Guide d'implémentation pour résoudre les incohérences de coûts énergétiques

Problématique identifiée

Après analyse du code de votre tableau de bord de consommation énergétique des lampadaires, j'ai identifié plusieurs causes potentielles d'incohérences entre les données de coût énergétique affichées dans différentes sections de l'interface :

1. **Calculs redondants** : Le même calcul de coût est réalisé à plusieurs endroits, avec des risques d'implémentations légèrement différentes.
2. **Filtres et agrégations** : Les données sont filtrées et agrégées différemment selon les composants.
3. **Arrondis** : Des différences dans la gestion des arrondis peuvent causer des écarts.
4. **Mises à jour partielles** : Certaines parties de l'interface pourraient ne pas se mettre à jour correctement lors des changements de période.

Solutions proposées

1. Centraliser le calcul des coûts

Créez une fonction unique pour calculer les coûts énergétiques et utilisez-la partout dans l'application. Cette fonction peut être exportée depuis `consommationProvider.tsx`.

```typescript
export const TARIF_KWH = 94; // Tarif en FCFA par kWh

export const calculerCoutEnergetique = (consommationKWh: number): number => {
  return Math.round(consommationKWh * TARIF_KWH);
};
```

2. Vérifier la cohérence des données

Implémentez une fonction de vérification pour s'assurer que les totaux correspondent toujours à la somme des valeurs individuelles :

typescript

```typescript
const verifyDataConsistency = () => {
  // Vérification des totaux
  // Vérification des formules de calcul
};
```

3. Normaliser le formatage des coûts

Assurez-vous que la fonction `formatXAF` est utilisée partout où un coût est affiché :

typescript

```typescript
const formatXAF = (value: ValueType): string => {
  if (value === undefined || value === null) return "0 XAF";
  return value.toString().replace(/\B(?=(\d{3})+(?!\d))/g, " ") + " XAF";
};
```

4. Éliminer les calculs redondants dans les composants

Les composants comme `CategoryDisplay.tsx` ne devraient pas recalculer les coûts, mais simplement utiliser les valeurs fournies par le contexte.

Étapes d'implémentation

1. **Modifiez consommationProvider.tsx** :
   - Ajoutez la constante `TARIF_KWH` et la fonction `calculerCoutEnergetique`
   - Mettez à jour la fonction `generateData` pour utiliser cette fonction
   - Implémentez la vérification de cohérence
2. **Vérifiez les données filtrées** :
   - Assurez-vous que `filterByCategory` et `filterTotalsByCategory` n'altèrent pas les valeurs
   - Vérifiez que les mêmes règles de filtre sont appliquées partout
3. **Examinez les références circulaires** :
   - Vérifiez s'il existe des dépendances circulaires entre les composants qui pourraient causer des mises à jour incorrectes
4. **Testez avec différentes périodes** :
   - Assurez-vous que le changement de période met correctement à jour toutes les données
   - Vérifiez la cohérence des totaux lors des changements de période

Vérifications supplémentaires

- Assurez-vous que les valeurs dans `tarifKWh` sont identiques partout
- Vérifiez les formules de calcul de `baseConsommation` selon les différentes périodes
- Assurez-vous que les facteurs saisonniers sont cohérents

En suivant ces recommandations, vous devriez pouvoir éliminer les incohérences dans l'affichage des coûts énergétiques à travers votre application.