



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112015004429-8 B1



(22) Data do Depósito: 06/09/2013

(45) Data de Concessão: 11/10/2022

(54) Título: MÉTODO PARA COMPARTILHAR DADOS EM UM SISTEMA DE SEGURANÇA, SISTEMA DE SEGURANÇA FÍSICO E MEIO LEGÍVEL EM COMPUTADOR NÃO TRANSITÓRIO

(51) Int.Cl.: H04L 12/28; G06F 21/62.

(52) CPC: H04L 12/2825; G06F 21/6272.

(30) Prioridade Unionista: 07/09/2012 US 13/607,447.

(73) Titular(es): AVIGILON CORPORATION.

(72) Inventor(es): RYAN LEE; SHAUN MARLATT; MATTHEW ADAM; ROSS WIGHTMAN; GREG MAGOLAN; ANDREW MARTZ.

(86) Pedido PCT: PCT CA2013050690 de 06/09/2013

(87) Publicação PCT: WO 2014/036656 de 13/03/2014

(85) Data do Início da Fase Nacional: 27/02/2015

(57) Resumo: SISTEMA DE SEGURANÇA FÍSICO QUE TEM MÚLTIPLOS NÓS DE SERVIDOR. A presente invenção refere-se a um sistema de segurança físico que tem múltiplos nós de servidor que podem ser construídos como uma rede distribuída. Para enviar dados entre os nós na rede, um primeiro nó pode acessar um identificador de nó que identifica um segundo nó, tanto com o primeiro quanto com o segundo nós que formam pelo menos parte de um cluster de servidor, e o primeiro nó pode então enviar os dados ao segundo nó. O identificador de nó forma pelo menos parte de informações de associação de cluster que identifica todos e é acessível por todos os nós de servidor nesse cluster de servidor. Funcionalidade tal como a capacidade de compartilhar visualizações entre usuários de sistema e a capacidade desses usuários de controlar um visor autônomo podem ser implantadas em uma rede distribuída, uma rede federada, ou outro tipo de rede.

Relatório Descritivo da Patente de Invenção para "**MÉTODO PARA COMPARTILHAR DADOS EM UM SISTEMA DE SEGURANÇA, SISTEMA DE SEGURANÇA FÍSICO E MEIO LEGÍVEL EM COMPUTADOR NÃO TRANSITÓRIO**".

REFERÊNCIA CRUZADA A PEDIDOS RELACIONADOS

[001] Este pedido é uma continuação-em-parte de Pedido de Patente Cedido com o nº de Série U.S. 13/607.447 e depositado em 7 de setembro de 2012, cujo conteúdo integral é incorporado a título de referência no presente documento.

CAMPO TÉCNICO

[002] A presente revelação é direcionada a um sistema de segurança físico que tem múltiplos nós de servidor.

ANTECEDENTES

[003] Um sistema de segurança físico é um sistema que implanta medidas para impedir que pessoas não autorizadas ganhem acesso físico a um ativo, como um edifício, uma instalação, ou informações confidenciais. Exemplos de sistemas de segurança física incluem sistemas de vigilância, como um sistema no qual câmeras são usadas para monitorar o ativo e aquelas próximas do mesmo; sistemas de controle de acesso, como um sistema que usa cartões de RFID para controlar acesso a um edifício; sistemas de detecção de intrusão, como um sistema de alarme de roubo doméstico; e combinações dos sistemas anteriores.

[004] Um sistema de segurança físico frequentemente incorpora computadores. À medida que esse tipo de sistema de segurança físico cresce, a potência de computação exigida para operar o sistema aumenta. Por exemplo, à medida que o número de câmeras e um sistema de vigilância aumentam, a quantidade exigida de potência de computação também aumenta para permitir que vídeo adicional seja armazenado e para permitir gerenciamento e uso simultâneo de um número maior de câmeras. Pesquisa e desenvolvimento consequentemente continuam a

superar problemas encontrados à medida que um sistema de segurança físico cresce.

SUMÁRIO

[005] De acordo com um primeiro aspecto, é fornecido um método para compartilhar dados em um sistema de segurança físico que compreende uma pluralidade de nós de servidor. O método compreende acessar, com o uso de um dos nós de servidor ("primeiro nó"), um identificador de nó que identifica outro dentre os nós de servidor ("segundo nó"), em que os primeiro e segundo nós compreendem pelo menos parte de um cluster de servidor e em que o identificador de nó compreende pelo menos parte de informações de associação de cluster que identifica todos e acessível por todos os nós de servidor no cluster de servidor; e enviar os dados do primeiro nó ao segundo nó.

[006] O cluster de servidor pode compreender pelo menos três nós de servidor.

[007] Os nós de servidor podem compreender câmeras, gravadores de vídeo em rede e servidores de controle de acesso.

[008] O método pode compreender ainda acessar, com o uso do segundo nó, um identificador de nó que identifica o primeiro nó; e enviar dados adicionais do segundo nó ao primeiro nó.

[009] As informações de associação de cluster podem compreender um identificador de nó que identifica exclusivamente cada um dos nós de servidor no cluster de servidor; e um identificador de cluster que identifica exclusivamente o cluster de servidor, ao qual os nós de servidor pertencem.

[0010] Cada um dos nós de servidor no cluster de servidor pode armazenar de modo persistente sua própria versão das informações de associação de cluster de modo localizado.

[0011] O método pode compreender ainda reinicializar um dos nós

de servidor ("nó de servidor reinicializado") no cluster de servidor; e uma vez que o nó de servidor reinicializado retorna online, usar o nó de servidor reinicializado para executar um método que compreende (i) acessar o identificador de cluster que identifica o cluster de servidor; e (ii) reunir automaticamente o cluster de servidor.

[0012] O método pode compreender ainda adicionar um novo nó de servidor ao cluster de servidor executando-se um método que compreende trocar uma versão das informações de associação de cluster armazenado no novo nó de servidor pela versão das informações de associação de cluster armazenado em um dos nós de servidor que já é parte do cluster de servidor ("nó de controle de associação"); e sincronizar as versões das informações de associação de cluster armazenado no novo nó de servidor com as versões das informações de associação de cluster armazenado em todos os nós de servidor no cluster antes do novo nó de servidor ingressar ao cluster.

[0013] Enviar os dados pode compreender pesquisar, com o uso do primeiro nó, um ponto de extremidade para o segundo nó do identificador de nó; e enviar os dados do primeiro nó ao ponto de extremidade.

[0014] O ponto de extremidade e o identificador de nó podem compreender entradas em um mapa de rede que relaciona identificador de nós para todos os nós de servidor no cluster de servidor a pontos de extremidade correspondentes, e cada um dos nós de servidor no cluster de servidor pode armazenar de modo persistente sua própria versão do mapa de rede de modo localizado.

[0015] O mapa de rede pode permitir que cada um dentre os nós de servidor no cluster de servidor envie os dados a qualquer outro dentre os nós de servidor no cluster de servidor sem usar um servidor centralizado.

[0016] Os dados podem ser armazenados de modo localizado no primeiro nó e o método pode compreender ainda modificar os dados com o uso do primeiro nó, em que enviar os dados do primeiro nó ao

segundo nó comprehende parte de sincronizar os dados nos primeiro e segundo nós após o primeiro nó ter modificado os dados.

[0017] Os dados podem comprehender informações de versão geradas com o uso de um mecanismo de controle de versão de casualidade e diferentes versões dos dados podem ser armazenadas nos primeiro e segundo nós, e sincronizar os dados pode comprehender comparar as informações de versão armazenadas nos primeiro e segundo nós e adotar tanto no primeiro quanto no segundo nós os dados cujas informações de versão indicam serem as mais recentes.

[0018] Os dados podem comprehender o identificador de nó do primeiro nó, informações de estado de batimento cardíaco do primeiro nó, informações de estado de aplicação do primeiro nó, e informações de versão, e enviar os dados pode comprehender disseminar os dados a todos os nós de servidor no cluster de servidor com o uso de um protocolo gossip que executa trocas de dados entre pares dos nós de servidor no cluster.

[0019] Os dados podem ser periodicamente disseminados a todos os nós de servidor no cluster de servidor.

[0020] Os dados podem ser enviados ao segundo nó quando o primeiro nó ingressa ao cluster.

[0021] Um domínio preenchido com entradas que pode ser modificado por qualquer um dos nós de servidor no cluster de servidor pode ser armazenado de modo localizado em cada um dos nós no cluster de servidor, e o método pode comprehender ainda gerar as informações de versão com o uso de um mecanismo de controle de versão de casualidade de modo que as informações de versão indiquem qual dos nós de servidor modificou mais recentemente uma das entradas.

[0022] As informações de estado de aplicação podem comprehender um hash de nível de topo gerado realizando-se hashing de todas as entradas no domínio.

[0023] O método pode compreender ainda comparar, com o uso do segundo nó, o hash de nível de topo com um hash de nível de topo geradas realizando-se hashing de uma versão de um domínio correspondente armazenado de modo localizado no segundo nó; e se os hashes de nível de topo diferem, sincronizar os domínios em ambos os primeiro e segundo nós com o uso das informações de versão.

[0024] Uma entrada de situação que pode ser modificada apenas pelo primeiro nó pode ser armazenada de modo localizado no primeiro nó, e as informações de versão podem compreender um número de versão que o primeiro nó incrementa sempre que o mesmo modifica a entrada de situação.

[0025] As informações de estado de aplicação podem compreender um par de entidades de situação que compreende um identificador de entidade de situação que identifica a entrada de situação e o número de versão.

[0026] O método pode compreender ainda comparar, com o uso do segundo nó, o número de versão recebido do primeiro nó com um número de versão de uma entrada de situação correspondente armazenado de modo localizado no segundo nó; e se os números de versões diferem, atualizar a entrada de situação armazenada de modo localizado no segundo nó com a entrada de situação armazenada de modo localizado no primeiro nó.

[0027] Atualizar a entrada de situação pode compreender enviar do primeiro nó ao segundo nó entradas de situação adicional armazenadas de modo localizado no primeiro nó que foram modificadas simultaneamente com a entrada de situação.

[0028] Os primeiro e segundo nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o primeiro nó pode enviar os dados de maneira totalmente ordenada a todos os nós de servidor no grupo, e enviar os dados pode compreender o primeiro

nó enviar os dados a todos os nós de servidor no grupo.

[0029] Os dados podem compreender dados não persistentes gerados durante o tempo de execução do sistema de segurança física.

[0030] Os dados também podem compreender o vídeo de transmissão por streaming transmitido por streaming de outro dentre os nós de servidor no cluster de servidor através do primeiro nó ao segundo nó.

[0031] De acordo com outro aspecto, é fornecido um sistema para compartilhar dados em um sistema de segurança física, em que o sistema compreende uma pluralidade de nós de servidor que compreende um primeiro nó e um segundo nó, em que o primeiro nó compreende um processador acoplado de modo comunicativo em um meio legível por computador que teve codificado no mesmo as demonstrações e instruções para fazer com que o processador execute um método que compreende acessar um identificador de nó que identifica o segundo nó, em que os primeiro e segundo nós compreendem pelo menos parte de um cluster de servidor e em que o identificador de nó compreende pelo menos parte de informações de associação de cluster que identifica todos e acessível por todos os nós de servidor no cluster de servidor; e enviar os dados ao segundo nó.

[0032] De acordo com outro aspecto, é fornecido um meio legível por computador não transitório que tem codificado no mesmo as demonstrações e instruções para fazer com que um processador execute um método para compartilhar dados em um sistema de segurança física que compreende uma pluralidade de nós de servidor, em que o método compreende acessar, com o uso de um dos nós de servidor ("primeiro nó"), um identificador de nó que identifica outro dentre os nós de servidor ("segundo nó"), em que os primeiro e segundo nós compreendem pelo menos parte de um cluster de servidor e em que o identificador de nó compreende pelo menos parte de informações de associação de cluster que identifica todos e acessível por todos os nós de servidor no cluster de

servidor; e enviar os dados do primeiro nó ao segundo nó.

[0033] De acordo com outro aspecto, é fornecido um método para interagir com um visor não assistido em um sistema de segurança física que compreende uma pluralidade de nós de servidor, em que o método compreende enviar, de um dos nós de servidor ("segundo nó") comunicativo com o visor não assistido para outro dentre os nós de servidor ("primeiro nó") que é comunicativo com um visor de cliente, dados de estado de visualização indicativos de uma visualização exibida no visor não assistido; e exibir, no visor de cliente, pelo menos uma porção da visualização exibida no visor não assistido. Em um aspecto, nenhum dos nós de servidor é um servidor de porta centralizada; em um aspecto alternativo, pelo menos um dos nós de servidor é um servidor de porta centralizada.

[0034] O método pode compreender ainda enviar, do primeiro nó ao segundo nó, uma mensagem para mudar a visualização do visor não assistido; e atualizar o visor não assistido de acordo com uma mensagem enviada do primeiro nó ao segundo nó.

[0035] Os primeiros e segundos nós e pelo menos outro dentre a pluralidade de nós de servidor podem compreender um cluster de servidor, os primeiros e segundos nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o segundo nó pode enviar os dados de estado de visualização de maneira totalmente ordenada a todos os outros nós de servidor no grupo, e enviar os dados de estado de visualização pode compreender o segundo nó enviar os dados a todos os outros nós de servidor no grupo.

[0036] Os primeiros e segundos nós e pelo menos outro dentre a pluralidade de nós de servidor podem compreender um cluster de servidor, os primeiros e segundos nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o primeiro nó pode enviar uma mensagem para mudar o estado do visor não assistido

de maneira totalmente ordenada a todos os outros nós de servidor no grupo, e o primeiro nó pode enviar uma mensagem para mudar o estado do visor não assistido a todos os outros nós de servidor no grupo.

[0037] O método pode compreender ainda enviar, do segundo nó ao primeiro nó, uma notificação de que a visualização do visor não assistido está disponível para ser controlada.

[0038] Enviar a notificação pode compreender disseminar a notificação a todos os nós de servidor no cluster de servidor com o uso de um protocolo gossip que executa trocas de dados entre pares dos nós de servidor no cluster.

[0039] Antes de enviar o estado do visor não assistido para o visor de controle, o método pode compreender acessar, com o uso do segundo nó, um identificador de nó que identifica o primeiro nó, em que os primeiro e segundo nós compreendem pelo menos parte de um cluster de servidor e em que o identificador de nó compreende pelo menos parte de informações de associação de cluster que identifica todos e acessível por todos os nós de servidor no cluster de servidor.

[0040] As informações de associação de cluster podem compreender um identificador de nó que identifica exclusivamente cada um dos nós de servidor no cluster de servidor; e um identificador de cluster que identifica exclusivamente o cluster de servidor, ao qual os nós de servidor pertencem.

[0041] Cada um dos nós de servidor no cluster de servidor pode armazenar de modo persistente sua própria versão das informações de associação de cluster de modo localizado.

[0042] De acordo com outro aspecto, é fornecido um sistema de segurança física, que compreende: um visor de cliente; um visor não assistido; e uma pluralidade de nós de servidor, em que um dos nós de servidor ("primeiro nó") é comunicativo com o visor de cliente e outro dentre os nós de servidor ("segundo nó") é comunicativo com o visor

não assistido, em que o segundo nó é configurado para enviar, ao primeiro nó, os dados de estado de visualização indicativos de uma visualização exibida no segundo visor, e o primeiro nó é configurado para exibir, no visor de cliente, pelo menos uma porção da visualização exibida no segundo visor. Em um aspecto, nenhum dos nós de servidor é um servidor de porta centralizada; em um aspecto alternativo, pelo menos um dos nós de servidor é um servidor de porta centralizada.

[0043] De acordo com outro aspecto, é fornecido um sistema de segurança físico, que compreende: um cliente que tem um visor de cliente; um visor não assistido; e uma pluralidade de nós de servidor, em que um dos nós de servidor ("primeiro nó") é comunicativo com o cliente e outro dentre os nós de servidor ("segundo nó") é comunicativo com o visor não assistido, em que o segundo nó é configurado para enviar ao primeiro nó dados de estado de visualização indicativos de uma visualização exibida no segundo visor e o cliente e primeiro nó são configurados para exibir, no visor de cliente, pelo menos uma porção da visualização exibida no segundo visor. Em um aspecto, nenhum dos nós de servidor é um servidor de porta centralizada; em um aspecto alternativo, pelo menos um dos nós de servidor é um servidor de porta centralizada.

[0044] O visor não assistido pode ser diretamente conectado ao segundo nó ou indiretamente conectado ao segundo nó por meio, por exemplo, de uma estação de trabalho ou cliente não atendido.

[0045] De acordo com outro aspecto, é fornecido um meio legível por computador não transitório que tem codificado no mesmo as demonstrações e instruções para fazer com que um processador execute um método para interagir com um visor não assistido em um sistema de segurança físico que compreende uma pluralidade de nós de servidor, em que o método compreende enviar, de um dos nós de servidor ("segundo nó") comunicativo com o visor não assistido para outro dentre os nós de servidor ("primeiro nó") que é comunicativo com um visor de cliente, os

dados de estado de visualização indicativos de uma visualização exibida no visor não assistido; e exibir, no visor de cliente, pelo menos uma porção da visualização exibida no visor não assistido.

[0046] De acordo com outro aspecto, é fornecido um método para compartilhar uma visualização ("visualização compartilhada") com o uso de um sistema de segurança físico que compreende uma pluralidade de nós de servidor, em que o método compreende: enviar, de um primeiro cliente para um dos nós de servidor ("primeiro nó"), dados de estado de visualização representativos da visualização compartilhada conforme exibido pelo primeiro cliente; enviar os dados de estado de visualização do primeiro nó para um segundo cliente por meio de outro dentre os nós de servidor ("segundo nó"); atualizar uma exibição do segundo cliente com o uso dos dados de estado de visualização para mostrar a visualização compartilhada; em resposta a uma mudança na visualização compartilhada no segundo cliente, enviar dados de estado de visualização atualizados do segundo cliente ao segundo nó, em que os dados de estado de visualização atualizados são representativos da visualização compartilhada conforme exibido pelo segundo cliente; enviar os dados de estado de visualização atualizados do segundo nó ao primeiro cliente por meio do primeiro nó; e atualizar o visor do primeiro cliente para mostrar a visualização compartilhada com o uso dos dados de estado de visualização atualizados. Em um aspecto, nenhum dos nós de servidor é um servidor de porta centralizada; em um aspecto alternativo, pelo menos um dos nós é um servidor de porta centralizada.

[0047] Os primeiro e segundo nós e pelo menos outro dentre a pluralidade de nós de servidor podem compreender um cluster de servidor, os primeiro e segundo nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o primeiro nó pode enviar os dados de estado de visualização de maneira totalmente ordenada a todos os outros nós de servidor no grupo, e enviar os dados

de estado de visualização podem compreender o primeiro nó enviar os dados a todos os outros nós de servidor no grupo.

[0048] Os primeiro e segundo nós e pelo menos outro dentre a pluralidade de nós de servidor podem compreender um cluster de servidor, os primeiro e segundo nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o segundo nó pode enviar os dados de estado de visualização atualizados de maneira totalmente ordenada a todos os outros nós de servidor no grupo, e enviar os dados de estado de visualização atualizados pode compreender o segundo nó enviar os dados de estado de visualização atualizados a todos os outros nós de servidor no grupo.

[0049] Antes de mostrar a visualização compartilhada no visor do segundo cliente, o método pode compreender enviar do primeiro cliente para o segundo cliente por meio dos primeiro e segundo nós uma notificação de que a visualização compartilhada conforme exibida pelo primeiro cliente está disponível para ser compartilhada com o segundo cliente.

[0050] Os primeiro e segundo nós e pelo menos outro dentre a pluralidade de nós de servidor podem compreender um cluster de servidor, os primeiro e segundo nós podem compreender pelo menos parte de um grupo de nós de servidor no cluster ao qual o primeiro nó pode enviar a notificação de maneira totalmente ordenada a todos os outros nós de servidor no grupo, e enviar a notificação pode compreender o primeiro nó enviar a notificação a todos os outros nós de servidor no grupo.

[0051] Antes de o primeiro nó enviar os dados de estado para o segundo cliente por meio do segundo nó, o método pode compreender acessar, com o uso do primeiro nó, um identificador de nó que identifica o segundo nó, em que os primeiro e segundo nós compreendem pelo menos parte de um cluster de servidor e em que o identificador de nó compreende

pelo menos parte de informações de associação de cluster que identifica todos e acessível por todos os nós de servidor no cluster de servidor.

[0052] As informações de associação de cluster podem compreender um identificador de nó que identifica exclusivamente cada um dos nós de servidor no cluster de servidor; e um identificador de cluster que identifica exclusivamente o cluster de servidor, ao qual os nós de servidor pertencem.

[0053] Cada um dos nós de servidor no cluster de servidor pode armazenar de modo persistente sua própria versão das informações de associação de cluster de modo localizado.

[0054] De acordo com outro aspecto, é fornecido um sistema de segurança físico, que compreende um primeiro cliente que tem uma exibição; um segundo cliente que tem uma exibição; e uma pluralidade de nós de servidor, em que um dos nós de servidor ("primeiro nó") é comunicativo com o primeiro visor e outro dentre os nós de servidor ("segundo nó") é comunicativo com o segundo visor, em que os primeiro e segundo clientes e os primeiro e segundo nós são configurados para enviar, do primeiro cliente ao primeiro nó, os dados de estado de visualização representativos de uma visualização compartilhada conforme exibido no visor do primeiro cliente; envia os dados de estado de visualização do primeiro nó para o segundo cliente por meio do segundo nó; atualiza o visor do segundo cliente com o uso dos dados de estado de visualização para mostrar a visualização compartilhada; em resposta a uma mudança na visualização compartilhada no segundo cliente, enviar dados de estado de visualização atualizados do segundo cliente ao segundo nó, em que os dados de estado de visualização atualizados são representativos da visualização compartilhada conforme exibido no visor do segundo cliente; envia os dados de estado de visualização atualizados do segundo nó ao primeiro cliente por meio do primeiro nó; e atualiza o visor do primeiro cliente para mostrar a

visualização compartilhada com o uso dos dados de estado de visualização atualizados. Em um aspecto, nenhum dos nós de servidor é um servidor de porta centralizada; em um aspecto alternativo, pelo menos um dos nós de servidor é um servidor de porta.

[0055] De acordo com outro aspecto, é fornecido um meio legível por computador não transitório que tem codificado no mesmo as demonstrações e instruções para fazer com que um processador execute um método para compartilhar uma visualização ("visualização compartilhada") com o uso de um sistema de segurança físico que compreende uma pluralidade de nós de servidor, em que o método compreende enviar, de um primeiro cliente para um dos nós de servidor ("primeiro nó"), os dados de estado de visualização representativos da visualização compartilhada conforme exibido pelo primeiro cliente; enviar os dados de estado de visualização do primeiro nó para um segundo cliente por meio de outro dentre os nós de servidor ("segundo nó"); atualizar uma exibição do segundo cliente com o uso dos dados de estado de visualização para mostrar a visualização compartilhada; em resposta a uma mudança na visualização compartilhada no segundo cliente, enviar dados de estado de visualização atualizados do segundo cliente ao segundo nó, em que os dados de estado de visualização atualizados são representativos da visualização compartilhada conforme exibido pelo segundo cliente; enviar os dados de estado de visualização atualizados do segundo nó ao primeiro cliente por meio do primeiro nó; e atualizar o visor do primeiro cliente para mostrar a visualização compartilhada com o uso dos dados de estado de visualização atualizados.

[0056] Este sumário não necessariamente descreve todo o escopo de todos os aspectos. Outros aspectos, recursos e vantagens serão aparentes para aqueles de habilidade comum na técnica mediante a revisão da seguinte descrição de modalidades específicas.

BREVE DESCRIÇÃO DOS DESENHOS

[0057] Nos desenhos anexos, que ilustram uma ou mais modalidades exemplificativas:

[0058] A Figura 1 é um diagrama de blocos de um sistema de segurança físico distribuído, de acordo com uma modalidade.

[0059] A Figura 2 é um diagrama de blocos de um pacote de protocolos usado pelo sistema da Figura 1.

[0060] A Figura 3 é um diagrama de sequência de UML que mostra como o sistema da Figura 1 compartilha definições entre diferentes usuários de sistema.

[0061] A Figura 4 é um diagrama de sequência de UML que mostra como o sistema da Figura 1 compartilha um estado entre diferentes usuários de sistema.

[0062] A Figura 5 é um diagrama de sequência de UML que mostra como o sistema da Figura 1 compartilha uma visualização entre diferentes usuários de sistema.

[0063] A Figura 6 é um diagrama de sequência de UML que mostra como o sistema da Figura 1 compartilha transferências por streaming entre diferentes usuários de sistema.

[0064] A Figura 7 é uma visualização vista por um usuário do sistema da Figura 1.

[0065] A Figura 8 é um método para compartilhar dados em um sistema de segurança físico, de acordo com outra modalidade.

[0066] A Figura 9 é um método para reunir automaticamente um cluster, de acordo com outra modalidade.

[0067] A Figura 10 é um diagrama de sequência de UML que mostra como o sistema da Figura 1 compartilha uma visualização não atendida com um usuário de sistema.

[0068] A Figura 11 é um método para interagir com um visor não assistido em um sistema de segurança físico que compreende uma

pluralidade de nós de servidor, de acordo com outra modalidade.

[0069] A Figura 12 é um método para compartilhar uma visualização com o uso de um sistema de segurança físico que compreende uma pluralidade de nós de servidor, de acordo com outra modalidade.

DESCRIÇÃO DETALHADA

[0070] Termos direcionais, como "topo", "fundo", "para cima", "para baixo", "verticalmente", e "lateralmente" são usados na descrição a seguir com o propósito de fornecer referência relativa apenas, e não se destinam a sugerir quaisquer limitações sobre como qualquer artigo deve ser posicionado durante o uso, ou a ser montado em uma montagem ou relativo a um ambiente. Adicionalmente, o termo "acoplar" e variações do mesmo, como "acoplado", "acopla", e "que acopla", conforme usado nesta descrição se destina a incluir conexões indiretas e diretas, exceto se indicado o contrário. Por exemplo, se um primeiro dispositivo for acoplado a um segundo dispositivo, aquele acoplamento pode ser através de uma conexão direta ou através de uma conexão indireta por meio de outros dispositivos e conexões. Similarmente, se o primeiro dispositivo for acoplado de modo comunicativo ao segundo dispositivo, a comunicação pode ser através de uma conexão direta ou através de uma conexão indireta por meio de outros dispositivos e conexões.

[0071] Uma vez que um sistema de vigilância cresce para incluir um determinado número de câmeras, torna-se impraticável ou impossível operar o sistema de vigilância com o uso de um único servidor devido à capacidade de armazenamento e limitações de potência de processamento. Consequentemente, para acomodar o maior número de câmeras, servidores adicionais são adicionados ao sistema. Isso resulta em inúmeros problemas.

[0072] Por exemplo, um usuário do sistema de vigilância pode desejar ter a capacidade para visualizar o que outro usuário está visualizando (a "visualização" daquele usuário) e transferir por

streaming vídeo que é capturado com o uso de uma câmera no sistema ou que é armazenado em um servidor no sistema mesmo se o usuário não está diretamente conectado àquela câmera ou àquele servidor, respectivamente. Similarmente, o usuário pode desejar ter a capacidade para acessar estados de usuário (por exemplo: se outro usuário do sistema está atualmente conectado no sistema) e eventos de sistema (por exemplo: se um alarme foi acionado) que estão ocorrendo em outro lugar no sistema, mesmo se os mesmos se originam em um servidor, ao qual o usuário não é diretamente conectado. Em um sistema de vigilância convencional que teve sua escala aumentada ao adicionar mais servidores, um modo típico de fornecer essa funcionalidade é adicionar um servidor de porta centralizada no sistema. Um servidor de porta centralizada roteia eventos de sistema, estados de usuário, visualizações, e vídeo de um servidor no sistema para outro através de si mesmo, permitindo, assim, que o usuário acesse ou visualize esses eventos, estados, visualizações, e vídeo independentemente do particular servidor, ao qual o usuário é diretamente conectado. Entretanto, o uso de um servidor de porta centralizada proporciona ao sistema de vigilância um único ponto de falha, visto que, se o servidor de porta centralizada falha, então os eventos, estados, visualizações, e vídeo não podem mais ser compartilhados. O uso de um servidor de porta centralizada também aumenta o custo do sistema de vigilância, visto que um servidor é adicionado ao sistema e é dedicado a fornecer a funcionalidade do servidor de porta centralizada.

[0073] O usuário também pode desejar definições comuns (por exemplo: informações de acesso de usuário na forma de nomes de usuário, senhas, direitos de acesso, etc.) a serem sincronizados através de múltiplos servidores no sistema. Em um sistema de vigilância convencional que teve sua escala aumentada ao adicionar mais servidores, essa funcionalidade é fornecida seja exportando-se

manualmente definições de um servidor para outros servidores, ou usando-se um servidor de gerenciamento centralizado que armazena todas essas definições que outros servidores comunicam conforme for necessário para recuperar essas definições. Exportar manualmente definições é problemático devido a atrasos de sincronização relativamente grandes, dificuldade de uso e definição, e devido ao fato de que grandes atrasos de sincronização prejudicam a redundância de sistema. O uso do servidor de gerenciamento centralizado é submetido aos mesmos problemas que o uso do servidor de porta centralizada, conforme discutido acima.

[0074] Algumas das modalidades descritas no presente documento são direcionadas a um sistema de segurança físico distribuído, como um sistema de vigilância, que pode compartilhar automaticamente dados, como visualizações, vídeo, eventos de sistema, estados de usuário, e usuário definições entre dois ou mais nós de servidor no sistema sem depender de um servidor centralizado como os servidores de porta ou gerenciamento discutidos acima. Essas modalidades são direcionadas a um sistema de vigilância ponto a ponto no qual os usuários conectam por meio de clientes a nós de servidores, como gravadores de vídeo em rede, câmeras, e servidores. Nós de servidor são agrupados em clusters, sendo que cada nó de servidor no cluster tem capacidade para compartilhar dados com os outros nós de servidor no cluster. Para compartilhar esses dados, cada um dos nós de servidor executa serviços que trocam dados com base em um conjunto de protocolos que compartilha dados entre os nós de servidor em diferentes maneiras, dependendo se os dados representam visualizações, vídeo, eventos de sistema, estados de usuário, ou definições de usuário. As Figuras 1 a 10 retratam essas modalidades.

[0075] Em modalidades alternativas, algumas das tecnologias usadas para compartilhar visualizações entre diferentes nós de servidor

são aplicáveis a redes federadas (isto é, redes que incluem um servidor centralizado) e a redes ponto a ponto como aquelas mostradas nas Figuras 1 a 9. As Figuras 10 e 11 retratam essas modalidades.

[0076] Referindo-se agora à Figura 1, é mostrado um sistema de segurança físico distribuído na forma de um sistema de vigilância 100, de acordo com uma modalidade. O sistema 100 inclui três clientes 102a-c (primeiro cliente 102a a terceiro cliente 102c e coletivamente "clientes 102"), seis servidores 104a-f (primeiro servidor 104a a sexto servidor 104f e coletivamente "servidores 104"), três câmeras de nó de servidor 106a-c (primeira câmera de nó 106a a terceira câmera de nó 106c e coletivamente "câmeras de nós 106"), e cinco câmeras de não nó 114.

[0077] Cada um dentre as câmeras de nós 106 e os servidores 104 inclui um processador 110 e uma memória 112 que são acoplados de modo comunicativo entre si, com a memória 112 que tem codificado na mesma as demonstrações e instruções para fazer com que o processador 110 execute quaisquer modalidades dos métodos descritos no presente documento. Os servidores 104 e as câmeras de nós 106 são agrupados em três clusters 108a-c (coletivamente "clusters 108"): o primeiro a terceiro servidores 104a-c são acoplados de modo comunicativo entre si para formar um primeiro cluster 108a; o quarto a sexto servidores 104d-f são acoplados de modo comunicativo entre si para formar um segundo cluster 108b; e as três câmeras de nós 106 são acopladas de modo comunicativo entre si para formar um terceiro cluster 108c. O primeiro a terceiro servidores 104a-c são referidos como "membros" do primeiro cluster 108a; o quarto a sexto servidores 104d-f são referidos como "membros" do segundo cluster 108b; e a primeira a terceira câmeras de nós 106a-c são referidas como "membros" do terceiro cluster 108c.

[0078] Cada um dentre os servidores 104 e as câmeras de nós 106 é um "nó de servidor" em que cada um está ciente da presença dos outros

membros de seu cluster 108 e pode enviar dados para os outros membros de seu cluster 108; ao contrário, as câmeras de não nó 114 não são nós de servidor em que estão cientes apenas dos servidores 104a,b,c,d,f aos quais as mesmas são diretamente conectadas. Na modalidade retratada, os nós de servidor estão cientes de todos os outros membros do cluster 108 por virtude de ter acesso a informações de associação de cluster, que lista todos os nós de servidor no cluster 108. As informações de associação de cluster são armazenadas de modo persistente e de modo localizado em cada um dos nós de servidor, que permite que cada um dos nós de servidor automaticamente reúna seu cluster 108 caso reinicie durante a operação do sistema 100. Uma referência dourante no presente documento a um "nó" é uma referência a um "nó de servidor", exceto se indicado o contrário.

[0079] Embora, na modalidade retratada, nenhum dos clusters 108 participe na comunicação intercluster, em modalidades alternativas (não mostrado) os membros de vários clusters 108 podem compartilhar dados entre si. Na modalidade retratada, os servidores 104 são servidores comerciais prontos para uso e as câmeras 106,114 são fabricadas pela Corporação Avigilon™ de Vancouver, Canadá; entretanto, em modalidades alternativas, outros tipos adequados de servidores 108 e câmeras 106, 114 podem ser usados.

[0080] O primeiro cliente 102a é acoplado de modo comunicativo aos primeiro e segundo clusters 108a,b por virtude de ser acoplado de modo comunicativo ao primeiro e quarto servidores 104a,d, que são membros daqueles clusters 108a,b; o segundo cliente 102b é acoplado de modo comunicativo a todos os três clusters 108 por virtude de ser acoplado de modo comunicativo ao segundo e ao quarto servidores 104b,d e à primeira câmera de nó 106a, que são membros daqueles clusters 108; e o terceiro cliente 102c é acoplado de modo comunicativo ao segundo e terceiro clusters 108b,c por virtude de ser acoplado de modo comunicativo ao

quinto servidor 104e e o segundo câmera de nó 106b, que são membros daqueles clusters 108b,c. Conforme discutido em mais detalhes abaixo, em qualquer um dos clusters 108a-c, cada um dos nós executa serviços que permitem que os nós comuniquem entre si de acordo com um conjunto de protocolos 200 (mostrado na Figura 2), que permite que qualquer um nó compartilhe dados, se aqueles dados forem visualizações, vídeo, eventos de sistema, estados de usuário, definições de usuário, ou outro tipo de dados, a qualquer outro nó com o uso de computação distribuída; isto é, sem usar um servidor centralizado. Cada um dos nós tem acesso a informações de associação de cluster que identificam todos os nós que fazem parte do mesmo cluster 108; ao acessar essas informações de associação de cluster, dados podem ser compartilhados e sincronizados entre todos os nós de um cluster 108.

[0081] A Figura 2 mostra um diagrama de blocos do conjunto de protocolos 200 empregado pelos nós do sistema 100. O conjunto de protocolos 200 é dividido em três camadas e inclui os seguintes protocolos, conforme resumido na Tabela 1:

Tabela 1: Sumário do Conjunto de Protocolos 200

<u>Nome de protocolo</u>	<u>Camada de protocolo</u>	<u>Recebe Dados desses Protocolos e Aplicações</u>	<u>Envia Dados para esses Protocolos</u>
UDP 202	Transporte	Protocolo de Descoberta 206, Protocolo de Nó 210, Protocolo de Sincronia 214	N/A
TCP/HTTP 204	Transporte	Protocolo de nó 210, Protocolo Gossip 208, Protocolo de Fidelidade 212, Protocolo de Consistência 216, Protocolo de Situação 218	N/A
Protocolo de Descoberta 206	Suporte de cluster	Protocolo de Nó 210	UDP 202
Protocolo Gossip 208	Supporte de cluster	Protocolo de Fidelidade 212, Protocolo de Consistência 216, Protocolo de Situação 218	TCP/HTTP 204, Protocolo de nó 210, Protocolo de Fidelidade 212
Protocolo de Nó 210	Supporte de cluster	Aplicação de transmissão por streaming de cluster 220, Sincronia 214, Protocolo de Consistência 216, Protocolo de Fidelidade 212, Protocolo de Situação 218, Protocolo Gossip 208	UDP 202, TCP/HTTP 204, Protocolo de Descoberta 206

Protocolo de Fidelidade 212	Suporte de cluster	Protocolo de Sincronia 214, Protocolo Gossip 208, Protocolo de Situação 218, Protocolo de Consistência 216	Protocolo Gossip 208, Protocolo de Nó 210, TCP/HTTP 204
Protocolo de Sincronia 214	Sincronização de dados	Aplicação de Colaboração e Visualizações Compartilhadas 222, Aplicação de Alarmes e Eventos Compartilhados 224, Aplicação de Compartilhamento de Visualização Não Assistida 225	UDP 202, Protocolo de Nó 210, Protocolo de Associação 212
Protocolo de Consistência 216	Sincronização de dados	Aplicação de Definições Compartilhadas 226, Aplicação de Objetos de Usuário Compartilhados 228, Aplicação de Compartilhamento de Visualização Não Assistida 225	Protocolo de Nó 210, Protocolo de Associação 212, Protocolo Gossip 208, TCP/HTTP 204
Protocolo de Situação 218	Sincronização de dados	Aplicação de Informações de Sistema (dispositivo, servidor, etc.) 230	Protocolo Gossip 208, Protocolo de Associação 212, Protocolo de Nó 210, TCP/HTTP 204

[0082] Uma descrição da função e operação de cada um dos protocolos no conjunto de protocolos 200 segue.

CAMADA DE TRANSPORTE

[0083] A Camada de transporte corresponde à camada 4 do modelo de Interconexão de Sistemas Abertos (OSI), e é responsável por fornecer serviços de transferência de dados confiável entre nós ao suporte de cluster, sincronização de dados e camadas de aplicação. A Camada de Transporte no sistema 100 inclui os protocolos UDP 202 e TCP/HTTP 204.

CAMADA DE SUPORTE DE CLUSTER

[0084] A Camada de Suporte de Cluster inclui os protocolos usados para constatar nós, verificar existência de nó, verificar vivacidade de nó, determinar se um nó é um membro de um dos clusters 108, e determinar como rotear dados entre nós.

PROTOCOLO DE DESCOBERTA 206

[0085] O protocolo de Descoberta 206 tem como base a versão 1.1 do protocolo de Descoberta WS publicado pela Organização para o Avanço de Padrões de Informações Estruturados (OASIS), cujo

conteúdo integral é aqui incorporado a título de referência no presente documento. Na modalidade retratada, a formatação de XML usada no padrão publicado é substituída por codificação Protobuf da Google™.

[0086] O protocolo de Descoberta 206 permite que qualquer nó no sistema 100 identifique os outros nós no sistema 100 por multidifusão de Mensagens de Sonda àqueles outros nós e aguardar para que os mesmos respondam. Um nó pode alternativamente difundir uma mensagem de Olá quando ingressar o sistema 100 para alertar outros nós sobre sua presença sem exigir que esses outros nós realizem primeiro a multidifusão da Mensagem de sonda. Tanto a mensagem de sonda quanto a mensagem de olá são modeladas no protocolo de Descoberta WS publicado por OASIS.

PROTOCOLO GOSSIP 208

[0087] O protocolo Gossip 208 é um protocolo epidêmico que dissemina dados de um dos nós para todos os nós daquele cluster 108 ao realizar aleatoriamente trocas de dados entre pares de nós no cluster 108. O protocolo Gossip 208 comunica a vivacidade ao trocar dados de "estado de batimento cardíaco" na forma de uma contagem de batimentos cardíacos para cada nó, o que permite que os nós determinem quando um dos nós no cluster 108 saiu inesperadamente (por exemplo: devido a uma queda de servidor). O protocolo Gossip 208 também comunica dados de "estado de aplicação" como hashes de nível de topo usados pelo protocolo de Consistência 216 e identificador de entidade de situação e seus números de versão usados pelo protocolo de Situação 218 para determinar quando sincronizar dados entre os nós, conforme discutido em mais detalhes abaixo. O espalhamento de dados com o uso do protocolo Gossip 208 eventualmente espalha para todos os nós no cluster 108 por meio de nó periódico para trocas de nó.

[0088] Uma troca de dados entre quaisquer dois nós do cluster 108

com o uso do protocolo Gossip 208 envolve realizar duas chamadas de procedimento remoto (RPCs) de um primeiro nó ("Nó A") para um segundo nó ("Nó B") no mesmo cluster 108, conforme segue:

[0089] O Nó A envia uma mensagem GreetingReq para o Nó B, que contém uma lista de condensados para todos os nós no cluster 108 do quais o Nó A está ciente. Para cada nó, um condensado inclui um identificador de nó exclusivo e informações de versão que são incrementados sempre que o estado de batimento cardíaco ou o estado de aplicação para aquele nó muda. As informações de versão podem ser, por exemplo, um número de versão unidimensional ou um vetor de versão multidimensional. O uso de um vetor de versão permite que o condensado sumarize o histórico das mudanças de estado que o nó foi submetido.

[0090] O Nó B envia uma mensagem GreetingRsp para o Nó A, que contém:

[0091] uma lista de condensados para nós sobre os quais o Nó B deseja receber mais informações de Nó A, cujo Nó B determina a partir das informações de versão enviadas para o mesmo na mensagem GreetingReq;

[0092] uma lista de condensados para nós sobre os quais o Nó A não faz parte do cluster 108;

[0093] uma lista de um ou ambos os estados de batimento cardíaco e aplicação que irão atualizar o Nó A sobre nós para os quais o mesmo tem informações desatualizadas; e

[0094] uma lista de nós que o Nó A acredita fazer parte do cluster 108, mas que o Nó B sabe ter sido removido do cluster 108.

[0095] O Nó A então envia uma mensagem ClosureReq para o Nó B, no qual o Nó A envia:

[0096] uma lista de condensados para nós sobre os quais o Nó A deseja receber mais informações do Nó B (por exemplo, o Nó A pode

solicitar informações para nós dos quais o Nô A não estava ciente até o Nô B enviar ao Nô A a mensagem GreetingRsp);

[0097] uma lista de estados que irá atualizar o Nô B sobre nós para os quais o mesmo tem informações desatualizadas; e

[0098] uma lista de nós que o Nô B acredita fazer parte do cluster 108, mas que o Nô A sabe ter sido removido do cluster 108.

[0099] O Nô B então envia uma mensagem ClosureRsp para o Nô A, no qual o Nô B envia:

[00100] uma lista de estados que irá atualizar o Nô A sobre nós sobre os quais o mesmo está desatualizado, em resposta à solicitação do Nô A em ClosureReq; e

[00101] uma lista de nós que foi removida do cluster 108 desde GreetingRsp.

[00102] Após os Nós A e B trocarem RPCs, os mesmos terão listas de nó ativas idênticas, que incluem as versões mais recentes do estado de batimento cardíaco e estado de aplicação para todos os nós no cluster 108 sobre o qual ambos sabiam antes dos RPCs e que não foram removidos do cluster 108.

PROTOCOLO DE NÔ 210

[00103] O protocolo de Nô 210 é responsável por gerar uma topologia de rede da visualização do sistema 100 para cada nó, que fornece cada nó com um mapa de rede que permite que o mesmo comunique com qualquer outro nó no sistema 100. Em algumas modalidades, o mapa de rede é uma tabela de roteamento. O mapa de rede referencia pontos de extremidade, que são um endereço (IP/FQDN), número de porta, e protocolo pelo qual um nó pode ser atingido através da rede de IP que conecta os nós.

[00104] O protocolo de Nô 210 faz isso de três modos:

[00105] por meio de uma "troca Poke", conforme descrito em maiores detalhes abaixo;

[00106] por meio do protocolo de Descoberta 206, que notifica o protocolo de nó 210 quando um nó ingressa ou deixa o sistema 100. Quando um nó ingressa o sistema 100 uma "troca Poke" é executada com aquele nó; e

[00107] manualmente, em resposta à inserção do usuário.

[00108] Uma troca Poke envolve realizar periodicamente os seguintes RPCs com o propósito de gerar mapas de rede para os nós:

[00109] Uma solicitação Poke, na qual o Nó A envia para o Nó B uma autovisualização de Nó A e uma lista de outros nós conhecidos por Nó A, conforme visualizado pelo Nó A, em que, em seguida, o Nó B atualiza seu mapa de rede na visualização dessas informações; e

[00110] uma resposta Poke, na qual o Nó B envia para o Nó A uma autovisualização de Nó B e uma lista de outros nós conhecidos ao Nó B, conforme visualizado pelo Nº B, em que, em seguida, o Nº A atualiza seu mapa de rede na visualização dessas informações.

[00111] Os RPCs são executados através do protocolo TCP/HTTP 204.

[00112] Para reduzir o uso de largura de banda, informações de nó são trocadas apenas entre Nós A e B se as informações de nó foram trocadas desde a última vez que as mesmas foram trocadas.

[00113] Uma troca Poke é executada após o protocolo de Descoberta 206 notificar o protocolo de Nº 210 que um nó ingressou no sistema 100 devido ao fato de que o protocolo de Descoberta 206 anuncia pontos de extremidade de um nó, mas não garante que o nó seja alcançável com o uso daqueles pontos de extremidade. Por exemplo, os pontos de extremidade podem não ser úteis devido a um firewall. Realizar uma troca Poke em um nó identificado com o uso do protocolo de descoberta 206 confirma se os pontos de extremidade são, de fato, úteis.

[00114] O protocolo de Nº 210 também pode confirmar se um ponto

de extremidade de comunicação de UDP anunciado é alcançável; entretanto, o protocolo de Número 210 na modalidade retratada não executa uma troca Poke através do protocolo UDP 202.

[00115] Para qualquer nó dado em um cluster 108, um mapa de rede relaciona identificadores de nó a pontos de extremidade de comunicação para cada um dos nós no mesmo cluster 108. Consequentemente, os outros protocolos na pilha de protocolo 200 que comunicam com o protocolo de Número 210 podem entregar mensagens a qualquer outro nó no cluster 108 apenas usando-se aquele identificador de nó do nó.

PROTOCOLO DE FIDELIDADE 212

[00116] O protocolo de Fidelidade 212 é responsável por garantir que cada nó de um cluster 108 mantenha informações de associação de cluster para todos os nós do cluster 108, e para permitir que nós ingressem e deixem o cluster 108 por meio de RPCs. Informações de associação de cluster são compartilhadas entre nós do cluster 108 com o uso do protocolo de Situação 218. Cada nó no cluster 108 mantém sua própria versão das informações de associação de cluster e aprende do protocolo de Situação 218 as informações de associação de cluster retidas pelos outros nós no cluster 108. Conforme discutido em maiores detalhes abaixo, as versões de informações de associação de cluster retidas por dois nós diferentes podem não corresponder devido à versão de informações de associação de cluster armazenadas em um nó e que foram recentemente atualizadas podem não ter sido ainda sincronizadas com os outros membros do cluster 108.

[00117] Para cada nó, as informações de associação de cluster incluem:

[00118] Uma lista de fidelização de todos os nós do cluster 108, em que cada um dos nós é representado por:

[00119] o identificador de nó, que é exclusivo dentre todos os nós no

sistema 100;

[00120] o estado do nó, que é qualquer um dentre:

[00121] Descobrir: o nó é um membro do cluster 108, mas que não foi sincronizado com os outros membros do cluster 108 desde que foi inicializado;

[00122] Ingressar: o nó está no processo de ingressar a um cluster 108;

[00123] Sincronização: o nó está no processo de sincronizar dados com o uso dos protocolos de Sincronia, Consistência e Situação 214,216,218 com o cluster 108 que o mesmo acabou de ingressar;

[00124] Válido: o nó completou a sincronização das informações de associação de cluster e é um nó válido do cluster 108; e

[00125] Tempo Limite Esgotado: o nó não está mais respondendo e não é mais um membro ativo do cluster 108 (o nó permanece um membro do cluster 108 até ser removido por um usuário);

[00126] um token de sessão;

[00127] o número de versão das informações de associação de cluster quando o nó ingressou o cluster 108; e

[00128] o número de versão das informações de associação de cluster da última vez que foi mudado.

[00129] Uma lista de marca de exclusão que lista todos os nós que foram removidos do cluster 108, em que cada nó removido é representado por:

[00130] o identificador de nó do nó; e

[00131] a versão daquelas informações de associação de cluster do nó quando o nó foi removido.

[00132] Na modalidade retratada, um nó é sempre um membro de um cluster 108 que compreende pelo menos o próprio; um cluster 108 de um nó é referido como um "cluster único". Adicionalmente, embora na modalidade retratada as informações de associação incluam a lista de

fidelização e lista de marca de exclusão, conforme descrito acima, em modalidades alternativas (não retratadas) as informações de associação podem ser compreendidas de modo diferente; por exemplo, em tal modalidade alternativa as informações de associação carecem de uma lista de marca de exclusão, embora em outra tal modalidade o estado do nó possa ser descrito de modo diferente do que o descrito acima.

[00133] Quando o Nó A deseja atuar como um novo nó de servidor e deseja ingressar em um cluster 108 que inclui Nó B, o mesmo se comunica com Nó B e o seguinte ocorre:

[00134] O Nó A envia um segredo de cluster ao Nó B, que, na modalidade retratada, é uma chave que o Nó B exige antes de deixar que outro nó ingresse em seu cluster 108. Um dos clientes 102 fornece o segredo de cluster ao Nó A. À medida que o Nó B controla o acesso ao cluster 108 do Nó A, o Nó B atua como um "nó de controle de associação".

[00135] Os Nós A e B trocam suas informações de associação. As versões das informações de associação nos Nós A e B são atualizadas para incluir os identificadores de nó de Nó A e de todos os nós do cluster 108 que o Nó A está ingressando.

[00136] O estado do Nó A é mudado para "Ingressar" conforme o Nó A ingressa o cluster.

[00137] Uma vez que ingressado, o estado do Nó A é alterado para "Sincronizando" já que dados são trocados entre o Nó A e o cluster 108 ao qual o mesmo acaba de ingressar. O Nó B também atualiza a versão das informações de associação armazenadas em todos os outros nós do cluster 108 com o uso do protocolo de Situação 218. O processo de atualizar as versões das informações de associação armazenadas no Nó A e todos os membros do cluster 108 que o Nó A está ingressando é chamado de "sincronização" das versões das informações de associação armazenadas em todos esses nós.

[00138] Após a sincronização ser concluída, o estado do Nó A muda para Válido.

CAMADA DE SINCRONIZAÇÃO DE DADOS

[00139] A camada de sincronização de dados inclui os protocolos que possibilitam que os dados sejam enviados entre os nós em um cluster com garantias de ordem diferentes e trocas de desempenho. Os protocolos na Camada de Sincronização de Dados usam diretamente os protocolos nas Camadas de Suporte de Cluster e Transporte.

PROTOCOLO DE SINCROIA 214

[00140] O protocolo de Sincronia 214 é usado para enviar dados na forma de mensagens de Nó A para Nó B no sistema 100 de modo que as mensagens cheguem em Nó B em uma ordem que o Nó A pode controlar, como a ordem na qual o Nó A envia as mensagens. Os serviços que transferem os dados com o uso do protocolo de Sincronia 214 são executados em sequências de serviço de I/O de alta prioridade dedicadas.

[00141] Na modalidade retratada, o protocolo de Sincronia 214 é baseado em uma implantação de sincronia virtual conhecido como o protocolo Totem, conforme descrito em Agarwal DA, Moser LE, Melliar-Smith PM, Budhia RK, "The Totem Multiple-Ring Ordering and Topology Maintenance Protocol", ACM Transactions on Computer Systems, 1998, páginas 93 a 132, cuja totalidade está incorporada a título de referência no presente documento. No protocolo de Sincronia 214, os nós são agrupados juntos em grupos chamados doravante nesta descrição de "anéis de Sincronia", e um nó em qualquer anel de Sincronia pode enviar mensagens totalmente ordenadas aos outros nós no mesmo anel. O protocolo de Sincronia 214 modifica o protocolo Totem como a seguir:

[00142] O protocolo de Sincronia 214 usa tanto um identificador de serviço e um identificador de anel para identificar um anel de Sincronia. O identificador de serviço identifica todos os casos de um dado anel de

Sincronia, enquanto que o identificador de anel identifica um caso particular de um dado anel de Sincronia. Por exemplo, cada vez que um nó se une ou deixa um anel de Sincronia aquele identificador de anel do anel mudará, mas não seu identificador de serviço. O identificador de serviço permite que um nó realize a difusão seletiva de mensagens totalmente ordenadas ao grupo de nós que compartilham o mesmo identificador de serviço (isto é, o grupo de nós que pertence ao mesmo anel de Sincronia).

[00143] 2. No protocolo Totem, em alguns casos quando os nós não estão enviando mensagens o anel de Sincronia visto pelos nós não reflete a configuração de anel final que converge quando os nós começam o envio de mensagens. O protocolo de Sincronia 214 permite que os nós enviem mensagens de sonda entre si para fazer com que os anéis de Sincronia converjam antes do envio de mensagens sem sonda.

[00144] 3. O protocolo Totem somente permite que mensagens ordenadas sejam enviadas a todos os nós que formam parte de um anel de Sincronia. Em contraste, o protocolo de Sincronia 214 usa um módulo de Expedição que abstrai a camada de rede do protocolo de Sincronia 214 fornecendo-se uma interface para difundir todos os nós atingíveis no sistema 100; unidifundir para cada um dos múltiplos nós no sistema 100 com o uso de uma lista de identificadores de nó de destino; e para unidifundir para um único nó no sistema 100 com o uso de seu identificador de nó. O módulo de Expedição também suporta a multiplexação dos serviços na mesma porta de IP com o uso de roteamento e filtragem de mensagens pelo identificador de serviço. As mensagens de saída de um nó são enviadas para o subconjunto de nós que tem o mesmo identificador de serviço a menos que seja difusão seletiva.

[00145] 4. O protocolo de Sincronia 214 usa mensagens fragmentadas e coalescência de fragmentação de carga de usuário para lidar com os problemas que surgem do tamanho de unidade de

transmissão máxima de aproximadamente 1.500 bytes.

[00146] 5. O protocolo de Sincronia 214 modifica a forma que os nós usam mensagens de União, que são mensagens que os nós usam no protocolo de totem para unir um anel de Sincronia:

[00147] mensagens de Ingresso são enviadas pelos nós somente se as mesmas tiverem o menor identificador de nó no conjunto atual de nós operacionais no anel de Sincronia.

[00148] Nós que não têm o menor identificador de nó em seu conjunto operacional unidifundem mensagens de Ingresso aos nós com o menor identificador de nó em seu conjunto operacional.

[00149] mensagens de Ingresso incluem o identificador de serviço, e nós que não são parte do anel de Sincronia correspondente não respondem.

[00150] Em relação ao protocolo Totem, essas modificações ajudam a reduzir a largura de banda agregada usada pelos nós para unir anéis de Sincronia.

[00151] O protocolo de Sincronia 214 detecta e adiciona à lista negra os nós que não têm a capacidade de unir um anel de Sincronia devido a alguns tipos de configurações errôneas de rede. Por exemplo, um nó que tem a capacidade de enviar para, mas não de receber mensagens dos, os outros nós aparecerão aos outros nós para somente enviar mensagens de sonda visto que todas as mensagens na presente modalidade são solicitadas, e consequentemente serão adicionadas à lista negra.

[00152] O protocolo de Sincronia 214 realiza a criptografia de carga e verificação de autenticidade das mensagens.

[00153] O protocolo de Sincronia 214 limita o tempo em que cada nó pode reter o token usado no protocolo Totem; na modalidade retratada, cada nó pode reter o token por 15 ms.

[00154] O protocolo de Sincronia 214 implanta um algoritmo para

evitar o congestionamento amigável a TCP.

[00155] Conforme discutido em maiores detalhes abaixo, o sistema 100 usa o protocolo de Sincronia para a aplicação de Colaboração e Visualização Compartilhada 222 e a aplicação de Alarme e Eventos Compartilhados 224; os dados compartilhados entre os membros de um cluster 108 nessas aplicações 222 são não persistentes e são compartilhados de forma benéfica rapidamente e em uma ordem conhecida.

PROTOCOLO DE CONSISTÊNCIA 216

[00156] O protocolo de Consistência 216 é usado para compartilhar automatica e periodicamente os dados através de todos os nós de um cluster 108 de modo que os dados que são compartilhados com o uso do protocolo de Consistência 216 sejam eventualmente sincronizados em todos os nós no cluster 108. Os tipos de dados que são compartilhados com o uso do protocolo de Consistência 216 são discutidos em mais detalhes abaixo nas seções que discutem a aplicação de Definições Compartilhadas 226 e a aplicação de Objetos de Usuário Compartilhados 228. Os dados compartilhados pelo protocolo de Consistência 216 são armazenados em um banco de dados em cada um dos nós, e cada entrada no banco de dados inclui um par chave-valor em que a chave identifica exclusivamente o valor e as chaves são independentes entre si. O protocolo de Consistência 216 sincroniza os dados através dos nós enquanto resolve modificações paralelas que os nós diferentes podem realizar em bancos de dados diferentes. Conforme discutido em detalhes adicionais abaixo, o protocolo de Consistência 216 realiza isso sendo primeiro notificado que os bancos de dados não são sincronizados; segundo, encontrar quais bancos de dados de entrada particulares são sincronizados; e terceiro, encontrar qual versão da entrada é mais recente, sincronizada, e mantida.

[00157] A fim de resolver modificações paralelas que determinam

quando as mudanças são feitas aos bancos de dados, cada nó que ingressa a um cluster 108 é atribuído a um mecanismo de controle de versão de causalidade usado para registrar quando aquele nó faz as mudanças aos dados e para determinar se as mudanças foram feitas antes ou após as mudanças aos mesmos dados feitos por outros nós no cluster 108. Na presente modalidade, cada um dos nós usa um relógio de árvore de intervalo (ITC) como um mecanismo de controle de versão de causalidade. Entretanto, em modalidades alternativas outros mecanismos de controle de versão como relógios de vetor podem ser usados. O sistema 100 também implanta um relógio de tempo universal (UTC), que é sincronizado entre nós diferentes com o uso de Protocolo de Tempo de Rede, para determinar a ordem na qual as mudanças são feitas quando os ITCs para dois ou mais nós são idênticos. Os ITCs são descritos em mais detalhes em P. Almeida, C. Baquero, e V. Fonte, "Interval tree clocks: a logical clock for dynamic systems", Princípios de Sistemas Distribuídos, Lecture Notes in Comp. Sci., vol. 5401, páginas 259 a 274, 2008, cuja totalidade é incorporada aqui a título de referência no presente documento.

[00158] O diretório que o protocolo de Consistência 216 sincroniza entre os nós é dividido em ramificações, cada uma das quais é chamada de um Domínio de Consistência Eventual (ECD). O protocolo de Consistência 216 sincroniza cada um dos ECDs independentemente a partir dos outros ECDs. Cada entrada de banco de dados dentro de um ECD é chamada de Entrada de Consistência Eventual (ECE). Cada ECE inclui uma chave; um carimbo de data/hora de um ITC e do UTC, que são ambos atualizados sempre que o ECE é modificado; um valor do hash do ECE que gera com o uso, por exemplo, de uma função Murmurhash; os próprios dados; e uma marca de exclusão que é adicionada se e quando o ECE for excluído.

[00159] O valor do hash é usado para comparar ECDs e ECEs correspondentes em dois nós diferentes para determinar se eles são

idênticos. Quando dois ECDs correspondentes são comparados, hashes de "nível de topo" para aqueles ECDs são comparadas. Uma hash de nível de topo para um ECD em um dado nó é gerada realizando-se hashing em todos os ECEs dentro do ECD. Se os hashes de nível de topo forem compatíveis, então os ECDs são idênticos; de outra forma, o protocolo de Consistência 216 determina que os ECDs são diferentes. Para determinar quais ECEs particulares nos ECDs são diferentes, hashes são tomadas para reduzir de forma sucessiva as faixas dos ECEs em ambos os nós. Os intervalos através dos quais os hashes são tomados eventualmente encolhem o suficiente para que os ECEs que diferem entre os dois nós sejam isolados e identificados. Uma lista a ignorar bidirecional pode ser usada, por exemplo, para determinar e comparar os valores do hash dos intervalos de ECD.

[00160] Dois nós que se comunicam com o uso do protocolo de Consistência 216 podem usar os seguintes RPCs:

[00161] SetEntries: SetEntries transmite ECEs novos ou atualizados para um nó, que insere os mesmos nós ECDs apropriados.

[00162] GetEntries: GetEntries transmite uma chave ou uma faixa de chaves para um nó, que retorna os ECEs correspondentes àquelas uma ou mais chaves.

[00163] SynEntries: SynEntries transmite uma chave ou uma faixa de chaves a um nó, e os dois nós então comparam hashes de faixas que reduzem de forma sucessiva dos ECEs para determinar quais ECEs diferem entre os dois nós, conforme descrito acima. Se os ECEs forem diferentes, os nós mesclam seus ECEs de modo que os mesmos ECEs sejam armazenados nos nós comparando-se os carimbos de data/hora de ITC; se os carimbos de data/hora de ITC corresponderem, os nós comparam os carimbos de data/hora de UTC associados com os ECEs. Esses carimbos de data/hora agem como informações de versão que permitem que os dois nós adotem os ECEs que foram modificados mais

recentes, conforme indicado por aquelas informações de versão dos ECEs.

[00164] Quando um nó muda os ECEs, aquele nó tipicamente executa SynEntries para informar os outros nós no cluster 108 que os ECEs foram alterados. Se alguns dos nós no cluster 108 estiverem indisponíveis (por exemplo: os mesmos estiverem offline), então o protocolo Gossip 208 em vez do SynEntries é usado para comunicar os hashes de nível de topo para os nós indisponíveis uma vez que eles voltam a ficar online. Conforme aludido na seção que discute o protocolo Gossip 208 no cluster 108 acima, cada um dos nós mantém sua hash de nível de topo, que é espalhada para os outros nós junto com um identificador de nó, informações de versão, e estado de batimento cardíaco com o uso do protocolo Gossip 208. Quando outro nó recebe esse hash, o mesmo compara o hash de nível de topo recebida com seu próprio hash de nível de topo. Se os hashes de nível de topo forem idênticos, os ECEs em ambos os nós correspondem; de outra forma, os ECEs diferem.

[00165] Se os ECEs forem diferentes, independentemente de se isso for determinado com o uso de SynEntries ou do protocolo Gossip 208, o nó que executa SynEntries ou que recebe o hash de nível de topo sincroniza os ECEs.

PROTOCOLO DE SITUAÇÃO 218

[00166] Conforme discutido acima, o protocolo Gossip 208 compartilha por todo o cluster 108 identificadores de entidade de situação e seus números de versão ("par de entidade de situação") para os nós no cluster 108. Os identificadores de entidade de situação podem representar, por exemplo, tipos diferentes de dados de situação na forma de entradas de situação como quanto armazenamento o nó tem disponível; quais dispositivos (como as câmeras de não nó 114) são conectados àquele nó; quais clientes 102 são conectados àquele nó; e

informações de associação de cluster. Quando um dos nós recebe esses dados por meio do protocolo Gossip 208, o mesmo compara o número de versão do par de entidade de situação ao número de versão da entrada de situação correspondente que está armazenando de modo localizado. Se os números de versão forem diferentes, o protocolo de Situação 218 começa um RPC ("RPC de Sincronização") com o nó a partir do qual o par de entidade de situação se origina para atualizar a entrada de situação correspondente.

[00167] Uma entrada de situação sincronizada com o uso do protocolo de Situação 218 é identificada exclusivamente tanto por um identificador de caminho quanto de nó. Diferente dos dados sincronizados com o uso do protocolo de Consistência 216, o nó que a entrada de situação descreve é o único que pode modificar a entrada de situação ou o par de entidade de situação. Consequentemente, e diferentes dos ECDs e ECEs sincronizados com o uso do protocolo de Consistência 216, a versão da entrada de situação para o Nó A armazenada de modo localizado no Nó A é sempre a versão mais recente daquela entrada de situação.

[00168] Se o Nó A modificar múltiplas entradas de situação simultaneamente, o protocolo de Situação 218 sincroniza todas as entradas de situação modificadas juntas ao Nó B quando o Nó B executa o RPC de Sincronização. Consequentemente, as entradas alteradas simultaneamente podem ser dependentes uma da outra porque elas serão enviadas juntas para o Nó B para análise. Em contraste, cada um dos ECEs sincronizados com o uso do protocolo de Consistência 216 é sincronizado independentemente dos outros ECEs, então os ECEs não podem ser dependentes um do outro já que o Nó B não pode depender de receber as entradas em qualquer ordem particular.

APLICAÇÕES

[00169] Cada um dos nós no sistema 100 executa serviços que

implantam o pacote de protocolos 200 descrito acima. Embora na modalidade retratada um serviço seja usado para cada um dos protocolos 202 a 218, nas modalidades alternativas (não representadas) mais ou menos serviços podem ser usados para implantar o pacote de protocolos 200. Cada um dos nós implanta o próprio pacote de protocolos 200; consequentemente, o sistema 100 é distribuído e está menos vulnerável a uma falha de qualquer nó, que está em contraste aos sistemas de segurança física convencionais que usam um servidor centralizado. Por exemplo, se um dos nós falhar no sistema 100 ("nó falho"), em cada um dos nós remanescentes o serviço que executa o protocolo de Situação 218 ("serviço de Situação") determinará que o nó falho está offline monitorando-se o estado de batimento cardíaco do nó falho e comunicará essa falha ao serviço que executa o nó e protocolos de associação 210,212 em cada um dos outros nós ("serviço de nó" e "serviço de associação", respectivamente). Os serviços em cada nó que implanta os protocolos de sincronia e consistência 214,216 ("serviço de sincronia" e "serviço de consistência", respectivamente) irão parar, subsequentemente, de compartilhar os dados com o nó falho até que o nó falho fique online novamente e ingresse a seu cluster 108.

[00170] O seguinte descreve as várias aplicações 220 a 230 que o sistema 100 pode implantar. As aplicações 220 a 230 podem ser implantadas como várias modalidades do método exemplificativo para compartilhar os dados 800 representados na Figura 8. O método 800 começa no bloco 802 e prossegue para o bloco 804 em que um primeiro nó no sistema 100 acessa um identificador de nó que identifica outro nó no sistema 100. Tanto o primeiro quanto o segundo nós são membros do mesmo cluster de servidor 108. O identificador de nó que o primeiro nó acessa é parte das informações de associação de cluster que identificam todos os membros do cluster 108. As informações de associação de cluster são acessíveis por todos os membros do cluster

108. Nas modalidades retratadas, cada um dos membros do cluster 108 armazena sua própria versão das informações de associação de cluster de modo persistente e de modo localizado; entretanto, nas modalidades alternativas (não representadas), as informações de associação de cluster podem ser armazenadas tanto remotamente dos nós quanto em uma localização central. Após acessar o identificador de nó para o segundo nó, o primeiro nó envia os dados ao segundo nó no bloco 806, após o qual o método 800 termina no bloco 808. Por exemplo, quando se usa o nó serviço descrito acima, os serviços de sincronia e consistência em execução no primeiro nó têm a capacidade de enviar os dados ao segundo nó através do uso do identificador de nó do segundo nó, e delegando-se ao nó serviço a responsabilidade para associar o ponto de extremidade do segundo nó ao seu identificador de nó. O envio dos dados do primeiro nó ao segundo nó no bloco 806 pode compreender parte de uma troca de dados bidirecional, como quando os dados estão trocados de acordo com o protocolo Gossip 208.

APLICAÇÃO DE DEFINIÇÕES COMPARTILHADAS 226 E APLICAÇÃO DE OBJETOS DE USUÁRIO COMPARTILHADOS 228

[00171] Durante a operação do sistema 100, as informações armazenadas de modo persistente são transferidas entre os nós de um cluster 108. Os exemplos dessas informações em tempo real que as aplicações de definições compartilhadas e objetos de usuário compartilhados 226,228 compartilham entre os nós são definições compartilhadas como regras para implantar em resposta a eventos de sistema como um acionador de alarme e objetos de usuário como nomes, senhas e temas de usuários. Esse tipo de dados ("dados de Consistência") é compartilhado entre os nós com o uso do protocolo de Consistência 216; de modo geral, os dados de Consistência são dados que não precisam ser compartilhados em tempo real ou em ordem total, e que são armazenados de modo persistente por cada um dos nós.

Entretanto, nas modalidades alternativas (não representadas), os dados de Consistência podem ser armazenados de modo não persistente.

[00172] A Figura 3 mostra um diagrama de sequência de UML 300 no qual os dados de Consistência na forma de definições de usuário são compartilhados entre os primeiro e segundo usuários 302a,b (coletivamente, "usuários 302"). Os usuários 302, os primeiro e segundo clientes 102a,b, e os primeiro e segundo servidores 104a,b, que são os primeiro e segundo nós nesse exemplo, são objetos no diagrama 300. Os servidores 104a,b fazem parte do mesmo cluster 108a. Já que os servidores 104a,b com os quais os clientes 102a,b se comunicam não são conectados diretamente um ao outro, o protocolo de Consistência 216 é usado para transferir os dados entre os dois servidores 104a,b, e portanto os dois usuários 302. Embora a modalidade retratada descreva as definições de compartilhamento, em uma modalidade alternativa (não representada), os usuários 302 podem compartilhar de forma analógica os objetos de usuário.

[00173] O diagrama 300 tem dois quadros 332a,b. No primeiro quadro 332a, o primeiro usuário 302a instrui o primeiro cliente 102a a abrir um painel de definições (mensagem 304), e o cliente 102a realiza subsequentemente o procedimento SettingsOpenView() (mensagem 306), que transfere as definições para o primeiro servidor 104a. Simultaneamente, o segundo usuário 302b instrui o segundo cliente 102b de forma analógica (mensagens 308 e 310). No segundo quadro 332b, os usuários 302 simultaneamente editam suas definições. O primeiro usuário 302a edita suas definições fazendo com que primeiro cliente 102a execute UTEditSetting() (mensagem 312), após o qual o primeiro cliente 102a atualiza as definições armazenadas no primeiro servidor 104a fazendo com que o primeiro servidor 104a execute SettingsUpdateView() (mensagem 314). O primeiro servidor 104a então executa ConsistencySetEntries() (mensagem 316), que realiza o

procedimento SetEntries e que transfere as definições inseridas pelo primeiro usuário 302a para o segundo servidor 104b. O segundo servidor 104b então envia as definições transferidas para o segundo cliente 102b executando-se SettingsNotifyViewUpdate() (mensagem 318), após o qual o segundo cliente 102b atualiza o segundo usuário 302b (mensagem 320). Simultaneamente, o segundo usuário 302b modifica de forma analógica as definições e envia aquelas definições para o primeiro servidor 104a com o uso do protocolo de Consistência 216 (mensagens 322, 324, 326, 328, e 330). Cada um dos servidores 104a,b armazena de modo persistente as definições de usuário de modo que elas não precisem ser resincronizadas entre os servidores 104a,b se qualquer um dos servidores 104a,b reinicializar.

APLICAÇÃO DE ALARMES E EVENTOS COMPARTILHADOS 224

[00174] Durante a operação do sistema 100, as informações em tempo real geradas durante tempo de execução são transferidas entre os nós de um cluster 108. Os exemplos dessas informações em tempo real que a aplicação de alarmes e eventos compartilhados 224 compartilha entre os nós são o estado de alarme (isto é, se um alarme já foi acionado em algum lugar no sistema 100); eventos de sistema como a detecção de movimento, se um dispositivo (como uma dentre as câmeras de nó 106) está enviando dados digitais para o restante do sistema 100, se um dispositivo (com um detector de movimento) está conectado ao sistema 100, se um dispositivo está atualmente gravando, se um alarme ocorreu ou foi reconhecido pelos usuários 302, se um dos usuários 302 está realizando uma auditoria no sistema 100, se um dos servidores 104 tiver sofrido um erro, se um dispositivo conectado ao sistema tiver sofrido um erro, se uma transação de texto de ponto de venda tiver ocorrido; e notificações de nó de servidor a cliente como se as definições/dados mudaram, estado de gravação atual, se uma linha de tempo está sendo atualizada, e resultados de consulta de banco de

dados. Na presente modalidade, os dados transferidos entre os nós com o uso do protocolo de Sincronia 214 são chamados de "dados de Sincronia", são gerados no tempo de execução, e não são salvos de modo persistente pelos nós.

[00175] A Figura 4 mostra um diagrama de sequência de UML 400 na qual uma notificação de alarme é compartilhada entre os servidores 104 com o uso do protocolo de Sincronia 214. Os objetos no diagrama 400 são um dentre as câmeras de não nó 114, os três servidores 104 no primeiro cluster 108a, e o segundo cliente 102b, que está conectado a um dos servidores 104c no primeiro cluster 108a.

[00176] Nos primeiros três quadros 402 do diagrama 400, cada um dos servidores 104 se une a um anel de Sincronia nomeado "Serverstate" de modo que o estado de qualquer um dos servidores 104 possa se comunicar a qualquer um dos outros servidores 104; na modalidade retratada, o estado que será comunicado é "AlarmStateTriggered", que significa que um alarme em um dos servidores 108 foi acionado em virtude de um evento que a câmera de não nó 114 detectou. No quadro 404, o segundo servidor 104b é eleito o "mestre" para a aplicação de alarmes; isso significa que é o segundo servidor 104b que determina se a entrada da câmera de não nó 114 satisfaz os critérios para transição para o estado AlarmStateTriggered, e que envia para os outros servidores 104a,c no anel de Sincronia uma mensagem para também realizar a transição dos mesmos para o estado AlarmStateTriggered.

[00177] O segundo usuário 302b realiza o logon no terceiro servidor 104c após os servidores 104 unirem o anel de Sincronia Serverstate (mensagem 406). Subsequente à realização de logon do usuário 302b, o terceiro servidor 104c se une a outro anel de Sincronia nomeado "ClientNotification" (mensagem 408); conforme discutido em maiores detalhes abaixo, esse anel é usado para comunicar estados de sistema ao

usuário 302b, enquanto que o anel de Sincronia Serverstate é usado para se comunicar somente entre os servidores 104. A câmera de não nó 114 envia uma entrada digital, como uma indicação de que uma porta ou janela foi aberta, para o primeiro servidor 104a (mensagem 410), após o qual o primeiro servidor 104a verifica para ver se essa entrada digital satisfaz um conjunto de regras usado para determinar se deve acionar um alarme no sistema 100 (mensagem 412). Na modalidade retratada, o primeiro servidor 104a determina que um alarme deveria ser acionado, e consequentemente executa o AlarmTrigger() (mensagem 414), que alerta o segundo servidor 104b para mudar estados. O segundo servidor 104 realiza a transição de estados para AlarmStateTriggered (mensagem 416) e envia uma mensagem para o anel de Sincronia Serverstate que instrui os outros dois servidores 104a,c para também mudarem os estados para AlarmStateTriggered (quadro 418). Após instruir os outros servidores 104a,c, o segundo servidor 104b executa AlarmTriggerNotification() (mensagem 420), que faz com que o segundo servidor 104b também ingresse ao anel de Sincronia de ClientNotification (quadro 422) e passa uma mensagem para o anel de Sincronia de ClientState que faz com que o terceiro servidor 104c, que é o outro servidor no anel de sincronia de ClientState, realize a transição para um estado "NotifyAlarmTriggered" (quadro 424). Uma vez que o terceiro servidor 104c muda para esse estado, o mesmo informa diretamente ao segundo cliente 102b que o alarme foi acionado, o qual retransmite essa mensagem para o segundo usuário 302b e aguarda pelo segundo usuário 302b reconhecer o alarme (mensagens 426). Uma vez que o segundo usuário 302b reconhece o alarme, o segundo servidor 104b muda, consequentemente, os estados para "AlarmStateAcknowledged" (mensagem 428), e então envia uma mensagem para o anel de Sincronia Serverstate de modo que os outros dois servidores 104a,c também mudem o estado de forma correspondente (quadro 430). O segundo servidor 104b subsequentemente muda o estado

novamente para "NotifyAlarmAcknowledged" (mensagem 432) e envia uma mensagem para o terceiro servidor 104c por meio do anel de Sincronia de ClientNotification para fazer com que o mesmo mude estado de forma correspondente (quadro 434). O terceiro servidor 104c então notifica o cliente 102c que o sistema 100 reconheceu o alarme (mensagem 436), que retransmite essa mensagem para o segundo usuário 302b (mensagem 438).

[00178] Em uma modalidade alternativa (não representada) na qual o segundo servidor 104b falha e não pode mais atuar como o mestre para o anel de sincronia, o sistema 100 elege automaticamente outro dentre os servidores 104 para atuar como o mestre para o anel. O mestre do anel de Sincronia é o único servidor 104 que tem permissão para fazer com que todos os outros nós no anel mudem o estado quando o anel de Sincronia for usado para compartilhar notificações de alarme dentre os nós.

[00179] A Figura 7 mostra uma visualização exemplificativa 700 apresentada aos usuários 302 quando se reconhece um estado de alarme de acordo com o diagrama 400 da Figura 4. A visualização 700 inclui painéis de vídeo 702a-c (coletivamente "painéis 702") que mostram vídeo de transmissão em tempo real da câmera de não nó 114; alertas 704 que indicam que um alarme foi acionado como um resultado do que a câmera de não nó 114 está gravando; e um botão de reconhecimento 706 que o segundo usuário 302b clica para reconhecer que o alarme foi acionado.

APLICAÇÃO DE COLABORAÇÃO E VISUALIZAÇÕES COMPARTILHADAS 222

[00180] Os usuários 302 do sistema 100 também podem desejar compartilhar suas visualizações 700 e colaborar, tal como enviando-se um ao outro mensagens e falando entre si através do sistema 100, enquanto compartilham visualizações 700. Essa aplicação de colaboração e

visualizações compartilhadas 222 permite, consequentemente, que os usuários 302 compartilhem dados como o estado de visualização e notificações de servidor para cliente como mensagens de usuário e compartilhem solicitações. Esse tipo de dados representa dados de Sincronia que são compartilhados em tempo real.

[00181] A Figura 5 mostra um diagrama de sequência de UML 500 no qual as visualizações 700 são compartilhadas entre os usuários 302 com o uso do protocolo de Sincronia 214. O diagrama 500 inclui seis objetos: os primeiro e segundo usuários 302a,b, os primeiro e segundo clientes 102a,b aos quais o primeiro e segundo usuários 302a,b estão respectivamente conectados, e os primeiro e segundo servidores 104a,b aos quais os primeiro e segundo clientes 102a,b estão, respectivamente, conectados.

[00182] O primeiro usuário 302a realiza o logon no primeiro servidor 104a por meio do primeiro cliente 102a (mensagem 502), após o qual o primeiro servidor 104a ingressa ao anel de sincronia de ClientNotification (quadro 504). De forma semelhante, o segundo usuário 302b realiza o logon no segundo servidor 104b por meio do segundo cliente 102b (mensagem 506), após o qual o segundo servidor 104b também ingressa ao anel de sincronia de ClientNotification (quadro 508).

[00183] O primeiro usuário 302a então instrui o primeiro cliente 102a que ele deseja compartilhar sua visualização 700. O primeiro usuário 302a faz isso clicando-se em um botão de compartilhamento (mensagem 510), o que faz com que o primeiro cliente 102a abra a visualização 700 a ser compartilhada ("visualização compartilhada 700") no primeiro servidor 104a (mensagem 512). O primeiro servidor 104a cria uma sessão de visualização compartilhada (mensagem 514), e então envia o identificador de sessão para o primeiro cliente 102a (mensagem 516).

[00184] Em um quadro 518, cada um dos clientes 102 ingressa a um anel de Sincronia que os permite compartilhar a visualização compartilhada 700. O primeiro servidor 104a ingressa ao anel de Sincronia SharedViewl no quadro 520. Simultaneamente, o primeiro cliente 106a instrui o primeiro servidor 104a a anunciar para o outro servidor 104b por meio do protocolo de Sincronia 214 que a visualização 700 do primeiro usuário 302a pode ser compartilhada passando-se para o primeiro servidor 104a uma lista de usuário e ao identificador de sessão (mensagem 522). O primeiro servidor 104a faz isso enviando-se uma mensagem para o segundo servidor 104b por meio do anel de Sincronia ClientNotify que faz com que o segundo servidor 104 mude para um estado NotifyViewSession (quadro 524). No estado NotifyViewSession, o segundo servidor 104b faz com que o segundo cliente 106b induza o segundo usuário 302b a compartilhar a visualização 700 do primeiro usuário 302a (mensagens 526 e 528), e a resposta afirmativa do segundo usuário 302b é retransmitida de volta para o segundo servidor 104b (mensagens 530 e 532). O segundo servidor 104b subsequentemente ingressa ao anel de Sincronia SharedViewl (quadro 534), que é usado para compartilhar a visualização 700 do primeiro usuário 302a'.

[00185] Em um segundo quadro 519, cada um dos usuários 106 atualiza a visualização compartilhada 700, e as atualizações são compartilhadas automaticamente entre si. O primeiro usuário 302a realiza o zoom em um primeiro painel 702a na visualização compartilhada 700 (mensagem 536), e o primeiro cliente 102a retransmite para o primeiro servidor 104a como o primeiro usuário 302a realizou o zoom no primeiro painel 702a (mensagem 538). O primeiro servidor 104a compartilha as particularidades do zoom com o segundo servidor 104b passando-se as mesmas ao longo do anel de Sincronia SharedViewl (quadro 540). O segundo servidor 104b consequentemente atualiza a visualização

compartilhada 700 conforme exibido no segundo cliente 106b (mensagem 542), e a visualização compartilhada atualizada 700 é então exibida para o segundo usuário 302b (mensagem 544). Simultaneamente, o segundo usuário 302b realiza o movimento panorâmico em um segundo painel 702b na visualização compartilhada 700 (mensagem 546), e o segundo cliente 102b retransmite para o segundo servidor 104b como o segundo usuário 302b realizou o movimento panorâmico desse painel 702b (mensagem 548). O segundo servidor 104b então compartilha as particularidades de movimento panorâmico com o primeiro servidor 104a passando-os com o uso do anel de Sincronia SharedViewl (quadro 550). O primeiro servidor 104a consequentemente atualiza a visualização compartilhada 700 conforme exibido no primeiro cliente 106b (mensagem 552), e a visualização compartilhada atualizada 700 é então exibida ao primeiro usuário 302a (mensagem 556).

[00186] Após o segundo quadro 519, o primeiro usuário 302a fecha sua visualização 700 (mensagem 556), que é retransmitida para o primeiro servidor 104a (mensagem 558). O primeiro servidor 104a consequentemente deixa o anel de Sincronia SharedViewl (mensagem e quadro 560). O segundo usuário 302b fecha de forma semelhante sua visualização 700, o que faz com que o segundo servidor 104b deixe o anel de Sincronia SharedViewl (mensagens 562 e 564, e mensagem e quadro 566).

[00187] No exemplo da Figura 5, os usuários 302 realizam o movimento panorâmico e zoom da visualização compartilhada 700. Nas modalidades alternativas (não representadas) os usuários 302 podem modificar a visualização compartilhada 700 de outras formas. Por exemplo, cada um dos usuários 302 pode mudar o layout dos painéis 702; escolher se o vídeo deve ser exibido ao vivo ou em modo de reprodução, caso no qual os usuários 302 também têm a capacidade para, iniciar a reprodução, ou avançar o vídeo; e exibir os objetos de

usuário como mapas ou páginas da web junto com informações sobre o objeto de usuário como um histórico de revisão. Nessas modalidades alternativas, os exemplos de informações de dados adicionais que são sincronizados com o uso de um anel de Sincronia incluem se um vídeo está sendo reproduzido, pausado, ou avançado e o histórico de revisão do objeto de usuário.

[00188] Embora a discussão acima foque na implantação da aplicação de colaboração e visualizações compartilhadas 222 no sistema de segurança físico ponto a ponto 100 da Figura 1, mais genericamente essa aplicação 222 pode ser implantada em um sistema de segurança físico que tem múltiplos servidores 104, como um sistema federado que inclui um servidor de porta centralizada. Um exemplo dessa modalidade mais geral é mostrado na Figura 12, que representa um método exemplificativo 1200 para compartilhar a visualização com o uso de um sistema de segurança físico que compreende uma pluralidade de nós de servidor. O método 1200 começa no bloco 1202 e prossegue para o bloco 1204, em que os dados de estado de visualização representativos da visualização exibida pelo primeiro cliente (como o primeiro cliente 102a), que é a visualização a ser compartilhada, são enviados do primeiro cliente para um primeiro nó de servidor (como o primeiro servidor 104a e os dados de estado de visualização enviados por meio da mensagem 538). No bloco 1206 os dados de estado de visualização são retransmitidos do primeiro nó de servidor para um segundo cliente (como o segundo cliente 102b) por meio de um segundo nó de servidor (como o segundo servidor 104b e os dados de estado de visualização enviados por meio do quadro 540 e mensagem 542). No bloco 1208 o segundo cliente então atualiza um visor com o uso dos dados de estado de visualização para mostrar a visualização compartilhada (como por meio da mensagem 544). Em resposta a uma mudança na visualização compartilhada no segundo

cliente, como uma mudança resultante da interação com um usuário no segundo cliente (como por meio da mensagem 546), no bloco 1210 os dados de estado de visualização atualizados são enviados do segundo cliente ao segundo nó de servidor (como por meio da mensagem 548). Os dados de estado de visualização atualizados são representativos da visualização compartilhada conforme exibido pelo segundo cliente. Os dados de estado de visualização atualizados são enviados do segundo nó de servidor ao primeiro cliente por meio do primeiro nó de servidor no bloco 1212 (como por meio do quadro 550 e mensagem 552), e no bloco 1214 o visor do primeiro cliente é então atualizado para mostrar a visualização compartilhada conforme foi modificada no segundo cliente com o uso dos dados de estado de visualização atualizados (como por meio da mensagem 554). O método 1200 termina no bloco 1216. Em uma modalidade alternativa, como quando se lida com um sistema federado que usa um servidor de porta centralizada, todos os dados de estado de visualização podem ser roteados através daquele servidor centralizado.

APLICAÇÃO DE COMPARTILHAMENTO DE VISUALIZAÇÃO AUTÔNOMA 225

[00189] Os usuários 302 do sistema 100 também podem desejar ter a capacidade de ver e controlar uma visualização em um visor que é conectado direta ou indiretamente a um dos servidores 104 que os usuários 302 não controlam diretamente (isto é, que os usuários 302 controlam por meio de outros servidores 104) (esse visor é um "visor não assistido", e a visualização no visor não assistido é a "visualização não assistida"). Por exemplo, o visor não assistido pode ser montado em uma parede na frente dos usuários 302 e ser conectado ao cluster de servidor 108 por meio de um dos servidores 104 no cluster 108, enquanto os usuários 302 podem ser conectados ao cluster de servidor 108 por meio de outros servidores 104 no cluster 108. Conforme discutido abaixo em

relação à Figura 10, a aplicação de compartilhamento de visualização não assistida 225 permite que os usuários 302 visualizem e controlem a visualização não assistida não obstante que nenhum dos usuários 302 esteja conectado diretamente ao servidor 104 que controla a visualização não assistida. Os dados de visualização trocados entre os servidores 104 para habilitar essa funcionalidade são dados de sincronia que são compartilhados em tempo real.

[00190] A Figura 10 mostra um diagrama de sequência de UML 1000 no qual a visualização não assistida é compartilhada com o primeiro usuário 302a com o uso do protocolo de Sincronia 214. O diagrama 1000 inclui seis objetos: o primeiro usuário 302a, o primeiro cliente 102a ao qual o primeiro usuário 302a está conectado e que inclui um visor ("visor de cliente") com o qual o primeiro usuário 302a interage, os primeiro e segundo servidores 104a,b, uma instância de monitor 1004 em execução no hardware com um cliente não assistido dentre um dos clientes 102 conectados tanto ao segundo servidor 104b quanto ao visor não assistido, e um administrador 1002 que configura a instância de monitor 1004. Em uma modalidade alternativa (não representada), o visor não assistido pode ser conectado diretamente ao segundo servidor 104b e a instância de monitor 1004 pode executar no segundo servidor 104b.

[00191] Na Figura 10, o administrador 1002 cria a instância de monitor 1004 (mensagem 1006) e a instância de monitor 1004 então realiza automaticamente o logon no segundo servidor 104b (mensagens 1008 e 1010). A instância de monitor 1004 torna a visualização não assistida disponível ao segundo servidor 104b executando-se SharedViewOpen(viewState) no segundo servidor 104, em que viewState são dados de estado de visualização indicativos da visualização não assistida (mensagem 1012). Após isso, o segundo servidor 104b cria uma sessão de visualização compartilhada (mensagem 1014) executando-se SharedViewSessionCreate() e então

envia o identificador de sessão correspondente à instância de monitor (mensagem 1016). Após receber o identificador de sessão, a instância de monitor 1004 ingressa ao anel de Sincronia SharedViewl (quadro 1018), que é usado para transmitir dados de estado de visualização para e a partir dos outros servidores 104 no cluster 108 que também são membros do anel de Sincronia SharedViewl.

[00192] Após unir o anel de Sincronia SharedViewl, a instância de monitor 1020 publica uma notificação aos outros servidores 104 no cluster 108 que a visualização não assistida está disponível para ser vista e controlada. A instância de monitor 1020 faz isso executando-se RegisterMonitor(sessionId) no segundo servidor 104b (mensagem 1018), que faz com que o identificador de sessão relacionado à visualização não assistida a ser registrada em um diretório de visualização (quadro 1022). O diretório de visualização é compartilhado com os outros servidores 104 no cluster 108 com o uso do protocolo de Consistência 216.

[00193] Uma vez que o diretório de visualização é disseminado aos outros servidores 104 no cluster 108, aqueles outros servidores 104 podem acessar o diretório de visualização para determinar quais visualizações não assistidas estão disponíveis para visualizar e controlar. Após o primeiro servidor 104a receber o diretório de visualização, o primeiro usuário 302a, por meio do primeiro cliente 102a, realiza o logon no primeiro servidor 104a, ganhando através disso o acesso ao cluster 108 (mensagens 1024) e ao diretório de visualização. O primeiro usuário 102a instrui o primeiro cliente 102a a exibir a visualização não assistida executando-se UIDisplayMonitor(sessionId) (mensagem 1026), o que faz com que o primeiro cliente 102a envie o identificador de sessão da visualização não assistida para o primeiro servidor 104a com instruções para abrir a visualização não assistida (mensagem 1028). O primeiro servidor 104a reconhece as instruções

do primeiro cliente 102a (mensagem 1030) e então ingressa ao anel de Sincronia SharedViewl (quadro 1032) a fim de receber automaticamente os dados de estado de visualização que descrevem a visualização atual do visor não assistido (mensagem 1034) e para ficar automaticamente informado de quaisquer mudanças subsequentes para a visualização não assistida.

[00194] O primeiro usuário 302a realiza o movimento panorâmico de forma subsequente de um dos painéis da visualização não assistida conforme é exibido no visor de cliente (mensagem 1036), e o primeiro cliente 102a retransmite a ação de movimento panorâmico e a identidade do painel particular para o qual se realiza o movimento panorâmico ao primeiro servidor 104a executando-se SharedViewUpdate(action=pan, panelId=2) (mensagem 1038). O primeiro servidor 104a envia dados de estado de visualização atualizados para todos os servidores 104 que são membros do anel de Sincronia SharedViewl (quadro 1040), que permite que todos aqueles servidores 104 reproduzam a versão atualizada da visualização não assistida. O segundo servidor 104b recebe esses dados de estado de visualização atualizados e retransmite os mesmos para a instância de monitor 1004 executando-se NotifySharedViewUpdate(action=pan, params, panelId=2) (mensagem 1042). A instância de monitor 1004 então atualiza o visor não assistido para mostrar a visualização não assistida modificado pelo primeiro usuário 302a (mensagem 1044).

[00195] No exemplo da Figura 10, o primeiro usuário 302a realiza o movimento panorâmico de um dos painéis da visualização não assistida. Nas modalidades alternativas (não representadas) o primeiro usuário 302a pode modificar a visualização não assistida de outras formas. Por exemplo, o primeiro usuário 302a pode mudar o layout de qualquer um ou mais dos painéis da visualização não assistida;

escolher se o vídeo deve ser exibido ao vivo ou em modo de reprodução, caso no qual o primeiro usuário 302a também tem capacidade para pausar, reproduzir, ou avançar através do vídeo; e exibir objetos de usuário como mapas ou páginas da web junto com informações sobre o objeto de usuário como histórico de revisão. Nessas modalidades alternativas, os exemplos de informações de dados adicionais que são sincronizadas com o uso de um anel de sincronia incluem se um vídeo está sendo reproduzido, pausado, ou avançar através do e o histórico de revisão do objeto de usuário.

[00196] Em outra modalidade alternativa (não representada), a aplicação de compartilhamento de visualização não assistida 225 pode ser usada para criar um visor de agregado que compreende uma matriz de $n \times m$ visores não assistidos. Por exemplo, quando $n = m = 2$ e há, consequentemente, quatro visores não assistidos, o primeiro usuário 302a pode controlar todos os quatro dentre os visores não assistidos simultaneamente para criar um visor virtual grande. Um único vídeo pode então ser ampliado de modo que cada uma das visualizações não assistidas seja de um quadrante do vídeo, permitindo através disso que o vídeo seja ampliado e mostrado através dos quatro visores não assistidos. Nessa modalidade, as instâncias de monitor 1004 para os visores não assistidos podem ser comunicativas com o cluster de servidor 108 por meio de qualquer um dentre um a quatro dos servidores 104.

[00197] Embora a Figura 10 mostre somente o primeiro usuário 302a, nas modalidades alternativas (não representadas) mais do que um dos usuários 302 pode ver e controlar a visualização não assistida ingressando-se também ao anel de Sincronia SharedViewl. No exemplo acima do visor agregado que compreende a matriz $n \times m$ de visores não assistidos, o visor agregado pode ser montado na sala para a visualização simultânea de diversos dos usuários 302, em que cada um

dos usuários 302 tem capacidade para controlar cada uma das visualizações não assistidas.

[00198] Embora a discussão acima foque na implantação da aplicação de compartilhamento de visualização não assistida 225 no sistema de segurança físico ponto a ponto 100 da Figura 1, mais genericamente essa aplicação 225 pode ser implantada em um sistema de segurança físico que tem múltiplos servidores 104, como um sistema federado que inclui um servidor de porta centralizada. Um exemplo dessa modalidade mais geral é mostrado na Figura 11, que representa um método exemplificativo 1100 para interagir com o visor autônomo em um sistema de segurança físico que compreende múltiplos nós de servidor. O método começa no bloco 1102 e prossegue para o bloco 1104 em que um segundo nó de servidor (como o segundo servidor 104b) que é comunicativo com o visor autônomo envia para um primeiro nó de servidor (como o primeiro servidor 104a) os dados de estado de visualização indicativos da visualização não assistida (como por meio do anel de Sincronia nos quadros 1020 e 1032 da Figura 10). O método 1100 então prossegue para o bloco 1106 em que pelo menos uma porção da visualização não assistida é exibida no visor de cliente (como a atualização do visor de cliente que resulta da mensagem 1034 de Figura 10) após o qual o método 1100 termina no bloco 1108. Em uma modalidade alternativa como quando se lida com um sistema federado que usa um servidor de porta centralizada, todos os dados de estado de visualização podem ser encaminhados através daquele servidor centralizado.

APLICAÇÃO DE TRANSMISSÕES DE CLUSTER 220

[00199] Um dos usuários 302 também pode querer transmitir vídeo a partir de uma das câmeras 106,114 se uma conexão ponto a ponto entre aquele usuário 302 e aquela câmera 106, 114 for indisponível; a aplicação de transmissões de cluster 220 habilita essa funcionalidade.

A Figura 6 mostra um diagrama de sequência de UML 600 no qual o vídeo é transmitido a partir da câmera de não nó 114 para o primeiro usuário 302a através dos primeiros e segundos servidores 104a,b e o primeiro cliente 102a. O diagrama de UML tem cinco objetos: o primeiro usuário 302a, o primeiro cliente 102a, os primeiros e segundos servidores 104a,b, e a câmera de não nó 114. O primeiro cliente 102a pode se comunicar diretamente com o primeiro servidor 104a, mas não pode se comunicar diretamente com o segundo servidor 104b. Entretanto, os primeiros e segundos servidores 104a,b podem se comunicar diretamente entre si. Adicionalmente, enquanto o segundo servidor 104b e a câmera de não nó 114 podem se comunicar diretamente entre si, o primeiro servidor 104a e a câmera de não nó 114 não podem se comunicar diretamente.

[00200] O segundo servidor 104b primeiro estabelece uma sessão com a câmera de não nó 114 de modo que o vídeo seja transmitido a partir da câmera de não nó 114 para o segundo servidor 104b. O segundo servidor 104b primeiro configura uma sessão de Protocolo de Transmissão em Tempo Real (RTSP) com a câmera de não nó 114 (mensagens 602 e 604), e instrui a câmera de não nó 114 para enviar seu vídeo (mensagens 606 e 608). A câmera de não nó 114 começa subsequentemente a transmissão (mensagem 610).

[00201] O primeiro usuário 302a estabelece uma conexão com o primeiro cliente 102a (mensagem 612) e então instrui o primeiro cliente 102a para abrir uma janela que mostra o vídeo em transmissão (mensagem 614). O primeiro cliente 102a então executa LookupRoute() para determinar para qual servidor 104 se conectar (mensagem 616); porque o primeiro cliente 102a não pode se conectar diretamente ao segundo servidor 104b, o mesmo configura uma conexão de RTSP com o primeiro servidor 104a (mensagem 618). O primeiro servidor 104b então executa LookupRoute() para determinar a qual nó se conectar

para acessar o vídeo em tempo real, e determina que o mesmo deveria se conectar ao segundo servidor 104b (mensagem 620). O primeiro servidor 104a configura subsequentemente uma conexão de RTSP com o segundo servidor 104b (mensagem 622), e o segundo servidor 104b retorna um identificador de sessão para o primeiro servidor 104a (mensagem 624). O primeiro servidor 104a retransmite o identificador de sessão ao primeiro cliente 102a (mensagem 626). Com o uso desse identificador de sessão, o primeiro cliente 102a instrui o segundo servidor 104b para começar a reproduzir o vídeo de RTSP (mensagens 628 a 634), e o segundo servidor 104b subsequentemente transmite o vídeo para o primeiro usuário 302a por meio do segundo servidor 104b, então o primeiro servidor 104a, e então o primeiro cliente 102a (mensagens 636 a 640).

[00202] Embora a Figura 6 encaminhe o vídeo a partir da câmera de não nós 114 conectada a um dos servidores 104 em um cluster 108 para outros servidores 104 no mesmo cluster 108, nas modalidades alternativas (não representadas) o vídeo também pode ser encaminhado a partir de uma dentre as câmeras de nó 106 em um cluster 108 através das outras câmeras de nó 106 no mesmo cluster 108.

REINICIALIZAÇÃO

[00203] Na presente modalidade, as informações de associação de cluster são armazenadas de modo persistente e de modo localizado em cada um dos nós. Quando um dos nós reinicializa, o mesmo se reingressa automaticamente ao cluster 108 do qual o mesmo era um membro antes da reinicialização. Isso é representado no método exemplificativo 900 mostrado na Figura 9. Após realizar o bloco 806, um dos nós no cluster 108 reinicializa (bloco 902). Mediante a reinicialização, esse nó acessa as informações de associação de cluster armazenadas de modo persistente que identificam o cluster 108 do qual o mesmo era um membro antes da reinicialização (bloco 904), e

subsequentemente reingressa a esse cluster 108 (bloco 906) antes de retornar ao bloco 808. Ter o reingresso dos nós automaticamente a um cluster 108 após a reinicialização é benéfico pelo fato de que isso ajuda o sistema 100 a se recuperar após a reinicialização de qualquer um ou mais de seus servidores. Já que cada um dos nós armazena de modo persistente as informações de Consistência, mediante o reingresso ao cluster 108 somente aquelas informações de Consistência que foram alteradas desde que o nó deixou o cluster 108 são sincronizadas novamente, economizando, desse modo, a largura de banda.

[00204] Embora determinadas modalidades exemplificativas sejam representadas, as modalidades alternativas, que não são representadas, são possíveis. Por exemplo, enquanto na modalidade retratada as câmeras de nó 106 e as câmeras de não nó 114 são distintas entre si, nas modalidades alternativas (não representadas) uma única câmera pode ser simultaneamente uma câmera de nó e uma câmera de não nó. Por exemplo, na Figura 1, a primeira câmera 106a é um nó que é um membro do terceiro cluster 108c; entretanto, se a primeira câmera 106a também foi diretamente acoplada ao quinto servidor 104e, mas somente reteve suas informações de associação de cluster para o terceiro cluster 108c, a primeira câmera 106a permaneceria um membro do terceiro cluster 108c enquanto atua simultaneamente como uma câmera de não nó 114 a partir da perspectiva do quinto servidor 104e.

[00205] O processador usado nas modalidades anteriores pode ser, por exemplo, um microprocessador, microcontrolador, controlador de lógica programável, arranjo de porta programável em campo, ou um circuito integrado de aplicação específica. Os exemplos da mídia legível por computador são não transitórios e incluem mídia à base de disco como CD-ROMs e DVDs, mídia magnética como discos rígidos e outras formas de armazenamento de disco magnético, mídia à base de semicondutor como mídia flash, memória de acesso aleatório, e

memória somente leitura.

[00206] Contempla-se que qualquer parte de qualquer aspecto ou modalidade discutida neste relatório descritivo pode ser implantada ou combinada com qualquer parte de qualquer outro aspecto ou modalidade discutida neste relatório descritivo.

[00207] Por motivos de conveniência, as modalidades exemplificativas acima são descritas como vários blocos funcionais interconectados. Isso não é necessário, entretanto, e pode haver casos em que esses blocos funcionais são agregados de forma equivalente em um único dispositivo lógico, programa ou operação com limites não claros. Em qualquer caso, os blocos funcionais podem ser implantados por eles próprios, ou em combinação com outras peças de hardware ou software.

[00208] Embora modalidades particulares tenham sido descritas anteriormente, deve-se entender que outras modalidades são possíveis e são destinadas a serem incluídas no presente documento. Será claro para qualquer indivíduo versado na técnica que as modificações das modalidades anteriores e ajustes às mesmas, não mostradas, são possíveis.

REIVINDICAÇÕES

1. Método para compartilhar dados em um sistema físico de segurança (100) que compreende uma pluralidade de nós de servidor (106), sendo que o método é **caracterizado por**:

(a) acessar, com o uso de um primeiro nó dos nós de servidor (106), um identificador de nó que identifica um segundo nó dos nós de servidor (106), em que os primeiro e segundo nós compreendem pelo menos parte de um cluster (108) de servidor e em que o identificador de nó compreende pelo menos parte de informações de associação de cluster que identificam todos e acessíveis por todos os nós de servidor (106) no cluster (108) de servidor; e

(b) enviar os dados do primeiro nó ao segundo nó, e
 (c) adicionar um novo nó de servidor ao cluster (108) de servidor, a adição compreendendo:

(i) trocar uma versão das informações de associação de cluster armazenadas no novo nó de servidor com a nova versão das informações de associação de cluster armazenada em um nó de controle de assinatura dos nós de servidores que já são parte do cluster (108) de servidor; e

(ii) sincronizar a versão das informações de associação de cluster armazenadas no novo nó de servidor com as versões das informações de associação de cluster armazenadas, antes da junção do novo nó de servidor no cluster (108) de servidor, em todos os nós de servidor (106) no cluster (108) de servidor.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de** que:

(a) o cluster (108) de servidor compreende pelo menos três nós de servidor; ou

(b) o método compreende:

(i) acessar, com o uso do segundo nó, um identificador de nó

que identifica o primeiro nó, e

(ii) enviar dados adicionais a partir do segundo nó ao primeiro nó.

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de** que os nós de servidor (106) compreendem câmeras, gravadores de vídeo em rede, e servidores de controle de acesso.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de** que as informações de associação de cluster compreendem:

(a) um identificador de nó que identifica exclusivamente cada um dos nós de servidor (106) no cluster (108) de servidor; e

(b) um identificador de cluster que identifica exclusivamente o cluster (108) de servidor ao qual os nós de servidor (106) pertencem.

5. Método, de acordo com a reivindicação 4, **caracterizado pelo fato de** que cada um dos nós de servidor (106) no cluster (108) de servidor armazena de modo persistente sua própria versão das informações de associação de cluster de modo localizado.

6. Método, de acordo com a reivindicação 5, **caracterizado pelo fato de** que compreende adicionalmente:

(a) reinicializar um dos nós de servidor (106) no cluster (108) de servidor; e

(b) uma vez que o nó de servidor reinicializado retorna online, com o uso do nó de servidor reinicializado para executar um método que compreende:

(i) acessar o identificador de cluster que identifica o cluster (108) de servidor; e

(ii) reunir automaticamente o cluster (108) de servidor.

7. Método, de acordo com a reivindicação 5, **caracterizado pelo fato de** que enviar os dados compreende:

(a) consultar, com o uso do primeiro nó, um ponto de

extremidade para o segundo nó a partir do identificador de nó; e

(b) enviar os dados do primeiro nó ao ponto de extremidade de comunicação,

em que o ponto de extremidade de comunicação e o identificador de nó compreendem entradas em um mapa de rede relacionado identificadores de nó para todos os nós de servidor (106) no cluster (108) de servidor aos pontos de extremidade de comunicação correspondentes, e em que cada um dos nós de servidor (106) no cluster (108) de servidor armazena de modo persistente sua própria versão do mapa de rede de modo localizado.

8. Método, de acordo com a reivindicação 7, **caracterizado pelo fato de** que o mapa de rede permite que cada um dos nós de servidor (106) no cluster (108) de servidor envie os dados a qualquer outro dos nós de servidor (106) no cluster (108) de servidor sem usar um servidor centralizado.

9. Método, de acordo com a reivindicação 5, **caracterizado pelo fato de** que os dados são armazenados de modo localizado no primeiro nó e que compreendem adicionalmente modificar os dados com o uso do primeiro nó, em que enviar os dados do primeiro nó ao segundo nó compreende parte de sincronização dos dados nos primeiro e segundo nós depois que o primeiro nó modificou os dados, e

em que os dados compreendem informações de versão geradas com o uso de um mecanismo de controle de versão de casualidade e diferentes versões dos dados são armazenadas nos primeiro e segundo nós, e em que sincronizar os dados compreende comparar as informações de versão armazenadas nos primeiro e segundo nós e adotar tanto no primeiro quanto no segundo nós que os dados cujas informações de versão indicam são mais recentes.

10. Método, de acordo com a reivindicação 5, **caracterizado pelo fato de** que os dados compreendem o identificador de nó do

primeiro nó, informações de estado de batimento cardíaco do primeiro nó, informações de estado de aplicação do primeiro nó, e informações de versão, e em que enviar os dados compreende disseminar os dados a todos os nós de servidor (106) no cluster (108) de servidor com o uso de um protocolo gossip que executa trocas de dados entre pares dos nós de servidor (106) no cluster (108) de servidor.

11. Método, de acordo com a reivindicação 10, **caracterizado pelo fato de** que:

(a) os dados são periodicamente disseminados a todos os nós de servidor (106) no cluster (108) de servidor, ou

(b) os dados são enviados ao segundo nó quando o primeiro nó se une ao cluster (108) de servidor.

12. Método, de acordo com a reivindicação 10, **caracterizado pelo fato de** que um domínio preenchido com entradas que podem ser modificadas por qualquer um dos nós de servidor (106) no cluster (108) de servidor é armazenado de modo localizado em cada um dos nós no cluster (108) de servidor, e que compreende adicionalmente gerar as informações de versão com o uso de um mecanismo de controle de versão de causalidade de modo que as informações de versão indiquem qual dos nós de servidor (106) modificou mais recentemente uma das entradas.

13. Método, de acordo com a reivindicação 12, **caracterizado pelo fato de** que as informações de estado de aplicação compreendem um hash de nível de topo gerado realizando-se hashing de todas as entradas no domínio, e compreendem adicionalmente:

(a) comparar, com o uso do segundo nó, o hash de nível de topo com um hash de nível de topo gerado realizando-se hashing de uma versão de um domínio correspondente armazenado de modo localizado no segundo nó; e

(b) se os hashes de nível de topo diferirem, sincronizar os

domínios tanto no primeiro quanto no segundo nós com o uso das informações de versão.

14. Método, de acordo com a reivindicação 10, **caracterizado pelo fato de** que uma entrada de estado que pode ser somente modificada pelo primeiro nó é armazenada de modo localizado no primeiro nó, e em que as informações de versão compreendem um número de versão que o primeiro nó incrementa sempre que o mesmo modifica a entrada de estado,

e em que as informações de estado de aplicação compreendem um par de entidade de estado que compreende um identificador de entidade de estado que identifica a entrada de estado e o número de versão.

15. Método, de acordo com a reivindicação 14, **caracterizado pelo fato de** que compreende adicionalmente:

(a) comparar, com o uso do segundo nó, o número de versão recebido do primeiro nó com um número de versão de uma entrada de estado correspondente armazenada de modo localizado no segundo nó; e

(b) se os números de versões diferirem, atualizar a entrada de estado armazenada de modo localizado no segundo nó com a entrada de estado armazenada de modo localizado no primeiro nó.

16. Método, de acordo com a reivindicação 15, **caracterizado pelo fato de** que atualizar a entrada de estado compreende enviar do primeiro nó ao segundo nó entradas de estado adicional armazenadas de modo localizado no primeiro nó que foram modificadas simultaneamente com a entrada de estado.

17. Método, de acordo com a reivindicação 5, **caracterizado pelo fato de** que:

(a) os primeiro e segundo nós compreendem pelo menos parte de um grupo de nós de servidor (106) no cluster (108) de servidor

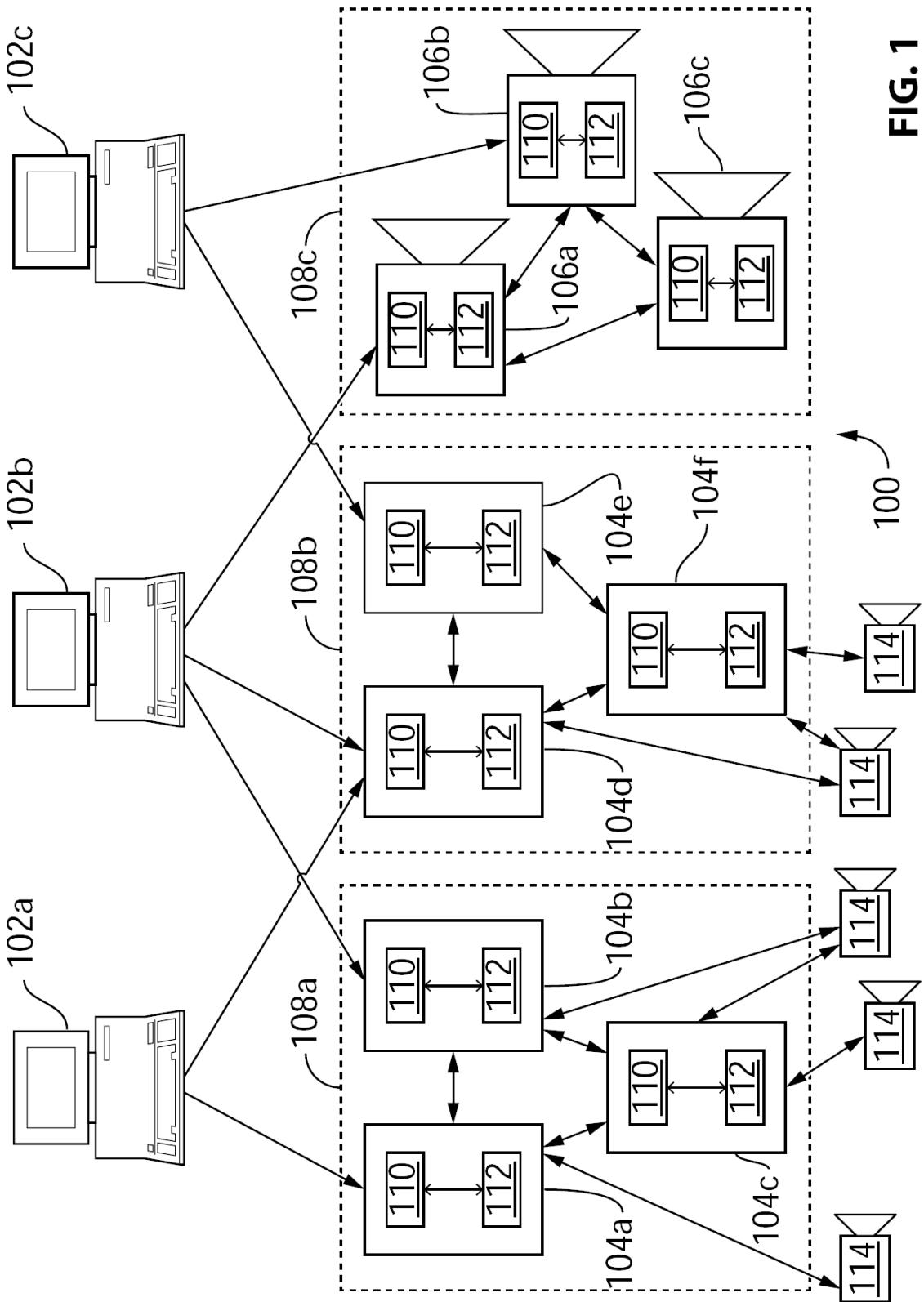
aos quais o primeiro nó pode enviar os dados de maneira totalmente ordenada a todos dos nós de servidor (106) no grupo, e em que enviar os dados compreende o primeiro nó que envia os dados a todos dos nós de servidor (106) no grupo; ou

(b) os dados compreendem transmitir por streaming vídeo transmitido por streaming de outro dos nós de servidor (106) no cluster (108) de servidor através do primeiro nó ao segundo nó.

18. Método, de acordo com a reivindicação 17, **caracterizado pelo fato de** que os dados compreendem dados não persistentes gerados durante a execução do sistema de segurança físico.

19. Sistema de segurança físico (100), **caracterizado pelo fato de** que o sistema compreende uma pluralidade de nós de servidor (106) que compreendem um primeiro nó e um segundo nó, em que o primeiro nó compreende um processador acoplado de modo comunicativo a um meio legível em computador que codificou no mesmo demonstrações e instruções para fazer com que o processador execute o método como definido em qualquer uma das reivindicações 1 a 18.

20. Meio legível em computador não transitório **caracterizado pelo fato de** que codificou no mesmo demonstrações e instruções para fazer com que um processador (110) execute um método para compartilhar dados em um sistema de segurança físico (100) que compreende uma pluralidade de nós de servidor (106) como definido em qualquer uma das reivindicações 1 a 18.



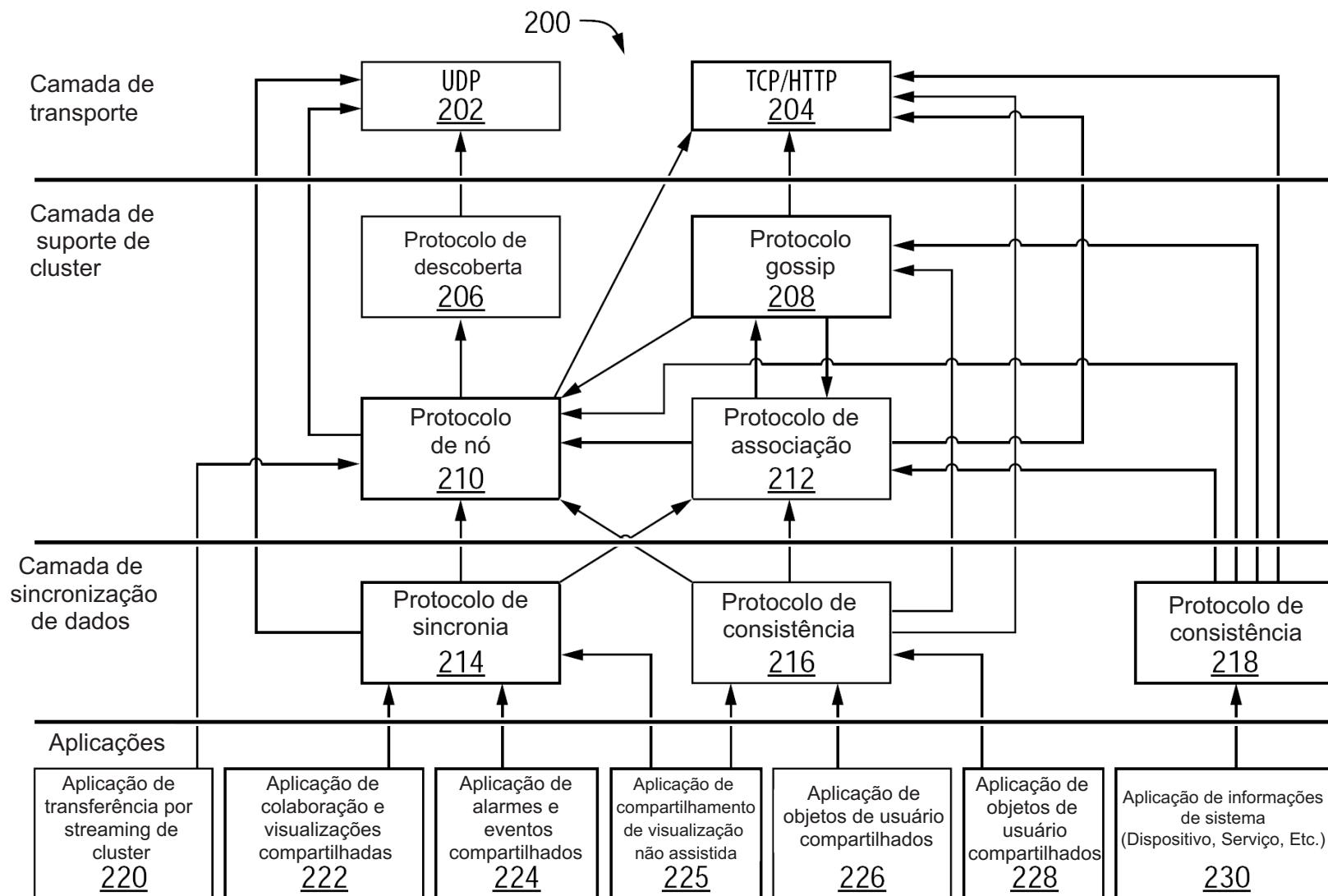
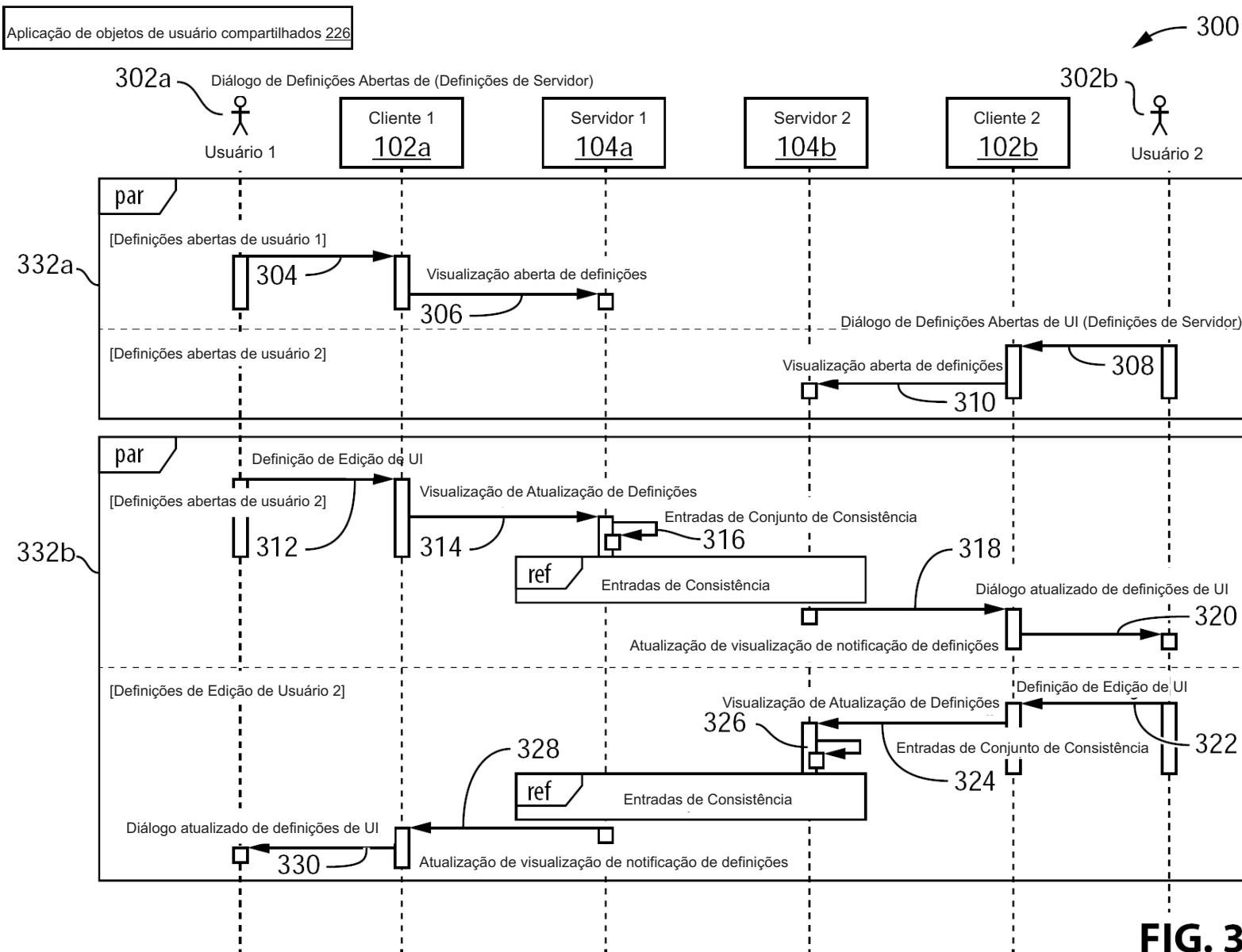
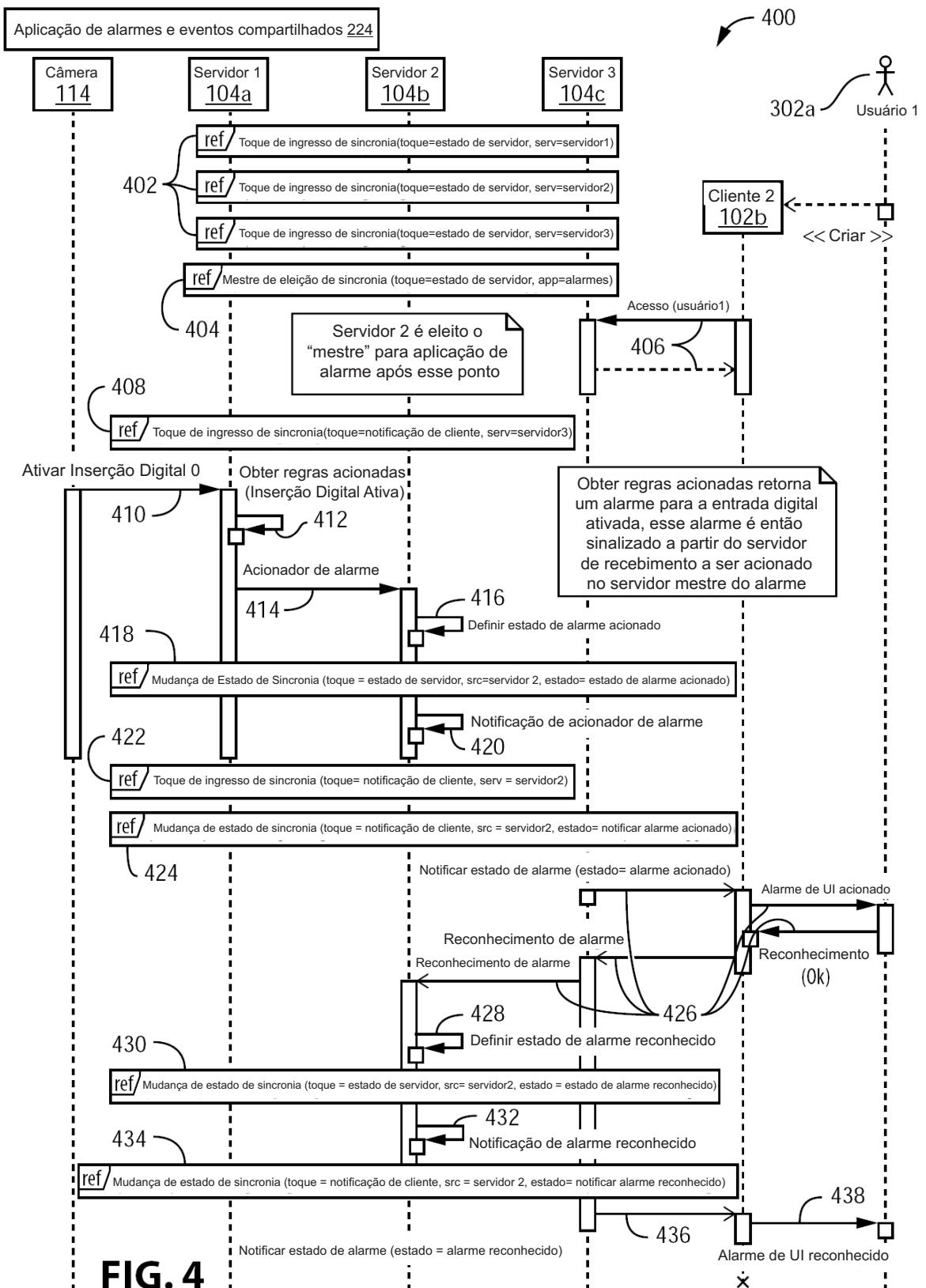


FIG. 2

**FIG. 3**



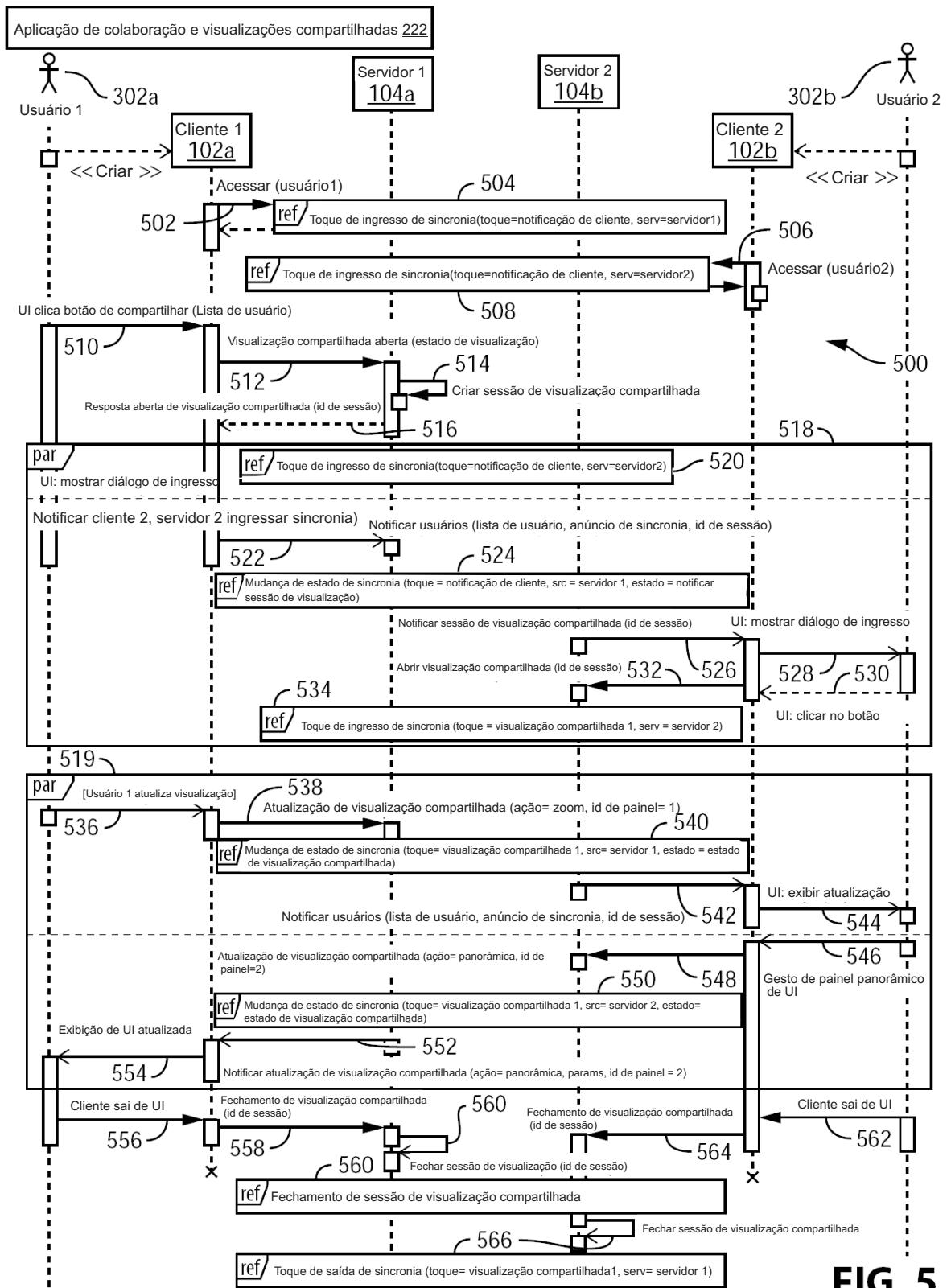
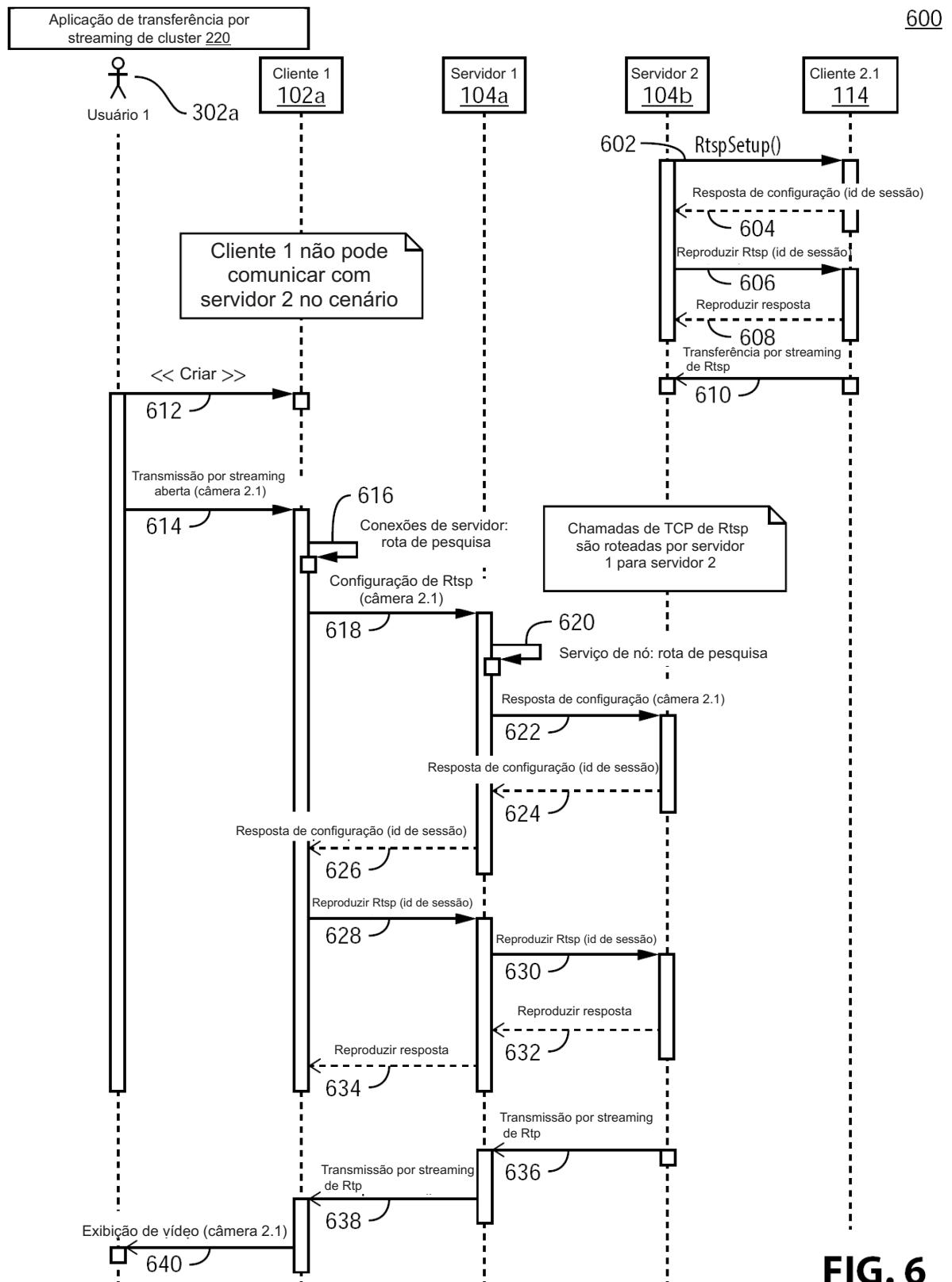
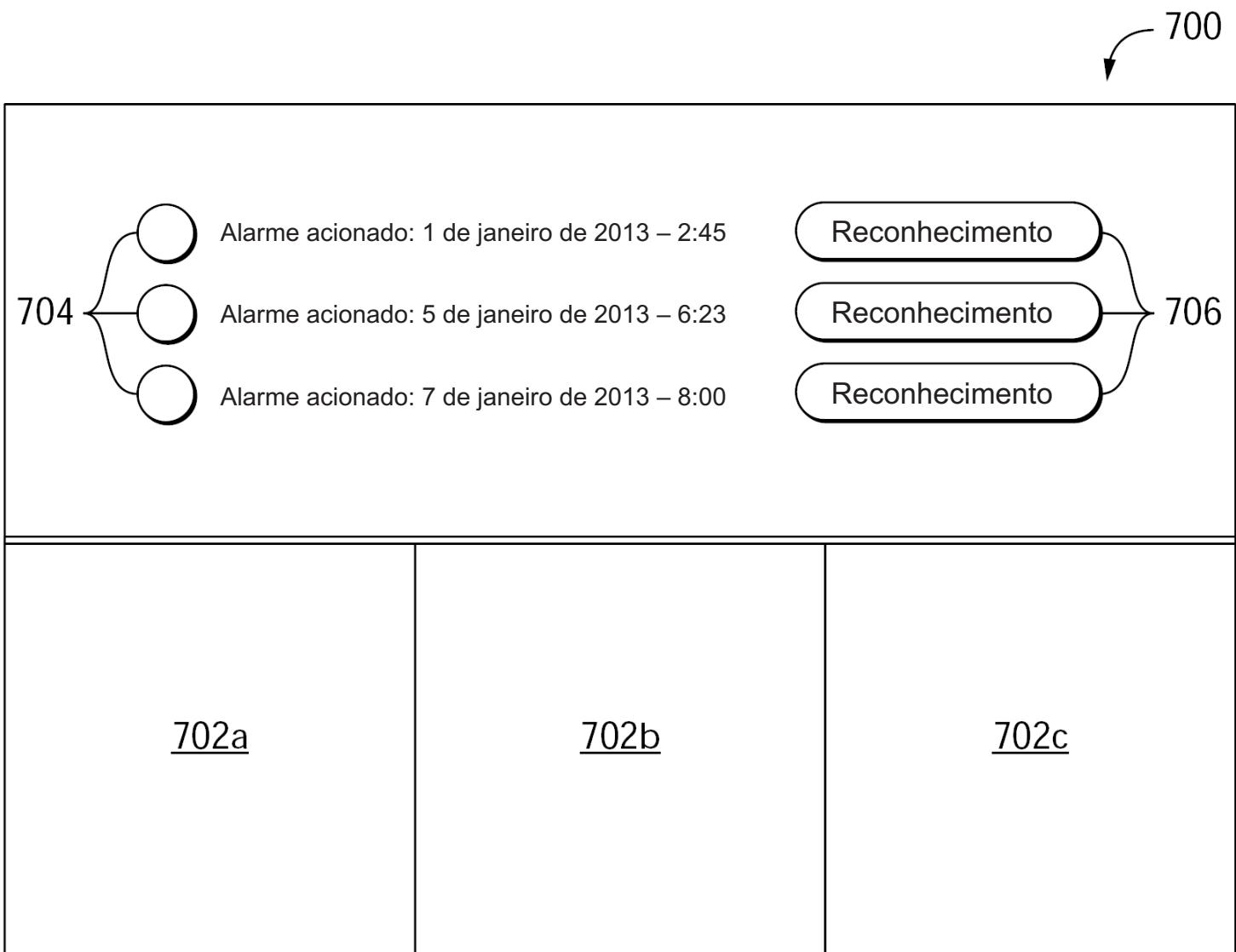


FIG. 5

**FIG. 6**

**FIG. 7**

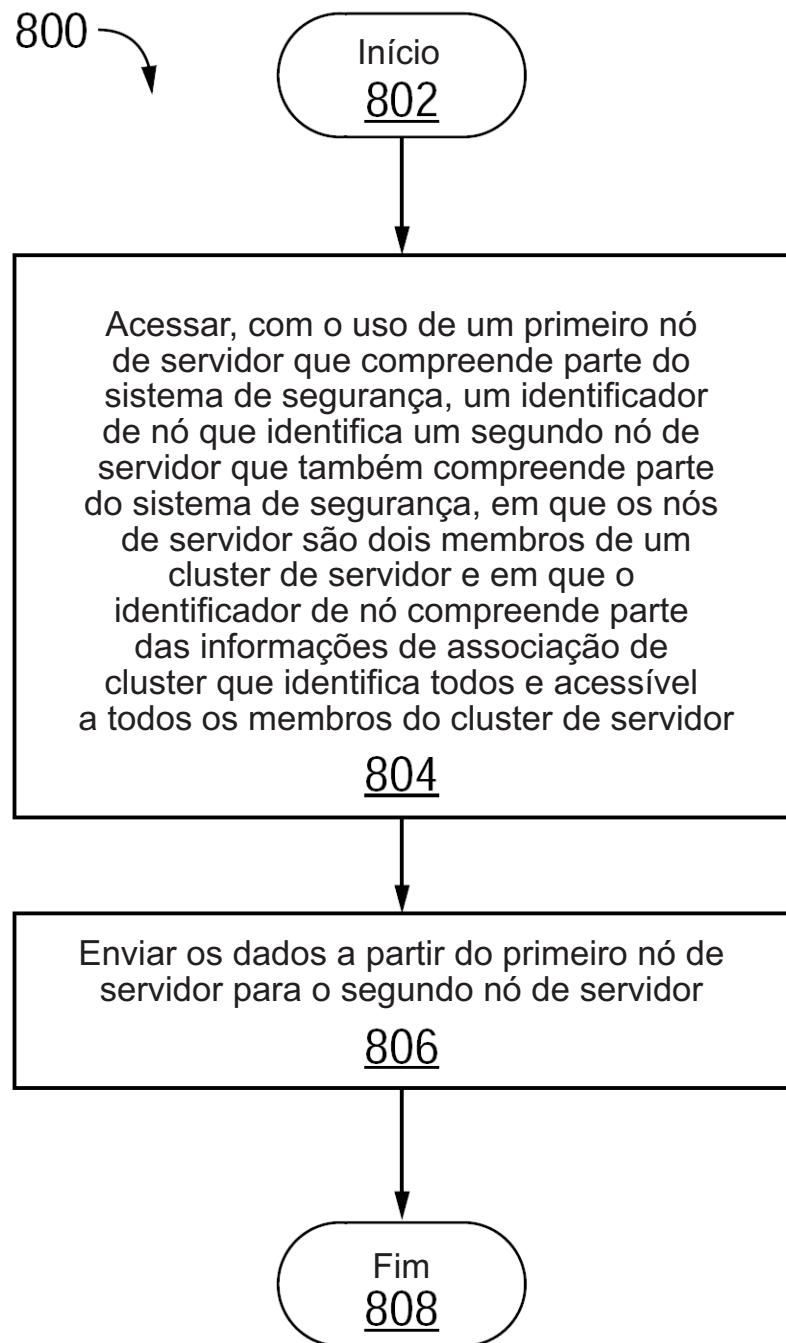
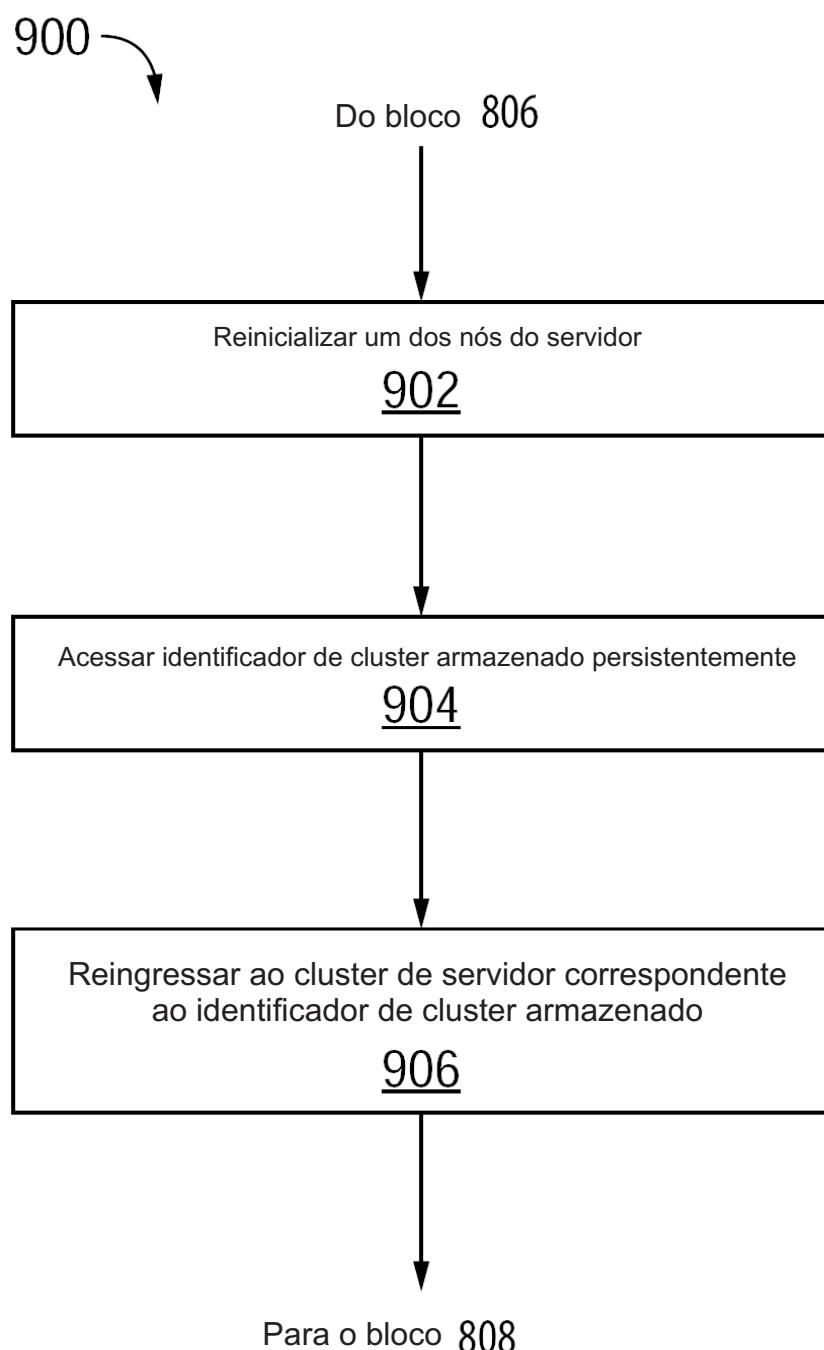
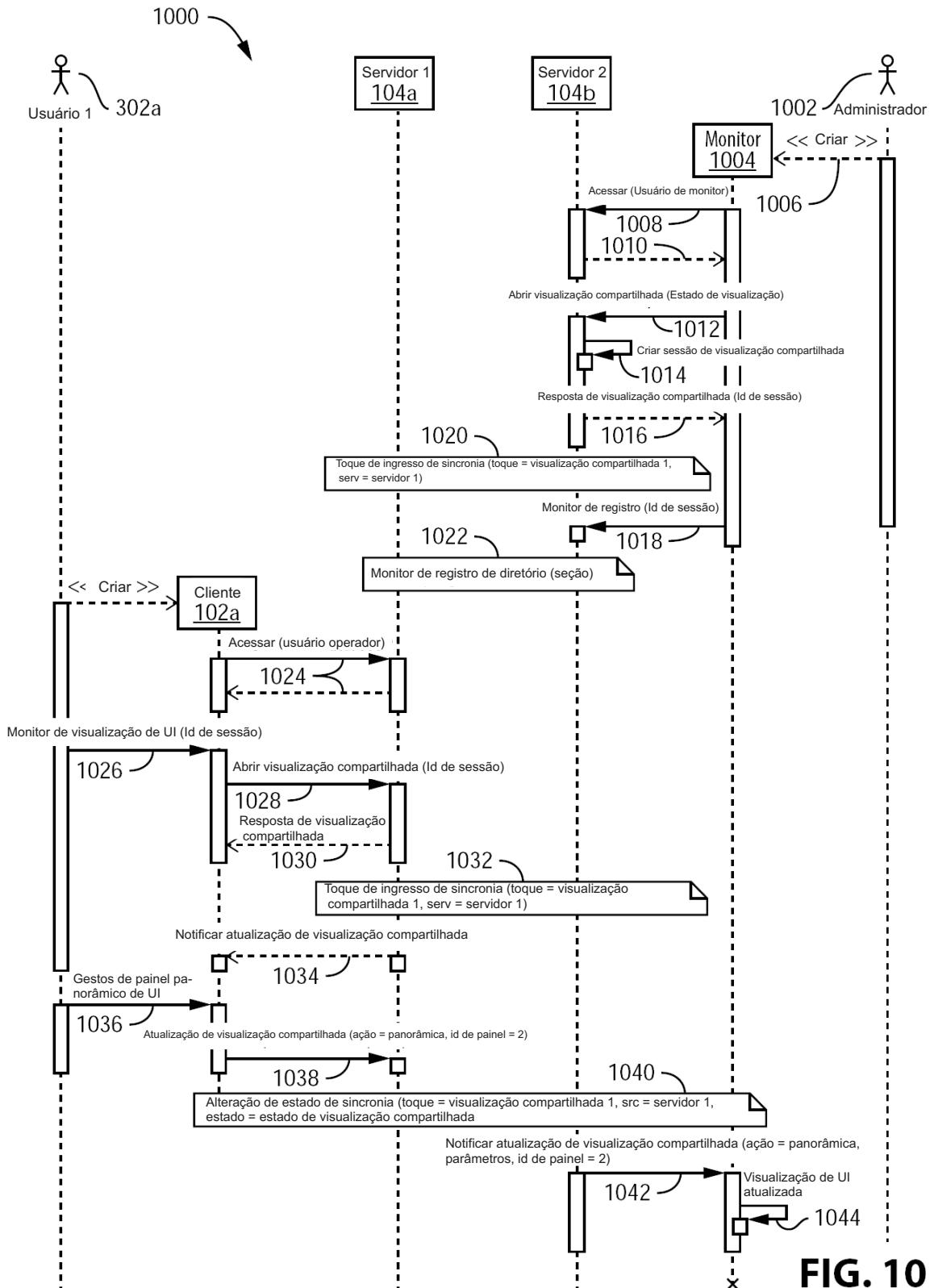
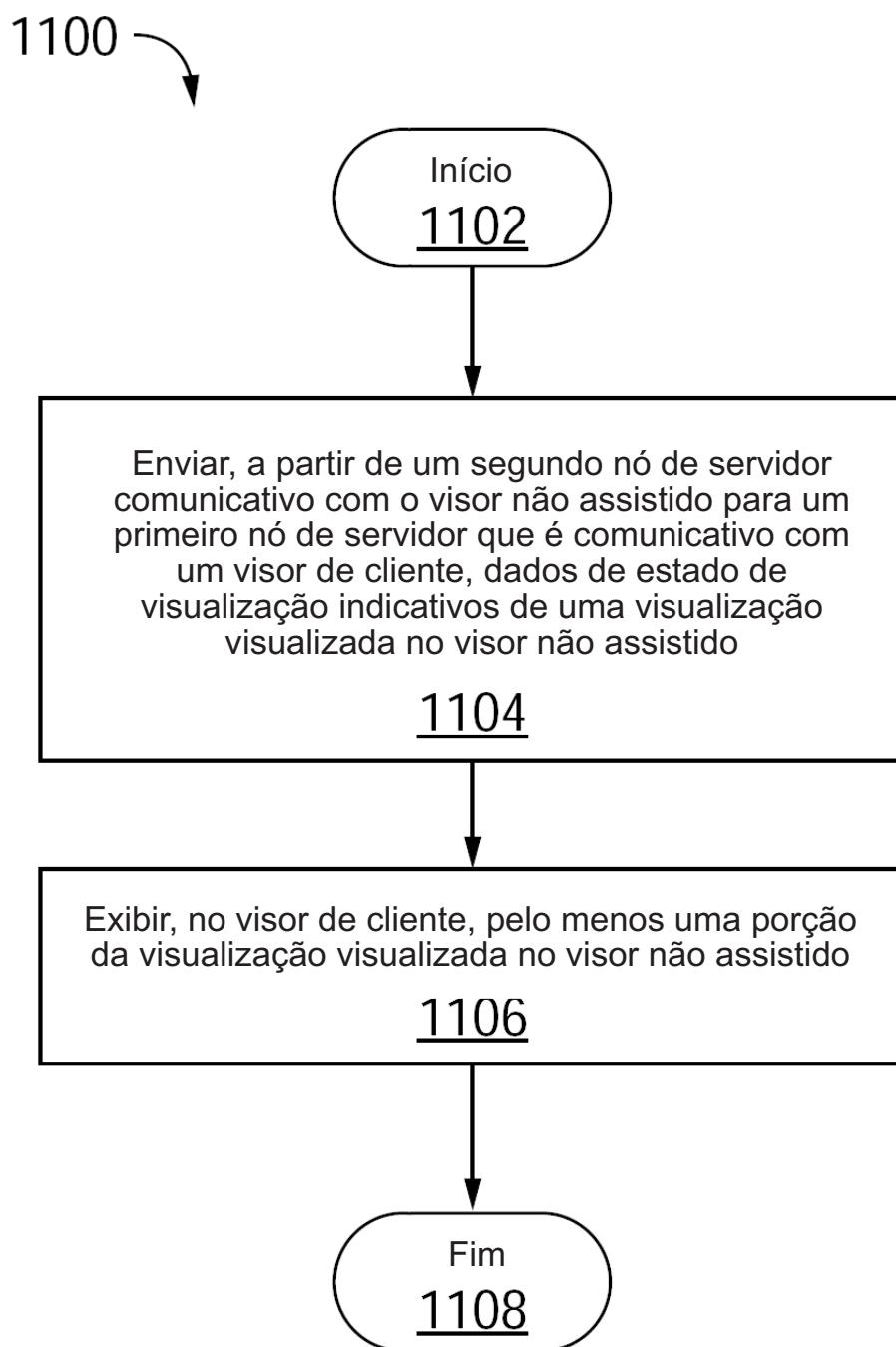


FIG. 8

**FIG. 9**

**FIG. 10**

**FIG. 11**

12/12

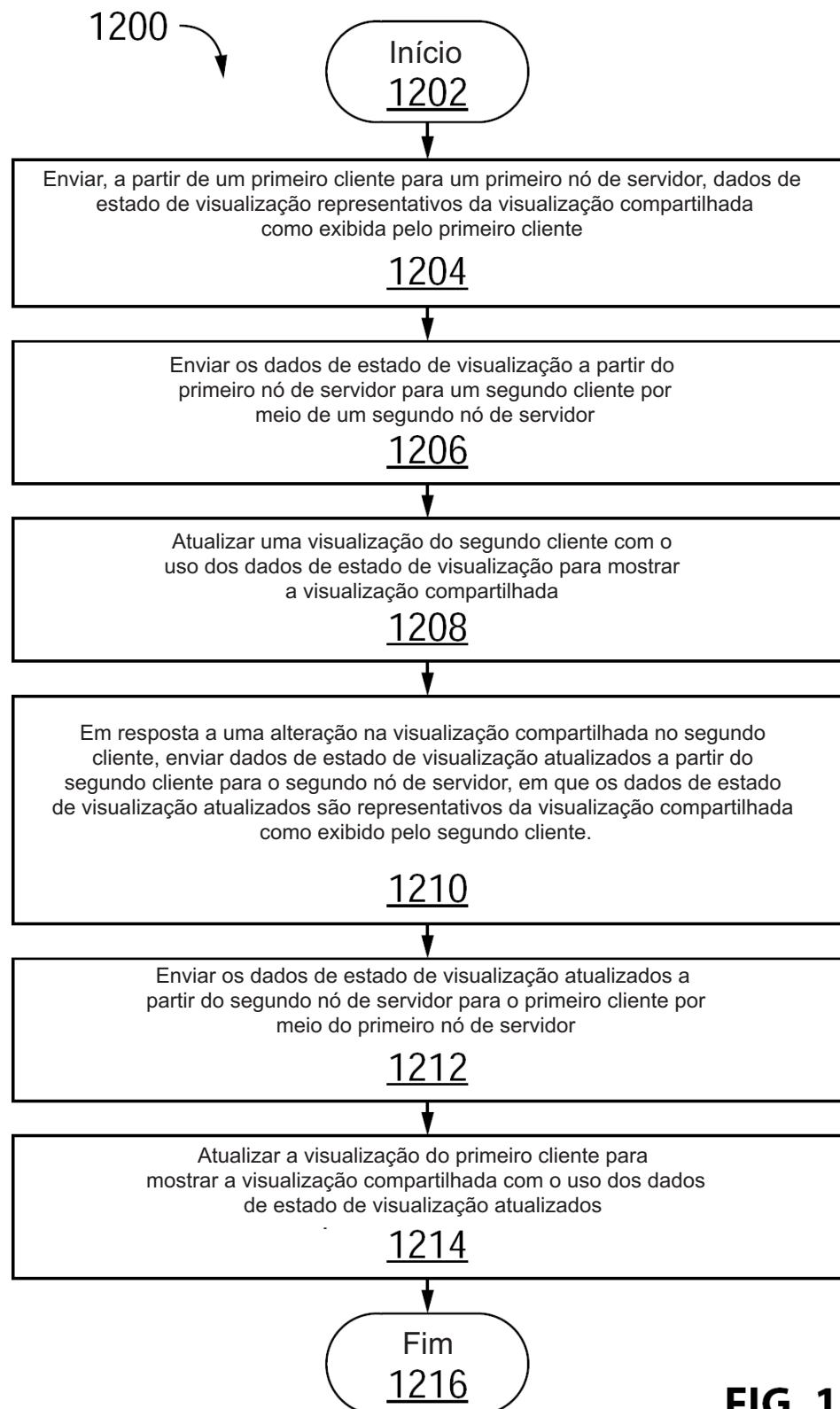


FIG. 12