Engineer

Gemini

a lightweight internet protocol for serving hypertext

What it is

Gemini is an application protocol for fetching documents in which the whole exchange is one encrypted connection, one address sent by the client, one response sent back and the connection closed. Its own documents call it the Gemini protocol; its addresses begin with `gemini://`, a site served over it is a capsule, and the network of such sites is Geminispace.

The project started in June 2019 under its founder, Solderpunk, who remains its "Benevolent Dictator". Its specification, version 0.24.1, is dedicated to the public domain under CC0 and describes the protocol as "an incremental improvement over Gopher [RFC1436] rather than a stripped down HTTP". The project's FAQ makes the comparison from the other side: the protocol "looks kind of like something between HTTP 0.9 and HTTP 1.0", with a single kind of request, roughly a `GET`.

A client connects to TCP port 1965, a number that marks the first crewed Gemini mission, flown in March 1965, and completes a TLS handshake; both sides must support TLS 1.2 or higher. It sends the absolute URL followed by a carriage return and a line feed, and that URL may be at most 1,024 bytes long. The server answers with a single header line: a two-digit status code, a space and a short piece of text whose meaning depends on the code. The codes fall into six groups: 10–19 ask for input, 20–29 mean success, 30–39 redirect, 40–49 are temporary failures, 50–59 are permanent ones and 60–69 concern client certificates. On success the text is a MIME type and the body follows as raw bytes, with no compression and no chunking. The server then closes the connection with TLS's close_notify message, which is how the client knows it has everything.

It can carry any kind of file, though it was designed alongside a document format of its own, gemtext, served as `text/gemini`.

Use the concept in a decision

Put “What is Gemini?” to work in a real decision. We can help relate the concept to the system you are building or buying. A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Developed 100 web applications using HTML/HTML5 — 100 web applications built

Developed 100 web applications using HTML/HTML5 Get in touch →

Why it matters

The real decision is rarely whether to move a site to Gemini. The FAQ says the protocol is not intended to replace the web or Gopher, and it has a name for publishing the same content on two or three of them: bihosting or trihosting. The owner’s question is whether a second front door is worth keeping open.

What it offers is privacy by construction. The FAQ states that Gemini “does not and never will contain an equivalent of the Cookie, Referer or User-Agent headers”. A request is a URL and nothing else, so a server has very little to record even if its operator wanted to, and there is no script in the reader’s client to report back. The FAQ names “defensive non-extensibility” among the priorities behind the design: the request and response formats have no room for a new header field, so a later version has no place to bring tracking back in. For a publisher of plain text, that is the whole appeal.

What it costs is much of what the web has grown since HTTP 1.0. There are no cookies, so a capsule that has to recognise a returning reader asks for a client certificate instead, and the reader has to create and keep one. There is no certificate authority by default: the specification strongly recommends Trust on First Use, or TOFU, in which a client accepts whatever certificate a server presents the first time and pins it afterwards, which turns the server’s certificate into a long-lived promise. Encryption hides the page but the server’s name travels in TLS’s Server Name Indication, and the FAQ admits that the hostname “is currently still leaked”. Readers need a dedicated Gemini client rather than a browser, so the audience is small. And large files are a poor fit, because a response carries neither a size nor a checksum.

A team that needs interactive forms, accounts with passwords, analytics or rich layout belongs on the web. A team that publishes documents and wants a channel without cookies or client-side scripts can add Gemini at little cost, as long

as the content already exists as files.

What it looks like in practice

engineer.company is itself a capsule, at `gemini://engineer.company` . The Credits page names Agate as the program that answers Gemini requests, beside Caddy for the web, Gophernicus for Gopher and Tor for the onion service. All four read the same folder of files, so one deploy updates all four and none of them can drift from the others. That is the FAQ's bihosting carried to four channels, done in the cheapest way there is: the capsule is one more reader of a folder every deploy already fills, not a second site to keep in step.

The same page is frank about the limit described above. It says that Gemini and Gopher each want their own client rather than a browser, and that Gopher carries which page you read in the clear. Gemini encrypts that, but the network you read from still sees which address you asked for. That is the server-name leak from the FAQ, described from the reader's chair rather than the designer's.

The Privacy page distinguishes application access logs from operating-system logs. The Gemini and Gopher applications keep no access log; since September 2026, their daily counters record only aggregate request totals. Gopher also has an operating-system connection log containing the source address; that log expires after 30 days.

Where it goes wrong

Most Gemini trouble shows on the first line of a response, and that line is easy to read by hand with nothing but OpenSSL:

```
printf 'gemini://engineer.company/\r\n' |
  openssl s_client -quiet -connect engineer.company:1965 \
    -servername engineer.company
```

The first line that comes back is the header, `20 text/gemini;lang=en` for this capsule, and the page follows it.

Returning readers are warned that the capsule may be under attack. Clients that follow the specification save the fingerprint and expiry date of the first certificate they see. A different certificate before that date has passed counts as "potential evidence of a Man-In-The-Middle attack". An operator who generates a fresh certificate on every deploy, or moves the capsule to a new machine without its key, sets off that warning for every regular reader. Back up the key and certificate with the rest of the server, issue the certificate for a long period, and replace it only after the expiry date that clients already have on file.

A search box answers with status 59. A status 10 response asks the reader for a line of input, and the client sends it back as the query part of the same URL, percent-encoded. The URL is capped at 1,024 bytes, and a server must reject anything longer as a bad request. A long answer fills that quickly, and text outside ASCII fills it faster: a Cyrillic letter is two bytes of UTF-8 and six once encoded. Ask for short answers, and use status 11 for sensitive input such as a passphrase, so that clients do not echo it to the screen.

Readers are signed out between pages. The protocol has no cookies, so session state kept the web way does not survive from one request to the next. A capsule that needs to know who is asking returns status 60, the reader's client creates a certificate, and the server recognises it on later requests. The specification limits that certificate to the host and port that asked for it and to the requested path plus everything below it, so put every page that needs the identity under one path. Say on the page what the certificate is for, because the reader has to agree to it.

A large download arrives cut short and nobody notices. A response carries no length and no checksum, so a transfer that stops halfway can look much like one that finished, apart from the missing close_notify. The best-practices guide suggests that files larger than 100MiB are best served some other way. Put large files on HTTPS and link to them from the capsule; a Gemini page may link to any URL scheme.

Further reading

- Gemini network protocol specification — requests, every status code, the use of TLS and certificate pinning, in one short document.
- Project Gemini FAQ — the project's aims, its comparison with HTTP and Gopher, and its own answer to the charge that TOFU is weak.

- Best practices for Gemini implementations — file naming, large files, redirect limits and TLS settings.
- History of Project Gemini — how a 2019 idea from Gopherspace grew into a network of its own, told by the project.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is Gemini?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/gemini/