de disponibilidad y rendimiento de los sistemas a gran escala, como las redes de pagos bancarios existentes y otros sistemas transaccionales de "big data". Estos sistemas, en cambio, dependen de la consistencia BASE, (disponibilidad básica, estado suave y consistencia eventual). Este modelo sostiene que esto es suficiente para que la base de datos eventualmente alcance un estado consistente. Los sistemas bancarios operan en este modo, por lo que a menudo necesitan pausar cualquier procesamiento de transacción y ejecutar verificaciones de conciliación para alcanzar un estado consistente. Esta noción de que las compensaciones deben realizarse en el procesamiento de transacciones de gran volumen está consagrada en el teorema de CAP, que en su forma básica establece que es imposible que un sistema informático distribuido proporcione las tres (C) consistencia, (A) disponibilidad y (P) tolerancia de la partición al mismo tiempo.

Las soluciones actuales de mejores prácticas contienen demasiadas limitaciones y concesiones para satisfacer los requisitos actuales y emergentes. El problema de cómo reconciliar los datos que se generan en la Internet de las Cosas está comenzando a surgir, un problema que surge debido a los efectos de las concesiones que los ingenieros creen que deben realizar al construir las redes y sistemas de procesamiento de transacciones. Uno de los efectos ha sido la falta de seguridad en las comunicaciones entre dispositivos y servidores que, en conjunto, conforman el Internet de las Cosas. Otra es la incapacidad de garantizar que los datos recopilados por un dispositivo realmente se relacionen con un evento específico detectado por ese dispositivo.

Los sistemas de almacenamiento de información basados en la nube también exhiben los efectos de estas compensaciones, que a menudo resultan en una gran cantidad de servidores y sistemas que pueden garantizar solo la eventual consistencia.

Por lo tanto, es necesario proporcionar una consistencia ACID con sistemas a gran escala que en sistemas conocidos solo puedan beneficiarse de la consistencia BASE.

Descripción Resumida de la Invención

De acuerdo con un aspecto, se proporciona un método para registrar una transacción de datos que comprende, en un dispositivo asociado con una primera entidad, determinar los primeros datos iniciales, generar un registro de una primera transacción de datos entre la primera entidad y una segunda entidad, determinar segundos datos iniciales combinando al menos los primeros datos iniciales y el registro de la primera transacción de datos, generando un primer resumen (donde resumen es conocido también en el arte como “resumen”) mediante la función de resumir (función conocida también en el arte como “hashing”) los segundos datos iniciales, en donde el primer resumen comprende un historial de transacciones de datos que involucran a la primera entidad y almacenando el primer resumen contra el registro de la primera transacción de datos en una memoria. Según otro aspecto, se proporciona un dispositivo asociado con una primera entidad, estando el dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático lleve a cabo el método.

De acuerdo con otro aspecto, se proporciona un dispositivo de licencia configurado para recibir un primer resumen de un dispositivo asociado con una primera entidad, en donde el primer resumen comprende un historial de transacciones de datos que involucran a la primera entidad, combina el primer resumen con un resumen de licencia para proporcionar una entrada de licencia, genera un segundo resumen de licencia mediante el resumen de la entrada de licencia y almacena el segundo resumen de licencia en una memoria.

De acuerdo con otro aspecto, se proporciona un dispositivo de directorio configurado para recibir un primer resumen de un dispositivo asociado con una primera entidad, en donde el primer resumen comprende un historial de transacciones de datos que involucran a la primera entidad combinando el primer resumen con un resumen de directorio para proporcionar un ingreso en el directorio, genera un segundo resumen de directorio mediante el resumen de la entrada de licencia y almacena el segundo resumen de directorio en una memoria.

De acuerdo con otro aspecto, se proporciona un método para acceder a un primer servicio desde un dispositivo que comprende proporcionar un identificador del dispositivo a un servidor de solicitud, autorizando al dispositivo a solicitar acceso al primer servicio basado en el identificador, permitiendo que el dispositivo acceda al primer servicio de un primer servidor huésped en donde se encuentra el primer servicio, siendo el acceso a través del servidor de solicitud. Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice o lleve a cabo el método.

Según otro aspecto, se proporciona un método de migración de datos que comprende proporcionar una solicitud o pedido para cambiar los primeros datos de un primer almacenamiento de datos a un segundo almacenamiento de datos, determinando, desde un servidor de directorio, un identificador del primer almacenamiento de datos basado en un identificador comprendido en la solicitud o pedido, migrando los primeros datos del primer almacenamiento de datos al segundo almacenamiento de datos. Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método. De acuerdo con otro aspecto, se proporciona un método de comunicación que comprende enviar una primera comunicación de una primera entidad a una segunda entidad, en donde la primera comunicación comprende dos o más campos de datos, donde cada campo comprende una etiqueta respectiva, y el envío de una segunda comunicación desde la primera entidad a la segunda entidad, donde la segunda comunicación comprende los dos o más campos de datos, en donde el orden de los campos en la segunda comunicación es diferente del orden de los campos en la primera comunicación.

Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método.

De acuerdo con otro aspecto, se proporciona un método de comunicación a través de datos de servicio suplementarios no estructurados "USSD" (según siglas del Inglés) que comprende abrir una sesión USSD entre un primer dispositivo y un segundo dispositivo, generando un texto cifrado para una comunicación en la sesión en el primer dispositivo, codificando el texto encriptado en el primer dispositivo, transmitiendo el texto encriptado codificado desde el primer dispositivo al segundo dispositivo para su descifrado en el segundo dispositivo. Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método.

Según otro aspecto, se proporciona un método de comunicación entre un primer dispositivo asociado con una primera entidad y un segundo dispositivo asociado con una segunda entidad que comprende, en el primer dispositivo, generar una primera sesión de PAKE entre el primer dispositivo y el segundo dispositivo usando un primer secreto compartido, recibiendo una clave de registro y un segundo secreto compartido del segundo dispositivo, y mezclar el primer secreto compartido, la clave de registro y el segundo secreto compartido para proporcionar un tercer secreto compartido para generar una segunda sesión de PAKE. Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método. De acuerdo con otro aspecto, se proporciona un método para acceder a un servicio que comprende proporcionar una credencial y un contexto para la credencial, autenticando el acceso al servicio basado en la credencial y el contexto.

Según otro aspecto, se proporciona un dispositivo configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método. Según otro aspecto, se proporciona un método de comunicación entre módulos en un sistema informático, comprendiendo el método transportar un canal de memoria compartida desde un primer módulo a un proxy, transportando el canal de memoria compartida desde el proxy a un segundo módulo, en donde el proxy comprende un módulo de transferencia configurado para

transmitir datos entre el primer módulo y el segundo módulo mediante la omisión del núcleo del sistema informático, que transmite datos del primer módulo al segundo módulo. Según otro aspecto, se proporciona un dispositivo informático configurado para realizar el método. De acuerdo con otro aspecto, se proporciona un medio legible por computadora que comprende partes de código que, cuando se ejecuta, hace que un dispositivo informático realice el método. Los primeros datos de inicio pueden comprender un resumen inicial. El resumen inicial puede ser el resultado de un resumen de un registro de una transacción de datos anterior que involucra a la primera entidad. El resumen inicial puede comprender un resumen aleatorio. El resumen aleatorio puede comprender al menos una firma del dispositivo, la fecha y / o la hora en que se generó el resumen aleatorio.

Proporcionar una segunda información inicial puede comprender además la combinación de una primera prueba de conocimiento cero y una segunda prueba de conocimiento cero con los primeros datos iniciales y el registro de la primera transacción de datos, en donde la primera prueba de conocimiento cero puede comprender una prueba de que el resumen inicial puede comprender el verdadero resumen de la transacción de datos anterior que involucra a la primera entidad, y la segunda prueba de conocimiento cero puede comprender la prueba de que un segundo resumen puede comprender el resumen verdadero de una transacción de datos previa que involucra a la segunda entidad. Proporcionar una segunda información inicial puede comprender además la combinación de una tercera prueba de conocimiento cero con la primera información inicial, el registro de la primera transacción de datos, la primera prueba de conocimiento cero y la segunda prueba de conocimiento cero. La tercera prueba de conocimiento cero se puede generar a partir de datos aleatorios. La tercera prueba de conocimiento cero puede ser una repetición de la primera prueba de conocimiento cero o la segunda prueba de conocimiento cero. La tercera prueba de conocimiento cero se puede construir utilizando un segundo registro de la primera transacción de datos que corresponde a la segunda prueba de conocimiento cero.

La primera transacción de datos puede comprender al menos dos etapas y proporcionar una segunda información inicial puede comprender combinar la primera prueba de conocimiento cero con un registro de la primera etapa de la primera transacción de datos, y combinar la

segunda prueba de conocimiento cero con un registro de la segunda etapa de la primera transacción de datos. Proporcionar una segunda información inicial o fatos de inicio puede comprender construir una tercera prueba de conocimiento cero a partir del registro de la segunda etapa de la primera transacción de datos, y combinar la segunda prueba de conocimiento cero y la tercera prueba de conocimiento cero con el registro de la segunda etapa de la primera transacción de datos. La primera transacción de datos puede comprender al menos tres etapas y proporcionar una segunda información inicial puede comprender además la combinación de la primera prueba de conocimiento cero con un registro de la tercera etapa de la primera transacción de datos y la combinación de la segunda prueba de conocimiento cero con el registro de la tercera etapa de la primera transacción de datos. La primera transacción de datos puede comprender al menos tres etapas y proporcionar una segunda información inicial puede comprender además la combinación de la primera prueba de conocimiento cero con un registro de la tercera etapa de la primera transacción de datos, y la combinación de la segunda prueba de conocimiento cero con datos aleatorios.

La primera transacción de datos puede comprender al menos tres etapas y proporcionar una segunda información inicial puede comprender además la combinación de la primera prueba de conocimiento cero con un registro de la tercera etapa de la primera transacción de datos, y la combinación de la segunda prueba de conocimiento cero con un registro de una cuarta etapa de la primera transacción de datos en la que la cuarta etapa de la primera transacción de datos puede ser una repetición de la tercera etapa de la primera transacción de datos.

La primera transacción de datos puede comprender al menos tres etapas y proporcionar una segunda información inicial puede comprender además la combinación de una tercera prueba de conocimiento cero con un registro de la tercera etapa de la primera transacción de datos.

La primera prueba de conocimiento cero puede ser construida por el dispositivo asociado con la primera entidad y la segunda prueba de conocimiento cero puede ser construida por un dispositivo asociado con la segunda entidad.

La construcción de la primera prueba de conocimiento cero y la segunda prueba de conocimiento cero puede comprender el uso de un algoritmo de intercambio de claves. El algoritmo de intercambio de claves puede comprender un algoritmo PAKE.

El método puede comprender además enviar el primer resumen a un dispositivo asociado con la segunda entidad que recibe un segundo resumen desde un dispositivo asociado con la segunda entidad, en donde el segundo resumen puede comprender un resumen de una transacción de datos previa que involucra a la segunda entidad, y generar un registro de una segunda transacción de datos entre la primera parte y la segunda parte, que determina terceros datos iniciales combinando el registro de la segunda transacción de datos con el primer resumen y el segundo resumen, generando un tercer resumen mediante la codificación de los terceros datos iniciales, en donde el tercer resumen comprende un historial de transacciones de datos que involucran a la primera entidad y un historial de transacciones de datos que involucran a la segunda entidad, y almacena el tercer resumen contra el registro de la segunda transacción de datos en la memoria.

La provisión de tercera información inicial puede comprender además la combinación de una tercera prueba de conocimiento cero y una cuarta prueba de conocimiento cero con el registro de la segunda transacción de datos, el primer resumen y el segundo resumen, en donde la tercera prueba de conocimiento cero puede comprender la prueba de que el primer resumen puede comprender un resumen verdadero de la primera transacción de datos, y la cuarta prueba de conocimiento cero puede comprender la prueba de que el segundo resumen puede comprender el resumen verdadero de la transacción de datos anterior que involucra a la segunda entidad. La transacción de datos anterior que involucra a la segunda entidad puede ser la primera transacción de datos.

El método puede comprender además asociar cada uno de los cortes con un identificador de la primera entidad y/o la segunda entidad. El método puede comprender además recalcular el primer resumen, y comparar el primer resumen generado con el segundo resumen recalculado para determinar una coincidencia. El método puede comprender además cancelar transacciones de datos adicionales si la comparación no tiene éxito. El método puede

comprender además generar, en un dispositivo del sistema, un resumen del sistema correspondiente a la primera transacción de datos.

La provisión de segundos datos iniciales puede comprender además la combinación del resumen del sistema con los primeros datos iniciales y el registro de la primera transacción de datos. El resumen del sistema puede ser el resultado de un resumen de un registro de una transacción de datos anterior en el dispositivo del sistema.

La provisión de segundos datos iniciales puede comprender además recibir un resumen de licencia desde un dispositivo de licencia, y combinar el resumen de licencia con los primeros datos iniciales y el registro de la primera transacción de datos para proporcionar los segundos datos iniciales. El método puede comprender además, en el dispositivo de licencia que recibe el primer resumen, combinar el primer resumen con el resumen de licencia para proporcionar una entrada de licencia, generar un segundo resumen de licencia mediante la aplicación de resumen de la entrada de licencia.

La provisión de segundos datos iniciales puede comprender además recibir un resumen de directorio desde un dispositivo de directorio, y combinar el resumen de directorios con los primeros datos iniciales y el registro de la primera transacción de datos para proporcionar los segundos datos iniciales.

El método puede comprender, además, en el servidor de directorio, recibir el primer resumen, combinar el primer resumen con el resumen de directorio para proporcionar una entrada de directorio, generar un segundo resumen de directorio mediante el resumen de la entrada de directorio.

Proporcionar segundos datos iniciales puede comprender además generar un resumen clave a partir de una clave de cifrado para la primera transacción de datos, y combinar el resumen clave con los primeros datos iniciales y el registro de la primera transacción de datos para proporcionar los segundos datos iniciales. La clave de cifrado puede comprender una clave pública o una clave privada.

La combinación de los primeros datos iniciales y el registro de la primera transacción de datos se puede realizar tan pronto como la primera transacción de datos se complete. La memoria puede estar ubicada en un dispositivo remoto. El método puede comprender además comparar, en el dispositivo remoto, el primer resumen con los cortes correspondientes recibidos de otros dispositivos. El método puede comprender además notificar otros dispositivos a los que se puede conectar el dispositivo para esperar recibir el primer resumen.

El método puede comprender además almacenar una cadena de cortes en la memoria. El método puede comprender además enviar la cadena de cortes a una segunda memoria ubicada en un dispositivo configurado para limitar el acceso a las cadenas de cortes que se han enviado. El método puede comprender además enmendar o eliminar un resumen en la cadena de cortes mediante la regeneración de un resumen de sujeto en la cadena de cortes, confirmando que el registro no se ha modificado, registrar el resumen regenerado, modificar o eliminar el registro, generar un nuevo resumen para el registro mediante el resumen de una combinación del resumen de sujeto y el registro modificado/eliminado, y registrar el nuevo resumen. El método puede comprender además generar un resumen del sistema utilizando el nuevo resumen.

El dispositivo puede comprender un servidor. El dispositivo puede comprender un dispositivo de usuario. El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para operar en la "IoT" Internet de las Cosas. El dispositivo de usuario puede configurarse para almacenar el primer resumen en una memoria en el dispositivo. El dispositivo del usuario puede configurarse para almacenar el primer resumen en una memoria en el dispositivo solo cuando puede estar desconectado de un servidor correspondiente. El dispositivo puede configurarse además para enviar el primer resumen a un dispositivo asociado con la segunda entidad. El dispositivo puede configurarse además para enviar una copia firmada y encriptada del registro de la primera transacción de datos al dispositivo asociado con la segunda entidad, en donde la firma puede comprender una indicación de un servidor de destino para ese registro. El dispositivo puede configurarse para firmar el registro con una clave pública específica fuera de línea. El dispositivo puede configurarse para firmar el registro con una clave que pertenece al dispositivo. Solo el

servidor de destino puede descifrar la copia encriptada del registro de la primera transacción de datos.

El dispositivo puede configurarse para enviar los registros cifrados de sus transacciones de datos fuera de línea y los resúmenes asociados a su servidor correspondiente cuando el dispositivo recupera la conexión con su servidor correspondiente. El dispositivo también puede configurarse para enviar copias de registros de transacciones de datos que involucren a otras entidades que mantiene en su servidor correspondiente para su transmisión a servidores correspondientes a las otras entidades. La transmisión puede comprender notificar a todos los servidores a los que se aplican los registros para esperar recibir los registros. El dispositivo puede configurarse para crear un número de transacción interno único para identificar su parte en la primera transacción de datos.

La autorización puede comprender confirmar que el dispositivo de usuario puede estar autorizado para acceder al primer servicio basado en el identificador. La confirmación puede comprender confirmar que el usuario cumple al menos un criterio basado en el identificador. Un primer criterio puede almacenarse en el primer servidor huésped o el servidor de solicitud, y un segundo criterio puede ubicarse en un servidor diferente. La autorización puede comprender verificar una firma en una comunicación entre el servidor de solicitud y el primer servidor huésped.

La autorización se puede realizar en el servidor de solicitud. La autorización puede comprender determinar, en el servidor de solicitud, que el dispositivo fue autorizado previamente para acceder al primer servicio.

La autorización se puede realizar en un servidor de directorio. La autorización puede comprender el servidor de solicitudes que solicita la autorización del dispositivo al servidor de directorios. La habilitación puede comprender que el servidor de directorio envíe un identificador para el primer servidor huésped al servidor de solicitud. Los datos que autorizan el identificador pueden almacenarse solo en el servidor de directorio.

El método puede comprender además solicitar acceso a un segundo servicio, autorizando al dispositivo a acceder al segundo servicio basado en el identificador, permitiendo el acceso del dispositivo al segundo servicio a través del servidor de solicitud. El segundo servicio puede estar ubicado en el primer servidor huésped. El segundo servicio puede estar ubicado en un segundo servidor huésped.

La autorización del dispositivo para acceder al primer servicio se puede realizar en un primer servidor de directorios, y la autorización del dispositivo de usuario para acceder al segundo servicio se puede realizar en un segundo servidor de directorios.

El método puede comprender, además, solicitar el acceso a un tercer servicio, autorizando al dispositivo a acceder al tercer servicio basado en el identificador, permitiendo el acceso del dispositivo al tercer servicio.

El segundo servicio puede estar ubicado en el primer servidor huésped, el segundo servidor huésped o un tercer servidor huésped. La autorización del dispositivo para acceder al tercer servicio se puede realizar en un tercer servidor de directorio.

Proporcionar un identificador puede comprender el dispositivo que se comunica con el servidor de solicitud a través de un túnel cifrado. El método puede comprender además datos de almacenamiento en caché recibidos en cada servidor respectivo. Cada servidor huésped puede ofrecer más de un servicio.

El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para Internet de las Cosas.

La migración puede comprender, en el servidor de directorio, asignar una marca de tiempo de inicio para los datos en el segundo almacenamiento de datos, y asignar una marca de tiempo final para los datos en el primer almacenamiento de datos.

El método puede comprender, además, dar instrucciones a un servidor solicitante que intente acceder a los datos a través del primer almacenamiento de datos después de la marca de hora final para buscar al usuario en el segundo almacenamiento de datos a través del servidor de directorio. Los datos en el primer almacenamiento de datos pueden comprender un primer registro de cuenta con un primer proveedor de cuentas, y los datos en el segundo almacenamiento de datos pueden comprender un segundo registro de cuenta con un nuevo proveedor de cuentas. La migración puede comprender enviar información sobre el primer registro de la cuenta del proveedor de la cuenta actual al nuevo proveedor de la cuenta. La información puede comprender al menos uno de los registros, saldos, configuraciones y/o instrucciones de pago. La migración puede comprender confirmar una código de autenticación que indica que el primer registro debe cambiarse del proveedor de la cuenta actual al nuevo proveedor de la cuenta. El primer registro de cuenta puede comprender una primera credencial de usuario, y el segundo registro de cuenta puede comprender una segunda credencial de usuario. La primera credencial de usuario puede registrarse en un primer servidor y la segunda credencial de usuario puede registrarse en un segundo servidor. El método puede comprender además recibir, por el primer proveedor de cuentas, una comunicación dirigida a un usuario que usa la primera credencial de usuario, enrutando o dirigiendo la comunicación al segundo proveedor de cuentas utilizando la segunda credencial de usuario. El método puede comprender además revertir una transacción de datos realizada con el primer proveedor de registro usando la primera credencial al segundo proveedor de registro usando la segunda credencial de usuario. El método puede comprender además determinar que el usuario usó la primera credencial del usuario en el momento de la transacción de datos. Es posible que un servidor que envía la comunicación deba ser aprobado para acceder a la segunda credencial de usuario. La primera credencial de usuario y la segunda credencial de usuario pueden ser las mismas.

El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para "IoT", es decir la Internet de las Cosas.

El método puede comprender además añadir un campo aleatorio a la segunda comunicación. Cada campo puede comprender dos o más caracteres, y el método comprende además mezclar casos de caracteres en al menos un campo.

El método puede comprender además descifrar y ordenar, por la segunda entidad, los campos en la segunda comunicación antes de procesar la segunda comunicación. El método puede comprender además descartar, por la segunda entidad, campos que no puede procesar. Al menos una de la primera entidad y la segunda entidad pueden comprender un servidor. Al menos una de la primera entidad y la segunda entidad puede comprender una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para "IoT", Internet de las Cosas. El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para la Internet de las Cosas.

La codificación puede comprender codificar el texto encriptado como una cadena de caracteres de 7 u 8 bits. El método puede comprender, además, si la longitud del texto cifrado es más larga que el espacio permitido en la sesión USSD, cortando el texto cifrado en dos o más partes, y transmitiendo las dos o más partes individualmente. El descifrado puede comprender, además, reensamblar las partes en todo el texto cifrado en el segundo dispositivo.

El método puede comprender además autenticar los dispositivos primero y segundo. La autenticación puede comprender el uso de un algoritmo que proporcione privacidad e integridad de datos entre dos aplicaciones informáticas en comunicación. La autenticación puede comprender el uso de seguridad de capa de transporte "TLS". El uso de TLS puede comprender además generar una primera clave de sesión.

El método puede comprender además usar la primera clave de sesión para cifrar una negociación del protocolo PAKE para generar una segunda clave de sesión, y cifrar otras comunicaciones en la sesión entre la primera parte y la segunda parte utilizando la segunda clave de sesión.

El método puede comprender además autenticar la primera entidad y la segunda entidad. La autenticación puede comprender el uso de un algoritmo que proporcione privacidad e integridad de datos entre dos aplicaciones informáticas en comunicación. La autenticación puede comprender el uso de TLS. El método puede comprender además generar una segunda sesión PAKE entre el primer dispositivo y un tercer dispositivo utilizando un cuarto secreto compartido. El cuarto secreto compartido puede comprender un código de autenticación generado por el tercer dispositivo para el primer dispositivo.

El primer secreto compartido puede comprender un código de autenticación generado por el segundo dispositivo para el primer dispositivo. El código de autenticación puede enviarse al primer dispositivo junto con un identificador para el primer dispositivo. El identificador puede comprender un número de teléfono o número de serie del primer dispositivo. El primer secreto compartido puede comprender un número de cuenta personal "PAN" de una tarjeta bancaria asociada con la primera entidad. El primer secreto compartido puede comprender un número de serie codificado de una tarjeta bancaria asociada con la primera entidad.

El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para la "Internet de las Cosas".

El acceso de autenticación al servicio puede comprender el acceso de autenticación a una parte del servicio en función de la credencial y/o el contexto. La credencial puede comprender una primera credencial asociada con un dispositivo y un usuario principal del dispositivo. La credencial puede comprender además una segunda credencial asociada con un dispositivo y un usuario secundario del dispositivo. El acceso de autenticación al servicio basado en la credencial puede comprender el acceso de autenticación a diferentes servicios para el usuario primario y el usuario secundario en base a la primera credencial y la segunda credencial respectivamente. El dispositivo puede comprender una tarjeta bancaria y los diferentes servicios son límites de gasto diferentes para el usuario principal y el usuario secundario. La credencial se puede seleccionar en función del contexto. El servicio puede comprender una pluralidad de servicios seleccionados en función del contexto. Un administrador o usuario puede modificar, agregar o cancelar el contexto o la credencial. La credencial puede comprender al menos una de una

contraseña, PIN y/u otra credencial de autenticación directa. El contexto puede comprender al menos uno de un dispositivo que proporciona la credencial, una aplicación en el dispositivo, una red a la que se puede conectar el dispositivo, la ubicación geográfica del dispositivo y/o el servicio al que se accede.

El dispositivo puede comprender al menos uno de una computadora personal, un teléfono inteligente, una tableta inteligente o un dispositivo habilitado para la Internet de las Cosas.

El método puede comprender además agrupar una pluralidad de solicitudes en un mensaje por lotes en una memoria intermedia del primer módulo, poner en cola el mensaje por lotes para enviar al segundo módulo, establecer al menos un indicador del sistema que autoriza una función del sistema, verificando el al menos un indicador del sistema en el segundo módulo y el procesamiento del mensaje por lotes en el segundo módulo.

El método puede comprender además establecer al menos un canal de memoria compartida entre el primer módulo y el segundo módulo. El método puede comprender además el segundo módulo que responde al primer módulo a través del al menos un canal de memoria compartida. El al menos un canal de memoria compartida puede recibir y ensamblar el mensaje por lotes y la propiedad de la mano de la memoria al segundo módulo. El al menos un canal de memoria compartida puede recibir un mensaje por lotes a través de una pila de red del sistema informático. El al menos un canal de memoria compartida puede comprender una puerta de enlace HTTP. La puerta de enlace HTTP se puede utilizar como un servicio web.

Las comunicaciones pueden utilizar una contraseña autenticada de protocolo de intercambio de claves. El método puede comprender además el uso de redes de copia cero en una pila de red del sistema informático. El método puede comprender además el uso de redes en modo usuario en una pila de red del sistema informático.

El método puede comprender además la serialización de datos de modo que los componentes de la transmisión de datos desde el primer módulo se combinen como un único flujo de datos y

luego se separen en los componentes en el segundo módulo. La serialización se puede abstraer al final de cada módulo.

Una memoria intermedia de cada módulo puede tener un umbral configurable de almacenamiento en búfer. El primer módulo y el segundo módulo pueden estar ubicados en el mismo dispositivo informático. El primer módulo y el segundo módulo pueden estar ubicados en diferentes dispositivos informáticos.

Los datos transmitidos desde el primer módulo al segundo módulo pueden llevar una ID de versión. El método puede comprender además verificar que la ID de versión puede ser actual para los datos transmitidos desde el primer módulo al segundo módulo. El método puede comprender además, si se actualiza alguno de los datos, volver a verificar que el ID de la versión es actual. Si no se verifica la ID de la versión, la transmisión de datos puede fallar.

Al menos uno del primer módulo y el segundo módulo puede comprender al menos un módulo de servicio de datos, en donde cada actividad de datos dentro del sistema informático puede ejecutarse a través del al menos un módulo de servicio de datos. El al menos un módulo de servicio de datos puede configurarse para comunicarse con un almacenamiento de datos que puede ser implementado por un almacenamiento de base de datos central. El al menos un módulo de servicio de datos puede ser solo un componente del sistema informático con acceso directo al almacenamiento de datos. El almacenamiento de base de datos central puede comprender al menos una base de datos distribuida. La al menos una base de datos distribuida puede tener canales separados de acceso de lectura y escritura. El almacenamiento de datos puede proporcionar una interfaz para al menos una base de datos heterogénea. El almacenamiento de datos puede proporcionar una pluralidad de tipos de interfaz. La pluralidad de tipos de interfaz puede comprender al menos uno de una interfaz "SQL" de lenguaje de consulta estructurado, una interfaz de celda y columna, una interfaz de documento y una capa de interfaz gráfica sobre el almacenamiento de la base de datos de datos centrales. Todas las escrituras en la capa de almacenamiento de datos pueden ser administradas por un único módulo compartido que controla la totalidad o parte de una o más transacciones de datos.

El método puede comprender además operar al menos una copia de seguridad redundante del módulo compartido. Todos los cambios de datos pueden fluir a través del único módulo compartido en una secuencia rápida en serie. El único módulo compartido puede usar un modelo de redundancia de respaldo en línea, o dinámico, que se presenta como un grupo de transactor de datos, en donde el grupo transactor de datos puede ser un conjunto de módulos en una jerarquía y cada módulo puede configurarse para controlar las transacciones de datos si falla un módulo maestro. El método puede comprender además la partición de datos en módulos o almacenes de datos basados en reglas configuradas por dominio. El método puede comprender además la función de resumen (hashing) de datos dirigidos de un registro de una transacción de datos o de un registro de una transacción de datos principal. El resumen puede tener una función cardinal igual al número de particiones de datos. El método puede comprender además el resumen de datos dirigidos por al menos uno de área geográfica, apellido y/o moneda enumerados.

El método puede comprender además realizar al menos una transmisión de datos a través del al menos un módulo de servicio de datos a través de múltiples particiones de datos. El método puede comprender además completar al menos una transmisión de datos a través del al menos un módulo de servicio de datos por múltiples módulos. El método puede comprender además la persistencia de la al menos una transmisión de datos en al menos un módulo de servicio de datos en múltiples nodos de almacenamiento de datos en el almacenamiento de datos.

El sistema informático puede comprender una pluralidad de módulos de servicio de datos y cada módulo de servicio de datos aloja un motor de base de datos en memoria/en proceso, incluidas las representaciones en caché de todos los datos importantes para esa instancia. El sistema informático puede comprender una pluralidad de módulos de servicio de datos y cada módulo de servicio de datos puede comprender una pluralidad de motores de base de datos heterogéneos u homogéneos.

El método puede comprender además el uso de un sistema de versión con Control de Concurrencia Multiversión (conocido por sus siglas en Inglés "MVCC") para administrar la concurrencia del acceso al almacenamiento de datos, de modo que todas las lecturas de datos

sean coherentes y reflejen las escrituras de datos correspondientes exactamente. El método puede comprender además el uso de una consistencia pesimista para gestionar la concurrencia del acceso al almacenamiento de datos, de manera que un registro de datos debe escribirse en el almacenamiento de datos y confirmarse como escrito antes de que cualquier transacción de datos subsiguiente pueda acceder al registro de datos. El sistema informático puede comprender además una capa de aplicación y en la que la capa de aplicación no puede continuar con una transacción de datos hasta que al menos un módulo de servicio de datos confirme que ha escrito el registro y completado la transmisión de datos.

Todas las características opcionales de los aspectos 1 al 26 se relacionan con todos los demás aspectos mutatis mutandis. Se contemplan variaciones de las realizaciones descritas, por ejemplo, las características de todas las realizaciones descritas pueden combinarse de cualquier manera.

Breve descripción de los dibujos

Ahora se describirán realizaciones de la presente divulgación, solo a modo de ejemplo, con referencia a los dibujos adjuntos en los que se utilizan números de referencia similares para representar partes similares. En los dibujos:

La Figura 1 ilustra el concepto modular detrás del sistema Tereon;

La Figura 2 ilustra un ejemplo de la arquitectura del sistema Tereon;

La Figura 2a ilustra cómo Tereon abstrae sus servicios y dispositivos en dominios y contextos funcionales, dispositivos, componentes y protocolos;

La Figura 3 muestra las comunicaciones iniciadas a través de conexiones TLS a través de un proxy intermediario;

La Figura 4 ilustra el uso de la memoria compartida y el paso de mensajes a la memoria proxy;

La Figura 4a ilustra una memoria compartida y un módulo de transferencia de semáforos;

La Figura 5 ilustra una cadena de resumen que involucra cuatro cuentas;

La Figura 6 ilustra una cadena de resumen que involucra dos cuentas en el mismo sistema;

La Figura 6a ilustra una cadena de resumen que involucra tres cuentas en el mismo sistema donde las etapas de transacción se intercalan;

La Figura 7 ilustra la naturaleza dendrítica de los resúmenes de licencia;

La Figura 8 ilustra una cadena de resumen que involucra cuatro dispositivos que se desconectan por un tiempo;

La Figura 9 ilustra una función de búsqueda inversa implementada para dos servidores; La Figura 10 ilustra el establecimiento de comunicaciones entre servidores Tereon;

La Figura 11 ilustra las comunicaciones donde un usuario ha migrado a otro servidor;

La Figura 12 ilustra cómo el servicio de directorio puede dirigir un servidor solicitante a dos servidores diferentes;

La Figura 13 ilustra el caso en donde un servidor necesita obtener credenciales de tres servidores para construir una credencial multifacética;

La Figura 14 ilustra la relación de un usuario con un banco;

La Figura 15 ilustra el proceso emprendido para transferir una cuenta;

La Figura 16 ilustra el proceso realizado para cambiar un número de móvil registrado;

La Figura 17 ilustra el mantenimiento de un número de móvil previamente registrado para acceder a dos monedas;

La Figura 17a ilustra el mantenimiento de un número móvil previamente registrado para acceder a dos monedas, con cada moneda en un servidor separado;

La figura 18 ilustra un flujo de trabajo;

La figura 19 ilustra un flujo de trabajo alternativo;

La figura 20 ilustra un flujo de trabajo alternativo; y

La figura 21 ilustra un sistema informático ejemplar.

Visión general

La presente descripción se relaciona con un nuevo método de procesamiento de transacciones que elimina la necesidad de considerar o estar limitado por las compensaciones presentes como se describió anteriormente. Esta descripción proporciona un método para autenticar y procesar transacciones en tiempo real a una velocidad de varios órdenes de magnitud mayor de lo que es posible con los sistemas existentes, y liquidar, o procesar y completar esas transacciones en tiempo real.

La liquidación en tiempo real no se aplicaría solo a las transacciones financieras. Se aplicaría a cualquier transacción que requiriera, o se beneficiara, de parte o la totalidad de la autenticación, autorización, procesamiento y finalización inmediatos. Estos pueden abarcar desde el control de acceso hasta la validación de registros, el intercambio de registros y documentos, las instrucciones de comando y control, etc. Este método comprende siete áreas principales:

- Un método **para escribir transacciones compatibles con ACID en una escala extremadamente alta** para cualquier producto de base de datos
- **Una implementación de cadena de resumen** que ofrece autenticación de registros en múltiples libros de contabilidad privados con pruebas matemáticas completas dentro de los límites de una sola sesión en tiempo real y en una escala extremadamente alta.
- **Un servicio de directorio** que admite una red de malla de proveedores de servicios de transacción en lugar de implementar una arquitectura de "centro y radio" que crea grandes desafíos de escalabilidad.
- **Un marco extensible** que permite a un comerciante o dispositivo de usuario actualizar la aplicación (o "app") que utiliza para procesar transacciones por aire y de una transacción a otra.
- **Una capa de servicios de datos** que actúa como una matriz de traducción entre aplicaciones que admiten diferentes tipos de transacciones y una estructura de base de datos común.
- Un método **para reunir y presentar un conjunto de credenciales ad hoc** que permite a un servicio o dispositivo acceder a un conjunto de servicios o funciones.
- Un método para crear **comunicaciones seguras en tiempo real** en cualquier protocolo, incluyendo NFC (Near Field Communications) y USSD (Unstructured Supplementary Service Data).

Exclusivamente entre los métodos de procesamiento, el sistema de la presente divulgación proporciona un método para lograr el procesamiento y la finalización de transacciones en

tiempo real con un costo incremental de cero a medida que aumenta el número de transacciones.

Descripción detallada

El sistema Tereon es un motor de procesamiento y autenticación de transacciones electrónicas. Puede ser implementado como un sistema de procesamiento de pagos móviles y electrónicos. También se puede utilizar en otras implementaciones, por ejemplo, como parte de un sistema de comunicaciones IoT.

Tereon proporciona capacidad de transacción con cualquier dispositivo habilitado para IP (protocolo de Internet), y cualquier dispositivo que pueda interactuar con dicho dispositivo habilitado para IP. Todo lo que se requiere es que cada dispositivo tenga una ID única. Los casos de uso de Tereon van desde dispositivos IoT, hasta el acceso y la gestión de registros médicos, hasta pagos con algo tan común como un teléfono móvil, un terminal de pago o un cajero automático (ATM). En una implementación de ejemplo inicial, Tereon admite teléfonos móviles, tarjetas, terminales de punto de venta y cualquier ID de referencia única. Tereon proporciona la funcionalidad necesaria para permitir que los consumidores y comerciantes realicen pagos, reciban pagos, transfieran fondos, reciban fondos, realicen reembolsos, reciban reembolsos, depositen fondos, retiren fondos, vean datos de cuentas y vean mini-extractos de transacciones pasadas. Tereon admite transacciones entre divisas y transfronterizas. Por lo tanto, un consumidor puede tener una cuenta en una moneda, pero hacer un pago de transferencia en otra, por ejemplo.

En la implementación inicial de Tereon, si un usuario final puede realizar una transacción particular depende de la aplicación que esté usando en ese momento. Los comerciantes o las terminales comerciales pueden iniciar algunas transacciones, mientras que un dispositivo del consumidor puede iniciar otras.

Cuando se usa Tereon para procesar los pagos, las transacciones se pueden segmentar en los siguientes modos: realizar y recibir pagos, consumidor móvil a comerciante móvil, consumidor

móvil a portal de comerciantes en línea, consumidor móvil a comerciante móvil donde el cliente no está presente, cuenta de consumidor a comerciante desde dentro del portal de la cuenta, NFC - consumidor de tarjeta de Tereon a comerciante móvil, NFC u otro consumidor de tarjeta a comerciante de la tarjeta, transferir y recibir fondos, cuenta de consumidor a cuenta del consumidor desde el portal de la cuenta, consumidor móvil a consumidor móvil de igual a igual, consumidor móvil a consumidor de tarjeta de igual a igual, consumidor de tarjeta a consumidor móvil de igual a igual, consumidor de tarjeta a consumidor de tarjeta de igual a igual, consumidor móvil a no-usuario de igual a igual , consumidor de tarjeta a no usuario de igual a igual, no usuario a no usuario de igual a igual, no usuario a consumidor móvil de igual a igual, y no usuario a consumidor de tarjeta de igual a igual . Los no usuarios pueden referirse a alguien que no se haya registrado previamente con el servicio de pago, como un destinatario no bancarizado de una remesa.

Arquitectura del sistema

Internamente, un servidor Tereon consta de dos componentes principales, el motor de reglas de Tereon y el marco de servicios de aplicación de dispositivos inteligentes (SDASF).

El SDASF le permite a Tereon administrar cualquier número de dispositivos e interfaces diferentes. Lo hace al permitir que Tereon use y vincule una serie de capas abstraídas para definir cómo operan esos dispositivos e interfaces y, por lo tanto, se interconectan con Tereon.

Por ejemplo, todas las tarjetas bancarias utilizarán una capa básica de abstracción de tarjetas. La capa de abstracción de banda magnética se aplicará a las tarjetas con una banda magnética, la capa NFC a las tarjetas con un chip NFC y la capa de microprocesador a las tarjetas con un contacto de chip. Si una tarjeta usa las tres, Tereon definirá esa tarjeta con la capa de abstracción de la tarjeta principal y las tres capas de interfaz. La capa NFC no solo se aplicará a las tarjetas. También se aplicará a cualquier dispositivo capaz de soportar NFC, incluidos los teléfonos móviles. El SDASF utiliza estas capas de abstracción para crear módulos para cada uno de los dispositivos o interfaces.

Externamente, cada servicio y cada conexión a un dispositivo o red es un módulo. Por lo tanto, todos los módulos son servicios tales como el servicio de pagos de igual a igual, el servicio de depósito y los mini-estados de cuenta. También lo son las interfaces para fabricantes de tarjetas, bancos, proveedores de servicios, terminales, cajeros automáticos, etc. La arquitectura de Tereon puede soportar cualquier número de módulos.

Vista modular

La Figura 1 ilustra el concepto modular detrás de Tereon. Esencialmente, Tereon es una colección de módulos, la mayoría de los cuales comprenden módulos. Los módulos están definidos por los contextos y dominios funcionales dentro de los cuales operan, y por la lógica de negocios que determina las funciones que deben realizar. Estas funciones pueden ser cualquier tipo de transacción electrónica, como, por ejemplo, gestionar el funcionamiento y las comunicaciones entre dispositivos de baja frecuencia, gestionar y tramitar pagos electrónicos o digitales, gestionar y construir credenciales de identificación o autorización a pedido, o gestionar y operar cualquier otra forma de transacción electrónica o dispositivo.

Servidor de Tereon

Los módulos que conforman el servidor 102 de Tereon, como se muestra en la Fig. 1, pueden verse en dos niveles: el SDASF 104 y el motor de reglas 106. El propio motor de reglas 106 define los dominios y contextos funcionales de cada uno de los módulos 108 (algunos de los cuales se ilustran en la Fig. 1; estos incluyen los módulos que definen los servicios, los protocolos (no ilustrados), los dispositivos inteligentes, los terminales, etc.), y estos módulos 108 definen la estructura del SDASF 104. El SDASF 104 y los servicios e interfaces resultantes que admite luego definen los protocolos del sistema que están disponibles para Tereon. Estos protocolos luego definen las reglas y los servicios que Tereon puede admitir, por ejemplo, dispositivos inteligentes, terminales, etc., que a su vez definen los dominios y contextos funcionales que proporciona Tereon. Este enfoque circular o iterativo se utiliza para garantizar que las definiciones de los módulos y las funciones o los requisitos que admiten sean coherentes entre sí. Esto permite que los módulos se actualicen, actualicen y reemplacen in situ sin limitar el funcionamiento del sistema.

Los bloques y los módulos se interconectan entre sí mediante interfaces de programación de aplicaciones (API) abstraídas, que a su vez definen los dominios y contextos funcionales que proporciona Tereon. Cuando es posible, se comunican entre sí utilizando módulos de transferencia de semáforos a medida, un ejemplo de los cuales se exponen en la FIG. 4a y que se explicará más adelante, que también puede hacer uso de la memoria compartida. De esta manera, la operación interna y las funciones de los bloques y módulos se pueden actualizar o reemplazar sin comprometer la operación del sistema en su totalidad.

Componentes de infraestructura de marco

Los componentes de infraestructura también son modulares. En el caso de SDASF, este componente comprende módulos.

Interfaces múltiples

Cada interfaz se construye como un módulo separado que se conecta al servidor central. La estructura modular de Tereon, por lo tanto, le permite admitir múltiples interfaces, incluidos back office y sistemas centrales, tarjetas, centros de intercambio de información, comerciantes, teléfonos móviles, servicios, proveedores de servicios, almacenamiento, terminales, puertas de enlace SMS (Servicio de mensajes cortos), HLR (Registro de ubicación del hogar).) puerta de enlaces, etc.

Las interfaces de la base de datos admiten tanto la entrada en el lenguaje de consulta estructurado (SQL) como el análisis gráfico de los datos almacenados. Las interfaces también admiten el control de acceso a campos separados dentro de las bases de datos. Los diferentes roles de usuario y niveles de autorización pueden acceder a conjuntos de datos y campos definidos. El acceso está controlado por una variedad de medidas de seguridad. El acceso, la autenticación y la autorización se pueden entregar a través de una gama de enfoques estándar de la industria que incluyen las ACL (listas de control de acceso), LDAP (protocolo ligero de

acceso a directorios) y acceso personalizado basado en roles, como la seguridad de celdas y filas, y el acceso interfaces que están restringidas a roles individuales.

Portales de comercio electrónico

Tereon puede admitir portales de comercio electrónico a través de una API, de modo que un operador del portal puede crear un complemento para ese portal.

Motor de reglas

El motor de reglas 106 permite que se construyan nuevos servicios al unir los diversos componentes abstraídos para una transacción o para admitir un nuevo dispositivo. Las reglas definen la lógica de negocios para los servicios desplegados, y el proveedor de servicios puede adaptar estos servicios a usuarios individuales.

Las reglas se pueden definir en UML (Lenguaje de modelado unificado) o en un código que es similar al inglés simple. El motor analizará las reglas y creará los servicios a partir de los componentes abstraídos.

La naturaleza abstracta de los componentes permite crear nuevos módulos de servicio o dispositivo rápidamente. Esto permite a Tereon admitir nuevos servicios o dispositivos a medida que surja la necesidad de hacerlo.

Las interfaces internas de Tereon son independientes del protocolo, por lo que los módulos de protocolo externos pueden intercambiarse sin afectar la funcionalidad. Por ejemplo, para interactuar con un sistema central bancario, un protocolo de intercambio de datos personalizado podría usarse con una parte de una organización, y un módulo de protocolo ISO 20022 con otra.

El SDASF 104 permite a Tereon soportar múltiples dispositivos y protocolos inteligentes. La idea del SDASF 104 es abstraer las entidades en tipos de dispositivos y protocolos. El SDASF 104

define múltiples protocolos, y cada dispositivo llama al protocolo que requiera para un servicio o función en particular.

El SDASF 104 se puede ampliar agregando nuevos módulos a las instalaciones existentes sin afectar las operaciones de esa instalación. Permite que todos los servicios se definan en un servidor de back-office utilizando el método que se prefiera. Una vez instaladas en las terminales comerciales, las aplicaciones de la terminal Tereon se comunican con la SDASF para proporcionar los servicios al cliente.

La Fig. 2 establece la arquitectura del sistema Tereon 200. Cuando el diagrama y la narrativa se refieren a un componente particular a través de una solución particular, esto es simplemente porque estos son los componentes o lenguajes que se eligen en una realización. Los sistemas a medida pueden construirse para reemplazar estos componentes, o usar otros lenguajes y sistemas donde resulten más eficientes.

El Servidor Tereon

El servicio 202 de Tereon es una construcción lógica que se identifica como un artefacto monolítico. En realidad, puede existir como un conjunto de microservicios aislados, cada uno de los cuales puede diferir según la función y el alcance.

La Capa de Comunicaciones

La capa de comunicaciones 204 se inicia a través de conexiones TLS (seguridad de la capa de transporte) a través de un proxy intermediario. Esto también se muestra en la Fig. 3. TLS es un protocolo criptográfico que proporciona seguridad de comunicaciones a través de una red informática, generalmente una red TCP/IP (Protocolo de control de transmisión/Protocolo de Internet). Cada componente tiene una ACL (lista de control de acceso), que especifica a qué usuarios o procesos de sistemas pueden acceder o conectarse a un sistema, objeto o servicio. Esto garantiza que solo el intermediario pueda establecer una conexión original entrante, aumentando la seguridad intrínseca y reduciendo el perfil de amenaza. En este ejemplo, el

proxy utiliza una plataforma de puerta de enlace HTTP conocida en la técnica con servicios personalizados de Tereon personalizados.

Red DNS Privada

DNS 206 se utiliza como la base para el servicio de directorio 216. El servicio de directorio 216 es altamente redundante y está replicado en todas las ubicaciones geográficas. Sin embargo, su estructura y capacidades son muy superiores a todo lo que pueden ofrecer los servicios DNS existentes, tal como se establece a continuación.

Abstracciones

La Fig. 2a ilustra cómo Tereon abstrae sus servicios y dispositivos en dominios y contextos funcionales, tales como reglas y actividades de clientes o consumidores, reglas y actividades de comerciantes, actividades y reglas bancarias, actividades y reglas de transporte, funcionalidad y reglas de dispositivos, etc. La Fig. 1 ilustra cómo Tereon da efecto a estas abstracciones al abstraer los componentes y servicios del sistema en bloques o módulos funcionales.

Los módulos de Tereon se construyen a partir de estas abstracciones. Cada dispositivo, cada interfaz y cada tipo de transacción se abstraen en sus dominios y contextos. Estas abstracciones son reutilizables y pueden, donde sea significativo o permitido, interactuar con otros. Por ejemplo, los módulos de tarjeta de carga, tarjeta de crédito, tarjeta de débito y tarjeta de fidelidad utilizarán una serie de abstracciones comunes. Así también lo harán los módulos de pago y transferencia de fondos.

Protocolos

Cada uno de los protocolos 204 y 212 que admite Tereon se implementa como un módulo. Tereon pone estos módulos a disposición de aquellos servicios o componentes que los requieran.

Los sistemas heredados se esfuerzan por procesar transacciones simultáneas en los 100s o 1000s antes de que tengan que agregar hardware. En lugar de actualizar sus sistemas, los bancos se han basado en sistemas de liquidación periódica que requieren cuentas de conciliación y altos costos para cubrir la exposición crediticia hasta el momento de la liquidación. Tereon elimina la exposición crediticia y, por lo tanto, la necesidad de tales cuentas. Ofrece sistemas altamente asequibles a los que ahora se les pide que procesen 100.000 transacciones por segundo. Tereon está diseñado para desarrollar resiliencia, admitir 1.000.000 de transacciones por segundo por servidor y operar con hardware de alta calidad para productos básicos en lugar de confiar en hardware costoso. Tereon también admite la escala horizontal y vertical de forma casi lineal sin comprometer las garantías de ACID o su rendimiento en tiempo real.

El Subsistema de Licencias

El servidor de licencias Tereon 210 permite que los componentes del sistema se aseguren de que se están comunicando con sistemas pares autorizados, autorizados y legítimos, tanto dentro de una única instancia implementada, donde los microservicios de una sola instancia participan en las comunicaciones entre procesos en una sola máquina, independientemente de si la máquina es, por ejemplo, una máquina física, una máquina lógica, una máquina virtual, un contenedor o cualquier otro mecanismo comúnmente utilizado para contener código ejecutable y en cualquier número o tipo de máquinas, y en todas las instancias de implementación (por ejemplo, plataformas de clientes separadas que se comunican entre sí). La plataforma de licencias se implementa a través de una estructura de autoridad de certificación conocida en la técnica.

Cuando se instalan componentes en el sistema, estos comunican sus detalles de instalación (organización, tipo de componente y detalles, clave de licencia, etc.) junto con una solicitud de firma de certificado al servidor de licencias a través de una conexión segura y autenticada a intervalos predeterminados y configurables (por ejemplo, mensual, con un plazo de una semana).

El servidor de certificados compara esos detalles con su directorio de componentes autorizado, y en una coincidencia, otorga al dispositivo que inicia una solicitud de instalación un nuevo certificado, firmado con una clave de firma aislada y segura (generalmente a través de un módulo de seguridad de hardware) en una jerarquía de autoridad de certificación interna, utilizable por un período de tiempo prescrito (por ejemplo, un mes). Todos los relojes en los sistemas conectados están sincronizados.

La persona que llama puede utilizar el certificado como certificado de cliente al iniciar las comunicaciones con otros módulos y como certificado de servidor cuando actúa como el destinatario de las conexiones. El servidor de licencias, que nunca recibió la clave privada, no tiene detalles que permitan a otra parte hacerse pasar por este certificado, incluso si está comprometido. Si lo prefiere, la persona que llama puede solicitar dos certificados, un certificado de cliente y un certificado de servidor, de un servidor de licencias.

Cada componente puede validar que el servidor y los certificados de cliente han sido firmados por un agente de la autoridad de certificados autorizada y confiable, y pueden comunicarse con una confianza significativa de que no están sujetos a ataques o vigilancia de intermediarios y que la contraparte es quien dice que es. Cada certificado se otorga con metadatos de códigos de uso que limitan la forma en que cada módulo puede presentarse; por ejemplo, como un servidor de búsqueda para una organización específica. La organización tiene la seguridad de que todas las partes operan en casos autorizados, legalmente válidos.

La mayoría de los certificados simplemente caducan y nunca se renuevan, habiendo sido otorgados por un plazo fijo. Sin embargo, en el raro caso de que un certificado esté comprometido, o que se haya terminado o suspendido una licencia, se usa una lista de revocaciones y se distribuye de forma asíncrona a los servicios de proxy según sea necesario. Siempre se mantiene un directorio de certificados activo, utilizable para auditorías periódicas.

Además de los beneficios de validación bidireccional (que el cliente es quien dice ser y el servidor en cada conexión es quien informa), esta implementación permite que los componentes se comuniquen de forma segura sin que cada conexión de conexión requiera

comunicaciones con servidores de licencias remotos, que se comunican de forma segura sin reducir potencialmente la confiabilidad general de la plataforma.

Comunicaciones sitio a sitio

Las comunicaciones de sitio a sitio se facilitan a través de una instancia de puerta de enlace HTTP identificada y expuesta 212, que ejecuta la funcionalidad de copia cero personalizada y el modo de usuario opcional. Esta es la plataforma a través de la cual los dispositivos móviles, terminales y otras partes externas se comunican con las instancias, además de la conectividad de sitio a sitio. Esto incluye la detección de intrusiones estándar de la industria, la limitación de velocidad y la protección contra ataques DDOS (denegación de servicio distribuida), la descarga de cifrado de hardware, etc. Funcionalmente, el mecanismo de proxy de instancia lógica es de gran tamaño y es compatible con todas las mismas funciones, incluidos los certificados cliente/servidor y la validación, al mismo tiempo que se usa una autoridad de certificación reconocida externamente ante terceros.

El Servicio de Datos de Tereon.

Una de las características clave del sistema Tereon es que puede manejar significativamente más transacciones (en términos de rendimiento) que los sistemas anteriores. Esto se debe a un diseño único que implementa una red de procesamiento altamente concurrente, rápida y escalable, que puede procesar datos y transacciones, y una capa de servicios de datos extremadamente eficiente, así como algoritmos y módulos personalizados que minimizan la sobrecarga de procesamiento.

Las características de rendimiento descritas están dirigidas principalmente a la ampliación, haciendo más en una determinada pieza de hardware informático, lo que conduce a reducciones significativas en los costos de funcionamiento y el consumo de energía. Sin embargo, el diseño no se limita a un solo sistema; El sistema Tereon es capaz de escalar vertical y horizontalmente en gran medida, con cada servicio capaz de ejecutarse simultáneamente en una gran cantidad de dispositivos.

Para lograr altos niveles de rendimiento en un solo sistema o servidor, el sistema minimiza preferiblemente su sobrecarga de procesamiento al evitar la serialización innecesaria, evitando el procesamiento innecesario de la corriente, evitando copias innecesarias de la memoria, evitando las transiciones innecesarias del usuario al modo kernel, evitando cambios de contexto innecesarios entre procesos y evitando E/S aleatorias o innecesarias. Cuando un sistema lo hace correctamente, es posible alcanzar niveles extremadamente altos de rendimiento transaccional en ese sistema.

En un modelo tradicional, el servidor A recibiría una solicitud. Luego construiría y serializaría una consulta al servidor B, e inmediatamente la enviaría al servidor B. El servidor B descifraría (si fuera necesario), deserializaría e interpretaría esa consulta. Luego generaría una respuesta, serializaría y, si fuera necesario, cifraría esa respuesta, y luego transmitiría esa respuesta al servidor A o a otro servidor. Los cambios de contexto de kernel y proceso ocurren en las docenas por mensaje, el mensaje individual se emite en varias formas varias veces y la memoria se copia entre varios buffers de trabajo. Estos cambios de contexto de proceso y kernel imponen una gran sobrecarga de procesamiento por mensaje procesado.

Arquitectura de Comunicaciones

El sistema Tereon logra su rendimiento al reestructurar la forma tradicional en que los datos y las comunicaciones son manejados por el sistema. Cuando es posible, Tereon omite el sistema operativo kernel para evitar la sobrecarga de procesamiento impuesta por el kernel y para evitar los problemas de seguridad que a menudo surgen con los modelos estándar de administración de datos.

Cada actividad de datos dentro del sistema se ejecuta a través de una instancia de servicios de datos 214. Esta es una capa de servicios de datos orientados a servicios escalados que es el único componente del sistema con acceso directo a la plataforma de datos. Por lo tanto, todas las actividades de datos en el sistema deben pasar a través de él.

La capa de servicio de datos 214 se comunica con una capa de almacenamiento de datos 220 a través de canales separados de acceso de lectura y escritura 226. La capa 220 de almacenamiento de datos se implementa sobre un almacenamiento 224 de base de datos central, que comprende al menos una base de datos distribuida. Estas bases de datos no necesitan ofrecer garantías ACID; Esto es administrado por la capa de almacenamiento de datos.

Todas las escrituras en la capa 220 de almacenamiento de datos son administradas por un único transactor compartido, a través del cual todos los cambios de datos fluyen en una secuencia rápida en serie para preservar la causalidad. El diseño del transactor utiliza un modelo de redundancia de copia de seguridad en línea que se presenta como un clúster de transactor de datos 222. Si un transactor falla o se detiene por algún motivo, uno de los otros transactores asumirá el control de inmediato.

Si bien la plataforma de datos admite la partición para todos los dominios de datos, esa compatibilidad no se muestra en una figura. Si, en cualquier caso, una sola capa de almacenamiento de datos (respaldada por nodos de datos ilimitados) se consideraba prohibitiva o si existían razones reglamentarias para hacerlo, los datos se pueden dividir por medios imperativos o declarativos para almacenar en diferentes grupos de datos utilizando diferentes transactores. Por ejemplo, un sitio puede tener cuatro plataformas de datos, dividiendo a los clientes según criterios geográficos o jurisdiccionales, o para cuentas que comienzan con 1 -5 para ir en una, 6-0 en otra. Hay ramificaciones de procesamiento para esto, pero esto está soportado por la plataforma.

La Fig. 3 muestra comunicaciones a través de la capa de comunicaciones 204 que enruta las comunicaciones hacia y desde la capa de servicios de datos 214. Cuando un módulo 350 necesita comunicarse con otro módulo 360, primero inicia una conexión con un proxy 370, pasa su certificado de cliente para autenticarse en el paso 302 y luego comprueba que el certificado de proxy es válido y confiable en la acumulación en el paso 304. El módulo 350 pasa el mensaje en el paso 306 al proxy 370. El proxy 370 establece una conexión de correlación con el módulo de destino 360 en el paso 308; primero se autentica a sí mismo en 308 y valida que el

certificado del módulo es válido y confiable en el paso 310. El proxy 370 luego pasa los detalles confirmados del iniciador (módulo 350) en el paso 312, antes de recibir la respuesta del módulo en el paso 314. El proxy 370 devuelve los detalles del objetivo (módulo 360) y su respuesta en el paso 316. Esto establece un canal de comunicaciones entre el módulo 350 y el módulo 360 a través del proxy 370, con ambos módulos autenticados e identificados entre sí con un alto grado de confianza y, cuando sea necesario, con todas las comunicaciones y los datos cifrados. El proxy 370 transmite los mensajes del módulo 350 en el paso 318 al módulo de destino 360 en el paso 320, y transmite la respuesta del módulo de destino en el paso 322 al módulo 350 en el paso 324.

Estas conexiones utilizan una función de mantenerse vivas (conocida en Inglés como keep-alive) y sesión compartidas según los detalles de los certificados de la persona que llama y del destinatario (por ejemplo, el módulo 350 puede "cerrar" la conexión con el módulo 360 de destino, a través del proxy 370, y volver a abrirlo sin crear una nueva conexión de extremo a extremo. La conexión nunca sería compartida por ningún otro circuito). El proxy de comunicación 370 puede ser una puerta de enlace HTTP, o algún otro módulo o componente adecuado.

Tal arquitectura tradicionalmente tiene un costo de rendimiento significativo con un uso intensivo de la memoria. Para que el módulo 350 se comunice con el módulo de destino 360, tradicionalmente necesitaría serializar la carga útil, cifrar la carga útil, transmitirla al proxy 370, donde el proxy 370 descifraría la carga útil, deserializaría e interpretaría el contenido, reserializaría la carga útil, y lo cifraría para el módulo de destino 360, antes de pasarlo al módulo 360 de destino. El módulo de destino 360 descifraría el contenido, deserializaría e interpretaría el contenido.

El sistema Tereon usa varias técnicas para reducir la latencia media y máxima, para reducir la carga de memoria y para mejorar el rendimiento de una sola plataforma en el hardware básico. Esto logra un rendimiento monolítico en proceso al mismo tiempo que mantiene todos los beneficios de seguridad, mantenimiento y despliegue de los microservicios. Lo hace sin comprometer los altos niveles de seguridad y control que dicho sistema debe proporcionar.

Tereon puede usar un modelo de mensajería por lotes sobre la capa de comunicaciones como se establece en la Fig. 3. Cada mensaje pasado, como el mensaje pasado del módulo 350 al proxy 370 en el paso 306, podría ser un lote de mensajes. Tereon puede, sin embargo, ir mucho más lejos que esto.

Además de la mensajería por lotes, la Fig. 4 muestra cómo dos servidores de módulos pueden comunicarse entre sí a través de un módulo proxy (el módulo de transferencia personalizado) para negociar un canal de memoria compartida entre ellos. Los pasos 402 a 412 son análogos a los pasos 302 a 312 en la Fig. 3 con la adición de que, si es necesario, los atributos del servicio se verifican para confirmar que coinciden con la solicitud del cliente, algo que también puede ocurrir en los pasos 302 a 312. El módulo 450 a la instancia del módulo 460 puede usar TLS, o TLS HTTPS tradicional, de manera óptima con el modo de usuario de la puerta de enlace HTTP y la copia cero también para las transacciones de la persona que llama.

Si el módulo de origen 450 y el módulo de destino 460 son locales, luego de establecer la conexión a través del proxy 470 de los pasos 402 a 412, el que llama y el destinatario pueden solicitar opcionalmente conectividad directa entre sí a través de la memoria compartida, y aquí es, con esta solicitud opcional, donde este método difiere del método establecido en la Fig. 3. Si la persona que llama y el destinatario solicitan conectividad directa entre sí, luego de las negociaciones, se transmite un canal compartido desde el módulo 460 en el paso 414 al proxy 470, y desde el proxy al módulo 450 en el paso 416, y los dos módulos desde ese punto al usar un mecanismo de proceso directo a directo que nuevamente utiliza semáforos y memoria compartida. Esto se ilustra mediante los mensajes entre el módulo 450 y el módulo 460 en los pasos 418, 420, 422 y así sucesivamente.

En el modelo de Tereon, el servidor 450 procesa por lotes una pluralidad de solicitudes en búferes de memoria nativa como óptimo para la tarea, pone en cola el mensaje para el servidor 460 y dispara un semáforo. El servidor 460 comprueba los indicadores, procesa la memoria directamente compartida y responde en la memoria compartida. La conexión utiliza la memoria

viva y compartida según los detalles de los certificados de la persona que llama y del destinatario, y la memoria compartida y los semáforos para la comunicación.

Al utilizar el método anterior, la comunicación puede sortear la sobrecarga de la serialización y la transmisión (dado que está contenida dentro de una máquina), a un destino seguro, controlado por ACL y de una sola persona que llama. No necesita cifrado; la conexión ha sido validada, autenticada y autorizada en la configuración y no puede ser usurpada y, cuando corresponda, los procesos pueden compartir estructuras de memoria propietarias mayoristas cuando sea apropiado.

Tanto el proxy 470 como los módulos de código de Tereon (450 y 460) admiten redes de copia cero y redes de modo de usuario siempre que sea posible (cuando se compilan con la biblioteca TCP/IP necesaria, un proxy HTTP puede proporcionar una solución que evite el costo significativo de Conmutadores de contexto del kernel para paquetes de red). Esto se facilita a través del código específico del controlador de red que pueden utilizar el proxy 470 y los módulos de código Tereon. Esto minimiza el uso de memoria para solicitudes y respuestas de paquetes pequeños; estos comprenden la gran mayoría de las operaciones de Tereon, donde la mayoría de las operaciones pueden caber en un solo paquete TCP.

La Fig. 4a ilustra cómo el sistema Tereon implementa un conjunto de módulos de transferencia de semáforos a medida 408a, que también puede hacer uso de la memoria compartida, que se utiliza para intercambiar datos de manera eficiente entre cualquiera de los dos componentes del sistema Tereon (como la puerta de enlace HTTP 406a y los microservicios 410a que proporcionan la funcionalidad dentro de Tereon). En la Fig. 4a, la capa de servicios de datos 214 está incorporada por los microservicios 410a. Sin embargo, los microservicios podrían representar cualquier tipo de módulo de servicio.

La pila de red 404a (incluido un dispositivo virtual de bucle invertido) recibe y ensambla la solicitud de un servidor 402a conectado y, en lugar de copiarla en la memoria de destino en modo de usuario, simplemente entrega la propiedad de la concesión de memoria al destinatario, en este caso la puerta de enlace HTTP 406a. Esto es principalmente beneficioso

bajo una carga muy pesada (por ejemplo, millones de solicitudes por segundo) donde comienza a producirse la saturación del ancho de banda de la memoria.

Un módulo 406a de puerta de enlace HTTP ascendente de Tereon personalizado permite que las instancias locales (en relación con la instancia de puerta de enlace HTTP, donde generalmente hay una instancia de puerta de enlace HTTP en cada contenedor o en cada máquina física, lógica o virtual) la opción de usar la memoria compartida y el paso de mensajes a la memoria proxy desde la puerta de enlace al módulo, y viceversa para esa conexión ascendente. En lugar de que la puerta de enlace HTTP 406a serialice una solicitud y la pase a través de los mecanismos tradicionales, cuando se configura para un proveedor ascendente de memoria compartida, la puerta de enlace HTTP 406a utiliza la memoria compartida que la pasa al destinatario.

En este caso, la memoria compartida puede haberse configurado utilizando otra puerta de enlace HTTP, una instancia de puerta de enlace HTTP u otro elemento como un proxy. El uso de una puerta de enlace HTTP puede ser particularmente eficiente.

En lugar de utilizar los enlaces de comunicaciones proporcionados por el sistema operativo kernel, cada módulo de intercambio de datos pasa por alto el sistema operativo kernel; esto aumenta el rendimiento del sistema al evitar la sobrecarga del kernel y aborda las áreas de inseguridad que pueden ocurrir cuando los datos se transfieren hacia y desde los servicios proporcionados por el kernel. Dentro de Tereon, se usa un módulo, por ejemplo, para intercambiar datos de manera eficiente desde un componente del sistema directamente a la capa de servicios de datos 214 y desde la capa de servicios de datos 214 a un componente del sistema.

Otro ejemplo de la ventaja que brinda esta arquitectura es la eficiencia mejorada de la puerta de enlace HTTP 406a, que se logra mediante el uso del módulo de transferencia 408a que permite que la puerta de enlace HTTP 406a entregue todos los datos entrantes a los microservicios 410a, como la capa de servicio de datos 214 u otros componentes, y todos los datos salientes de los microservicios 410a o la capa de servicio de datos 214, a la puerta de

enlace HTTP 406a. En lugar de utilizar los datos de la puerta de enlace HTTP predeterminada y la entrega de mensajes, que en sí misma es eficiente, el módulo de transferencia de semáforos, que también puede usar la memoria compartida, permite que los datos se entreguen directamente a la capa de datos 214 y desde la capa de datos 214 a la puerta de enlace HTTP 406a, sin pasar por el núcleo. Esto no solo aumenta el rendimiento del sistema; sino que tiene una ventaja adicional en que esto asegura una de las áreas comunes de vulnerabilidad en los sistemas que usan puertas de enlace HTTP.

Ya sea el módulo que proporciona el canal de memoria compartida o el módulo que se comunica con el canal de memoria compartida cualquiera puede realizar un proceso por lotes y serializar o deserializar y separar las solicitudes. El módulo que realice esa tarea se reducirá a la función de ese módulo y la sobrecarga de procesamiento en la que incurre el módulo en su funcionamiento normal. Por ejemplo, en un caso, un módulo que a su vez recibe una gran cantidad de mensajes (que pueden o no ser pedidos) puede pasar sus mensajes a un módulo de memoria compartida que, por sí mismo, procesará por lotes y serializará esos mensajes para el módulo receptor, como la sobrecarga de procesamiento por lotes y serialización puede impedir que ese módulo procese los mensajes de manera eficiente y en carga. En otro caso, un módulo puede agrupar y serializar sus mensajes a un destinatario en particular antes de pasar ese lote a ese destinatario a través de un canal de memoria compartida.

En otro caso más, un módulo que pasa mensajes a un módulo receptor puede depender del módulo que proporciona el canal de memoria compartida para agrupar y serializar los mensajes, pero el módulo que recibe los mensajes por lotes puede deserializar y separar los mensajes. La cuestión de qué módulo lleva a cabo la tarea de agrupar y serializar, deserializar y separar se reduce a qué opción proporciona el nivel de rendimiento óptimo para las funciones que realizan los módulos. El orden de los lotes y la serialización dependerán del tipo de mensaje y de las funciones proporcionadas por los módulos de comunicación.

El sistema Tereon utiliza una puerta de enlace HTTP 406a para enmascararse como un servicio web y evitar así posibles problemas con los operadores de red que bloquean servicios no

estándar. Tereon puede, por supuesto, hacerse pasar por cualquier otro servicio si es necesario, y por lo tanto puede trabajar con configuraciones de seguridad de red conocidas con facilidad.

Siguiendo este diseño, el sistema aplica este enfoque modular a través de toda la arquitectura, donde el sistema utiliza módulos diseñados para explotar los recursos disponibles, y para evitar la sobrecarga del núcleo cuando sea posible. Un ejemplo adicional es el sistema de redes, donde Tereon hace uso, cuando es posible, de módulos que admiten redes en modo usuario o redes de copia cero en la pila de redes 404a. Esto evita la pesada carga de uso del núcleo para la creación de redes. El diseño modular también permite a Tereon operar en múltiples tipos de sistemas, donde módulos similares a medida proporcionan una funcionalidad similar y se pueden personalizar para cada sistema operativo o configuración de hardware.

La utilización de un intermediario de la manera representada en la Fig. 3 y la Fig. 4 permite un punto de control centralizado para todas las comunicaciones, ya sea dentro o fuera de la máquina. Es un único punto de control para los controles de velocidad y seguridad, el monitoreo y la auditoría, y para reglas o re-instrucciones especializadas. Esto permite flexibilidad en la implementación de sistemas, incluso cuando esos sistemas están operativos, sin incurrir en tiempo de inactividad o riesgos significativos. También facilita de manera sencilla el equilibrio de carga y las redundancias sin que el cliente tenga conocimiento o sienta complejidad.

Si el módulo 350 de la Fig. 3 desea hablar con el módulo 360 de destino, el uso de un intermediario permite que el módulo 360 de destino tenga un equilibrio de carga en las máquinas "n" y se mueva a través de cualquier número o tipo de máquinas sin reconfigurar todos los clientes potenciales, en lugar de eso simplemente reconfigura el intermediario.

El sistema utiliza un protocolo PAKE (intercambio de claves autenticado por contraseña) que se creó para proporcionar a dos partes comunicantes la capacidad de autenticar mutuamente su intercambio de claves. Esto es imposible con otros protocolos conocidos de intercambio de claves públicas, como el protocolo de intercambio de claves Diffie-Hellman, que hace que esos

protocolos sean vulnerables a un ataque de hombre en el medio. El protocolo PAKE, si se usa correctamente, es inmune a los ataques de cualquiera que se meta en el medio.

Cuando Tereon se comunica con sistemas externos, como un dispositivo o servidor externo, agrega una capa adicional al sistema de comunicaciones. Muchos protocolos de intercambio de claves son teóricamente susceptibles a los ataques de personas metidas en la comunicación. Una vez que se establece una conexión, utilizando los certificados y los mensajes firmados para confirmar que la comunicación es entre dos entidades conocidas, el sistema utiliza el protocolo PAKE para establecer una segunda clave de sesión segura y, por lo tanto, hacer que las comunicaciones sean impermeables al ataque de una persona que se meta en la comunicación. Por lo tanto, las comunicaciones utilizarán la clave de sesión TLS y luego la clave de sesión del protocolo PAKE para cifrar o encriptar todas las comunicaciones.

Cuando la comunicación es con dispositivos que tienen una cadena de identidad inviolable, se puede prescindir del TLS si es necesario y, en su lugar, se usa el protocolo PAKE como protocolo de clave de sesión principal. Esto puede ocurrir, por ejemplo, cuando los dispositivos son pequeños sensores de hardware que forman un conjunto de componentes de la Internet de las Cosas.

Métodos de Comunicación

El servicio de datos 214 de Tereon se basa en un almacenamiento de valores clave con funcionalidad de gráfico que ofrece una redundancia $n + 1$ o superior y replicación de sitios múltiples opcional, y que ofrece garantías ACID completas a través de un transactor coordinador (un dispositivo o módulo que lleva a cabo, gestiona, o controla, la totalidad o parte de una o más transacciones). El servicio de datos 214 está encapsulado en un servicio de dominio de datos que, además de la funcionalidad de memoria compartida, ofrece además funcionalidad de copia cero y escalado de lectura ilimitado, almacenamiento en memoria caché y niveles extremadamente altos de rendimiento de escritura. Esto se mantiene en un grupo o clúster de datos de tamaño variable, con un gran almacenamiento de memoria caché. En

circunstancias altamente únicas, el servicio de datos puede ser evitado para el uso directo de las tiendas de valor clave.

El servicio de datos 214 ofrece tanto una funcionalidad de estilo SQL tradicional de alto rendimiento como un procesamiento de gráficos para admitir funciones como el análisis de flujo de dinero. El servicio de datos 214, junto con la arquitectura de comunicaciones del módulo de muy alto rendimiento (que proporciona la eficiencia y el rendimiento de la plataforma), proporciona un diseño extremadamente eficiente que ha superado los 2,8 millones de transacciones por segundo en pruebas en hardware de servidor de productos básicos (con una red de 10 Gbps consolidada).

Mediante la implementación de las siguientes prioridades de arquitectura, el sistema puede reducir drásticamente la cantidad de cambios de contexto del sistema operativo kernel y proceso necesarios para procesar los mensajes transmitidos dentro del sistema y entre sistemas:

- a) La red de copia cero está disponible para minimizar los costos de transporte desde el borde de la red hasta los servicios.
- b) La red en modo usuario está disponible para minimizar los costos de transporte desde el borde de la red hasta los servicios.
- c) Cuando es necesaria la serialización (principalmente cuando se cruzan los límites de la máquina o el servidor), se usa una serialización de alta eficiencia, por ejemplo, los búferes de protocolo o Avro, a diferencia de una serialización de alta sobrecarga, como el Protocolo Simple de Acceso a Objetos (conocido por sus siglas en Inglés SOAP). Esto se abstrae en el borde de cada servidor, de modo que un servidor determinado puede comunicarse fácilmente a través de Internet con un servidor de otro continente, aunque a un nivel de rendimiento y eficiencia más bajo.
- d) Los servidores tienen un umbral configurable de almacenamiento en búfer en donde intentarán agrupar las solicitudes para minimizar los cambios de contexto del proceso y maximizar la coherencia de la memoria caché para cualquier servidor dado. Si el servidor A tiene 10,000 solicitudes que llegan dentro de un período de 20 ms, para

dar un ejemplo, y la plataforma apunta a una ventana de búfer de 20 ms, y necesita la asistencia del servidor B para esas 10,000 solicitudes, entonces reunirá las 10,000 solicitudes en una sola solicitud, y luego pone en cola el mensaje asíncrono para el servidor B, marcando el semáforo. El servidor B luego puede procesar rápidamente las 10000 solicitudes, proporcionando una respuesta única al servidor A. Esto se puede configurar basándose en la eficiencia de optimización frente al tiempo de respuesta máximo.

En la práctica, la reducción de la cantidad de cambios de contexto de kernel y proceso ha producido enormes mejoras en el nivel de rendimiento de la plataforma. En lugar de incurrir en una serie de cambios de contexto de kernel y proceso por mensaje, el modelo de Tereon incurre en esa cantidad de interruptores de contexto de kernel y proceso por bloque de mensajes, debido a la comunicación por lotes de mensajes. Las pruebas indican que al usar este modelo, la diferencia de rendimiento entre el modelo tradicional y el modelo de Tereon es de 1:1000 y más para muchas cargas de trabajo.

Sin embargo, los módulos y sus beneficios no están restringidos a sistemas únicos. Incluso cuando el servidor A y el servidor B existen, por ejemplo, en máquinas separadas, el sistema Tereon seguirá utilizando una serialización y procesamiento por lotes eficientes. Independientemente de si esto se combina con las redes opcionales de copia cero o en modo de usuario, el modelo de Tereon mejora dramáticamente el rendimiento de la red y el procesamiento.

Las pruebas han mostrado que estos elementos de diseño han demostrado hacer operaciones de servidor local a servidor en decenas de millones de solicitudes de mensajes y respuesta en viajes de ida y vuelta por segundo (en modo de memoria compartida, por lotes), y en millones por segundo a través de un cable de red de alta velocidad (por ejemplo, 10Gbps en condiciones de consolidado).

Debido a que todas estas transacciones se pueden manejar en tiempo real y reconciliarse de inmediato, hay muchas ventajas, en particular para los entornos bancarios, de IoT, médicos, de

identificación, transporte y otros entornos que requieren un procesamiento de datos preciso. Específicamente, dichos sistemas no concilian actualmente las transacciones en tiempo real. En cambio, las transacciones se reconcilian después de un período de tiempo, a veces en lotes. Es por eso que, por ejemplo, las transacciones financieras generalmente se procesan en lotes con procesos de conciliación separados que se ejecutan después del horario de atención. Al utilizar el sistema Tereon, es posible que los bancos concilien todas las transacciones financieras en tiempo real de una manera que antes no era posible. Entonces es posible evitar la necesidad de que los bancos tengan cuentas de conciliación para cubrir transacciones financieras que aún no se hayan reconciliado, o que no puedan conciliarse con precisión, ya que, por definición, todas las transacciones se habrán conciliado cuando se procesen.

Transacciones y Partición de Datos.

Todas las actividades atómicas en el sistema de Tereon son transacciones: tienen éxito en su totalidad, o fallan en su totalidad, como un requisito fundamental de cualquier sistema que respalde las garantías de ACID para las transacciones. Esta sección explica brevemente cómo se logra esto, y establece los detalles del enfoque que Tereon ha aplicado a las transacciones y la partición de datos para mitigar los efectos de la partición en el logro de las garantías de ACID para las transacciones.

Como se mencionó anteriormente, cada actividad de datos dentro de la plataforma Tereon se ejecuta a través de una instancia 214 de Tereon Servicio de Datos (Tereon Data Service), que a su vez puede operar como un conjunto de microservicios 410a. Este es un sistema orientado a servicios escalado que es el único componente del sistema con acceso directo a la plataforma de datos y, por lo tanto, todas las actividades de datos deben pasar a través de él. Estos servicios de datos se escalan de manera tal que las transacciones paralelas dentro del sistema se pueden realizar a través de diferentes instancias de servicios de datos, utilizando MVCC (Control de Concurrencia de Multiversión) de datos en caché de instancia para tener siempre datos de lectura coherentes.

Las actividades de datos ocurren a través de mensajes atómicos a una instancia de servicio de datos, con el mensaje que contiene la totalidad del trabajo de datos; por ejemplo, un trabajo puede implicar la lectura de varios registros y atributos correlacionados, o la actualización o inserción de datos basados en datos dependientes, o una combinación de tareas. La instancia del servicio de datos ejecuta el trabajo como una transacción de confirmación de dos fases en todos los almacenes de datos transaccionales de respaldo.

El modelo de Tereon garantiza la consistencia de los datos mediante las siguientes técnicas:

- a) Cualquier conjunto de datos de lectura lleva una ID de versión.

Todas las escrituras (actualizaciones e inserciones dependientes) verifican que este ID de versión sea actual para todos los datos relevantes como una transacción optimista. Esto significa que si una fuente lee tres registros para obtener varios atributos de cuenta (por ejemplo, permisos, saldo y datos de moneda), entonces este grupo de datos tiene una ID de versión coherente. Si alguno de esos valores se actualiza, o se escriben datos dependientes (por ejemplo, una transferencia financiera), la ID de la versión se confirma nuevamente como actual y si difiere, digamos que las suposiciones cambiarías o los tipos de cambio se modificaron, la escritura en su conjunto, falla por completo. El servicio en sentido descendente vuelve a leer, si corresponde, y evalúa si los datos modifican la transacción de alguna manera material. Si no es así, la transacción se envía de nuevo. Si, nuevamente, la transacción falla, se repite hasta que se excede el número configurable de intentos de reintento y se emite una falla de hardware. Una falla dura sería extremadamente improbable en circunstancias normales.

En la gran mayoría de los escenarios del mundo real, una transacción optimista fallida nunca ocurriría, incluso a través de enormes volúmenes de transacciones y diversidad de cuentas. Y en el raro caso de que uno lo hiciera, los datos nunca se ven comprometidos y hay una sobrecarga de procesamiento mínima. Este modelo MVCC/optimista también protege completamente los registros eliminados, dado que

la plataforma que se utiliza es una base de datos de historial perpetuo (más allá de las eliminaciones reglamentarias que pueden requerirse en circunstancias excepcionales).

b) Escribe en la plataforma, para una partición de datos dada (que es un concepto separado de la escala horizontal de los servicios de datos).

Muchas instancias de servicio de datos pueden escribir y leer desde una partición de datos, y una única instancia de servicio de datos puede almacenar y leer desde múltiples particiones de datos. Todas las lecturas y escrituras se producen a través de una única instancia de transactor maestro 222 con una o más copias de seguridad operativas redundantes según sea necesario. Sin embargo, solo una instancia está siempre activa. Esto garantiza que la validez transaccional y causal se mantenga en todas las circunstancias (por ejemplo, sin sesgo durante una división de la red, por ejemplo, o durante un breve retardo de comunicación). Este transactor confirma todas las transacciones optimistas como válidas o no, y actualiza continuamente los administradores de caché en las instancias del servicio de datos con información actualizada y actual como contextualmente importante para esa instancia.

c) Partición de datos opcional

La limitación a un solo transactor podría potencialmente limitar la escalabilidad para instancias muy grandes de Tereon (entendiendo que una sola organización podría administrar múltiples instancias de Tereon por región, etc.). La partición de datos es la noción de que un grupo de Tereon Data Service puede particionar los datos en los transactores 222 o en los almacenamientos de datos 224 según las reglas de Tereon configuradas por dominio. La plataforma Tereon admite las siguientes reglas de partición actualmente, como una estrategia de función de resumen multicomponente heterogénea:

i) Función de resumen de datos específicos de un elemento determinado o de cualquier elemento superior (por ejemplo, resumen de detalles según el registro principal). La

función de resumen de alto rendimiento tiene una cardinalidad igual al número de particiones.

Actualmente, el sistema no proporciona el rebalanceo, por lo que en la implementación actual el resumen debe estar al frente, aunque el rebalanceo se proporcionará en una implementación futura (aunque las particiones aún se pueden agregar en la actualidad, usando una regla de varias partes que incluye función de resumir por fecha y hora de origen).

ii) Los datos configurados contienen datos específicos de un elemento dado o de cualquier elemento superior, p. ej. por área geográfica enumerada. Por apellido A-K o L-Z, etc. por moneda, etc.

La función de resumen objetiva de los datos admite rangos de códigos de caracteres alfanuméricos, Unicode y otros, rangos enteros, rangos de punto flotante y conjuntos enumerados.

iii) Combinaciones de las anteriores.

En una implementación, por ejemplo, las dos letras A y B pueden referirse a dos conjuntos de datos separados que son comunes en toda una región geográfica, con los números 1 y 2 en referencia a dos divisiones de esa región. Una sola regla de partición puede admitir, por ejemplo, particiones entre las particiones de nivel superior 1AB y 2AB a través de una regla de datos, como una región geográfica, y luego particiones adicionales entre las subparticiones A y B a través de un resumen de número de cuenta.

d) Un solo trabajo realizado a través de una única instancia de servicio de datos puede cruzar múltiples particiones de datos, ser completado por múltiples transactores y persistir en una gran cantidad de nodos de almacenamiento de datos.

Esto presenta obvias complejidades de integridad de los datos. Sin embargo, la integridad de los datos está garantizada, ya que todos los componentes de la transacción están vinculados en una única envoltura de confirmación de dos fases. La totalidad de la transacción, en contra de todos los nodos y actores persistentes, completa o falla en su totalidad, y proporciona todas las mismas garantías versionadas.

El resultado final de esta confluencia de diseños arquitectónicos es que el sistema es completamente transaccional, seguro, altamente redundante y altamente escalable, tanto vertical como horizontalmente. Si bien las transacciones de escritura (que en la mayoría de los escenarios abarcan un pequeño porcentaje de la actividad) pueden estar limitadas por la necesidad transaccional de un único transactor por partición, la adición de la partición basada en reglas, especialmente de elementos de datos superiores, proporciona una enorme flexibilidad para expandir el sistema hasta un grado conceptualmente ilimitado, incluso antes de considerar bifurcar instancias.

La implementación de almacenamiento de datos de Tereon

La infraestructura de Tereon es capaz de manejar más de 1, 000,000 de transacciones garantizadas por ACID por segundo. Esto se logra abstrayendo o de otra manera implementando una capa de almacenamiento de datos 220 sobre una base de datos distribuida o bases de datos 224, usando una base de datos distribuida de clave/valor de alto rendimiento para el nivel de almacenamiento (esto puede ser a cualquier nivel de profundidad, desde una abstracción vía el Servicio de Datos de Tereon, a través del uso directo de la base de datos al nivel de almacenamiento, con canales de acceso de lectura y escritura independientes 226. El uso y configuración de Tereon de almacenamiento de datos es único.

La capa de servicios de datos se comunica con la capa de almacenamiento de datos a través de sus módulos de intercambio de datos personalizados. Las bases de datos en sí mismas no necesitan ofrecer ninguna garantía de ACID en absoluto, lo cual es manejado por la capa 220 de almacenamiento de datos. Tampoco necesitan ofrecer capacidades gráficas, ya que ralentizan significativamente los procesos de escritura. La capa 220 de almacenamiento de datos

proporciona la interfaz a las capas de datos heterogéneas y proporciona la funcionalidad de interfaz que requieren las diferentes partes del sistema. Por lo tanto, la funcionalidad de escritura proporciona una estructura rápida de celdas y columnas, mientras que la interfaz de lectura proporciona una interfaz gráfica que le permite atravesar el almacenamiento de datos distribuidos en microsegundos. La capa de almacenamiento de datos proporciona la interfaz SQL y la capa de la interfaz gráfica sobre las bases de datos de almacenamiento de datos centrales 224, y proporciona una serie de ventajas arquitectónicas significativas que distinguen a Tereon. Cada instancia de cliente (las instancias de Tereon Data Service 214) aloja un motor de base de datos en memoria/en proceso, incluidas las representaciones en caché de todos los datos importantes para esa instancia. En efecto, la instancia hospeda el motor de la base de datos y las representaciones en caché de todos los datos transaccionales actuales, el estado de cada transacción actual y toda otra información que se relaciona con el estado actual de esa instancia dentro de su parte de la RAM u otra memoria rápida de la máquina o máquinas en las que opera esa instancia.

Esto permite que el servicio de datos de Tereon facilite la mayoría de las tareas orientadas a la lectura a una tasa enorme (millones de consultas discretas por segundo, por ejemplo, donde los datos importantes y relevantes se almacenan en la caché localmente), magnitudes por encima de los niveles de rendimiento que se lograrían al serializar y realizar solicitudes externas o fuera de la máquina a sistemas de bases de datos externos. Cuando los datos no están en el caché en proceso, se recuperan del almacenamiento de valores clave.

Un sistema de control de versiones de MVCC se usa para administrar la concurrencia, y un atributo de la capa de datos es que los datos nunca se eliminan (fuera de las eliminaciones forzadas para el cumplimiento de la normativa) -el sistema conserva el historial completo de cada cambio de registro durante la vida del sistema de datos. Esto hace que las operaciones como las consultas "a partir de" sean triviales y hace la auditoría de cualquier cambio de plataforma.

La implementación de escritura de la capa de datos utiliza un único transactor compartido a través del cual deben fluir todos los cambios de datos, procesados en una serie, secuencia

rápida. Esto garantiza que las transacciones sean válidas, coherentes y minimiza la sobrecarga de concurrencia de cambios, que es un peso oneroso en la mayoría de las plataformas de bases de datos. El diseño del transactor utiliza un modelo de redundancia de respaldo en línea. A medida que el transactor procesa los cambios, notifica a todos los motores de consulta activos (que en este caso existen en el servicio de datos de Tereon) y actualizan sus cachés en memoria según corresponda.

El diseño proporciona una latencia de microsegundos para lecturas, escrituras y búsquedas, independientemente del tamaño de almacenamiento de datos. También proporciona una construcción modular que permite actualizar y reemplazar componentes sin afectar su funcionamiento. Este almacenamiento de datos se abstrae de la implementación subyacente y se puede sustituir por otras tiendas en el servicio de datos de Tereon.

Si la capa de almacenamiento de datos está configurada para operar con garantías pesimistas de ACID 226, es decir, dar un paso adicional para confirmar que ha escrito un registro antes de pasar a la siguiente transacción, esto agrega un breve retraso, pero proporciona una garantía absoluta de consistencia ACID e integridad de datos.

La ventaja de este diseño es que proporciona garantías ACID, ya que la capa de aplicación no puede continuar hasta que la capa de datos confirma que ha escrito el registro y completado la transacción.

Esto significa que, por ejemplo, en banca, pagos y otros tipos de transacciones que deben preservar la causalidad, se eliminan los problemas causados por la eventualidad de la coherencia. Al diseñar en las garantías de ACID, también se elimina cualquier necesidad de cuentas de conciliación para cubrir cualquier déficit cuando los sistemas bancarios descubren procesos no coincidentes. El procesamiento en tiempo real significa que también se elimina la demora en el tiempo en que incurren los procesos de reconciliación en sistemas de consistencia eventual.

El diseño de esta plataforma ofrece niveles muy altos de redundancia y confiabilidad en el hardware básico, y una escalabilidad extraordinaria (tanto vertical como horizontalmente). Las preocupaciones teóricas sobre los posibles límites del sistema transactor llevaron a construir una plataforma de partición en el servicio de datos para superar esos límites, pero en la gran mayoría de los escenarios nunca sería necesario usar esa plataforma.

Servicio de Búsqueda/Directorio

El sistema Tereon tiene un servicio de directorio 216 que es un directorio de las credenciales e información en el sistema que identifica en qué servidor está registrado un usuario o dispositivo 218, o qué servidor ofrece una función, recurso, instalación en particular, tipo de transacción, u otro tipo de servicio. El servicio de directorio permite que se realicen múltiples métodos de autenticación de un usuario 218, ya que almacena una serie de diferentes tipos de credenciales relacionadas con ese usuario en particular. Por ejemplo, un usuario 218 puede autenticarse utilizando su número de teléfono móvil, dirección de correo electrónico, ubicación geográfica, PAN (números de cuenta principales), etc., y almacena en caché esos datos para que no sea necesario autenticarse cada vez.

El servicio de directorio 216 proporciona una capa de abstracción que separa el ID de autenticación del usuario de los servicios subyacentes, los servidores y las cuentas de usuario reales. Esto proporciona una abstracción entre las credenciales que un usuario 218 o comerciante puede usar para acceder a un servicio y la información que Tereon requiere para realizar el servicio en sí. Por ejemplo, en un servicio de pagos, el servicio de directorio 216 simplemente vincularía un ID de autenticación, como un número de móvil, y quizás un código de moneda con una dirección de servidor. No hay absolutamente ninguna manera de determinar si el usuario 218 tiene una cuenta bancaria, o con qué banco opera ese usuario.

La arquitectura del sistema permite a Tereon proporcionar varios servicios o características novedosas que están simplemente más allá del alcance de los sistemas existentes.

La arquitectura del sistema Tereon es útil porque permite sistemas escalables y redundantes. Los sistemas centrales del banco tienden a ofrecer módulos dedicados a canales individuales, por ejemplo, administración de tarjetas, comercio electrónico, pagos móviles. Esto refuerza los silos y aumenta la complejidad de sus sistemas de TI. Esa complejidad es una de las razones por las cuales los bancos no actualizan sus servicios y sistemas con regularidad.

El sistema Tereon está diseñado para soportar todos los dispositivos y todos los casos de uso con una arquitectura modular que lo hace altamente configurable y personalizable. El corazón de esto es el SDASF 104 discutido anteriormente, y el motor de reglas de negocios 106, junto con un alto nivel de abstracción. Es esto junto con el marco extensible lo que permite la flexibilidad de Tereon.

Tereon permite a un operador utilizar sistemas estándar de nivel de operador para proporcionar y soportar numerosos tipos de transacciones. Tereon admitirá cualquier transacción, ya sea que esa transacción requiera o no autenticación.

Procesos especiales

Los procesos especiales 208 aprovechan idealmente la funcionalidad de los servicios de datos. Sin embargo, puede haber casos en los que un requisito único no justifique cambiar o ampliar el servicio de datos centrales, de modo que la biblioteca de datos se aproveche dentro del proceso especial para extraer los datos directamente. Esto puede, por ejemplo, incluir procesos de funcionalidad gráfica como las funciones de AML (lavado de dinero), CRM (gestión de relaciones con el cliente) o ERP (planificación de recursos empresariales).

Servicios múltiples

Como cada servicio es un módulo, la estructura modular de Tereon le permite admitir múltiples tipos de servicios y dispositivos. En los pagos, por ejemplo, esta estructura le permite a Tereon respaldar una pluralidad de dispositivos y tipos de pago, incluidos bancos, tarjetas de crédito, servicios de crédito, cooperativas de crédito, servicios de débito, planes de empleados, billetera

electrónica (ePurse), planes de fidelidad, planes de membresía, microfinanzas, prepago, servicios para estudiantes, emisión de boletos, notificaciones por SMS, búsquedas de HLR, etc.

Dispositivos de punto final múltiple

La estructura modular de Tereon le permite admitir casi cualquier dispositivo de punto final con el que pueda comunicarse, ya sea directa o indirectamente, incluidas tarjetas de banda magnética, tarjetas inteligentes, teléfonos con funciones, teléfonos inteligentes, tabletas, terminales de tarjetas, terminales de puntos de venta, cajeros automáticos, PC, pantallas de visualización, controles de acceso electrónico, portales de comercio electrónico, dispositivos de uso en la muñeca y otros dispositivos portátiles, etc.

Múltiples bases de datos

La arquitectura modular tiene otro beneficio, ya que el sistema no se limita a una base de datos. En su lugar, se pueden conectar varias bases de datos, cada una con un módulo específico a la base de datos en cuestión, y así usar bases de datos específicas para propósitos específicos, o usar una combinación de registros de datos en múltiples bases de datos heterogéneas.

La implementación de un subsistema de licencias 210 es novedosa en el uso de las autoridades de certificación para fines de licencias, además de los beneficios de autorización y autenticación que proporciona. En lugar de que cada módulo confíe en las afirmaciones de los demás, use una autenticación simple con una base de datos compartida o delegue infinitamente en un servidor de licencias independiente en cada acumulación de conexión (con la sobrecarga de rendimiento y confiabilidad que conlleva), los patrones de implementación más comunes son los sistemas distribuidos, basados en módulos. En Tereon, el subsistema de licencias garantiza que las conexiones entre los módulos sean intrínsecamente seguras y tengan metadatos validados y confiables sobre los actores, con un rendimiento mínimo y una sobrecarga de confiabilidad.

La implementación también limita el alcance de la vulnerabilidad potencial en la instancia de un compromiso del servidor de licencias - en una implementación tradicional, tal compromiso merecería una reconstrucción tipo “Scorched Earth” (nombre del famoso videojuego) de todos los componentes. En el modelo de Tereon, hay una exposición basada en el tiempo que exigiría un nuevo certificado de firma intermedio (si no estuviera protegido por un módulo de seguridad de hardware). Todos los certificados existentes, otorgados antes del compromiso, serían eximidos y podrían renovarse en el horario normal. Los nuevos certificados serían otorgados bajo la nueva autoridad, y cualquier otro certificado deshonesto sería rechazado como un compromiso posterior. Este control de la ventana de exposición beneficia a los peores escenarios. Los datos almacenados por el servidor de licencias son, fuera de la clave privada del certificado de firma que, idealmente, se guarda en un módulo de seguridad de hardware, información completamente sin privilegios.

El diseño de Tereon también lleva a la opción de combinar un dispositivo de punto final, como un teléfono móvil o un dispositivo de baja frecuencia con un servidor Tereon miniaturizado que se comunicará con otros servidores de Tereon como parte de una red de dichos servidores. Todavía se comunicarán con un servidor de licencias de Tereon 210, y quizás uno o más servidores de Tereon ejecutados por operadores para recopilar datos y coordinar actividades. Sin embargo, la distinción entre un dispositivo de punto final y un servidor Tereon puede ser abstracta, ya que cualquier distinción depende solo de los casos de uso a los que se asignan los dispositivos y servidores.

Cadenas de resumen

Uno de los grandes inconvenientes de las cadenas de bloques (más conocidas como “blockchains”) es que la cadena de bloques almacena una auditoría de todas las transacciones anteriores (es decir, es posible determinar el historial de transacciones en la cadena de bloques, que luego se utiliza con fines de autenticación). Esto significa que el enfoque de la cadena de bloques no es infinitamente escalable, ya que el tamaño de la cadena de bloques eventualmente se vuelve demasiado grande para administrar en un marco de tiempo realista,

mientras que el tamaño de cada bloque limita el número de transacciones por segundo máximas que la cadena de bloques puede registrar.

Un segundo inconveniente es que el historial de transacciones está disponible para cualquier persona que pueda acceder a la cadena de bloques, y por lo tanto les brinda la capacidad de determinar quiénes son las partes en una transacción. Esto presenta importantes desafíos de privacidad y normativos para usar tecnología “blockchain” en cualquier actividad significativa en la que la privacidad y/o la confidencialidad sean requisitos primordiales.

Otro inconveniente es que la cadena de bloques solo puede contener el resultado o el registro final de una transacción y no puede validar los procesos reales o los pasos de la transacción en sí.

La cadena de resumen descrita en este documento busca superar estos problemas mediante el uso de un enfoque de función de resumen específico para mantener los registros privados entre las partes que realizan las transacciones, y aun así proporcionar una red de autenticación distribuida que abarque a todos los usuarios de Tereon, independientemente de si operan en redes abiertas o redes privadas.

Esto se logra mediante la construcción continua de una cadena distribuida de resumen que opera en tiempo real a través de redes tanto públicas como privadas, sin revelar a ningún tercero el contenido de las comunicaciones subyacentes. Esto contrasta directamente con el modelo estándar de un resumen o un libro mayor distribuido, donde cada parte tiene que ver y aceptar el contenido de cada comunicación, independientemente de si es o no parte de esa comunicación.

Cuando la cadena de resumen utiliza un protocolo que incluye una prueba de conocimiento cero, puede autenticar cada uno de los pasos de una transacción y la información o el resultado generado por esos pasos.

La implementación puede hacer ya sea que las partes en una comunicación generen el mismo resumen intermedio, o pueden generar resúmenes intermedios únicos para la misma comunicación. La estructura también permite que las partes migren a nuevos algoritmos de resumen, ya que los algoritmos existentes están en desuso, sin afectar la integridad de la cadena de resumen. Esto contrasta directamente con la dificultad de actualizar o poner al día los algoritmos utilizados en las soluciones en vivo existentes, tal como la tecnología “blockchain”.

Tereon crea una cadena de auditoría de resumen para cada lado (cuenta) de una transacción, donde:

- Tereon genera un resumen asociado con un registro y almacena el resumen contra ese registro. Tereon generará ese resumen tan pronto como se complete la acción que genera el registro, ya que utiliza los pasos que generan el registro y la información o el resultado que surge de esos pasos;
- Tereon usa el resumen para el registro anterior como parte de los datos para el registro actual; y
- El primer resumen en cualquier cadena de registros será un resumen aleatorio con la firma del servidor, la fecha y la hora en que Tereon genera ese resumen y, si es necesario, un número aleatorio.

Si el registro es de una acción que involucra a dos o más partes y cada parte debe tener un registro de su lado de la acción, entonces para cada una de las partes en una acción, Tereon hará lo siguiente:

- compartirá el resumen del registro de cada parte con la otra parte o partes;
- usará ese resumen para formar parte del registro de la parte receptora para el cual Tereon creará el resumen de registro;
- creará un resumen intermedio del registro que incluya el resumen de la otra parte o partes.
- compartirá ese resumen intermedio con la otra parte o partes, de modo que cada parte tenga un resumen que encapsule la parte de la otra parte en la acción (si las partes usan

el protocolo correcto, entonces no hay necesidad de compartir sus resúmenes intermedios ya que estos son exactamente el mismo);

- incluirá ese resumen intermedio en el registro de la acción;
- generará un resumen final que almacenará contra la acción y usará como parte del siguiente registro; y
- asociará cada uno de los resúmenes transferidos, o el resumen intermedio generado con el protocolo utilizando la prueba de conocimiento cero, con la identificación del cedente o el número de Tereon.

Tereon puede proporcionar las garantías ACID y la transacción de sesión en tiempo real y la velocidad de procesamiento que se requieren para esto, como se explicará a continuación. Además, la prevalencia de la cadena de bloques ha hecho que los desarrollos en esta área no hayan sido considerados.

La cadena de bloques solo puede copiar un registro de una transacción una vez que se ha completado la transacción. No hay garantía de que el registro pasado a la cadena de bloques sea realmente el registro genuino de la transacción en sí. De esta manera, la cadena de bloques está limitada, ya que su estructura de resumen subyacente está diseñada para recopilaciones estáticas de datos, no transacciones dinámicas en tiempo real, y se basa en la mayoría de los operadores que actúan honestamente. La tecnología de cadena de bloques (blockchain) en sí misma presenta una limitación adicional, ya que solo puede ofrecer consistencia eventual; no la consistencia de ACID determinada por el orden cronológico de las transacciones, sino por el orden en que esas transacciones se incorporan a los bloques, y por el modelo de consenso para administrar las bifurcaciones en la cadena de bloques cuando se descubren dos o más bloques que contienen conjuntos de transacciones ligeramente diferentes. o menos simultáneamente.

La Fig. 5 ilustra la naturaleza dendrítica de una cadena de resumen que involucra cuatro cuentas 502, 504, 506 y 508. Las cuentas pueden estar en el mismo servidor, o pueden estar en servidores separados. Cada sistema puede admitir uno o más servidores, y cada servidor puede admitir una o más cuentas. Donde residen las cuentas es irrelevante. La Fig. 5 también ilustra cinco transacciones que ocurren entre pares de cuentas. Hay dos transacciones que ocurren

entre las cuentas 502 y 504, dos transacciones que ocurren entre las cuentas 502 y 506, y una transacción que ocurre entre las cuentas 506 y 508. En la figura, cada cuadro es un paso que se relaciona con la cuenta en la parte superior de la columna. Cada paso implica una acción o transacción invisible, como una búsqueda dentro de esa cuenta, o una transacción entre esa cuenta y otra cuenta invisible o un sistema. Esas transacciones o acciones son irrelevantes. Todo lo que importa es que involucran algo que un sistema de Tereon registra en su auditoría.

En la etapa 510, el sistema Tereon toma $h(502)$, el resumen anterior para esta cuenta. Como se mencionó anteriormente, el primer resumen es un resumen aleatorio con la firma del servidor, la fecha y la hora en que Tereon genera ese resumen y, si es necesario, un número aleatorio. Tereon agrega este resumen al registro de la transacción o acción que ocurre en el paso 510, y luego lo usa como el inicio para calcular $h(512)$, el resumen para esta transacción. El registro en esta etapa contiene $h(502)$ y $h(512)$.

En la etapa 512, el sistema intercambia el resumen, $h(510)$, con el servidor que tiene la cuenta 504. Este agrega el resumen $h(504)$, el resumen para esta transacción para la cuenta 504, al registro, genera un resumen intermedio $h(512)$, agrega esto a su registro y luego lo intercambia por el resumen intermedio $h(514)$.) desde la cuenta 504 (generada en el paso 514, como se explica a continuación). Luego agrega este resumen a su registro y genera el resumen $h(512)$.

Este resumen $h(512)$ ahora contiene información que valida la cadena de resúmenes para la cuenta 502 hasta el paso 512, y para la cuenta 504 hasta la etapa intermedia del paso 514. El registro contiene $h(510)$, $h(512j)$, $h(514j)$, $h(504)$ y $h(512)$.

En la etapa 514, el sistema intercambia el resumen, $h(504)$, con el servidor que tiene la cuenta 502. Agrega el resumen $h(510)$ de la cuenta 502 al registro, genera un resumen intermedio $h(514)$, agrega esto a su registro y luego lo cambia por el resumen intermedio $h(512)$ de la cuenta 502. Luego agrega este resumen a su registro y genera el resumen $h(514)$.

Esta cadena ahora contiene información que valida la cadena de resúmenes en la cuenta 502 hasta el paso 512 y para la cuenta 504 hasta el paso 514.

Este procedimiento se lleva a cabo para transacciones adicionales entre las cuentas 502, 504, 506 y 508 con el fin de generar resúmenes para cada transacción exactamente de la misma manera que se establece anteriormente. Por ejemplo, en el paso 534, el sistema toma h(528), el resumen anterior para la cuenta 502 generado en el paso 528, agrega esto al registro de la transacción o acción (invisible) que conduce a un registro de auditoría, y genera h(534), el resumen para esta transacción. Esta cadena ahora contiene información que valida la cadena de resúmenes en la cuenta 502 hasta el paso 534, para la cuenta 504 hasta el paso 526, para la cuenta 506 hasta el paso 530, y para la cuenta 508 hasta el resumen intermedio desde la cuenta 508 que se utilizó para generar h(530) en el paso 530. El registro contiene los resúmenes h(534) y h(528). Tereon genera el resumen h(528) en el paso 528 a partir de un registro que incluyó h(530j), que a su vez se generó a partir de h(524) en el paso 530. El resumen h(524) contiene información que validó la cuenta 508 hasta el resumen intermedio de la cuenta 508 que se utilizó para generar h(524) en el paso 524.

Reconciliación

Con el fin de garantizar que una transacción no pueda tener lugar si un estafador ha alterado el registro de la transacción anterior, la reconciliación se puede realizar primero en las últimas transacciones "N". Así, por ejemplo, antes de que Tereon realice la transacción representada por el paso 522, primero puede volver a calcular los resúmenes para el paso 516, y quizás el paso 512 y así sucesivamente, hasta las transacciones 'N' anteriores para la cuenta 502. La pista de auditoría tendrá información suficiente para recalcular los valores de resúmenes finales para las transacciones. Del mismo modo, el sistema que posee la cuenta 504 puede recalcular los resúmenes para el paso 526, el paso 520, etc. Tereon no necesita recalcular ninguno de los resúmenes para la cuenta 506 para la transacción del paso 522.

En una cadena de resumen, si alguno de los resúmenes registrados no coincide con los resúmenes recalculados, esto significa que un registro se ha modificado sin autorización, y el operador puede investigar el problema de inmediato o bloquear otras transacciones.

Cadena de resumen del sistema

También se puede agregar un resumen del sistema a cada registro. Este será un resumen del registro donde el valor inicial será el resumen de la acción anterior en el sistema, independientemente de si esa acción se relaciona o no con la cuenta a la que pertenece el registro que se está copiando. Si se agrega el resumen del sistema, se proporciona una cadena de resumen dentro de cada cuenta y una cadena de resumen del sistema en su conjunto.

La Fig. 6 ilustra la naturaleza dendrítica de una cadena de resumen que involucra dos cuentas 602 y 604 en el mismo sistema, cuya "cuenta del sistema" que registra todos los eventos del sistema es 606. El sistema crea un nuevo resumen de un registro para cada acción que resulta en un registro, independientemente de donde resida ese registro. Estos son los resúmenes del sistema, h(606), h(608), h(612), etc.

Las funciones administrativas también dan como resultado registros que el sistema asigna a las cuentas administrativas, independientemente de si esas funciones implican aportes humanos o si son funciones automatizadas.

En el paso 608, Tereon crea un resumen del registro de una acción o transacción invisible en la cuenta 602 que activa una entrada en el registro de auditoría del sistema (el registro para la cuenta 602 incluye el resumen h(602), el resumen del registro anterior para esa cuenta), y usa h(606) para ese nuevo resumen del sistema h(608). El sistema luego registra este resumen contra el registro de la transacción y calcula el resumen h(610) para la cuenta 602 en el paso 610.

Si el rendimiento informático del sistema lo permite, entonces puede usar una variación más fuerte para los resúmenes del sistema que refleja el funcionamiento de los resúmenes de cuentas.

En el paso 610, Tereon intercambia el resumen, h(602), con la cuenta del sistema 606 para el resumen h(606). Agrega el resumen h(606) de la cuenta del sistema 606 a su registro y genera

un resumen intermedio h(610i). Esto se genera después de que haya completado la acción o transacción invisible en la cuenta 602 que activa una entrada en el registro de auditoría del sistema y agrega el resumen a su registro. Tereon luego intercambia este resumen intermedio para el resumen del sistema intermedio h(608i). Luego agrega esto y h(608) a su registro y genera un nuevo resumen de cuenta h(610).

En el paso 612, Tereon intercambia el resumen, h(608) generado en el paso 608, con las cuentas 602 y 604. Este agrega h(610) generada en el paso 610 y h(604) a su registro y genera un resumen intermedio h(612,). Intercambia esto con las cuentas 602 y 604 por sus resúmenes del sistema de cuentas intermedias h(614Sj) y h(616Sj), y los resúmenes intermedios h(614,) correspondientes a las cuentas 602 y h(616i) correspondientes a la cuenta 604. Luego genera un nuevo resumen del sistema h(612). El sistema entonces registra este resumen.

En el paso 614, Tereon intercambia el resumen, h(610) generado en el paso 610, con la cuenta del sistema 606. Este agrega el resumen h(608) de la cuenta del sistema 606, generado en el paso 608, a su registro, genera un resumen del sistema de cuenta intermedio (614Ssi). Genera este resumen una vez que ha completado la transacción con la cuenta 604 (e intercambia los resúmenes de transacción intermedios h(614i) y h(616i)), lo agrega a su registro y luego lo intercambia por el resumen del sistema intermedio h(612i). Luego agrega esto y h(608) a su registro y genera el resumen de cuenta h(614).

En el paso 616, Tereon intercambia el resumen, h(604), con la cuenta del sistema 606. Agrega el resumen h(608) de la cuenta del sistema a su registro, genera un resumen del sistema de cuenta intermedio (616Si). Esto se genera después de que se haya completado la transacción con la cuenta 602 (y se intercambian los resúmenes de transacción intermedios h(614i) y h(616i), se agrega el resumen a su registro y luego se intercambia por el resumen del sistema intermedio h(612i). Luego agrega esto y h(608) a su registro y genera el resumen de cuenta h(616).

En la etapa 612, una opción es que el sistema envíe el resumen h(614Si) del sistema intermedio a la cuenta 604, y el resumen h(616si) del sistema intermedio a la cuenta 602. Esto significaría

que los resúmenes de registros finales para esas cuentas, h(614) y h(616) contendrían registros de los tres resúmenes del sistema intermedio, h(614Si), h(614si) y h(612) y así se proporciona una capa extra de certeza.

El sistema cadena de resumen ahora contiene resúmenes de ambos lados de cada transacción individual, así como las transacciones en su conjunto, lo que fortalece considerablemente la cadena de resumen.

Si Tereon administra una transacción entre cuentas en un sistema diferente, entonces el proceso es como en los pasos 608 y 610 en cada uno de esos sistemas.

Resúmenes de licencias de servidor

Los resúmenes anteriores se relacionan con aquellos generados en y entre sistemas Tereon separados. A medida que estos sistemas interactúan entre sí, eventualmente se unirán al árbol de resumen que contiene información que validará las transacciones en todos esos sistemas. Sin embargo, esto solo crecerá al ritmo en que estos sistemas interactúen entre sí. El sistema puede ir aún más lejos y construir otra capa que garantice que cada servidor se unirá de inmediato al árbol de resúmenes global. Esto distingue la cadena de resumen completamente de la cadena de bloques.

Cuando un operador de cadenas de bloques (blockchain) configura una cadena de bloque privada, esa cadena de bloque opera aislada de todos los demás. Lo que gana en la velocidad de procesamiento general se pierde en cualquier seguridad que de otro modo podría ofrecer, ya que un usuario no puede confiar en una gran red de cadenas de bloques para validar una transacción. Una de las afirmaciones de seguridad de la cadena de bloques es que un atacante tendría que comprometer varios nodos de la red de la cadena de bloques para comprometer su seguridad (comprometer entre el 25 y el 33% de los nodos podría ser suficiente para comprometer la cadena de bloques). Una sola cadena de bloques privada reduce ese número, por definición, a uno.

Con la cadena de resumen, incluso un servidor o una red privados de Tereon pueden beneficiarse de las cadenas de resumen generadas por los servidores y redes públicos de Tereon. Operar un servidor o una red privada de Tereon no significa que el operador deba comprometer la fuerza de autenticación del sistema Tereon porque ese sistema seguirá siendo miembro de la cadena de resumen global. Es simplemente que sus transacciones, distintas de las relacionadas con el servidor de licencias, permanecerán completamente privadas para ese sistema.

Para lograr esto, cada servidor debe interactuar con el servidor de licencias, independientemente de si interactúa o no con otros servidores Tereon. Si un servidor de Tereon opera en un sistema de circuito cerrado, solo interactuará con otros servidores de Tereon dentro de ese circuito, y solo si ese circuito comprende más de un servidor.

Al agregar un resumen de servidor de licencias, cada servidor se unirá a la cadena de resumen del servidor global tan pronto como interactúe con el servidor de licencias, lo que debe hacerse diariamente. Los resúmenes de servidor de licencias son esencialmente generados por una transacción de dos partes entre un servidor de Tereon y el servidor de licencias. Las transacciones del servidor de licencias no afectan las transacciones de datos subyacentes entre los servidores de Tereon, excepto el hecho de que los resúmenes del sistema para cada servidor ahora también contendrán información derivada de los resúmenes del servidor de licencias, y viceversa.

La Fig. 7 ilustra la naturaleza dendrítica de los resúmenes de licencia. En este ejemplo simple, el servidor del sistema 702 es un sistema de bucle cerrado, con el que se interconectarán los servidores 704 y 706 de los sistemas. Los tres deben interactuar con el servidor de licencias 708 de forma periódica.

En su primera consulta con el servidor de licencias 708, cada servidor genera su primer resumen a partir de su clave pública, la fecha y la hora en que el servidor obtuvo la primera licencia, y un conjunto de datos aleatorio.

En el paso 710, Tereon usa su resumen, h(708), para generar un resumen de licencia intermedia h(710,)), agrega esto a su registro y lo intercambia por el resumen del sistema intermedio h(712) del servidor 702. Luego agrega este resumen a su registro y luego genera el resumen de licencia h(710) que agrega a su registro.

En el paso o etapa 712, Tereon usa su resumen, h(702), para generar un resumen del sistema intermedio h(712,)), agrega esto a su registro y lo intercambia por el resumen de la licencia intermedia h(710,) del servidor de licencias 708 . Luego agrega este resumen a su registro y genera el resumen del sistema h (712), que agrega a su registro.

En el paso 714, Tereon usa su resumen, h(710) generado en el paso 710, para generar un resumen de licencia intermedia h(714), lo agrega a su registro y lo intercambia por el resumen del sistema intermedio h(716) de servidor 704. Luego agrega este resumen a su registro y genera el resumen de licencia h(714), que agrega a su registro.

En la etapa 716, Tereon usa su resumen, h (704), para generar un resumen del sistema intermedio h(716,)), agrega esto a su registro y lo intercambia por el resumen de la licencia intermedia h(714,) del servidor de licencias 708 . Luego agrega este resumen a su registro y genera el resumen del sistema h(716), que agrega a su registro.

En el paso 718, Tereon genera un resumen de licencia intermedio h(718,)), lo agrega a su registro y lo intercambia por el resumen del sistema intermedio h(720j) desde el servidor 706. Luego agrega este resumen a su registro y genera el resumen de licencia h(718), que agrega a su registro.

En la etapa 720, Tereon usa su resumen, h(706), para generar un resumen del sistema intermedio h(720j)), agrega esto a su registro y lo intercambia por el resumen de licencia intermedia h(718,) del servidor de licencias 708. Luego agrega este resumen a su registro y genera el resumen del sistema h (720), que agrega a su registro.

Estas tres transacciones del servidor de licencias al servidor Tereon han dado como resultado los siguientes resultados:

- El resumen $h(712)$ generado en el paso 712 contiene información que valida el estado de:
 - la cadena de resumen del servidor de licencias 708 hasta el resumen intermedio $h(710i)$
 - la cadena de resumen del servidor 702's hasta el resumen $h(712)$.
- El resumen $h(716)$ generado en el paso 716 contiene información que valida el estado de:
 - la cadena de resumen del servidor de licencias 708's hasta el resumen intermedio $h(714i)$;
 - la cadena de resumen del servidor 702's hasta el resumen intermedio $h(k_{702i})$; y
 - la cadena de resumen del servidor 704's hasta el resumen $h(716)$.
- El resumen $h(720)$ generado en el paso 720 contiene información que valida el estado de:
 - la cadena de resumen del servidor de licencias 708's hasta el resumen intermedio $h(718i)$;
 - la cadena de resumen del servidor 702's hasta el resumen intermedio $h(k_{702i})$;
 - la cadena de resumen del servidor 704's hasta el resumen intermedio $h(716i)$; y
 - la cadena de resumen del servidor 706's hasta el resumen $h(720)$.
- El resumen $h(718)$ generado en el paso 718 contiene información que valida el estado de:
 - la cadena de resumen del servidor de licencias 708's hasta el resumen $h(718)$;
 - cadena de resumen del servidor 702's hasta el resumen intermedio $h(k_{702i})$;
 - la cadena de resumen del servidor 704's hasta el resumen $h(k_{704})$; y
 - la cadena de resumen del servidor 706's hasta el resumen $h(720)$.

La licencia y los resúmenes del sistema, por lo tanto, contienen información que les permite verificar las transacciones en cada servidor de la red, independientemente de si esos servidores están interconectados o no funcionan como un ciclo cerrado.

Tereon puede implementar una capa similar con el servicio de directorio de búsqueda que operará de manera similar a la cadena de resumen creada por el servicio de licencias.

Transacciones fuera de línea

Usando este enfoque, las transacciones fuera de línea ahora pueden tener la misma validez que las transacciones en línea, ya que se elimina la necesidad de tener enlaces de comunicaciones constantes entre los dispositivos y sus servidores. Por lo tanto, los dispositivos, como sensores, terminales de pago portátiles, etc., podrían comunicarse entre sí y luego conectarse con sus servidores a intervalos predeterminados para descargar y cargar datos. El sistema funcionaría sin problemas entre entornos conectados y no conectados.

La cadena de resumen permite a los dispositivos validar y auditar las transacciones entre ellos mientras no pueden comunicarse con sus respectivos servidores, usando reglas comerciales para determinar si pueden participar en las transacciones fuera de línea o no. Los dispositivos simplemente reconciliarán esos registros de auditoría y transacción con sus servidores cuando se conecten a esos servidores una vez más.

La Fig. 8 ilustra un ejemplo de una cadena de resumen que involucra cuatro dispositivos que se desconectan de sus respectivos servidores Tereon por un tiempo. Tres de estos, 802, 804 y 806, son visibles (el cuarto, el dispositivo 808, interactúa con la cadena en el paso 828).

Para admitir transacciones fuera de línea entre dispositivos, los dispositivos en sí mismos generarán un resumen de cada transacción en la que participan. Cuando el dispositivo vuelva a estar en línea y se comunique con su servidor, ese dispositivo enviará el resumen para esa transacción a su servidor.

Cuando el dispositivo que inicia una transacción está fuera de línea, generará un resumen para su transacción y almacenará ese resumen. También enviará ese resumen a su dispositivo de contraparte (el dispositivo con el que está realizando transacciones), y ese dispositivo de contraparte enviará su resumen al primer dispositivo. Esto se logra de la misma manera que las

cadenas de resumen descritas anteriormente. Los dispositivos pueden comunicarse entre sí a través de cualquier canal de dos vías, como Bluetooth, NFC, Wi-Fi local, etc. Incluso podrían publicar códigos de barras para cada etapa de transacción en sus pantallas para que las otras las lean. Cada dispositivo también enviará una copia firmada y encriptada de su registro de transacción al otro dispositivo, donde la firma también contendrá el servidor de destino para ese registro. Sólo el servidor de destino podrá descifrar ese registro.

Una vez que un dispositivo recupera las comunicaciones con su servidor Tereon, ese dispositivo enviará al servidor los registros cifrados de sus transacciones fuera de línea y sus resúmenes asociados. También enviará a ese servidor las copias de otras transacciones que tenga, como los registros de sus contrapartes, y el servidor transmitirá esos registros y sus resúmenes asociados a los servidores con los que están registrados esos dispositivos de contraparte. Cada dispositivo creará su propio número de transacción interno único (como el generado por un contador monotónico, por ejemplo) que identificará su parte en una transacción. Si la transacción está en línea, entonces el servidor al que está conectado el dispositivo también generará un número de transacción único que utilizarán tanto el dispositivo como el servidor.

Los dispositivos pueden combinar sus números de transacción internos únicos con sellos de fecha y hora, información sobre el sesgo del reloj de los dispositivos y otra información para preservar la causalidad de cada transacción. Cuando sus respectivos servidores reciban la información de la transacción, podrán reconstruir el orden de las transacciones y así preservar la causalidad de las transacciones en línea y fuera de línea para todos los dispositivos.

Volviendo a la Fig. 8, en el paso 812, el dispositivo 802 resume el registro de la transacción, que incluye el resumen $h(802)$, el resumen del registro anterior y el resumen $h(810)$ del servidor 810 para generar $h(812)$. Luego pasa este resumen al servidor 810, donde ese resumen forma parte del registro utilizado para calcular $h(814)$ en el paso 814. El dispositivo 802 está en línea en este punto, lo que significa que está conectado a su servidor Tereon 810. En el paso 814, Tereon usa $h(810)$, el resumen anterior para el servidor 810, agrega esto y $h(812)$ al registro, y luego calcula $h(814)$. El registro contiene $h(810)$, $h(812)$ y $h(814)$.

Si el operador ha configurado Tereon para incluir el resumen del sistema, lo agregará al registro antes de calcular el resumen $h(814)$, como se describe anteriormente. El registro contendría entonces $h(812)$, $h(810)$, el resumen del sistema intermedio si es relevante y $h(814)$.

En la etapa 816, el dispositivo 802 ahora está fuera de línea, ya que no puede conectarse al servidor 810. Este hace la transacción con el dispositivo 804, que también está fuera de línea desde su respectivo servidor Tereon. Los dispositivos 802 y 804 siguen el procedimiento de resumen descrito anteriormente para generar un resumen intermedio $h(816)$ del dispositivo 802, un resumen intermedio $h(818)$ del dispositivo 804, el resumen $h(816)$ del dispositivo 802 y el resumen $h(818)$ desde el dispositivo 804 en el paso 818. Los dispositivos 802 y 804 ahora firman sus resúmenes con sus claves públicas fuera de línea y lo pasan al otro dispositivo, junto con una copia encriptada del registro para esa transacción. Esta es la primera transacción fuera de línea del dispositivo 802, ya que perdió contacto con el servidor 810 y la primera transacción fuera de línea del dispositivo 804, ya que perdió contacto con su servidor. El administrador puede configurar el sistema para que la aplicación transfiera hasta sus últimas n transacciones a cada dispositivo único con el que realice transacciones fuera de línea.

Este procedimiento se repite para las transacciones adicionales en la cadena entre el dispositivo 802 y el dispositivo 804 y entre el dispositivo 804 y el dispositivo 806. En estas transacciones, los dispositivos 802 y 804 no necesitan intercambiar su resumen y registrar las transacciones anteriores, ya que cada uno ya tiene una copia.

El dispositivo 802 continuará funcionando de esta manera hasta que restablezca el contacto con su servidor 810 en el paso 830. El dispositivo 802 ahora carga todos los registros cifrados de sus transacciones fuera de línea y sus resúmenes asociados, en este ejemplo $h(816)$, $h(822)$ y $h(826)$, generados en los pasos 816, 822 y 826, respectivamente. También carga los registros de transacciones cifrados y los resúmenes que posee para los dispositivos 804, 806 y 808. El servidor los almacena y los carga, respectivamente, a los servidores correspondientes a los dispositivos 804, 806 y 808. El servidor 810 registra esta carga como una transacción y genera el resumen $h(832)$ en el paso 832. El dispositivo 802 borra su registro de resúmenes de los

dispositivos 804, 806 y 808, y los registros de transacción respectivos, y genera el resumen h(830) en el paso 830.

El dispositivo 802 contiene el resumen y el registro cifrado de la transacción entre los dispositivos 806 y 808, lo que dio como resultado el resumen h(820) en los pasos 820 y h(808). En este ejemplo, h(808) se usa para referirse al resumen para el dispositivo 808 generado para esa transacción aquí ya que no se sabe cuántas transacciones sin conexión han ocurrido.

El servidor 810 reconciliará los registros fuera de línea que recibe del dispositivo 802, con los que recibe de los dispositivos 804, 806 y 808, y cualquier otro servidor que contenga esas transacciones. El servidor 810 sabrá de qué servidores recibirá los registros, ya que estos se corresponderán con los servidores a los que envió registros para las transacciones que involucraron al dispositivo 802. El dispositivo 802 no esperará recibir registros del dispositivo 808, ya que el dispositivo 802 no realizó transacciones con el dispositivo 808. Si los dispositivos 804 u 806 realizaron transacciones con dispositivos fuera de línea conectados a otros servidores, el servidor 810 puede recibir registros adicionales de esos otros servidores.

El servidor 810 utilizará las marcas de fecha y hora en los registros de transacciones y las firmas para ordenar y numerar esas transacciones, y marcarlas como transacciones fuera de línea.

El modo fuera de línea (off-line) presenta diversas variaciones. La primera es prescindir de los resúmenes intermedios fuera de línea y simplemente usarlos para la transacción anterior de cada dispositivo. Esto funcionará igual de bien, aunque pierde una capa de certeza. El segundo es generar resúmenes de dispositivo solo para transacciones fuera de línea. Esto simplifica ligeramente las transacciones en línea, pero nuevamente pierde una capa de certeza. La tercera variación no es firmar los registros para transacciones fuera de línea con una clave pública específica fuera de línea, sino simplemente firmar cada registro con la clave del dispositivo. Tanto el servidor como el dispositivo sabrán qué transacciones están en línea y cuáles están fuera de línea, ya que se registrarán en el registro de auditoría de la cuenta. Sin embargo, al ejecutar una clave separada y una serie de números de transacción para el dispositivo, resulta trivial mostrar las transacciones fuera de línea frente a las transacciones en línea.

Una cuarta variación es para cada servidor, cuando recibe registros de transacciones fuera de línea de sus dispositivos conectados, para notificar a todos los servidores a los que se aplican esos registros para esperar registros de esos servidores. Por ejemplo, en el diagrama fuera de línea que se muestra en la Fig. 8, se supone que el dispositivo 804 se conectó a su servidor más tarde, y que el dispositivo 806 realizó transacciones con otro dispositivo (no se muestra). Una vez que el dispositivo 804 se conecta con su servidor, ese servidor enviará los registros correspondientes al dispositivo 802 al servidor 810. El dispositivo 804 no realizó transacciones fuera de línea con ningún otro dispositivo y no tiene registros fuera de línea para ningún otro dispositivo. El servidor 810, por otro lado, envía sus registros para el dispositivo 804 al servidor correspondiente al dispositivo 804, y notifica a ese servidor que puede esperar recibir copias de los mismos registros del dispositivo 806 (el dispositivo 802 los pasó al dispositivo 806 durante la transacción en los pasos 826 y 828). Del mismo modo, una vez que el dispositivo 806 se conecta a su servidor, ese servidor enviará sus registros para el dispositivo 802 al servidor 810, para el dispositivo 804 al servidor correspondiente al dispositivo 804, para el dispositivo 808 al servidor correspondiente al dispositivo 808 y para el otro dispositivo a su respectivo servidor. También informará a los servidores correspondientes al dispositivo 802 (servidor 810) y al 804 para que esperen registros del servidor correspondiente al otro dispositivo.

El uso de una cadena de resumen no impone una carga cada vez mayor en Tereon. Una acción rara vez involucrará a más de dos partes, y donde lo hace, entonces esa acción generalmente será una transferencia de uno a muchos, que a su vez será una colección de transferencias uno a uno. Una transferencia de muchos a uno también suele ser una serie de transferencias de uno a uno, que es simplemente una recopilación de acciones de dos partes.

Modificación de registros

Si un usuario modifica un registro, Tereon no sobrescribirá el registro original. En su lugar, Tereon simplemente creará un nuevo registro que contenga el registro modificado, y esta será la versión a la que Tereon se refiere hasta que el registro se vuelva a modificar; la enmienda es una acción. Esto es lo que ocurrirá con todos los registros financieros y de transacciones, donde

el resultado de una transacción, como un pago, modifica efectivamente el resultado de la transacción anterior; también ocurrirá cuando un operador utilice un subconjunto de Tereon para administrar otros tipos de registros, como correos electrónicos, registros médicos, etc. Al hacer esto, Tereon conservará una copia de cada versión de un registro.

Puede haber circunstancias en las que un tribunal de justicia, o el funcionamiento general de la ley, requiera que un operador elimine un registro por completo, o que modifique el registro original. En estas circunstancias, Tereon eliminará o enmendará el contenido del registro original, y quizás el contenido de los registros asociados. Tereon puede lograr esto sin invalidar los resúmenes posteriores.

Si Tereon debe eliminar o enmendar un registro histórico, entonces hará lo siguiente:

- regenerar el resumen de ese registro para confirmar que el registro no se ha modificado o cambiado antes de que Tereon haya eliminado o enmendado el registro, y registre ese resumen regenerado
- registrar en un nuevo campo en el registro original que el contenido del registro que se eliminó o enmendó, y los motivos de la eliminación o enmienda
- eliminar o enmendar los campos relevantes en el registro, y agregar la fecha y hora de dicha eliminación o enmienda
- generar un nuevo resumen para ese registro; y
- grabar el nuevo resumen.

Al seguir este procedimiento, Tereon no tendrá que modificar la cadena de resumen de ninguna manera. Todos los resúmenes para registros válidos que se generaron a partir del resumen original del registro eliminado o enmendado seguirán siendo válidos. El resumen del sistema incluirá el nuevo resumen del registro eliminado o modificado, ya que esa eliminación o enmienda es una acción. De esta manera, la actividad fraudulenta se puede reconocer fácilmente encontrando cualquier resumen registrado que no coincida con los resúmenes recalculados.

Cadena de resumen con pruebas de conocimiento cero

La cadena de resumen proporciona una capa adicional que permite que ambos lados de una transacción demuestren a la otra que han resumido los registros a los que pertenecen los resúmenes. Esto se hace mediante la inclusión de un algoritmo de intercambio de claves dentro de la cadena de resumen, que permite a una parte demostrarle a una segunda parte (el verificador) que el resumen del registro es el verdadero resumen de ese registro.

Aquí se puede usar cualquier algoritmo que permita a dos partes negociar una clave común, y no hay necesidad de usar una prueba de conocimiento cero. Sin embargo, los algoritmos PAKE (intercambio de clave autenticado por contraseña) que utilizan pruebas de conocimiento cero son los más eficientes de usar aquí. El uso del protocolo PAKE correcto y la prueba de conocimiento cero en la etapa intermedia eliminan la necesidad de intercambiar resúmenes, ya que las partes generarán los mismos resúmenes intermedios.

Con un algoritmo tal como un algoritmo PAKE que permite a ambas partes generar el mismo resumen utilizando la prueba de conocimiento cero, las partes pueden ir más lejos. Al utilizar una prueba de conocimiento cero que puede incluir y utilizar la información que comprende la transacción para generar la prueba, ambas partes pueden generar un resumen intermedio idéntico. Esto elimina la necesidad de intercambiar sus resúmenes intermedios entre sí. También significa que los pasos que generan el registro y la información o el resultado que surgen de esos pasos se convierten en componentes del proceso de la cadena de resumen.

Si hay más de dos partes involucradas, entonces Tereon puede usar una variante de grupo de un protocolo y cero pruebas de conocimiento para permitir que todas las partes generen un resumen común.

Los algoritmos PAKE que permiten a las partes generar el mismo resumen normalmente tomarán dos o tres pases de información entre las partes antes de que puedan generar el resumen intermedio. Si una transacción solo requiere dos etapas para completarse (por ejemplo, una solicitud y una aceptación/verificación), entonces las partes solo generarán un resumen intermedio. Si una transacción requiere tres etapas y el algoritmo genera un resumen

en dos pasadas, las partes intercambiarán cuatro conjuntos de información, repitiendo la tercera etapa dos veces y generarán dos resúmenes, el primer resumen después de los primeros dos pasos de la transacción, luego el segundo resumen después de la repetición del tercer paso.

Un ejemplo de tal prueba de conocimiento cero es la Prueba Schnorr NIZK. Esta prueba de conocimiento cero puede extenderse simplemente agregando información adicional tanto a la información que se envía como parte de la prueba, como a la información que se utiliza para generar el resumen que forma parte de la prueba, como se muestra en el documento de especificaciones para la prueba de Schnorr NIZK.

También se puede utilizar otro método, tal como el de adaptar el método para generar la clave común en el protocolo SPEKE (intercambio de clave exponencial de contraseña simple), y la forma de hacerlo es trivial dado lo anterior.

También es un ejercicio trivial para extender los protocolos de intercambio de claves para permitir que las partes generen una clave común basada en los datos de la transacción. Nuevamente, estos no se ilustran aquí simplemente con el propósito de ser breves.

Para generar el resumen común, las partes simplemente generan un resumen de la clave común. El resumen contendrá información que puede validar la información de la transacción, porque esa información se usó en el proceso para generar la clave común y, por lo tanto, el resumen.

Transacción en dos etapas.

Un ejemplo que ilustra cómo funciona esto se hará con referencia nuevamente a la Fig. 5, que ilustra la naturaleza dendrítica de una cadena de resumen que involucra cuatro cuentas, 502, 504, 506 y 508. Las cuentas pueden estar en el mismo sistema, o pueden estar en sistemas separados. Donde residen las cuentas es irrelevante. Esta transacción en los pasos 512 y 514 toma dos etapas.

PAKE dos pasos

En la primera pasada en el paso 512, la cuenta 502 toma $h(510)$, el resumen anterior para esta cuenta generado en el paso 510, agrega esto a la primera etapa de información transaccional, construye la primera prueba de conocimiento de cero y pasa esto a la cuenta 504. La prueba de conocimiento de cero acompaña la información que constituye la primera etapa de la información transaccional y el resumen $h(510)$.

En la segunda pasada, la cuenta 504 toma $h(504)$, el resumen anterior para esa cuenta, agrega esto a la segunda etapa de información transaccional, construye la segunda prueba de conocimiento de cero y la pasa a la cuenta 502. La segunda prueba de conocimiento de cero acompaña la información que constituye la segunda etapa de la información transaccional y el resumen $h(504)$.

Las cuentas 502 y 504 ahora construyen independientemente el resumen $h(512,514)$, que es el resumen intermedio para ambas cuentas. Ambas cuentas 502 y 504 agregan este resumen a sus registros. La cuenta 502 genera el resumen $h(512)$ de su registro de la transacción en el paso 512, y la cuenta 504 genera el resumen $h(514)$ de su registro de la transacción en el paso 514.

PAKE tres pasos

En este ejemplo, la transacción en los pasos 512 y 514 toma dos etapas, con un algoritmo PAKE que permite a las partes construir un resumen común después de tres pasos o pasadas.

La primera pasada y la segunda pasada se realizan como se describió anteriormente. En una tercera pasada, la cuenta 502 toma la información que la cuenta 504 envió en la segunda pasada, construye la tercera prueba de conocimiento de cero con esa información y la envía a la cuenta 504. La tercera prueba de conocimiento de cero también acompaña la información que constituye la segunda etapa de la información transaccional y el resumen $h(504)$.

Las cuentas 502 y 504 ahora construyen independientemente el resumen $h(512i514i)$. Ambas cuentas 502 y 504 agregan este resumen a sus registros. La cuenta 502 genera el resumen $h(512)$ de su registro de la transacción en el paso 512 y la cuenta 504 genera el resumen $h(514)$ de su registro de la transacción en el paso 514, como en el enfoque PAKE de dos pasos.

En ambos casos, la cadena ahora contiene información que valida la cadena de resúmenes en la cuenta 502 hasta el paso 512 y para la cuenta 504 hasta el paso 514. Ambas cuentas 502 y 504 tienen el resumen intermedio $h(512i, 514i)$, así como sus resúmenes para sus registros. Sin embargo, el resumen intermedio aquí es sutilmente diferente al de los resúmenes intermedios que se intercambiaron entre los sistemas en los ejemplos anteriores que no usan pruebas de conocimiento cero. Aquí el resumen intermedio es el resumen de la transacción entre las cuentas 502 y 504, por lo que es común a las cuentas 502 y 504. El resumen es el resumen de la transacción y se generó como parte de la transacción. Es contemporáneo a la transacción. El resumen $h(512)$ es el resumen de la cuenta 502 de su registro de la transacción, que incluirá información que le es privada, mientras que el resumen de la cuenta 504, $h(514)$, es el resumen de su registro de la transacción. Así, las cuentas 502 y 504 pueden probar tanto los pasos reales en la transacción entre ellos como sus registros de esa transacción.

Transacción en tres etapas.

Como otro ejemplo usando la FIG. 5, se supondrá que la transacción en los pasos 528 y 530 implica tres etapas separadas en lugar de dos.

PAKE dos pasos

En la primera pasada, la cuenta 502 toma $h(522)$, el resumen anterior para esta cuenta generado en la etapa 522, agrega esto a la primera etapa de información transaccional, construye la primera prueba de conocimiento cero y la pasa a la cuenta 506. La prueba de conocimiento cero acompaña la información que constituye la primera etapa de la información transaccional y el resumen $h(522)$.

En la segunda pasada, la cuenta 506 toma $h(524)$, el resumen anterior para esa cuenta generado en la etapa 524, agrega esto a la segunda etapa de información transaccional, construye la segunda prueba de conocimiento cero y la pasa a la cuenta 502. La segunda prueba de conocimiento cero acompaña la información que constituye la segunda etapa de la información transaccional y el resumen $h(524)$.

Las cuentas 502 y 506 ahora pueden construir independientemente el resumen $h(528i, 530i)$, ya que el algoritmo PAKE permite a las partes construir un resumen común después de dos pasadas. Sin embargo, la transacción aún tiene una tercera etapa por realizar.

En este ejemplo, el sistema simplemente ejecuta un segundo conjunto de pases o pasadas con el algoritmo PAKE, comenzando con la tercera etapa de la transacción. El segundo paso de este segundo conjunto de pasos podría simplemente utilizar datos aleatorios. Alternativamente, podría repetir la última etapa, que es similar a usar un PAKE de tres pasos con una transacción de dos etapas.

En este último caso, se realiza un tercer paso (el primer paso del nuevo algoritmo PAKE), donde la cuenta 502 toma $h(528i, 530i)$, que ha firmado, agrega esto a la tercera etapa de información transaccional, construye el tercer conocimiento cero prueba con la información, y envía esto a la cuenta 506. Se realiza una cuarta pasada (la segunda pasada del nuevo algoritmo PAKE), donde la cuenta 506 toma $h(528i, 530i)$, que ha firmado, agrega esto a la tercera etapa de información transaccional que cuenta 502 enviada, construye la cuarta prueba de conocimiento cero con la información, y envía esto a la cuenta 502. Las cuentas 502 y 506 ahora pueden construir independientemente el resumen $h(528i2, 530i2)$. Este es el segundo resumen común generado en esta transacción, y ahora es el resumen de la transacción entre las cuentas 502 y 506, ya que contiene las tres etapas de la transacción. Ambas cuentas 502 y 506 agregan este resumen a sus registros. La cuenta 502 genera el resumen $h(528)$ de su registro de la transacción en el paso 528 y la cuenta 506 genera el resumen $h(530)$ de su registro de la transacción en el paso 530.

Este procedimiento se lleva a cabo para transacciones adicionales entre las cuentas 502, 504, 506 y 508 con el fin de generar resúmenes para cada transacción exactamente de la misma manera que se establece anteriormente.

PAKE tres pasos

La primera pasada y la segunda pasada se realizan como se describió anteriormente. En el tercer paso, la cuenta 502 construye la tercera prueba de conocimiento cero con la información que constituye la tercera etapa de la información transaccional, y la envía a la cuenta 506. La prueba de conocimiento cero acompaña la información que conforma la tercera etapa de la información transaccional.

Las cuentas 502 y 506 ahora construyen independientemente el resumen $h(528i, 530i)$. Ambas cuentas 502 y 506 agregan este resumen a sus registros. La cuenta 502 genera el resumen $h(528)$ de su registro de la transacción en el paso 528 y la cuenta 506 genera el resumen $h(530)$ de su registro de la transacción en el paso 530.

En los ejemplos anteriores relacionados con la Fig. 5, donde el sistema utiliza pruebas de conocimiento cero para generar los resúmenes intermedios o de transacción, el resumen $h(530)$ contiene información que valida todos los resúmenes de la cuenta 502 a $h(528i)$, todos los resúmenes de la cuenta 504 a $h(526i)$, todos de resumen de la cuenta 508 hasta el resumen intermedio o de transacción de la cuenta 508 que se creó cuando se creó la cuenta 506 $h(524)$, y todos los resúmenes de la cuenta 506 a $h(530)$. Sin embargo, aunque valida todos los resúmenes en su red de transacciones, la cuenta 506 solo guarda los registros de transacciones para las transacciones que ha realizado con otras cuentas, sistemas o servidores. No sabe nada sobre el contenido de los registros transaccionales de las transacciones entre las cuentas 502 y 504, aunque su resumen contiene información que podría ser usada por la cuenta 502 o la cuenta 504 para verificar los resúmenes para esas transacciones.

Lo importante es que el algoritmo que ambas partes usan para generar independientemente el mismo resumen intermedio usa los pasos que las partes intercambian para dar efecto a la

transacción. Por lo tanto, la transacción que genera el registro se convierte en un componente del proceso de la cadena de resumen, y el proceso que genera la entrada de la cadena de resumen es el mismo que el proceso que da efecto a la transacción. Otra forma de verlo es que la transacción genera el resumen como parte de la transacción, y que el resumen y la información que lo acompaña se convierten en la auditoría de la transacción. Se vuelven una y la misma. Con la cadena de bloques, el iniciador de una transacción completa la transacción y envía su registro a la cadena de bloques para su posterior auditoría, lo que agrega otro paso al proceso, en lugar de integrarse en la transacción.

A medida que la transacción en sí se convierte en un componente contemporáneo de la pista de auditoría que proporciona la cadena de resumen, se vuelve imposible tener una transacción cuyos detalles no sean capturados y validados por la pista de auditoría. La mayoría de las pistas de auditoría son "después del evento", ya que el registro de la transacción completada se pasa al sistema de auditoría generalmente después de que se completa la transacción. En esas circunstancias, existe la posibilidad de que el registro recibido por la auditoría no sea el mismo que el registro generado por la transacción. Por lo tanto, los registros de computadora son generalmente considerados como rumores. Integrar una prueba de conocimiento cero con el protocolo PAKE correcto o similar significa que la transacción genera la pista de auditoría, y que la transacción y sus registros se convierten en parte de la pista de auditoría. Esto tiene profundas implicaciones para las transacciones en tiempo real, ya que ahora se auditan y, por lo tanto, se informan en tiempo real.

El proceso de construcción del resumen utilizando pruebas de conocimiento cero puede aplicarse a cualquiera de los escenarios que generan resúmenes en la cadena de resumen. Se puede usar para resúmenes del sistema, resúmenes del servidor de licencias e incluso los resúmenes fuera de línea representados en la Fig 8. Lo único que importa es que el resumen implica una transacción entre dos o más entidades, independientemente de si esas entidades son partes, dispositivos o sistemas. El proceso tampoco excluye el uso de resúmenes estándar. Por lo tanto, un sistema podría usar los resúmenes generados con pruebas de conocimiento cero para las transacciones entre cuentas, independientemente de si los dispositivos están en línea o fuera de línea, pero usar los resúmenes estándar para los resúmenes del sistema y los

resúmenes de licencias. Un segundo sistema puede usar pruebas de conocimiento cero para todos los resúmenes, mientras que un tercer sistema puede usar solo resúmenes estándar.

PAKE de pasos múltiple con múltiples etapas de transacción

Si bien los ejemplos anteriores son ejemplos de cómo usar transacciones que involucran dos o tres etapas con PAKE que requieren dos o tres pasos para permitir que ambos lados de una transacción creen una clave común, el sistema no está restringido a esos ejemplos. La realidad es que el mismo método funcionará para que un sistema admita transacciones que involucren una pluralidad de etapas para usar PAKE que requieren una pluralidad diferente de pasos o pasadas. El sistema simplemente utiliza, sin embargo, la cantidad de PAKE que se ejecute para cubrir todas las etapas de una transacción. Repite la etapa final cualquier cantidad de veces para generar los pasos PAKE requeridos para generar la clave común final, y así generar el resumen de transacción.

Sistema de cadena de resumen con pruebas de conocimiento cero.

Volviendo a la Fig. 6, se muestra una cadena de resumen que puede usar ambos resúmenes generados con pruebas de conocimiento cero y resúmenes clásicos. La Figura muestra dos cuentas 602 y 604, en el mismo sistema 606, junto con los resúmenes del sistema, $h(606)$, $h(608)$, $h(612)$, etc. El sistema crea un nuevo resumen de un registro para cada acción que resulta en un registro, independientemente de donde resida ese registro. Las transacciones entre las cuentas utilizarían pruebas de conocimiento cero para generar los resúmenes intermedios o de transacción para cada una de las cuentas, tal como se establece anteriormente. El resumen del sistema comprendería el resumen del sistema de cada registro a medida que genera ese registro.

Supongamos que la transacción entre las cuentas 602 y 604 en los pasos 614 y 616 involucra tres etapas separadas, con un algoritmo PAKE que permite a las partes construir un resumen común después de tres pasos.

En el primer paso de la transacción, la cuenta 602 intercambia el resumen, $h(610)$, que es el resumen de su registro anterior, con la cuenta del sistema 606 para el resumen del sistema $h(608)$ generado en el paso 608. Agrega este resumen del sistema y su resumen $h(610)$ a la primera etapa de información transaccional generada en el paso 610, construye la primera prueba de conocimiento cero y la pasa a la cuenta 604. La prueba de conocimiento cero acompaña la información que constituye la primera etapa de la información transaccional, el resumen $h(610)$ y el resumen $h(608)$.

En el segundo paso de la transacción, la cuenta 604 intercambia el resumen, $h(604)$, con la cuenta del sistema para el resumen del sistema $h(608)$ generado en el paso 608. Agrega este resumen del sistema y su resumen $h(604)$, que es el resumen de su registro anterior a la primera etapa de la información transaccional, construye la segunda prueba de conocimiento cero y la pasa a 602. La prueba de conocimiento cero acompaña la información que constituye la segunda etapa de la información transaccional, el resumen $h(604)$ y el resumen $h(608)$.

En el tercer paso de la transacción, la cuenta del sistema 606 agrega $h(610)$ y $h(604)$ a su registro y genera un resumen del sistema intermedio $h(612)$.

En el cuarto paso, la cuenta 602 construye la tercera prueba de conocimiento cero con la información que conforma la tercera etapa de la información transaccional, y la envía a la cuenta 604. La tercera prueba de conocimiento cero acompaña la información que constituye la tercera etapa de la información transaccional.

En el quinto paso, las cuentas 602 y 604 construyen independientemente el resumen $h(614i616i)$. Ambas cuentas 602 y 604 agregan este resumen a sus registros. El resumen $h(614i616i)$ es el resumen de la transacción.

En el sexto paso, la cuenta 602 intercambia $h(614i616i)$ con la cuenta del sistema 606 para $h(612i)$, agrega $h(612i)$ a sus registros y genera el resumen $h(614)$ de su registro de la transacción en el paso 614. La cuenta 604 intercambia $h(614i616i)$ con la cuenta del sistema 606 para $h(612i)$, agrega $h(612i)$ a sus registros y genera el resumen $h(616)$ de su registro de la

transacción en el paso 616, y la cuenta del sistema 606 agrega las dos copias de h(614i616i) a su registro, y genera el nuevo resumen del sistema h(612) en el paso 612.

El registro de la cuenta 602 para la transacción en el paso 614 contiene el resumen h(610), el resumen h(604), el resumen del sistema h(608), el resumen de la transacción h(614i616i), el resumen del sistema intermedio h(612i), las tres etapas de la información transaccional, su registro de la transacción, el ID de cuenta y el resumen h(614).

El registro de la cuenta 604 de la transacción en el paso 616 contiene el resumen h(610), el resumen h(604), el resumen del sistema h(608), el resumen de la transacción h(614i616i), el resumen del sistema intermedio h(612i), las tres etapas de la información transaccional, su registro de la transacción, el ID de cuenta y el resumen h(616).

(Los registros de la transacción de la cuenta 602 diferirán de los de la cuenta 604, ya que cada una comenzó y finalizó la transacción en diferentes estados, y cada una es una cuenta diferente con detalles e ID de cuenta diferentes).

El sistema resumen h(612) contiene resúmenes de ambos lados de cada transacción individual, así como la transacción en su conjunto, lo que fortalece considerablemente la cadena de resumen.

Si Tereon administra una transacción entre cuentas en un sistema diferente, entonces el proceso es ligeramente diferente, ya que aquí, cada sistema intercambiará su resumen del sistema y su resumen del sistema intermedio con la cuenta que administra. De lo contrario, el método descrito anteriormente en relación con la Fig. 6 es el mismo, excepto que, en lugar de tener las cuentas 602 y 604 y el sistema 606, la Figura mostraría el sistema 606 con la cuenta asociada 602 y un segundo sistema 605 con la cuenta asociada 604. Con la transacción que tendrá lugar en los pasos 614 y 616, los resúmenes del sistema que resultarán representarán la transacción del sistema en el paso 612 y la transacción equivalente en el segundo sistema 605 correspondiente a la cuenta 604. En realidad, en un sistema que contiene varias cuentas que

pueden realizar transacciones simultáneamente, el sistema generará resúmenes para cada interacción que genere un registro.

Aunque la Figura. 6 muestra resúmenes secuenciales y resúmenes intermedios la realidad será diferente. La Fig. 6a muestra tres cuentas, 602a, 604a y 606a, todas las cuales interactúan con cuentas en servidores externos, junto con la cuenta del sistema 608a. Las etapas de las transacciones se intercalan para ilustrar lo que puede ocurrir cuando las transacciones tienen lugar simultáneamente en un sistema. Para simplificar, todos estos se muestran en el mismo servidor.

En los ejemplos anteriores, en la etapa 612a, la cuenta 602a intercambiará su resumen h(602a) con el sistema 608a para obtener h(612a). El sistema 608a ahora generará lo que los ejemplos anteriores muestran como el resumen intermedio h(616ai). Este subíndice "i" se usa para que la claridad indique que cada transacción involucraría tres resúmenes del sistema, el resumen original antes de la transacción, un resumen del sistema durante una etapa particular de la transacción (el resumen intermedio) y el resumen del sistema al final de la transacción. El subíndice "i" indica el resumen intermedio. El resumen final del sistema sería h(616a) con el razonamiento anterior. Con múltiples transacciones concurrentes o intercaladas, este etiquetado ya no deja claro lo que está sucediendo. En cambio, cada resumen del sistema, independientemente de si se genera o no durante o después de una transacción, es un resumen del sistema, aunque sea un incremento del resumen anterior. Si las tres transacciones tienen lugar para que se inicie la cuenta 602a, entonces se inicia la cuenta 604a, se inicia la cuenta 606a, se termina la cuenta 602a y esa cuenta 606a termina antes de que finalice la cuenta 604a, el orden de los resúmenes puede ser similar al siguiente, siempre que no tengan lugar otras transacciones o acciones en estas o en cualquier otra cuenta en el servidor, y el diagrama, como resultado, es sutilmente diferente de las cifras anteriores.

La cuenta 602a intercambiará su resumen h(610a) con el sistema para obtener h(612a). El sistema ahora usa ese resumen h(610a) para generar el siguiente resumen del sistema h(616a) (esto se habría etiquetado originalmente como h(628ai), ya que el resumen h(628a) es el

resumen final del sistema para esa transacción una vez que la transacción cuenta 602a completa).

La cuenta 604a intercambiará su resumen h(614a) con el sistema para obtener h(616a). El sistema ahora usa ese resumen h(614a) para generar el siguiente resumen h(620a) del sistema.

La cuenta 606a intercambiará su resumen h(618a) con el sistema para obtener h(620a). El sistema ahora usa ese resumen h(618a) para generar el siguiente resumen del sistema h(624a).

Una vez que la cuenta 602a haya creado su resumen intermedio o de transacción, cambiará ese resumen h(622a) por el resumen del sistema h(624a). El sistema ahora usa ese resumen h(622a) para generar el siguiente resumen del sistema h(628a).

Una vez que la cuenta 606a haya creado su resumen intermedio o de transacción, cambiará ese resumen h(626a) por el resumen del sistema h(628a). El sistema ahora usa ese resumen h(626a) para generar el siguiente resumen del sistema h(632a).

Una vez que la cuenta 604a haya creado su resumen intermedio o de transacción, cambiará ese resumen h(630a) por el resumen del sistema h(632a). El sistema ahora usa ese resumen h(630a) para generar el siguiente resumen del sistema h(636a) (no se muestra).

La cadena de resumen permite que un sistema procese una transacción, audite esa transacción y autentique los datos transmitidos o generados por esa transacción al mismo tiempo. Estos pasos ahora se vuelven contemporáneos. No es necesario asumir que un dispositivo informa honestamente una transacción a un sistema de auditoría. La transacción crea la auditoría y la auditoría crea la transacción.

Esto cambia la naturaleza de una transacción realizada por un dispositivo programado. Cualquier dispositivo programado, incluido un dispositivo “IoT”, ahora puede validar y confiar en las transacciones y los datos transmitidos entre él y cualquier otro dispositivo, ya que la transacción, y su auditoría y autenticación son contemporáneas.

No es necesario asumir que un dispositivo enviará un registro preciso de la transacción a un sistema de auditoría, ya que esa transacción y la auditoría se generan como parte del mismo proceso, y esta naturaleza contemporánea cambia la calidad del valor probatorio de ese proceso de auditoría. Cada dispositivo puede confiar en la información transmitida por el otro sin hacer suposiciones en cuanto a la honestidad de ese otro dispositivo. Los datos transmitidos y recibidos son los datos que se tramitan y los datos que se autentican y auditan.

Cuando se combinan con el servicio de búsqueda, los dispositivos que no han interactuado antes ahora pueden autenticarse entre sí, determinar los servicios o funciones que realiza cada uno, y luego comunicarse entre sí y confiar en esa comunicación para llevar a cabo nuestras tareas según lo programado, sin requerir cualquier intervención humana para hacerlo.

La cadena de resumen permite que los dispositivos programados, incluidos los dispositivos “IoT”, funcionen tanto en línea como fuera de línea. Si cuando están fuera de línea, los dispositivos incluyen marcas de tiempo, información sobre el sesgo del reloj de ese dispositivo, el ID de transacción único del dispositivo (generado, por ejemplo, por un contador monótono interno) y otra información de sincronización en la información transaccional, entonces permiten que sus servidores reconstruyan líneas de tiempo precisas que preservan la causalidad de cada transacción cuando esos servidores finalmente reciben los registros de las transacciones fuera de línea de los dispositivos o de los servidores de terceros. La cadena de resumen, tanto en sus modos en línea como fuera de línea, permite a los servidores confiar en el contenido de los registros transaccionales.

Cuando se combinan con el modelo de seguridad de comunicaciones que protege las comunicaciones entre dispositivos, los dispositivos y los servidores pueden comunicarse de una manera impermeable a los ataques de intermediarios. Tereon permite que “IoT” y otros dispositivos programados se comuniquen de forma segura y se basen en los datos transmitidos entre esos dispositivos.

Un ejemplo de este tipo podría ser una red de “IoT” y otros dispositivos programados que funcionan como un conjunto de sensores y controles industriales. El modelo de seguridad permite que estos dispositivos se comuniquen entre sí de forma segura, y al usar el servicio de directorio de búsqueda, permite que esos dispositivos interactúen con nuevos dispositivos a medida que estos se agregan a la colección original. Tereon elimina la necesidad de reconfigurar los dispositivos para permitirles reconocer nuevos dispositivos y confiar en esos nuevos dispositivos. La cadena de resumen permite que los dispositivos confíen en el contenido y los tiempos de las comunicaciones entre ellos, y permite que el operador pueda confiar en los datos generados y transmitidos, sin necesidad de ninguna evaluación humana en cuanto a la veracidad de los datos transmitidos. Un tercero no puede interferir con esos datos, cuyas cadenas de auditoría y autenticación son contemporáneas con su transmisión.

El servicio de búsqueda, cuando se combina con el modelo de seguridad y el servicio de búsqueda, permite a los dispositivos crear interconexiones *ad hoc* en las que pueden confiar y autenticar sin necesidad de intervención humana. Una vez que un dispositivo está autorizado y sus detalles se agregan al servicio de búsqueda, otros dispositivos pueden conectarse a ese dispositivo cuando surja la necesidad de hacerlo. Si ese dispositivo está comprometido de alguna manera, entonces todos los accesos a él se pueden desactivar a través del mismo servicio de búsqueda.

El sistema proporciona un beneficio adicional que surge de su cadena de resumen y su servicio de búsqueda. Como todos los dispositivos están autorizados y auditados individualmente, el sistema puede instruir a dispositivos particulares para que descarguen actualizaciones del software de esos dispositivos cuando sea necesario, los dispositivos solo pueden hacerlo desde fuentes seguras y confiables. El servicio de búsqueda detallará los servicios, interfaces y formatos de datos (por ejemplo) que ofrece y utiliza un dispositivo en particular. Por lo tanto, si un dispositivo desea conectarse a otro dispositivo para acceder a un dispositivo sobreviviente particular y no tiene el software necesario para admitir la interfaz o el formato requerido, entonces éste o el dispositivo al que se está conectando, o ambos dispositivos, si es necesario, pueden comunicarse con un servidor de sistema para descargar el software o la configuración necesarios para permitir que los dos dispositivos se comuniquen entre sí. El hecho de que los

dispositivos conserven el software después de que finalice la comunicación entre dispositivos estará determinado por los servicios que realizan el dispositivo o los dispositivos y la capacidad de dichos dispositivos. La cadena de resumen significa que incluso si eliminan ese software (pueden reinstalarlo cuando vuelvan a comunicarse), los dos dispositivos conservarán una auditoría completa y un registro de las comunicaciones entre dispositivos que luego podrán cargar en otro dispositivo o servidor si es necesario. Esta instalación se extiende a cualquier tipo de dispositivo, desde un dispositivo de “IoT” completamente autónomo hasta cualquier otro dispositivo programado, como un dispositivo de pagos.

Registros distribuidos de la cadena de resumen.

Para proporcionar una replicación distribuida de toda la cadena de resumen, los sistemas Tereon pueden cargar sus cadenas de resumen a un conjunto central de servidores, como el servidor de licencias, servidores de búsqueda o algún otro conjunto de servidores, para todas las transacciones que se produjeron entre última conexión a ese servidor y la conexión actual. El mismo sistema Tereon puede descargar las correspondientes cadenas de resumen para los otros sistemas Tereon. Esto proporciona un libro de contabilidad distribuido de las cadenas de resumen para todas las transacciones para todos los sistemas Tereon, pero sin la sobrecarga de tener que recalcular cada cadena de resumen para cada transacción. Sin embargo, sí impone una carga de almacenamiento adicional en los sistemas Tereon. Los servidores centrales pueden ser globales, como los de los servidores de licencias y de búsqueda, o pueden ser específicos de una industria, regiones o alguna otra restricción. Al restringir el alcance de las copias de las cadenas de resumen, se puede reducir la carga computacional y de almacenamiento de esta variación.

En lugar de limitar el alcance de los servidores centrales, los sistemas que pueden descargar las cadenas de resumen que fueron cargados por otros sistemas pueden ser limitados. Por lo tanto, las cadenas de resumen de un banco solo pueden ser descargables por otro banco, restringidas por el hecho de que ese banco esté en la misma región que el banco que realiza la carga, o si ha realizado transacciones con ese otro banco. De manera similar, el sistema de un

hospital solo puede descargar las cadenas de resumen que se cargaron en un hospital en la misma región. La flexibilidad es no está restringida.

La cadena de resumen utilizada en Tereon tiene una propiedad que es invaluable. Proporciona libros de contabilidad locales pero con autenticación distribuida. Mantiene la información transaccional privada para los usuarios y servicios involucrados en la transacción, pero distribuye la autenticación proporcionada por los resúmenes en todos los servidores, servicios y dispositivos. Los resúmenes generados con las pruebas de conocimiento cero ilustran esto. Solo los sistemas involucrados en una transacción particular tienen la información transaccional. Sin embargo, todos los sistemas y dispositivos que luego interactúan con esos sistemas generan resúmenes que contienen información sobre los resúmenes anteriores de esos sistemas.

La autenticación distribuida es clave, ya que proporciona una barrera computacionalmente imposible para un posible estafador que desea ocultar un registro manipulado.

Con la cadena de bloques, el estafador solo tiene que controlar entre el 25 y el 33% de los servidores para ocultar un registro manipulado y cambiar la cadena de bloques para registrar la manipulación como un registro válido. Una vez hecho esto, el proceso es virtualmente imposible de revertir.

Con la cadena de resumen de Tereon, un estafador tendría que controlar cada servidor de Tereon, cada servicio de Tereon y cada dispositivo de Tereon, y recalculer cada resumen de la cadena en cada uno de esos servidores y dispositivos. Esto es computacionalmente inviable.

La cadena de resumen ofrecerá al menos el mismo nivel de ahorro financiero y eficiencia económica que los proponentes de la cadena de bloques predicen para este último. La diferencia es que la cadena de resumen de Tereon es capaz de hacerlo; la tecnología de “blockchain”, debido a su diseño y las limitaciones inherentes a ese diseño, simplemente no puede hacerlo.

La ventaja de este sistema es que un estafador no podrá eliminar o enmendar un registro de una base de datos sin volver a calcular todos los resúmenes y los resúmenes vinculados asociados con ese registro. Aunque, en teoría, esto podría ser posible si Tereon opera en un solo servidor sin ningún resumen del sistema y sin ninguna conexión a un servidor de licencias, si alguna de las cadenas vinculadas implica una transacción con una parte en otro servidor o dispositivo, entonces el estafador también necesitará volver a calcular todos los resúmenes en el otro servidor o dispositivo. La dificultad de hacerlo aumenta exponencialmente con cada servidor o dispositivo adicional que interactúa con la cadena de resumen después de la fecha y la hora del registro original.

La cadena de resumen permite que una organización pueda garantizar la veracidad de los datos recopilados, generados o administrados por cualquier dispositivo, garantizar el contenido original y la integridad de un registro, y garantizar la integridad y el contenido de cualquier transacción basada en un registro anterior. Esto puede aplicarse a cualquier dispositivo o transacción, desde un dispositivo de pago, un dispositivo médico, un sensor de tráfico, un sensor meteorológico, un detector de flujo de agua, etc.

Esto tiene claros beneficios de gobernabilidad ya que cada libro de contabilidad local es responsabilidad de cada organización individual, sin embargo, aprenden de las demás organizaciones y se apoyan en ellas de una manera que proporciona fortaleza compartida con responsabilidad y capacidad de contabilidad claramente delineadas. La cadena de resumen crea una herramienta técnica para imponer y respaldar la gestión de la información y las transacciones.

Además, cuando la cadena de resumen se utiliza como un componente de un sistema de pagos, como Tereon procesa el dinero fiduciario, su arquitectura se alinea con la forma en que funcionan los pagos hoy en día y ofrece beneficios que igualan o son superiores a las criptomonedas como Bitcoin. Ofrece a los proveedores de servicios de pago establecidos y a los bancos centrales un "batidor de Bitcoin".

Las cadenas de resumen son una parte particularmente emocionante del sistema Tereon, ya que permiten la autenticación muy rápida que es muy segura.

Una de las capacidades únicas de Tereon es su capacidad para crear registros completos, en tiempo real y pistas o vías de auditoría. Los registros de transacciones de Tereon contienen cada pulsación de tecla (aparte de las credenciales de autenticación reales, como un PIN y una contraseña) que requiere una transacción, junto con todos los datos y metadatos relacionados con esa transacción, siempre que sean necesarios para cumplir con las normativas y los requisitos de negocios. Es importante hacer que esos registros sean evidentes, y la secuencia de transacciones hasta y después de la transacción en cuestión sea inviolable, cuando se almacenan en múltiples proveedores de servicios.

La tecnología de cadenas de bloques “blockchain” no puede hacer esto. Solo puede aceptar un registro de una transacción después de que se haya generado ese registro, pero antes de que se autorice. La tecnología de cadenas de bloques acumula varios registros, crea un bloque y luego lo agrega al sistema de cadenas de bloques. Se basa en el hecho de que la cadena de bloques contiene bloques que a su vez contienen información relacionada con todas las transacciones anteriores. A medida que la cadena de bloques agrega bloques adicionales, depende de la existencia de estos bloques para validar los registros y todos los registros anteriores dentro de la cadena de bloques. Esto causa problemas de escala a medida que aumentan los tamaños de los archivos y, si hay una inconsistencia, toda la rama pierde la autenticación.

En lugar de usar la cadena de bloques, o un derivado de ella, la cadena de resumen de Tereon utiliza una estrategia de resumen que aísla cualquier registro sospechoso para la investigación sin socavar la autenticación de las transacciones posteriores. También evita el problema de escalamiento al tener un diseño que se adapta a cualquier tipo de registro, ya sea para registros estáticos o para transacciones en tiempo real.

Los resúmenes, incluidos los resúmenes intermedios, pueden proporcionar la información necesaria para que un administrador recorra la cadena de resumen rápidamente para

determinar y verificar los resúmenes y sus registros respectivos. Así también pueden hacerlo los registros.

Si se produce alguna transacción o acción, eso significa que los resúmenes anteriores se reconciliaron y, por lo tanto, el usuario y el sistema pueden confiar en el resultado de la nueva transacción. Por lo tanto, Tereon puede confiar en los totales acumulados de cada cuenta antes de realizar una transacción. La validez de la cadena de resumen confirma que los totales acumulados son correctos.

Es esta capacidad para aislar el efecto de un registro modificado, eliminado o manipulado, lo que distingue a la cadena de resumen de la cadena de bloques y sus derivados. Por definición, cualquier registro modificado o manipulado que se oculte con éxito en la cadena de bloques efectuará un recálculo completo de esa cadena de bloques. No existe ningún método para detectar y enmendar un registro falso o falsificado que no sea por medio de una decisión democrática de toda la comunidad de “blockchain”, ya que cada cadena de bloque debe modificarse. Esta característica fue identificada por los investigadores de seguridad como un defecto importante en el diseño de la cadena de bloques. Ese diseño no puede ser cambiado.

Con la cadena de resumen, un registro manipulado no puede afectar el resto de la cadena de resumen a menos que el atacante pueda recalcular todos los resúmenes posteriores. Como los resúmenes anteriores a cualquier manipulación fueron, y siguen siendo, válidos, cualquier transacción basada en esos resúmenes y los valores relacionados con dichos resúmenes seguirán siendo válidos.

La cadena de resumen dendrítica para transacciones fuera de línea significa que un servidor puede registrar transacciones fuera de línea realizadas por un dispositivo fuera de línea, incluso si ese dispositivo se pierde o se daña antes de que pueda volver a conectarse al servidor.

La cadena de resumen proporciona soporte completo para validar transacciones fuera de línea, que es algo que la cadena de bloques y sus derivados simplemente no pueden lograr. Los nodos que operan sus copias de la cadena de bloques deben estar en línea para validar los bloques.

Aunque una billetera de bitcoin puede crear una transacción fuera de línea, no puede validar esa transacción hasta que se pone en línea y lleva el registro de esa transacción a los nodos. Incluso entonces la transacción no se valida hasta que uno de los nodos gana la competencia para generar el siguiente bloque en el blockchain y agrega el registro a ese bloque.

Servicio de directorio

Los sistemas existentes, como los sistemas de transporte, las redes de pago como EMV (Europay, MasterCard, Visa) y otros sistemas heredados usan una arquitectura de centro y radio de tal manera que todas las transacciones pasan por una utilidad central que representa un posible punto único de falla o vulnerabilidad y que es caro de escalar.

El sistema Tereon es de igual a igual, donde un servidor se comunica directamente con otro, por lo que las cadenas de resumen para la seguridad son tan importantes, ya que la verificación de la cadena de resumen se produce en todos los elementos de la red de igual a igual.

Como se comentó más arriba, el sistema Tereon tiene un servicio de directorio 216 que es un directorio de las credenciales e información en el sistema que identifica en qué servidor está registrado un usuario o dispositivo 218, o qué servidor ofrece un servicio o función en particular, y habilita que tengan lugar múltiples métodos de autenticación de un usuario 218, ya que almacena una serie de diferentes tipos de credenciales relacionadas con un usuario en particular. Por ejemplo, un usuario 218 puede autenticarse usando su número de teléfono móvil, dirección de correo electrónico, ubicación geográfica, PAN (números de cuenta principales), etc. y almacena en caché todo lo que no es necesario autenticar cada vez.

El servicio de directorio 216 proporciona una capa de abstracción que separa el ID de autenticación del usuario de los servicios subyacentes, los servidores y las cuentas de usuario reales. Esto proporciona una abstracción entre las credenciales que un usuario 218 o comerciante puede usar para acceder a un servicio y la información que Tereon requiere para realizar el servicio en sí. Por ejemplo, en un servicio de pagos, el servicio de directorio 216 vincularía un ID de autenticación, como un número de teléfono móvil, y quizás un código de

moneda con una dirección de servidor. No hay absolutamente ninguna manera de determinar si el usuario 218 tiene una cuenta bancaria, o con qué banco tiene ese usuario 218 bancos.

El servicio de directorio 216 actúa como un intermediario entre varios servicios, de modo que los proveedores de servicios no pueden verse entre sí y, por lo tanto, se proporciona la seguridad de los datos del usuario. Cada servicio definirá un conjunto de campos (variables) y valores que son específicos de ese servicio. Sin embargo, cada servicio tendrá un campo específico y un valor que identifica el servicio.

Cuando se debe completar una transacción con una parte que no se conoce, un servidor Tereon asociado con un usuario 218 envía un URN (nombre de recurso uniforme) al servicio de directorio 216, que devuelve una dirección IP para el servidor Tereon del Proveedor de servicios de pago para un servicio solicitado por el usuario 218. Esto permite que la transacción se complete directamente entre el usuario 218 y el proveedor del servicio en una base de igual a igual. Además, el servidor Tereon conserva la dirección IP en la memoria caché para que las transacciones posteriores no tengan que usar el servicio de directorio 216.

Esta abstracción proporciona seguridad y privacidad para los usuarios y sus detalles de servicio, la flexibilidad para agregar y modificar los servicios subyacentes sin afectar las credenciales de los usuarios públicos, y la capacidad de segmentar y admitir múltiples servicios, cada uno de los cuales puede mantenerse aislado de los demás si es necesario. Ninguno de los campos en el servicio de datos contiene los datos necesarios para iniciar una transacción, y ningún dato de usuario, aparte del ID de autenticación del usuario, se almacena en el servicio de directorio 216.

El servicio de directorio Tereon 216 es, sin embargo, mucho más que esto. Es compatible con múltiples credenciales. Por lo tanto, un usuario 218 puede usar cualquier número de credenciales como ID de pago. Los ejemplos incluyen números de teléfonos móviles, PAN, direcciones de correo electrónico, etc. Mientras la credencial sea única, Tereon la admitirá.

El servicio de directorio 216 puede soportar múltiples servicios. Aquí es donde surge el concepto de una credencial multifacética, o "papel psíquico". Cuando un proveedor de servicios

verifica una credencial en el servicio de directorio 216, solo puede ver si esa credencial está registrada para su servicio, y el servidor de Tereon que atiende esa credencial. El proveedor de servicios no puede ver ningún detalle de ningún otro servicio al que el usuario 218 pueda tener derecho o esté registrado.

Por ejemplo, un teléfono móvil o tarjeta puede convertirse en una credencial de tarjeta de biblioteca en una biblioteca, un boleto de transporte en un autobús o tren, una clave segura para acceder a una habitación o instalación, un dispositivo de pago interno en la cantina de una empresa, un boleto de teatro, y un dispositivo de pago estándar en un supermercado. También podría convertirse en una licencia de conducir, una tarjeta de atención médica o una tarjeta de identificación para demostrar el derecho a un servicio, que podría mostrar una identificación con foto en el dispositivo del comerciante si el servicio lo requiere, etc. Hay pocos límites, si los hay, para el tipo de credencial en que se puede convertir un dispositivo.

Aunque sería difícil disimular el aspecto externo original de una tarjeta (esto podría hacerse una vez que las tarjetas incorporen cubiertas OLED o cubiertas de papel electrónico en color, por ejemplo, donde el servicio podría indicar a la tarjeta que muestre el conjunto y la información que una credencial o un servicio particular requiere), la apariencia de las aplicaciones del teléfono es cambiada por Tereon para reflejar la naturaleza de la credencial y el servicio.

Se puede implementar una función de búsqueda inversa para cada servidor. Esa función permitirá que un servidor compruebe si el servidor que se está comunicando tiene licencia y está autenticado. Esa función no es necesaria, ya que todas las comunicaciones entre dispositivos Tereon, ya sean tarjetas, terminales, móviles o servidores, deben estar firmadas. Sin embargo, puede haber circunstancias en las que un operador necesite o desee la seguridad adicional que brindará una búsqueda inversa. Aquí, el servicio de directorio 216 contendrá varios campos, como servicio, dirección de dominio del servidor Tereon, número del servidor Tereon, operador del servidor Tereon, tiempo de vida, ID de autenticación de terminal, etc. La etiqueta de servicio aquí se referirá a la búsqueda inversa del servidor, en lugar de a un servicio de transacciones.

La Fig. 9 muestra un ejemplo con dos servidores, el servidor 202a y el servidor 202b. Un usuario 218 está registrado con el servidor 202b y accede a un servicio a través de un terminal que está conectado al servidor 202a.

En la etapa 902, un usuario 218 se identifica al terminal con su propio dispositivo, que se identifica automáticamente al terminal. El terminal también pasa su identificación al dispositivo del usuario si usa un dispositivo inteligente. (Si el usuario 218 usa una tarjeta, entonces el terminal solo podría pasar su identificación al dispositivo del usuario si ese dispositivo es una tarjeta de microprocesador. En este caso, la tarjeta se comunicaría con el servidor 202b, el servidor en donde está registrado, a través de un túnel cifrado a través del terminal para pasar la ID del terminal al servidor 202b).

En la etapa 904, el servidor 202a toma la identificación proporcionada por el dispositivo del usuario y compara esa ID con la lista que mantiene. No contiene esa ID y, por lo tanto, nunca ha tratado con el usuario 218 antes. El servidor 202a ahora se contacta con el servicio de directorio 216. El servicio de directorio 216 verifica la firma en la comunicación del servidor 202a y ve que es válida. El servicio de directorio 216 compara el ID con la etiqueta de servicio para el servicio solicitado (la firma del servidor 202a confirma que el servidor está autorizado para realizar una solicitud de ese servicio) y responde con la información que identifica al servidor 202b, junto con el tiempo de caché para información de vida.

En el paso 906, el servidor 202a ahora se contacta con el servidor 202b para confirmar que el dispositivo del usuario está registrado con el servidor 202b para el servicio. El servidor 202a también pasa la ID del terminal al servidor 202b. En la etapa 908, el servidor 202b, si aún no lo ha hecho, puede realizar una solicitud similar al servicio de directorio 216 para buscar el servidor en donde está registrado el terminal. También puede confirmar que el terminal está registrado para el servicio solicitado con el servidor 202a. El servicio de directorio 216 responde con la información que identifica al servidor 202a, junto con el tiempo de caché para la información de vida.

En la etapa 910, el servidor 202a y el servidor 202b ahora se comunican directamente entre sí para llevar a cabo la transacción requerida. Esto puede ser cualquier cosa, desde hacer un pago hasta abrir una puerta.

Los propios servidores de Tereon contienen la información necesaria para iniciar una transacción, y solo se comunicarán con otros servidores o dispositivos con licencia y autenticados.

Una vez que los servidores se hayan comunicado con el servicio de directorio 216 y entre sí, almacenarán en caché los datos hasta que los datos caduquen en sus propios servicios de directorio mini.

En este caso, las comunicaciones para establecer la conexión entre el servidor 202a y 202b de Tereon son obvias y simples. Esto se muestra en la Figura. 10.

En la etapa 1002, el usuario 218 se identifica con el terminal conectado al servidor 202a con su propio dispositivo, que se identifica automáticamente con el terminal. El terminal también pasa su identificación al dispositivo del usuario si usa un dispositivo inteligente.

En la etapa 1004, el servidor 202a toma la identificación proporcionada por el dispositivo del usuario y compara esa ID con la lista que mantiene. Los datos que contiene son válidos y, por lo tanto, el servidor 202a se pone en contacto con el servidor 202b para confirmar que el dispositivo aún está registrado para el servicio solicitado. El servidor 202a también pasa la ID del terminal al servidor 202b. El servidor 202b confirma que el dispositivo está registrado en él. La memoria caché del servidor 202a contiene datos válidos en la ID del terminal y, por lo tanto, se contacta con el servidor 202b para confirmar que el terminal todavía está registrado en él. El servidor 202b confirma esto.

En la etapa 1006, el servidor 202a y el servidor 202b ahora se comunican directamente entre sí para llevar a cabo la transacción requerida.

Si los datos almacenados en la memoria caché caducan en un servidor, ese servidor simplemente contacta con el servicio de directorio 216 como antes. Si un usuario 218 ha migrado a otro servidor, entonces las comunicaciones difieren ligeramente. La Fig. 11 ilustra este caso. La diferencia es que la primera comunicación con el servidor 202b, basada en la información en caché ahora desactualizada, obligará al servidor 202a a buscar los nuevos datos en el servicio de directorio 216.

En el paso 1102, el usuario 218 se identifica con el terminal conectado al servidor 202a con su propio dispositivo, que se identifica automáticamente con el terminal. El terminal también pasa su identificación al dispositivo del usuario si usa un dispositivo inteligente. El servidor 202a toma la identificación provista por el dispositivo del usuario y verifica esa identificación con la lista que mantiene. Contiene esa ID y ve que los datos almacenados en la memoria caché muestran que la ID está registrada con el servidor 202b.

En el paso 1104, el servidor 202a ahora se pone en contacto con el servidor 202b para confirmar que el dispositivo del usuario está registrado con el servidor 202b para el servicio. El servidor 202a también pasa la ID del terminal al servidor 202b. El servidor 202b responde que el ID ya no está registrado con él.

En el paso 1106, el servidor 202a ahora se contacta con el servicio de directorio 216. El servicio de directorio 216 verifica la firma en la comunicación del servidor 202a y ve que es válida. El servicio de directorio 216 compara el ID con la etiqueta de servicio para el servicio solicitado y responde con la información que identifica al servidor 202c, junto con el tiempo de caché para la información de vida.

En el paso 1108, el servidor 202a ahora se pone en contacto con el servidor 202c para confirmar que el dispositivo del usuario está registrado con el servidor 202c para el mismo servicio, y así lo hace. El servidor 202a también pasa la ID del terminal al servidor 202c, y actualiza su caché con los nuevos detalles de la ID del dispositivo del usuario.

En el paso 1110, el servidor 202c, si aún no lo ha hecho, puede realizar una solicitud similar al servicio de directorio 216 para buscar el servidor en donde está registrado el terminal. También puede confirmar que el terminal está registrado para el servicio solicitado con el servidor 202a. El servicio de directorio 216 responde con la información que identifica al servidor 202a, junto con el tiempo de caché para la información de vida.

En el paso 1112, el servidor 202a y el servidor 202c ahora se comunican directamente entre sí para realizar la transacción requerida.

El servicio de directorio 216 siempre conservará un rastro completo de las ID de usuario que un usuario 218 ha registrado, tanto antiguas como nuevas, junto con las fechas en que se asignaron al usuario 218.

El servidor 202c solo retiene la información que se relaciona con la identificación registrada desde la fecha en que se registró la identificación. El servidor 202b conservará los datos relacionados con el período en donde atendió esa ID.

La capa de abstracción proporcionada por el servicio de directorio 216 va más allá al segmentar los servicios. Por lo tanto, en el ejemplo anterior, el servidor 202a solo puede solicitar la información que identifica al servidor que ha registrado el dispositivo del usuario para el servicio requerido.

El servidor 202a debe firmar cada comunicación que realiza con un dispositivo, y esa firma identificará el servicio en donde está involucrada la comunicación. Si un servidor puede ofrecer más de un servicio, tendrá una clave privada para cada uno de esos servicios y utilizará esa clave para firmar las comunicaciones relevantes.

Los propios servidores de Tereon, en el caso de estos servidores 202a y 202b, contienen la información de búsqueda que identifica los datos de la cuenta del usuario a partir de las etiquetas o la información proporcionada. Por lo tanto, solo el servidor 202b contiene los datos que asignan la ID del dispositivo del usuario a la cuenta del usuario; La información en el

servicio de directorio 216 es simplemente un puntero al servidor 202b. El dispositivo del usuario se puede registrar fácilmente en diferentes servidores para diferentes servicios. Lo que permite a los servidores de Tereon encontrar el servidor correcto es la combinación de la ID del dispositivo del usuario y las credenciales que definen el servicio.

Una vez que el servidor 202a se comunica con el servidor 202b y pasa la etiqueta de servicio, la identificación del usuario y cualquier otro dato transaccional relevante (por ejemplo, edad, moneda, cantidad, etc.), el servidor 202b busca los datos relevantes del usuario y realiza su parte de la transacción. El servidor 202a nunca ve los datos del usuario. Todo lo que ve es el ID de autenticación del usuario y los datos transaccionales que le pasa el servidor 202b.

Del mismo modo, el servidor 202b nunca ve la información que identifica la cuenta a la que está conectado el terminal. Simplemente ve el ID de terminal y los datos transaccionales que le pasa el servidor 202a.

Papel psíquico - la credencial multifacética

Uno de los efectos más intrigantes de la estructura del servicio de directorio es su capacidad para crear credenciales multifacéticas *ad hoc* que se adaptan a servicios particulares, como y cuando se requieren esas credenciales. No es necesario que los servicios se hayan previsto en el momento en que se crea el servicio de directorio para que el servicio de directorio pueda proporcionar esas credenciales. Esto se conoce como “papel psíquico”.

La credencial multifacética *ad hoc* significa que el dispositivo del usuario se convierte en la credencial que un servicio en particular puede requerir, y no más. Ofrece exactamente la información requerida para autenticar, autorizar o beneficiarse de un servicio, y eso es todo lo que ve el proveedor del servicio.

Como ejemplo, el usuario 218 se ha registrado para una serie de servicios diferentes, como un servicio de pagos de su banco y un servicio de préstamo de la biblioteca en su biblioteca local.

Debido a que tuvo que proporcionar su fecha de nacimiento cuando se registró en Tereon, automáticamente tiene acceso a un servicio de verificación de edad.

La Fig. 12 ilustra cómo el servicio de directorio 216 puede dirigir un servidor solicitante (servidor 202a) a dos servidores diferentes (servidores 202b y 202c), dependiendo del servicio que el usuario 218 haya solicitado. Si es necesario, también se pueden usar dos o más servicios de directorio separados para servicios separados. Lo que es importante es que los datos transaccionales son parte de un resumen y están separados de los datos de la cuenta subyacente.

El usuario 218 necesita verificar su edad, por ejemplo, para comprar una bebida alcohólica en un bar (servicio 2). En este caso, los pasos 1202 a 1210 se realizan como pasos 902 a 910 en la Fig. 9, aunque entre los servidores 202a y 202c, en lugar de los servidores 202a y 202b. En consecuencia, en la etapa 1210, el servidor 202a y el servidor 202c se comunican directamente entre sí. En este caso, el servidor 202a quiere verificar que el usuario 218 tiene más de 21 años. El servidor 202c simplemente confirma que tiene más de 21 años.

Si el operador requiere una confirmación adicional debido a requisitos legales o reglamentarios, el servidor 202c podría enviar una imagen tipo pasaporte del usuario 218 para que se muestre en el terminal, de modo que el operador pueda ver que realmente está hablando con el usuario 218. El servidor también podría enviar una pregunta para que el usuario 218 responda con el fin de proporcionar una confirmación adicional de que es el usuario correcto, aunque no es necesario hacerlo ya que el usuario 218 ya se ha identificado con el servidor 202a. El operador nunca puede ver la edad real del usuario o cualquier información personal que no sea necesaria, ya que no es necesaria. Todo lo que el operador necesita saber es que el usuario 218 tiene la edad suficiente para comprar una bebida alcohólica. Si el usuario 218 usa su dispositivo para pagar su bebida, entonces el terminal conectado al servidor 202a se pondrá en contacto nuevamente con el servidor 202c, pero esta vez para un servicio de pago (servicio 1).

El usuario 218 ahora va a su biblioteca local y quiere pedir prestado un libro (servicio 3). En la etapa 1212, el usuario 218 se identifica con el terminal en la biblioteca con su propio

dispositivo, que se identifica automáticamente con el terminal. El terminal en la biblioteca está conectado al servidor 202b. El terminal también pasa su identificación al dispositivo del usuario si usa un dispositivo inteligente.

En la etapa 1214, el servidor 202b toma la identificación proporcionada por el dispositivo del usuario y compara esa ID con la lista que mantiene. Contiene esa identificación pero el caché está desactualizado. El servidor 202b ahora contacta con el servicio de directorio 216. El servicio de directorio 216 verifica la firma en la comunicación del servidor 202b y ve que es válida. El servicio de directorio 216 compara el ID con la etiqueta de servicio para el servicio solicitado y responde con la información que identifica al servidor 202c, junto con el tiempo de caché para la información de vida.

En la etapa 1216, el servidor 202b ahora se pone en contacto con el servidor 202c para confirmar que el dispositivo del usuario está registrado con el servidor 202c para el servicio, lo que hace. El servidor 202b también pasa la ID del terminal al servidor 202c, y actualiza su caché con los nuevos detalles de la ID del dispositivo del usuario.

En la etapa 1218, el servidor 202c, si aún no lo ha hecho, puede realizar una solicitud similar al servicio de directorio 216 para buscar el servidor en donde está registrado el terminal. También puede confirmar que el terminal está registrado para el servicio solicitado con el servidor 202b. El servicio de directorio 216 responde con las credenciales que identifican al servidor 202b.

En la etapa 1220, el servidor 202b y el servidor 202c ahora se comunican directamente entre sí para realizar la transacción requerida. El servidor 202b desea saber si el usuario 218 puede pedir prestado un libro (servicio 3), y el servidor 202c confirma que el usuario 218 está registrado en el servicio de bibliotecas para prestar libros (este es un servicio que el operador de Tereon proporciona para las bibliotecas). Si el usuario 218 necesita usar su dispositivo para pagar una tarifa para pedir prestado un libro, entonces el terminal contactará al servidor 202c nuevamente, pero esta vez para un servicio de pago (servicio 1).

El servidor 202c no necesita proporcionar ningún servicio a la biblioteca. El usuario 218 podría haberse registrado fácilmente con otro servidor, por ejemplo el servidor 202d (no se muestra), en cuyo caso el servidor 202d confirmaría al servidor 202b que el usuario 218 podría pedir prestado libros. Lo importante es que, en el primer caso, el servidor 202a solo confirmó que el usuario 218 tenía más de 21 años. No sabe que puede pedir prestados libros, y no sabe que el usuario 218 puede pagar con Tereon. Del mismo modo, el servidor 202b sabe que el usuario 218 puede pedir prestados libros, pero no tiene idea de que tiene más de cierta edad o que puede pagar con Tereon.

Un servidor solicitante también puede realizar múltiples solicitudes a servidores separados si necesita reunir un conjunto de credenciales para una transacción en particular. Por ejemplo, supongamos que el usuario 218 quiere tomar prestada una película que tiene restricción de edad. En este caso, el servidor solicitante haría dos solicitudes separadas, una para verificar la edad del usuario y otra para verificar que está registrado para tomar prestadas películas de la biblioteca. Tereon reunirá las credenciales individuales y verificadas para construir el conjunto de credenciales que requiere la biblioteca.

La estructura del servicio de directorio 216 permite que los servidores que entregan las credenciales individuales se separen. Por lo tanto, un servidor solicitante puede interrogar a cualquier número de servidores para obtener las credenciales individuales que requiere para construir el conjunto de credenciales necesarias para determinar si puede entregar un servicio particular a un usuario 218.

La Fig. 13 ilustra el caso en que el servidor 202a necesita obtener credenciales de tres servidores 202c, 202d y 202e para construir una credencial multifacética para ofrecer un servicio a un usuario 218. Por ejemplo, el servicio 2 en el servidor 202d puede ser un servicio para alquilar una película, lo que requeriría la verificación de la edad como primera credencial del servidor 202c, una credencial de miembro del servidor 202d y una credencial de fondos suficientes del servidor 202e.

La relación no es necesariamente de uno a uno, es decir, en la que cada uno de los tres servidores tiene una y solo una credencial. Cualquiera de los tres servidores puede entregar más de una credencial al servidor 202a. Solo pueden entregar una credencial al servidor 202a. El número de credenciales es irrelevante. Lo que importa es que el servidor 202a puede comunicarse con más de un servidor externo para obtener las credenciales que requiere para permitir que un usuario 218 acceda a un servicio.

Puede ser que el servidor 202a, en donde el usuario 218 accede a un terminal, ya tenga algunas credenciales que necesita para entregar algunos servicios al usuario 218. Sin embargo, a los efectos de la protección de datos, el usuario 218 no desea proporcionar ciertos detalles al servidor 202a (por ejemplo, su edad, etc.). Si todo lo que debe hacer el servidor 202a es verificar que el usuario 218 tiene más de una cierta edad, o si se le permite ordenar ciertos productos, entonces simplemente puede contactar a los servidores que confirmarán o rechazarán esas preguntas. Esto es extremadamente útil para los sitios web de comercio electrónico: pueden confirmar ciertos hechos o parámetros sin conocer los detalles exactos. Esencialmente, el servicio de directorio 216 puede actuar como un proveedor de prueba de conocimiento cero o un notario confidencial. Tereon puede probar o refutar un hecho o parámetro al servidor 202a sin revelar qué es ese hecho.

Por lo tanto, la credencial para un servicio particular podría comprender credenciales de 202a, 202c, 202d, 202e y otros servidores. Las credenciales pueden estar en un servidor o pueden distribuirse en varios servidores.

Esto es extremadamente poderoso ya que les permite a las personas y organizaciones probar que tienen derecho a un servicio sin necesidad de revelar información que no es necesario divulgar. Nuevamente, tomando el ejemplo del sitio web de comercio electrónico, el usuario 218 puede registrar su nombre y dirección en el sitio web. Sin embargo, su banco tiene sus credenciales de pago, un servidor del gobierno registra el hecho de que está autorizado para comprar artículos restringidos, su compañía ferroviaria local tiene su autorización de viaje y el servidor de la autoridad de salud puede confirmar su edad.

El método para reunir un conjunto de credenciales ad hoc para un servicio no se aplica solo a los usuarios y sus dispositivos. Se puede aplicar igualmente a sensores, dispositivos y servicios autónomos, como, por ejemplo, los dispositivos de baja frecuencia que necesitan conectarse a diferentes servicios en diferentes momentos. Simplemente pueden reunir las credenciales necesarias para esos servicios a medida que se requieran esos conjuntos de credenciales.

Cambio de cuenta

Un problema importante que a menudo retrasa la adopción de nuevos sistemas es la dificultad percibida de transferir datos de sistemas heredados a esos nuevos sistemas sin pérdida o interrupción del servicio. El mismo problema afecta a las actualizaciones del sistema, donde los operadores a menudo optan por quedarse con las configuraciones iniciales de hardware y software en lugar de poner esto al día y actualizarlo debido a su percepción de los peligros de perder datos en cualquier actualización o puesta al día.

El servicio de directorio 216 contrarresta estos problemas al proporcionar un mecanismo para mover datos, cuentas e información de configuración sin problemas de un servidor o almacenamiento de datos a otro. Uno de los bloques para respaldar las transferencias de cuentas en tiempo real entre instituciones es la cuestión de cómo capturar y tratar los pagos “en el aire”. Esta industria actualmente tiene un sistema de transferencia de cuentas que toma 18 meses en total (7 días para el cambio inicial y luego 18 meses para detectar pagos o transferencias). Esto también podría aplicarse al cambio de un conjunto de datos de un almacenamiento de datos a otro.

El servicio de directorio 216 proporciona una capa de abstracción que separa el ID de autenticación del usuario de los servicios subyacentes, los servidores y las cuentas de usuario reales. Por lo tanto, un usuario 218 puede mantener su ID de autenticación mientras cambia los servicios y los servidores subyacentes en los que está registrado su dispositivo.

El proceso de cambio de cuenta se describe mejor con un ejemplo. En este ejemplo, el usuario 218 realiza sus operaciones bancarias con el Banco A. La Fig. 14 ilustra la relación del usuario

con el banco A y su servidor 202a de Tereon. El Banco B también es compatible con Tereon en el servidor 202b, aunque el usuario 218 todavía no es un cliente. El usuario 218 decide mover su cuenta del banco A al banco B.

La Fig. 15 ilustra el proceso que el usuario 218 se compromete a transferir su cuenta del banco A al banco B. En este ejemplo, el usuario 218 no está sobregirado y no tiene préstamos del banco A.

En el paso 1502, el usuario 218 abre una cuenta con el Banco. B y registra su tarjeta y su teléfono móvil con ese banco y su servidor Tereon 202b.

En el paso 1504, el servidor 202b de Tereon del banco B busca el número de teléfono móvil del usuario y el PAN de su tarjeta en el servicio de directorio de Tereon 216 y detecta que ambos están registrados en el banco A.

En el paso 1506, el servidor 202b de Tereon del banco B ahora contacta al usuario 218 para confirmar que desea trasladar su registro al banco B y el usuario 218 confirma esto ingresando un código de autenticación adicional que se le envió específicamente para este propósito.

En el paso 1508, el servidor 202b de Tereon del banco B ahora se contacta con el servidor 202a del servidor y le informa que el usuario 218 ha solicitado mover sus cuentas e ID al banco B, y lo ha confirmado.

En el paso 1510, el servidor Tereon 202a del bnco A ahora envía al usuario 218 una solicitud para confirmar que desea mover su cuenta y el usuario 218 confirma su movimiento.

En el paso 1512, el servidor Tereon 202a del banco A ahora confirma esto con el servidor 202b Tereon del banco B, e informa al servidor 202b del banco B sobre los registros de cuentas del usuario, los saldos, las configuraciones, las instrucciones de pago, etc. El servidor 202b del Banco B configura estas cuentas exactamente de la misma manera que las del Banco A, o tan cerca como sea posible para proporcionar los servicios que está autorizado a brindar.

Por ejemplo, el usuario 218 tiene tres cuentas de divisas separadas con el banco A que le permiten mantener GBP, USD y EUR. Desafortunadamente, el banco B solo proporciona cuentas en GBP y USD, aunque puede pagar y recibir EUR desde y hacia cualquier cuenta. El servidor 202b del Banco B informa al usuario 218 de esto cuando el usuario abre la cuenta, y decide convertir el EUR a GBP. El Banco B le ordenaría entonces al Banco A que envíe el EUR como GBP.

En el paso 1514, el servidor 202b Tereon del banco B ahora informa al servicio de directorio 216 que las ID del usuario ahora están registradas con su servidor 202b.

En el paso 1516, el servidor 202b Tereon del banco B informa al servidor 202a del banco A que ha registrado las ID de usuario en el servicio de directorio 216 y le indica al banco A que le transfiera los saldos.

En el paso 1518, el banco A confirma con el servicio de directorio 216 que ya no administra las ID del usuario. El servicio de directorio 216 establece una fecha y hora de inicio contra el nuevo registro de ID en el banco B, y establece una fecha y hora de finalización en el campo contra el registro anterior en el banco A. El banco A ahora configura su servicio de directorio para informar a cualquier servidor que intente para pagarle al usuario 218 que ya no posee las cuentas del usuario e instruir a ese servidor para que busque los detalles del usuario en el servicio de directorio 216. Lo hace ingresando la fecha y la hora en su campo de fecha de finalización. El banco B ahora recibirá todos los pagos realizados al usuario 218 que inicialmente se dirigieron al banco A.

El servicio de directorio 216 ahora puede capturar pagos al aire, que son pagos realizados a la cuenta anterior del usuario después de que el usuario 218 haya cambiado a una nueva cuenta. De manera similar, Tereon también puede detectar los pagos aplazados que deben salir de la cuenta anterior. Estos saldrán ahora de la nueva cuenta una vez que se hayan transferido los saldos, una tarea que toma minutos en lugar de días, semanas o meses.

En el paso 1520, el banco A transfiere los saldos al banco B. El banco B informa al banco A que ha recibido los fondos.

En el paso 1522, el banco A cierra las cuentas del usuario, e informa al usuario 218 que lo ha hecho y transfirió sus saldos a su nuevo banco. En el paso 1524, el banco B informa al usuario 218 que ahora recibió sus saldos del banco A.

Si el usuario 218 estaba sobregirado en una o más de sus cuentas en el banco A, y el banco B estuvo de acuerdo en aceptar su negocio, entonces el banco B transferiría los balances al banco A en los pasos 516 y 520, y las cuentas correspondientes del usuario en el banco B lo harían estar sobregirado. El usuario 218 también podría decidir transferir fondos entre sus cuentas en el banco A para eliminar cualquier sobregiro antes de transferir sus cuentas al banco B.

Para los pagos, el sistema de numeración Tereon distingue entre usuario, organización, cuenta, tipo de servicio y transacción. Todos ellos tienen sistemas de numeración separados. Estas características permiten que el servidor de directorios administre el proceso mediante el cual un usuario 218 mueve su cuenta a un nuevo proveedor de servicios en tiempo real. La estructura del servicio de directorio 216, junto con la capacidad de procesar transacciones en tiempo real, permite a los usuarios cambiar cuentas en minutos, en lugar de días.

El servicio de directorio 216, como se describe anteriormente, junto con el procesamiento en tiempo real de todas las transacciones, elimina el problema de las transacciones en el aire, como los pagos en el aire. Con Tereon, las transacciones simplemente no pueden entrar en un estado en el aire. Las mismas o bien se completan o bien se cancelan.

Tereon también apoya la noción de portabilidad de la cuenta, como la portabilidad de la cuenta bancaria, una característica que aumentaría la competencia en el mercado y, sin embargo, una que los bancos y reguladores creen que es imposible de implementar. Debido a que Tereon no usa los detalles de la cuenta directamente, pero usa una credencial por separado para identificar a cada pagador y beneficiario, inserta una abstracción entre el usuario 218 y los

detalles de la cuenta bancaria del usuario. Es esta abstracción, que proporciona el servicio de directorio 216, la que facilita el cambio de cuentas y la portabilidad.

Cambio de credenciales

El servicio de directorio 216 permite a los operadores y usuarios reemplazar las credenciales de identificación existentes con nuevas credenciales, y reutilizar las credenciales anteriores sin confundir las transacciones con los usuarios anteriores de la identificación. La capa de abstracción proporcionada por el servicio de directorio 216 permite a Tereon hacer esto.

Si un usuario 218 transfiere su cuenta a otro servidor, entonces ese usuario 218 podrá retener una credencial particular, como un PAN, o el servidor puede emitir al usuario 218 una nueva credencial. En este último caso, el servidor original puede reutilizar la credencial casi inmediatamente. Debido a que cada credencial tiene una marca de fecha y hora que refleja cuando se emite a un usuario 218, un nuevo usuario 218 de una credencial particular podría usar esa credencial casi de inmediato.

Cada credencial tiene una marca de fecha y hora para cuando se emite a un usuario en particular en un servidor en particular. Como cada transacción también conserva una marca de fecha y hora, cada servidor Tereon retiene la credencial utilizada para cada transacción, Tereon simplemente utiliza estos componentes para enrutar las transacciones al destino correcto. Por ejemplo, un usuario 218 puede comprar algo de un comerciante con la credencial A (por ejemplo, un número de teléfono móvil) y luego, unos días más tarde, cambiarse a otro banco cuando necesite usar otra credencial B (por ejemplo, una nueva Número de teléfono móvil). Más tarde, el usuario 218 devuelve el artículo al comerciante, ya que está defectuoso. El comerciante simplemente localiza la transacción y presiona el reembolso. Aunque la transacción original utilizó la credencial A, el servidor para la credencial A informa de una marca de fecha y hora que indica un cambio en la credencial. El servidor del comerciante busca la credencial A y descubre que el usuario 218 que usó la credencial A en el momento de la transacción ahora usa la credencial B. El servidor ahora se comunica con el servidor para la

credencial B, lo que confirma que el usuario 218 para la credencial B usó la credencial A en el momento de la transacción, y los servidores comienzan el proceso de reembolso.

El usuario A puede estar seguro de que el usuario de B no es fraudulento, ya que el modelo de seguridad de Tereon requiere que todas las comunicaciones estén firmadas. El servidor 202b solo podrá firmar sus comunicaciones si tiene una licencia válida del servidor de licencias, y el dispositivo del usuario B solo podrá firmar sus comunicaciones si el servidor 202b es válido, ya que habrá emitido y verificará la licencia del dispositivo. El usuario B no podrá completar una transacción a menos que conozca las credenciales correctas para autorizar una transacción o para acceder a la aplicación en el dispositivo.

En otro ejemplo, un usuario puede haber ingresado el número de teléfono móvil de un contacto en su directorio telefónico y ahora quiere hacer una transferencia P2P sorpresa a ese contacto. Tereon busca en los registros para ese número y descubre, como se indicó anteriormente, que el contacto ha cambiado los números móviles (si el contacto es un usuario de Tereon). Confirma con el servidor correcto que el usuario que usa el nuevo número usado para usar el número anterior registrado con el servidor anterior. Tereon también admite la función en la que un contacto puede configurar su cuenta para permitir que el servidor de directorio actualice el número de móvil de ese usuario u otra credencial de Tereon cuando ciertos contactos aprobados intentan realizar una transacción con ellos a través de una credencial antigua. En este ejemplo, la sobrina de la tía ha configurado su cuenta para actualizar a todos los miembros de la familia, por lo que la próxima vez que su tía acceda a su lista de contactos, verá el nuevo número de teléfono móvil de su sobrina.

La Fig. 16 ilustra un ejemplo para el servidor 202a, el servidor 202b y el servicio de directorio 216. Aquí el usuario anterior ha migrado sus cuentas del servidor 202a al servidor 202b. 202a es el banco como servidor, y 202b es el servidor del banco B.

El usuario antiguo usó inicialmente el número de teléfono móvil 1 como su ID. Después de migrar sus cuentas, continuó usando el número de teléfono móvil 1 por un tiempo. Las comunicaciones entre el usuario 218, el servicio de directorio 216 y los servidores 202a y 202b

procedieron como se describe anteriormente y se exponen en la Fig. 15. Las entradas en el servicio de directorio muestran que el usuario 218 usó el servidor 202a desde la fecha-hora 1 hasta la fecha-hora 3, y que el usuario usó el servidor 202b desde la fecha-hora 2. La ligera superposición es para asegurar que todos los pagos en el aire se capturen y que no haya un intervalo de tiempo en donde el usuario no tenga un servidor en donde esté registrada su identificación. (Es posible evitar la superposición de las entradas de fecha y hora asegurándose de que el servidor al que migra la cuenta controla todas las entradas de fecha y hora e ID para esa migración, y así es como podría funcionar la migración del sistema).

En algún momento, el usuario 218 decide cambiar los números móviles. Registra su nuevo número de móvil 2 como su ID con el servidor 202b y anula el registro del número de móvil 1. El servidor 202b notifica al servicio de directorio 216 del cambio, que ahora muestra que el usuario comenzó a usar el número de móvil 2 como su ID en la fecha y hora 4 y que el número de móvil 1 dejó de ser una ID para el servidor 202b en la fecha y hora 5.

Más tarde, un nuevo usuario crea una cuenta con el servidor 202a y registra el número de teléfono móvil 1 como su ID en la fecha y hora 6. Es posible que al nuevo usuario se le haya dado el teléfono móvil antiguo, o que el operador móvil haya liberado ese número, el servidor 202a notifica al servicio de directorio 216 que ha registrado la ID (después de verificar que la ID esté disponible), y así, el servicio de directorio ahora muestra que el número de teléfono móvil 1 está registrado en el servidor 202a a partir de la fecha y hora 6.

En el ejemplo mostrado en la FIG. 16, si el usuario anterior usó una tarjeta emitida por el banco A 202a, una vez que el usuario 218 ha transferido sus cuentas al banco B 202b, el banco puede emitir una nueva tarjeta al usuario 218 con una credencial, como un PAN, que está registrado para ello. El usuario 218 activa esa tarjeta una vez que la recibe y el servidor 202b del banco B informa al servidor 202a del banco A que la credencial original del usuario ya no está en uso. El Banco B registra la nueva credencial con el servicio de directorio de Tereon 216. El usuario 218 podría haber solicitado conservar la credencial original, en cuyo caso el banco A podría haberle cobrado una pequeña tarifa por hacerlo si el banco A aceptó la solicitud. Por lo tanto, Tereon admite el número de tarjeta o la portabilidad PAN.

El usuario podría, en algún momento en el futuro, decidir dejar de usar la tarjeta originalmente emitida por el banco A y así liberar esa credencial. Es posible que el banco A no pueda reutilizar esa credencial PAN durante seis meses calendario completos después de que el banco B la haya liberado o después de que el usuario haya transferido sus cuentas al banco B; el tiempo exacto dependerá de lo que permitan los reguladores del banco. Después de ese tiempo, puede usar la credencial porque el servicio de directorio 216 no solo contiene una lista de números móviles, PAN u otras credenciales; también contiene una lista de las fechas en que esas credenciales se registraron y las fechas en que expiraron o se publicaron usuario por usuario.

El método de cambio de cuenta permite que el sistema capture los pagos al aire. También proporciona una forma extremadamente flexible y robusta de dirigir las transacciones que siguen de transacciones anteriores, en función de las credenciales utilizadas para esas transacciones anteriores. Los reembolsos por transacciones anteriores son un ejemplo real de esto. Un comerciante que realice un reembolso contra una identificación antigua podrá reembolsar la cuenta correcta, ya que el servicio de directorio 216 indicará a su servidor que pague la identificación correcta, incluso si la identificación original se reutilizó posteriormente. EMV y las tecnologías de búsqueda móvil actuales asumen que los números nunca se reutilizan. Desafortunadamente, a veces lo son.

La Fig. 16 ilustra esto. Supongamos que en algún momento entre la fecha y la hora 1 y la fecha y la hora 2, el usuario anterior compra un artículo de un comerciante que utiliza un dispositivo con el número de teléfono móvil 1 como su ID. Más tarde, ese artículo demuestra ser defectuoso y el usuario desea un reembolso.

Si ese usuario 218 va al comerciante entre la fecha y hora 1 y la fecha y hora 2 para obtener un reembolso, entonces el sistema Tereon indicará al sistema del comerciante que realice el pago de reembolso a la cuenta del usuario en el sistema 202a (ya que el usuario todavía no ha cerrado su cuenta).

Si ese usuario 218 va al comerciante entre la fecha y hora 2 y la fecha y hora 4 para obtener un reembolso, entonces el sistema Tereon indicará al sistema del comerciante que realice el pago

de reembolso a la cuenta del usuario en el servidor 202b, aunque el pago por el artículo vino originalmente del servidor 202a.

El método de cambio de cuenta también tendrá en cuenta la nueva identificación del usuario. Si ese usuario 218 va al comerciante después de la fecha y hora 4 para obtener un reembolso, y usó su número de móvil 2 como su ID, entonces el sistema Tereon indicará al sistema del comerciante que realice el pago a la cuenta del usuario en el servidor 202b aunque el pago del artículo provino originalmente del servidor 202a e incluso el usuario usó originalmente el número de teléfono móvil 1 como su ID de pago.

Lo mismo se aplicará a los registros de PAN, direcciones de correo electrónico y cualquier otra credencial reutilizable. (Las credenciales biométricas no se pueden reutilizar por razones obvias).

El sistema permite segmentar las credenciales a cualquier nivel de minuciosidad. Un ejemplo de esto en los pagos implica monedas o códigos de moneda, donde un usuario puede usar diferentes ID para diferentes monedas en la misma o en servidores separados.

La Fig. 17 ilustra un ejemplo para el servidor 202b, el servidor 202c y el servicio de directorio 216. El usuario 218 ya ha migrado sus cuentas del servidor 202b al servidor 202c de manera similar a la ilustrada en la Fig. 16, y con las comunicaciones entre servidores gestionadas como se ilustra en la Fig. 15.

El usuario 218 usa inicialmente el número de móvil 1 como su ID. Después de migrar sus cuentas, continúa usando el número de teléfono móvil 1 por un tiempo para transacciones tanto en la moneda 1 como en la moneda 2. Las entradas en el servicio de directorio 216 muestran que el usuario 218 usó el servidor 202b desde la fecha-hora 1 hasta la fecha-hora 3, y que el usuario usó el servidor 202c desde la fecha-hora 2. La ligera superposición es para asegurar que todos los pagos en el aire se capturen y que no haya un intervalo de tiempo en donde el usuario no tenga un servidor en donde se haya registrado su ID.

En algún momento, el usuario 218 decidió usar un nuevo móvil para transacciones en la moneda 2. Registró su nuevo número de móvil 2 como su ID con el servidor 202c para transacciones en la moneda 2. El servidor 202c notificó al servicio de directorio 216 del cambio, que ahora muestra que el usuario comenzó a usar el número de móvil 2 como su ID para todas las transacciones en la moneda 2 en la fecha y hora 4 y que el número de móvil 1 dejó de ser una ID para cualquier transacción en moneda 2 a fecha-hora 5.

La Figura 17a ilustra otro ejemplo para el servidor 202b, el servidor 202c y el servicio de directorio 216. En la Figura, el usuario 218 ya ha migrado su cuenta de moneda 1 del servidor 202b al servidor 202c de manera similar a la ilustrada en la Fig. 16, y con las comunicaciones entre servidores gestionadas como se ilustra en la Fig. 15.

Después de migrar su cuenta, el usuario continuó durante un tiempo utilizando el número de teléfono móvil 1 durante un tiempo para transacciones tanto en la moneda 1 como en la moneda 2. Las entradas en el servicio de directorio 216 muestran que el usuario 218 usó el servidor 202b desde la fecha-hora 1 hasta la fecha-hora 3 para transacciones en ambas monedas, y que desde la fecha-hora 2 usó el número de teléfono móvil 1 como su ID con el servidor 202c para transacciones en moneda 1. Las entradas del servicio de directorio también muestran que el usuario continuó utilizando el número de teléfono móvil 1 como su ID con el servidor 202b para transacciones en la moneda 2.

En algún momento, el usuario 218 decidió usar un nuevo móvil para transacciones en la moneda 2. Registró su nuevo número de móvil 2 como su ID con el servidor 202b para transacciones en la moneda 2. El servidor 202b notificó al servicio de directorio 216 del cambio, que ahora muestra que el usuario comenzó a usar el número de móvil 2 como su ID para todas las transacciones en la moneda 2 en la fecha y hora 4 y que el número de móvil 1 dejó de ser una ID para cualquier transacción en moneda 2 a fecha-hora 5.

Antes de la fecha y hora 4, el usuario 218 usaba su número de teléfono móvil 1 como ID para todas sus transacciones. El servicio de directorio 216 simplemente dirigió las transacciones al servidor 202b si esas transacciones estaban en la moneda 2, y al servidor 202c si esas

transacciones estaban en la moneda 1. El hecho de que el usuario haya registrado la misma ID en dos servidores es irrelevante, ya que es el conjunto completo de credenciales que rigen a qué servidor se dirige la transacción. El sistema de un comerciante que realiza transacciones en la moneda 1 con el usuario por primera vez después de la fecha y hora 2 nunca sabría que el usuario había utilizado previamente el servidor 202b para transacciones en esa moneda. Del mismo modo, el sistema de ese comerciante no sabría que el usuario usó la misma ID para las transacciones en la moneda 2 en el servidor 202b a menos que el sistema ingresara en una transacción con ese usuario en la moneda 2.

Tereon hace más que simplemente cambiar un usuario 218 de una red a otra. Como ya se mencionó, los métodos usuales de cambio de usuarios no tratan con los pagos al aire. El sistema de cambio de cuenta más avanzado actualmente disponible, según lo reclaman sus creadores, requiere un proceso manual de 18 meses para detectar dichos pagos antes de que el usuario pueda valerse por sí mismo. Durante el período de 18 meses, tanto los bancos como el usuario deben trabajar para garantizar que transfieran todas las instrucciones de pago existentes de la cuenta antigua a la cuenta nueva. Tereon elimina este requisito por completo.

Los bancos corrientes no pueden reutilizar ninguna credencial de pago. El mecanismo de cambio de cuenta de Tereon elimina esta limitación, y los bancos pueden volver a emitir los PAN y los números de cuenta después de un cierto período de tiempo si los reguladores desean permitirles hacerlo.

Aunque el método se conoce como una función de cambio de cuenta, en realidad tiene muchas aplicaciones más allá del cambio de cuenta básico. Por ejemplo, puede proporcionar una conmutación por error a un proveedor de servicios de respaldo en caso de que fallen los sistemas centrales del banco, por lo que proporciona una forma de migrar datos de un sistema a otro mediante la traducción de un formato de datos a otro sin pérdida de información.

Otro ejemplo es agilizar la portabilidad numérica en sistemas móviles. Actualmente, si un usuario cambia su número de teléfono móvil de un proveedor a otro, el primer proveedor debe redirigir todas las llamadas al nuevo proveedor. Si el usuario cambia a un tercer proveedor,

entonces el primer proveedor debe enrutar la llamada al segundo proveedor, quien luego debe enrutar la llamada al tercer proveedor. Esto es extremadamente ineficiente y costoso de hacer y, sin embargo, los operadores deben admitir la portabilidad numérica. Tereon evita la necesidad de redireccionar llamadas varias veces.

Si los operadores usaran Tereon para soportar la portabilidad numérica, entonces no necesitarían soportar múltiples sobresaltos. Una vez que un usuario decide transferir su número del primer operador al segundo operador, el segundo operador simplemente deberá informar al servidor de directorio que ahora admite ese número de teléfono móvil. El primer operador desviaría las llamadas para ese número al servidor de directorio, lo que enrutaría la llamada al segundo operador. Cada vez que el usuario volviera a portar su número, el nuevo operador informaría al servidor del directorio del cambio y el servidor del directorio simplemente enrutaría las llamadas al operador que atendió ese número. (Si los usuarios tienen cuentas bancarias, como IBAN, que son únicas a nivel mundial, Tereon admitirá la portabilidad de las cuentas bancarias de la misma manera que lo hace con la portabilidad de números móviles).

Un ejemplo similar es uno en donde un operador migra servicios y dispositivos “IoT” de un servidor a otro para actualizar el sistema Tereon donde una simple migración de, por ejemplo, una máquina física, una máquina lógica, una máquina virtual, un contenedor o cualquier otro mecanismo comúnmente usado para contener código ejecutable, no será suficiente.

Otro ejemplo es operar como una herramienta de migración del sistema. Esto sería, por ejemplo, cuando un operador desea migrar un servicio, junto con las cuentas en las que se registran los dispositivos, de una versión del sistema Tereon a una versión actualizada. El operador simplemente configuraría el servidor antiguo para transferir los registros del dispositivo, las cuentas y las configuraciones del sistema al nuevo servidor y el sistema llevaría a cabo la transferencia. Cada cuenta se transferiría, junto con sus datos y registros de auditoría, y los servidores actualizarían el servicio de directorio 216 a medida que avanzaban las transferencias. Ahora, cuando los dispositivos en el campo, ya sean dispositivos de pago, sensores de tráfico, dispositivos “IoT”, etc., deseen comunicarse con su servidor, el servicio de

directorio 216 simplemente los redireccionará a su servidor antiguo o nuevo, dependiendo de si contactaron con su servidor antes o después de que sus cuentas fueran transferidas.

Los ejemplos anteriores demuestran cómo Tereon facilita la portabilidad de credenciales y admite credenciales multifacéticas ad hoc. Esto tiene aplicaciones de gran alcance y lleva a Tereon a prácticamente cualquier ámbito de red donde esa red necesita administrar credenciales.

Marco extensible

Los flujos de trabajo para los sistemas de procesamiento de transacciones existentes son muy a menudo de naturaleza estática. Una vez implementados, son muy difíciles de cambiar, y los servicios u operaciones que soportan los sistemas siguen siendo inflexibles.

Hasta ahora, si un proveedor de pagos lanzaba un servicio, entonces el patrón de pago para ese servicio se volvía estático. El proveedor solo podría modificar el servicio lanzando un servicio de reemplazo o enmendado y emitiendo nuevas tarjetas o aplicaciones para respaldar ese servicio. Esta es una de las razones por las que, a pesar del conocimiento universal de las graves deficiencias de EMV, es imposible reparar el sistema, ya que eso significaría retirar todas las tarjetas EMV existentes, reprogramar y lanzar la infraestructura de pagos de EMV y luego emitir nuevas tarjetas. Esto requeriría la colaboración de miles de emisores y compradores.

Tereon pone toda la funcionalidad en el “detrás de escena” (conocido como back-end) utilizando el SDASF y el “detrás de escena” puede guiar el dispositivo del comerciante en tiempo real a través del proceso. Esto permite al proveedor de servicios crear nuevos servicios que pueden ser tan granulares como el usuario individual.

El marco extensible es un marco que se encuentra dentro del sistema Tereon y permite la adición de nuevos servicios sin tener que necesariamente reconfigurar el sistema Tereon. El marco extensible funciona con el servicio de directorio 216 para proporcionar una serie de ventajas para el sistema Tereon.

Estructura de mensaje flexible

El marco extensible se proporciona en parte mediante una estructura de mensaje flexible en la que cualquier tipo de datos o registro puede proporcionarse con un campo de longitud variable, de modo que el sistema de Tereon pueda modificar la longitud del campo para operar con sistemas heredados o incompatibles.

El marco extensible permite agregar un nivel adicional de seguridad a la infraestructura de comunicaciones al cambiar el orden estándar de los procesos. En muchas industrias, los pagos son solo un ejemplo, las comunicaciones utilizan estructuras de mensajes fijas. Esto conduce a una debilidad que los delincuentes pueden explotar incluso si las comunicaciones están cifradas. Los mensajes estructurados son vulnerables a un ataque en profundidad. Aunque las organizaciones y otros pueden seguir protegiendo la integridad de un mensaje utilizando un código de autenticación de mensaje resumen (HMAC), el HMAC no conserva el secreto absoluto que el mensaje debería tener.

El marco extensible diseña el problema de los sistemas estáticos para cualquier sistema de procesamiento de transacciones. Ofrece la flexibilidad para operar junto con los sistemas y servicios existentes, y permite a los proveedores actualizar los servicios existentes y construir nuevos servicios sin necesidad de reiniciar una infraestructura o emitir nuevos dispositivos de punto final, como las tarjetas. El marco es lo suficientemente flexible como para permitir que los proveedores creen servicios que pueden personalizar para usuarios individuales. Esto se explicará a continuación.

Ofuscación

Uno de los riesgos teóricos que enfrenta cualquier sistema con formatos de mensajes estructurados es que el uso repetido de un formato de mensaje proporcionará suficiente material para que un pirata informático lo utilice en un ataque de fuerza bruta. Esto es cierto

para los sistemas que no implementan algoritmos de cifrado correctamente con algún tipo de siembra aleatoria. Sin embargo, es un riesgo que debe superarse.

El marco extensible permite a los operadores y usuarios interrumpir la necesidad de enviar un mensaje estructurado entre dispositivos y servidores. En su lugar, el mensaje puede ser ofuscado.

Cada una de las comunicaciones de transacción en Tereon comprenderá dos o más campos junto con las etiquetas de esos campos. En lugar de seguir un orden fijo de campos para cada comunicación, el orden se puede modificar de manera aleatoria. Dado que cada campo siempre estará acompañado por su etiqueta de identificación, se debe garantizar que los dispositivos en cada extremo de una comunicación primero descifren y luego ordenen los campos antes de procesarlos.

Por ejemplo, al usar un extracto del ejemplo provisto por la documentación de la Notación de Objetos de JavaScript (JSON) (aunque, por supuesto, se pueden usar otros formatos en el sistema), las siguientes tres representaciones serían las mismas:

- {"versión": 1, "primer Nombre": "John", "Apellido": "Smith", "está Vivo": cierto, "edad": 25}
- {"versión": 1, " primer Nombre ": "John", " está Vivo ": cierto, " Apellido ": "Smith", "edad": 25}
- {"edad": 25, " primer Nombre ": "John", " está Vivo ": cierto, " Apellido ": "Smith", "versión": 1}

Un atacante no sabría qué textos cifrados, si los hubiera, contienen información conocida y en el mismo orden. El modo exacto de ofuscación dependería del formato utilizado y del protocolo de serialización utilizado, en su caso, pero el principio sigue siendo el mismo.

El modo de ofuscación tiene una ventaja adicional. El contenido de las comunicaciones predefinidas se puede ampliar sin romper el protocolo de comunicaciones. Si un dispositivo recibe campos que no puede procesar, simplemente descartará esos campos y sus valores. Por lo tanto, se podrían incluir uno o más pares de campos y valores aleatorios que el sistema

descarta, pero que agregan incertidumbre adicional a las comunicaciones. Las siguientes tres comunicaciones serían las mismas:

- {"versión": 1, "primerNombre": "John", "nunavez": 5780534, "Apellido": "Smith", "estáVivo": cierto, "edad": 25}
- {"quiénsabe": "698gtHGF", "versión": 1, "primerNombre": "John", "estáVivo": cierto, "Apellido": "Smith", "edad": 25}
- {"edad": 25, "primerNombre": "John", "estáVivo": cierto, "Apellido": "Smith", "queesesto": "Jor90% hr," "versión": 1}

En cada una de las comunicaciones anteriores, los dispositivos descartarían el par desconocido de campo y valor.

Los nombres de los campos podrían confundirse aún más mezclando los casos de manera aleatoria para cada comunicación. Los dispositivos procesarán estos campos a su forma canónica.

Así, las siguientes tres comunicaciones serían las mismas:

- {"versión": 1, "primerNombre": "John", "nUnavez": 5780534, "Apellido": "Smith", "estáVivo": cierto, "Edad": 25}
- {"quiénsabe": "698gtHGF", "versión": 1, "primerNombre": "John", "estáVivo": cierto, "Apellido": "Smith", "edad": 25}
- {"edad": 25, "primerNombre": "John", "estáVivo": cierto, "Apellido": "Smith", "que es esto": "Jor90% hr," "versión": 1}

Si se envía un mensaje de la versión 2 que puede contener campos adicionales, cualquier dispositivo que solo comprenda la versión 1 rechazará el mensaje o, si se garantiza la compatibilidad con versiones anteriores, procesará los campos que comprende y descartará el resto. Esto podría mejorarse aún más al proporcionar un campo que indicara qué versiones eran compatibles con algunos de los campos.

Esto elimina la vulnerabilidad a un ataque en profundidad. La estructura del mensaje también se puede conservar, pero con campos de longitud variable. Nuevamente esto logra un resultado similar. Al utilizar también un HMAC, tanto la integridad del mensaje como su secreto están protegidos. Si los sistemas centrales de la organización final requieren mensajes en un formato estructurado, Tereon simplemente reestructurará los mensajes una vez que hayan llegado a un servidor y los reformateará de la forma requerida por los sistemas centrales de la organización. Por lo tanto, el marco extensible permite superar los problemas de seguridad con los sistemas heredados, y aun así operar con dichos sistemas.

El marco extensible admite cualquier tipo de datos o registro, con exactamente el mismo nivel de seguridad y flexibilidad que se mencionó anteriormente.

Componentes de flujo de trabajo abstraídos

En las soluciones existentes, un proceso de pagos se definiría en el software, se implementaría, se probaría y luego se lanzaría. Esa estructura de transacciones de pago ahora sería fija y no podría cambiarse sin un esfuerzo significativo para recuperar y reemplazar o reprogramar dispositivos, terminales y servidores.

Tereon no hace esto. En su lugar, construye el proceso de pagos a partir de componentes individuales, cada uno de los cuales interactúa con sus componentes conectados. Esos componentes esencialmente establecen el flujo de trabajo del proceso. Cada componente se puede actualizar y agregar funciones sin afectar el proceso de pago en sí. Esto abstrae los componentes del proceso del dispositivo, de modo que una transacción, una vez definida, puede aplicarse a cualquier número de dispositivos, ya sean tarjetas y terminales de tarjetas, teléfonos móviles o portales web.

Cada componente pasa las instrucciones y la información al siguiente componente, dependiendo del resultado de la instrucción que recibió. Las instrucciones pueden ser transaccionales o pueden incluir controles, como la forma en que debería funcionar el siguiente

componente (por ejemplo, solicitar un PIN si es opcional, ofrecer un conjunto de opciones, mostrar un mensaje en particular y las respuestas esperadas o permitidas).

Esto proporciona la capacidad de alterar los servicios de pago existentes y construir nuevos servicios sin necesidad de reprogramar o reemplazar los puntos finales existentes. En este momento, una vez que un proveedor de servicios de pago implementa un sistema de pago, el proveedor de servicios de pago no puede cambiar fácilmente el sistema sin reemplazar los puntos finales. Los sistemas existentes son esencialmente estáticos. Esto los reemplaza con un sistema dinámico.

El marco extensible permite al operador planificar el flujo de trabajo para una transacción en particular utilizando estos componentes. Permite la creación de flujos de trabajo, incluidos árboles de decisión y similares. Un operador podría modificar un flujo de trabajo existente, simplemente reorganizando los componentes existentes, agregando nuevos componentes que proporcionen nuevas funciones o eliminando componentes. Para hacer esto en un sistema existente, los servidores y los terminales necesitarían ser reprogramados, y es posible que las tarjetas deban ser reemplazadas.

Un ejemplo de esto se muestra en las Figs. 18 a 20. Los componentes mismos se representan como bloques en una pantalla de terminal para facilitar la visualización de lo que hace cada componente. Sin embargo, los componentes se aplican igualmente a las transacciones móviles, las transacciones del portal web y las transacciones de terminales de tarjeta. Para cambiar un flujo de trabajo existente, el orden y la conexión de los componentes simplemente se modificarían. Para crear un nuevo flujo de trabajo, los componentes necesarios simplemente se conectarán en el orden deseado.

Los procesos de pago normales crearían procesos de pago separados para pagos sin contacto, para pagos con contacto y para pagos móviles. Por lo tanto, el componente 1804 normalmente aparecerá a la izquierda en la cadena, justo después del componente 1802 de *“transacción completa en el tiempo”* como se muestra en la Fig. 18.

Sin embargo, al mover este componente más hacia la derecha, como se muestra en la Fig. 19, e insertando dos componentes de decisión adicionales 1902 y 1904 en la cadena, el operador puede crear un proceso de pagos único que pueda administrar los pagos por contacto, pagos sin contacto y pagos móviles en un solo proceso de pagos.

El operador podría ir más lejos. Quizás le gustaría agregar una oferta especial de temporada al proceso una vez que el sistema haya identificado al cliente. Como se muestra en la Fig. 20, podría en cualquier momento mover el componente 1804 más hacia la derecha e insertar en su posición original un nuevo componente 2002 que automáticamente hace al cliente una oferta antes de que el comerciante necesite ingresar la cantidad y el PIN. El operador podría configurar ese componente para que funcione en los 24 días previos a la Navidad, por ejemplo, y proporcionar un componente diferente para los días posteriores a Año Nuevo. Esto alteraría dinámicamente el proceso de pago para la temporada de Navidad y Año Nuevo, sin requerir que un operador recuerde y re programe dispositivos. Los componentes simplemente instruirían al dispositivo de pantalla, ya sea un teléfono móvil o un terminal de tarjeta, para mostrar la oferta al cliente. El operador podría deshabilitar fácilmente el requisito de PIN configurando el componente 1804 para deshabilitar el requisito de PIN. Del mismo modo, si el componente no tenía una función para requerir un PIN, el operador podría actualizar el componente para incluir esa función.

El operador podría ir aún más lejos y construir un árbol de decisiones completo para permitir al cliente elegir entre una gama de ofertas si así lo deseara. Una vez que finaliza la temporada de ofertas, el operador simplemente eliminará los nuevos componentes y el proceso reanudará su estructura original.

Lo que es importante tener en cuenta es que en ningún momento el operador debe recordar los dispositivos para cambiar el proceso. Simplemente reconfigura el proceso en la parte de “back-end” y luego implementa ese cambio en el momento y la fecha de su elección.

El marco para proporcionar la administración interna y el funcionamiento de los servidores Tereon se puede configurar exactamente de la misma manera, donde los componentes del

marco interactúan con el contexto del acceso para controlar cómo y a qué información pueden acceder los usuarios y administradores y las tareas que pueden realizar.

Servicios dinámicos

El marco extensible permite a una organización crear e implementar nuevos servicios rápidamente. El operador simplemente define estos servicios enlazando los bloques que se requieren y definiendo los mensajes relevantes. En lugar de tener que emplear programadores para escribir el código de un servicio, el marco permite que los departamentos de marketing y TI (Tecnología Informática) implementen los servicios escribiendo un archivo de definición para definir el flujo de trabajo, utilizando un sistema gráfico para "dibujar el flujo de trabajo", o cualquier otro proceso de definición de flujo de trabajo. Una vez que ha verificado el flujo de trabajo, el operador simplemente implementa el flujo de trabajo mediante la vinculación de pasos o bloques definidos, y Tereon pone el servicio a disposición de todos los usuarios calificados.

Por ejemplo, un operador necesitaría usar un bloque para aceptar un pago de cualquier valor con un bloque posterior para solicitar un PIN. Sin embargo, si un operador desea ofrecer un sistema de control de acceso, ese mismo operador puede crear un bloque para permitir el acceso sin PIN a un conjunto de espacios mientras usa un bloque para solicitar un PIN para acceder a otro conjunto de espacios.

Esto significa que, a diferencia de los sistemas existentes, el sistema permite a las organizaciones diseñar e implementar nuevos servicios, o modificar o eliminar servicios existentes, incluso después de que esa organización haya lanzado el sistema de procesamiento de transacciones, sin necesidad de reemplazar los dispositivos emitidos a los usuarios. Si un dispositivo entiende y puede operar cualquiera de los pasos definidos, ese dispositivo admitirá cualquier servicio que la organización defina usando esos pasos. Una vez que una organización define un servicio, el sistema hará que ese servicio esté inmediatamente disponible para el usuario o usuarios seleccionados.

Dispositivos abstraídos

El marco extensible lleva el principio de abstracción más allá y abstrae los dispositivos en sí. El marco define los componentes del proceso para cada clase de dispositivo que se relacionan con las funciones de esos dispositivos. Los componentes del proceso interactuarán con esos componentes funcionales. Dependiendo de las funciones disponibles, los componentes del proceso indicarán a los componentes funcionales que realicen tareas, como qué generar y qué introducir.

Granularidad

Tereon puede identificar cada dispositivo, usuario y cuenta individualmente, y puede acceder al contexto dentro del cual un usuario está utilizando un dispositivo para acceder a un servicio. Por lo tanto, el operador puede configurar componentes y opciones dentro de esos componentes para desencadenar una acción basada en el contexto dentro del cual un usuario individual accede al servicio. Tereon efectivamente permite que el operador adapte los servicios a cada usuario, al dispositivo de cada usuario y al contexto dentro del cual el usuario usa ese dispositivo para acceder a los servicios.

Por ejemplo, un usuario puede ver una selección de tres ofertas en una transacción, otro usuario solo puede ver una oferta que recibiría automáticamente, mientras que un tercero puede no ver una oferta en absoluto.

Si el proceso se relaciona con el acceso a registros, por ejemplo, registros de pacientes, entonces un usuario puede acceder a sus registros y administrar derechos de acceso si el usuario accede a esos registros en una instalación médica o en un dominio local. Sin embargo, si el usuario (u otra persona) accede a esos registros fuera de esos dominios, entonces el usuario solo puede ver un subconjunto de esos registros, o no podrá acceder a esos registros en absoluto (dependiendo de la configuración de contexto para ese servicio).

Si el usuario accede al servicio utilizando un terminal de tarjeta, los componentes indicarán al terminal de tarjeta que muestre la información relevante. Si el usuario accede al mismo servicio utilizando un teléfono móvil u otro dispositivo de pantalla, los componentes indicarán a esa pantalla que muestre la información relevante. De esta manera, la capa de abstracción del marco extensible se convierte en un dispositivo agnóstico. Puede utilizar cualquier pantalla y punto de acceso adecuados para controlar la interacción del usuario con el sistema.

Lo mismo se aplica a los servicios que se proporcionan. La cuenta de cada usuario tendrá el nivel de servicios predeterminado del proveedor. Cuando un operador agrega nuevos servicios o modifica los servicios existentes para uno o más usuarios, las cuentas de esos usuarios tendrán esos servicios. La clave del servicio será una combinación de la etiqueta de sus proveedores, el número de cuenta del usuario y la etiqueta de registro del dispositivo del usuario. Esto crea una ruta dendrítica corta a las definiciones de servicio y las reglas para ese usuario.

Por ejemplo, el remitente puede usar un teléfono móvil, en donde ha establecido sus reglas para permitir transferencias interactivas o automáticas. El destinatario puede haber configurado su dispositivo para aceptar transferencias automáticas. En este caso, el dispositivo del remitente simplemente seguirá los pasos para realizar una transferencia automática. La etiqueta de servicio no incluye ninguna información sobre si la transferencia es interactiva o no; que se deja a la información sobre el servicio almacenado en los servidores del remitente y del destinatario.

Si el destinatario configuró su dispositivo para que acepte transferencias interactivas o automáticas, entonces el dispositivo del remitente le preguntará al remitente qué modo usar. El destinatario puede haber configurado su dispositivo para aceptar transferencias automáticas entre ciertos tiempos y transferencias interactivas en otros momentos. En este caso, el servidor Tereon del destinatario simplemente informará al servidor del remitente sobre el modo de transferencia que se debe utilizar, según la hora del día del destinatario.

Si el remitente o el dispositivo del destinatario solo aceptarán transferencias interactivas, entonces si el destinatario y el remitente están en línea al mismo tiempo, seguirán los pasos para llevar a cabo la transferencia. Si el destinatario solo tiene una tarjeta, entonces el destinatario deberá ir a la terminal de un comerciante para realizar su parte de la transacción. Si el destinatario está desconectado, entonces el remitente seguirá sus pasos, pero el destinatario deberá seguir sus pasos en la transacción, como aceptar la transferencia e ingresar su PIN, antes de que Tereon complete la transferencia. Hasta entonces, Tereon mantendrá la transferencia en una facilidad de custodia, similar a la forma en que trata con las transferencias a los usuarios que no son de Tereon.

Interfaces dinámicas

El marco extensible conduce a servicios dependientes del contexto, p. ej. ofrece ayuda a un usuario a encontrar su asiento en un evento, procesos específicos del comerciante, etc. Permite a una organización personalizar los servicios y experiencias que cada usuario tendrá cuando ese usuario interactúe con Tereon, el grado en que los servicios están disponibles según el contexto, qué botones pueden aparecer, qué opciones pueden estar disponibles, etc.

La cantidad de servicios con los que cada usuario y cada comerciante pueden interactuar depende completamente de la superposición entre los servicios a los que el usuario individual puede acceder y los servicios que el comerciante puede ofrecer.

Por ejemplo, cuando un comerciante puede ofrecer pagos, depósitos y retiros, y si un usuario acude a ese comerciante, y ese usuario solo puede acceder a los pagos en un comerciante, entonces el usuario y el comerciante solo verán las funciones relacionadas con un pago. A saber, el pago y el reembolso. Si un usuario acude al mismo comerciante y ese usuario puede acceder a pagos, depósitos y retiros, ese usuario verá todas esas funciones. Si ese comerciante ya no tiene fondos suficientes para respaldar depósitos o retiros, entonces, cuando el usuario de servicio completo acuda a ese comerciante, el usuario solo verá las funciones de pago en su dispositivo o en la terminal del comerciante. Ese comerciante tampoco aparecerá en ninguna búsqueda de comerciantes que ofrezcan depósitos o retiros en relación al comerciante. Puede

ser que un usuario no pueda acceder a ciertos servicios en algunos comerciantes, pero puede acceder a esos servicios en otro comerciante. El marco manejará estos casos.

La interfaz dinámica complementa el uso de una credencial multifacética y permite que el dispositivo y sus aplicaciones asociadas se conviertan en algo parecido al "papel psíquico", como se explicó anteriormente. En este caso, el dispositivo proporciona solo los servicios disponibles, y la interfaz se adapta a solo esos servicios, independientemente de la pluralidad de servicios para los cuales el usuario pueda estar registrado. Puede parecer un dispositivo de pago de un servicio, un boleto de transporte a otro, una llave de la puerta a otra, etc. Los proveedores de servicios no necesitan emitir dispositivos separados para acceder a sus servicios y, como tal, esto reduce tanto la complejidad como el costo de ofrecer servicios y la actualización de dichos servicios.

El marco extensible permite que el dispositivo cambie su apariencia, y presente las credenciales y los servicios requeridos por el contexto dentro del cual y para el cual se utiliza el dispositivo. Así, por ejemplo, puede adaptar la pantalla de un cajero automático independiente, como uno en una tienda de comestibles, para que adopte la apariencia del operador del usuario cuando accede al cajero automático, y presenta solo los servicios a los que el usuario se ha suscrito.

Interacción con otras capas

La capacidad del marco extensible para interactuar con otros componentes dentro del sistema Tereon es una característica fundamental del marco extensible. Aparte de la seguridad contextual, que a su vez incorpora el modelo de seguridad más amplio, las instrucciones del marco extensible se pueden incrustar dentro de la información transaccional transmitida a través de la cadena de resumen (como se describe en relación con la cadena de resumen sin pruebas de conocimiento).

Modo fuera de línea

Tereon ofrece tres modos fuera de línea; usuario fuera de línea, comerciante fuera de línea y ambos fuera de línea.

En los dos primeros casos, Tereon completa una transacción en tiempo real yendo al otro lado de la plaza; es decir, el usuario se comunica con su servidor Tereon a través de la terminal del comerciante y el servidor Tereon del comerciante. Ni el comerciante ni el usuario experimentarán un deterioro del servicio. Tereon usa un protocolo PAKE, o un protocolo con funciones similares, para crear la ruta segura a través de los tres lados del cuadro para el dispositivo relevante.

En el tercer caso, cuando ambos dispositivos están fuera de línea, la impresión inmediata sería que Tereon no podría verificar en tiempo real si el usuario o comerciante tenía fondos suficientes para respaldar una transacción y crear el muy expuesto riesgo crediticio para el que Tereon fue diseñada para superar. Este no es el caso.

Al utilizar las características del marco extensible y una versión de la cadena de resumen, Tereon puede garantizar que el sistema aún pueda verificar los fondos. Tanto el usuario como el comerciante podrán realizar todas sus funciones. El usuario deberá usar una tarjeta móvil o una tarjeta de microprocesador, pero ni el usuario ni el comerciante verán un deterioro en el servicio que experimentan. Tanto el dispositivo del comerciante como el dispositivo del usuario almacenarán los detalles cifrados de la transacción entre ellos, y una muestra aleatoria de las transacciones fuera de línea anteriores que el comerciante ha realizado. El dispositivo del comerciante establece el número máximo de copias de cada transacción que pasará a la tarjeta o al teléfono de un usuario.

Tereon usaría una combinación de lógica de negocios y sus modelos de seguridad y cadenas de resumen para evitar que cualquier usuario use una combinación de dispositivos fuera de línea y dispositivos en línea para retirar más de lo que existe dentro de una cuenta. Una cuenta solo puede admitir dispositivos fuera de línea si esa cuenta proporciona una función de crédito. La lógica fuera de línea no requiere crédito, aunque los reguladores de un proveedor de servicios pueden requerir un permiso para ofrecer crédito.

Si un dispositivo no está autorizado para operar fuera de línea, no podrá realizar transacciones con ningún otro dispositivo cuando esté fuera de línea. Su modelo de seguridad y autenticación evitará que lo haga, ya que su firma lo identificará como que solo admite transacciones en línea, y el dispositivo no podrá procesar ninguna transacción que afecte el valor de cualquier cuenta en la que esté registrado.

Si un dispositivo puede admitir transacciones fuera de línea, entonces el proveedor del servicio limitará esto a una cierta cantidad (ya sea un límite de crédito o una fracción del saldo de la cuenta, que siempre se actualiza con el dispositivo en línea), que es la asignación fuera de línea. El dispositivo solo podrá autorizar transferencias o pagos de fondos desde la cuenta hasta el valor total o esa asignación fuera de línea. El proveedor de servicios puede, por supuesto, autorizar al dispositivo para que acepte transferencias o fondos, y puede limitar el valor de esas aceptaciones (la asignación de aceptación fuera de línea). Si el usuario accede a la cuenta mientras el primer dispositivo está fuera de línea, ya sea directamente a través de un portal o con otro dispositivo en línea, entonces el usuario solo podrá autorizar transferencias o pagos fuera de la cuenta hasta el valor del saldo de la cuenta menos la asignación fuera de línea.

Tereon reconcilia todas las transacciones fuera de línea una vez que uno de los dispositivos que contiene los registros relevantes vuelve a estar en línea. Como norma, recibirá varias copias de algunas transacciones, pero puede usarlas para confirmar las conciliaciones anteriores.

Por lo tanto, si el servidor recibe registros de las transacciones fuera de línea de los servidores de terceros que se relacionan con los pagos o transferencias realizadas al dispositivo fuera de línea, una vez que haya recibido suficientes copias de esas transacciones, las procesará. y agregar esos fondos al saldo de la cuenta. Del mismo modo, si el servidor recibe registros de transacciones de terceros fuera de línea que se relacionan con los pagos o transferencias realizadas desde el dispositivo fuera de línea, una vez que haya recibido copias suficientes de esas transacciones, procesará esas transacciones y restará esos fondos del saldo de la cuenta y de la asignación restante fuera de línea.

Aunque las ilustraciones anteriores se refieren al pago, ya que son fáciles de visualizar, los mismos modos de operación pueden aplicarse a cualquier tipo de sistema transaccional. Un ejemplo sería la interacción entre dispositivos “IoT” u otros componentes industriales. Al crear flujos de trabajo que comprenden módulos que se pueden reorganizar, insertar o eliminar, los operadores pueden reconfigurar los dispositivos para que funcionen de nuevas maneras sin necesidad de recuperarlos, reprogramarlos y reinstalarlos.

Los operadores pueden reutilizar dispositivos en el campo, cambiar la forma en que operan, o incluso permitir que los dispositivos controlen otros dispositivos y modificar sus flujos de trabajo en función de los cambios que esos dispositivos detecten en el entorno en donde operan dichos dispositivos.

Los dispositivos “IoT” también pueden modificar los flujos de trabajo de los demás modificando el conjunto de módulos que forman los flujos de trabajo cuando sea necesario. El modelo de seguridad que rige las comunicaciones entre dispositivos hará que la comunicación sea resistente a los ataques de intermediarios, mientras que el servicio de búsqueda permitirá que los dispositivos se identifiquen y autenticuen entre sí.

El modo fuera de línea permite que dichos dispositivos funcionen de forma autónoma o semiautónomamente e interoperen entre sí, validen y verifiquen las transacciones entre esos dispositivos, e interactúen con los sistemas de un operador solo cuando sea necesario.

El modelo de seguridad contextual que se explica a continuación se extiende a cualquier tipo de dispositivo, como un dispositivo “IoT”. Siempre que un dispositivo esté autorizado para operar, y mientras los servicios de ese dispositivo estén listados en un servicio de búsqueda relevante, cualquier dispositivo puede comunicarse con cualquier otro dispositivo, y cada uno usará la cadena de resumen para permitirle confiar y validar la transacción y la comunicación de datos entre los dispositivos, incluidas las instrucciones para modificar los flujos de trabajo de los dispositivos, actualizar los sistemas de un dispositivo o simplemente para pasar o recopilar datos entre esos sistemas. Cada dispositivo conservará una auditoría completa de sus transacciones.

Seguridad

El sistema Tereon utiliza una serie de modelos de seguridad únicos que superan las fallas y restricciones que existen en los modelos y protocolos de seguridad actuales utilizados en los sistemas de procesamiento de transacciones heredados. Los modelos de seguridad, por ejemplo, eliminan la necesidad de almacenar datos en un dispositivo. Este es un problema importante con los sistemas existentes.

Seguridad USSD

El sistema USSD (datos de servicio suplementario no estructurados) se usa comúnmente como el canal de comunicaciones para numerosos tipos de transacciones, incluidos los pagos hacia y desde teléfonos con funciones. Tereon permite que USSD se utilice de forma segura.

La mayoría de las implementaciones requieren que el usuario ingrese un código USSD o elija una acción de un menú numerado. Una serie de mensajes sin cifrar retroceden y avanzan. Esto conduce a problemas de costo, poca seguridad y poca experiencia del usuario.

En lugar de enviar mensajes como texto de 7 u 8 bits, que es donde surgen los problemas de seguridad, Tereon utiliza USSD y canales de comunicaciones similares de una manera nueva. Tereon simplemente lo ve como un canal de comunicaciones de envíos cortos y basado en sesiones.

Tereon no adapta un mensaje para que se ajuste a USSD, que es lo que hacen los sistemas existentes. En cambio, para cada comunicación cifrada en una sesión de transacción, Tereon cifra la comunicación como lo haría para una comunicación a través de TCP/IP (es decir, GPRS, 3G, 4G, WiFi, etc.) para generar un texto cifrado, y luego codifica el texto cifrado como una cadena de caracteres base de 7 bits.

Tereon luego verifica la longitud del texto cifrado. Si es más largo que el espacio permitido en los mensajes de USSD, corta el texto cifrado en dos o más partes y las transmite individualmente utilizando USSD. En el otro extremo, Tereon vuelve a ensamblar las partes en toda la cadena de caracteres, las convierte de nuevo al texto encriptado y luego las descifra.

Tereon puede usar este método para usar primero TLS (seguridad de la capa de transporte) para identificar y autenticar a las partes. Esto generará la primera clave de sesión. Luego, Tereon puede usar esta clave de sesión para cifrar la negociación del protocolo PAKE que genera la segunda clave de sesión que las partes usarán para cifrar todas las comunicaciones posteriores en la sesión.

Algunos teléfonos con funciones son compatibles con WAP (protocolo de aplicación inalámbrica). Donde estas implementaciones usan WAP sobre USSD, entonces Tereon simplemente usará la pila de protocolos WAP como una forma de comunicación a través de USSD. Esto proporcionará la capa de seguridad de la capa de transporte inalámbrico (WTLS), que simplemente actuará como un nivel adicional de autenticación (es más débil que el cifrado TLS y el estándar de cifrado avanzado 256 (AES256) que Tereon utiliza como predeterminado, por lo que Tereon usará AES256 para cifrar las comunicaciones en cualquier caso).

También es así como Tereon puede asegurar otros canales de comunicaciones que se perciben como que carecen de seguridad (por ejemplo, NFC, Bluetooth, etc.). Al construir una sesión de mensajería cuidadosamente, la naturaleza de USSD y otros canales "no seguros" se puede cambiar por completo.

Modelo de seguridad para dispositivos activos (y la Internet de las Cosas).

El modelo de seguridad para dispositivos activos, como teléfonos móviles, terminales de tarjetas, etc., funciona de manera similar al modelo de seguridad para tarjetas (ver más abajo). La tarjeta SIM no se utiliza, ya que los algoritmos de seguridad se descifraron hace algún tiempo. En su lugar, se utiliza una clave de registro, que se cifra y almacena en el dispositivo, junto con una clave única que genera la red. En los dispositivos móviles, Tereon puede usar esa

clave para realizar una búsqueda para verificar que el IMSI (identidad de abonado móvil internacional) reportado por el móvil sea genuino.

Cuando un usuario ejecuta una aplicación por primera vez (los usuarios pueden tener múltiples aplicaciones si lo desean), la aplicación solicitará un código de autenticación de una sola vez que el servidor de Tereon genera para la cuenta del usuario, junto con el número de teléfono móvil o el número de serie del dispositivo (si la aplicación no puede determinar primero ese número). El usuario también puede registrar su aplicación con múltiples servidores de Tereon, donde cada servidor generará un código de activación único para cada cuenta o servicio que el servidor opera para el usuario.

Una vez que el usuario ingresa el código de activación de una sola vez, la aplicación usa ese código como el secreto compartido entre él y el servidor para generar la primera sesión PAKE (después de que la aplicación y el servidor de Tereon se hayan validado entre sí mediante TLS o un protocolo similar, si es necesario). Una vez que hayan establecido la primera sesión de PAKE, el servidor de Tereon enviará a la aplicación una clave de registro cifrada y firmada, junto con un nuevo secreto compartido. Tanto el servidor como la aplicación utilizarán el código de activación de una sola vez, la clave de registro y el secreto compartido para generar un nuevo secreto compartido al crear un resumen de los tres.

Cada vez que el servidor y la aplicación se comunican, crearán un secreto compartido mediante la función de resumen del secreto compartido anterior con un resumen de los mensajes anteriores que se comunicaron entre ellos en las comunicaciones en línea. Cada vez que la aplicación y el servidor se comunican entre sí, generarán un resumen de los contenidos de la transacción, el resumen de transacción, que han intercambiado con el resumen de los intercambios anteriores. Ambos usan este resumen de transacción para generar el nuevo secreto compartido.

Si un usuario pierde su dispositivo, o si necesita volver a registrar una aplicación o cambiar de dispositivo, el servidor Tereon generará un nuevo código de autenticación y clave de registro de

una sola vez. El nuevo secreto compartido que el servidor pasará a la aplicación se generará a partir del resumen de los mensajes anteriores intercambiados entre ese servidor y la aplicación.

Este reenvío de claves permite que la aplicación y el servidor Tereon siempre tengan un secreto nuevo y compartido para cada sesión de PAKE. Por lo tanto, si un atacante pudiera romper la sesión TLS (lo que sería extremadamente difícil ya que tanto el servidor como la aplicación firmarían sus mensajes), el atacante todavía tendría que romper la clave de sesión PAKE subyacente. Si la parte logró esa hazaña, eso le daría a la parte la clave para esa sesión y solo para esa sesión. El proceso de generar una nueva clave para cada comunicación significa que la parte tendría que repetir esa hazaña para cada comunicación, una tarea que es virtualmente imposible computacionalmente.

Debido a que la aplicación se autentica contra un servicio en particular en cualquier sesión, la aplicación del usuario interactuará solo con ese servicio. El servidor no tendrá conocimiento de ninguno de los otros servicios en los que está registrada la aplicación del usuario. En efecto, las aplicaciones se convierten en algo parecido al "papel psíquico", un dispositivo de identificación que proporciona solo las credenciales requeridas por un servicio, independientemente de la pluralidad de servicios para los cuales el usuario podría estar registrado. Puede parecerse a un dispositivo de pago a un servicio, un boleto de transporte a otro, una llave de la puerta a otro, etc. Los proveedores de servicios no necesitan emitir dispositivos separados para acceder a sus servicios y, como tal, esto reduce tanto la complejidad como el costo de ofrecer servicios y la actualización de dichos servicios.

El modelo de seguridad tiene un beneficio adicional. Si un usuario pierde su dispositivo, puede obtener un nuevo dispositivo con exactamente el mismo número. El dispositivo antiguo con sus aplicaciones no funcionará, mientras que el nuevo dispositivo, una vez que se haya registrado, funcionará, ya que tendrá la clave secreta y el código de registro que son válidos.

Aunque puede haber un intervalo de tiempo entre la pérdida y la notificación de un dispositivo perdido, nadie podrá realizar ninguna transacción, ya que nadie tendrá la contraseña y el PIN necesarios, ni ningún otro "permiso" (token) de autenticación.

El usuario, o el administrador del sistema de Tereon, también puede configurar la aplicación para que requiera una contraseña antes de que el usuario pueda acceder a la aplicación. Esta contraseña se verifica con el servidor de Tereon. Si es válida, entonces el servidor Tereon le indicará a la aplicación que opere (con una comunicación que siempre esté firmada y encriptada). Si la contraseña no es válida, entonces el servidor de Tereon indicará a la aplicación que solicite una nueva contraseña para un número limitado de intentos. A partir de entonces, el servidor de Tereon bloqueará la aplicación del usuario y el usuario deberá comunicarse con el administrador para desbloquear la aplicación y volver a registrar el dispositivo.

Cada credencial es cronometrada. Eso significa que un usuario puede tener una credencial particular asignada a él o ella durante un período de tiempo definido, y todas las transacciones que tienen lugar con esa credencial durante ese período de tiempo están vinculadas a ese usuario. Si ese usuario cambia de credencial, entonces la credencial original se puede asignar a otro usuario. Sin embargo, el servidor de búsqueda continuará vinculando las transacciones y las credenciales en función de la combinación de las credenciales y los periodos de tiempo registrados en esas credenciales.

El mismo modelo se puede adaptar para asegurar las comunicaciones entre dispositivos en la “Internet de las Cosas”. Aquí se puede utilizar un certificado o un número de serie cableado para identificar cada dispositivo. Ese se convertirá en el primer secreto compartido que intercambiará cada dispositivo en el primer contacto, cuando se marque con la fecha de la transacción o con los mensajes anteriores enviados entre los dispositivos. Se utilizarían dos números, un número de serie abierto que identificaría el dispositivo y que actuaría en lugar de un certificado PKI (infraestructura de clave pública), y un número de serie protegido criptográficamente que actuaría como el secreto compartido. Alternativamente, se podría usar un solo número de serie como la ID y el primer secreto compartido, y se cargaría una nueva clave secreta a través del canal de comunicaciones seguro (ver descripción sobre las capas de comunicación en la arquitectura de los sistemas).

El modelo de seguridad móvil de Tereon tiene otra ventaja. Un operador puede usarlo para establecer derechos de acceso a servicios individuales y configurar el nivel de acceso según el dispositivo y la red a través de los cuales un uso en particular está intentando tener éxito en ese servicio. Esto significa, por ejemplo, que un proveedor puede especificar que un administrador pueda ver los registros del sistema a través de una red pública segura, pero solo acceder a las funciones de administración del sistema a través de una red interna, y luego solo a través de una red fija, a diferencia de un dispositivo móvil.

Aunque esta capacidad tiene alguna aplicación en los pagos (asegura el acceso a las funciones de administración del sistema a redes y dispositivos definidos), se aplica a otros servicios donde se requiere acceso limitado a contenido sensible o privilegiado, de modo que los usuarios puedan controlar exactamente quién puede ver ciertos datos, qué datos pueden ver estos terceros y desde qué ubicación pueden hacerlo.

El modelo de seguridad permite que una organización pueda garantizar la privacidad y seguridad de cualquier información recopilada, generada o transmitida por cualquier dispositivo. Esto se puede aplicar a cualquier dispositivo o transacción, desde un pago hasta un dispositivo médico, un sensor de tráfico, un sensor meteorológico, un detector de flujo de agua, etc.

Modelo de seguridad con tarjeta

Las tarjetas EMV y los teléfonos móviles que utilizan la emulación de la tarjeta del huésped almacenan un PIN en el chip o en un elemento seguro en el teléfono. Las tarjetas sin contacto, y los dispositivos móviles que emulan esas tarjetas, también almacenan la mayoría de los detalles de la tarjeta en forma clara o en un formato fácil de leer. Los terminales de la tarjeta comprueban el PIN que el usuario ingresa contra el PIN almacenado en la tarjeta. Aquí es donde muchas de las debilidades del sistema EMV salen a la luz y hacen que el proceso EMV esté abierto a una serie de ataques bien documentados.

Tereon almacena solo una clave de autenticación en la tarjeta y verifica el valor ingresado contra un valor almacenado en el servicio de Tereon (en un área segura de la base de datos cerrada a los administradores que solo ven que los valores no coinciden con el valor real). Se autentica contra el servicio y la función, el recurso, la instalación o el tipo de transacción en particular, u otro tipo de servicio proporcionado por ese servicio. Tereon usa dos modelos de seguridad, uno de los cuales es un subconjunto del otro.

La mayoría de las tarjetas mostrarán un PAN (el número largo). Tereon no usa este número para identificar la cuenta. Más bien, utiliza el PAN de la misma manera que un número móvil; es simplemente una credencial de acceso. Cada tarjeta tiene un PAN encriptado. La tarjeta también tiene una clave de registro cifrada que identifica la tarjeta como válida para cada servicio en donde está registrada, de la misma manera que la clave de registro en un dispositivo móvil autentica ese dispositivo. El código cifrado tendrá un prefijo que simplemente apunta al servicio de directorio de búsqueda de país que el servicio Tereon del comerciante deberá solicitar, si aún no tiene los detalles de la dirección relacionados con esa cadena PAN cifrada registrada en su servicio Tereon.

Cuando el usuario presenta la tarjeta al terminal, el terminal leerá el PAN encriptado, y usará eso y la clave de registro encriptada para validar la tarjeta con el terminal registrado de la tarjeta. Una vez que el servicio Tereon del usuario haya validado y autenticado tanto la tarjeta como el servicio Tereon del comerciante, el servicio del usuario enviará al PAN del servicio Tereon del comerciante, en su forma no encriptada para que pueda registrar esto, junto con la forma encriptada, en su cache. Por lo tanto, si el usuario ingresa posteriormente al PAN en forma transparente, por ejemplo, a través de un portal de comercio electrónico o un terminal de un comerciante, el servicio sabrá qué otro servicio contactar.

Si el lector de tarjetas no puede leer la tarjeta por alguna razón, entonces el usuario o comerciante puede ingresar el PAN y el servicio Tereon del comerciante usará ese PAN para obtener la dirección del servicio Tereon del usuario. El usuario también podría ingresar su dirección de correo electrónico, número de teléfono móvil o cualquier otra credencial única

siempre que esa credencial se registre en la cuenta de ese usuario. El PAN de la tarjeta es simplemente una de las muchas credenciales que el usuario puede usar.

Una vez que el servicio Tereon del comerciante haya validado la tarjeta, el terminal del comerciante configurará un TLS y luego una sesión PAKE con su servicio Tereon, utilizando su clave resumen para hacerlo (cada vez que el terminal se comunica con su servicio, este agrupa o resume su clave anterior con su clave de registro para generar el nuevo secreto compartido para la sesión PAKE). El proceso del comerciante continuará hasta que el terminal del comerciante necesite solicitar un PIN (si el servicio Tereon del usuario requiere un PIN para esa transacción, según lo determine el proveedor del servicio de pago y se incluya en el motor de reglas comerciales del servicio Tereon). El servicio Tereon del usuario generará una sesión PAKE con el servicio del comerciante y luego enviará una clave única al servicio del comerciante y un mensaje cifrado al terminal a través de otra sesión PAKE creada con TLS primero.

El terminal del comerciante recibirá la clave y descifrá el mensaje para mostrar un texto seleccionado por el usuario que muestra que el terminal está autorizado por el servicio del comerciante. El usuario ingresa su PIN, que se comunica a través de la sesión PAKE del terminal con el servicio del usuario. Este proceso solo ocurre cuando el usuario tiene que ingresar su PIN en un terminal comercial. El terminal del comerciante nunca ve el PIN a la vista, ya que se ingresa en una aplicación segura a la que el terminal del comerciante accede desde el servicio Tereon del usuario y se cifra con una segunda clave única que el servicio del usuario transmite al terminal de forma de intercambio de claves seguro y firmado. Normalmente, todas las comunicaciones se realizarán a través del servicio del comerciante, las comunicaciones directas entre el terminal y el servicio Tereon del usuario también se pueden establecer donde el terminal puede admitir esa funcionalidad.

Si la tarjeta es una tarjeta de microprocesador (Chip y PIN, sin contacto, o ambos), entonces la tarjeta también puede tener un secreto compartido que se generó inicialmente cuando se emitió.

Una tarjeta de microprocesador también usaría PAKE para establecer una sesión con su Servicio Tereon registrado (o el servicio para el servicio). Esta sesión sería junto con la sesión establecida por el terminal de la tarjeta (que podría ser una tableta móvil o un terminal de la tarjeta PoS) con su servicio Tereon. Esto elimina de inmediato la vulnerabilidad clave que exhiben los terminales existentes y las tarjetas Chip y PIN, que es la vulnerabilidad de la infraestructura existente para interferir y subvertir el proceso de verificación de PIN a través de una serie de ataques por parte de "personas metidas en el medio" o "cuñas".

La tarjeta utilizará este canal para generar una clave que transmitirá a su servicio y que, a su vez, su servicio transmitirá al terminal del comerciante para cifrar el PIN. También utilizará este canal para facilitar las transacciones fuera de línea, cuando la tarjeta almacenará el balance de la última transacción en línea, la clave que usará como inicio para generar la serie de claves que usará para las transacciones fuera de línea y los registros de varias transacciones fuera de línea de terceros.

Si una tarjeta se pierde o es robada, el modelo de seguridad de Tereon significa que el emisor no necesita emitir un nuevo PAN.

Seguridad basada en contexto

La mayoría de los protocolos de seguridad utilizan algunas credenciales y se basan en suposiciones subyacentes. Son estas suposiciones las que pueden conducir a errores y, por lo tanto, a una pérdida de seguridad. El sistema Tereon no se basa en ninguna suposición subyacente que no sea la suposición de que la red de comunicaciones, sin este sistema, puede ser insegura y no puede ser confiable, y que el entorno en donde opera un dispositivo también puede ser inseguro.

El sistema de Tereon va varias etapas más allá y analiza tanto un conjunto de credenciales como el contexto dentro del cual se presentan esas credenciales. Esto proporciona seguridad adicional y asegura uno de los medios por los cuales las organizaciones pueden permitir que sus

empleados o miembros utilicen sus propios dispositivos (a veces denominados BYOD) en algunas o todas las circunstancias.

Tereon no solo puede usar las contraseñas, PIN u otras credenciales de autenticación directa del usuario; también utilizará los detalles del dispositivo, la aplicación en ese dispositivo, la red a través de la cual el dispositivo accede a Tereon, la ubicación geográfica de ese dispositivo en el momento y durante la sesión, y el servicio o la información que el usuario está accediendo con ese dispositivo.

Tereon toma las credenciales y, en función del contexto establecido por y contra esas credenciales, controlará el acceso a la información, otorgando un nivel de acceso adecuado a la credencial.

Por ejemplo, un administrador que intente acceder a los servicios de administración profunda en un dispositivo privado que no haya sido aprobado por Tereon será bloqueado de esos servicios, independientemente de si ese administrador está en el lugar de trabajo y en la red del lugar de trabajo. Sin embargo, ese mismo administrador puede tener derecho a ver algunos de los registros del sistema en ese mismo dispositivo.

Un segundo ejemplo sería cuando el modelo de seguridad de contexto gobierna los servicios que un usuario secundario puede ver. Un usuario tiene un teléfono o tarjeta que proporciona múltiples funciones, como depósitos, retiros y pagos sin límites establecidos (por supuesto, hasta cualquier límite de crédito o fondos disponibles). Ese usuario ha frecuentado un café en varias ocasiones y siempre ha comprado un café y un croissant de almendras. Hoy, el usuario le dio su tarjeta a su hijo y estableció un límite de gasto total de £ 40 para la tarjeta. El usuario también ha configurado un segundo PIN para el uso de su hijo, quien lleva la tarjeta al mismo café para comprar un café. El sistema Tereon normalmente habría ofrecido un croissant de almendra gratis al usuario hoy, ya que ya compró 6 en el pasado, y la cafetería utiliza Tereon para impulsar las ofertas a sus clientes. Sin embargo, cuando el hijo del usuario ingresa su PIN, el sistema de Tereon detecta que es el hijo del usuario el que realiza el pago (no conoce el PIN de su padre) y bloquea la oferta de hoy, ya que tiene una alergia a las nueces, y su padre ha

vinculado el PIN de su hijo al perfil de su hijo. El comerciante no ve ningún aviso de la oferta de un croissant gratuito y Tereon sabe que el hijo del usuario no puede comer nueces. Todo lo que el comerciante puede ver es un pago por un café.

El usuario también ha permitido a su hijo retirar dinero en efectivo de hasta £ 10, pero no depositar fondos. Por lo tanto, cuando el hijo del usuario ingresa a un comerciante que puede ofrecer un retiro de hasta £ 10, verá la opción en la terminal del comerciante.

La seguridad basada en el contexto va más allá del control de acceso. Dependiendo del contexto dentro del cual un usuario presenta o usa un dispositivo, ese dispositivo presentará solo las credenciales necesarias para ese contexto; el mismo se convierte en *“papel psíquico”*. De esta manera, el servicio de directorio 216 proporciona una funcionalidad que puede soportar la seguridad basada en el contexto.

La seguridad basada en el contexto elimina la necesidad de credenciales y dispositivos separados para contextos particulares. Ahora, un solo dispositivo puede convertirse en una credencial de tarjeta de biblioteca en una biblioteca, un boleto de transporte en un autobús o tren, una llave segura para acceder a una habitación o instalaciones, un dispositivo de pago interno en la cantina de una empresa, un boleto de teatro, un dispositivo de pago estándar en un supermercado, un permiso de conducir, una tarjeta NHS, una tarjeta de identificación para demostrar el derecho a un servicio, que podría mostrar una identificación con foto en el dispositivo del comerciante si el servicio lo requiere, etc.

Debido a que Tereon proporciona el procesamiento y liquidación de transacciones dinámicas en tiempo real, un administrador o usuario puede modificar, agregar o incluso cancelar un contexto o credencial permitidos en tiempo real. La enmienda se refleja inmediatamente en el servidor de Tereon que proporciona un servicio, en el servicio de directorio de búsqueda 216, o en ambos. Los dispositivos perdidos ya no deben representar un riesgo de exposición financiera o de identificación hasta que los sistemas actuales desactiven el dispositivo. Una vez que un usuario o administrador cancela o modifica una credencial o contexto, ese cambio se activará de inmediato.

Transacción de un toque

Tereon implementa un método de acceso y autorización de transacción con un solo botón que elimina las fallas de seguridad en los sistemas existentes. Por ejemplo, los pagos actuales sin PIN o NFC son extremadamente peligrosos ya que no proporcionan autenticación para un pago. Hasta que el emisor de la tarjeta cancele una credencial de teléfono o tarjeta en el sistema EMV sin contacto, el usuario será responsable de todos los pagos. Incluso si el dispositivo es cancelado por el emisor, el consumidor todavía tiene que intentar probar que no activó el pago. ¿Cómo puede hacerlo si el pago nunca requirió un PIN para autenticarlo? Esto deja un enorme agujero que le permite a cualquiera recoger una tarjeta o un teléfono sin contacto y simplemente tocar y continuar para hacer pagos. Hasta que se cancele, el dispositivo seguirá siendo válido.

Tereon admite la función "pulsar y listo" en uno de los tres modos, cada uno de los cuales depende del contexto para operar. Uno de estos proporciona una transacción de un solo toque que utiliza un enfoque para identificar a una persona. Cuando tanto el usuario como el proveedor de servicios acuerdan que el nivel de autenticación proporcionado es satisfactorio, el sistema proporcionará un método de autenticación de un solo toque mediante el cual el dispositivo mostrará un botón grande o configurará un área grande en la pantalla para que el usuario la toque. Los otros modos son completamente táctiles, como la transacción sin contacto existente en la que el usuario no ingresa las credenciales, y una, donde el usuario ingresa sus credenciales de pago estándar una vez que los dispositivos se han identificado entre sí.

El botón o área en sí proporciona la autenticación a través de la pantalla táctil. Cada individuo presiona una pantalla de una manera única, tanto en términos de dónde presiona ese individuo, como del patrón de presión que utiliza. Si una persona tiene la intención de usar esta función, entonces Tereon le pedirá que presione el botón o el área varias veces hasta que se haya enterado del toque o pulsación a modo de firma de esa persona. La pantalla se divide lógicamente en una cantidad de celdas discretas, y Tereon observará la proximidad y el patrón

de las celdas que el usuario toca durante el período de entrenamiento, y donde sea posible con el patrón de presión y cualquier movimiento del dispositivo que ocurra cuando el usuario presiona la pantalla. Utilizará y supervisará esos datos para crear el perfil que utiliza para autenticar al usuario.

La Fig. 21 ilustra un diagrama de bloques de una implementación de un dispositivo de computación 2100 dentro del cual se puede ejecutar un conjunto de instrucciones, para hacer que el dispositivo de computación realice una o más de las metodologías discutidas en este documento. En implementaciones alternativas, el dispositivo informático puede estar conectado (por ejemplo, en red) a otras máquinas en una red de área local (LAN), una intranet, una extranet o Internet. El dispositivo de computación puede operar en la capacidad de un servidor o una máquina cliente en un entorno de red cliente-servidor, o como una máquina de igual a igual en un entorno de red de igual a igual (o distribuido). El dispositivo informático puede ser una computadora personal (PC), una tableta, un decodificador (STB), un asistente digital personal (PDA), un teléfono celular, un dispositivo web, un servidor, un enrutador de red, un conmutador o Bridge, un procesador o cualquier máquina capaz de ejecutar un conjunto de instrucciones (secuenciales o de otro tipo) que especifican las acciones que debe realizar esa máquina. Además, aunque solo se ilustra un solo dispositivo informático, el término "dispositivo informático" también debe incluir cualquier conjunto de máquinas (p. ej., computadoras) que ejecuten individual o conjuntamente un conjunto (o varios conjuntos) de instrucciones para realizar cualquiera o más de las metodologías aquí descritas.

El dispositivo informático 2100 a modo de ejemplo incluye un dispositivo de procesamiento 2102, una memoria principal 2104 (por ejemplo, memoria de solo lectura (ROM), memoria flash, memoria de acceso aleatorio dinámico (DRAM) como DRAM sincrónica (SDRAM) o DRAM Rambus (RDRAM), etc.), una memoria estática 2106 (por ejemplo, memoria flash, memoria de acceso aleatorio estática (SRAM), etc.) y una memoria secundaria (por ejemplo, un dispositivo de almacenamiento de datos 2118), que se comunican entre sí a través de un bus 2130.

El dispositivo de procesamiento 2102 representa uno o más procesadores de propósito general tales como un microprocesador, unidad central de procesamiento, o similares. Más

particularmente, el dispositivo de procesamiento 2102 puede ser un microprocesador de computación de conjunto de instrucciones (CISC), microprocesador de computación de conjunto de instrucciones reducido (RISC), microprocesador de palabra de instrucción muy larga (VLIW), procesador que implementa otros conjuntos de instrucción o procesadores que implementan una combinación de conjuntos de instrucción. El dispositivo de procesamiento 2102 también puede ser uno o más dispositivos de procesamiento de propósito especial, tales como un circuito integrado específico de aplicación (ASIC), una matriz de compuerta programable de campo (FPGA), un procesador de señales digitales (DSP), un procesador de red, o similares. El dispositivo de procesamiento 2102 está configurado para ejecutar la lógica de procesamiento (instrucciones 2122) para realizar las operaciones y los pasos descritos en la presente solicitud.

El dispositivo informático 2100 puede incluir además un dispositivo de interfaz de red 2108. El dispositivo informático 2100 también puede incluir una unidad de visualización de video 2110 (por ejemplo, una pantalla de cristal líquido (LCD) o un tubo de rayos catódicos (CRT)), un dispositivo de entrada alfanumérico 2112 (por ejemplo, un teclado o una pantalla táctil), un dispositivo de control tipo cursor 2114 (por ejemplo, un mouse o pantalla táctil), y un dispositivo de audio 2116 (por ejemplo, un altavoz).

El dispositivo de almacenamiento de datos 2118 puede incluir uno o más medios de almacenamiento legibles por máquina (o, más específicamente, uno o más medios de almacenamiento legibles por computadora no transitorios) 2128 en donde se almacenan uno o más conjuntos de instrucciones 2122 que incorporan uno o más de Las metodologías o funciones aquí descritas. Las instrucciones 2122 también pueden residir, total o al menos parcialmente, dentro de la memoria principal 2104 y/o dentro del dispositivo de procesamiento 2102 durante su ejecución por el sistema informático 2100, en donde la memoria principal 2104 y el dispositivo de procesamiento 2102 también constituyen un medio de almacenamiento legible por computadora.

Los diversos métodos descritos anteriormente pueden implementarse mediante un programa informático. El programa de computadora puede incluir un código de computadora dispuesto

para instruir a una computadora para que realice las funciones de uno o más de los varios métodos descritos anteriormente. El programa informático y/o el código para realizar tales métodos pueden proporcionarse a un aparato, tal como una computadora, en uno o más medios legibles por computadora o, más generalmente, un producto de programa informático. Los medios legibles por computadora pueden ser transitorios o no transitorios. El uno o más medios legibles por computadora podrían ser, por ejemplo, un sistema electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor, o un medio de propagación para la transmisión de datos, por ejemplo, para descargar el código a través de Internet. Alternativamente, el uno o más medios legibles por computadora podrían tomar la forma de uno o más medios físicos legibles por computadora tales como un semiconductor o una memoria de estado sólido, una cinta magnética, un disquete de computadora extraíble, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), un disco magnético rígido y un disco óptico, como un CD-ROM, CD-R/W o DVD.

En una implementación, los módulos, componentes y otras características descritas en este documento pueden implementarse como componentes discretos o integrarse en la funcionalidad de componentes de hardware como ASICs, FPGA, DSP o dispositivos similares como parte de un servidor de individualización.

Un "componente de hardware" es un componente físico tangible (por ejemplo, no transitorio) (por ejemplo, un conjunto de uno o más procesadores) capaz de realizar ciertas operaciones y puede configurarse o disponerse de cierta manera física. Un componente de hardware puede incluir circuitos o lógica dedicados y que están configurados permanentemente para realizar ciertas operaciones. Un componente de hardware puede ser o incluir un procesador de propósito especial, como una matriz de compuerta programable de campo (FPGA) o un ASIC. Un componente de hardware también puede incluir un circuito o lógica programable que está temporalmente configurado por el software para realizar ciertas operaciones.

En consecuencia, debe entenderse que la frase "componente de hardware" abarca una entidad tangible que puede construirse físicamente, configurarse permanentemente (por ejemplo,

cablearse) o configurarse temporalmente (por ejemplo, programar) para operar de cierta manera o para realizar ciertas operaciones aquí descritas.

Una máquina puede ser, por ejemplo, una máquina física, una máquina lógica, una máquina virtual, un contenedor o cualquier otro mecanismo comúnmente utilizado para contener código ejecutable. Una máquina puede ser una sola máquina o puede referirse a una pluralidad de máquinas conectadas o distribuidas, independientemente de si esas máquinas son del mismo tipo o de una pluralidad de tipos de máquinas.

Además, los módulos y componentes pueden implementarse como firmware o circuitos funcionales dentro de dispositivos de hardware. Además, los módulos y componentes pueden implementarse en cualquier combinación de dispositivos de hardware y componentes de software, o solo en software (por ejemplo, código almacenado o incorporado de otro modo en un medio legible por máquina o en un medio de transmisión).

A menos que se especifique lo contrario, como se desprende de la siguiente discusión, se aprecia que a lo largo de la descripción, las descripciones utilizan términos como "enviar", "recibir", "determinar", "comparar", "habilitar", "mantener", "identificar", o similares, las cuales se refieren a las acciones y procesos de un sistema informático, o dispositivo informático electrónico similar, que manipula y transforma datos, representados como cantidades físicas (electrónicas) dentro de los registros y memorias del sistema informático, en otros datos representados de manera similar como cantidades físicas dentro de las memorias o registros del sistema informático u otros dispositivos de almacenamiento, transmisión o visualización de información similares.

Debe entenderse que la descripción anterior pretende ser ilustrativa y no restrictiva. Muchas otras implementaciones serán evidentes para los expertos en la técnica al leer y comprender la descripción anterior. Aunque la presente divulgación se ha descrito con referencia a implementaciones de ejemplo específicas, se reconocerá que la divulgación no se limita a las implementaciones descritas, sino que se puede llevar a la práctica con modificaciones y alteraciones dentro del espíritu y alcance de las reivindicaciones adjuntas. En consecuencia, la

descripción y los dibujos deben considerarse en un sentido ilustrativo en lugar de un sentido restrictivo. Por lo tanto, el alcance de la divulgación debe determinarse con referencia a las reivindicaciones adjuntas, junto con el alcance completo de los equivalentes a los que tienen derecho dichas reivindicaciones.

Todas las características opcionales de los diversos aspectos se relacionan con todos los demás aspectos *mutatis mutandis*. Se contemplan variaciones de las realizaciones descritas, por ejemplo, las características de todas las realizaciones descritas pueden combinarse de cualquier manera.