Vorlesung Wintersemester 2011/12

Konzepte und Anwendung von Workflowsystemen

Kapitel 7:

Workflow Modellierung mit BPMN

Teil 2: Modellierung von Sicherheit und Privatheit

Lehrstuhl für Systeme der Informationsverwaltung, Prof. Böhm

Institut für Programmstrukturen und Datenorganisation (IPD)

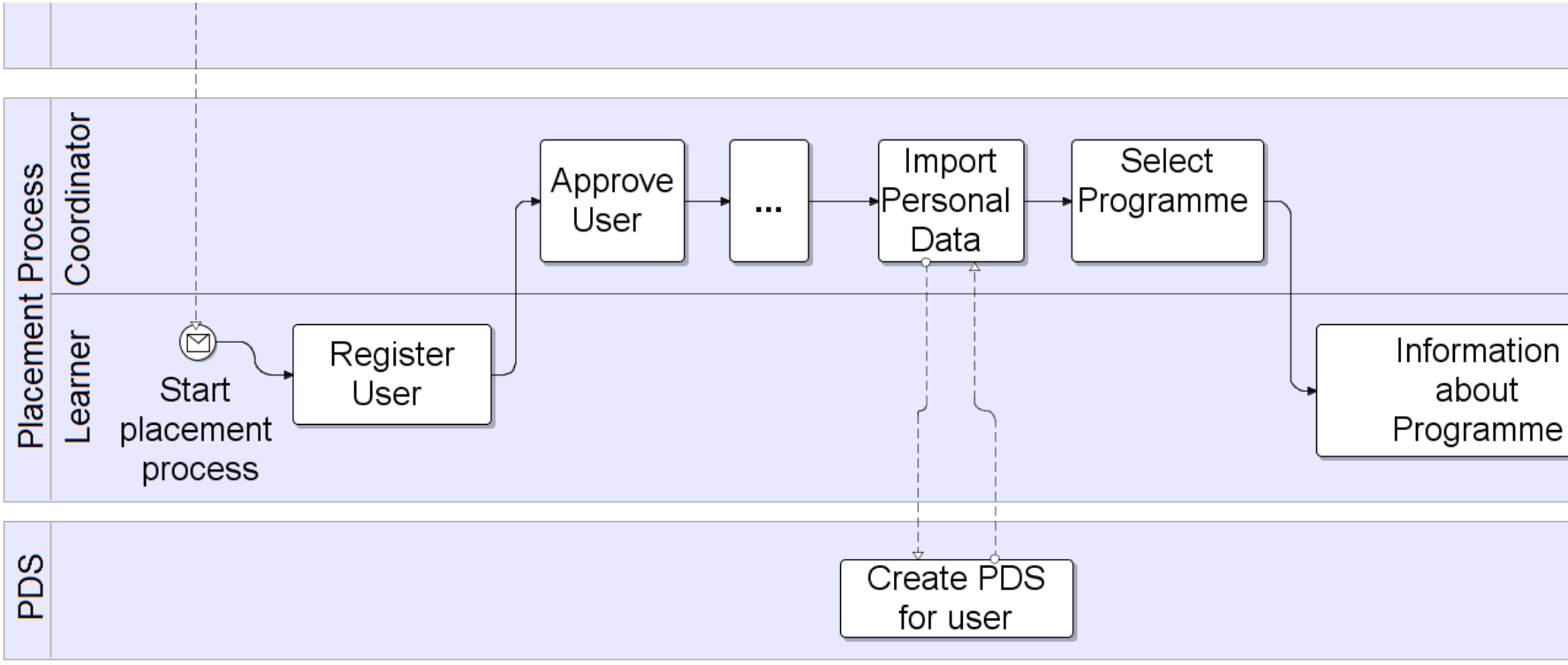
Überblick Kapitel 7 – Teil 2

- Wiederholung BPMN
- Motivation am Beispiel „Praktikumsvermittlung“
- Umsetzung von Sicherheit und Privatheit in BPMS
  - Überblick Sicherheit und Privatheit
  - Annotationen für
    - Sicherheit
    - Privatheit
  - Transformation der Annotationen
- Zusammenfassung
- Übung
  - Übungsaufgabe („Hausaufgabe“ mit Incentive)
  - Test (freiwillig)

Wiederholung BPMN

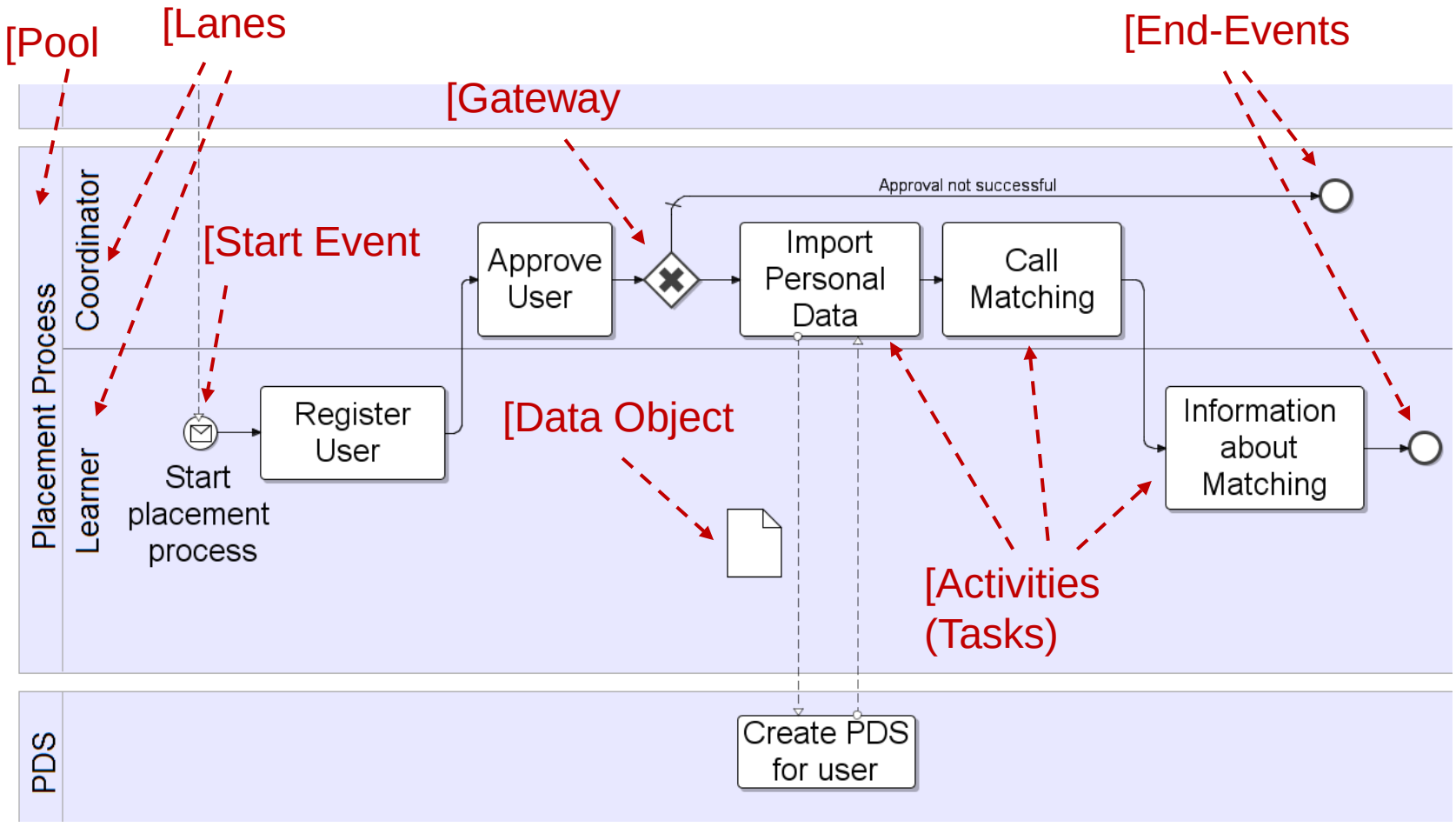
- Verbreiteter Standard (OMG) zur Modellierung von Prozessen (Workflows)
- Beschreibt die Notation und Semantik von Prozessen sowie der Kommunikation zwischen Prozessen
  - Graphische Modellierung von Prozessen als sogenannte „Diagramme“
- Sowohl für fachliche als auch für technische Modelle

Einführungsbeispiel Workflow
- Teilablauf einer Praktikumsvermittlung -

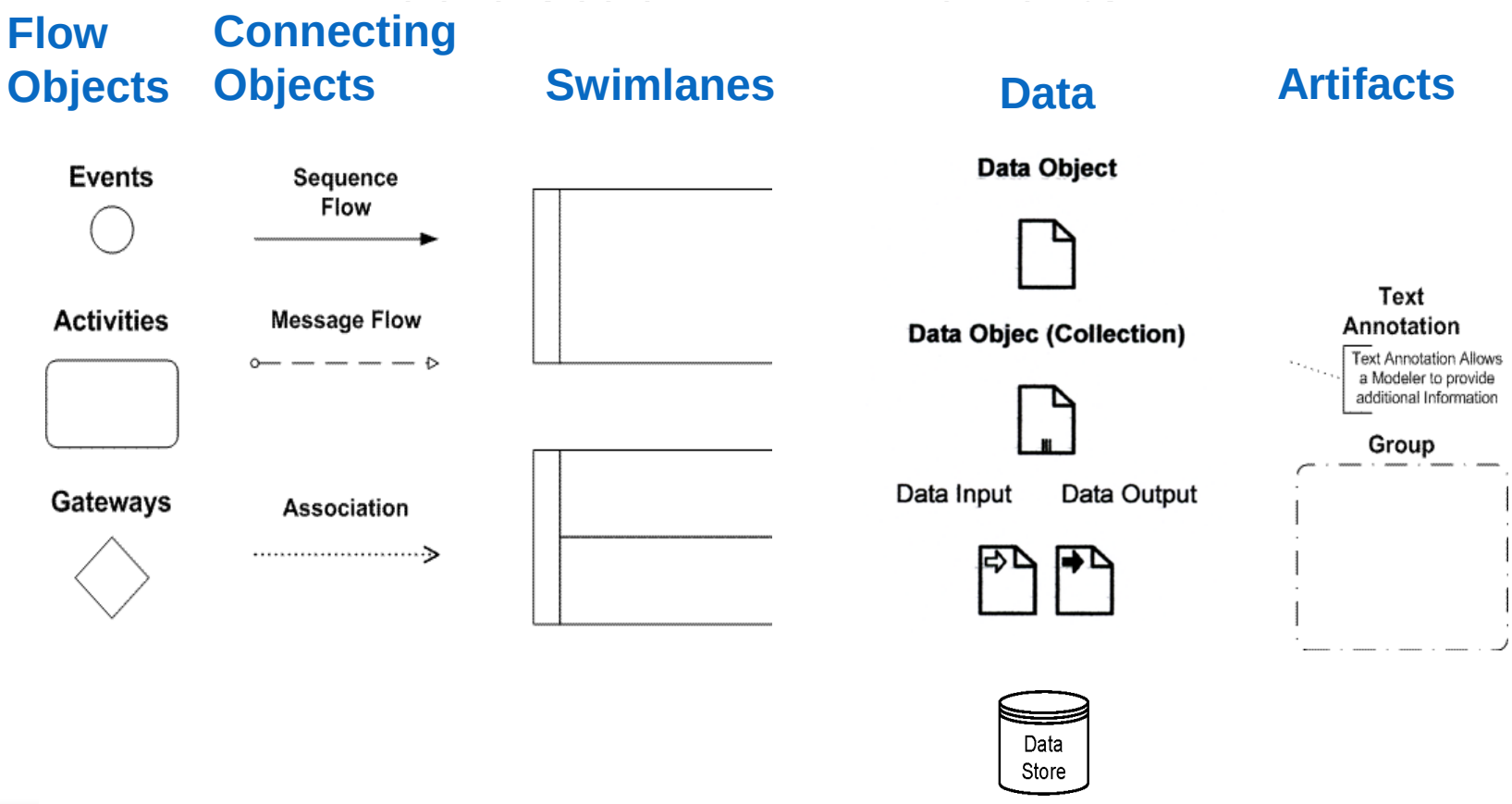


PDS: Personal Data Store

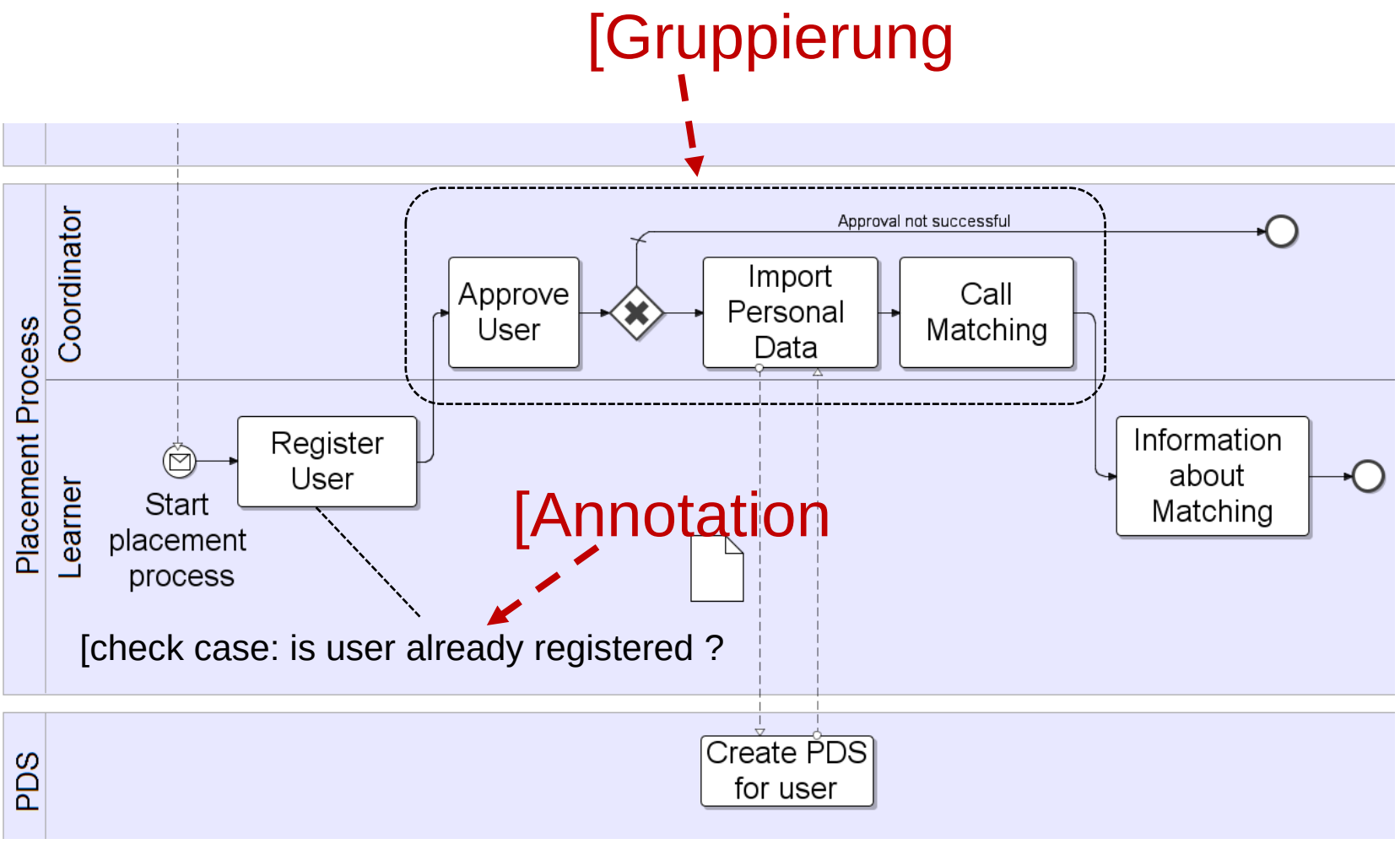
Beispiel BPMN
– Praktikumsvermittlung -



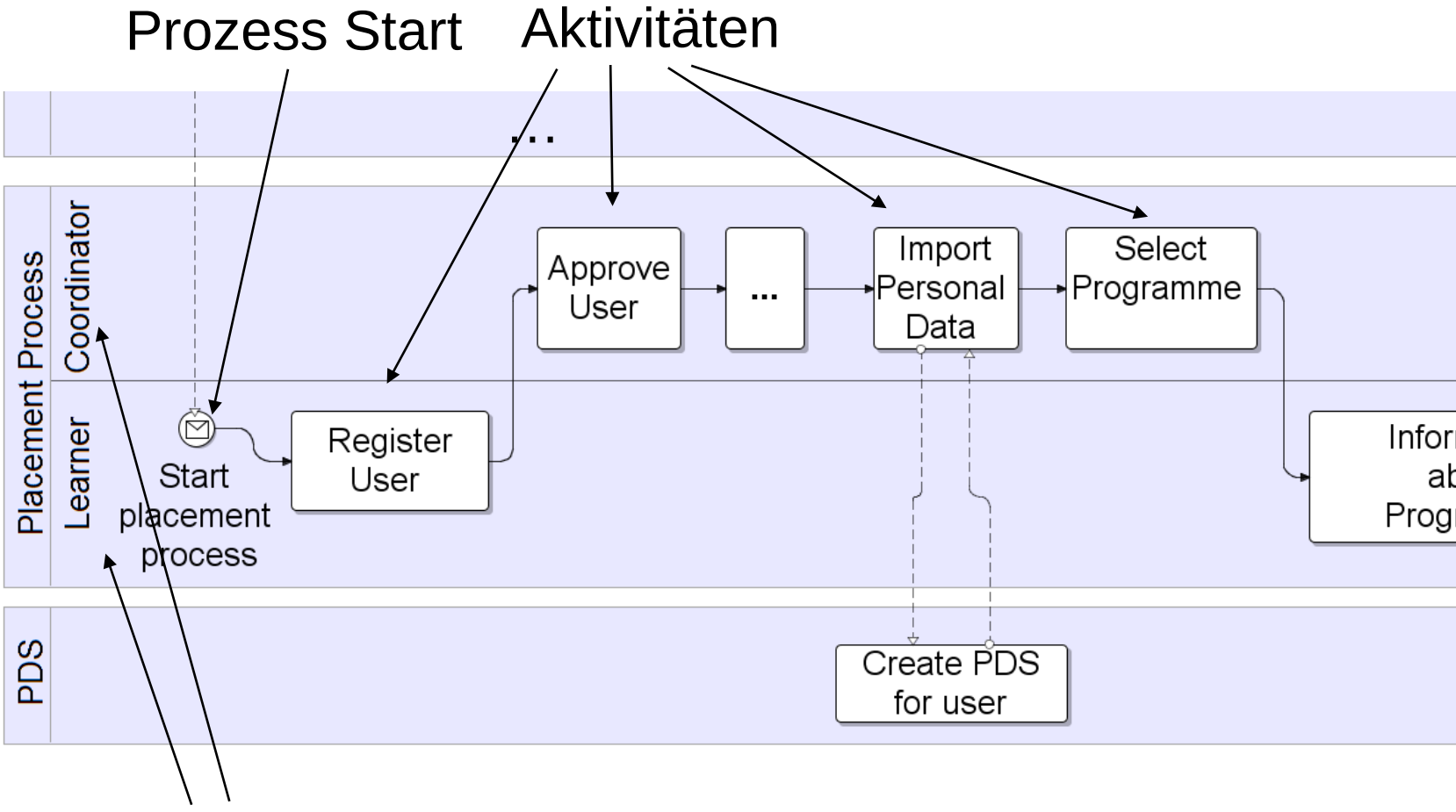
Wiederholung der Kernelemente BPMN



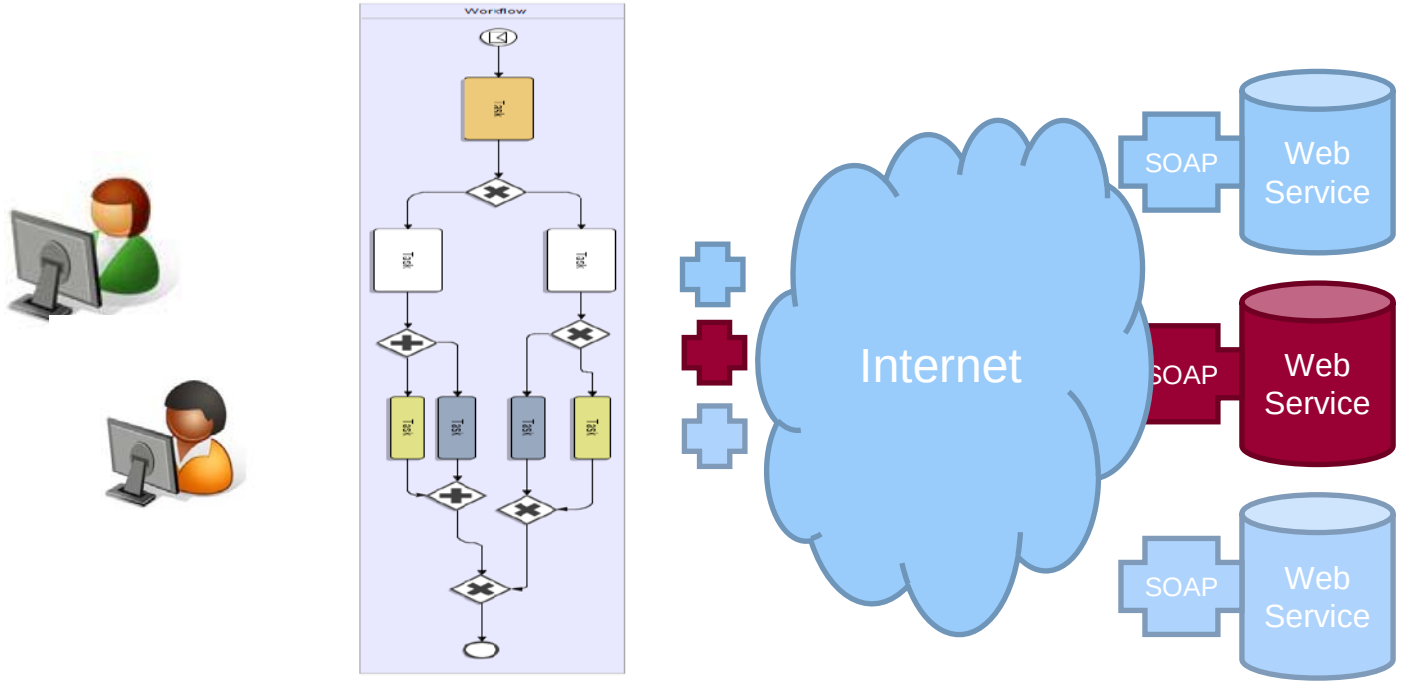
Wiederholung „Artifacts“ in BPMN – Beispiel Praktikumsvermittlung -



Einführungsbeispiel
- Teilablauf einer Praktikumsvermittlung -



Service-orientierte Architekturen (SOA) und Workflows



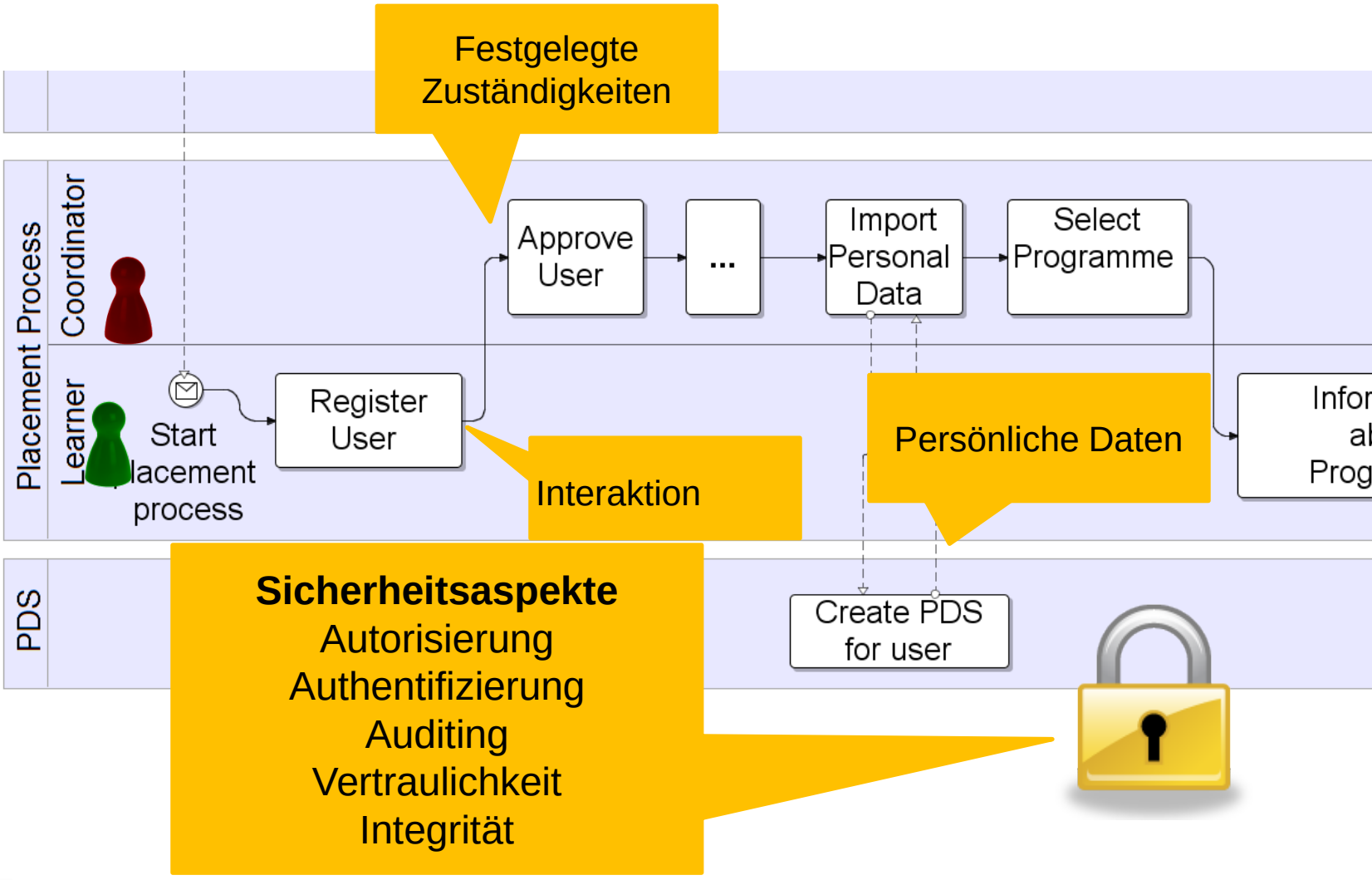
- SOA: lose gekoppelte Services in einem Netzwerk
- Workflow-Management-System (WfMS) koordiniert Prozesse durch Service-Aufrufe
- Interaktionen mit Akteuren

Zusammenfassung (bisher)

- BPMN weit verbreitete, graphische Modellierungssprache
- In Vorlesung wichtigste Konstrukte behandelt
- Was passiert mit Diagrammen? → Prozess-Ausführung durch WfMS

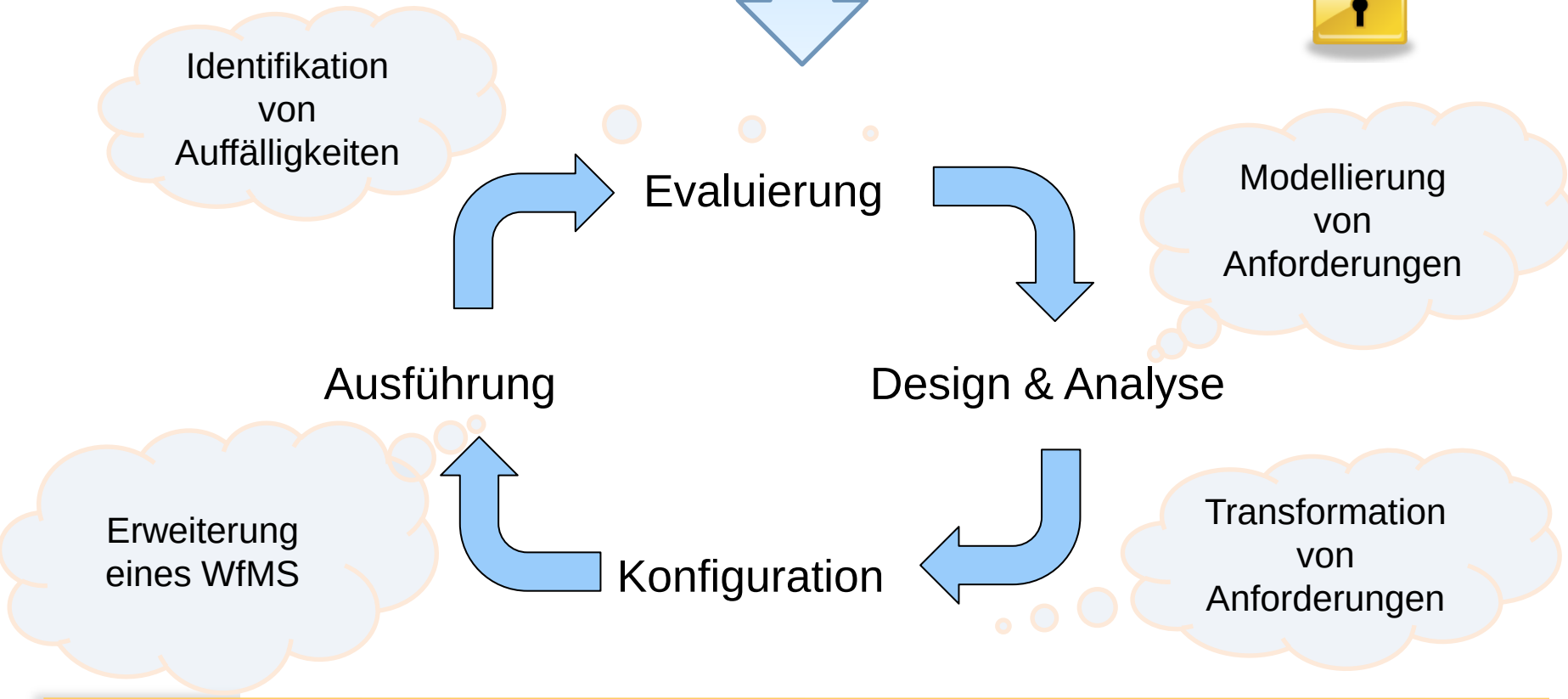
- bisher: generische Modellierung von Workflows
- Aber: in Praxis weitere Aspekte, wie Sicherheit und Privatheit, von Bedeutung
- Forschungsfrage: Einbindung von Sicherheit und Privatheit bei Modellierung (und Ausführung) von Workflows

Überblick Sicherheit und Privatheit in Workflows - Beispiel Praktikumsvermittlung -

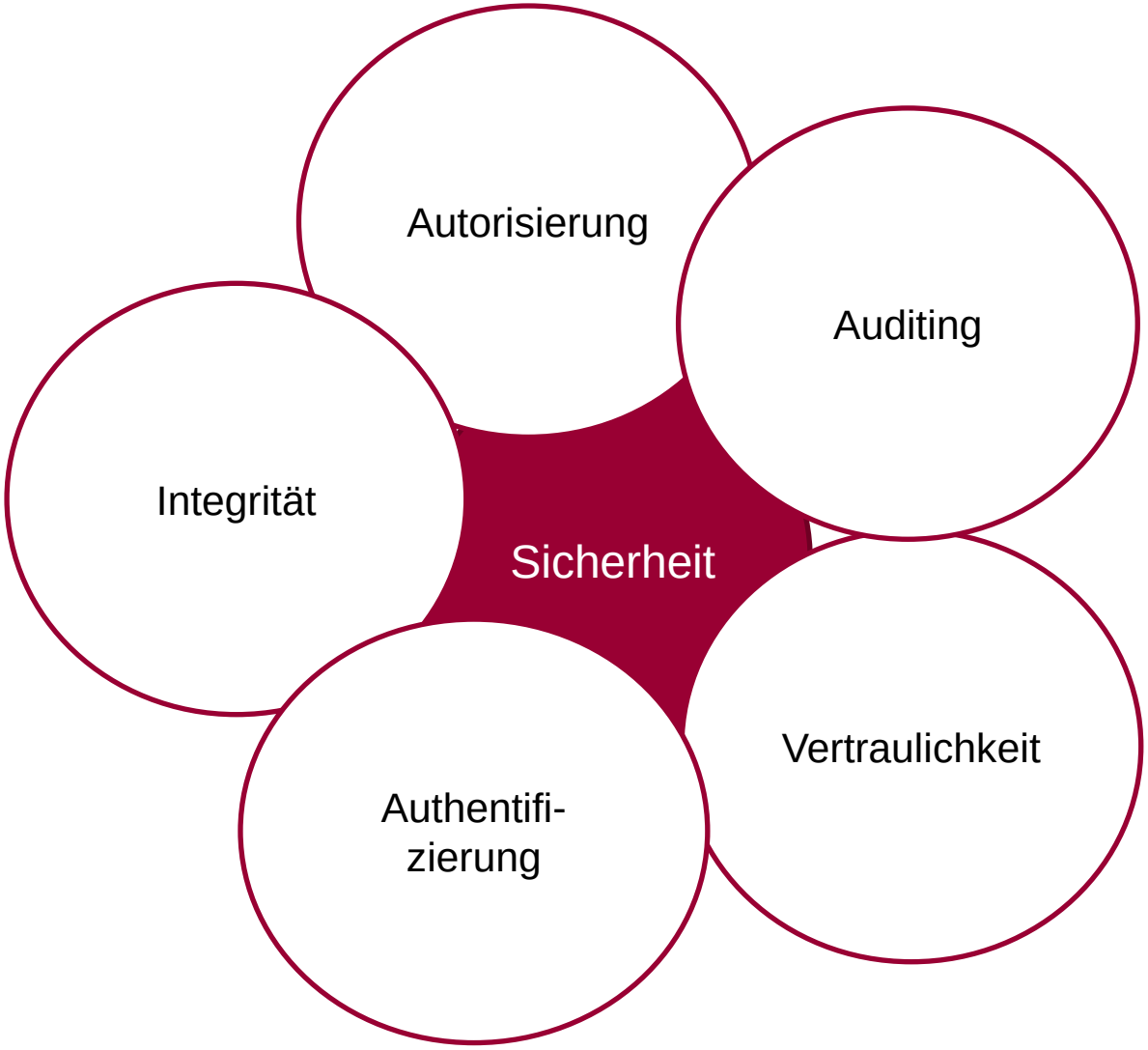


Forschungsfragen

Ziel: Effiziente Integration von Sicherheitsmechanismen und Privatheitsaspekten in WfMS



Sicherheits-Facetten in SOA



Sicherheit in Workflows

- Autorisierung (Access Control)
  - Rechtevergabe an Akteure oder Web Services für die Durchführung von Aufgaben (Aktivitäten)
  - Rollenkonzepte
  - Workflow-spezifische Einschränkungen
    - z.B. Pflichtenbindung (Binding of Duty (BoD)), Pflichtentrennung (Separation of Duty (SoD))
  - Delegation von Rechten auf Daten
- Authentifizierung
  - Beglaubigte Identifikation von Beteiligten
- Vertraulichkeit von Datenflüssen
  - Verschlüsselung/ Signatur
- Datenintegrität
  - Korrektheit und Schutz vor unberechtigter Manipulation der Daten
- Auditing
  - Protokollierung (Logging)

Privatheit in Workflows

- Personenbezogene Daten
  - Rechtliche Rahmenbedingungen (z.B. Zustimmungen, Nachverfolgung → Auditing)
  - Trust (Reputation) von beteiligten Services
- Interaktive Einbeziehung der sog. „Betroffenen“ bei **Ausführung** von Workflowinstanzen
  - Berücksichtigung persönlicher Präferenzen
    - Regelungen für Datenzugriffe
    - Informationspolitik (Notifikationen) bzgl. Umsetzung von Richtlinien
    - Auswahl von Services zur Speicherung personenbezogener Daten anhand der Reputation (Trust-Level)
  - Zustimmungen zu Geschäftsbedingungen („Give Consent“)

Konventionelle Ansätze

- Basismechanismen für Sicherheit in WfMS verfügbar
  - Rollenkonzepte zur Autorisierung
- Nicht hinreichend unterstützt
  - Autorisierungen (z.B. für Bereiche des Workflows)
  - Separation of Duty (SoD), Binding of Duty (BoD)
  - Rechtemanagement für Daten
- Umsetzung von Sicherheit durch proprietäre Software-Erweiterungen
  - aufwändig, kostspielig
- Privatheit: existierende Ansätze berücksichtigen nicht, dass Entscheidungen **kontextabhängig** sein können

Ansatz zur Umsetzung von Sicherheit und Privatheit in Workflows

- Entwicklung einer Sprache
  - Sicherheits- und Privatheitsbedingungen in Workflows
- Einbindung in BPMN-Diagramme
  - Bedingungen als Annotationsterme an BPMN Elemente
- Generelle Repräsentation:

```
<< Annotationterm:
   list(parameter-name=„value“) >>
```

- Umsetzung der Annotationen durch WfMS mit Hilfe von neu entwickelten Komponenten
  1. Transformation
  2. Ausführung

Annotationssprache: Autorisierung (I)

- „Role Assignment“ (Rollenzuweisung)
  - Autorisierung von Rolleninhabern zur Ausführung einer Aktivität
- „Assignment Mechanism“ (Zuordnungsmechanismus)
  - Zuordnung der Aufgaben an Rolleninhaber (z.B. Erfahrungsgrad, Kundenbetreuer)
- „User Assignment“
  - Direkte Zuordnung eines Benutzers

Annotationssprache: Autorisierung (II)

- „Separation of Duty“ (Pflichtentrennung)
  - Annotierte Aktivitäten müssen von unterschiedlichen Benutzern durchgeführt werden
- „Binding of Duty“ (Pflichtenbindung)
  - Annotierte Aktivitäten müssen vom gleichen Benutzer durchgeführt werden
- Delegation
  - Weitergabe der Zugriffsrechte auf Daten
  - Weitergabe der Ausführungsrechte für Tasks

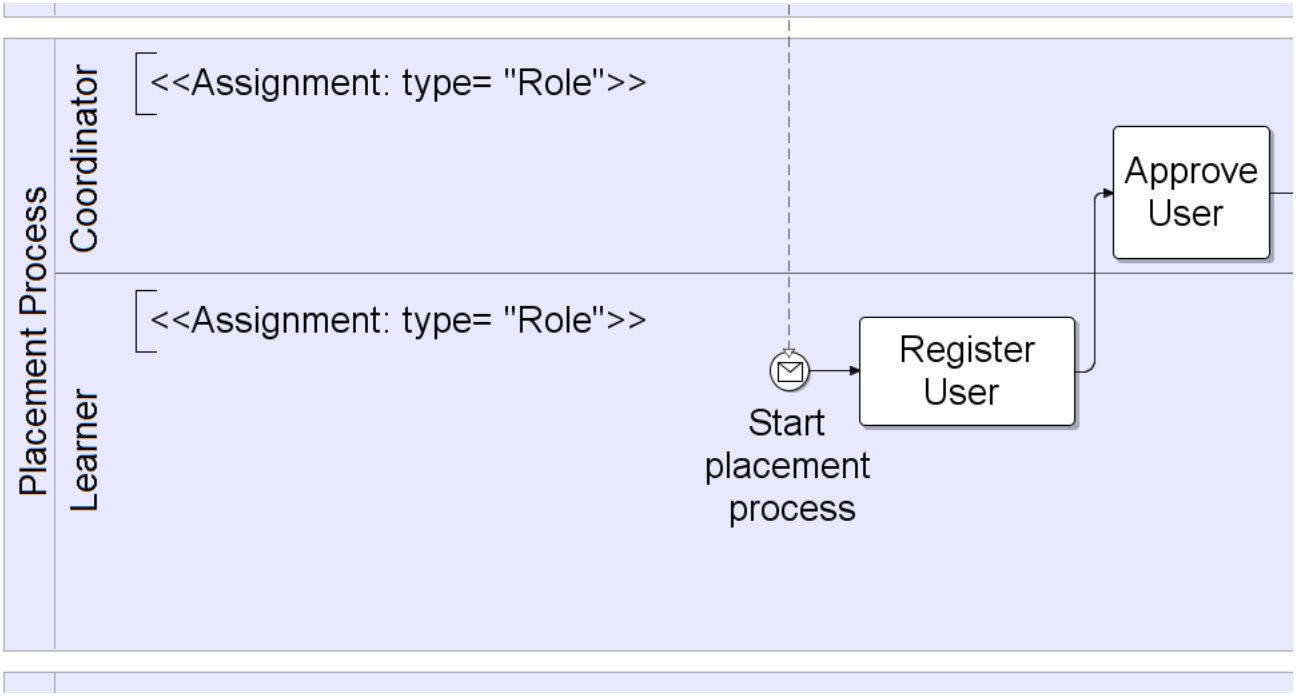
„Role Assignment“

- Für Aktivitäten, Gruppen von Aktivitäten (GoA), oder Swimlanes
- Syntax:

```
<< Assignment: type=„Role“
   name=„$rolename“ >>
```

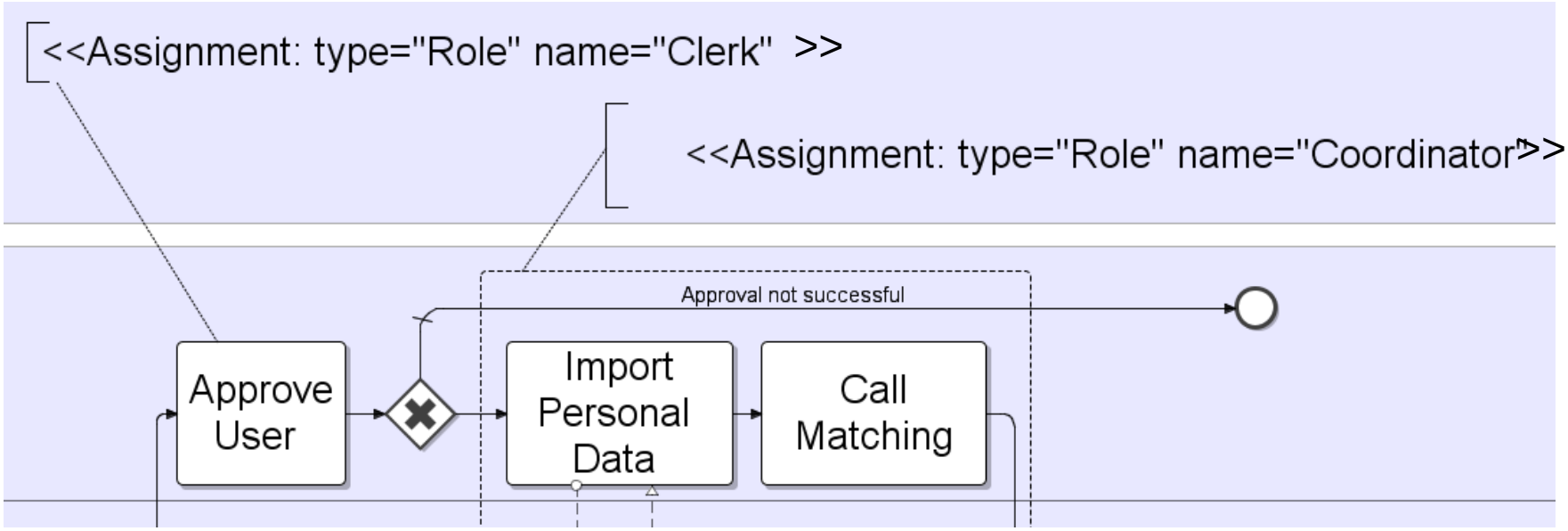
- Bei Aktivitäten `name=„$rolename“` obligatorisch, ansonsten optional
- Einschränkung: max. eine Annotation pro BPMN-Element

Beispiel „Role Assigment“ für „Lanes“



- Autorisierung der Rolleninhaber „Coordinator“ für alle Aktivitäten von „Lane“ *Coordinator* (analog „Lane“ *Learner*)

Beispiel „Role Assignment“ (Aktivitäten)



- Autorisierung von Aktivität „Approve User“ durch Rolleninhaber von „Clerk“
- Analog Gruppe von Aktivitäten (GoA) durch Rolleninhaber von „Coordinator“

Assignment Mechanism

- Assignment Mechanism legt fest, wie Zuordnung von Aufgaben an Rolleninhaber erfolgt
- Korrespondiert mit „Role Assignment“
- Syntax:

```
<<Assignment: type=„Mechanism“
    name=„$mechanismname“ >>
```

- „$mechanismname“ a priori spezifiziert
- Beispiel für Mechanismus:
  - falls Rolleninhaber Coordinator verfügbar, der frühere Vermittlungen von Learner bearbeitet hat, dann Zuweisung dieses Coordinators; falls nicht, nächst verfügbaren Coordinator zuweisen

„Binding of Duty“ (BoD)

- Pflichtenbindung für Aufgaben
  - Durchführung durch gleiche Akteure
- Für Aktivitäten, GoA, „Swimlanes“
- Syntax:

```
<<BoD: spec=„weak“>>
```

- `spec` optionaler Parameter
- `spec=„weak“` lässt explizite Änderung der Zuweisung zu („Re-assignment“)

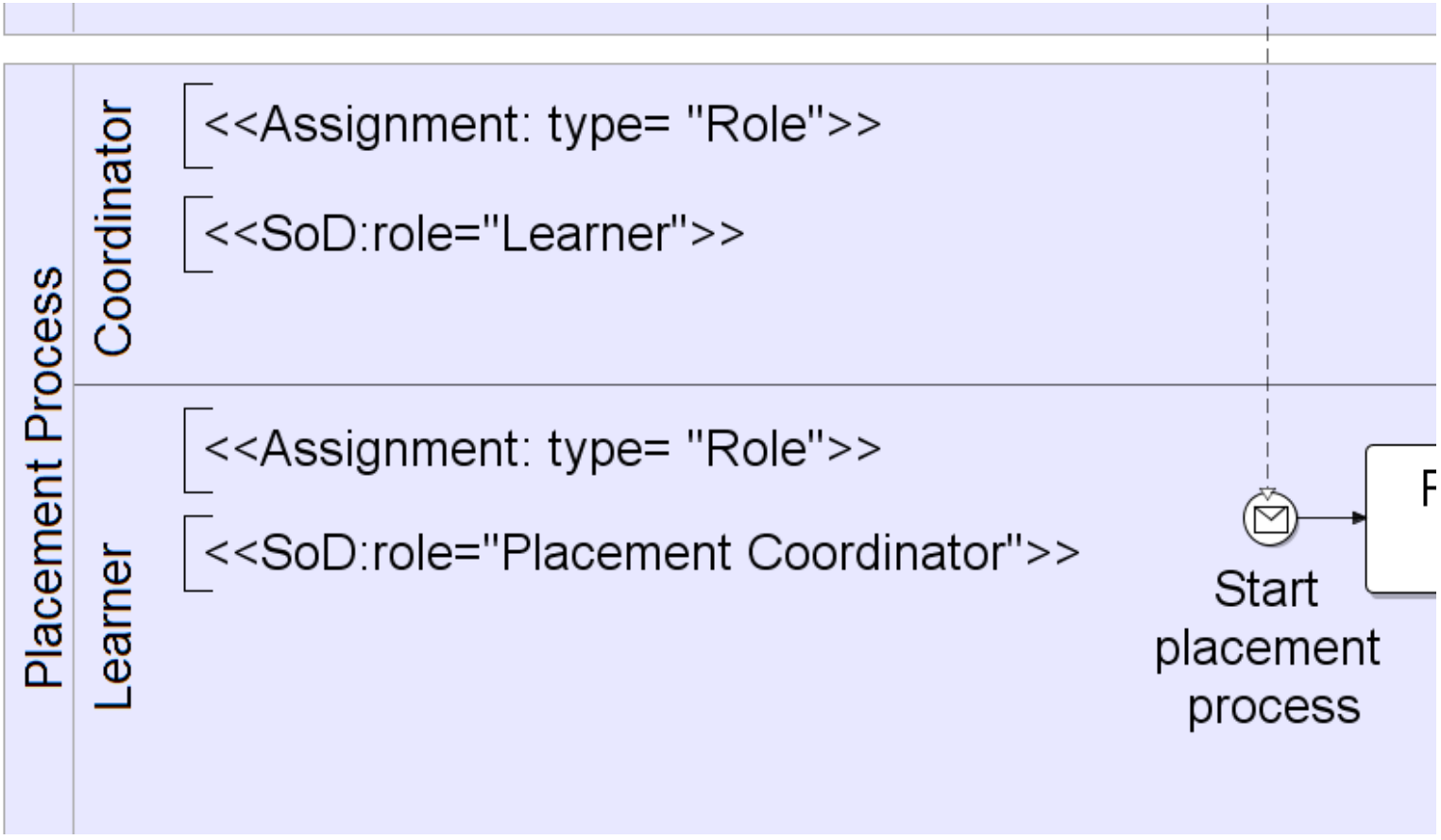
„Separation of Duty“ (SoD)

- Pflichtentrennung
  - Durchführung der Aufgaben von unterschiedlichen Benutzern
  - Motivation oft zusätzliche Sicherheit (z.B. 4-Augen-Prinzip)
  - Für Aktivitäten, GoA, „Swimlanes“
- Syntax:

```
<< SoD: role=„$rolename“
    number =„$value“
    threshold =„$value“ >>
```

- `role`, `number`, `threshold` optional
- Ausschluss von Rollen: `role=„$rolename“`

Beispiel SoD für Lanes

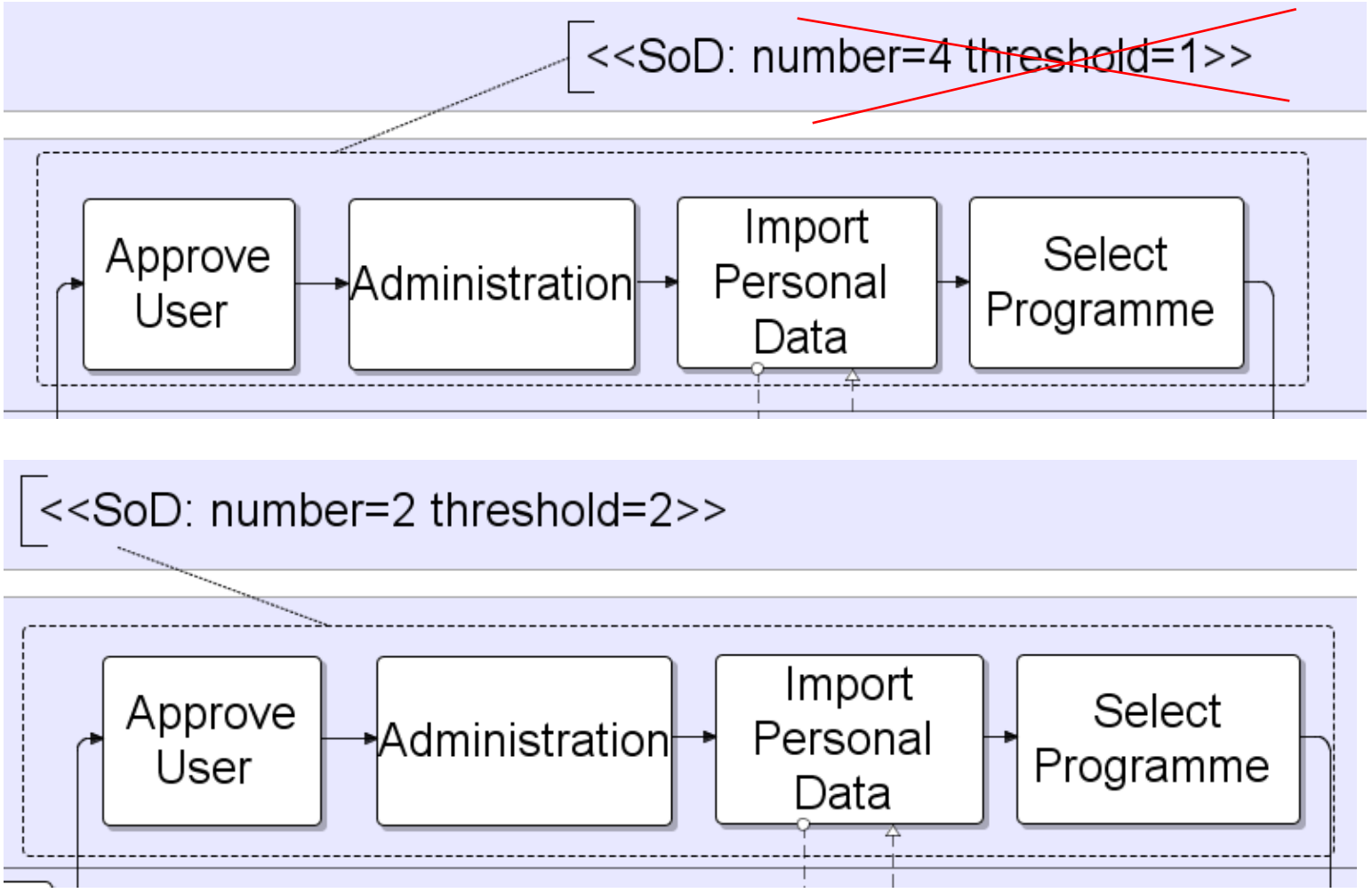


- SoD mit role=… als Ausschluss, d.h. Learner und Coordinator müssen unterschiedliche Personen sein

SoD mit Kardinalitäten

- Zusätzliche Ausdrucksmächtigkeit
- `number =„$value“:` minimale Anzahl unterschiedlicher Benutzer
- `threshold =„$value“`: maximale Anzahl von Aktivitäten (Instanzen), die von einem Benutzer durchgeführt werden dürfen (der gesamten, mit SoD annotierten Aktivitäten)
- Default-Semantik SoD: (number=2, threshold=1) bei 2 Aktivitäten bzw. bei wiederholender Aktivität

Beispiele SoD



Min. 2 Benutzer, jeder Benutzer maximal 2 Aufgaben

D-Delegation

- Weitergabe von Zugriffsrechten auf (geschützte) Daten
  - Prozessinterne Daten („data objects“)
  - Externe Daten („data stores“)
- Syntax:

```
<< D-Delegation:
      rights=“list ($rightsname)”
      object=“$objectname”
      interval=“($activityname1,$activityname2)|$group”
      owner=“$rolename”
      receiver=“$rolename”|“$webservicename”
      spec=“$specification” >>
```

D-Delegation: Parameter

- `rights`: read, write, delete, oder policy (deskriptive Beschreibung)
- `object`: Datenobjekt, für das Rechte vergeben werden (bzw. *View* auf Datenobjekt)
- `owner:` Rechteinhaber, Betroffene
- `interval`: Zeitintervall für Rechtevergabe
- `receiver` (Empfänger): `rolename` or `webservicename`
- Spezifikation `spec`: grant (optional)

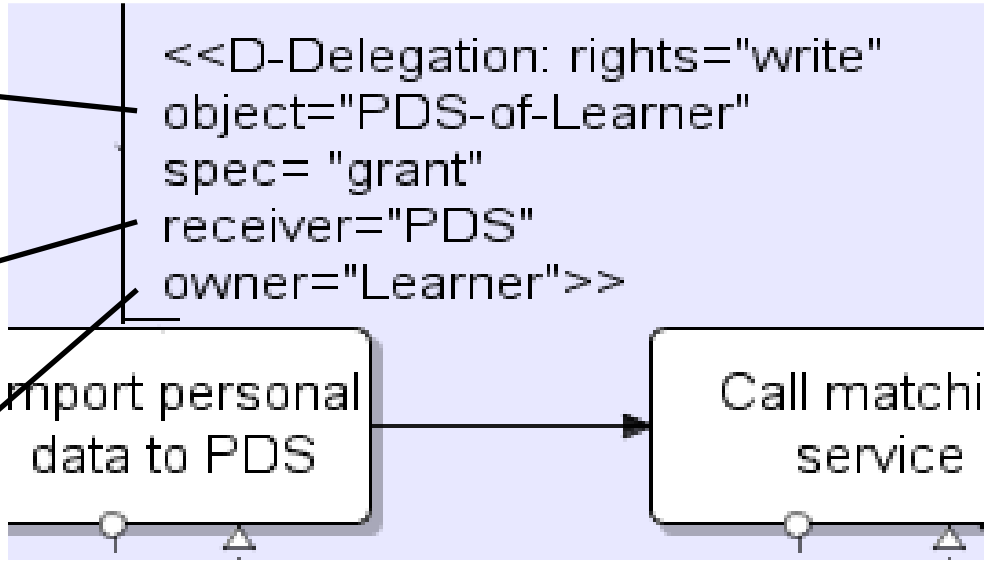
Beispiel D-Delegation

Delegation von Rechten für den Zugriff auf Daten

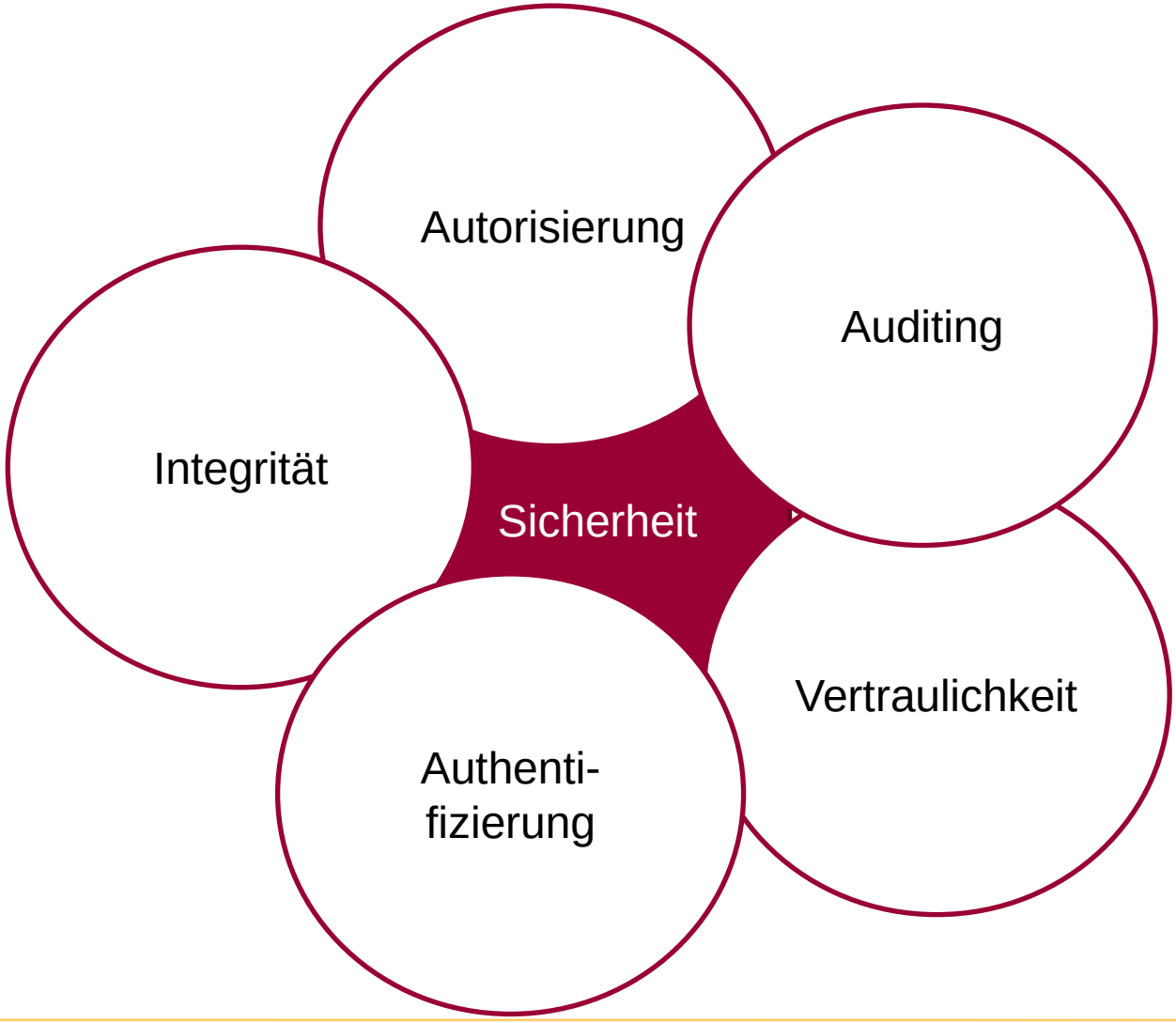
Objekt, für das Rechte vergeben werden

Web Service „PDS“ ist Empfänger der Rechte

„Learner“ ist nicht Ausführender der Task, aber Rechte-Inhaber!



Sicherheits-Facetten in SOA



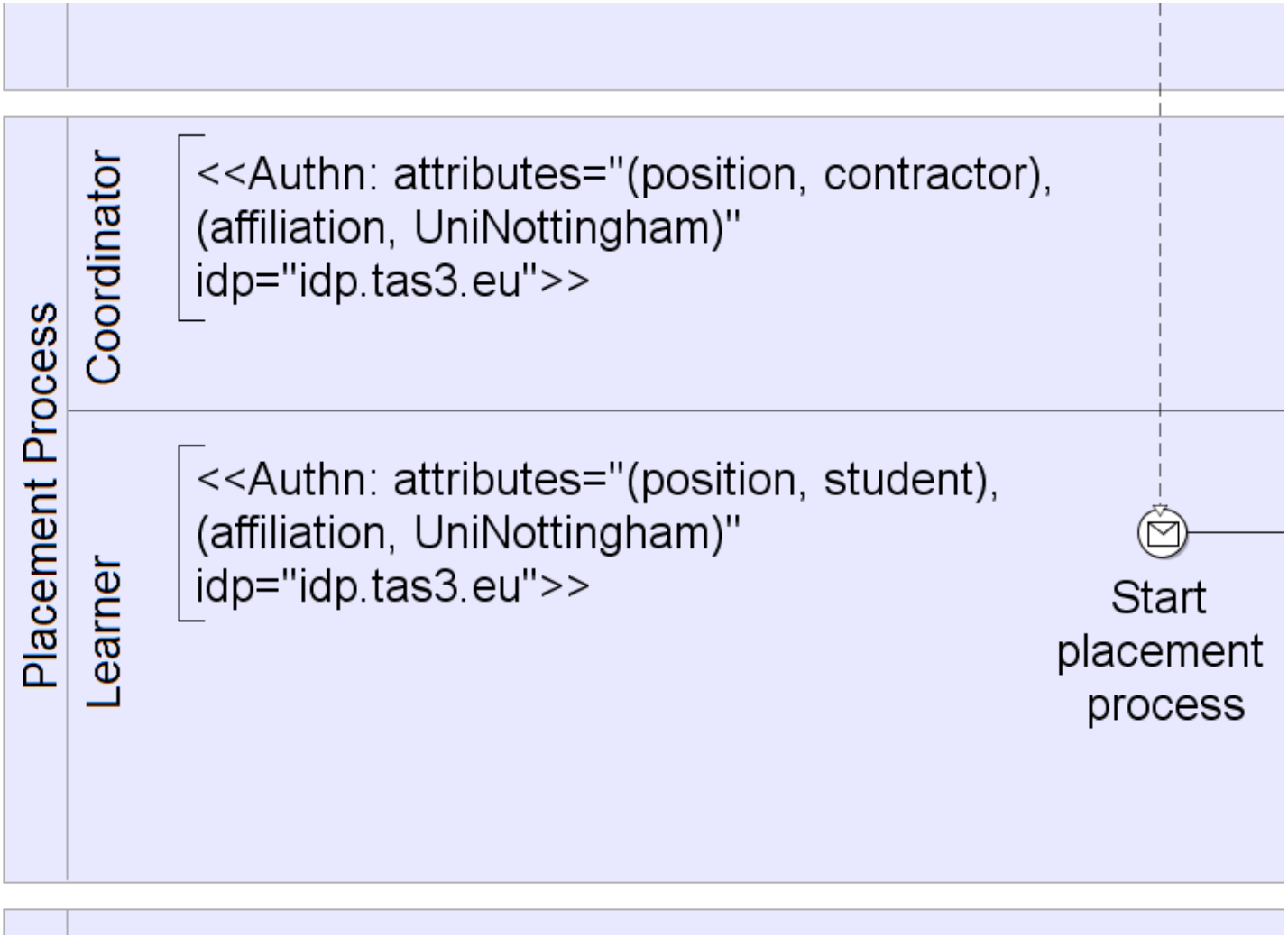
Authentifizierung

- Identifikation von Beteiligten
  - Attribut/Werte-Paare
  - Bestätigung durch Identity-Provider
- Syntax:

```
<< Authn: attributes=„[list[(attributename,
    value)“]
    idp=„$identityprovider“ >>
```

Anmerkung: Trennung der Listenelemente durch „ ,“ (siehe folg. Bsp.)

Beispiel Authentifizierung



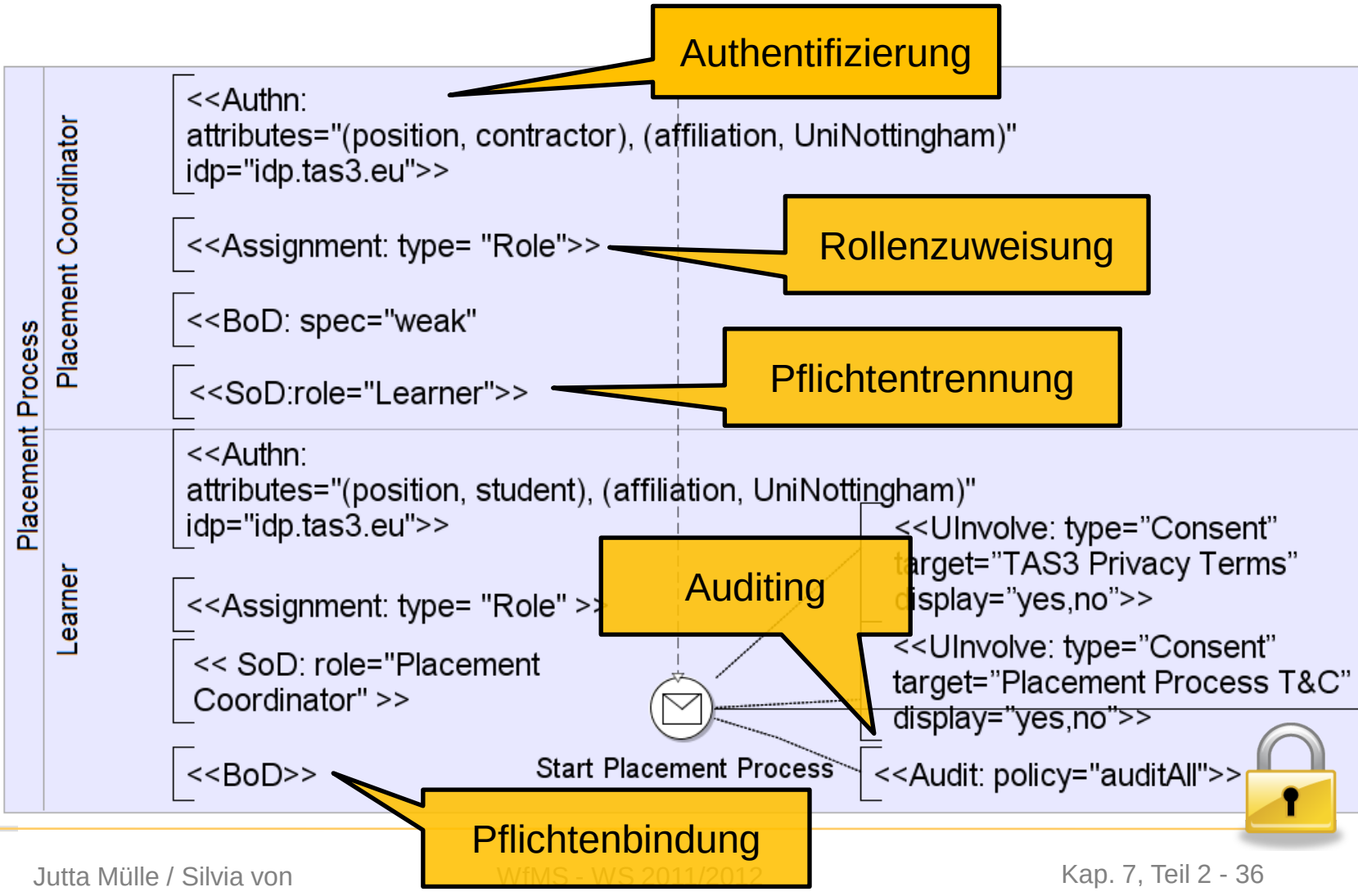
Auditing

- Protokollierung (Logging) des Prozesses bzw. von Teilen des Prozesses
- Motivation u.a. auch gesetzliche Regelungen
- Syntax:

```
<< Audit: policy=„$policyname“ >>
```

`policy` spezifiziert, was protokolliert wird (z.B. Zugriff auf personenbezogene Daten gemäß Datenschutzrichtlinien)

Beispiele Sicherheits-Annotationen in BPMN - Prozessanfang



Einbeziehung von Benutzern

- Zusätzlich zur Sicherheits-Sprache: Einbeziehung von Benutzern (sog. „*User Involvements*“) zur Spezifikation von Privatheits- und „Trust“-Aspekten
  - Einwilligung zu Geschäftsbedingungen (z.B. Terms & Conditions)
  - Datenzugriffs-Regeln (Policy)
  - Regeln für Auswahl beteiligter Services anhand der Vertrauenswürdigkeit („Trust“)
  - Trust-Feedback nach Nutzung von Services
  - Interaktions-Präferenzen
- Nicht a priori, sondern **dynamisch** zur Laufzeit von Workflows
  - Abhängig von **Workflow-Kontext**

Ansatz zur Einbeziehung von Benutzern

- Für alle Optionen vormodellierte Teilprozesse
- Sprache zur Darstellung von Teilprozessen
  - Parametrisierbar
  - Prozessmodellierer greift auf vordefinierte Prozesse zurück, kein Bedarf an Umsetzungs-Kenntnissen
- Dynamische Einbindung der Teilprozesse bei Ausführung eines Workflows (Schema-Erweiterung)

Überblick Sprache „User Involvements“

- Syntax

```
<< UInvolve: type=„$UInvolvetype“  ... >>
```

- Vordefinierte Werte für `UInvolvetype:`
  - `SetIAPref`
  - `Consent`
  - `SelectService`
  - `SetDataPolicy, SelectDataPolicy`
  - `SetTrustPolicy, SelectTrustPolicy`
  - `TrustFeedback`

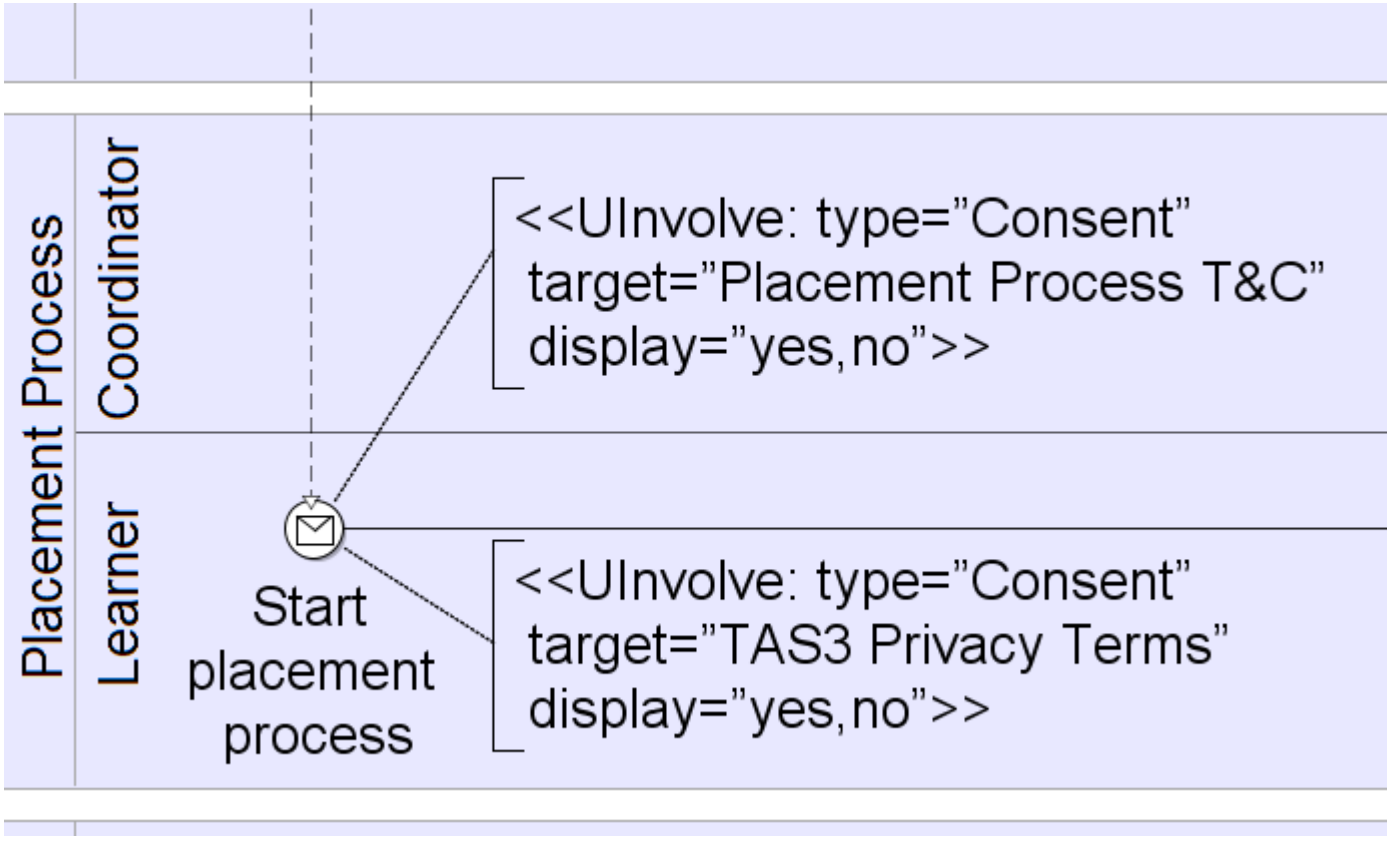
„Consent“ (Zustimmungen)

- Zustimmung der Benutzer zu den Geschäftsbedingungen des Workflows
- Syntax

```
<<UInvolve: type=„Consent“
   target=„$targetname of privacy terms“
   role=„$rolename“ insertplace=„$activityname“
   display = „$option1,…,optionn“ >>
```

- `target` spezifiziert Geschäftsbedingungen
- `display` gibt Benutzeroptionen an
- Optional: `role`, `insertplace`

Beispiel „Consent“



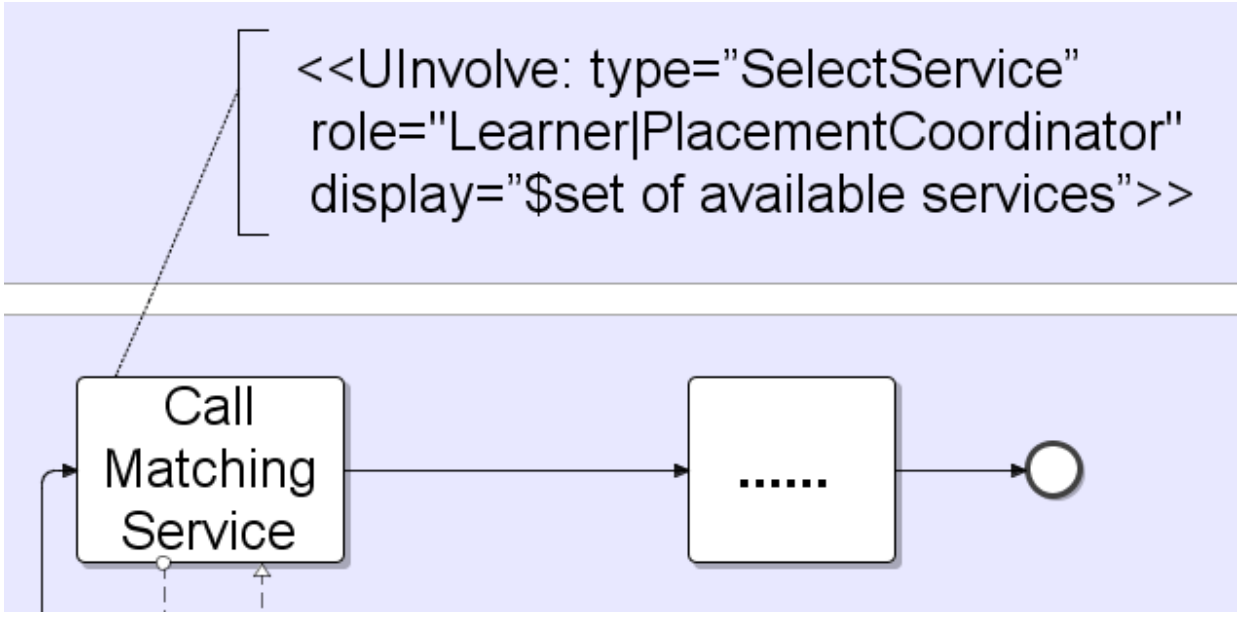
„Select Service“

- Auswahl aus den verfügbaren Web Services
- Voraussetzung: Spezifikation des Trust Levels
- Syntax:

```
<<UInvolve: type=„$SelectService“
  display=„list($option)“
  role=„$rolename“
  insertplace=„$activityname“>>
```

- `display` gibt verfügbare Services an
- optional: `role`, `insertplace`

Beispiel „Select Service“



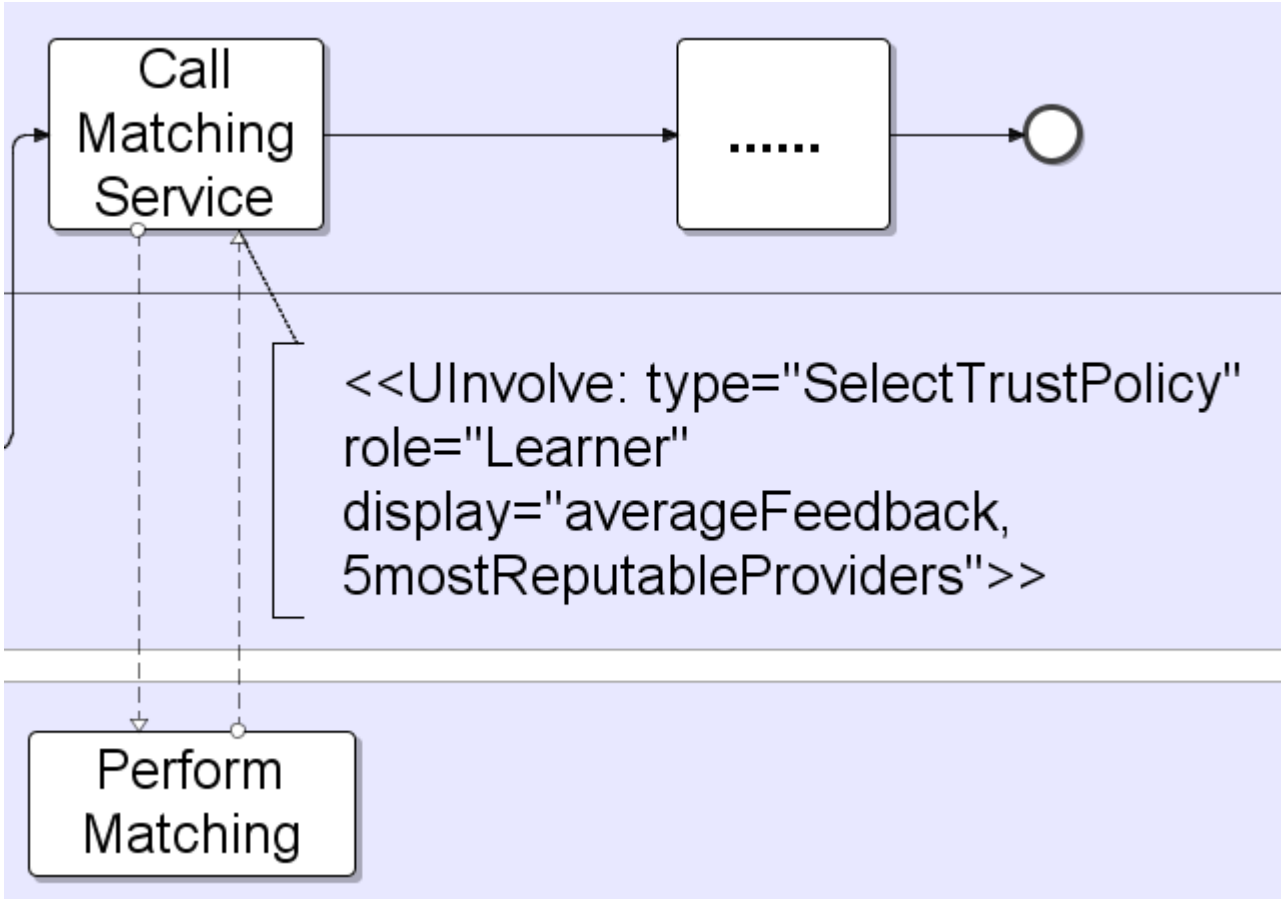
„Select Trust Policy“

- Trust Policy spezifiziert die Anforderungen an die Vertrauenswürdigkeit (Trust) eines Web Services
  - Notwendige Spezifikation bei Service Selection
- Benutzer wählt z.B. Trust Level für beteiligte Services
- Syntax:

```
<<UInvolve: type=„SelectTrustPolicy“
  display=„list($option)“
  role=„$rolename“
  insertplace=„$activityname“>>
```

- `display` gibt Benutzeroptionen an
- optional: `role`, `insertplace`

Beispiel „Select Trust Policy“

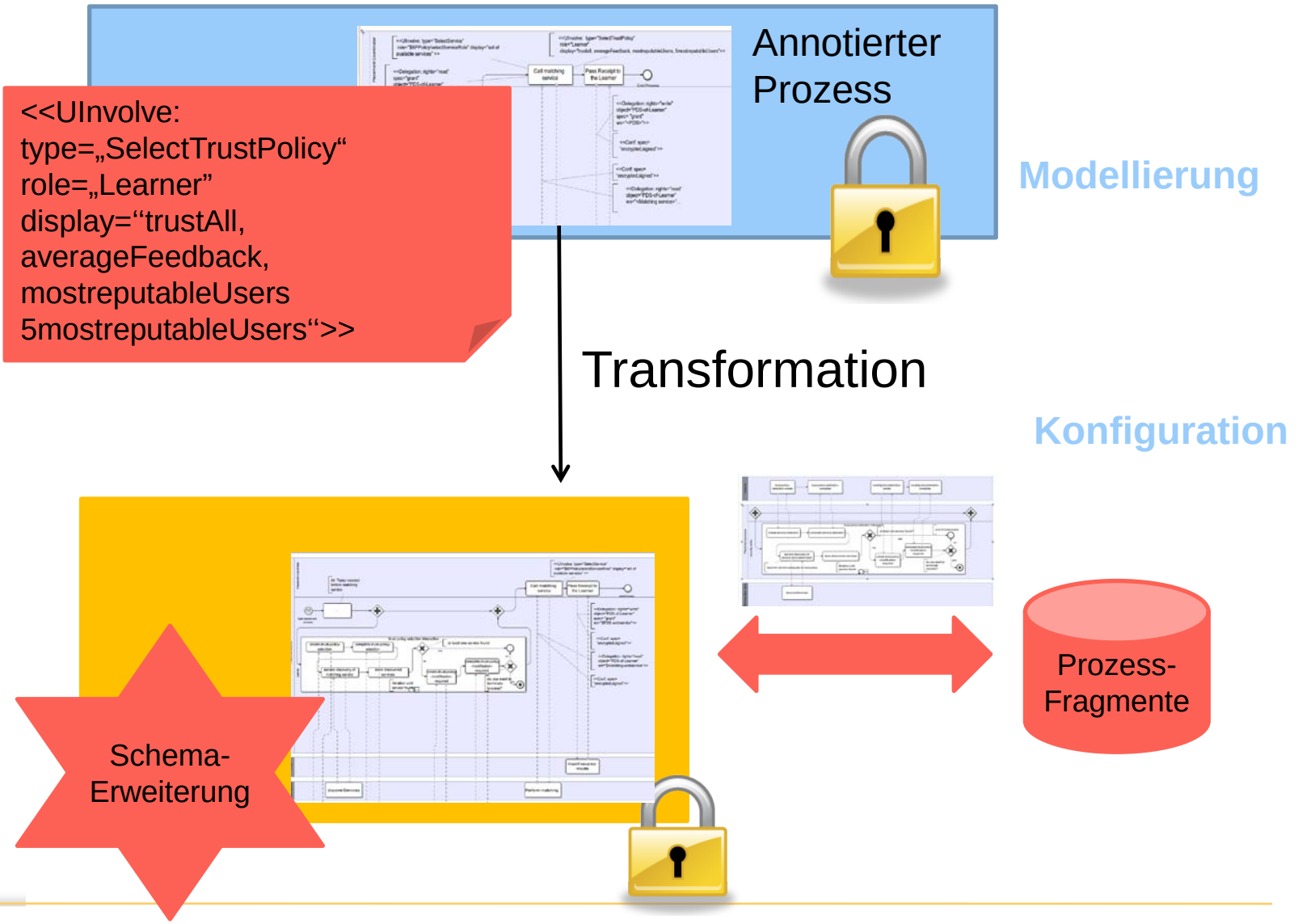


Transformation der Annotationen

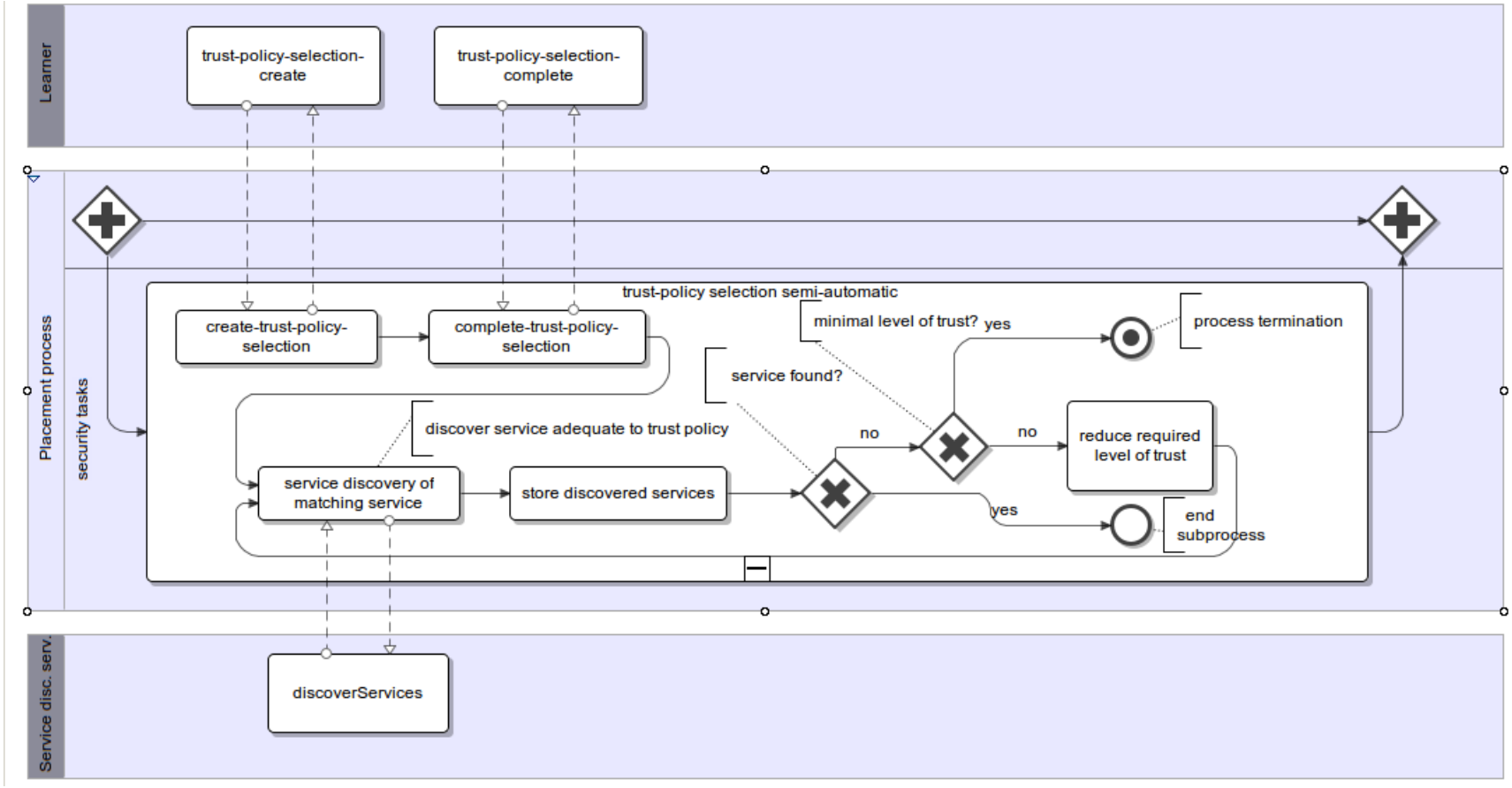
- Transformation „verarbeitet“ die Annotationen und bereitet die Ausführung für die Engine vor

- Erweiterung des Workflow-Schemas vor Ausführung
- Im Folgenden exemplarisch für `<<UInvolve: type=„SelectTrustPolicy“ … >>`

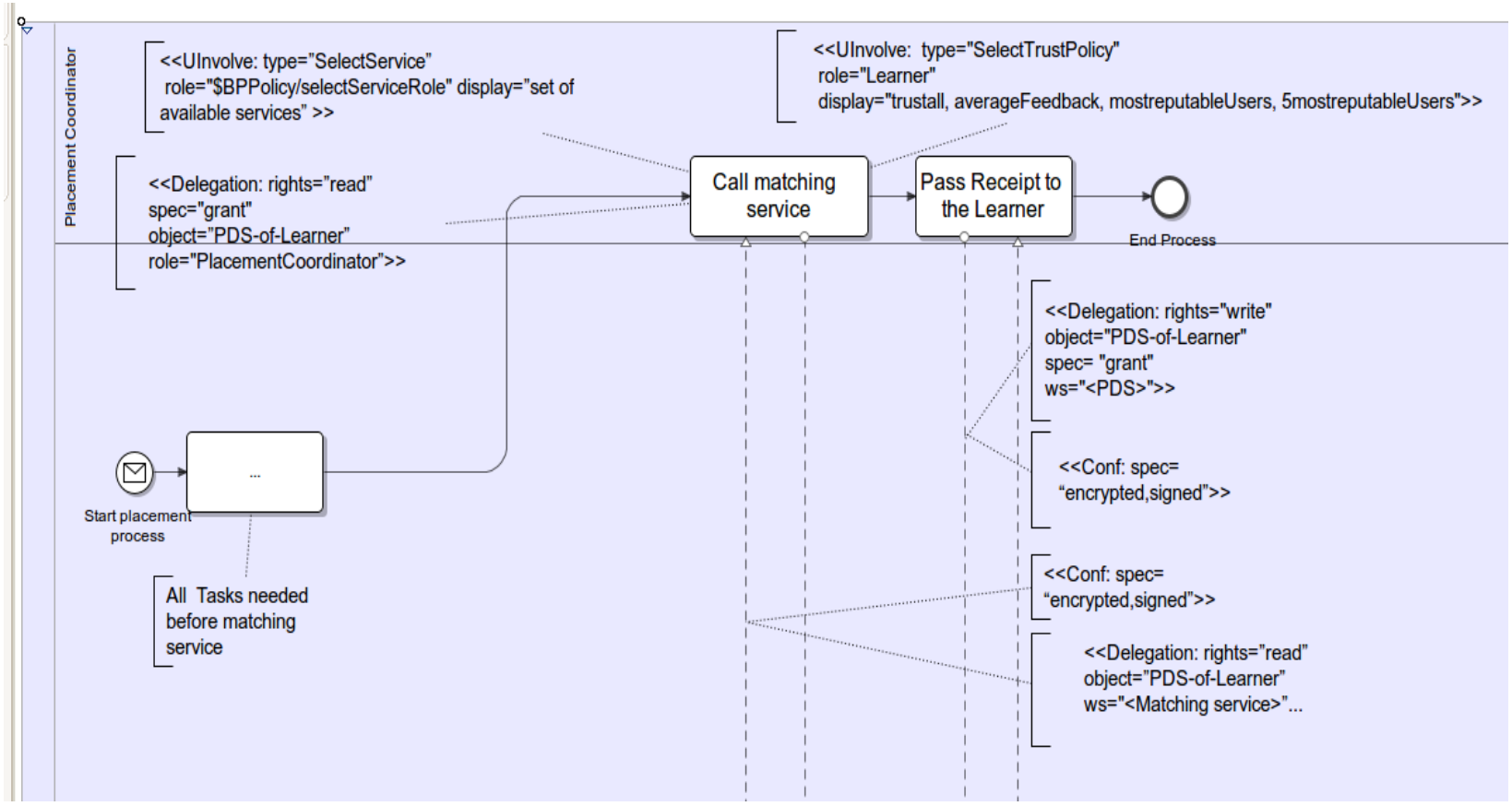
Transformation



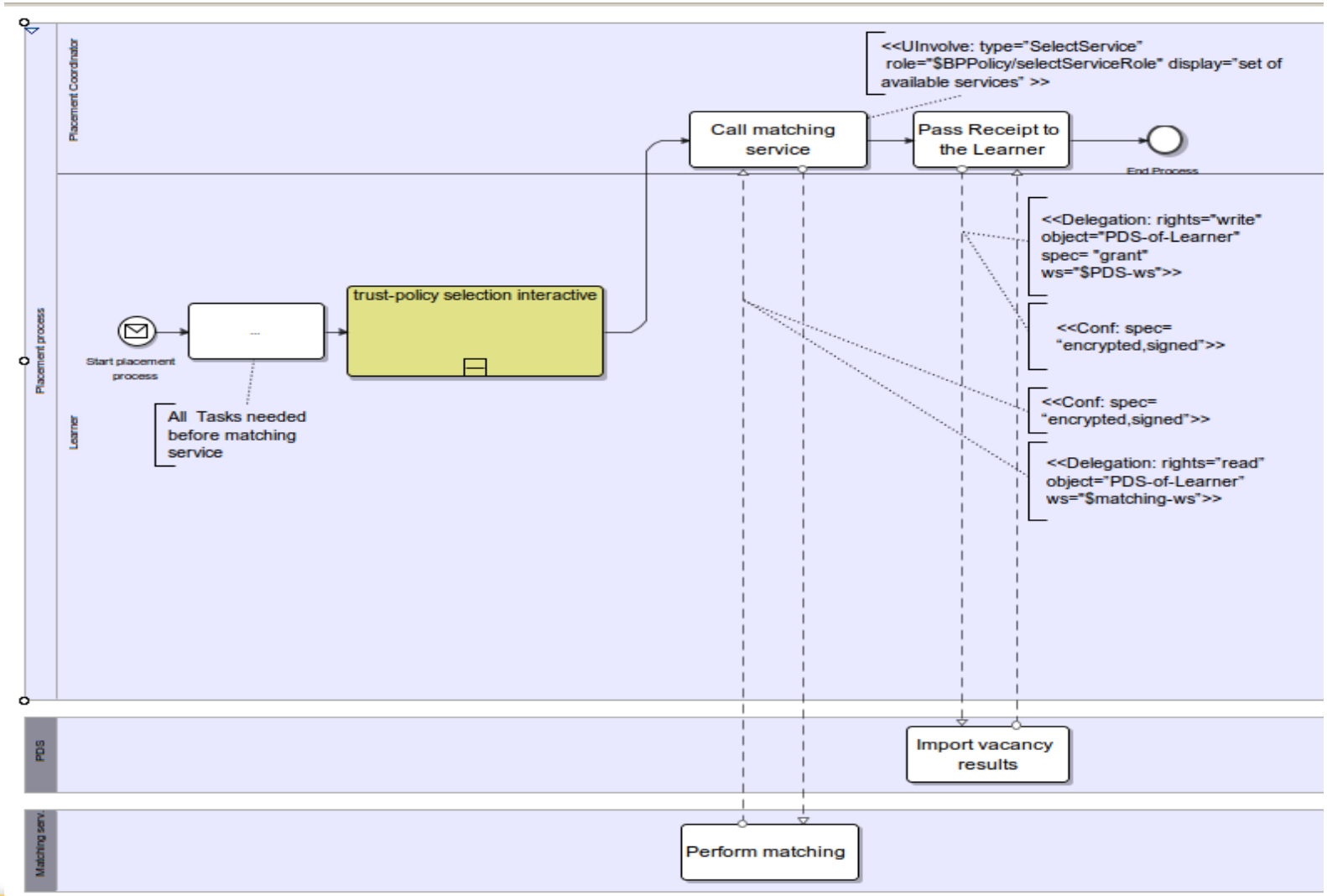
Prozessfragment "Select Trust Policy"



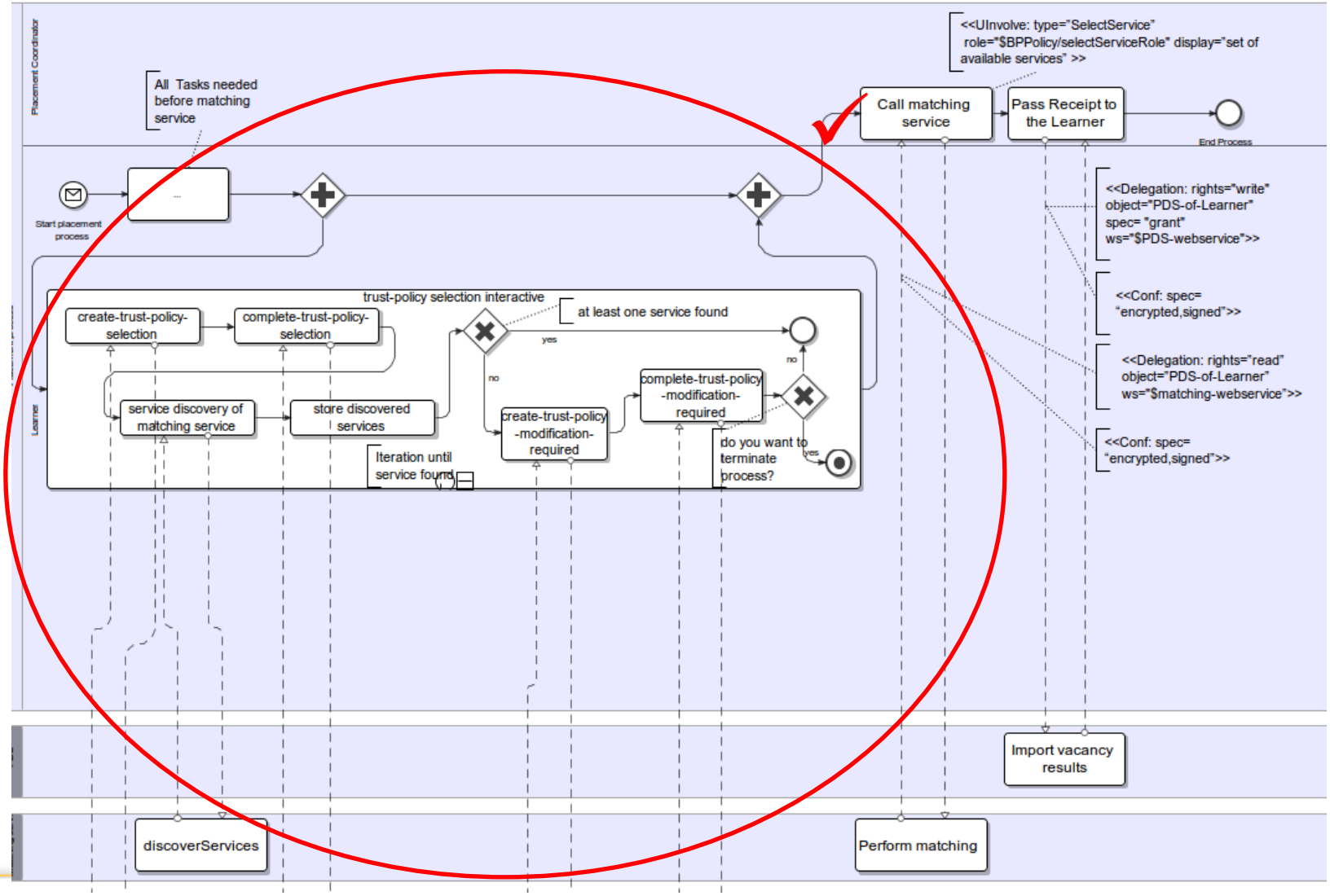
Prozess mit Annotationen



Workflow mit geschlossenem Sub-Prozess



Workflow mit integriertem Prozessfragment



Zusammenfassung

- Modellierung von Workflows wichtiger Schritt zur autom. Koordination von Abläufen
- Aktuelles Forschungsthema zur Modellierung von Sicherheitsaspekten, Privatheit und Trust am IPD
  - Sprache
    - Sicherheitsaspekte
    - Einbeziehung von Benutzern zur Spezifikation von Präferenzen
      - Vorwiegend für das Verwalten von Daten

Übung

- „Hausaufgabe“
- Aufgabenstellung siehe Übungsblatt (Website Lehrveranstaltungen IPD)
- Freiwillige, sofern gewünscht anonymisierte Teilnahme
- Bearbeitungszeit **bis 21. Dez. 2011**
- Schriftliche Abgabe, danach Feedback
- Incentives abhängig von Leistung

Exemplarische Fragen Kap. 7- Teil 2

- Klassifizieren Sie Sicherheit.
- Beschreiben Sie kurz die Idee, Sicherheit in WfMS umzusetzen.
- Geben Sie Beispiele für konkrete Annotationen an (z.B. Authorisierung, Authentisierung, Einbeziehung von Benutzern)

Ergänzende Literatur Kap. 7 - Teil 2

Sicherheit in Workflows

- Mülle, Jutta, von Stackelberg, Silvia, Böhm, Klemens: A Security Language for BPMN Process Models. Tech. Rep. 2011-09, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, http://www.ubka.uni-karlsruhe.de/eva/index.html
  (alte Syntax)
- Mülle, Jutta, von Stackelberg, Silvia, Böhm, Klemens: Modelling and Transforming Security Constraints in Privacy-Aware Business Processes, to appear in: SOCA, Dec. 2011
  (modifizierte Syntax)