

Compte rendu du 26/01/2026

MATIN ET APRES-MIDI

ESP : Rédaction des documents guide technique et document d'assurance qualité pour la partie connecteurs. Les documents ne sont pas terminés mais ils le seront avant la fin de la semaine.

Organisation du travail à faire avec Erlan côté connecteurs, répartition des tâches.

Plan d'Assurance Qualité – Couche Connecteurs Projet EXO NATIVE

1. Objectifs qualité

Garantir la fiabilité, la robustesse et la traçabilité des connecteurs.

Réduire les erreurs humaines par des validations bloquantes (Poka-Yoke).

2. Principes qualité

Validation systématique avant exécution.

Traçabilité complète des actions.

Comportement déterministe et reproductible.

3. Critères d'acceptation généraux

Le connecteur doit établir une connexion valide avec l'outil cible.

Toute erreur bloquante empêche la poursuite du workflow.

Un retour utilisateur explicite est systématiquement fourni.

4. Stratégie de tests

Tests unitaires des méthodes IConnector.

Tests d'intégration avec outils cibles (Blender, Unreal, GitLab, Notion).

Tests de non-régression à chaque évolution.

5. Validation spécifique Blender

Naming conforme aux conventions définies.

Scale appliqué à (1,1,1).

Textures présentes et accessibles.

Topologie analysée avec remontée de warnings.

6. Validation spécifique Unreal

Import sans erreur dans le moteur.

Cohérence des métadonnées.

Guide Technique – Couche Connecteurs

Projet EXO NATIVE

1. Périmètre du document

Ce document décrit l'architecture, le fonctionnement et les responsabilités de la couche Connecteurs du projet EXO NATIVE.

Le périmètre couvre les connecteurs Blender, Unreal Engine, GitLab et Notion, à l'exclusion du backend, de l'infrastructure et de la couche IA.

2. Rôle de la couche Connecteurs

La couche Connecteurs assure l'interface entre les outils externes de production et le Hub EXO MCP.

Elle garantit la standardisation des échanges, l'automatisation des actions techniques et la fiabilité des workflows déclenchés depuis le Hub.

3. Architecture générale des connecteurs

Chaque connecteur implémente une interface standardisée IConnector.

Les connecteurs sont des modules Python indépendants communiquant avec le Hub via HTTP REST ou socket TCP selon l'outil cible.

4. Interface IConnector

Méthodes obligatoires : `connect()`, `disconnect()`, `healthcheck()`, `execute(command, params)`, `get_capabilities()`.

Cette interface garantit l'interchangeabilité et la maintenabilité des connecteurs.

5. Connecteur Blender

Implémenté sous forme d'addon Python Blender.

Communication via JSON-RPC 2.0 sur socket TCP.

Réalise les validations pré-vol (naming, scale, textures, topologie) et l'export des assets.

6. Connecteur Unreal Engine

Basé sur le plugin Unreal Remote Control API.

Permet l'import automatique des assets, la génération de LODs et l'exécution de scripts Python.