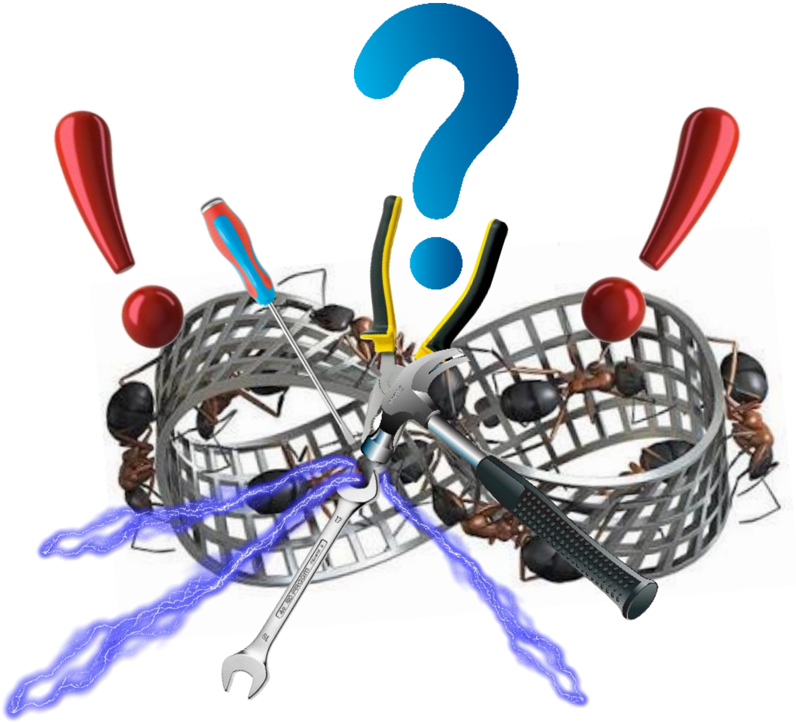
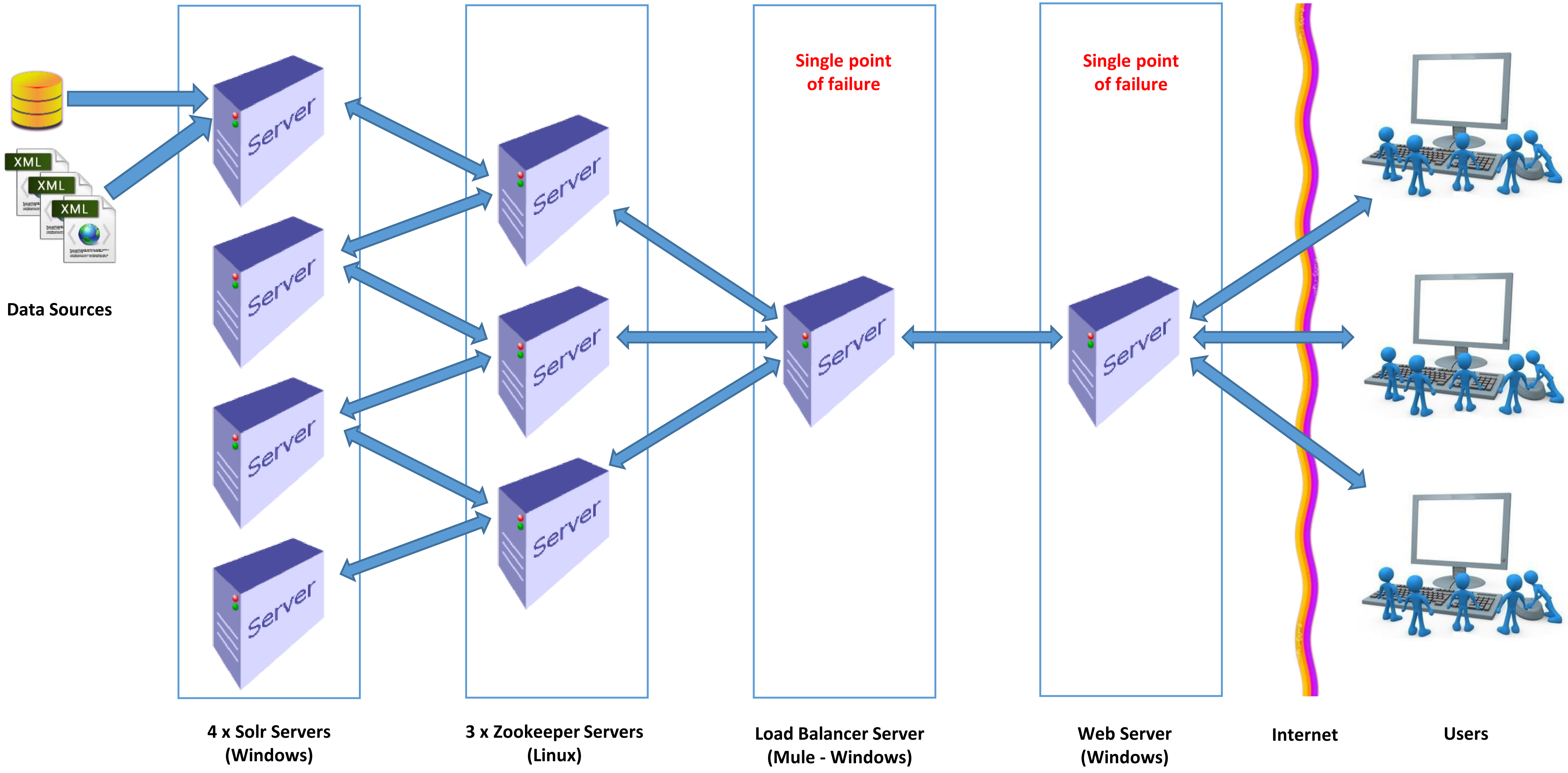
Solr Cloud vs Replication



vs

Solr Cloud implementation



Solr Cloud implementation

- **Solr Cloud working is *supposed* to provide a fault-tolerant, high availability system, providing a stable platform for blazing fast full-text and faceted searches.**

Solr Cloud implementation

- Solr Cloud working is *supposed* to provide a fault-tolerant, high availability system, providing a stable platform for blazing fast full-text and faceted searches.
- **A Solr Cloud implementation would typically consist of (x) number of Solr servers, (x – 1) number of Zookeeper servers and a load balancer.**

Solr Cloud implementation

- Solr Cloud working is *supposed* to provide a fault-tolerant, high availability system, providing a stable platform for blazing fast full-text and faceted searches.
- A Solr Cloud implementation would typically consist of (x) number of Solr servers, (x – 1) number of Zookeeper servers and a load balancer.
- **The Zookeepers deal with the replication of configuration files and indexes across the Solr server ensemble and report the server state to the load balancer.**

Solr Cloud implementation

- Looks good in theory BUT…

Solr Cloud implementation

- Looks good in theory BUT…
  - **We had three environments, Integration (development), Staging (testing) and Live. All three environments were different though…**

Solr Cloud implementation

- Looks good in theory BUT…
  - We had three environments, Integration (development), Staging (testing) and Live. All three environments were different though…
    - **The Live environment had FOUR Solr servers, THREE Zookeepers and a ‘Mule’.**
    - **The Staging environment had TWO Solr servers, ONE Zookeeper and a ‘Mule’.**
    - **The Integration environment had ONE Solr server, NO Zookeeper and a ‘Mule’.**

Solr Cloud implementation

- Looks good in theory BUT…
    - We had three environments, Integration (development), Staging (testing) and Live. All three environments were different though…
        - The Live environment had FOUR Solr servers, THREE Zookeepers and a ‘Mule’.
        - The Staging environment had TWO Solr servers, ONE Zookeeper and a ‘Mule’.
        - The Integration environment had ONE Solr server, NO Zookeeper and a ‘Mule’.
    - **The Live environment suffered from a maintenance issue (the ‘C-drive’ on the Master Solr server ran out of space). This brought the entire system down! We had to switch Live to the simplified Staging environment to provide service to our users.**

Solr Cloud implementation

- Looks good in theory BUT…
  - We had three environments, Integration (development), Staging (testing) and Live. All three environments were different though…
    - The Live environment had FOUR Solr servers, THREE Zookeepers and a ‘Mule’.
    - The Staging environment had TWO Solr servers, ONE Zookeeper and a ‘Mule’.
    - The Integration environment had ONE Solr server, NO Zookeeper and a ‘Mule’.
  - The Live environment suffered from a maintenance issue (the ‘C-drive’ on the Master Solr server ran out of space). This brought the entire system down! We had to switch Live to the simplified Staging environment to provide service to our users.
  - **The Zookeeper notion was unreliable, with indexes getting out of synchronisation daily, hence the maintenance overhead was huge in order to simply try and provide a stable and reliable service for our users.**

Solr Cloud implementation

- Time for a rethink…

Solr Cloud implementation

- Time for a rethink…
  - What do we need?
    - **Reliability**
      - **Search results should be consistent.**

Solr Cloud implementation

- Time for a rethink…
  - What do we need?
    - Reliability
      - Search results should be consistent.
    - **Low Maintenance**
      - **Apart from routine checks a Solr environment should be fairly maintenance free, with errors encountered being reported by email.**

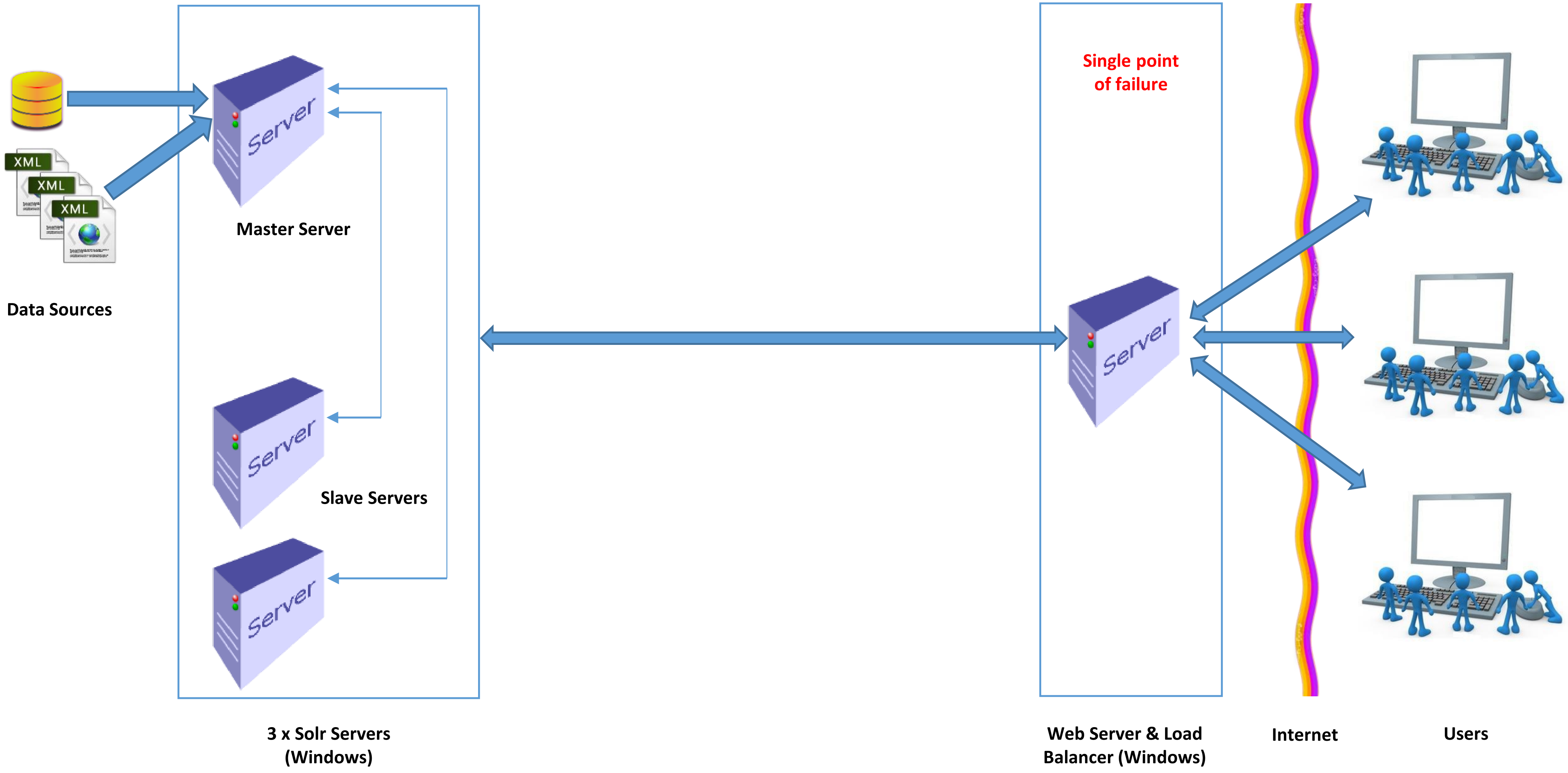
Solr Cloud implementation

- Time for a rethink…
  - What do we need?
    - Reliability
      - Search results should be consistent.
    - Low Maintenance
      - Apart from routine checks a Solr environment should be fairly maintenance free, with errors encountered being reported by email.
    - **Scalability**
      - **The ability to grow or shrink the number of servers in an environment should be made as simple as possible.**
      - **One should be able to easily change which is the ‘master’ server and which are to be the ‘slaves’.**

Solr Cloud implementation

- Time for a rethink…
  - What do we need?
    - Reliability
      - Search results should be consistent.
    - Low Maintenance
      - Apart from routine checks a Solr environment should be fairly maintenance free, with errors encountered being reported by email.
    - Scalability
      - The ability to grow or shrink the number of servers in an environment should be made as simple as possible.
      - One should be able to easily change which is the ‘master’ server and which are to be the ‘slaves’.
    - **Easily Configurable**
      - **A simple system for uploading the Solr configuration files across the different environments, with the ability of allowing different behaviours for each Core within them.**

Solr Replication implementation



Solr Replication implementation

- Advantages…
  - **All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.**

Solr Replication implementation

- Advantages…
  - All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.
  - **All the hardware specifications are the same across all servers in all environments.**

Solr Replication implementation

- Advantages…
  - All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.
  - All the hardware specifications are the same across all servers in all environments.
  - **Everything now runs on the Windows OS (no more Linux and no need for bespoke Java coding).**

Solr Replication implementation

- Advantages…
  - All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.
  - All the hardware specifications are the same across all servers in all environments.
  - Everything now runs on the Windows OS (no more Linux and no need for bespoke Java coding).
  - **The 'Mule' has been replaced by an in-house produced load balancer.**

Solr Replication implementation

- Advantages…
  - All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.
  - All the hardware specifications are the same across all servers in all environments.
  - Everything now runs on the Windows OS (no more Linux and no need for bespoke Java coding).
  - The ‘Mule’ has been replaced by an in-house produced load balancer.
  - **The single points of failure have been reduced to just one, which is in the actual Web Server and is where the load balancer now resides.**

Solr Replication implementation

- Advantages…
  - All environments (Integration, Staging, Live & QA (a new one)) use the same layout and number of servers.
  - All the hardware specifications are the same across all servers in all environments.
  - Everything now runs on the Windows OS (no more Linux and no need for bespoke Java coding).
  - The ‘Mule’ has been replaced by an in-house produced load balancer.
  - The single points of failure have been reduced to just one, which is in the actual Web Server and is where the load balancer now resides.
  - **Everything is now easily configurable, scalable, almost maintenance-free and above all is ultra-reliable.**

Solr Replication implementation

- Disadvantages…
  - **A slight delay, barely noticeable in a Web UI, may be experienced if a request to a given Solr server fails and the load balancer has to redirect the request to the other available server.**

Solr Replication implementation

- So what’s so different in Solr Replication?
  - **Indexing is performed on a ‘master’ server.**

Solr Replication implementation

- So what’s so different in Solr Replication?
  - Indexing is performed on a ‘master’ server.
  - **As there is no Zookeeper component to replicate the indexes to the other servers, the onus is now on the ‘slave’ servers to poll the ‘master’ periodically and update their indexes from the ‘master’ copies. This is a built-in feature of Solr and the location of the ‘master’ server and the polling time is easily configurable.**

Solr Replication implementation

- So what’s so different in Solr Replication?
  - Indexing is performed on a ‘master’ server.
  - As there is no Zookeeper component to replicate the indexes to the other servers, the onus is now on the ‘slave’ servers to poll the ‘master’ periodically and update their indexes from the ‘master’ copies. This is a built-in feature of Solr and the location of the ‘master’ server and the polling time is easily configurable.
  - **Unlike the ‘Mule’ in Cloud working, the extremely lightweight in-house load balancer does not know the state of the servers in the Solr ensemble. It knows the last server it tried so for the next query it will try another server it knows about. If it is unable to process the query then it will switch to another one, until there are no more servers to try, whereupon it will send an error email to concerned parties (ADM).**

Solr Replication implementation

- Scalablity?
    - **We normally perform Solr searches on the two ‘slave’ servers. This is configured in the load balancer and each Solr server in the environment.**

Solr Replication implementation

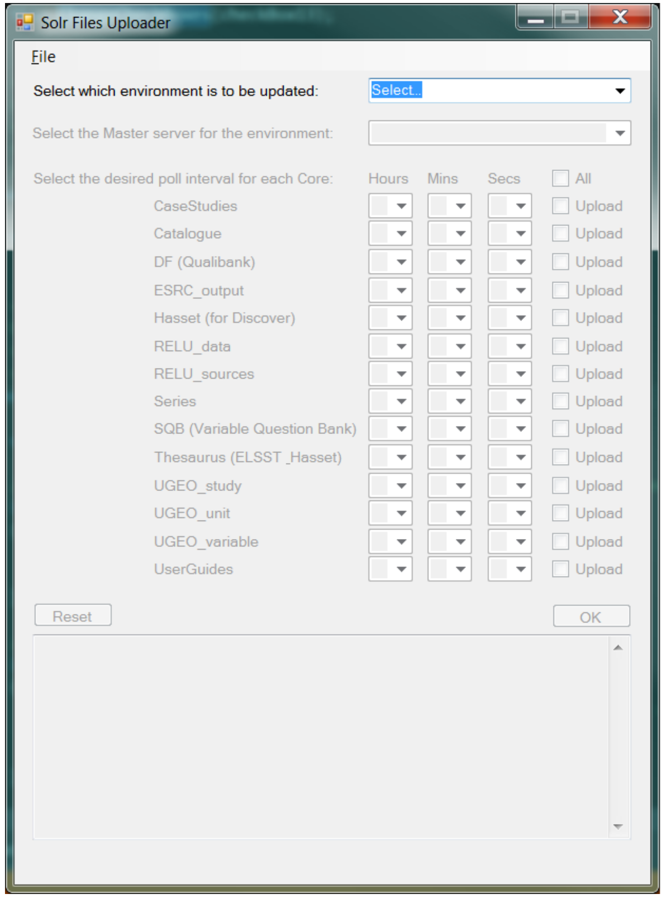
- Scalablity?
  - We normally perform Solr searches on the two ‘slave’ servers. This is configured in the load balancer and each Solr server in the environment.
  - **We could however put the ‘master’ server as a server used for Solr searches too, if required.**

Solr Replication implementation

- Scalablity?
  - We normally perform Solr searches on the two ‘slave’ servers. This is configured in the load balancer and each Solr server in the environment.
  - We could however put the ‘master’ server as a server used for Solr searches too, if required.
  - **We could easily add in another ‘slave’ server if required, it’s all in the configuration files.**

Solr Replication implementation

- Configuration
  - There is a new in-house tool that has been built for easily changing and uploading configuration files:



Solr Files Uploader

File

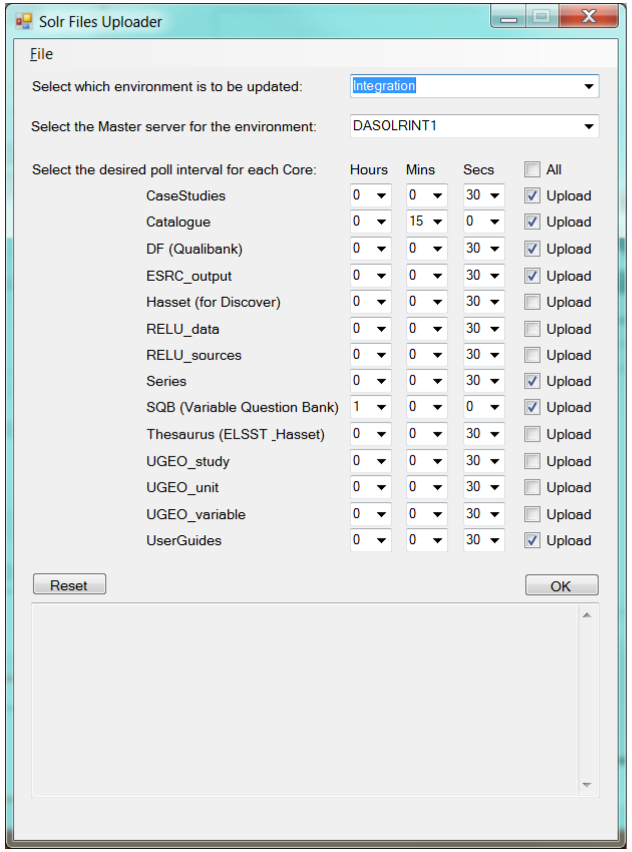
Select which environment is to be updated: Select..

Select the Master server for the environment:

Select the desired poll interval for each Core:	Hours	Mins	Secs	□ All
CaseStudies				□ Upload
Catalogue				□ Upload
DF (Qualibank)				□ Upload
ESRC_output				□ Upload
Hasset (for Discover)				□ Upload
RELU_data				□ Upload
RELU_sources				□ Upload
Series				□ Upload
SQB (Variable Question Bank)				□ Upload
Thesaurus (ELSST_Hasset)				□ Upload
UGEO_study				□ Upload
UGEO_unit				□ Upload
UGEO_variable				□ Upload
UserGuides				□ Upload

Solr Replication implementation

- Configuration
  - Environments, ‘master’ servers, Cores and replication polling times can easily be configured:



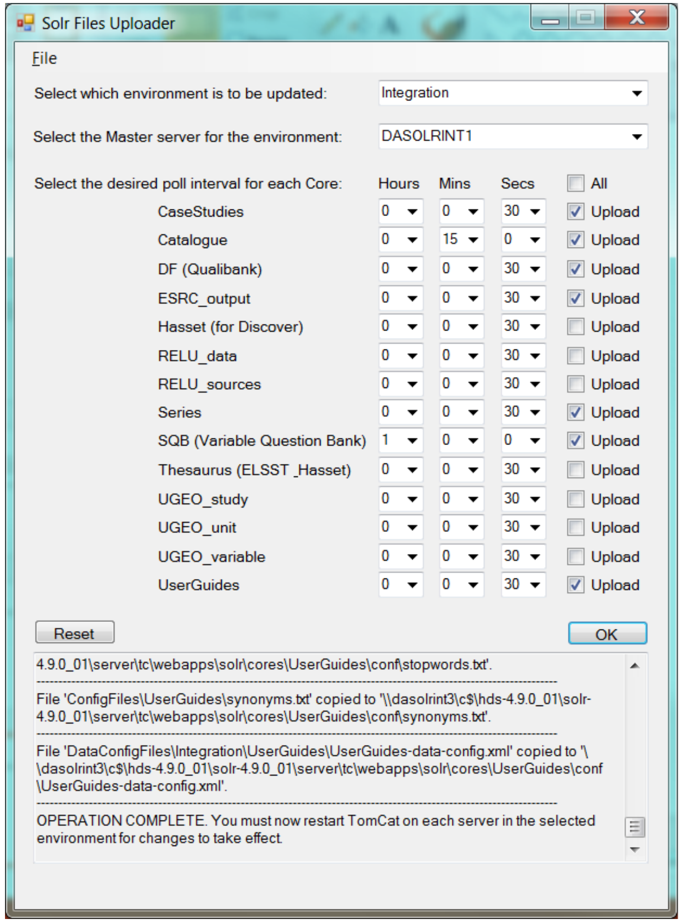
Select which environment is to be updated: Integration

Select the Master server for the environment: DASOLRINT1

Select the desired poll interval for each Core:	Hours	Mins	Secs	□ All
CaseStudies	0	0	30	☑ Upload
Catalogue	0	15	0	☑ Upload
DF (Qualibank)	0	0	30	☑ Upload
ESRC_output	0	0	30	☑ Upload
Hasset (for Discover)	0	0	30	□ Upload
RELU_data	0	0	30	□ Upload
RELU_sources	0	0	30	□ Upload
Series	0	0	30	☑ Upload
SQB (Variable Question Bank)	1	0	0	☑ Upload
Thesaurus (ELSST _Hasset)	0	0	30	□ Upload
UGEO_study	0	0	30	□ Upload
UGEO_unit	0	0	30	□ Upload
UGEO_variable	0	0	30	□ Upload
UserGuides	0	0	30	☑ Upload

Solr Replication implementation

- Configuration
  - Progress and event logging is presented clearly:



Select which environment is to be updated: Integration

Select the Master server for the environment: DASOLRINT1

Select the desired poll interval for each Core:	Hours	Mins	Secs	□ All
CaseStudies	0	0	30	☑ Upload
Catalogue	0	15	0	☑ Upload
DF (Qualibank)	0	0	30	☑ Upload
ESRC_output	0	0	30	☑ Upload
Hasset (for Discover)	0	0	30	□ Upload
RELU_data	0	0	30	□ Upload
RELU_sources	0	0	30	□ Upload
Series	0	0	30	☑ Upload
SQB (Variable Question Bank)	1	0	0	☑ Upload
Thesaurus (ELSST_Hasset)	0	0	30	□ Upload
UGEO_study	0	0	30	□ Upload
UGEO_unit	0	0	30	□ Upload
UGEO_variable	0	0	30	□ Upload
UserGuides	0	0	30	☑ Upload

```
4.9.0_01\server\tc\webapps\solr\cores\UserGuides\conf\stopwords.txt'.
--------------------------------------------------------------------------------
File 'ConfigFiles\UserGuides\synonyms.txt' copied to '\\dasolrint3\c$\hds-4.9.0_01\solr-
4.9.0_01\server\tc\webapps\solr\cores\UserGuides\conf\synonyms.txt'.
--------------------------------------------------------------------------------
File 'DataConfigFiles\Integration\UserGuides\UserGuides-data-config.xml' copied to '\
\dasolrint3\c$\hds-4.9.0_01\solr-4.9.0_01\server\tc\webapps\solr\cores\UserGuides\conf
\UserGuides-data-config.xml'.
--------------------------------------------------------------------------------
OPERATION COMPLETE. You must now restart TomCat on each server in the selected
environment for changes to take effect.
```

Solr Replication implementation

- Configuration
  - **Solr configuration files are under source control.**

Solr Replication implementation

- Configuration
  - Solr configuration files are under source control.
  - **The ‘solr.properties’ file, that should be present in each Core, controls the replication timeout for each Core. This is altered by the configuration application to the required timings (hours, minutes, seconds), and which server in the environment is to be the ‘master’.**

Solr Replication implementation

- Configuration
  - Solr configuration files are under source control.
  - The ‘solr.properties’ file, that should be present in each Core, controls the replication timeout for each Core. This is altered by the configuration application to the required timings (hours, minutes, seconds)), and which server in the environment is to be the ‘master’.
  - **The in-house load balancer must be configured separately and told what Solr servers to point at.**

Solr Replication implementation

- Configuration
  - Solr configuration files are under source control.
  - The ‘solr.properties’ file, that should be present in each Core, controls the replication timeout for each Core. This is altered by the configuration application to the required timings (hours, minutes, seconds)), and which server in the environment is to be the ‘master’.
  - The in-house load balancer must be configured separately and told what Solr servers to point at.
  - **The Web application should be pointed at the load balancer.**

Solr Replication implementation

- Conclusion