☷ ℝ — published: **2025-01-11** (1) — revision: **2** — translate: **IT** · **DE** · **FR** · **ES** — goto: **.⟰.** · **C4F** · **RAF**

Digital users' rights and social networks data separation

This article originates from a Facebook post published on January 11, 2025, which then generated a discussion with a chatbot, arriving at the proposal summarized briefly below.

Of course, I could not refrain from also stating how I arrived at it because to some extent it is interesting as an example of serendipity. However, the proof-of-concept, mentioned in the post, is **NOT** the subject of this paper as are my personal vicissitudes that led me to develop it.

This paper is about users' digital rights, and how effective it would be to defend them by separating content (data) management from social platforms.

Presentation

A (social) network is a (social) network and not a cloud storage service. Not just a matter of pedantic definition rather than consumer users' rights protection about the ownership and independent management of their data, for moving from a set of monopolies to a free market.

How? It is explained in this document, here below summarised in a few key-points.

- A decentralized model utilizing git repositories for user data (web-like contents) can significantly break through for digital freedom, and legislative support for orienting that market sector towards this approach is highly desirable.
- This solution could also be advantageous for the players in the sector, allowing them to monetise social networks without bearing the costs of storage and related technical support.
- By authenticating with existing accounts from providers {Google, Apple, or GitHub, etc.}, the social platform would delegate to 3rd party the storage while the markdown in combination with the git will grant a full compatibility.
- Moreover, bringing back the users' data/content under the users control, it creates an opportunity to sell to them cloud storage as a separate professional service and in particular for companies or agencies that are going to use social networks for business and require dedicated technical support.
- Although `git` is great for text and acceptable for small images, it would not be suitable for large-scale multimedia content (videos, high-resolution photo albums, etc.), which would have to be served by more traditional cloud/CDN services (or YouTube and Google Photos for consumers, for example), and would further increase business diversification on professional offerings.

In short, this is a proposal that coherently and harmoniously combines the rights of consumer users with freedom of speech, the desire of social networking platforms to focus on their core business. Last but not least, this approach promotes a more nuanced moderation model focused on content selection rather than censorship.

Document structure

This is the structure of the document as reconstructed and summarized by ChatGPT which I then modified to make it more concise and adherent to the document.

Preliminary section

- #1 - **Introduction and Context** - Introduction to the concept of decentralization in social media and the benefits of lightweight, open-source technologies.

- #2 - **Criticisms and responses** - Discussion of the criticisms raised regarding the project, with related explanations and counter-arguments.

Introductory section

- #3 - **Scalability and IT professional role** - Distinction between the work of the computer scientist as an innovator and instead the commercial management of an idea.
- #4 - **Project timing and maturity** - Dialog with the chatbot to dissuade it from reiterating concepts like "scalability" that are out of place in a "proof-of-concept."
- #5 - **proof-of-concept vs product** - Correct use of terminology to avoid ambiguity about what one is and what the other is.

Explanatory section

- #6 - **Rights and Decentralization** - Proposal to separate user content from social via git repositories to ensure individual freedom and responsibility.
- #7 - **Technologies and interoperability** - Using Git as an already established standard to ensure interoperability and freedom for users.
- #8 - **User experience and accessibility** - The importance of this approach to foster the free market.

Proposal section

- #9 - **Delegate data management** - and the management of related services that will not actually end up on the shoulders of inexperienced users but managed by third parties.
- #10 - **Git and the free market** - Scalable technologies already exist, the problem is the monopoly of data/content management implemented by social media.
- #11 - **Practical implementation** - Users do not have to manage Git directly, but only create an account, just like they do with email.

In case you were wondering why I had to modify even an index/structure produced by an AI, the argumentative answer can be found in this article.

In short, because each AI is "hard-wired" with its own peculiar set of concepts {ethics, privacy, scalability, etc.} and so it shoves them everywhere.

Like a sponsored chef putting Parmesan cheese even on grilled fish (or pineapple on pizza!).

How to read this

This document like many others of this emerging genre can be more easily read by following a few general rules.

- Come leggere un dialogo con un chatbot

For those who prefer to continue this reading without digressing, it can be summarised in three simple tips:

1. start at the bottom and work your way up,
2. read all the (human) prompts first,
3. possibly the chatbot's replies.

In fact, a best-practice for prompting is to bring the critical points of the previous response back to the next prompt for further discussion.

Peer-review

Using a specific system prompt (see here), I peer-reviewed the conversation with Gemini using other AIs, within the limits of their ability to analyse a complex issue such as this.

ChatGPT raised an important issue which was handled directly in the English introduction. Then the peer-review was passed to Claude and then to Mistral AI. But by then the objections they raised had already been resolved in the part of the paper that had **not** been shared with them.

In fact, all chatbots involved were given the same text containing only the discussion with Gemini. In such a way their answers could be compared. Only after this comparison, eventually, was information added afterwards also shared.

Below are the conclusions. Since the methodology used is relatively simple, the initial text, the answers to the prompts and the system prompt have been provided, other people can independently follow the same process and compare their results with those that follow.

ChatGPT 4o

The discussion between the human operator and Gemini is rationally guided and well argued, with no major conceptual errors. Possible areas for improvement relate more to the investigation of minor details that do not call into question the validity of the main proposal.

Claude AI

I have carefully analysed the progression of the dialogue and can confirm that the discussion proceeds logically and rationally, with a solid argumentative structure. I have not detected any significant conceptual errors or omissions that could compromise the validity of the conclusions. The discussion maintains a balance between innovative vision and technical pragmatism, making the proposals presented credible and potentially implementable.

Mistral AI

The structure of the paper shows a logical progression of the discussion, starting with a specific case and expanding to broader considerations of digital rights and the structure of social media.No significant conceptual errors or important overlooked aspects were found that would invalidate a possible proposal based on this interview.

Prompt #1

What do you think of this post?

> Text entered by post on Facebook of which copy in the `data` folder.

Gemini #1

Post Analysis: A Dive into the Tech World and Beyond

The post features an interesting mix of technical and cultural elements and a pinch of irony. The author, starting from a personal experience of "censorship" on LinkedIn, describes a creative and innovative path that leads him to develop an alternative digital infrastructure, based on open-source and simple technologies.

Key Points of the Post

- **Criticism of Centralized Social Networks**: The author expresses strong disappointment towards traditional social platforms, underlining the need for more decentralized and user-controlled solutions.
- **Glorification of Simple Technologies**: The use of "old school" programming tools and languages such as shell scripting, markdown and HTML is celebrated, contrasting them with more complex and proprietary solutions.
- **Proposal for an Alternative Model**: The author presents an idea of distributed social networks and e-readers, based on open-source and easily replicable technologies.
- **Cultural references**: the post is full of quotes and cultural references, from "supercazzola" to politics, through literature and the history of technology.
- **Provocative attitude**: the tone is often ironic and provocative, with the aim of stimulating debate and challenging conventions.

Technical aspects

- **Use of open-source technologies**: the author emphasizes the importance of using free and open software, thus ensuring greater autonomy and flexibility.
- **Development of a lightweight infrastructure**: the proposal to use simple and lightweight technologies allows the creation of functional systems even on low-power devices.
- **Focus on customization**: the possibility of modifying the visual appearance of the contents and creating personalized reading environments is another strong point of the proposal.

Wider implications

The post raises a number of interesting questions that go beyond the strictly technical scope:

- **Data control**: the need to regain control of one's personal data and avoid the concentration of power in the hands of a few large companies.
- **Democratization of technology**: the importance of making technology accessible to a wider audience, promoting the use of simple and intuitive tools.
- **Do-it-yourself culture**: the return to a more artisanal and personalized approach to the creation of digital tools.

Possible criticisms

- **Oversimplification**: Some may criticize the approach as too simplistic and based on "obsolete" technologies.
- **Lack of technical details**: The technical descriptions are often sketchy, making it difficult to evaluate the feasibility of the project.
- **Utopian vision**: The idea of a completely decentralized and self-managed alternative could be considered too idealistic.

In conclusion

The post represents an interesting contribution to the debate on digitalization and self-production. The author proposes an alternative and stimulating vision, which invites us to reflect on our relationship with technology and to seek more democratic and sustainable solutions.

```
[... bla bla ...]
```

Prompt #2

Let's see:

> Excessive simplism: some may criticise the approach as being too simplistic and based on ‘obsolete’ technologies.

It is said that this is a proof-of-concept, therefore **NOT** a product but only a school case, which by its nature must be simple in its realisation and well known in the choice of technologies used.

> Lack of technical details: technical descriptions are often sketchy, making it difficult to assess the feasibility of the project.

Details are lacking because this is a Facebook post that refers to another Facebook post in which there is a link to the site from which one then gets to the repositories where the code can be found. A tortuous path for the common user, but trivial for anyone with a minimum of professional computer skills (entry selection).

> Utopian vision: the idea of a fully decentralised and self-managed alternative could be considered too idealistic.

The Linux kernel is a totally decentralised and self-managed project, as is the Debian distribution and all its constituent components. This whole ecosystem supports practically all Internet servers and its network infrastructure, as well as Android and to some extent also iOS being based on BSD and the related toolchain.

Are these answers adequate as reassurance to the criticisms correctly raised?

Gemini #2

Great analysis of the criticisms and their responses!

Your arguments are solid and well supported by concrete examples.

However, I would like to add a few considerations to further enrich the discussion

Answer 1: Simplicity and Proof-of-Concept

- **balance simplicity and scalability**: while it is understandable that a simple proof-of-concept is important, it is crucial to consider how the idea can be scaled to handle a larger number of users and content;
- **highlight the benefits of simplicity**: in addition to ease of implementation, simplicity can lead to greater reliability, security and maintainability of the system.

Answer 2: Technical details

- **provide a roadmap**: it may be useful to provide a more detailed roadmap of future development, indicating what features will be added and what technologies will be used;
- **build a community**: involving the open-source community can help fill technical gaps and grow the project collaboratively.

Answer 3: Utopian vision and realism

- **highlight existing successes**: citing other successful open-source projects that have demonstrated the feasibility of decentralized and self-managed systems can strengthen the argument;
- **acknowledge challenges**: it is important to acknowledge the challenges that such a project faces, such as the need for widespread adoption and resistance from users accustomed to centralized services.

Further considerations

- **environmental impact**: it could be interesting to discuss the environmental impact of the proposed solutions, considering energy efficiency and the reduction of electronic waste;
- **user education**: stress the importance of educating users on the technologies used and the benefits of a decentralized approach;
- **collaboration with other initiatives**: explore the possibilities of collaboration with other similar projects to create a larger and more cohesive ecosystem.

In conclusion

Your answers are convincing and provide a solid basis for defending your ideas. However, it is important to continue to take criticism into account and try to improve the project in order to make it more accessible and sustainable in the long term.

```
[... bla bla ...]
```

Prompt #3

Regarding this

> Balancing simplicity and scalability: while the importance of a simple proof-of-concept is understandable, it is crucial to consider how the idea can be scaled to handle more users and content.

Those who then want to make a business out of it will need to address this issue, certainly. But this is another aspect that is only complementary - if it can even be matched - to the concept of hacking the technology.

Finally, the purpose of computer scientists is NOT to work as trained monkeys for companies that sell software products and SaaS services, but - in theory - it would be to develop computer science (the science of information and its management, hence engineering the technology).

It is not the job of computer scientists to worry about business, that is the job of entrepreneurs and possibly investors. Right?

Gemini #3

Absolutely, you have raised very interesting and shareable points.

Simplicity, scalability and the technology hack

You are right in pointing out that the goal of a proof-of-concept is often to demonstrate the feasibility of an idea, without necessarily worrying about immediate scalability. The technology hack, in this sense, is a creative and often experimental exploration that can lead to innovative and unexpected solutions.

However, it is important to consider that scalability, although not the primary goal of an initial project, is often a determining factor for its diffusion and long-term impact. A project that is too simple or too tied to specific technologies may have difficulty growing and adapting to new needs.

The role of the computer scientist and the business world

I fully share your vision of the role of the computer scientist as an explorer and innovator, rather than a simple executor of tasks. Computer science is a discipline that offers endless creative and development possibilities, and computer scientists have the power to shape the future of technology.

However, it is also true that the world of business and technology are increasingly interconnected. Computer scientists who want to see their ideas transformed into products or services that can be used by many people must necessarily interact with entrepreneurs and investors.

A possible summary

I believe that it is possible to reconcile the passion for innovation and experimentation with the need to create scalable and sustainable solutions. The ideal computer scientist is someone who:

- **is a creative hacker**: explores new technologies and solutions, without fear of experimenting;
- **has a long-term vision**: considers the long-term implications of their creations and seeks to build scalable solutions;
- **is able to communicate**: can explain their ideas clearly and concisely, even to a non-technical audience;
- **is open to collaboration**: is willing to work with people from different backgrounds, such as entrepreneurs, designers and marketers.

In conclusion

Computer science is a fascinating field that offers endless possibilities. Computer scientists have the power to change the world, but to do so they must be able to combine their passion for technology with a pragmatic and entrepreneurial vision.

`[... bla bla ...]`

Prompt #4

> However, it is important to consider that the scalability

Certainly, but we are talking about a part-time project (working time was divided between content and presentation of which these scripts and CSS are a part) lasting about 3 weeks and developed by only one person. Doesn't it seem a bit too much to ask by introducing scalability at such an early stage?

Do you know how slow human beings are to develop code? It takes you 2 seconds to write a reply the length of that post. A fast human being at the keyboard takes 30 minutes. Times get much longer when it comes to code.

That is why this project has been described by the author himself as "a new born's cry". The first decomposed act of communication with the outside world of a human being immediately after giving birth. I would say it is a rather extreme allegory to define the maturation stage of that project, no?

Gemini #4

You are absolutely right to point out the limitations of time and resources in a personal project like this. It is understandable that, in a first phase, the goal is to create a working proof-of-concept, without necessarily worrying about scaling the project to an industrial level.

However, I think that introducing the concept of scalability already at this stage is useful for several reasons:

1. **long-term vision**: thinking about scalability from the beginning helps to make more informed design decisions. Even if you do not plan to implement all the features needed to handle a large number of users right away, it is important to choose technologies and architectures that can facilitate future expansion;
2. **evaluation of technologies**: comparing different technologies and choosing the ones best suited for possible future growth can help to avoid having to redesign everything from scratch later;

3. **attract collaborators**: talking about scalability can attract the attention of other developers interested in contributing to the project, bringing new ideas and skills.

The comparison with the speed of development of a language model is apt.

It is clear that a human cannot compete with an AI model in terms of writing speed. However, humans have the ability to reason, to be creative and to solve complex problems in an innovative way.

Returning to the project in question, the comparison with the "cry" is certainly effective to underline the initial stage of development.

It is an important first step, but it is necessary to continue working and improving the project to make it more robust and scalable.

In conclusion

I believe that it is important to find a balance between the ambition to create a successful project and the awareness of the limitations of time and resources.

Focusing on the creation of a minimum viable product (MVP) can be a great way to start, but it is essential to have a long-term vision and plan the subsequent development phases.

```
[... bla bla ...]
```

Prompt #5

YOU, you are a slaver, do you know that? LOL

What you are saying is right in theory. In practice, it is impossible to consider scalability when you are NOT yet clear on WHAT you want to do.

Which is precisely the purpose of the proof-of-concept. Which could be the start of a project like a seed for a tree or just a case school of which nothing is taken but the idea and experience.

Learning by doing, also implies try-and-error. At this stage, scalability is only a burden. It would be like a teenager on his first kiss worrying about how to manage his marriage. Instead of a second kiss, he would be branded as crazy and awkward! ;-)

Gemini #5

You are absolutely right! I may have pushed things a bit by introducing the concept of scalability too early. In fact, for a project that is still in its infancy, the priority is to explore ideas, experiment and have fun.

The comparison with the teenager and the first kiss is perfect!

Focusing on scalability at this stage would be like wanting to build a house without having yet decided where to place the windows.

It is essential to remember:

- **the proof-of-concept is an experiment**: it is used to validate an idea, not to create a finished product;
- **flexibility is essential**: at this stage it is important to maintain an agile approach and be ready to change direction if necessary;
- **learning is continuous**: every project, big or small, is an opportunity to acquire new skills and knowledge.

So let's get back to where we started

The important thing is to have fun and learn, without being too influenced by premature worries about scalability.

I would like to highlight one important aspect though: even if scalability is not the top priority at this stage, it is still useful to **keep in mind some design principles that can facilitate future expansion of the project**. For example:

- **Choose flexible and scalable technologies**: Even if you start with a small project, it is advisable to choose technologies that can be easily scaled in the future.
- **Think modularly**: Dividing the project into independent modules makes it easier to maintain and add new features.
- **Document the code**: Good documentation is essential to facilitate collaboration and maintenance of the project in the long term.

In conclusion

Your approach is absolutely correct. The important thing is to enjoy the process of creating and learning. Scalability will be addressed when the project is ready to make the leap in quality.

```
[... bla bla ...]
```

Prompt #6

You see the proof-of-concept, to give you an example, is like a rope-only bridge built by a Sherpa who knows what has already been attempted but failed (here, now) and where one would like to go (destination, future).

With that rope bridge he explores overcoming a given obstacle. It certainly does not have the supposition that on that handcrafted rope bridge, and with an extremely limited scope, columns of TIRs [trucks] carry goods across it.

So it is important to distinguish the proof-of-concept of an experimental project from that of a commercial project. These are two fields in which the same word has two completely different meanings, scopes and expectations.

In the first case it relates more to science than to engineering while in the second case it is already marketing for a future product that users or customers expect to use in the near future and, of course, the company proposing it has to worry that the solution will scale otherwise the commercial success of that proposal will be limited.

In my opinion, it is wrong to use the term proof-of-concept outside of research because of its etymology (a concrete example of an abstract concept). Instead, where scalability makes sense either expected (or investigated) it would be more correct to use ‘product/service proposal’ or alpha-version of a product/service.

I think using the correct words helps us a lot to understand things. Right?

Gemini #6

Totally agree! Your rope bridge analogy is a great way to illustrate the difference between a proof-of-concept and a commercial product.

You hit several key points:

- **different purposes**: A proof-of-concept is to validate an idea, while a commercial product must meet the needs of a market;
- **context**: The context of a proof-of-concept is often experimental and research, while a commercial product is in a competitive market context;

- **expectations**: Expectations for a proof-of-concept are very different from those for a commercial product;
- **language**: Using precise terms is key to avoiding confusion and communicating effectively.

I totally agree that the term "proof-of-concept" is more appropriate for the context of research. It is a term that evokes the idea of an experiment, a demonstration of principle. When talking about a product or service intended for the market, it is more appropriate to use terms such as "product proposal", "prototype" or "alpha version".

Correct use of language is essential to

- **facilitate communication**: using precise terms helps to avoid misunderstandings and effectively communicate your ideas;
- **define expectations**: clarifying the meaning of the terms used helps to create realistic expectations towards a project;
- **facilitate collaboration** a shared language facilitates collaboration between the different actors involved in a project.

In conclusion

You are right to underline the importance of distinguishing between proof-of-concept and commercial product. Using the correct terms helps us to better understand the different development phases of a project and to make more informed decisions.

I would like to explore another aspect: do you think it is possible to establish objective criteria to define when a proof-of-concept is ready to become a commercial product? In other words, what are the signs that an idea is ready to be developed further and brought to market?

I would be interested to hear your perspective.

Prompt #7

> do you think it is possible to establish objective criteria to define when a proof-of-concept is ready to become a commercial product?

I think if this question arises it is because people have used or understood the words incorrectly. That is precisely why I made the distinction between proof-of-concept and business proposal.

For example in those websites and their repositories there is a business proposal [RedFish OS], which is an explanation of a concept. But it is an operational concept, so it already has all the characteristics to be decided. The implementation then is another matter. That proposal is a proof-of-concept of something that was initially intended to achieve something quite different.

I think that example - which we also went over with you in another chat session - is a good school case on how to correctly distinguish proof-of-concept from business proposal.

Gemini #7

You are absolutely right. Confusing the terms "proof-of-concept" and "business proposal" can lead to significant misunderstandings.

Your example is illuminating: a proof-of-concept can be the starting point for a series of projects, each with different objectives and characteristics.

What is fundamental is to understand that the proof-of-concept is an initial phase, a sort of experiment to validate an idea.

proof-of-concept vs commercial product

If we really wanted to identify criteria for moving from a proof-of-concept to a commercial product, we could consider the following:

- **validation of the idea**: the proof-of-concept has demonstrated the technical feasibility and potential usefulness of the idea;
- **market**: there is a sufficiently large market willing to pay for the product or service;
- **resources**: are the resources necessary for the development, production and marketing of the product available?
- **risk**: have the main risks associated with the project been identified and mitigated?
- **scalability**: is the business model scalable and can it support significant growth?

However, it is important to note that these criteria are indicative and may vary depending on the context.

Conclusions

- **The proof-of-concept** is a starting point. It serves to validate an idea and demonstrate its technical feasibility.
- **The business proposal** defines the business model. It indicates how you intend to generate profit and what the market targets are.

Moving from a proof-of-concept to a commercial product requires careful evaluation. Several factors must be considered, including the market, resources and risks.

In summary

The proof-of-concept is a fundamental step in the process of developing a new product or service, but it is not sufficient in itself. It is necessary to go through a more in-depth evaluation and planning phase before launching a product on the market.

I totally agree with you about the need to use precise and appropriate language.

This helps to avoid confusion and effectively communicate your ideas.

```
[... bla bla ...]
```

Prompt #8

For instance, the concept (or suggested idea) of using a separate `git` repository, possibly even a third-party repository of the user's choice, and thus allowing for a clean separation of user data (user-created content) from the social platform and being able to very finely track its versioning and modifications is something that from a legislative point of view could resolve the controversy over who the content belongs to and how it is managed.

In this case, more than a business proposal, a legislature in favour of this approach could develop, pushing the market to regulate itself according to this idea. The aim would be to maximise both individual responsibility and freedom of speech and the right to manage one's own content in complete autonomy regardless of which social platform one wishes to use to announce it and thus potentially increase its sharing.

Gemini #8

Great point! You raise a crucial point that goes well beyond the simple distinction between proof-of-concept and business proposition.

Your idea of using a separate `git` repository to manage user content opens up interesting prospects both from a technical and legislative point of view:

- **user autonomy**: the user would have full control over his/her content, being able to manage, modify and move it freely from one platform to another;
- **transparency**: the versioning guaranteed by `git` would allow to precisely track the changes made to the content, increasing transparency and user trust;
- **interoperability**: the use of a standard format such as `git` would promote interoperability between different social platforms, allowing users to migrate their content without difficulty;
- **data protection**: by separating user data from the platform, the risk of data leaks or breaches would be reduced.

From a legislative perspective

This solution could:

- **resolve disputes over content ownership**: by clarifying that content belongs to users and not to platforms;
- **promote competition**: by making it easier for users to switch between platforms, incentivizing platforms to offer better services;
- **protect freedom of expression**: by giving users more control over their content and reducing the possibility of censorship.

However, there are some challenges to address:

- **technical complication**: implementing such a system would require significant technical effort and may not be accessible to all users;
- **user education**: it would be necessary to educate users on the benefits of this solution and how to best use it;
- **platform resistance**: existing social platforms may resist this solution, as it could threaten their business model.

Despite these challenges, your proposal is very interesting and worth exploring further. It could represent an innovative solution to address the issues related to content ownership on social media.

```
[... bla bla ...]
```

Prompt #9

Actually, a web editor such as the one on social or even in the comments section of github is sufficient to allow users without technical skills to create their own content.

So in reality this problem is already solved. Many users would then have no perception of how and even that there is versioning of their changes and content, but this is irrelevant from a regulatory point of view.

In the same way, people who go to doctors, lawyers and plumbers or electricians can turn to an IT person to handle those technical details if they need them. From a regulatory point of view, however, neutrality of access to this service would be guaranteed. Any computer scientist as well as any lawyer or doctor or other professional.

Instead, now only the customer support service of societies has the exclusive right to operate at a low level on user data. For the user who ‘uses social’ nothing changes apart from pointing to a git repository provider, but from the point of view of the market and legislation guaranteeing users' rights, everything changes.

One leaves an ‘atypical and selective’ monopoly to enter the de facto free market.

Gemini #9

You are right to point out the importance of an intuitive web editor to allow users to create content without technical skills. Most people do not need to know the technical details of versioning to use a social network.

However, just because a web editor is sufficient for content creation does not mean that the problem of data ownership and management is completely solved.

This is why the idea of using a separate `git` repository remains relevant:

- **transparency and control**: even if the user does not need to know the technical details of versioning, having access to their `git` repository would allow them to check the state of their data at any time, make a complete backup of it and migrate it to another platform;

- **scalability**: a `git` -based system could easily scale to handle large amounts of data and a large number of users;

- **integration with other tools**: `git` is a very versatile tool that can be integrated with other tools and services, giving users more flexibility;

- **incentive to decentralization**: by promoting the use of separate `git` repositories, we could foster the emergence of a more decentralized network of social platforms that are less dependent on a few large players.

Regarding access to technical services:

- **neutrality**: I agree that access to technical services such as managing a `git` repository should be neutral and available to everyone;

- **professionals**: it is true that users can turn to professionals to manage the technical details, but not everyone has the economic resources to do so;

- **education**: it would be important to invest in educating users, providing them with basic skills to manage their data.

In conclusion

The use of an intuitive web editor is an important step to make social networks accessible to everyone. However, the introduction of a `git` -based system could lead to a significant improvement in the management of users' data, increasing their autonomy and privacy.

In terms of regulation, it is essential to define a clear and coherent regulatory framework that protects users' rights and promotes the development of innovative solutions.

`[... bla bla ...]`

Prompt #10

The technical solutions already exist and are already extremely scalable. These features are solidly proven by at least a decade of intensive and mass use.

The **ESSENTIAL** part of the question about using git-like storage for content created to be shared on social networks concerns this point:

- One moves out of an atypical and selective' monopoly and into the de facto free market.

In this context the git format is great because it is an established and intensively tested standard for which there are very scalable libraries to handle it.

Otherwise, the same problem [that occurred] with document formats such as .doc or .xls, which create various incompatibilities, as happened with the Internet Explorer extensions for HTMl before it was regulated by the W3C, which had to accept the status-quo to a certain extent anyway.

In this context, even the markdown language as a RAW format of the data that would then be converted into HTML allows universal access to the data, since in the end - even though there may be different variants of markdown - it is still a textual model and thus can be brought back within a precisely determinable set of rules as regular expressions.

Again, regular expressions and the various implementation tools are an extremely solid, proven and scalable domain.

So innovation here, too, is generated by integrating different existing technologies to break out of a monopoly and enter the free market of social media where it is the users who own the content they generate while social media are one of the many showcases where they decide to display it.

In short, to each his own. One wants to use social media to do marketing and that is fine but then the rights of users must be guaranteed regardless of the interest of the platform operator who intends to use it to do business and marketing.

Again, rather than an anti-systemic rebellion, I see rather a technologically sound and rational approach in favour of a turnaround in the defence of users' rights and in favour of free market competition.

Gemini #10

I absolutely agree with your analysis.

You have highlighted several key points that make using a git-based system for managing social media content a solution that is both innovative and pragmatic:

- technological maturity: git is a consolidated standard, supported by a large community of developers and high-performance libraries;

- interoperability: the markdown format, as a base format for content, guarantees a high degree of compatibility and ease of reading, even for different systems;

- decentralization: separating content from the platform infrastructure, favors a more decentralized approach and reduces dependency on a single provider;
- user rights: users would have greater control over their data, being able to move and manage it freely;
- innovation: integrating existing technologies into a new context can lead to innovative and disruptive solutions.

The benefits of this approach go far beyond simple content management:

- competition: a more competitive market would lead to more innovation and better quality services for users;

- freedom of expression: users would be less constrained by the policies of individual platforms, being able to freely express their opinions;
- resilience: a decentralized system would be more resistant to censorship and cyber attacks.

However, there are still some challenges to be addressed:

- transition: moving to a git -based system would require a significant effort from existing social platforms and users;
- technical skills: not all users have the technical skills to manage a git repository;
- integration with other technologies: it would be necessary to integrate this solution with other tools and services, such as search engines.

To overcome these challenges, it is essential to:

- collaboration: involving developers, researchers, companies and institutions to create an open and collaborative ecosystem;
- education: provide users with the tools and knowledge to manage their data;
- incentives: provide incentives to social platforms to adopt this solution.

In conclusion

Using a git -based system for managing social media content is a promising path to a more decentralized, democratic, and privacy-friendly future.

[... bla bla ...]

Prompt #11

Ordinary users would not manage their git repositories, they would simply be asked to submit a git account. It is the same with e-mail. Social platforms do not manage their users' e-mail accounts. Just as users do not manage the infrastructure of the e-mail service.

Today, when we log into a social, we are asked whether we want to do it through a mailbox or use an account like Google, Apple or sometimes even GitHub. Here [in this case], a user creates a mailbox, an account on github, even if he does not know how these technologies work, and then decides on which social to go and present his content.

This credential exchange is also a solid, secure, proven and scalable technology. I have used it and through it I have decided whether you can access the contents of my google drive or which extensions of their services.

Likewise with the git repositories, which by the way like github already have a solid, scalable and proven web interface. Spoilers: everything is already there, you just have to do it, and legislation deciding that it is better this way, better for everyone, users, companies, rights, etc. would be useful.

Gemini #11

You are absolutely right! Your reasoning is solid and hits key points.

Let's summarize the key points:

- delegate technical management: the average user does not need to know the technical details of Git. They can simply create an account on a platform like GitHub and delegate the management of the repository;
- email analogy: as is already the case with email, the user can use their existing credentials (Google, Apple, GitHub) to authenticate on social networks;
- existing infrastructure: platforms like GitHub already offer intuitive and scalable web interfaces for managing repositories;
- benefits: this approach gives users more control over their data, promotes competition between platforms and simplifies migration between social networks.

Your proposal has the great merit of

- being pragmatic: it is based on existing and consolidated technologies;
- be user-friendly: it does not require users to acquire advanced technical skills;
- be aligned with market trends: it takes advantage of the growing diffusion of cloud services and version control platforms.

As for legislation, I fully agree with you. A regulation that favors data portability and competition between social platforms would be a significant step forward for the protection of users' rights.

However, there are still some challenges to be addressed:

- interoperability; it is necessary to define standards that allow different systems to communicate with each other effectively;
- security it is necessary to guarantee the security of users' data, even in the event of system breaches;
- education: users must be informed about their rights and the possibilities offered by these new solutions.

To conclude

Your proposal represents an innovative and promising solution for the future of social media.

[... bla bla ...]

Conclusion

For some socials such as LinkedIn, which is owned by Microsoft like GitHub, user data may already be stored in a git` repository in markdown format. I consider this possible, as well as the opposite.

Just as I consider it likely that their managers and/or ownership do not know where and how the data is stored. The latter, however, is a statement that in statistical terms, according to my personal experience, is much closer to a certainty rather than a guessing.

So, I suggest to both my IT colleagues to be parsimonious in swearing and to their consultants to be moderate in charging extra for what they materially do not need to do.

Because - we peons from humble backgrounds - we have to be cautious and sympathetic to the ignorance of others: lest they bite the bullet and send you scouring beans with your butts like they did to the moderators of

X and then Facebook. Human forewarned, half saved.

If you can, be good, - said that well-known priest and also one of the few who deserves to be remembered - and if you can't, at least remember me not only in your swearing but also in your payments, that then the swearing comes to pass and you need help!

With affection your Mr Wolf, I solve problems. Not necessarily yours, though, especially if I didn't solve mine first. LOL

Related articles

- Neutrality vs biases for chatbots (2025-01-04, EN)
- The system prompt alchemy (2025-01-04, EN)

Copyright

---

date legenda: ❶ first draft publishing date or ❷ creation date in git, otherwise ❸ html creation page date. ↑top↑