Pros	Cons	
Read is very quick - currently sub 25ms average in release/ma cores		
Will help with keeping all data in sync	Still relies on polled jobs to get data so always going to be out of date with original ingestion	
Can scale horizontally quickly by dropping in a new instance and having it replicate from the master	Horizontal scale is limited as it requires a full replication on each server rather than a shard	we could implement sharding based on shop ids, we don't need to query across multiple shops, and >30 potential shards should be more than enough
	Master index is a single point of failure on the indexing side	
Can browse by field:id to return a unique document very quickly as it only searches that exact field for an exact match	There is no guarantee that your field id is unique as the only unique id specified is currently used as a unique id for the entire document (e.g. ShopProduct rather than product)	Solr can only enfore the unique constraint on one field per document. To avoid index collisions, we create a compound IDs e.g. for masterartists: "ma_<shopId>_<masterArtistId>". A search for artist Id may not be unique, but if filtered by shop it will be as the two were combined at index time.
	Search layer is overkill, "query" engine still has to be used, even if your searching a specific field	
Technology is in a mature state in domain and has been proven in production	We haven't tried anything else!	
Most of these concerns are easily resolved and will be fully resolved in Solr 4.0		