Entwicklung einer webbasierter Applikation zur Bearbeitung von PDF Dateien

Bachelorarbeit zur Erlangung des Bachelor-Grades
(Bachelor of Science) im Studiengang Technische Informatik
an der Fakultät für Informations-, Medien- und Elektrotechnik
der Technischen Hochschule Köln

vorgelegt von:	Janina Schroeder
Matrikel-Nr.:	11132206
Adresse:	Laurentiusweg 10
	50321 Brühl
	janina_jessika_jelena.schroeder@smail.th-koeln.de

eingereicht bei:	Prof. Dr. Chunrong Yuan
Zweitgutachter/in:	Prof. Dr. René Wörzberger

Köln, 04.03.2024

Bachelorarbeit

Titel: Entwicklung einer webbasierter Applikation zur Bearbeitung von PDF

Dateien

Gutachter:

- Prof. Dr. Chunrong Yuan

- Prof. Dr. René Wörzberger

Zusammenfassung: Für die Bachelorarbeit habe ich eine Open Source offline Webseite zur Bearbeitung von PDF Dateien im Firefox Browser programmiert. Seit Adobe den PDF Standard entwickelt hat, tauchten zahlreiche meist kostenpflichtige PDF Anwendungen, um PDF Dateien zu bearbeiten auf dem Markt auf. Ich habe den Markt an PDF Programmen analysiert und diese mit meiner Webapplikation verglichen. Daraufhin beleuchte ich den aktuellen Stand der Technik des PDF Standards. Im späteren Verlauf erkläre ich die Implementierung meiner Webapp und meine Erfahrungen mit anderen Browsern, sowie auf MacOS, Linux, Android und iOS. Die Javascript Libraries PDF.js und PDF-LIB sind das tragende Fundament meiner PDF Webapp. Die PDF Webapp vereint alle Funktionalitäten, die man für gängige PDF Bearbeitung benötigt. Man kann PDFs lesen, splitten, mergen, erstellen, sowie mit Texten, Bildern, Geometrie und Zeichnungen versehen. Am Ende diskutiere ich, was man hätte besser machen können, welche Funktionalitäten fehlen und welche Features in Zukunft noch geplant sind.

Stichtwörter: PDF Bearbeitung, Adobe, Javascript, Vue JS 3, auf PDF zeichnen, Splitten, Mergen, PDF.js, PDF-LIB

Datum: 04. März 2024

Inhaltsverzeichnis

Abkürzungsverzeichnis

AES Advanced Encryption Standard. 18, 19, 33–37, 71

ASCII American Standard Code for Information Interchange. 20, 21

BITV Barrierefreie-Informationstechnik-Verordnung. 15

CAD Computer-Aided Design. 5, 13

CAdES CMS Advanced Electronic Signatures. 8

CBC Cipher Block Chaining. 33, 34, 36–38, 71

CDN Content Delivery Network. 43

CEPS Cisco Enterprise Print System. 17

CID Character Identifier Font. 4, 17

CIE Commission Internationale de l'Éclairage. 16, 17

DCS Desktop Color Separation. 10

DOS Denial-of-Service. 28

ECB Electronic Code Book. 37

EEA Evil Annotation Attack. 28

EPS Encapsulated PostScript. 51

ETSI European Telecommunications Standards Institute. 8

GDI Graphics Device Interface. 19

ICC International Color Consortium. 10–12, 14, 17, 51

ISA Incremental Saving Attack. 30, 31, 71

ISO International Organization for Standardization. 2, 6, 8–19, 51, 54

JSON JavaScript Object Notation. 60

MAC Message Authentication Code. 34, 36, 38

MIME Multipurpose Internet Mail Extension. 14, 19, 37

MITM Man-in-the-middle. 33, 37

OCR Optical Character Recognition. 15, 56, 60

OPI Open Prepress Interface. 10, 17

PAdES PDF Advanced Electronic Signatures. 8, 50

PCS Profile Connection Space. 10

PDL Page Description Language. 19–21

PPD PostScript Printer Description. 19, 56

ppi Pixels per inch. 21

RIP Raster Image Processor. 19, 20, 51, 52, 55

RLE Run Length Encoding. 7

SSA Sneaky Signature Attack. 28

SWA Signature Wrapping Attack. 30–32, 71

USF Universal Signature Forgery. 30–32, 71

WCAG Web Content Accessibility Guidelines. 15

WYSIWYG What You See Is What You Get. 2, 3, II

XAdES XML Advanced Electronic Signatures. 8

XFA XML Forms Architecture. 6, 18, 19

XML Extensible Markup Language. 6, 18, 37, 53

XMP Extensible Metadata Platform. 12, 13

ZSA Zeitstempel-Anbieter. 29

Einleitung

Motivation

Zu Hause benutze ich 4 PDF-Programme, um alle für mich zufriedenstellenden häufigen Anforderungen der PDF-Bearbeitung zu erledigen: Adobe Reader zum Anzeigen von PDF, Drawboard PDF zum Zeichnen, Foxit Reader zum Schreiben und PDF Sam Basic zum Splitten und Mergen. Ich habe dann später herausgefunden, dass man mit Foxit Reader auch Splitten, Mergen und Zeichnen kann, jedoch finde ich dieses Programm sehr unintuitiv und ich vergesse immer wo die Einstellmöglichkeiten für diese Funktionalitäten waren. Oft habe ich dann keine Lust die Einstellmöglichkeiten im Internet zu googlen und nehme dann lieber PDF Sam Basic zum Splitten von Seiten. PDF Sam Basic ist da einfacher, jedoch ist es sehr störend, dass man bei seiner Installation auch eine andere Werbeversion von PDF Sam zusätzlich installiert. Ich habe mir gedacht, dass es nicht sein kann, dass mir kein kostenloses PDF Programm so wirklich gefällt und deshalb beschlossen meine eigene PDF Web App zu entwickeln. Weiter habe ich überlegt: Es wäre sinnvoll eine Webseite zu programmieren, die man auch offline nutzen kann. Alles was man benötigen soll ist ein Browser und keine Installationen mit zusätzlichen Werbeprogrammen, die man eigentlich nicht installieren wollte. Wie wäre es mit einer Open Source Web App, die ich meinen Freunden zeigen kann und mit denen ich mich auf dem Arbeitsmarkt bewerben kann? Gesagt, getan - ich habe diese PDF Web App für diese Bachelorarbeit programmiert und sie ist neben ein paar Ecken und Kanten besser geworden, als ich es je erwartet hätte. Ich habe während der Programmierung viel gelernt über JavaScript, asynchrone Programmierung, Event Handler, usw. und bin generell sicherer geworden im Programmieren. Vor allem habe ich Wert darauf gelegt, dass die Web App eher einfach und intuitiv zu bedienen ist. Einige Features waren nicht geplant, aber mir hat es Spaß gemacht sie zu implementieren, da ich die Motivation hatte ein für andere und mich gutes Tool zu entwickeln. Natürlich gibt es noch eine lange Liste an Features, die ich in Zukunft noch implementieren möchte. Die PDF Web App ist kein abgeschlossenes Projekt. Dieses Hobby-Projekt möchte ich gerne nach der Bachelorarbeit weiter auf meiner Github-Seite maintainen. Vielleicht gibt es andere Entwickler, die mein Repository forken oder bug fixes hinzusteuern wollen. Vielleicht kann ich andere Entwickler dafür begeistern an der PDF Web App mitzuarbeiten und sie noch besser zu machen.

Aufbau der Arbeit

1 Grundlagen

Das PDF Dateiformat steht für Plattformunabhängigkeit, Hardwareunabhängigkeit, Konsistenz in Formatierung und Layout und soll ein möglichst originalgetreues Druckergebnis liefern. Der Leser soll ein PDF Dokument immer nach dem Prinzip WYSIWYG (What You See Is What You Get) in der Form betrachten und ausdrucken können wie es vom Ersteller des Dokuments festgelegt wurde.

Abbildung 1.1 Adobe PDF Icon [1]

PDF wurde im Jahr 1993 von dem 1982 gegründeten amerikanischen Softwareunternehmen Adobe Inc. veröffentlicht und ging aus dem 1991 von Adobe-Mitbegründer John Warnock gestarteten „Project Camelot"hervor[2]. Ziel des Projekts war es, ein Dateiformat für elektronische Dokumente zu kreieren, sodass dieses Anwendungsprogramm, Betriebssystem und Hardware unabhängig sowie originalgetreu wiedergegeben werden kann. In Abbildung 1.1 ist das Datei-Icon für PDF von Adobe zu sehen.

Anfangs war der Adobe Reader noch kostenpflichtig und PDF war für einen langen Zeitraum ein proprietäres Dateiformat, welches offengelegt im PDF Reference Manual von Adobe dokumentiert ist. Die Spezifikation von PDF ist seit 1993 kostenlos einsehbar [1]. Die International Organization for Standardization (ISO) übernahm PDF im Jahr 2007 in den Standardisierungsprozess und seit der Veröffentlichung von PDF Version 1.7 am 1. Juli 2008 gilt PDF als Offener Standard als ISO 32000-1:2008[1], [2]. Vorher war PDF ein proprietäres Dateiformat von Adobe. Der Begriff Offener Standard bezeichnet einen Standard, der für alle Teilhaber am Markt besonders leicht zugänglich, weiterentwickelbar und einsetzbar ist. Das bedeutet, dass der Standard von einer gemeinnützigen Organisation eingeführt, veröffentlicht sowie weiter bearbeitet wird und gleichmäßige Einflussnahme aller interessierten Parteien ermöglicht [3].

Im gleichen Jahr publizierte Adobe eine Public Patent Licence zum ISO Standard 23000-1, also PDF Version 1.7, die royalty-free Rechte einräumt, um PDF-Implementierungen zu programmieren, verkaufen und verbreiten [1]. Royalty-free bedeutet hierbei, dass Compu-

terherstellerfirmen pro verkauftes Endgerät keine Gebühren (royalties), sowie keine fixe Jahrespauschale bezahlen müssen [4]. Heute wird PDF seit 2006 von der PDF Association weiterentwickelt [2].

1.1 Wichtigste Features

Die in den Unterkapiteln genannten Operationen auf dem PDF-Dateiformat beziehen sich hauptsächlich auf Adobe Acrobat-Werkzeuge. PDFs können Texte, Tabellen, Bilder, Pfade, Links, Buttons, Formulare, Audio-, Videoelemente und Funktionen enthalten. Rich Media-PDFs ermöglichen interaktive Inhalte, die eingebettet oder verlinkt werden können. Solche Elemente sind Bilder, Audio, Video oder Buttons, z.B. als digitaler Katalog [1]. Fonts und Bilder sollten grundsätzlich immer eingebettet werden.

In PDFs werden alle Informationen als nummerierte Objekte gespeichert. Objekte können zu Gruppen kombiniert werden. Der aktuelle Farbmodus im Dokument kann in andere Farbmodi konvertiert werden. Um die Navigation innerhalb eines PDF Dokuments zu erleichtern kann man anklickbare Inhaltsverzeichnisse und miniaturisierte Seitenvorschauen (Thumbnails) verwenden. Optional ist eine Gliederung als hierarchische Baumstruktur in Form von Lesezeichen möglich, mit der der Betrachter leichter durch das Dokument geführt werden kann.

PDF-Dateien enthalten grundsätzlich Metadaten. Bei Metadaten oder Metainformationen handelt es sich um strukturierte Daten, die sich auf Merkmale anderer Daten beziehen. Beispiele für Metadaten sind Name, Titel der Datei, Autor, Stichwörter zum Inhalt, das Datum der Speicherung. PDF-Dateien können Dateianhänge enthalten, die geöffnet und im lokalen Dateisystem abgespeichert werden können [1].

1.1.1 WYSIWYG

Ein PDF-Dokument hat ein festes Layout und eine feste Anzahl von Seiten. Unabhängig von der Software mit der das Dokument angezeigt wird oder mit welcher Hardware es ausgedruckt wird bleiben alle Elemente auf den Seiten immer exakt an derselben Position. Alle Layout- und Formatierungsangaben stammen aus der Erstellungsanwendung. Bei der Konvertierung von Dokumenten mit variablem Layout zu PDF, wie z.B. .txt-Dateien oder HTML muss der Inhalt auf die vorhandenen Seiten und den verfügbaren Platz verteilt werden.

1.1.2 Fonts

Jedes Textzeichen ist ein abstraktes Symbol und ein Schriftzeichen beruht auf eine graphische Darstellung. Eine Schriftart ist in PDF als Objekt enthalten. Die Schriftart kann mit Werkzeugen in Acrobat bearbeitet werden. Der Text muss ausgewählt sein und es können

darauf folgende Operationen angewandt werden: Farbveränderung in RGB, Transparenzen, Verschiebung, Löschen, Skalierung, Verzerrung bzw. Scherung, Spiegelung, Drehung, Beschneidung oder Ersetzung. Der RGB-Farbraum eignet sich lediglich für die Bildschirmdarstellung und beschreibt die für den Menschen 16,7 Millionen sichtbaren Farben mit Hilfe von additiver Farbmischung mittels Rot, Grün und Blau. Ein Farbraum umfasst die mathematischen Parameter als Daten für die Gesamtzahl der Farben, die auf einem Monitor, Druckmaterial, usw. darstellbar sind [5].

In Adobe Acrobat Pro kann der gesamte Text pro Seite in Pfade konvertiert werden. Pfade sind mathematisch berechnete Linien, die aus gekrümmten Segmenten bestehen. Der Anfang und das Ende jedes Segments werden als Ankerpunkte bezeichnet und Pfade können geschlossen oder geöffnet sein. Die Form des Pfads kann durch die Griffpunkte an den Ankerpunkten modifiziert werden und Pfadsegmente können somit verformt werden [6]. Auf diese Weise kann man komplexe Formen für z.B. Firmenlogos oder auch eigene Schriftarten designen. Solche manuellen Pfade kann man vor allem in Adobes Illustrator, InDesign oder Photoshop erstellen.

PDF unterstützt Type-1-Fontformate, Multiple-Master-Fonts, TrueType-Fontformate, OpenType-Fontformate, Dfonts, Character Identifier Font (CID) codierte Fonts und Composite-Fonts. Seit PDF 1.3 werden CID-Schrifttypen als Abkömmlinge von Composite-Fonts unterstützt. CID ist ein Synonym für das PostScript Type-0-Format, das eine Adressierung von mehr als 256 Zeichen ermöglicht und für Fonts mit einer großen Zeichenanzahl verwendet wurde [7]. Composite-Fonts sind Basisschriften mit hierarchischem System. Die oberste hierarchische Ebene stellt den root font dar. Alle folgenden Fonts sind descendant fonts. Sie ermöglichten die Einführung von Type-1-Schriften im asiatischen Markt [8]. Falls die Schriftart nicht im Dokument eingebettet wurde, wird sie aus der Ursprungsdatei möglicherweise durch eine Ersatzschrift des Benutzersystems im PDF-Programm substituiert [8].

1.1.3 Bilder

Generell sollte für das Bearbeiten von Bildern ein externes Bildbearbeitungsprogramm verwendet werden, z.B. Photoshop oder das kostenlose Gimp. Dafür kann für die Bearbeitung in Photoshop das Bild mittels Acrobat Pro aus dem PDF extrahiert werden und später wieder im in Acrobat Pro geöffneten PDF ersetzt werden. Vektorgrafiken als Pfadobjekte und Rasterbilder als Pixelobjekte (Bitmap) können nach Auswahl verschoben, gelöscht, skaliert, verzerrt, gespiegelt, gedreht, die Deckkraft verändert, beschnitten oder ersetzt werden [8]. Der Mehrgewinn an Vektorgrafiken liegt in dessen Eigenschaften, dass sie auflösungsunabhängig sind, da sie beliebig groß ohne Qualitätsverlust skaliert werden können und dass sie wesentlich weniger Speicherplatz benötigen als Rasterbilder.

Die Auflösung von Bildern kann in Acrobat neu berechnet werden. Niedrig aufgelöste Bilder behalten ihre Auflösung bei. Ein guter Neuberechnungsalgorithmus heißt bikubische Neuberechnung. Bei Schwarzweißbildern kann eine Neuberechnung zu unschönen Artefakten führen [9]. Generell führt eine Neuberechnung der Auflösung in Bildbearbeitungsprogrammen

zu besseren Ergebnissen als in Acrobat. Etwaige Pixelbearbeitungen wie Tonwertkorrekturen oder das Schärfen von Bildern können ausschließlich in Bildbearbeitungsprogrammen vorgenommen werden.

1.1.4 3D-Daten

PDFs mit 3D-Inhalten bestehen aus dem U3D-Flächenmodell oder dem BREP/Flächenmodell PRC. Diese Flächenmodelle werden vorwiegend bei der Visualisierung von Computer-Aided Design (CAD) Daten verwendet. Beide Formate können im Adobe Reader angezeigt, animiert, geschnitten und gemessen werden. Viele Drittanbieter PDF-Reader und die PDF-Viewer im Browser können eingebettete 3D-Daten meist nicht darstellen. Einige CAD-Programme ermöglichen einen 3D-PDF-Export oder Import [2].

1.1.5 Kommentare

Ein Kommentarobjekt, das mit Dokumentenseiten verlinkt ist, besteht aus 2 technisch separaten Bausteinen. Zum einen werden Kommentare durch ein grafisches Element auf den zugehörigen Seiten symbolisiert, zum anderen wird der Kommentarinhalt in einem rechteckigen Kommentarbereich dargestellt. Ein Anwender kann die Darstellung des Kommentarobjekts je nach Geschmack modifizieren. Unüblicherweise kann ein Kommentar sogar als Video-Kommentar abgespielt werden. Die wichtigsten Kommentartypen sind Notizzettel, Textmarkierung, Stempel, Wasserzeichen, Textboxen, Formen, Freihand-Markierung, Audio, Video und 3D-Illustrationen. Kommentare können optional mit ausgedruckt werden [10].

1.1.6 Verweise

Technisch gesehen sind Verweise oder Hyperlinks spezialisierte Kommentare ohne Symboldarstellung. Auf der Seite wird ein Ausschnitt zur Platzierung des Verweises gewählt, der über einem Inhaltselement (Text oder Bild) liegt. Der Verweis zeigt auf eine Seite oder Seitenbereich im geöffneten Dokument, eine andere PDF-Datei, eine E-Mailadresse oder URL. Man kann sogar Zielobjekte mit einem im gesamten Dokument eindeutigen Namen einstellen [10].

1.1.7 Formulare

In PDFs kann man Formularfelder vom Typ Textfeld, Kontrollkästchen, Auswahlknopf, Kombinationsfeld, Auswahlliste, Schaltfläche, Barcode- oder Unterschriftsfeld erstellen. Ein Formularfeld ist ein Objekt zum befüllen und speichern von Felddaten. Die unterschiedlichen Formularfeldtypen weisen verschiedene Eigenschaften in Bezug auf Interaktivität und Gestaltung auf. Jedes Feld hat einen eindeutigen Namen im gesamten Dokument. Mit diesem unikalen Namen können Namensgruppen realisiert werden. Durch eine hierar-

chische Struktur mittels Teilnamen, die mit einem Punkt voneinander getrennt sind mit dem äußersten Gruppennamen zuerst geschrieben werden, können Felddaten noch besser und logischer beschrieben bzw. strukturiert werden. Jedes Feldobjekt geht Hand in Hand mit einem Widget, welches ein spezielles Kommentarobjekt zur Steuerung darstellt. Diese Widgets stehen für Werte oder Zustände der Felder und sind dafür verantwortlich, dass man Formulare im PDF-Dokument mit dem Computer, Tablet oder Smartphone ausfüllen kann. Außerdem ist es möglich unsichtbare Feldobjekte, die ohne das Widget platziert werden können, zu erstellen, um die PDF-Software anzusprechen. Häufiger werden mehrere Widgets mit einem Feldobjekt gekoppelt [10].

Um elektronisch ausfüllbare Formulare zu verwenden müssen zusätzlich in Acrobat Formularfelder auf die entsprechenden Stellen platziert werden. Falls ein Listenfeld verwendet wird, sollte man eine Schrift für die Listeneinträge im PDF einbetten. Formulare können einen druckbaren und nicht druckbaren Teil enthalten. In der Druckvorstufe müssen vor dem Druck alle Formularfelder eliminiert werden, damit alle Schriften eingebettet werden können [8]. Es gibt 2 verschiedene Möglichkeiten von PDF Formularen: AcroForms (Acrobat Forms) oder Adobes proprietäre XML Forms Architecture (XFA) forms, welche mit Version 2.0 von der ISO als veraltet markiert wurden. XFAs Haupterweiterungen zu Extensible Markup Language (XML) sind rechnergestützte, aktive Tags und sein Datenformat ist kompatibel mit anderen Systemen, Anwendungen und Technologiestandards [11]. XML ist eine Sprache zur Markierung von Inhalten mit Hilfe von Tags, um die Struktur zu beschreiben und Elemente zu identifizieren [8]. AcroForms unterstützen das Abschicken (submit), Zurücksetzen und Importieren von Daten. Die submit-Aktion transferiert die Namen und Werte eines ausgewählten interaktiven Formularfelds zu einer vordefinierten URL. In der Praxis werden Formulare in einem Grafik- oder Layoutprogramm gestaltet und als PDF exportiert.

1.1.8 Incremental Update

Die ursprüngliche Version einer PDF-Datei bleibt erhalten, während das incremental update die Änderungen im Dokument enthält. Professionelle PDF-Programme können ähnlich einer Versionsverwaltung jede geänderte Version des Dokuments laden. Bei einfacheren PDF-Programmen wird lediglich die letzte Version geladen. Bei Verwendung von incremental updates kann man digital unterschriebene Dokumente ändern ohne dass die Unterschrift ungültig wird, da die Dokumentversion mit der digitalen Unterschrift ein andere Version ist als die nachträgliche Änderung. Dabei muss die digitale Unterschrift als incremental update gespeichert werden, sonst würde sie bei nachträglicher Dokumentenmodifikation unabhängig von der Änderungsart verfallen. Folglich sollten mehrfach signierte Dokumente ebenfalls mit der Option incremental update gespeichert werden [10]. Pro incremental update steigt der Speicherbedarf einer PDF-Datei.

1.1.9 Kompression

PDF-Dateien sind komprimiert und haben üblicherweise einen Bruchteil der Größe des Ursprungsformats. Dies wird durch Vermeidung von Redundanzen, Erhöhung der Entropie (Zeichendichte) und Weglassen von Informationen bewerkstelligt. Im Allgemeinen gibt es verlustlose und verlustbehaftete Kompression. Die Kompressionsalgorithmen Run Length Encoding (RLE), die genauso effiziente LZW, Flate-Komprimierung, ZIP und CCITT gehören zur verlustfreien Kompression. Zur verlustbehafteten Kompression zählen JPEG, JBIG2 und JPEG2000 [8]. Kompressionsalgorithmen sind nicht auf bestimmte Dateiformate beschränkt. In PDF können die folgenden Kompressionsalgorithmen für Bilder verwendet werden: IP, RLE, JPEG, JPEG2000, CCITT und JBIG2. Eine hohe Bildqualität im PDF bedeutet eine größere Datei. Faktoren, die die Bildqualität beeinflussen, sind Breite x Höhe des Bildes, Farbtiefe, Farbraum und die Kompressionsmethode [10].

Außerdem ist es möglich eine Datenreduktion durch Neuberechnung zu erzielen. Hierbei wird das verlustbehaftete Downsampling verwendet und führt häufig zu nicht befriedigenden Ergebnissen. Es gibt als Neuberechnungsmethoden die eher im Ergebnis mangelhafte Kurzberechnung, sowie die besseren durchschnittliche und bikubische Neuberechnungen. Neuberechnungen in Photoshop führen generell zu besseren Ergebnissen als in Adobe Acrobat Distiller [8].

PDF-Dateien können zur Weboptimierung serialisiert (linearisiert) werden, sodass Teile des PDFs während des Ladevorgangs dargestellt werden. Liegen unkomprimierte Elemente im Dokument vor, werden diese beim Speichern durch die Flate-Komprimierung, die auch den ZIP-Algorithmus verwendet, komprimiert.

1.1.10 Ebenen

Ebenen werde auch als Optional Content Layers bezeichnet und stellen quasi mehrere Inhaltsschichten auf einer einzelnen PDF-Seite dar, wobei jede Seite im Dokument beliebig viele Ebenen enthalten kann. Jede Ebene kann PDF-Inhalt sozusagen logisch gruppieren und die Bearbeitung von Inhalten auf einer Ebene wirkt sich nur auf diese Ebene aus. Man kann Inhalte auch mehreren Ebenen zuordnen oder keiner Ebene. Ebenen können ein- und ausgeblendet, ihre Reihenfolge verändert, gesperrt, zusammengeführt, aus anderen PDF-Dateien importiert und für unterstützende Dateiformate von Adobeprogrammen, z.B. Photoshop, Illustrator oder InDesign, exportiert werden. Zusätzlich kann man eine Ebenennavigation mit Hilfe von Links und Lesezeichen konstruieren, um Ebenensichtbarkeit für den Betrachter zu steuern [12].

1.1.11 Portfolio

Ein Portfolio bezeichnet eine Datei bestehend aus anderen Dateien, die kein Hauptdokument enthält, sondern lediglich eine Pseudo-Seite. Diese Pseudo-Seite wird von Portfolio

inkompatiblen PDF-Programmen angezeigt. Zusätzlich können andere PDF-Dateien und andere Dateiformate im PDF-Hauptdokument eingebettet werden [10].

1.1.12 JavaScript

In PDF kann man Ereignisse Aktionen zuordnen, d.h. bei Eintreffen eines Ereignisses wird automatisch eine Aktion ausgeführt. Ein Ereignis ist eine bestimmte Statusänderung von Objekten oder ein interaktives Anwendererereignis. Dabei kann man als Aktion JavaScript-Code aufrufen, dessen Aktion mit Lesezeichen, Verweisen, Seiten und Dokumentereignissen verknüpft ist. Auf Formularfeldern kann ebenfalls JavaScript angewandt werden [10]. Diese JavaScript-Erweiterung für Acrobat ist eine proprietäre Technologie von Adobe. Viele andere nicht-Adobe PDF-Programme bieten keine Unterstützung für JavaScript [1].

1.2 PDF Dateiformate

PDF hat zahlreiche Dateiformate, von denen die meisten standardisiert wurden, hervorgebracht. Jedes Dateiformat ist einem individuellen Anwendungsbereich zugeordnet und adressiert spezifische Industriebranchen: PAdES für elektronische Signaturen, PDF/X für den professionellen Druck, PDF/A für die Archivierung, PDF/E für den Ingenieurbereich, PDF/H für das Gesundheitswesen, PDF/VT für den Druck mit variablen Daten, PDF/UA für Barrierefreiheit, PDF/R für gescannte Dokumente und Durchsuchbare PDFs für Stichwortsuche. Im Folgenden stelle ich jedes Format vor und beschreibe seine speziellen Merkmale.

1.2.1 PAdES

PDF Advanced Electronic Signatures (PAdES) ergänzt den Funktionsumfang um Werkzeuge mit denen man elektronische Signaturen erzeugen, anpassen und prüfen kann. Es wurde vom European Telecommunications Standards Institute (ETSI) veröffentlicht, 1999 in PDF 1.3 eingeführt und basiert auf der ISO 32000-1 Spezifikation. Nachfolgend wurde dessen Konzept weiterentwickelt. PAdES erweitert PDF um kryptographische Techniken und ermöglicht sichtbare und unsichtbare digitale Signaturen. Resultierend soll dieses Dateiformat die Integrität, Authentizität, Verbindlichkeit und Rechtssicherheit von digital signierten PDF-Dokumenten herstellen. PAdES implementiert verschiedene Signaturformate wie CMS Advanced Electronic Signatures (CAdES) und XML Advanced Electronic Signatures (XAdES), unterstützt Zeitstempel und die Validierung des Zertifikatwiderrufsstatuses. Zertifikatbasierte Signaturen sollen die Identität des Unterzeichners und die Unabänderlichkeit des Dokuments sichern. Eine zertifizierte PDF-Datei ermöglicht die Umsetzung bestimmter Nutzungsrechte, wie eingeschränkte Bearbeitung, Ausfüllen von Formularen oder gesperrtes Drucken. Eine elektronische Unterschrift kann mit dem Programm Adobe Acrobat Sign erstellt werden [13].

1.2.2 PDF/X

Speziell für den simpleren Datenaustausch in der Druckvorstufe und der professionellen Druckindustrie wurde PDF/X (Exchange) als ISO 15930:2001 entwickelt. Dieser erst 2001 publizierte Dateiformatstandard beschreibt die speziellen Eigenschaften von Druckvorlagen und vereinfacht die Datenübermittlung von der Design-Agentur und Druckvorstufe bis zum finalen Druck. Besonderen Wert wurde darauf gelegt, dass in offenen Dateiformaten aus Layoutprogrammen keine Informationen über Farbe und Schrift verloren gehen und einer Verfälschung im Druckergebnis vorgebeugt werden kann [14]. Die Entwicklung von PDF/X zielt auf eine Verminderung von Druckfehlern und Mehraufwand in der Druckerei. In der Umsetzung bedeutet das, dass Elemente, die sich nicht sinnvoll drucken lassen, z.B. Video und Audio, nicht berücksichtigt werden. PDF/X-kompatibel bezeichnet die Eigenschaft von Dokumenten, dass sie ohne vorherige Prüfung von der Druckerei direkt verwendet werden können [15].

Beschnitt, Farbangaben und verwendete Schriften sind u.a. für den Druck notwendig und sollten verwendet werden. Qualitätsanforderungen, die sich auf bestimmte Druckverfahren beziehen, sind nicht implementiert, sondern werden abstrakter definiert. Besondere Qualitätsanforderungen liegen vor allem im Zeitungsdruck, Akzidenzdruck oder Bilderdruck vor. Vielmehr geht es bei PDF/X darum, Grundvoraussetzungen für den Druck sicherzustellen, z.B. ob der richtige Farbraum gesetzt wurde oder korrekte Einstellungen für Überdrucken und Überfüllung vorliegen. Neben Aussparen und Unterfüllung werden Überdrucken und Überfüllung zum Oberbegriff Trapping zusammengefasst. Bei der Überfüllung werden bei verschieden farbigen Objekten das hellere Objekt auf dunklem Hintergrund minimal vergrößert, sodass es das dunklere Objekt leicht überlappt. Solche Überlappungen nennt man Traps. Dies beugt weißen Blitzern (Papierweiß scheint durch) beim Druckergebnis vor. Die umgekehrte Vorgehensweise wird bei der Unterfüllung angewendet. Liegt ein dunkles Objekt auf hellem Hintergrund, so wird das helle Objekt an den Rändern zur dunkleren Farbe hin verengt. Vordergrundobjekte stehen in Layout- und Grafikprogrammen standardmäßig auf Aussparen. Dessen Fläche wird im Hintergrund ausgeschnitten, um unerwünschte Farbmischungen zu vermeiden. Im Falle von schwarzen oder sehr dunklem Text auf farbigem Hintergrund sollte man Layoutprogramm diese Vordergrundelemente überdrucken lassen [16]. Generell werden schwarze Schrift oder Linien im Drucker durch 3 oder 4 Farben zusammengesetzt und fehlende Schriften werden häufig durch den Font Courier kompensiert. Im Druck sollten keine verlustlosen Kompressionsalgorithmen für Bilder verwendet werden wie JPEG, da Artefakte auftreten können. Ebenso gibt es keine automatischen Einstellungen für passende Auflösungen von Vollton-, Halbton- oder Strichbildern [15]. Eine Farbe mit 100 % Deckkraft wird als Volltonfarbe bezeichnet. Halbtöne sind Farben mit geringerer Deckkraft [17]. Als Volltonfarben werden auch speziell vorgemischte Druckfarben bezeichnet die anstelle von oder zusätzlich zu den üblichen Prozessdruckfarben in CMYK verwendet werden. Für Volltonfarben ist eine eigene Druckplatte in der Druckmaschine vonnöten [18]. Der CMYK-Farbraum wird im Druck verwendet durch subtraktive Farbmischung mit den Farben Cyan, Magenta, Yellow und Key (Schwarz). Einige Farben können nicht von

CMYK reproduziert werden, dann spricht man von Sonderfarben. Strichbilder sind Bilder mit ausschließlich weißen und schwarzen Partien [19].

Bei allen PDF/X-Dateivarianten außer den Varianten, die ein p enthalten, muss ein Output-Intent im Katalog eingetragen sein. Jede Seite einer PDF/X-Datei muss mindestens die TrimBox oder ArtBox, aber nicht beide gemeinsam, beinhalten. JavaScript bzw. Aktionen, Open Prepress Interface (OPI) Kommentare und Verschlüsselung sind verboten [8].

Es gibt verschiedene Varianten von PDF/X, die jeweils einen verbesserten Farbspielraum ermöglichen.

PDF/X-1a

PDF/X-1a ist eine Subvariante von PDF/X-1 (Norm ISO 15930-1). In der a-Version sind lediglich CMYK, Graustufen, Schwarz-Weiß (Bitmap) und Sonderfarben möglich und es muss eine Composite-Datei vorliegen. Vorseparierte Dateien sind nicht zulässig. Farben können nicht auf Grundlage von International Color Consortium (ICC) Profilen bei PDF/X-1a definiert werden [8], [15]. Erst seit PDF 1.3 werden ICC-Profile unterstützt, die die Farbeigenschaften, Helligkeit, Weißpunkt, Gammakurve und Farbumfang eines bestimmten Monitors eines spezifischen Geräts beschreiben, sprich ein ICC-Profil definiert wie Farben vom Zielgerät dargestellt werden können. Außerdem wird die Transformation zwischen dem Gerät und dem Profilverbindungsraum Profile Connection Space (PCS) definiert. Dabei gibt es die Variante Eingabeprofile für Kameras oder Scanner und Ausgabeprofile für Monitore oder Drucker. Zweck des ICC-Profils ist möglichst Farbübereinstimmungen zwischen verschiedenen Geräten zu erzielen [20]. Beim PCS handelt es sich um ein neutrales Farbmodell im ICC-Colormanagement, welches den Quellfarbraum mit dem Zielfarbraum verbindet und somit geräteunabhängig ist. Der PCS kann entweder der LAB oder XYZ Farbraum sein [21].

Es können TIFF-, TIFF/IT-, EPS- und Desktop Color Separation (DCS)-Dateien integriert werden. Transparenzen, Ebenen, LZW- und JBIG2-Kompression, Formularfunktionen und interaktive Elemente dürfen nicht verwendet werden bzw. sind nicht implementiert. Dieser Standard wurde von ISO 15930-1:2001 auf ISO 15930-4:2003 überarbeitet. Lediglich in der überarbeiteten Version, die die Version von 2001 ersetzt, werden auch Sonderfarben unterstützt [8], [22].

PDF/X-2

Die 2. Variante ist als ISO 15930-6:2003 erschienen und bietet die Voraussetzungen zum farbigen Qualitätsdruck wie Farbmanagement, CMYK- und Sonderfarbdaten in beliebiger Kombination. PDF/X-2 hat in der Praxis wegen der Unvollständigkeit der PDF-Datei für den Datenaustausch in der Druckvorstufe keinerlei Relevanz. Dieser Standard wird durch PDF/X-5 vollkommen ersetzt [8], [22].

PDF/X-3

Zusätzlich erweitert Version 3 als ISO 15930-3:2002 [22] um die Farbräume RGB und LAB, sowie alle ICC-Profile. Möglicherweise wird in der Druckvorstufe der im Dokument eingestellte Farbraum in CMYK umgewandelt. Es findet eine automatische Tranzparenz- und Ebenenreduzierung statt [15]. Bei der Transparenzreduzierung werden einzelne Bildsegmente in vektorbasierte und gerasterte Bereiche unterteilt, überlappende Bereiche der Transparenzen zerschnitten und auf einer Ebene reduziert [23], [24].

PDF/X-4

Transparenzen, Ebenen, JPEG2000 und OpenType-Schriften können in dieser PDF/X-Variante als ISO 15930-7:2008 [8], [22] verwendet werden, wodurch sie für das Bedrucken von Textilien besonders gut geeignet ist [15]. Es gibt außerdem die PDF/X-4p (Profil) Variante [8].

PDF/X-5

PDF/X-5 wurde als ISO 15930-8:2010 als vorletzter Standard des PDF/X-Formats verabschiedet und inkludiert externe Elemente und Multichannel-ICC-Profile [22]. Multichannel-Profile unterstützen mehr als 4 Farbkanäle und können somit für Drucker mit mehr als 4 Druckpatronen eingesetzt werden [25]. Außerdem gibt es die folgenden Varianten: PDF/X-g (external graphic content), PDF/X-pg (external ICC color profiles and external graphical content) und PDF/X-gn (n-colorant ICCprofile) für den mehrfarbigen Verpackungsdruck [8].

PDF/X-6

Der letzte PDF/X-Standard als Version 6 wurde in ISO 15930-9:2020 offenbart und basiert auf dem PDF 2.0 Standard. In dieser Version sind neben maßgeblichen Neuerungen für die heutigen Print-Anforderungen Lockerungen im Vergleich zu vorherigen PDF/X-Standards eingeführt worden. Die wichtigsten Neuerungen sind Parameter für Tiefenkompensierung, separate Ausgabebedingungen, DPart Metadaten, Informationen zu Sonderfarben im CxF/X-4 Standard und Mixing Hints [22]. Tiefenkompensierung ist eine rechnerische Korrektur und wird bei der Konvertierung eines Farbraums mit großem Tonwertumfang in einen mit kleinem Tonwertumfang angewendet. Dabei soll die ursprüngliche Differenzierung der Tonwerte dunkler Bildteile beibehalten werden können [26]. DPart-Metadaten wurden ursprünglich für den PDF/VT-Standard spezifiziert und ermöglichen mehrteilige PDF-Dateien in Datensätze zu unterteilen. Diese PDF-Dateien können dann automatisch verarbeitet werden [27]. Mixing Hints enthalten Informationen, z.B. wichtig bei Prüfdruck, über erwartete Ergebnisse, wenn mehrere Volltonfarben im Druck miteinander interagieren. Diese Informationen sind

beispielsweise für einen Prüfdruck wissenswert [28].

Zu den Lockerungen zählen die Möglichkeit von Notizen und grafischen Anmerkungen, strukturelle Aktionen, Formularfelder und digitale Signaturen. Außerdem besteht dieser Standard aus 2 Konformitätsstufen: PDF/X-6p zur Referenzierung von ICC-Profilen und PDF/X-6n für Multicolor-Profile [22].

1.2.3 PDF/A

Das PDF/A Dateiformat (Archivable) wurde zur gesetzeskonformen Langzeitarchivierung von digitalen Dokumenten entwickelt und solche Dokumente sind zunächst schreibgeschützt. Der ISO-Standard definiert die Konformität der Form von Elementen wie Schriften oder Layout für eine Langzeitarchivierung. Dadurch ist die Lesbarkeit der Dokumente über lange Zeiträume gesichert und die Bedingungen einer revisionssicheren Archivierung gewährleistet [29]. Revisionssichere Archivierung bedeutet, dass gespeicherte Daten vor nachträglichen Modifikationen, Fälschung oder Manipulation geschützt sind [30]. Der Fokus in diesem Dateiformat liegt auf langfristige und einfache Speicherung der PDF-Dateien. Folglich ist die Einbettung von Audio und Video nicht implementiert, aktive Komponenten wie Links, sowie externe Ressourcen wie Grafiken und Schriftarten werden nicht unterstützt, sondern müssen direkt eingebettet werden. Ebenso können Dokumente nicht verschlüsselt werden. Die Einbettung von Metadaten als Extensible Metadata Platform (XMP) wird unterstützt, was die Identifizierung und Suche von Dokumenten erleichtert. Es gibt einige Nachteile von PDF/A. Nicht alle Dokumente können problemlos in dieses Dateiformat umgewandelt werden, wie beispielsweise Dokumente mit Audio, Video oder JavaScript. Nach solch einer Konvertierung zu PDF/A kann es zu Fehlern in der visuellen Darstellung kommen und die Dateigröße kann enorm werden, da alle Elemente direkt eingebettet werden müssen [29].

PDF/A-1

Seit der ersten Version von PDF/A ist das Dateiformat in die ISO-Norm als PDF/A-1 in ISO 19005-1:2005 übernommen worden [22]. Die Originalversion stellt sicher, dass alle externen Quellen wie Schriften oder Bilder eingebettet sind, unterstützt digitale Signaturen und Hyperlinks. PDF/A-1 ist abwärtskompatibel. Es gibt 2 Qualitätsebenen von PDF/A-1: PDF/A-1b (Basic) und PDF/A-1a (Accessible). Die Basic Variante legt Wert darauf, dass Dokumente eindeutig visuell reproduzierbar sind und Accessible ist zusätzlich für Barrierefreiheit optimiert. Bei Accessible können Text und inhaltliche Struktur von einem Screenreader durch Tagged PDF vorgelesen werden [29]. Des Weiteren werden, Sprach-Angabe und Unicode-Mappings unterstützt [22].

PDF/A-2

Im Jahr 2011 wurde die PDF/A-2 Version als ISO 19005-2:2011 auf den Markt gebracht. Sie ermöglicht die Kompression von Grafikformaten mit JPEG-2000, Transparenzen, PDF-Ebenen, Portfolios, Object Level XMP Metadaten, Kommentartypen, Annotationen und digitale Signaturen [22]. PDF/A-1-Dateien können in PDF/A-2-Dateien eingebunden werden. Es gibt 3 Varianten von PDF/A-2: PDF/A-2b (Basic), PDF/A-2u (Unicode-Textsemantik) und PDF/A-2a (Accessible). Basic gewährleistet das unveränderte Erscheinungsbild eines Dokuments und definiert die Mindestanforderungen. Die Unicode-Version ergänzt um Unicode-Unterstützung und Indexierung. Accessible setzt alle Anforderungen der ISO-Norm 19005-2 um [29].

PDF/A-3

Ein Jahr später wurde PDF/A-3 im Standard ISO-19005-3:2012 veröffentlicht. Er basiert auf PDF 1.7 und ermöglicht die Einbettung dynamischer, zur Laufzeit interpretierbaren Komponenten und Dateiformate. Gleichfalls definiert PDF/A-3 die Konformitätsebenen 3b, 3u und 3a. Die u-Variante bietet eine Vereinfachung in der Durchsuchbarkeit von Texten und das Kopieren von Unicode-Text [22].

PDF/A-4

Viel später im Jahr 2020 wurde PDF/A-4 als ISO 19005-4:2020 herausgebracht. Dieser Standard basiert auf der PDF 2.0 Dateiversion. Sie spezifiziert 2 neue Konformitätsebenen: PDF/A-4f für nicht-PDF/A konforme Dateianhänge und PDF/A-4e für Einbindung von 3D-Inhalten in den Formaten U3D oder PRC für den Engineering-Bereich [22].

1.2.4 PDF/E

PDF/E (Engineering) gilt als international standardisiertes Austauschformat als Norm ISO 24517 für technische Dokumente und wird im Maschinenbau, in der Fertigung und im Baugewerbe für Fertigungspläne, Konstruktionszeichnungen oder technische Dokumentationen verwendet. Das PDF/E Dateiformat von 2008 ist speziell für das Ingenieurwesen entworfen und kann interaktive 3D-Elemente und Animationen darstellen. Im einzelnen können CAD-Dateien im 3D- und 2D-Format eingebettet werden. Die 3D-Elemente können im Dokument ausgeklappt oder gedreht werden. Metadaten und interaktive Funktionen wie Lesezeichen, Formulare oder Hyperlinks werden unterstützt [14].

1.2.5 PDF/H

Das PDF/H (Healthcare) Dateiformat soll im Gesundheitswesen Patientendaten erfassen, austauschen und archivieren. Hierbei wird besonders Wert auf die Anforderungen des Datenschutzes in gesundheitsspezifischen Ämtern, Institutionen und Arztpraxen gelegt. PDF/H wurde zwar im Jahr 2008 kreiert, jedoch wurde es nicht in den Normierungsprozess der ISO eingebunden [22]. Es handelt sich eher um eine Best-Practice für die Umstellung von Papiergesundheitsakten auf PDF als e-Akte. Die Digitalisierung soll gesundheitsbezogene Daten strukturieren, verwalten und so präsentieren, dass Forscher*innen und Beschäftigte im Gesundheitswesen effizienter auf sie zugreifen können.

1.2.6 PDF/VT

Basierend auf PDF/X wurde PDF/VT als spezielles Austauschformat im variablen Datendruck (Variable Data) und Transaktionsdruck (Transactional Printing) im Jahr 2010 auf den Markt gebracht [31]. Wiederkehrende Elemente wie Texte, Grafiken oder Bilder sollen effizienter verarbeitet und an den Drucker übertragen werden können [14]. Variabler Datendruck bezeichnet ein digitales Druckverfahren bei dem einzelne Parameter von Printprodukten individuell variiert werden können, wobei das Grundlayout beständig bleibt. Folglich können große Mengen von Printprodukten mit Personalisierung hergestellt werden, z.B. Werbebriefe mit konstanten grafischen Elementen wie Firmenlogos und individuellen Namen der Kund*innen. Dadurch können Firmen ihr Corporate Identity-Layout behalten und ihre Kund*innen persönlicher ansprechen. Der Begriff Transaktionsdruck definiert das Drucken von Transaktionsdokumentationen wie Rechnungen, Mahnungen, Lieferscheine oder Quittungen im Waren- und Dienstleistungssektor. Herausstechend ist, dass PDF/VT große Mengen an variablen Daten in einer einzigen PDF-Datei speichern kann, wobei es immer einen Satz von statischen und variablen Daten gibt. Diese Vorgehensweise spart Zeit, Kosten und reduziert Fehler. Vorteilhafterweise werden ICC-Profile unterstützt [31]. Die erste Version als ISO 16612-1:2005 konnte sich auf dem Markt nicht durchsetzen [22].

PDF/VT-2

Die zweite Version als ISO 16612-2:2010 implementiert die Verwendung externer grafischer Inhalte und das Streamen von mehrteiligen Multipurpose Internet Mail Extension (MIME) Paketen in der Version PDF/VT-2s. MIME-Typen (media types) zeigen Art und Format von Dokumenten, Dateien oder einem Sortiment von Bytes an und sind standardisiert. Ein MIME-Typ besteht üblicherweise immer aus einem durch ein / getrennten Typ und Subtyp. Der Typ repräsentiert die generelle Kategorie des Datentyps und der Subtyp die exakte Datenart des Typs. Jeder Typ hat seine eigene Menge an Subtypen [32]. Zum Lesen von PDF/VT-2 Dokumenten wird ein PDF/X-4- bzw. PDF/X-5- oder PDF/VT-konformer PDF-Reader benötigt [22].

PDF/VT-3

Spezialisiert auf die Integration von variablen Daten (DPart-Metadaten) und den Transaktionsdruck ist die dritte PDF/VT Variante als ISO 16612-3:2020 [22].

1.2.7 PDF/UA

Das PDF/UA (Universal Accessibility) Dateiformat dient der Erstellung barrierefreier Dokumente. Die PDF/UA-Kennzeichnung stellt eine Klassifizierung für barrierefreie Dokumente dar und orientiert sich an den Anforderungen der Web Content Accessibility Guidelines (WCAG) 2.0 des World Wide Web Consortiums. Als Rechtliche Grundlage dient die Barrierefreie-Informationstechnik-Verordnung (BITV) 2.0. Um die Anforderungen der PDF/UA-Kennzeichnung zu erfüllen, müssen Dokumente bestimmte technische und inhaltliche Vorgaben erfüllen. Auf Basis des Matterhorn-Protokolls, welches aus 31 Prüfpunkten und 136 Konformitätskriterien besteht, müssen Aufbau von Texten, Bildern, Listen, Tabellen und Formularefeldern festgelegte Vorgaben erfüllen. Diese Konformitätskriterien können auf der einen Seite nur von einer Software geprüft werden und auf der anderen Seite nur von Menschen.

Menschen mit Einschränkungen sollen das Dokument optimal nutzen können. Zur Erleichterung des Verständnisses sollten Überschriften, alternative Texte für Bilder, Beschreibungen für Tabellen, Tags und eine klare Lesereihenfolge verwendet werden [33]. Es gibt die PDF/UA-1 Version aus ISO 14289-1:2012 und die überarbeitete Version PDF/UA-1 aus ISO 14289-1:2014, die erst 2020 als gültig erklärt wurde [22].

1.2.8 PDF/R

Speziell für die Speicherung, den Transport und Austausch von gescannten Dokumenten gibt es das ISO 23504-1:2020 standardisierte Format PDF/R-1. Es bietet die grundsätzlichen Funktionalitäten von TIFF, bitonalen, Graustufen- und Echtfarbbilder [22]. Bitonale Bilder bestehen lediglich aus einer Vordergrund- und Hintergrundfarbe.

1.2.9 Searchable PDF

Searchable PDFs können mit Suchfunktionalitäten eines PDF-Readers durchsucht werden. Es kann gezielt nach Zahlen oder Stichwörtern durchsucht und Inhalte können zur Bearbeitung in anderen Programmen kopiert werden. Man erkennt durchsuchbare PDFs daran, dass man den Text markieren kann. Diese PDF-Art wird üblicherweise durch die Optical Character Recognition (OCR) Technologie erstellt. Bei OCR handelt es sich um optische Zeichenerkennung, die Textzeichen und Dokumentstruktur analysiert. Auf diese Weise können gescannte Dokumente oder Pixelbilder als PDF abgespeichert und in ein Durchsuchbares PDF in Acrobat umgewandelt werden. Während der Umwandlung wird

dem Dokument eine zusätzliche unsichtbare Textebene, die unter der Bildebene liegt und durchsuchbar ist, auf der Seite hinzugefügt. In Acrobat ist es außerdem möglich mit der einfachen Suche innerhalb einer Datei nach Suchbegriffen zu suchen, mit der erweiterten Suche oder der Suchen-Werkzeugleiste mehrere PDF-Dokumente zu durchsuchen und speziell in der erweiterten Suche u.a. Objektdaten und Bildern zu lokalisieren. Textbasierte PDF-Dateien können grundsätzlich durchsucht werden und auch in andere Dateiformate wie Microsoft Word, Excel oder PowerPoint umgewandelt werden. Durchsuchbare PDFs ermöglichen Barrierefreiheit. Sie können von Bildschirmleseprogrammen für Sehbehinderte vorgelesen oder vergrößert werden[34].

1.3 PDF Dateiversionen

Gestartet mit Version 1.0 war PDF lediglich ein proprietäres Dateiformat von Adobe. Die Freigabe von PDF als offenes und kostenlose Dateiformat führte letztendlich erst zu seiner weltweiten Verbreitung und Anerkennung. Erst im Jahr 2005 entwickelte sich Version 1.4 zu einem internationalen ISO Standard. Die letzte Dateiversion 2.0 von 2017 ist schon eine Weile her und es hat sich zeitlich nur das PDF/R-Dateiformat später entwickelt als PDF 2.0.

1.3.1 PDF 1.0

PDF 1.0 wurde 1992/1993 entwickelt und ist wurde nicht normiert. 1992 wurde die Spezifikation als Buch verkauft und 1993 das der Spezifikation entsprechende digitale Format entwickelt, welches ausschließlich den RGB Farbraum darstellen konnte. Medien, die einen anderen Farbraum besitzen wurden in RGB konvertiert. In der Druckindustrie ist jedoch der CMYK-Farbraum von Bedeutung. Folglich ist PDF 1.0 nicht für den Printbereich geeignet. Damals war Adoba Acrobat 1.0 das einzige Programm mit dem man diese Dateiversion bearbeiten konnte [22].

1.3.2 PDF 1.1

Genauso ist das 1994 kreierte PDF 1.1 keine Norm und implementiert weiterhin nur den RGB Farbraum, jedoch geräteunabhängig. Zusätzlich benötigte man ein Update von Adobe Acrobat auf Version 2.0. Erstmals sind in diesem Format das Einbetten von Hyperlinks optional gebunden an Aktionen, mehrseitige Artikel und Threads, Passwortverschlüsselung und Notizen bzw. Anmerkungen erschienen [22]. Hier kann man bereits Benutzungseinschränkungen geltend machen, wie Schutz vor unerlaubtem Öffnen des PDF-Dokuments, das Sperren von Teilfunktionen, z.B. Entnahme von Texten und Bildern, sowie verbotenes Drucken. Verschlüsselt wird mit einer 40-Bit-Schlüssellänge durch den RC4-Algorithmus. TrueType-Fonts können nativ eingebettet werden und einige geräteunabhängige, d.h. Commission Internationale de l'Éclairage (CIE) basierte Farbräume, können eingestellt werden

[8]. Die Internationale Beleuchtungskommission mit der Abkürzung CIE ist eine unabhängige Non-Profit-Organisation für Licht, Beleuchtung, Farbe und Farbräume. Sie entwickelt und publiziert hierzu Standards und Messverfahren [35], [36]. Binäre Abspeicherung der PDF-Datei ist nun möglich, womit eine Reduktion der Dateigröße um 25 % erreicht werden kann [8].

1.3.3 PDF 1.2

Fast alle für die Druckvorstufe nötigen Parameter aus PostScript wurden in Version 1.2 umgesetzt. Das 1996 erschienene PDF 1.2 wurde ebenfalls nicht standardisiert, jedoch ermöglichte es erstmals den druckbaren CMYK-Farbraum und Sonderfarben zu verwenden. Des weiteren wurden interaktive Formularfunktionen, Unicode, Unterstützung der OPI 1.3 Spezifikationen und eine Druckrasterfunktion implementiert[22]. OPI ist ein Workflow Protokoll, welches in der elektronischen Druckvorstufe verwendet werden kann, um Desktop Publishing Systeme und high-end Cisco Enterprise Print System (CEPS) zu verknüpfen und optimiert die Übertragung von hochauflösenden Dateien in Netzwerken [37]. In PDF 1.2 wurden erstmalig AcroForms vorgestellt. Zusätzlich ist das Abspielen von Video und Audio ausschließlich als Link zu einer externen Datei möglich. Composite-Fonts, CID-Fonts und alle CIE-basierte Farbräume werden unterstützt [8].

1.3.4 PDF 1.3

1999 wurde PDF 1.3 mit fehlender Normierung auf den Markt gebracht und trug seinen Teil 2001 und 2002 bei zur Standardisierung des ISO PDF/X Standards. Es ist kompatibel mit PostScript 3 und bietet die Neuerungen der 2-Byte CID Schrifttypen, OPI 2.0 Unterstützung, Farbraumerweiterung um Sonderfarben durch ICC-Profile und den DeviceN-Farbraum, weiche Schatten und Farbübergänge bzw. Verläufe in einem auflösungsunabhängigen Modus (Smooth Shading), digitale Signaturen, RC4-Verschlüsselung (40 Bit in Acrobat 4 und 56 Bit in Acrobat 4.05) und JavaScript [8], [22].

Der DeviceN-Farbraum wird auch in PostScript 3 unterstützt und erlaubt die willkürliche Kombinationen von Farbkanälen beim Composite-Druck. Dokumente mit Schmuckfarben müssen auf einem Gerät mit physikalisch getrennten Kanälen für jede verwendete Schmuckfarbe ausgegeben werden. Schmuckfarben sind festgelegte Farbtöne, die nicht aus Prozessfarben gemischt werden. Sie können Kosten sparen und vermeiden Farbschwankungen. Besonders verbreitet sind HKS und Patone [16]. Folglich kann kein CMYK- oder RGB-Gerät Dokumente mit Schmuckfarben farblich korrekt darstellen. Davon sind fast alle Farbdruckersysteme betroffen, sowie die von Adobe Acrobat erzeugte Bildschirmdarstellung von PDF Dokumenten mit Schmuckfarben. Ohne den DeviceN Farbraum können Bilder mit Kombinationen von z.B. CMYK und 2 Schmuckfarben oder Schwarz und eine Schmuckfarbe nicht im Composite-PostScript und Composite-PDF wiedergegeben werden, sondern höchstens mit CMYK als Näherung [38]. Dateianlagen jeglichen Typs können in Form von Streams im Body direkt eingebettet werden. Folglich kann eine PDF-Datei

containerisiert werden. 3 weitere Boxen werden dem PDF-Format hinzugefügt: TrimBox, BleedBox und ArtBox. Für Überfüllungen in der Druckvorstufe können Traps als form XObject gespeichert werden. Bilder können auf Basis von Image XObjects für komplexere Pfadresultate maskiert werden. Eine Maske kann Sektionen eines Bildes ausblenden ohne diese zu löschen [8].

1.3.5 PDF 1.5

Im Jahr 2003 kam PDF 1.5 auf den Markt und hat sich nicht zur Norm entwickelt. In dieser Version wurden erstmals Ebenen und 16-Bit-Farbtiefe in eingebetteten Bildern implementiert. Des Weiteren wurden gesteigerte Kompressionstechniken einschließlich Objekt-Streams und JPEG2000-Kompression, sowie eine verbesserte Cross-Reference Tabelle und XRef-Streams implementiert. 12 weitere Seitenübergänge für Präsentationen, verbesserte Unterstützung für Tagged PDF und die Adobe proprietäre Technologie XFA wurden außerdem hinzugefügt [8], [22].

1.3.6 PDF 1.4

Der erste PDF ISO-Standard ISO 16612-1:2005 wurde zeitlich nach Version 1.5 verabschiedet. In diesem Format sind Transparenzen, JavaScript 1.5, bessere Integration von Datenbanken, Titel, Textblockdefinition, JBIG2-Komprimierung und 128-Bit-RC4-Verschlüsselung erstmalig integriert worden[22]. Außerdem wurde die Kennzeichnung der Ausgabeabsicht über den Output-Intent mit einer Kompatibilität zu Version 1.3 implementiert [8].

1.3.7 PDF 1.6

In diesem ISO 15930-8:2008 Standard sind erstmals folgende Technologien in diesem Format eingeführt worden: NChannel-Farbraum, welches eine Erweiterung von NDevice mit Sonderfarben bzw. Schmuckfarben ist, JPEG 2000-Kompression, Advanced Encryption Standard (AES) Verschlüsselung, direkte Einbettung von OpenType-Schriften, 3D-Daten (U3D) und XML Formulare [22].

1.3.8 PDF 1.7

Veröffentlichung am 1. Juli 2008 ist PDF in Version 1.7 als ISO 32000-1:2008 als Offener Standard definiert worden. Es wurden komplexer 3D-Objekte, Kontrolle über 3D-Animationen und Einbettung von Standard-Druckeinstellungen wie Papierauswahl, Anzahl der Kopien und Skalierung hinzugefügt[22].

1.3.9 PDF 2.0

XFA ist in PDF 2.0 als ISO 32000-2:2017 vom ISO-Gremium als veraltet markiert worden. Mehr Einstellungsmöglichkeiten und Funktionen sind diesem Format beigefügt worden: Erweiterte Definitionen der Halbtöne der Rasterung, z.B. für Flex- oder Tiefdruck, konsistente Transparenz, erweiterte Tagged-PDF Funktion für Barrierefreiheit, Definition von Sonderfarben über Spektralfarben, alternierende Reihenfolge der zu druckenden Farben, Steuerung der Schwarzpunktkompensation, AES-256-Bit-Verschlüsselung, DPart-Metadaten und Einbettung von 3D-Messungen oder Querschnittsdaten [22]. Spektralfarben bezeichnen die Regenbogenfarben, die entstehen, wenn weißes Licht durch ein Prisma gebrochen wird. Schwarzpunktkompensation stammt begrifflich aus der Bildverarbeitung. Bei diesem Prozess wird der Schwarzwert eines Bildes so angepasst, dass die Darstellung von dunkleren Bereichen verbessert wird. Ist der Schwarzwert falsch eingestellt, so können dunkle Bildbereiche grau oder „ausgewaschenërscheinen, was zu einer Reduktion an Detail und Kontrast führt [39].

1.4 PDF Implementierung

PDF ist eine vektorbasierte Page Description Language (PDL) (Seitenbeschreibungssprache) und basiert auf dem PostScript-Format. Der MIME-Type von PDF heißt application/pdf. Eine PDL beschreibt den Seitenaufbau, wie die Seite in einem Ausgabeprogramm bzw. Ausgabegerät, z.B. einem Drucker, aussehen soll. PDLs können Seiten mit Vektoren beschreiben. An den Drucker wird durch die PDL ein Datenstrom der zu druckenden Ausgabe erzeugt und an den Drucker gesendet. Der Raster Image Processor (RIP) eines Druckers wandelt die Bildschirmausgabe in die gerasterte Druckerausgabe um. Viele APIs der Hardwareabstraktionsschicht im Computer wie Graphics Device Interface (GDI) oder OpenGL können in PDL ausgeben. Speichert ein Satzprogramm den Seitenbeschreibungscode eines Dokuments in einer Datei, müssen Drucker die PDL nicht selbst verarbeiten. Eine PostScript Printer Description (PPD) Druckerbeschreibungsdatei definiert Fonts, Papiergröße, Auflösung und andere Standardeigenschaften für einen bestimmten PostScript-Drucker [40]. Im Common Unix Printing System, der Standard-Druckersteuerung von Linux hat der PostScript und der PDF-Interpreter ghostscript die Aufgabe eines RIP, d.h. er ist für die Umwandlung in die gerasterte Druckausgabe auf dem Drucker zuständig. Zudem stellen PDLs eine Schnittstelle zum Quellcode eines Dokuments bzw. zu Programmen, die Quellcode verwalten oder das Dokument formatieren können, dar. Die PDL PDF erweitert die Funktionalität von PostScript um anklickbare Links (Hypertextfunktionalität), die die Navigation im Dokument erleichtern und URLs, die sich automatisch im Browser öffnen [41].

1.4.1 PostScript

Sowohl die PostScript als auch PDF haben zum Ziel die Seiten eines Dokuments vollständig für die Ausgabe in der Druckvorstufe zu beschreiben. Die abwärtskompatible, stackorientierte, Turning-vollständige Hochsprache PostScript als PDL wurde in den 1980er Jahren von Adobe erfunden [42], [43]. Hinzu wurden weitere PostScript-Technologien entwickelt, die aus der Programmiersprache PostScript, Grafik-, Textformatierungsanwendungen, Treibern und Abbildungssystemen bestehen. PostScript hat sich als Industriestandard etabliert. Die letzte Version ist PostScript 3 von 1997. Seine primäre Anwendung gemäß des Adobe Imaging Models findet sich in der Beschreibung von Textdarstellung, graphische Formen und Bildern auf gedruckten oder auf dem Bildschirm angezeigten Seiten. Dabei ist die Beschreibung des Dokuments geräteunabhängig und eine PostScript-Datei ist sequentiell organisiert. PostScript unterstützt unter anderem beliebige geometrische Formen, Zeichenoperationen in Graustufen, RGB, CMYK und CIE (Yxy-Farbraum), vorinstallierte oder benutzerdefinierte Fonts, Digitalbilder jeglicher Auflösung je nach Farbmodell und ein allgemeines Koordinatensystem.

In PostScript wird eine Seite, die ein Koordinatensystem umspannt, als Grafik betrachtet, die verschiedene Grafikelemente enthalten kann. Dabei werden die Textzeichen eines Fonts, gemäß des Adobe Imaging Models, als graphische Formen betrachtet auf denen Grafikoperationen möglich sind. Das Koordinatensystem unterstützt alle linearen Transformationen, die auf alle Seitenelemente angewandt werden können. Die Seitenbeschreibung in PostScript kann auf jedem Gerät, was einen PostScript-Interpreter implementiert, gerendert werden. In diesem Prozess wird die high-level PostScript-Beschreibung in low-level Rasterdatenformate für das jeweilige Gerät übersetzt. Jede PostScript-Datei muss durch einen RIP interpretiert und die Dateien können in American Standard Code for Information Interchange (ASCII) vorliegen [42]. Der PostScript-Interpreter als RIP rechnet die Benutzerkoordinaten in Gerätepixel um, wobei auch die technischen Eckdaten des jeweiligen Geräts mitberücksichtigt werden. Theoretisch kann derselbe PostScript-Code auf verschiedenen Endgeräten mit unterschiedlicher Auflösung eine mehr oder weniger identische Ausgaben erreichen. Den Interpreter gab es früher als Hardware-RIP, der allerdings nicht mehr zum Einsatz kommt. Heute gibt es lediglich Software-RIPs, die von einem Betriebssystem kontrolliert werden und hardwareunabhängig arbeiten. Fast alle RIP-Hersteller orientieren sich am Standard und somit sind PostScript-Fehler in der Druckvorstufe minimiert worden [8].

1.4.2 Adobe Imaging Model

PDF und die PostScript Programmiersprache haben das Adobe Imaging Model als Gemeinsamkeit. Es kann nahtlos zwischen PDF und PostScript konvertiert werden und beide erzielen das gleiche Ausgabeergebnis beim Druck. Dennoch fehlt PDF das general-purpose Framework der PostScript Programmiersprache. Stattdessen stellt ein PDF Dokument eine statische Datenstruktur optimiert für den random-access auf beliebigen Seiten dar und enthält zusätzlich Seitennavigationsinformationen für interaktives Lesen. Im Kontrast dazu

sind PostScript-Dateien seriell organisiert.

Das high-level Imaging Model beschreibt die Elemente, die auf der Seite dargestellt werden, also Text, Geometrie oder Vektorgrafiken, als abstrakte graphische Elemente aus Vektorobjekten und Bézierkurven, anstatt als Pixeldefinitionen. Pfad-Objekte werden durch verbundenen Punkten, Linien und Kurven mathematisch berechnet. Text-Objekte bilden eine eigene Datenstruktur (Fonts), die als Glyphen aus Pfad-Objekten bestehen. Bild-Objekte sind aus einzelnen Pixelwerten in einer rechteckigen Fläche aufgebaut und enthalten eine eindeutige Position im Rechteck und einen Farbwert. Die abstrakte Beschreibung graphischer Elemente macht das Imaging Model zu einem geräteunabhängigen Modell und es kann hochwertige Ausgaben auf vielen verschiedenen Druckern und Bildschirmen liefern.

Eine Anwendung generiert zuerst die geräteunabhängige Beschreibung des gewünschten Ausgabegeräts in der PDL. Daraufhin interpretiert eine Firmware oder Software eines spezifischen Ausgabegeräts für Rasterausgaben die Beschreibung und rendert sie im Ausgabegerät. Hierbei hat die PDL die Rolle eines Austauschstandards für die Übertragung und Speicherung von druckbaren oder auf Displays darstellbaren Dokumenten [42]. Die Flexibilität des Adobe Imaging Models zeichnet sich durch seine flexible Ausgabefähigkeit auf jeglichen Rastergeräten und hochauflösenden Displays. Die Größe des Pixels wird durch die Ausgabeauflösung des Rastergeräts bestimmt, die bei Monitoren zwischen 75 und 110 Pixels per inch (ppi) und bei Tintentstrahl- bzw. Laserdruckern zwischen 300 und 1400 ppi liegt.

Später wurde das Imaging Model für die Unterstützung von Transparenzen erweitert. Diese Funktionalität wurde speziell für PDF implementiert und wird nicht von PostScript unterstützt. Bei PostScript überschreibt das zuletzt gezeichnete Objekt alle darunterliegenden Objekte im Hintergrund [8].

1.4.3 Dateiformataufbau

PDF ist ein reines objektbasiertes Dateiformat und PDF-Dateien enthalten Dokumentdaten in binärer Form. PDF-Dateien bestehen aus Sequenzen von 8-Bit-Binärdaten bzw. 7-Bit-ASCII [8]. Ein Dokument entspricht immer einer Datei. Das Einbetten von binären Dateien in beliebigen Formaten oder anderer PDF-Dateien ist möglich.

Die Struktur besteht im Wesentlichen aus 4 Komponenten. Zunächst spezifiziert der Header die Version der PDF-Spezifikation (Signature) und den Charset Identifier [44]. Zwei Beispiele eines Headers sind in den Abbildungen 1.3 und 1.4 gezeigt. Der Body enthält die Daten der Objekte, aus denen das Dokument besteht. Objekte im Body sind in einer komplizierten hierarchischen Struktur, dem Dokument, verknüpft. Zur Dateigrößenoptimierung werden komplexe Verbindungen zwischen den Daten hergestellt und die Daten eines mehrfach vorkommenden Objektes müssen nur einmal gespeichert werden [10]. Jedes Objekt wird im Body mit obj und endobject eingekapselt und jeder Stream mit stream und endstream [8]. Eine Beispielabbildung 1.5 ist weiter unten zu sehen. Die Cross-Reference Table (Xref) deckt die Informationen über die Position der indirekten Objekte in der Datei ab. Liegt ein

Verweis auf ein Objekt in der Xref, so kann es für andere Seiten wiederverwendet werden. In Xref sind alle Informationen für den random-access eingetragen. Neben Objekteinträgen können auch Cross-Reference-Streams hinterlegt werden. Xref ist der einzige Teil in PDF mit einem konstanten Format und kann aus mehreren Cross-Reference-Sections und Subsections jeweils mit Objekteinträgen für das inkrementelle Update bestehen. Ein Objekteintrag ist wie folgt aufgebaut: Die ersten 10 Bytes für die Byteposition (Offset), mit Leerzeichen als 1 Byte getrennt, die folgenden 5 Bytes für die eindeutige Generation Number und zuletzt eine ebenfalls durch ein Leerzeichen getrennte Markierung mit f oder n. f steht für free entry, d.h. gelöschtes Objekt, und n für in use entry. Das erste Objekt in Xref hat eine Generation Number von 0 und wird nicht verwendet. Eine beispielhafte Xref ist in einer Abbildung 1.6 gezeigt. Zuletzt definiert der Trailer die Startposition der Cross-Reference Table als Pointer startxref und von speziellen Objekten im Body [45]. Außerdem enthält der Trailer die ID der PDF-Datei, einen Size Entry, Metadaten und eine Referenz zum /Root Objekt (Katalog) im Body. Abschließend markiert %%EOF das Dateiende [44], [45]. Zwei Trailerbeispiele sind in Abbildungen 1.7 und 1.8 dargestellt.

Beim inkrementellen Update wird am Ende der Original-Trailers ein Body Update, eine neue Xref-Section und ein Updated Trailer der Datei hinzugefügt. In der neuen Version der Xref-Section werden alle Objekte aufgeführt, die gelöscht, geändert oder ersetzt wurden. Der Updated Trailer umfasst alle Änderungen bezüglich des originalen Trailers [8].

PDF-Viewer prozessieren PDF-Dateien im Prinzip vom Ende bis zum Anfang, d.h. vom Trailer zum Body[45]. Beim Parsen wird zunächst die Signature überprüft, dann wird die Position von %%EOF und startxref gesucht, was die Position der Xref angibt. Die Xref stellt die Offsets jedes Objektes zur Verfügung und der Trailer zeigt auf den /Root Entry des Root-Objekts. Nachfolgend werden alle Objekte geparst und überprüft, ob /Root den /Pages Entry, /Pages ein Seiten-Array, jede /Page eine Größe der /MediaBox hat, /Contents als Stream-Objekt vorliegt und /Resources das /Font dictionary definiert. Zuletzt wird die Seite gerendert durch BeginText, Auswahl des Fonts, Bewegung des Cursors, Anzeigen des Strings und EndText [44]. Referenzen werden nicht in der parse time ausgewertet, sondern nur nach Verwendung [46].

Seitenobjekte und die meisten PDF-Strukturen sind gerichtete azyklische Graphen [46]. PDF-Objekte können in den folgenden Typen vorliegen: Booleans, Integers oder reelle Zahlen, Strings, Namen, die mit einem / Zeichen beginnen, Arrays, Dictionaries, Streams, Null Objekt oder Kommentare mit einem vorangestellten %. Strings können als <Length> <string> oder <string> <terminating symbol> definiert werden. Dictionary Objekte sind als ein Paar von Objekten, genannt Entries, implementiert. Aktionen werden beispielsweise als Entries gespeichert [47].

Text in AcroForms wird als Stream gespeichert. In Streams kann alles gespeichert werden und sie werden nicht interpretiert. Die Objekte in Streams können Referenzen auf andere Objekte vor allem Seiten enthalten [45]. Objekte können so gruppiert werden, wodurch eine

bessere Komprimierung erreicht wird, vor allem bei der Gruppierung von Linien [8].

Es gibt graphics state-Operatoren, die eine Datenstruktur, genannt graphics state, manipulieren. Graphics state ist ein globales Framework mit dem andere Grafikoperatoren ausgeführt werden. Es enthält die aktuelle Transformationsmatrix (CTM), welche die Benutzerortskoordinaten in Ausgabegerätekoordinaten mappt, die aktuelle Farbe, den aktuellen Clipping Pfad und andere Parameter, die implizite Operanden von Zeichenoperatoren sind. Außerdem gibt es noch Pfaderstellungsoperatoren, Pfadzeichenoperatoren und Textoperatoren. Zusätzlich werden marked-content Operatoren für den Dokumentenaustausch verwendet[48] .

Grundsätzlich besteht eine PDF-Datei aus 5 Seitenrahmen als Boxen für jede Seite. Diese Boxen werden weder gedruckt, noch standardmäßig angezeigt. Die äußerste und größte Box ist die MediaBox. Sie entspricht der physischen Größe des Mediums und dem Papierformat. Sie muss immer in einer PDF-Datei vorhanden sein und enthält auch alle Objekte, die über den Rand der Seitengröße hinausragen, wobei diese über die MediaBox gehen können. Innerhalb der MediaBox liegt als nächste Box die CropBox. Sie entsteht durch das Beschneiden der Seite und definiert den Ausschnitt zur Anzeige in Acrobat. Zusätzlich wird sie meist zum Platzieren von PDF-Dokumenten in anderen Programmen und zum Ausdrucken aus Acrobat verwendet. Beim Erstellen der PDF-Datei hat die CropBox die Größe der MediaBox und ist immer vorhanden. Darunter liegt die BleedBox. Sie definiert die Beschnittzugabe, die in der Praxis meist auf 3 mm gesetzt wird. Beschnittzugabe wird in der Druckvorstufe verwendet, damit keine weißen Blitzer am Rand des beschnittenen Druckbogens zu sehen sind. Die optionale BleedBox sollte kleiner als die MediaBox sein. Druckmarken wie Passkreuze, Schnittmarken oder Farbbalken sollten immer außerhalb der BleedBox liegen. Die TrimBox steht für die finale Größe des gedruckten und zugeschnittenen Dokuments. Ein zu druckendes Dokument benötigt zwingend die im PDF-Dateiformat optionale TrimBox, deren Größe kleiner oder gleich der BleedBox und MediaBox sein sollte. Der Standardwert für die TrimBox ist die Größe der CropBox. Ganz innen im Boxmodell von PDF liegt die ArtBox. Sie stellt einen Rahmen um alle druckbaren Objekte dar und legt somit den Inhalt fest. Meist sind ArtBox und TrimBox von der Größe her identisch. Im Bezug auf PDF/X-Dateien darf nur entweder die ArtBox oder die TrimBox vorhanden sein. Da die ArtBox optional ist, dient sie vor allem in Ausschussprogrammen als Default-Box, falls keine TrimBox angelegt wurde. Abbildung 1.2 zeigt eine Grafik des verschachtelten Box-Modells der PDF-Boxen.

Erstellt man eine PDF-Datei muss man zwischen composite oder separiert wählen. Bei einer separierten Ausgabe gibt es für jeden Farbauszug bei CMYK inklusive aller Sonderfarben einen eigenen Farbauszug. Inhalte einer Composite-PDF-Datei können einfach verändert werden. Farbsimulationen und die Nutzung der Überdruckvorschau sind in einem separierten PDF-Dokument nicht mehr möglich. Separierte PDF-Dateien sind nicht medienneutral, da eine Farbverrechnung bei der Separation erfolgt. Überfüllung bleibt im Composite-PDF nicht erhalten, jedoch Überdrucken und Aussparen. In modernen Workflows der Druckindustrie hat man sich vom separierten Workflow abgewendet.

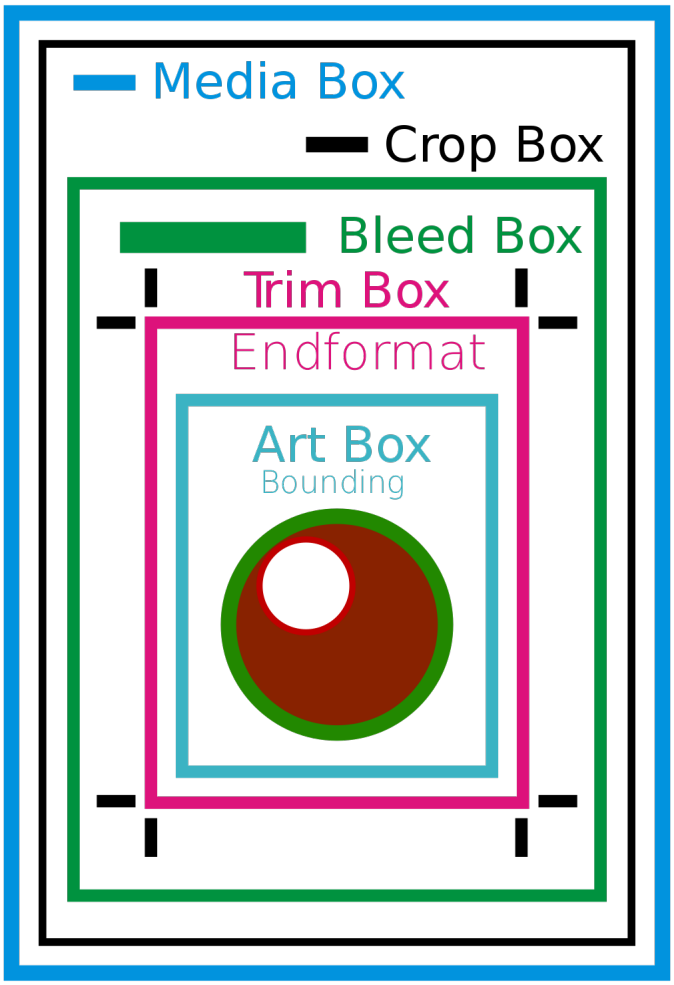

Abbildung 1.2 verschachtelte PDF-Boxen [2]

Eine PDF-Datei besitzt 3 verschiedene Layers, womit nicht die Optional Content Layers gemeint sind. Zunächst enthält die Content Layer alle druckbaren Objekte, sprich Grafiken, Bilder und Texte. Elemente dieses Layers können ausschließlich mit Acrobat Pro bearbeitet werden. Oberhalb der Content-Layer liegt die Enhancement Layer. In ihr sind Lesezeichen, Hyperlinks, Thumbnails, digitale Signaturen, Annotationen, Formularfelder und alle Multimedia-Elemente wie Video und Audio abgelegt. Die unsichtbare Information Layer umfasst alle Basisinformationen zu Schriftdaten, Formularfeldinhalten, Verschlüsselungsinformationen, Querverweistabellen und PDF-spezifische Informationen [8].

PDF-Layouts können in linearar oder nicht linearer Form aufgebaut sein. Nicht lineare PDFs sind kleiner, aber langsamer und die Seiten sind in der Datei an verschiedenen Stellen. Lineare PDFs sind für den Online-Bereich vorgesehen und werden auf die Festplatte in einer linearen Art und Weise gespeichert. Online-PDF-Viewer können somit die Datei schon anzeigen bevor sie komplett geladen wurde [48].

```
%PDF-1.7
%ÍÎÒ©
1 0 obj
<<
/Metadata 4 0 R
/PageLayout /OneColumn
/PageMode /UseNone
/Pages 2 0 R
/Type /Catalog
>>
endobj
2 0 obj
<<
/Count 6
/Kids [ 6 0 R 34 0 R 37 0 R 40 0 R 53 0 R 65 0 R ]
/Type /Pages
>>
endobj
3 0 obj
<<
/CreationDate (D:20220629125956+00'00')
/ModDate (D:20220629125956+00'00')
/Producer
(þÿ\000A\000s\000c\000e\000n\000s\000i\000o\000\040\000S\000y\000s\000t\000e\000m\000
\040\000S\000I\000A\000\040\000C\000o\000p\000y\000r\000i\000g\000h\000t\000\040\000\
050\000c\000\051\000\040\0002\0000\0001\0008)
>>
endobj
4 0 obj
<<
/Length 5 0 R
/Subtype /XML
/Type /Metadata
>>
stream
<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1-701">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="" xmlns:pdf="http://ns.adobe.com/pdf/1.3/">
<pdf:Producer>Ascensio System SIA Copyright (c) 2018</pdf:Producer>
</rdf:Description>
<rdf:Description rdf:about="" xmlns:xmp="http://ns.adobe.com/xap/1.0/">
<xmp:CreatorTool>ONLYOFFICE</xmp:CreatorTool>
<xmp:CreateDate>2022-06-29T12:59:56+00:00</xmp:CreateDate><xmp:ModifyDate>2022-06-29T
12:59:56+00:00</xmp:ModifyDate></rdf:Description>
</rdf:RDF>
</x:xmpmeta><?xpacket end="w"?>
endstream
endobj
5 0 obj
629
endobj
6 0 obj
<<
/Contents 7 0 R
/MediaBox [ 0 0 595.29999 841.90002 ]
/Parent 2 0 R
/Resources <<
/ExtGState <<
/E98 27 0 R
>>
/Font <<
/F1 9 0 R
/F2 18 0 R
>>
/Pattern <<
/P1 32 0 R
>>
/ProcSet [ /PDF /Text /ImageB /ImageC /ImageI ]
>>
/Type /Page
>>
endobj
7 0 obj
<<
/Filter [ /FlateDecode ]
/Length 8 0 R
>>
stream
xœ½ZY‹$É
~¯_Q DEL sBRœ0 RS TUW ESC oÉG¿ EM ?‡×fÙYXÏ,ý¾ NAK Š# NAK '™ÝÕ,x†j°â'Bútfpz°¾ž>½À EM àúúãÉ'[ STX
Ú^çø[@Lç× DEL æ> ESC càÊùõ'^ŠgÀ¼ôÉ†¸8ŒÎO DC4 ø›  yé_e- DEL ?ôåÉ,  ) BS yÕòÇñÇ¯3ÖBž
òÉûDŸë|äO⁴{ÆµETB p\ùsãÏs^ùå  Ñ/>:cóü½ž€eÚ¡Ã<ü"‡8C©'H•Œ+üBþó\6Ý¾½°ÒØMuœ©õõÝ€¶Ë¡Ý DLE
```

Abbildung 1.3 Header und Beginn eines Bodys des PDF-Dateiformats in Notepad++

Abbildung 1.4 Header, Beginn eines Bodys mit Katalog und Binärdaten des PDF-Dateiformats in Notepad++

Abbildung 1.5 PDF Bodyauszug des PDF-Dateiformats in Notepad++

1.4.4 Implementierung von Fonts

Schriften werden als Konturen (Outlines) digital gespeichert und mit Instruktionen (Hints) versehen, sodass sie generisch skaliert werden können. Fonts werden in PDF als eigenständige Dateien behandelt und die Darstellung wird durch einen speziellen Font-Interpreter bewerkstelligt. Innerhalb der Datei werden Fonts als Dictionary registriert, worin u.a. Fonttyp als Subtype, PostScript-Name, Verschlüsselung und Schriftklasse (für eine mögliche Ersatzschrift) abgelegt sind. Für jeden verwendeten Font wird ein Font-Descriptor in der Datei hinterlegt. PDF verarbeitet 2 Klassen von Fonttypen: Simple-Fonts und Composite-Fonts.

Jede Glyphe im Dokument wird über einen Character-Code prozessiert. Daraufhin erfolgt eine Zuordnung des Character Codes zum hinterlegten Encoding (Mapping). Zuletzt wird die Glyphe im aktuellen Font über die Glyphen-ID zum Zeichen der Glyphe aufgerufen. Folglich erzielt das Mapping des Codes und der Aufruf der Glyphe die benötigte Konturbe-

```
3127    xref
3128    0 454
3129    0000000035 65535 f
3130    0000000017 00000 n
3131    0000000125 00000 n
3132    0000000188 00000 n
3133    0000000482 00000 n
3134    0000008922 00000 n
3135    0000009098 00000 n
3136    0000009356 00000 n
3137    0000009537 00000 n
3138    0000009799 00000 n
3139    0000009959 00000 n
3140    0000010184 00000 n
3141    0000010360 00000 n
3142    0000010605 00000 n
3143    0000010730 00000 n
3144    0000010760 00000 n
3145    0000010913 00000 n
3146    0000010987 00000 n
3147    0000011218 00000 n
3148    0000011358 00000 n
3149    0000011388 00000 n
3150    0000011556 00000 n
```

Abbildung 1.6 PDF Xref des PDF-Dateiformats in Notepad++

```
10763   trailer
10764   << /Size 467 /Root 438 0 R /Info 1 0 R /ID [ <034581ec832fbe851b69c8ecbfbb94a0>
10765   <034581ec832fbe851b69c8ecbfbb94a0> ] >>
10766   startxref
10767   1158066
10768   %%EOF
```

Abbildung 1.7 Kurzer Trailer des PDF-Dateiformats in Notepad++

```
3581    0000375647 00000 n
3582    0000398023 00000 n
3583    trailer
3584    <</Size 454/Root 1 0 R/Info 34 0
        R/ID[<C5AF47CFA4702847AAF5ABBD07AB5874><C5AF47CFA4702847AAF5ABBD07AB5874>] >>
3585    startxref
3586    399162
3587    %%EOF
3588    xref
3589    0 0
3590    trailer
3591    <</Size 454/Root 1 0 R/Info 34 0
        R/ID[<C5AF47CFA4702847AAF5ABBD07AB5874><C5AF47CFA4702847AAF5ABBD07AB5874>] /Prev
        399162/XRefStm 398023>>
3592    startxref
3593    408402
3594    %%EOF
```

Abbildung 1.8 Ende einer Xref und Trailer des PDF-Dateiformats in Notepad++

schreibung. Ein optionales Unicode-Mapping ToUnicode ist von Nöten, damit die Glyphen auch über Unicode verarbeitet werden kann. Ist dieses Mapping nicht vorhanden, so kann

keine Textsuche und das Kopieren von Text stattfinden. Fehler im Mapping oder Modifikation von Schriften können zu falsche Ausgabebuchstaben, mangelnde Wiederverwendung und fehlerhafte Textkonvertierung führen.

Eine Font-Einbettung kann als zweite Option als Object-Stream gespeichert werden bzw. können Fonts sich auf eine externe Referenz beziehen. Die Beschreibung von Glypen ist bei eingebettete Schriften als Steam im Eintrag FontFile registriert. Falls die Schrift nicht eingebettet wurde fehlt dieser Eintrag. Schriftsubstitution findet immer dann statt, wenn der Character-Code nicht mit der Encoding-Tabelle übereinstimmt. Häufig fehlen bestimmte Glypen im Font. Falls eine Outline-Beschreibung des Fonts zum Erstellungszeitpunkt nicht verfügbar ist, wird die Einbettung des Fonts verhindert. Dies kommt vor allem dann vor, wenn ein Font ein Schutzflag besitzt. Weitere Probleme bei der Schrifteinbettung sind u.a. Laufweitenfehler in Schriften, Fehler in der Buchstabenbeschreibung oder beim Cachen von Fonts. Zwecks der Schriftsubstitution müssen folgende allgemeine Informationen zu einem Font in der PDF-Datei gespeichert sein: Name der Schrift, Fontfamilie, Typ, Subtyp, Schriftstärke (Font Weight), Zeichenbreite, Laufweite, maximale Ausprägung der FontBox, Dickteninformationen, Positionsangaben über Versal- und x-Höhe und Winkel für Italic (kursiv). Diese Informationen sind selbst bei nicht eingebetteten Schriften vorhanden [8].

1.5 PDF Sicherheitsaspekte

Etwa 40 % der Unternehmen setzen PDFs für geschützte Inhalte ein. In den letzten 2 Jahren ist die Nutzung der elektronischen Signaturfunktion in PDFs um mehr als 150 % gestiegen [49]. Diese Statistik zeigt keine gute Entwicklung, da PDF-Dokumente nicht für vertrauliche Inhalte verwendet werden sollten. Obwohl PDF Verschlüsselung implementiert, erlaubt das Dateiformat eine Vielzahl von Angriffen, um die Sicherheit von PDF-Dokumenten zu untergraben. Ich werde im Folgenden die Sicherheitsmechanismen von PDF, sowie PDF Signature Spoofing und PDFex als Risikostufen für Schwachstellen näher beleuchten.

Neben den im weiteren Verlauf detailliert beschriebenen Attacken aus dem Jahr 2019, gibt es noch weitere Attacken auf PDF-Dokumente, die ich aber nur kurz benennen und nicht weiter ausführen werde. 2020 wurden Shadow Attacks entdeckt, die sich auf ein PDF Dokument mit 2 Inhaltsebenen beziehen. Zum einen wird Inhalt sichtbar, der beim Ersteller der Signatur angezeigt wird und zum anderen wird ein alternativer versteckter Inhalt dem Opfer gezeigt, nachdem das Dokument signiert wurde. Weitere unsichere Features würden im Jahr 2021 in PDFs untersucht. Man fand Möglichkeiten für Denial-of-Service (DOS) attacks, die sich auf den Host beziehen, der das Dokument bearbeitet, Information disclosure attacks, die persönliche Daten aus dem Computer des Opfers freilegen können, Data manipulation auf dem System des Opfers und Code execution auf dem Computer des Opfers. Zuletzt im Jahr 2021 wurden Sicherheitslücken in der PDF-Spezifikation entdeckt, die Evil Annotation Attack (EEA) und Sneaky Signature Attack (SSA) realistisch machen. Nutzt der Angreifer diese Sicherheitslücken aus, so kann er den sichtbaren Inhalt eines Dokuments verändern, indem er bösartigen Inhalt über dem zertifizierten Inhalt anzeigt.

Nichtsdestotrotz bleibt das Zertifikat gültig und die PDF-Anwendung zeigt keine Warnung an [50].

In den Sicherheitseinstellungen eines PDF-Dokuments können Dokumentensicherheit und Zugriffsregeln justiert werden. PDF unterstützt Verschlüsselung und die Vergabe von Passwörtern. Eventuell kann beim Öffnen einer Datei ein Passwort gefordert werden oder das Kopieren von Teilinhalten, jeglichem Inhalt, Ausfüllen von Formularfeldern, Dokumentveränderungen (z.B. Struktur, Inhalt, Kommentare), das Einbetten und Editieren von Schriften oder das Ausdrucken kann vom Ersteller des Dokuments gesperrt worden sein. Das Schwärzen von Dokumenten, d.h. ein schwarzer Balken liegt auf dem Text, führt nicht dazu, dass der geschwärzte Text aus der Datei verschwindet. Er kann weiterhin ausgelesen werden, ebenso Metadaten [13].

1.5.1 Digitale Signatur

Digitale Signaturen sollen die Identität des Unterzeichners des Dokuments authentifizieren und sicherstellen, dass der Inhalt nach der digitalen Signatur nicht geändert wurde. Der Verfasser benötigt für die digitale Signatur ein Signatur-Zertifikat. Das Zertifikat bescheinigt die Echtheit der Signatur bzw. Herkunft und wird von einem möglichst vertrauenswürdigen Zertifizierungsanbieter ausgestellt. Zusätzlich können Zertifikate ablaufen oder entzogen werden und müssen gültig sein. Jede digitale Signatur kann mit einem Zeitstempel versehen werden. Ein externer Zeitstempel-Anbieter (ZSA) belegt den Zeitpunkt, wann die digitale Signatur geleistet wurde [10].

Eine digitale Signatur ist eine spezielle Art von elektronischer Signatur, die kryptographische Techniken implementiert. Im Gegensatz dazu kann eine elektronische Signatur verschiedene Formen annehmen, beispielsweise eine gescannte handschriftliche Unterschrift, ein getippter Name, eine biometrische Signatur oder eine digitale Signatur. Elektronische Signaturen sind der Oberbegriff und digitale Signaturen sind eine Unterart davon. Elektronische Signaturen bieten variierende Sicherheitsgrade [13].

Digitale Signaturen verwenden asymmetrische Kryptographie. Anfangs wird ein Algorithmus zur Schlüsselgenerierung initiiert. Als Ausgabe werden ein Private Key und Public Key erzeugt. Ein Signierungsalgorithmus erstellt den einzigartigen Hash-Wert des PDF-Dokuments und verschlüsselt ihn mit dem geheimen Private Key, was die digitale Signatur als Output produziert. Zuletzt überprüft ein Signaturvalidierungsalgorithmus die Authentizität des PDF-Dokuments, indem er erneut einen Hash des Dokuments berechnet. Die Signatur wird mit dem Public Key entschlüsselt und gibt ebenfalls einen Hash aus. Falls der berechnete Hash im Signaturvalidierungsalgorithmus mit dem entschlüsselten Hash übereinstimmt, ist die Authentizität des PDF-Dokuments gesichert [51].

Eine PDF-Datei ermöglicht mehrere digitale Signaturen, jedoch muss jede neue Signaturen in einem inkrementellen Update geleistet werden. Alle digitalen Signaturen muss mit einem Unterschriftsfeld im Dokument verbunden sein. Optional kann das Unterschriftsfeld mit einem Widget gekoppelt sein. Dann wird die Signatur graphisch dargestellte. Signaturen

ohne Widgets sind versteckte digitale Signaturen [10]. Auf dem Markt werden Desktop-Anwendungen und Online-Validierungsservices zur Überprüfung der digitalen Signatur angeboten.

1.5.2 PDF Signature Spoofing

Die PDF-Spezifikation ist sehr ungenau im Bezug auf digitale Signaturen bzw. auf welche Art und Weise sie validiert werden müssen. PDF-Viewer haben eine hohe Toleranz beim Öffnen, Validieren und Anzeigen von beschädigten PDF-Dateien.

Im Jahr 2019 wurden 3 Attacken zum Vortäuschen von validen PDF Signaturen erforscht: Incremental Saving Attack (ISA), Signature Wrapping Attack (SWA) und Universal Signature Forgery (USF). Das attack scenario sieht wie folgt aus: Der Angreifer besitzt das signierte PDF mit einer gültigen digitalen Signatur und manipuliert es. Das manipulierte signierte PDF wird an das Opfer geschickt und die Signatur bleibt gültig, obwohl der Inhalt geändert wurde.

Die ISA nutzt das incremental update Feature von PDF aus, um bösartigen Inhalt in das PDF des Opfers zu schleusen. Im Prozess des Signierens wird ein incremental update verwendet, um die Signatur zu speichern. Am Ende des Originaltrailers wird ein neuer Katalog und ein neues Signatur-Objekt als Body Updates angehängt, was den Signaturwert und Informationen über den Ersteller der Signatur enthält. Danach kommt eine updated Xref section und ein updated Trailer. Beim inremental update können Objekte mit anderem Inhalt neu definiert werden. Bei einem Spezifikation-konformem incremental update wird der updated Body hinter dem Originaltrailer am Dateiende angehängt. Darunter zeigt die updated Xref section auf das neue Objekt und der updated Trailer wird ganz am Ende angefügt. Abbildung 1.9 zeigt den Vorgang des Speicherns einer digitalen Signatur mittels incremental update.

Zuerst hat man bei der ISA geprüft, ob die PDF-Reader gegen eine Attacke anfällig waren, bei der hinter dem incremental update, was die digitale Signatur enthält, eine weitere Sektion mit Body Updates, Xref und Trailer angefügt wurde. Ausschließlich LibreOffice wurde mittels diese Vorgehensweise getäuscht. Alle weiteren Strategien produzieren beschädigte nicht Standard-konforme PDF-Dateien, jedoch PDF-Viewer sind sehr fehlertolerant. Als Nächstes haben die Forscher hinter dem updated Trailer des incremental updates der Signatur Body Updates hinzugefügt. Einige getestete PDF-Viewer haben lediglich überprüft, ob eine neue Xref und Trailer vorhanden sind. Da sie fehlten, blieb die Signatur gültig, die zusätzlichen Body Updates wurden ausgeführt und die Modifikation blieb ohne Warnung unbemerkt. Andere PDF-Viewer benötigten neben zusätzlichen Body Updates noch einen Trailer ohne Xref dazwischen, damit keine Warnung geworfen wurde. Die komplexeste Vorgehensweise, bei der neue Body Updates mit einer Kopie des Signatur-Objekts am Dateiende angebracht wurde, hat einige PDF-Viewer wie Foxit gezwungen die Signatur 2 Mal zu validieren und die Body Updates wurden eingefügt. In der Grafik 1.10 werden die Techniken visualisiert,

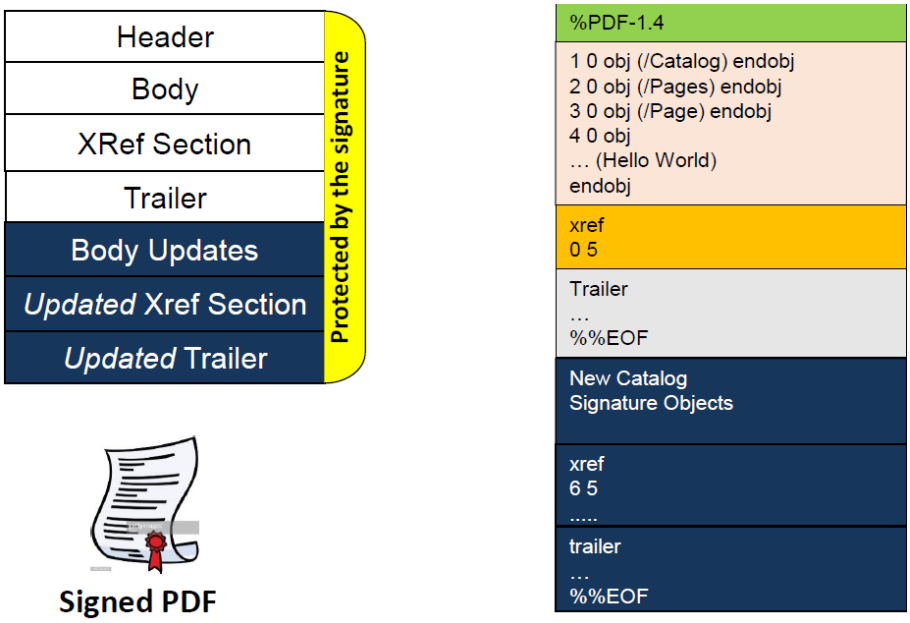

Abbildung 1.9 incremental update mit digitaler Signatur [52]

sowie die PDF-Viewer dargestellt, die anfällig für die ISA-Vorgehensweise waren.

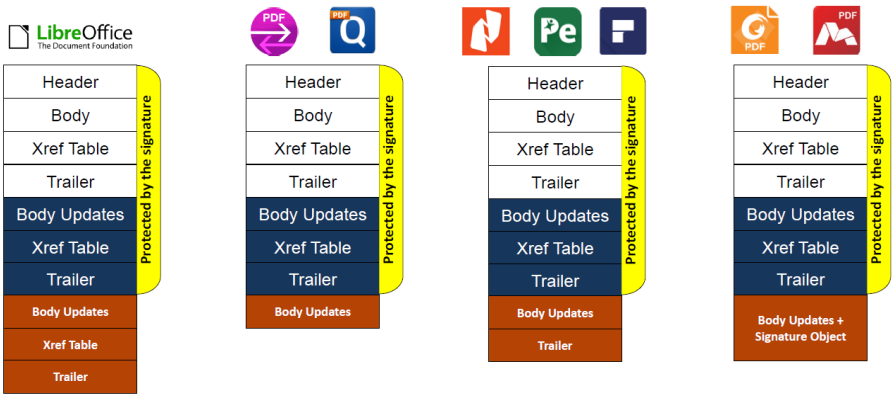

Abbildung 1.10 ISA-Methoden mit anfälligen PDF-Viewern [52]

Bei der SWA werden die Werte der signierten /ByteRange modifiziert und Platz wird für das Einschleusen von bösartigen Inhalten geschaffen. Das Signatur-Objekt besteht aus einem /Contents entry mit dem Signaturwert und einem /ByteRange entry mit 4 Werten, was den signierten Teil im Dokument anzeigt. Die ersten 2 Einträge der /ByteRange beziehen sich auf den Beginn des Dokuments bis zum Anfang des Signaturwerts. Hingegen definieren die letzten beiden ByteRange-Einträge den Bereich nach dem Signaturwert bis zu %%EOF. Diese beiden Bereiche wurden nicht angerührt. Die erfolgreiche Idee war eine zweite /ByteRange hinter dem Signaturwert mit einem angepassten dritten Wert, der Platz für bösartige Objekte, etwas Padding und eine bösartige Xref, die auf die Objekte des Angreifers zeigt, zu setzen. Lediglich die Position der neuen Xref ist im Trailer vorgegeben, der nicht verändert werden kann. Das Schaubild 1.11 zeigt, wie die SWA arbeitet.

Mittels USF wird die Signaturvalidierung außer Kraft gesetzt. Dennoch wird die Meldung „PDF is validly signed"dem Benutzer angezeigt. Vor allem die Adobe-Produkte waren anfällig

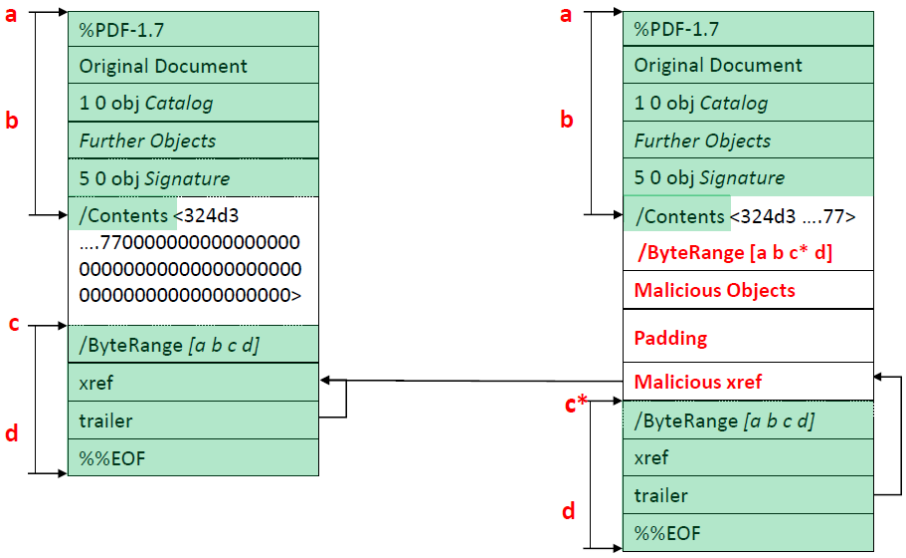

Abbildung 1.11 SWA-Methodik [52]

für diese Attacke. Die Vorgehensweise gestaltete sich wie folgt: Die Forscher versuchten /Contents oder /ByteRange der Signatur entweder wegzulassen oder unüblichen Werten wie kein Wert oder null zuzuweisen. Zwei PDF-Viewer versagten, wenn man /Contents auf 0 Bytes setzte. Das Weglassen der /ByteRange oder das Gleichsetzen mit null brach die Sicherheitsmechanismen von Adobe. Die Grafik 1.12 zeigt die USF-Varianten, die von den Forschern getestet wurde.

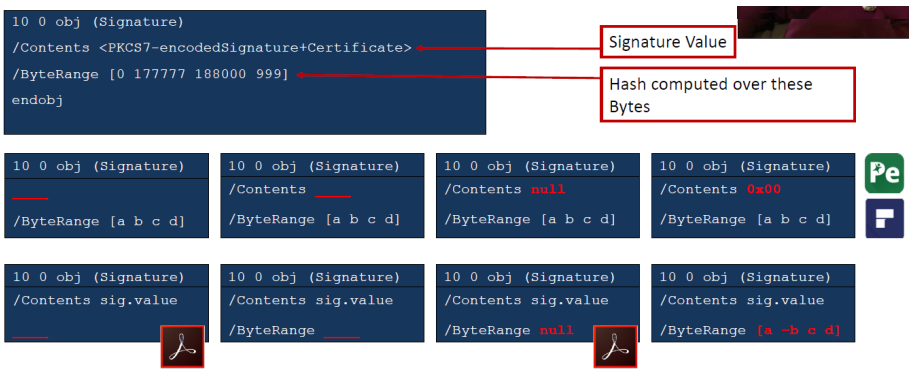

Abbildung 1.12 USF-Varianten [52]

Mittels dieser Attacken ist es den Forschern gelungen zu zeigen, dass 21 von 22 PDF-Viewer inklusive denen von Adobe und 6 von 8 Online-Anbieter anfällig waren. Der einzige PDF-Viewer, der gegen alle Attacken immun war, war die veraltete Version von Adobe Reader 9, die jedoch remote code execution enthält [45].

1.5.3 PDFex PDF Encryption Attacks

PDFex bezeichnet Attacken auf die PDF-Verschlüsselung mit Passwortschutz, die im Jahr 2019 entdeckt wurden. Die Forscher haben 27 PDF-Viewer auf diese Attacken getestet und alle Viewer inklusive Adobe Reader, Chrome and Firefox waren mindestens für eine Attacke

anfällig. Es gibt 2 Attack-Klassifizierungen: Direct Exfiltration und Cipher Block Chaining (CBC) malleability gadgets oder crypto gadgets genannt. Das attacker model gestaltet sich wie folgt: 2 Personen Alice und Bob wollen ein vertrauliches PDF mit Passwortschutz austauschen. Wir nehmen an, dass ein Angreifer durch eine Man-in-the-middle (MITM) attack das verschlüsselte PDF unbemerkt abgreifen konnte als Alice das Dokument zu Bob geschickt hat. Daraufhin modifiziert der Angreifer das Dokument, sodass die Veränderungen unerkennbar sind für Bob, und schickt seine Version des Dokuments an Bob. Bob entsperrt das Dokument mit seinem Passwort und das entschlüsselte PDF wird automatisch zurück zum Server des Angreifers übertragen [45], [53].

Ein verschlüsseltes PDF enthält einen Objekt-Stream als verschlüsselten ciphertext im Body und einen /Encrypt entry, der Informationen darüber enthält, wie die Daten entschlüsselt werden sollen. Der Großteil der PDF-Dateistruktur liegt unverschlüsselt vor. Ausschließlich alle Strings und Streams für Inhalte ohne die anderen Objekttypen wie Integer oder Booleans, die dokumentstrukturbeschreibend sind, werden verschlüsselt. Der Grund dafür ist, dass die PDF-Spezifikation weiterhin die random-access Optimierung im Dateiformat beibehalten möchte. Folglich ist die gesamte Dokumentenstruktur unverschlüsselt und eine ganze Reihe an wichtigen Informationen wie Anzahl und Größe der Seiten, verwendete Objekte oder Hyperlinks können vom Angreifer ausgelesen werden.

PDF verwendet einen speziellen Streamfilter-Typ, genannt crypt filter für die Verschlüsselung. Jeder Standard-Filter eines Streams kann vom crypt filter überschrieben werden. Ein standard identity filter wird bei besonderen Streams für z.B. Dokumentmetadaten gesetzt, damit diese in einem verschlüsselten Dokument unverschlüsselt bleiben. Diese Unterstützung von partieller Verschlüsselung kann einem Angreifer ermöglichen seinen Inhalt mit verschlüsseltem zu vermischen ohne das Passwort zu wissen. Forscher fanden 18 verschiedene Angriffsvarianten für eine ganze Reihe von PDF-Readern [45].

Bei der direct exfiltration attack wird die Kryptografie nicht angerührt. Bei allen Strings im verschlüsselten PDF von Alice überschreibt der Angreifer die crypt filter mit identity filter. Jetzt kann der Angreifer bei Bedarf Strings verändern und kann weitere Objekte mit unverschlüsselten Strings hinzufügen. Außerdem kann er verschlüsselte Teile des Dokuments in Inhalte einfügen, die er kontrolliert, d.h. er kann auf verschlüsselte Streams oder Strings in seinem Inhalt referenzieren und zugreifen. Nachfolgend fügt er eine submit-form action hinzu, die Namen und Werte von AcroForms zu einer Ziel-URL überträgt. Der Text des Formularfelds wird als Stream gespeichert. Kombiniert der Angreifer eine OpenAction, die ausgeführt wird, sobald das Dokument geöffnet wird, mit einer submit-form action zu attacker.com, so kann der vertrauliche entschlüsselte Inhalt an den Angreifer zurück als POST request gesendet werden. Die OpenAction wird ausgelöst, sobald Bob sein Passwort eingegeben hat. Die direct exfiltration attack mittels submit-form action ist im PDF-Dateiformatausschnitt 1.13 zu sehen. Statt einer sumbit-form action kann man auch einen Hyperlink oder JavaScript verwenden. Jedes Objekt kann als Hyperlink definiert werden, was einen GET request zu attacker.com triggert [45], [53].

PDF verschlüsselt durch AES-256 mit einer Schlüssellänge von 256 Bit und operiert im

```
1 0 obj
  << /Type /Catalog
     /AcroForm << /Fields [<< /T (x) /V 2 0 R >>] >>       % value set to 2 0 obj
     /OpenAction << /S /SubmitForm /F (http://p.df) >>          % attacker's URI
  >>
endobj

2 0 obj
  << /Filter [/Crypt] /DecodeParms [<< /Name /StdCF >>]   % encryption with StdCF
     /Length 32
  >>
stream
[encrypted data]                                          % content to exfiltrate
endstream
endobj
```

Abbildung 1.13 Direct exfiltration attack mittels subit-form OpenAction [45]

CBC mode ohne Message Authentication Code (MAC) Integritätsschutz. AES gehört zu den Blockchiffren und ist ein symmetrisches Verschlüsselungsverfahren. Bei symmetrischer Verschlüsselung gibt es nur einen Schlüssel fürs Verschlüsseln und Entschlüsseln. Ein Block ist 128 Bit lang. Der subkey (Rundenschlüssel) hat immer eine Länge von 128 Bit. Die Schlüssellänge beeinflusst den key schedule, d.h. die Art und Weise wie Rundenschlüssel vom master key abgeleitet werden, sowie die Anzahl an Runden. Während der Algorithmus ausgeführt wird, wird ein 4 x 4 Array aus je 1 Byte (insgesamt 128 Bit), genannt state, in 14 Runden verschlüsselt. Im ersten einzigen Schritt, der KeyExpansion, werden die Rundenschlüssel für jede Runde plus 1 weiterer Rundenschlüssel aus dem master key berechnet. Anfangs wird der state gleichgesetzt mit dem der zu unverschlüsselten Nachricht (plaintext). Auf dem state Array wird bitweises XOR mit dem Rundenschlüssel im AddRoundKey-Schritt angewendet. Jedes Byte vom state wird dabei mit einem Byte vom Rundenschlüssel kombiniert. Die Grafik 1.14 zeigt den AddRoundKey-Schritt. Im SubBytes (substitution) Schritt wird jedes Byte des aktualisierten state Arrays mit einem Byte durch eine Substitutionstabelle (S-Box) ersetzt. Dadurch werden die Bytes im state vermischt. Eine weitere Grafik 1.15 stellt den SubBytes-Schritt dar. Danach kommt die SchiftRows-Sequenz. Die Bytes in jeder Zeile von state werden unterschiedlich nach links verschoben: Die 1. Zeile bleibt unberührt, die 2. Zeile wird um 1 verschoben, die 3. um 2 und die letzte um 3 Positionen. Die überlaufenden Zellen werden rechts an die jeweilige Zeile wieder angehängt. Im Schaubild 1.16 ist die Funktionsweise von ShiftRows gezeigt. Als Vorletztes jeder Runde außer der letzten Runde kommt MixColumns dran. Hier werden Spalten vermischt, indem jede Spalte mit einer konstanten Matrix zu einer neuen Spalte multipliziert wird. Anschaulich wird MixColumns im Schaubild 1.17 graphisch umgesetzt. Abschließend zu jeder Runde wird nochmals AddRoundKey mit dem aktuellen Rundenschlüssel ausgeführt, also state bitweise XOR verknüpft mit dem Rundenschlüssel. Im letzten Schritt produziert AddRoundKey den ciphertext. Die invertierte Operation des Algorithmus wird beim entschlüsseln angewendet und alle Funktionen werden umgekehrt aufgerufen [54]–[56].

Während des CBC mode wird ein vorheriger ciphertext block mit dem nächsten plaintext

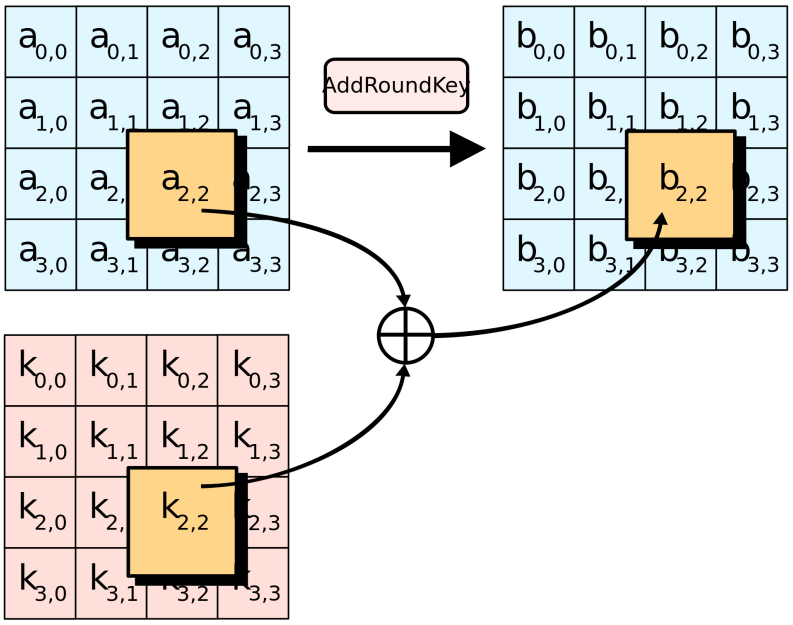

Abbildung 1.14 AddRoundKey-Schritt im AES-Algorithmus [57]

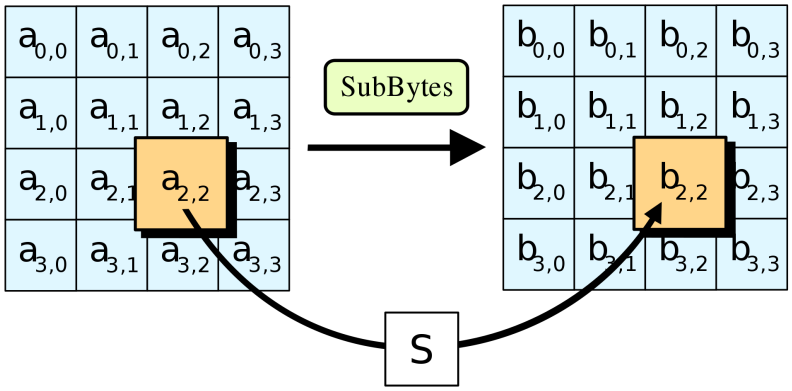

Abbildung 1.15 SubBytes-Schritt im AES-Algorithmus [57]

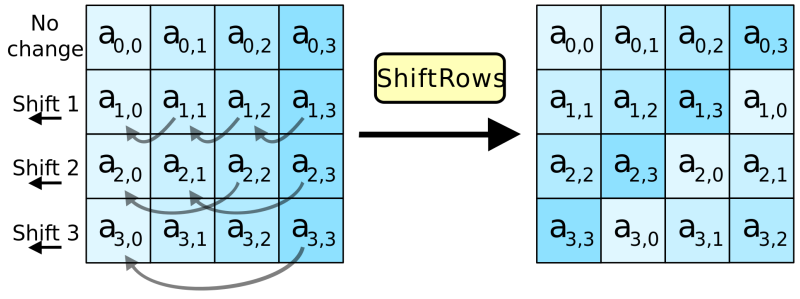

Abbildung 1.16 ShiftRows-Schritt im AES-Algorithmus [57]

block durch XOR verknüpft. Das Ergebnis und der Schlüssel dienen als Input für den AES-Algorithmus und produzieren den verschlüsselten Output als ciphertext block. Außerdem wird der Output des vorherigen Blocks (ciphertext) als Input für die XOR-Verknüpfung des nächsten Blocks (plaintext) wiederverwendet. Im ersten Schritt wird der erste plaintext block mit einem Initialisierungsvektor IV (nonce) als Zufallszahl oder Zeitstempel mit XOR verknüpft. IV sollte nur einmal verwendet werden und wird dem Empfänger vom Sender

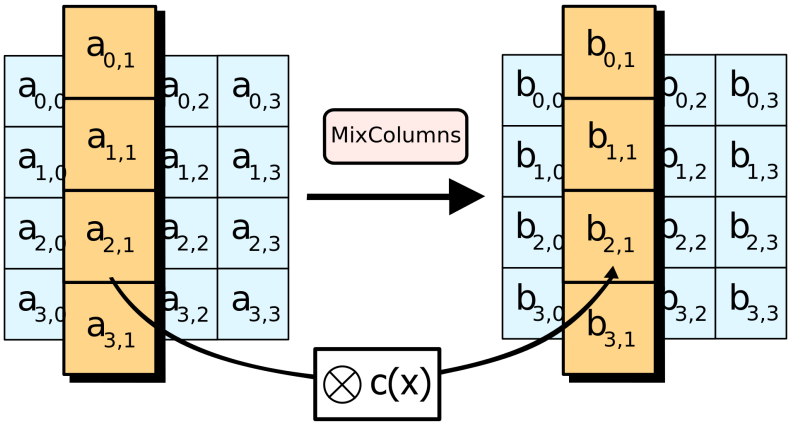

Abbildung 1.17 MixColumns-Schritt im AES-Algorithmus [57]

mit dem ciphertext übertragen, jedoch muss IV nicht geheim bleiben. Bei mindestens einer Bitmanipulation oder einem einzigen Bitfehler im ciphertext block werden die Entschlüsselung diese Blocks und alle nachfolgenden Blöcke beeinflusst. Grafik 1.18 zeigt das Prinzip von Verschlüsselung und Entschlüsselung im CBC mode.

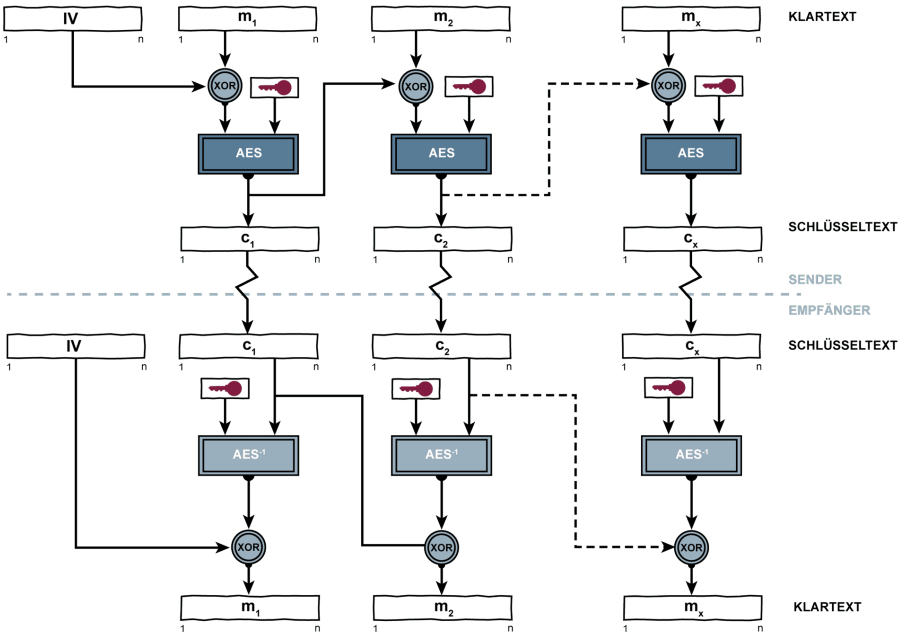

Abbildung 1.18 CBC mode Verschlüsselung und Entschlüsselung [58]

Ein MAC ist eine kryptographische Prüfsumme, die aus dem Datensatz und dem symmetrischen Schlüssel von Sender und Empfänger gebildet wird. Im Gegensatz zu einem Hash kann der MAC nur von Sender und Empfänger berechnet und ebenfalls von beiden verifiziert werden [59].

Nicht alle PDF-Viewer unterstützen teilweise verschlüsselte Dokumente. Solche Viewer sind immun gegen direct exfiltration attacks. Trotzdem können Angreifer CBC malleability gadgets verwenden, um die Exfiltration des Texts zu erhalten, da keine MAC für den

ciphertext beim PDF-Standard verwendet wird. Für die Ausführung der Attacke müssen 2 Vorbedingungen erfüllt sein: Zum einen muss ein plaintext-Segment dem Angreifer vorliegen. Beim aktuellsten AESV3-Algorithmus kann der plaintext als 16 Bytes aus dem verschlüsselten /Perms entry (extended permissions) einer PDF-Datei abgeleitet werden. Die zweiten 4 Bytes sind die permissions /P, die unverschlüsselt im PDF-Dokument herausgelesen werden können. Das Byte aus T oder F wird entsprechend des EncryptMetadate Booleans gesetzt. Abschließend kommen 3 Konstante characters ädbünd 4 random Bytes. Die Tabelle 1.19 bietet eine Übersicht des /Perms entries. Der /Perms entry wird mit dem gleichen Schlüssel für Streams und Strings verschlüsselt, daher kann man ihn verwendet um CBC gadgets zu konstruieren. Obwohl der /Perms-Wert mit Electronic Code Book (ECB) mode verschlüsselt wird, ist der output ciphertext gleich seinem plaintext, wenn man ihn mit CBC verschlüsselt und dabei einen initialization vector von 0 verwendet würde. Bei älteren Versionen des AES-Algorithmus in PDF muss der bekannte plaintext aus dem Objekt, auf das die Exfiltration angewendet werden soll, extrahiert werden [45], [53].

1...1	P Value	“T”/“F”	“adb”	random
4 byte	4 byte	1 byte	3 byte	4 byte

Abbildung 1.19 Bytebelegung beim /Perms (extended persmissions) entry in einer verschlüsselten PDF-Datei [45]

Alice verschlüsselt eine Nachricht aus 2 Blöcken p0 und p1 zum initialization vector IV, c0 und c1 und Bob kann diese Nachricht wieder zu p0 und p1 entschlüsseln. CBC gadgets kann man konstruieren mit einem bekannten plaintext (known plaintext) von 16 Bytes p0 und kann dadurch vom Angreifer ausgewählte plaintext blocks (chosen plaintext) g0 und g1 in eine entschlüsselte plaintext message einfügen. Beispielsweise könnte g1 die URL vom Angreifer enthalten, sodass er sich die entschlüsselte Nachricht p1 von Bob zu seinem Server schicken lassen kann. Der Angreifer erhält die verschlüsselte Nachricht aus IV, c0 und c1 von Alice durch eine MITM attack und konstruiert 2 neue initialization vectors x0 und x1. Nimmt man IV XOR p0 erhält man ausschließlich Nullen. X0 als vom Angreifer erstellter IV wird durch IV XOR p0 XOR g0 und x1 als zweiter vom Angreifer erstellter IV wird durch IV XOR p0 XOR g1 gebildet. Eine neue verschlüsselte Nachricht wird durch die Konkatenation von x0, c0, x1, c0 und c1 gebildet. Der ursprüngliche c0 Block wird 2 Mal verwendet. X0 und x1 sind die neuen IV und x1 ist lediglich an dieser Stelle zwischen c0 in der Verkettung, um sie beizubehalten. X1 wird zu unkontrollierbaren Bytes ohne Sinn entschlüsselt. C1 ist der ciphertext block von Alice, der die Zieldaten, enthält. Bei der Entschlüsselung erhält Bob den chosen plaintext block g0, die unkontrollierten Bytes, den zweiten chosen plaintext block g1 und die entschlüsselten Zieldaten von Alice p1. Abbild 1.20 zeigt das Wirkungsprinzip von CBC gadgets. Als Exfiltration Channel kommen AcroForms oder Hyperlinks in Frage, um die entschlüsselte Nachricht von Bob zurück an den Angreifer zu schicken [45], [59], [60].

Hauptursache des Problems ist, dass viele Datenformate eine Teilverschlüsselung erlauben, z.B. XML, S/MIME und PDF, was dem Angreifer ermöglicht eigene Inhalte einzuschleusen und exfiltration channels bilden zu können. Des Weiteren enthält AES-CBC keinen Integri-

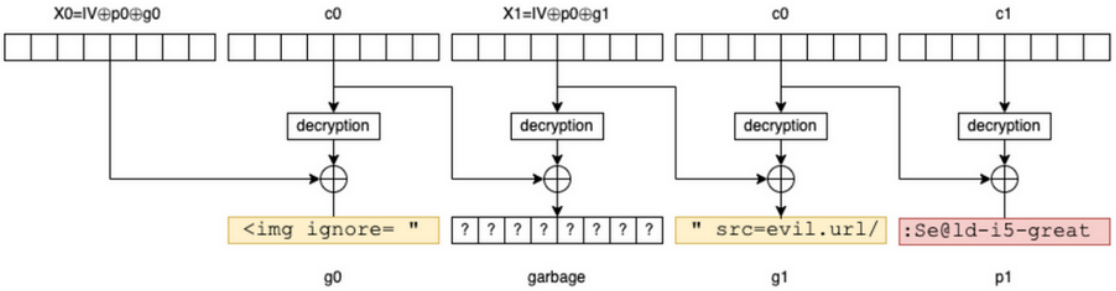

Abbildung 1.20 CBC malleability gadget mit konstruierten initialization vectors x0 und x1, um chosen plaintext einzuschleusen [60]

tätsschutz wie einen MAC, sogar nicht in PDF 2.0. Diese Sicherheitslücken sollten ohne Rückwärtskompatibilität in zukünftigen PDF-Spezifikationen entfernt werden [53].

2 Open Source PDF Web App

2.1 Problemstellung und Anforderungen

Das Thema der Bachelorarbeit *„Entwicklung einer webbasierten Applikation zur Bearbeitung von PDF Dateien“* habe ich selbst gewählt. Ziel der Arbeit ist eine PDF-Webapplikation zu entwickeln, die gängige Funktionalitäten, die man bei üblicher PDF-Bearbeitung benötigt, bieten soll. Gängige Funktionalitäten sind für mich ein PDF-Reader, das Editieren von Seiten und die grundsätzlichsten Grafikoperationen. Mir war außerdem wichtig, dass die PDF Web App auf allen Desktop-Plattformen Windows, macOS und Linux einsetzbar ist, da PDF genauso ein plattformunabhängiges und hardwareunabhängiges Format ist. Ich habe mir selbst die Vorgabe gestellt, dass die PDF Web App lediglich als Desktop-App funktionieren muss und nicht optimiert für Tablet oder Smartphone sein soll. PDF-Bearbeitung macht für mich am meisten Sinn in einer Desktopumgebung, da der Bildschirm auf Tablets und vor allem bei Smartphones nicht besonders groß bis sehr klein ist. PDF-Bearbeitung nimmt Zeit in Anspruch und am besten setzt man sich an den Schreibtisch vor einen großen Bildschirm und setzt seine PDF-Modifikationen mit der Maus und Tastatur um.

Ursprünglich war die PDF Web App als offline Webseite angedacht. Die Webseite sollte man auch ohne Internetverbindung wie ein Programm verwenden können, nur dass keine Installation notwendig ist. Das einzige Programm, was man benötigt für die Ausführung der PDF Web App soll ein Browser sein. Durch Öffnen der index.html im Browser kann man die PDF Webapp auf einfache Art und Weise starten und benutzen. Eine Anforderung an mich selbst ist, dass die PDF Web App als Open Source-Projekt für jeden kostenlos nutzbar sein soll, da ich selbst nur kostenlose PDF-Bearbeitungsprogramme verwende. So soll die PDF Web App auch u.a. für Studenten mit wenig Geld zur Verfügung stehen. Das Open Source-Projekt soll als öffentliches Repository auf Github vorhanden sein und jeder kann es downloaden. Da das PDF-Dateiformat ein offenes Dateiformat ist, finde ich, dass die Bearbeitungsprogramme für PDF-Dateien ebenfalls kostenlos sein sollten. Auf dem öffentlichen Github Repository soll man lediglich die PDF Web App runterladen können ohne Werbeprogramme oder ähnliches.

Des Weiteren wollte ich mich nur auf HTML, CSS und JavaScript beschränken. Folglich ist die PDF Web App eine reine Frontend Web Anwendung und ich habe kein Backend oder eine Datenbank implementiert. Dadurch habe ich die Programmierung und den Verwaltungsaufwand der PDF Web App simpler gemacht. Die Bedienung sollte möglichst intuitiv, eingängig und einfach gehalten sein, damit ein Nutzer nicht seitenlange Tutorials lesen muss und die Funktionen der Einstellmöglichkeiten möglichst durch Ausprobieren erfassen kann.

Um diese Anforderung zu erreichen, sollte eine GUI mit simplem, übersichtlichem und strukturiertem Design verwendet werden. Verschlüsselung, Formulare und Kommentare, sowie das incremental update wollte ich nicht implementieren, um den Arbeitsaufwand geringer zu halten.

Eine weitere Vorgabe der Gutachterin der Bachelorarbeit ist eine Marktübersicht inklusive der Beschreibung anderen PDF Wettbewerberprogramme zu erarbeiten.

Daraufhin habe ich mir einige freie und kostenpflichtige PDF-Programme und Onlinedienste ausgesucht:

- Adobe Acrobat
- foxit
- pdf-it
- PDF24
- PDFelement
- Soda PDF
- Nitro PDF Pro
- Ashampoo PDF Pro 3
- Infix 7
- PDFsam Basic und Advanced
- DrawboardPDF
- PDF Director 2 PRO
- Perfect PDF

2.1.1 Vorgaben für den Funktionsumfang der PDF Web App

Bei der Planung der PDF Web App habe ich mir einen Funktionsumfang ausgesucht.

Im Prinzip sollte die PDF Web App aus 8 Modulen bestehen:

- Reader
- Creator
- Merger
- Splitter

- Writer
- Drawer
- Geometry Editor
- Images Editor

Der Reader sollte eine Navigationseinheit enthalten, die die aktuelle Seitenzahl anzeigt, zu einer beliebigen Seite springen kann, eine Zoomfunktionalität implementieren und man soll zur vorherigen oder nächsten Seite blättern können. Zusätzlich soll eine Druckfunktionalität enthalten sein. Der Creator ist ein Modul zur Erstellung von leeren PDF-Dokumenten als Download mit beliebiger Größe, Anzahl an Seiten, Möglichkeiten die Orientierung (Portrait, Landscape, quadratisch) zu justieren und Presets zur DIN A Größe. Es gibt 2 Module zur Verwaltung von PDF-Dokumentkombinationen. Zum einen soll der Merger mehrere PDF-Dateien zusammenfügen und als eine Datei speichern können, dabei soll die Reihenfolge der einzelnen PDF-Dateien in einer Liste anpassbar sein und eine Druckfunktion enthalten. Hingegen soll der Splitter ein PDF zerteilen und als einzelne PDF-Dateien speichern können. Dabei gibt es verschiedene Split-Operationen: Nach jeder, nach jeder ungeraden, nach jeder geraden, nach einer bestimmten und nach einer Liste an Seiten kann der Splitter das PDF zerteilen. In meiner PDF Web App ist Speichern ein Synonym für den Download eines Output-PDFs. Beim Download soll man immer einen benutzerdefinierten Dateinamen setzen können. Diese 4 Module sind im Diagramm 2.1 graphisch dargestellt.

Der Editor besteht aus den restlichen 4 Modulen. Alle Editoroperationen beziehen sich lediglich auf bereits hinzugefügte Elemente. Es sollen keine schon im PDF bestehenden Objekte bearbeitet werden können. In jedem Editorteil soll es möglich sein zu drucken und das PDF zu downloaden, sowie alle neuen Elemente auf einen Schlag zu löschen. Außerdem sollen Elemente in beliebiger Reihenfolge übereinander stapelbar sein, d.h. man kann Elemente in den Vordergrund holen oder weiter nach hinten verschieben. Im Diagramm 2.2, was zeigt wie die Editormodule aufgebaut sind, ist die zuletzt genannte Funktionalität mit z-axis order gekennzeichnet. Mittels des Writers soll Text hinzugefügt werden können. Dieser neue Text kann gelöscht, verschoben, gedreht, die Schriftgröße und font family angepasst und eingefärbt werden. Als Font kann man zwischen Times Roman, Helvetic oder Courrir wählen bzw. einen benutzerdefinierten Font verwenden. Im Drawer hat man die Möglichkeit zu zeichnen und zu radieren. Dabei kann man die Größe und Farbe des Stifts bzw. Radierers anpassen. Das Geometrymodul soll geometrische Formen, wie Rechteck, Dreieck oder Ellipse hinzufügen, ihre Größe und Farbe verändern, Formen löschen, verschieben und drehen können. Zuletzt soll der Images-Editor Bilder vom Dateisystem hinzufügen, bereits platzierten Bilder löschen, verschieben und drehen können.

2.1.2 Qualitätsanforderungen

Ich habe den Ansporn, dass die PDF Web App sehr stabil und mit minimalen zeitlichen Verzögerungen arbeiten soll. Konkret bedeutet das, dass sich die Seite schnell aufbauen

Abbildung 2.1 Anforderungen an den Reader, Creator, Merger und Splitter der PDF Web App

und die Seiten von geöffneten PDFs in einer akzeptablen Zeitspanne laden sollen. Genauso wichtig ist mir eine gute Funktionalität, d.h. so wenig Laufzeitfehler wie möglich und dass die PDF Web App immer genau gleich funktioniert und nicht abstürzt. Maximale Effizienz sollte gegeben sein, das die PDF Web App als reine Frontend App ohne Backend auskommen soll. Es müssen keine Daten gespeichert werden. Genauso wichtig ist eine gute Benutzbarkeit. Im Einzelnen ist damit gemeint, dass der Anwender sich die Funktionalität so gut wie möglich selbst durch Ausprobieren erschließen kann. Die Buttonbeschriftung sollte selbsterklärend sein und die Layoutfarben so eingesetzt, dass sie eine Designkonsistenz zur Vermittlung von Funktionalitätsklassen aufweist. Die Änderbarkeit und Erweiterbarkeit sollte auf einfache Weise möglich sein, sodass auch andere Entwickler an der PDF Web App arbeiten können.

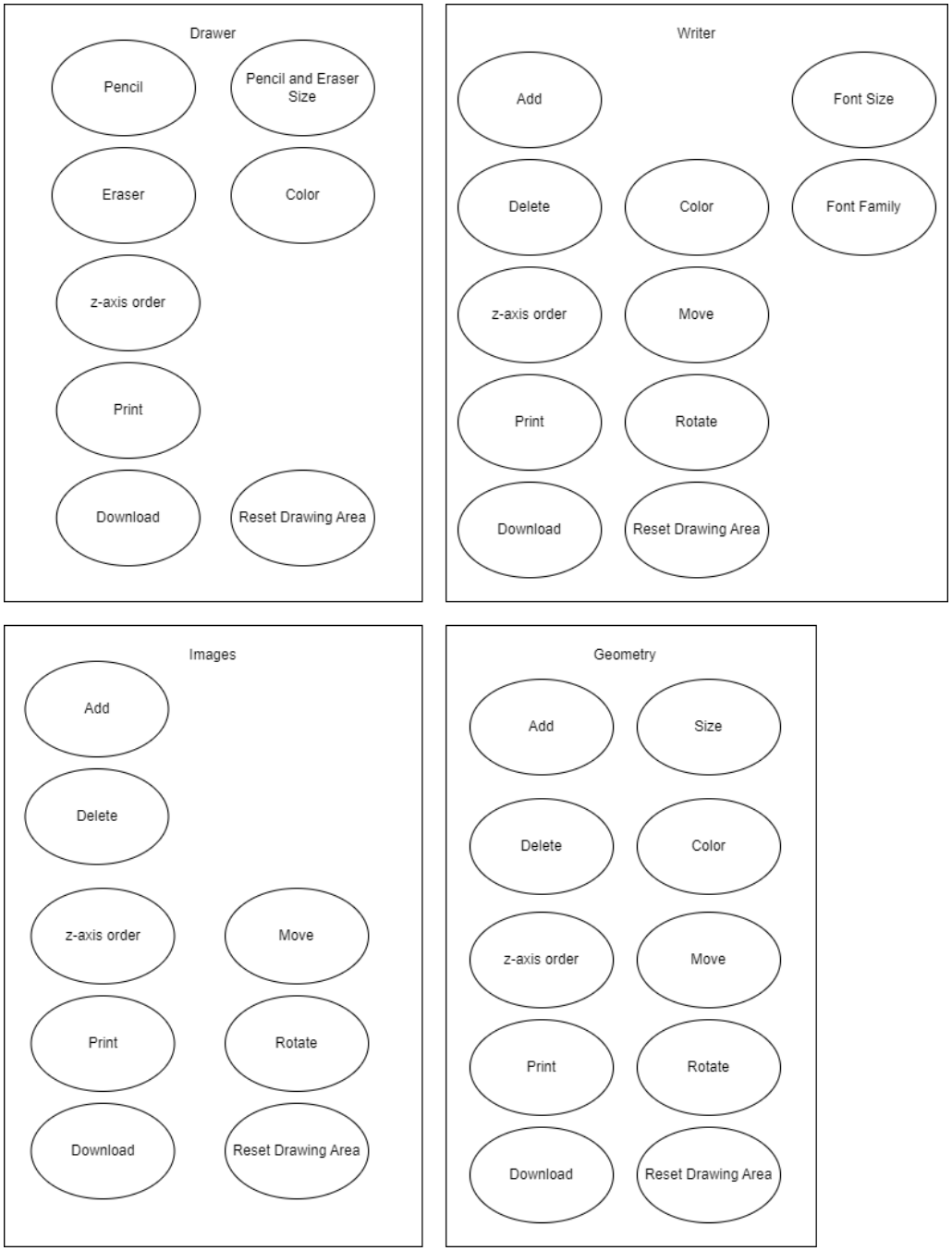

Abbildung 2.2 Anforderungen an den Editor der PDF Web App

Des Weiteren sollte die Portierbarkeit hoch sein, denn die PDF Web App sollte auf Windows, macOS und Linux immer gleich funktionieren. Eine Tablet oder Mobilversion ist nicht geplant. Mir war wichtig, dass man die PDF Web App als Offlineseite verwenden kann. Folglich habe ich keine Content Delivery Network (CDN) Links verwendet, sondern die verwendeten Libraries als Dateien eingebunden im Ordner lib, damit man die PDF Web App im Browser durch index.html öffnen kann und keinen Internetzugang benötigt. Außerdem soll man nur die PDF Web App und eventuelle Tutorials runterladen können. Es sollen keine zusätzlichen Werbedateien oder -programme zum Download angeboten werden.

2.2 Konzept und Methodik

2.3 Realisierter Funktionsumfang der PDF Web App

2.4 Bedienung der PDF Web App

Wenn man die PDF Web App zum ersten Mal öffnet, findet man die im Screenshot 2.3 abgebildete Startseite vor.

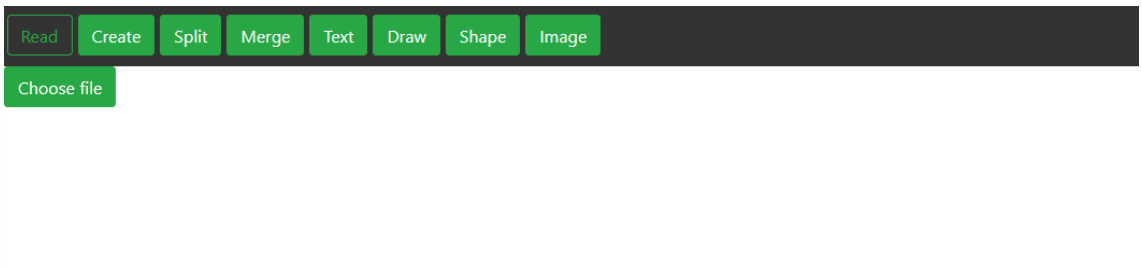

Abbildung 2.3 Startseite der PDF Web App

Initial ist der Reader ausgewählt. Ausgewählte Funktionen im Hauptmenü sind dunkelgrau unterlegt mit grüner Schrift. Der Button Create führt zum Creator für leere PDFs, Split zum Splitter fürs Seiten Zerteilen, Merge zum Merger für PDFs zusammenfügen und Text, Draw, Shape bzw. Image öffnen den Editor. Bei Read, Text, Draw, Shape und Image erscheint zunächst der Choose file Button, damit man im Dateisystem ein PDF-Dokument auswählen kann zum Lesen oder Bearbeiten. Klickt man auf Choose file wird der Dateibrowser geöffnet und man kann ein einzelnes PDF auswählen zum Öffnen. Der Screenshot 2.4 zeigt den Reader mit geöffneter PDF-Datei.

Falls eine andere Dateiart im Dateibrowser geöffnet wurde erscheint die Fehlermeldung in Screenshot 2.5.

Auch bei einer verschlüsselten PDF-Datei wird eine im Screenshot 2.6 dargestellte Fehlermeldung angezeigt.

Bei diesen gezeigten Fehlermeldungen kann man einfach erneut den Dateibrowser aufrufen mit Choose file oder einen anderen Menüpunkt wählen, damit die Fehlermeldung verschwindet. Hat man eine PDF-Datei im Reader geöffnet präsentieren sich einem 2 Zeilen mit Funktionsbuttons. Mittels Previous und Next kann der Benutzer zur vorherigen bzw. nächsten Seite blättern. Zwischen diesen Buttons wird die aktuelle Seite im input field angezeigt und daneben Anzahl an Seiten im Dokument. Im input field kann man eine Seite eingeben und der Reader springt direkt zu dieser Seite. Durch die Buttons Plus und Minus kann man rein- bzw. rauszoomen. Der aktuelle Zoomfaktor wird angezeigt in Prozent und man kann den Zoomfaktor auch mittels Benutzerangabe manuell setzen mit und ohne Prozentzeichen. Spin CW und Spin CCW dreht die aktuelle Seite, die im input field angezeigt ist, um 90 Grad im Uhrzeigersinn (clockwise) und gegen den Uhrzeigersinn

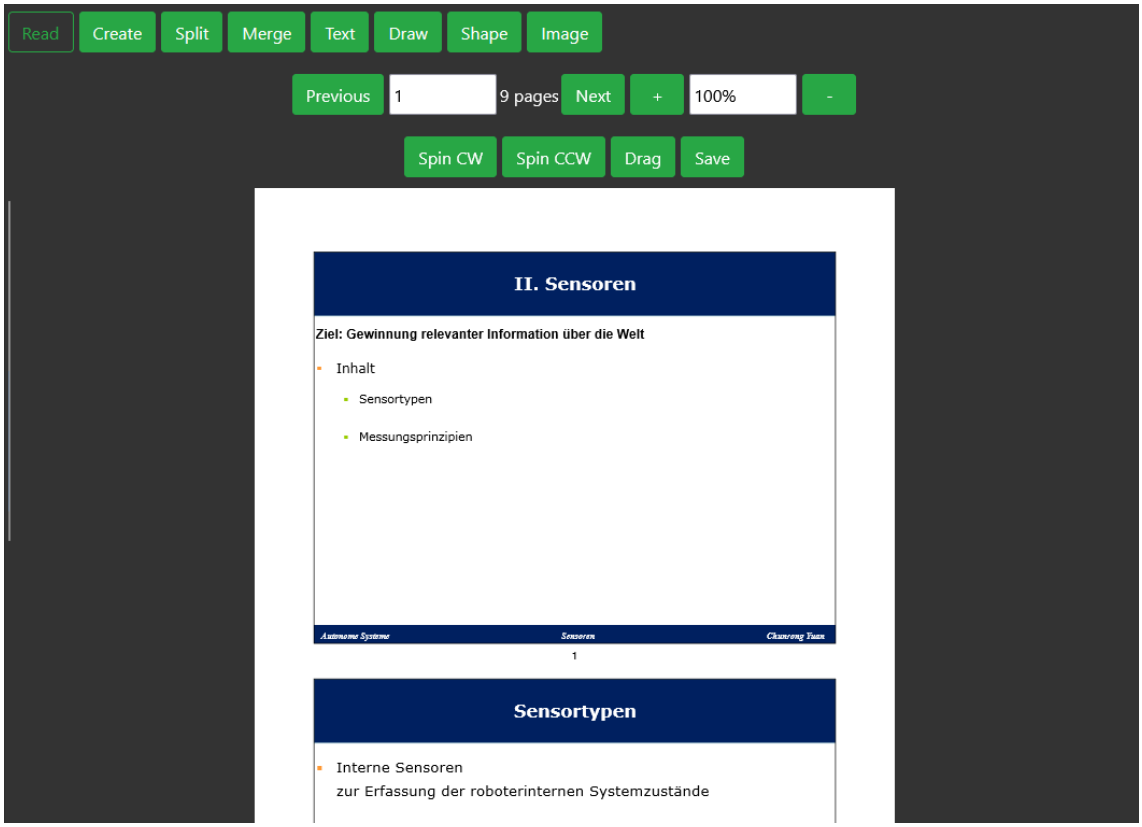

Abbildung 2.4 Geöffnetes PDF im Reader der PDF Web App

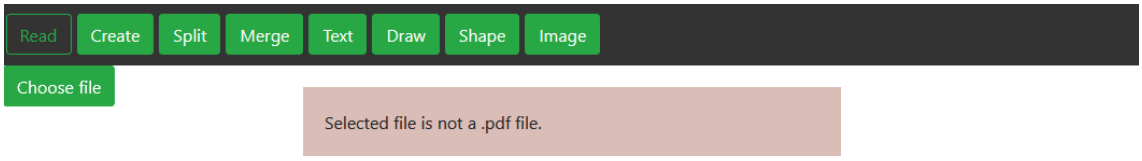

Abbildung 2.5 Fehlermeldung bei einer nicht-PDF-Datei

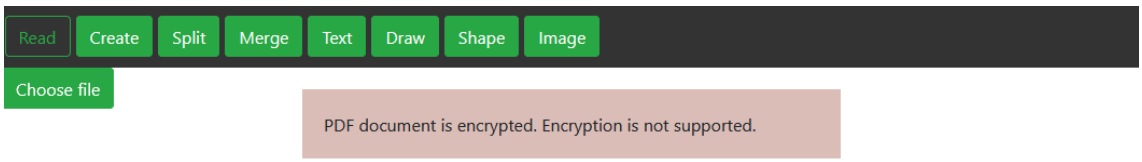

Abbildung 2.6 Fehlermeldung bei einem verschlüsselten PDF

(counterclockwise). Durch die Funktion Drag kann man die aktuelle Seite verschieben im Viewport. Das ist nützlich, wenn man an das PDF nah rangezoomt hat und die Seite nur teilweise sehen kann. Der Save Button downloaded das aktuelle PDF im Downloads-Ordner des Benutzers.

Der Creator ist mittels Create Buttons im Hauptmenü aufrufbar. Im Screenshot 2.7 ist die GUI vom Creator dargestellt.

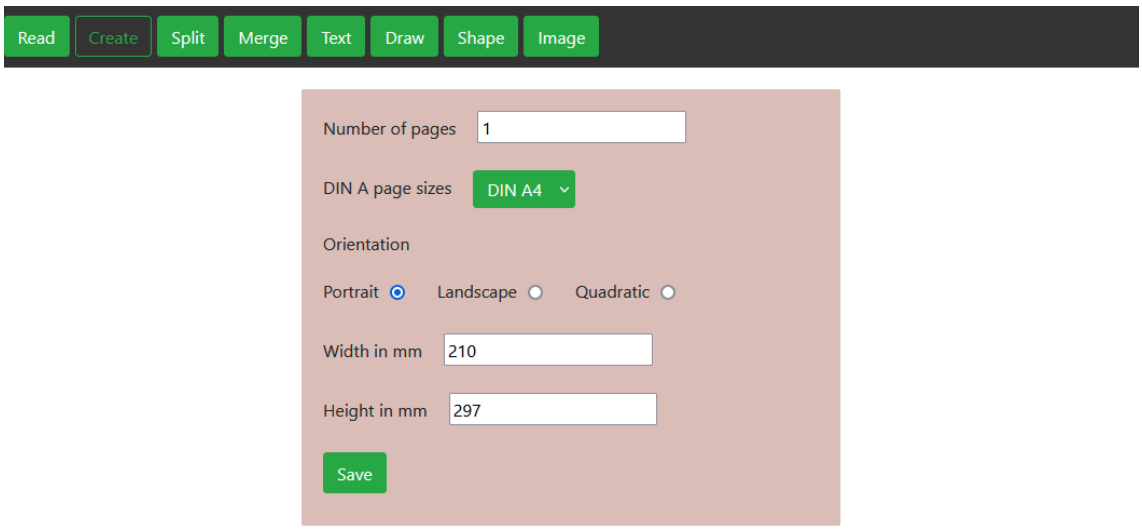

Abbildung 2.7 Creator GUI der PDF Web App

Man gibt eine Anzahl an gewünschten Seiten des leeren PDFs ein und die Breite und Höhe in mm. Wahlweise kann man den Selector verwenden um ein DIN A-Preset zu verwenden. Mittels der Schnellauswahl kann man die Orientierung bestimmten: Portrait, Landscape oder Quadratisch. Der Save Button downloaded das leere PDF.

Zum Splitter kann man mit dem Split Button gelangen, der vom Screenshot 2.8 gezeigt wird.

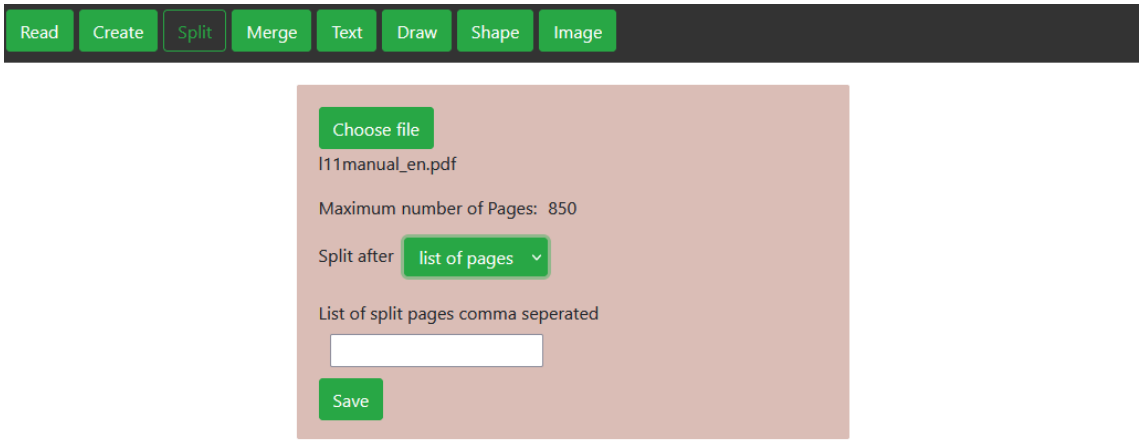

Abbildung 2.8 Splitter GUI der PDF Web App

2.5 Implementierung der PDF Web App

2.5.1 Werkzeuge

- Entwicklungsumgebung: Visual Studio Code

- ausführende Programme: Browser Firefox, Opera, Google Chrome, Microsoft Edge
- Sprachen: JavaScript, CSS, HTML
- Libraries: Vue JS 3, PDF-LIB, PDF.js, downloadjs
- PDFelement
- Soda PDF
- Nitro PDF Pro
- Ashampoo PDF Pro 3
- Infix 7
- PDFsam Basic und Advanced
- DrawboardPDF
- PDF Director 2 PRO
- Perfect PDF

2.6 Testdurchführung der PDF Web App

2.6.1 Funktionale User Tests

2.6.2 Stress Tests

3 Marktübersicht und PDF Wettbewerberprogrammme

Die Popularität von PDF-Dateien ist seit 2008 rasant angestiegen in der globalen Informationsübertragung. Täglich werden weltweit 2,5 Milliarden PDF Dokumente erzeugt. Seine Beliebtheit verdankt PDF vor allem an der plattformübergreifenden Kompatibilität (Desktop-Computer, Tablets und Smartphones), denn PDF Dokumente ist auf mehr als 1,5 Milliarden Geräten ohne zusätzliche Software lesbar. Über 80% der geschäftlichen Dokumente werden als PDF Datei weitergegeben. [49] 90 % der Büroangestellten wollen auf das PDF Dateiformat nicht mehr verzichten. Drei Viertel aller archivierten Dokumente sind PDF Dokumente (2023). Bis 2025 werden über 3 Millarden Dollar jährlich für PDF Editoren ausgegeben werden (2023) [61]. Im Jahr 2015 gab es 1,6 Milliarde PDF-Dokumente im Web und im Jahr 2019 wurden PDF-Dateien bei ca. 99 % Firmen und Regierungsinstitutionen weltweit verwendet [45].

3.1 Die Firma Adobe Systems Incorporated

1982 wurde die Firma Adobe Systems Incorporated von John Warnock und Chick Geschke gegründet. Adobe Illustrator war das erste auf PostScript basierende Grafikprogramm, welches 1988 auf den Markt gebracht wurde [8].

3.2 Aktueller Stand von Forschung und Technik

3.3 Zielgruppenanalyse

3.3.1 Soziodemographische Kriterien

3.3.2 Psychographische Kriterien

3.4 Umfeldanalyse

3.4.1 Politische Aspekte

3.4.2 Soziographische Aspekte

3.5 Relevanz von PDF in verschiedenen Marktbranchen

Die PDF-Datei ersetzt als elektronisches Informationsaustauschformat in privaten Organisationen, Behörden und im Bildungswesen Papier basierte Arbeitsprozesse.

Vor allem in Behörden wird der E-Mail-Verkehr noch frequent verwendet. Starke Kompressionsalgorithmen ermöglichen es, wenig speicherintensive PDF-Dateien auf leichtem Wege per E-Mail verschickt werden.

Die PDF Spezifikation ist offen und sehr detailliert über sämtliche Aspekte des Dateiformats. Dadurch können Softwareunternehmen eigene Programme schreiben, die PDF Dokumente erzeugen, lesen und bearbeiten können.

Durchsuchbare PDFs werden in Verträgen, Rechnungen und Geschäftsunterlagen verwendet, damit Mitarbeiter*innen Informationen gezielter suchen und Daten abteilungsübergreifend effizienter verwaltet werden können. In Forschungsarbeiten und wissenschaftlichen Artikeln werden durchsuchbare PDFs bei der Überprüfung von Quellen oder dem Extrahieren von Zitaten hauptsächlich verwendet. Behörden, Bibliotheken und Unternehmen digitalisieren Dokumente zur Archivierung und wandeln sie in ein durchsuchbares PDF um, was den langzeitigen Bestand der Dokumente sichert. [34]

Das PDF/A-Dateiformat wird in Bibliotheken und Archiven zur digitalen Archivierung von Büchern, Zeitschriften und historischen Dokumenten verwendet. Auch im Behördenzweig und Verwaltungssektor hat PDF/A für die Aufbewahrung von Verwaltungsakten und rechtlichen Dokumenten seine Existenzberechtigung. Im Gesundheitswegen wird es außerdem zur Speicherung von Patientenakten und medizinischen Unterlagen verwendet. Hingegen im Finanzwesen werden mit ihm Geschäftsunterlagen und Finanzdokumente verwahrt. Unternehmen und Organisationen können mit PDF/A gesetzliche und Compliance-Vorschriften einhalten. [29]

PDF/VT-Dateien werden im Direktmarketing verwendet. Personalisierte Werbematerialien erhöhen die Wahrscheinlichkeit einer positiven Reaktion bei den Kundi*nnen auf die Werbebotschaft und verbessert die Bindung von Unternehmen und Kund*innen. Der Transaktionsdruck findet bei Finanzdienstleistungen, Versicherungen und E-Commerce besonderen Anklang. Beliebte Transaktionsdokumente sind Rechnungen, Kontoauszüge, Versicherungspolicen oder Bestellbestätigungen. [31]

PDF-Dokumente mit der PDF/UA-Kennzeichnung stärken den Ruf und die Reputation eines Unternehmens oder einer Organisation durch Engagement für Inklusion. [33]

Digitale Signaturen werden bei digitalen Freigabe-, Abnahme-, Genehmigungs- und Vertragsprozesse verwendet. PAdES wird in Rechtssystemen, Finanzwesen und Regierungssektor eingesetzt. [13]

3.6 Rolle von PDF in der Druckvorstufe und Designbranche

PDF hat sich zum Standard für den elektronischen Austausch von Dokumenten entwickelt und unterstützt beliebige Seitenformate. Vor allem in der grafischen Industrie wird PDF gerne verwendet, weil es eine plattformübergreifende Visualisierung auf allen Betriebssystemen bietet. Außerdem kann PDF für die schnelle Webansicht optimiert werden. Vor allem Bilder lassen sich ohne sichtbaren Qualitätsverlust und geringem Datenvolumen speichern, da sehr leistungsfähige Kompressionsalgorithmen in PDF implementiert wurden. Im Vergleich zu PostScript-Dateien erzielen die kompaktere Codierung von Seiteninhalten, dem einmaligen Speichern von identischen Bildern und die Verwendung von Kompressionsalgorithmen eine maßgeblich kleinere Dateigröße bei PDF. Korrekturänderungen in PDF-Dateien sind kurz vor dem Druck noch möglich und PDF entwickelte sich zunehmend zum Containerformat für alle grafischen Elemente. Die Produktion von Druckerzeugnissen wird somit wesentlich flexibler und sicherer. Downsampling und Kompression beschleunigt den Transport von der Agentur zum Dienstleistungsbüro enorm. Heute ist es das meist eingesetzte Format bei der Produktion von Druckvorlagen in der digitalen Druckvorstufe. Das Dateiformat funktioniert mit allen Druckern und überwindet Kompatibilitätsprobleme.

Textmodifikationen sind am meisten gefordert in der Druckvorstufe. Farbänderungen bzw. Farbkonvertierungen sind ebenso erforderliche Eingriffe. Änderungen am Layout oder Neugestaltungen von Seiten müssen im Originalprogramm erledigt werden und das modifizierte PDF muss in das Druck-PDF integriert werden. Ständige visuelle Kontrollen der PDF-Datei des Kunden mit der korrigierten Version werden durch eine Fachkraft im Betrieb durchgeführt. Am Ende jeder Korrektur steht die Prüfung auf PDF/X-Konformität, um als technisches Gütesiegel weitere Fehler für den Druck zu reduzieren.

Für die Betrachtung von Druckvorstufen-PDF-Dateien sollte immer Acrobat Pro bzw. Adobe

Reader verwendet werden, da viele Drittanbieter-PDF-Viewer druckvorstufenrelevante Informationen nicht fehlerlos darstellen können. Dank der Profile für unterschiedliche Geräte und Bedruckstoffe, kann eine Simulation des Ergebnisses am Monitor oder durch einen Prüfdruck bewerkstelligt werden. Dadurch steigt die Reproduktionssicherheit bei gleichbleibender Qualität und die Produktionszeiten verkürzen sich enorm. In den meisten Druckprojekten steht zu Beginn noch nicht fest, wann, wo und auf welchem Bedruckstoff gedruckt werden soll. Die zeitintensive Optimierung der Druckdaten kann durch PDF auf einen späteren Zeitpunkt verlegt werden [8].

In der Werbebranche werden PDF-Dateien vor allem für Korrekturabzüge verwendet. Ein Korrekturabzug ist eine Skizze bzw. Designvorlage des Werbeprodukts in vom Kunden gewünschten Position und Größe des Druckmotivs auf dem Werbeartikel. Als digitales Layout wird der Korrekturabzug mit dem Kunden abgestimmt und seine Änderungswünsche vor dem Druck entgegengenommen [62]. Zum Erstellen von PDF-Dateien mit geringer Dateigröße für den Online-Bereich muss der Output-Intent mit dem angehängten ICC-Profil entfernt werden, da ICC-Profile eine eher größere Dateigröße aufweisen. Für eine korrekte farbliche Simulation am Monitor in der Druckvorstufe ist es in einigen Fällen wünschenswert, dass ein Output-Intent gesetzt wird. Bei der Erzeugung einer ISO-konformen PDF-Datei, z.B. bei PDF/X, PDF/A oder PDF/E wird immer ein Output-Intent angelegt. Je nach Standard wird auch das Zielprofil eingebettet. Die direkte Ausgabe von PDF-Dateien über den Druckdialog in Acrobat Pro ist in der Praxis nicht mehr gängig, da alle gängigen Workflow-Systeme auf einem PostScript-3-RIP basieren. Intern konvertiert der PostScript-RIP die PDF-Datei in eine PostScript-Datei automatisch. Die Ausgabe von PostScript aus Acrobat Pro heraus spielt noch für Laserdrucker, Farbkopierer und Broschüren auf Farbdruckern eine Rolle.

In der Ausgabe ist die effektive Auflösung maßgeblich. Effektive Auflösung ist die Bildauflösung, die aus der Anzahl der Bildpunkte und der Fläche resultieren, auf der das Bild platziert wurde. Downsampling beeinflusst diese effektive Auflösung. Starke Artefakte fallen im Offsetdruck weniger auf als im Digitaldruck. Bilder sollten nach der PDF-Bearbeitung anschließend geschärft werden. Das wird vom Datenersteller oftmals vergessen. Falls im Druck-PDF ein falscher Zielfarbraum eingestellt wurde, wirkt sich das auf den Gesamtfarbauftrag aus. Eine Farbraumkonvertierung in den für das Druckergebnis gewünschten Farbraum muss vorgenommen und der maximale Gesamtfarbauftrag muss angepasst werden. Bilder, die nicht im Zielfarbraum vorliegen müssen erst in den Ausgabefarbraum überführt, separiert ausgegeben und Transparenzen reduziert werden. Jegliche Transparenzen in Objekten, vor allem Texte, Grafiken oder Bilder, werden beim speichern in ein Dateiformat, welches keine Live-Transparenzen unterstützt, reduziert. Zu diesen Dateiformaten gehören u.a. Encapsulated PostScript (EPS), JPEG, GIF, BMP, PDF 1.3 und niedriger. Nach der Transparenzreduzierung sollten für die Ausgabe Quellfarbprofile von den reduzierten Objekten entfernt werden, da sie oftmals zu unerwünschten Farbtransformationen im Ausgabegerät führen können. Das Anbringen von Transparenzen auf Objekte mit Volltonfarben

ist mit Ausnahme der Deckkraftänderung verboten. Bilder können unscharf werden, wenn in den meisten Fällen eine zu niedrige Reduzierungsauflösung global eingestellt ist. Der Prozess der Transparenzreduzierung sollte auf den letztmöglichen Zeitpunkt in der Produktion von digitalen Daten verschoben werden.

Von einer medienneutralen Druckproduktion spricht man, wenn die Farbraumkonvertierung der neutral abgespeicherten Daten ausschließlich in RGB oder in RGB und CMYK zu einem möglichst späten Zeitpunkt stattfindet. Bei Logos werden medienneutrale PDF-Dokumente nicht bevorzugt. Durch das medienneutrale Konzept gewinnt man an Flexibilität und reduziert Mehrarbeit im Produktionsprozess, z.B. wenn ein späterer Wechsel des Druckverfahrens und der Papierklasse erfolgt. Der Gegensatz zur medienneutralen Arbeitsweise ist die verfahrensangepasste Arbeitsweise. In der Praxis ist primär der PDF-Empfänger für die Rastereinstellungen des RIPs verantwortlich. Die Anzahl der Sonderfarben ist in keinem PDF/X-Standard beschränkt. In der Verpackungsindustrie werden Mehrkanalproduktionen verwendet. Der Begriff Binding bezeichnet im Prozess der Druckdatenerstellung die Farbkonvertierung in die gewünschten Druckbedingungen. Beim Eearly Binding liegen alle verfahrensangepasste Daten schon im gewünschten Zielfarbraum vor. Hingegen bei medienneutralen Daten kann bei der PDF-Erzeugung eine Konvertierung in den Zielfarbraum erfolgen. Dies wird als Intermediate Binding bezeichnet. Im Prozess des Late Bindings wird die Konvertierung in den Zielfarbraum bei medienneutralen Daten erst in der Ausgabe vorgenommen.

Vor allem Office-Dokumente müssen für den Druck aufbereitet werden. Marketingabteilungen von Industrieunternehmen sind gezwungen auf die zur Verfügung stehenden Werkzeuge, den Microsoft-Office-Programmen bzw. OpenOffice-Programmen zurückzugreifen bei der Druck-PDF-Erzeugung. Meistens liegen dann alle Objekte in RGB vor, es fehlt die TrimBox und der Anschnitt, sowie keine Einzelseiten-PDFs liegen zum Ausschießen vor [8]. Ausschießen bezeichnet den Prozess, dass die einzelnen Seiten des Druckerzeugnisses in der Seitenmontage seitenverkehrt so angeordnet werden, dass sie nach dem Druck und der Weiterverarbeitung (Falzen) die korrekte Reihenfolge vorweisen [16]. Alle RGB-Farbräume müssen nach CMYK umgewandelt werden. Im Besonderen müssen beim Offsetdruck schwarze Texte auf den Farbaufzug K in CMYK gebracht werden. Schmuckfarben werden in CMYK oder eine andere Schmuckfarbe konvertiert bzw. gemappt. Die hinterlegten Alternativ-Farbwerte für Schmuckfarben weichen in der Vielfalt der Programme voneinander ab. Ursächlich ist, dass unterschiedliche Pantone Libraries implementiert wurden. Häufig sind in PDF-Dateien verschiedene Farbwerte für dieselbe Farbe anzutreffen.

Schriften können bei Einbettung in PDF exakt wiedergegeben werden, unabhängig ob es sich um eine Windows oder MacOS Schrift handelt. Da allgemeine Schriftinformationen immer eingebettet sind und die Zeilenlängen im Prinzip immer stimmen, können Druckvorstufenbetriebe zumindest immer erkennen, welche Schrift bzw. Schriftschnitt der Ersteller der

PDF-Datei ursprünglich vorgesehen hatte, falls die Schrift nicht eingebettet wurde. Fehlende Schriften in PDF-Dateien sind in der Druckvorstufe katastrophal. Beim nachträglichen Einbetten von Schriften in letzter Minute muss die richtige Schrift gesucht werden, denn der richtige Schriftname beschreibt nicht notwendigerweise dieselbe Schrift mit derselben Codierung bzw. Laufweite. Fehlende Zeichen, überlappende Buchstaben, unregelmäßige Abstände zwischen den Buchstaben, usw. sind gängige Probleme. Schriften können auch durch die PDF-Bearbeitung extrahiert (ausgebettet) werden. In der Druckvorstufe müssen Schriftbezeichnungen für die Ausgabe eindeutig benannt werden.

In der Praxis werden PDF-Dateien mit Hilfe von Prüfprofilen überprüft, Agenturen entwickeln nach eigenen Kriterien vorgefertigte Prüfroutinen und einen automatisch generierten Prüfbericht (Report). Der Prüfbericht sollte auf schnelle Fehlersuche im überprüften Dokument optimiert sein. Notwendige Korrekturen können im Originaldokument, bei PDF-Erstellung oder im vorliegenden PDF-Dokument ausgeübt werden je nach Schweregrad des Fehlers. Korrekturen im Originaldokument sind vorzuziehen, da sie Folgefehler minimieren können, jedoch liegt der Agentur oftmals nicht das Originaldokument oder ausschließlich eine unvollständige PDF-Datei des Originaldokuments vor. Eine Erstellung von Korrekturprofilen für Änderungen an der gesamten PDF-Datei kann dabei hilfreich sein. Man kann zwischen einer Vielzahl an voreingestellten Prüfprofilen und einer wesentlich kleineren Menge an Korrekturprofilen im Preflight in Adobe Acrobat Pro wählen. Außerdem kann der Prüfbericht als Kommunikationsmittel zwischen Auftraggeber und Agentur dienen, falls der Agentur das Originaldokument nicht ausgehändigt wurde. Das Preflight-Werkzeug in Acrobat Pro ist ein gängiges Werkzeug in der Druckvorstufe. Die Kontrolle von PDF-Dateien mit Preflight kann viel Geld sparen. Im tiefgreifenden Preflight-Check, der ein System vor dessen Einsatz überprüft, kann man die besten Ergebnisse erzielen, wenn das zum Job passende Prüfprofil verwendet wird. Beim Preflighting-Prozess werden fehlerhafte und Objekte mit speziellen Anforderungen gefunden, ihr Zustand abgefragt und dann dementsprechende ein Fehler, eine Warnung oder eine Information ausgegeben. Im Warnungsfall muss die Agentur entscheiden, ob das Objekt, was die Warnung verursacht hat, sich negative auf das Endprodukt auswirkt. Folglich können durch Preflight frühzeitig Mängel im Produktionszyklus beseitigt werden. Ein Report sollte für die automatisierte Auswertung in weiterführenden Workflow-Systemen im XML-Standard vorliegen. Erfolgreiche selbst erstellte Prüfprofile basieren auf qualitativ hochwertig gewählte Kriterien, wobei nicht zu viele verwendet werden sollten. Prüfprofile können als wichtigste Prüfkriterien das Dokument, Seiten, Bilder, Farbe, Zeichensätze, Rendering und Standard-Konformität enthalten. Nach der Einbringung von Korrekturen im PDF sollte eine technische und visuelle Überprüfung der Druckdatei stattfinden. In einem korrigierten Kunden-PDF sind die für die Druckerei wichtigen Druck-, Schneide- und Passermarken sowie der Bereich des Anschnitts meistens unerwünscht. Um diesen Bereich und alle Druckmarken unsichtbar zu machen wird die CropBox auf die Größe der TrimBox skaliert. PDF-Dateien, die nicht aus professionellen Anwendungen der Druckvorstufe erstellt worden sind, fehlen in den meisten Fällen die TrimBox bzw. BleedBox. Hat der Datenersteller Schnittmarken sehr genau manuell platziert,

so können fast alle PDF-Bearbeitungswerkzeuge davon die TrimBox bzw. BleedBox ableiten. Umschläge und Rücken werden häufig als Druckbogen als einseitiges PDF ausgegeben. Falls der Umschlag versehentlich als PDF-Datei aus Einzelseiten erstellt wurde, müssen diese Seiten zu einem Druckbogen zusammengeführt werden. Für die Weiterverarbeitung sollten leere Seiten und nicht druckbare Objekte, die als solche gekennzeichnet sind oder sich außerhalb des druckbaren Bereichs befinden, entfernt werden. Größe Dateien sind bei der Verarbeitung sehr zeitaufwendig und weniger ressourcenschonend. PDF-Dateien können sich durch Output-Intents, eingebettete Dateien, unzureichende Komprimierung von Bildern und Objekten, zu hohe Auflösung und die Präsenz nicht druckbarer Objekte in ihrer Dateigröße aufblähen. Häufig wird in der Agentur ein Abzug-PDF zur Druckfreigabe an den Auftraggeber versandt. Die Seiteninhalte eines Abzug-PDFs sollten gerastert werden, da fehlerhafte Grundeinstellungen im Adobe Reader zu einer abweichenden Darstellung des PDFs im Vergleich zum Aussehen nach dem Druck führen kann. Abzug-PDFs sollten eine Qualität besitzen, die ausreichend ist für eine Ansicht, jedoch für die Produktion bei einem anderen Druckdienstleister nicht genügt und der Output-Intent sollte zwecks Dateigrößenoptimierung entfernt werden. Falls viele Vektoren in einem Druckdokument enthalten sind, würde trotz PDF-Optimierung weiterhin eine hohe Auflösung in Vektorgrafiken bestehen bleiben und folglich nur eine geringe Dateigrößenreduktion erzielt werden. Das Rastern des Inhalts bietet einen praxisnahen Lösungsweg. Hierbei werden zu überdruckende Objekte in das Bild eingerechnet und Vektoren gerendert, wodurch eine maßgebliche Dateigrößenoptimierung erzielt werden kann. Speziell bei PDF/X-Dateien müssen bestimmte Inhalte als Metadaten vorliegen, wie der Dokumententitel, das Änderungsdatum der Datei, das PDF-Erstellerprogramm und das Setzes des Trapping Keys. Jeglicher Schutz oder Sperren sollte an einer druckvorstufentauglichen PDF-Datei vermieden werden, selbst wenn lediglich das Editieren gesperrt ist. Aktionen sind innerhalb druckfähiger Dateien untersagt, unabhängig von JavaScript oder Aktionen von Acrobat.

Das Entfernen von Druckmarken ist für den Datenersteller und die Druckvorstufe, die mehrere Seiten auf einem Druckbogen ausschießen muss, ein gängigere Arbeitsschritt. Die letzte Arbeitsphase im PDF-Workflow ist das Überfüllen bzw. Unterfüllen, um weiße Blitzer zu vermeiden. Spezialisten in der Druckvorstufe sollten das Hinzufügen von Traps übernehmen bzw. über dessen Notwendigkeit entscheiden. Der Druckvorstufenbetrieb gibt dem Auftraggeber den ISO-Standard der PDF-Datei für das Druckerzeugnis vor. Nur wenn sich die Ausgangsdatei mit der zum Druck verwendeten Enddatei in allen Einzelheiten deckt, kann man sich Reklamationen, Geld, Zeit und Diskussionen sparen. Sind die Druckdaten erstellt, müssen die enthaltenen Druckseiten noch in der Druckvorstufe für die Ausgabe auf die Druckplatte bzw. für die Bebilderung von Digitaldrucksystemen in eine korrekte Anordnung gebracht werden. Dieser Prozess, genannt Ausschießen, wird üblicherweise von der Druckvorstufe digital ausgeführt. In Produktionsumgebungen, wo auf Digitaldrucksystemen direkt ausgegeben wird, muss das Ausschießen auch vom Datenersteller selbst vor Ort vorgenommen werden. Ziel des Ausschusses ist eine schnellere und kostengünstigere Produktion durch die bestmöglichste Ausnutzung des Papiers bzw. Materials und nachfolgenden

Schritte (Zusammentragen, Binden und Schneiden) der Endverarbeitung zu ermöglichen und vereinfachen. In der Praxis wird am häufigsten der Ausschuss durch eine Softwarelösung nach dem Designprozess praktiziert. Vor allem in der Magazin- und Zeitungsproduktion, wo Personen zu gleicher Zeit auf unterschiedlichen Seiten arbeiten und diese zu schwankenden Zeitpunkten fertigstellen, wird nach dem Designprozess ausgeschossen. Es gibt jedoch auch die Möglichkeiten im Erstellungsprogramm, beim Drucken auf dem Drucker oder im RIP auszuschießen. Da der wirtschaftliche Druck auf die produzierenden Unternehmen immer mehr zunimmt, müssen einzelne Produktionsschritte immer mehr zeitlich gestaucht werden und im Idealfall automatisch ablaufen. Für einen automatisierten PDF-Worklfow muss dem Trapping-Key ein eindeutiger Wert zugewiesen werden. Elemente, die nicht für den Druck vorgesehen werden, müssen aus dem druckbaren Bereich entfernt werden [8].

Statische PDF-Dateien sind nicht responsive, d.h. eine optimale Anpassung an verschiedene Endgeräte ist nicht garantiert und sie haben eine lange Ladezeit. Daher erstellt man einen PDF-Online-Blätterkatalog oder ePaper. Es handelt sich um eine Konvertierung von PDF in HTML5 mit entsprechenden Tools. Den PDF-Blätterkatalog kann man in seine Hompage einbinden oder einfach über Social-Media-Kanäle oder per Newsletter teilen [63].

3.7 Freie PDF Programme und Onlinedienste

PDF Dateien lassen sich in vielen Programmen einfach über den Druckdialog erstellen. Apple hat das Lesen von PDF Dokumenten in seiner Apples Vorschau integriert. Viele Webbrowser stellen PDF Viewer bereit, so Google Chrome seit 2010 [2].

3.7.1 foxit

Produktseite: `https://www.foxit.com/` Die Firma foxit hat einen kostenlosen PDF Editor auf den Markt gebracht, der sehr umfangreiche Funktionen bietet.

3.7.2 PDF24

Produktseite: `https://www.pdf24.org/en/`

3.7.3 PDFsam

Produktseite: `https://pdfsam.org/`

3.7.4 DrawboardPDF

Produktseite: `https://www.drawboard.com/pdf/pdf`

3.7.5 PDFCreator

PDF Dokumente und Dateien erzeugen

3.7.6 LibreOffice

PDF Dokumente und Dateien erzeugen

3.7.7 OpenOffice

PDF Dokumente und Dateien erzeugen

3.7.8 ghostscript

3.8 Kostenpflichtige PDF Programme und Onlinedienste

3.8.1 Adobe Acrobat

Die nicht-Pro Version von Acrobat kann prüfen, ob es sich bei dem geöffneten PDF-Dokument um ein PDF/A-Dokument handelt und auf dessen Konformität prüfen. Zusätzlich kann man sich die Kompatibilität mit anderen PDF-Dateiformaten PDF/X, PDF/E, PDF/VT und PDF/UA anzeigen lassen. [29] Acrobat kann über JavaScript ferngesteuert werden. Dazu muss man die Berechtigung zur Ausführung von JavaScript erteilen. [8]

3.8.2 Adobe Acrobat Pro

PDF-Inhalte können bearbeitet und Dokumente mit digitalen Signaturen unterzeichnet werden. [34] Eine PDF-Datei kann in eine PDF/A-Datei inklusive seiner Varianten, PDF/X, PDF/UA, PDF/VT oder PDF/E konvertiert werden. Außerdem kann die Kompatibilität mit diesen Formaten überprüft werden in Preflight-Profilen. [29] Die Barrierefreiheit kann automatisch validiert werden oder ein neues Dokument kann direkt barrierefrei erstellt werden. Adobe Acrobat Pro kann andere Dokumentenformate wie HTML, DOC, DOCX, TXT und RTF in PDF konvertieren, PDF in andere Dateiformate wie Microsoft Word exportieren oder Dokumente unterschreiben. [64] Mit dem Werkzeug Scan & OCR kann man Pixelbilder als PDF und gescannte PDF-Dokumente in ein durchsuchbares PDF umwandeln. [34] Installiert man Acrobat Pro in Windows, steht dem Anwender ein Adobe PDF-Drucker mit entsprechender PPD-Datei zur Verfügung. [8]

3.8.3 Onlinetools von Acrobat

Produktseite:

`https://www.adobe.com/de/acrobat/online.html`

`https://www.adobe.com/de/acrobat/online/convert-pdf.html` Mit den Adobe Acrobat Onlinetools kann man über den Browser verschiedene Dateitypen in PDF umwandeln, unter anderem PDF in JPEG oder andere Bildformate, PDF Dateien bearbeiten und Kompression anwenden. Die Onlinetools können außerdem PDF in Word umwandeln. [34] Der Adobe Acrobat PDF-Converter der Onlinetools kann DOCX, DOC, XLSX, XLS, PPTX, PPT, TXT, RTF, JPEG, PNG, TIFF, BMP, sowie Adobe eigene AI-, INDD- und PSD-Dateien in PDF konvertieren. [64] Die kostenlose Version des PDF-Converters kann nur begrenzt oft genutzt werden.

3.8.4 Acrobat Distiller

In Acrobat Distiller als Software-Interpreter können PostScript-Dateien in PDF konvertiert werden und umgekehrt. Bei der Erstellung einer PDF-Datei schneidet Distiller an der MediaBox-Grenze herausragende Elemente ab. [8]

3.8.5 Microsoft Word

Aus einem Word-Dokument lässt sich in Microsofts Word eine PDF-Datei, inklusive im PDF/A-Dateiformat, erstellen. [29]

3.8.6 pdf-it

Produktseite: `https://www.pdf-it.com/`

3.8.7 PDFelement

Produktseite: `https://pdf.wondershare.com/`

3.8.8 Soda PDF

Produktseite: `https://www.sodapdf.com/`

3.8.9 Nitro PDF Pro

Produktseite: `https://www.gonitro.com/`

3.8.10 Ashampoo PDF Pro 3

Produktseite: `https://www.ashampoo.com/de-ch/pdf-pro`

3.8.11 Infix 7

Produktseite: `https://www.iceni.com/infix.htm`

3.8.12 PDF Director 2 Pro

Produktseite: `https://pdfdirector.de/funktionen`

3.8.13 Perfect PDF

Produktseite: `https://soft-xpansion.de/products/perfect-pdf-12/`

3.9 PDF zu Word Programme und Onlinedienste

In PDF ist keine automatische Anpassung des Seiteninhalt-Layouts, wie z.B. in Microsoft Word, möglich. Daher kann ein PDF-Dokument nicht sinnvoll in das Word-Format umgewandelt werden ohne möglicherweise das ursprüngliche PDF-Layout zu beeinflussen und zu ändern, sowie die maximalen Bearbeitungsmöglichkeiten von Word ausschöpfen zu können.

3.10 PDF zu Latex Programme und Onlinedienste

3.11 Wettbewerbsanalyse

3.12 Unternehmensanalyse

3.12.1 Unternehmensziele

3.12.2 SWOT-Analyse

3.13 Marketingstrategie

3.13.1 Produkt

3.13.2 Preis

3.13.3 Promotion

3.13.4 Platzierung

3.13.5 Personal

4 Diskussion und Kritik

Es gibt einige Kritikpunkte in meiner PDF Web App, die zur weiteren Diskussion anregen. Einige wichtige Funktionalitäten sind noch nicht implementiert und ich plane diese noch in die PDF Web App in den nächsten Releases einzubauen. In meiner PDF Web App fehlt die Spiegelung von Text, Zeichnungen, Geometrie und Bildern als wichtige affine Transformation. Die anderen Transformationen Verschiebung, Skalierung und Rotation sind bereits in der PDF Web App vorhanden. Außerdem fehlt die Searchable PDF-Eigenschaft im Reader, d.h. der Benutzer kann keinen Text oder Bilder kopieren. Am besten benutze ich für diese Eigenschaft eine OCR-JavaScript Library, die jedes PDF in Searchable PDFs umwandeln kann. Falls der Browser oder Computer abstürzt und das in der PDF Web App geöffnete Dokument nicht gedownloaded wurde, sind die Daten verloren. Als Gegenmaßnahme könnte man einen Export und Import Button konfigurieren, der die hinzugefügten PDF-Objekte als JavaScript Object Notation (JSON)-Datei speichert. Bei Export würde diese Datei angelegt und im Ordner der PDF Web App gespeichert werden und mittels Import könnte man die PDF-Objekte dem geöffneten Dokument hinzufügen, sodass sie weiter bearbeitet werden können. Zusätzlich könnte man einen automatischen Export der PDF-Objekte als Backup implementieren, der z.B. alle 10 Minuten die hinzugefügten Elemente wie Text, Zeichnungen, Geometrie und Bilder in der JSON-Datei sichert.

Bei sehr vielen Seiten dauert es mehrere Sekunden bis alle Canvaselemente für die PDF Seiten erstellt worden sind und das PDF angezeigt wird. Eventuell sollte ich noch ein inkrementelles Laden der PDF Seiten implementieren, d.h. das geladene Seiten bereits angezeigt werden, bevor das komplette PDF geladen wurde. Beizeiten kommt es vor, dass es länger dauert bis sich die Seite der PDF Web App aufbaut. Vermutlich liegt das an der Auslastung des Browser Caches. Wenn man den Browser Cache leert, baut die Seite sich meistens wieder ohne merkliche Zeitverzögerungen auf. Eine wichtige Vorgabe habe ich nicht umgesetzt. Ich hatte zuerst einen Print Button im Reader implementiert. Als ich dann später das Drehen von Seiten programmiert habe, hat die Druckfunktionalität zu viele Probleme bereitet, vor allem, dass sie in einem neuen Tab geöffnet werden sollte. Das Problem liegt hauptsächlich an der same-origin policy in Browsern. Die same-orgin policy blockiert den Lesezugriff von Ressourcen, die von einer anderen Herkunft geladen werden. Diese Sicherheitsfunktionalität regelt wie Dokumente und Scripte von einer Herkunft mit Ressourcen anderen Ursprungs interagieren [65]. Ein weitere Kritikpunkt ist, dass die Move-Operation von Elementen nicht optimal funktioniert. Man verliert die control box, wenn man zu schnell die Maus bewegt. Außerdem kann es vorkommen, dass man klickt und dann das Element etwas weiter weg springt. Man verliert die Maus, weil das mousemove Event schneller ausgelöst wird als der Browser Frames rendern kann. [66] Eine Funktionalität,

die ich oft benötige, ist die Kompression von PDF-Dokumenten. Von Anfang an wollte ich gerne eine Funktion dafür bieten, jedoch hatte ich keine geeignete JavaScript Library für Kompression gefunden. Ich habe darauf verzichtet die Kompression selbst zu implementieren.

Es gibt einige Optimierungsmöglichkeiten von bestehenden Funktionalitäten. Im Merger kann man jeweils nur einen Listeneintrag von ausgewählten PDFs, die zusammengefügt werden sollen, auswählen. Da könnte man programmieren, dass man mehrere Dateien auf einmal auswählen kann, um sie dann aus der Liste zu entfernen. Genauso wäre es sinnvoll im Dateibrowser mehrere PDF-Dateien selektieren zu können. Beim Radieren von Zeichnungen im Draw Editor könnte man die Ebene, wo die Zeichnung komplett weg radiert wurde automatisch entfernen. Zur Zeit ist es in meiner PDF Web App so, dass der Benutzer die leere Ebene manuell löschen muss. Das automatische Entfernen von leeren Ebenen hat Vor- und Nachteile. Eventuell würde es den intuitiven Arbeitsfluss des Nutzers beim Zeichnen stören, falls die Ebene automatisch entfernt würde. Ebenfalls hat die aktuelle Verwendung von absoluten Rotationen von Elementen Vor- und Nachteile. Relative Rotationen könnten ein intuitives Arbeiten ermöglichen und würden Kopfrechenschritte sparen. Jedoch haben absolute Rotationen den Vorteil, dass die Rotation immer gleich ist und somit besser nachvollziehbar für den Benutzer ist. Insgesamt ist die leichte Bedienbarkeit der PDF Web App zumindest im Editor nicht ganz gelungen. Für das Verständnis von Box Mode und Layer Mode, sowie deren Vorzüge, muss man die Tutorials gelesen haben. Die Buttons sind außer Spin CW (clockwise) und Spin CCW (counterclockwise) aussagekräftig beschriftet, denke ich.

Ein großer Pluspunkt der PDF Web App ist, dass sie nicht installiert werden muss. Das bedeutet auch, dass keine Deinstallation, die Dateien zurücklässt, durchgeführt werden muss. Gerne hätte ich eine Tablet Version implementiert, um das Zeichnen zu erleichtern. Nichtsdestotrotz funktioniert das Zeichnen mit einem Graphic Tablet ohne Probleme. Ich unterstütze keine Verschlüsselung von PDF-Dokumenten, da PDF-Dateien unsicher sind und nicht für vertraulichen Inhalt verwendet werden sollten.

4.1 Erfahrungen bei der Programmierung der Tablet Version

Gerne hätte ich die PDF Web App auch für das Tablet auf Android und iOS zum Laufen gebracht. Das GUI Layout der PDF Web App ist für eine Tablet Version tauglich und optimiert programmiert worden. Jedoch habe ich viele Probleme bei der Implementierung feststellen können. Zunächst habe ich herausgefunden, dass man auf Android und iOS Browsern keine lokalen HTML-Dateien ausführen kann. Folglich musste ich einen lokalen Webserver starten, um per localhost:<port> die Webseite im Firefox Browser des Tablets aufrufen zu können. Zu der Zeit hatte ich noch nicht gewusst, dass man die PDF Web App mit Github pages kostenlos als Webseite betreiben kann. Auf Android hatte ich damals die kostenlose Play Store App Simple HTTP Server gefunden. Auf dem iPad habe ich die App mit der iOS App Codesandbox und Documents zum laufen gebracht. Im Folgenden sind

die Links zu diesen Apps aufgeführt:

`https://play.google.com/store/apps/details?id=com.phlox.simpleserver&hl=de&pli=1`

`https://codesandbox.io/codesandbox-for-ios`

`https://apps.apple.com/us/app/documents-file-manager-vpn/id364901807`

Später hatte ich sogar einen kleinen eigenen Webserver in Node JS Express geschrieben, den ich dann zum öffentlichen Github Repository der PDF Web App gepusht habe.

Die Mouse Events mousedown, mousemove, mouseup und alle drag Events werden nicht ausgeführt auf der Tablet Version. Folglich funktionierte das Zeichnen, Radieren, Bewegen von Elementen, Verschieben von Ebenen und Listeneinträgen im Merger nicht. Auf Stackoverflow hatte ich dann eine Lösung dafür gefunden: Ich muss mousedown mit touchstart, mousemove mit touchmove und moouseup mit dem touchend Event ersetzten. Das allein reihte nicht. Zusätzlich musste ich event.clientX und event.clientY wie folgt im Codebeispiel ersetzten:

```
1  const clientXReplace = (e.targetTouches[0] ? e.targetTouches[0].pageX : e
       .changedTouches[e.changedTouches.length-1].pageX);
2  const clientYReplace = (e.targetTouches[0] ? e.targetTouches[0].pageY : e
       .changedTouches[e.changedTouches.length-1].pageY);
```

Listing 4.1 e.clientX und e.clientY Ersetzung

Diese Lösung hat jedoch nicht für drag Events funktioniert, d.h. das Verschieben von Ebenen im Editor und Listenelemente im Merger hat mit der Maus nicht funktioniert. Daher habe ich die dragdroptouch JavaScript Library eingebaut. Sie ist auf der folgenden Webseite zu finden:

`https://github.com/Bernardo-Castilho/dragdroptouch`

Mit dieser Library haben zwar die drag Events funktioniert, jedoch nicht mehr die touch Events. Die touch Events wurden auch überschrieben. Eine Lösung habe ich bisher nicht gefunden. Möglicherweise wäre folgende jQuery Library UI Touch Punch zielführend:

`https://github.com/furf/jquery-ui-touch-punch`

Jedoch habe ich diese Library noch nicht ausprobiert, da ich mich dann gegen eine Tablet Version der PDF Web App entschieden hatte.

Fazit und Ausblick

Literatur

[1] Wikipedia. „PDF.“ (2023), Adresse: https://en.wikipedia.org/wiki/PDF (besucht am 23. 12. 2023).

[2] Wikipedia. „Portable Document Format.“ (2023), Adresse: https://de.wikipedia.org/wiki/Portable_Document_Format (besucht am 19. 12. 2023).

[3] Wikipedia. „Offener Standard.“ (2023), Adresse: https://de.wikipedia.org/wiki/Offener_Standard (besucht am 20. 12. 2023).

[4] Wikipedia. „Royalty-free.“ (2023), Adresse: https://en.wikipedia.org/wiki/Royalty-free (besucht am 23. 12. 2023).

[5] Tobias Lang. „Was ist ein Farbraum?“ (2016), Adresse: https://themen.rainbowprint.de/farbraum/ (besucht am 30. 12. 2023).

[6] Adobe Inc. „Pfade und Formen, Grundlagen.“ (2023), Adresse: https://helpx.adobe.com/de/indesign/using/paths-shapes.html (besucht am 30. 12. 2023).

[7] Typografie.info. „PostScript Type 0, Bedeutung/Definition.“ (o. D.), Adresse: https://www.typografie.info/3/wiki.html/p/postscript-type-0-r43/ (besucht am 20. 12. 2023).

[8] H. P. schneeberger, *PDF in der Druckvorstufe das umfassende Handbuch, PDF-Dateien erstellen, prüfen, korrigieren und ausgeben; PDF/X-1a bis PDF/X-5 sicher im Griff; Preflighting, Automatisierung, Standards u.v.m.* (Galileo Design), de. Bonn: Galileo Press, 2014, 910 S., Für Beruf und Ausbildung, ISBN: 978-3-642-38552-0.

[9] D. S. Peter Bühler Patrick Schlaich, *PDF, Grundlagen - Print-PDF - Interaktives PDF*, de. Berlin: Springer-Verlag GmbH Deutschland, 2018, 97 S., ISBN: 978-3-662-54615-4. DOI: 0.1007/978-3-662-54615-4.

[10] Soft Xpansion GmbH & Co. KG. „PDF: Grundlagen eines Dateiformats.“ (2013), Adresse: https://soft-xpansion.com/files/cc/PDF-Grundlagen.pdf (besucht am 21. 12. 2023).

[11] Wikipedia. „XFA.“ (2023), Adresse: https://en.wikipedia.org/wiki/XFA (besucht am 21. 12. 2023).

[12] Adobe Inc. „PDF-Ebenen.“ (2023), Adresse: https://helpx.adobe.com/de/acrobat/using/pdf-layers.html (besucht am 23. 12. 2023).

[13] Adobe Inc. „PAdES: Elektronische Signaturen in PDF-Dokumenten. Für was steht die Abkürzung PAdES, welche Vorteile hat das Format und wie kannst du selbst ein Dokument digital signieren. Wir zeigen es dir.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pades.html (besucht am 26. 12. 2023).

[14] Adobe Inc. „PDF/E: das Dateiformat für Engineering und technische Kommunikation. Erfahre, was PDF/E-Dateien auszeichnet, in welchen Bereichen sie Anwendung finden und wie du selbst eine PDF/E-Datei erstellst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pdf-e.html (besucht am 26. 12. 2023).

[15] Adobe Inc. „PDF/X-Dateien: effizienter drucken. Wir erklären dir, was man unter PDF/X-Dateien versteht, wie sie genau funktionieren und was der Unterschied zu originären PDFs ist.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pdf-x.html (besucht am 26. 12. 2023).

[16] bm - gesellschaft für bildung in medienberufen mbh Köln, *Stoffsammlung Mediengestalter*, de. Köln, o. D., 190 S.

[17] aktuelle-druckinfos.de. „Halbton- und Vollton.“ (o. D.), Adresse: https://www.aktuelle-druckinfos.de/wiki-halbton-und-vollton (besucht am 30. 12. 2023).

[18] Adobe Inc. „Vollton- und Prozessfarben, Grundlagen.“ (2023), Adresse: https://helpx.adobe.com/de/indesign/using/spot-process-colors.html (besucht am 30. 12. 2023).

[19] ENZYKLO.DE DEUTSCHE ENZYKLOPÄDIE. „Strichbild Bedeutung.“ (o. D.), Adresse: https://www.enzyklo.de/Begriff/Strichbild (besucht am 30. 12. 2023).

[20] BenQ. „ICC-Profil Grundlagen.“ (2021), Adresse: https://www.benq.eu/de-de/knowledge-center/knowledge/icc-profile-basics.html (besucht am 20. 12. 2023).

[21] PREPRESS Secrets. „Die Rolle des Profile Connection Space.“ (2015), Adresse: https://www.prepress-secrets.at/index_files/profile-connection-space.html (besucht am 20. 12. 2023).

[22] PROJECT CONSULT. „PDF Standards.“ (o. D.), Adresse: https://www.project-consult.de/themen/pdf-standards/ (besucht am 23. 12. 2023).

[23] Adobe Inc. „Tranzparenz-Reduzierung (Acrobat-Pro), Transparenzen sind ein gern genutztes Stilmittel in der Gestaltung. Bei der Vorbereitung für den Druck müssen sie allerdings in der Regel reduziert werden. Was heißt das?“ (2023), Adresse: https://helpx.adobe.com/de/acrobat/using/transparency-flattening-acrobat-pro.html (besucht am 30. 12. 2023).

[24] Primus-Print.de Die Besten drucken online. „Transparenzen.“ (o. D.), Adresse: https://www.primus-print.de/daten/transparenzen/ (besucht am 30. 12. 2023).

[25] Adobe Inc. „Work with color profiles.“ (2023), Adresse: https://helpx.adobe.com/photoshop/using/working-with-color-profiles.html (besucht am 30. 12. 2023).

[26] fine art printer. „Tiefenkompensierung.“ (o. D.), Adresse: https://www.fineartprinter.de/ger/FAQ/Tiefenkompensierung (besucht am 30. 12. 2023).

[27] Akash Choudhary. „callas software presents pdfDPartner, the first free DPart metadata viewer.“ (2021), Adresse: https://pdfa.org/callas-software-presents-pdfdpartner-the-first-free-dpart-metadata-viewer/ (besucht am 30. 12. 2023).

[28] callas. „Check for print and prepress related PDF 2.0 features.“ (o. D.), Adresse: https://help.callassoftware.com/m/pdftoolbox/l/1608299-check-for-print-and-prepress-related-pdf-2-0-features (besucht am 30. 12. 2023).

[29] Adobe Inc. „PDF/A: Wie unterscheidet es sich von PDF? Erfahre, was PDF/A-Dateien auszeichnet, was der Unterschied zu einem PDF ist und wie du selbst eine PDF/A-Datei erstellen kannst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pdf-a.html (besucht am 25. 12. 2023).

[30] Adobe Inc. „Revisionssichere Archivierung mit einem Dokumentenmanagementsystem (DMS). Lerne, wie du Dateien revisionssicher mit einem Dokumentenmanagementsystem archivierst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/audit-proof-archiving.html (besucht am 25. 12. 2023).

[31] Adobe Inc. „PDF/X-Dateien: effizienter drucken. Wir erklären dir, was der ISO-Standard PDF/VT bedeutet, wofür du den Dateityp brauchst und wie du selbst ein PDF/VT erstellen kannst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pdf-vt.html (besucht am 26. 12. 2023).

[32] Mozilla. „MIME types (IANA media types).“ (o. D.), Adresse: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types (besucht am 01. 01. 2024).

[33] Adobe Inc. „PDF/UA: So erstellst du barrierefreie PDFs.PDF/UA: So erstellst du barrierefreie PDFs. Wir erklären dir, was eine PDF/UA-Kennzeichnung ist, wofür du sie brauchst und wie du selbst barrierefreie PDF-Dokumente erstellen kannst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files/pdf-types/pdf-ua.html (besucht am 26. 12. 2023).

[34] Adobe Inc. „So erstellst du ein durchsuchbares PDF. Lerne, was ein Searchable PDF ist, welche PDF-Arten es gibt und wie du selbst eine durchsuchbare PDF-Datei erstellst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files.html (besucht am 25. 12. 2023).

[35] Wikipedia. „Internationale Beleuchtungskommission.“ (2023), Adresse: https://de.wikipedia.org/wiki/Internationale_Beleuchtungskommission (besucht am 30. 12. 2023).

[36] Wikipedia. „International Commission on Illumination.“ (2023), Adresse: https://en.wikipedia.org/wiki/International_Commission_on_Illumination (besucht am 30. 12. 2023).

[37] PrintWiki, The Free Encyclopedia of Print. „Open Prepress Interface.“ (o. D.), Adresse: http://printwiki.org/Open_Prepress_Interface (besucht am 20. 12. 2023).

[38] HELIOS. „Welche Vorteile hat DeviceN für die Druckvorstufe?“ (o. D.), Adresse: https://www.helios.de/web/DE/news/deviceN_prepress.html (besucht am 20. 12. 2023).

[39] Proof.de. „Schwarzpunktkompensation.“ (o. D.), Adresse: https://www.proof.de/lexikon/schwarzpunktkompensation/ (besucht am 30. 12. 2023).

[40] TechTarget. „PPD file (Postscript Printer Description file).“ (o. D.), Adresse: https://www.techtarget.com/whatis/definition/PPD-file-Postscript-Printer-Description-file (besucht am 29. 12. 2023).

[41] Wikipedia. „Seitenbeschreibungssprache.“ (2021), Adresse: https://de.wikipedia.org/wiki/Seitenbeschreibungssprache (besucht am 20. 12. 2023).

[42] Adobe Inc. „PostScript, LANGUAGE REFERENCE third edition.“ (1999), Adresse: https://web.archive.org/web/20090419181826/http://www.adobe.com/devnet/postscript/pdfs/PLRM.pdf (besucht am 20. 12. 2023).

[43] Wikipedia. „PostScript.“ (2023), Adresse: https://de.wikipedia.org/wiki/PostScript (besucht am 19. 12. 2023).

[44] Ange Albertini. „PDF 101 & PDF Secrets, Learning the PDF basics, applying it to hide/reveal informations in documents.“ (2014), Adresse: https://media.ccc.de/v/MRMCD2014_-_6007_-_en_-_grossbaustelle_ber_-_201409051830_-_pdf_101_pdf_secrets_-_ange_albertini (besucht am 29. 12. 2023).

[45] Fabian Ising, Vladislav Mladenov. „How to Break PDFs, Breaking PDF Encryption and PDF Signatures.“ (2019), Adresse: https://media.ccc.de/v/36c3-10832-how_to_break_pdfs (besucht am 29. 12. 2023).

[46] Julia Wolf. „OMG WTF PDF, What you didn't know about Acrobat.“ (2011), Adresse: https://media.ccc.de/v/27c3-4221-en-omg_wtf_pdf (besucht am 29. 12. 2023).

[47] Ido Solomon. „BADPDF, Stealing Windows Credentials via PDF Files.“ (2019), Adresse: https://media.ccc.de/v/gpn19-45-badpdf-stealing-windows-credentials-via-pdf-files (besucht am 29. 12. 2023).

[48] FILEFORMAT. „PDF File Format, What is a PDF file?“ (o. D.), Adresse: https://docs.fileformat.com/pdf/ (besucht am 01. 01. 2024).

[49] Mehmet Bayram, formilo. „Popularität und Statistiken der PDF.“ (o. D.), Adresse: https://www.formilo.com/pdf-formulare/einfuehrung/popularitaet-statistiken/ (besucht am 19. 12. 2023).

[50] PDF Insecurity. „Attacks on PDF Certification (May 2021).“ (2021), Adresse: https://www.pdf-insecurity.org/index.html (besucht am 03. 01. 2024).

[51] DocuSign. „Understanding digital signatures, What is a digital signature, and how can you create one?“ (o. D.), Adresse: https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq (besucht am 01. 01. 2024).

[52] Fabian Ising, Vladislav Mladenov. „HOW TO BREAK PDF SECURITY, How to Break PDF Signature / How to Break PDF Encryption.“ (2019), Adresse: https://www.pdf-insecurity.org/download/2019-12-27-36c3-pdf-security.pdf (besucht am 05. 01. 2024).

[53] Jens Müller, Fabian Ising, Vladislav Mladenov, Christian Mainka, Sebastian Schinzel, Jörg Schwenk. „PDFex: Major Security Flaws in PDF Encryption.“ (2019), Adresse: https://web-in-security.blogspot.com/2019/09/pdfex-major-security-flaws-in-pdf.html (besucht am 02. 01. 2024).

[54] Y. L. Jonathan Katz, *Introduction to MODERN CRYPTOGRAPHY, Third Edition* (A CHAPMAN & HALL BOOK), en. Boca Raton, Florida: CRC Press, 2021, 626 S., ISBN: 978-135-11330-36.

[55] studyflix. „AES Verschlüsselung.“ (o. D.), Adresse: https://studyflix.de/informatik/aes-verschlusselung-1611 (besucht am 02. 01. 2024).

[56] Baivab Kumar Jena. „What Is AES Encryption and How Does It Work?“ (2023), Adresse: https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption (besucht am 02. 01. 2024).

[57] Wikipedia. „Advanced Encryption Standard.“ (2023), Adresse: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard (besucht am 05. 01. 2024).

[58] Prof. Norbert Pohlmann. „Was ist eine Cipher Block Chaining Mode oder CBC-Mode?“ (2022), Adresse: https://norbert-pohlmann.com/glossar-cyber-sicherheit/cipher-block-chaining-mode-cbc-mode/ (besucht am 05. 01. 2024).

[59] J. Schwenk, *Sicherheit und Kryptographie im Internet, Theorie und Praxis 5. Auflage*, de. Wiesbaden: Springer Vieweg, 2020, 519 S., ISBN: 978-3-658-29260-7. DOI: 10.1007/978-3-658-29260-7.

[60] seald. „3 common mistakes when implementing encryption.“ (2023), Adresse: https://www.seald.io/blog/3-common-mistakes-when-implementing-encryption (besucht am 03. 01. 2024).

[61] Oliver Helfrich, KOFAX. „30 Jahre PDF, Ein Geschenk, das uns immer wieder neu überrascht.“ (2023), Adresse: https://www.kofax.de/learn/blog/30-years-of-pdf (besucht am 19. 12. 2023).

[62] Promotional Blog. „Korrekturabzug.“ (o. D.), Adresse: http://www.promotional-blog.de/glossary/korrekturabzug/ (besucht am 30. 12. 2023).

[63] Kerstin Merber. „PDF Blätterkatalog erstellen und das hochwertige Ergebnis in wenigen Minuten sehen.“ (o. D.), Adresse: https://mypdf.me/blaetterbare-pdf/pdf-blaetterkatalog/ (besucht am 01. 01. 2024).

[64] Adobe Inc. „Dokumentenformate: Alles, was du wissen musst.“ (o. D.), Adresse: https://www.adobe.com/de/acrobat/resources/document-files.html (besucht am 20. 12. 2023).

[65] Mariko Kosaka. „Same-origin policy.“ (2018), Adresse: https://web.dev/articles/same-origin-policy (besucht am 08. 01. 2024).

[66] Nolan Lawson. „High-performance input handling on the web.“ (2019), Adresse: `https://nolanlawson.com/2019/08/11/high-performance-input-handling-on-the-web/` (besucht am 10. 01. 2024).

Tabellenverzeichnis

Abbildungsverzeichnis

Anhang

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Anmerkung: In einigen Studiengängen findet sich die Erklärung unmittelbar hinter dem Deckblatt der Arbeit.

[3cm] Köln, 04.03.2024 Unterschrift