

Policy-Management

Baustein einer business-orientierten Security innerhalb einer SOA

Günter Waller (IBM Deutschland GmbH)

Detlef Sturm (Beta Systems Software AG)



BITKOM, AK PRS & AK SOA

23.09.2008, Darmstadt



Teil 1**(Günter Waller)**

- ▶ **SOA und Policy-Management**
- ▶ **Konsistenz von Policies**
- ▶ **Policy Enforcement**
- ▶ **Architektur und Basiskomponenten**
- ▶ **Relevante Standards**

Teil 2**(Detlef Sturm)**

- ▶ **Handlungsebenen**
- ▶ **Policy-Definition und -Enforcement**
- ▶ **Client-Policy-Provisioning**
- ▶ **Deployment-Varianten**
- ▶ **Business-IT Alignment**



Wer interessiert sich für Policies?



- Geschäftsbereiche: Ableitung aus den Geschäftszielen, Policies in natürlicher Sprache.
 - z.B. kontrollierter Zugriff auf sensitive Daten



- Anwendungseigner: Policy als Berechtigung (Entitlement)
 - z.B. Anwendungen prüfen Berechtigungen



- IT Betrieb: Policy steckt in Konfigurationen/Einstellungen
 - Administratoren müssen Einhaltung sicherstellen, z.B. WS-Security Policy

Welche Arten von Policies bestimmen eine SOA?

- Beispiele:



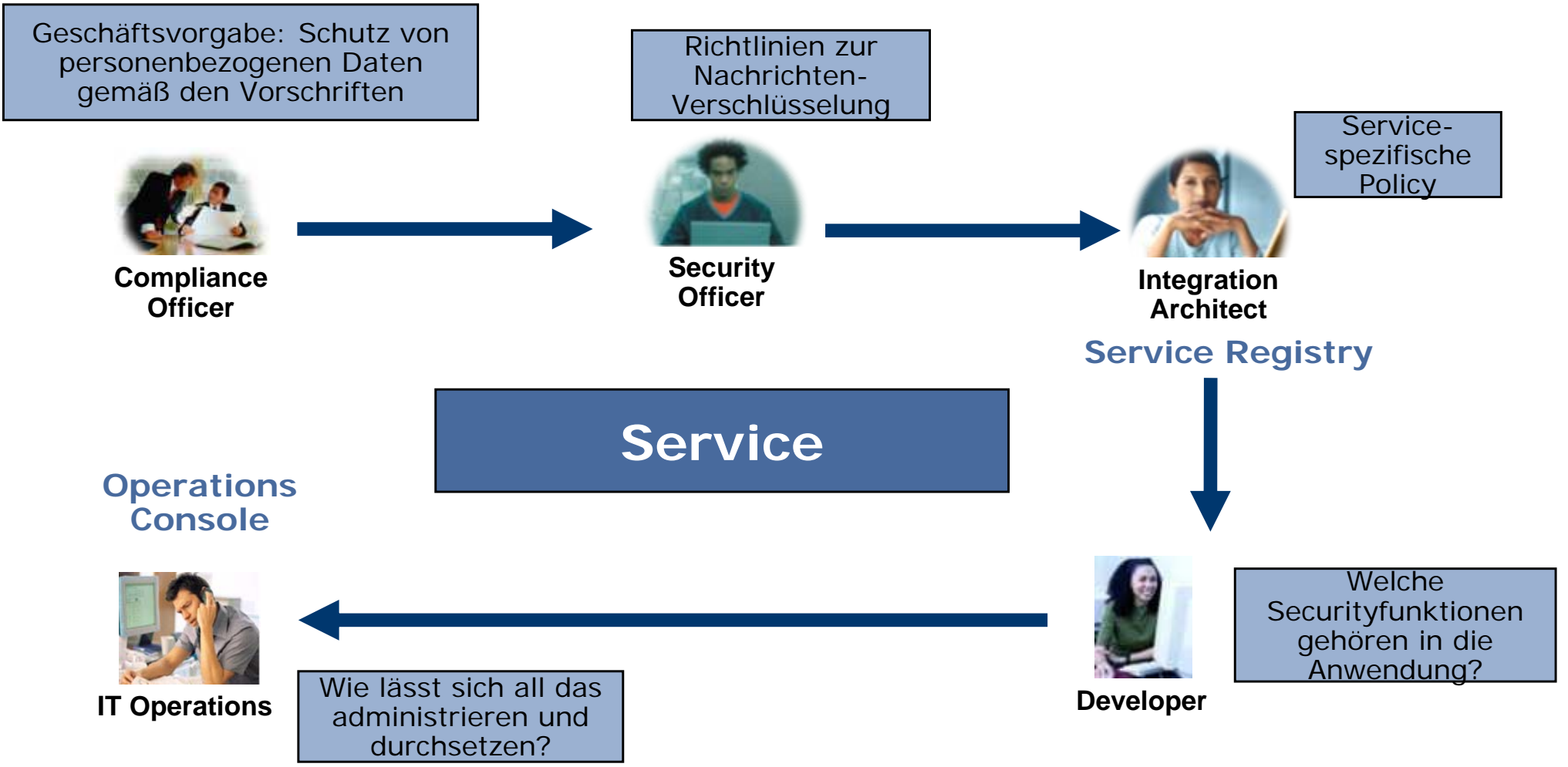
- Message Protection Policies: *Integrität und Vertraulichkeit von nachrichten*



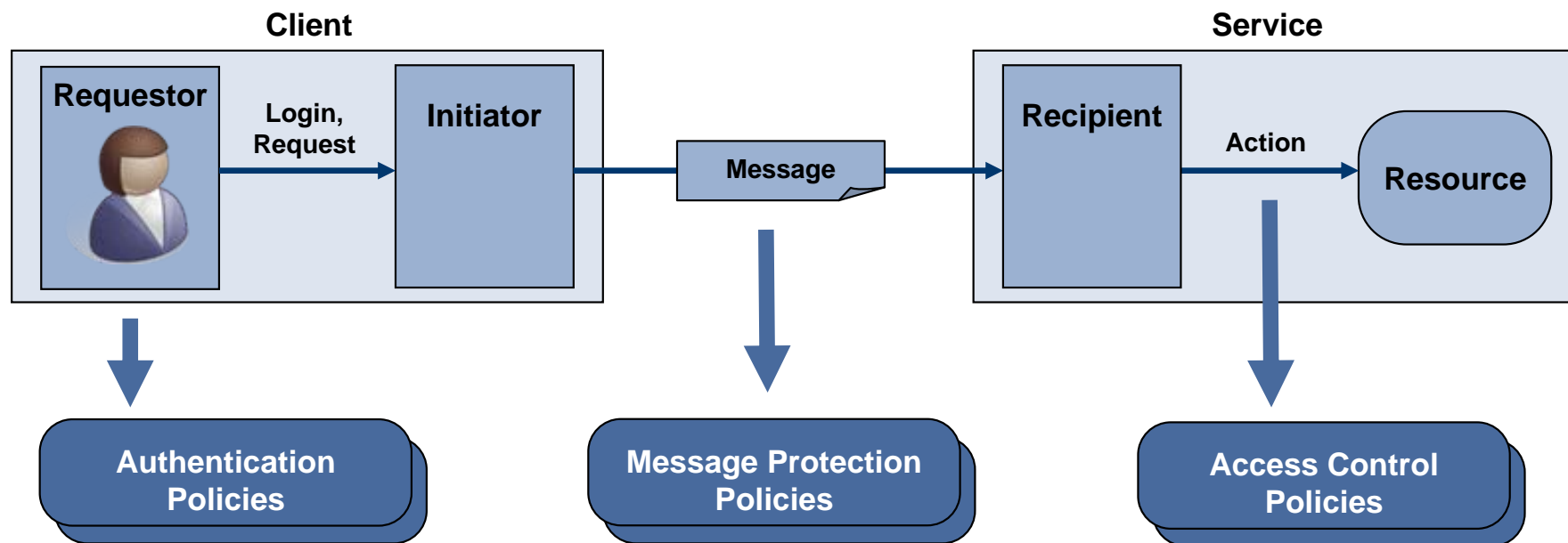
- Authorization Policies: *Wer darf worauf zugreifen? Wann und unter welchen Voraussetzungen?*



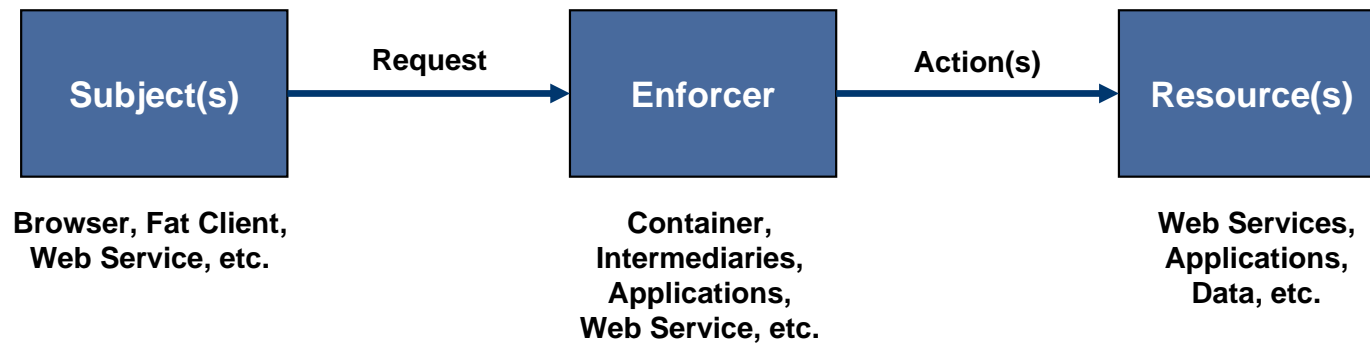
- Authentication/Identity Policies: *In welcher Form müssen die identitäten belegt werden?*



Problem: Unterschiedliche Produkte beteiligt mit jeweils spezifischen Definitionen (sowohl syntaktisch als auch semantisch). Wie lässt sich da die durchgängige Einhaltung garantieren?

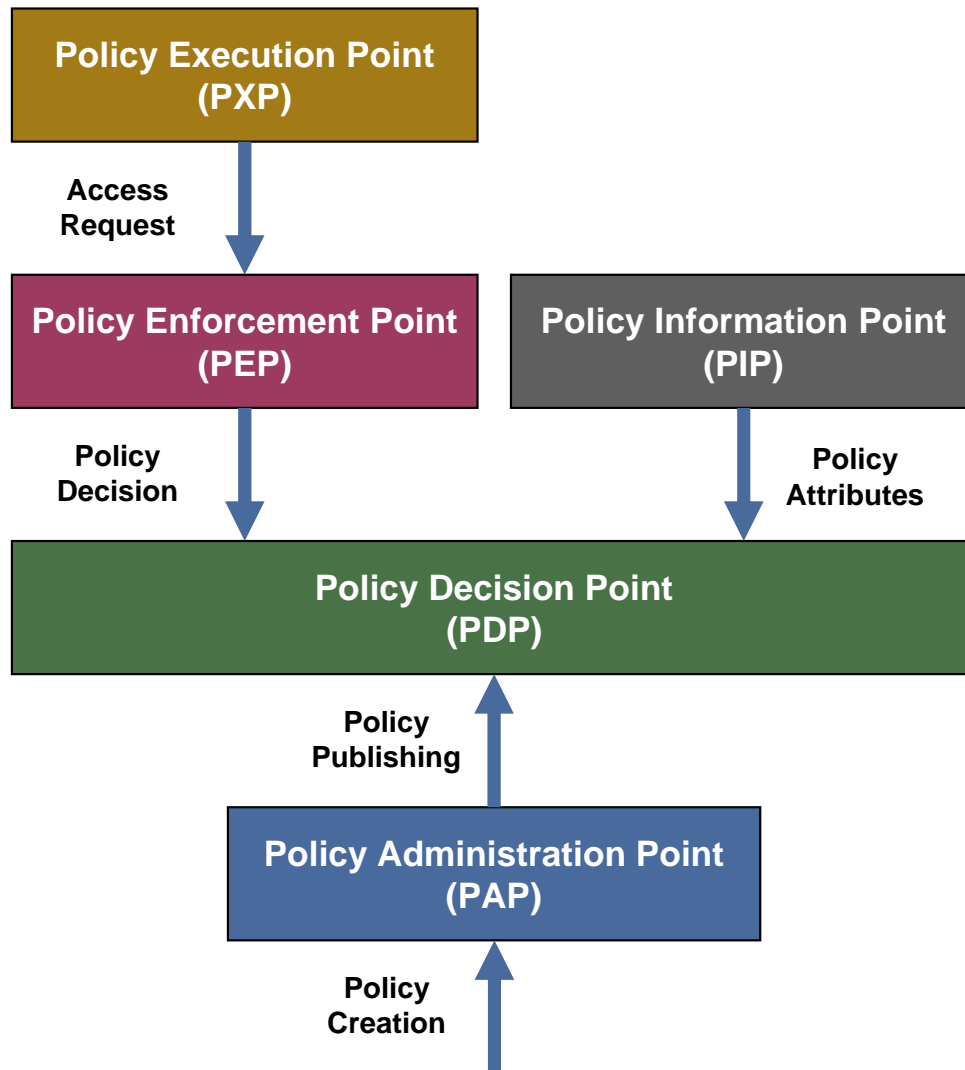


- ▶ Herausforderung:
 - ▶ Verteilte Ansatzpunkte für die einzelnen Policies
 - ▶ Dennoch zentrale Kontrolle erforderlich
- ▶ Gesucht wird ein gemeinsamer Nenner
 - ▶ Sprache zur Formulierung der Policies
 - ▶ Protokoll zur Verteilung
 - ▶ Laufzeitumgebung zur Überwachung

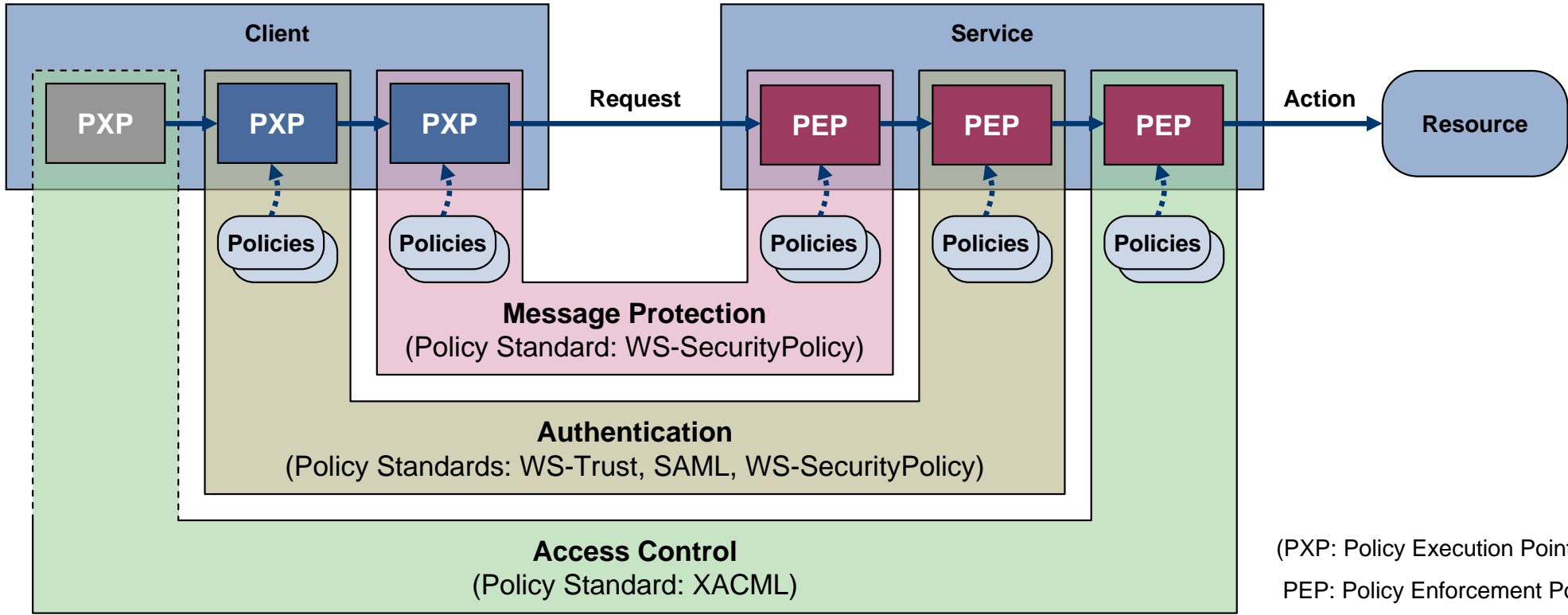


Klassisches Modell für Autorisierung (Access Control)

- ▶ **Wer** (Subject) will **wie** (Action) **was** (Resource) bearbeiten?
 - ▶ Antwort (in einfachster Form): Ja oder Nein
- ▶ Erweiterung dieses Prinzips erlaubt Abbildung beliebiger Policies
 - ▶ durch Heranziehen zusätzlicher Informationen
 - ▶ Eigenschaften des Subjekts (Rollen, Attribute)
 - ▶ Eigenschaften der Ressourcen (Grad der Sensitivität, Schutzwürdigkeit)
 - ▶ Bezug zwischen Subjekt und Objekt (personenbezogene Daten)
 - ▶ Echtzeitinformationen zum Zugriffszeitpunkt (Zeit, Ort, Qualität der Authentifizierung)
 - ▶ durch Ergänzung der Entscheidung
 - ▶ Auflagen (Obligations)
 - ▶ durch Kombination von Policies

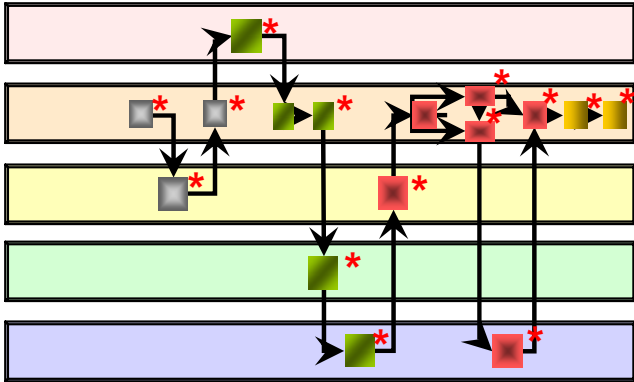
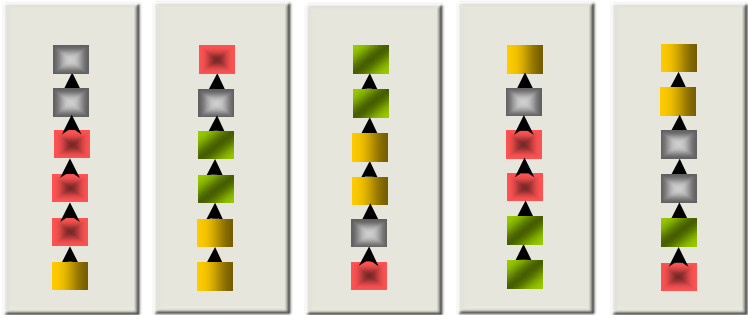


- ▶ Funktionale Darstellung
 - ▶ Spielraum für Implementierungen (Architektur)
 - ▶ Optionale Funktionen
- ▶ Administration: zentral
 - ▶ Beschreibung (einheitlich, Standard)
 - ▶ Verteilung (Profile, mehrere Standards)
- ▶ Entscheidung: Runtime, verteilt, wenige Instanzen
 - ▶ Policy Cache
 - ▶ Optional: Hinzuziehen weiterer (dynamischer) Daten
- ▶ Enforcement: Architekturentscheidung
 - ▶ Umgehung (Bypass) verhindern
 - ▶ Abhängig vom Szenario/Systemumgebung
- ▶ Execution: Intelligenter Client „spielt mit“
 - ▶ Zuerst Policy-Abfrage
 - ▶ Konfiguriert sich dynamisch
 - ▶ Servicebetreiber hat keinen Einfluss

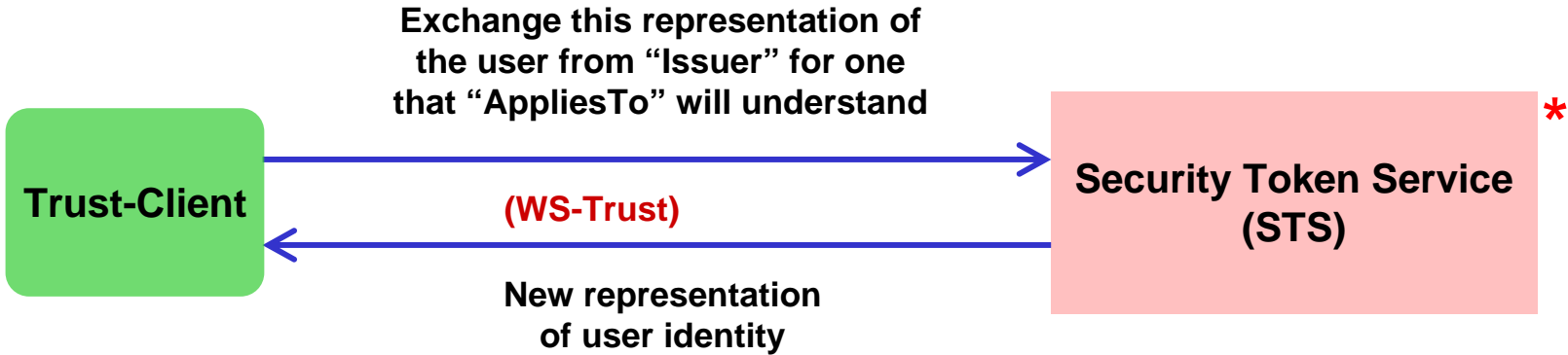


- ▶ In einer komplexen SOA kommt das Problem der wechselnden Identitäten hinzu
 - ▶ Umwandlung/Mapping von Identitäten: WS-Trust

- SOA: Aufbrechen der Silos



- WS-Trust: Standard zur Umwandlung von Identitäten
 - Trust-Client kann WS-Client oder WS-Service sein.



* = Potentieller Policy Enforcement Point (PEP)

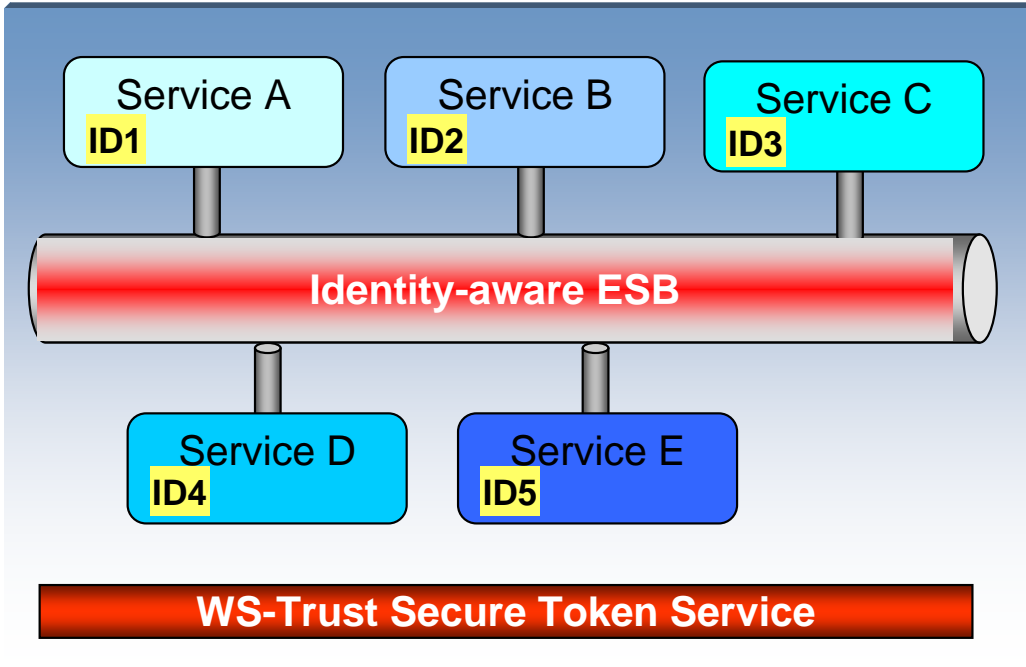


Policy Execution Point (PXP)

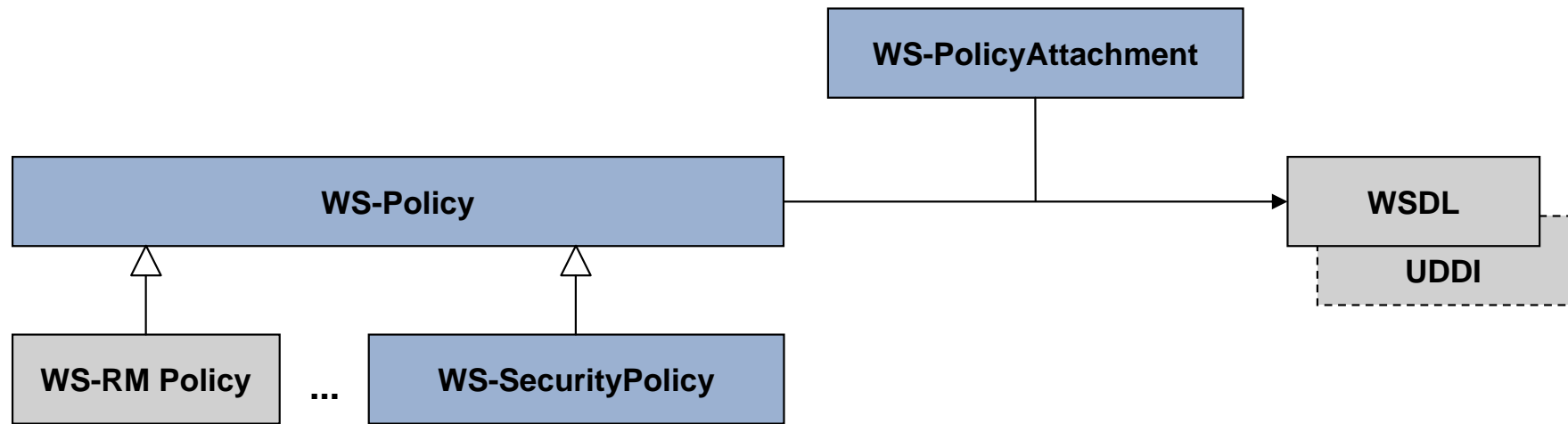
Policy Execution Point (PXP)

Policy Execution Point (PXP)

Policy Enforcement Point (PEP)

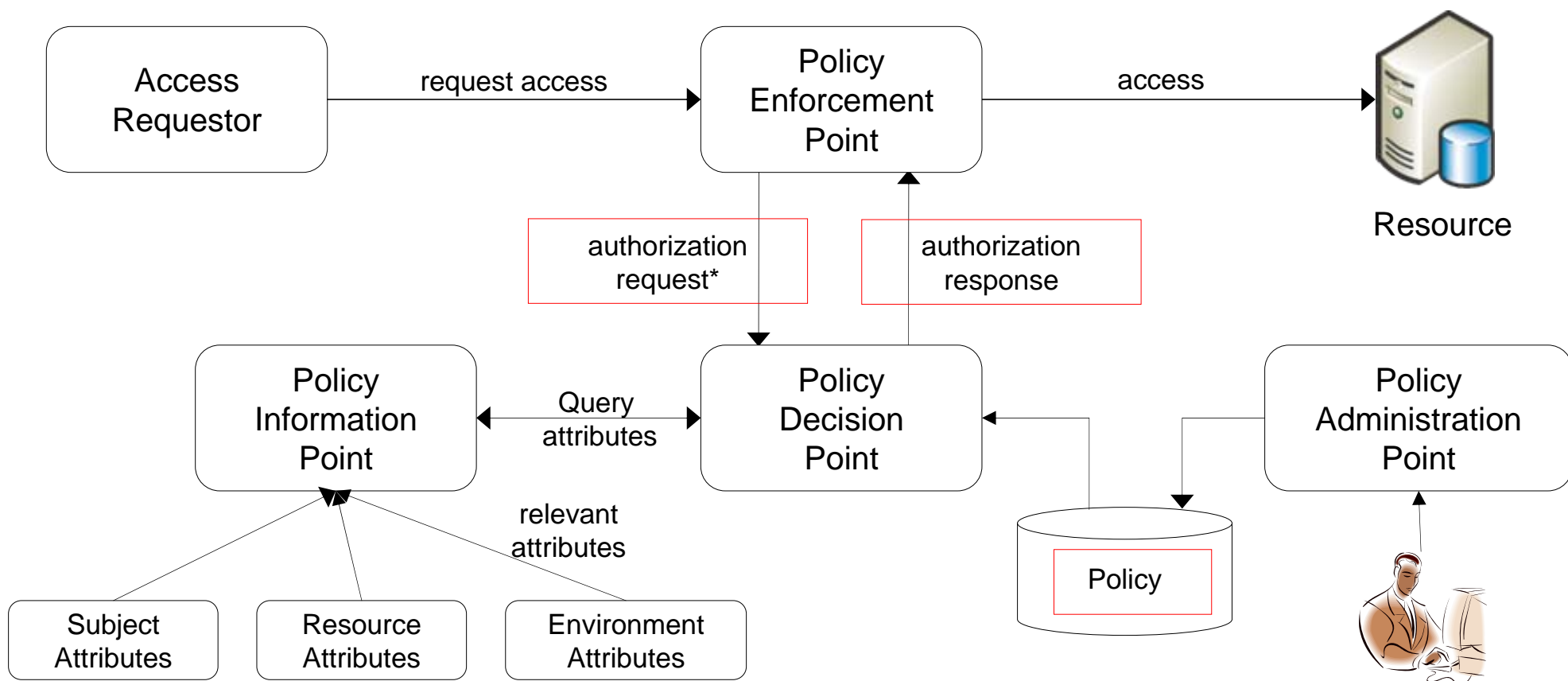


- ▶ WS-Policy
 - ▶ Abstrakter Rahmen (Syntax) zur Charakterisierung eines (Web) Service
 - ▶ Anforderung an den Client
 - ▶ Fähigkeiten des Service
 - ▶ besteht aus Assertions
 - ▶ Zuordnung zum Service über UDDI oder WSDL
 - ▶ Die Semantik der Assertions wird in anderen Standards beschrieben (z.B. WS-Security Policy)
- ▶ WS-Security Policy
 - ▶ Syntax und Semantik von Policy Assertions (Anwendung von WS-Security, WS-Trust, WS-Secure Conversation)
 - ▶ Kategorien: Protection, Tokenformate, Security Binding, Token Assertions



- ▶ OASIS XACML Committee
- ▶ Version 2.0 (2005)
- ▶ Ziel: Autorisierungsentscheidung
 - ▶ Kontext- (bzw. attributs-) basierte Entscheidung
- ▶ Umfang:
 - ▶ Policy Language
 - ▶ Request/Response-Nachrichten
- ▶ Verallgemeinerung sowohl von Role-based als auch ACL-based Access Control
- ▶ Decision Request: Subjekt, Ressource, Aktion, Environment (optional)
- ▶ Policies:
 - ▶ Policy Set
 - ▶ Policy
 - ▶ Regeln (Deny, Permit)
 - ▶ Algorithmus (Rule Combining: Deny-overrides, Permit-overrides, First-applicable, Only-one-applicable)
 - ▶ Target (relevante Attribute aus Resource/Subject/Action)
 - ▶ Obligation (Auflagen, „bedingte Autorisierung“)
- ▶ Operationales Modell





- ▶ XACML definiert die “authorization request/response” Message Formate *und* das Policy Format
 - ▶ Verschiedene *Profile* beschreiben unterschiedliche request/response Formate
 - ▶ z.B. SAML Profil

Teil 1

(Günter Waller)

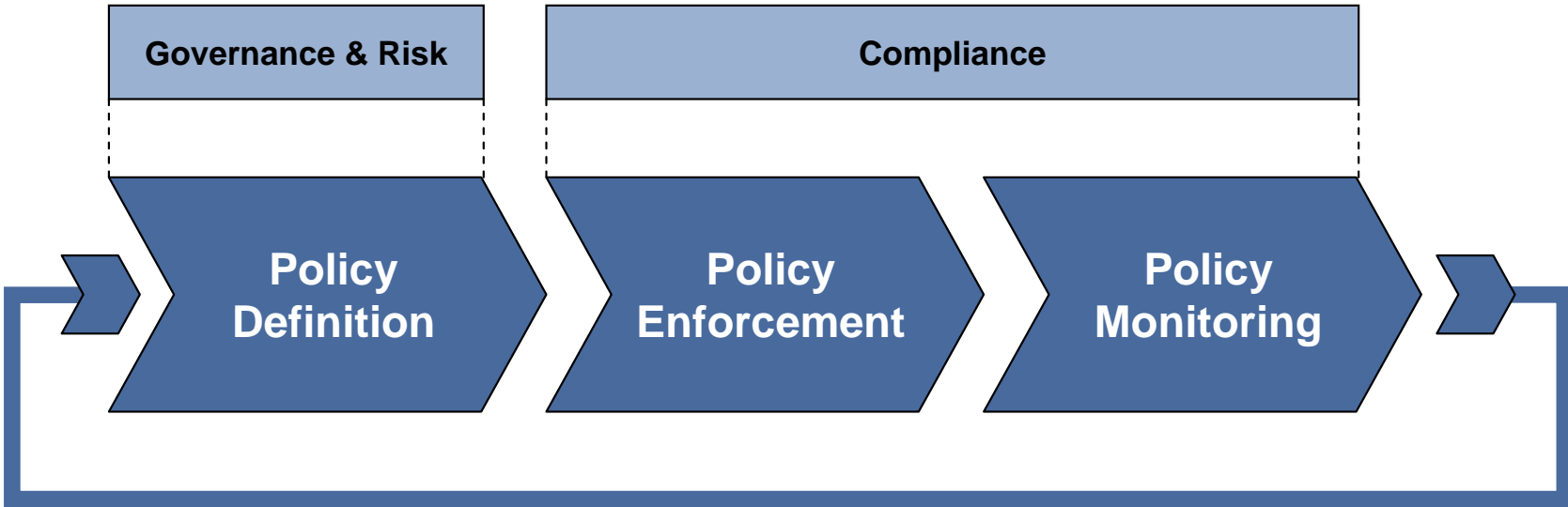
- ▶ SOA und Policy-Management
- ▶ Konsistenz von Policies
- ▶ Policy Enforcement
- ▶ Architektur und Basiskomponenten
- ▶ Relevante Standards

Teil 2

(Detlef Sturm)

- ▶ Handlungsebenen
- ▶ Policy-Definition und -Enforcement
- ▶ Client-Policy-Provisioning
- ▶ Deployment-Varianten
- ▶ Business-IT Alignment

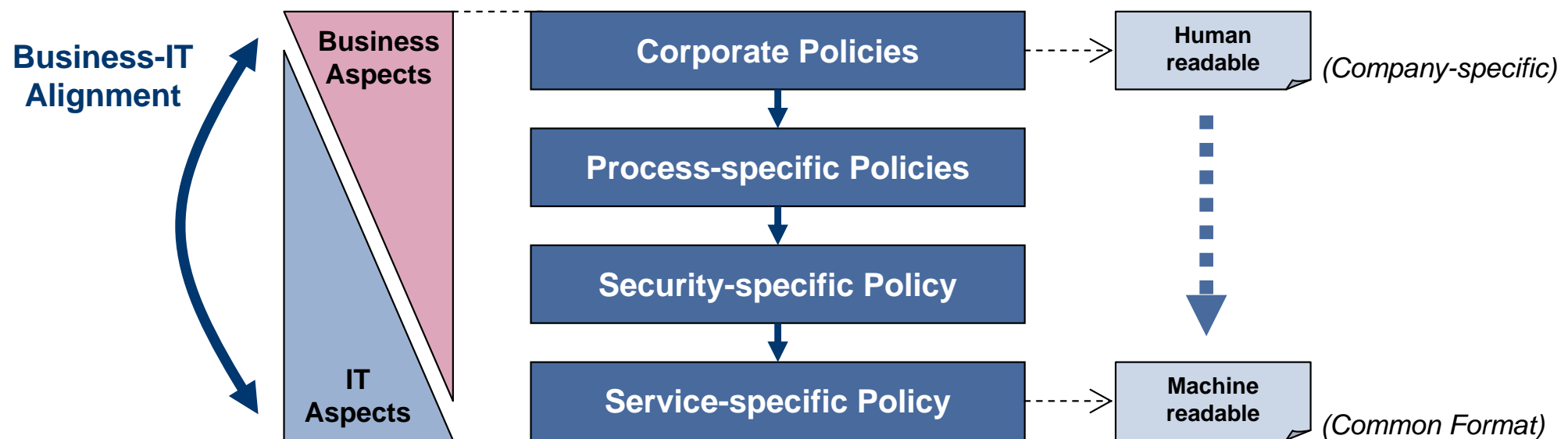




Policy Definition
<ul style="list-style-type: none">▶ Erstellung der Richtlinien▶ Orientierung an den geschäftlichen Anforderungen▶ Aufbereitung der Richtlinien für das Policy-Enforcement▶ Governance-relevant:<ul style="list-style-type: none">- Vorgaben von unternehmensweiten Richtlinien

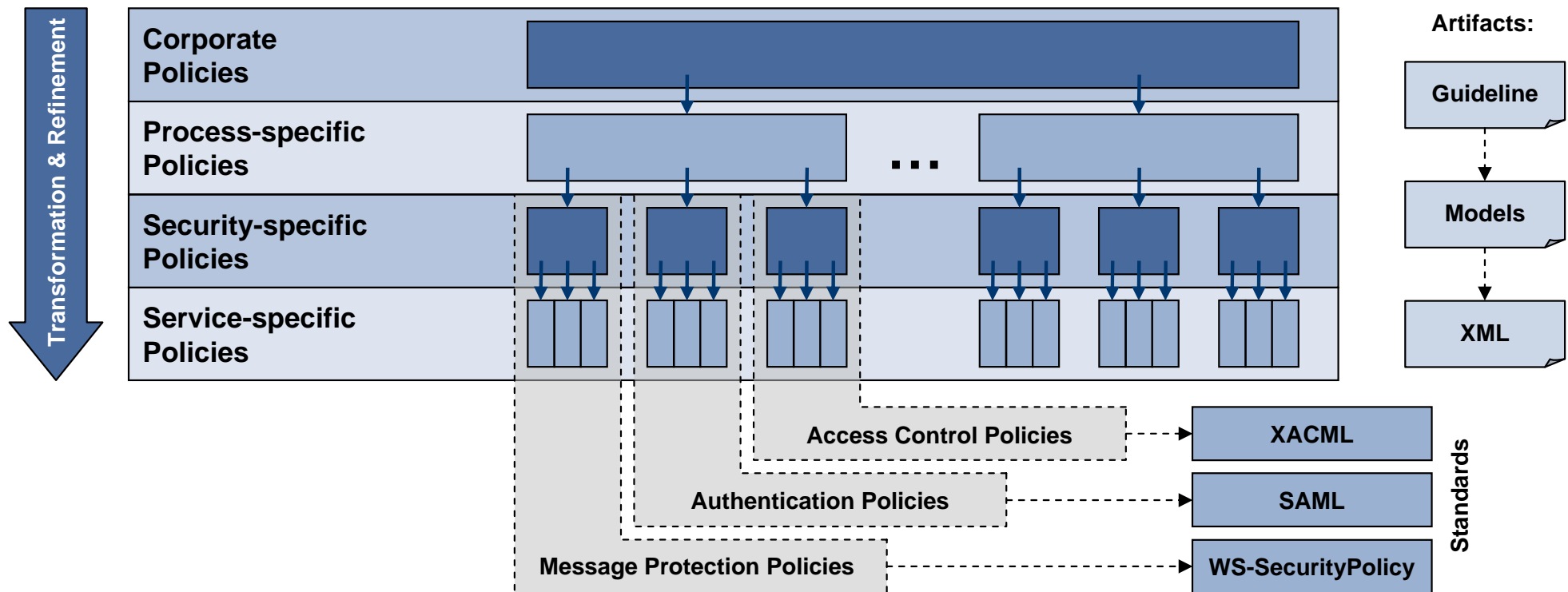
Policy Enforcement
<ul style="list-style-type: none">▶ Durchsetzung der Richtlinien in den Applikationen / Services▶ IT-getriebene und technologieabhängige Phase▶ Hoher Standardisierungsgrad▶ Ansatz für Security-as-a-Service▶ Compliance-relevant

Policy Monitoring
<ul style="list-style-type: none">▶ Überprüfung der Durchsetzung der Richtlinien im operativen Betrieb▶ Kontrolle bzgl. der Wirksamkeit▶ Rückkopplung auf die Policy-Definition▶ Compliance-relevant

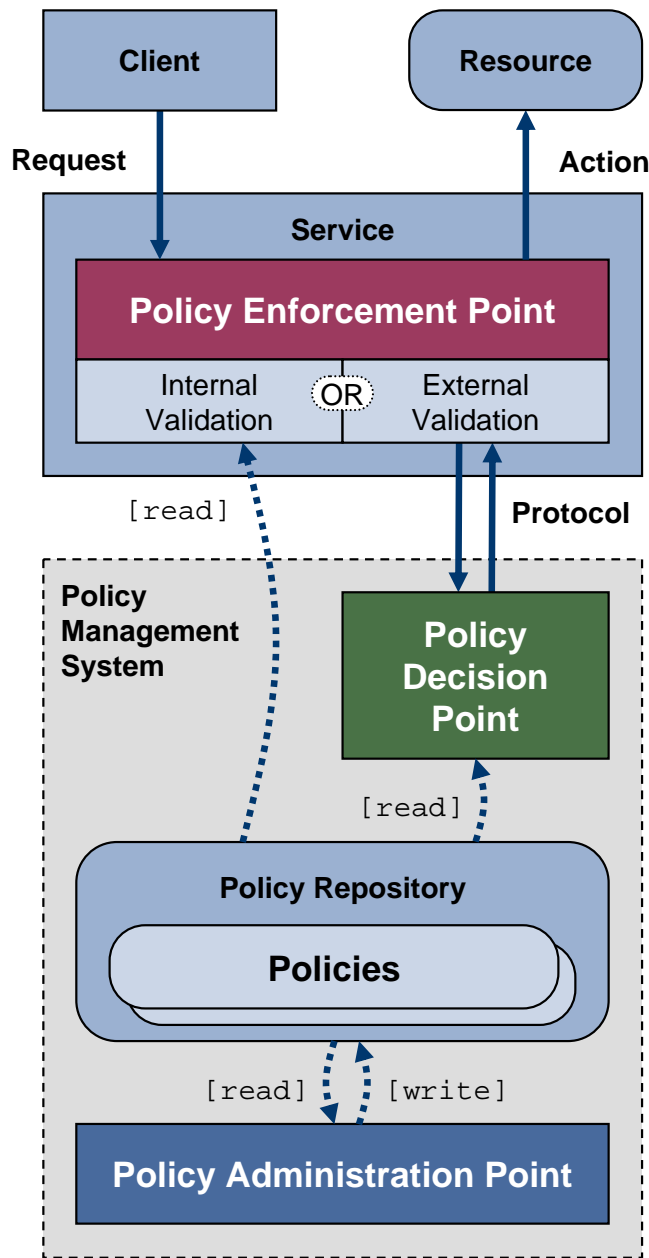


► Inkrementeller Ansatz:

- Unternehmensrelevante Richtlinien → Service-spezifische Richtlinien
- Allgemeingültige Aussagen → Konkrete Aussagen (Verfeinerung)
- Unternehmensspezifischen Formate → Standardisierte Formate
- Ergebnis: Hierarchie von Richtlinien
 - Jede Ebene adressiert den entsprechenden Stakeholder (Rolle im Unternehmen):
Compliance Office → Security Office → Integration Architect → IP Operations
- Herausforderung im Rahmen des Business-IT Alignment: Übergänge zwischen den einzelnen Sichtweisen



- ▶ Schrittweise Verfeinerung: Identifizierung von Subjekten, Aktionen, Ressourcen und Attributen
 - ▶ Top-down vs. Bottom-up
- ▶ Transformation: Syntaktische Umwandlung (z.B. UML-Modell → XML-basiertes Format)
 - ▶ Automatisierbar
- ▶ Übersetzung (Translation): Umwandlung von syntaxfreien Formaten (Sprache, freier Text)
 - ▶ i.d.R. manueller Vorgang → Kluft zwischen den verschiedenen Sichten
- ▶ Ergebnis: IT-interpretierbare Richtlinien, gruppiert nach den jeweiligen Sicherheitsverfahren



- ▶ Policy Enforcement
 - ▶ Komponente, die die Richtlinien durchsetzt
 - ▶ Ermöglicht die Auslagerung Richtlinienüberprüfung
 - ▶ Entkopplung: Security und Applikationslogik

- ▶ Varianten der Richtlinienüberprüfung
 - ▶ Intern: mit direktem Zugriff auf die Richtlinien
 - ▶ Typisch für Nachrichtenschutz (WS-SecurityPolicy)
 - ▶ Extern: PDP übernimmt Entscheidungsfindung
 - ▶ Typisch für Zugriffskontrolle (XACML)

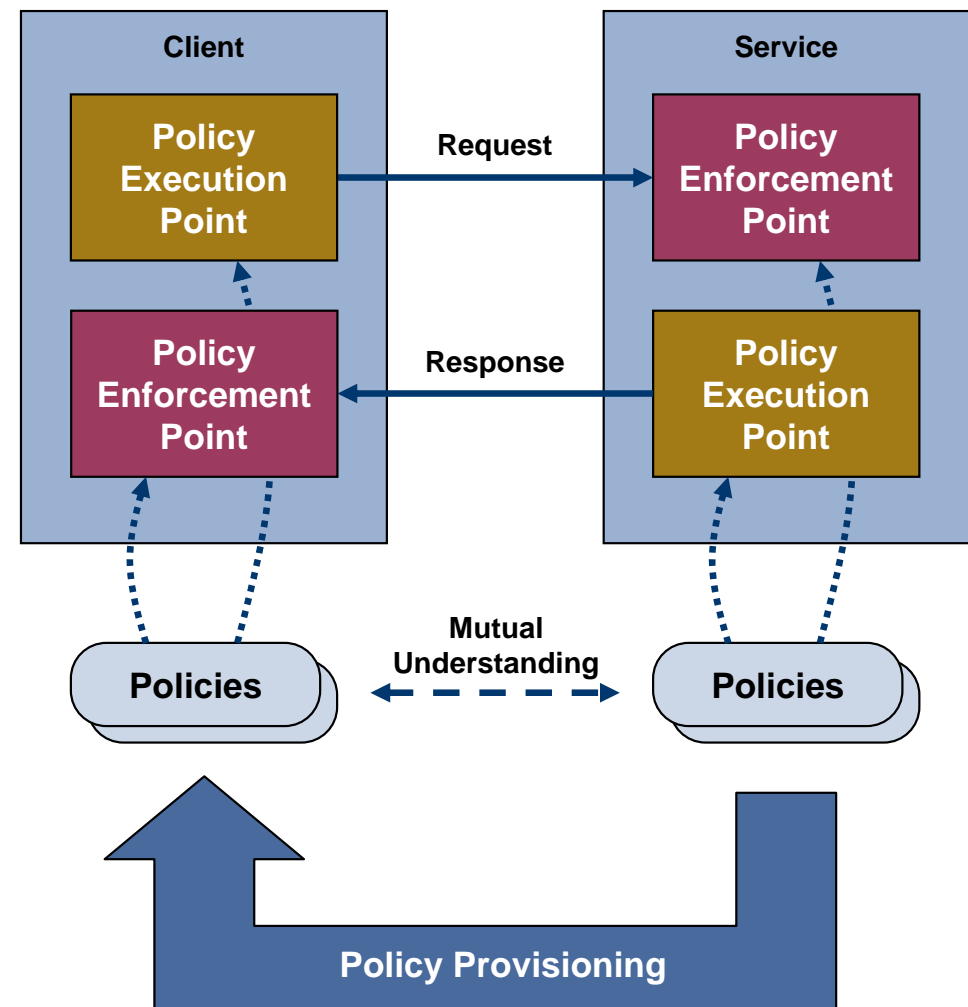
- ▶ Policy Decision
 - ▶ Autorisierungs-Service (Security as a Service)
 - ▶ Erfordert Protokoll zwischen PEP und PDP

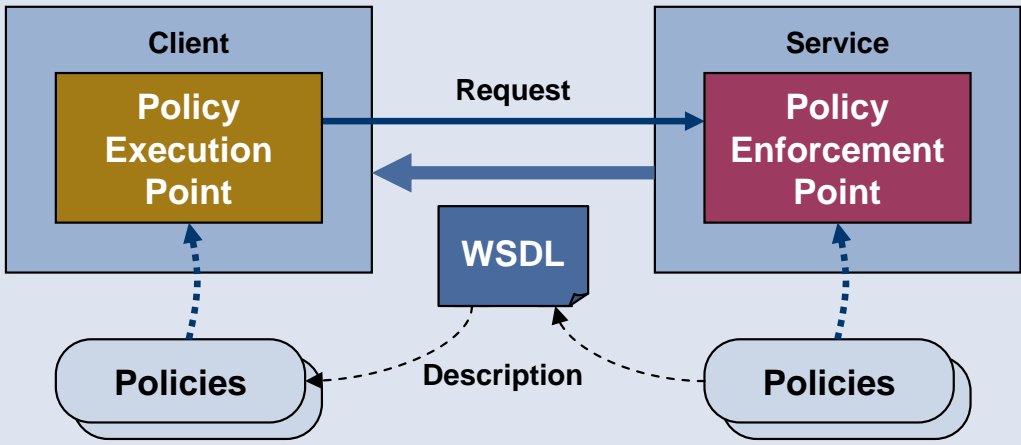
- ▶ Policy Administration
 - ▶ Besonderer Zugriffsschutz
 - ▶ Freigabeprozesse: Change Management und Versionierung

- ▶ Service-seitiges Policy-Enforcement
 - ▶ Überprüft den Client-Request bzgl. der Einhaltung der Richtlinien
 - ▶ Erfordert, dass der Client die Richtlinien kennt

- ▶ Client-Policy-Provisioning
 - ▶ Bereitstellung der Richtlinien
 - ▶ Gemeinsames Verständnis
 - ▶ Konsistente Security-Einstellungen
 - ▶ Unterschiede für interne und externe Clients
 - ▶ Policy Execution Point (PXP) übernimmt die „Anwendung“ der Richtlinien

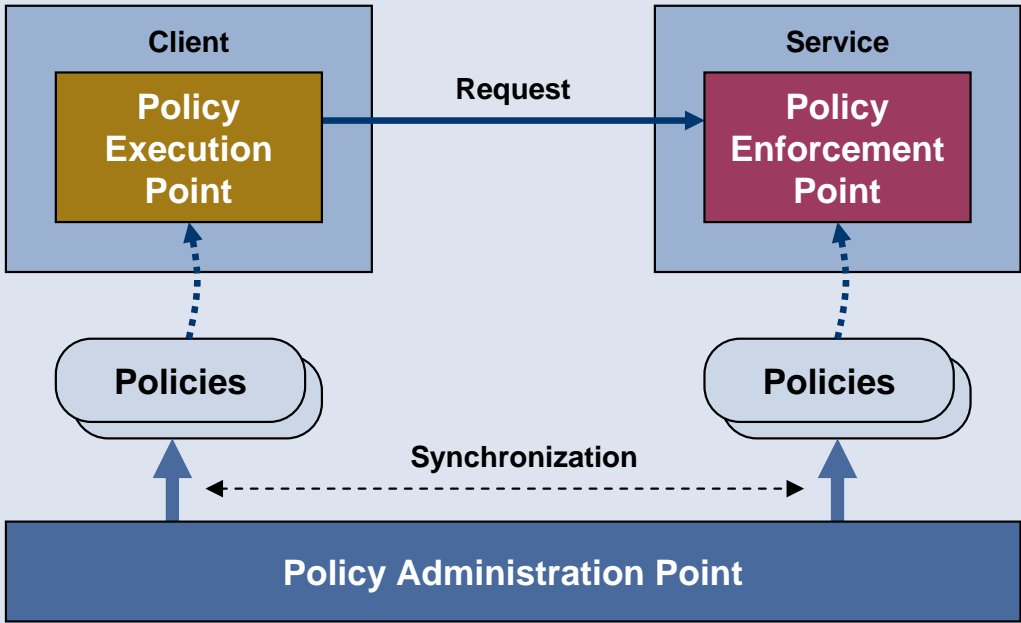
- ▶ Client-seitiges Policy-Enforcement
 - ▶ Überprüfung von Response-Nachrichten (z.B. Kriterien für den Nachrichtenschutz)
 - ▶ Rollentausch:
 - ▶ Client: PEP, Service: PXP





WSDL-based

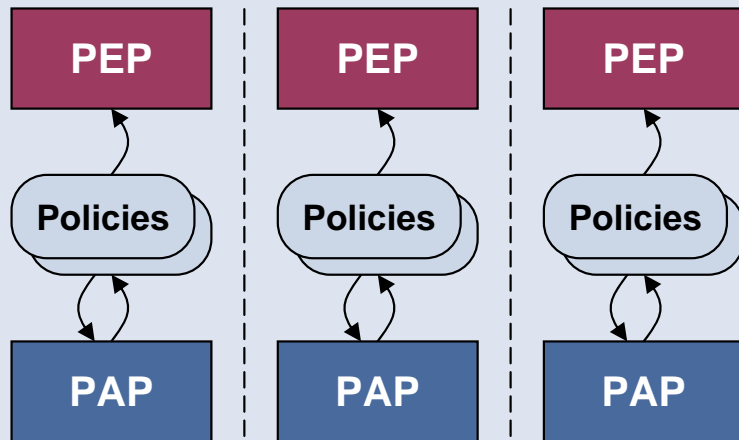
- ▶ WSDL dient der Schnittstellenbeschreibung
 - ▶ Operationen, Protokoll, Adressen
 - ▶ Erweiterbar mit Policy-Assertions (z.B. WS-SecurityPolicy)
- ▶ **Problematisch:**
 - ▶ WSDL-Abfrage nur in der Entwicklungs- bzw. Konfigurationsphase



PAP-controlled

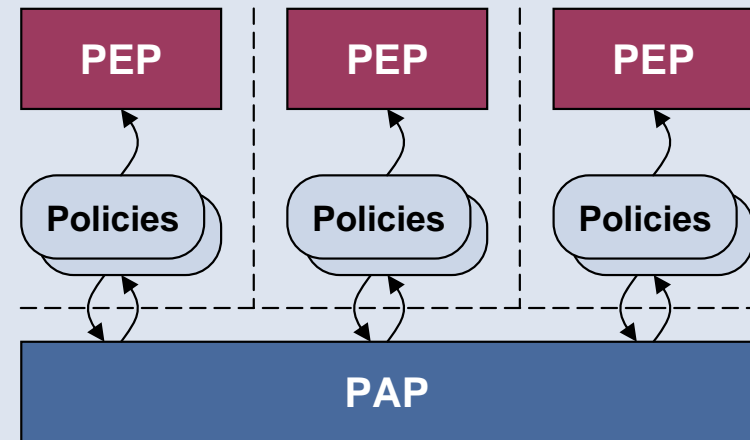
- ▶ Policy-Management-Systeme unterstützen auch die Administration und Verteilung der Client-seitigen Richtlinien
 - ▶ Übernimmt die Synchronisation der Richtlinien
- ▶ **Problematisch:**
 - ▶ Anwendbarkeit bei externen Client
 - ▶ Verteilungsaufwand bei sehr vielen Client

(1) Initial Situation



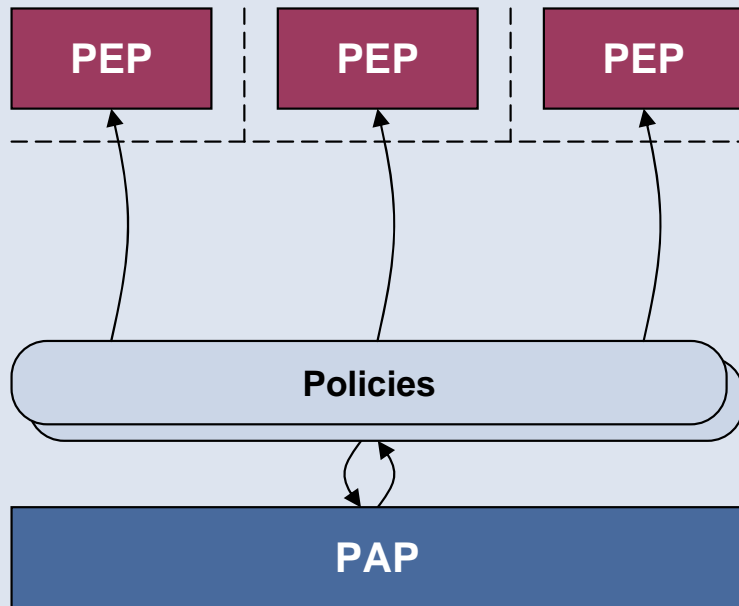
- ▶ Service und Richtlinien befinden sich auf der gleichen Maschine bzw. Container
- ▶ Für jeden Service (PEP) eine separate Administration
- ▶ Erheblicher Aufwand für Administration
 - ▶ z.B. für die Einstellung einer einheitlichen Zugriffskontrolle für einen Business-Prozess
- ▶ Ausgangslage, bevor ein Policy-Management-System zum Einsatz kommt

(2) Single Point of Administration



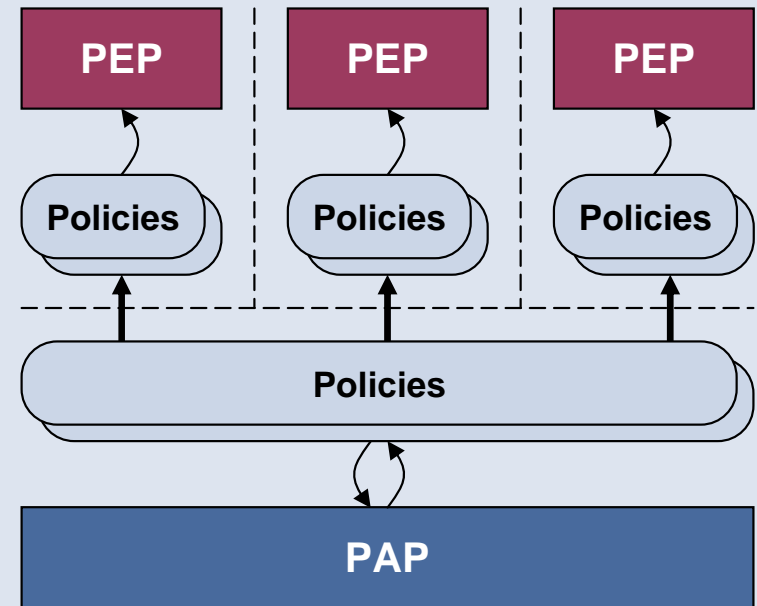
- ▶ Zentrale Administration der lokalen Policy-Repositories
- ▶ PAP übernimmt die Synchronisation der service-übergreifenden Richtlinien

(3) Central Policies



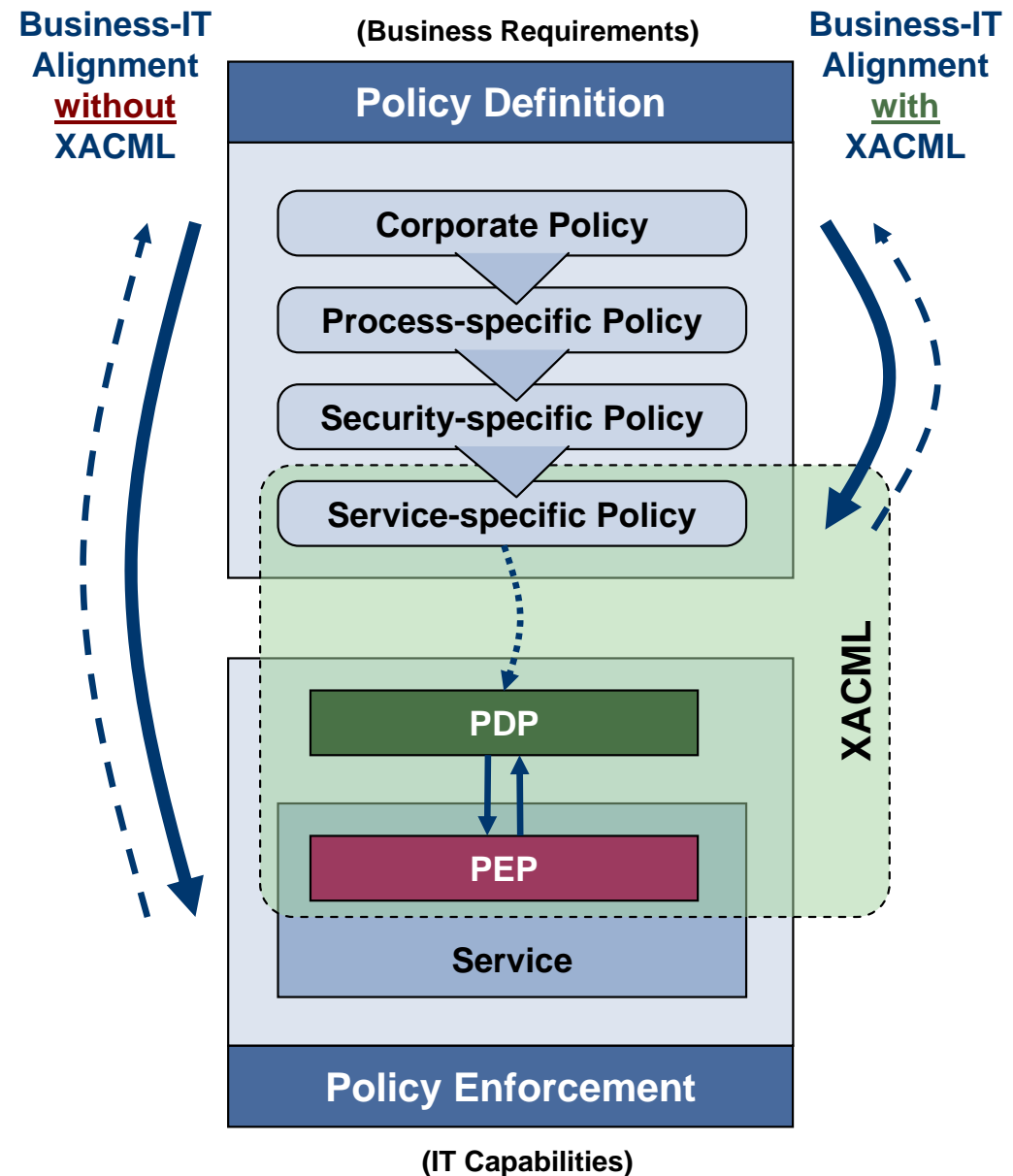
- ▶ Zentrales Repository für alle Richtlinien
- ▶ Jeder Enforcement-Vorgang benötigt Netzwerkzugriffe
 - ▶ Führt ggf. zu Performance-Probleme
 - ▶ Probleme in der Skalierbarkeit
 - ▶ Kein Offline-Modus möglich
- ▶ Remote-Zugriff erfordert zusätzliche Sicherheitsmaßnahmen

(4) Centralized and Local Copy



- ▶ Zentrale Richtlinien werden auf die lokalen Repositories verteilt
 - ▶ Keine direkte Netzwerkabhängigkeit
 - ▶ Offline-Modus möglich
 - ▶ Verteilungsvarianten: Push- oder Pull-Prinzip
- ▶ Erfordert zusätzliche Sicherheitsmaßnahmen zur Verhinderung von Manipulationen an den lokalen Daten (Last-mile Security)

- ▶ Policy-Definition: Verschiedene Sichten auf die Richtlinien (Stakeholders, Artefakte)
- ▶ Policy-Enforcement: IT-relevanter Teil des Policy-Managements
- ▶ Business-Nutzen mit XACML
 - ▶ Zentrale Verwaltung der Richtlinien
 - ▶ Service-übergreifende Zugriffskontrolle
 - ▶ Entkopplung der Zugriffskontrolle von der Applikationslogik
 - ▶ Standard-Schnittstellen (Lego-Prinzip)
 - ▶ Architekturmodell: Rollenverteilung
- ▶ Business-IT Alignment
 - ▶ XACML verkleinert die Kluft zwischen Business und IT
 - ▶ Herausforderung: Transformation und Übersetzung der Business-orientierten in die IT-orientierten Policies



Vielen Dank für die
Aufmerksamkeit

