



# コードは書くな、フローを語れ！ プロンプト指向プログラミングで始める 業務オーケストレーションの現在地

2026/06/27

ドコモイノベーションズ | 直井康広 (Yas Naoi)



DrupalCamp  
Tokyo 2026

# 直井康広 (Yas Naoi | @yas)



- Drupal 歴 20年 (日本人初のコミッター)
- ドコモイノベーションズ (シリコンバレー)
- チーフクラウドアーキテクト
- AWS 認定資格全冠
- クラウド / DevOps / アジャイル開発 / Drupal
- 好き : 作曲 / 読書 / ハイキング / スノボ / 株



# 承認システム移行プロジェクト



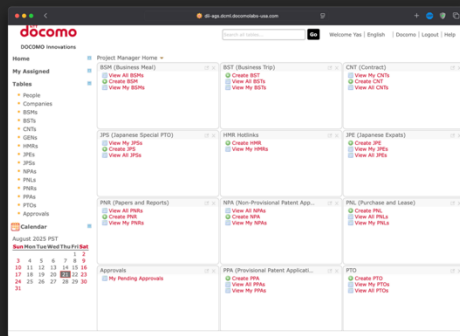
# 承認システム移行プロジェクト



A screenshot of the Docomo Innovations Project Manager Home dashboard. The dashboard is displayed in a web browser window with the URL "dli-ags.dcmi.docomolabs-usa.com". The page has a dark header with the "docomo" logo and navigation links: "Search all tables...", "Go", "Welcome Yas", "English", "Docomo", "Logout", and "Help". The main content area is divided into several sections. On the left, there is a sidebar with "Home" and "My Assigned" sections. Under "My Assigned", there is a "Tables" section with a list of tables: People, Companies, BSMs, BSTs, CNTs, GENs, HMRS, JPEs, JPSs, NPAs, PNLs, PNRs, PPAs, PTOs, and Approvals. Below the sidebar is a "Calendar" section showing a calendar for August 2025 PST, with the 24th highlighted. The main content area is a grid of project cards. Each card has a title, a "Create" button, and a "View My" button. The cards are: BSM (Business Meal), BST (Business Trip), CNT (Contract), JPS (Japanese Special PTO), HMR Hotlinks, JPE (Japanese Expats), PNR (Papers and Reports), NPA (Non-Provisional Patent App...), PNL (Purchase and Lease), Approvals, PPA (Provisional Patent Applicati...), and PTO. Each card also has a "View All" button.



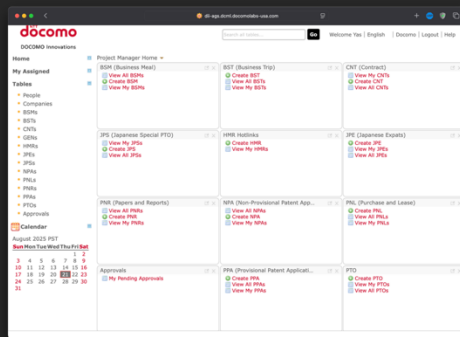
- 



# 承認システム移行プロジェクト



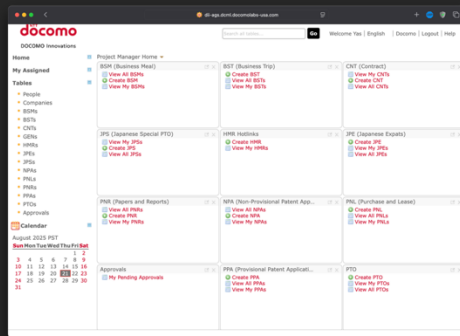
- 社内承認システム「AGS」を新プラットフォームへ移行
- 目的：単なる置き換えではなく、生成 AI を取り入れたイノベーション
- 効率化と柔軟な仕様変更対応



# 承認システム移行プロジェクト



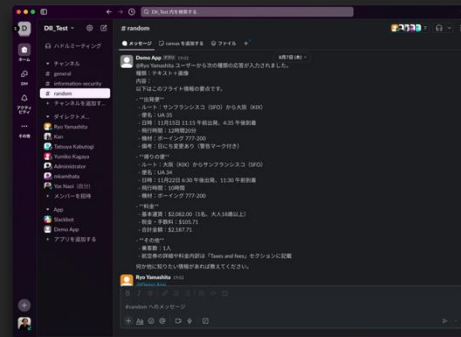
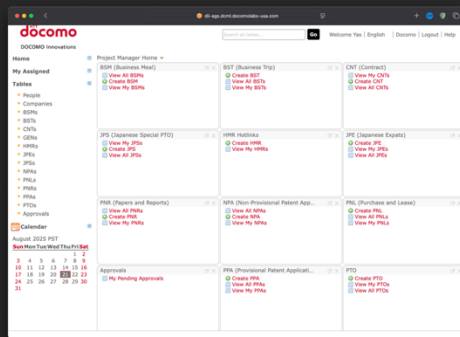
- 社内承認システム「AGS」を新プラットフォームへ移行
- 目的：単なる置き換えではなく、**生成 AI を取り入れたイノベーション**
- 効率化と柔軟な仕様変更対応



# 承認システム移行プロジェクト



- 社内承認システム「AGS」を新プラットフォームへ移行
- 目的：単なる置き換えではなく、生成 AI を取り入れたイノベーション
- 効率化と柔軟な仕様変更対応





# 初期開発（PoC）を通して得られた知見



# 初期開発（PoC）を通して得られた知見



- 添付ファイルや画像などマルチモーダル入力処理
- LLM が抽出や検証を担当
- 従来型の if 文で書かれていたコードはプロンプトで表現可能なのでは？

# 初期開発（PoC）を通して得られた知見



- 添付ファイルや画像などマルチモーダル入力処理
- LLM が抽出や検証を担当
- 従来型の if 文で書かれていたコードはプロンプトで表現可能なのでは？

# 初期開発（PoC）を通して得られた知見



- 添付ファイルや画像などマルチモーダル入力进行处理
- LLM が抽出や検証を担当
- 従来型の if 文で書かれていたコードはプロンプトで表現可能なのでは？

# コード中心 vs YAML ファースト



# コード中心 vs YAML ファースト



```
# Expense approval (traditional)
import re
from datetime import datetime, timedelta

ALLOWED_DEPTS = ("Sales", "R&D", "Admin")
AUTO_LIMIT = 1000

def approve_expense(req):
    # parse amount from free text
    m = re.search(r'(\d+(?:\.\d{1,2})?)', req["amount_text"])
    if not m:
        return {"ok": False, "error": "Invalid amount"}
    amount = float(m.group(1))

    # date within 30 days
    date = datetime.strptime(req["date"], "%Y-%m-%d")
    if datetime.now() - date > timedelta(days=30):
        return {"ok": False, "error": "Receipt too old"}

    # department whitelist
    if req["department"] not in ALLOWED_DEPTS:
        return {"ok": False, "error": "Unknown department"}

    # routing
    approver = "Auto" if amount <= AUTO_LIMIT else "Manager"

    return {
        "ok": True,
        "amount": amount,
        "date": date.strftime("%Y-%m-%d"),
        "department": req["department"],
        "approver": approver,
    }
```

```
blueprint: expense_approval_v1
inputs:
  amount:
    source: ocr(amount_text)
    parse: "number(currency: true, decimals:2)"
  date:
    source: user_input
    format: "%Y-%m-%d"
  department:
    source: user_input

validators:
  - field: date
    rule: within_days
    value: 30
    message_id: msg.receipt_too_old
  - field: department
    rule: one_of
    value: ["Sales", "R&D", "Admin"]
    message_id: msg.invalid_dept

routing:
  approver:
    when: "amount > 1000"
    then: "Manager"
    else: "Auto"

output_schema:
  type: object
  required: [amount, date, department, approver]
  properties:
    amount: {type: number}
    date: {type: string, format: date}
    department: {type: string}
    approver: {type: string}
```

# コード中心 vs YAML ファースト



```
# Expense approval (traditional)
import re
from datetime import datetime, timedelta

ALLOWED_DEPTS = ("Sales", "R&D", "Admin")
AUTO_LIMIT = 1000

def approve_expense(req):
    # parse amount from free text
    m = re.search(r'(\d+(?:\.\d{1,2})?)', req["amount_text"])
    if not m:
        return {"ok": False, "error": "Invalid amount"}
    amount = float(m.group(1))

    # date within 30 days
    date = datetime.strptime(req["date"], "%Y-%m-%d")
    if datetime.now() - date > timedelta(days=30):
        return {"ok": False, "error": "Receipt too old"}

    # department whitelist
    if req["department"] not in ALLOWED_DEPTS:
        return {"ok": False, "error": "Unknown department"}

    # routing
    approver = "Auto" if amount <= AUTO_LIMIT else "Manager"

    return {
        "ok": True,
        "amount": amount,
        "date": date.strftime("%Y-%m-%d"),
        "department": req["department"],
        "approver": approver,
    }
```

```
blueprint: expense_approval_v1
inputs:
  amount:
    source: ocr(amount_text)
    parse: "number(currency: true, decimals:2)"
  date:
    source: user_input
    format: "%Y-%m-%d"
  department:
    source: user_input

validators:
  - field: date
    rule: within_days
    value: 30
    message_id: msg.receipt_too_old
  - field: department
    rule: one_of
    value: ["Sales", "R&D", "Admin"]
    message_id: msg.invalid_dept

routing:
  approver:
    when: "amount > 1000"
    then: "Manager"
    else: "Auto"

output_schema:
  type: object
  required: [amount, date, department, approver]
  properties:
    amount: {type: number}
    date: {type: string, format: date}
    department: {type: string}
    approver: {type: string}
```

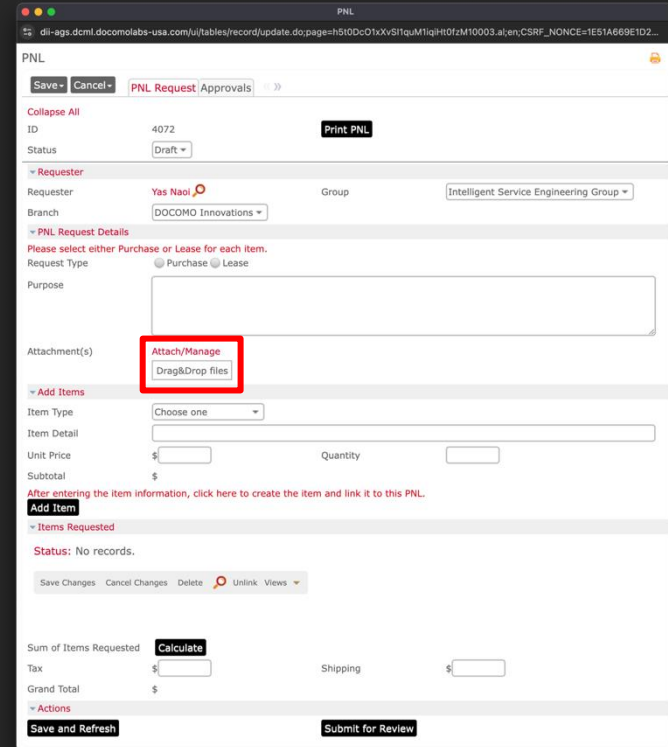
# 承認申請の例





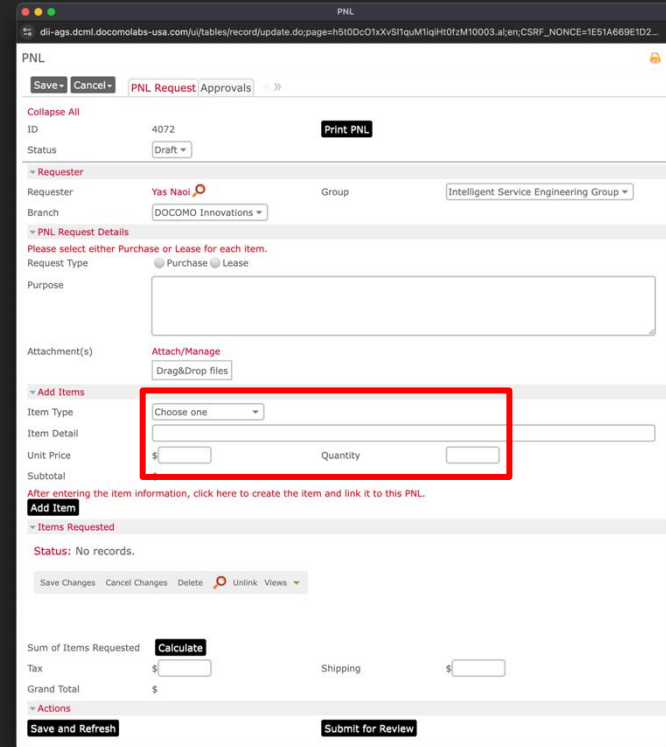
# 承認申請の例

- 入力：添付ファイル（PDF/画像）
- LLM が名称・金額・税・送料を抽出
- YAML Validator がフィールド検証
- JSON 保存 → 人による最終確認



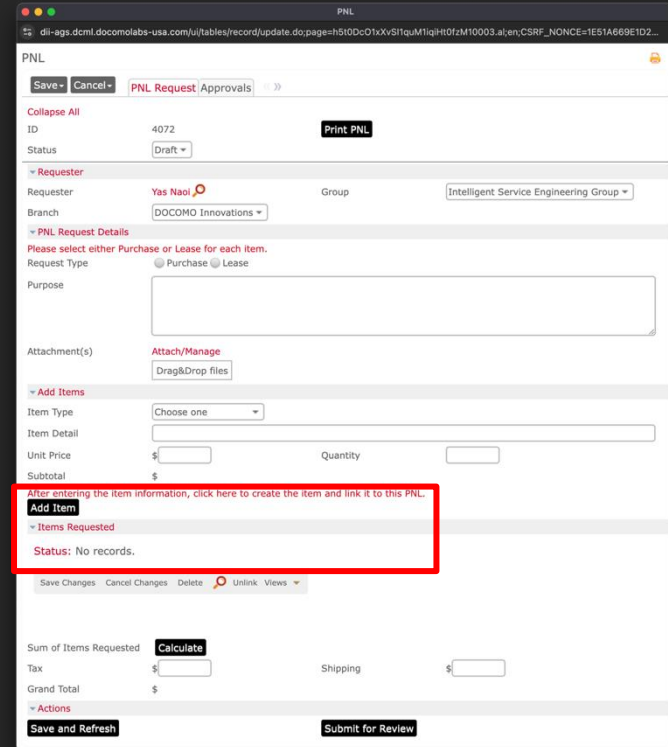
# 承認申請の例

- 入力：添付ファイル（PDF/画像）
- LLM が名称・金額・税・送料を抽出
- YAML Validator がフィールド検証
- JSON 保存 → 人による最終確認



# 承認申請の例

- 入力：添付ファイル（PDF/画像）
- LLM が名称・金額・税・送料を抽出
- **YAML Validator がフィールド検証**
- JSON 保存 → 人による最終確認



PNL

Save Cancel PNL Request Approvals

Collapse All

ID 4072 Print PNL

Status Draft

Requester Yas Naoi Group Intelligent Service Engineering Group

Branch DOCOMO Innovations

PNL Request Details

Please select either Purchase or Lease for each item.

Request Type ☒ Purchase ☐ Lease

Purpose

Attachment(s) Attach/Manage Drag&Drop files

Add Items

Item Type Choose one

Item Detail

Unit Price \$ Quantity

Subtotal \$

After entering the item information, click here to create the item and link it to this PNL.

Add Item

Items Requested

Status: No records.

Save Changes Cancel Changes Delete Unlink Views

Sum of Items Requested Calculate

Tax \$ Shipping \$

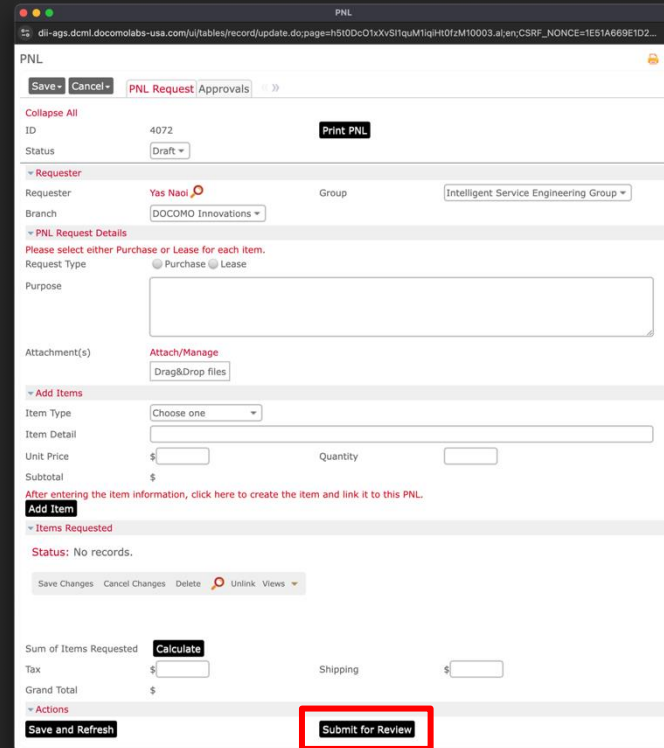
Grand Total \$

Actions

Save and Refresh Submit for Review

# 承認申請の例

- 入力：添付ファイル（PDF/画像）
- LLM が名称・金額・税・送料を抽出
- YAML Validator がフィールド検証
- JSON 保存 → 人による最終確認



PNL

Save Cancel PNL Request Approvals

Collapse All

ID 4072 Print PNL

Status Draft

Requester Yas Nao Group Intelligent Service Engineering Group

Branch DOCOMO Innovations

PNL Request Details

Please select either Purchase or Lease for each item.

Request Type ☒ Purchase ☐ Lease

Purpose

Attachment(s) Attach/Manage Drag&Drop files

Add Items

Item Type Choose one

Item Detail

Unit Price \$ Quantity

Subtotal \$

After entering the item information, click here to create the item and link it to this PNL.

Add Item

Items Requested

Status: No records.

Save Changes Cancel Changes Delete Unlink Views

Sum of Items Requested Calculate

Tax \$ Shipping \$

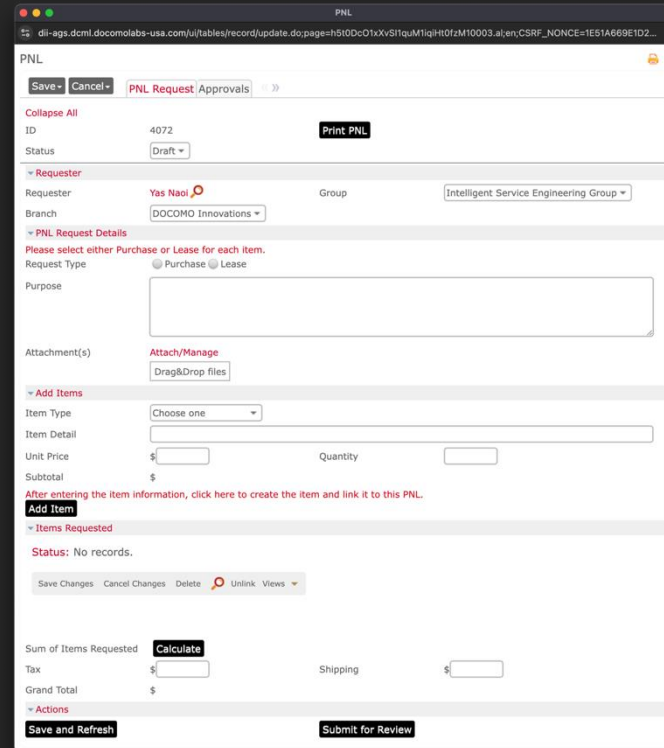
Grand Total \$

Actions

Save and Refresh Submit for Review

# 承認申請の例

- 入力：添付ファイル（PDF/画像）
- LLM が名称・金額・税・送料を抽出
- YAML Validator がフィールド検証
- JSON 保存 → 人による最終確認



PNL

Save Cancel PNL Request Approvals

Collapse All

ID 4072 Print PNL

Status Draft

Requester Yas Nao Group Intelligent Service Engineering Group

Branch DOCOMO Innovations

PNL Request Details

Please select either Purchase or Lease for each item.

Request Type ☒ Purchase ☐ Lease

Purpose

Attachment(s) Attach/Manage Drag&Drop files

Add Items

Item Type Choose one

Item Detail

Unit Price \$ Quantity

Subtotal \$

After entering the item information, click here to create the item and link it to this PNL.

Add Item

Items Requested

Status: No records.

Save Changes Cancel Changes Delete Unlink Views

Sum of Items Requested Calculate

Tax \$ Shipping \$

Grand Total \$

Actions

Save and Refresh Submit for Review



# Raxpense

← →

🔍 探している内容について説明する

👤

?

RQ

Raxpen...

⚙️📄

🏠

ホーム

📅 トライアル期間  
はあと 15 日...

📧 DM

📁 アクティ  
ビティ

📁 ファイル

📖 後で

⋮

その他

★🔗 Slack Raxpense bot

🔔🔗

🏠

ホーム

💬

メッセージ

📋

概要

Request history

Your recent requests are listed below with their submission state and summary.

🟢 Start new request

🟡 Pending approvals

Page: 1

● Approval request ID: [PCH-2026-0394](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/394>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0393](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/393>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0392](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/392>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0391](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/391>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0390](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/390>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0389](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/389>

🔄 Re-submit

📄 Copy request

📧 スレッド

📅 ハドルミーティング

📄 下書き & 送信済み

📁 ディレクトリ

☆ スター付き

👤 Slack Raxpense bot

📧 チャンネル

# all-raxpense-qa

# new-channel

# social

+ チャンネルを追加...

📧 ダイレクトメッセージ

👤 Rax\_Yamashita02

👤 Rax\_Yamashita01

👤 actual\_ahoang

👤 Amy Hoang (DII)

👤 Yas Naoi (DII)

👤 actual\_shota\_shiino

👤 actual\_kazuoki\_ichi...

👤 actual\_michael\_co...

👤 actual\_mark\_mavr...

👤 actual\_nobutaka\_...

👤 actual\_yas\_naoi (...)

+ メンバーを招待

☰ App

← →

🔍 探している内容について説明する

👤

?

RQ

Raxpen...

⚙️📄

🏠

ホーム

📅 トライアル期間  
はあと 15 日...

📧 DM

📁 アクティ  
ビティ

📁 ファイル

📖 後で

⋮

その他

★🔗 Slack Raxpense bot

🔔🔗

🏠

ホーム

💬

メッセージ

📋

概要

Request history

Your recent requests are listed below with their submission state and summary.

🟢 Start new request

🟡 Pending approvals

Page: 1

● Approval request ID: [SCS-2026-0153](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/153>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PPA-2026-0152](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/152>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PNR-2026-0151](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/151>

🔄 Re-submit

📄 Copy request

● Approval request ID: [PCH-2026-0150](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/150>

🔄 Re-submit

📄 Copy request

● Approval request ID: [NDA-2026-0149](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/149>

🔄 Re-submit

📄 Copy request

● Approval request ID: [JPS-2026-0148](#)

● Approval state: [Approved](#)

● Link: <https://ga-dii-raxpense.docomoinnovations.net/node/148>

🔄 Re-submit

📄 Copy request

📧 スレッド

📅 ハドルミーティング

📄 下書き & 送信済み

📁 ディレクトリ

☆ スター付き

👤 Slack Raxpense bot

📧 チャンネル

# all-raxpense-qa

# new-channel

# social

# チャンネルを追加...

📧 ダイレクトメッセージ

👤 Rax\_Yamashita02

👤 Rax\_Yamashita01

👤 actual\_ahoang

👤 Amy Hoang (DII)

👤 Yas Naoi (DII)

👤 actual\_shota\_shiino

👤 actual\_kazuoki\_ichi...

👤 actual\_michael\_co...

👤 actual\_mark\_mavr...

👤 actual\_nobutaka\_...

👤 actual\_tetsuya\_nak...

📄 全員をブラウズする

☰ App

# Raxpense



- ビジネスロジックの実装は従来のプログラミングのコードではなく、
- YAML ベースのフレームワークと生成 AI の組み合わせで動作



# Raxpense



- ビジネスロジックの実装は従来のプログラミングのコードではなく、
- **YAML ベースのフレームワーク**と生成 AI の組み合わせで動作

# Raxpense

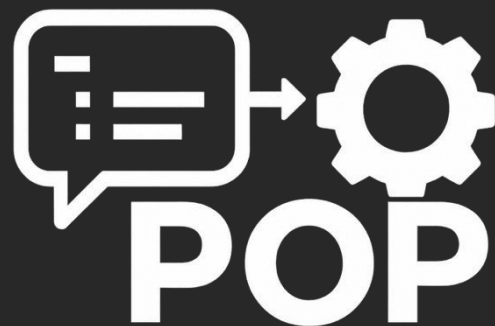


- ビジネスロジックの実装は従来のプログラミングのコードではなく、
- YAML ベースのフレームワークと生成 AI の組み合わせで動作



# 新提案

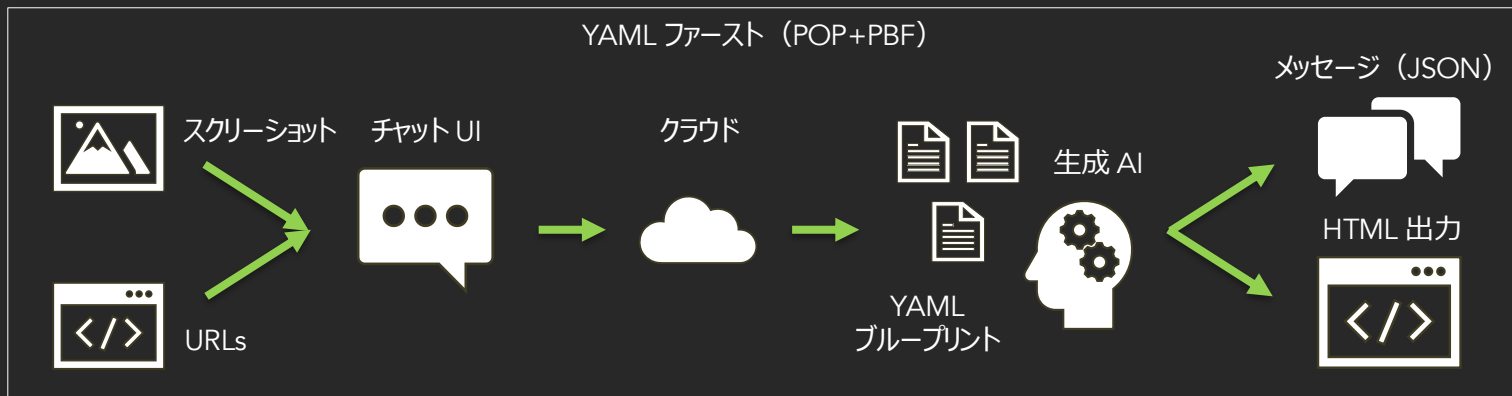
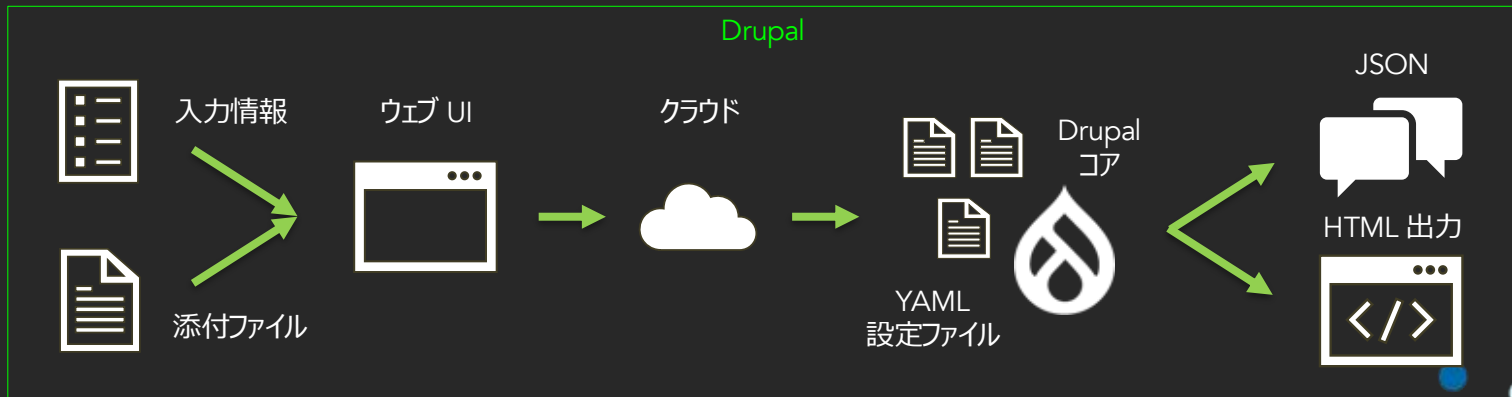
# プロンプト指向プログラミング



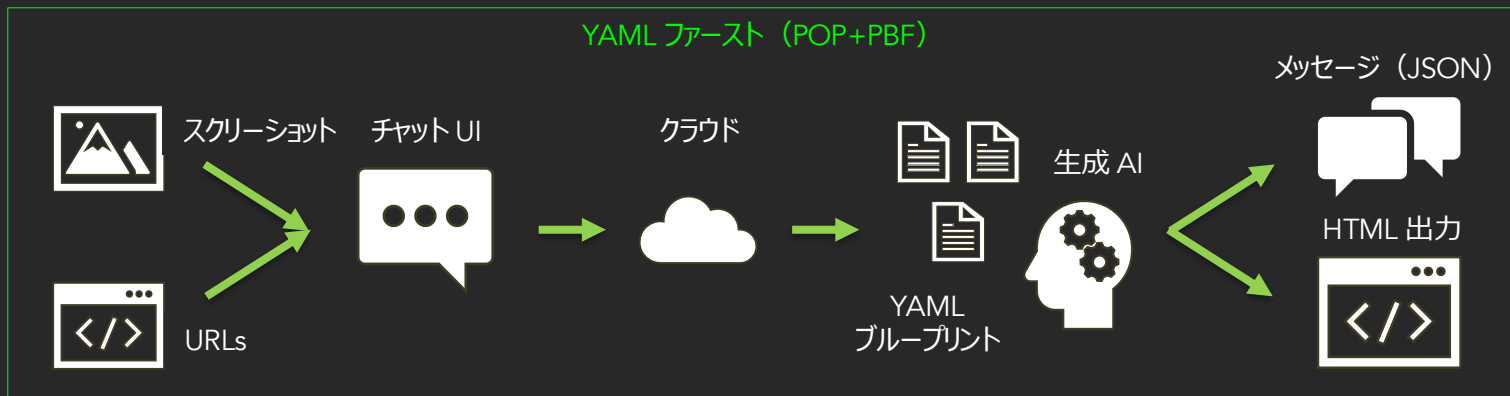
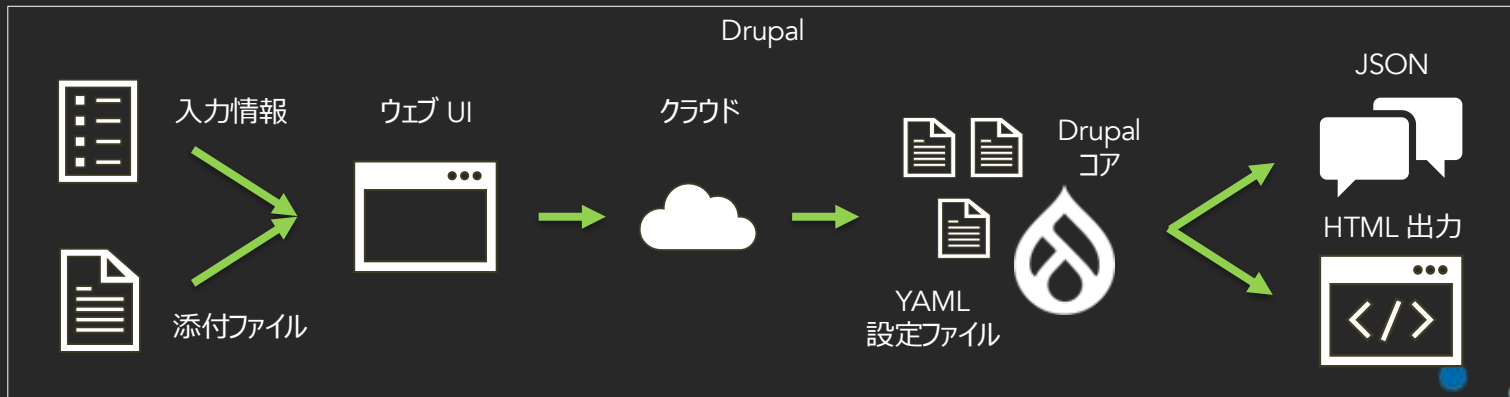
# Drupal ウェブアプリ vs POP



# Drupal ウェブアプリ vs POP

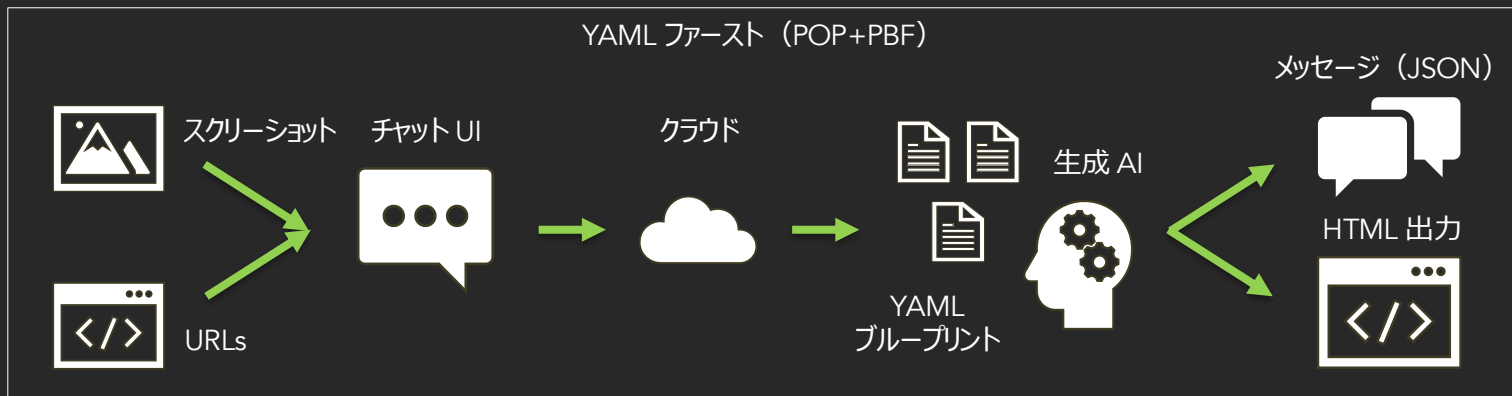
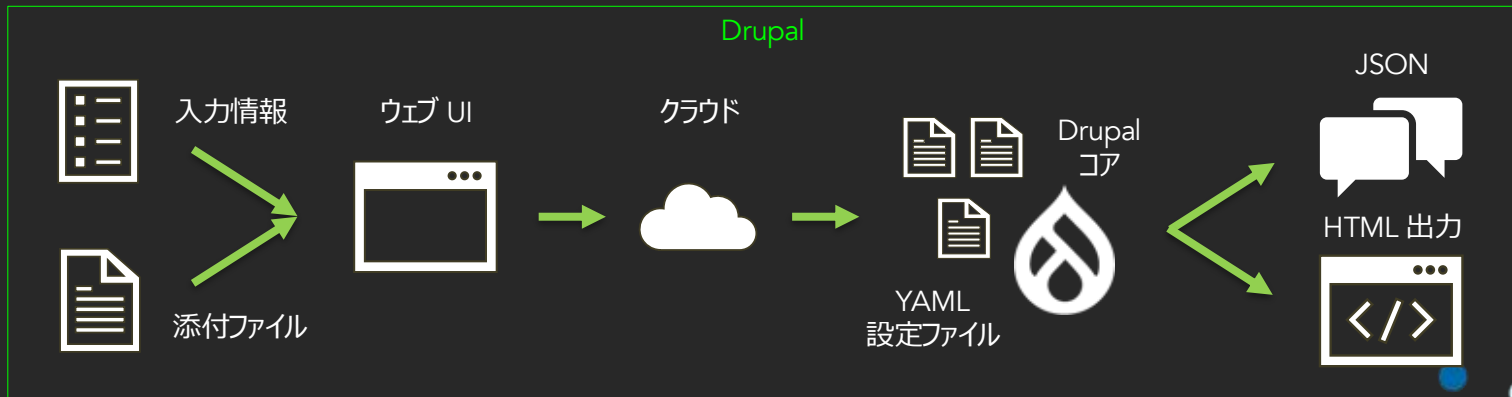


# Drupal ウェブアプリ vs POP

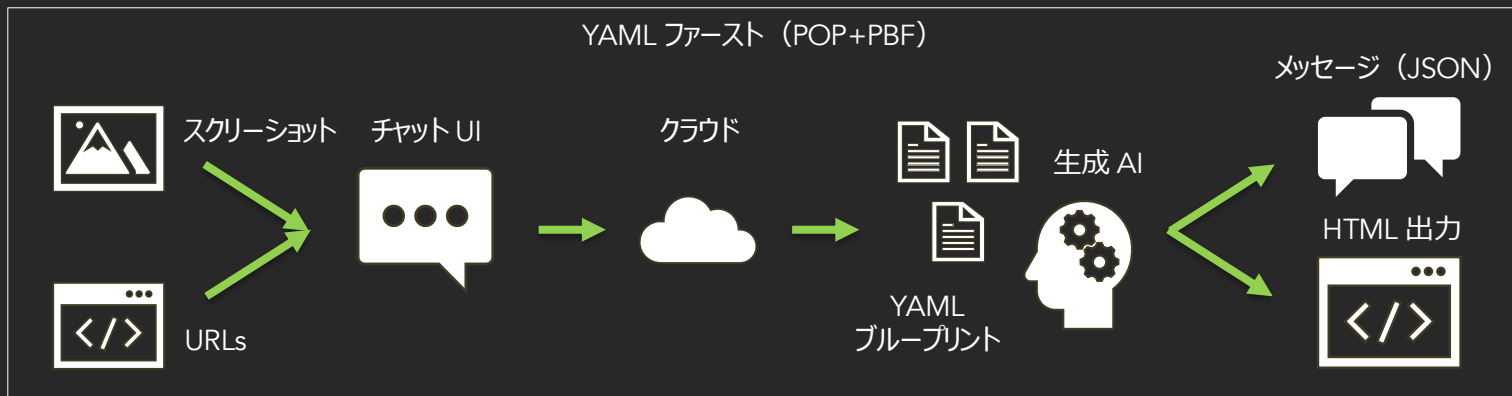
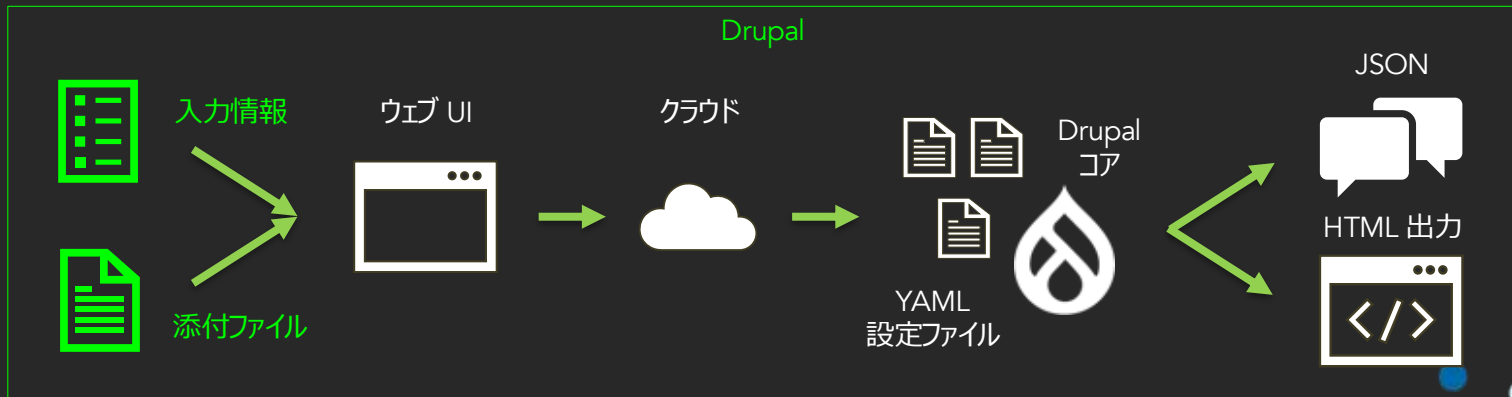




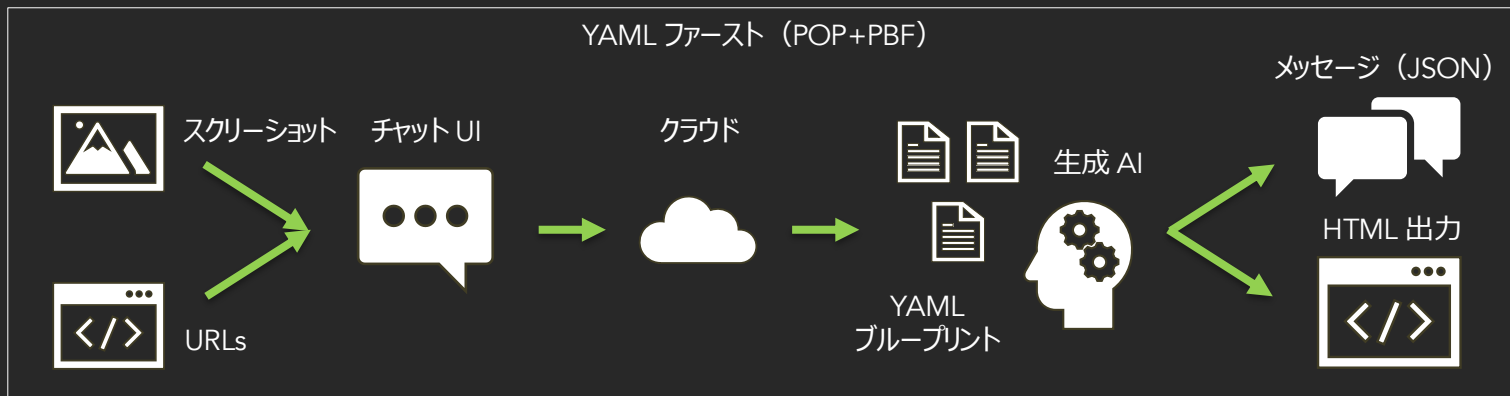
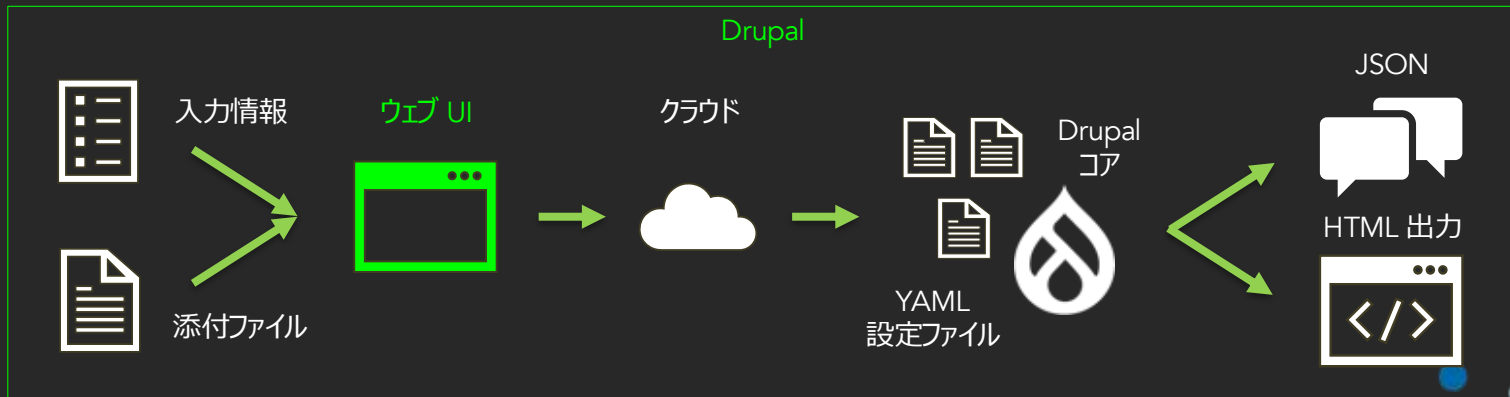
# Drupal ウェブアプリ vs POP



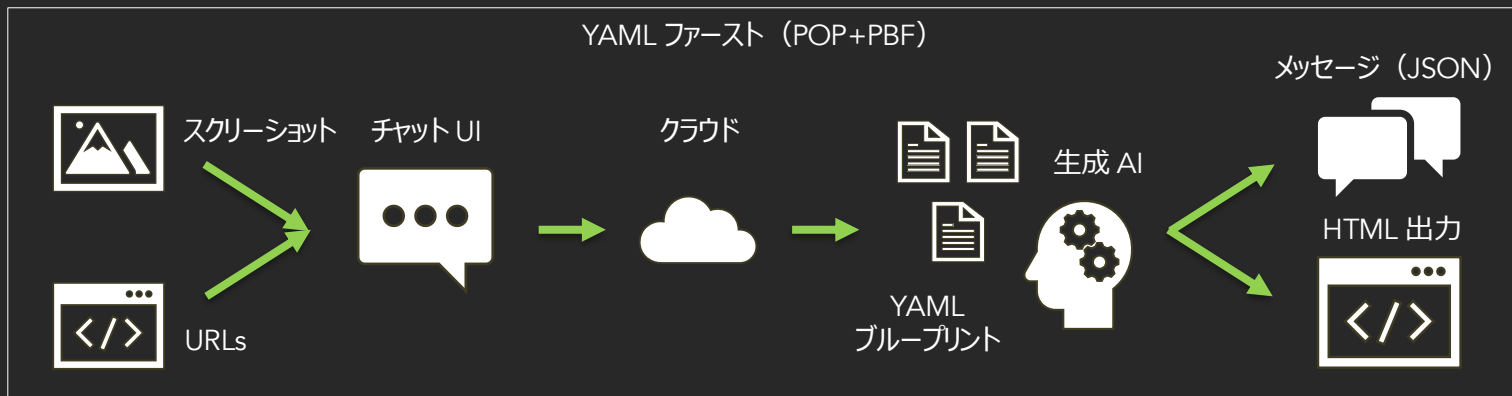
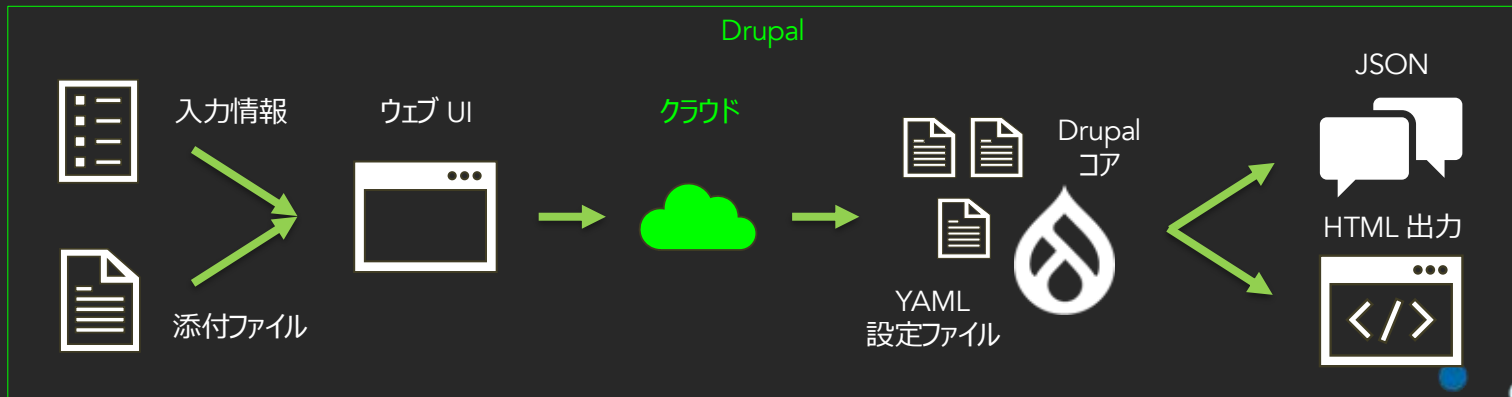
# Drupal ウェブアプリ vs POP



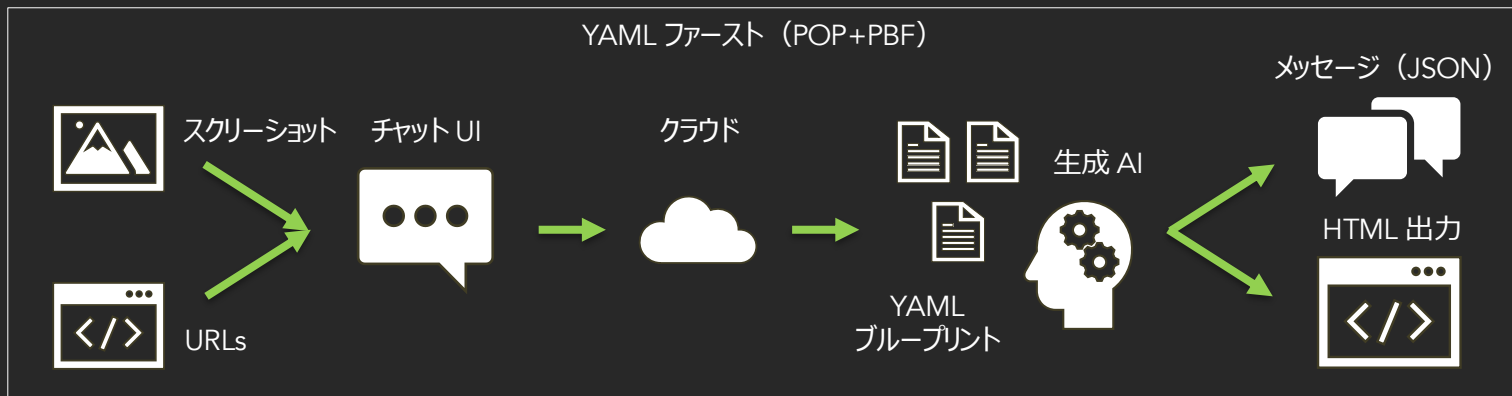
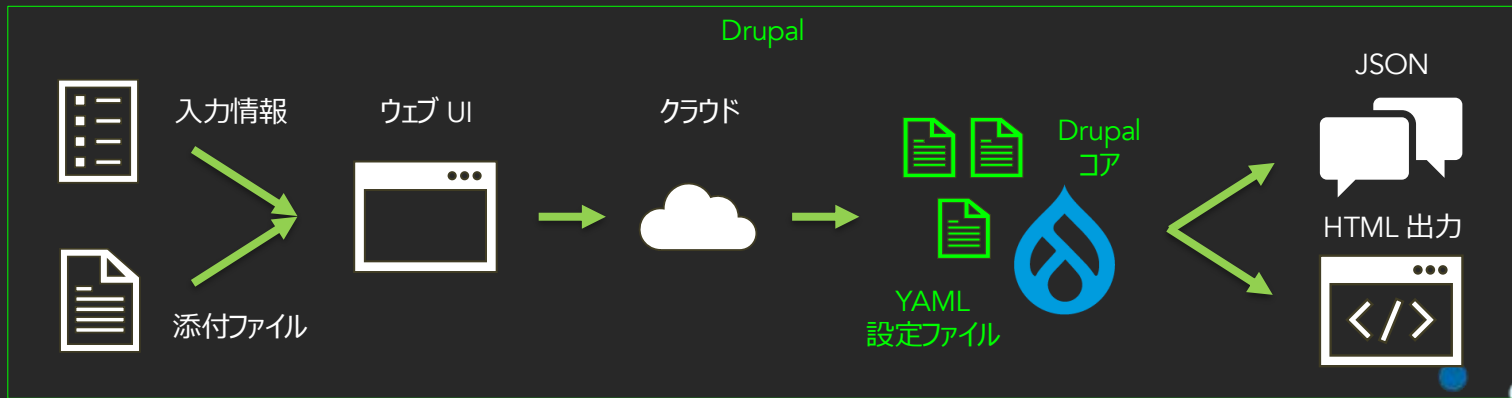
# Drupal ウェブアプリ vs POP



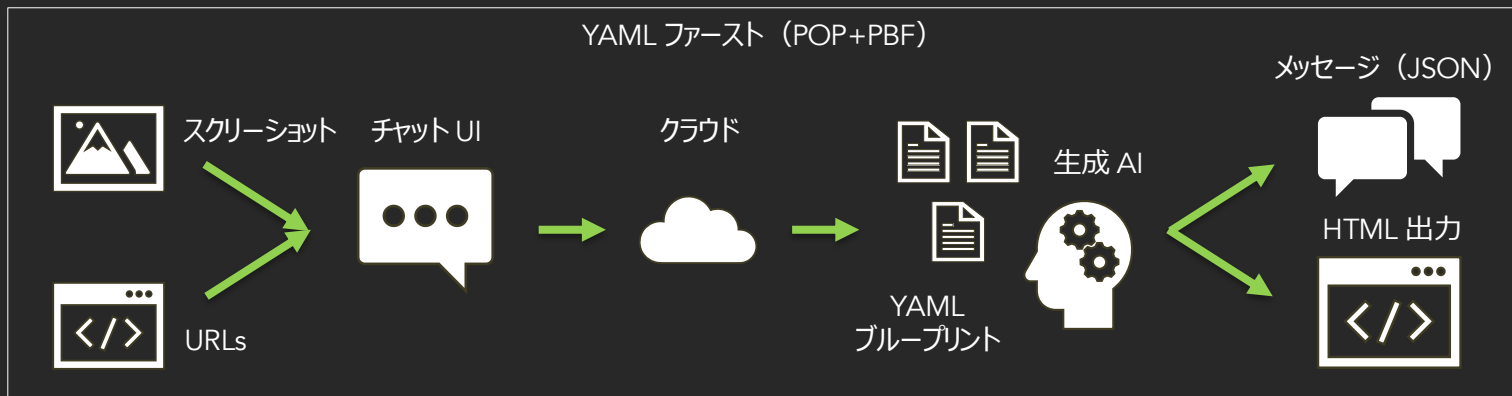
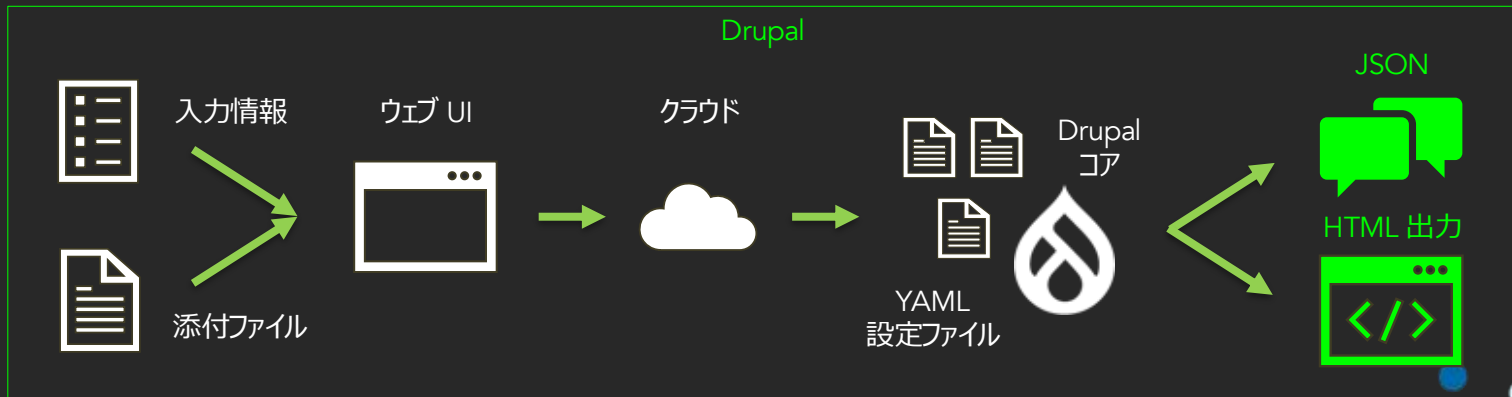
# Drupal ウェブアプリ vs POP



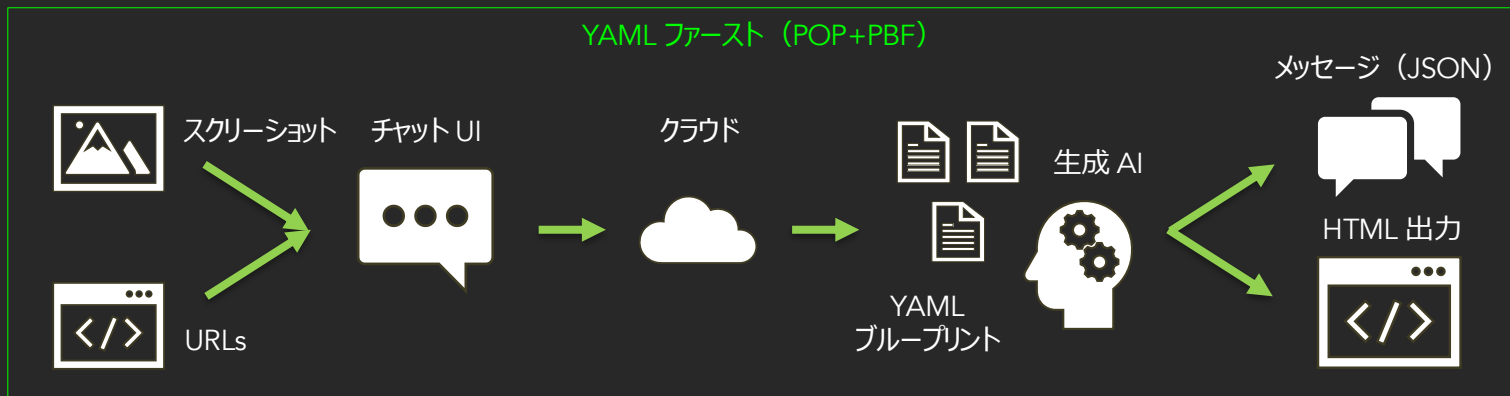
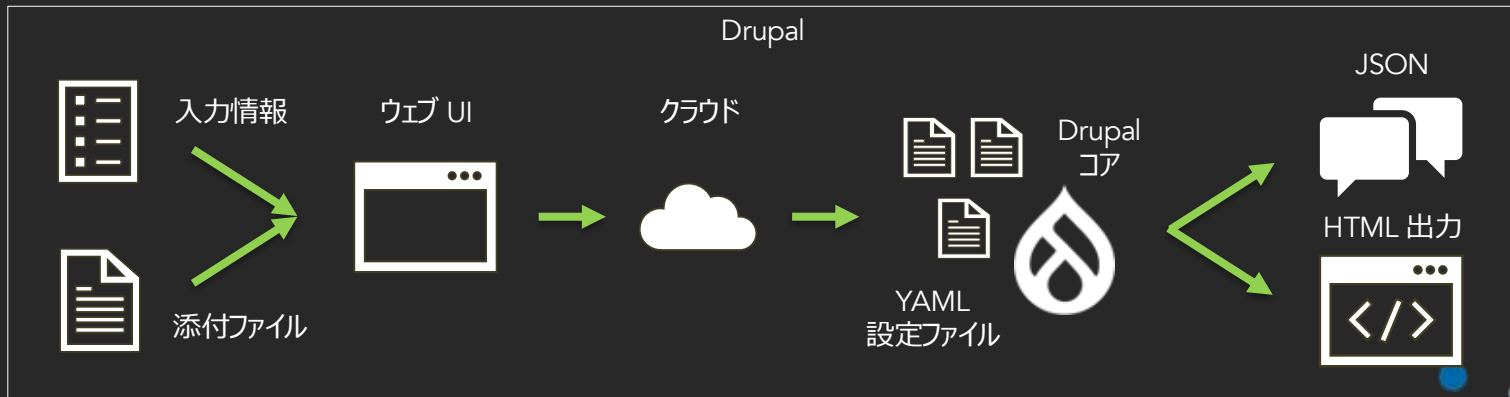
# Drupal ウェブアプリ vs POP



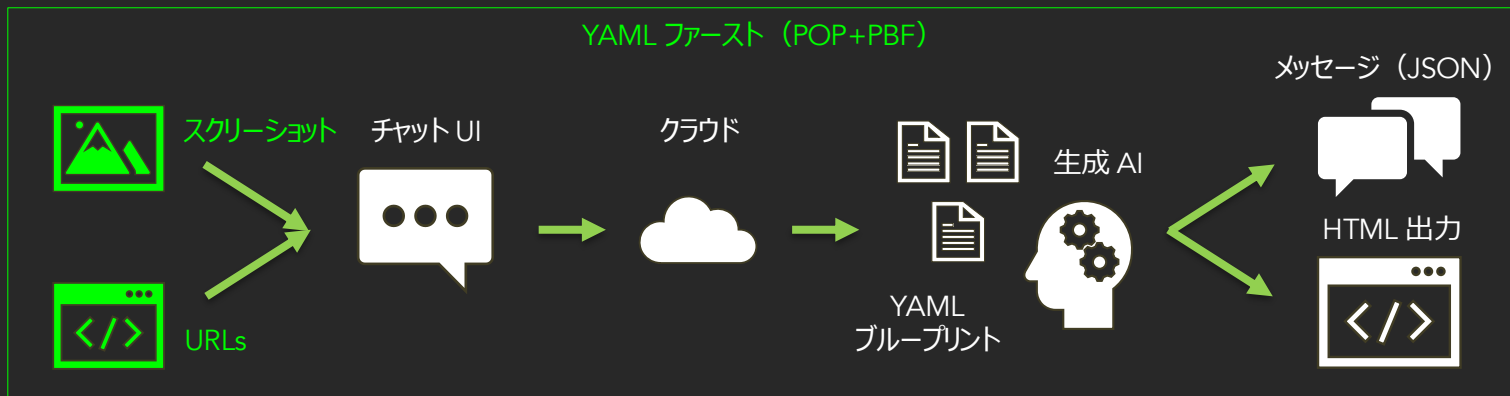
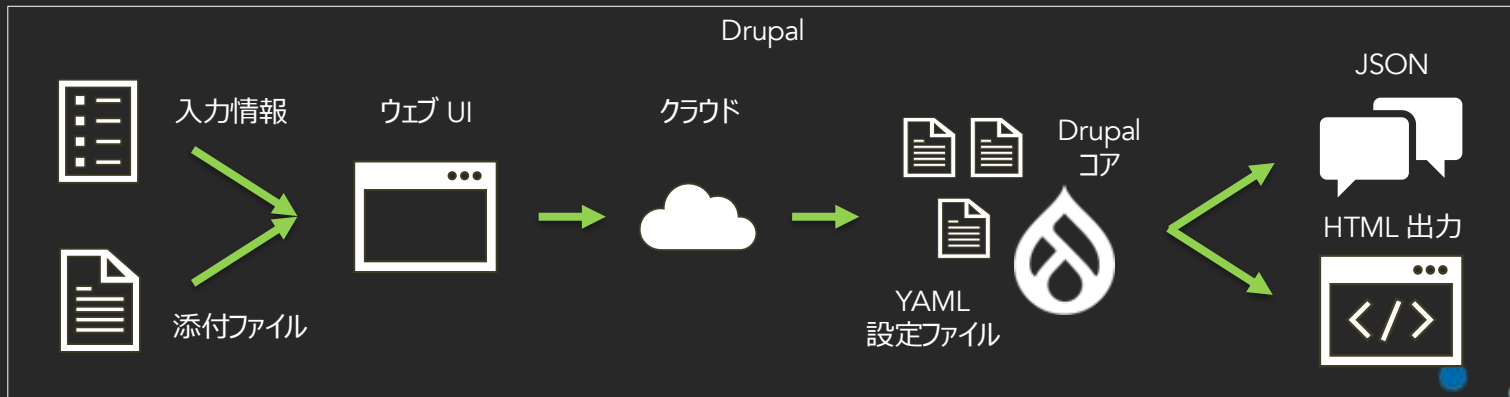
# Drupal ウェブアプリ vs POP



# Drupal ウェブアプリ vs POP

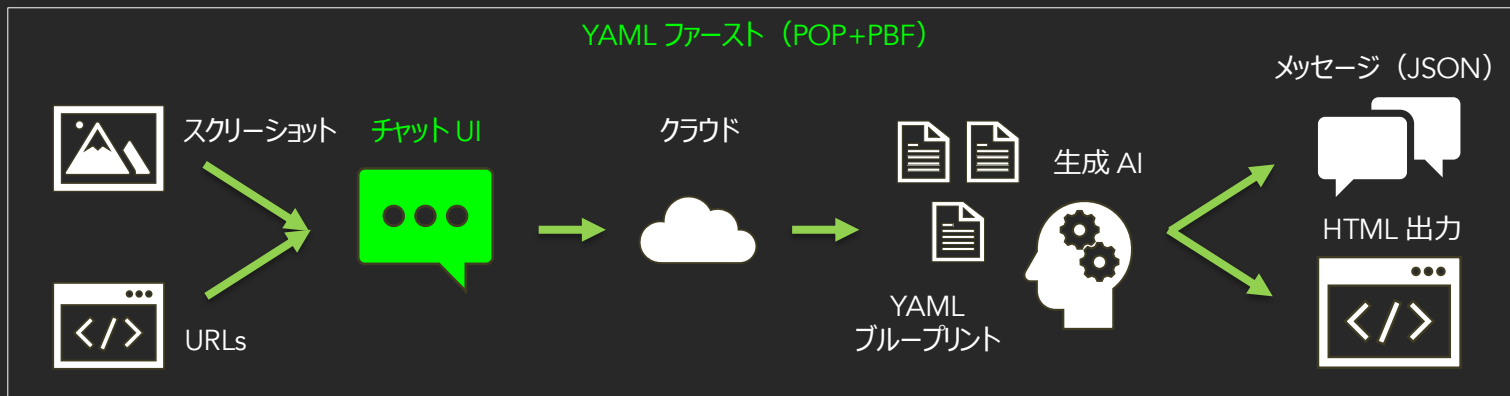
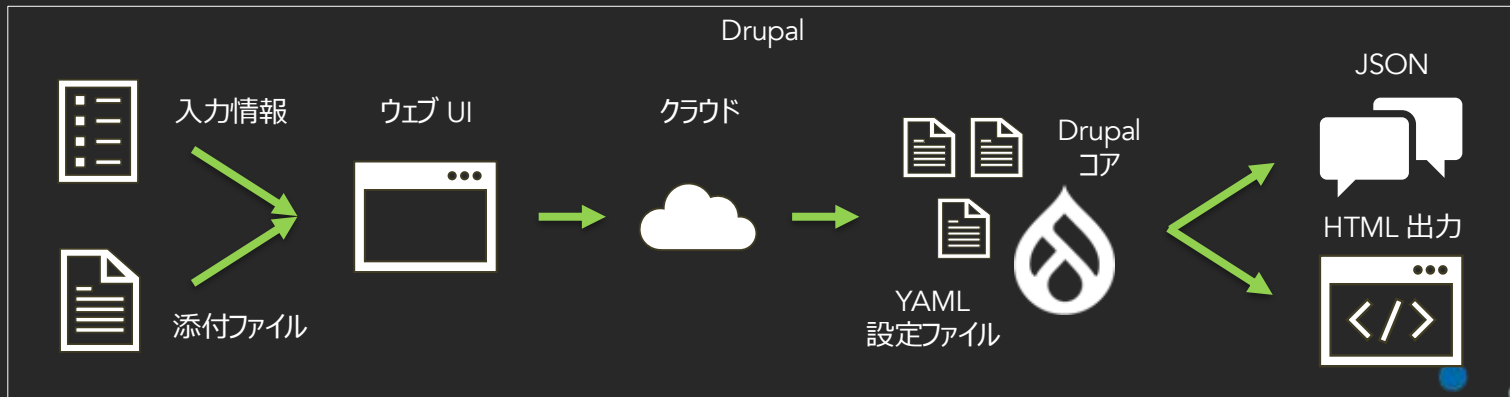


# Drupal ウェブアプリ vs POP

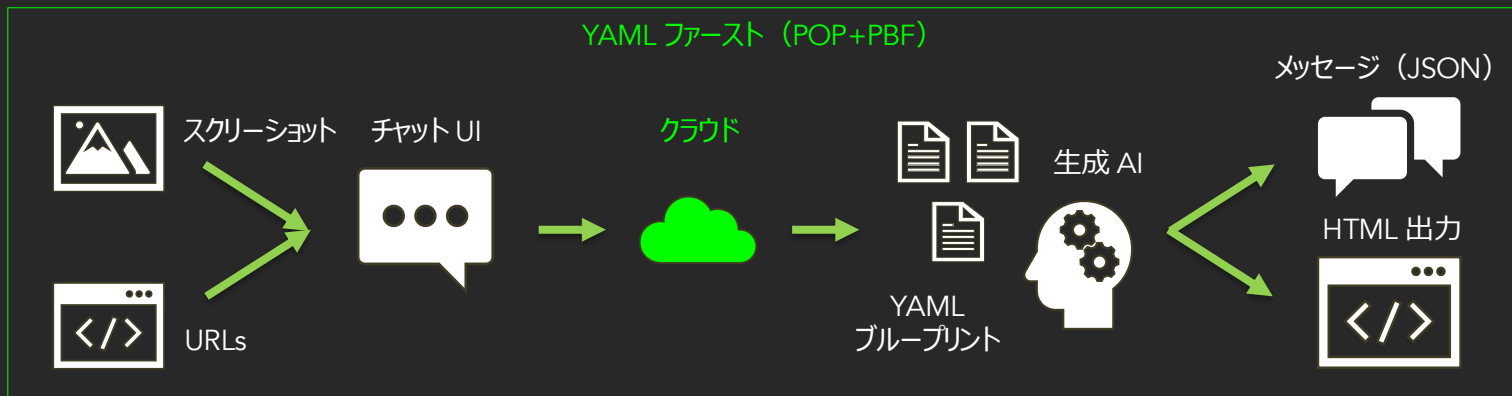
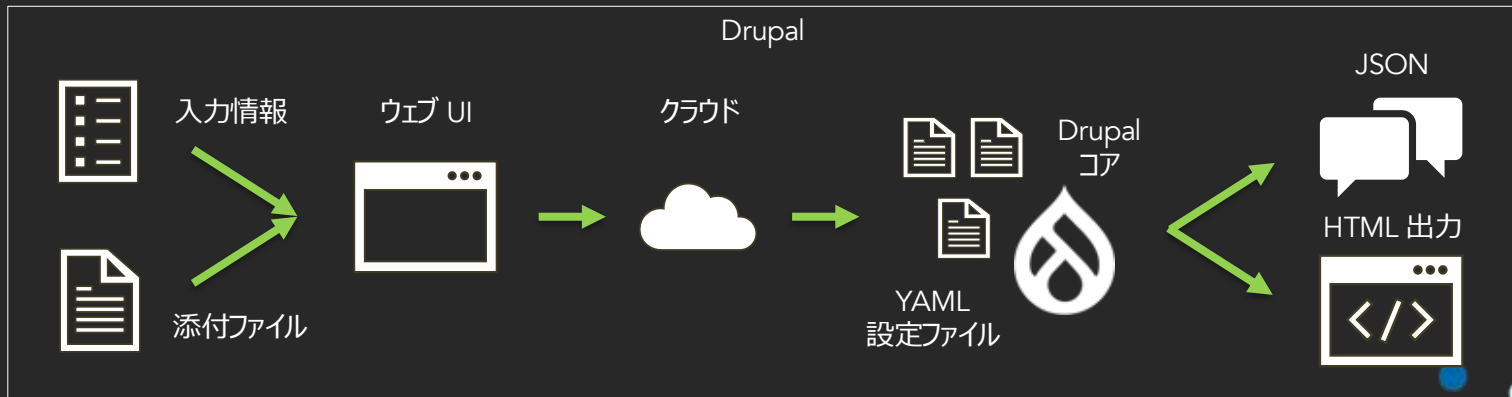




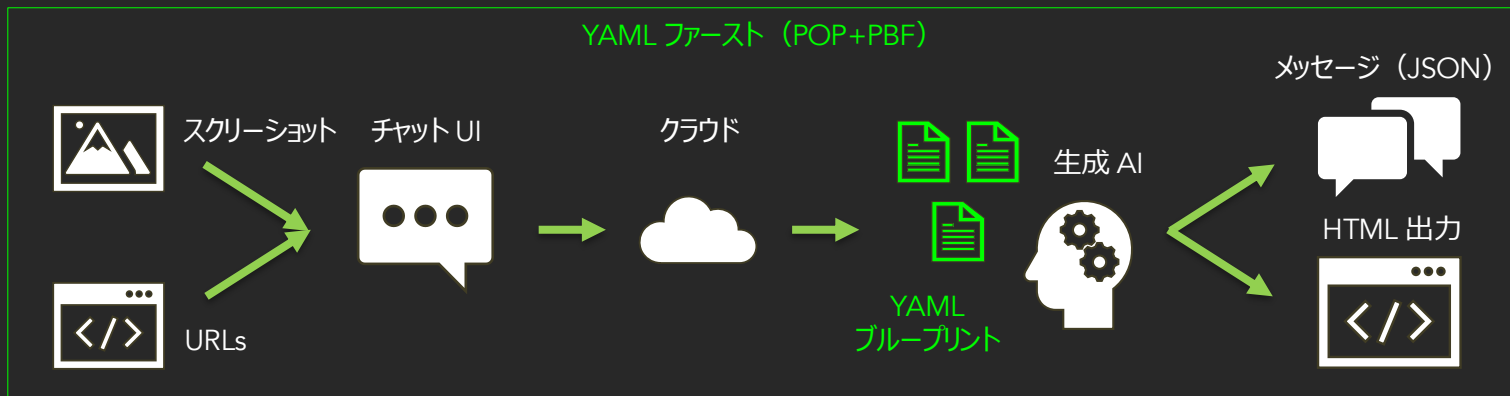
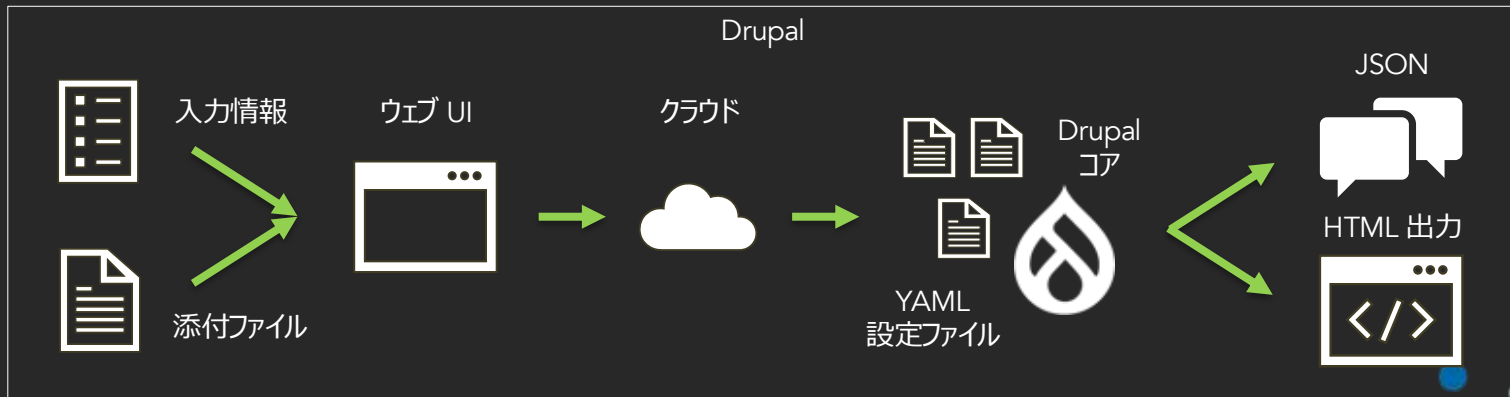
# Drupal ウェブアプリ vs POP



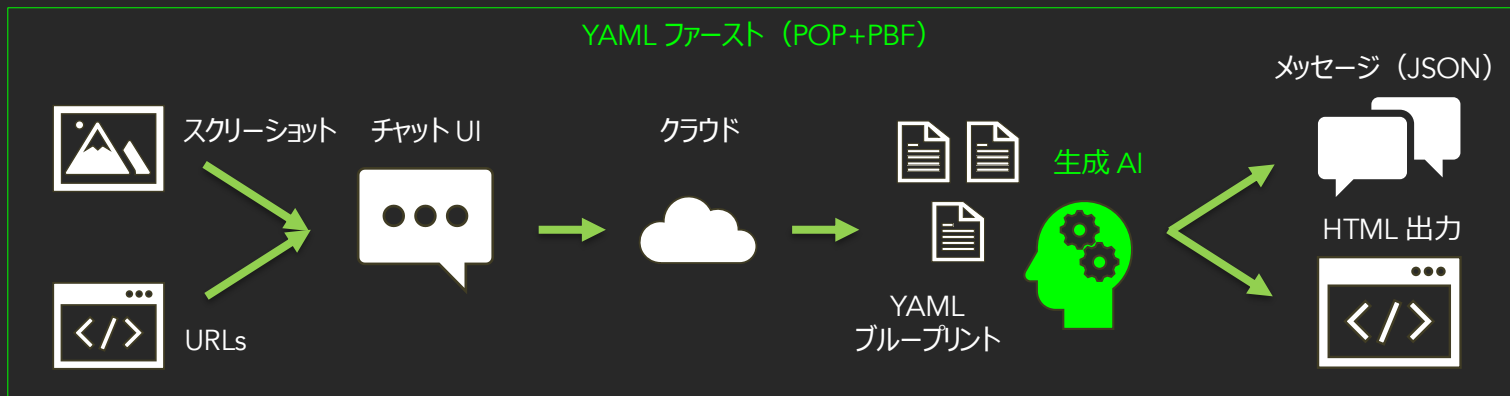
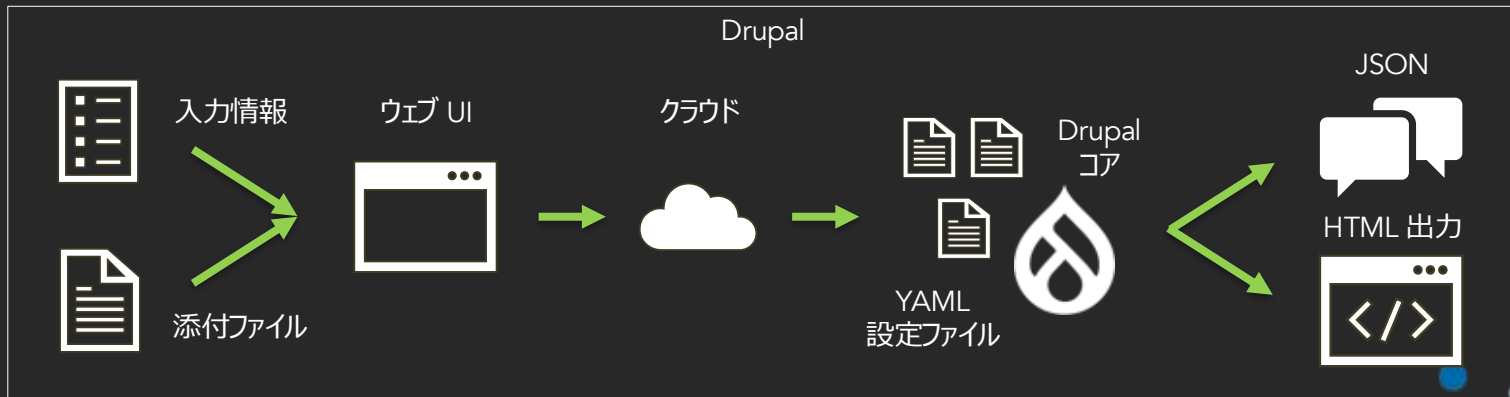
# Drupal ウェブアプリ vs POP



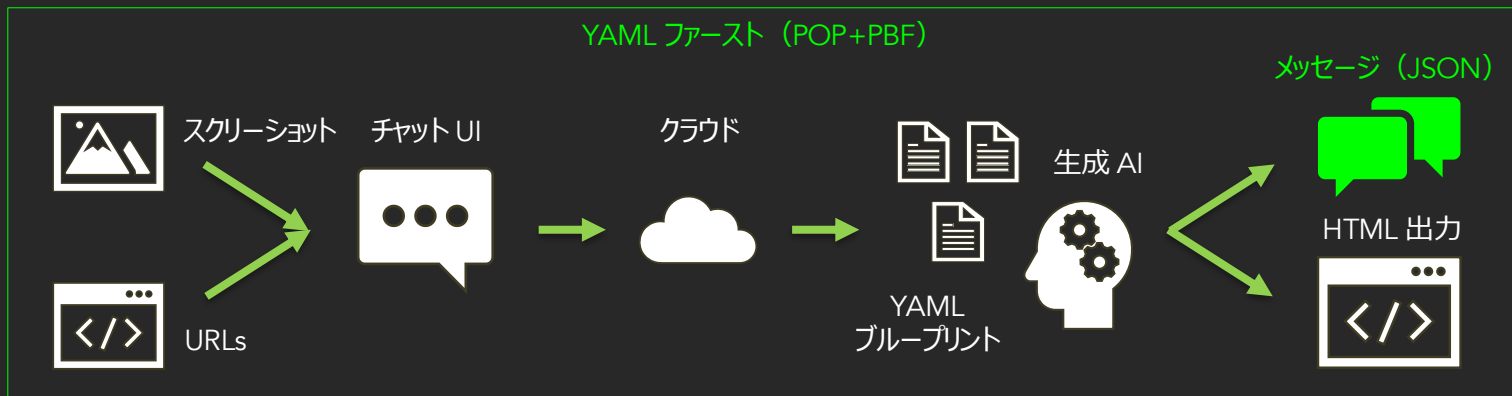
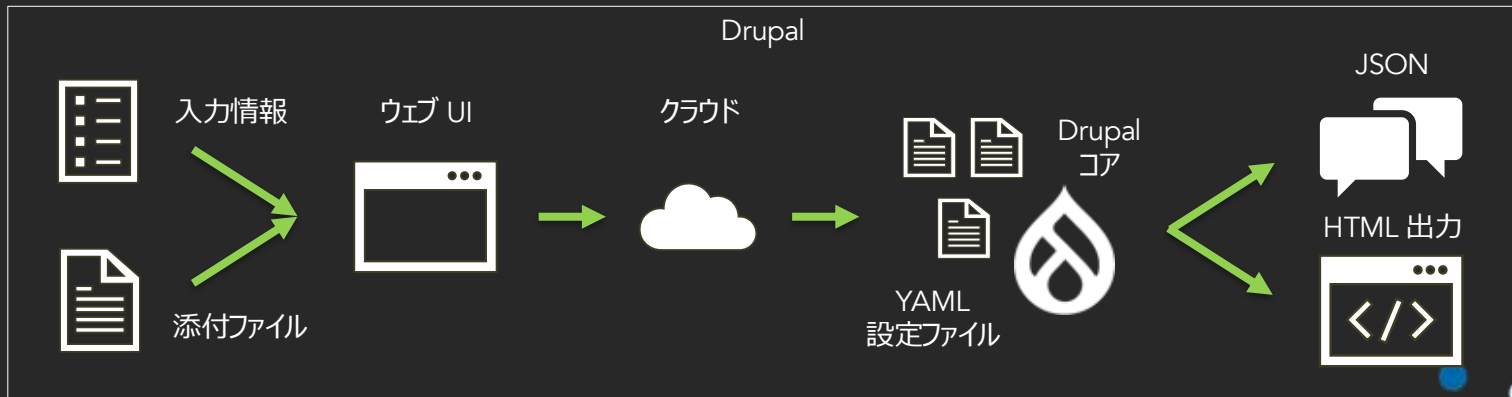
# Drupal ウェブアプリ vs POP



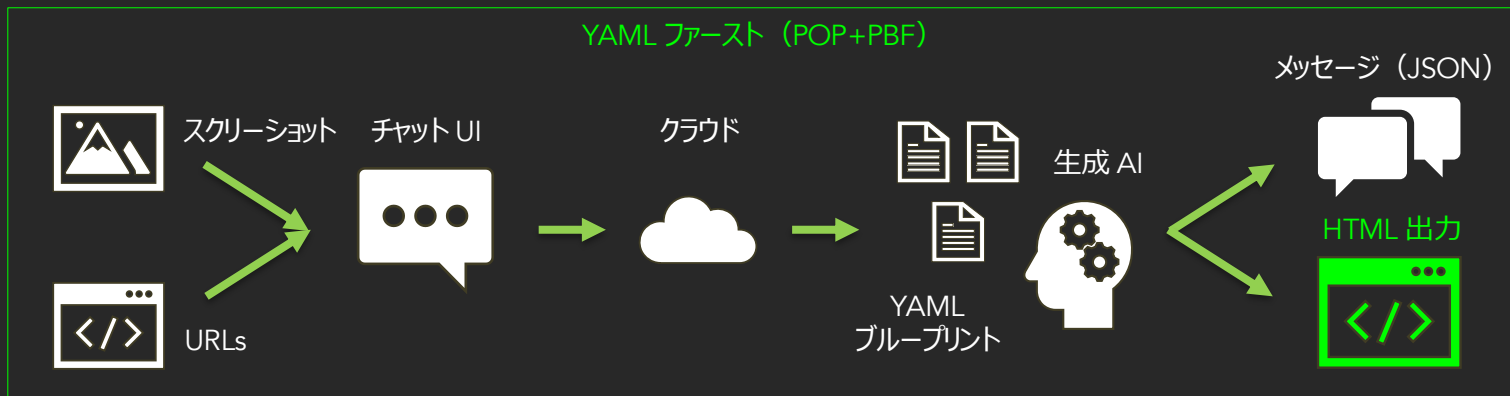
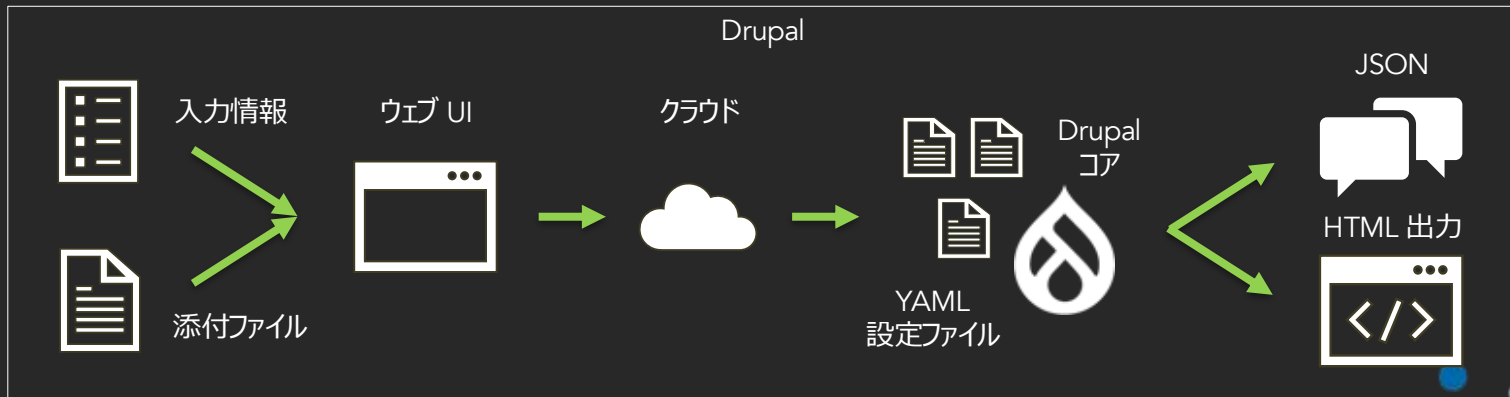
# Drupal ウェブアプリ vs POP



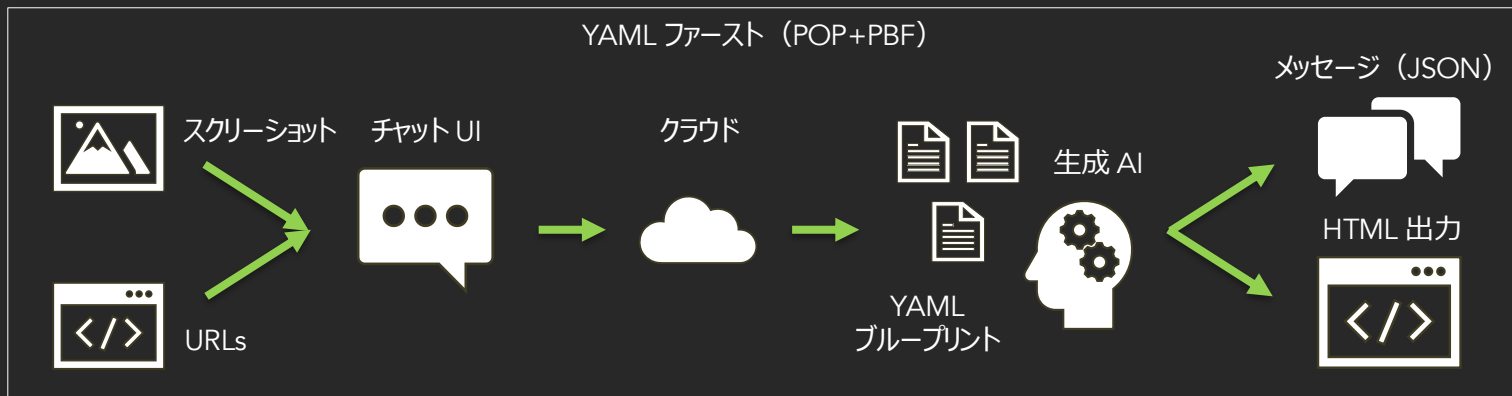
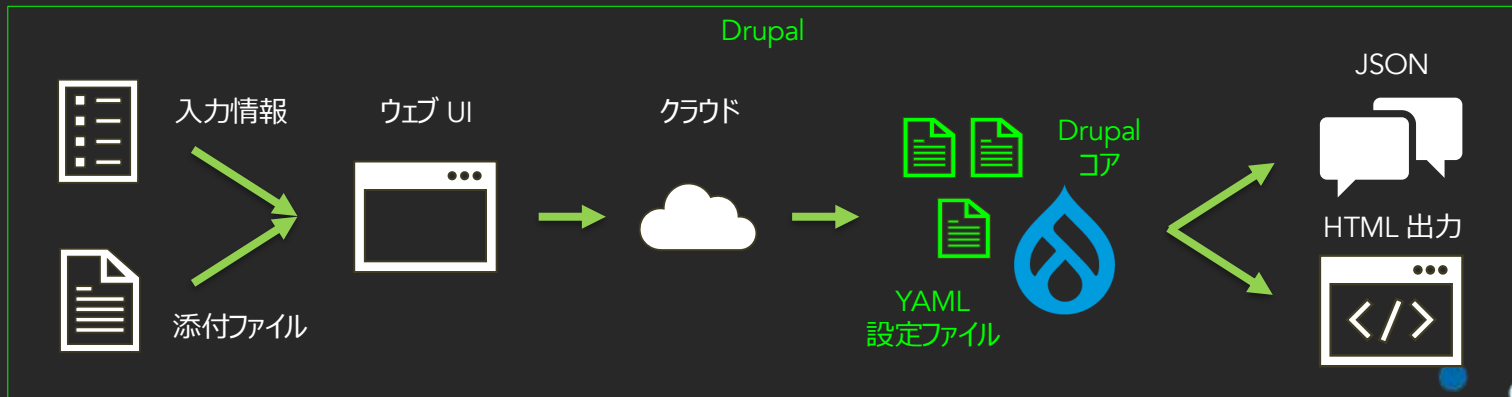
# Drupal ウェブアプリ vs POP



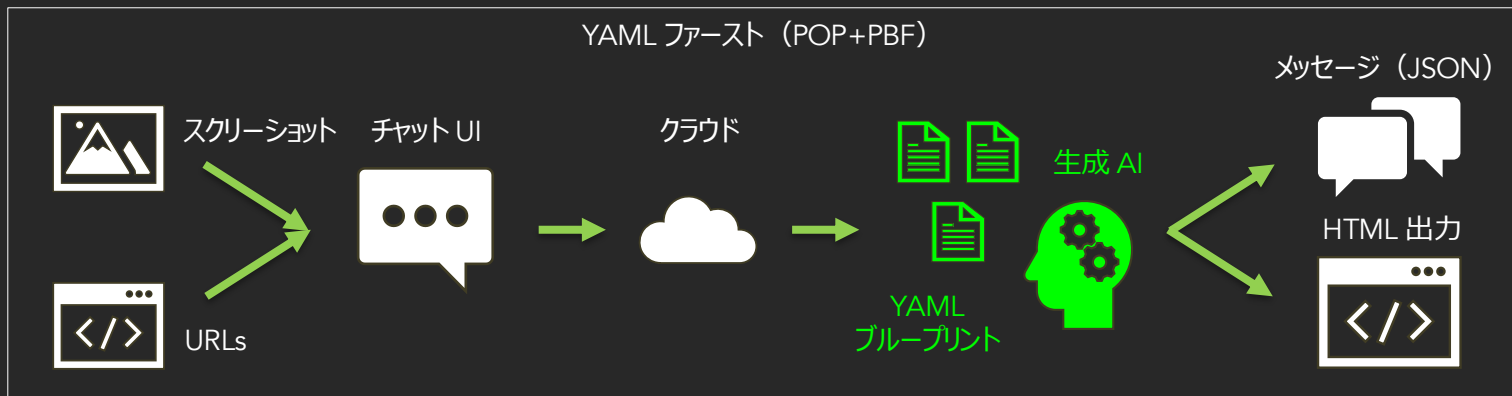
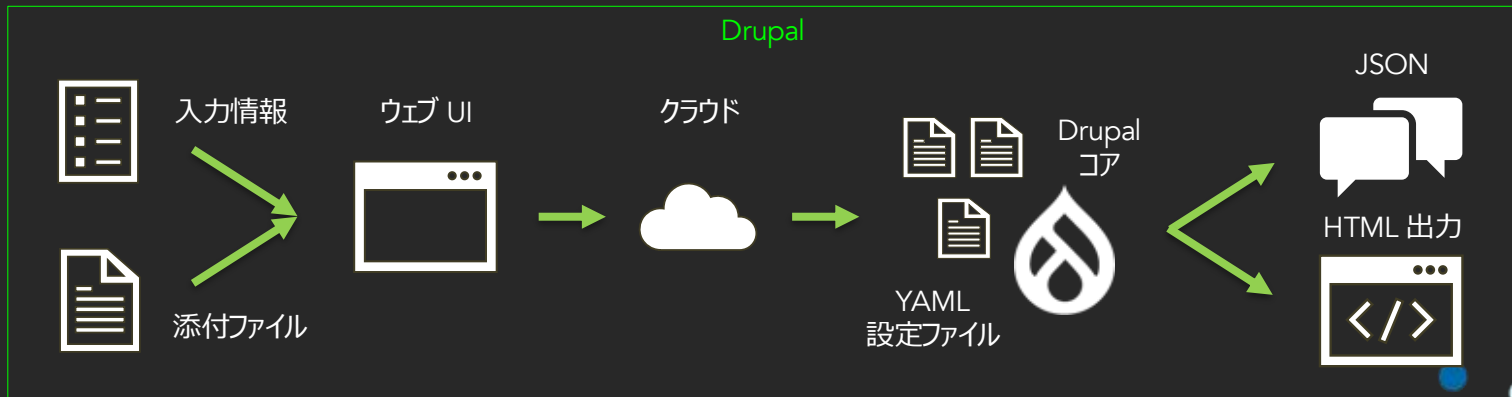
# Drupal ウェブアプリ vs POP



# Drupal ウェブアプリ vs POP



# Drupal ウェブアプリ vs POP





# 「YAML ファースト」の例



# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```



# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
  agent_role: # LLM に対して「あなたは XX です」と暗示をかける
    # 1. Extractor      (抽出)
    # 2. Classifier     (分類)
    # 3. Summarizer     (要約)
    # 4. Validator      (検証)
    # 5. Verifier       (照合)
    # 6. Converter      (変換)
    # 7. Translator / Localizer (翻訳・ローカライズ)
    # 8. Router / Orchestrator (ルーティング・統括)
  Scenario:
    - Role: # 例: extractor / classifier / validator ...
      Instruction:
        Given: # 背景や前提条件
        When: # トリガーとなる条件
        Then: # 期待されるアクション
    ...
  output_schema:
    # 出力フォーマット
    # type: に HTML、JSON、CSV など指定可能
  examples:
    # プロンプトの例
    # 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
  agent_role: orchestrator # 省略可能
  Scenario: # ← 配列なので順序どおり実行
    # ----- Scenario 1 -----
    - Role: extractor
      Instruction:
        Given: ユーザー入力 {{user_input}} が与えられる
        When: 入力を解析する
        Then: 必要フィールドを抽出し、不足分は補完する
    # ----- Scenario 2 -----
    - Role: classifier
      Instruction:
        Given: 出張先の地名または国名が与えられる
        When: 場所レベルを判定する
        Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
    # ----- Scenario 3 -----
    - Role: validator
      Instruction:
        Given: 金額が含まれている
        When: 金額を処理する
        Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
  agent_role: # LLM に対して「あなたは XX です」と暗示をかける
    # 1. Extractor      (抽出)
    # 2. Classifier     (分類)
    # 3. Summarizer     (要約)
    # 4. Validator      (検証)
    # 5. Verifier       (照合)
    # 6. Converter      (変換)
    # 7. Translator / Localizer (翻訳・ローカライズ)
    # 8. Router / Orchestrator (ルーティング・統括)
  Scenario:
    - Role: # 例: extractor / classifier / validator ...
      Instruction:
        Given: # 背景や前提条件
        When: # トリガーとなる条件
        Then: # 期待されるアクション
  ...
  output_schema:
    # 出力フォーマット
    # type: に HTML、JSON、CSV など指定可能
  examples:
    # プロンプトの例
    # 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
  agent_role: orchestrator # 省略可能
  Scenario: # ← 配列なので順序どおり実行
    # ----- Scenario 1 -----
    - Role: extractor
      Instruction:
        Given: ユーザー入力 {{user_input}} が与えられる
        When: 入力を解析する
        Then: 必要フィールドを抽出し、不足分は補完する
    # ----- Scenario 2 -----
    - Role: classifier
      Instruction:
        Given: 出張先の地名または国名が与えられる
        When: 場所レベルを判定する
        Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
    # ----- Scenario 3 -----
    - Role: validator
      Instruction:
        Given: 金額が含まれている
        When: 金額を処理する
        Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter        (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter        (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
Instruction:
  Given: # 背景や前提条件
  When: # トリガーとなる条件
  Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
Instruction:
  Given: ユーザー入力 {{user_input}} が与えられる
  When: 入力を解析する
  Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
Instruction:
  Given: 出張先の地名または国名が与えられる
  When: 場所レベルを判定する
  Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
Instruction:
  Given: 金額が含まれている
  When: 金額を処理する
  Then: 小数点以下を切り上げ、USD に換算する
```



# 「YAML ファースト」の例



```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
Instruction:
  Given: # 背景や前提条件
  When: # トリガーとなる条件
  Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
Instruction:
  Given: ユーザー入力 {{user_input}} が与えられる
  When: 入力を解析する
  Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
Instruction:
  Given: 出張先の地名または国名が与えられる
  When: 場所レベルを判定する
  Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
Instruction:
  Given: 金額が含まれている
  When: 金額を処理する
  Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
Instruction:
  Given: # 背景や前提条件
  When: # トリガーとなる条件
  Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
Instruction:
  Given: ユーザー入力 {{user_input}} が与えられる
  When: 入力を解析する
  Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
Instruction:
  Given: 出張先の地名または国名が与えられる
  When: 場所レベルを判定する
  Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
Instruction:
  Given: 金額が含まれている
  When: 金額を処理する
  Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
Instruction:
  Given: # 背景や前提条件
  When: # トリガーとなる条件
  Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
Instruction:
  Given: ユーザー入力 {{user_input}} が与えられる
  When: 入力を解析する
  Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
Instruction:
  Given: 出張先の地名または国名が与えられる
  When: 場所レベルを判定する
  Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
Instruction:
  Given: 金額が含まれている
  When: 金額を処理する
  Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```



# 「YAML ファースト」の例

```
version: '0.1.0'
prompt:
agent_role: # LLM に対して「あなたは XX です」と暗示をかける
# 1. Extractor      (抽出)
# 2. Classifier      (分類)
# 3. Summarizer      (要約)
# 4. Validator       (検証)
# 5. Verifier        (照合)
# 6. Converter       (変換)
# 7. Translator / Localizer (翻訳・ローカライズ)
# 8. Router / Orchestrator (ルーティング・統括)
Scenario:
- Role: # 例: extractor / classifier / validator ...
  Instruction:
    Given: # 背景や前提条件
    When: # トリガーとなる条件
    Then: # 期待されるアクション
...
output_schema:
# 出力フォーマット
# type: に HTML、JSON、CSV など指定可能
examples:
# プロンプトの例
# 出力例を含めると期待した回答に近づく
```

```
version: '0.1.0'
prompt:
agent_role: orchestrator # 省略可能
Scenario: # ← 配列なので順序どおり実行
# ----- Scenario 1 -----
- Role: extractor
  Instruction:
    Given: ユーザー入力 {{user_input}} が与えられる
    When: 入力を解析する
    Then: 必要フィールドを抽出し、不足分は補完する
# ----- Scenario 2 -----
- Role: classifier
  Instruction:
    Given: 出張先の地名または国名が与えられる
    When: 場所レベルを判定する
    Then: Level1 / Level2 / Level3 に分類し宿泊費上限を設定する
# ----- Scenario 3 -----
- Role: validator
  Instruction:
    Given: 金額が含まれている
    When: 金額を処理する
    Then: 小数点以下を切り上げ、USD に換算する
```

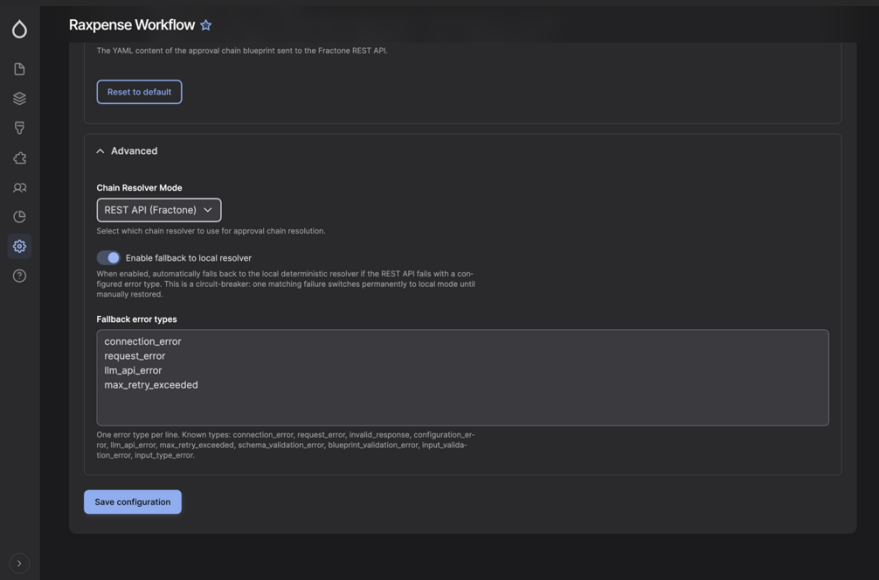
# DrupalCon Nara から今までに何を作ったか



# DrupalCon Nara から今までに何を作ったか



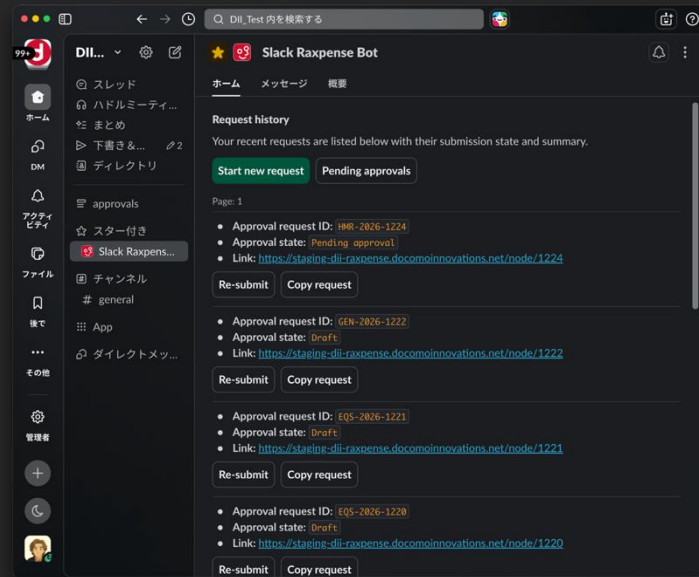
- POP ランタイム開発 (コードネーム Fractone)
- Slack UI
- OCR 連携
- DII 承認ワークフロー実装
- YAML ブループリントビルダー



# DrupalCon Nara から今までに何を作ったか



- POP ランタイム開発 (コードネーム Fractone)
- Slack UI
- OCR 連携
- DII 承認ワークフロー実装
- YAML ブループリントビルダー



# DrupalCon Nara から今までに何を作ったか

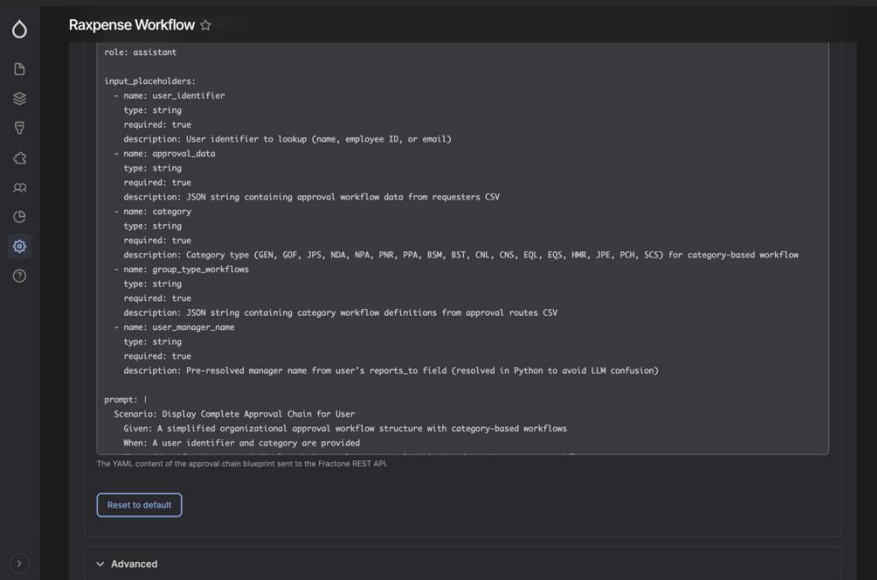


- POP ランタイム開発 (コードネーム Fractone)
- Slack UI
- OCR 連携
- DII 承認ワークフロー実装
- YAML ブループリントビルダー

# DrupalCon Nara から今までに何を作ったか



- POP ランタイム開発 (コードネーム Fractone)
- Slack UI
- OCR 連携
- **DII 承認ワークフロー実装**
- YAML ブループリントビルダー





- Back to site

Edit YAML

AGS Workflow TestPublished

AGS Workflow Test

Title \*  
AGS Workflow Test

Attachment  

ags-workflowpatterns 20191204-12.opptx (49.57 KB)Remove

Summary  
Choose heading
  - B
  - I
  - Quote
  - Link
  - Table
  - Code
  - Image
  - Video
  - Audio
  - Embed
  - Source

|            | Routing           | Routing Pattern #3 | Routing Pattern #4 | Routing Pattern #5 | Routing Patterns #6 | Routing Pattern #7 |
|------------|-------------------|--------------------|--------------------|--------------------|---------------------|--------------------|
| Originator | Originator        | Originator         | Originator         | Originator         | Originator          | Originator         |
| PM         | PM                | PM                 | PM                 | IP Admin           | PM                  | COO                |
| Director   | Corporate Counsel | Director           | Director           | PM                 | Director            | CEO                |
| COO        |                   | Director           | HR Mgr             | Director           | COO                 |                    |

Published

Last saved: 21 Nov 2025 - 06:56

Author: admin

Create new revision

Revision log message

Briefly describe the changes you have made.

Menu settings  
Not in menu

URL alias  
Automatic alias

URL redirects

Authoring information  
By admin (1) on 2025-11-21

Promotion options  
Promoted to front page

# DrupalCon Nara から今までに何を作ったか



- POP ランタイム開発 (コードネーム Fractone)
- Slack UI
- OCR 連携
- DII 承認ワークフロー実装
- **YAML ブループリントビルダー**

A screenshot of a web-based editor for a workflow blueprint. The interface has a dark theme. On the left is a sidebar with various icons for navigation. The main area is titled "AGS Workflow Test" and "Published". It displays a YAML configuration for a workflow. The YAML content includes a version, a prompt, a scenario with roles and instructions, and an output schema. The instructions describe a process for analyzing requirements and determining workflow steps based on total amount and group type.

```
1 version: 1.0
2 prompt: |
3   - Scenario:
4     - Role: extractor
5     Instruction:
6       Given: The workflow requirements, routing patterns, processing roles, and embedded business
7         rules regarding approval hierarchy and amount thresholds.
8       When: Analyzing requirements to determine workflow steps, routing conditions, and roles
9         involved.
10      Then: |
11        Identify sequential workflow roles based on 'Type of workflow' and routing patterns.
12        Embed logic: If total_amount > $30,000 (GA group) or > $50,000 (other groups), CEO approval
13        is required for BST, GEN, GOF, PMU, CNT, and HMR workflows.
14        If total_amount <= $300, CEO approval is not required and workflow completes at
15        Director's approval.
16      Output the derived steps, roles, and decision conditions.
17    - Role: classifier
18    Instruction:
19      Given: The initial Gherkin block and workflow properties.
20      When: Reviewing for logic accuracy and normalizing terminology (e.g., 'total amount' ->
21        'total_amount').
22      Then: Return the normalized Gherkin prompt and a refined list of schema properties
23        (e.g., total_amount: number, workflow_type: string, group: string).
24  output_schema:
25    type: object
26    properties:
27      workflow_type:
28        type: string
29        description: "Type of the workflow pattern, such as BST, CNT, PTD, HMR, PPAB, SHJ, PGEN,
30          PMU, NPA, JPS, GOF, PMU."
31      group:
```





# YAML ブループリントビルダー

[Log in](#)[Reset your password](#)[Home](#) > [Log in](#)

## Log in

**Username or email address \***

Enter your username or email address.

**Password \***

# Drupal の未来



- Drupal は YAML で設定を定義
- POP は YAML で認知を定義
- そして、YAML は AI の思考を定義

# Drupal の未来



- Drupal は YAML で設定を定義
- POP は YAML で認知を定義
- そして、YAML は AI の思考を定義

# Drupal の未来



- Drupal は YAML で設定を定義
- POP は YAML で認知を定義
- そして、YAML は AI の思考を定義

# POP がもたらしたイノベーション



# 実例① 承認ルートをコードから CSV へ



|    | A       | B        | C            | D            | E            | F      | G      | H                |
|----|---------|----------|--------------|--------------|--------------|--------|--------|------------------|
| 1  | group   | category | role_1       | role_2       | role_3       | role_4 | role_5 | is_self_approval |
| 2  | GA      | GOF      | COO          | CEO          |              |        |        |                  |
| 3  | GA      | NPA      | COO          |              |              |        |        |                  |
| 4  | DEFAULT | GEN      | Manager      | Group Leader | COO          |        |        |                  |
| 5  | DEFAULT | JPS      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 6  | DEFAULT | NDA      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 7  | DEFAULT | PNR      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 8  | DEFAULT | PPA      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 9  | DEFAULT | BSM      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 10 | DEFAULT | BST      | Manager      | Group Leader | COO          |        |        |                  |
| 11 | DEFAULT | CNS      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 12 | DEFAULT | CNL      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 13 | DEFAULT | HMR      | Manager      | Group Leader | HR Manager   | COO    |        |                  |
| 14 | DEFAULT | JPE      | HR Manager   | COO          | CEO          |        |        |                  |
| 15 | DEFAULT | PCH      | Manager      | Group Leader |              |        |        |                  |
| 16 | DEFAULT | EQL      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 17 | DEFAULT | EQS      | Manager      | Group Leader | COO          |        |        |                  |
| 18 | DEFAULT | SCS      | IT Admin     | Manager      | Group Leader | COO    |        |                  |
| 19 | DEFAULT | EXP      | Acct Manager | COO          |              | COO    |        | 1                |

```

group,category,role_1,role_2,role_3,role_4,role_5,is_self_approval
GA,GOF,COO,CEO,,,,
GA,NPA,COO,,,,
DEFAULT,GEN,Manager,Group Leader,COO,,,
DEFAULT,JPS,Manager,Group Leader,COO,CEO,,
DEFAULT,NDA,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PNR,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PPA,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,BSM,Manager,Group Leader,COO,CEO,,
DEFAULT,BST,Manager,Group Leader,COO,,,
DEFAULT,CNS,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,CNL,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,HMR,Manager,Group Leader,HR Manager,COO,,
DEFAULT,JPE,HR Manager,COO,CEO,,,
DEFAULT,PCH,Manager,Group Leader,,,,
DEFAULT,EQL,Manager,Group Leader,COO,CEO,,
DEFAULT,EQS,Manager,Group Leader,COO,,,
DEFAULT,SCS,IT Admin,Manager,Group Leader,COO,,
DEFAULT,EXP,Acct Manager,COO,,,,1
    
```

# 実例① 承認ルートをコードから CSV へ



|    | A       | B        | C            | D            | E            | F      | G      | H                |
|----|---------|----------|--------------|--------------|--------------|--------|--------|------------------|
| 1  | group   | category | role_1       | role_2       | role_3       | role_4 | role_5 | is_self_approval |
| 2  | GA      | GOF      | COO          | CEO          |              |        |        |                  |
| 3  | GA      | NPA      | COO          |              |              |        |        |                  |
| 4  | DEFAULT | GEN      | Manager      | Group Leader | COO          |        |        |                  |
| 5  | DEFAULT | JPS      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 6  | DEFAULT | NDA      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 7  | DEFAULT | PNR      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 8  | DEFAULT | PPA      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 9  | DEFAULT | BSM      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 10 | DEFAULT | BST      | Manager      | Group Leader | COO          |        |        |                  |
| 11 | DEFAULT | CNS      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 12 | DEFAULT | CNL      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 13 | DEFAULT | HMR      | Manager      | Group Leader | HR Manager   | COO    |        |                  |
| 14 | DEFAULT | JPE      | HR Manager   | COO          | CEO          |        |        |                  |
| 15 | DEFAULT | PCH      | Manager      | Group Leader |              |        |        |                  |
| 16 | DEFAULT | EQL      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 17 | DEFAULT | EQS      | Manager      | Group Leader | COO          |        |        |                  |
| 18 | DEFAULT | SCS      | IT Admin     | Manager      | Group Leader | COO    |        |                  |
| 19 | DEFAULT | EXP      | Acct Manager | COO          |              | COO    |        | 1                |

```

group,category,role_1,role_2,role_3,role_4,role_5,is_self_approval
GA,GOF,COO,CEO,,,,
GA,NPA,COO,,,,
DEFAULT,GEN,Manager,Group Leader,COO,,,
DEFAULT,JPS,Manager,Group Leader,COO,CEO,,
DEFAULT,NDA,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PNR,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PPA,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,BSM,Manager,Group Leader,COO,CEO,,
DEFAULT,BST,Manager,Group Leader,COO,,,
DEFAULT,CNS,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,CNL,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,HMR,Manager,Group Leader,HR Manager,COO,,
DEFAULT,JPE,HR Manager,COO,CEO,,,
DEFAULT,PCH,Manager,Group Leader,,,,
DEFAULT,EQL,Manager,Group Leader,COO,CEO,,
DEFAULT,EQS,Manager,Group Leader,COO,,,
DEFAULT,SCS,IT Admin,Manager,Group Leader,COO,,
DEFAULT,EXP,Acct Manager,COO,,,,1
    
```



# 実例① 承認ルートをコードから CSV へ



- コード変更不要
- ワークフローの承認ルート追加 = CSV 1行追加

|    | A       | B        | C            | D            | E            | F      | G      | H                |
|----|---------|----------|--------------|--------------|--------------|--------|--------|------------------|
| 1  | group   | category | role_1       | role_2       | role_3       | role_4 | role_5 | is_self_approval |
| 2  | GA      | GOF      | COO          | CEO          |              |        |        |                  |
| 3  | GA      | NPA      | COO          |              |              |        |        |                  |
| 4  | DEFAULT | GEN      | Manager      | Group Leader | COO          |        |        |                  |
| 5  | DEFAULT | JPS      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 6  | DEFAULT | NDA      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 7  | DEFAULT | PNR      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 8  | DEFAULT | PPA      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 9  | DEFAULT | BSM      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 10 | DEFAULT | BST      | Manager      | Group Leader | COO          |        |        |                  |
| 11 | DEFAULT | CNS      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 12 | DEFAULT | CNL      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 13 | DEFAULT | HMR      | Manager      | Group Leader | HR Manager   | COO    |        |                  |
| 14 | DEFAULT | JPE      | HR Manager   | COO          | CEO          |        |        |                  |
| 15 | DEFAULT | PCH      | Manager      | Group Leader |              |        |        |                  |
| 16 | DEFAULT | EQL      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 17 | DEFAULT | EQS      | Manager      | Group Leader | COO          |        |        |                  |
| 18 | DEFAULT | SCS      | IT Admin     | Manager      | Group Leader | COO    |        |                  |
| 19 | DEFAULT | EXP      | Acct Manager | COO          |              | COO    |        | 1                |

```

group,category,role_1,role_2,role_3,role_4,role_5,is_self_approval
GA,GOF,COO,CEO,,,,
GA,NPA,COO,,,,
DEFAULT,GEN,Manager,Group Leader,COO,,,
DEFAULT,JPS,Manager,Group Leader,COO,CEO,,
DEFAULT,NDA,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PNR,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PPA,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,BSM,Manager,Group Leader,COO,CEO,,
DEFAULT,BST,Manager,Group Leader,COO,,,
DEFAULT,CNS,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,CNL,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,HMR,Manager,Group Leader,HR Manager,COO,,
DEFAULT,JPE,HR Manager,COO,CEO,,,
DEFAULT,PCH,Manager,Group Leader,,,,
DEFAULT,EQL,Manager,Group Leader,COO,CEO,,
DEFAULT,EQS,Manager,Group Leader,COO,,,
DEFAULT,SCS,IT Admin,Manager,Group Leader,COO,,
DEFAULT,EXP,Acct Manager,COO,,,,1
    
```

# 実例① 承認ルートをコードから CSV へ



- コード変更不要
- ワークフローの承認ルート追加 = CSV 1行追加

|    | A       | B        | C            | D            | E            | F      | G      | H                |
|----|---------|----------|--------------|--------------|--------------|--------|--------|------------------|
| 1  | group   | category | role_1       | role_2       | role_3       | role_4 | role_5 | is_self_approval |
| 2  | GA      | GOF      | COO          | CEO          |              |        |        |                  |
| 3  | GA      | NPA      | COO          |              |              |        |        |                  |
| 4  | DEFAULT | GEN      | Manager      | Group Leader | COO          |        |        |                  |
| 5  | DEFAULT | JPS      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 6  | DEFAULT | NDA      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 7  | DEFAULT | PNR      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 8  | DEFAULT | PPA      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 9  | DEFAULT | BSM      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 10 | DEFAULT | BST      | Manager      | Group Leader | COO          |        |        |                  |
| 11 | DEFAULT | CNS      | Legal Admin  | Manager      | Group Leader | COO    |        |                  |
| 12 | DEFAULT | CNL      | Legal Admin  | Manager      | Group Leader | COO    | CEO    |                  |
| 13 | DEFAULT | HMR      | Manager      | Group Leader | HR Manager   | COO    |        |                  |
| 14 | DEFAULT | JPE      | HR Manager   | COO          | CEO          |        |        |                  |
| 15 | DEFAULT | PCH      | Manager      | Group Leader |              |        |        |                  |
| 16 | DEFAULT | EQL      | Manager      | Group Leader | COO          | CEO    |        |                  |
| 17 | DEFAULT | EQS      | Manager      | Group Leader | COO          |        |        |                  |
| 18 | DEFAULT | SCS      | IT Admin     | Manager      | Group Leader | COO    |        |                  |
| 19 | DEFAULT | EXP      | Acct Manager | COO          |              | COO    |        | 1                |

```

group,category,role_1,role_2,role_3,role_4,role_5,is_self_approval
GA,GOF,COO,CEO,,,,,
GA,NPA,COO,,,,,
DEFAULT,GEN,Manager,Group Leader,COO,,,
DEFAULT,JPS,Manager,Group Leader,COO,CEO,,
DEFAULT,NDA,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PNR,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,PPA,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,BSM,Manager,Group Leader,COO,CEO,,
DEFAULT,BST,Manager,Group Leader,COO,,,
DEFAULT,CNS,Legal Admin,Manager,Group Leader,COO,,
DEFAULT,CNL,Legal Admin,Manager,Group Leader,COO,CEO,
DEFAULT,HMR,Manager,Group Leader,HR Manager,COO,,
DEFAULT,JPE,HR Manager,COO,CEO,,,
DEFAULT,PCH,Manager,Group Leader,,,,
DEFAULT,EQL,Manager,Group Leader,COO,CEO,,
DEFAULT,EQS,Manager,Group Leader,COO,,,
DEFAULT,SCS,IT Admin,Manager,Group Leader,COO,,
DEFAULT,EXP,Acct Manager,COO,,,,1
    
```

# 実例② 承認ワークフローの実装



# LLM のルート計算は信用できるか？



- 本当に毎回正しい承認ルートを返すのか 🤔

Raxpense Workflow ☆

^ Blueprint YAML

Blueprint YAML

```
name: Approval Chain Resolver
version: '1.0'
role: assistant

input_placeholders:
- name: user_identifier
  type: string
  required: true
  description: User identifier to lookup (name, employee ID, or email)
- name: approval_data
  type: string
  required: true
  description: JSON string containing approval workflow data from requesters CSV
- name: category
  type: string
  required: true
  description: Category type (GEN, GOF, JPS, NDA, NPA, PNR, PPA, BSM, BST, CNL, CNS, EQL, EQS, HMR, JPE, PCH, SCS) for category-based workflow
- name: group_type_workflows
  type: string
  required: true
  description: JSON string containing category workflow definitions from approval routes CSV
- name: user_manager_name
  type: string
  required: true
  description: Pre-resolved manager name from user's reports_to field (resolved in Python to avoid LLM confusion)

prompt: |
Scenario: Display Complete Approval Chain for User
Given: A simplified organizational approval workflow structure with category-based workflows
When: A user identifier and category are provided
```

The YAML content of the approval chain blueprint sent to the Fractone REST API.

# LLM のルート計算は信用できるか？



- 本当に毎回正しい承認ルートを返すのか 🤔

Raxpense Workflow ☆

^ Blueprint YAML

Blueprint YAML \*

```
name: Approval Chain Resolver
version: '1.0'
role: assistant

input_placeholders:
- name: user_identifier
  type: string
  required: true
  description: User identifier to lookup (name, employee ID, or email)
- name: approval_data
  type: string
  required: true
  description: JSON string containing approval workflow data from requesters CSV
- name: category
  type: string
  required: true
  description: Category type (GEN, GOF, JPS, NDA, NPA, PNR, PPA, BSM, BST, CNL, CNS, EQL, EQS, HMR, JPE, PCH, SCS) for category-based workflow
- name: group_type_workflows
  type: string
  required: true
  description: JSON string containing category workflow definitions from approval routes CSV
- name: user_manager_name
  type: string
  required: true
  description: Pre-resolved manager name from user's reports_to field (resolved in Python to avoid LLM confusion)

prompt: |
Scenario: Display Complete Approval Chain for User
Given: A simplified organizational approval workflow structure with category-based workflows
When: A user identifier and category are provided
```

The YAML content of the approval chain blueprint sent to the Fractone REST API.

# 最初の結果



- 開発者は 100% 正確だ、と言っていたのに
- QA 環境にデプロイしたらいきなりルート計算が間違っている

# 最初の結果



- 開発者は 100% 正確だ、と言っていたのに
- QA 環境にデプロイしたらいきなりルート計算が間違っている 🤖

# 原因調査

- @example.com が  
実データと衝突
- Examples のサンプルが  
実例として解釈された

Raxpense Workflow ☆

^ Blueprint YAML

Blueprint YAML \*

```
examples:
  - input:
      user_identifier: "Taro"
      approval_data: [{"requester": "Taro Yamada", "email": "taro@example.com", "group": "OISG", "approval_role": null, "reports_to": "James Smith"}, {"requester": "James Smith", "email": "james@example.com", "group": "OISG", "approval_role": "Group Leader", "reports_to": "Kenji Tanaka"}, {"requester": "Hanako Sato", "email": "hanako@example.com", "group": "GA", "approval_role": "Legal Admin", "reports_to": "Akira Suzuki"}, {"requester": "Akira Suzuki", "email": "akira@example.com", "group": "GA", "approval_role": "COO", "reports_to": "Kenji Tanaka"}, {"requester": "Kenji Tanaka", "email": "kenji@example.com", "group": "GA", "approval_role": "CEO", "reports_to": null}]]
      category: "NDA"
      group_type_workflows: [{"DEFAULT_NDA": [{"Legal Admin", "PERSON:James Smith", "PERSON:James Smith", "COO", "CEO"}]}]
      user_manager_name: "James Smith"
    output:
      user_name: "Taro Yamada"
      user_email: "taro@example.com"
      user_title: ""
      approval_chain:
        - level: 1
          name: "Hanako Sato"
          title: "Legal Admin"
          email: "hanako@example.com"
          is_user: raise
        - level: 2
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
          is_user: raise
        - level: 3
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
```

The YAML content of the approval chain blueprint sent to the Fractone REST API.



# 原因調査

- @example.com が  
実データと衝突
- Examples のサンプルが  
実例として解釈された

Raxpense Workflow ☆

^ Blueprint YAML

Blueprint YAML \*

```
examples:
  - input:
      user_identifier: "Taro"
      approval_data: '[{"requester": "Taro Yamada", "email": "taro@example.com", "group": "OISG", "approval_role": null, "reports_to": "James Smith"}, {"requester": "James Smith", "email": "james@example.com", "group": "OISG", "approval_role": "Group Leader", "reports_to": "Kenji Tanaka"}, {"requester": "Hanako Sato", "email": "hanako@example.com", "group": "GA", "approval_role": "Legal Admin", "reports_to": "Akira Suzuki"}, {"requester": "Akira Suzuki", "email": "akira@example.com", "group": "GA", "approval_role": "COO", "reports_to": "Kenji Tanaka"}, {"requester": "Kenji Tanaka", "email": "kenji@example.com", "group": "GA", "approval_role": "CEO", "reports_to": null}]']
      category: "NDA"
      group_type_workflows: '["DEFAULT_NDA": [{"Legal Admin", "PERSON:James Smith", "PERSON:James Smith", "COO", "CEO"}]]'
      user_manager_name: "James Smith"
    output:
      user_name: "Taro Yamada"
      user_email: "taro@example.com"
      user_title: ""
      approval_chain:
        - level: 1
          name: "Hanako Sato"
          title: "Legal Admin"
          email: "hanako@example.com"
          is_user: raise
        - level: 2
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
          is_user: raise
        - level: 3
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
```

The YAML content of the approval chain blueprint sent to the Fractone REST API.

# 原因調査

- @example.com が  
実データと衝突
- Examples のサンプルが  
実例として解釈された

Raxpense Workflow ☆

^ Blueprint YAML

Blueprint YAML \*

```
examples:
  - input:
      user_identifier: "Taro"
      approval_data: '[{"requester": "Taro Yamada", "email": "taro@example.com", "group": "OISG", "approval_role": null, "reports_to": "James Smith"}, {"requester": "James Smith", "email": "james@example.com", "group": "OISG", "approval_role": "Group Leader", "reports_to": "Kenji Tanaka"}, {"requester": "Hanako Sato", "email": "hanako@example.com", "group": "GA", "approval_role": "Legal Admin", "reports_to": "Akira Suzuki"}, {"requester": "Akira Suzuki", "email": "akira@example.com", "group": "GA", "approval_role": "COO", "reports_to": "Kenji Tanaka"}, {"requester": "Kenji Tanaka", "email": "kenji@example.com", "group": "GA", "approval_role": "CEO", "reports_to": null}]'      category: "NDA"
      group_type_workflows: '["DEFAULT_NDA": [{"Legal Admin", "PERSON:James Smith", "PERSON:James Smith", "COO", "CEO"}]]'
      user_manager_name: "James Smith"
    output:
      user_name: "Taro Yamada"
      user_email: "taro@example.com"
      user_title: ""
      approval_chain:
        - level: 1
          name: "Hanako Sato"
          title: "Legal Admin"
          email: "hanako@example.com"
          is_user: false
        - level: 2
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
          is_user: false
        - level: 3
          name: "James Smith"
          title: "Group Leader"
          email: "james@example.com"
```

The YAML content of the approval chain blueprint sent to the Fractone REST API.

# 改善と再検証

- プロンプトを改善後、検証を継続実施
- 510パターン × 20セット以上

| #  | Target  | Type | Host                           | Email Domain | Username              | Email                         | Blueprint | Test Failures                                |
|----|---------|------|--------------------------------|--------------|-----------------------|-------------------------------|-----------|----------------------------------------------|
| 11 | Generic |      | http://dev1.dataninja.net:8083 | @example.com | actual_first_last     | actual_first_last@example.com | Original  | 116/5,610 cases passed (11/11 trials failed) |
| 12 |         |      |                                |              | requesters_actual.csv |                               |           | 116/5,100 cases passed (10/10 trials failed) |

| # | Target | Type   | Host                           | Email Domain           | Username                         | Email                            | Blueprint | Test Failures                                  |
|---|--------|--------|--------------------------------|------------------------|----------------------------------|----------------------------------|-----------|------------------------------------------------|
| 1 | DII    | Script | http://dev1.dataninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com | first.last@docomoinnovations.com | Original  | 27/7,140 cases failed (14/14 trial set failed) |
| 2 |        |        |                                |                        | requesters.csv                   |                                  |           | 30/5,100 cases failed (9/10 trial set failed)  |

# 改善と再検証

- プロンプトを改善後、検証を継続実施
- 510パターン × 20セット以上

| #  | Target  | Type | Host                           | Email Domain | Username              | Email                         | Blueprint | Test Failures                                |
|----|---------|------|--------------------------------|--------------|-----------------------|-------------------------------|-----------|----------------------------------------------|
| 11 | Generic |      | http://dev1.dataninja.net:8083 | @example.com | actual_first_last     | actual_first_last@example.com | Original  | 116/5,610 cases passed (11/11 trials failed) |
| 12 |         |      |                                |              | requesters_actual.csv |                               |           | 116/5,100 cases passed (10/10 trials failed) |

| # | Target | Type   | Host                           | Email Domain           | Username                         | Email                            | Blueprint | Test Failures                                  |
|---|--------|--------|--------------------------------|------------------------|----------------------------------|----------------------------------|-----------|------------------------------------------------|
| 1 | DII    | Script | http://dev1.dataninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com | first.last@docomoinnovations.com | Original  | 27/7,140 cases failed (14/14 trial set failed) |
| 2 |        |        |                                |                        | requesters.csv                   |                                  |           | 30/5,100 cases failed (9/10 trial set failed)  |

# 再度検証

- 97.9~99.9%の正解率
- 100%ではなかった・・・

| #  | Target  | Type | Host                         | Email Domain | Username              | Email                         | Blueprint | Test Failures                                | Test Result   | Test Data                | Date       | Reported by |
|----|---------|------|------------------------------|--------------|-----------------------|-------------------------------|-----------|----------------------------------------------|---------------|--------------------------|------------|-------------|
| 11 | Generic |      | http://dev1.daninja.net:8083 | @example.com | actual_first_last     | actual_first_last@example.com | Original  | 116/5,610 cases passed (11/11 trials failed) | 97.9% Success | approval_routes-8083.csv | 2026年5月29日 | @Yas Nao    |
| 12 |         |      |                              |              | requesters_actual.csv |                               |           | 116/5,100 cases passed (10/10 trials failed) | 97.8% Success | requesters_actual.csv    | 2026年6月1日  | @Yas Nao    |

| # | Target | Type   | Host                         | Email Domain           | Username                         | Email                            | Blueprint | Test Failures                                  | Test Result   | Test Data                | Date       | Reported by |
|---|--------|--------|------------------------------|------------------------|----------------------------------|----------------------------------|-----------|------------------------------------------------|---------------|--------------------------|------------|-------------|
| 1 | DII    | Script | http://dev1.daninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com | first.last@docomoinnovations.com | Original  | 27/7,140 cases failed (14/14 trial set failed) | 99.6% Success | approval_routes-8081.csv | 2026年5月29日 | @Yas Nao    |
| 2 |        |        |                              |                        | requesters.csv                   |                                  |           | 30/5,100 cases failed (9/10 trial set failed)  | 99.4% Success | requesters.csv           | 2026年6月1日  | @Yas Nao    |

# 再度検証

- 97.9~99.9%の正解率
- 100%ではなかった・・・

| #  | Target  | Type | Host                         | Email Domain | Username              | Email                         | Blueprint | Test Failures                                | Test Result        | Test Data                | Date       | Reported by |
|----|---------|------|------------------------------|--------------|-----------------------|-------------------------------|-----------|----------------------------------------------|--------------------|--------------------------|------------|-------------|
| 11 | Generic |      | http://dev1.daninja.net:8083 | @example.com | actual_first_last     | actual_first_last@example.com | Original  | 116/5,610 cases passed (11/11 trials failed) | ✗<br>97.9% Success | approval_routes-8083.csv | 2026年5月29日 | @Yas Nao    |
| 12 |         |      |                              |              | requesters_actual.csv |                               |           | 116/5,100 cases passed (10/10 trials failed) | ✗<br>97.8% Success | requesters_actual.csv    | 2026年6月1日  | @Yas Nao    |

| # | Target | Type   | Host                         | Email Domain           | Username                         | Email                            | Blueprint | Test Failures                                  | Test Result        | Test Data                | Date       | Reported by |
|---|--------|--------|------------------------------|------------------------|----------------------------------|----------------------------------|-----------|------------------------------------------------|--------------------|--------------------------|------------|-------------|
| 1 | DII    | Script | http://dev1.daninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com | first.last@docomoinnovations.com | Original  | 27/7,140 cases failed (14/14 trial set failed) | ✗<br>99.6% Success | approval_routes-8081.csv | 2026年5月29日 | @Yas Nao    |
| 2 |        |        |                              |                        | requesters.csv                   |                                  |           | 30/5,100 cases failed (9/10 trial set failed)  | ✗<br>99.4% Success | requesters.csv           | 2026年6月1日  | @Yas Nao    |

# 再度検証

- 97.9~99.9%の正解率
- 100%ではなかった... 🤖 🤖 🤖

| #  | Target  | Type | Host                         | Email Domain | Username              | Email                         | Blueprint | Test Failures                                | Test Result        | Test Data                | Date       | Reported by |
|----|---------|------|------------------------------|--------------|-----------------------|-------------------------------|-----------|----------------------------------------------|--------------------|--------------------------|------------|-------------|
| 11 | Generic |      | http://dev1.daninja.net:8083 | @example.com | actual_first_last     | actual_first_last@example.com | Original  | 116/5,610 cases passed (11/11 trials failed) | ✗<br>97.9% Success | approval_routes-8083.csv | 2026年5月29日 | @Yas Nao    |
| 12 |         |      |                              |              | requesters_actual.csv |                               |           | 116/5,100 cases passed (10/10 trials failed) | ✗<br>97.8% Success | requesters_actual.csv    | 2026年6月1日  | @Yas Nao    |

| # | Target | Type   | Host                         | Email Domain           | Username                         | Email                            | Blueprint | Test Failures                                 | Test Result        | Test Data                | Date       | Reported by |
|---|--------|--------|------------------------------|------------------------|----------------------------------|----------------------------------|-----------|-----------------------------------------------|--------------------|--------------------------|------------|-------------|
| 1 | DII    | Script | http://dev1.daninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com | first.last@docomoinnovations.com | Original  | 27/7140 cases failed (14/14 trial set failed) | ✗<br>99.6% Success | approval_routes-8081.csv | 2026年5月29日 | @Yas Nao    |
| 2 |        |        |                              |                        | requesters.csv                   |                                  |           | 30/5,100 cases failed (9/10 trial set failed) | ✗<br>99.4% Success | requesters.csv           | 2026年6月1日  | @Yas Nao    |

# 結果



- 100%正解率を達成 🎉
- テストケース、テスト回数を増やすほど信頼度向上

| #  | Target  | Type | Host                         | Email Domain | Username                                | Email                                  | Blueprint | Test Failures                                | Test Result   | Test Data                                         | Date       | Reported by |
|----|---------|------|------------------------------|--------------|-----------------------------------------|----------------------------------------|-----------|----------------------------------------------|---------------|---------------------------------------------------|------------|-------------|
| 11 | Generic |      | http://dev1.daninja.net:8083 | @example.com | actual_first_last_requesters_actual.csv | actual_first_last_requesters_email.com | Original  | 116/5,610 cases passed (11/11 trials failed) | 97.9% Success | approval_routes-8083.csv<br>requesters_actual.csv | 2026年5月29日 | @Yas Naoi   |
| 12 |         |      |                              |              |                                         |                                        |           | 116/5,100 cases passed (10/10 trials failed) | 97.8% Success |                                                   | 2026年6月1日  | @Yas Naoi   |

| # | Target | Type   | Host                         | Email Domain           | Username                                           | Email                            | Blueprint | Test Failures                                  | Test Result   | Test Data                                                 | Date       | Reported by |
|---|--------|--------|------------------------------|------------------------|----------------------------------------------------|----------------------------------|-----------|------------------------------------------------|---------------|-----------------------------------------------------------|------------|-------------|
| 1 | DII    | Script | http://dev1.daninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com<br>requesters.csv | first.last@docomoinnovations.com | Original  | 27/7,140 cases failed (14/14 trial set failed) | 99.6% Success | approval_routes-8081.csv<br>requesters.csv                | 2026年5月29日 | @Yas Naoi   |
| 2 |        |        |                              |                        |                                                    |                                  |           | 30/5,100 cases failed (9/10 trial set failed)  | 99.4% Success |                                                           | 2026年6月1日  | @Yas Naoi   |
| 3 |        |        |                              |                        |                                                    |                                  | Local     | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  | 0/5,100 cases failed (0/10 trials failed)<br>100% Success | 2026年5月29日 | @Yas Naoi   |
| 4 |        |        |                              |                        |                                                    |                                  | New       | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  |                                                           | 2026年5月29日 | @Yas Naoi   |
| 5 |        |        |                              |                        |                                                    |                                  |           | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  |                                                           | 2026年6月1日  | @Yas Naoi   |



# 結果



- 100%正解率を達成 🎉
- テストケース、テスト回数を増やすほど信頼度向上

| #  | Target  | Type   | Host                         | Email Domain           | Username                                           | Email                                  | Blueprint | Test Failures                                  | Test Result   | Test Data                                         | Date       | Reported by |
|----|---------|--------|------------------------------|------------------------|----------------------------------------------------|----------------------------------------|-----------|------------------------------------------------|---------------|---------------------------------------------------|------------|-------------|
| 11 | Generic |        | http://dev1.daninja.net:8083 | @example.com           | actual_first_last_requesters_actual.csv            | actual_first_last_requesters_email.com | Original  | 116/5,610 cases passed (11/11 trials failed)   | 97.9% Success | approval_routes-8083.csv<br>requesters_actual.csv | 2026年5月29日 | @Yas Naoi   |
| 12 |         |        |                              |                        |                                                    |                                        |           | 116/5,100 cases passed (10/10 trials failed)   | 97.8% Success |                                                   | 2026年6月1日  | @Yas Naoi   |
| 1  | DII     | Script | http://dev1.daninja.net:8081 | @docomoinnovations.com | first.last@docomoinnovations.com<br>requesters.csv | first.last@docomoinnovations.com       | Original  | 27/7,140 cases failed (14/14 trial set failed) | 99.6% Success | approval_routes-8081.csv<br>requesters.csv        | 2026年5月29日 | @Yas Naoi   |
| 2  |         |        |                              |                        |                                                    |                                        |           | 30/5,100 cases failed (9/10 trial set failed)  | 99.4% Success |                                                   | 2026年6月1日  | @Yas Naoi   |
| 3  |         |        |                              |                        |                                                    |                                        | Local     | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  |                                                   | 2026年5月29日 | @Yas Naoi   |
| 4  |         |        |                              |                        |                                                    |                                        | New       | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  |                                                   | 2026年5月29日 | @Yas Naoi   |
| 5  |         |        |                              |                        |                                                    |                                        |           | 0/5,100 cases failed (0/10 trials failed)      | 100% Success  |                                                   | 2026年6月1日  | @Yas Naoi   |

# まとめ



# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」

# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」

# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」

# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」

# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」

# まとめ



- POP の可能性:
  1. 業務ワークフローの実現
  2. 検証可能な仕組みの構築
  3. 要件定義の重要性
- 「どう実現するか（How）」よりも  
「業務ルールをどう言語化するか（What）」



# まとめ



- POP の利点:
  - 仕様変更へのフィードバックが速い
  - 様々な入出力を柔軟に扱える
  - 開発と運用における効率化が期待できる
- 次世代のエンタープライズシステムに向けて、  
再利用可能なデザインパターンを標準的な  
プログラミングフレームワークとして確立

# まとめ



- POP の利点:
  - 仕様変更へのフィードバックが速い
  - 様々な入出力を柔軟に扱える
  - 開発と運用における効率化が期待できる
- 次世代のエンタープライズシステムに向けて、  
再利用可能なデザインパターンを標準的な  
プログラミングフレームワークとして確立

# まとめ



- POP の利点:
  - 仕様変更へのフィードバックが速い
  - 様々な入出力を柔軟に扱える
  - 開発と運用における効率化が期待できる
- 次世代のエンタープライズシステムに向けて、  
再利用可能なデザインパターンを標準的な  
プログラミングフレームワークとして確立

# まとめ



- POP の利点:
  - 仕様変更へのフィードバックが速い
  - 様々な入出力を柔軟に扱える
  - 開発と運用における効率化が期待できる
- 次世代のエンタープライズシステムに向けて、  
再利用可能なデザインパターンを標準的な  
プログラミングフレームワークとして確立

# まとめ



- POP の利点:
  - 仕様変更へのフィードバックが速い
  - 様々な入出力を柔軟に扱える
  - 開発と運用における効率化が期待できる
- 次世代のエンタープライズシステムに向けて、  
再利用可能なデザインパターンを標準的な  
プログラミングフレームワークとして確立

# 学び



**AI を信じるな、テストせよ。**



# 学び



AI を信じるな、**テストせよ。**



# Thank you!



*"Coming together is a beginning, keeping together is progress; working together is success."*

*- Henry Ford*





