Engineer

Engineer in a terminal

Read engineer.company from the command line: Markdown over HTTP, a Gemini capsule, a Gopher hole and an onion service — the same site, four ways.

This site answers on four protocols, out of one folder of files on one server. If you live in a terminal, you never have to open a browser to read it.

The public site formats below need no account, key or client you have to build. Source repositories are private; the SSH section explains their access boundary.

Reach beyond the browser

Make useful information accessible to people and tools outside a graphical browser.

Explore our services Get in touch →

Inspect the text interfaces

Examine the text interfaces here as examples of one source reaching several clients.

Cases Get in touch →

Every page is also Markdown

Each page has an HTML version and clean Markdown beside it. Ask for either.

```
# The twin, by name
curl -sL https://engineer.company/about/index.md

# Or ask the server for it and let it choose
curl -sL -H 'Accept: text/markdown' https://engineer.company/about/
```

Pipe it into whatever renders Markdown for you:

```
curl -sL https://engineer.company/portfolio/index.md | glow -
```

```
curl -sL https://engineer.company/what-it-costs/index.md | bat -l md
```

The other languages work the same way — `/da/about/index.md` , `/uk/about/index.md` .

Beside each page's QR code, the format links also offer JSON, gemtext and a Gopher menu. The JSON at `/about/index.json` contains that page's metadata and Markdown content; `/feed.json` is the separate portfolio feed. Request `index.json` directly: `Accept: application/json` does not select it from the HTML URL.

View PDF opens the page's PDF in a new tab; **Download PDF** offers the same file for download. These static files are generated before publication, using the page's print layout. The links use native browser controls and add no browser JavaScript.

A Gemini capsule

Gemini is a small protocol with the sensibilities of this site built in: text documents over TLS, no scripting, no cookies, nothing to track you with. Every page of this site exists as gemtext too.

```
gemini://engineer.company/
```

Any Gemini client reads it — **Amfora**, **Lagrange** or **Bombadillo** in a terminal, **Castor** or **Kristall** in a window. On a Mac with Homebrew:

```
brew install amfora && amfora gemini://engineer.company/
```

Content that can leave **the browser it was written for** will outlive the framework that built it.

We build systems the same way: the data is the asset, the interface is replaceable.

A Gopher hole

Gopher is a menu protocol from 1991 — plain TCP, no encryption, no scripting — and the site is a set of menus there, one per directory.

```
gopher://engineer.company/
```

Lynx speaks it out of the box, as do **Bombadillo** and **gopher-cli**. Or ask for a selector by hand, which is the entire protocol in one line:

```
printf '/contact\r\n' | nc engineer.company 70
```

An onion service

The same site over Tor, for a reader whose network should not know what they are reading:

```
http://engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion/
```

It is plain `http://` deliberately: an onion address encodes the service's identity public key. Tor authenticates the service and encrypts the connection before HTTP is exchanged. With Tor running locally, `curl` reaches it directly:

```
curl -sL --socks5-hostname 127.0.0.1:9050 \
  http://engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion/index.md
```

Reading the work as data

The portfolio is syndicated in three formats, each carrying the full text of every item rather than a teaser:

```
curl -s https://engineer.company/feed.json | jq -r '.items[] | .title'
curl -s https://engineer.company/index.xml     # RSS
curl -s https://engineer.company/atom.xml      # Atom
```

And the site keeps a map of itself for anything that reads rather than crawls:

```
curl -s https://engineer.company/llms.txt
curl -s https://engineer.company/humans.txt
curl -s https://engineer.company/.well-known/security.txt
```

The source, over SSH

The site's source is kept in private repositories on our git server. Authorized collaborators access them over SSH using an approved SSH key:

```
ssh git.engineer.company
```

The command opens the server interface. Which repositories you can inspect depends on the access granted to your SSH key. The public page at git.engineer.company describes the service; it does not publish repository contents.

Why any of this exists

One deploy writes one folder, and four front doors read it — so none of them can drift from the others, and none of them is a separately maintained copy of the site. A page that reads well as gemtext is a page whose structure is real rather than painted on, which is the same property that makes it read well in a screen reader and print correctly on paper. The credits page names the daemons behind each one.

We build things that read well without a browser because that is usually a sign they were built properly — ask us to build you one.

One deploy, **four front doors** — and none of them can drift from the others.

A page that reads correctly without a browser is a page whose structure is real.

Ask us to build you one

Identify the client software and its needs

Start with the information your users need and the clients through which they need it.

See what it costs Get in touch →

Commission a usable text interface

Ask about stable text outputs and interfaces your own systems can consume.

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://daryna.engineer.company/cli/