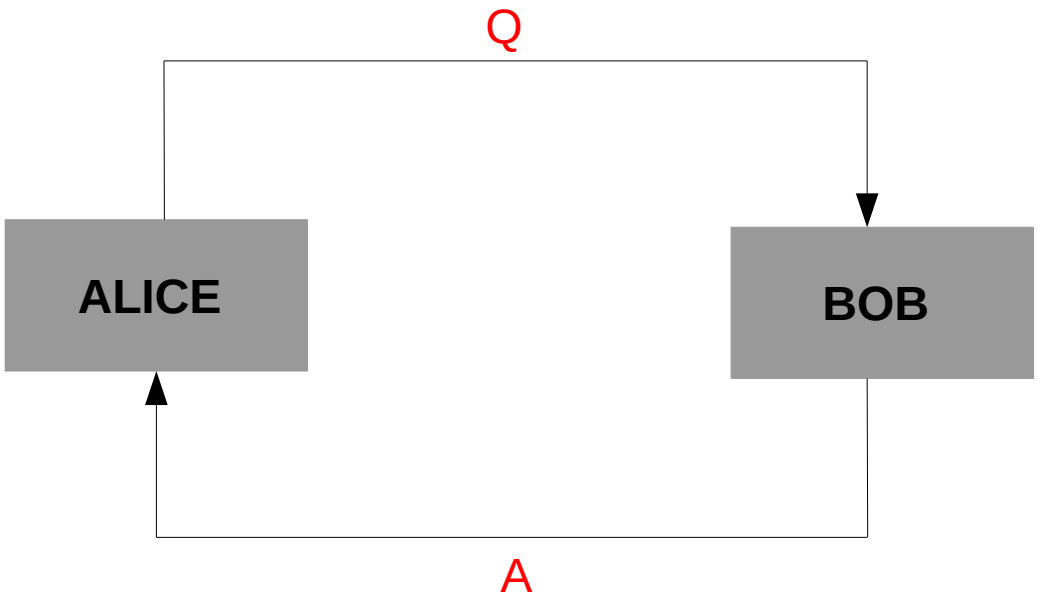
Transazioni

La transazione può essere vista come uno scambio di messaggi tra due o più entità che comunicano all'interno di un sistema.
Una transazione viene detta "coerente" se ha avuto un buon esito. Cioè ci chiediamo se una transazione sia avvenuta nel modo corretto. All'atto pratico sono le "certificazioni" ad assicurare che una transazione sia avvenuta secondo quanto ci si aspetta.

Modello Semplice

Consideriamo un modello di transazione elementare tra due entità Alice e Bob In questo scenario sarà Alice che per prima invierà un messaggio Q("domanda") a Bob che dopo averlo averlo ricevuto risponderà con un un nuovo messaggio A("risposta").



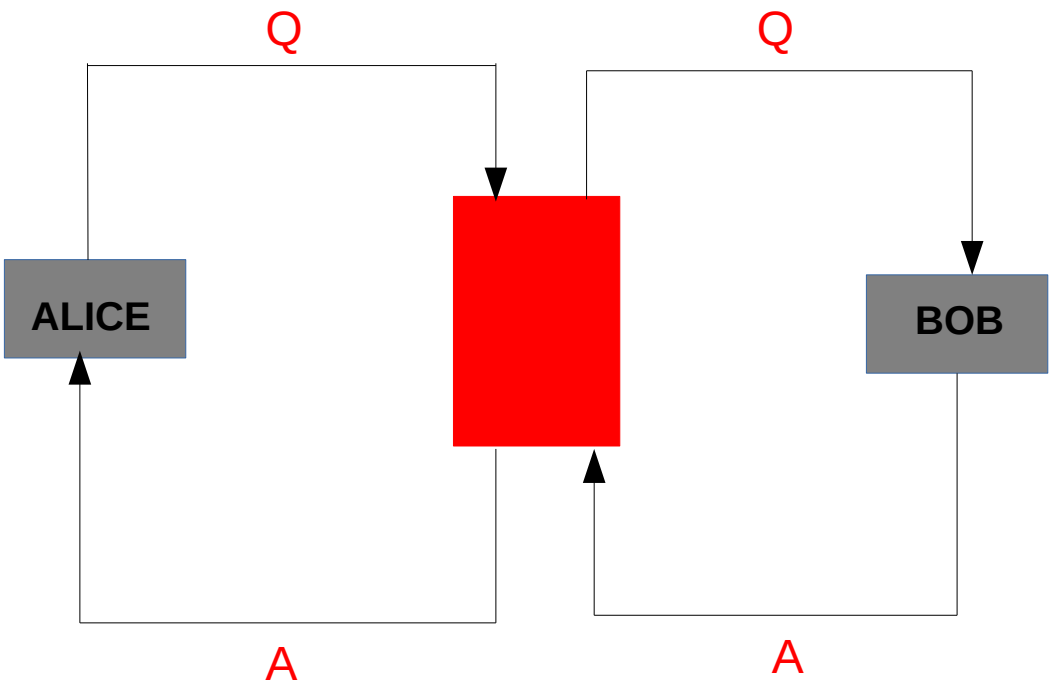
Dunque affinché la transazione abbia buon esito deve avvenire che Bob riceva Q e successivamente Alice riceva A.
Nel caso in cui Alice inviasse Q a Bob senza ricevere la risposta possono essersi verificati due scenari:
1)Bob ha ricevuto Q ed ha risposto con A ad Alice. Avviene che il messaggio A non arriva ad Alice.
2)Bob non ha ancora ricevuto Q. In questo caso Bob non invierà mai A fin quando non riceve Q.

Il problema può essere risolto in vari modi .
Il modo più semplice è di carattere organizzativo e consiste nel doversi affidare ad una terza parte.
Ad esempio se la transazione consiste nell'invio di una mail tra due entità una terza parte che ne garantisce il buon esito potrebbe essere un Server Mail.Oppure più semplicemente potremmo considerare nel caso della comunicazione telefonica il Gestore come una terza parte a cui le due entità si affidano.

Una soluzione più efficacie e che non coinvolge una terza parte potrebbe essere quella di implementare un protocollo che faccia sì che la comunicazione avvenga in un determinato modo.
Cioè il protocollo impone delle regole sulla comunicazione che servono a garantirne la correttezza e a far in modo che le due entità possano interfacciarsi con efficacia.
Un esempio di protocollo è il TCP nel quale le due entità prima di iniziare la comunicazione devono scambiarsi una serie di messaggi volti a stabilire la connessione(handshake a tre fasi).
Dopodiché durante la comunicazione TCP le due entità dovranno scambiarsi messaggi attraverso pacchetti al cui interno ci sono dei dati di riscontro tramite i quali controllare che tutto stia avvenendo nel modo corretto(controllo di flusso).

Modello con intermediario

Abbiamo visto che un modo pratico e organizzativo di ottenere il buon esito di una transazione consiste nel coinvolgere un terzo attore che funge da intermediario e al quale le entità si affidano.

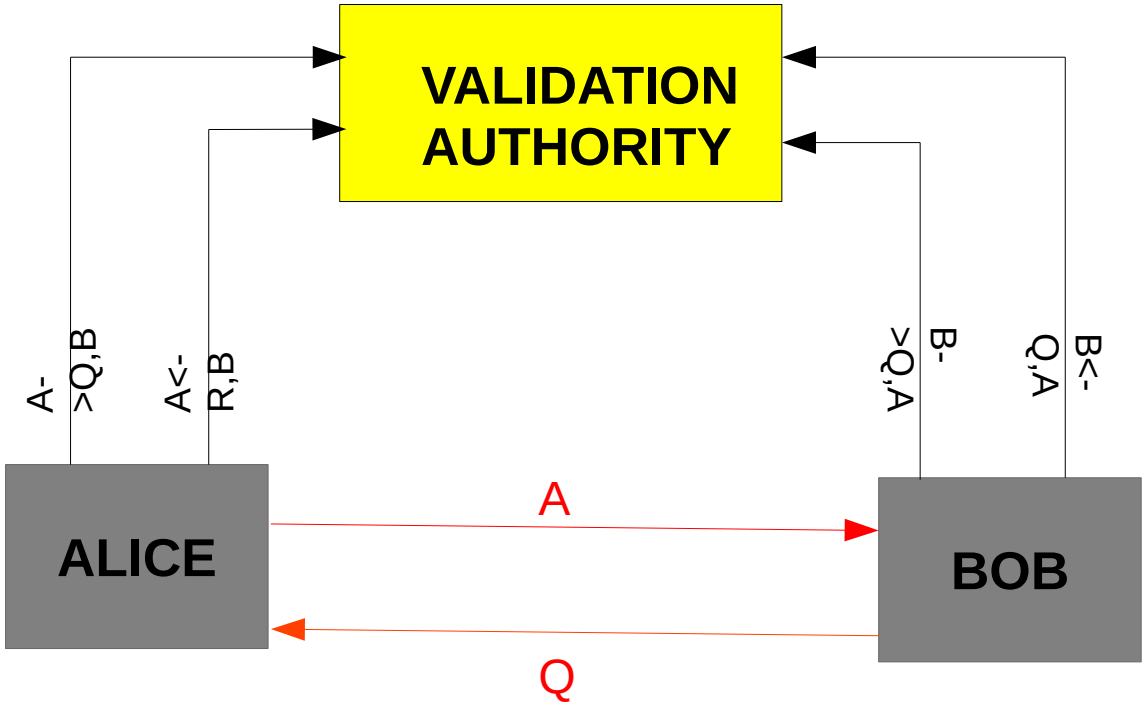


Analizzando il modello si osservano più svantaggi che vantaggi.
Gli svantaggi più significativi sono:
1)Avere introdotto un unico punto di rottura. Un’entità esterna potrebbe compromettere la comunicazione tra Alice e Bob semplicemente attaccando l’intermediario.
Osserviamo infatti che secondo questo modello la comunicazione non potrebbe avvenire in assenza dell’intermediario.
2)L’intermediario deve sostenere una banda di trasmissione dati superiore ad Alice e Bob. Cioè deve supportare sia l’upload che il download di Alice e Bob. Di conseguenza vi saranno maggiori costi realizzativi dovuti ad una dimensione di

banda superiore.Consideriamo il caso in cui Alice sia un utente medio e Bob sia un motore di ricerca (Google).Allora in questo modello Bob dovrebbe replicare l'attività di trasmissione/ricezione di Google.
3)Questo modello non presenta cooperazione tra parti(abbiamo già descritto i vantaggi di un sistema distribuito cooperativo rispetto ad uno centralizzato). Alice e Bob devono affidarsi completamente all'intermediario. Ad esempio nella comunicazione telefonica gli estremi si affidano totalmente al gestore(discorso analogo per i sistemi bancari tradizionali).
Per quel che riguarda i vantaggi di questo modello non si osserva nulla di significativo se non l'aumento di problematiche e l'introduzione di un punto di rottura(vulnerabilità) che potrebbe compromettere la comunicazione.

Modello con Validation Authority

Un modello che rimedia alle problematiche portate dal precedente potrebbe consistere nella Validation Authority:
Alice e Bob scambiano messaggi in maniera diretta senza un intermediario. Non appena uno dei due invia o riceve un messaggio lo comunica alla Validation Authority che tiene traccia dei vari stati della transazione.



La Validation Authority è effettivamente una terza entità da cui la comunicazione tra Alice e Bob non dipende in maniera diretta.
I messaggi con cui Alice e Bob comunicano con la Validation Authority sono i seguenti:
-INVIATI DA ALICE
1) A→Q,B:"sto inviando Q a Bob"
2) A←R,B:"ho ricevuto R da Bob"
-INVIATI DA BOB
1) B→R,A:"sto inviando R ad Alice"
2) B←Q,A:"ho ricevuto Q da Alice"

Quando la Validation Authority riceve uno di questi 4 messaggi da una delle 2 entità riesce a comprendere quale sia lo stato corrente della transazione.Pertanto in caso di problemi di comunicazione questo sistema può risalire alle cause(cosa è accaduto se per esempio Alice non ha ricevuto R da Bob).
Ad esempio se la Validation Authority ha ricevuto solo A→Q,B e Alice reclama di non aver ricevuto R da Bob ci si rende subito conto che il problema è dovuto al fatto che Bob non ha ancora ricevuto Q(altrimenti la Validation Authority avrebbe ricevuto anche B←Q,A da parte di Bob).

Quali svantaggi sono stati eliminati rispetto al modello con intermediario visto precendentemente?
Sicuramente poiché la Validation Authority riceve solo messaggi di stato della transazione da parte delle due entità non ha bisogno di sovradimensionare la banda di trasmissione.
Si può notare inoltre che alla Validation Authority non interessa il contenuto del messaggio ,ma solo che un messaggio x inviato da qualcuno è stato ricevuto da un altro. È cruciale invece che il messaggio sia inviato interamente lungo il canale di comunicazione tra Alice e Bob.
Se Alice e Bob vogliono comunicare segretamente in questo modello, possono farlo mettendosi d'accordo su una chiave pubblica. In questo modo invieranno alla Validation Authority solo l'hash del messaggio(criptato con la chiave pubblica).I messaggi non criptati invece potranno essere inviati lungo il canale di comunicazione che li collega in maniera diretta.

Quindi i messaggi di stato diventerebbero ad esempio:
1) A→hash(Q),B:"sto inviando l'hash di Q a Bob"
2) A←hash(R),B:"ho ricevuto l'hash di R da Bob"
Osserviamo come questo sistema si adatta bene anche nel caso in cui gli estremi della transazione volessero comunicare in segreto(cioè nascondendo a terzi il contenuto dei messaggi).