CICS and Recovery

Andy Wright

CICS L3 Change Team

IBM Hursley

Virtual CICS User Group meeting – Tuesday 14th November 2023

Notices and disclaimers

- This presentation is provided by IBM Corporation. Copyright © IBM Corporation, 2023. Redistribution is prohibited.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.

- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

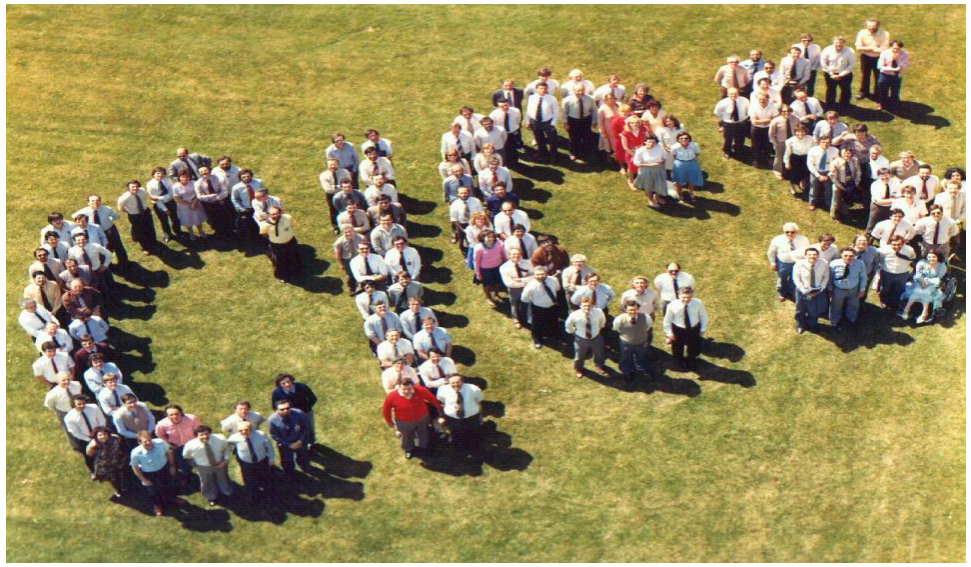
- IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM Z, IBM z Systems, IBM z13, IBM z14, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplication, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

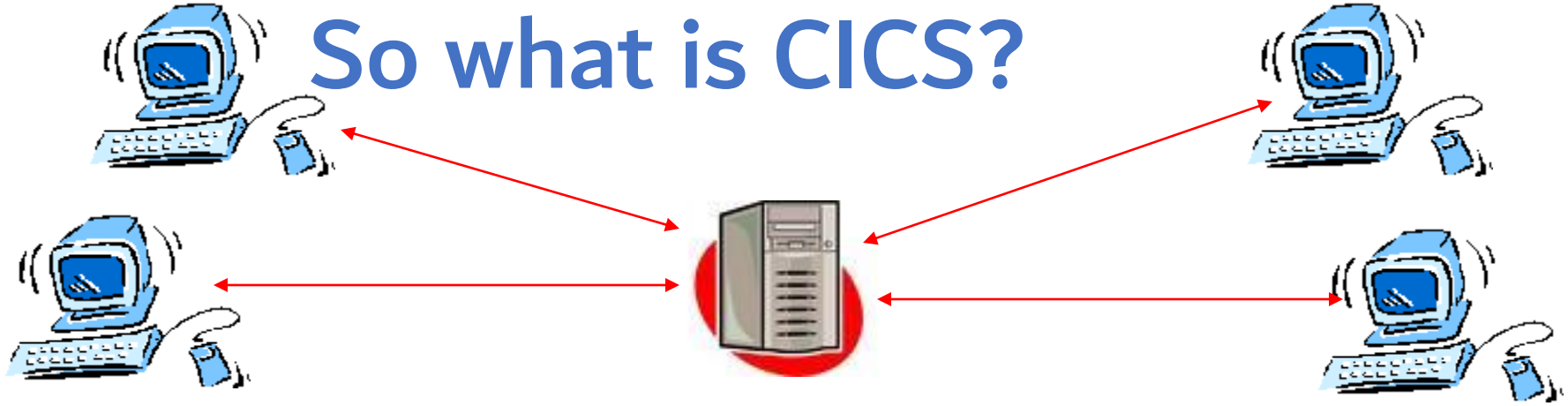
Some background

- I work at IBM Hursley Park
  - near Winchester, Hampshire
  - IBM software development Lab
    - e.g. CICS, WebSphereMQ, etc
  - Hursley House was built in the 1720s
    - used as hospital in WW1
    - used by Vickers in WW2
      - Spitfire development
  - IBM arrived in 1958
    - many new buildings added
  - www.ibm.com/

Some Hursley sights



So what is CICS?

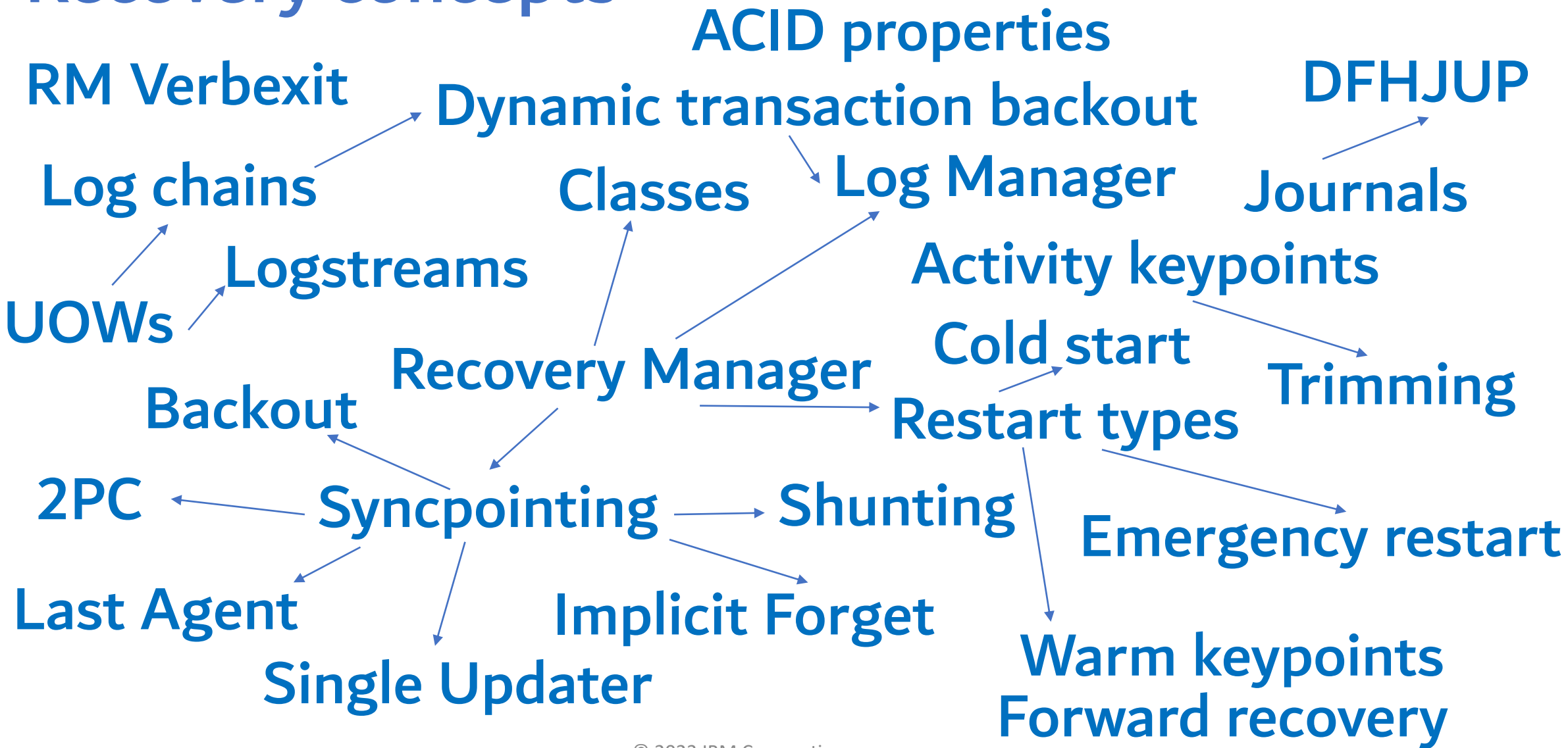


- the world's premier transaction processing software
  - originally developed in 1968
  - many new releases developed over the years
  - acts as middleware between the operating system and its users
  - used by many Fortune 500 companies
    - insurance, stock control, reservations, banking (e.g. ATMs), government, commerce (e.g. market trading), retailing …
  - runs user programs in Java, C, assembler, COBOL, PL1
    - the latest versions support SOAP, Web Services, JSON …

Recovery concepts

Where to start..?

Recovery concepts



Atomic properties of transactions

- ***Bernstein, Hadzilacos, Goodman***
- "The goal of concurrency control and recovery is to ensure that transactions execute atomically, meaning that
  - each transaction accesses shared data without interfering with other transactions, and
  - if a transaction terminates normally, then all of its effects are made permanent; otherwise it has no effect at all." *

- * P.A. Bernstein, V. Hadzilacos, N. Goodman - Concurrency Control and Recovery in Database Systems, 1987

ACID properties

- **Atomic**
  - All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.
- **Consistent**
  - Data is in a consistent state when a transaction starts and when it ends.
- **Isolated**
  - The intermediate state of a transaction is invisible to other transactions.
- **Durable**
  - After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure.

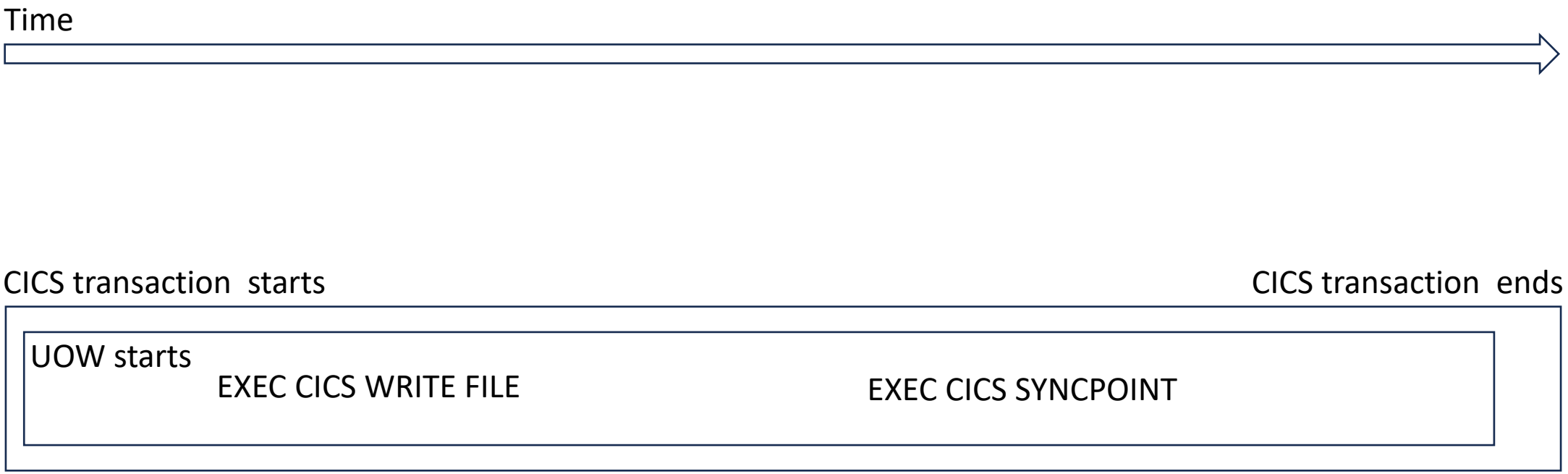
But just what *is* a transaction??

- A complete set of recoverable operations performed by a transaction processor such as CICS
- Adherence to ACID properties, so work is committed or all backed out, etc
- Such indivisible sets of operations are known as *transactions* in the industry
- In CICS they are referred to as *Units of Work (UOWs)*
  - *Other terms include Units of Recovery (UORs)*
- A CICS "transaction" is the environment for an application performing work
  - It can contain one or multiple UOWs
- A CICS transaction is **not** the same as a Unit of Work!
  - CICS Transaction Manager handles transactions
  - CICS Recovery Manager handles Units of Work
- You often hear CICS transactions called tasks – the terms get interchanged

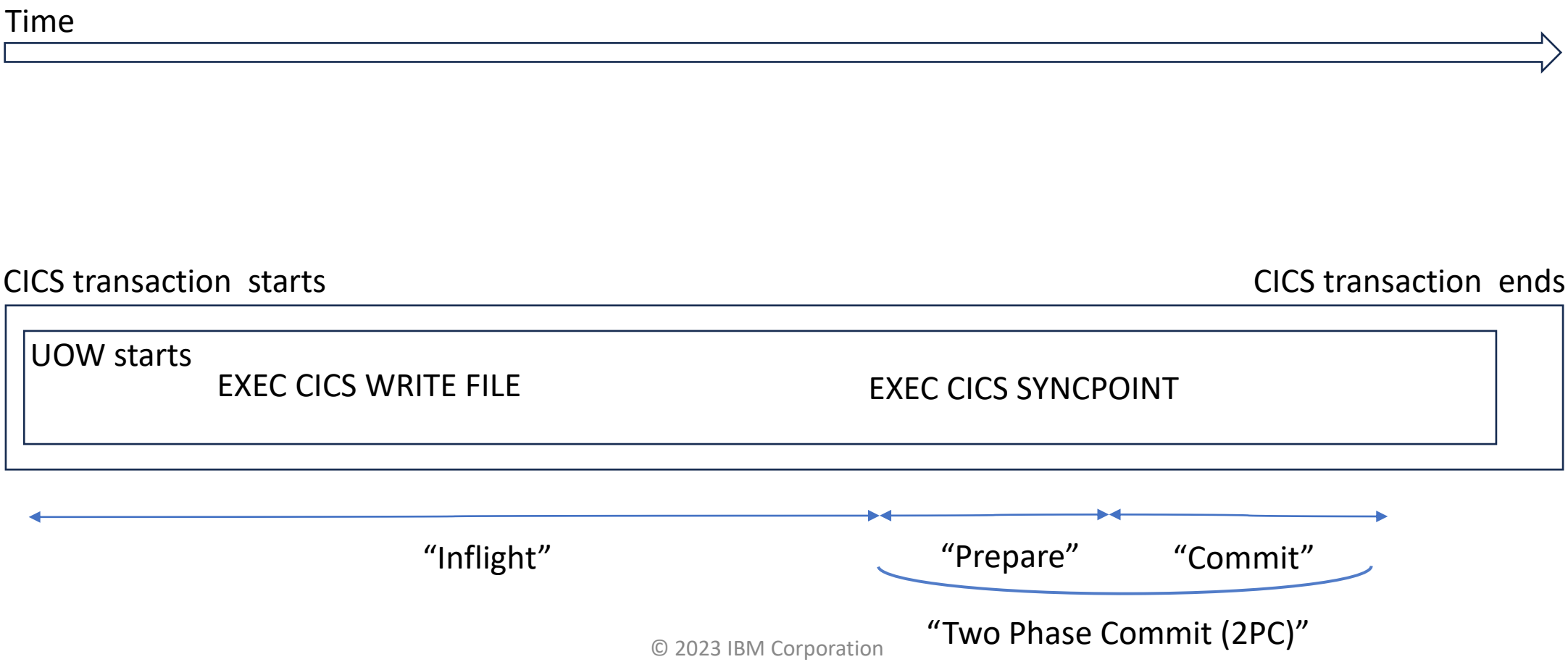
Units of Work and syncpoints

- A Unit of Work includes all the recoverable operations that a task wants to perform in CICS as one logical set of work
  - Updates to files
  - Changes to data bases
  - Reads and writes to queues of data
  - etc
- When the task wants to commit these it can either end (a final EXEC CICS RETURN) or commit them (EXEC CICS SYNCPOINT)
  - The end of task EXEC CICS RETURN drives an implicit syncpoint
  - The EXEC CICS SYNCPOINT ends this UOW and begins another

A stroll through a Unit of Work



A stroll through a Unit of Work



A comparison with a wedding...

- A Unit of Work and its syncpoint can be loosely compared with a wedding...
- The couple meet
  - *(The task starts, its UOW starts)*
- They spend time together
  - *(The UOW carries out recoverable operations)*
- They decide to get married
  - *(The task issues EXEC CICS SYNCPOINT and the UOW enters syncpoint processing)*
- People can object to the wedding or not)
  - *(The UOW prepares itself – Phase 1)*
- The wedding goes ahead
  - *(The UOW logs the fact it is committing, then does the commit work – Phase 2)*
- **Or...** the wedding is stopped because someone objected...
  - *(The UOW logs the fact it is backing out, then does the backout work)*

Stretching the comparison…

- The person conducting the wedding = CICS Recovery Manager
- The paperwork and wedding certificate = CICS Log Manager

- Recovery Manager:
  - Maintains the status and progress of the Units of Work (UOWs)
  - Coordinates recoverable changes made by local resource managers
  - Coordinates recoverable conversations with other remotely connected systems
  - Uses system log information to reconstruct UOWs over restarts

- Log Manager
  - Writes log data to the CICS system logs DFHLOG and DFHSHUNT
  - Writes user data to user logs
  - Maps log definitions to z/OS logstreams

Logs, Logstreams and Journals

- CICS system log data written to DFHLOG (primary logstream)
  - DFHSHUNT (secondary logstream) also provided, for “long duration” data
- Logs maps to a logstream
  - Owned by the z/OS Logger subsystem
  - Reside on CF or DASD
- Other journals are for “general logs”
  - Autojournalling of files
  - Forward recovery
  - Replication logging
  - Terminal I/O logging
  - Security audit logs

Backwards and forwards recovery

- CICS uses the system log for backward recovery
  - “Before images” of resources being changed
  - Used to restore recoverable resources to a previously committed state
- CICSVR (for example) provides forward recovery
  - Using “After images” of VSAM records that were changed
  - Used to restore data sets to the state they were in before being damaged
- General logs may not used for recovery purposes at all

Restart types

- INITIAL
  - A completely fresh run of CICS
- COLD
  - Local resources are reinstated from the CSD; remote obligations are preserved
- AUTO
  - WARM
    - The system is restored to its previous state from before a controlled shutdown
  - EMERGENCY
    - The system is restored to its previous state from before an uncontrolled shutdown
- The restart types make use of the system log in different ways

Log chains

- CICS writes out blocks of records to the system log
- Each block contains records for 1 to many log chains
- Each log chain is associated with a UOW
- The log can be read sequentially
  - Backwards one record at a time
- A log chain can be read individually
  - Backwards one record on the chain at a time

Log chains

Block n-1

The past

"Now"

Time

Log chains

Block n-1

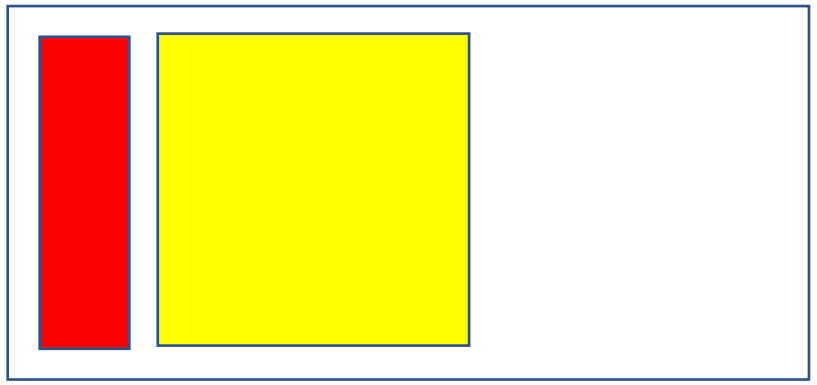
The past

"Now"

Time

Log chains

Block n-1



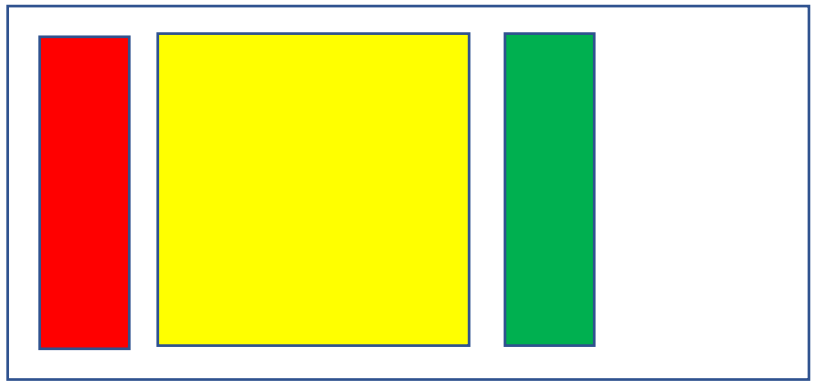
The past

"Now"

Time

Log chains

Block n-1



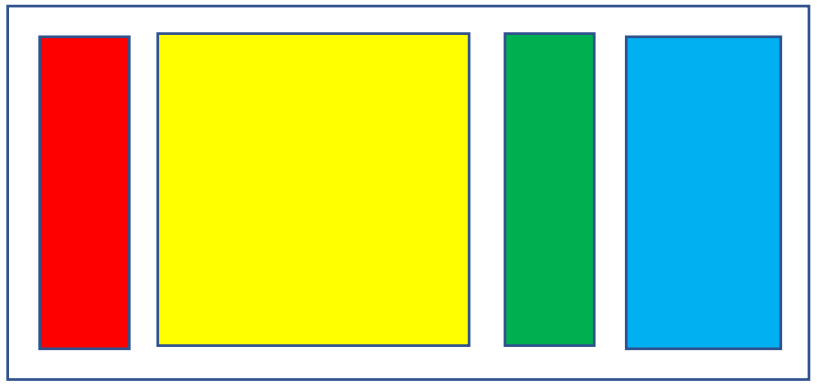
The past

"Now"

Time

Log chains

Block n-1

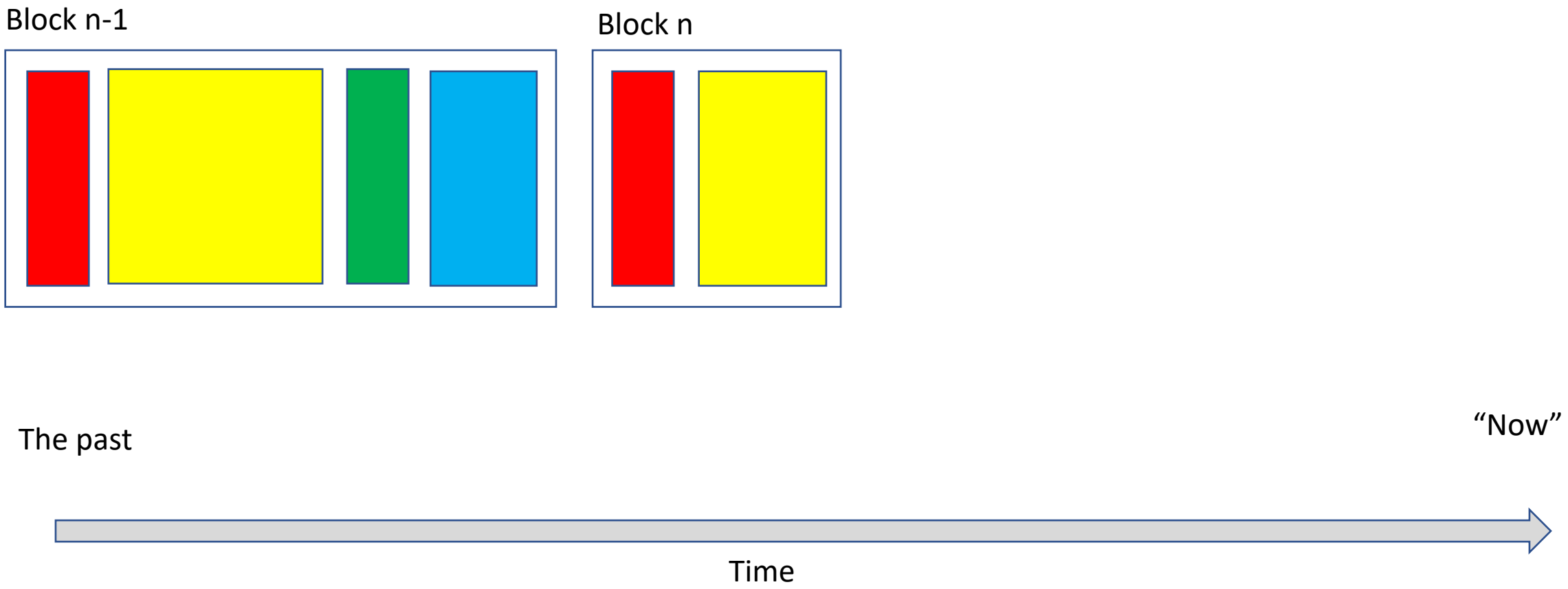


The past

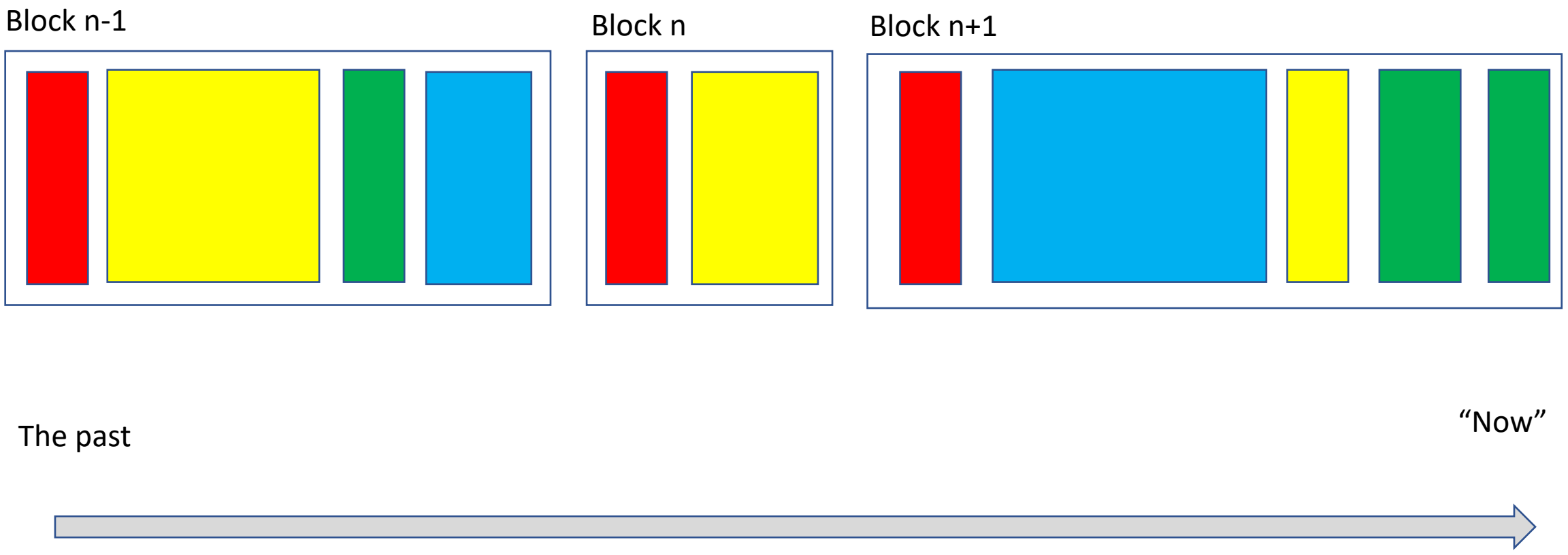
"Now"

Time

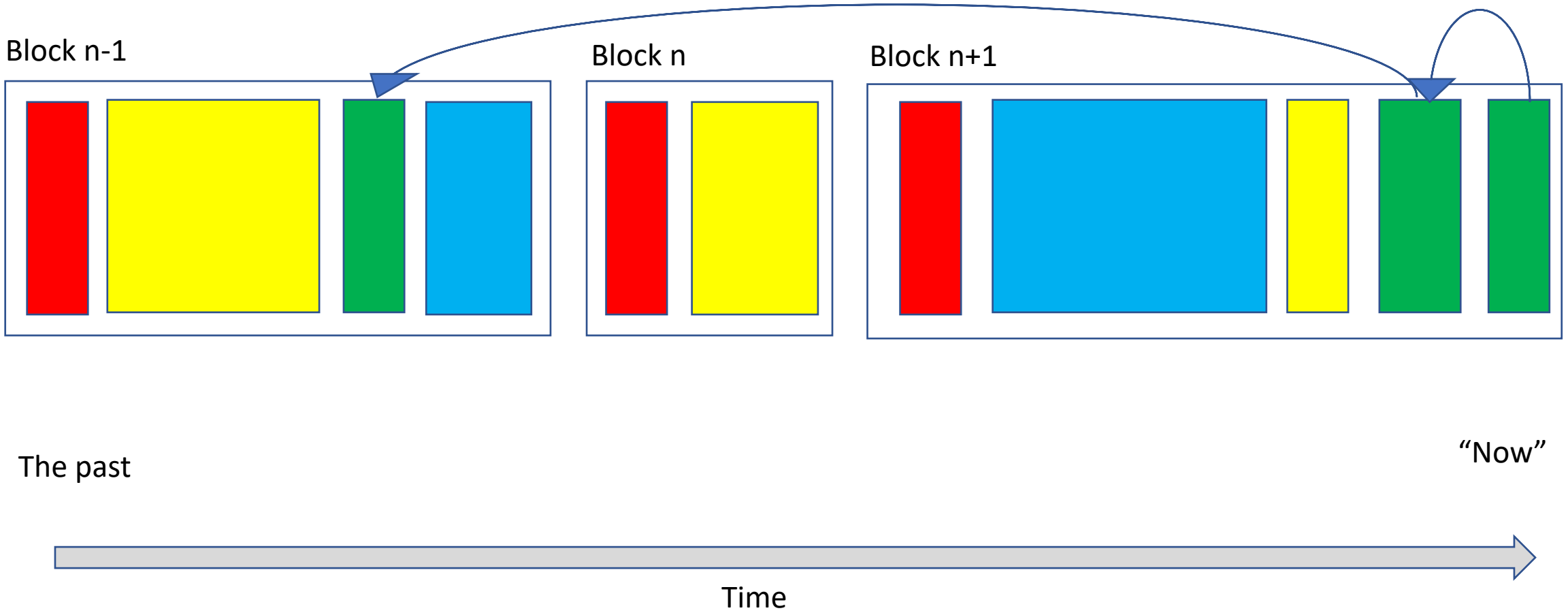
Log chains



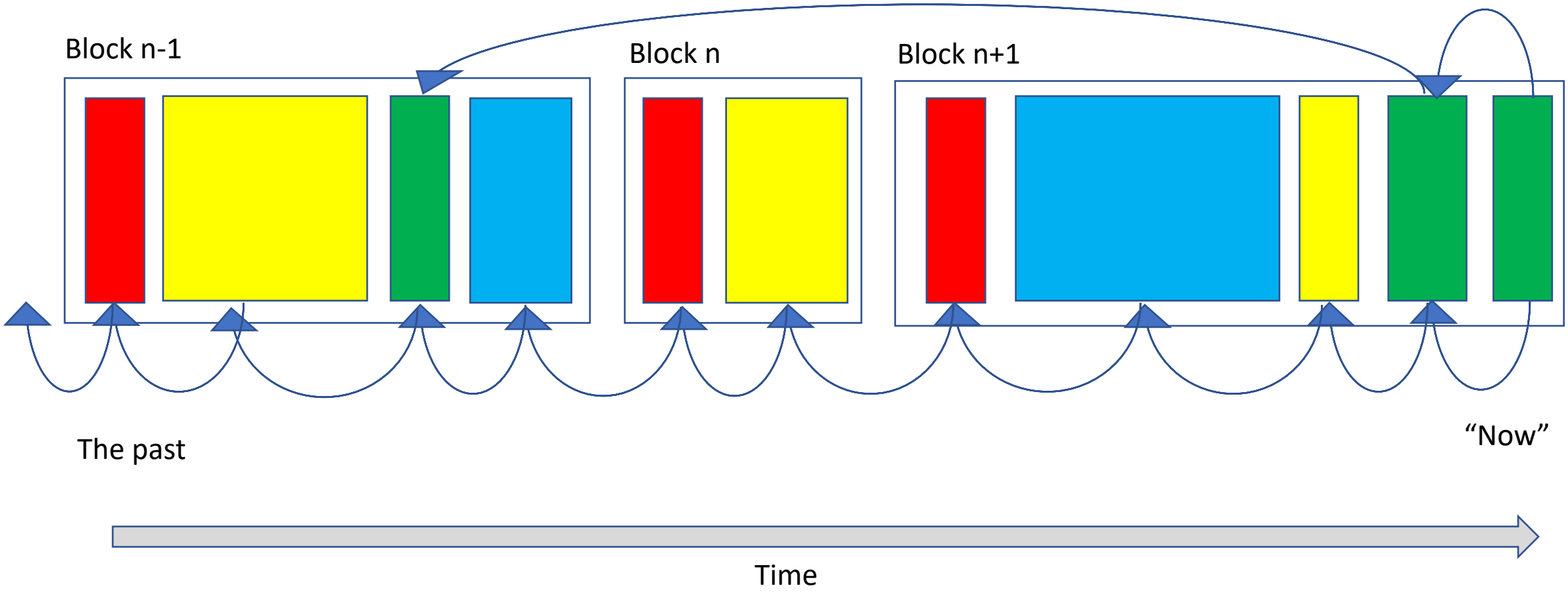
Log chains



Log chains



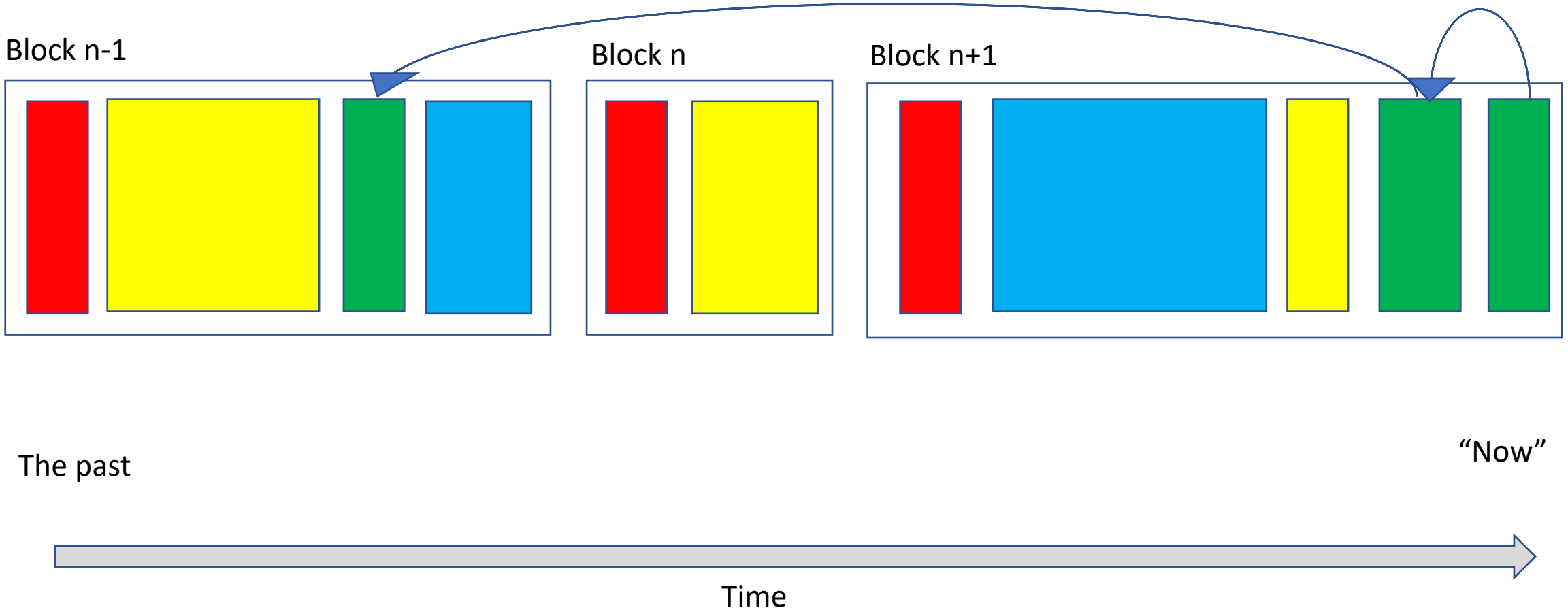
Log chains



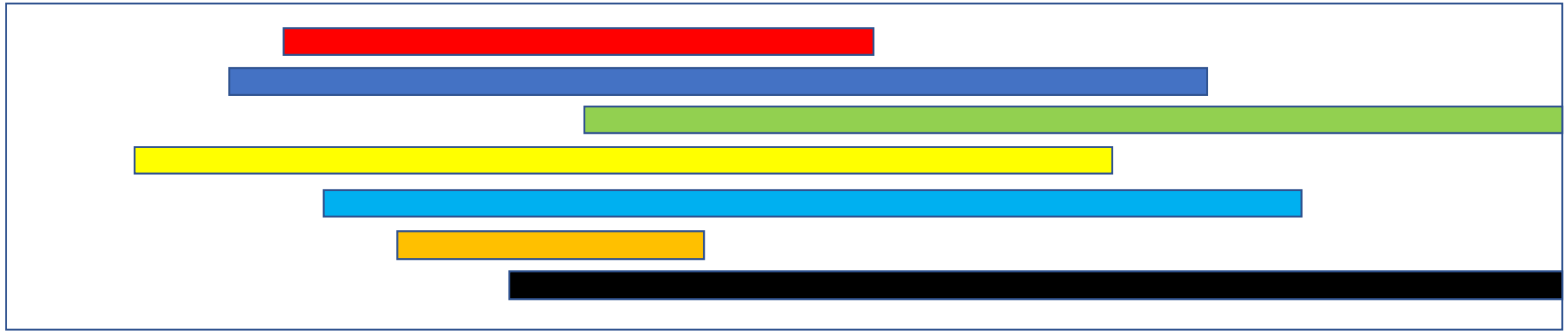
DTB – reading one log chain

- Driven as the result of
  - An abend
  - An EXEC CICS SYNCPOINT ROLLBACK
- Recovery Manager calls Log Manager to read back the UOW log chain
- Records are presented to the appropriate client to back out changes

DTB – reading one log chain



Chain and Stream history points

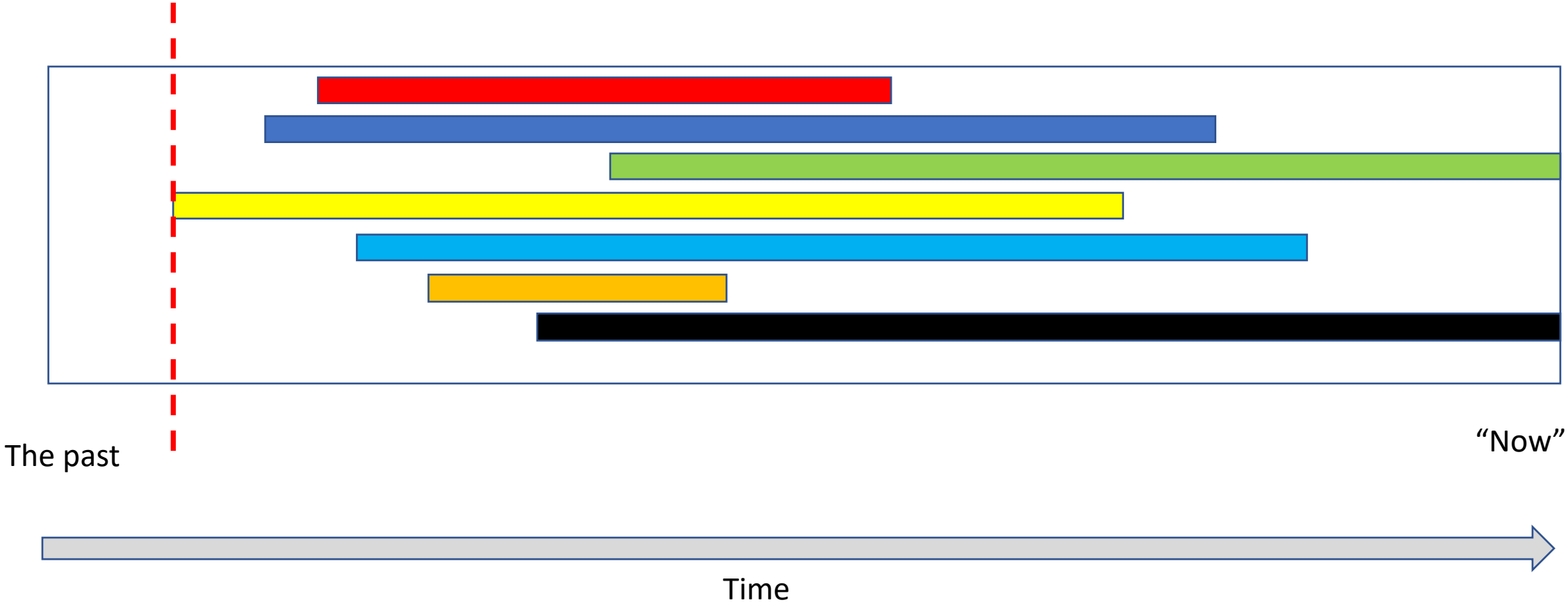


The past

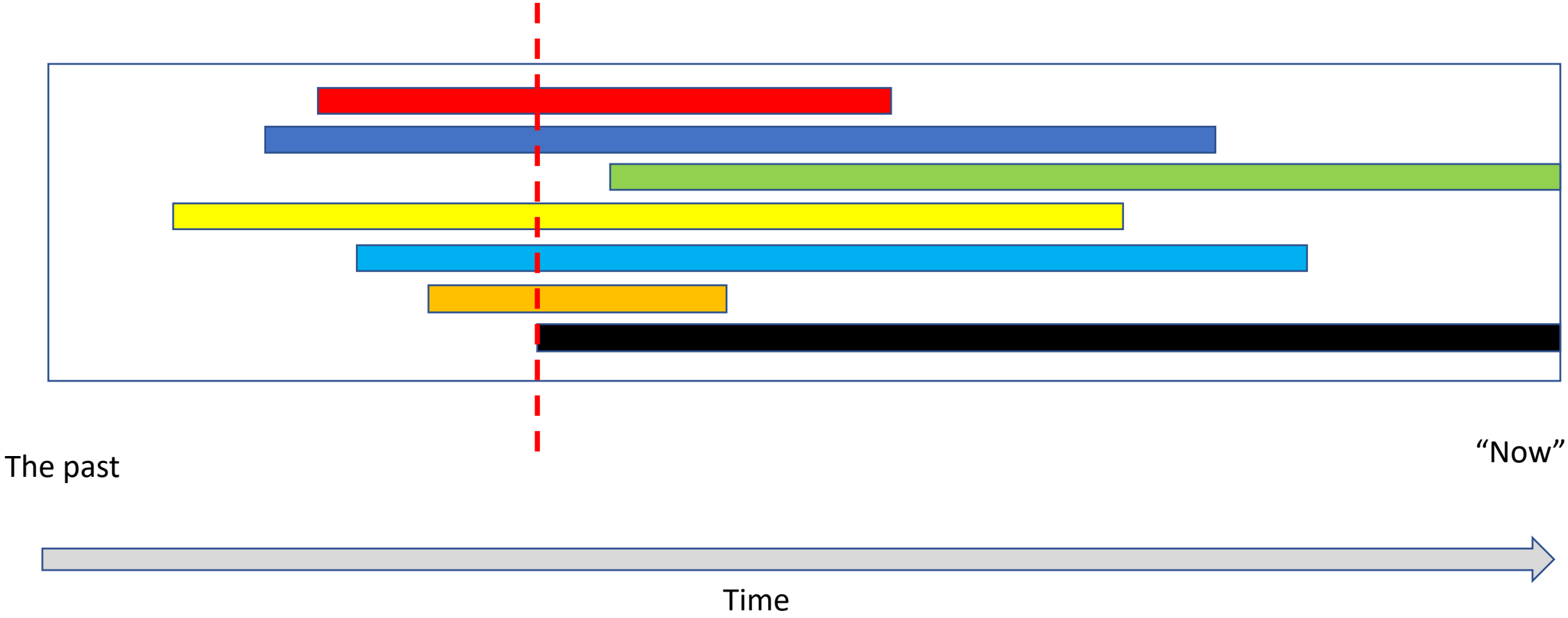
"Now"

Time

Chain and Stream history points



Chain and Stream history points



Two phase commit sequence

- Driven by EXEC CICS SYNCPOINT, EXEC DLI TERM, etc
- Recovery Manager drives 2PC processing for remote coordination
- Phase 1 is **Prepare**
  - The local resources are polled to vote
  - The remote systems are polled to vote
  - If clients vote yes they must be prepared to commit forwards if told to later
    - Remote systems enter an **Indoubt** phase until told which way to commit
- CICS logs the fact the UOW is committing forwards
- Phase 2 is **Commit**
  - The remote systems are called to commit
  - The local resources are called to commit

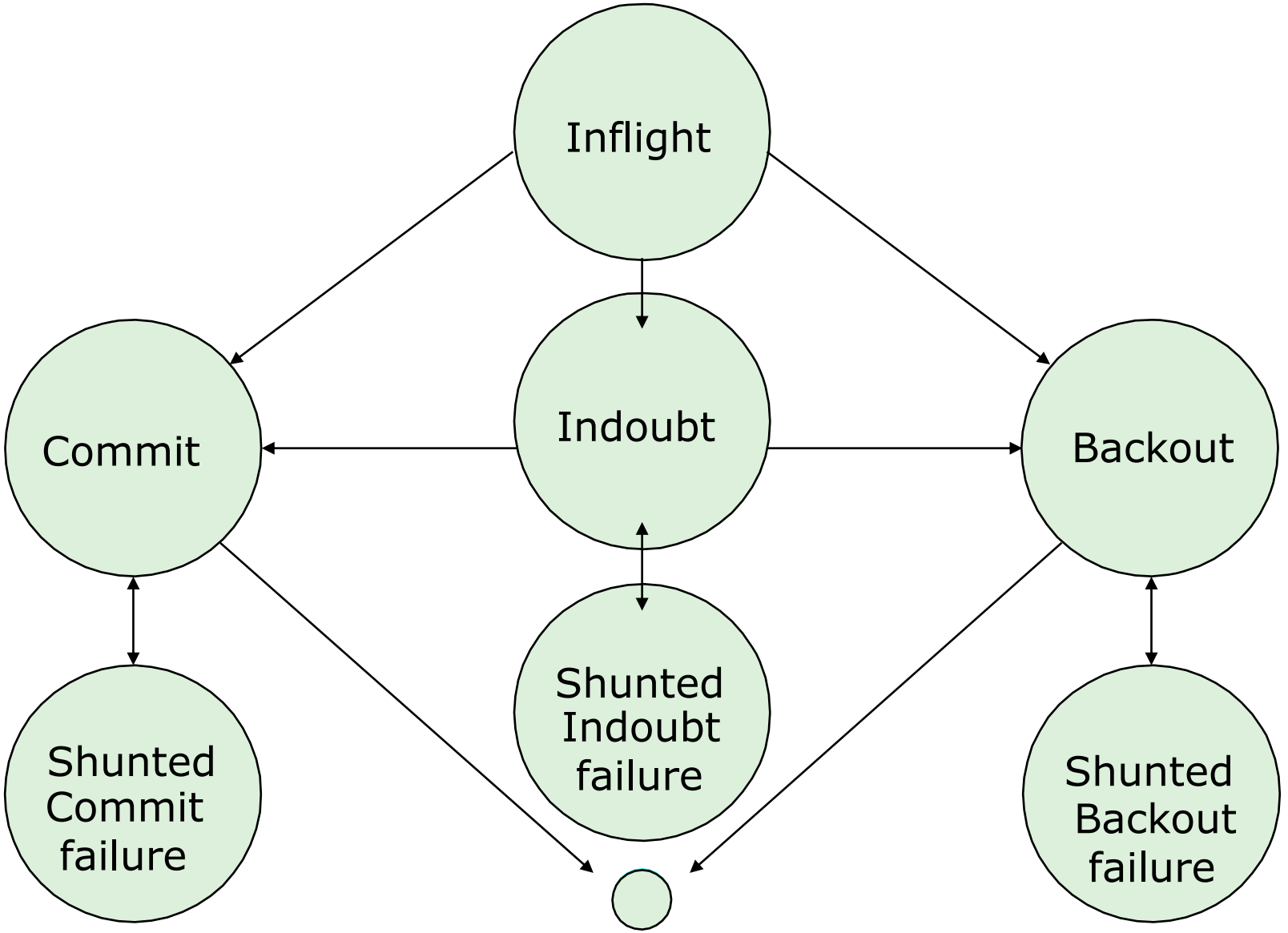
Backout (commit backwards) sequence

- Driven by EXEC CICS SYNCPOINT ROLLBACK, an ABEND, etc
- Backout is a single phase process to commit backwards
  - The remote systems are called to commit backwards
  - The local resources are called to commit backwards
- Resource managers do not need to prepare to backout
  - Note - “Any resource manager worth its salt should default to backout”
- CICS logs the fact the UOW has committed backwards

Shunting and unshunting

- CICS can shunt UOWs that fail at key moments during a syncpoint
- A shunted UOW awaits resolution of an indoubt failure, a commit failure or a backout failure
  - Recovery manager attempts to complete a shunted UOW when the failure that caused it to be shunted has been resolved
- Shunting releases resources such as terminals, user programs, working storage, etc, normally by abending the transaction
  - CICS retains locks on recoverable data until it can attempt an unshunt
- Shunted UOW log data tends to persist for longer durations, so is moved from DFHLOG to DFHSHUNT
  - **Note** - DFHSHUNT is not just for shunted UOW's log records

Shunted Unit of Work state



Last Agent

- Recovery Manager invokes every participating system to prepare
- The final one can be called **directly** to commit
- CICS passes the coordination role to this system
  - **Note** – the old coordinating system is now Indoubt
- The new coordinator will then either commit forwards or backwards (backout)
- This response is returned to the old coordinator which does the same
- It then instructs its own participants to do likewise
- **Note** – Last Agents may be daisy-chained

Single Updater

- An optimized form of Last Agent
- When the coordinator passes the role to a Last Agent, and no local logging has occurred up to this point
- This means the old coordinator is not indoubt
  - It doesn't care which way it commits

Implicit Forget

- The coordinator could wait for a response to each commit
  - This would delay processing
- Implicit Forget avoids this
- Recovery Manager retains the UOW and its state information
  - Decouples from the communication layer object (e.g. session)
  - The session can be reused by another task
  - When data is sent down the session, the code notifies CICS of this
  - CICS can now forget the remote system (and potentially the old UOW)

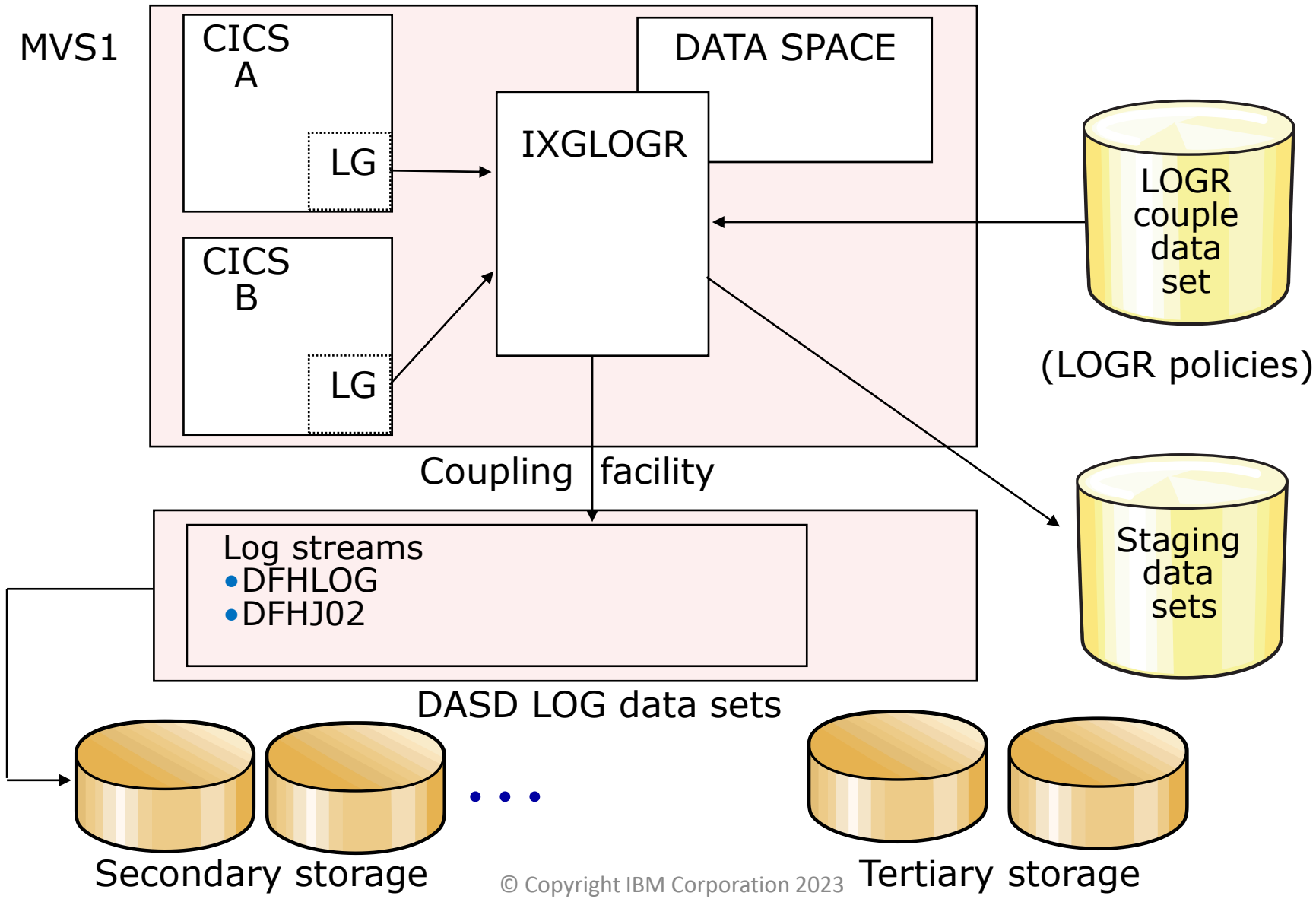
Keypointing

- Driven by AKPFREQ being reached
- Causes a CSKP system transaction to be attached
  - Note - You cannot attach CSKP by entering it at a terminal
- Keypointing will write state data to DFHLOG
- It also will trim DFHLOG and DFHSHUNT
- Important messages:
  - **DFHRM0205** *date time applid* **An activity keypoint has been successfully taken.**
  - **DFHLG0743** *date time applid* **Tail of log stream** *lsn* **deleted at block id** *X'blockid'*.
  - **DFHLG0760** *date time applid* **Log stream** *lsn* **not trimmed by keypoint processing. Number of keypoints since last trim occurred:** *trimnum*. **History point held by transaction:** *transid*, **task number:** *trannum*.

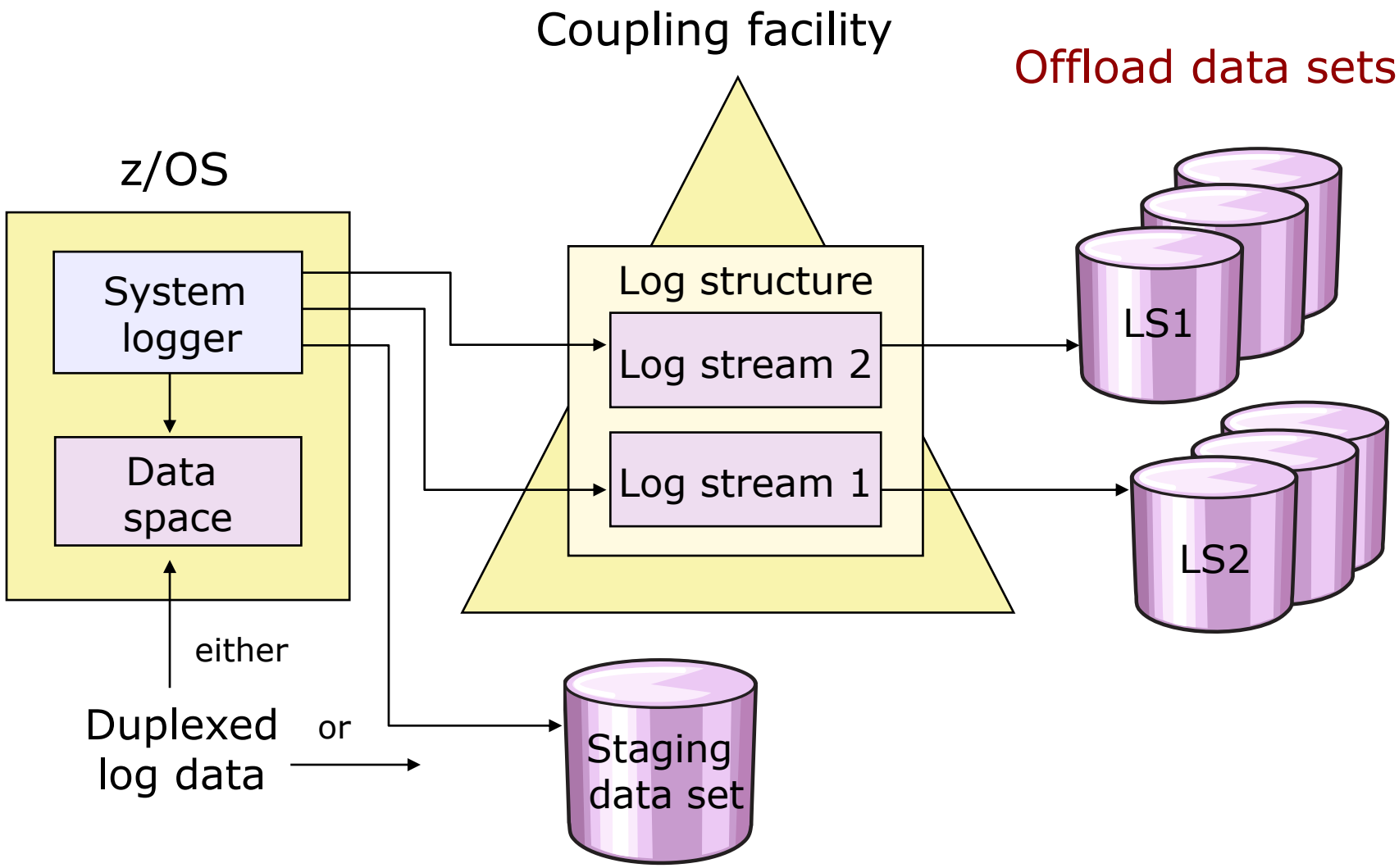
Log trimming

- Keypoints are an opportunity for CICS to delete unwanted log data
- Log Manager issues delete calls to the z/OS Logger
  - These logically delete ranges of the DFHLOG and DFHSHUNT logstreams
- When the logstream reaches its HIGHOFFLOAD percentage, the z/OS Logger will perform offload housekeeping work
  - Logically deleted log blocks are physically deleted
  - If necessary, remaining log data is moved to secondary storage (offload datasets)
- This continues until the LOWOFFLOAD percentage is reached
- Offload housekeeping is a good thing
  - Offload I/O to DASD is not so good
- Note – check for DFHLG0760 messages (or no DFHRM0205 messages)
  - Evidence that log trimming is not taking place

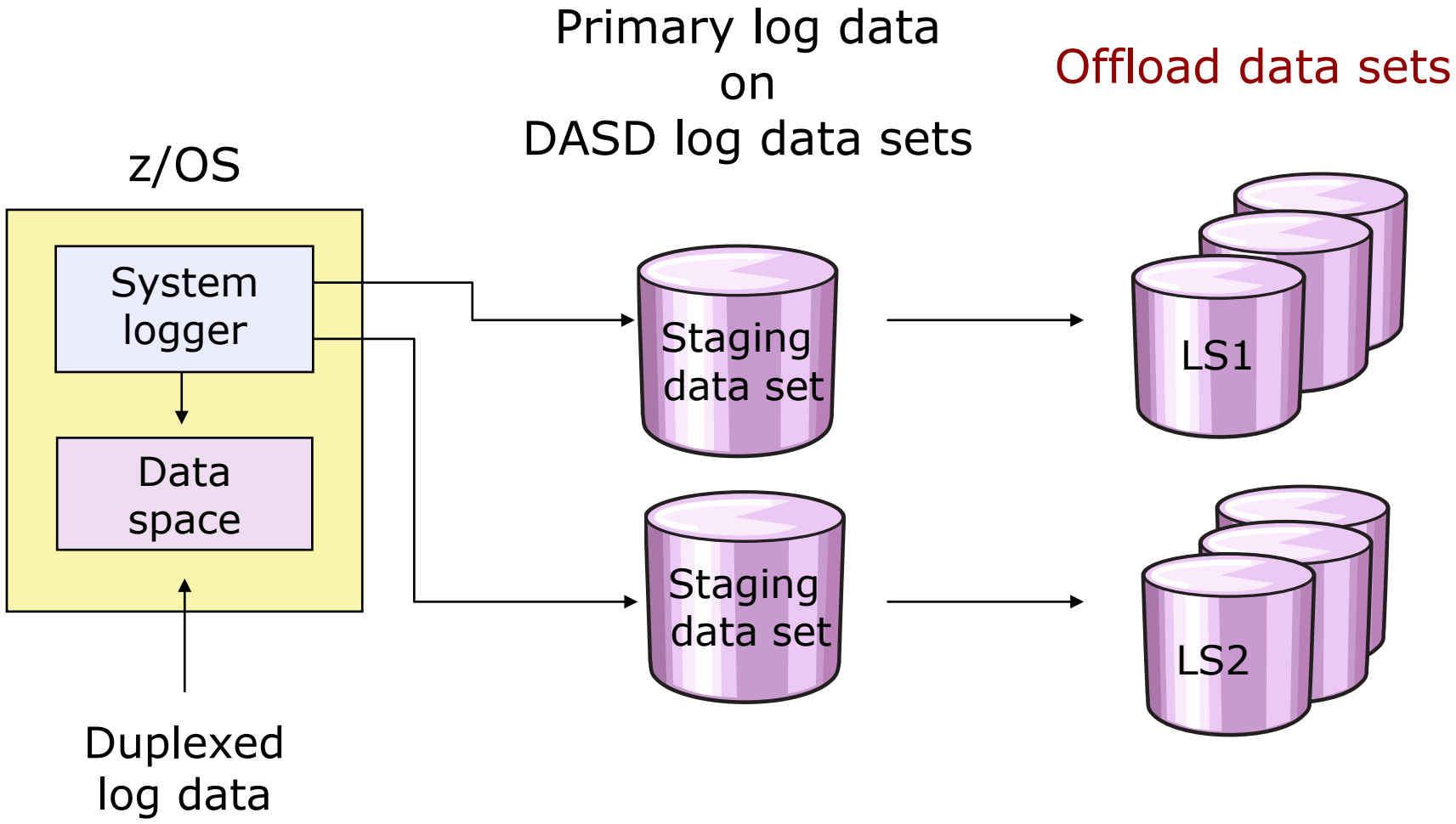
z/OS System Logger overview



Log stream storage for CF logging



Log stream storage for DASD-only logging



Log analysis

- CICS system logs are written to and read back by CICS
- CICS general logs are only written to by CICS
- DFHJUP is provided to read back logstreams for offline purposes
  - You can PRINT or COPY log data
- DFHJUP provides COMPAT41 to return log data in “old-style format”
  - Some batch utilities still exist that work with such old-style CICS log data