Presentation Layer

- jQuery.js
- less.js
- Bootstrap.js
- backbone.js
- d3.js
- highcharts

Webserver/ API Layer

- **nginx** — Webserver, SSL Termination
- **Frontend Service/ API Service** — HTML Service/ REST Service

Security& Support Layer

- **Cluster Service** — Handles all aspects of Cluster Management. Can control all other services.
- **User Service** — Provides Authentication and Authorization and User/Group/Role information
- **Configuration Service** — Provides Configuration for all Services per Tenant, Project, Group or User
- **Scheduler Service** — Run Tasks(e.g. trigger mails)
- **Email Service** — Send Mails via SMTP Server

Data Layer

Data Provider Service

- RSS/Atom
- Bing News
- Bing Web
- Blog Search
- Twitter
- YouTube
- Reddit
- Track Site
- Crawler
- CSV/Excel
- SDK/REST
- ...

Standard API for Data Loading

Pipeline Service

- Language Detection
- Thumbnail Generation
- Content Converting — Apache Tika
- Deduplication/ RelatedStory Detection
- Pipelet Plugins
- Filtering/ Alerting

Asynchronous Data Processing Pipeline, pluggable adaptor for Queuing based on Redis, SQS, ActiveMQ etc.

Topic Service

- Indexing
- Search/ Analzyse
- SmartFilter

Provides access to the documents for indexing and querying

Persistence Layer

- **Filesystem** — Stores and serves binary documents. In clustered setup GlusterFS is used.
- **MySQL** — Provides configuration databases for all services
- **Redis** — Used for query caching, queuing and as in flight data storage
- **Zookeeper** — Used by the cluster for leader election and cluster state
- **Elasticsearch** — IR framework based on lucene, used to store documents.

OS Layer

- **RHEL or CentOS**
- **Syslog** — All logs can be forwarded using syslog
- **Statsd** — Used by all services to track load and performance
- **Monit** — Monitors and restarts all services
- **Puppet** — Configures and controls the whole system

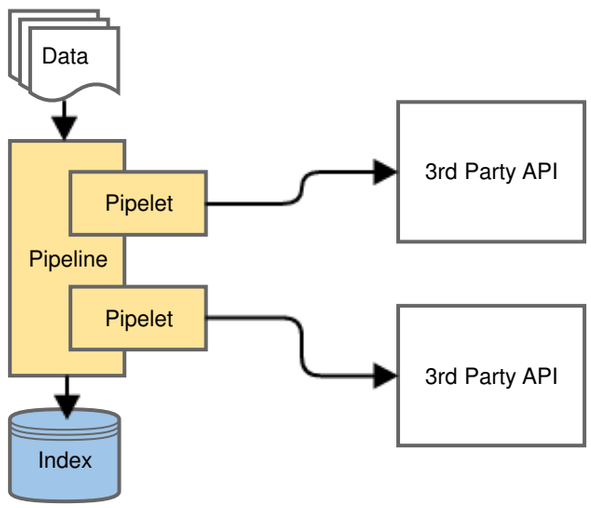
Each Squirro Python Service consists of the following stack:

- nginx(webservice)
- WSGIService(API framework)
- Application Code
- Storage Adapter
- Storage Systems(persistence)

All services provide a restful, versioned, stateless API using https://github.com/pneff/wsgiservice

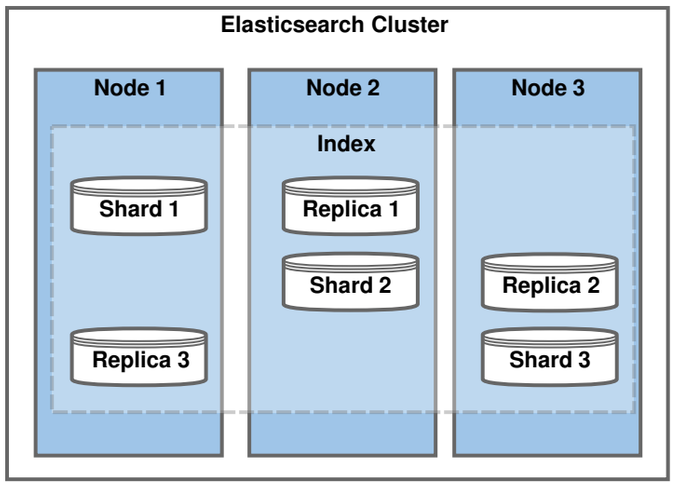
The storage layer is optional and not shared among the services. e.g. the topic service can only access information in the users service database through the user services API.

The Data Pipeline can be extended by Plugins(Pipelets) to filter, clean& enrich the data.



Some of the commonly used 3rd party systems are: TextRazor, AlchemyAPI, Lexalytics, OpenCalais. While these are cloud based API's most can be deployed on premise.

ElasticSearch is distributed, which means that indices can be divided into shards and each shard can have zero or more replicas. Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shard(s). Rebalancing and routing are done automatically



Legend: JavaScript | Python | Java | Linux

Version 1.4- 2016/02/11