RACF® and the Parallel Sysplex

A RACFers View of the z/OS® Parallel Sysplex®

Mark Nelson, CISSP®, CSSLP®

z/OS Security Server (RACF®) Design and Development
markan@us.ibm.com
IBM® Poughkeepsie

14 October 2021

RACF and the Parallel Sysplex - Agenda

- **An overview of the RACF data base and its structure**
- **An overview of the Sysplex Technologies**
- **How and why RACF uses these Sysplex Technologies**

Introduction to the RACF Data Base

What is the RACF Database?

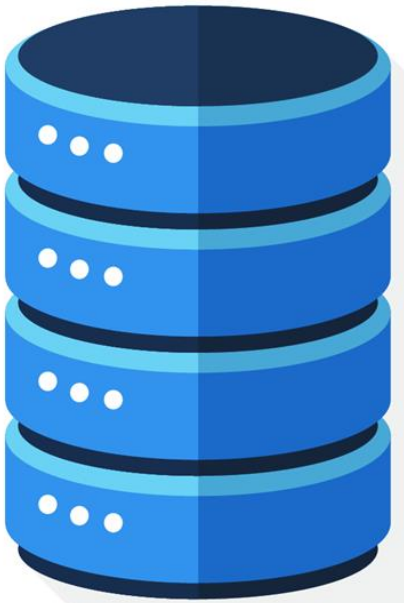
- **Storage location for the vast majority of RACF's operational and control information.**
- **Can be configured with an on-line backup**
- **Can consist of a single data set or it can be split across multiple data sets.**
- **Has its own high-speed data access mechanism**
- **Has its own serialization mechanisms (not SYSDSN ENQs)**

RACF Data Base Contents

- **The RACF database contains:**
  - System settings (SETROPTS) such as which classes are active/RACLISTed/etc., password options
  - User, group, data set, and other resource definitions (profiles)
  - Index information to quickly locate profiles
  - Meta data (templates) that define the fields within profiles
  - Control information for the RACF data base

RACF Data Base Contents…

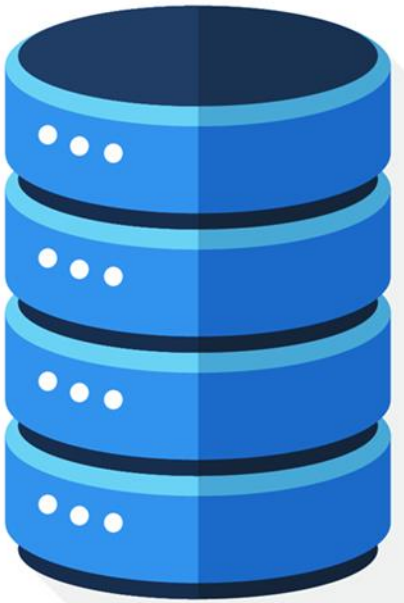
- **All of the information in the RACF data base is located by RACF using the relative byte address (RBA) of the data**
  - The RBA is the offset of the information from the RBA 0, the start the data set
  - Certain RBAs are “defined by the RACF architecture”:
    - RBA X’000000000000’: The most important block in any RACF data set **The inventory control block (ICB).**
    - RBAs X’000000001000’ through X’000000008000’: The RACF data base templates
    - All other data blocks in a RACF data set can be located at any RBA



- **All I/O to the RACF database is done in 4K blocks**
  - Data is addressed by CCHHRR (non-VSAM) or block number (VSAM)
  - Translating the RBA to the block number is trivial: Shift right by 12-bits

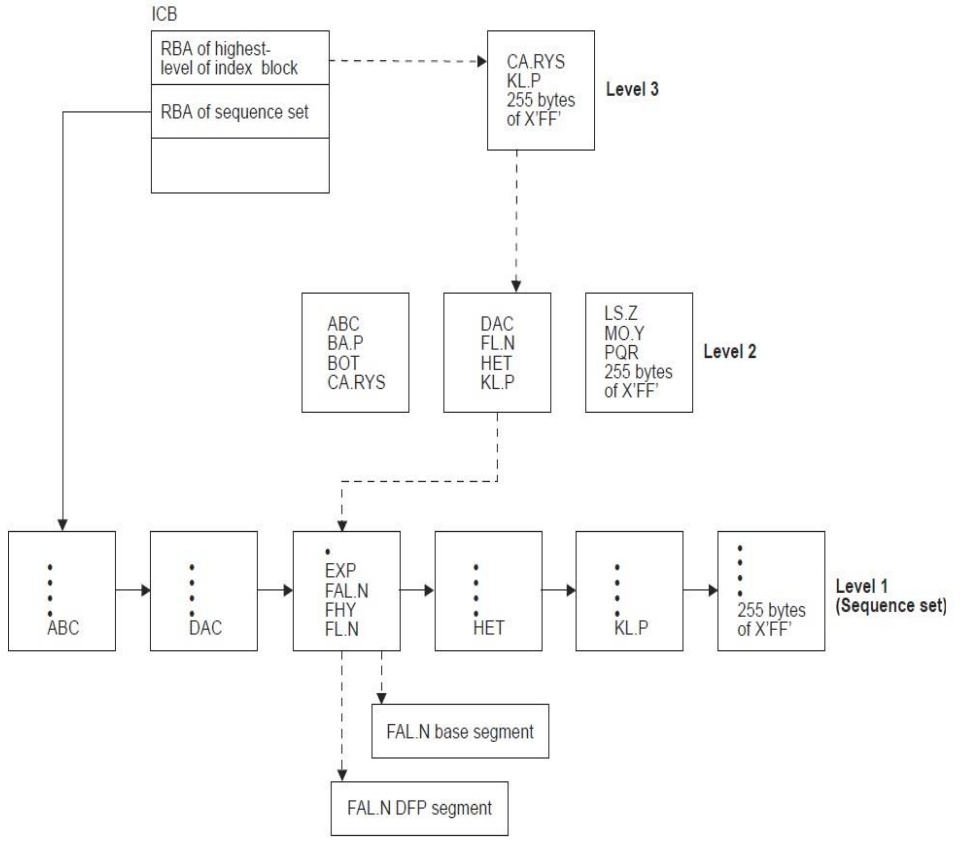
RACF Data Base Contents…

- **The ICB contains control information for the RACF data set, including:**
  - Global system options (classes active, RACLISTed, password options…)
  - Index control information (pointer to the top-level index blocks, allocation information, top level alias index blocks)
  - Update count information for each level of indexes and for data block



RACF Data Base Contents…

- **RACF has a high-performance index which allows RACF to quickly find the RBA of any profile in the RACF data set so that the data in the profile can be read**
  - Each RACF data set has its own index structure
  - The index blocks point to data blocks which contain the actual profile data
  - The ICB ties the index components together
- **RACF translates the RBA of the profile into a cylinder/head/record (CCHHRR) address for the actual reading and writing of data (non-VSAM) or record number (VSAM)**



RACF is its Own Access Method

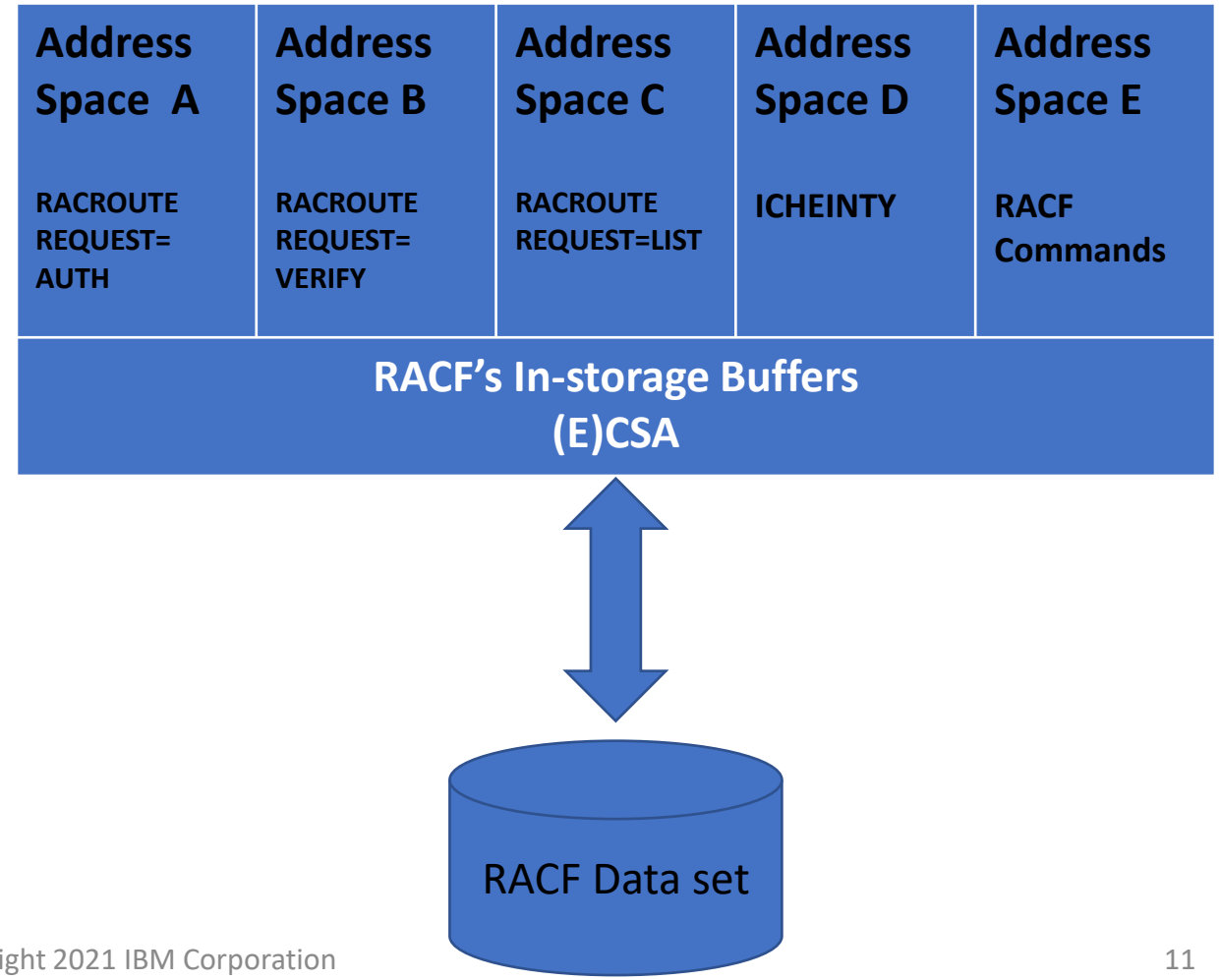
- **Acting as its own “access manager” means that RACF has to provide its own serialization mechanisms, among these:**
  - Serialization bit in the RACF ICB
    - Controlled by IRRUT400/IRRDBU00 “LOCKINPUT”/”UNLOCKINPUT”
  - In non-RACF datasharing mode, RESERVEs against the device containing the RACF data set(s)
    - Recommendation: Convert to global (“SCOPE=SYSTEMS”) ENQs
  - SYSZRACF, SYSZRACn ENQs
    - Documented in RACF Systems Programmers Guide
- **Utilities which access the RACF data base must honor RACF’s serialization**
- **You must use only the RACF utilities for copying a live RACF data base**

RACF In-Storage Buffers

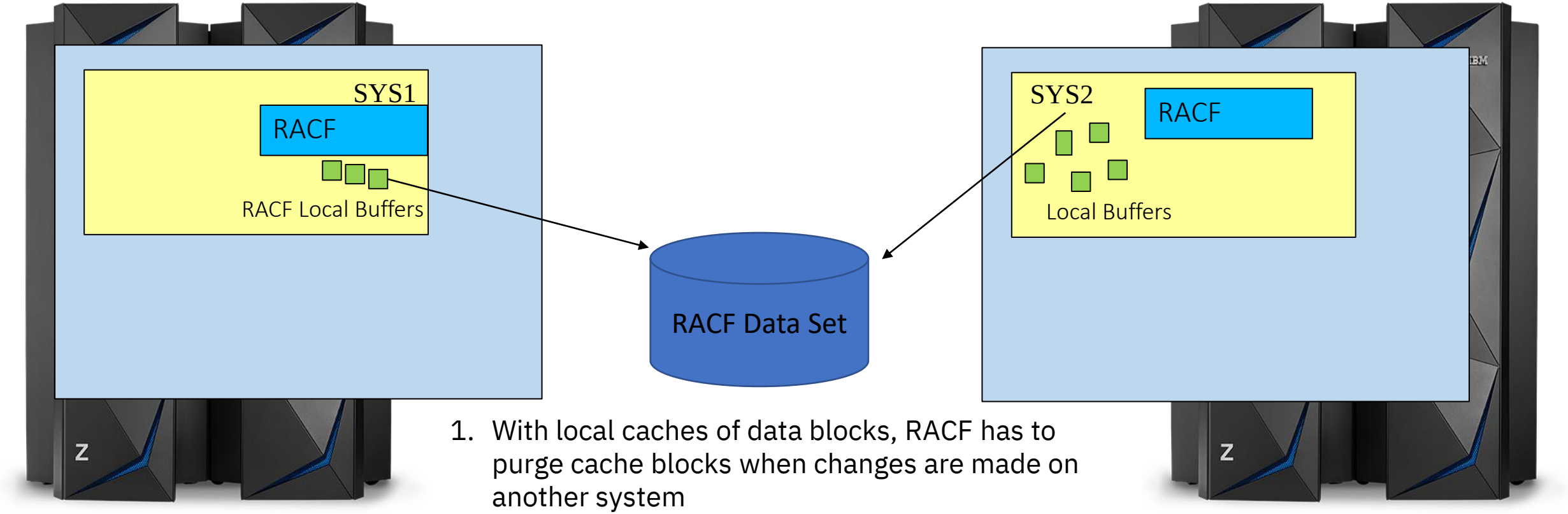
- **RACF allows the definition of up to 255 4K-byte in-storage buffers (called "resident data blocks") *per RACF data set* to allow RACF to cache frequently-used RACF database blocks**
  - Specified in the RACF data set names table ICHRDSNT or in the IRRPRMxx PARMLIB member
  - Reside in fetch-protected (E)CSA
  - "Least recently used" cast-out algorithm
  - *RACF must invalidate the buffers when blocks are changed on other systems*
    - Invalidation is based on counters maintained in the ICB
- **Storage and retrieval is done by the RBA of the block (index, data, ICB, BAM) in the RACF database**

What RACF Functions use these Buffers?

- **Any RACF command, RACROUTE, RACF independent system macro (RACINIT, RACHECK,…) or SAF callable service which results in a call to the RACF manager (ICHEINTY) references the RACF in-storage buffers**



Sharing a RACF Data Set Between Systems



1. With local caches of data blocks, RACF has to purge cache blocks when changes are made on another system
2. Counters are made in the first block (“ICB”) with update counts for data blocks and for each level of index block
3. Each system reads the ICB and looks at the counts. The counts are updated on the system doing the update.
4. If a count has changed, then all the in-storage blocks of that type are marked as no longer valid

The RACF Database Bottom Line

- **RACF always reads/writes 4K blocks, using RBA addresses which are converted to CCHHRR**
- **When reading or writing these blocks, RACF performs its own serialization**
- **RACF uses a local cache of these blocks to avoid I/O**

The Whats and Whys of a z/OS Sysplex

The Basics of the z/OS Sysplex

- A **sysplex** refers to a tightly-coupled cluster of independent instances of the z/OS operating system. A distinct z/OS instance is often called an **image**.

- z/OS images in the sysplex communicate with other images in the sysplex using either channel-to-channel (**CTC**) connections or using the **Cross-system Coupling Facility (XCF)** component of z/OS

- Information about the systems in the sysplex, sysplex policies, the XCF configuration and general status information is stored in the **sysplex couple data set (CDS).** Additional couple data sets are used to store policy and function-related information.

- A **sysplex time reference** provides a consistent time value across the sysplex

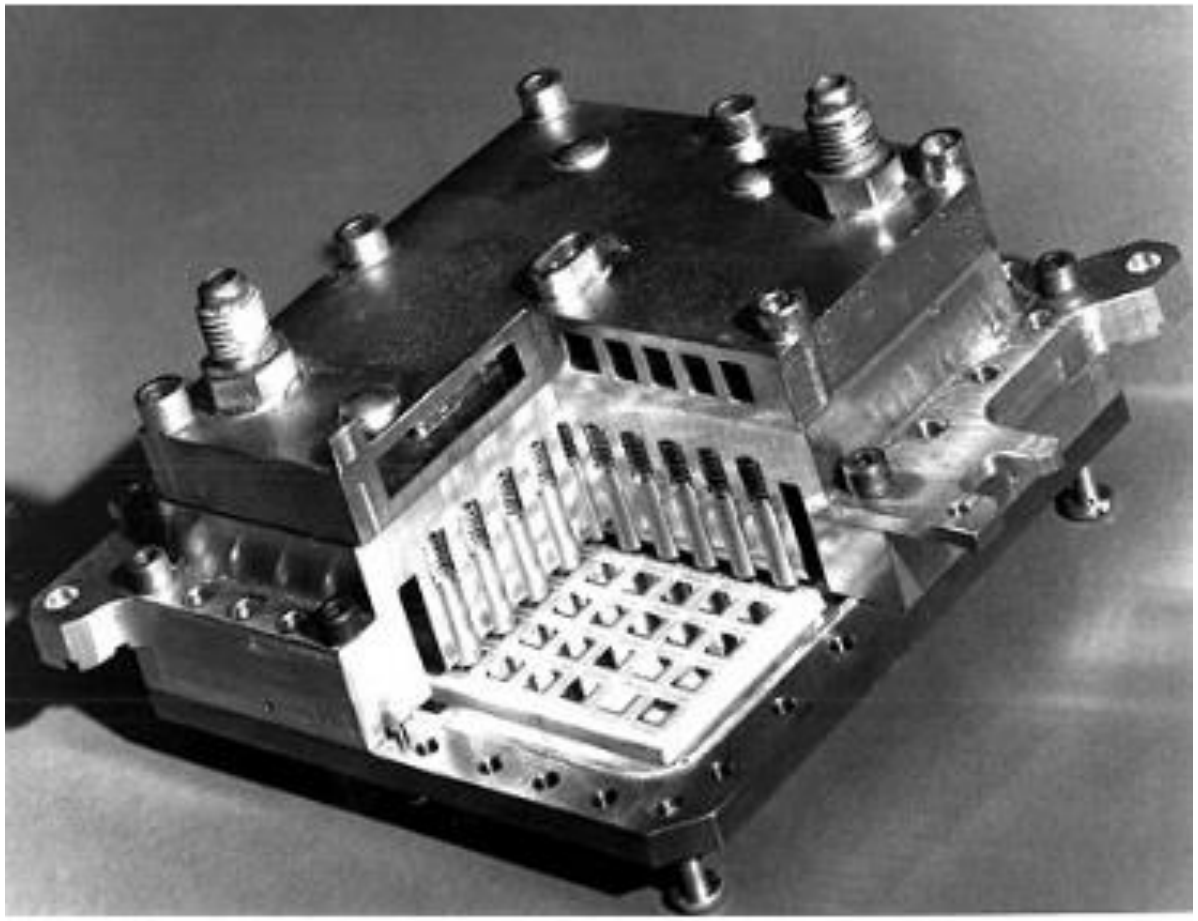
The Basics of the z/OS Sysplex…

- **A coupling facility (CF)** provides multiple z/OS images access to shared data

- **Sysplex Services for Data Sharing (XES)** provides services to access data in a coupling facility and to **automatically notify applications that their local copies are no longer valid.**

- **What do CF's provide?** Very high-speed access to:
  - **Lock structures**, for shared serialization processing
  - **List structures,** for shared list processing
  - **Cache structures**, for shared caches. This is the only type of CF structure used by directly by RACF.

Why Sysplex?



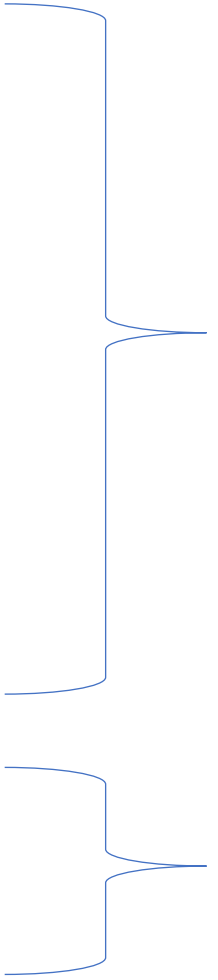
Why Sysplex?...

- **Improve availability:** Remove nearly all single points of failure from your application delivery environment and allow system upgrades without affecting availability by providing a single, consistent view of data (***including security data***)
- **Workload balancing across images:** Automatically distribute work (batch, OLTP, ….) around all systems in the sysplex
- **Reduce software costs:** Various pricing models based on aggregation of sysplex workloads
- **Provide nondisruptive, scalable growth:** Incremental, granular growth with near linear scalability

RACF and the Sysplex: Perfect Together!

RACF and the Sysplex

- **RACF uses the sysplex in several ways:**
  - Enable granular deletion of ACEE objects in the RACF VLF cache
  - Ensure a consistent RACF data set name table and RACF range table across the RACF sysplex communications group
  - Communicate specific RACF administrative commands across the members of a sysplex
  - Provide a high-speed and cross system data cache for the RACF



Using XCF services

Using XCF, XES, and CF services

The Many Modes of RACF in a Sysplex

- **“Base RACF” Mode**
  - This is the starting point. RACF uses RESERVE/RELEASE for I/O serialization
  - Only mode which should be used to share with systems outside of the GRSPLEX
- **Sysplex Communications Mode**
  - Requires enablement in ICHRDSNT or IRRPRMxx PARMLIB
  - A minimum of 50 resident data blocks will be acquired
  - Assures a consistent ICHRDSNT/ICHRRNG across the RACF sysplex communications
  - More precise ACEE VLF deletion
  - Enables RACF command propagation
  - RESERVE/RELEASE is (still) used for I/O serialization
  - If not enabled for RACF Data Sharing, this is called “non-data sharing” mode

The Many Modes of RACF in a Sysplex…

- **Data Sharing Mode**
  - Requires enablement in ICHRDSNT or IRRPRMxx PARMLIB
  - Requires enablement of RACF Sysplex Communications
  - RACF uses the CF as a systems-wide data cache
  - Elimination of RESERVEs for serialization
  - Ability to delete specific RACF local cache entries (instead of by type (data block, specific index block level)
- **Read-Only Mode**
  - Entered as a result of an operational error
  - Database updates are not allowed
  - After error is analyzed/fixed, RVARY DATASHARE can be issued
    - RVARY NODATASHARE takes all systems in the RACF data sharing group out of data sharing mode

More Precise Deletion of Security Caches

- **RACF sends out an event notification facility (ENF) signal when certain RACF administrative commands are issued.**
  - ENF listeners listen for this signal and take actions based on the signal content. For example, RACF listens for this signal to delete specific VLF cache entries affected by a RACF administrative command.

- **ENF signaling is used for many other notifications by RACF**
  - SETR RACLIST: ENF 62
  - CONNECT, REMOVE, ALTUSER REVOKE, DELUSER, DELGROUP: ENF 71
  - PERMIT, DEFINE, RALTER, RDELETE command that affects authorizations: ENF 79
- **Also used by Db2 and the z/OS Communications Server**

RACF Command Propagation

- **To ensure a consistent view of security, once in sysplex communications mode, RACF propagates these RACF commands to other members of the sysplex:**
  - **RVARY**
    - ACTIVE
    - INACTIVE
    - SWITCH
    - DATASHARE | NODATASHARE
  - **SETROPTS**
    - RACLIST | RACLIST REFRESH | NORACLIST
    - GLOBAL | GLOBAL REFRESH
    - GENERIC REFRESH
    - WHEN(PROGRAM) | WHEN(PROGRAM) REFRESH
- **Other SETROPTS commands, such as CLASSACT, GENERIC, AUDIT, GENCMD, STATS, and LOGOPTIONS are propagate using the ICB**

RACF Command Propagation...

- **The system upon which the propagated RVARY or SETROPTS command is issued is called the *coordinator***
- **The command is sent to all the systems (called *peers*) in the sysplex who have IPLed into the RACF sysplex communications group**
  - The coordinator waits for all the peers to execute the command before the command is considered completed.

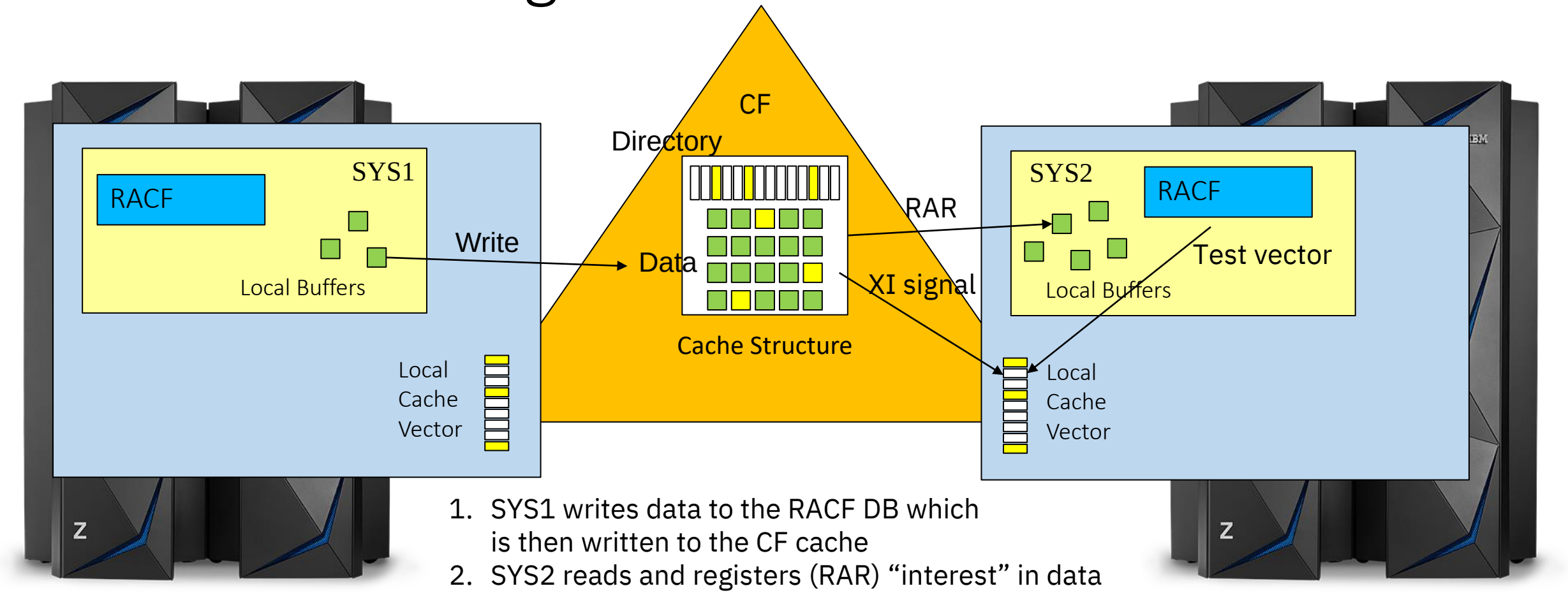
Single View of Security

- **A key design point of z/OS Data Sharing is that all applications on all systems see exactly the same data (including security data) at all times**
- **This means that when a SETROPTS RACLIST(REFRESH) is propagated through RACF's sysplex communications, every system must be loading the same profiles into storage**
- **What happens if profiles are altered and then a system is IPLed? It would see a different set profiles than the running systems, which means that it has a different view of security.**
- **The solution to the is RACGLIST, which:**
- **Stores a "frozen" copy of the *merged* profiles during a SETROPTS RACLIST, SETROPTS RACLIST(class-name) REFRESH or a RACROUTE REQUEST=LIST,GLOBAL=YES**
  - Stored in the RACGLIST class-name class-name_0000n profiles
- **The “propagated” SETROPTS RACLIST(class-name) REFRESHes use the data stored in the RACGLIST class-name profiles instead of reading the profiles themselves from the RACF database and thus see the "frozen" copy.**

RACF Data Sharing: Using the CF as a Cache

- **RACF Data Sharing allows you to define a coupling facility cache structure to act as a high-speed buffer for your RACF data sets**
- **RACF data sharing is entered by setting options in the ICHRDSNT or IRRPRMxx in PARMLIB.**
- **The minimum size for a primary RACF data set structure is:**
  - # of in-storage buffers x 4K +
    .1 x #instorage buffers x 4K x #systems in the sysplex
- **The minimum size for the backup RACF data set structure is:**
  - .2 x primary structure size
- **Forget the math, use the CFSIZER at:**
  https://www.ibm.com/support/pages/cfsizer
- **The effective maximum is the size of your RACF data set.**

RACF's In-Storage Buffers and the CF



1. SYS1 writes data to the RACF DB which is then written to the CF cache
2. SYS2 reads and registers (RAR) "interest" in data
3. SYS1 updates that data entry in cache structure
4. CF sends XI signal to update local cache vector on SYS2 to indicate that the copy of the data in the local buffers from step 2 is no longer current
5. Before using the local copy, SYS2 tests the local vector, discovers the need to refresh its local copy before proceeding

Appendix 1: Defining the RACF Data Sets in IRRPRMxx

IRRPRMxx PARMLIB Member

```
DATABASE_OPTIONS
SYSPLEX(NOCOMMUNICATIONS | COMMUNICATIONS | DATASHARING)
DATASETNAMETABLE
ENTRY
PRIMARYDSN(data-set-name)
BACKUPDSN(data-set-name)
UPDATEBACKUP(ALL | NONE | NOSTATS)
BUFFERS(value)
RANGETABLE
START(start-value [CHAR | HEX]) ENTRYNUMBER(entry-
sequence-number)
```

RACF® and the Parallel Sysplex

A RACFers View of the z/OS® Parallel Sysplex®

Mark Nelson, CISSP®, CSSLP®
markan@us.ibm.com
IBM®

14 October 2021