

## **Vorteile:**

### **Höhere Verfügbarkeit von Daten:**

- Bei Ausfall eines lokalen Datenbank-Teils stehen die anderen Teile einer verteilten Datenbank noch zur Verfügung.
- Wenn Daten über mehrere Server dupliziert (repliziert) sind, stehen sie bei Ausfall eines lokalen Datenbank-Teils immer noch zur Verfügung.

Eine zentrale Datenbank ist bei Ausfall des Rechners, auf dem sie läuft, zur Gänze nicht mehr verfügbar.

### **Kürzere Zugriffszeiten auf Daten:**

- Die lokalen Datenbank-Teile sind kleiner als die gesamte verteilte Datenbank, und daher laufen Anfragen und Transaktionen auf ihnen, solange sie sich auf den lokalen Datenbank-Teil beschränken, schneller.
- Daten, die nur lokal von einem Client benötigt werden, können auf einem Server in seiner (unmittelbaren) geografischen Nähe gehalten werden .

Eine zentrale Datenbank ist in der Regel groß und für viele ihrer Clients geografisch weit entfernt.

## **Nachteile:**

**Notwendigkeit einer Kommunikationsstruktur zwischen den einzelnen Rechnern, auf denen die Datenbank installiert ist.**

**Bei Zugriff zunächst Lokalisierung der gewünschten Daten notwendig.**

**Komplexe Ausführungsstrategien für Anfragen und Transaktionen.**

**Komplexere Concurrency- und Recovery-Algorithmen notwendig.**

**Replikate-Verwaltung notwendig, insbesondere Konsistenz-Erhaltung bei schreibenden Zugriffen.**

# **Verteilte Datenbanken: Vor- und Nachteile**

VS 5.2

**Mitarbeiter**

<u>manr</u>	vname	nname	gebdat	adresse	gehalt	chefnr	anr
111111	Lore	Kaiser	30.01.60	Badstr. 83, 42651 Solingen	100000	Null	Null
200200	Frank	Schmidt	15.02.61	Ilmstr. 14, 42327 Wuppertal	80000	111111	1
246246	Gregor	Gordon	15.11.80	Seeweg 4, 51379 Opladen	60000	200200	1
500500	Hanni	Saalfeld	13.09.64	Baude 17, 42349 Wuppertal	85000	111111	2
753753	Birgit	Meier	19.03.86	Stiege 9, 42349 Wuppertal	80000	500500	2
300300	Rainer	Rilke	17.04.70	Ahrweg 6, 42853 Remscheid	75000	111111	3
157157	Birgit	Zweistein	24.08.92	Wies 29, 42855 Remscheid	50000	300300	3
941941	Anne	Droste	01.05.98	Turmweg 9, 42653 Solingen	40000	500500	2
864864	Daniel	Littmann	16.01.95	Jostr. 23, 40822 Mettmann	90000	200200	1
303303	Martin	Hallauer	05.11.94	Romallee 7, 42655 Solingen	40000	500500	2

Projekt	pname	<u>pnummer</u>	pstandort	anum
	Solarzellen	10	Remscheid	2
	Optimierung	20	Solingen	1
	Speicherchip	30	Wuppertal	2
	Digitalisierung	40	Wuppertal	1
	Gateways	50	Solingen	2
	Hybridantriebe	60	Wuppertal	2
	NeueSoftware	70	Wuppertal	2
	UnitTests	80	Remscheid	3

Arbeitet_an	<u>mnr</u>	<u>pnr</u>	stunden
	246246	10	30,0
	246246	20	2,5
	246246	30	7,5
	753753	70	20,0
	753753	80	20,0
	157157	70	40,0
	500500	40	10,0
	941941	50	20,0
	941941	60	10,0
	941941	70	10,0
	864864	60	30,0
	864864	80	10,0
	303303	10	5,0
	303303	50	35,0

Abteilung	aname	<u>anummer</u>	leitnr
	Verwaltung	1	200200
	Entwicklung	2	500500
	Revision	3	300300

Standorte	<u>anr</u>	<u>astandort</u>
	1	Solingen
	1	Wuppertal
	2	Remscheid
	2	Solingen
	2	Wuppertal
	3	Remscheid

Angehöriger	<u>mnum</u>	<u>agname</u>	gebdat	verwgrad
	246246	Jochen	14.04.02	Sohn
	246246	Trude	13.09.04	Tochter
	300300	Jochen	25.10.01	Sohn
	500500	Sabine	11.01.03	Tochter
	500500	Hanni	13.09.05	Tochter
	500500	Karl	24.11.99	Sohn
	157157	Karl	06.10.12	Sohn

(adaptiert von Elmasri/Navathe:  
Grundlagen von  
Datenbanksystemen)

**Die FIRMA-Datenbank ist an den Standorten A (Verwaltung = Abteilung 1), B (Entwicklung = Abteilung 2) und C (Revision = Abteilung 3) positioniert.**

**Die Verwaltung kann am Standort A auf die gesamte Datenbank lokal zugreifen (Ist das wirklich sinnvoll?).**

**An den Standorten B und C der Abteilungen 2 bzw. 3 werden alle Informationen über die jeweilige Abteilung lokal vorgehalten. Bzgl. der Mitarbeiter interessieren an den Abteilungsstandorten nur die Mitarbeiterinformationen Mitarbeiter-Nummer, Name und Chef (und nicht Gehalt, Geburtsdatum und Adresse). Auch die Informationen über die Verwandten von Mitarbeitern interessieren an den Abteilungsstandorten nicht.**

**Und natürlich interessiert an einem einzelnen Standort nicht die Nummer dieses Standorts, die kennen dort alle. Auch die weiteren Informationen über diese Abteilung (Abteilungsname, Leiter der Abteilung) sind dort bekannt.**

## **FIRMA Datenbank: Anforderungen an Verteilung**

**Standort A (gesamte FIRMA-Datenbank, Abteilung 1): MA1, ABT1, STANDORTE1, PROJ1, ARBEITET\_AN1, ANGEHÖRIGER1**

**Standort B (Abteilung 2):**

MA2	<u>MANR</u>	VNAME	NNAME	CHEFNR
	500500	Hanni	Saalfeld	111111
	753753	Birgit	Meier	500500
	941941	Anne	Droste	500500
	303303	Martin	Hallauer	500500

PROJ2	PNAME	<u>PNUMMER</u>	PSTANDORT
	Solarzellen	10	Remscheid
	Speicherchip	30	Wuppertal
	Gateways	50	Solingen
	Hybridantriebe	60	Wuppertal
	Neue Software	70	Wuppertal

STANDORTE2	ASTANDORT
	Remscheid
	Solingen
	Wuppertal

ARBEITET_AN2	<u>MNR</u>	<u>PNR</u>	STUNDEN
	246246	10	30,0
	246246	30	7,5
	753753	70	20,0
	753753	80	20,0
	157157	70	40,0
	500500	40	10,0
	941941	50	20,0
	941941	60	10,0
	941941	70	10,0
	864864	60	30,0
	303303	10	5,0
	303303	50	35,0

**Standort C (Abteilung 3, analog zu Standort B mit Abteilung 2): MA3, STANDORTE3, PROJ3, ARBEITET\_AN3**

### Schema:

Wi	<u>MNR</u>	<u>PNR</u>	STUNDEN
----	------------	------------	---------

mit  $i = 1, 2, \dots, 9$

### Fragmentation:

W1: Mitarbeiter der Abteilung 1, Projekt der Abteilung 1  
W2: Mitarbeiter der Abteilung 1, Projekt der Abteilung 2  
W3: Mitarbeiter der Abteilung 1, Projekt der Abteilung 3  
W4: Mitarbeiter der Abteilung 2, Projekt der Abteilung 1  
W5: Mitarbeiter der Abteilung 2, Projekt der Abteilung 2  
W6: Mitarbeiter der Abteilung 2, Projekt der Abteilung 3  
W7: Mitarbeiter der Abteilung 3, Projekt der Abteilung 1  
W8: Mitarbeiter der Abteilung 3, Projekt der Abteilung 2  
W9: Mitarbeiter der Abteilung 3, Projekt der Abteilung 3

### Allokation:

Standort A (Abteilung 1)

$ARBEITET\_AN1 = ARBEITET\_AN = W1 \cup \dots \cup W9$

Standort B (Abteilung 2):

$ARBEITET\_AN2 = W2 \cup W4 \cup W5 \cup W6 \cup W8$

Standort C (Abteilung 3):

$ARBEITET\_AN3 = W3 \cup W6 \cup W7 \cup W8 \cup W9$

### Replikation:

W6 und W8 sind zwischen den Standorten B und C repliziert. Weitere Replikationen existieren zwischen Standort A einerseits und den Standorten B und C andererseits.

**Relation ARBEITET\_AN:**

**Fragmentierung, Allokation, Replikation**

VS 5.6

**Standort A** (*gesamte Datenbank, Abteilung 1, also für jede Tabelle: Attribute-Liste: \*, Guard Condition: true*):

**Tabellen:** MA1, ABT1, ABT1\_STANDORTE, PROJ1, ARBEITET\_AN1, ANGEHÖRIGER1

**Standort B (Abteilung 2):**

**MA2:**

**Attribute-Liste:** MANR, VNAME, NNAME, CHEFNR

**Guard Condition:** ANR = 2

**STANDORTE2:**

**Attribute-Liste:** ASTANDORT

**Guard Condition:** ANR = 2

**PROJ2:**

**Attribute-Liste:** PNAME, PNUMMER, PSTANDORT

**Guard Condition:** ANUM = 2

**ARBEITET\_AN2:**

**Attribute-Liste:** \*

**Guard Condition:** MNR IN ( $\pi_{\text{MANR}}(\text{MA2})$ )  
OR PNR IN( $\pi_{\text{PNUMMER}}(\text{PROJ2})$ )

**Standort C (analog für Abteilung 3)**

\* : alle Attribute der zugrunde liegenden Relation

**Fragmentation, Allokation von FIRMA :  
Attribute-Listen und Guard Conditions**

VS 5.7

MITARBEITER an Standort X:

MANR	VNAME	NNAME	GEBDAT
------	-------	-------	--------

...

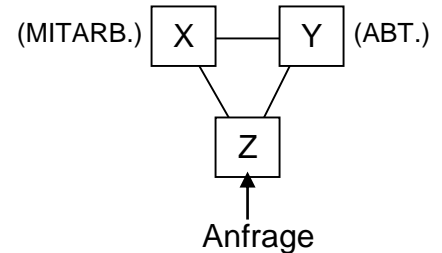
....	ADRESSE	GEHALT	CHEFNR	ANR
------	---------	--------	--------	-----

10000 Tupel von je  $10+10+10+10+35+10+10+5 = 100$  Bytes ( $\Rightarrow 1\,000\,000$  Bytes) .

ABTEILUNG an Standort Y:

ANAME	ANUMMER	LEITNR
-------	---------	--------

10 Tupel von je  $10+5+10 = 25$  Bytes ( $\Rightarrow 250$  Bytes).



**Anfrage an Standort Z (Name und Abteilungsname aller Mitarbeiter):**

$\pi_{VNAME, NNAME, ANAME} (MITARBEITER \bowtie_{ANR=ANUMMER} ABTEILUNG)$ .

**RESULTAT**

VNAME	NNAME	ANAME
-------	-------	-------

10000 Tupel von je  $10+10+10 = 30$  Bytes ( $\Rightarrow 300\,000$  Bytes).

**Strategie 1:**

Übertragung von MITARBEITER und ABTEILUNG nach Standort Z und dortige Ausführung des Joins.

Kosten: Übertragung von  $1\,000\,000 + 250 = 1\,000\,250$  Bytes.

**Strategie 2:**

Übertragung von MITARBEITER nach Standort Y, dortige Ausführung des Joins und Übertragung des Resultats nach Standort Z.

Kosten: Übertragung von  $1\,000\,000 + 300\,000 = 1\,300\,000$  Bytes.

**Strategie 3:**

Übertragung von ABTEILUNG nach Standort X, dortige Ausführung des Joins und Übertragung des Resultats nach Standort Z.

Kosten: Übertragung von  $250 + 300\,000 = 300\,250$  Bytes.

## Kommunikationskosten bei verteilten DB-Abfragen I



MITARBEITER an Standort X:

MANR	VNAME	NNAME	GEBDAT
------	-------	-------	--------

...

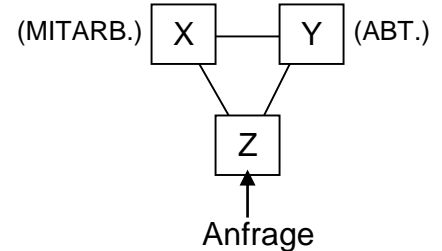
....	ADRESSE	GEHALT	CHEFNR	ANR
------	---------	--------	--------	-----

10000 Tupel von je  $10+10+10+10+35+10+10+5 = 100$  Bytes ( $\Rightarrow 1\,000\,000$  Bytes) .

ABTEILUNG an Standort Y:

ANAME	ANUMMER	LEITNR
-------	---------	--------

10 Tupel von je  $10+5+10 = 25$  Bytes ( $\Rightarrow 250$  Bytes).



**Anfrage an Standort Z (Name und Abteilungsname aller Mitarbeiter):**

$\pi_{VNAME, NNAME, ANAME} (MITARBEITER \bowtie_{ANR=ANUMMER} ABTEILUNG)$ .

**RESULTAT**

VNAME	NNAME	ANAME
-------	-------	-------

10000 Tupel von je  $10+10+10 = 30$  Bytes ( $\Rightarrow 300\,000$  Bytes).

**Strategie 1\* (Optimierung von Strategie 1):**

Übertragung von  $\pi_{VNAME, NNAME, ANR} (MITARBEITER)$  und  $\pi_{ANAME, ANUMMER} (ABTEILUNG)$  nach Standort Z und dortige Ausführung des Joins.

Kosten: Übertragung von  $250\,000 + 150 = 250\,150$  Bytes.

**Strategie 2\* (Optimierung von Strategie 2):**

Übertragung von  $\pi_{VNAME, NNAME, ANR} (MITARBEITER)$  nach Standort Y, dortige Ausführung des Joins und Übertragung des Resultats nach Standort Z.

Kosten: Übertragung von  $250\,000 + 300\,000 = 550\,000$  Bytes.

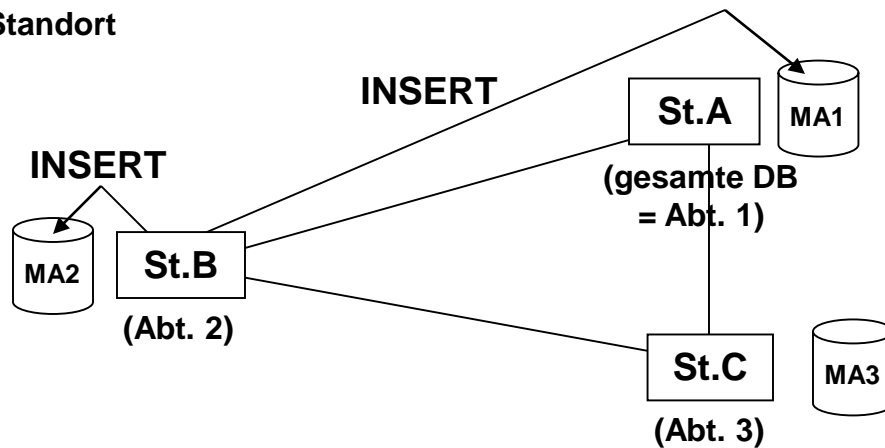
**Strategie 3\* (Optimierung von Strategie 3):**

Übertragung von  $\pi_{ANAME, ANUMMER} (ABTEILUNG)$  nach Standort X, dortige Ausführung des Joins und Übertragung des Resultats nach Standort Z.

Kosten: Übertragung von  $150 + 300\,000 = 300\,150$  Bytes.

## Kommunikationskosten bei verteilten DB-Abfragen II

St.=Standort



Einfügen am Standort B (Abteilung 2) durch einen Benutzer:

**INSERT INTO MITARBEITER VALUES**

(898898', 'Tamara', 'Sonderburg', '17.08.99', Erbsenweg 13,  
51379 Opladen', 21000, '500500', 2 )

an Standort A (Abteilung 1):

**INSERT INTO MA1 VALUES**

(898898', 'Tamara', 'Sonderburg', '17.08.99', Erbsenweg 13,  
51379 Opladen', 21000, '500500', 2 )

an Standort B (Abteilung 2):

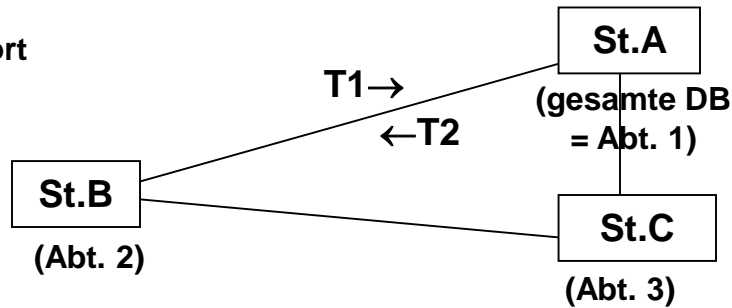
**INSERT INTO MA2 VALUES**

(898898', Tamara', Sonderburg', 500500')

an Standort C (Abteilung 3): kein INSERT

## Insert-Dekomposition in verteilter DB

St.=Standort



Anfrage an Standort B (Abteilung 2):

Namen und Stunden pro Woche für jeden Mitarbeiter, der an einem Projekt der Abteilung 2 mitarbeitet.

SQL-Query an Standort B (Abteilung 2):

```
SELECT  VNAME, NNAME, STUNDEN
FROM    MITARBEITER, ARBEITET_AN, PROJEKT
WHERE   MANR=MNR AND PNR=PNUMMER AND ANUM=2
```

Relationale Algebra an Standorten A und B

Ausführung an

$T1 \leftarrow \pi_{MNR} (PROJ2 \mid x \mid_{PNUMMER=PNR} ARBEITET\_AN2)$   
(T1 wird an Standort A geschickt)

Standort B

$T2 \leftarrow \pi_{MNR, VNAME, NNAME} (T1 \mid x \mid_{MNR=MANR} MA1)$   
(T2 wird an Standort B geschickt)

Standort A

$RS \leftarrow \pi_{VNAME, NNAME, STUNDEN} (T2 \mid x \mid_{MNR=MNR} ARBEITET\_AN2)$

Standort B

Beispiel aus Elmasri/Navathe: Grundlagen von Datenbanksystemen

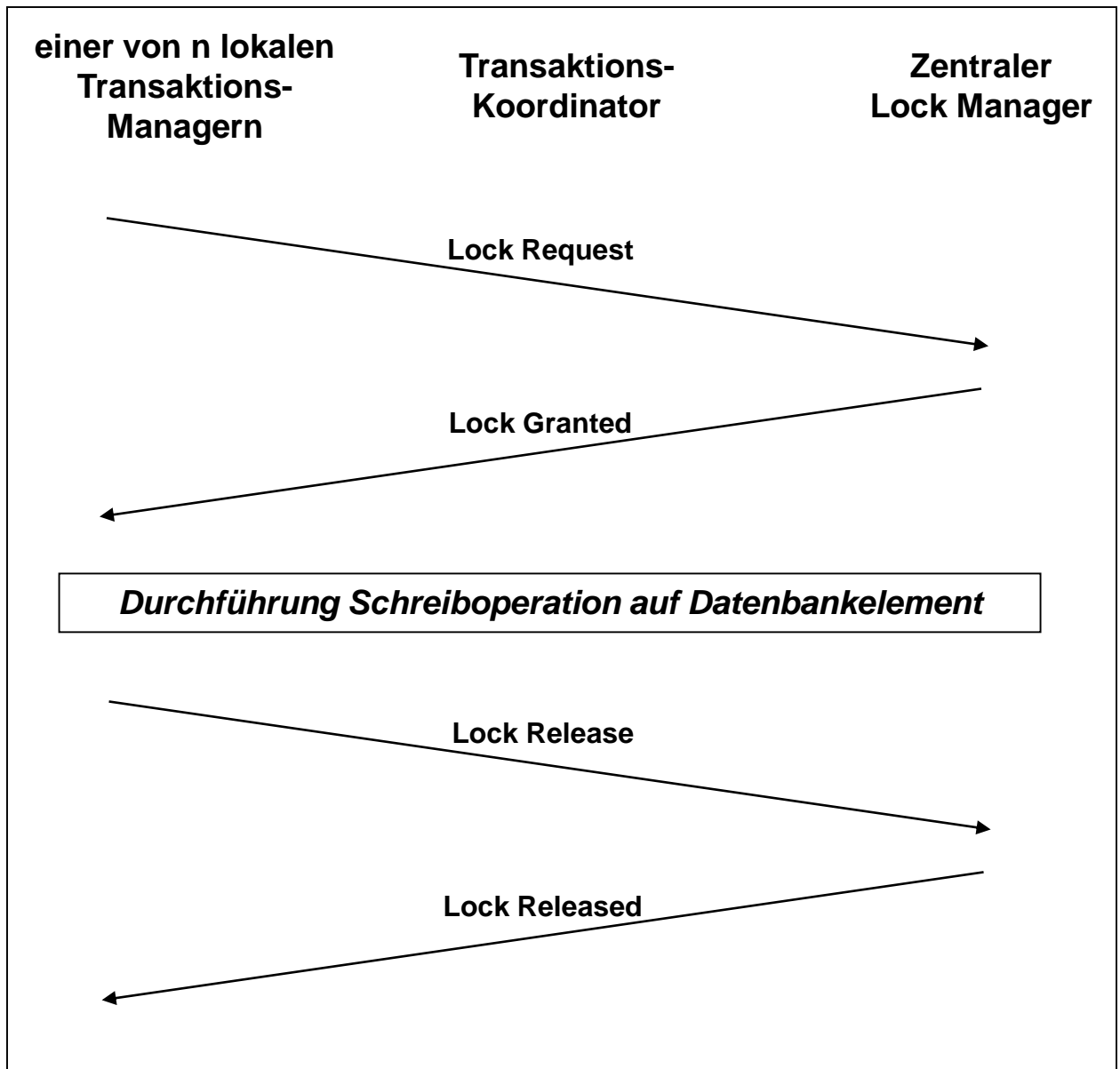
T1: Nummern der Mitarbeiter, die an Projekten der Abteilung 2 arbeiten  
(diese Mitarbeiter können auch anderen Abteilungen angehören,  
deren Namen müssen an Standort A erfragt werden)

T2: Nummern und Namen dieser Mitarbeiter

RS: Namen und Stunden pro Woche dieser Mitarbeiter

## Select-Dekomposition in verteilter DB

VS 5.11

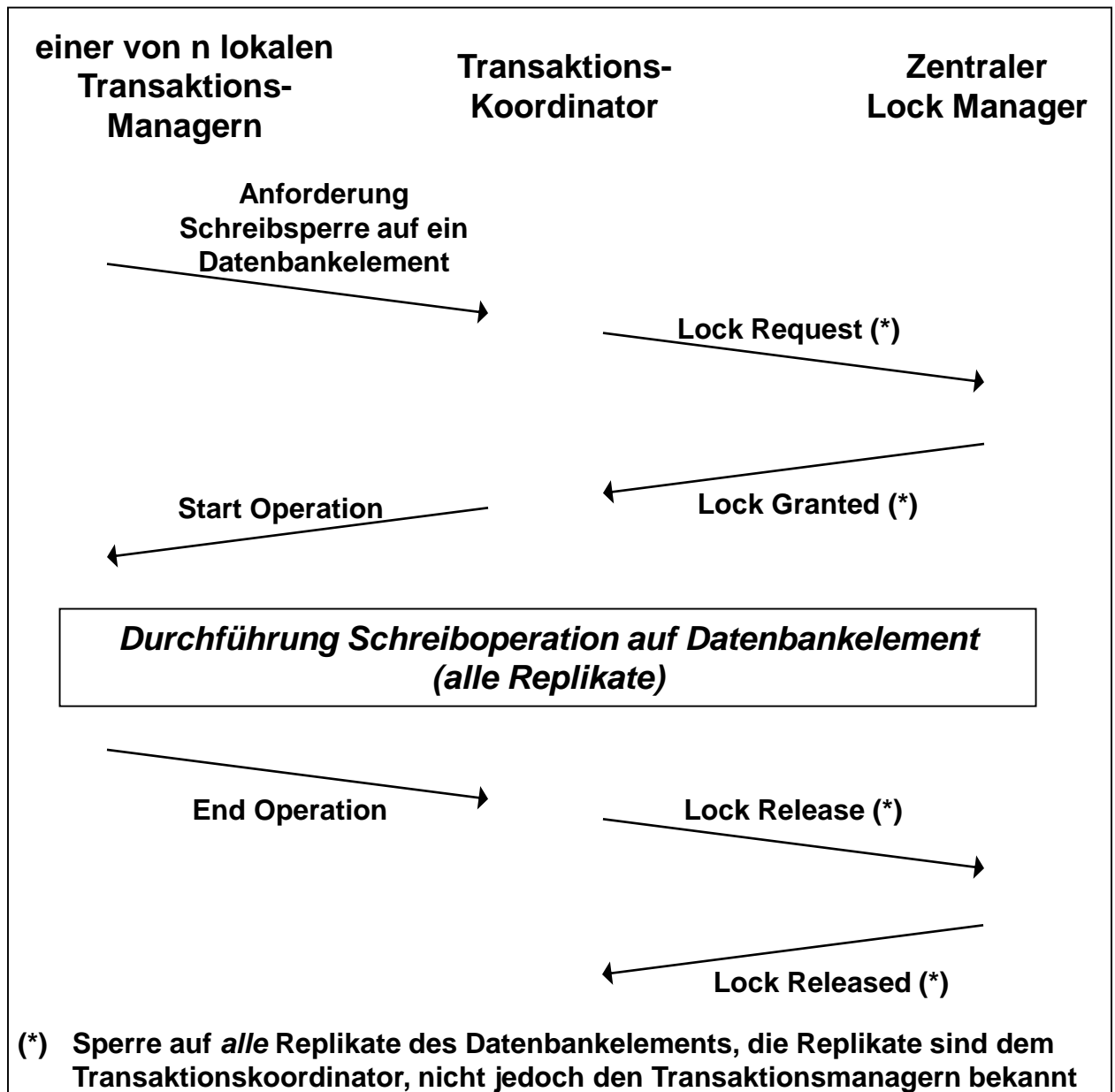


verteiltes System mit  $n$  Standorten

Ein zentraler Lock Manager verwaltet alle Sperren im verteilten System an einem einzigen Standort.

**Managerkommunikation bei zentralisiertem  
2PL (keine Replikate vorhanden)**

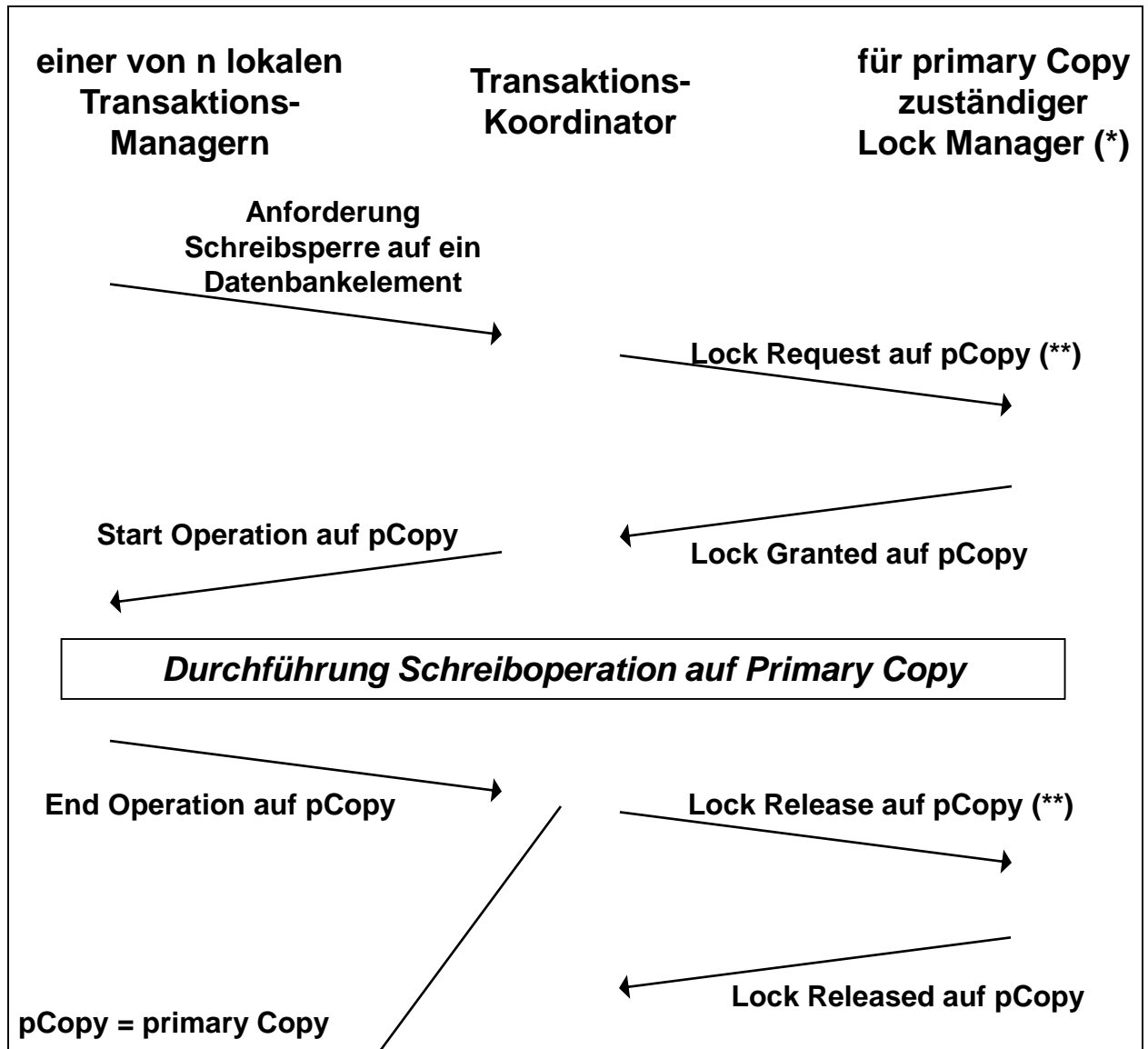
VS 5.12



verteiltes System mit n Standorten

Ein zentraler Lock Manager verwaltet alle Sperren im verteilten System an einem einzigen Standort.

## Managerkommunikation bei zentralisiertem 2PL (Replikate vorhanden)



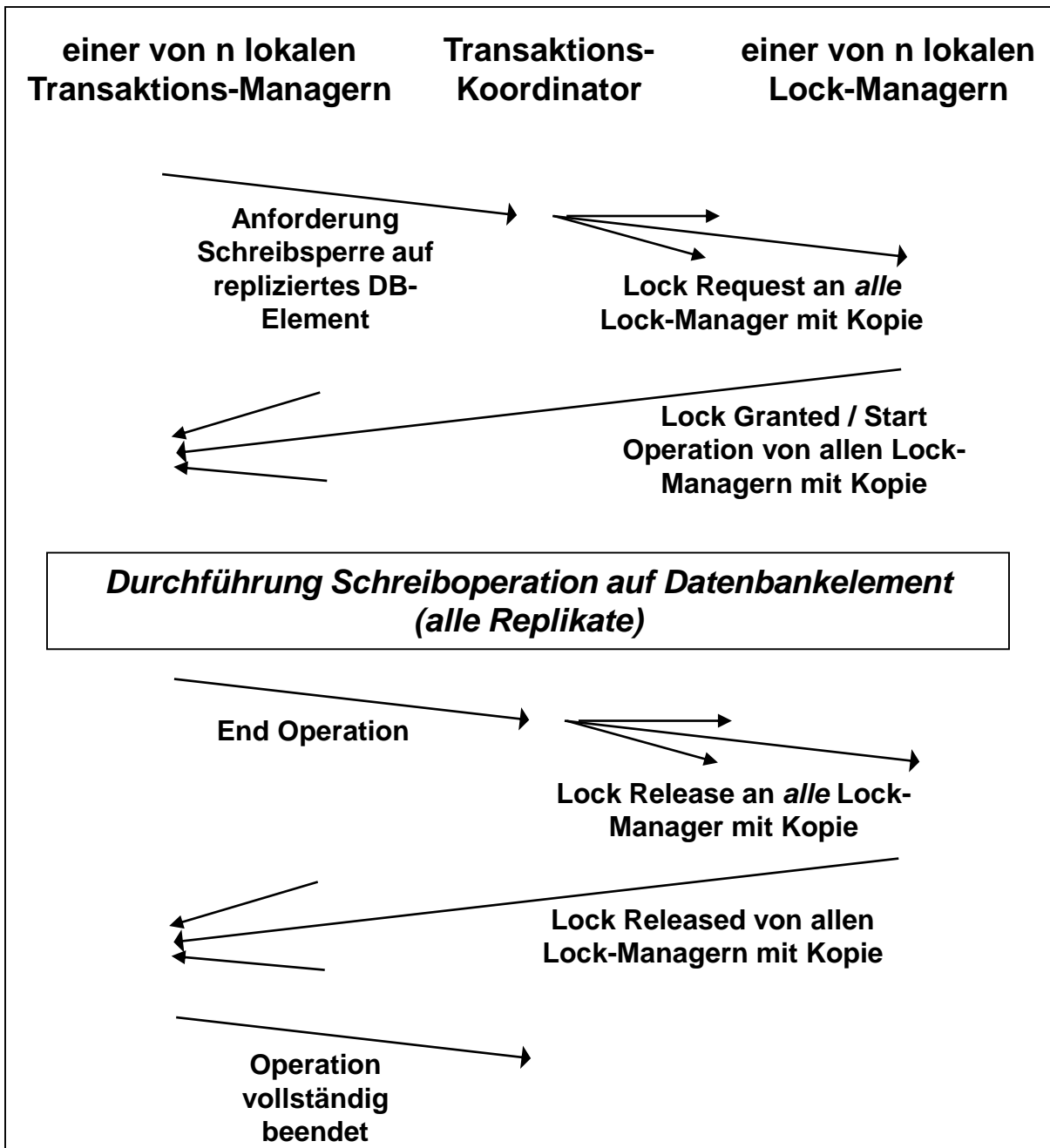
verteiltes System mit n Standorten

asynchron: auf Veranlassung durch den Transaktions-Koordinator aktualisiert der Replication Manager alle Slave Copies

(\*) Für jede pCopy im verteilten System ist ein bestimmter Lock Manager zuständig.

(\*\*) Die pCopy ist dem Transaktionskoordinator, jedoch nicht den Transaktionsmanagern bekannt.

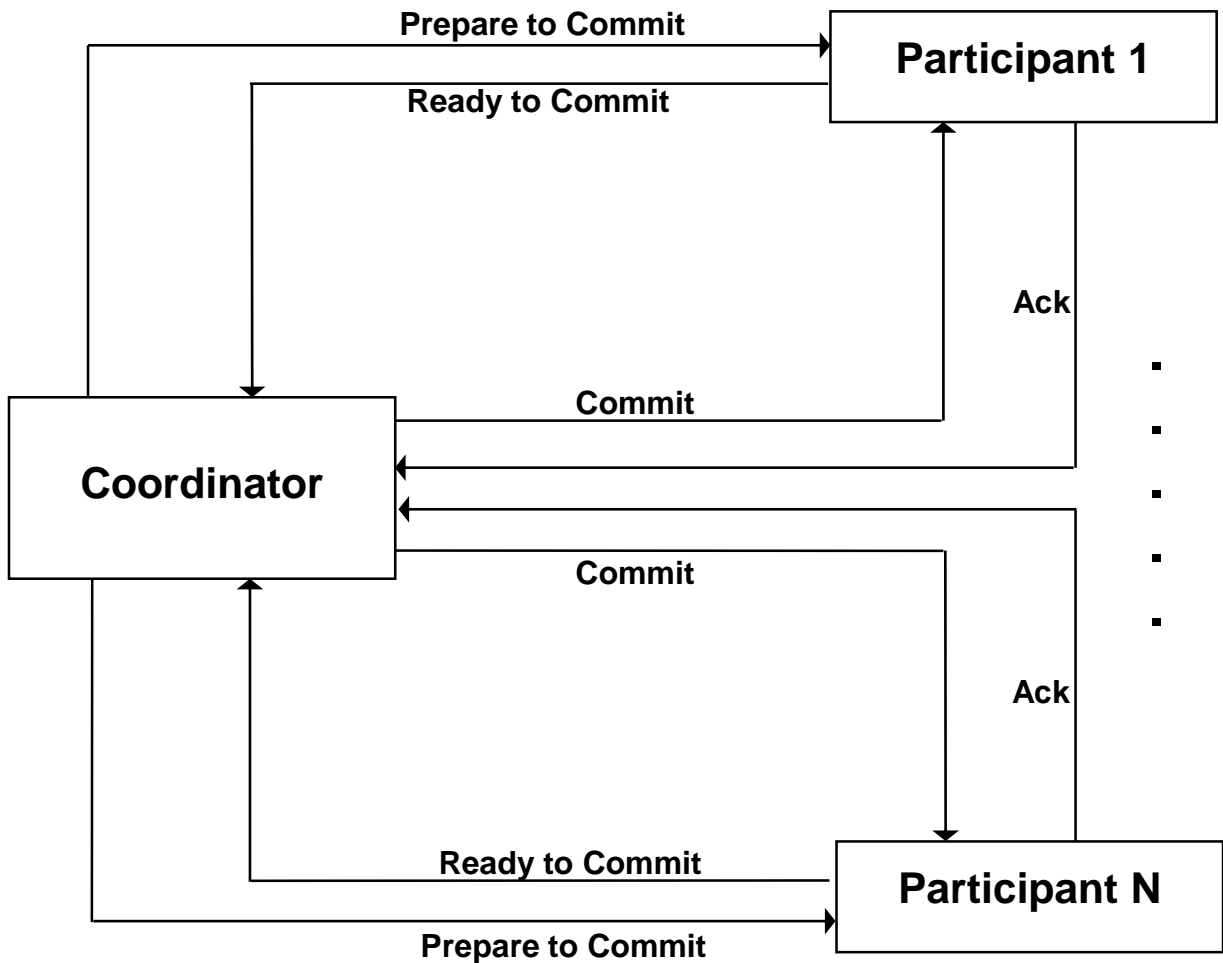
## Managerkommunikation bei primary copy 2PL



verteiltes System mit n Standorten

Auf jeder Komponente des verteilten Systems arbeitet ein lokaler Lock-Manager mit Verantwortung für die auf dieser Komponente gespeicherten Datenbankelemente.

## Managerkommunikation bei verteiltem 2PL



zeitliche Reihenfolge:    **Prepare to Commit**    } **Voting Phase**  
                                   **Ready to Commit**    }  
                                   **Commit**                } **Decision Phase**  
                                   **Ack**

## 2PC-Protokoll (positiver Fall)

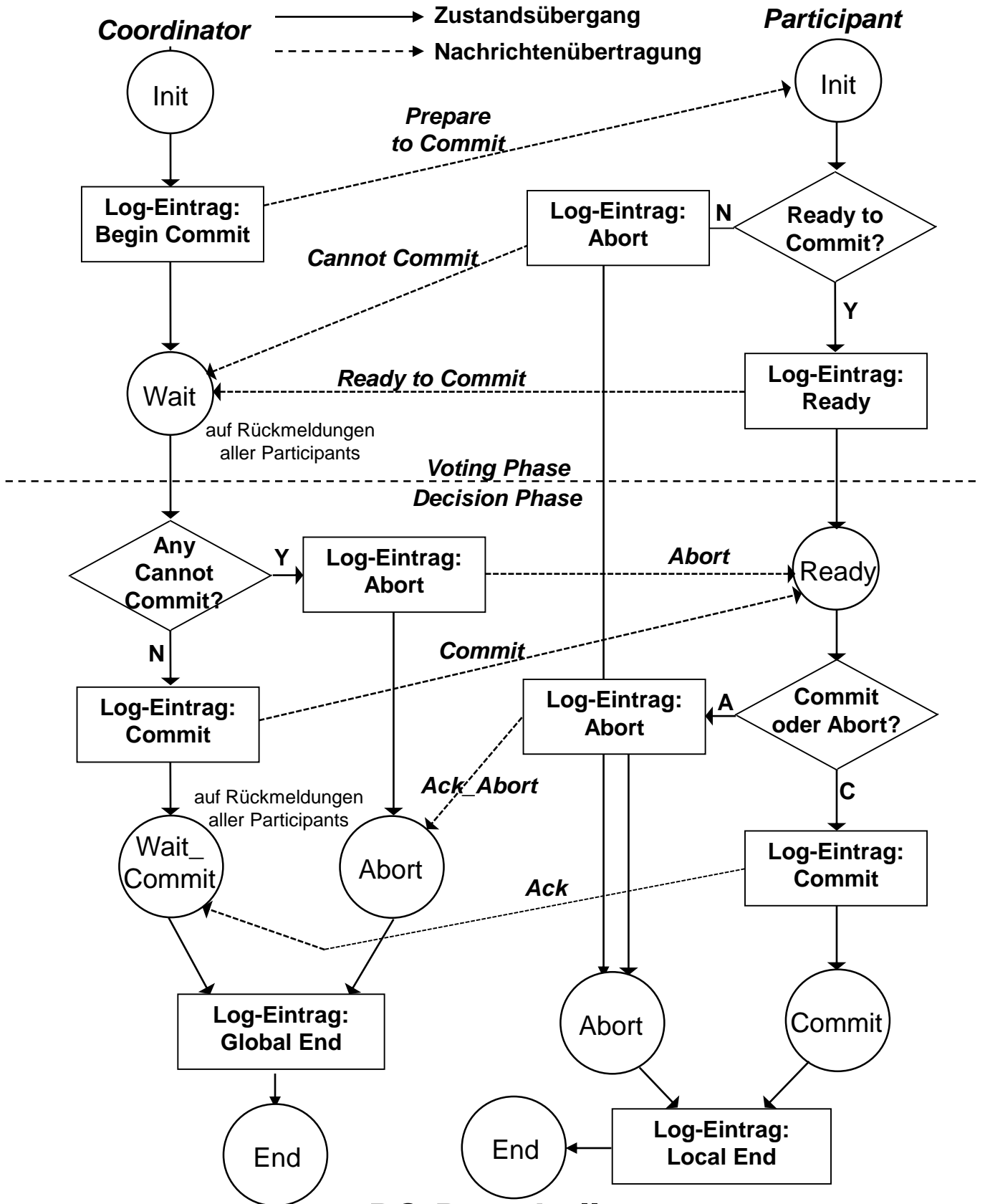


**Coordinator**

**Participant**

————→ Zustandsübergang

- - - - -> Nachrichtenübertragung



## 2PC-Protokoll

