HAW Hamburg

Performance Untersuchung von WS-Security Implementierungen in interoperablen Umgebungen

Master Thesis Outline

Eike Falkenberg

Im Master Studiengang Informatik
Wintersemester 2006 / 2007
Department Informatik
Fakultät Technik und Informatik
Hochschule für Angewandte Wissenschaften Hamburg

Agenda

- Motivation
- Grundlagen
- Anwendungsszenario
- Untersuchungsraum
- Realisierung
- Risiken

Motivation

Die Web Services Security Standards sind inzwischen so weit entwickelt, dass der Einsatz in Unternehmen immer öfter ein Thema wird.

Bisher gibt es nur wenige Untersuchungen zu der Performance und der Netzwerklast von Web Services Security.

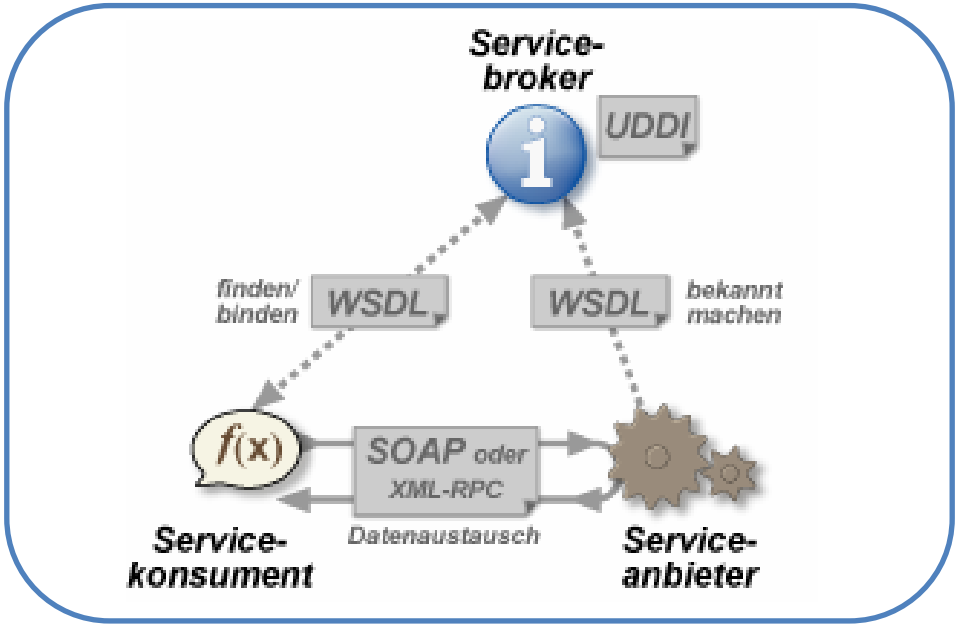
Die vorhandenen Untersuchungen beschränken sich auf einfache Lasttests von Prototypen ohne die Komplexität der realen Anforderungen zu berücksichtigen

Web Services

Interoperable- und plattformunabhängige

Kommunikation mittels XML Nachrichten (meist)

via HTTP



Web Services Security (WSS)

WSS definiert SOAP Header, die den sicheren Zugriff auf Web Services ermöglichen sollen

WSS definiert, wie Authentifiziert, Signiert und Verschlüsselt werden soll

WSS ist ein Menge von OASIS Spezifikationen, die von Firmen wie IBM, Microsoft, Sun und Verisign entwickelt wurden

WSS dient als Grundlage zu weiteren Spezifikationen

WSS UsernameToken

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
    <S:Header>
    ...
        <wsse:Security>
            <wsse:UsernameToken >
                <wsse:Username> Zoe </wsse:Username>
                <wsse:Password> ILoveDogs </wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    ...
    </S:Header>
...
</S:Envelope>
```

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
    <S:Header>
    ...
        <wsse:Security>
            <wsse:UsernameToken
                xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext"
                xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility">
                <wsse:Username> NNK </wsse:Username>
                <wsse:Password Type="wsse:PasswordDigest">
                    D2A12DFE8D9F0C6BB82C89B091DF5C8A872F94DC
                </wsse:Password>
                <wsse:Nonce> EFD89F06CCB28C89 </wsse:Nonce>
                <wsu:Created> 2001-10-13T09:00:00Z </wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    ...
    </S:Header>
...
</S:Envelope>
```

WSS BinarySecurityToken

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
    <S:Header>
    ...
        <wsse:Security>
            <wsse:BinarySecurityToken ValueType="wsse:Kerberosv5ST"
                EncodingType="wsse:Base64Binary" wsu:Id="KerberosToken">
                MIIEZzCCA9CgAwIBAgIQEmtJZc0...
            </wsse:BinarySecurityToken>
        </wsse:Security>
    ...
    </S:Header>
...
</S:Envelope>
```

Kerberos Ticket

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
    <S:Header>
    ...
        <wsse:Security>
            <wsse:BinarySecurityToken ValueType="wsse:X509v3"
                EncodingType="wsse:Base64Binary" wsu:Id="X509Token">
                AIIEZzCCA6DgAwIBAgIQEmtJZc0rqfKh6i...
            </wsse:BinarySecurityToken>
        </wsse:Security>
    ...
    </S:Header>
...
</S:Envelope>
```

X.509 Zertifikat

WS-Security Stack (WS-*)

Als WS-Security Stack bezeichnet man eine Menge von Spezifikationen, die auf der WSS Spezifikation aufbauen

- WS-SecurityPolicy
- WS-Trust
- WS-SecureConversation
- WS-Federation
- WS-Authorisation
- WS-Privacy

WS-SecurityPolicy

WS-Policy beschreibt ein Werkzeug, welches Web Services die Möglichkeit bietet, Ihre Bedingungen und Anforderungen zu beschreiben. Diese Beschreibungen werden Policy Assertions genannt.

WS-Policy deklariert:

- welche Algorithmen verwendet werden können
- die erforderlichen Signaturen und Verschlüsselungen
- welche SecurityTokens akzeptiert werden

WS-SecurityPolicy

```
<wsp:Policy xmlns:wsp="..." xmlns:sp="...">
    <sp:SymmetricBinding>
        <wsp:Policy>
            <sp:ProtectionToken>
                <wsp:Policy>
                    <sp:Kerberos sp:IncludeToken=".../IncludeToken/Once" />
                    <wsp:Policy>
                        <sp:WSSKerberosV5ApReqToken11/>
                    <wsp:Policy>
                    </sp:Kerberos>
                </wsp:Policy>
            </sp:ProtectionToken>
            <sp:SignBeforeEncrypting />
            <sp:EncryptSignature />
        </wsp:Policy>
    </sp:SymmetricBinding>
    <sp:SignedParts>
        <sp:Body/>
        <sp:Header Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
    </sp:SignedParts>
    <sp:EncryptedParts>
         <sp:Body/>
    </sp:EncryptedParts>
</wsp:Policy>
```

Welche Tokens werden akzeptiert, wie muss verschlüsselt bzw. signiert werden

Erforderliche Signaturen

Erforderliche Verschlüsselung

Web Services Trust Language

WSS gesicherte Zugriffe können zu folgenden Problemen führen:

- Kompatibilitätsprobleme beim Security Token Format
- Glaubwürdigkeit des Security Tokens
- Unterschiedliche Namespaces

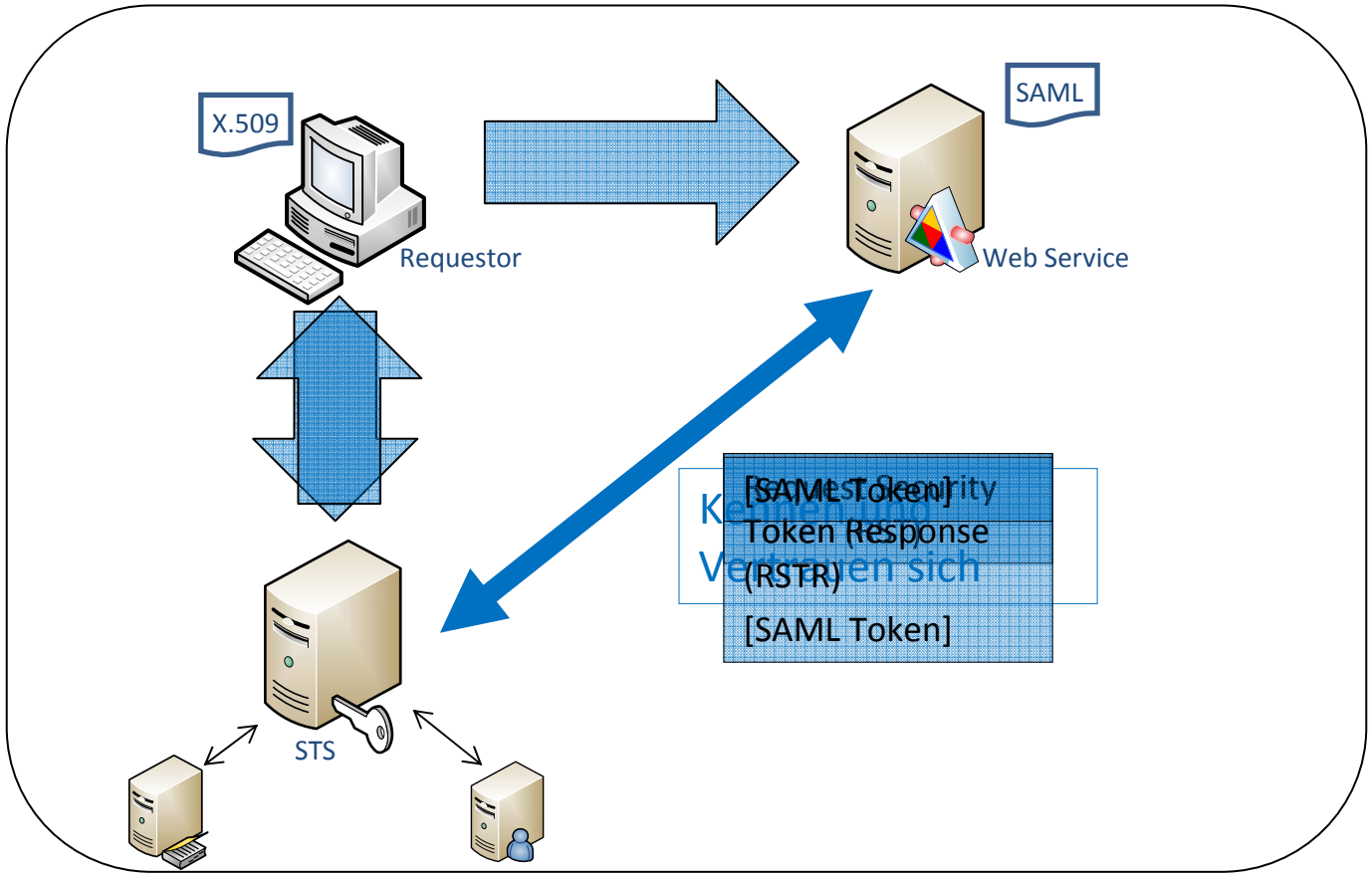
Web Services Trust Language

WS Trust definiert ein Request-Response Protokoll, bestehend aus

- `RequestSecurityToken` (RST)
- `RequestSecurityTokenResponse` (RSTR)

Und führt einen Security Token Service (STS) als zentrale Stelle zum erzeugen, prüfen und erneuern von Security Tokens ein.

Web Services Trust Language



WS-Trust RequestSecurityToken

```
<soap:Body>
    <wstrust:RequestSecurityToken>
        <wstrust:TokenType>SAML</TokenType>
        <wstrust:RequestType>ReqExchange</RequestType>
        <wstrust:OnBehalfOf>
            <ws:BinarySecurityToken id="x509OriginalToken"
                ValueType="X.509">
            ssjhkhSjshjsaHJS...
            </ws:BinarySecurityToken>
        </wstrust:OnBehalfOf>
    </wstrust:RequestSecurityToken>
</soap:Body>
```

```
<soap:Body>
    <wstrust:RequestSecurityTokenResponse>
        <wstrust:TokenType>SAML</TokenType>
            <wstrust:RequestedSecurityToken>
                <saml:Assertion AssertionID="2se8e/vaskfsdif="
                    Issuer="www.sts.com"
                    IssueInstant="2007-01-10T16:58:33.173Z">
                    <saml:Conditions NotBefore="2007-01-10T16:53:33.173Z"
                        NotOnOrAfter="2007-01-10T17:08:33.173Z"/>
                    <saml:AuthenticationStatement
                        AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X.509"
                        AuthenticationInstant="2007-01-10T16:57:30.000Z">
                        <saml:Subject>
                            ...converted client identifier...
                        </saml:Subject>
                    </saml:AuthenticationStatement>
                    <ds:Signature><-- calculated by STS --></ds:Signature>
                </saml:Assertion>
            </wstrust:RequestedSecurityToken>
    </wstrust:RequestSecurityTokenResponse>
</soap:Body>
```

WS-SecureConversation

Nach WS-Trust muss für jede Anfrage des Requestors erneut ein Token beim STS angefordert werden.

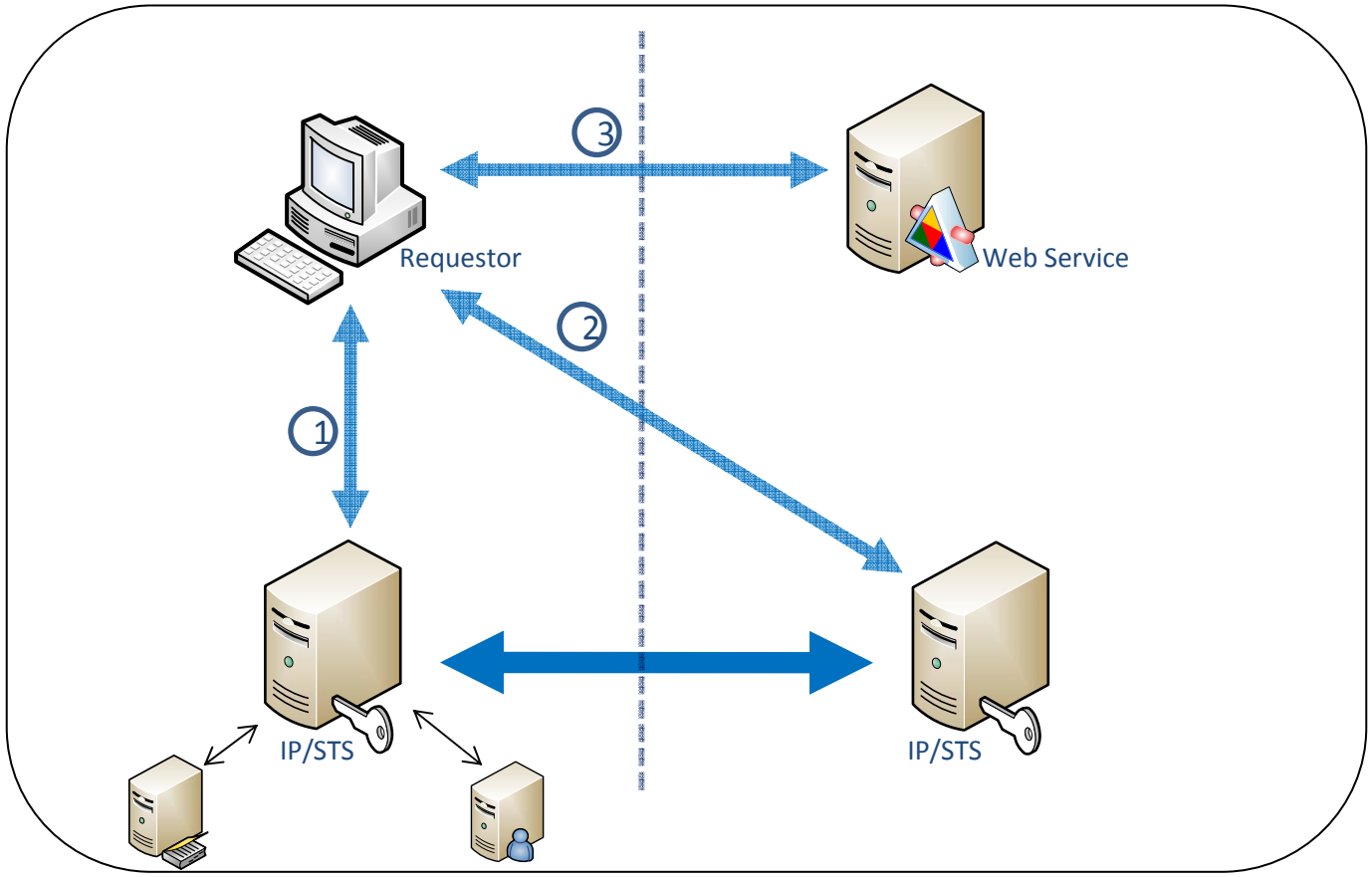
Um die Last auf den STS zu reduzieren, beschreibt WS-SecureConversation einen SecurityContextToken mit dessen Hilfe die beiden Teilnehmer einen gemeinsamen Sicherheitskontext aufbauen.
Von einem Basisschlüssel leiten die Teilnehmer selber Folgeschlüssel ab

WS-Federation

Identitäten sind normalerweise an einen Realm gebunden. Wenn gesichert Zugriffe über Unternehmensgrenzen hinweg ermöglicht werden sollen, entstehen besondere Herausforderungen.

WS-Federation beschreibt Mechanismen die es erlauben sollen, Identitäten, Authentisierungen und Authorisierungen über Realm Grenzen hinweg nutzen zu können.

WS-Federation



Anwendungsszenario

Das weltweit agierende Unternehmen *Falkenberg Worldwide* hat ein Data-Warehouse mit sensiblen Geschäftsdaten.
Die Geschäftsleitung soll auf alle, die Leiter der Divisionen auf Teile der Daten zugreifen dürfen.

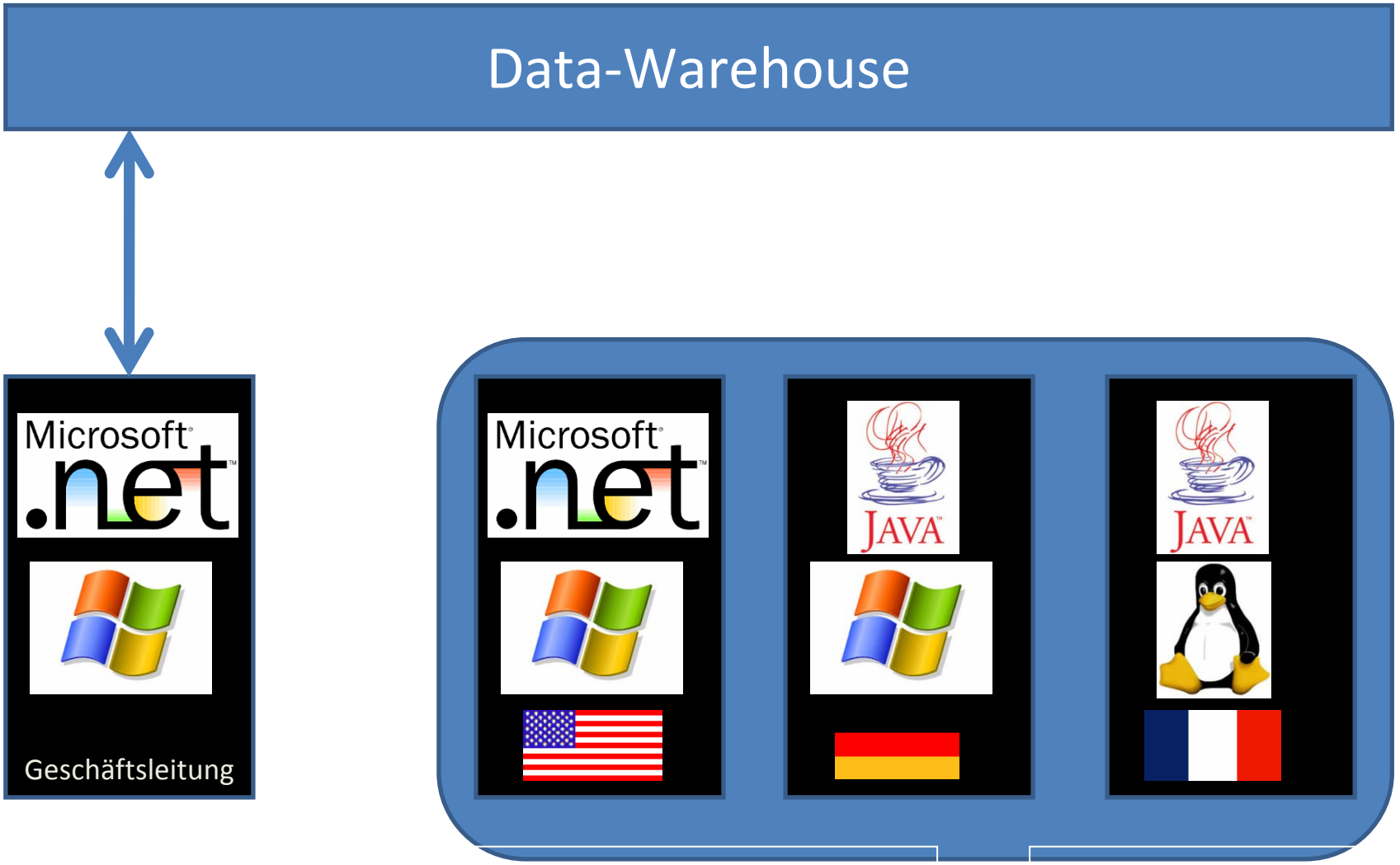
Das Unternehmen ist IT-technisch unkontrolliert gewachsen, was dazu geführt hat, dass die Divisionen unterschiedliche Betriebssysteme einsetzen (Windows und Linux) und Software in verschiedenen Programmiersprachen (Java und C#).

Anwendungsszenario

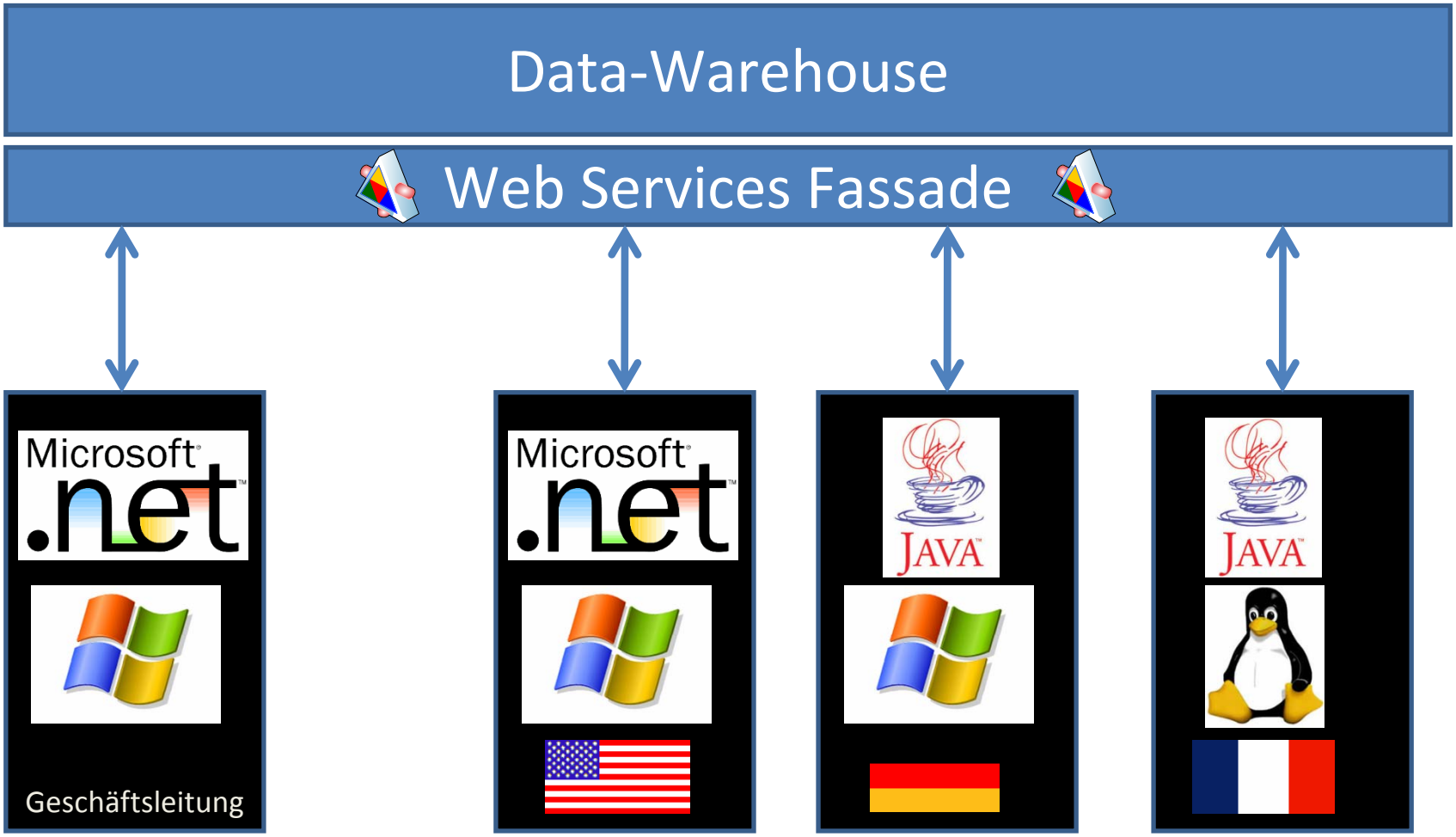
Bisher hat die Geschäftsleitung mit Ihrer C# Software direkt auf das Data-Warehouse zugegriffen und die Divisionsleiter bekamen monatliche Reportings per Email zugesendet.

Im Zuge der Neustrukturierung soll ein einheitlicher Zugriff auf das Data-Warehouse über Web Services erfolgen. Die Divisionsleiter sollen die Möglichkeit bekommen, jederzeit Ihre Daten aus dem Data-Warehouse zu bekommen um schneller auf Marktveränderungen reagieren zu können.

Anwendungsszenario



Anwendungsszenario (Ziel)



Anwendungsszenario

Der CIO hat kürzlich etwas über WSS gehört und hat nach einiger Prüfung befunden, dass WS-Trust hier sinnvoll eingesetzt werden könnte

Da nun mehrere Betriebssysteme und Programmiersprachen kombiniert auftreten, steht zur Diskussion, die Infrastruktur zu vereinheitlichen.

Um dies abwägen zu können, bedarf es Daten über die Performance, Netzwerklast und Skalierung der auf dem Markt verfügbaren Technologien …

Untersuchungsraum

Für das Anwendungsszenario soll folgendes untersucht werden:

- Netzwerklast
- Performance
- Skalierbarkeit
- Interoperabilität

der verfügbaren WSS-Implementierungen

Sicher auch spannend, aber hier nicht:
Migrationskosten, Wartbarkeit, Vorstandsmeinung, …

Untersuchungsraum

Folgende WSS Implementierungen sollen untersucht, verglichen und bewertet werden:

- .NET 2.0 & WSE (Windows)
- .NET 3.0 WCF (Windows)
- Apache Axis & WSS4J (Windows/Linux)
- Sun's Project Tango (Windows/Linux)

Realisierung

Tests bezüglich der Netzwerklast können optimal mit virtuellen Maschinen (VmWare, VirtualPC, …) durchgeführt werden.

Die Performance Tests können ebenfalls mit VMs durchgeführt werden, da man mittels Snapshots reproduzierbare Ergebnisse erhält.

Die Skalierung sollte auf echter Hardware realisiert erpropt werden, da zu viele VMs auf einem System die Resultate verwischen könnten

Ziele

Untersuchung und Bewertung von Performance, Netzwerklast und Skalierbarkeit der derzeit verfügbaren WSS-Implementierungen. Besonderes Augenmerk liegt auf WS-Trust unter interoperablen Bedingungen auf unterschiedlichen Betriebssystemen.

Auf bestehenden Arbeiten aufbauend, soll eine praxisrelevantere Untersuchung zu verwendbaren Ergebnissen führen.

Risiken

- Hoher Konfigurationsaufwand und somit viel Zeit
- Mangelhafte Interoperabilität bei den Implementierungen
- Abweichung von Standards

Quellen

- **Organization for the Advancement of Structured Information Standards (OASIS)**
  http://www.oasis-open.org
- **WS-SecurityPolicy**
  http://docs.oasis-open.org/ws-sx/ws-securitypolicy/
- **WS-Trust**
  http://docs.oasis-open.org/ws-sx/ws-trust/
- **WS-SecureConversation**
  http://docs.oasis-open.org/ws-sx/ws-secureconversation/
- **WS-Federation**
  http://www-128.ibm.com/developerworks/library/specification/ws-fed/

Quellen

- **A Performance Evaluation of Web Services Security**
  Tang et Al., IEEE, 2006
- **Performance of Web Services Security**
  Liu et Al, In Proceedings of the 13th Annual 13th Mardi Gras Conference, Baton Rouge, Louisiana, USA, February 2005
- **Understanding Web Services Specifications and the WSE**
  Jeannine Hall Gailey, MS Press 2004
- **Programming Indigo**
  David Pallmann, MS Press 2005
- **The year ahead in Java Web services**
  Sosnoski, IBM
  http://www-128.ibm.com/developerworks/java/library/ws-java1.html
- **Evaluation and Modeling of Web Services Performance**
  Chen et Al, IEEE, 2006