

New Bank



**Sourour GAZZEH
Imene YAHIAOUI
Nadim BEN AISSA
Badr AL ACHKAR**

Architectures Logicielles : EVOLUTION

Besoins fonctionnels



Création d'un système NewBank :

- Gestion des comptes (personnels et commerciaux) et cartes (crédit/débit)
- Virement entre comptes, Consultation solde et historique transactions

Implémentation d'un kit de développement logiciel (SDK) :

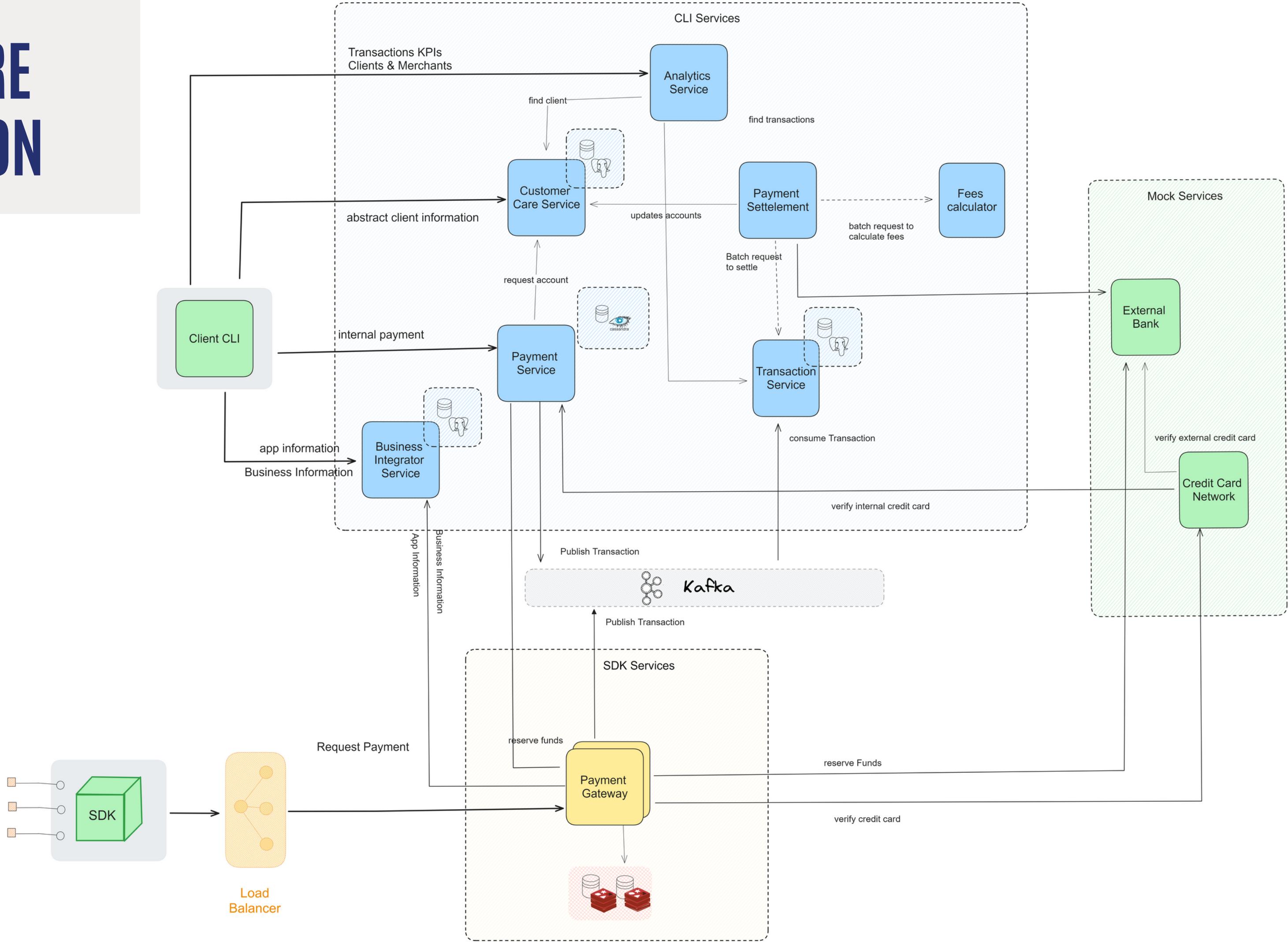
- Gérer les paiements sur les sites web des commerçants.
- Gérer les frais de transaction en ligne

Extension



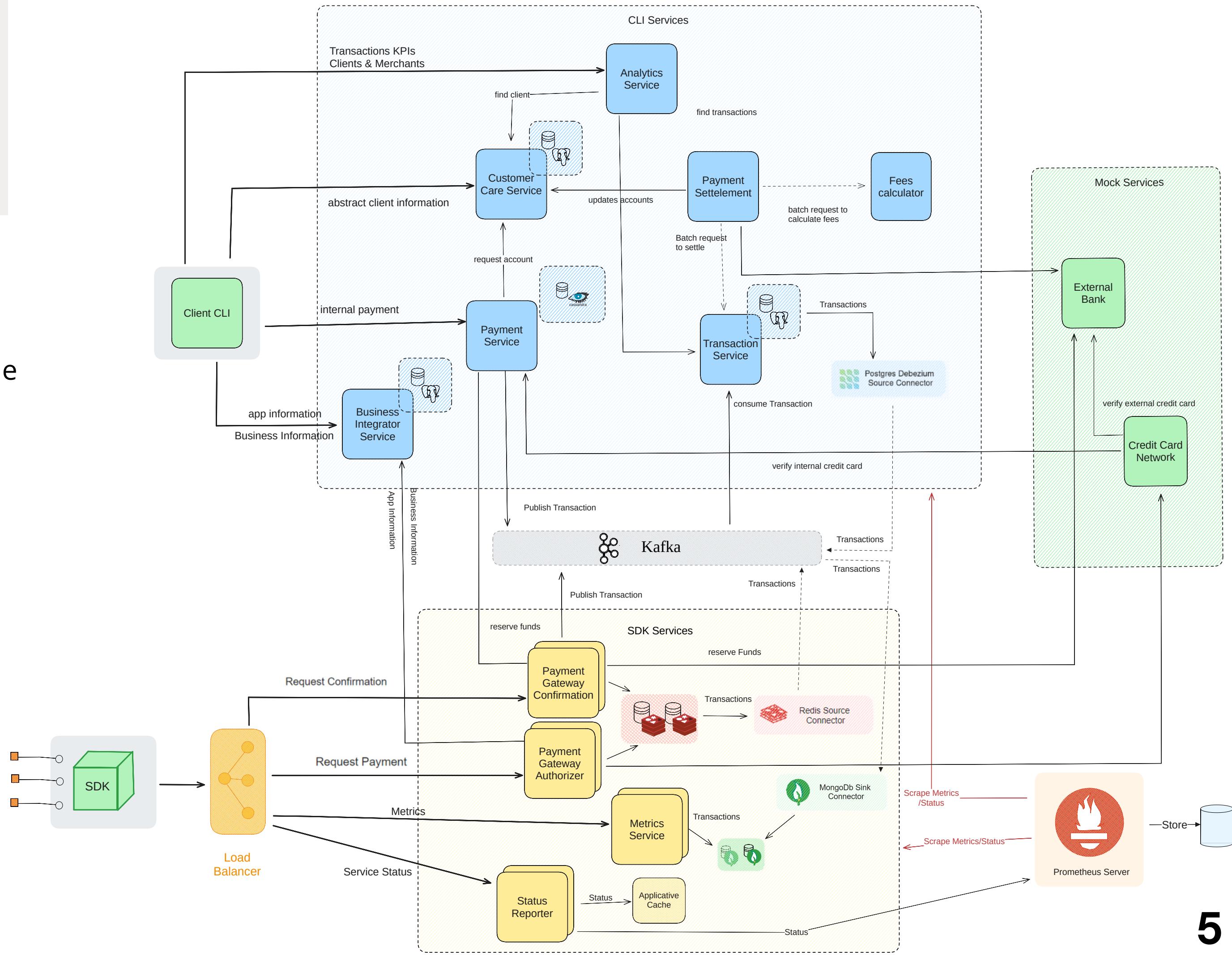
- Statut des services pour les clients
- Dashboard des métriques métier pour les clients
- Résilience (élimination du SPOF)

ARCHITECTURE CONSTRUCTION



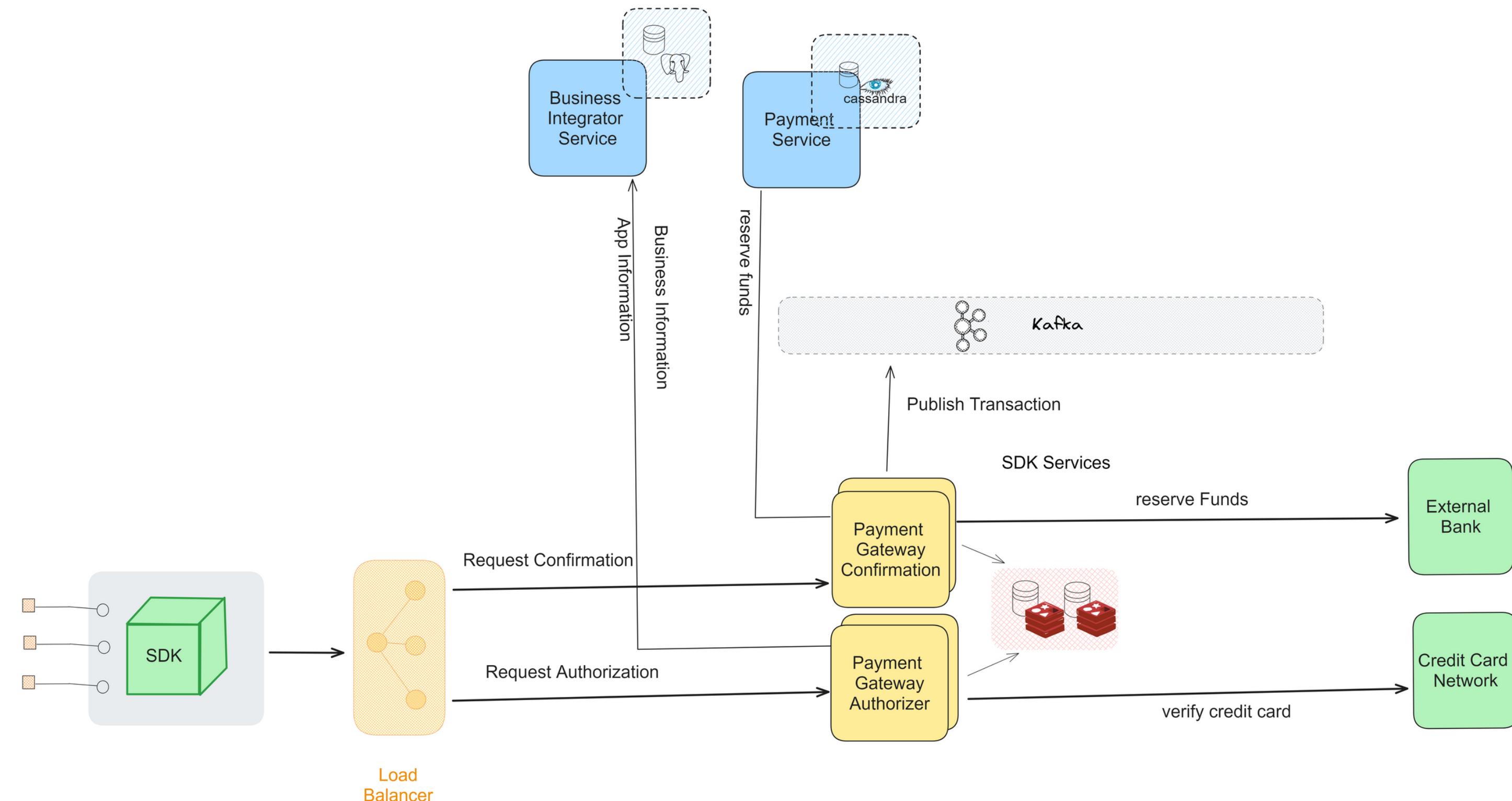
ARCHITECTURE EVOLUTION

- 10 ADRS qui ont guidé l'évolution
 - 3 grandes principales
 - Autres concernant le protocole de communication du SDK avec le reste des services
 - Concentration sur l'évolution des services SDK et évolution interne du SDK



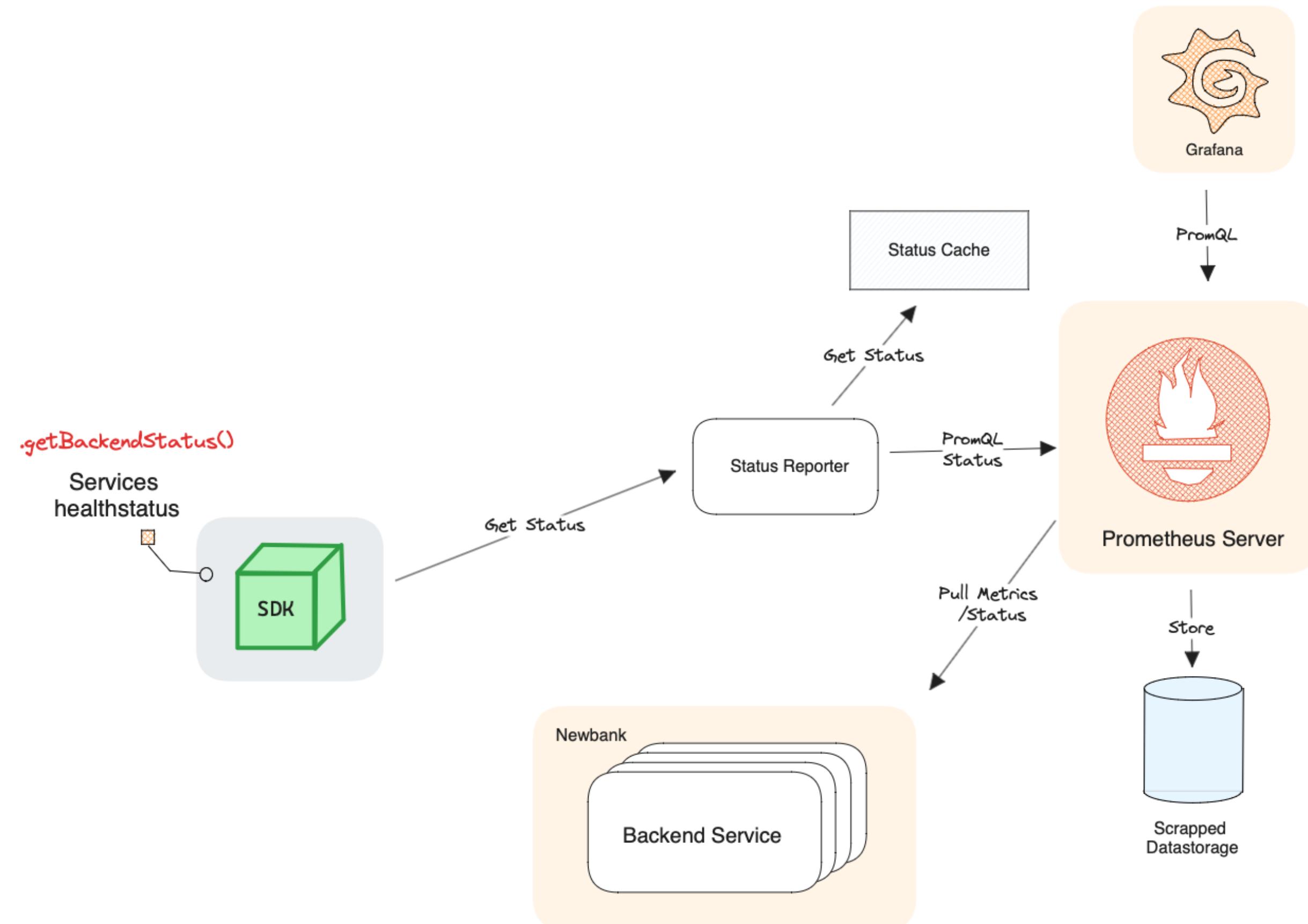
adr001 : payment-gateway-spoof-mitigation

Explication des évolutions



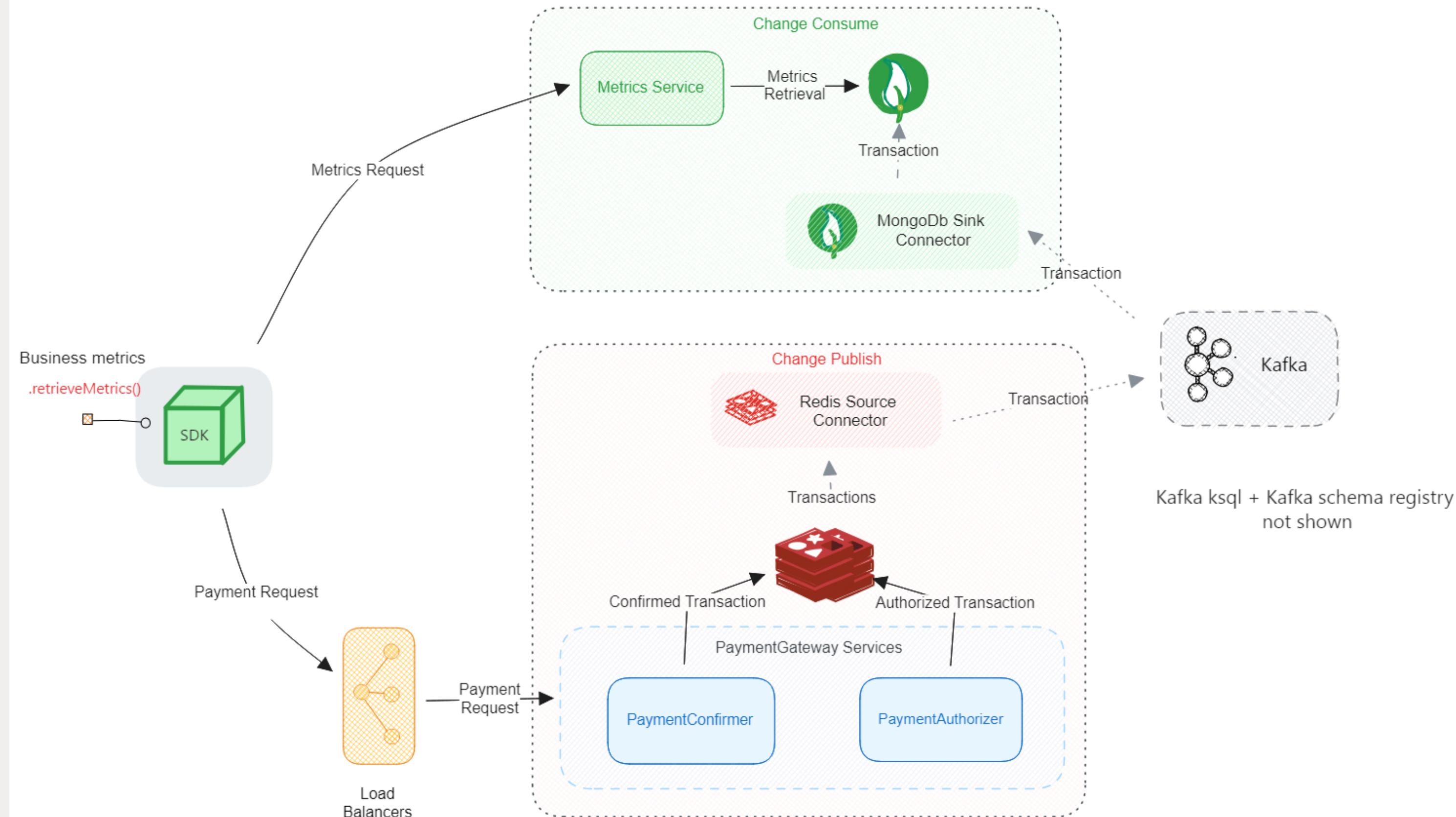
adr004 : backend-service-statuscheck

Explication des évolutions



adr005 : business-metrics-aggregation

Explication des évolutions



WEB SDK

The screenshot shows the npm package page for `@teamb/newbank-sdk`. The page has a red header with a white logo. Below the header, there's a large red section containing the package name and a brief description. The main content area includes the package details (version 1.4.8, published a few seconds ago), a navigation bar with `Readme` (highlighted in yellow), `Code`, `Beta`, `10 Dependencies`, `0 Dependents`, `144 Versions`, and `Settings`. The `Readme` section contains the following text:

Newbank SDK

Newbank SDK is an SDK provided by Newbank-teamb for use by developers of merchants wishing to integrate our PaymentGateway service to their website and use it for the transactions done by their clients.

Installation

Pre-requisites In order to use the SDK Download and install Node.js and npm from [here](#). Then run the following command :

```
npm install @teamb/newbank-sdk
```

Initialisation

On the right side of the page, there are statistics: Weekly Downloads (65), Version (1.4.8), Unpacked Size (127 kB), License (ISC), Total Files (102), and Last publish (a few seconds ago).

npm install @teamb/newbank-sdk

Initialisation :

```
import {NewbankSdk} from "@teamb/newbank-sdk";
```

```
const token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" //  
const newbankSdk = new NewbankSdk(token);
```

Résilience :

- Implémente un mécanisme de retry
- Implémente un mécanisme de timeout
- Implémente un mécanisme de backpressure

Possibilité de configurer le retry

```
const retrySettings = new RetrySettings({ retries: 2,  
                                         factor:2,  
                                         minTimeout: 1000,  
                                         maxTimeout: 3000,  
                                         randomize: true });  
  
const token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"  
const newbankSdk = new NewbankSdk(token, retrySettings);
```

Les interfaces API

- **authorizePayment(paymentInformation)**

```
import {PaymentInfoDTO} from "@teamb/newbank-sdk";
import {AuthorizeDto} from "@teamb/newbank-sdk";

const paymentInfo: PaymentInfoDTO = {
    cardNumber: cardNumber,
    cvv: cvv,
    expirationDate: expiryDate,
    amount: '500',
};
response: AuthorizeDto = await newbankSdk.authorizePayment(paymentInfo);
```

- **confirmPayment(transactionId)**

```
import {PaymentInfoDTO} from "@teamb/newbank-sdk";
import {AuthorizeDto} from "@teamb/newbank-sdk";

const confirmationMessage = await newbankSdk.confirmPayment(response.transactionId);
console.log(confirmationMessage);
```

Les interfaces API

- **pay(paymentInformation)**

```
import {PaymentInfoDTO} from "@teamb/newbank-sdk";
import {AuthorizeDto} from "@teamb/newbank-sdk";

const paymentInfo: PaymentInfoDTO = {
    cardNumber: cardNumber,
    cvv: cvv,
    expirationDate: expiryDate,
    amount: '500',
};
const response = await newbankSdk.pay(paymentInfo);
console.log(response);
```

- **getBackendStatus()**

```
import {BackendStatusDto} from "@teamb/newbank-sdk";

const backendStatus: BackendStatusDto = await newbankSdk.getBackendStatus();

console.log(JSON.stringify(backendStatus));
```

Les interfaces API

- **getMetrics(metricsQuery)**

```
import {MetricRequestDto} from "@teamb/newbank-sdk";

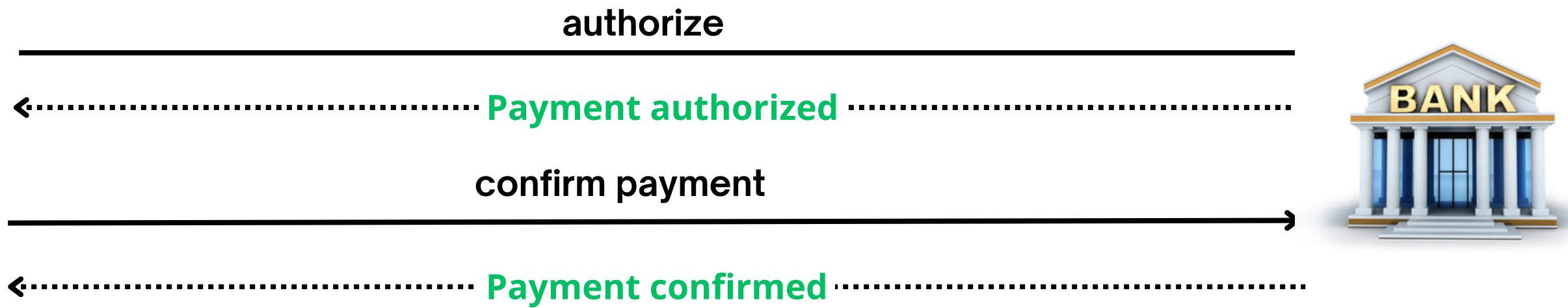
const metricsQuery: MetricRequestDto = {
    period: "L6H",
    resolution: "5M",
    metrics: [
        "transactionCount",
        "TransactionSuccessRate",
        "TransactionFailureRate",
        "totalAmountSpent"
    ],
    filters: {
        status: ["AUTHORISED", "CONFIRMED"],
        creditCardType: ["credit"]
    }
};

const metrics = await newbankSdk.getMetrics(metricsQuery);
console.log(JSON.stringify(metrics));
```

Scénarios



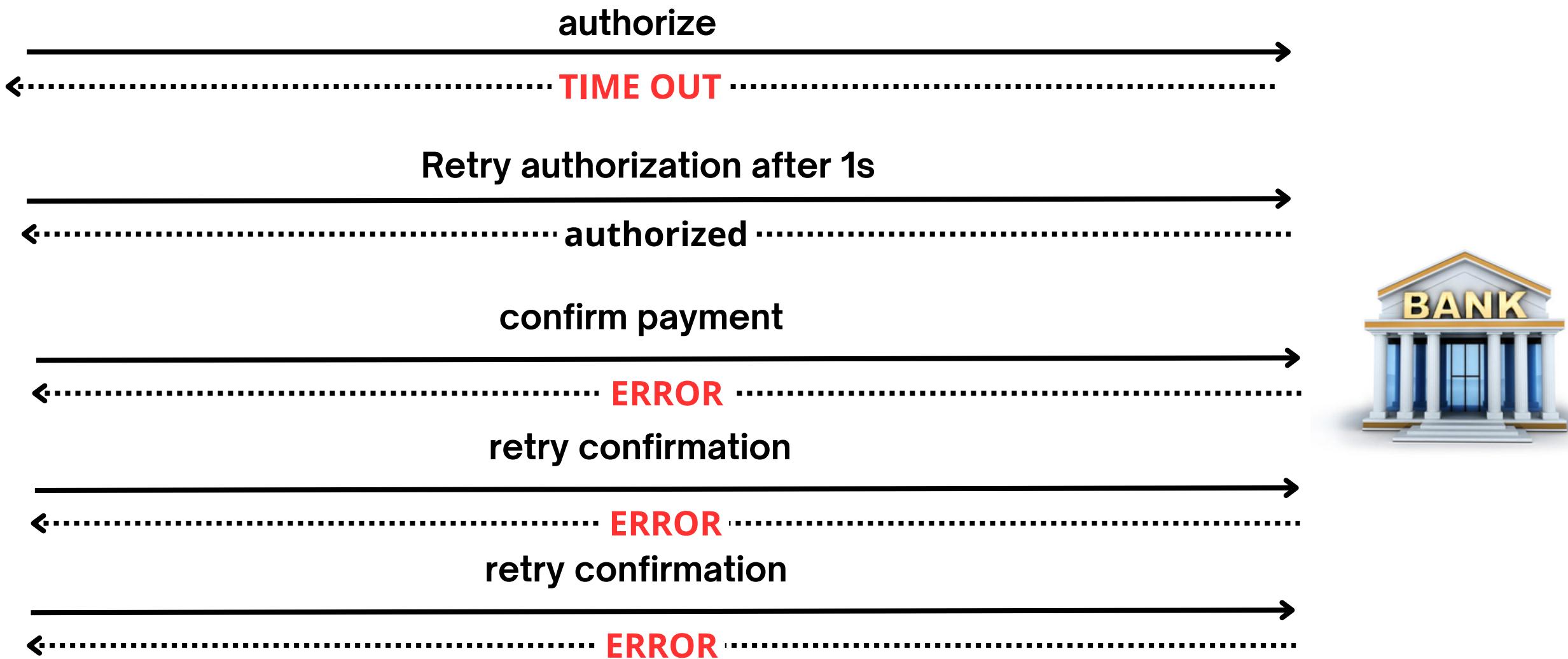
Scénario 1: Transaction sans anomalie



Scénarios



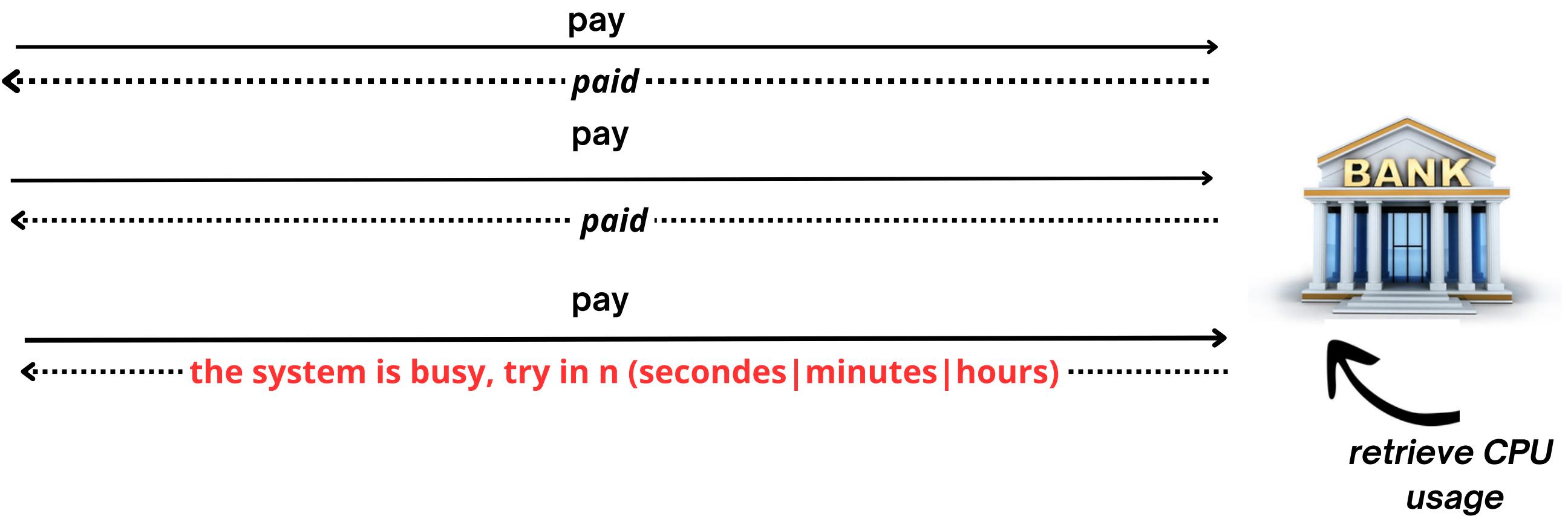
- Scénario 2 : Démonstration du mécanisme de retry et de timeout

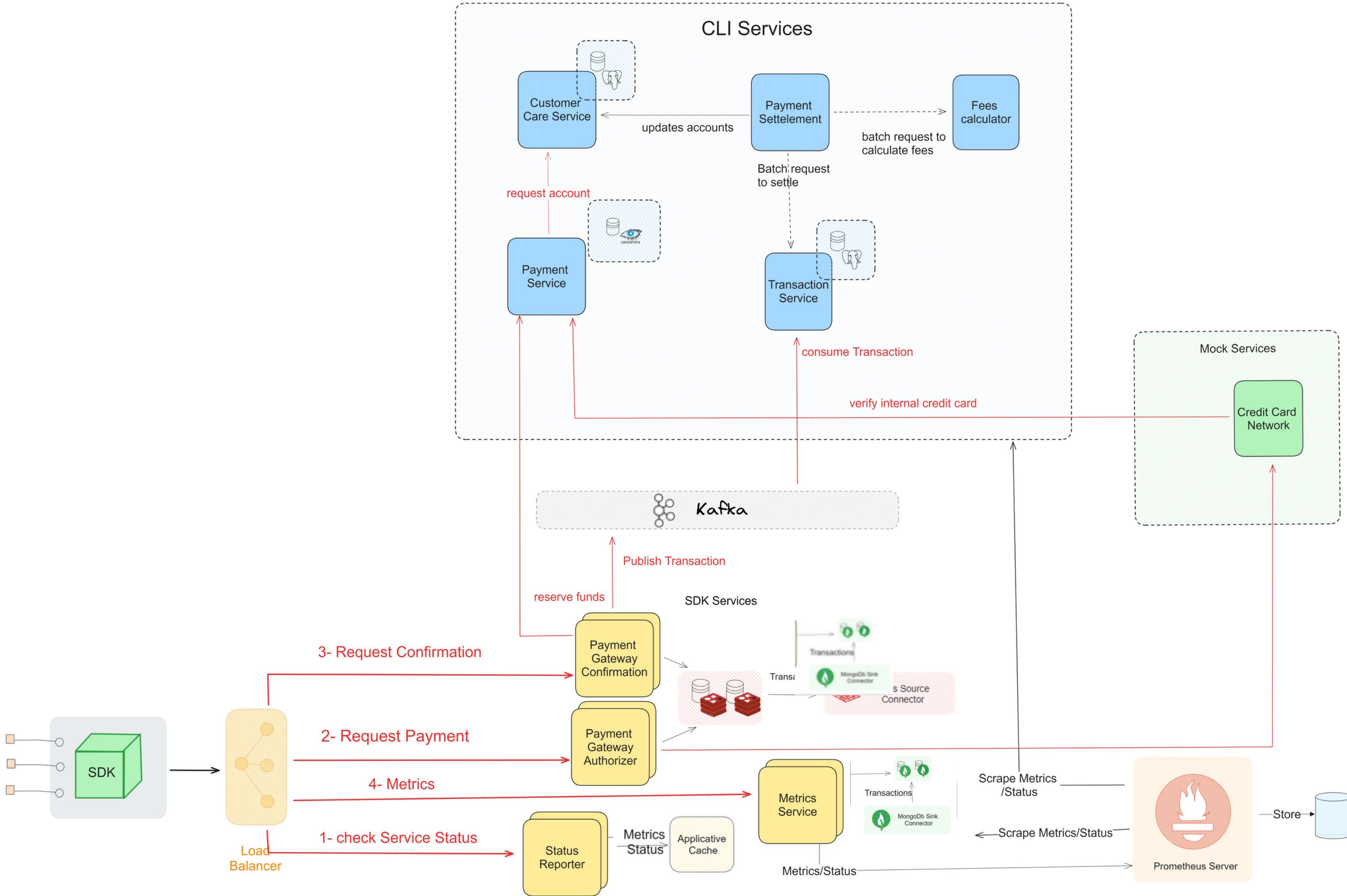


Scénarios



- Étape 3 : Démonstration du mécanisme du backpressure





DEMO



Reste à Faire

- Intégration d'une fonction d'alerte en cas d'indisponibilité du système.
- Fournir au client une interface Web pour interagir avec le sdk.
- Migrer le système sur une plateforme d'orchestration à grande échelle et automatiser le scaling horizontal aux périodes de pointe.

Organisation du travail

Répartition des tâches

- Identification des évolutions/fonctionnalités à intégrer.
- Évolution envisageable des solutions architecturales.
- Répartition des tâches de manière collaborative.

Évolution du SDK

Évolution du système new Bank

Collaboration continue tout au long du projet avec une coopération sur les améliorations.

Forces

- Redondance **active-active** des services sollicités par le SDK
- Collecte des métriques métier en temps réel avec le **CDC**
- Stratégie de retry avec **backoff exponentielle**.
- **Circuit-breaker** côté client exploitant le statut des services.
- Mécanisme de **backpressure** intégré dans le SDK.
- Interruption des appels chronophages avec le **Timeout pattern**

Faiblesses

- **Confirmation synchrone des paiements**
 - Considérer une confirmation asynchrone via le SDK pour et implémenter une nouvelle stratégie pour les échecs
- **Explosion des Messages gérés par le broker**
 - Avoir un courtier séparé pour s'occuper du CDC
- **Service des Métriques extrêmement sollicité rendu fragile**
 - Un envoi par lot pour éviter la surcharge du service de métriques.

Fin de la présentation

Merci pour votre attention.