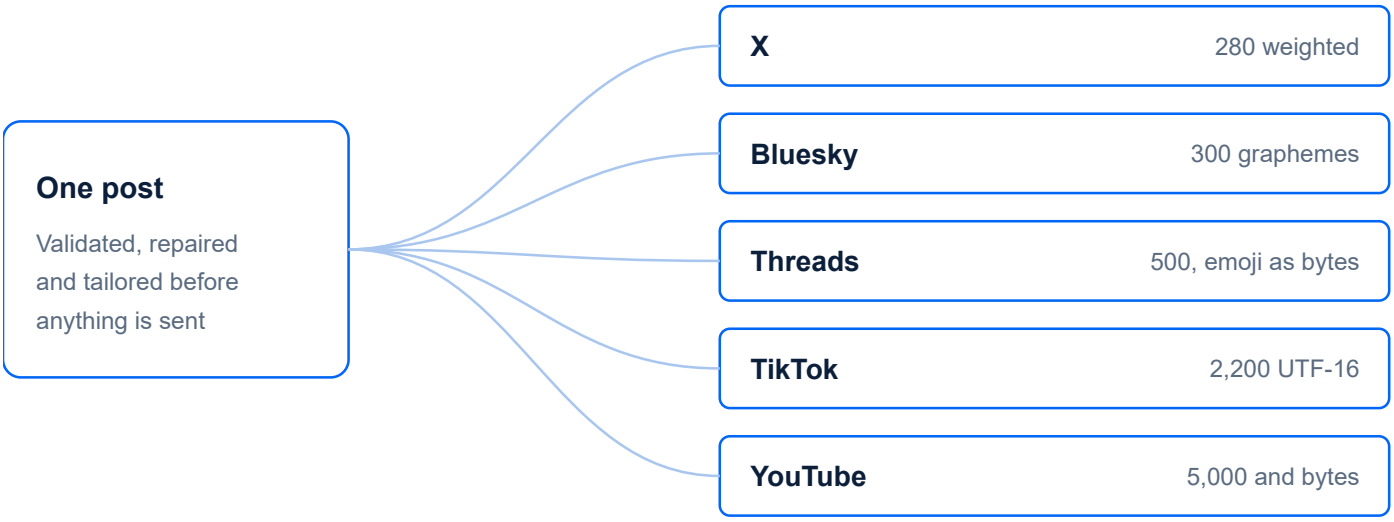


WHITE PAPER

Publishing Across Ten Social Networks

Modeling each network's real limits, validating before sending, and catching the rules when they change.

Ron Alterio, Founder

November 2026

EXECUTIVE SUMMARY

"Post everywhere" is ten different contracts

Social networks differ in what they count as a character, which edge of an image a size limit applies to, which actions exist at all, and when a rejection arrives. Many of those rejections come after an upload has been accepted, in a message the person who wrote the post cannot act on. The rules also change without notice.

This paper presents how Xorventa models those differences as data, validates a post against them before anything is sent, repairs what can be repaired automatically, and monitors each network's documentation monthly for changes.

98%

Of posts sent were accepted, across ten networks.

0%

Rejected for length, format, size, frame rate or duration.

5 states

For every network and action, separating "the network has no API" from "not built".

21 pages

Of developer documentation compared with the recorded contract every month.

Written for: engineers and technical leaders building or evaluating multi-network publishing.

1. The problem

A publishing integration can be connected, authorized and working, and still fail on the next post. Connection says almost nothing about what will happen when this particular text and this particular file reach this particular network.

1.1 The same words have different lengths

A character count is not one measurement. Each of these networks measures text differently, so one counter over one number is wrong on most of them.

- **X** uses a weighted count of 280, where a link always costs 23 and many characters outside Latin scripts count double.
- **Bluesky** allows 300 graphemes, and also 3,000 bytes.
- **TikTok** counts 2,200 in UTF-16 units.
- **YouTube** allows 5,000 characters and 5,000 bytes in a description, so accented or CJK text reaches the byte limit first.
- **Threads** allows 500 characters and charges emoji by their bytes. Applying that byte rule to every character would block a valid caption over two em dashes.

1.2 Limits that read alike mean different things

"1440" on Instagram and Threads is a maximum width. "1080p" for a TikTok photo is best read as the shorter edge, which is why 1080 x 1920 passes. On most other networks the limit is the longest edge. Applying the wrong one quietly shrinks every tall image further than the network asks. Under-width images show the same pattern: Meta scales them up, so refusing them blocks posts it would have published.

1.3 The expensive failures arrive late

Several networks accept a request and fail it during asynchronous processing. A video with an out-of-range frame rate can be accepted and uploaded, then fail after a poll of up to five minutes. By then the person has moved on, and "the post failed" tells them nothing about what to change.

1.4 Some rules are not in the documentation

TikTok's photo documentation publishes no list of formats. A PNG carousel passed initialization and then failed asynchronously with `file_format_check_failed`, while the same images converted to JPEG, on the same account and code path, published. X's documentation lists a video size range that the network does not enforce, and that its own recommended 720 x 1280 would fall outside. Written limits need testing against what the network actually does, in both directions.

2. Why the conventional approaches fall short

Common approach	Why it falls short for this problem
One green "connected" status	It describes the credential, not the action. A connected network may allow posts and not replies, or may offer no such API at all.
Send, then report the network's error	Errors arrive late and in the network's own terms, after the time and money of an upload, with nothing the writer can act on.
Use the strictest limit everywhere	Every network inherits the worst rule. A caption fit for 280 weighted characters wastes a 63,206-character Facebook post.
Copy limits from consumer help pages	App limits and API limits differ, and an outdated figure can block a whole feature while looking cautious. Video limits on a page about direct messages do not apply to posts.
Treat missing features as a backlog	"The network offers no such API" and "we have not built it" are different facts for planning, and showing them alike misleads everyone downstream.
Review documentation when something breaks	Developer pages change without a changelog or a date. The first signal is a customer's failed post.

3. The solution

The design treats each network's contract as data that the product, its documentation and its monitoring all read, so they cannot disagree. Three layers use it: capability says what can happen, validation says whether this post can, and monitoring says whether the record is still true.

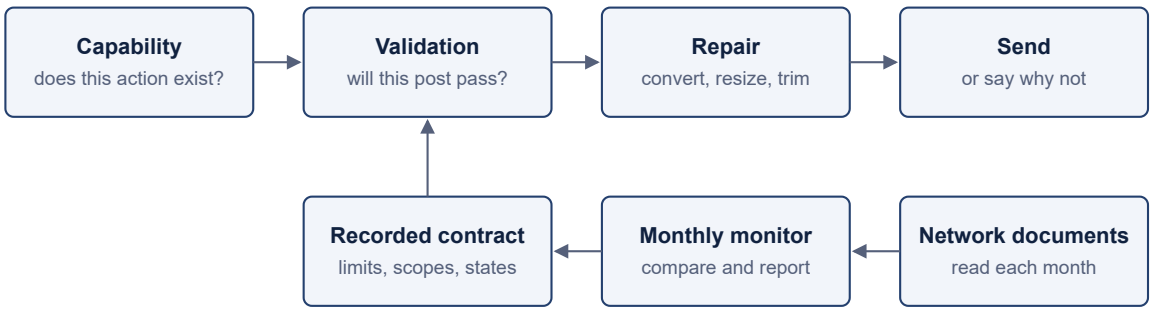


Figure 1. The recorded contract drives validation before each post (top). The monitor compares it with what each network now publishes (bottom), and a person acts on any difference.

3.1 Capability is a state, not a yes or no

Every network and action has one of five states: available, needs a connection, not authorized (built, but the permission is not held), not implemented (our gap) and not supported (the network has no such API). The last two look identical to a user and mean opposite things for planning, so they are separate states. A network's reply can be structurally absent, as with organic comments on TikTok and Pinterest, and it stays absent regardless of what we build.

Network	Post	Reply	Direct message
Facebook	Yes	Yes	Yes
Instagram	Yes	Yes	Yes
Bluesky	Yes	Yes	Yes
Threads	Yes	Yes	No API
LinkedIn	Yes	Yes	No API
YouTube	Yes	Yes	No API
Google Business Profile	Yes	Yes	No API
X	Yes	Not offered	Not offered
TikTok	Yes	No API	No API
Pinterest	Yes	No API	No API

The matrix is generated from the same record the product uses to decide whether an action may run, and a test asserts that each implemented action resolves to real code.

3.2 Each limit is recorded with its unit and its axis

Network	Text limit	Unit	Images	Size limit applies to	Video frame rate
X	280	Weighted	4	Longest edge	Up to 60
Bluesky	300	Graphemes, and bytes	4	Longest edge	—
Threads	500	Characters, emoji as bytes	20	Width	23–60
Pinterest	800	Characters	5	Longest edge	—
Google Business Profile	1,500	Characters	Text post	—	—
Instagram	2,200	Characters	20	Width	23–60
TikTok	2,200	UTF-16 units	35	Shorter edge	23–60
LinkedIn	3,000	Characters	20	Longest edge	—
YouTube	5,000	Characters, and bytes	Video only	—	—
Facebook	63,206	Characters	10	Longest edge	—

Limits as recorded November 2026, read from each network's developer documentation. Networks revise these, which is why Section 3.5 exists.

A hard limit is recorded only when it is documented and the network really rejects past it. That posture keeps validation free of false refusals: a limit we cannot confirm stays unset, and a value that cannot be measured from the file never blocks a post.

3.3 Repair first, refuse only when it cannot be repaired

Situation	What happens	Reasoning
Image in a format the network does not take	Converted to JPEG	Converted and untested is safe. Passed through and untested is how a format rejection arrives late.
Image over the size limit	Downscaled on the correct axis	The person's intent is clear, and the network's limit decides the rest.
Instagram image outside its shape limits	Padded to shape	Cropping would remove part of what the person chose to show.
More images than allowed	Extras dropped, with a warning	A shorter carousel is better than no post.
Media on a network that takes text only	Media dropped, and the text posts	The message still reaches the network.
Text over the limit	Shortened by the network's own unit, keeping links intact	A trimmed link is a broken post. The count is checked again on what is actually shown.
Video out of range	Held before upload, with the limit and the network named	A video cannot be repaired here, and an early, specific message saves the wait.

3.4 Validation runs from the file, at the cost of one read

Duration, width and frame rate all come from one probe of the video, so adding a bound costs an attribute read rather than a second download. Each network is checked separately, so one network holding a post back does not stop the others. A check that needs a value it cannot read does not refuse the post.

3.5 A monthly monitor for silent changes

On the first of each month, an automated job reads 21 developer documentation pages across all ten networks and compares them with the recorded contract: permissions, media limits, rate limits, new mandatory interface requirements, review requirements, pricing and re-authorization deadlines. For each network it reports changed, unchanged or can't tell. A page that cannot be read is reported as "can't tell" instead of passing silently.

- **It never edits code.** Limits are contracts, and a model's reading of a web page should not change one on its own. A person reads the report and decides.
- **Decisions are recorded.** Where we satisfy a rule differently or have chosen to differ, the decision is stored with its reason, so it does not return as a false alarm each month.
- **It works.** The November 2026 run flagged two changes out of ten networks: Instagram's carousel limit and X's video frame rate. Both were confirmed against the source and applied the same day.

3.6 Limits that are not technical

A post can be valid and still unwise. Posting to the same X Community more than once in 24 hours can get a business removed from it, so the second post is held with that reason, even though the API would accept it. Costs differ by network as well: on X, a post carrying a link costs many times what one without does, so the cost is shown when it is decided, not on an invoice afterwards.

4. The evidence

4.1 How we test

Question	Method
Does the code match each network's rules?	More than 1,900 automated tests in the social and media areas, plus a stored snapshot of each network's permissions that fails if changed by accident.
Do the written limits hold in practice?	Live posts to all ten networks. Where documentation and behaviour disagree, the observed result decides, as with TikTok's photo formats.
Does the record stay true?	The monthly monitor in Section 3.5.
Does it work end to end?	Our own publishing through the product, since September 2026, read from the publishing record.

4.2 Results

Of posts sent to a network	Share
Accepted and published	**98%**
Rejected by the network	2%
Rejected for length, format, size, frame rate or duration	0%

All ten networks. Every rejection had one cause: X refusing a post to a Community the account had not joined. That is a membership state, which the product now explains to the person.

Network decisions before sending	Share of all posts
Sent	91%
Deliberately not sent, with a reason shown	9%

Reasons for not sending were the 24-hour Community limit, a choice to post to X Communities and not the timeline, and video on a network that could not take it at the time.

> **The result that matters is the zero. Every rejection for length, format or size that validation prevents is one that would otherwise have arrived late, in the network's words, to someone who had moved on.**

4.3 Scope of these measurements

These results come from our own publishing through the product, across all ten networks. They show that the validation and the recorded limits agree with what the networks accept. Results for any particular account also depend on its type and history, because some network limits vary by account.

5. How this behaves when someone posts

Moment	What the person sees
Choosing networks	What each one can do, by action, before anything is written.
Writing	A count in each network's own unit, with a tailored version per network instead of one shared caption.
Attaching media	Recommended sizes for the chosen networks, and a warning for any network that will not take the file.
Before scheduling	Each problem named by network and limit, with the fix where there is one.
After publishing	A result per network, so one problem never hides nine successes.

6. Guidance for teams building similar systems

If you are deciding	Our recommendation	What would change it
How to record limits	As data with a unit and an axis, in one place that the product and its checks both read.	One or two networks, where a simple constant may be enough.
What counts as a hard limit	Only what is documented and confirmed. Never block on a value you could not measure.	A network that penalizes the account for a rejected attempt.
Repair or refuse	Repair when the person's intent is clear. Refuse early and specifically when it is not.	Brand or legal content that must not be altered.
What to do with missing actions	Separate "the network has no API" from "not built", in the data and on screen.	Nothing. This one is cheap and pays back quickly.
Which limit to believe	Test written limits against real behaviour, with live accounts, in both directions.	A contractual API specification.
How to keep up	Monitor documentation on a schedule, compare it with your record, and keep a person in charge of the change.	A network with a real, versioned changelog.

7. Conclusion

Publishing to many networks is a data problem before it is an integration problem. Record each contract precisely, with its unit, axis and evidence, and the product can say in advance what will happen and repair most of what would otherwise fail. A record of this kind goes out of date without warning, so monitoring it is part of the design.

These results come from our own publishing across all ten networks, within the scope in Section 4.3. The method transfers directly, and we encourage teams to run it against the networks they depend on.

References

Each network's developer documentation, as used for the recorded limits.

- X, Manage Posts: integration guide
- Meta, Instagram media reference
- Meta, Threads posts
- TikTok, Content Posting API media transfer guide
- Bluesky, AT Protocol image embed lexicon
- Pinterest, API v5: create pin
- Google, YouTube Data API: videos
- LinkedIn, Comments API

THE PRACTICAL CONCLUSION

Publishing to many networks is a data problem before it is an integration problem.

Limits as data, with a unit and an axis

Validate first, and repair what can be repaired

Separate "no API" from "not built"

Monitor the rules, with a person in charge of the change

Explore Xorventa at xorventa.ai