

CA IDMS™ 2-Phase Commit

IUA Workshop

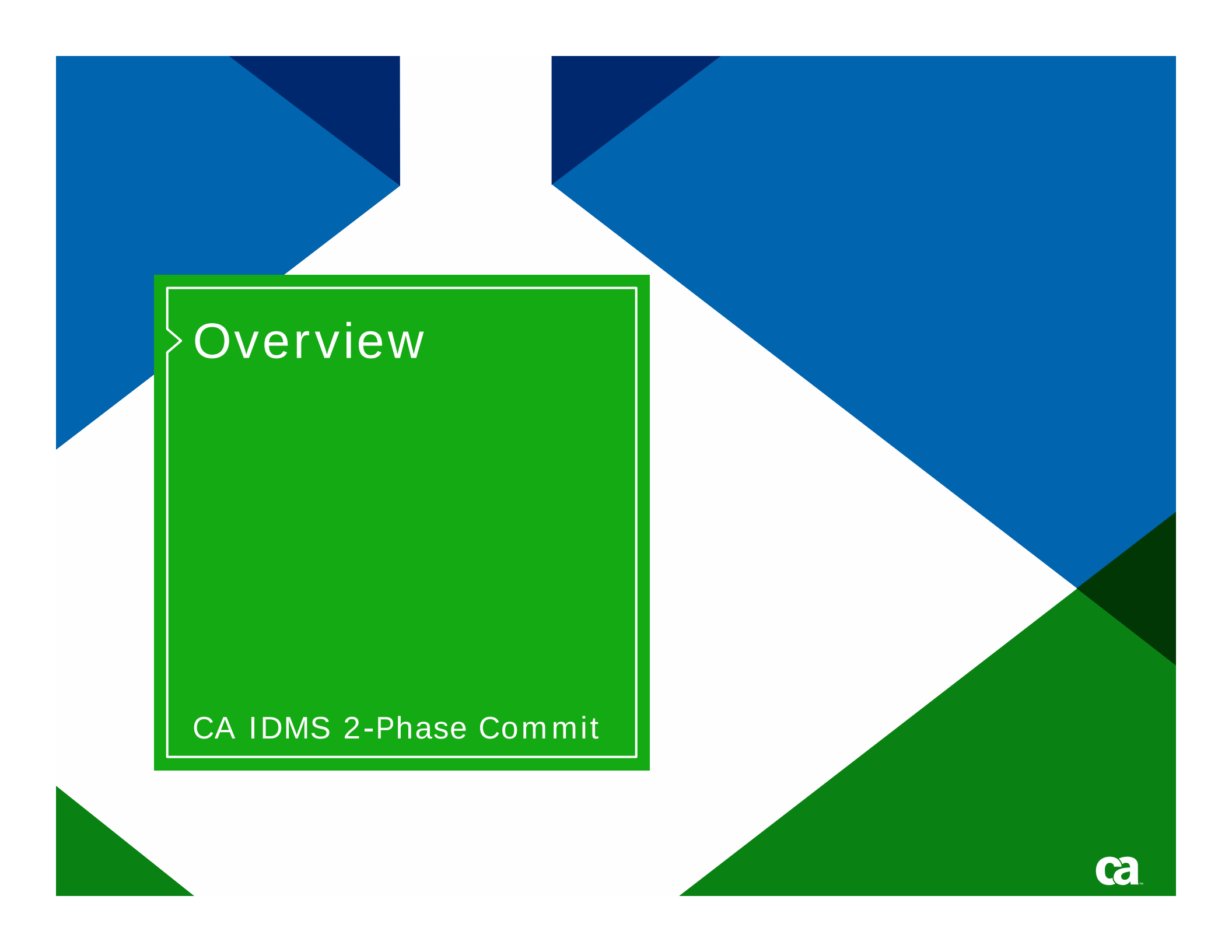
October 2007

Abstract

This session describes how the 2-phase commit capabilities within CA IDMS™ enable coordinated commit and recovery across a distributed transaction. It defines what is meant by a 2-phase commit operation and describes the specifics of its implementation within CA IDMS. This presentation is appropriate for Database Administrators and others who would like to know about the ability for CA IDMS to participate in work units that span multiple resource managers.

Agenda

- > Overview
- > What Happens During a 2-phase Commit?
- > Journaling and 2-phase Commit
- > Resynchronization
- > Completing Transactions Manually
- > Operational Considerations



> Overview

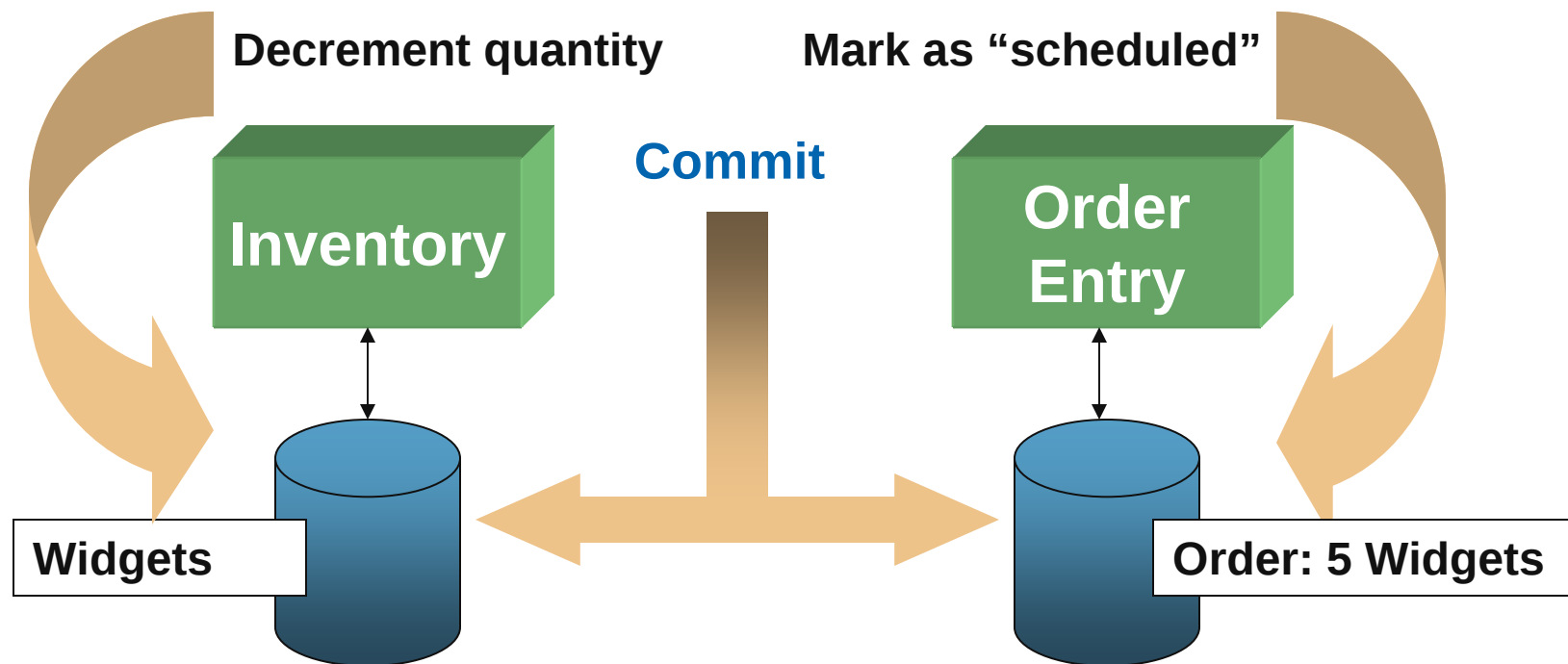
CA IDMS 2-Phase Commit

What is a 2-phase commit?

**A commit operation that ensures that
all changes made within a
distributed unit of work
are either applied or backed out.**

Distributed Unit of Work

Transaction: Schedule Delivery



Terminology

- > Transaction manager: a software component responsible for commit and backout operations
- > Resource manager: a software component responsible for a recoverable resource such as a database
- > Coordinator: a transaction manager that directs a 2-phase commit operation
- > Participant: a resource manager involved in a 2-phase commit operation

2-phase Commit Processing

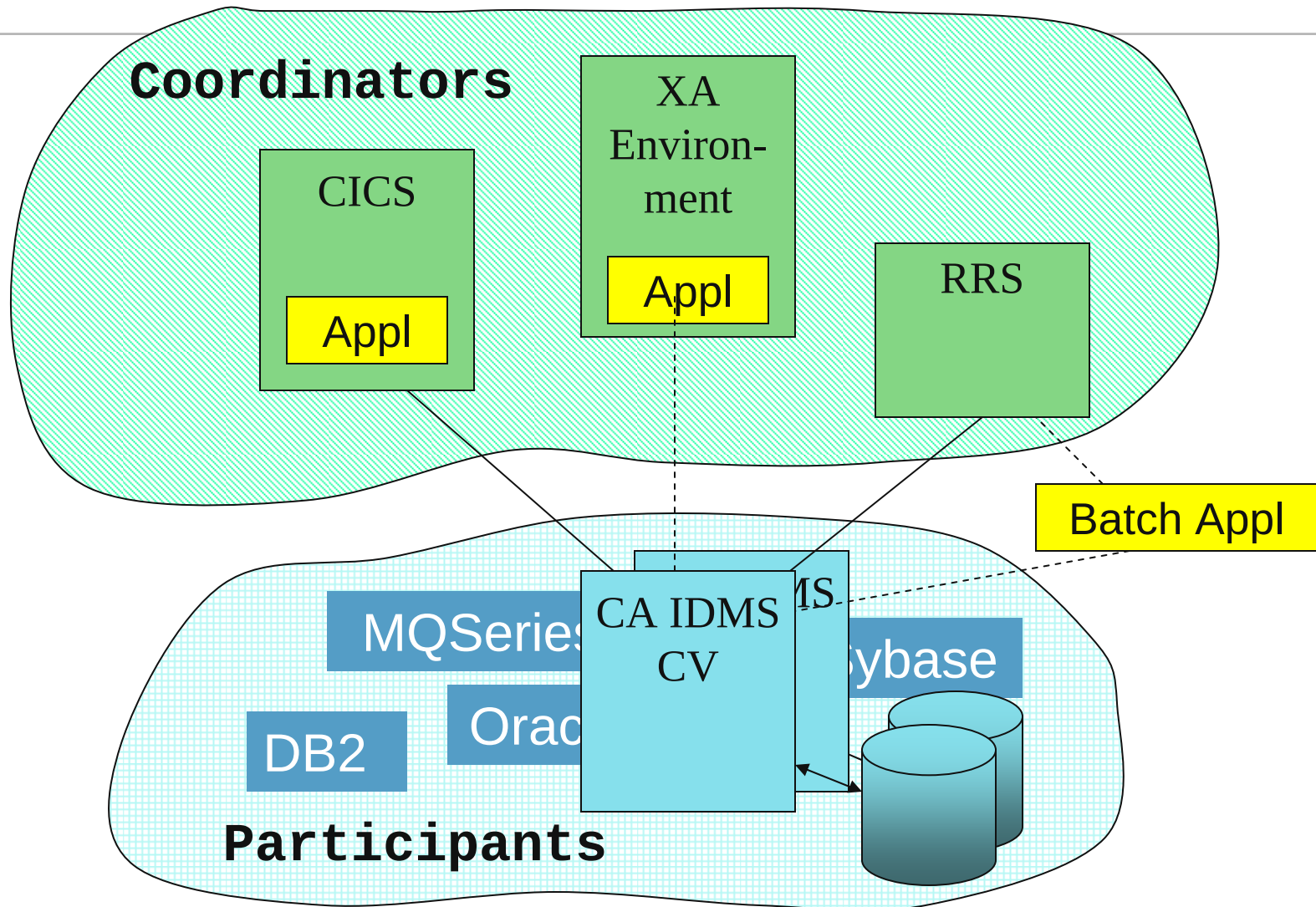
Phase 1

- > Coordinator receives commit request
- > Participants prepare their resources for commit
 - Write updates to database
 - Journal “prepared” state
- > Coordinator journals “committed” state

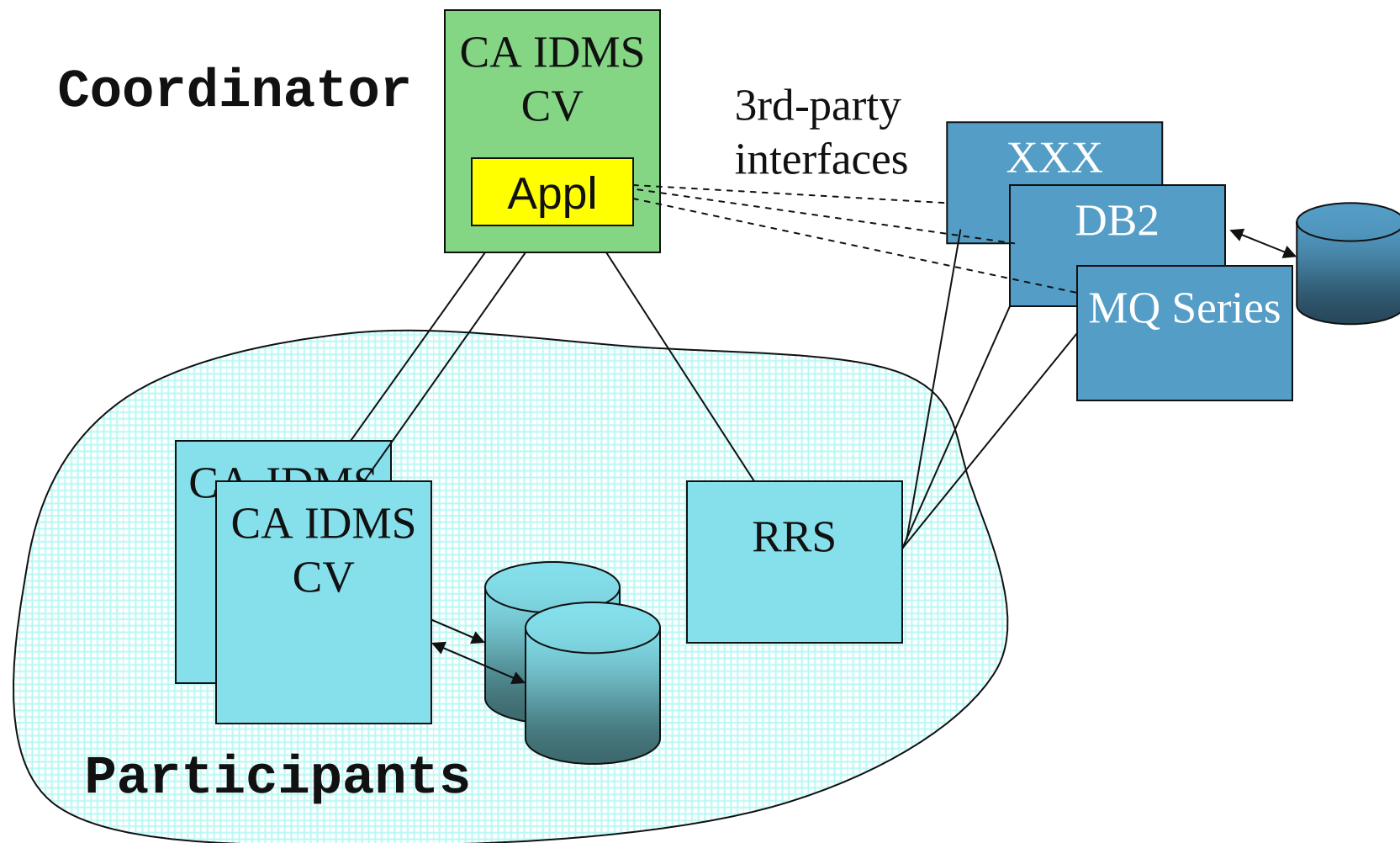
Phase 2

- > Participants complete commit operation
 - Journal “committed” state
 - Release update locks
- > Coordinator journals “completed” state

CA IDMS as a Participant



CA IDMS as a Coordinator



Initiating a 2-phase Commit

> Environment dependent initiation

- CICS SYNCPOINT
- RRS commit function (call RRSCMIT)
- XA environment controlled

> In DC/UCF

COMMIT/FINISH TASK

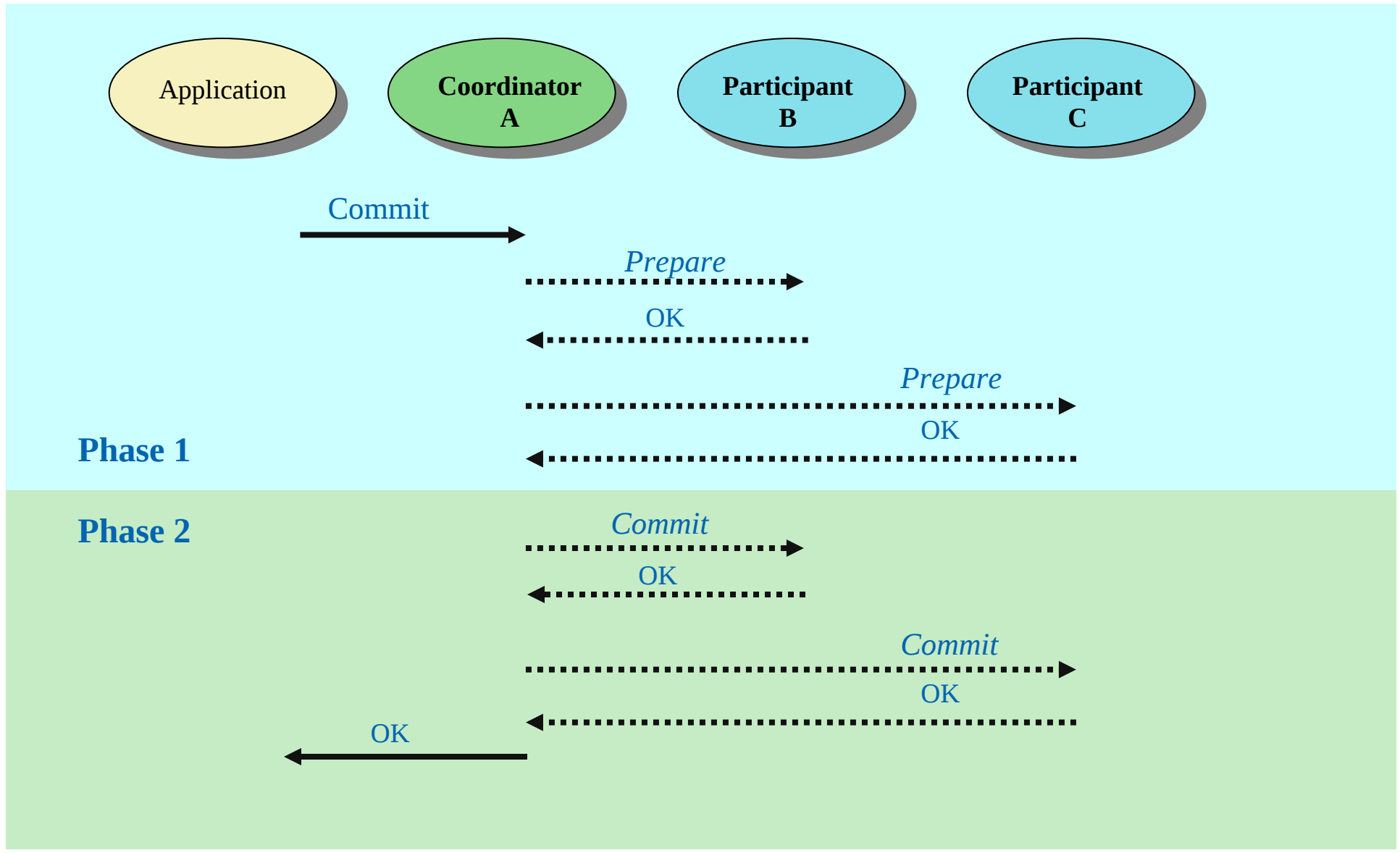
COMMIT/FINISH (nDML or SQL)

- Always two-phase operation if distributed updates
 - Multiple database sessions (at least one remote)
 - Transparent to application

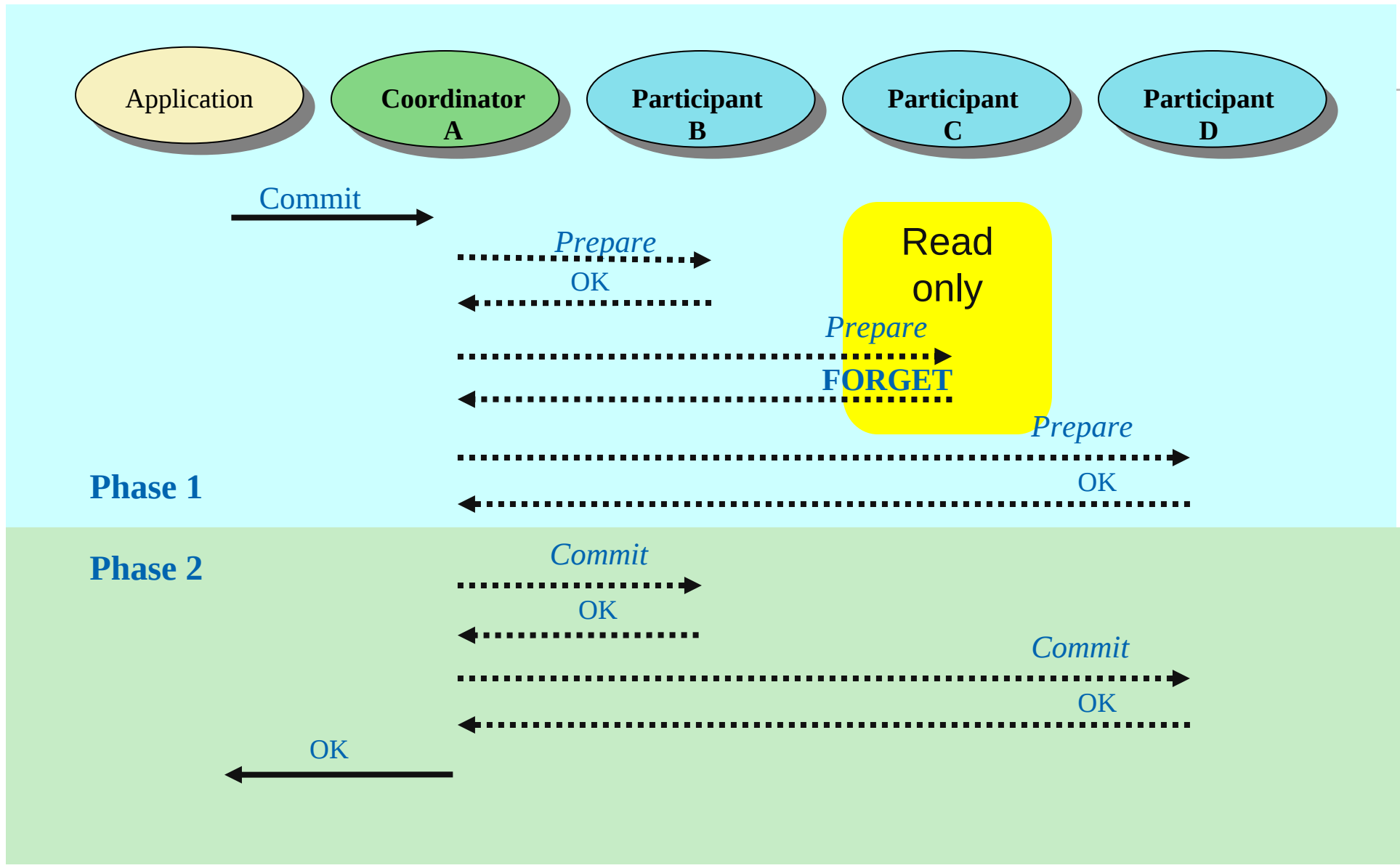
> What Happens?

CA IDMS 2-Phase Commit

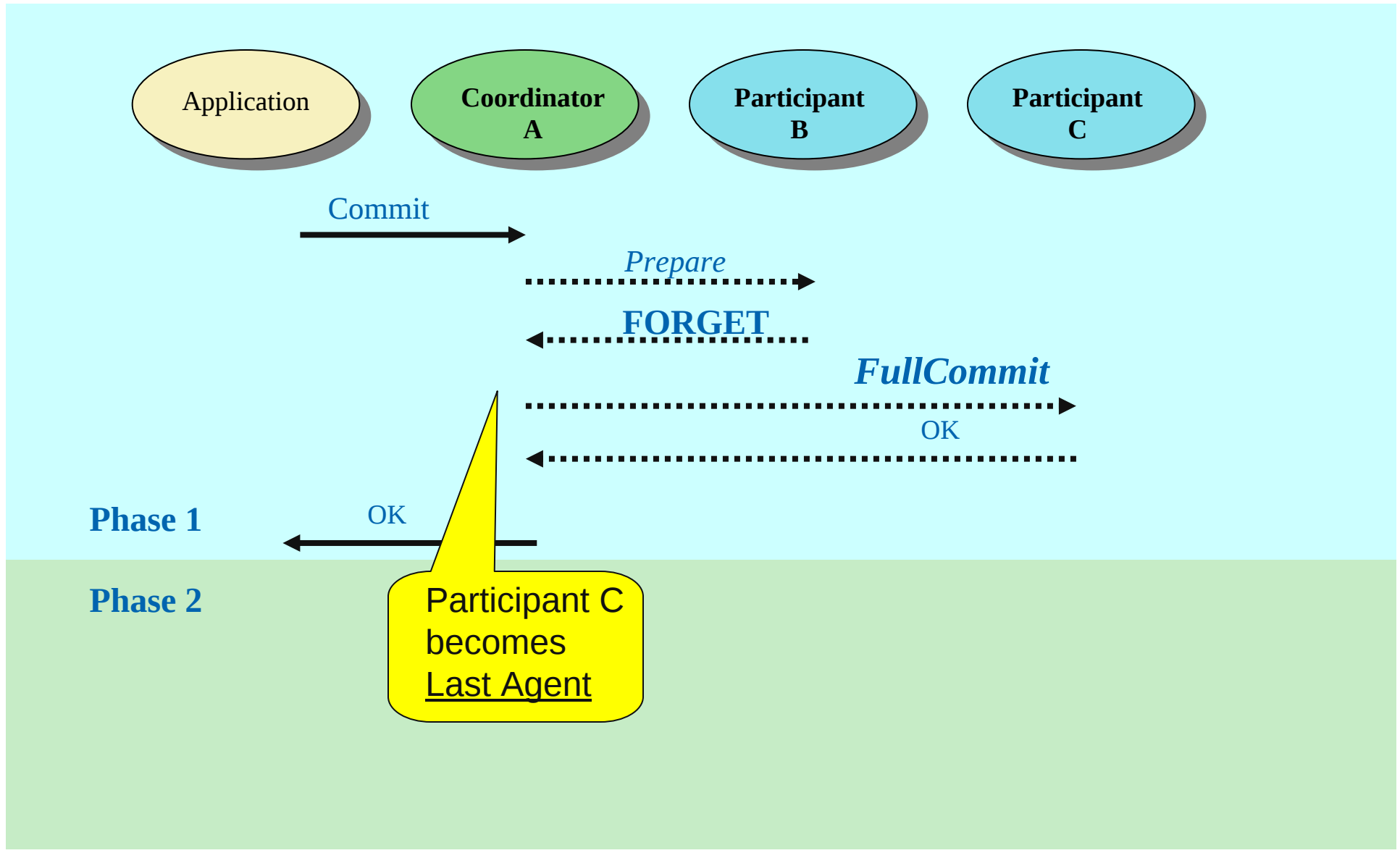
Typical Commit Flow



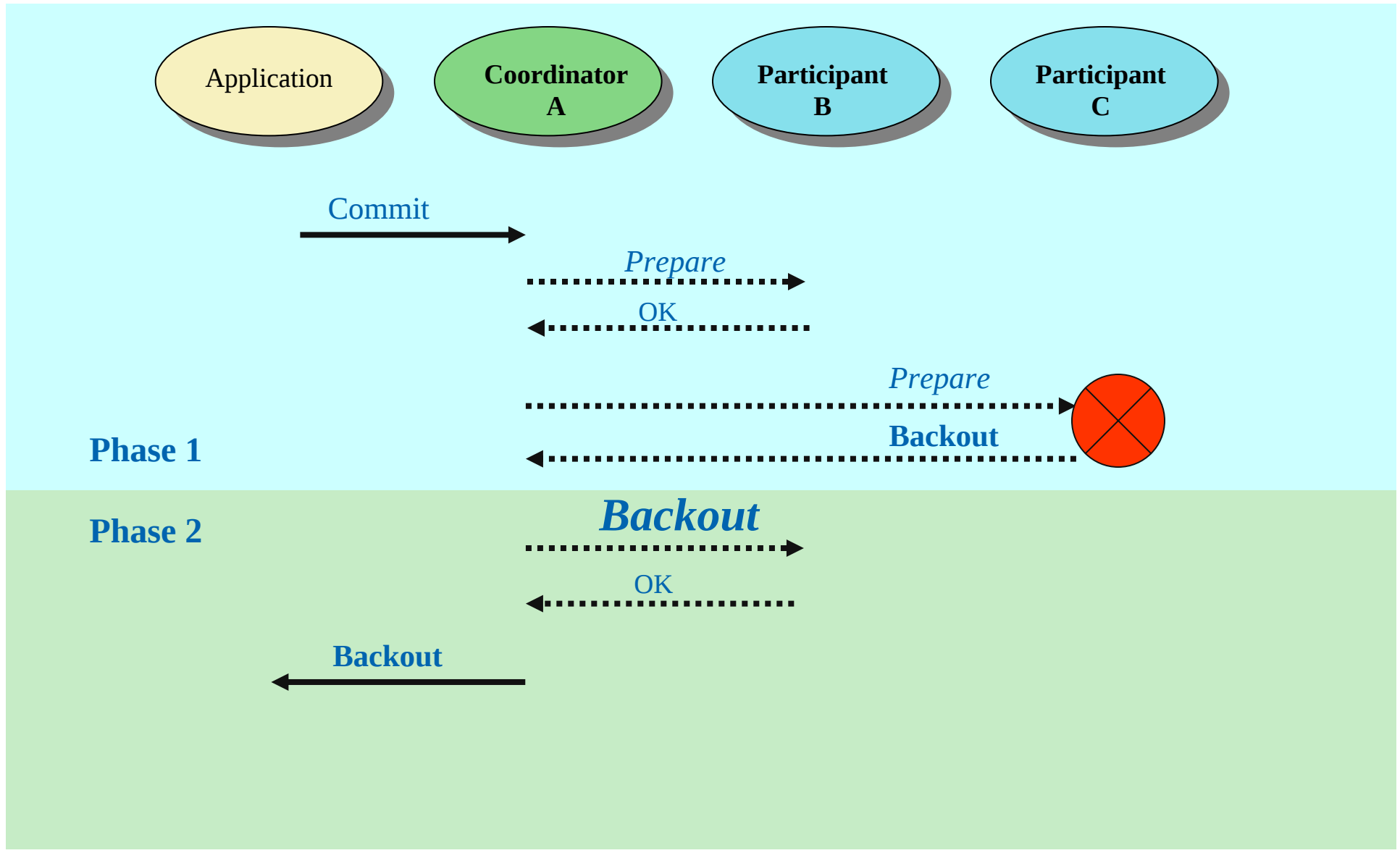
Read Only Optimization



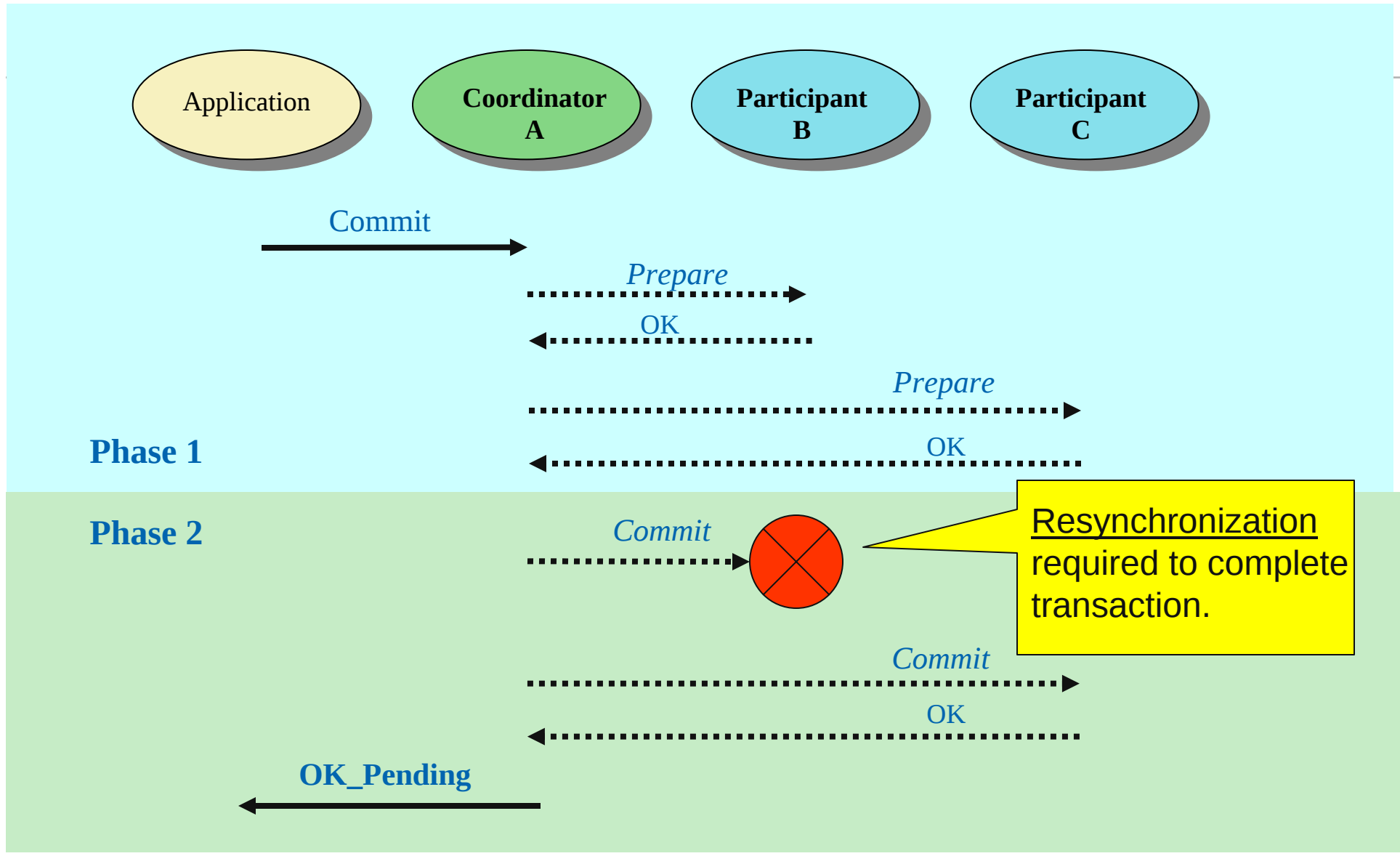
Single Agent Optimization



Failure During Phase 1

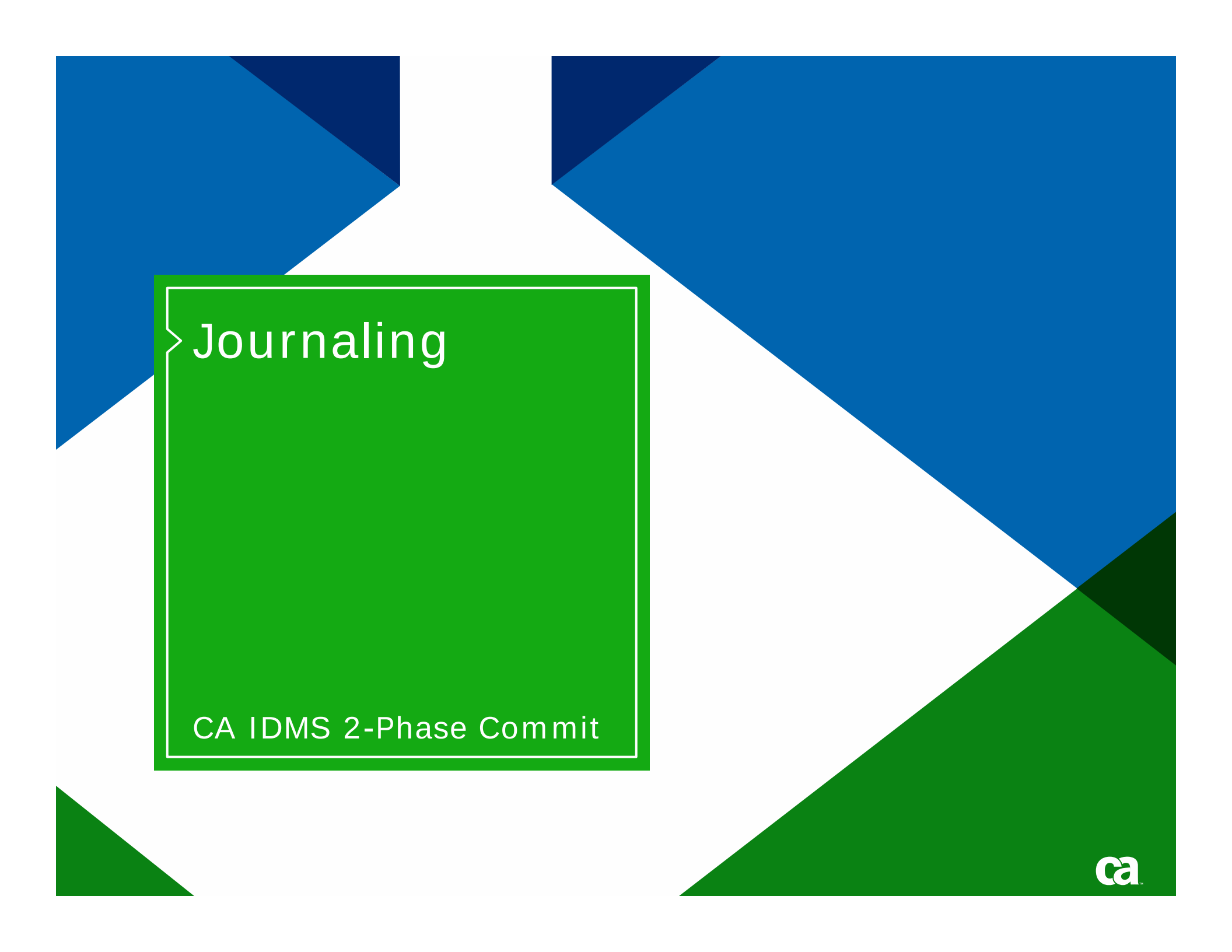


Failure During Phase 2



2-phase Commit Outcomes

OK	Commit operation successful
Forget	Commit successful from read-only participant
OK_Pending	Commit successful but not yet complete
Backout	Commit failed, changes backed out
Backout_Pending	Commit failed, changes will be backed out
HC	Commit successful; forced heuristically
HR	Commit failed; backout forced heuristically
HM	Commit failed; heuristic-mixed outcome



> Journaling

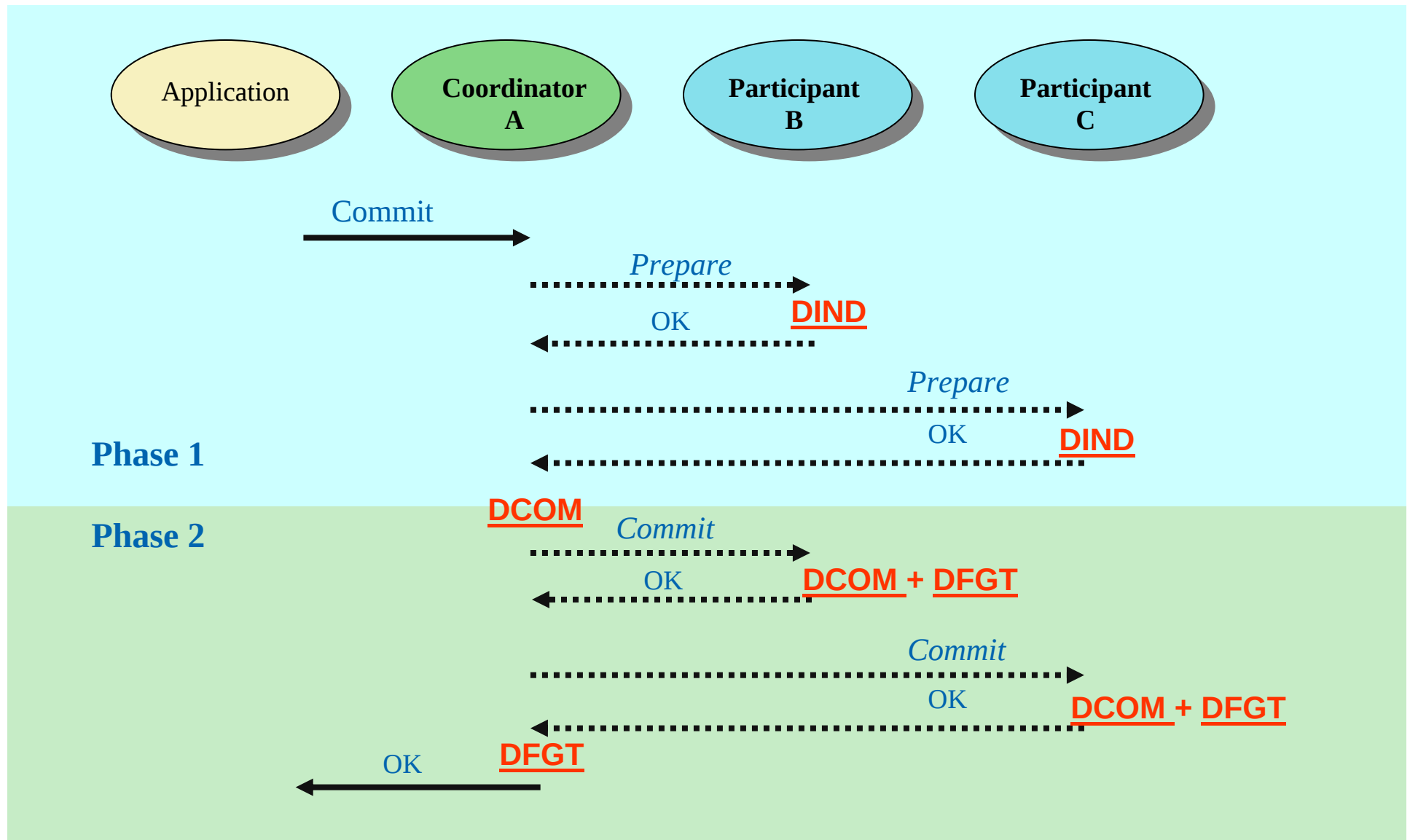
CA IDMS 2-Phase Commit

Distributed Checkpoint Records

<u>Type</u>	<u>Participant</u>	<u>Coordinator</u>	<u>Signifies</u>
DIND	When Prepare is complete	N/A	InDoubt state
DCOM	When Commit received	At end of phase-1	InCommit state
DBAK *	When Backout received	When backout decision made	InBackout state
DPND	When forced to complete <u>heuristically</u>	Failure during phase-2; HM outcome	An interim result
DFGT *	End of commit or backout	End of two-phase operation	End of distributed transaction

* Only if another Dxxx record previously written

2-phase Commit Journaling



Sequence of Journal Checkpoints

BGIN

{BFOR, AFTR...}

DIND on a participant

If successful

DCOM

COMT or ENDJ

{DPND}

DFGT

If unsuccessful

DBAK

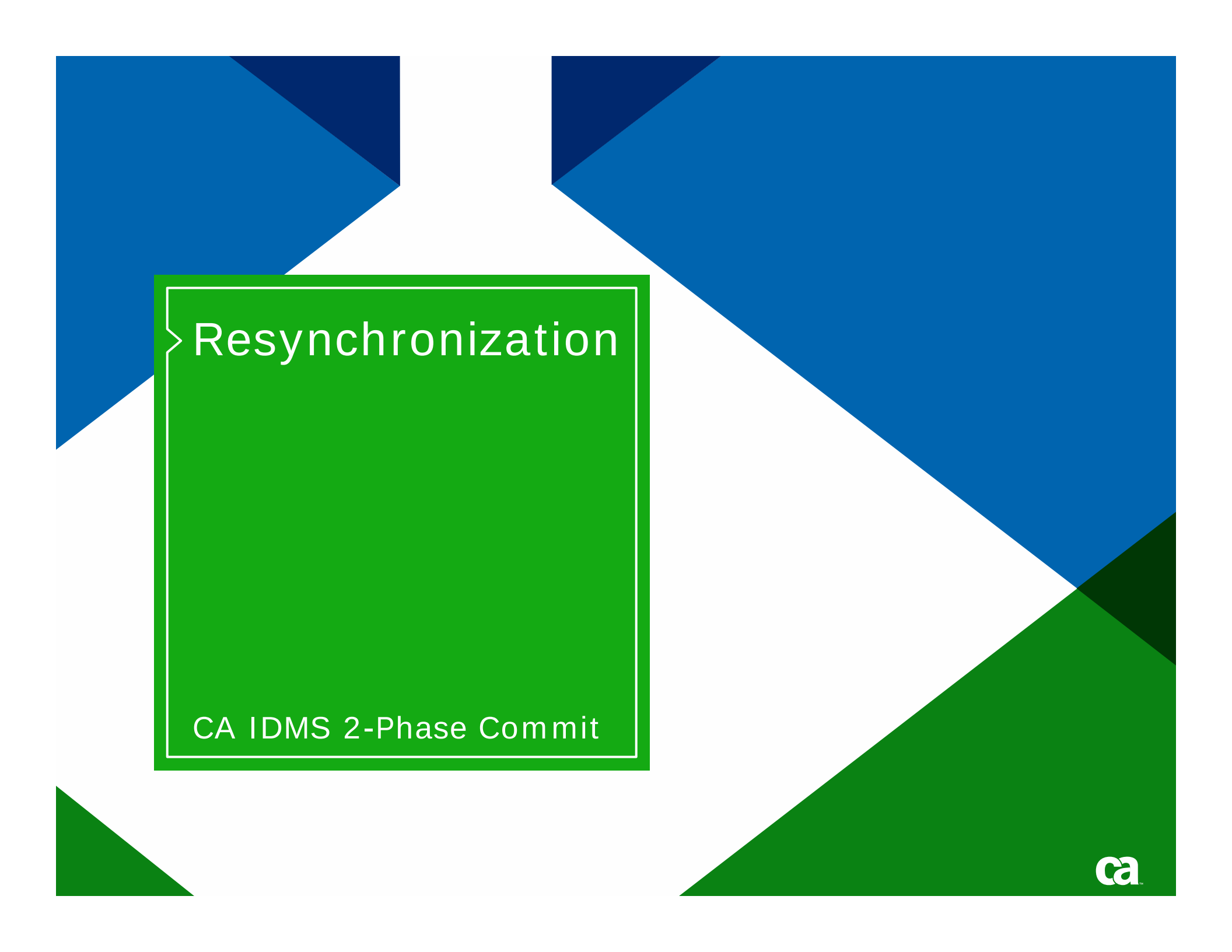
ABRT

{DPND}

DFGT

Transaction Identifiers

<u>LID</u>	<u>Local transaction identifier</u> >Part of key for BGIN, BFOR, AFTR, ... > Included in Dxxx checkpoint records >4-byte value						
<u>DTRID</u>	<u>Distributed transaction identifier</u> >Part of key for Dxxx checkpoint records >16-byte value <table><tr><td><u>Bytes 1-8</u></td><td><u>Bytes 9-16</u></td></tr><tr><td>CV Nodename</td><td>Binary timestamp</td></tr><tr><td>CICS CICS system id</td><td>CICS UOW</td></tr></table>	<u>Bytes 1-8</u>	<u>Bytes 9-16</u>	CV Nodename	Binary timestamp	CICS CICS system id	CICS UOW
<u>Bytes 1-8</u>	<u>Bytes 9-16</u>						
CV Nodename	Binary timestamp						
CICS CICS system id	CICS UOW						
<u>RRS URID</u> <u>XA XID</u>	<u>External transaction identifiers</u> > Included in Dxxx checkpoint records						



> Resynchronization

CA IDMS 2-Phase Commit

Resynchronization

- > A process that completes interrupted distributed transactions
- > Occurs automatically
 - During startup of coordinator
 - Immediate if participant active
 - Queued if not (retried on timer basis)
 - On first database communication following recycle of either system
- > Can be manually driven

DCMT VARY DIST RESOURCE MANAGER ... RESYNC

Resynchronization Messages

[Restarting SYSTEM74]

**IDMS DC329018 V74 T23 Resynchronization terminated for SYSTEM73::DSI_CLI
*connect failed**

IDMS DC224001 V74 T24 Registered with RRS services as IDMS.RM.SYSTEM74.CA

IDMS DC224002 V74 T24 RRS log name ATR.BAA3A68073264180.IBM

IDMS DC224002 V74 T24 Resource Manager log name
IDMS.LOG.2003_12_11_00.48.12.067048.SYSTEM74.CA

IDMS DC224006 V74 T24 Resynchronization with RRS complete

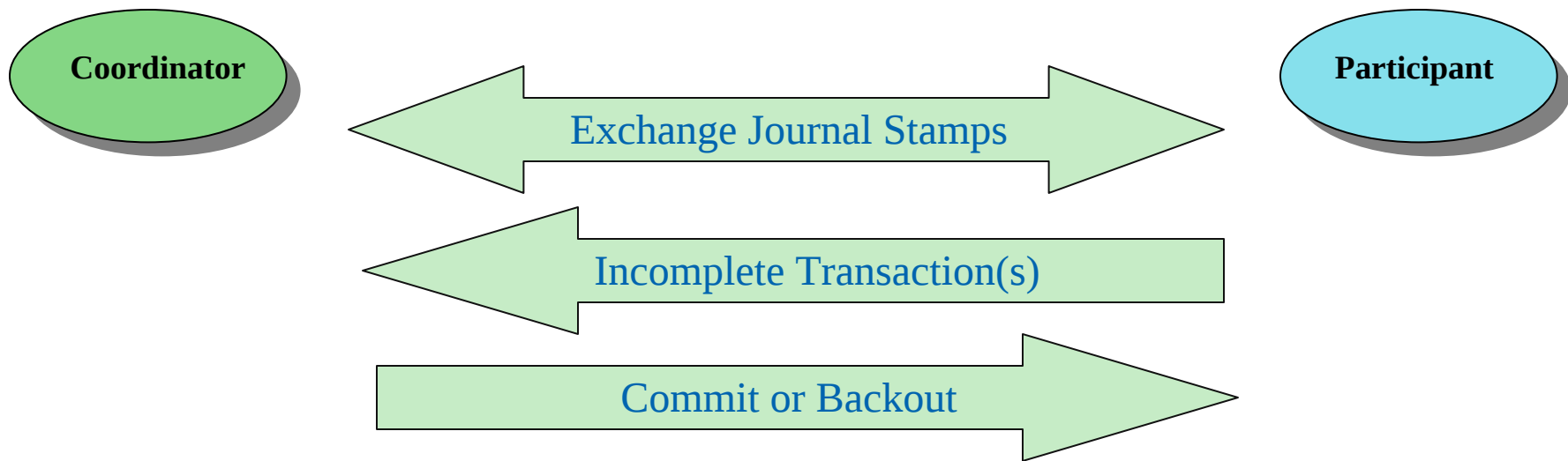
IDMS DC329013 V74 T22 Startup resynchronization complete

. . .

**IDMS DC329019 V74 T48 Resynchronization completed successfully for
SYSTEM73::DSI_CLI**

IDMS DC329020 V74 T49 DTRID SYSTEM74::01653D8B91E10DD7-01653D872C0639DC
complete, outcome OK

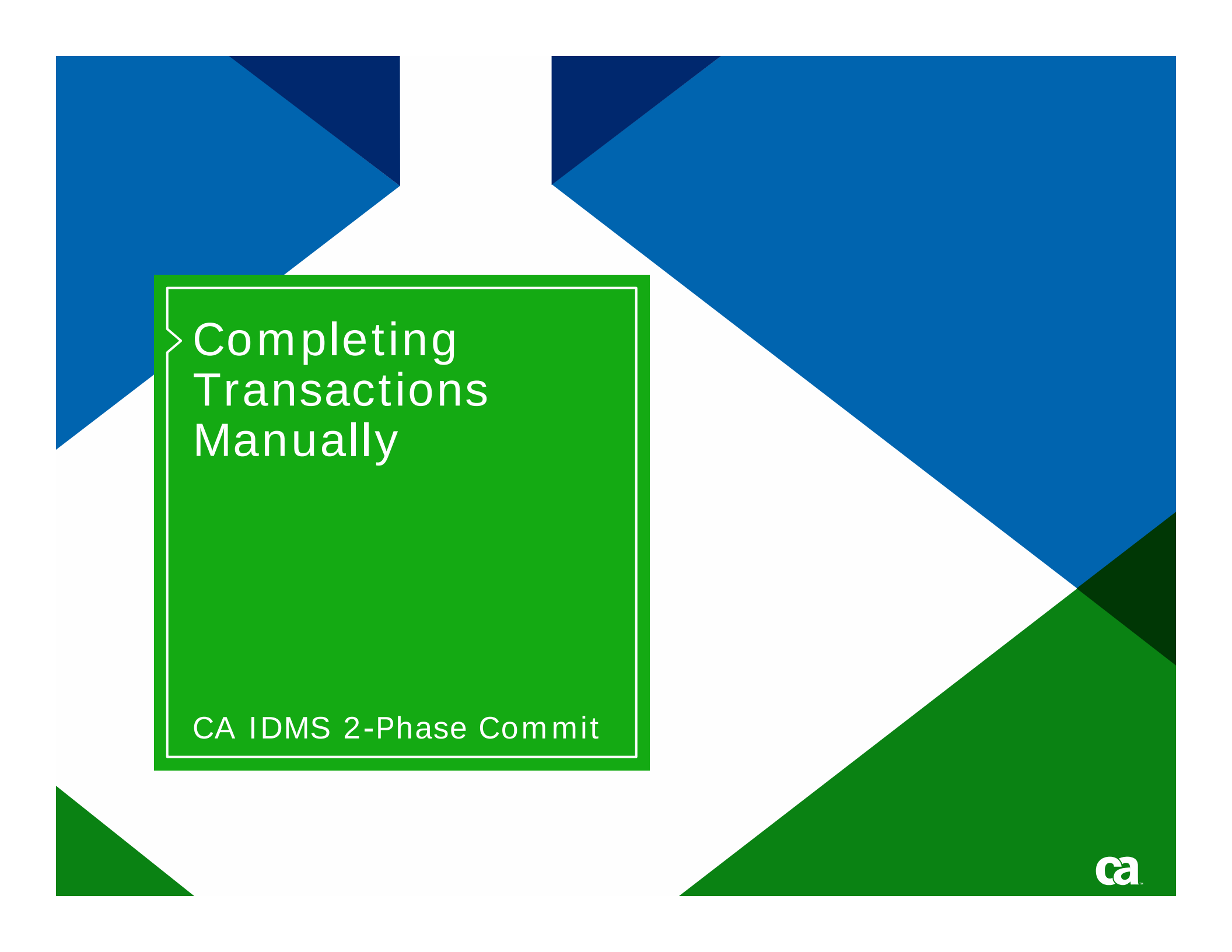
Resynchronization Flow



- > Partners' journal stamp saved on journal file
- > If changed, partner's journal has been formatted
- > If incomplete transactions exist
 - Must be manually completed
 - Operator prompted: Defer or Ignore

Causes of Resynchronization Failures

- > Participant not active
- > Unable to communicate
 - Line is down
 - Participant or coordinator brought up on different OS image
 - Invalid sysgen changes
- > Journal stamp mismatch
 - Journal files prematurely initialized
 - Another system masquerading as participant or coordinator



Completing Transactions Manually

CA IDMS 2-Phase Commit

Completing Transactions Manually

> Don't do it...unless

- Journal files prematurely initialized
- Coordinator or participant permanently inaccessible

> Research before taking action

- Wrong decision could result in mixed outcome

> Use DCMT V DISTRIBUTED TRANSACTION if possible

- System must be runnable
- Distributed transaction must still exist on disk journals

> Manual recovery control file for utilities

- FIX ARCHIVE, ROLLBACK, ROLLFORWARD, EXTRACT JOURNAL

Completing Transactions Manually

DCMT V DIST TR ID 'xxxx' COMMIT|BACKOUT|FORGET

Vary Action	Transaction State		
	InDoubt	InCommit	InBackout
COMMIT	HC outcome DPND written	Commit retried Forgotten if complete & no heuristic outcome	n/a
BACKOUT	HR outcome DPND written	n/a	Backout retried Forgotten if complete & no heuristic outcome
FORGET	n/a	Commit retried Forgotten if complete	Backout retried Forgotten if complete



Operational Considerations

CA IDMS 2-Phase Commit

Operational Considerations

- > Restart failed systems as soon as possible
- > Restart on same operating system image
- > Always restarted with same name
 - Name comes from journal file
- > Incomplete distributed transactions at shutdown

IDMS DC200241 V74 T1 Active transactions exist. Abending.

 - System abends (3937)

Operational Considerations

> Use with CICS and RRS

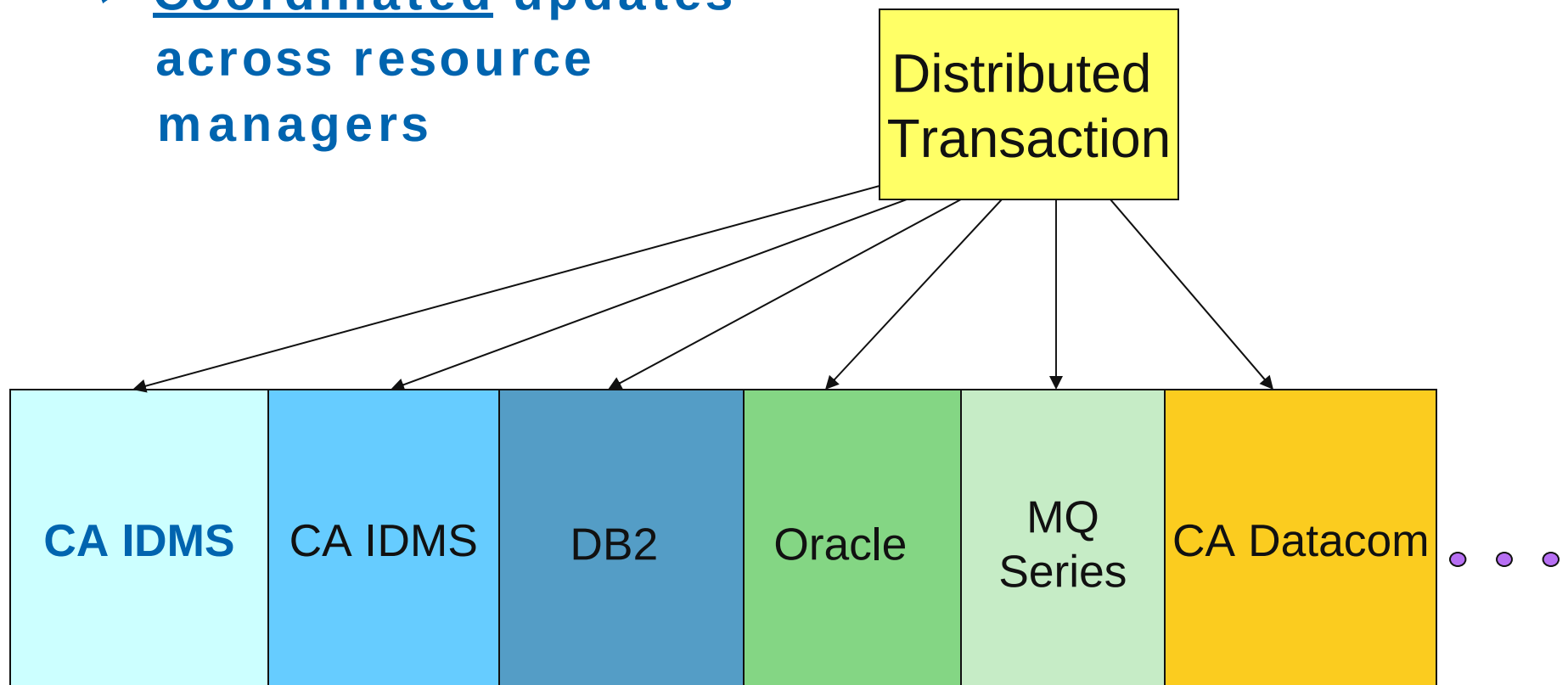
- Requires additional setup work
- And explicit activation

> Support for XA-transactions

- Requires CA IDMS™ Server r16

Session Summary

- ✓ Coordinated updates across resource managers



Legal

This presentation was based on current information and resource allocations as of October 10, 2007 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Legal

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised of the possibility of such damages.

Questions & Answers

CA IDMS 2-Phase Commit