

# Droonga の 紹介

未来検索ブラジル  
森大二郎





# 今日の内容

- Droongaとは?
- 設計方針
- 実装
- 現状
- 今後の予定



# Droongaって何ですか？

- Distributedの"D"
- リアルタイム分散処理エンジン
- 分散データストア
- 分散型全文検索エンジン



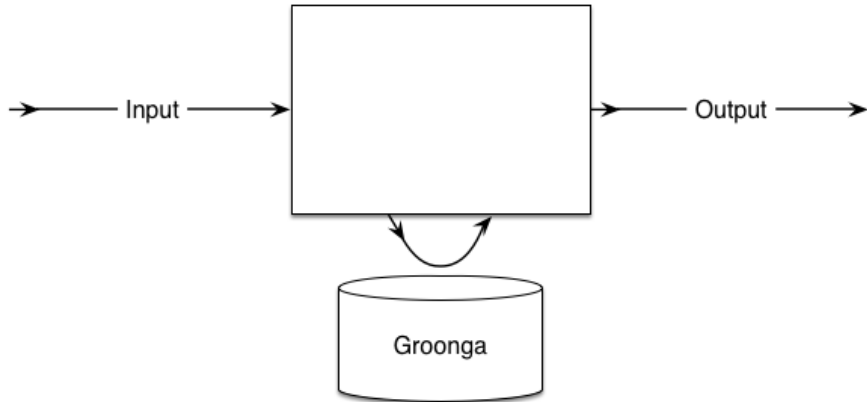
# 設計方針

- ストリーム指向
- リアルタイム指向
- 可用性重視
- 拡張性重視

ストリー  
ム指向



# ストリーム処理

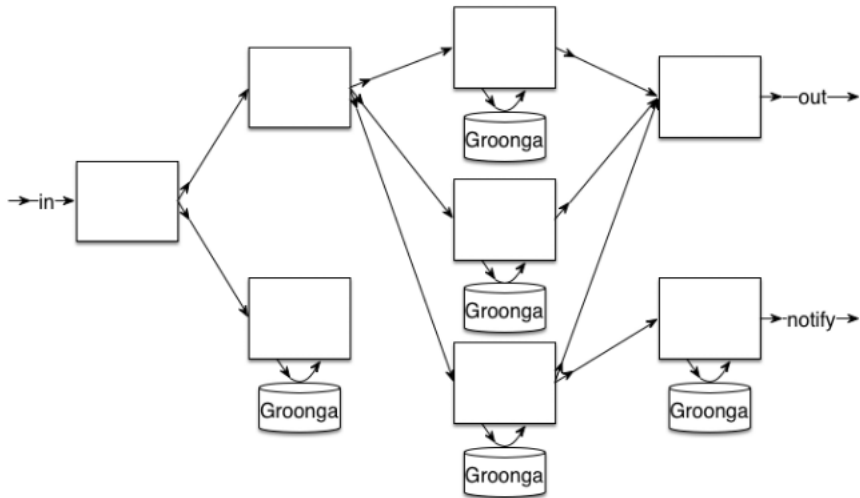




# ストリーム処理

- 入出力: JSONデータストリーム
- 入力データを処理
- データストアを参照・更新  
(必要なら)
- 0個以上のデータを出力

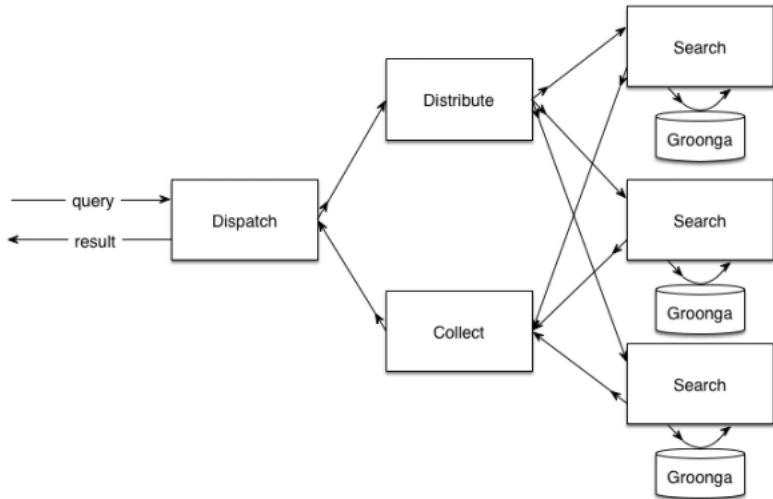
# カスケードして様々な処理







# 分散検索エンジンも..



リアルタイム指向

# ここで言うリアルタイム:

- 情報が発生したら
  - 必要とする人に
  - 最短の遅延時間で届ける
- ... そのために

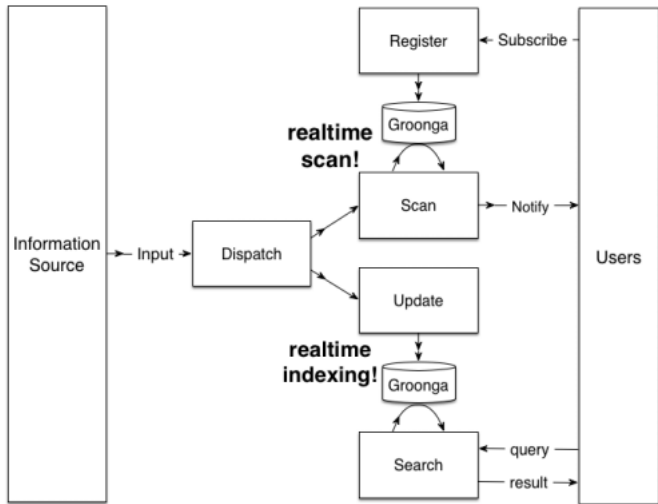


# 情報を受け取ったら

- 索引をすぐに更新する
- 購読してた人に即届ける



# リアルタイム指向





# 可用性重視

- shared nothing
- レプリケーション
- 結果整合性重視

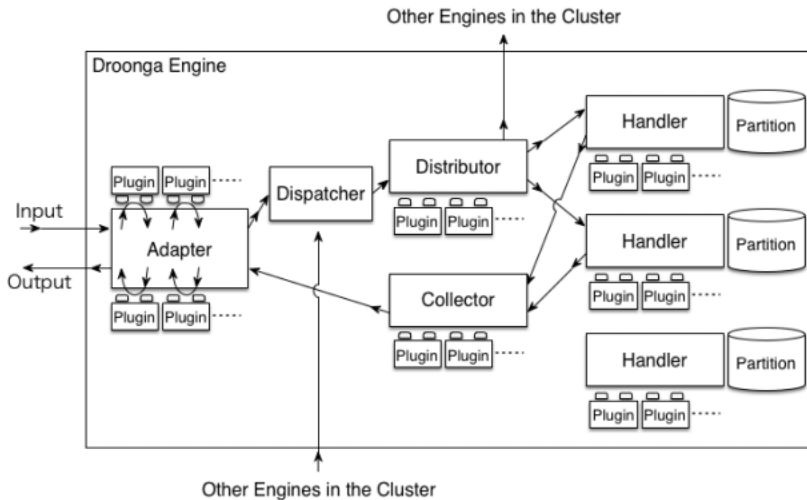


# 拡張性重視

- Pluginで機能拡張
- Rubyで書ける!



# Pluginで拡張







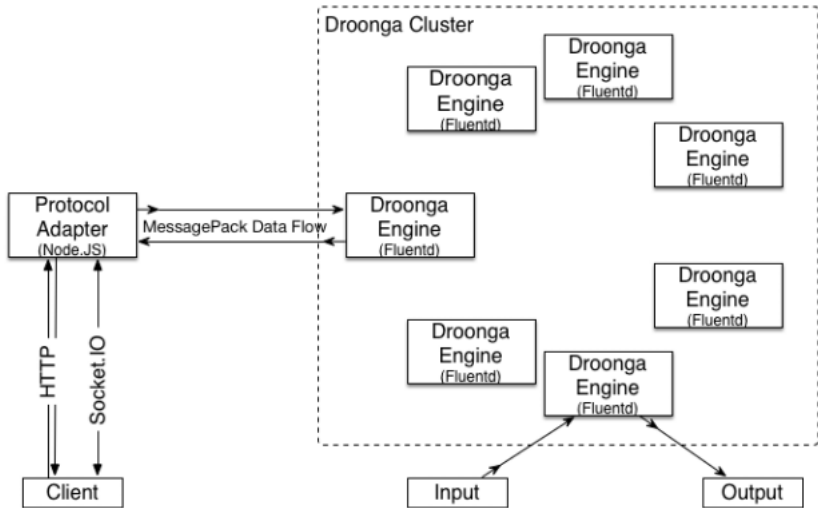
# 実装

二つのコンポーネント

- Protocol Adapter
- Droonga Engine



# コンポーネント





# Droonga Engine

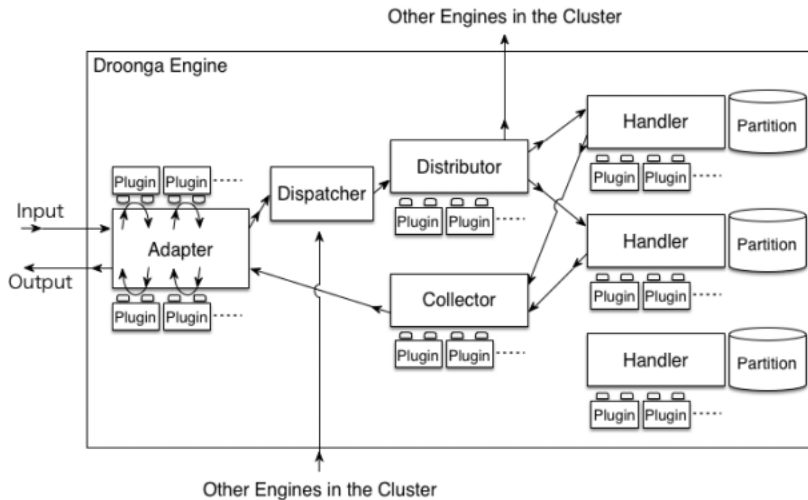
- 実処理エンジン
- 複数エンジンでクラスタを構成
- 通信: MessagePack Stream
- Fluentdのplugin



# Protocol Adapter

- APIを提供
  - HTTP (Rest) API
  - Socket. IO (WebSocket) API
- Node. JS上で動作
- Expressのplugin

# Engineのモジュール構成





# モジュール構成

- Adapter: JSONデータを整形
- Dispatcher: 処理フローを計画
- Distributor: データを配布
- Handler: 実処理
- Collector: データを集約

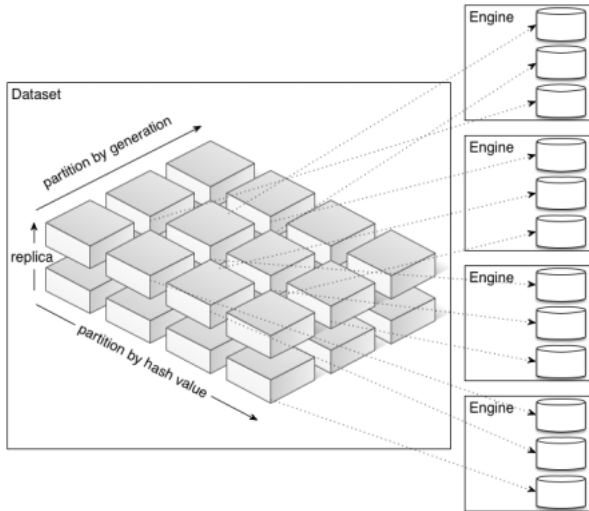


# クラスタの構成

- 1クラスタに複数のdataset
- dataset毎にpartition/replica数を設定
- ハッシュ値と世代でpartition
- catalog.json
  - 全パーティション構成を記述
  - 全ノードで共有



# クラスタの構成







# 現状

- version0.7.0 リリース!
- <http://droonga.org/>
- 分散検索が一通り動作!
- チュートリアルあります



# 使い方

```
gem install fluent-plugin-droonga  
fluentd.confとcatalog.jsonを準備
```

```
<source>  
  type forward  
  port 24224  
</source>  
<match taiyaki.message>  
  name localhost:24224/taiyaki  
  type droonga  
  proxy true  
</match>  
<match output.message>  
  type stdout  
</match>
```

```
fluentd --config fluentd.conf &  
fluent-cat taiyaki.message < jsons
```



# 今後の予定

version0.8.0 (安定版)

2013/12/29リリース予定

version0.9.0 (開発版)

Groonga Http Server 互換機能  
2014/1/29リリース予定

version1.0.0 (安定版)

2014/2/9リリース予定



# 協力者募集してます!

- バグレポート
- Plugin開発