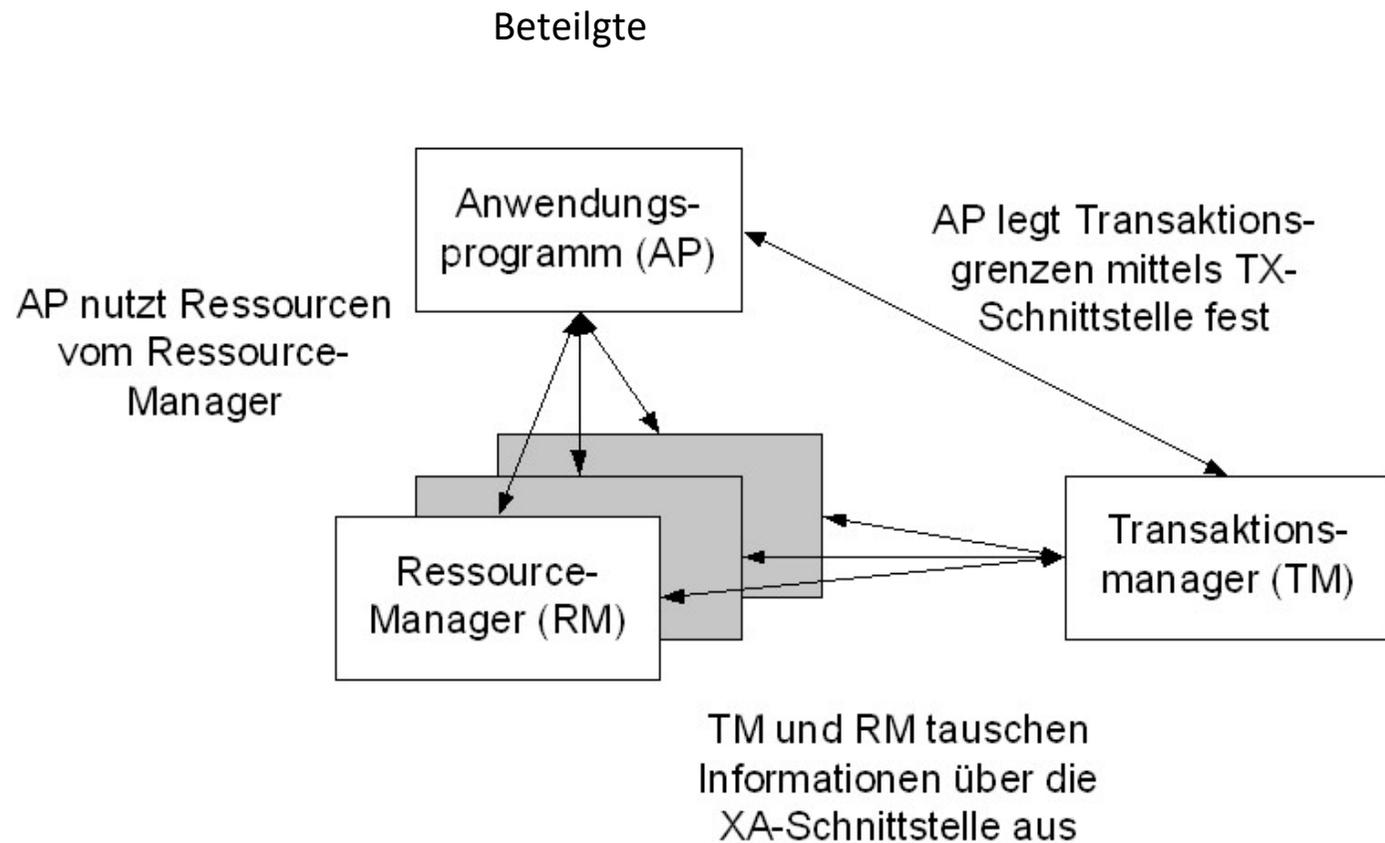


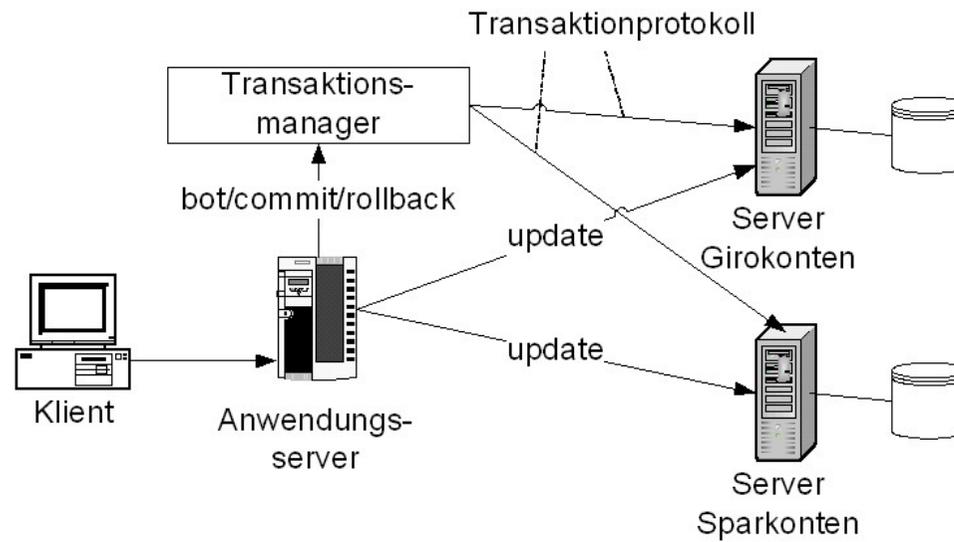
- Transaktionen, die mehr als eine Ressource umfassen
- Typische Ressource-Manager sind Datenbank- oder Messaging-Systeme

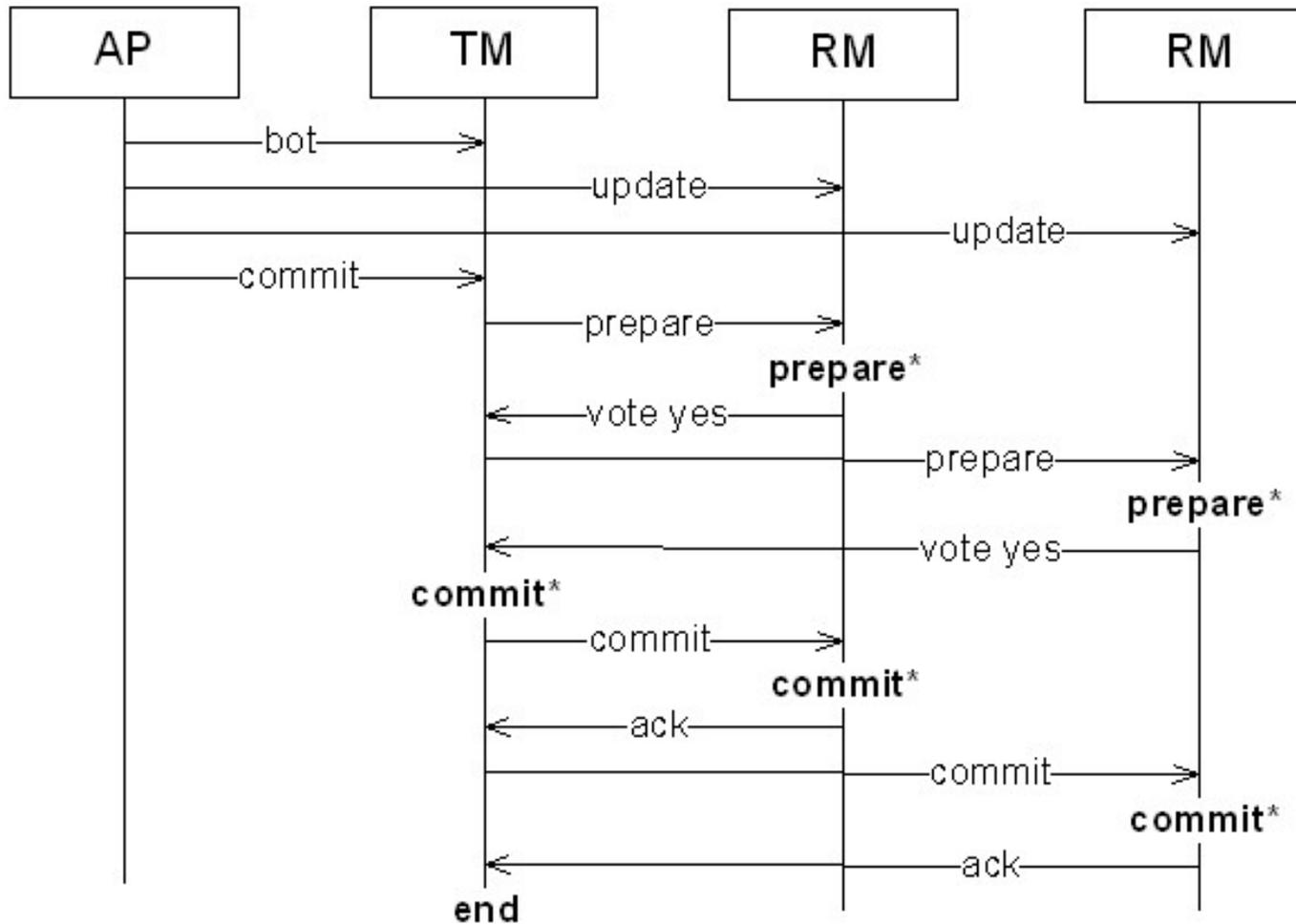


- ACID muss wie bei nicht verteilten Transaktionen gewahrt werden.
- Insbesondere müssen für die Atomarität alle RM entweder die Transaktion erfolgreich beenden oder alle ihre Änderungen rückgängig machen.
- Dafür ist ein spezielles Protokoll erforderlich, das sogenannte Zwei-Phasen-Commit-Protokoll, das die beteiligten RM steuert.
- Dieses Protokoll muss insbesondere mit Systemfehlern umgehen:
 - Kommunikationsfehler, wenn die beteiligten RM auf verschiedenen Servern laufen, die über ein Netzwerk miteinander verbunden sind.
 - TM stürzt ab, RM stürzt ab.
 - Der Wiederherstellungsprozess muss die notwendigen Aktionen zur Sicherstellung der Atomarität durchführen.

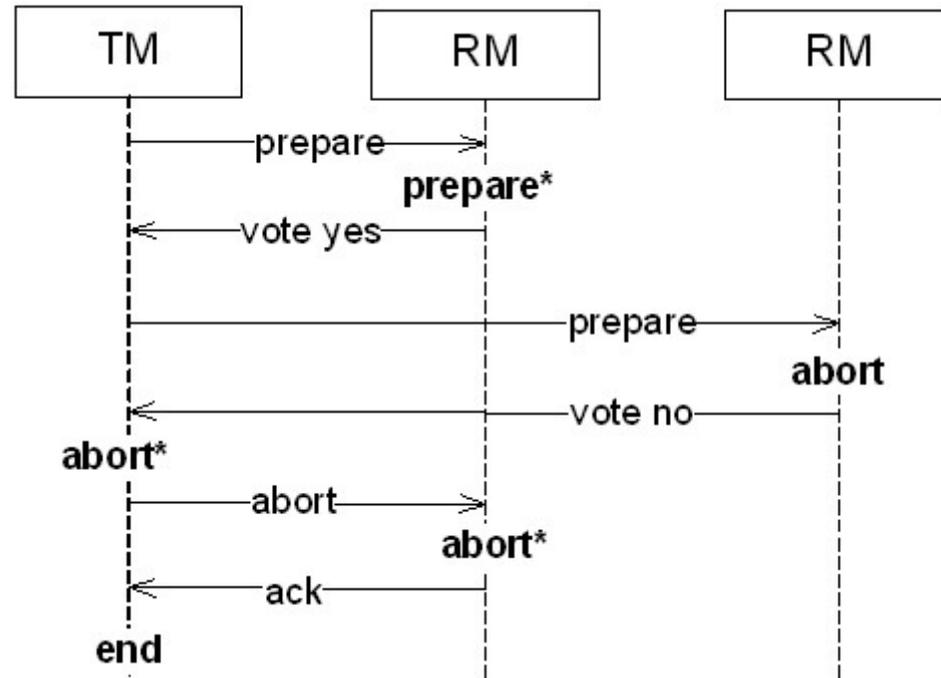
- Grundlage: Distributed-Transaction-Processing-Modell (DTP)
- TX-Spezifikation
 - Beschreibt die Schnittstelle zwischen dem AP und dem TM.
 - Das AP steuert die Transaktion durch Anweisungen zum Starten (bot), erfolgreichem Beenden (commit) und Abbruch (rollback).
- XA-Spezifikation
 - Beschreibt die Schnittstelle zwischen dem TM und den RM.
 - Mit Funktionen dieser Schnittstelle können Operationen im RM mit einer Transaktion assoziiert werden.
 - Stellt Funktionen zur Abbildung des Zwei-Phasen-Commit-Protokolls bereit.

- Verschiedene Unternehmenseinheiten für Giro- und Sparkonten
- Jeweils eigene IT-Infrastruktur

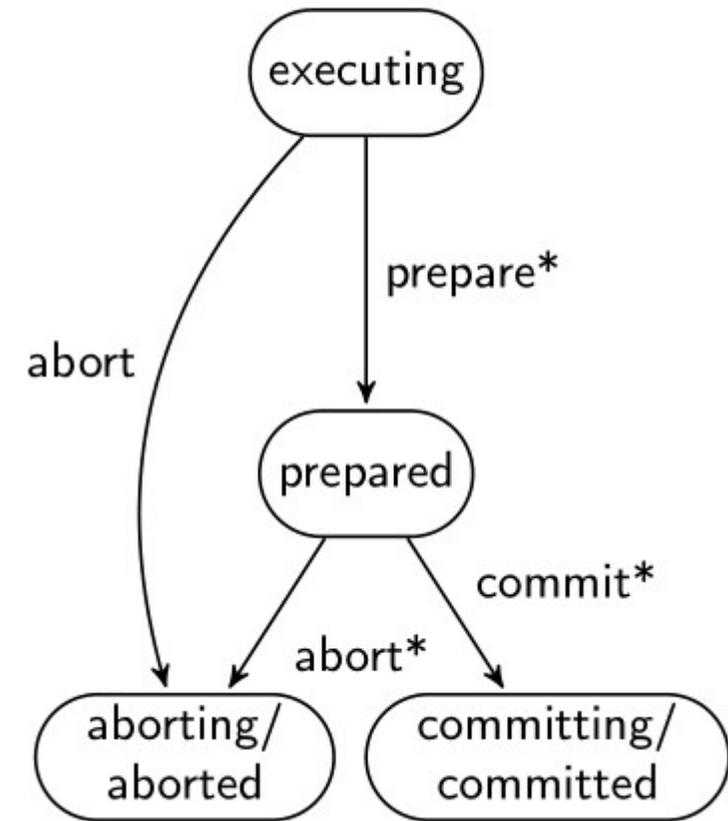




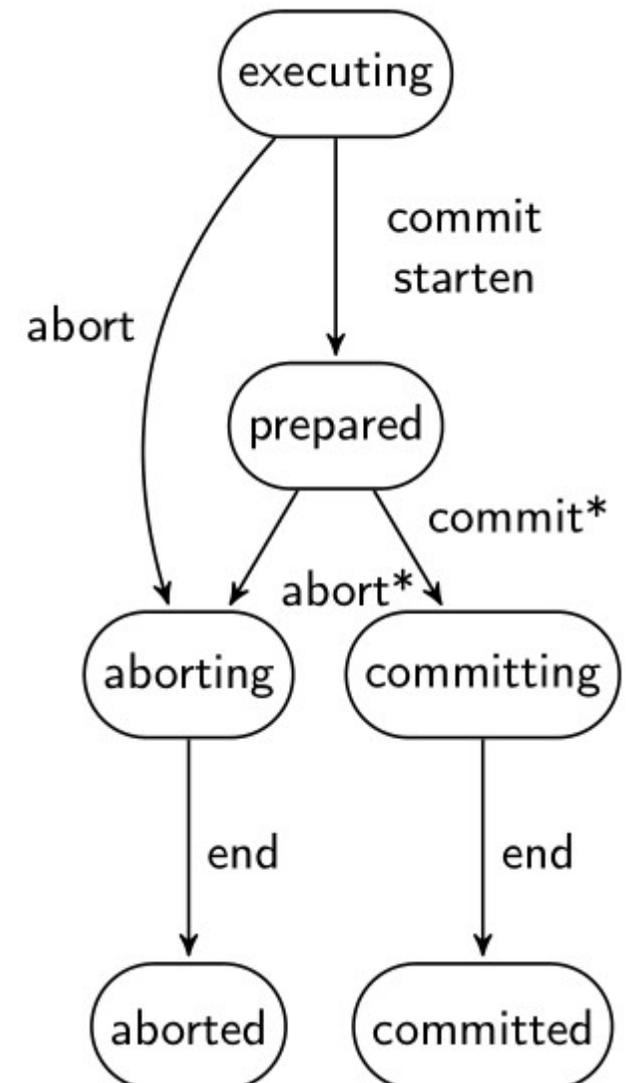
* Speicherung von Protokolleinträgen auf Platte erzwingen (force write)



- **executing**
 - RM führt Operationen durch, z. B. Änderungen an Daten. Bei Fehler kann sofort in den aborting/aborted-Zustand gewechselt werden.
 - Nach der prepare-Aufforderung durch den TM wird bei positivem Ergebnis in den prepared-Zustand gewechselt. Der RM muss auf die Entscheidung durch den TM warten.
- **prepared**
 - Abhängig von der Entscheidung des TM wird die Transaktion erfolgreich beendet oder es findet ein Rückgängigmachen aller Operationen statt.
- **aborting/aborted**
 - Transaktion wurde abgebrochen
- **committing/committed**
 - Transaktion wurde erfolgreich beendet



- **executing**
 - TM startet commit-Bearbeitung.
 - Sendet prepare-Meldung an alle RM.
 - Kann aber Transaktion auch abbrechen.
- **prepared**
 - TM wartet auf Ergebnisse von den RM.
 - Antworten alle RM positiv geht er in den Zustand committing über.
 - Antwortet nur ein RM negativ geht er in den aborting-Zustand über.
- **aborting, committing**
 - TM wartet auf Bestätigungen.
- **aborted/committed**
 - Information zur Transaktion wird in den Programmstrukturen des TM gelöscht, da sie komplett abgeschlossen ist.



- RM haben dem prepare-Aufruf mit vote yes geantwortet und danach stürzt der TM ab oder kann aus anderen Gründen nicht das Ende der Transaktion einleiten
 - Alle Sperren müssen im RM gehalten werden.
 - Andere Transaktionen auf diesem RM werden möglicherweise blockiert.
- Pragmatische Lösung für diesen Fall: RM können von sich aus, d. h. heuristisch, eine Entscheidung zum Ausgang der Transaktion treffen, entweder commit oder abort.
 - Dadurch wird die Atomarität der Transaktion durchbrochen, da der TM möglicherweise eine andere Entscheidung zum Ausgang getroffen hat.
 - In diesem Fall muss ggf. manuell wieder ein konsistenter Gesamtzustand des Systems hergestellt werden.