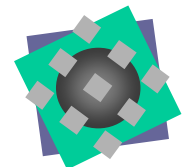


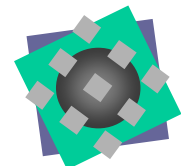
# **WareLite BOSS for Real Time Transaction Activity Monitoring**



# Transaction Activity Monitoring – The Business Need

- **Transaction Activity Monitoring is critical for Financial Services Providers**
  - *The failure of processing even a single transactions within given time parameters can result in high financial losses*
  - *Under Basel II, increasing “the risk of loss resulting from inadequate or failed internal processes, people or systems, or from external events” means increasing the amount of capital that a bank must place in reserve as a buffer against such risk*

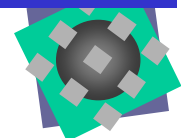
**The failure to process transactions within pre-determined time windows causes direct financial losses and increases risk**



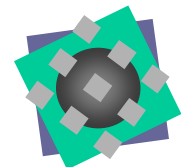
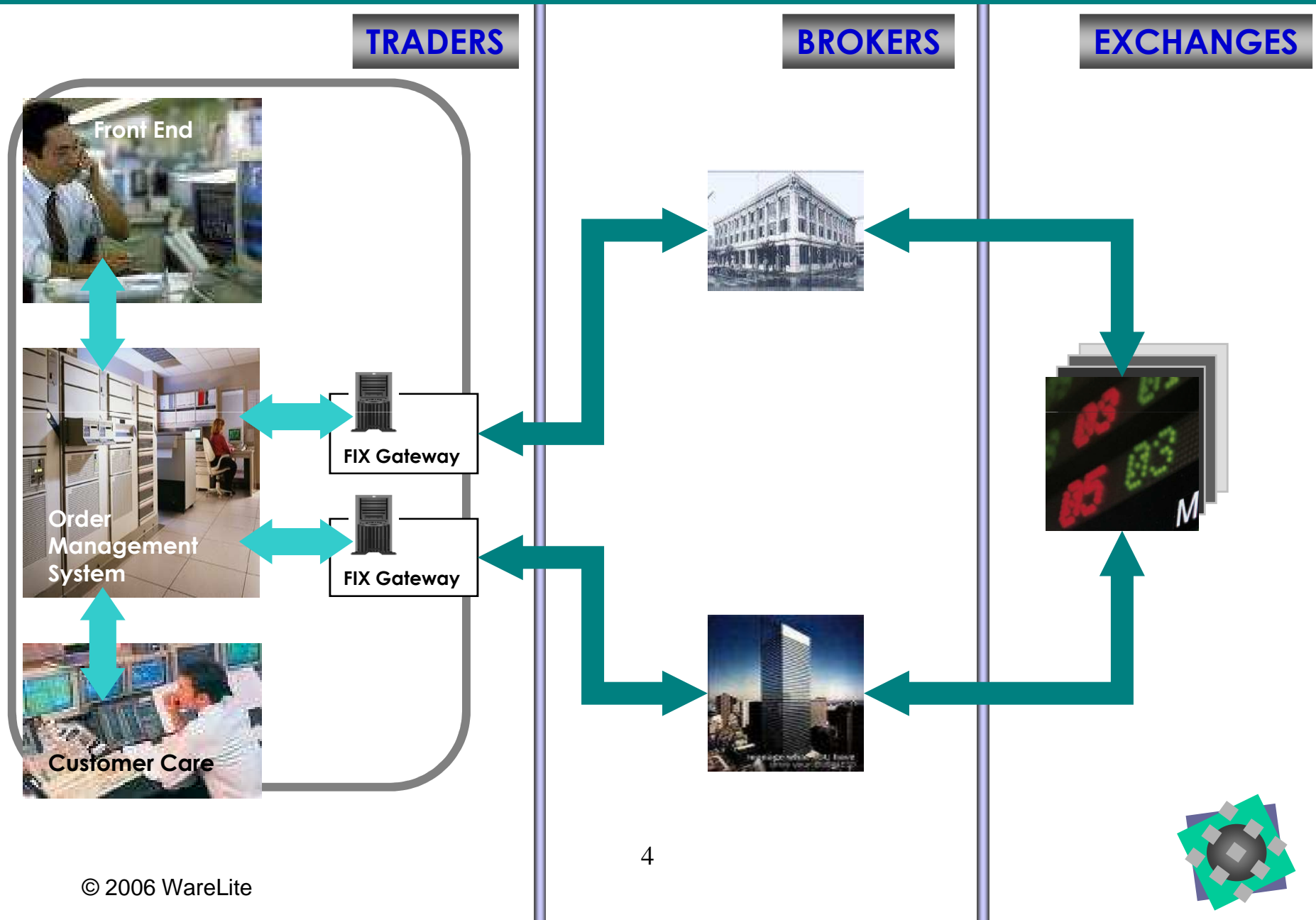
## Transaction Monitoring – Current Situation

- **The current approach to direct monitoring of financial transactions is mostly manual: operators check transactions integrity and fulfillment times through specialised GUIs (e.g. Cameron Fix Monitoring) and take corrective actions when they identify anomalies**
  - *This approach has all the limitations connected to human activities, i.e. mistakes, slow reactivity, inconsistency*
- **Automated monitoring is limited to simulating a ‘enduser transaction to measure and set baseline metrics for the end-user experience’ (Ptak, Noel & Associates, “BMC: Tracking Transactions for Performance”, February 2006)**

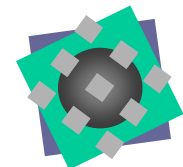
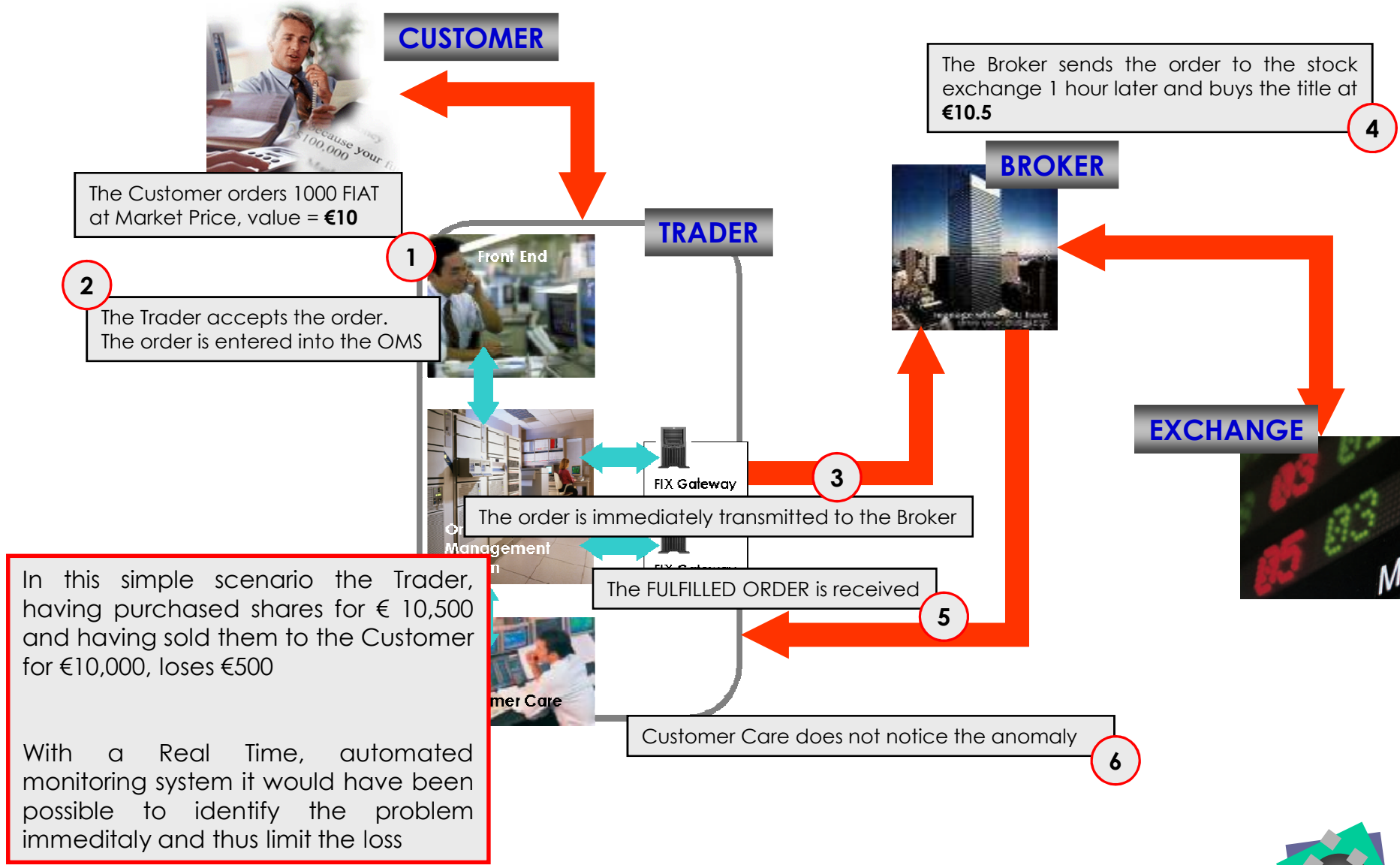
**By automating Activity Monitoring for each single, actual transaction and implementing Real Time responsiveness to any anomaly, it would be possible to dramatically decrease financial losses and risk**



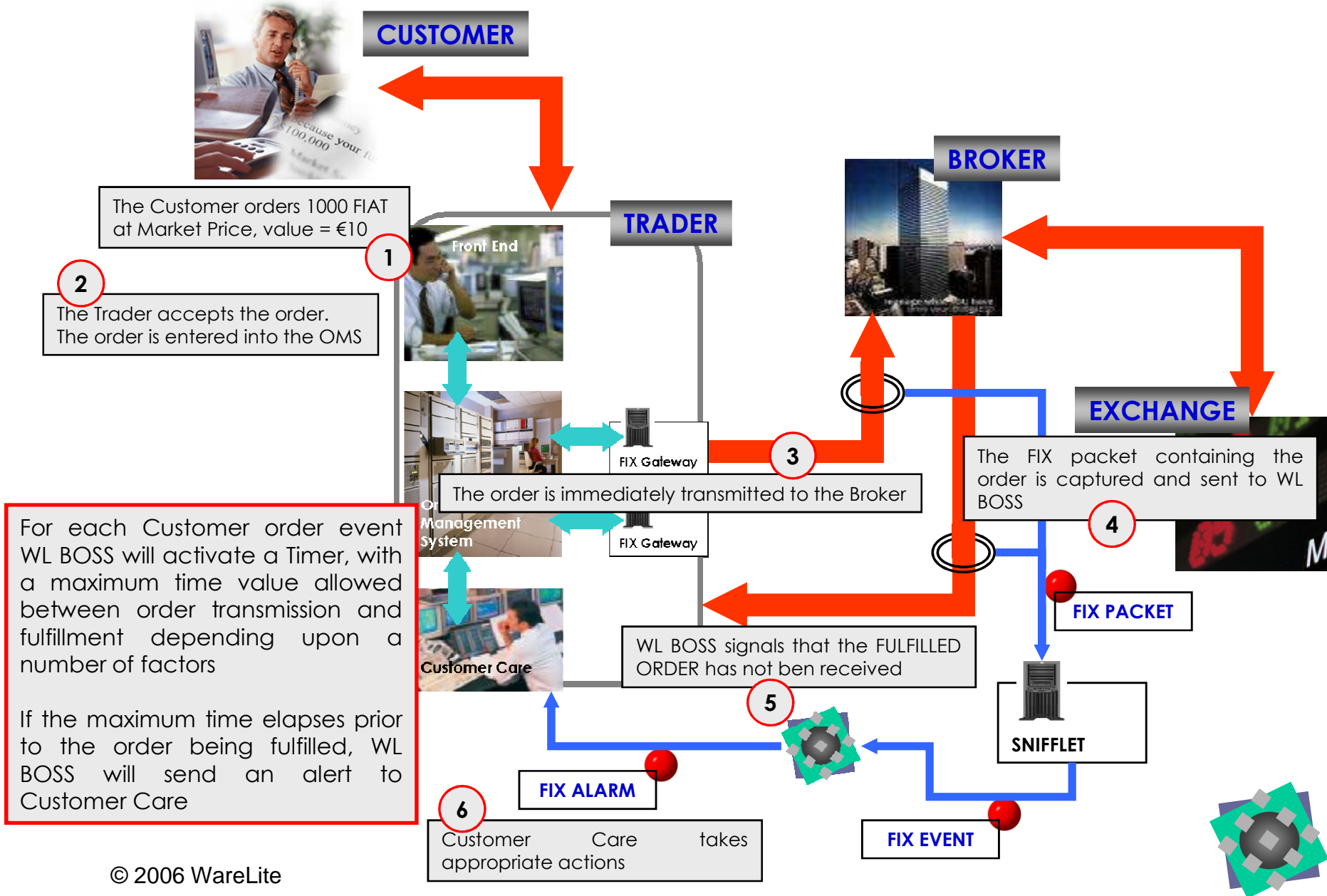
# Financial Transactions Execution – the main Stakeholders



# Financial Transactions Execution – an AS IS Scenario

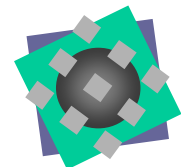
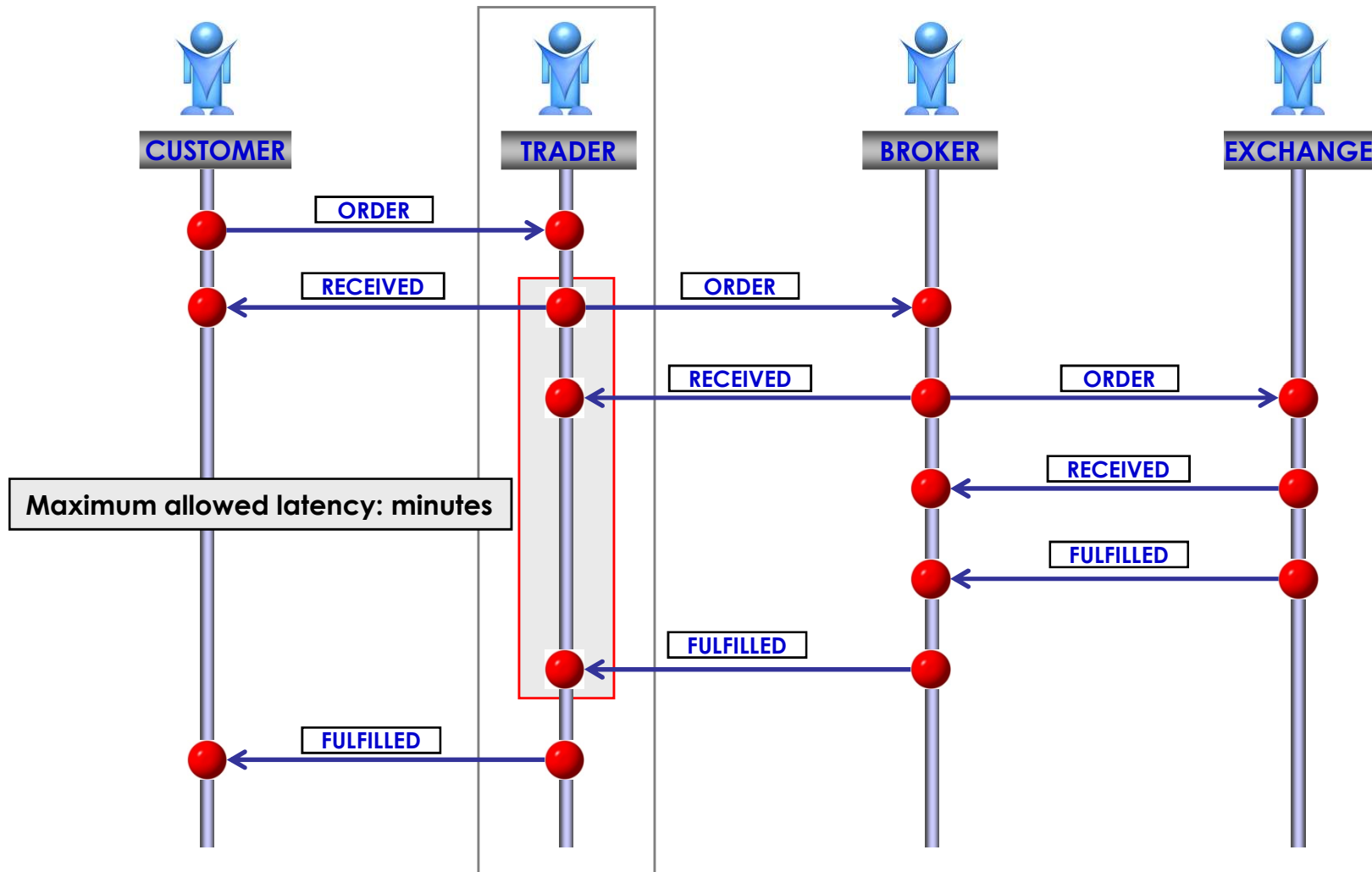


# Financial Transactions Execution – a TO BE Scenario



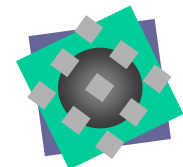
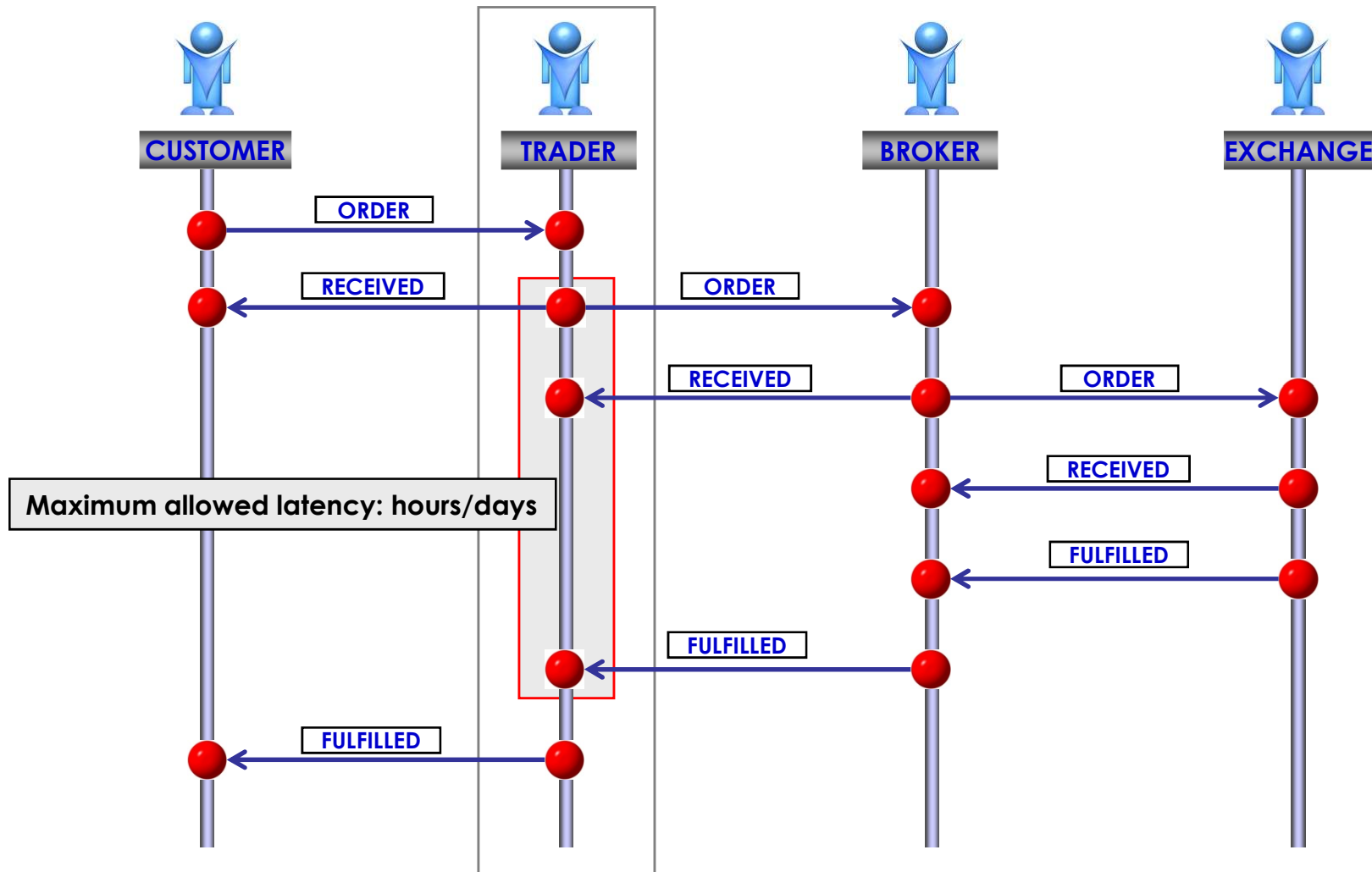
# Financial Transactions Execution – the FIX Protocol

Order at market price



# Financial Transactions Execution – the FIX Protocol

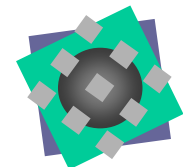
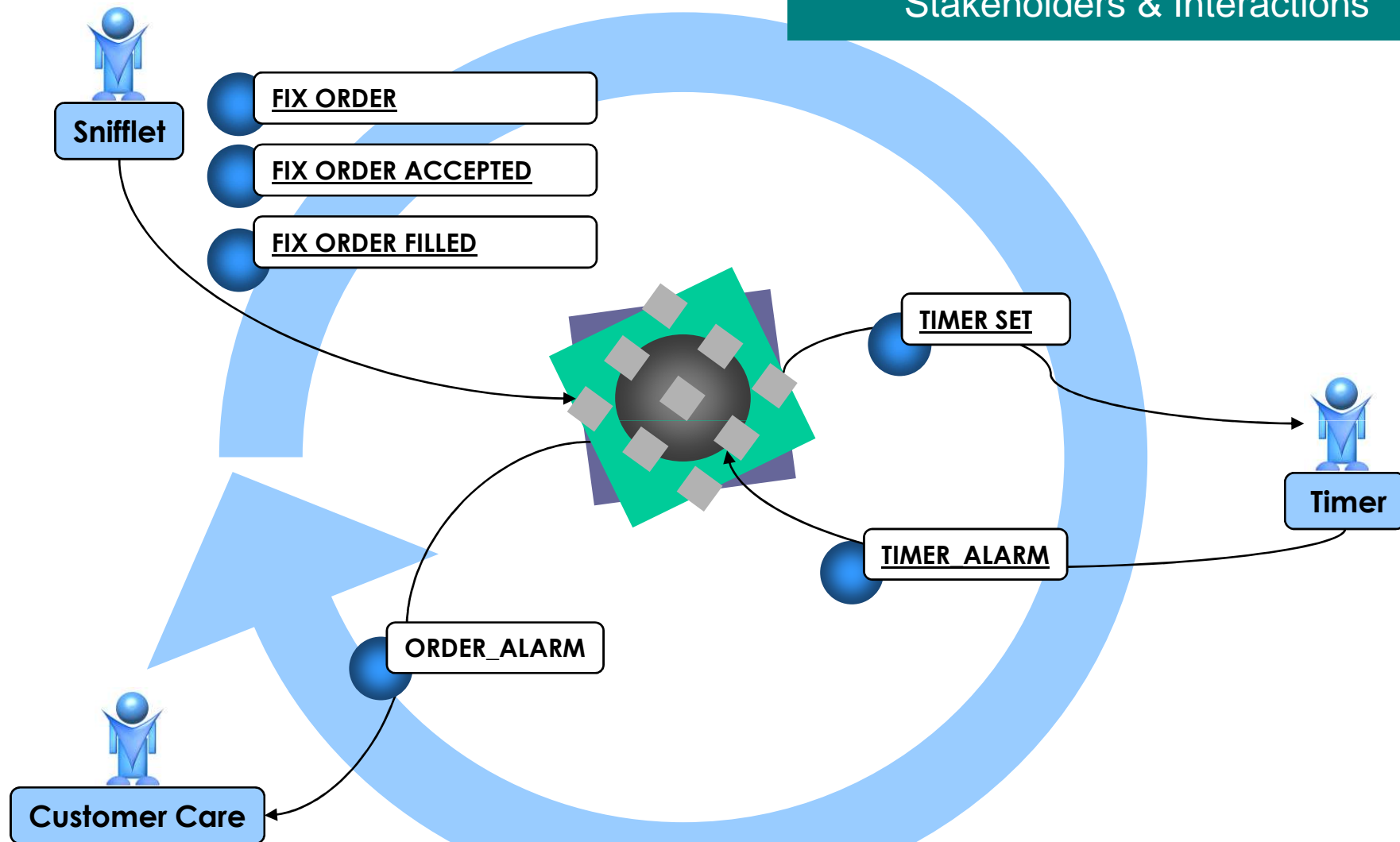
Order below market price





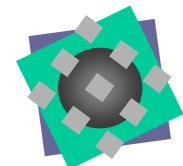
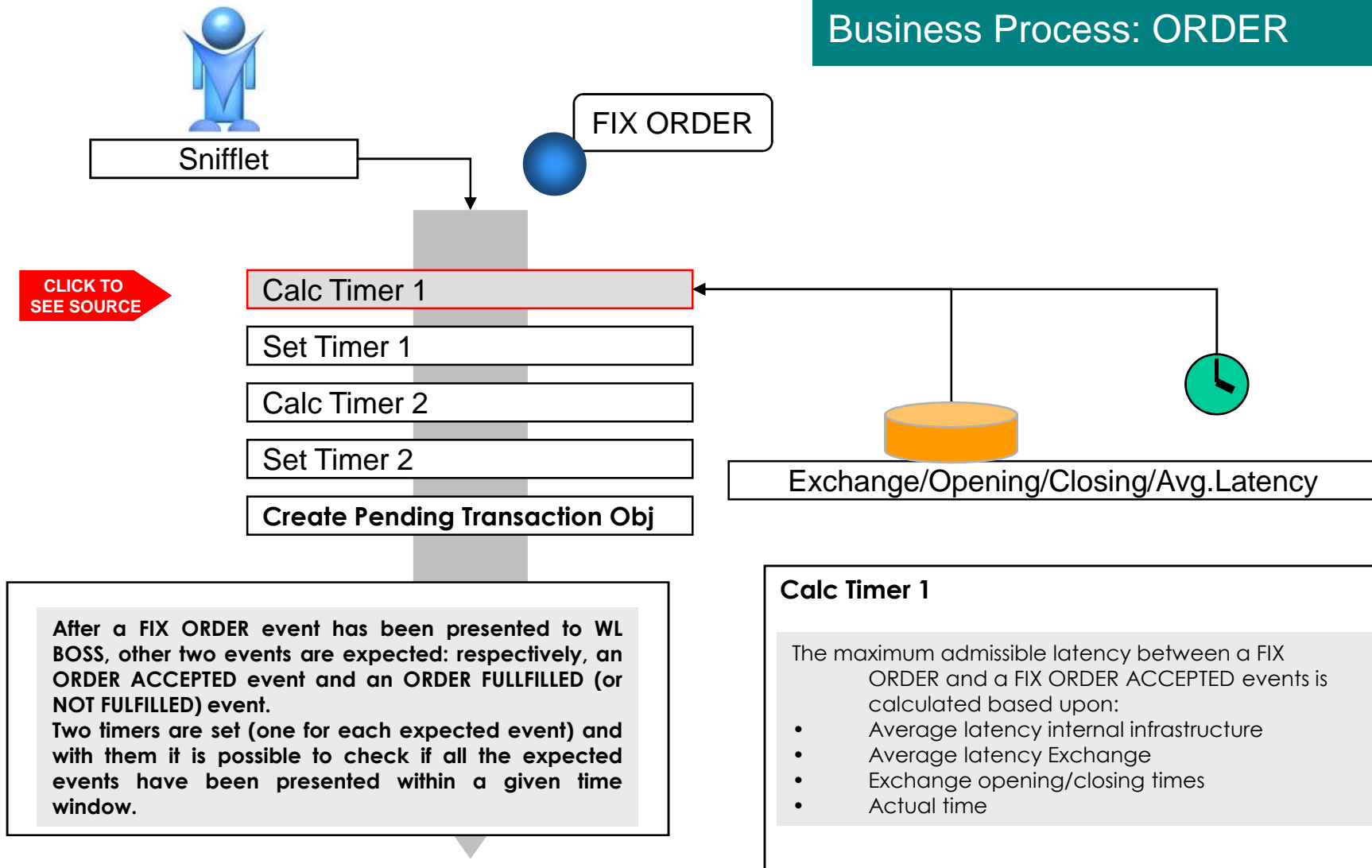
# Financial Transactions Execution – a TO BE Scenario

## Stakeholders & Interactions



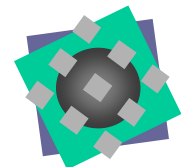
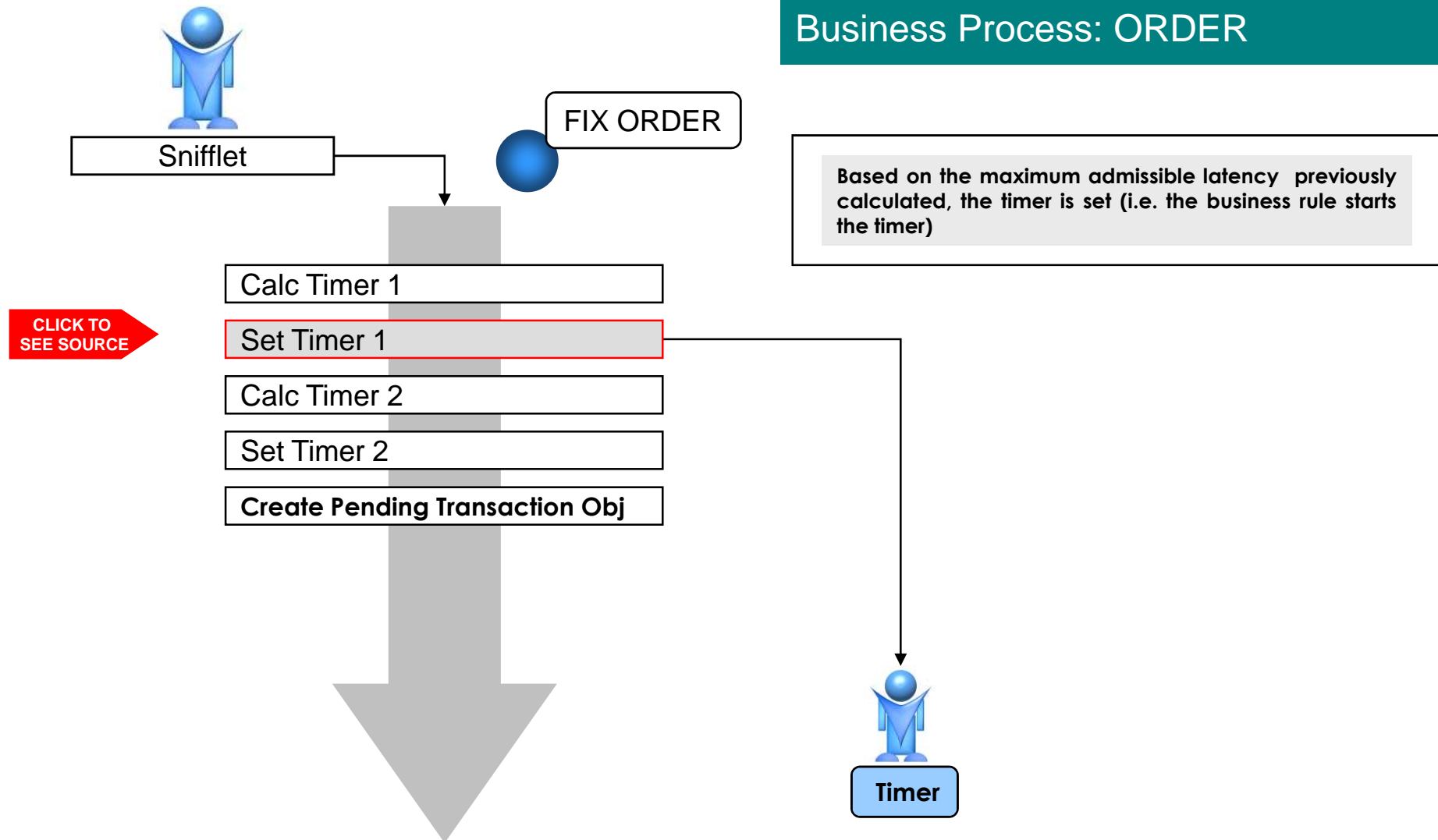
# Stakeholder: Snifflet

Business Process: ORDER



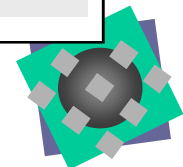
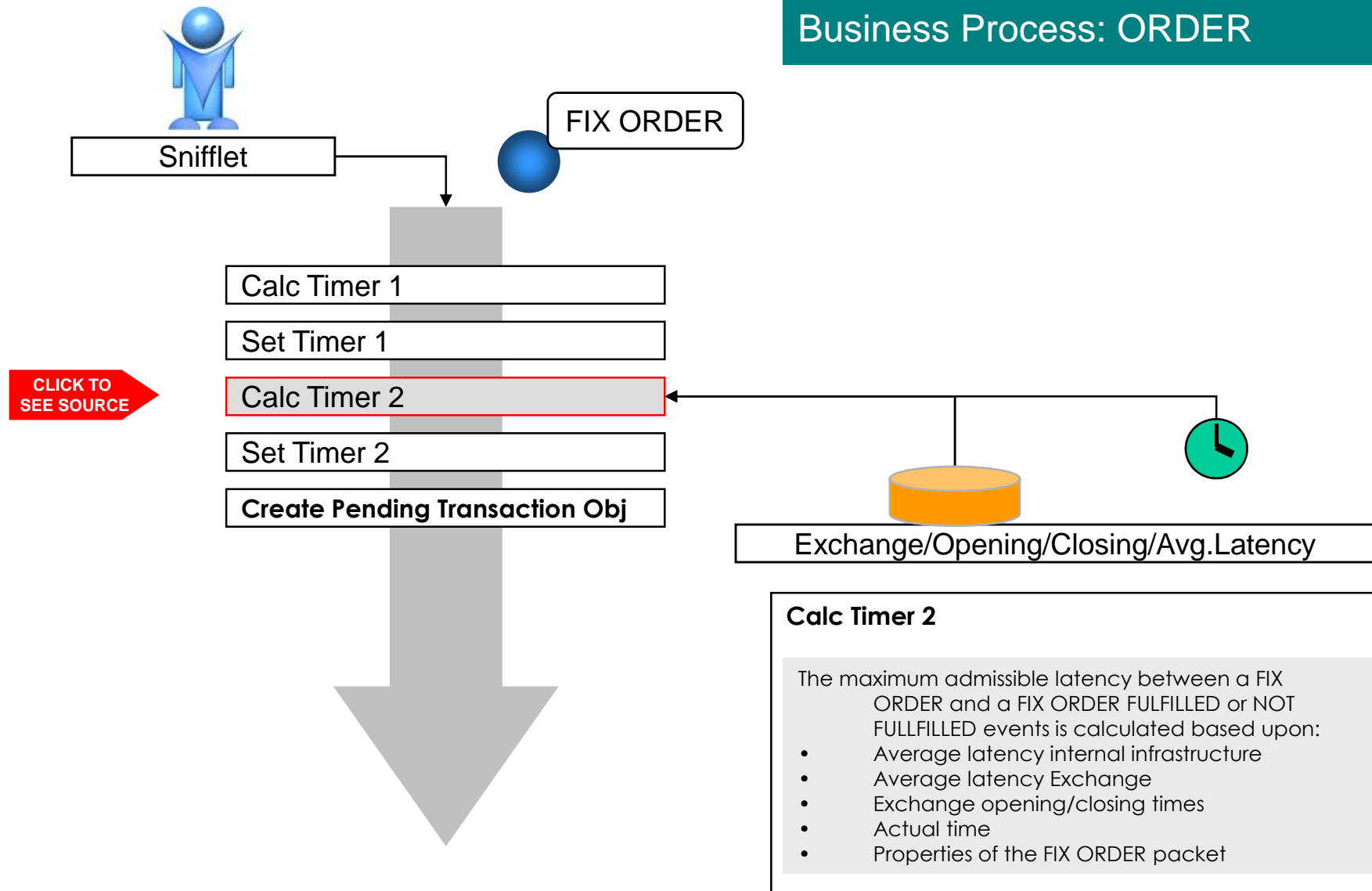
# Stakeholder: Snifflet

## Business Process: ORDER



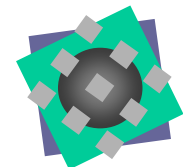
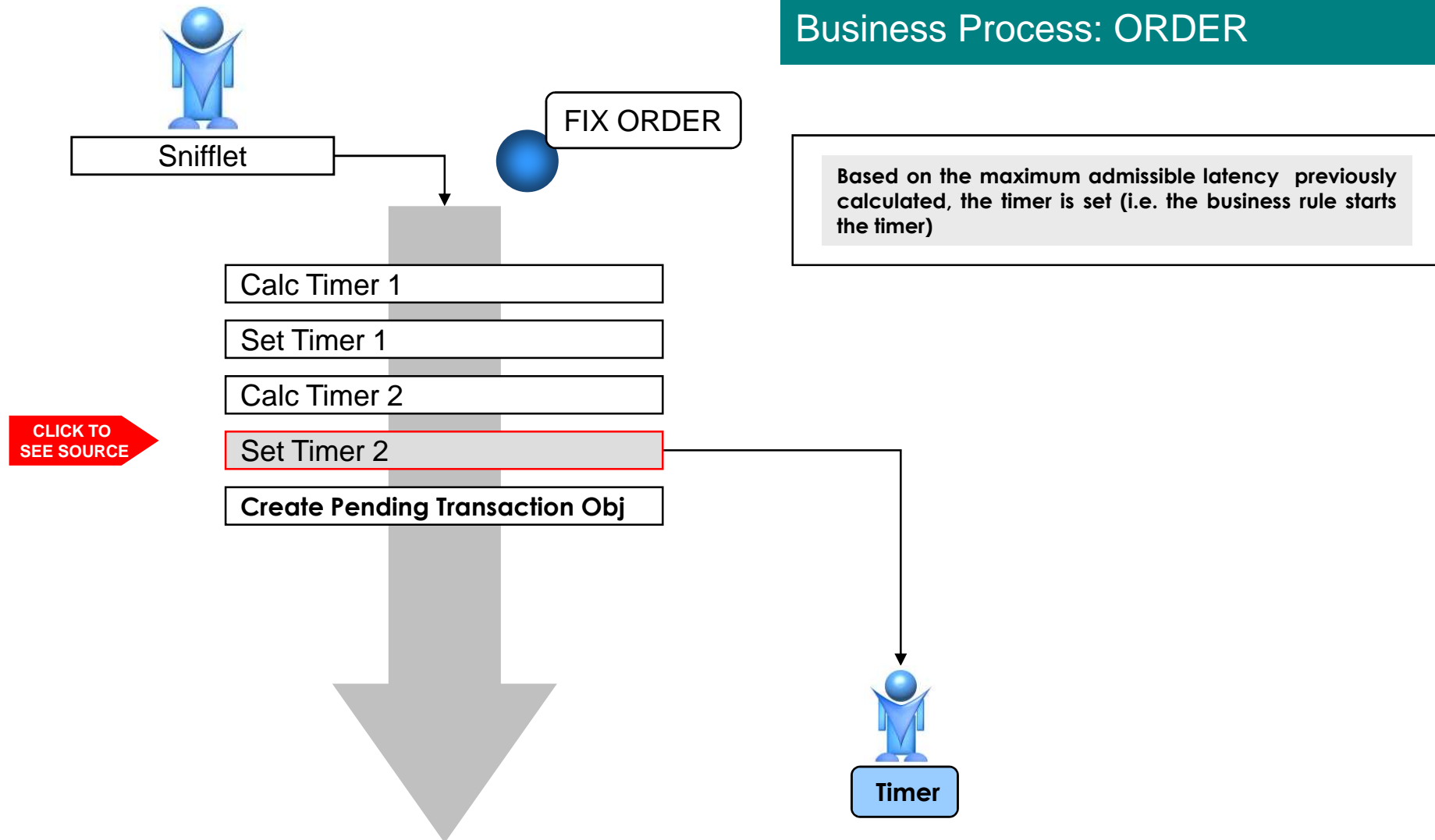
# Stakeholder: Snifflet

Business Process: ORDER



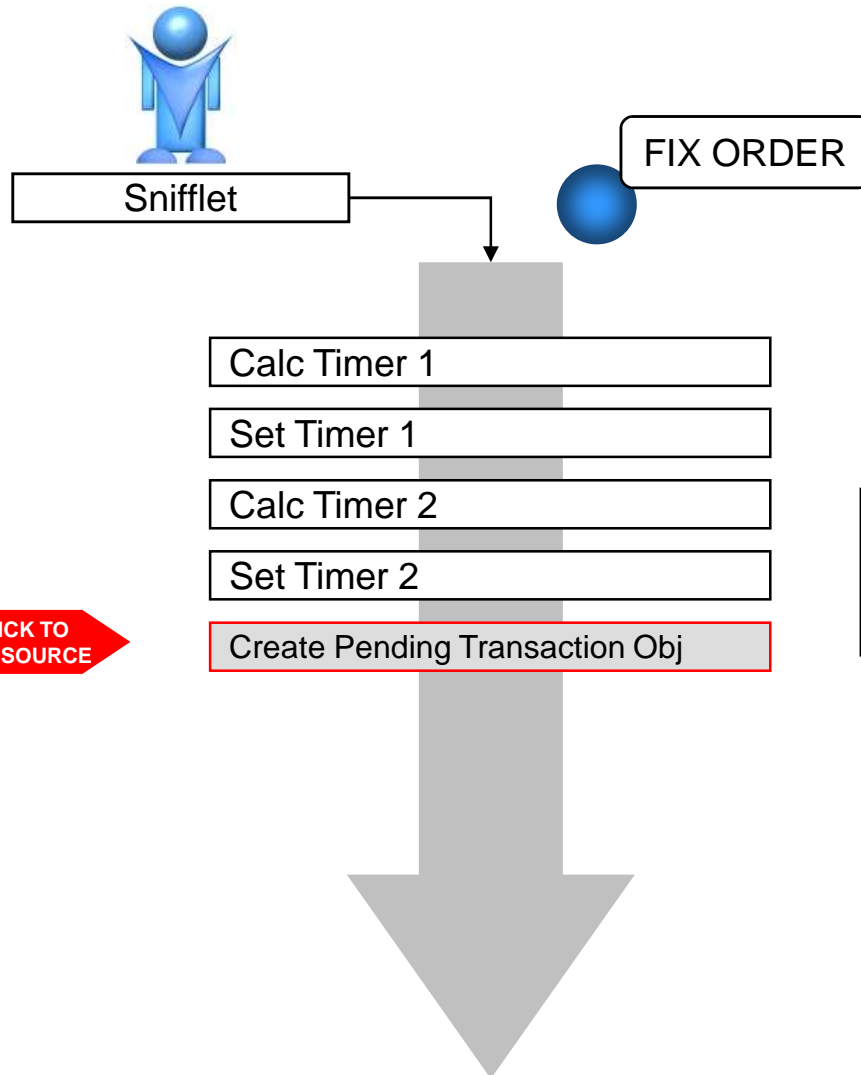
# Stakeholder: Snifflet

## Business Process: ORDER



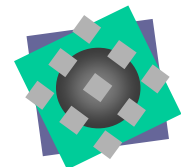
# Stakeholder: Snifflet

Business Process: ORDER



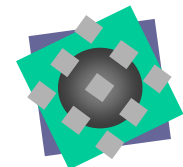
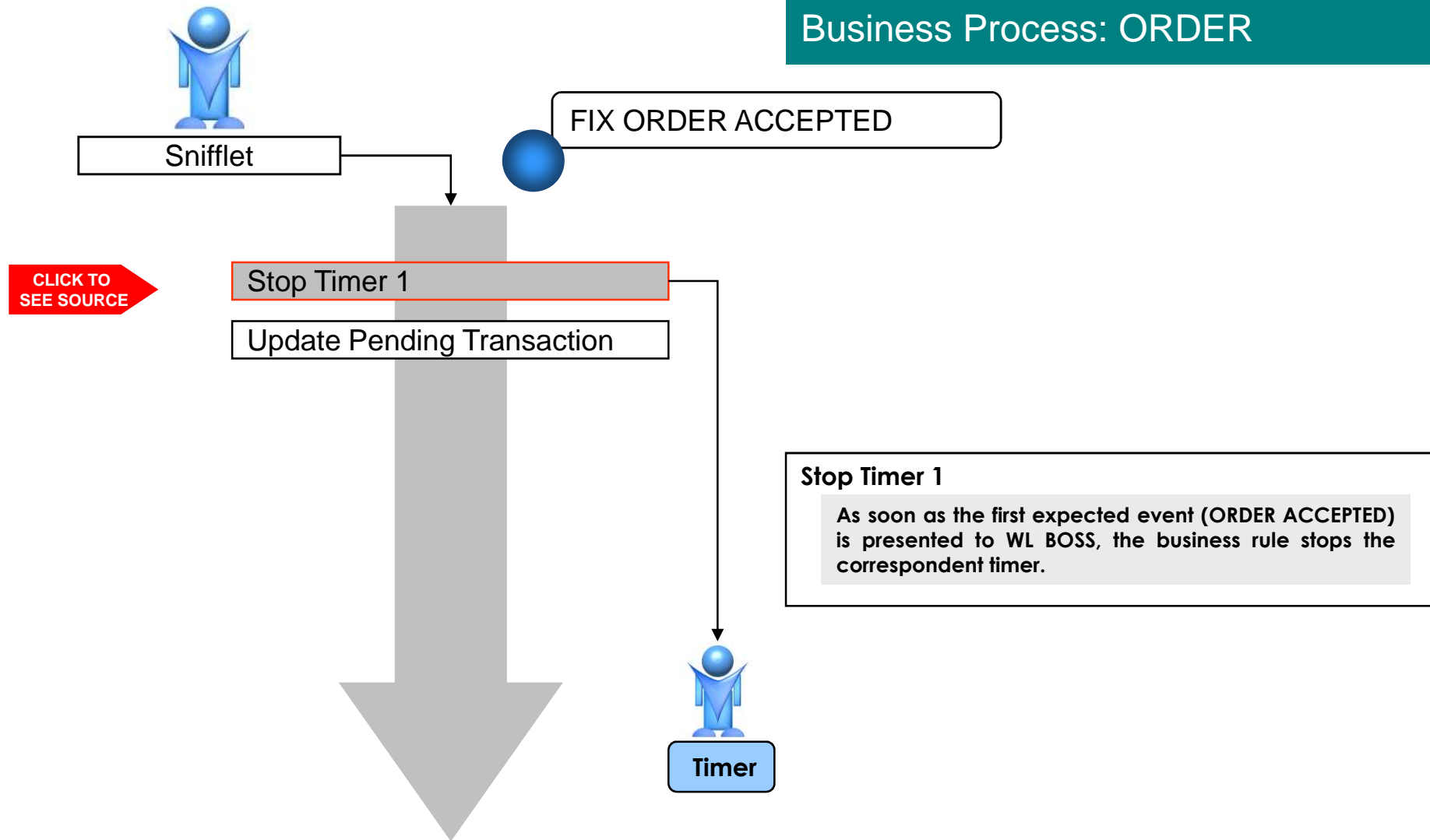
[CLICK TO SEE SOURCE](#)

An object with persistence is created. This object (pending transaction) will be used to represent the various states/phases of a FIX TRANSACTION.



# Stakeholder: Snifflet

Business Process: ORDER



# Stakeholder: Snifflet

Business Process: ORDER\_ACCEPTED



Snifflet

FIX ORDER ACCEPTED

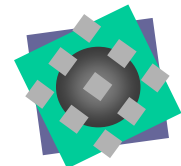
Stop Timer 1

Update Pending Transaction

CLICK TO  
SEE SOURCE

## Update Pending Transaction

As soon as the event ORDER ACCEPTED is presented to WL BOSS, the business rule updates the status of the object representing the pending FIX transaction.





# Stakeholder: Snifflet

Business Process: ORDER\_FULFILLED



Snifflet

FIX ORDER FULFILLED

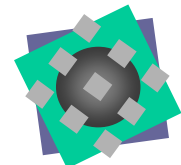
CLICK TO  
SEE SOURCE

Update Pending Transaction

Stop Timer 2

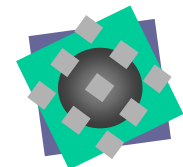
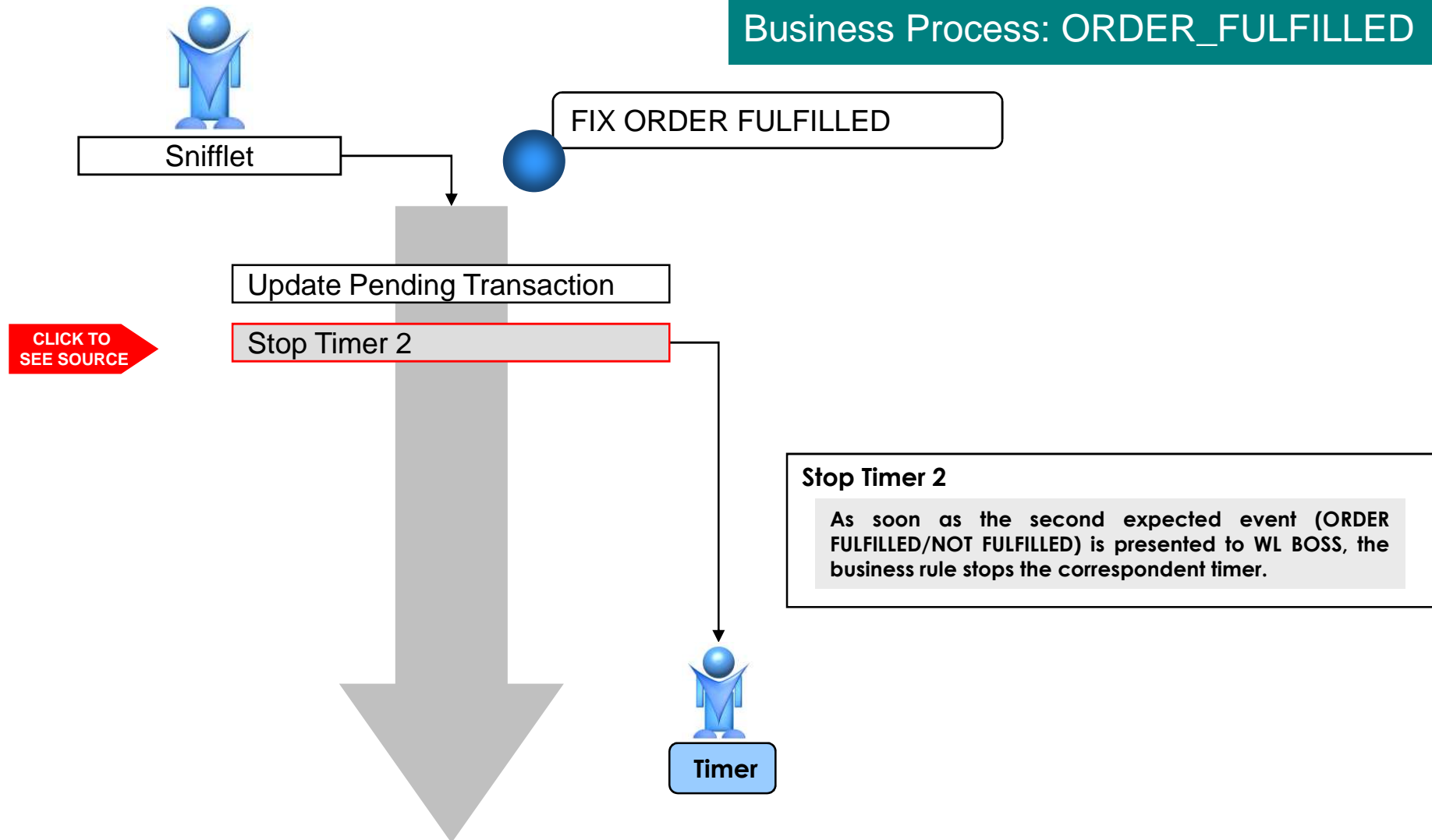
## Update Pending Transaction

As soon as the event ORDER FULFILLED is presented to WL BOSS, the business rule updates the status of the object representing the pending FIX Transaction. The business rule checks whether the sum of the quantity contained in the incoming event and the partial total contained in the pending transaction is less than or equal to the quantity expressed in the FIX ORDER event. If the sum is lower than the order quantity, then the pending transaction partial quantity is updated (and the timer is not stopped). If the sum is equal to the order quantity, then the pending transaction is updated (status=FULFILLED) and the timer is stopped.



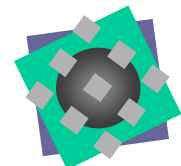
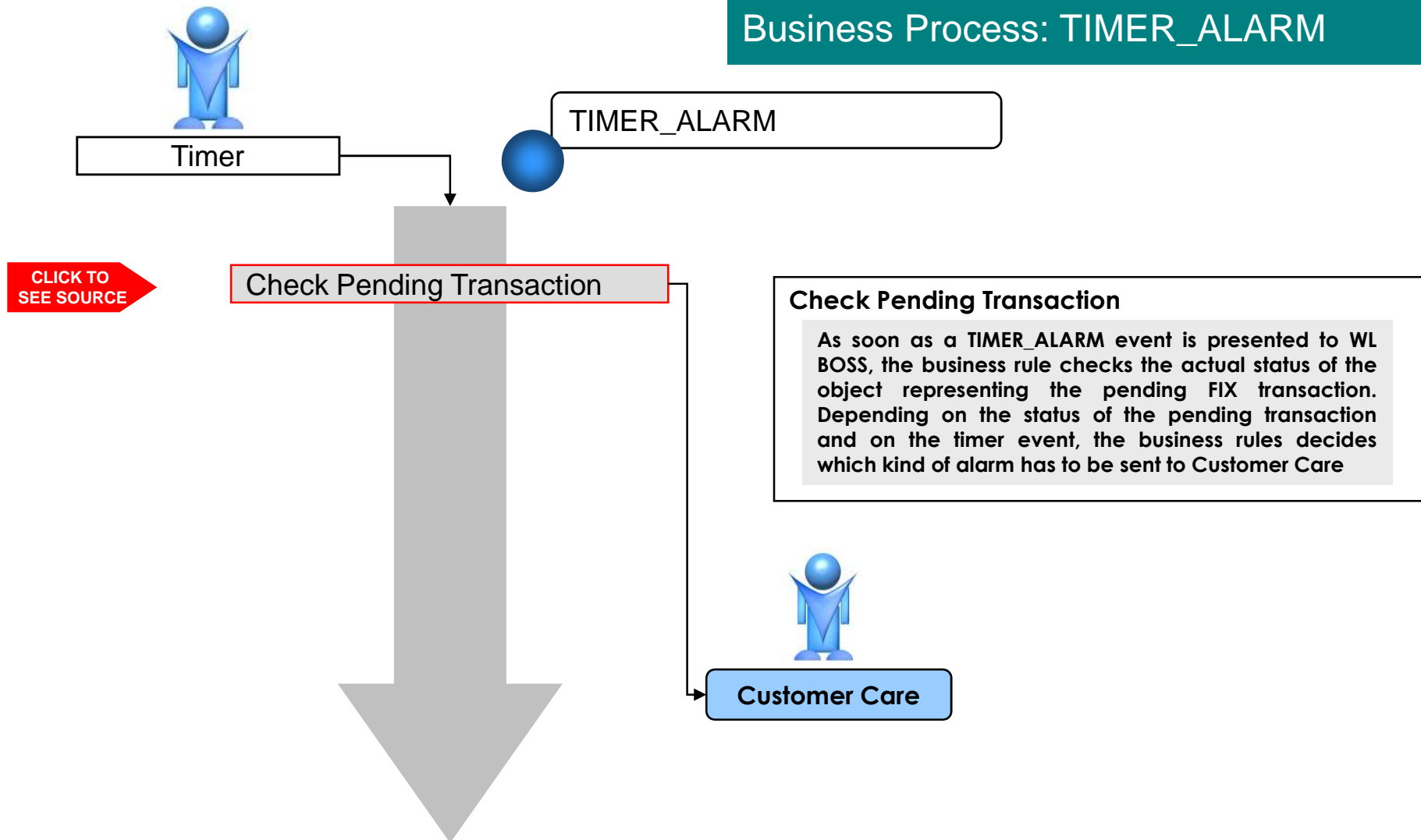
# Stakeholder: Snifflet

Business Process: ORDER\_FULFILLED



# Stakeholder: Timer

Business Process: TIMER\_ALARM



# WL BOSS for Real Time Transaction Activity Monitoring: Benefits

- **A Completely Retrofit Solution**
  - *Transactions are detected as they transit the network, with no need to interfere with applications*
- **A Unified View**
  - *The same monitoring and responsiveness logic can be applied across the whole network, whatever the OS supporting applications*
- **Real Time Responsiveness**
  - *Each single transaction is monitored in real time. If a reply to a request is not provided within a pre-defined time window, a pre-defined reaction is executed automatically*
- **Decreased Losses**
  - *As each single transaction is monitored in real time and, in case of malfunction or delayed response, appropriate reactions are implemented automatically, the financial loss associated to delayed or missed order fulfillment is dramatically reduced*
- **Decreased Risk**
  - *As each single transaction is monitored in real time, it is possible to meet Basel II requirements with automated, real time risk calculations*

