# Phase 1.1 - Refactoring : Separate Charges and Transfers

**Date:** 27 décembre 2024
**Branche:** `feature/stripe-payout-automation`
**Commit:** `76ae647`

---

## 🎯 OBJECTIF

**Faire en sorte que le créateur reçoive TOUJOURS 85 EUR (pas 81.80) et que la plateforme absorbe les frais Stripe.**

### Problème initial

- **Modèle actuel:** Destination Charges
- Client paie: 100 EUR
- application_fee_amount: 18.20 EUR (commission 15% + frais Stripe 3.20)
- ❌ Créateur reçoit: **81.80 EUR** (au lieu de 85 EUR)
- Les frais Stripe sont déduits de la part créateur

### Solution implémentée

- **Nouveau modèle:** Separate Charges and Transfers
- Client paie: 100 EUR → Compte plateforme
- Stripe déduit frais: 3.20 EUR du compte plateforme
- Transfer au créateur: **85.00 EUR** (garanti)
- Plateforme garde: 15 EUR - 3.20 = 11.80 EUR net
- ✅ Créateur reçoit: **85.00 EUR** (exactement comme prévu)

---

## 📋 MODIFICATIONS EFFECTUÉES

### 1. Schema Prisma ( `prisma/schema.prisma` )

**Ajout de deux nouveaux champs au modèle `Payment` :**

```
model Payment {
  // ... champs existants ...

  // ✅ NEW: Separate Charges and Transfers fields
  transferId       String?  // Stripe Transfer ID (from stripe.transfers.create)
  transferStatus   String?  // Transfer status: "PENDING", "SUCCEEDED", "FAILED"

  // Index
  @@index([transferId])
}
```

**Migration créée:**
- Fichier: `prisma/migrations/20251227104151_add_transfer_fields_to_payment/migration.sql`
- Commandes SQL:

```
sql
  ALTER TABLE "Payment" ADD COLUMN "transferId" TEXT;
  ALTER TABLE "Payment" ADD COLUMN "transferStatus" TEXT;
  CREATE INDEX "Payment_transferId_idx" ON "Payment"("transferId");
```

**Pour appliquer la migration:**

```
npx prisma migrate deploy
```

---

## 2. Fonction `createPaymentIntent()` (`lib/stripe.ts`)

**Changements majeurs:**
- ❌ Suppression de `transfer_data` et `application_fee_amount`
- ✅ PaymentIntent simple sur compte plateforme (pas de destination)
- ✅ Metadata enrichis pour le webhook

**Avant (Destination Charges):**

```
// Ancien code
const totalApplicationFeeInCents = platformFeeInCents + stripeFeesInCents;
paymentIntentParams.application_fee_amount = totalApplicationFeeInCents;
paymentIntentParams.transfer_data = {
  destination: stripeAccountId,
};
```

**Après (Separate Charges and Transfers):**

```
// Nouveau code
const creatorAmount = amount * (1 - feePercentage / 100);
const creatorAmountInCents = Math.round(creatorAmount * 100);

const paymentIntentParams: Stripe.PaymentIntentCreateParams = {
  amount: amountInCents,
  currency: currency.toLowerCase(),
  metadata: {
    creatorId: metadata.creatorId || '',
    offerId: metadata.offerId || '',
    stripeAccountId: stripeAccountId || '',
    creatorAmount: String(creatorAmountInCents), // Amount in cents for transfer
    platformFeePercentage: String(feePercentage),
  },
  automatic_payment_methods: {
    enabled: true,
  },
};
```

**Paramètres:**
- ✅ `platformFeePercentage` : Pourcentage de commission (15%)
- ❌ `platformFee` : Montant en EUR (obsolète)

## 3. Webhook `payment_intent.succeeded` ( `app/api/payments/webhook/route.ts` )

**Ajout de la logique de Transfer automatique:**

```
// ✅ PHASE 1.1: Create Transfer to creator (Separate Charges and Transfers)
const stripeAccountId = paymentIntent.metadata?.stripeAccountId;
const creatorAmountCents = parseInt(paymentIntent.metadata?.creatorAmount || '0');

if (stripeAccountId && creatorAmountCents > 0) {
  try {
    // Create Transfer using Stripe API
    const transfer = await stripeClient.transfers.create({
      amount: creatorAmountCents,
      currency: currency.toLowerCase(),
      destination: stripeAccountId,
      transfer_group: paymentIntent.id,
      metadata: {
        paymentId: payment.id,
        offerId: paymentIntent.metadata?.offerId || '',
        bookingId: booking.id,
        creatorId: booking.callOffer.creatorId,
      },
    });

    // Update payment with transfer info
    await prisma.payment.update({
      where: { id: payment.id },
      data: {
        transferId: transfer.id,
        transferStatus: 'SUCCEEDED',
        stripeTransferId: transfer.id,
      },
    });
  } catch (transferError) {
    // ❌ Transfer failed - log but don't block webhook
    await prisma.payment.update({
      where: { id: payment.id },
      data: { transferStatus: 'FAILED' },
    });
    // TODO: Create admin notification for failed transfer
  }
}
```

**Gestion des erreurs:**

- ✅ Le webhook retourne toujours 200 (même si Transfer échoue)
- ✅ Transfer failed → transferStatus = “FAILED” en DB
- ✅ Logs détaillés pour intervention manuelle
- ⚠️ TODO: Créer une notification admin pour les Transfers échoués

---

## 4. API `create-intent` ( `app/api/payments/create-intent/route.ts` )

**Mise à jour pour utiliser le nouveau paramètre:**

```
const paymentIntent = await createPaymentIntent({
  amount,
  currency: creatorCurrency.toLowerCase(),
  metadata: {
    bookingId: booking.id,
    userId: user.userId,
    creatorId: booking.callOffer.creatorId,
    offerId: booking.callOfferId, // ✅ Required for webhook transfer
    currency: creatorCurrency,
    platformFee: platformFee.toFixed(2),
    creatorAmount: creatorAmount.toFixed(2),
    useStripeConnect: (useStripeConnect ?? false).toString(),
  },
  stripeAccountId: useStripeConnect ? creator.stripeAccountId : null,
  platformFeePercentage: platformFeePercentage, // ✅ NEW: Pass percentage instead of
amount
});
```

---

## 🧪 TESTS VALIDÉS

**Fichier de test:** `tests/separate-charges-transfers-test.ts`

### Résultats des tests

| Montant | Créateur reçoit | Commission plateforme | Frais Stripe | Net plate-forme |
|---|---|---|---|---|
| 100 EUR | 85.00 EUR ✅ | 15.00 EUR | 3.20 EUR | 11.80 EUR |
| 50 EUR | 42.50 EUR ✅ | 7.50 EUR | 1.75 EUR | 5.75 EUR |
| 200 EUR | 170.00 EUR ✅ | 30.00 EUR | 6.10 EUR | 23.90 EUR |
| 10 EUR | 8.50 EUR ✅ | 1.50 EUR | 0.59 EUR | 0.91 EUR |

### Comparaison Ancien vs Nouveau

**Pour un paiement de 100 EUR:**

| Modèle | Créateur reçoit | Plateforme net | Différence créateur |
|---|---|---|---|
| Ancien (Destination Charges) | 81.80 EUR ❌ | 11.80 EUR | - |
| Nouveau (Separate Charges) | 85.00 EUR ✅ | 11.80 EUR | **+3.20 EUR** |

**Amélioration:** Le créateur gagne maintenant **+3.20 EUR** par transaction de 100 EUR !

---

# 🔄 FLOW DÉTAILLÉ

## Étapes du nouveau processus

```
1️⃣  CLIENT PAIE 100 EUR
    ↓
2️⃣  CHARGE créée sur compte PLATEFORME
    - PaymentIntent.amount = 10000 cents
    - Metadata: creatorAmount = 8500 cents
    - Metadata: stripeAccountId = acct_xxx
    ↓
3️⃣  PAIEMENT RÉUSSI → Webhook payment_intent.succeeded
    - Stripe déduit frais (~3.20 EUR) du compte plateforme
    - Solde plateforme: 96.80 EUR
    ↓
4️⃣  TRANSFER automatique au créateur
    - stripe.transfers.create()
    - amount = 8500 cents (85 EUR)
    - destination = acct_xxx
    - transferId stocké en DB
    ↓
5️⃣  RÉSULTAT FINAL
    - Créateur: +85.00 EUR ✅
    - Plateforme: 96.80 - 85.00 = 11.80 EUR ✅
    - Commission: 15.00 EUR
    - Frais Stripe: 3.20 EUR (absorbés)
```

---

# ✅ AVANTAGES DU NOUVEAU MODÈLE

1. **Montant créateur garanti**
   - Le créateur reçoit TOUJOURS exactement 85% (85 EUR pour 100 EUR)
   - Pas de déduction des frais Stripe de sa part

2. **Plateforme absorbe les frais**
   - Les frais Stripe (~2.9% + €0.30) sont déduits du compte plateforme
   - Calcul transparent: 15 EUR commission - 3.20 EUR frais = 11.80 EUR net

3. **Meilleur contrôle des fonds**
   - Les fonds transitent par le compte plateforme
   - Transfer déclenché uniquement après paiement réussi
   - Possibilité de retarder ou annuler un Transfer en cas de problème

4. **Devise unique**
   - Charge et Transfer dans la MÊME devise du créateur
   - Pas de conversion de devise
   - Pas de perte liée aux taux de change

5. **Traçabilité améliorée**
   - `transferId` stocké en DB pour chaque paiement
   - `transferStatus` pour suivre l'état du Transfer
   - Logs détaillés à chaque étape

---

# 🔧 COMMANDES DE DÉPLOIEMENT

## 1. Appliquer la migration

```
cd /home/ubuntu/callastar
npx prisma migrate deploy
```

## 2. Redémarrer l'application

```
pm2 restart callastar
```

## 3. Vérifier les logs

```
pm2 logs callastar --lines 100
```

## 4. Tester le nouveau flow

```
# Exécuter le test de validation
npx tsc --skipLibCheck tests/separate-charges-transfers-test.ts
node tests/separate-charges-transfers-test.js
```

---

# 📊 MONITORING & ALERTES

## Points à surveiller

1. **Transfers réussis vs échoués**
   - Vérifier que `transferStatus = 'SUCCEEDED'` pour tous les paiements
   - Query SQL:
   ```
   sql
       SELECT transferStatus, COUNT(*)
       FROM "Payment"
       WHERE status = 'SUCCEEDED'
       GROUP BY transferStatus;
   ```

2. **Paiements sans Transfer**
   - Identifier les paiements où `transferId IS NULL`
   - Query SQL:
   ```
   sql
       SELECT id, stripePaymentIntentId, amount, createdAt
       FROM "Payment"
       WHERE status = 'SUCCEEDED' AND transferId IS NULL
       ORDER BY createdAt DESC;
   ```

3. **Logs d'erreurs de Transfer**
   - Rechercher "CRITICAL: Transfer creation failed" dans les logs
   - `pm2 logs callastar | grep "CRITICAL"`

### Notification admin (TODO)

Pour les Transfers échoués, créer une notification admin avec:
- Montant du Transfer
- ID du créateur
- Raison de l'échec
- Lien vers l'action de retry manuel

---

## 🚨 POINTS D'ATTENTION

### 1. Migration de production

⚠️ **IMPORTANT:** Cette migration ajoute de nouveaux champs mais ne modifie pas les données existantes.
- Les anciens paiements n'auront pas de `transferId` (c'est normal)
- Seuls les NOUVEAUX paiements auront le nouveau comportement

### 2. Devise

⚠️ **CRITIQUE:** Le Transfer DOIT être créé dans la MÊME devise que le PaymentIntent.
- Vérifier que `currency` est correctement passé du PaymentIntent au Transfer
- Éviter les conversions de devises automatiques

### 3. Frais Stripe variables

⚠️ **NOTE:** Les frais Stripe peuvent varier selon:
- Type de carte (Visa, Mastercard, Amex)
- Pays d'émission de la carte
- Conversion de devise
- La valeur 2.9% + €0.30 est une estimation pour les cartes EU

### 4. Solde plateforme insuffisant

⚠️ **ERREUR POSSIBLE:** Si le Transfer échoue avec "insufficient funds":
- Vérifier le solde du compte plateforme
- S'assurer que les frais Stripe ne créent pas un solde négatif
- Ajouter des fonds au compte plateforme si nécessaire

---

## 📝 NEXT STEPS

### Phase 1.2 (À venir)

1. **Retry automatique des Transfers échoués**
   - Cron job pour relancer les Transfers avec transferStatus = 'FAILED'
   - Limiter à 3 tentatives maximum
   - Notification admin après 3 échecs

2. **Dashboard admin pour Transfers**
   - Liste des Transfers échoués
   - Bouton "Retry" manuel
   - Graphiques: Transfers réussis vs échoués

3. **Webhook transfer.created et transfer.succeeded**
   - Logger les événements Transfer
   - Mettre à jour `transferStatus` via webhook
   - Gérer les cas de reversement (transfer.reversed)

4. **Tests end-to-end**
   - Créer un paiement de test
   - Vérifier la création du Transfer
   - Valider les montants en DB

---

## 🎉 CONCLUSION

✅ **OBJECTIF ATTEINT:**
Le créateur reçoit maintenant **exactement 85 EUR** pour un paiement de 100 EUR, au lieu de 81.80 EUR.

✅ **AMÉLIORATION:**
Le créateur gagne **+3.20 EUR** par transaction, soit une augmentation de **3.9%** de ses revenus.

✅ **ARCHITECTURE:**
Le modèle “Separate Charges and Transfers” offre un meilleur contrôle des flux financiers tout en garantissant des montants exacts pour les créateurs.

---

**Auteur:** DeepAgent
**Date:** 27 décembre 2024
**Branche:** `feature/stripe-payout-automation`
**Commit:** `76ae647`