

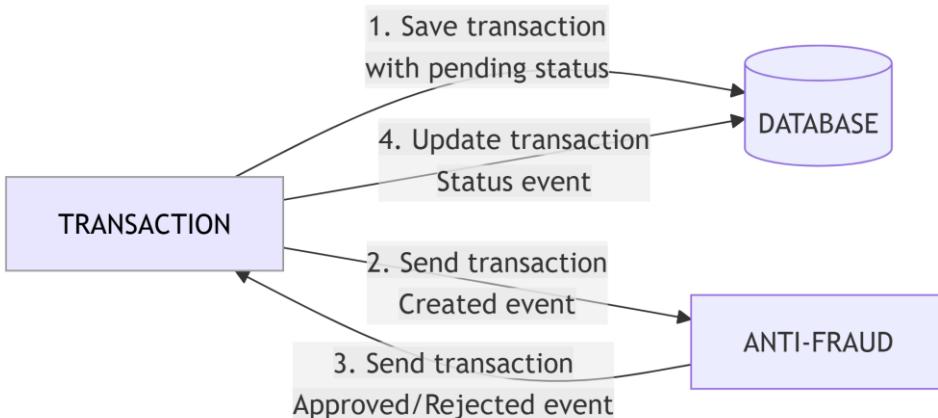
## Challenge

Each time a financial transaction is created, it is processed by the anti-fraud microservice for validation. Once the verification is completed, the same service sends a response to update the transaction's status. At this stage, the system supports three possible statuses.

1. pending
2. approved
3. rejected

The transactions will be rejected under the following conditions:

- When the individual transaction amount exceeds 2500.
- When the total accumulated amount in a single day exceeds 20500.



## Tech Stack

1. .Net 8
2. Any database
3. Kafka

A **Dockerfile** is provided for initializing the development environment.

You must have two resources:

1.Resource to create a transaction that must contain:

```
{  
  "sourceAccountId": "Guid",  
  "targetAccountId": "Guid",  
  "transferTypeId": 1,  
  "value": 120  
}
```

2. Resource to retrieve a transaction

```
{  
  "transactionExternalId": "Guid",  
  "createdAt": "Date"  
}
```

## Evaluation Criteria

- Microservice Design & Approach, effective use of microservice principles, decoupling, and communication patterns (e.g., event-driven, messaging, solid principles). You can follow the modularization, and style that you feel is the most appropriate solution.
- Clean Architecture: proper separation of concerns, dependency injection, modularity, and maintainability,
- Testing: presence and quality of unit tests(xunit).
- Security: adherence to OWASP Top 10 best practices (e.g., input validation, authentication, authorization, data exposure).
- Code Quality: clarity, consistency, and documentation.

## Send us your challenge

You have **5 days** to complete the challenge. Once you're done, please send us your solution, sharing a **GitHub public repository** is preferred.

If you have any questions or need any clarification, feel free to reach out.