



República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e Comércio Exterior
Instituto Nacional de Propriedade Industrial

(21) **PI0708228-2 A2**

(22) Data de Depósito: 08/03/2007
(43) Data da Publicação: 17/05/2011
(RPI 2106)



(51) *Int.Cl.:*
G01C 21/00
G06F 9/45

(54) Título: **DISPOSITIVO DE PROCESSAMENTO PARA DETECTAR UM CERTO COMANDO DE COMPUTADOR**

(30) Prioridade Unionista: 08/03/2006 GB 0604704.7, 08/03/2006 GB 0604706.2, 08/03/2006 GB 0604708.8, 08/03/2006 GB 0604709.6, 08/03/2006 GB 0604710.4

(73) Titular(es): Tomtom International B.V.

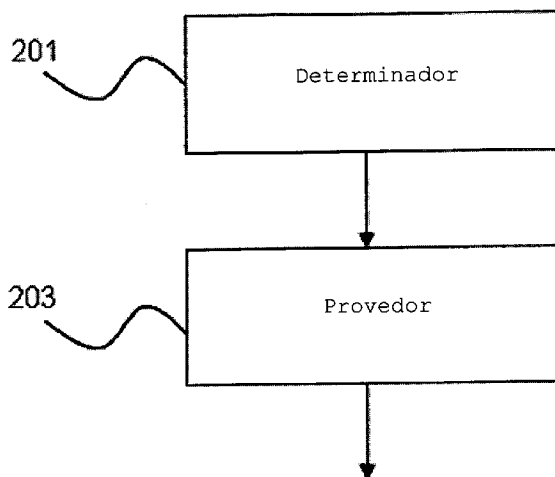
(72) Inventor(es): David Vismans , James Tebutt , Michiel Salters

(74) Procurador(es): Alexandre Ferreira

(86) Pedido Internacional: PCT EP2007002189 de 08/03/2007

(87) Publicação Internacional: WO 2007/101726 de 13/09/2007

(57) Resumo: DISPOSITIVO DE PROCESSAMENTO PARA DETECTAR UM CERTO COMANDO DE COMPUTADOR A invenção provê um dispositivo de processamento para detectar um certo comando de computador em uma sequência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predefinido, o cabeçalho do comando sendo seguido por um nome do comando de uma pluralidade de nomes do comando predefinidos. O dispositivo de processamento compreende um determinador 201 para determinar se a sequência de caracteres compreende o cabeçalho de comando predefinido, o determinador sendo também configurado para determinar se uma subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando se a sequência dos caracteres compreende o cabeçalho do comando predefinido e um provedor 203 para prover o cabeçalho de comando predefinido e o nome do comando se o cabeçalho do comando compreende um nome do comando como o certo comando do computador.





PI0708228-2

"DISPOSITIVO DE PROCESSAMENTO PARA DETECTAR UM CERTO
COMANDO DE COMPUTADOR"

A presente invenção refere-se a tecnologias de informação em particular no campo da navegação móvel.

5 A demanda crescente por dispositivos de navegação móvel impõe várias restrições relacionadas com, por exemplo, o gerenciamento de conteúdos de navegação (itens de navegação como a informação de roteamento, informação de tráfego, informação de mapa, etc.) provendo, por exemplo, conteúdos
10 de navegação específicos do usuário para dispositivos de navegação, operando os dispositivos de navegação ou gerenciando os programas executáveis instalados, por exemplo, no dispositivo de navegação.

Entretanto, a complexidade crescente dos dispositivos de navegação e a demanda maior por informação implicam
15 em problemas associados com a distribuição e detecção de certos comandos para programas diferentes ou para entidades programáveis diferentes.

É o objetivo de a invenção prover um conceito para
20 prover um certo comando para um programa de computador ou para uma entidade organizada de modo programável mesmo se uma interface direta para interagir com o programa de computador ou em uma entidade organizada de modo programável não é provida.

25 A invenção é baseada na descoberta que um campo do identificador constante de recurso (URI) padrão provido por uma variedade de programas ou entidades organizadas de modo programável pode formar uma interface para, por exemplo, in-

teragir indiretamente com o programa de computador ou com a entidade organizada de modo programável. Assim, os recursos já existentes podem ser eficientemente explorados.

Uma modalidade provê um dispositivo de processamento para detectar um certo comando de computador em uma seqüência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predefinido, o cabeçalho de comando sendo seguido por um nome do comando de uma pluralidade de nomes do comando predefinidos. O dispositivo de processamento compreende um determinador para determinar se a seqüência de caracteres compreende o cabeçalho de comando predefinido, onde o determinador pode também determinar se uma subsequência de caracteres seguindo o cabeçalho de comando compreende o nome do comando se a seqüência de caracteres compreende o cabeçalho de comando predefinido. O dispositivo de processamento também compreende um provedor para prover o cabeçalho do comando predefinido e o nome do comando se o cabeçalho do comando compreende um nome do comando como o certo comando de computador. A estrutura de comando compreendendo o cabeçalho de comando predefinido e o nome do comando seguindo o cabeçalho do comando permite uma identificação de baixa complexidade do certo comando no URI.

De acordo com uma modalidade adicional, o determinador pode determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo menos um argumento de comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando. Além

do mais, o determinador pode determinar se o pelo menos um argumento do comando corresponde com um parâmetro predeterminado do certo comando, por exemplo, um caractere predeterminado. O provedor pode prover o cabeçalho de comando predeterminado, o nome do comando e o pelo menos um argumento do comando como o certo comando do computador se o pelo menos um argumento do comando corresponde com o parâmetro predeterminado. Assim, o certo comando do computador é composto em uma base das partes de comando detectadas no URI e pode, por exemplo, ser executado por, por exemplo, um processador.

De acordo com uma modalidade adicional, o dispositivo de processamento pode compreender, por exemplo, um processador organizado de modo programável para executar o certo comando.

Uma outra modalidade provê um método para detectar um certo comando de computador em uma seqüência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predeterminado, o cabeçalho de comando sendo seguido por um nome de comando de uma pluralidade de nomes de comando predefinidos. O método compreende determinar se a seqüência de caracteres compreende o cabeçalho de comando predefinido, determinar se uma subsequência de caracteres seguindo o cabeçalho de comando compreende o nome de comando se a seqüência de caracteres compreende o cabeçalho de comando predefinido e prover o cabeçalho de comando predefinido e o nome do comando se o cabeçalho do comando compreende um nome de comando como o certo comando do computador.

De acordo com uma modalidade adicional, o nome do comando é seguido por um ou mais argumentos de comando e o método compreende determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo me-
 5 nos um argumento de comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando e prover o cabeçalho de comando predefinido, o nome do comando e o pelo menos um argumento do comando como o certo comando de computador se a subsequência adicional de caracteres seguindo o nome do comando compreende o pelo menos um
 10 argumento do comando.

De acordo com uma modalidade adicional, o método compreende determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo menos um
 15 argumento do comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando, determinar se o pelo menos um argumento do comando corresponde com um parâmetro predeterminado do certo comando e prover o cabeçalho do comando predefinido, o nome do comando e o pelo
 20 menos um argumento do comando como o certo comando do computador se o pelo menos um argumento do comando corresponde com o parâmetro predeterminado.

De acordo com uma modalidade adicional, o parâmetro predeterminado separa o nome do comando e um argumento
 25 do comando adicional e o método também compreende prover o cabeçalho do comando predefinido, o nome do comando e o pelo menos um argumento do comando e o argumento do comando adicional como o certo comando do computador.

De acordo com uma modalidade adicional, o certo comando indica o estabelecimento de uma conexão de rede em uma entidade de rede remota via uma rede de comunicação para obter informação específica do usuário.

5 De acordo com uma modalidade adicional, o método compreende executar de modo programável o certo programa de computador.

De acordo com um outro aspecto, uma entidade organizada de modo programável pode controlar uma entidade adicional transmitindo ou solicitando para transmitir um certo comando para controlar uma entidade organizada de modo programável adicional para um endereço de rede não existente, isto é, fictício. A transmissão ou a solicitação pela transmissão pode ser interceptada por uma entidade organizada de modo programável adicional (por exemplo, controle) que provê o certo comando para a entidade organizada de modo programável adicional. Assim, os comandos são indiretamente distribuídos, o que reduz a complexidade adicional. Sob esse aspecto, o termo "entidade organizada de modo programável" pode se referir a dispositivo organizado de modo programável ou a um programa de computador sendo capaz de funcionar em um computador.

10

15

20

Uma modalidade provê um dispositivo de comunicação com um primeiro dispositivo organizado de modo programável, um segundo dispositivo organizado de modo programável e um terceiro dispositivo organizado de modo programável. O primeiro dispositivo organizado de modo programável pode ser configurado para solicitar que o segundo dispositivo organi-

25

zado de modo programável transmita o certo comando para um endereço de rede não existente, por exemplo, um endereço fictício. O segundo dispositivo organizado de modo programável pode ser configurado para informar o terceiro dispositivo organizado de modo programável sobre a solicitação para 5 transmitir o certo comando. Em resposta à informação indicando a solicitação, o terceiro dispositivo organizado de modo programável pode executar o certo comando. Portanto, o primeiro dispositivo organizado de modo programável pode 10 controlar indiretamente o terceiro dispositivo organizado de modo programável mesmo se o primeiro dispositivo organizado de modo programável não pode interagir diretamente com o terceiro dispositivo organizado de modo programável.

De acordo com uma modalidade adicional, o segundo 15 dispositivo organizado de modo programável pode executar um segundo programa de computador provendo uma máquina virtual provendo, por exemplo, um hospedeiro ou um ambiente para um outro programa. Portanto, o primeiro dispositivo organizado de modo programável pode executar um primeiro programa de 20 computador na máquina virtual. Portanto, o primeiro programa de computador funciona na máquina virtual provida pelo segundo programa de computador, de modo que o segundo programa de computador pode supervisionar o primeiro programa de computador e, por exemplo, capturar a solicitação.

25 De acordo com uma modalidade adicional, o primeiro programa de computador compreende um roteiro não executável. Além do mais, o segundo programa de computador pode prover, quando executado, uma funcionalidade de navegação de rede,

por exemplo, um navegador de rede. Além disso, o terceiro programa de computador pode prover, quando executado, uma aplicação do usuário, por exemplo, um ambiente de usuário (por exemplo, uma aplicação residencial).

5 Uma outra modalidade provê um dispositivo de comunicação sendo configurado para executar um primeiro programa de computador, um segundo programa de computador e um terceiro programa de computador, o primeiro programa de computador solicitando que o segundo programa de computador
10 transmita um certo comando para um endereço de rede não existente, o segundo programa de computador informando o terceiro programa de computador sobre a solicitação para transmitir o certo comando, o terceiro programa de computador executando o certo comando em resposta à informação indicando
15 a solicitação. Assim, quando executado, o primeiro programa de computador pode controlar indiretamente o terceiro programa de computador mesmo se, por exemplo, o primeiro programa de computador não compreende uma rotina executável para interagir diretamente com o terceiro programa de computador.
20 dor.

De acordo com uma modalidade adicional, o segundo programa de computador pode prover, quando executado, uma máquina virtual, onde o primeiro programa de computador pode funcionar na máquina virtual provida pelo segundo programa
25 de computador. Assim, quando executada, por exemplo, em um computador, o primeiro programa de computador provê em virtude da máquina virtual um ambiente possibilitando supervisionar o primeiro programa de computador e capturar a soli-

citação e, indiretamente, o certo comando.

De acordo com uma modalidade adicional, o primeiro programa de computador pode prover um roteiro não executável, o segundo programa de computador pode prover uma funcionalidade de navegação de rede, por exemplo, um navegador
5 de rede, e o terceiro programa de computador pode prover uma aplicação do usuário, por exemplo, um ambiente de usuário.

Por exemplo, o certo comando indica, por exemplo, o estabelecimento de uma conexão de rede para uma entidade
10 de rede remota via uma rede de comunicação pelo terceiro programa de computador.

Uma outra modalidade provê um método de comunicação compreendendo solicitar um primeiro programa de computador por um segundo programa de computador para transmitir um
15 certo comando para um endereço de rede não existente, prover informação indicando a solicitação para um terceiro programa de computador e executar o certo comando pelo terceiro programa de computador em resposta à informação indicando a solicitação para transmitir o certo comando.

20 Uma outra modalidade provê um programa de computador compreendendo um primeiro programa de computador, um segundo programa de computador e um terceiro programa de computador, o primeiro programa de computador, quando funcionando em um computador, solicitando o segundo programa de
25 computador para transmitir um certo comando para um endereço de rede não existente, o segundo programa de computador, quando funcionando em um computador, informando o terceiro programa de computador sobre a solicitação para transmitir o

certo comando, o terceiro programa de computador, quando funcionando em um computador, executando o certo comando em resposta à informação sobre a solicitação.

De acordo com um outro aspecto, um comportamento
5 do dispositivo de navegação pode ser emulado usando um outro programa contanto que o outro programa emule o firmware do dispositivo instalado no dispositivo de navegação. Assim, o outro programa pode ser instalado como, por exemplo, um emulador embutido em um outro dispositivo e executado para emu-
10 lar o dispositivo de navegação. Entretanto, antes da emulação do dispositivo de navegação, tem que ser garantido que o outro programa corresponde com o firmware do dispositivo, por exemplo, que um código de fonte do outro programa corresponde com o código de fonte do código de fonte do firmware do dispositivo. Assim, o usuário pode trabalhar com o
15 dispositivo de navegação usando o programa de emulação mesmo se o dispositivo de navegação está desligado.

Uma modalidade provê um dispositivo de comunicação para emular um comportamento de um dispositivo de navegação
20 em resposta à execução de um programa de firmware do dispositivo instalado no dispositivo de navegação. De preferência, o dispositivo de comunicação compreende um provedor para prover informação relacionada com o programa de firmware do dispositivo, um determinador para determinar se um pro-
25 grama de firmware atual instalado no dispositivo de comunicação corresponde com o programa de firmware do dispositivo instalado no dispositivo de navegação e um processador para executar o programa de firmware atual no dispositivo de co-

municação a fim de emular o comportamento do dispositivo de navegação se o programa de firmware atual corresponde com o firmware do dispositivo.

De acordo com uma modalidade adicional, o processador pode estabelecer uma conexão de rede para uma entidade de rede remota via uma rede de comunicação para obter uma versão atualizada do programa de firmware atual se o programa de firmware atual não corresponde com o programa de firmware do dispositivo. Além do mais, o processador pode executar a versão atualizada do programa de firmware atual para emular o comportamento do dispositivo de navegação.

De acordo com uma modalidade adicional, o provedor pode conectar no dispositivo de navegação para recuperar a informação indicando o programa de firmware do dispositivo.

De acordo com uma modalidade adicional, o determinador pode determinar se o programa de firmware atual instalado no dispositivo de comunicação corresponde com o firmware do dispositivo em uma base de uma tabela de arquivo de conteúdos compreendendo informação indicando o programa de firmware do dispositivo ou em uma base de uma tabela de arquivo de conteúdos compreendendo informação indicando o programa de firmware atual.

De acordo com uma modalidade adicional, o processador pode controlar o dispositivo de navegação em uma base da emulação do comportamento do dispositivo de navegação.

Uma outra modalidade provê um método para emular um comportamento de um dispositivo de navegação em resposta à execução de um programa de firmware do dispositivo instalado.

lado no dispositivo de navegação. O método compreende prover informação relacionada com o programa de firmware do dispositivo do dispositivo de navegação, determinar se um programa de firmware atual corresponde com o firmware do dispositivo instalado no dispositivo de navegação e executar o programa de firmware atual em um dispositivo de comunicação a fim de emular o comportamento do dispositivo de navegação se o programa de firmware atual corresponde com o firmware do dispositivo.

10 De acordo com uma outra modalidade, o método compreende estabelecer uma conexão de rede para uma entidade de rede remota via uma rede de comunicação para obter uma versão atualizada do programa de firmware atual se o programa de firmware atual não corresponde com o programa de firmware do dispositivo e executar a versão atualizada do programa de firmware atual para emular o comportamento do dispositivo de navegação.

De acordo com uma outra modalidade, o método compreende conectar no dispositivo de navegação para recuperar a informação indicando o programa de firmware do dispositivo.

20 De acordo com uma outra modalidade, o método compreende determinar se o programa de firmware atual corresponde com o programa de firmware do dispositivo em uma base de uma tabela de arquivo de conteúdos compreendendo informação indicando o programa de firmware do dispositivo ou em uma base de uma tabela de arquivo de conteúdos compreendendo informação indicando o programa de firmware atual.

Uma outra modalidade provê um dispositivo de comunicação organizado de modo programável para emular um comportamento de um dispositivo de navegação em resposta à execução de um programa de firmware do dispositivo instalado no dispositivo de navegação. O dispositivo de comunicação organizado de modo programável pode executar um primeiro programa de computador para prover informação relacionada com o programa de firmware do dispositivo, executar um segundo programa de computador para determinar se um programa de firmware atual disponível para o dispositivo de comunicação organizado de modo programável corresponde com o programa de firmware do dispositivo instalado no dispositivo de navegação e executar o programa de firmware atual no dispositivo de comunicação a fim de emular o comportamento do dispositivo de navegação se o programa de firmware atual corresponde com o programa de firmware do dispositivo.

De preferência, o dispositivo de comunicação organizado de modo programável é configurado para executar as etapas do método descrito acima.

De acordo com um outro aspecto, certas informações podem ser providas em tempo se, em resposta a uma solicitação pela certa informação provida para o servidor remoto, ao invés de transmitir a certa informação, a informação de tempo é provida que indica um instante de tempo no qual novamente se conectar no servidor remoto para obter a certa informação. Assim, os problemas associados com o gerenciamento ou avaliação da informação obsoleta por, por exemplo, um usuário são evitados.

Uma modalidade provê um método para prover certa informação para uma entidade de rede (por exemplo, um dispositivo de comunicação) de uma entidade de rede adicional (por exemplo, um servidor de rede). O método compreende

5 transmitir uma solicitação pela certa informação da entidade de rede via uma rede de comunicação para a entidade de rede adicional e transmitir a informação de tempo da entidade de rede adicional para a entidade de rede via a rede de comunicação, a informação de tempo indicando um certo instante de

10 tempo no qual se conectar na entidade de rede adicional para obter a certa informação.

De acordo com uma modalidade adicional, o método compreende estabelecer uma conexão de rede entre a entidade de rede e a entidade de rede adicional no certo instante de

15 tempo e transmitir a certa informação da entidade de rede adicional para a entidade de rede via a rede de comunicação.

De acordo com uma modalidade adicional, o método compreende estabelecer uma conexão entre a entidade de rede e a entidade de rede adicional no certo instante de tempo,

20 transmitir a certa informação da segunda entidade de rede adicional para a entidade de rede via a rede de comunicação e exibir a certa informação pela entidade de rede.

De acordo com uma modalidade adicional, o método compreende receber a solicitação pela certa informação pela

25 entidade de rede.

De acordo com uma modalidade adicional, a certa informação compreende informação de tráfego relacionada com uma rota de navegação específica do usuário.

De acordo com uma modalidade adicional, o método compreende armazenar a solicitação na entidade de rede adicional.

Uma outra modalidade provê uma entidade de rede para obter certa informação de uma entidade de rede adicional via uma rede de comunicação. A entidade de rede pode transmitir uma solicitação pela certa informação via uma rede de comunicação para a entidade de rede adicional e receber informação de tempo via a rede de comunicação da entidade de rede adicional, a informação de tempo indicando um certo instante de tempo no qual se conectar na entidade de rede adicional para obter a certa informação.

De acordo com uma modalidade adicional, a entidade de rede de acordo pode se conectar na entidade de rede adicional no certo instante de tempo para obter a certa informação.

Uma outra modalidade provê uma entidade de rede para prover uma certa informação para uma entidade de rede adicional via uma rede de comunicação em resposta a uma solicitação pela certa informação transmissível pela entidade de comunicação adicional via a rede de comunicação, a entidade de rede sendo configurada para transmitir informação de tempo via a rede de comunicação, o instante de tempo indicando um instante de tempo no qual se conectar na entidade de rede para obter a certa informação.

De acordo com uma modalidade adicional, a entidade de rede pode transmitir a certa informação no certo instante de tempo para a entidade de rede adicional se a entidade de

rede adicional estabeleceu uma conexão de rede com a entidade de rede.

De acordo com um outro aspecto, conteúdos de navegação (que podem ser específicos do usuário) podem ser eficientemente controlados em um servidor remoto sendo acessível via uma rede de comunicação. De preferência, o servidor remoto provê os conteúdos de navegação na dependência, por exemplo, dos conteúdos de navegação atualmente instalados no direito de acesso do usuário do dispositivo de navegação ou do dispositivo ou outros parâmetros. Assim, conteúdos de navegação atualizados podem ser talhados sob encomenda no servidor remoto, o que reduz a complexidade do dispositivo de navegação ou dos programas de computador gerenciando os mesmos.

Uma modalidade provê um dispositivo de comunicação para atualizar os conteúdos de navegação atuais armazenados em um dispositivo de navegação. O dispositivo de comunicação compreende um elemento de recuperação para recuperar a informação indicando conteúdos de navegação atuais do dispositivo de navegação, um transmissor para transmitir a informação indicando os conteúdos de navegação atuais para um dispositivo de rede remoto via uma rede de comunicação, um receptor para receber conteúdos de navegação atualizados do dispositivo de comunicação remoto e um processador para exibir informação relacionada com os conteúdos de navegação atualizados ou para instalar os conteúdos de navegação atualizados no dispositivo de navegação.

De acordo com uma modalidade adicional, o elemento

de recuperação pode recuperar a informação indicando a identificação do dispositivo ou identificação do usuário proveniente do dispositivo de navegação, onde o transmissor pode transmitir o dispositivo de identificação do usuário para o
5 dispositivo de comunicação remoto.

De acordo com uma modalidade adicional, o transmissor pode transmitir a informação indicando os conteúdos de navegação atuais como metadados ou como uma tabela de arquivo de conteúdos.

10 De acordo com uma modalidade adicional, o elemento de recuperação pode se conectar no dispositivo de navegação para obter conteúdos de navegação atuais do dispositivo de navegação. Por exemplo, o elemento de recuperação ou o dispositivo de comunicação pode se conectar de modo sem fio no
15 dispositivo de navegação. De acordo com uma outra modalidade, o dispositivo de navegação ou o dispositivo de comunicação pode ser conectável no elemento de recuperação via um cabo, de modo que o elemento de recuperação pode iniciar uma conexão ligada por cabo no dispositivo de navegação (ou vice-versa).
20

Uma outra modalidade provê um método para atualizar os conteúdos de navegação atuais armazenados em um dispositivo de navegação. De preferência, o método compreende recuperar a informação indicando conteúdos de navegação atuais do dispositivo de navegação, transmitir a informação in-
25 dicando os conteúdos de navegação atuais para um dispositivo de rede remoto via uma rede de comunicação, receber os conteúdos de navegação atualizados do dispositivo de comunica-

ção remoto e exibir informação relacionada com os conteúdos de navegação atualizados ou instalar os conteúdos de navegação atualizados no dispositivo de navegação.

De acordo com uma outra modalidade, o método compreende recuperar a informação indicando identificação do dispositivo ou identificação do usuário do dispositivo de navegação ou transmitir o dispositivo de identificação do usuário para o dispositivo de comunicação remoto.

De acordo com uma outra modalidade, o método compreende transmitir a informação indicando os conteúdos de navegação atuais como metadados ou como uma tabela de arquivo de conteúdos.

De acordo com uma outra modalidade, o método compreende verificar o direito de acesso de um usuário associado com o dispositivo de navegação ou verificar o direito de acesso do dispositivo de navegação para obter certos conteúdos de navegação atualizados.

De acordo com uma outra modalidade, os conteúdos de navegação ou os conteúdos de navegação atualizados compreendem informação indicando informação de roteamento ou de firmware atualizado ou informação de mapa.

De acordo com uma outra modalidade, a informação indicando os conteúdos de navegação atualizados compreende informação de prioridade indicando que um certo conteúdo de navegação tem uma prioridade maior do que um outro conteúdo de navegação.

De acordo com um outro aspecto, se um comando de computador (por exemplo, executável) compreende um nome do

comando e um ou mais argumentos do comando (por exemplo, uma lista de argumentos do comando), então um comando de computador verificável criptograficamente com tempo de processamento reduzido pode ser obtido quando, por exemplo, automaticamente assinando o nome do comando e compondo um comando de computador verificável usando o nome do comando (assinado) e sua assinatura criptográfica como argumento adicional.

Uma modalidade provê um método para prover um comando assinado criptograficamente, o comando compreendendo um nome do comando e pelo menos um argumento do comando. O método compreende gerar uma assinatura criptográfica do nome do comando e compor o comando assinado criptograficamente usando o nome do comando, a assinatura e o pelo menos um argumento do comando.

De acordo com uma modalidade adicional, o método compreende concatenar o nome do comando, a assinatura criptográfica do nome do comando e o pelo menos um argumento do comando para compor o comando assinado criptograficamente.

De acordo com uma modalidade adicional, o método compreende gerar a assinatura criptográfica do nome do comando em uma base de um algoritmo criptográfico assimétrico, em particular o algoritmo RSA (Rivest-Shamir-Adleman) ou em uma base de um algoritmo de assinatura digital.

De acordo com uma modalidade adicional, o nome do comando indica determinar uma rota de navegação para um destino determinado por pelo menos um argumento do comando e/ou indica prover um mapa de navegação determinado pelos dados de endereço indicados por pelo menos um argumento do comando

ou executar um programa de computador com o pelo menos um argumento do comando.

Uma outra modalidade provê um método para prover um comando verificado criptograficamente em uma base de um
5 comando assinado criptograficamente, o comando assinado criptograficamente compreendendo um nome do comando, uma assinatura criptográfica do nome do comando e pelo menos um argumento do comando. O método compreende decriptografar a
10 assinatura criptográfica para obter um primeiro valor criptográfico, processar o nome do comando para obter um segundo valor criptográfico, comparar o primeiro e o segundo valores criptográficos para obter um resultado de comparação e prover ou executar o nome do comando e o pelo menos um argumen-
15 to do comando como o comando verificado criptograficamente na dependência do resultado da comparação.

De acordo com uma modalidade adicional, o primeiro valor criptográfico compreende um valor de prova da assinatura criptográfica e o segundo valor criptográfico compreende um valor de prova do nome do comando.

20 De acordo com uma modalidade adicional, o método compreende prover ou executar o comando verificado criptograficamente se o primeiro e o segundo valores criptográficos são iguais. Por exemplo, o método também compreende comparar os valores de prova.

25 Uma outra modalidade provê um dispositivo para prover um comando assinado criptograficamente, o comando compreendendo um nome do comando e pelo menos um argumento do comando. O dispositivo compreende um gerador para gerar

uma assinatura criptográfica do nome do comando e um compositor para compor (por exemplo, concatenar) o comando assinado criptograficamente usando o nome do comando, a assinatura e o pelo menos um argumento do comando.

5 De preferência, o dispositivo (ou seus elementos) é configurado para executar as etapas do método acima descrito para prover o comando assinado criptograficamente.

 Uma outra modalidade provê um dispositivo para prover um comando verificado criptograficamente em uma base
10 de um comando assinado criptograficamente, o comando assinado criptograficamente compreendendo um nome do comando, uma assinatura criptográfica do nome do comando e pelo menos um argumento do comando. O dispositivo compreende um decriptografador para decriptografar a assinatura criptográfica para
15 obter um primeiro valor criptográfico, um processador para processar o nome do comando para obter um segundo valor criptográfico, um comparador para comparar o primeiro e o segundo valores criptográficos para obter um resultado de comparação e um provedor para prover ou executar o nome do
20 comando e o pelo menos um argumento do comando como o comando verificado criptograficamente na dependência do resultado da comparação. De preferência, o dispositivo (ou seus elementos) é configurado para executar as etapas do método acima descrito para prover o comando verificado criptografica-
25 mente.

 Uma outra modalidade provê um programa de computador para executar pelo menos um dos métodos inventivos quando o programa de computador funciona em um computador.

Modalidades adicionais da presente invenção serão descritas com relação às figuras seguintes, nas quais:

Figura 1 mostra um diagrama de blocos de um dispositivo de comunicação,

5 Figura 2 mostra um diagrama de blocos de um dispositivo de processamento,

Figura 3 mostra um diagrama de blocos de um dispositivo de comunicação,

10 Figura 4 demonstra um método para prover certa informação,

Figura 5 demonstra um método para prover um comando assinado criptograficamente,

15 Figura 6 mostra um diagrama de blocos de um dispositivo de comunicação para atualizar conteúdos de navegação e

Figura 7 mostra um diagrama de blocos de um cenário de comunicação.

20 A figura 1 mostra um diagrama de blocos de um dispositivo de comunicação compreendendo um primeiro dispositivo organizado de modo programável 101, um segundo dispositivo organizado de modo programável 103 conectado no primeiro dispositivo organizado de modo programável 101 e um terceiro dispositivo organizado de modo programável 105 conectado no primeiro dispositivo organizado de modo programável 105.

25 O primeiro dispositivo organizado de modo programável 101 solicita que o segundo dispositivo organizado de modo programável transmita o certo comando para um endereço de rede não existente, isto é, para um endereço HTTP não e-

xistente. Em resposta a isso, o segundo dispositivo organizado de modo programável 103 informa o terceiro dispositivo organizado de modo programável 105 sobre a solicitação para transmitir o certo comando. Em resposta à informação indicando a solicitação, o terceiro dispositivo organizado de modo programável 105 executa o certo comando em resposta à informação indicando a solicitação.

O primeiro dispositivo organizado de modo programável 101 pode, por exemplo, executar um primeiro programa compreendendo um roteiro não executável, por exemplo, um roteiro Java, dentro de um ambiente provido por um segundo programa de computador (por exemplo, um navegador de rede) sendo executável pelo segundo dispositivo organizado de modo programável 103.

O terceiro dispositivo organizado de modo programável 103 preferivelmente executa um terceiro programa de computador, por exemplo, uma aplicação do usuário (aplicação residencial), que recebe o comando porque o primeiro programa de computador indaga sobre o ambiente no qual ele funciona (o navegador de rede, e de maneira transitiva a aplicação do usuário) para enviar o comando em seu nome. O navegador de rede (o segundo programa de computador) é provido porque o roteiro Java existe em uma página da rede e não pode se comunicar diretamente com o terceiro programa de computador.

Por exemplo, o conceito acima pode ser utilizado para gerenciar conteúdos de navegação como, por exemplo, informação de roteamento ou mapas de roteamento. Por exemplo, o primeiro programa de computador pode iniciar uma comunica-

ção entre, por exemplo, o servidor remoto (compreendendo, por exemplo, páginas ativas) e o terceiro programa de computador (provendo uma aplicação de cliente residencial). Por exemplo, a aplicação residencial (ou o servidor remoto) contém uma loja em linha (rede) que vende voz, mapas, etc. Essa loja da rede é preferivelmente implementada usando um navegador de rede integrado. O navegador embutido é instruído pela aplicação residencial para transferir as páginas de rede representando a loja. Além do mais, as páginas da loja da rede podem conter roteiro Java, então as páginas transferidas podem se comunicar com o servidor remoto usando, por exemplo, a assim chamada tecnologia AJAX. Em um certo ponto, entretanto, o roteiro Java na página tem que fornecer certos comandos para o cliente de aplicação residencial. Um comando exemplar é 'transferir essa voz' depois que o usuário comprou uma voz na loja da rede. Embora o roteiro Java possa implementar o processamento de cartão de crédito, ele não pode fazer a transferência real e instalar, por exemplo, em um dispositivo de navegação, então ele envia um comando ou solicitações para enviar o comando.

O comando é enviado para o cliente de aplicação da esperança primeiro preparando uma solicitação especial (por exemplo, em JavaScript, na página da rede na loja) e a seguir enviando a solicitação para um endereço não existente (por exemplo, `http://ttlds`). O navegador de rede (por exemplo, embutido) informa o seu hospedeiro, a aplicação do cliente residencial, sobre a solicitação. O cliente residencial sabe que não existe tal endereço como `http://ttlds`, então o

cliente residencial, por exemplo, extrai o(s) comando(s) e os executa. Para acelerar o processamento adicional, a solicitação fictícia para <http://ttlds> pode então ser cancelada.

Além de transferir, por exemplo, conteúdos de navegação como mapas e voz, o cliente residencial (provido pelo terceiro programa de computador) pode também ser instruído a abrir mais páginas da rede (por exemplo, uma segunda página de loja) ou fechá-las, mostrar uma mensagem instantânea, alternar para uma certa janela dentro da aplicação residencial (por exemplo, o painel instalar no dispositivo que é útil depois da transferência de uma voz), verificar uma nova versão de aplicação residencial, verificar atualizações no firmware do dispositivo, verificar atualizações no conteúdo de navegação, verificar atualizações no emulador emulando o dispositivo de navegação ou indagar e enviar as credenciais do usuário, por meio de exemplo.

A figura 2 mostra um diagrama de blocos de um dispositivo de processamento para detectar um certo comando de computador em uma seqüência de caracteres representando um identificador constante de recursos (URI). O certo comando pode compreender um cabeçalho de comando predefinido que é seguido por um nome do comando de uma pluralidade de nomes de comando predefinidos e, opcionalmente, por um ou mais argumentos de comando. O dispositivo de processamento compreende um determinador 201 para determinar se a seqüência de caracteres compreende o cabeçalho de comando predefinido, o determinador sendo também configurado para determinar se uma subsequência de caracteres seguindo o cabeçalho do comando

compreende o nome do comando se a sequência de caracteres compreende o cabeçalho do comando predefinido e um provedor 203 conectado no determinador 201, o provedor 203 provendo o cabeçalho do comando predefinido e o nome do comando se o
5 cabeçalho do comando compreende um nome do comando como o certo comando do computador.

O conceito representado na figura 2 pode também ser implementado usando programas de computador (executáveis), por meio de exemplo. Além do mais, as descrições relacionadas com o URI também se aplicam a URLs (identificador
10 constante de recursos).

Quanto ao URI, o mesmo URI pode também ser usado como o "alvo" de uma hiperligação em uma página da rede ou em um e-mail. Nesse caso, o URI pode ser ativado por, por
15 exemplo, clicar em uma tal hiperligação. É também possível usar o URI de um programa de roteiro Java embutido em uma página da rede. Entretanto, outros programas instalados localmente no, por exemplo, computador podem também usar um tal URI. Por exemplo, um programa auxiliar de livro de endereço de e-mail pode usar esse URI para enviar um endereço de
20 um programa de e-mail para um programa de aplicação residencial sendo instalado em, por exemplo, um PC.

Por exemplo, a aplicação residencial (por exemplo, gerenciando conteúdos de navegação) oferece seu próprio UI.
25 Entretanto, conteúdos adicionais poderiam ser usados para finalidades de navegação armazenados fora da aplicação residencial. Por exemplo, um programa de e-mail contém um livro de endereços compreendendo, como a rede mundial, uma plura-

lidade de endereços. Seria útil se o cliente residencial pudesse tirar vantagem desses. Outras partes podem, por exemplo, estar interessadas em oferecer conteúdos de navegação na rede mundial em um formato que permita a instalação pela aplicação residencial que pode implementar um esquema sob encomenda de <header>:///URI que permite tal acesso.

O esquema URI que é definido por RFC 3986 define uma estrutura dentro da qual múltiplos protocolos podem ser implementados. Sistemas de operação modernos permitem que as aplicações se registrem como o manipulador indicado para URIs com um dado protocolo. A aplicação residencial (cliente residencial) provida, por exemplo, pelo programa de computador mencionado acima se registra como o manipulador para URIs começando com <header>:///.

O restante do URI é usado como uma lista de comando no cliente residencial. Usando essa abordagem, um endereço como "Rembrandtplein 35, Amsterdã" pode, por exemplo, ser codificado como <header>:///address?Amsterdam&Rembrandtplein&35.

Sob esse aspecto, os caracteres "?" e "&" são opcionais e substituíveis por outros caracteres predefinidos ou sequência de caracteres.

Qualquer aplicação exceto o cliente residencial que suporta URIs pode, quando ela encontra esse URI, solicitar que o sistema operacional (OS) resolva o URI. Por sua vez, o OS informará o cliente residencial que o usuário pretende usar "address?Amsterdam&Rembrandtplein&35". O cliente residencial então oferece ao usuário uma lista de escolhas sobre o que fazer com o endereço. Entretanto, existem múlti-

plas tarefas de navegação para as quais um endereço pode ser usado.

Usando a mesma abordagem e assumindo que "http://shop.<web address>/voices/dutch/bram.toc" é uma voz
5 de alemão, é possível criar o seguinte URI:

```
<header>:///Install?http%3a%2f%2fshop.<web  
address>%2fvoices%2fdutch%2fbram.toc
```

Se um tal URI aparecesse em uma página da rede, e um usuário clicasse nele, o navegador da rede poderia soli-
10 citar que o OS o enviasse. O OS solicitará que o cliente residencial adote as ações implicadas por Install?http%3a%2f%2fshop.<web address>%2fvoices%2fdutch%2fbram.toc, que nesse caso é a transferência de bram.toc (tabela de conteúdos) do dado endereço.

15 No seguinte, a especificação do formato dos comandos será descrita.

Com referência ao formato do URL genérico (que pode ser complacente com RFC 3986), um comando pode ter a seguinte estrutura:

20 <header>:///command-list

onde:

command-list = command ['#' command]+

command = command-name '?' argument-list command-

name = (nome de um comando suportado, ver abaixo)

25 argument-list = UriEncode(argument) ['&' argument-list]+

argument = UTF8-string

Com referência ao formato URL legado, um comando

pode ter a seguinte estrutura:

```
<header>://legacy-command-list
```

onde:

```
legacy-command-list = legacy-command [ '&' legacy-  
5 command ]+
```

```
legacy-command = command-name '(' legacy-argument-  
list ')'
```

```
legacy-argument-list = UriEncode(argument) [ ','  
legacy-argument-list ]+
```

```
10 argument = UTF8-string
```

URLs legados não podem conter o caractere "?" enquanto URLs apropriados sempre contêm. Isso torna possível distingui-los. É para ser observado que esses URLs não são complacentes com RFC. Muitas entidades de processamento de URL os alterarão durante o processamento. Em particular, muitos navegadores da rede do Windows adicionarão um "/" extra em uma tentativa de tornar o URL conforme. Alguns navegadores da rede não os suporta absolutamente. A aplicação residencial tenta interpretar URLs mal formados, para lidar com a ampla variedade de aplicações habilitadas em URL, mas usar o formato do URL legado torna essa abordagem mais complicada.

Os comandos podem ter a seguinte estrutura e argumentos:

```
25 Estrutura: EMailContact?folder-ID&entry-ID
```

Argumentos: IDs de contato de e-mail.

A aplicação residencial recuperará os dados de contato do programa de e-mail e perguntará ao usuário o que

fazer com eles (mostrar no mapa, navegar para, adicionar como favoritos, etc.).

Estrutura: Address?city&street&number

Argumentos: dados do endereço.

- 5 A aplicação residencial perguntará ao usuário o que fazer com o endereço (mostrar no mapa, navegar para, adicionar como favoritos, etc.).

Estrutura: ShowOnMap?city&street&number

Argumentos: dados do endereço.

- 10 Pelo menos um entre cidade ou rua não deve ficar vazio.

Estrutura: NavigateTo?city&street&number

Argumentos: dados de endereço.

- 15 Pelo menos um entre cidade ou rua não deve ficar vazio.

Estrutura: AddFavourite?favourite-name&city&street
&number

Argumentos: nome do favorito e dados do endereço.

- 20 Pelo menos um entre cidade ou rua não deve ficar vazio.

Install?URL

Argumentos: URL de HTTP da descrição TOC para um item a transferir e instalar se um dispositivo está conectado.

- 25 O recurso indicado pelo URL tem "text/xml" tipo MIME.

O formato da TOC (tabela de conteúdos) pode ser especificado por, por exemplo, um esquema XML no arquivo

"TOC format.xsd".

O tipo seguinte de itens pode ser instalado por um usuário (por exemplo, 3ª partes): conjuntos de dados de POI (pontos de interesse), vozes, esquemas de cor.

5 Estrutura: Execute?URL

Argumentos: URL de HTTP de um arquivo de texto contendo a lista de comando real.

O recurso indicado pelo URL tem "text/plain" do tipo MIME.

10 Esse comando pode ser usado, por exemplo, se passar os comandos como parte do URL produziria um URL muito longo.

Quando o nome do comando Execute() é usado em um comando legado, o URL pode prover uma lista de comando legado em resposta a ele.

Estrutura: SwitchTab?Tabname

Argumentos: nome de uma aba criada pelo servidor.

A aplicação residencial alternará para essa aba se presente. Se não presente, a aplicação residencial lembrará o nome e alterará para a aba depois que o servidor a criar realmente. A aplicação residencial lembrará somente um nome, um segundo comando SwitchTab substituirá o primeiro.

Estrutura: <user or host name>Page?Tabname& host-name&URL-path&anchor

25 Argumentos: Tabname da aba recentemente criada, nome do hospedeiro, trajetória (relativa ao nome do hospedeiro) e âncora URL.

A aplicação residencial primeiro verifica se uma

aba com o dado nome de aba existe ou senão ela cria uma. Essa aba então tem o seu URL ajustado para `http:// hostname.<webaddress>/URL-path#anchor`

A aplicação residencial pode rejeitar os nomes de
 5 hospedeiro com caracteres não alfanuméricos, o que impede a injeção do nome. Por exemplo, o nome de hospedeiro "badguys.com/ignore" pode ser rejeitado porque "." e "/" são inválidos.

Como será tratado abaixo, os comandos podem ser
 10 providos com uma assinatura criptográfica permitindo verificar os comandos. Por exemplo, as seguintes estruturas de comando podem ser usadas:

Estrutura: `<PublicKeyIdentifier>?command-list&signature`

Argumentos: uma lista de comando e uma assinatura.

15 Esse comando é usado para executar uma lista de comandos que pode somente ser executada por um usuário. A assinatura confirma a autenticidade da lista de comando.

`assinatura = Sign<PrivateKey> (lista de comando)`

É para ser observado que qualquer um pode verifi-
 20 car uma assinatura (com a chave pública).

`Sign<PrivateKey> (lista de comando)` pode ser implementada como segue:

A lista de comando é uma sequência UTF-8 e, portanto pode ser expressa como uma sequência de bytes. A lista
 25 de comando não será codificada no URL quando usada como a entrada de `Sign<PrivateKey>`.

Com referência ao esquema de codificação, a prova H SHA-256 dessa sequência de byte é calculada (256 bits).

Além do mais, usando a chave privada (Q, M) , o valor da assinatura $(Q^H \text{ modulo } M)$ é calculado. A assinatura é armazenada como uma sequência de bytes little-endian codificadas na base 64 (o enchimento não é requerido nem adicionado, para
 5 manter o URL curto). Como todos os argumentos, os argumentos da lista de comando e assinatura serão codificados no URL. Quando o `<PublicKeyIdentifier>( )` command-name é usado em um comando legado, o primeiro argumento deve ser preferivelmente uma lista de comando legado.

10 A decriptografia é o processo inverso: a aplicação residencial mantém a chave pública (P, M) . A prova H SHA-256 da lista de comando (depois da decodificação do URL) é calculada (256 bits). Usando a chave pública (P, M) , a assinatura adicional $(p^H \text{ modulo } M)$ é calculada. A assinatura é de-
 15 codificada na base 64 para obter $(Q^H \text{ modulo } M)$, onde o produto $(p^H \cdot Q^H) \text{ modulo } M$ é calculado. A assinatura é aceita se ela iguala a assinatura adicional, que significa $(P^H \cdot Q^H) \text{ modulo } M$ iguala 1.

O conceito acima é também aplicável se argumentos
 20 de comando não estão presentes. Além do mais, o nome do comando pode compreender um número de parâmetros atribuindo semântica aos argumentos. Por exemplo, cada argumento pode ser igualado com um parâmetro em duas maneiras possíveis citadas como a abordagem posicional e a abordagem nomeada no
 25 seguinte, que é similar a uma sintaxe de chamada de função nas linguagens de programação.

Com referência à sintaxe posicional, se um nome de comando tem, por exemplo, três parâmetros, e três argumentos

são providos, então o primeiro parâmetro é igualado com o primeiro argumento, etc. Por exemplo, o comando do endereço tem três parâmetros: City, Street e HouseNumber. O URI atual então conterá três argumentos, onde o primeiro argumento é

5 interpretado como o nome da cidade, etc.

Alternativamente, a sintaxe seguinte pode ser suportada:

Address?Street=RembrandtPlein&City=Amsterdam.

Nesse caso, cada argumento inclui o nome do parâmetro. Portanto, esses são conhecidos como argumentos "nomeados".

10

Além do mais, os parâmetros mencionados acima podem ser indicados como sendo parâmetros obrigatórios. Isso significa que deve existir um argumento correspondente. Ao invés de, por exemplo, "?" outros parâmetros podem ser usados para indicar a separação entre o nome do comando e os argumentos do comando. Além do mais, também outros mecanismos como mandar um nome de comando com comprimento fixo (que implica que a primeira letra depois dessa posição começa o

15 primeiro argumento do comando) podem ser utilizados.

20

Um <header>:///URI em geral pode conter um ou mais comandos. Assim, o caractere "#" (ou qualquer outro caracteres, por exemplo, "e") pode ser usado para concatenar os comandos.

25 A figura 3 mostra um diagrama de blocos de um dispositivo de comunicação para emular um comportamento de um dispositivo de navegação em resposta à execução de um programa de firmware do dispositivo instalado no dispositivo de

navegação. O dispositivo de comunicação compreende um provedor 301 para prover informação relacionada com o programa de firmware do dispositivo, um determinador 303 conectado no provedor 301 para determinar se um programa de firmware atual
5 instalado no dispositivo de comunicação corresponde com o programa de firmware do dispositivo instalado no dispositivo de navegação e um processador 305 conectado no determinador 303 para executar o programa de firmware atual no dispositivo de comunicação a fim de emular o comportamento do dispositivo de navegação se o programa de firmware atual corres-
10 ponde com o firmware do dispositivo.

O dispositivo de comunicação pode ser organizado de modo programável e configurado para executar a aplicação residencial acima mencionada que pode controlar o processo
15 de emulação.

O conceito de emulação também suporta a portabilidade e a usabilidade do dispositivo de navegação. Geralmente, o dispositivo de navegação (portátil) tem, por exemplo, uma pequena tela que também serve como sua entrada. Em con-
20 traste, por exemplo, um computador (de mesa) (por exemplo, um PC) tem uma tela muito maior e métodos de entrada mais avançados. Portanto, muitas tarefas que poderiam ser executadas no dispositivo de navegação são mais facilmente feitas se o dispositivo de navegação é controlado pelo computador
25 que pode, por exemplo, ser conectado no dispositivo de navegação. Além disso, o processador (por exemplo, a CPU) do computador é um mais rápido.

Além do mais, o conceito de emulação pode ser im-

plementado de modo programável e, por exemplo, embutido na aplicação residencial para efetivamente possibilitar, por exemplo, que o computador (representando uma modalidade do dispositivo de comunicação) controle o dispositivo de navegação usando os recursos de computador mais capazes.

Quando o dispositivo de navegação é conectado no cliente residencial (ou no dispositivo de comunicação), a versão de firmware instalada no dispositivo de navegação é determinada. Se o cliente residencial tem a versão equivalente do computador do firmware, esse firmware é então iniciado. Se nenhuma versão equivalente está disponível, porém uma conexão de rede (por exemplo, uma conexão da Internet) está disponível, o cliente residencial pode perguntar ao servidor residencial qual é a versão equivalente e onde que ela pode ser transferida.

A figura 4 demonstra um método para prover certa informação para uma entidade de rede de uma entidade de rede adicional. O método compreende transmitir 401 uma solicitação pela certa informação da entidade de rede via uma rede de comunicação para a entidade de rede adicional e transmitir 403 informação de tempo da entidade de rede adicional para a entidade de rede via a rede de comunicação, a informação de tempo indicando um certo instante de tempo no qual se conectar na entidade de rede adicional para obter a certa informação. O método pode ser, por exemplo, implementado em software (por exemplo, como uma aplicação residencial) que, quando executado, executa as etapas do método.

Por exemplo, um usuário pode solicitar da aplica-

ção residencial um relatório de tráfego diário (certa informação e específica do usuário) em certas estradas. A aplicação residencial então mostrará, por exemplo, diariamente na hora solicitada, um relatório sobre congestionamentos de tráfego e incidentes nas estradas solicitadas. Quando a solicitação do relatório de tráfego é feita no cliente residencial, o cliente, por exemplo, imediatamente envia a solicitação para o servidor. Os dados da solicitação são armazenados no servidor, para o dado usuário. Em resposta à solicitação, o servidor envia de volta um tempo no qual o cliente residencial deve sondar o servidor residencial. Também, sempre que o mesmo usuário entra no sistema novamente, o servidor residencial informará o cliente residencial em cujo tempo sondar o servidor. O relatório de tráfego atual é enviado pelo servidor em reação ao relatório de sondagem. A aplicação residencial, por exemplo, formata o relatório de tráfego para finalidades de exibição. O resultado é que todos os dados são armazenados no lado do servidor, mas a conexão, contudo, é iniciada pelo cliente. Isso permite que o cliente residencial, quando executado como um programa de software em um computador, mostre alertas de tráfego ao invés de barreiras de proteção ou bloqueadores de instantâneos.

Para determinar a informação do usuário a fim de obter, por exemplo, informação específica do usuário, credenciais de entrada no sistema manualmente inseridas por um usuário podem ser exploradas. Alternativamente, o cliente residencial pode ter armazenado-as anteriormente. Em qual-

quer maneira, se o cliente residencial tem credenciais de entrada no sistema, ele pode enviá-las automaticamente sempre que o cliente residencial provido por um programa de software é executado. Se o cliente residencial não tem as credenciais ainda, o usuário pode manualmente iniciar a entrada no sistema. Nesse caso, o cliente residencial apresentará as credenciais, as enviará imediatamente para o servidor remoto e também as armazenará para uso futuro.

A figura 5 demonstra um método para prover um comando assinado criptograficamente, o comando compreendendo um nome de comando e pelo menos um argumento de comando. O método compreende gerar 501 uma assinatura criptográfica do nome do comando e compor 503 (por exemplo, concatenar) o comando assinado criptograficamente usando o nome do comando, a assinatura e o pelo menos um argumento do comando.

Por exemplo, os esquemas de criptografia e decriptografia descritos em conjunto com a modalidade da figura 2 podem ser utilizados para prover e verificar a assinatura digital.

Por exemplo, um URI autenticado pode compreender um comando com dois argumentos, o primeiro argumento é uma sequência de caracteres que é um comando. O segundo argumento é a "assinatura digital" do primeiro argumento. De preferência, a criptografia da chave pública padrão é utilizada para assinar o primeiro argumento usando uma chave privada que provê a assinatura digital. Por exemplo, o cliente residencial tendo a chave pública correspondente pode assinar adicionalmente o primeiro argumento. A assinatura e a assi-

natura adicional são complementares se e somente se as chaves são iguais. Se esse é o caso, o cliente residencial acredita no primeiro comando e o executará mesmo se o comando no primeiro argumento fosse considerado perigoso.

5 O mecanismo do URI autenticado acima pode também ser usado para atualizar o firmware no dispositivo de navegação. De preferência, ele pode ter permissão para copiar ou para atualizar o comando, entretanto, sem alterá-lo. Assim, por exemplo, a atualização do dispositivo de navegação com
10 firmware errado pode causar dano irreparável para o dispositivo de navegação.

De acordo com uma modalidade, o comando assinado é então usado como um primeiro argumento em um outro comando (comando assinado criptograficamente), onde a assinatura é
15 usada como, por exemplo, um segundo argumento. A estrutura do comando assinado criptograficamente pode ser como segue:

```
verify_command (commandname1 (argument1, argument2, argument3), signature)
```

Três assinaturas (segundo argumento para o comando
20 externo) podem ser geradas aplicando, por exemplo, uma chave privada (secreta) no comando que deve ser autenticado, isto é:

```
commandname1 (argument1, argument2, argument3) +  
private_key = signature
```

25 De modo a gerar uma assinatura adicional, um algoritmo similar pode ser usado:

```
Commandname1 (argument1, argument2, argument3) +  
public_key = countersignature
```

O comando é verificado se a assinatura e a assinatura adicional correspondem entre si. De preferência, as abordagens de criptografia de chave pública padrão como, por exemplo, o algoritmo RSA ou o DSA podem ser utilizadas para as finalidades de assinatura.

Por exemplo, o comando

```
<header>:///<PublicKeyIdentifier>?Install%3fhttp%
3a%2f%2fintranet%2fplus%2fdocuments%2ftemp%2fnavcore_6.522.7709.
go510-go710.toc&JYrSIQ
```

é considerado, onde <PublicKeyIdentifier> é um nome de comando real usado para comandos autenticados por um usuário. O primeiro argumento é

```
Install%3fhttp%3a%2f%2fintranet%2fplus%2fdocuments
%2ftemp%2 fnavcore_6.5 22.7709.go510-go710.toc
```

Esse é o comando embutido. O nome do comando real é instalar, o "?" é codificado como %3f e o restante é o argumento do comando instalar. O usuário assinou o comando instalar com sua chave privada, e a assinatura que apareceu foi JYrSIQ. Para verificar o comando, uma assinatura adicional pode ser gerada. A assinatura adicional igualará se o comando foi corretamente assinado.

Se um outro usuário tenta editar o comando, por exemplo, <header>:///<PublicKeyIdentifier>?Install%3fhttp %3a%2f%evil.com%2fnavcore_6.522.7709.go510-go710.toc&JYrSIQ, uma assinatura adicional diferente será calculada que não iguala a assinatura. Assim, a instalação do conteúdo de evil.com será rejeitada.

A figura 6 mostra um dispositivo de comunicação

para atualizar conteúdos de navegação atuais armazenados em um dispositivo de navegação. O dispositivo de comunicação compreende um elemento de recuperação 601 para recuperar a informação indicando conteúdos de navegação atuais do dispositivo de navegação, um transmissor 603 conectado no elemento de recuperação para transmitir a informação indicando os conteúdos de navegação atuais para um dispositivo de rede remoto via uma rede de comunicação, um receptor 605 para receber os conteúdos de navegação atualizados do dispositivo de comunicação remoto e um processador 607 para exibir a informação relacionada com os conteúdos de navegação atualizados ou para instalar os conteúdos de navegação atualizados no dispositivo de navegação.

A informação indicando o conteúdo de navegação atual ou atualizado pode ser transmitida como uma tabela de arquivo de conteúdos indicando, por exemplo, um tipo de conteúdo, um nome de conteúdo, um tamanho de conteúdo ou uma versão de conteúdo. Além do mais, informação relacionada com a identificação do dispositivo ou identificação do usuário pode ser transmitida.

O dispositivo de comunicação remoto (por exemplo, um servidor) pode compreender um banco de dados armazenando, por exemplo, uma lista de itens de navegação disponíveis (por exemplo, mapas, informação de roteamento, fixações de GPS, etc.). Além do mais, o servidor remoto pode verificar o direito de acesso do usuário devido a, por exemplo, o estado de assinatura ou o direito de acesso do dispositivo de navegação devido a, por exemplo, a identificação do dispositivo

antes da transmissão do conteúdo atualizado.

Como mencionado acima, a informação de conteúdo pode ser resumida em um arquivo TOC (tabela de conteúdos). Um arquivo TOC exemplar é listado abaixo:

```

5  <Program>
    <Id>13214</Id>
    <Version> 1.2<Version>
    <Target>target</Target>
    <Name>NameOfItem</Name>
10 <Preview>http://download.<download_address>/somepreview.exe</Preview>
    <Location>http://download.<web address>/someitem.cab</Location>
    <Size> 546484 </Size> </Program>
    <Map>
    <Id>8977</Id>
15 <Version> 1.2<Version>
    <Target>Thistarget</Target>
    <Name>Benelux</Name><Category>Category</Category>
    <Preview>http:// download.<download_address>
    /mappreview.png</Preview> <Location>http://download.
20 <download_address>/mapdownload.cab</Location> <Icon>http://download.
    <download_address>/mapicon.png</Icon>
    <Size>46464987</Size>
    <InstallRestrictions DeviceId="35265252" />
    </Map>
25 <Voice language="en" gender="male">
    <Id>4657</Id>
    <Version>1.1 <Version> <Name>user_name</Name>
    <Preview>http://download. <download_address>/voicepreview.mp3</Pre-

```

```

view>
<Location>http://download.
<download_address>/voicedownload.cab</Location> 25 <Size>468797946231
</Size>
5 <Picture>http:// <download_address>/voicepicture.png</Picture>
<Icon>http:// <download_address>/voiceicon.png</Icon>
<Noice>
<Colorscheme>
<Id>98797</Id>
10 <Version> 1.1 </Version>
<Name>Bright Black</Name>
<Location>http://download. <download_address>/colordownload.cab</Location>
<Size>6464</Size>
</Colorscheme>
15 <Poi>
<Id>65446</Id>
<Version> 1.3</Version>
<Name>Wifi Hotspots</Name>
<Category>Con nectivity</Category>
20 <Location>http://download. <download_address>/poidownload.cab</Location>
<Size>54567</Size>
</Poi>
</Toc>

```

O campo <InstallRestrictions DeviceId="35265252"

25 /> especifica restrições na capacidade de instalação desse item (à parte o elemento <Target>). Ele não tem conteúdo. Atributos opcionais são: deviceId (seqüência) representando que o item pode somente ser instalado em um dispositivo com

esse ID de dispositivo e minNavigatorVersion (seqüência) representando a versão mínima do programa de navegador que deve ser instalada para esse item ser instalado.

A figura 7 mostra um diagrama de blocos de um cenário de comunicação com um programa aplicativo executável 701 (aplicação suave, aplicação residencial) funcionando, por exemplo, em um computador que se comunica via uma rede de comunicação 703 com um servidor remoto 705 para trocar informação como representado na figura 7. O servidor 707 pode prover mensagens expondo em mensagens de balão 707 ou em componentes do navegador, por exemplo, providos pelo programa aplicativo 701.

As combinações particulares de elementos e aspectos nas modalidades acima descritas são exemplares somente, a troca e a substituição desses ensinamentos por outros ensinamentos nessa e nas patentes/pedidos incorporados por referência são também expressamente consideradas. Como aqueles versados na técnica reconhecerão, variações, modificações e outras implementações do que é descrito aqui podem ocorrer para aqueles versados na técnica sem se afastar do espírito e do escopo da invenção como reivindicada. Dessa maneira, a descrição precedente é por meio de exemplo somente e não é planejada como limitadora. O escopo da invenção é definido nas reivindicações seguintes e nos seus equivalentes. Além do mais, os sinais de referência usados na descrição e reivindicações não limitam o escopo da invenção como reivindicada.

REIVINDICAÇÃO

1. Dispositivo de processamento para detectar um certo comando de computador em uma sequência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predefinido, o cabeçalho de comando sendo seguido por um nome do comando de uma pluralidade de nomes do comando predefinidos, o dispositivo de processamento **CARACTERIZADO** pelo fato de que compreende:

um determinador (201) para determinar se a sequência de caracteres compreende o cabeçalho de comando predefinido, o determinador sendo também configurado para determinar se uma subsequência de caracteres seguindo o cabeçalho de comando compreende o nome do comando se a sequência de caracteres compreende o cabeçalho de comando predefinido e

um provedor (203) para prover o cabeçalho do comando predefinido e o nome do comando como o certo comando do computador se o cabeçalho do comando compreende um nome do comando.

2. Dispositivo de processamento, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o determinador (201) é também configurado para determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo menos um argumento de comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando, o determinador (201) sendo também configurado para determinar se o pelo menos um argumento do comando corresponde com um parâmetro predeterminado do certo

comando, o provedor (203) sendo configurado para prover o cabeçalho de comando predefinido, o nome do comando e o pelo menos um argumento do comando como o certo comando do computador se o pelo menos um argumento do comando corresponde
5 com o parâmetro predeterminado.

3. Dispositivo de processamento, de acordo com a reivindicação 1 ou 2, **CARACTERIZADO** pelo fato de que também compreende um processador organizado de modo programável para executar o certo comando.

10 4. Método para detectar um certo comando de computador em uma sequência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predefinido, o cabeçalho de comando sendo seguido por um nome de comando de uma pluralidade de nomes de comando predefinidos, o método **CARACTERIZADO**
15 pelo fato de que compreende:

determinar se a sequência de caracteres compreende o cabeçalho de comando predefinido,

determinar se uma subsequência de caracteres seguindo o cabeçalho de comando compreende o nome de comando
20 se a sequência de caracteres compreende o cabeçalho de comando predefinido e

prover o cabeçalho de comando predefinido e o nome do comando como o certo comando do computador se o cabeçalho do comando compreende um nome de comando.
25

5. Método, de acordo com a reivindicação 4, **CARACTERIZADO** pelo fato de que o nome do comando é seguido por um ou mais argumentos de comando, o método compreendendo:

determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo menos um argumento de comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando e

5 prover o cabeçalho de comando predefinido, o nome do comando e o pelo menos um argumento do comando como o certo comando de computador se a subsequência adicional de caracteres seguindo o nome do comando compreende o pelo menos um argumento do comando.

10 6. Método, de acordo com a reivindicação 4 ou 5, **CARACTERIZADO** pelo fato de que compreende:

determinar se uma subsequência adicional de caracteres seguindo o nome do comando compreende pelo menos um argumento do comando se a subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando,

15

determinar se o pelo menos um argumento do comando corresponde com um parâmetro predeterminado do certo comando e

prover o cabeçalho do comando predefinido, o nome do comando e o pelo menos um argumento do comando como o certo comando do computador se o pelo menos um argumento do comando corresponde com o parâmetro predeterminado.

20

7. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que o parâmetro predeterminado se para o nome do comando e um argumento do comando adicional, o método também compreendendo prover o cabeçalho do comando predefinido, o nome do comando e o pelo menos um argumento do comando e o argumento do comando adicional como o certo

25

comando do computador.

8. Método, de acordo com qualquer uma das reivindicações 4 a 7, **CARACTERIZADO** pelo fato de que o certo comando indica o estabelecimento de uma conexão de rede em uma entidade de rede remota via uma rede de comunicação para obter informação específica do usuário.

9. Método, de acordo com qualquer uma das reivindicações 4 a 8, **CARACTERIZADO** pelo fato de que compreende executar de modo programável o certo programa de computador.

10 10. Programa de computador, **CARACTERIZADO** pelo fato de que executa o método de acordo com qualquer uma das reivindicações 4 a 9 quando o programa de computador funciona em um computador.

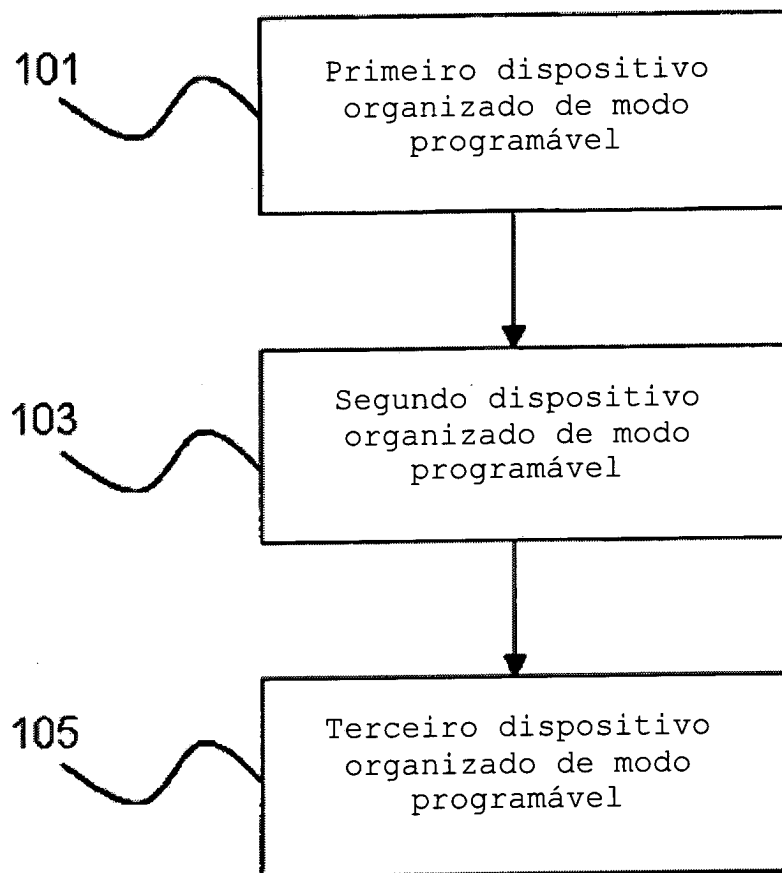


Fig. 1

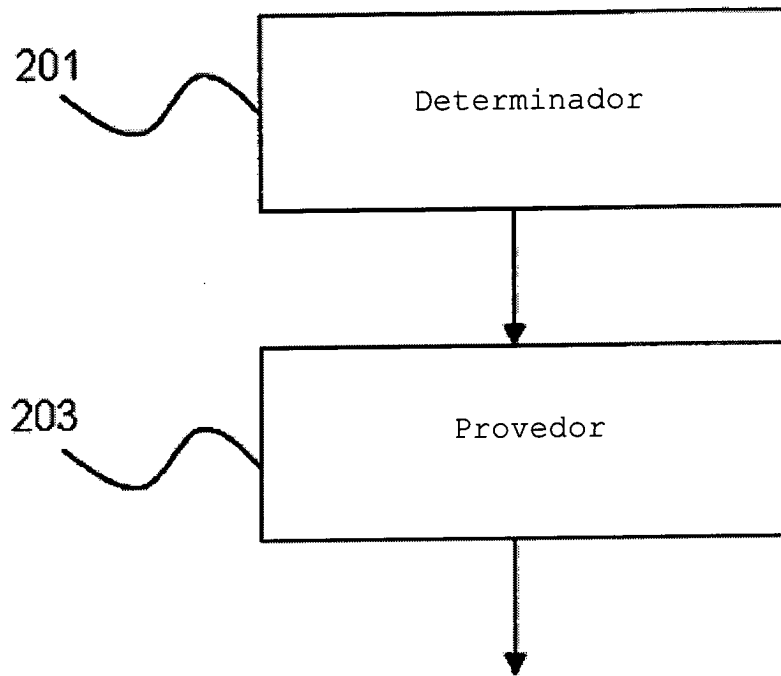


Fig. 2

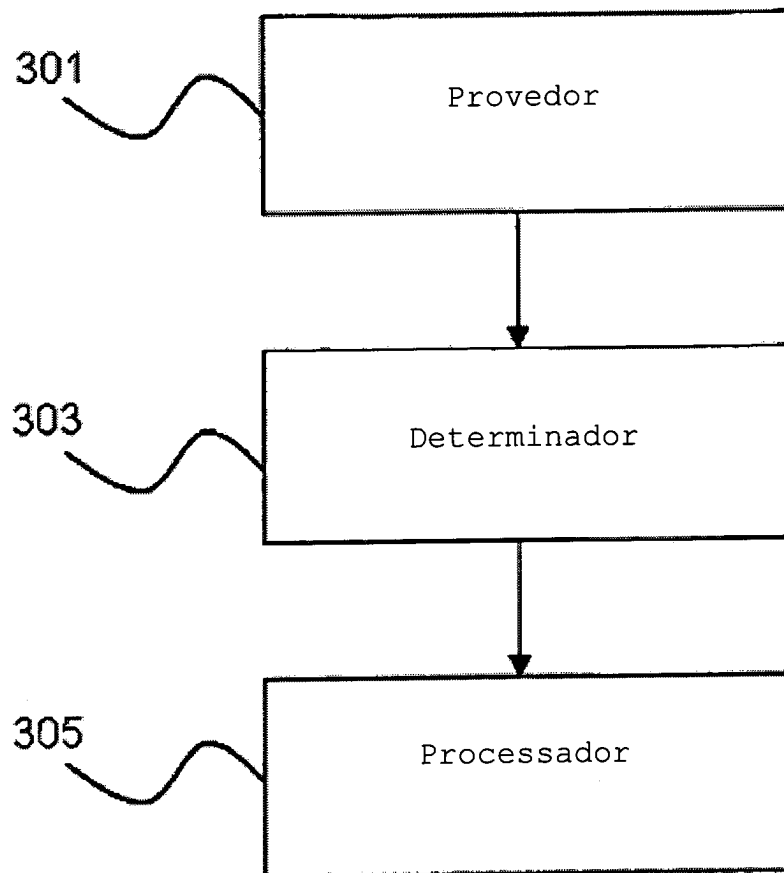


Fig. 3

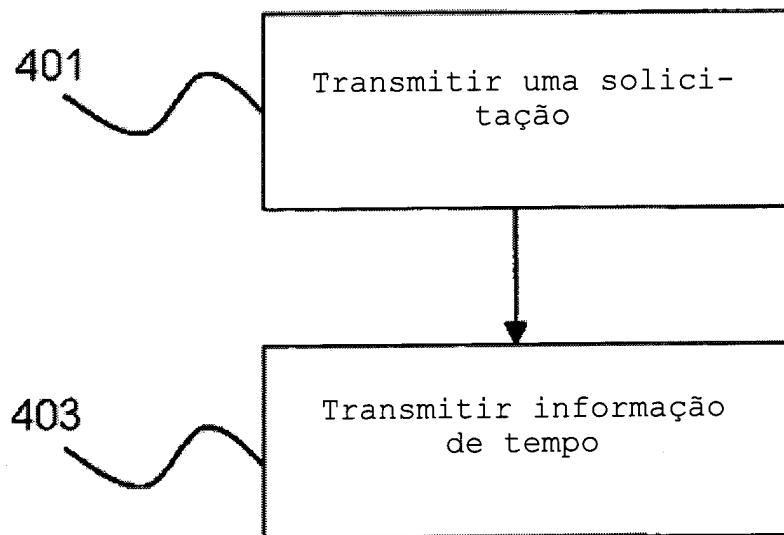


Fig. 4

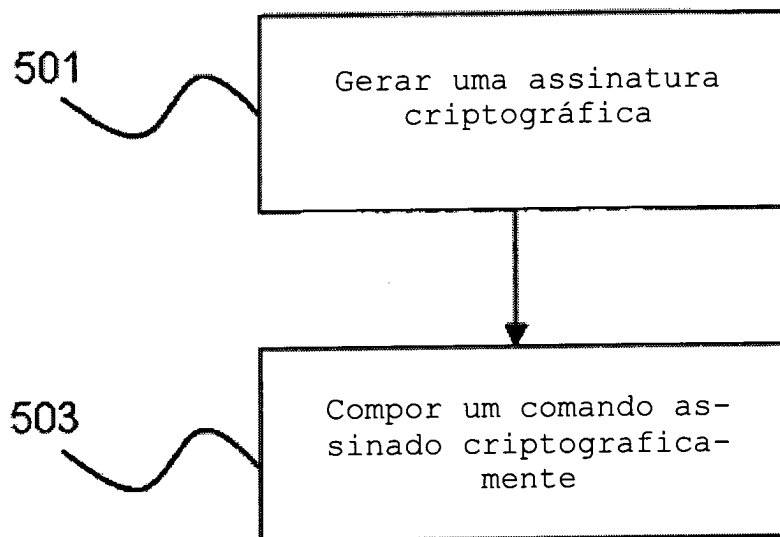


Fig. 5

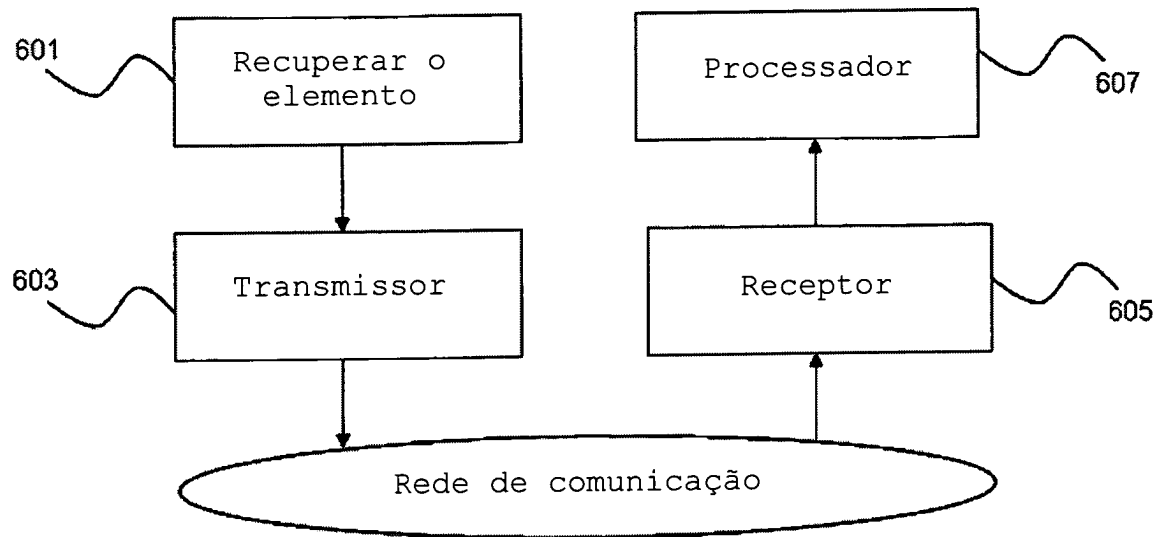
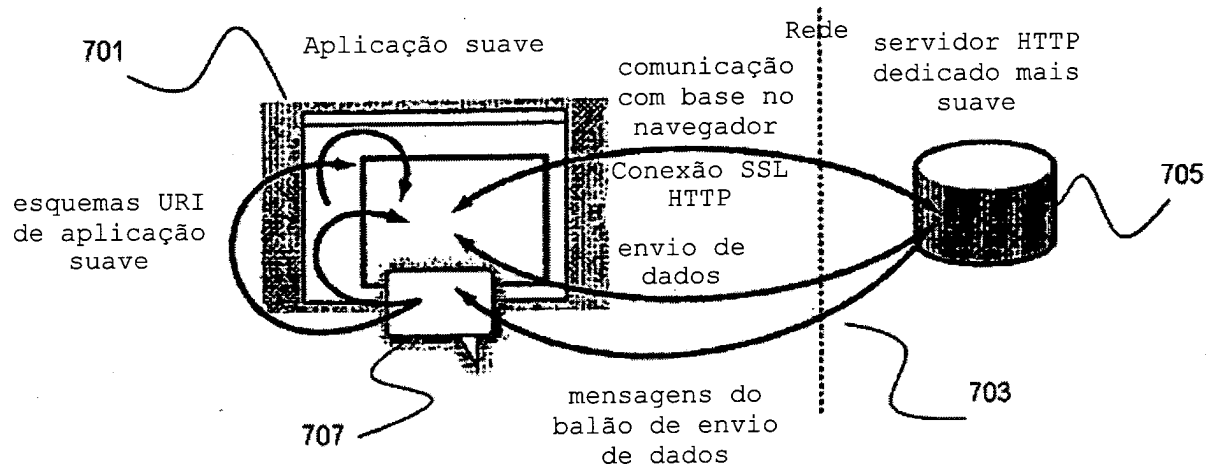


Fig. 6



RESUMO

"DISPOSITIVO DE PROCESSAMENTO PARA DETECTAR UM CERTO COMANDO DE COMPUTADOR"

A invenção provê um dispositivo de processamento para detectar um certo comando de computador em uma seqüência de caracteres representando um identificador constante de recursos, o certo comando compreendendo um cabeçalho de comando predefinido, o cabeçalho do comando sendo seguido por um nome do comando de uma pluralidade de nomes do comando predefinidos. O dispositivo de processamento compreende um determinador 201 para determinar se a seqüência de caracteres compreende o cabeçalho de comando predefinido, o determinador sendo também configurado para determinar se uma subsequência de caracteres seguindo o cabeçalho do comando compreende o nome do comando se a seqüência dos caracteres compreende o cabeçalho do comando predefinido e um provedor 203 para prover o cabeçalho de comando predefinido e o nome do comando se o cabeçalho do comando compreende um nome do comando como o certo comando do computador.