

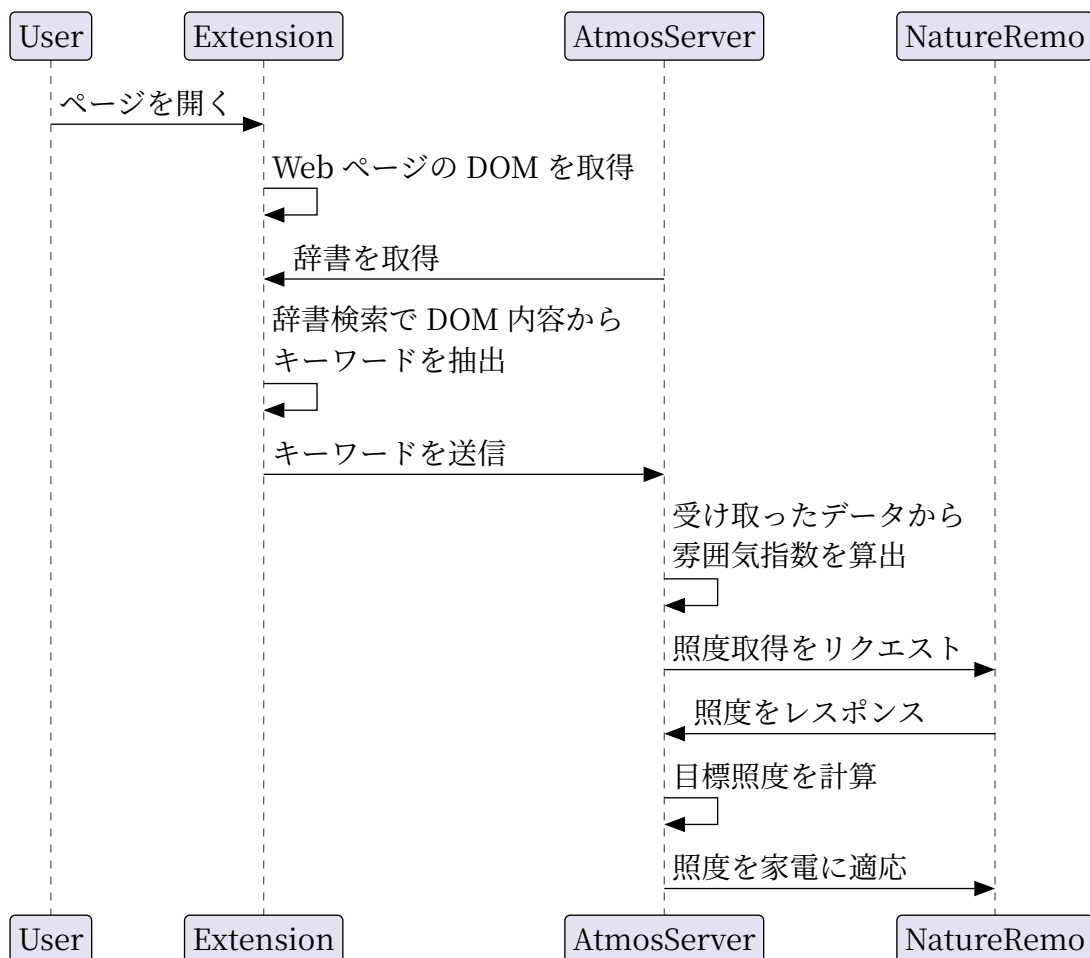
## 設計内容の概要

- 現在開いているページの内容を辞書検索し、特定のキーワードを抽出する。
  - DOM 操作でサイトのテキスト部分を取得し、辞書検索からフィルタリングする。
  - 辞書内容はサーバーに保存しておき、API で取得する。
- サーバーはキーワードを受け取り、サイトの種類によって計算方式を変えながら雰囲気指数を計算する。
  - YouTube はタグまたはタイトルをキーワードとする。
  - Netflix タイトルをキーワードとする。タイトルをネットで検索するなどの処理を行い、雰囲気のわかるタグを取得する
  - その他サイトは、辞書検索でヒットしたワードをキーワードとする。
- 雰囲気指数を照度に変換する。
- Nature Remo API を用いて照度を家電に適応する。
- パソコンの電源を落とした場合、現在の明るさを保持する。

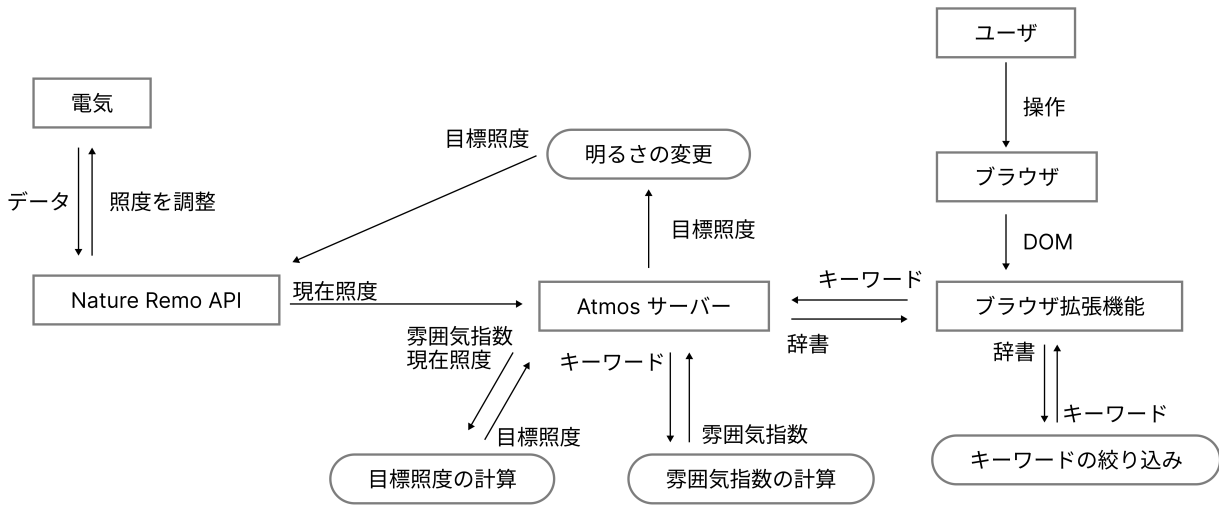
## システム処理の流れ

システム処理の流れを簡易的にモデル化したものを下に示す。

### シーケンス図



## データフロー図



## 必要なモジュール

Atmos サーバーはヘキサゴナルアーキテクチャを採用している。

- `atmos_server`
  - ▶ `domain.rs`
    - `models.rs`
      - `atmosfreq.rs`
      - `lighting.rs`
    - `ports.rs`:
      - `lighting.rs`
    - `service.rs`
      - `lighting.rs`
  - ▶ `inbound.rs`: API の定義
    - `http.rs`
      - `api.rs`
      - `routes.rs`
        - ▶ `adjust_lighting.rs`
        - ▶ `get_lighting_signals.rs`
  - ▶ `outbound.rs`: 外部とのやりとり
    - `remo.rs`
- `atmos_extension`
  - ▶ `popup/`
    - `App.svelte`: 設定の表示

- content.ts: DOM の操作