Oxide

Cahier des charges S4 2026

Alexandre Joaquim Lima Salgueiro
alexandre-joaquim.lima-salgueiro

Willian Huang Hong
william.huang-hong

Romain Bailly
romain.bailly

Tom Huynh
tom.huynh

January 25, 2026

Contents

1 Introduction **2**

2 État de l'art **2**
2.1 7-Zip (Format .7z) . . . 2
2.2 WinRAR (Format .rar) . . . 2
2.3 WinZip (Format .zip et .zipx) . . . 2
2.4 Zstandard (Format .zst) . . . 3

3 Stratégie d'Oxide **3**

4 Fonctionnalités et Algorithmes **3**
4.1 Algorithmes de Pré-traitement . . . 3
4.2 Compression et Codage Entropique . . . 4
4.3 Autres Fonctionnalités . . . 4

5 Répartition des tâches **4**
5.1 Soutenance 1 (23 - 27 Mars 2026) . . . 5
5.2 Soutenance 2 (18 - 22 Mai 2026) . . . 5
5.3 Descriptif des tâches spécifiques . . . 5

6 Conclusion **6**

1 Introduction

Oxide est un projet de développement d'un logiciel d'archivage de fichiers, conçu pour effectuer des opérations de compression et de décompression de manière efficace et sécurisée. S'inspirant de solutions reconnues telles que 7-Zip et WinRAR, Oxide a pour ambition de fournir une alternative moderne, tirant parti des performances et de la sûreté de mémoire offertes par le langage Rust.

L'objectif principal de ce projet est d'implémenter des algorithmes de compression robustes tout en offrant une interface utilisateur simple et performante. Oxide permettra aux utilisateurs de réduire la taille de leurs fichiers pour faciliter le stockage et le partage, tout en garantissant l'intégrité des données lors de l'extraction.

2 État de l'art

2.1 7-Zip (Format .7z)

- **Extension utilisée :** `.7z`
- **Algorithmes et Pipeline :** Utilise principalement LZMA et LZMA2. Le pipeline inclut des filtres de pré-traitement (BCJ/BCJ2 pour les exécutables) suivis d'une compression par dictionnaire et d'un codage par intervalle (Range Coder).
- **Capacité :** Taux de compression très élevé, souvent considéré comme la référence pour l'archivage généraliste open-source. Supporte des fichiers jusqu'à 16 exabytes.
- **Limites :** La vitesse de compression est relativement lente en mode "Ultra" et l'extraction nécessite plus de ressources processeur que les formats plus anciens.

2.2 WinRAR (Format .rar)

- **Extension utilisée :** `.rar`
- **Algorithmes et Pipeline :** Utilise une variante propriétaire de LZSS avec prédiction par contexte (PPM) pour le texte et des filtres spécialisés pour le multimédia et les exécutables.
- **Capacité :** Excellent équilibre entre vitesse et taux de compression. Sa fonctionnalité clé est l'enregistrement de restauration, qui permet de réparer une archive physiquement endommagée.
- **Limites :** Format propriétaire et fermé. La création d'archives .rar nécessite une licence payante.

2.3 WinZip (Format .zip et .zipx)

- **Extension utilisée :** `.zip`, `.zipx`
- **Algorithmes et Pipeline :** Le `.zip` classique utilise DEFLATE. Le format moderne `.zipx` utilise une approche "content-aware" : LZMA pour les données génériques, Jpeg pour les images, WavPack pour l'audio et PPMd pour le texte.
- **Capacité :** Le standard absolu pour la compatibilité (ouvert nativement par tous les systèmes d'exploitation). Le `.zipx` offre une compression compétitive face au 7z.
- **Limites :** Le format `.zip` classique compresse peu. Le format `.zipx` souffre de problèmes de fragmentation : il est difficile de trouver un décompresseur compatible hors de l'écosystème WinZip.

2.4 Zstandard (Format .zst)

- **Extension utilisée :** `.zst` (Développé par Facebook/Meta)
- **Algorithmes et Pipeline :** Basé sur un codage entropique à états finis (tANS) et LZ77. Conçu pour scaler massivement.
- **Capacité :** Vitesse de décompression extrêmement rapide tout en maintenant un taux de compression comparable à zlib ou LZMA selon le niveau réglé. Devenu le standard pour le Big Data, Linux et le Gaming.
- **Limites :** Moins orienté "archivage utilisateur" (pas de chiffrement natif ou de gestion de fichiers multiples sans l'aide du format `.tar`).

3 Stratégie d'Oxide

Oxide adopte une stratégie "content-aware" (sensible au contenu) inspirée de WinZip, mais structurée autour d'un pipeline en deux étapes : Pré-traitement → Compression. L'objectif n'est pas seulement de changer d'algorithme de compression, mais d'appliquer des transformations intelligentes en amont pour faciliter le travail du compresseur final.

Concrètement, Oxide analysera les fichiers en entrée pour déterminer la chaîne de traitement optimale :

- Pour les types de fichiers spécifiques (images, audio, texte, exécutables), des filtres de pré-traitement dédiés sont appliqués. Ces transformations réversibles visent à réduire l'entropie ou à exposer la redondance cachée des données.
- Le flux de données résultant (ou les données brutes pour les types génériques) est ensuite confié à un algorithme de compression universel.

Cette architecture hybride permet de combiner la puissance des modèles spécialisés (pour la préparation des données) avec l'efficacité d'un compresseur généraliste robuste.

4 Fonctionnalités et Algorithmes

Oxide met en œuvre une chaîne de traitement en deux étapes : le pré-traitement (Pre-processing) et la compression (Compression). Le pré-traitement vise à transformer les données pour les rendre plus compressibles par l'algorithme final.

4.1 Algorithmes de Pré-traitement

Ces algorithmes ne compressent pas nécessairement les données (voire peuvent augmenter légèrement leur taille), mais réorganisent l'information pour réduire l'entropie ou exposer des motifs répétitifs.

- **Images :**
  - *YCoCg-R :* Transformation d'espace colorimétrique réversible qui décorrèle les canaux RVB en luminance (Y) et chrominance (Co, Cg), facilitant la compression ultérieure.
  - *PAETH :* Filtre prédictif 2D qui réduit l'entropie en stockant la différence entre la valeur d'un pixel et une prédiction basée sur ses voisins.
  - *LOCO-I / MED :* Détection de contours (Median Edge Detection) pour choisir le meilleur prédicteur local, utilisé pour préparer les données d'image.

- **Texte :**
  - *BWT (Burrows-Wheeler Transform) :* Réorganise les caractères pour regrouper les occurrences identiques, créant de longues suites de caractères identiques idéales pour un codeur RLE ou entropique.
  - *BPE (Byte Pair Encoding) :* Remplace les paires d'octets fréquentes par des symboles uniques, réduisant la redondance avant la compression finale.
- **Audio :**
  - *LPC (Linear Predictive Coding) :* Modélise l'onde sonore et ne conserve que le résidu (l'erreur de prédiction), qui est beaucoup plus facile à compresser que le signal brut.
- **Fichiers Binaires (Exécutables) :**
  - *BCJ (Branch / Call / Jump) :* Normalise les instructions de saut dans le code machine (x86/ARM) en convertissant les adresses relatives en absolues, ce qui augmente les répétitions pour le compresseur.

4.2 Compression et Codage Entropique

Après l'étape de pré-traitement, les données transformées passent par l'étage final de compression. Le choix de l'algorithme universel dépend du mode sélectionné par l'utilisateur :

- **Mode Rapide :** Utilise *LZ4*, un algorithme axé sur une vitesse de compression et de décompression extrêmement élevée, idéal pour les systèmes temps réel ou le stockage rapide.
- **Mode Équilibré :** Combine *LZ77* et le *codage de Huffman* (similaire à DEFLATE). Ce mode offre un excellent compromis entre ratio de compression et vitesse, adapté à un usage général.
- **Mode Ultra :** Utilise *LZMA* (Lempel-Ziv-Markov chain Algorithm) pour les données génériques et *PPM* (Prediction by Partial Matching) pour le texte. Ce mode privilégie un taux de compression maximal au prix d'un temps de traitement plus long et d'une utilisation mémoire accrue.

4.3 Autres Fonctionnalités

- **Chiffrement :** Protection des archives via un système de chiffrement.
- **Vérification d'intégrité :** Mécanisme de détection de corruption via hashage pour s'assurer que les fichiers extraits sont strictement identiques aux originaux.
- **Multithreading :** Utilisation de la parallélisation pour accélérer les opérations de compression et de décompression en traitant plusieurs fichiers simultanément.
- **Modes de Compression Configurables :** Support de plusieurs niveaux de compression (ex: Rapide, Équilibré, Ultra) permettant à l'utilisateur de choisir le compromis idéal entre la vitesse d'exécution et la réduction de taille.
- **Interface Graphique :** Oxide sera accompagné d'une application graphique intuitive permettant la manipulation simplifiée des archives.

5 Répartition des tâches

La répartition des fonctionnalités est divisée en deux phases correspondant aux deux soutenances.

5.1 Soutenance 1 (23 - 27 Mars 2026)

Fonctionnalité	Alexandre	Willian	Romain	Tom
Compression Universelle				
LZ4 (Mode Rapide)				X
LZ77 + Huffman (Mode Équilibré)				X
Pré-traitement Image				
YCoCg-R	X			
LOCO-I / MED			X	
Pré-traitement Texte				
BPE	X			
Pré-traitement Binaire				
BCJ		X		
Intégrité				
Hashing			X	
Architecture et Interface				
CLI		X		
Format de fichier				X
Pipeline Manager (Routing)				X
Communication				
Site Web			X	

5.2 Soutenance 2 (18 - 22 Mai 2026)

Fonctionnalité	Alexandre	Willian	Romain	Tom
Compression Avancée (Mode Ultra)				
LZMA				X
Pré-traitement Audio				
LPC		X		
Pré-traitement Texte Avancé				
BWT	X			
PPM				X
Sécurité				
Chiffrement	X			
Application et Communication				
Interface Graphique (GUI)			X	
Site Web			X	

5.3 Descriptif des tâches spécifiques

Voici une description des éléments de la matrice n'ayant pas fait l'objet d'un descriptif détaillé dans les sections précédentes :

- **Format de fichier :** Définition et implémentation de la structure binaire du conteneur (magic number, en-têtes de fichiers, blocs de données compressées, métadonnées d'intégrité et index final).
- **Pipeline Manager (Routing) :** Développement de la logique d'analyse des fichiers en entrée pour aiguiller automatiquement les données vers les filtres de pré-traitement adéquats avant la compression finale.

6 Conclusion

Oxide est un projet qui vise à explorer en profondeur les mécanismes de compression modernes en tirant parti des atouts du langage Rust. En combinant une approche "content-aware" inspirée de l'état de l'art avec une architecture performante et sécurisée, l'objectif est de livrer un outil d'archivage performant, modulaire et facile d'utilisation. Ce projet permettra la maîtrise d'algorithmes complexes tout en assurant une expérience utilisateur finale soignée.