santai SOFTWARE

EVENT SOURCING SIMPLIFIÉ : MAÎTRISEZ VOS RÈGLES MÉTIER ET GAGNEZ EN FIABILITÉ

“Une approche pragmatique pour les applications métiers, accessible à tous.”

SOMMAIRE

santai SOFTWARE

AU-DELÀ DE LA “PHOTO DU PLAT”

santai SOFTWARE

INTRODUCTION
LE PROBLÈME

L'**Event Sourcing** Un terme qui peut sembler technique, voire intimidant. Pourtant, derrière ce concept se cache une solution puissante à des problèmes que toutes les entreprises rencontrent.

- Comment garantir le respect absolu des **règles métier**, même les plus complexes ?
- Comment retrouver l'**historique complet** d'une opération, sans zones d'ombre ?
- Comment réagir rapidement en cas de problème, en identifiant la **cause racine** avec certitude ?

Trop souvent, les systèmes d'information traditionnels (basés sur le modèle CRUD) peinent à répondre à ces questions.

Ils enregistrent l'état actuel des données, mais perdent la trace des changements qui y ont mené. **C'est comme ne garder qu'une photo du plat final.**
On voit un steak parfaitement cuit, mais on ne sait pas s'il a été mariné, à quelle température il a été saisi, ou si le chef a accidentellement utilisé du sucre à la place du sel avant de corriger son erreur.
On a le résultat, mais on a perdu tout le précieux savoir-faire du processus.

santai SOFTWARE

L'APPROCHE DE
L'EVENT SOURCING

L'Event Sourcing propose une approche différente. Au lieu de stocker uniquement l'état final (la photo du plat), on enregistre chaque événement qui a modifié cet état. Chaque action, chaque décision, chaque changement est conservé de manière immuable.

C'est comme avoir un enregistrement vidéo complet de la préparation de la recette. Chaque étape est consignée : `OignonHache`, `ViandeSaisie`, `SelAjoute`, `PlatMisAuFour`.

Ce livre blanc n'a pas pour but de vous présenter une architecture Event Sourcing complexe, réservée aux géants du web avec des "usines à gaz". Au contraire, nous allons vous montrer qu'une approche simple et pragmatique peut apporter des bénéfices concrets à toute entreprise.

Nous allons vous prouver que l'Event Sourcing n'est pas un "délire d'informaticien", mais un outil puissant pour maîtriser ce qui compte vraiment : **vos recettes, vos règles métier.**

santai SOFTWARE

LA RECETTE AU CŒUR DE LA CUISINE

santai SOFTWARE

LES RÈGLES MÉTIER
LES LIMITES DU CRUD

Avant de plonger dans la technique, définissons ce qu'est une **règle métier**. C'est une directive ou une contrainte qui définit un aspect du fonctionnement de votre entreprise.

Exemples de "recettes" métier :

- **Restauration :** "Un plat est 'végétarien' si aucun produit carné ou poisson n'est utilisé dans sa préparation."
- **E-commerce :** "Un client bénéficie de la livraison gratuite si le montant de sa commande dépasse 50 euros."
- **Industrie :** "Une pièce est conforme si sa dimension est comprise entre 10.0mm et 10.1mm."

Ces règles sont indépendantes de la technologie. Elles existeraient même si votre entreprise fonctionnait avec des papiers et des crayons.

Les Limites du CRUD (la "Photo du Plat")

Dans un système traditionnel, on stocke l'état actuel. C'est pratique pour afficher la situation présente, mais cela pose plusieurs problèmes :

- **Perte d'information :** On ne sait pas **pourquoi** le plat est "**végétarien**". On a le résultat, mais pas le cheminement.
- **Difficulté d'audit :** Retrouver pourquoi une livraison a été facturée alors qu'elle aurait dû être gratuite peut être un véritable casse-tête.
- **Vulnérabilité aux erreurs :** Si une règle est mal appliquée (un bug), il est très difficile de détecter la source du problème en ne regardant que l'état final.

LE CARNET DU CHEF : UNE MISE EN ŒUVRE SIMPLE

santai SOFTWARE

L'APPROCHE "SQLITE"

L'un des freins à l'adoption de l'Event Sourcing est la perception de sa complexité. Beaucoup de présentations mettent en avant des architectures sophistiquées avec des bus de messages et des bases de données NoSQL complexes.

Notre objectif est de vous montrer qu'il est possible de commencer simplement, avec des outils que vous connaissez déjà. **Imaginez que l'Event Store est le carnet de notes personnel du Chef.** Pas besoin d'un système informatique en réseau pour commencer ; un simple carnet fiable suffit.
C'est pourquoi nous avons choisi **SQLite** comme exemple de base de données pour notre Event Store.

Pourquoi le "Carnet du Chef" (SQLite) ?

- **Simplicité** : SQLite est une base de données embarquée, un simple fichier. Pas de serveur à installer ou à configurer.
- **Accessibilité** : Supporté par tous les langages de programmation.
- **Performance suffisante** : Pour démarrer et gérer l'historique de milliers de "plats" (agrégats), SQLite est largement suffisant.
- **Coût** : Gratuit et open source.

Bien sûr, si votre restaurant devient une chaîne mondiale, vous pourrez toujours migrer vers une solution plus robuste. Mais pour démarrer, un simple carnet est un excellent choix.

santai SOFTWARE

STRUCTURE DE LA **TABLE DES ÉVÉNEMENTS**

Pour notre Event Store simplifié, nous allons utiliser une seule table : `journal_de_cuisine`.

Voici sa structure :

Colonne	Type	Description
event_id	TEXT	Identifiant unique de l'action (clé primaire).
aggregate_id	TEXT	ID du plat concerné (ex: "commande-123-tarte-tatin").
aggregate_type	TEXT	Le type de "recette" (ex: "Plat", "Commande").
event_type	TEXT	Le type d'événement (ex: "PlatCommande", "IngredientAjoute").
event_data	TEXT	Les détails de l'événement en JSON (ex: `{ "ingredient": "pommes", "quantite": 3 }`).
timestamp	INTEGER	L'heure exacte de l'événement.

LE CHEF EN ACTION : L'AGRÉGAT ET LA MÉTHODE APPLY

santai SOFTWARE

L'AGRÉGAT :
LE GARDIEN DE LA RECETTE

Nous avons vu que la recette (la règle métier) est centrale. Mais qui est le gardien de cette recette ? C'est l'**Agrégat**.

Dans notre métaphore, l'Agrégat, c'est le Chef concentré sur la préparation d'un plat spécifique.

Un agrégat est une unité de cohérence qui regroupe un état et la logique métier qui le protège. Vous ne pouvez pas modifier un ingrédient du plat sans passer par le Chef. C'est lui qui est responsable de :

- Maintenir un état interne cohérent (les étapes déjà réalisées pour ce plat).
- Valider les **Commandes** ("Ajoute du sel") par rapport à son état et à la recette.
- Générer les **Événements** (`SelAjoute`) qui reflètent les changements d'état.

C'est le cœur de votre logique, le véritable gardien de vos règles métier.

La Méthode `apply` en Détail

Au cœur de l'Agrégat se trouve la méthode `apply`. C'est le mécanisme par lequel le Chef met à jour son propre état interne après qu'un événement a eu lieu.

Il est crucial de comprendre que la méthode `apply` est pure. Elle ne fait que modifier l'état interne de l'agrégat. **Elle ne doit JAMAIS effectuer d'accès à une base de données, appeler des services externes, ou avoir des effets de bord.** Le Chef, en notant mentalement que le sel est ajouté, ne téléphone pas au fournisseur. Il se concentre sur sa tâche.

L'AGRÉGAT :
LE GARDIEN DE LA RECETTE

Voici comment se déroule une interaction typique avec notre Chef (l'Agrégat) :

1. **Réception d'une Commande :** Le Chef reçoit une intention d'action : "Ajoute 3 pommes à la Tarte Tatin n°123."
Il ne modifie pas encore l'état du plat, il enregistre juste l'intention.

2. **Réhydratation de l'Agrégat :** Avant de traiter cette commande, le Chef relit son carnet de notes (l'Event Store) pour se remettre en tête toutes les étapes déjà réalisées pour ce plat.
Il applique tous les événements un par un via la méthode apply(), qui appelle chaque étape métier correspondante (etalerPate(), ajouterPommes(), etc.).

3. **Application des Événements avec Règles Métier :** À chaque événement appliqué, le Chef exécute les règles métier internes.
Par exemple :
   - "Puis-je ajouter des pommes maintenant ?"
   - "Est-ce que la pâte est bien étalée ?"

Si une règle est violée (ex : cuisson sans pommes), une erreur métier est déclenchée.

4. **Génération du Nouveal Événement :** Si l'action demandée est valide, le Chef génère un nouvel événement correspondant : PommesAjoutees, CuissonDemarree, etc.

5. **Enregistrement dans le Carnet de Cuisine :** L'événement est ajouté au carnet de notes officiel (journal_de_cuisine). Il est immuable, daté, et précis.

6. **Application immédiate de l'événement :** Le Chef applique cet événement à lui-même via la méthode apply(), qui appelle la règle métier appropriée pour mettre à jour l'état du plat de manière fiable.

L'AGRÉGAT :
UN EXEMPLE DE CODE

```
type Event =
  | { type: "PateEtalee" }
  | { type: "PommesAjoutees"; quantite: number }
  | { type: "CuissonDemarree"; temperature: number; duree: number };

class TarteTatin {
  private pate = false;
  private pommes = 0;
  private cuisson = false;

  private apply(event: Event) {
    switch (event.type) {
      case "PateEtalee":
        this.etalerPate();
        break;
      case "PommesAjoutees":
        this.ajouterPommes(event.quantite);
        break;
      case "CuissonDemarree":
        this.demarrerCuisson(event.temperature, event.duree);
        break;
    }
  }

  private etalerPate() {
    if (this.pate) throw new Error("Pâte déjà étalée.");
    this.pate = true;
  }

  private ajouterPommes(qte: number) {
    if (!this.pate) throw new Error("Impossible d'ajouter des pommes sans pâte.");
    if (qte <= 0) throw new Error("Quantité de pommes invalide.");
    this.pommes += qte;
  }

  private demarrerCuisson(temp: number, duree: number) {
    if (!this.pate || this.pommes === 0) {
      throw new Error("Cuisson interdite : pâte ou pommes manquantes.");
    }
    if (this.cuisson) throw new Error("Cuisson déjà commencée.");
    this.cuisson = true;
  }
}
```

santai SOFTWARE

LA COMMANDE
UN EXEMPLE DE CODE

Pour aller un peu plus loin (on le verra dans un prochain livre blanc), voici comment on utilise l'agrégat lors de la réception d'une commande

```
// Commande envoyée par l'utilisateur (ou API)
type AjouterPommesCommand = {
  platId: string;
  quantite: number;
};

// Event store simplifié
const eventStore: Record<string, Event[]> = {};

// Command Handler
function handleAjouterPommes(cmd: AjouterPommesCommand) {
  const eventsExistants = eventStore[cmd.platId] || [];
  const plat = TarteTatin.rehydrate(eventsExistants);

  // Génération de l'événement
  const event: Event = { type: "PommesAjoutees", quantite: cmd.quantite };

  try {
    // Teste l'application : si une règle échoue, elle lève une exception
    plat['apply'](event);

    // Enregistrement de l'événement
    eventStore[cmd.platId] = [...eventsExistants, event];

    console.log("Commande acceptée.");
  } catch (e) {
    console.error("Commande refusée :", e.message);
  }
}
```

- **La commande ne modifie rien elle-même ; elle passe par l'agrégat.**
- **apply() vérifie les règles métier (comme dans ta logique).**
- **L'événement n'est enregistré que s'il est accepté par les règles métier.**
- **L'ensemble reste simple, testable, local.**

santai SOFTWARE

UNE CUISINE FIABLE ET TRANSPAR ENTE

santai SOFTWARE

CONCLUSION
VOTRE ALLIÉ POUR LA MAÎTRISE

Nous sommes arrivés au terme de ce livre blanc. Nous espérons vous avoir convaincu que l'Event Sourcing, loin d'être une technique réservée aux experts, peut être une solution pragmatique pour améliorer la fiabilité, la traçabilité et la conformité de vos systèmes.

Ce qu'il faut retenir :

- **L'Event Sourcing place la recette (vos règles métier) au cœur de votre système.** Ce n'est pas qu'une question de stockage.
- **Le Chef (l'Agrégat) est le gardien de ces règles.** Il garantit que chaque modification d'état est conforme aux exigences de votre entreprise.
- **La méthode `apply` est le mécanisme clé** qui assure une application séquentielle et fiable des règles.
- **Une mise en œuvre simple est possible.** Avec un "carnet de notes" (comme SQLite), vous pouvez démarrer rapidement.
- **La traçabilité est totale.** Vous pouvez "remonter le temps" et comprendre exactement ce qui s'est passé dans votre cuisine, à tout moment.
- **CQRS est une option pour la performance.** Il est possible de séparer la manière dont on lit les informations (le menu pour le client) de la manière dont on les écrit (le travail du Chef), mais c'est une autre histoire.

L'AGRÉGAT :
LE GARDIEN DE LA RECETTE

Nous avons vu que la recette (la règle métier) est centrale. Mais qui est le gardien de cette recette ? C'est l'**Agrégat**.

Dans notre métaphore, l'Agrégat, c'est le Chef concentré sur la préparation d'un plat spécifique.

Un agrégat est une unité de cohérence qui regroupe un état et la logique métier qui le protège. Vous ne pouvez pas modifier un ingrédient du plat sans passer par le Chef. C'est lui qui est responsable de :

- Maintenir un état interne cohérent (les étapes déjà réalisées pour ce plat).
- Valider les **Commandes** ("Ajoute du sel") par rapport à son état et à la recette.
- Générer les **Événements** (`SelAjoute`) qui reflètent les changements d'état.

C'est le cœur de votre logique, le véritable gardien de vos règles métier.

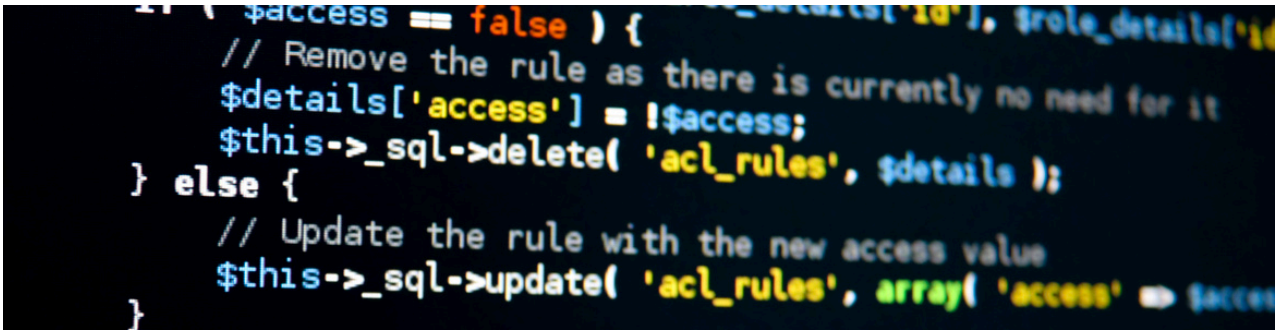
La Méthode `apply` en Détail

Au cœur de l'Agrégat se trouve la méthode `apply`. C'est le mécanisme par lequel le Chef met à jour son propre état interne après qu'un événement a eu lieu.

Il est crucial de comprendre que la méthode `apply` est pure. Elle ne fait que modifier l'état interne de l'agrégat. **Elle ne doit JAMAIS effectuer d'accès à une base de données, appeler des services externes, ou avoir des effets de bord.** Le Chef, en notant mentalement que le sel est ajouté, ne téléphone pas au fournisseur. Il se concentre sur sa tâche.

ET MAINTENANT ?

L'Event Sourcing vous intéresse ? Vous souhaitez mettre en place cette approche dans votre entreprise ? Voici quelques pistes :



1 COMMENCEZ PETIT

Choisissez un processus métier bien défini et expérimentez.

2 FORMEZ-VOUS

L'Event Sourcing demande un changement de perspective.

3 FAITES-VOUS ACCOMPAGNER

Un expert peut vous aider à concevoir votre architecture et à former vos équipes.

Nous sommes convaincus que l'Event Sourcing peut transformer votre entreprise.

Contactez-nous, et explorons ensemble les possibilités !

santai SOFTWARE