LIVRE BLANC

NumFranc (NFR)

Une monnaie numérique pair-à-pair conçue dès son origine pour résister aux ordinateurs quantiques

Version 1.1 — octobre 2026

ShadowRook

Clé publique de signature des versions (SLH-DSA-SHAKE-128s)

5e1d1c80fdd07d3bc4095f45a05649fc006bce347609b1566b3c25e940881736

Sommaire

Résumé

1. Pourquoi NumFranc

2. Vue d'ensemble

3. Cryptographie

4. Émission monétaire

5. Blocs et preuve de travail

6. Transactions

7. Consensus et sécurité de la chaîne

8. Réseau

9. Portefeuille

10. Application et distribution

11. Signature des versions

12. Feuille de route

13. Conclusion

Références

Annexe A — Paramètres de consensus et de réseau

Annexe B — Calculs de la preuve de travail et de la difficulté

Annexe C — Format binaire des transactions

Annexe D — Formats de texte

Annexe E — Calendrier d'émission

Annexe F — Messages réseau

Annexe G — Glossaire

Résumé

NumFranc (symbole **NFR**) est une monnaie numérique pair-à-pair, sans banque, sans serveur central et sans émetteur. Elle reprend les principes qui ont fait leurs preuves avec Bitcoin : un registre public de blocs chaînés, des pièces représentées par des sorties de transaction non dépensées, et une preuve de travail qui rend toute falsification du passé coûteuse.

Sa différence tient à un choix fait **dès le premier bloc** : toute la cryptographie de NumFranc est post-quantique. Les transactions sont signées avec ML-DSA-87 (norme NIST FIPS 204, niveau de sécurité le plus élevé), toutes les empreintes utilisent SHA3-512, et les versions du logiciel sont signées avec SLH-DSA (FIPS 205), dont la sécurité ne repose que sur des fonctions de hachage. Un ordinateur quantique capable de casser les signatures de Bitcoin ne permettrait pas de voler des NFR.

La quantité totale est fixée à **69 100 000 NFR**, émis sur environ 98 ans au rythme d'un bloc toutes les **691 secondes**. Le minage utilise Argon2d, une fonction gourmande en mémoire qui laisse un ordinateur ordinaire compétitif. NumFranc ajoute enfin des fonctions pensées pour un usage quotidien : adresses lisibles avec somme de contrôle, messages joints aux paiements, paiements datés ou annulables, multi-signature, pseudos (« alice.nfr ») et portefeuille protégé par deux mots de passe et un code de secours en trois morceaux.

Ce document décrit NumFranc tel qu'il est implémenté dans la version 1.0 du logiciel. Les fonctions seulement prévues sont présentées à part, dans la feuille de route.

1. Pourquoi NumFranc

1.1 La menace quantique

Bitcoin et la plupart des cryptomonnaies signent leurs transactions avec des courbes elliptiques (ECDSA ou Schnorr sur secp256k1). Leur sécurité repose sur la difficulté du logarithme discret. En 1994, Peter Shor a montré qu'un ordinateur quantique suffisamment grand résoudrait ce problème en un temps raisonnable. Un tel ordinateur pourrait retrouver une clé secrète à partir de la clé publique, puis dépenser les fonds à la place de leur propriétaire.

Ces machines n'existent pas encore à l'échelle nécessaire. Mais une monnaie doit protéger des fonds pendant des décennies, et une clé publique publiée aujourd'hui dans un registre permanent pourra être attaquée demain. Migrer une monnaie existante vers d'autres signatures est difficile : il faut l'accord de tout le réseau, et les fonds dont les propriétaires ont disparu restent exposés.

1.2 Les nouvelles normes

En août 2024, le NIST (l'institut américain des normes) a publié les premières normes de cryptographie post-quantique, après huit ans d'évaluation publique :

- **FIPS 204 — ML-DSA** : signatures fondées sur les réseaux euclidiens (problème Module-LWE) ;
- **FIPS 205 — SLH-DSA** : signatures fondées uniquement sur des fonctions de hachage.

Aucun algorithme quantique connu ne casse ces problèmes plus vite que par une recherche exhaustive, à peine accélérée.

1.3 Le choix de NumFranc

NumFranc n'a pas de passé à migrer : il naît post-quantique. Ses objectifs, par ordre de priorité, sont :

1. **résister aux ordinateurs quantiques**, pour toutes les signatures et toutes les empreintes ;
2. **résister à l'attaque des 51 %** et aux manipulations de la chaîne ;
3. **résister à la falsification du logiciel**, grâce aux versions signées ;
4. **rester simple à utiliser** : un seul programme, une fenêtre, en français ou en anglais.

2. Vue d'ensemble

NumFranc est un réseau de nœuds égaux. Chaque nœud conserve la chaîne de blocs complète, vérifie chaque bloc et chaque transaction selon les mêmes règles, et relaie ce qu'il reçoit. Aucun nœud n'a d'autorité particulière.

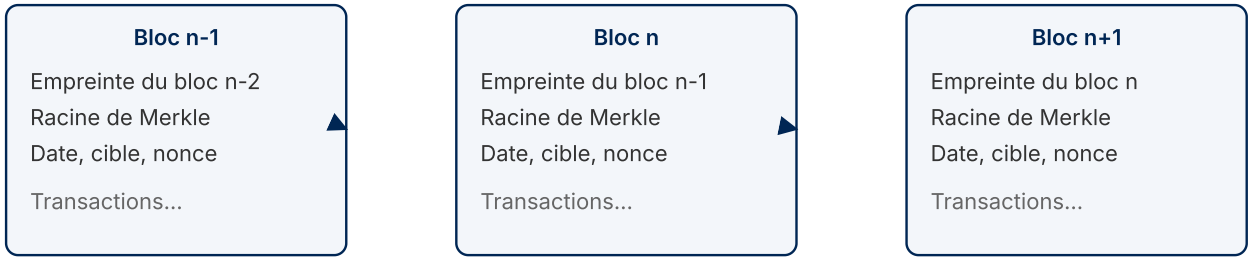


Figure 1. Chaque en-tête contient l'empreinte du bloc précédent : modifier un bloc ancien change toutes les empreintes suivantes.

Les principaux paramètres sont résumés ci-dessous. La liste complète figure en annexe A.

Paramètre	Valeur
Symbole	NFR
Quantité totale	69 100 000 NFR
Divisibilité	1 NFR = 100 000 000 unités (8 décimales)
Temps cible entre deux blocs	691 secondes (11 min 31 s)
Ajustement de la difficulté	tous les 691 blocs
Preuve de travail	Argon2d, 4 Mio de mémoire par essai
Signatures des transactions	ML-DSA-87 (FIPS 204)
Empreintes	SHA3-512 (FIPS 202)
Signatures des versions	SLH-DSA-SHAKE-128s (FIPS 205)
Taille maximale d'un bloc	8 Mio

Paramètre	Valeur
Port réseau	6910 (TCP et UDP)
Bloc genesis	17 octobre 2026, 00:00:00 UTC

3. Cryptographie

NumFranc n'invente aucune primitive cryptographique. Il assemble des algorithmes normalisés, implémentés par des bibliothèques Rust publiques et testées. Le seul calcul écrit pour le projet est le partage de secret de Shamir, dont l'arithmétique est vérifiée par des tests.

3.1 Empreintes : SHA3-512

Toutes les empreintes de NumFranc utilisent **SHA3-512** (FIPS 202), de la famille Keccak, dont la construction « éponge » est différente de celle de SHA-2. Les identifiants de blocs et de transactions sont des **doubles empreintes**, notées `H(x) = SHA3-512(SHA3-512(x))`.

Face à un ordinateur quantique, l'algorithme de Grover accélère la recherche d'une préimage, mais seulement en racine carrée. Avec 512 bits de sortie, la sécurité restante est d'environ 256 bits, très au-delà de ce qui est atteignable.

Chaque usage d'une empreinte est précédé d'une étiquette propre (« séparation de domaine »), par exemple `NFR-SIGHASH`, `NFR-MULTISIG` ou `NFR-wallet-key`. Une même donnée ne peut donc pas être réinterprétée dans un autre contexte.

3.2 Signatures des transactions : ML-DSA-87

Chaque dépense est autorisée par une signature **ML-DSA-87**, le jeu de paramètres le plus fort de FIPS 204 (catégorie 5 du NIST, comparable à AES-256). Les signatures utilisent le contexte `NFR-tx-v1`, ce qui les rend inutilisables en dehors de NumFranc.

	ML-DSA-87	ECDSA (Bitcoin)
Clé publique	2 592 octets	33 octets
Signature	4 627 octets	71 à 73 octets
Problème mathématique	réseaux euclidiens	logarithme discret
Attaque quantique connue	aucune efficace	algorithme de Shor

Ce choix a un coût : une transaction NumFranc est environ trente fois plus volumineuse qu'une transaction Bitcoin. NumFranc l'accepte et le compense par des blocs plus grands (section 5.4).

ML-DSA est un schéma **sans état** : signer plusieurs fois avec la même clé ne l'affaiblit pas. Une adresse peut donc être réutilisée sans risque cryptographique.

3.3 Signatures des versions : SLH-DSA

Les versions publiées du logiciel sont signées avec **SLH-DSA-SHAKE-128s** (FIPS 205). Sa sécurité repose uniquement sur des fonctions de hachage, l'hypothèse la plus étudiée et la plus prudente de la cryptographie. La signature est volumineuse (7 856 octets) et lente à produire, mais elle n'est calculée qu'une fois par version. La section 11 décrit son utilisation.

3.4 Preuve de travail : Argon2d

La preuve de travail utilise **Argon2d** (RFC 9106), vainqueur de la Password Hashing Competition, avec 4 Mio de mémoire, une passe et une sortie de 64 octets. Chaque essai oblige à remplir puis relire 4 Mio de mémoire dans un ordre imprévisible. Les puces spécialisées (ASIC) et les cartes graphiques perdent ainsi l'essentiel de leur avance sur un processeur ordinaire, ce qui garde le minage ouvert à tous.

3.5 Protection du portefeuille

Le fichier du portefeuille est protégé par une dérivation de clé **Argon2id**, lente et gourmande en mémoire, puis par **deux chiffrements authentifiés successifs** : AES-256-GCM, puis XChaCha20-Poly1305 sur le résultat, avec deux clés indépendantes de 256 bits. Si l'un des deux algorithmes était un jour cassé, l'autre protégerait encore les données. Le détail figure à la section 9.

3.6 Résumé de la résistance quantique

Élément	Algorithme	Effet d'un ordinateur quantique
Signatures des transactions	ML-DSA-87	aucune attaque efficace connue
Signatures des versions	SLH-DSA-SHAKE-128s	aucune attaque efficace connue
Empreintes, Merkle, adresses	SHA3-512	Grover : sécurité divisée par deux en bits, encore ≥ 128 bits
Chiffrement du portefeuille	AES-256 et XChaCha20	Grover : environ 128 bits restants

« Aucune attaque connue » ne signifie pas « prouvé incassable ». C'est pourquoi NumFranc utilise les jeux de paramètres les plus forts disponibles, et une signature fondée sur le seul hachage pour l'élément le plus critique : l'authenticité du logiciel.

4. Émission monétaire

4.1 La quantité totale

Il n'existera jamais plus de **69 100 000 NFR**. Ce nombre est inspiré de la population de la France au 1er janvier 2026, estimée par l'INSEE à environ 69,1 millions d'habitants : symboliquement, un NFR par habitant. Le nombre 691 revient aussi dans le temps entre deux blocs (691 secondes) et dans la fenêtre d'ajustement de la difficulté (691 blocs).

Tous les montants sont des **entiers** exprimés en unités minimales (1 NFR = 10^8 unités). Aucun calcul de consensus n'utilise de nombres à virgule flottante : chaque nœud obtient exactement le même résultat, sur n'importe quel ordinateur.

4.2 Le calendrier par paliers

L'émission est découpée en **paliers de 100 000 blocs**, soit 69,1 millions de secondes : environ **2 ans et 2 mois** par palier.

- **Palier 0 (démarrage lent)** : 6,91 NFR par bloc. Ce palier laisse au réseau le temps de se former avant que l'essentiel des pièces soit émis.
- **Palier 1 (pic)** : 69,10 NFR par bloc.
- **Paliers suivants** : la récompense baisse de **10 %** à chaque palier, en division entière (`r ← r × 9 / 10`).

Le dernier bloc récompensé est le bloc **4 471 948**, qui reçoit exactement le reliquat (0,44971536 NFR). L'émission se termine environ **98 ans** après le genesis.

Palier	Blocs	Récompense par bloc	Émis à la fin du palier	Années écoulées (environ)
0	0 – 99 999	6,91 NFR	691 000	2,2
1	100 000 – 199 999	69,10 NFR	7 601 000	4,4
2	200 000 – 299 999	62,19 NFR	13 820 000	6,6
3	300 000 – 399 999	55,971 NFR	19 417 100	8,8
5	500 000 – 599 999	45,33651 NFR	28 988 141	13,1
10	1 000 000 – 1 099 999	26,77075578 NFR	45 697 319,79	24,1
44	4 400 000 – 4 471 948	0,74457068 NFR, puis reliquat	69 100 000	97,9

L'annexe E donne le calendrier détaillé. Quelques repères :

Part de la quantité totale émise	Atteinte vers le bloc	Années après le genesis
25 %	361 728	7,9
50 %	740 345	16,2
75 %	1 379 416	30,2
90 %	2 195 221	48,1
99 %	3 813 590	83,5

4.3 La récompense de minage

Le mineur d'un bloc a droit à la récompense de ce bloc **plus** les frais des transactions qu'il inclut. La transaction de récompense (« coinbase ») ne peut pas dépasser ce total. Un mineur qui réclame moins perd la différence : ces unités ne sont jamais créées. La récompense ne peut être dépensée

qu'après **100 confirmations**.

La récompense du bloc genesis (6,91 NFR) est attribuée à une adresse nulle et n'est pas dépensable, comme celle du bloc genesis de Bitcoin.

4.4 Distribution

Il n'y a **ni prévente, ni réserve des créateurs, ni pièces créées hors minage**. Chaque NFR est émis par un bloc miné publiquement, selon le calendrier ci-dessus. Le palier 0, à faible récompense, limite la part des tout premiers mineurs, y compris celle de l'auteur du projet.

5. Blocs et preuve de travail

5.1 L'en-tête de bloc

Chaque bloc commence par un en-tête de taille fixe, **212 octets**, écrit en ordre big-endian :

Champ	Taille	Rôle
Version	4 octets	version du protocole (actuellement 4)
Bloc précédent	64 octets	empreinte de l'en-tête du bloc précédent
Racine de Merkle	64 octets	résumé de toutes les transactions du bloc
Date	8 octets	heure Unix, en secondes
Cible	64 octets	seuil de difficulté (entier de 512 bits)
Nonce	8 octets	compteur modifié par le mineur

L'**empreinte du bloc** est `H(en-tête)`, la double SHA3-512. La **preuve de travail** est un calcul distinct : `Argon2d(en-tête)`, avec le sel fixe `NumFranc-PoW-v1!`. Un bloc est valide si cette valeur, lue comme un entier de 512 bits, est **inférieure à la cible**.

Les transactions sont résumées par un **arbre de Merkle** avec séparation de domaine : chaque feuille est `H(0x00 || txid)`, chaque nœud `H(0x01 || gauche || droite)`. Cette distinction empêche de faire passer un nœud interne pour une transaction.

5.2 Le bloc genesis

Le premier bloc est inscrit dans le logiciel. Il est daté du **17 octobre 2026 à 00:00:00 UTC** et porte le message `NumFranc - 69,1 millions - 17/10/2026`. Son nonce est 452 924 et son empreinte commence par `7e0e17a3e283a029`. Sa cible, $2^{492} - 1$, demande en moyenne environ un million d'essais Argon2 : quelques minutes pour un ordinateur de bureau.

5.3 L'ajustement de la difficulté

Tous les **691 blocs**, la cible est recalculée à partir du temps réellement écoulé :

```
nouvelle cible = ancienne cible × temps mesuré / temps attendu
```

Le temps attendu correspond aux 690 intervalles de la fenêtre : 690 × 691 = 476 790 secondes. Le temps mesuré est borné entre le quart et le quadruple de cette valeur, pour qu'un seul ajustement ne puisse pas changer la difficulté de plus d'un facteur 4. Tout le calcul est fait en entiers de grande taille, sans arrondi flottant.

Depuis la version 1.0, une règle reprise de Bitcoin (BIP94) interdit l'attaque dite du « time warp » : le premier bloc d'une fenêtre ne peut pas être daté plus de 2 heures avant le bloc précédent. Sans elle, un mineur majoritaire pourrait dater ce bloc dans le passé pour fausser la mesure et faire baisser la difficulté. Comme les autres règles ajoutées en 1.0, elle s'applique aux blocs dont le temps médian passé dépasse le 19 octobre 2026 : les blocs plus anciens restent valides.

5.4 Taille des blocs et débit

Un bloc peut contenir jusqu'à **8 Mio**. Une transaction simple, avec une entrée et deux sorties, occupe environ 7,4 ko, à cause de la taille des clés et signatures ML-DSA. Un bloc peut donc contenir environ 1 100 transactions de ce type, soit un débit d'environ **1,6 transaction par seconde**. C'est inférieur à Bitcoin : c'est le prix de la sécurité post-quantique. L'amélioration du débit fait partie des pistes de recherche (section 12).

5.5 Le travail cumulé

Le travail d'un bloc vaut `2²⁵⁶ / (cible + 1)`. La chaîne retenue par un nœud est celle qui a le plus grand **travail cumulé**, et non la plus longue. Un attaquant ne peut donc pas imposer une chaîne plus longue mais minée avec une difficulté artificiellement basse.

6. Transactions

6.1 Le modèle des sorties non dépensées

Comme dans Bitcoin, les pièces sont des **sorties de transaction non dépensées**. Une transaction consomme des sorties existantes (ses entrées) et en crée de nouvelles (ses sorties). La somme des entrées doit être strictement supérieure à la somme des sorties : la différence constitue les frais, versés au mineur.

NumFranc n'a **pas de langage de script**. Toutes les règles de dépense sont décrites par des champs fixes et vérifiées par un code simple. Cela réduit fortement la surface d'erreur, au prix d'une flexibilité moindre que celle de Bitcoin.

6.2 Structure

Élément	Contenu
Version, verrou global	version de transaction (4) ; le verrou global n'est pas utilisé par les règles actuelles
Entrées	sortie dépensée (txid, index), puis soit clé publique et signature ML-DSA, soit une preuve multi-signature
Sorties	montant, adresse, verrou de déblocage, adresse d'annulation facultative
Message joint	texte UTF-8 de 140 octets au plus

Élément	Contenu
Pseudo	enregistrement ou transfert facultatif d'un pseudo

L'identifiant d'une transaction est `H(transaction complète)`. La signature porte sur une empreinte `H("NFR-SIGHASH" ‖ transaction)` dans laquelle les clés et signatures sont remplacées par des zéros. Le message joint, les verrous et le pseudo sont donc signés avec le reste : personne ne peut les modifier en chemin. Le format octet par octet est donné en annexe C.

6.3 Les adresses

Une adresse est formée des **256 premiers bits de SHA3-512(clé publique)**. Elle est affichée sous une forme lisible :

`NFR1-XXXX-XXXX-…`

Le corps encode, en base 32 de Crockford, les 32 octets de l'adresse suivis de 4 octets de somme de contrôle (58 caractères en groupes de 4). Cet alphabet exclut I, L, O et U ; à la lecture, I et L sont compris comme 1, et O comme 0. Une faute de frappe est détectée par la somme de contrôle avant tout envoi.

6.4 Les frais

Les frais minimaux sont de **1 000 unités par kilo-octet entamé**. Une transaction simple de 7,4 ko coûte donc au moins 8 000 unités, soit 0,00008 NFR. Les mineurs choisissent en priorité les transactions qui paient le plus par octet.

6.5 Les paiements datés

Chaque sortie peut porter un **verrou de déblocage**. S'il vaut 0, la sortie est dépensable tout de suite. S'il est inférieur à 500 000 000, c'est une hauteur de bloc. Sinon, c'est une date Unix, comparée au **temps médian passé** des 11 derniers blocs, comme dans le BIP 113 de Bitcoin. Un mineur ne peut donc pas avancer l'heure pour débloquer des fonds plus tôt.

Usage typique : de l'argent de poche versé maintenant mais disponible seulement le mois prochain.

6.6 Les paiements retenus (annulables)

Une sortie verrouillée peut aussi désigner une **adresse d'annulation**. Jusqu'à la date de déblocage, seule cette adresse peut reprendre les fonds. Après, seul le destinataire peut les dépenser. C'est la base de la **protection des gros paiements** du portefeuille (section 9.5) : en cas d'erreur ou de vol, l'expéditeur garde un délai pour annuler.

6.7 La multi-signature

Une adresse peut appartenir à un **groupe de 2 à 15 membres**, dont au moins **M** doivent signer pour dépenser. L'adresse du groupe est l'empreinte du seuil et de la liste triée des adresses des membres. Le seuil et la liste des membres sont couverts par les signatures ; les signatures elles-mêmes ne le sont pas, ce qui permet aux membres de signer chacun leur tour.

6.8 Les pseudos

Une transaction peut **enregistrer un pseudo**, par exemple `alice.nfr`, associé à une adresse. On peut ensuite payer ce pseudo au lieu de recopier une adresse. Un pseudo compte de 3 à 32 caractères parmi `a-z`, `0-9` et le tiret (jamais en début, en fin, ni doublé). Seul le propriétaire actuel peut le transférer.

Pour éviter l'accaparement des noms courts, l'enregistrement coûte des frais supplémentaires, versés au mineur :

Longueur du pseudo	Frais supplémentaires
3 caractères	100 NFR
4 caractères	25 NFR
5 à 7 caractères	5 NFR
8 caractères et plus	1 NFR

7. Consensus et sécurité de la chaîne

7.1 Les règles de validation

Chaque nœud vérifie lui-même chaque bloc reçu. Un bloc est refusé si l'une des règles suivantes n'est pas respectée :

- version du protocole correcte, et conformité aux points de contrôle inscrits dans le logiciel ;
- cible égale à celle qu'impose l'ajustement de la difficulté ;
- date strictement postérieure au temps médian des 11 blocs précédents, et au plus **2 heures** dans le futur ;
- preuve de travail Argon2d inférieure à la cible ;
- taille d'au plus 8 Mio ;
- première transaction et seulement elle de type coinbase, contenant la hauteur du bloc ;
- racine de Merkle exacte, et aucune transaction en double ;
- chaque transaction valide : signatures ML-DSA-87, entrées existantes et non dépensées, verrous respectés, frais suffisants, pseudos valides ;
- coinbase ne dépassant pas la récompense plus les frais.

Les vérifications bon marché sont faites en premier ; la preuve de travail (environ 2 ms) et les signatures viennent ensuite. Un bloc refusé n'est jamais relayé aux autres nœuds.

7.2 Les protections contre l'attaque des 51 %

Un attaquant qui contrôle la majorité de la puissance de minage pourrait tenter de réécrire l'historique récent, par exemple pour annuler un paiement qu'il a fait. NumFranc combine cinq protections :

1. **Travail cumulé** : seule compte la chaîne qui a demandé le plus de calcul.

2. **Limite de réorganisation** : un nœud refuse de remplacer plus de **10 blocs** de sa chaîne active (environ 1 h 55). Au-delà, l'historique est considéré comme définitif.
3. **Points de contrôle** : les empreintes de blocs connus pourront être inscrites dans chaque nouvelle version du logiciel. Cette liste est vide dans la version 1.0.
4. **Maturité des récompenses** : 100 confirmations avant de pouvoir dépenser une récompense de minage.
5. **Règles d'horodatage** : le temps médian passé, la limite de 2 heures et la règle anti « time warp » (section 5.3) empêchent de manipuler la difficulté en trichant sur les dates.
6. **Branches mortes refusées** : un bloc d'une branche qui part de plus de 10 blocs en arrière, et qui ne pourrait donc jamais être adoptée, n'est même pas enregistré.

La limite de réorganisation a une contrepartie, qu'il faut connaître : si le réseau était coupé en deux pendant plus de 10 blocs, les deux moitiés ne se réconcilieraient pas automatiquement. Une intervention manuelle serait alors nécessaire. NumFranc considère ce risque comme préférable à celui d'une réécriture profonde de l'historique.

Pour un paiement important, il est conseillé d'attendre **6 confirmations**, soit environ 70 minutes.

8. Réseau

8.1 Le protocole

Les nœuds communiquent en TCP sur le **port 6910**. Chaque message est encadré ainsi :

`identifiant « NFR\x04 » (4 octets) | commande (12) | longueur (4) | somme de contrôle (4) | contenu`

La somme de contrôle correspond aux 4 premiers octets de `H(contenu)`. Un contenu ne peut pas dépasser 8 Mio + 1 Kio. Les commandes sont : `version`, `verack`, `getblocks`, `block`, `tx`, `tip`, `getaddr`, `addr`, `ping` et `pong` (annexe F). Un nœud ouvre au plus 8 connexions sortantes et accepte au plus 32 connexions entrantes.

8.2 Trouver les autres nœuds sans serveur central

NumFranc ne dépend d'aucun serveur. Un nœud trouve ses pairs de quatre façons :

1. **Sur le réseau local** : une annonce UDP toutes les 10 secondes sur le port 6910.
2. **Par l'annuaire décentralisé de BitTorrent** (DHT « Mainline », utilisé par des millions d'ordinateurs) : chaque nœud s'inscrit sous un identifiant propre à NumFranc, dérivé de SHA3-512 et de l'empreinte du genesis, puis cherche les autres inscrits. La recherche a lieu toutes les minutes au démarrage, puis toutes les 10 minutes.
3. **Par l'ouverture automatique du port** sur la box internet (UPnP, puis NAT-PMP si l'UPnP ne répond pas), pour être joignable de l'extérieur. La demande est renouvelée toutes les 10 minutes et répétée toutes les 5 minutes tant qu'elle échoue ; si le programme s'arrête, la box referme le port d'elle-même en moins de 20 minutes. L'état (ouvert, ou pourquoi pas et quoi faire) s'affiche sur l'accueil et dans Réglages > Connexion. Sans port ouvert, le nœud fonctionne quand même : il se connecte aux autres, mais les autres ne peuvent pas se connecter à lui en premier.

4. **Par la mémoire des pairs** déjà rencontrés, enregistrés dans le fichier `peers.txt`.

Au démarrage, un nœud attend 20 secondes de synchronisation avant de commencer à miner, pour ne pas travailler sur une chaîne en retard. Il demande les blocs manquants par lots de 500 au plus.

8.3 La file des transactions

Les transactions valides mais pas encore incluses dans un bloc attendent dans une file de 64 Mio au plus. Un seul enregistrement de pseudo à la fois peut y viser un même nom : cela évite qu'un mineur produise un bloc contenant deux enregistrements contradictoires. Quand la file est pleine, comme dans Bitcoin, une transaction qui paie plus de frais par octet prend la place des moins chères au lieu d'être refusée.

8.4 Protections contre les abus

Un réseau ouvert reçoit forcément des messages de pairs malveillants. Depuis la version 1.0, et à la suite d'un audit de sécurité, NumFranc reprend les protections de Bitcoin :

- **Preuve avant confiance** : un nœud suspend son minage pour se synchroniser seulement si un pair lui a **livré** récemment un bloc valide. Une simple annonce de hauteur, invérifiable, ne suffit pas.
- **Blocs envoyés un par un** : en réponse à une demande, les blocs sont lus et envoyés un à un, dans la limite de 500 blocs ou 32 Mio. Les demandes trop rapprochées sont ignorées et un pair trop insistant est déconnecté.
- **Vérifications de la moins chère à la plus chère** : une transaction relayée ne doit pas dépasser 2 Mio ; on vérifie ensuite que les pièces dépensées existent, et seulement à la fin les signatures, hors des verrous. Une signature fausse vaut l'exclusion du pair pendant 24 heures.
- **Annuaire des pairs à deux listes**, sur le modèle d'*addrman* de Bitcoin : les adresses déjà testées et les nouvelles. Les adresses sont rangées par quartier d'Internet (les deux premiers nombres de l'adresse IPv4) : au plus 8 nouvelles et 4 testées par quartier, au plus 64 apportées par un même pair, et un remplacement au hasard quand une liste est pleine. Les adresses privées (box, réseau local) venant d'Internet sont refusées : les ordinateurs de la maison se trouvent par la découverte locale. Un nœud ouvre au plus une connexion sortante par quartier et accepte au plus 4 connexions entrantes d'un même quartier. Un pair ne peut envoyer qu'une adresse toutes les 10 secondes, en plus de la réponse à notre propre demande.
- **Les pairs honnêtes ne sont pas punis** (version 1.1) : un bloc daté de plus de 2 heures dans le futur selon notre horloge est ignoré sans exclure son envoyeur (l'horloge de l'un ou l'autre ordinateur peut être décalée), et le programme invite à vérifier l'heure. Un bloc déjà rejeté, ou construit sur un bloc rejeté, fait seulement couper la connexion, sans exclusion de 24 heures. Plus de 20 blocs orphelins (parent inconnu) par minute venant d'un même pair font aussi couper la connexion.
- **Un pair lent ne retarde personne** : blocs et transactions sont envoyés à chaque pair en parallèle, si bien qu'une connexion surchargée ne bloque plus les autres.
- **Pas de gel pendant une dépense** : le gros calcul du code de dépense (Argon2, plusieurs secondes) est fait avant de prendre le moindre verrou, si bien que la chaîne, le minage et la fenêtre continuent de tourner pendant la préparation d'un paiement.

9. Portefeuille

9.1 Un portefeuille déterministe

Toutes les clés d'un portefeuille proviennent d'une **graine secrète unique de 32 octets**. La clé numéro i est obtenue ainsi :

```
clé_i = ML-DSA-87.génération(SHA3-512("NFR-wallet-key" ‖ graine ‖ i)[0..32])
```

Une réserve de 1 000 adresses est calculée à l'avance. Recevoir de l'argent et miner ne demandent donc jamais la graine. Le portefeuille garde une **adresse principale** permanente pour recevoir ; chaque récompense de minage et chaque monnaie rendue arrivent sur une adresse neuve.

9.2 Deux mots de passe, deux niveaux

Le fichier du portefeuille contient deux sections chiffrées séparément :

Section	Ouverte par	Permet	Coût d'un essai (Argon2id)
La vue (adresses, carnet)	mot de passe d'ouverture, 12 caractères minimum	voir le solde, recevoir, miner	256 Mio, 3 passes (environ 1 s)
Le coffre (la graine)	code de dépense, 16 caractères minimum	envoyer de l'argent	1 Gio, 4 passes (environ 3 à 5 s)

Chaque section est chiffrée deux fois (AES-256-GCM puis XChaCha20-Poly1305) avec une clé de 512 bits dérivée par Argon2id. Le nom de la section est authentifié : on ne peut pas intervertir la vue et le coffre. Un voleur qui copie le fichier doit payer plusieurs secondes de calcul et 1 Gio de mémoire **pour chaque mot de passe essayé**.

9.3 Le code de secours en trois morceaux

À la création, la graine est découpée par le **partage de secret de Shamir** (2 parmi 3, sur le corps fini GF(256)) en trois morceaux écrits `NFRS-…`. **N'importe lesquels deux** reconstituent la graine. Un seul morceau ne révèle rien. On peut ainsi garder un morceau chez soi, un deuxième chez un proche et un troisième dans un autre lieu : la perte ou le vol d'un seul morceau ne fait rien perdre.

9.4 Le gel

Le **gel** retire la graine de l'ordinateur ; il n'en reste qu'une empreinte. Même avec les deux mots de passe, plus rien ne peut être envoyé depuis cet ordinateur, mais recevoir et miner continuent. Pour dégeler, il faut le mot de passe d'ouverture, le code de dépense et **deux morceaux** du code de secours.

9.5 La protection des gros paiements

L'utilisateur peut fixer un seuil. Au-dessus, chaque paiement est envoyé comme **paiement retenu** (section 6.6) pendant un délai choisi, durant lequel il peut être annulé. Un **gardien** facultatif, c'est-à-dire un autre portefeuille ou un groupe multi-signature, peut aussi l'annuler. Un voleur qui obtiendrait les mots de passe ne pourrait donc pas vider le portefeuille instantanément.

9.6 Fonctions d'usage quotidien

- **Empreinte visuelle** : chaque adresse est associée à 4 mots simples et une couleur, pour vérifier d'un coup d'œil qu'on paie la bonne personne.
- **Message joint** : jusqu'à 140 octets de texte, publics et signés avec le paiement.
- **Carnet d'adresses** et paiement vers un **pseudo**.
- **Liens et QR codes de paiement** : `nfr:NFR1-…?amount=12.5&message=Merci`. Sur Windows et Linux, un clic sur le lien ouvre NumFranc avec le formulaire d'envoi déjà rempli ; rien n'est payé sans vérification et sans le code de dépense.
- **Preuve de réserve** (`NFRPROOF1-…`) : prouver qu'on détient au moins un montant, en ne révélant que les adresses nécessaires et en incluant un mot choisi par le vérificateur, pour empêcher la réutilisation d'une ancienne preuve.
- **Propositions multi-signature** (`NFRMS1-…`) : une transaction à faire circuler entre les membres d'un groupe jusqu'à atteindre le seuil.

10. Application et distribution

NumFranc est **un seul programme**. Un double-clic ouvre une fenêtre qui crée ou ouvre le portefeuille, puis lance le nœud et le minage. La fenêtre montre l'essentiel : Accueil, Envoyer, Recevoir, Historique, Carnet et Réglages, plus Sécurité et Explications dans un menu « Plus ». Le **mode expert** (dans les Réglages) ajoute Explorateur, Émission, Avancé et Réseau. L'interface existe en **français et en anglais**, en thème clair ou sombre. Au premier lancement, le programme se range tout seul dans le menu des applications et enregistre les liens de paiement `nfr:`, uniquement dans le dossier personnel de l'utilisateur. Depuis la version 1.1, il est distribué par des **installeurs « double-clic »** : un disque `.dmg` pour le Mac (on glisse NumFranc vers Applications), un installeur `.exe` pour Windows (sans droits d'administrateur ; la désinstallation ne touche jamais au portefeuille) et des paquets `.deb` pour Linux (amd64 et arm64). Ils ne sont pas signés par Apple ni Microsoft, ce qui demande des comptes payants liés à une identité réelle : la première ouverture affiche donc un avertissement, expliqué sur la page de téléchargement. Les empreintes des installeurs figurent dans le même fichier signé `SHA3SUMS.txt` : la signature des versions les couvre aussi.

Le logiciel est écrit en **Rust**, un langage qui empêche par construction des familles entières d'erreurs mémoire. Il est publié sous **licence MIT** sur GitHub. À chaque version, un service d'intégration continue exécute la suite de tests, puis compile le programme pour cinq plateformes : Linux x86-64, Linux ARM64, macOS Apple Silicon, macOS Intel et Windows x86-64. Il produit aussi l'archive `NumFranc.zip` et le fichier d'empreintes `SHA3SUMS.txt`. La compilation est **reproductible** : la version de Rust, les dépendances et les chemins sont fixés, si bien que deux compilations du même code donnent exactement le même programme. Le service le vérifie en compilant deux fois la version Linux ; chacun peut refaire l'essai avec `scripts/build-reproductible.sh`.

11. Signature des versions

11.1 Le problème

Un attaquant pourrait publier une fausse version de NumFranc, par exemple pour voler des portefeuilles. Les empreintes `SHA3SUMS.txt` détectent un fichier abîmé, mais pas un faux complet : l'attaquant pourrait publier ses propres empreintes.

11.2 La solution

Chaque version publiée est accompagnée de `SHA3SUMS.txt.sig`, une signature **SLH-DSA-SHAKE-128s** du fichier d'empreintes. Seul le détenteur de la clé secrète de signature peut la produire. Tout le monde peut la vérifier avec la **clé publique** (32 octets), publiée dans le dépôt :

`5e1d1c80fdd07d3bc4095f45a05649fc006bce347609b1566b3c25e940881736`

La vérification se fait avec l'outil fourni :

```
signer_version verifier SHA3SUMS.txt SHA3SUMS.txt.sig cle-publique.txt
```

Si la signature est fausse, l'outil affiche « signature FAUSSE : ne pas utiliser ce fichier ». Ce livre blanc lui-même est inclus dans les fichiers signés.

Le bouton **« Vérifier les mises à jour »** (Réglages) fait la même vérification automatiquement : il télécharge `SHA3SUMS.txt` et sa signature, contrôle la signature avec la clé publique inscrite dans le programme, et ne propose la nouvelle version que si elle est valide. Le numéro de version est lu dans la première ligne du fichier d'empreintes, elle aussi couverte par la signature.

11.3 Protection de la clé de signature

La clé secrète a été créée sur un ordinateur déconnecté du réseau. Elle n'existe plus en entier nulle part : elle a été découpée par partage de Shamir (deux morceaux sur trois suffisent), chaque morceau a été chiffré, puis rangé hors ligne dans un lieu distinct. Elle n'est reconstituée que le temps de signer une version, hors ligne.

11.4 En cas de compromission

Si la clé de signature était un jour compromise, une nouvelle paire serait créée, la nouvelle clé publique serait publiée dans le dépôt avec un avis explicite, et les versions signées après la date de compromission seraient déclarées non fiables. Les versions signées auparavant resteraient valides.

12. Feuille de route

Les éléments suivants **ne sont pas encore implémentés**. Ils sont listés sans date, par ordre approximatif de priorité :

- **Audit de sécurité par des experts humains** du code de consensus et du portefeuille. Un premier audit, réalisé par une intelligence artificielle indépendante, a conduit aux protections de la section 8.4.

- **Nœuds d'amorçage publics** : des nœuds toujours allumés pour faciliter la première connexion. La découverte actuelle (DHT, réseau local, UPnP) fonctionne sans eux.
- **Ordinateur de signature dédié**, jamais connecté, pour les versions.
- **Interface RPC locale** (`127.0.0.1:6911`), accessible uniquement depuis l'ordinateur lui-même, pour de futures passerelles d'échange. L'adresse est réservée dans le code mais le service n'existe pas encore.
- **Synchronisation par en-têtes** (comme Bitcoin) : télécharger d'abord les en-têtes, légers, avant les blocs.
- **Points de contrôle** ajoutés à chaque nouvelle version.
- **Débit** : étude de techniques pour réduire la place des signatures post-quantiques dans les blocs.

13. Conclusion

NumFranc applique les mécanismes éprouvés de Bitcoin avec une cryptographie choisie pour durer : signatures ML-DSA-87, empreintes SHA3-512, versions signées en SLH-DSA, minage Argon2d accessible à tous. Son émission est fixe, entièrement minée et publique. Son portefeuille et ses transactions intègrent des protections concrètes contre les erreurs et les vols. Le projet est ouvert : chacun peut lire le code, le vérifier, faire tourner un nœud et proposer des améliorations.

Références

1. S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.
2. NIST, *FIPS 204 — Module-Lattice-Based Digital Signature Standard* (ML-DSA), août 2024.
3. NIST, *FIPS 205 — Stateless Hash-Based Digital Signature Standard* (SLH-DSA), août 2024.
4. NIST, *FIPS 202 — SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, 2015.
5. A. Biryukov, D. Dinu, D. Khovratovich, S. Josefsson, *RFC 9106 — Argon2 Memory-Hard Function*, IETF, 2021.
6. P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, 1994.
7. L. Grover, *A Fast Quantum Mechanical Algorithm for Database Search*, 1996.
8. A. Shamir, *How to Share a Secret*, Communications of the ACM, 1979.
9. Y. Nir, A. Langley, *RFC 8439 — ChaCha20 and Poly1305 for IETF Protocols*, IETF, 2018.
10. NIST, *SP 800-38D — Galois/Counter Mode* (GCM), 2007.
11. Bitcoin, *BIP 113 — Median time-past as endpoint for lock-time calculations*, 2015.
12. BitTorrent, *BEP 5 — DHT Protocol*.
13. D. Crockford, *Base 32 Encoding*.
14. INSEE, *Population au 1er janvier — Données annuelles de 1990 à 2026*.

Annexes techniques

Ces annexes décrivent NumFranc avec assez de précision pour qu'un autre programmeur puisse vérifier le logiciel ou en écrire une version compatible. En cas de différence, le code source de la version correspondante fait foi.

Annexe A — Paramètres de consensus et de réseau

Constante	Valeur	Signification
`MIN_UNIT`	100 000 000	unités par NFR
`HARD_CAP`	6 910 000 000 000 000 unités	69 100 000 NFR
`PALIER_BLOCKS`	100 000	blocs par palier d'émission
`SLOW_START_REWARD`	691 000 000	6,91 NFR, palier 0
`PEAK_REWARD`	6 910 000 000	69,10 NFR, palier 1
`TARGET_BLOCK_TIME`	691 s	temps cible entre blocs
`DIFFICULTY_WINDOW`	691 blocs	fenêtre d'ajustement
`EXPECTED_TIMESPAN`	476 790 s	690 × 691
`MTP_BLOCKS`	11	blocs du temps médian passé
`MAX_FUTURE_DRIFT`	7 200 s	avance maximale d'une date
`HEADER_SIZE`	212 octets	en-tête de bloc
`MAX_BLOCK_SIZE`	8 388 608 octets	8 Mio
`MAX_TX_SIZE`	8 384 512 octets	8 Mio − 4 096
`PROTOCOL_VERSION`	4	version des blocs
`TX_VERSION`	4	version des transactions
`MAX_MEMO`	140 octets	message joint
`MIN_FEE_PER_KB`	1 000 unités	par kilo-octet (1 000 octets) entamé
`COINBASE_MATURITY`	100 blocs	maturité des récompenses
`MAX_REORG_DEPTH`	10 blocs	réorganisation maximale
`SAFE_CONFIRMATIONS`	6	confirmations conseillées
`MAX_TARGET`	$2^{512} - 1$	cible la plus facile
`GENESIS_TARGET`	$2^{492} - 1$	cible initiale
`GENESIS_TIMESTAMP`	1 792 195 200	17/10/2026 00:00:00 UTC
`GENESIS_NONCE`	452 924	nonce du genesis

Constante	Valeur	Signification
`POW_MEMORY_KIB`	4 096	mémoire Argon2d (Kio)
`POW_SALT`	`NumFranc-PoW-v1!`	sel de la preuve de travail
`LOCK_TIME_THRESHOLD`	500 000 000	hauteur ou date
`MAX_MULTISIG_MEMBERS`	15	membres d'un groupe
`NAME_MIN_LEN` / `NAME_MAX_LEN`	3 / 32	longueur d'un pseudo
`SIG_CONTEXT`	`NFR-tx-v1`	contexte ML-DSA
`P2P_PORT`	6910	port TCP et UDP
`MAGIC`	`4E 46 52 04`	« NFR » et version 4
`MAX_OUTBOUND` / `MAX_INBOUND`	8 / 32	connexions
`MAX_BLOCKS_PER_REPLY`	500	blocs par réponse
`STARTUP_SYNC_SECS`	20 s	attente avant de miner

Empreinte complète du bloc genesis :

`7e0e17a3e283a0291917d8f8c04605bbd5ab00b8e0459baafc0dd0d698238c74`
`fb5fbc288403453f5f3ad2aef5aed636b367f8b654e6bd5e7327360257cbf79e`

Annexe B — Calculs de la preuve de travail et de la difficulté

- Empreinte du bloc : `H(en-tête) = SHA3-512(SHA3-512(en-tête de 212 octets))`.
- Preuve de travail : `Argon2d(mot de passe = en-tête, sel = "NumFranc-PoW-v1!", m = 4 096 Kio, t = 1, p = 1, sortie = 64 octets)`, version 0x13.
- Validité : la sortie Argon2d, lue comme un entier big-endian de 512 bits, est strictement inférieure à la cible.
- Travail d'un bloc : $\lfloor 2^{512} / (\text{cible} + 1) \rfloor$.
- Ajustement, à toute hauteur h multiple de 691 : $T = \text{date}(h-1) - \text{date}(h-691)$, borné à $[119\,197, 1\,907\,160]$; $\text{nouvelle cible} = \lfloor \text{ancienne cible} \times T / 476\,790 \rfloor$, bornée à $[1, 2^{512} - 1]$.
- Aux autres hauteurs, la cible est celle du bloc parent.

Annexe C — Format binaire des transactions

Tous les entiers sont en big-endian. Un « varint » est un entier de taille variable, sous forme canonique obligatoire.

Champ	Taille
version	4
verrou global (non utilisé par les règles actuelles)	4

Champ	Taille
nombre d'entrées	varint
pour chaque entrée : txid précédent	64
index de la sortie précédente	4
— coinbase (txid nul, index 0xFFFFFFFF) : longueur des données	1
— coinbase : données (8 octets de hauteur, puis 0 à 92 octets libres)	8 à 100
— entrée simple : type = 0, clé publique ML-DSA, signature	1 + 2 592 + 4 627
— entrée multi-signature : type = 1, M, N, N adresses	1 + 1 + 1 + 32 × N
— puis nombre de signatures K, et K fois (index, clé publique, signature)	1 + K × (1 + 2 592 + 4 627)
nombre de sorties	varint
pour chaque sortie : montant	8
adresse	32
verrou de déblocage	4
indicateur d'annulation (0 ou 1), puis adresse d'annulation si 1	1 (+ 32)
longueur du message joint	varint
message joint (UTF-8)	0 à 140
indicateur de pseudo (0 ou 1)	1
si 1 : type (0 = enregistrer, 1 = transférer), longueur, nom, adresse propriétaire	1 + 1 + n + 32

Identifiants et signatures :

- `txid = H(transaction sérialisée)` ;
- `sighash = H("NFR-SIGHASH" || transaction)`, où les clés et signatures des entrées simples sont remplacées par des zéros de même taille, et où, pour les entrées multi-signature, le seuil et la liste des membres sont conservés mais les signatures retirées ;
- adresse simple : `SHA3-512(clé publique)[0..32]` ;
- adresse de groupe : `SHA3-512("NFR-MULTISIG" || M || N || adresses triées)[0..32]` ;
- frais minimaux : `⌈taille / 1 000⌉ × 1 000` unités, plus les frais de pseudo s'il y a lieu.

Règle de dépense d'une sortie : si son verrou est levé, seul le destinataire peut la dépenser ; sinon, seule l'adresse d'annulation le peut, si elle existe. Un verrou nul est toujours levé ; un verrou inférieur à 500 000 000 est levé à partir de cette hauteur ; sinon, il est levé quand le temps médian passé atteint cette date.

Annexe D — Formats de texte

Objet	Forme	Détail
Adresse	`NFR1-XXXX-…`	base 32 de Crockford de (adresse ‖ `H(adresse)[0..4]`), 58 caractères en groupes de 4
Morceau de secours	`NFRS-…`	partage de Shamir de la graine, avec somme de contrôle
Lien de paiement	`nfr:<adresse ou pseudo>?amount=…&message=…`	lisible en QR code
Preuve de réserve	`NFRPROOF1-<hex>`	adresses révélées, signatures et mot de défi
Proposition multi-signature	`NFRMS1-<hex>`	transaction sérialisée à compléter
Pseudo	`nom.nfr`	3 à 32 caractères `a-z 0-9 -`

Annexe E — Calendrier d'émission

Récompense par bloc au début de chaque palier, en NFR. Chaque palier compte 100 000 blocs, soit environ 2,19 ans.

Palier	Récompense	Émis avant le palier	Début (années)
0	6,91	0	0,0
1	69,10	691 000	2,2
2	62,19	7 601 000	4,4
3	55,971	13 820 000	6,6
4	50,3739	19 417 100	8,8
5	45,33651	24 454 490	11,0
6	40,802859	28 988 141	13,1
7	36,7225731	33 068 426,9	15,3
8	33,05031579	36 740 684,21	17,5
9	29,74528421	40 045 715,79	19,7
10	26,77075578	43 020 244,21	21,9
11	24,0936802	45 697 319,79	24,1

Les paliers suivants continuent la baisse de 10 % par palier, en division entière sur les unités. La somme atteint exactement 6 910 000 000 000 000 unités au bloc 4 471 948, au palier 44. À partir du bloc 4 471 949, la récompense est nulle et les mineurs ne sont plus rémunérés que par les frais.

Annexe F — Messages réseau

Commande	Contenu	Rôle
`version`	version, hauteur, port d'écoute, nonce	présentation à l'ouverture d'une connexion
`verack`	vide	accusé de réception de `version`
`getblocks`	liste d'empreintes connues	demande des blocs manquants
`block`	un bloc complet	transmission d'un bloc
`tx`	une transaction	relais d'une transaction
`tip`	une hauteur	annonce de la hauteur de sa chaîne
`getaddr`	vide	demande d'adresses de pairs
`addr`	liste d'adresses	réponse à `getaddr`
`ping` / `pong`	nonce	vérification que la connexion est vivante

Le nonce de `version` permet à un nœud de détecter qu'il s'est connecté à lui-même.

Annexe G — Glossaire

- **Bloc** : paquet de transactions ajouté à la chaîne par un mineur.
- **Coinbase** : première transaction d'un bloc, qui crée la récompense du mineur.
- **Confirmation** : nombre de blocs ajoutés après celui qui contient une transaction.
- **Empreinte (hash)** : résumé de taille fixe d'une donnée ; la moindre modification change tout le résumé.
- **Genesis** : le tout premier bloc, inscrit dans le logiciel.
- **Graine** : secret de 32 octets dont proviennent toutes les clés d'un portefeuille.
- **Mempool** : file des transactions en attente d'un bloc.
- **Nonce** : nombre que le mineur fait varier pour trouver une preuve de travail valide.
- **Nœud** : ordinateur qui fait tourner NumFranc et vérifie la chaîne.
- **Post-quantique** : se dit d'un algorithme conçu pour résister aux ordinateurs quantiques.
- **Preuve de travail** : calcul coûteux à produire mais rapide à vérifier, qui protège la chaîne.
- **Réorganisation** : remplacement des derniers blocs de la chaîne par une chaîne concurrente ayant plus de travail.
- **Temps médian passé (MTP)** : médiane des dates des 11 derniers blocs.
- **UTXO** : sortie de transaction non encore dépensée ; ce sont les « pièces » de NumFranc.

Le code de NumFranc n'a pas encore fait l'objet d'un audit de sécurité indépendant. Aucune vente de NFR n'est proposée et aucune valeur n'est promise. Ce document décrit un logiciel libre ; il ne constitue ni une offre, ni un conseil financier.