

“ **Périmètre** — Sous-titres sur flux live poussé depuis un encodeur (RTMP/RTMPS, SRT, HLS ingest). Le cas du VOD posté après coup est signalé séparément quand il diffère.

Dernière vérification — 3 septembre 2026. Les plateformes sociales font évoluer ces mécanismes sans préavis : revalider avant tout engagement client.

1. Les quatre mécanismes possibles

Avant le tableau par plateforme, il faut distinguer les voies de transport, parce que « supporter les sous-titres » ne veut pas dire la même chose selon la plateforme.

Mécanisme	Description	Transport
In-band SEI	CEA-608 / CEA-708 encodés dans le flux vidéo élémentaire H.264, dans le user_data SEI (ATSC A/53 ou A/72). Standard historique broadcast.	RTMP, SRT/TS, HLS ingest
Tag script RTMP	<code>onCaptionInfo</code> , tag script AMF0 transportant une matrice ECMA. Format courant chez les encodeurs et media servers internet.	RTMP uniquement
Canal hors bande	URL d'ingestion HTTP dédiée, alimentée en parallèle du flux vidéo.	HTTP, hors du flux
Burn-in (open captions)	Texte incrusté dans l'image. Fonctionne partout, non désactivable par le spectateur, invisible pour les lecteurs d'écran.	Tous

À quoi s'ajoute l'**ASR plateforme** : auto-captions générés côté plateforme depuis la piste audio. Ne demande rien à l'encodeur, mais on ne le contrôle pas.

2. Tableau de synthèse

Plateforme	In-band 608/708	onCaptionInfo	Canal dédié	ASR plateforme	Multi-langue live
YouTube Live	Oui	Non documenté	Oui, URL signée	Oui	Non, une piste
LinkedIn Live	Oui (pass-through)	Non documenté	Non	Oui, anglais seul	Non
Facebook Live	Oui, via Live API	Non documenté	Non	À vérifier	Non documenté
Twitch	Oui (ATSC A/72)	Oui	Non	Non	Non documenté
X / Twitter	Non documenté	Non documenté	Non	Non	—
Instagram Live	Non applicable	—	—	Oui	Non

3. YouTube Live

In-band

Dans les réglages d'encodeur, choisir les sous-titres EIA-608 / CEA-708, parfois appelés « embedded ». Selon le setup, ils sont lus depuis un format de fichier ou encodés en temps réel.

Limitation importante : le standard 608/708 supporte jusqu'à quatre pistes de langue, mais YouTube n'en prend en charge qu'une seule.

Canal dédié

En bas de l'onglet *Setup* des paramètres, une section *Closed Caption* permet d'activer les sous-titres et d'exposer une **Captions Ingestion URL**, qui est une URL HTTP signée. Chaque point d'entrée de flux ne peut avoir qu'un seul feed de sous-titres.

C'est la voie utilisée par les logiciels de sténotypie et les prestataires CART (Eclipse/AccuCap, Case CATalyst BCS, CaptionMaker, etc.).

Contrainte de délai

Avec les logiciels supportés, ajouter un délai de diffusion de **30 ou 60 secondes** dans l'onglet *Advanced settings* avant l'événement.

HLS ingest

Les sous-titres embarqués passent dans le conteneur M2TS comme n'importe quel flux TS. Voie à privilégier si on est déjà en HLS ingest pour du HEVC ou du HDR.

4. LinkedIn Live

Deux sources, avec priorité

Le sous-titrage peut venir soit du **pass-through closed captioning**, soit des **auto-captions**, ces derniers étant uniquement en anglais. Si les deux sont disponibles sur un même flux, le lecteur n'affiche **que les auto-captions**. Le bouton CC apparaît en bas à droite du lecteur.

Point de scéno à ne pas rater

Les sous-titres sont **toujours affichés en bas de l'écran**, et LinkedIn demande explicitement de ne placer aucun titrage, bandeau ou sticker en bas de l'image pour éviter le chevauchement.

→ À intégrer dans les gabarits de lower-thirds **dès la conception graphique**.

Périmètre de l'ASR

L'auto-captioning LinkedIn Live a été lancé via l'API, avec une technologie Microsoft, pour les diffuseurs passant par les partenaires API (Restream, StreamYard, Switcher, Socialive), **en anglais uniquement**.

Conséquence pour les clients francophones : l'ASR LinkedIn ne couvre pas le français. Pour du sous-titrage FR sur LinkedIn Live, il faut passer par du pass-through 608 injecté en amont par un prestataire, ou par du burn-in.

VOD LinkedIn (hors live)

Ajout de sous-titres avec un fichier **SRT** attaché à la vidéo avant publication, depuis le desktop uniquement, pour les comptes personnels comme pour les Pages. Les auto-captions à l'upload sont réservés aux vidéos en anglais, avec une option de relecture avant publication.

5. Facebook Live

Le support repose sur les sous-titres **CEA-608 injectés dans le flux**, pour les éditeurs utilisant la Live API, éventuellement via un prestataire tiers qui les insère en temps réel.

Au lancement, Facebook ne proposait **pas** de sous-titrage automatique pour le live, contrairement aux vidéos de Page enregistrées. Seuls les éditeurs passant par la Live API et générant du CEA-608 pouvaient ajouter des sous-titres.

“ ⚠ Ces éléments sont anciens et l'API Meta a beaucoup bougé depuis. À revalider dans la doc développeur Meta avant tout engagement client.

6. Twitch

Twitch documente plusieurs méthodes de transmission, dont deux clairement identifiées :

1. **CEA-708/EIA-608 embarqués dans le flux vidéo élémentaire**, comme décrit dans ATSC A/72 (SEI user_data). Format courant des encodeurs broadcast.
2. **CEA-708/EIA-608 transmis via un tag script/AMF0 RTMP** `onCaptionInfo`, format le plus répandu chez les encodeurs internet et media servers, dont Elemental et Wowza. Le payload doit contenir une matrice ECMA avec deux paires d'éléments :
 - une chaîne nommée `type` contenant les caractères `708`
 - une chaîne nommée `data` contenant le payload CEA-708/EIA-608 encodé en base64

En pratique, l'écosystème Twitch grand public passe surtout par des **extensions d'overlay** installables depuis le dashboard, activables en panel, composant ou overlay. Ce n'est pas une piste CC dans le lecteur, c'est une surcouche.

7. X / Twitter et Instagram Live

X — Aucun mécanisme d'ingestion de sous-titres live documenté. Burn-in ou rien.

Instagram Live — Pas de chaîne d'ingestion tierce pertinente : l'ASR plateforme est la seule voie, et il n'y a pas de piste CC contrôlable depuis l'encodeur.

8. Règles opérationnelles à retenir

Le multilingue live ne passe pas par 608

Le standard offre quatre canaux, mais YouTube n'en lit qu'un et LinkedIn ne fait de l'ASR qu'en anglais. Pour un événement multilingue, les trois options réalistes sont :

- un flux et un événement **par langue**
- du **burn-in** par langue
- une **application de sous-titres déportée**, hors du lecteur

Le 608 est un tunnel étroit

Positionnement, casse et polices sont figés, sans personnalisation possible.

Le passthrough se casse silencieusement

OBS ne fait pas passer les 608 embarqués dans une source média : le passthrough de flux média n'est pas dans le périmètre des développeurs. Toute chaîne avec un mixeur logiciel au milieu doit être **testée bout en bout**, jamais supposée.

Toujours répéter en unlisted

Le sous-titrage est la fonction la plus fragile de la chaîne : elle dépend de l'encodeur, du transcodeur, du transport et du lecteur, et elle échoue **sans alarme**.

Prestataires à connaître

Pour industrialiser : SyncWords, AI-Media, EEG/Falcon, 3Play, Verbit, StreamText, Videolinq. Tous parlent 608, la plupart s'insèrent en relais RTMP(S) ou SRT sans matériel dédié.

Sources

Sujet	Référence
YouTube — prérequis sous-titres live	support.google.com/youtube/answer/3068031
YouTube — HLS ingest	support.google.com/youtube/answer/10349430
YouTube — comparaison protocoles d'ingest	developers.google.com/youtube/v3/live/guides/ingestion-protocol-comparison
LinkedIn — côté spectateur	linkedin.com/help/linkedin/answer/a554237
LinkedIn — côté diffuseur	linkedin.com/help/linkedin/answer/a569473
LinkedIn — sous-titres VOD (SRT)	linkedin.com/help/linkedin/answer/a552177
LinkedIn — auto-captions à l'upload	linkedin.com/help/linkedin/answer/a1327025
Twitch — méthodes de transport	doc Twitch sous-titres / ATSC A/72, onCaptionInfo
Facebook Live — CEA-608 via Live API	annonce 2017, à revalider dans la doc Meta