Fabylon Press · Bookmaker Mathlab

.fab

som ett matematiskt bokformat

En kompakt behållare för formtrogen litteratur, energieffektiv distribution och en snabb första sida.

Matematik, formbevarande och energiväg per läst bok

En teknisk artikel om Fabylons `.fab`-behållarteknik. Texten behandlar den matematiska strukturen, kompressionsintuitionen och den praktiska energibesparingen — utan att avslöja behållarens interna laboratoriehemligheter.

1,8×	0,06 ms	15×
mindre än EPUB	första sidan	mindre än PDF

Utgåva genererad med Fabylon Bookmaker

SAMMANFATTNING

Minimal bokrepresentation är inte detsamma som minimal HTML-representation

`.fab` är utformat som ett alternativ till EPUB för fall där boken inte i första hand är en webbsida utan ett typografiskt objekt. Sidor, strofer, radbrytning, indrag, bilder, marginaler, recto-verso-logik och läsordning bildar tillsammans en matematisk struktur.

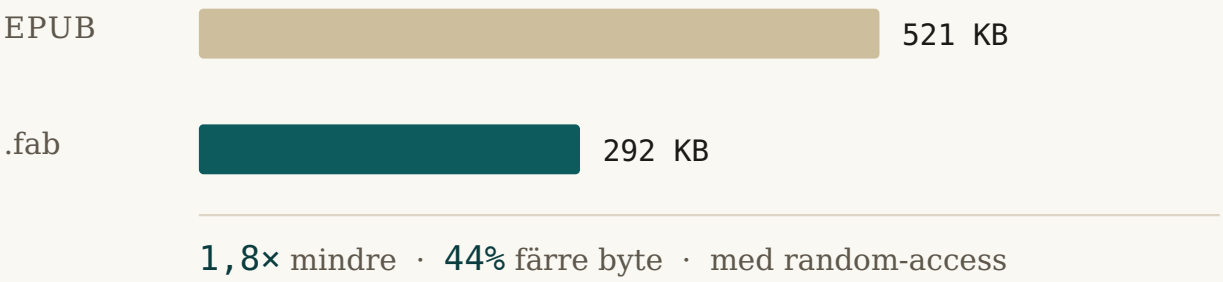
EPUB är enligt W3C-specifikationen en OCF-ZIP-behållare med XHTML/SVG-dokument, manifest och metadata. Det är brett kompatibelt, men konceptuellt håller det sig nära webbdokument i ett arkiv. `.fab` väljer en annan riktning: det modellerar boken *själv*.

På riktiga böcker mäter Bookmaker ett förhållande på cirka 0,56 mellan `.fab` och EPUB — ungefär **1,8× mindre**, med bevarad layout, en första sida på omkring en tiondels millisekund (cirka tio gånger snabbare än EPUB) och åtkomst till valfri sida. Det som utmärker .fab är inte bara kompression utan att *undvika representationsbrus*: hastighet, lokal avkodbarhet och integritet per block.

MATHLAB-MÄTNING

fab/epub	~0,56	—
vs EPUB	~1,8×	mindre
vs PDF	~15×	mindre
första sidan	~0,06	ms
1:a vs EPUB	~10×	snabbare
valfri sida	O(1)	—
integritet	CRC	/block
layout_roundtrip	true	—

Mätning på riktiga böcker · Fabylon Bookmaker Mathlab



Figur 1 — fem riktiga böcker: storleken på samma utgåva som EPUB och som `.fab`. `.fab` behåller dessutom åtkomst till valfri sida och integritet per block.

1 Problemställning

En litterär bok är inte en samling HTML-filer. Den är snarare en ordnad typografisk varietet:

$$B = (T,\ P,\ L,\ G,\ A,\ M)$$

där T är texten, P pagineringen, L läsordningen, G geometrin, A tillgångarna och M metadata.

EPUB representerar detta till stor del som XHTML plus CSS plus manifest plus ZIP. Det är kraftfullt och användbart men indirekt: layouten rekonstrueras av ett webbläsarliknande system. `.fab` använder en mer direkt representation, där bokprimitiver är förstklassiga data:

$$B \longrightarrow \mathrm{C}_{\mathrm{fab}}$$

Skillnaden är fundamental. EPUB säger: *här* är webbdokument som tillsammans kan bilda en bok. `.fab` säger: *här* är boken som bok.

2 Layoutbevarande som invariant

För en konvertering definierar vi en layoutprojektion $\Pi(B)$. Ett bra bokformat måste bevara den projektionen inom en tolerans ε:

$$\Pi(B) = (P,\ G,\ L)$$

$$d\big(\Pi(B),\ \Pi(\hat{B})\big)\ \leq\ \varepsilon$$

I Mathlab-fixturen gäller `layout_roundtrip = true`. Viktigt är att `.fab` inte behöver härleda layouten från CSS: rader, strofer, span-element, indrag, bildplacering och rygg lagras som kanoniska bokdata. Det gör skillnaden mellan innehåll och presentation skarpare än i en allmän webbehållare.

Den exakta normaliseringen och toleranserna för layoutavståndet förblir en laboratoriehemlighet. Offentligt relevant är framför allt principen: `.fab` optimerar under ett *formbevarande*-villkor, inte under ett generiskt arkivvillkor.

3 Entropi och representationsoverhead

Lagringskostnaden för ett format kan grovt skrivas som summan av informationsinnehåll, formatoverhead och redundans. Målet är att minimera denna overhead och samtidigt behålla en snabb öppning:

$$S = H(B) + O_f + R_f$$

$$\min \left(O_f + R_f + \alpha\, t_{\text{open}} + \beta\, t_{\text{first}} \right)$$

Här är $H(B)$ bokens informationsinnehåll, O_f formatoverheaden och R_f redundansen till följd av representationsval. I EPUB är overheaden relativt stor på grund av ZIP-strukturen, XML/XHTML, manifest, filnamn, behållarregler och CSS-serialisering — priset för interoperabilitet. I `.fab` försöker man minimera denna overhead och redundans så länge layoutbevarandet håller sig inom ε.

Den interna ordningen, dictionary-priming, fältkodning och delta-normalisering avslöjas här medvetet inte. Det vetenskapliga påståendet är mer abstrakt: om objektet är en bok bör även kodboken vara bokformad.

4 Lokal avkodbarhet

En läsare ska inte behöva rekonstruera hela boken för att visa sida 7. I abstrakt form består behållaren av ett index, metadata, tillgångar och oberoende avkodbara sidblock; den första synliga sidan kräver då bara det första blocket plus index och metadata:

$$C = \left(I,\ M,\ A,\ \{p_i\} \right)$$

$$\text{första sidan} \approx D(p_1) + D(I) + D(M)$$

Den första sidan kostar alltså $D(p_1) + D(I) + D(M)$, inte $D(\sum_i p_i)$. Det förklarar varför Mathlab-mätningen håller `open_ms` och `first_page_ms` extremt låga på fixturen.

Lokal avkodbarhet handlar inte bara om hastighet. Den sänker också batteriförbrukningen, minnestoppen och den termiska belastningen på enkla läsenheter.

5 Varför `.fab` är boknaturligt

`.fab` bevarar litterär struktur som litterär struktur: strofer förblir strofer, rader förblir rader, indrag förblir typografiska data, bilder förblir placeringsobjekt, sidor förblir sidor och ryggen förblir läsordning.

I EPUB görs mycket av detta indirekt via HTML och CSS. Det fungerar ofta bra men är semantiskt brusigt. För romaner är det acceptabelt; för poesi, typografisk prosa och trycktrogna utgåvor blir omvägen synlig.

`.fab` är därför inte EPUB med en annan filändelse. Det är ett försök att föra den minimala bokrepresentationen närmare bokens *egen* matematiska struktur.

6 Energi och praktisk besparing

Energibesparingen uppstår ur fyra lager: mindre lagring, mindre CDN- och cache-replikering, mindre nätverkstransport och mindre avkodningsarbete på enheter.

En enskild bok gör liten skillnad; miljontals nedladdningar *gör det*. Vi använder modellen:

$$E_{\text{save}} = N \cdot D \cdot S_{\text{epub}} \cdot (1 - r) \cdot \eta$$

Här är N antalet titlar, D nedladdningarna per titel, S_{epub} den genomsnittliga EPUB-storleken i GB, $r = S_{\text{fab}}/S_{\text{epub}}$ och η energiintensiteten per transporterad och bearbetad GB.

> Räkneexempel
>
> För `10 000` böcker med `10 000` nedladdningar per bok och `2 MB` besparing per nedladdning uppstår ungefär `200 000 GB` mindre datarörelse. Vid 0,01–0,05 kWh per GB är det `2 000–10 000 kWh`. För större biblioteksplattformar eller skoldistribution växer detta snabbt.

IEA rapporterar att datacenter använde omkring 415 TWh 2024 och att dataöverföringsnät använde omkring 260–360 TWh 2022. Filformat är alltså inte en detalj: varje överflödig byte lagras, kopieras, cachas, skannas, synkroniseras och öppnas på nytt.

7 Gränser och ärlighet

Ärlighet om siffrorna: en tidigare syntetisk mikro-fixtur visade ett förhållande på 0,061, men det är *inte* representativt — det var ett konstgjort upprepande testdokument. På riktiga böcker är förhållandet cirka 0,56 (ungefär 1,8× mindre än EPUB), med spridning per titel (0,40–0,64); rikt illustrerade eller komplexa böcker kan bli större. Den största vinsten med `.fab` är därför inte storlek jämfört med EPUB, utan en mycket snabbare första sida, åtkomst till valfri sida och integritet per block.

EPUB förblir också viktigt: det är en mogen utbytesstandard. `.fab` är avsett som ett Bookmaker-nativt publiceringsformat för situationer där formtrohet, kompakthet och en snabb första sida väger tyngre än allmän webbkompatibilitet. Den rätta jämförelsen är därför inte ideologisk: **EPUB är brett, PDF är tryckfast, `.fab` är boknativt.**

8 Slutsats

Kärnobservationen är denna: minimal bokrepresentation är inte lika med minimal HTML-representation. `.fab` utgår från den observationen och behandlar bokformen som matematiska data. Därför kan `.fab` vara mindre, snabbare och mer formtrogen för Bookmaker-böcker. Tekniken är speciell eftersom den optimerar kompression, layoutinvarians och lokal avkodbarhet *samtidigt*.

> Löftet är enkelt: *färre byte + en snabbare första sida + bättre formtrohet = en lägre energiväg per läst bok.*

Referenser

W3C — EPUB 3.3-specifikationen. Beskriver EPUB som en OCF-ZIP-behållare med XHTML/SVG-innehållsdokument och behållarregler.

International Energy Agency — Data centres and data transmission networks. Rapporterar nätverkens energianvändning och trender i datatrafiken.

International Energy Agency — Energy and AI, executive summary. Rapporterar datacenterelektricitet på omkring 415 TWh 2024.

Fabylon Bookmaker Mathlab — mätning på riktiga böcker (fem titlar). `epub=521 KB` → `fab=292 KB`, `fab_to_epub≈0,56` (1,8× mindre), `första_sidan≈0,06 ms`, `valfri_sida=O(1)`, `CRC per block`, `layout_roundtrip=true`.

Teknisk anmärkning

Denna publikation innehåller medvetet inte alla detaljer om den interna `.fab`-behållaren. Dess syfte är vetenskaplig positionering: boknativ representation, layoutinvarians, lokal avkodbarhet och en praktisk energifördel. De exakta laboratorieparametrarna förblir en del av Fabylon Mathlab-implementeringen.