



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

(11) PI 0820975-8 B1



(22) Data do Depósito: 12/12/2008

(45) Data de Concessão: 10/11/2020

(54) Título: MÉTODO EXECUTADO POR UM SERVIDOR, MEIO QUE PODE SER LIDO EM COMPUTADOR E DISPOSITIVO DE REDE PARA ESPECIFICAÇÃO, APLICAÇÃO E EXTENSÃO DE ASPECTOS RELACIONADOS A APLICATIVO REFERÊNCIA CRUZADA A PEDIDOS RELACIONADOS

(51) Int.Cl.: H04L 12/58; H04L 29/08; H04W 4/029; G06Q 10/10.

(52) CPC: H04L 51/38; H04L 67/24; H04L 67/26; H04L 67/18; H04L 51/04; (...).

(30) Prioridade Unionista: 14/12/2007 US 61/013,834; 29/05/2008 US 61/056,889.

(73) Titular(es): BLACKBERRY LIMITED.

(72) Inventor(es): BRIAN MCCOLGAN; GAELLE MARTIN-COCHER; MICHAEL SHENFIELD.

(86) Pedido PCT: PCT CA2008002160 de 12/12/2008

(87) Publicação PCT: WO 2009/076752 de 25/06/2009

(85) Data do Início da Fase Nacional: 14/06/2010

(57) Resumo: MÉTODO E SISTEMA PARA ESPECIFICAÇÃO, APLICAÇÃO E EXTENSÃO DE ASPECTOS RELACIONADOS A APLICATIVO ATRAVÉS DE POLÍTICAS, REGRAS E/OU GATILHOS. Um método e um sistema para a criação de aspectos a partir de um serviço ou aplicativo o método definido aspectos relacionados a serviço; a inserção ou a encapsulação de aspectos de serviço como aspectos denominados em uma camada de abstração denominados com uma lógica na camada de abstração, para suporte de pontos de função de aplicativo ou serviço.

**MÉTODO EXECUTADO POR UM SERVIDOR, MEIO QUE PODE SER LIDO EM
COMPUTADOR E DISPOSITIVO DE REDE PARA ESPECIFICAÇÃO,
APLICAÇÃO E EXTENSÃO DE ASPECTOS RELACIONADOS A APLICATIVO
REFERÊNCIA CRUZADA A PEDIDOS RELACIONADOS**

5 O presente pedido reivindica o benefício do Pedido de
Patente Provisória U.S. 61/013.834; depositado em 14 de
dezembro de 2007, e no Pedido de Patente Provisória U.S.
61/056.889, depositado em 29 de maio de 2008, cujas
exposições são incorporadas como referência aqui em suas
10 totalidades.

CAMPO DA EXPOSIÇÃO

A presente exposição se refere à ciência de contexto
de aplicativo e, em particular, à ciência de contexto de
aplicativo em uma rede móvel.

15 **ANTECEDENTES**

Os aplicativos possuem utilidades funcionais que têm
características importantes conhecidas como contexto. O
contexto é definido como “o conjunto de informação que
circunda e dá significado a alguma outra coisa”. Os
20 exemplos de contexto podem ser encontrados, por exemplo, em
aplicativos de presença, aplicativos de localização, dentre
outros.

Com respeito à informação de presença, os metadados de
presença provêm significado, e a informação de presença é
25 a base do contexto. O significado é aplicado a ou faz parte
de uma função em particular ou de um recurso em particular
de uma função em um aplicativo para o estabelecimento de um
conjunto apropriado de etapas de processamento.

Em um exemplo, um aplicativo de cliente de mensagem
30 instantânea (IM) operável em um dispositivo móvel de

primeiro usuário pode requerer uma funcionalidade para se estabelecer se outros indivíduos ou pares são alcançáveis para se permitir que o primeiro usuário inicie uma sessão de bate-papo de IM. Também é possível que em um cliente de IM as funcionalidades sejam requeridas para o estabelecimento de um ícone de status de usuário de par para a representação de um segundo usuário. No primeiro cenário, o contexto se refere a se o segundo usuário é alcançável para a iniciação de um bate-papo. No segundo cenário, o cliente de IM de primeiro usuário discrimina e deriva um ícone de status com base no status de segundo usuário e na disponibilidade para exibição do ícone de status correto, indícios ou avatar. No contexto do cliente de IM, a alcançabilidade como se refere a um recurso de ícone de status de par pode não ser relevante, ao passo que a alcançabilidade é útil para facilitar a função de bate-papo iniciada.

O dito acima demonstra, em um ambiente de presença, que o contexto tem um papel significativo em como uma informação de presença de indivíduo pode ser computada para a derivação de aspectos relacionados à presença, incluindo alcançabilidade, disponibilidade, dentre outros. Conforme será apreciado, o contexto também se aplica em outros cenários, além de presença.

Um serviço de presença captura uma informação de presença a partir de uma ou mais fontes de presença. Uma vez que estes dados sejam coletados, um serviço de presença compõe os metadados capturados e distribui um documento de metadados de presença brutos para os observadores autorizados. A plataforma de serviço de OMA-Presence é um

exemplo demonstrativo de um serviço de presença. O habilitador OMA-Presence destaca, em uma forma escrita muito detalhada, a semântica e a política relacionadas à publicação e ao consumo de uma informação de presença.

5 SUMÁRIO

A presente exposição provê um método de habilitação de um mecanismo ciente de contexto em um ambiente de execução, o método compreendendo: a definição de aspectos que pelo menos um aplicativo emprega, um aplicativo sendo uma
10 abstração de nível de aplicativo relevante para uma fonte ou um serviço; a associação de pelo menos uma das regras e gatilhos em relação a certos aspectos a partir da etapa de definição; e a associação de políticas com um contexto em um ponto em um ciclo de vida de aplicativo, as referidas
15 políticas sendo usadas como uma base para avaliação de regras no mecanismo ciente de contexto, onde os referidos aspectos, regras e políticas permitem que pelo menos um referido aplicativo abstraia uma funcionalidade para o mecanismo ciente de contexto.

20 A presente exposição ainda provê um mecanismo ciente de contexto que compreende: aspectos, os referidos aspectos sendo abstrações de nível de aplicativo relacionadas a um contexto; regras e gatilhos provendo uma sequência de etapas ou lógica empregada para a computação de aspectos; e
25 políticas associadas a um contexto em um ponto em um ciclo de vida de aplicativo, as referidas políticas sendo usadas como uma base para avaliação de regras no mecanismo ciente de contexto, onde os referidos aspectos, regras e políticas permitem que pelo menos um cliente abstraia uma
30 funcionalidade para o mecanismo ciente de contexto.

BREVE DESCRIÇÃO DOS DESENHOS

A presente exposição será mais bem entendida com referência aos desenhos, nos quais:

5 a Figura 1 é um diagrama de blocos que mostra uma plataforma de presença de exemplo com um cliente e um servidor de *push-to-talk* por celular;

a Figura 2 é um fluxograma que ilustra um método de processamento de exemplo em um dispositivo de cliente para a derivação de aspectos de alcançabilidade;

10 a Figura 3 é um diagrama de blocos que mostra um sistema de presença de exemplo no qual um mecanismo ciente de contexto de presença foi adicionado;

a Figura 4 é um diagrama de blocos que mostra um sistema de presença de exemplo no qual um mecanismo ciente de contexto de presença foi adicionado e distribuído entre
15 um servidor e agentes;

a Figura 5 é um diagrama de blocos que mostra um sistema de presença de exemplo no qual um mecanismo ciente de contexto de presença foi adicionado a um servidor de
20 PoC;

a Figura 6 é um diagrama de blocos que mostra um sistema de presença de exemplo no qual um mecanismo ciente de contexto de presença foi adicionado a uma Plataforma Presence;

25 a Figura 7 é um diagrama de blocos que mostra um sistema de localização de exemplo no qual um mecanismo ciente de contexto de presença foi adicionado;

a Figura 8 é um diagrama de blocos que mostra um sistema genérico de exemplo no qual um mecanismo ciente de
30 contexto de presença foi adicionado;

a Figura 9 é um fluxograma que mostra um método de exemplo para a determinação de alcançabilidade, quando se utiliza um P/CAM;

5 a Figura 10 é um fluxograma que mostra um método de exemplo para se determinar se um prospecto é elegível para receber anúncios utilizando um P/CAM;

a Figura 11 é um fluxograma que mostra um método de exemplo para se determinar se um cliente de push pode ter um conteúdo pressionado para ele utilizando-se um P/CAM;

10 a Figura 12 é um fluxograma de chamada de exemplo que mostra um fluxo de chamada para estabelecimento de política;

a Figura 13 é um fluxograma de chamada de exemplo que mostra o fluxo de chamada para aspectos em um ambiente de
15 OMA/PRS; e

a Figura 14 é um fluxograma de chamada de exemplo que mostra o fluxo de chamada para gatilhos de aspecto.

DESCRIÇÃO DETALHADA

Termos:

20 Na presente descrição, os termos a seguir são usados e definidos conforme se segue:

Contexto: Aquilo que circunda e dá significado a alguma outra coisa.

OMA: Aliança Móvel Aberta (Open Mobile Alliance).

25 **PEEM:** Habilitador de avaliação, cumprimento e gerenciamento de política.

Informação de Presença (Inf. de Presença): Uma unidade básica de presença (por exemplo, atividade ou humor é uma informação de presença).

30 **Serviço de Presença:** Entidade ou plataforma que recebe uma

informação de presença de fontes de presença.

Fonte de Presença: Entidade que se relaciona a uma informação de presença em nome de uma ou mais presentidades.

- 5 **Presentidade:** Entidade que tem uma informação de presença relacionada a ela.

Observador (Watcher): Entidade que deseja consumir uma informação de presença

- 10 **Camada Ciente de Contexto:** Uma camada que pode ser uma camada de acesso, de aplicativo, abstração ou proxy. Esta camada pode fazer uso de aspectos. Esta camada pode ser empregada por uma rede e pode ser adaptada para lidar com requisições de uma pluralidade de clientes de vários tipos. Esta camada pode incluir mecanismos cientes de contexto,
- 15 tais como, por exemplo, x/CAM, o que é um mecanismo ciente de contexto não específico (genérico), ou mecanismos específicos, tais como presença (p/CAM) e localização (L/CAM).

Descrição:

- 20 A Figura 1 ilustra um diagrama de blocos de uma plataforma de presença de exemplo que é empregada no contexto de um sistema de *push-to-talk* por celular (PoC). O uso de uma plataforma de presença é meramente um exemplo, e outras plataformas, tal como uma plataforma de localização
- 25 ou genérica são possíveis. Mais ainda, a plataforma de presença (ou outra plataforma de localização ou genérica) pode ser empregada em outros contextos, tal como, por exemplo, IM. Especificamente, na Figura 1, os dispositivos de usuário 110 se comunicam por um sistema de comunicação
- 30 sem fio (por exemplo, celular) com uma estação base 112, a

qual, então, se comunica com uma rede de protocolo de internet 120 ou outra rede, conforme conhecido por aqueles versados na técnica. Conforme será apreciado, a estação base 112 poderia ser uma estação base para qualquer serviço de comunicação sem fio conhecido (por exemplo, celular, PCS, iDEN, etc.). Ainda, os dispositivos 110 poderiam se comunicar com uma rede 120 por outros meios, tal como uma comunicação sem fio de alcance curto como o Bluetooth™, por WiFi ou WLAN, através de uma conexão com fio, tal como através de uma porta USB ou serial ou através de um modem de computador. De fato, outros meios de conexão com a rede 120 seriam conhecidos por aqueles versados na técnica.

No sistema da Figura 1, um computador de mesa 114 (por exemplo, um dispositivo de computação, que é similar a ou diferente de dispositivos de usuário 110) com um cliente de PoC pode se comunicar com um ou mais dos dispositivos de usuário 110 através de uma rede de área ampla 118 e uma rede 120.

Uma plataforma de presença 130 recebe e envia um fluxo de informação de presença a partir da rede 120 para os dispositivos de usuário 110 ou o computador de mesa 114. A plataforma de presença 130 é adaptada para armazenar dados brutos referentes a estados de clientes e para atualizar registros de cliente, quando novos dados de estado forem recebidos. A plataforma de presença 130 é adicionalmente adaptada para prover uma informação de presença para um observador. Assim, a informação de presença flui para a e a partir da plataforma de presença 130.

Um servidor de *push-to-talk* por celular (PoC) 140 existe no sistema da Figura 1 e, em uma modalidade, poderia

publicar uma informação de estado em nome de uma presentidade ou um observador. Conforme será apreciado por aqueles versados na técnica com referência à Figura 1, o consumo e a interpretação de metadados de presença para a obtenção de funções ou de recursos no contexto de um aplicativo relativo a um assunto de interesse podem ser realizados pelo aplicativo. Um aplicativo neste caso poderia ser o servidor de PoC, um cliente de PoC ou um cliente de IM, dentre outros.

Os dispositivos de usuário 110, o computador de mesa 114 e o servidor de PoC 140 poderiam atuar como observadores e fontes de presença no exemplo da Figura 1.

Conforme será apreciado por aqueles versados na técnica, a exigência para o consumo e a interpretação de metadados de presença para a obtenção de funções ou recursos no contexto de um aplicativo aumenta a complexidade de um aplicativo de cliente. Independentemente, esta complexidade tem o efeito líquido de aumentar a área ocupada de memória, bem como as exigências de processamento geral, consumo de potência e largura de banda de rede para o aplicativo. Além disso, um aplicativo relacionado à presença ainda se torna susceptível a mudanças ou adições aos metadados subjacentes ou mudanças em semântica de plataforma de presença ou política. Por exemplo, uma correção de bug ou uma mudança nas normas da OMA poderiam requerer que um aplicativo de cliente fosse atualizado ou mudado, de modo a se interpretarem corretamente os metadados no futuro. Também, os metadados poderiam ser adicionados ou mudados na semântica de presença.

O dito acima tem o efeito líquido de mudanças frequentes no aplicativo empregado em um ambiente de execução de usuário, de modo a se manter apropriadamente uma vista apropriada de observador ou presentidade. Também,
5 ainda há um custo de tempo e um custo relacionado ao emprego de um dado aplicativo.

Isto é adicionalmente ilustrado com referência ao exemplo da Figura 2. Uma referência é feita, agora, à Figura 2, a qual mostra um fluxograma de uma transação de
10 exemplo em que um aplicativo de cliente de PoC está para iniciar um alerta de PoC para um assunto de interesse. Neste caso, um primeiro usuário, Alice, deseja enviar um alerta de PoC para uma presentidade, Bob, usando seu cliente de PoC (um observador).

15 O processo começa no bloco 210 e prossegue para o bloco 212, no qual o cliente de PoC busca ou é notificado do documento de presença de Bob por um servidor de presença. Conforme será apreciado por aqueles versados na técnica, quando um serviço é implementado para Bob e Alice
20 serem capazes de usar o *push-to-talk* com cada outro, uma assinatura é feita junto ao servidor de presença para a provisão de um documento de presença relacionado a Bob, ou quando o PoC deseja se comunicar com Bob, então, uma busca é feita a partir do servidor de presença e esta informação
25 é recebida como um documento de presença no bloco 212.

O processo então prossegue para o bloco 214, no qual uma checagem é feita para se ver se o documento de presença contém quaisquer tuplas de serviço de alerta de PoC. Conforme será apreciado, isto é uma checagem para ver se
30 qualquer coisa no documento de presença está ou não

relacionada ao identificador de serviço para este serviço (neste caso, o serviço de alerta de PoC).

Se, no bloco 214, o documento de presença contiver realmente tuplas de serviço de alerta de PoC, o processo
5 prosseguirá para o bloco 216, no qual o cliente de PoC busca de forma aprofundada no documento de presença para encontrar tuplas de serviço de alerta de PoC de acordo com a semântica de presença da OMA. Conforme será apreciado, isto provê uma forma de destilar uma informação relevante
10 do serviço sendo requisitado. O cliente neste caso emprega um conhecimento embutido da semântica de presença de OMA, de modo a fazer isto.

O processo então prossegue para o bloco 218, no qual o cliente de PoC encontra o elemento de pessoa mais relevante
15 no documento de presença de acordo com a semântica de presença da OMA. Conforme será apreciado, o documento de presença poderia incluir múltiplos elementos de pessoa. A OMA / Presence define a semântica para a determinação da pessoa mais relevante.

20 O processo no bloco 220 em seguida checa para ver se Bob está desejando ser contatado pelo alerta de PoC, e se ele está disponível para a tupla de serviço resolvida. Conforme será apreciado, os termos "desejando" e "disponível" são específicos para a presença e têm
25 critérios definidos para se resolver se alguém está "desejando" e/ou "disponível".

Se Bob estiver "desejando" e "disponível", o processo prosseguirá para o bloco 222 no qual o cliente de PoC estabelece um meio de contato que inclui o dispositivo para
30 o serviço de alerta de PoC para Bob. Conforme será

apreciado, múltiplos endereços poderiam ser providos, e uma prioridade para aqueles endereços também poderia ser provida.

O processo então prossegue para o bloco 224, no qual uma checagem é feita para se ver se Bob é "contatável. De novo, isto tem um significado específico na semântica de presença e indica que se Bob estiver "desejando" e "disponível", então, um meio de contato poderá ser estabelecido.

O processo então prossegue para o bloco 226, se Bob for "contatável. No bloco 226, uma checagem é feita para se ver se o meio de contato é válido. O meio de contato pode ser inválido se tiver expirado, ou se for antigo demais e um limite de tempo quanto à validade do meio de contexto tiver sido posto, dentre outras coisas.

A partir dos blocos 214, 440, 224 ou 226, se uma conclusão negativa for obtida, o processo prosseguirá para o bloco 230, o qual indica que Bob é inalcançável, e o processo termina no bloco 232.

A partir do bloco 226, se o meio de contato for válido, o processo prosseguirá para o bloco 240, no qual cada elemento de dispositivo no documento de presença é identificado. Para cada documento de presença o processo prossegue para o bloco 242, no qual o identificador de dispositivo é combinado com o meio de contato. Se uma combinação for feita, o processo prosseguirá para o bloco 250. Caso contrário, o processo prossegue para o bloco 244, no qual uma checagem é feita para se ver se há mais elementos de dispositivo disponíveis. Caso sim, o processo prossegue de volta através dos blocos 240 e 242. Caso

contrário, o processo prossegue para o bloco 230, no qual Bob é julgado inalcançável, e o processo termina no bloco 232.

No bloco 250, o processo isola cada subelemento da rede na disponibilidade da rede dentro do dispositivo e uma verificação é feita no bloco 252 para ver se a rede é equivalente ao tipo de rede aplicável ou requerida para o serviço de alerta de PoC, e se a rede está disponível. Esta é uma decisão que o aplicativo de cliente faz com base na política ou ela pode estar embebida no cliente (ou servidor) que se comunica com a camada P/CAM. Se o processo do bloco 252 recebe um resultado positivo, o processo prossegue para o bloco 260, no qual Bob é julgado alcançável e o processo termina então no bloco 262.

Caso contrário, o processo prossegue para o bloco 254, no qual uma checagem é feita para se ver se há outros subelementos de rede que possam ser utilizados e, caso sim, o processo prossegue através dos blocos 250 e 252 de novo, para se fazer uma checagem para ver se a rede é ou não equivalente ao tipo de rede requerido ou aplicável e está disponível. A partir do bloco 254, se nenhum outro subtipo de rede estiver disponível, o processo prossegue para o bloco 230 no qual Bob é julgado inalcançável e o processo termina em 232.

Com respeito às Figuras 1 e 2 acima, a interpretação contextual de uma informação de presença pode ser embutida em cada aplicativo de cliente. Cada aplicativo de cliente pode receber um conjunto diferente ou o mesmo de metadados de presença e, em situações em que múltiplos requerentes compartilham os mesmos metadados de presença, o fato de que

a interpretação contextual é individualmente ligada a cada um deles aumenta a possibilidade de dois aplicativos de cliente diferentes chegarem a conclusões diferentes sobre um aspecto de presença específico. Isto pode não prover o
5 resultado desejado e pode levar a questões de interoperabilidade, particularmente, entre aplicativos de cliente que compartilhem ou tratem de aspectos de presença específicos de uma maneira ortogonal e consistente.

Por exemplo, um e-mail e um cliente de IM que derivem
10 a alcançabilidade de uma pessoa a partir do mesmo documento de presença bruto podem chegar a conclusões diferentes quanto a se alguém é alcançável com base em variações sutis em cada uma das etapas de processamento de presença de aplicativo de cliente. Isto pode resultar no cliente de e-mail concluir que o indivíduo é inalcançável. Além de uma
15 má qualidade de serviço, isto poderia resultar em questões com interoperabilidade, tal como não ser capaz de gerar uma sessão de bate-papo de IM a partir de um cliente de e-mail quando revendo o e-mail de um indivíduo, devido a um erro
20 de não combinação de estado.

A abstração da informação de presença bruta em uma camada ciente de contexto dedicada a qual suporta "aspectos de presença" com base em regras contextuais e políticas permite a possibilidade de aplicativos trabalharem de forma
25 colaborativa para a obtenção de uma funcionalidade derivada e para a realização de fluxos de trabalho inteligentes, como resultado de uma presença de contexto composta. Por exemplo, um gerenciador de projeto deseja hospedar uma reunião sobre status de projeto. O gerenciador de projeto
30 estabelece um convite para a reunião (por exemplo, a partir

de um aplicativo de e-mail / calendário de empresa) em seu ambiente de execução de mesa para os participantes da reunião. Uma plataforma de presença - contexto trabalhando em nome do aplicativo de correio / calendário pode ser capaz de suportar os tipos a seguir de funções, como resultado de o usuário iniciar o convite:

- Determinar um horário apropriado com base na disponibilidade de participante;
- Com base na política de contexto, reservar uma sala de reuniões apropriadas para a reunião;
- Determinar com base na localização de participante (e na política da empresa) se equipamentos de som e áudio de conferencia devem ser reservados (e refletir isto para os indivíduos apropriados na requisição da reunião);
- Com base em sugestões ou na política dada pelo moderador da reunião através do aplicativo, convidar os participantes relevantes que atendam a um dado critério (por exemplo, um membro da equipe de marketing, um membro da equipe de desenvolvimento, um membro da equipe de garantia de qualidade (QA), um indivíduo com uma habilidade ou um conhecimento específico, etc.).

Ainda, vários servidores de aplicativo podem integrar o mecanismo ciente de contexto de presença (P/CAM) para se ganhar eficiência pela redução do número de etapas de comunicação e de processamento. Por exemplo, um servidor de propaganda móvel poderia se integrar com o P/CAM para simplificar e otimizar seus aspectos de presença para focalizar em uma funcionalidade de núcleo, tal como na

entrega de propagandas móveis relevantes de forma contextual.

A presente exposição provê um método e um sistema para o estabelecimento de um contexto, onde um mecanismo é conectado a uma plataforma de servidor para suportar um dado aplicativo. A ciência de contexto reside no todo ou em parte na rede, e provê uma visão compósita de presença / localização ou outros aspectos relacionados a um aplicativo ou a múltiplos aplicativos em nome de várias entidades, tais como uma dada presentidade e/ou um observador no caso de presença. Para cada caso, isto é obtido pela associação de regras, gatilhos e políticas com relação a aspectos relacionados à presença, tais como disponibilidade, contatibilidade, alcançabilidade, estado, dentre outros, em uma camada ciente de contexto. As regras ou os gatilhos podem ser estendidos ou suprimidos para a provisão de um comportamento adicional ou específico de aplicativo para diferentes classes de aplicativos ou habilitadores.

A ciência de contexto pode ser replicada para um mecanismo ciente de contexto de presença ou localização conectado a uma plataforma de serviço de presença ou localização para a provisão a um aplicativo de cliente ou serviço de aspectos relacionados à localização. Um mecanismo ciente de contexto de localização (L/CAM) faz uso de uma informação de localização provida por um habilitador de localização, uma informação de localização armazenada em um serviço de presença ou outro armazenamento de informação de localização. Por exemplo, a localização poderia ser derivada usando-se GPS, estação base ou uma informação de torre de celular estendida.

As regras e políticas específicas de localização estão associadas a aspectos relacionados à localização, tal como em uma área geográfica, quem está próximo, já estou lá, dentre outros, em uma camada ciente de contexto de localização. Como com um P/CAM, regras ou gatilhos podem ser estendidos ou suprimidos para a provisão de um comportamento adicional / específico de aplicativo para diferentes classes de aplicativos ou habilitadores de serviço.

De modo similar, uma camada ciente de contexto "genérica" (mecanismo ciente de contexto) poderia conter uma combinação de um P/CAM, um L/CAM e um mecanismo ciente de contexto de aplicativo específico. Um exemplo poderia ser uma plataforma de propaganda móvel, onde uma presença, localização e uma informação relacionada à campanha são usadas em combinação com propagandas alvos de interesse para um usuário. Outras plataformas genéricas poderiam incluir um serviço de catálogo de endereços, um serviço comunitário de rede, dentre outros.

Conforme será apreciado por aqueles versados na técnica, um mecanismo ciente de contexto é aplicável a um ambiente de execução com fio e sem fio e a um domínio de computação. Esta abordagem tem vários benefícios incluindo uma redução dramática na complexidade de um aplicativo associado rodando em um ambiente de execução de usuário. Uma plataforma ciente de forma contextual localizada na rede permite que um dado aplicativo de cliente ou habilitador se concentre em sua competência de lenço úmido, tal como um bate-papo em um cliente de IM, visualizando a localização de uma pessoa em um cliente de localização,

dentre outros. Uma funcionalidade é obtida pela injeção (por exemplo, em um momento de execução) das políticas aplicáveis e pela invocação de regras especiais e/ou gatilhos relevantes para o contexto do aplicativo de cliente ou o habilitador, para a provisão de um utilitário em nome do usuário.

Em uma modalidade adicional, uma plataforma ciente de contexto ou uma camada ciente de contexto inclui um servidor de x/CAM e um cliente ou agente de x/CAM que trabalham em consonância. Ainda, em algumas modalidades do x/CAM, os mesmos aspectos distribuídos ou não distribuídos como o P/CAM e L/CAM mencionados acima são possíveis. Por exemplo, a camada ciente de contexto pode existir apenas no lado de servidor em algumas modalidades. O cliente ou agente de camada ciente de contexto é embutido em um ambiente de execução. A interface para uma plataforma ciente de contexto pode ser centrada na web. Os exemplos incluem serviços da web de linguagem de marcação extensível (XML), tal como um protocolo de acesso de objeto simples (SOAP), uma transferência de estado de representação (REST) ou XML por um protocolo de transferência de hipertexto (HTTP). O dito acima suporta um cenário de emprego de camada ciente de contexto, por meio do qual um aplicativo ou habilitador poderia interagir diretamente ou manipular o mecanismo ciente de contexto para modelar de forma mais próxima o comportamento apropriado. Por exemplo, um servidor de propaganda móvel co-localizado com um agente de P/CAM poderia ser usado para a supressão de políticas de presença para mais bem alinhar a presença com a funcionalidade subjacente da plataforma. Por exemplo, um

servidor de propaganda móvel pode integrar ou fazer uso de uma 'camada' de x/CAM. Esse x/CAM poderia ser um superconjunto de um P/CAM, um L/CAM e um CAM específico de propaganda.

5 Uma referência é feita, agora, à Figura 3. A Figura 3 ilustra um diagrama de sistema para uma plataforma de presença com um aplicativo de cliente de PoC utilizando um P/CAM como uma camada ciente de contexto. Conforme será apreciado, a Figura 3 utiliza aspectos de rede similares
10 àqueles da Figura 1 com a adição do P/CAM.

Na Figura 3, os dispositivos de usuário 310 se comunicam através de uma estação base 312 com uma rede 320. Ainda, um computador de mesa 314 (por exemplo, um dispositivo de computação, que é similar a ou diferente de
15 dispositivos de usuário 310) se comunica através de uma rede de área ampla 316 com a rede 320.

Uma plataforma de presença 330 é adaptada para o armazenamento de dados brutos e atualizações de estado que foram recebidas a partir de clientes.

20 Ainda, um servidor de PoC 340 existe e é adaptado para a publicação ou o consumo de uma informação de estado em nome de usuários.

Um servidor de mecanismo ciente de contexto de presença 350 provê a camada ciente de contexto e se
25 comunica com a rede 320 e recebe políticas, regras dinâmicas e/ou gatilhos de clientes pela rede 320 e, ainda, publica e recebe aspectos de presença através da rede 320.

Um servidor de mecanismo ciente de contexto de presença 350 ainda se comunica com a plataforma de presença
30 330 para a provisão e o recebimento de um fluxo de

informação de presença.

A Figura 3 ainda ilustra um enlace 332 entre a rede 320 e a plataforma de presença 330. Conforme será apreciado, este enlace 332 pode não ser omitido, apesar do
5 enlace de comunicação entre a plataforma de presença 330 e o servidor de P/CAM 350, de modo a se permitir a clientes que queiram se comunicar diretamente com a plataforma de presença a capacidade de fazê-lo, e para a provisão de comunicações com a plataforma para uma nova informação ou
10 uma informação avançada de que o servidor de P/CAM 350 ainda pode não estar ciente.

Com base no dito acima, o servidor de P/CAM 350 recebe políticas, regras e gatilhos e é adaptado para prover e receber aspectos de presença com base nas regras e na
15 lógica para clientes, tais como os dispositivos 310 ou o computador de mesa 314, ou o servidor de PoC 340.

Conforme será apreciado, em outras modalidades, vários aspectos ou uma funcionalidade do P/CAM podem ser distribuídos por toda a rede e, em alguns casos, o P/CAM
20 inteiro pode ser posto em outros dispositivos ou clientes na rede.

Uma referência é feita, agora, à Figura 4. A Figura 4 mostra um sistema similar àquele das Figuras 1 e 3, no qual a funcionalidade de P/CAM foi distribuída através de
25 agentes de P/CAM em vários dispositivos.

Especificamente, os dispositivos de usuário 410 se comunicam através de uma estação base 412 com a rede 420. Ainda, um computador de mesa 414 (por exemplo, um dispositivo de computação que é similar ou diferente dos
30 dispositivos de usuário 410) se comunica por uma rede de

área ampla 416 com a rede 420.

Uma plataforma de presença 430 é adaptada para o armazenamento de dados brutos e atualizações de estado que são recebidas a partir de clientes.

5 Ainda, um servidor de PoC 440 é adaptado para comunicação com a rede 420 e publicação ou consumo de estado em nome de aplicativos de cliente.

10 A camada ciente de contexto concretizada como um servidor de P/CAM 450 é adaptada para comunicação com a rede 420 e para recebimento de política, regras e limites e prover e receber aspectos de presença para e a partir de clientes, tais como os dispositivos de usuário 410 e o computador de mesa 414 através do agente de P/CAM 460 ou do servidor de PoC 440 através do agente de P/CAM 462.

15 O P/CAM 450 ainda é adaptado para comunicação com a plataforma de presença 430 para o recebimento e o envio de um fluxo de informação de presença.

20 Na modalidade da Figura 4, parte da funcionalidade do servidor de P/CAM 450 pode ser distribuída de modo a se permitir que a funcionalidade plena do P/CAM, ou parte dela seja realizada no dispositivo 410, no computador de mesa 414 ou no servidor de PoC 440, por exemplo. Isto é ilustrado pelo agente de P/CAM 460 nos dispositivos de usuário 410 ou no computador de mesa 414 e pelo agente de
25 P/CAM 462 no servidor de PoC 440. Neste caso, a camada ciente de contexto compreende o servidor de P/CAM 450 e o agente de P/CAM 460 e/ou 462.

30 O agente de P/CAM 460 ou 462 poderia conter regras e/ou políticas que são pré-definidas. Ainda, o agente de P/CAM 460 ou 462 pode ser usado para a manipulação da

informação de presença ou para interoperar com metadados ou clientes no ambiente de execução principal em algumas modalidades.

Conforme será apreciado, em algumas modalidades, o
5 P/CAM inteiro pode estar localizado em um cliente ou outro servidor.

Uma referência é feita, agora, à Figura 5. A Figura 5 ilustra um diagrama de sistema no qual o servidor de P/CAM (camada ciente de contexto) é embutido no servidor de PoC.

10 Especificamente, na Figura 5, os dispositivos de usuário 510 se comunicam através da estação base 512 com uma rede 520. Ainda, o computador de mesa 514 (por exemplo, um dispositivo de computação que é similar aos ou diferente dos dispositivos de usuário 510) se comunica por uma rede
15 de área ampla 516 e uma rede 520.

Uma plataforma de presença 530 é adaptada para o armazenamento de dados brutos e atualizações recebidas de clientes com referência à presença.

Um servidor de PoC 540 é adaptado para comunicação com
20 a rede 520 e para publicação e consumo de estado em nome de clientes.

O servidor de PoC 540 ainda inclui o P/CAM 550 embutido ali. O P/CAM 550 se comunica com a plataforma de presença 530 para a troca de um fluxo de informação de
25 presença e ainda se comunica pela rede 520 para receber uma informação de política, regras e limites e para ainda receber e publicar aspectos de presença. Especificamente, as comunicações 552 provêm ao P/CAM 550 regras de política e sobrecarregadas dinâmicas, ao passo que as comunicações
30 554 provêm à rede 520 os aspectos de presença.

Ainda, uma implementação poderia ser definida como uma camada de P/CAM integrada em um habilitador, por exemplo, como parte da plataforma de presença em si. A última implementação, conforme ilustrado na Figura 6, também
5 poderia suportar uma variação por meio da qual uma camada ciente de contexto concretizada como um cliente / agente de P/CAM reside no dispositivo móvel, ou como parte de um habilitador associado (por exemplo, um servidor MobAd).

Uma referência é feita, agora, à Figura 6. A Figura 6
10 ilustra um diagrama de sistema no qual o servidor de P/CAM é embutido na plataforma de presença 630.

Especificamente, na Figura 6, os dispositivos de usuário 610 se comunicam através da estação base 612 com uma rede 620. Ainda, o computador de mesa 614 (por exemplo,
15 um dispositivo de computação que é similar aos ou diferente dos dispositivos de usuário 610) se comunica por uma rede de área ampla 616 com a rede 620.

Uma plataforma de presença 630 é adaptada para o armazenamento de dados brutos e atualizações recebidas de
20 clientes com referência à presença.

Um servidor de PoC 640 é adaptado para comunicação com a rede 620 e para publicação e consumo de estado em nome de clientes.

A plataforma de presença 630 ainda inclui o P/CAM 650
25 embutido ali. O P/CAM 650 se comunica com a plataforma de presença 630 para a troca de um fluxo de informação de presença e ainda se comunica pela rede 620 para receber uma informação de política, regras e limites e para ainda receber e publicar aspectos de presença. A comunicação 652
30 mostra 650 regras de política e sobrecarregadas dinâmicas

sendo recebidas pela rede 620. A comunicação 654 mostra os aspectos de presença sendo enviados e recebidos entre a plataforma de presença 630 e a rede 620. A comunicação 656 mostra um fluxo de informação de presença entre a
5 plataforma de presença 630 e a rede 620.

Conforme será apreciado com referência às Figuras 3, 4, 5 e 6, a ciência de contexto reduz a latência da rede pela redução da quantidade de dados transmitidos entre um ambiente de execução de usuário e uma plataforma de
10 presença. Isto é útil em um domínio sem fio, onde o uso de CPU, consumo de bateria e largura de banda de rede são recursos preciosos. Ainda, dado um contexto que se abstraia dos detalhes específicos de uma plataforma de presença, um aplicativo de cliente ou um habilitador é menos frágil e
15 significativamente mais resistente a mudanças subjacentes no modelo ou na semântica da plataforma de presença.

Conforme será apreciado, as Figuras 3, 4, 5 e 6 descritas acima são providas com referência a um P/CAM. Contudo, os sistemas de exemplo e os métodos aqui poderiam
20 ser igualmente aplicáveis a uma plataforma de localização e um L/CAM ou uma plataforma genérica e um x/CAM. Ainda, uma combinação destas plataformas é possível. O P/CAM, o L/CAM, o x/CAM ou uma combinação formam a camada ciente de contexto.

25 Com referência à Figura 7, os dispositivos de usuário 710 se comunicam através de uma estação base 712 com uma rede 720. Ainda, um computador de mesa 714 pode se comunicar através de uma rede de área ampla 716 com a rede 720. Uma plataforma de localização 730 é adaptada para
30 prover e armazenar dados brutos referentes à localização de

dispositivos de usuário 710 e, ainda, para receber atualizações de dispositivos de usuário 710 e armazenar esta informação.

Um servidor de localização 740 ainda é adaptado para
5 comunicação com uma rede 720 e pode prover a localização de vários clientes.

Um L/CAM 750 poderia ser um servidor independente em comunicação com uma rede 720 e com uma plataforma de localização 730. Em uma modalidade alternativa, o servidor
10 de L/CAM pode estar co-localizado no servidor de localização, conforme ilustrado pelo número de referência 755. Em modalidades adicionais, os agentes de L/CAM podem estar localizados em dispositivos, tal como o agente 760 nos dispositivos de usuário 710 ou no servidor de
15 localização, tal como o agente 762. No caso em que os agentes 760 e 762 são usados, várias funcionalidades ou toda a funcionalidade do L/CAM pode ser distribuída para os dispositivos de usuário ou o servidor de localização.

Em modalidades adicionais, o L/CAM pode ser parte da
20 plataforma de localização 730, conforme mostrado pelo L/CAM 770.

Em modalidades adicionais, o L/CAM pode ser parte da plataforma de localização 730, conforme mostrado pelo L/CAM 770.

25 Com referência à Figura 8, um ambiente genérico é provido. Na Figura 8, os dispositivos de usuário 810 se comunicam através de uma estação base 812 com uma rede 820. Ainda, um computador de mesa 814 (por exemplo, um dispositivo de computação que é similar aos ou diferente
30 dos dispositivos de usuário 810) se comunica por uma rede

de área ampla 816 com a rede 820. Também, uma plataforma genérica 830 é adaptada para o armazenamento de dados e estados para vários dispositivos. Outros servidores, tal como o servidor genérico 840, podem existir na rede e podem se comunicar pela rede 820.

Ainda, um x/CAM genérico 850 é adaptado para comunicação com a rede 820 e com a plataforma genérica 830. Em outras modalidades, o x/CAM pode estar localizado no servidor 840, e isto é mostrado como o x/CAM 855.

Em ainda outras modalidades, o x/CAM pode ter os agentes 860 ou 862, que estão localizados em dispositivos de usuário 810 ou no servidor 840, respectivamente.

Em modalidades adicionais, o x/CAM pode fazer parte da plataforma genérica 830, conforme mostrado pelo x/CAM 870.

A Figura 8 ilustra como uma plataforma, seja ela de presença, localização, genérica ou uma combinação das anteriores, pode ser abstraída para uma camada ciente de contexto usando-se mecanismos cientes de contexto ou camadas para suporte de uma multiplicidade de tipos de aplicativo ou habilitadores.

O dito acima pode ser implementado utilizando-se políticas e regras / gatilhos. Um processo relacionado a este mecanismo é provido abaixo.

De acordo com uma modalidade, um contexto ou mecanismo, seja ele de presença, localização ou genérico, pode incluir um ou mais dentre políticas, aspectos, regras e gatilhos. Cada um é descrito em detalhes abaixo. A descrição abaixo será apresentada com referência a um contexto de presença ou mecanismo. Contudo, isto não tem por significado ser limitante, e aqueles versados na

técnica apreciariam que o dito abaixo poderia ser igualmente aplicável a uma localização ou a um contexto genérico ou mecanismos.

Política:

5 A política está associada a um contexto de presença em particular em um ponto apropriado do ciclo de vida do aplicativo, para a especificação do comportamento ou do tratamento de aspectos relacionados à presença, localização ou genéricos. As políticas aumentam os fluxos de regras /
10 lógicos em termos de como eles operam, para a provisão de uma computação mais acurada e significativa de aspectos em nome de um aplicativo de cliente ou habilitador. Conforme será apreciado, uma política pode se aplicar a uma classe de aplicativos, um aplicativo individual ou mesmo a um
15 usuário e pode ser a provisionado com regulagens sobre como os aspectos são computados.

 A política pode ser expressa usando-se a Avaliação, Cumprimento e Gerenciamento de Política (PEEM) / Linguagem de Expressão de Política (PEL) da Aliança Móvel Aberta
20 (OMA). A PEL define uma gramática genérica e extensível, na qual as políticas podem ser expressas usando-se uma linguagem de conjunto de regras. A PEL é com base na requisição da Força Tarefa de Engenharia da Internet (IETF) para comentários (rfc) 4745. As condições e/ou ações
25 (conforme especificado na rfc 4745) podem ser melhoradas no escopo de PEEM, através das extensões de política comuns de OMA XDM (Gerenciamento de documento em XML), conforme detalhado em OMA-SUP-XSD_XSD_xdm_extensions-V1_0. A política também pode ser expressa na rfc 4745 da IETF.

30 Conforme será apreciado, o PEEM é um esforço contínuo

de normas pela OMA para a definição de funções comuns necessárias por seus habilitadores.

Como um exemplo, a tabela a seguir descreve as políticas de presença relevantes para uso por um contexto de presença na computação de aspectos de presença. Estas políticas têm aplicabilidade à plataforma de presença de OMA. Contudo, dado que as políticas podem ser adicionadas ou removidas do dado contexto conforme requerido e o conceito é aplicável a uma multiplicidade de políticas de presença. Na tabela abaixo, o valor padrão, se aplicável, é mostrado em *itálico*.

Política	Descrição	Valores
opt-in-source	Indica qual elemento de presença é um indicador de serviço opt-in. O valor padrão indica que opt-in não é relevante para o dado serviço de comunicação.	desejando <i>ignorar</i>
applicable-network-type	Indica o(s) tipo(s) de rede aplicável(is) para o dado serviço de comunicação.	<i>IMS</i> , SIP, <token>,...
threshold-value-equals	Estabelece um limite de operação de comparação de igualdade denominado label, com qn-elem, e value. Um valor booleano de 'true' ou '1' ou 'yes' aplicar-se-ia, se a política fosse aplicada a xml-ns e o alvo combinasse com o valor resultante.	<label> <qn-elem> <value>
threshold-	Idêntico à igualdade, com	<label> <qn-elem>

value-less-than	exceção do operador de comparação que é menor do que (<).	<value>
threshold-value-greater-than	Idêntico à igualdade, com exceção do operador de comparação que é maior do que (>).	<label> <qn-elem> <value>
unavailable-activities-set	Indica o subconjunto de atividades da perspectiva do observador que tornariam um contato indisponível. Este conjunto pode ser definido como vazio, o que é uma indicação que as atividades não têm importância sobre a disponibilidade.	ocupado, feriado, refeição, em trânsito, ausência permanente, dormindo, desconhecido , na igreja
undef-servcaps-sub-elements	Indica como interpretar a ausência ou a omissão de subelementos específicos <servcaps> nos metadados de presença.	desconhecido não suportado
undef-barring-state	Indica como interpretar a ausência ou a omissão de subelementos específicos <barring-state> nos metadados de presença.	ignorar ativo terminado
undef-registration-state	Indica como interpretar a ausência ou a omissão de <registration-state> nos metadados de presença.	ignorar ativo terminado

undef- willingness	Indica como interpretar a	(open,indefinite)	
	ausência ou a omissão de	(closed,indefinite)	
	<willingness> para o dado	(open,time-ofs-value)	
	serviço de comunicação.	(closed,time-ofs-value)	

TABELA 1: Políticas de Presença

A Tabela 1 acima define várias políticas e valores para as políticas. Conforme indicado na tabela, existem várias políticas, e a descrição da política e dos valores é
5 provida.

Na primeira linha da tabela, uma primeira política é "opt-in-source". A política é usada para indicar qual elemento de presença é um indicador de serviço opt-in. O valor padrão indica que opt-in não é relevante para o dado
10 serviço de comunicação.

Os valores que são possíveis para a política opt-in-source são desejando ou ignorar. Conforme será apreciado, estes poderiam ser selecionados por várias entidades, tal como o provedor de serviços, dentre outras. A entidade que
15 escolhe a política pode escolher quais valores utilizar. Assim, por exemplo, o provedor de serviços poderia escolher ignorar a fonte de opt-in para a primeira política.

A segunda política descrita na Tabela 1 é applicable-network-type e indica os tipos de rede aplicáveis para um
20 dado serviço de comunicação. Um padrão, conforme mostrado, é IMS. Contudo, outros valores incluem protocolo de iniciação de sessão (SIP) ou uma ficha, e podem ser escolhidos pela entidade de seleção.

A terceira política é "threshold-value-equals", e
25 poderia ser utilizada para o estabelecimento de um limite de operação de comparação de igualdade denominado rótulo

(label) com um elemento de XML de nome qualificado e um valor (value). Um valor booleano de um ou verdadeiro ou sim aplicar-se-ia, se a política fosse aplicada no espaço de nome de XML e o alvo resultante combinasse com o valor.

5 A próxima política na Tabela 1 é "threshold-value-less-than". Isto é similar à política threshold-value-equals, exceto pelo fato de utilizar o comparador menor do que.

10 De modo similar, a próxima política é "threshold-value-greater-than" a qual é similar às políticas de valor de limite mencionadas acima, exceto pelo operador maior do que.

15 A próxima política é "unavailable-activities-set", e poderia incluir um subconjunto de atividades que tornaria o contato indisponível no contexto do aplicativo, serviço ou habilitador. Na regulação padrão, isto é desconhecido, mas poderia incluir coisas como ocupado, feriado, refeição, dentre outros.

20 A próxima política é "undef-servcaps-sub-elements" e indica capacidades indefinidas de serviço, e como o aplicativo tem que as interpretar. Por exemplo, a Tabela 1 indica que se a capacidade de serviço for indefinida, ela poderia ser considerada como sendo não suportada.

25 A próxima política na Tabela 1 é "un-def-barring-state" e indica como interpretar a ausência ou a omissão de um elemento de XML de estado de exceção em metadados de presença, e poderia incluir que o estado é ativo ou terminado. O padrão é que o estado seja ignorado.

30 De modo similar, um "undef-registration-state" indica como interpretar a ausência ou a omissão de um elemento de

XML de registrar estado, e por padrão é ignorado, mas também poderia ser ativo e terminado no exemplo acima da Tabela 1.

A política final definida na Tabela 1 acima é "undefined-willingness" e indica como interpretar a ausência ou a omissão de um elemento de XML de desejo para um dado serviço de comunicações, e poderia incluir um par consistindo em um estado (aberto ou fechado) juntamente com um período de validade (um período indefinido ou um período de validade pré-regulado).

Conforme será apreciado por aqueles versados na técnica, a Tabela 1 acima é meramente significativa como um exemplo, e outras políticas são possíveis, com base nas necessidades de um sistema ou usuário.

Para suportar as políticas na tabela precedente, o P/CAM requer tipos adicionais de XML e definições de elementos, de modo a se estenderem as "ações" de política comum de PEL. O documento de esquema em XML a seguir provê maiores detalhes relativos a como estas ações podem ser estendidas para uso por um P/CAM.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

    targetNamespace="urn:oma:xml:xm:extensions:cam"

    xmlns="urn:oma:xml:xm:extensions:cam"

    xmlns:xs="http://www.w3.org/2001/XMLSchema"

    elementFormDefault="qualified" attributeFormDefault="unqualified">

    <!--Esta importação traz o atributo de linguagem XML xml:lang -->

    <xs:import namespace="http://www.w3.org/XML/1998/namespace"

        schemaLocation="http://www.w3.org/2001/xml.xsd"/>

    <!-- Extensões de elemento filho de "ações" específicas de P/CAM -->
```

```

<!-- namespace urn:ietf:params:xml:ns:common-policy -->

<xs:element name="opt-in-source" type="OptInSourceType"/>

<xs:element name="applicable-network-type" type="ApplicableNetworkType"/>

<xs:element name="threshold-value-equals" type="ThresholdEqType"/>

5 <xs:element name="threshold-value-less-than" type="ThresholdLtType"/>

<xs:element name="threshold-value-greater-than" type="ThresholdGtType"/>

<xs:element name="unavailable-activities-set" type="UnavailActivityType"/>

<xs:element name="undef-servcaps-sub-elements"

    type="UndefServCapsSubElemsType"/>

10 <xs:element name="undef-barring-state" type="UndefBarringStateType"/>

<xs:element name="undef-registration-state"

    type="UndefRegistrationStateType"/>

<xs:element name="undef-willingness" type="UndefWillingnessType"/>

<!--Definições de tipo para este documento -->

15 <!-- Indicador de OptInSource -->

<xs:simpleType name="OptInSourceType">

    <xs:annotation>

        <xs:documentation>

            Policy: opt-in-source

20            Os serviços associados utilizam 'willing', ou 'ignore' como indicador

            opt-in indicator.

            O padrão é 'ignore' que significa que nenhum indicador opt-in é

            relevante.

            </xs:documentation>

25        </xs:annotation>

        <xs:restriction base="xs:token">

            <xs:pattern value="willing|ignore"/>

        </xs:restriction>

    </xs:simpleType>

30 <!-- NetType -->

```

```

<xs:simpleType name="NetType">
  <xs:restriction base="xs:string">
    <xs:pattern value="IMS|SIP|[a-zA-Z][a-zA-Z0-9][a-zA-Z0-9] +"/>
  </xs:restriction>
5 </xs:simpleType>

<!-- Indicador ApplicableNetworkType -->
<xs:simpleType name="ApplicableNetworkType">
  <xs:annotation>
    <xs:documentation>
10 Policy: applicable-network-type

    Indicador do tipo de rede aplicável para os dados
    serviços de comunicação.

    </xs:documentation>
  </xs:annotation>

15 <xs:list itemType="NetType"/>
</xs:simpleType>

<!-- Tipos de indicador de limite -->
<xs:complexType name="BaseThresholdType" abstract="true">
  <xs:annotation>
20 <xs:documentation>

    Definição dos tipos base para os tipos de limites. Especifica a 'label'
    que é usada para identificar o limite específico, junto com o nome qualificado.

    </xs:documentation>
  </xs:annotation>

25 <xs:all>

  <xs:element name="label" type="xs:token"/>

  <xs:element name="qn-elem" type="xs:QName"/>

  <xs:element name="value" type="xs:anyType"/>

  </xs:all>

30 </xs:complexType>

```

```
<xs:complexType name="ThresholdEqType">
```

```
  <xs:annotation>
```

```
    <xs:documentation>
```

```
      Policy: threshold-value-equals
```

5 Comparação do limite de operação (igualdade) para 'label' para o nome do elemento qualificado 'qn-elem' com o valor especificado como 'value'.

```
    </xs:documentation>
```

```
  </xs:annotation>
```

```
  <xs:complexContent>
```

```
    <xs:extension base="BaseThresholdType"/>
```

```
  </xs:complexContent>
```

```
</xs:complexType>
```

```
<xs:complexType name="ThresholdLtType">
```

```
  <xs:annotation>
```

```
    <xs:documentation>
```

```
      Policy: threshold-value-less-than
```

15 Comparação do limite de operação (inferior a) para 'label' para o nome do elemento qualificado 'qn-elem' com o valor especificado como 'value'.

```
    </xs:documentation>
```

```
  </xs:annotation>
```

```
  <xs:complexContent>
```

```
    <xs:extension base="BaseThresholdType"/>
```

```
  </xs:complexContent>
```

```
</xs:complexType>
```

```
<xs:complexType name="ThresholdGtType">
```

```
  <xs:annotation>
```

```
    <xs:documentation>
```

```
      Policy: threshold-value-greater-than
```

25 Comparação do limite de operação (superior a) para 'label' para o nome do elemento qualificado 'qn-elem' com o valor especificado como 'value'.

30

```

        </xs:documentation>

    </xs:annotation>

    <xs:complexContent>

        <xs:extension base="BaseThresholdType"/>

5    </xs:complexContent>

</xs:complexType>

<!-- Indicador de atividades indisponíveis -->

<xs:simpleType name="UnavailActivityType">

    <xs:annotation>

10    <xs:documentation>

        Policy: unavailable-activities-set

        Usado para descrever todas as atividades relacionadas a um aplicativo ou
        habilitador que deveria fornecer um individuo não disponível.

    </xs:documentation>

    </xs:annotation>

15    <xs:list itemType="xs:QName"/>

</xs:simpleType>

<!-- Indicador de UndefServCapsSubElems -->

<xs:simpleType name="UndefServCapsSubElemsType">

20    <xs:annotation>

        <xs:documentation>

            Policy: undef-servcaps-sub-elements

            Indica como interpretar a ausência ou omissão de subelementos

            &lt;servcaps&gt; específicos na presença de metadados. Valor

25    'unknown' é considerado padrão que não dá ao conteúdo qualquer dica de
            como lidar com falta ou ausência de subelementos &lt;servcaps&gt;..

        </xs:documentation>

    </xs:annotation>

    <xs:restriction base="xs:token">

30    <xs:pattern value="unknown|unsupported"/>

```

```

    </xs:restriction>

</xs:simpleType>

<!-- Indicador de UndefBarringState -->

<xs:simpleType name="UndefBarringStateType">
5      <xs:annotation>

        <xs:documentation>

            Policy: undef-barring-state

            Indica como interpretar a ausência ou omissão de subelementos

            &lt;barring-state&gt; específicos na presença de metadados.
10      </xs:documentation>

        </xs:annotation>

        <xs:restriction base="xs:token">

            <xs:pattern value="ignore|active|terminated"/>

        </xs:restriction>

15    </xs:simpleType>

<!-- Indicador de UndefRegistrationState -->

<xs:simpleType name="UndefRegistrationStateType">

    <xs:annotation>

        <xs:documentation>

20      Policy: undef-registration-state

            Indica como interpretar a ausência ou omissão de subelementos

            &lt;registration-state&gt; específicos na presença de metadados.

            O valor padrão 'ignore' indica que o subelemento não tem

            significado neste contexto.

25      </xs:documentation>

        </xs:annotation>

        <xs:restriction base="xs:token">

            <xs:pattern value="ignore|active|terminated"/>

        </xs:restriction>

30    </xs:simpleType>

```

```

<!-- Indicador de UndefinedWillingnessType -->

<xs:simpleType name="UndefinedWillingnessType">

  <xs:annotation>

    <xs:documentation>

      Policy: undef-willingness

      Indica como interpretar a ausência ou omissão do subelemento

      <willingness> para um dado serviço.

      O valor padrão é 'closed/indefinite'.

    </xs:documentation>

  </xs:annotation>

  <xs:restriction base="xs:token">

    <xs:enumeration value="open/indefinite"/>

    <xs:enumeration value="closed/indefinite"/>

    <xs:enumeration value="open/time-ofs-value"/>

    <xs:enumeration value="closed/time-ofs-value"/>

  </xs:restriction>

</xs:simpleType>

</xs:schema>

```

O esquema em XML acima provê a definição de nome de elemento nas linhas que começam com `<xs:element name="opt-in-source" type="OptInSourceType"/>`. Os nomes de elemento são adicionalmente definidos para as políticas remanescentes na Tabela 1 acima.

Conforme pode ser visto por aqueles versados na técnica, o restante do Esquema XML acima define os tipos de política conforme indicado pela descrição e os campos de valor na Tabela 1. Especificamente, para o "OptInSourceType", um valor `xs:pattern` é regulado para desejando ou ignorar. Portanto, o dito acima provê as definições adicionais de tipo de XML e elemento, de modo a

se estenderem as ações de política comum de PEL.

Pela extensão de ações de política comuns, as políticas de P/CAM podem ser incorporadas em um documento de XML de 'conjunto de regras' (ruleset) de PEL de política comum. Um 'conjunto de regras' pode se aplicar em um escopo de usuário ou um escopo global. Por exemplo, o 'conjunto de regras' pode se aplicar a uma classe de serviço ou a um aplicativo específico, o conjunto de regras também pode se aplicar a um usuário individual ou a um grupo de usuários.

As políticas relacionadas a P/CAM são manipuladas e avaliadas através das várias interfaces de requisitante de PEEM pelo servidor de P/CAM em si ou um cliente / agente habilitado para P/CAM. Isto é, protocolos de aplicativo ou de alongamento total podem prover os metadados específicos, tal como a identidade de requisitante, para a interface requisitante de PEEM, juntamente com outros metadados disponíveis para os servidores de PEEM como a base para aplicação de regras.

O que vem a seguir é um exemplo de um documento em XML de conjunto de regras de PEL de política comum, o qual consiste em uma regra única 'a101'. Esta regra se associa a um habilitador de serviço, tal como um alerta de PoC, e define regulagens / valores de política específicos a serem aplicados como resultado de uma combinação para um recurso alvo. Nesse caso, o recurso alvo é o identificador de serviço em si. Conforme será apreciado por aqueles versados na técnica, este exemplo faz uma correlação intencional entre o valor atributo 'ext:service[@enabler]' da extensão de política comum e o service-id de alerta de PoC, conforme definido pela presença de OMA.

O dito acima é ilustrado com referência à Figura 12, a qual mostra como uma camada cliente (AL), tal como uma camada cliente de contexto (CAL), por exemplo, pode pré-carregar um dado conjunto de tipo de política XSD. Conforme
 5 será apreciado, estes são tipos conforme mostrado pela Tabela 1 acima.

Um dispositivo de cliente de AL 1210 se comunica com um AL 1212, o qual se comunica com um PEEM 1214.

O AL 1212 envia uma mensagem
 10 loadPolicyExtension(xsd,service-id) 1220 para o PEEM 1214, a qual é processada, conforme mostrado pela seta 1222. O PEEM 1214 então envia uma mensagem de aceitação 1224 para o AL 1212.

Em algum ponto mais tarde, o dispositivo de cliente
 15 habilitado para AL 1210 tenta iniciar e se autenticar junto ao habilitador de serviço de AL 1212, tal como um serviço de alerta de PoC. Isto é feito com a mensagem authenticate (watcher-id, service-id, user-id) 1230.

Como parte da iniciação e da autenticação, o AL 1212
 20 envia uma mensagem pellnit (watcher-id, service-id, user-id) 1240 para o PEEM 1214. O PEEM 1214 avalia a política, conforme mostrado pela seta 1242, e retorna a política na mensagem 1244. A avaliação 1242 permite que o PEEM aplique um conjunto específico de regulagens de política em uma
 25 base por servidor ou por usuário.

O AL 1212 inicia a seta de contexto 1244 e ainda retorna, opcionalmente, o contexto de AL como uma mensagem 1250 de volta para o dispositivo de cliente de AL 1210.

É possível que, como um exemplo, o critério de
 30 combinação pudesse ser o service-id relacionado a um

habilitador de OMA (tal como um alerta de PoC). Outros critérios de combinação poderiam ser com base em uma esfera de usuário ou de grupo.

```

5      <?xml version="1.0" encoding="UTF-8"?>

      <!-- Conjunto de regras de política para serviço de alerta de PoC de OMA. -->

      <!-- Um conjunto de regras pode se aplicar em uma base por usuário ou global. -->

      <cr:ruleset      xmlns="urn:ietf:params:xml:ns:common-policy"

                      xmlns:ext="urn:oma:xml:xdm:extensions"

                      xmlns:cr="urn:ietf:params:xml:ns:common-policy"

10      xmlns:cs="urn:oma:xml:xdm:extensions:cam"

      xmlns:rp="urn:ietf:params:xml:ns:pidf:rp">

      <!-- Uma regra para serviço de alerta de PoC, estabelecer políticas de
contexto -->

      <cr:rule id="a101">

15      <cr:conditions>

          <ext:service-list>

              <!-- Combinação com relação a um habilitador de OMA específico
por service-ID... -->

              <ext:service enabler="org.openmobilealliance.PoC-alert"/>

20      </ext:service-list>

          </cr:conditions>

          <cr:actions>

              <!-- Valores de política a seguir para escopo de documento... -->

              <cs:undef-servcaps-sub-elements>

25      unsupported

              </cs:undef-servcaps-sub-elements>

              <cs:undef-willingness>

                  closed/indefinite

              </cs:undef-willingness>

30      <cs:opt-in-source>willing</cs:opt-in-source>

```

```

5
    <cs:unavailable-activities-set>
        rpid:busy rpid:sleeping
    </cs:unavailable-activities-set>
    <cs:undef-registration-state>
        terminated
    </cs:undef-registration-state>
    <cs:undef-barring-state>
        ignore
    </cs:undef-barring-state>
10    <cs:applicable-network-type>
        IMS
    </cs:applicable-network-type>
    </cr:actions>
    </cr:rule>
15 </ruleset>

```

Conforme será apreciado por aqueles versados na técnica, o dito acima define a regra 'a101'. Neste caso, o service-id é definido como "org.openmobilealliance.PoC-alert", o serviço de alerta de PoC de OMA, e as extensões de política de P/CAM são definidas como parte do namespace de XML "urn:oma:xml:xdm:extensions:cam". Portanto, o dito acima é uma manifestação do esquema definido com respeito à Tabela 1 acima. Os valores de camada ciente de contexto baseados no disparo de regra 'a101' são mostrados abaixo com referência à Tabela 1A.

Política	Valor
opt-in-source	"desejando"
applicable-network-type	IMS
unavailable-activities-set	rpid:busy rpid:sleeping
undef-servcaps-sub-elements	"não suportado"

undef-barring-state	"ignorar"
undef-registration-state	"terminado"
undef-willingness	(closed, indefinite)

TABELA 1A - Regulagem de Política / Valores (Serviço de Alerta de PoC de OMA)

Conforme será apreciado, o PEEM poderia utilizar múltiplas políticas de aplicativo e múltiplos serviços ou exclusões poderiam ser estabelecidos como parte de um conjunto de regra.

As ações conforme visto na XML acima definem valores de política específicas para um escopo de documento.

Aspectos:

Os aspectos são abstrações de nível de aplicativo relevantes para uma fonte, por exemplo, dispositivos de nó associados são abstrações de nível de aplicativo relevantes para a presença. Os aspectos de presença podem ser considerados a interface conceitual de um contexto de presença para um aplicativo de cliente de P/CAM ou habilitador. A Tabela 2 abaixo esboça um conjunto de base de aspectos de presença aplicáveis que podem ser incorporados para uso por um mecanismo ciente de contexto de presença e expostos para aplicações de cliente. Para cada aspecto de presença, é provida uma descrição, juntamente compensação de imagem as associações a que o aspecto se refere em termos do modelo de dados de presença padronizados esboçados na rfc 4479 da IETF.

Em particular, para especificação e aplicação de um comportamento relevante de forma contextual através de um conjunto díspar de componentes de intertrabalho e dispositivos de usuário, um mecanismo geral é requerido

para a encapsulação de aspectos relacionados a uma plataforma de presença. Isto é, um aspecto captura uma abstração de primeira ordem relacionada a um dado aplicativo ou habilitador. Os aspectos relativos a uma
5 plataforma de presença descreveriam ou se relacionaram a indicações subjacentes de presença. Os aspectos podem ser expandidos para encapsularem outras indicações também. Por exemplo, uma localização pode ser incorporada (ou inferida) para a derivação ou a computação de um aspecto associado em
10 uma plataforma de presença. Isto é ilustrado na Tabela 2 abaixo com respeito ao aspecto quem está próximo (who-is-nearby).

A presente exposição provê um mecanismo para um número arbitrário de aspectos, conforme requerido pela plataforma
15 de presença. Estes podem incluir aspectos comuns, tais como disponibilidade e alcançabilidade. Eles também podem incluir aspectos específicos de aplicativo. Um mecanismo na plataforma de presença ou na interface de gerenciamento existe para associação de um conjunto apropriado de
20 aspectos com um dado serviço. A associação de aspectos de natureza contextual pode se aplicar em níveis diferentes. Por exemplo, um dado aspecto pode se aplicar a um habilitador de serviço, tal como todo serviço em conformidade com *push-to-talk* por celular (isto é, PoC) da
25 OMA.

Um aspecto também pode ser aplicável a um nível de usuário ou de grupo.

Para cada aspecto, um conjunto associado de regras ou lógica pode ser definido, o que esboça as etapas ou um
30 processamento requerido para a obtenção do dado aspecto. A

lógica também identifica os indicadores / elementos de presença / dados brutos relevantes para o cálculo do aspecto associado. Um dado aspecto pode combinar duas ou mais regras pré-definidas em conjunto como parte de seu processamento lógico. Ainda, uma lógica subjacente pode ser reusada como uma biblioteca ou como rotinas no suporte de aspectos em uma plataforma de presença. Esta biblioteca pode incluir aspectos como outros módulos de nível alto ou componentes, os quais podem ser incorporados. Isto permite que múltiplos tipos de aplicativo de cliente utilizem uma camada ciente de contexto.

Em uma modalidade, os aspectos de presença são extensíveis. Por exemplo, se um dado serviço ou habilitador requerer uma funcionalidade específica, a plataforma de presença poderia suportar a extensão ou a redefinição em um ou mais aspectos, conforme requerido.

Conforme será apreciado por aqueles versados na técnica, a Tabela 2 pode ser modificada ou estendida para suportar outras políticas de presença ou exigências de aplicativo / habilitador. Os aspectos de presença em particular mostrados na Tabela 2 são demonstrativos de uma plataforma de presença de OMA.

Aspecto de Presença	Descrição	Associações	Visibilidade	Visibilidade Comum
opt-in	A presentidade está desejando participar em uma sessão para um dado serviço ou aplicativo.	Pessoa -> serviço.	OTA, Servidor	Servidor

Available	A presentidade disponível para comunicação usando um dado serviço ou aplicativo.	Pessoa -> serviço.	OTA, Servidor	Servidor
contact-means	Método de contato mais aplicável de presentidades para um dado serviço ou aplicativo.	Pessoa (end.) -> serviço.	OTA, servidor	Servidor
contactable	A presentidade está desejando, disponível, e tem um meio de contato válido atualmente para um dado serviço ou aplicativo.	Pessoa (end.) -> serviço.	OTA, servidor	Servidor
reachable	A presentidade é contatável para um dado serviço ou aplicativo. NOTA: Uma indicação positiva para alcançável	Pessoa -> serviço -> dispositivo	OTA, servidor	OTA

	indica que uma presentidade está desejando, disponível, contatável e seu dispositivo está em cobertura para o estabelecimento de comunicação pelo serviço definido.			
where-are-you	Localização atual de presentidades.	Pessoa, Pessoa -> serviço -> dispositivo	OTA, servidor	OTA
personal-avatar	Representação icônica pessoal atual de presentidades.	Pessoa	OTA, servidor	OTA
service-avatar	Representação icônica pessoal atual de presentidades para um dado serviço ou aplicativo.	Pessoa -> serviço	OTA, servidor	OTA
personal-interests	Interesses ou hobbies atuais de	Pessoa (inf. estendida)	OTA, servidor	Servidor

	presentidades.			
who-is-subscribing-to-me	Observadores que atualmente têm assinaturas 'pendentes' para uma dada presentidade.	Winfo	OTA, servidor	Servidor
who-is-nearby	Uma lista de zero ou mais presentidades que estão em grande proximidade e se adequam a um conjunto opcional de critérios (por exemplo, interessadas em futebol americano).	Pessoa -> serviço	OTA, servidor	Qualquer
who-is-blocked	Observadores que tenham tido assinaturas terminadas ou tenham sido bloqueados por uma dada presentidade.	Winfo, política comum	OTA, servidor	Servidor
eligible-	Se uma	Pessoa ->	OTA,	Servidor

session-participant	presentidade é alcançável e se adéqua a um conjunto opcional de critérios, de modo a participar em uma sessão do serviço associado.	serviço -> Servidor dispositivo, Shared UserProfile, Outros metadados de XDMS		
Session-answermode	Um indicador quanto a se uma presentidade aceitará uma sessão entrando para um dado serviço em modo automático (sem intervenção) ou manual (o usuário deve aceitar / rejeitar).	Pessoa -> serviço	OTA, Servidor	OTA

TABELA 2: Aspectos de Presença

A Tabela 2 define vários aspectos de presença / aplicativo / serviço aplicáveis a uma plataforma de presença. Para cada aspecto, há uma descrição curta
5 juntamente com a associação ou aplicabilidade do aspecto ao modelo de dados de presença padronizado. Além disso, a

visibilidade é declarada. A visibilidade descreve o ponto aplicável no qual o aspecto associado é referido. Uma visibilidade comum define ou declara o ponto mais comum ou relevante no qual o aspecto associado provavelmente será
5 referido. As escolhas para visibilidade incluem pelo ar (OTA) versus servidor. Conforme seria apreciado, "servidor" surgiria no lado de rede em um servidor de aplicativo.

Na primeira linha da Tabela 2 acima, o aspecto opt-in é definido, o qual indica que a presentidade está desejando
10 participar em uma dada sessão para um dado serviço ou aplicativo. Conforme indicado na Tabela 2, a pessoa está associada ao serviço.

Uma segunda linha da Tabela 2 indica que um aspecto de presença está 'disponível'. Este aspecto indica que a
15 presentidade está disponível para comunicação usando um dado serviço ou aplicativo e, novamente, há uma associação entre a pessoa e o serviço.

A próxima linha na Tabela 2 indica o aspecto de presença de contact-means. Um método mais aplicável de
20 presentidade de contato para um dado serviço ou aplicativo é provido, e a associação é entre o endereço da pessoa e o serviço.

A próxima linha na Tabela 2 indica o aspecto de 'contactable'. Este aspecto mostra se a presentidade está
25 desejando, disponível e tem um meio de contato válido atualmente para um dado serviço ou aplicativo. De novo, neste caso, a associação é entre o endereço de uma pessoa e o serviço.

A próxima linha na Tabela 2 indica o aspecto de
30 'reachable'. Isto mostra que a presentidade é contatável

para um dado serviço ou aplicativo. Uma indicação positiva para reachable mostra que a presentidade está desejando, disponível, contatável e que seu dispositivo está na cobertura para o estabelecimento de uma comunicação pelo serviço definido. A associação é, portanto, entre a pessoa, o serviço e o dispositivo.

'Where-are-you' é o próximo aspecto definido na Tabela 2, e mostra a localização atual da presentidade. Conforme indicado, a associação para este aspecto é na pessoal, e na pessoa, no serviço e no dispositivo.

Outros aspectos são adicionalmente definidos na Tabela 2, e incluem várias associações a eles.

Para uma realização de presença de OMA, um fluxograma de chamada de plataforma de presença de exemplo pode parecer com aquilo mostrado na Figura 13. Aqueles versados na técnica apreciarão que a Figura 13 mostra que a camada cliente de contexto pode ser configurada entre um dispositivo de cliente e a camada de presença de OMA / XDM. Em uma modalidade, a camada de acesso pode ser uma camada de aplicativo ou proxy. Uma camada cliente de contexto como essa poderia ser uma camada separada ou uma camada interna do aplicativo (por exemplo, um aplicativo de propaganda móvel com uma camada cliente de contexto dividida ou integrada).

Conforme mostrado na Figura 13, o aspecto "alcançável" pode incluir, na back end, um processamento adicional, o que incorpora regras e, possivelmente, o uso de outros aspectos na computação. Conforme citado previamente, estes aspectos podem existir em uma biblioteca padrão de aspectos para reutilização em aplicativos de nível mais alto ou

aspectos de serviço, quando requerido.

Uma referência é feita, agora, à Figura 13. A Figura 13 mostra um dispositivo de cliente 1310 o qual se comunica com uma camada de acesso (AL) 1312 (por exemplo, uma camada
5 cliente de contexto (CAL)), a qual, por sua vez, comunica-se com uma entidade de OMA PRS/XDM 1314.

O dispositivo de cliente 1310 envia uma consulta concernente ao aspecto de presença "alcançável", mostrado como a comunicação 1320. Por sua vez, a camada de acesso
10 (AL) 1312 envia uma requisição HTTP/GET 1322 para a OMA PRS/XDM 1314.

A OMA PRS/XDM 1314 autentica, conforme mostrado por 1330 e retorna uma resposta na forma de HTTP/1.1 <pdf>
1332.

15 A camada de acesso (AL) 1312 então checa se o processo é alcançável conforme mostrado pela seta 1340. O processamento na AL para o aspecto "alcançável" invoca outras regras, tais como "contactable", "contact-means", "available" e "opt-in or willing".

20 A seta mostrada por 1340 determina que a presentidade é inalcançável, e retorna isto na mensagem 1350.

Conforme mostrado na Figura 13, a consulta de alcançável 1320 e a resposta de inalcançável 1350 viajam pelo ar. Contudo, isto tem significado apenas como um
25 exemplo, e outras técnicas de comunicações seriam aplicáveis, em modalidades diferentes.

Regras / Gatilhos:

Uma terceira ramificação da solução de mecanismo cliente de contexto consiste em regras e/ou gatilhos. O
30 exemplo abaixo usa a presença como um exemplo.

As regras residem em um contexto de presença e estabelecem uma sequência de etapas ou fluxos lógicos requeridos para a computação de aspectos de presença com base nos metadados providos pela plataforma de presença subjacente. As regras são conceitualmente similares aos procedimentos armazenados em banco de dados ou a funções definidas por usuário (UDFs). Regras de presença de base ou padronizadas podem ser mudadas ou suplementadas por um cliente de aplicativo ou um usuário individual. Por exemplo, a injeção por um cliente de regras dinâmicas pode suprimir ou estender o comportamento de regra de base. Além disso, as regras incorporam políticas associadas ao contexto de presença pelo aplicativo ou habilitador para aumento ou provisão de sugestões circundando a interpretação de metadados. Isto permite que um aplicativo ou serviço afete diretamente o resultado de um ou mais aspectos de presença, conforme requerido.

A Tabela 3 mostra abaixo um conjunto de regras referentes à computação de aspectos relacionados à presença com uma pseudológica específica para a plataforma de presença de OMA. Deve ser notado que isto é apenas um subconjunto de regras / lógica que pode ser exposto por um contexto de presença. É possível mudar a composição ou a granularidade de regras, conforme requerido pelo contexto de presença. Além disso, conforme citado com referência às Figuras 3 a 6 acima, é possível que uma presentidade ou um observador continue a buscar ou ser notificado de uma informação de presença bruta pela plataforma de presença subjacente, de modo a atingir conclusões específicas, se um contexto não estiver disponível. Isto poderia ocorrer,

conforme seria apreciado, em situações específicas.

Conforme usado na Tabela 3 abaixo, 'def' indica definido e significa que a entidade existe e é estabelecida com valores razoáveis, ao passo que 'undef' significa "indefinido" - o complemento de 'def'. Assim, 'undef' tem valores como nil (nada), null (nulo), ou invalid (inválido).

Na Tabela 3 abaixo, 'valid' (válido) significa que a entidade associada ainda contém dados tempestivos ou significativos.

Regra	Descrição	Pseudológica
findServicePresInfo	Retorna o elemento de informação de presença mais aplicável 'svc' para o dado serviço ou aplicativo na 'list' (lista) de serviço. NOTA: o método de pseudológica 'resolveService()' implementa uma semântica esboçada em OMA-TS-Pres V2_0 Seção (5.2.3).	▪ For each <tuple> 't' in list with t.service-id == service-id ○ Items.add('t') ▪ If Items.size == 1 ○ Res=Items[0] ▪ Else ○ Res=resolveService(Items) ▪ Return Res
hasOptedInForService	Faz uso da política opt-in-source para o estabelecimento	▪ Switch (opt-in-source policy) ▪ Case willing: ○ Uwp=undef-willingness

	de um <i>desejo</i> de usuário 'p' para comunicação dado um serviço ou aplicativo 'svc'. O desejo é um par ordenado (open closed, indefinite time-ofs-value). NOTA: o método de pseudológica implementa uma semântica esboçada em OMA-TS-Pres V2_0 Seção (10.4.1).	<pre> policy o If svc.willingness undef ▪ Return Uwp o Else ▪ Return svc.willingness ▪ Case session-participation: o Return Willingness(svc.session -participation, indefinite) ▪ Default: // ignore o Return Willingness(open, indefinite) </pre>
isAvailable	Retorna um valor booleano indicando se uma presentidade 'p' está <i>disponível</i> para comunicação para um dado serviço ou aplicativo 'svc'. NOTA: o método de pseudológica implementa uma semântica esboçada	<pre> ▪ Urs=undef-registration-state policy ▪ Ubs=undef-barring-state policy ▪ Uas=unavailable-activities- set policy ▪ If (p.activities valid and <activities> non-empty-set) o For each <activities> 'a' in p: ▪ If ('a' match 1+ element in Uas) ▪ Return </pre>

	em OMA-TS-Pres V2_0 Seção (10.4.3). A lógica neste método também se fatora em atividade (se dirigida por política) em um cálculo de disponibilidade.	false ▪ If (svc.reg-state undef) ○ If (Urs == 'ignore') ▪ Reg-state=active ○ Else ▪ Reg-state=Urs ▪ Else ○ Reg-state=svc.reg-state ▪ If (svc.bar-state undef) ○ If (Ubs == 'ignore') ▪ Bar-state=active ○ Else ▪ Bar-state=Ubs ▪ Else ○ Bar-state=svc.bar-state ▪ If (Reg-State == 'active' AND Bar-state == 'active' AND svc.status.basic == 'open') ○ Return true ▪ Return false
establishContactMeans	Retorna um contato aplicável 'c' para um dado serviço ou serviço de aplicativo 'svc'.	▪ Return svc.contact

	NOTA: a pseudológica segue a rfc 3863.	
isContactable	Retorna um ContactMeans válido consistindo na tupla (contact, ldev, validity) se uma presentidade 'p' for contatável para um dado serviço ou aplicativo 'svc'. NOTA: o método de pseudológica implementa uma semântica esboçada em OMA-TS-Pres V2_0 Seção (10.4).	▪ W = - hasOptedInForService(p, svc) ▪ If (W valid AND - isAvailable(p, svc)) ○ C = - establishContactMeans(svc) ○ If (C def AND - svc.deviceID def) ▪ Cm=ContactMeans(Contact, svc.deviceID(s), w.validity) ▪ Return Cm
isReachable	Retorna um valor booleano indicando se um dispositivo aplicável 'dev' pode ser alcançado pelo tipo de rede requerido dado um contact-means contatável.	▪ Ant=applicable-network-type policy ▪ If (cm valid) ○ For each 'd->deviceID' in ldev: ▪ Find 'dev' in <device> elements where dev.deviceID == 'd->deviceID'

		▪ If match ▪ For each <network> 'n' in 'dev': ▪ If ('n'.id - match 1+ - element in - Ant and 'n' - available) ▪ Return - true ▪ Return false
--	--	---

Tabela 3: Regras

A Tabela 3 acima descreve várias regras. A primeira regra definida é 'findServicePresInfo', a qual retorna o elemento de informação de presença mais aplicável para o

5 dado serviço ou aplicativo em uma lista de serviço. Conforme indicado na pseudológica, para cada tupla t na lista, uma checagem é feita para se ver se o service-id de 't' combina com o service-id desejado e, se assim for, a tupla t é adicionada a uma lista. Após isso, uma vez que a

10 compilação tenha terminado, se o tamanho de item for 1, então, aquele item será retornado. Caso contrário, a função 'resolveService' é invocada. Conforme será apreciado por aqueles versados na técnica, a função 'resolveService' é uma função específica de OMA que encontra o serviço mais

15 relevante.

Regras similares são definidas com respeito ao

restante da Tabela 3, em que várias pseudológicas são utilizadas para a definição do que será retornado, quando uma regra for implementada.

As regras de presença e/ou os fluxos lógicos podem ser especificados usando-se PEEM/PEL de OMA. O que vem a seguir é um exemplo de um documento de 'processo de abstração de PEEM/PEL, o qual caracteriza o fluxo lógico para a regra 'findServicePresInfo', conforme mostrado na pseudológica da Tabela 3 acima:

```

10      <process name="findServicePresInfo"

          targetNamespace="http://example.com/ws-bp/purchase"

          xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/abstract"

          xmlns:pcam="http://pcam.example.com/wsd/oma-pres-pcam">

15      <documentation xml:lang="EN">

          Um processo WS-BPEL para encontrar a(s) tupla(s) de serviço apropriadas(s).

          </documentation>

          <!-- Parâmetros de entrada / saída: -->

          <!-- presinfo - corpo de entrada contendo service-ID, e inf. de presença -->

          <!-- theResult - a tupla de serviço mais relevante para service-ID -->

20      <variables>

          <variable name="presinfo" messageType="##opaque"/>

          <variable name="matchingTupleList" messageType="##opaque"/>

          <variable name="theResult" messageType="##opaque"/>

          </variables>

25      <partnerLinks>

          <partnerLink name="service" partnerLinkType="##opaque"

              partnerRole="##opaque"/>

          <partnerLink name="customer" partnerLinkType="##opaque"

              partnerRole="##opaque"

30      myRole="##opaque"/>

```

```

</partnerLinks>

<sequence>

    <receive partnerLink="customer" operation="findServicePresInfoRequest"
        variable="presinfo" createInstance="yes">
5      </receive>

    <forEach counterName="i" parallel="no">

        <!-- Iterate over $presinfo.msg/tuple and find all matches -->

        <!-- between $presinfo.msg/service-id and -->

        <!-- $presinfo.msg/tuple[i]/service-description/service-id -->
10      <!-- Store in matchingTupleList -->

    </forEach>

    <if>

        <condition opaque="yes">$matchingTupleList.num-items == 1</condition>

        <flow>
15      <!-- $theResult is the first item in $matchingTupleList -->

        </flow>

        <else>

            <!-- $theResult é o resultado de invocar resolveService -->

            <!-- método com $matchingTupleList -->
20      </else>

    </if>

    <reply partnerLink="service" portType="##opaque"
        operation="##opaque" variable="theResult">

    </reply>
25 </sequence>

</process>

```

A outra porção da ramificação de regras / gatilhos é gatilhos. Os gatilhos residem em um contexto de presença e associam uma seqüência de etapas (ou fluxos lógicos) com
 30 base em uma mudança de estado de presença detectada na

plataforma de presença. Os gatilhos são conceitualmente similares aos gatilhos de banco de dados. Os gatilhos, por padrão, são inicialmente notificações. Os gatilhos podem ser definidos por um cliente de aplicativo ou um usuário individual, conforme necessário. Por exemplo, a injeção por um cliente de gatilhos dinâmicos podem suprimir ou estender o(s) comportamento(s) de gatilho de base.

A Tabela 4 lista um conjunto de gatilhos relativos à computação de aspectos relacionados à presença com uma pseudológica específica para o gatilho em particular. Deve ser notado que os aspectos também podem ser definidos com uma definição de gatilho correspondente.

Gatilho	Descrição	Pseudológica
onOptIn/Out	Gatilho definido por aplicativo, o qual é invocado quando uma presentidade é determinada como tendo optado por entrar / sair para o dado serviço ou aplicativo.	▪ notification (default)
onUn/Available	Gatilho definido por aplicativo, o qual é invocado quando uma presentidade está (in)disponível para o dado serviço ou aplicativo.	▪ notification (default)
onUn/Reachable	Gatilho definido por aplicativo, o qual é	▪ notification (default)

	invocado quando uma presentidade está (in)alcançável para o dado serviço ou aplicativo.	
<u>onNearby/onOutOfRange</u>	Invocado quando uma presentidade está próxima ou se moveu para fora de um alcance especificado para o dado serviço ou aplicativo.	▪ notification (default)
<u>on-pending-subscription</u>	Invocado quando uma presentidade tem uma ou mais assinaturas em um estado 'pendente'.	▪ notification w/list<AOR> (default)
<u>on-terminated-subscription</u>	Invocado quando uma presentidade tem uma ou mais assinaturas em um estado 'terminado'.	▪ notification w/list<AOR> (default)
<u>on-update-note</u>	Quando uma presentidade adiciona ou atualiza uma nota pessoal.	▪ notification w/note-text (default)
<u>on-is-in/eligible-session-participant</u>	Quando uma presentidade está in/alcançável e in/elegível para o dado serviço ou aplicativo.	▪ notification (default)

Tabela 4: Gatilhos

O primeiro gatilho na Tabela 4 acima indica que o gatilho será invocado quando uma presentidade optar por entrar ou sair de um dado serviço ou aplicativo. O gatilho

permite que uma funcionalidade específica seja realizada quando o estado associado ocorrer no contexto. A pseudológica pode ser definida pelo cliente de aplicativo, se o cliente desejar que o P/CAM faça alguma coisa sobre a ocorrência de um dado evento o qual é quando um gatilho é invocado.

Os outros gatilhos definidos pela Tabela 4 têm uma funcionalidade similar e são invocados em consonância com uma condição pré-definida sendo encontrada.

Os gatilhos são especificados usando-se PEEM / PEL (linguagem de expressão de política) de OMA e são substancialmente similares (na estrutura e na composição) às regras de presença. Assim, o exemplo de código usado acima com referência a regras poderia ser adaptado para os gatilhos da Tabela 4.

Os gatilhos são úteis em um sistema complexo ciente de presença. Os gatilhos provêm que uma encapsulação iniciada por rede seja definida e aplicada a um dado cenário. Os gatilhos provêm, em uma modalidade, uma notificação simples para um cliente ou serviço, ou podem incorporar uma lógica comercial complexa que é executada completamente na rede. Isto é útil em um domínio sem fio, onde uma largura de banda de rede e os recursos de processamento são limitados.

Por exemplo, um serviço de entrega de conteúdo sem fio pode requerer um comportamento específico com base no estado de usuários e em suas capacidades de dispositivo associadas. Isto é, dois usuários que tenham optado por entrar em um serviço de anúncio pequeno / alerta de esportes com dispositivos diferentes pode receber um

conteúdo de formas diferentes. Por exemplo, um primeiro usuário que tenham um dispositivo sem fio baseado em texto muito simples e apenas capaz de receber um serviço de mensagem curta (SMS) com conteúdo relacionado a beisebol e/ou um URL baseado na web apontando para informação adicional requer dados diferentes de um segundo usuário que tenha um assistente digital pessoal pleno de recursos / smartphone com uma capacidade de manipulação de mídia embutida. O segundo usuário pode receber alertas de multimídia contendo clipes de vídeo coloridos curtos de uma 'jogada do dia' esportiva.

Cada caso acima ilustra a complexidade subjacente de um serviço de entrega de conteúdo para a entrega de um conteúdo apropriado / tempestivo relevante para cada dispositivo de usuário. Isto é, um serviço de entrega de conteúdo tipicamente tem algum entendimento do estado atual de um dado usuário, juntamente com seu interesse associado, e as capacidades de dispositivo relevantes para o recebimento de conteúdo. Um serviço de entrega de conteúdo trabalhando em combinação com uma capacidade de presença ciente de forma contextual é uma plataforma como essa. Ainda, uma palavra chave ciente de forma contextual que expõe os "gatilhos de aspecto" relevantes em nome de um serviço de entrega de conteúdo provê um meio útil para motorista ou pressionar uma informação relevante para uma base de assinante associado.

Um aspecto de um gatilho associado é um "aspecto monitorado" em uma base contínua ou especificada. Isto é, quando uma entidade, seja uma pessoa ou uma entidade lógica, atinge ou se qualifica para um gatilho de aspecto.

associado, o gatilho associado "dispara", e um conjunto de lógicas ou ações ocorre. A lógica é de natureza contextual e permite que serviços e/ou ações específicas de usuário sejam definidas e executadas. Isto pode ser enviar ou pressionar uma informação relevante para o dispositivo de cliente apropriado. Como com os aspectos, os gatilhos de aspecto podem ser expandidos para encapsularem uma variedade de indicadores não de presença, tal como localização.

Os presentes sistemas e métodos incluem um mecanismo para um número arbitrário de aspectos, conforme requerido pela plataforma de serviço / presença. Isto pode incluir um conjunto de gatilhos de aspecto comuns, tais como "disponibilidade", "opt-in", "alcançável", dentre outros, bem como gatilhos específicos de aplicativo. Existe um método em uma modalidade na plataforma de presença ou em uma interface de gerenciamento para associação de um conjunto apropriado de gatilhos de aspecto com um dado serviço. A associação de gatilhos de aspecto é de natureza contextual e pode se aplicar em níveis diferentes. Por exemplo, um dado gatilho de aspecto pode se aplicar a um habilitador de serviço, tais como serviços em conformidade com o *push-to-talk* por celular PoC da OMA. Ainda, o gatilho pode ser aplicável ou delimitado em uma classe de nível de serviço. Por exemplo, isto pode aplicar "disponibilidade" a toda classe de serviços. Ainda, um gatilho pode ser aplicável em um nível de usuário ou de grupo.

Conforme será apreciado com referência às Figuras 2 e 9, a determinação quanto a se um cliente é "alcançável" é simplificada pela abstração do aspecto para a camada cliente

de contexto. Ainda, um gatilho pode invocar o aspecto ou o aspecto pode ser invocado em nome do gatilho. Isto poderia ser feito pelo habilitador de serviço subjacente, sem qualquer envolvimento de qualquer dispositivo de cliente.

- 5 Os gatilhos podem invocar aspectos definidos e/ou podem incorporar uma lógica consistindo em regras / procedimentos os quais incluem a invocação de outros aspectos.

Os gatilhos de aspecto por padrão enviarão uma notificação apropriada de volta para um cliente associado.

- 10 Contudo, é possível que um serviço, uma classe de serviço, um habilitador, um usuário ou grupo modifique / defina um gatilho, o qual realiza ações exclusivamente na rede, sem qualquer envolvimento de cliente.

- Um fluxo de chamada é mostrado abaixo com respeito à
- 15 Figura 14. Os gatilhos de aspecto não requerem uma assinatura associada em nome de um cliente ou serviço. Os dados gatilhos são calculados ou derivados na rede, um observador interessado, seja um dispositivo de cliente ou um serviço de intertrabalho / habilitador, pode receber uma
- 20 notificação não alertada ou assíncrona, como resultado de um gatilho de aspecto. As notificações podem ser manipuladas usando-se meios de comunicação diferentes. Por exemplo, um dispositivo de cliente pode receber uma notificação de SMS como resultado de um disparo de gatilho
- 25 de aspecto. Adicionalmente, outros serviços podem receber uma notificação de SIP / PUSH 1.0 de OMA ou notificações, em resposta a um gatilho associado.

- Os conteúdos de uma notificação são específicos para o gatilho e poderiam incluir itens tais como o endereço de
- 30 registro de uma ou mais presentidades, um indicador de

aspecto ou máscara para um ou mais aspectos de relevância, um URL, uma máscara de roteamento de serviço ou aplicativo para a entidade de recepção, para garantir que o aspecto seja dirigido ou associado ao observador apropriado, dentre
5 outros.

Cada cliente ou serviço recebendo uma notificação pode responder de acordo com o protocolo de transporte associado. Adicionalmente, é possível que indicações de gatilho de aspecto sejam duráveis. Isto é, se um gatilho
10 for calculado para um dado "observador interessado", mas aquele observador for inalcançável, a indicação de aspecto poderá ser persistida ou enfileirada, até que o dado usuário seja capaz de receber apropriadamente o gatilho associado. Isto é útil em cenários em que uma dada
15 notificação pode durar mais do que uma dada sessão de usuário cliente.

Com referência à Figura 14, um dispositivo de cliente 1410 se comunica com um habilitador de serviço 1412, o qual se comunica, ou é integrado com uma camada cliente (AL) 1414
20 (por exemplo, uma camada cliente de contexto (CAL)).

Conforme visto na Figura 14, um gatilho é estabelecido com uma mensagem 1420, em cujo ponto a AL 1414 regula um gatilho, conforme mostrado em 1422, e avalia o gatilho, conforme mostrado pela seta em 1424.

25 A seta 1422 estabelece o gatilho. Isto pode incluir a supressão ou a extensão de etapas padronizadas para o gatilho, a obtenção / avaliação de dados a partir de várias fontes e, possivelmente, o envio de notificações para um ou mais usuários.

30 A avaliação mostrada pela seta 1424 mostra que, quando

um gatilho dispara um endereço de registro, um aspecto ou uma informação de aplicativo é acondicionado e uma notificação é enviada para o dispositivo de cliente ou serviço. Esta notificação é mostrada com a seta 1430.

5 Em alguns casos, uma resposta ou um reconhecimento pode ser retornado, e isto é mostrado pela seta 1432.

Conforme mostrado na Figura 14, a AL 1414 poderia então continuar a monitorar e avaliar se o gatilho deve disparar, conforme mostrado pela seta 1450.

10 As políticas acima, os aspectos e as regras / limites poderiam utilizar uma linguagem de execução de processo comercial de serviços da web na forma de WSBPEL 2.0. WSBPEL 2.0 provê um mecanismo com o qual se expressam seqüências lógicas requeridas para a implementação de regras de
15 presença ou gatilhos (no todo ou em parte) em uma solução de P/CAM. Uma linguagem formal (como PEEM/PEL) para especificação de fluxos lógicos e invocação de primitivos (através de ligações de tipo de linguagem de descrição de
serviço da web (WSDL)) provê um contexto de presença com
20 combinações ilimitadas de regras e/ou gatilhos em nome de um aplicativo ou serviço. Também deve ser notado que fluxos de contexto mais complexos podem ser criados e encadeados em conjunto (por exemplo, através de enlaces de parceiros) para a realização de fluxos de trabalho e/ou uma lógica
25 comercial que esteja relacionada à presença e seja relevante de forma contextual para a plataforma conectada. As regras são capazes de invocar outras regras, como regras embutidas. De modo similar, os gatilhos também podem invocar regras, onde aplicável. Em outras modalidades, a
30 expressão de regras poderia ser realizada utilizando-se uma

linguagem de programação tradicional (por exemplo, Java) ou ferramentas de diagramação (por exemplo, um diagrama Sequence, Flow-Chart, ou Use-Case UML sendo traduzido para uma regra(s)).

5 Conforme será apreciado por aqueles versados na técnica, o uso de uma camada ciente de contexto economiza recursos de dispositivo e rede pela redução da quantidade de informação fluindo entre um dispositivo móvel e uma rede, e pela remoção do processamento do dispositivo móvel.

10 Para comparação dos presentes sistema e método, um exemplo de fluxo de informação é mostrado aqui adiante com respeito à Figura 1. Especificamente, quando Alice deseja enviar um alerta de PoC para Bob, a busca XDM a seguir poderia ser feita:

15 GET/pidf-manipulation/users/sip:bob@example.com/index/~/tuple/service-description/service-id=%22org.openmobilealliance:PoC-alert%22 HTTP/1.1

Em resposta, um 'documento de presença bruto' conforme ilustrado abaixo é retornado:

HTTP/1.1 200/OK

20 Etag: "eti87"

Content-Type: application/pidf+xml

...

<?xml version="1.0" encoding="UTF-8"?>

<presence xmlns="urn:ietf:params:xml:ns:pidf"

25 xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"

xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"

xmlns:caps="urn:ietf:params:xml:ns:pidf:caps"

xmlns:op="urn:oma:xml:prs:pidf:oma-pres"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

30 entity="sip:bob@example.com">

```

<!-- documento retornado para agente, a partir da presentidade Bob... -->

<tuple id="a1232">

    <!-- Disponibilidade básica do usuário 'Bob' (disponível)... -->

    <status>

        <basic>open</basic>

    </status>

    <!-- Desejo do usuário 'Bob' (desejando)... -->

    <op:willingness>

        <op:basic>open</op:basic>

    </op:willingness>

    <!-- Estado de registro de usuário 'Bob'... -->

    <op:registration-state>active</op:registration-state>

    ...

    <!-- Descrição de serviço de usuário 'Bob'... -->

    </op:service-description>

        <op:service-id>org.openmobilealliance:PoC-alert</op:service-id>

        <op:version>1.0</op:version>

        <op:description>PoC Alert Service v1.0</op:description>

    </op:service-description>

    <!-- Meio de contato de usuário 'Bob'... -->

    <contact priority="0.90">sip:bob@example.com</contact>

    <!-- deviceID de usuário 'Bob'... -->

    <pdm:deviceID>urn:uuid:d27459b7-8213-4395-aa77-ed859</pdm:deviceID>

    <timestamp>2007-02-22T20:07:07Z</timestamp>

</tuple>

<!-- Tupla de serviço adicional para PoC-Alert... -->

<tuple id="a1233">

    <status>

        <basic>open</basic>

    </status>

```

```

5      <op:willingness>
          <op:basic>open</op:basic>
      </op:willingness>
      <op:registration-state>active</op:registration-state>
      <caps:servcaps>
          <caps:audio>true</caps:audio>
          <caps:text>true</caps:audio>
          <caps:video>false</caps:video>
      </caps:servcaps>
10     ...
      </op:service-description>
          <op:service-id>org.openmobilealliance:PoC-alert</op:service-id>
          <op:version>1.0</op:version>
          <op:description>PoC Alert Service v1.0</op:description>
15  </op:service-description>
      <contact priority="0.90">sip:bob@example.com</contact>
      <pdm:deviceID>urn:uuid:d27459b7-8213-4395-aa77-ed859</pdm:deviceID>
      <timestamp>2007-02-22T22:07:27Z</timestamp>
      </tuple>
20  <!-- Definição de pessoa para Bob (conforme autorizado para a classe
      'forFriends')... -->
      <pdm:person id="a1234">
          <!-- Atividades (reunião)... -->
          <rpид:activities>
25      <rpид:meeting/>
          </rpид:activities>
          <rpид:class>forFriends</rpид:class>
          <!--Coloca tupla de serviço adicional para PoC-Alert... -->
          <rpид:place-type> <lt:office/> </rpид:place-type>
30  <pdm:timestamp>2007-02-22T22:07:07Z</pdm:timestamp>

```

```

</pdm:person>

<!--Dispositivo associado a PoC-Alert... -->

<pdm:device id="a1235">

    <op:network-availability>

        <op:network id="IMS">

            <op:active>

                </op:network>

            </op:network-availability>

            <pdm:deviceID>urn:uuid:d27459b7-8213-4395-aa77-ed859</pdm:deviceID>

10    <pdm:timestamp>2007-02-22T22:07:07Z</pdm:timestamp>

        </pdm:device>

    </presence>

```

O dito acima ilustra, portanto, o grande documento de presença (em termos do número de bytes ou caracteres) que é retornado por sistemas e métodos convencionais, requerendo recursos de bateria significativos para o recebimento e recursos de rede para a transmissão.

Conforme será apreciado por aqueles versados na técnica, o 'documento de presença bruto' resultante ilustrado acima também poderia ser entregue por uma requisição SIP: NOTIFY de OMA/Presence (em nome de um observador autorizado). Uma busca de XDM é usada para simplificação dos fluxos de rede para este exemplo.

Uma referência é feita, agora, à Figura 9. A Figura 9 mostra um processo de exemplo em um dispositivo móvel, quando uma camada ciente de contexto (P/CAM) é utilizada. O método da Figura 9 substitui e simplifica aquele da Figura 2.

Na Figura 9, o processo começa no bloco 910, e prossegue para o bloco 912, no qual o P/CAM é invocada para

o estabelecimento de um aspecto de presença 'alcançável' para 'Bob' e o serviço 'alerta de PoC'. O bloco 912 espera que o P/CAM retorne um resultado e, com base no resultado, o processo pode prosseguir para o bloco 920, o qual indica 'Bob inalcançável', e termina no bloco 922. Alternativamente, o processo prossegue a partir do bloco 912 para o bloco 930, o qual indica 'Bob alcançável' e termina no bloco 932.

Conforme será apreciado a partir do dito acima, a alcançabilidade em um cliente de alerta de PoC agora é um bloco de processamento único (alcançável / inalcançável). Deve ser notado que o número de blocos de processamento para um dado aplicativo ciente de contexto é reduzido na proporção à complexidade e ao número de funcionalidades relacionadas à presença associadas àquele aplicativo ou serviço.

Uma comparação das exigências de largura de banda de rede entre o aplicativo de cliente de PoC tradicional destacado na Figura 2 e o aplicativo de cliente ciente de contexto na Figura 9, uma redução de ordem de magnitude no tempo de processamento de rede é provida. Uma comparação da busca de XDM usando uma informação de presença para o cliente de PoC tradicional, com uma invocação de método de SOAP (como um cenário de emprego de exemplo) para o aspecto de presença 'alcançável' ciente de contexto, o que vem a seguir é um exemplo de uma requisição:

```
POST /CAM-1 HTTP/1.1
```

```
...
```

```
HOST: pcam.example.com
```

```
Content-Type: text/xml; charset="utf-8"
```

```

...
<!-- SOAP request... -->

<SOAP-ENV:Envelope

  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

5    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

  <SOAP-ENV:Body>

    <pcam:reachable xmlns:pcam="http://pcam.example.com/wsdl/oma-pres-pcam">

      <aor>sip:bob@example.com</aor>

      <service>org.openmobilealliance:PoC-alert</service>

10    </pcam:reachable>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

O que vem a seguir é um exemplo de uma resposta:

HTTP/1.1 200/OK

Connection: close

...

Content-Type: text/xml; charset="utf-8"

...

<!-- Resposta de SOAP... -->

```

20 <SOAP-ENV:Envelope

  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

  <SOAP-ENV:Body>

    <pcam:reachableResp xmlns:pcam="http://pcam.example.com/wsdl/oma-pres-pcam">

25    <result>reachable</result>

    </pcam:reachable>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

Portanto, o dito acima ilustra a redução nos dados
 30 requerida para serem transferidos e também a redução no

processamento requerido por um cilindro.

Um exemplo adicional é ilustrado abaixo, com referência à Figura 10. A Figura 10 é provida para se mostrar um exemplo em que uma agência de anúncio 'Ad-Inc' deseja alcançar consumidores com uma campanha de propaganda móvel genérica. A Ad-Inc. gostaria de otimizar a entrega de anúncios para os dispositivos de recurso restrito com base em critérios específicos. Por exemplo, a campanha de anúncio consiste em pequenas seqüências de vídeo; portanto, o dispositivo deve ser alcançável, ter capacidades de mídia específicas e ter uma quantidade módica de nível de bateria, de modo que os anúncios possam i) ser apresentados apropriadamente no dispositivo e ii) não façam com que o dispositivo perca significativamente bateria, de modo a aborrecer o consumidor prospectivo e causar uma associação negativa com a(s) marca(s) na campanha. Um habilitador de propaganda móvel "MobAd" especifica um novo aspecto de presença conhecido como 'elegibilidade para anúncio' para o P/CAM através da introdução de um documento de 'processo' de Política (por exemplo, PEEM/PEL) com os fluxos lógicos adequados. De modo similar ou em combinação, o aplicativo MobAd poderia especificar um aspecto de localização.

Na Figura 10, o processo começa no bloco 1010 e prossegue para o bloco 1012, no qual o P/CAM é invocado para estabelecer se o dispositivo é 'elegível para anúncio, o aspecto de presença para o prospecto de presentidade e o serviço 'MobAd'.

O bloco 1012 espera por uma resposta do P/CAM e, dependendo do resultado, prossegue para o bloco 1020, no qual o prospecto é julgado NÃO 'elegível para anúncio'. O

processo prossegue para o bloco 1022 a partir do bloco 1020 e termina.

Inversamente, a partir do bloco 1012, o processo poderia prosseguir para o bloco 1030, se o resultado do P/CAM julgasse o prospecto como sendo 'elegível para anúncio'. Conforme será apreciado, 'elegível para anúncio' poderia ser considerado uma variante específica do aspecto 'eligible-session-participant', conforme definido na Tabela 2 acima. O processo então prossegue para o bloco 1032 e termina.

Os blocos de processamento relacionados ao aspecto de presença de elegibilidade para anúncio de MobAd na Figura 10 é significativamente menor do que seria requerido por um agente ou cliente de MobAd independente processando estes metadados por si. Uma complexidade adicional precisaria ser adicionada por e acima do fluxo lógico mostrado na Figura 2, para se suportar a lógica adicional de resolução de capacidades de política e mídia de limite. Esta lógica é incorporada, ao invés disso, como um novo aspecto de presença (bloco de construção de aspecto de presença) em uma camada ciente de contexto e ligada para a computação da presença contextual em nome de MobAd (elegibilidade para anúncio).

Um exemplo adicional é ilustrado com referência à Figura 11. A Figura 11 ilustra o exemplo em um cenário no qual um Servidor de entrega de conteúdo dinâmica (DCD) deseja enviar um conteúdo de DCD para um aplicativo de cliente habilitado para DCD (DECA) através de um cliente de DCD residente em um dispositivo de usuário. O servidor de DCD é considerado um observador do aplicativo de cliente

habilitado para DCD (uma presentidade). O servidor de DCD gostaria de enviar um conteúdo para o aplicativo de cliente habilitado para DCD associado apenas se o cliente de DCD fosse alcançável e a capacidade de armazenamento do dispositivo associado estivesse acima de um limite de memória mínimo pré-definido após o cliente de DCD ter pressionado o conteúdo. Desta forma, o servidor de DCD busca assegurar que o conteúdo pressionado ou enviado de outra forma não exaure indesejavelmente a capacidade de armazenamento do dispositivo. Para esta finalidade, o DCD estabelece um novo aspecto de presença conhecido como 'content-pushable' (conteúdo pressionável) para o P/CAM pela introdução de um documento de 'processo' de PEEM/PEL com fluxos lógicos adequados. De novo, isto é análogo ao aspecto 'eligible-session-participant', exceto pelo fato de aqui o critério ou aspecto ter sido adaptado, suprimido ou configurado de outra forma para um habilitador de DCD. No presente caso, o cliente é alcançável, e tem uma dada capacidade de armazenamento livre.

Com referência à Figura 11, o processo começa no bloco 1110. O processo então prossegue para o bloco 1112, no qual o P/CAM é invocado para o estabelecimento de um aspecto de presença de 'conteúdo pressionável' para a presentidade 'Prospect' e o serviço 'DCD'. Um resultado do P/CAM determina se o processo prossegue para o bloco 1120, e decide que DECA não é de 'conteúdo pressionável', ou para o bloco 1130, e decide que o DECA é de 'conteúdo pressionável'.

O processo prossegue para o bloco 1122 ou o bloco 1132 a partir dos blocos 1120 e 1130, respectivamente, e

termina.

Os blocos de processamento relacionados ao aspecto de presença de conteúdo pressionável de DCD na Figura 11 são executados pelo P/CAM, de modo que o servidor de DCD simplesmente invoque a regra no P/CAM provendo parâmetros de entrada, tais como `devideID`, `serveid` e `content-size`. Esta regra agora pode ser incorporada como um novo aspecto de presença (bloco de construção de aspecto de presença) no P/CAM e ligada para a computação de uma presença contextual em nome do DCD (`content-pushable`).

O dito acima é ilustrado nos exemplos abaixo.

Cliente de Envio de Mensagem Instantânea

Um aplicativo de cliente de exemplo para o uso de uma camada ciente de contexto é um aplicativo de envio de mensagem instantânea. O aplicativo de envio de mensagem instantânea é denominado "MyFriendlyChat" aqui.

Em um cenário de universidade, por exemplo, vários amigos podem ter o aplicativo "MyFriendlyChat" carregado nos seus dispositivos móveis. Neste exemplo, o usuário Alice é uma estudante da universidade que acabou um dia de aulas. Ela está se encaminhando para o restaurante da faculdade e se pergunta se qualquer um de seus amigos está próximo para se unir a ela no jantar.

Alice saca seu dispositivo sem fio e começa o aplicativo "MyFriendlyChat" e invoca a função "Invite-nearby-friends-to-chat", esta função utiliza a presença e a localização para retornar uma lista de amigos que estão a uma distância predeterminada e têm um status alcançável. O aplicativo "MyFriendlyChat" retorna a lista de companheiros ativos mostrando que Bob e Jane estão próximos e

alcançáveis.

Alice introduz uma mensagem curta em seu dispositivo, deixando os amigos saberem que ela está indo para o restaurante da faculdade. Bob e Jane recebem a mensagem e
5 Alice e respondem que eles se unirão a ela brevemente.

O dito acima mostra um aplicativo de cliente o qual utiliza a presença e a localização, de modo a fazer determinações e retornar uma informação relevante para um usuário. Em particular, a função "invite-nearby-friends-to-
10 chat" requer um conhecimento da localização de amigos próximos, bem como uma informação de presença para se permitir que o envio de mensagem instantânea ocorra.

Sob um modelo tradicional de envio de mensagem instantânea, uma plataforma de presença precisaria ser consultada para a obtenção de uma lista de dados brutos, os
15 quais então deveriam ser processados pelo aplicativo de cliente. Ainda, neste caso, uma plataforma de localização também seria requerida para ser consultada para se encontrar a localização de indivíduos em uma lista de
20 companheiros.

De acordo com a presente exposição, os aspectos podem ser abstraídos para uma camada ciente de contexto, que está localizada na rede. A camada ciente de contexto pode ser parte de uma plataforma, tal como a porção longitudinal de
25 localização e de presença, parte de um servidor dedicado, parte de um servidor de presença ou de localização, ou poderia ser distribuída dentre estas entidades. Em alguns casos, um agente para a camada ciente de contexto também poderia existir no dispositivo sem fio ou em um outro
30 computador.

A funcionalidade do aplicativo de cliente é colocada na camada cliente de contexto, assim se provendo resultados consistentes entre aplicativos de cliente variados e também se reduzindo uma sinalização requerida entre o dispositivo móvel e a rede.

Para o dito acima, o aplicativo de cliente "MyFriendlyChat" funciona como um observador e uma fonte de presença em uma realização de OMA/PRS e funciona como uma fonte de presença em uma realização de camada cliente de contexto.

A camada cliente de contexto faz uso de um aspecto pré-definido para determinar se Bob e Jane podem ser alcançados. Neste caso, o aspecto pode ser "eligible-session-participant", o qual é definido para a seleção de uma ou mais presentidades com base em um dado critério. Neste caso, o aspecto "eligible-session-participant" é suprimido para o aplicativo "MyFriendlyChat" selecionar a partir de uma lista de grupo aqueles "companheiros" que estejam "desejando, sejam alcançáveis e próximos". O aspecto de presença suprimido é configurado antes da indicação de quaisquer aspectos a partir de um cliente de "MyFriendlyChat" em execução no dispositivo sem fio.

Com respeito a fluxos de chamada, o aplicativo de cliente deve terminar quem está desejando, é alcançável e está próximo para iniciar um datagrama de mensagem para convidar estes "companheiros" para o jantar. Para se cumprir esta funcionalidade, é assumido que o aplicativo "MyFriendlyChat" tem assinaturas de membros da lista de companheiros de Alice através de componentes de OMA PRS/RLS.

O aplicativo de cliente precisa apenas, após isso, iniciar as comunicações em direção a participantes elegíveis para a sessão com base no resultado de camada cliente de contexto.

5 Várias regras poderiam ser aplicadas ao aspecto para se estreitá-lo mais. Por exemplo, um limite poderia ser imposto em um subconjunto de companheiros quando se determinasse quem está próximo e é alcançável. Assim, a regra poderia ser que apenas companheiros da universidade
10 fossem retornados quando a referência fosse feita.

 Em uma continuação do exemplo acima, uma vez que Alice, Bob e Jane alcancem o restaurante, Alice poderia regular um gatilho de aspecto em seu dispositivo móvel para alertá-la se qualquer um dos seus amigos estiver a uma
15 certa distância do restaurante em um período de tempo predeterminado. Por exemplo, Alice poderia regular um gatilho no seu dispositivo para indicar que, se qualquer um de seus "companheiros" estiver a meio quilômetro na próxima meia hora, ela seja alertada.

20 Neste exemplo, Jim atende a estes critérios e Alice recebe uma notificação no seu dispositivo móvel que Jim entrou na área especificada e Alice assim pode convidar Jim a se juntar ao grupo.

 Conforme será apreciado, o dito acima ilustra um
25 exemplo de um gatilho de aspecto. Especificamente, um gatilho é estabelecido para o aspecto "eligible-session-participant" e pode ser denominado, por exemplo, "isEligibleSessionParticipant", o que faria com que um alerta fosse enviado para Alice, uma vez verdadeiro.
30 Conforme será apreciado, esse alerta poderia incluir um tom

audível, uma vibração ou qualquer notificação como essa para indicar a um usuário que as condições de gatilho foram encontradas.

Novamente, o uso de uma camada ciente de contexto facilita o uso de gatilhos, bem como reduz as comunicações entre o dispositivo móvel e a rede, desse modo se economizando vida de bateria e potência de processamento no dispositivo móvel, bem como recursos de rede.

Cenário de Propaganda de Móvel

Em um exemplo adicional, uma companhia de carros XYZ Motors Cars quer que uma campanha de propaganda coincida com o lançamento de um novo modelo de carro para atividade esportiva. A XYZ Motors Cars contrata a Split-second Advertising Company para fazer uma campanha de anúncio e a Split-second faz uso da ABC Telecom como o provedor de serviço sem fio / entrega de conteúdo.

A Split-second estabeleceu uma campanha de propaganda para o novo modelo de carro almejando indivíduos entre 23 e 30 anos de idade com interesses em ciclismo, camping e andar de caiaque. O anúncio contém várias fotos, videoclipes ou similares do novo modelo sendo usado com diferentes atividades esportivas.

Jack, Phyllis, Lynn e George todos concordaram em receber um conteúdo relacionado à propaganda. Andrew está no mercado alvo para a XYZ Motors, mas não optou por receber conteúdo de propaganda. Jack, Lynn e George estão no mercado alvo para a XYZ Motors.

Com o cenário acima, a ABC Advertising Company configura sua plataforma de propaganda sem fio para a campanha de propaganda. Um gatilho é estabelecido na

plataforma de propaganda sem fio, pode o gatilho monitora os indivíduos que se adequam ao critério da Split-second para a dada campanha de anúncio, que tenham optado por receber a propaganda, sejam "alcançáveis" e tenham um
5 dispositivo apropriado com capacidades de receber um videoclipe associado.

A ABC faz a campanha coincidir com a data de lançamento do novo modelo para a XYZ, resultando no gatilho de camada ciente de contexto, definido acima, disparar.

10 Em um tempo posterior, Jack, Lynn e George recebem mensagens contendo uma informação relacionada ao novo veículo sendo introduzido pela XYZ Motors. O conteúdo de anúncio é adaptado apropriadamente para cada dispositivo. Por exemplo, Jack poderia receber uma WAP-Push SMS com o
15 WAP-URL para o sítio inicial da XYZ Motors, enquanto Lynn e George receberiam ambos mensagens de multimídia (MMS) com um videoclipe curto anexado.

Uma vez que Phyllis e Andrew não se adequaram ao critério para a campanha de anúncio, eles não são
20 contatados. Contudo, se em um tempo futuro, mas ainda durante a campanha de anúncio, Andrew optar por receber mensagens de propaganda sem fio, o anúncio da XYZ Motor Company seria enviado para Andrew.

O dito acima é implementado utilizando-se vários
25 aspectos. O aspecto "alcançável" pode ser usado para se determinar se Jack, Lynn e George podem ser alcançados para o envio de mensagens de propaganda para eles. Um aspecto tal como "opt-in" pode ser usado para se determinar se o usuário optou por receber uma propaganda.

30 Os gatilhos também poderiam ser utilizados. Neste

caso, um gatilho tal como "isEligibleSessionParticipant" é usado para se retornar um ou mais usuários que tenham optado por serviços de propaganda sem fio e entrega de conteúdo, sejam alcançáveis e tenham um dispositivo com um conjunto apropriado de capacidades de mídia. Neste caso, a ação padrão para o gatilho de aspecto poderia ser dirigir a camada ciente de contexto para iniciar um conteúdo apropriado para o usuário. Assim, por exemplo, nenhuma indicação direta pelo ar poderia ser enviada para um aplicativo de propaganda no dispositivo de cliente.

A camada ciente de contexto poderia incluir uma informação TA, como "MobileAdvertisingPreferences", definindo uma coleção de preferências específicas de propaganda móvel armazenadas em um XDMS apropriado. O cliente de propaganda sem fio localizado no dispositivo pode invocar esta entidade para retornar as preferências relacionadas à propaganda móvel.

Uma outra informação poderia incluir "ContentDeliveryPreferences" tendo uma coleção de preferências de entrega de conteúdo armazenadas em um XDMS apropriado. O cliente de propaganda sem fio ou outro componente no dispositivo pode invocar esta entidade para retornar as preferências de entrega de conteúdo / serviço / aplicativo / dispositivo.

O exemplo de propaganda provê uma camada ciente de contexto que utiliza dois habilitadores em separado trabalhando em conjunto. Especificamente, um habilitador de propaganda móvel e um de entrega de conteúdo são usados para a obtenção de um ponto de função específico. Essas interações não são possíveis segundo os serviços presentes.

Uma pesquisa mostrou que economias de transferência de dados utilizando uma camada ciente de contexto estão entre em torno de 40% e 75% sob certas condições. Assim, o uso da camada ciente de contexto provê economias de recursos de rede e vida de bateria no dispositivo móvel.

A camada ciente de contexto ainda provê a conexão de múltiplos e variados aplicativos de cliente ao permitir que aspectos, regras, políticas e gatilhos sejam definidos na camada ciente de contexto. Isto provê a vantagem de a camada ciente de contexto poder servir a múltiplos aplicativos de cliente e não precisar ser recriada para cada aplicativo de cliente específico.

As modalidades descritas aqui são exemplos de estruturas, sistemas e métodos tendo elementos correspondentes a elementos das técnicas para este pedido. Esta descrição por escrito pode permitir que aqueles versados na técnica façam e usem as modalidades tendo elementos alternativos específicos que correspondem, da mesma forma, aos elementos das técnicas deste pedido. O escopo pretendido das técnicas deste pedido assim inclui outras estruturas, sistemas ou métodos que não diferem das técnicas deste pedido, conforme descrito aqui, e ainda inclui outras estruturas, sistemas ou métodos com diferenças não substanciais das técnicas deste pedido, conforme descrito aqui.

REIVINDICAÇÕES

1. Método executado por um servidor (350, 440, 450, 540, 550, 650, 740, 750, 840, 850) que está em comunicação com um servidor de presença (330, 430, 530, 630), o servidor de presença sendo adaptado para armazenar informação de presença em forma de dados brutos e atualizações de estado recebidas a partir de presentidades dentro de uma rede sem fios (320), o método caracterizado por compreender:

receber, a partir de um cliente de presença em um dispositivo de usuário (310, 410), uma mensagem (1230) incluindo um atributo service-id, a mensagem solicitando o estabelecimento de um contexto de presença associado ao atributo service-id para fornecer informação de presença abstraída ao cliente de presença de acordo com uma política, um aspecto de presença e uma regra que são constituintes do contexto de presença;

estabelecer (1244), através de comunicação com o servidor de presença, o contexto de presença no servidor para um aplicativo ou um serviço empregado pelo dispositivo de usuário com base na mensagem, o contexto de presença configurado para fornecer a informação de presença abstraída como um valor do aspecto de presença com base em pelo menos um dentre a regra e a política para pelo menos um dentre o serviço, o aplicativo ou uma classe relacionada ao referido aplicativo ou ao referido serviço;

receber, a partir do cliente de presença após o estabelecimento, uma solicitação (1320) do valor do aspecto de presença para uma presentidade em relação atributo service-id; e

enviar, para o cliente de presença em resposta à solicitação, uma resposta (920, 930, 1020, 1030, 1120, 1130, 1250, 1350) transmitindo o valor do aspecto de presença para a presentidade em relação ao atributo
5 service-id,

em que o contexto de presença abstrai informação de presença no valor do aspecto de presença mediante processamento da informação de presença (1332) a partir do servidor de presença (330, 430, 530, 630) de acordo com
10 pelo menos um dentre a regra e a política, desse modo reduzindo uma quantidade de dados a serem transferidos para o cliente de presença e também causando uma redução de recursos de processamento usados pelo cliente de presença.

2. Método, de acordo com a reivindicação 1,
15 caracterizado pelo fato da mensagem (1230) incluir ainda um atributo user-id que identifica um usuário do dispositivo de usuário.

3. Método, de acordo com a reivindicação 1,
caracterizado pelo fato do servidor (350, 440, 450, 540,
20 550, 650, 740, 750, 840, 850) ser integrado com um servidor de aplicativo ou um habilitador da Aliança Móvel Aberta.

4. Método, de acordo com a reivindicação 3,
caracterizado pelo fato do servidor 350, 440, 450, 540,
550, 650, 740, 750, 840, 850) ser pelo menos um dentre um
25 servidor push-to-talk por celular (PoC) e um servidor de mensagem instantânea (IM).

5. Método, de acordo com a reivindicação 1,
caracterizado pelo fato do servidor (350, 440, 450, 540,
550, 650, 740, 750, 840, 850) ser um servidor dedicado que
30 é distinto do servidor de presença (330, 430, 530, 630).

6. Método, de acordo com a reivindicação 2, caracterizado pelo fato do atributo service-id indicar pelo menos um dentre um serviço de envio de mensagem instantânea (IM) e um serviço de push-to-talk por celular (PoC).

5 7. Método, de acordo com a reivindicação 1, caracterizado pelo fato do atributo service-id corresponder à classe de serviço que agrupa o aplicativo ou o referido serviço a outros aplicativos ou a outros serviços, respectivamente, com base em uma característica mútua.

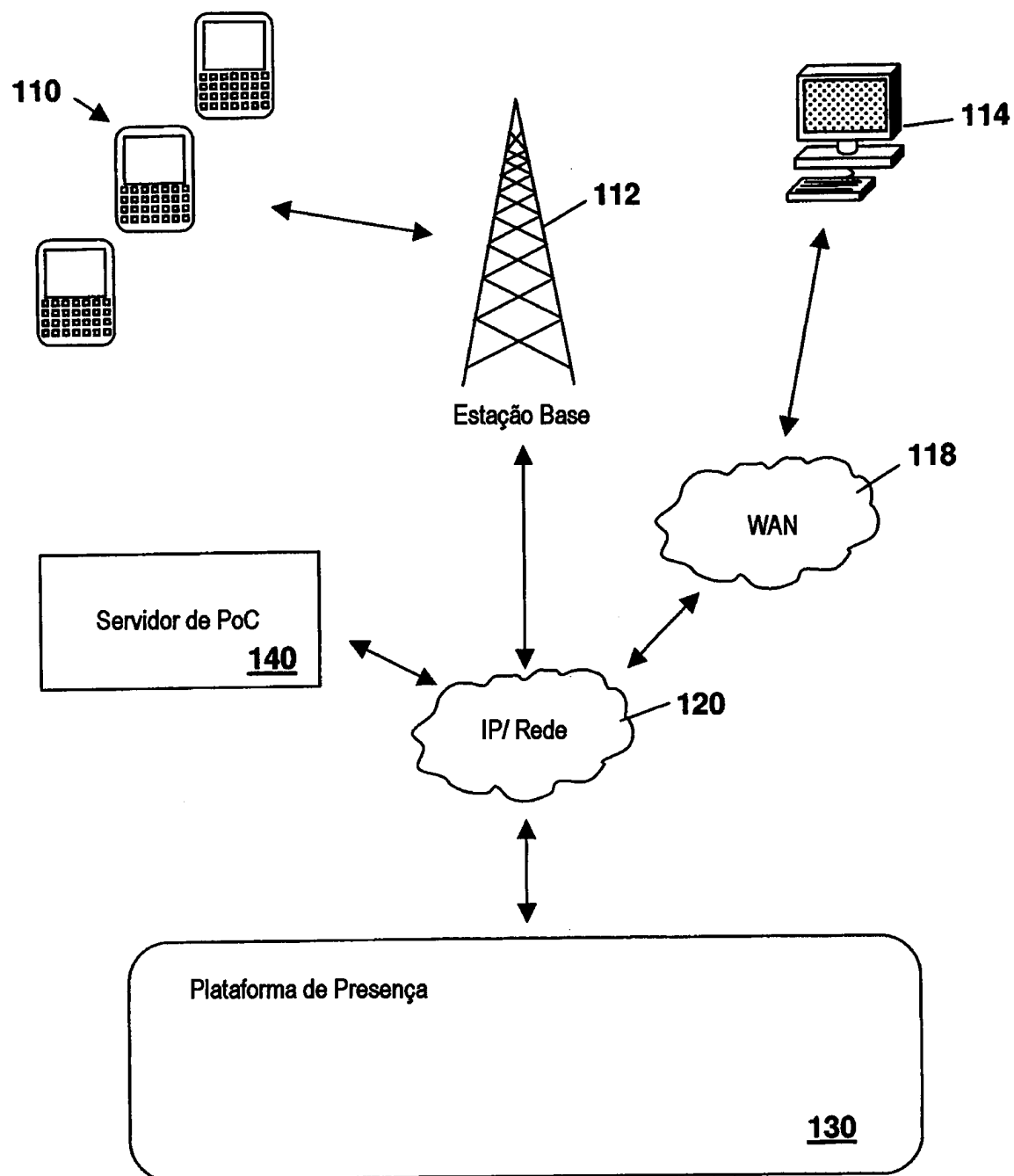
10 8. Método, de acordo com a reivindicação 7, caracterizado pelo fato da característica mútua ser o emprego de aspectos de presença similares ou comuns.

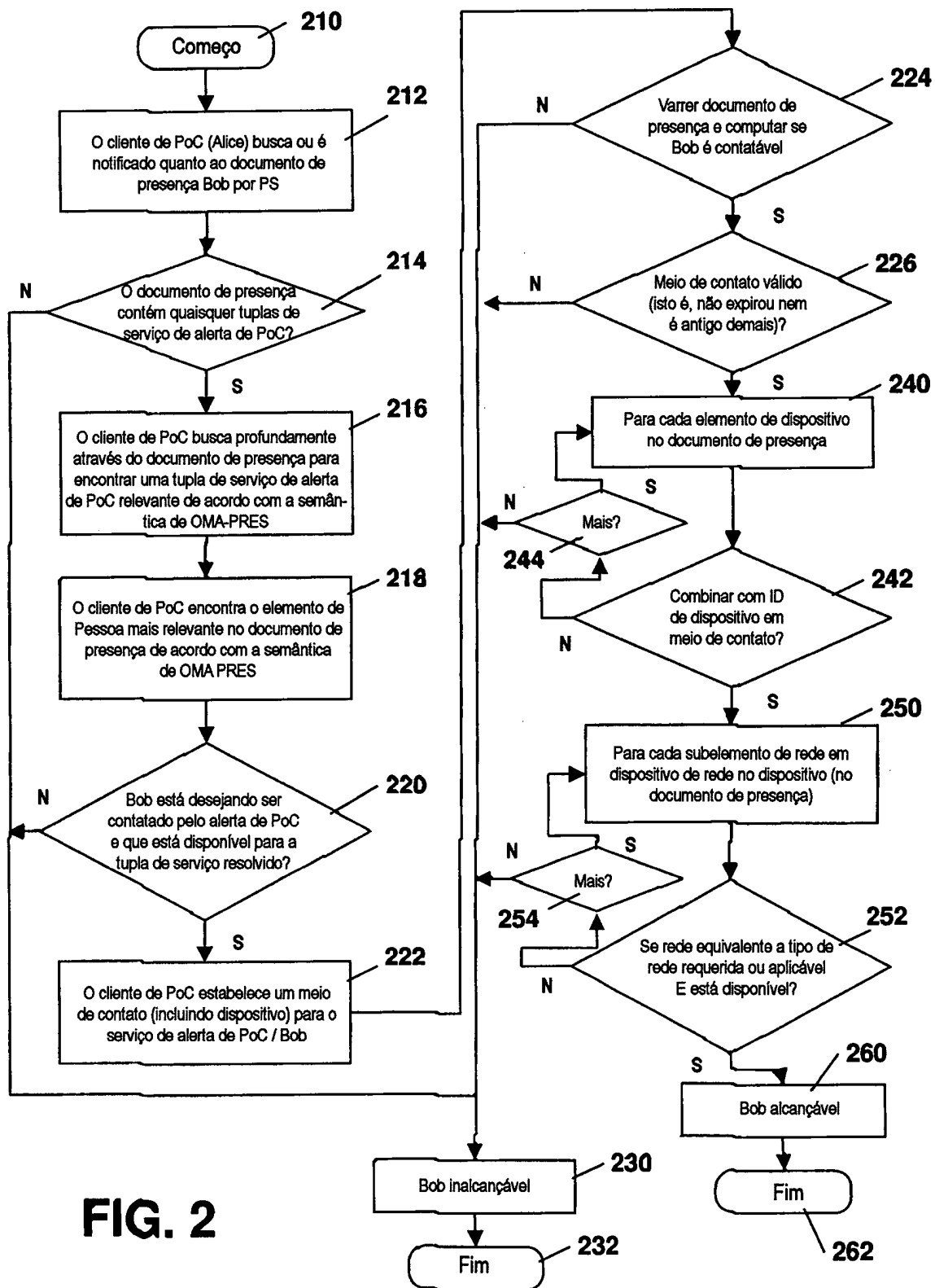
9. Método, de acordo com a reivindicação 7, caracterizado pelo fato da política especificar um
15 tratamento do aspecto de presença pelo servidor.

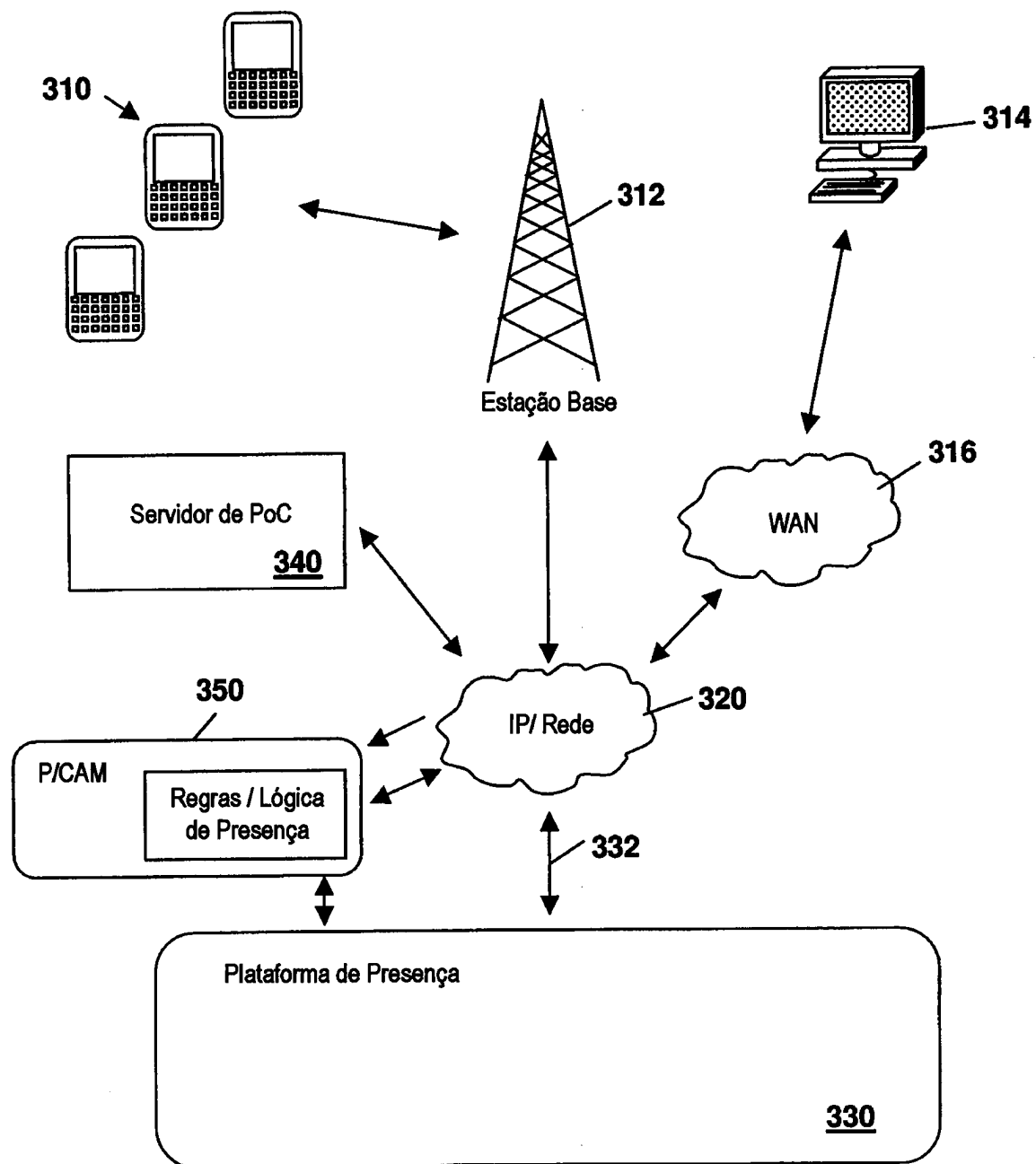
10. Método, de acordo com a reivindicação 1, caracterizado pelo fato do contexto de presença compreender ainda um gatilho que, quando invocado, faz com que uma notificação seja enviada para o cliente de presença em
20 relação ao aspecto de presença.

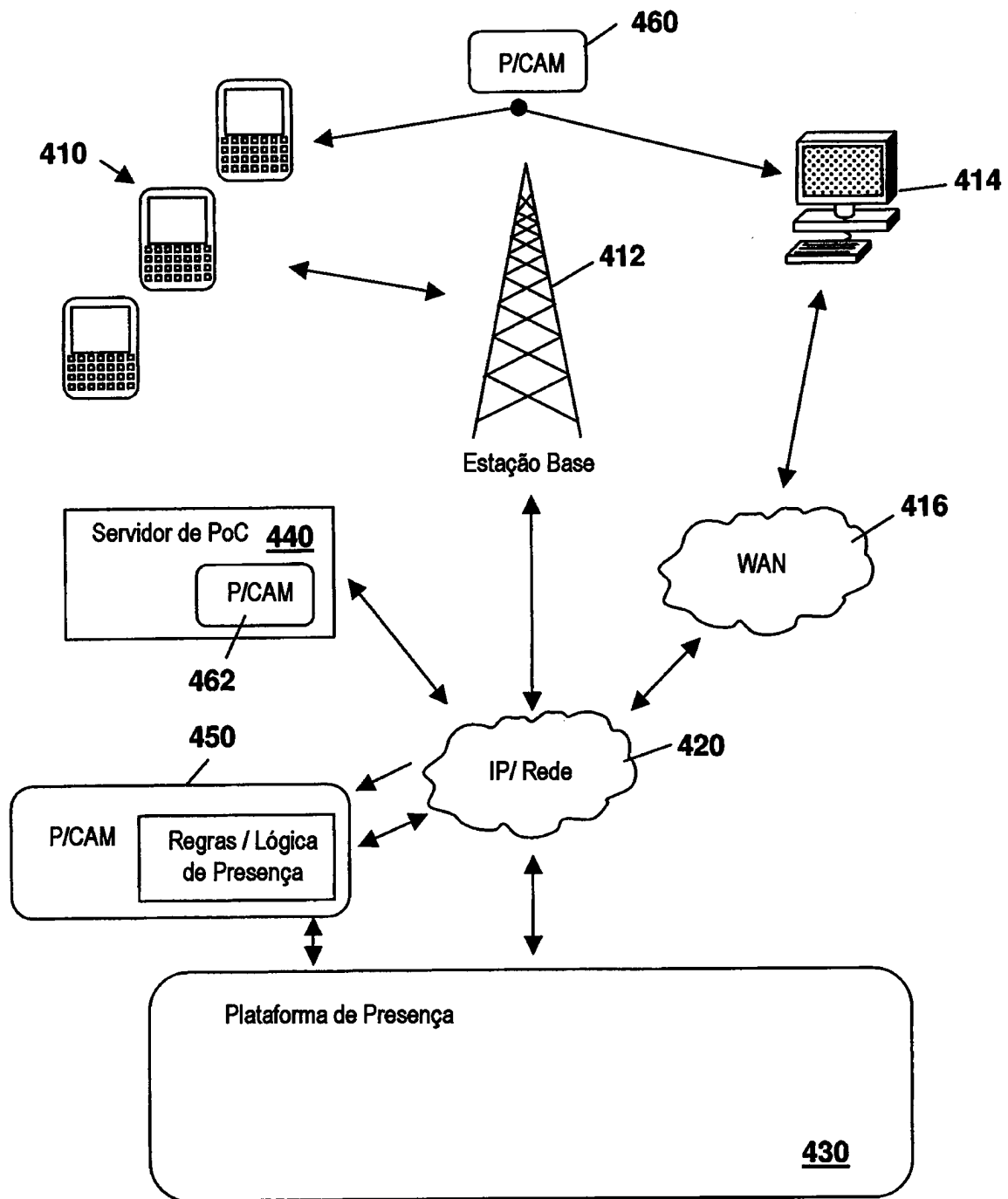
11. Meio que pode ser lido em computador, caracterizado por armazenar nele instruções que fazem com que um processador execute o método como definido em qualquer das reivindicações 1 a 10.

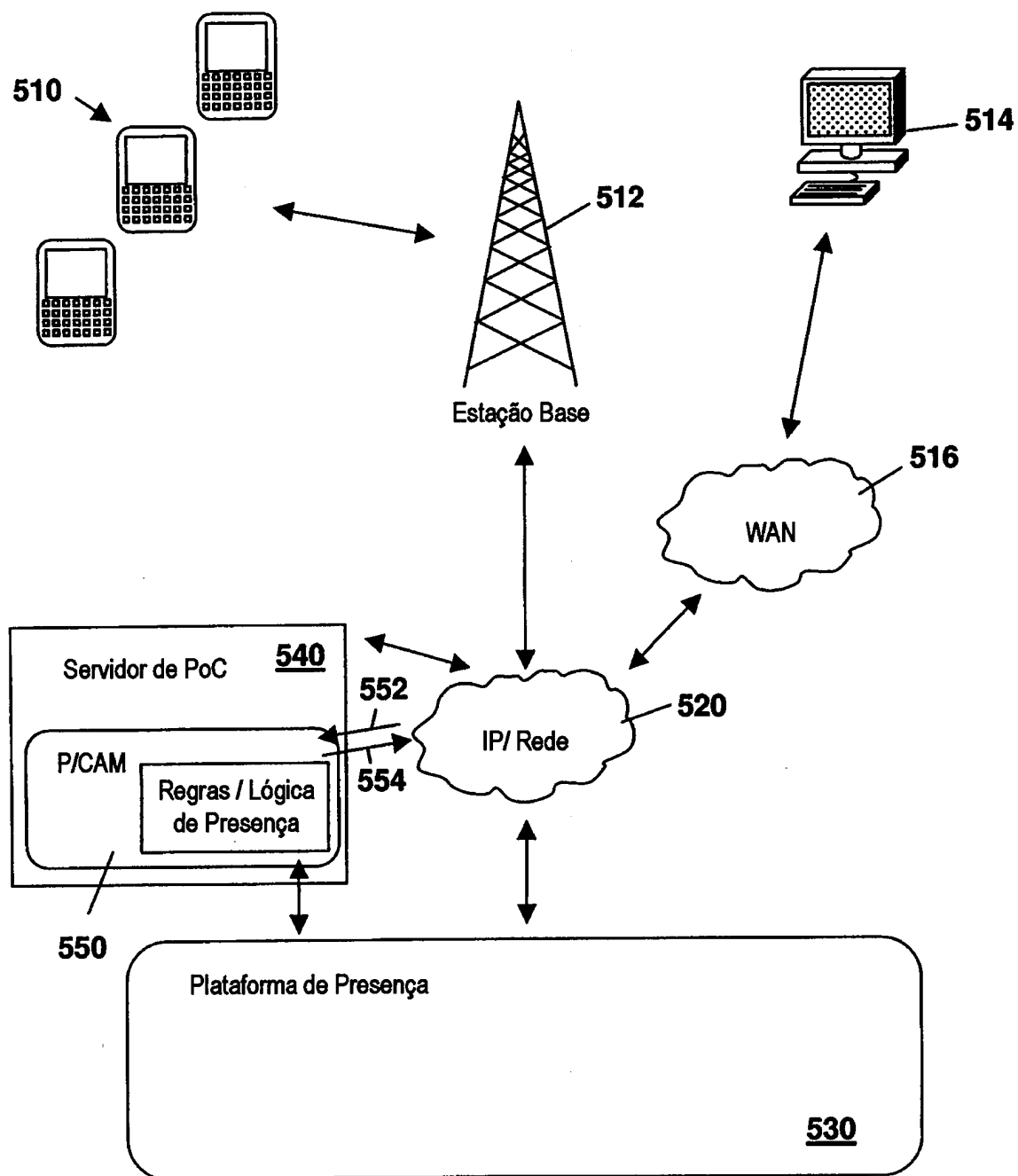
25 12. Dispositivo de rede, caracterizado por compreender um servidor configurado para executar o método como definido em qualquer uma das reivindicações 1 a 10.

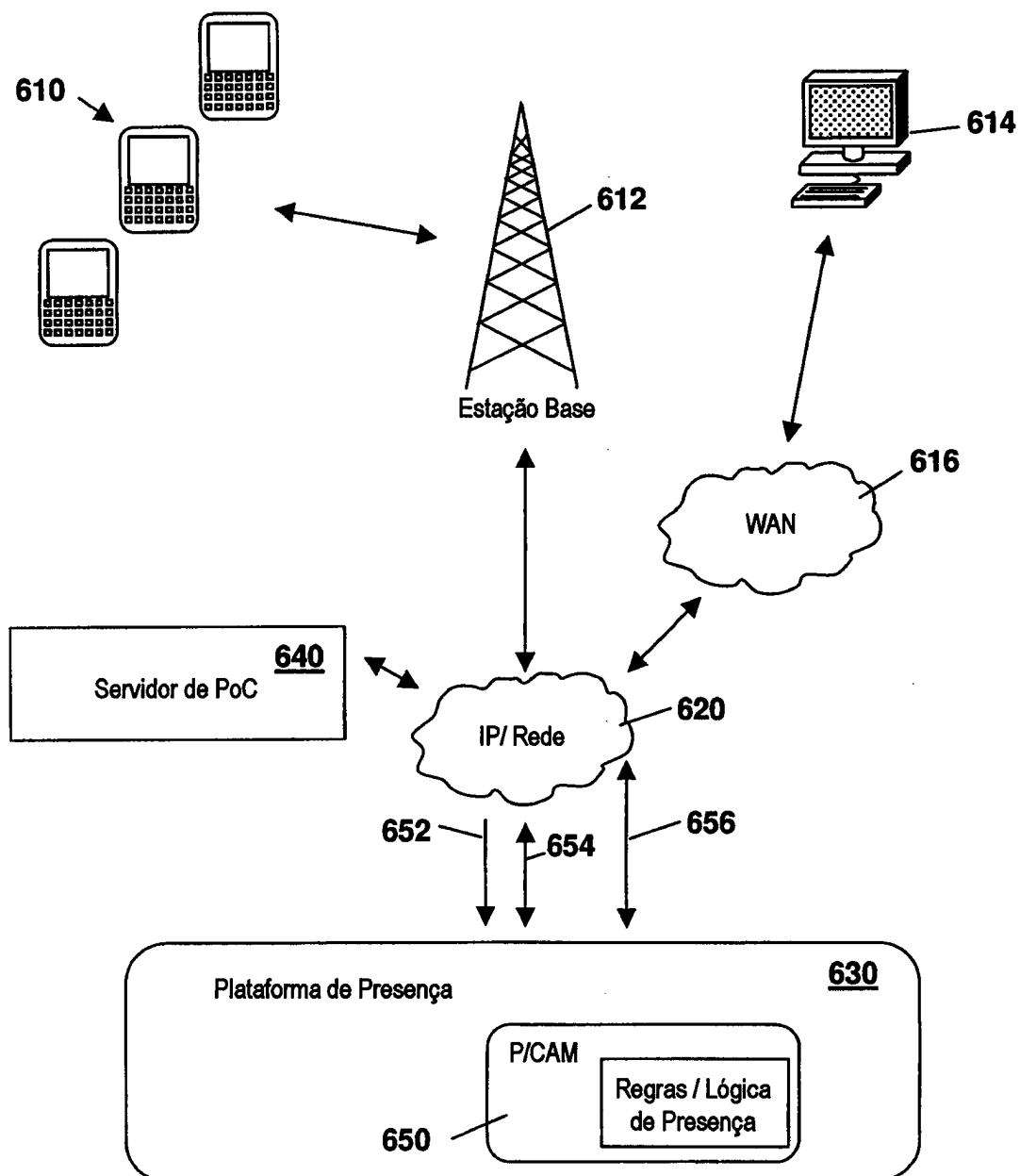
**FIG. 1**

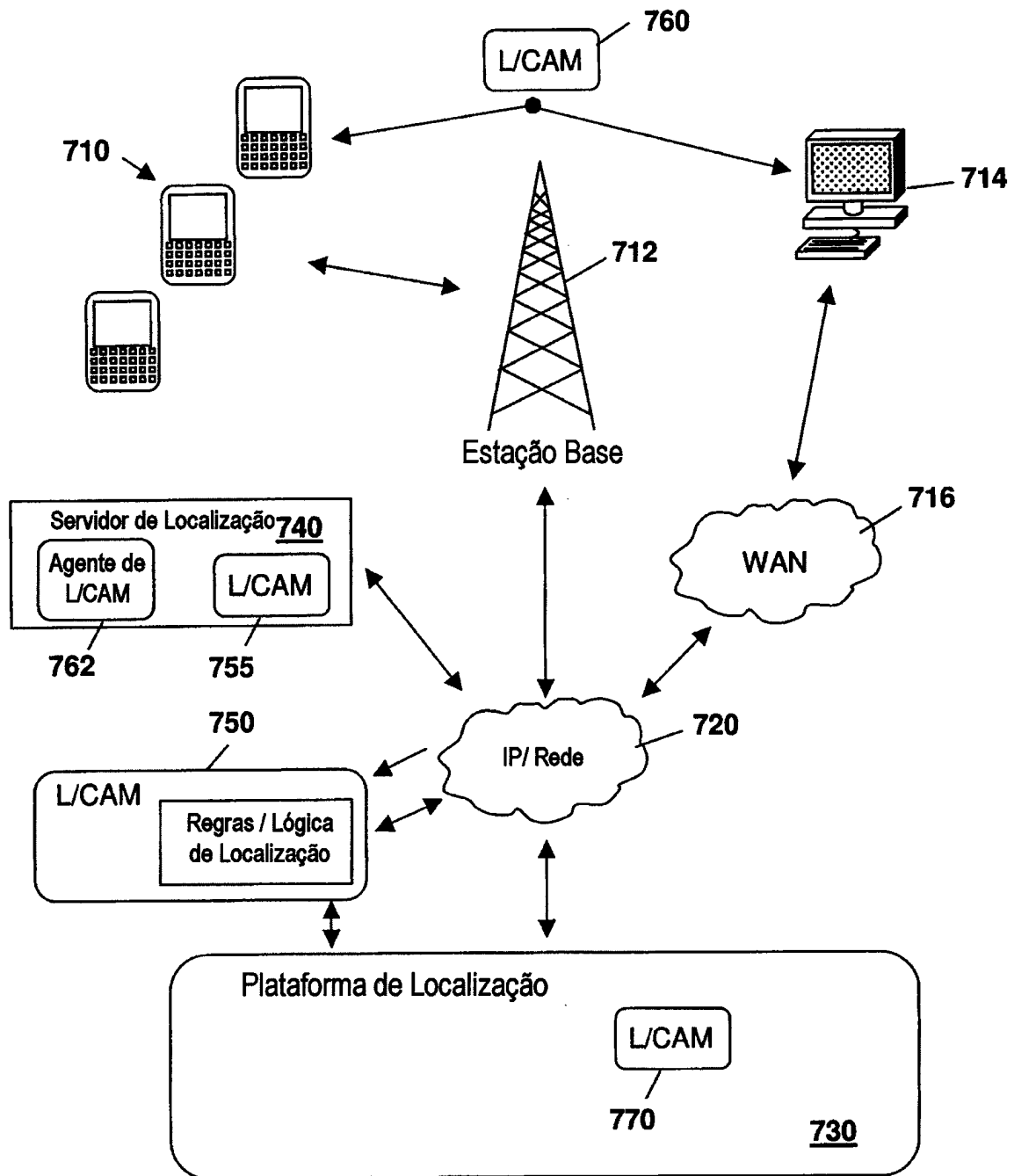


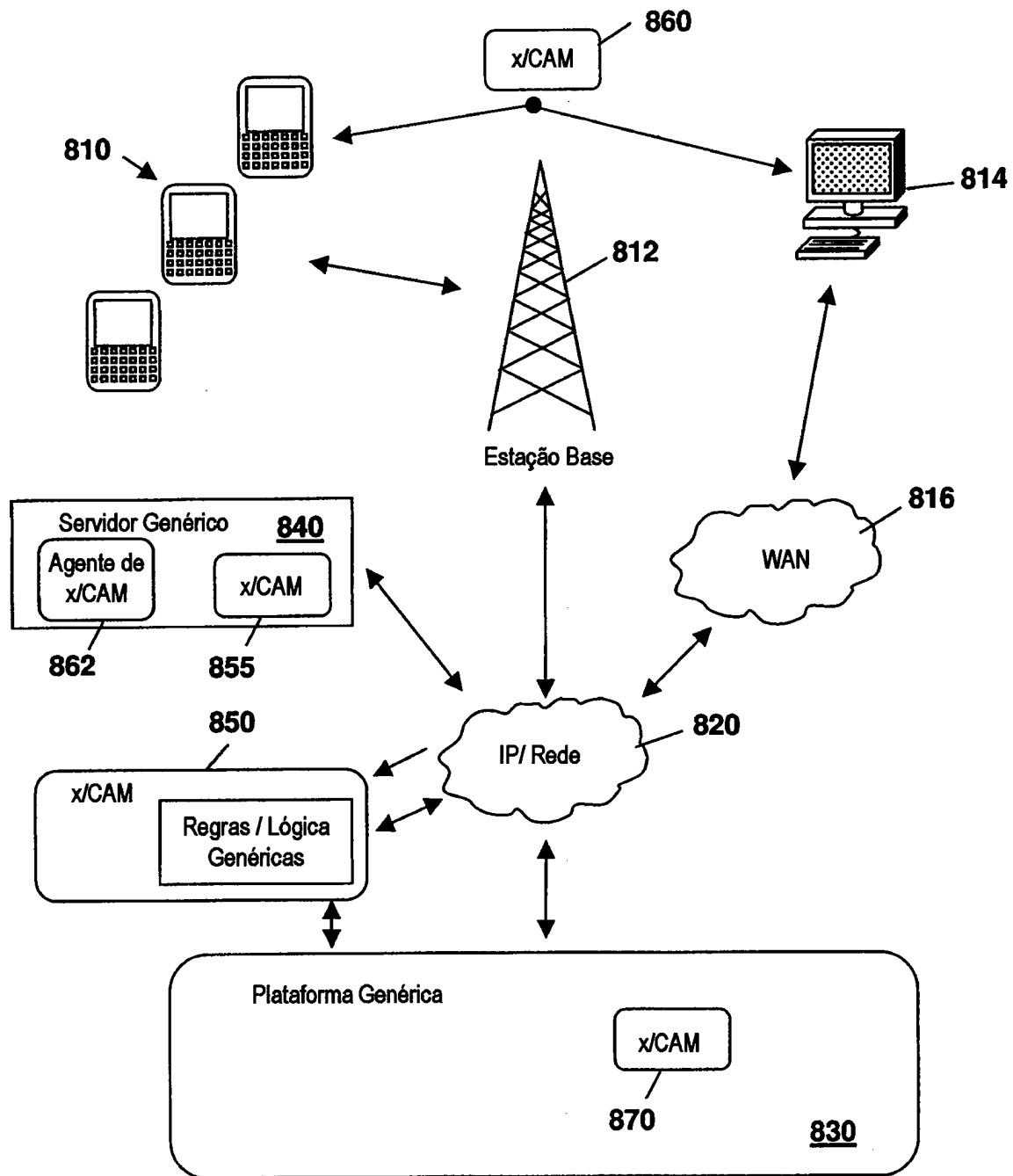
**FIG. 3**

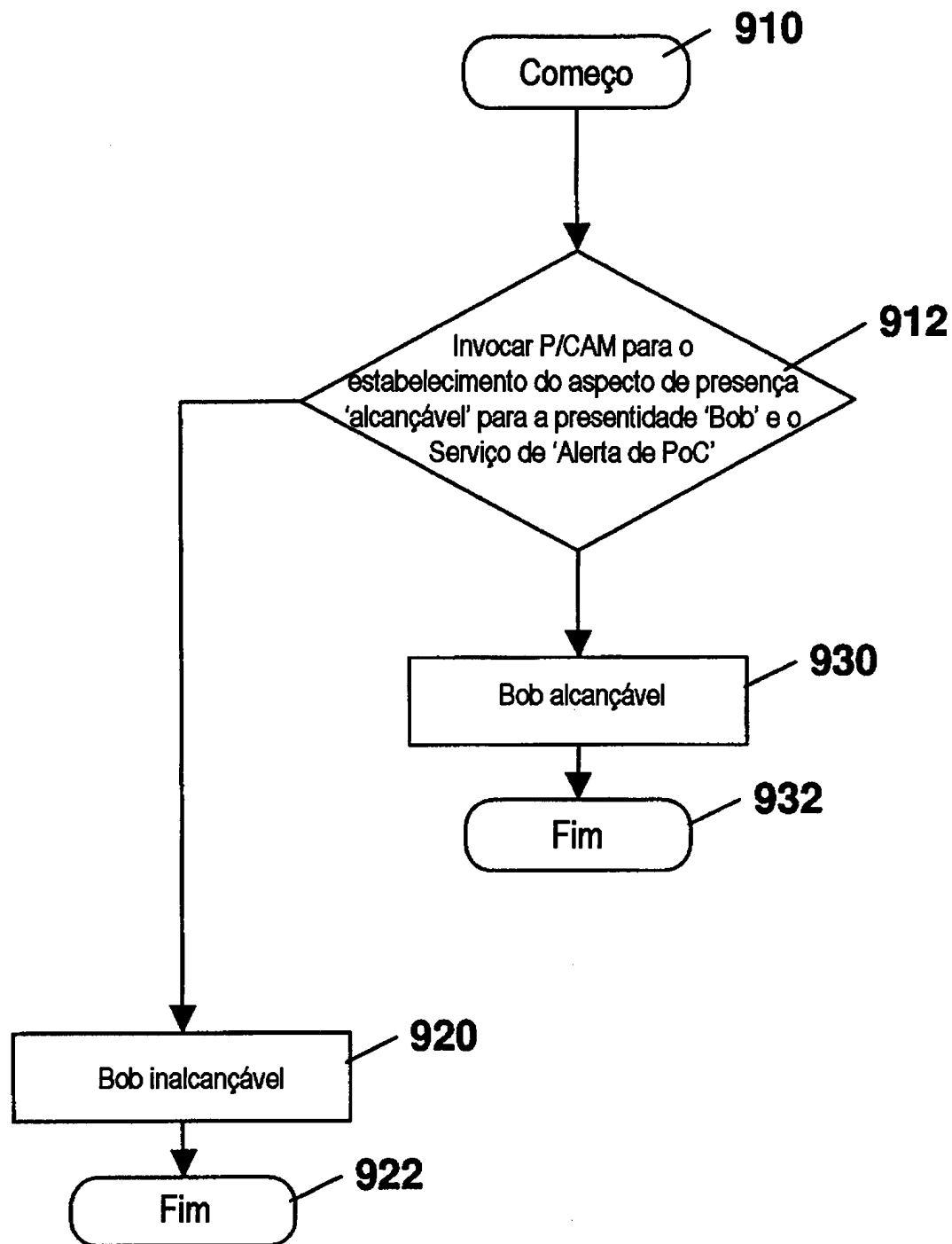
**FIG. 4**

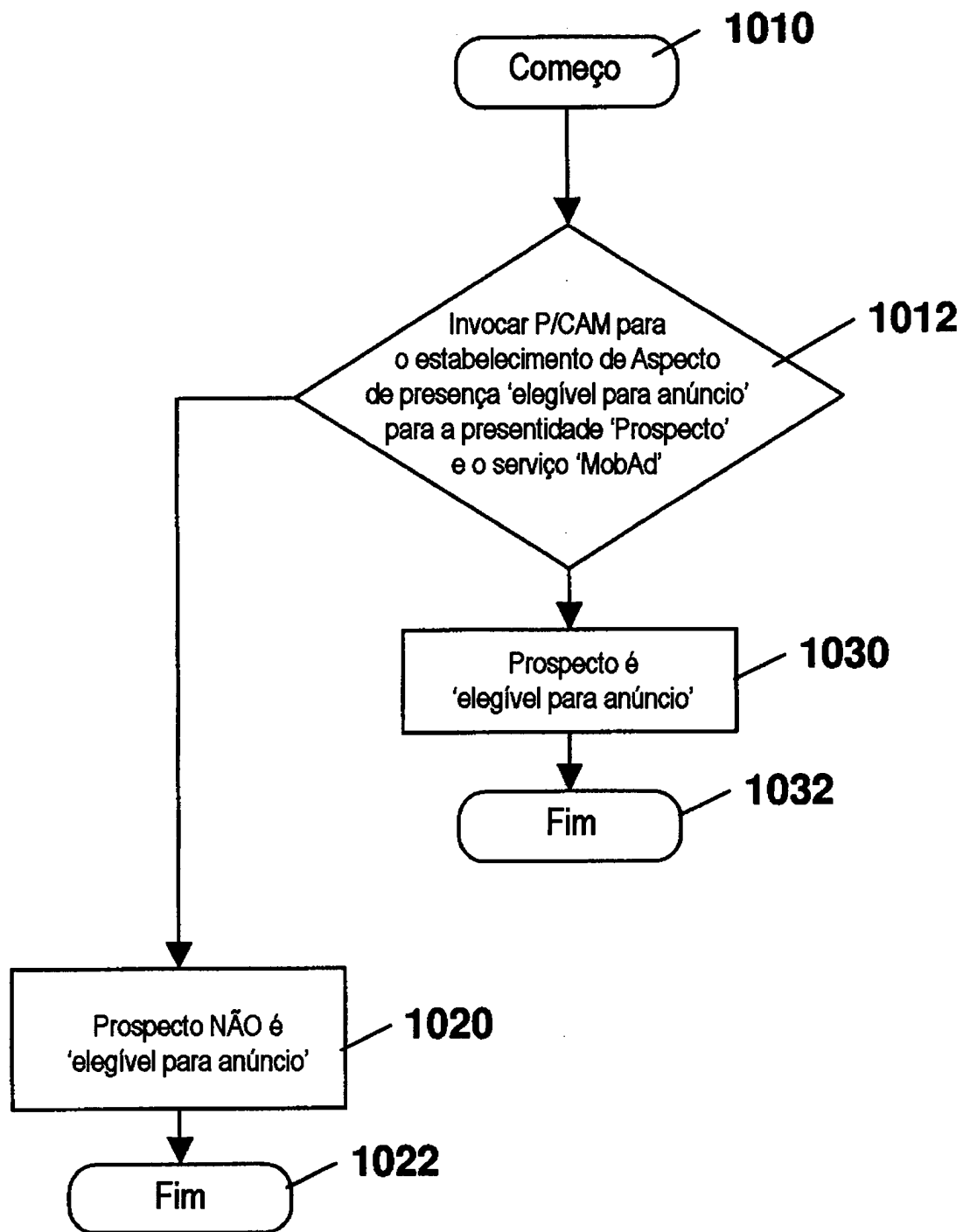
**FIG. 5**

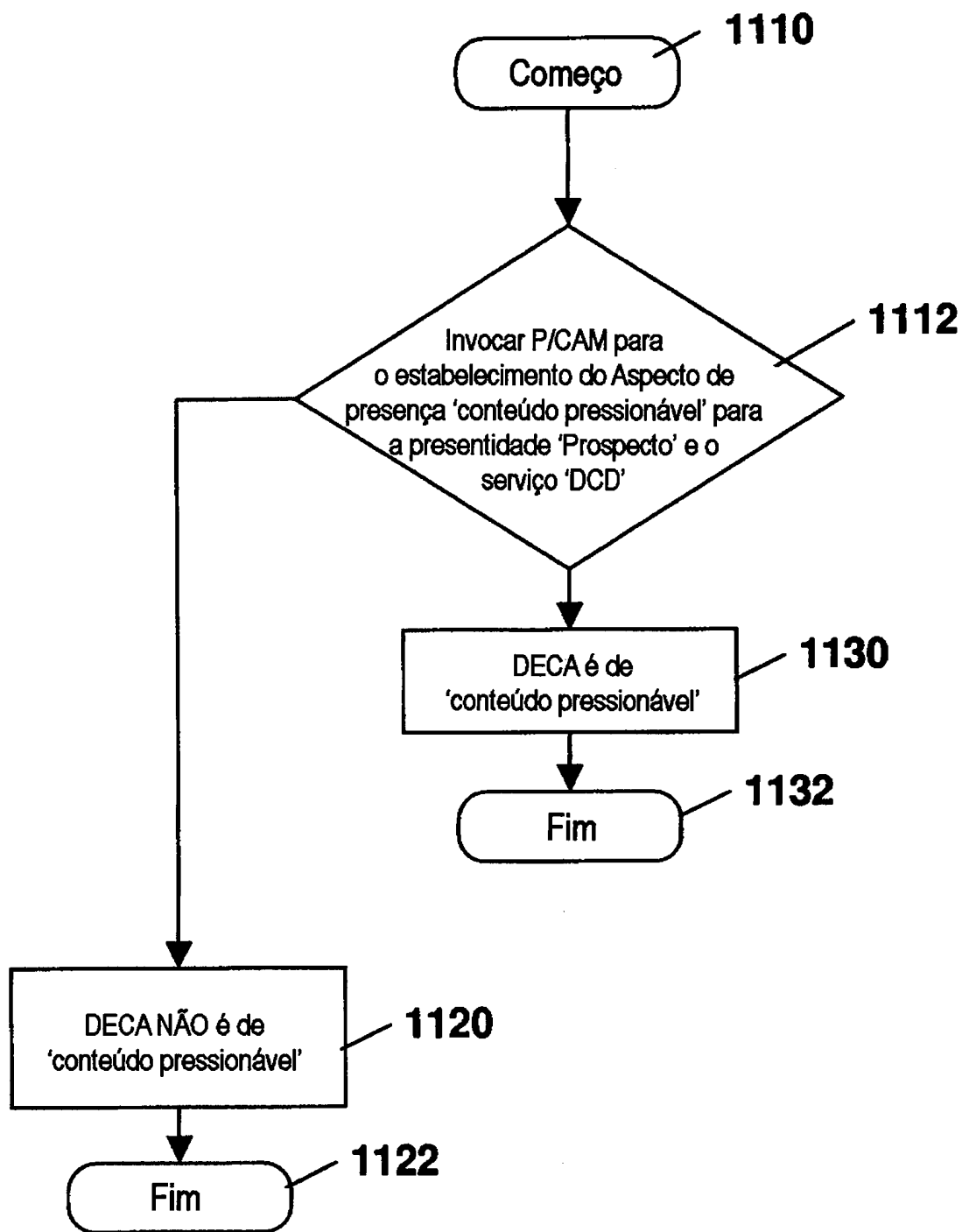
**FIG. 6**

**FIG. 7**

**FIG. 8**

**FIG. 9**

**FIG. 10**

**FIG. 11**

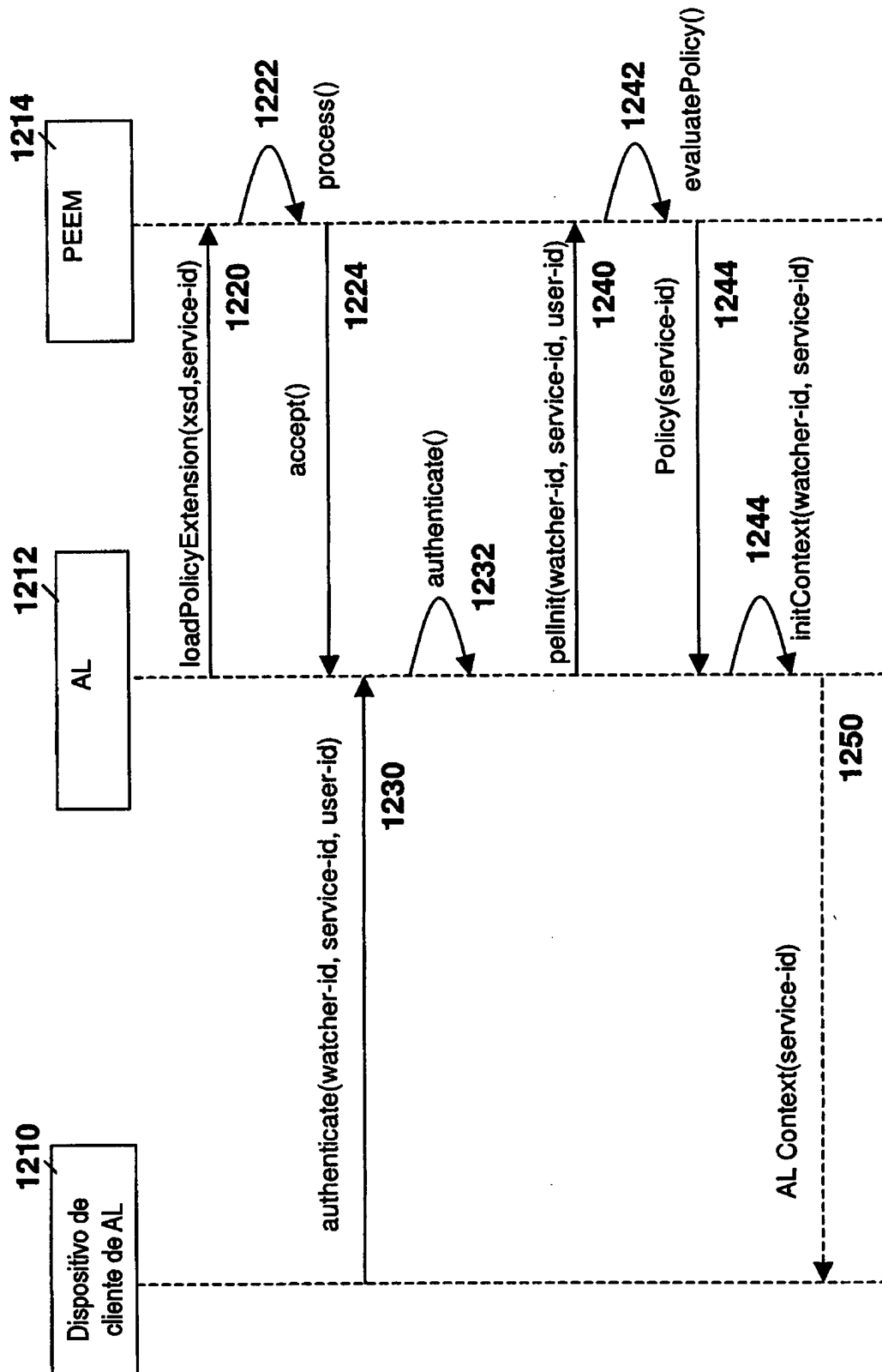
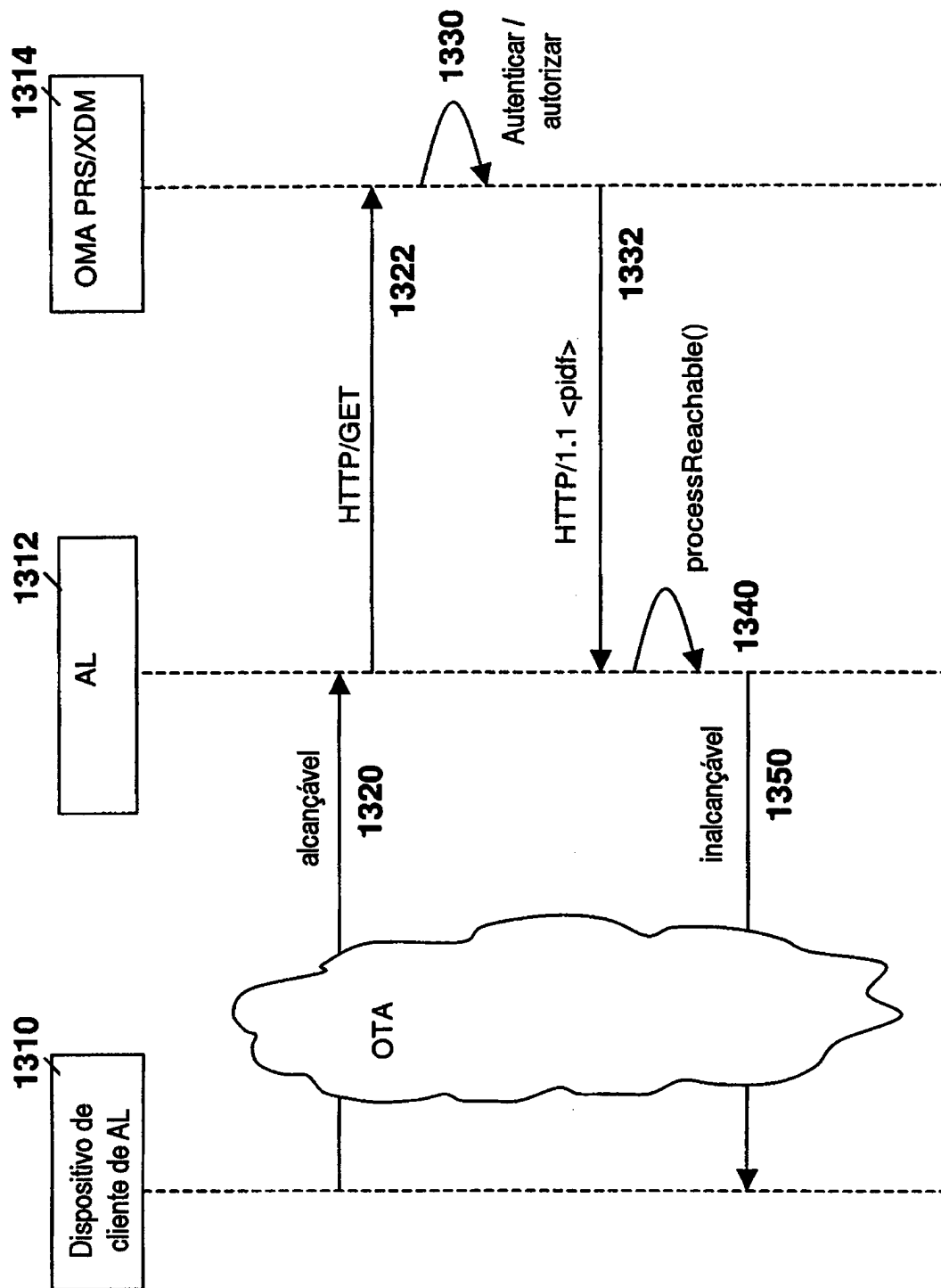


FIG. 12

**FIG. 13**

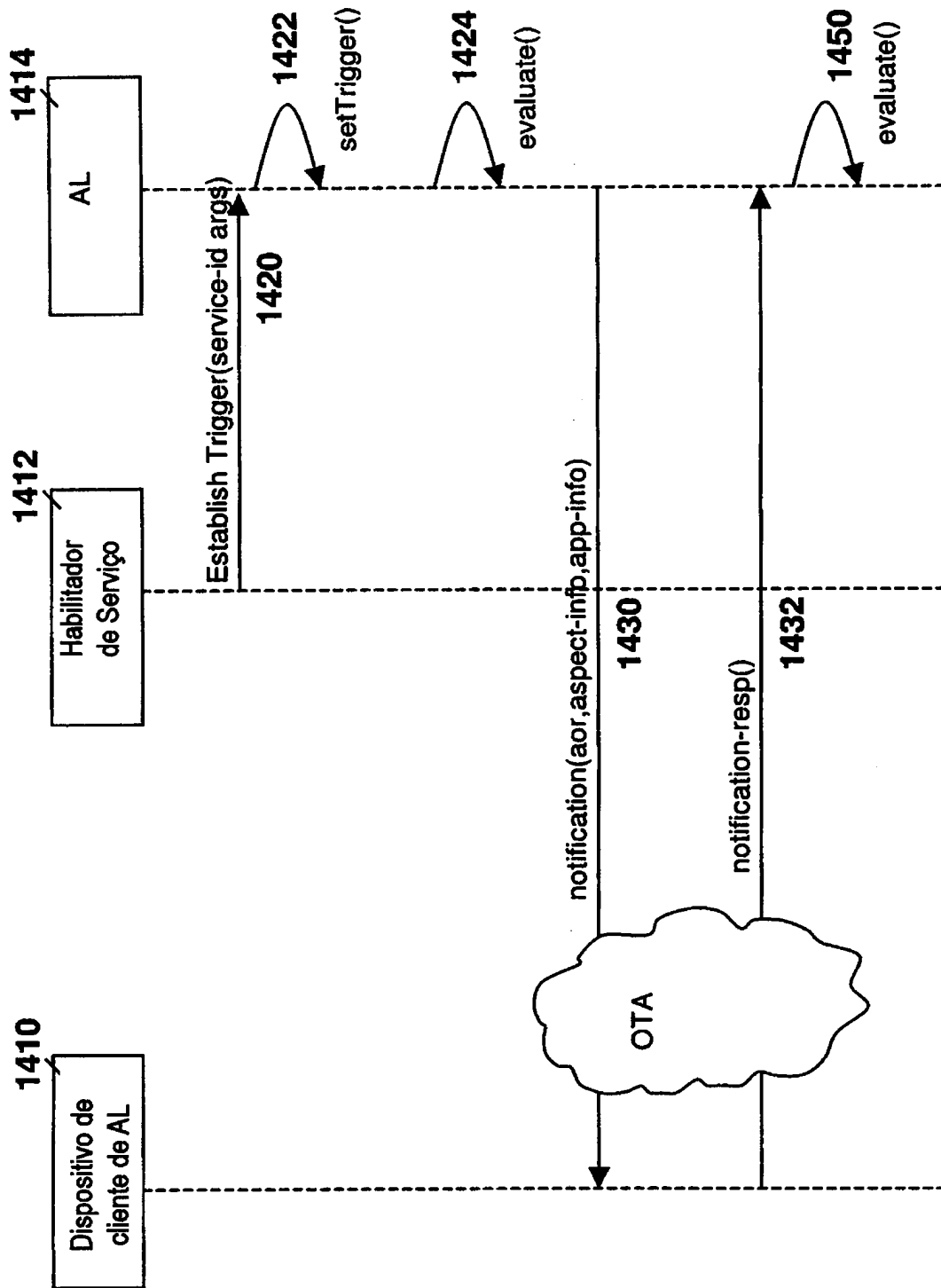


FIG. 14