

Jewelly
Bespoke jewelry

LOCARNO SILVIA
MINOTTI LUCA

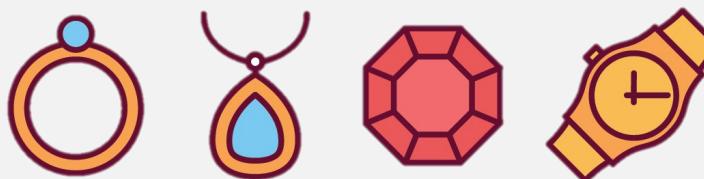
20TH JULY 2021

OUR COMPANY

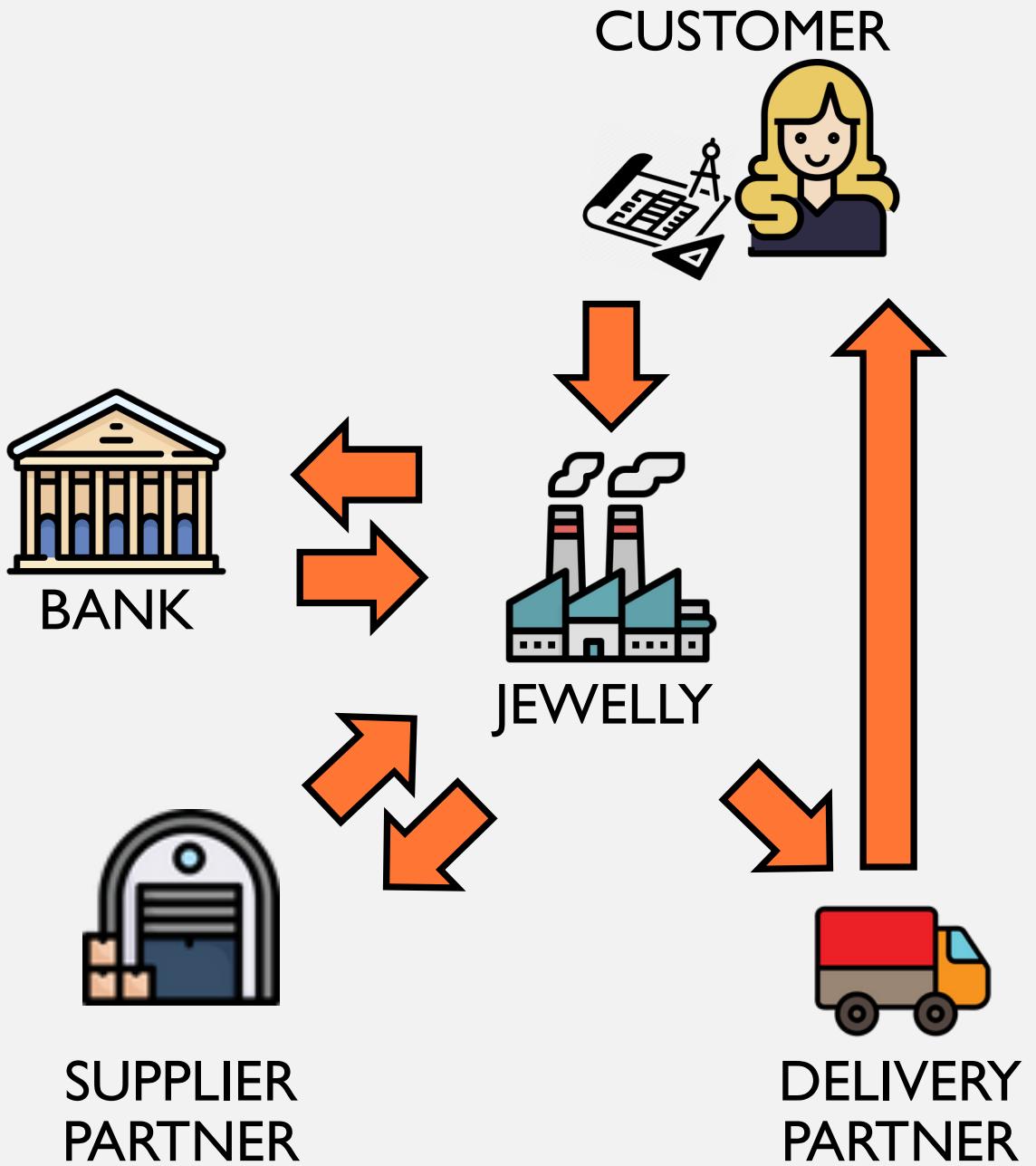
SERVICE DESCRIPTION

Jewelly is a **manufacturing company** that **produces bespoke jewelry**.

Starting from the design of the jewel that the customer wants, our company is able to produce it and deliver it directly to his home.



AN OVERVIEW



WHAT WE HAVE DONE

- Archimate Modeling
- Business Process Choreography
- Business Process Orchestration
- Soundness Checks
- Design and implementation of 3rd-party APIs
- Executable Process

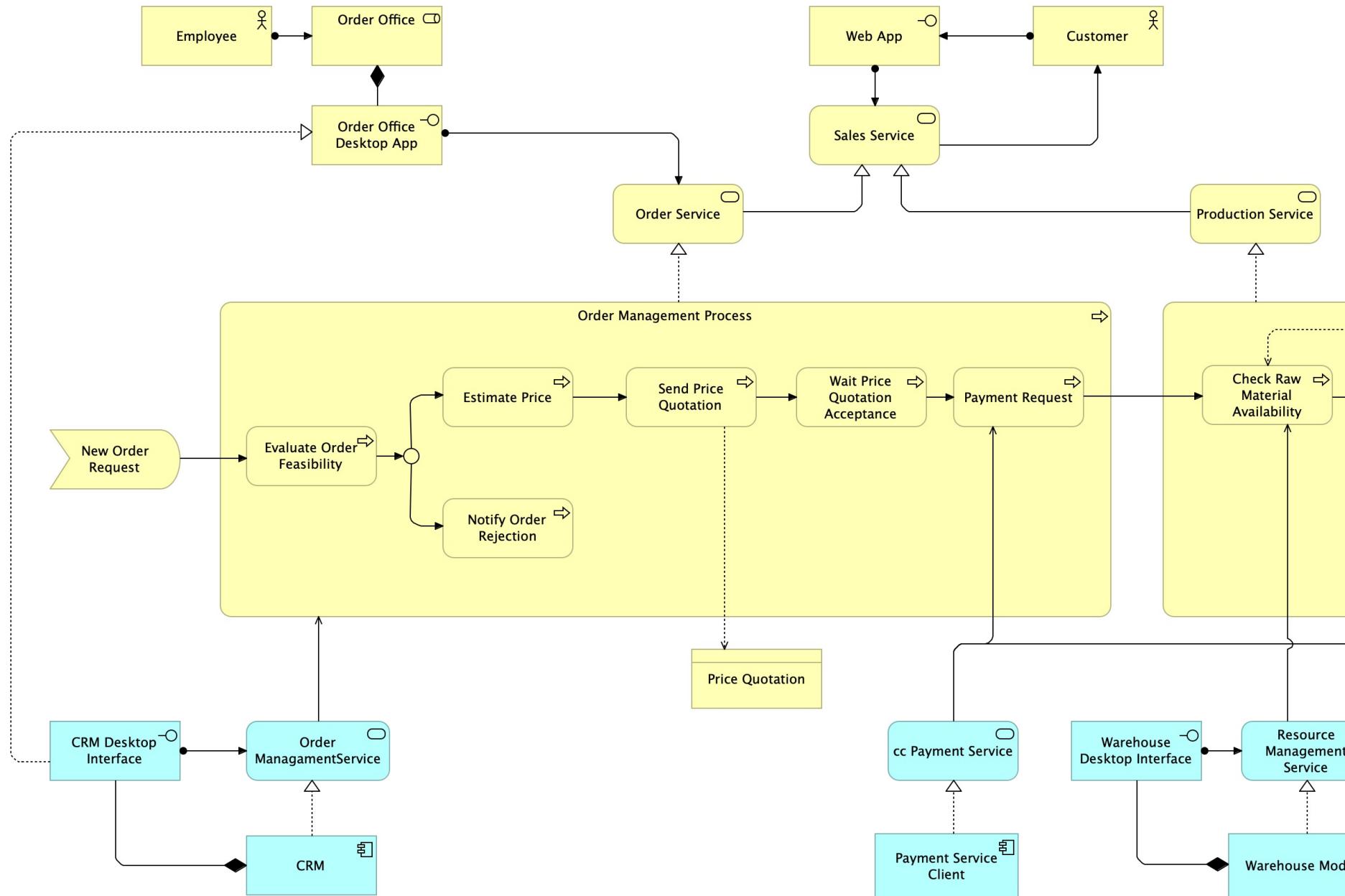
ARCHIMATE MODEL

Archimate is a language for **enterprise architecture modelling** to support the description analysis and visualization of the architecture within and across business domain.

The goal of the Archimate model is to define the **process at high level of granularity**, how the consumer is connected to the enterprise and how the processes are related

ARCHIMATE MODEL

PART I

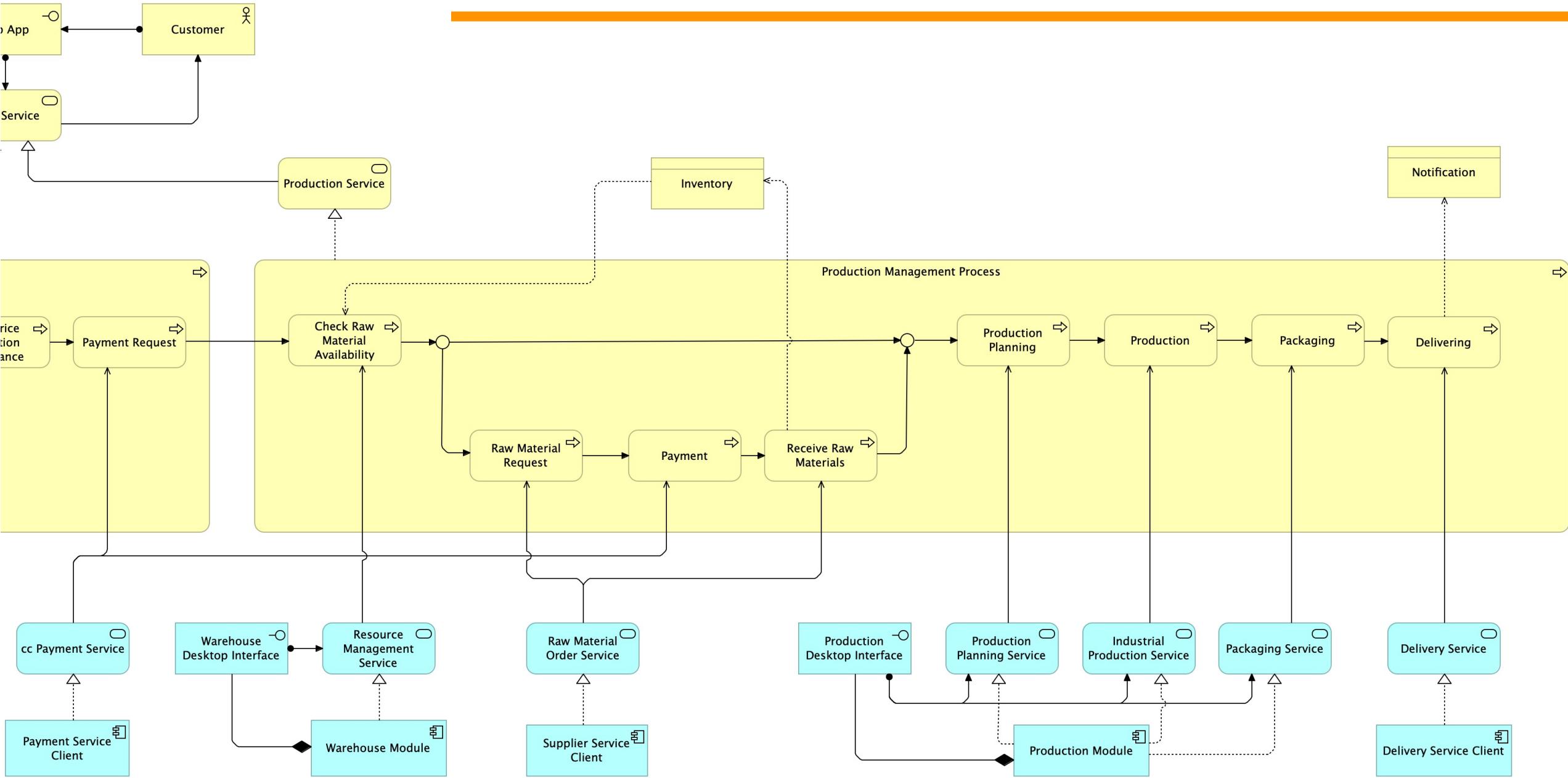


Our model consists of two major processes:

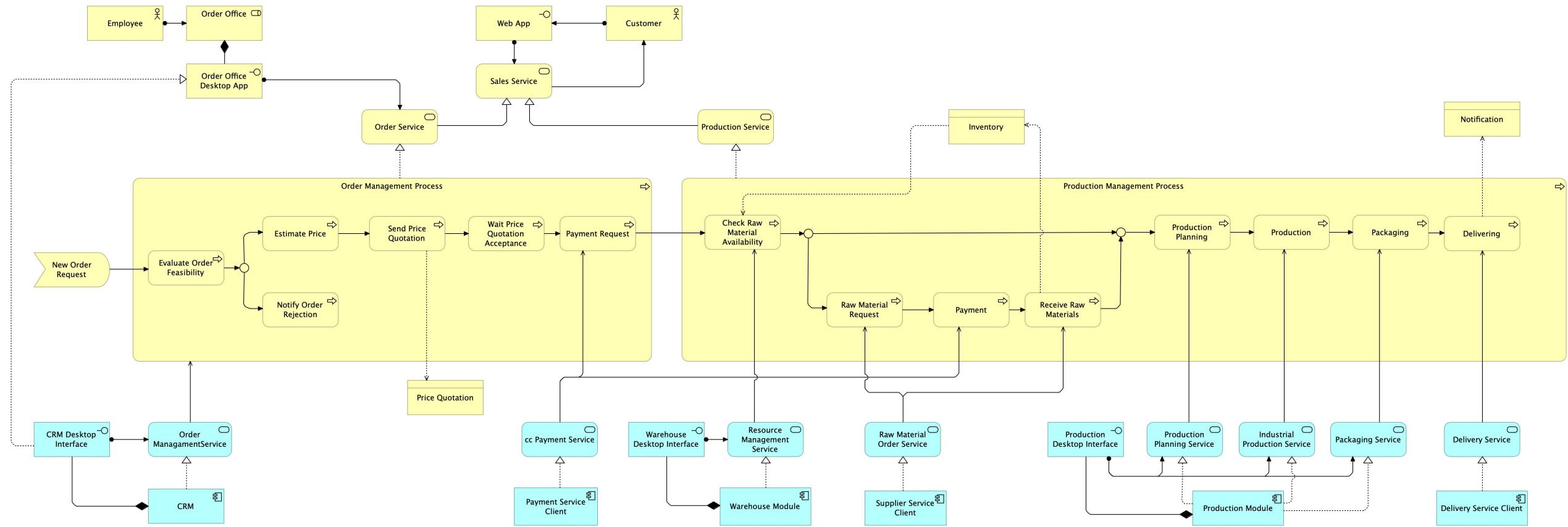
- **Order Management:** realizes the *order service*
 - **Production Management:** realizes the *production service*

The two services contribute to the realization of the ***sales service***, which is the main service of Jewelly.

ARCHIMATE MODEL PART II



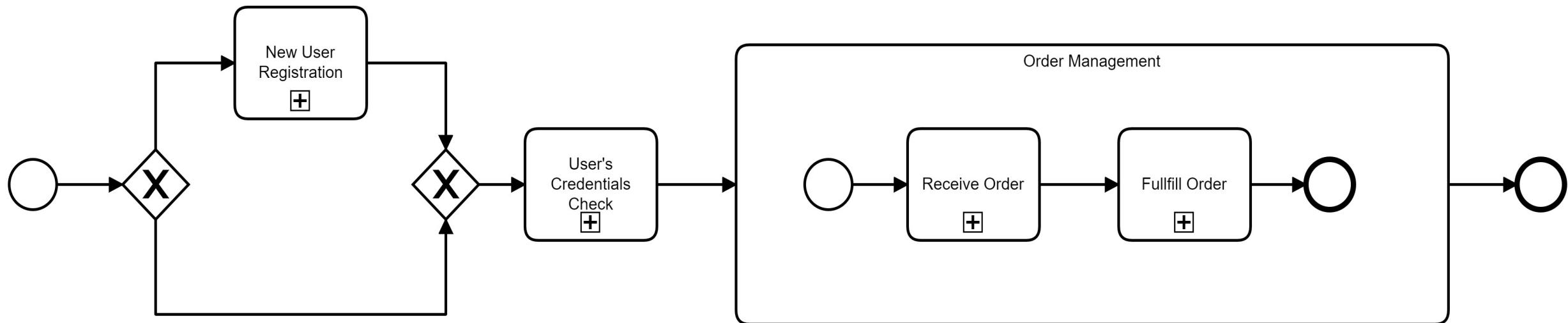
ARCHIMATE MODEL FULL VIEW



HIGH-LEVEL ORCHESTRATION

The **main subprocesses** of our organization:

- User registration
- User login
- Receive order
- Fulfil order



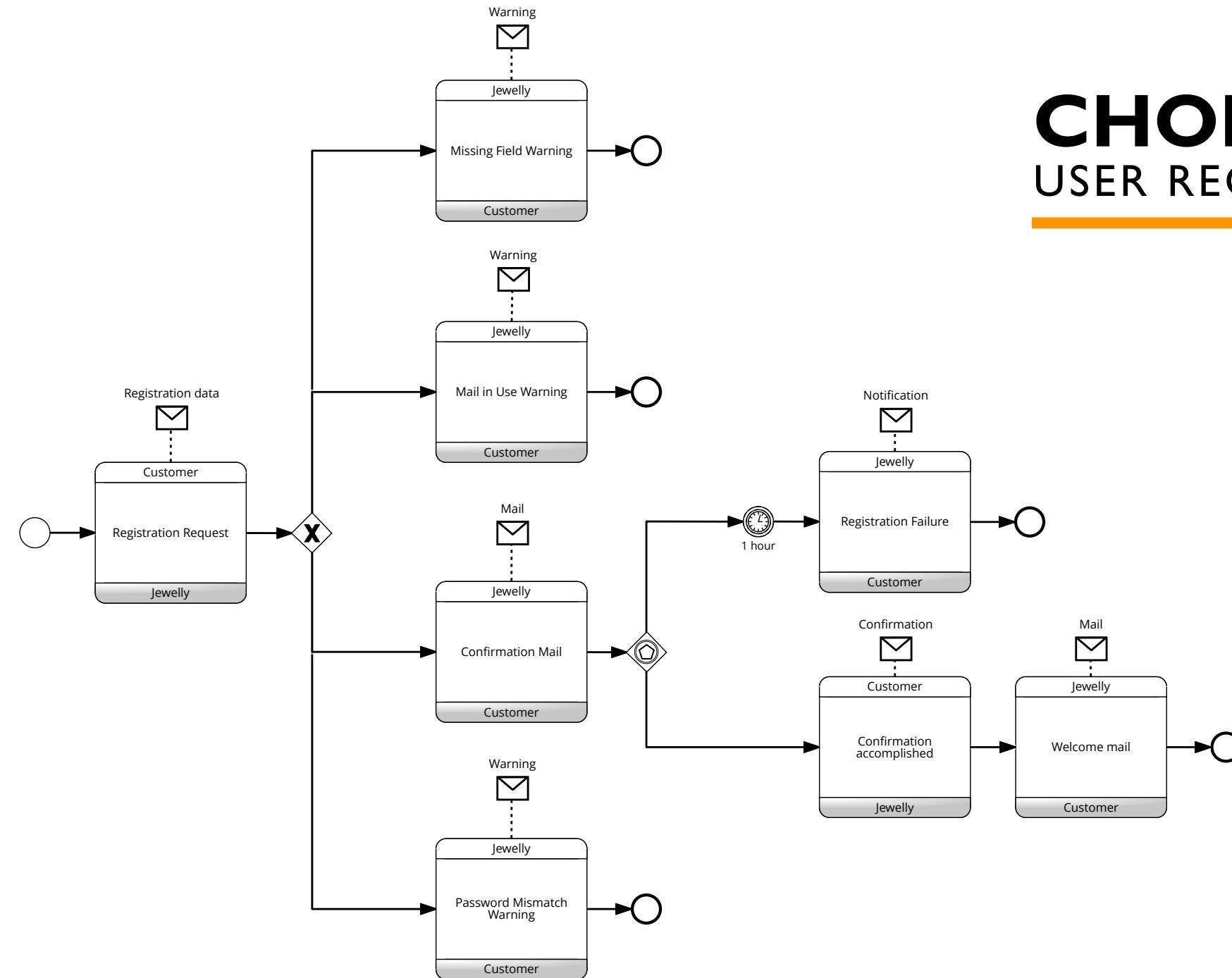
CHOREOGRAPHY

The choreography diagram provides a **high-level perspective** focused **only on the relationships among the organization**, but without specifying the internal processes.

It does not take the standpoint of any of the participants and typically is the base from which the collaboration diagram is derived.

CHOREOGRAPHY

USER REGISTRATION



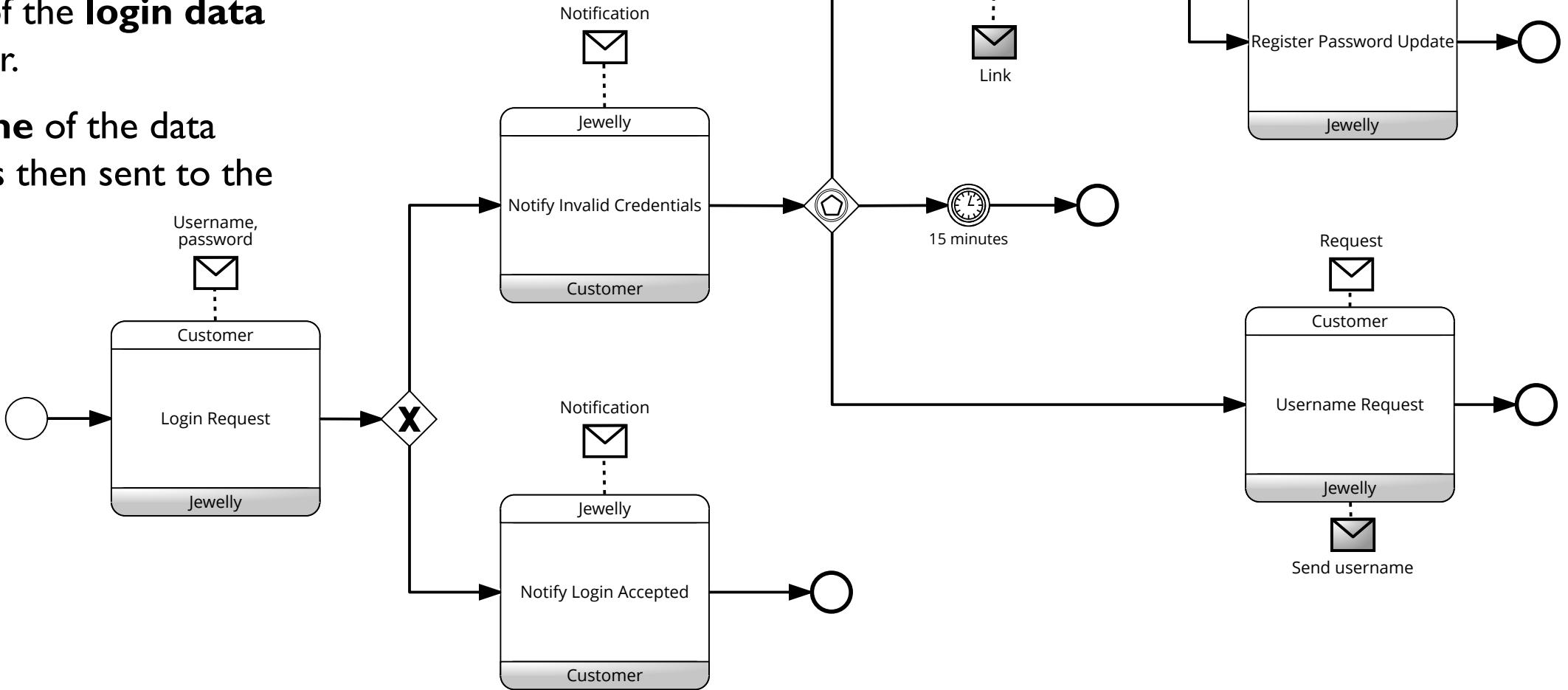
The interaction begins with the receipt of the **registration data** from the customer. If the data are correct then an email to **confirm the registration** is sent to the customer. The confirmation email contains a link that must be clicked by the user within one hour to complete the registration. Following this, Jewelly sends a **registration confirmation** email and a **welcome email**.

CHOREOGRAPHY

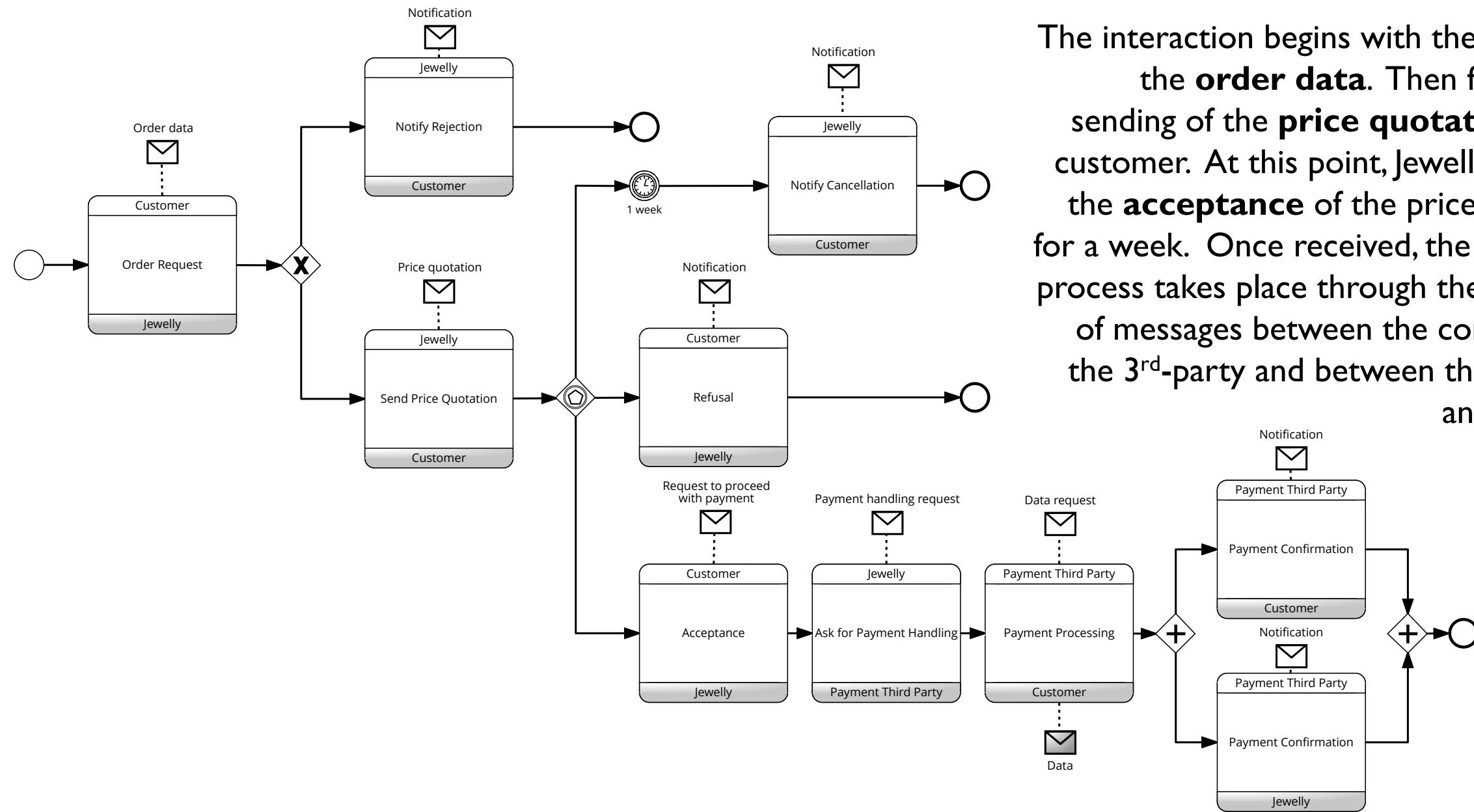
USER LOGIN

The interaction begins with the receipt of the **login data** from the user.

The **outcome** of the data verification is then sent to the customer.



CHOREOGRAPHY RECEIVE ORDER

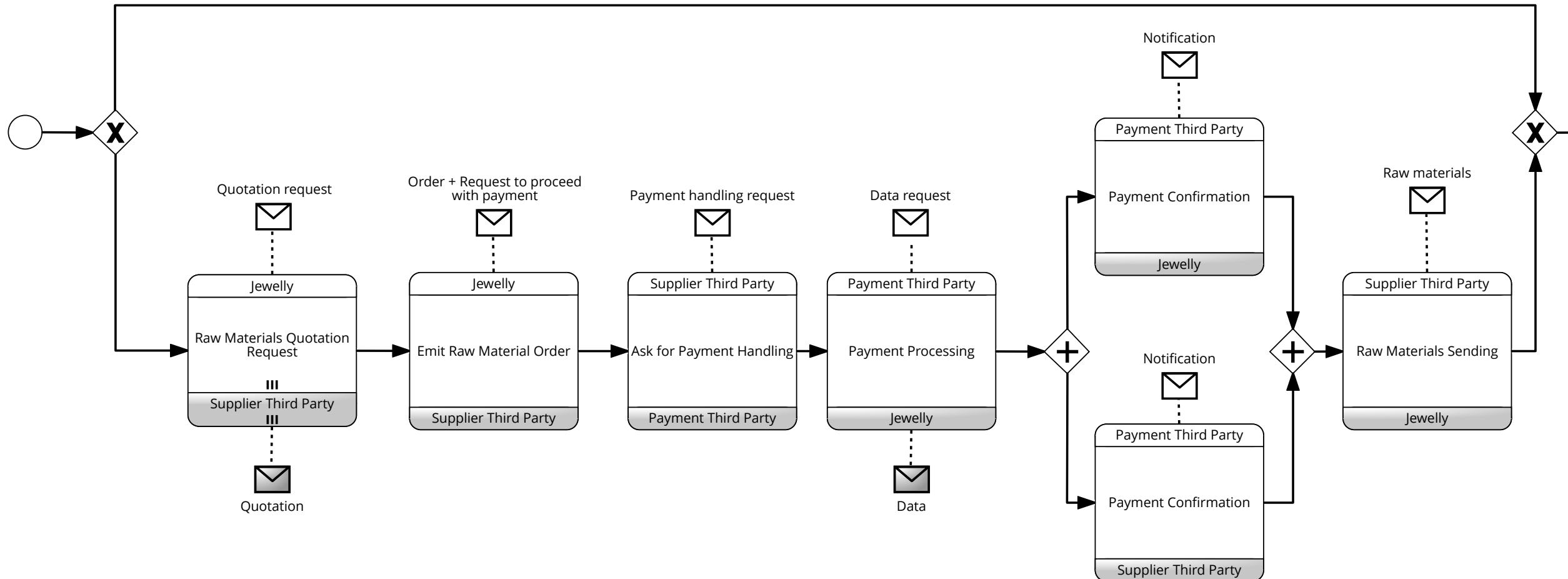


The interaction begins with the receipt of the **order data**. Then follows the sending of the **price quotation** to the customer. At this point, Jewelly waits for the **acceptance** of the price quotation for a week. Once received, the **payment** process takes place through the exchange of messages between the company and the 3rd-party and between the 3rd-party and the user.

CHOREOGRAPHY

FULFILL ORDER PART I

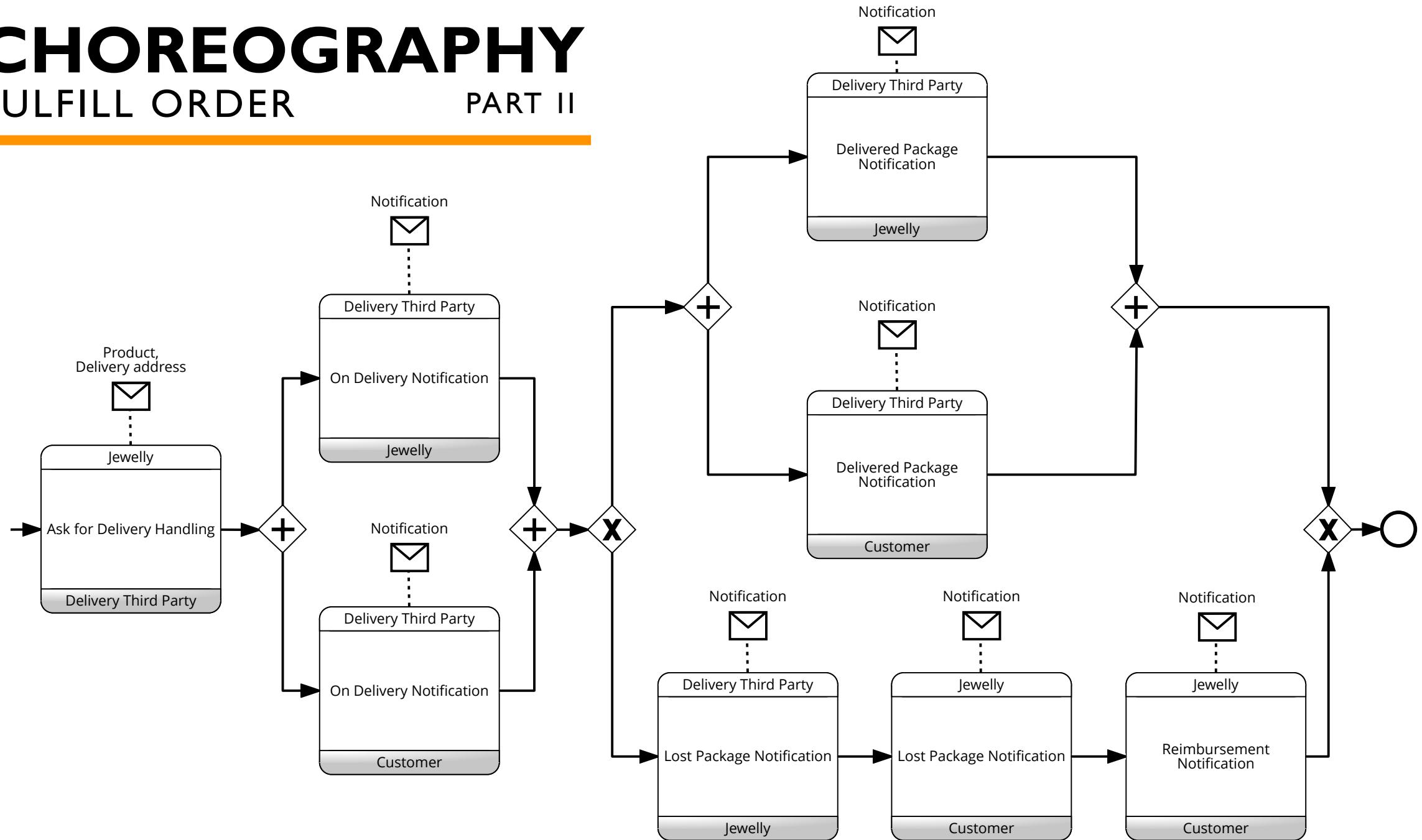
The beginning of the process follows the payment of the order by the customer. If necessary, a **price quotation request** is sent to **multiple suppliers** for the order of materials. After the choice of the most convenient supplier, the **order** is emitted. Then follows the actual **payment**.



CHOREOGRAPHY

FULFILL ORDER

PART II

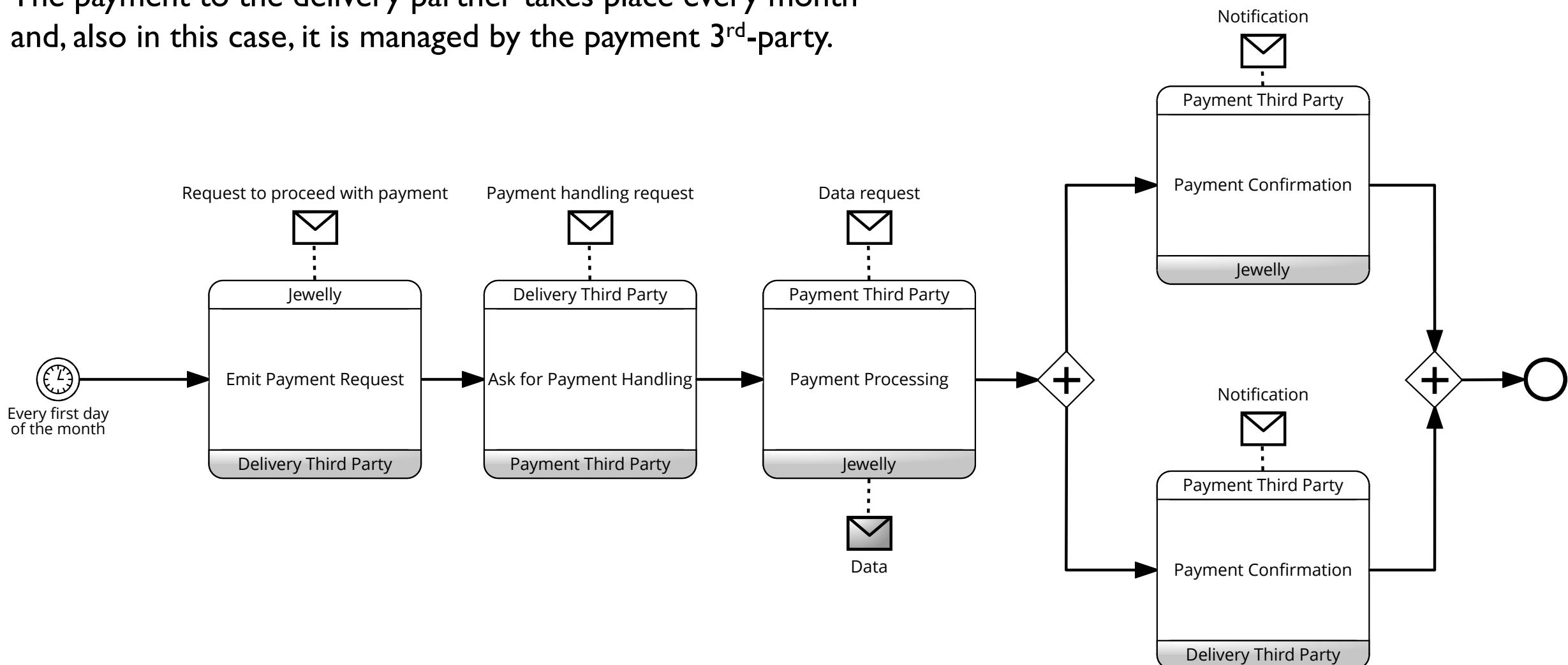


CHOREOGRAPHY

FULFILL ORDER

PART III

The payment to the delivery partner takes place every month and, also in this case, it is managed by the payment 3rd-party.



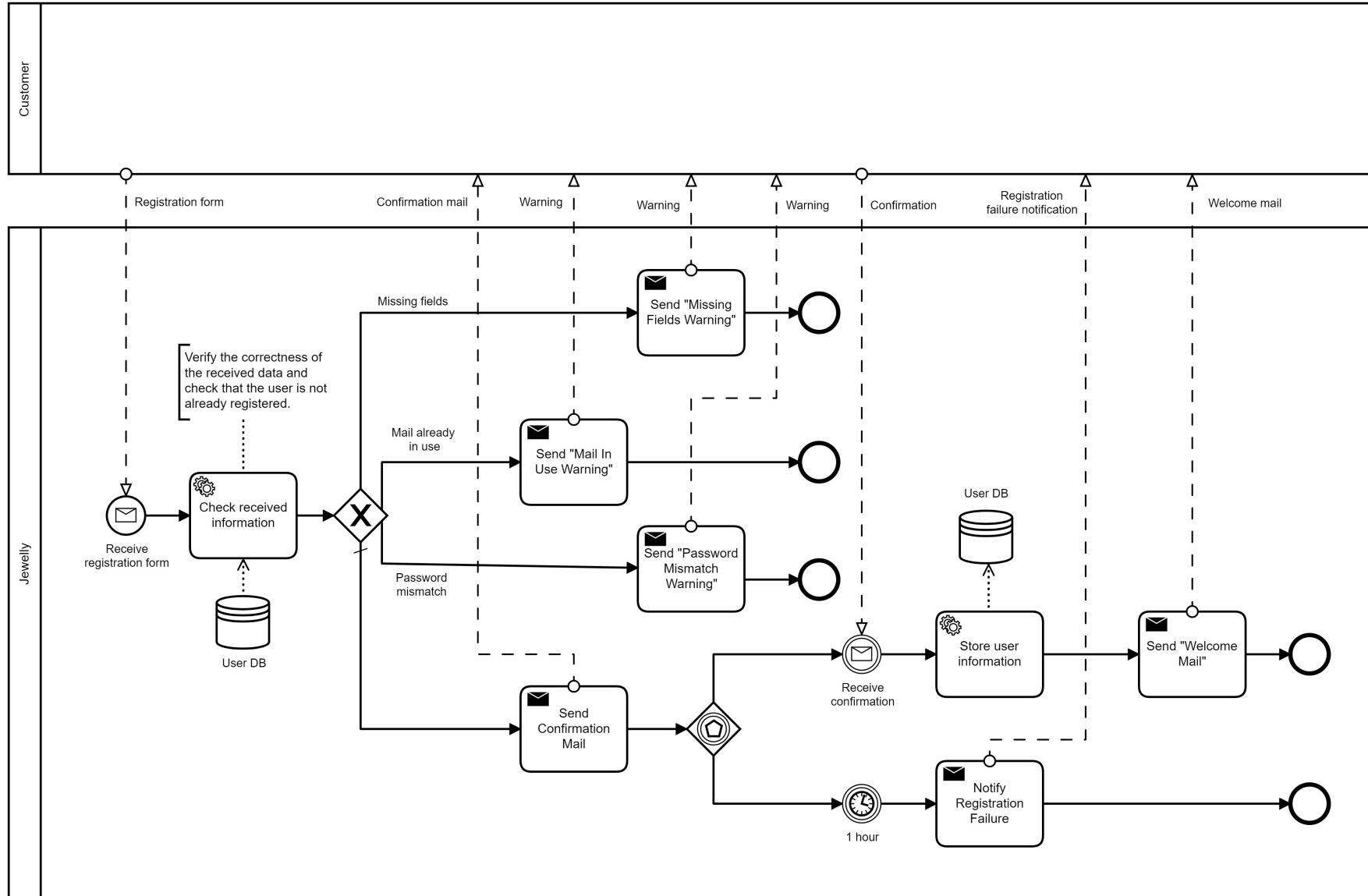
ORCHESTRATION

Orchestration concerns the management of **activities executed under a common controlling element** (the orchestrator). Usually, the boundary of an orchestrator is an organization.

The orchestrator controls the execution of the business process according to a **control-flow model**. The control-flow is usually modelled based on a set of patterns.

ORCHESTRATION

USER REGISTRATION

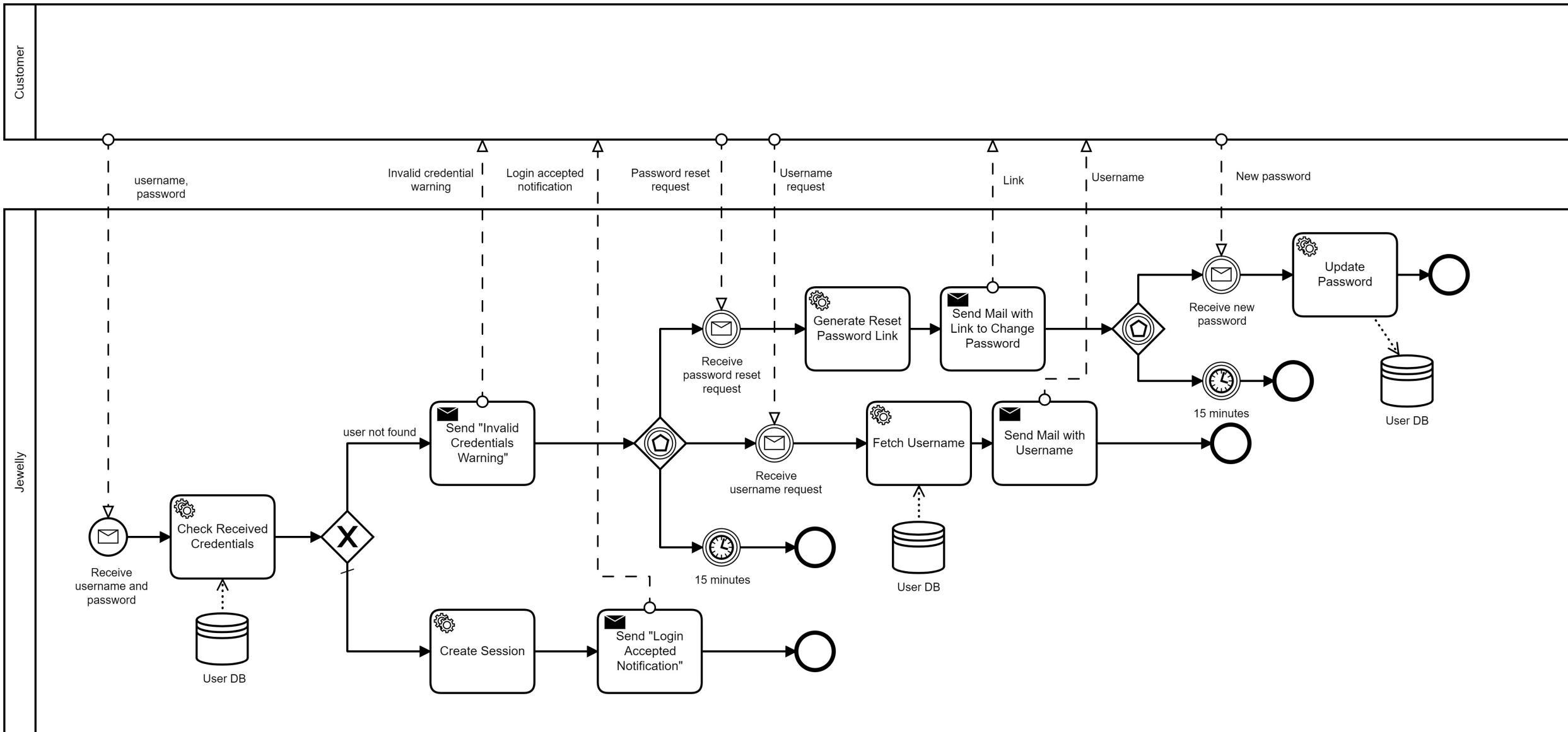


During the registration we control of the **correctness** of the data.

The **database** is used both to verify the customer data and to memorize them when the registration is confirmed.

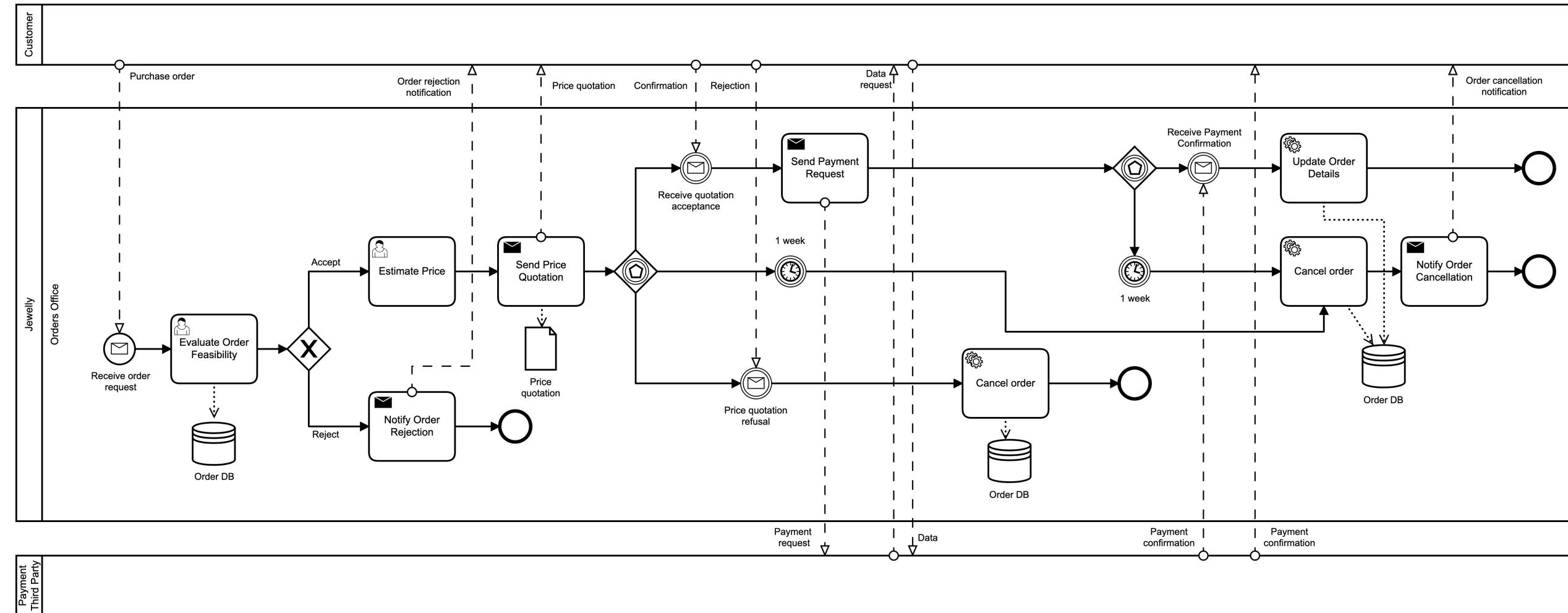
ORCHESTRATION

USER LOGIN



ORCHESTRATION

RECEIVE ORDER



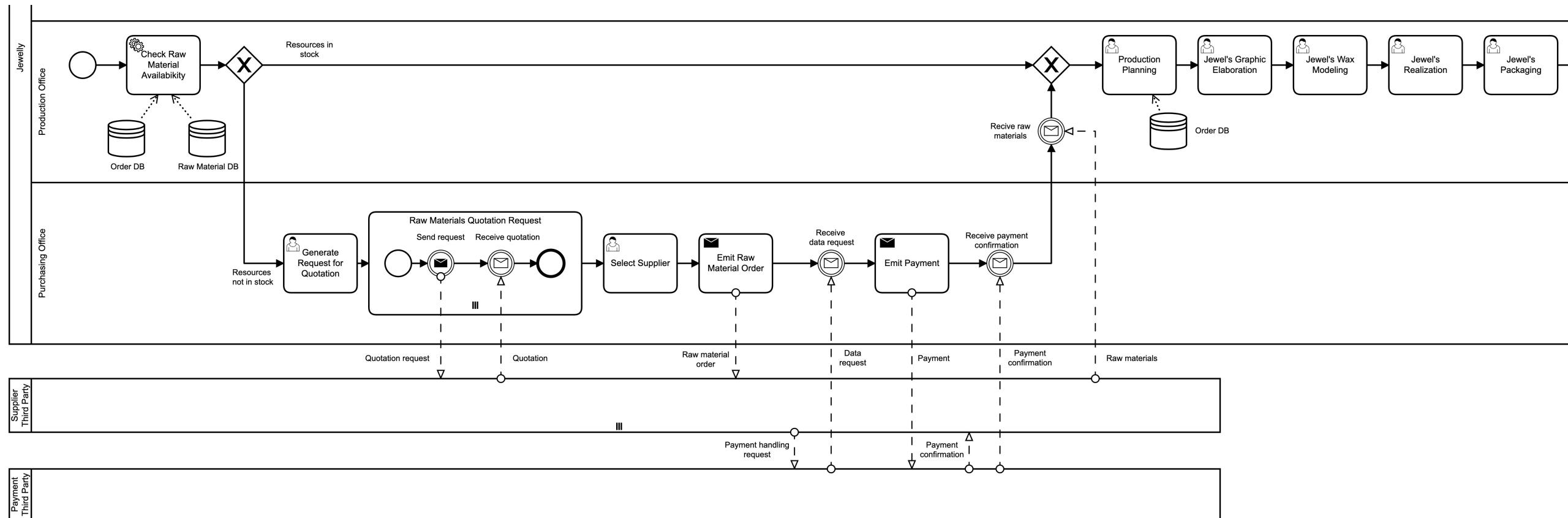
ORCHESTRATION

FULFILL ORDER PART I

A service task, verify the **availability of raw materials**.

If the resources are not in stock, the company has to **purchase materials** before proceeding with the production.

Once the company has received the materials, or if the resources are already in stock, a set of user tasks, concerning the **production**, are executed.



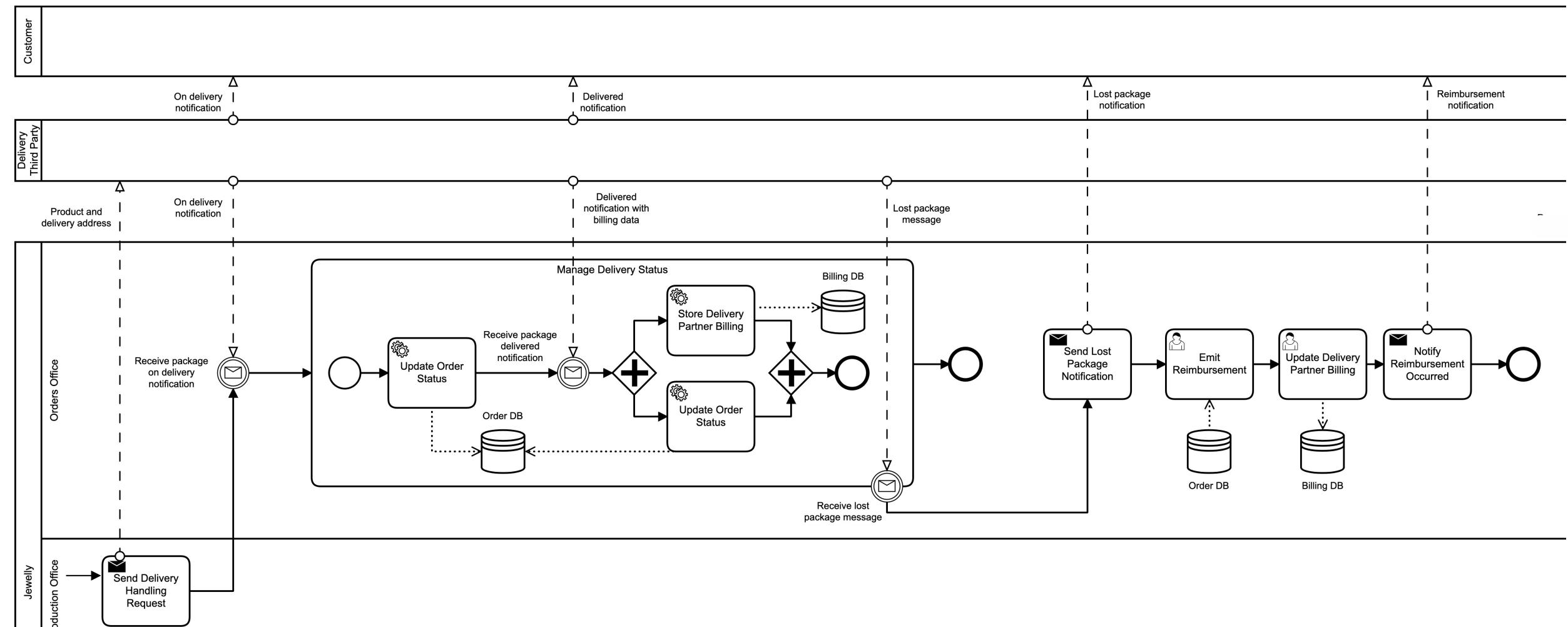
ORCHESTRATION

FULFILL ORDER PART II

After the production, the company sends a **request for delivery** to a third-party service.

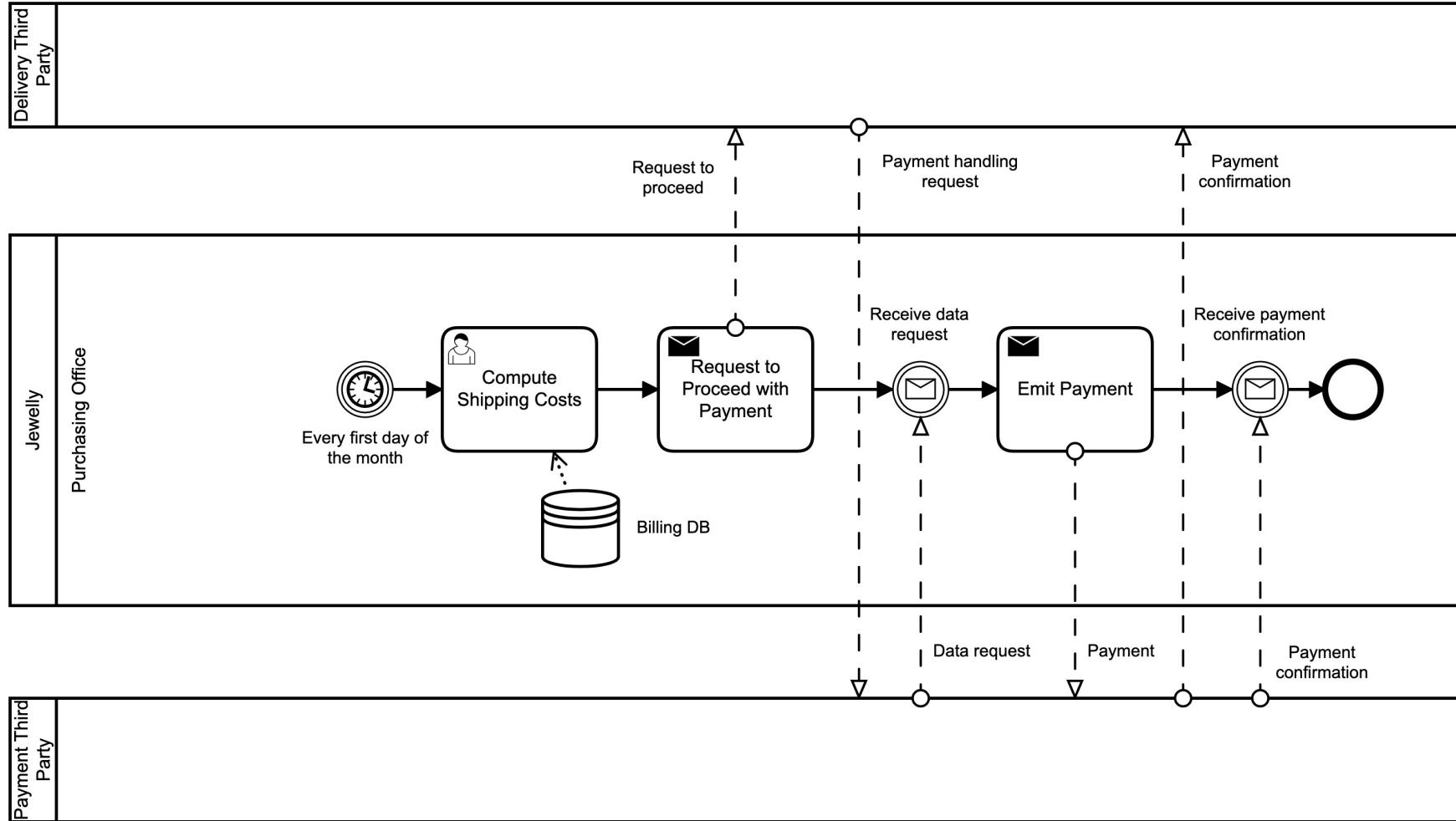
After the package has been delivered, in addition to the update of the status, the **delivery partner billing**, that will be useful when the delivery partner will be paid, is saved on a dedicated database.

During the delivery process, an **exception** can generates: the loss of the package. In that case, the user is reimbursed.



ORCHESTRATION

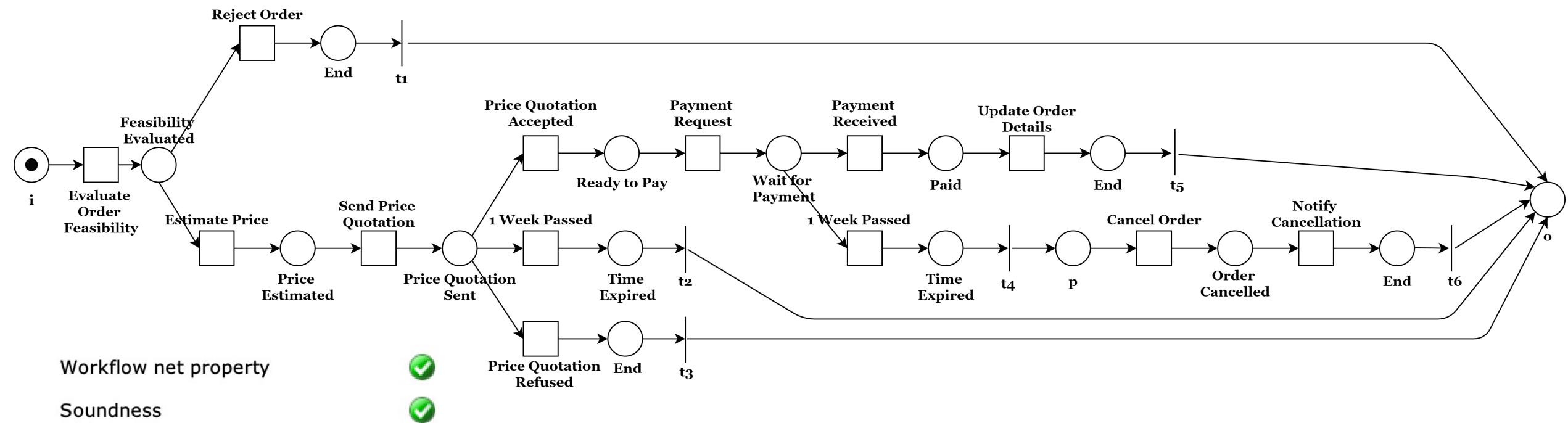
FULFILL ORDER PART III



Every **first day of the month** our company pays the delivery partner. The **shipping cost** is computed through the data saved in the billing database

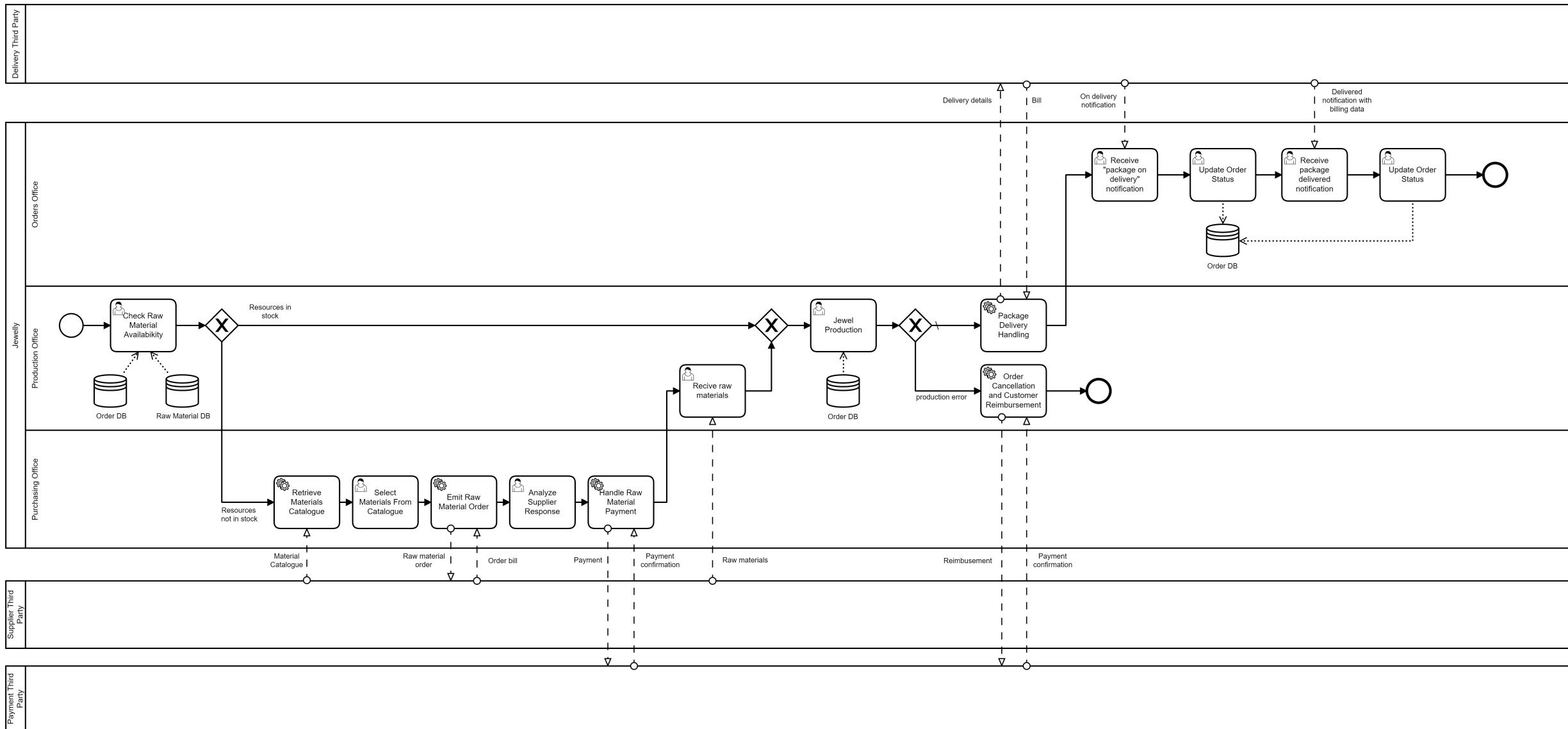
SOUNDNESS RECEIVE ORDER

The **soundness** is a property that allow to identify correct WF-nets and so business processes. A WF-Net (corresponding to a BP) is sound if and only if: for any case, the procedure will **terminate** eventually, and at the moment the procedure terminates there is a **token** in place ‘o’ (the final place) and all the other places are **empty**.



IMPLEMENTATION

SIMPLIFIED FULFILL ORDER



REST API

deliverly APIs of the Jewelly's delivery third party for requesting and managing deliveries



POST /deliverly/delivery Make a shipping request to Deliverly

GET /deliverly/delivery/{id} Get details of an existing shipping request

supplierly APIs of the Jewelly's supplier third party for requesting and managing raw material orders



GET /supplierly/catalogue Get the catalogue of the available raw materials

POST /supplierly/order Place a raw material order to Supplierly

bankly APIs of the Jewelly's payment third party for requesting and managing payments



POST /bankly/payment Emit a payment

We have three external services:

- **Deliverly**: for requesting and managing deliveries.
- **Supplierly**: for requesting and managing raw materials orders.
- **Bankly**: for requesting and managing payments.

REST API SCHEMAS

Schemas

[DeliveryRequest >](#)

[Delivery >](#)

[Catalogue >](#)

[OrderRequest >](#)

[Order >](#)

The majority of the exchanged messages are structured according to some **schemas** that we have defined.

[PaymentRequest >](#)

[Catalogue_materials >](#)

[OrderRequest_materials >](#)

POST

/deliverly/delivery Make a shipping request to Deliverly

This API allows to submit a request for a new delivery

Parameters

No parameters

Request body required

Details of the delivery to be placed

Examples:

```
{  
  "address": "Via Pola, 12 - Milano (MI) 22060",  
  "recipient": "Mario Rossi"  
}
```

[Example Value](#) | [Schema](#)

Responses

Code

200

Description

The shipment has been taken over

Media type

application/json

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "address": "Via Pola, 12 - Milano (MI) 22060",  
  "cost": 6,  
  "recipient": "Mario Rossi",  
  "id": 0,  
  "status": "processing"  
}
```

400

Bad request, the address is invalid or not served by Deliverly

REST API

DELIVERY PARTNER

GET**/deliverly/delivery/{id}** Get details of an existing shipping request

This API allows to get details about a delivery

REST API

DELIVERY PARTNER

Parameters

Name

Description

id * required**integer****(path)**

Delivery identifier

id - Delivery identifier

Responses

Code

200

Description

Delivery details

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
    "address": "Via Pola, 12 – Milano (MI) 22060",  
    "cost": 6,  
    "recipient": "Mario Rossi",  
    "id": 0,  
    "status": "processing"  
}
```

404

Delivery not found

GET**/supplierly/catalogue** Get the catalogue of the available raw materials

This API allows to get the catalogue of the available raw materials

REST API

SUPPLY PARTNER

Parameters

No parameters

Responses

Code	Description
------	-------------

200	Catalogue
-----	-----------

Media type

application/json

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "materials": [  
    {  
      "id": 1,  
      "name": "Gold thread",  
      "price": 100.5,  
      "availability": 2  
    }  
  ]  
}
```

POST**/supplierly/order** Place a raw material order to Supplierly

This API allows to submit a raw material order

Parameters

No parameters

Request body required

Details of the order to be placed

Examples: [Example Value](#) | [Schema](#)

```
{  
  "materials": [  
    {  
      "id": 1,  
      "quantity": 4  
    },  
    {  
      "id": 2,  
      "quantity": 5  
    }  
  ]  
}
```

Responses**Code** **Description**

200

Order details

Media type

application/json [Example Value](#) | [Schema](#)

```
{  
  "deliveryDate": "10/12/2022",  
  "cost": 1050.5,  
  "iban": "IT60-X054-2811-1010-0000-0123-456",  
  "materials": [  
    {  
      "id": 1,  
      "quantity": 4  
    },  
    {  
      "id": 2,  
      "quantity": 5  
    }  
  ]  
}
```

400

Bad request, required materials do not exist or invalid order

REST API**SUPPLY PARTNER**

POST

/bankly/payment Emit a payment

This API allows to emit a payment

Parameters

No parameters

Request body required

Details of the payment to be placed

Examples:

[Example Value](#) | [Schema](#)

```
{  
  "iban": "IT60-X054-2811-1010-0000-0123-456",  
  "amount": 1050.5  
}
```

Responses

Code	Description
200	Payment successfully processed
400	Insufficient credit to execute the transaction
404	Invalid IBAN

REST API

PAYMENT PARTNER

Jewelly

20TH JULY 2021