ClearCode

Hatoholのログ監視機能 2014/10版

須藤功平

株式会社クリアコード

2014/10/07

内容

- Hatoholのログ監視機能の概要
  - ただし2014年10月時点での情報
- 詳細はWikiを参照
  - https://github.com/project-hatohol/hatohol/wiki/Log-monitoring

目的

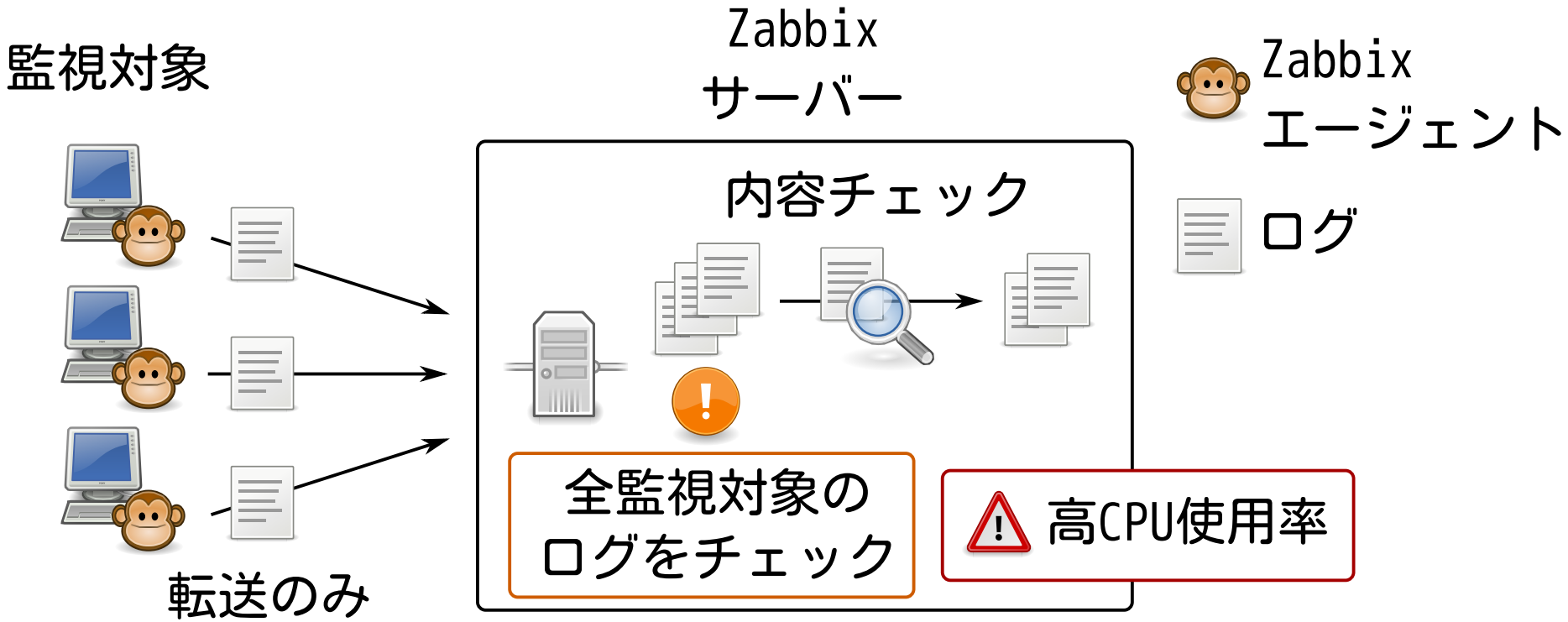
- ✓ 現状を共有すること

解決したい問題

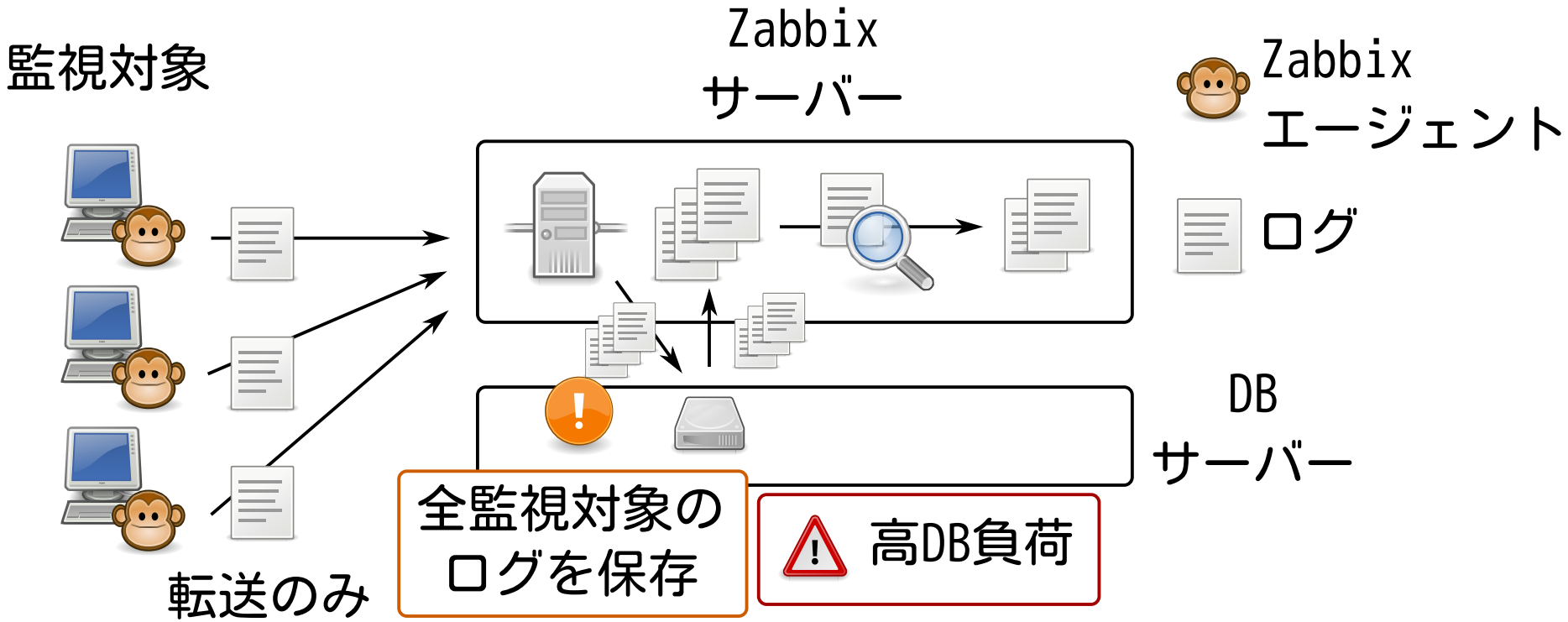
Zabbixのログ監視機能の問題点

- ✓ サーバーのCPU使用率が高い
- ✓ 大量のログだとDBの負荷が高い
- ✓ エージェント・サーバー間の通信が安全ではない

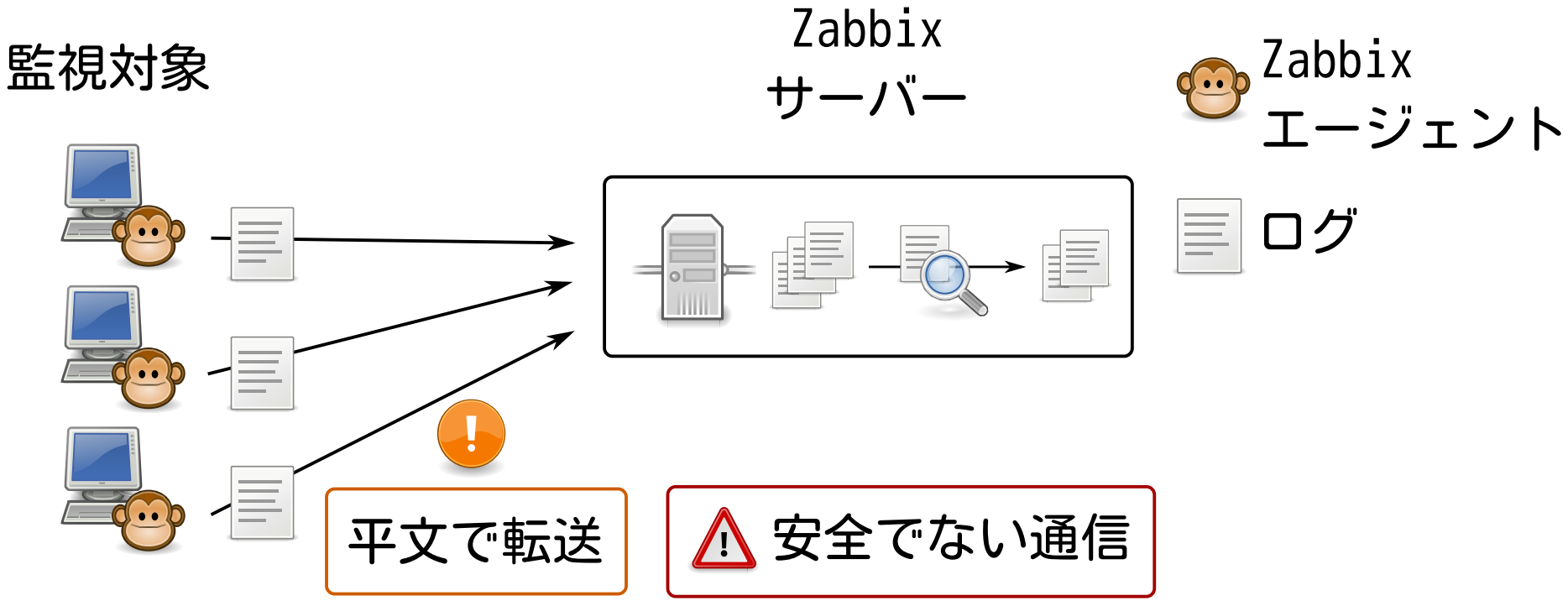
Zabbix：高CPU使用率



Zabbix：高DB負荷



Zabbix：安全でない通信



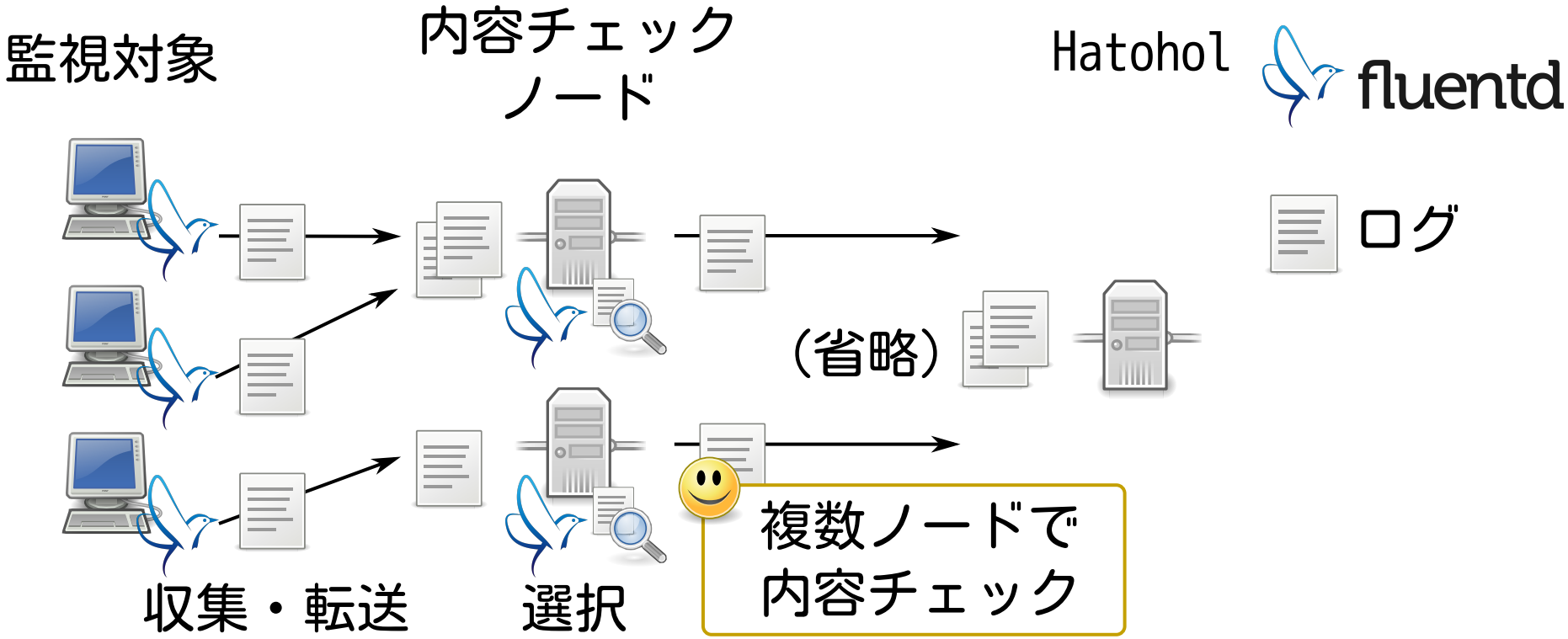
解決方針

- Fluentdと連携
- Fluentd：データ配送システム
  - ログ収集
  - フィルター・転送
  - 出力

解決方法：高CPU使用率

- ✓ サーバーのCPU使用率が高い
  - ✓ 処理を複数ノードで分散
- ✓ 大量のログだとDBの負荷が高い
- ✓ エージェント・サーバー間の通信が安全ではない

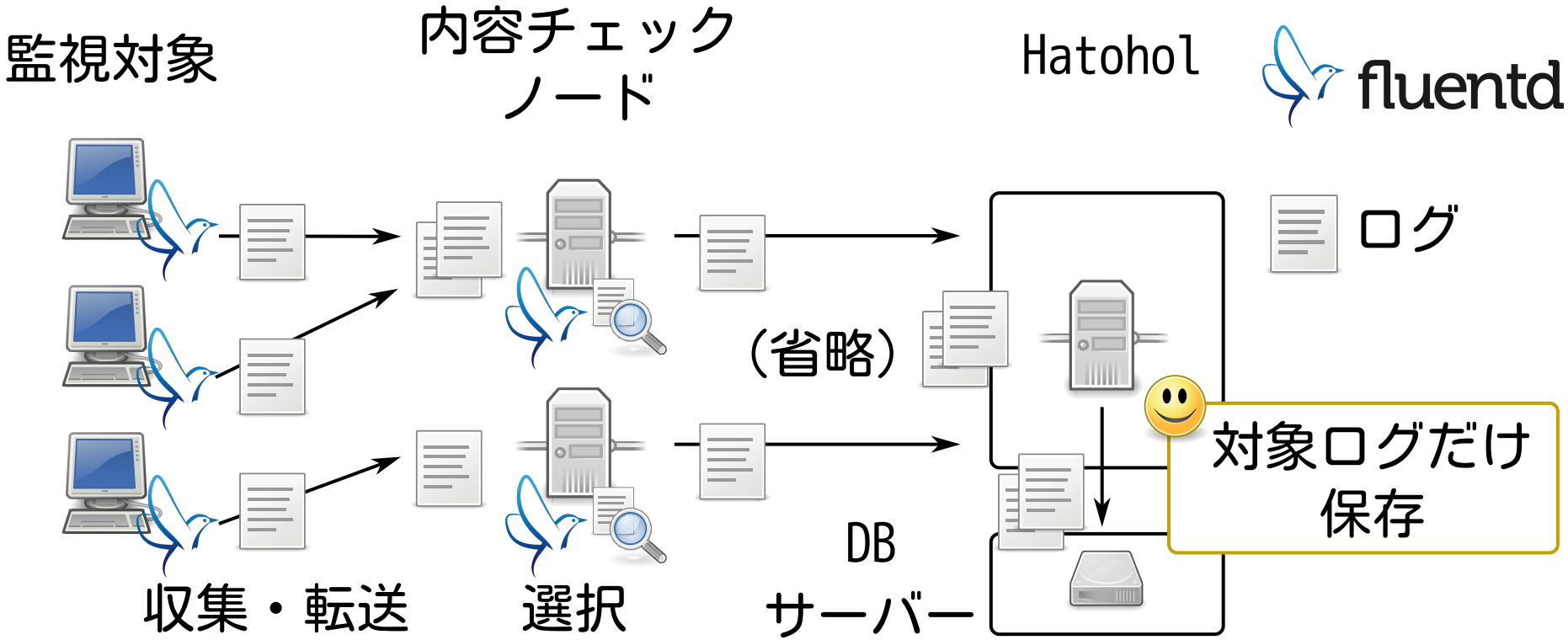
処理を分散



解決方法：高DB負荷

- サーバーのCPU使用率が高い
- 大量のログだとDBの負荷が高い
  - 対象ログのみ保存
- エージェント・サーバー間の通信が安全ではない

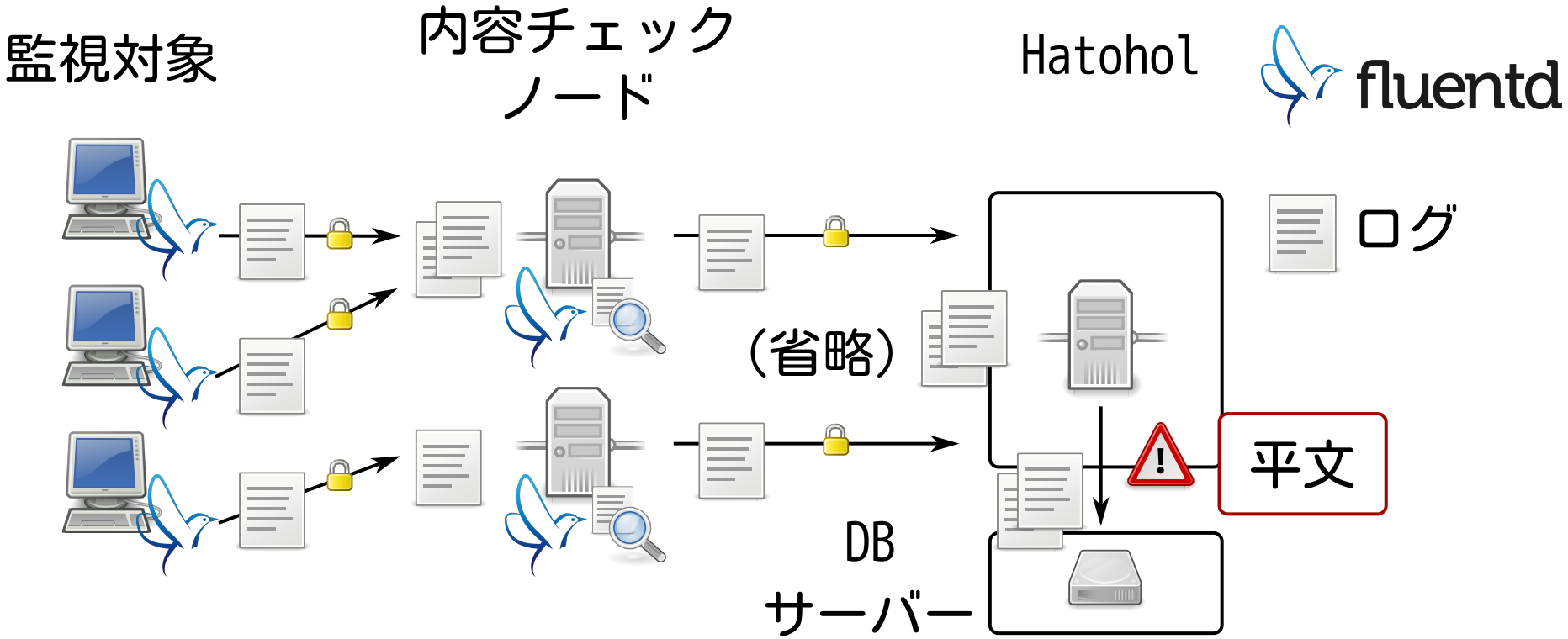
対象ログのみ保存


監視対象
内容チェック
ノード
Hatohol
fluentd
ログ
（省略）
対象ログだけ
保存
DB
サーバー
収集・転送
選択

解決方法：安全でない通信

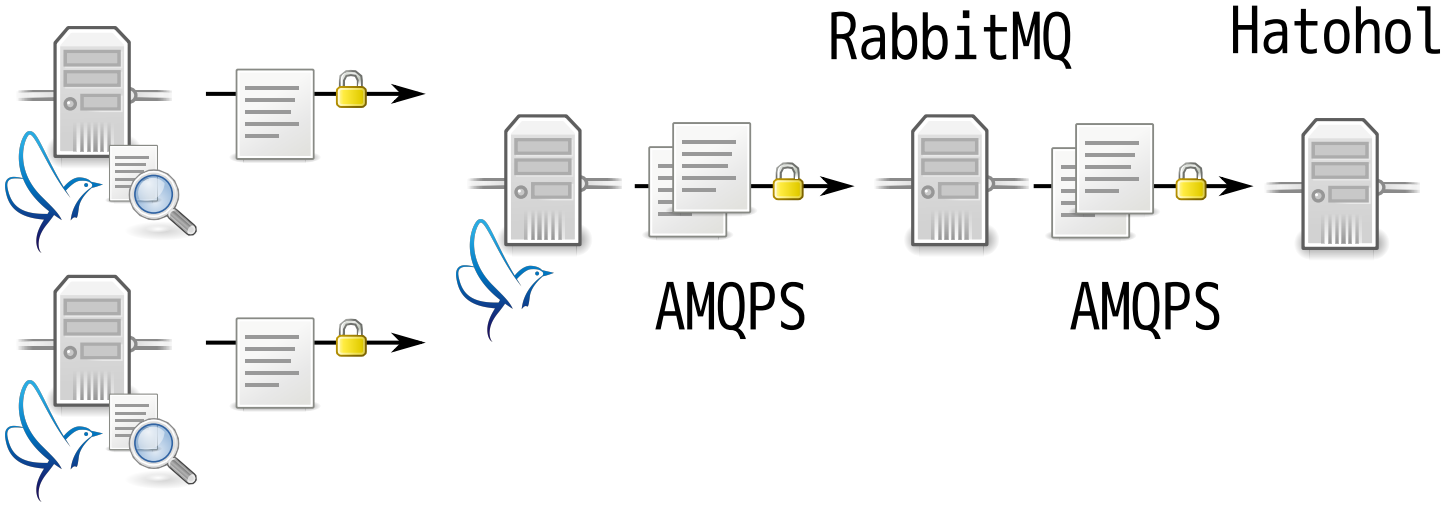
- サーバーのCPU使用率が高い
- 大量のログだとDBの負荷が高い
- エージェント・サーバー間の通信が安全ではない
  - 通信路を暗号化

通信路を暗号化



通信路を暗号化（AMQP）

内容チェック
　ノートAMQPプロデューサー



fluentd

ログ

解決

- ✓ サーバーのCPU使用率が高い
  - ✓ 処理を複数ノードで分散
- ✓ 大量のログだとDBの負荷が高い
  - ✓ 対象ログのみ保存
- ✓ エージェント・サーバー間の通信が安全ではない
  - ✓ 通信路を暗号化
    （Hatohol・DB間は安全ではない）

課題

- ✓ 導入が面倒
- ✓ 導入後の設定が面倒

課題：導入が面倒

- ✓ ノード数が増える
- ✓ TLSの設定が増える

ノード数：Zabbix

- ✓ サーバー：数台
- ✓ エージェント：ホスト数

ノード数：Hatohol

- 同じ
  - サーバー：1台
  - 収集用Fluentd：ホスト数
- 増加
  - RabbitMQ：1台
  - 監視用Fluentd：数台
  - AMQPコンシューマーFluentd：数台

TLSの設定

- ✓ 認証局を作成
- ✓ 各ノード用の鍵を作成
- ✓ 認証局で公開鍵証明書を発行
- ✓ ↑を使う設定を追加

認証局を作成

```
[ca]# hatohol-ca-initialize
```

ノード用の鍵を作成

```
# クライアント用
[client1]% hatohol-client-certificate-create \
  --host-name client1.example.com
# サーバー用
[server1]% hatohol-server-certificate-create \
  --host-name server1.example.com
```

証明書を発行

```
# クライアント用
[client1]% scp req.pem ca:
[ca]# hatohol-ca-sign-client-certificate \
  /../req.pem
# サーバー用
[server1]% scp req.pem ca:
[ca]# hatohol-ca-sign-server-certificate \
  /../req.pem
```

設定：RabbitMQ

```
[
  {rabbit, [
    {ssl_listeners, [5671]},
    {ssl_options, [
      {cacertfile, "/etc/rabbitmq/ca-cert.pem"},
      {certfile, "/etc/rabbitmq/server-cert.pem"},
      {keyfile, "/etc/rabbitmq/key.pem"},
      {verify, verify_peer},
      {fail_if_no_peer_cert, false}
    ]}
  ]}
].
```

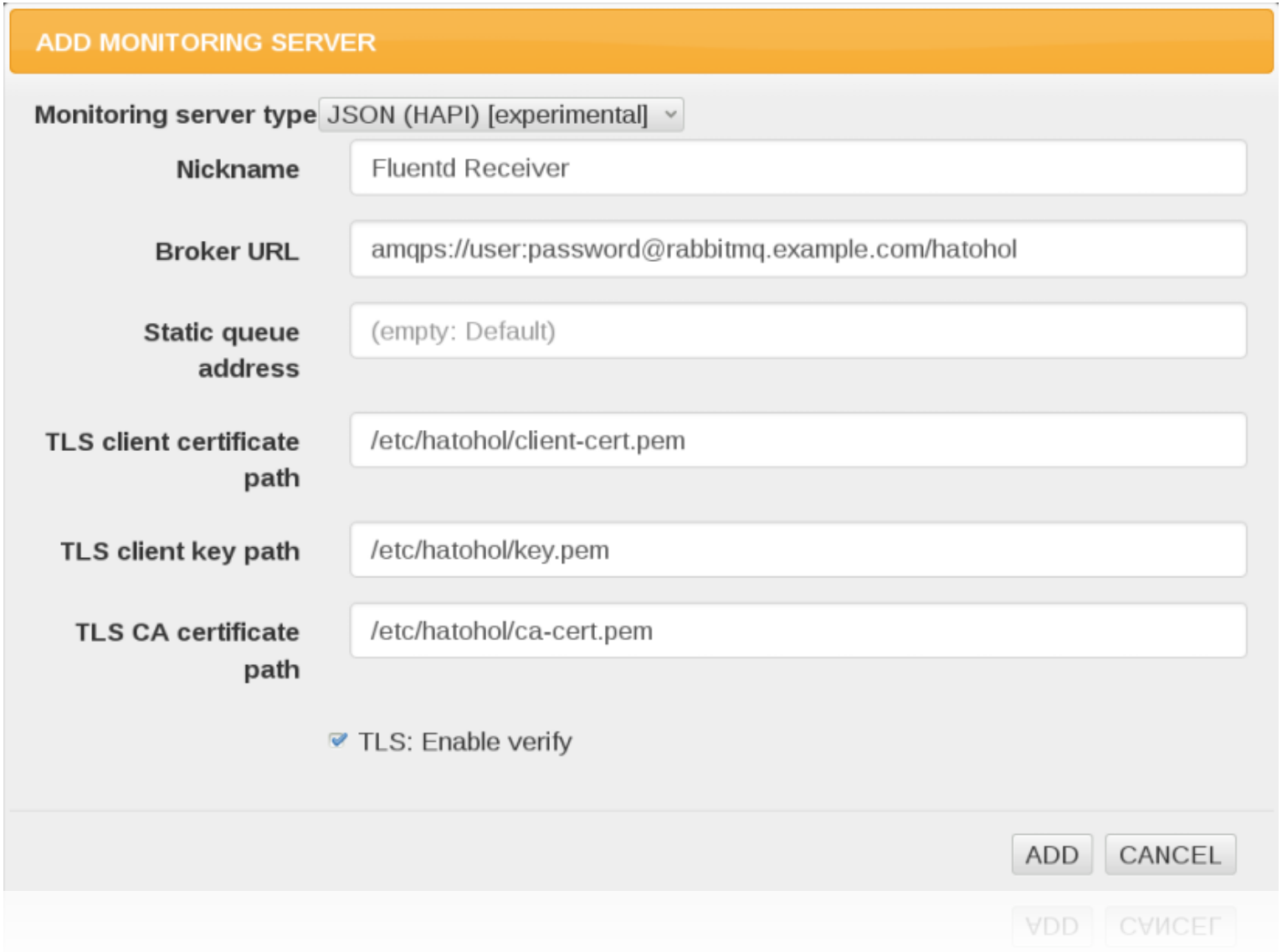
設定：Fluentd

```
<match hatohol.**>
  type hatohol

  url "amqps://user:password@rabbitmq.example.com/hatohol"

  tls_cert "/../client-cert.pem"
  tls_key "/../key.pem"
  tls_ca_certificates ["/../ca-cert.pem"]
</match>
```

設定：Hatohol


ADD MONITORING SERVER
Monitoring server type
JSON (HAPI) [experimental]
Nickname
Fluentd Receiver
Broker URL
amqps://user:password@rabbitmq.example.com/hatohol
Static queue address
(empty: Default)
TLS client certificate path
/etc/hatohol/client-cert.pem
TLS client key path
/etc/hatohol/key.pem
TLS CA certificate path
/etc/hatohol/ca-cert.pem
TLS: Enable verify
ADD
CANCEL

課題：導入後の設定が面倒

- 設定は各ノードで行う
  - Zabbixはサーバーで一括管理
  - fluentd-server：設定を配布
- Fluentdの設定GUIがない
  - fluentd-ui：いくつか設定可能
  - ↑ローカルのFluentdを設定

設定はどこから実施する？

- 前提
  - Hatohol利用者はZabbixに
    直接アクセスできないかもしれない
  - 監視環境内のノードにアクセス不可
- 全部Hatohol経由で設定？
  - Zabbixも？Fluentdも？Redmineも？

今後の方向

- 課題
  - ホスト管理機能と連携することで解決可能？
- 改善
  - 実際に使ってフィードバック
  - 使ってもらってフィードバック

フィードバック対応案

- 移行方法集の作成？
  - Zabbixでのログ監視設定と
    Hatohol（Fluentd）での設定の対応
- ワークフローの提案？
  - ZabbixとHatoholで
    監視システム管理のワークフローは
    同じでいいの？もっと改善できる？