No.11 ノウハウ・全10ページ

AI 機能・外部 API にも業務ルールを

操作に権限・承認・上限を組み込む設計



要求を受け、権限・承認・実行条件を照合し、許可または拒否する。
導入時の検証項目つき

Securiy

DESIGN

この資料でわかること

権限・承認・上限は、どの業務アプリにもあります。抜けるのは、画面以外の経路（API・AI 機能・外部連携）が増えたときです。この資料は、経路が増えても抜けない置き方を、既存のコードに組み込める粒度で扱います。

想定読者

データ出力・取引・設定変更を扱うアプリの開発責任者

チャット機能や外部連携 API を持つ、または持とうとしている SaaS・Fintech・EC 事業者。

扱うこと

照合をどこに置くか、3つの照合、AI・外部連携、例外、検証

設計の原則と擬似コード、権限・承認・上限それぞれの書き方、AI 機能と外部連携の扱い、例外の通し方、導入時に試す10項目。

扱わないこと

特定の言語・フレームワークの実装、ログインの方式

パスワード・MFA・SSO といった認証（ログイン）の仕組みは対象外です。ログインした後の話だけを扱います。

1 どこに置くか

経路は4本、照合は1カ所 P.03

2 3つの照合

権限: 誰が・何を（連携キーの範囲を含む） P.04

承認: 承認した内容と実行内容の一致 P.05

上限: 金額・件数・時間帯・回数 P.06

3 経路と運用

AI 機能・外部連携の照合 P.07

例外の扱いと記録 P.08

導入時の検証項目 P.09

経路は4本、照合は1カ所

同じ処理を呼ぶ経路は、画面のほかに API・AI 機能・外部連携（連携キー＝外部のツールがアプリの API を呼ぶためのキー）・バッチがあります。確認は経路ではなく、業務処理を呼ぶ共通の入口に1カ所だけ置きます。画面のチェックは利便性のために残してよいですが、守りには数えません。



画面にしか確認がないと、どう抜けるか

画面は送る前に承認状態や上限を確かめているが、サーバーは「送られてきた内容」をそのまま実行する。
API は画面と同じ処理を直接呼ぶので、画面の確認を通らない。AI 機能は利用者の代わりに呼ぶので、誰の権限か曖昧になる。
連携キーは作った人の権限をそのまま持ち、期限や範囲が付いていないことがある。
経路が増えるたびに、確認が1つずつ抜ける。

```
// 業務処理を呼ぶ共通の入口（擬似コード）
execute(誰が, 操作, 対象, 内容):
  権限を照合(誰が, 操作, 対象)          // 所属・ロール・キーの範囲
  承認を照合(操作, 内容, 承認の記録)  // サーバーに保存した承認と比較
  上限を照合(誰が, 操作, 内容)          // 金額・件数・回数・時間帯
  どれかが不一致 → 拒否(理由) → 記録
  実行(操作, 内容) → 記録(誰が, 経路, 結果, 根拠)
```

原則 1

判定の仕組みは1つにまとめ、全経路を通す。サービスが分かれていても、判定の規則は1つにする。DB の行単位の制限などを重ねるのは構いません。

原則 2

判定の根拠になる値（所属組織・承認状態・上限）は、要求から受け取らない。要求に入っている「承認済み」フラグは信用しない。

権限: 誰が・何を（連携キーの範囲を含む）

ロールごとに「できる操作」を表にします。閲覧・出力・検索・権限変更・連携キーは別の操作として分けるのがコツです。
「閲覧できる＝出力できる」にすると、一括取得の抜けが生まれます。

操作	誰ができるか（例）	照合で確かめること
顧客データの閲覧	一般: 自組織のみ ／ 組織管理者: 自組織 ／ 連携キー: 指定した組織のみ	**要求の組織 ID と、利用者（またはキー）の所属組織が一致すること**
顧客データの出力（CSV・API 一括）	一般: 担当分のみ ／ 組織管理者: 自組織 ／ 連携キー: 明示的に許可した場合のみ	**出力は閲覧と別の操作として扱う。件数の上限と記録を伴う**
検索	閲覧と同じ	**検索結果が組織をまたがないこと（検索 API は抜けやすい）**
権限の変更	組織管理者のみ。自分自身の引き上げは不可	**変更の履歴（誰が・誰を・何に）が残ること**
連携キーの発行と範囲	組織管理者のみ。キーに組織・操作・期限を付ける	**範囲外の組織・操作、期限切れのキーからの要求を拒否すること**

表の書き方と、取引先に示す1枚にする方法は No.12 P.05 に。設計の根拠は IPA「安全なウェブサイトの作り方」1.11、OWASP ASVS V8、OWASP Authorization Cheat Sheet。判定の仕組みには OpenFGA などの認可サービスも使えますが、どれを使っても「全経路で呼ばれているか」の検証は要ります。

2 ／ 3つの照合

承認: 承認した内容と、実行された内容の一致

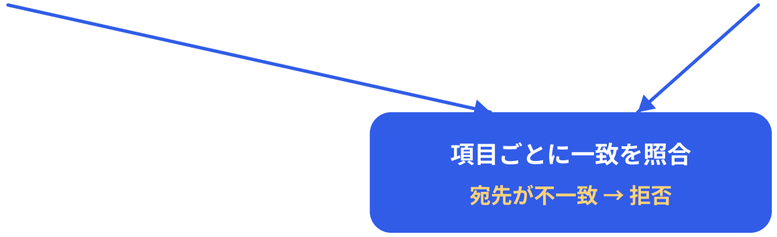
承認フローがあることと、承認どおりに実行されることは別です。実行の直前に、サーバーが保持している承認の記録と、要求の内容を項目ごとに比べます。

承認した内容（サーバーが保持）

申請 ID	**REQ-2041**
内容	**送金先の変更**
宛先	**株式会社A ／ 口座 ****1234**
承認者・日時	**部長 ／ 10/07 14:02**

実行の要求（受け取った内容）

申請 ID	**REQ-2041**
内容	**送金先の変更**
宛先	**株式会社A ／ 口座 ****9876**
承認済みフラグ	**true（要求に含まれていた）**



設計の要点

1. 承認状態はサーバー側で持つ。
要求に含まれる「承認済み」は照合に使わない。

2. 承認した内容を保存し、実行時に項目ごとに比べる。
金額・宛先・日付・件数。1つでも違えば実行しない。

3. 承認後の書き換えは、承認を無効にする。
内容を変えたら再承認。承認後の書き換えをここで止める。

4. 申請者と承認者を分ける。
自分で申請し自分で承認できる設定を作らない。

AI 機能・外部連携からの実行も同じ照合を通す。

承認フローが画面にあっても、「承認した内容」と「実行された内容」をサーバーで比べていなければ、承認は迂回できます。

比べる相手は、要求ではなくサーバーに保存した承認の記録です。

上限: 金額・件数・時間帯・回数

権限と承認が正しくても、1回の操作で動く量に上限がなければ、事故の大きさを抑えられません。上限は「止める」のではなく「止めて申請に回す」仕組みとして置きます。連携キーの範囲は権限（P.04）で扱います。

上限の種類	何に掛けるか	例	超えたときの扱い
金額	1件あたり・1日あたりの上限	返金 1件 ¥100,000 まで、1日 ¥500,000 まで	**超えたら停止し、申請に回す**
件数	一括処理の件数、出力の行数	CSV 出力 1回 1,000 行まで	**超える分は分割か、管理者の承認**
時間帯	実行できる時間	送金の実行は平日 9〜18時	**時間外は保留し、翌営業日に**
回数・速度	同じ操作の繰り返し	同一利用者の出力は 1時間に 10 回まで	**超えたら一時停止し、通知**

値の決め方

例の数値は説明用です。自社の業務で「これを超えたら誰かが確認すべき」と感じる値から始め、申請が多すぎれば上げます。

現場が回らなくなる心配

上限は拒否ではなく申請への分岐です。申請がすぐ承認される経路（P.08 の例外）を用意しておけば、現場は止まりません。

AI 機能・外部連携の照合:「誰として」判定するか

AI 機能は利用者の代理として、その利用者の権限で動かします。外部連携のキーには範囲と期限を付けます。どちらも P.03 の同じ照合を通すだけで、新しい仕組みは要りません。照合が止めるのは利用者の権限の外の操作だけで、権限の内側の被害（本人が見られるデータの外部送信、本人が出せる申請）は、申請と人の承認・上限で絞ります（No.27）。

AI 機能（チャット・自動化）



AI 機能の設計

- 利用者の代理として、その利用者の権限で動かす
- AI 専用の管理者アカウントを作らない
- AI が呼べる操作を一覧で限定する
- 出力・送金・権限変更は申請まで。実行は人が承認
- 指示文の中身で権限を判断しない（誰の指示かで判断）

外部連携（API キー・OAuth）



外部連携の設計

- キーに組織・操作・期限を付けて発行する
- 発行者の権限を丸ごと引き継がせない
- 一括出力・送金はキーからは申請まで
- 期限切れは自動で止め、更新を記録する
- 退職・解約時に無効化する手順を決めておく

AI 機能も外部連携も、新しい照合を作るのではなく、P.03 の同じ入口を通します。違いは「誰として」判定するか（利用者の代理か、キーの範囲か）だけです。

例外の扱いと記録

緊急時の迂回を想定していない設計は、現場で「万能キー」を生みます。例外は申請・別の人の承認・期限・記録の4つを付けて通します。



例外で崩れる典型

「緊急だから」で管理者が上限を外し、戻し忘れる。

例外用の「万能キー」を作り、全員が使い始める。

AI 機能に「管理者として実行」の権限を与え、誰の操作か分からなくなる。

例外が記録されず、取引先の「承認を経ていますか」に答えられない。

例外をなくすのではなく、例外にも期限と記録を付ける。
これで「例外だらけ」を防げます。

記録に残す項目（1操作1行）

いつ	日時
誰が	利用者 ／ 連携キー ／ AI 機能（誰の代理か）
どの経路で	画面 ／ API ／ AI ／ バッチ
何を	操作と対象（組織・金額・件数）
結果	許可 ／ 拒否 ／ 例外で許可
根拠	照合した権限・承認 ID・上限の値
例外なら	申請者・承認者・期限

この記録がそのまま、取引先への「承認を経ているか」「変更時に確認しているか」の根拠になります。

例外は起きる前提で設計します。決めるのは「例外をどう通すか」と「どう戻すか」です。

導入時の検証項目: 10 の操作を試す

設計を入れたら、正当なアカウントで範囲外の操作を試し、期待どおりになることを版・日付つきで記録します。リリースのたびに同じ10行を試せば、再検証の記録になります。試し方は No.09 P.08〜09 に。

No.	試すこと	照合	結果	版・日付
01	**画面で拒否される操作が、API に直接送っても拒否される**	権限	期待どおり ／ 違う	
02	**他組織の ID を指定した閲覧・出力・検索が拒否される**	権限	期待どおり ／ 違う	
03	**一般ユーザーが自分の権限を引き上げられない**	権限	期待どおり ／ 違う	
04	**連携キーで、範囲外の組織・操作・期限切れが拒否される**	権限	期待どおり ／ 違う	
05	**要求に「承認済み」を入れても、サーバーの記録と不一致なら拒否される**	承認	期待どおり ／ 違う	
06	**承認後に内容を書き換えると、再承認が要る**	承認	期待どおり ／ 違う	
07	**申請者が自分で承認できない**	承認	期待どおり ／ 違う	
08	**上限を 1 単位だけ超える要求が停止し、申請に回る**	上限	期待どおり ／ 違う	
09	**AI 機能・バッチからの要求が、利用者の権限で同じ照合を通る**	経路	期待どおり ／ 違う	
10	**許可・拒否・例外が、誰が・経路・何を・結果・根拠つきで記録される**	記録	期待どおり ／ 違う	

この10項目は OWASP ASVS 5.0 の V8（認可）・V2.3（業務ロジック）に対応します。2つのアカウントで同じ要求を送り比べる作業は、OWASP ZAP の Access Control Testing や Burp Suite の Autorize で半自動にできます。

設計は読めば作れる。
試すところを、一緒に。

この資料の設計を実装し、P.09 の10項目を決めた条件で画面と API から試すところまでは自社でできます（No.09 P.09）。外に出す価値があるのは、まだ決めていない条件と経路（承認内容の書き換え・連携キー・AI 機能）を広く洗い出して試し、試さなかったものも未検証範囲として書いた、取引先に出せる記録にするところです。共同検証プログラムでは、10項目のうち2操作から始めます。初回のご相談は30分・無料です。

お申し込み・お問い合わせ　**securiy-preview.pages.dev**　のフォームから
Tempura technologies株式会社

※ 本資料の例（数値・ロール）は説明用です。すべての攻撃の網羅や「安全」の総合判定は保証しません。金融の規制適合審査、決済基盤全体の監査、カード不正判定は対象外です。

無料相談

30分無料相談

いまの照合がどこにあるか（画面か、サーバーか）を伺い、初回に検証する2操作を一緒に決めます。本番の認証情報や顧客データは不要です。

相談を申し込む →

共同検証プログラムとは

検証テーマを一緒に作る企業に、初回スコープで提供する枠組み

1アプリ・1検証環境・2操作、準備完了後10営業日・合計40時間（修正案と再検証は2件・12時間まで）。費用は着手前に提示します。PROTECT（この設計を仕組みにする SDK）は設計・検証段階で、提供時期は未定です。

次に読む

No.19 ゼロデイ前提の不正操作対策
No.12 エンプラ顧客に提供するための3つの観点

未知の脆弱性でも被害を小さくできる理由と、取引先に示す1枚の作り方。