

camome勉強会

## camome

- みんなの研究を実現するWebアプリケーション

### <実現したいこと>

- Googleカレンダーライクなカレンダー
- 予定の予測
- アプリケーション情報の整理
- メールの読み書き

# 現在のcamome

3つの環境に分岐している

nomlab/master

みんなで開発しているブランチ

kobayashi/demo

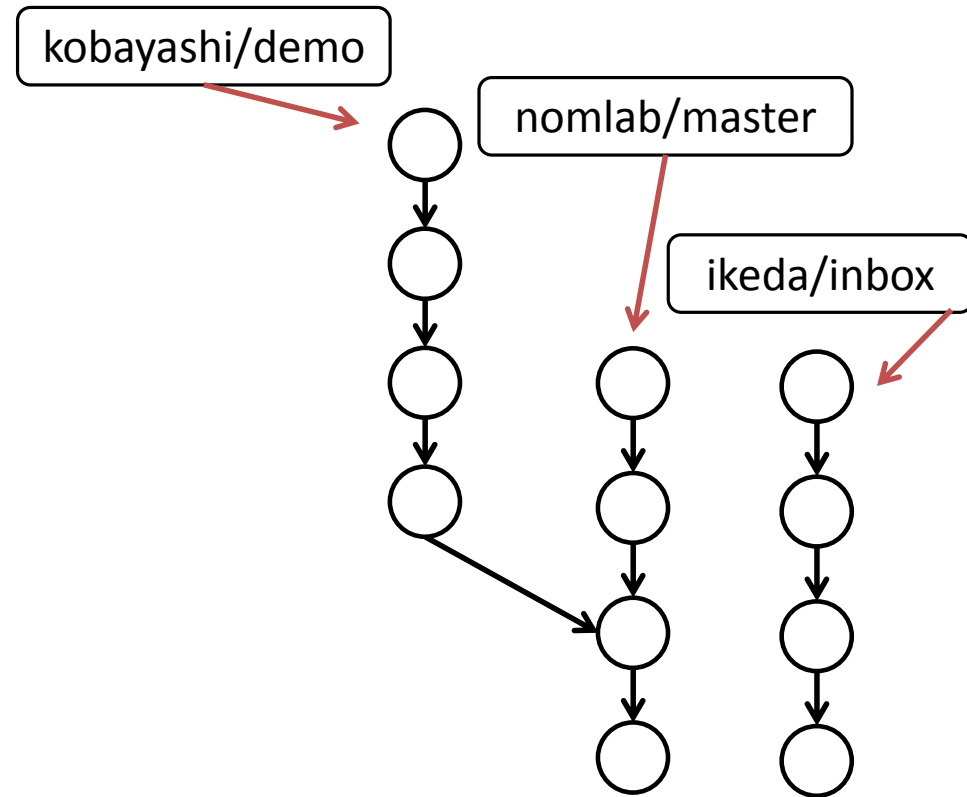
小林のデモ環境用ブランチ

ikedda/inbox

池田個人の別プロジェクト

nomlab/masterと機能は一部かぶってるけど、コードは全く別

正確にはcamomeではない



## nomlab/master

大きく分けて3つの機能を保持

カレンダー (/events/)

- 予定のCRUD
- 過去の予定を一覧表示&コピー
- Googleカレンダーの予定をインポート (同期は不可)
- 1ユーザ1カレンダー (複数のカレンダーは持てるが、作るインタフェースがない)

カレンダーの予定を整理するためのInbox (/inbox/recurrences)

- 予定をRecurrenceに整理
- 新しいRecurrence作成

メールを整理するためのInbox (/missions/inbox)

- メールと予定を関連付けて管理 (中間テーブルを介して)
- メールの再利用
- メールの読み書き

## nomlab/master (理想形)

### カレンダー (/events/)

- 予定のCRUD
- 過去の予定を一覧表示&コピー
- Googleカレンダーの予定をインポート (同期は不可)
- 1ユーザ1カレンダー (複数のカレンダーは持てるが、作るインタフェースがない)



### カレンダー (/events/)

- 予定のCRUD
- 過去の予定を一覧表示&コピー
- Googleカレンダーと**同期**
- **1ユーザ複数カレンダー**
- **予定の予測**

## nomlab/master (理想形)

カレンダーの予定を整理するためのInbox /(inbox/recurrences)

- 予定をInboxに一覧表示
- 新しいRecurrence作成
- 予定をRecurrenceに整理

メールを整理するためのInbox (/missions/inbox)

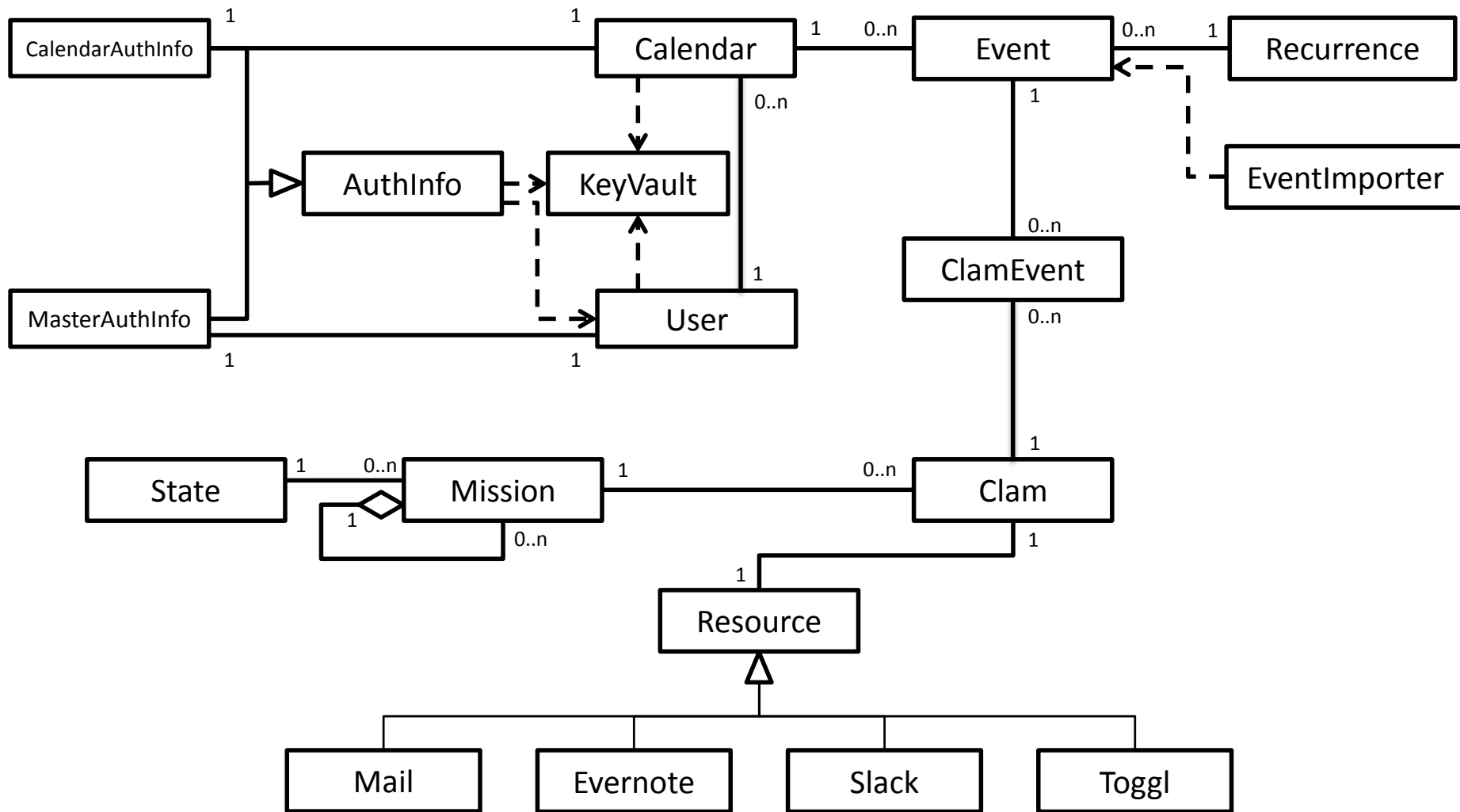
- メールと予定を関連付けて管理 (中間テーブルを介して)
- メールの再利用
- メールの読み書き



アプリケーション情報を整理するInbox

- カレンダー, メール, 文書, Webブラウザ, チャットの会話をInboxに一覧表示
- Mission, Task, Recurrence作成
- Mission, Task, Recurrenceに関してそれぞれ整理
- MissionとTaskを階層構造で表現

# クラス図 (現状)



# AuthInfo

何かの認証情報をもつモデル

encrypted\_pass: 暗号化されたパスワード

salt: 暗号化際に付与されるデータ

(入力パスワード + salt)でハッシュ化 = encrypted\_pass で認証

## AuthInfo

login_name:	string
encrypted_pass:	string
salt:	string
url:	string
type:	string
parent_id:	integer
parent_type:	string



# AuthInfo

何かの認証情報をもつモデル

encrypted\_pass: 暗号化されたパスワード

salt: 暗号化際に付与されるデータ

(入力パスワード + salt)でハッシュ化 = encrypted\_pass で認証

## CalendarAuthInfo

AuthInfoを継承したモデル

Googleカレンダーの認証情報

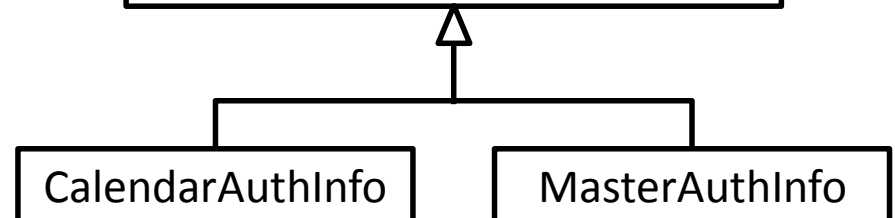
## MasterAuthInfo

AuthInfoを継承したモデル

Userの認証情報

## AuthInfo

login_name:	string
encrypted_pass:	string
salt:	string
url:	string
type:	string
parent_id:	integer
parent_type:	string

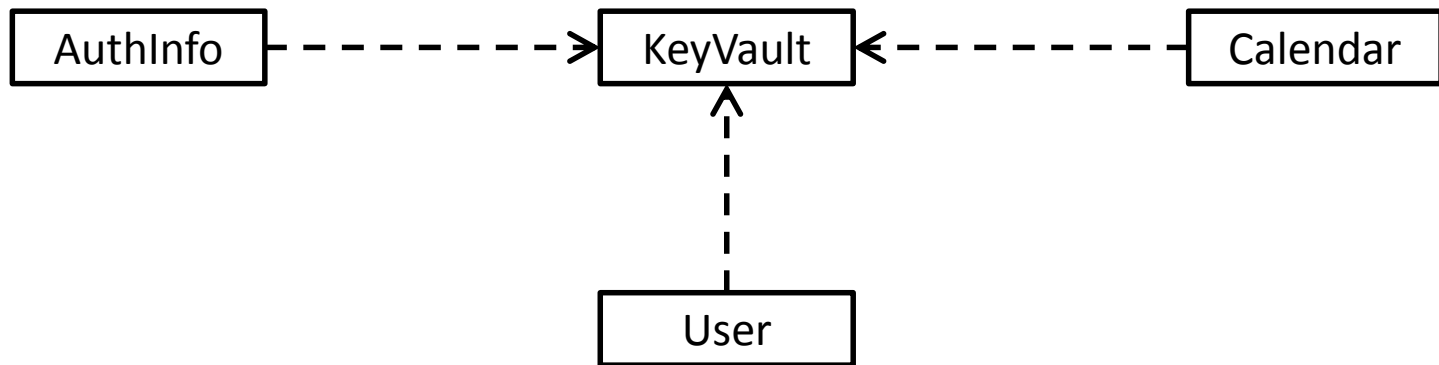


# KeyVault

暗号化周りを受け持つクラス

- 受け取ったauth\_infoに暗号化されたパスワードを格納
- 受け取ったauth\_infoに復号されたパスワードを格納

User, Calendar, AuthInfo(CalendarAuthInfo, MasterAuthInfo)が利用



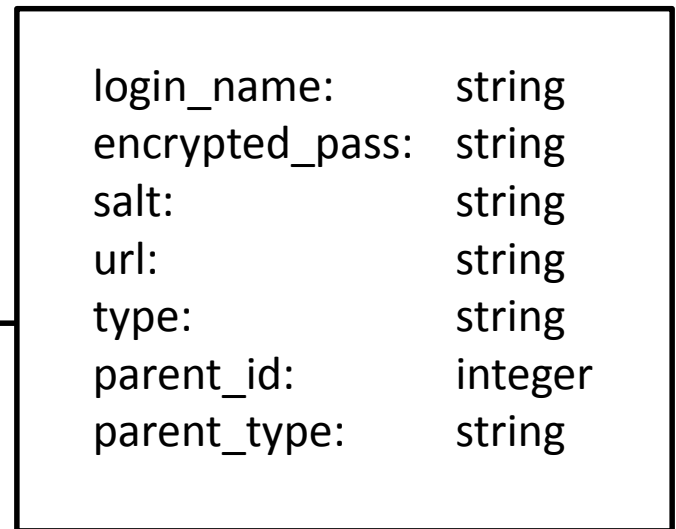
# User

- ユーザの情報をもつ
- MasterAuthInfoと1対1で関連 (外部キー: parent\_id)

## User



## MasterAuthInfo



# Calendar

- カレンダの情報をもつ
- CalendarAuthInfoと1対1で関連 (外部キー: parent\_id)
- Userとは多対1で関連

## Calendar

displayname:	string
color:	string
description:	text
user_id:	integer

## CalendarAuthInfo

login_name:	string
encrypted_pass:	string
salt:	string
url:	string
type:	string
parent_id:	integer
parent_type:	string

1

1

## Event

- 予定の情報をもつ
- カラムはicalendar形式にそろえてある？

## EventImporter

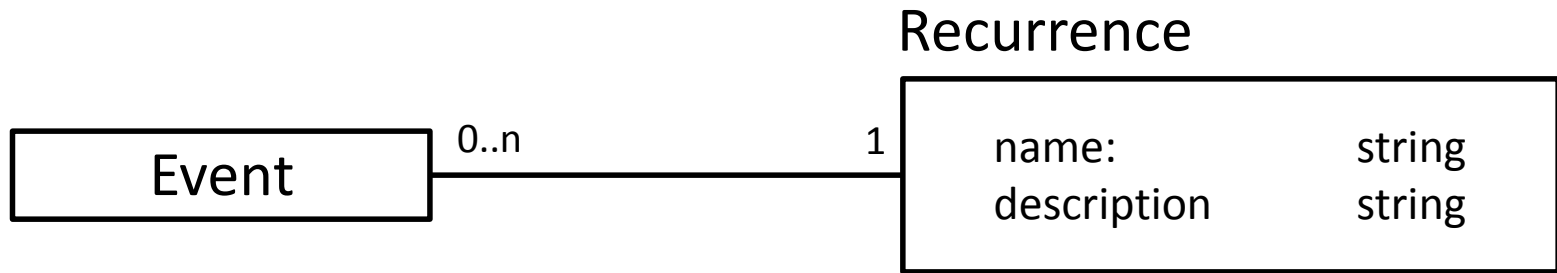
- Googleカレンダーから予定を取得するクラス

## Event

uid:	text
categories:	text
description:	text
location:	text
status:	text
summary:	text
dtstart:	datetime
dtend:	datetime
recurrence_id:	integer
related_to:	text
exdate:	datetime
rdate:	datetime
created:	datetime
last_modified:	datetime
sequence:	sequence
rrule:	rrule
calendar_id:	integer

# Recurrence

- 繰り返される予定のまとまりを表現
- 同じrecurrence\_idをもつ予定群は繰り返し発生している

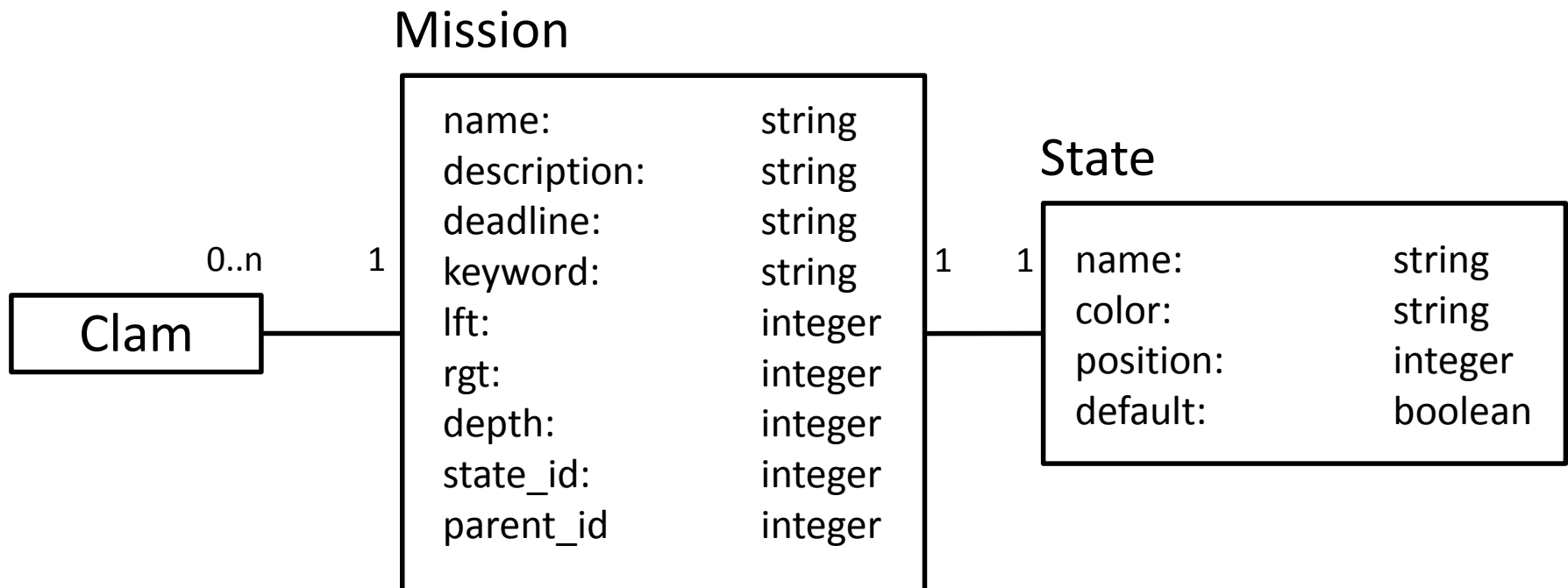


## Mission

- 同じ作業に関するClamの集まりを表す

## State

- Missionの状態を表す (todo, done, waitなど)



# Clam

- 各アプリケーションから収集した情報を統一的な規格に変換したもの
- 貝殻の意味(クラムチャウダーのクラム)

各APPから情報を収集

- 形式やもつ情報がさまざま
- Inboxに一覧表示しにくい



- メタデータ(summary, date, uid)を抜き出してClamとする
- ClamをInboxに一覧表示

メタデータじゃない生データをもつモデルも存在

Resource

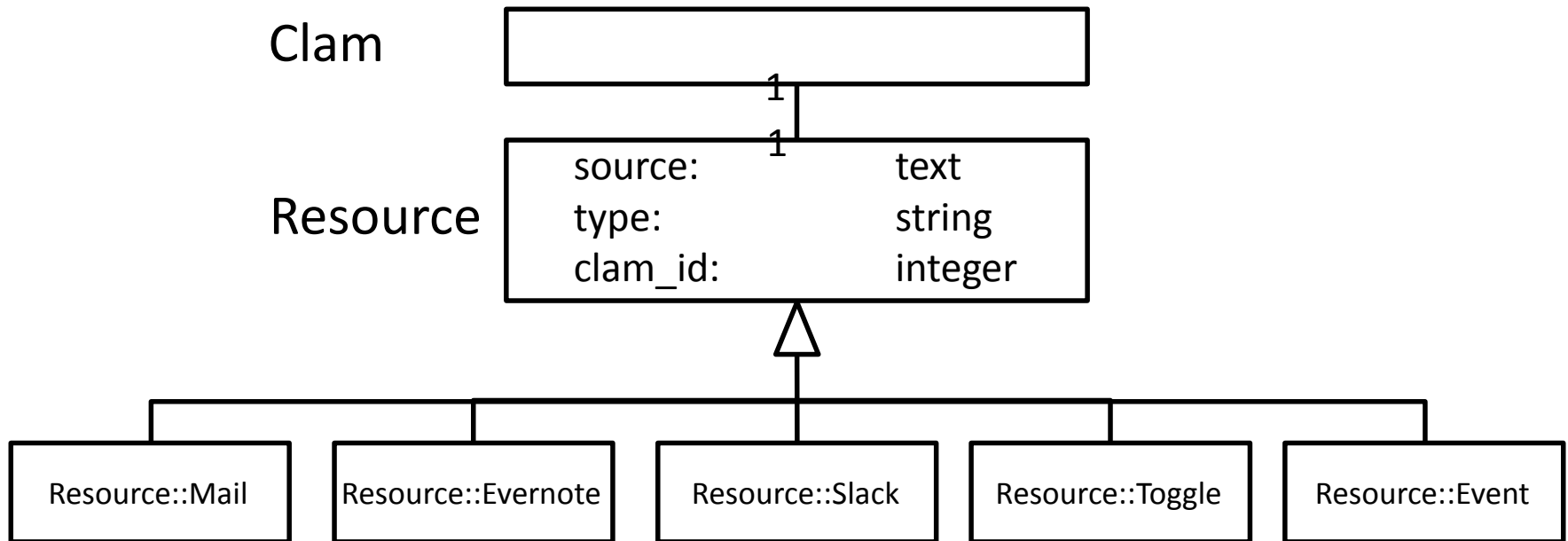
## Clam

uid:	string
date:	datetime
summary:	string
options:	text
content_type:	string
fixed:	boolean
mission_id:	integer

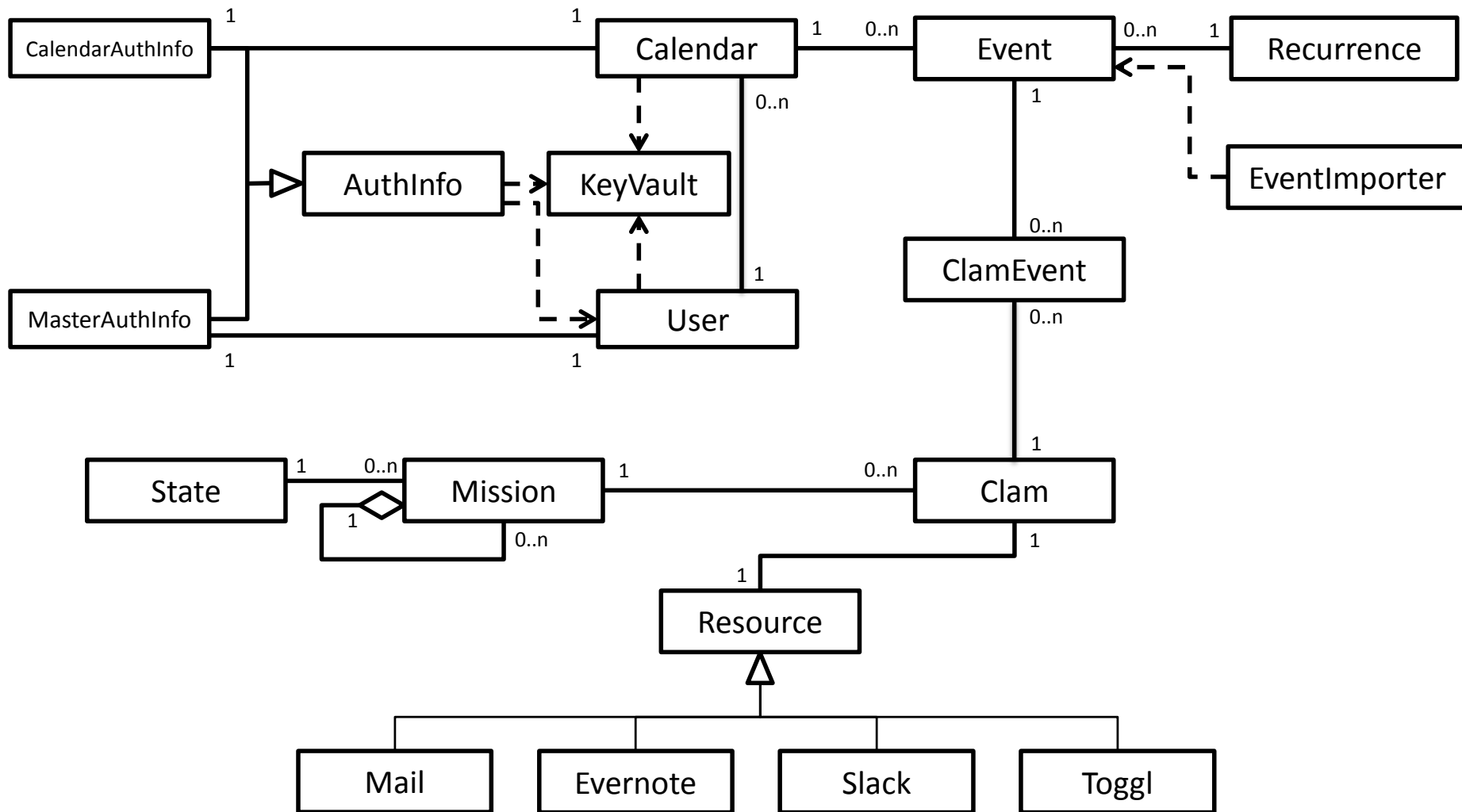


# Resource

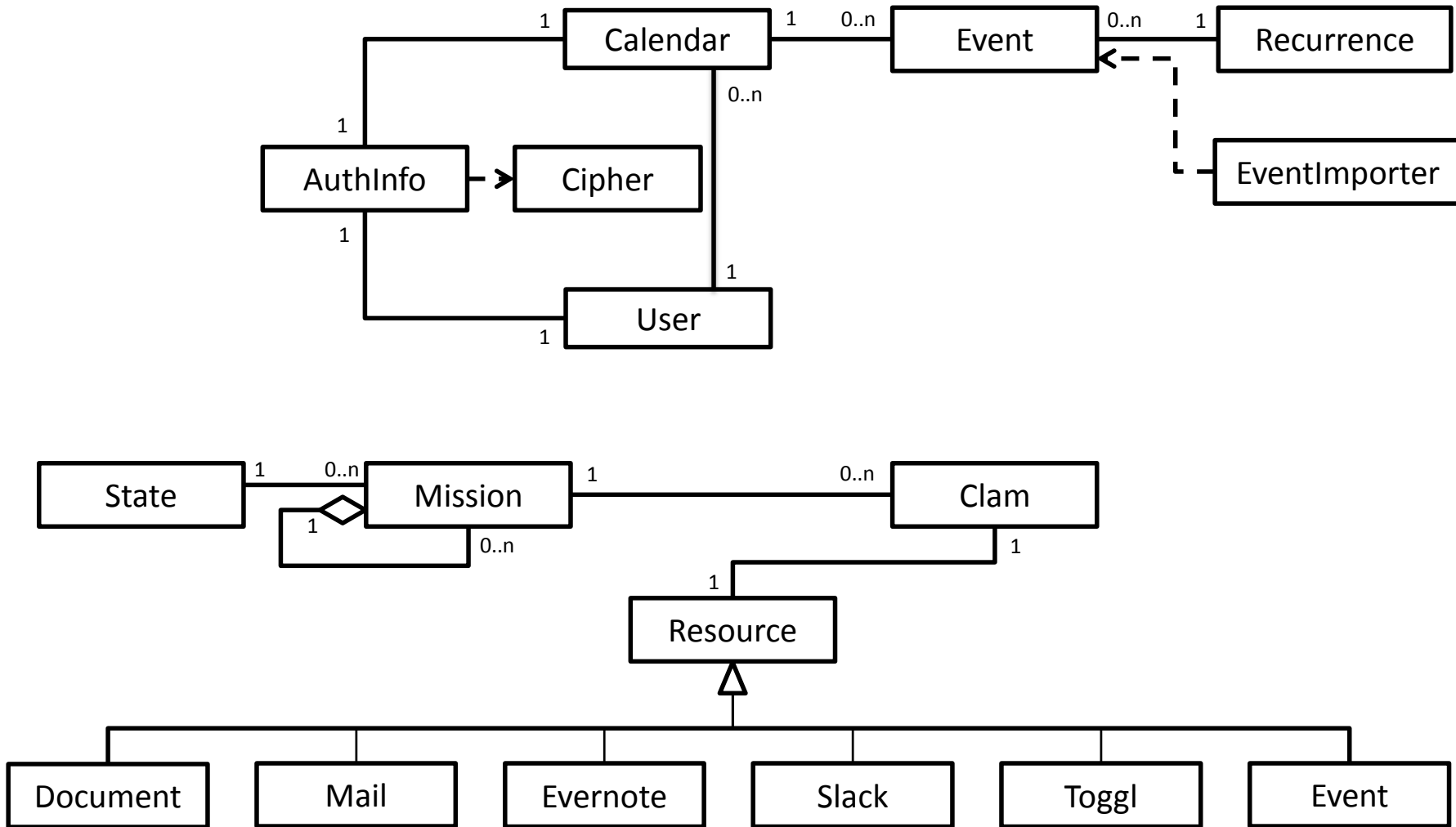
- 各アプリケーションから収集した情報の生データをもつ
- 各アプリケーション毎にResourceを継承したモデルが存在
- sourceに生データをstringで保存  
ex. {}



# クラス図 (現状)



# クラス図 (理想)



## 変更点

- MasterAuthInfoとCalendarAuthInfoを削除
- KeyVault → Cipher
- Cipherを使うのはAuthInfoのみ
  
- Missionはhas\_many Mission

## Clamを各アプリケーションから取り込む仕組み

- Clamは, 各アプリケーションで利用した情報
- [perican](#)を利用

perican

- gnで作成した CLI(command line interface)ツール
- 各アプリケーションから情報をGET
- GETした情報をClamの形に変換
- それをcamomeにPOSTする

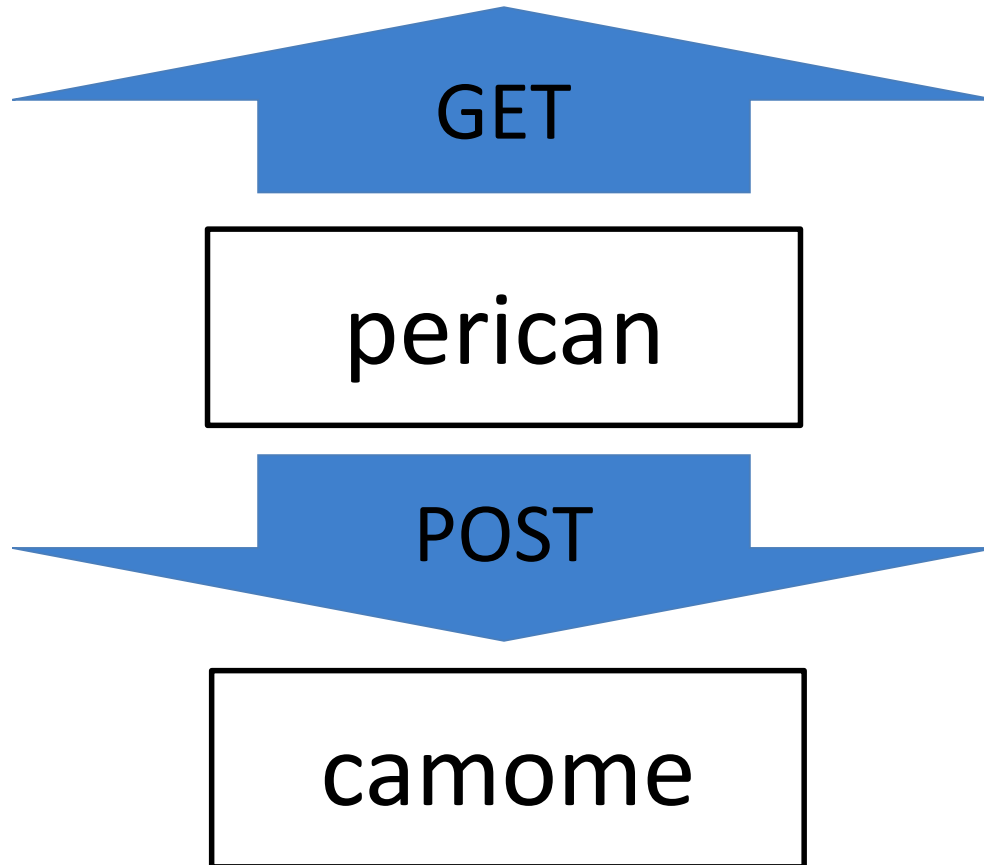
# Clamを各アプリケーションから取り込む仕組み



perican

camome

# Clamを各アプリケーションから取り込む仕組み



# Inbox(池田個人)バージョン



Fluentd



Inbox



## todo

- kobayashi/demo と nomlab/master を merge

### <Clamを収集する仕組みの実装>

- perican, fluentd, camomeの関係の検討
- ClamのPOSTを受け取る仕組みの実装

### <Inboxの実装>

- Mission, Taskを作成・管理 (like DTB ?)
- Mission, Taskに整理するInboxとRecurrenceに整理するInboxをどうするか
- Clamの整理

### <Calendarの実装>

- 予定の予測
- GoogleCalendarと同期
- 複数Calendarを作成できるインタフェース作成

予備

## nomlab/master

大きく分けて3つの機能を保持

カレンダー (/events/)

- 予定のCRUD
- 過去の予定を一覧表示&コピー
- Googleカレンダーと同期
- 1ユーザ1カレンダー (複数のカレンダーは持てるが, 作るインタフェースがない)

カレンダーの予定を整理するためのInbox /(inbox/recurrences)

- 予定をRecurrenceに整理
- 新しいRecurrence作成

メールを整理するためのInbox (/missions/inbox)

- メールと予定を関連付けて管理 (中間テーブルを介して)
- メールの再利用
- メールの読み書き

# nomlab/masterの理想形

## カレンダー (/events/)

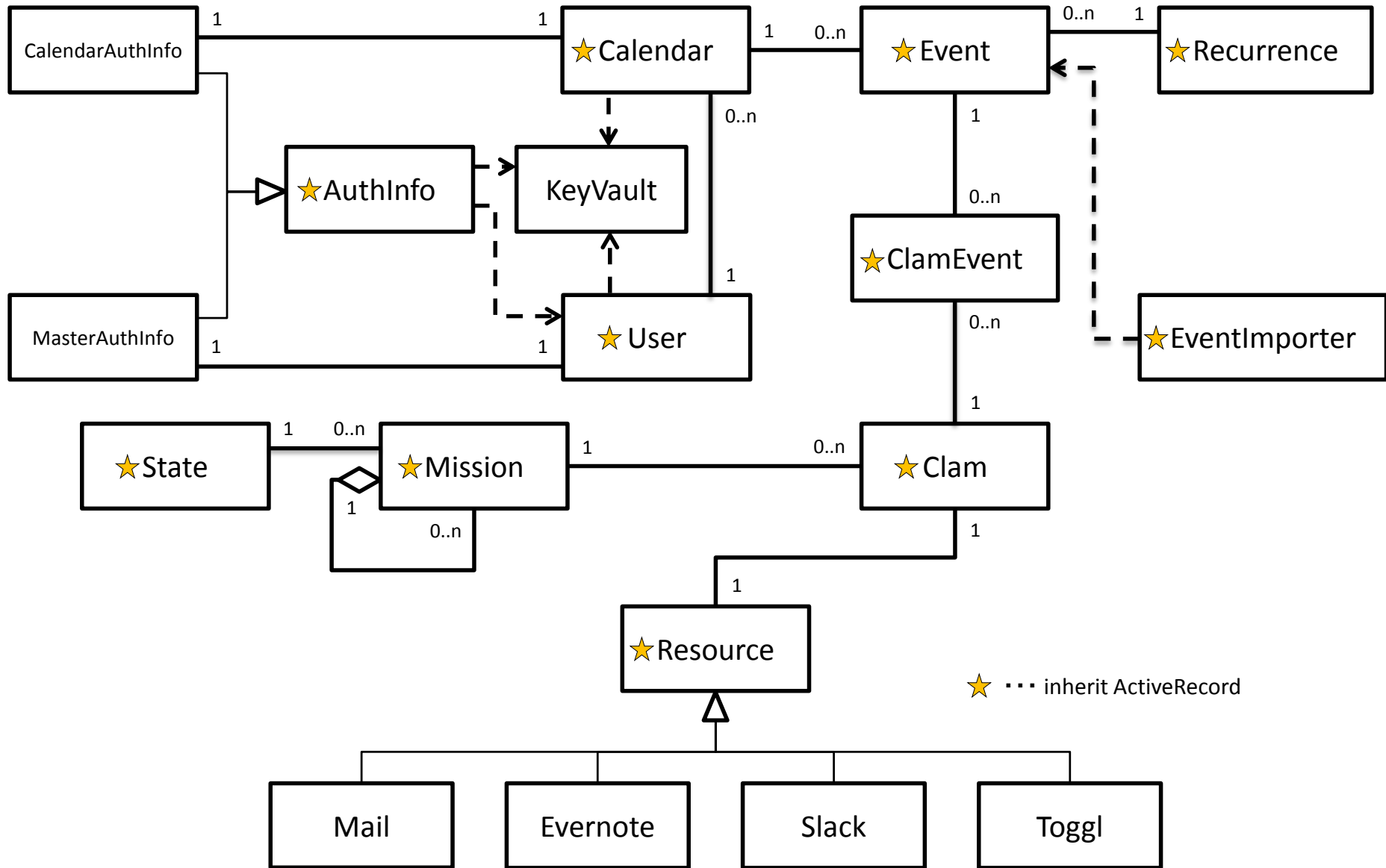
- Googleカレンダーと同期&Googleカレンダーライクなカレンダー機能
- 過去の予定を一覧表示&コピー
- 予定の予測

## アプリケーション情報を整理するInbox

- カレンダー, メール, 文書, Webブラウザ, チャットの会話をclamとして一覧表示
- Mission, Task, Recurrenceに関してそれぞれ整理
- MissionとTaskを階層構造で表現

## メールの管理

- メールと予定を関連付けて管理 (Missionを介して)
- メールの再利用
- メールの読み書き



nomlab/master (理想形)