# Dangerzone

# Considering PyMuPDF

Evaluation of PDF Rendering Engine Change
November 2023

# Overview

- Code impact
- Installation impact
- Metrics: conversion speed, size and success rate
- Rendering Impact (manual analysis)
- License Impact

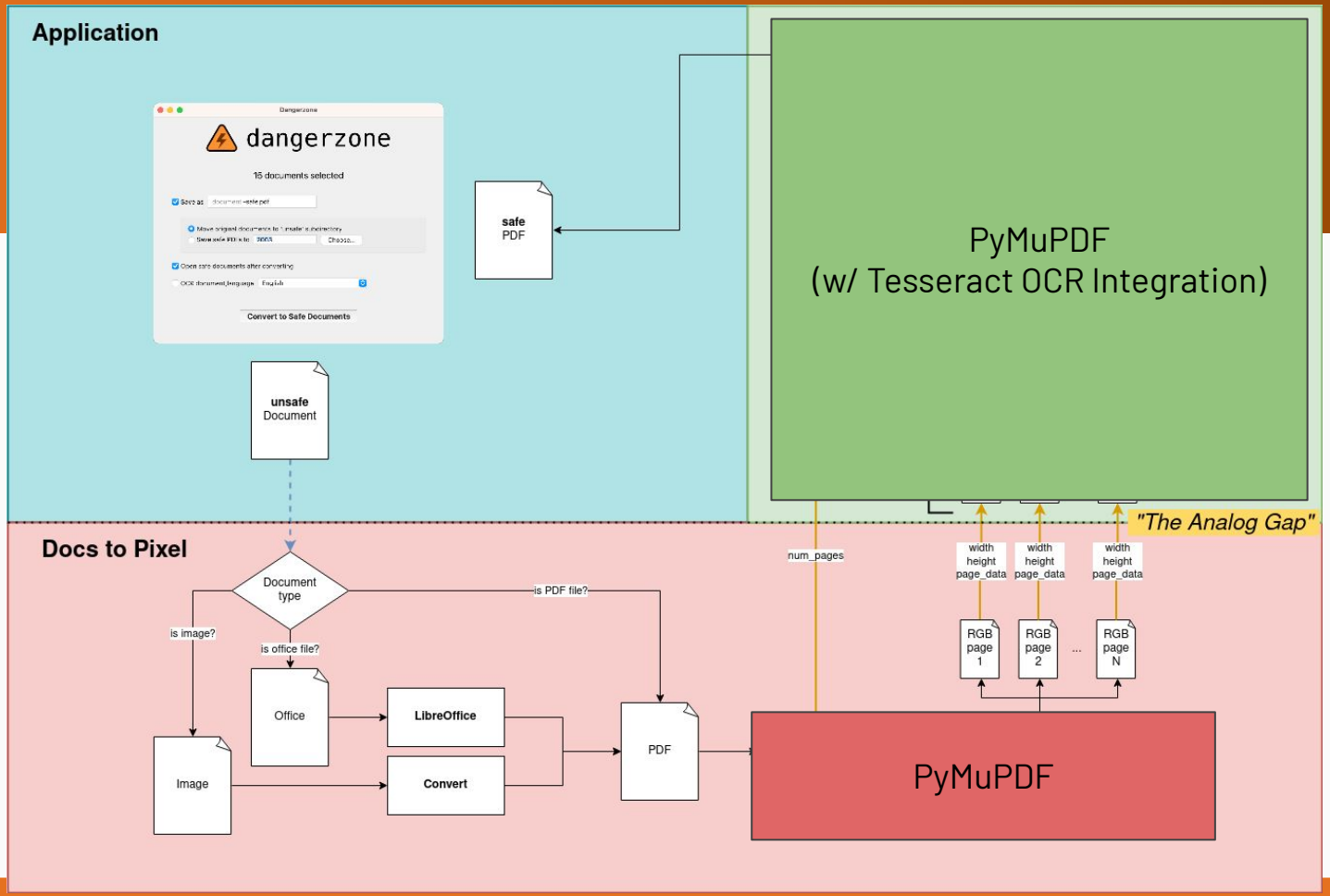
# Code Impact

- Simplifies significantly server code
- Replaces multiple tools:
  - Convert (GraphicsMagick)
  - PDFtoPPM (poppler-tools)
  - ps2pdf (GhostScript - for compression)
  - pdfunite (for merging PDFs)
- Integration with Tesseract-OCR
  - Whatever it does it's faster than our use of tesseract
- No more need for subprocess calls except LibreOffice

# Code Impact

**[big] No bottlenecks:** no need to write files to disk (like PDFtoPPM) in first conversion

This is what made us consider PyMuPDF in the first place (see #616)


Application
dangerzone
15 documents selected
Convert to Safe Documents
safe
PDF
PyMuPDF
(w/ Tesseract OCR Integration)
unsafe
Document
"The Analog Gap"
Docs to Pixel
Document
type
is PDF file?
is image?
is office file?
Office
LibreOffice
Image
Convert
PDF
num_pages
width
height
page_data
RGB
page
1
RGB
page
2
RGB
page
N
PyMuPDF

# Docker Build Challenges

- Fetches MuPDF binary from MuPDF.com
- ! Does not verify it
- Increases container build time significantly (though we may be able to reduce it)

# Metrics: Performance

Faster on larger files, slower on smaller files

| Test Data | | | | | Results | | |
|---|---|---|---|---|---|---|---|
| Branch | Commit | Test Type | Test Device | Command | average | Performance boost (versus main) | Failure Rate |
| 53115b3 | 443-stream-pa | 214 page (no OCR) | Dell XPS 13 (9305) | time python ./de | 45.5502 | 1.36 | - |
| 04f8fa8 | main | 214 page (no OCR) | Dell XPS 13 (9305) | time python ./de | 62.111 | 1.00 | - |
| 53115b4 | 443-stream-pa | 214 page (OCR) | Dell XPS 13 (9305) | time python ./de | 706.8675 | 1.41 | - |
| 04f8fa9 | main | 214 page (OCR) | Dell XPS 13 (9305) | time python ./de | 1201.3112 | 1.00 | - |
| 04f8fa83 | 443-stream-pa | 214 page (no OCR) | M1 Mac | time python ./de | 45.26366667 | 1.17 | - |
| 6876fa4 | main | 214 page (no OCR) | M1 Mac | time python ./de | 54.2516 | 1.00 | - |
| 04f8fa83 | 443-stream-pa | 214 page (OCR) | M1 Mac | time python ./de | 228.0674 | 1.33 | - |
| 6876fa5 | main | 214 page (OCR) | M1 Mac | time python ./de | 340.2052 | 1.00 | - |
| 53115b4 | 443-stream-pa | 4000 docs test (no O | M1 Mac | make test-large | 9740.88 | 0.79 | 1.33% |
| 04f8fa9 | main | 4000 docs test (no O | M1 Mac | make test-large | 8080.741 | 1.00 | 0.77% |

# Metrics: Document Size

**Shrunk:** 4762, average: -34.4%

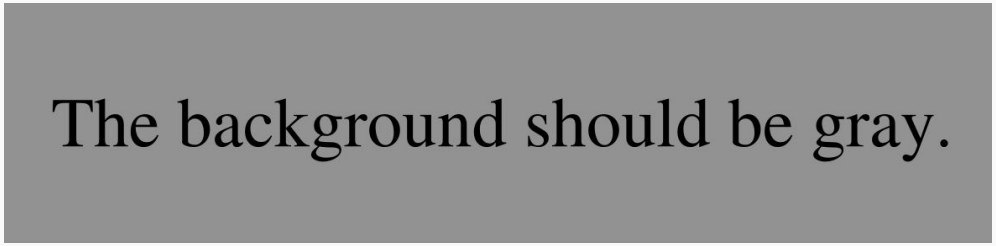
**Increased:** 133, average: 68.5%

**Total average:** -4.3%

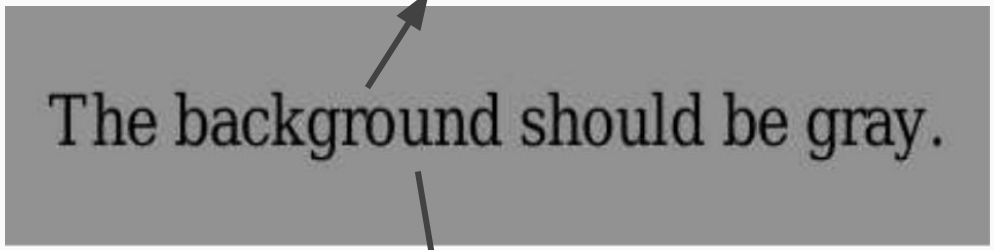
**Bonus points:**
**Less Compression Artifacts**

**Note:** low resolution in PyMuPDF will be resolved by #626

original (adobe acrobat)

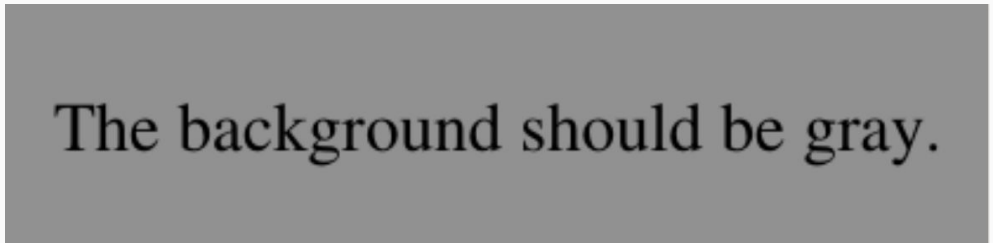



PDFtoPPM (ps2pdf) Compression artifacts




Yes, also font differences

PyMuPDF



# Other PyMuPDF honorable mentions

- **Supports more file formats** (XPS, EPUB, MOBI, FB2, CBZ, SVG and various image formats), which mean we can add these at any time *with no extra cost*
- **Opens possibility of on-host pixels2pdf**
  Dropping extra binary dependencies (poppler, ps2pdf, pdfunite) opens up path for on-host pixels2pdf. We only need to figure out Tesseract-OCR install on Windows and MacOS and PyMuPDF in the old ubuntu 20.04

# Open Questions

- Container image size impact:
  - Does MuPDF already include CJK so we can remove its dependencies
- Can our vuln. Scanner (grype) detect PyMuPDF?

# WARMUP

PDF is technically a standard but rendering can vary widely

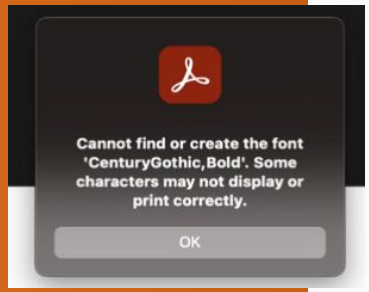
Akin to the web standards

# WARMUP

PDF is a standard but...

This particular issue seems to be the lack of [font embedding](#) in the document

[issue4722.pdf](#)




PDFtoPPM

' (6&5,37,2 1

Evince

' ( 6&5,37,2 1

PyMuPDF

F GUE TKRVKQ P

GitHub.com rendering

DESCRIPTION

MacOS Preview

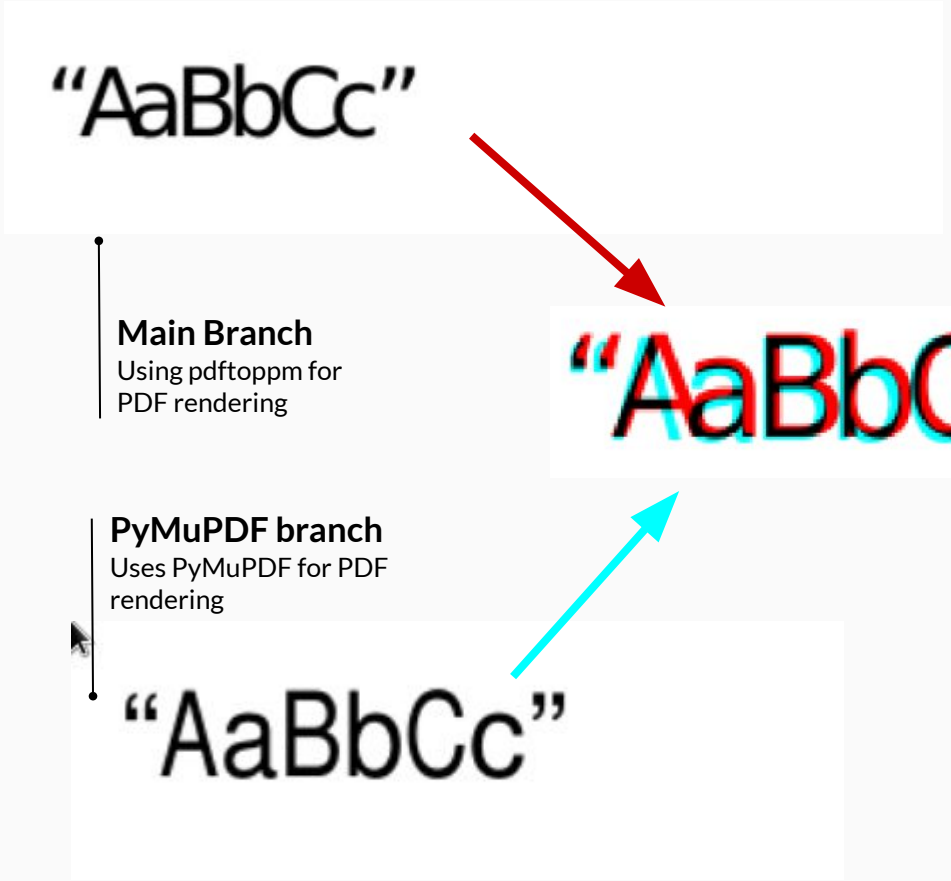
DESCRIPTION

Adobe Acrobat

# PDFs Visual Diffing

## Using the tool

github.com/
vslavik/diff-pdf



# Methodology

**1.** Run large test on <10KB docs from this set and save results

**2.** Repeat on the other branch for comparison

**3.** Manual inspection of diffs (data here)

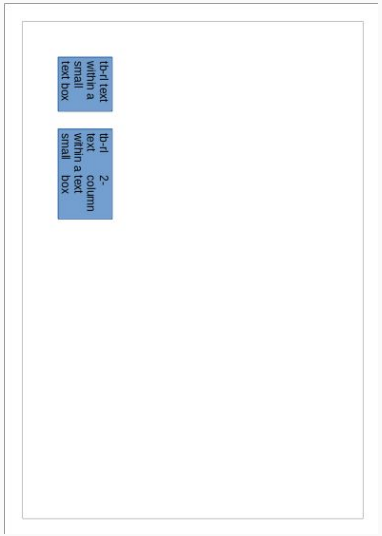
| File Name | main (pdftoppm) | pymupdf2 branch | |
|---|---|---|---|
| ./simple3.pdf-safe.pdf | 64331 | 29108 | widely different |
| ./issue4722.pdf-safe.pdf | 9716 | 6697 | text widely differ |
| ./tdf135035.odt-safe.pdf | 18620 | 13143 | some lines are fi |
| ./tb-rl-textbox.odg-safe.pdf | 45911 | 31511 | rotation |
| ./tbrl-frame.odt-safe.pdf | 17389 | 11676 | rotation |
| ./tdf108482.odt-safe.pdf | 22383 | 28316 | rotation |
| ./tdf130657.xlsx-safe.pdf | 59897 | 43520 | rotation |
| ./tdf136364.xlsx-safe.pdf | 24264 | 17423 | rotation |

# Rotations

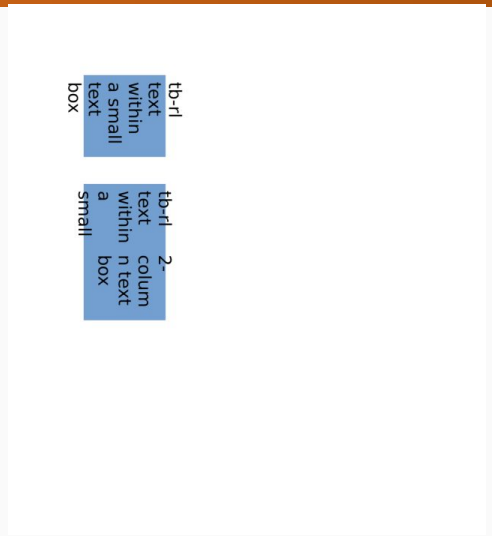
There were lots of them in many different file types.

PyMuPDF seems to have always the rotation right.

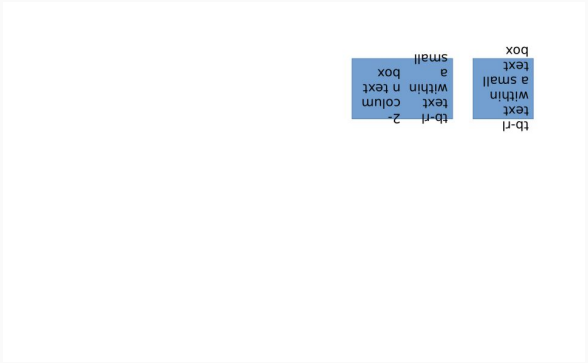
# Rotations example




original




PyMuPDF




PDFtoPPM

## Font Differs

**There were lots of them in many different file types.**

PyMuPDF seems to have always the rotation right compared to the original


# Font Differences


Only certain fonts differ

1 Helvetica Helvetica

2 Arial Arial Arial Aria

**One case of PyMuPDF weirdness**

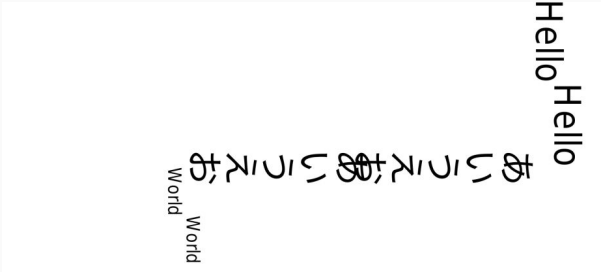
**This must be a particularly problematic PDF** (simple3.pdf)

Even adobe reader fails to render it!

original (apple preview)


⚠️ original (adobe reader)

PDFtoPPM




PyMuPDF

# PyMuPDF characters missing

tdf148706.odt
Issue5244.pdf
operator-in-TJ-array.pdf



# PDFtoPPM characters missing

One single case detected:

issue5954.pdf

# Line Rendering differs

From .ods/.odt file

PDFtoPPM

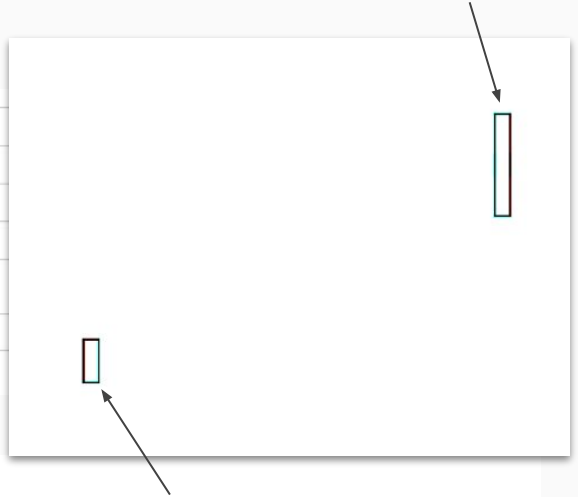
| | | |
|---|---|---|
| X | | |
| X | | |
| X | | |
| X | | |

| | |
|---|---|
| One | Two |

PyMuPDF

| | |
|---|---|
| X | |
| X | |
| X | |
| X | |

| | |
|---|---|
| One | Two |

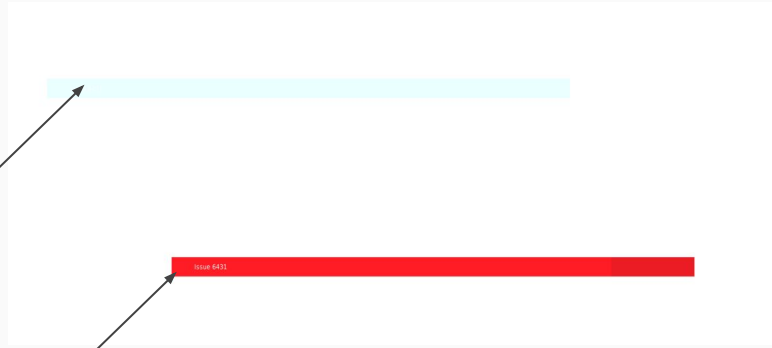


veraPDF test suite 6-3-3-t01-fail-j.pdf

PyMuPDF closer to Preview (adobe doesn't even render)

# Placement

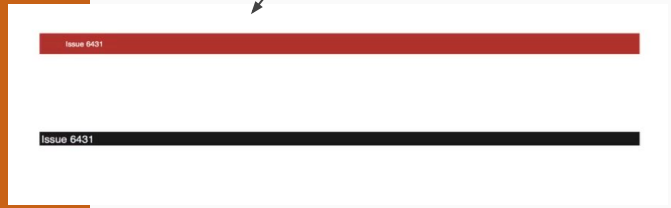
PyMuPDF had better placement



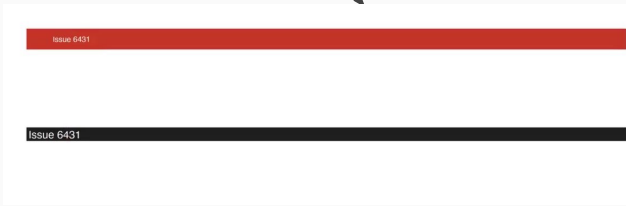

PDFtoPPM

PyMuPDF & evince placement

**But the above are missing the black bar**




macOS preview




Adobe Acrobat

# AGPL License Impact

# Questions?