Evolutionsszenario

# EAA Evolution: Altes unvernetztes IT System zu neuem Vernetzen IT System

**Vorher**

* Mehrere IT Systeme wie (e.g. SammlungsDB, CRM, Fotoserver, StandortDB, … (Siehe „EAA PCM.jpg“)) die unabhängig voneinander betrieben wurden
* Problem:
  + Informationen waren teilweise doppelt vorhanden in den Systemen weil sie dort ebenfalls benötigt wurden
  + Doppelte Informationen mussten Manuell nachgepflegt werden was erhebliche Arbeitszeit kostete und zu Inkonsistenzen führte

**Ziel**

* Unabhängige IT Systeme mittels einer Middleware/Broker miteinander vernetzen
  + Sichten auf Informationen statt Zugriff auf einzelne Systeme/DB
* (+ Eigene DB Schlanker gestalten um unnötige redundante Informationen in den verschiedenen Systemen zu vermeiden)
  + Informationen möglichst nicht mehr Redundant Speichern, sondern vom Hauptort Zentral abrufen für die benötigten Sichten
* Vorteil:
  + Kein Zeitaufwand für manuelle Pflege von redundanten Informationen
  + Informationen überall synchron
  + Leichter konfigurierbare, erweiterbare Sichten für Rollen
  + (Effizientere Speichernutzung)
* -> Geschäftsprozesse bleiben gleich, nur EAA hat sich verändert
  + Das Ziel ist es ja auch die bestehenden Prozesse mit der neuen EAA zu unterstützen

Logical & Design Mistakes

* **Logical & Design Mistakes**
  + Logical Mistakes: Flüchtigkeits fehler, Etwas wurde zum Beispiel falsch verdrahtet auf falschen Service Call, oder der Service Call wurde falsch gestaltet
  + Design Mistakes: Anforderungen wurden missverstanden und es ist ein falscher Service Call entstanden oder man hat etwas mit dem Falschen Service Call verdrahtet
  + -> Rolle nutzt falschen Service Call (ELSC in IntBIIS)
    - Rolle bekommt Zugriff auf Daten (Lesen, Schreiben) auf die es laut BP keinen Zugriff hat
  + -> Service Call wurde falsch verändert (OperationSignature im Repository)
    - Rolle bekommt Zugriff auf Daten (Lesen, Schreiben) auf die es laut BP keinen Zugriff hat
  + -> Idee: Behandele Logical und Design mistakes nicht separat sondern Zusammen. Das bedeutet zu einem Mistake muss ich nicht argumentieren ob es ein Logical oder Design mistake ist, es könnte durch beides entstehen. Stattdesssen schauen wir uns die unterschiedlichen Kategorien in denen Logical und Design mistakes passieren können -> Vorteil: Einfachere Argumentation, Man erspart sich das ausdenken und verargumentieren der Mistakes (also wie sie entstanden sind)
* **Wieviele Mistakes brauchen wir?** Wir sollten damit eine gewisse Abdeckung erreichen – Alle Relevanten Fälle abdecken. **Relevante Kategorien Bilden und für jede ein Mistake bauen. – An welchen Stellen können Logical und Design Mistkes entstehen in EAA - Kategorien:**
  + **Assembly diagram/System diagram**:
    - **Cat1**: Falsely connected Required/Provided Role between a) System and Subsystem or b) Subsystems or c) Subsystem and Components or d) Components
      * **a) c) Required Delegation Connector:** Connected to false 1) Assembly Context Required Delegation Connector 2) Outer Required Role Provided Delegation Connector
      * **a) c) Provided Delegation Connector:** Connected to false 1) Assembly Context Provided Delegation Connector 2) Outer Provided Role Provided Delegation Connector
      * **b) d) Assembly Connector**: Connected to false 1) Providing Assembly Context Assembly Connector 2) Provided Role Assembly Connector
      * **L&D Mistake:** System.X (Provided Role an System muss noch erstellt werden) zeigt auf LeihverkehrsverwaltungDB.ProvIZugriffLeihverkehrsverwaltungDB (statt auf Leihverkehr.ProvISystemLeihverkehr)

Wahr. passt das nicht

Idee: 2 Schnittstellen, Schnittstelle A hat zugriff auf Comp. Datatype A, Schnittstelle B hat zugriff auf Comp. Datatype B der in A enthalten ist.

Im Prozess müssen A und B bekannt sein und man braucht nur Zugriff auf B

* + **Repository**:
    - **Cat2**: Falsely chosen Interface for SubSystem/Component
      * **Operation Provided Role**: Chosen false Provided Interface Operation Provided Role
      * **Operation Required Role**: Chosen false Required Interface Operation Required Role
      * **L&D Mistake:**
    - **Cat3**: Falsely chosen Parameter and Return of an Operation Signature
      * **Operation Signature - Parameter**: Chosen false Data Type Parameter
      * **Operation Signature**: Chosen false Return Type operation Signature
      * **L&D Mistake:** iZugriffSammlungsDB.gibAusstellungsobjekt statt „Fremdes Ausstelungsobjekt DBO“ gibt zurück „AusstellungsobjektDBO“
    - **~~Cat4~~**~~: Too many Parameters of an Operation Signature~~ ?Ist doch das selbe wie cat3: Wenn man sich vorstellt das EAA designed wird dann muss es die Parameter doch sowieso neu erstellen, außerdem ob ein parameter falsch verbunden wurde oder zu viele parameter definiert wurden ist doch das selbe am Ende
      * **Operation Signature - Parameter**: Too many Data Type Parameter
      * **L&D Mistake:** ISystemLeihverkehr.fügeProtokollHinzu zusätzlich den parameter Standort (Kommt in irgendwelchen BPs vor wo man) <-? Ist zu überprüfen
    - **Cat4**: Falsely build Composite and Collection Data Types
      * **Collection Data Type**: Chosen false Inner Type Collection Data Type
      * **Composite Data Type – Inner Declaration**: Chosen false Datatype Inner Declaration
      * **L&D Mistake:** 
        + Comp **Leihvertrag** **InnerDeclaration:VerliehensAusstellungsobjekt** zeigt auf **Ausstellungsobjekt** statt auf öffentlich sichtbares Ausstellungsobjekt
        + **BP:** UC3 AS Leihvertrag archivieren **?Problem Leivertrag hat innerDeclaration:Ausstellungsobject für welches doch in dem AS gar keine Berechtigung besteht. ?Ist es überhaupt sinvoll das Leihvertrag intern das Ausstellungsobjkekt hat, oder hat es nur AusstelungobjektNamen drin falls nicht müsste man den Prozess erweitern**
    - **Cat5**: Falsely build inheritance Hierarchy
      * **Composite Data Type**: Chosen false Parent Type Composite Data Type
        + Ist eine Kategorie die wir nicht erkennen
  + **SEFF**:
    - **Cat6**: Falsely chosen interface and operationSignature
      * **External Call Action**: Chosen false 1) Role External Service 2) Called Service External Service
      * **L&D Mistake:** iSystemAusstelung\_erstelleAusstellungskonzept – External Call Action erstelle Ausstelungskonzept zeigt auf ISystemAusstellung.erstelleAusstellungskonzept (?Passt das überhaupt Interface mässig oder müsste man da was umstecken?)
    - **~~Cat8~~**~~: Too many External Call Actions~~ ?Ist doch das selbe wie cat6: Wenn man sich vorstellt das EAA designed wird dann mussen die External calls ja eh neu gemacht werden. Ob jetzt ein ECA falsch verbunden wird oder zu viele sind ist ja das selbe, und zwar das der SEFF falsch gemacht wurde
      * **External Call Actions**: Too many External Call Actions
  + **BPusagemodell**: Hier sind keine Fehler möglich, da es hier nur darum geht das die Rolle gewisse Systemfunktionen aufruft, Die Systemfunktionen selber können Falsch sein. Das aber genug systemfunktionen da sein müssen um die Arbeit zu erfüllen sollte angenommen werden ?Stephans Meinung
  + **Regeln für Fehler**: (Betrachten unter dem Aspekt was wir erkennen und was nicht)
    - Es muss **ein anderer Datentyp fließen der im BP bekannt** (Datenmodell parallel aufmachen hierfür) ist und **für den die Rolle in diesem AS keinen Zugriff hat** (oder generell keinen Zugriff hat?)
      * (Parallel Rollenmodell aufmachen und schauen ob Rolle Zugriff auf den anderen Datentyp hat) => Eig. Sind die Vorgegeben Datenobjekte auf die zugegriffen werden darf spezifisch für den AS, also die aufgerufenen ELSC. Heißt also selbst wenn die Rolle auf den Datentypen zugriff hat ist es trotzdem Falsch wenn dieser nicht in diesem AS erlaubt ist. (Da der BP immer richtig ist müsste er dadrin stehen, da er es nicht tut ist die EAA falsch)
    - Immer BPUsageModell und AS wo der Fehler auftritt mit aufschreiben
    - 1 Fehler zu jeder Kategorie reicht
  + **FAQ**
    - **Ausstellungskonzeptobjekt** wird nie verwendet? Was ist überhaupt der Sinn davon im IT System
    - **„Öffentliche und private Foschungsliteratur“**.InnerTypeCollectionDT = Fremdes Ausstellungsobjekt? (Siehe UC2)
    - **UC2**: **AS** **Literatur nach verwandten Werken recherchieren** – Wurde nicht modelliert woher die DO Literatur und öffentliche und private Forschungsliteratur herkommen
    - **UC3**~~:~~ **~~AS Leihvertrag Verhandeln~~**, **AS** **Leihvertrag prüfen, AS Leivertrag unterschreiben, AS Leihvertrag an Museum verschicken** haben keine ELSC

# BP Evolution

* Ziel hiervon: ist es zu zeigen, dass man einfach die Zugriffberechtigungen neu extrahiert und dann die Analyse auf der Architektur durchführt
  + Ist eine Veränderung in BP, keine in EAA -> Heißt Veränderungen im BP sollen bereits bestehende Service Calls nutzen
* Welche Elemente in BPMN haben Einfluss auf Berechtigungen der Rollen. Diese Kategorien in dem Evolutionsbeispiel abdecken
  + Cat1: Hinzufügen/Entfernen von Input DO im BP
  + Cat2: Hinzufügen/Entfernen von Output DO im BP
* 1) UC 5 – fremdes Werk wird abgeholt
  + Vorher: Protokoll wird vom Kurator und Restaurator geschrieben
    - Problem a: Vereinzelt traten Fälle auf wo nach dem Transport Unstimmigkeiten am Werk auftraten die nicht zum beschriebenen zustand im Protokoll vor dem Transport übereinstimmten
    - Problem b: Gesetze haben bestimmt dass die Verantwortlichkeit des Transports von fremden Werken in der Zuständigkeit der ausleihenden Kunsthalle liegt -> Aus diesem Grund soll der Transport ebenfalls protokolliert werden von eine Kurier der Kunsthalle um sich rechtlich abzusichern
  + Nahher: Neue Rolle Kurier welche einen AS hat bei dem sie den Transport begleitet und diesen protokolliert (Für diesen Fall müssen die BP noch angepasst werden)
    - Input: Protokoll das vorher geschrieben wurde um es zu erweitern
    - Output: Erweitertes Protokoll
  + Was soll die Analyse am ende ausgeben?: Das man x Schritte benötigt um die Analyse auszuführen (Im Fall der BP evolution -> keinen Schritt, Ausführung direkt möglich)

# Sonstiges:

**Logical&Design Mistakes für LoyaltyCard**

* **Wieviele Mistakes brauchen wir?** Wir sollten damit eine gewisse Abdeckung erreichen – Alle Relevanten Fälle abdecken. **Relevante Kategorien Bilden und für jede ein Mistake bauen. – An welchen Stellen können Logical und Design Mistkes entstehen in EAA - Kategorien:**
  + **Assembly diagram/System diagram**:
    - **Cat1**: Falsely connected Required/Provided Role between a) System and Subsystem or b) Subsystems or c) Subsystem and Components or d) Components
      * **a) c) Required Delegation Connector:** Connected to false 1) Assembly Context Required Delegation Connector 2) Outer Required Role Provided Delegation Connector
        + **L&D Mistake**: Store.provided.iOnlineShop verbinden mit CashDesk.providedIOfflineShop
      * **a) c) Provided Delegation Connector:** Connected to false 1) Assembly Context Provided Delegation Connector 2) Outer Provided Role Provided Delegation Connector
        + **L&D Mistake:** Marketing.provided.IAdvertisement verbinden mit AdvertisementAdmissionDB.provided.iAdvertisementAdmissionDB
      * **b) d) Assembly Connector**: Connected to false 1) Providing Assembly Context Assembly Connector 2) Provided Role Assembly Connector
        + **L&D Mistake:** LoyaltyManagementLogic.required.iLoyaltyCustomerDB verbinden mit LoyaltyOrderDB.provided.ILoyaltyOrderDB
  + **Repository**:
    - **Cat2**: Falsely chosen Interface for SubSystem/Component
      * **Operation Provided Role**: Chosen false Provided Interface Operation Provided Role
        + **L&D Mistake:** CustomerDB.provided.ICustomerDB – Provided Interface operation Provided Role setzen aufILoyaltyCustomerDB
      * **Operation Required Role**: Chosen false Required Interface Operation Required Role
        + **L&D Mistake:** LoyaltyManagementLogic.required.ILoyaltyCustomerDB - Required Interface Operation Required Role setzen auf ICustomerDB
    - **Cat3**: Falsely chosen Parameter and Return of an Operation Signature
      * **Operation Signature - Parameter**: Chosen false Data Type Parameter
        + **L&D Mistake:** IMarketingAnalysis.getCustomerProfiles - Parameter orders setzen auf Collection Orders
      * **Operation Signature**: Chosen false Return Type operation Signature
        + **L&D Mistake:** IMarketingAnalysis.getLoyaltyOrders – Return Type setzen auf Collection orders

**Funktion sollte ja auch heißen GetAllOrders()**

* + - **Cat4**: Too many Parameters of an Operation Signature
      * **Operation Signature - Parameter**: Too many Data Type Parameter
        + **L&D Mistake:** IMarketingAnalysis.getCustomerProfiles - Parameter
    - **Cat5**: Falsely build Composite and Collection Data Types
      * **Collection Data Type**: Chosen false Innter Type Collection Data Type
        + **L&D Mistake:**
      * **Composite Data Type – Inner Declaration**: Chosen false Datatype Inner Declaration
        + **L&D Mistake:**
    - **Cat6**: Falsely build inheritance Hierarchy
      * **Composite Data Type**: Chosen false Parent Type Composite Data Type
        + ?Können wir dies überhaupt erkennen – Weglassen?
  + **SEFF**:
    - **Cat7**: Falsely chosen interface and operationSignature
      * **External Call Action**: Chosen false 1) Role External Service 2) Called Service External Service