

AFP mit PDF-Container mit secondary Ressourcen

GSE | GUIDE
SHARE
EUROPE

GSE Tagung – Application Enterprise Documents

Tino Rosner



Guido Schnepf

CREEDO

- Jedes Format hat seine individuellen Stärken als auch Schwächen
- Beide Formate haben große Gemeinsamkeiten, besonders auf technischer Basis:
- AFP als auch PDF-Datenströme ...
 - ... können gleichzeitig ein oder mehrere Dokumente respektive Sendungen enthalten
 - ... unterstützen globale Ressourcen, die für den gesamten Datenstrom gelten
 - ... unterstützen Inline-Ressourcen, die etwa
 - nur für eine Sendung gelten
 - nur auf einer Seite
 - nur in einem Bündel von Sendungen oder Seiten, ...
 - ... nutzen standardisierte Container für (druckreife) Fremdformate
 - ... unterstützen die Koexistenz von statischen als auch variablen Druckdaten
 - ...

- **Schriftarten**
 - Zeichensätze (Character Sets)
 - Zeichentabellen (Code Pages)
- Statische Seitenelemente, Formulare (Kopf- und Fußzeilen, **Overlays**, Underlays)
- **Bilder**, Logos
- (Vektor-) **Grafiken**
- **Fremdobjekte** jeder Art, die vom Zielsystem interpretiert werden kann (PDF)
- Steuerinformationen (PJL)
- Metadaten (ggfs., XMP, ...)

1. **Dokument** -> **Ressource** („Direkt“)

- Beispiel:
 - Dokument benötigt Schriftart XY zum Drucken von Text

2. **Dokument** -> Ressource -> **Ressource** („Secondary“)

- Beispiele:
 - Dokument enthält ein PDF, das eine Schriftart benötigt
 - Bild, das eine Color-Management-Ressource (ICC-Profil) referenziert

3. **Dokument** -> Ressource -> Ressource -> **Ressource** („Tertiary“)

- Beispiele:
 - Dokument enthält ein SVG, das ein (Raster-)Bild enthält, welches eine Color-Management-Ressource (ICC-Profil) referenziert
 - 2D-Barcodes auf der Seite mit eingebetteten Rasterbildern

Ressourcen können dem Dokument auf verschiedene Art präsentiert werden:

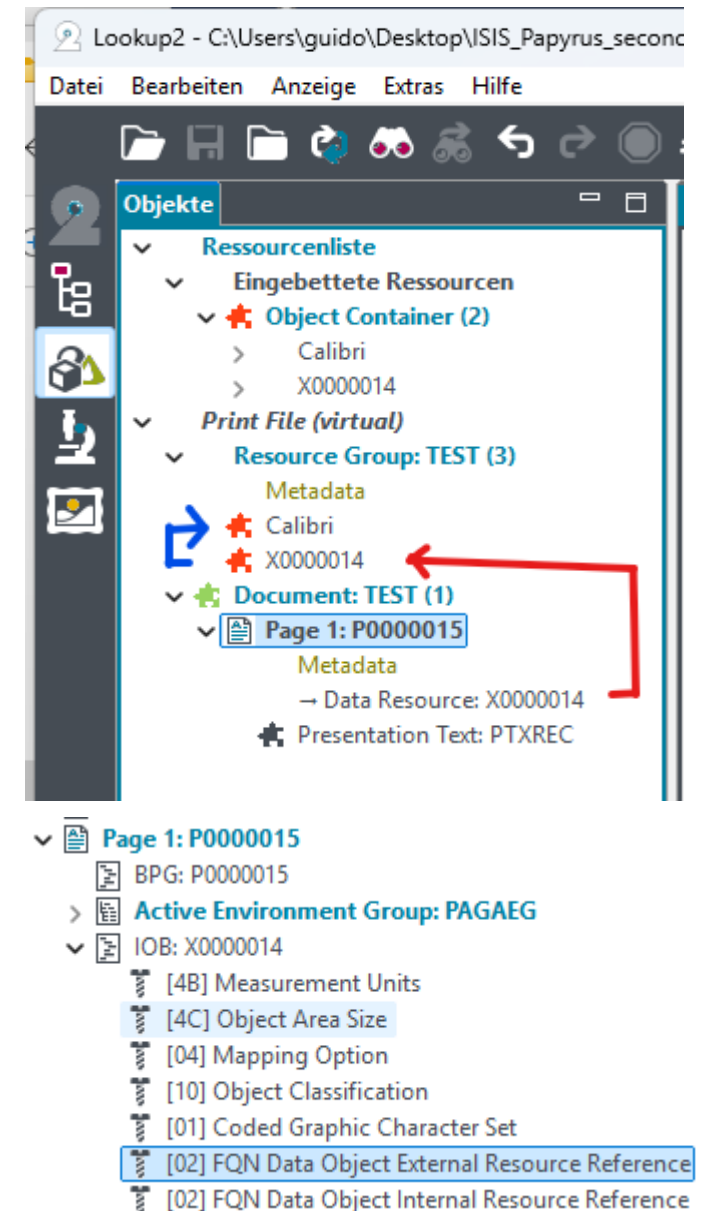
- **Extern in einer Library**, einem Verzeichnis, oder einem Archiv
- **Eingebettet** in den gleichen Datenstrom
- **Resident am Zielsystem**
 - Achtung: Dazu gehören natürlich auch die PDF Standard Fonts, die etwa dem Acrobat Reader, aber auch jedem anderen Anzeigeprogramm oder Drucker vorab vorliegen müssen!

- Datenströme enthalten nicht alle zur Verarbeitung benötigten Ressourcen 🤖
 - Dokumente oder Seiten verweisen auf falsche Ressourcen
 - Etwa veraltete Overlays, Fußzeilen, o.ä. 🤖
 - Ressourcen sind mehrfach im Datenstrom vorhanden,
 - unter gleichem Namen mit unterschiedlichem Inhalt 🤖
 - unter immer neuen Namen mit gleichem Inhalt 🤖
 - In Ressourcen eingebettete Zweitressourcen (Fonts in PDF-Containern etwa) blähen den Datenstrom unnötig auf, da vom Drucksystem nicht kontrollierbar. In der Folge steigen Speicherbedarf und CPU-Ressourcenverbrauch exponentiell. 🤖
- Eine aktive und frühzeitige Ressourcenverwaltung, ein **Management** möglichst bereits beim Generieren ist SEHR wünschenswert!
- Ressourcen sind alles andere als unwichtiges Beiwerk!

- Grundsätzlich benötigen alle Dokumente, **die nicht ausschließlich als bereits fertig gerastertes Bild vorliegen**, zusätzliche Ressourcen, um dargestellt werden zu können.
- Das Rendern, die Bildsynthese ist häufig ein Prozess erst unmittelbar vor der eigentlichen Darstellung. Das bedeutet, dass neben den Dokumenteninhalten bis zu diesem Zeitpunkt auch alle Ressourcen bereitstehen müssen.
- Das AFP-Consortium hat neben dem AFP-Druckdatenformat mit dem IPDS-Protokoll einen klar definierten Weg der Verwaltung und Bereitstellung von insbesondere externen Ressourcen geschaffen, die in anderen Formaten wie etwa PDF schlicht nicht existieren.
- In der Folge hat sich im Rahmen von PDF-Dateiformaten weitgehend durchgesetzt, alle im späteren Workflow genutzten Ressourcen frühzeitig in den Datenstrom einzubetten und auf diesem Weg bis zum Renderer zu transportieren.

Das AFP Consortium hat vor mindestens zehn Jahren eine Lösung unter dem Namen Secondary Resource Objects skizziert und in der MO:DCA-Spezifikation definiert:

- Das Einbetten (SPOC) oder Verlinken (IOB) von Ressourcen, etwa einem PDF funktioniert wie üblich.
- Diese Ressourcen enthalten jedoch **keine** eigenen Ressourcen!
- Stattdessen werden den Objekteigenschaften External Resource References und Internal Resource References mitgegeben, die der Umgebung ermöglichen, die externe Ressource wie üblich zu finden
- **ABER:** Bis heute hat leider kein Hersteller diese Spezifikation in eigenen Produkten realisiert. 🤖



PDF verwaltet Ressourcen ...

... eigentlich nicht. 🙄

Eine vernünftige Ressourcenverwaltung ist inhärent unbekannt, da das PDF-Format niemals mit dem Ziel einer effizienten Massendruckstrategie, überhaupt einem Bewusstsein für Speicher- oder CPU-Limitierungen angetreten ist.

PDF-Dateien enthalten darüber hinaus in aller Regel nur Einzelsendungen, (außerhalb unserer Branche) so gut wie nie mehr als ein Dokument gleichzeitig.

Erst in den letzten Jahren wurde mit PDF/VT („Variable Transaktion Data“) ein Unterstandard des für Druckprozesse relevante PDF/X („Exchange“) Standard entwickelt, der hier nachgelegt und im Wesentlichen bei der AFP-Spezifikation *abgeschrieben* hat.

Die Verbreitung von PDF/VT ist bis heute „homöopathisch“ zu nennen.

Wo es keine Regeln gibt ...

... gibt es allerdings auch keine Verbote!

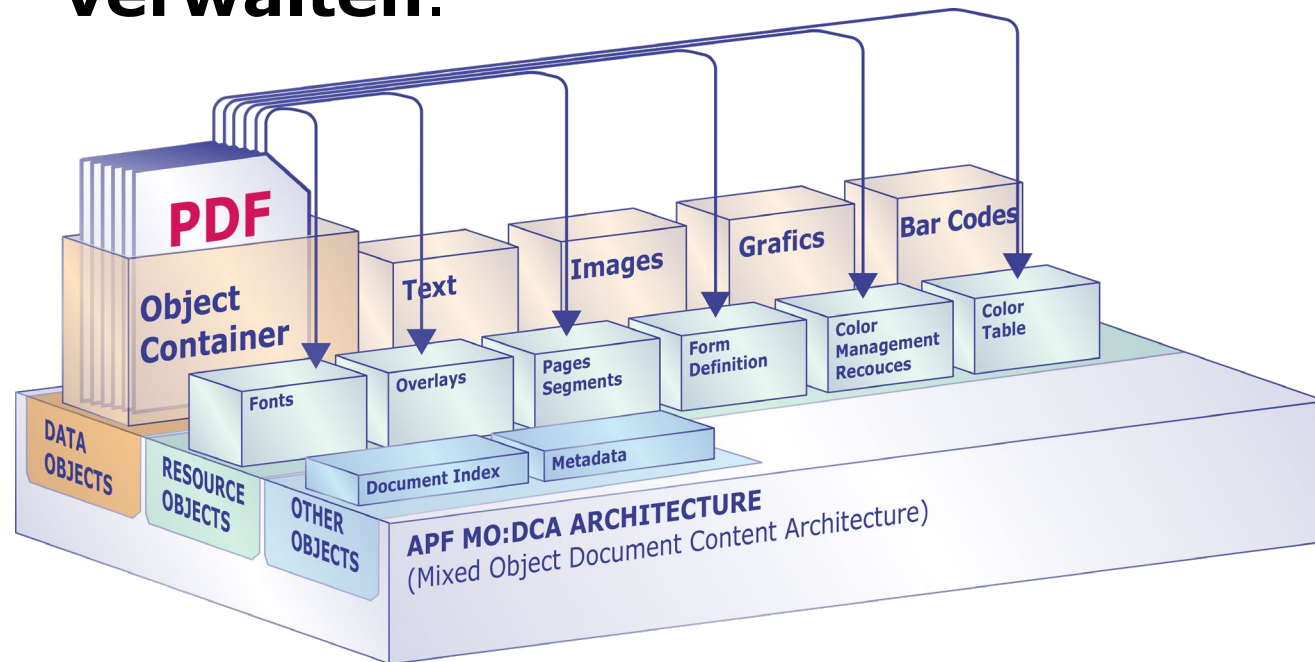
Die im Folgenden skizzierte Lösung stellt den derzeit wohl effizientesten Weg dar, Konflikte in Bezug auf überbordende Ressourcenmengen im Kontext von PDF und PDF-Containern im Rahmen von AFP-Druckdatenströmen zu lösen.

Warum entstehen große PDF-Dateien?

- Eingang: Kunden liefern oft PDF/A-Dateien mit eingebetteten Ressourcen → optimal für Archivierung.
- Ausgang (Druckdatenstrom): Wir brauchen möglichst kleine Dateien für schnelle Verarbeitung.

Variante 1:

Ressourcen (Bilder, Schriften) **aus PDF extrahieren und extern verwalten.**



Nachteile:

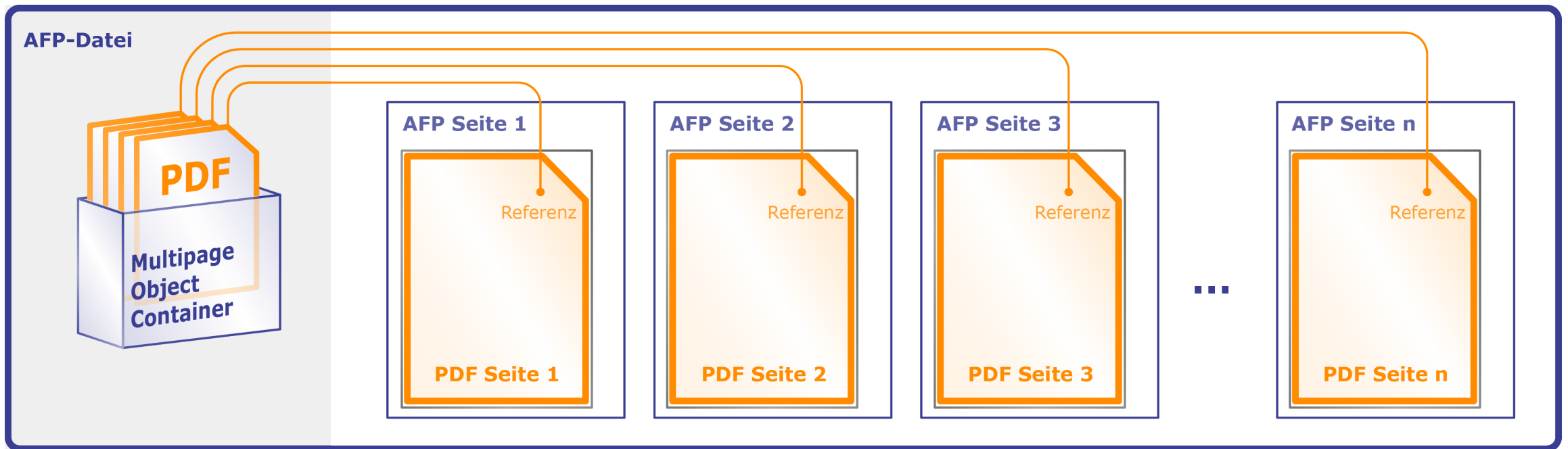
Verlust der Konsistenz & PDF-Semantik
Höherer Pflegeaufwand
Verlust der PDF-Vorschau-Vorteile
Abhängigkeit vom AFP-Rendering

Variante 2:

Optimierten PDF-Multipage-Container neu schreiben

– mit gezielter Ressourcen-Wiederverwendung.

Architektur



Transparenter PDF-Container

AFPLookup

Triplet: **x'10' (Object Classification)**
 Start-Byte: 342 (x'00 00 01 56')
 Länge: 24

Objektklasse: Zur Laufzeit feste Daten
 Die Objektdaten werden in einem MO:DCA-Container geliefert: Ja
 Der Objektcontainer enthält OEG-Structured-Fields: Ja
 Die Objektdaten befinden sich in OCD-Structured-Fields: Ja
 ASN.1 Object Identifier (OID): 06 07 2B 12 00 04 01 01 40
Registrierter Objekt-Typ: PDF Multiple-page File with Transparency

Lookup2

Object Classification Triplet
 Kürzel: [10]
 Fingerabdruck: 0x10
 Start-Byte: 394
 Start-Byte (Hex): 0x018A
 Länge (Bytes): 23

Object Class
 ObjClass: x'01' - Time-invariant paginated presentation object

Structure Flags
 StrucFlgs: 11-11-11-... Carried in object container, carried in OCD-SFs, with OEG

Registered ASN.1 Object Identifier
RegObjId: Multiple-Page PDF Object with Transparency

Hex-Ansicht
 [EBCDIC] 500 - International

00000145	34	D3	A8	CE	00	00	11	D6	ES	D3	F0	F0	F0	F0	F1	18	10	00	01	00	00	18	10	...
00000158	00	01	00	00	FC	00	06	07	2B	12	00	04	01	01	40	00	00	00	00	00	00	00	00	...
0000016b	00	00	00	0A	21	92	00	00	00	00	00	00	00	00	5A	00	28	D3	A8	92	...	...	...	
0000017e	00	00	12	D6	ES	D3	F0	F0	F0	F0	F1	18	10	00	01	00	00	FC	00	...	...	...	...	
00000191	06	07	2B	12	00	04	01	01	40	00	00	00	00	00	00	00	5A	00	08	...	...	...	...	
000001a4	D3	A8	C7	00	00	13	5A	00	1C	D3	A6	6B	00	00	14	03	43	01	08	...	...	...	...	
000001b7	4B	00	00	38	40	38	40	09	4C	02	00	2E	82	00	41	C6	5A	00	20	...	...	...	...	
000001ca	D3	AC	6B	00	00	15	01	17	00	00	00	00	00	00	00	00	2D	00	00	...	...	...	...	

In dem Structured Field "Begin Ressource Group" (BRS) gibt es das Triplet Object Classification, das letzte Byte der RegObjId, bzw. das 24. Byte des Triplets gibt Aufschluss über den Typen, z.B. 0E für TIFF, 3F für PDF, **40 für PDF mit Transparenz**

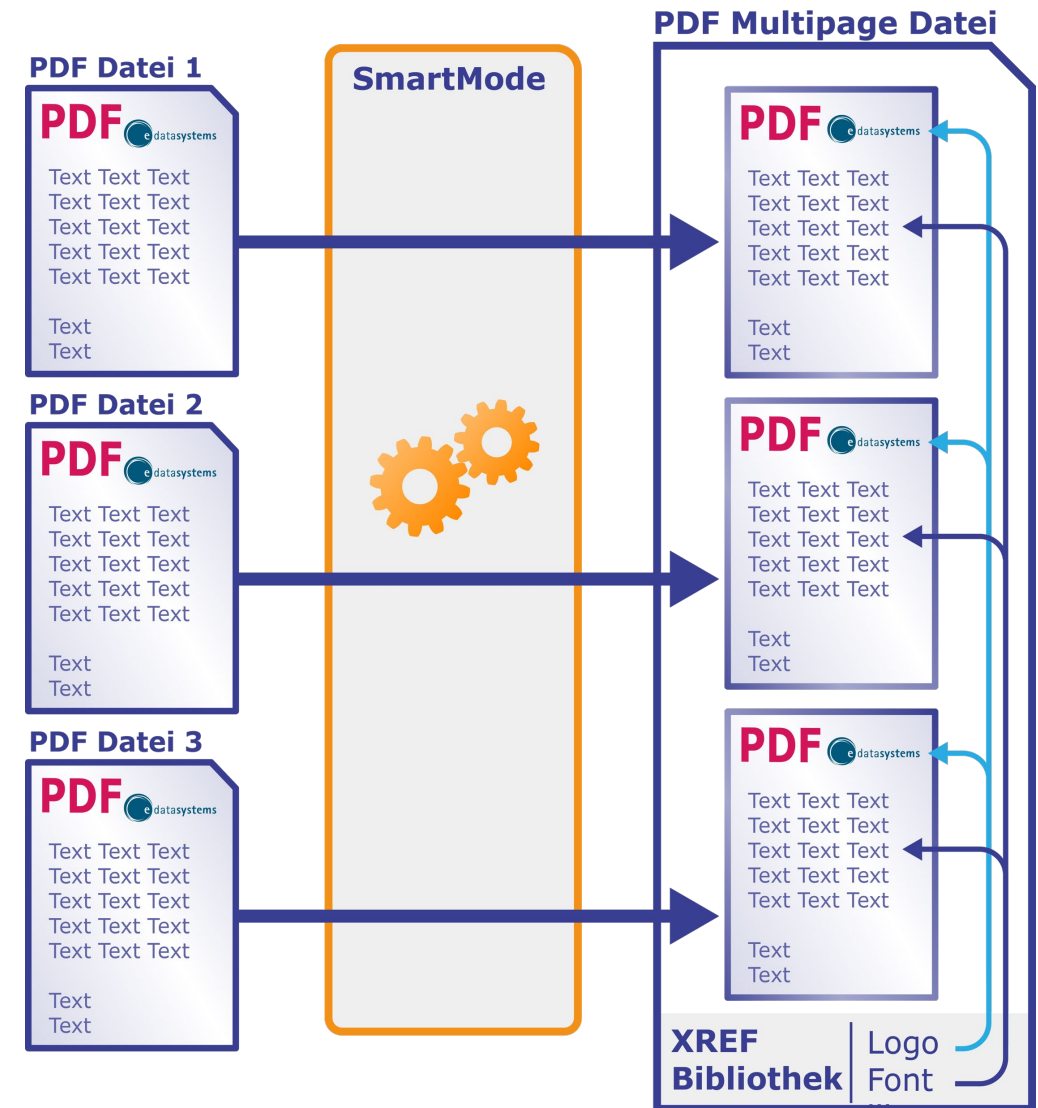
Neu schreiben: PDF-Container im Smart-Mode

Was ist der Smart Mode?

- Reduziert Dateigröße durch Wiederverwendung identischer Ressourcen
- Bilder, Schriftarten und Streams werden nur einmal eingebettet

Wie funktioniert das?

- Inhalte werden gehasht und verglichen
- Wiederverwendbare Objekte werden referenziert, nicht dupliziert



Erkennungsmechanismen:

1. Hash-Vergleich von Inhalten
2. Wiederverwendung von Fonts
3. Identische Bilder nur einmal speichern
4. Verwendung von PdfIndirectObject
5. Optimierung von PDF-Streams

Vorteile:

- Kleinere PDF-Dateien
- Weniger Speicherverbrauch
- Schnellere Verarbeitung

Aktivierung (Java) unter Verwendung von iText:

```
WriterProperties properties = new WriterProperties().useSmartMode();  
PdfWriter writer = new PdfWriter("output.pdf", properties);
```

Vorteile der PDF-Einbettung in AFP

✓ Qualitätssteigerung

- PDF bleibt **1:1 unverändert erhalten** – Farben, Transparenzen, Schriften bleiben wie im Original.

✓ Effizienzgewinn

- Kein Nachbauen von PDF-Layouts in AFP → **Schnellere Umsetzung neuer Dokumente.**

✓ Kostensenkung

- **Weniger Fehler durch direkte PDF-Nutzung**, weniger Aufwand für IT und Druckabteilung.

✓ Multichannel-Fähigkeit

- Gleiches PDF kann für Druck, **E-Mail-Versand und Archivierung** verwendet werden.

✓ Produktionssicherheit

- AFP behält **IPDS-Tracking & Stückkontrolle** → Jede Seite kann exakt nachverfolgt werden.

Praxisbeispiele

Beispiel 1: Einfache und schnelle Kundenintegration

Druckdienstleister integriert neue Kunden in zentrale Produktion

- ➡ Früher: Aufwendige PDF-Integration da prinzipiell AFP-Workflow.
- ➡ Jetzt: Kundendokumente als PDF geliefert & im AFP-Datenstrom eingebettet.
- ✅ Ergebnis: Schnelle und sichere Integration inklusive aller Features.
- ✅ Vorteil: Integration von Einzel-PDFs auf Sendungsbasis ohne Systemüberlastung

Beispiel 2: Automatische Beilagenverwaltung

Ein Versicherer musste gesetzliche AGB und Flyer in Sendungen einfügen.

- ➡ Lösung: Einfach entweder als digitale oder als physische Beilage konfigurieren → bei digitaler Beilage sogar individuelle Beilage je Kunde möglich.
- ✅ Ergebnis: Weniger Entwicklungsaufwand, einfachere Dokumentenpflege, schneller Austausch.
- ✅ Vorteile: Flexible Individualisierung, ohne Original-PDFs zu ändern

Best Practices

- **AFP mit PDF-Multipage-Container immer nutzen:**
Workflow benutzt AFP mit PDF-Multipage-Container als einzige Standardverarbeitung
- **Optimierte PDF-Multipage-Datei Erstellung:**
Überflüssige Elemente reduzieren, Schriften immer einbetten.
- **PDF-Preflights & Monitoring:**
Automatisierte PDF-Preflight-Checks durchführen, um Druckqualität sicherzustellen.
- **IT & Fachabteilungen vernetzen:**
Klare Standards für PDF-Erstellung definieren.

Fazit: Das Beste aus beiden Welten nutzen

- **PDF für maximale Flexibilität und Qualität – weil unsere Kunden es nutzen.**
- **AFP für Produktionssicherheit und Steuerung – weil es millionenfache Druckprozesse zuverlässig managt.**

**Die Kombination aus AFP und PDF ist keine Übergangslösung
– sie ist die Zukunft des Hochvolumendrucks.**

Danke für Ihre Aufmerksamkeit!

Kontakt:

Tino Rosner ©
edatasystems GmbH

tino.rosner@edatasystems.de
Mobil: 0151 14842115

Guido Schnepf ©
CRE-DO GmbH

guido.schnepf@cre-do.de
T +49 (2102) 305 818-0