



Scribe PDF
FOR PROGRAMMERS

32 TOOLS · DOCS, REVIEWS & RELEASES

The design doc, the code review, *the release note.*

Design docs, API references, RFCs, offer letters, release notes: every PDF that moves through an engineering org, handled on-device in one app, without uploading files to six different tools.

32

TOOLS IN ONE APP

0

FILES LEAVE YOUR DEVICE

1

APP, DESIGN DOC TO RELEASE

Download · App Store

scribepdf-tools.netlify.app

iOS & macOS · App Store & Mac App Store

Built for the paperwork behind the build.

An engineering org runs on PDFs: design docs routed for review, API references shipped to partners, RFCs circulated for comment, offer letters signed at onboarding, release notes archived at ship. Scribe PDF covers what happens to every one of those documents between the first draft and the release.

Why on-device matters here: API keys, database credentials, and internal architecture are exactly the files that shouldn't round-trip through someone else's server. Every tool below runs locally, inside Apple's app sandbox. Nothing is uploaded to process a PDF.

● Create & Organize	8 tools
● Read & Extract	7 tools
● Protect & Secure	6 tools
● Sign & Mark Up	5 tools
● Present & Listen	4 tools
● Optimize	2 tools

32

tools, organized
into six clear
workflows

2

platforms:
iPhone, iPad,
and Mac

0

servers your
engineering
docs touch

1

app, from
design doc to
release



Merge PDFs

[Create & Organize](#)

The release packet needs one file, not six exports. Merge the API reference, the migration guide, and the changelog into a single PDF for the release.



Split PDF

[Create & Organize](#)

A 90-page system design doc needs the API section pulled out for one team and the data model section pulled out for another, without re-exporting from the docs tool.



Images to PDF

[Create & Organize](#)

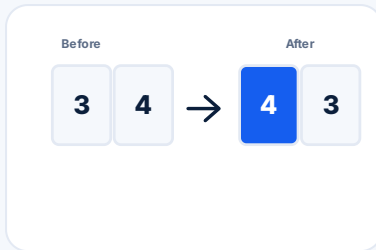
Twenty photos of a whiteboard architecture sketch become one clean PDF for the design doc appendix, with no separate scanner and no re-drawing the diagram.



Clipboard to PDF

[Create & Organize](#)

Copy a stack trace straight out of the terminal and paste it into a one-page PDF for the incident summary, without a screenshot tool in between.



Page Organiser

[Create & Organize](#)

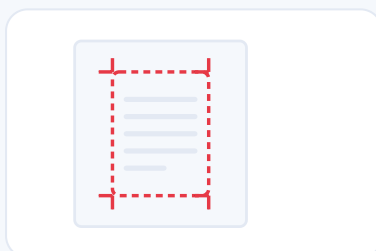
The onboarding packet came back from the printer with the deployment section ahead of the setup section; reorder the pages to match the onboarding template before it goes to new hires.



Rotate Pages

[Create & Organize](#)

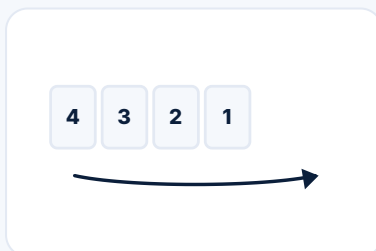
Half of the 30 scanned signed NDAs arrived sideways. Rotate every page back to portrait in one pass before they're filed with the offer letters.



Crop Pages

[Create & Organize](#)

A scanned whiteboard photo from a design review has a wide margin from the office scanner; crop it to match the rest of the bound design doc archive.



Reverse Pages

[Create & Organize](#)

The scanner fed 40 pages of signed offer letters in reverse order. Flip the page order back to chronological sequence before they're filed with HR.



OCR

[Read & Extract](#)

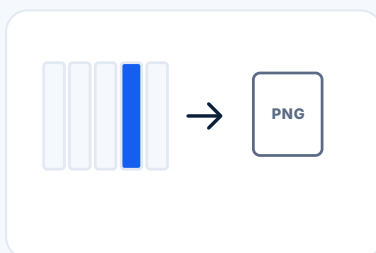
A five-year-old paper architecture diagram has to be referenced in this year's redesign, but it's a photocopy. OCR turns it into real, selectable, searchable text before it goes in the design archive.



Extract Text

[Read & Extract](#)

Pull just the error-handling section out of a 40-page API reference into plain text, ready to drop straight into the onboarding doc without retyping.



Extract Pages

[Read & Extract](#)

Export only the sequence diagram slide from a 60-page architecture review deck as a clean PNG for the design doc, with no screenshotting and no re-opening the slide file.

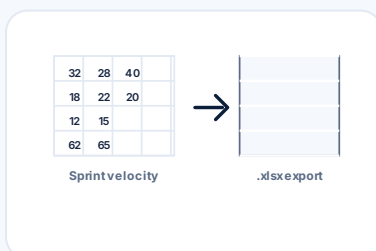


Table Extraction

[Read & Extract](#)

A sprint velocity table is buried in a PDF export from the planning tool. Pull it straight into a real spreadsheet, cell by cell, instead of re-keying story points by hand.



Find in PDF

[Read & Extract](#)

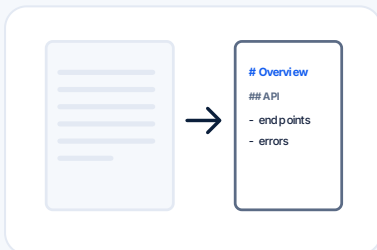
Find every reference to a deprecated function name across a 200-page API reference before the migration: every match, across every page, in seconds.



Document Stats

[Read & Extract](#)

Check the page count, word count, and read time on a draft RFC before it goes out for review, a quick sanity check before the real review.



PDF to Markdown

[Read & Extract](#)

Convert a vendor's API reference PDF into clean Markdown so it drops straight into the internal wiki, headings and code blocks intact.

Protect & Secure

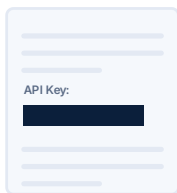
Lock what's internal, remove what shouldn't travel, and prove what changed.



Password

Protect & Secure

Encrypt the file containing a set of database credentials before it's emailed to a new contractor: real password protection, not a folder that just looks locked.



Redact

Protect & Secure

Black out API keys and internal hostnames on a sample of debug logs before they go into a public bug report: gone for good, not just covered.



Watermark

Protect & Secure

Stamp "DRAFT, NOT FOR DISTRIBUTION" across every page of the preliminary RFC before it's shared outside the team.

TITLE	System Design Doc
AUTHOR	Priya Natarajan
VERSION	2.3

Metadata

Protect & Secure

Set the Title, Author, and Version fields correctly on the final system design doc before it's archived: the details a doc search actually reads.

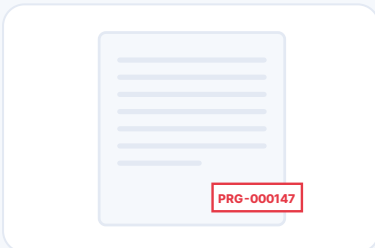
AUTHOR	priya.natarajan@company.com
FILE PATH	/internal/eng/design-docs/vs.pdf

Removed before external release

Strip Metadata

Protect & Secure

Strip the author's email and internal file path out of a PDF before it's shared externally with an open-source contributor.



Bates Number

Protect & Secure

Stamp PRG-000001 through PRG-001900 across every page of a source-code documentation set for a security and license-compliance review, sequential and consistent, the way the audit expects it.

Sign & Mark Up

Sign it, stamp it, mark it up, ready for the review or the release.

Contractor NDA

Priya Natarajan

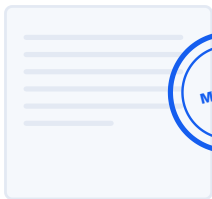
Due: 09/17/2026



Fill & Sign

Sign & Mark Up

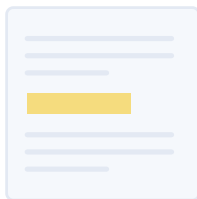
Fill in the contractor NDA fields and add a signature before onboarding day, with no printing, signing, or re-scanning.



Stamp Library

Sign & Mark Up

Stamp MERGED across the completed design doc the moment the review is approved, before the implementation ticket opens.

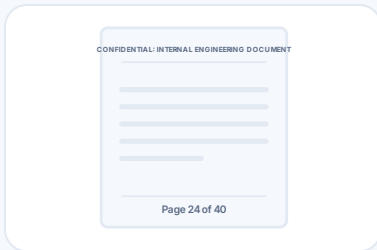


Data race
here, add a
mutex lock

Annotate

Sign & Mark Up

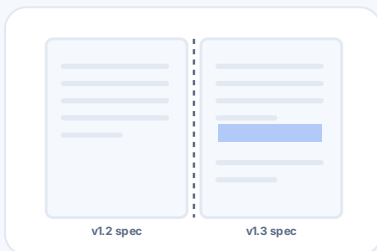
Highlight the race condition in the design doc and leave a comment for the author, directly on the diagram.



Headers & Footers

[Sign & Mark Up](#)

Add "Confidential: Internal Engineering Document" and automatic page numbers across a 40-page architecture doc before it goes out for a cross-team review.



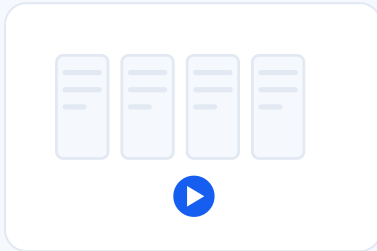
Compare PDFs

[Sign & Mark Up](#)

See exactly what changed between the v1.2 spec and the v1.3 spec, side by side, before approving the API change.

Present & Listen

Take the design doc into the review meeting, or into your commute.



Slideshow

Present & Listen

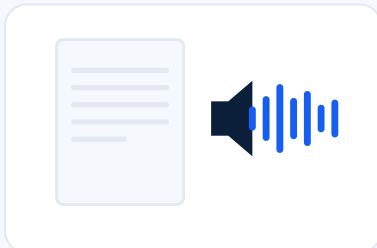
Present the architecture roadmap full-screen to the team, straight from the PDF, with no separate deck to keep in sync with the design doc.



Presenter Mode

Present & Listen

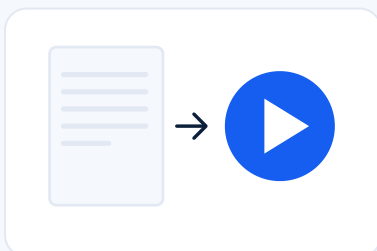
Walk the team through the architecture diagram with a laser pointer and on-slide markup, pointing straight at the component that matters.



Read Aloud

Present & Listen

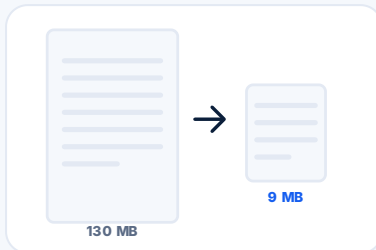
Have a 15-page RFC read aloud on the commute in, instead of squeezing the reading in before the design review.



PDF to Audiobook

Present & Listen

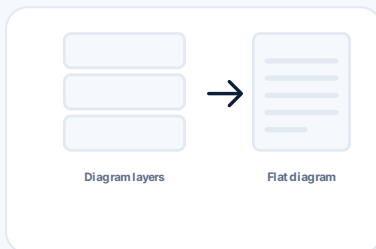
Turn the certification exam reading into an MP3 for the commute, with the renewal deadline closing in.



Compress PDF

[Optimize](#)

A 130MB scanned onboarding packet from HR won't attach to an email. Compress it down to something that actually sends, without losing legibility.



Merge Layers

[Optimize](#)

Flatten the optional-content layers a diagramming tool leaves behind in its PDF export, so the architecture diagram prints clean for the file instead of blank.



Get Scribe PDF.

Every workflow on these pages, in one download. Available now on the App Store and the Mac App Store.

Download Scribe PDF

scribepdf-tools.netlify.app