



FLAM®



limes datentechnik® gmbh
Philipp-Reis-Passage 2
61381 Friedrichsdorf
06172-59190
info@flam.de
www.flam.de



FLAM®

- Was ist FLAM?
- Wozu dient FLAM?
- Einsatzbeispiele
- Wie macht FLAM das?



Was ist FLAM ?

FLAM ist ein Software Werkzeug zur

- Verschlüsselung
- Komprimierung
- Erleichterung des heterogenen Datenaustauschs

FLAM = Frankenstein-Limes-Access-Method

Was ist FLAM ?

FLAM ist eine Zugriffsmethode für komprimierte und verschlüsselte Daten

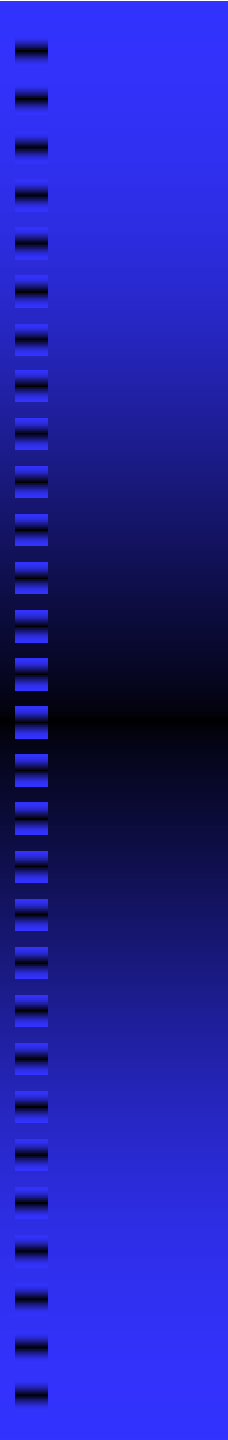
- Dateiebene
- Satzebene
- Direktzugriff mit Schlüssel des Originalsatzes
- Subsystem

FLAM = Frankenstein-Limes-Access-Method

Was ist FLAM ?

FLAM wurde 1986 für den Mainframebereich entwickelt,
es ist mittlerweile verfügbar für

- FSC BS2000/OSD
 Sinix
 Reliant Unix
- HP HPUX
 Windows
 OpenVMS (DEC)
 True64 UNIX (DEC)
 Non Stop OS (Tandem)
 OSS (Tandem)

- 
- IBM z/OS, OS/390, MVS, MVS-Subsystem
Linux (S/390, z-Series)
VM, VSE
OS400
AIX
 - NCR Unix
 - SCO SCO-Open Server
 - SUN SOLARIS
 - PCs Windows (9x, NT, 200x, XP, Server)
Linux

Der Hersteller

- limes datentechnik gmbh ist der Hersteller von FLAM
- Gegründet 1986 in Friedrichsdorf (15 km von Frankfurt/Main).
- Eigenes Entwicklungsrechenzentrum für Weiterentwicklung und Pflege.
- Zu allen IT-Herstellern besteht ein enges Vertragsverhältnis.
- Für die internationalen Vertragskunden: 7 x 24 Stunden Hotline.
- Alle Eigentums- und Vertriebsrechte liegen ausschließlich bei limes datentechnik gmbh.
- Gewähr, daß alle Anforderungen in die Entwicklung einfließen.
- Abwärtskompatibilität bleibt erhalten.

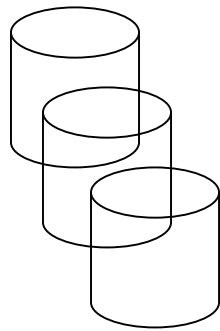
Was ist FLAM ?

FLAM ist Industriestandard

- im Zahlungsverkehr (deutsches Kreditwesen)
- im Meldewesen (deutsches Bank- und Börsenwesen)
- im Mobilfunk (europäisches Clearing von Gebührendaten)
- bei der Gebührennachverarbeitung (Deutsche Telekom AG)
- GATEWAY der Deutsche Bahn AG (Transportdaten)
- bei ePOST (Deutsche Post AG)

Funktionsschema für FLAM

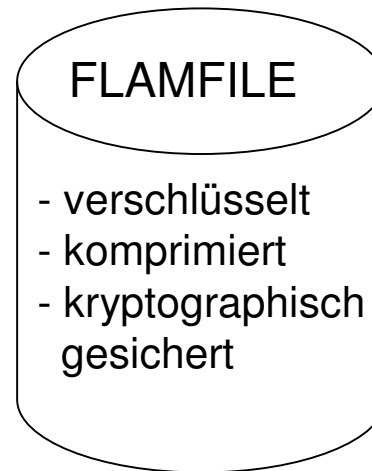
plattformspezifisch



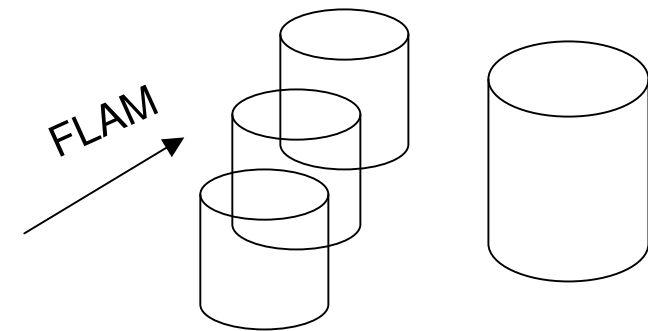
input Dateien in der beim
Anwender vorhandenen
Form

Sequentiell
Indexsequentiell
Undefiniert
Textdateien
VSAM-Dateien
Bibliotheken

neutral



plattformspezifisch



output Dateien in der vom
Anwender gewünschten
Form

Sequentiell
Indexsequentiell
Undefiniert
Textdateien
VSAM-Dateien
Bibliotheken

Der input wird von FLAM neutralisiert, komprimiert, verschlüsselt und gesichert in eine FLAMFILE® geschrieben. Diese ist frei formatierbar (plattformspezifisch), der Inhalt aber neutral. Bei Ausgabe können die Daten in eine oder mehrere (plattformspezifische) Dateien mit anderem Zeichensatz konvertiert werden.



FLAM®

- ✓ Was ist FLAM?
- Wozu dient FLAM?
 - Einsatzbeispiele
 - Wie macht FLAM das?

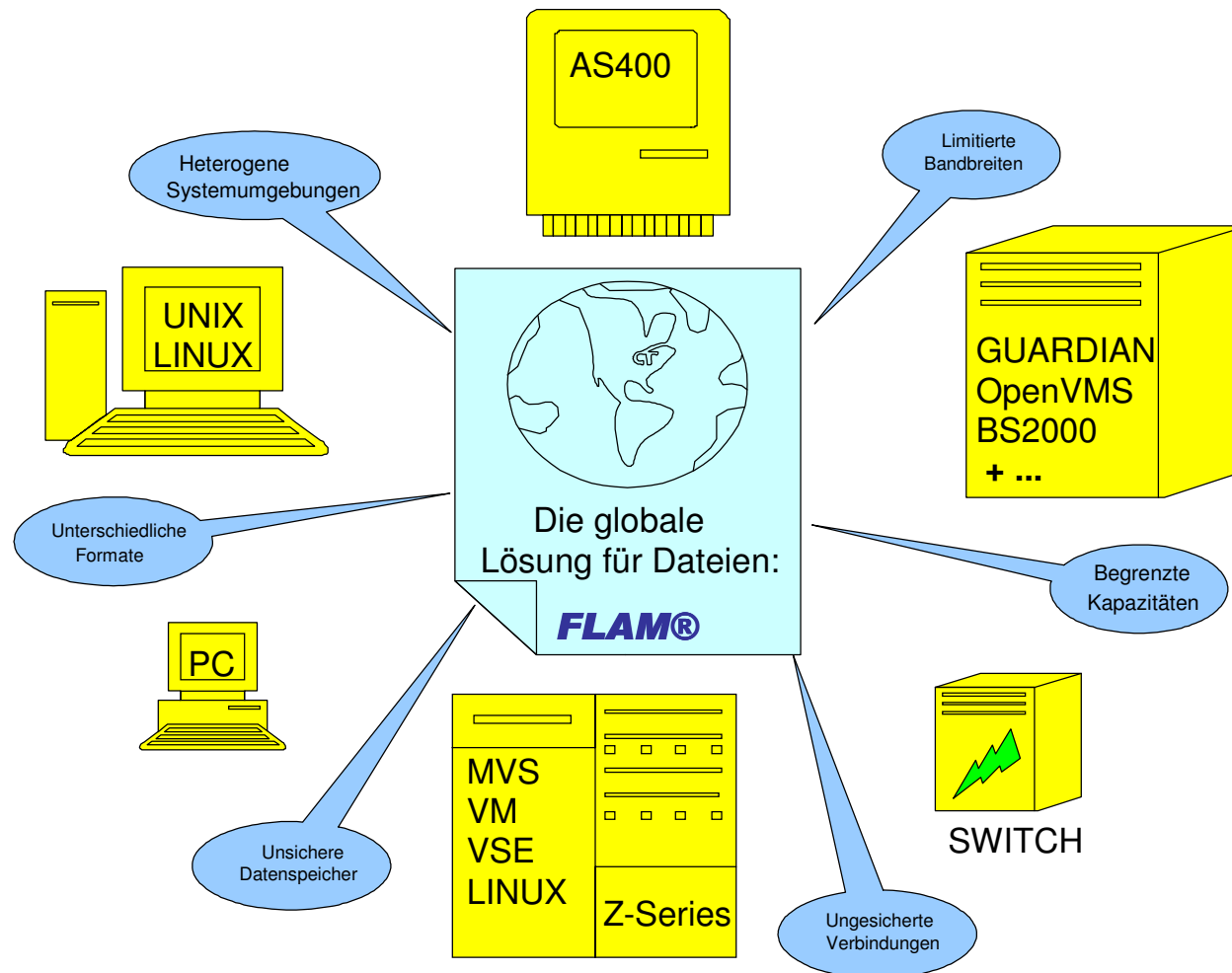


Wozu dient FLAM ?

- FLAM ist ein Software Werkzeug zur
 - ◆ Verschlüsselung
 - ◆ Komprimierung
 - ◆ Erleichterung des heterogenen Datenaustauschs

- FLAM ist eine Zugriffsmethode für komprimierte und verschlüsselte Daten
 - ◆ Dateiebene
 - ◆ Satzebene
 - ◆ Direktzugriff mit Schlüssel des Originalsatzes
 - ◆ Subsystem

Überall FLAM®





Wozu dient FLAM ?

FLAM wird überall da verwendet, wo es auf

- Datenaustausch
- Datensicherheit
- Wirtschaftlichkeit

ankommt.



Datenaustausch

- Unterstützung aller betriebssystem-spezifischer Dateiformate
- Erstellen der Ausgabedateien unabhängig vom originalen Format möglich
- Unterstützung ‚fremder‘ Dateiformate auf Windows und Unix Systemen
- FLAMFILE als neutrale, komprimierte und verschlüsselte ‚Austauschdatei‘
- Transport über beliebige, auch ungesicherte Leitungen (z.B. Internet)



Datenaustausch

- Zusammenfassung vieler (kleiner) Dateien in eine FLAMFILE
- ‚Zusammengefasste‘ Dateien können in eine Datei ausgegeben werden
- Konkatinierung von FLAMFILEs
(Aneinanderfügen mehrerer Dateien)
- Zeichensatz Umsetzung (ASCII / EBCDIC)



Datensicherheit

- FLAM stellt Vollständigkeit und Unversehrtheit sicher
- FLAM unterstützt AES (Advanced Encryption Standard). AES ist der aktuelle Weltstandard (u.a. seit Mai 2002 der US-Behörden-Standard).
- FLAM läßt User-Key von bis zu 64 Bytes ($64 \times 8 = 512$ Bits) zu.
- FLAM erschwert durch seine Architektur „brute force attacks“ und Krypto-Analyse.
- FLAM ermöglicht neue Sicherheitsverfahren z.B. durch Parallel-Splitting.



Wirtschaftlichkeit

- FLAM garantiert einen geringen Einführungsaufwand in die Produktion.
- FLAM hat einen auffällig geringen Ressourcenverbrauch.
- FLAM erleichtert Anwendergruppen die Entscheidung für solche IT-Prozesse.
- FLAM gewährleistet einen schnellen ROI.
- FLAM sichert Daten mit dem besten Preis-/Leistungsverhältnis.



FLAM®

- ✓ Was ist FLAM?
- ✓ Wozu dient FLAM?
- Einsatzbeispiele
 - Wie macht FLAM das?



Einsatzbeispiele von FLAM

- Bank Account Clearing
Im Zahlungsverkehr zwischen Bankenorganisationen untereinander und mit den Zentralbanken ist FLAM Standard.
- Zahlungsverkehr zwischen Banken und deren Kunden
Alle Bankenverbände haben von limes datentechnik gmbh das Recht erworben, FLAM in die Softwareprodukte im Zahlungsverkehr mit ihren Geschäftskunden einzubinden.
- Billing Record Clearing
Da viele Telco's und die Switchhersteller (z.B. SIEMENS, ALCATEL und ERICSSON) sich auf den FLAM-Standard geeinigt haben, findet Billing Record Clearing auf allen fünf Kontinenten auf Basis des Produktes FLAM statt. Dabei kommen neben den Switches alle Rechnersysteme und Plattformen im nationalen und internationalen Clearing zum Einsatz.



Einsatzbeispiele von FLAM

- Geldinstitut schickt Daten verschlüsselt zum "Kreditkartenabrechner" (RJE).
- Krankenkasse liefert Versichertendaten auf verschlüsselter DVD zum Dienstleister.
- Dezentral gescannte vertrauliche Daten werden mit FLAM verschlüsselt und übertragen.



Einsatzbeispiele von FLAM

■ Langzeitarchivierung

Verschiedene Anwender bevorzugen FLAMFILEs für die Langzeitarchivierung, weil

- durch die gewährte Kompatibilität auch in Zukunft von neuen Plattformen die alten Formate bearbeitet werden können;
- auch bei Wechsel von Betriebssystemen oder Hardwareplattformen die ursprünglichen Formate der komprimierten Daten durch einfache Dekomprimierung der neuen Systemumgebung angepasst werden können.



Einsatzbeispiele von FLAM

- Performancegewinn durch Zusammenfassen von Mini-Dateien (Kassenautomaten / Paketaufkleber)

Ein Transportunternehmen sichert in Fahrkartenautomaten in viertelstündlichen Abständen Geldkartenumsätze. Die daraus resultierende große Anzahl von Minidateien werden mit FLAM zu einer Datei zusammen gefasst und einem Dienstleister zur Abrechnung zugeschickt. Dort entsteht durch FLAM nur eine (große) Datei. Durch diese Technik wird sowohl bei der Übertragung als auch bei der Verarbeitung ein Performancegewinn erzielt.



Einsatzbeispiele von FLAM

- Versenden von Druckdateien

Ein internationales Geldinstitut verschickt komprimierte und verschlüsselte Druckdaten in Abhängigkeit von vorhandenen Druckkapazitäten und regionalen Anforderungen an dezentrale Druckzentren. Dabei wird die Fähigkeit von FLAM genutzt, ein bestimmtes Datenformat für das Spoolingsystem erzeugen zu können.



Einsatzbeispiele von FLAM

- Kostengünstiges Retrieval (Rechnungen Telekom, Schecks bei Bundesbank, trotz hoher Komprimierung schneller Zugriff)

Mehr als 3 TB Daten in VSAM-KSDS FLAMFILEs im Direktzugriff. Bis zu 245 x FLAM in einem Lauf.

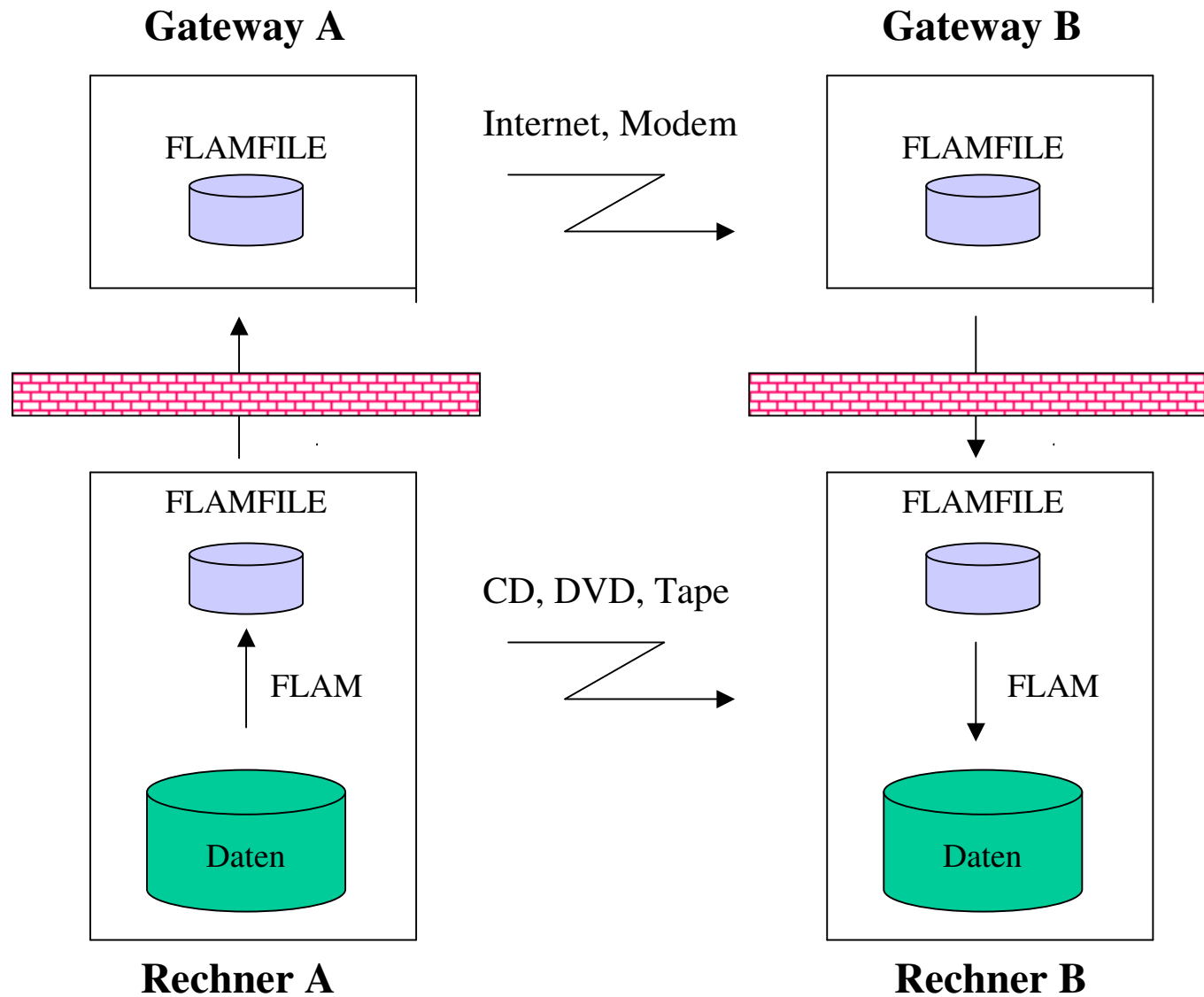
Komprimierte Daten auf Magnetbandkassetten werden sekundenschnell durchsucht.



Einsatzbeispiele von FLAM

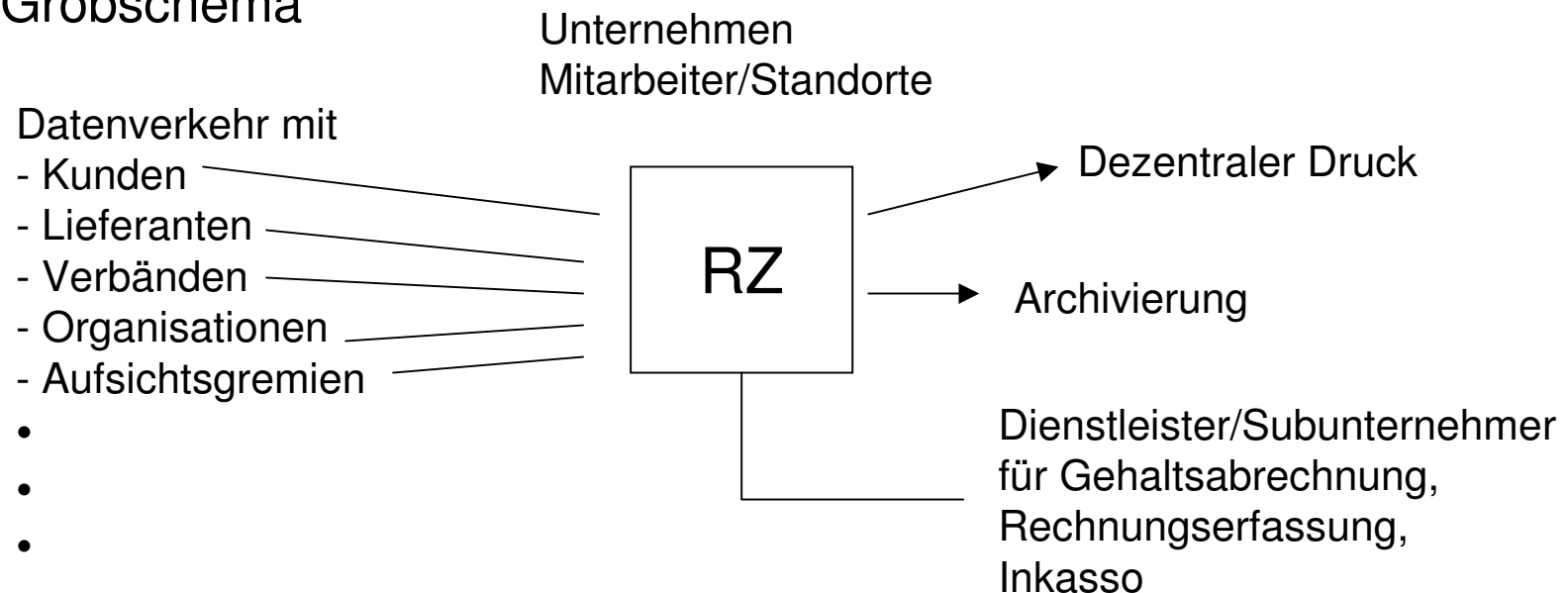
- Softwareversand/Software-Maintenance
 - ◆ Großer IT-Anbieter liefert individuell lizenzierte Software an Endkunden.
 - ◆ Ausliefern von Software-Updates an Kunden.
 - ◆ Auch im Fehlerfall Sicherheit gewährleisten, z.B. Dump und Daten verschlüsselt zur Wartung und Diagnose versenden.

Einsatzbeispiele von FLAM



Analyse der Datenströme, um Einsatzfälle für FLAM zu finden

Grobschema



Wo sind vertrauliche, schützenswürdige Daten, die bisher nicht "state of the art" behandelt werden? (Bänder, CD's, online, Papierform)

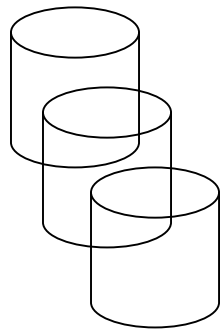


FLAM®

- ✓ Was ist FLAM?
- ✓ Wozu dient FLAM?
- ✓ Einsatzbeispiele
- Wie macht FLAM das?

Funktionsschema für FLAM

plattformspezifisch



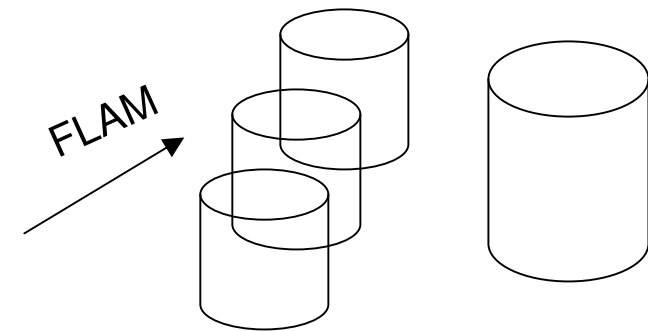
input Dateien in der beim Anwender vorhandenen Form

Sequentiell
Indexsequentiell
Undefiniert
Textdateien
VSAM-Dateien
Bibliotheken

neutral



plattformspezifisch



output Dateien in der vom Anwender gewünschten Form

Sequentiell
Indexsequentiell
Undefiniert
Textdateien
VSAM-Dateien
Bibliotheken

Der input wird von FLAM neutralisiert, komprimiert, verschlüsselt und gesichert in eine FLAMFILE® geschrieben. Diese ist frei formatierbar (plattformspezifisch), der Inhalt aber neutral. Bei Ausgabe können die Daten in eine oder mehrere (plattformspezifische) Dateien mit anderem Zeichensatz konvertiert werden.



Wie macht FLAM das?

- FLAM kennt den **Satzbegriff**
- Liest und schreibt **Datensätze**
- Zugriff auf Dateien erfolgt ‚**logisch**‘
- Fasst mehrere Sätze zu einer komprimierten und verschlüsselten Einheit zusammen (**Segment**)
- Auf diese Segmente kann **autark** zugegriffen werden
- Neutrale Datenablage mit spezieller Syntax (**FLAMFILE**)

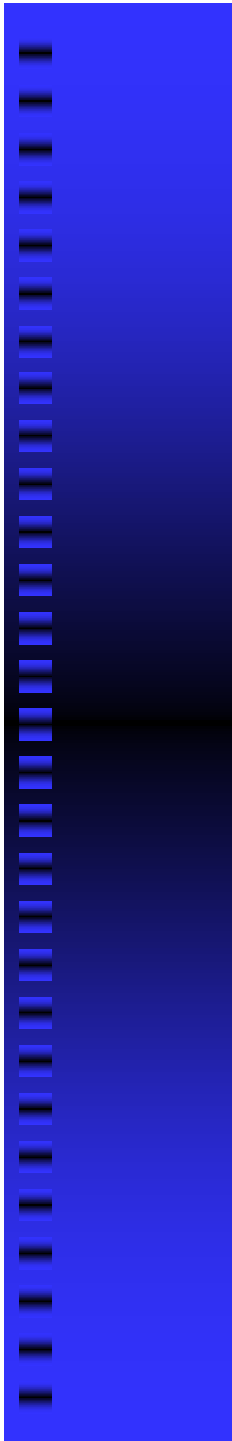
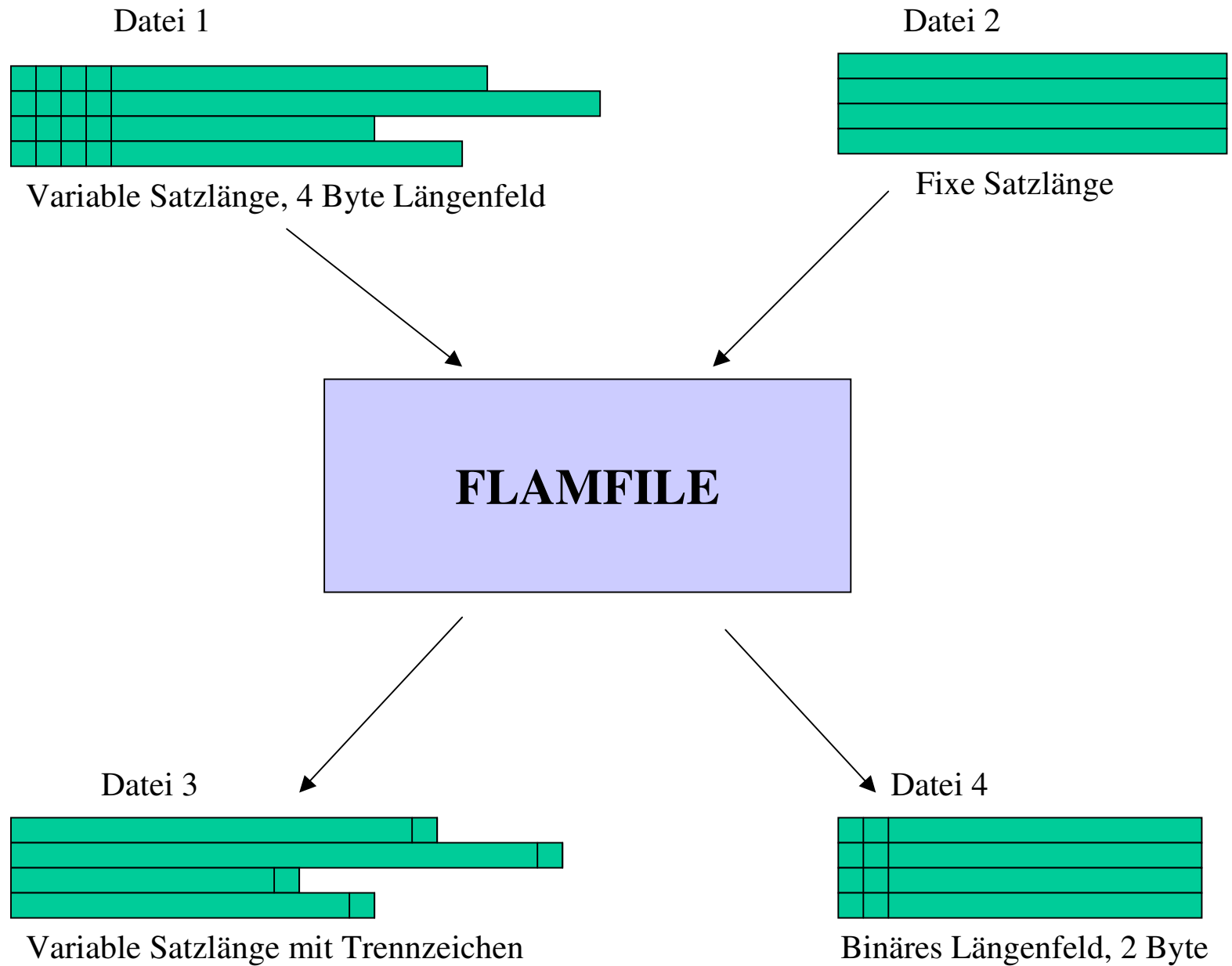


Wie macht FLAM das?

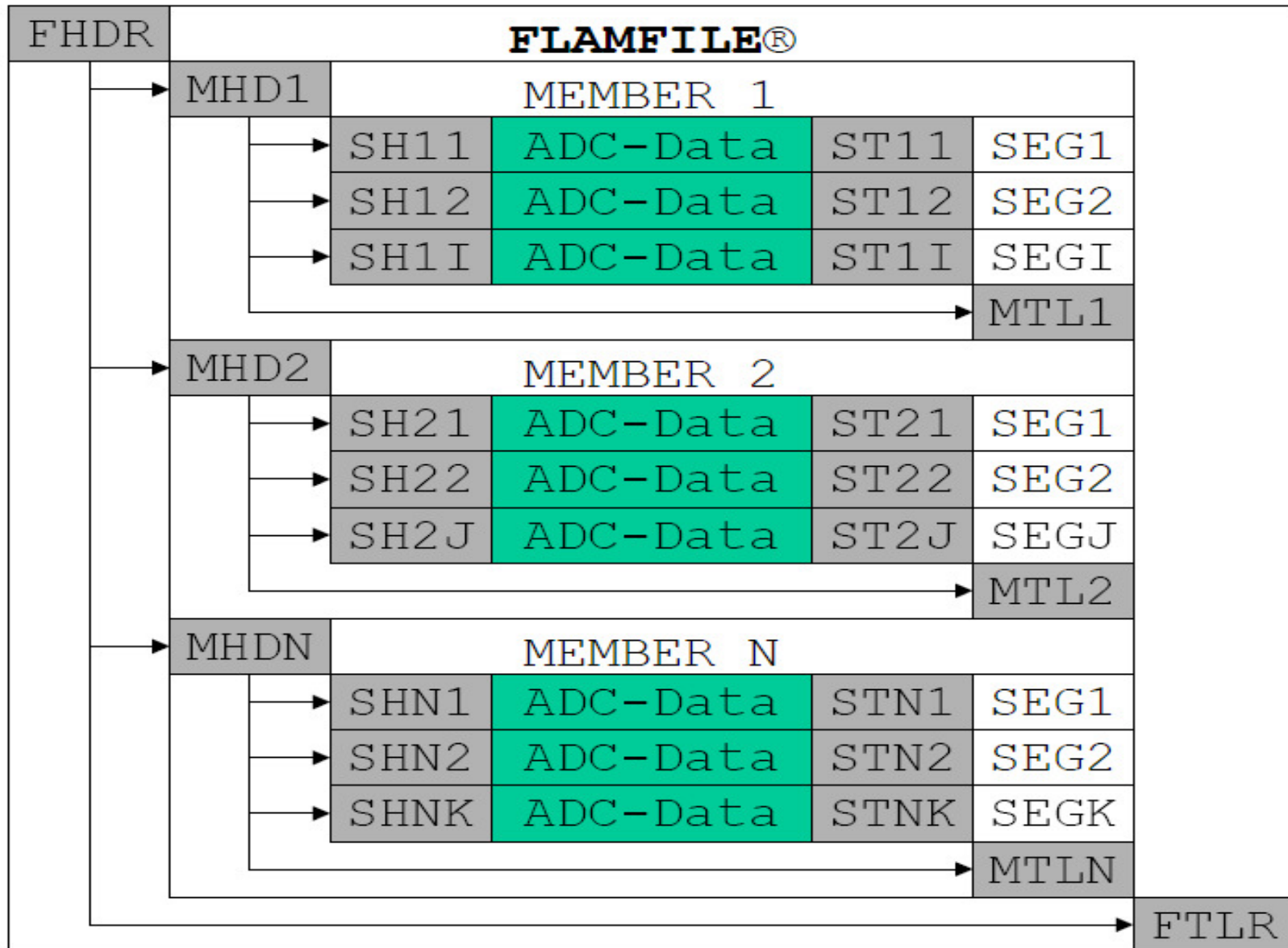
FLAMFILE

- Beliebige Datei- und Satzformat
- Satzlänge von 80 bis 32760 Byte möglich
- Vollständigkeit sichergestellt
- Manipulationssicher (AES-MACs)
- Ist ein Unikat (gleiche Daten im zweiten Lauf führen sichtbar zu einem anderen Inhalt)

Konvertierung von Satzformaten



Aufbau einer FLAMFILE





Funktionsbeispiele

- Wechsel von Plattformen
- FLAM® und FTP
- FLAM® und RJE
- Mehrere Dateien
- Mehrere FLAMFILE's
- Serielles Splitten von FLAMFILE's
- Paralleles Splitten
- Legendieren
- Programmversand/Wartung
- MVS-Subsystem

Wechsel von Plattformen

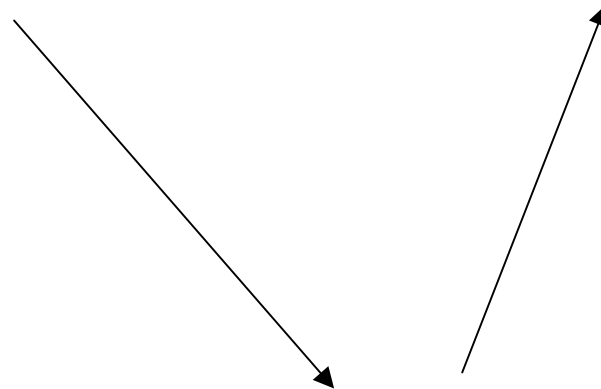
Problemstellung: unterschiedliche Dateiformate,
Zeichensätze

■ MVS

- ◆ PS, PO, VSAM
- ◆ Variabel, fix, undefined
- ◆ EBCDIC

■ UNIX

- ◆ Textdateien
(Satztrenner X'0A')
- ◆ Binärdaten
- ◆ Daten mit speziellen
Längefeldern (2 Byte, 4
Byte, MVS-Längefeld,
...)
- ◆ ASCII



FLAMFILE®

Frei formatierbar für Transportzwecke

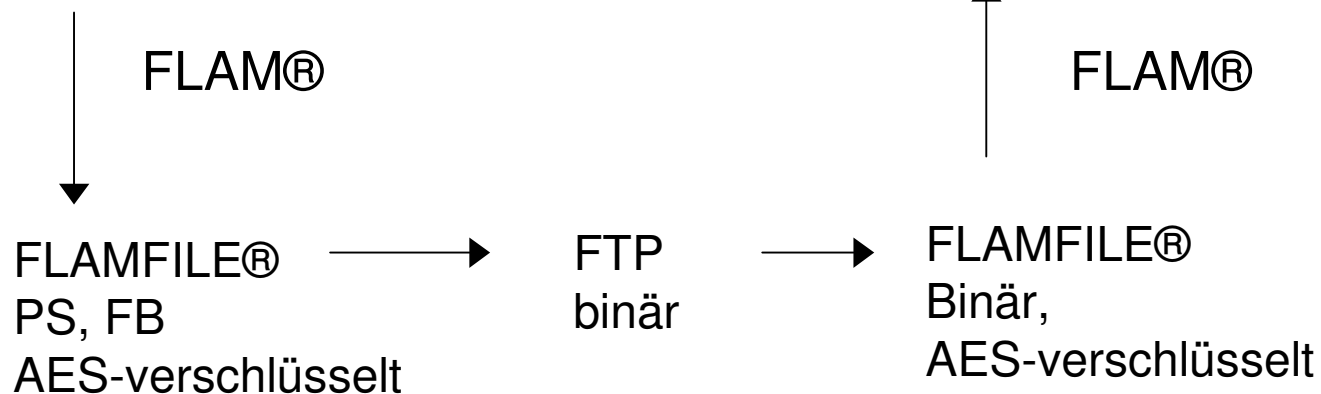
Funktionsbeispiel FLAM® und FTP

■ Problemstellung

- ◆ FTP beherrscht nicht alle Datei- und Satzformate
- ◆ Vollständigkeit nicht sicher
- ◆ Öffentliches Netz

MVS

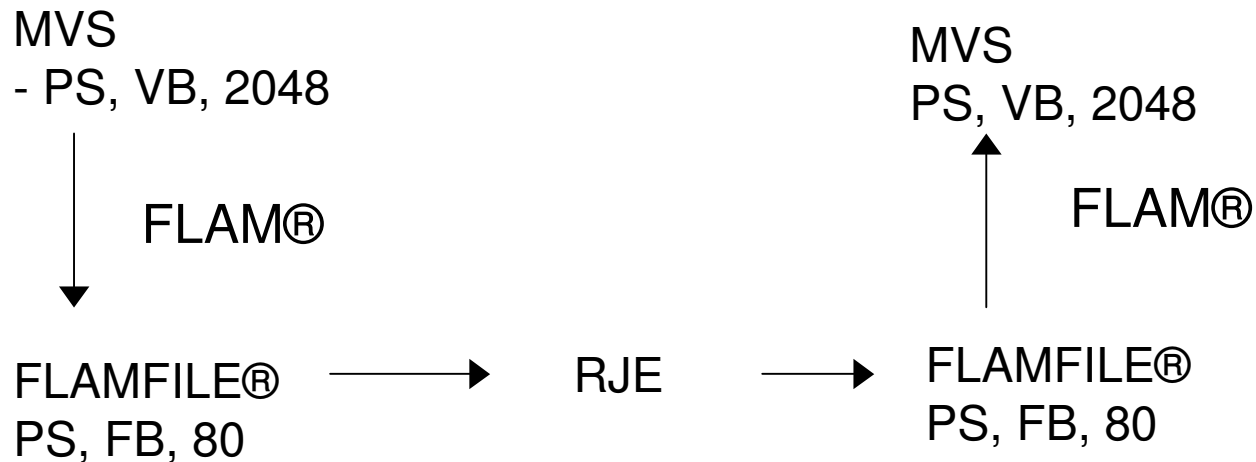
- PS, VB, binär
- VSAM-KSDS



Funktionsbeispiel FLAM® und RJE

■ Problemstellung

- ◆ RJE benötigt eine fixe Satzlänge von 80 Byte



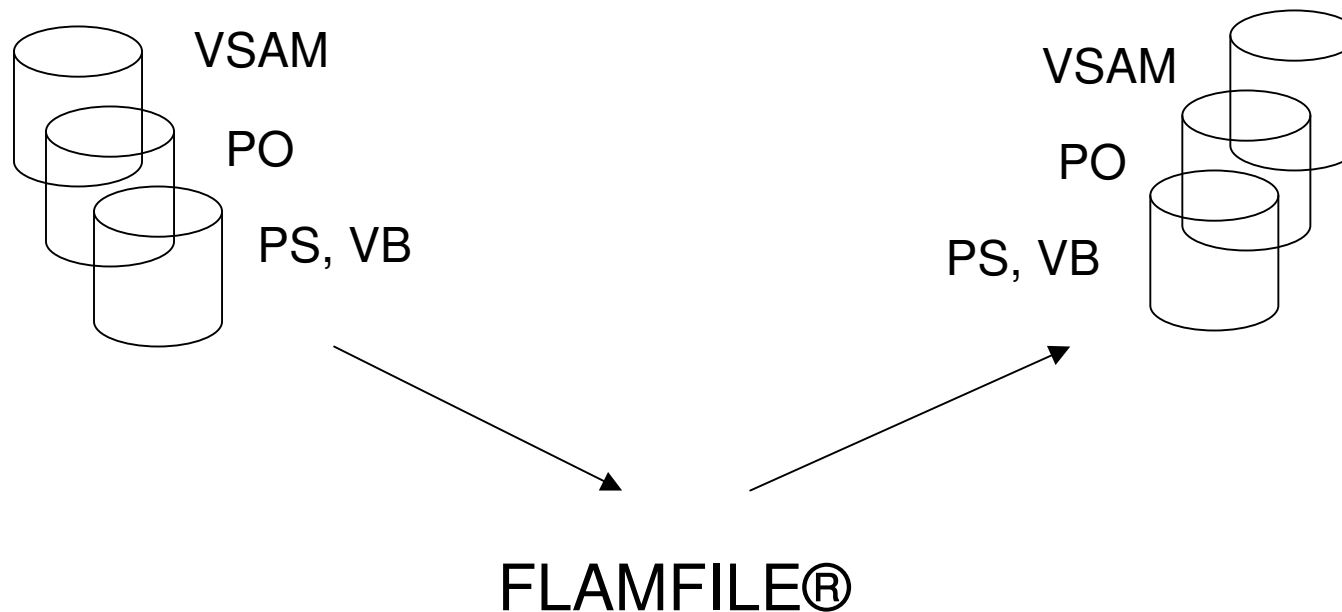
RJE steht hier nur als Beispiel für Transferprogramme mit speziellen Randbedingungen.

Funktionsbeispiel

Mehrere Dateien

■ Problemstellung

- ◆ Bei Übertragung: Overhead in Transportprotokollen, Handling, unterschiedliche Dateiformate

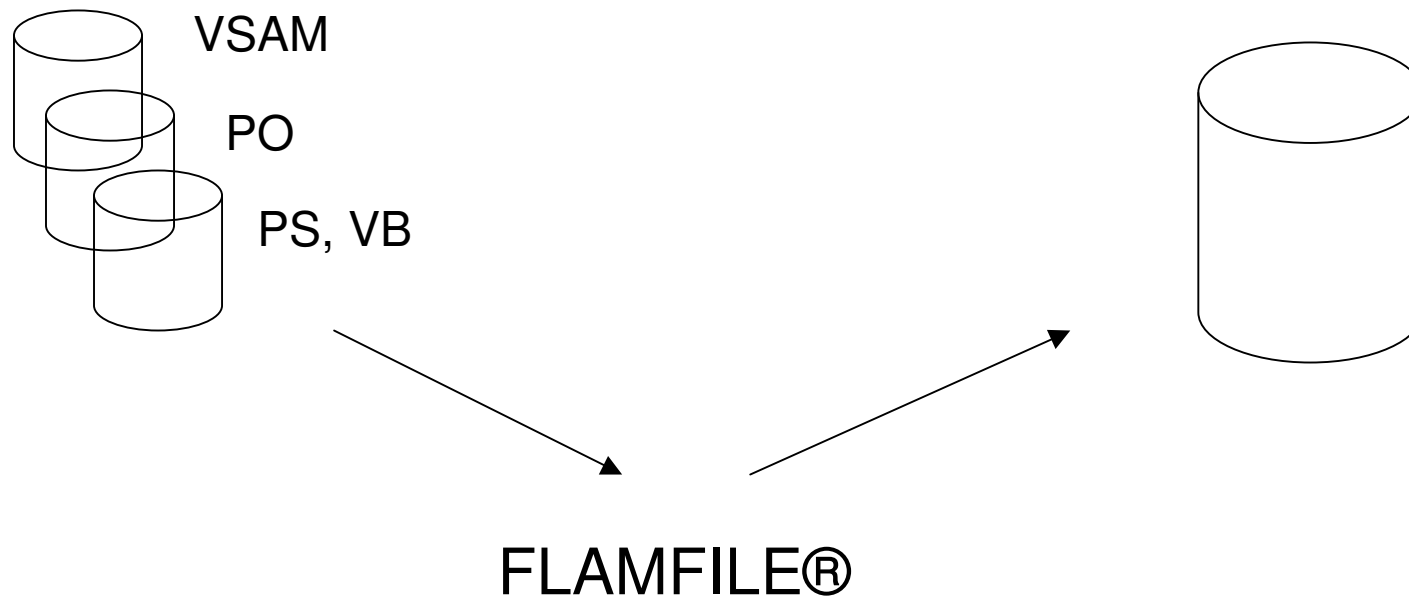


Funktionsbeispiel

Mehrere Dateien

■ Problemstellung

- ◆ Viele Einzeldateien, bei Verarbeitung wird nur eine benötigt

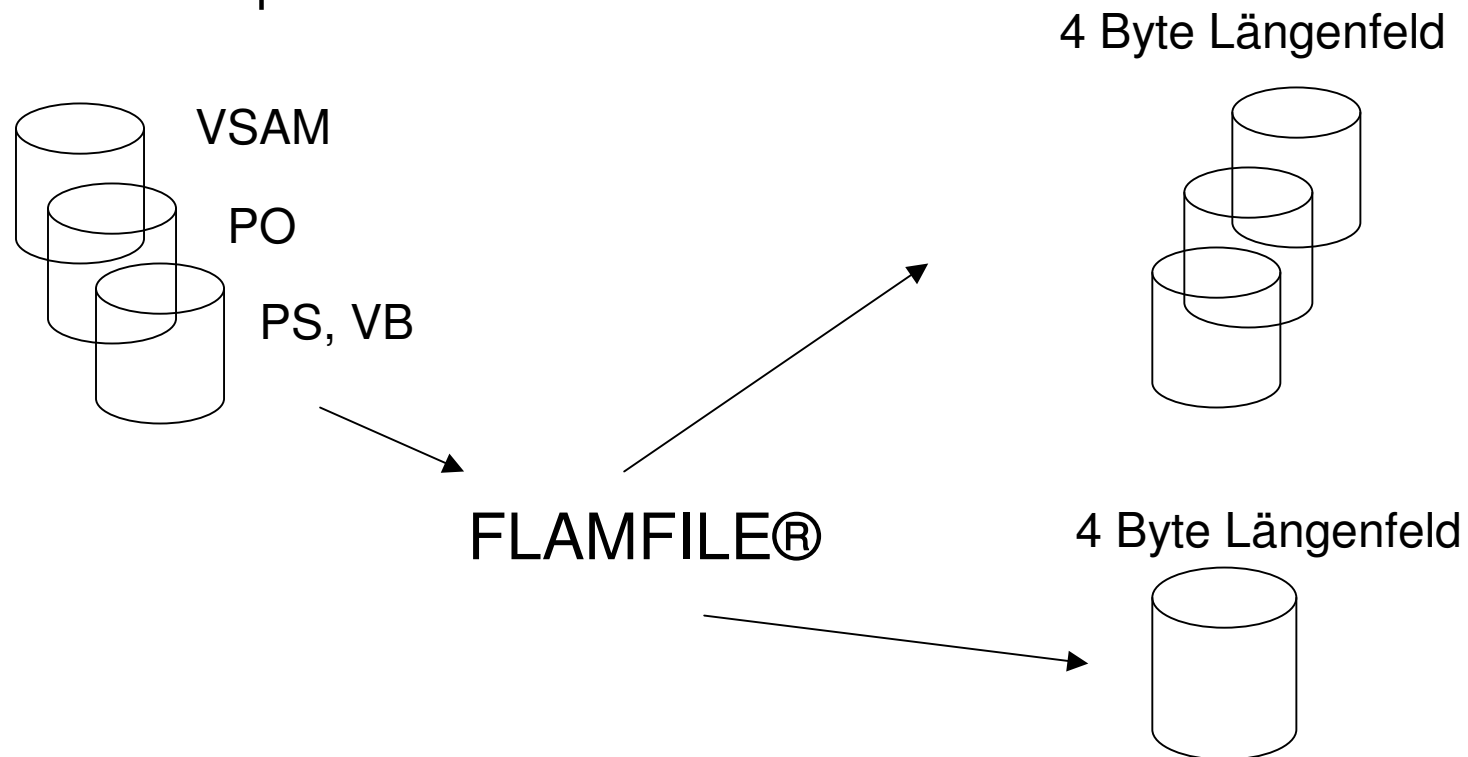


Funktionsbeispiel

Mehrere Dateien

■ Problemstellung

- ◆ Viele Einzeldateien unterschiedlichen Formats, benötigt wird ein spezielles Format für alle

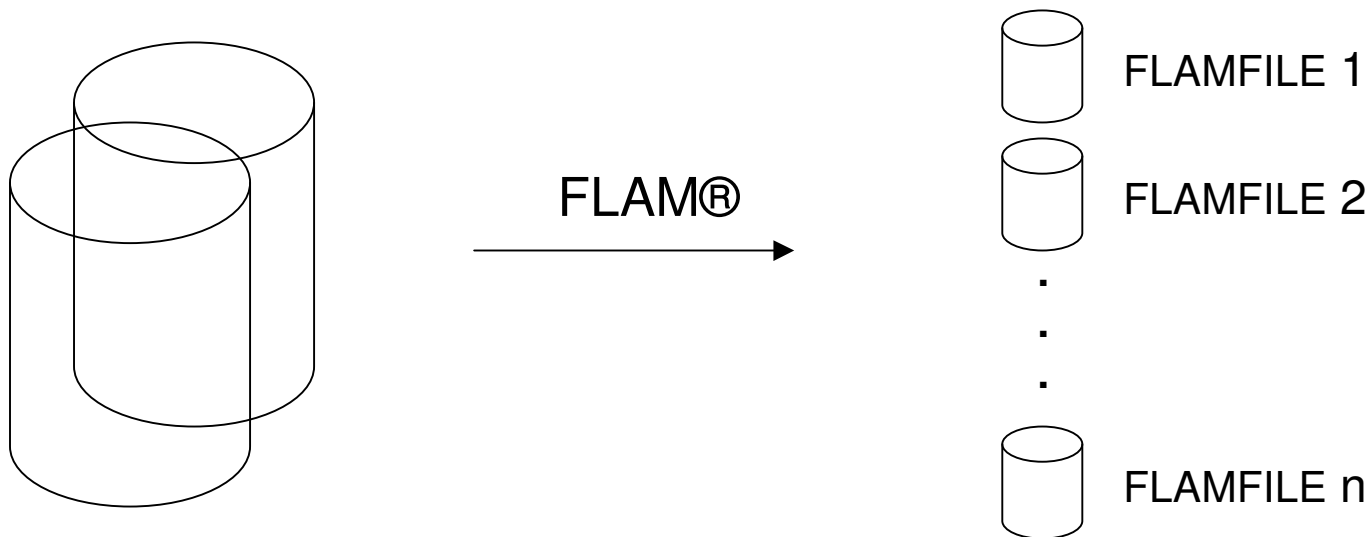


Funktionsbeispiel

Serielles Splitten

■ Problemstellung

- ◆ Anhänge für eMail-Attachments zu groß, File Transfer bricht ab bei großen Dateien, Speichermedium zu klein, Wartezeit bei großen Dateien



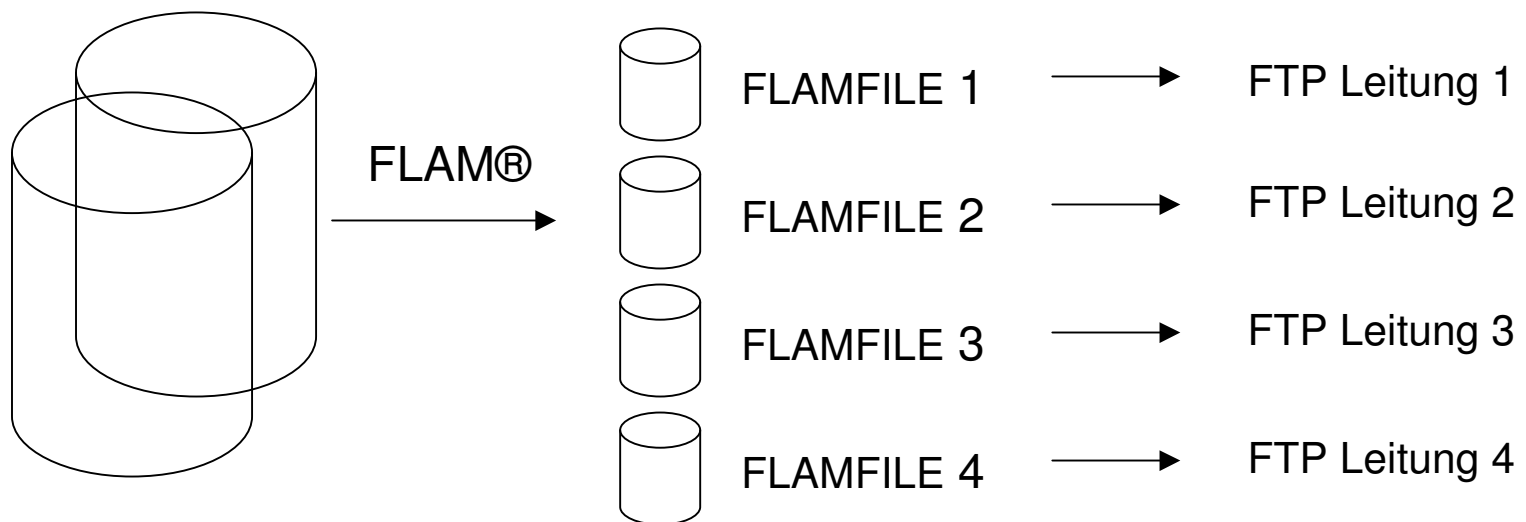
Die Größe eines FLAMFILE®-Fragments kann definiert werden.

Funktionsbeispiel

Paralleles Splitten

■ Problemstellung

- ◆ Mehrere Transportwege gleichzeitig bedienen



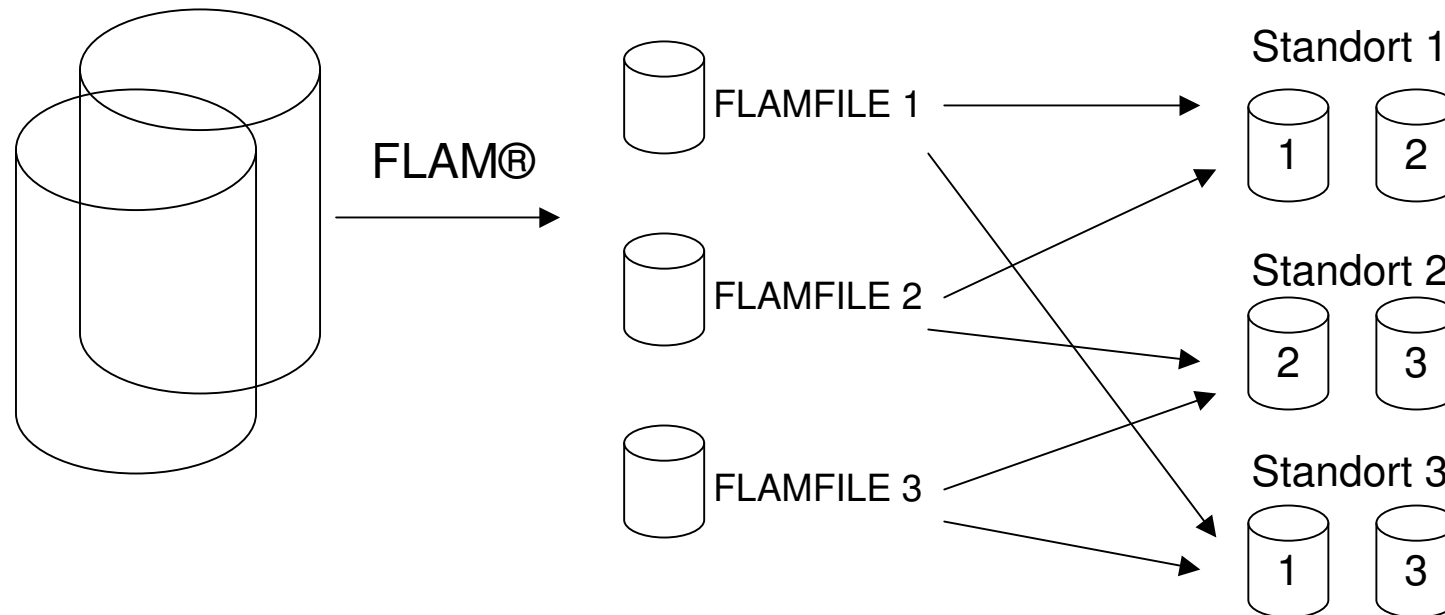
Alle Fragmente der FLAMFILE® müssen zur Dekomprimierung/
Entschlüsselung gleichzeitig zur Verfügung stehen!

Funktionsbeispiel

Paralleles Splitten

■ Problemstellung

- ◆ Sichere Archivierung ohne Aufwand

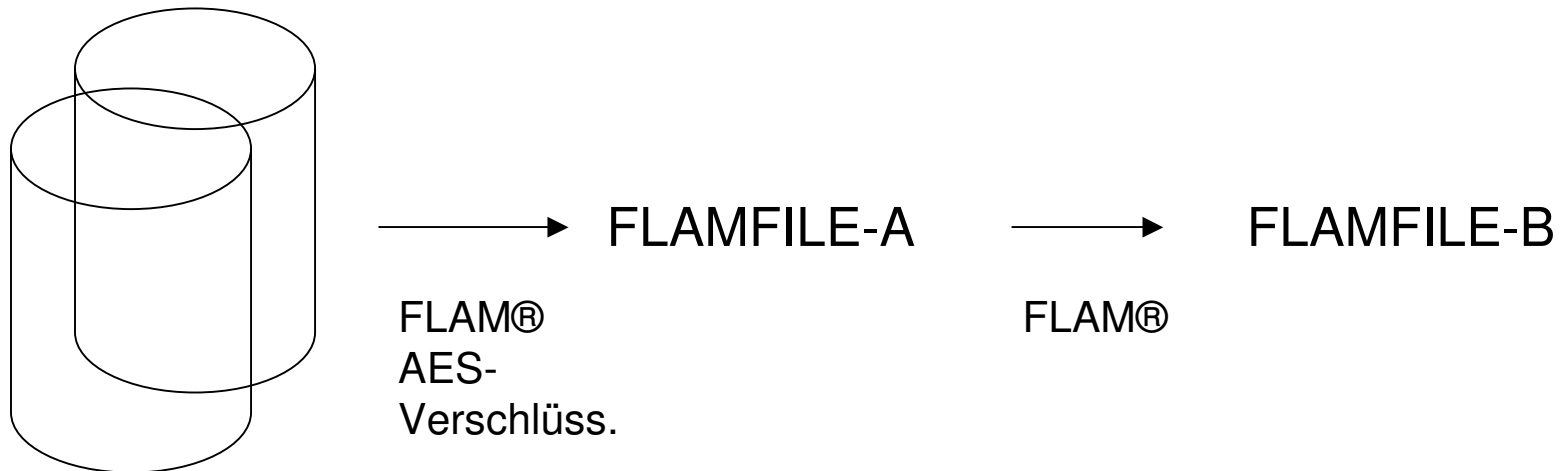


Kein Standort allein kann die Daten lesen, die Daten eines Standorts können verloren gehen und das Archiv ist rekonstruierbar.

Funktionsbeispiel Legendieren

■ Problemstellung

- ◆ Der Datei soll nicht angesehen werden, daß sie verschlüsselt ist.



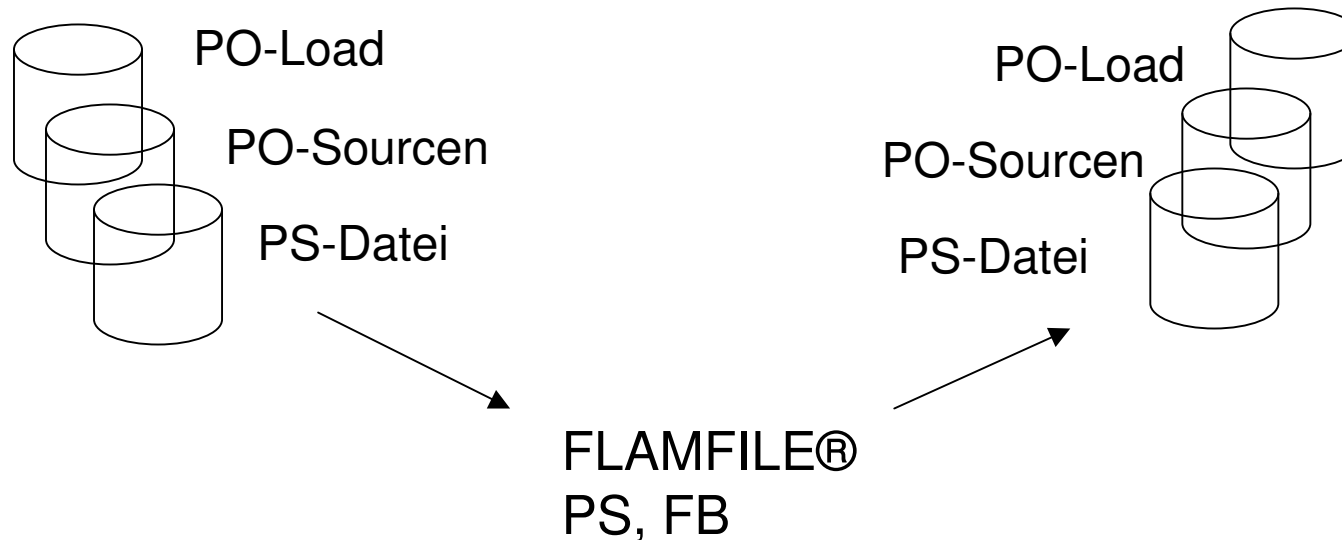
FLAMFILE-B sieht aus wie eine unverschlüsselte Datei.

Funktionsbeispiel

Programmversand/Wartung

■ Problemstellung

- ◆ Installation von Programm, Updates über Magnetbänder (auf Mainframes) wegen unterschiedlichster Dateiformate, Wartungsdaten enthalten vertrauliche Daten (z.B. in Dumps)

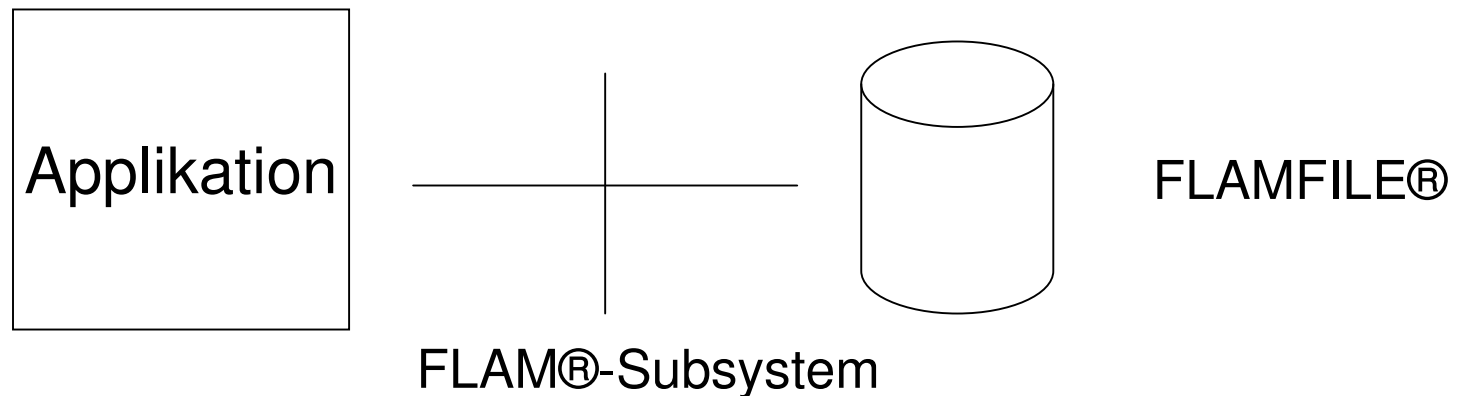


Die FLAMFILE® kann wie eine gewöhnliche sequentielle Datei übertragen werden.

Funktionsbeispiel MVS-Subsystem

■ Problemstellung

- ◆ Einführung von FLAM® ohne Änderung in den Applikationen.



FLAM®-Subsystem ist im MVS-(OS/390, z/OS)-Betriebssystem integrierbar, die Steuerung erfolgt über JCL (// DD SUBSYS=FLAM).