Ishmael Thomas

Business & Interface Processing (Service Layer)
Design & Development with Java

Overview

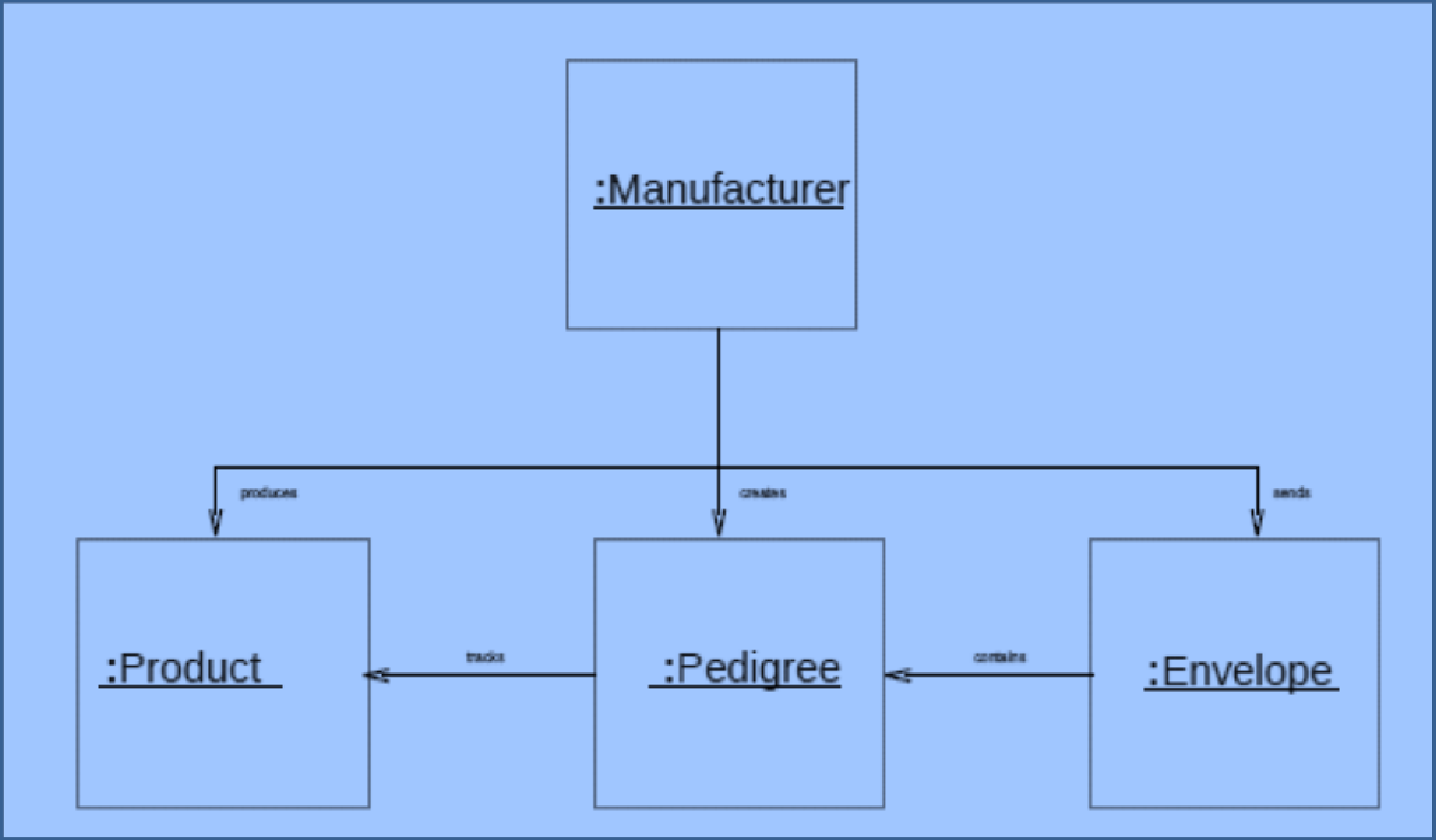
This is an e-Pedigree application design document to track and trace drug prescriptions with pedigrees (electronic documents) throughout a pharmaceutical supply chain. Members of a pharmaceutical supply chain are manufacturers, distributors, and retailers. According to GS1, each supply chain member is responsible for producing separate pedigrees for their logistical responsibilities (shipping and receiving). In addition, each supply chain member is responsible for completing the pedigree chain of custody. For example, the retailer would complete the chain of custody with their own pedigree and two additional pedigrees from the manufacturer and distributor. This application design document builds upon the following high priority use cases:

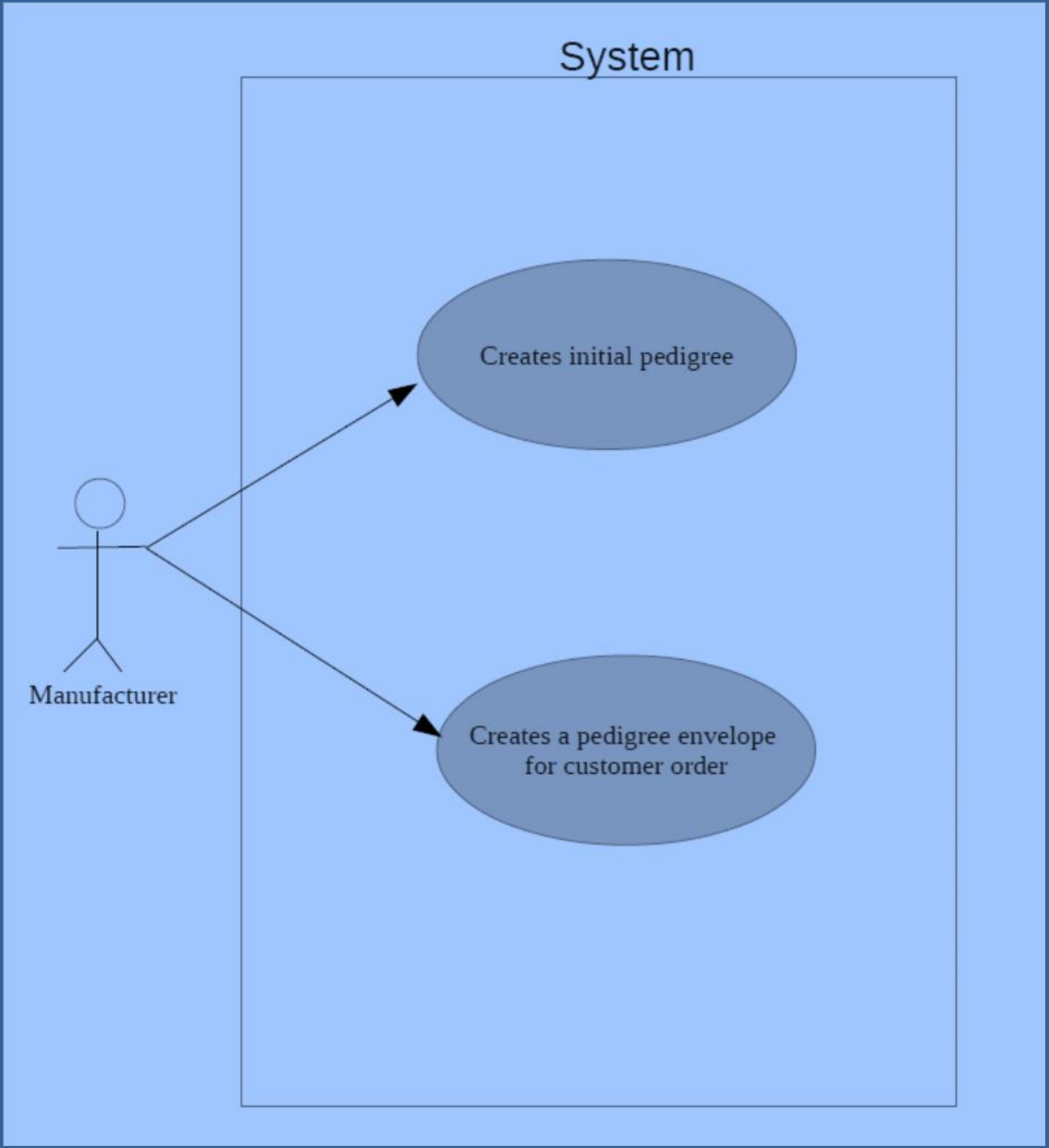
•**Manufacturer creates an initial pedigree**
•**Manufacturer creates a pedigree envelope for customer order**

The scope of this application design document is to develop the following four application domain classes for the high priority use cases:

•**Envelope**
•**Manufacturer**
•**Pedigree**
•**Product**

Domain Class Diagram





Use Case Diagram:

UC-1 Manufacturer creates an initial pedigree

UC -2 Manufacturer creates a pedigree envelope for customer order

Use Case 1 - (Fully Dressed): Manufacturer creates an initial pedigree

Main goal path:

1.1 Manufacturer generates the initial pedigree, which contains the serial number product information and item information (identifies the specific items represented by the pedigree) elements.
1.2 Manufacturer adds transaction information for the sale and signs the pedigree.

Alternative goal path:

1.1 Manufacturer generates an initial pedigree in a hard-copy format.
1.2 Manufacturer creates an electronic format using a word processing system.
1.3 Manufacturer scans and uploads either option into the e-Pedigree application when it comes available.

Functionality:

The functionality bridges requirements and features with this use case. The following depicts the features for this use case:

1.1 Create pedigree.
1.2 Add information to pedigree.
1.3 Certify (digitally sign) pedigree.
1.4 Electronically authenticate pedigrees.
1.5 Manually authenticate transactions that were not electronic.

Use Case 1 - (Summarized): Manufacturer creates an initial pedigree

Goal	Manufacturer creates an initial pedigree
Actor(s)	Manufacturer
Description	Initial pedigree
Frequency/ Trigger	Occurs when manufacturer is ready to ship prescription drug items to a distributer or directly to a retailer.
Pre-processing	The application has been successfully installed on the manufacturer's machine.
Post-processing	The pedigree has been successfully created with initial pedigree elements representing a manufacturer's pedigree.
Included use case	None
Extended use case	None
Technology	JAVA graphical user interface or web browser

Use Case 2 - (Fully Dressed): Manufacturer creates a pedigree envelope for customer order

Main goal path:

2.1 Manufacturer generates the pedigree envelope, which contains the version, serial number, date, source routing code, and destination routing code elements.
2.2 Manufacturer adds a container element for the case or tote.
2.3 Manufacturer adds one pedigree handle element for each pedigree associated with products in the case. Repeat for each case in the shipment.
2.4 Manufacturer adds each pedigree representing each physical prescription drug item in the shipment to the pedigree envelope.

Alternative goal path:

2.1 Manufacturer generates a pedigree envelope in a hard-copy format containing version, serial number, date, source routing code, and destination routing code element. In addition, the manufacturer adds container and handle elements to the hard-copy formats. Last, the manufacturer adds a pedigree (standard or alternative) representing each physical prescription drug item in the shipment to the pedigree envelope.
2.2 Manufacturer creates an electronic format using a word processing system containing – version, serial number, date, source routing code, and destination routing code element. In addition, the manufacturer adds container and handle elements to the alternative electronic formats. Last, the manufacturer adds a pedigree (standard or alternative) representing each physical prescription drug item in the shipment to the pedigree envelope.
2.3 Manufacturer scans and uploads either option into the e-Pedigree application when it comes available.

Functionality:

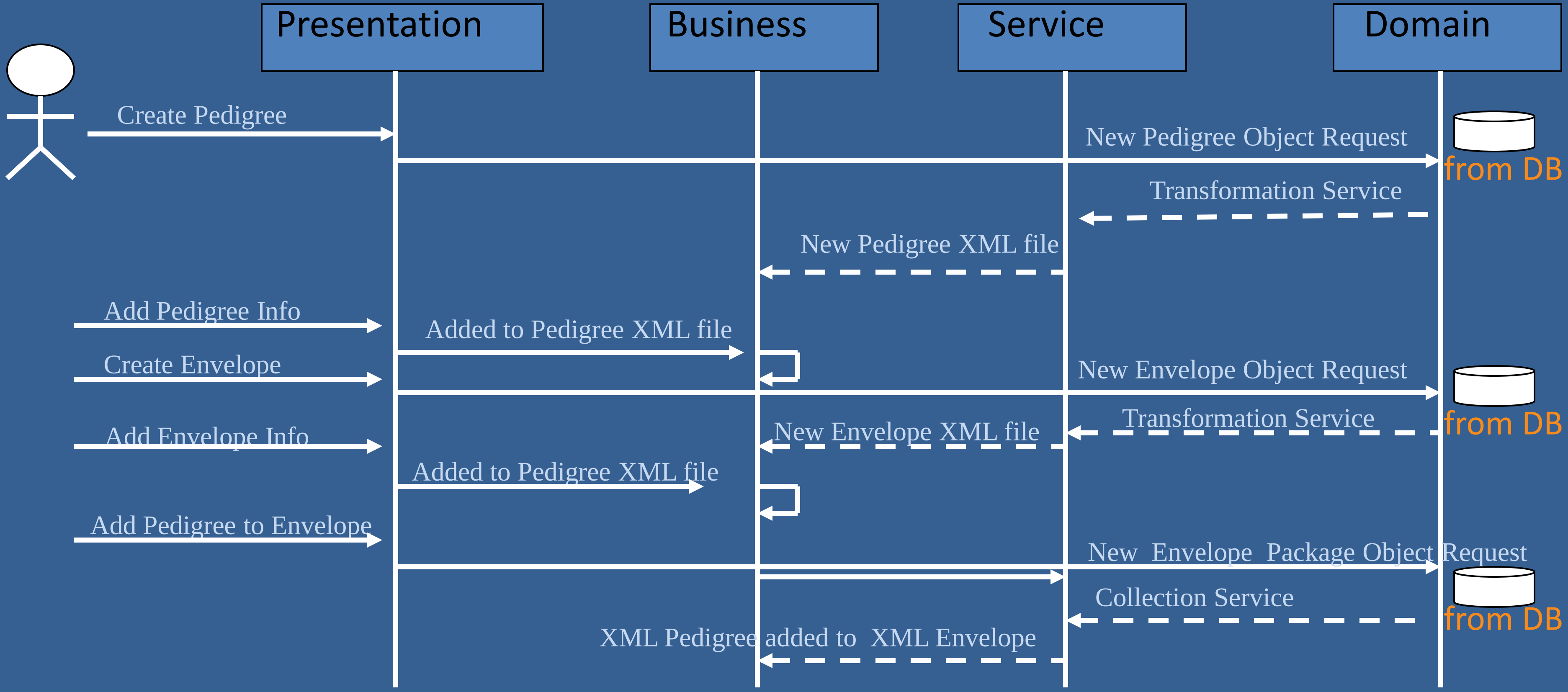
The functionality bridges requirements and features with this use case. The following depicts the features for this use case:
2.1 Send pedigrees for products in shipment to customer.
2.2 Verify products received against authenticated pedigrees.
2.3 Certify (digitally sign) pedigree for receipt and authentication.
2.4 Electronically authenticate pedigrees.
2.5 Manually authenticate transactions that were not electronic.

Use Case 2 – (Summarized):
Manufacturer creates a pedigree envelope for customer order

Goal	Manufacturer creates a pedigree envelope for customer order
Actor(s)	Manufacturer
Description	Pedigree envelope
Frequency/ Trigger	Occurs when the manufacturer transmits a collection of pedigrees associated with an outbound customer shipment.
Pre-processing	Successfully installation of application and creation of an initial pedigree.
Post-processing	The pedigree envelope has been successfully created and contains a copy of the manufacturer's initial pedigree.
Included use case	None
Extended use case	None
Technology	JAVA graphical user interface or web browser

Use Case Driven
Services Sequence Diagram
(SSD)

UC1 & UC2 with SSD



Interfaces

java.util.*
Interface Set

public interface **Set**

A Set is a Collection that cannot contain duplicate elements. It models the mathematical set abstraction.

Code example:

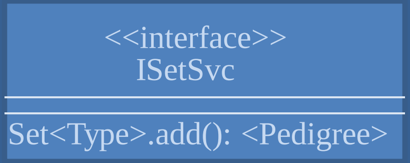
```
public class Envelope {
  public static void main(String[] args) {
    Set<Pedigree> np = new HashSet<Pedigree>();
      for (Pedigree a : Pedigree) {
        if (!np.add(a))
          System.out.println("New pedigree added to envelope: " + a);
     }
  }
}
```

Design example:

ISetSvc extends Set

Design usage:

Enable the Envelop object to contain a set “collection” of non-duplicated pedigrees.
Support Use Case 2 – Enabler for use case 2 functionalities.



javax.xml.bind
Interface Marshaller

public interface **Marshaller**

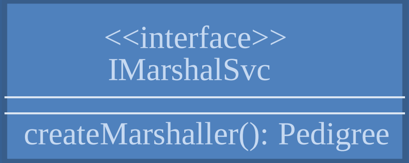
The Marshaller inteface is responsible for governing the process of serializing Java content trees back into XML data.

Code example:

```
JAXBContext jc = JAXBContext.newInstance( "com.e_pedigree.pedigree" );
Marshaller m = jc.createMarshaller();
Object element = u.unmarshal( new File( "pedigree.xml" ) );
Unmarshaller u = jc.createUnmarshaller();
```

Design example:
IMarshalSvc extends Marshaller

Design usage:
Serialize Pedigree and Envelop object states into XML files.
Support Uses Cases 1 & 2 – Enabler for use case functionalities.



javax.xml.soap
Interface SOAPBody

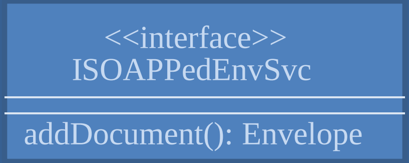
public interface **SOAPBody extends SOAPElement**

The SOAPBody inteface is responsible for the contents of the SOAP body element in a SOAP message.

Code example:
SOAPBodyElement docElement = pedigreeEnvelope.addDocument(Envelope);

Design example:
ISOAPPedEnvSvc extends SOAPBody

Design usage:
Adds Pedigree and Envelop XML files into an SOAP Message body for message delivery.
Supports Use Case 2: – functionality 2.1



javax.xml.ws
Interface Provider<T>

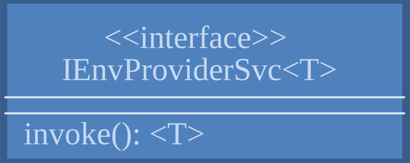
public interface **Provider<T>**

The Provider inteface supports web service endpoints for SOAP messages.

Interface specification: - No code example
Constant Variables – none.
Methods – invoke(T request); Invokes an operation an operation according to the contents of the request message
Parameters – the request message or message payload.
Returns – the response message or message payload.

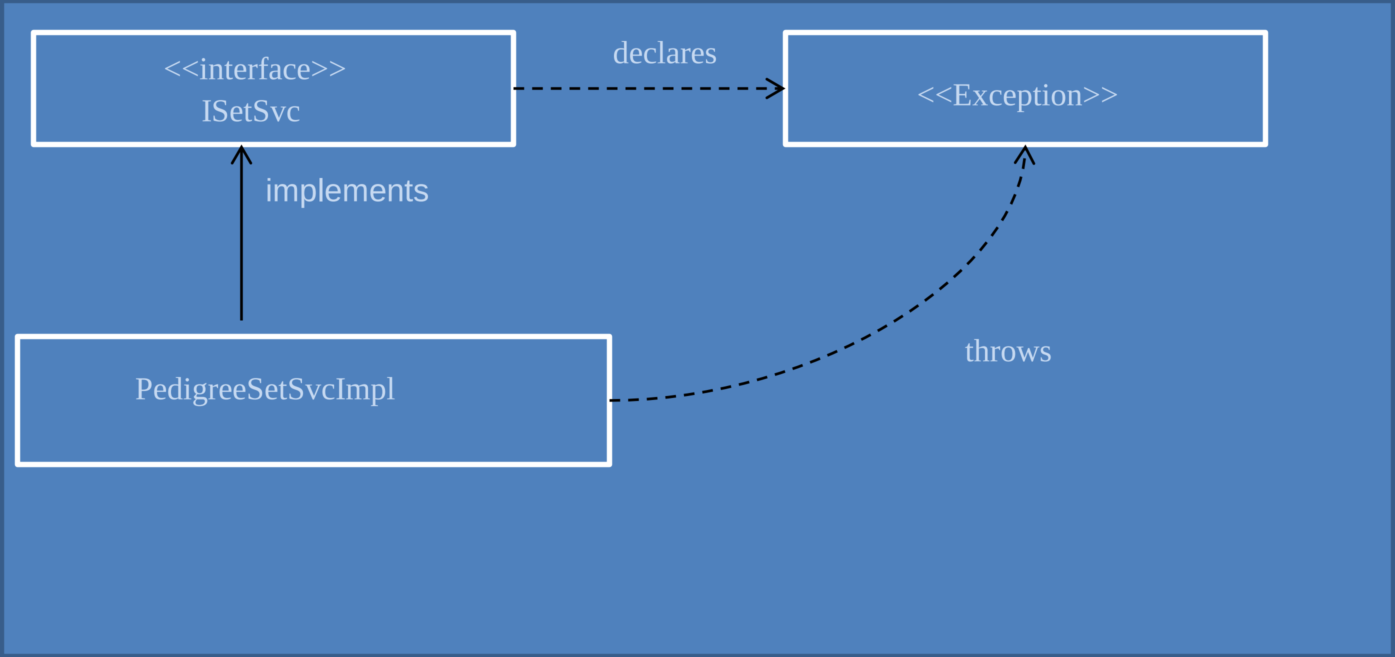
Design example:
IEnvProviderSvc<T> extends Provider<T>

Design usage:
Provider web services delivery mechanism for SOAP message with XML Pedigree and Envelop content.
Support Use Case 2 – Enabler for use case 2 functionalities.

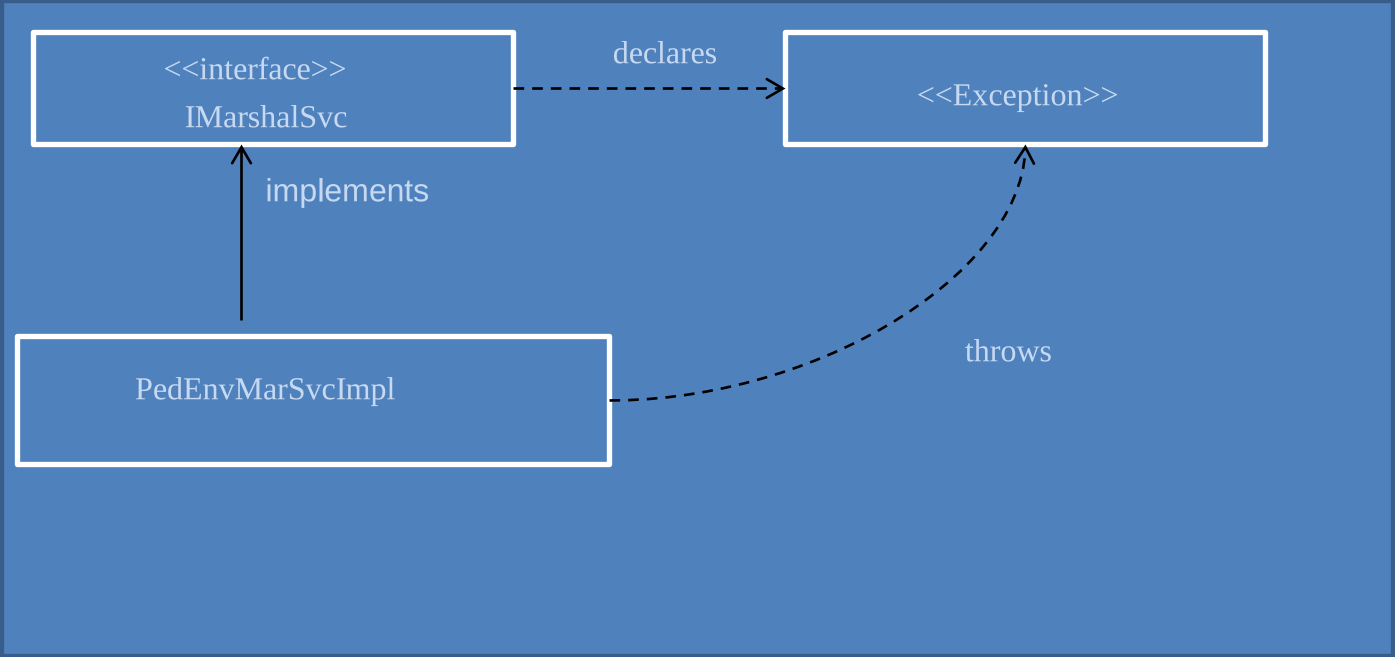


Services

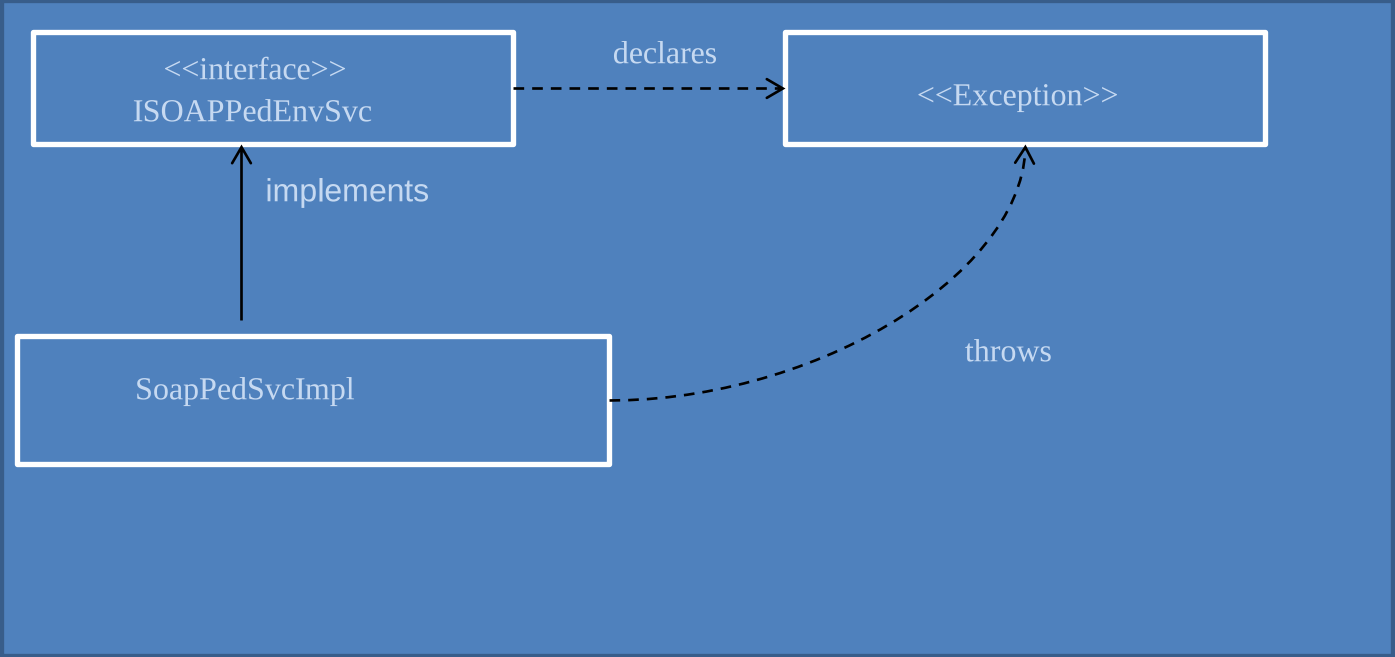
Collection Service



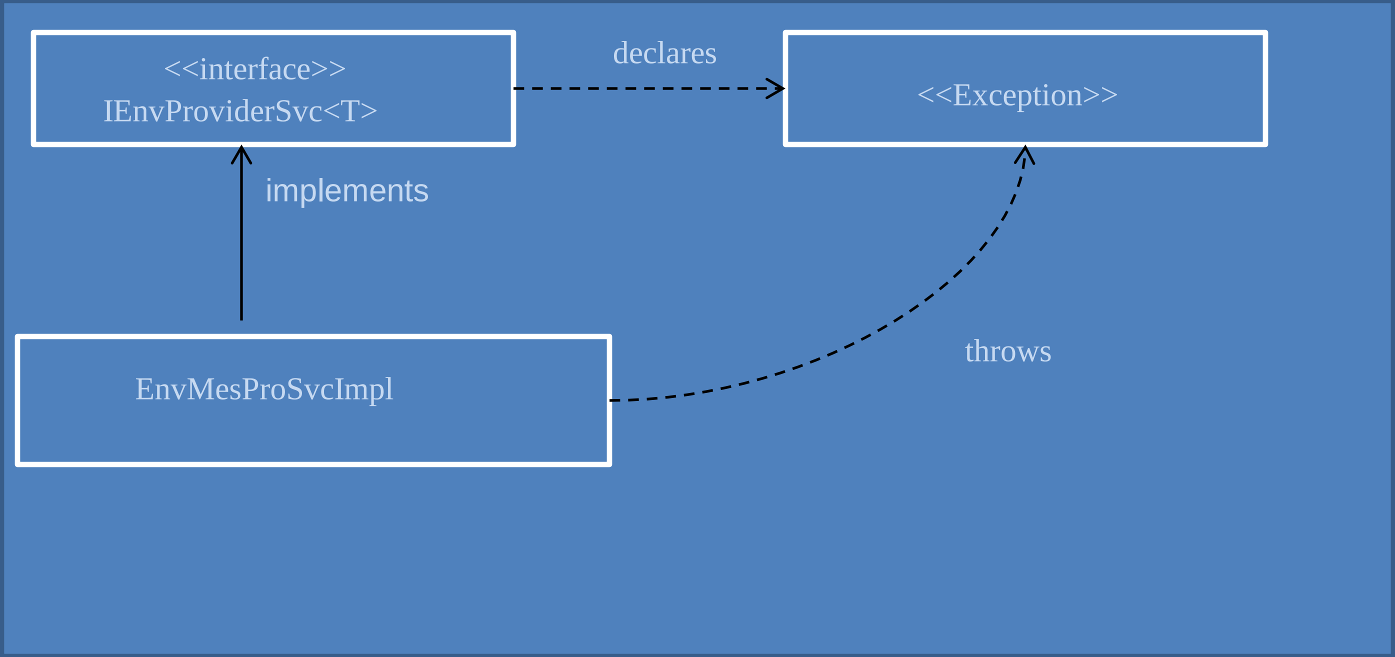
Transformation Service



Package Service

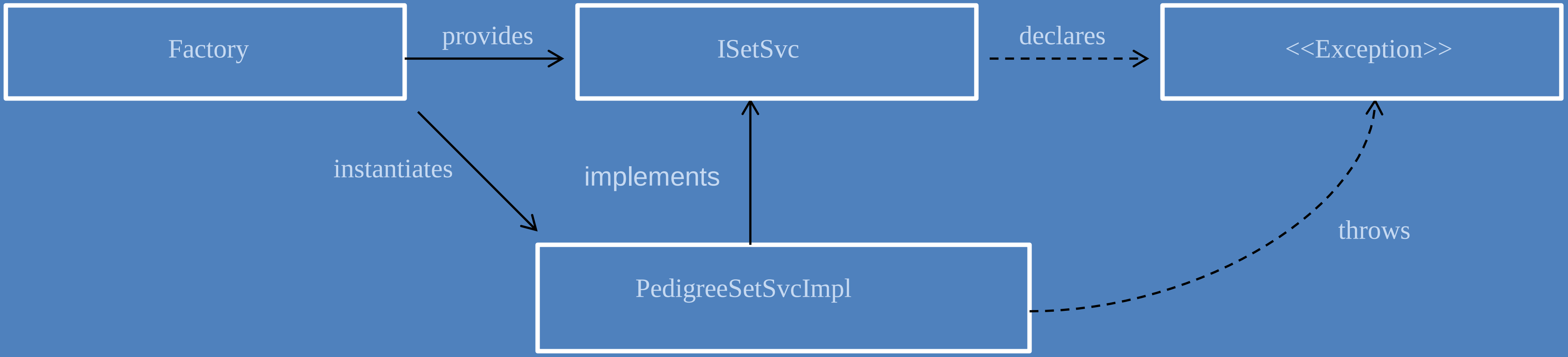


Message Service

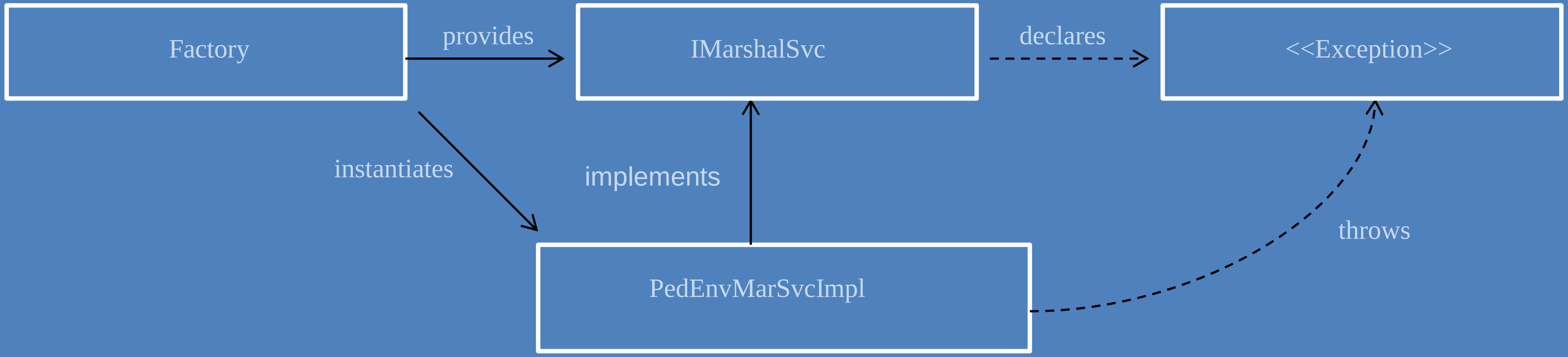


Factory

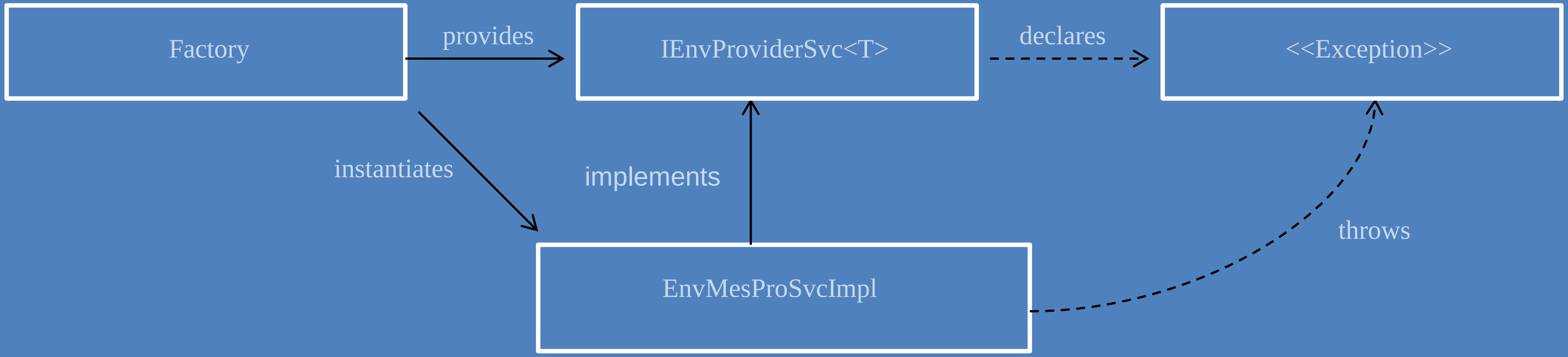
Factory implements Collection Service



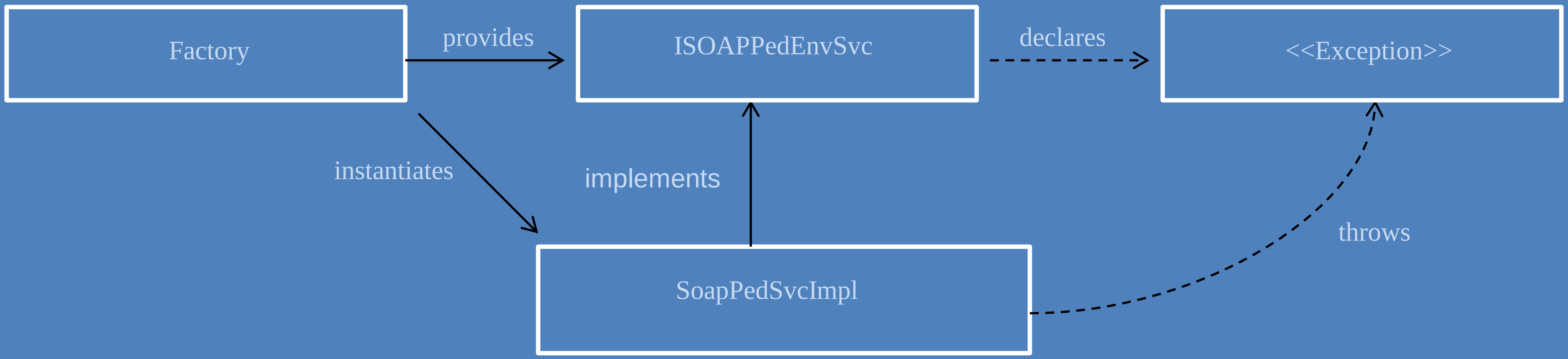
Factory implements Transformation Service



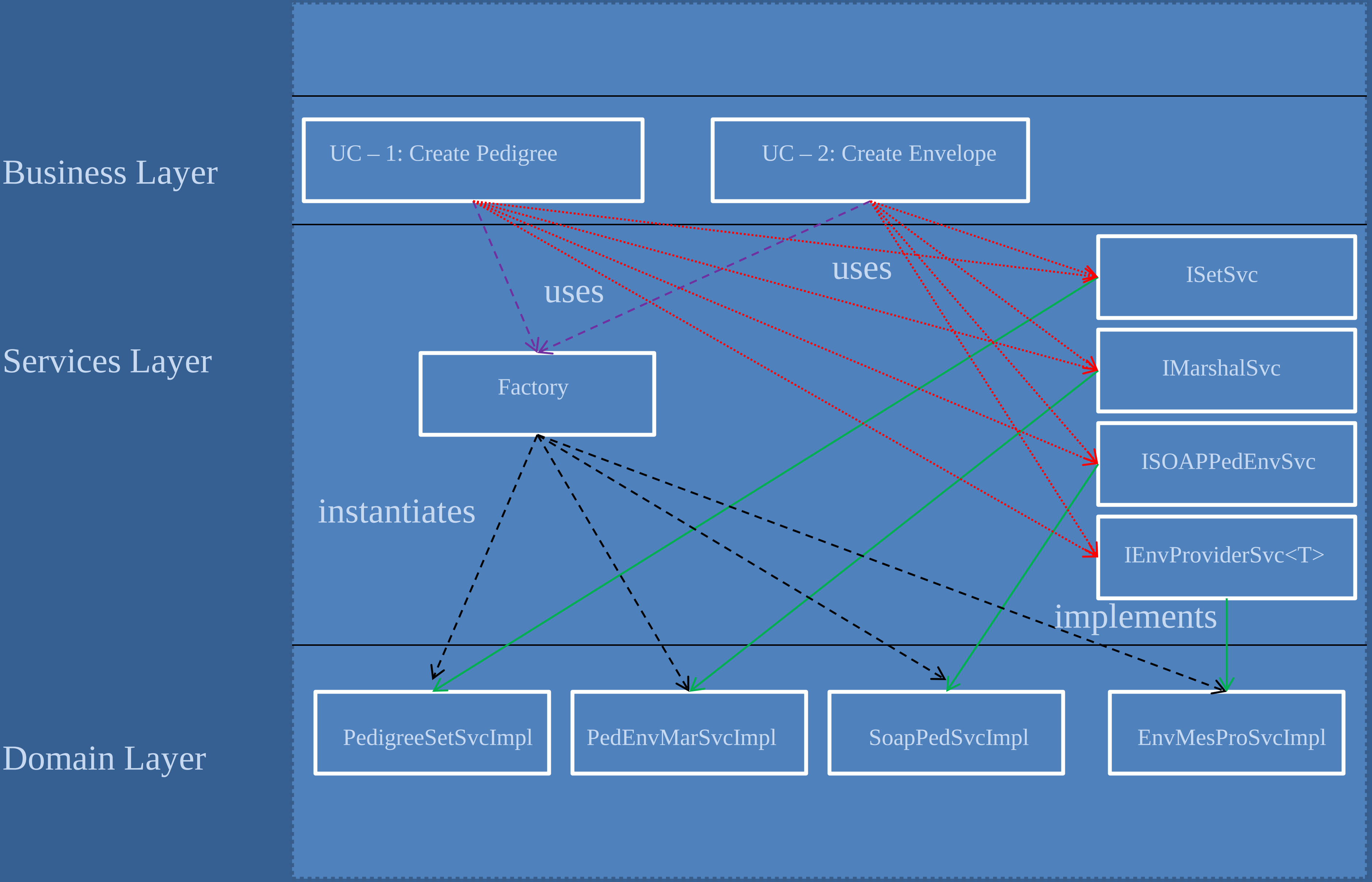
Factory implements Message Service



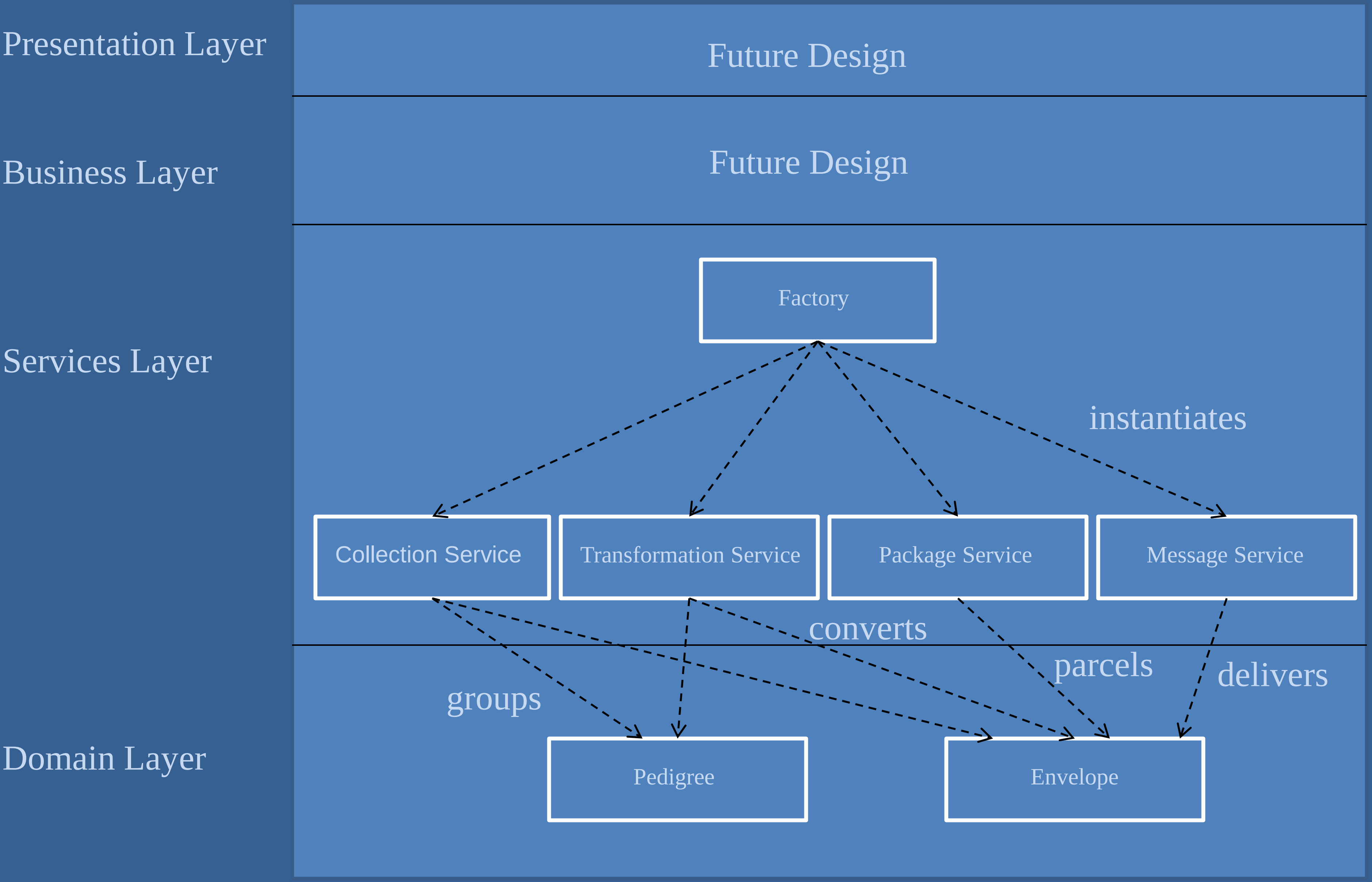
Factory implements Package Service



Use case and Services Integration


Business Layer
Services Layer
Domain Layer
UC – 1: Create Pedigree
UC – 2: Create Envelope
uses
uses
Factory
ISetSvc
IMarshalSvc
ISOAPPedEnvSvc
IEnvProviderSvc<T>
instantiates
implements
PedigreeSetSvcImpl
PedEnvMarSvcImpl
SoapPedSvcImpl
EnvMesProSvcImpl

Services & Domain Integration



Presentation Layer
Future Design
Business Layer
Future Design
Services Layer
Factory
instantiates
Collection Service
Transformation Service
Package Service
Message Service
converts
parcels
delivers
groups
Domain Layer
Pedigree
Envelope