UNIVERSIDADE FEDERAL DE SANTA CATARINA
MARCO ANTONIO FERREIRA VERA

APLICAÇÃO DE SINGLE SIGN-ON E APOIO À TOMADA DE DECISÃO UTILIZANDO FERRAMENTAS OPENSOURCE

Florianópolis
2009

MARCO ANTONIO FERREIRA VERA

APLICAÇÃO DE SINGLE SIGN-ON E APOIO À TOMADA DE DECISÃO UTILIZANDO FERRAMENTAS OPENSOURCE

Trabalho de conclusão de curso apresentado ao Curso de Sistemas da Informação da Universidade Federal de Santa Catarina (UFSC) como requisito à obtenção do grau de Bacharel em Sistemas da Informação.

Orientador: José Eduardo De Lucca, Dr.

Florianópolis

2009

Marco Antonio Ferreira Vera

Aplicação de single sign-on e apoio à tomada de decisão utilizando ferramentas opensource

Trabalho de conclusão de curso apresentado ao Curso de Sistemas da Informação da Universidade Federal de Santa Catarina (UFSC) como parte dos requisitos à obtenção do grau de Bacharel em Sistemas da Informação**.**

Banca Examinadora:

_____
Prof. Dr. José Eduardo De Lucca
Universidade Federal de Santa Catarina
Orientador

_____
Prof. Dr. Vitório Bruno Mazzola
Universidade Federal de Santa Catarina
Banca

_____
Prof. Dr. Mario Dantas
Universidade Federal de Santa Catarina
Banca

AGRADECIMENTOS

Ao professor De Lucca, pela paciência, competência e profissionalismo nos trabalhos de orientação e por colocar seu amplo conhecimento ao dispor deste trabalho;

À minha mãe Márcia, que sempre me amparou e me incentivou a buscar novos horizontes;

À minha namorada Mayra, que me apoiou e me colocou para cima nas horas mais difíceis;

Aos meus amigos, que sempre entenderam a minha ausência e me incentivaram para que eu pudesse concluir este trabalho.

RESUMO

Com a falta de um sistema de *single sign-on*, a empresa gera um transtorno para os administradores de sistemas, pois têm que gerenciar as diferentes bases de dados em que os sistemas buscam as informações dos usuários, como, por exemplo, os usuários e as senhas, além de um inconveniente aos usuários que, a cada aplicação de acesso, deverão fornecer as suas credenciais para a autenticação. Com o crescente número de sistemas que auxiliam nos processos de uma corporação, fica visível que cada usuário possui várias credenciais para as devidas aplicações. Com isso, gera-se um transtorno para o administrador de sistemas, pois sempre há esquecimento, por parte do usuário, das credenciais. Para contornar o problema, é possível utilizar-se de ferramentas que proveem o *single sign-on*. Essas ferramentas, apesar de vários benefícios que podem trazer para a empresa, às vezes detêm alto custo por se usufruir de ferramentas que possuem código proprietário. Será mostrado que o *opensource*, ou código aberto, também fornece ferramentas adequadas para a implantação de um *single sign-on*.

Palavras-chave: *Single sign-on. Opensource.* Credenciais. Gerenciamento de identidade.

ABSTRACT

With the lack of a system of single sign-on, the company creates a nuisance for system administrators, because they have to manage the different databases that systems seek information from users, as, for example, users and passwords, as well as an inconvenience to users, each application for access, should provide their credentials for authentication. With the growing number of systems that assist in the processes of a corporation, it becomes apparent that each user has multiple credentials for the appropriate applications. Thus it causes a disruption to the system administrator, there is always forgetting, by the user, the credentials. To circumvent the problem, it is possible to use tools that provide single sign-on. These tools, despite the many benefits that can bring to the company, sometimes hold high cost for the use of tools that have proprietary code. Will be shown that opensource, also provides the tools to implement a single sign-on.

Keywords: Single sign-on. Opensource. Credentials. Identity management

LISTA DE FIGURAS

Figura 1: Mapa de gerenciamento de identidade 28
Figura 2: Diagrama de Web Single Sign-On 31
Figura 3: Fluxo de autenticação SAML 36
Figura 4: Arquitetura Penrose 39
Figura 5: Exemplo de meta diretório 40

LISTA DE SIGLAS

BI: Business Inteligence
CLR: Common Language Runtime
CRM: Customer Relationship Management
DNS: Domain Name System
ERP: Enterprise Resource Planning
HTTP: Hypertext Transfer Protocol
HTTPS: Hypertext Transfer Protocol Secure
IdPs: Identity Providers
LAN: Local Area Network
LDAP: Lightweight Directory Access Protocol
NASA: National Aeronautics and Space Administration
NIS: Network Information Service
NTLM: NT LAN Manager
PHP: Personal Home Page
PPP: Point-to-Point Protocol
RADIUS: Remote Authentication Dial In User Service
RPs: Relying Parties
SAML: Security Assertion Markup Language
SASL: Simple Authentication and Security Layer
SOAP: Simple Object Access Protocol
SSO: Single Sign-On
TI: Tecnologia da Informação
URL: Uniform Resource Locator
XRI: Extensible Resource Identifier

SUMÁRIO

1 INTRODUÇÃO 10
1.1 TEMA 11
1.2 DELIMITAÇÃO DO ESCOPO 11
1.3 OBJETIVOS 11
1.3.1 Objetivos Gerais 11
1.3.2 Objetivos Específicos 11
1.4 JUSTIFICATIVA 12
2 REFERENCIAL TEÓRICO 13
2.1 GERENCIAMENTO DE IDENTIDADE 13
2.1.1 Gerenciamento de usuário 14
2.1.2 Autenticação 14
2.1.3 Gerenciamento de acesso 15
2.1.4 Sistemas de identidade 17
2.1.5 Provisioning 18
2.1.6 User Centric 19
2.2.1 Kerberos 21
2.2.2 Radius 21
2.2.2.1 Authentication and Authorization 22
2.2.2.2 Accounting 22
2.2.3 LDAP 23
2.2.4 SASL 24
2.3 SINGLE SIGN-ON 24
3 APLICAÇÃO DE SINGLE SIGN-ON E APOIO À TOMADA DE DECISÃO UTILIZANDO FERRAMENTAS OPENSOURCE 26
3.1 VISÃO GERAL 26
3.2 DEFINIÇÕES E FERAMENTAS DE SINGLE SIGN-ON 28
3.2.1 Tipos de Single Sign-On 29
3.2.1.1 WebSSO 29
3.2.1.2 Federated SSO 32
3.2.1.3 Workstation SSO 35
3.2.2 Tipos de base de dados 36

3.2.2.1 Serviços de diretório....37
3.2.2.2 Diretório virtual....37
3.2.2.3 Meta diretórios....39
3.2.3 Tipos de linguagens de programação utilizadas....40
3.2.3.1 Java....41
3.2.3.2 Personal Home Page....43
3.2.3.3 Perl....44
3.2.3.4 Active Server Page....44
3.2.3.5 NET Framework....46
3.3 IMPLANTAÇÃO DE SINGLE SIGN-ON....47
3.3.1 Autenticação....47
3.3.1.1 Diretório virtual....48
3.3.1.2 Meta diretório....49
3.3.1.3 Serviço de diretório....49
3.3.2 Levantamento de linguagens....51
3.3.3 Escolha de ferramentas SSO....52
3.3.3.1 Java....52
3.3.3.2 Personal Home Page....53
3.3.3.3 Perl....53
3.3.3.4 Active Server Page....54
3.3.3.5 .NET Framework....54
4 CONCLUSÃO....55
REFERÊNCIA BIBLIOGRAFICA....56

1 INTRODUÇÃO

Em virtude da constante evolução das tecnologias computacionais, verifica-se uma maior utilização dos sistemas automatizados aplicados no cotidiano das pessoas e das corporações. Iniciou-se, nas empresas, o uso de aplicativos que auxiliavam no cadastro de produtos, como, por exemplo, aqueles que continham um arquivo ou pequenos bancos de dados e armazenavam os dados dessa aplicação.

À medida que a utilização desses softwares foi crescendo, bem como o desenvolvimento da computação, começou-se a utilizar outros aplicativos agregados a estes, utilizando outras bases de dados independentes.

Com o aumento da utilização de sistemas para processos internos e externos da empresa, houve maior preocupação com a segurança dos dados que estavam armazenados nos banco de dados. Por essa razão, criou-se o conceito atualmente definido de usuário e senha. Esse conceito consiste em cada usuário do sistema ter sua própria identificação e, com isso, apenas os usuários do sistema têm acesso à aplicação. Em razão da preocupação com a segurança, ocorreram, na maioria das aplicações de processos empresariais, as regras e as permissões dentro dos softwares, sendo baseado nas permissões das identificações.

Atualmente, encontra-se uma diversidade de sistemas, nos quais se aplicam a todos os níveis da organização (operacional, tático e estratégico), como, por exemplo, aplicativos para controle de estoque, programas CRM, até BI, e para cada aplicativo há um usuário e uma senha.

Identificado esse problema, nasceu o conceito de SSO, em que o usuário final não precisaria ter vários usuários e senhas, mas apenas um, através do qual teria acesso a todos os aplicativos necessários dentro da corporação.

1.1 TEMA

Neste trabalho, será abordado a definição tecnologia SSO, bem como as ferramentas que implementam essa tecnologia. O foco do uso das ferramentas será naquelas de código aberto, com o objetivo de não apresentar altos custos na utilização dessas ferramentas.

1.2 DELIMITAÇÃO DO ESCOPO

Este trabalho visa estudar a tecnologia SSO para apoiar a compreensão desta e apoiar na tomada de decisão no que tange escolher, por parte da empresa, as ferramentas para que se obtenha um *login* único através de SSO, objetivando a implantação da tecnologia em uma organização com ferramentas de código aberto.

1.3 OBJETIVOS

1.3.1 Objetivos Gerais

Fazer um estudo teórico e prático sobre a tecnologia de SSO, demonstrando a integração de diferentes aplicações e linguagens em uma identidade única de usuário utilizando-se ferramentas de código aberto.

1.3.2 Objetivos Específicos

a) Estudo de conceitos de gerenciamento de identidade e SSO;
b) Estudo de tecnologias para integração entre linguagens e autenticação de usuários; e
c) Análise das linguagens usadas no ambiente da organização.

1.4 JUSTIFICATIVA

Este trabalho tem como justificativa a melhoria no acesso dos usuários de uma organização aos seus sistemas de apoio, diminuindo problemas, como:

a) Vários usuários para o operador do sistema;
b) Diversas bases a gerenciar;
c) Esquecimento das senhas dos usuários; e
d) Falha de segurança, em razão de os operadores anotarem suas senhas em lugar de fácil acesso, causando roubo da informação.

Propõe-se, através de um sistema de SSO, um modo melhor de gerenciar usuários, permitindo o controle de acessos às mais variadas aplicações disponíveis em uma corporação para apoio a processos.

2 REFERENCIAL TEÓRICO

2.1 GERENCIAMENTO DE IDENTIDADE

O profissional de tecnologia da informação de hoje em dia trabalha na tentativa de fornecer aos clientes disponibilidade de serviço, transações seguras e acesso a dados a partir de qualquer computador pessoal ou dispositivo conectado à internet, em um ambiente regulador incrivelmente exigente, com o objetivo de fornecer acesso seguro a *e-mails*, aplicações, documentos e dados.

Com o crescente número de sistemas que apoiam funcionários, gerentes, diretores e analistas dentro de uma organização, houve aumento da complexidade de gerenciar controles de acessos e usuários para essas diversas aplicações, gerando maior tempo, o que acarreta em custo para a área de tecnologia da informação de uma empresa.

O conceito chamado gerenciamento de identidade, do inglês *Identity Management*, visa à resolução desse problema. Segundo Wikipedia (2009), gerenciamento de identidade são sistemas integrados de políticas e processos organizacionais que pretende facilitar e controlar o aceso aos sistemas de informação.

De acordo com ComputerWeekly (2007), o gerenciamento de identidade tem como características:

a) Responsabilidade;
b) Acesso limitado;
c) Comportamento do usuário;
d) Proteção individual de identidade; e
e) Criação e remoção de identidade.

Através do gerenciamento de identidade, é possível administrar os usuários para que seja possível criá-los, dando as permissões nos sistemas e fazendo com que os usuários obtenham acessos limitados aos sistemas. O gerenciamento de usuários gerencia também a remoção desses usuários, quando requisitado por desligamento ou por qualquer outro motivo de dentro de uma corporação, removendo-os de todas as bases e retirando todos os seus acessos.

Segundo o *Identity Management OSS Map* (2006), o gerenciamento de identidade possui até seis conceitos reunidos para a integração da identidade, com segurança e acesso.

2.1.1 Gerenciamento de usuário

O gerenciamento de usuário provê um local no qual se armazenam as credencias de cada usuário dos sistemas, sendo eles sistemas operacionais ou sistemas corporativos.

No que diz respeito ao armazenamento, segundo o *Identity Management OSS Map* (2006), existem três tipos de armazenamento, quais sejam:

a) Serviço de diretório: é uma base de dados para armazenamento, que tem como padrão de busca o LDAP, que armazena credenciais de usuários, certificados, serviços de rede, entre outros, seguindo os padrões RFC;
b) Diretório virtual: segundo o *Identity Management OSS Map* (2006), é um serviço de virtualização de dados para coleções de diretórios de rede e banco de dados que proveem dados sincronizados, replicação e acesso dinâmico ao conteúdo;
c) Meta diretório: de acordo com o *Identity Management OSS Map* (2006), é um serviço principal de dados para uma coleção de diretórios de uma rede que provê alta qualidade de interfaces com o usuário e muitas respostas interativas. O meta diretório não faz monitoramento LDAP direto ou transações de consultas LDAP em tempo real através de serviços de dados.

2.1.2 Autenticação

Autenticação é o processo no qual se verifica se alguém é realmente quem diz ser. Geralmente, isso envolve um usuário e uma senha, mas podem ser incluídos ou substituídos por outros métodos, como, por exemplo, *smartcard*, reconhecimento de voz, retina, digital.

No caso do gerenciamento de identidade, a autenticação é o *front-end* para as aplicações ou os sistemas efetuarem a verificação das credenciais dos usuários, através de protocolos de comunicação que fazem a validação na base de usuários.

Para validar uma credencial, existem alguns tipos de métodos se autenticação:

a) *Single Sign-On*: segundo o Wikipedia (2009), são sistemas independentes, que proveem autenticação do usuário apenas uma vez para permitir acesso a todos os sistemas no qual o usuário seja permitido a acessar, sem pedir as credencias cada vez que o usuário for acessar um sistema;
b) *Web Services*: segundo Loi (2007), para serem feitas as trocas de informações de autenticação entre o *Web Service Consumer* e o *Web Service Provider*, é utilizada a troca de mensagens num formato padrão, definido na especificação SOAP. Essas mensagens são requisições do cliente e respostas do servidor específicas para definir o mecanismo SASL (ou modo de autenticação);
c) *Strong Authentication*: também chamada de autenticação de dois fatores, é definida como dois das três provas: algo que se sabe, como senha; algo que se possua, como um cartão; e algo que seja único, como sua aparência ou pessoal, por exemplo, a digital biométrica;
d) Por mecanismos: nesse, utiliza-se um ou mais dos vários mecanismos já existentes e em comum uso, que podem ser através de *Kerberos*, certificados digitais, sistemas Proxy, NTLM (quando os usuários estão logados em sistemas operacionais Windows), RADIUS, entre outros.

2.1.3 Gerenciamento de acesso

Conforme o *Identity Management OSS Map* (2006), o gerenciamento de acesso é frequentemente utilizado para descrever os mais amplos sistemas que usam a autenticação e a autorização de serviços. Estes garantem que um determinado usuário tenha permissão a um determinado sistema ou recurso de rede e quais as permissões dentro desses sistemas ou da rede.

No *Identity Management OSS Map* (2006) e segundo Loi (2007), estão listados alguns projetos, como:

a) Controle de acesso a rede: provê uma forma de controlar o acesso dentro de uma rede corporativa, denominada LAN, incluindo todos os tipos de dispositivos, entre eles: servidores, compartilhamentos de arquivos, dispositivos compartilhados, entre outros;
b) OpenSSO: fornece, para uma infraestrutura de rede, o núcleo de serviços de identidade para simplificar a implementação de um SSO transparente como um componente de segurança;
c) Linguagem de política: é uma linguagem que descreve políticas de acesso aos diversos recursos da *web*. O mais comumente usado é o XACML, que é uma definição de linguagem utilizando padrões XML;
d) WP6: é um estudo para definir mecanismos de processos de autorização dentro de um *framework* para um projeto de DataGrid e para desenvolver ferramentas para implementar esse processo;
e) PAPI: é um sistema para prover controle de acesso e restringir as informações e os recursos através da internet. Essa ferramenta consiste em dois elementos independentes, o servidor de autenticação e o ponto de acesso. Isso faz com que o sistema seja muito mais flexível e se integre com diferentes ambientes;
f) SWITCH AAI: é o núcleo do sistema com o objetivo de simplificar o acesso interorganizacional para os recursos da *web*. Com apenas um *login*, um estudante pode acessar os sistemas de *e-learning* das diversas universidades na Suíça;
g) AthensAM: auxilia na proteção dos dados pessoais dos usuários através do uso de atributos pessoais;
h) VOMS: através de certificados dos atributos, a outra parte passa a confiar nas políticas dos sistemas;
i) AKENTI: é um modelo de segurança e arquitetura que tem a intenção de prover um serviço de segurança escalável em ambientes com redes distribuídas;
j) PERMIS: é uma infraestrutura que provê todas as facilidades necessárias para que os usuários gerenciem privilégios e políticas de autorização e para que as aplicações possam tomar decisões de autorização;

k) Shibboleth: provê *web* SSO através de ou com limites organizacionais e permite que os sítios tomem decisões informadas de autorização de acesso a recursos protegidos dentro dos sistemas.

2.1.4 Sistemas de identidade

Com o surgimento da web 1.0, houve a necessidade de segurança da informação disponibilizada através dela. Com isso, originou-se a autenticação através de um usuário e senha nesses sítios, nos quais são buscadas as informações e cedido o acesso às estas. O problema, porém, é que esse tipo de autenticação não provê nenhuma interoperabilidade e, assim, os usuários têm que ceder suas credenciais a cada *login*. Segundo Loi (2007), as principais características desse modelo são:

a) Registro local;
b) Falta de verificação;
c) Diretório centralizado;
d) Usuário e senha;
e) Não portátil; e
f) Falta de transparência.

Entre os modelos que estão presentes no *Open Source Identity Management Map* (2006), destacam-se:

a) Bandit: é um projeto inovador de código aberto disponibilizado para desenvolvedores como um conjunto de interfaces de programação para simplificar o processo de identidade e sistemas que permitam proporcionar uma abordagem consistente para garantir e gerenciar a identidade;
b) OSIS: integra projetos de identidade digital *opensource* e comercial a fim de permitir a esses projetos trabalhar de forma independente, mas alinhada, para construção de uma camada de identificação com um alto grau de interoperabilidade para internet;
c) Concordia: é uma iniciativa destinada a conduzir a interoperabilidade entre os protocolos de identidade em uso hoje, tendo como principal objetivo auxiliar no desenvolvimento de cenários de casos e uso, em que múltiplas

especificações de identidade e padrões ou outras iniciativas podem coexistir, reconhecendo ambientes heterogêneos.

d) Baseado em *Login*: sistemas baseados em fornecimento de usuário e senha, que tem a maior utilização atualmente na web. Esses sistemas permitem que uma pessoa ou uma máquina acesse serviços de terceiros. São sistemas utilizados pelo Google e pela Yahoo, por exemplo.

2.1.5 Provisioning

Muitos sistemas existentes no mercado ou desenvolvidos por equipes de programação usam, como base para armazenamento dos usuários, bancos de dados, pois, nessas bases, existem atributos que são necessários à aplicação, por exemplo.

Desse modo, foram encontrados problemas com autenticação em uma base única, pois uma base LDAP não possui todos os atributos necessários para uma aplicação em que se utilizam atributos específicos. Assim, aplicou-se o conceito de *provisioning*, no qual se sincroniza a base de dados única com a base de dados da aplicação.

Segundo o *Identity Management OSS Map* (2006), existem dois tipos de sincronização:

a) SPML: de acordo com o Wikipedia (2009), o *Service Provisioning Markup Language* (SPML) é um *framework* baseado em XML que foi desenvolvido pela OASIS para a troca de informações de usuário, recursos e serviços de *provisioning* entre organizações; e

b) SyncML: segundo o Wikipedia (2009), *Synchronization Markup Language* (SyncML) é o nome de um padrão de sincronização de informações independente de plataforma, definido pela Open Mobile Alliance (OMA). Muitas companhias, como Motorola, Nokia, Sony Ericsson, IBM, já suportam em seus produtos essa tecnologia.

2.1.6 User Centric

De acordo com Audun Jøsang e Simon Pope (2005), *User Centric Identity* muda o foco do *domain-centric identity management* para o usuário, oferecendo grande flexibilidade em como e onde armazenar a sua identidade, gerenciando como essa identidade será usada e compartilhada com segurança e privacidade. De acordo com Loi (2007), para alguns, o termo significa hospedado no cliente, para outros significa dar ao usuário mais opções de como e onde na rede ele poderá guardar sua própria identidade.

Segundo Loi (2007), existem diversos modelos de *user centric*. Dentro *do Open Source Identity Management Map* (2006,) os mais importantes são:

a) *CardSpace*: é um software cliente que possibilita ao usuário disponibilizar sua identidade digital a serviços *on-line* de uma forma simples, segura e confiável. Ele é conhecido como um seletor de identidade, pois, quando um usuário precisa se autenticar em um sítio ou em um *web service*, o *CardSpace* abre uma interface com um conjunto de informações do cartão para que o usuário escolha quais informações serão disponibilizadas;
b) *OpenID*: segundo o Wikipedia (2009), é um sistema de identificação que se trata de uma rede distribuída na qual a identidade do utilizador é dada por uma URL ou XRI que pode ser verificada por qualquer servidor executando o protocolo. Independente das arquiteturas SSO, *OpenID* não define um mecanismo de autenticação. Assim, a força de um *login* por *OpenID* depende de quanto o sítio sabe sobre as políticas de autenticação do provedor de identidade;
c) *Higgins*: é um *framework* que possibilita a integração de identidade de usuários com aplicações. O *Higgins* é organizado em três áreas principais, o *Higgins Selector*, o *Identity Services* e o *Higgins Identity Data Services*. O *Higgins Selector* fornece um caminho simples para gerenciar as identidades digitais. O *Identity Services* tem dois componente, o IdPs e o RPs, e é responsável por conectar os I-Card com as fontes de dados. E o Higgins *Identity Data Services* é onde se encontra os atributos dos usuários;
d) FOAF: o projeto *Friend of a Friend* (FOAF) é um tecnologia aberta e descentralizada, que permite a conexão de sítios sociais e as pessoas que eles descrevem;

e) AIAKOS: permite que se provenha um sistema central de *login* para uma rede de web sítios. Todas as atividades de *logins* e registros são feitas em um único local, permitindo que os sítios participantes recebam pacotes de autenticação criptografados pelo servidor de autenticação.

2.2 AUTENTICAÇÃO

Com o início do maior uso computacional, as organizações começaram a descobrir o que um computador poderia proporcionar através de sistemas que possibilitavam agilidades em alguns processos ou até mesmo segurança nos dados da empresa.

Para maior segurança, iniciou-se, então, o uso de autenticação a fim de prevenir que usuários não autorizados acessassem os dados dos sistemas no qual trabalhavam e mantinham atualizados.

Entre os diversos tipos de autenticação, a mais comumente usada é a por usuário e senha. A autenticação por usuário e senha, ou baseada em *login*, se dá pelo fornecimento de duas informações, a informação do usuário, no qual geralmente está atrelado a alguma permissão de acesso, e o fornecimento de senha, que está ligado diretamente ao usuário.

Esse uso comum se dá pelo fato de ter mais facilidade de implantação, pois requer apenas uma base de dados na qual se armazena o usuário, a senha e as permissões que aquele determinado usuário tem.

Em razão do maior uso em larga escala de redes, tanto intranet quanto internet, começou-se a pensar na segurança desses *logins*, tanto no que se refere a protocolo, quanto ao mau uso do próprio usuário.

No que tange à segurança de protocolo, surgiram alguns protocolos para a autenticação e validação de usuários ou até de computadores dentro de uma rede ou para uso de algum recurso específico.

2.2.1 Kerberos

Kerberos é o nome de um protocolo de rede de autenticação, desenvolvido para prover autenticação forte para aplicações cliente/servidor usando criptografia de chave secreta. O *kerberos* foi desenvolvido pelo *Massachusetts Institute of Technology* (MIT) para solucionar problemas de segurança na rede.

Segundo o Wikipedia (2009), o *kerberos* permite comunicações individuais seguras e identificadas, em uma rede insegura. Para isso, o MIT disponibiliza um pacote de aplicativos que programam esse protocolo. O protocolo *kerberos* previne *eavesdropping* e *replay attack* e ainda garante a integridade dos dados. Seus projetores inicialmente o modelaram na arquitetura cliente-servidor, e é possível a autenticação mútua entre o cliente e o servidor, permitindo, assim, que ambos se autentiquem.

O *kerberos* utiliza basicamente o protocolo Needham-Schroeder. O sistema de confiança tripla, chamado de Centro de Distribuição de Chaves (CDC), é composto por duas partes separadas: um Servidor de Autenticação (SA) e Servidor de Concessão de Ticket (SCT). O *kerberos* trabalha baseado em Tickets que identificam os usuários.

O CDC mantém um banco de dados de chaves secretas; toda entidade de rede, tanto clientes como servidores, compartilham uma chave secreta que é apenas conhecido por eles mesmos e pelo CDC. O conhecimento da chave secreta pelo CDC é necessário para a identificação das entidades de rede. Para a comunicação entre as entidades, o CDC gera uma chave de sessão temporária, que serve para garantir a privacidade das informações.

As falhas do *kerberos* estão na ausência do servidor *kerberos* que, se não estiver disponível, ninguém poderá se autenticar na rede. Para resolver esse problema, é possível ter vários servidores *kerberos* na rede.

2.2.2 Radius

O *Remote Authentication Dial In User Service* (RADIUS) é um protocolo de rede, desenvolvido pela Livingston Enterprises, Inc, em 1991, e provê uma centralização de autenticação, autorização e gerenciamento de contas

(*Authentication, Authorization and Accounting - AAA*) para computadores conectarem e usarem os serviços de rede.

As características de *Authentication* e *Authorization* é especificado pela RFC2865, e o *Accounting* é especificado pelo RFC2866.

2.2.2.1 Authentication and Authorization

O usuário ou a máquina envia um requerimento para um servidor *Network Access Server* (NAS) para adquirir acesso a um recurso de rede particular, utilizando credenciais. Essas credenciais são passadas através de um PPP, provedores DSL ou até mesmo através de formulários web baseados em HTTPS.

O NAS, então, envia uma mensagem Radius *Access Request* para o servidor Radius, solicitando autorização para garantir acesso via protocolo Radius. Essa solicitação é feita através de credenciais, que podem ser por meio de usuário e senha ou por certificados digitais providas pelo usuário. Assim, o servidor Radius verifica se as informações passadas estão corretas usando esquemas como PAP, CHAP ou EAP. Dessa forma, são verificadas as credenciais passadas, possibilitando umas das três respostas:

a) *Access reject*: o usuário é incondicionalmente negado para todas as solicitações aos recursos de rede;
b) *Access challenge*: é um método mais complexo de diálogo, que pede informações adicionais do usuário, como PIN, *token* ou cartão;
c) *Access accept*: o usuário tem o acesso requerido. Uma vez que o usuário é autenticado, o servidor Radius verifica os recursos usados pelo usuário.

2.2.2.2 Accounting

Quando o acesso é dado ao usuário, um pacote contendo o *status* da conta é enviado com uma *flag* dada como *start*. Quando o usuário termina o acesso, outro pacote é enviado como *status stop*.

Periodicamente, um pacote *interim-update* é enviado pelo NAS para o servidor Radius para atualizar o *status* da conta como ativa e, assim, mantém o usuário conectado aos recursos.

2.2.3 LDAP

Segundo o Wikipedia (2009), o *Lightweight Directory Access Protocol* (LDAP) é um protocolo para atualizar e pesquisar diretórios rodando sobre TCP/IP. Um diretório LDAP geralmente segue o modelo X.500, que é uma árvore de nós, sendo que cada nó consiste de um conjunto de atributos com seus respectivos valores.

Um diretório LDAP tende a refletir vários limites políticos, geográficos e/ou organizacionais, dependendo do modelo adotado. A utilização do LDAP hoje em dia tende a se basear nos nomes já existentes do sistema DNS, na estruturação dos níveis mais básicos de hierarquia. Mais profundamente, podem aparecer estruturas representando pessoas, unidades organizacionais, impressoras, documentos, grupos de pessoas ou qualquer outra coisa que represente um nó. O LDAP é uma definição de protocolo para acesso a bancos de dados especializados chamados diretórios. Para realizar operações nessa base, são utilizadas as seguintes operações:

a) *Bind*: autentica e especifica a versão do protocolo LDAP;
b) *Search*: procura por e/ou recupera entradas dos diretórios;
c) *Compare*: testa se uma entrada tem determinado valor como atributo;
d) ADD: adiciona uma nova entrada;
e) *Delete*: apaga uma entrada;
f) *Modify*: modifica uma entrada;
g) *Modify* DN: move ou renomeia uma entrada;
h) *Start* TLS: protege a conexão com a *Transport Layer Security* (TLS) através da porta 636;
i) *Abandon*: aborta uma requisição prévia;
j) *Extended Operation*: operação genérica para definir outras operações; e
k) *Unbind*: fecha a conexão.

2.2.4 SASL

O *Simple Authentication and Security Layer* (SASL) é um *framework* para autenticação e segurança de dados nos protocolos de internet. O SASL separa os mecanismos de autenticação dos protocolos de aplicação. Na teoria, permite que qualquer mecanismo de autenticação suportado pelo SASL seja usado em quaisquer protocolos de aplicação que usa SASL. Os mecanismos de autenticação podem também oferecer serviços para a camada de segurança de dados e confiabilidade dos dados.

Segundo Wikipedia (2009), o mecanismo do SASL é definido por uma série de solicitações e respostas, que são:

a) EXTERNAL: nesse, autenticação está implícita no contexto, por exemplo, para protocolos usando IPSec ou TLS;
b) ANONYMOUS: acesso sem autenticação;
c) PLAIN: mecanismo simples através de texto limpo;
d) OTP: mecanismo *one-time* de senha e está obsoleto pelo mecanismo SKEY;
e) SKEY: mecanismo S/KEY;
f) CRAM-MD5: esquema simples de solicitação e resposta baseado no HMAC-MD5;
g) DIGEST-MD5: esquema compatível com o esquema de solicitação e respostas baseado em MD5. Oferece uma camada de segurança dos dados;
h) NTLM: mecanismo de autenticação NT LAN *Manager*;
i) GSSAPI: para autenticação v5 do kerberos. Oferece uma camada de segurança dos dados; e
j) GateKeeper ou GateKeeperPassport: mecanismo de solicitação e resposta desenvolvida pela Microsoft para o MSN.

2.3 SINGLE SIGN-ON

Devido ao crescente aumento das aplicações, foram criando-se diversas dificuldades no gerenciamento de usuários dos vários sistemas com suas bases de dados de usuários próprias e com customizações de acesso aos diferentes

aplicativos. Além do problema de gerenciamento, os usuários, com o constante aumento de aplicações *desktop* e web, têm de digitar diversas vezes diferentes credencias para ter acesso aos aplicativos, gerando, assim, além de transtornos aos utilizadores, sistemas mais propícios às falhas de segurança, pois os usuários colocam a mesma senha para não se esquecerem e/ou a anotam em algum lugar para consulta em caso de esquecimento. Nesse caso, um invasor poderia facilmente ter acesso aos aplicativos, como *homebanking*, *e-mail*, entre outros.

Com sistemas com suporte ao *single sign-on*, é possível, através de uma base de usuários única, prover acesso por meio de permissões previamente configuradas através de gerenciamento de acesso. Segundo o Wikipedia (2009), há vários benefícios do *single sign-on,* dos quais:

a) Redução de combinações de diferentes usuários e senhas;
b) Redução do tempo de reautenticação da mesma credencial;
c) Possibilidade de suportar autenticações convencionais, como autenticação do Windows;
d) Redução de custo da tecnologia da informação, diminuindo as ligações ao *helpdesk* por problemas de senhas;
e) Segurança em todos os níveis de entrada, saída e acesso às aplicações;
f) *Single sign-on* utiliza servidores de autenticação centralizada, assegurando que não será mais necessário o usuário digitar sua credencial novamente; e
g) *Report* centralizado.

Os mecanismos mais comuns de autenticação usado na tecnologia *single sign-on* são baseados em:

a) *Kerberos*;
b) *Smartcard*;
c) *OTP Token*;
d) Autenticação integrada ao Windows; e
e) Certificados digitais de clientes.

3 APLICAÇÃO DE SINGLE SIGN-ON E APOIO À TOMADA DE DECISÃO UTILIZANDO FERRAMENTAS OPENSOURCE

Conforme o referencial teórico, sabe-se que há a possibilidade de utilizar-se de tecnologias para resolver problemas atuais de empresas que contêm os mais variados sistemas de apoio aos seus processos, tanto internos quanto externos.

Atualmente, vê-se que as organizações possuem sistemas de ERP, CRM, BI entre outros, para apoiar funcionários dos vários níveis hierárquicos. Para acessar cada sistema, geralmente o usuário tem que fornecer um *login* e uma senha em todas as aplicações para poder ter acesso aos recursos desejados, gerando tempo gasto nas autenticações e mais chamados ao *helpdesk*, perdendo tempo resolvendo problemas, como esquecimento de senhas, por exemplo.

Para tenta encontrar uma solução para contornar esse problema, diminuindo-o ou eliminando-o, será mostrada uma forma de implantar o conceito de *single sign-on* utilizando-se de ferramentas *opensource* descritas no *Identity Management OSS Map*.

3.1 VISÃO GERAL

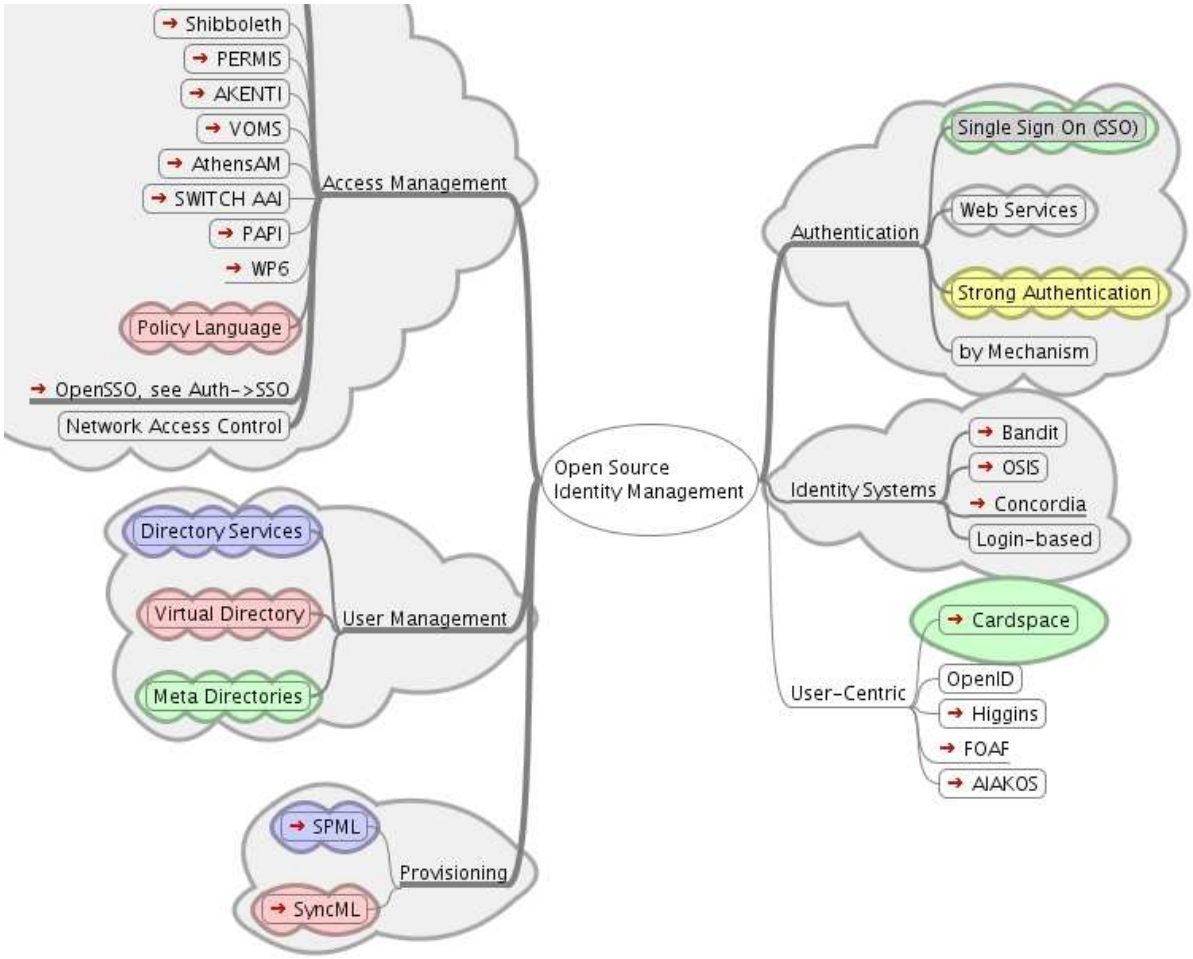
Como se pode perceber, existem muitas tecnologias, para gerenciamento de identidade, dentro das seis áreas dadas pelo *Identity Management OSS Map* (2006). Para isso, há de se escolher quais ferramentas devem ser utilizadas e quais ferramentas são mais aderentes dentro da organização, analisando os seguintes quesitos:

a) Tipos de aplicações existentes;
b) Modo de autenticação já utilizado;
c) Sistema operacional das estações; e
d) Tecnologia utilizada nas aplicações.

Com a coleta dessas informações, é possível analisar quais tecnologias poderão e/ou deverão ser utilizadas para se aplicar os conceitos de gerenciamento de identidade, obtendo-se os benefícios que foram citados no referencial teórico.

Para desenvolver um escopo de projeto SSO, devem-se utilizar algumas ferramentas do gerenciamento de identidade para que o SSO integrado com algumas dessas ferramentas possa disponibilizar um serviço de *login* único.

As ferramentas utilizadas são de código aberto ou *opensource*. A seguir, apresenta-se o mapa do Id OSS Map no qual são listadas todas as ferramentas *opensource* para o gerenciamento de identidade.



Fonte: Id OSS Map, 2006.

Figura 1: Mapa de gerenciamento de identidade

Dentro desse mapeamento feito pelo sítio Safehaus, as áreas utilizadas para implantar SSO serão o *User Management* e o *Authentication*, sendo que deste último utilizar-se-á uma subárea que será o SSO. Desse modo, conseguir-se-á

implantar um projeto de SSO dentro de uma organização e prover autenticação única para as aplicações da empresa.

3.2 DEFINIÇÕES E FERRAMENTAS DE SINGLE SIGN-ON

Com o surgimento da web, houve a disseminação de aplicativos, ou portais, que começaram a atender às corporações tanto em suas necessidades internas quando no atendimento a clientes. Com diferentes focos e necessidades, os portais passaram a suprir as necessidades do mercado, ou necessidades específicas das empresas, e, consequentemente, houve o surgimento de vários aplicativos web. Para solucionar problemas encontrados com as muitas bases de usuários, surgiu a tecnologia SSO, que visa integrar essas aplicações na autenticação e autorização.

Considerando que empresas usam vários tipos de linguagens existentes no mercado, serão mostrados os vários tipos de linguagens e como estas podem se integrar com o tipo de autenticação de SSO.

Para que uma linguagem possa fazer a autenticação do usuário, é exigido que o usuário forneça dados que são pertinentes a usuário que está tentando acessar, geralmente um usuário e uma senha. Assim, a aplicação faz autenticação em algum banco de dados, geralmente relacional, que valida ou não as informações passadas para o sistema. No caso de validação aceita, a aplicação libera o acesso aos recursos desse sistema ao usuário requerente.

Vendo esta tendência como um problema para o mercado, houve o surgimento de várias ferramentas que suprissem essa deficiência, no que diz respeito à autenticação e à autorização. No mundo *opensource*, também existem várias ferramentas capazes de trazer a funcionalidade de autenticação única para um sistema. Para escolher a ferramenta necessária, é preciso mapear alguns tópicos para que seja possível integrar o sistema, o método de autenticação e a base de dados.

Entre esses tópicos, existem:

a) Base de dados utilizada;
b) Quais as linguagens dos sistemas que serão implantadas o SSO;
c) Quais as plataformas que rodam as aplicações; e

d) Qual o método de autenticação a ser utilizado.

Estes tópicos podem auxiliar na tomada de decisão de qual ferramenta *opensource* pode ser utilizada para a integração da autenticação dos usuários em uma única aplicação sendo que será reaproveitada em várias.

3.2.1 Tipos de Single Sign-On

Atualmente, existem inúmero tipos de linguagens de programação, umas com base para a web, outras para *desktop*, outras para celulares, etc.

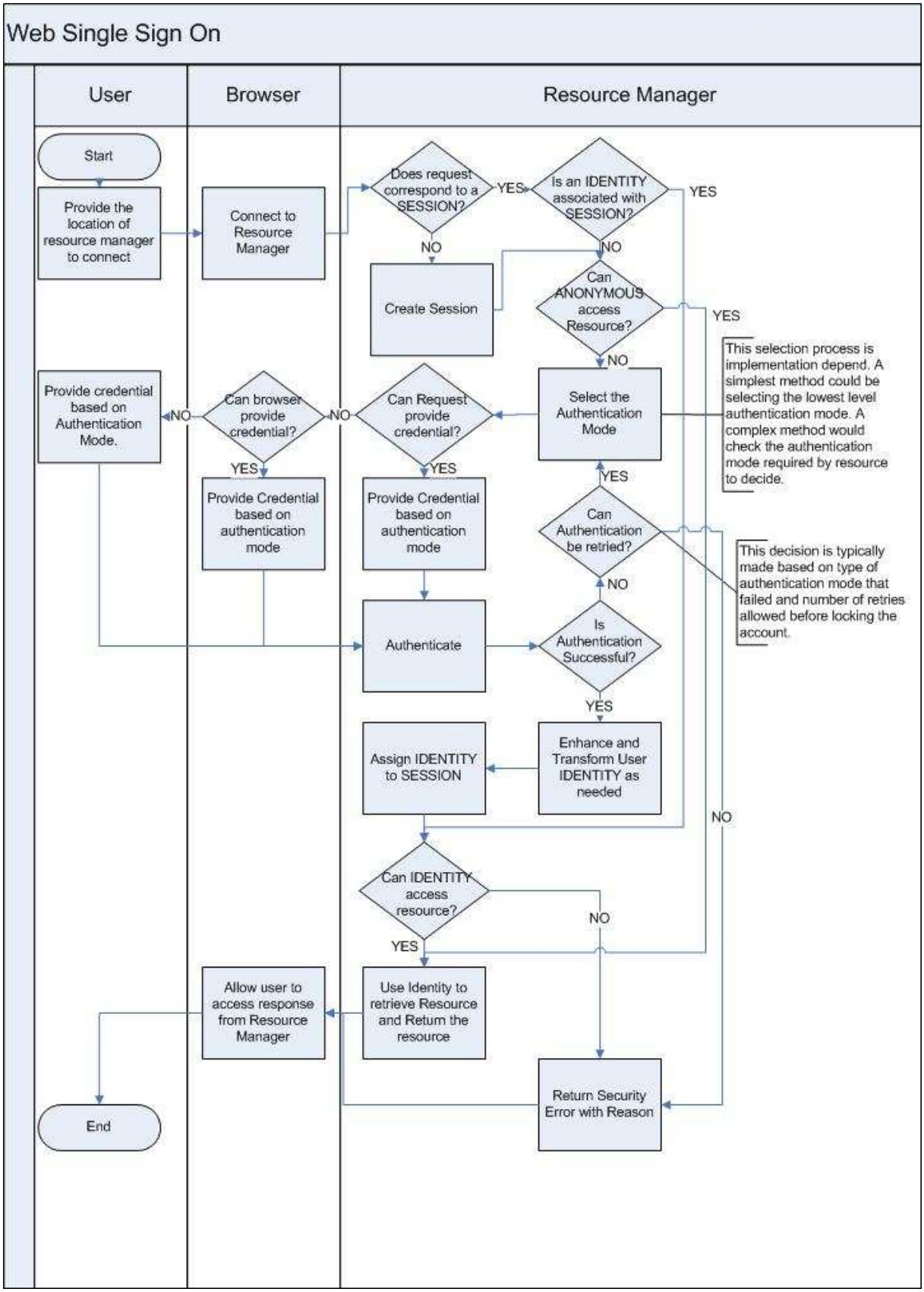
Para a implantação de *Single Sign-On*, deve-se mapear as linguagens utilizadas dentro da corporação para que se possa ver em quais tipos de SSO é possível a aplicação.

Segundo o Id OSS Map (2006), existem, em ferramentas *opensource*, três tipos de Single Sign-On: WebSSO, *Federated* SSO e *Workstation* SSO.

3.2.1.1 WebSSO

Segundo Safehaus, o WebSSO são aplicativos que proveem autenticação e acesso aos aplicativos web, extranets e portais. Seus componentes são serviço de *weblogin*, serviço de verificação, agente de aplicação web, aplicação web e web *browser*.

Como mostra o diagrama a seguir, para realizar acesso a um recurso, o usuário faz requisição de acesso, e o servidor SSO verifica se o usuário já está autenticado e se existe alguma sessão vinculada ao usuário requerente. Se não houver, o WebSSO faz requisição de autenticação para o *browser*, e se este não contiver as credenciais, o *browser* faz o pedido ao usuário. Assim, o SSO faz a autenticação através de uma base de identificação. Se a autenticação ocorrer com sucesso, o provedor de serviço garante, ou não, o acesso do usuário ao recurso.



Fonte: Identity and Access Management, 2006.

Figura 2: Diagrama de Web Single Sign-On

As ferramentas *opensource* destacadas no Id OSS Map são:

a) JASIG CAS: é uma ferramenta de SSO baseada num modelo em *kerberos* e escrita em Java. Aceita alguns padrões de autenticação, como LDAP, SPNEGO, e suporta SAML, OpenID e autenticação com o Google;
b) OpenSSO: ferramenta baseada em Java para promover SSO transparentemente. Tem como principal mantenedora a empresa Sun Microsystem;
c) JOSSO: esse SSO é baseado em J2EE. Tem como prós a integração com ASP e PHP, além de Java. Tem suporte a múltiplos mecanismos de autenticação;
d) *Pubcookie*: é um servidor de *login* que tem módulos para Apache e Microsoft ISS. Suporta autenticações, como x509, kerberos, LDAP e NIS;
e) *Cosign*: desenvolvido em C e no princípio para a Universidade de Michigan, tem suporte à autenticação x509, *kerberos* e LDAP;
f) WebAuth: sistema de autenticação que suporta *kerberos* e se integra com Apache. É mantido pela Universidade de Stanford e tem problemas de segurança por ter o *cookie* não opaco;
g) Shaj: sistema de autenticação web simples apenas suportado para Java. Faz autenticação em SAM do Microsoft Windows ou PAM do Unix;
h) A-Select: ferramenta que tem suporte aos mecanismos de autenticação, como LDAP, Radius, SMS, OTP, PKI, *Tokens* e biométrica. É compatível apenas com aplicações Java e roda apenas em Tomcat;
i) GLAM: projeto descontinuado;
j) BlueSterm: esse projeto estava indisponível e não foi encontrado mais nenhuma documentação na internet;
k) eID WebAuth: ferramenta desenvolvida apenas para a Universidade do estado de Colorado.

Visto a grande gama de aplicações, será realizado um filtro nessas ferramentas; serão eliminados os sistemas de autenticação para não web; e não será recomendado o uso de ferramentas que:

a) Têm projeto descontinuado;
b) Têm problemas de segurança;

c) Foram desenvolvidos para atender às necessidades internas e se mantêm até hoje sem amadurecimento da ferramenta; e
d) Não foram encontradas informações necessárias para a implantação e o *link* para *download*.

Os sistemas de autenticação que não são recomendados para o uso e, portanto, não serão citados posteriormente neste trabalho são: WebAuth, eID WebAuth, BlueSterm e GLAM.

Com as outras ferramentas, pode-se implantar um projeto de SSO e, assim, escolher as ferramentas dependendo de linguagens, base de usuários e métodos de autenticação que estão sendo utilizados dentro da corporação. Ressalta-se que essas ferramentas são apenas para aplicativos web.

3.2.1.2 Federated SSO

Um sistema *Federated* SSO tem um funcionamento similar ao WebSSO, com a diferença de autenticação de usuários e privilégios em outros domínios, e utiliza o protocolo SOAP e SAML para gerar o *token*.

No padrão do *Federated* SSO, é usado o padrão SAML para fazer a autenticação em outro domínio. Para entender o mecanismo, estudar-se-á um pouco o SAML, que é proposto pela OASIS.

Segundo o Wikipedia (2009), *Security Assertion Markup Language* (SAML) é um padrão baseado em XML para autenticação e autorização de dados entre domínios seguros.

O SAML foi desenvolvido para solucionar problemas encontrados em WebSSO, que podem ser resolvidos dentro de ambientes nos quais são disponibilizados para a internet, em que as tecnologias não têm interoperabilidade. SAML, então, define um padrão entre soluções de SSO.

Segundo o Wikipedia (2009), o SAML, baseado em XML, define asserções, protocolos, ligação e perfis. O SAML *Core* refere-se à sintaxe e semântica geral das asserções SAML, bem como o protocolo usado para solicitar e transmitir essas afirmações de uma entidade do sistema para outro. O protocolo SAML refere-se ao

que irá ser transmitido e não como, e isso é determinado pelo *binding*. O SAML *bindings* define uma requisição e uma resposta de um SAML, mapeando em uma mensagem padrão ou protocolo de comunicação. Uma importante ligação é o SAML SOAP. Um perfil SAML é uma manifestação concreta de um caso de uso definido, usando uma combinação específica de declarações, protocolos e vinculações.

Asserções SAML, conforme Wikipedia (2009), são normalmente transferidos de provedores de identidade para os prestadores de serviços. As afirmações contêm declarações que os prestadores de serviços utilizam para tomar decisões de controle de acesso. Três tipos de declarações são fornecidos pelo SAML:

a) Declaração de autenticação;
b) Declaração de atributos; e
c) Declaração de decisão de autorização.

Declaração de autenticação garante ao prestador do serviço que o *principal* realmente autentique com o provedor de identidade a uma determinada hora, utilizando um método específico de autenticação. Outras informações sobre os *principals* autenticados (chamado de contexto de autenticação) podem ser divulgadas em uma declaração de autenticação (WIKIPEDIA, 2009).

Uma declaração de atributos garante que um sujeito esteja associado com determinados atributos. Um atributo é simplesmente um par nome-valor. Baseia-se em usar as partes dos atributos para tomar decisões de controle de acesso (WIKEPEDIA, 2009).

Uma declaração de autorização de decisão, de acordo com Wikipedia (2009), garante que um sujeito esteja autorizado a realizar uma ação A em um recurso R dado prova E. A expressividade das declarações de autorização de decisão em SAML é intencionalmente limitada.

Um protocolo de SAML descreve como alguns elementos SAML (incluindo afirmações) são empacotados dentro dos elementos de requisição e resposta do SAML e dá as regras de processamento que as entidades devem seguir ao SAML quando produzir ou consumir esses elementos. Para a maioria, um protocolo de SAML é um simples protocolo de pedido de resposta (WIKIPEDIA, 2009).

Segundo Wikipedia (2009), o tipo mais importante de protocolo de requisição SAML é chamado de *query*. Um prestador de serviço faz uma consulta diretamente a um provedor de identidade através de um canal seguro. Assim, a consulta está

geralmente ligada ao SOAP. Correspondente aos três tipos de declarações, há três tipos de consultas SAML:

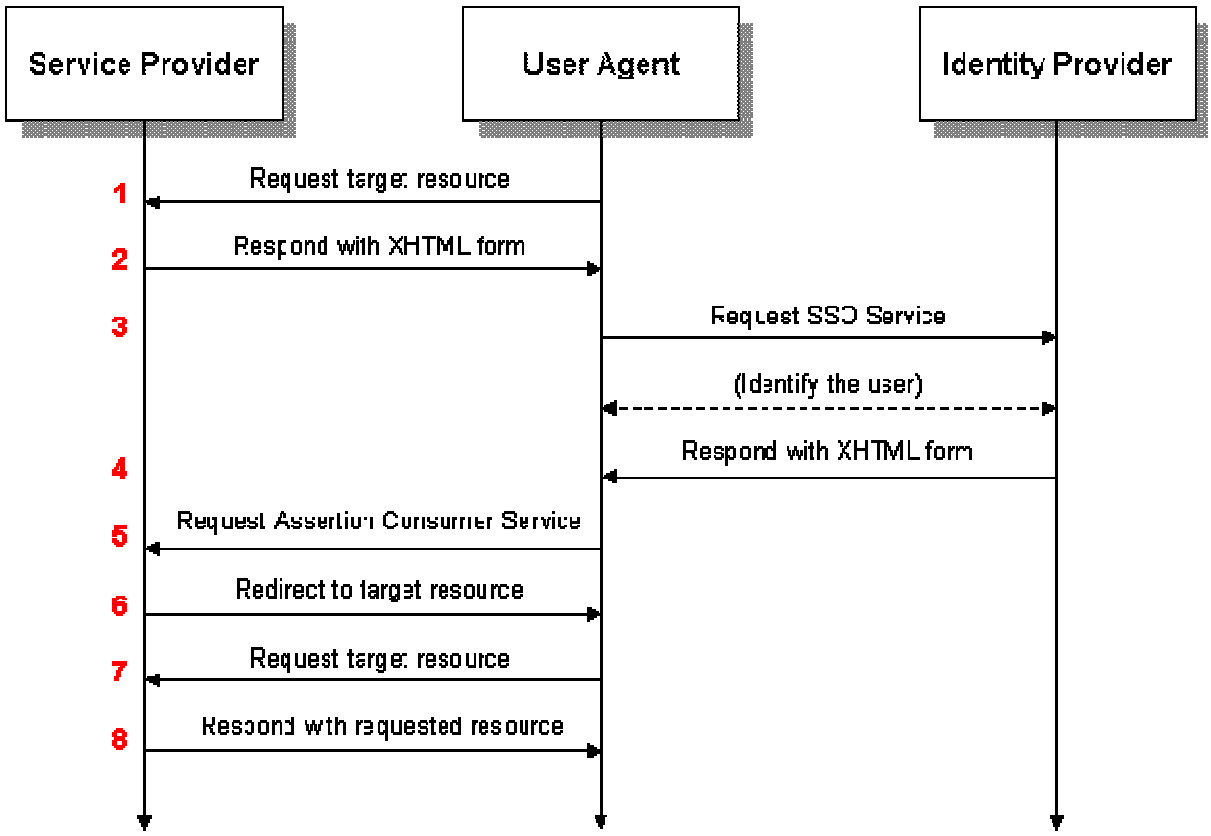
a) Consulta de autenticação;
b) Consulta de atributo; e
c) Consulta de decisão de autorização.

Desses, a consulta atributo seja, talvez, a mais importante (é ainda objeto de muita pesquisa). O resultado de uma consulta de atributo é uma resposta SAML com uma afirmação que contém em si uma declaração de atributo (WIKIPEDIA, 2009).

Um SAML *bindings* é um mapeamento de uma mensagem de protocolo padrão SAML para formatos de mensagens e/ou protocolos de comunicação. Por exemplo, um SAML SOAP *bindings* especifica como uma mensagem de SAML é encapsulada em um envelope SOAP, que, por sua vez, está ligado a uma mensagem HTTP, conforme Wikipedia (2009).

Um perfil SAML descreve em detalhes, de acordo com Wikipedia (2009), como afirmações SAML, protocolos e *bindings* combinam-se para apoiar um caso de uso definido. O mais importante é o perfil SAML Web Browser SSO.

O caso de uso primário SAML é chamado de navegador da Web Single Sign-On (SSO). Um usuário controla um agente do usuário (geralmente um navegador web) e solicita um recurso da web protegido por um prestador de serviços SAML. O prestador de serviços, que pretende conhecer a identidade do usuário solicitante, efetua uma solicitação de autenticação a um provedor de identidade SAML através do agente do usuário. O fluxo resultante do protocolo é descrito no diagrama a seguir.



Fonte:Wikipedia, 2009.

Figura 3: Fluxo de autenticação SAML

Após algumas correções de segurança e de comunicação de protocolo, o SAML teve uma nova versão do padrão lançada pela OASIS, estando na versão 2.0.

Dentro de um escopo de SSO, tem-se a *Federated* SSO como uma única opção de resolver autenticação e autorização entre domínios seguros. Seguir-se-á um mapeamento na ajuda de tomada de decisão para a escolha de uma ferramenta *opensource*, não dando muito foco à *Federated* SSO, pois esta demanda tempo demais.

Apenas foi demonstrada para visualizar que essa tecnologia é de extrema importância em ambientes mais complexos.

3.2.1.3 Workstation SSO

Workstation SSO é um sistema de autenticação que serve para fazer a interoperabilidade de uma autenticação dos sistemas operacionais em um servidor

de SSO. Essas autenticações substituem uma autenticação através de um domínio ou servidor NIS.

Segundo o Id OSS Map (2006), existem duas ferramentas para gerenciar essas autenticações com os sistemas operacionais:

a) pGina: um servidor de autenticação que é um autenticador de ambiente Microsoft Windows;
b) PingID SAML Windows Logon: ferramenta que habilita o Windows, através de SAML, para autenticar em um PingID Federate Server.

Apesar de essas ferramentas serem úteis para autenticação de sistemas operacionais Microsoft Windows, o uso desses mecanismos de SSO delimita os sistemas operacionais clientes a um único sistema operacional. Assim, não será citado ou incorporado em um escopo de projeto de SSO.

As autenticações dos usuários podem ser feitas através de uma base LDAP, sendo que o único inconveniente para o usuário é redigitar o usuário e a senha em caso de acesso pela primeira vez ou após os *cookies* expirarem.

3.2.2 Tipos de base de dados

Para validar os usuários, os sistemas autenticam, geralmente, o usuário e a senha em uma base de dados. Por padrão, muitos sistemas assumem uma base de dados própria para a autenticação, frequentemente armazenados em banco de dados relacional.

Como se está lidando com autenticação de várias aplicações e de autenticação de sistemas operacionais, terá de ser usada uma base de dados padrão.

Segundo o Id OSS Map (2006), existem três tipos gerenciamento de usuários: serviços de diretório, diretório virtual e meta diretórios.

3.2.2.1 Serviços de diretório

Os serviços de diretórios são diretórios físicos que armazenam objetos, como usuários e computadores e seus respectivos atributos. Esses diretórios usam o protocolo *Lightweight Directory Access Protocol* (LDAP). Segundo Wikipedia (2006), o diretório LDAP segue, geralmente, o padrão X.500, que é uma árvore de nós, em que cada um consiste em um conjunto de atributos com seus respectivos valores.

Segundo Id OSS Map (2006), entre os diretórios padrões *opensource*, os mais conhecidos e utilizados no mundo de código livre são:

a) Fedora DS (389 Directory Server): é um servidor LDAP patrocinado pela empresa do mundo *opensource* Red Hat e baseado no Netscape DS;
b) OpenLDAP: é uma implementação *opensource* do protocolo LDAP e tem como mantenedora a OpenLDAP Foundation;
c) ApacheDS: esse servidor LDAP é uma implementação em Java do LDAPv3 que é mantido pelo Apache Foundation; e
d) OpenDS: servidor LDAP baseado em Java, que tem como principal patrocinadora a Sun Microsystems. Esse servidor de diretório segue os padrões LDAP e *Directory Service Markup Language* (DSML).

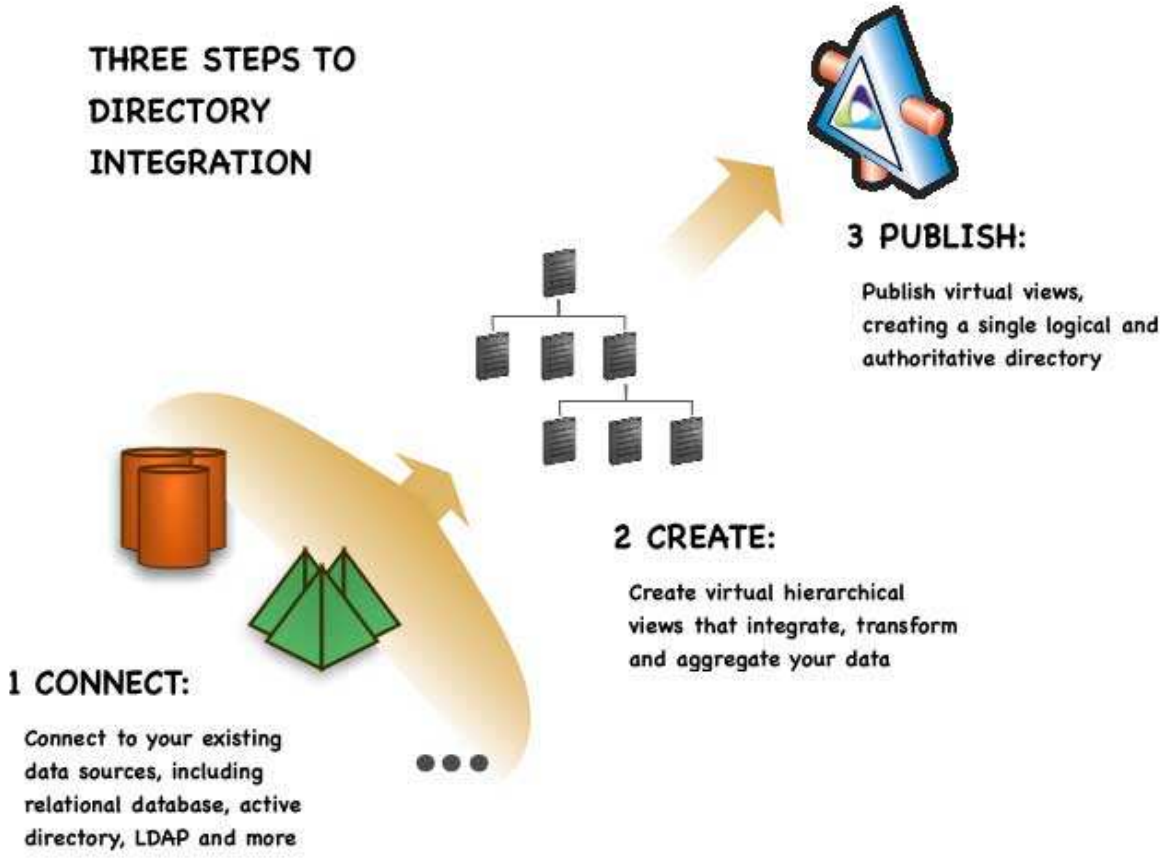
Todos os serviços de diretórios LDAP supracitados podem ser utilizados na autenticação de usuários no modelo de SSO proposto neste trabalho, ficando a cargo do coordenador, gerente ou responsável pelo projeto a escolha da ferramenta.

3.2.2.2 Diretório virtual

Segundo o Wikipedia (2009), diretório virtual é uma tecnologia que oferece uma maneira de prover uma visão consolidada de identidade de usuários sem ter que construir uma infraestrutura inteira de diretório. O diretório virtual é implementado na forma de *middleware*, em que esse *middleware* opera entre a aplicação e o repositório de identidade.

Usando o protocolo padrão LDAP, o diretório virtual recebe as requisições e as direciona para a fonte de dados apropriada. Com o direcionamento de fonte de

dados, o diretório virtual é uma boa opção para ambientes distribuídos com várias bases de dados em diversos lugares diferentes. Na Figura 4, é possível ver a integração utilizando a ferramenta Penrose:



Fonte: Safehaus, 2009.

Figura 4: Arquitetura Penrose

Para realizar essa integração, formando uma única visão para a aplicação, segundo o Id OSS Map (2006), existem dois tipos de diretório virtual no chamado mundo *opensource*, que são:

a) Penrose: é um serviço de virtualização de dados de identidade para uma coleção de diretórios de rede e servidores de bancos de dados. O *Penrose* utiliza o ApacheDS, sendo que o OpenLDAP e o FedoraDS (389 *Directory Server*) também são suportados. Essa ferramenta é patrocinada pela Safehaus; e

b) MyVD: é uma ferramenta *opensource* que implementa um diretório virtual, possibilitando a integração de dados de identidade.

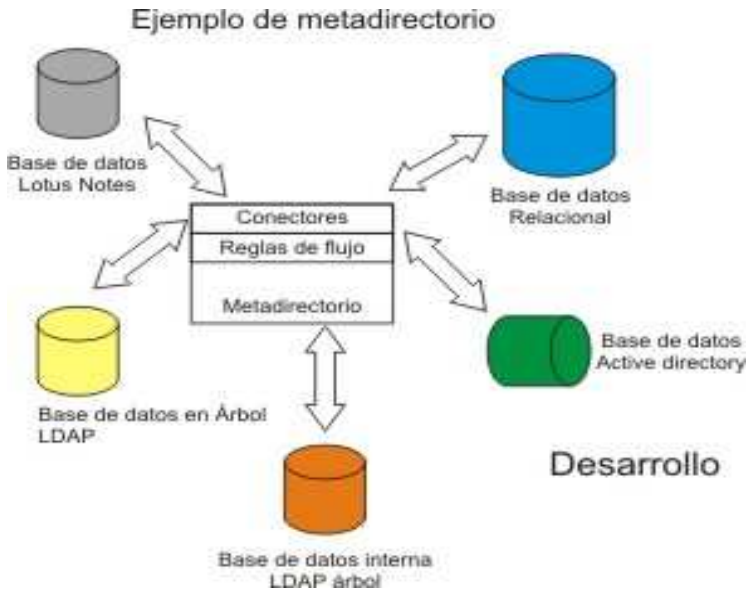
Pelo fato de se aplicar o padrão LDAP, é possível também se encaixar no escopo de um projeto de SSO. A utilização deste depende da forma como é implementado o armazenamento de informações do usuário.

3.2.2.3 Meta diretórios

Segundo o Wikipedia (2009), um sistema de meta diretório provê sincronização de dados para o fluxo entre um ou mais servidores de diretório e banco de dados. Muitos meta diretórios fazem a sincronização de pelo menos um servidor LDAP de uma fonte não LDAP para que o acesso de um serviço de SSO ou portal, por exemplo, tenha os dados dos usuários atualizados.

Os meta diretórios são muito importantes para ambientes em que se deseja implantar o conceito de SSO e em que haja muitas bases que não são LDAP. O meta diretório, nesse caso, torna-se imprescindível.

O que difere meta diretórios e diretórios virtuais é que o meta diretório faz a sincronização dos dados e, ao se criar um usuário em qualquer base onde o meta diretório esteja abrangendo, este usuário passará a outras bases, enquanto o diretório virtual não sincroniza bases; ele apenas busca os dados na fonte onde retém esses dados. Na Figura 5, é possível verificar a arquitetura de um meta diretório.



Fonte:Wikimedia, 2009.

Figura 5: Exemplo de meta diretório

Segundo o Id OSS Map da Safehaus (2006), os tipos de meta diretórios *opensource* são:

a) Ganymede: esse meta diretório é mantido pela ARL The Univesity of Texas at Austin e gerencia base de dados Oracle, PostgreSQL, MySQL e autencicação NIS e LDAP; e
b) Slapd Back-Meta: faz o *proxy* básico de um LDAP Server com um LDAP remoto.

Devido à falta de documentação encontrada sobre as ferramentas Slapd Back-Meta, recomenda-se o uso do Ganymede, pois há documentações de instalação, e é relacionado em alguns *sítios*, como Wikipedia e o mapa da Safehaus.

3.2.3 Tipos de linguagens de programação utilizadas

Com o avanço tecnológico, as empresas começaram a usufruir de pequenos aplicativos que realizavam tarefas simples em que se entrava com dados, o sistema os computava e, então, surgiam os resultados.

Para que esses pequenos softwares pudessem ser rodados, eles tinham que ser programados, através de instruções, para que um computador soubesse o que fazer e como processar.

Com o avanço da computação, surgiram as linguagens chamadas de alto nível, ou as linguagens de programação. Um programador não necessitava mais programar um software em instruções de máquina. Ele poderia programar uma linguagem com melhor reconhecimento humano, e um compilador seria necessário para traduzir essa linguagem para as instruções.

Segundo o Wikipedia (2009), uma linguagem de programação é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Uma linguagem que permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como esses dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

Vemos atualmente no mercado uma gama de linguagens de programação, para atender vários tipos de demanda. As aplicações, no início, eram voltadas a

servidores; posteriormente, foram migradas para os *Personal Computer* (PC); e, atualmente, com o avanço da internet, foram migradas para a internet.

Para que as aplicações pudessem evoluir a esse ponto, as linguagens de programação tiveram que evoluir também, para que pudessem atender estas novas demandas.

Como o SSO é apenas utilizado apenas em aplicações web, há uma delimitação de escopo e levantamento de linguagens e aplicações voltadas à web.

Para que se possa arquitetar e implantar um projeto de SSO, devem-se verificar alguns tópicos, dos quais o levantamento das aplicações e das linguagens usadas nessas aplicações para saber os tipos de ferramentas necessárias.

Neste trabalho, caso haja linguagens não relacionadas, deve-se verificar as ferramentas de SSO para conferir se há suporte a essas linguagens ou verificar se a linguagem possui algum suporte a XML.

A seguir, abordar-se-ão algumas linguagens.

3.2.3.1 Java

A linguagem Java é, hoje, amplamente usada na programação de aplicações complexas. Muitas ferramentas são desenvolvidas sobre essa linguagem ou para essa linguagem, cujo desenvolvimento é voltado para sistemas *desktops*, celulares e web.

Segundo o Wikipedia (2009), Java é uma linguagem de programação originalmente desenvolvida por James Gosling na Sun Microsystems e foi lançada, em 1995, como um componente principal para o Sun Microsystems Java Platform. A linguagem foi derivada das linguagens C e C++, mas é uma linguagem orientada a objeto. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um *bytecode*, que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não é sua única linguagem.

Ainda de acordo com Wikipedia (2009), em 1997 a Sun *Microsystems* tentou submeter a linguagem à padronização pelos órgãos ISO/IEC e ECMA, mas acabou desistindo. Java ainda é um *standard* de fato e é controlada através da *Java Community Process* (JCP). Em 13 de novembro de 2006, a Sun lançou a maior

parte do Java como Software Livre sob os termos da GNU General Public License (GPL). Em oito de maio de 2007, a Sun finalizou o processo, tornando praticamente todo o código Java como software de código aberto, menos uma pequena porção da qual a Sun não possui *copyright*.

Java se difere de outras linguagens por seu compilador gerar *bytecode* e ser executado por uma máquina virtual: *Java Virtual Machine*. Com isso, a Sun Microsystems desenvolveu essa máquina virtual para as diferentes plataformas existentes no mercado. Assim, um programador não precisa reprogramar uma aplicação para adaptar para um sistema operacional específico, pois, como é a mesma máquina virtual, esta se encarrega de traduzir o código para o sistema operacional hospedeiro.

Para aplicações de celulares, existe o Java ME, que detém menos funções e, desse modo, usa menos recursos para que possam rodar em um *hardware* mais limitado. Também há o Java *Runtime*, que é a máquina virtual instalado em *desktops* para rodar programas em Java e *applets* de alguns sítios de internet. E por último o Java EE, que é o Java que roda em servidores, em que se hospedam sítios através de aplicações como tomcat e jboss.

Na linguagem Java, no que tange à web, são utilizados alguns formatos na apresentação de suas páginas web. Essa tecnologia Java denomina-se JavaServer Pages (JSP).

Conforme o Wikipedia (2009), JSP é uma tecnologia Java alocada ao lado do servidor e permite aos desenvolvedores de software criar páginas web que são geradas dinamicamente, com HTML, XML ou outros tipos de documentos, em resposta a uma solicitação do cliente da web para um recipiente de Aplicação Java Web (servidor). Para permitir isso, uma página HTML é dada pela extensão do arquivo .jsp e uma página de marcação XML é dada pela extensão do arquivo.Jspx para que o servidor Java (*container*) reconheça que o arquivo JSP requer processamento antes de enviá-lo ao cliente. Páginas JSP são carregadas no servidor e operado a partir de um servidor de pacotes especiais instalados chamado de Java J2EE *Web Application*, muitas vezes empacotado em um arquivo .war. ou .ear.

Essa tecnologia, ainda de acordo com Wikipedia (2009), permite que o código Java e determinadas ações pré-definidas sejam incorporadas no conteúdo da página estática e compilada no servidor em tempo de execução de cada solicitação

de página. Tanto o servidor de programação Java (especificação J2EE) e quanto os *scripts* de páginas e/ou programação são operados (no contexto de tempo de execução) por um programa base especial pré-instalado chamado de máquina virtual que se integra ao sistema operacional do servidor. Este tipo é o Java *Virtual Machine* (JVM).

No mundo *opensource*, principalmente no mapa Id OSS Map, a linguagem Java é a mais aceita e suportada pelas ferramentas de SSO, inclusive algumas ferramentas são programadas na linguagem Java. Para o funcionamento dessa linguagem, dentro de um projeto SSO, existem ferramentas que disponibilizam alguns métodos para serem inseridos dentro do código que possibilita usufruir da autenticação baseada em SSO.

3.2.3.2 Personal Home Page

Com o surgimento das primeiras páginas Web, surgiram as primeiras versões da linguagem HTTP. Com o crescimento da popularização da internet, mais especificamente da web, iniciou-se uma demanda maior de funções para os *sítios* que surgiam. Como o HTTP é uma linguagem limitada, foram surgindo algumas linguagens para suprir essas necessidades de mais funções. Então, criou-se o *Personal Home Page* (PHP) para atender às novas necessidades e ampliar a facilidade na construção de *websítios*.

Segundo Wikipedia (2009), o PHP foi criado originalmente por Rasmus Lerdorf, em 1995, e está em desenvolvimento contínuo desde então. A aplicação principal do PHP agora é produzida pelo grupo PHP e serve como o padrão de fato para o PHP, pois não há especificação formal. PHP é um software livre, liberado sob a Licença do PHP, que é incompatível com a GNU *General Public License* (GPL) por causa de restrições à utilização do termo PHP.

O PHP tem como propósito *scripts* para páginas web, tornando-as dinâmicas através de chamadas de funções e variáveis. Para executar o código PHP, é necessária a utilização de um servidor web, no qual se hospeda a página .php, em que estão embutido os códigos dentro de uma página HTML. O servidor interpreta e traduz os códigos PHP para gerar a exibição dos resultados de funções para os usuários.

Como foi uma das primeiras linguagens a ser lançada, logo após o uso em larga escala do HTTP, e por ter fácil implementação, o PHP teve uma disseminação grande. Existem ferramentas dentro do Id OSS Map que tem suporte a essa linguagem para que seja possível a integração de autenticação SSO.

3.2.3.3 Perl

Em razão da necessidade de uma programação para auxílio em servidores baseados em UNIX, Larry Wall, um administrador de sistemas da NASA, criou o Perl para desenvolver relatórios mais facilmente.

Segundo o Wikipedia (2009), Perl é uma linguagem de programação estável e multiplataforma, usada em aplicações de missão crítica em todos os setores, sendo destacado o seu uso no desenvolvimento de aplicações web de todos os tipos. Permite a criação de programas em ambientes UNIX, MSDOS, Windows, Macintosh, OS/2 e outros sistemas operacionais. Além de ser muito utilizada para programação de formulários www e em tarefas administrativas de sistemas UNIX - onde a linguagem nasceu e se desenvolveu - possui funções muito eficientes para manipulação de textos.

Ainda de acordo com o Wikipedia (2009), Perl é uma das linguagens preferidas por administradores de sistema e é especialmente versátil no processamento de cadeias (*strings*), na manipulação de texto e no *pattern matching* implementado através de expressões regulares, além de ser bastante adequada para o desenvolvimento de projetos que utiliza uma metodologia ágil. A linguagem Perl já foi portada para mais de 100 diferentes plataformas e é bastante usada em desenvolvimento web, finanças e bioinformática.

3.2.3.4 Active Server Page

Seguindo as tendências de evolução da web, a empresa Microsoft, a qual detinha a linguagem *Visual Basic* já existente e utilizada no mercado, construiu uma estrutura de programação para que fosse possível desenvolver *scripts* e páginas web dinâmicas.

Baseado nas ferramentas dbWeb e iBasic, o *Active Server Page* (ASP), conhecida também como ASP clássico, é o primeiro motor de *script* desenvolvido pela Microsoft, que roda do lado do sevidor, para gerar páginas web dinamicamente. Inicialmente foi disponibilizado como um *add-on* para o *Internet Information Services* (IIS), através do Windows NT 4.0 Option Pack, e depois foi incluído como um componente grátis do Windows Server (inicialmente no Windows Server 2000).

O desenvolvimento de funcionalidades em *sítios* ASP é obtido através do motor de *scripts* ativo do *Component Object Model* (COM), em que cada objeto fornece um grupo relacionado de funções usadas com frequência e atributos de dados. No ASP 2.0, há seis objetos internos: *Application*, *ASPError*, *Request*, *Response*, *Server* e *Session*. Este, por exemplo, é um objeto baseado em *cookies* de sessão que mantém o estado das variáveis de página para página. A funcionalidade é prorrogada por objetos que, quando instanciada, proporciona o acesso ao ambiente do servidor web. Por exemplo, o *FileSystemObject* (FSO) é usado para criar, ler, atualizar e apagar arquivos (WIDIPEDIA, 2009).

As páginas da web com a extensão .asp usam ASP, embora alguns sítios disfarcem a sua escolha de linguagem de *script* para fins de segurança (por exemplo, continuam a utilizar a mais comum .htm ou .html). Páginas com a extensão .aspx são ASP.NET (baseado no *framework* .NET da Microsoft) e compilados, o que as tornam mais rápidas e mais robustas do que em ASP, que é interpretado em tempo de execução. Entretanto, muitas páginas ASP.NET ainda incluem alguns *scripts* ASP. Tais diferenças acentuadas entre ASP e ASP.NET levaram os termos ASP clássico a ser utilizado, o que também implica uma certa nostalgia da mais simples plataforma.

Apesar de ASP ser uma tecnologia proprietária da Microsoft e, portanto, não ser uma opção de código aberto, essa estrutura de programação é amplamente utilizada pelo mercado e está inclusa no mapeamento de linguagens, pois as ferramentas de SSO que constam no mapa Id OSS Map suportam esse tipo de estrutura.

3.2.3.5 NET Framework

Com o início do uso do ASP através dos componentes do Windows Server, Mark Anders, em 1997, após fazer parte da equipe da no Microsoft, entrou para o time de gerentes do IIS e, juntamente com Scott Guthrie, criou o então chamado XSP, que posteriormente veio a tornar-se o ASP.NET.

De acordo com o sítio Wikipedia (2009), ASP.NET é um *framework* para aplicação web, desenvolvido e comercializado pela Microsoft, e possibilita aos programadores construir páginas web dinâmicas, aplicações web e *webservices*. O ASP.NET foi disponibilizado pela primeira vez em janeiro de 2002 com a versão 1.0 do .NET Framework e é o sucessor do ASP.

O Microsoft .NET Framework é uma estrutura de software que pode ser instalado em computadores que executam sistemas operacionais Microsoft Windows. Ele inclui uma grande biblioteca de soluções codificadas para problemas de programação comum e uma máquina virtual que gerencia a execução de programas escritos especificamente para o *framework*. .NET Framework é uma oferta da Microsoft e se destina a ser utilizado pela maioria das novas aplicações criadas para a plataforma Windows (WIKIPEDIA, 2009).

Conforme o mesmo sítio, o *framework's Base Class Library* fornece uma ampla gama de recursos, incluindo interface com o usuário, de dados e acesso a dados, conectividade do banco de dados, criptografia, desenvolvimento de aplicações web, algoritmos numéricos e comunicações de rede. A biblioteca de classes é usada por programadores que combina com seu próprio código para produzir aplicações.

Os programas escritos para o .NET Framework executam em um ambiente de software que gerencia os requisitos de *runtime* do programa. Também como parte do *framework* .NET, esse ambiente de execução é conhecido como o *Common Language Runtime* (CLR), que fornece a aparência de uma aplicação da máquina virtual, assim como no Java, para que os programadores não precisem considerar as capacidades do processador específico que irá executar o programa. O CLR também oferece outros serviços importantes, como segurança, gerenciamento de memória e manipulação de exceção. A biblioteca de classes e CLR, juntos, constituem o .NET Framework.

Dentro do .NET Framework é possível se escrever aplicações em linguagens, como: C##, VB.NET e J##.

Assim como o ASP, o .NET Framework, incluindo a plataforma ASP.NET, é software proprietário, mas, como há uso em larga escala, por optarem por tecnologias proprietárias, o ASP.NET também é suportado pelas ferramentas *opensource* de SSO.

3.3 IMPLANTAÇÃO DE SINGLE SIGN-ON

Dentro de um escopo de um projeto de SSO, a escolha de ferramentas é imprescindível para o sucesso de uma boa implantação de SSO dentro de uma organização. Visto as ferramentas supracitadas, iniciar-se-á o mapeamento para relatar, neste trabalho, o apoio de decisão para o projetista.

A partir desse subtítulo, será descrito como realizar os passos para identificar as ferramentas necessárias para a integração e o funcionamento do conceito de SSO.

Na implantação de SSO, deve-se separar em três tópicos para se definir um projeto de SSO: autenticação, levantamento de linguagens e escolhas de ferramentas SSO.

3.3.1 Autenticação

Como visto anteriormente, através da demonstração do mapa Id OSS Map, o que se precisa para que os usuários sejam autenticados é de um local no qual sejam armazenados esses usuários. Assim, tem-se um ponto único de identificação de usuário para que as ferramentas de integração de aplicação com base de usuários possam fazer a identificação dos usuários e fornecê-la para a aplicação.

Conforme já abordado, há três tipos de gerenciamento de usuários: diretório virtual, meta diretório e serviço de diretório, em que cada um deles visa prover uma base de usuários na qual é possível fazer verificação de usuário e senha, por exemplo. A escolha de qual ou quais tipos será usado depende do ambiente de tecnologia da informação da organização em que será implantado o projeto SSO.

3.3.1.1 Diretório virtual

Existem algumas organizações que possuem aplicações já implantadas. Para realizar autenticação e/ou buscas de dados de usuários, essas aplicações geralmente fazem consultas SQL em bancos de dados relacionais. No sentido de integrar as autenticações com alguma base de usuários e utilizar tecnologias suportadas para as ferramentas de SSO, é preciso trabalhar essa camada de banco de dados, migrando a autenticação para uma base de usuários, mas mantendo as informações necessárias em um banco de dados relacional, em que esses dados não têm como serem inseridos por não haver suporte em um serviço de diretório.

O diretório virtual, na verdade, não é uma única base de dados de usuário, mas sim uma única visão de informações para a aplicação, na qual essa visão busca os dados através de bancos de dados relacionais e/ou serviços de diretórios. Dessa forma, resolve-se o problema da utilização de uma base legada de dados de usuários ou utilização de atributos em que é possível apenas armazenar em um banco de dados relacional.

Para que seja necessário desenhar uma implantação de um projeto de SSO utilizando essa tecnologia, a empresa deve precisar de bancos de dados relacional com atributos de usuários e um de um serviço de diretório LDAP, e este pode ser um novo servidor ou um já existente do ambiente. Sempre há a possibilidade de verificar se é possível utilizar a estrutura de um serviço de diretórios LDAP ou importar os dados para dentro desse serviço, utilizando, assim, menos ferramentas e, consequentemente, resultando em melhor administração da infraestrutura de Tecnologia da Informação (TI).

Com uma ferramenta de diretório virtual, será possível que as ferramentas de SSO façam consultas LDAP através de uma visão global disponibilizada pelo diretório virtual, e o diretório virtual irá buscar as informações nas fontes em que se integra. Assim, é possível ter suporte a quase todas, se não a todas, às ferramentas de SSO e, com isso prover também todas as informações necessárias para as aplicações utilizando-se de um ponto único de consulta.

Na implantação do diretório virtual, recomenda-se o uso da tecnologia chamada Penrose, que tem como mantenedora Safehaus, pois, dentro do mapa Id OSS Map, é a única que se mantém atualizada e contém documentações suficientes

para a implantação e a configuração dentro do ambiente desejado em uma área de TI de uma organização.

3.3.1.2 Meta diretório

Através de escolhas de coordenadores e gerentes de TI, veem-se ambientes das mais variadas peculiaridades dentro das empresas. Assim, sabe-se que nunca há sempre um ponto único de dados, que seria o ideal. Há a possibilidade de uso de base de dados relacional e, pelo menos, uma base de usuários LDAP para que seja configurado um meta diretório.

Nessa tecnologia, é fornecida uma autenticação LDAP para a ferramenta SSO que provê também sincronização das bases, seja ela em um banco de dados relacional ou em um servidor de diretórios. Essa sincronização não acontece em tempo real se um usuário for adicionado, ou alguma informação for atualizada, pois levará um tempo, o chamado *delay*, para que os dados se sincronizem.

Em razão de alguns requisitos de aplicações, é possível que o meta diretório não seja possível de ser implantado dentro de um projeto de SSO, pois, como seu tempo de replicação não é em tempo real, em algumas aplicações, como um sistema CRM, por exemplo, o sistema mostrará ao cliente ou a um funcionário da organização informações desatualizadas, gerando, assim, um transtorno para ambos os lados. Ao fazer o levantamento dos tipos de aplicações e visto se é de missão crítica ou não, é possível verificar a possibilidade do uso dessa tecnologia no auxílio de SSO.

Seguindo o guia optado nesse trabalho para a escolha de ferramentas, Id OSS Map, sabe-se que há duas ferramentas, o Ganymede e o Slapd Back-meta. Por não encontrar documentações da ferramenta Slapd Back-meta, recomenda-se o uso apenas do Ganymede para a implantação de um meta diretório.

3.3.1.3 Serviço de diretório

Em um ambiente computacional, envolvendo centenas e até milhares de computadores em uma rede computacional corporativa, há a necessidade, por parte

da empresa, do uso de ferramentas que auxiliem o setor, ou departamento, de TI para o gerenciamento de usuários.

Entre as tecnologias de tipos de base de usuários mostrada, essa, com certeza, é a mais utilizada dentro das corporações, não pelo motivo de prover autenticação aos sistemas, mas sim para que sistemas operacionais possam autenticar nessa base de serviço de diretório.

Todas as ferramentas SSO apresentadas anteriormente neste trabalho são suportadas ao serviço de diretório LDAP, por ser um padrão comum e vastamente utilizado em uma rede corporativa.

O motivo para o uso tão comum dessa tecnologia de autenticação de usuários em sistemas operacionais é pelo fato de que, com um serviço de diretório LDAP implantado, é factível que se tenha um ponto único de autenticação, fazendo com que, independentemente do computador em que o usuário esteja, seja possível que o usuário faça a sua autenticação com o mesmo usuário e senha em qualquer computador da rede corporativa na empresa onde se encontra.

Existe, no mercado, uma grande quantidade de ferramentas, proprietárias ou *opensource*, que proveem serviços de diretórios através de um protocolo padrão LDAP. Assim, todos os sistemas operacionais, com suporte a LDAP, podem se autenticar nessa base, validando o usuário e a senha passada pela pessoa que está usando aquele equipamento. No mundo *opensource*, existem várias ferramentas de serviço de diretório, das quais as mais comuns encontradas em organizações que optaram por ferramentas *opensource* são OpenLDAP e Fedora-DS.

Devido ao sistema operacional Microsoft Windows ser um produto da empresa Microsoft e de essa empresa ter um produto lançado no mercado que provê serviço de diretório através da família Microsoft Windows Server, denominado *Active Directory*, a plataforma Microsoft Windows só faz autenticação em produtos Microsoft. Para que um cliente Windows possa se autenticar em uma base LDAP que não seja proveniente da empresa Microsoft, é necessário o uso e a integração com uma ferramenta chamada de Samba. Essa ferramenta, também de código aberto, é que faz a integração entre uma base LDAP e uma plataforma Windows, disponibilizando para um cliente Windows uma autenticação para os usuários dos equipamentos.

Para uma implantação de um projeto de SSO, essa tecnologia é imprescindível para o sucesso deste projeto, pois essa tecnologia é a base de tudo

para que as ferramentas SSO possam se autenticar e prover *login* único às aplicações.

3.3.2 Levantamento de linguagens

Atualmente, há muitas aplicações que ajudam na tomada de decisão, na operação e no atendimento ao cliente dentro das empresas. No mercado, existem diversas empresas que oferecem softwares prontos ou desenvolvimento para suprir as necessidades das organizações.

Nessa etapa, o coordenador ou o gerente de TI ou ainda a pessoa que será responsável pelo projeto SSO, podendo ser um terceiro contratado através de outra empresa para esse projeto, deverá mapear todas as aplicações existentes que auxiliam no processo dentro da empresa.

Com o levantamento feito dos sistemas que operam dentro da organização, deverá ser feito a separação das aplicações em dois tipos: aplicação web e aplicação não web. Essa divisão é feita para separar as aplicações web, as que têm suporte às ferramentas SSO, das aplicações não web, em que não há suporte pelas ferramentas SSO.

Nas aplicações em que as ferramentas SSO dão suporte à integração com *login* único, deve-se verificar entre as aplicações restantes quais sistemas são possíveis se ter acesso ao código fonte, ou seja, onde é possível se customizar e quais são as aplicações fechadas em que não é possível fazer nenhum tipo de alteração. Mesmos nas aplicações fechadas, deve-se verificar junto à empresa que desenvolveu essa aplicação se o sistema não possui suporte à implantação e/ou customização para habilitar a funcionalidade de autenticação através de uma ferramenta SSO.

Com os sistemas já filtrados para suporte à funcionalidade de SSO, deve-se verificar, por último, quais as linguagens escritas nos sistemas. Nas ferramentas descritas dentro do mapa Id OSS Map, é possível verificar que as ferramentas abrangem quase todas, se não todas, as linguagens para aplicações web, sendo que as mais utilizadas no mercado estão na alçada dessas ferramentas SSO e, portanto, não será um entrave para o coordenador de projeto de SSO.

Juntamente com a linguagem, deve-se mapear os servidores de aplicação para que seja possível escolher as ferramentas de SSO em razão dos pré-requisitos de algumas dessas ferramentas.

3.3.3 Escolha de ferramentas SSO

Devido à grande gama de ferramentas SSO, deve-se escolher quais ferramentas usar para atender às exigências do ambiente de TI da empresa. Para isso, serão utilizadas informações de autenticação conforme o tipo de diretório utilizado dentro da organização e será usado também as informações do levantamento de linguagens feita pelo coordenador do projeto de SSO.

As informações principais que são necessárias para a tomada de decisão de um projetista de uma implantação de *login* único através de SSO é o levantamento das linguagens utilizadas nas aplicações que auxiliam as operações da empresa. Esse levantamento faz-se necessário para a escolha, pois algumas ferramentas de SSO só se integram com algumas linguagens e, com isso, dependendo do número de aplicações e de linguagens utilizadas nessas aplicações, tem de usar mais de uma ferramenta SSO.

Por causa da maior abrangência de algumas linguagens do que outras, pelas ferramentas de SSO, recomenda-se iniciar a escolha das ferramentas de SSO pelas linguagens em que há menor abrangência pelas ferramentas.

A seguir, serão mostradas as principais linguagens e os requisitos necessários para a implantação de um *login* único.

3.3.3.1 Java

Quando se diz o uso Java nas aplicações está se considerando o uso de J2EE que provê uso da linguagem Java para sistemas web, onde se roda através de um navegador.

Para que se tenha uma autenticação única através de SSO, é preciso de ferramentas de SSO. Muitas ferramentas possuem suporte à esta linguagem, assim esta linguagem não será um grande problema na integração com uma ferramenta

SSO. Seguindo o mapa Id OSS Map, as ferramentas que têm suporte à linguagem Java são: CAS, OpenSSO, JOSSO e Shaj. Todas essas ferramentas possuem autenticação para servidores de diretório utilizando o padrão LDAP, o que facilita a integração com um ambiente de TI de uma organização.

A escolha de ferramenta é importante em caso de haver mais de uma aplicação e estas outras aplicações serem escritas em outras linguagens. Assim, quanto menor o número de ferramentas instaladas, mais fácil fica a administração desse ambiente de SSO.

3.3.3.2 Personal Home Page

PHP é uma linguagem que já tem como pré-requisito o uso do Apache para a instalação do PHP e, desse modo, tem como vantagem atender o pré-requisito de algumas ferramentas SSO descritas no mapa Id OSS Map.

O número de ferramentas SSO que abrange esta linguagem é menor do que a do Java. Por isso, recomenda-se começar pela que tem menor abrangência para a maior. As ferramentas que tem suporte ao PHP são: CAS e JOSSO. Essas ferramentas possuem suporte ao serviço de diretório através do protocolo LDAP e, assim, fazem integração com a autenticação que é comumente usada pelo mercado.

3.3.3.3 Perl

Na utilização dessa linguagem para a escrita de uma aplicação de página web dinâmica, deve-se verificar, para um projeto de autenticação SSO, quais métodos de autenticação e base de usuários vai se utilizar, pois há pequeno número de ferramentas SSO, providas pelo mapa Id OSS Map, que tem integração com essa linguagem.

A ferramenta de SSO do mundo *opensource* que está listado no mapa Id OSS Map é CAS.

Devido ao suporte de apenas uma ferramenta *opensource* para integração, deve-se tomar cuidado com o uso dessa linguagem na integração de uma

autenticação SSO não pelo desempenho, pela qualidade ou pela robustez, mas sim pelo suporte das ferramentas.

3.3.3.4 Active Server Page

Pelo fato de ASP ser uma linguagem proprietário e ser investida por uma empresa de grande porte como a Microsoft, o uso dessa linguagem é feito em grande escala pelo mercado e, com isso, o suporte a essa ferramenta é importante para que não gere nenhum bloqueio ao projeto de SSO, por ser uma linguagem mais obsoleta e por já ter um sucessor e com tendência de entrar em *end of life*.

A única ferramenta que faz integração com o ASP para uma autenticação única é JOSSO.

Vê-se, pelos suportes das ferramentas, que há uma incompatibilidade do ASP com a linguagem Perl, pois não existem ferramentas em comum que deem suporte ao ASP e Perl.

Para que seja possível fazer a autenticação SSO em um ambiente no qual haja as linguagens Perl e ASP simultaneamente, é preciso, obrigatoriamente, a implantação das ferramentas CAS e JOSSO.

3.3.3.5 .NET Framework

Por ser sucessor ao ASP, o .NET tem suporte nas ferramentas mais novas e está na lista como linguagens suportadas para a autenticação SSO. Uma coisa que se deve considerar em algumas ferramentas SSO é que o suporte a esta linguagem não garante, obrigatoriamente, suporte ao seu antecessor ASP.

As ferramentas SSO que dão suporte ao .NET framework são CAS e JOSSO.

4 CONCLUSÃO

Neste trabalho, foi possível analisar os benefícios da implantação de um sistema SSO, que são trazidos a uma organização através de ferramentas que possuem código aberto.

Além de a empresa resolver inúmeros problemas com a administração de sistemas com diferentes bases de autenticação, por meio da implantação de SSO com ferramentas de código aberto, é possível implantar essa tecnologia com custos menores, sem precisar fazer a aquisição dessas ferramentas.

Verificando-se que existem várias ferramentas e vários conceitos de gerenciamento de identidade, este trabalho focou no apoio ao gestor do projeto de implantação de SSO ou do gestor da área de TI de uma organização para a escolha de quais ferramentas serão necessárias à implantação da tecnologia de SSO, integrando, assim, as credencias das aplicações web.

Com as diversas áreas de gerenciamento de identidade e até com o aprofundamento da tecnologia SSO, é possível realizar trabalhos futuros em autenticação através de interorganização, no qual é possível realizar a autenticação utilizando bases de usuários em lugares de terceiros por meio do SAML.

REFERÊNCIA BIBLIOGRÁFICA

.NET Framework. Disponível em: <http://en.wikipedia.org/wiki/.NET_Framework>. Acesso em: 14 set de 2009.

389 Directory Server (Open Source LDAP). Disponível em: <http://directory.fedoraproject.org>. Acesso em: 02 set de 2009.

ACTIVE Server Pages. Disponível em: <http://en.wikipedia.org/wiki/Active_Server_Pages>. Acesso em: 12 set de 2009.

ASP. Disponível em: http://pt.wikipedia.org/wiki/ASP. Acesso em: 12 set de 2009.

ASP.NET. Disponível em: <http://en.wikipedia.org/wiki/ASP.NET>. Acesso em: 12 set de 2009.

ASP.NET. Disponível em: <http://pt.wikipedia.org/wiki/ASP.NET>. Acesso em: 12 set de 2009.

CAS | Jasig Community. Disponível em: <http://www.jasig.org/cas>. Acesso em: 20 set de 2009.

CENQUA OS: Shaj. Disponível em: <http://opensource.cenqua.com/shaj>. Acesso em: 07 set de 2009.

CUSTOMER Relationship Management. Disponível em: <http://en.wikipedia.org/wiki/Customer_relationship_management>. Acesso em: 05 set de 2009.

DSML. Disponível em: <http://en.wikipedia.org/wiki/DSML>. Acesso em: 02 set de 2009.

HIGGINS Home. Disponível em: <http://www.eclipse.org/higgins>. Acesso em: 29 set de 2009.

HOME - Safehaus. Disponível em: <http://docs.safehaus.org/display/PENROSE/Home>. Acesso em: 16 set de 2009.

IDENTITY Management – Wikipedia, La enciclopédia libre. Disponível em: <http://es.wikipedia.org/wiki/Identity_Management>. Acesso em: 22 set de 2009.

IDENTITY Management OSS Map. Disponível em: <http://docs.safehaus.org/display/HAUS/Id+OSS+Map>. Acesso em: 22 jun de 2009.

IDENTITY management: the expert view | 23 Jul 2007 | ComputerWeekly.com. Disponível em: <http://www.computerweekly.com/Articles/2007/07/23/225715/identity-management-the-expert-view.htm>. Acesso em: 22 set de 2009.

IT Services: Stanford WebAuth. Disponível em: <http://webauth.stanford.edu>. Acesso em: 07 set de 2009.

JAVA. Disponível em: <http://en.wikipedia.org/wiki/Java_(programming_language)>. Acesso em: 09 set de 2009.

JAVA. Disponível em: <http://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)>. Acesso em: 09 set de 2009.

JAVASERVER Page. Disponível em: <http://en.wikipedia.org/wiki/JavaServer_Pages>. Acesso em: 10 set de 2009.

JØSANG, Audun; POPE, Simon. **User centric identity management**. Austrália: The University Quennsland, 2005

KERBEROS. Disponível em:< http://pt.wikipedia.org/wiki/Kerberos>. Acesso em: 22 jun de 2009.

LDAP. Disponível em:< http://pt.wikipedia.org/wiki/LDAP>. Acesso em: 31 ago de 2009.

LINGUAGEM de programação. Disponível em: <http://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o>. Acesso em: 09 set de 2009.

LOI, Leandro Nascimento. **UM ESTUDO DAS METODOLOGIAS OPEN SOURCE IDENTITY MANAGEMENT E INDICAÇÃO DA MELHOR A SER IMPLANTADA NO PROJETO VIA DIGITAL**. Florianópolis, 2007.

METADIRECTORIO. Disponível em: <http://es.wikipedia.org/wiki/Metadirectorio>. Acesso em: 17 set de 2009.

METADIRECTORY. Disponível em: <http://en.wikipedia.org/wiki/Metadirectory>. Acesso em: 02 set de 2009.

OASIS Security Services (SAML) TC. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#overview>. Acesso em: 07 set de 2009.

OPENDS: **the Open Source Java LDAP Directory Service**. Disponível em: <http://www.openldap.org>. Acesso em: 02 set de 2009.

OPENLDAP. Disponível em: <http://www.openldap.org>. Acesso em: 02 set de 2009.

OPENSSO: Home. Disponível em: <https://opensso.dev.java.net>. Acesso em: 06 set de 2009.

PERL. **Wikipedia, a enciclopédia livre**. Disponível: <http://pt.wikipedia.org/wiki/Perl>. Acesso em: 20 set de 2009.

PERL - **Wikipedia, the free encyclopedia**. Disponível em: <http://en.wikipedia.org/wiki/Perl>. Acesso em: 20 set de 2009.

PHP. Disponível em: <http://en.wikipedia.org/wiki/PHP>. Acesso em: 10 set de 2009.

PUBCOOKIE. Disponível em: <http://www.pubcookie.org>. Acesso em: 06 set de 2009.

SAMBA - opening window to a wider world. Disponível em: <http://us4.samba.org/samba>. Acesso em: 18 set de 2009.

SAML 2.0. Disponível em: <http://en.wikipedia.org/wiki/SAML_2.0>. Acesso em: 07 set de 2009.

SAMLDiff. Disponível em: <https://spaces.internet2.edu/display/SHIB/SAMLDiffs>. Acesso em: 08 set de 2009.

SECURITY Assertion Markup Language. Disponível em: <http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language>. Acesso em: 07 set de 2009.

SERVICE Provisioning Markup Language. Disponível em: <http://en.wikipedia.org/wiki/Service_Provisioning_Markup_Language>. Acesso em: 12 out de 2009.

SINGLE Sign-On. Disponível em: <http://en.wikipedia.org/wiki/Single_sign-on>. Acesso em: 22 jun de 2009.

SOAP. Disponível em: <http://en.wikipedia.org/wiki/SOAP>. Acesso em: 07 set de 2009.

THE A-Select Authentication System. Disponível em: <http://a-select.surfnet.nl/home.html>. Acesso em: 07 set de 2009.

VIRTUAL Directory. Disponível em: <http://en.wikipedia.org/wiki/Virtual_directory>. Acesso em: 02 set de 2009.

VIRTUAL vs Meta Directory. Disponível em: <http://docs.safehaus.org/display/PENROSE/Virtual+vs+Meta+Directory>. Acesso em: 17 set de 2009.

WINDLEY, Phillip J. **Digital identity.** USA: O'Reilly Media, Inc., 2005.

Aplicação de Single Sign-on e Apoio à Tomada de Decisão Utilizando Ferramentas Opensource

Marco Antonio Ferreira Vera

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

marco1@inf.ufsc.br

Abstract. *With no single sign-on would create a nuisance for system administrators because they have to manage different databases systems that seek users' information, and an inconvenience to users, every application access, should provide their credentials for authentication. As a result, causes a disruption to the system administrator, there is always forgetting your credentials. To circumvent the problem, it is possible to use tools that provide single sign-on, which sometimes hold high cost, because they take advantage of tools that have proprietary code. It will be seen that the open-source or open source, also provides the tools to implement a single sign-on.*

Resumo. *Com a ausência de single sign-on, gera-se um transtorno para os administradores de sistemas, pois têm que gerenciar diferentes bases de dados em que os sistemas buscam as informações dos usuários, além de um inconveniente aos usuários que, a cada aplicação de acesso, deverão fornecer as suas credenciais para a autenticação. Com isso, acarreta em um transtorno para o administrador de sistemas, pois sempre há esquecimento de suas credenciais. Para contornar o problema, é possível utilizar-se de ferramentas que proveem o single sign-on, que às vezes detêm alto custo, por se usufruir de ferramentas que possuem código proprietário. Será visto que o opensource, ou código aberto, também fornece ferramentas adequadas para a implantação de um single sign-on.*

1. INTRODUÇÃO

Em virtude da constante evolução das tecnologias computacionais, verifica-se uma maior utilização dos sistemas automatizados aplicados no cotidiano das pessoas e das corporações. Iniciou-se, nas empresas, o uso de aplicativos que auxiliavam no cadastro de produtos, como, por exemplo, aqueles que continham um arquivo ou pequenos bancos de dados e armazenavam os dados dessa aplicação.

À medida que a utilização desses softwares foi crescendo, bem como o desenvolvimento da computação, começou-se a utilizar outros aplicativos agregados a estes, utilizando outras bases de dados independentes.

Com o aumento da utilização de sistemas para processos internos e externos da empresa, houve maior preocupação com a segurança dos dados que estavam armazenados nos banco de dados. Por essa razão, criou-se o conceito atualmente definido de usuário e senha. Esse conceito consiste em cada usuário do sistema ter sua própria identificação e, com isso, apenas os usuários do sistema têm acesso à aplicação. Em razão da preocupação com a segurança, ocorreram, na maioria das aplicações de processos empresariais, as regras e as permissões dentro dos softwares, sendo baseado nas permissões das identificações.

Atualmente, encontra-se uma diversidade de sistemas, nos quais se aplicam a todos os níveis da organização (operacional, tático e estratégico), como, por exemplo, aplicativos para controle de estoque, programas CRM, até BI, e para cada aplicativo há um usuário e uma senha.

Identificado esse problema, nasceu o conceito de SSO, em que o usuário final não precisaria ter vários usuários e senhas, mas apenas um, através do qual teria acesso a todos os aplicativos necessários dentro da corporação.

2. REFERENCIAL TEÓRICO

O profissional de tecnologia da informação de hoje em dia trabalha na tentativa de fornecer aos clientes disponibilidade de serviço, transações seguras e acesso a dados a partir de qualquer computador pessoal ou dispositivo conectado à internet, com o objetivo de fornecer acesso seguro.

Com o crescente número de sistemas que apoiam funcionários, gerentes, diretores e analistas dentro de uma organização, houve aumento da complexidade de gerenciar controles de acessos, gerando maior tempo, o que acarreta em custo para a área de tecnologia da informação de uma empresa.

O conceito chamado gerenciamento de identidade, do inglês Identity Management, visa à resolução desse problema. Segundo Wikipedia (2009), gerenciamento de identidade são sistemas integrados de políticas e processos organizacionais que pretende facilitar e controlar o aceso aos sistemas de informação.

Através do gerenciamento de identidade, é possível administrar os usuários para que seja possível criá-los, dando as permissões nos sistemas e fazendo com que os usuários obtenham acessos limitados aos sistemas.

Segundo o Identity Management OSS Map (2006), o gerenciamento de identidade possui até seis conceitos reunidos para a integração da identidade, com segurança e acesso, que são:

a) Gerenciamento de usuário;
b) Autenticação;
c) Gerenciamento de acesso;
d) Sistemas de identidade;
e) Provisioning; e
f) User Centric

2.1. Gerenciamento de Usuário

O gerenciamento de usuário provê um local no qual se armazenam as credencias de cada usuário dos sistemas, sendo eles sistemas operacionais ou sistemas corporativos.

2.2. Autenticação

Autenticação é o processo no qual se verifica se alguém é realmente quem diz ser. Geralmente, isso envolve um usuário e uma senha, mas podem ser incluídos ou substituídos por outros métodos, como, por exemplo, smartcard, reconhecimento de voz, retina, digital.

No caso do gerenciamento de identidade, a autenticação é o front-end para as aplicações ou os sistemas efetuarem a verificação das credenciais dos usuários, através de protocolos de comunicação que fazem a validação na base de usuários.

2.3. Gerenciamento de acesso

Conforme o Identity Management OSS Map (2006), o gerenciamento de acesso é frequentemente utilizado para descrever os mais amplos sistemas que usam a autenticação e a autorização de serviços. Estes garantem que um determinado usuário tenha permissão a um determinado sistema ou recurso de rede e quais as permissões dentro desses sistemas ou da rede.

2.4. Sistemas de identidade

Com o surgimento da web 1.0, houve a necessidade de segurança da informação disponibilizada através dela. Com isso, originou-se a autenticação através de um usuário e senha nesses sítios, nos quais são buscadas as informações e cedido o acesso às estas. O problema, porém, é que esse tipo de autenticação não provê nenhuma interoperabilidade e, assim, os usuários têm que ceder suas credenciais a cada login. Segundo Loi (2007), as principais características desse modelo são:

a) Registro local;
b) Falta de verificação;
c) Diretório centralizado;
d) Usuário e senha;
e) Não portátil; e
f) Falta de transparência.

2.5. Provisioning

Muitos sistemas existentes no mercado ou desenvolvidos por equipes de programação usam, como base para armazenamento dos usuários, bancos de dados, pois, nessas bases, existem atributos que são necessários à aplicação, por exemplo.

Desse modo, foram encontrados problemas com autenticação em uma base única, pois uma base LDAP não possui todos os atributos necessários para uma aplicação em que se utilizam atributos específicos. Assim, aplicou-se o conceito de provisioning, no qual se sincroniza a base de dados única com a base de dados da aplicação.

2.6. User Centric

De acordo com Audun Jøsang e Simon Pope (2005), User Centric Identity muda o foco do domain-centric identity management para o usuário, oferecendo grande flexibilidade em como e onde armazenar a sua identidade, gerenciando como essa identidade será usada e compartilhada com segurança e privacidade. De acordo com Loi (2007), para alguns, o termo significa hospedado no cliente, para outros significa dar ao usuário mais opções de como e onde na rede ele poderá guardar sua própria identidade.

3. APLICAÇÃO DE SINGLE SIGN-ON E APOIO À TOMADA DE DECISÃO UTILIZANDO FERRAMENTAS OPENSOURCE

Conforme o referencial teórico, sabe-se que há a possibilidade de utilizar-se de tecnologias para resolver problemas atuais de empresas que contêm os mais variados sistemas de apoio aos seus processos, tanto internos quanto externos.

Atualmente, vê-se que as organizações possuem sistemas de ERP, CRM, BI entre outros, para apoiar funcionários dos vários níveis hierárquicos. Para acessar cada sistema, geralmente o usuário tem que fornecer um login e uma senha em todas as aplicações para poder ter acesso aos recursos desejados, gerando tempo gasto nas autenticações e mais chamados ao helpdesk, perdendo tempo resolvendo problemas, como esquecimento de senhas, por exemplo.

Para tenta encontrar uma solução para contornar esse problema, diminuindo-o ou eliminando-o, será mostrada uma forma de implantar o conceito de single sign-on utilizando-se de ferramentas opensource descritas no Identity Management OSS Map.

Como se pode perceber, existem muitas tecnologias, para gerenciamento de identidade, dentro das seis áreas dadas pelo Identity Management OSS Map (2006). Para isso, há de se escolher quais ferramentas devem ser utilizadas e quais ferramentas são mais aderentes dentro da organização, analisando os seguintes quesitos:

a) Tipos de aplicações existentes;
b) Modo de autenticação já utilizado;

c) Sistema operacional das estações; e
d) Tecnologia utilizada nas aplicações.

Com a coleta dessas informações, é possível analisar quais tecnologias poderão e/ou deverão ser utilizadas para se aplicar os conceitos de gerenciamento de identidade, obtendo-se os benefícios que foram citados no referencial teórico.

Para desenvolver um escopo de projeto SSO, devem-se utilizar algumas ferramentas do gerenciamento de identidade para que o SSO integrado com algumas dessas ferramentas possa disponibilizar um serviço de login único.

As ferramentas utilizadas são de código aberto ou opensource. A seguir, apresenta-se o mapa do Id OSS Map no qual são listadas todas as ferramentas opensource para o gerenciamento de identidade.

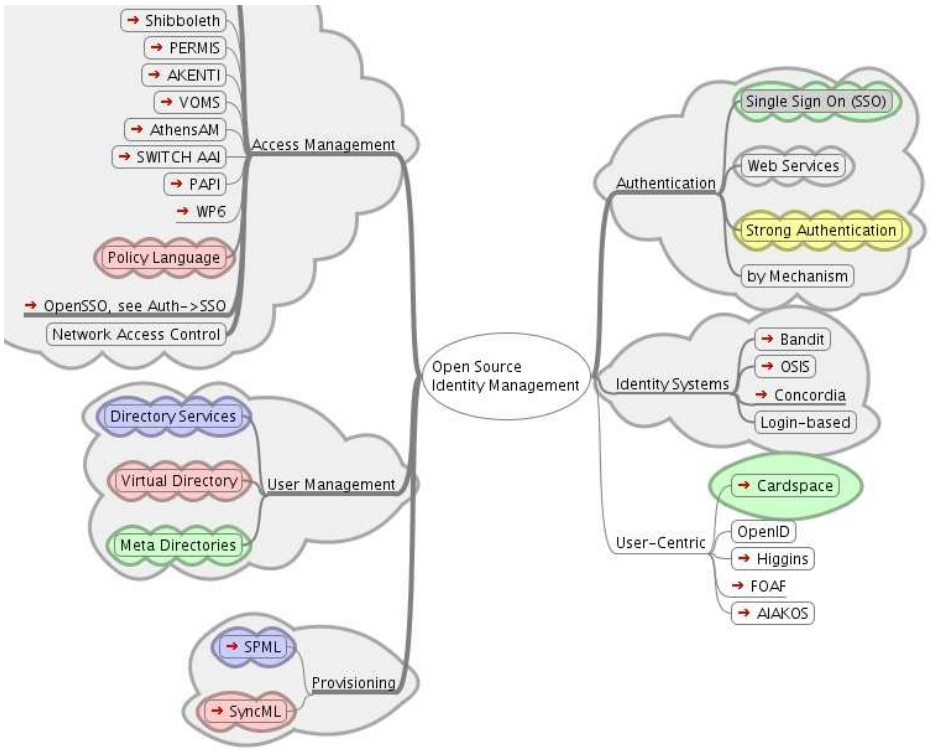


Figura 1. Mapa de gerenciamento de identidade

Dentro desse mapeamento feito pelo sítio Safehaus, as áreas utilizadas para implantar SSO serão o User Management e o Authentication, sendo que deste último utilizar-se-á uma subárea que será o SSO. Desse modo, conseguir-se-á implantar um projeto de SSO dentro de uma organização e prover autenticação única para as aplicações da empresa.

3.2. Definições e Ferramentas de Single Sign-on

Para solucionar problemas encontrados com as muitas bases de usuários, surgiu a tecnologia SSO, que visa integrar essas aplicações na autenticação e autorização.

Considerando que empresas usam vários tipos de linguagens existentes no mercado, serão mostrados os vários tipos de linguagens e como estas podem se integrar com o tipo de autenticação de SSO.

Para que uma linguagem possa fazer a autenticação do usuário, é exigido que o usuário forneça dados que são pertinentes a usuário que está tentando acessar, geralmente um usuário e uma senha. Assim, a aplicação faz autenticação em algum banco de dados, geralmente relacional, que valida ou não as informações passadas para o sistema. No caso de validação aceita, a aplicação libera o acesso aos recursos desse sistema ao usuário requerente.

Vendo esta tendência como um problema para o mercado, houve o surgimento de várias ferramentas que suprissem essa deficiência, no que diz respeito à autenticação e à

autorização. No mundo opensource, também existem várias ferramentas capazes de trazer a funcionalidade de autenticação única para um sistema. Para escolher a ferramenta necessária, é preciso mapear alguns tópicos para que seja possível integrar o sistema, o método de autenticação e a base de dados.

Entre esses tópicos, existem:

a) Base de dados utilizada;
b) Quais as linguagens dos sistemas que serão implantadas o SSO;
c) Quais as plataformas que rodam as aplicações; e
d) Qual o método de autenticação a ser utilizado.

Estes tópicos podem auxiliar na tomada de decisão de qual ferramenta opensource pode ser utilizada para a integração da autenticação dos usuários em uma única aplicação sendo que será reaproveitada em várias.

3.3. Implantação de Single Sign-on

Dentro de um escopo de um projeto de SSO, a escolha de ferramentas é imprescindível para o sucesso de uma boa implantação de SSO dentro de uma organização. Visto as ferramentas supracitadas, iniciar-se-á o mapeamento para relatar, neste trabalho, o apoio de decisão para o projetista.

A partir desse subtítulo, será descrito como realizar os passos para identificar as ferramentas necessárias para a integração e o funcionamento do conceito de SSO.

Na implantação de SSO, deve-se separar em três tópicos para se definir um projeto de SSO: autenticação, levantamento de linguagens e escolhas de ferramentas SSO.

4. CONCLUSÃO

Neste trabalho, foi possível analisar os benefícios da implantação de um sistema SSO, que são trazidos a uma organização através de ferramentas que possuem código aberto.

Além de a empresa resolver inúmeros problemas com a administração de sistemas com diferentes bases de autenticação, por meio da implantação de SSO com ferramentas de código aberto, é possível implantar essa tecnologia com custos menores, sem precisar fazer a aquisição dessas ferramentas.

Verificando-se que existem várias ferramentas e vários conceitos de gerenciamento de identidade, este trabalho focou no apoio ao gestor do projeto de implantação de SSO ou do gestor da área de TI de uma organização para a escolha de quais ferramentas serão necessárias à implantação da tecnologia de SSO, integrando, assim, as credencias das aplicações web.

Com as diversas áreas de gerenciamento de identidade e até com o aprofundamento da tecnologia SSO, é possível realizar trabalhos futuros em autenticação através de interorganização, no qual é possível realizar a autenticação utilizando bases de usuários em lugares de terceiros por meio do SAML.

REFERENCES

.NET Framework. Disponível em: <http://en.wikipedia.org/wiki/.NET_Framework>. Acesso em: 14 set de 2009.

389 Directory Server (Open Source LDAP). Disponível em: <http://directory.fedoraproject.org>. Acesso em: 02 set de 2009.

ACTIVE Server Pages. Disponível em: <http://en.wikipedia.org/wiki/Active_Server_Pages>. Acesso em: 12 set de 2009.

ASP. Disponível em: http://pt.wikipedia.org/wiki/ASP. Acesso em: 12 set de 2009.

ASP.NET. Disponível em: <http://en.wikipedia.org/wiki/ASP.NET>. Acesso em: 12 set de 2009.

ASP.NET. Disponível em: <http://pt.wikipedia.org/wiki/ASP.NET>. Acesso em: 12 set de 2009.

CAS | Jasig Community. Disponível em: <http://www.jasig.org/cas>. Acesso em: 20 set de 2009.

CENQUA OS: Shaj. Disponível em: <http://opensource.cenqua.com/shaj>. Acesso em: 07 set de 2009.

CUSTOMER Relationship Management. Disponível em: <http://en.wikipedia.org/wiki/Customer_relationship_management>. Acesso em: 05 set de 2009.

DSML. Disponível em: <http://en.wikipedia.org/wiki/DSML>. Acesso em: 02 set de 2009.

HIGGINS Home. Disponível em: <http://www.eclipse.org/higgins>. Acesso em: 29 set de 2009.

HOME - Safehaus. Disponível em: <http://docs.safehaus.org/display/PENROSE/Home>. Acesso em: 16 set de 2009

IDENTITY Management – Wikipedia, La enciclopédia libre. Disponível em: <http://es.wikipedia.org/wiki/Identity_Management>. Acesso em: 22 set de 2009.

IDENTITY Management OSS Map. Disponível em: <http://docs.safehaus.org/display/HAUS/Id+OSS+Map>. Acesso em: 22 jun de 2009.

IDENTITY management: the expert view | 23 Jul 2007 | ComputerWeekly.com. Disponível em: <http://www.computerweekly.com/Articles/2007/07/23/225715/identity-management-the-expert-view.htm>. Acesso em: 22 set de 2009.

IT Services: Stanford WebAuth. Disponível em: <http://webauth.stanford.edu>. Acesso em: 07 set de 2009.

JAVA. Disponível em: <http://en.wikipedia.org/wiki/Java_(programming_language)>. Acesso em: 09 set de 2009.

JAVA. Disponível em: <http://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)>. Acesso em: 09 set de 2009.

JAVASERVER Page. Disponível em: <http://en.wikipedia.org/wiki/JavaServer_Pages>. Acesso em: 10 set de 2009.

JØSANG, Audun; POPE, Simon. User centric identity management. Austrália: The University Quennsland, 2005

KERBEROS. Disponível em:< http://pt.wikipedia.org/wiki/Kerberos>. Acesso em: 22 jun de 2009.

LDAP. Disponível em:< http://pt.wikipedia.org/wiki/LDAP>. Acesso em: 31 ago de 2009.

LINGUAGEM de programação. Disponível em: <http://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o>. Acesso em: 09 set de 2009.

LOI, Leandro Nascimento. UM ESTUDO DAS METODOLOGIAS OPEN SOURCE IDENTITY MANAGEMENT E INDICAÇÃO DA MELHOR A SER IMPLANTADA NO PROJETO VIA DIGITAL. Florianópolis, 2007.

METADIRECTORIO. Disponível em: <http://es.wikipedia.org/wiki/Metadirectorio>. Acesso em: 17 set de 2009.

METADIRECTORY. Disponível em: <http://en.wikipedia.org/wiki/Metadirectory>. Acesso em: 02 set de 2009.

OASIS Security Services (SAML) TC. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#overview>. Acesso em: 07 set de 2009.

OPENDS: the Open Source Java LDAP Directory Service. Disponível em: <http://www.openldap.org>. Acesso em: 02 set de 2009.

OPENLDAP. Disponível em: <http://www.openldap.org>. Acesso em: 02 set de 2009.

OPENSSO: Home. Disponível em: <https://opensso.dev.java.net>. Acesso em: 06 set de 2009.

PERL. Wikipedia, a enciclopédia livre. Disponível: <http://pt.wikipedia.org/wiki/Perl>. Acesso em: 20 set de 2009.

PERL - Wikipedia, the free encyclopedia. Disponível em: <http://en.wikipedia.org/wiki/Perl>. Acesso em: 20 set de 2009.

PHP. Disponível em: <http://en.wikipedia.org/wiki/PHP>. Acesso em: 10 set de 2009.

PUBCOOKIE. Disponível em: <http://www.pubcookie.org>. Acesso em: 06 set de 2009.

SAMBA - opening window to a wider world. Disponível em: <http://us4.samba.org/samba>. Acesso em: 18 set de 2009.

SAML 2.0. Disponível em: <http://en.wikipedia.org/wiki/SAML_2.0>. Acesso em: 07 set de 2009.

SAMLDiff. Disponível em: <https://spaces.internet2.edu/display/SHIB/SAMLDiffs>. Acesso em: 08 set de 2009.

SECURITY Assertion Markup Language. Disponível em: <http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language>. Acesso em: 07 set de 2009.

SERVICE Provisioning Markup Language. Disponível em: <http://en.wikipedia.org/wiki/Service_Provisioning_Markup_Language>. Acesso em: 12 out de 2009.

SINGLE Sign-On. Disponível em: <http://en.wikipedia.org/wiki/Single_sign-on>. Acesso em: 22 jun de 2009.

SOAP. Disponível em: <http://en.wikipedia.org/wiki/SOAP>. Acesso em: 07 set de 2009.

THE A-Select Authentication System. Disponível em: <http://a-select.surfnet.nl/home.html>. Acesso em: 07 set de 2009.

VIRTUAL Directory. Disponível em: <http://en.wikipedia.org/wiki/Virtual_directory>. Acesso em: 02 set de 2009.

VIRTUAL vs Meta Directory. Disponível em: <http://docs.safehaus.org/display/PENROSE/Virtual+vs+Meta+Directory>. Acesso em: 17 set de 2009.

WINDLEY, Phillip J. Digital identity. USA: O'Reilly Media, Inc., 2005.Dyer, S., Martin, J. and Zulauf, J. (1995) "Motion Capture White Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.