ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

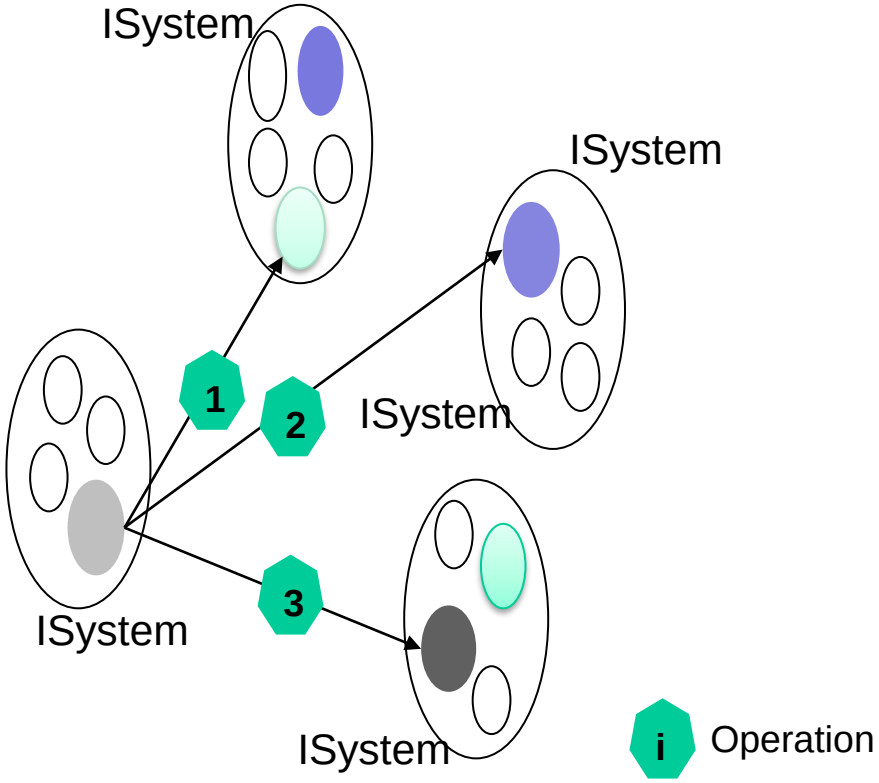
Transacções Distribuídas

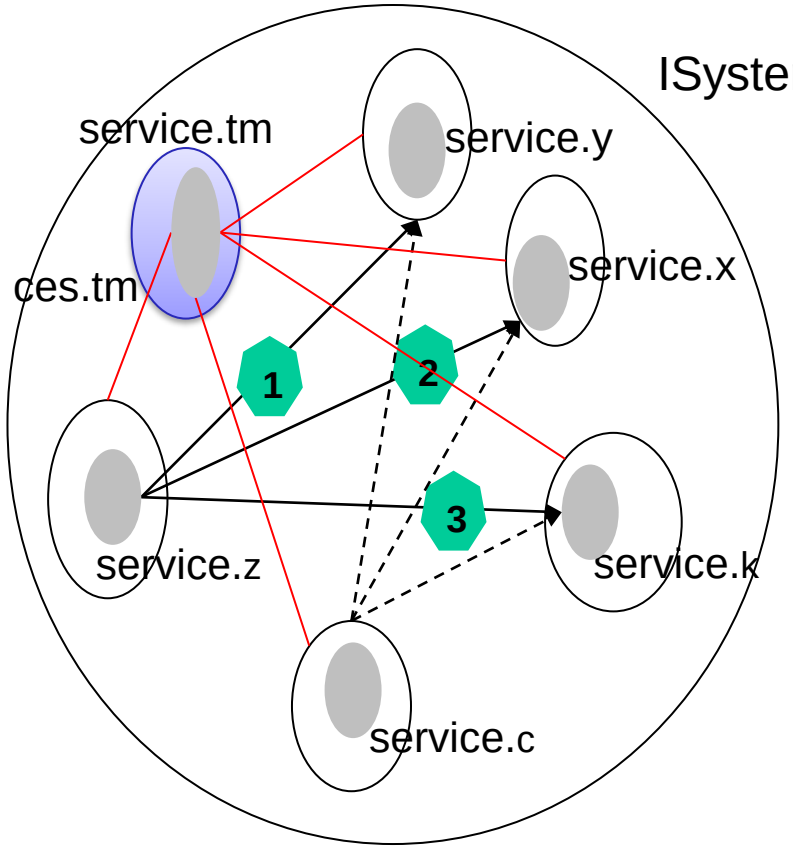
Coordenação em (Infraestruturas de) Sistemas Distribuídos

Quadro de Serviços (SOA)

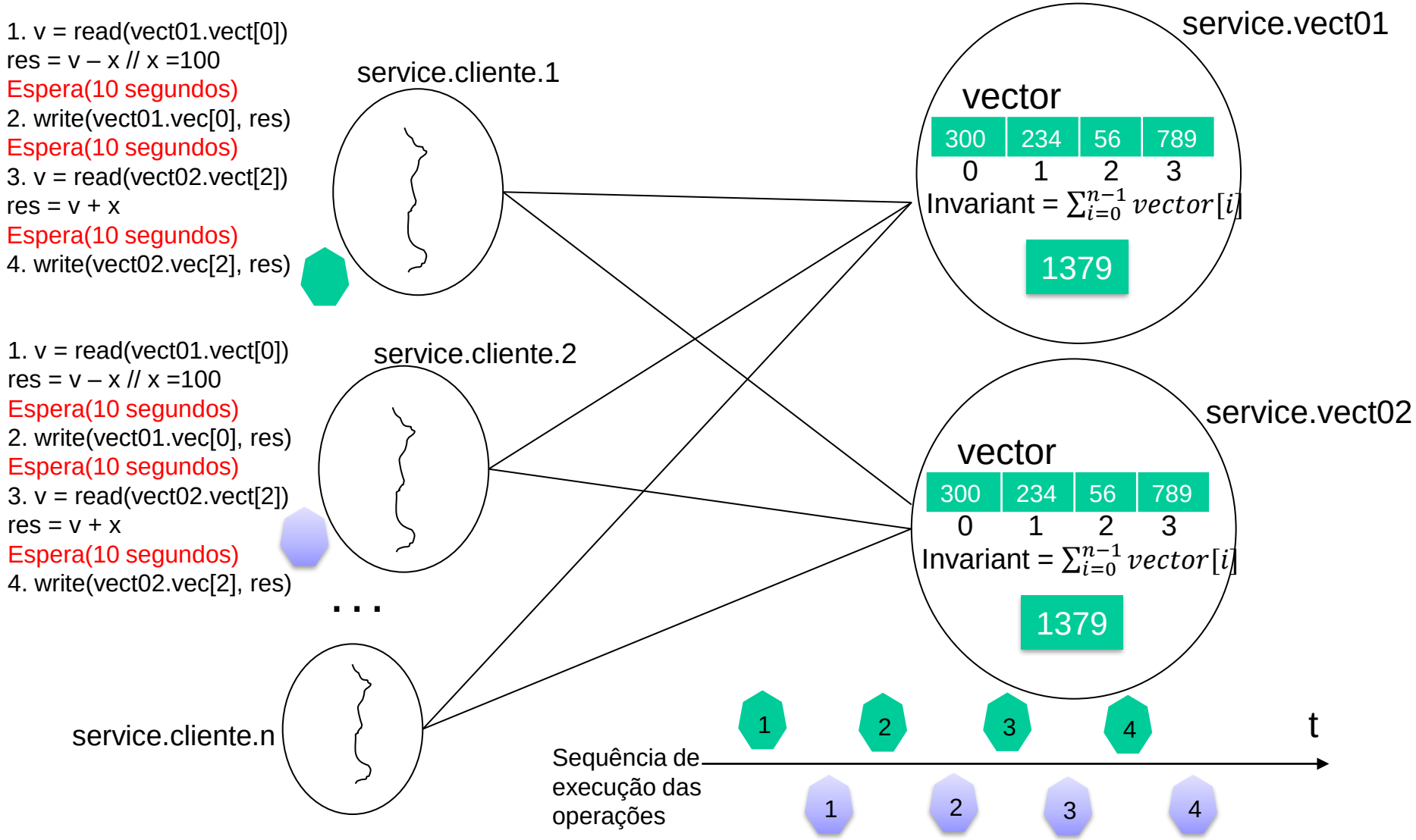
The Problem

- In a ISoS (Informatics System of Systems)
  - ISystem elements (ces) interact through services. Interactions can occur intra-ISystem (internal services) or inter-ISystems (external services)

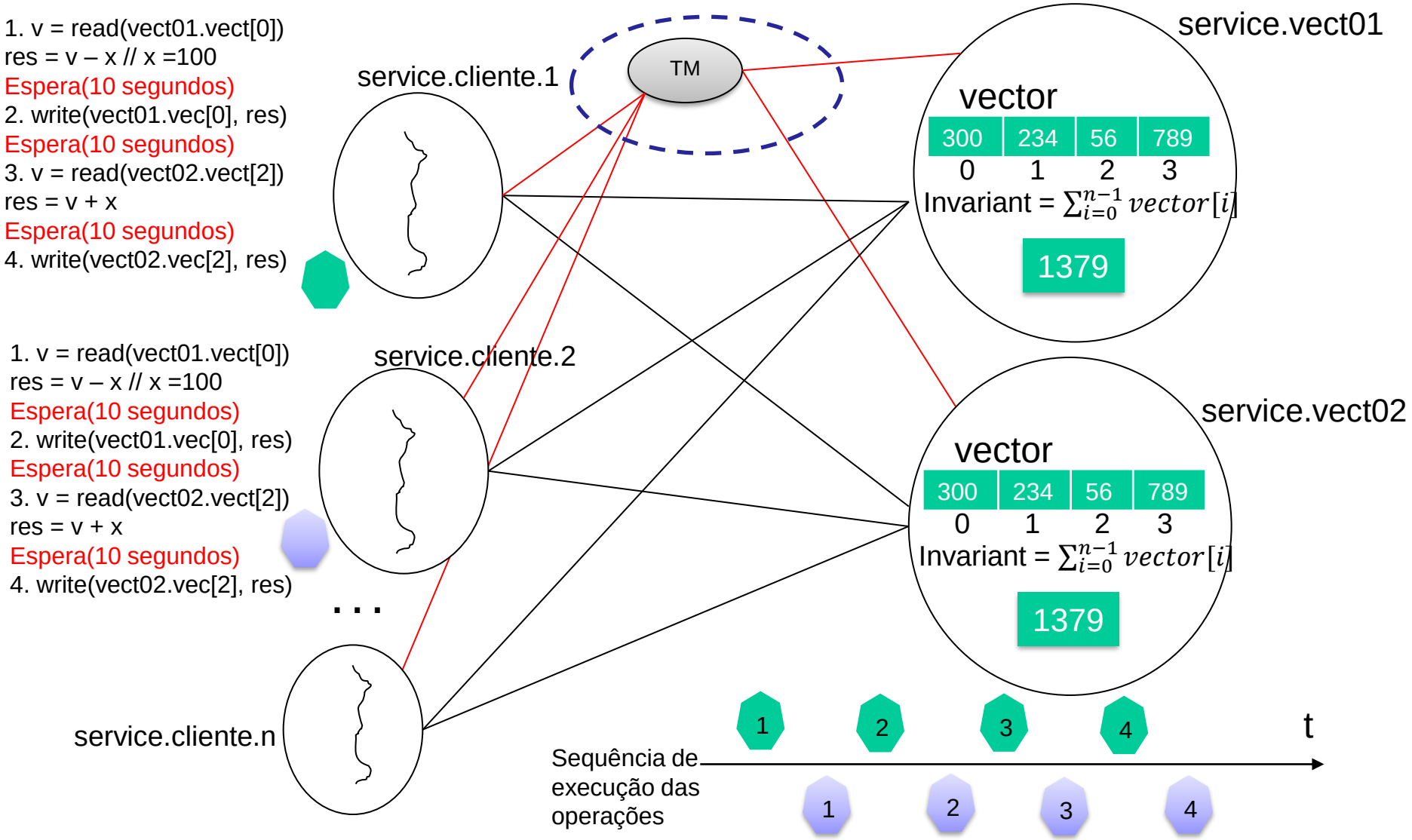




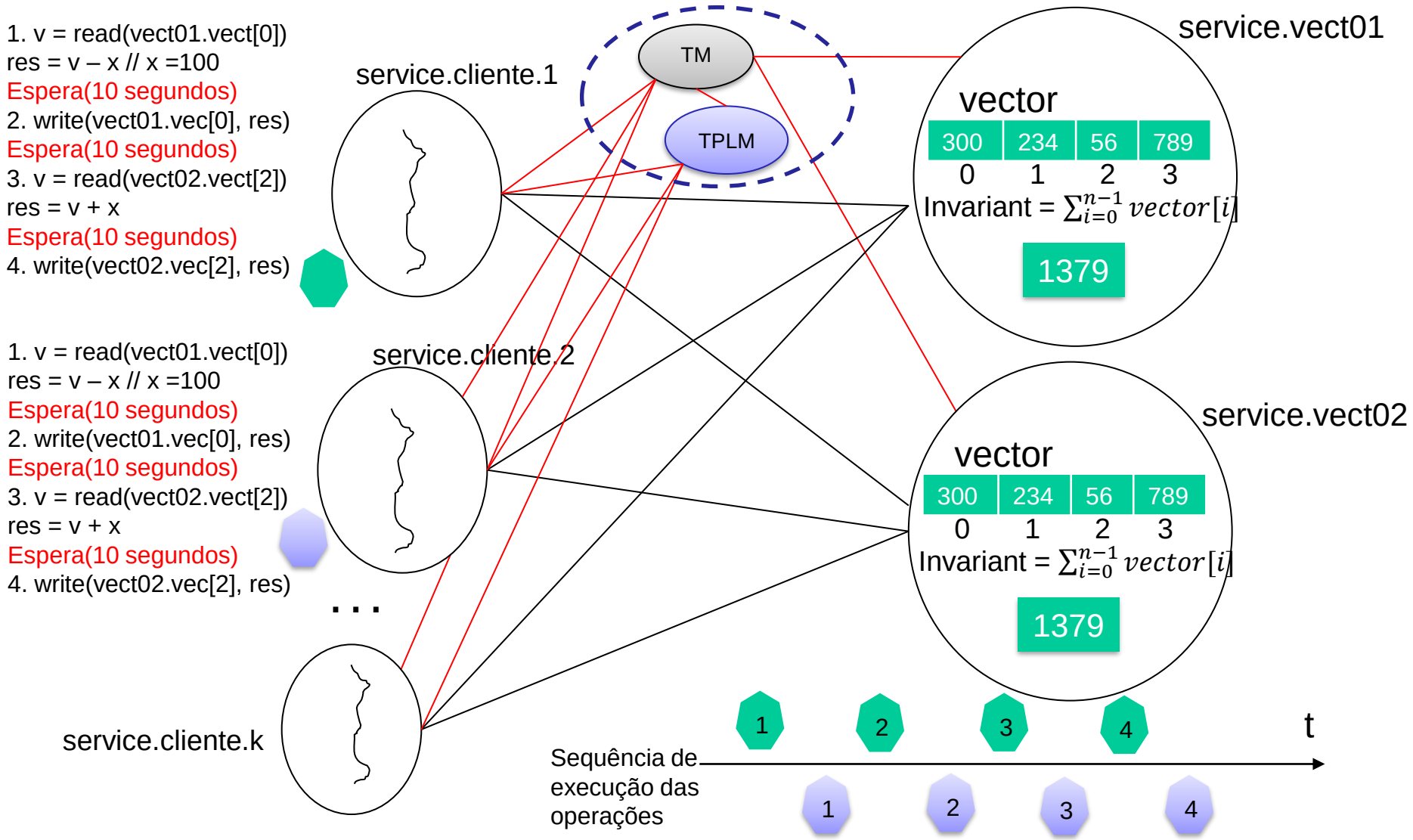
A acesso concorrente a múltiplos Service.vectxx



A acesso concorrente a múltiplos Service.vectxx



A acesso concorrente a múltiplos Service.vectxx



Definições X/Open

- Standard
  - ISO/IEC 14834:1996; Information technology - Distributed Transaction Processing - Distributed Transaction Processing: The XA Specification (link)
- Transação
  - Unidade de trabalho (computacional), constituída por duas ou mais tarefas, cuja execução se pretende atómica; ou todas as tarefas constituintes são concluídas com êxito ou nenhuma é realizada.
- Propriedades (ACID)
  - Atomicidade
    - O resultado é tudo ou nada
  - Consistência
    - Transforma um estado válido noutro estado válido
  - Isolamento
    - Mudanças nos recursos partilhados apenas são visíveis fora da transação depois de validada (*committed*)
  - Durabilidade
    - As alterações resultantes de uma transação sobrevivem para além da falha

Definições X/Open (1)

- Transacção global (*Global Transaction*)
  - Descreve o trabalho realizado por um conjunto de **gestores de recursos** (**RM**) enquanto participantes numa unidade de trabalho (atómica)
- Transacção distribuída
  - A transacção global poderá envolver **participantes** distribuídos (terminologia)
- Commitment
  - Acto que desencadeia o fim da transacção tornando permanente todas as alterações realizadas pelos RM envolvidos
- Rollback
  - Acto que desencadeia o fim da transacção anulando todas as alterações realizadas pelos RM envolvidos (estado anterior)

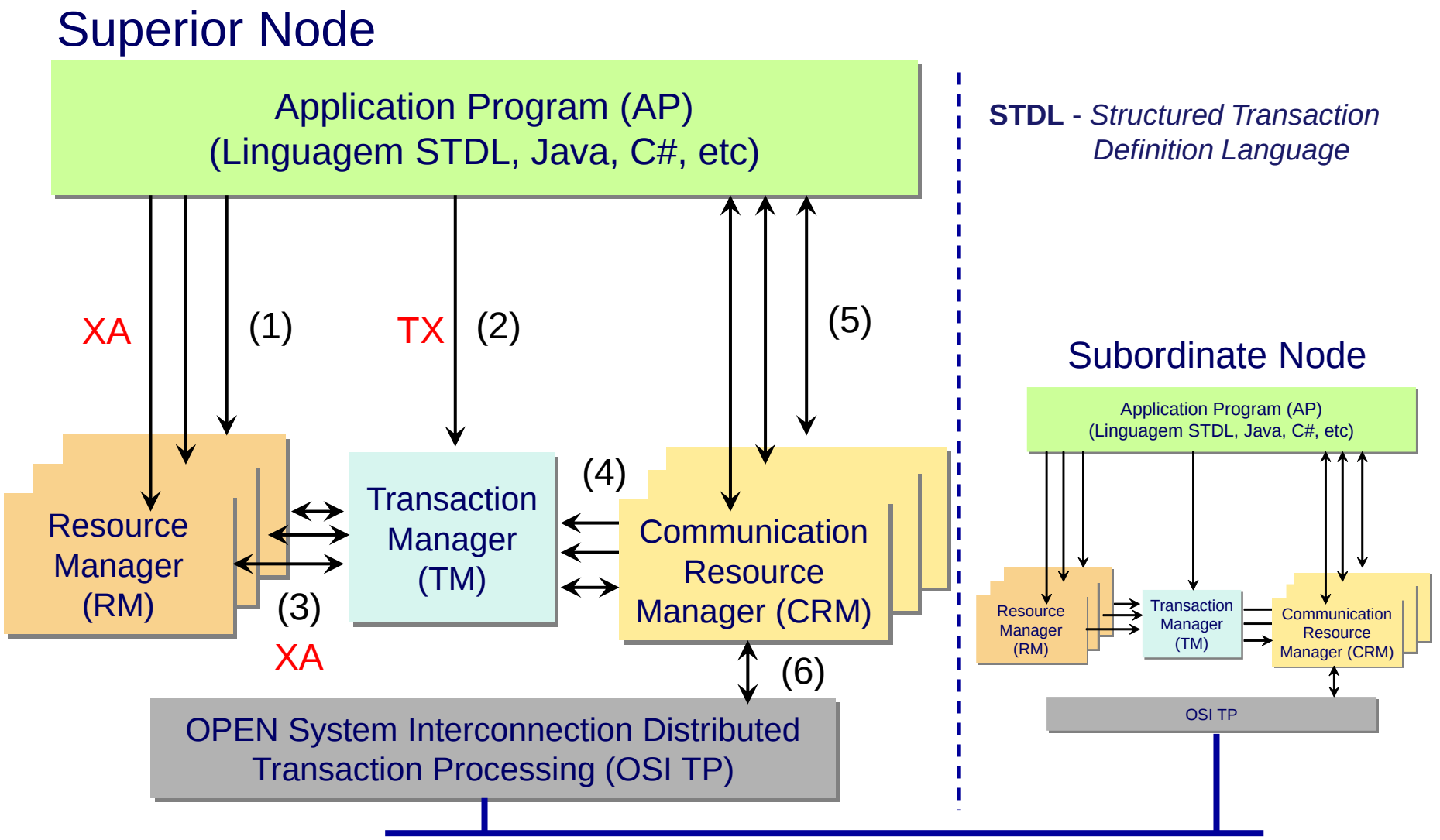
Definições X/Open (2)

- Transaction Completion
  - Refere as operações Commitment ou Rollback
- Commitment Protocol
  - Processo de coordenação quando a transação é terminada. O modelo X/Open DTP sugere o protocolo "***two-phase commit with presumed rollback***"
  - Uma transação pode ainda ser terminada com base em heurísticas (***Heuristic Transaction Completion***)
- Resource (Resource Manager)
  - Conjunto de recursos (dados) geridos por um RM (*Resource Manager*)
- RM Interface Nativa
  - *Application Programming Interface* (API) disponibilizada por um RM aos AP (*Application Programs*). Um RM deve suportar uma interface standard

X/Open Distributed Transaction Processing (DTP)

- Application Program (AP)
  - Estabelece a fronteira de uma transação, a que associa um conjunto de operações a executar de forma atómica (todas ou nenhuma)
- Resource Managers (RM)
  - Gere um conjunto de recursos (bases de dados, ficheiros, **serviços**, etc.)
- Transaction Manager (TM)
  - Gere transações com um identificador único associado, monitoriza-as, é responsável pela sua conclusão e coordenação de falhas e recuperação
- Communication Resource Managers (CRMs)
  - Gere a comunicação entre diferentes gestores de transações (*Transaction Managers*)

Modelo funcional X/Open DTP



Questões de Investigação

- Evolução da especificação para uma lógica (padrão arquitetural) SOA
  - Transaction Manager um elemento Serviço
    - E se for implementado sobre diferentes quadros tecnológicos, e.g., ecossistema Java ou Microsoft?
    - Qual o protocol (wire-protocol) para a cooperação entre Service (client) e Service (provider, o TM)?
  - Coordenação da concorrência
    - Funcionalidades integradas no mesmo element Service (TM)?
    - Ou um element oServiço Autónomo
- Elemento Service ou ISystem
  - Como estruturar as responsabilidades computacionais?
- Lidar com a Heterogeneidade (interoperabilidade)
  - Elementos Serviço cliente e elementos Serviço servidores (fornecem serviços computacionais; como garantir?
    - Cooperação (como *interoperam*); elementos Serviço heterogéneos
    - Transparência à localização
    - De entre outras questões em aberto

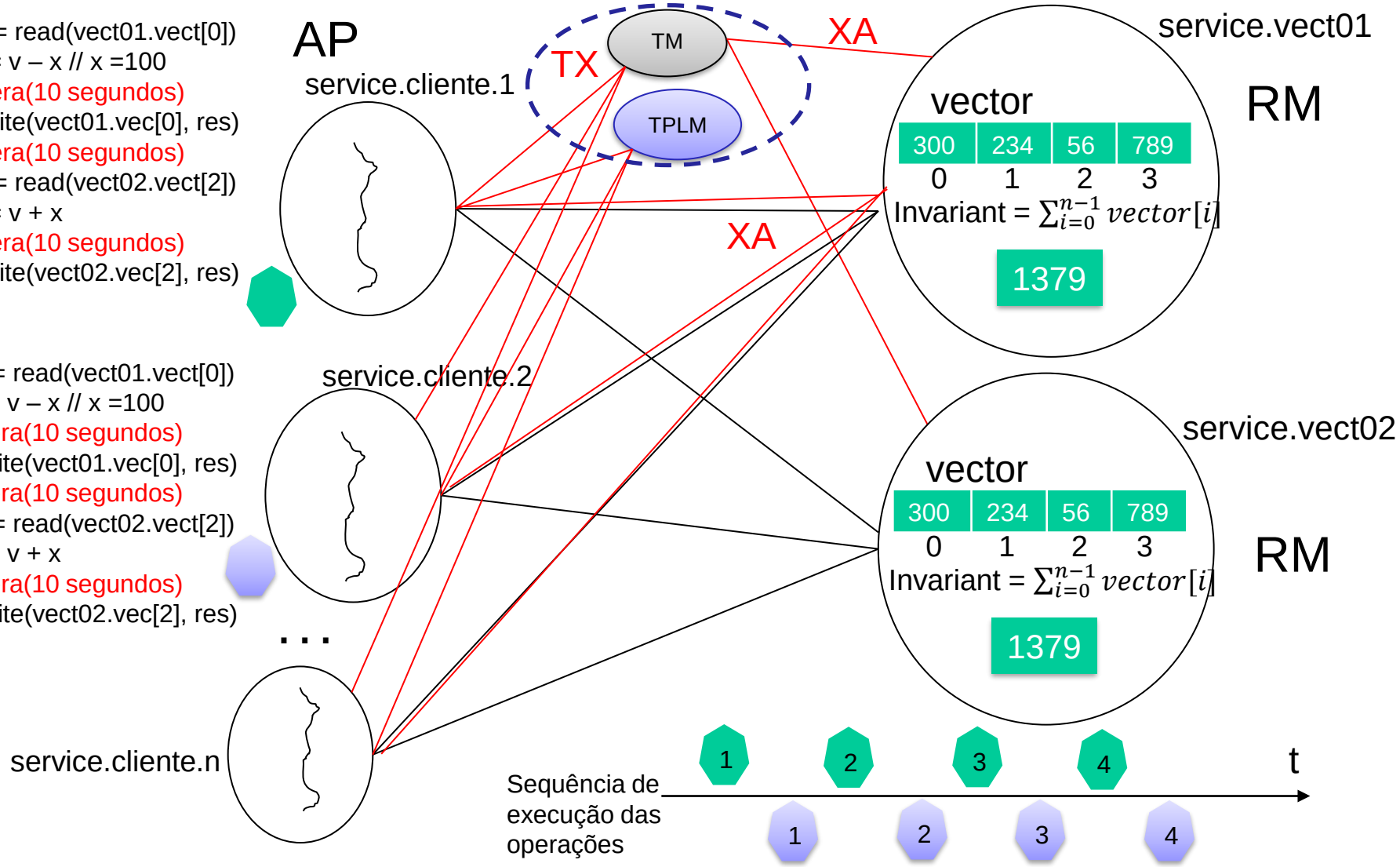
Principais especificações de interface entre componentes

- (1) AP↔RM
  - Especificação XA
- (2) AP↔TM
  - Especificação TX (*Transaction Demarcation Specification*)
- (3) TM↔RM
  - Especificação XA
- (4) TM↔CRM
  - Especificação XA+
- (5) AP↔CRM
  - TxRPC e outras infra-estruturas
- (6) CRM↔OSI TP
  - Especificação XAP-TP

A acesso concorrente a múltiplos Service.vectxx

1. v = read(vect01.vect[0])
res = v – x // x =100
Espera(10 segundos)
2. write(vect01.vec[0], res)
Espera(10 segundos)
3. v = read(vect02.vect[2])
res = v + x
Espera(10 segundos)
4. write(vect02.vec[2], res)

1. v = read(vect01.vect[0])
res = v – x // x =100
Espera(10 segundos)
2. write(vect01.vec[0], res)
Espera(10 segundos)
3. v = read(vect02.vect[2])
res = v + x
Espera(10 segundos)
4. write(vect02.vec[2], res)



Interface TX (AP <-> TM)

Operações	**Descrição**
tx_begin	Inicia uma transacção global
tx_close	Encerra um conjunto de *resource managers* (RM)
tx_commit	*Commit* de uma transacção global
tx_info	Retorna informação sobre uma transacção global
tx_open	Abre um conjunto de *resource managers* (RM)
tx_rollback	*Rollback* de uma transacção global
tx_set_commit_return	Activa/desactiva retorno antecipado de *tx_commit*
tx_set_transaction_control	Activa/desactiva encadeamento de transacções
tx_set_transaction_timeout	Posiciona o tempo máximo para execução de uma transacção

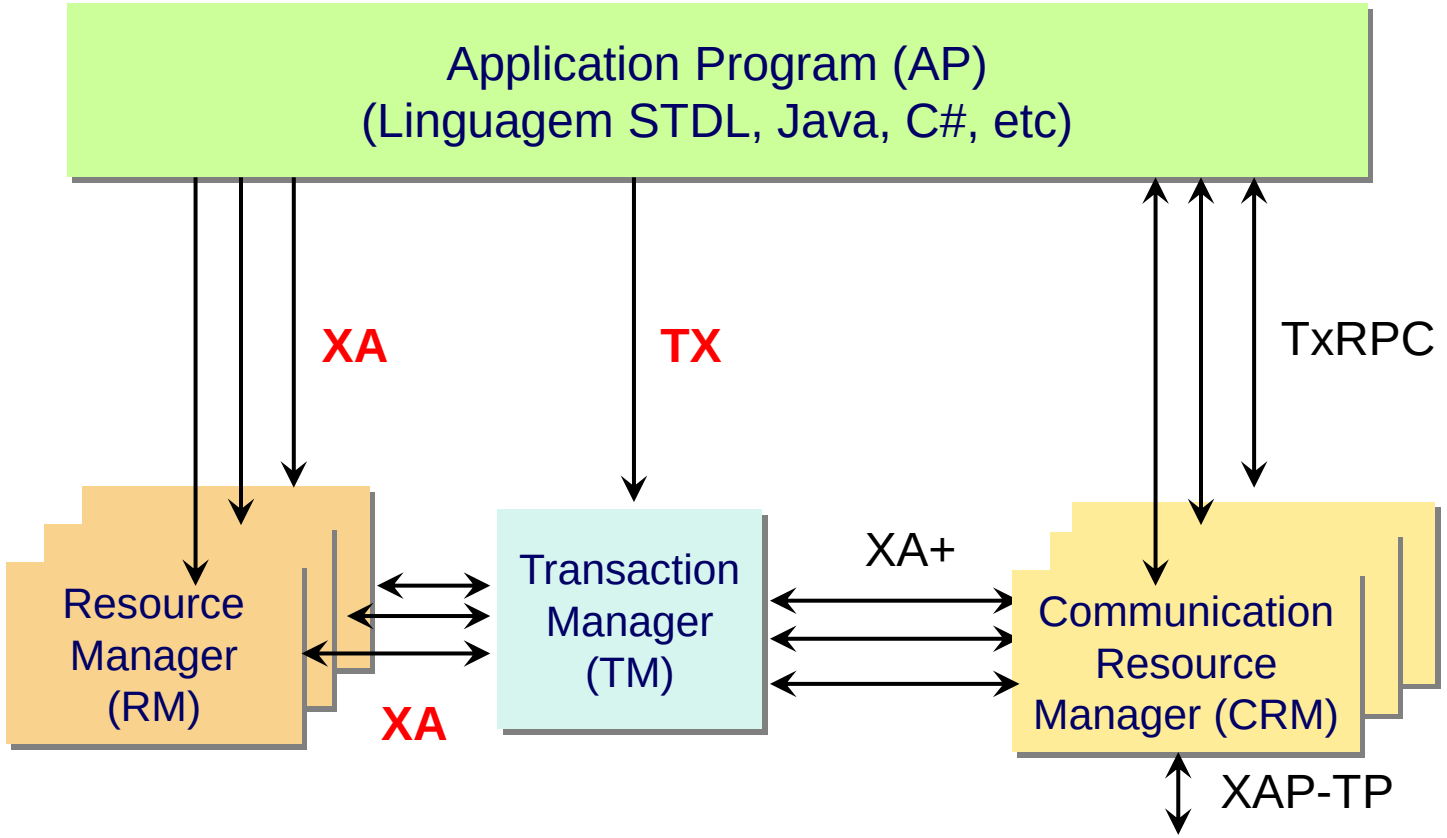
A estrutura XID (linguagem C)

```
#define XIDDATASIZE 128 /* size in bytes */
struct xid_t {
  long formatID; /* format identifier */
  long gtrid_length; /* value not to exceed 64 */
  long bqual_length; /* value not to exceed 64 */
  char data[XIDDATASIZE]; /* may contain binary data
*/
};
typedef struct xid_t XID;

/*
* A -1 in formatID means that the XID is null.
*/
```

Interface XA (AP <-> RM, RM <-> TM)

Operations	Description
ax_reg	The XA RM calls ax_reg() to inform a TM that it is about to do work
ax_unreg	The XA RM calls ax_reg() to inform a TM exits the association
xa_close	A TM calls xa_close() to close a currently open resource manager
xa_commit	A TM calls xa_commit() to commit the work associated with XID
xa_complete	A TM calls xa_complete() to wait / completion of an asynchronous operation
xa_end	A TM calls xa_end() if an AP finishes/suspend work on a transaction
xa_forget	A TM calls xa_forget() to forget a heuristically completed transaction
xa_open	A TM calls xa_open() for RM to open and prepare for XA distributed transaction
xa_prepare	TM calls xa_prepare() for RM to prepare commitment of performed work
xa_recover	A TM calls xa_recover() to obtain the transactions in prepared state
xa_rollback	A TM calls xa_rollback() to roll back work performed
xa_start	A TM calls xa_start() to inform a RM that an application may do work

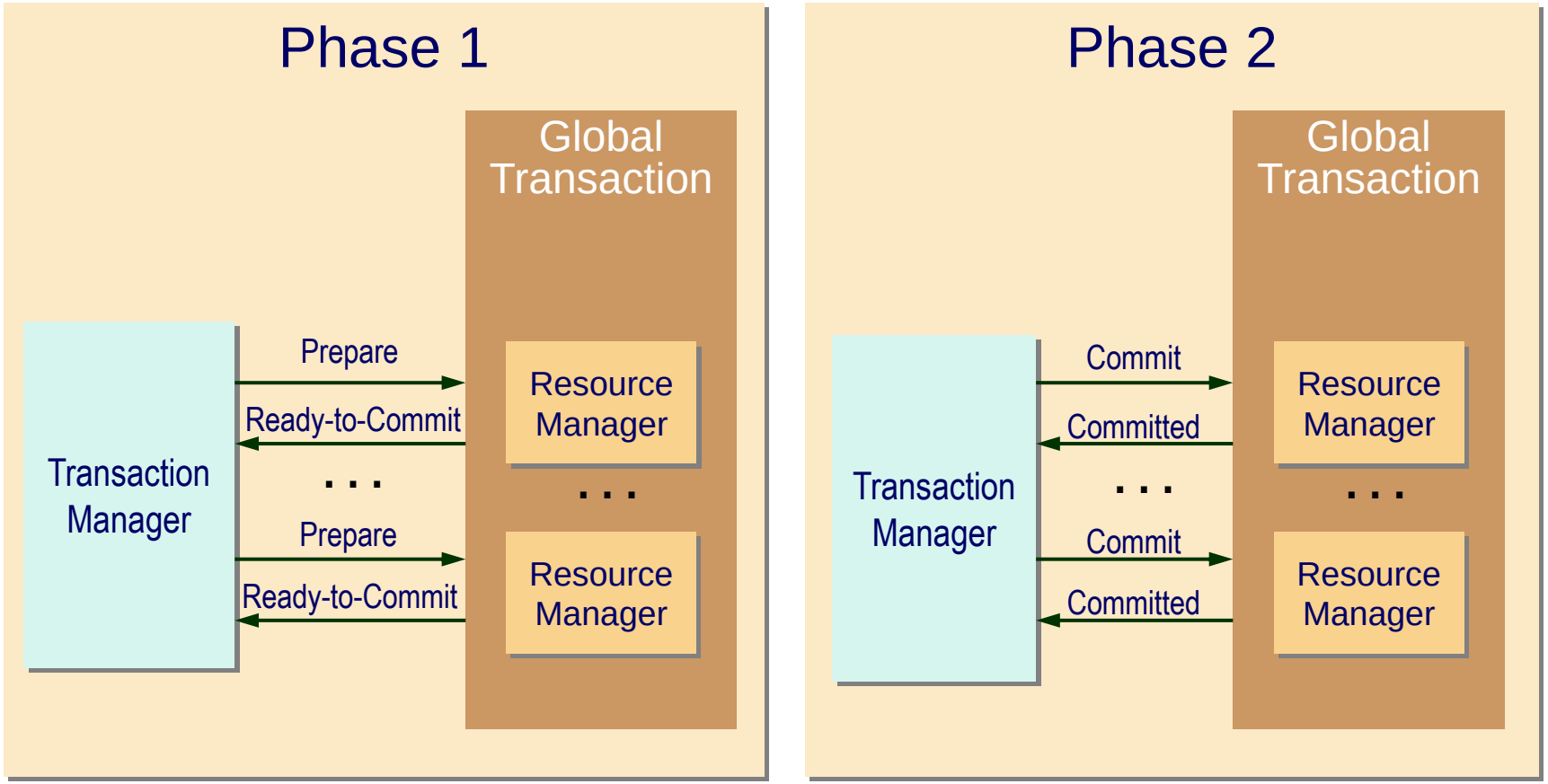
Protocolos TX e XA



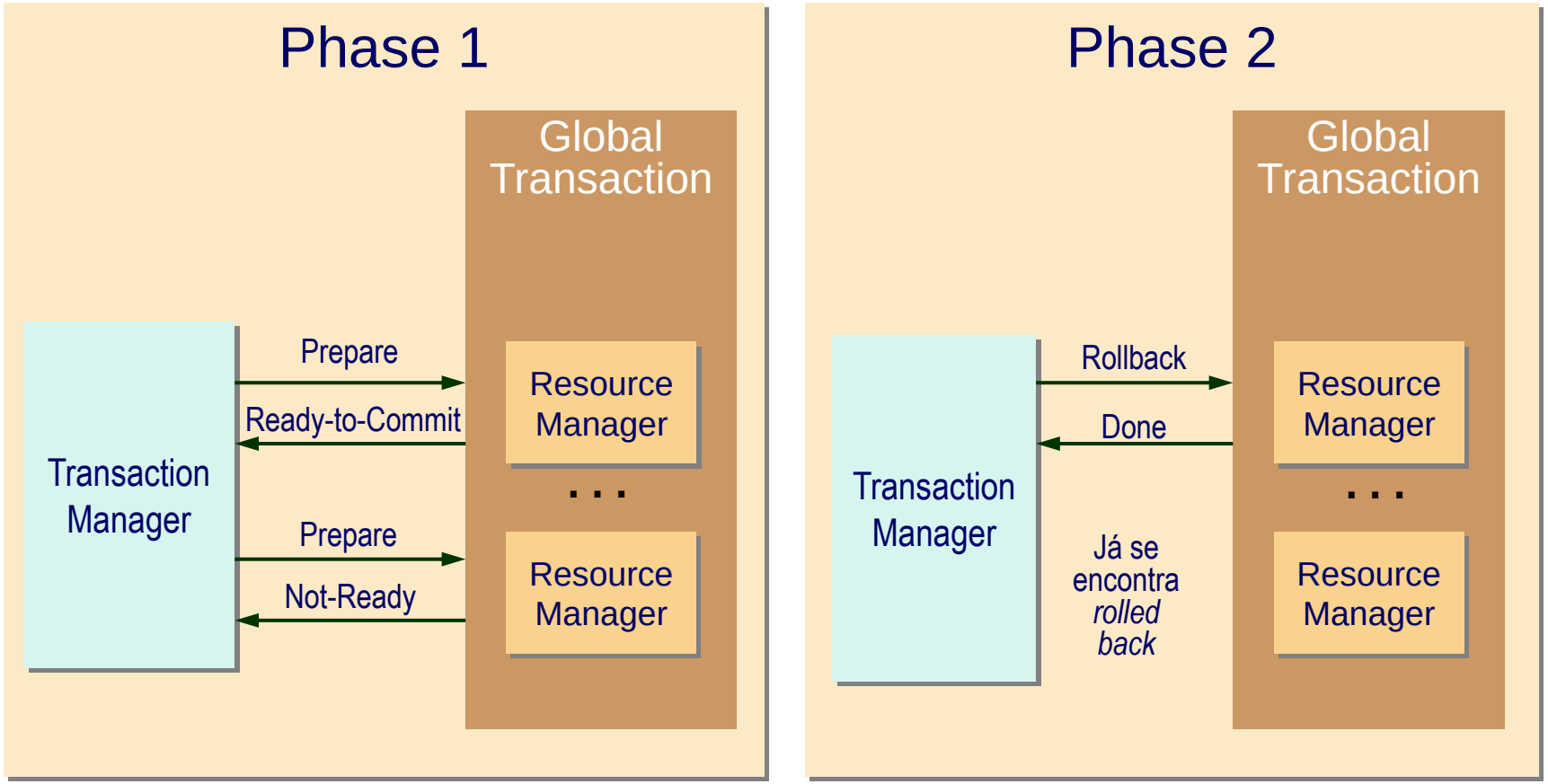
Two Phase Commit Protocol (2-PC)

- Phase 1
  - O TM chama *xa_prepare()*; pede a cada RM participante que se prepare para o *commit*
  - Se pode realizar *commit*, responde afirmativamente ao TM
- Phase 2
  - Dependendo das respostas recebidas durante a primeira fase, o TM pode ou chamar *xa_commit()* ou *xa_rallback()*
  - Se todos os RM responderam afirmativamente o TM chamará o método *xa_commit()*
  - Se algum não respondeu afirmativamente o TM chamará o método (operação) *xa_rollback()*
- Optimizações definidas na especificação XA
  - Um RM abandona a participação numa transacção se não realizou qualquer atualização (escrita)
  - Um TM pode usar *one-phase-commit* se apenas envolve um RM

Situação de sucesso no modelo *two-phase commit*



Situação de insucesso no modelo *two-phase commit*



“Transaction Branch” (ramo de uma transação)

- Uma transação global pode envolver mais do que um ramo (*branch* ou unidade de trabalho)
  - Quando uma aplicação (sistema informático) envolve múltiplos processos ou múltiplas aplicações remotas
- Um ramo é uma parte do trabalho no suporte de uma transação global no qual o RM e o TM cooperam para a sua validação
- Um identificador único, XID, identifica uma transação global e um ramo (*branch*)

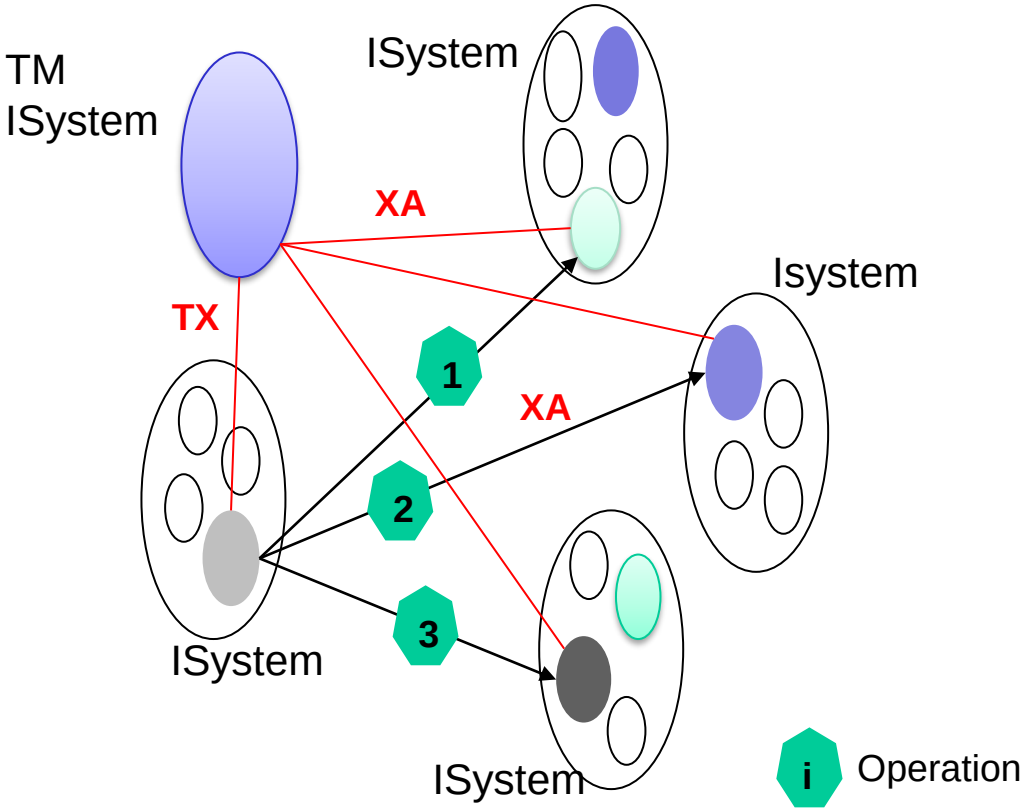
Outros Standards e sistemas Transaccionais de referência

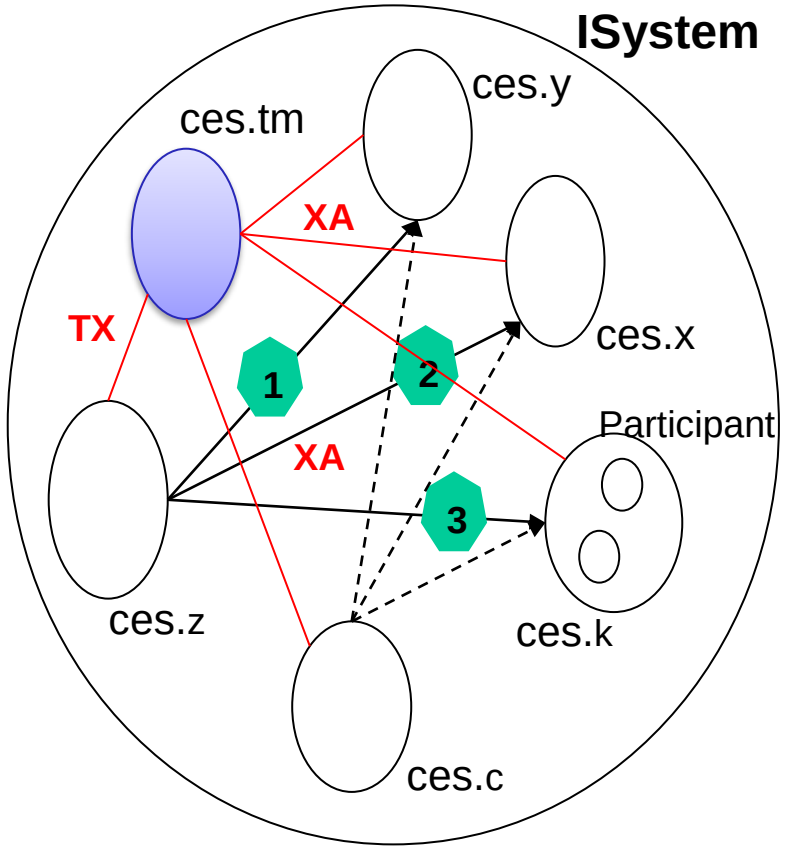
- CORBA (OMG)
  - Object Transaction Service (OTS)
- TUXEDO (BEA, agora ORACLE)
  - Portable Transactions Processing Monitor
- ACMS (Digital, actualmente HP)
  - Application Control and Management System; Transaction Processing System
- Encina (Carnegie-Mellon University)
  - Desenvolvimento académico de um Transaction Processing Monitor (TPM)
  - Comercializado por Transarc, actualmente da IBM
- CICS (Customer Information Control System) e IMS (*Information Management System*) da IBM
  - Transactions Processing Monitors

Detalhes em: Principles of Transaction Processing, por Philip A. Bernstein - *Chap.* 5

Solution based on a Transaction Manager (TM)

- In a ISoS (Informatics System of Systems)
  - Isystem elements (ces) interact through services. Interactions can occur intra-ISystem (internal services) or inter-ISystems (external services)



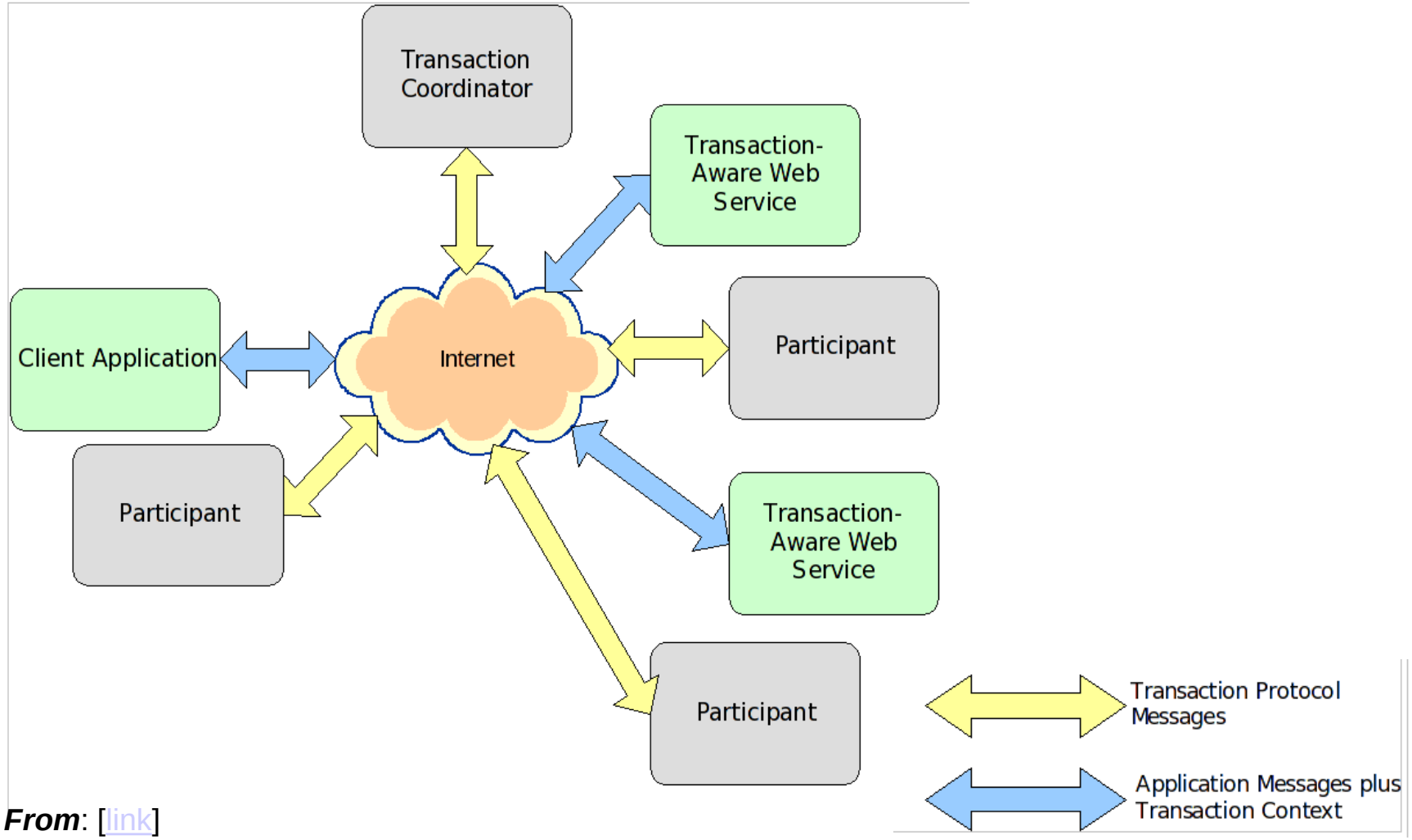


TRANSAÇÕES - WEB SERVICES

O consórcio de normalização Open Group (X/Open)

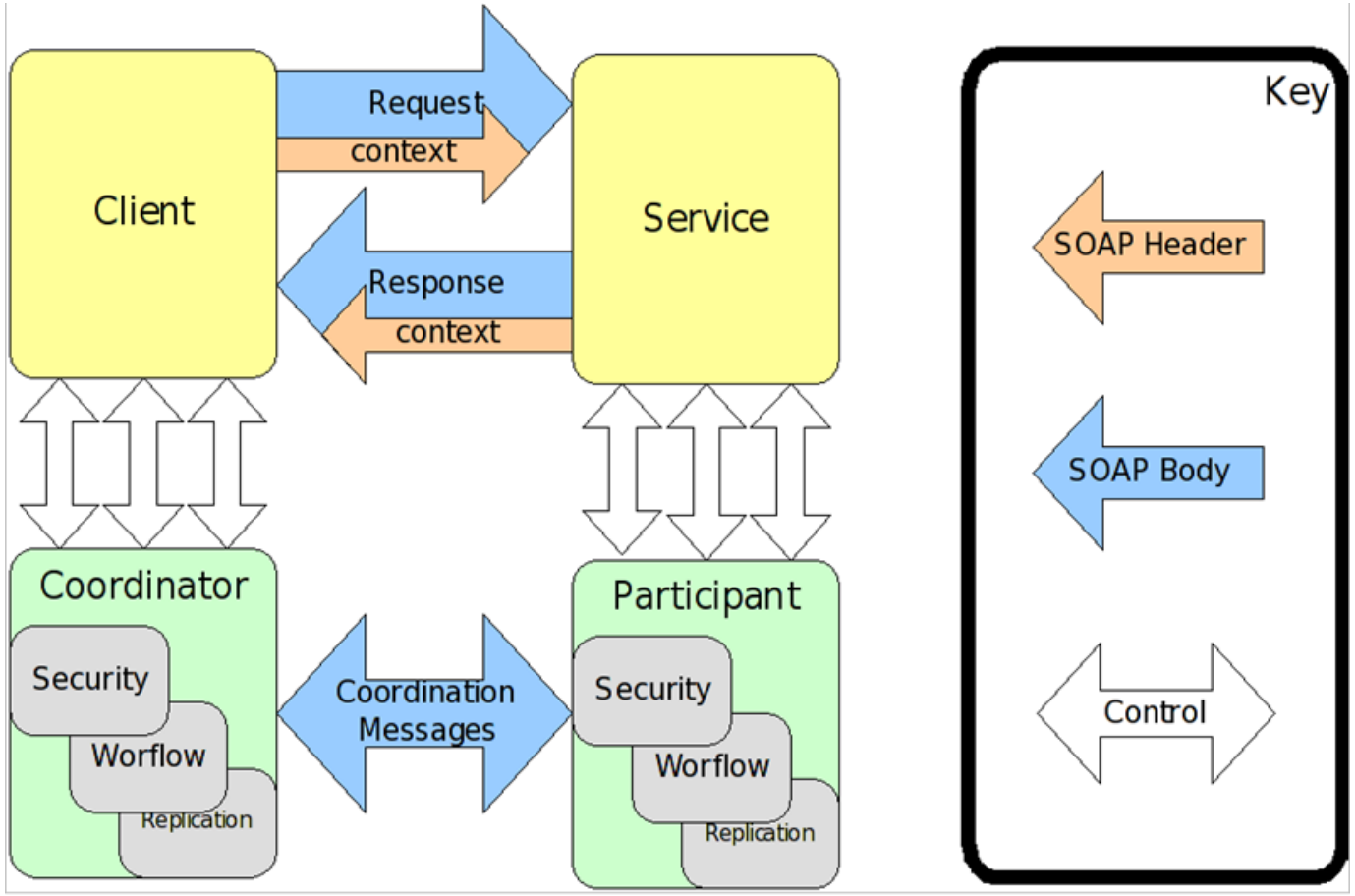
- Facilitador no acesso a informação integrada e interoperabilidade global na base de standards abertos (http://www.opengroup.org)
- A estratégia baseia-se no estabelecimento de um *ecosistema* designado por - *Common Applications Environment* (CAE)
- Tipos de especificações X/Open
  - Especificações CAE
  - Especificações preliminares
  - Guias
  - Estudos técnicos
  - Documentos de trabalho (*Snapshots)*

WS-Transaction Architecture



From: [link]

Standard W3C WS-Coordination



From: [link]

Web Services-Coordination (WS-C) Architecture

TRANSACÇÕES NO JINI
SERVIÇO MAHALO (TRANSACTION SERVER)

Estratégia da especificação TX - Jini Transactions

Sistemas Tradicionais

- Baseiam-se em TPM (*Transaction Processing Monitor*) os quais asseguram que a implementação da semântica transacional é cumprida por todos os participantes numa transação.

No servidor transaccional Mahalo

- É da responsabilidade de cada participante (serviço/objeto) a implementação da semântica transacional.
- O servidor de gestão de transações (TM) apenas é responsável por disponibilizar os mecanismos de coordenação do conjunto de serviços participantes (objetos), necessários para que acordem na transação.
- Opta pelo conjunto mínimo de protocolos e interfaces que permite aos serviços participantes a implementação da semântica transacional.
- Em oposição de um conjunto (máximo) de interfaces e protocolos que implementem as políticas que assegurem a correção de qualquer semântica transacional (sistemas tradicionais referidos acima).

Transactions (net.jini.core.transaction)

- net.jini.core.transaction.Transaction
  - Interface para classes representando transacções retornadas por um TransactionManager
- net.jini.core.transaction.Transaction.Created
  - Class dos objectos retornados pelo método create de TransactionFactory
- net.jini.core.transaction.TransactionFactory
  - Responsável pela criação de transacções

Transactions (net.jini.core.transaction) (1)

- net.jini.core.transaction.server.TransactionManager
  - Interface para um servidor transaccional two-phase-commit
- net.jini.core.transaction.server.NestableTransactionManager
  - Interface para um servidor transaccional two-phase-commit hierárquico
- net.jini.core.transaction.server.TransactionParticipant
  - Interface para os clientes de um servidor transaccional (nas chamadas deste)

Transactions (net.jini.core.transaction) (2)

net.jini.core.transaction.server.TransactionConstants

- ABORTED
  - Transaction has been aborted.
- ACTIVE
  - Transaction is currently active.
- COMMITTED
  - Transaction has been committed.
- NOTCHANGED
  - Transaction has been prepared with nothing to commit.
- PREPARED
  - Transaction has been prepared but not yet committed.
- VOTING
  - Transaction is determining if it can be committed

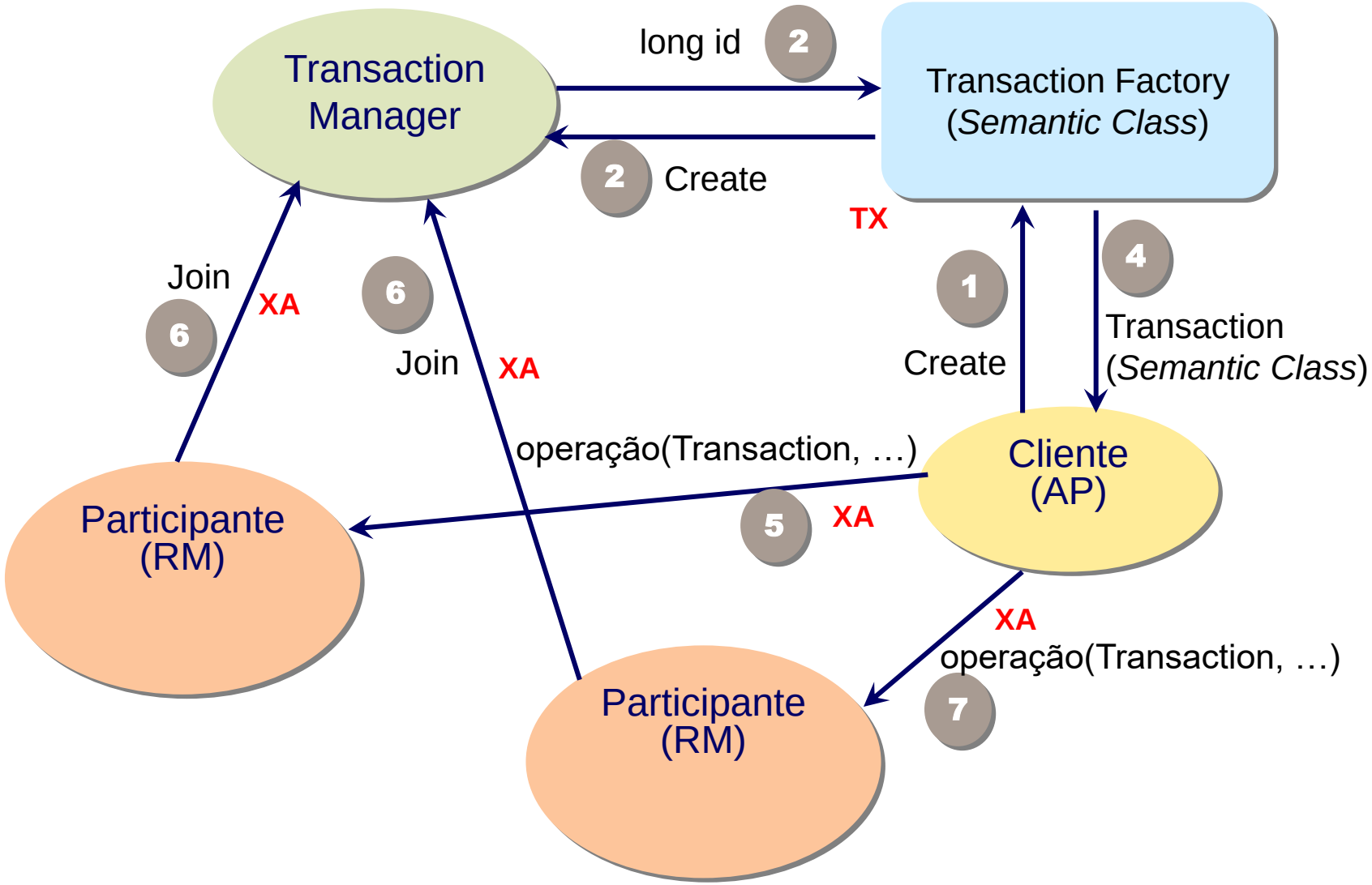
Interface implementada pelo Transaction Manager

```
package net.jini.core.transaction.server;

public interface TransactionManager  extends Remote, TransactionConstants {

    public static class Created implements Serializable {
        public final long id;
        public final Lease lease;
        public Created(long id, Lease lease) {...}
    }
    Created create(long leaseFor) throws LeaseDeniedException, RemoteException;

    void join(long id, TransactionParticipant part, long crashCount)
            throws UnknownTransactionException, CannotJoinException,
                    CrashCountException,  RemoteException;

    int getState(long id) throws UnknownTransactionException, RemoteException;

    void commit(long id) throws UnknownTransactionException, CannotCommitException, RemoteException;
    void commit(long id, long waitFor) throws UnknownTransactionException, CannotCommitException,
                                              TimeoutExpiredException, RemoteException;

    void abort(long id) throws UnknownTransactionException, CannotAbortException, RemoteException;
    void abort(long id, long waitFor) throws UnknownTransactionException,
                                            CannotAbortException, TimeoutExpiredException, RemoteException;
}
```

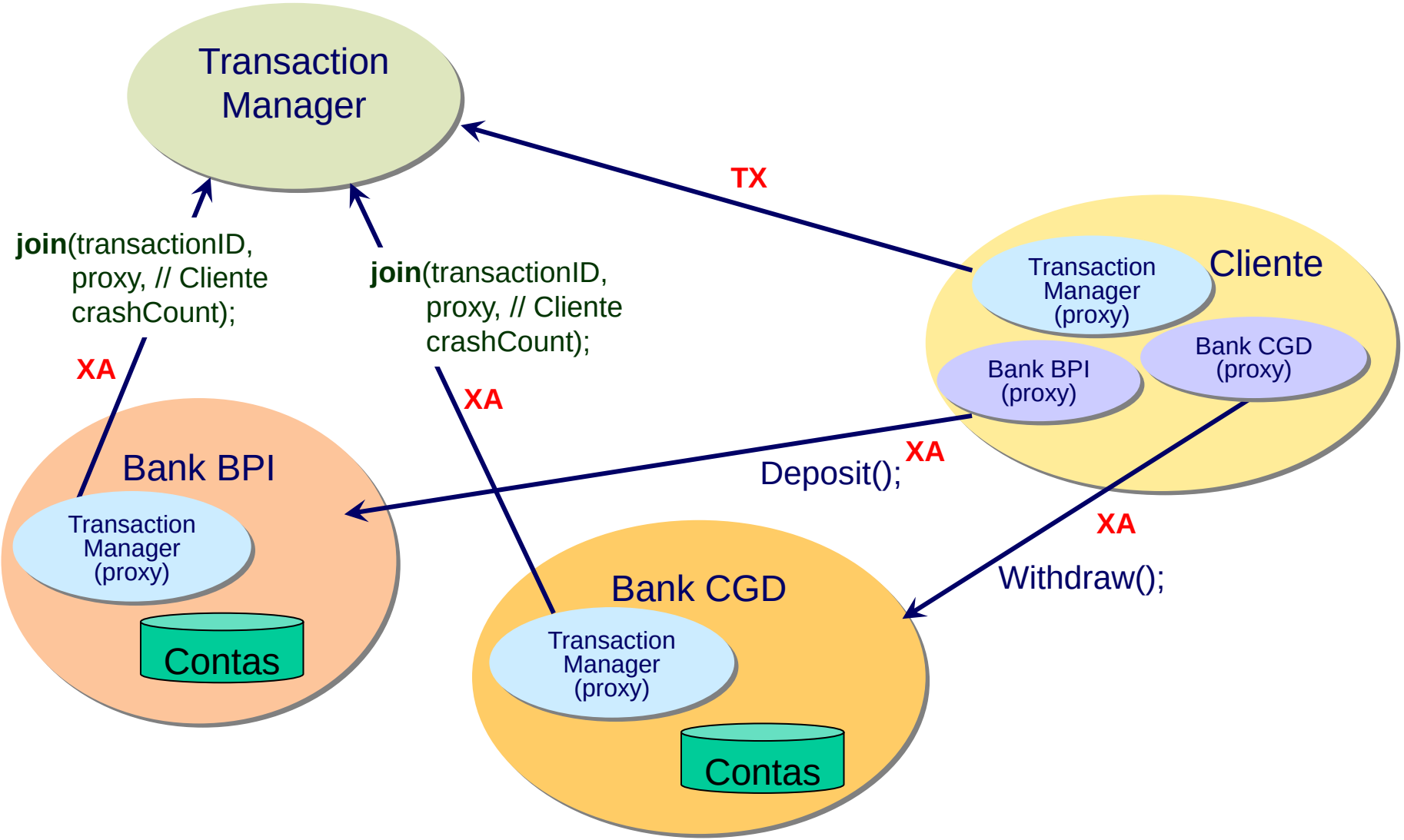
Interface a ser implementada pelos participantes

```
package net.jini.core.transaction.server;

public interface TransactionParticipant  extends Remote, TransactionConstants {

    int prepare(TransactionManager mgr, long id)
                                throws UnknownTransactionException, RemoteException;

    void commit(TransactionManager mgr, long id)
                                throws UnknownTransactionException, RemoteException;

    void abort(TransactionManager mgr, long id)
                                throws UnknownTransactionException, RemoteException;

    int prepareAndCommit(TransactionManager mgr, long id)
                                throws UnknownTransactionException, RemoteException;

}
```

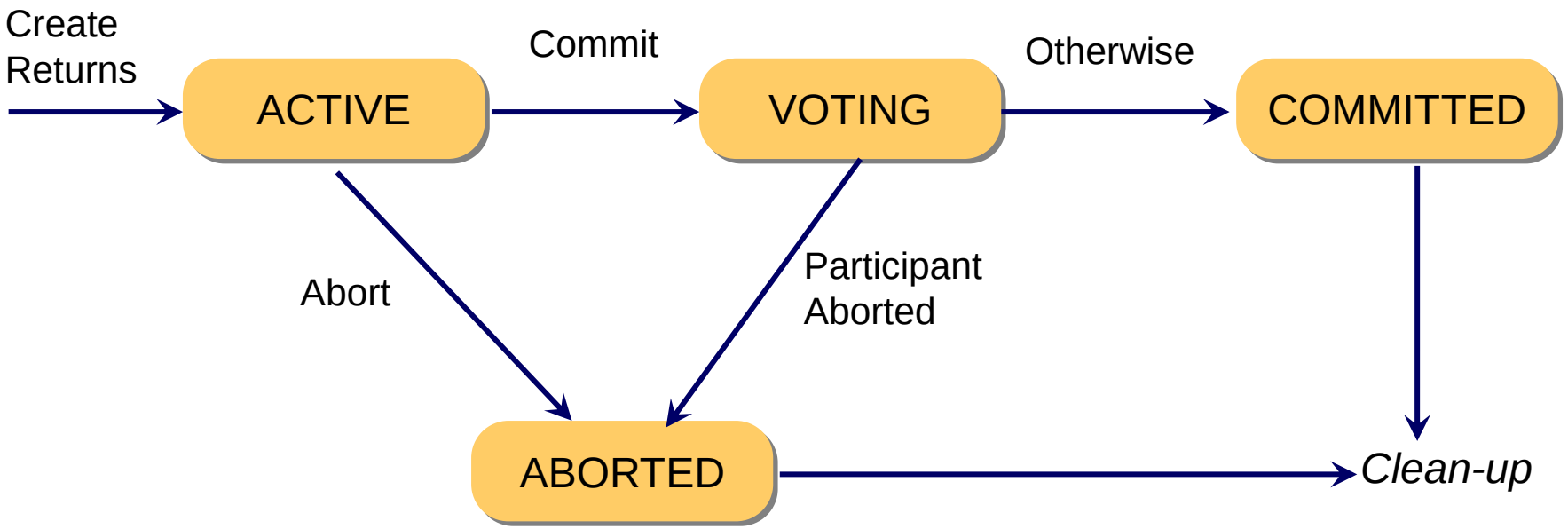
Criação e utilização de transacções



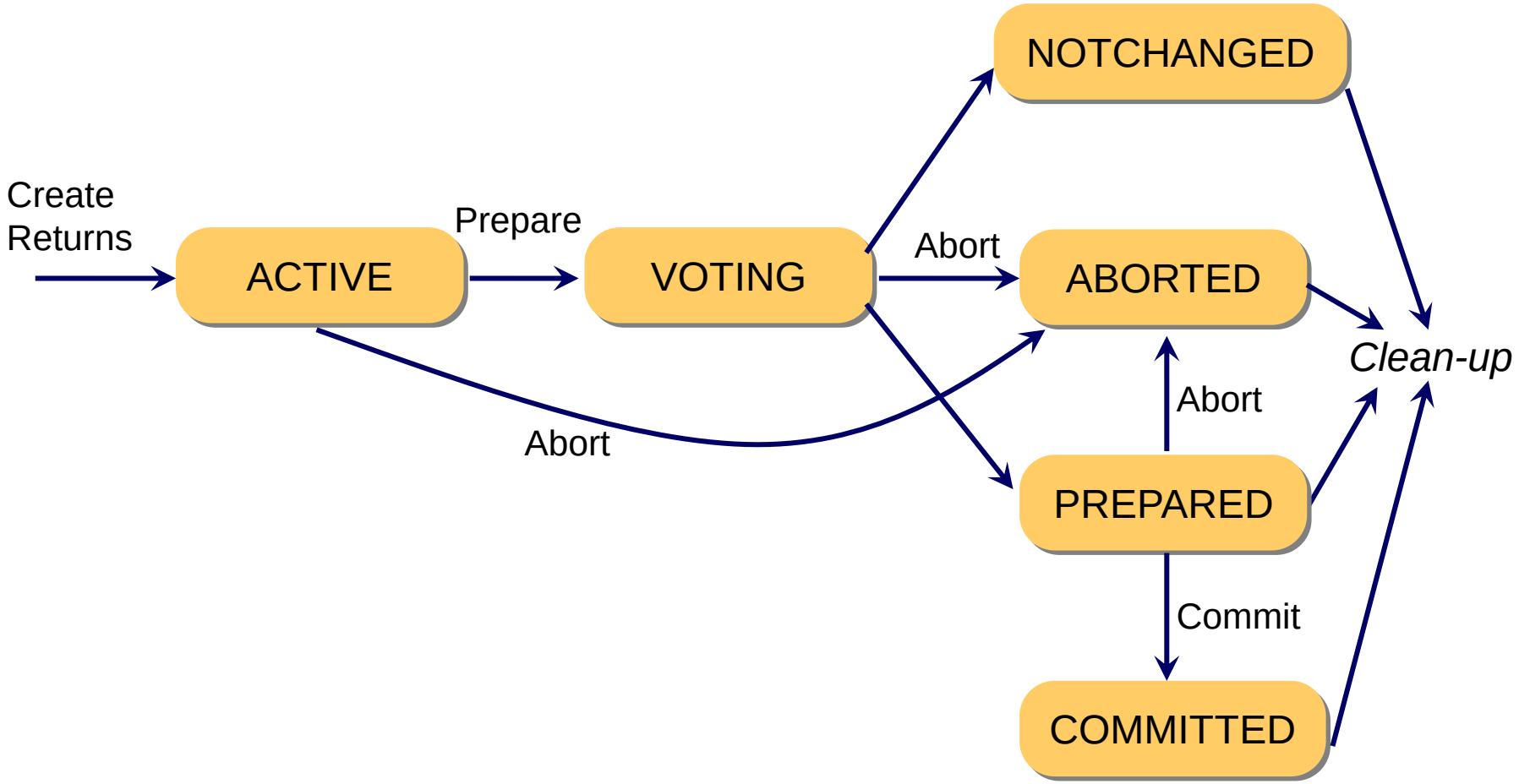
Exemplo de Transferência entre bancos



Fim de transacção na perspectiva de um cliente



Fim de transacção na perspectiva de um participante



Fim de transacção na perspectiva do Transaction Manager