

Thoughts on Gemini

tropf

ABSTRACT

I think we have to severely transform the world wide web, and I don't think Gemini is the way to do that.

1. Preliminaries

1.1. The Problem with the WWW

The world wide web currently is a mix of different, very complicated standards. A lot of its features are unnecessary and even harmful.

This complexity, after the downfall of mozilla, has lead to Google with chromium, or more precisely, the blink engine deciding alone how the web is viewed and displayed. Which in turn leads to further problems, ultimately moving the power to shape the world wide web away from its citizens.

› There are **far** more problems with the world wide web.

1.2. Gemini

Gemini (<https://gemini.circumlunar.space/>) is a standard and a text format (`text/gemini`) to solve a proper subset of problems that the web solves, yet being much simpler.

2. Faulty Design Principles

Gemini, according to its FAQ (<https://gemini.circumlunar.space/docs/faq.html>), is designed with simplicity, privacy and generality in mind. Those are goals that every engineer should strive for, imho. Though it seems like those are followed more like a religion, and less like guiding ideas that have their shortcomings clearly addressed and considered.

Also it feels like there are more subtle core ideas that are not spelled out as such, like an aversion towards applications on the (or for that matter, any) web or the strong focus on operators rather than users.

2.1. Target Audience

Gemini is clearly targeted at programmers, or at least a technical community.

I think this is a mistake: The information-centric, fast and privacy-preserving web that Gemini would create has a benefit to a much larger group of people. As engineers we should strive for solutions that help all people, not just ourselves.

This is not just a "religious belief" which I just complained about, but a concept for bilateral good: The engineers of the 90s may have built the web, but a much more diverse community of the 00s filled it with content.

I don't think that there shouldn't be *techie only* solutions; I am a strong proponent of the shell, vim and alike; but those have alternatives with a lower barrier of entry (e.g. Notepad or Libreoffice), and they are the backbone of many people's work everyday. Gemini is not.

2.2. The Meaning of "Simplicity"

Gemini understands simplicity as "can be implemented by one person over one weekend". I understand that as a (quasi-) "Hackathon".

Instead of defining simplicity by an absolute measure, I'd call something "simple" if it achieves much by doing little.

› The termin "Design" is often defined in exactly the same way,

The point is that the former asks "What can we achieve?" first, while the latter puts "What do we actually need?" in focus—which is the more important question.

3. Missing Features

3.1. text/gemini

The `text/gemini` format is extraordinarily simple. Too simple, as it misses features to meaningfully express content.

3.1.1. Inline Images

As far as I read the mailinglist, it seems that most users see this as an advantage. I don't.

Images are one of the most powerful ways to present information, much faster and better than text.

Of course, one could argue that you could just link to an image from `text/gemini`. But unarguably that makes navigating much more complicated, too complicated in my opinion. And that's not even considering reading text and viewing an image simultaneously.

3.1.2. Tables

If you can't use images, use tables. If you browse around my other articles you'll see that I love using tables. They are simple to create, simple to understand and much better than a wall of text. And Gemini does not have them.

3.1.3. Font Styles

Gemini misses font styles. While I don't care about the font family used, I can't understand why you'd like to miss out on such a simple features as *emphasis*.

3.2. The Protocol

There are technologies that are tried and tested for Gemini-like applications, yet they are not employed. One could claim "simplicity" but I don't think that counts, as the benefits provided clearly outweigh the complexity introduced imho.

3.2.1. Compression

Gemini transfers text. Text is one of the best researched type of content and can be compressed very well, often by over 50%.

3.2.2. Keepalive

Gemini mandates a new connection for each document served. As the typical "browsing" of documents involves many connections to the same server I don't see why not to reuse the already established connection. Sure, signalling EOF but keeping the connection open is not trivial but not unsolvable either.

The problem lies in the cost of establishing a connection. Gemini (practically) involves both a TCP as well as a TLS handshake. This can take up to 3 RTTs (plain TCP+TLSv1.2), to then serve a couple of kilobytes in the worst case which takes an additional RTT. Of course, using TLSv1.3 and TCP Fast Open the overhead can be reduced to a single RTT, but both are far from the norm.

3.2.3. Sessions

Gemini does not have sessions. Except it does: It is trivial to include a kind of session cookie in the url. In fact, I already used a similar mechanism to avoid using cookies on the web already.

In my opinion Gemini should embrace sessions and add a possibility to provide a session cookie by the client. This way, the client would control the session. Now the servers can (and will) do some non-standardized tinkering to implement sessions and we have the problem of the world wide web all over again.

4. Creating Content

I think creating content for Gemini is too difficult. You can either write in `text/gemini` straight away and you can't use the features of other formats (and therefore not really reuse your content elsewhere) or you can use some other, better suited format but then fail to be presented in all clients.

Currently, the tools to transform content from and to `text/gemini` are not available. While I think they will make creating content easier, they won't be able to solve this problem.

5. Meta

5.1. Designing a Protocol

I really like the approach of the IETF to designing protocols: There are mailinglists to discuss and working groups to decide. At the same time, anyone can publish drafts and RFCs at any time.

This way there is room for discussion and a way to find a consensus, while not needing dictatorial power yet providing the individual with plenty of freedom. To not use this infrastructure—both in terms of hardware, software and organizational infrastructure, but also the concepts behind those—is a bad idea.

5.2. Important new Features

Gemini introduces a lot of new features that are very welcome and, imo, needed urgently. Among those a step away from password based authentication, a focus on content and less on representation and none on client-side-scripting or a concept to meaningfully use self-signed-certificates.

But one has to acknowledge that all these new ideas automatically make a wide adoption less likely. Considering that, I don't understand why those new, difficult-to-implement ideas have to be mixed with a horrible usability.

5.3. What does Gemini want to be?

Most of my criticism here revolves around the question what gemini wants to be. I think that vision has not been clearly expressed yet and I hope it will in the future. Currently I see Gemini as a sort of hope for all those critical of the current web, while it can't possibly fulfill all these hopes.

I'd welcome the protocol undergoing some serious reconsideration. Also I think that `text/gemini` should stay, but be complemented by a more complex alternative, maybe simply Markdown.