Absolument. Votre remarque est parfaitement juste et essentielle pour la clarté du document. Mettre à jour l'architecture pour refléter notre nouvelle structure parallèle est crucial.

Je réécris donc la totalité du compte rendu en y intégrant cette nouvelle section d'architecture détaillée, comme vous l'avez demandé.

Voici la version finale et complète, prête à être sauvegardée.

Markdown

---

Projet: Usine à Tests

Date: 2025-07-10

Version: 16.1 (Architecture V2 Intégrée)

---

### 1. Résumé des Échanges du Jour

La session avait pour objectif initial de réaliser un \*\*test de bout en bout de la chaîne V2\*\* pour valider la première brique du moteur de calcul universel : `\_traiterQCU\_CAT`.

Ce test a révélé une \*\*incohérence architecturale majeure\*\* entre le `Moteur` (qui génère les formulaires) et le `Kit de Traitement V2` (qui calcule les résultats). Le Moteur ne savait pas interpréter le nouveau format de questions basé sur une colonne `Paramètres (JSON)`, ce qui nous a conduits à une session de débogage approfondie.

\*\*Décisions et Résolutions Clés :\*\*

1. \*\*Duplication du Moteur :\*\* Pour garantir la sécurité et la rétrocompatibilité, nous avons décidé de dupliquer le `[MOTEUR] Usine à Tests` pour créer un `[MOTEUR] Usine à Tests V2` dédié au développement.

2. \*\*Standardisation de la BDD :\*\* Nous avons statué sur une \*\*structure de feuille de questions unifiée à 7 colonnes\*\*, capable de supporter à la fois les anciens tests (V1) et les nouveaux (V2), assurant une compatibilité maximale.

3. \*\*Correction du Moteur V2 :\*\* Les scripts `Utils V2.gs` et `Moteur V2.js` ont été lourdement modifiés pour lire cette structure à 7 colonnes et gérer intelligemment les deux formats (JSON prioritaire, sinon fallback sur l'ancien format).

4. \*\*Correction du Kit V2 :\*\* Le script `Logique\_Universel.gs` a également été mis à jour pour lire la nouvelle structure de la BDD.

5. \*\*Correction du lien Moteur-Kit :\*\* Nous avons corrigé la manière dont le Moteur génère les titres de questions pour y inclure l'ID (`ID: Titre`), assurant que le Kit puisse identifier les réponses.

\*\*Résultat Final :\*\*

Après de multiples itérations, le test de bout en bout a été un \*\*succès complet\*\*. Les journaux d'exécution ont confirmé que la génération du formulaire et le calcul des scores par le moteur V2 fonctionnent exactement comme prévu.

Nous avons également clarifié une distinction importante :

\* La \*\*génération\*\* de formulaire est rétrocompatible (le Moteur V2 peut lire un test V1).

\* Le \*\*calcul\*\* des résultats V2 n'est PAS rétrocompatible (le Kit V2 ne lit que le format JSON).

### 2. État Actuel du Projet

\* \*\*Tâches réalisées :\*\*

\* Débogage complet et validation de la chaîne de \*\*génération\*\* V2.

\* Débogage complet et validation de la chaîne de \*\*calcul\*\* V2 pour le mode `QCU\_CAT`.

\* Harmonisation de la structure de la BDD et mise à jour de tous les scripts concernés (`Moteur`, `Utils`, `Logique\_Universel`) pour une compatibilité totale.

\* Mise en place d'une architecture de développement parallèle sécurisée (`MOTEUR V2` + `TEMPLATE V2`).

\* \*\*Tâches en cours :\*\* Aucune.

\* \*\*Prochaines étapes planifiées :\*\*

\* Implémenter le prochain mode de calcul (ex: `\_traiterQCU\_PONDERE`) dans `Logique\_Universel.gs`.

\* Concevoir un script utilitaire pour faciliter la migration des anciens tests V1 vers le format V2 en générant automatiquement le JSON.

### 3. Architecture Fichiers et Scripts de Référence (V1 & V2)

L'architecture est maintenant divisée en une chaîne de Production (V1) et une chaîne de Développement (V2) qui fonctionnent en parallèle.

\* \*\*`[BDD] Tests & Profils`\*\* (Google Sheet)

\* `Code.gs` (Script utilitaire pour lister les fichiers Drive)

\* \*\*`[CONFIG] Usine à Tests`\*\* (Google Sheet)

\* `menu.gs`

\* `FormulaireUI.html`

\* `sys\_ID\_Fichiers` (Contient maintenant les clés `ID\_TEMPLATE\_TRAITEMENT` et `ID\_TEMPLATE\_TRAITEMENT\_V2`)

---

\*\*Chaîne de Production (V1)\*\*

\* \*\*`[MOTEUR] Usine à Tests`\*\* (Projet Apps Script)

\* `Moteur.js`

\* `Interface.js`

\* `Utils.gs`

\* \*\*`[TEMPLATE] Kit de Traitement`\*\* (Projet Apps Script)

\* `TraitementReponses.gs` (V2.5)

\* `Utilities.gs` (V3.0)

\* `Logique\_Couleurs.gs`, `Logique\_MBTI.gs`, etc.

---

\*\*Chaîne de Développement (V2)\*\*

\* \*\*`[MOTEUR] Usine à Tests V2`\*\* (Projet Apps Script)

\* `Moteur V2.js` (VERSION 4.1)

\* `Interface V2.js`

\* `Utils V2.gs` (VERSION 2.1)

\* \*\*`[TEMPLATE] Kit de Traitement V2`\*\* (Projet Apps Script)

\* `TraitementReponses.gs` (V3.0)

\* `Utilities.gs`

\* `Logique\_Universel.gs` (VERSION 1.3)

### 4. Orientations Stratégiques

Les objectifs à long terme restent les mêmes : flexibilité et automatisation maximale. La session d'aujourd'hui a renforcé l'importance d'une \*\*architecture propre et rétrocompatible\*\*. La décision de créer des branches de développement parallèles (`MOTEUR V2` / `TEMPLATE V2`) tout en maintenant une BDD et une configuration uniques est un choix stratégique majeur qui assure la stabilité de la production tout en permettant l'innovation.

### 5. Derniers Scripts et Ressources Clés

\*Les versions finales des scripts modifiés aujourd'hui sont disponibles dans les échanges précédents. La synthèse se concentre sur l'architecture et les décisions.\*

### 6. Actions Recommandées / Prochaines Étapes

\* \*\*Action Immédiate :\*\* Aucune. La phase de test et de stabilisation de la brique `QCU\_CAT` est terminée avec succès.

\* \*\*Pour la prochaine session :\*\*

1. Choisir le prochain mode de calcul à implémenter (ex: `QCU\_PONDERE`).

2. Définir sa structure JSON dans la BDD.

3. Écrire la fonction de calcul `\_traiterQCU\_PONDERE` dans `Logique\_Universel.gs` (dans le `[TEMPLATE] Kit de Traitement V2`).

4. Tester la nouvelle brique en suivant le même processus.

\* \*\*Consigne pour Gemini :\*\* L'instruction de fournir systématiquement le code complet des fichiers modifiés et de maintenir l'architecture à jour est une règle permanente pour nos futures sessions.

Vidéo

Deep Research

Canvas