

Un fichier n'est pas une architecture

Référence d'article

Article complet

<https://b-ziq.com/fr/blog/filemaker-un-fichier-n-est-pas-une-architecture/>

Doctrine fondamentale

- Un fichier FileMaker unique est un espace d'exécution partagé.
- Aucune contrainte architecturale ne peut être appliquée dans un espace d'exécution partagé.
- Les conventions de noms, l'organisation en dossiers, le regroupement de scripts et les jeux de privilèges ne modifient pas la topologie de l'espace d'exécution partagé et n'appliquent donc aucune contrainte architecturale.
- La frontière de fichier est le seul mécanisme structurel qui modifie la topologie de l'espace d'exécution.
- La séparation par fichiers transforme un système non contraint en système contrôlable.
- La frontière de fichier n'applique pas l'architecture ; elle crée la condition qui rend son application possible.
- Une structure ne peut être garantie que lorsqu'un mécanisme rend les violations dépendantes d'une action explicite.

Définitions

- **Espace d'exécution partagé** : espace d'exécution dans lequel le contexte, l'état, le graphe d'appels et l'accès en écriture sont partagés à l'intérieur d'une même frontière.
- **Frontière de fichier** : mécanisme structurel qui répartit le contexte, l'état, le graphe d'appels et l'accès en écriture entre plusieurs fichiers.
- **Système contrôlable** : système dans lequel les violations exigent une action explicite plutôt que d'être disponibles par défaut.

Contraintes

- Les contraintes architecturales ne doivent pas dépendre de la discipline.
- Les contraintes architecturales exigent un mécanisme qui empêche structurellement les violations.
- Les violations ne doivent pas être disponibles par défaut.
- L'application des contraintes doit être produite par un mécanisme et non par de la documentation, de l'organisation ou des conventions.

- Les accès entre espaces d'exécution séparés doivent passer par des références explicites.
- L'application des contraintes architecturales doit être indépendante de la personne qui construit le système, de ce qu'elle se rappelle et du temps dont elle dispose.

Invariants

- Un fichier FileMaker unique est un espace d'exécution partagé.
- Dans un espace d'exécution partagé, le contexte implicite est toujours possible.
- Dans un espace d'exécution partagé, la mutation non contrôlée est toujours possible.
- Dans un espace d'exécution partagé, le couplage implicite est toujours possible.
- Dans un espace d'exécution partagé, l'état ambiant est toujours possible.
- Dans un espace d'exécution partagé, la fuite de logique d'interface est toujours possible.
- Les conventions de noms ne modifient pas la topologie de l'exécution.
- L'organisation en dossiers ne modifie pas la topologie de l'exécution.
- Le regroupement de scripts ne modifie pas la topologie de l'exécution.
- Les jeux de privilèges ne modifient pas la topologie de l'exécution.
- Une frontière de fichier modifie la topologie de l'espace d'exécution.
- La séparation par fichiers isole la portée.
- La séparation par fichiers force les références explicites.
- La séparation par fichiers élimine la co-localisation.
- Les espaces d'exécution séparés ne communiquent qu'à travers des connexions déclarées et visibles.
- La frontière de fichier n'applique pas l'architecture.

Validation

- Vérifier si le contexte, l'état, le graphe d'appels et l'accès en écriture demeurent partagés.
- Vérifier si les contraintes architecturales dépendent de la discipline.
- Vérifier si les violations demeurent disponibles sans action explicite.
- Vérifier si les accès entre frontières exigent des connexions déclarées.
- Vérifier si la portée de l'exécution est isolée par les frontières de fichier.
- Vérifier si les dépendances sont explicites plutôt qu'implicites.
- Vérifier si les interactions entre espaces séparés passent uniquement par des connexions visibles.
- Vérifier si le système rend les violations délibérées plutôt que gratuites.

Conséquences des violations

- Le contexte implicite demeure disponible.
- La mutation non contrôlée demeure disponible.
- Le couplage implicite demeure disponible.
- L'état ambiant demeure disponible.
- La fuite de logique d'interface demeure disponible.
- Les contraintes architecturales demeurent impossibles à appliquer.
- Les violations surviennent sans action explicite.
- Les violations demeurent invisibles jusqu'à ce qu'elles soient déjà intégrées.
- La correction du système dépend de la discipline plutôt que d'un mécanisme.
- Le système demeure non contraint.
- La structure ne peut pas être garantie.
- L'application des contraintes architecturales demeure impossible.