

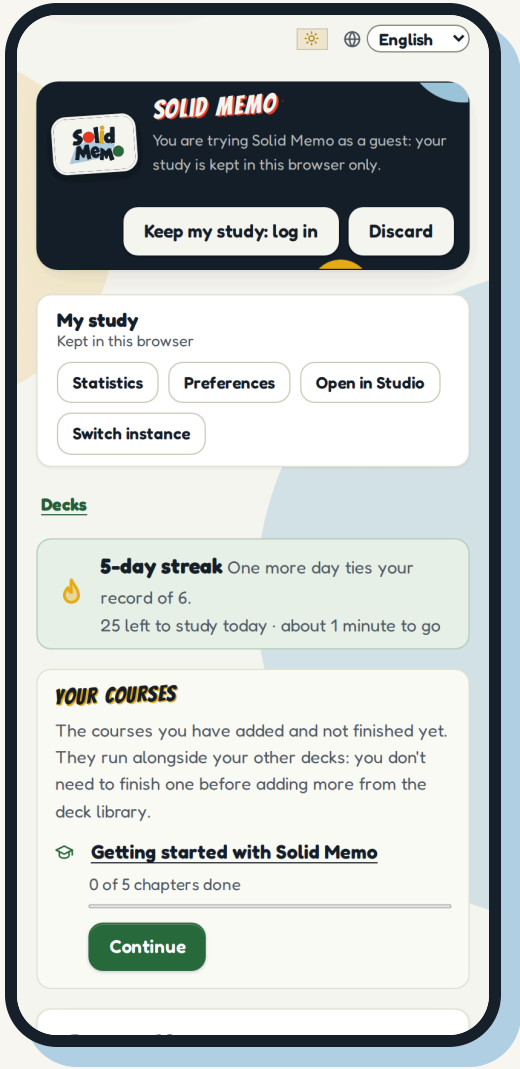

# NINE SERVER SETUPS, NO SERVER OF ITS OWN

What a Solid app for everyone learned in production, and what it asks of the ecosystem

# SOLID MEMO

Spaced-repetition flashcards in **your own pod**.

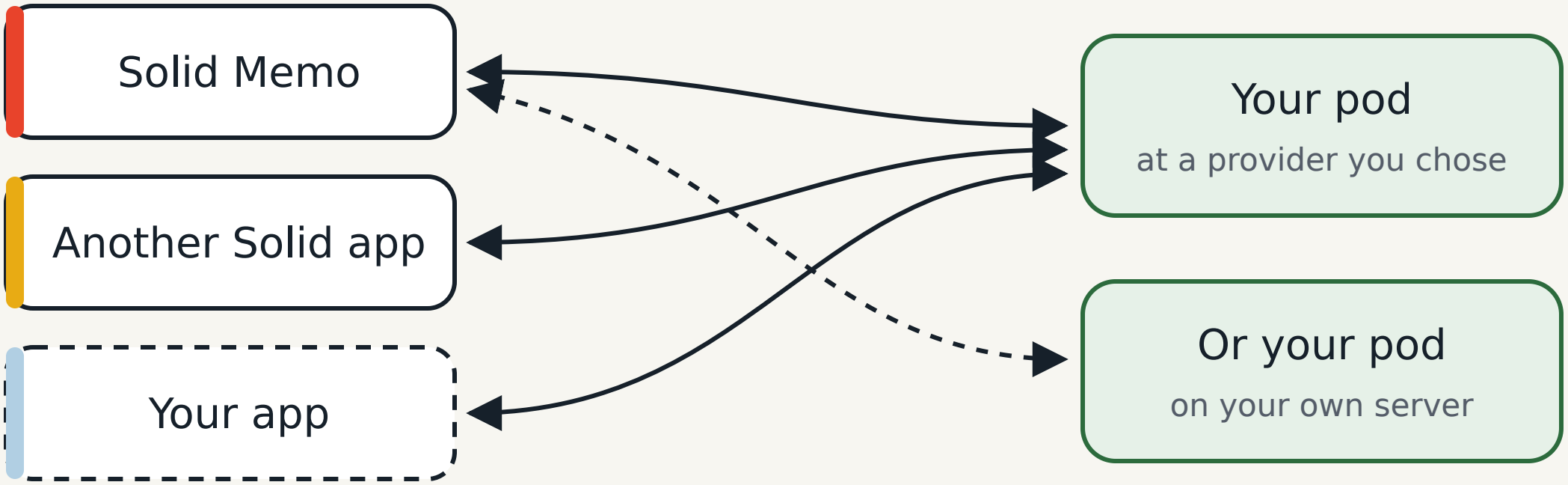
- Static files on GitHub Pages: **no backend, no secret**
- Guest mode, six UI languages, **115** decks and courses
- **1.0.0** on October 13, 2026; **168** releases in all
- Costs **123.75 SEK a year**: the domain



# WHERE EVERYTHING LIVES

**Apps: static files in your browser**

**Pods: the data, where you choose**




Each app reads and writes your pod directly, with your permission.

# 1. Logging in with no backend

# A client ID document, written by the build

```
const document = {
  "@context": [OIDC_CONTEXT],
  client_id: clientIdUrl(origin),     // https://solid-memo.com/clientid.jsonld
  redirect_uris: [`${origin}/`],
  scope: "openid offline_access webid",
  grant_types: ["authorization_code", "refresh_token"],
  token_endpoint_auth_method: "none",
  application_type: "web",
};
```

Used only when the provider lists `webid` in `scopes_supported`. Anywhere else (dev, test journeys, a fork without its own domain): dynamic registration.

# One redirect URI for every issuer

Every provider gets the same redirect URI and, on solid-memo.com, the same `client_id`.

Only **RFC 9207's** `iss` says which provider an answer came from.

- An answer **without** `iss`, or with another issuer's: refused
- The issuer each tab was sent to is kept **per tab**
- A provider that sends no `iss` **can't be used** to log in
- Parameters the tab never asked for are **stripped** before the library sees them

# DPoP binds a token to a key, not to a server

The login's fetch sends the user's token **to any address it's given**.

- **A credential gate**: as the user only inside their own storages and the Solid Memo folders they registered; elsewhere, read as no one, and never write
- `redirect: "error"` on every credentialed request
- **Signed out** on a 401 only when the session is gone, or on `error="invalid_token"` from the user's own storage or provider

# Behind a seam: a second client

```
export interface SessionClient {
  restore(): Promise<RestoredSession>;
  sessionIssuer(): Promise<string | null>;
  requestIssuer(state: string): Promise<string | null>;
  login(options: LoginOptions): Promise<void>;
  logout(): Promise<void>;
  fetch: typeof globalThis.fetch;
  webId(): string | undefined;
}
```

Inrupt's libraries by default. Behind a build flag, a client with **no Inrupt library**: its login **about 800 lines** on oauth4webapi. The same contract tests for both.

# 2. Data in someone else's pod

# Documents by how often they change

```
solid-memo/main/
  meta.ttl              what this instance is
  catalog.ttl           every deck, DCAT: one GET for the deck list
  decks/<deck>.ttl      a deck's cards: slow churn
  reviews/<deck>.ttl    their review states: fast churn
  history/2026-11.ttl   the answers, one document a month
  digest.ttl            derived facts, each with the ETag it came from
  drafts/  releases/    the Studio's
```

Not a resource per card: Solid has **no batch read**.

# Writing with no transactions

- `If-Match` on every edit, `If-None-Match: *` on every create
- On a **412**, read again and apply the change again, for documents others share: the type index, the catalog
- An answer is an **insert-only PATCH** with no precondition: its subject is unique
- **The order is the transaction**: create-only writes first, the link that names them last

```
INSERT DATA {
<#answer-20261003T081544001Z-p0m2zz> <https://solid-memo.com/ns/vocab/v1.ttl#grade> "1"^^<…#integer> .
};
```

# Formats that move, data that stays

- `formatVersion` on every subject; a **SHACL shape** per class and version, no `sh:targetClass`
- Old formats read **in memory**; the pod rewritten only **when the user agrees**
- **Never over a newer app's data**; others' triples survive
- Terms **never taken back**, shapes **never edited**

# A variant, not a migration

```
<#card-1> a solid-memo:ReviewState ;
    solid-memo:formatVersion 2 ;
    solid-memo:scheduler solid-memo:sm2 ;
    solid-memo:easeFactor 2.5 ; solid-memo:intervalDays 6 ;
    solid-memo:reviewOf <https://pod.example/decks/deck-1.ttl#card-1> .

<#review-3f1c2a9e-…> a solid-memo:ReviewState ;
    solid-memo:formatVersion 1 ;
    solid-memo:scheduler solid-memo:fsrs6 ;
    solid-memo:stability 6.2 ; solid-memo:difficulty 4.93 ;
    solid-memo:reviewOf <https://pod.example/decks/deck-1.ttl#card-1> .
```

# A GOOD NEIGHBOR IN A SHARED POD

- Reads **one** type registration, its own `solid-memo:Instance`
- Registers its decks, cards, review states and answers **for other apps**: the last two in the **private** index only
- Writes registrations **only when asked**, never because it opened
- Follows type index links only on the **WebID's origins or its storages**
- Removes **only what it wrote**, and keeps folders holding other apps' files

# 3. Nine server setups

# THE MATRIX

| Every change: blocks a merge | Weekly and on main: reported, not blocking |
|---|---|
| Community Solid Server 7.2.0 and 6.1.0 | Community Solid Server 8.0.0-alpha.3 |
| CSS 7.2.0 with **ACP** in place of WAC | Pivot 1.8.2, solidcommunity.net's server |
| Node Solid Server 6.0.0 and 5.8.8 | Solid-Nextcloud, on Nextcloud 32 |
| | JavaScript Solid Server 0.0.224 |

Real servers in Docker, pinned by digest, lockfile or commit. **About 100 tests** in each.

# PROBE, DON'T ASSUME

```
const CSS: Traits = { etag: true, etagEveryEdit: true, edits: true,
  creations: true, sparqlUpdate: true, n3: true };
const NSS: Traits = { etag: false, etagEveryEdit: false, edits: false,
  creations: true, sparqlUpdate: true, n3: true };

const PINNED: Record<string, Traits> = {
  "css-7": CSS, "css-6": { ...CSS, etagEveryEdit: false },
  "nss-6": NSS, "nss-5": NSS, "css-acp": CSS,
};
```

A test may skip only where **the protocol allows** what the server does. What the app needs is never a reason.

# What we found: Node Solid Server and CSS

| Server | What it does | What the app does |
|---|---|---|
| NSS 5, 6 | No ETag; `If-Match` ignored | Unconditional edits; reads everything each visit |
| NSS 5, 6 | PATCH: no `.` touching a term (`1.}`, as Inrupt's solid-client writes), at most 100 kB | Its own SPARQL Update; PUT over 64 KiB |
| NSS 5, 6 | `1.0` matched only as `1`; a patched `true` stored as false | Retries a 409; preferences by PUT |
| CSS 6 | ETags from whole seconds | Lives with it; the tests probe five edits |
| CSS 7, in-memory store | Cuts non-ASCII text short after a PATCH (the file store doesn't) | PUTs drafts whole |
| CSS with ACP | 404 for an ACR not written yet: the library sees no ACP | Reads it as an empty ACR |

# What we found: the other servers, and a library

| Server or library | What it does |
|---|---|
| JSS 0.0.224 | A patched literal loses its language tag: `"…"^^rdf:langString`, which RDF doesn't allow |
| JSS 0.0.224 | Takes ACLs in JSON-LD only: Turtle gets 415 |
| Solid-Nextcloud | `DELETE DATA` deletes **every value** of the subject and predicate |
| Solid-Nextcloud | Every document's `rel=acl` is the **folder's** `.acl` |
| Solid-Nextcloud | `HEAD` answers 204 for a missing resource |
| Pivot, CSS 8 alpha | **Pass everything** |
| @inrupt/solid-client 4 | `getSolidDataset` copies the graph per quad, **O(n²)**: 9 s for 3,010 cards; fetch and parse 50 ms |

# 4. Patterns worth taking

# THE POD IS A FETCH

```
export function routedFetch({ origin, local, remote }) {
  return (input, init) => {
    const url = input instanceof Request ? input.url : String(input);
    return url.startsWith(origin) ? local(input, init) : remote(input, init);
  };
}
```

- **Guest mode**: a small pod in IndexedDB, at `https://guest.solid-memo.invalid/`
- After login, the study can move into the real pod: **a copy from one URL to another**
- The same pod code again: the Studio's **sandbox**, and a **test double**

# Publishing, with no server of mine

1. A **draft** in the author's own pod
2. `PUT` **with** `If-None-Match: *`: `releases/<name>/v2.ttl`, written once
3. **Public read** on that one document: its own `.acl` or `.acr`, nothing else
4. **Linked** from the author's catalog, last

Readers add it **by link**: read as no one, within limits, SHACL-checked, and **the read that passed** is what's imported.

# What I'd ask of you

## Identity providers

- Send `iss`, always
- List `none` for public clients
- Spell your issuer **one way**

## Servers

- A strong **ETag on every write**
- Enforce `If-Match`
- Keep literals **as written**
- Delete **only what's named**
- **No redirects** for pod data

## Specs and libraries

- A **batch read**
- Client ID Document **caching and logout**
- A **latest version** link
- **Don't send tokens** everywhere

# What this proves

A polished app for anyone, built on Solid **by one person**,

tested against **five pod servers in nine setups**, with **no server** of its own,

for **123.75 SEK a year**.

# JOIN IN

- **Try it**: solid-memo.com
- **Read the findings**: a test and a reason for each, in the repository
- **Report a server**: the **Interop report** issue form
- **Take them upstream** with me



solid-memo.com



Sponsor the domain:

github.com/sponsors/ixuz

# NINE SETUPS.
# NO SERVER.

Thank you. Questions?

Slides: github.com/solid-memo/talks · CC BY 4.0 · Code: github.com/solid-memo/website