

Rubyを用いた台湾におけるインターネット政党政治の実践

2023.11.8

@RubyWorldConf





Teng Mu-Fan

CEO of 5xRuby CO., LTD

- 台湾のRuby伝道師
- 日本では竜堂 終と呼ばれています
- Chief Organizer of RubyConf Taiwan



2023
Dec 15 ~ 16

NATIONAL TAIPEI
UNIVERSITY OF EDUCATION

Register Now

▶ FOLLOW US ◀



SPEAKER



Yukihiro Matsumoto (Matz)



Cristian Planas



Ryo Kajiwara



RubyConf Taiwan 2023

- 2023/12/15~16 at 国立台北教育大学
- 台湾最大級のRubyカンファレンス
- MDD (Matz Driven Development) Conference
- <https://2023.rubyconf.tw>



AGENDA

1. 政党「時代力量（New Power Party）」の紹介及びオンライン参加の仕組み
2. システム開発経緯及び技術的負債の蓄積
3. リニューアルのプロセスと課題





時代力量

時代力量の紹介

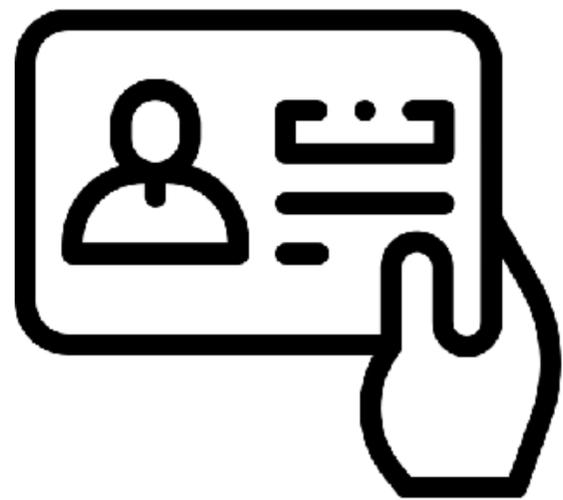
- 2015年に設立。
- <https://www.newpowerparty.tw/>
- 現在は台湾第4の政党で、3名の国会議員と6名の地方議会議員がいる。
- 台湾唯一インターネットで主な運営している政党。



時代力量ウェブサイトの主な機能



入党申込



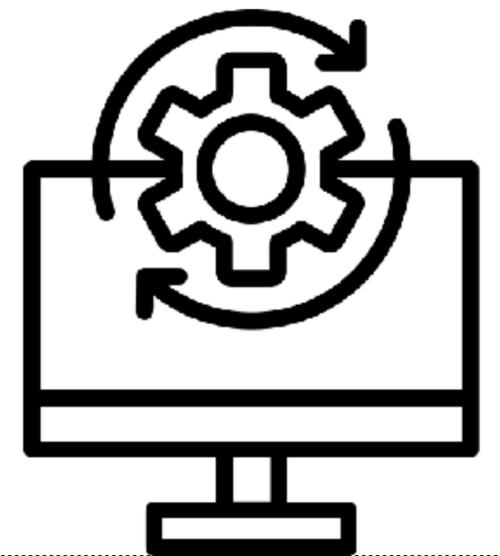
党費納入



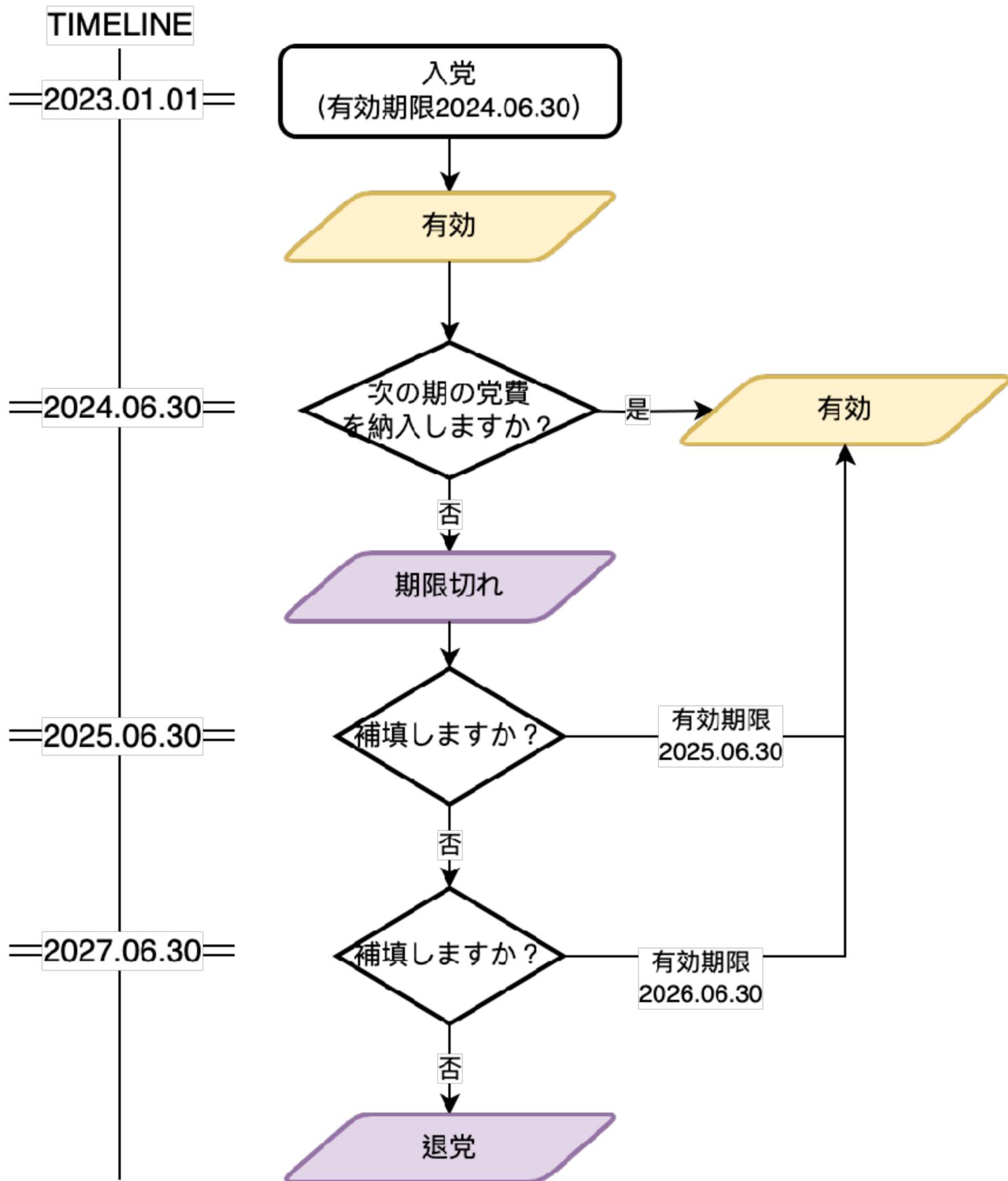
寄付



党内選挙



サイト運営



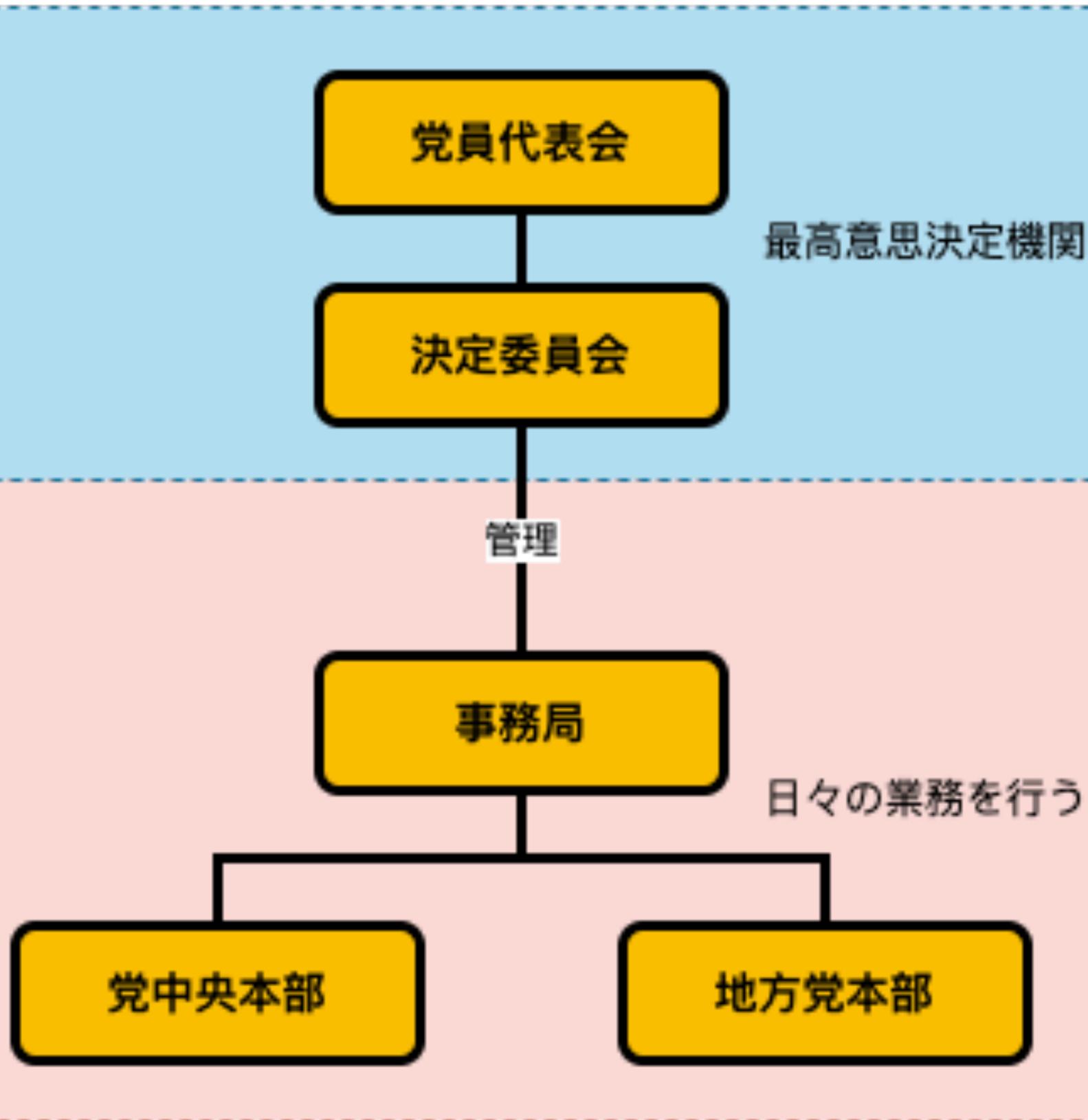
入党、会費期限計算とステータス変化

- 一年の最初の半年を「前期」、後の半年を「後期」とする。
- 期限日までに会費を納入しないと期限切れ(expired)ステータスになる。
- 期限切れの場合2年以内は補填できる。
- 2年を超えると退党(withdrawn)となり、補填でも党籍は戻らない。

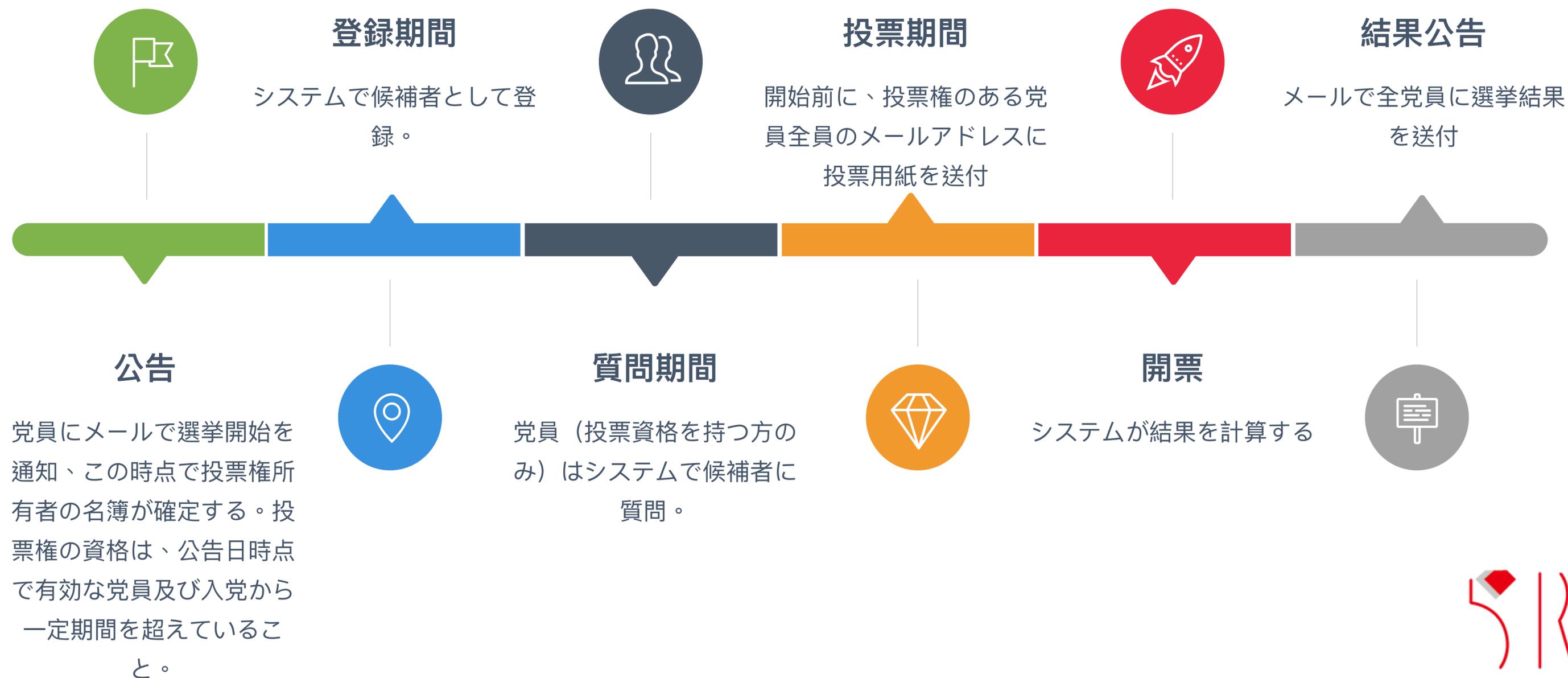


党内選挙

- 2つの最高意思決定機関委員を選出。
- 国会=立法院選挙の比例代表の候補者を党内投票で決める。



党内選挙のプロセス



旧システムの問題

- 一部の機能は開発後に継続利用できない
- ほとんど仕様書がなく、テストコードもないため技術的負債が巨大
- 人員交代時に十分な引き継ぎがない
- 担当者も全ての機能の使い方を把握していない
- ほぼ機能変更が不可能な状態

Application Stack の変更

	旧バージョン	新バージョン
Ruby	2.6	3.1
Rails	5.2	7.0
Database	PostgreSQL 9	PostgreSQL 14
Worker	Sidekiq 5 + Redis	GoodJob
CSS Framework	Bootstrap 3	TailwindCSS 3 + CSS Bundling
JS Framework	jQuery	Stimulus JS + Importmap
Deploy Tool	Capistrano	Docker Swarm



新バージョンの主な改善ポイントと課題

- 入党と党員ステータス関連機能の強化
- 投票システムの独立化
- 自動化CI/CDの導入
- データマイグレーションプロセス





入党と党員ステータス 関連機能の強化

入党プロセス



入党申請書提出



メールとSMSで認証



党費の支払い



資格審査

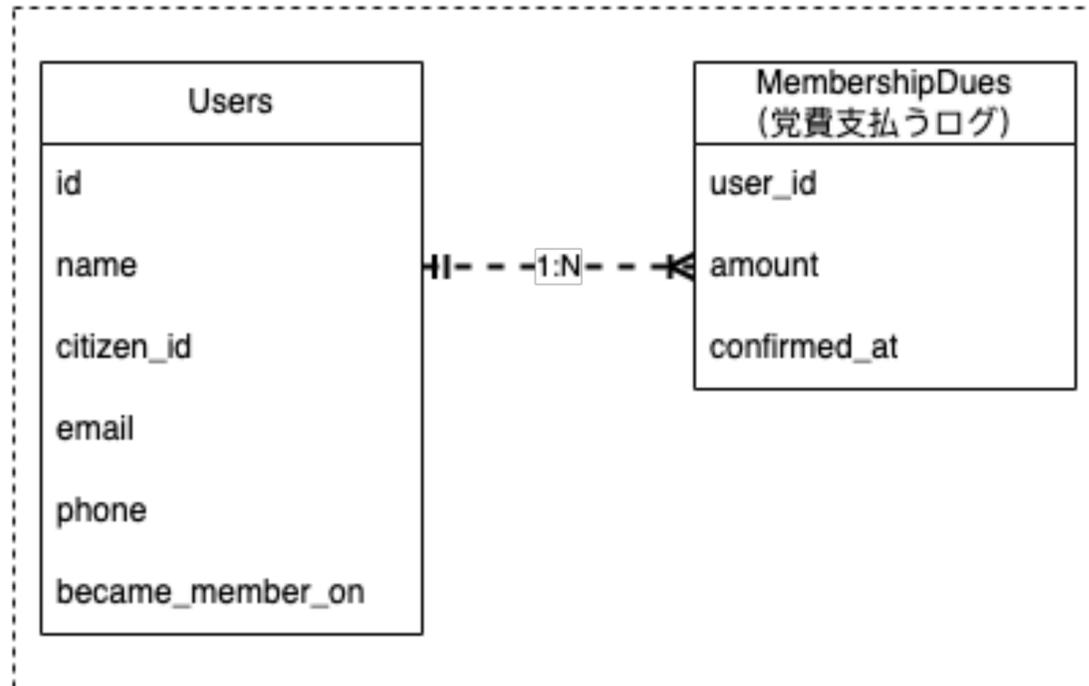


完成

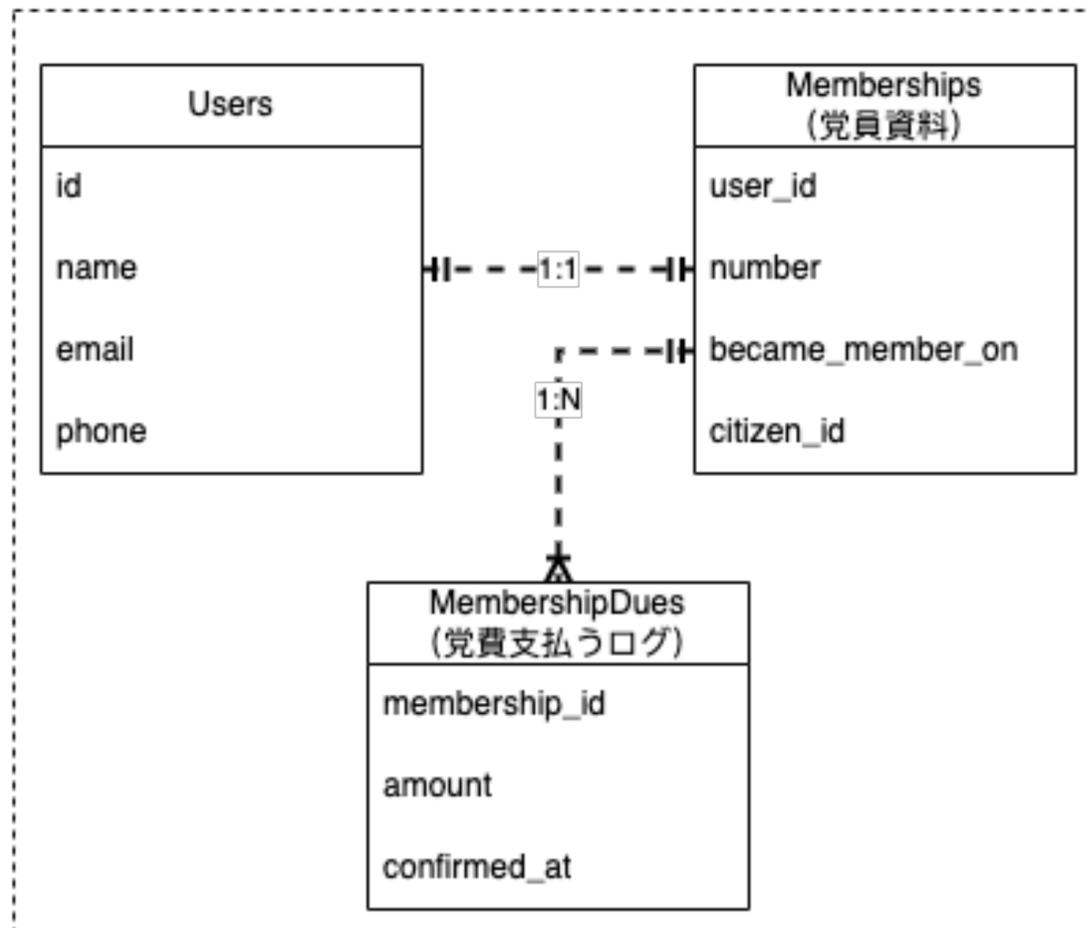
モデルの改善

- テーブルの正規化
- 非常に長かったメソッドを、より小さいメソッドに分割し、コード複雑度を下げる

BEFORE



AFTER



改善の結果 - コード複雑度

モデル	改善前	改善後	差分
User	1,212	190	-1,022
Membership	0	225	225
MembershipDue	369	157	-212
まとめ	1,581	572	-1,009

* コメントは含まれていません





党内選挙システムの 分離と再構築

旧バージョンの問題

1. 全てのバッチ処理（投票権所有者名簿の生成、投票用紙の生成と送信など）がエンジニアによる手動実行が必要
2. 投票にログイン不要
3. 投票データが暗号化されていない

改善—バッチ処理の自動化

- バッチ処理はActiveJob化
- `set_wait`を使ってタイミング実行
- データ作成や変更時に時間を再設定



改善—セキュリティの強化

- ActiveRecord Encryption を用いて投票データを暗号化
- 投票にログインは必要となります



再構築の最大の課題—新旧の互換性

- システム開発中に定期的に党内選挙も行うため、選挙システムを独立して構築する必要があった。
- 新旧ウェブサイトのログインを連携する必要がある。
- 新旧ウェブサイトでOAuth Authorization Server機能が必要。
- 選挙システムはウェブサイトのOAuthを利用してログインする。



All Files (81.58% covered at 89.03 hits/line)

39 files in total.

646 relevant lines, 527 lines covered and 119 lines missed. (81.58%)

Search:

File	% covered	Lines	Rele
app/components/election_register_button_component.rb	100.00 %	35	
app/jobs/application_job.rb	100.00 %	9	
app/jobs/ballot_production_job.rb	100.00 %	11	
app/jobs/send_bulletin_job.rb	100.00 %	11	
app/jobs/send_election_after_announcement_job.rb	100.00 %	11	
app/jobs/send_initial_election_announcement_job.rb	100.00 %	11	
app/jobs/set_election_voters_job.rb	100.00 %	11	
app/jobs/stop_election_job.rb	100.00 %	11	
app/mailers/application_mailer.rb	100.00 %	13	
app/mailers/ballot_mailer.rb	100.00 %	35	
app/mailers/candidate_question_mailer.rb	100.00 %	21	
app/mailers/election_mailer.rb	100.00 %	45	
app/models/application_record.rb	100.00 %	5	
app/models/ballot_vote.rb	100.00 %	43	
app/models/candidate_question_reply.rb	100.00 %	52	
app/models/concerns/election_schedule/after_announcement_sendable.rb	100.00 %	24	
app/models/concerns/election_schedule/bulletin_senable.rb	100.00 %	27	
app/models/concerns/election_schedule/initial_announcement_sendable.rb	100.00 %	28	
app/models/concerns/election_schedule/stoppable.rb	100.00 %	47	
app/models/concerns/election_schedule/voter_registrable.rb	100.00 %	33	
app/models/election_announcement.rb	100.00 %	66	
app/models/election_voter.rb	100.00 %	28	

改善の結果 - 信頼性

- テストカバレッジを0から80%以上に
- 2人のチームで1ヶ月で開発





自動化CI/CDの導入 とレビューアプリ

CI/CD 導入の目標

- 自動化されたコード品質検査、テストの実行により、開発プロセスを加速し、製品品質を向上させる。
- 各プルリクに対して自動的にサンドボックス環境を生成することで、関係者が開発結果を実際にテストできる。

自動化CI/CD の導入手順

- RubyGemsを使って Dockerize
 - boxing: DSLでDockerfileを生成
 - openbox: Docker Entrypointを生成
 - liveness: ヘルスチェックエンドポイントを生成
- Docker Composeファイルでデプロイ設定を行う
- GitLab CIのテンプレートをベースにCIパイプラインを構築



admin periodic_order add default_sorting

Delete

Passed SSU-CHI-YEH created pipeline for commit 287e3190 finished 2 weeks ago

Related merge request 1365 to merge feat/5587-admin-periodic-notifications-add-default-sorting

latest merge request 12 Jobs 0 18 minutes 18 seconds, queued for 2 seconds

Pipeline Needs Jobs 12 Tests 0

Group jobs by Stage Job dependencies

prepare	lint	test	compile	build	deploy
<p>✓ rubygems</p>	<p>✓ brakeman</p> <p>✓ bundler-audit</p> <p>✓ bundler-leak</p> <p>✓ rubocop</p>	<p>✓ rspec</p>	<p>✓ assets:precompile</p>	<p>✓ docker</p>	<p>✓ review:create_bucket</p> <p>✓ review:deploy</p> <p>✓ review:drop_bucket</p> <p>✓ review:stop</p>

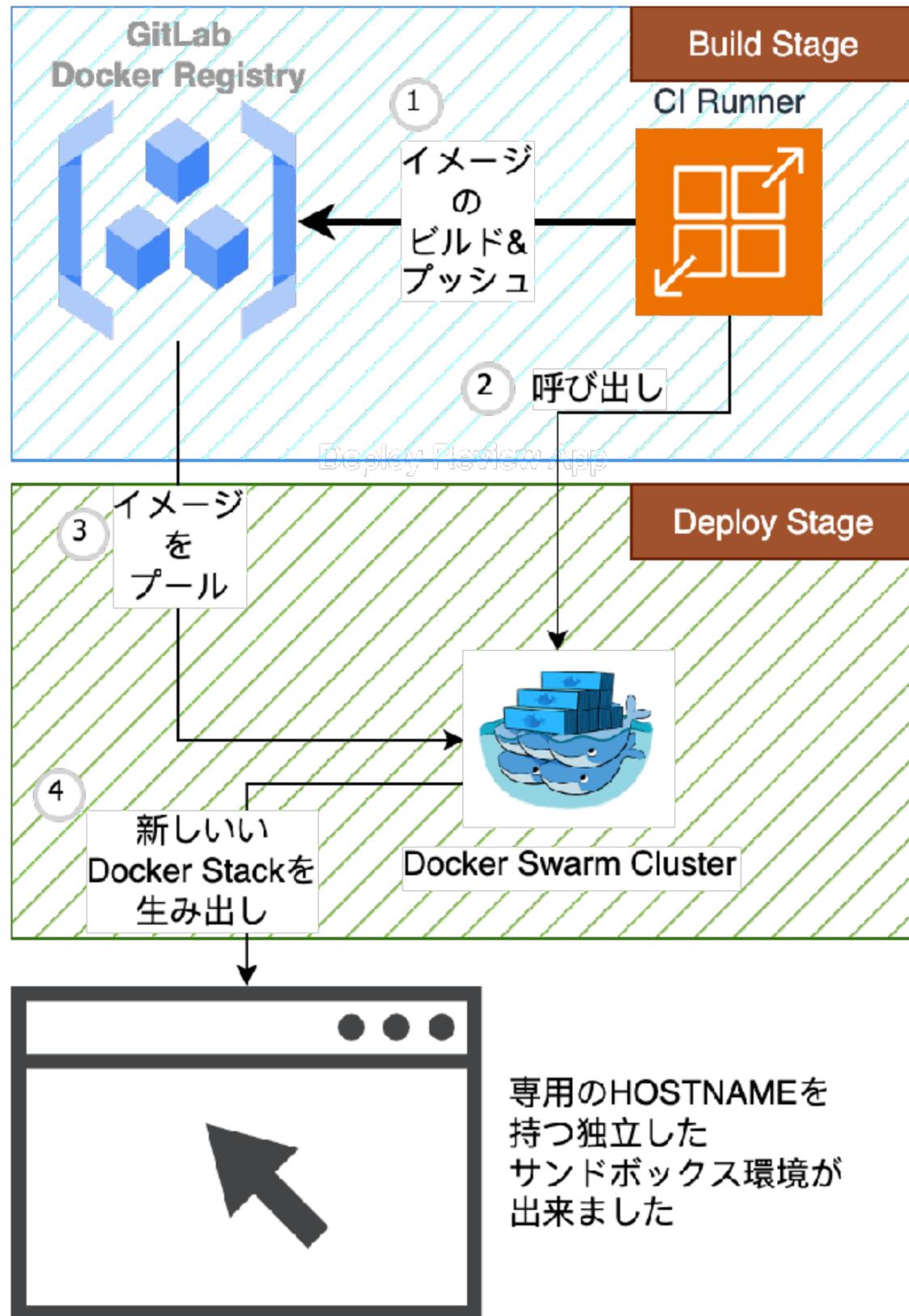
セキュリティと
Syntax Styleのチェック

テストを執行

イメージのビルド&プッシュ

デプロイ





Review Apps

(レビューアプリ)

- プルリクごとにその修正内容のみを反映した環境を立ててデプロイしてくれる便利機能
- Docker Swarm + Traefikをランタイム環境として利用
- マージ後は、CIジョブにより自動的に削除されます。





データマイグレーション

Railsエンジンの利用

```
150 gem 'image_processing'~
151 gem 'mini_magick', '~> 4.8'~
152 ~
153 gem 'administrate-bootstrap-theme', '~> 1.0'~
154 gem 'validate_url'~
155 ~
156 gem 'maintenance_tasks', '~> 2.0'~
157 ~
158 gem 'administrate-field-i18n_enum'~
159 ~
160 gem 'good_job', '~> 3.12'~
161 ~
162 gem 'twsms2', '~> 1.3'~
163 ~
164 gem 'doorkeeper'~
165 ~
166 gem 'administrate_ransack'~
167 ~
168 gem 'caxlsx', '~> 3.4', '>= 3.4.1'~
169 ~
170 gem 'caxlsx_rails', '~> 0.6.3'~
171 ~
172 gem 'legacy', git: 'https://gitlab.com/newpowerparty/legacy_migrator.git', branch: 'main'~
173 ~
174 gem 'scenic', '~> 1.7'~
175 ~
```

- マイグレーションに含まれるもの：
 - データマイグレーション
 - CarrierWaveからActive Storageへの添付ファイル変換
- マイグレーションロジックをRailsエンジンでカプセル化



Dockerfile

```
ARG APP_ROOT=/src/app
ARG RUBY_VERSION=3.1.3

FROM ruby:${RUBY_VERSION}-alpine AS gem
ARG APP_ROOT
ARG GITLAB_TOKEN_NAME
ARG GITLAB_TOKEN_TOKEN

RUN apk add --no-cache build-base postgresql-dev gcompat
git

RUN mkdir -p ${APP_ROOT}
COPY Gemfile Gemfile.lock ${APP_ROOT}/

ENV BUNDLE_GITLAB_COM=${GITLAB_TOKEN_NAME}:${GITLAB_TOKEN_TOKEN}
```

CI Script

```
docker:
  needs:
    - job: assets:precompile
      artifacts: true
  script:
    - docker buildx build
      --cache-from type=local,src=tmp/docker
      --cache-to type=local,dest=tmp/docker
      --build-arg GITLAB_TOKEN_NAME=${GITLAB_TOKEN_NAME}
      --build-arg GITLAB_TOKEN_TOKEN=${GITLAB_TOKEN_TOKEN}
      --build-arg REVISION=${REVISION}
      ${DOCKER_TAG_OPTIONS}
      --push .
```

GemfileにGitトークンを直接記載しない

- DockerのBuild Argumentsを利用する
- Dockerイメージをビルド時に、CI環境変数をBuild Argumentに渡す



稼働当日

- 実際のマイグレーションは約30分
- データ検証含めて約2時間
- 党職員による静的データ入力に約2日



ご清聴ありがとうございました

Special Thanks to @koic & @h_ryo_12 for Proofreading