API × MCP

生成AI・業務システム連携 実践ガイド



Start Challenge Consulting co., ltd.
挑む力が、世界を照らす

Start Challenge Consulting co., ltd.
API・MCP連携／生成AI業務実装 支援

2026年8月 発行

Confidential

生成AIを「回答するAI」から「業務を実行するAI」へ

生成AIは単体では社内データや業務システムを利用できない。従来のAPIによる個別連携に加え、AIと外部ツールの接続を共通化するMCPが登場し、両者を役割で使い分ける段階に入っている。

生成AI活用の発展段階

STEP 1　回答するAI

学習済み知識の範囲で文章生成・要約・分類を行う。社内データは参照できない。

↓

STEP 2　外部接続の追加

APIによる個別連携、MCPによる共通接続で、データと機能をAIに提示する。

↓

STEP 3　業務を実行するAI

検索・参照・整理に加え、承認を前提とした登録・更新まで業務に関与する。

本書の前提

- 生成AIを業務で利用する企業が増えている
- AI単体では、社内データや業務システムを自由に利用できない
- 従来はAPIを使った個別連携が中心だった
- MCPにより、AIと外部ツールを接続するための共通化が進んでいる
- APIとMCPは競合関係ではなく、役割の異なる仕組みである
- 本書では、構成・使い分け・実装方法までを整理する

注記／事実：MCPはAIアプリケーションと外部システムを接続するためのプロトコルである。解釈：APIとMCPは役割が異なり、併用が前提となる。

生成AIを導入しても、業務データとつながらない

生成AIと社内データ・業務システムが分断されているため、検索・参照・登録が人手に依存する。接続先ごとに異なるAPI仕様と権限管理が、個別開発の増加と統制上の不安を招いている。

生成AI
要約・分類・文章生成

× 接続手段が整っていない

社内データ
規程・マニュアル・議事録

× 仕様・権限が個別に異なる

業務システム
CRM・SFA・会計・BI

現場で挙がる主な課題

01 データ参照
AIから社内データを検索できない

02 顧客情報
CRMやSFAの情報をAIが参照できない

03 登録・更新
AIの回答を業務システムへ登録できない

04 仕様の差異
システムごとにAPI仕様が異なる

05 開発負荷
接続先が増えるほど個別開発が増える

06 認証・権限
認証・アクセス権限の管理が複雑になる

07 操作範囲
AIにどこまで操作させてよいか判断できない

08 リスク不安
誤操作や情報漏えいへの不安がある

注記／事実：AIは接続手段がなければ外部データを参照できない。解釈：課題は「接続」「開発負荷」「統制」の3層に整理できる。

システム同士がデータや機能をやり取りするための接続口

APIはリクエストとレスポンスにより、データの取得・登録・更新を行う標準的な接続口である。実行する処理と条件は、あらかじめプログラムやワークフローで定義される。

押さえておくべき基本要素

API	機能やデータを外部から利用するための接続口
リクエスト／レスポンス	要求を送り、結果を受け取る往復の通信
エンドポイント	要求の宛先となるURL単位の接続先
HTTP／HTTPS	通信の方式。HTTPSは暗号化通信
JSON	やり取りに用いられる代表的なデータ形式
HTTPメソッド	GET・POST・PUT・PATCH・DELETE
認証方式	APIキー、OAuthなどによる認可

API連携の具体例

- CRMから顧客情報を取得する
- 問い合わせ内容をCRMへ登録する
- OpenAI APIへ文章を送り、要約結果を取得する
- GA4からアクセスデータを取得する
- TeamsやSlackへ通知する

POINT　APIでは「どの処理を、どの条件で、どのように実行するか」を、あらかじめプログラムやワークフローで定義する。

注記／事実：APIはHTTP等の通信方式とJSON等のデータ形式を用い、認証を経て処理を実行する。

APIは、事前に定義された処理を正確に実行する

起点・認証・実行・結果返却までが定義済みの手順で進むため、定型業務では高い再現性と処理速度を確保できる。一方で、接続先ごとに個別の実装が必要となる。

基本フロー

01 システムやユーザーが処理を開始 → 02 接続先APIへリクエストを送信 → 03 API側で認証・権限を確認 → 04 データ取得・登録・更新などを実行 → 05 処理結果を呼び出し元へ返却 → 06 呼び出し元で結果を表示・保存

構成例



レスポンス例（JSON）

```
{
  "customer_id": "C-001",
  "company_name": "サンプル株式会社",
  "status": "商談中"
}
```

項目名・値の形式は接続先のAPI仕様に依存する。

注記／事実：API処理は認証・実行・返却の手順で構成される。解釈：処理内容が固定されるため再現性が高い。

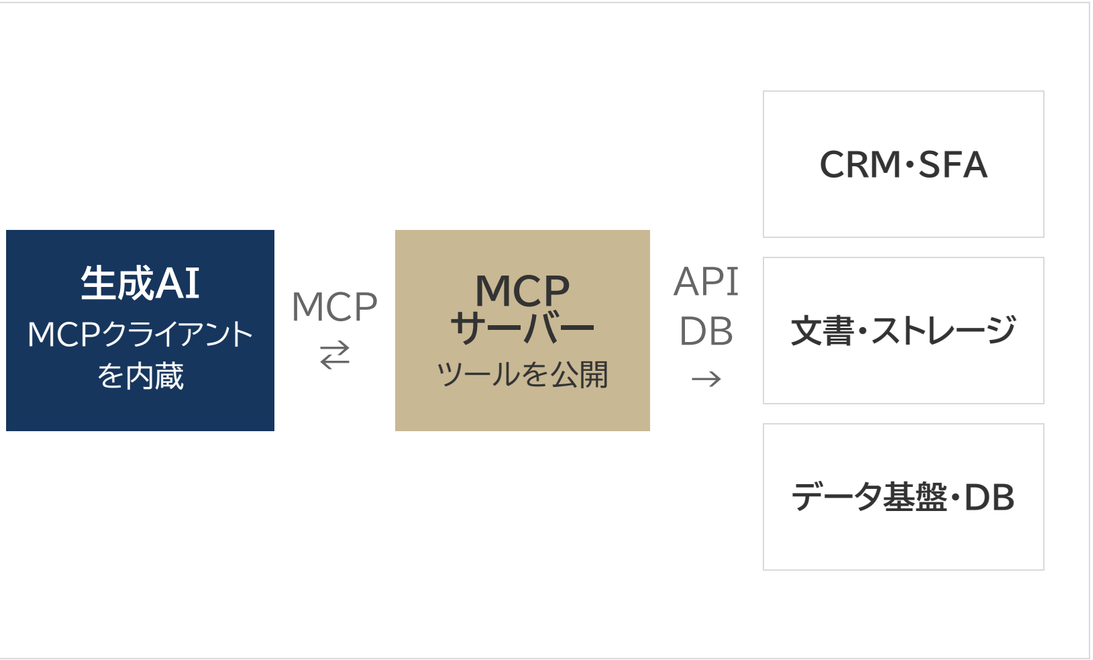
AIが外部のデータやツールを利用するための共通接続方式

MCPは利用可能なデータや機能をAIに提示し、AIが依頼内容に応じて必要なツールを選択できるようにする。接続先ごとの仕様差はMCPサーバー側で吸収される。

MCPの基本理解

- MCPは「Model Context Protocol」の略称
- AIアプリケーションと外部システムを接続するためのプロトコル
- AIに利用可能なデータや機能を提示できる
- AIがユーザーの依頼に応じて必要なツールを選択する
- 接続先ごとの違いをMCPサーバー側で吸収できる
- AIエージェントの検索・参照・処理実行に活用できる

接続イメージ



POINT　MCP自体が顧客情報や業務データを保有するとは限らない。AIと外部データ・業務機能をつなぐための接続方式である。

注記／事実：MCPはAIアプリと外部システムを接続するプロトコルである。解釈：データ保有主体は接続先システム側にある。

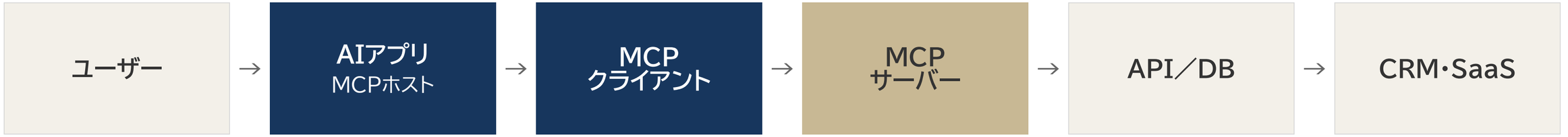
MCPホスト・MCPクライアント・MCPサーバーの関係

MCP連携は、AIアプリ側の「ホスト」と「クライアント」、機能を公開する「MCPサーバー」、実データを保有する外部システムの分担によって成立する。

構成要素と役割

構成要素	主な役割
MCPホスト	AIアプリケーション全体を動かす環境
MCPクライアント	MCPサーバーとの接続・通信を管理する
MCPサーバー	AIに利用可能なデータやツールを公開する
外部システム	実際の業務データや機能を保有する
API・DB	MCPサーバーと外部システムを接続する

構成例



補足：MCPクライアントは、通常、ChatGPTやAIエージェントなどのAIアプリケーション側に組み込まれる。

注記／事実：MCPはホスト・クライアント・サーバーの構成をとる。解釈：業務データの保有主体は外部システム側にある。

連携する双方がMCPサーバーを持つ必要はない

MCP連携に必要なのは「MCPクライアント」と「MCPサーバー」の組み合わせである。接続先システムがMCP非対応でも、APIやDBの接続手段があればMCPサーバー側から連携できる。

一般的な構成と必要な機能

連携する側	必要な機能	役割
AIアプリケーション側	MCPクライアント	MCPサーバーへ接続する
MCP連携部分	MCPサーバー	AIが利用できるツールを公開する
CRM・SaaS側	API・DBなど	実際のデータや処理機能を提供する

正しい構成（○）

AIアプリ MCPクライアント → MCP サーバー → CRM・SaaS API／DB

クライアントとサーバーが1組あれば連携は成立する。

誤解されやすい構成（×）

MCP サーバー（AI側） × MCP サーバー（業務側）

サーバー同士を直接接続する構成は基本形ではない。

双方がMCPサーバーを用意する必要はない。

具体例　Salesforce側がMCPサーバーを提供していなくても、Salesforce APIを呼び出すMCPサーバーを用意すれば連携できる。

ただし接続先にAPIやDBなどの接続手段は必要。公式MCPサーバーがある場合は、それを利用できることがある。

注記／事実：MCPはクライアント・サーバー型の接続方式である。解釈：AI側と業務システム側の双方にMCPサーバーは不要。

公式MCPを使う方法と、自社でMCPサーバーを構築する方法

用意方法は「公式MCPの利用」「自社構築」「MCP対応基盤の利用」の3パターンに整理できる。開発負荷・制御の自由度・運用責任のバランスで選定する。

パターン1　公式MCPサーバーの利用

- 接続先サービスがMCPサーバーを提供
- 独自開発の負担を抑えられる
- 公式の認証・権限方式を利用しやすい
- 提供される機能の範囲内で利用する

パターン2　自社でMCPサーバーを構築

- 既存のAPIやデータベースに接続する
- 自社業務に合わせたツールを定義できる
- 取得項目や更新可能範囲を制限できる
- 開発・運用・セキュリティ管理が必要

パターン3　MCP対応の外部基盤を利用

- 複数サービスとの接続を共通化
- 開発期間を短縮できる可能性がある
- 利用料金・権限・データ経路の確認が必要

選定の視点

開発負荷を抑える間➡パターン1・3	操作範囲を厳密に制御間➡パターン2	接続先が多い間➡パターン3

注記／解釈：3パターンは排他ではなく併存しうる。提案：接続先ごとに公式MCPの提供状況を確認した上で方式を選定する。

定型的なシステム連携と、AIを前提としたツール連携

APIは実際の処理を実行し、MCPはAIが利用できる処理を認識・選択する。両者は競合ではなく、MCPの裏側でAPIが動く構成が多い。

比較項目	API	MCP
主な目的	システム間のデータ・機能連携	AIとデータ・ツールの接続
処理の選択	プログラムで事前に指定	AIが依頼内容に応じて選択
主な利用者	アプリ・システム・ワークフロー	AIアプリ・AIエージェント
接続方法	API仕様ごとに個別実装	MCP形式で機能を公開
得意な処理	定型・反復・大量処理	対話型・探索型・複数ツール利用
実行の再現性	比較的高い	AI判断により変動する場合がある
制御方法	プログラム・入力チェック	権限・ツール制限・承認・ガードレール
主な用途	取得・登録・更新・削除	検索・参照・判断・ツール選択

結論

API
実際の処理を実行する

MCP
AIが利用できる処理を認識・選択する

併用が前提
MCPの裏側でAPIが動く構成が多い

注記／事実：APIは処理実行、MCPはツール提示の仕組みである。解釈：両者は代替関係ではなく補完関係にある。

AIがMCPでツールを選び、APIが実処理を行う

自然言語の依頼に対し、AIは利用可能なツールを確認して選択し、MCPサーバー経由でAPIを呼び出す。取得結果はAIが整理して回答する。

> 依頼例：**「先月問い合わせがあった企業を一覧にして」**

処理の流れ

STEP 01
ユーザーがAIへ自然言語で指示

STEP 02
AIが依頼内容を理解

STEP 03
MCPクライアントが利用可能なツールを確認

STEP 04
AIが顧客検索ツールを選択

STEP 05
MCPサーバーがCRM APIを呼び出す

STEP 06
CRMが顧客データを返却

STEP 07
MCPサーバーが結果をAIへ返す

STEP 08
AIが一覧を整理して回答

重要な整理　MCPがAPIを置き換えるのではなく、AIがAPIなどの外部機能を利用しやすくする関係である。

注記／解釈：ツール選択はAI、処理実行はAPIが担う。提案：更新系ではSTEP08の前に人による承認を挟む。

実行条件と処理内容が決まっている業務

処理条件が固定され、高い再現性と定時実行が求められる業務はAPI連携が適する。AIによる判断を必要としない領域では、API連携が最も安定した選択肢となる。

代表例

01 CRM登録
問い合わせフォームからCRMへの自動登録

02 通知連携
メール受信を起点としたTeams通知

03 在庫連携
ECサイトと在庫管理システムの連携

04 会計連携
会計システムへの請求データ登録

05 定期取得
GA4データの定期取得

06 BI連携
Power BIへのデータ連携

07 ステータス更新
顧客ステータスの自動更新

08 定期処理
日次・週次の定期処理

APIが適している判断基準

- ■ 処理条件が固定されている
- ■ 高い再現性が必要
- ■ 大量データを処理する
- ■ 決められた時間に自動実行する
- ■ AIによる判断が不要

注記／解釈：定型・反復・大量処理領域ではAPI連携の優位性が高い。提案：まず定型業務をAPIで整備し、その上でMCPを重ねる。

依頼内容に応じて、AIがデータやツールを選ぶ業務

依頼内容が毎回異なり、複数のデータソースを横断する業務ではMCP連携が適する。会話しながら検索・選択・整理をAIへ委ねられる点が価値となる。

代表例

01 横断検索
社内規程やマニュアルの横断検索

02 提案書作成
顧客情報を確認しながら提案書を作成

03 CRM更新候補
会議内容からCRMの更新候補を作成

04 横断分析
複数データソースを横断した分析

05 振り分け判定
問い合わせ内容に応じた担当部署の判定

06 事例検索
関連資料や過去事例の検索

07 エージェント活用
AIエージェントによる複数ツールの使い分け

MCPが適している判断基準

- ユーザーの依頼内容が毎回異なる
- 複数のツールやデータを利用する
- 会話しながら処理を進めたい
- AIに検索・選択・整理を任せたい

注記／解釈：探索型業務ではMCPの適合度が高い。提案：更新を伴う業務は「更新候補の作成」に留め、実行前に人が承認する。

生成AIからCRM・データ基盤・業務SaaSへ接続する

連携対象は生成AI・CRM・データ基盤・BI・ストレージ・自動化基盤まで広がる。製品名だけで可否を判断せず、API・認証・権限・公式MCPの提供状況を確認する必要がある。

分類	ソリューション例	主な連携内容
生成AI	ChatGPT、Azure OpenAI、Claude、Gemini	要約、分類、生成、ツール利用
CRM／SFA	Salesforce、HubSpot、Dynamics 365	顧客検索、商談登録、履歴更新
グループウェア	Microsoft 365、Google Workspace	メール、予定、ファイル、会議
コミュニケーション	Teams、Slack	通知、検索、メッセージ送信
データ基盤	Snowflake、Databricks、BigQuery	検索、集計、データ分析
BI	Power BI、Tableau、Sigma	レポート参照、分析結果出力
データベース	SQL Server、PostgreSQL、MySQL	データ検索、登録、更新
ストレージ	SharePoint、Google Drive、Box	ファイル検索、取得、保存
自動化基盤	Power Automate、Zapier、Make	ワークフロー、通知、承認
開発管理	GitHub、Jira、Confluence	コード、課題、文書の検索

注記　製品名だけでMCP連携の可否を判断せず、API・認証・権限・公式MCPの提供状況を個別に確認すること。

注記／事実：表中の製品は代表例であり網羅ではない。解釈：連携可否は各製品のAPI・認証仕様に依存する。

商談内容を整理し、CRMの更新候補を作成する

AIは文字起こしの要約からCRM項目に整形した更新候補までを作成し、登録実行は担当者の承認後にCRM APIが担う。
承認前の自動更新は行わない設計とする。

業務フロー

STEP 01
TeamsやZoomなどから文字起こしを取得

STEP 02
AIが商談内容を要約

STEP 03
顧客課題・提案内容・次回対応を抽出

STEP 04
MCP経由で既存の顧客情報を検索

STEP 05
CRM項目に合わせてデータを整形

STEP 06
担当者へ更新候補を表示

STEP 07
担当者が確認・修正（承認ゲート）

STEP 08
承認後にCRM APIで登録

設計上のポイント

- 顧客の特定方法
- CRM項目とのマッピング
- 必須項目のチェック
- 重複登録の防止
- AI生成内容の確認
- 承認前の自動更新禁止

注記／解釈：更新系連携ではAIの出力を「候補」に限定する。提案：承認ゲートをSTEP07に置き、登録はAPI実行に限定する。

MCPで検索機能を利用し、RAGで根拠のある回答を生成する

MCPは検索・取得ツールをAIに認識させ、RAGは取得結果を根拠として回答を生成する。両者は代替関係ではなく、組み合わせて利用する構成となる。

対象データ

- SharePoint
- Google Drive
- Box
- ファイルサーバー
- 社内データベース
- Confluence
- FAQ・マニュアル・社内規程

役割の違い

技術	主な役割
MCP	AIが検索・取得ツールを認識して利用する
RAG	検索結果を根拠として回答を生成する
API	ファイルや権限情報を取得する
ベクトル検索	質問と意味的に近い文書を探す
ガードレール	閲覧範囲や出力内容を制御する

重要ポイント　MCPとRAGは代替関係ではなく、組み合わせて利用できる。MCPが「取りに行く手段」、RAGが「根拠づける仕組み」を担う。

注記／解釈：MCPは接続方式、RAGは回答生成手法であり階層が異なる。提案：閲覧権限をユーザー単位で継承させる設計とする。

SnowflakeやBIデータを自然言語で分析する

AIがテーブル定義を確認しSQL案を生成する構成では、読み取り専用アカウントと接続先制限が前提となる。制御なしの直接実行は避ける必要がある。

質問例

- 前週よりサイト訪問数が減った原因を調べて
- 問い合わせにつながった流入元を比較して
- 売上が低下した顧客層を抽出して
- 分析結果をTeamsへ投稿して

処理イメージ

STEP 01
AIが質問を理解

STEP 02
MCP経由でテーブル定義や指標を確認

STEP 03
AIがSQL案を生成

STEP 04
許可された環境でSQLを実行

STEP 05
AIが結果を分析

STEP 06
BIやTeamsへ結果を出力

必要な制御

- 読み取り専用アカウント
- 接続可能なテーブルの制限
- SQL実行時間・取得件数の上限
- 個人情報のマスキング
- SQL実行ログの保存
- 不正・危険なSQLの拒否

注記／解釈：SQL生成をAIに委ねる構成は実行環境側の制御が前提となる。提案：更新権限を持たない専用アカウントで運用する。

対象業務とシステム環境から連携方式を選ぶ

連携方式は業務の定型性・システム側の接続手段・リスクの大きさで決まる。単一方式に統一するのではなく、業務特性に応じて組み合わせることが前提となる。

連携方法	適しているケース	主な注意点
API	定型処理、高速・安定した連携	個別実装が必要
MCP	AIによる検索・判断・ツール利用	AIの操作範囲を制限する
RPA	APIがないシステムの画面操作	画面変更の影響を受けやすい
iPaaS	ノーコード・ローコード連携	複雑な処理には限界がある
DB直接接続	大量データの検索・集計	更新権限と負荷に注意
ファイル連携	CSVなどによる一括処理	リアルタイム性が低い

選定基準

定型業務	会話型・探索型業務	APIがない画面操作	大量データ分析	高リスクな更新
API・iPaaS	MCP	RPA	DB・データ基盤	承認付きAPI連携

注記／解釈：方式選定は業務の定型性と接続手段の有無で決まる。提案：高リスク更新は承認付きAPI連携に限定する。

参照系の小規模PoCから段階的に導入する

導入は「検索・参照」から始め、更新系は人による承認を経て条件付き自動実行へ拡張する。最小権限とログ・監視の整備が前提条件となる。

実装手順

01　対象業務と利用者を決める
02　AIに利用させたいデータ・機能を整理
03　接続先のAPI・DB・認証方式を確認
04　公式MCPの有無を確認
05　MCPサーバーを利用または構築
06　入出力項目と権限を設計
07　正常系・異常系をテスト
08　ログ・監視・承認を設定
09　限定部門でPoCを実施
10　効果測定後に対象業務を拡張

推奨する導入順序

検索・参照 → 下書き作成 → 更新候補作成 → 人による承認 → 条件付き自動実行

必須のセキュリティ対策

最小権限	読み取りと更新権限の分離
個人情報・機密情報のマスキング	APIキーやトークンの安全な管理
操作ログ・監査ログの保存	プロンプトインジェクション対策
重要操作への人の承認	誤動作時の停止機能

更新系連携では、AI出力の妥当性確認と承認記録の保存を運用設計に含める。

導入ロードマップ（想定：全6か月・所要期間は仮定値）



注記／仮説：ロードマップの所要期間は6か月想定（仮定値）。提案：参照系PoCで効果測定後に更新系へ拡張する。

APIとMCPを組み合わせ、AIが安全に業務へ参加できる環境をつくる

APIは処理の実行、MCPはAIによるツール利用を担う。双方にMCPサーバーは不要であり、参照系から段階的に導入し、更新系は承認とガードレールで統制する。

構造の整理

APIはシステム間でデータや機能を連携する仕組み

MCPはAIが外部データやツールを利用するための共通接続方式

AI側にはMCPクライアント、連携機能側にはMCPサーバーが必要

連携する双方がMCPサーバーを持つ必要はない

接続先がMCP非対応でも、APIなどを利用してMCPサーバーから接続できる

導入判断の要点

MCPサーバー同士を接続することが基本構成ではない

MCPの裏側でAPIが実際の取得・登録・更新処理を担うことが多い

API、MCP、RAG、iPaaSは目的に応じて組み合わせる

導入は参照系から始め、更新系には承認とガードレールを設ける

CONTACT

自社の業務システムにAPI・MCPをどのように組み込めるか、現行環境の整理から連携方式の選定、PoC設計、本番導入まで支援します。

Start Challenge Consulting
info@startchg.com
担当：平出

注記／提案：参照系から段階導入し、更新系は承認とガードレールを前提に拡張する。詳細はP21の問い合わせ窓口まで。

Contact

お問い合わせ先

現行環境の整理から連携方式の選定、PoC設計、本番導入まで一貫して支援します。API・MCPの適用可否の初期診断についても、下記窓口までお問い合わせください。

Start Challenge
Consulting co., ltd.
挑む力が、世界を照らす

Start Challenge Consulting co., ltd.

API・MCP連携／生成AI業務実装 支援

項目	内容
問い合わせ先	Start Challenge Consulting
メールアドレス	info@startchg.com
担当	平出

SUPPORT MENU

現状整理・連携方式の選定／MCPサーバー構築／PoC設計・実施／セキュリティ設計・運用支援

注記／本資料の内容に関するお問い合わせは上記窓口まで。記載内容は発行時点の情報に基づく。