The Ins and Outs of VSAM … and MORE!

Jerry Spencer
Precisely – Senior Software Engineer
Barbara McDonald
IBM – DFSMS Product Manager

Thursday February 26, 2026

9:15 AM

Room: Salon 15

AGENDA

AGENDA

THE HISTORY OF VSAM
TYPES OF VSAM DATA SETS
HOW TO DEFINE VSAM DATA SETS
VSAM SHARING OPTIONS
HOW TO ACCESS VSAM DATA SETS
MOST ADVANCED METHOD OF SHARING
EZNOSQL
VSAM DOCUMENTATION
WRAP UP
QUESTIONS

THE HISTORY OF VSAM

THE HISTORY OF VSAM

VSAM – Virtual Storage Access Method

Introduced in 1973 as part of the OS/VS2 operating system

Designed to replace ISAM (Indexed Sequential Access Method)

TYPES OF VSAM DATA SETS

TYPES OF VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

Key-Sequenced Data Set (KSDS)

Stores records using a key for direct access and supports sequential processing.

Entry-Sequenced Data Set (ESDS)

Stores records in the order they are written, with each record assigned a Relative Byte Address (RBA).

Relative Record Data Set (RRDS)

Stores records in fixed slots, addressed by relative record numbers

Linear Data Set (LDS)

Stores data in an unstructured, byte-oriented format, often used for database files or user-managed storage.

HOW TO DEFINE VSAM DATA SETS

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

KSDS

```
//STEP1      EXEC    PGM=IDCAMS
//SYSPRINT   DD      SYSOUT=*
//SYSIN      DD      *
    DEFINE CLUSTER(NAME(MY.KSDS.DATASET)      -
           INDEXED                            -
           KEYS(10,0)                         -
           RECORDSIZE(80,200)                 -
           TRACKS(5,5)                        -
           FREESPACE(10,10))                  -
           VOLUME(STOR01))                    -
      DATA(NAME(MY.KSDS.DATASET.DATA))        -
      INDEX(NAME(MY.KSDS.DATASET.INDEX))
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

ESDS

```
//STEP1      EXEC    PGM=IDCAMS
//SYSPRINT   DD      SYSOUT=*
//SYSIN      DD      *
     DEFINE CLUSTER(NAME(MY.ESDS.DATASET)      -
            NONINDEXED                         -
            RECORDSIZE(80,200)                 -
            TRACKS(5,5)                        -
            VOLUME(STOR01))                    -
     DATA(NAME(MY.ESDS.DATASET.DATA))
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

RRDS

```
//STEP1      EXEC    PGM=IDCAMS
//SYSPRINT   DD      SYSOUT=*
//SYSIN      DD      *
    DEFINE CLUSTER(NAME(MY.RRDS.DATASET)     -
           NUMBERED                          -
           RECORDSIZE(100,100)               -
           TRACKS(5,5)                       -
           VOLUME(STOR01))                   -
    DATA(NAME(MY.RRDS.DATASET.DATA))
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

LDS

```
//STEP1      EXEC    PGM=IDCAMS
//SYSPRINT   DD      SYSOUT=*
//SYSIN      DD      *
     DEFINE CLUSTER(NAME(MY.LDS.DATASET)      -
            LINEAR                            -
            TRACKS(5,5)                       -
            VOLUME(STOR01))                   -
     DATA(NAME(MY.LDS.DATASET.DATA))
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

JCL DEFINED VSAM

```
//STEP1    EXEC   PGM=IEFBR14
//KSDS     DD   DSNAME=MY.KSDS.DATASET,DISP=(NEW,CATLG),
//            SPACE=(TRK,(5,5)),RECORG=KS,
//            KEYLEN=10,KEYOFF=0,LRECL=200,VOL=SER=STOR01
//ESDS     DD   DSNAME=MY.ESDS.DATASET,DISP=(NEW,CATLG),
//            SPACE=(TRK,(5,5)),RECORG=ES,
//            LRECL=200,VOL=SER=STOR01
//RRDS     DD   DSNAME=MY.RRDS.DATASET,DISP=(NEW,CATLG),
//            SPACE=(TRK,(5,5)),RECORG=RR,
//            LRECL=100,VOL=SER=STOR01
//LDS      DD   DSNAME=MY.LDS.DATASET,DISP=(NEW,CATLG),
//            SPACE=(TRK,(5,5)),RECORG=LS,
//            VOL=SER=STOR01
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

IDCAMS DEFINED VSAM LISTCAT

```
      LISTC ENT(MY.KSDS.DATASET) ALL
     CLUSTER ------- MY.KSDS.DATASET
          ASSOCIATIONS
            DATA-----MY.KSDS.DATASET.DATA
            INDEX----MY.KSDS.DATASET.INDEX
        DATA ------- MY.KSDS.DATASET.DATA
          ASSOCIATIONS
            CLUSTER--MY.KSDS.DATASET
          ATTRIBUTES
            KEYLEN----------------10      AVGLRECL--------------80      BUFSPACE-----------37376      CISIZE-------------18432
            RKP-------------------0       MAXLRECL-------------200      EXCPEXIT----------(NULL)      CI/CA----------------15
            SHROPTNS(1,3)   RECOVERY      UNIQUE           NOERASE      INDEXED       NOWRITECHK      UNORDERED        NOREUSE
            NONSPANNED
          ALLOCATION
            SPACE-TYPE---------TRACK      HI-A-RBA----------276480
            SPACE-PRI--------------5      HI-U-RBA---------------0
            SPACE-SEC--------------5
          VOLUME
            VOLSER------------STOR01      PHYREC-SIZE--------18432      HI-A-RBA----------276480      EXTENT-NUMBER----------1
            DEVTYPE------X'3010200F'      PHYRECS/TRK------------3      HI-U-RBA---------------0      EXTENT-TYPE--------X'40'
            VOLFLAG------------PRIME      TRACKS/CA--------------5
            EXTENTS:
            LOW-CCHH-----X'0034000C'      LOW-RBA----------------0      TRACKS-----------------5
            HIGH-CCHH----X'00350001'      HIGH-RBA----------276479
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

IDCAMS DEFINED VSAM LISTCAT

```
      INDEX ------ MY.KSDS.DATASET.INDEX
          ASSOCIATIONS
            CLUSTER--MY.KSDS.DATASET
          ATTRIBUTES
            KEYLEN----------------10     AVGLRECL---------------0     BUFSPACE---------------0     CISIZE---------------512
            RKP-------------------0      MAXLRECL-------------505     EXCPEXIT----------(NULL)     CI/CA-----------------49
            SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     NOWRITECHK     UNORDERED     NOREUSE
          ALLOCATION
            SPACE-TYPE---------TRACK     HI-A-RBA-----------25088
            SPACE-PRI----------------1   HI-U-RBA---------------0
            SPACE-SEC----------------1
          VOLUME
            VOLSER------------STOR01     PHYREC-SIZE----------512     HI-A-RBA-----------25088     EXTENT-NUMBER----------1
            DEVTYPE------X'3010200F'     PHYRECS/TRK-----------49     HI-U-RBA---------------0     EXTENT-TYPE--------X'40'
            VOLFLAG------------PRIME     TRACKS/CA--------------1
            EXTENTS:
            LOW-CCHH-----X'001F0007'     LOW-RBA----------------0     TRACKS-----------------1
            HIGH-CCHH----X'001F0007'     HIGH-RBA-----------25087
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

JCL DEFINED VSAM LISTCAT

```
 LISTC ENT(MY.KSDS.DATASET) ALL
CLUSTER ------- MY.KSDS.DATASET
     ASSOCIATIONS
       DATA-----MY.KSDS.DATASET.DATA
       INDEX----MY.KSDS.DATASET.INDEX
   DATA ------- MY.KSDS.DATASET.DATA
     ASSOCIATIONS
       CLUSTER--MY.KSDS.DATASET
     ATTRIBUTES
       KEYLEN----------------10     AVGLRECL-------------200     BUFSPACE-----------37376     CISIZE-------------18432
       RKP-------------------0      MAXLRECL-------------200     EXCPEXIT----------(NULL)     CI/CA-----------------15
       SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     INDEXED       NOWRITECHK     UNORDERED        NOREUSE
       NONSPANNED
     ALLOCATION
       SPACE-TYPE---------TRACK     HI-A-RBA----------276480
       SPACE-PRI--------------5     HI-U-RBA----------------0
       SPACE-SEC--------------5
     VOLUME
       VOLSER------------STOR01     PHYREC-SIZE--------18432     HI-A-RBA----------276480     EXTENT-NUMBER----------1
       DEVTYPE------X'3010200F'     PHYRECS/TRK------------3     HI-U-RBA----------------0     EXTENT-TYPE--------X'40'
       VOLFLAG------------PRIME     TRACKS/CA--------------5
       EXTENTS:
       LOW-CCHH-----X'0034000C'     LOW-RBA----------------0     TRACKS-----------------5
       HIGH-CCHH----X'00350001'     HIGH-RBA----------276479
```

HOW TO DEFINE VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

JCL DEFINED VSAM LISTCAT

```
      INDEX ------ MY.KSDS.DATASET.INDEX
          ASSOCIATIONS
            CLUSTER--MY.KSDS.DATASET
          ATTRIBUTES
            KEYLEN----------------10     AVGLRECL---------------0     BUFSPACE---------------0     CISIZE---------------512
            RKP-------------------0      MAXLRECL-------------505     EXCPEXIT----------(NULL)     CI/CA-----------------49
            SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     NOWRITECHK     UNORDERED     NOREUSE
          ALLOCATION
            SPACE-TYPE---------TRACK     HI-A-RBA-----------25088
            SPACE-PRI----------------1   HI-U-RBA---------------0
            SPACE-SEC----------------1
          VOLUME
            VOLSER------------STOR01     PHYREC-SIZE----------512     HI-A-RBA-----------25088     EXTENT-NUMBER----------1
            DEVTYPE------X'3010200F'     PHYRECS/TRK----------49      HI-U-RBA---------------0     EXTENT-TYPE--------X'40'
            VOLFLAG------------PRIME     TRACKS/CA--------------1
            EXTENTS:
            LOW-CCHH-----X'001F0007'     LOW-RBA----------------0     TRACKS-----------------1
            HIGH-CCHH----X'001F0007'     HIGH-RBA-----------25087
```

Comparing LISTCAT vs LISTSTAT

Catalog Information with LISTCAT

LISTCAT offers detailed insights into dataset attributes, locations and catalog management, aiding dataset organization and tracking.

Statistical Insights with LISTSTAT

LISTSTAT provides statistical metrics such as record counts and space usage, helping monitor performance and analyze storage.

Distinct Focus Areas

LISTCAT is key for catalog management and dataset properties, while LISTSTAT is vital for performance monitoring and storage analysis.

```
  SHCDS LISTSTAT('HL1.KSDS0001')                                    00110073
LIST STATISTICS (LISTSTAT):
CLUSTER----------HL1.KSDS0001
  DATA-------HL1.KSDS0001.DATA
    TOTAL RECORDS---------          6      CI SPLITS-------------          1
    RECORDS DELETED-------          0      CA SPLITS-------------          0
    RECORDS INSERTED------          1      EXCPS-----------------         24
    RECORDS UPDATED-------          2      EXTENTS---------------------    1
    RECORDS RETRIEVED-----          2      FREE SPACE CIS------            9
    HI-A-CI-------------           12      HI-U-CI-------------           12
    CI-SIZE-------------         4096
  INDEX------HL1.KSDS0001.INDEX
    TOTAL RECORDS---------          1      CI SPLITS-------------          0
    CA RECLAIMS-----------          0      CA SPLITS-------------          0
    RECLAIMED-CA REUSES---          0      EXCPS-----------------         31
    RECORDS UPDATED-------          0      EXTENTS---------------------    1
    RECORDS RETRIEVED-----          0      FREE SPACE CIS------           47
    HI-A-CI-------------           48      HI-U-CI-------------            1
    HI-LEVEL-CI-----------          0      INDEX LEVELS----------------    1
    CI-SIZE---------------        512

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 4
```

IDCAMS DEFINED VSAM LISTCAT

```
      LISTC ENT(MY.KSDS.DATASET) ALL
     CLUSTER ------- MY.KSDS.DATASET
          ASSOCIATIONS
            DATA-----MY.KSDS.DATASET.DATA
            INDEX----MY.KSDS.DATASET.INDEX
        DATA ------- MY.KSDS.DATASET.DATA
          ASSOCIATIONS
            CLUSTER--MY.KSDS.DATASET
          ATTRIBUTES
            KEYLEN----------------10     AVGLRECL--------------80     BUFSPACE-----------37376     CISIZE-------------18432
            RKP-------------------0      MAXLRECL-------------200     EXCPEXIT----------(NULL)     CI/CA-----------------15
            SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     INDEXED       NOWRITECHK     UNORDERED        NOREUSE
            NONSPANNED
          ALLOCATION
            SPACE-TYPE---------TRACK     HI-A-RBA----------276480
            SPACE-PRI--------------5     HI-U-RBA----------------0
            SPACE-SEC--------------5
          VOLUME
            VOLSER------------STOR01     PHYREC-SIZE--------18432     HI-A-RBA----------276480     EXTENT-NUMBER----------1
            DEVTYPE------X'3010200F'     PHYRECS/TRK------------3     HI-U-RBA----------------0    EXTENT-TYPE--------X'40'
            VOLFLAG------------PRIME     TRACKS/CA--------------5
            EXTENTS:
            LOW-CCHH-----X'0034000C'     LOW-RBA----------------0     TRACKS-----------------5
            HIGH-CCHH----X'00350001'     HIGH-RBA----------276479
```

IDCAMS DEFINED VSAM LISTCAT

```
      INDEX ------ MY.KSDS.DATASET.INDEX
          ASSOCIATIONS
            CLUSTER--MY.KSDS.DATASET
          ATTRIBUTES
            KEYLEN----------------10     AVGLRECL---------------0     BUFSPACE---------------0     CISIZE---------------512
            RKP-------------------0      MAXLRECL-------------505     EXCPEXIT----------(NULL)     CI/CA-----------------49
            SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     NOWRITECHK     UNORDERED     NOREUSE
          ALLOCATION
            SPACE-TYPE---------TRACK     HI-A-RBA-----------25088
            SPACE-PRI--------------1     HI-U-RBA---------------0
            SPACE-SEC--------------1
          VOLUME
            VOLSER------------STOR01     PHYREC-SIZE----------512     HI-A-RBA-----------25088     EXTENT-NUMBER----------1
            DEVTYPE------X'3010200F'     PHYRECS/TRK-----------49     HI-U-RBA---------------0     EXTENT-TYPE--------X'40'
            VOLFLAG------------PRIME     TRACKS/CA--------------1
            EXTENTS:
            LOW-CCHH-----X'001F0007'     LOW-RBA----------------0     TRACKS-----------------1
            HIGH-CCHH----X'001F0007'     HIGH-RBA-----------25087
```

VSAM SHARING OPTIONS

VSAM SHARING OPTIONS

SHAREOPTIONS(x,y)

x - Sharing across a single LPAR

y - Sharing across the entire SYSPLEX

Both can range from 1 to 4

1 - Only one user at a time
2 - Multiple readers or one writer
3 - Multiple readers and writers
4 - Wild West … all bets are off - **VSAM WILL NOT PROTECT YOU**

VSAM SHARING OPTIONS

SHARE
EDUCATE • NETWORK • INFLUENCE

VSAM ACCESS MODES

NSR – Non-Shared Resources

Each task/job has its own buffers

No buffer sharing between users

Minimal internal coordination

Provides “read ahead” sequential access for improved performance.

VSAM SHARING OPTIONS

SHARE
EDUCATE • NETWORK • INFLUENCE

VSAM ACCESS MODES

LSR – Local Shared Resources

Shared buffers within one region

Buffers are shared across tasks in the same address space or region

Relies on ENQs and SHAREOPTIONS

No cross-region coordination

VSAM SHARING OPTIONS

SHARE
EDUCATE • NETWORK • INFLUENCE

VSAM ACCESS MODES

GSR – Global Shared Resources

Shared buffers coordinated globally

Shared buffering on a system

Permits cross-system sharing coordinated by the application via the VSAM Shared Interface (VSI).

VSAM SHARING OPTIONS

SHARE
EDUCATE • NETWORK • INFLUENCE

VSAM ACCESS MODES

NOT specified as part of the DEFINE CLUSTER

Specified at OPEN time

Can specify AMP= in JCL

```
//STEP1   EXEC  PGM=NSRPGM
//DD1     DD    DSN=SHARED.KSDS,DISP=SHR,
//            AMP=('BUFND=20,BUFNI=10)
```

HOW TO ACCESS VSAM DATA SETS

HOW TO ACCESS VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

COBOL Code for KSDS

```
IDENTIFICATION DIVISION.
 PROGRAM-ID.    READVSAM.
*AUTHOR. JERRY SPENCER
*REMARKS. READ A RECORD FROM A VSAM KSDS
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT VSAMFILE ASSIGN TO MYKSDS
         ORGANIZATION IS INDEXED
         ACCESS MODE IS SEQUENTIAL
         RECORD KEY IS MY-KEY.
```

HOW TO ACCESS VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

COBOL Code for KSDS

```
DATA DIVISION.
FILE SECTION.

 FD  VSAMFILE
     RECORD CONTAINS 200 CHARACTERS
     DATA RECORD IS MY-RECORD.
 01  MY-RECORD.
     05  MY-KEY      PIC X(10).
     05  FILL        PIC X(190).
```

HOW TO ACCESS VSAM DATA SETS

SHARE
EDUCATE • NETWORK • INFLUENCE

COBOL Code for KSDS

```
WORKING-STORAGE SECTION.
77  FILLER PIC X(37)  VALUE
    "READVSAM WORKING STORAGE BEGINS HERE".

PROCEDURE DIVISION.
    OPEN INPUT VSAMFILE
    READ VSAMFILE
    CLOSE VSAMFILE
    STOP RUN.
```

MOST ADVANCED METHOD OF SHARING

MOST ADVANCED METHOD OF SHARING

SHARE
EDUCATE • NETWORK • INFLUENCE

RLS (Record-Level Sharing)

Allows readers and updaters across a sysplex to share the dataset with full integrity

Uses record-level locks to improve performance

Address space SMSVSAM manages buffers

Buffers are consistent across systems

EZNOSQL

VSAM Record Level Sharing (RLS) and EzNoSQL

Thursday, February 26, 2026

Barbara McDonald, IBM z/OS DFSMS Product Manager (bawhite@us.ibm.com)
Alex Hong, DFSMS RLS Development

VSAM RLS

What is VSAM RLS?

- VSAM RLS provides full data sharing to your existing VSAM files (ie KSDS, ESDS,(V)RRDS), in a parallel SYSPLEX.
- Allows for high availability (HA) by enabling data sets to be shared cross LPARs/CECs via structures in the Coupling Facility (CF).
- Allows for high scalability by enabling many address spaces and LPARs to share the same files.
- Provides record level serialization, 64-bit buffering, and global caching for better performance.

Why VSAM RLS?

Problem:

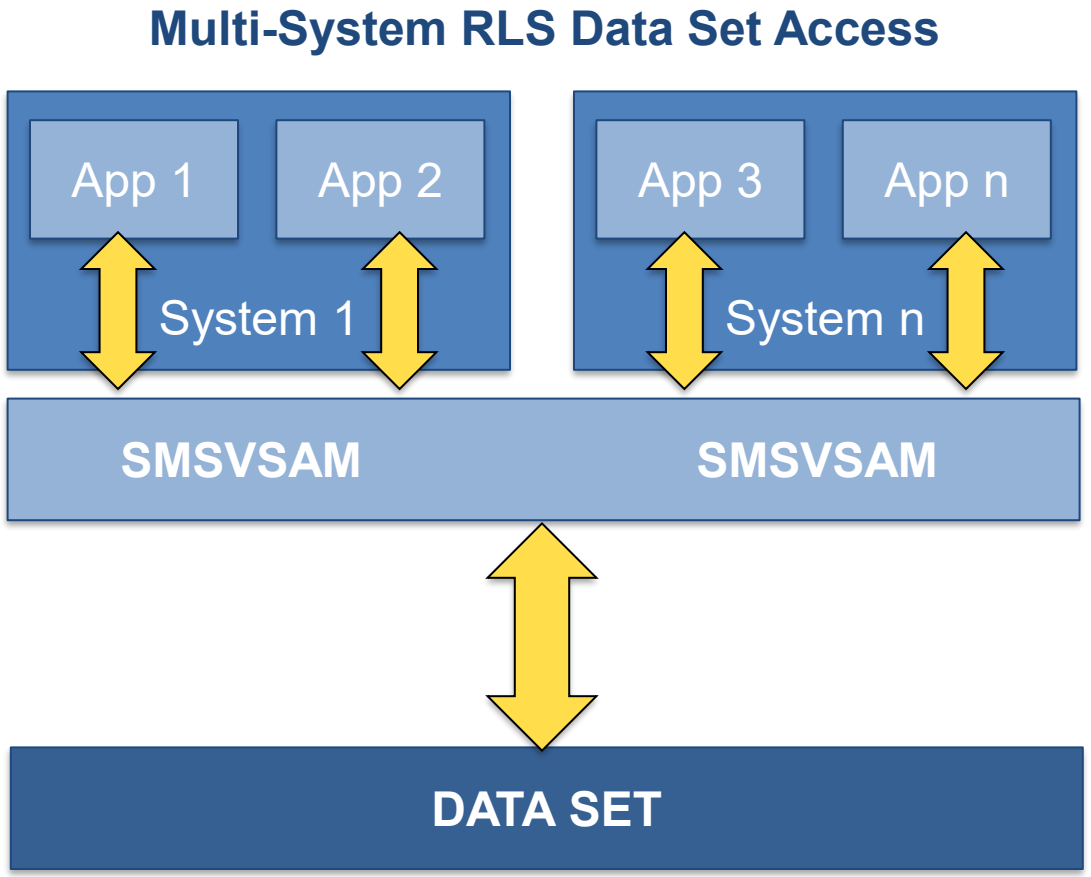
- One data set, many users, many systems
- Serialization of concurrent access is complex
- Data can get lost
- Data sets can be broken

Previous solutions:

- Homegrown methods via GRS
- CICS AOR / FOR (Function Shipping)

RLS solution:

- VSAM: Record Level Sharing
- All access goes through SMSVSAM
- Sysplex-wide serialization through record locks in the CF



RLS Configuration for the System Programmer

- Requires at least one z/OS image running in **sysplex mode**
  - Refer to z/OS MVS Setting Up a Sysplex for more information on configuring a Parallel Sysplex.
- Sysplex mode requires at least one internal or external **Coupling Facility** (CF)
  - Refer to Processors Resource/Systems Manager (PR/SM) Planning Guide for more information on Coupling Facility planning considerations.
- Initialize **SMSVSAM server address space** on each z/OS instance in the sysplex
  - If transactional consistency is required, enable and configure Transactional VSAM (DFSMStvs), including the availability of forward recovery log streams, if required by the application
  - Refer to Administering VSAM record-level sharing, or VSAM Demystified (Chapters 5 and 6)

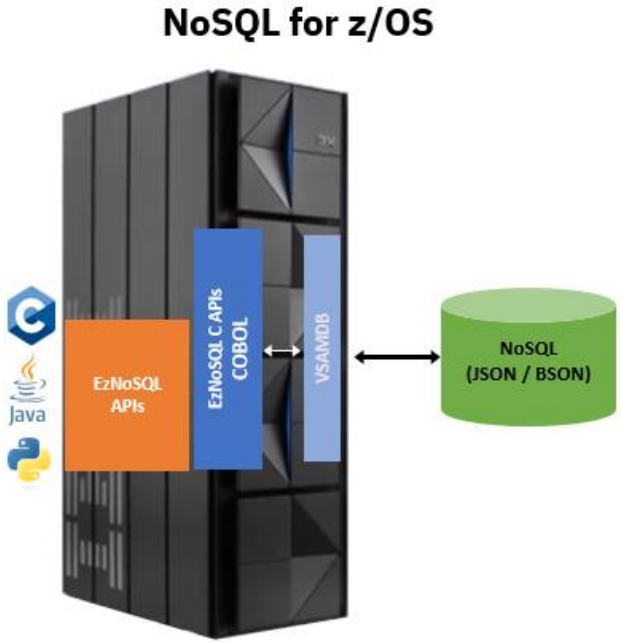
RLS Configuration Requirements for the Storage Admin

- Define and activate at least two **sharing control data sets** (SHCDS), and one spare SHCDS for recovery purposes
  - Refer to Defining sharing control data sets
- Define **CF cache and lock structures** using the CF resource manager (CFRM) policy
  - Refer to Defining CF cache structures and Defining the primary CF lock structure
- Define CF cache and lock structures in the **SMS base configuration** (ie SCDS)
  - Refer to Defining CF cache structures in the SMS base configuration
- Associate CF cache set and lock names with SMS **storage class definitions**, and write **ACS routines** to associate storage class definitions that map to CF cache and lock structures with data sets
  - Refer to Defining storage classes for VSAM RLS
- Optionally define VSAM RLS attributes in SMS **data classes** (DATACLAS)
  - Refer to Defining Shareoptions and RLS attributes for data class

EZNOSQL

What is EzNoSQL?

- z/OS-native NoSQL API solution for document-oriented databases
- Utilizes full data-sharing capabilities of IBM's Parallel Sysplex technology
- Inherits from z/OS: in-memory caching, system-managed storage, encryption, compression, integrated disaster recovery, and lifecycle management
- Language APIs (COBOL, C, Java, and Python)



"*By making mainframe data available via standard cloud APIs and supporting popular languages like* ***Python for NoSQL database access****, IBM is working to lower the barrier to entry for a new generation of developers… This is an essential step towards modernizing how enterprises interact with their most valuable data assets.*" -- z/OS 3.2: Is Mainframe AI the Enterprise Game Changer? Steven Dickens, CEO HyperFRAME Research

Why EzNoSQL?

SHARE
EDUCATE • NETWORK • INFLUENCE

Unified data access

Offers shared view, high-performance without data partitioning

3-tier caching hierarchy (local buffer, global CF, physical I/O)

Consistent updates

- Allows transaction consistency through simple commit and backout APIs
- Cross-invalidates buffer pool automatically

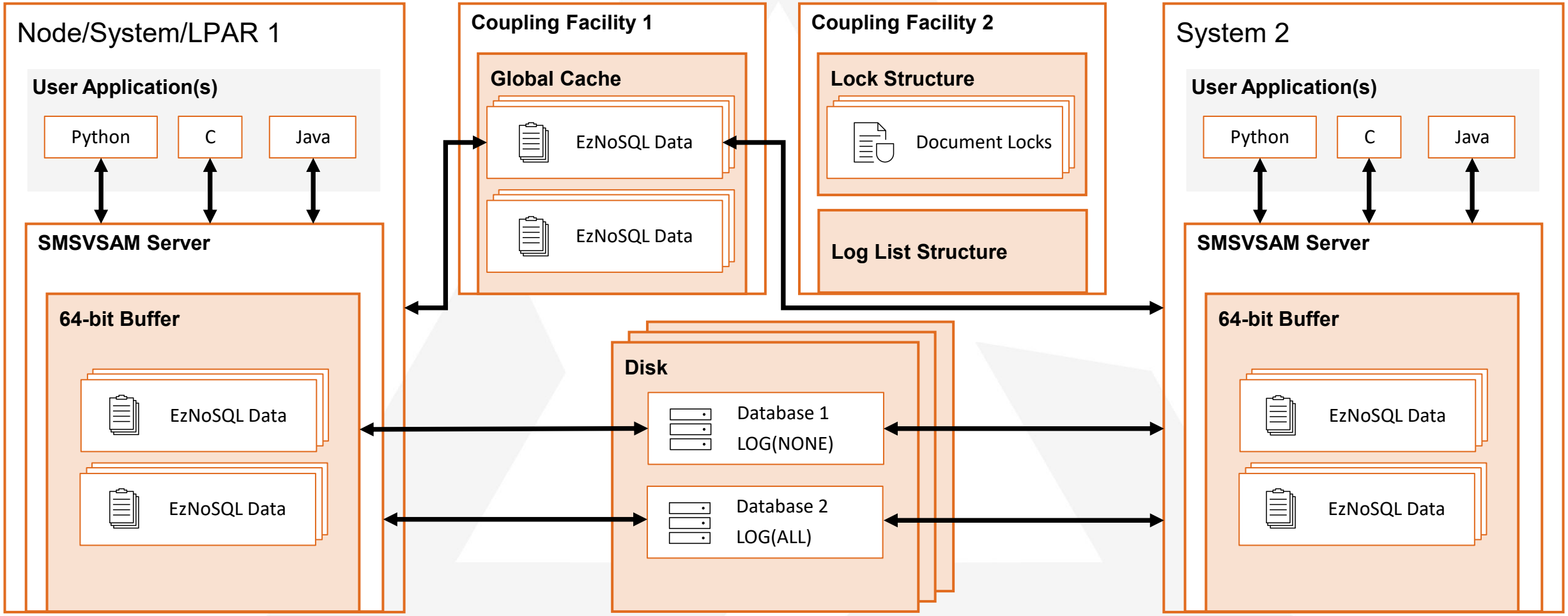
Simplified programming

- Eliminates complex logic for replication and eventual consistency
- Simplifies application functionality

EzNoSQL Indices

- **Primary keys**
  - Must be unique, but can be autogenerated for users (i.e., “znsq_id”)
  - Database can be defined with sequential insert (ordered_index)
    - Learn about ordered indexes - EZ Ordered Indexing! blog post
- **Secondary keys**
  - Optionally defined
  - May be either unique or non-unique
- **Multi-keys**
  - Can be both a primary or secondary key
  - Enables key values within embedded documents or arrays (secondary indices only)

EzNoSQL Sysplex Design



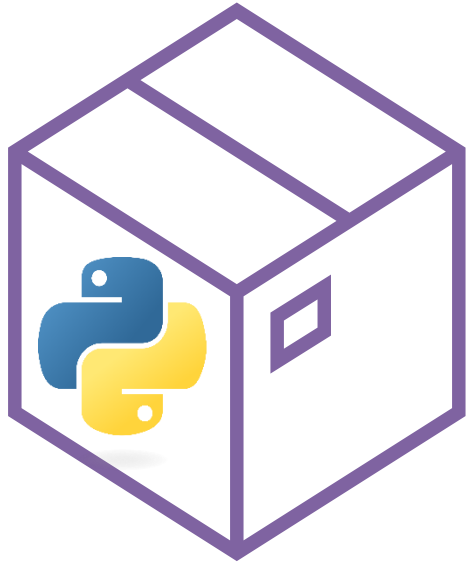
API Architecture

Unix System Services	*.java | *.py	Source Files
Language Interoperability Layer	libigwznsqj.so | c_ext.cpython-312.so	Shared Object Libraries
C Native Layer	libigwznsqd64.so	
PL/X Layer	IGWVRG64	VSAM RLS Module

- Java and Python interfaces with Shared Object Libraries.
- The language interoperability layer performs data conversion and argument parsing.

pyeznosql Package

- Contains definitions, statements, and imports that consolidate the individual modules as one package.
- User can import pyeznosql to access all available API functions.
  - import pyeznosql



pyeznosql Modules

SHARE
EDUCATE • NETWORK • INFLUENCE

Database management APIs	Connection management and document operation APIs	Result set operation APIs
– znsq_create() – znsq_create_index() – znsq_add_index() – znsq_drop_index() – znsq_destroy() – znsq_last_result()	– znsq_open() – znsq_close() – znsq_write() – znsq_delete() – znsq_replace() – znsq_commit() – znsq_abort() – znsq_set_autocommit() – znsq_position()	– znsq_write_result() – znsq_next_result() – znsq_delete_result() – znsq_replace_result() – znsq_close_result()

API System Requirements

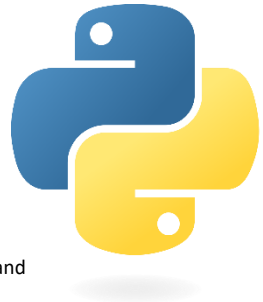
- **IBM C/C++ for z/OS**
  - z/OS V2.4 XL C/C++
  - Available with **OA62553** (z/OS V2.5)
- **IBM Semeru / Java**
  - Java 8+
  - Available with **OA64811** (z/OS 3.1), **OA64018** (z/OS V2.5)
- **IBM Open Enterprise SDK for Python**
  - Python 3.12.0+
  - Available with **OA66418** (z/OS 3.1)
- **z/OS UNIX System Services**
  - C - z/OS 2.4+
  - Java – z/OS 2.4+
  - Python – z/OS 3.1+

API Performance Metrics[1]

- z16 LPAR with 8 threads
- 100% read hit-rate
- 10 million documents
- 100 GB buffer

300,000 reads/s
32,600 writes/s

280,000 reads/s
32,200 writes/s

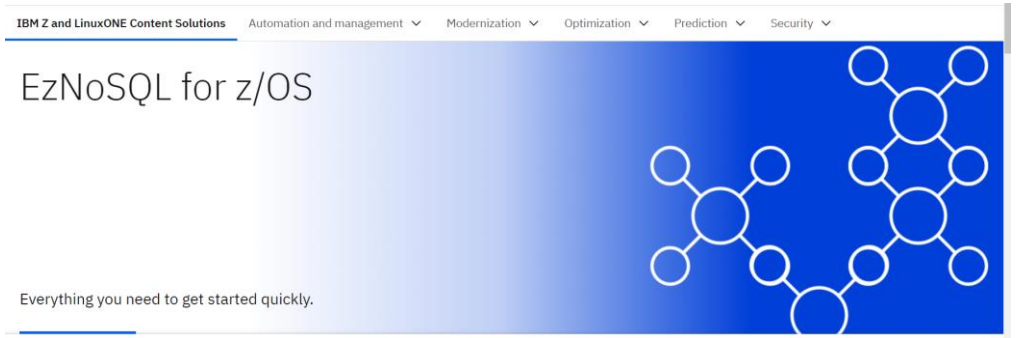


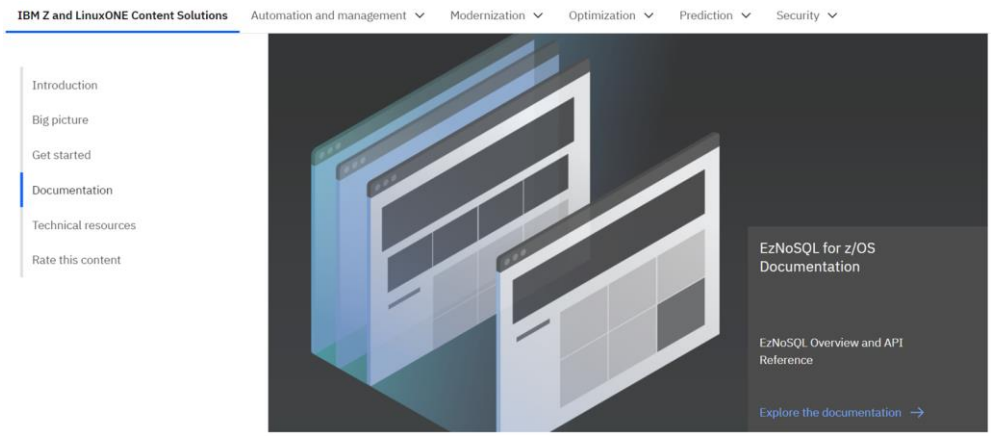
260,500 reads/s
32,300 writes/s

Source: 1. EzNoSQL Performance Proof Points
*Measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels

Where to go for more information

- EzNoSQL Content Solution website has everything needed to get started!
  - EzNoSQL Documentation
  - Javadoc API Reference
  - Python API Reference
- Informational blogs
  - "NoSQL for z/OS? So EZ!"
  - EZ Ordered Indexing!
  - NoSQL for z/OS? Even EZer!
- Submit a requirement via IBM Ideas Portal
  - Add your idea under Product Z and operating system z/OS, then select category DFSMS EzNoSQL





VSAM DOCUMENTATION

VSAM DOCUMENTATION

DFSMS Using Data Sets

DFSMS Macro Instructions for Data Sets

VSAM Demystified

https://www.jaymoseley.com/hercules/vs_tutor/vstutor.htm

WRAP UP

WRAP UP

THE HISTORY OF VSAM

- Introduced in 1973
- Replaced ISAM and BDAM

TYPES OF VSAM DATA SETS

- KSDS
- ESDS
- RRDS
- LDS

HOW TO DEFINE VSAM DATA SETS

- IDCAMS DEFINE CLUSTER
- JCL Defined VSAM

56

WRAP UP

HOW TO ACCESS VSAM DATA SETS

- Simple COBOL program
- Reads first record of a KSDS

VSAM DOCUMENTATION

- IBM documentation
- Non-IBM discussion of VSAM

57

Connect with DFSMS on Social Media

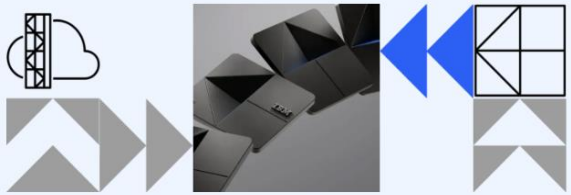
Join the IBM DFSMS Community and Mainframe Data Management LinkedIn Group!

Participate in the System Z Enthusiasts Discord

SHARE
EDUCATE • NETWORK • INFLUENCE

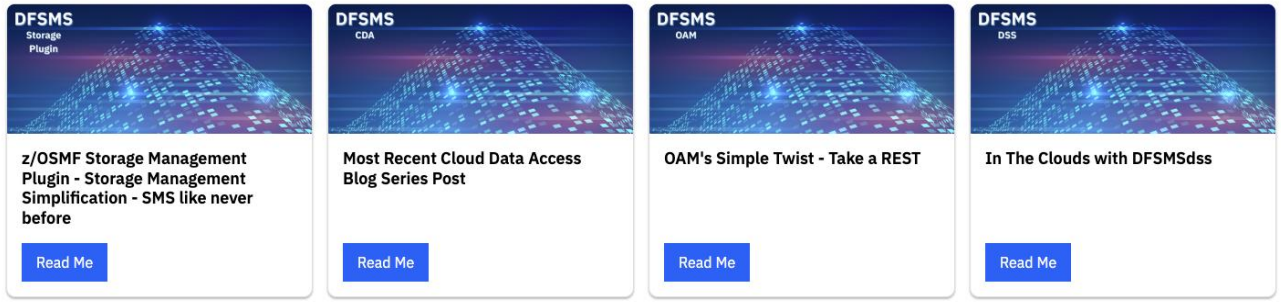
z/OS DFSMS

z/OS Data Subsystem that delivers core Data, Data Resiliency and Data Lifecycle Management Solutions for your Enterprise.



Group Home | Threads 19 | Blogs 43 | Upcoming Events 0 | Library 11 | Members 205 | Settings

Edit Featured Cards



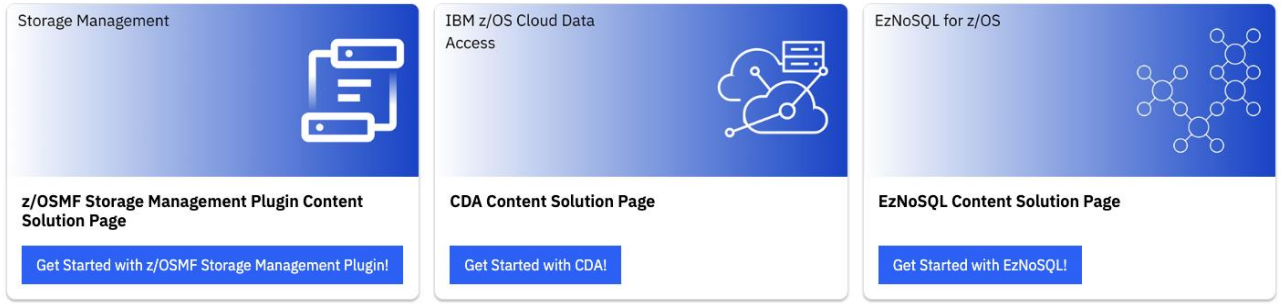
z/OSMF Storage Management Plugin - Storage Management Simplification - SMS like never before
Read Me

Most Recent Cloud Data Access Blog Series Post
Read Me

OAM's Simple Twist - Take a REST
Read Me

In The Clouds with DFSMSdss
Read Me

Edit Featured Cards



z/OSMF Storage Management Plugin Content Solution Page
Get Started with z/OSMF Storage Management Plugin!

CDA Content Solution Page
Get Started with CDA!

EzNoSQL Content Solution Page
Get Started with EzNoSQL!

Alexis Kapica • You
z/OS DFSMS Product Manager at IBM | Data Management | Cloud | Math ...
1w •

Hey Mainframers, another exciting day: z/OS 3.2 is here and we have so many great offerings in DFSMS!!

Check out this new blog to learn more about the future of DFSMS Storage Management with the **NEW** z/OSMF Storage Management Plugin: https://lnkd.in/dy2EuMr3

Take a look at the other blogs on the DFSMS Community or the announcement from July to see more information about the new functionality coming your way in z/OS 3.2: https://lnkd.in/dECHpwEj

We would love to hear your thoughts on the brand new z/OSMF Storage Management Plugin and the z/OS 3.2 offerings! Leave a comment on this post (or the blog post), start a discussion on the DFSMS Community Page, or join the Mainframe Data Management LinkedIn group to discuss this topic there!

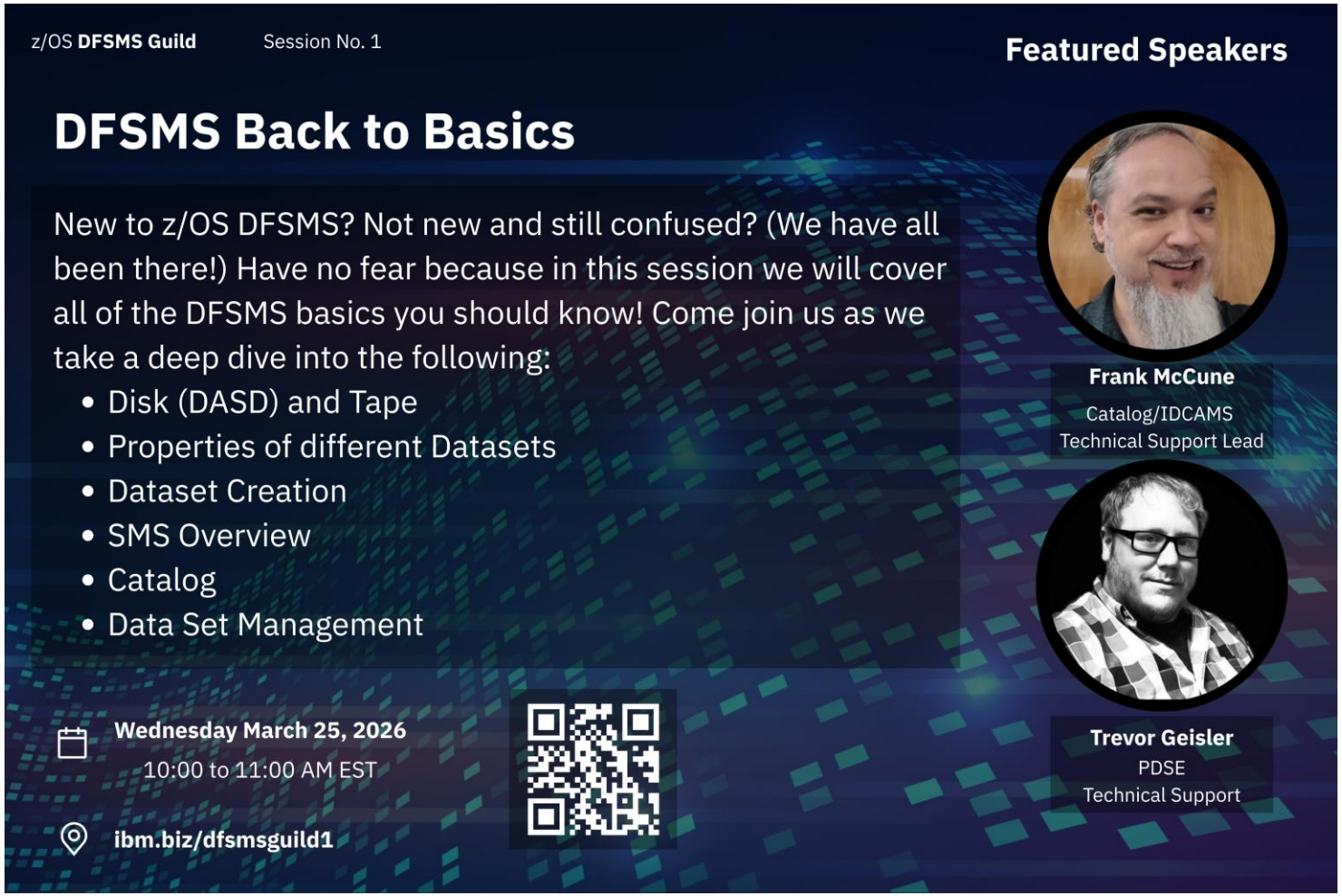
DFSMS Community Page: https://lnkd.in/ea-WqMt9

Mainframe Data Management Group: https://lnkd.in/eDx7rshx

#IBMZ #IBMz17 #DFSMS #StorageManagement #CloudDataManagement #EzNoSQL #NoSQL #OAM #zOSMF #RMM #DataManagement #HybridCloud #mainframe #MainframeDataManagement #CDA #CloudDataAccess #Cloud #HSM #DSS

Announcing the brand-new z/OS DFSMS Guild!

SHARE
EDUCATE • NETWORK • INFLUENCE



QUESTIONS

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation