Web Services

XML, WSDL, SOAP und UDDI
Einblicke und Ausblicke

Architektur von Web Services und ergänzende Technologien

- Inhalt
  - Sicherheit
    - WS-License und WS-Security
  - Prozessfluss
    - XLANG
  - Transaktions-Koordination
    - BTP
    - Erweiterte Transaktionen
  - Messaging
    - WS-Inspection, WS-Referral, WS-Routing
    - BEEP
    - Zuverlässiges HTTP
  - Web Service Grundlagen
    - RosettaNet
    - XML-RPC

Architektur und Ergänzungen

- *Zusätzliche Technologien müssen die Web Service Architektur vervollständigen.*
  - SOAP, WSDL und UDDI sind die Basistechnologien für Web Sevices.
  - Was wird zusätzlich gebraucht?
    - Sicherheit
    - Prozessfluss
    - Transaktionen
    - Garantiertes Messaging

Architektur und Ergänzungen

- *Sicherheit, Prozessfluss, Transaktionen und Messaging.*
  - Security
    - Integrität, Authentifizierung und Autorisierung sind zwingend um Web Services vertrauenswürdig einsetzen zu können.
  - Prozessfluss
    - Der Ausführungsfluss, die Kombination mehrerer Web Services, muss spezifizierbar und kontrollierbar sein.
  - Transaktionen
    - Die Koordination mehrerer Web Services muss möglich sein.
  - Messaging
    - Konfiguration, Pfadspezifikation und Routing muss zuverlässig möglich sein, auch über Zwischenknoten.

Architektur und Ergänzungen

- *Eine Web Service Referenzarchitektur muss geschaffen werden.*
  - Bisher haben wir Web Services im Wesentlichen aus SOAP und WS`DL zusammengebaut.
  - Das Zusammenspiel im Kontext mit Sicherheitssystemen und Transaktionen muss sauber definiert werden.
  - Denkbar sind Web Service Container, in C#, Java, ... welche mit anderen Containern (J2EE, .NET) mit WSDL als Schnittstellenbeschreibung zusammenarbeiten.
    - W3C Web Service Architecture Working Group.

Architektur und Ergänzungen
Sicherheit

- *Security ist eine Grundanforderung.*
  - Security ist eine Grundanforderung aber auch ein Grundproblem Web basierter Informationssysteme.
    - Zuviel Security blockiert die Weiterentwicklung
    - Zuwenig Security verhindert die generelle Akzeptanz.
  - Grundanforderung:
    - Sichere und vertrauliche Daten (Kreditkarte, u.s.w.)
      - Startpunkt: SSL/TLS (IETF), HTTP over SSL (HTTPS)
    - Authentifizierung und Autorisierung (Rechte)
      - Startpunkt: Benutzername/Passwort
    - Firewalls

Architektur und Ergänzungen
Sicherheit

- *Zusätzliche Security Ansätze.*
  - Autorisierung / Authentifizierung
    - Standard for Authentification and Authorization (SAML)
    - Standard für Public Key Management (XKMS)
  - HTTPS genügt nicht
    - SOAP über HTTPS reicht nicht aus.
    - SAML bietet mehr Möglichkeiten,

Architektur und Ergänzungen
Sicherheit - SAML

- *Security Assertions Markup Language (SAML) von OASIS.*
  - Bietet einen Standard, mit dessen Hilfe Autorisierungs- und Authentifizierungs-Informationen definiert werden können.
  - Propagiert Authentifizierungs- Information.
  - SAML ist an XML Framework für den Austausch von Security Informationen über das Internet.
  - SAML gestattet es unterschiedlichen Security Services zusammenzuarbeiten.

Architektur und Ergänzungen
Sicherheit - SAML

- *SAML Beispiel:Request*
  -
```
<samlp: Request ...>
    <samlp: AttributeQuery>
    <saml: Subject>
        <saml: NameIdentifier
            SecurityDomain="sun. com"
            Name="rimap"/>
    </ saml: Subject>
    <saml: AttributeDesignator
        AttributeName="Employee_ ID"
        AttributeNamespace="sun. com">
    </ saml: AttributeDesignator>
    </ samlp: AttributeQuery>
</ samlp: Request>
```

Architektur und Ergänzungen
Sicherheit - SAML

- *SAML Beispiel:Response*
  - **<samlp: Response MajorVersion="1" MinorVersion="0"**
    RequestID="128.14.234.20.90123456"
    InResponseTo="123.45.678.90.12345678"
    StatusCode="Success">
    <saml: Assertion MajorVersion="1" MinorVersion="0"
    AssertionID="123.45.678.90.12345678"
    Issuer="Sun Microsystems, Inc."
    IssueInstant="2002- 01- 14T10: 00: 23Z">
    <saml: Conditions NotBefore="2002- 01- 14T10: 00: 30Z"
    NotAfter="2002- 01- 14T10: 15: 00Z" />
    <saml: AuthenticationStatement
    AuthenticationMethod="Password“
    AuthenticationInstant="2001- 01- 14T10: 00: 20Z">
    <saml: Subject> <saml: NameIdentifier
    SecurityDomain="sun. com" Name="rimap" />
    </ saml: Subject>
    </ saml: AuthenticationStatement>
    </ saml: Assertion>
    </ samlp: Response>

Architektur und Ergänzungen
Sicherheit - XKMS

- *XML Public Key Management Specification*
  - Definiert ein Protokoll für die Verteilung und Registrierung von Public Keys für Verschlüsselung und Entschlüsselung von Messages über SOAP.
  - XKMS besteht aus zwei Teilen:
    - X-KISS: XML Key Information Service Specification
      - Definiert einen abstrakten Mechanismus es erlaubt unterschiedliche Spezifikationen (PKI für X.509 PKIX,...) einzubinden.
    - X-KRSS: XML Key Registration Service Specification
      - Definiert einen Web Service, mit dessen Hilfe Public Key Informationen registriert werden können.

Architektur und Ergänzungen
Sicherheit – WS-License, WS-Security

- *MS WS-Licence definiert Security Token-Formate für WS-Security.*
  - WS-Security ordnet SOAP Messages Lizenzen zu.
  - Sowohl X.509 als auch Kerberos Tickets werden als gültige Lizenzen akzeptiert.

Architektur und Ergänzungen
Process Flow

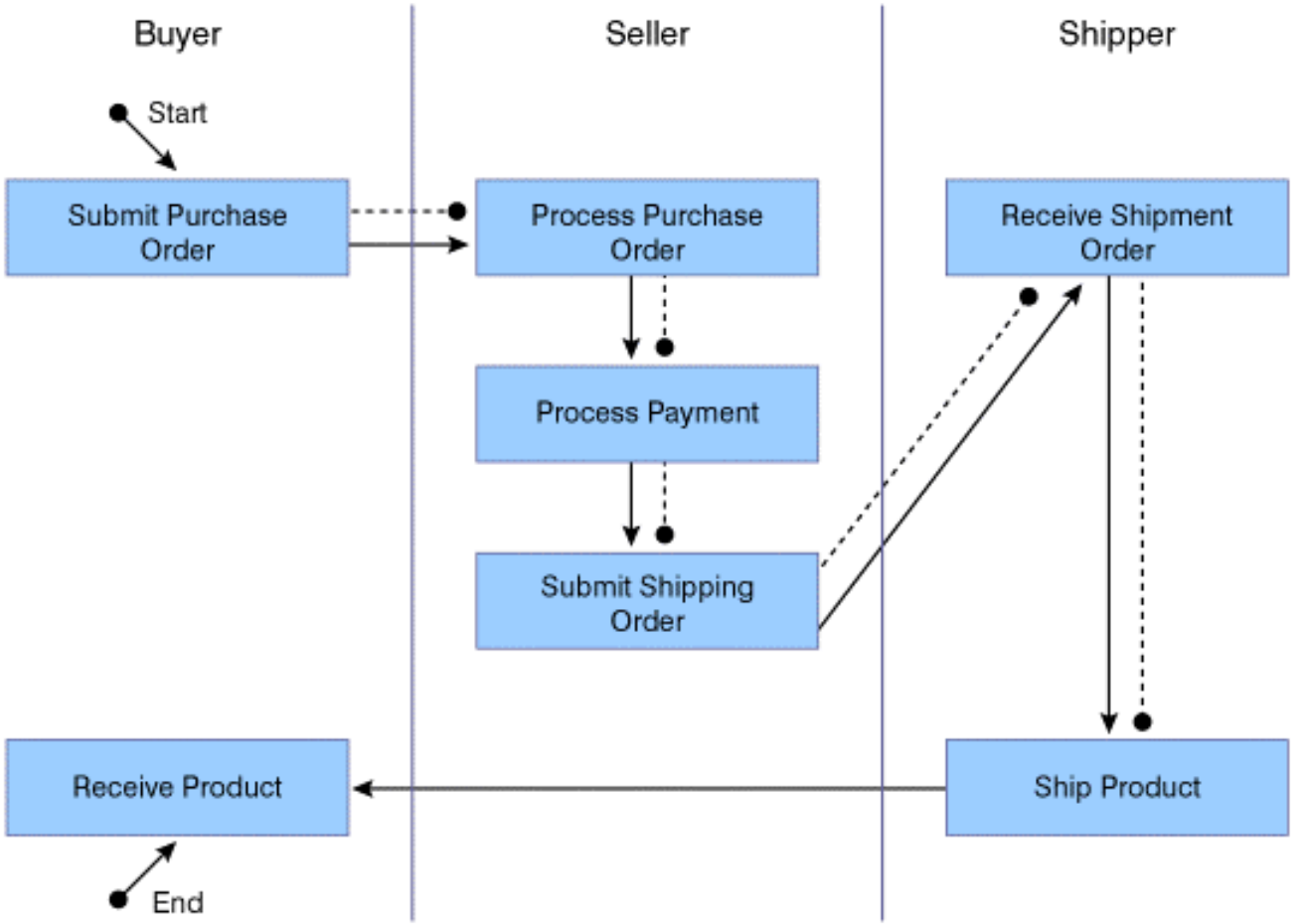
- *Die Orchestrierung komplexer Geschäftsbeziehungen verknüpft Web Services.*
  - Zwei Ansätze sind im Moment in Diskussion:
    - XLang von MS
    - WSFL Web Services Flow Language von IBM
  - XLang ist businessnäher
  - Beide versuchen ganze Prozessketten zu beschreiben.

Architektur und Ergänzungen
Process Flow - XLang

- *XLang basiert auf BizTalk.*
  - Erweitert WSDL analog zu RosettaNet
    - Sequenzen von Messages, welche zusammen einen Business Prozess implementieren.
  - XLang definiert neue Elemente
    - Für die Verzweigung (Switching, Branching)
    - Für Gruppenbildung
  - XLang wird typischerweise mit BizTalk von MS eingesetzt.

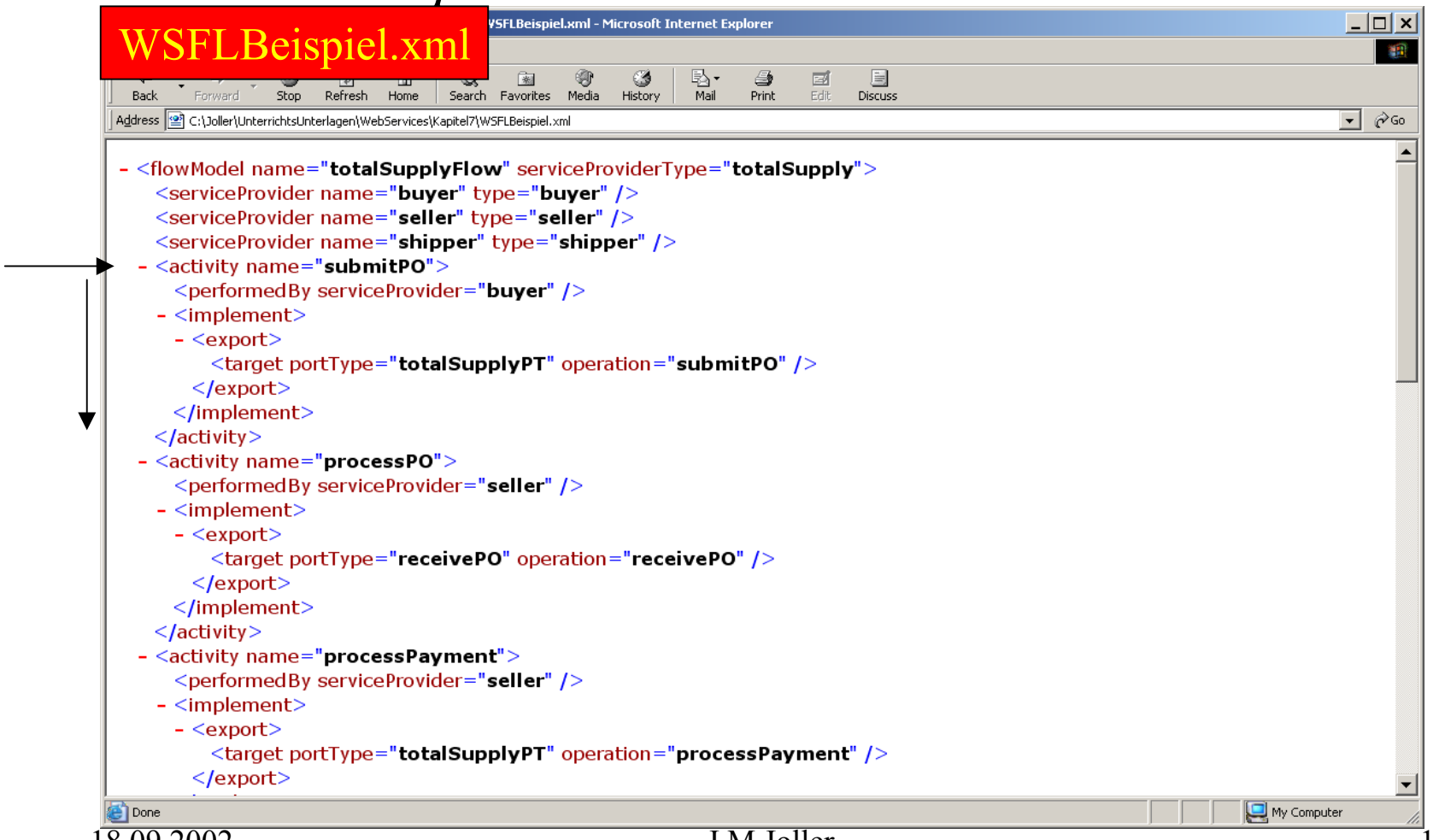
Architektur und Ergänzungen
Process Flow - WSFL

- *WSFL basiert auf IBM's MQ Series Produkten.*



Architektur und Ergänzungen
Process Flow - WSFL

- *WSFL Beispiel:*



Architektur und Ergänzungen Transaktions-Koordination

- *Transaktionen garantieren koordinierte Resultate*
  - Aber
    - Klassische Konzepte (ACID, 2-Phase Commit) stossen an Ihre Grenzen.
    - Grund:
      - Die grossen Zeitabstände auf dem Web können zu Zeitüberschreitungen führen
        - » Keine Transaktion wird mehr durchgeführt.
  - Ansätze
    - BTP : Business Transaction Protocol (OASIS)
      - Löst das Dilemma mit dem 2-Phase-Commit.
    - Extended Transaction Model (OMG: OTS, Java Community)

Architektur und Ergänzungen
Messaging

- *Messaging Ergänzungen betreffen das Routing und die Zuverlässigkeit.*
  - WS-Inspect von MS/IBM
    - Definiert einen Weg, wie man die an einer Adresse verfügbaren Web Services bestimmen kann.
    - Die Adresse muss also bekannt sein, nur die Dienste nicht.
  - WS-Referral
    - Ermöglicht eine dynamische Definition der SOAP Knoten in einem Message Pfad.
    - Arbeitet mit SOAP Header-Informationen und Attributen.
    - Definiert intermediäre Knoten, welche von SOAP Mesages benötigt werden.

Architektur und Ergänzungen
Messaging

- *Messaging Ergänzungen betreffen das Routing und die Zuverlässigkeit.*
  - WS-Routing
    - Definiert vollständige Message Pfade für das Routing von SOAP Messages.
    - Definiert einen SOAP Header, welcher Informationen enthält:
      - Betreffend Message Ersteller
      - Betreffend Message Endempfänger
      - Den nächsten Hop im Pfad
      - Einen Rückwärtspfad (für Rückmeldungen, falls überhaupt)

Architektur und Ergänzungen
Messaging

- *Messaging Ergänzungen betreffen das Routing und die Zuverlässigkeit.*
  - BEEP
    - Block Extensible Exchange Protocol der IETF
    - Generisches verbindungsorientiertes asynchrones Internet Protokoll.
    - Die Verbindung wird in Form eines Channel definiert.
    - BEEP unterstützt binäre und Text-Messages.
    - BEEP definiert einen Framing Mechanismus für den Austausch beliebiger MIME und XML Messages.

Architektur und Ergänzungen
Web Service Begründer

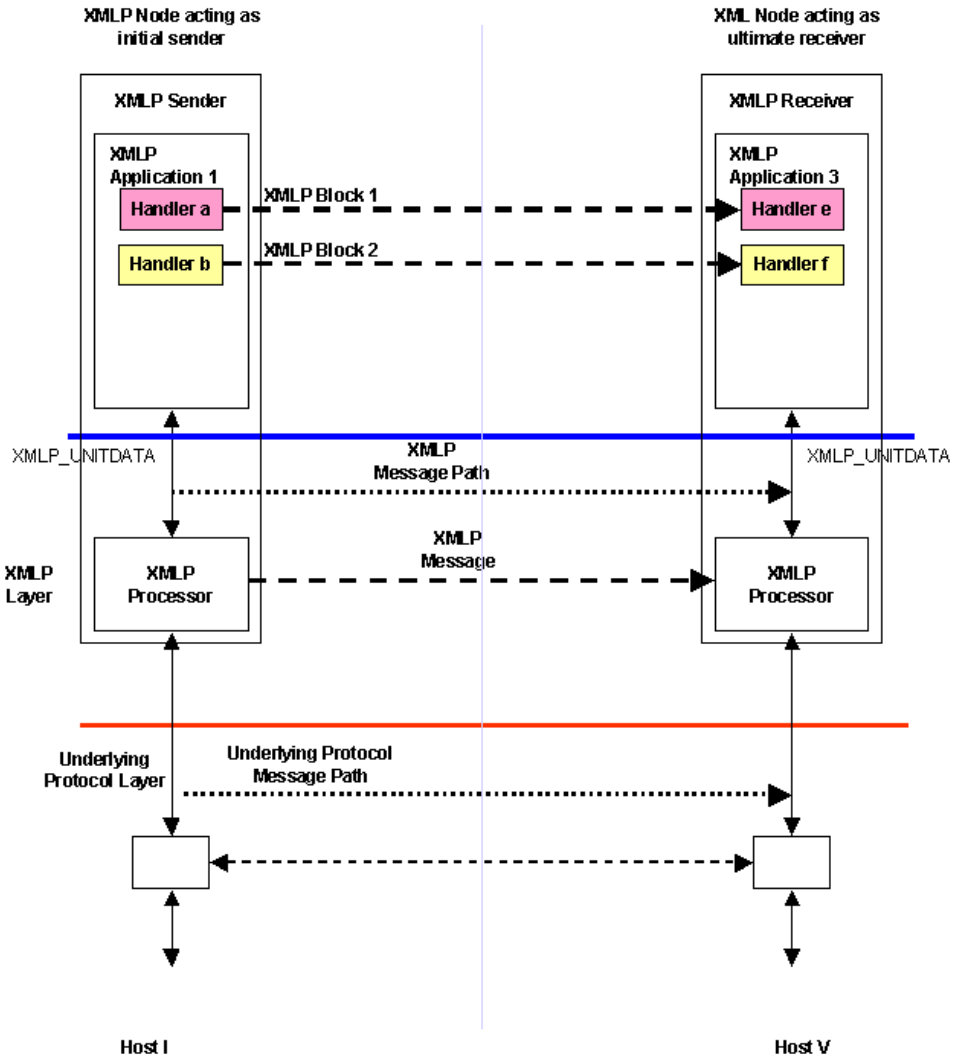
- *XML-RPC und Rosetta haben die Web Service Bewegung ausgelöst*
  - XML-RPC:
    - Praktische Bedeutung gering
      - SOAP bietet mehr bei vergleichbarem Aufwand
      - Userland startete die Bewegung zwecks Lösung von Web Problemen im Content Management Umfeld,
      - SOAP ist in etwa die Weiterentwicklung von XML-RPC
  - Rosetta:
    - Ist ähnlich gelagert wie Userland
      - Hauptkunden sind Zeitungen
      - Gesucht war eine bessere Nutzung des Internet.

Das XML Protocol Abstract Model

- Zielsetzungen
  - Keine Implementierung!
  - Abstrakte Darstellung, mit Bezug zu konkreten Protokollen
  - Soll keine API Spezifikation liefern!
  - Beschreibt das äussere Verhalten des XML Protokolls und Frameworks
  - Schreibt keine Implementationsarchitektur vor

  - Status:
    W3C Working Draft 9 July 2001
  - Link:
    http://www.w3.org/TR/xmlp-am/

Das XML Protocol Abstract Model



Drei grundlegende Begriffe

- XMLP Applikation:
  - Client oder User eines Services, der vom XML Protocol Layer angeboten wird.
    Eine XMLP Applikation kann auch vermittelnd zwischen Client und Server aktiv sein.

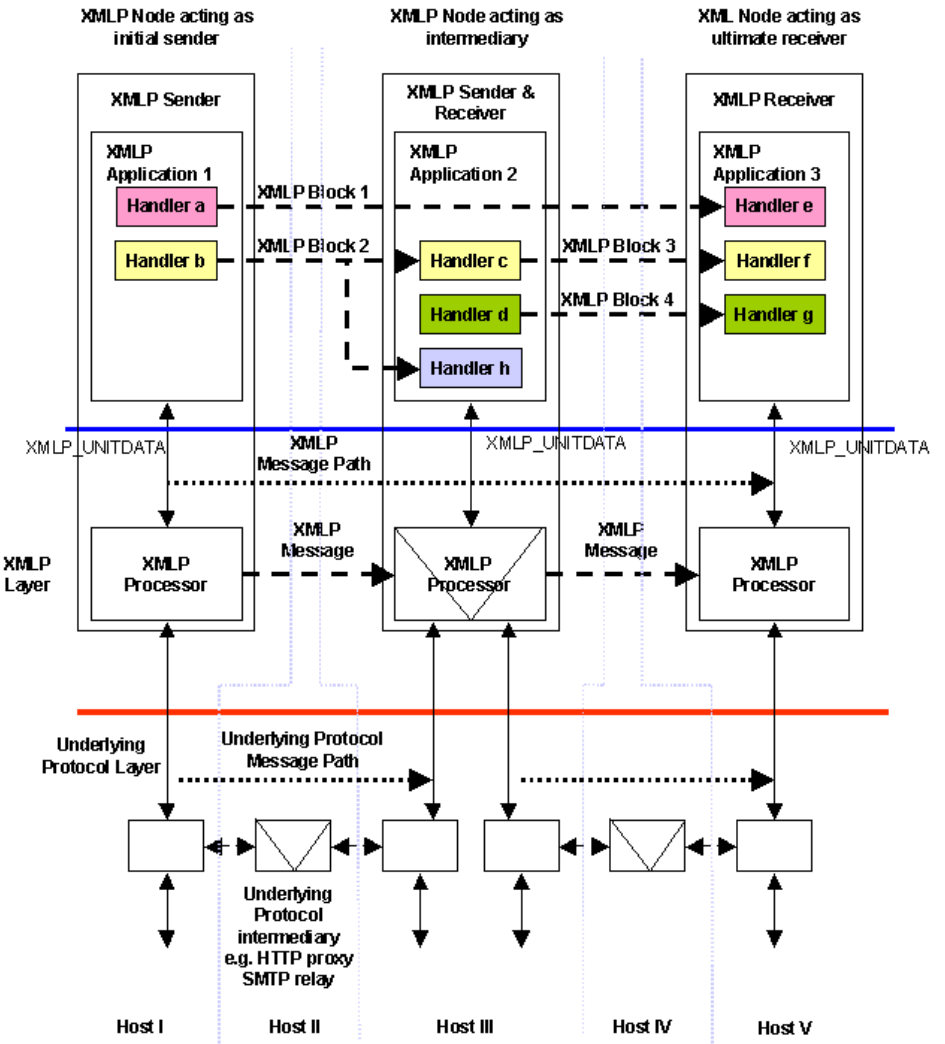
    XML Protocol Handlers
    - Sind in XMLP Anwendungen eingebettet

- XMLP Layer:
  - Stellt Dienste zur Verfügung, um Pakete von XMLP Clients zu XMLP Servern zu transferieren, eventuell über Zwischenknoten.

- XMLP Operationen
  - Elementare Funktionen oder Services des XMLP Layers

Komplexeres Szenario

Host III agiert als Gateway (unterschiedliche Protokolle bei I, V)



XML Protocol Layer Service Definition

- Definition eines abstrakten Interfaces zwischen dem XML Protokoll Layer und der XML Protokoll Applikation
  - XMLP_UnitData Operation, mit vier Events / Primitives

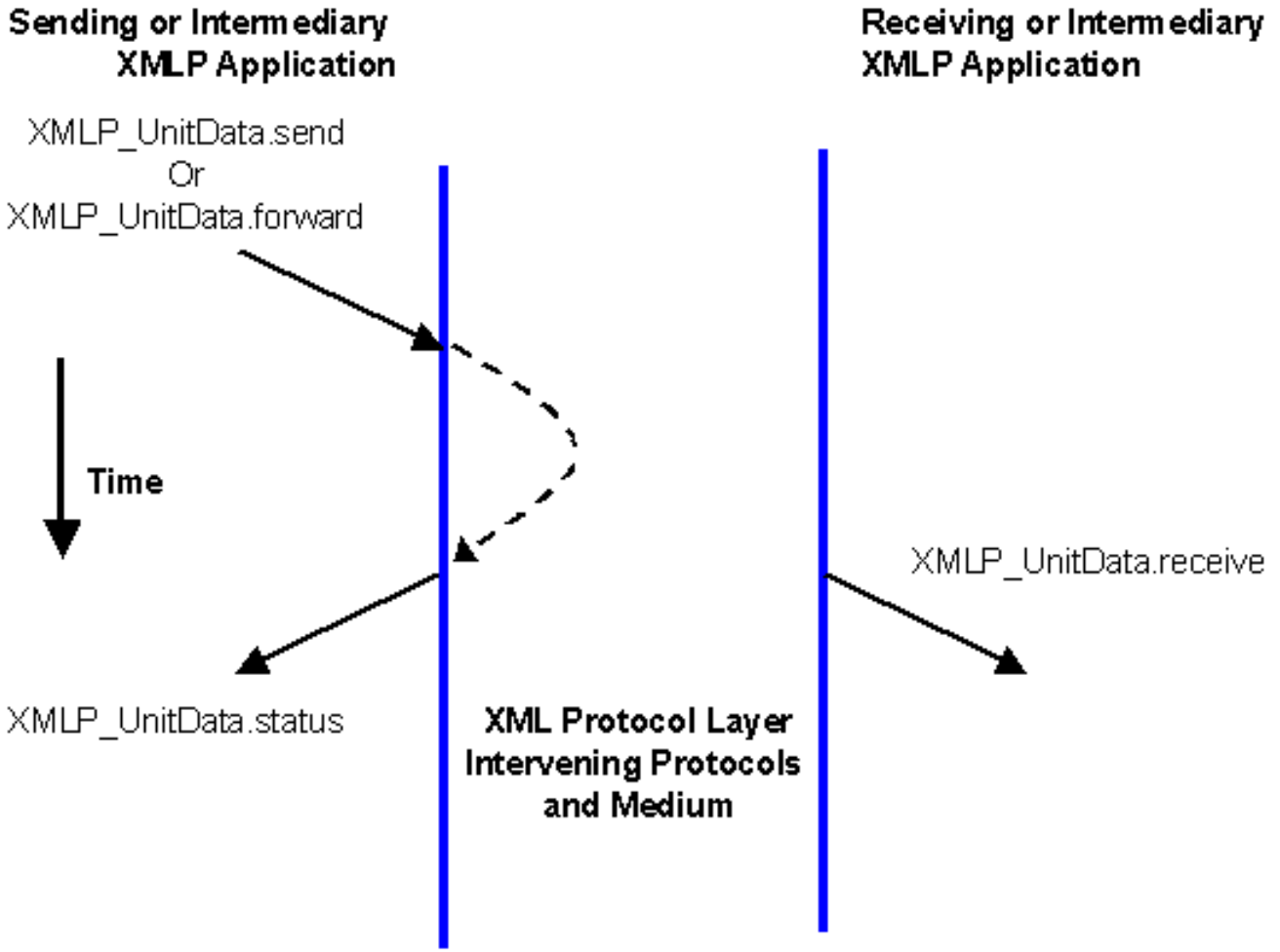
    XMLP_UnitData.**send**(To, [ImmediateDestination], Message, [Correlation], [BindingContext]);

    XMLP_UnitData.**receive**([To], [From], Message, [Correlation], [BindingContext]]);

    XMLP_UnitData.**status**([From], Status, [BindingContext]);

    XMLP_UnitData.**forward**([ImmediateDestination], Message, [BindingContext]]);
- Konzeptionell:
  - Die XMLP_UnitData Operation kapselt die Übertragung einer XML Protokoll Message vom senden zur empfangenden Appl.

XML Protocol Layer Service Definition : XMLP UnitData



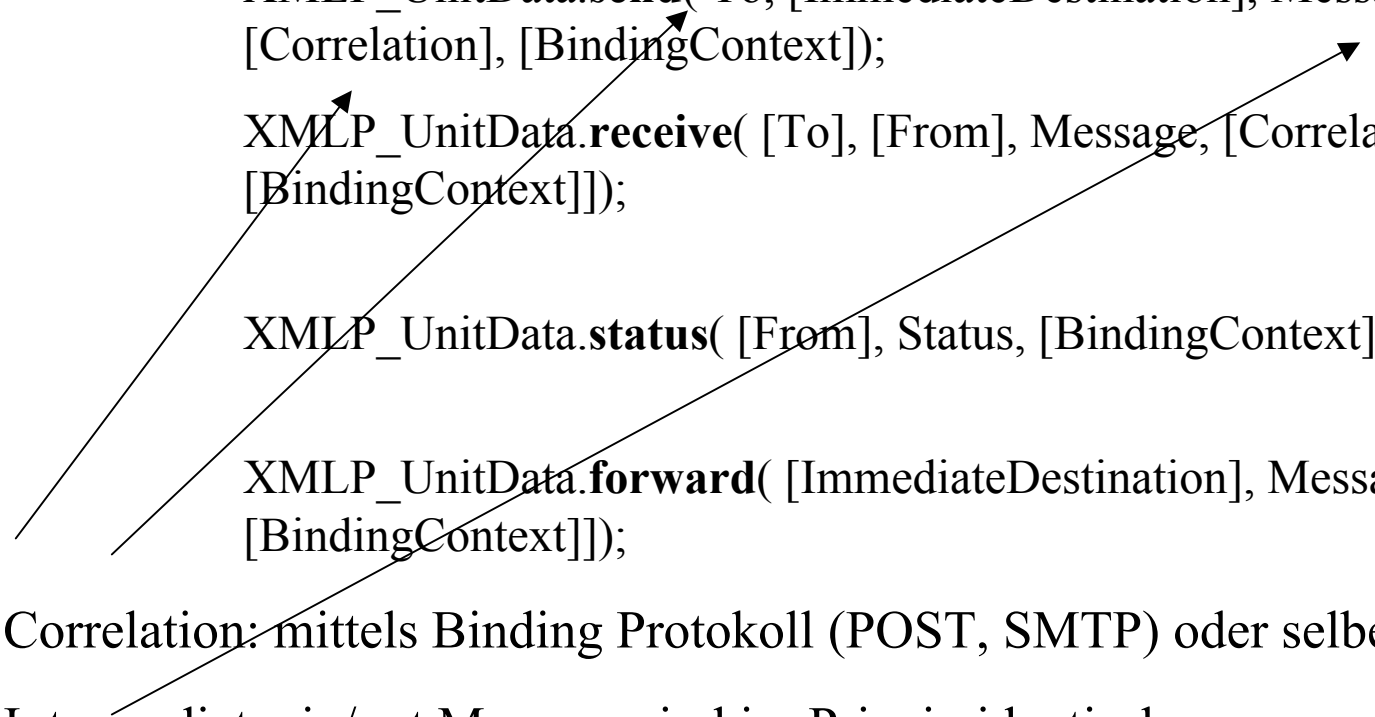
XML Protocol Layer Service Definition : XMLP_UnitData

XMLP_UnitData.**send**(To, [ImmediateDestination], Message, [Correlation], [BindingContext]);

XMLP_UnitData.**receive**([To], [From], Message, [Correlation], [BindingContext]]);

XMLP_UnitData.**status**([From], Status, [BindingContext]);

XMLP_UnitData.**forward**([ImmediateDestination], Message, [BindingContext]]);



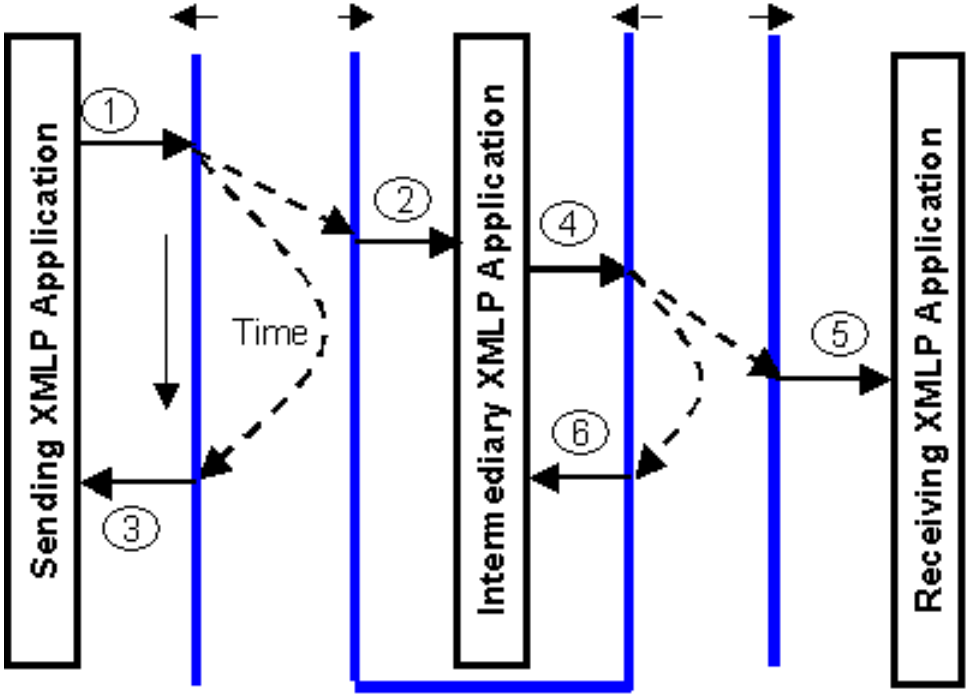
Correlation: mittels Binding Protokoll (POST, SMTP) oder selber bauen

Intermediate: in/out Message sind im Prinzip identisch

BindingContext : Protokollbezug

XML Protocol Layer Service Definition : XMLP_UnitData

Intermediatery



Layer Primitives Key

1. XMLP_UnitData.send
2. XMLP_UnitData.receive
3. XMLP_UnitData.status (any time after 1.)
4. XMLP_UnitData.forward
5. XMLP_UnitData.receive
6. XMLP_UnitData.status (anytime after 4)

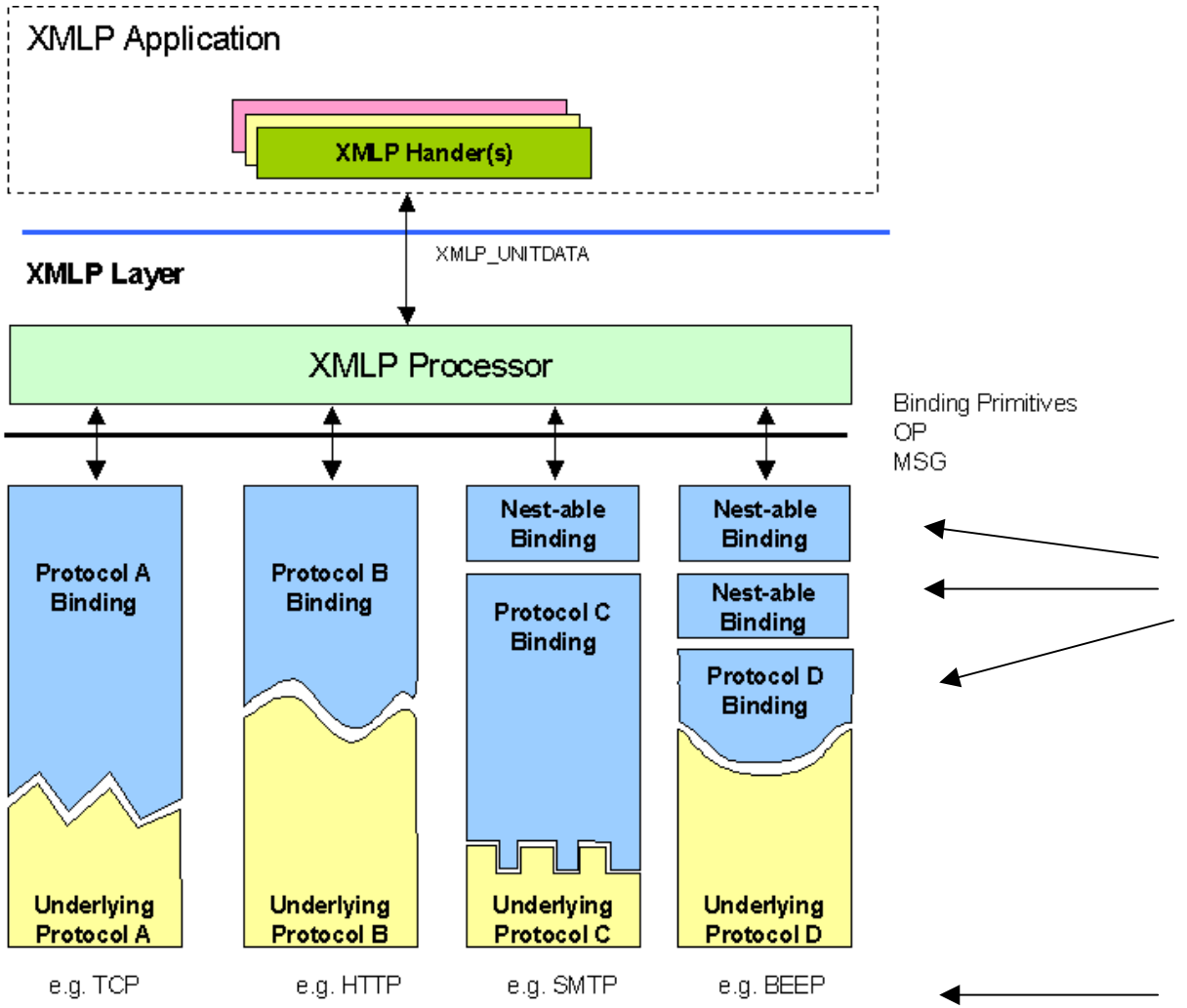
XML Protocol Layer Service Definition : XMLP_UnitData

- Typische Operation-Parameter
  - To
  - From
  - ImmediateDestination
  - Message
  - Message.Fault
  - Message.Blocks
  - Message.Attachments
  - Correlation
  - Correlation.MessageRef
  - BindingContext
  - Status

Protokoll-Bindung

- Grundsätzliches:
  - XML Protokolle sollen an unterschiedliche Kommunikations- Protokolle gebunden werden können.
  - W3C wird das HTTP Binding genauer beschreiben (als Möglichkeit)
  - TCP, SSL, BEEP, SMTP, ... sind auch gültige Varianten

Protokoll-Bindung



Architektur und Ergänzungen
Referenzen

- *Sicherheit*
  - *SSL/TLS*
    - **http://www.ietf.org/ids.by.wg/tls.html**
  - *XKMS*
    - **http://www.w3.org/TR/xkms/**
  - Digitale Signatur
    - **http://www.w3.org/2000/09/xmldsig**
  - *WS-Licence*
    - **http://msdn.microsoft.com/ws/2001/10/licence**
  - *WS-Security*
    - **http://msdn.microsoft.com/ws/2001/10/security**
  - *WS-Inspection*
    - **http://msdn.microsoft.com/ws/2001/10/inspection**

Architektur und Ergänzungen
Referenzen

- *Process Flow*
  - *WSFL*
    - **http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf**
  - *XLang*
    - **http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm**
  - Web Service Choreography (*WSCI*)
    - **http://dev2dev.bea.com/techtrack/wsci.jsp;**