**FAST ESP Content Processing Workflow**

FAST ESP content processing workflow, Numbers in bold refer to the same bold red numbers in the diagram.

**Integrity proxy:**

For each new "control unit" (.ctl) - "load unit" (.zip) pair, after integrity proxy receives a "status" file (.sts) back from each client **(6)**, it then uploads a status file (.sts) back to the FTP repository **(7)** to indicate that the control/load unit has been processed by each integrity client and all of its content has been persisted to the appropriate (hashed) integrity client content stores. The integrity client content stores allow us to refeed any content to FAST (either individual docs or control/load units) as necessary, e.g., if content needs to be refed because of a change to the document processing.

**Integrity clients:**

The total content is distributed across 5 machines because it's too large to fit on a single machine plus it allows for faster, parallel processing. Each client runs every hour from Wed, 3PM UTC thru Sat 3PM UTC. During each run, it asks the integrity proxy if anything new is available to process. If so then each integrity proxy:

 copies the new content (.ctl and .zip files) to its own "content store" **(4)**

 extracts the "portion" of content that it's responsible for.  A hash function is applied to each docid that maps it to 1-5 and if this matches the "index" of the integrity client, then that client is responsible for that doc and it is persisted on the client as an individual file

 sends the docs it is responsible for to "fidelity" **(5)**

 Returns a "status" file back to integrity proxy **(6)** , indicating that it finished processing the new content it's responsible for

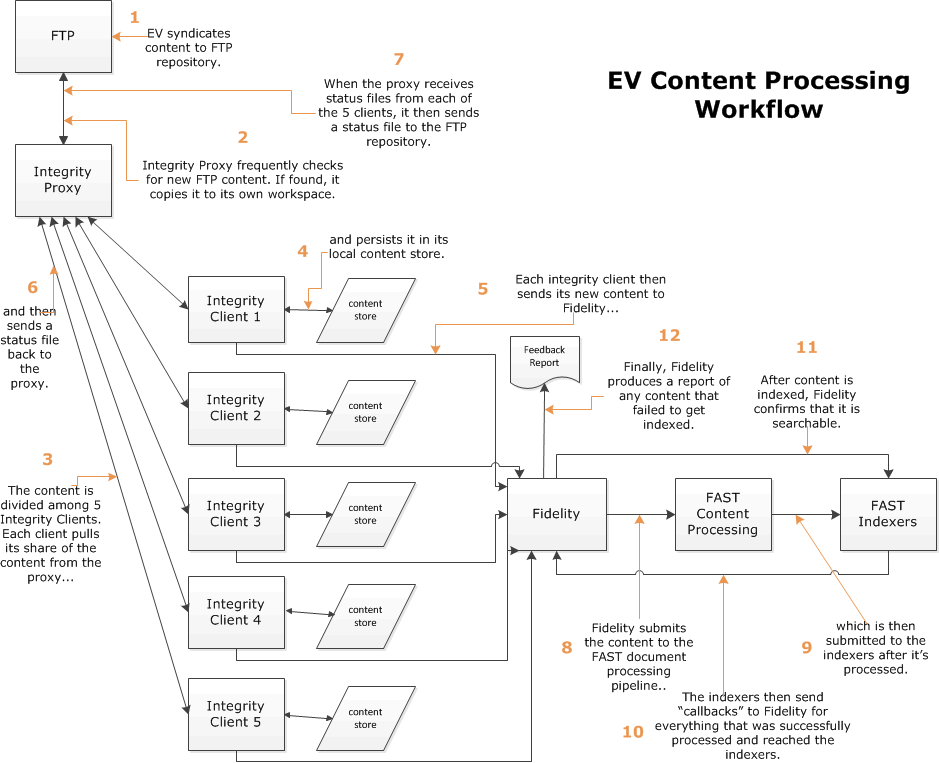
**Fidelity**

 feeds each doc that it receives to the FAST ESP document processing pipeline **(8)**

 If a doc is successfully processed (means the source content is successfully processed into an internal format that can be ingested by the indexers) and reaches the indexer, a "callback" is sent back to fidelity **(10)**. Any docs that don't receive callbacks lets us identify and report docs that failed to get processed. (Very rare, in fact I haven't seen this happen.)

 after a doc is indexed, a confirmation query against the docid and its new load/control unit is run (this doesn't interfere with production queries) to confirm that the doc is searchable **(11)**.  If a doc fails to get confirmed it may be because the doc failed to get indexed, e.g., it has a word that exceeds the maximum length or perhaps there was a transient issue, e.g., some queries were submitted before a search component loaded the new index. We have monitoring tools that show the status of each indexer, including the number of non-indexed docs, so we can identify an index that may have issues. We can also identify the docs that failed to be confirmed and if necessary, resubmit them to the document processing pipeline to re-route them to the indexers to re-index them and try again (this is rare).

 Fidelity produces a "feedback report" of any docs that failed to get indexed. Typically this only includes docs that failed for legitimate reasons (this is rare).

****