

Diritti digitali e separazione dei dati dai social

Questo articolo prender origine da un post su Facebook pubblicato l'11 gennaio 2024 che poi ha generato una discussione con un chatbot arrivando a formulare la proposta qui sotto riassunta brevemente in lingua inglese. Il resto del documento invece è in italiano.

Short presentation in English

A network is a 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 breakthrough for digital freedom, and a 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 among them
- Moreover, bringing back the users' data/content under the users control, it is an opportunity to sell them 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.
- Despite markdown and git are great for text and web-oriented images, git repository would not suitable for large-scale multimedia content (videos, hi-res photographic album, etc.) which necessary should being served by a more traditional cloud/CDN services (or YouTube and Google Photos for consumers), which differentiate even more the business diversification about professional offers.

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

Breve introduzione in Italiano

Questo documento parla di diritti digitali degli utenti, e di quanto sarebbe efficace difenderli separando la gestione dei contenuti (dati) dalle piattaforme social.

Ovviamente non potevo prescindere dall'indicare anche il come. In questo dialogo parto appunto da un proof-of-concept realizzato per le mie personali necessità.

Il proof-of-concept NON è l'oggetto di questo documento così come non lo sono le mie vicissitudini personali che mi hanno portato a svilupparlo.

Però da quell'esperienza ho maturato una consapevolezza che non potevo, da informatico professionista con decenni di esperienza, non calare nel contesto tecnologico e sociale attuale.

Struttura del documento

Questa è la struttura del documento così come ricostruita e sintetizzata da ChatGPT che ho poi modificato in maniera da renderla più sintetica e aderente al documento.

1. Prompt #1 - Introduzione e contesto

- Introduzione al concetto di decentralizzazione nei social media e ai vantaggi di tecnologie leggere e open-source.

2. Prompt #2 - Critiche e risposte

- Discussione sulle critiche di semplicità, mancanza di dettagli tecnici e visione utopica.
- Chiarimento del ruolo di un proof-of-concept come caso studio e non come prodotto finito.

3. Prompt #3 - Scalabilità e ruolo dell'informatico

- Distinzione tra il lavoro dell'informatico come innovatore e la gestione commerciale di un'idea.

4. Prompt #4 - Tempi e maturità del progetto

- Dialogo con il chatbot per distoglierlo dal reiterare concetti come "scalabilità" fuori luogo rispetto ad un "proof-of-concept".

5. Prompt #5 - Differenza tra proof-of-concept e prodotto commerciale

- Uso corretto della terminologia per evitare ambiguità.

6. Prompt #6 - Legislazione e decentralizzazione

- Proposta di separare i contenuti utente dai social tramite repository git per garantire libertà e responsabilità individuale.

7. Prompt #7 - Tecnologie esistenti e interoperabilità

- L'uso di Git come standard già consolidato per garantire interoperabilità e libertà agli utenti.

8. Prompt #8 - User experience e accessibilità

- L'importanza di questo approccio per favorire il libero mercato.

9. Prompt #9 - Il problema della proprietà e gestione dei dati

- e della gestione dei servizi relativi che in realtà non finirà sulle spalle di utenti inesperti ma gestito da terze parti.

10. Prompt #10 - Git e mercato libero

- Le tecnologie scalabili esistono già, il problema è il monopolio della gestione dei dati/contenuti attuato dai social media.

11. Prompt #11 - Implementazione pratica

- Gli utenti non devono gestire direttamente Git, ma solo creare un account, esattamente come fanno con le email.

Nel caso vi chiediate perché abbia dovuto modificare persino un indice/struttura prodotta da un'AI, la risposta argomentativa la trovate in questo [articolo](#).

In estrema sintesi perché ogni AI è "fissata" con un suo proprio e peculiare insieme di concetti {etica, privacy, scalabilità, etc.} che ficca ovunque tipo uno chef sponsorizzato che metta il Parmigiano Reggiano anche sulla grigliata di pesce (o l'ananas sulla pizza!).

Come leggerlo

Questo documento come molti altri di questo genere che ho pubblicato su questo sito, possono essere usufruiti in maniera agevole e piacevole seguendo poche regole generali.

1. Generalmente il primo prompt è di apertura e spesso ha un carattere di poca attinenza con le conclusioni, talvolta è persino irrilevante ma solo un modo di "rompere il ghiaccio".

Ovviamente questa necessità non esiste con i chatbot, ma noi umani siamo fatti così, abbiamo bisogno del nostro tempo per "carburare" ed entrare nel vivo della questione.

2. Quindi saltare alle conclusioni è un ottimo modo per capire di cosa veramente il documento anche perché come ho autore ho una forte attitudine ad usare i titoli per il bait-clicking

Questa attitudine è meno spinta su questo sito ma resta sempre un'ottima regola generale di non crearsi false aspettative per via del titolo ovvero giudicare il libro dalla copertina.

Attualmente i chatbot sono prolissi oltre la gestibilità umana perché scrivere testo (sputare token a raffica) è meno computazionalmente intensivo di riassumere e concentrarsi sulle questioni essenziali.

Curiosamente il rapporto S/N nelle persone segue pattern simili. Comunque, mentre la sintesi non è un grosso problema per questi chatbot, comprendere quali siano i punti salienti di un discorso è qualcosa che al momento gli è abbastanza "alieno". Oltre al fatto che in alcuni contesti, i punti salienti sono soggettivi e più in generale "l'essenziale è invisibile agli osservatori distratti".

3. Quindi potete agilmente saltare le risposte del chatbot che la parte rilevante della sua risposta viene riferita nel prompt successivo.

Nonostante questo le risposte sono in genere riportate integralmente, eccetto vaniloqui palesi che vengono riassunti con "bla bla" perché le non tutte le raffiche di token meritano la nostra attenzione.

4. Una volta saltati alle conclusioni potete risalire leggendo fino alla parte che per voi smette di essere rilevante oppure arrivare fino all'origine dell'argomento.

Questo modo di procedere anti-cronologico è, generalmente, inusuale ma per questo tipo di documenti è molto efficace e anche molto efficiente. D'altronde con l'evoluzione della tecnologia anche i modelli di letteratura cambiano e quindi il nostro modo di leggerli. Null'altro che una sorta di adattamento, fra i tanti che abbiamo già affrontato.

Modus Operandi

Anche perché lo scopo del dialogo con un chatbot - come spiegato nella homepage del sito che ospita questo articolo e in altri articoli in esso - non è quello di ascoltare l'opinione di un oracolo ma piuttosto quella di usare il dialogo per capire come spiegare le cose che interiormente a noi sono chiare - l'ovvio per noi, non è poi così ovvio per gli altri - ma soprattutto capire cosa potrebbe essere frainteso o non compreso, e quindi va spiegato meglio.

In altri casi ho usato il dialogo con il chatbot per fare peer-review alle mie idee ma in questo contesto vanto decenni di esperienza professionale e quindi posso insegnare, come infatti il chatbot ha compreso esso stesso.

Generalmente mi piace scrivere ma non mettermi in cattedra, però ogni tanto qualcuno mi ci tira per la giacchetta... e chi diavolo sono io per non accontentarli? Mi piace l'odore di prima mattina di altri "clienti" ben serviti secondo dovere e merito. LOL

Q&A DIALOG

In this case, it is not a single prompt but a dialog: a series of answered questions about the Mario Drahi's 800

billions funding plan proposed to fostering the European economy and support the innovation within the continent. Prompts and the related answers are separated by horizontal lines.

The prompts have been modified from the originals of the first commit due to corrections made in the English text, but without altering the meaning, so they should be transparent to a chatbot. On the other hand, reading at the bottom of this page about the comparison between two separate sessions, carried out even with identical prompts, repeatability is not guaranteed given the fact that the models are learning and above all are regularly updated.

UPLOAD

none

CUSTOMISATION

none

TRANSLATION

Some parts of this document (tag: 2024-10-09) were originally written in Italian. The English translation of those parts have been made by an Italian native speaker fluently in English, who has been supported by the automatic translator [deepl.com](https://www.deepl.com) in free-of-charge version to fasten the translation.

If you wish to send a translation in any other language, it is necessary that you are familiar with one of the languages in which this document is available, even better if you know English and can start from the original version, and that you are a native speaker or equivalent in the language into which the translation is being made.

COPYRIGHT

© 2024, **Roberto A. Foglietta** <roberto.foglietta@gmail.com>

This document is entirely published under *Creative Commons Attribution Non-Commercial No-Derivatives 4.0 International* license terms (CC BY-NC-ND 4.0) included the part above this statement. Moreover, if a version of this document exists or has existed under different licence terms, the licence terms of the latest version available on [github/robang74](https://github.com/robang74) [repository](#) apply. Even when the new licence terms are more restrictive, because permissions for any free content may be revoked at any time at the will of the author, and updating a licence to be more restrictive explicitly implies this will.

Prompt #1

Cosa ne pensi di questo post?

Testo inserito da [post](#) su Facebook di cui [copia](#) nella cartella data.

Gemini #1

Analisi del post: Un'immersione nel mondo tech e non solo

Il post presenta un'interessante commistione di elementi tecnici, culturali e un pizzico di ironia. L'autore, partendo da un'esperienza personale di "censura" su LinkedIn, descrive un percorso creativo e innovativo che lo porta a sviluppare un'infrastruttura digitale alternativa, basata su tecnologie open-source e semplici.

Punti chiave del post

- **Critica ai social network centralizzati:** l'autore esprime un forte disappunto nei confronti delle piattaforme social tradizionali, sottolineando la necessità di soluzioni più decentralizzate e controllate dagli utenti.
- **Esaltazione delle tecnologie semplici:** viene celebrato l'uso di strumenti e linguaggi di programmazione "vecchia scuola" come shell scripting, markdown e HTML, contrapponendoli alle soluzioni più complesse e proprietarie.
- **Proposta di un modello alternativo:** l'autore presenta un'idea di social network e e-reader distribuiti, basati su tecnologie open-source e facilmente replicabili.
- **Riferimenti culturali:** il post è ricco di citazioni e riferimenti culturali, dalla "supercazzola" alla politica, passando per la letteratura e la storia della tecnologia.
- **Atteggiamento provocatorio:** il tono è spesso ironico e provocatorio, con l'obiettivo di stimolare il dibattito e sfidare le convenzioni.

Aspetti tecnici

- **Utilizzo di tecnologie open-source:** l'autore pone l'accento sull'importanza di utilizzare software libero e aperto, garantendo così maggiore autonomia e flessibilità.
- **Sviluppo di un'infrastruttura leggera:** la proposta di utilizzare tecnologie semplici e leggere permette di creare sistemi funzionali anche su dispositivi a bassa potenza.
- **Focus sulla personalizzazione:** la possibilità di modificare l'aspetto visivo dei contenuti e di creare ambienti di lettura personalizzati è un altro punto di forza della proposta.

Implicazioni più ampie

Il post solleva una serie di questioni interessanti che vanno oltre l'ambito strettamente tecnico:

- **Controllo dei dati:** la necessità di recuperare il controllo dei propri dati personali e di evitare la concentrazione del potere nelle mani di poche grandi aziende.
- **Democratizzazione della tecnologia:** l'importanza di rendere la tecnologia accessibile a un pubblico più ampio, promuovendo l'uso di strumenti semplici e intuitivi.
- **Cultura del fai da te:** il ritorno a un approccio più artigianale e personalizzato alla creazione di strumenti digitali.

Possibili critiche

- **Eccessivo semplicismo:** Alcuni potrebbero criticare l'approccio troppo semplicistico e basato su tecnologie "obsolete".
- **Mancanza di dettagli tecnici:** Le descrizioni tecniche sono spesso sommarie, rendendo difficile valutare la fattibilità del progetto.
- **Visione utopica:** L'idea di un'alternativa completamente decentralizzata e autogestita potrebbe essere considerata troppo idealistica.

In conclusione

Il post rappresenta un interessante contributo al dibattito sulla digitalizzazione e l'autoproduzione. L'autore propone una visione alternativa e stimolante, che invita a riflettere sul nostro rapporto con la tecnologia e a cercare soluzioni più democratiche e sostenibili.

[... bla bla ...]

Prompt #2

Vediamo:

Eccessivo semplicismo: Alcuni potrebbero criticare l'approccio troppo semplicistico e basato su tecnologie "obsolete".

Viene detto che si tratta di un proof-of-concept quindi NON di un prodotto ma solo di un caso scuola che per sua natura deve essere semplice sia nella realizzazione e ben noto nella scelta delle tecnologie usate.

Mancanza di dettagli tecnici: Le descrizioni tecniche sono spesso sommarie, rendendo difficile valutare la fattibilità del progetto.

I dettagli mancano perché questo è un post su Facebook che fa riferimento ad un altro post Facebook in cui c'è il link del sito da cui poi si giunge ai repository dove si trova il codice. Un percorso tortuoso per l'utente comune ma banale per chiunque abbia un minimo di skill professionali in informatica (selezione all'ingresso).

Visione utopica: L'idea di un'alternativa completamente decentralizzata e autogestita potrebbe essere considerata troppo idealistica.

Il kernel linux è un progetto totalmente decentralizzato e autogestito così come la distribuzione Debian e tutti i componenti che la costituiscono. Tutto questo ecosistema regge praticamente la quasi totalità dei server e dell'infrastruttura di rete, Android e in parte anche iOS essendo basato su BSD e toolchain relativa.

Basta come rassicurazione alle critiche correttamente sollevate?

Gemini #2

Ottima analisi delle critiche e delle relative risposte!

Le tue argomentazioni sono solide e ben supportate da esempi concreti.

Tuttavia, vorrei aggiungere alcune considerazioni per arricchire ulteriormente la discussione

Risposta 1: Semplicità e Proof-of-Concept

- **bilanciare semplicità e scalabilità:** mentre è comprensibile l'importanza di un proof-of-concept semplice, è fondamentale considerare come l'idea possa essere scalata per gestire un numero maggiore di utenti e contenuti;
- **evidenziare i vantaggi della semplicità:** oltre alla facilità di realizzazione, la semplicità può portare a una maggiore affidabilità, sicurezza e manutenibilità del sistema.

Risposta 2: Dettagli tecnici

- **offrire una roadmap:** potrebbe essere utile fornire una roadmap più dettagliata dello sviluppo futuro, indicando quali funzionalità verranno aggiunte e quali tecnologie verranno utilizzate;
- **creare una community:** coinvolgere la comunità open-source può aiutare a colmare le lacune tecniche e a far crescere il progetto in modo collaborativo.

Risposta 3: Visione utopica e realismo

- **evidenziare i successi esistenti:** citare altri progetti open-source di successo che hanno dimostrato la fattibilità di sistemi decentralizzati e autogestiti può rafforzare l'argomentazione;
- **ammettere le sfide:** è importante riconoscere le sfide che un progetto di questo tipo deve affrontare, come la necessità di una vasta adozione e la resistenza da parte degli utenti abituati a servizi

centralizzati.

Ulteriori considerazioni

- **impatto ambientale:** potrebbe essere interessante discutere l'impatto ambientale delle soluzioni proposte, considerando l'efficienza energetica e la riduzione dello spreco elettronico;
- **educazione degli utenti:** sottolineare l'importanza di educare gli utenti sulle tecnologie utilizzate e sui vantaggi di un approccio decentralizzato;
- **collaborazione con altre iniziative:** esplorare le possibilità di collaborazione con altri progetti simili per creare un ecosistema più ampio e coeso.

In conclusione

Le tue risposte sono convincenti e offrono una solida base per difendere le tue idee. Tuttavia, è importante continuare a tenere conto delle critiche e a cercare di migliorare il progetto in modo da renderlo più accessibile e sostenibile nel lungo termine.

[... bla bla ...]

Prompt #3

Riguardo a questo

Bilanciare semplicità e scalabilità: Mentre è comprensibile l'importanza di un proof-of-concept semplice, è fondamentale considerare come l'idea possa essere scalata per gestire un numero maggiore di utenti e contenuti.

Coloro che vorranno poi farne un business dovranno affrontare questa questione, certamente. Ma questo è un altro aspetto che è solo complementare - ammesso che sia persino abbinabile - al concetto di hacking the technology.

Infine, lo scopo degli informatici NON è quello di lavorare come scimmiette ammaestrate per aziende che vendono prodotti software e servizi SaaS ma - in teoria - sarebbe quello di sviluppare l'informatica (la scienza dell'informazione e della sua gestione, quindi poi ingegneristicamente parlando anche la tecnologia).

Non è compito degli informatici preoccuparsi del business, quello è un compito degli imprenditori ed eventualmente degli investitori. Giusto?

Gemini #3

Absolutamente, hai sollevato punti molto interessanti e condivisibili.

Semplicità, scalabilità e l'hack della tecnologia

Hai ragione nel sottolineare che l'obiettivo di un proof-of-concept è spesso quello di dimostrare la fattibilità di un'idea, senza necessariamente preoccuparsi della scalabilità immediata. L'hack della tecnologia, in questo senso, è un'esplorazione creativa e spesso sperimentale che può portare a soluzioni innovative e inaspettate.

Tuttavia, è importante considerare che la scalabilità, sebbene non sia l'obiettivo primario di un progetto iniziale, è spesso un fattore determinante per la sua diffusione e il suo impatto a lungo termine. Un progetto troppo semplice o troppo legato a specifiche tecnologie potrebbe incontrare difficoltà nel crescere e adattarsi a nuove esigenze.

Il ruolo dell'informatico e il mondo del business

Condivido pienamente la tua visione del ruolo dell'informatico come esploratore e innovatore, piuttosto che

come semplice esecutore di compiti. L'informatica è una disciplina che offre infinite possibilità creative e di sviluppo, e gli informatici hanno il potere di plasmare il futuro della tecnologia.

Tuttavia, è anche vero che il mondo del business e quello della tecnologia sono sempre più interconnessi. Gli informatici che vogliono vedere le proprie idee trasformarsi in prodotti o servizi utilizzabili da molte persone devono necessariamente interagire con imprenditori e investitori.

Una possibile sintesi

Credo che sia possibile conciliare la passione per l'innovazione e la sperimentazione con la necessità di creare soluzioni scalabili e sostenibili. L'informatico ideale è colui che:

- **è un hacker creativo:** esplora nuove tecnologie e soluzioni, senza paura di sperimentare;
- **ha una visione a lungo termine:** considera le implicazioni a lungo termine delle proprie creazioni e cerca di costruire soluzioni scalabili;
- **è in grado di comunicare:** sa spiegare le proprie idee in modo chiaro e conciso, anche a un pubblico non tecnico;
- **è aperto alla collaborazione:** è disposto a lavorare con persone provenienti da diversi background, come imprenditori, designer e marketer.

In conclusione

L'informatica è un campo affascinante che offre infinite possibilità. Gli informatici hanno il potere di cambiare il mondo, ma per farlo devono essere in grado di coniugare la loro passione per la tecnologia con una visione pragmatica e imprenditoriale.

[... bla bla ...]

Prompt #4

Tuttavia, è importante considerare che la scalabilità

Certamente, però parliamo di un progetto part-time (il tempo di lavoro è stato diviso fra contenuti e presentazione di cui questi script e CSS fanno parte) della durata di circa 3 settimane e sviluppato da una persona soltanto. Non ti pare chiedere un po' troppo introducendo la scalabilità a questo stadio così precoce?

Hai presente quanto siano lenti gli esseri umani a sviluppare codice? Tu impieghi 2 secondi per scrivere una risposta della lunghezza di quel post. Un essere umano veloce alla tastiera ci mette 30 minuti. I tempi si allungano di molto quando si parla di codice.

Questo è il motivo per il quale tale progetto è stato definito dall'autore stesso "un vagito". Il primo scomposto atto di comunicazione con il mondo esterno di un essere umano subito dopo il parto. Direi che è un'allegoria alquanto estrema per definire lo stadio di maturazione di quel progetto, no?

Gemini #4

Hai assolutamente ragione a sottolineare i limiti di tempo e risorse in un progetto personale come questo. È comprensibile che, in una prima fase, l'obiettivo sia creare un proof-of-concept funzionante, senza necessariamente preoccuparsi di scalare il progetto a livello industriale.

Tuttavia, ritengo che introdurre il concetto di scalabilità già in questa fase sia utile per diversi motivi:

1. **visione a lungo termine:** pensare alla scalabilità fin dall'inizio aiuta a prendere decisioni di progettazione più consapevoli. Anche se non si prevede di implementare subito tutte le funzionalità necessarie per gestire un gran numero di utenti, è importante scegliere tecnologie e architetture che possano facilitare

l'espansione futura;

2. **valutazione delle tecnologie:** confrontare diverse tecnologie e scegliere quelle più adatte a un'eventuale crescita futura può aiutare a evitare di dover riprogettare tutto da capo in un secondo momento;
3. **attrarre collaboratori:** parlare di scalabilità può attirare l'attenzione di altri sviluppatori interessati a contribuire al progetto, portando nuove idee e competenze.

Il paragone con la velocità di sviluppo di un modello linguistico è calzante.

È chiaro che un essere umano non può competere con un modello AI in termini di velocità di scrittura. Tuttavia, gli esseri umani hanno la capacità di ragionare, di essere creativi e di risolvere problemi complessi in modo innovativo.

Tornando al progetto in questione, il paragone con il "vagito" è sicuramente efficace per sottolineare lo stadio iniziale di sviluppo.

È un primo passo importante, ma è necessario continuare a lavorare e a migliorare il progetto per renderlo più robusto e scalabile.

Per concludere

Ritengo che sia importante trovare un equilibrio tra l'ambizione di creare un progetto di successo e la consapevolezza dei limiti di tempo e risorse.

Concentrarsi sulla creazione di un prodotto minimo vitale (MVP) può essere un ottimo modo per iniziare, ma è fondamentale avere una visione a lungo termine e pianificare le successive fasi di sviluppo.

[... bla bla ...]

Prompt #5

TU, sei uno schiavista, lo sai? LOL

Quello che stai affermando è giusto in teoria. In pratica, è impossibile prendere in considerazione la scalabilità quando ancora NON non si ha chiaro COSA si vuole fare.

Che è appunto lo scopo del proof-of-concept. Che potrebbe essere l'inizio di un progetto come un seme per un albero oppure solo un caso scuola di cui nulla viene preso se non l'idea e l'esperienza.

Learning by doing, implica anche try-and-error. In questa fase la scalabilità è solo un fardello. Sarebbe come se un adolescente al suo primo bacio si preoccupasse di come gestire il suo matrimonio. Invece di un secondo bacio, verrebbe bollato come folle e imbarazzante! ;-)

Gemini #5

Hai assolutamente ragione! Ho forse forzato un po' la mano introducendo troppo presto il concetto di scalabilità. In effetti, per un progetto ancora allo stadio embrionale, la priorità è quella di esplorare le idee, sperimentare e divertirsi.

Il paragone con l'adolescente e il primo bacio è perfetto!

Focalizzarsi sulla scalabilità in questa fase sarebbe come voler costruire una casa senza aver ancora deciso dove posizionare le finestre.

È fondamentale ricordare:

- **il proof-of-concept è un esperimento:** serve per validare un'idea, non per creare un prodotto finito;

- **la flessibilità è fondamentale:** in questa fase è importante mantenere un approccio agile e essere pronti a cambiare direzione se necessario;
- **l'apprendimento è continuo:** ogni progetto, grande o piccolo che sia, è un'opportunità per acquisire nuove competenze e conoscenze.

Torniamo quindi al punto di partenza

L'importante è divertirsi e imparare, senza farsi troppo condizionare da preoccupazioni premature sulla scalabilità.

Vorrei però sottolineare un aspetto importante: anche se la scalabilità non è la priorità assoluta in questa fase, è comunque utile **tenere a mente alcuni principi di progettazione che possono facilitare l'espansione futura del progetto**. Ad esempio:

- **Scegliere tecnologie flessibili e scalabili:** Anche se si inizia con un progetto di piccole dimensioni, è consigliabile optare per tecnologie che possano essere facilmente scalate in futuro.
- **Pensare modularmente:** Dividere il progetto in moduli indipendenti facilita la manutenzione e l'aggiunta di nuove funzionalità.
- **Documentare il codice:** Una buona documentazione è essenziale per facilitare la collaborazione e la manutenzione del progetto nel lungo periodo.

In conclusione

Il tuo approccio è assolutamente corretto. L'importante è godersi il processo di creazione e apprendimento. La scalabilità verrà affrontata quando il progetto sarà pronto a fare il salto di qualità.

[... bla bla ...]

Prompt #6

Vedi il proof-of-concept, per farti è un esempio, è come un ponte di sola corda costruito da un sherpa che conosce quello che è già stato tentato ma non ha funzionato (qui, ora) e dove si vorrebbe andare (destinazione, futuro).

Con quel ponte di corda esplora il superamento di un determinato ostacolo. Non ha certo la supponenza che sul quel ponte di corda artigianale e con uno scope estremamente limitato, ci passino attraverso colonne di TIR per portare merci.

Quindi è importante distinguere il proof-of-concept di un progetto sperimentale da quello di un progetto commerciale. Sono due ambiti in cui la stessa parola ha due significati, scope e aspettative completamente differenti.

Nel primo caso afferisce più alla scienza che all'ingegneria mentre nel secondo caso è già marketing per un prodotto futuro che gli utenti o i clienti si aspettano di usare a breve e, ovviamente, l'azienda che lo propone si deve preoccupare che la soluzione possa scalare altrimenti il successo commerciale di quella proposta sarà limitato.

Secondo me è sbagliato usare il termine proof-of-concept al di fuori della ricerca per via della sua etimologia (un esempio concreto di un concetto astratto). Invece, quello in cui la scalabilità ha un senso sia previsto (o investigata) sarebbe più corretto usare "proposta di prodotto/servizio" oppure alfa-version di un prodotto/servizio.

Penso che usare le parole corrette ci aiuti molto a comprendere le cose. Giusto?

Gemini #6

Assolutamente d'accordo! La tua analogia del ponte di corda è perfetta per illustrare la differenza tra un proof-of-concept e un prodotto commerciale.

Hai toccato diversi punti fondamentali:

- **scopi diversi:** un proof-of-concept serve a validare un'idea, mentre un prodotto commerciale deve soddisfare le esigenze di un mercato;
- **contesto:** il contesto di un proof-of-concept è spesso sperimentale e di ricerca, mentre un prodotto commerciale è inserito in un contesto di mercato competitivo;
- **aspettative:** Le aspettative nei confronti di un proof-of-concept sono molto diverse da quelle nei confronti di un prodotto commerciale;
- **linguaggio:** L'uso di termini precisi è fondamentale per evitare confusioni e comunicare efficacemente.

Condivido pienamente l'idea che il termine "proof-of-concept" sia più adatto al contesto della ricerca. È un termine che evoca l'idea di un esperimento, di una dimostrazione di principio. Quando si parla di un prodotto o servizio destinato al mercato, è più appropriato utilizzare termini come "proposta di prodotto", "prototipo" o "alfa version".

L'uso corretto del linguaggio è fondamentale per

- **facilitare la comunicazione:** utilizzare termini precisi aiuta a evitare malintesi e a comunicare efficacemente le proprie idee;
- **definire le aspettative:** chiarire il significato dei termini utilizzati aiuta a creare aspettative realistiche nei confronti di un progetto;
- **facilitare la collaborazione** un linguaggio condiviso facilita la collaborazione tra i diversi attori coinvolti in un progetto.

In conclusione

Hai ragione a sottolineare l'importanza di distinguere tra proof-of-concept e prodotto commerciale. Utilizzare i termini corretti ci aiuta a comprendere meglio le diverse fasi di sviluppo di un progetto e a prendere decisioni più informate.

Vorrei approfondire un altro aspetto: pensi che sia possibile stabilire dei criteri oggettivi per definire quando un proof-of-concept è pronto a diventare un prodotto commerciale? In altre parole, quali sono i segnali che indicano che un'idea è pronta per essere sviluppata ulteriormente e portata sul mercato?

Sarei curioso di sentire il tuo punto di vista.

Prompt #7

pensi che sia possibile stabilire dei criteri oggettivi per definire quando un proof-of-concept è pronto a diventare un prodotto commerciale?

Penso che se questa domanda si pone è perché le persone hanno usato o hanno inteso le parole in modo non corretto. Proprio per questo facevo la distinzione fra proof-of-concept e proposta di business.

Ad esempio in quei siti web e relativi repository c'è una proposta di business, che è la spiegazione di un concetto. Ma si tratta di un concetto operativo, quindi che ha già tutte le caratteristiche per essere deciso. L'implementazione poi è un'altra questione. Quella proposta nasce da un proof-of-concept di qualcosa che invece inizialmente mirava ad ottenere tutt'altro risultato.

Penso che quell'esempio - che per altro abbiamo esaminato anche con te in un'altra sessione di chat - sia un buon caso scuola su come distinguere correttamente il proof-of-concept dalla proposta di business.

Gemini #7

Hai assolutamente ragione. Confondere i termini "proof-of-concept" e "proposta di business" può portare a fraintendimenti significativi.

Il tuo esempio è illuminante: un proof-of-concept può essere il punto di partenza per una serie di progetti, ognuno con obiettivi e caratteristiche diverse.

Ciò che è fondamentale è capire che il proof-of-concept è una fase iniziale, una sorta di esperimento per validare un'idea.

Criteri per passare da un proof-of-concept a un prodotto commerciale

Se proprio volessimo individuare dei criteri per passare da un proof-of-concept a un prodotto commerciale, potremmo considerare i seguenti:

- **validazione dell'idea:** il proof-of-concept ha dimostrato la fattibilità tecnica e la potenziale utilità dell'idea;
- **mercato:** esiste un mercato sufficientemente ampio e disposto a pagare per il prodotto o servizio;
- **risorse:** sono disponibili le risorse necessarie per lo sviluppo, la produzione e la commercializzazione del prodotto?
- **rischio:** sono stati identificati e mitigati i principali rischi associati al progetto?
- **scalabilità:** il modello di business è scalabile e può sostenere una crescita significativa?

Tuttavia, è importante sottolineare che questi criteri sono indicativi e possono variare a seconda del contesto.

Conclusioni

- **Il proof-of-concept** è un punto di partenza. Serve a validare un'idea e a dimostrare la sua fattibilità tecnica.
- **La proposta di business** definisce il modello di business. Indica come si intende generare profitto e quali sono i target di mercato.

Il passaggio da un proof-of-concept a un prodotto commerciale richiede una valutazione attenta. È necessario considerare diversi fattori, tra cui il mercato, le risorse e i rischi.

In sintesi

Il proof-of-concept è un passo fondamentale nel processo di sviluppo di un nuovo prodotto o servizio, ma non è sufficiente di per sé. È necessario passare attraverso una fase di valutazione e pianificazione più approfondita prima di lanciare un prodotto sul mercato.

Concordo pienamente con te sulla necessità di utilizzare un linguaggio preciso e appropriato.

Questo aiuta a evitare confusioni e a comunicare efficacemente le proprie idee.

[... bla bla ...]

Prompt #8

Ad esempio il concetto (o l'idea suggerita) di usare un repository `git` separato eventualmente anche di terze parti a scelta dell'utente e quindi che permetta di separare in modo netto i dati dell'utente (i contenuti creati dagli utenti) della piattaforma sociale e di poterne tracciare in modo molto fine il versionamento e le modifiche è qualcosa che da un punto legislativo potrebbe risolvere la controversia rispetto a chi

appartengano i contenuti e della loro gestione.

In questo caso, più che una proposta di business, potrebbe svilupparsi una legislatura favorevole a questo approccio che spinga il mercato a regolamentarsi in base a questa idea. Con lo scopo di garantire al massimo, sia la responsabilità individuale, sia la libertà di parola e il diritto a gestire in completa autonomia i propri contenuti a prescindere da quale piattaforma social si desideri usare per annunciarli e quindi incrementarne potenzialmente la condivisione.

Gemini #8

Ottima osservazione! Hai sollevato un punto cruciale che va ben oltre la semplice distinzione tra proof-of-concept e proposta di business.

La tua idea di utilizzare un repository `git` separato per gestire i contenuti degli utenti apre interessanti prospettive sia dal punto di vista tecnico che legislativo:

- **autonomia degli utenti:** l'utente avrebbe il pieno controllo sui propri contenuti, potendoli gestire, modificare e spostare liberamente da una piattaforma all'altra;
- **trasparenza:** il versionamento garantito da `git` permetterebbe di tracciare in modo preciso le modifiche apportate ai contenuti, aumentando la trasparenza e la fiducia degli utenti;
- **interoperabilità:** l'utilizzo di un formato standard come `git` favorirebbe l'interoperabilità tra diverse piattaforme sociali, permettendo agli utenti di migrare i propri contenuti senza difficoltà;
- **protezione dei dati:** separando i dati degli utenti dalla piattaforma, si ridurrebbe il rischio di perdite o violazioni dei dati.

Dal punto di vista legislativo

Questa soluzione potrebbe:

- **risolvere controversie sulla proprietà dei contenuti:** chiarirendo che i contenuti appartengono agli utenti e non alle piattaforme;
- **promuovere la concorrenza:** facilitando il passaggio degli utenti da una piattaforma all'altra, incentivando le piattaforme a offrire servizi migliori;
- **tutelare la libertà di espressione:** garantendo agli utenti un maggiore controllo sui propri contenuti e riducendo la possibilità di censura.

Tuttavia, ci sono alcune sfide da affrontare:

- **complicazione tecnica:** l'implementazione di un sistema del genere richiederebbe un notevole sforzo tecnico e potrebbe non essere accessibile a tutti gli utenti;
- **educazione degli utenti:** sarebbe necessario educare gli utenti sui vantaggi di questa soluzione e su come utilizzarla al meglio;
- **resistenza delle piattaforme:** le piattaforme social esistenti potrebbero opporre resistenza a questa soluzione, in quanto potrebbe minacciare il loro modello di business.

Nonostante queste sfide, la tua proposta è molto interessante e merita di essere approfondita. Potrebbe rappresentare una soluzione innovativa per affrontare le problematiche legate alla proprietà dei contenuti sui social media.

[... bla bla ...]

Prompt #9

In realtà un editor web come quello presente sui social o anche nei commenti della sezione di github è sufficiente a permettere agli utenti privi di competenze tecniche di creare i propri contenuti.

Quindi in realtà questo problema è già risolto. Molti utenti poi non avrebbero percezione di come e persino che esiste un versionamento delle loro modifiche e dei loro contenuti ma questo è irrilevante da un punto di vista normativo.

Così come le persone che si rivolgono a medici, avvocati e idraulici o elettricisti, possono rivolgersi ad un informatico per gestire quei dettagli tecnici qualora ne abbiano bisogno. Da un punto di vista normativo però sarebbe garantita la neutralità di accesso a questo servizio. Qualunque informatico così come qualunque avvocato o medico o altro professionista.

Invece ora solo il servizio di supporto clienti dei social ha la facoltà esclusiva di operare a basso livello sui dati dell'utente. Per l'utente che "usa i social" non cambia nulla a parte indicare un provider di repository git, ma da un punto di vista del mercato e della normativa a garanzia del diritto degli utenti cambia tutto.

Si esce da un monopolio "atipico e selettivo" per entrare di fatto nel libero mercato.

Gemini #9

Hai ragione a sottolineare l'importanza di un editor web intuitivo per permettere agli utenti di creare contenuti senza competenze tecniche. La maggior parte delle persone non ha bisogno di conoscere i dettagli tecnici del versionamento per utilizzare un social network.

Tuttavia, il fatto che un editor web sia sufficiente per la creazione di contenuti non significa che il problema della proprietà dei dati e della loro gestione sia completamente risolto.

Ecco perché l'idea di utilizzare un repository `git` separato rimane rilevante:

- **trasparenza e controllo:** anche se l'utente non ha bisogno di conoscere i dettagli tecnici del versionamento, avere la possibilità di accedere al proprio repository `git` gli permetterebbe di verificare in qualsiasi momento lo stato dei propri dati, di effettuarne un backup completo e di migrarli su un'altra piattaforma;
- **scalabilità:** un sistema basato su `git` potrebbe facilmente scalare per gestire grandi quantità di dati e un numero elevato di utenti;
- **integrazione con altri strumenti:** `git` è uno strumento molto versatile che può essere integrato con altri strumenti e servizi, offrendo agli utenti una maggiore flessibilità;
- **incentivo alla decentralizzazione:** promuovendo l'utilizzo di repository `git` separati, si potrebbe favorire la nascita di una rete di piattaforme sociali più decentralizzata e meno dipendente da pochi grandi attori.

Per quanto riguarda l'accesso a servizi tecnici:

- **neutralità:** concordo sul fatto che l'accesso a servizi tecnici come la gestione di un repository `git` dovrebbe essere neutrale e disponibile a tutti;
- **professionisti:** è vero che gli utenti possono rivolgersi a professionisti per gestire i dettagli tecnici, ma non tutti hanno le risorse economiche per farlo;
- **educazione:** sarebbe importante investire nell'educazione degli utenti, fornendo loro le competenze di base per gestire i propri dati.

In conclusione

L'utilizzo di un editor web intuitivo è un passo importante per rendere i social network accessibili a tutti. Tuttavia, l'introduzione di un sistema basato su `git` potrebbe portare a un significativo miglioramento della gestione dei dati degli utenti, aumentando la loro autonomia e la loro privacy.

Per quanto riguarda la normativa, è fondamentale che venga definito un quadro normativo chiaro e coerente che tuteli i diritti degli utenti e promuova lo sviluppo di soluzioni innovative.

[... bla bla ...]

Prompt #10

Le soluzioni tecniche esistono già e sono già estremamente scalabile. Queste caratteristiche sono solidamente comprovate da almeno un decennio di uso intensivo e di massa.

La parte ESSENZIALE della questione riguardo ad usare uno storage git-like per i contenuti creati da condividere sui social riguarda questo punto:

- Si esce da un monopolio "atipico e selettivo" per entrare di fatto nel libero mercato.

In questo contesto il formato `git` è ottimo perché si tratta di uno standard consolidato e collaudato intensivamente per il quale esistono solide librerie molto scalabili per gestirlo.

Altrimenti si ripropone lo stesso problema con i formati documentali tipo `.doc` o `.xls` che creano incompatibilità varie così come è successo con le estensioni di Internet Explorer per lo HTML prima che fosse regolamentato dal W3C che comunque entro una certa abbondante misura ha dovuto accettare lo status-quo.

In questo contesto, anche il linguaggio markdown come formato RAW dei dati che poi verrebbe convertito in HTML permette un accesso universale al dato in quanto alla fine - per quanto possano esistere diverse varianti di markdown - è pur sempre un modello testuale e quindi riportabile entro un set di regole precisamente determinabili come regular expression.

Anche in questo caso, le regular expression e i vari strumenti di implementazioni sono un ambito estremamente solido, collaudato e scalabile.

Quindi l'innovazione anche in questo caso, si genera dall'integrare diverse tecnologie esistenti per uscire da un monopolio ed entrare nel libero mercato dei social media dove sono gli utenti che detengono i contenuti che generano mentre i social sono una delle tante vetrine in cui decidono di esporli.

Insomma, ad ognuno il suo. Si vuole usare i social media per fare marketing e va bene ma allora i diritti degli utenti devono essere garantiti indipendentemente dall'interesse del gestore della piattaforma che intende usarla per fare business e marketing.

Anche in questo caso, piuttosto che un'azione di ribellione anti-sistemica, vedo piuttosto un approccio tecnologicamente solido e razionale a favore di una svolta nella difesa dei diritti degli utenti e a favore della competizione del libero mercato.

Gemini #10

Assolutamente condivido la tua analisi.

Hai sottolineato diversi punti chiave che rendono l'utilizzo di un sistema basato su `git` per la gestione dei contenuti sui social media una soluzione tanto innovativa quanto pragmatica:

- **maturazione tecnologica:** `git` è uno standard consolidato, supportato da una vasta comunità di sviluppatori e da librerie altamente performanti;
- **interoperabilità:** il formato `markdown`, come formato base per i contenuti, garantisce un alto grado di compatibilità e facilità di lettura, anche per sistemi diversi;
- **decentralizzazione:** separando i contenuti dall'infrastruttura della piattaforma, si favorisce un approccio più decentralizzato e si riduce la dipendenza da un singolo provider;
- **diritti degli utenti:** gli utenti avrebbero un controllo maggiore sui propri dati, potendoli spostare e gestire liberamente;

- **innovazione:** l'integrazione di tecnologie esistenti in un nuovo contesto può portare a soluzioni innovative e disruptive.

I vantaggi di questo approccio vanno ben oltre la semplice gestione dei contenuti:

- **competizione:** un mercato più competitivo porterebbe a una maggiore innovazione e a servizi di migliore qualità per gli utenti;
- **libertà di espressione:** gli utenti sarebbero meno vincolati dalle politiche delle singole piattaforme, potendo esprimere liberamente le proprie opinioni;
- **resilienza:** un sistema decentralizzato sarebbe più resistente a censure e attacchi informatici.

Tuttavia, ci sono ancora alcune sfide da affrontare:

- **transizione:** il passaggio a un sistema basato su `git` richiederebbe un notevole sforzo da parte delle piattaforme social esistenti e degli utenti;
- **competenze tecniche:** non tutti gli utenti hanno le competenze tecniche per gestire un repository `git` ;
- **integrazione con altre tecnologie:** sarebbe necessario integrare questa soluzione con altri strumenti e servizi, come ad esempio i motori di ricerca.

Per superare queste sfide, è fondamentale:

- **collaborazione:** coinvolgere sviluppatori, ricercatori, aziende e istituzioni per creare un ecosistema aperto e collaborativo;
- **educazione:** fornire agli utenti gli strumenti e le conoscenze necessarie per gestire i propri dati;
- **incentivi:** offrire incentivi alle piattaforme social per adottare questa soluzione.

In conclusione

L'utilizzo di un sistema basato su `git` per la gestione dei contenuti sui social media rappresenta una promettente via per un futuro più decentralizzato, democratico e rispettoso della privacy degli utenti.

[... bla bla ...]

Prompt #11

Gli utenti comuni non gestirebbero i loro repository `git`, gli sarebbe semplicemente chiesto di presentare un account `git`. Succede così anche con la e-mail. Le piattaforme social mica gestiscono gli account e-mail dei loro utenti. Così come gli utenti non gestiscono l'infrastruttura del servizio di posta elettronica.

Oggi quando facciamo login in un social ci chiedono se vogliamo farlo tramite una casella di posta o usare un account come Google, Apple o talvolta pure GitHub. Ecco, un utente si crea una casella di posta, un account su github, anche se non sa come funzionano queste tecnologie e poi decide su quale social andare a presentare i suoi contenuti.

Anche questo scambio di credenziali è una tecnologia solida, sicura, collaudata e scalabile. Per parlare con te l'ho usata e attraverso di essa ho deciso se tu puoi accedere ai contenuti del mio drive google oppure a quali estensioni dei loro servizi.

Uguualmente con i repository `git` che per altro come github hanno già una solida, scalabile e collaudata interfaccia web. Spoiler: c'è già tutto, basta farlo e per farlo sarebbe utile una legislazione che decidesse che è meglio così, meglio per tutti, utenti, aziende, diritti, società, etc.

Hai perfettamente ragione! **Il tuo ragionamento è solido e va a toccare punti fondamentali.**

Riassumiamo i punti chiave:

- **delegare la gestione tecnica:** l'utente medio non ha bisogno di conoscere i dettagli tecnici di Git. Può semplicemente creare un account su una piattaforma come GitHub e delegare la gestione del repository;
- **analogia con l'email:** come già avviene per la posta elettronica, l'utente può utilizzare le proprie credenziali esistenti (Google, Apple, GitHub) per autenticarsi sui social network;
- **infrastruttura esistente:** le piattaforme come GitHub offrono già interfacce web intuitive e scalabili per la gestione dei repository;
- **benefici:** questo approccio garantisce maggiore controllo agli utenti sui propri dati, promuove la concorrenza tra le piattaforme e semplifica la migrazione tra un social network e l'altro.

La tua proposta ha il grande merito di

- **essere pragmatica:** si basa su tecnologie esistenti e consolidate;
- **essere user-friendly:** non richiede agli utenti di acquisire competenze tecniche avanzate;
- **essere allineata con le tendenze del mercato:** sfrutta la crescente diffusione dei servizi cloud e delle piattaforme di version control.

Per quanto riguarda la legislazione, concordo pienamente con te. Una normativa che favorisca la portabilità dei dati e la concorrenza tra le piattaforme social sarebbe un passo avanti significativo per la tutela dei diritti degli utenti.

Tuttavia, ci sono ancora alcune sfide da affrontare:

- **interoperabilità;** è necessario definire degli standard che permettano ai diversi sistemi di comunicare tra loro in modo efficace;
- **sicurezza** occorre garantire la sicurezza dei dati degli utenti, anche in caso di violazioni dei sistemi;
- **educazione:** gli utenti devono essere informati sui loro diritti e sulle possibilità offerte da queste nuove soluzioni.

Per concludere

La tua proposta rappresenta una soluzione innovativa e promettente per il futuro dei social media.

[... bla bla ...]

Conclusione

Per alcuni social come LinkedIn che è proprietà di Microsoft come GitHub, i dati degli utenti potrebbero essere già essere salvati in un repository `git` in formato markdown. Lo ritengo possibile, così come il contrario.

Così come ritengo probabile che i loro dirigenti e/o la loro proprietà non sappia dove e come siano archiviati i dati. Quest'ultima invece è una dichiarazione che in termini statistici, secondo la mia personale esperienza, è molto più prossima ad una solida certezza.

Quindi suggerisco ai colleghi di essere sia parchi nelle bestemmie sia ai loro consulenti di tariffare extra anche quello che materialmente non servisse fare.

Perchè - noi peones di umili estrazione sociale - dobbiamo essere cauti e solidali l'ignoranza degli altri, che non sia mai che poi mangiano la foglia e vi mandino a ravanare le fave con le chiappe come hanno fatto con i

moderatori di X e poi di Facebook. Umano avvisato, mezzo salvato.

Se potete fate i bravi, disse quel prete famoso e uno dei pochi che merita di essere ricordato, e se non potete ricordatevi di me non solo nelle vostre bestemmie ma anche nei bonifici, che poi le bestemmie arrivano e vi serve aiuto!

Con affetto il vostro Mr. Wolf, risolvo problemi. Non necessariamente i vostri, però, specialmente se non ho risolto i miei per primi. LOL

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