TX²

Technical Solution:
High Volume, High Speed, Workflow

Summary:

Purpose: Distribute work to a pool of operators
Technology Used: WebSphere MQ, Message Broker and customised plug-ins

Overview:

In this solution, a fully-blown workflow product was considered too cumbersome for the high-speed work required from the operators. For example, a response time requirement of 3 seconds to process each image set, combined with an average size of over 100k for each image set, meant that no delay could be tolerated between the images, they need to be pre-cached in the browser. A constantly changing pool of operators, with operators coming and going all the time also requires close monitoring by the system of each work item to ensure it is handled by an alternative operator where the original operator is no longer available.

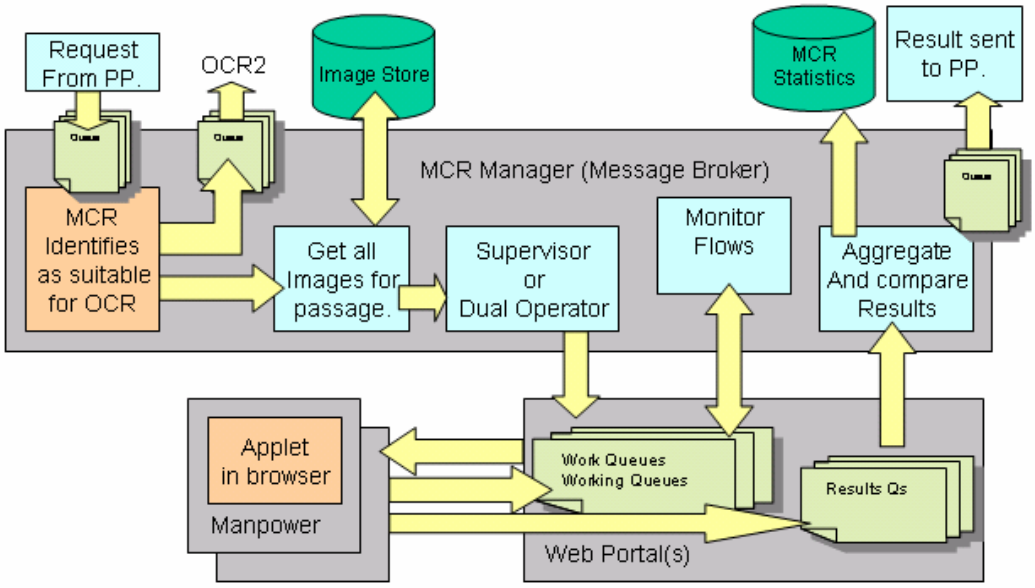
Further requirements including the ability to reduce error rates by sending the same work to two operators and to compare their entries; where they differ, the work item must be sent to a supervisor for arbitration.

All activity within the system needed to be logged to produce reports on individuals' performance in terms of processing time/volume and error rate.

Key Attributes:

- Tested with many hundreds of human operators (MQ cluster technology across Linux servers was used to scale up user numbers)
- Tested with large data volumes (images from traffic cameras in real time)
- Large volumes of work in short timescales (human operators respond on average in 2 or 3 seconds to each piece of work)
- Multiple “roles” supported (Operator, Supervisor, etc.)
- Several pieces of work can be co-ordinated and treated as one transaction (eg. Work sent to more than one operator, with results compared, and if they do not agree, work sent to a supervisor)
- Work can be directed to a particular operator if required (specific work is prioritized and rises to the top of the operators work)
- Fully transactional; even if the operator just leaves or turns off the application, work will be recovered and re-assigned
- Timeouts and re-distribution of work after specified times
- Work for specific operators can be timed out and re-distributed to any operator (within the specified role, or to another role)
- New roles easy to add.

Use of Technology Components in a Specific Implementation



- MQ is used for transporting all requested actions between components (and servers), providing a fully asynchronous model that is tolerant of periods of down-time for each component or server.
- Message Broker is used to provide the routing and transformation of the requests.
- MQ Cluster is used to distribute the operator work across multiple (Linux) servers running Application Server on each.
- Message Broker flows and plug-ins are used to manage the image store (retrieve the images to be placed in the messages on the work queues).
- Message Broker plug-ins are used to monitor the state of the work and working queues to provide timeout and recovery facilities.
- Message Broker aggregation is used to coordinate multiple pieces of work that are part of the same work set, and a final result only sent back when all parts of the work item are complete.
- An MQ application can also simulate a real operator, either for testing purposes, or for partial automation of the process.

