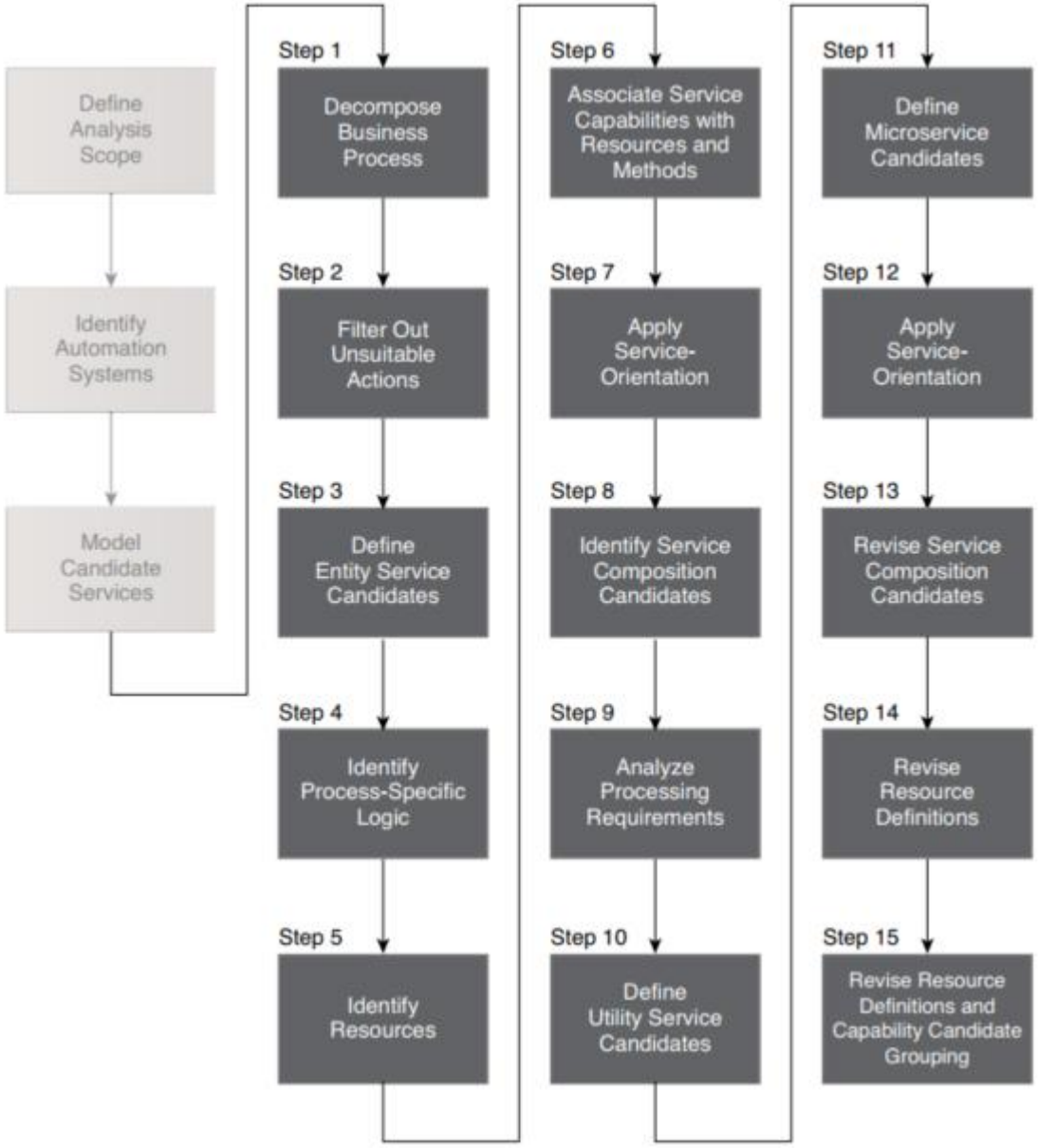
Analysis and Modeling with REST Services and Microservices

hungdn@ptit.edu.vn

7.1 REST Service Modeling

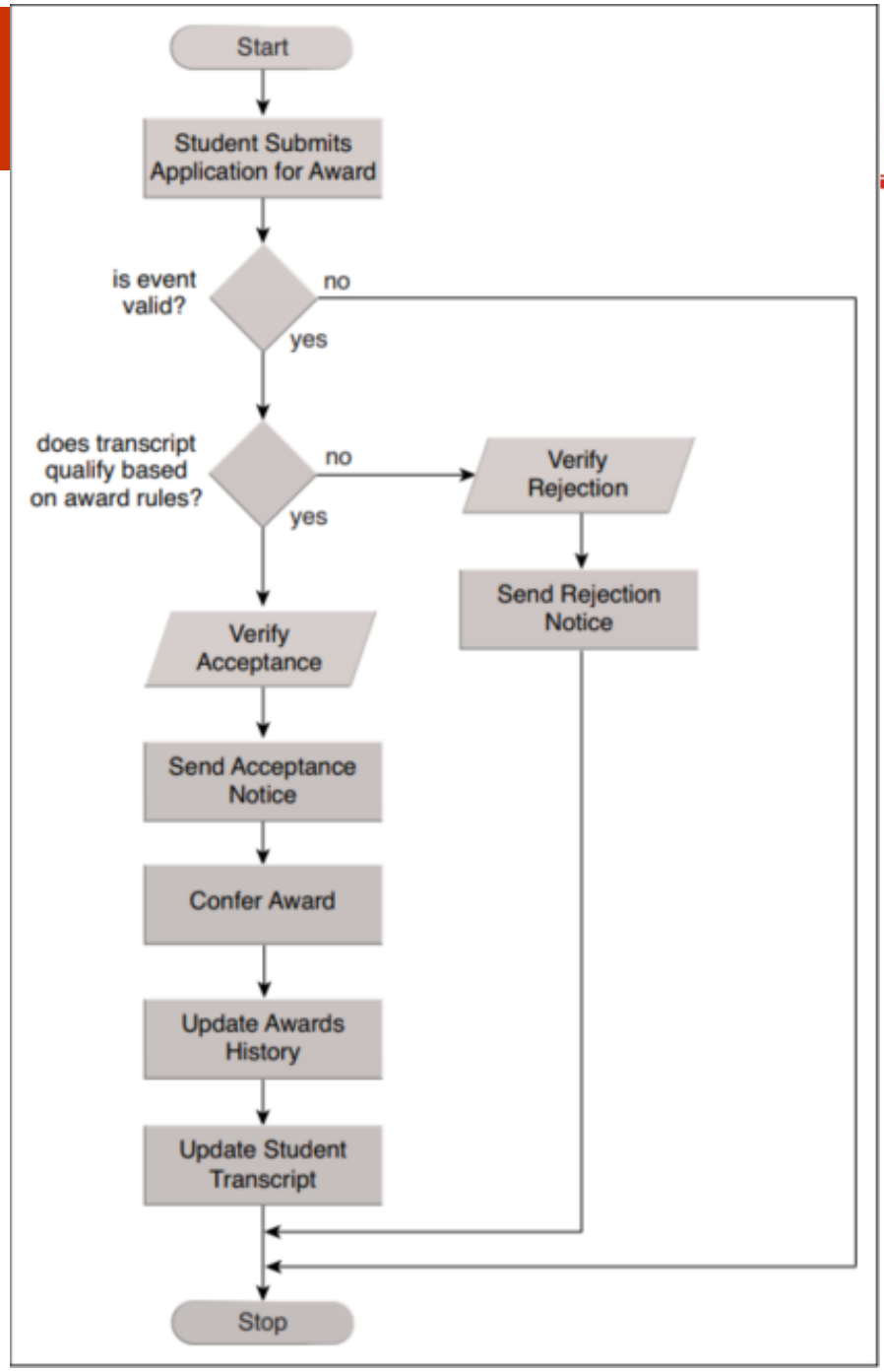
REST Service Modeling Process

- **The incorporation of resources and uniform contract features adds new dimensions to service modeling.**



Case study

- **MUA architects adopt SOA to unify systems & data. Focus on entity services to track campus information.**
- **Requirements:**
  - Start at main campus → Expand to others.
  - Virtual machines for scalability.
  - MUA must ensure accurate award conferrals for students
- **Solution**
  - SOA team models the **Student Achievement Award Conferral** process using REST service



Step 1: Decompose Business Process

- **Break the process into a series of granular actions.**
- **Student Award Conferral Process Breakdown**
  - Application Initiation & Event Verification
  - Award & Student Transcript Verification
  - Decision: Approve or Reject
  - Recording & Documentation

CASE STUDY EXAMPLE

The original Student Award Conferral business process is broken down into the following granular actions:

- Initiate Conferral Application
- Get Event Details
- Verify Event Details
- If Event is Invalid or Ineligible for Award, End Process
- Get Award Details
- Get Student Transcript
- Verify Student Transcript Qualifies for Award Based on Award Conferral Rules
- If Student Transcript Does Not Qualify, Initiate Rejection
- Manually Verify Rejection
- Send Rejection Notice
- Manually Verify Acceptance
- Send Acceptance Notice
- Confer Award
- Record Award Conferral in Student Transcript
- Record Award Conferral in Awards Database
- Print Hard Copy of Award Conferral Record
- File Hard Copy of Award Conferral Record

Step 2: Filter Out Unsuitable Actions

- **Identify actions unsuitable for REST automation.**
- **Filtered Out (Manual or Non-Automatable Actions)**
  - Manually Verify Rejection
  - Manually Verify Acceptance
  - Confer Award
  - Print & File Hard Copy of Records

CASE STUDY EXAMPLE

After assessing each of the decomposed actions, a subset is identified as being unsuitable for automation or unsuitable for service encapsulation, as indicated by the crossed-out items.

- Initiate Conferral Application
- Get Event Details
- Verify Event Details
- If Event is Invalid or Ineligible for Award, End Process
- Get Award Details
- Get Student Transcript
- Verify Student Transcript Qualifies for Award Based on Award Conferral Rules
- If Student Transcript Does Not Qualify, Initiate Rejection
- ~~Manually Verify Rejection~~
- Send Rejection Notice
- ~~Manually Verify Acceptance~~
- Send Acceptance Notice
- ~~Confer Award~~
- Record Award Conferral in Student Transcript
- Record Award Conferral in Awards Database
- Print Hard Copy of Award Conferral Record
- ~~File Hard Copy of Award Conferral Record~~

Step 3: Define Entity Service Candidates

- **Separate concerns – identify agnostic vs. non-agnostic logic**
- **Identify reusable service capabilities for REST modeling**
- **Group service candidates by:**
  - Functional **relationships**
  - Business-centric vs. utility-centric **context**
  - Service inventory needs

CASE STUDY EXAMPLE

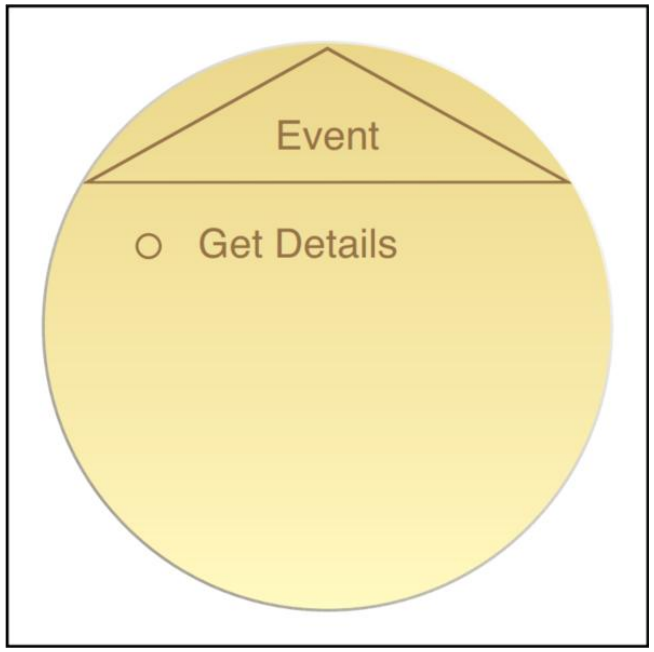
By analyzing the remaining actions from Step 2, the MUA service modeling team identifies and categorizes those actions considered agnostic. Those that are classified as non-agnostic are in bold:

- **Initiate Conferral Application**
- Get Event Details
- **Verify Event Details**
- **If Event is Invalid or Ineligible for Award, Cancel Process**
- Get Award Details
- Get Student Transcript
- **Verify Student Transcript Qualifies for Award Based on Award Conferral Rules**
- **If Student Transcript Does Not Qualify, Initiate Rejection**
- Send Rejection Notice
- Send Acceptance Notice
- Record Award Conferral in Student Transcript
- Record Award Conferral in Awards Database
- Print Hard Copy of Award Conferral Record

Step 3: Define Entity Service Candidates (2)

Event Service Candidate

- Get Event Details

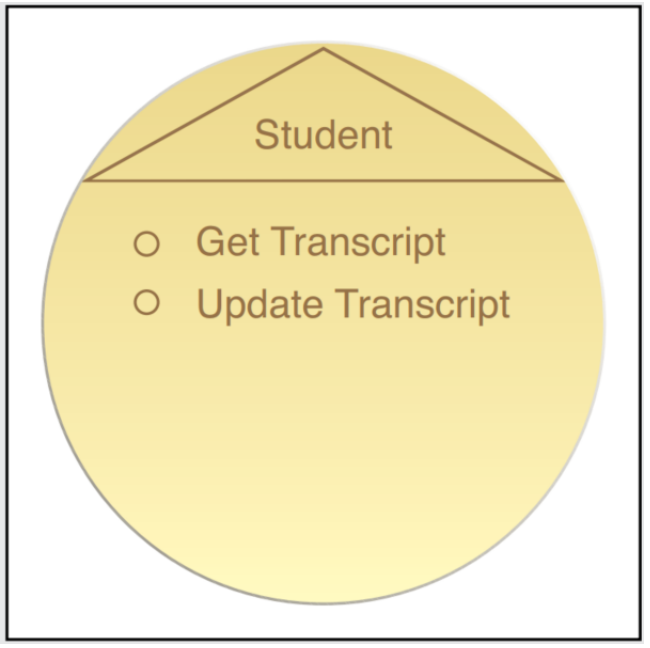


Award Service Candidate

- Get Award Details
- Record Award Conferral in Awards Database

Student Service Candidate

- Get Student Transcript
- Record Award Conferral in Student Transcript

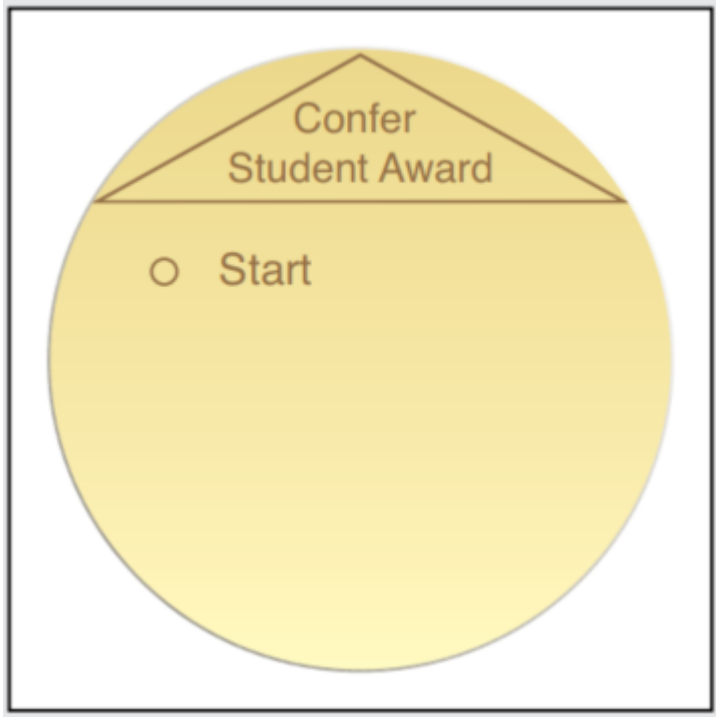


Step 4: Identify Process-Specific Logic

- **Process-specific logic is separated into its own service layer**
- **Logic that is unique to the Student Award Conferral process**
  - Initiate Conferral Application
  - Other process-specific actions
- **Confer Student Award Task Service Candidate**
  - **Start capability** is triggered by an **external software program** - Composition initiator

Confer Student Award Service Candidate

- Initiate Conferral Application



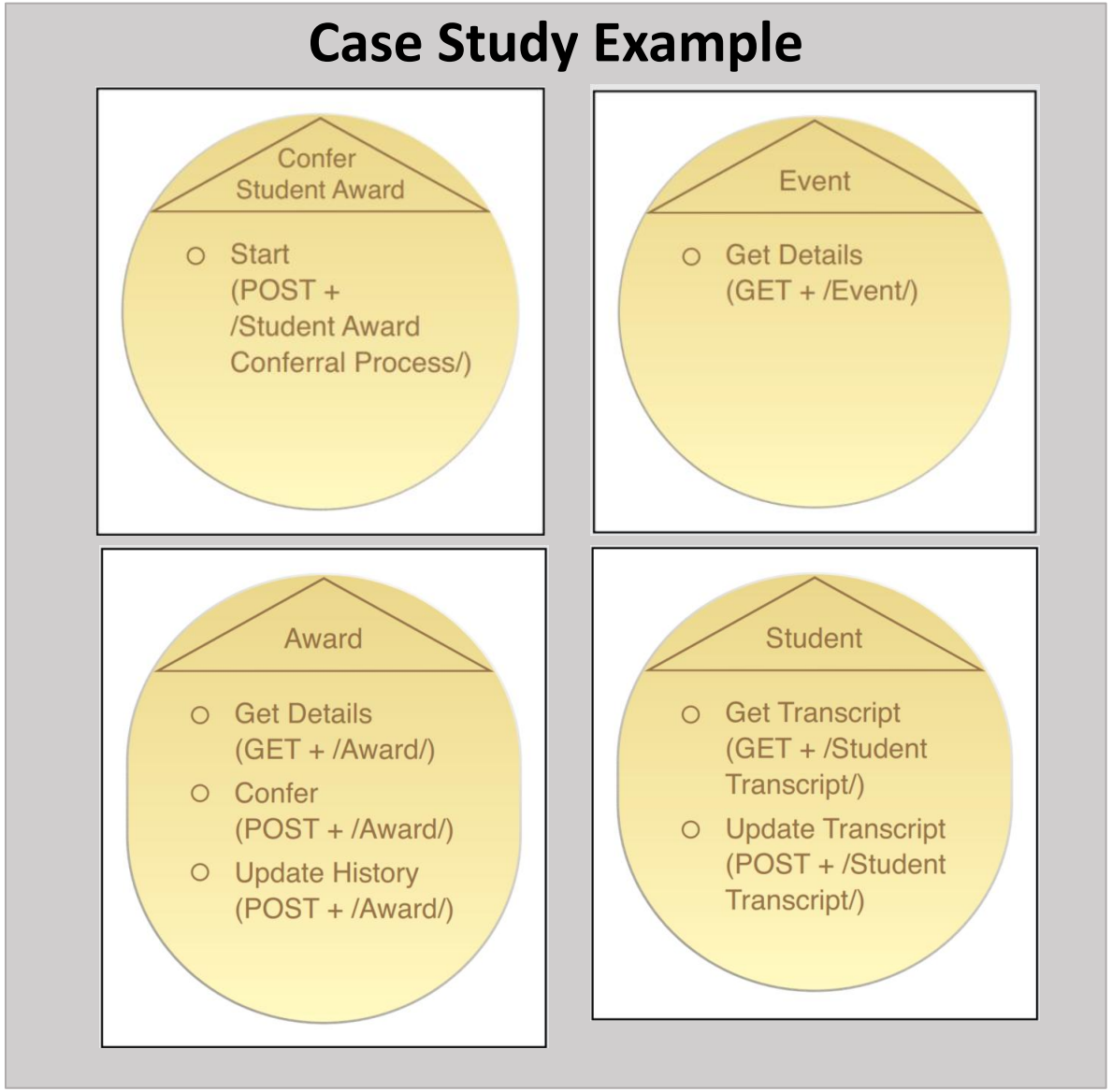
Step 5: Identify Resources

- **List functional contexts to define resources and classify resources as agnostic (multipurpose) or non-agnostic (single-purpose)**
- **Resources identified in this step include**
  - /Student Award Conferral Process/
  - /Conferral Application/
  - /Event/
  - /Award/
  - /Student Transcript/
- **Mapping business entities to resources**

Entity	Resource
Event	/Event/
Award	/Award/
Student	/Student Transcript/

Step 6: Associate Service Capabilities with Resources and Methods

- **Link service capability candidates (Steps 3 and 4) with resources (Step 5) & available uniform contract (HTTP) methods**
- **Refine service contracts with uniform contract definitions**
  - Confer Student Award Service (Task)
  - Event Service (Entity)
  - Award Service (Entity)
  - Student Service (Entity)



Step 7: Apply Service-Orientation

- **Business process documentation helps define service scope. Consider Service-Orientation principles to refine service capabilities**
- **Adjust functional service boundaries based on implementation constraints**

Case Study Example

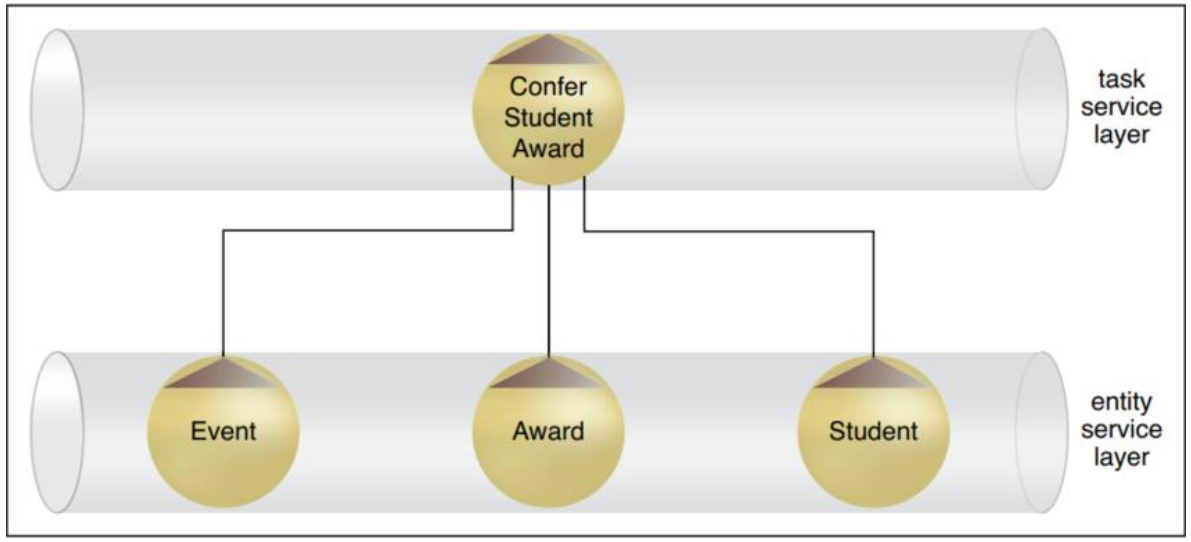
- **MUA architects identify resource dependencies on a large legacy system**
- **Service Autonomy is impacted due to system constraints**
- **SOA architects analyze deployment environment to optimize service design**

Step 8: Identify Service Composition Candidates

- **Document common service interactions during business process execution**
  - Map interactions to service capability candidates to identify composition scenarios
- **Analyze service composition size, complexity, and data exchange needs**
  - Identify **required media types** for uniform contract modeling.
  - Review and refine service boundaries if compositions become too complex.

Case Study Example

- MUA team analyzes success & failure scenarios in the Student Award Conferral process



Step 9: Analyze Processing Requirements

- **Shift focus from business logic to underlying application logic**
- **Identify utility-centric vs. business-centric resources & actions**

Case Study Example

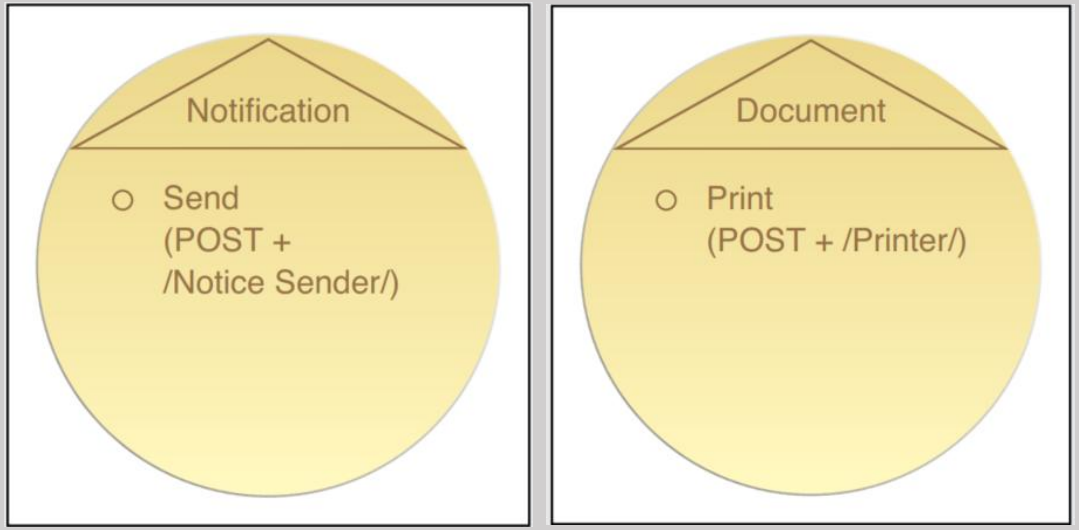
- **MUA team reviews processing logic for service candidates**
- **No new utility-centric actions found beyond**
- **Critical issue found**
  - Verify Student Transcript Qualifies for Award relies on an external Rules service
  - This service experiences delays and failures due to high usage
  - Business analysts stress the need for immediate & legally binding verification
- **Solution: Move verification logic to a dedicated microservice**

Step 10: Define Utility Service Candidates (and Associate Resources and Methods)

- **Group utility-centric processing steps based on pre-defined contexts. Utility service candidates are logically grouped by**
  - Legacy system association
  - Solution component association
  - Functional type grouping
- **Establish preliminary utility service layer, associating candidates with resources & methods**
- **Utility service modeling is challenging due to lack of predefined business models**

Case Study Example

Notification and Document Utility Service Candidate

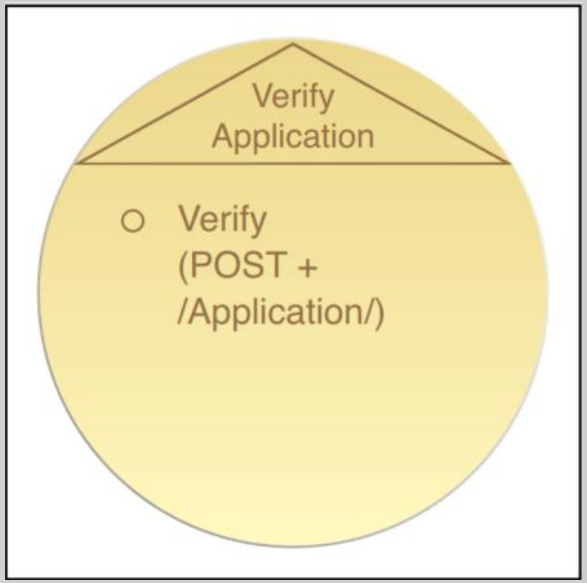


Step 11: Define Microservice Candidates (and Associate Resources and Methods)

- **Identify non-agnostic processing logic that qualifies for microservice encapsulation**
- **Microservices offer independent & autonomous service implementation**
- **Key considerations for microservice candidates**
  - Autonomy
  - Performance
  - Reliability & Failover
  - Versioning & Deployment

Case Study Example

Verify Application Microservice Candidate

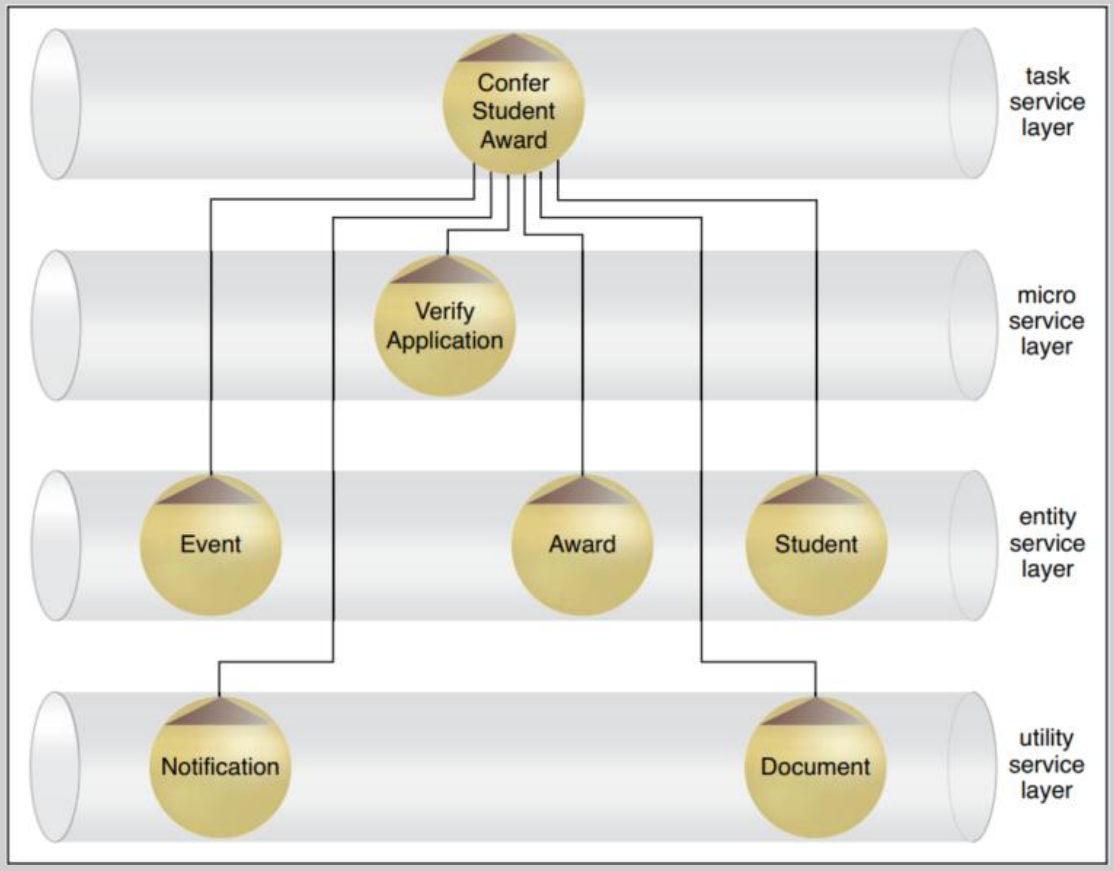


Step 12 + 13 + 14: Apply Service-Orientation and Revise

- **Step 12: Apply Service-Orientation**
- **Step 13: Revise Candidate Service Compositions**
- **Step 14: Revise Resource Definitions and Capability Candidate Grouping**

Case Study Example

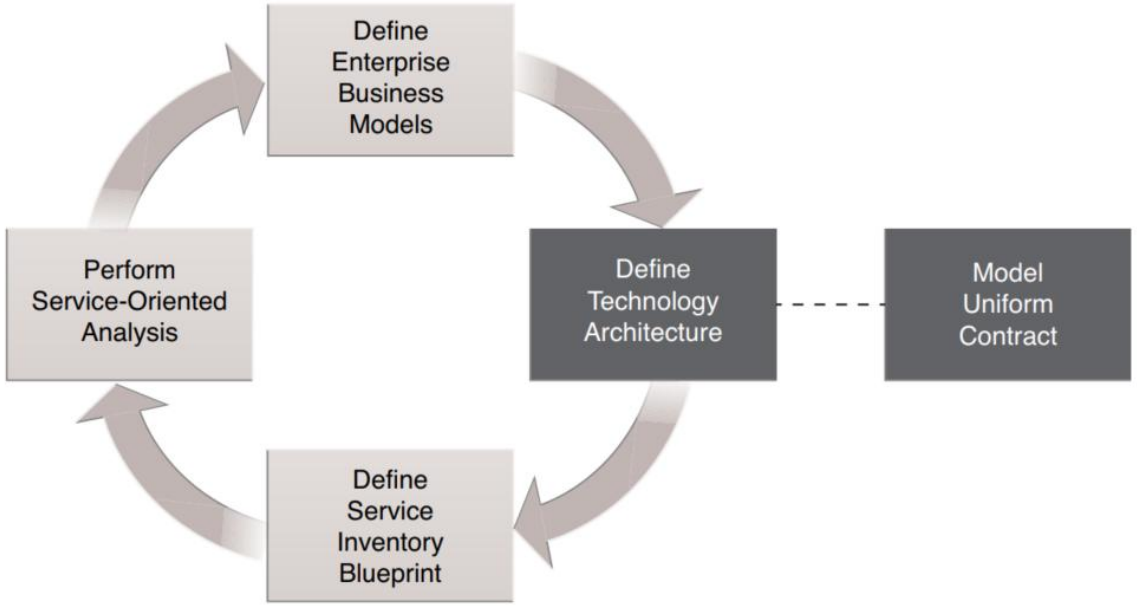
The Confer Student Award service composition



7.2 Additional Considerations

Uniform Contract Modeling and REST Service Inventory Modeling

- **Uniform contract standardizes service interactions within a service inventory**
  - Service inventories can be single or domain-based, requiring different contracts
- **Uniform contract modeling aligns with REST service inventory analysis**
  - Helps define methods, media types, policies, and composition rules
  - Supports standardization of reliability, security, and transaction handling

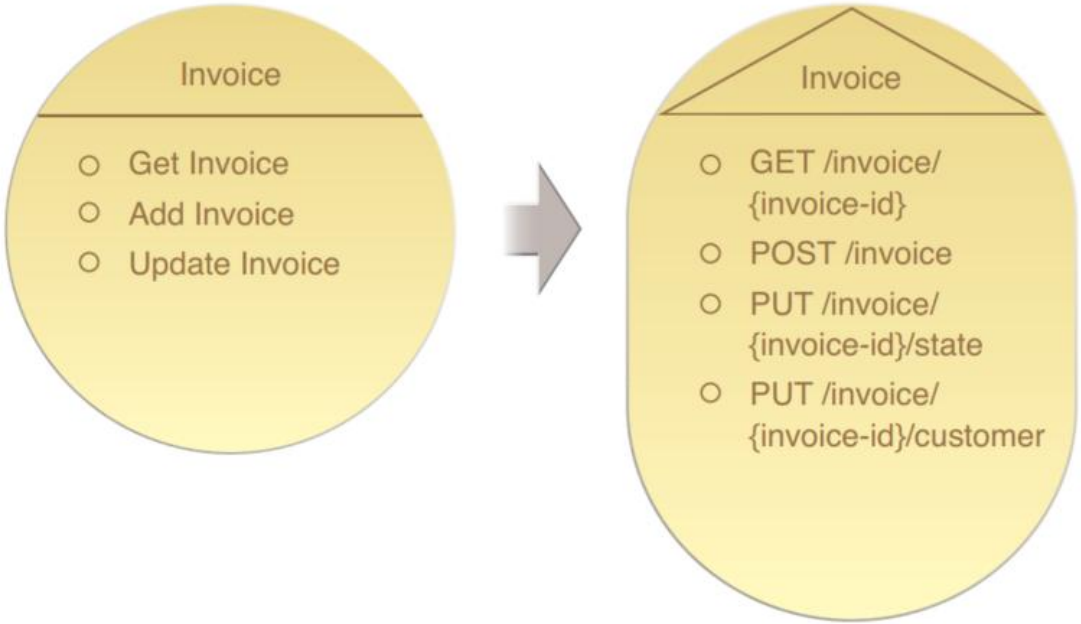


REST Constraints and Uniform Contract Modeling

- REST constraints shape the uniform contract during service analysis
  - **Stateless:** Ensure services do not retain client state between requests.
  - **Cache:** Identify reusable responses to avoid redundant processing.
  - **Uniform Contract:** Ensure all methods & media types are reusable.
  - **Layered System:** Understand if services interact directly or via middleware.
- Influence depends on:
  - Service inventory architecture definition.
  - Business automation requirements.

REST Service Capability Granularity

- **Fine-Grained Actions: Clearly defined with a specific purpose, but can include variations.**
  - **SOAP-Based Services:** WSDL operations handle functional variations
  - **REST Services:** Must align with uniform contract methods & media types
- **Impact of REST Constraints**
  - Limits granularity based on methods & media types
  - May require more fine-grained service capabilities
- **Example:**
  - Splitting **PUT /invoice/** into separate actions for state and customer updates



Resources vs. Entities

- **Entities: Business-centric, derived from models (e.g., ER diagrams, data models).**
  - Examples: Invoices, claims, customers.
  - Coarse-grained, can encapsulate other entities (e.g., invoice detail).
- **Resources: Can be business or non-business-centric, any "thing" tied to business automation logic.**
  - Examples: Data, documents, or operations needed for service execution
  - Often fine-grained, not typically encapsulating other resources
- **REST Service Modeling: Aligns resources with business entities to ensure consistency**
- **Pure modeling perspective: Helps maintain alignment with business artifacts and their developing automation needs.**

Q & A